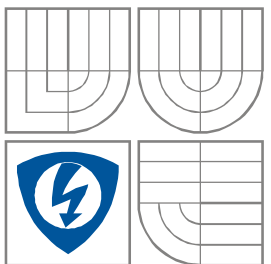


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

TISKOVÝ USB PŘEPÍNAČ

USB PRINTER SWITCH

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR
VEDOUCÍ PRÁCE
SUPERVISOR

Martin Mrník

Ing. Jiří Hermany

od 30. 4. 2010

prof. Ing. Stanislav Hanus, CSc.

BRNO, 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Bakalářská práce

bakalářský studijní obor
Elektronika a sdělovací technika

Student: Martin Mrnk

ID: 111106

Ročník: 3

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Tiskový USB přepínač

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s protokolem USB a navrhnete zařízení pro připojení několika počítačů k více tiskárnám pomocí rozhraní USB jako alternativu síťového tisku. Přepínač bude odesílat příchozí data na zvolenou tiskárnu. Vyberte vhodný mikrokontrolér a navrhnete zapojení obvodu.

Realizujte navržené zařízení, oživte je a otestujte.

DOPORUČENÁ LITERATURA:

[1] KAINKA, B. USB. Měření, řízení a regulace pomocí sběrnice USB. Praha: BEN, 2002. 1. vyd. 247 s. ISBN 80-7300-073-3.

[2] AXELSON, J. USB complete: everything you need to develop custom USB peripherals. New York: Lakeview Research, 2005. 3rd ed., 572 s. ISBN 978-1-931448-02-4.

Termín zadání: 8.2.2010

Termín odevzdání: 28.5.2010

Vedoucí práce: Ing. Jiří Hermany

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 části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Mrník Martin
Bytem: Halalovka 15, Trenčín, 911 08
Narozen/a (datum a místo): 29. září 1987 v Trenčíne

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 53, Brno, 602 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
 - diplomová práce
 - bakalářská práce
 - jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP: Tisový USB přepínač
Vedoucí/ školitel VŠKP: prof. Ing. Stanislav Hanus, CSc.
Ústav: Ústav radioelektroniky
Datum obhajoby VŠKP: _____

VŠKP odevzdal autor nabyvateli*:

- v tištěné formě – počet exemplářů: 2
- v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy
(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 28. května 2010

.....
Nabyvatel

.....
Autor

ABSTRAKT

Práca sa zaoberá špecifikáciou USB zbernice, jej výhodami, nevýhodami a rôznymi možnosťami použitia. Čitateľovi ponúka bližší pohľad na procesy neviditeľné bežnému užívateľovi v priebehu inicializácie, prevádzky a správy zariadenia, ich význam, štruktúru a časovanie. Práce tiež rozoberá možnosti pripojenia viac počítačov k USB zariadeniam prostredníctvom prepínačov a venuje sa návrhu takéhoto prepínača.

KLÍČOVÁ SLOVA

USB, výhody, nevýhody, enumerácia, prenosy, viac počítačov, prepínač, tlačiarne

ABSTRACT

This work deals with USB bus specification, its advantages, disadvantages, and different possibilities of use. Readers offers a closer look at the processes invisible to user during startup, operation and management of the device, their meaning, structure and timing. The work also discusses the possibility of connecting more computers to a USB device through switch and is dedicated to design of such a switch.

KEYWORDS

USB, advantages, disadvantages, enumeration, transmissions, more computers, switch, printers

MRNÍK, M. *Tiskový USB přepínač*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2009. 27 s., 1 s. příloh. Bakalářská práce. Vedoucí práce: prof. Ing. Stanislav Hanus, CSc.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma Tiskový USB přepínač 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

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce prof. Ing. Stanislavovi Hanusovi, CSc. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne

.....

(podpis autora)

Obsah

1. Úvod.....	1
2. Základy USB.....	2
3. Topológia.....	2
4. Povinnosti hosta	3
4.1. Detekcia zariadení.....	3
4.2. Správa toku dát	3
4.3. Overovanie chýb	4
4.4. Poskytovanie napájania.....	4
4.5. Výmena dát s pripojenými zariadeniami	4
5. Povinnosti zariadenia.....	4
5.1. Detekcia komunikácie	4
5.2. Odpovedanie na štandardné požiadavky	4
5.3. Kontrola chýb.....	5
5.4. Správa napájania	5
5.5. Výmena dát s host-om	5
6. Prenosy	6
6.1. Rýchlosti prenosov	7
6.2. Endpointy-zdroje a ciele dát	7
6.3. Dátové trubice – spojenie endpointov a hosta.....	8
7. Komunikácia.....	9
7.1. Proces enumerácie.....	9
7.2. Kroky enumerácie	10
7.3. Komunikácia používaná pri vlastnej funkcii zariadenia.....	12
7.4. Veľkosť dát.....	14
7.5. Kódovanie dát	14
8. USB prepínač	14
8.1. Hardwarové riešenie	14
8.1.1. Prvá časť - ovládanie prepínania.....	14
8.1.2. Druhá časť - prepínače	15
8.1.3. Tretia časť – napájanie a uzemnenie.....	15
8.2. Softwarové riešenie	16
8.3. Výber triedy zariadenia	16

8.4.	Princíp komunikácie mikrokontrolérov	17
8.5.	Zdieľaný prístup k pamäťovým médiám	17
9.	Záver	18
10.	Použité informačné zdroje	19
11.	Obsah disku CD	19
	Prílohy	20

Zoznam obrázkov

Obrázek 1:	Topológia USB	3
Obrázek 2:	Schéma prenosových rámcov	6
Obrázek 3:	Názorné rozdelenie rýchlostí na zbernici podľa rýchlostí použitých zariadení	7
Obrázek 4:	Hierarchia prvkov prenosu	13

Zoznam tabuliek

Tabulka 1:	Typy transakcií	8
Tabulka 2:	Maximálne dosiahnuteľné prenosové rýchlosti pre jednotlivé typy prenosov a rýchlosti zbernice	13

1. Úvod

Táto práca sa zaoberá technickou realizáciou štandardu USB a jeho implementácie do zariadení. Pojednáva postupne o všetkých krokoch ktoré je treba vykonať či už automaticky alebo za asistencie užívateľa od pripojenia zariadenia po začatie využívania jeho funkcie. Rozoberá možnosti pripojenia niekoľkých počítačov k viacerým tlačiarňam pomocou rozhrania USB a navrhuje riešenie pre diaľkovo ovládaný prepínač.

Je členená do deviatich celkov z ktorých každý podrobnejšie opisuje vnútorné deje a interakciu medzi jednotlivými prvkami systému. Postupne informuje o výhodách riešenia zariadenia prostredníctvom USB rozhrania, fyzickom prepojení jednotlivých členov, ich vzájomných povinnostiach, spôsobe a vlastnostiach komunikácie medzi nimi. Ďalej rozoberá návrh prepínača, výber vhodných riešení a realizáciu jeho jednotlivých častí.

2. Základy USB

Zbernica USB je rozšíreným a široko podporovaným štandardom pre pripojenie periférií k počítaču. Svoju obľubu si získala jednoduchosťou používania, flexibilitou, nízkou cenou a spotrebou energie. Medzi hlavné výhody patrí:

automatická konfigurácia: operačný systém automaticky rozpozná pripojené zariadenie a načíta potrebný ovládač. Ak tento nie je obsiahnutý v OS, pri prvom pripojení sa od užívateľa požaduje vloženie média s ovládačom, ale okrem toho je inštalácia automatická. Systém sa po inštalácii nemusí reštartovať.

jednoduchosť pripojenia: užívateľ nepotrebuje otvárať skrinku počítača ktorá často býva chránená plombou, ale iba pripojí zariadenie do portu na skrinke. Počet portov sa dá v prípade potreby jednoducho rozšíriť pomocou hubu až na 127.

prepojovacie káble: USB káble majú rôzne koncovky, čo zabraňuje nesprávnemu zapojeniu. Kábel môže byť dlhý až 5 metrov a ak sa použijú huby, zariadenie môže byť vzdialené až 30 metrov od počítača ku ktorému je pripojené.

pripojenie za chodu: zariadenia sa môžu pripájať a odpájať kedykoľvek, či je počítač alebo zariadenie zapnuté alebo nie a to bez rizika poškodenia. Pripojené zariadenie operačný systém detekuje, nastaví a pripraví na používanie.

