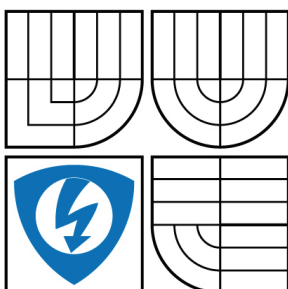


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

PARALELNÍ ROZHRANÍ ŘÍZENÉ SBĚRNICÍ USB

PARALEL INTERFACE CONTROLLED BY USB

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

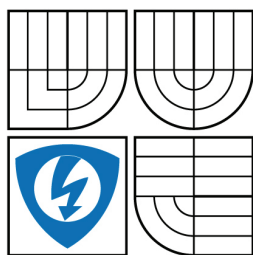
AUTOR PRÁCE
AUTHOR

MAREK VONDRA

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. ALEŠ PROKEŠ, Ph.D.

BRNO 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Bakalářská práce

bakalářský studijní obor
Elektronika a sdělovací technika

Student: Marek Vondra

ID: 101884

Ročník: 3

Akademický rok: 2008/2009

NÁZEV TÉMATU:

Paralelní rozhraní řízené sběrnici USB

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte možné koncepce USB řadičů pro připojení externího zařízení s paralelním portem k PC prostřednictvím USB 2.0. Navrhněte možná řešení s ohledem na realizovatelnost potřebného HW a SW a dosažení maximální přenosové rychlosti.

Seznamte se s činností a programovacími postupy digitálního kvadrurního modulátoru AD9857 firmy Analog Devices. Navrhněte a realizujte převodník rozhraní USB na paralelní port vývojové desky kvadrurního modulátoru a ověřte funkčnost vzájemné komunikace.

DOPORUČENÁ LITERATURA:

[1] BUKHARD, K. USB-měření, řízení a regulace pomocí sběrnice USB. Praha: BEN, 2002.

[2] AD9857-CMOS 200 MSPS 14-bit Quadrature Digital Upconverter [online], dostupné na <http://www.analog.com>

Termín zadání: 9.2.2009

Termín odevzdání: 5.6.2009

Vedoucí práce: doc. Ing. Aleš Prokeš, Ph.D.

prof. Dr. Ing. Zbyněk Raida
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Abstrakt

Bakalářská práce je zaměřena na využití sběrnice USB, v tomto případě na připojení převodníku USB -> 14-bit paralelní port, na který se následně připojí další zařízení a to kvadrurní modulátor. Sběrnice USB byla zvolena díky své univerzálnosti a také možnosti přenášet data velkou rychlostí do dalších připojených zařízení. Práce se dále zabývá prostředky pro splnění problematiky, zejména tak rozbořem výběru vhodného typu řadiče pro USB, jeho nastavením a naprogramováním, naprogramováním ovladačů pro systém Windows a také obslužným programem v systému Matlabu. Bakalářská práce navazuje na diplomovou práci Ing. Igora Majky, protože vybraný řadič byl v obou případech stejný, který navrhl konstrukci převodníku a nastínil možnost dalšího postupu v řešení problematiky.

Klíčová slova

Převodník USB na paralelní port, sběrnice USB, řadič sběrnice USB, Cypress, kvadrurní modulátor, ovladače pro Windows

Abstract

Bachelor's work is bent on usage busbar USB, in this case on interface inverter USB -> 14-bit parallel port, whereupon subsequently will add next device namely quadrature modulator. Busbar USB was elect thanks his universality as well as possibilities transfer data big at speeds of to the of other incorporate arrangement. Work further deal with resources for performance problems, especially so analysis selection fit type controller for USB, his setting and programmed, programmed drivers for system Windows as well as service programme in system Matlab. Bachelor's work tie together on diploma work Ing. Igor Majko, because choice controller was in either event same that the suggested construction inverter and foreshadowed possibility next progress in buckthorn problems.

Keywords

Inverter USB on paralel port, busbar USB, controller busbar USB, Cypress, quadrature modulator, drivers for Windows

VONDRA, M. *Paralelní rozhraní řízené sběrnici USB*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 65 s. Vedoucí bakalářské práce doc. Ing. Aleš Prokeš, Ph.D.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Paralelní rozhraní řízené sběrnicí USB jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 5. června 2009

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce doc. Ing. Aleši Prokešovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne 5. června 2009

.....
podpis autora

OBSAH

Seznam obrázků	iv
Seznam tabulek	vi
Úvod	1
1 O sběrnici USB	2
1.1 Základní specifikace a softwarové vlastnosti	2
1.2 Přenos dat.....	3
1.2.1 Funkce paketů	4
1.2.2 Funkce rámců.....	5
1.2.3 Trubice a koncové body.....	5
1.2.4 Druhy přenosů.....	6
1.3 Zařízení Plug and Play a enumerace.....	7
1.3.1 Koncový bod 0.....	8
1.4 Deskriptory	9
1.5 Architektura USB	12
1.6 Elektrická specifikace a hardwarové připojení.....	12
1.7 Jednotka SIE a součásti.....	14
2 Řadiče sběrnice USB a vybraný řadič Cypress CY7C68013a	15
2.1 Řadiče firmy Texas Instruments	15
2.2 Řadiče firmy FTDI Chip.....	15
2.3 Řadiče firmy Cypress.....	17
2.4 Podrobnější popis C7CY68013	18
2.4.1 Obvody řadiče pro komunikaci s okolím.....	18
2.4.2 Stav šetření energie a reset.....	19
2.4.3 Funkce mikroprocesoru a pamětí.....	20
2.4.4 Enumerace a Renumerace.....	21
2.5 Řadiče ostatních firem	22
3 Kvadrurní modulátor AD9857	24
3.1 Základní vlastnosti modulátoru	24

3.2	Mód kvadrurní modulace	25
3.3	Jednotónový mód	25
3.4	Interpolační digitálně-analogový mód	26
3.5	Funkce dalších bloků AD9857	26
3.5.1	Časování vstupních dat	26
3.5.2	CIC filtry a fixní interpolátor 4x	27
3.5.3	Kvadrurní modulátor	28
3.5.4	Jádro DDS	28
3.5.5	Inversní SINC filtr	28
3.5.6	Zbylé bloky	28
3.6	Programování a nastavování	29
3.6.1	Instrukční bajt	29
3.6.2	Ovládací registry	29
3.6.3	Programovací software	30
3.7	Power Management	32
4	Konstrukce převodníku	33
4.1	Výchozí požadavky na konstrukci	33
4.2	Samotná hardwarová konstrukce	33
4.2.1	Napájení	33
4.2.2	Paralelní rozhraní	34
4.2.3	Sériové rozhraní	34
4.2.4	Spínač S2	34
4.2.5	Sériová paměť EEPROM	34
4.2.6	Sériový port SIO-1	35
4.2.7	Konektor SV2	35
4.2.8	Ostatní piny	35
5	Software a drivery převodníku	36
5.1	Struktura ovládání	36
5.2	Ovladače	37
5.2.1	Základní poznatky o ovladačích	37
5.2.2	Zdrojový kód ovladače	37
5.2.3	Tvorba ovladače	39
5.2.4	Aspekty tvorby	40

5.3	GPIF Designer	41
5.4	Firmware FrameWorks	42
5.5	Procesor 8051	43
5.6	EZ-USB Control Panel	44
5.7	Uživatelské prostředí v Matlabu	44
6	Závěr	45
	Literatura	47
	Seznam symbolů, veličin a zkratk	48
	Seznam příloh	49

SEZNAM OBRÁZKŮ

Obr. 1.1:	Logo USB [9].....	2
Obr. 1.2:	Nákres propojovacího kabelu	3
Obr. 1.3:	Přenášený paket USB [8]	5
Obr. 1.4:	Pakety řídicího přenosu [8]	6
Obr. 1.5:	Paket pro přenos při přerušení [8].....	6
Obr. 1.6:	Pakety pro hromadný přenos [8].....	7
Obr. 1.7:	Paket pro Izochronní přenos [8].....	7
Obr. 1.8:	Topologie USB	12
Obr. 1.9:	Zásuvka typu a a zástrčka typu a pro hostitelské zařízení, zásuvka typu B a zástrčka typu B pro připojená zařízení.....	13
Obr. 1.10:	Zařízení s „low-speed“ [10]	13
Obr. 1.11:	Zařízení s „full-speed“ [10]	14
Obr. 2.1:	Zjednodušené blokové schéma čipu FT2232D [6]	16
Obr. 2.2:	Zjednodušené blokové schéma čipu VNC1L [6].....	17
Obr. 2.3:	Zjednodušené blokové schéma čipu Cypress CY7C68013 [3].....	18
Obr. 2.4:	Uspořádání vnitřního registru čipu FX2 [3].....	20
Obr. 3.1:	Blokové schéma modulátoru AD9857 [5]	24
Obr. 3.2:	Blokové schéma jednotónového módu [5]	25
Obr. 3.3:	Blokové schéma interpolačního digitálně-analogového módu AD9857 [5] 26	
Obr. 3.4:	Časování paralelního portu v módu kvadrurní modulace [5]	27
Obr. 3.5:	Časování paralelního portu v interpolační DAC módu [5].....	27
Obr. 3.6:	Charakteristika CIC filtru [5].....	27
Obr. 3.7:	Časový diagram pro zápis dat do AD9857 [5].....	29
Obr. 3.8:	Výběr a nastavení jednoho ze čtyř profilů	31
Obr. 3.9:	Nastavení registrů v daném profilu a kontrolních registrů	31
Obr. 3.10:	Nastavení filtrů, komunikace, vnitřního kmitočtu a módu AD9857	32
Obr. 5.1:	Blokové schéma ovládání pro FX2 [8]	36
Obr. 5.2:	Blokové zobrazení funkce systému z hlediska ovladače	40
Obr. 5.3:	Hlavní panel	41
Obr. 5.4:	Nastavení funkce při čtení paměti FIFO	42

Obr. 5.5: Vývojový diagram zdrojového kódu pro FX2 [8]	43
--	----

SEZNAM TABULEK

Tab. 1.1:	Druhy PID v USB 2.0	4
Tab. 1.2:	Naznačení struktury Setup paketu [8].....	8
Tab. 1.3:	Standardní kódy požadavků.....	8
Tab. 1.4:	Typy deskriptorů [8]	9
Tab. 1.5:	Deskriptor zařízení [8]	9
Tab. 1.6:	Deskriptor konfigurace [8].....	10
Tab. 1.7:	Deskriptor rozhraní [8]	11
Tab. 1.8:	Deskriptor koncového bodu [8]	11
Tab. 2.1:	Obsah EEPROM pro enumeraci [8]	21
Tab. 2.2:	Obsah paměti při načtení firmwaru [8].....	22
Tab. 2.3:	Konfigurační bajt [8].....	22
Tab. 3.1:	Instrukční bajt [8].....	29

ÚVOD

S postupným rozvojem technologií je možné sloučit práci více zařízení do jednoho, využít jejich větší přenosovou rychlost a možnost připojení jedním kabelem, který snadno obstará veškerý přenos dat. k těmto technologiím patří i USB. Ve verzi 2.0 se jedná o univerzální sběrnici, která je schopná přenášet data velkou rychlostí, má vlastní napájení sběrnice a zařízení se ke sběrnici mohou připojovat bez nutnosti vypínání nebo restartu zařízení či hostitele. Instalace ovladačů probíhá vždy jen při prvním připojení, poté je ovladač sám opětovně nalezen.

Pro připojení zařízení ke sběrnici je nutný převodník, který převede data z USB na data použitelná pro dané zařízení, případně s nimi bude dále pracovat, aby je upravil na požadovaný formát. Jednou z možností, která i zde bude popsána, je připojení řadiče USB od firmy Cypress CY68013, který převede sériová data z USB na 14 bitový paralelní port. k tomu je dále nutné přenášet řídicí data k připojenému zařízení na paralelní port, kterým v tomto případě bude kvadrurní modulátor AD9857.

Pro úplné pochopení a popsání problematiky je zde na počátku rozbor funkce a vlastností sběrnice USB. Popis sběrnice by byl poměrně obsáhlý, tak jsou zde popsány jen základní prvky pro získání základního pojmu o funkci, které si je možno dále prohloubit v některé z použitých literatur. Další kapitolou v této bakalářské práci je srovnání a zhodnocení některých řadičů USB od různých firem a to z konstrukčního i softwarového hlediska. Zbývá část práce je věnována konstrukčnímu řešení převodníku a také návrhu softwaru. Ovládaní převodníku je poměrně složitá a dlouhodobá problematika, protože se jedná o program, který bude uložen v paměti řadiče, a který bude vykonáván při jeho funkci. Dále jsou to ovladače pro PC, ke kterému je řadič připojen a poslední v řadě i ovládací aplikace, se kterou bude uživatel pracovat.

1 O SBĚRNICI USB

Sériová sběrnice USB (Universal Serial Bus) byla vyvíjena od roku 1995 firmami Compaq, HP, Intel, Philips, NEC, Microsoft atd., jako náhrada připojení pro zařízení, která si nevystačí s menší přenosovou rychlostí a jsou připojována k PC, zejména pomocí RS232. Bylo pak možné pomocí jednoho standardního konektoru připojit libovolné zařízení (až 127 zařízení), které tento standart podporovalo. Týkalo se to zejména tiskáren, klávesnic a polohovacích zařízení. k tomu vystačoval standart USB 1.1, který pracoval v režimech „low-speed“ s přenosovou rychlostí 1,5Mbit/s a později „full-speed“ s přenosovou rychlostí 12Mbit/s. Postupně se zvyšovaly nároky, protože USB začínala využívat další zařízení a to zejména scannery a videokamery, které zejména do té doby využívaly připojení pomocí FireWire (IEEE 1394). To umožňovalo přenos 400Mbit/s. S důrazem na vyšší přenosovou rychlost a zpětnou kompatibilitu vznikl v roce 2000 standart USB 2.0, který má v režimu „high-speed“ přenosovou rychlost 480Mbit/s. Nyní je ve vývoji standart USB 3.0, který ale v době tvorby této práce nebyl blíže specifikován, pouze jeho předpokládané vlastnosti, kterými se není potřeba pro tuto práci zabývat. Pro následující podkapitoly bylo využito informací ze zdrojů [1], [2], [8], [9], [10] a [11].

1.1 Základní specifikace a softwarové vlastnosti

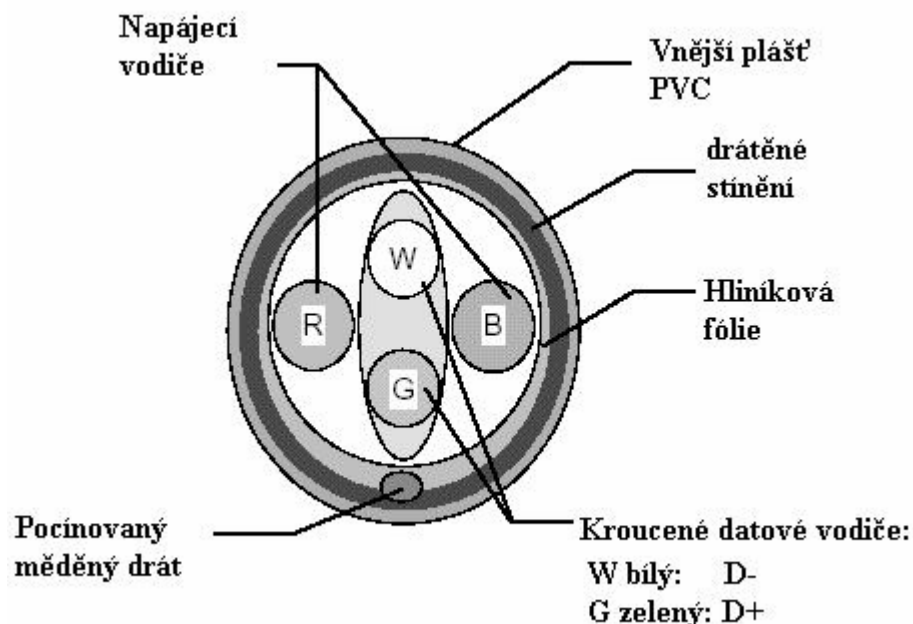
S USB, stejně jako s mnohými jinými standarty, je spojena rozsáhlá softwarová i hardwarová specifikace. Pro lepší přiblížení zde budou uvedeny ty nejdůležitější parametry. Zařízení USB jsou označena, ne jen na konektorech, nepřehlédnutelným symbolem, který je na Obr. 1.1. Dále jsou konektory nezáměnné s jinými zařízeními.



Obr. 1.1: Logo USB [9]

USB nyní existuje ve dvou verzích a to USB 1.1 a USB 2.0. Jak již bylo řečeno výše. USB 1.1 má dvě fyzické vrstvy a to „low-speed“ s přenosovou rychlostí 1,5Mbit/s a „full-speed“ s přenosovou rychlostí 12Mbit/s. USB 2.0 doplňuje třetí vrstvu a to „high-speed“ s přenosovou rychlostí 480Mbit/s. Na port USB může být připojeno až 127 zařízení, mezi které se rozdělí přenosová rychlost. Pro dodržení parametrů jsou normalizovány propojovací kabely, které mají maximální délku mezi dvěma připojenými zařízeními 5 metrů. Kabel je tvořen čtyřmi vodiči, kde dva jsou napájecí (+5V a zemnicí) a dva kroucené jsou datové (D+ a D-), jak je na Obr. 1.2. Na rozdíl od RS232 nejsou u USB vodiče křížené, ani nejsou potřebné další vodiče pro

synchronizaci. Vodič VBus (+5V) je červený, GND je černý, D+ je zelený a D- je bílý.



Obr. 1.2: Nákres propojovacího kabelu

Mezi další vlastnosti USB, některé z nich budou dále více popsány, zejména patří automatická identifikace periférií, kde si počítač přesně zjistí, o jaké zařízení se jedná a podle toho nainstaluje a nakonfiguruje odpovídající ovladače. Dále lze zařízení libovolně připojovat a odpojovat bez nutnosti restartu operačního systému. Více zařízení může pracovat současně, což bude vysvětleno níže, a je i zde podpora synchronních i asynchronních přenosů. Používá se jednoduchý a velice efektivní protokol, který umožňuje široký rozsah délky přenášených paketů. Součástí protokolu je i zpracování chyb a řízení přenosu buffery. Mohou se sdružovat zařízení s mnoha sdruženými funkcemi. Šířky pásma přenosu jsou garantována pro jednotlivá zařízení, které to vyžadují, pohybují se od několika kb/s do několik Mb/s. Celá šířka pásma může být využita i jedním zařízením. Mimo toto vše je zde i podpora pro identifikaci vadných zařízení, identifikace chyb při přenosu a možnost jejich korekce. Momentálně je USB rozšířeno prakticky do všech operačních systémů a tak není problém s kompatibilitou. Nevýhodou použití USB je její volný vývoj, protože každé zařízení USB má interní číslo dodavatele (Vendor-ID), které je udělováno organizací USB. Dále je zde poměrně velká složitost hardwaru a rozsáhlé softwarové ovladače na straně PC.

1.2 Přenos dat

USB má pouze jedno zařízení master, všechny akce tedy pocházejí od hostitele a připojená zařízení plní jeho požadavky, samotná zařízení nemohou sama od sebe nic vysílat.

Průběh přenosu dat je určován masterem (hostitel) a připojená zařízení (slave) se musí podle mastera synchronizovat. Synchronizace se získá z přenášeného signálu tak, že se využije metody NRZI (Non-Return-To Zero). Log. 0 v datech znamená změnu logické úrovně a log. 1 je nechávající beze změny. Toto se provádí čistě na hardwarové úrovni, tedy kódování a dekodování. Přijímač v zařízení je navržen tak, aby byl schopen přijmout signál a dekodovat ho. Přijímač a vysílač jsou hardwarově realizovány v jedné součástce. Dalšími prostředky je zajištěno, aby nedocházelo ke ztrátě synchronizace. Pokud v datovém signálu je šest po sobě jdoucích log. 1, je nutné za ně přidat log.0, aby došlo ke změně úrovně. Tento proces je označen jako vkládání bitů (bit stuffing). Přijímač pak automaticky tuto změnu odstraní. Každý datový packet má na začátku zaváděcí bajt (00000001b) kvůli synchronizaci. Přijímač tak vidí, díky kódování NRZI, několik střídajících se stavů, na které se může zasynchronizovat. Během dalšího přenosu zůstane synchronizace zachována.

Řadič USB je vybaven jednotkou SIE (Serial Interface Engine), která obstarává komunikaci se zařízením, ke kterému je připojeno. k tomu, aby data z jednotky SIE mohla být doručena k mikrořadiči k dalšímu zpracování, slouží paměti FIFO (First In – First Out). Tyto paměti pracují jako posuvné registry, tedy první bit, který se do nich uloží je také jako první čten. Mikrořadiči tedy stačí si data z jedné FIFO přečíst a potřebná data do druhé zapsat. Ostatní následující operace opět vyřídí SIE. Zařízení USB disponuje několika pamětmi FIFO, pomocí nichž se data přenášejí.

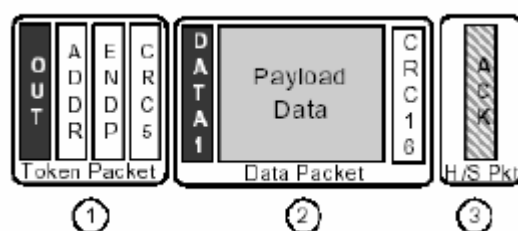
1.2.1 Funkce paketů

Základní částí přenosu jsou pakety. Posílaná data se vysílají a přijímají v paketech, přičemž má každý paket svou hlavičku (Packet ID nebo PID). Hlavička rozlišuje o jaký paket se jedná a může to být jeden ze čtyř druhů, které je uvedeny v Tab. 1.1.

