

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra teorie obvodů



Diplomová práce

Monitor sběrnice Foundation Fieldbus

Bc. Monika Svědřihová

Vedoucí práce: Ing. Jiří Závada

Studijní program: Biomedicínské inženýrství a informatika

Obor: Biomedicínské inženýrství

12. května 2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Monika Svědřihová
Studijní program: Biomedicínské inženýrství a informatika (magisterský)
Obor: Biomedicínské inženýrství
Název tématu: Monitor sběrnice Foundation Fieldbus

Pokyny pro vypracování:

Zjistěte, v jakých zařízeních se průmyslová sběrnice Foundation Fieldbus používá a diskutujte možnosti využití této sběrnice ve zdravotnictví. Navrhněte a realizujte monitor průmyslové sběrnice Foundation Fieldbus definované normou IEC 61158 a dokumenty vydanými organizací Fieldbus Foundation. Monitor se bude skládat z následujících dvou částí propojených sběrnicí USB:

- hardwarová sonda s funkcí digitalizace signálu na sběrnici
- software pro osobní počítač s operačním systémem Microsoft Windows pro zobrazování obsahu datových rámců a průběhu linkového signálu na sběrnici v grafické formě – datové rámce budou zapisovány do tabulky a průběh signálu do okna ve formě podobné obrazovce osciloskopu.

Práce zahrnuje tyto činnosti:

- analýza, během níž získáte informace o jednotlivých vrstvách sběrnice Foundation Fieldbus
- návrh, který bude obsahovat popis způsobu realizace monitoru
- realizace návrhu
- testování, během něhož ověříte výsledky diplomové práce.

Seznam odborné literatury:

[1] IEC 61158-2 Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 2: Physical layer specification and service definition. IEC, Switzerland, 2003.

[2] <http://www.fieldbus-international.com/>

[3] <http://www.fieldbus.org/>

Vedoucí diplomové práce: Ing. Jiří Závada

Platnost zadání: do konce letního semestru 2014/2015

L.S.

prof. Ing. Pavel Sovka, CSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 10. 1. 2014

Prohlášení

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

V Praze dne 12. 5. 2014

.....

Poděkování

Přednostně bych chtěla poděkovat vedoucímu diplomové práce Ing. Jiřímu Zavadovi a kolegům z práce za cenné rady a připomínky při psaní této diplomové práce. Dále bych chtěla poděkovat příteli a rodině za podporu a pevné nervy.

Abstract

The aim of the thesis is to design and realize a bus monitor for Foundation Fieldbus. The thesis describes automation in healthcare, the bus Foundation Fieldbus itself, design and realization of hardware probe and development of software to display the data frames including generating waveforms. The result of the thesis is a comprehensive system including a hardware probe. The system is able to decode the data frames published on the bus. Moreover, it is capable of a detailed analysis of individual frames by decoding the frames and generate a waveform from the decoded data. The system is useful for testing or debugging the communication on Foundation Fieldbus.

Abstrakt

Cílem této diplomové práce bylo navrhnout a realizovat monitor sběrnice Foundation Fieldbus. Práce popisuje automatizaci ve zdravotnictví, sběrnici Foundation Fieldbus, návrh i realizaci hardwarové sondy a vývoj softwaru pro zobrazení rámců včetně průběhu signálu. Výsledkem práce je komplexní systém včetně hardwarové sondy, schopný vypisovat datové rámce publikované na sběrnici Foundation Fieldbus. Následně umožňuje podrobnou analýzu jednotlivých rámců v podobě dekodování rámců a zobrazení jejich průběhu. Daný systém je využitelný při testování či ověření komunikace na sběrnici Foundation Fieldbus.

Obsah

1	Úvod	1
2	Automatizace ve zdravotnictví	2
2.1	Senzory komunikujících po sběrnici Foundation Fieldbus	3
2.1.1	Liquiline M CM42 – pro měření pH	3
2.1.2	Radarový hladinoměr Sitrans LR250	3
2.1.3	Snímač tlaku VEGABAR 55	4
2.1.4	Provozní kalibrátor a komunikátor Beamex MC6	5
3	Foundation Fieldbus	6
3.1	Komunikační protokoly	7
3.2	Komunikační model	7
3.2.1	Fyzická vrstva	8
3.2.2	Komunikační zásobník	9
3.2.3	Uživatelská vrstva	9
3.3	Komunikace po sběrnici	10
3.3.1	Plánovaný přenos dat	10
3.3.2	Neplánovaný přenos dat	11
3.3.3	Popis komunikace, znázorněné na obrázku 3.5	12
3.4	Struktura rámců	13
3.4.1	Frame control	13
3.4.2	Zdrojová a cílová adresa (Destination and source address)	14
3.4.3	Parametry	14
3.4.4	Kontrolní součet	16
4	Hardware FF Monitoru	18
4.1	Analýza	18
4.2	Návrh Hardware	18
4.2.1	Filtrace	22
4.3	Realizace	24
4.3.1	Návrh plošného spoje	24
5	Firmware Monitoru	27
5.1	Obsluha A/D převodníku	27
5.2	Obsluha USB sběrnice	28
5.3	Vývojové prostředí IAR Embedded Workbench	29

6	Software FF Monitoru	30
6.1	Analýza	30
6.1.1	Programovací jazyk Java	30
6.1.2	Knihovna JChart2D	31
6.2	Návrh software	31
6.2.1	UsbDataStream	31
6.2.2	Obslužný program FFpipe64	32
6.2.2.1	Dekódování rámců na sběrnici	32
6.2.3	DataRecevier	35
6.2.4	Třídy Frame a DataHolder	35
6.2.5	Grafické uživatelské rozhraní FF Monitoru	36
6.2.6	Třída FrameDecode	36
6.2.7	Zobrazení průběhu rámce	37
6.3	Implementace	37
6.3.1	Vývojové prostředí Netbeans IDE	37
6.3.2	JNI (Java Native Interface)	38
7	Testování a výsledky	39
7.1	Hardwarová sonda	39
7.2	Monitor Foundation Fieldbus	40
7.2.1	Detekování rámců	40
7.2.2	Dekódování rámců	40
7.2.3	Analýza průběhu rámce	42
8	Závěr	43
9	Příloha A	48
9.1	Dokumentace k hardwarové sondě FF Monitoru	48
10	Příloha B	50
10.1	Dekódování hlavičky rámce	50
11	Příloha C	52
11.1	Vytvoření a nastavení grafu	52
12	Příloha D	54
12.1	Obsah přiloženého CD	54

Seznam obrázků

2.1	Převodník pro měření PH Liquiline M CM42, převzato [10]	3
2.2	Radarový hladinoměr Sitrans LR250, převzato [11]	4
2.3	Snímač Vegabar a jeho použití, převzato [13]	4
2.4	Provozní kalibrátor a komunikátor Beamex MC6, převzato [21]	5
3.1	Ukázka implementace protokolů, převzato [20]	7
3.2	Uspořádání vrstev OSI, H1 a HSE modelu	8
3.3	Manchester kódování, převzato [25]	9
3.4	Plánovaný přenos dat	11
3.5	Ukázka průběhu komunikace na sběrnici	12
3.6	Přehled rámců FF sběrnice a jejich dekódování dle frame control	14
3.7	Ukázka výpočtu CRC	17
4.1	Průběh signálu na sběrnici Foundation Fieldbus	19
4.2	Blokové schéma zpracování signálu	19
4.3	Vnitřní zapojení AD623, dle katalogového listu [18]	21
4.4	Schéma zapojení přístrojového zesilovače AD623	21
4.5	Zapojení, frekvenční a fázová charakteristika filtru typu HP	22
4.6	Zapojení symetrického filtru typu DP	24
4.7	Porovnání původního a upraveného signálu	25
4.8	Vrchní strana desky plošných spojů, zobrazena v měřítku 2:1	25
4.9	Vyrobená a osazená hardwarová sonda FF Monitoru	26
5.1	Ukázka získávání dat od A/D převodníku	28
6.1	Blokové schéma návrhu softwaru	31
6.2	Struktura jednotlivých sekvencí na sběrnici	33
6.3	Stavový automat na dekódování rámců	34
6.4	Blokový diagram receiveru	35
6.5	Hlavní okno grafického uživatelského rozhraní	36
6.6	Zobrazení grafu s možnostmi nastavení	38
6.7	Ukázka implementace JNI, převzato [7]	38
7.1	Porovnání původního a upraveného signálu	39
7.2	Dekódování rámce PN programem NI-FBUS Monitor	41
7.3	Dekódování rámce PN programem FF Monitor	41

7.4	Jednotlivé pulzy rámce v programu FF Monitor	42
7.5	Jednotlivé pulzy upraveného signálu	42
9.1	Schéma zapojení hardwarové sondy	48
9.2	První prototyp realizace	49
9.3	Vrchní a spodní strana desky plošných spojů, zobrazena v měřítku 1:1	49

Seznam tabulek

3.1	Virtuální komunikační relace	10
3.2	Obecná struktura rámce na sběrnici	13
3.3	Struktura rámce CD	13
3.4	Priorita rámce	15
3.5	Formát adres	15
3.6	Parametry PN rámce	15
3.7	Pravděpodobnostní tabulka XOR	16
12.1	Obsah přiloženého CD	54

Kapitola 1

Úvod

V dnešním světě rychlého růstu technického pokroku a velké spotřeby výrobků v elektronickém, potravinářském, ale třeba i farmaceutickém průmyslu si již skoro nedokážeme představit výrobu bez automatizace výrobní linky. Tento velký vzestup automatizace výroby způsobil rozmach průmyslových sběrnic. Z důvodu rostoucích nároků vznikaly nové standardy a specifikace, pro větší, robustnější a hlavně bezpečnější provoz automatizačních linek.

Jedna z nejrozšířenějších průmyslových sběrnic, která vznikla právě na základě snahy standardizovat a upřesnit průmyslové sběrnice zejména v nebezpečném prostředí se nazývá Foundation Fieldbus. Dle článku [5] se jednalo již v roce 2006 o jednu z populárních průmyslových sběrnic ve zpracovatelském průmyslu. Studie naznačuje, že prodané systémy komunikující pomocí sběrnice Foundation Fieldbus se až dvěma třetinami podíleli na celkových tržbách průmyslových komunikačních systémů.

Cílem této diplomové práce vznikla v rámci spolupráce s firmou Siemens oddělením CTDC je realizovat monitor průmyslové sběrnice Foundation Fieldbus, který umožní přijímání dat, jejich rozkódování a analýzu jednotlivých rámců posílaných po sběrnici. Navržený systém by měl ulehčit vývoj a testování různých prototypů zařízení, komunikujících po sběrnici Foundation Fieldbus.

Součástí diplomové práce je popis průmyslové sběrnice Foundation Fieldbus, její použití zejména ve farmaceutickém průmyslu, návrh a vytvoření hardwaru pro příjem dat, návrh firmwaru na zpracování a odeslání dat do PC a v poslední řadě vytvoření přehledného grafického uživatelského rozhraní, které umožní vypsání rámců vysílaných po sběrnici, dekódování a zobrazení průběhu signálu daného rámce.

Kapitola 2

Automatizace ve zdravotnictví

Automatizaci ve zdravotnictví můžeme najít hned v několika formách. Nejčastěji se s ní setkáme při výrobě léčiv ve farmaceutickém průmyslu, či při výrobě lékařských přístrojů. Výrobci ve farmaceutickém průmyslu vyrábějí denně léky za miliony dolarů, ale zároveň potřebují mít možnost rychlé změny výrobního postupu. To umožňuje takzvaná šaržově (dávkově) koncipovaná automatizace využívající tzv. Batch nadstavbu, která kromě genetování receptur nabízí mnoho souvisejících funkcí, například elektronické podepisování zásahů prováděných operátory, zajištění automatického protokolování všech výrobních kroků a technických veličin do elektronického záznamu. Právě dokumentace správného výrobního postupu je v současnosti hlavním požadavkem výrobců léčiv a potravin.

Příkladem použití této technologie může být automatizace nové biotovárny ve firmě Lonza Biotenic, s.r.o v Kouřimi, která se zabývá výrobou biotechnologických produktů pro potravinářský a farmaceutický průmysl. Jak uvádí Ing. Vlastimil Braun v článku [3]: „Zde se pracuje se živou hmotou, která se nemusí za stejných podmínek chovat vždy stejně. Ačkoliv je možné pro různé produkty vytvořit různé receptury nastavení regulací, přesto mají operátoři většinou nezvykle velkou volnost při zásahu do nastavení žádaných hodnot regulací, nebo je jim v určitých případech umožněno i přímé ruční řízení. Batch systém proto musí umožňovat jak automatické, tak i poloautomatické nebo ruční řízení s přepínáním těchto režimů v libovolném okamžiku výrobního postupu. Tento komfort také zahrnuje pozastavení výrobního kroku (tzv. fáze), přerušení fáze, její dokončení, resp. přeskočení do dalšího výrobního kroku.“

Batch technologii můžeme využít například prostřednictvím řídicího systému SIMATIC PCS7, který nabízí přímo doplňkové balíky SIMATIC BATCH a umožňuje integraci inteligentních polních přístrojů pomocí průmyslových standardů Profibus, Foundation Fieldbus a HART. Díky této integraci umožňuje používat obrovské množství různých senzorů komunikujících po těchto sběrnicích. Obecně ve zdravotnictví, zvláště pak v farmaceutickém průmyslu se sběrnice FF využívá jako systémová sběrnice pro komunikaci automatizačních čidel různých fyzikálních veličin, např. teploty, tlaku, pH, napětí apod. V další části jsou uvedeny příklady senzorů komunikujících po sběrnici Foundation Fieldbus a zároveň splňující náročné podmínky farmaceutického průmyslu.

2.1 Senzory komunikujících po sběrnici Foundation Fieldbus

2.1.1 Liquiline M CM42 – pro měření pH

Při výrobě produktů s definovanými parametry (například při výrobě léčiv, chemických sloučenin) je důležité kontrolovat hodnotu pH jednotlivých látek. Protože přesná hodnota pH napomáhá k požadovaným parametrům výsledných produktů.

K měření pH může být použit dvou vodičový převodník, který zvládne měřit pH v rozsahu -2 až 16 . Dále umožňuje měření oxidačně redukční potenciálu (ORP) v rozsahu od $-2\ 000$ do $+2\ 000$ mV, vodivosti, koncentrace, množství rozpuštěného kyslíku a teploty. Velkou výhodou použití tohoto přístroje je snadná obsluha zařízení, možnost výměny měřené proměnné díky výměnným modulům a jeho celková odolnost i v prostředí s nebezpečím výbuchu. Převodník Liquiline M je k dispozici v digitální nebo analogové verzi. V digitální verzi může komunikovat prostřednictvím protokolu HART, sběrnice Profibus-PA nebo Foundation Fieldbus. Může být zapojen do nových nebo již existujících měřících a řídicích systémů.

Jde o první převodník pro analytická měření vyvinutý v souladu s mezinárodní normou pro funkční bezpečnost elektrických a elektronických programovatelných přístrojů IEC 61508. Může tedy být použit i jako součást systémů zajišťujících funkční bezpečnost technologických procesů (Safety Instrumented System – SIS), a to na úrovni SIL 2 (Safety Integrity Level). Pouzdro z plastu i jeho hygienická varianta z korozivzdorné oceli, odolné proti chemickým vlivům, jsou navrženy s ohledem na provozní podmínky vyskytující se v běžných i nebezpečných prostorách výrobních provozů především v chemické výrobě, farmaceutickém průmyslu, potravinářství apod. Celý přístroj je podrobně popsán v článku [4].

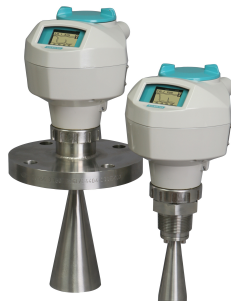


Obrázek 2.1: Převodník pro měření PH Liquiline M CM42, převzato [10]

2.1.2 Radarový hladinoměr Sitrans LR250

Pro kontinuální měření hladiny kapalin a kalů, můžeme využít pulzní radarový hladinoměr Sitrans LR250, který pracuje na frekvenci 25 GHz. Vysoká frekvence umožňuje použít antény s velmi malým rozměrem: 40, 48, 75, popř. 95 mm. V kombinaci s malým vyzářovacím úhlem 8° je hladinoměr vhodný pro měření médií s nízkou dielektrickou konstantou nebo pro velmi přesném měření v malých zásobnících. Rozsah měření je velmi široký: od 50 mm (měřeno od antény) až po velké zásobníky o velikosti 20 m. Elektronika je v hlavici pevně zapouzdřena, takže přístroj spolehlivě odolává vibracím a rázům. Přístroj je konstruován

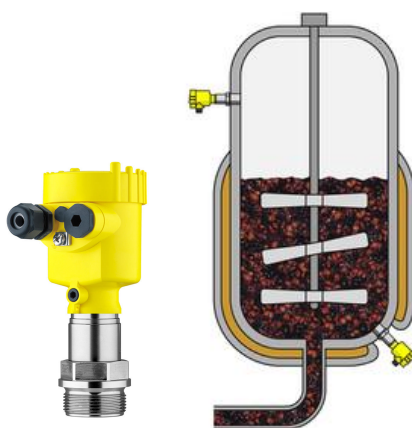
pro provoz v náročných podmínkách s teplotou až do 200 °C, tlakem až do 4 MPa a je v krytí IP68. Měření hladiny kapalin a kalů je možné zahájit bezprostředně po uvedení přístroje do provozu. Sitrans LR250 má univerzální použití. Je vhodný zejména pro měření hladin v chemickém, farmaceutickém a petrochemickém průmyslu, a to i v korozivním prostředí.[28]



Obrázek 2.2: Radarový hladinoměr Sitrans LR250, převzato [11]

2.1.3 Snímač tlaku VEGABAR 55

Pro měření relativního nebo absolutního tlaku můžeme využít snímač tlaku od firmy Vegabar. Jedná se o převodník tlaku s membránou METEC, která se skládá z vestavěného keramického kapacitního tlakoměrného čidla se speciálním izolačním systémem s kovovou oddělovací membránou. Snímač zajišťuje měření v rozsahu -1 až 25 bar (-100 až 2500 kPa) s přesností $\pm 0,075\%$ a provozní teplotou -12 až 200 °C. Standardní výstupní signál ze snímače je proudový, 4 až 20 mA, komunikuje po sběnicích HART, Profibus PA, Foundation Fieldbus. Snímače splňují náročné podmínky pro měření hladiny a tlaku v potravinářství, farmacii i chemickém průmyslu. V těchto oborech se výborně uplatní nízká drsnost povrchu smáčených částí $Ra < 0,5 \mu\text{m}$, splněný hygienický standard (EHEDG, 3A) a vynikající odolnost proti teplotním rázům [14].