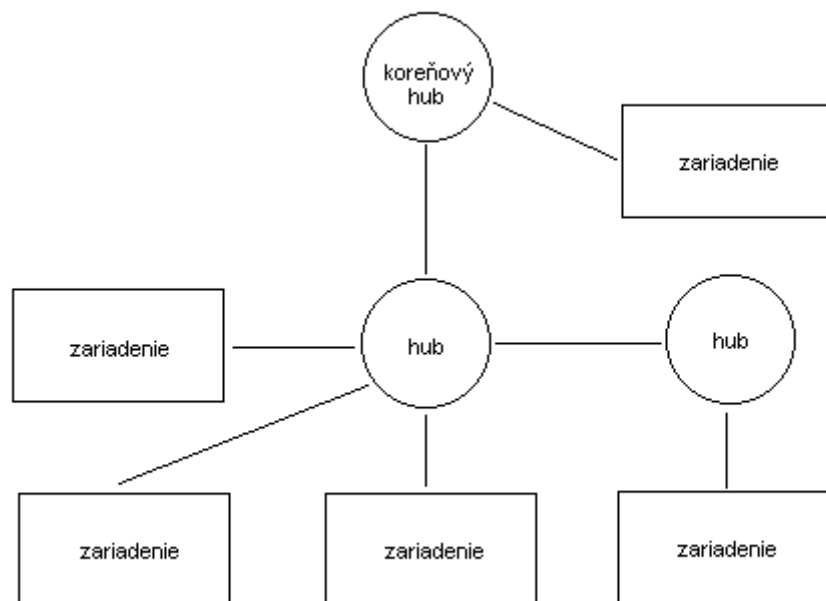
bez nutnosti nastavovania užívateľom: USB zariadenia nemajú žiadne nastavenia ktoré vyžadujú zásah užívateľa ako napríklad adresa portu, linka pre požiadavky na prerušenie (IRQ), preto na ňom nie sú ani jumpre, ani nie je treba spúšťať nastavovacie utility.

uvolňuje hardwarové zdroje: použitie USB pre čo najväčšie množstvo periférií uvoľňuje IRQ linky pre ostatné zariadenia ktoré ich potrebujú. Počítač vyhradí určité adresy pre porty a jednu IRQ linku pre USB kontrolér, ale jednotlivé zariadenia nepotrebujú dodatočné zdroje pre svoj chod.

napájanie: keďže rozhranie USB obsahuje aj napájanie na +5V a uzemnenie, mnohé zariadenia môžu byť napájané výlučne z USB až do 500 mA, čím v porovnaní s ostatnými zbernicami odpadá nutnosť mať ku každému zariadeniu externý zdroj napájania.

3. Topológia

Topológia zbernice USB sa dá nazvať ako stupňovitá hviezda. Stredom každej hviezdy je hub a každý lúč z neho vychádzajúci je zariadenie alebo ďalší hub. Huby majú obvykle 4 či 7 portov. Fyzické umiestnenie prvkov v hviezde nemá vplyv na logické spojenie. Na komunikáciu, hostiteľ ani zariadenie nepotrebujú vedieť cez koľko hubov signály prechádzajú. V niektorých zariadeniach sa nachádzajú aj dvojportové huby a je ich možné zapájať do reťaze. USB hostiteľ alebo v skratke host s nimi pracuje tak isto ako s inými hubmi.



Obrázek 1: Topológia USB

4. Povinnosti hosta

Na komunikáciu s USB zariadeniami, počítač potrebuje hardware a software ktorý mu umožní zastávať funkciu USB hosta. Hardware pozostáva z kontroléra a koreňového hubu s jedným alebo viac portami. Softwarom je operačný systém ktorý poskytuje mechanizmus na komunikáciu ovládačov s USB hardwarem. Host má na starosti zbernicu. Musí vedieť aké zariadenia sú pripojené a aké sú schopnosti každého z nich. Host musí takisto zaistiť, že všetky zariadenia na zbernici môžu prijímať a odosielať dáta podľa potreby v rámci možností a obsadenia zbernice.

4.1. Detekcia zariadení

Pri zapínaní huby informujú host o všetkých pripojených USB zariadeniach. V procese zvanom enumerácia host priradí zariadeniu adresu a získa doplňujúce údaje od každého zariadenia. Kedykoľvek je pripojené nové zariadenie host ho enumeruje a odpojené zariadenie odstráni zo správcu zariadení dostupných aplikáciám.

4.2. Správa toku dát

Host spravuje tok dát na zbernici. Viaceré pripojené zariadenia môžu požadovať prenos dát v rovnaký čas, preto host kontrolér rozdelí dostupný čas do segmentov ktoré sa nazývajú rámce a mikro rámce a každému prenosu prideli určitú časť rámca.

Prenosy, ktoré musia mať zaistenú určitú prenosovú rýchlosť majú garantovaný čas v rámci. V priebehu enumerácie ovládač zariadenia požaduje určitú šírku pásma ktorú bude potrebovať na všetky prenosy ktoré potrebujú garantovanú prenosovú rýchlosť. Ak nie je táto šírka pásma dostupná, host nepovolí začatie komunikácie. Ovládač musí buď požiadať o menšiu šírku pásma alebo čakať kým nie je požadované prenosové pásmo dostupné. Prenosy, ktoré nemajú garantovaný čas v rámci používajú zostávajúcu časť a ak je celá zbernica obsadená, musia čakať.

4.3.Overovanie chýb

Pri prenose dát, host pridá overovacie bity. Prijímacie zariadenie vykoná kontrolný súčet prijatých dát a porovná výsledok s prijatými overovacími bitmi. Ak sa výsledok nezhoduje, zariadenie nepotvrdí príjem dát a host vie, že ich má odoslať znova. USB podporuje aj typ prenosu ktorý nepovoľuje opätovné odosielanie dát v záujme zachovania konštantnej prenosovej rýchlosti. Host takisto overuje dáta prijaté od pripojených zariadení.

Host môže prijať aj iné informácie, nie len že zariadenie nemôže prijať alebo odoslať dáta. V takom prípade potom môže informovať ovládač zariadenia o probléme, ten môže upozorniť aplikáciu a tá môže vykonať zodpovedajúcu akciu.

4.4.Poskytovanie napájania

Spolu s dvoma signálovými vodičmi má USB kábel vodič s napájaním +5V a uzemňovací vodič. Niektoré zariadenia odoberajú všetku potrebnú energiu z týchto vodičov. Host poskytne energiu všetkým zariadeniam pri zapnutí alebo pripojení a prikazuje zariadeniam znížiť odber vždy keď je to možné. Zariadeniam s vysokým odberom môže poskytnúť až do 500 mA. Ak je host napájaný z batérií alebo nepodporuje vysoko odberové zariadenia, môže byť dodávaná energia obmedzená na 100 mA. Zariadenia ktoré požadujú vyšší odber musia mať vlastný zdroj napájania.

4.5.Výmena dát s pripojenými zariadeniami

Všetky vyššie popísané činnosti podporujú hlavnú úlohu hosta, výmenu dát s periférnymi zariadeniami. V niektorých prípadoch ovládač zariadenia ovládač zariadenia požaduje výmenu dát v špecifikovaných intervaloch, kým iné zariadenia komunikujú len keď aplikácia alebo ovládač požaduje. V prípade problémov ovládač zariadenia informuje obslužnú aplikáciu.

5.Povinnosti zariadenia

Z mnohých hľadísk, povinnosti zariadenia sú opačné k povinnostiam hosta. Keď host začne komunikáciu, zariadenie musí odpovedať. Zariadenia majú aj určité unikátne povinnosti. Zariadenie nemôže začať komunikáciu samé od seba, musí čakať a odpovedať na požiadavky hosta. USB kotrolér v zariadení vykonáva mnohé z povinností na úrovni hardwaru, miera softwarovej podpory závisí od použitého kontroléra. Zariadenie musí vykonávať všetky nasledujúce úlohy.