Tab. 1.1: Druhy PID v USB 2.0

Typ PID	Jméno PID
Token	IN, OUT, SOF, SETUP
Data	DATA0, DATA1, DATA2, MDATA
Handshake	ACK, NAK, STALL, NYET
Special	PRE, ERR, SPLIT, PING

Obr. 1.3 ilustruje jeden příklad přenosu dat po sběrnici USB. První paket má PID OUT a patří tedy do typu Token, to znamená, že data jsou přenášena směrem do připojeného zařízení. První paket dále obsahuje adresu koncového bodu a zabezpečení CRC 5. Druhý paket má PID DATA1 a patří tedy do typu Data. Obsahuje přenášená data a zabezpečení CRC 16. Posledním, třetím paketem je paket typu Handshake, protože PID je ACK. To znamená, že paket byl úspěšně přenesen.



Obr. 1.3: Přenášený paket USB [8]

Pro lepší pochopení určení paketů je dobré vysvětlit funkce některých PID. a to např. IN, což je přenos dat do PC, OUT je přenos dat z počítače, SOF (Start of Frame) je počátek rámce a SETUP se používá pro kontrolní přenosy. v druhé skupině Data mají pakety zhruba stejný význam, posílají se postupně za sebou, což značí jejich číslo. PID typu Handshake se užívají pro kontrolu přenosu. Např. ACK znamená bezchybné přijetí paketu, NYET má ekvivalentní význam, dále ale doplňuje, že koncový bod nemůže zrovna přijmout další paket. NAK vypovídá o špatně přijatých datech a STALL říká o jiné chybě v komunikaci. Ve skupině typu Special jsou PID, která slouží ke zkoušení trasy, její rychlosti apod.

1.2.2 Funkce rámců

Data v USB se přenáší v rámcích o definované délce 1ms. Komunikace může probíhat buď v malých rámcích o délce 8 bajtů nebo v delších o délce 256 bajtů. Každý rámeček začíná SOF (Start of Frame) paketem, od kterého se určuje časová základna každého připojeného zařízení na sběrnici USB. SOF paket obsahuje jedenácti bitové číslo, které je vždy inkrementováno s každým rámcem a také je vždy uloženo v registru zařízení.

Při rychlosti přenosu „high-speed“ je jeden 1ms rámeček rozdělen do osmi 125 mikrosekundových mikrorámčů. Každý mikrorámeček začíná paketem SOF, který je inkrementován jen jednou za 1 ms. Proto je u rychlejších zařízení nutný ještě čítač mikrorámčů. Tímto způsobem mohou být v jednom rámcu pakety pro rychlá a pomalá zařízení. Posílá-li tedy PC data více zařízením, může je posílat plnou rychlostí a rozdělovač HUB pak rozdělí data daným zařízením a zabrání i tomu, aby se data s větší přenosovou rychlostí dostala na „pomalá“ zařízení.

1.2.3 Trubice a koncové body

Každé zařízení připojené přes USB má několik koncových adres (endpoint), na která jsou přenášena data. Endpoint 0 je primárně určen k inicializaci. Pokud hostitel požaduje nějaká data, připojené zařízení je v určených časových intervalech zapíše na endpoint 1 a odtud si je hostitel vybere. Software pro USB vytváří datový kanál k jednomu endpointu k jednomu zařízení. Tento kanál se nazývá trubice (pipe). Pro lepší pochopení si trubici můžeme představit jako jeden vodič vedený k jednomu endpointu. z principu USB ale vyplývá, že data přenášená v trubicích jsou datové pakety přenášeny v 1ms rámcích a pak jsou následně hardwarem rozdělována do fyzických pamětí, podle jejich koncové adresy. k jednomu zařízení může směřovat několik trubic, tím přenosová rychlost vzroste.

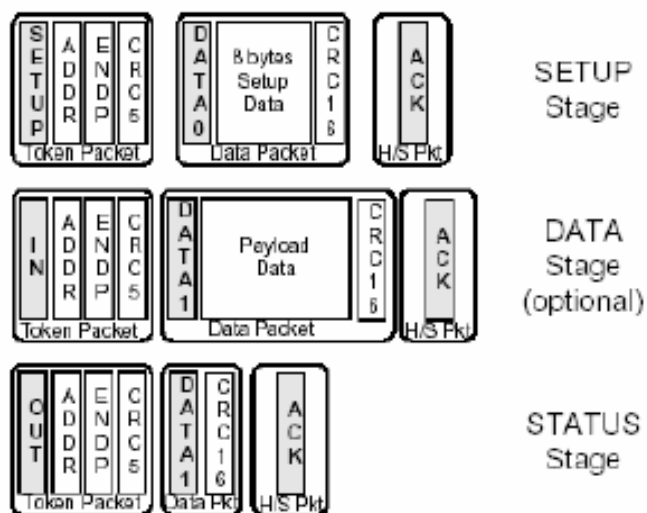
K přenosu se využívají paměti FIFO, kterých je obecně několik. Do nich se data

načtou a pak postupně vyčítají. Koncových bodů může být několik a tak je při přenosu důležité, aby byla na počátku obsažena adresa zařízení a taky adresa koncového bodu.

1.2.4 Druhy přenosů

Při komunikaci mezi PC a připojeným zařízením jsou přenášena data jedním ze čtyř možných druhů přenosů. Je to buď řídicí přenos, přenos při přerušení, hromadný přenos nebo izochronní přenos. Liší se délkou jednotlivých paketů a jejich uspořádáním.

Řídicí přenos (Control transfer) využívá k ovládání hardwaru řídicích dotazů (Control requests). Hlídání chyb je automatické a pracují s vysokou prioritou. v každé části jednoho rámce je rezervovaná část pro řídicí přenos a může být tvořen ze dvou nebo i tří částí. První část Setup stage obsahuje PID SETUP, dále následuje adresa koncového zařízení a koncového bodu, zabezpečení CRC 5. Poté PID DATA0, 8 bajtů určovacích dat, zabezpečení CRC 16 a nakonec PID ACK. v druhé části se už jedná přímo o přenos dat do PC a je volitelná, pokud je potřeba přenést více dat. v poslední, třetí části, se indikuje úspěšné dokončení přenosu. Viz. Obr. 1.4.



Obr. 1.4: Pakety řídicího přenosu [8]

Přenos při přerušení (Interrupt transfer) se využívá pro zařízení, která vysílají malé množství dat, což jsou např. polohovací zařízení. Funkce je taková, že zařízení master se dotazuje po každém určitém časovém intervalu na nová data. Pro nižší rychlost se přenáší data o velikosti 8, 16, 32 nebo 64 bajtů, pro nejvyšší rychlost „High-speed“ se přenáší až 1024 bajtů. Struktura paketu je zobrazena na Obr. 1.5.



Obr. 1.5: Paket pro přenos při přerušení [8]

Hromadný přenos (Bulk transfer) slouží pro přenos velkého množství dat, kde je vyžadováno hlídání chyb, ale tím zde odpadá požadavek na čas přenosu. Typicky je tento přenos určen pro zařízení, jakými jsou třeba tiskárna nebo skener. Priorita přenosu je tady nízká a rychlost přenosu dat závisí na okamžitém vytížení sběrnice. Struktura a velikost paketu se podobá přenosu při přerušení, což je znázorněno na Obr. 1.6.



Obr. 1.6: Pakety pro hromadný přenos [8]

Izochronní přenos (Isochronous transfer) slouží k přenosu velkého množství dat, kde je i definována přenosová rychlost. Neprovádí se zde korekce chyb, protože jde hlavně o dostatečný datový tok a jednotlivé chyby nejsou omezující nebo rozhodující. Maximální délka paketů je 1024 bajtů a při rychlosti „high-speed“ je možné přenést během jednoho mikrorámce do jednoho koncového bodu tři pakety. Pro tento přenos je stále rezervována část kapacity sběrnice. Struktura je na Obr. 1.7.



Obr. 1.7: Paket pro Izochronní přenos [8]

1.3 Zařízení Plug and Play a enumerace

Výhodu, kterou Plug and Play poskytuje je odpojování a hlavně připojování zařízení bez nutnosti restartování systému. Další výhodou je nepotřebnost opětovného instalování ovladačů, to se děje pouze při prvním připojení, pak jsou drivery zaváděny automaticky při rozpoznání zařízení. Při připojení zařízení počítač pozná, že bylo připojeno nové zařízení podle zdvižené datové linky, tím i pozná rychlost připojení, kterou může zařízení komunikovat (viz. Elektrická specifikace). Poté se provedou následující kroky:

- hub dá informaci hostiteli o tom, že bylo připojeno nové zařízení
- hostitel si dotazem na hub zjistí, na kterém portu je zařízení připojeno
- po zjištění portu dá hostitel pokyn k připojení portu a k vynulování sběrnice (reset)
- hub poté vytvoří reset o délce 10 ms a přidělí zařízení defaultní adresu 0, na kterou se bude hostitel prozatím dotazovat, uvolní se i proud 100mA pro zařízení
- hostitel se nyní zařízení dotazuje přes end-point 0 na deskriptory zařízení (deskriptory obsahují základní informace o zařízení, viz. níže), pomocí nich stanoví, jak velké datové pakety je možno vysílat

- poté je hostitelem přidělena zařízení adresa na sběrnici
- hostitel nyní načte ze zařízení na nové adrese všechny potřebné konfigurační informace
- podle načtených informací přiřadí hostitel zařízení velikost proudu (max. 500mA), zařízení je poté připraveno na použití

1.3.1 Koncový bod 0

Koncový bod 0 (Endpoint0) je základní částí každého zařízení a je důležitý pro enumeraci zařízení, která již byla popsána výše. Koncový bod 0 je obousměrný a pro přístup k němu se využívá řídicího přenosu. Všechny požadavky na parametry apod. se vysílají v Setup paketu, jeho struktura je pevně dána a je naznačena v Tab. 1.2.

Tab. 1.2: Naznačení struktury Setup paketu [8]

Offset	Název	Velikost	Popis
0	bmRequestType	1	charakteristika požadavku D7: směr přenosu (0 = PC -> zařízení, 1 = zařízení -> PC) D6..5: typ požadavku (0 = standardní, 1 = třída, 2 = výrobce, 3 = rezervováno) D4..0: příjemce (0 = zařízení, 1 = rozhraní, 2 = koncový bod, 4..31 = rezervováno)
1	bRequest	1	specifický požadavek, viz. Tab. 1.3
2	wValue	2	pole šířky slov a udává parametry k předchozímu poli
4	wIndex	2	pole šířky slova, typicky přenáší index nebo offset
6	wLength	2	počet přenášených bajtů pokud následuje datová část

Tab. 1.3: Standardní kódy požadavků

bRequest	Value
GET_STATUS	0
CLEAR_FEATURE	1
Rezervováno pro další použití	2
SET_FEATURE	3
Rezervováno pro další použití	4
SET_ADRESS	5
GET_DESCRIPTOR	6
SET_DESCRIPTOR	7
GET_CONFIGURATION	8
SET_CONFIGURATION	9
GET_INTERFACE	10
SET_INTERFACE	11
SYNCH_FRAME	12

Sběrnice USB podporuje více druhů požadavků. Mimo standardní požadavky k nim patří i požadavky třídy a výrobce. Požadavky třídy jsou přístupné, pokud se

zařízení přihlásí jako příslušník určité třídy a také jsou specifické pro daný typ. Požadavky výrobce jsou pro každé zařízení nadefinovány výrobcem a mohou se od sebe značně lišit. Standardní požadavky, viz. Tab. 1.3, jsou pro všechna zařízení USB nadefinována. Ta na ně musí odpovédět, i když ještě plně neproběhla enumerace.

1.4 Deskriptory

Jednotlivé vlastnosti a nastavení jsou určována pomocí deskriptorů. Jsou to datové struktury daného formátu, kde je na prvním místě pole udávající velikost v bajtech a následně za ním pole, které udává typ deskriptoru. Typy deskriptorů jsou v Tab. 1.4.

Tab. 1.4: Typy deskriptorů [8]

Typ deskriptoru	Hodnota
Zařízení	1
Konfigurace	2
Řetězec	3
Rozhraní	4
Koncový bod	5
Kvalifikace zařízení	6
Konfigurace jiné rychlosti	7
Výkon rozhraní	8

První z deskriptorů je deskriptor zařízení. Ten podává základní informace o zařízení jako celku a pro dané zařízení je vždy jen jeden. Jeho přiblížení umožňuje Tab. 1.5.

Tab. 1.5: Deskriptor zařízení [8]

Offset	Název	Velikost	Popis
0	bLength	1	velikost deskriptoru v bajtech
1	bDescriptorType	1	typ deskriptoru, zde DEVICE
2	bcdUSB	2	Číslo USB specifikace, které zařízení odpovídá, udává se v BCD kódu
4	bDeviceClass	1	Číslo třídy dle USB-IF
5	bDeviceSubClass	1	Číslo podtřídy dle USB-IF
6	bDeviceProtocol	1	Číslo protokolu dle USB-IF
7	bMaxPacketSize0	1	maximální velikost paketu pro koncový bod 0
8	idVendor	2	ID výrobce, přiděluje USB-IF
10	idProduct	2	ID výrobku, přiděluje výrobce
12	bcdDevice	2	verze zařízení v BCD kódu
14	iManufacturer	1	index deskriptoru řetězce popisujícího výrobce
15	iProduct	1	index deskriptoru řetězce popisujícího výrobek
16	iSerialNumber	1	index deskriptoru řetězce se sériovým číslem
17	bNumConfigurations	1	počet možných konfigurací

Dalším je deskriptor konfigurace. Ten udává specifické informace o konfiguraci zařízení, dále poskytuje informace o rozhraních v dané konfiguraci. v případě potřeby se společně pak posílají všechny deskriptory rozhraní a koncových bodů, které patří k dané konfiguraci. z tohoto vyplývá, že jedno zařízení může mít více deskriptorů konfigurace, konfigurace může mít více rozhraní a rozhraní může mít několik koncových bodů nebo jen koncový bod 0, který je základní. Aby nedocházelo ke kolizím, může sdílet stejný koncový bod více rozhraní, ale tyto rozhraní nemohou být zároveň v jedné konfiguraci. Pouze, kdyby byl koncový bod veden jako alternativní. Každé rozhraní má své vlastní nastavení, mezi nimiž je možno přepínat. Přiblížení deskriptoru konfigurace je v Tab. 1.6.

Tab. 1.6: Deskriptor konfigurace [8]

Offset	Název	Velikost	Popis
0	bLength	1	velikost deskriptoru v bajtech
1	bDescriptorType	1	typ deskriptoru, zde CONFIGURATION
2	wTotalLength	2	celková délka dat, která se budou odesílat, tedy včetně deskriptorů rozhraní, koncových bodů, tříd a specifických pro výrobce
4	bNumInterfaces	1	počet rozhraní podporovaných touto konfigurací
5	bConfigurationValue	1	hodnota se používá jako argument pro požadavek SetConfiguration
6	iConfiguration	1	index deskriptoru řetězce popisujícího konfiguraci
7	bmAttributes	1	charakteristika konfigurace D7: rezervováno D6: (0 = zařízení je napájeno externě, 1 = zařízení je napájeno ze sběrnice) D5: vzdálené probuzení (<i>Remote Wakeup</i>) D4..0 rezervováno
8	bMaxPower	1	maximální proud, který zařízení požaduje ze sběrnice, vyjádřeno v mA dělených dvěma (50 = 100mA)

Deskriptor rozhraní byl již zmíněn u deskriptoru konfigurace. Obsahuje specifické informace o rozhraní uvnitř dané konfigurace. Není možné ho vyžádat samostatně, pouze jako součást deskriptoru konfigurace. i když je zařízení nakonfigurováno, je možné měnit charakteristiky koncových bodů, k jejich změnám a monitorování slouží SetInterface a GetInterface. Přiblížení deskriptoru je v Tab. 1.7.

Tab. 1.7: Deskriptor rozhraní [8]

Offset	Název	Velikost	Popis
0	bLength	1	velikost deskriptoru v bajtech
1	bDescriptorType	1	typ deskriptoru, zde INTERFACE
2	bInterfaceNumber	1	Číslo tohoto rozhraní
3	bAlternateSetting	1	hodnota, která určuje aktivní alternativní nastavení z rozhraní definovaném v předchozím poli
4	bNumEndpoints	1	počet koncových bodů, které používá toto rozhraní
5	bInterfaceClass	1	Číslo třídy dle USB-IF
6	bInterfaceSubClass	1	Číslo podtřídy dle USB-IF
7	bInterfaceProtocol	1	Číslo protokol dle USB-IF
8	iInterface	1	index deskriptoru řetězce popisujícího rozhraní

Deskriptor koncového bodu je velice zásadní, je v něm určeno o jaký přenos se jedná, s jakou délkou přenášených dat a jaký je i směr přenosu. Pomocí tohoto deskriptoru se určuje i šířka pásma pro přenos dat. Popis je v Tab. 1.8.

Tab. 1.8: Deskriptor koncového bodu [8]

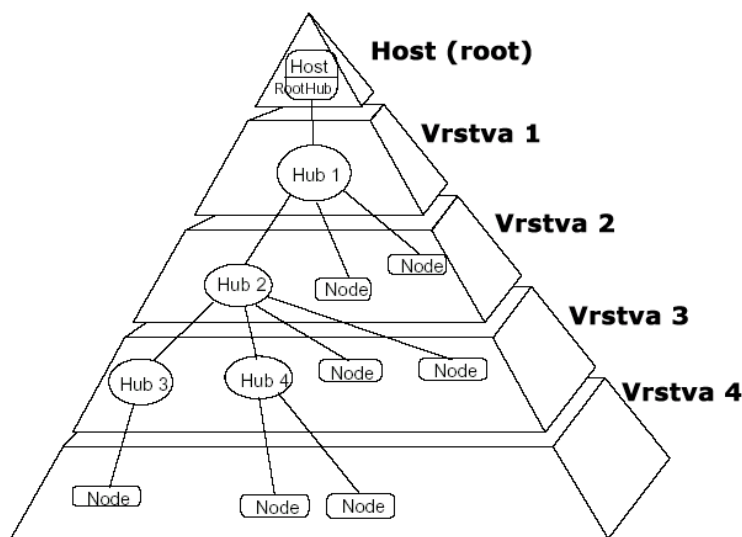
Offset	Název	Velikost	Popis
0	bLength	1	velikost deskriptoru v bajtech
1	bDescriptorType	1	typ deskriptoru, zde ENDPOINT
2	bEndpointAddress	1	Bit 3..0: číslo koncového bodu Bit 6..4: rezervováno Bit 7: směr přenosu (0 = OUT, 1 = IN)
3	bmAttributes	1	Bit 1..0: typ přenosu (00 = řídící, 01 = izochronní, 10 = hromadný, 11 = přerušení) Pro izochronní přenos se nastavují i další bity: Bit 3..2: typ synchronizace (00 = žádná synchronizace, 01 = asynchronní, 10 = adaptivní, 11 = synchronní) Bit 5..4: typ použití (00 = datový koncový bod, 01 = koncový bod zpětné vazby, 10 = implicitní datový koncový bod zpětné vazby, 11 = rezervováno)
4	wMaxPacketSize	2	maximální velikost paketu pro daný koncový bod Bit 10..0 udávají velikost paketu v bajtech Bit 12..11 určují počet přídavných přenosů v mikrorámci: (00 = 1 transakce v mikrorámci, 01 = 2 transakce v mikrorámci, 10 = 3 transakce v mikrorámci, 11 = rezervováno) Bit 15..13 jsou rezervovány.
6	bInterval	1	interval pro výzvu koncových bodů k přenosu

Poslední je deskriptor řetězce, ten je ale volitelný, pokud ho zařízení nepodporuje,

jeho hodnoty jsou nastaveny jako nulové.

1.5 Architektura USB

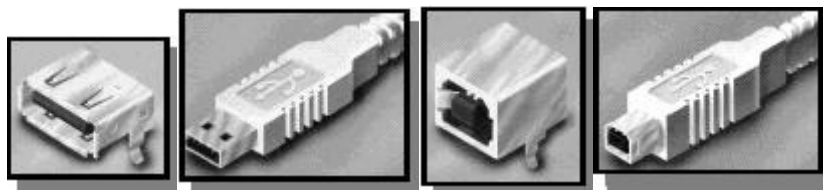
Zařízení, které se připojují k USB jsou buď koncová zařízení (myš, klávesnice, tiskárna) nebo rozbočovače (hub). Počítač je vždy zařízením typu master a zařízením typu slave. Připojení zařízení je tedy řešeno víceúrovňovou hvězdicovou strukturou. Středem propojení ve hvězdici je hub a ten spojuje buď centrální rozdělovač, který je v počítači (USB host), se zařízeními na nižší úrovni nebo jiný hub se zařízeními na vyšší úrovni (jinými rozdělovači). Každé zařízení má svoji přidělenou adresu a má i několik koncových jednotek, tzv. end-points, se kterými počítač komunikuje a tím předává a dostává data od zařízení.