Obrázek 2.3: Snímač Vegabar a jeho použití, převzato [13]

2.1.4 Provozní kalibrátor a komunikátor Beamex MC6

Beamex MC6 je vysoce přesný provozní kalibrátor a komunikátor, který umožňuje kalibraci tlaku, teploty a různých elektrických signálů. Umožňuje několik provozních režimů: Měřicí přístroj, Kalibrátor, Dokumentační kalibrátor, Záznamník dat (datalogger) a Sběrníkový komunikátor. Přístroj MC6 je schopný komunikovat po průmyslových sběrnících HART, Foundation Fieldbus a Profibus PA. Mezi jeho výborné vlastnosti patří například robustnost, odolnost vůči prachu, voděodolná skříňka s krytím IP65, ergonomická konstrukce, nízká hmotnost a hlavně snadná ovladatelnost. Díky těmto vlastnostem již našel uplatnění různých odvětvích průmyslu, jako je farmacie, energetika, plynárenství, potravinářství, chemie a mnoho dalších [21].



Obrázek 2.4: Provozní kalibrátor a komunikátor Beamex MC6, převzato [21]

Kapitola 3

Foundation Fieldbus

System průmyslové sběrnice označovaný Foundation Fieldbus (FF) vznikl ve snaze vnést stabilizaci do nepřehledné oblasti systémů pro průmyslovou komunikaci. Vývojem standardu se zabývá nezisková stejnojmenná organizace Fieldbus Foundation (sdružující významné výrobce z oblasti regulační a automatizační techniky), která od základu navrhovala daný systém na podporu kritických kontrolních aplikací. Standardizace podle uznávaných mezinárodních řídicích orgánů byla [20]:

- v září 1992 American National Standards Institute (ANSI) / International Society of Automation (ISA),
- v prosinci 1999 Mezinárodní elektrotechnická komise (IEC),
- v březnu 2000 Evropský výbor pro normalizaci v elektrotechnice (CENELEC).

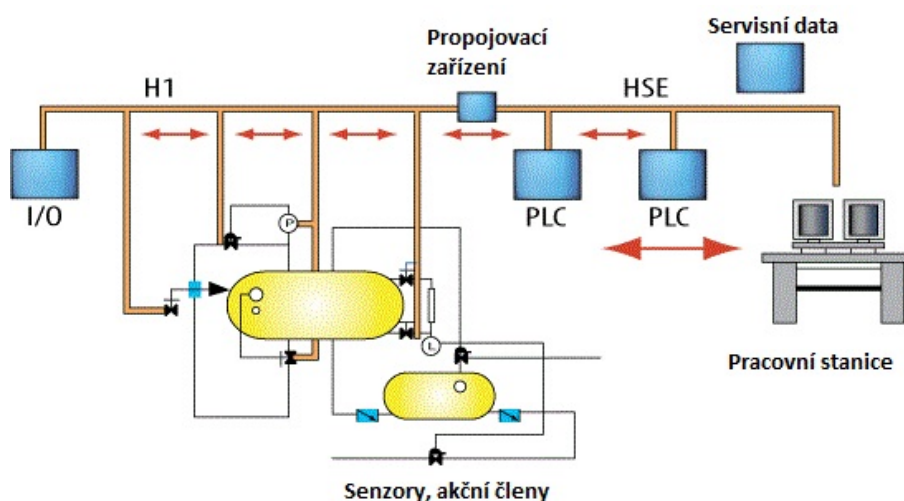
Průmyslová sběrnice Foundation Fieldbus je číslicová sběrnice s poloduplexním sériovým přenosem, určená ke komunikaci mezi regulačními prvky (senzory veličin, akční členy) a řídicími automaty. Významné rysy sběrnice Foundation Fieldbus [26]:

- standardizované fyzické rozhraní, napájení koncových zařízení po komunikačním vedení, aplikovatelnost ve výbušném výrobním prostředí,
- zachovává základní výhody dosavadních analogových komunikačních systémů s proudovou smyčkou 4 -20 mA, instalace s využitím původní kabeláže,
- umožňuje na jeden pár vodičů připojit více jednotek,
- rychlá diagnostika jednotek a jejich propojení,
- implementace principů distribuovaného řízení, snižuje nároky na výkonnost a počet řídicích terminálů.

3.1 Komunikační protokoly

Sběrnice Foundation Fieldbus (FF) je definována dvěma komunikačními protokoly. Protokol H1 je určen pro propojení průmyslových prvků (senzory, převodníky a akční členy) s komunikační rychlostí 31,25 kbit/s. Zajišťuje napájení připojených zařízení po komunikačním vedení s omezeným odběrem proudu. Zařízení lze připojit i odpojit za běhu (plug and play). Umožňuje připojit až 32 zařízení na jeden segment s maximální délkou 1900 m. Sběrnice je zakončena terminátory a je doporučena stíněná kabeláž.

Druhým protokolem je HSE (High Speed Ethernet), který zajišťuje vysokorychlostní propojení (100/1000 Mbit/s) pracovních stanic. HSE je založen na klasické technologii ethernet a je kompatibilní s klasickými prvky sítě Ethernet. Řízení kolizí je zajišťováno metodou CSMA/CD (stanice při svém vysílání současně kontroluje přenosové médium, zda nezachytí jiné vysílání, které koliduje s jejím). Množství kolizí je minimalizováno plánováním vysílacích časových oken každého zařízení Foundation Fieldbus. Protokol HSE dokáže podporovat logické funkce prováděné v programovatelných automatech (PLC) i datově náročná zařízení (například analyzátoři a síťové brány). HSE zlepšuje přístup k průmyslovým sběrnicím za pomoci propojovacího zařízení a umožňuje vysokorychlostní rozšíření pro automatizační zařízení. Zároveň podporuje standardní Ethernetové propojení (možnost zařízení zdvojit), včetně možnosti použití optické kabeláže [20], [16]. Ukázka implementace je na obrázku 3.1.

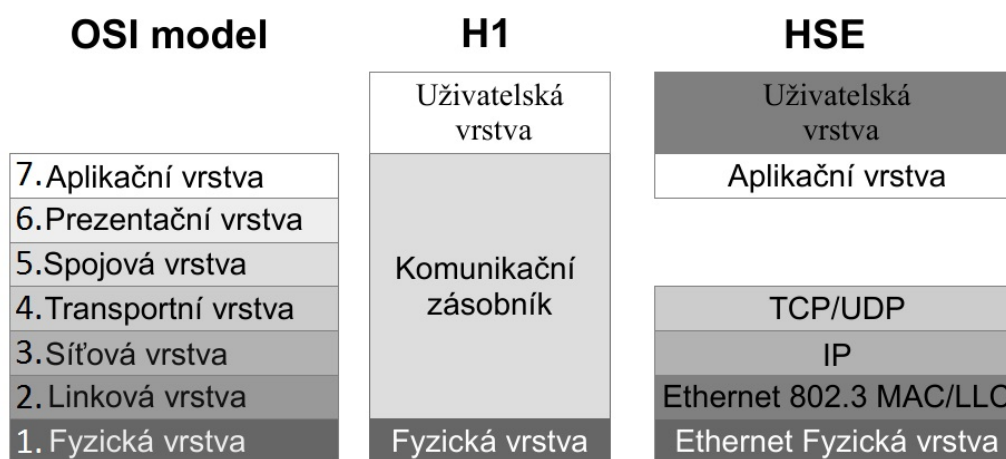


Obrázek 3.1: Ukázka implementace protokolů, převzato [20]

3.2 Komunikační model

Technická specifikace komunikačního modelu je založená na mezinárodním standardu ISO/OSI modelu, skládající se ze tří základních vrstev: fyzická vrstva (Physical Layer), komunikační zásobník (Communication Stack) a uživatelská vrstva (User Layer). Fyzická vrstva odpovídá fyzické vrstvě ISO/OSI modelu, přijímá kódované zprávy z horních vrstev

a převádí je do fyzických signálů na sběrnici. Komunikační zásobník nahrazuje vrstvy 2 a 7, linková vrstva řídí přenos zpráv na sběrnici a Aplikační vrstva kóduje a dekóduje příkazy uživatele. Vrstvy 3, 4, 5 a 6 OSI modelu nejsou ve sběrnici použity. Uspořádání vrstev je znázorněno na obrázku 3.2.



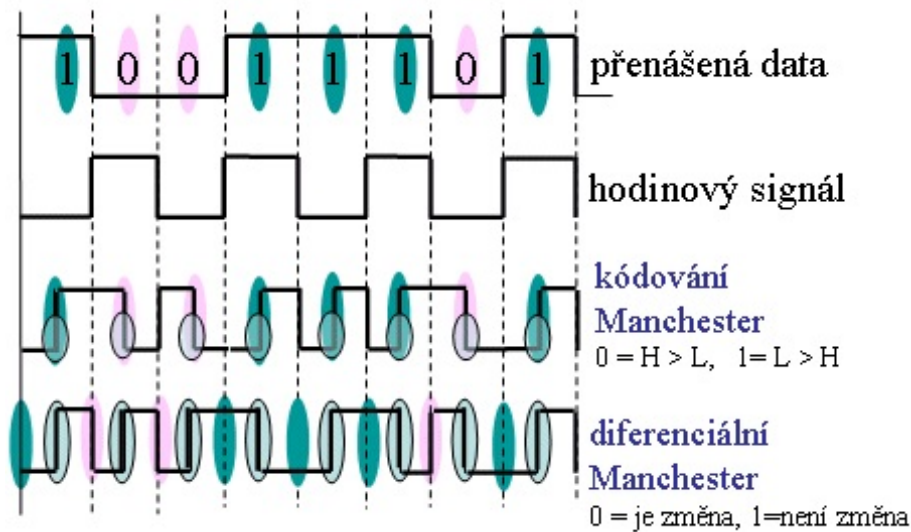
Obrázek 3.2: Uspořádání vrstev OSI, H1 a HSE modelu

3.2.1 Fyzická vrstva

Fyzická vrstva připravuje data z vyšších vrstev na přenos po sběrnici, specifikuje parametry pro kódování, napěťové úrovně a napájení. K datům přidává synchronizační znaky:

- synchronizační hlavičku (Preamble) – synchronizuje hodiny přijímače a vysílače,
- začátek rámce (Start Frame) – ohraničení začátku posílaných dat,
- konec rámce (End Frame) – označuje konec vysílaných dat.

Protokol H1 používá synchronizační kódování Manchester, které slučuje hodinový a datový signál. V každém bitovém intervalu dochází nejméně k jedné změně signálu. Změna uprostřed bitového intervalu nese užitečná data, jde-li z vysoké logické úrovně do nízké (sestupná hrana) reprezentuje logickou hodnotu „0“, vzestupná logickou hodnotu „1“. Současně tato změna slouží k potřebám synchronizace. Pokud se logická hodnota nemění je reprezentována stejnou hranou. Princip Manchester kódování je znázorněn na obrázku 3.3.



Obrázek 3.3: Manchester kódování, převzato [25]

3.2.2 Komunikační zásobník

Komunikační zásobník můžeme rozdělit na linkovou a aplikační vrstvu..

Linková vrstva (Data Link layer) rozděluje data do rámců, předává je fyzické vrstvě k odeslání, provádí kontrolu chyb rámců a řídí přístup ke sběrnici. Kontroly a plány komunikace na sběrnici zajišťuje zařízení Link Active Scheduler (LAS), které periodicky vysílá synchronizační signál na sběrnici. Zařízení LAS se zvolí ze zařízení typu Link Masters (LM). Pokud se na sběrnici připojí více LM současně zařízením LAS se stává ten s nejnižší adresou.

Aplikační vrstva (Application layer) se dělí na dvě podvrstvy Fieldbus Access Sublayer (FAS) a Fieldbus Message Specification (FMS), které tvoří rozhraní mezi linkovou a uživatelskou vrstvou. Vrstva FAS tvoří virtuální komunikační relaci (Virtual Communication Relationships – VCR), která popisuje jednotlivé parametry vrstvy FAS a tím umožňuje související činnosti zpracovávat rychleji. Foundation Filedbus využívá tři různé VCR, vypsané v tabulce 3.1.

Fieldbus Message Specification (FMS) poskytuje služby pro standardizovanou komunikaci. Pro jednotné a jasné zadání jsou použity popisy objektů, obsahující definice všech standardních formátů přenosu zprávy a zahrnující specifické údaje dané aplikace. Pro každý typ objektu existují speciální předdefinované komunikační služby.

3.2.3 Uživatelská vrstva

Jedním z nejdůležitějších kritérií průmyslových sběrnic, které se mají uplatnit na trhu, je jejich interoperabilita zařízení. Interoperabilita je schopnost zařízení komunikovat i se zařízeními od jiných výrobců. To vyžaduje otevřenou specifikaci protokolu, která definuje jednotné funkce zařízení a aplikační rozhraní. FF sběrnice přiřazuje všechny funkce i data zařízení do tří různých bloků:

KLIENT/ SERVER	Oznámení distribuce (Report distribution)	Vysílač/ Přijímač (Publisher/ Subscriber)
Operátor komunikace	Upozornění na události, alarmy	Publikace dat
Nahrávání a Stahování dat Nastavuje změny Dálková diagnostika Přístup k parametrům Zobrazení dat	Posílá alarmy procesů operátorovi konzole	Posílá aktuální hodnotu vysílače do PID bloku

Tabulka 3.1: Virtuální komunikační relace

- Zdrojový blok (Resource block) obsahuje parametry zařízení (název, výrobce, sériové číslo, verze hardwaru a softwaru...).
- Funkční blok (Function block) popisuje jednotlivé funkce a přístup k zařízení.
- Převodní blok (Transducer block) odděluje funkční bloky od fyzického rozhraní akčních členů a snímačů. Snímače umožňují rozšířit aplikační možnosti zařízení, ovlivňují vstupní a výstupní parametry funkčních bloků. Mohou být použity pro kalibraci, vynulování měření, linearizaci charakteristiky apod.

Kromě těchto tří bloků, existuje řada pomocných objektů. Linkové objekty (Links objects) definují spojení mezi funkčními bloky. Zobrazované parametry (View objects) obsahující nadefinované parametry uživatelem, které chce zobrazit. Výstražné objekty (Alert objects) umožňující vytvořit a odeslat varovnou zprávu zařízení. Časové objekty (Trend objects) uchovávají časovou značku hodnot pro reprezentaci stavu v grafu. Při popisu komunikačního modelu bylo čerpáno z literatury [15] a [16].

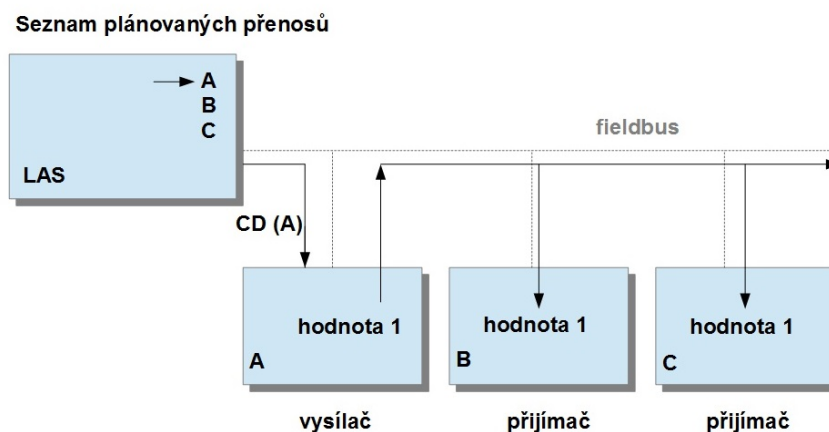
3.3 Komunikace po sběrnici

Komunikace na sběrnici Foundation Fieldbus využívá plánovaný (scheduled) a neplánovaný (unscheduled) přenos dat. Důležité (časově kritické) příkazy, které mají na starosti například řízení procesů jsou prováděny výhradně pomocí plánované komunikace. Do neplánovaného přenosu jsou zahrnuty například příkazy pro nastavení parametrů a různé diagnostické funkce.

3.3.1 Plánovaný přenos dat

Všechny časově kritické příkazy jsou založeny na přenosovém plánu, který vytvořil provozovatel systému při konfiguraci FF systému. Link Active Scheduler (LAS) periodicky vysílá synchronizační signál (TD: Time Distribution) na sběrnici. Jedná se o deterministický

systém, kde okamžik i pořadí vysílání jsou pevně stanoveny. Pokud je zařízení vyzváno, aby zveřejnilo své naměřené údaje, LAS vydá příkaz "vyslat data" (CD: Compel Data), zařízení po obdržení CD příkazu, publikuje data do vyrovnávací paměti, kde je posluchač této zprávy (například regulační ventil) vyhodnotí a zareaguje. Tato situace je přehledně znázorněna na obrázku 3.4 a způsob plánování komunikace je zobrazen na obrázku 3.5.



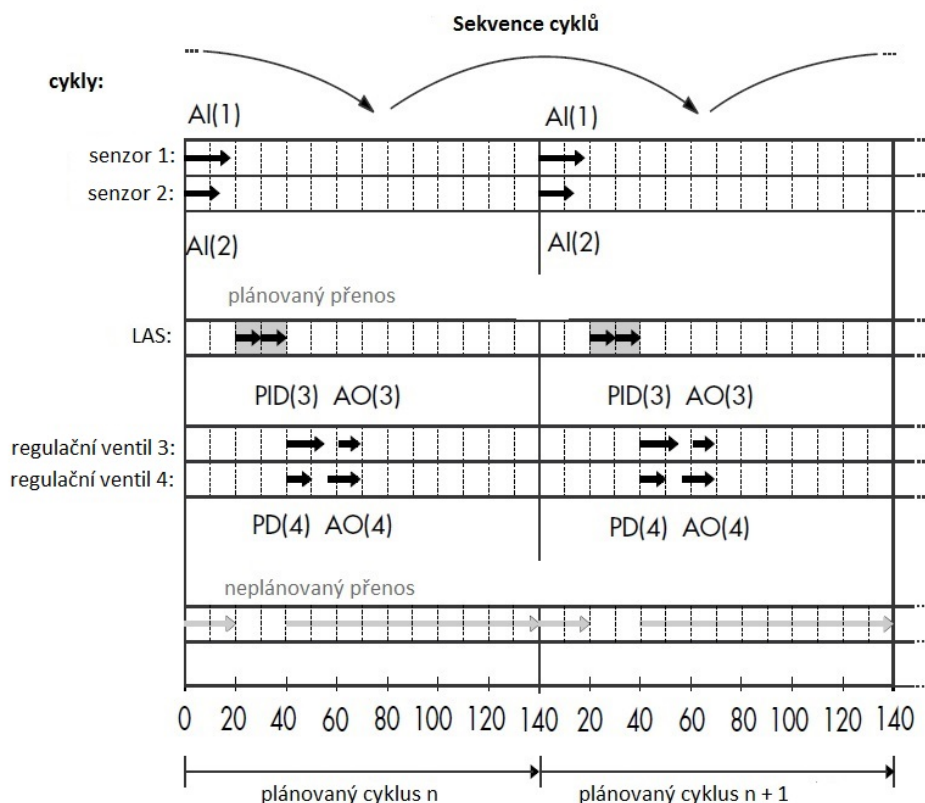
Obrázek 3.4: Plánovaný přenos dat

Každá regulační smyčka přistupuje ke sběrnici pouze jednou na nezbytně krátkou dobu, což umožňuje využití sběrnice pro mnoho dalších smyček a činností. Zmíněný přístup ukazuje výhody distribuované řídicí strategie, která snižuje počet datových přenosů na minimum.