5.1.Detekcia komunikácie

Každé zariadenie sleduje adresu zariadenia obsiahnutú v každej komunikácii na zbernici. Ak adresa nezodpovedá adrese zariadeniu pridelenej, zariadenie komunikáciu ignoruje. Ak sa zhoduje, zariadenie uloží dáta do prijímacieho buffera a spustí prerušenie aby indikovalo, že dáta dorazili. Takmer vo všetkých čipoch sú tieto funkcie zabudované v hardwari a nepotrebujú softwarovú podporu.

5.2.Odpovedanie na štandardné požiadavky

Pri zapnutí alebo pri pripojení zariadenia k zapnutému počítaču, zariadenie musí odpovedať na štandardné požiadavky vyslané hostom počas enumerácie. Host ale môže posilať požiadavky kedykoľvek po skončení enumerácie.

Všetky zariadenia musia odpovedať na tieto požiadavky, zahŕňajúc dotazy na schopnosti a stav, alebo požiadavky na vykonanie určitej akcie. Po prijatí požiadavku, zariadenie umiestni dáta alebo informácie o stave do prenosového buffera na odoslanie k hostovi. Pri niektorých požiadavkách ako napríklad výber konfigurácie, zariadenie vykoná aj inú akciu okrem odpovedania. USB špecifikácia definuje jedenásť požiadaviek, ale trieda zariadenia alebo samotný predajca môže špecifikovať ďalšie. Zariadenie nemusí vykonať každú požiadavku, musí však na ňu odpovedať v zrozumiteľnej forme. Napríklad, keď host požaduje konfiguráciu, ktorú zariadenie nepodporuje, musí odpovedať kódom ktorý znamená že daná konfigurácia nie je podporovaná.

5.3.Kontrola chýb

Podobne ako host, zariadenie vkladá overovacie bity do vysielaných dát. Pri prijme dát vykonáva kontrolný súčet. Podľa toho či zariadenie potvrdí prenos alebo nepotvrdí host vie či poslať dáta znova. Tieto funkcie sú obvykle zabudované v hardwari kontroléru a nie je ich treba programovať.

5.4.Správa napájania

Zariadenie môže byť napájané zo zbernice alebo mať vlastné napájanie. Tie ktoré sú napájané zo zbernice keď nevykonávajú žiadnu aktivitu musia obmedziť svoju spotrebu prúdu na minimum. Keď host vstúpi do stavu nízkeho odberu, všetka komunikácia na zbernici končí včetně periodických časových značiek ktoré host posielal v normálnom režime. Pri detekovaní chýbajúcej aktivity na zbernici dlhšie ako tri milisekundy, zariadenie musí vstúpiť do stavu nízkeho odberu a obmedziť svoj odber prúdu. Zároveň ale stále musí sledovať zbernicu a vrátiť sa do normálneho režimu pri obnovení aktivity na zbernici.

Zariadenia ktoré nepodporujú funkciu vzdialeného zobudenia by nemali v stave nízkeho odberu spotrebovať viac než 500 mikroampérov. Tie ktoré funkciu vzdialeného zobudenia podporujú a zároveň host povolil túto funkciu môžu odoberať maximálne 2,5 miliampéra. Tieto hodnoty sú priemerné za jednu sekundu, špičkové hodnoty môžu byť vyššie.

5.5.Výmena dát s host-om

Všetky vyššie spomenuté povinnosti podporujú hlavnú úlohu USB portu zariadenia a to výmenu dát s hostom. Po nakonfigurovaní, zariadenie musí odpovedať na komunikáciu, ktorá môže obsahovať dáta a zároveň požadovať dáta alebo iné informácie. Schopnosti zariadenia, ovládač hosta a aplikácia ktorá zariadenie používa určuje typ komunikácie a aj to kedy sa vyskytuje.

Pri väčšine prenosov, kedy host posielal dáta do zariadenia, zariadenie musí odpovedať na každý pokus o prenos poslaním kódu ktorý vyjadruje či zariadenie dáta prijalo alebo bolo zaneprázdnené. Pri väčšine prenosov kedy zariadenie posielal dáta hostovi, zariadenie musí na požiadavku o prenos odpovedať zaslaním príslušných dát alebo informáciou, že žiadne dáta na odoslanie nie sú alebo zariadenie je zaneprázdnené. Obvykle hardware odpovedá automaticky, podľa nastavenia vo firmware. Niektoré prenosy nepoužívajú potvrdenie prijímu a odosielateľ predpokladá že príjemca prijal všetky prenášané dáta. Hardware kontroléru sa takisto stará o formátovanie dát pre zbernicu. To zahŕňa pridávanie overovacích bitov, vykonávanie kontrolného súčtu a posielanie a prijímanie jednotlivých bitov po zbernici.

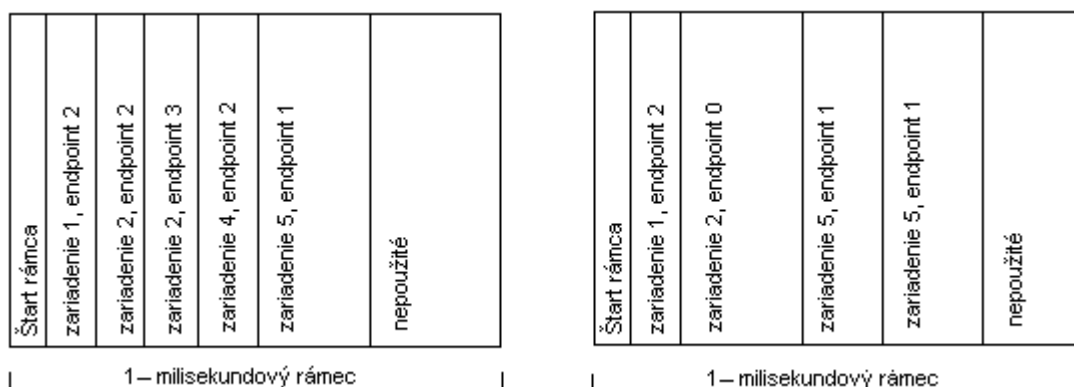
Okrem toho zariadenie musí vykonávať svoju funkciu pre ktorú si ju zákazník zakúpil.

6. Prenosy

Signálové linky prenášajú dáta z a do všetkých zariadení na zbernici a tvoria jednu prenosovú cestu ktorú musia všetky zariadenia zdieľať. Na rozdiel od RS-232 ktoré má TX linku na prenos dát jedným smerom a RX na prenos druhým smerom, dvojica USB liniek prenáša jeden diferenčný signál ktorého smer sa strieda.

Host sa snaží zaistiť aby sa všetky prenosy uskutočnili čo možno najrýchlejšie. Spravuje tok dát rozdelením času na intervaly nazývané rámce (pri použití low a full speed) a mikrorámce (pri použití high speed). Host priradí časť každého rámca alebo mikrorámca každému prenosu. Rámce majú periódu jednu milisekundu. Pre high-speed prenosy host rozdelí každý rámec do ôsmich 125 mikrosekundových mikrorámcov. Každý rámec alebo mikrorámec začína so Start-of-frame časovou referenciou, čo je referencia ktorú host posiela v milisekundových intervaloch pri full-speed a 125 mikrosekundových intervaloch pri high-speed. Táto Start-of-frame referencia udržuje zariadenia od vstúpenia do pozastaveného stavu keď na zbernici nie je iná aktivita. Low-speed zariadenia nevidia Start-of-frame. Namiesto toho, hub ku ktorému je zariadenie pripojené používa jednoduchšie End-of-packet signály nazývané aj low-speed udržiavajúce signály, posielané jeden krát v rámci. Tak isto ako Start-of-frame aj End-of-packet udržuje low-speed zariadenia od vstúpenia do pozastaveného stavu.