Obr. 1.8: Topologie USB

1.6 Elektrická specifikace a hardwarové připojení

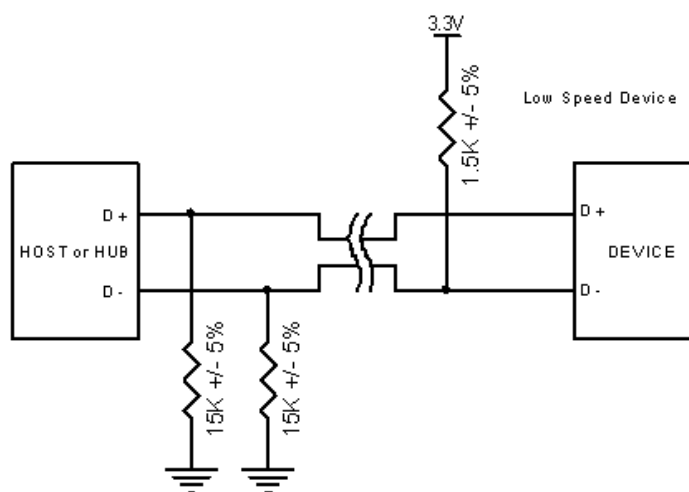
Zařízení USB lze připojit několika typy konektorů. Konektor typu „A“ je plochý konektor, který obsahují zejména hostitelská zařízení (počítač). Druhý typ konektoru je čtverhranný „B“, který je určen zejména pro připojená zařízení, jejich čelní pohled je na Obr. 1.9. Tímto je i určen standart kabelů, které slouží pro připojení zařízení. Dále jsou používány konektory typu micro USB a mini USB.



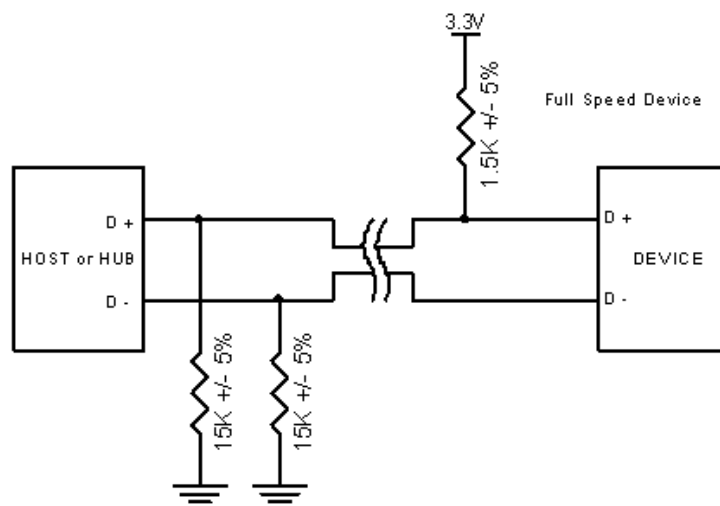
Obr. 1.9: Zásuvka typu A a zástrčka typu A pro hostitelské zařízení, zásuvka typu B a zástrčka typu B pro připojená zařízení

USB port může vydat maximálně proud 500mA a to, pokud si o to zařízení samo požádá pomocí deskriptoru. Standardní proud, který je zařízením přidělen je 100mA. Při zvyšování se proud zvětšuje vždy o 100mA. Proud spícího zařízení je maximálně 0,5mA. Pokud je potřebná velikost proudu větší, je nutný externí zdroj. Když bude proud větší jak maximálních 500mA, je zařízení ihned odpojeno od napájení, aby nedošlo k poškození.

Zapojením rezistoru mezi 3,3V a D+ nebo D- je definována přenosová rychlost zařízení. Zařízení „full-speed“ mají připojeno D+ k 3,3V. Tím je definována přenosová rychlost 12Mbit/s. Odporů jsou zároveň použity pro detekci, zda bylo do portu zapojeno zařízení. U rychlostí „full-speed“ a „high-speed“ je logická 1 přenášena diferencially uvedením D+ přes 2,8V s připojeným 15kΩ odporem na zem a D- je pod 0,3V s 1,5kΩ odporem připojeným na 3,6V. Logická 0 je řešena přesně opačně se stejnými hodnotami odporů. Zapojení ilustrují následující obrázky 1.10 a 1.11.



Obr. 1.10: Zařízení s „low-speed“ [10]



Obr. 1.11: Zařízení s „full-speed“ [10]

Zařízení, která podporují rychlost přenosu „high-speed“ jsou připojována stejně jako zařízení „full-speed“, jen je pak přenosová rychlost změněna softwarově. Pro zvětšení úspory nemusejí zařízení USB 1.1 podporovat přenosovou rychlost „full-speed“ a stejně tak standart USB 2.0 nemusí podporovat přenosovou rychlost „high-speed“.

Hub dodává napětí na napěťové v rozsahu 4,75V až 5,25V. Maximální pokles napětí z hubu je 0,35V. Dále musí být hub schopen poslat data na napětí 4,4V, ale musí fungovat jen funkce „low-power“.

1.7 Jednotka SIE a součásti

Jednotka SIE (Serial Interface Engine) je jedna z hlavních částí řadiče USB. v jednotce je umístěn přijímač a vysílač, řízení je nejčastěji vytvořeno mikroprocesorem. Sama o sobě ale nemůže fungovat, proto ji doplňují další součásti.

Před jednotkou SIE je USB Transceiver. Ten má za úkol získat z přijímaných dat hodinový signál a také data přijmout a dekodovat. Způsob získání synchronizace byl již popsán výše, k tomu slouží metoda NRZI (Non-Return-To-Zero). Kódování a dekodování signálu se provádí pouze hardwarově. Dalšími prostředky je zajištěno, aby nedocházelo ke ztrátě synchronizace, což by vedlo k chybnému přenosu. Před přenosem se zavádí synchronizační paket, pro USB 1.1 má délku 8 bitů a pro USB 2.0 je to 32 bitů. Přijímač mývá ve většině konstrukcí kmitočet 12 MHz, který je násoben na hodnotu 480 MHz. Na výstupu mohou být sériová nebo paralelní data. Záleží na provedení čipu. Novější čipy zpravidla zvládají oboje v závislosti na konfiguraci, pomalejší a starší jsou uzpůsobeny pouze pro sériový výstup dat.

Dalším blokem je jednotka SIE. v ní je obsažen USB protokol. Jednotka přebírá signál ke zpracování a buď předá informaci dále ke zpracování mikroprocesoru nebo sama odpoví. Jednotka SIE ovládá koncový bod 0 a také provádí enumeraci zařízení. Za jednotkou SIE jsou paměti FIFO, které slouží k výměně dat mezi jednotkou a zbytkem zařízení.

2 ŘADIČE SBĚRNICE USB A VYBRANÝ ŘADIČ CYPRESS CY7C68013A

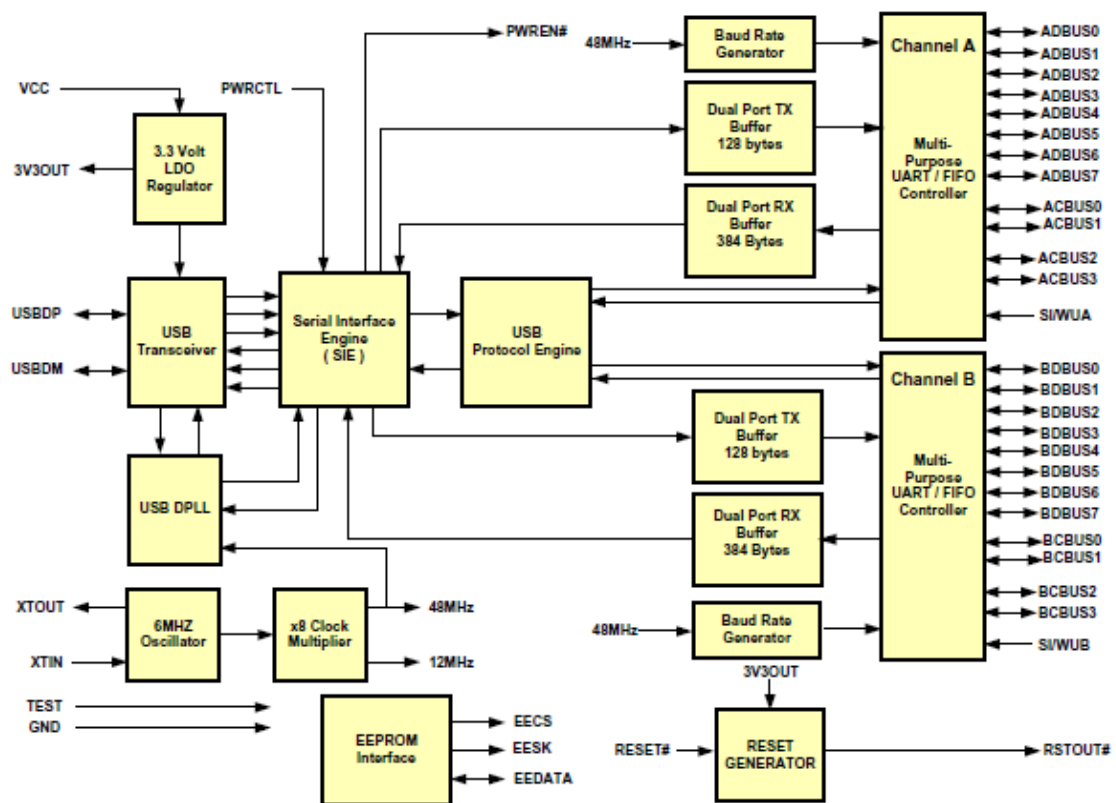
Pro komunikaci zařízení s hostitelem, což je nejčastěji PC, je potřeba převodník, který uspořádá a upraví informace tak, aby odpovídaly správnému tvaru pro přenos. k tomu slouží řadič, který má jednotku SIE, paměti FIFO a dále pak řídicí jednotku, která pracuje podle programu, který je do ní při startu uložen. Mikrořadiče se liší zejména podle verze USB, kterou podporují, přenosovou rychlostí, kterou jsou schopny zpracovat, velikostí FIFO a také jádrem mikroprocesoru. Pro naše účely je nejvhodnější mikrořadič, který bude podporovat verzi USB 2.0 s přenosovou rychlostí „high-speed“ 480Mbit/s a jádrem, které bude lehce programovatelné, jako je například 8051. Pro následující podkapitoly bylo využito informací ze zdrojů [3], [4], [6] a [7].

2.1 Řadiče firmy Texas Instruments

Známa firma Texas Instruments, která se zabývá vývojem a výrobou integrovaných obvodů přispěla na trh hlavně jedním čipem, který by byl vhodný pro použití v této práci. Jedná se o TUSB6015, který podporuje USB 2.0 s rychlostí „high-speed“. Má šest endpointů, které lze použít jako vstupní i výstupní. Čip používá napájecí napětí 3,3V při proudu 20mA. Čip dále obsahuje 16 bitovou flash paměť, která podporuje synchronní i asynchronní čtení a zápis. Pouzdro čipu je ZQE (S-PBGA-N80), takže by bylo nutné čip umístit do příslušné patice. Bohužel jsem se blíže nedopátral přesně typu mikroprocesoru, který je v čipu obsazen a ani informaci o jeho programování. Dále mi chyběla informace o vývodech, kde by mohl být vyveden paralelní port, který je pro tuto práci stěžejní. Ostatní informace o čipu lze dohledat v příslušném datasheetu.

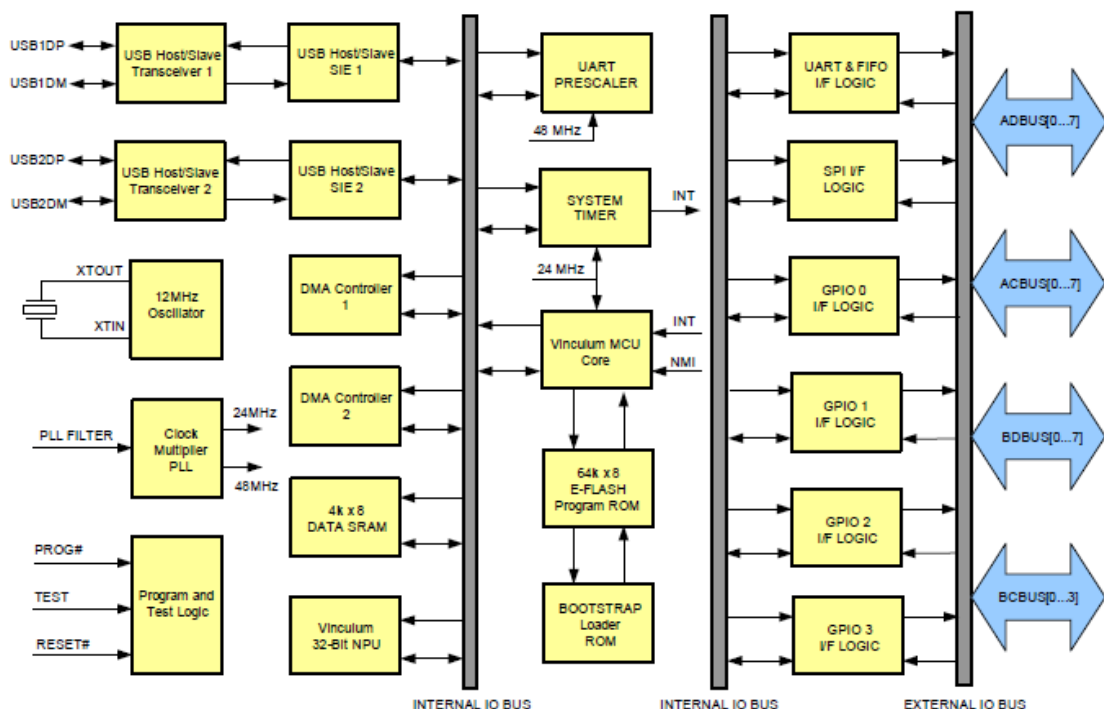
2.2 Řadiče firmy FTDI Chip

Sortiment firmy FTDI Chip je poměrně obsáhlý, hlavně co se týče nabídky v oblasti využití USB. Obsahuje druhou generaci čipů, které vycházejí z první nabídky, ale byly modernizovány, což jsou zejména čipy FT245R, FT2232 a dále dvě řady FT232xx a 245xx. z těchto zmiňovaných by byly pro práci nejvhodnější čipy FT245BL, FT245BM, FT245BQ a FT245R. Jejich napájecí napětí je 3,3V, pracují s maximální frekvencí 48MHz, jsou snadno programovatelné, obsahují vlastní Vendor ID a Product ID, které lze snadno pozměnit a mají již vytvořené ovladače. Podporují USB 2.0 s vysokou přenosovou rychlostí a paměti FIFO dokáží přijímat velké přenášené packety najednou. Čip neobsahuje paměť a proto je nutné uložit veškeré informace na externí paměť. Nevýhoda, která znemožňuje použít čip pro použití v této práci je možný pouze 8 bitový paralelní port. Tento problém lze vyřešit součástkou FT2232, která disponuje duální FIFO a tak lze použít dva 8 bitové paralelní porty. Bloková struktura je na Obr. 2.1. Celý rozsah není potřebný a tak by bylo možno použít jeden a část druhého. Bohužel tento čip disponuje pouze rychlostí „full-speed“ při USB 2.0.



Obr. 2.1: Zjednodušené blokové schéma čipu FT2232D [6]

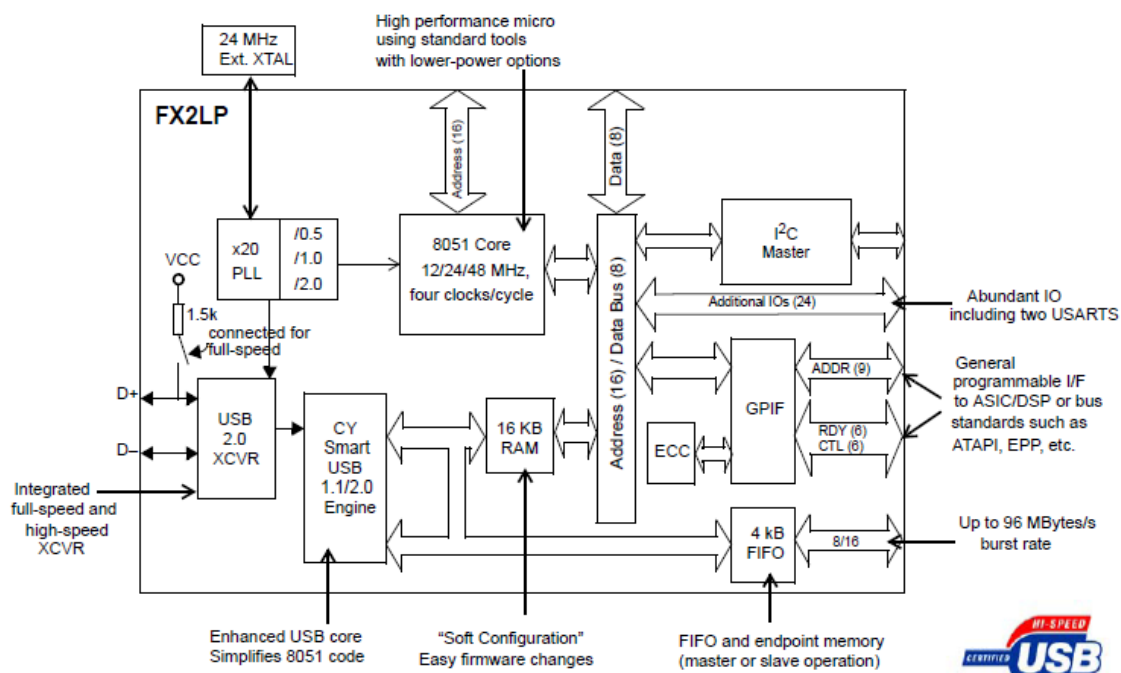
Další možností, kterou tato firma nabízí je použít čip, který je výsledkem vlastního vývoje firmy a to Vinculum, blokové schéma je na Obr. 2.2. Jedná se o součástku, u které byl kladen důraz na maximální možnost využití v různých aplikacích. Není proto potřeba mnoho externích součástek, většina je implementována přímo do čipu. Je zde možná konfigurace zařízení jako master nebo slave, vnitřní paměť flash má kapacitu 64k bajtů a paměť RAM 4k bajtů. Dále obsahuje numerický koprocessor pro 32 bitové operace. 28 pinů je plně konfigurovatelných. Napětí, pro funkci čipu, je 3,3V a odebíraný proud je 25mA při plném provozu, v režimu stand-by jsou to 2mA. Na konec jsou tu dva nezávislé porty USB 2.0, které ale bohužel podporují jen „low-speed“ a „full-speed“. Tím odpadá i použitelnost v této bakalářské práci, i když by to byla výborná alternativa.



Obr. 2.2: Zjednodušené blokové schéma čipu VNC1L [6]

2.3 Řadiče firmy Cypress

Původní řadiče, které byly vyráběny pod označením AN2131 (EZ-USB) vyráběla firma Anchor Chips, kterou později převzala firma Cypress a tím i koncepci těchto řadičů. Nyní jsou firmou Cypress nabízeny řadiče pro nejrůznější použití. Pro tuto práci jsou ale z jejich nabídky hlavní čipy pod označením CY7C68013a (FX2), které mají ve svém jádře upravený procesor, založený na architektuře 8051, který je i určen pro nejrychlejší přenos USB 2.0 a dále transceiver a SIE. Čip má v závislosti na pouzdře (56 vývodů SSOP/QFN, 100 vývodů TQFP, 128 vývodů TQFP) velikost vnitřní paměti RAM 8 nebo 16kB, nemá však žádnou paměť ROM, externí paměť programu a informací o čipu pak může mít stejné velikosti. Řídicí program je tedy uložen v RAM, kam se nahraje při připojení ke sběrnici USB nebo z externí EEPROM hned po zapnutí. Napětí, pro funkci čipu, je 3,3V při maximálním proudu 85mA. Operační kmitočet procesoru může být 48MHz, 24MHz nebo 12MHz. Čip dále podporuje sběrnici I²C, standart USB 2.0 a rychlost „high-speed“. Krom tohoto má dále programovatelné rozhraní GPIF a několik standardních rozhraní jako ATA, UTOPIA, EPP a PCMCIA. Je možné použít vývody pro vytvoření 14 bitového paralelního portu. Tyto parametry plně vyhovují požadavkům, které jsou kladeny na tuto práci. Na rozdíl od standardního procesoru řady 8051 je doba hodinových cyklů zkrácena na čtyři takty. Standardně je dvanáct taktů. Procesor má dvě rozhraní USART, tři časovače a dva data pintery. Posledním parametrem je 4kB paměť FIFO, která je externě přístupná i bez zásahu procesoru a lze ji synchronizovat vnějším hodinovým signálem.



Obr. 2.3: Zjednodušené blokové schéma čipu Cypress CY7C68013 [3]

2.4 Podrobnější popis C7CY68013

V podkapitole 2.3 byly stručně uvedeny parametry, možnosti a vlastnosti vybraného řadiče. Pro lepší seznámení s ním, zde budou popsány funkce jednotlivých částí a vlastnosti.

2.4.1 Obvody řadiče pro komunikaci s okolím

Řadič umožňuje komunikaci s okolím pěti možnostmi. Standardní pro novější mikroprocesory jsou sériové porty, dále řadič obsahuje sběrnici I²C, I/O porty, Slave FIFO a GPIF Master. Obě sériová rozhraní slouží pro spíše pro komunikaci s ostatními zařízeními nebo k řízení. Nedisponují příliš velkou přenosovou rychlostí, takže pro větší přenos dat nejsou příliš vhodná. Zbývající tři možnosti jsou vhodná pro přenos dat většími rychlostmi, protože jsou paralelní. Nevýhodou je, že jsou výstupy umístěny na stejných pinech, proto může pracovat jen jedna z těchto možností.