3.3.2 Neplánovaný přenos dat

Do neplánovaného přenosu dat řadíme příkazy na vyžádání například parametry a diagnostiku zařízení, které nejsou časově kritické (nemají zásadní vliv na systém). Neplánovaná komunikace je omezena na přestávky plánovaného vysílání, její povolení umožňuje zařízení LAS, pomocí tokenu PT (Pass Token). Pass token je vyslán všem zařízením z aktivního seznamu, který spravuje LAS. Dané zařízení může využít sběrnici pro neplánovaný přenos dokud je potřeba. Nesmí však překročit maximální dobu platnosti tokenu, která určuje dobu neplánované komunikace.

Aktivní seznam zařízení je průběžně aktualizován, LAS vyšle speciální příkaz PN (Probe Node) na hledání nových zařízení na adresy, které nejsou v aktivním seznamu, jestliže zařízení odpoví příkazem PR (Probe Response) znamená to, že předtím obdržel token pro neplánovanou komunikaci a LAS přidá zařízení do aktivního seznamu. Pokud zařízení na příkaz PT nereaguje a nebo vrací token již tři po sobě jdoucí cykly, pak je z aktivního seznamu odstraněno. Změny o přidání či odebrání zařízení ze seznamu pošle LAS všem zařízením, aby každý Link Master měl aktuální kopii seznamu a mohl se kdykoliv stát LAS. [15]



Obrázek 3.5: Ukázka průběhu komunikace na sběrnici

3.3.3 Popis komunikace, znázorněné na obrázku 3.5

- v čase 0, senzor 1 a 2 zahajují měření,
- v čase 20, LAS vyzve senzor 1 o vyslání naměřených údajů, které si může přečíst PID regulátor přidruženého ventilu,
- v čase 30, LAS vyzve senzor 2 o vyslání naměřených údajů, které si může přečíst PID regulátor přidruženého ventilu,
- v čase 40, jsou data ze senzorů zpracována regulátory,
- v čase 57, regulační ventil 4 reaguje na dané hodnoty,
- v čase 62, regulační ventil 3 reaguje na dané hodnoty,
- po 140 časových intervalech se posloupnost příkazů opakuje.

3.4 Struktura rámců

Některé rámce byly již zmiňovány v předchozí podkapitole při vysvětlování plánované a neplánované komunikace, v této podkapitole bude ukázána jejich struktura a celkový přehled všech rámců. Struktura rámce a jeho délka závisí na typu daného rámce a jeho funkci. Struktura rámců na sběrnici je dána průchodem vrstvami komunikačního modelu, každá vrstva přidává rámci specifické informace. Například fyzická vrstva přidává synchronizační značky, linková vrstva připojuje vypočtený kontrolní součet celého rámce. V tabulce 3.3 je ukázána obecná struktura rámce na sběrnici a velikost každé sekvence. Začátkem vysílání je vždy hlavička (preamble), následuje počáteční oddělovač rámce (Start delimiter), samotný rámec (frame), který má velikost dle funkce (obsahu dat). Pro kontrolu přenosu se zaslá kontrolní součet (CRC - Cyclic redundancy check), dále následuje konečný oddělovač (end delimiter) a zákaz vysílání na sběrnici (Post TX Gab). Oba oddělovače mají vždy pevně daný počet i sekvenci jednotlivých bitů, to usnadňuje detekování rámce na sběrnici. Obecná struktura rámce na sběrnici je ukázána v tabulce 3.3.

Hlavička	Počáteční oddělovač	Rámec	Kontrolní součet	Konečný oddělovač	Zákaz vysílání
1B	1B	XB	2B	1B	1B

Tabulka 3.2: Obecná struktura rámce na sběrnici

Jak již bylo zmíněno v podkapitole 3.3.1 pokud chce LAS požádat zařízení o zaslání dat vyšle rámec CD (Compel Data). Tento rámec můžeme rozdělit na dva typy CD1 (obsahuje i adresu zdroje) a CD2 (obsahuje jen adresu cíle). První bajt rámce se nazývá Frame control, v tomto bajtu je zakódován: typ rámce v našem případě CD, jeho priorita, formát cílových, případně zdrojových adres a identifikační bit zda se jedná o finální token. Následuje cílová případně zdrojová adresa a parametry. Zatímco formát parametrů se liší podle jejich druhu, formát Frame control, cílové i zdrojové adresy se neliší, takže níže uvedené rozkódování platí pro všechny rámce.

Format	Frame control	Cílová adresa	Zdrojová adresa	Parametry
1L	1111 1FPP	HL.N.S	HL.N.S	o-SD-p
1S	1111 0FPP	N.S	N.S	o-SD-p
2L	1011 1FPP	HL.N.S	-	o-SD-p
2S	1011 0FPP	N.S	-	o-SD-p

Tabulka 3.3: Struktura rámce CD

3.4.1 Frame control

Frame control obsahuje 8 bitů. Horní čtyři bity slouží k prvotnímu rozkódování typu rámce, strukturu dolních 4 bitů lze obecně napsat ve tvaru LFPP, kde L udává délku jednot-

livých adres, pokud odpovídá logické hodnotě "1" jedná se o dlouhý formát adresy (H.L.N.S) v opačném případě se jedná krátký formát adresy (N.S). Tyto formáty budou vysvětleny v podkapitole 3.4.2. Bit F určuje zda se jedná o finální token dle dvojice bitů PP, rozepsaných v tabulce 3.4 poznáme prioritu rámce. Na obrázku 3.6 je znázorněn přehled všech rámců a jejich prvotní rozkódování podle prvního bajtu (Frame control).

7	6	5	4	3	2	1	0	Rámec
15 (0xF)				L	F	0	0	EC1 (Establish Connection)
				L	F	P	P	CD1 (Compel Data)
14 (0xE)				L	F	0	0	EC2 (Establish Connection)
13 (0xD)				L	F	P	P	DT1 (Data) – contains SA
				0	F	0	0	RR (Round Trip Delay Reply)
12 (0xC)				0	F	0	0	RQ (Round Trip Delay Query)
11 (0xB)				L	F	P	P	CD2 (Compel Data)
9				L	F	P	P	DT2 (Data) – no SA
7				L	F	0	0	DC1 (Disconnect)
6				L	F	0	0	DC2 (Disconnect)
5				L	F	P	P	DT3 (Data) – contains SA
				L	F	0	0	DT5 (Data) – no SA
3				0	F	P	P	PT (Pass Token)
				4				RT (Return Token)
2				6				PN (Probe Node)
				7				PR (Probe Response)
				0				RI (Request Interval)
1				0	F	1	1	SR (Status Response)
				0	F	0	0	CT (Compel Time)
				0	F	0	1	TD (Time Distribution)
				0	F	1	0	Idle
0				1				CL (Claim LAS)
				6				TL (Transfer LAS)

Obrázek 3.6: Přehled rámců FF sběrnice a jejich dekódování dle frame control

3.4.2 Zdrojová a cílová adresa (Destination and source address)

Adresy dělíme dle délky na dlouhý formát adresy o velikosti 4B a krátký formát o velikosti 2B. Jejich formát poznáme podle bitu 3 ve Frame control, jak již bylo zmiňováno. Dlouhý formát adresy má tvar (H.L.N.S), který obsahuje: link, uzel (node) a selector zařízení. Zatímco krátký formát zahrnuje jen uzel a selector. Oba formáty i jejich velikosti jsou ukázány v tabulce 3.5.

3.4.3 Parametry

Obsah parametru je proměnný dle funkce rámce. Rámec CD může mít parametry nedefinované, proto byl pro názornou ukázkou parametrů vybrán rámec Probe Node, který je vyslán

jakmile LAS aktualizuje aktivní seznam zařízení. Tento rámec má 8 B obsah parametrů, jak je vypsáno v tabulce 3.6.

Formát	Priorita
00	Tato kombinace není použita
01	Urgetní (high) priorita
10	Normální (medium) priorita
11	Nízká (time - available) priorita

Tabulka 3.4: Priorita rámce

Dlouhý formát	Link	Uzel (Node)	Selector
Velikost	2B	1B	1B
Krátký formát	-	Uzel (Node)	Selector
Velikost		1B	1B

Tabulka 3.5: Formát adres

Index Bajtu	Obsah parametrů
2 maska 0xF0	V(PhLS)
2 maska 0x0F	Verze 0VVV
3 maska 0xF0	V(PhGE)
3 maska 0x0F	V(PhPE)
4..5	V(ST)
6	V(MRD)
7	V(MID)

Tabulka 3.6: Parametry PN rámce

- $V(\text{PhS})$ - maximální mezikanálová deformace signálu (PhL maximum inter-channel signal skew),
- Verze 0VVV,
- $V(\text{PhGE})$ - určení délky mezery po přenosu signálu (PhL post transmission gap extension units),
- $V(\text{PhPE})$ - velikost zpoždění signálu (PhL preamble extension units),
- $V(\text{ST})$ - délka přenosu (Slot-Time),
- $V(\text{MRD})$ - maximální čas vyčkání na odpověď (Maximum - Response - Delay),
- $V(\text{MID})$ - minimální doba mezi jednotlivými rámci (Minimum Inter-DLPPDU Delay) .

3.4.4 Kontrolní součet

Kontrolní součet je realizován pomocí cyklického redundantního součtu (CRC - Cyclic Redundancy Check), který se využívá pro detekci chyb během přenosu. Kontrolní součet je přikládán k datům a je odeslán společně s daty. Jakmile data přijmeme, spočítáme kontrolní součet a porovnáme s přijatým součtem. Pro výpočet CRC součtu využíváme funkci exkluzivního logického součtu XOR (eXclusive OR) s pravděpodobnostní tabulkou 3.7. Pro kontrolní výpočet CRC, musíme znát Generační Polynom (GP), za pomoci kterého byl původní součet počítán.

XOR		
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Tabulka 3.7: Pravděpodobnostní tabulka XOR

Výpočet CRC vysvětluje obrázek 3.7, nejdříve vezmeme naše data 101101110 a generační polynom $GP = 1001$, data doplníme počtem nul dle velikosti našeho polynomu (v našem případě 4 nuly). S doplněnými daty a generačním polynomem provedeme operaci xor, odstraníme nepodstatné nuly a znovu provedeme operaci xor s generačním polynomem. Tento postup opakujeme, dokud se délka dat nerovná Generačnímu polynomu.

Při popisování struktury rámců bylo čerpáno ze standardu [2] poskytnutým firmou Siemens.

$$\begin{array}{r}
 1011011100000 \\
 \\
 1011011100000 \\
 1001 \\
 \hline
 0010011100000 \\
 \\
 1001 \\
 \hline
 00001100000 \\
 1001 \\
 \hline
 0101000 \\
 1001 \\
 \hline
 \underline{001100} = C
 \end{array}$$

Obrázek 3.7: Ukázka výpočtu CRC

Kapitola 4

Hardware FF Monitoru

4.1 Analýza

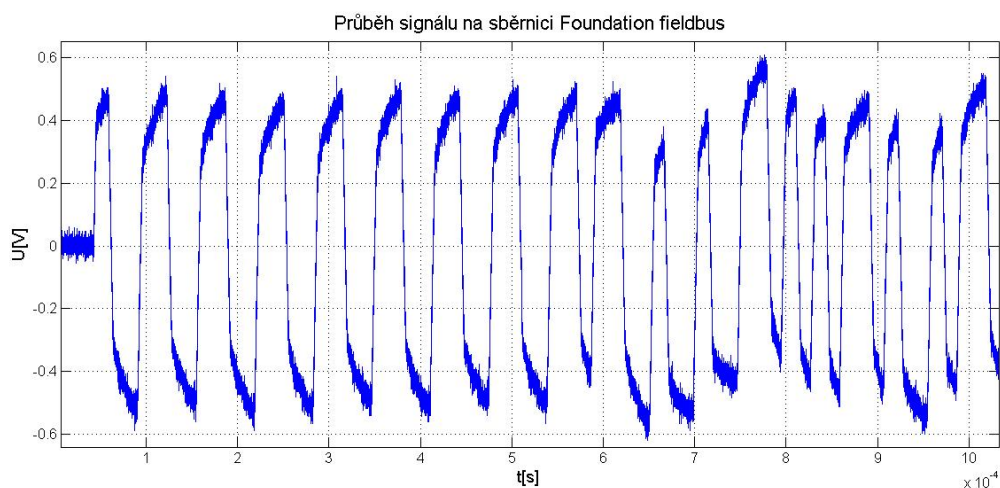
Při návrhu hardwaru bylo nutné si nejprve ujasnit nároky kladené na dané zařízení. Úkolem zařízení je zachytávat rámce na FF sběrnici, následně tyto rámce poslat do PC k následnému rozkódování a rekonstrukci signálu. Signál na sběrnici je přenášen pomocí dvou vodičů, které umožňují i napájení cílového zařízení. Signál se tedy skládá ze stejnosměrné složky 20 až 30V, na kterou je namodulovaný obdélníkový signál s maximální amplitudou 1,4 Vpp (peak to peak, špička-špička). Maximální amplituda signálu je definovaná v standardu [1].

Na základě těchto požadavků, byly vybrány periferie, které dokáží splnit tyto požadavky. Sběrnice USB, zajistí komunikaci s PC (komunikace po sériovém i paralelním portu byla vyloučena z důvodu absence těchto portů u nových PC). Analogově – Digitální převodník ADC (Analog to Digital Converter) s přesností minimálně 8bitů, který umožní měření napětí na sběrnici. Pro zpětnou rekonstrukci signálu, je důležité aby ADC převodník umožnil měřit vzorky ze sběrnice dostatečně rychle. „Nyquistův (Shannonův, Kotělnikovův) vzorkovací teorém říká, že dokonalá rekonstrukce signálu je možná pouze tehdy, když je vzorkovací frekvence větší než dvojnásobek maximální frekvence vzorkovaného signálu” [30]. Jak již bylo popsáno v teoretické části, přenosová rychlost signálu je 31,25 kbit/s. Data které přenáší daný obdélníkový signál jsou zakódována pomocí Manchester kódování, což znamená, že se střídají krátké $16\mu\text{s}$ a dlouhé $32\mu\text{s}$ pulzy. Perioda (frekvence) je vždy dodržena sousledností těchto pulzů. Daný A/D převodník pro zpětnou rekonstrukci signálu musí zachytit i $16\mu\text{s}$ pulzy. Podle vzorkovacího teorému musí být tedy vzorkovací frekvence A/D převodníku minimálně dvojnásobná – 62,5kHz. Tím zajistíme přenos první harmonické daného signálu, což vychází na 1 vzorek za $16\mu\text{s}$, tedy 1 vzorek na jeden pulz. Ale protože chceme rekonstruovat daný signál, byla snaha dosáhnout minimálně 4 vzorků na daný pulz, tedy frekvence 250kHz.

4.2 Návrh Hardware

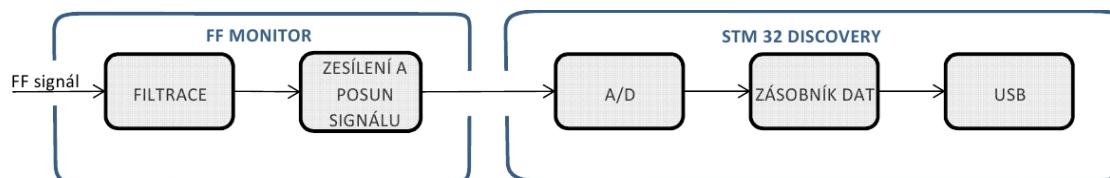
První prototyp zařízení byl postaven na vývojovém kitu STM32F4 DISCOVERY od firmy STMicroelectronics, jehož základem je mikrokontrolér STM32F407VGT6, jedná se o 32 bitový ARM (Advanced Risc Machine) s jádrem CORTEX M4 pracující na frekvenci

až 168 MHz. Dále vývojový kit umožňuje použít námi potřebnou USB sběrnici, A/D převodník s možností programově volitelné vzorkovací frekvence, bitového rozsahu a přímého přístupu do paměti (DMA - Direct Memory Access). Implementaci daných komponent ulehčí možnost programování, debugování (testování programu) mikrokontroléru pomocí USB a rozsáhlé množství knihoven. Velkou výhodou je možnost nastavení vzorkovací frekvence A/D převodníku a možnost DMA módu, kdy výsledné vzorky jsou ukládány do externí paměti RAM (Random-Access Memory). Tento režim nezatěžuje výkon vlastního výpočetního jádra a proto je možné procesor zároveň využít k posílání dat po USB sběrnici. Dalším důležitým parametrem byla frekvence mikrokontroléru. Daný A/D převodník umožňuje převádět analogové napětí v rozsahu 0V (GND) až po vstupní napájecí napětí mikrokontroléru 3,3V. Zatímco na sběrnici Foundation Fieldbus je signál v rozpětí -0,7 V do 0,7 V. Typický průběh signálu je znázorněn na obr. 4.1. Průběh signálu byl změřen na osciloskopu Tetronix, za pomoci diferenciální sondy. Data byla exportována do textového souboru a zobrazena pomocí programu Matlab.



Obrázek 4.1: Průběh signálu na sběrnici Foundation Fieldbus

Z tohoto důvodu byl navržen hardwarový modul FF Monitor, který zajišťuje posun signálu do rozsahu A/D převodníku, potlačení stejnosměrné složky signálu a odfiltrování nežádoucích frekvencí. Blokové schéma zpracování signálu je zobrazeno na obrázku 4.2.



Obrázek 4.2: Blokové schéma zpracování signálu

Hlavní blok hardwarového modulu tvoří zesilovač s posunem signálu. Pro tuto funkci byl vybrán přístrojový zesilovač AD623, který zajišťuje zesílení rozdílového signálu na jeho vstu-

pech. Zesilovač AD623 byl vybrán zejména pro svoje vlastnosti: rail-to-rail výstup (výstupní napětí může dosahovat téměř hodnot napájecího napětí), šířka pásma 800 kHz a snadné nastavení zesílení, připojením odporu mezi vývody RG (1 a 8). Vnitřní zapojení z katalogového listu [18] přístrojového zesilovače je zobrazeno na obrázku 4.3.