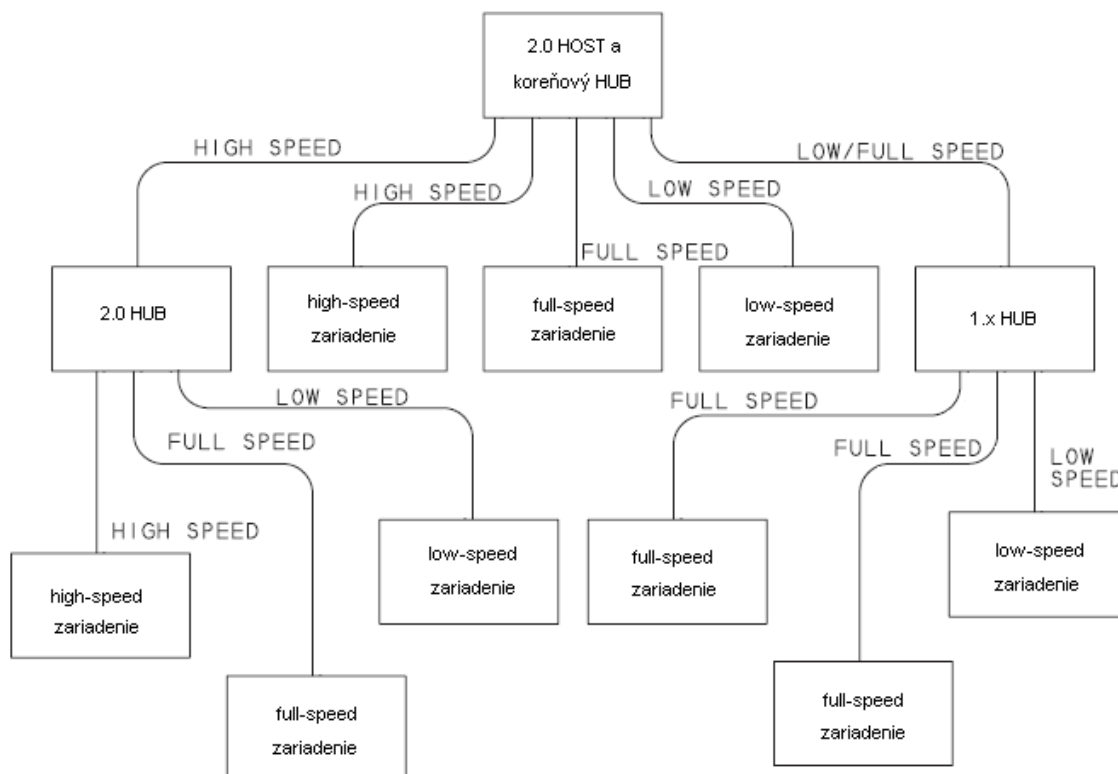
Každý prenos pozostáva z jednej alebo viacerých transakcií. Kontrolné prenosy majú vždy viacero transakcií, pretože majú niekoľko etáp z ktorých každá sa skladá z jednej alebo viacerých transakcií. Ostatné prenosy majú viac etáp iba ak je potrebné preniesť viac dát ako sa zmestí do jednej transakcie. Podľa toho ako host naplánuje transakcie a podľa rýchlosti odpovedi zariadenia, môžu byť prenosové transakcie v jednom rámci či mikrorámci alebo môžu byť rozdelené medzi viaceré (mikro)rámce.



Obrázek 2: Schéma prenosových rámcov

6.1. Rýchlosti prenosov

Prenosy na úseku zbernice sú high-speed iba v prípade, že zariadenie podporuje high-speed a host a všetky huby medzi ním a zariadením vyhovujú USB 2.0 špecifikácii. V prípade, že na takúto zbernicu pripojíme USB 1.x hub, časť zbernice pod týmto hubom bude full-speed alebo low-speed.



Obrázek 3: Názorné rozdelenie rýchlostí na zbernici podľa rýchlostí použitých zariadení

Prenosy z alebo do zariadení ktoré sú full a low speed sa na úsekoch zbernice medzi hostom a high-speed hubom prenášajú rýchlosťou high-speed. Prenosy medzi 2.0 hubom a 1.x hubom alebo iným full alebo low speed zariadením sa prenášajú rýchlosťou full alebo low speed. Zbernica ktorej host je iba 1.x podporuje iba full a low speed aj keby mala pripojené 2.0 huby a high-speed kompatibilné zariadenia.

6.2. Endpointy-zdroje a ciele dát

Pretože všetka výmena dát prebieha prostredníctvom jedného kanálu, každý prenos musí obsahovať adresu zariadenia ktorá jednoznačne určuje cieľ dát. Každé zariadenie má jedinečnú adresu priradenú hostom a všetky dáta idú smerom z alebo do hosta. Každý prenos začína tým, že host vyšle blok dát obsahujúci adresu prijímajúceho zariadenia a špecifické umiestnenie v rámci zariadenia ktoré sa nazýva endpoint.

Všetky prenosy na zbernici idú z alebo do endpointu. Je to zásobník ktorý uchováva viacero bytov. Obvykle je to blok pamäte alebo register na čipe kontroléra. Dáta uložené v endpointe môžu byť prijaté alebo čakajúce na odoslanie. Host má tiež zásobník dát na prijaté a odosielané dáta ale nie je to endpoint. Miesto toho host slúži ako štart a cieľ pre komunikáciu s endpointami zariadení.

Adresa endpointu pozostáva z jeho čísla a smeru. Číslo je hodnota od 0 do 15. Smer je definovaný z perspektívy hosta. IN-vstupný endpoint poskytuje dáta na odoslanie hostovi a OUT-výstupný endpoint ukladá dáta prijaté. Endpoint konfigurovaný pre kontrolné prenosy musí prenášať dáta v oboch smeroch, takže v skutočnosti pozostáva z páru vstupnej a výstupnej adresy ktorá zdieľa jedno číslo endpointu. Každé zariadenie musí mať endpoint 0 nastavený ako kontrolný.

Ostatné typy prenosov posielajú dáta len jedným smerom ale stavové a kontrolné informácie môžu ísť opačným smerom. Jedno číslo endpointu môže mať vstupnú aj výstupnú adresu. Napríklad zariadenie môže mať endpoint 1 IN adresu pre posielanie dát hostovi a endpoint 1 OUT adresu pre príjem dát.

Okrem endpointu 0, full a high speed zariadenie môže mať spolu až 30 endpointov (od 1 do 15, každý podporujúc dva smery IN a OUT). Low speed zariadenie je obmedzené na dva dodatočné endpointy v akejkolvek kombinácii smerov (napríklad ep 1 IN a ep 1 OUT alebo ep 1 IN a ep 2 IN). Každý prenos na zbernici začína paketom obsahujúcim adresu zariadenia, číslo endpointu, kód ktorý indikuje smer toku dát a informáciu či je alebo nie je inicializovaný kontrolný prenos. Kódy sú IN, OUT a Setup.

Typ transakcie	Zdroj dát	Typy prenosov, ktoré používajú tento typ	Obsah
IN	zariadenie	všetky	dáta alebo informácie o stave
OUT	host	všetky	
Setup	host	kontrolné prenosy	požiadavka

Tabuľka 1: Typy transakcií

Tak isto ako pri smere endpointov, pomenovanie pre IN a OUT transakcie vychádza z perspektívy hosta. Pri IN transakcii dáta idú smerom zo zariadenia k hostovi a pri OUT transakcii je to naopak. Transakcia Setup je ako OUT, pretože dáta idú od hosta k zariadeniu, ale Setup transakcia je špeciálna lebo zahajuje kontrolný prenos. Zariadenia musia vedieť identifikovať Setup transakcie, lebo tieto sú jediným typom, ktorý musia vždy prijať a odpovedať na ne. Akýkoľvek iný prenos môže použiť IN alebo OUT transakcie.

Každá transakcia obsahuje adresu zariadenia a adresu endpointu. Keď zariadenie prijme OUT alebo Setup paket obsahujúci jeho adresu, endpoint uloží dáta ktoré nasledujú a hardware obvykle spustí prerušenie. Následne obslužná rutina pre prerušenie môže spracovať prijaté dáta a vykonať požadovanú akciu. Keď zariadenie prijme IN paket obsahujúci jeho adresu, tak ak má dáta čakajúce na odoslanie, hardware ich pošle zo špecifikovaného endpointu na zbernicu a obvykle spustí prerušenie. Obslužná rutina pre prerušenie potom urobí všetko čo je potrebné aby bolo zariadenie pripravené na ďalšiu IN transakciu.

6.3. Dátové trubice – spojenie endpointov a hosta

Predtým, než sa môže uskutočniť prenos, medzi hostom a zariadením sa musí vytvoriť dátová trubica. Je to pomenovanie pre spojenie medzi endpointom zariadenia a softwarom host kontroléra. Host zriadi trubicu počas enumerácie. Ak odoberieme zariadenie zo zbernice, host nepotrebné trubice odstráni. Môže takisto vytvoriť nové alebo odstrániť nepotrebné aj inokedy pri požadovaní inej konfigurácie alebo iného rozhrania zariadenia. Každé zariadenie má štandardnú kontrolnú trubicu ktorá používa endpoint 0.