V součástce CY68013 jsou obsaženy dva sériové porty schopné synchronní i asynchronní komunikace. Při synchronní komunikaci port pracuje v poloduplexním módu a hodinový signál je generován FX2. Při asynchronní komunikaci pracuje port v duplexním módu. Výhodou je dvojitý buffer, kdy je možno přijímat jeden bajt a jiný předchozí zpracovávat. Kmitočet, s nímž se pracuje může být stejná jako frekvence procesoru dělená čtyřmi nebo dvanácti, může být generována čítači 1 a 2 nebo zdrojem může být externí vysokorychlostní generátor.

Sériová sběrnice I²C je zejména použita pro komunikaci čipu s externí pamětí EEPROM, kde jsou obsažena bootovací data. Tato funkce však není jediný, čip může být zapojen stejně, jak standart této sběrnice popisuje. Vygenerovaný hodinový signál je

vyveden na pin SCL, datové propojení se sběrnicí je na pinu SDA. Rychlost sběrnice je možná v rychlostech 100 kHz a 400 kHz.

Řadič obsahuje několik osmibitových obousměrných portů, jejich počet je omezen na pět. Jak je známo z konstrukce 8051 a později i AVR, je každý port ovládán dvěma registry, první OEx udává směr komunikace každého pinu a v registru IOx jsou zapsána data, která jsou určena pro odeslání, pokud je nastavení prvním registru jako výstupní, a nebo jsou v něm uložena příchozí data, pokud je první registr nastaven jako vstupní. Je zde i možnost, že pin plní jinou funkci, protože může být společný pro I/O port i pro jinou komunikaci. v tom případě funkci pinu řídí jiný registr další funkce a obsah registru pro daný pin v I/O je ignorován.

Funkce použití Slave FIFO využívá řízení FIFO pamětí externí logikou, přičemž mikroprocesor obsažený v pouzdře není do přenosu dat zapojen, neovlivňuje tak přenos a může plnit jiné úkoly, pokud jsou naprogramovány. Čtení a zápis dat u pamětí FIFO je řízen externí logikou. Hodinový signál může být vnitřní i vnější v závislosti na požadavcích. Vnitřní signál má kmitočet 30 MHz nebo 48 MHz a může být odebírán i na výstupu jiným zařízením. Pokud je hodinový signál vnější, může být kmitočet v rozmezí 5 MHz až 48 MHz. Datová směrnice může být buď osmi nebo šestnácti bitová.

Rozhraní GPIF (General Programmable Interface) nahrazuje externí řídicí logiku pro paměti FIFO. Je to stavový automat s maximálně sedmi stavy plně určujících výstupní formát z řadiče. Jádro rozhraní generuje šest ovládacích a devět adresních signálů a je schopno přijmout šest vnějších a čtyři vnitřní signály ready. GPIF v závislosti na vygenerovaných signálech obstará přenos dat z pamětí FIFO na výstupní piny. Principiálně je funkce tohoto přenosu dat stejná, jako v předchozím případě u Slave FIFO. Rozdíl je jen v tom, že řízení ovládá vnitřní jádro GPIF a ne externí procesor. Výstupní datová sběrnice i vnitřní a vnější hodinový signál mají stejné parametry a vlastnosti. Hodinové signály je možno softwarově invertovat v případě potřeby.

2.4.2 Stav šetření energie a reset

Při požadavku sběrnice USB, aby čip a celé zařízení přešlo do šetrného módu (suspend) se nejprve provedou kroky, kdy firmware vypne všechny vnější spotřebiče poté se vypne i oscilátor. Do normálního stavu může zařízení přejít, pokud je nějaká aktivita na sběrnicí USB nebo je aktivován jeden ze dvou wakeup pinů.

V čipu FX2 jsou dva druhy resetu, kdy jeden je ovládán pinem \overline{RESET} a po jeho spuštění se vše uvede do předem daného stavu a to, že buffery koncových bodů jsou nezměněny, pracovní kmitočet procesoru se nastaví na hodnotu 12 MHz a aktivuje se pin CLKOUT, piny všech portů se nastaví jako aktivní, jsou vypnuty požadavky na přerušení ze sběrnice USB a koncové body jsou nastaveny jako Default USB Device, to je Power-On Reset. Dalším je CPU Reset. Ten je ovládán jádrem a spustí se vždy při Power-On Resetu. Zde dojde jen k resetování resetování běhu programu. Sběrnice USB je sama schopná resetovat všechna připojená USB zařízení, což lze nalézt jako specifikaci USB Bus Reset.

2.4.3 Funkce mikroprocesoru a paměti

Upravený mikroprocesor je na bázi 8051 s rozdílem, že má čtyřnásobnou rychlost, takt až 48 MHz, menší počet hodinových cyklů, druhý data pointer, dva autopiontery (tj. schopnost automatické inkrementace), druhý USART, třetí šestnácti bitový časovač, vysokorychlostní rozhraní pro připojení vnější paměti, které má šestnáctibitovou nemultiplexovanou adresní sběrnici, řadič I²C atd. Velkou výhodou je struktura 8051, protože pro funkci procesoru je možno použít standardní programy, u kterých by jediná změna spočívala v přepočítání časových konstant. Tímto je možné použít i standardní assembly a kompilaci.

Interní paměť Data RAM má velikost 128 bajtů a adresy 0x00 až 0x7F a SFR na adresách 0x80 až 0xFF. Dále je zde nepřímě adresovatelná paměť interní paměť Data RAM, který má také 128 bajtů. Zbytek paměti je adresován jako vnější.

Zobrazení struktury registru je na Obr. 2.4. Jak je vidět, nejnižší je paměť RAM 8051. Ovládání periférií je na adresách 0xE400 až 0xFFFF. Registry a buffery jsou sloučeny do skupin (GPIF deskriptory, konfigurace koncových bodů, apod.).

FFFF	4 KBytes EP2-EP8 buffers (8 x 512)
F000	2 KBytes RESERVED
EFFF	
E800	64 Bytes EP1IN
E7FF	
E7C0	64 Bytes EP1OUT
E7BF	
E780	64 Bytes EP0 IN/OUT
E77F	
E740	64 Bytes RESERVED
E73F	
E700	8051 Addressable Registers (512)
E6FF	
E500	Reserved (128)
E4FF	
E480	128 bytes GPIF Waveforms
E47F	
E400	Reserved (512)
E3FF	
E200	512 bytes 8051 xdata RAM
E1FF	
E000	

Obr. 2.4: Uspořádání vnitřního registru čipu FX2 [3]

Registr SFR je tvořen tak, aby byl co nejpodobnější registru 8051. Rozpis jeho funkcí je možno najít v příslušné datasheetu.

2.4.4 Enumerace a Renumerace

Proces enumeraci již byl popsán v kapitole o USB. Protože čip FX2 nemá žádnou paměť ROM, jsou všechna data (řídící program) uložena v paměti RAM a musí se vždy na počátku načíst. z tohoto důvodu je nutná renumerace. Jednotka SIE na počátku zařízení přihlásí v základní konfiguraci a pak zajistí načtení programu do RAM procesoru. Po této proceduře se musí zařízení odhlásit a poté znovu přihlásit už s novými údaji, tj. provést znova enumeraci dle konfigurace, která byla přijata. To je proces renumerace a je vhodná třeba pro lazení chodu ovládacího programu. Připojení a odpojení zařízení probíhá bez fyzického přerušení trasy.

Pro start zařízení jsou čtyři možnosti. První z nich je, že externí paměť není připojena, takže FX2 se enumeruje jako Default USB Device s hodnotami deskriptoru (PID, VID, DID), které jsou dány od výroby. Když se načtou data, provede se renumerace. Druhou možností je připojená paměť EEPROM, která obsahuje deskriptory. Tak se provede stejný proces jako v předchozím případě, jen budou jiné zaváděcí hodnoty a to ty, které jsou uloženy. v prvním bajtu musí být hodnota 0xC0. Třetí možnost je taková, že je připojena paměť EEPROM, ve které je uložen firmware. Ten se pak automaticky nahraje do vnitřní paměti RAM a další postup se provádí podle firmware. v prvním bajtu musí být hodnota 0xC2. Poslední možností je, že externí paměť není EEPROM, ale jiný druh. Princip je prakticky stejný jako v předchozím případě.

Při zavádění dat z paměti EEPROM jsou možné tři způsoby, které již byly naznačeny výše a to že v paměti buď nejsou data, v tom případě se zařízení přihlásí s VID 0x04B4 (Cypress Semiconductor) a PID 0x8613 (EZ-USB FX2), nebo jsou jen uložena čísla PID a DID, viz. Tab. 2.1.

Tab. 2.1: Obsah EEPROM pro enumeraci [8]

Adresa v EEPROM	Obsah
0	0xC0
1	Vendor ID L
2	Vendor ID H
3	Product ID L
4	Product ID H
5	Device ID L
6	Device ID H
7	Konfigurační bajt

Posledním případem je možnost, kdy je v paměti uložen firmware, počátek je stejný s předchozím, dále pak navazují další bloky. Viz. Tab. 2.2.

Tab. 2.2: Obsah paměti při načtení firmwaru [8]

Adresa v EEPROM	Obsah
0	0xC2
1	Vendor ID L
2	Vendor ID H
3	Product ID L
4	Product ID H
5	Device ID L
6	Device ID H
7	Konfigurační bajt
8	Délka H
9	Délka L
10	Počáteční adresa H
11	Počáteční adresa L
	Data
	Délka H
	Délka L
	Počáteční adresa H
	Počáteční adresa L
	Data
	0x80
	0x01
	0xE6
	0x00
poslední	0

V Tab. 2.3 je uveden konfigurační bajt, kde DISCON je bit, který se používá při renumeraci. Při log. 1, se provede chvilkové odpojení a opětovné přihlášení, pak se bit vynuluje a lze ho kdykoliv měnit. Bit b0 nastavuje rychlost sběrnice I²C v log. 0 pracuje sběrnice s taktem 100 kHz, jinak 400 kHz.

Tab. 2.3: Konfigurační bajt [8]

b7	b6	b5	b4	b3	b2	b1	b0
0	DISCON	0	0	0	0	0	400 kHz

2.5 Řadiče ostatních firem

Na trhu s řadiči pro USB působí mnoho firem s nejrůznějším sortimentem pro ilustraci je možno jmenovat zejména National Semiconductor, MAXIM, AMD, Atmel, Intel, Infineon Technologies, Kawasaki LSI, Microchip Technology, Mitsubishi

Semiconductor, Motorola, Philips Semiconducotr, ScanLogic a také STMicroelectronic. v jejich nabídkách se jedná zejména o čipy, které jsou navrženy pro již jeden účel a to pro propojení USB s ATA, videokamerami, MP3 přehrávači nebo zařízeními typu HID. Nebo čipy, které slouží pro použití v rozbočovačích USB. Tento sortiment můžeme nalézt i u firem, které byly zmiňovány v předchozích odstavcích. Za zmínku ještě stojí řešení firmy Atmel, kde čip AT43USB320A je sjednocením běžných mikrokontrolérů jak je známe a rozhraním USB. Našel by použití zejména v aplikacích, které by byly určeny pro rozsáhlé řízení dalších zařízení.

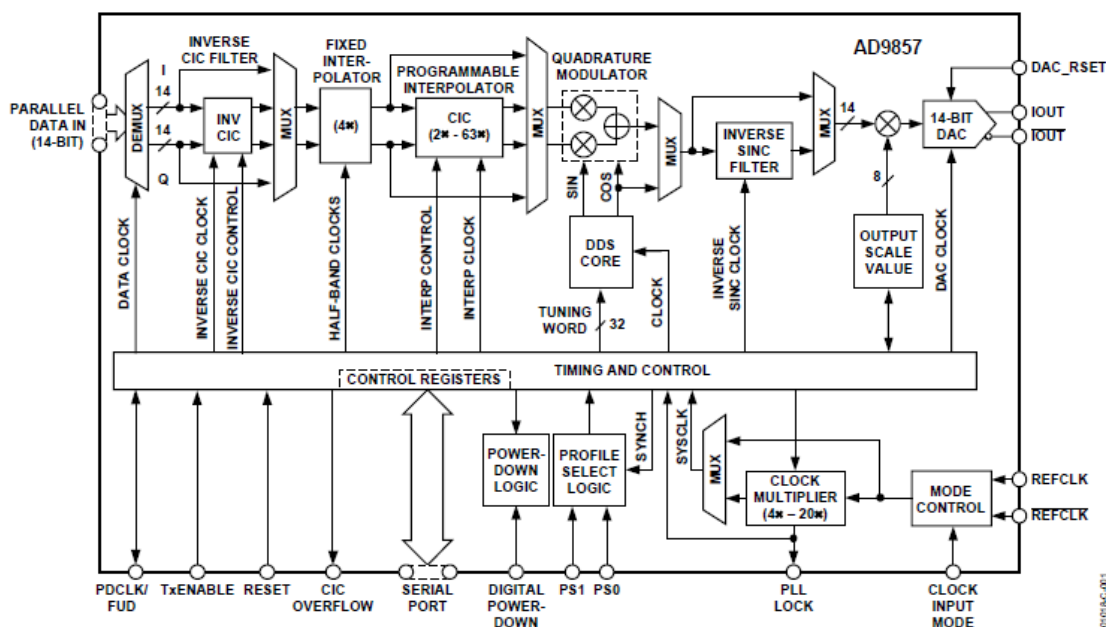
3 KVADRATURNÍ MODULÁTOR AD9857

V této kapitole je stručný popis kvadraturního modulátoru AD9857. Informace byly čerpány z [5] a [8].

3.1 Základní vlastnosti modulátoru

Modulátor AD9857 vyrábí firma Analog Device. v jeho struktuře je integrován vysokorychlostní přímý digitální syntezátor (DDS), čtrnácti bitový digitálně-analogový převodník (DAC), násobiče frekvence a digitální filtry. Má několik možných funkcí a tak je schopen pracovat jako jednotonový digitální syntezátor, univerzální modulátor I/Q a jako digitálně-analogový převodník pro aplikace v telekomunikační technice. Vyniká hlavně svou výkonností, úsporou energie, velikostí za poměrně příznivou cenu.

Vnitřní frekvence po vynásobení, tedy za pomoci fázového závěsu může být až 200 MHz. Vstup modulátoru je paralelní a přivádí se na něj čtrnácti bitová slova ve formátu dvojkového doplňku. Je umožněno ovládat výstupní amplitudu signálu v rozsahu osmi bitů s podporou korekce $\sin(x)/x$. Nastavení je umožněno pomocí sériového rozhraní s maximálním kmitočtem 10 MHz. Tak lze nastavit jeden ze tří módů funkce modulátoru, který již byly jmenovány. Kromě nich lze i nastavit funkce digitálních filtrů a jiných součástí modulátoru. Na Obr. 3.1 je uvedeno zjednodušené blokové schéma.



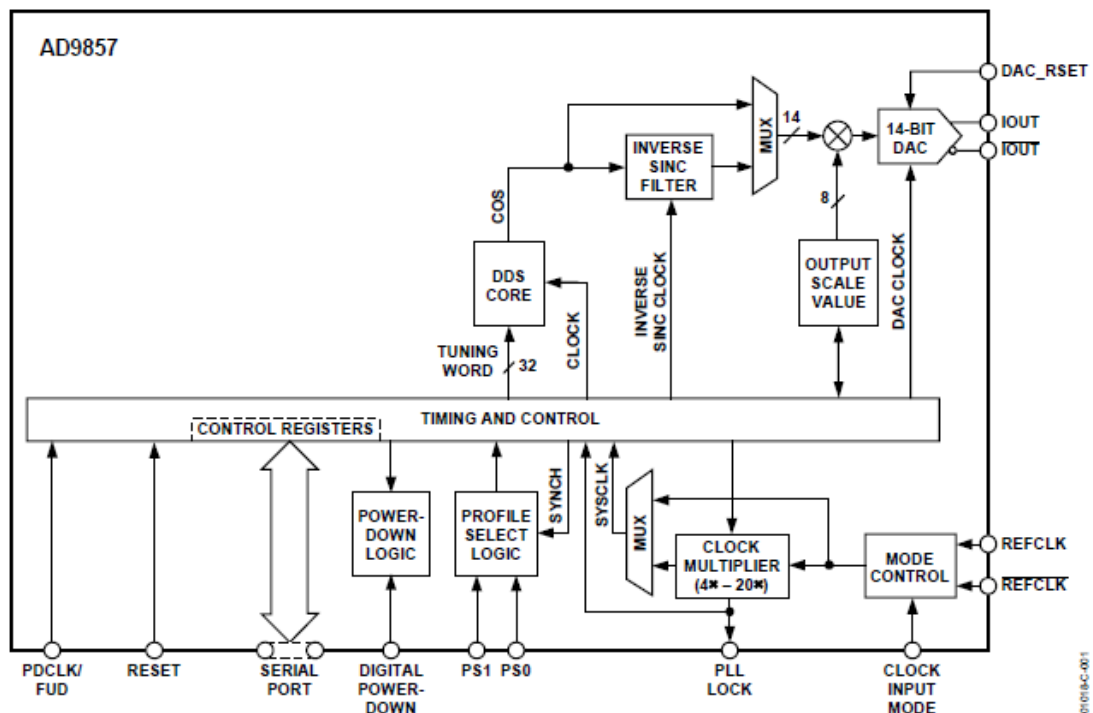
Obr. 3.1: Blokové schéma modulátoru AD9857 [5]

3.2 M3d kvadrurní modulace

Blokové schéma je stejné jako na Obr. 3.1. Cesty I a Q jsou obě aktivní pro přenos dat. Přenos probíhá tak, že vstupní data jsou synchronizována signálem náběžnou hranou signálu PDCLK, který má dvojnásobný kmitočet, než cesty I/Q, proto jsou data na vstupu reprezentována jako dvě čtrnácti bitová slova sdružená v čase. Obě slova pak spolu tvoří jeden vnitřní vzorek. Tím, že se vynásobí data z kanálů I/Q se signálem vytvořeným v jádře DDS, které vytvoří signály v kvadratuře, a pak je sečte se vytvoří kvadrurně modulovaný datový signál. Tento digitální signál je pak následně převeden do digitálně-analogového převodníku, kde vznikne kvadrurně modulovaný analogový signál.

3.3 Jednotónový m3d

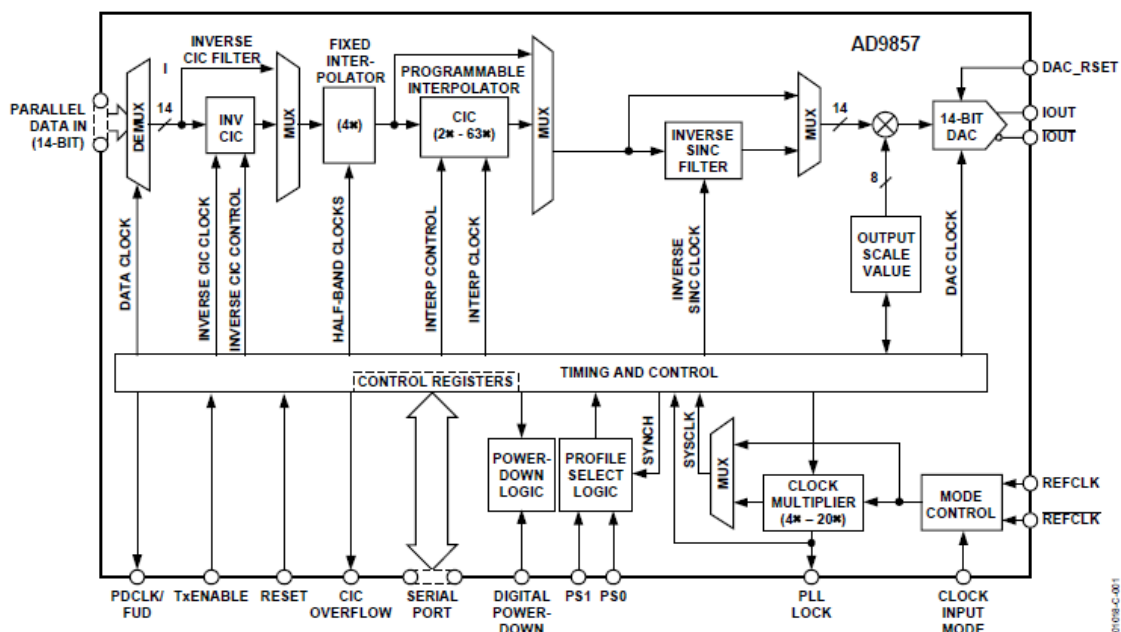
V tomto módu nejsou kanály I/Q aktivní. Dochází zde ke změně vlastnosti signálu PDCLK/FUD. Signál nyní plní funkci ovládání zápisu FTW(Frequency Tuning Word) do aktivního profilu, protože FTW je programováno přes asynchronní sériový port. Jádro DDS generuje signál cosinus o frekvenci FTW. Ten je pak upraven inverzním SINC filtrem a je nastavena jeho amplituda. Nakonec je signál převeden na analogový. Princip je patrný i z blokového schématu na Obr. 3.2.



Obr. 3.2: Blokové schéma jednotónového módu [5]

3.4 Interpolační digitálně-analogový mód

V tomto režimu nepracuje jádro DDS ani kvadrurní modulátor. Funkční je pouze kanál I. Signál PDCLK má tentokrát stejný kmitočet jako kanál I, protože je kanál Q vypnut. Modulace se neprovádí, vstupní signál je pouze převzorkován s vyšší rychlostí. Blokové schéma je na Obr. 3.3.