Aby nebylo dosaženo hraničních hodnot rozsahu A/D převodníku, bylo počítáno s rozsahem od 0,1 V do 3,2 V místo (0 do 3,3 V). Tedy počítaný celkový dynamický rozsah A/D převodníku je $Vd = 3,2 - 0,1 = 3,1$ V. Signál na sběrnici má maximální rozsah signálu 1,4 V_{pp} pro rezervu bylo počítáno s rozsahem $Vs = 2V$. Výpočet potřebného zesílení signálu.

$$G = \frac{Vd}{Vs} = \frac{3,1}{2} = 1,55 [-] \quad (4.1)$$

Pro dané zesílení určíme odpor R_G . Protože $R_1 = R_3, R_2 = R_4, R_6 = R_8$, můžeme výpočet zjednodušit:

$$G = \left(1 + \frac{2 \cdot R_6}{R_G}\right) \cdot \frac{R_2}{R_1} \quad (4.2)$$

dosadíme hodnoty odporů dle vnitřního zapojení (obrázek 4.3) a vypočteme odpor R_G .

$$R_G = \frac{2 \cdot R_6 \cdot R_2}{G \cdot R_1 - R_2} \quad (4.3)$$

$$R_G = \frac{2 \cdot 50 \cdot 10^3 \cdot 50 \cdot 10^3}{1,55 \cdot 50 \cdot 10^3 - 50 \cdot 10^3} \quad (4.4)$$

$$R_G = 181,818 k\Omega \rightarrow \underline{180 k\Omega} \quad (4.5)$$

Zpětně ověříme zesílení:

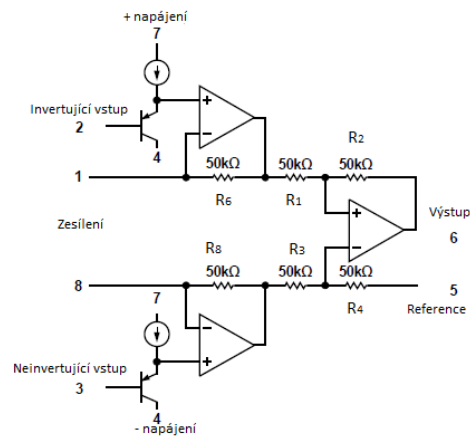
$$G = \left(1 + \frac{100 \cdot 10^3}{180 \cdot 10^3}\right) = 1,556 [-] \quad (4.6)$$

a rozsah A/D převodníku

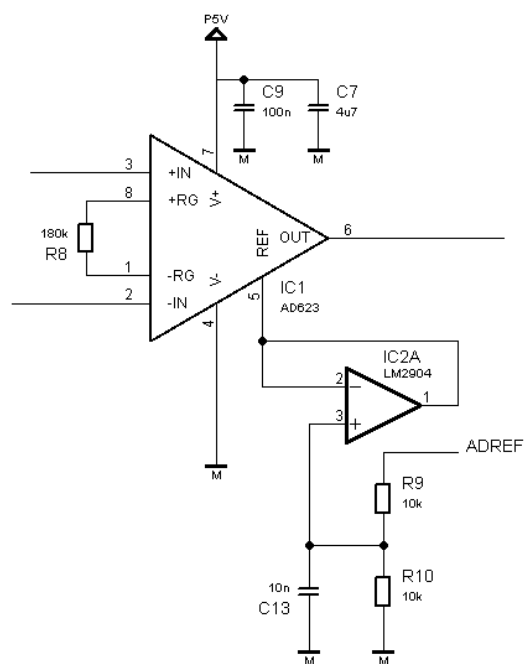
$$V_D = 1,556 \cdot 2 = 3,112 V \quad (4.7)$$

Zvětšení rozsahu o 0,012 V nezpůsobí dosah hraničních hodnot, protože na začátku byl dynamický rozsah A/D převodníku uměle snížen. Dané zesílení je tedy použitelné.

Posun signálu ze záporných hodnot je zajištěn změnou reference na polovinu referenčního napětí pro A/D převodník, tím posuneme signál na polovinu referenčního napětí pro A/D. Pin 5 určený pro připojení reference byl měl být připojen k nízké impedanci, protože připojením zdroje s vysokou impedancí by mohlo mít za následek snížení činitele potlačení souhlasné složky CMRR (Common Mode Rejection Ratio). Z tohoto důvodu bylo realizováno připojení reference za pomoci operačního zesilovače, jak je znázorněno na obrázku 5.1.



Obrázek 4.3: Vnitřní zapojení AD623, dle katalogového listu [18]

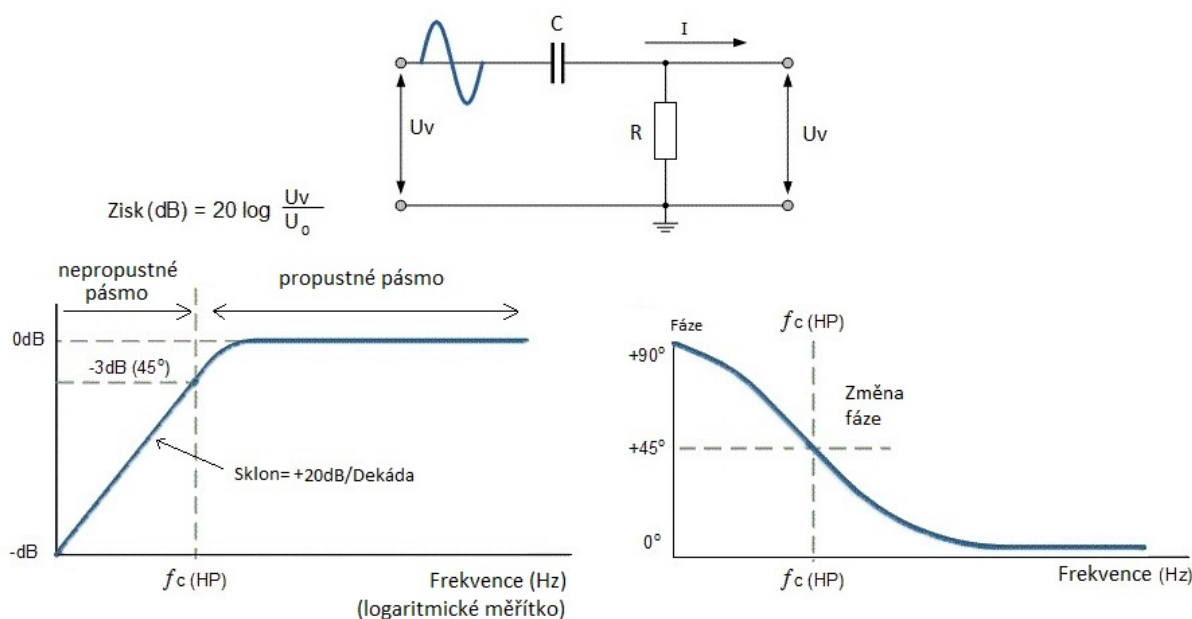


Obrázek 4.4: Schéma zapojení přístrojového zesilovače AD623

4.2.1 Filtrace

Pro určení potřebné filtrace, bylo na osciloskopu zobrazeno frekvenční spektrum signálu. Na frekvenčním spektru byla vidět výraznější amplituda spektra na frekvenci 31,25 kHz a 15,6 kHz to odpovídá $16\mu\text{s}$ a $32\mu\text{s}$ impulzům. Ale námi potřebný signál obsahuje harmonické složky i na vyšších frekvencích, proto byl stanoven rozsah propustného pásma filtru od 100 Hz do 400 kHz. Tím neovlivníme námi přenášený signál, ale zároveň potlačíme nežádoucí frekvence, například 50 Hz ze síťového napětí.

Pro potlačení stejnosměrné složky signálu a nízkých frekvencí do 100 Hz byl zvolen RC filtr typu horní propust (HP) prvního řádu (derivační článek). Zapojení HP, obecná frekvenční a fázová charakteristika jsou zobrazeny na obrázku 4.5.



Obrázek 4.5: Zapojení, frekvenční a fázová charakteristika filtru typu HP

Pro mezní frekvenci $f_C = 100 \text{ Hz}$ určíme hodnoty R a C dle následujících rovnic.

$$f_C = \left(\frac{1}{2 \cdot \pi \cdot R \cdot C} \right) \quad (4.8)$$

$$R \cdot C = \left(\frac{1}{2 \cdot \pi \cdot f_C} \right) = 1,59 \cdot 10^{-3} \quad (4.9)$$

Zvolíme hodnotu kondenzátoru dle hodnot, které máme k dispozici a rozumného poměru velikostí odporu a kondenzátoru. Pro tento případ byl zvolen kondenzátor $C = 220 \text{ nF}$. Následně dopočítáme velikost odporu:

$$R = \left(\frac{1}{2 \cdot \pi \cdot f_C \cdot C} \right) \quad (4.10)$$

$$R = \frac{1,59 \cdot 10^{-3}}{220 \cdot 10^{-9}} = 7234,31 \Omega \rightarrow \underline{6,8 \text{ k}\Omega} \quad (4.11)$$

Zpětně ověříme mezní frekvenci:

$$f_C = \left(\frac{1}{2 \cdot \pi \cdot R \cdot C} \right) = 106,39 \text{ Hz} \quad (4.12)$$

Při návrhu filtru typu Hp, bylo čerpáno z literatury [12].

Pro potlačení vysokých frekvencí nad 400 kHz byl použit filtr typu dolní propust. Bylo použito symetrické zapojení dle literatury [17] a [19], které je zobrazeno na obrázku 4.6. Tento filtr zabezpečuje potlačení vysokých frekvencí na obou vstupech přístrojového zesilovače. Měl by být navržen tak, aby propouštěl pásmo minimálně o 10% menší než je šířka pásma přístrojového zesilovače. Přístrojový zesilovač AD623 má šířku pásma 800kHz, zatímco mezní frekvence při návrhu filtru je stanovena na 400 kHz, tím je daný předpoklad splněn. Při návrhu symetrického zapojení musí být kladen důraz na stejné hodnoty kondenzátorů C4 a C6. Pokud možno měli by být vybrány se stejným dielektrikem (například X7R) a ve stejné procentní toleranci výroby. Tím zajistíme opravdu symetrické potlačení vysokých frekvencí. Parametry filtru navrhne dle následujících rovnic.

$$BW_{DIFF} = \left(\frac{1}{2 \cdot \pi \cdot R \cdot (2 \cdot C_D + C_C)} \right) \quad (4.13)$$

$$R \cdot (2 \cdot C_D + C_C) = \left(\frac{1}{2 \cdot \pi \cdot BW_{DIFF}} \right) \quad (4.14)$$

kde $R = R_1 + R_2$ a kondenzátor C_C je sériovou kombinací kondenzátorů C_{CA} a C_{CB} tedy:

$$C_C = \left(\frac{C_{CA} \cdot C_{CB}}{C_{CA} + C_{CB}} \right) \quad (4.15)$$

Při návrhu tohoto filtru je doporučení dle literatury [19], že kondenzátor C_D by měl být 10 krát větší než kondenzátory C_{CA} a C_{CB} . Byla zvolena hodnota $C_{CA} = 100 \text{ pF}$, $C_{CB} = 100 \text{ pF}$ tedy kondenzátor $C_D = 1 \text{ nF}$.

Dopočítáme velikost odporu R:

$$R = \left(\frac{1}{2 \cdot \pi \cdot BW_{DIFF} \cdot (2 \cdot C_D + C_C)} \right) \quad (4.16)$$

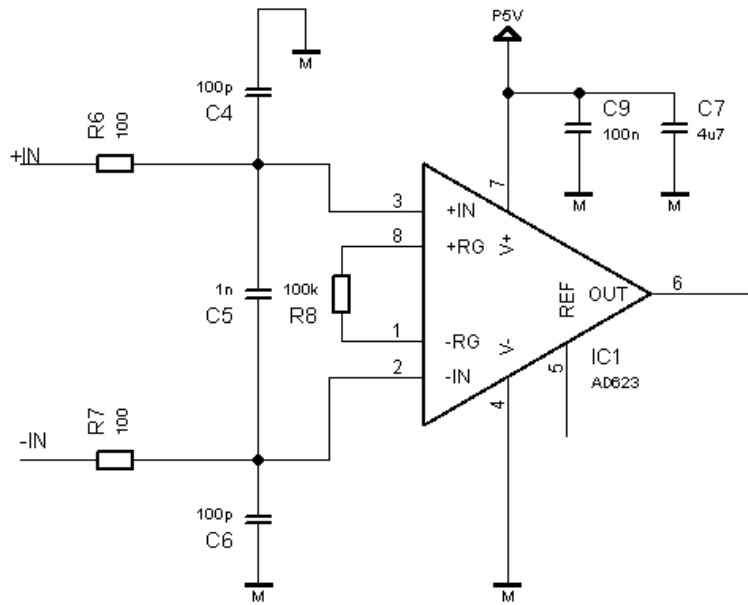
$$R = \left(\frac{1}{2 \cdot \pi \cdot 400 \cdot 10^3 \cdot (2 \cdot 10^{-9} + 5 \cdot 10^{-11})} \right) \quad (4.17)$$

$$R = 194 \Omega \rightarrow \underline{200 \Omega} \quad (4.18)$$

Celkový odpor $R = 200 \Omega$ tedy odprory $R_1 = 100 \Omega$ a $R_2 = 100 \Omega$

Na závěr provedeme kontrolu výpočtu šířky pásma:

$$BW_{DIFF} = \left(\frac{1}{2 \cdot \pi \cdot R \cdot (2 \cdot C_D + C_C)} \right) = 388,182 \text{ kHz} \quad (4.19)$$



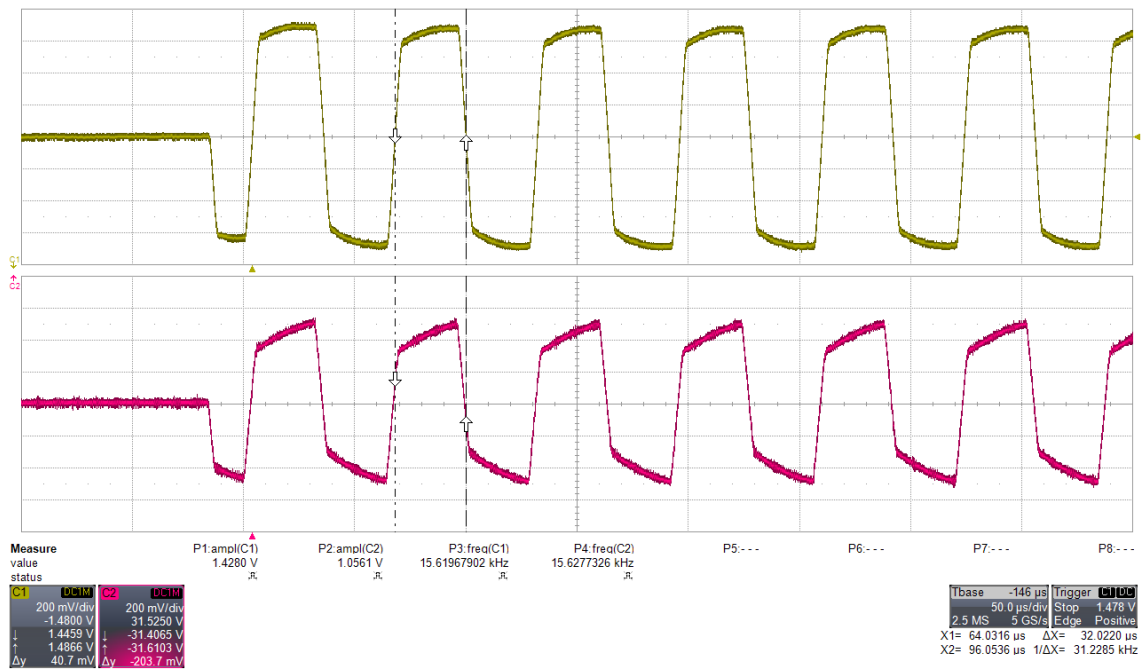
Obrázek 4.6: Zapojení symetrického filtru typu DP

4.3 Realizace

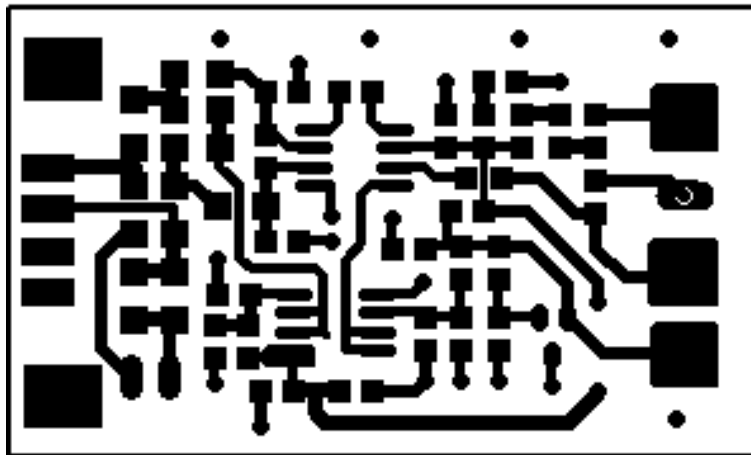
Funkčnost navrženého zařízení byla nejdříve ověřena na provizorním prototypu jehož fotografie je přiložena v příloze 9.1. Na tomto prototypu byl změřen výstupní signál (žlutě), který je zobrazen na obrázku 7.1 vůči původnímu signálu (červeně). Na obrázku je vidět, že upravený signál má nepatrné zpoždění vůči původnímu signálu. Toto zpoždění, ale nemá žádný vliv na deformaci signálu. Signál splňuje námi požadovaný rozsah, neovlivňuje Manchester kódování a tím zachovává potřebné informace pro budoucí dekodování.