Konfiguračné informácie, ktoré host získa zahŕňajú deskriptor pre každý endpoint ktorý chce zariadenie používať. Tento deskriptor je blok informácií z ktorých host vie všetko čo potrebuje o endpointe aby s ním mohol komunikovať. Zahŕňa adresu endpointu, typ používaného prenosu, maximálnu veľkosť dátových paketov a v prípade potreby aj želaný interval pre prenosy.

Pri podrobnejšom klasifikovaní trubíc ich môžeme rozdeliť podľa toho či informácia ide jedným alebo oboma smermi na stream a message, čiže s tokom dát alebo určitými správami.

6.3.1. Kontrolné prenosy používajú obojsmerné trubice so správami (message pipes).

Každý začína nastavovacou Setup transakciou obsahujúcou požiadavku. Na dokončenie prenosu si host a zariadenie vymenia dáta a stavové informácie, alebo len stavové informácie. Každý kontrolný prenos má minimálne jednu transakciu ktorá posielala informácie v každom smere.

6.3.2. Všetky ostatné prenosy používajú trubice s tokom dát. V takejto trubici dáta nemajú definovanú štruktúru. Prijímacia strana (host alebo zariadenie) prijme všetko čo príde. Firmware zariadenia alebo software hosta potom môže dáta spracovať akýmkoľvek vhodným spôsobom. Samozrejme aj pri toku dát sa obe strany musia dohodnúť na nejakom formáte. Napríklad, aplikácia môže požadovať zaslanie série bytov indikujúcich teplotu a čas. Mohol byť použitý aj kontrolný prenos s výrobcom definovanou požiadavkou Get_temperature, ale prerušovaný (interrupt) prenos môže byť vhodnejší kôli jeho garantovanej šírke pásma.

7. Komunikácia

USB komunikáciu môžeme rozdeliť do dvoch kategórií. Komunikácia používaná na enumeráciu zariadenia a komunikácia používaná na vlastnú funkciu zariadenia.

Predtým ako obslužná aplikácia môže komunikovať s pripojeným zariadením, host potrebuje získať od zariadenia informácie, nastaviť potrebné parametre a priradiť ovládač. Enumerácia je proces ktorý zahŕňa všetky tieto úlohy. Patrí medzi ne priradenie adresy zariadeniu, prečítanie deskriptorov, priradenie a načítanie ovládača, výber konfigurácie ktorú zodpovedá energetickým nárokom zariadenia, endpointom a ostatným funkciami. Potom je zariadenie pripravené prenášať dáta cez ktorýkoľvek endpoint v jeho konfigurácii.

7.1. Proces enumerácie

Jednou z povinností hubu je detekovanie pripojenia a odpojenia zariadení. Každý hub má prerušenie IN endpoint pre hlásenie takejto udalosti hostovi. Pri štarte systému host zisťuje od svojho koreňového hubu či sú pripojené nejaké zariadenia, včetne prídavných hubov a zariadení pripojených na ne. Po štarte sa host periodicky dotazuje na akékoľvek nové pripojené alebo odpojené zariadenia.

Pri zistení nového zariadenia host pošle hubu na ktorý je zariadenie pripojené sériu požiadaviek ktorými sa vytvorí komunikačná cesta medzi hostom a zariadením. Host sa potom pokúsi o enumeráciu zariadenia posielaním kontrolných prenosov obsahujúcich štandardné USB požiadavky na endpoint 0 daného zariadenia. Všetky USB zariadenia musia podporovať kontrolné prenosy,

štandardné požiadavky a endpoint 0. Pre úspešnú enumeráciu zariadenie musí odpovedať na každú požiadavku zaslaním žiadaných informácií alebo vykonaním požadovanej akcie.

Z používateľského hľadiska je enumerácia neviditeľná a automatická okrem oznamu hlásiaceho že bolo pripojené nové zariadenie a či bola jeho inštalácia úspešná. Pri prvom použití zariadenia pre ktoré nemá operačný systém ovládače je potrebný zásah užívateľa a ich poskytnutie. Keď je enumerácia kompletná, operačný systém pridá zariadenie do správcu zariadení a takisto ho odtiaľ odoberie ak užívateľ zariadenie odpojí.

7.2.Kroky enumerácie

Špecifikácia USB definuje šesť stavov zariadenia. Počas enumerácie sa zariadenie dostáva cez nasledovné: Napájané, predvolené, adresované a konfigurované. Ďalšie dva sú pripojené a pozastavené. V každom stave má zariadenie definované schopnosti a správanie.

Nasledovné kroky sú v typickom poradí v akom prebieha enumerácia pod Windows. Firmware zariadenia však nesmie predpokladať že požiadavky a stavy budú nasledovať v špecifickom poradí. Zariadenie musí detekovať a odpovedať na všetky kontrolné požiadavky alebo iné deje na zbernici kedykoľvek.

1. Používateľ pripojí zariadenie na USB port, alebo sa systém zapne so zariadením už pripojeným. Port môže byť na koreňovom hube alebo na hube pripojenom naň. Hub poskytne zariadeniu energiu a zariadenie je v napájanom stave.
2. Detekcia zariadenia. Hub monitoruje napätie na oboch signálových linkách každého portu a má pull-down rezistor od 14,25 do 24,8 k Ω na každej z nich (D+ aj D-). Zariadenie má pull-up rezistor od 900 do 1575 Ω na buď D+ pre full-speed zariadenie alebo na D- pre low-speed zariadenie. Zariadenia ktoré majú zabudovanú podporu high-speed sa pripájajú ako full-speed. Keď sa zariadenie pripojí do portu, jeho pull-up rezistor privedie linku na vysokú úroveň a hub ho detekuje.
3. Host sa dozvie o novom zariadení. Každý hub používa svoje prerušenie endpoint na hlásenie nových udalostí. Hlásenie sa objaví iba ak hub alebo port (a ak áno tak aký port) zaznamenal nejakú udalosť. Ak sa nejaká stane, host vyšle do hubu požiadavku Get_Port_Status aby zistil viac. Get_Port_Status je štandardná požiadavka triedy hub ktorú všetky huby podporujú. Návratová informácia povie hostovi či bolo pripojené nové zariadenie.
4. Hub detekuje či je zariadenie low alebo high speed. Tesne predtým ako hub resetuje zariadenie, určuje či je zariadenie low speed alebo full speed preskúmaním napätí na signálových linkách. Určuje to podľa toho ktorá signálová linka má vyššie napätie pri nečinnosti. Hub pošle túto informáciu hostovi ako odpoveď na nasledujúcu Get_Port_Status požiadavku. Hub 1.x detekuje rýchlosť zariadenia po resete zbernice. Hub 2.0 detekuje pred resetom aby vedel, či má overiť kompatibilitu zariadenia s high-speed počas resetu ako bude popísané nižšie.
5. Hub resetuje zariadenie. Keď host zistí nové zariadenie, vyšle hubu požiadavku na Set_Port_Feature ktorá resetuje port. Reset je špeciálny stav kedy sú obe signálové linky D+ aj D- na nízkej logickej úrovni. Hub drží tento stav minimálne 10 milisekúnd. Hub resetuje iba nové zariadenie, ostatné huby a zariadenie na zbernici rest neovplyvní.
6. Host zisťuje či zariadenie full-speed podporuje high-speed.

Na toto sa používa metóda cvrlikania (chirp). Sú definované dva stavy, chirp J a chirp K ako diferenčné jednosmerné napätie. V stave chirp J je linka D+ kladnejšia ako D- a v stave chirp K je D- kladnejšia ako D+. Počas resetu, zariadenie podporujúce high-speed vyšle chirp K. Hub schopný rozpoznať high-speed zariadenie odpovie sériou meniacich sa chirp K a chirp J. Ak zariadenie rozpozna vzorec KJKJKJ, prejde z full-speed komunikácie na high-speed. Ak hub neodpovie na vyslaný chirp K, zariadenie vie, že musí pokračovať v komunikácii rýchlosťou full-speed, preto všetky high-speed zariadenia musia byť schopné odpovedať na enumeráciu vo full-speed.