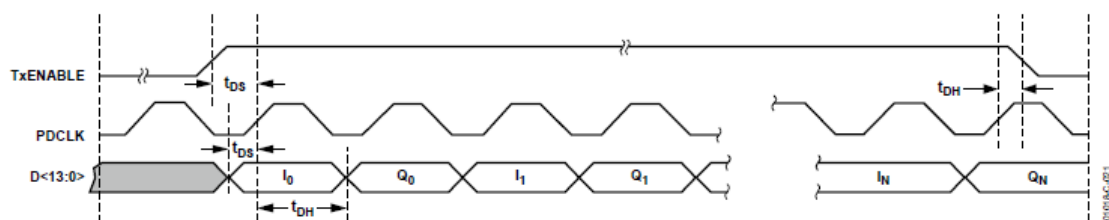
Obr. 3.3: Blokové schéma interpolačního digitálně-analogového módu AD9857 [5]

3.5 Funkce dalších bloků AD9857

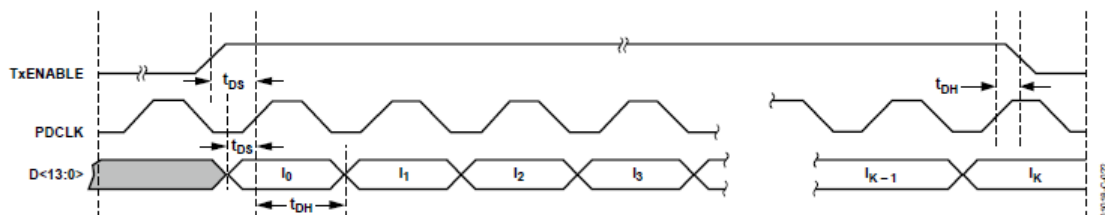
3.5.1 Časování vstupních dat

Časování vnitřních částí převodníku probíhá pomocí vnitřního hodinového signálu SYSCLK. Pokud se použije externí signál, může být použit přímo nebo vynásoben konstantou (4 až 20) a potom použit. Zbylé signály se odvozují od tohoto základního.

Signál PDCLK je stále tvořen aktivní a chvilé, kdy se mají přijímat data určuje TxEnable. Pokud je TxEnable na hodnotě log. 0, tak nejsou data ze vstupu přijímána. Při náběžné hraně signálu TxEnable se přijmou data do kanálu I. Musí být tedy zajištěno, aby na vstup modulátoru přišel sudý počet čtrnácti bitových slov. v závislosti na použitém módu jsou na Obr. 3.4 a 3.5 zobrazeny časové průběhy příjmu signálu dvou z nich, které nejlépe vypovídají o funkci.



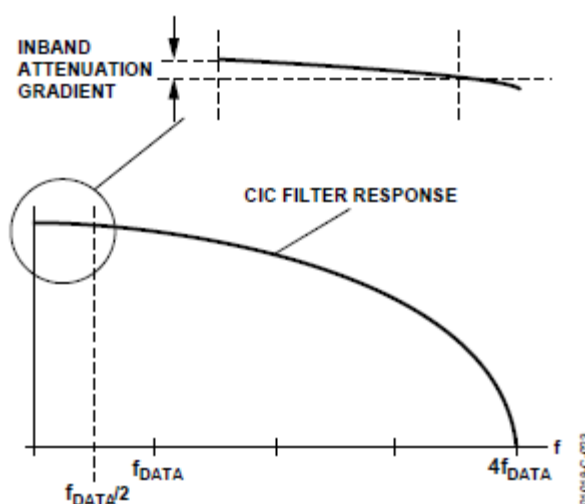
Obr. 3.4: Časování paralelního portu v módu kvadrurní modulace [5]



Obr. 3.5: Časování paralelního portu v interpolační DAC módu [5]

3.5.2 CIC filtry a fixní interpolátor 4x

Filtr CIC je inverzní kaskádový hřebenový filtr, který překompenzuje vliv programovatelného interpolačního CIC filtru, ten je zařazen dále v systému. Jeho účel je takový, aby harmonické složky na vyšších kmitočtech nebyl více utlumovány než ty na nižších. Zařazení filtru a inverzního filtru kompenzuje vliv na přenášený signál. Je tvořen jako filtr FIR s přenosovou charakteristikou, která je inverzní k použitému CIC interpolátoru. Na Obr. 3.6 je charakteristika filtru.



Obr. 3.6: Charakteristika CIC filtru [5]

V AD9857 je programovatelný interpolátor vytvořen jako CIC filtr. Je možné ho nastavit pomocí šestibitového slova, tím je možná šířka interpolace $2 \div 63x$. Vstupní signál může být převzorkován $8 \div 252x$. frekvenční charakteristika se podobá dolní

propusti a je kompenzována inverzním CIC filtrem. Oba filtry je možno obejít tím, že se interpolací nastaví na 1x.

Fixní interpolátor je tvořen dvěma filtry s poloviční šířkou pásma. Data jsou čtyřnásobně převzorkována, proto 4x. Tato vlastnost je pevně dána a nelze ji nijak ovlivnit. k fázovému zkreslení nedochází, protože filtry mají lineární přenosovou charakteristiku.

3.5.3 Kvadrurní modulátor

Modulátor se používá pro přesun signálu na jiný kmitočet (nosnou), než je jeho v základním pásmu, zpravidla to bývá vyšší kmitočet. Předchozí bloky převzorkují data ze vstupní vzorkovací frekvence na vzorkovací kmitočet, který se rovná nosné. Ta je nastavena číselně v DDS, které vytvoří nosnou z odvození z vnitřního hodinového kmitočtu. Nosný kmitočet je kvadrurně násoben daty z kanálu I/Q a poté sečtena. Toto je řízeno spektrálním inverujícím bitem a jsou možné dvě funkce v závislosti na log. úrovni a to $Ix \cos(\varpi) \mp Qx \sin(\varpi)$.

3.5.4 Jádno DDS

Digitální syntezátor generuje nosné sin a cos s kmitočtem, který je dán FTW. Kmitočet je dán vztahem

$$f_{out} = \frac{(FTW \cdot SYSCLK)}{2^{32}}, \quad (1)$$

přičemž f_{OUT} a $SYSCLK$ jsou kmitočty v Hz a FTW je dekadické číslo $0 \div 2147483647$.

3.5.5 Inverzní SINC filtr

Signál z DAC má tvar funkce $\text{sinc}(x)$, proto se před DAC zapojuje inverzní SINC filtr, který toto zkreslení odstraňuje. Filtr je typu FIR s přenosovou charakteristikou, která je inverzní k SINC. Korekce je ale omezená pro nosné, které dosahují 45% $SYSCLK$. Je zde možnost filtr vypnout a tím i uspořit část odebírané energie.

3.5.6 Zbylé bloky

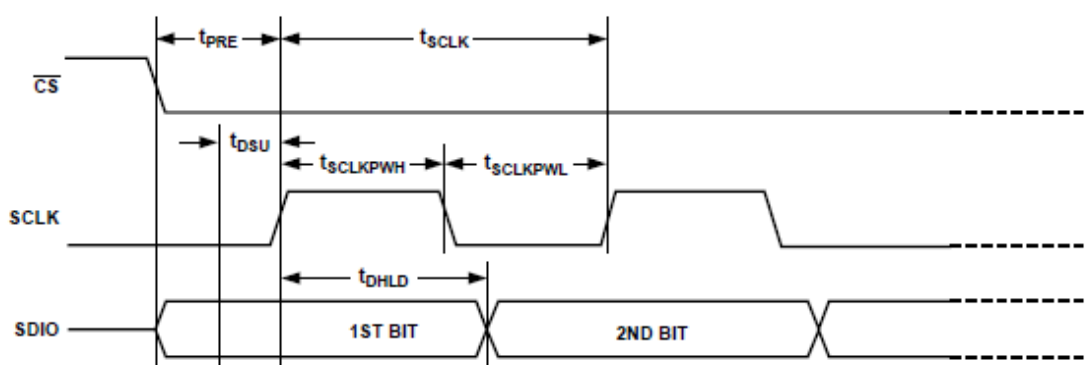
Mezi další části AD9857 patří výstupní osmi bitový násobič amplitudy, který umožňuje programově stanovit amplitudu výstupního signálu. Nejnižší bit má váhu 2^{-7} , takže výstupní koeficient může nabývat hodnot $0 \div 1,9921875$.

Dalším blokem je čtrnácti bitový digitálně-analogový převodník, pro převod signálu na výstupní analogový.

Násobič referenčního kmitočtu je postaven na principu fázového závěsu a tak je možno vytvořit přesný signál v rozmezí od 100 MHz až 200 MHz, i když je připojen krystal a malém kmitočtu.

3.6 Programování a nastavování

Čip je možno připojit k PC pomocí flexibilního sériového rozhraní. Tak je možno zapsat do všech konfiguračních registrů požadované hodnoty, které nastaví požadovanou funkci AD9857. Přenos je možno nakonfigurovat po jednom pinu nebo po dvou jednosměrných. Při komunikaci je nejprve zapsán instrukční bajt a hned poté v další fázi jsou vykonány instrukce uvedené v instrukčním bajtu. v instrukčním bajtu je uvedeno, zda se data budou číst nebo zapisovat v datovém přenosu. Dále počáteční adresa registru pro první bajt přenosu. Při přenosu je prvními osmi náběžnými hranami hodinového signálu komunikačního kanálu určeno zapisování instrukčního bajtu, poté už se náběžnými hranami určuje přenos dat. Po instrukčním bajtu se přenáší jeden až čtyři bajty, podle toho, kolik jich určil instrukční bajt. Po poslání těchto dat se opět vysílá instrukční bajt. Všechny vstupy reagují na náběžné hrany, kdežto všechny výstupy při sestupné, jak je vidět na Obr. 3.7.



Obr. 3.7: Časový diagram pro zápis dat do AD9857 [5]

3.6.1 Instrukční bajt

Tab. 3.1: Instrukční bajt [8]

MSB	D6	D5	D4	D3	D2	D1	LSB
R/W#	N1	N0	A4	A3	A2	A1	A0

V Tab. 3.1 je uveden instrukční registr, kde sedmý bit R/W# slouží pro určení zápisu (log. 0) nebo čtení (log. 1). N1 a N0 určují počet bajtů, které budou následovat při přesouvání dat. Jejich kombinace je stejná jako počet 1 až 4 v binární kombinaci. Zbylé bity určují registr, který bude ovlivněn.

3.6.2 Ovládací registry

AD9857 obsahuje několik registrů, které by šly rozdělit do dvou částí a to registry, pomocí kterých se nastaví vlastnosti čipu a pak registry, do kterých se uloží konfigurace čtyř volitelných profilů. v každém profilu je možné si zvolit zapnutí a vypnutí inverzního CIC filtru, koeficienty pro interpolační CIC filtr, velikost výstupní

amplitudy, FTW a také nastavení spektrálního invertujícího bitu. Bližší specifikace je v datasheetu. Pro lepší přehled je zde několik registrů, pomocí kterých se dají nastavit vlastnosti čipu.

- Reference Clock Multiplier – pětibitové číslo, pomocí kterého lze nastavit hodnotu, kterou se bude násobit kmitočet vstupního signálu pro vytvoření interního hodinového signálu.

- PLL Lock Control – dle nastavení vnitřní logika buď ignoruje nebo využívá k vnitřnímu řízení operací podél čtrnácti bitové paralelní datové cesty podle stavu pinu PLL Lock

- LSB First – udává, která data budou akceptována jako první, buď LSB nebo MSB

- SDIO Input Only – sériový port používá pro vstup a výstup dat buď dva jednosměrné piny nebo jeden obousměrný

- Operating Mode – dle nastavení určuje v jakém módu bude AD9857 pracovat, zda kvadraturní modulátor, jednotunový mód nebo interpolační digitálně-analogový převodník

- Auto Power Down – čip přechází automaticky do úsporného režimu, pokud nejsou indikována data nebo manuálně při aktivování pinu digital power down

- Full Slep Mode – nastaví kompletní vypnutí zařízení v nastavené log. 1

- Inverse SINC Bypass – při log. 1 je vypnut inverzní SINC filtr

- CIC Clar – při log. 1 dojde k vymazání CIC filtrů, když ne, pracují normálně

3.6.3 Programovací software

Pro nastavování hodnot registrů pro požadovanou uživatelskou funkci je vhodné použít programovací software, který je určen pro čip AD8957. Software je hlavně určen pro kit od firmy Analog Devices, na kterém je čip umístěn. Lze tak snadno nastavit veškeré požadované funkce, které jsou obsaženy a byly v textu již popsány. Jedná se hlavně o mód a také o nastavení registrů. Nevýhodou ale je, že toto nelze provádět za provozu a kit by se musel přepojovat k PC, aby AD9857 mohlo být nastaveno a pak opět zpět. Proto je vhodné vytvořit na PC software, který by ovládal požadovanou funkci a také by uměl nastavit funkci AD9857 ale aktuálních požadavků. Toto je více popsáno v kapitole o Softwaru. Pro ilustraci jsou na Obr. 3.8, 3.9 a 3.10 zobrazeny možnosti programu pro nastavení.

Profile Setup - Profile 1 Active

☒ **Profile 1 Active**

Frequency Tuning Word
 Output Freq: 0.00000 MHz Hex 00000000 Load Profile
 Binary 00000000 00000000 00000000 00000000 FUD

Output Scale Factor
 OSF Value: 1.4140625
 Dec 32 Hex 20
 Binary 10110101 Hex B5 Binary 100000

CIC Interpolation Rate
 Dec 32 Hex 20
 Binary 100000

☐ Spectral Inversion ☐ Inverse CIC Bypass

☐ Profile 2

Frequency Tuning Word
 Output Freq: 0.00000 MHz Hex 00000000 Load Profile
 Binary 00000000 00000000 00000000 00000000 FUD

Output Scale Factor
 Dec 32 Hex B5
 Binary 10110101

CIC Interpolation Rate
 Dec 32 Hex 20
 Binary 100000

☐ Spectral Inversion ☐ Inverse CIC Bypass

☐ Profile 3

Frequency Tuning Word
 Output Freq: 0.00000 MHz Hex 00000000 Load Profile
 Binary 00000000 00000000 00000000 00000000 FUD

Output Scale Factor
 Dec 1.4140625 Hex B5
 Binary 10110101

CIC Interpolation Rate
 Dec 32 Hex 20
 Binary 100000

☐ Spectral Inversion ☐ Inverse CIC Bypass

☐ Profile 4

Frequency Tuning Word
 Output Freq: 0.00000 MHz Hex 00000000 Load Profile
 Binary 00000000 00000000 00000000 00000000 FUD

Output Scale Factor
 Dec 1.4140625 Hex B5
 Binary 10110101

CIC Interpolation Rate
 Dec 32 Hex 20
 Binary 100000

☐ Spectral Inversion ☐ Inverse CIC Bypass

Obr. 3.8: Výběr a nastavení jednoho ze čtyř profilů

Read Back

☐ Profile 1

☐ Register 02h 00000000
☐ Register 03h 00000000
☐ Register 04h 00000000
☐ Register 05h 00000000
☐ Register 06h 00000000
☐ Register 07h 00000000

☐ Profile 2

☐ Register 08h 00000000
☐ Register 09h 00000000
☐ Register 0Ah 00000000
☐ Register 0Bh 00000000
☐ Register 0Ch 00000000
☐ Register 0Dh 00000000

☐ Profile 3

☐ Register 0Eh 00000000
☐ Register 0Fh 00000000
☐ Register 10h 00000000
☐ Register 11h 00000000
☐ Register 12h 00000000
☐ Register 13h 00000000

☐ Profile 4

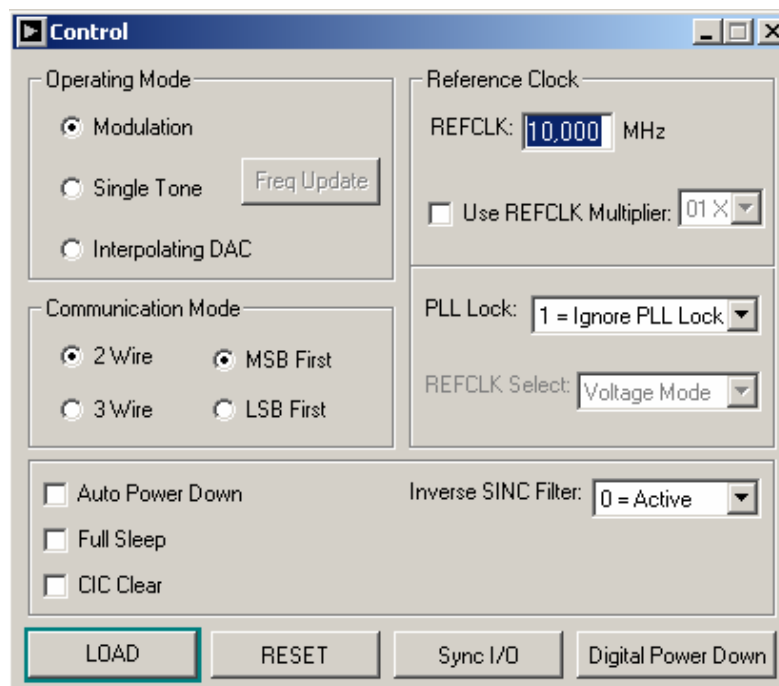
☐ Register 14h 00000000
☐ Register 15h 00000000
☐ Register 16h 00000000
☐ Register 17h 00000000
☐ Register 18h 00000000
☐ Register 19h 00000000

☐ Control Registers

☐ Register 00h 00000000
☐ Register 01h 00000000

Read Back Clear Form

Obr. 3.9: Nastavení registrů v daném profilu a kontrolních registrů



Obr. 3.10: Nastavení filtrů, komunikace, vnitřního kmitočtu a módu AD9857

3.7 Power Management

Pokud nedochází k žádnému přenosu dat, je možné snížit spotřebu výkonu na 60 až 70%. To je možno kontrolovat softwarově i hardwarově. Pro zotavování není potřebná žádná doba, AD9857 se přepne okamžitě do plné funkce v případě potřeby. Úspora spočívá ve vypnutí všech vnitřních hodinových signálů. Pokud je tento stav určen hardwarově, na pin DPD se přivede log.1 a restartováním se provede přepsání úrovně na log. 0. Softwarově kontrolovaná metoda využívá v zapnutí nebo vypnutí automatického úsporného módu. Tento mód se zapíná ve chvíli, kdy dojdou všechny zpracovávaná data na výstup. Poslední možností jak ušetřit co nejvíce energie je tzv. Full Sleep. Tento mód vypíná jak digitální, tak i analogovou část modulátoru. Jeho ovládání je určeno hodnotou v registru. Po použití tohoto módu není AD9857 schopen opět ihned pracovat jako v předchozích případech, ale je nutno počkat na stav indikátoru PLL Lock, který ukazuje, že všechny vnitřní kmitočty byly ustáleny.

4 KONTRUKCE PŘEVODNÍKU

Pro konstrukci se jevil jako nejlepší možnost čip CY68013 FX2 v pouzdře se 100 vývody. Jelikož na konstrukci s tímto čipem již pracoval Ing. Igor Majko, bylo vhodné v jeho práci pokračovat a doplnit nedostatky, které již nestihl dokončit. Tato kapitola se bude zabývat jeho prací a dále nastíní směr, kterým se bude ubírat zbytek práce. Konstrukce má nevýhodu, že je vytvořena na dvou vrstevném oboustranném plošném spoji. Výrobce pak nezaručuje maximální rychlost „high-speed“. Pro následující podkapitoly bylo využito informací ze zdrojů [3], [4], [5] a [8].

4.1 Výchozí požadavky na konstrukci

Ze specifikace kvadrurního modulátoru AD9857 vyplývá, že je potřebná 16-bitová sběrnice s možností synchronizace vnějším hodinovým signálem. Frekvence, které může signál dosáhnout při $N=1$, je 100MHz. Pokud budou při této frekvenci vstupní slova 14-bitová, datový tok bude mít velikost 1,4Gb/s. Přitom N udává interpolační koeficient, který může nabývat hodnot od 1 do 63. z důvodu potřeby extrémně velkého datového toku je omezena možnost použití nízkých hodnot N . Dále je dle specifikace nutný sériový port, který je kompatibilní s portem modulátoru. Výhodou je, že pomocí sběrnice USB je možné posílat jak pracovní data modulátoru, tak sériová data k nastavení konfigurace a to pouhým přesměrováním dat do jiného end pointu softwarově, podle toho, který end point bude patřit k paralelnímu nebo sériovému portu.

4.2 Samotná hardwarová konstrukce

Zde je stručný popis hardwarové konstrukce převodníku. Deska plošného spoje je oboustranná, prokovená s nepájivou maskou, potiskem a pozlacenými kontaktními ploškami z důvodu nižší výrobní ceny (výrobce čipu doporučuje čtyř vrstvou desku plošného spoje). Návrh měl jemné nedostatky, a tak byl návrh převodníku upraven. Jedná se zejména o osazený konektor typu A, který by neměl být na zařízení slave, ten byl vyměněn za zásuvku typu B. Pozměněný plošný spoj, osazení a schéma jsou uvedeny v příloze. Vše bylo měněno v návrhovém systému Eagle 5.4.

4.2.1 Napájení

Samotný čip CY68013 potřebuje pro svou činnost napětí 3,3V. Lze tedy využít napětí ze sběrnice USB, kde je 5V, a pomocí stabilizátoru ho snížit na 3,3V. v konstrukci je použit SMD stabilizátor v pouzdru SOT223 od firmy Semiconductor pod označením MC33275ST3.3. Maximální výstupní proud tohoto stabilizátoru je 300mA, což pro funkci čipu plně postačuje, protože jeho samotný maximální proud je o něco menší. Při prvním připojení je zařízení USB přidělen proud 100mA, který nebude pro funkci čipu postačovat, bude tedy nutné do softwaru zanést požadavek na navýšení proudu v průběhu enumerace. Maximální proud sběrnice je 500mA, není tedy nutný externí zdroj a zařízení si vystačí pouze se sběrnici USB. Napětí ze sběrnice je filtrováno

keramickým kondensátorem SMD. Napětí 5V na sběrnici je pak indikováno červenou LED a stabilizované napětí 3,3V je indikováno zelenou LED. Stínění USB a RS232 jsou propojena a přes paralelní spojení kondensátoru a rezistoru připojena na zem. Každý napájecí pin čipu je filtrován samostatným kondensátorem s kapacitou 100nF.