4.3.1 Návrh plošného spoje

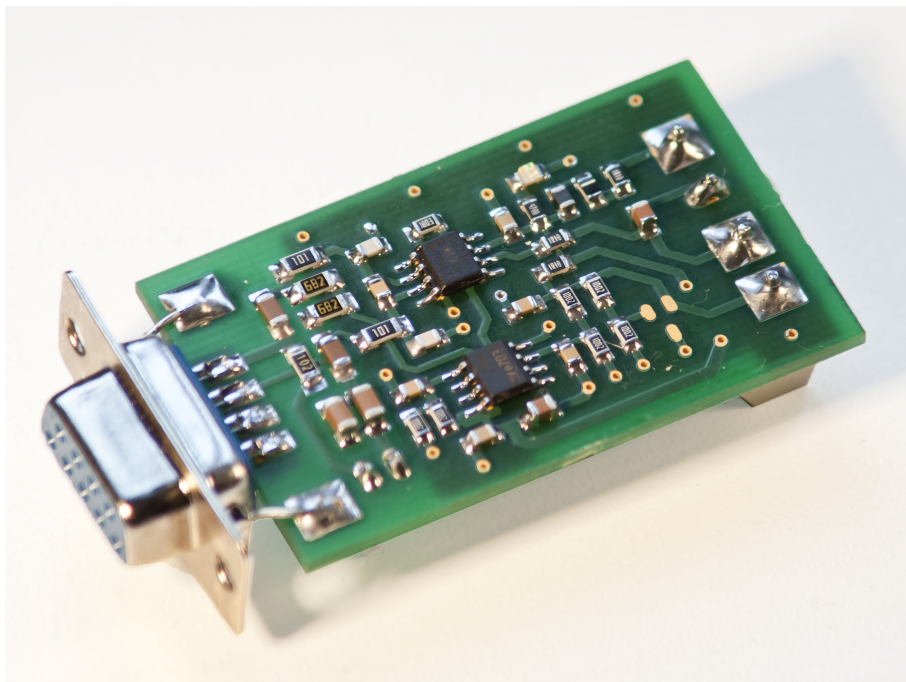
Při návrhu plošného spoje byl kladen důraz na rozměry desky, aby šířka desky byla stejná jako konektoru vystupující z desky. Tím bylo dosaženo snadné manipulace a možnosti zasunutí desky přímo do sběrnice rozvaděče. Dále bylo důležité rozmístění konektorů včetně zkratovací propojky, tak aby byly dobře dostupné i při připojení ke sběrnici. Následovalo rozmístění součástek s ohledem na funkci a na pravidla návrhu plošného spoje. Filtrační kondenzátory byly umístěny vždy co nejblíže k dané součástce, aby nedocházelo k nežádoucímu rušení signálu. Z důvodu dostatečného uzemnění a minimalizování počtu propojek byla vytvořena oboustranná deska s rozměry 3x5cm s rozlitou zemí na spodní straně desky. Na obrázku 9.3 je vidět vrchní strana desky plošných spojů. Kompletně osazená deska je na fotografii 4.9. Další dokumentace k návrhu desky je v příloze 9.1. Celý návrh plošného spoje byl realizován pomocí programu Eagle a deska byla vyrobena firmou Pragoboard s.r.o. Gerber data pro výrobu desky jsou součástí přiloženého CD.



Obrázek 4.7: Porovnání původního a upraveného signálu



Obrázek 4.8: Vrchní strana desky plošných spojů, zobrazena v měřítku 2:1



Obrázek 4.9: Vyrobená a osazená hardwarová sonda FF Monitoru

Kapitola 5

Firmware Monitoru

Úkolem firmwaru na miktokontroléru STM32F407VGT6 je získávat napětové úrovně z průmyslové sběrnice a posílat je k dalšímu zpracování do PC. Pro získání napětových úrovní byl využit A/D (analogově-digitální) převodník s přímým přístupem do paměti (DMA - Direct Memory Access) a jejich odeslání je umožněno díky sběrnici USB.

5.1 Obsluha A/D převodníku

Obsluha A/D převodníku je realizována pomocí knihovny `adc.h` od firmy STMicroelectronics volně dostupné ze stránek [9], která dodává i referenční manuál [27], kde jsou k dispozici všechny možnosti nastavení a funkce A/D převodníku. Nejdříve jsou inicializovány všechny potřebné periferie pro A/D převodník v DMA módu. Pro obsluhu inicializace byla napsána metoda `ADC3_CH12_DMA_Config`, která zapne jednotlivé hodiny pro A/D převodník a DMA mód. Dále nakonfiguruje DMA mód pro A/D převodník dle referenčního manuálu, inicializuje převodník ADC3 jako 8bitový a nastaví vzorkovací rychlost A/D převodníku. Ukázka nastavení vzorkovací rychlosti AD převodníku:

```
//nastavení hodin ABP2 pro AD převodník (ABP2 = 84 Mhz)
RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC3, ENABLE);
//zvolíme děličku 2 pro ADC
ADC_CommonInitStructure.ADC_Prescaler = ADC_Prescaler_Div2;
// zvolíme dobu odečtu jednoho vzorku na 84 cyklů
ADC_RegularChannelConfig(ADC3, ADC_Channel_12, 1, ADC_SampleTime_84Cycles);
```

Systémové hodiny mikrokontroléru mají rychlost 168 Mhz, hodiny pro obsluhu periférií využívají děličku 2 tedy běží na 84MHz. A/D převodník má také nastavenou děličku 2, a proto jsou hodiny A/D převodníku (ADCclock) nastaveny na 42 Mhz. Daný A/D převodník je nastavený jako 8bitový, takže jeho minimální doba (md) odečtu jednoho vzorku trvá 8 cyklů. Programově volitelná rychlost (pd) odečtu byla zvolena na 84 cyklů. Vzorkovací rychlost A/D převodníku můžeme spočítat dle následujících rovnic.

Nejdříve vypočteme celkovou dobu (cd) převodu jednoho vzorku:

$$cd = md + pd = 8 + 84 = 92 \text{ cyklu} \quad (5.1)$$

a následně vypočítáme vzorkovací rychlost (vzr) A/D převodníku:

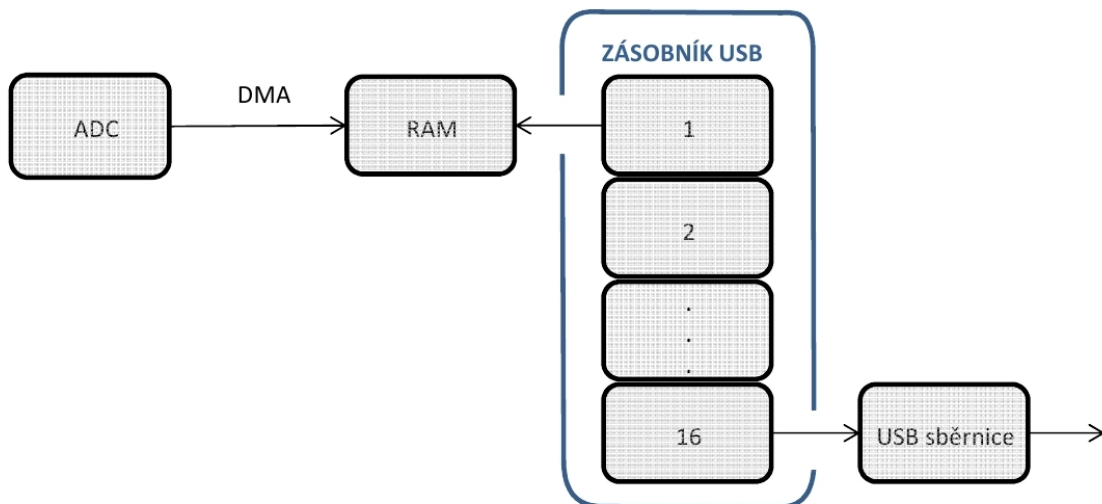
$$vzr = \frac{ADCclock}{cd} = \frac{42 \cdot 10^6}{92} = 456,522 \text{ kHz} \quad (5.2)$$

Což vychází na 1 vzorek za 2,19 μ s. Tedy skoro 8 vzorků na 16 μ s pulz.

Jakmile A/D převodník převede analogovou hodnotu na digitální, hodnota je pomocí DMA módu automaticky zapsána do RAM paměti. Pokud je naplněn jeden ze dvou zásobníků alokovaných pro hodnoty A/D převodníku je vyvoláno přerušení ve kterém je obsah zásobníku zkopírován do zásobníků pro USB komunikaci. Díky alokaci dvou zásobníků je eliminováno zpoždění měření hodnot A/D převodníkem.

5.2 Obsluha USB sběrnice

Pro obsluhu USB sběrnice byl využit soubor knihoven a příkladně vytvoření virtuálního com portu USB VCP od firmy STMicroelectronics volně dostupné ze stránek [9]. USB je nastaveno do režimu Full Speed, kde jeho maximální přenosová rychlost dosahuje až 12 Mbits/s. Při této rychlosti však často docházelo k výpadkům, stabilní přenos dat bylo dosaženo při rychlosti 4 Mbits/s. K této rychlosti přenosu byla přizpůsobena i vzorkovací frekvence A/D převodníku na 456,522KHz, která je popsána v předchozí kapitole. Ve vzorovém příkladu byly přepsány funkce *VCP_DataRx* a *VCP_DataTx*, aby posílání dat probíhalo bez zbytečných mezer v komunikaci (jakmile je naplněn jeden ze zásobníků, data jsou okamžitě posílána po sběrnici). Pro vytvoření dostatečné paměťové rezervy bylo vytvořeno 16 zásobníků, které fungují jako velký kruhový zásobník. Jakmile dojde k naplnění zásobníku A/D okamžitě se zkopíruje do volného USB zásobníku a ten se začne posílat po sběrnici. Celý proces je znázorněn na obrázku 5.1.



Obrázek 5.1: Ukázka získávání dat od A/D převodníku

5.3 Vývojové prostředí IAR Embedded Workbench

Pro vývoj firmwaru bylo použito vývojové prostředí (IDE) IAR Embedded Workbench, které obsahuje potřebné nástroje pro vestavěné (embedded) aplikace. Umožňuje přehledné programování pomocí editoru zdrojového kódu, nástroji pro řízení a správu objektů, C/C++ kompilátorem s možností generovat optimalizovaný kód a plně integrovaný C-SPY debugger (nástrojem pro odladění chyb v programu). Dále obsahuje obrovské množství knihoven usnadňujících vývoj daných aplikací. V neposlední řadě současný kompilátor umožňuje kontrolu zdrojového kódu s pravidly MISRA C (MISRA C: 2004 a 1998), jedná se standard vyvinutý asociací MISRA (Motor Industry Software Reliability Association) pro vývoj softwaru v programovacím jazyku C. Cílem standardu je napomáhání bezpečnosti kódu, přenositelnosti a hlavně spolehlivosti. Více informací o tomto vývojovém prostředí je dostupných [8].

Kapitola 6

Software FF Monitoru

6.1 Analýza

Při návrhu softwaru byly diskutovány dva programovací jazyky C/C++ a Java. Oba jsou velice rozšířené a pro tvorbu grafického uživatelského rozhraní vhodné. Nejdříve bylo nutné stanovit si požadavky budoucího softwaru:

- přehledné grafické uživatelské prostředí,
- možnost vykreslování grafů,
- příjem dat z virtuálního komunikačního portu,
- rozkódování rámců,
- výpis a uchování daných rámců s velkým nárůstem dat v čase.

Po zhodnocení těchto požadavků s hlavním ohledem na mé zkušenosti s programovacími jazyky. Byl zvolen programovací jazyk Java. Java umožňuje multiplatformní aplikace, snadné a přehledné vytvoření grafického rozhraní, obsahuje knihovny na vykreslování grafů a umožňuje připojit dynamickou knihovnu vytvořenou v jazyce C, kdyby byl problém s příjmem dat či k datově méně náročnému rozkódování rámců.

6.1.1 Programovací jazyk Java

Java je objektově orientovaný jazyk, vytvořený firmou Sun Microsystems, která ho představila v roce 1995. Programovací jazyk Java byl od začátku vyvíjen hlavně pro „vestavěné systémy“, ale v roce 1993 si firma Sun Microsystems uvědomila důležitost rozšiřujících se webových stránek a možnosti Javy i pro webové aplikace. Tímto se Java stala jedním z nejpopulárnějších programovacích jazyků. V roce 2007 byly od firmy Sun Microsystems uvolněny zdrojové kódy a nadále se Java vyvíjí jako otevřený (open source) software.

Původní záměr Javy pro „vestavěné systémy“ vyžadoval poskytování služeb na všech platformách, to vedlo ke vzniku interpretovatelného jazyka na úrovni mezikódu. Překlad neprobíhá do jazyka relativních adres (.OBJ), ale do pseudojazyka bajtkódu (byte-code), který

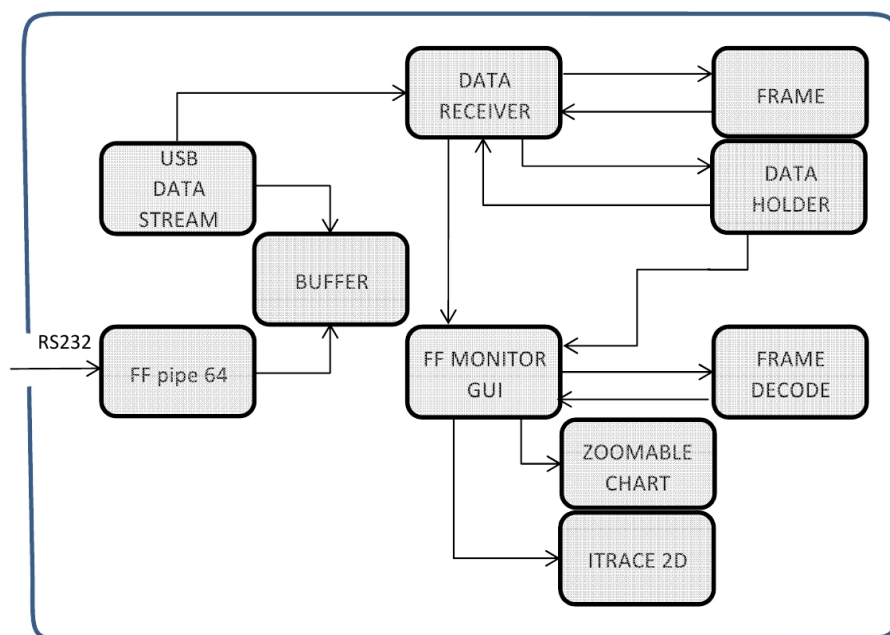
není závislý na cílovém počítači. Interpretace je však výkonnově náročná, proto se používají různé metody optimalizace. Nejnovější způsob optimalizace je tzv. Adaptivní optimalizace, kde je kód zpočátku interpretován a podle četnosti spouštění jednotlivých částí kódu jsou následně některé části zkompileovány. [24]

6.1.2 Knihovna JChart2D

Pro reprezentaci grafů byla vybrána knihovna JChart2D, která podporuje čárové a bodové grafy. Knihovna umožňuje ukládání dat do různých tříd založených na seznamu či kruhovém zásobníku, tisknutí grafů, ukládání ve formě obrázků a nastavení parametrů grafů (popisky os, barvu, mřížku apod.). V neposlední řadě podporuje i přiblížení jednotlivých částí grafu včetně přizpůsobení daného měřítka. Kompletní dokumentace této knihovny je k dispozici v literatuře [6].

6.2 Návrh software

Na obrázku 6.1 je znázorněn celkový návrh softwaru. Blokový diagram zobrazuje jednotlivé třídy a jejich sdílení dat. Nejdříve budou postupně popsány jednotlivé třídy a jejich metody. Následně bude vysvětlena celková funkce programu.



Obrázek 6.1: Blokové schéma návrhu softwaru

6.2.1 UsbDataStream

Bohužel se v průběhu tvoření softwaru ukázalo, že program FF Monitor nedokáže komunikovat za pomoci knihovny RXTX s virtuálním sériovým portem dostatečně rychle.

Dokázal načíst ze sériové linky pouze 100 000 vzorků za 1.5 s, což bylo nedostačující, protože firmware umožňuje vyčítat ze sériové linky 456 521 vzorků za sekundu. Proto byla implementována dynamická knihovna vytvořená v Jazyce C pomocí JNI (Java Native Interface). JNI je rozhraní, které umožňuje propojit programy napsané v jiných programovacích jazycích (zkompilované pro daný hardware) s virtuálními stroji Javy. Třída `UsbDataStream` obsahuje deklarace nativních metod. Tyto metody umožňují volání nativní dynamické knihovny v jazyce C. Implementace JNI je podrobně popsána v podkapitole Implementace software.

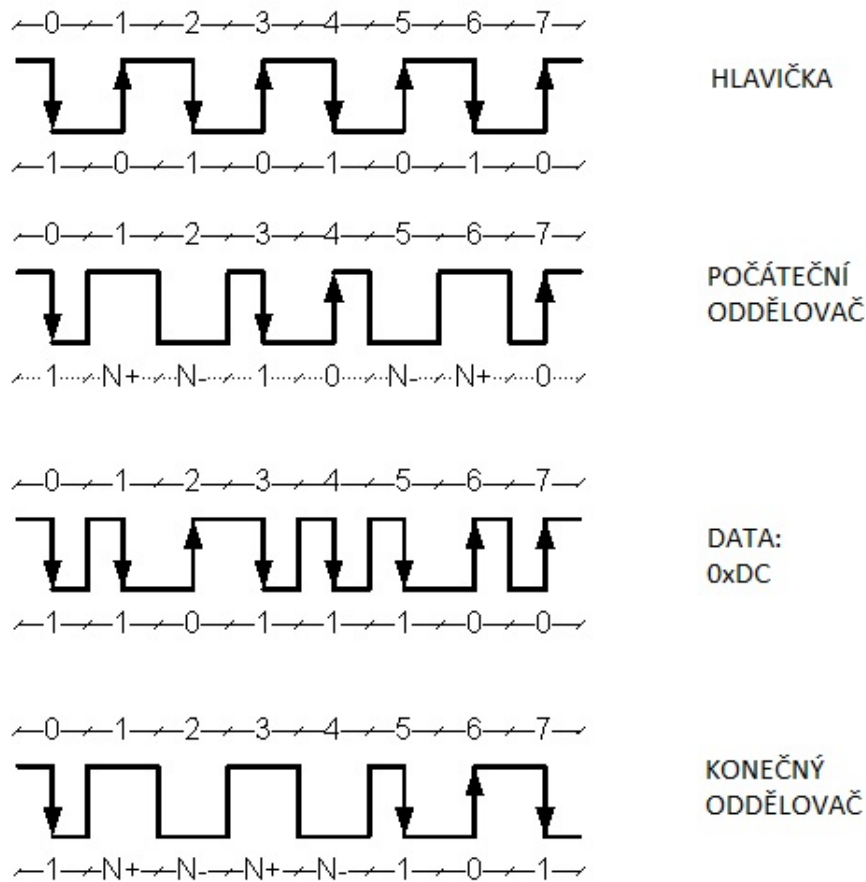
Vytvořená třída `UsbDataStream` neobsahuje přímo obsluhu sériové linky a dekodování rámců na sběrnici, tyto procesy zajišťuje obslužný program `ffpipe64`, který má nedefinovanou stejnou datovou strukturu (na obrázku 6.1 znázorněnou jako `BUFFER`) jako program `UsbDataStream`. Sdílení této datové struktury zajišťuje tzv. Meziprocesorová komunikace (IPC - Inter-Process Communication). Jedná se o mechanismus umožňující komunikaci a sdílení dat mezi procesy. Daná struktura se namapuje přímo na paměť v adresovém prostoru. Proces pak může za pomoci ukazatelů číst a modifikovat obsah paměti. Jedná se o jednodušší a hlavně rychlejší způsob sdílení dat. Nejdříve vytvoříme namapování objektu pomocí funkce `CreateFileMapping`, kde určíme práva přístupu k souboru, minimální případně maximální velikost mapovaných bitů, název souboru apod. Dále vytvoříme obraz objektu pro zápis, který přiřadíme na adresu výše zmiňované datové struktury. Tímto způsobem máme zajištěný přístup přes datovou strukturu přímo do paměti. Struktura musí být namapovaná ve všech procesech, kterými do ní chceme přistupovat. Struktura obsahuje příznak nového rámce, zakódovaný rámec a jeho body pro rekonstrukci signálu.