7. Hub vytvorí signálovú cestu medzi zariadením a zbernicou. Host overuje, či sa zariadenie už resetovalo poslaním požiadavky `Get_Port_Status`. Bit v návratových dátach označuje či zariadenie skončilo reset. Ak je potrebné, host opakuje požiadavku kým zariadenie reset skončí. Keď hub odstráni reset, zariadenie je v predvolenom stave. Registre zariadenia sú resetované a zariadenie je pripravené odpovedať na kontrolné prenosy na endpoint 0. Zariadenie môže zo zbernice odoberať do 100 miliampérov.
8. Host vyšle požiadavku `Get_Descriptor` na zistenie maximálnej veľkosti paketu pre štandardnú dátovú cestu. Host vyšle požiadavku na adresu 0 endpoint 0. Pretože host v rovnaký čas enumeruje iba jedno zariadenie, iba jedno zariadenie odpovie na komunikáciu adresovanú na adresu 0 aj keby bolo pripojených viacero nových zariadení súčasne. Ôsmy byte deskriptoru zariadenia obsahuje informáciu o maximálnej veľkosti paketov podporovaných endpointom 0. Windows host si vyžiada 64 bytov, ale po prijatí prvého paketu (bez ohľadu na to či má 64 bytov alebo nie) host začne status fázu prenosu. Po dokončení status fázy, Windows host požiada hub o reset zariadenia tak ako v bode 5, aj keď špecifikácia USB nevyžaduje reset v tomto kroku. Reset je iba prevencia na zaistenie, že zariadenie bude v známom stave po skončení resetu.
9. Host priradí zariadeniu vyslaním požiadavky `Set_Address` unikátnu adresu. Zariadenie dokončí status fázu požiadavky s pôvodnou adresou a potom začne používať novú. Zariadenie je teraz v adresovanom stave. Od tohto bodu sa všetka komunikácia smeruje na novú adresu. Tá je platná kým je zariadenie pripojené, kým nie je vykonaný reset portu alebo sa nerešartuje celý systém. Pri ďalšej enumerácii môže byť zariadeniu pridelená iná adresa.
10. Host zistí schopnosti zariadenia. Vyšle požiadavku `Get_Descriptor` na novú adresu a tentokrát prijme celý deskriptor. Deskriptor je dátová štruktúra obsahujúca maximálnu veľkosť paketu pre endpoint 0, počet konfigurácií ktoré zariadenie podporuje a ostatné základné informácie. Všetky tieto potom použije v nasledujúcej komunikácii. Host potom pokračuje v zisťovaní informácií o zariadení požadovaním jedného alebo viacerých konfiguračných deskriptorov špecifikovaných v deskriptore zariadenia.
11. Host priradí a načíta ovládač zariadenia (okrem združených zariadení). Po zistení informácií o zariadení z jeho deskriptorov, host hľadá najlepší ovládač pre zariadenie. Pri výbere sa Windows snaží o najlepšiu zhodu výrobcovho ID, ID produktu alebo čísla vydania so svojimi INF súbormi. Ak sa v ničom nezhoduje, pokračuje v hľadaní zhody s akoukoľvek triedou, podtriedou a protokolom zariadenia. Ak bolo zariadenie niekedy predtým úspešne enumerované, Windows použije informácie z registrov namiesto hľadania v INF súboroch. Potom ako operačný systém priradí a načíta príslušný ovládač, ten si môže vyžiadať preposlanie deskriptorov alebo iných triedovo-špecifických deskriptorov. Výnimkou z tohto postupu sú združené zariadenia, ktoré môžu mať rozdielne ovládače priradené rozdielnym

rozhraniam v konfigurácií. Host môže priradiť tieto ovládače iba ak sú rozhrania povolené, čo vyžaduje aby bolo zariadenie nakonfigurované.

12. Ovládač zvolí konfiguráciu. Po zistení potrebných informácií z deskriptorov zariadenia si ovládač vyžiada konfiguráciu poslaním požiadavky Set_Configuration s želaným číslom konfigurácie. Niektoré zariadenia podporujú iba jednu konfiguráciu. Ak má zariadenie viacero konfigurácií, ovládač sa môže rozhodnúť ktorú si vyžiada na základe informácií ktoré má o tom ako bude zariadenie využívané, alebo sa môže pýtať užívateľa, alebo zvoliť prvú. Zariadenie prijme požiadavku a nastaví želanú konfiguráciu. Zariadenie je v nakonfigurovanom stave a rozhrania zariadenia sú povolené.

Pre združené zariadenia host v tomto bode priradí ovládače. Tak ako s inými, podľa informácií získaných od zariadenia host nájde príslušný ovládač pre každé rozhranie v konfigurácii. Zariadenie je pripravené na používanie.

Dva zostávajúce stavy sú pripojené a pozastavené.

Pripojený stav – Ak hub na linke Vbus neposkytuje energiu pre zariadenie, je v pripojenom stave. Môže to nastať ak hub zistil príliš veľký odber prúdu alebo ak host požiadal o odpojenie energie od portu. Bez energie na Vbus linke zariadenie nemôže s hostom komunikovať, čiže ako keby nebolo vôbec pripojené.

Pozastavený stav – Zariadenie vstupuje do pozastaveného stavu potom ako nie je na zbernici žiadna aktivita dlhšie ako 3 milisekundy. V tomto stave musí obmedziť svoj odber energie. Pozastavený stav musia podporovať nakonfigurované aj nenakonfigurované zariadenia.

Enumerácia hubu

Huby sú tiež USB zariadenia a host enumeruje pripojený hub tak isto ako iné zariadenia. Ak má hub pripojené ďalšie zariadenia, host enumeruje každé z nich po tom ako ho hub informuje o ich prítomnosti.

Odpojenie zariadenia

Keď používateľ odstráni zariadenie zo zbernice, hub zakáže port zariadenia. Host zistí že zariadenie bolo odstránené keď dotazuje hub, zistí že sa stala nejaká udalosť a vyslaním požiadavky Get_Port_Status zisťuje aká udalosť to bola. Windows odstráni zariadenie zo Správcu zariadení a jeho adresa je použiteľná pre novo pripojené zariadenie.

7.3.Komunikácia používaná pri vlastnej funkcii zariadenia

USB je dizajnované pre zvládnutie rôznych typov periférií s odlišnými nárokmi na prenosovú rýchlosť, čas odozvy a korekciu chýb. Existujú štyri typy dátových prenosov, každý má iné parametre a každý inak poslúži pre rozdielne použitie a účel zariadenia.

7.3.1.**Kontrolné (control) prenosy** sú jediným typom ktorého funkciu definuje USB špecifikácia. Kontrolné prenosy dovoľujú hostovi zistiť informácie o zariadení, nastavenie jeho adresy, výber jeho konfigurácie a iné nastavenia.

7.3.2.**Hromadné (bulk) prenosy** sú zamerané na situáciu kedy prenosová rýchlosť nie je kritická, ako napríklad posielanie dát pre tlačiareň, príjem dát od scanneru alebo prístup k súborom na disku. Pre tieto aplikácie sú rýchle prenosy vítané, ale dáta môžu