4.2.2 Paralelní rozhraní

Paralelní rozhraní pro posílání dat je vyvedeno na konektoru SV2. Konektor je realizován pomocí lišty a lze na něj přímo připojit plochý kabel. z toho ale plyne omezení, že je vyvedeno pouze 14 prvních bitů datové sběrnice. Proto jsou na konektoru JP1 vyvedeny dva zbylé piny. Tím lze získat celý rozsah 16-bitové sběrnice.

Pro posílání dat do modulátoru je využito rozhraní GPIF přímo od firmy Cypress. Použitím tohoto rozhraní je umožněno posílat data z FIFO přímo na výstup. Není tedy nutné, aby data byla zpracována mikroprocesorem a tím se přenos značně urychlí. z rozhraní je použito prvních 14 bitů datové sběrnice pro přenos dat na piny FD0 až FD13, dále kontrolní signál CTL0, který aktivuje zápis dat a další dva signály CTL1 a CTL2, které slouží pro výběr aktivního profilu modulátoru . Jako poslední je hodinový signál z modulátoru PDCLK, který je nutné v FX2 invertovat softwarově pro správnou funkci, přiveden na pin IFCLK.

4.2.3 Sériové rozhraní

Sériové rozhraní je vyvedeno na konektor SV2, stejně jako paralelní rozhraní v předchozí podkapitole. Pro komunikaci modulátoru a FX2 byl zvolen přenos přes dva jednosměrné piny, protože tuto možnost jako jedinou FX2 podporuje. Je nutné se přizpůsobit maximální frekvenci portu modulátoru, která je 10MHz. Hodinový signál FX2, jehož kmitočet je odvozen od vnitřní frekvence procesoru, se určuje dělením čtyřmi nebo dvanácti. Pokud bude FX2 pracovat na nejvyšším kmitočtu 48MHz, bude nutné použít dělení 12 a výsledný kmitočet pak bude mít hodnotu 4MHz, což splňuje průvodní podmínku.

Při frekvenci procesoru 48MHz bude perioda 21nanosekund. Protože je použito dělení dvanácti, bude perioda hodinového signálu sériového portu lehce přes 250 nanosekund. Minimální předstih dat musí být 30 nanosekund, což je zde splněno zhruba osmkrát. Dále musí být splněna podmínka minimální doby trvání vysoké napěťové úrovně, která je 40 nanosekund. Zapisovací impuls CLKOUT trvá tři periody, takže jeho hodnota vyhází na 63 nanosekund. Tím je podmínka splněna. Piny SYNCIO a \overline{CS} jsou připojeny na zem, protože pro toto zapojení nemají přímé opodstatnění.

4.2.4 Spínač S2

Spínač slouží jako náhrada za tlačítka pro zjednodušení konstrukce a úsporu místa. Spínač je určen ke spínání tří signálů a to RESET a WAKEUP pro FX2 a zbylý RESET modulátoru, který je vyveden na paralelní port.

4.2.5 Sériová paměť EEPROM

Jako paměť byla použita součástka 24LC64 v provedení SMD v pouzdru SO-08

s kapacitou 8kB, což je stejná velikost, jakou disponuje čip FX2. Paměť lze použít buď pro uložení celého programu nebo pouze pro uložení identifikačních údajů o zařízení. Paměť má adresní linky A1 a A2 připojeny na zem a A0 je připojeno na napájecí napětí. Linky SDA a SCL jsou přes dva „pull-up“ rezistory připojeny k napájecímu napětí.

4.2.6 Sériový port SIO-1

Pro připojení k PC slouží také sériový port RS232, který využívá druhou sériovou bránu čipu FX2. Je určen pro ladění zdrojového kódu pomocí vývojového prostředí KEIL μ Vision2 a vyšší. Přes rozhraní USB se do paměti FX2 nahraje testovací program Monitor. Ten pak přes sériový port odesílá stavy registrů a obsazení paměti. Když zavedení programu do paměti proběhne úspěšně, rozsvítí se zelená dioda LED. Hardwarově je převodník sériové linky na RS232 realizován součástkou MAX3232CPE od firmy Maxim. v součástce jsou obsaženy dva převodníky, ale použit je pouze jeden, protože první sériový port je již využíván modulátorem.

4.2.7 Konektor SV2

Na konektoru SV2 je vyvedeno paralelní rozhraní pro posílání dat a sériový port pro konfiguraci kvadrurního modulátoru. Na konektoru jsou dále ještě některé řídicí signály pro modulátor. Konektor umožňuje připojit modulátor přes plochý kabel, tím se ale objevuje omezení na použití prvních čtrnácti bitů datové sběrnice. Aby byla sběrnice kompletní, jsou zbylé dva piny vyvedeny na konektor JP1.

4.2.8 Ostatní piny

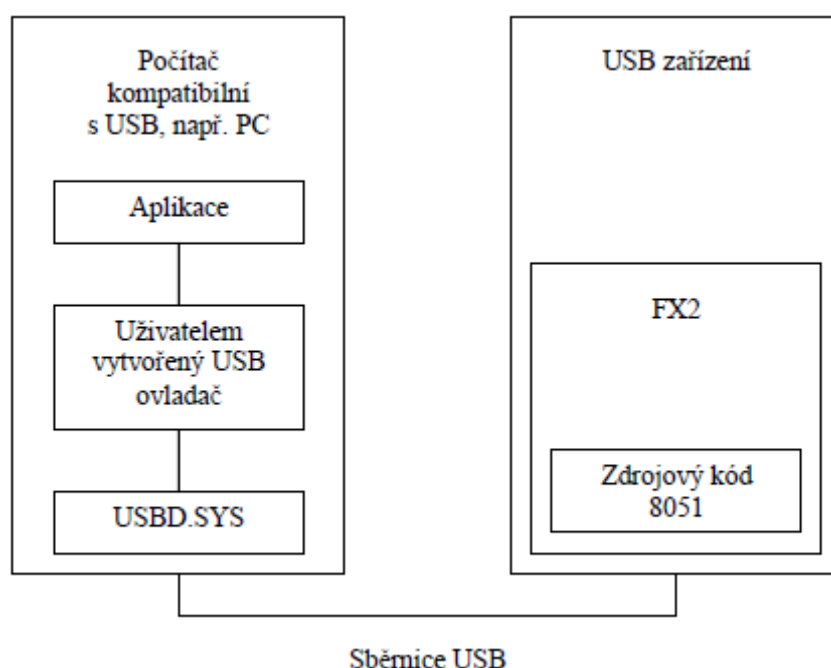
Poslední dva piny, které je nutno připojit z konektoru modulátoru je Reset a DPD. Pin DPD se propojí s TxEnable a tím je dáno, že když modulátor přestane přijímat data z převodníku, počká až se zpracují zbylá načtená data a přejde do úsporného režimu. Pin Reset je vyveden na spínač S2.

5 SOFTWARE A DRIVERY PŘEVODNÍKU

Pro převodník je stejně důležité jak konstrukční řešení, tak i ovládací software, se kterým se bude přímo pracovat a drivery, které budou zprostředkovávat komunikaci mezi zařízením a ovládacím programem.

5.1 Struktura ovládání

K tomu, aby mohlo být ovládáno jedno zařízení připojené na k počítači je potřeba několik částí, které spolu vytvoří funkční celek. Na Obr. 5.1 je zobrazen blokový diagram částí funkce systému pro čip FX2.



Obr. 5.1: Blokové schéma ovládání pro FX2 [8]

V první části schématu jsou tři bloky. Jako první je aplikace, která tvoří učitelské prostředí, přes které je možno ovládat připojené zařízení dle požadavků a možností. Aby mohla aplikace komunikovat s připojeným zařízením, je zapotřebí vlastní vytvořený ovladač, který bude určovat co je pro zařízení možné, jak se má ohlašovat a co může dělat. Tento ovladač je přímo určen pro dané zařízení. Poslední částí je systémový soubor USBD.SYS, což je soubor ovladače sběrnice, který je nutný pro komunikaci. Je to vlastně přímo ovladač sběrnice. Vytvořený ovladač se na tento soubor odkazuje a pomocí něho je možné ovládat dané připojené zařízení.

V dalším bloku, který je spojen přes sběrnici USB, je čip FX2, který má své vlastní specifické vlastnosti a také mikroprocesor, který čip ovládá dle zadaného programu. Tyto části tvoří s dalšími periferiemi celek, čili USB zařízení.

5.2 Ovladače

Aby mohlo připojené zařízení k počítači správně komunikovat, je zapotřebí ovladače, který zprostředkuje instrukce a jejich určení z ovládacího softwaru do zařízení.

5.2.1 Základní poznatky o ovladačích

Všeobecně je ovladač sled určitých daných instrukcí, které říkají systému jak se má k danému zařízení chovat a jaké meze mu může umožnit. Také určuje o jaké zařízení se jedná. Systém Windows si při prvním připojení zařízení hledá ovladače, které toto zařízení ovládají a specifikují ho. Pokud nebude nalezen správný ovladač, systém Windows nebude se zařízením dále komunikovat, protože není zajištěna jeho plná funkčnost. Určujícími faktory pro ovladače jsou identifikační údaje, které jsou skryty pod Vendor ID (VID), což je identifikátor dodavatele a Produkt ID (PID), což je identifikátor výrobku, který si určuje sám výrobce. VID si výrobce nemůže sám určovat, ale musí požádat o přidělení tohoto čísla organizaci USB. Tím se velice znesnadňuje vývoj aplikací pro USB, zvláště pro volné nebo amatérské použití. Celá ovladač se bude skládat z jednoho souboru s koncovkou *.inf, který bude obsahovat jméno daného ovladače *.sys a použité identifikátory. Jeho další části mohou být údaje o verzi operačního systému (odstavec Version), údaje o výrobci hardware (odstavec Manufacturer) a další nepovinné údaje. Poté co se systém dotáže na ovladače, se určí jako výchozí soubor *.inf a společně s ovladačem *.sys se zavedou do operačního systému. Poté můžeme vidět ve správci zařízení nově přidané zařízení, které odpovídá našim daným údajům

5.2.2 Zdrojový kód ovladače

Soubor *.inf popisuje daný ovladač. v souboru je na začátku uvedeno číslo GUID, které dělá ovladač jedinečným v daném systému, aby nedošlo ke konfliktu. Dále je zde uvedena třída, o jaké zařízení se jedná, např. USB nebo může být USB HID (tj, myš apod.). Poté zde lze najít výrobce a verzi ovladače.

Po těchto údajích následují informace o zařízeních, pro které je ovladač vhodný. To je určeno pomocí VID a PID. Každé z těchto zařízení patří do skupiny, pro kterou mohou být vytvořena vlastní pravidla, kterými se bude systém řídit. Jde hlavně o zavádění potřebného souboru *.sys k ovládání, vymezení paměti a odkazy na instrukce v souboru *.sys.

V poslední části je nastavení parametrů a odkazy pro přidružený soubor *.sys a hlavně je zde i uvedeno jméno každého zařízení podle jeho vlastního VID a PID. Toto jméno se zobrazuje jako název zařízení v systému. Ukázka struktury takového souboru *.inf je uvedena níže.

```
[Version]
Signature="$CHICAGO$"
Class=USB ;trida ovladace
ClassGUID={36FC9E60-C465-11CF-8056-444553540000} ;GUID cislo
provider=%CYPRESS% ;vyrobce
DriverVer=mm/dd/yyyy,x.y.v.z ;datum a cislo verze
;-----;
```

```

[SourceDisksNames]
1=%CYUSB_INSTALL%, , ,

[SourceDisksFiles]
CyUsb.sys = 1
CyUsb.spt = 1

[Manufacturer]
%MfgName%=Cypress      ;jmena vyrobce

[Cypress]
%VID_04B4&PID_8613.DeviceDesc%=CyUsb, USB\VID_04B4&PID_8613
;urcene zarizeni podle VID a PID
[DestinationDirs]
CyUsb.Files.Ext = 10,System32\Drivers
;-----;

[CyUsb.NT]
CopyFiles=CyUsb.Files.Ext
;-----;
AddReg=CyUsb.AddReg

[CyUsb.NT.HW]
AddReg=CyUsb.AddReg.Guid

[CyUsb.NT.Services]
Addservice = CyUsb, 0x00000002, CyUsb.AddService

[CyUsb.AddService] ;servisni nastaveni dle *.sys
DisplayName      = %CyUsb.SvcDesc%
ServiceType      = 1                      ; SERVICE_KERNEL_DRIVER
StartType        = 3                      ; SERVICE_DEMAND_START
ErrorControl      = 1                      ; SERVICE_ERROR_NORMAL
ServiceBinary     = %10%\System32\Drivers\CyUsb.sys
LoadOrderGroup   = Base

[CyUsb.AddReg]
HKR,,DevLoader,,*ntkern
HKR,,NTMPDriver,,CyUsb.sys

[CyUsb.Files.Ext]
CyUsb.sys

[CyUsb.AddReg.Guid]
HKR,,DriverGUID,,%CyUsb.GUID%
;-----;

[Strings]
CYPRESS="Cypress"      ;jmeno ve skupine
MfgName="Cypress"      ;jmeno tvurce
CYUSB_INSTALL="Cypress Generic USB Driver Installation Disk"
VID_04B4&PID_8613.DeviceDesc="Cypress      Generic      USB      Device"
;zobrazovane jmeno v ovladacich
CyUsb.SvcDesc="Cypress Generic USB Driver"
CyUsb.GUID="{AE18AA60-7F6A-11d4-97DD-00010229B959}" ;GUID      pro
tento ovladac

```


Soubor *.inf by byl k ničemu, kdyby neodkazoval na žádný systémový soubor *.sys, ve kterém jsou obsaženy hlavní instrukce ovladače. Tento systémový soubor musí být uveden v *.inf. Počet funkcí a jejich rozsah je značný a tato práce není vypracovávána jako příručka, proto zde bude uvedeno jen několik funkcí ovladače, přičemž některé zdrojové kódy s popisy jsou na doprovodném CD. Programovacími jazyky, kterých se hlavně využívá jsou Visual Basic.net, Visual C#, Visual C++ a Visual J#. Příklad zdrojového kódu pro transfer dat ke koncovému bodu:

```

USBDeviceList usbDevices = new
USBDeviceList(CyConst.DEVICES_CYUSB);

CyUSBDevice MyDevice = usbDevices[0x04B4,0x1003] as CyUSBDevice;

if (MyDevice != null)

if (MyDevice.IsocInEndPt != null)

{

int len = MyDevice.IsocInEndPt.MaxPktSize * 8;

byte [] buf = new byte[len];

MyDevice.IsocInEndPt.XferSize = len;

MyDevice.IsocInEndPt.XferData(ref buf, ref len);

}

```

Příklad zdrojového kódu pro získání deskriptoru:

```

USBDeviceList Devices = new USBDeviceList(CyConst.DEVICES_CYUSB);

CyUSBDevice MyDevice = Devices[0] as CyUSBDevice;

USB_INTERFACE_DESCRIPTOR descriptor = new
USB_INTERFACE_DESCRIPTOR();

MyDevice.GetIntfcDescriptor(ref descriptor);

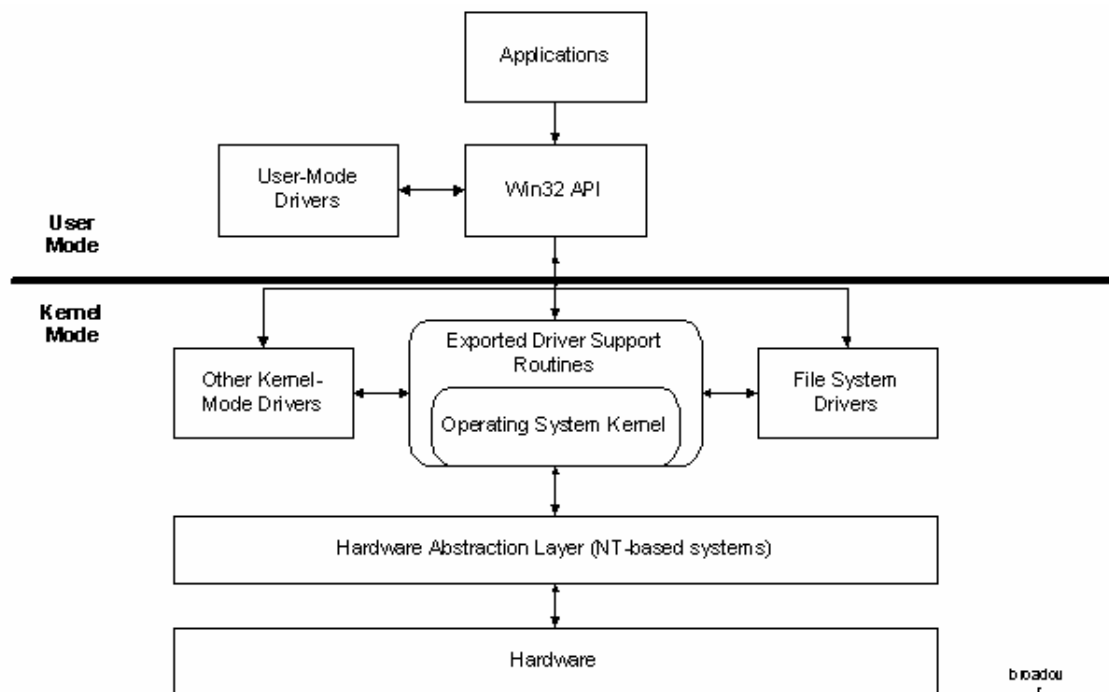
```

5.2.3 Tvorba ovladače

Tvorba ovladače pro ovládání zařízení není nijak jednoduchá a vyžaduje dobrou znalost programovacího jazyka C. Vytvoření ovladače usnadňuje nástroj od firmy Microsoft a to je vývojový balík pro ovladače DDK (Driver Development Kit), který je volně dostupný. Obsahuje základní konzoly pro vytvoření ovladače a také konzoly pro odlazování a hledání chyb. Krom toho ještě obsahuje řadu nástrojů jako grafické nadstavby a testery. Možnost vyhledávat zařízení, která ještě nebyla systémem zdárně definována apod.

Při celkovém pohledu na komponenty můžeme udělat rozdělení do dvou skupin a to na uživatelské součásti a jádro. Ovladač může využívat uživatelské rutiny, které

jsou definovány ve Win32 v rozhraní pro aplikační programy (API), které jsou popsány blíže v SDK. Jádro Win32 API vyvolá ovladač a jádro systému vyvolá svoje pracovní rutiny pro správnou funkci, které napomohou nakonfigurovat I/O porty, výkon zařízení, stabilitu systému apod. Na Obr. 5.2 je blokové zobrazení tohoto systému.



Obr. 5.2: Blokové zobrazení funkce systému z hlediska ovladače

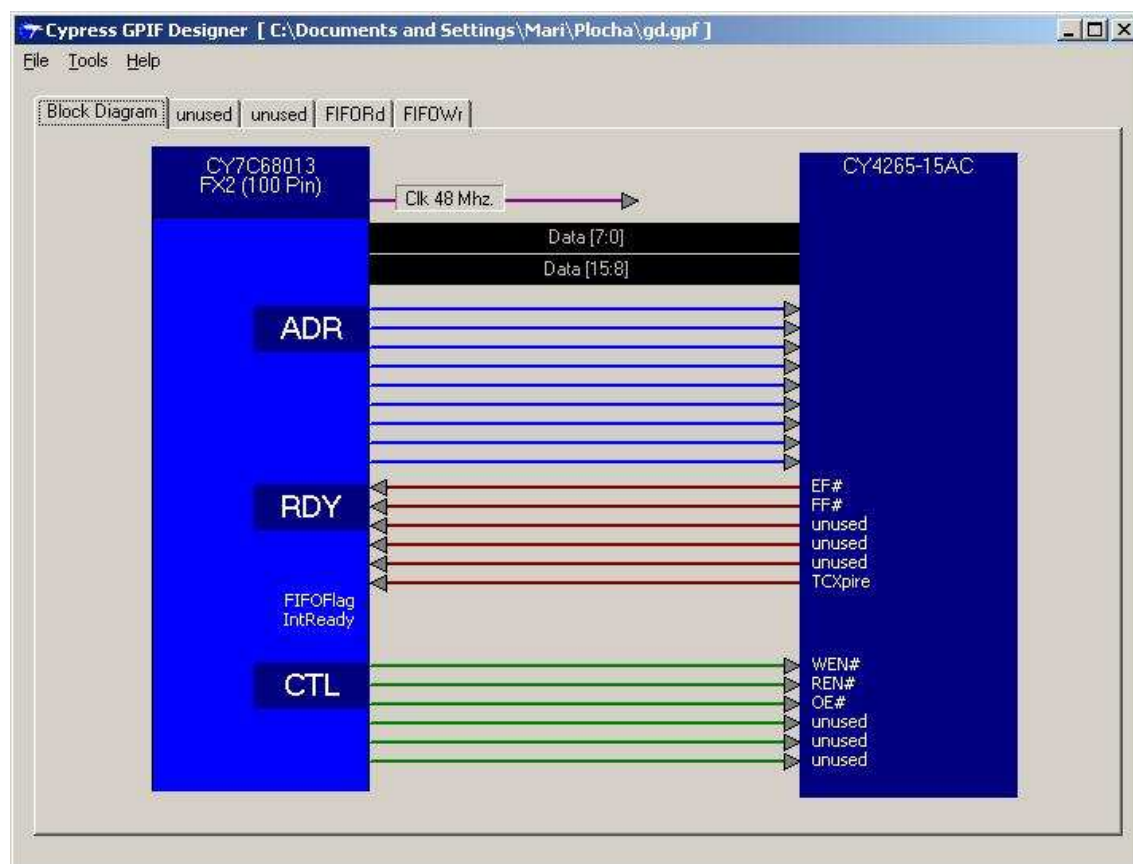
Pro tvorbu je několik fází, prvně by mělo následovat porozumění operačnímu systému a jeho ovladačům, což je nezbytný předpoklad. Dále určení požadavků pro ovladač a jeho návrh. Poté už jen jeho sestavení a testování. Pro všechny tyto úkony jsou ve Windows DDK obsaženy nástroje. Lazení ovladače je ale poměrně časově náročné a složité, neboť špatně napsaný zdrojový kód může vést k obrazovce s „modrou smrtí“ nebo i pádu celého systému.