6.2.2 Obslužný program `FFpipe64`

Obslužný program `FFpipe64` má za úkol přijímat data po sériové lince a následně je zpracovávat. Prvotní částí programu je otevření sériového portu a jeho obsluha. Jakmile přijme vzorek (napětíové úrovně od ADC) detekuje, zda se jedná o logickou hodnotu „1“, „0“ či střední hodnotu. Střední hodnota určuje období ticha (nevysílání rámců) na sběrnici. Tato informace je předána funkci `decodAppend`, která počítá počet přijatých vzorků a s ohledem na dobu převodu vzorků od ADC určí, zda se jedná o krátký $16\mu\text{s}$ nebo dlouhý $32\mu\text{s}$ pulz. Při detekování pulzu zavolá obslužnou rutinu v podobě stavového automatu na dekodování rámců. Jakmile se detekuje po rámci období ticha, data jsou zapsána do sdílené struktury a pomocí dynamické knihovny `UsbData Stream` předána do programu `FF Monitor`, kde jsou následně rozkódována. Data se předávají ve formátu řetězce (`String`), kde jsou jednotlivé bajty zapsány v hexadecimální soustavě a oddělena mezerou (například: `26 08 DC 16 ...`). Společně se zakódovanými daty se odesílají ve formátu bajt (`byte`) i původní napětíové úrovně dat, pro rekonstrukci datového signálu. Program `FFpipe64` je samostaně spustitelný program, který vypisuje do příkazového řádku počet přijatých rámců a jednotlivé rámce v zakódované podobě.

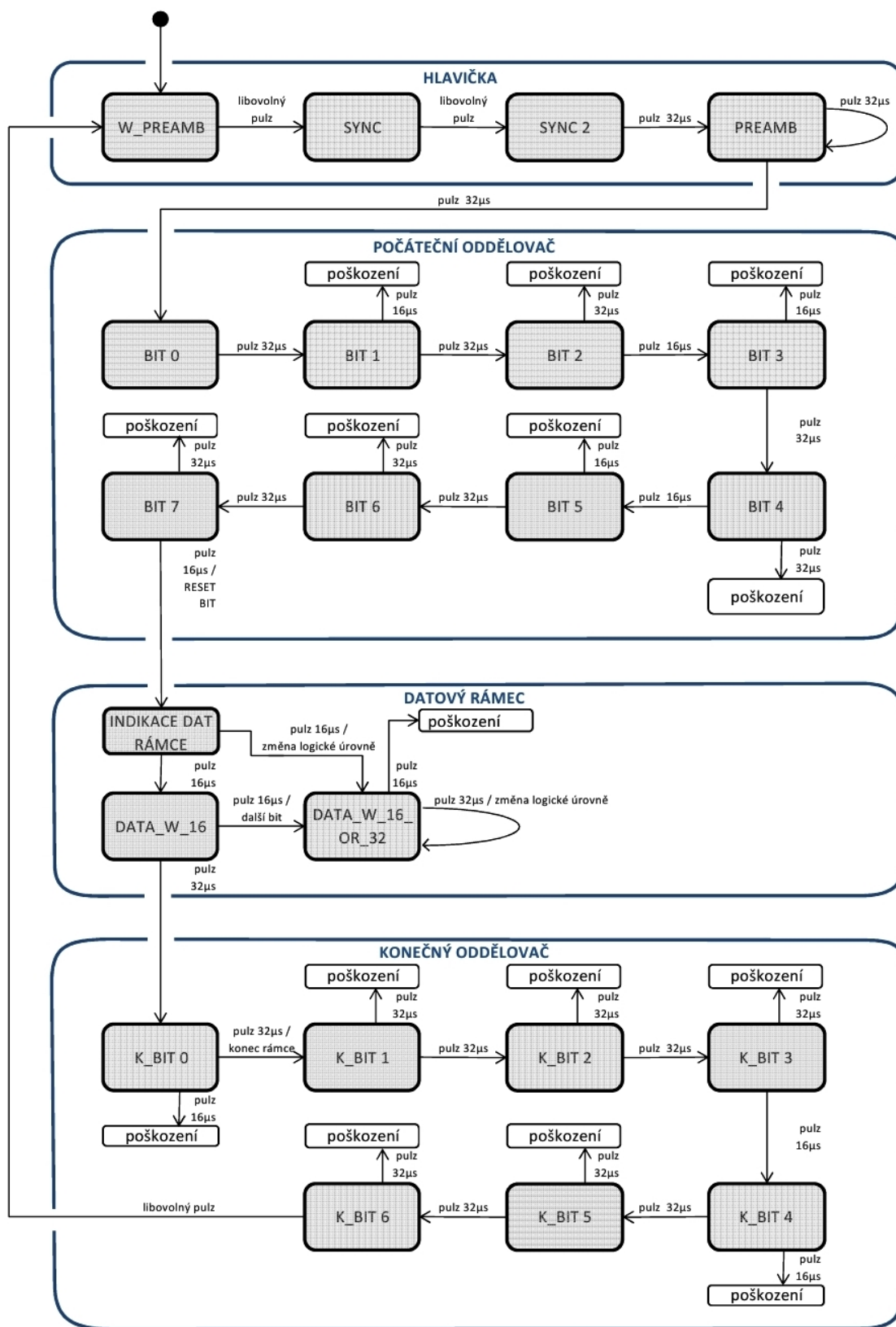
6.2.2.1 Dekódování rámců na sběrnici

Jak již bylo řečeno každý rámec začíná synchronizační hlavičkou a počátečním oddělovačem (`Start Frame`) a je zakončen konečným oddělovačem (`End Frame`). Tyto sekvence mají danou strukturu, která je znázorněna na obrázku 6.2.



Obrázek 6.2: Struktura jednotlivých sekvencí na sběrnici

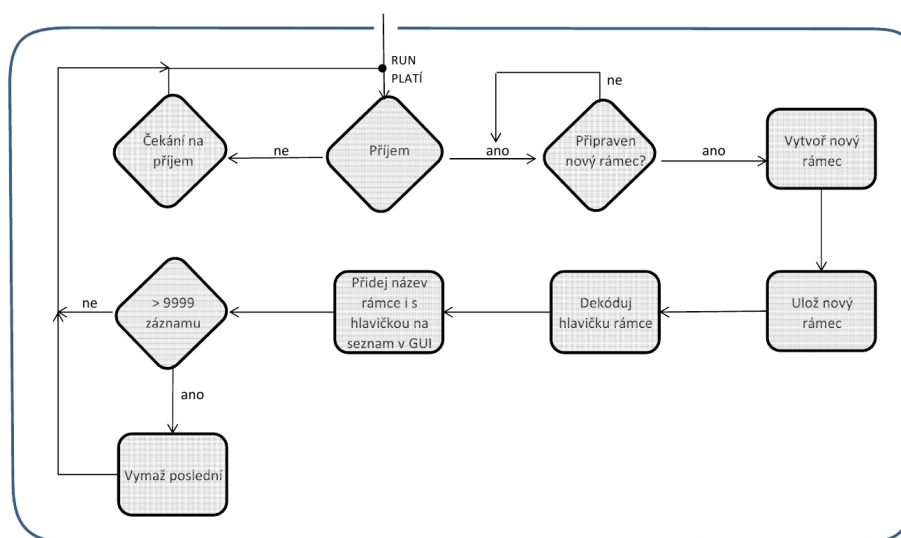
Když se podíváme na tuto strukturu je vidět, že ji tvoří krátké $16\mu\text{s}$ a dlouhé $32\mu\text{s}$ impulzy. Podle posloupnosti těchto impulzů, můžeme dekodovat o jakou sekvenci se jedná a zároveň je možné určit změnu logické úrovně v datech. Ta se totiž mění vždy po dlouhém pulzu. Pokud jsou za sebou jen krátké pulzy zůstává stejná. Nesmíme zapomenout, že na fyzické vrstvě se používá Manchester kódování, kde je logická změna úrovně detekována náběžnou či sestupnou hranou. Na tomto principu byl napsán jednoduchý stavový automat, jehož diagram je znázorněn na obrázku 6.3. Automat si pamatuje vždy svůj stav do příchodu dalšího pulzu, podle toho jaký pulz přijde posune se do dalšího stavu. Tím detekuje jednotlivé sekvence a zjistí kde začínají data rámce. Jakmile detekuje data, stanovuje logickou úroveň na základě předchozího pulzu. Pokud přijde krátký, má stejnou logickou úroveň jako předchozí, ale když přijde dlouhý, logická úroveň se změní. Data jsou rozdělena na jednotlivé bajty a pak předána k následnému rozkódování.



Obrázek 6.3: Stavový automat na dekódování rámců

6.2.3 DataRecevier

Třída Data Recevier obsahuje instance tříd `UsbDataStream` `data`, `DataHolder` `dh` a komponentu `list` z `awt` knihovny, která umožňuje výpis aktuálních rámců v grafickém rozhraní. Dále obsahuje vlákno, které zajišťuje vyčítání dat z dynamické knihovny. V metodě `run` se kontroluje zapnutí příjmu dat a zda je připraven nový rámeček, pokud ano je vytvořena nová instance třídy `Frame`. Pomocí této instance je vytvořen nový rámeček podle přijatého řetězce. Tento rámeček je společně s navzorkovaným signálem uložen do seznamu rámců v `DataHolder`. Dále je dekodována hlavička rámečku a ta je s názvem rámečku přidána na seznam v listu do grafického prostředí. Následně se zkontroluje počet záznamů, pokud přesahuje nastavenou konstantu (9999) vymaže se první a je čekáno na další rámeček. Celý proces je znázorněn v blokovém diagramu na obrázku 6.4.



Obrázek 6.4: Blokový diagram receiveru

6.2.4 Třídy `Frame` a `DataHolder`

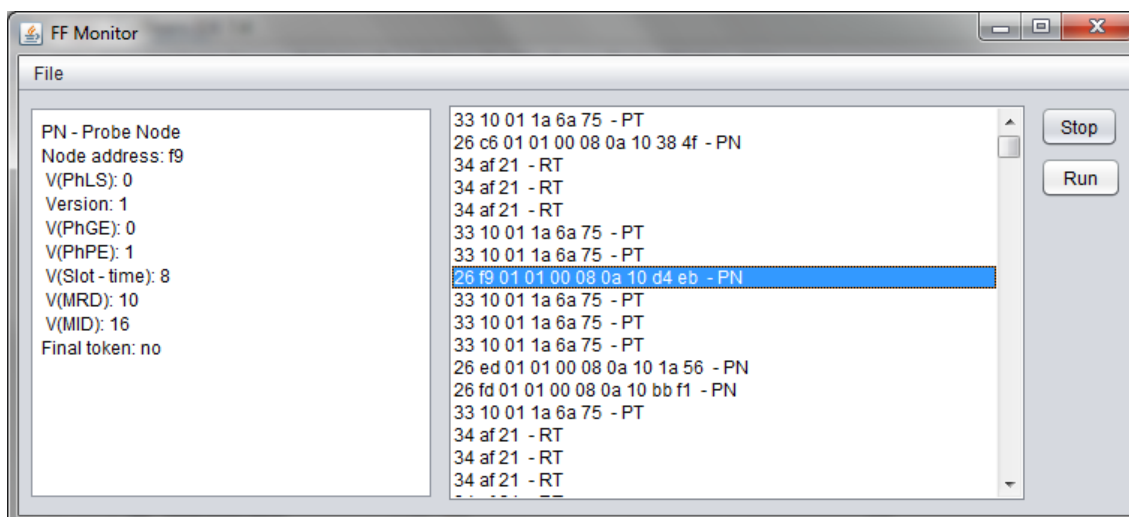
Třída `Frame` obsahuje metody na nastavení a získání názvu rámečku (`setFrame`, `getFrame`), dále metody na přidání bodů daného rámečku (`setPoint`, `getPoints`), metodu pro formát výpisu rámců do souboru (`toString`) a v poslední řadě metodu, která umožní dekodovat pouze hlavičku daného rámečku. Seznam instancí této třídy udržuje třída `DataHolder` v podobě `LinkedList`, který umožňuje vrátit daný frame dle indexu, vymazat první či poslední záznam s automatickým posunem indexů. Dále obsahuje metody pro přidání rámečku (`addFrame`), odmazání prvního rámečku (`removeFirst`), vymazání celého seznamu (`removeAll`), získání konkrétního rámečku (`getFrame`) a nastavení či získání celého seznamu (`setFrameList`, `getFrameList`).

6.2.5 Grafické uživatelské rozhraní FF Monitoru

Při návrhu grafického uživatelského prostředí bylo postupováno nejdříve z pohledu uživatele. Uživatel potřebuje vidět aktuální seznam rámců detekovaných na sběrnici a zároveň mít možnost jednotlivě dekódovat daný rámec (zobrazit jeho prioritu, adresu, data apod.). Proto bylo vytvořeno grafické uživatelské rozhraní fungující ve dvou režimech: režim RUN, kdy je zapnut příjem rámců ze sběrnice a režim STOP, při kterém je umožněno jejich dekódování.

Režim RUN zobrazuje v listu seznam rámců přijatých ze sběrnice. Na každém řádku je vypsán rámec v jeho bajtové zakódované podobě s dekódovanou hlavičkou například 34 af 21 – RT. Obsluha listu je podrobně popsána v podkapitole DataReceiver.

Režim STOP ukončí příjem rámců. Uživatel si může rozkliknout jakýkoliv rámec ze seznamu a zobrazit si jeho rozkódování v textovém poli. Po zobrazení rozkódování rámce je automaticky zobrazen v novém okně i jeho grafický průběh k podrobnější analýze. Dekódování rámců je zajištěno zavoláním statické metody `getDecodedFrame` ze třídy `FrameDecode`. V tomto režimu je zároveň umožněno uložit aktuální seznam rámců do souboru včetně jejich bodů. Hlavní okno grafického rozhraní je zobrazeno na obrázku 6.5.



Obrázek 6.5: Hlavní okno grafického uživatelského rozhraní

Při tvorbě grafického uživatelského rozhraní bylo čerpáno z literatury [23].

6.2.6 Třída `FrameDecode`

Třída `FrameDecode` zajišťuje kompletní rozkódování rámců, nejdříve dekóduje hlavičku rámce, včetně priority rámce, formátu adresy a finálního tokenu. Pokud obsahuje adresu odesilatele či příjemce zavolá se metoda pro dekódování adresy `getDestAdress`. Po rozkódování hlavičky jsou postupně volány metody dle struktury rámce. Například pro rámec `Probe Node`, jsou zavolány metody `getNodeAdress` a `getProbeNode`. Tyto metody vrací rozkódovaná data v podobě řetězce. Jednotlivé řetězce se sestaví pomocí metody `append` ze třídy `StringBuilder` do formátovaného řetězce, který je vypsán v textovém poli grafického uživatelského

prostředí. Použití metody `append` má tu výhodu, že java nevytváří pokaždé nový řetězec. Výsledek rozkódování je znázorněn na obrázku 6.5.

6.2.7 Zobrazení průběhu rámce

Vykreslení grafického průběhu rámce probíhá přímo v grafickém uživatelském prostředí. Jakmile je zobrazen dekodovaný rámeček, vytvoří se nové okno s průběhem rámce. K vykreslení daného průběhu byla využita knihovna `Jchart2D` popsána v kapitole 6.1.2. Nejdříve byla vytvořena instance třídy `ZoomableChart`, která zajistí možnost přiblížení či oddálení vybrané oblasti grafu. Dále byl vytvořen objekt trace pro ukládání množiny dat ze třídy `Trace2DSimple`. Tato třída představuje základní třídu pro ukládání objektů, umožňuje nastavit barvu zobrazovaných dat, název dat a nastavení jednotek pro obě osy, které budou zobrazovány za názvem dané množiny bodů. Body do objektu trace byly přidány pomocí metody `addPoint(x,y)`, do které jako parametr zadáme souřadnice jednotlivých bodů. Takto je vytvořen základní graf bez popisků. Pro nastavení popisků os slouží metoda `setAxisTitle(AxisTitle title)`, aby nemusela být vytvořena nová instance třídy `AxisTitle`, byla využita metoda `getAxisTitle()`, díky níž byla získána aktuální instance, kterou může být nastavena. Vytvoření grafu včetně jeho nastavení je ukázáno v příloze 11.1. Při psaní kódu pro zobrazení průběhu rámce bylo čerpáno z literatury [6] a [22].

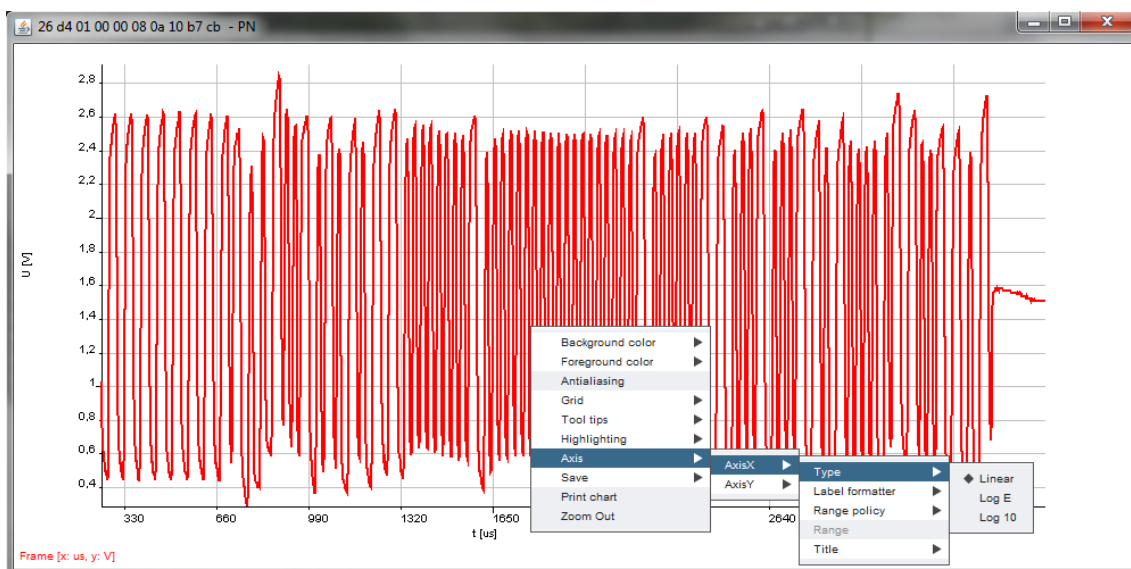
Další parametry grafu mohou být nastaveny přímo uživatelem v grafickém rozhraní, jakmile klikne levým tlačítkem myši na libovolnou část grafu zobrazí se menu s nastavením. Uživatel má k dispozici kompletní nastavení os včetně změny měřítka, nastavení pozadí grafu, zapnutí mřížky jednotlivých os, možnost uložení souboru ve formátu `.eps` nebo ve formátu obrázku (`.bmp`). Dále uživatel může využít zobrazení štítku se souřadnicemi vybraného bodu či přímý tisk grafu z uživatelského prostředí. Na obrázku 6.6 je ukázáno grafické okno s průběhem signálu včetně zobrazeného menu. Jak již bylo zmíněno výše, průběh signálu lze přehledně analyzovat, díky možnosti přiblížení libovolné části signálu. Kliknutím pravého tlačítka myši a jejím tažením uživatel vybere část, která má být zvětšena. Tento proces lze několikrát opakovat, dokud nebude uživatel spokojen. Samozřejmě lze průběh kdykoliv vrátit k původnímu zobrazení pomocí menu a funkcí `ZoomOut`.