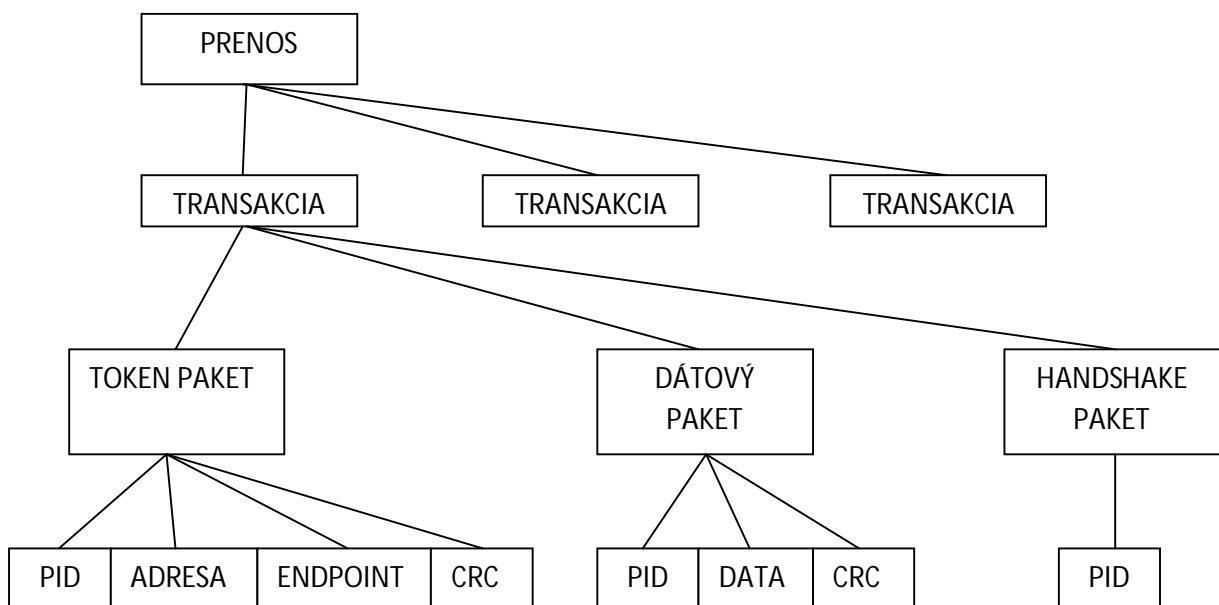
v prípade potreby počkať. Ak je zbernica obsadená, hromadné prenosy sú odložené ale v prípade že nie je sú veľmi rýchle. Iba high-speed a full-speed zariadenia podporujú hromadné prenosy. V globále sa od nich podpora hromadných prenosov nevyžaduje, ale špecifická trieda zariadení ho môže potrebovať.

7.3.3. **Prerušované (interrupt) prenosy** sú pre zariadenia, ktoré potrebujú periodickú pozornosť hosta. Na rozdiel od kontrolných prenosov, prerušovaný prenos je jediný, ktorý môžu využiť low-speed zariadenia. Klávesnice a myši používajú tento prenos na periodické odosielanie dát o stlačených klávesách alebo o polohe ukazovateľa. Prerušované prenosy môžu využívať akúkoľvek rýchlosť. Od zariadení sa nevyžaduje podpora prerušovaných prenosov, ale špecifická trieda zariadení ich môže podporovať.

7.3.4. **Izochrónne prenosy** majú garantovaný čas doručenia, ale nemajú opravu chýb. Dáta ktoré využívajú tieto prenosy sú väčšinou audio a video prehrávané v reálnom čase. Izochrónne prenosy sú jediným typom ktoré nepodporujú automatický opätovný prenos chybné prijatých dát, takže príležitostné chyby musia byť akceptovateľné. Používajú ich iba high-speed a full-speed zariadenia. Takisto sa nevyžaduje ich podpora, ale špecifické triedy zariadení môžu izochrónne prenosy vyžadovať.

Typ prenosu	Maximálna prenosová rýchlosť na endpoint (v kilobytoch za sekundu)		
	Low Speed	Full Speed	High Speed
kontrolný	24	832	15 872
prerušovaný	0,8	64	24 456
hromadný	nie je to povolené	1216	53 248
izochrónny		1023	24 546

Tabuľka 2: Maximálne dosiahnuteľné prenosové rýchlosti pre jednotlivé typy prenosov a rýchlosti zbernice



Obrázek 4: Hierarchia prvkov prenosu

Každá transakcia teda zahŕňa identifikáciu, overovanie chýb, informácie o stave, kontrolné informácie a samotnú výmenu dát. Prenos môže pozostávať z viacerých rámcov alebo mikrorámcov, ale transakcia musí prebehnúť bez prerušenia.

USB prenos pozostáva z transakcií. Tá sa skladá z paketov, a ten obsahuje identifikátor paketu (PID) a potom podľa typu paketu obsahuje ďalšie informácie. Token paket obsahuje adresu zariadenia, adresu endpointu a overovacie dáta. Dátový paket obsahuje samotné prenášané dáta a overovacie dáta. Handshake potvrdzovací paket obsahuje iba identifikátor.

7.4. Veľkosť dát

Maximálna veľkosť dátových paketov sa rôzni v závislosti od rýchlosti zariadenia. Pre low-speed je to 8 bytov. Pre full-speed môže maximum byť 8,16,32 alebo 64 bytov. Pre high-speed maximum musí byť 64 bytov. Tieto byty zahŕňajú iba informácie prenášané v dátovom pakete nepočítajúc PID a CRC bity. Všetky dátové pakety okrem posledného musia mať maximálnu veľkosť pre daný endpoint. Maximálna veľkosť paketu pre štandardnú kontrolnú trubicu je v deskriptore zariadenia ktorý host získa počas enumerácie. V niektorých čítacích prenosoch sa môže veľkosť dát odosielaných zariadením líšiť. Ak ich je menej ako požadovaný počet, alebo dokonca násobok maximálnej veľkosti paketu pre endpoint, zariadenie po odoslaní všetkých dát oznámi že už nemá ďalšie dáta na odoslanie tým, že na ďalší prichádzajúci IN transakčný paket odpovie paketom s nulovou dĺžkou.

7.5. Kódovanie dát

Všetky dáta na zbernici sú kódované formátom Non-Return to Zero Inverted (NZRI) s dopĺňaním bitov čo znamená že po šiestich po sebe nasledujúcich jednotkových bitoch sa pridáva nulový bit, a čo zaručuje že prijímač zostane synchronizovaný s vysielateľom bez ďalšej potreby vysielania samostatného hodinového signálu alebo štart a stop bitov s každým bytom. Metóda NZRI namiesto definovania logických stavov 1 a 0 ako úrovně napätia používa pre vyjadrenie logickej 0 zmenu napätia a logickej 1 ako žiadnu zmenu napätia. Prenos je usporiadaný od najmenej významného bitu (LSB).

8. USB prepínač

Primárne určenie prepínača je zdieľanie viacerých tlačiarň medzi viacerými počítačmi. Keďže USB bolo navrhované ako štruktúra jedného hlavného host kontroléra v pozícii správcu zbernice ku ktorému sa pripájajú všetky ostatné zariadenia ako podriadené, nedá sa pre takúto funkciu použiť klasický hub. Úlohou preto bolo nájsť riešenie umožňujúce súčasné pripojenie viacerých host kontrolérov z ktorých každý by mohol ovládať ku ktorému počítaču bude tlačiareň alebo iné USB zariadenie pripojené. Pre pripojenie budú použité dva USB káble, jeden k ovládaniu prepínača a druhý pre výmenu dát s koncovým zariadením.

8.1. Hardwarové riešenie

8.1.1. Prvá časť - ovládanie prepínania

Prepínač je navrhovaný maximálne pre štyri pripojené počítače a štyri zariadenia. Preto sa ovládanie prepínača skladá zo štyroch mikrokontrolérov ATmega8-16PU od výrobcu Atmel. Každý z nich je

k zbernici usb pripojený ako low-speed zariadenie čo zabezpečuje 1,5 kiloohmový rezistor na D-. Dátová linka D+ je cez 68 ohmový rezistor privedená na druhý bit D portu, lebo na ňom má ATmega8 hardwarové prerušenie INT0 s najvyššou prioritou a linka D- je cez rezistor s takou istou hodnotou privedená na štvrtý bit D portu kde je vstup pre čítač/časovač. Týmto spôsobom môže firmware monitorovať aktivitu na D- kde sa vysielajú USB rámcové pulzy kontrolovaním hodnoty čítača v pravidelných intervaloch. Ak nie je zaznamenaná žiadna aktivita, mal by firmware podľa USB špecifikácie prejsť do šetriaceho režimu – pozastaveného stavu. Pull-down rezistor 1 M na linke D+ zaisťuje že nebudú vykonané žiadne prerušenia ak nie je pripojený kábel ku konkrétnemu mikrokontroléru.

Taktovaciú frekvenciu zabezpečuje 12MHz kryštál. Jeho hodnota bola vybratá s ohľadom na dodržanie časovania pre USB prenosy.

Komunikácia jednotlivých mikrokontrolérov je riešená pomocou TWI/I2C. Jeden vybraný mcu ktorý je nastavený ako master a zároveň riadi prepínanie oboch multiplexerov. Linky SDA aj SCL sú cez 4,7 kiloohmové rezistory pripojené na napájanie 5V.