5.2.4 Aspekty tvorby

Ovladač sám o sobě je standardní zdrojový kód, který obsahuje informace o nastavení zařízení, jeho funkci apod. Nevýhodou je, že zdrojový kód ovladače není k čemu vztáhnout a tak je zpočátku obtížné si uvědomit, jaký je princip psaní. Syntaxe jsou spíše pevně dány a lze je najít v poučkách, v případě programování ovladače pro čip FX2 to bylo v souhrnných manuálech. Problém spíše nastává v souslednosti funkcí, které následují za sebou. Při nedodržení logiky nastává chyba, která v případě ovladače může vést k obrazovce s modrou smrtí, případně dále až k pádu systému. Ovladač ovlivňuje registry. To si vyžaduje poměrně zdlouhavé lazení a je tedy lepší nepsat z počátku přesný ovladač, který odpovídá danému zařízení, ale nechat určitý rozptyl parametrů, zvláště, pokud se zároveň ještě vyvíjí ovládací aplikace nebo další software pro ovládání připojeného zařízení, tedy pro jeho mikroprocesor.

5.3 GPIF Designer

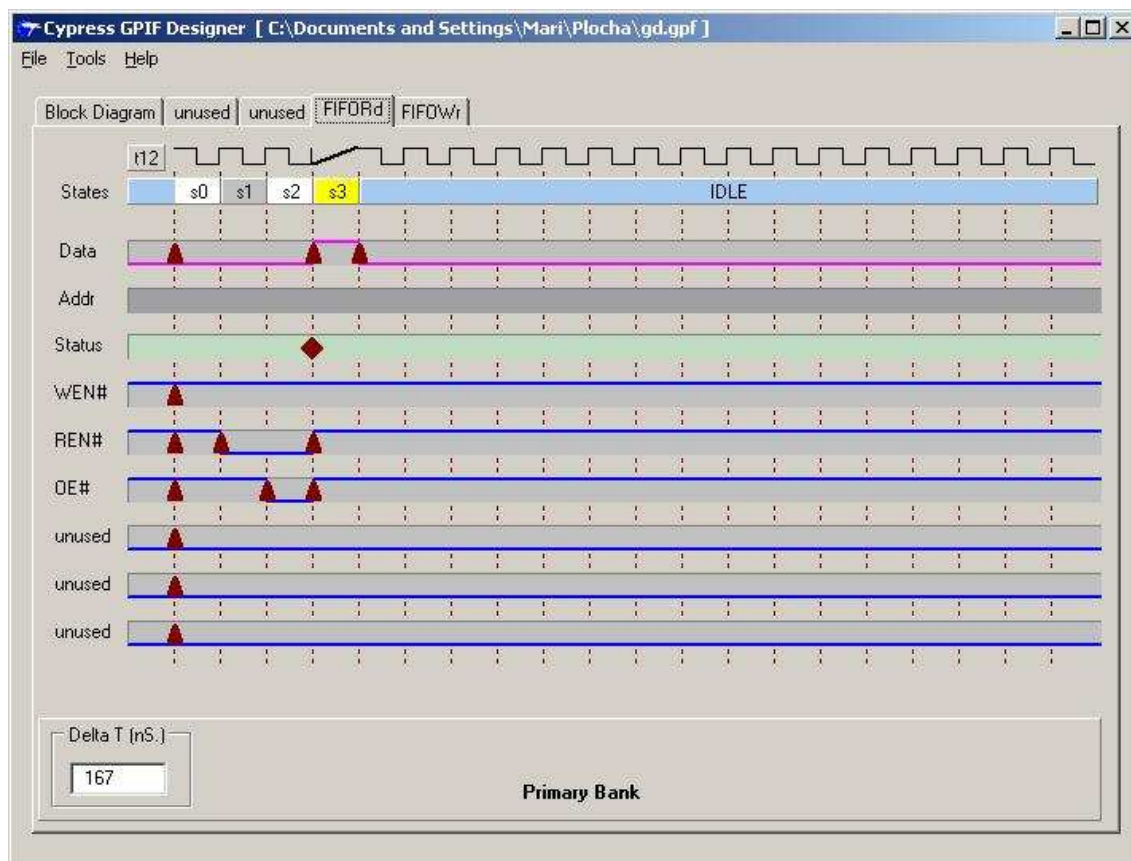
GPIF Designer je program, pomocí kterého lze lehce nastavit vlastní funkci zařízení. Lze nastavit hodinový kmitočet čipu, šířku využívané sběrnice datové i adresní, veškeré řídicí signály atd. Mimo to lze v programu ještě nastavit čtení a zapisování do FIFO. Při určování časových průběhů lze ještě nastavovat podmínky, za kterých budou probíhat. Výhoda je, že vše probíhá vizuálně a není tak potřeba se zdlouhavě zabírat zdrojovým kódem v jazyce C. Výsledné nastavení program vyexportuje do zdrojového kódu v jazyce C. Vyexportovaný soubor pak stačí připojit k hlavnímu programu, buď přímo do kódu, nebo vložit do hlavičky. Ukázky programu jsou na Obr. 5.3 a 5.4.



Obr. 5.3: Hlavní panel

Aby byl hlavní program kompletní, musí se ještě přidat rozpis deskriptorů, pro dané zařízení, aby se správně zvolil ovladač a neomezovala ji nějak funkce.

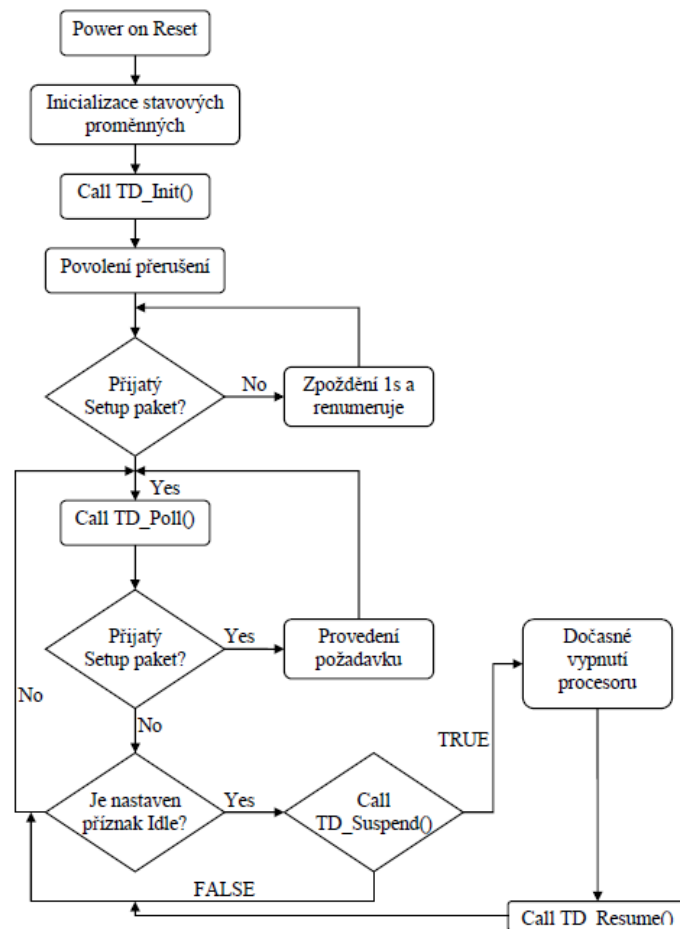
Ovládání programu je uživateli na první pohled zřejmé, překážkou může být neznalost některých parametrů čipu a jejich návaznost na sebe. Potom by mohl vzniknout chybný program. Jak bylo popsáno výše, kde se popisovaly vlastnosti čipu, toto rozhraní slouží pro přímý výstup dat z pamětí. Nejde zde nastavit nic, co je společné se mikroprocesorem.



Obr. 5.4: Nastavení funkce při čtení paměti FIFO

5.4 Firmware FrameWorks

Firmware FrameWorks je přístup, který umožní programátorovi naprogramovat aplikaci pro zařízení v co nejkratším čase a s co největší efektivitou. Obsahuje zdrojový kód pro inicializaci čipu, obsluhu požadavků USB a také dočasné odpojení USB od napájení. Stačí pak tedy jen definovat tabulku USB deskriptorů a zdrojový kód, který bude obstarávat funkci periférií k vytvoření funkčního USB zařízení.



Obr. 5.5: Vývojový diagram zdrojového kódu pro FX2 [8]

Podle vývojového diagramu se nejprve provede inicializace stavových proměnných, poté následuje volání funkce `TD_Init()`, která inicializuje rozhraní a aktivuje koncové body. Po ukončení těchto úkonů se povolí přerušení a program očekává Setup packet. Pokud packet zatím není přijat, probíhá renumerace a zpoždění 1s. Pokud je packet přijat, následuje volání funkce `Call TD_Poll()`, kde jsou naprogramovány instrukce, které se mají provést. Dále se určí, zda je v packetu standardní požadavek, pokud ano, požadavek je zpracován. Nyní se určí, zda je požadavek na ukončení činnosti procesoru, pokud tomu tak není, tak se celý cyklus opakuje, pokud je, zavolá se funkce `TD_Suspend()`. Po potvrzení se přepne procesor do suspens módu. z toho se může vrátit kdykoliv, kdy obdrží požadavek na obnovení činnosti. To se zavolá funkce `TD_Resume()` a pokračuje se opět od začátku voláním `TD_Poll()`.

5.5 Procesor 8051

Jednou z možností, jak docílit u čipu požadavků je vytvořit program pro 8051. Jak již bylo popsáno výše, nejsou u tohoto procesoru téměř žádné odlišnosti, až na rychlejší chod programu, takže se musí přepočítávat časové konstanty. Pokud rozhraní GPIF, je

nejlepší využít vlastností procesoru. Jeho nastavení skýtá mnohem větší možnosti a funkce, pokud však bude vykonávat více instrukcí a je požadována vysoká rychlost, může přenos dat nepatrně brzdit.

5.6 EZ-USB Control Panel

Firma Cypress poskytuje jako podporu řadu aplikací, jednou z nich je i EZ-USB Control Panel. Program slouží pro práci se zařízeními USB za pomoci ovladače EZUSB.SYS. Program disponuje všemi standardními požadavky na zařízení USB (USB request) a dále také nestandardními, jako je Vendor request. Pomocí tohoto programu si můžeme vyžádat deskriptory zařízení, uložit vlastní zdrojový kód do připojeného USB zařízení a to buď do čipu nebo do externí paměti, zasílat požadavky výrobce, odesílat a přijímat data nebo soubor typu BULK, testovat smyčkou přes koncové body přenosem typu BULK a vyzkoušet isochronní přenos. Při nahrání programu do zařízení se zařízení renumeruje uložený program se poté okamžitě začne vykonávat.

5.7 Uživatelské prostředí v Matlabu

Pro vytvoření uživatelského programu, kterým bude možno ovládat přenos dat a volit přenášaná data na výstup do modulátoru, je možné zvolit z několika možností. Asi dvě nejlepší jsou buď napsání programu v jazyku C nebo vytvořit příslušné rozhraní v softwaru Matlab, který nabízí další možnosti a prvky pro práci při měření a jeho složitost pro zobrazování výsledků není tak složitá. Překážkou je nastavení Matlabu na port USB, na který se budou posílat data a to jak řídicí, tak data určená pro zpracování.

Instrukce programu je možno napsat standardním způsobem nebo využít Toolboxů. Pro normální použití je možné vytvořit objekt VISA-USB.

```
vu = visa('ni','USB::0x1234::125::A22-5::INSTR');
```

Kde „vu“ je název nového VISA objektu, 'ni' je daná konstanta, USB určuje zařízení, 0x1234 je VID požadovaného zařízení, 125 je modelovací kód, A22 je sériové číslo a 5 je počet rozhraní. Po vytvoření objektu jsou v něm automaticky dané hodnoty. Tyto hodnoty je možno si lehce zobrazit pomocí funkce get.

```
get(vu,{'Name','RsrcName','Type'})
```

```
ans = 'VISA-USB-0-0x1234-125-A22-5-0' 'USB::0x1234::125::A22-5::INSTR'
```

Objektech VISA se lze lehce na zařízení dotazovat a měnit jeho parametry podle potřeby. Bohužel nebyla možnost zařízení odzkoušet a tak i zjistit potřebnou věc, jako je posílání dat do určených koncových bodů nebo podle toho rozdělení datových toků na dva rozdílné.

6 ZÁVĚR

V této bakalářské práci bylo úkolem vytvořit převodník z portu USB na čtrnácti bitový paralelní port, na který bude dále připojen kvadraturní modulátor a to s tím, že data budou posílána co možná nejrychlejším způsobem při standartu USB 2.0. Dále pak by bylo vhodné nastavovat kvadraturní modulátor pomocí sběrnice USB, aby se nemusel odpojovat při požadované změně nastavení.

Práce by se dala rozdělit do dvou částí, první z nich byla mechanická realizace pouze převodníku, protože deska plošného spoje kvadraturního modulátoru byla již zhotovena výrobcem osazeného čipu AD9857. Tohoto úkolu se již zhostil Ing. Igor Majko a navrhl plošný spoj, který byl osazen vybraným řadičem CY7C89013a. Důraz byl kladen na to, aby převodník mohl pracovat s co největší rychlostí, což se nakonec povedlo a změřená hodnota přenosu byla 36MB/s. v této bakalářské práci byla provedena úprava na návrhu plošného spoje a zapojení tak, aby vše odpovídalo standartu USB. Byl tedy změněn konektor na konektor typu B a to v úpravě pro povrchovou montáž. Dále byly upraveny vodivé cesty pro tento konektor a snaha zmenšit počet prokůvů na desce. Na již zhotoveném převodníku není možné konektor zaměnit, protože uspořádání vývodů neodpovídá vodivým cestám a i rozměry konektoru jsou rozdílné. Proto tam byl ponechán stávající konektor typu A, pro který musí být jiný kabel, než pro běžná propojení PC s připojeným zařízením.

V druhé části se jedná o softwarové vybavení, které bude sloužit pro zprostředkování komunikace řídicího programu se sběrnici USB a tím i připojeného zařízení, tedy ovladačem (driverem). Dále o program, kterým se bude v systému ovládat převodník, přesněji data, která do něj budou proudit a také data pro sériový port, kterým se bude nastavovat režim a profil modulátoru. Posledním článkem je ovládací program pro samotný čip převodníku (FX2). Zde se skýtají dvě možnosti a to, že je možno pomocí rozhraní GPIF posílat data přímo na výstup bez toho, aby byla nějak přeprocessována procesorem nebo doplnit program pro mikroprocesor, který bude zpracovávat příchozí data a dále je bude odesílat na patřičný výstup.

Při řešení softwaru se vyskytlo několik problémů. První je rozhodnutí, zda je lepší rozhraní GPIF nebo s výpomocí procesoru. Mikroprocesor by podle výrobce neměl nijak omezovat rychlost, za to však je pro něj nutný program, který je složitější na vytvoření, než u GPIF. Rozhraní GPIF nabízí rychlé řešení, které je možné snadno vytvořit a nijak více neomezuje čip. Nevýhoda však je při posílání sériových dat pro řízení modulátoru, ale je stejně nutný zaváděcí soubor, který provede enumeraci zařízení. Pro první zkoušení bylo vhodnější využít rozhraní GPIF.

Dalším problémem je vytvoření ovladače pro systém Windows, který by spolehlivě ovládal převodník. První možnost, kdy by bylo možné využít předtvořeného ovladače firmou Cypress je nevhodná, protože jádra ovladačů sys jsou již zkompileovaná a nepodporují rozsáhlejší přenosy dat. Jsou určeny hlavně pro zkoušení funkčnosti zařízení a to pro jeden koncový bod. Povětšinou EndPoint0, který je standardní pro všechna USB zařízení. Tyto ovladače ale výborně posloužili pro studium funkčnosti. Možností, která byla využita, je vzít část vytvořených ovladačů a část vlastního dotvoření podle dodávaných manuálů. Poslední možností bylo vytvořit celý vlastní ovladač. Nicméně tvoření ovladače bylo poměrně problematické, protože vytvoření

správného zdrojového kódu bylo velice náchylné na chyby programátora a tak při kompilaci často vznikala kolize ovladačů v jádře a tím i odezva systému jako „modrá smrt“ nebo v horším případě i pád samotného systému. Další znesnadnění pocházelo od samotného řadiče Cypress, protože při aktualizaci ovládacího kódu čipu se objevila nová VID a PID, která byla i leckdy nestandardní a tak proto, aby bylo možné čip vybavit novým programem, musel být vytvořen i nový ovladač, který byl schopen se zařízením pracovat a systém Windows ho uznal jako vhodný k vybranému zařízení.

Poslední část je ovládací software, se kterým bude pracovat přímo uživatel. Vytvořit program v jazyce C by bylo poměrně náročné na čas a tak se přešlo ke schůdnějšímu řešení, které nabízelo i více možností, a to k vytvoření ovládacího programu v softwaru Matlab. Ten přímo podporuje komunikaci s USB, nebylo však zjištěno jak docílit rozdělení koncových bodů v datovém toku, protože je stále problém s enumerací zařízení pro přehrání vnitřního kódu.

Práce je tedy stále ve stádiu rozpracování, kdy je vytvořeno několik ovladačů pro zařízení, stejně tak několik programů pro čip FX2, ale není vytvořen funkční program pro užívání ve Windows, protože komunikace zařízení je omezena díky jinému vnitřnímu programu a nekompatibilnímu driveru.

Vhodným postupem, kterým měl být aplikován již od začátku je prvně vytvořit ovládací program pro systém Windows, který by umožňoval nastavování modulátoru přes sériový port a tvar posílání dat do modulátoru přes převodník. Podle požadavků tohoto programu by měl být vytvořen předpokládaný program pro čip FX2 a v důsledku tohoto ovladač pro systém Windows, který by byl následně odlazen, aby nedocházelo ke kolizím.