Pro snadné propojení celého zařízení byl na CD přiložen uživatelský manuál, který přehledně popisuje propojení hardwarové sondy s DISCOVERY kitem a rozchození celé komunikace k detekci rámců. Dále byla přiložena i instalace virtuální platformy, aby měl uživatel všechny potřebné prostředky na jednom CD.

6.3 Implementace

6.3.1 Vývojové prostředí Netbeans IDE

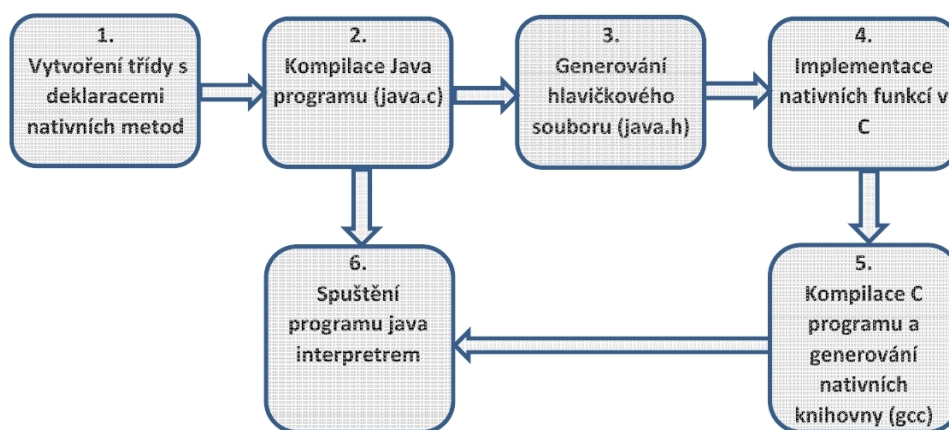
Pro vývoj softwaru bylo použito vývojové prostředí Netbeans IDE, jedná se o prostředí napsané v jazyce Java. Prostředí je primárně určeno na vývoj aplikací v programovacím jazyce Java, ale podporuje i další (například C++, PHP, Ruby). Celé vývojové prostředí je k dispozici zdarma, je platformově nezávislé a usnadňuje tvorbu grafického rozhraní možností využití grafického editoru [29].



Obrázek 6.6: Zobrazení grafu s možnostmi nastavení

6.3.2 JNI (Java Native Interface)

Implementace nativní knihovny a propojení s Javou je znázorněno na obrázku 6.7. Příklad ilustruje vytvoření C knihovny, a její volání z Java aplikace. Takto byla vytvořena třída s názvem `UsbDataStream` (.java), která obsahuje deklarace nativních metod v Jave. Tato třída byla zkompileována v Jazyce Java a byl vytvořen bajt-kód (.class). Tyto soubory byly zabaleny do balíčku `usb1`. Následovala aplikace nástroje `javah` pro generování hlavičkového souboru. Ve vytvořeném hlavičkovém souboru je vygenerovaná C funkce, která je propojena s Java nativní metodou. Následovala implementace deklarovaných funkcí v jazyce C. Kompilace C programu a vytvoření nativní dynamické knihovny (ve Windows .dll). [7]



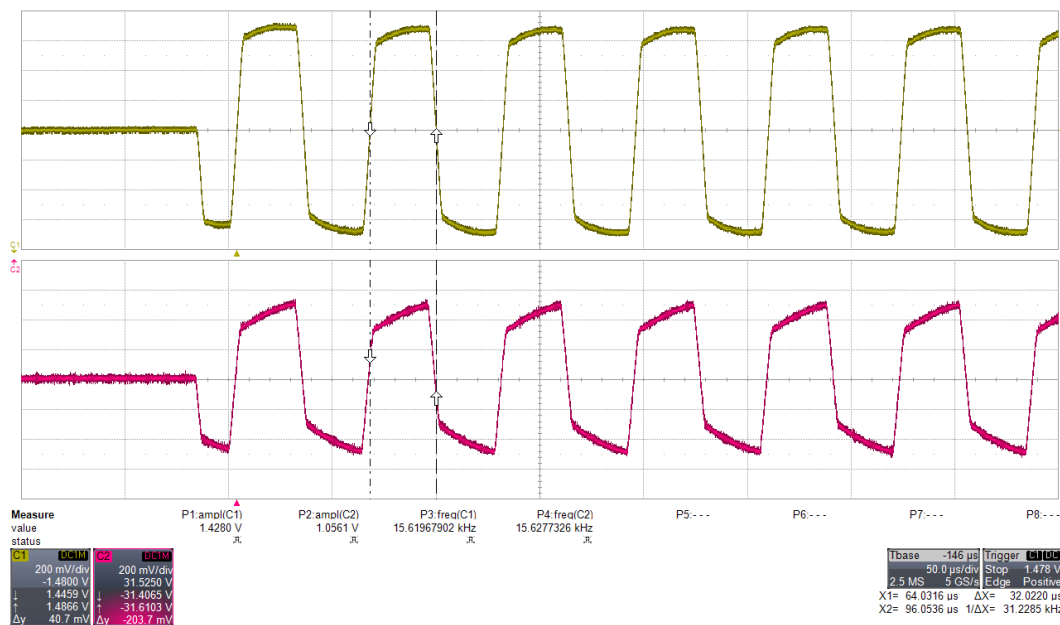
Obrázek 6.7: Ukázka implementace JNI, převzato [7]

Kapitola 7

Testování a výsledky

7.1 Hardwarová sonda

Nejdříve byla otestována hardwarová sonda monitoru, zda upravuje signál na sběrnici jak bylo v návrhu požadováno. Na obrázku 7.1 je porovnán upravený signál s původním signálem ze sběrnice. Kde upravený signál byl při měření posunut o 1,5 V, pro snadnější porovnání s původním signálem. Na zobrazeném průběhu je vidět, že námi upravený signál splňuje dané požadavky. Amplituda signálu je 1,42 V_{PP} a celkové posunutí signálu je 1,5V. Signál tedy splňuje dynamický rozsah A/D převodníku od 0,1V do 3,2V. Potlačuje stejnosměrnou složku a nežádoucí frekvence původního signálu. Neovlivňuje Manchester kódování a zachovává potřebné informace pro budoucí dekódování.



Obrázek 7.1: Porovnání původního a upraveného signálu

7.2 Monitor Foundation Fieldbus

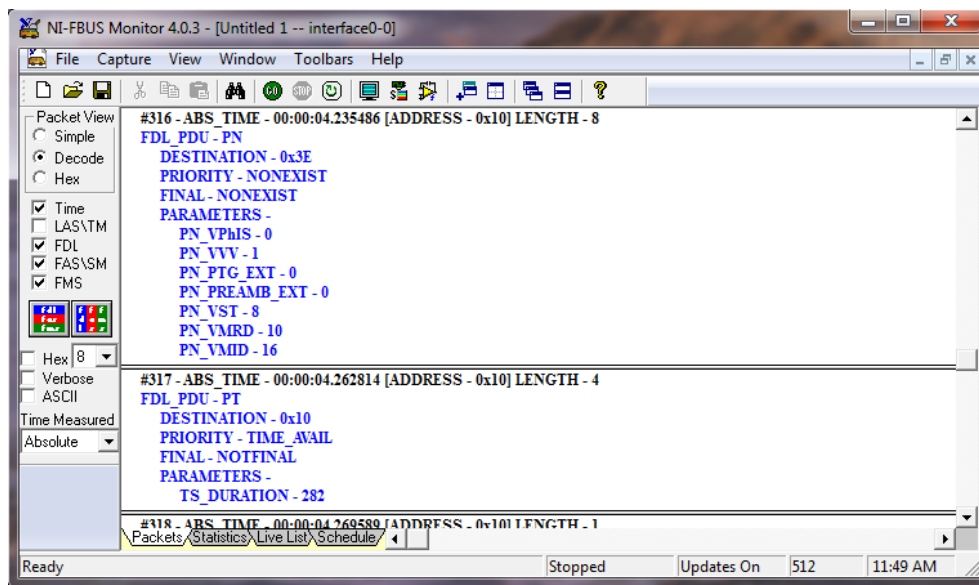
Celý systém detekce rámců byl otestován za pomoci zařízení NI-USB-8486 od firmy National Instruments, které za pomoci softwaru NI-FBUS Monitor umožňuje zobrazení, dekodování, filtrování dle parametrů a statistiku přijatých rámců na sběrnici. Bohužel není zde k dispozici zobrazení jednotlivých průběhů rámce. Zařízení NI-USB-8486 bylo využíváno v rámci firmy Siemens v oddělení CTDC, kde byl poskytnut i software NI-FBUS Monitor. Zařízení NI-USB-8486 a vyrobené zařízení FF Monitor byly připojeny na stejnou sběrnici pomocí provizorního rozvaděče. Tímto způsobem bylo otestováno detekování rámců na sběrnici a jejich správné dekodování.

7.2.1 Detekování rámců

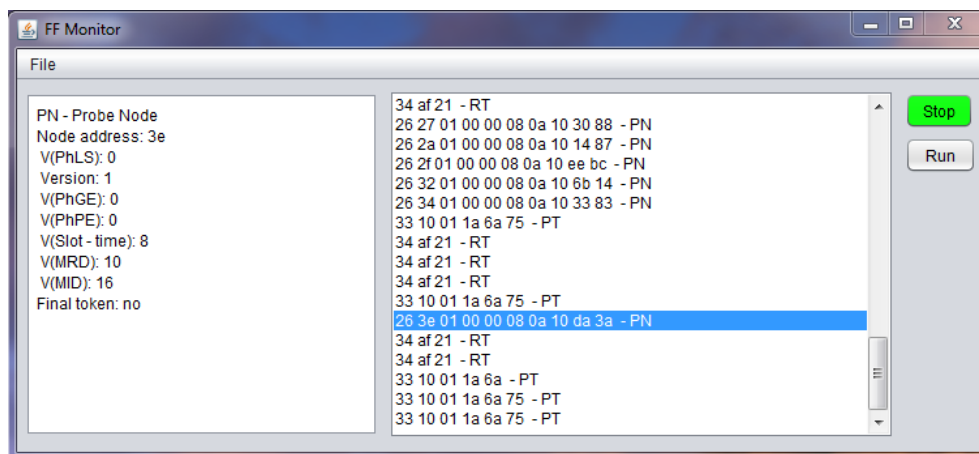
Detekování rámců bylo ověřeno vypisováním počtu přijatých rámců. Program fpipe64, který vypisuje rámce v zakódované podobě do příkazového řádku dosahoval stejné rychlosti příjmu rámců jako program NI-FBUS Monitor. Bohužel výpis v grafickém uživatelském prostředí byl 7x pomalejší, ale na rozdíl od programu NI-FBUS Monitor zachovává i průběh rámce a umožňuje jeho zobrazení. Když byla obsluha grafického prostředí vypnuta a zároveň vypnuto uchovávání dat rámce, vypisování probíhalo jen v příkazovém řádku, bylo dosaženo stejné rychlosti detekování rámců jako u programu FFpipe64. Tato volba vypnutí výpisu v grafickém prostředí a přerušení načítání dat rámce, bude začleněna do grafického prostředí.

7.2.2 Dekodování rámců

Na obrázcích 7.2 a 7.3 je znázorněno dekodování stejného rámce PN, pomocí systému NI-FBUS Monitor a FF Monitoru Foundation Fieldbus. Na obou systémech je rámec dekodován shodně, takto byly ověřeny i rámce RT a PT. Další rámce v dané sestavě nebyly k dispozici. Záznam rámců detekovaných na sběrnici je k dispozici na přiloženém CD.



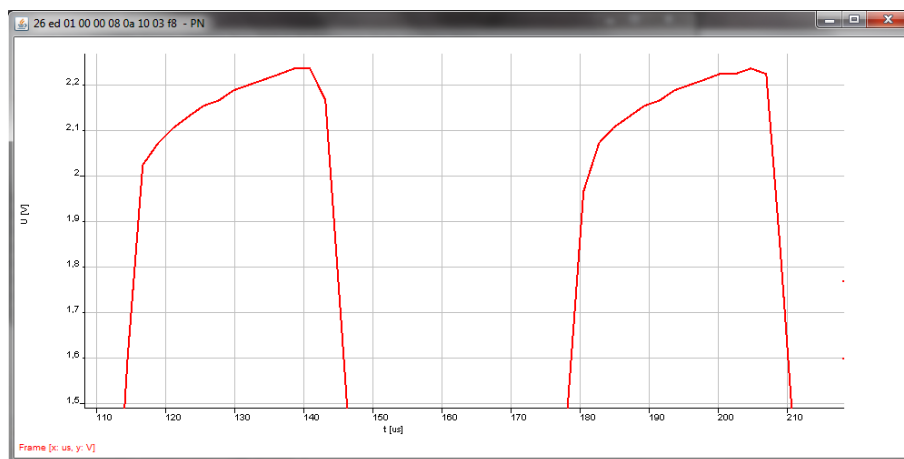
Obrázek 7.2: Dekódování rámce PN programem NI-FBUS Monitor



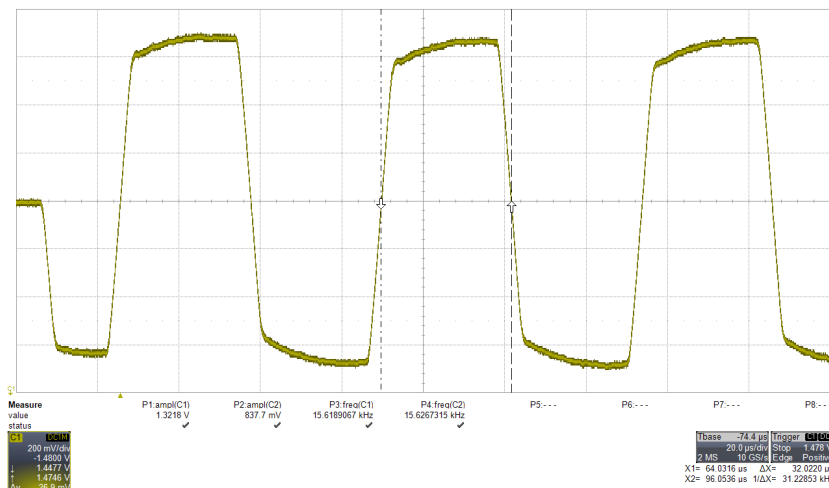
Obrázek 7.3: Dekódování rámce PN programem FF Monitor

7.2.3 Analýza průběhu rámce

Průběh rámce zobrazovaný v grafickém prostředí byl porovnán s průběhem signálu z hardwarové sondy zobrazeného osciloskopem. Na obou signálech byla změřena délka jednotlivých impulzů 32μ a amplituda signálu, která v obou případech byla $1,3V_{pp}$. Dále bylo ověřeno pomocí funkce Hihlighting, která slouží v prostředí grafu ke zvýraznění jednotlivých vzorků, že doopravdy dlouhý pulz $32\mu s$ obsahuje 16 bodů (vzorků), tím byla ověřena vypočtená vzorkovací frekvence a rychlost přenosu USB komunikace. Výsledky z měření jsou zobrazeny na obrázcích 7.4 a 7.5. U obrázku z programu FF Monitor, je pro přehlednost signál přiblížen jen na dva pulzy, aby byla vidět šířka pulzů.



Obrázek 7.4: Jednotlivé pulzy rámce v programu FF Monitor



Obrázek 7.5: Jednotlivé pulzy upraveného signálu

Kapitola 8

Závěr

Cílem této diplomové práce bylo navrhnout a realizovat monitor sběrnice Foundation Fieldbus, jehož úkolem bylo detekovat a dekodovat rámce na sběrnici, včetně možnosti zobrazení jejich grafického průběhu. Pro kompletní realizaci bylo nutné navrhnout hardwarovou sondu pro detekci rámců, napsat obslužný firmware a software, který umožní zobrazování rámců v grafickém uživatelském prostředí.

Před samotným návrhem bylo potřeba se seznámit se specifikací průmyslové sběrnice Foundation Fieldbus. Zejména se strukturou signálu na sběrnici, způsobem komunikace a kódování dat jednotlivých rámců. Tato problematika je popsána v kapitole 3. Následně byla navržena hardwarová sonda pro detekci rámců, součástí návrhu byl návrh filtrů na potlačení nežádoucích složek signálu a upravení signálu s ohledem na dynamický rozsah A/D převodníku, který slouží k měření napěťových úrovní na sběrnici. Celý návrh a samotná realizace sondy je popsána v kapitole 4.

Kapitola 5 pojednává o návrhu obslužného firmwaru pro A/D převodník a USB sběrnici, která zajišťuje přeposílání naměřených dat k dalšímu zpracování do PC. Součástí kapitoly je také nastavení a výpočet vzorkovací frekvence A/D převodníku. Hlavní částí celého systému byl návrh a vytvoření obslužného softwaru. Byl vytvořen program pro detekování rámců z jednotlivých napěťových úrovní signálu a přehledné grafické prostředí, které obstarává výpis a dekodování rámců. Součástí grafického prostředí je možnost zobrazení průběhu rámce, včetně možnosti využití různých nástrojů k podrobnější analýze celého signálu. O návrhu obslužného softwaru a grafickém uživatelském prostředí pojednává kapitola 6.

Nedílnou součástí této práce bylo testování systému na průmyslové sběrnici a porovnání výsledků s konkurenčním softwarem určeným k detekci rámců. Zhodnocení výsledků je ukázáno v kapitole 7. Z výsledků je vidět, že daný systém je schopen detekovat a dekodovat rámce na sběrnici Foundation Fieldbus, včetně zobrazení průběhu rámce se zachováním frekvence a napěťových úrovní signálu. Obsahuje základní nástroje k podrobnější analýze daného průběhu, například přiblížení či oddálení průběhu bez deformace měřítka a možnosti zobrazení souřadnic jednotlivých bodů. Součástí systému je možnost uložení a načtení záznamu detekovaných rámců, včetně jejich průběhů. Z předchozího shrnutí a výsledků lze říci, že všechny body zadání byly splněny.

Systém je využitelný i v oblasti zdravotnictví při vývoji různých čidel a zařízení komunikujících po dané sběrnici. Daný systém bude v budoucnosti využíván firmou Siemens k testování nových zařízení komunikujících prostřednictvím sběrnice Foundation Fieldbus, proto

se předpokládá jeho rozšíření o vzniklé požadavky včetně vytvoření dokumentace a návodu v anglickém jazyce. V budoucnosti je i možnost rozšíření na jiné průmyslové sběrnice.