LITERATURA

- [1] BURKHARD, Kainka. USB - měření, řízení a regulace pomocí sběrnice USB. Praha, BEN 2003.
- [2] USB Implementers Forum. Universal Serial Bus Specification Revision 2.0 [online]. Dostupný na WWW: <http://www.usb.org>.
- [3] Cypress Semiconductor. EZ-USB FX2 Technical Reference Manual [online]. Ver. 2.2. Dostupný na WWW: <http://www.cypress.com>.
- [4] Cypress Semiconductor. CY7C68013 EZ-USB FX2 USB Microcontroller [online]. Rev F. Dostupný na WWW: <http://www.cypress.com>.
- [5] Analog Devices. Datsheet k modulátoru AD9857 [online]. Rev. C. Dostupný na WWW: <http://www.analog.com>.
- [6] FTDI Chip. Datsheety k FT245xx, FT2232D a Vinculum VNC1L [online]. Dostupný na WWW: <http://www.ftdichip.com>.
- [7] Texas Instruments. Datsheet k TUSB6015 [online]. Dostupný na WWW: <http://www.ti.com>.
- [8] MAJKO, Igor. Diplomová práce: Řízení kvadrurního modulátoru AD9857 pomocí sběrnice USB. VUT Brno.
- [9] Wikipedie. Universal Serial Bus [online]. Dostupný na WWW: <http://cs.wikipedia.org>.
- [10] HW.cz. USB - Universal Serial Bus – Popis rozhraní [online]. Dostupný na WWW: <http://www.hw.cz>.
- [11] Compaq, HP, Intel, Lučeny, Microsoft, NEC, Philips. Universal Serial Bus Specification. 2000. www.usb.org
- [12] Analog Device. AD9857 Customer Evaluation Board. 2001

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

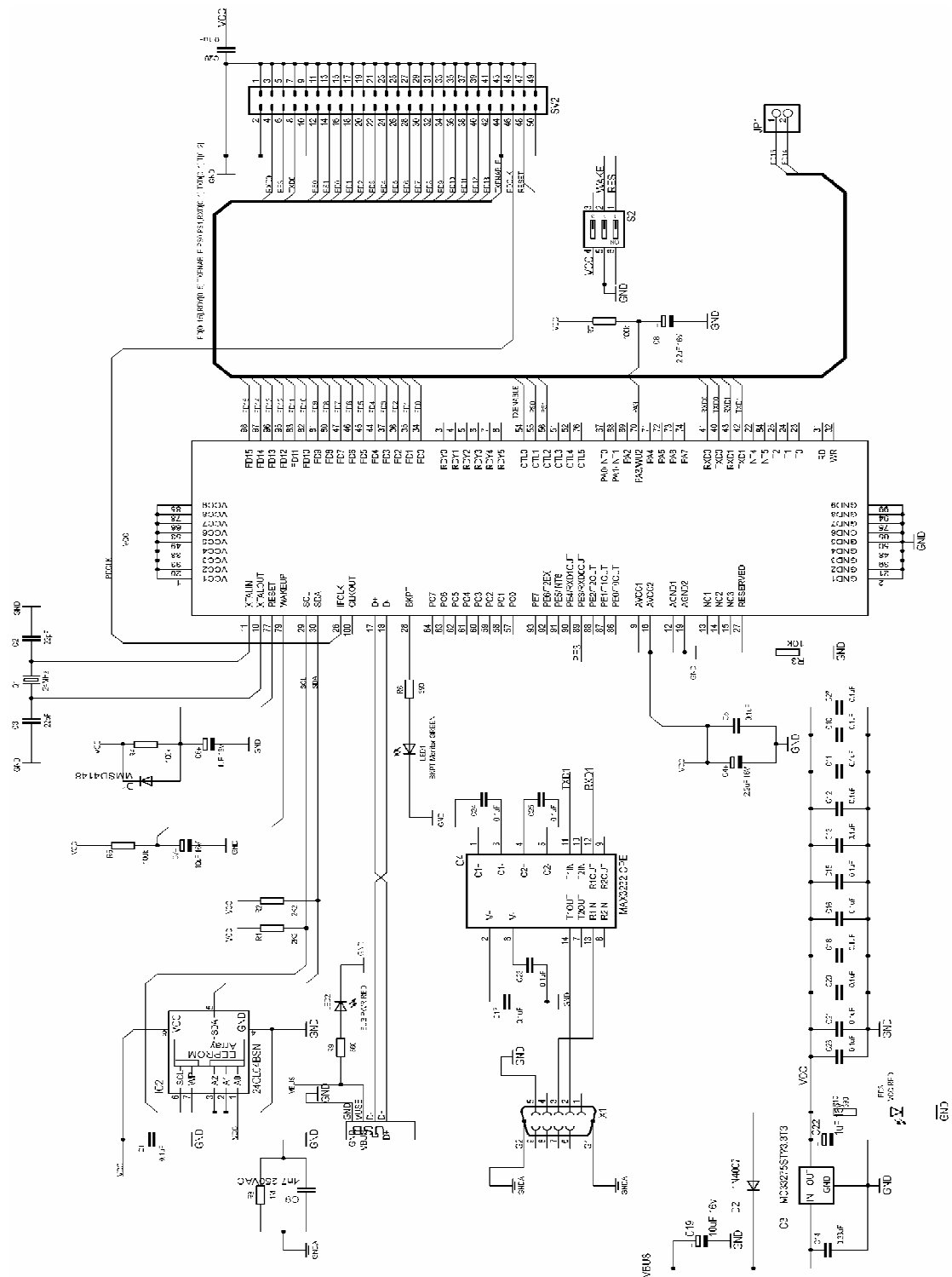
API	Application Programming Interface
ATA	Advanced Technology Attachment
CIC	Cascaded Integrator-Comb
DAC	Digital-to-Analog Converter
DDK	Driver Development Kit
DDS	Digital Data Storage
DID	Direct inward dialing
EPP	Enhanced Parallel Port
EEPROM	Electrically Erasable Programmable Read-Only Memory
FIFO	First In, First Out
FIR	Finite Impulse Response
GPIO	General Programmable Interface
HID	Human Interface Device
I ² C	Inter-Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
LSB	Least Significant Bit
MSB	Most Significant Bit
NRZI	Non Return to Zero, Inverted
PCMCIA	Peripheral Component MicroChannel Interconnect Architecture
PID	Product-ID
PLL	Phase Locked Loop
RAM	Random-Access Memory
ROM	Read-Only Memory
SDK	Software Development Kit
SIE	Serial Interface Engine
SMD	Surface Mount Device
USART	Universal Synchronous Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VID	Vendor-ID

SEZNAM PŘÍLOH

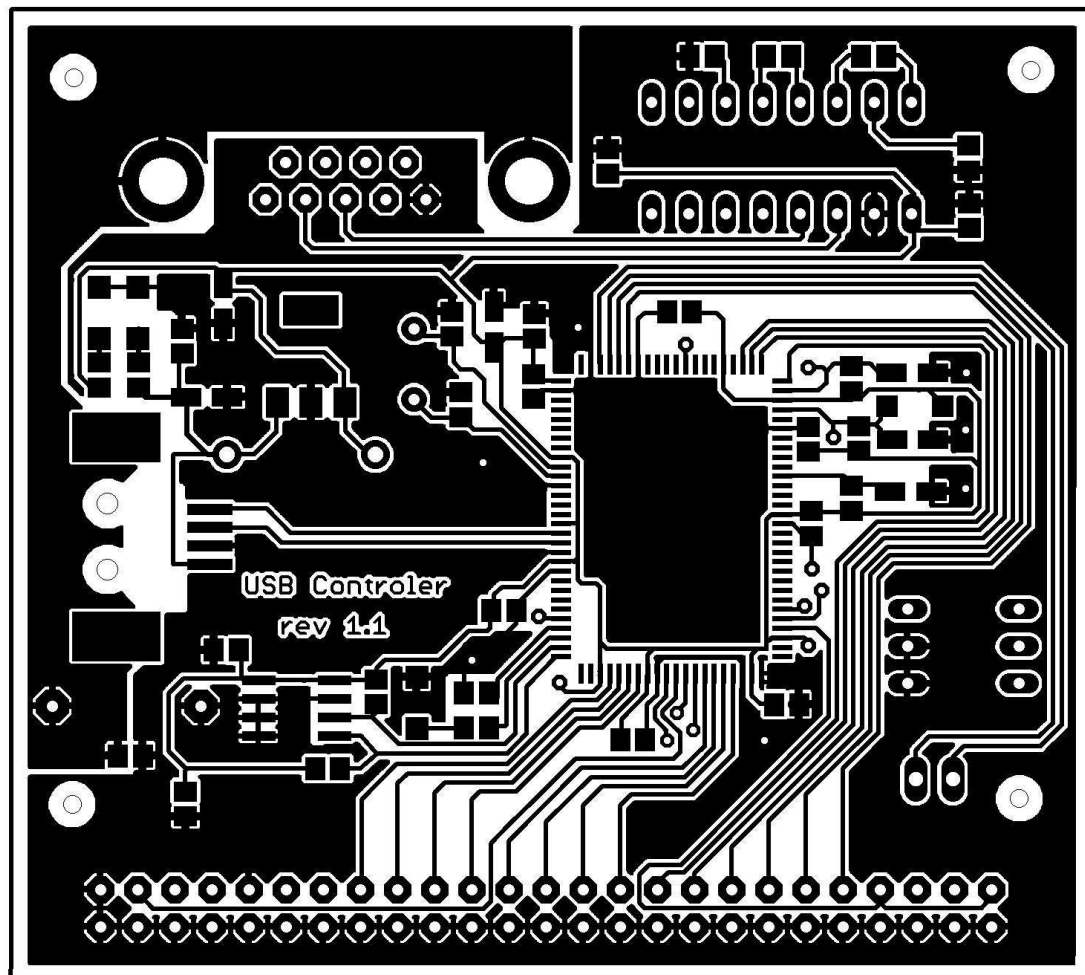
A	Návrh zařízení	50
A.1	Obvodové zapojení	50
A.2	Deska plošného spoje – top	51
A.3	Deska plošného spoje – bottom	52
A.4	Osazovací plán desky.....	53
B	Seznam součástek	54
C	Zapojení kitu s AD9857	55
D	Označení vývodů CY7C68013	56
E	SFR registr CY7c68013	57

A NÁVRH ZAŘÍZENÍ

A.1 Obvodové zapojení

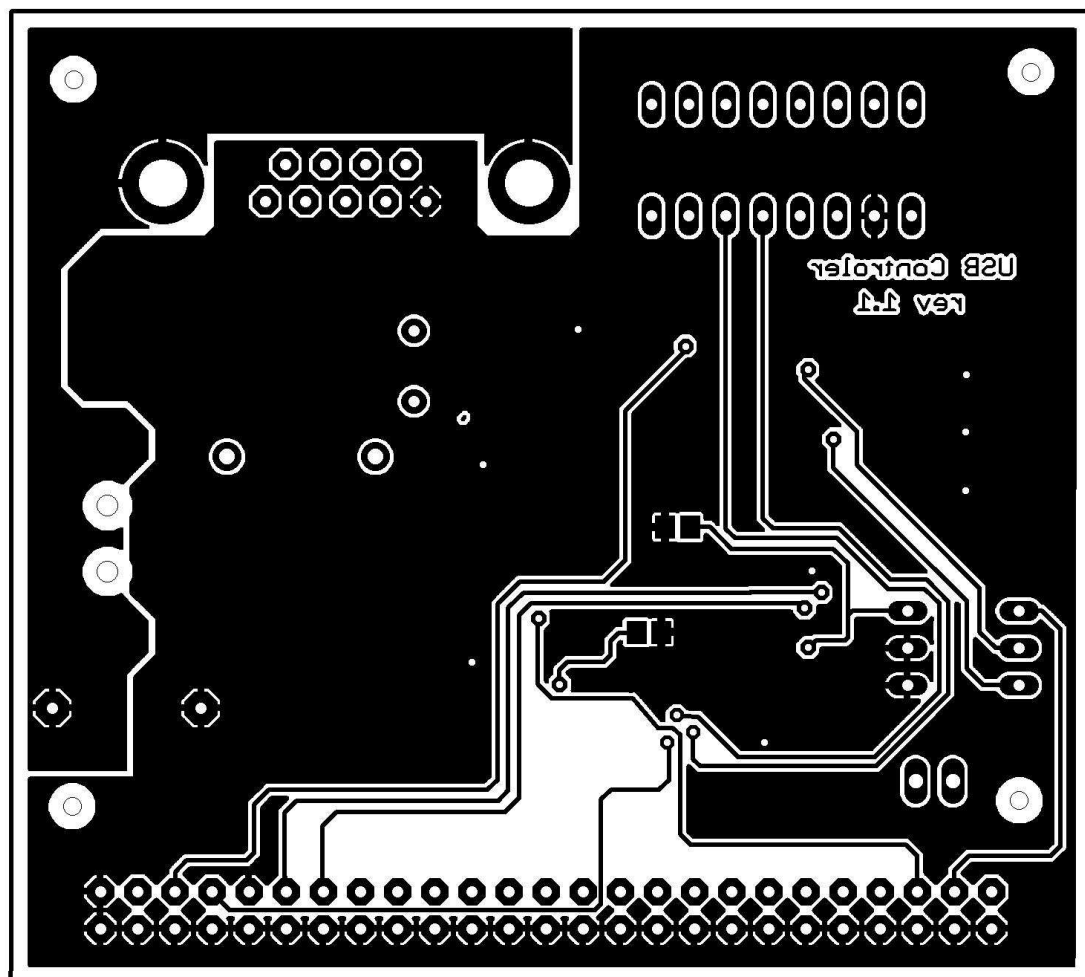


A.2 Deska plošného spoje – top



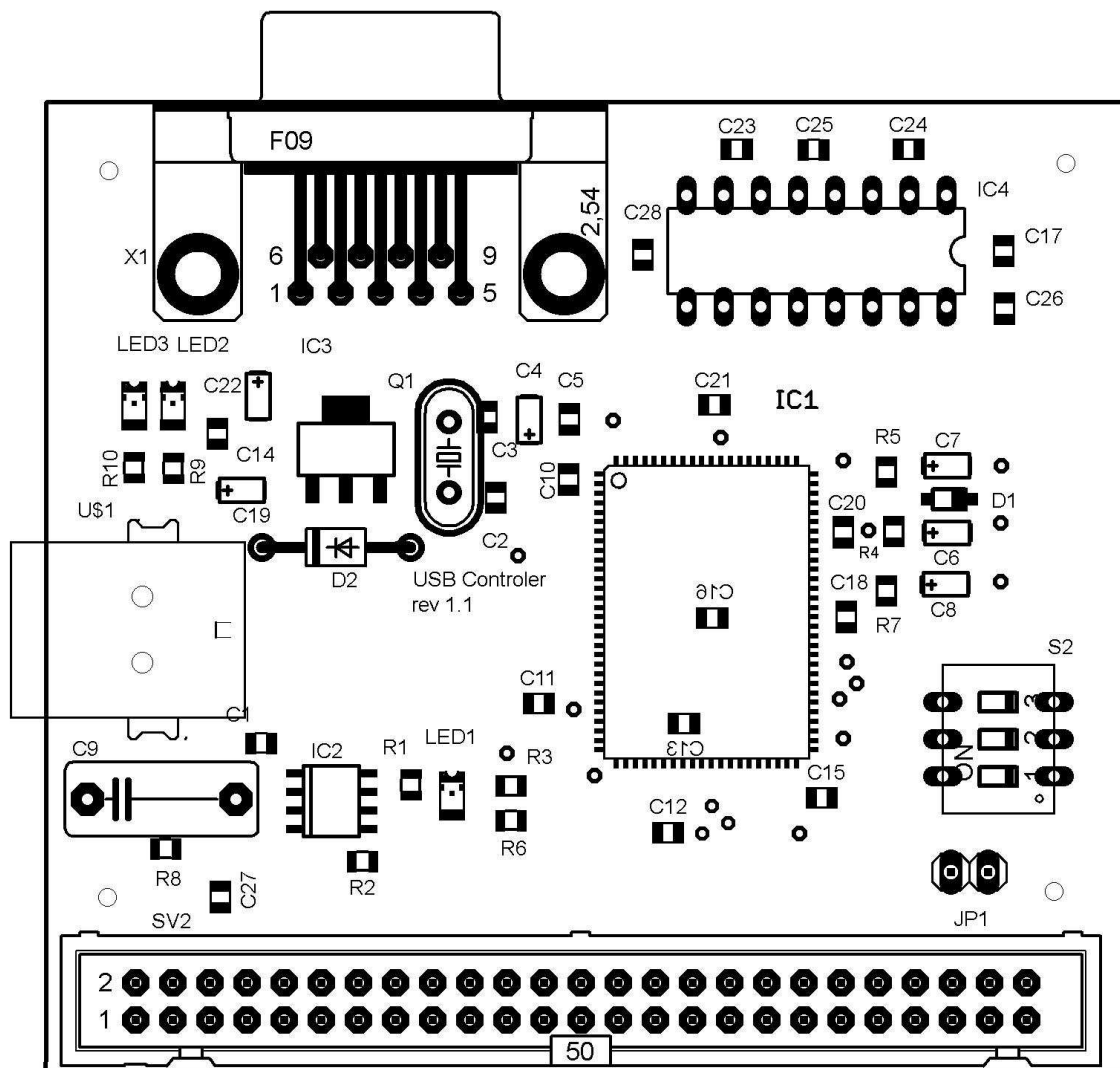
Rozměr desky nejsou uvedeny, protože obrázek slouží jen pro orientační přehled spojů. Velikost desek a celý jejich návrh je na doprovodném CD.

A.3 Deska plošného spoje – bottom



Viz. výše.

A.4 Osazovací plán desky

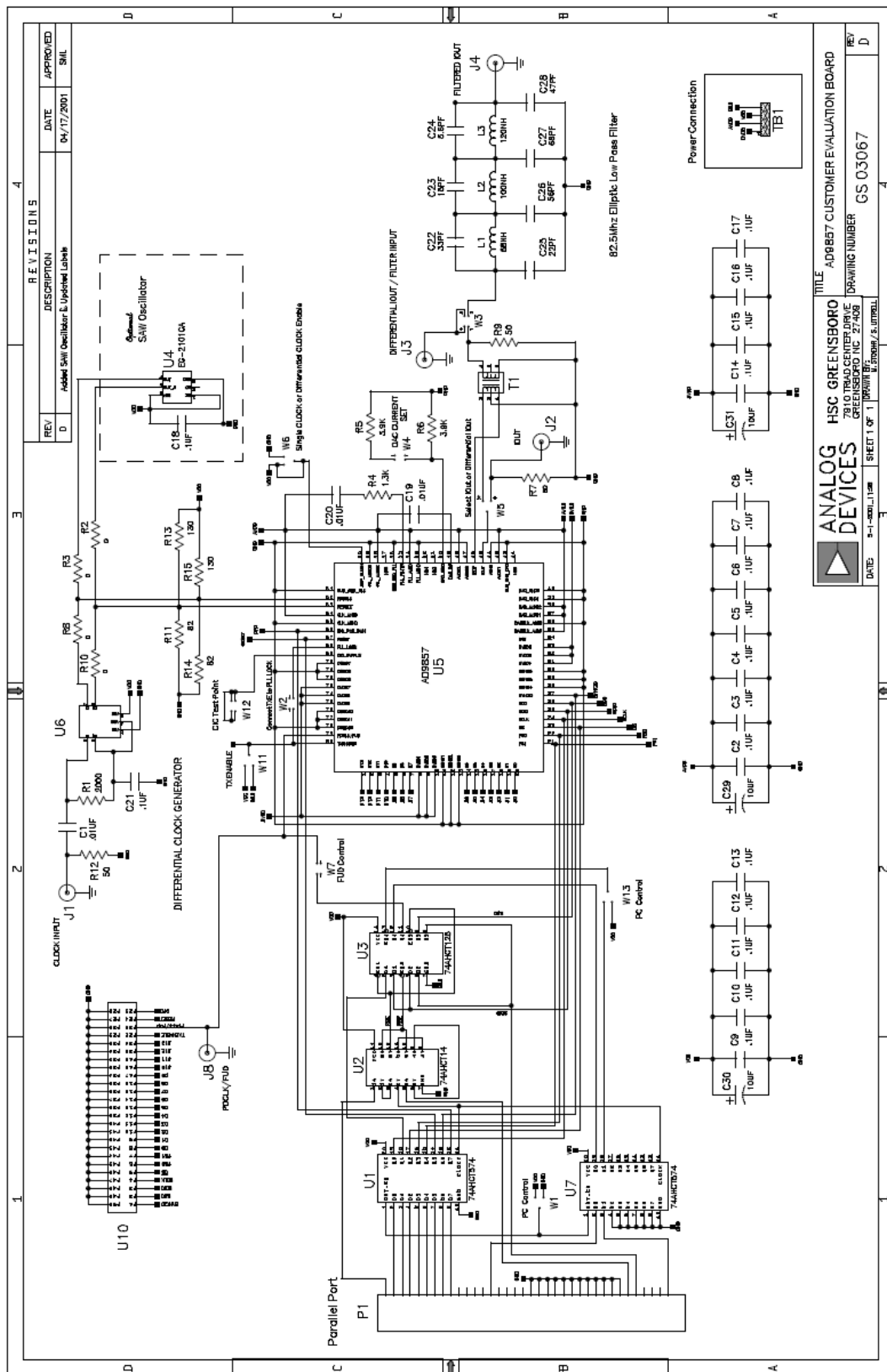


B SEZNAM SOUČÁSTEK

Seznam součástek			
Počet	Hodnota	Název	Součástka
1		CY68013	IC1
1		DIP03YL	S2
1		F09HP	X1
1		ML50	SV2
1		PINHD-1X2	JP1
18	0.1uF	C-EUC0805	C1, C5, C10-18, C20, C21, C23-28
1	0.33uF	C-EUC0805	C14
1	1M	R-EU_R0805	R8
1	1N4007	1N4004	D2
2	1uF 16V	CPOL-EUCT3216	C6, C22
2	2.2uF 16V	CPOL-EUCT3216	C4, C8
2	2K2	R-EU_R0805	R1, R2
1	4n7 250VAC	C-EU102-054X133	C9
1	10k	R-EU_R0805	R3
2	10uF 16V	CPOL-EUCT3216	C7, C19
2	22pF	C-EUC0805	C2, C3
1	24CL04BSN	24CL04BSN	IC2
1	24MHz	CRYTALHC49S	Q1
3	100k	R-EU_R0805	R4, R5, R7
2	390	R-EU_R0805	R6, R10
1	560	R-EU_R0805	R9
1	BKPT Monitor GREEN	LEDCHIPLED_1206	LED1
1	BUS PWR GREEN	LEDCHIPLED_1206	LED2
1	MAX3232 CPE	MAX232	IC4
1	MC33275ST?3.3T3	MC33275	IC3
1	MMSD4148	DIODE-SOD123	D1
1	USB1	USB1	U\$1

C ZAPOJENÍ KITU S AD9857

Z katalogového listu [12].



Z katalogového listu [3].



E SFR REGISTR CY7C68013

Z katalogového listu [3].

x	8x	9x	Ax	Bx	Cx	Dx	Ex	Fx
0	IOA	IOB	IOC	IOD	SCON1	PSW	ACC	B
1	SP	EXIF	INT2CLR	IOE	SBUF1			
2	DPL0	MPAGE	INT4CLR	OEA				
3	DPH0			OEB				
4	DPL1			OEC				
5	DPH1			OED				
6	DPS			OEE				
7	PCON							
8	TCON	SCON0	IE	IP	T2CON	EICON	EIE	EIP
9	TMOD	SBUF0						
A	TL0	AUTOPTRH1	EP2468STAT	EP01STAT	RCAP2L			
B	TL1	AUTOPTRL1	EP24FIFOFLGS	GPIFTRIG	RCAP2H			
C	TH0	reserved	EP68FIFOFLGS		TL2			
D	TH1	AUTOPTRH2		GPIFSGLDATH	TH2			
E	CKCON	AUTOPTRL2		GPIFSGLDATLX				
F		reserved	AUTOPTRSETUP	GPIFSGLDATLNOX				