Literatura

- [1] *Digital data communication for measurement and control – Fieldbus for use in industrial control systems – Part 2 : Physical Layer specification*. 2003-01-24.
65C/289/FDIS, Poskytnutý firmou Siemens.
- [2] *Digital data communication for measurement and control – Fieldbus for use in industrial control systems – Part 4 : Data Link Layer protocol specification*. 2003-03-28.
65C/291/FDIS, Poskytnutý firmou Siemens.
- [3] *Unikátní řešení ve farmaceutickém průmyslu*. 2004 [online; navštíveno 10.2.2014].
Dostupné z: [/http://www.siemens.cz/siemjet/public/downloads/ForumAD_2004_02.pdf](http://www.siemens.cz/siemjet/public/downloads/ForumAD_2004_02.pdf).
- [4] *Liquiline M – skok vpřed v analýze kapalin*. 2005, [online; navštíveno 4.3.2014].
Dostupné z: [/http://www.odbornecasopisy.cz/index.php?id_document=30662](http://www.odbornecasopisy.cz/index.php?id_document=30662).
- [5] *Foundation Fieldbus dominuje na poli průmyslových sběrnic*. 2007 [navštíveno 15.3.2014].
Dostupné z: [/http://www.odbornecasopisy.cz/=zindex.php?id_document=33942t](http://www.odbornecasopisy.cz/=zindex.php?id_document=33942t).
- [6] What is jchart2d, 2010 [online; navštíveno 8.2.2014].
Dostupné z: <http://jchart2d.sourceforge.net/>.
- [7] 12 - struktury, union, výčtové typy, volání c z javy, 2011 [online; navštíveno 18.2.2014].
Dostupné z: <https://edux.feld.cvut.cz/courses/A0B36PR2/tutorials/12/start>.
- [8] Iar embedded workbench, 2014 [online; navštíveno 10.4.2014].
Dostupné z: <http://www.iar.com/Products/IAR-Embedded-Workbench/>.
- [9] High-performance and dsp with fpu, arm cortex-m4 mcu with 1 mbyte flash, 168 mhz cpu, art accelerator, ethernet, [online; navštíveno 17.4.2014].
Dostupné z: <http://www.st.com/web/catalog/mmc/FM141/SC1169/SS1577/LN11/PF252140#>.
- [10] Safety in a double packr, [online; navštíveno 4.3.2014].
Dostupné z: http://www.cpp-net.com/home/-/article/32218883/34697966/Safety-in-a-double-pack/art_co_INSTANCE_0000/maximized/.
- [11] Level measurement, [online; navštíveno 7.3.2014].
Dostupné z: [/http://www.waterfordsystems.com/Waterford_Systems/Siemens_Level.html](http://www.waterfordsystems.com/Waterford_Systems/Siemens_Level.html).
- [12] High pass filters, [online; navštíveno 7.4.2014].
Dostupné z: http://www.electronics-tutorials.ws/filter/filter_3.html.

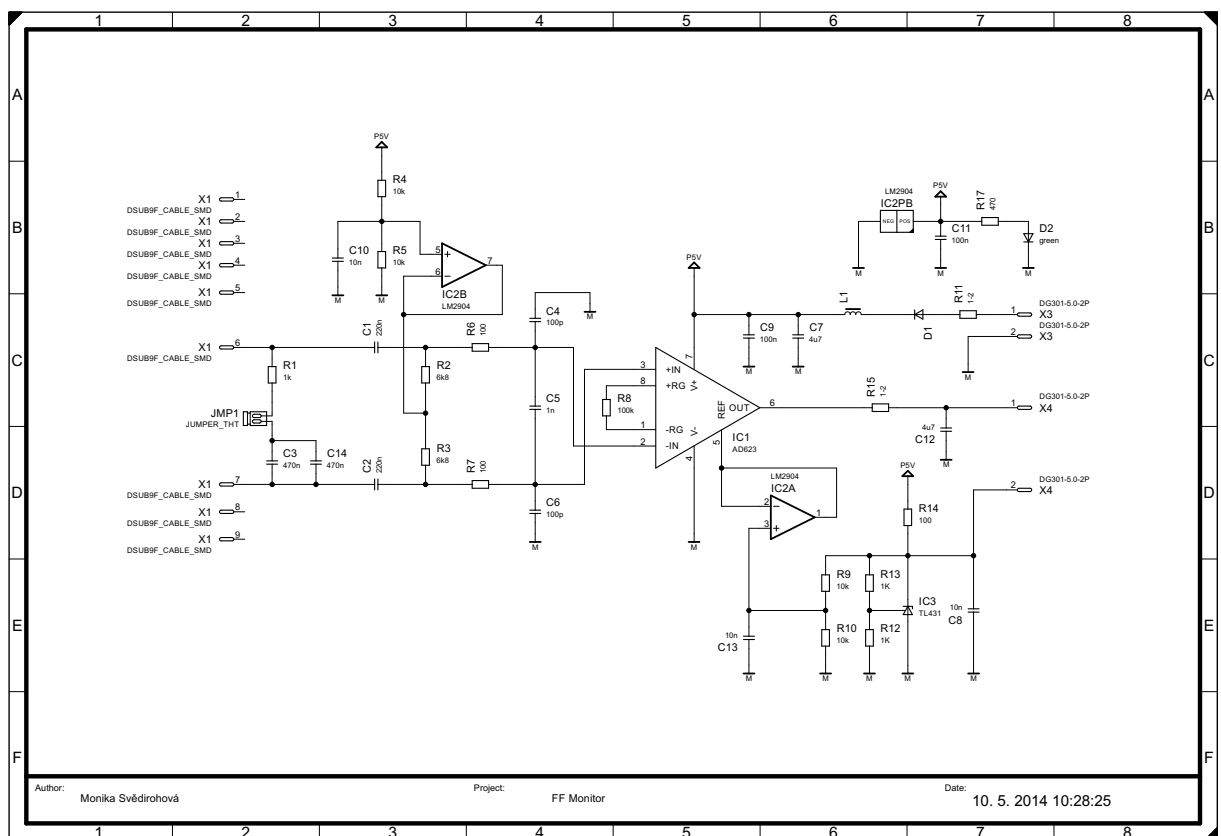
- [13] Pressure transmitter for food applications, [online; navštíveno 8.5.2014].
Dostupné z: [/http://www.directindustry.com/prod/vega-griehaber-kg/pressure-transmitters-food-applications-9136-787587.html](http://www.directindustry.com/prod/vega-griehaber-kg/pressure-transmitters-food-applications-9136-787587.html).
- [14] Snímače tlaku vegabar, [online; navštíveno 8.5.2014].
Dostupné z: [/http://www.odbornecasopisy.cz/index.php?id_document=33905](http://www.odbornecasopisy.cz/index.php?id_document=33905).
- [15] SAMSON AG. Foundation fieldbus, [online; navštíveno 21.3.2014].
Dostupné z: [/http://www.samson.de/pdf_en/1454en.pdf](http://www.samson.de/pdf_en/1454en.pdf).
- [16] Pavel Bohm. Diplomová práce: Komunikační modul pro průmyslovou sběrnici foundation fieldbus, 2012 [online; navštíveno 3.3.2014].
Dostupné z: [/https://otik.uk.zcu.cz/bitstream/handle/11025/3053/dp_bohm.pdf?sequence=1](https://otik.uk.zcu.cz/bitstream/handle/11025/3053/dp_bohm.pdf?sequence=1).
- [17] Lew Counts Charles Kitchin and Wilmington Moshe Gerstenhaber, Analog Devices. input filter prevents instrumentation-amp rf-rectification errors. [online; navštíveno 8.4.2014].
Dostupné z: <http://m.eet.com/media/1135724/111303di.pdf>.
- [18] Analog Devices. Instrumentation amplifier ad623, [online; navštíveno 22.2.2014].
Dostupné z: [/http://www.analog.com/static/imported-files/data_sheets/AD623.pdf](http://www.analog.com/static/imported-files/data_sheets/AD623.pdf).
- [19] Matt Duff. Five basic mistakes to avoid when using instrumentation amplifiers, 2007 [online; navštíveno 8.4.2014].
Dostupné z: http://www.planetanalog.com/document.asp?doc_id=527518.
- [20] Fieldbus Foundation. Foundation technology, [online; navštíveno 6.3.2014].
Dostupné z: [/http://www.fieldbus.org/index.php?option=com_content&task=view&id=45&Itemid=195](http://www.fieldbus.org/index.php?option=com_content&task=view&id=45&Itemid=195).
- [21] Dave Greenfield. Combined calibrator and fieldbus communicator, [online; navštíveno 8.5.2014].
Dostupné z: [/http://www.automationworld.com/combined-calibrator-and-fieldbus-communicator](http://www.automationworld.com/combined-calibrator-and-fieldbus-communicator).
- [22] Vít Hampl. Grafické rozhraní pro UltraGrid, 2009 [online; navštíveno 20.4.2014].
Dostupné z: .
- [23] Pavel Herout. *JAVA grafické uživatelské prostředí a čeština*, volume 1 vyd. Kopp, České Budějovice, 2001. ISBN 80-7232-150-1.
- [24] Pavel Herout. *Učebnice jazyka JAVA*, volume 1 vyd. Kopp, České Budějovice, 2002. ISBN 80-7232-115-3.
- [25] Jiří Peterka. Báječný svět počítačových sítí Část viii. - přenosové techniky, ii. , [online; navštíveno 5.4.2014].
Dostupné z: [/http://www.earchiv.cz/b05/b1100001.php3](http://www.earchiv.cz/b05/b1100001.php3).

- [26] Ing. Radek Šponar. Vlastnosti a užití průmyslových sběrnic , [online; navštíveno 6.3.2014].
Dostupné z: /<http://www.elektrorevue.cz/clanky/04019/index.html>.
- [27] STMicroelectronics. Rm0090 reference manual, 2014 [online; navštíveno 17.4.2014].
Dostupné z: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/reference_manual/DM00031020.pdf.
- [28] Filip Tůma. *Nové typy radarových hladinoměrů značky Siemens-Milltronics*. 2007, [online; navštíveno 7.3.2014].
Dostupné z: /http://www.odbornecasopisy.cz/index.php?id_document=36298.
- [29] Wikipedie. Netbeans, [online; navštíveno 2.2.2014].
Dostupné z: <http://cs.wikipedia.org/wiki/NetBeans>.
- [30] Wikipedie. Vzorkovací frekvence, [online; navštíveno 5.4.2014].
Dostupné z: /http://cs.wikipedia.org/wiki/Vzorkovac%C3%AD_frekvence.

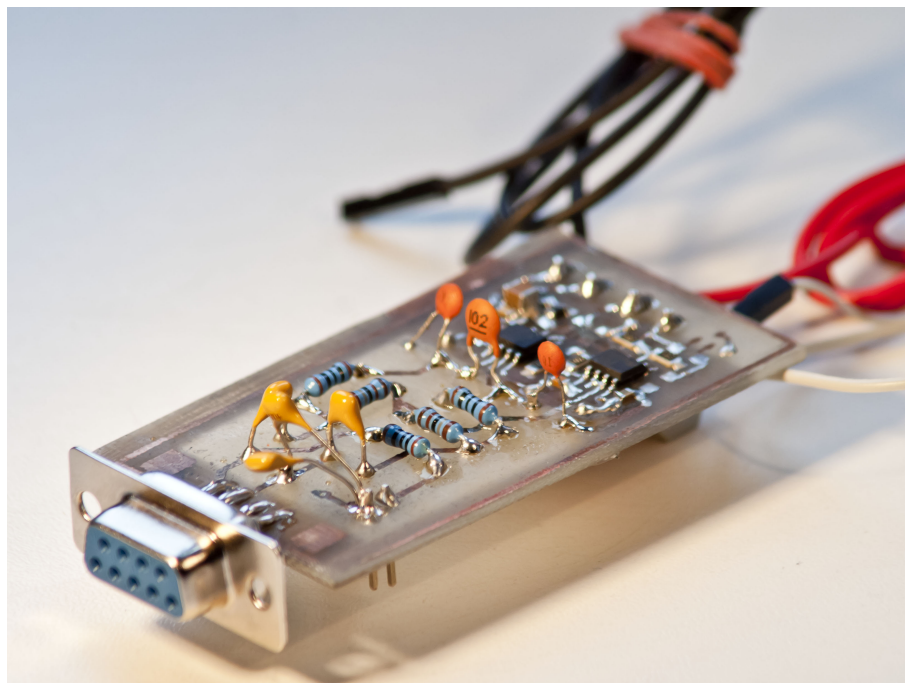
Kapitola 9

Příloha A

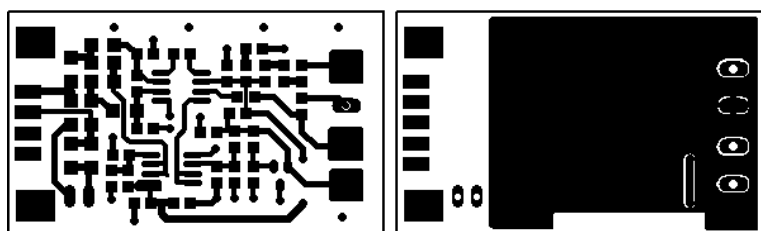
9.1 Dokumentace k hardwarové sondě FF Monitoru



Obrázek 9.1: Schéma zapojení hardwarové sondy



Obrázek 9.2: První prototyp realizace



Obrázek 9.3: Vrchní a spodní strana desky plošných spojů, zobrazena v měřítku 1:1

Kapitola 10

Příloha B

10.1 Dekódování hlavičky rámce

Při dekódování hlavičky bylo vycházeno z obrázku 3.6, protože rozkódování všech 23 rámců je velice dlouhé, je zde ukázáno rozkódování pouze 12 rámců.

```
//Dekodovani hlavicky
public static String DecodePreamble(String frame) {
    char preamble = frame.charAt(0);
    // převedeni druhého znaku (dolních 4 bitů) na číslo
    int druhy = convertToInt(frame.charAt(1));
    // prvotni rozkodovani, dle horních 4 bitů (prvního znaku)
    switch (preamble) {

        case '0':
            if (druhy == 6) {
                //ramec("TL-Transfer LAS");
                return "TL";
            } else {
                //ramec("CL -Claim LAS");
                return "CL";
            }
        case '1':
            //pokud jsou spodní 4 bity celočíselně dělitelné 2
            if (druhy % 2 == 0) {
                //pokud jsou spodní 4 bity celočíselně dělitelné 4
                if (druhy % 4 == 0) {
                    //ramec("CT - Compel Time");
                    return "CT";
                } else {
                    //ramec("Idle");
                    return "Idle";
                }
            }
    }
}
```

```
    } else {
        if ((druhy - 1) % 4 == 0) {
            //ramec("TD - Time Distribution");
            return "TD";
        } else {
            //ramec("SR - Status Response");
            return "SR";
        }
    }
case '2':
    if (druhy == 6) {
        //ramec("PN - Probe Node");
        return "PN";
    } else {
        if (druhy == 7) {
            //ramec("PR - Probe Response");
            return "PR";
        } else {
            //ramec("RI - Request Interval");
            return "RI";
        }
    }
}
case '3':
    if (druhy == 4) {
        //ramec("RT - Return Token");
        return "RT";
    } else {
        //ramec("PT - Pass Token");
        return "PT";
    }
}
case '5':
    if (druhy % 4 == 0) {
        //ramec("DT5 - Data(no SA)");
        return "DT5";
    } else {
        //ramec("DT3 - Data (contains SA)");
        return "DT3";
    }
}
default:
    // hlavicka nebyla dekodovaná jedná se o jiný rámeček
    System.out.println("Jiny ramec ");
    return "Jiny frame";
}
}
}
```

Kapitola 11

Příloha C

11.1 Vytvoření a nastavení grafu

```
// vytvoření nového okna s hlavičkou rámce
JFrame window = new JFrame(frame.getWindow());
//ziskani kontejneru pro přidávání jednotlivých komponent
Container c = window.getContentPane();
//nastavení layout manažeru
c.setLayout(new BorderLayout());
// vytvoření grafu s možností zoom
ZoomableChart chart = new ZoomableChart();
//nastavení popisků os
chart.getAxisX().setAxisTitle(new AxisTitle("t [us]"));
chart.getAxisY().setAxisTitle(new AxisTitle("U [V]"));
//vytvoření objektu pro ukládání dat
ITrace2D trace = new Trace2DSimple("Frame");
//nastavení barvy zobrazovaných dat
trace.setColor(Color.RED);
//nastavení tloušťky vynášecí křivky
trace.setStroke(new BasicStroke(2));
//nastavení popisků dat
trace.setPhysicalUnits("us", "V");
// přidání datového objektu do grafu
chart.addTrace(trace);
// načtení bodů daného rámce
byte[] points;
points = frame.getPoints();
//pomocné proměnné
double g = 0;
double v = 0;
double invert = 0;
// přidání bodů do grafu
```

```
for (int i = 0; i < points.length - 1; i++) {
    //uprava měřítka osy x, dle vzorkovací frekvence
    g = g + 2.19;
    //invertování hodnot, protože sonda vrací převrácený signál
    invert = points[i] * (-1);
    // neznaménkový byte (java ma původně byte znaménkový)
    if (invert < 0) {
        v = invert + 255;
    } else {
        v = invert;
    }
    //vypočtení správné hodnoty Y dle rozsahu A/D převodníku
    trace.addPoint(g, v / 255 * 3);
}
//nastavení ukazování aktuální polohy myši (štítek)
chart.setToolTipText(Chart2D.ToolTipType.VALUE_SNAP_TO_TRACEPOINTS);
// přidání grafu do kontejneru hlavního okna
c.add(new ChartPanel(chart));
window.pack();
//nastavení velikosti okna
window.setSize(1000, 500);
//nastavení viditelnosti okna
window.setVisible(true);

// přidání reakcí na události
this.addWindowListener(new WindowAdapter() {
    // zavření okna pomocí křížku
    @Override
    public void windowClosing(final WindowEvent e) {
        System.exit(0);
    }
});
} else {
}
}
```

Kapitola 12

Příloha D

12.1 Obsah přiloženého CD

V tabulce 12.1 jsou uvedena jména všech kořenových adresářů přiloženého CD s popisem obsahu.

Jméno adresáře	Popis obsahu
dp	- diplomová práce v pdf formátu
hardware	- schéma zapojení a deska plošných spojů hardwarové sondy
firmware	- zdrojové kódy pro DISCOVERY kit
software	- zdrojové kódy pro FF Monitor
instalace	- uživatelský manuál a instalace virtuální platformy
log	- záznam rámců z testování
foto	- fotografie z testování a hardwarové sondy
obrázky	- obrázky použité v práci

Tabulka 12.1: Obsah přiloženého CD