8.1.2. Druhá časť - prepínače

Pre uskutočnenie spojenia medzi počítačom a vybraným zariadením slúžia dva multiplexery MAX 4899AE od spoločnosti Maxim. Tento model má pomer 4:1, vyhovuje špecifikácii až USB 2.0, má nízky prevádzkový prúdový odber 200 μ A a veľmi nízky odber v pohotovostnom režime, iba 3 μ A. Pre ovládanie prepínania má dva vstupy, kde kombinácia logických úrovní určuje aké prepnutie sa má vykonať. Tieto multiplexery v Českej Republike maloobchodne nepredáva nikto, preto som si ich musel objednať ako vzorky priamo z Maximu. Dodáva sa vo vyhotovení TQFN ktoré navyše obsahuje exposed pad zo spodnej strany púzdra ktorý treba uzemniť a jeho osadzovanie na dosku je náročné na zručnosť a vybavenie.

Prepínače sú v zapojení orientované vzájomne oproti sebe (4:1 – 1:4) kedy výstup jedného je vstupom druhého. Toto aj podmieňuje, že súčasne môže byť spojený len jeden počítač a jeden vstupný port. Ich ovládanie má na starosti jeden z mikrokontrolérov.

8.1.3. Tretia časť – napájanie a uzemnenie

Zariadenie je konštruované pre napájanie zo zbernice USB. Aby bolo zaistené napájanie všetkých komponentov v prípade pripojenia kábla k hociktorému konektoru, boli všetky napájacie vstupy spojené a privedené na vstup low-drop stabilizátora LE33-CZ ktorý napätie 5V zníži na 3,3V a tým sú potom napájané všetky mikrokontroléry aj multiplexery. Keďže napätie na pripojených počítačoch nebude vždy rovnaké a bude sa pohybovať v malej odchýlke od 5V, sú všetky vstupy ošetrené schottkyho diódou. Keďže primárne je prepínač určený pre tlačiarne ktoré majú vlastné napájanie zo siete, nepredpokladá sa pripojenie zariadenia ktoré môže vyžadovať prúd až 500 mA. Štandardná hodnota prúdu z USB 100 mA pre napájanie postačuje.

Použitie stabilizátora LE33-CZ má však nevýhodu v tom, že pri šetriacom režime – pozastavenom zariadení nevyhovuje USB špecifikácii pre prúdový odber 500 μ A. Pretože ak 1,5 k Ω pull-up rezistor odoberie 200 μ A a ATmega v stave stand-by odoberia menej ako 1 μ A, zostáva pre stabilizátor menej ako 299 μ A. Pre vyhoveniu štandardom USB by bolo vhodnejšie použiť low-drop regulátor s veľmi nízkym odberom akým je napríklad TPS71533DCK od Texas Instruments, ktorý sa ale obtiažne zháňa a dá sa získať v málo kusoch ako vzorka.

Uzemnenie je riešené spojením všetkých zemí z USB portov a použitím tienených koncoviek.

8.2. Softwarové riešenie

Keďže mikrokontroléry ATmega8 nemajú implementovanú podporu štandardu USB od výroby, bolo ju treba riešiť softwarovo.

Jedna možnosť bola zapojiť medzi mcu a počítač prevodník USB -> UART napríklad FT232 od firmy FTDI. Riešenie by to bolo výhodné z hľadiska firmware, kedy by nebolo potrebné programovať softwarovú podporu USB priamo v mcu. Keďže ale tieto prevodníky sú relatívne drahé (cca 120 až 140 korún) tak vzhľadom na čo možno najnižšiu cenu konštrukcie som sa pre toto riešenie nerozhodol.

Druhá možnosť bola použiť zapojenie Igora Češka IgorPlug, ktoré má však viaceré nevýhody. Medzi najväčšie patrí nezaručená spoľahlivosť, vysoké nároky na skúsenosť návrhára a aj porušenie špecifikácie USB.

Nakoniec som sa rozhodol využiť open source projekt V-USB zo stránok Objective Development www.obdev.at. Na tejto stránke sa dajú nájsť mnohé dobre a zrozumiteľne popísané projekty ktoré využívajú softwarovú implementáciu USB, ktoré mi boli inšpiráciou.

8.3. Výber triedy zariadenia

Pomocou firmwaru od V-USB sa mikrokontrolér môže hlásiť ako rôzne triedy USB. Medzi najpoužívanejšie patrí trieda vlastná (custom class) a trieda HID (human interface device), ktorá sa rozdeľuje na štandardnú a vlastnú. Existuje ešte kombinácia predchádzajúcich dvoch ako špecifikovaná ako vendor type requests pre zariadenie custom HID class.

Zariadenie vlastnej triedy má výhodu v tom, že sa jednoducho implementuje a je jednoduché preň vytvoriť ovládací software na systémoch Unix. Nevýhoda je, že pre Windows je potrebné vytvoriť a nainštalovať ovládač a takisto sa zariadenie dá ovládať iba pomocou dodaného programu.

Zariadenie štandardnej triedy HID nepotrebuje dodávať ovládač, lebo všetky súčasné operačné systémy majú ovládač HID v sebe štandardne inštalovaný. Nevýhoda toho riešenia je, že môže byť použité iba pre zariadenia ako myši, klávesnice, joysticky a podobné.

Trieda HID definuje jazyk pre popis druhu dát odosielaných alebo prijímaných zo zariadenia. Popis obsahuje nielen formát dát, ale aj význam jednotlivých zápisov. Je však možné tieto správy použiť pre príjem alebo posielanie ľubovoľných dát. Rozdiel zariadenia vlastnej triedy HID oproti zariadeniu vlastnej triedy je ten, že tak ako štandardná trieda HID, nepotrebuje ovládač. Nevýhody sú v tom, že keďže je dopredu známy popis prenášaných dát, tie musia mať vždy konštantnú dĺžku a zvýšená komplexnosť programovania.

Keďže prepínač bude zariadenie ku ktorému je nevyhnutné dodať ovládací program, rozhodol som sa ho realizovať ako zariadenie vlastnej triedy (custom class) a používateľ spolu s ovládacím programom bude musieť nainštalovať aj ovládač.

K samotnej realizácii programového vybavenia však nedošlo, dôvody budú objasnené v závere.

8.4.Princíp komunikácie mikrokontrolérov

Mikrokontroléry medzi sebou komunikujú pomocou TWI (two wire interface – pomenovanie firmy Atmel) / I2C (Inter-Integrated Circuit – pomenovanie všeobecné).

Jeden mcu je nakonfigurovaný ako master I2C zbernice a ostatné ako slave. Ten istý mcu aj riadi prepínanie multiplexerov. Základná myšlienka riadenia je nasledovná:

Všetky mcu majú v registroch uloženú určitú počiatočnú hodnotu nastavenia multiplexerov. Master mcu pravidelne vysiela požiadavky na zistenie stavu týchto registrov všetkých slave mcu. Ak niektorý zo slave mcu dostane príkaz na prepnutie multiplexerov, uloží si túto hodnotu do registrov, master mcu zistí zmenu a uloží si nové hodnoty nastavenia do svojich registrov a následne multiplexery prepne podľa požiadavky. Potom uloží novú hodnotu do svojich registrov a pošle všetkým slave mcu nové hodnoty ktoré si majú zapísať do registrov a pokračuje v cyklickom overovaní stavu.

K realizácii tejto myšlienky avšak taktiež nedošlo.

8.5.Zdieľaný prístup k pamäťovým médiám

Zo špecifikácie USB je zrejmé, že zariadenie pripojené na zbernicu môže obsluhovať iba jeden host kontrolér, ktorý ho adresuje, nastaví a pripraví na používanie. Preto prístup k pamäťovým médiám či typu flash alebo mass storage je možný jednotlivo, podľa toho ku ktorému počítaču je prepínač prepnutý. Pri ovládaní z každého počítača a ich odpájaní je ale potrebné pred prepnutím prepínača zariadenie bezpečne odobrať zo systému a taktiež hrozí, že užívateľ prepne prepínač počas zápisu dát, čo môže viesť k ich strate alebo poškodeniu média.

9. Záver

Bolo popísané fungovanie štandardu USB, jeho topológia, jednotlivé prvky zbernice, zariadení a ich vzájomná spolupráca. Boli priblížené formáty, rýchlosti a spôsoby komunikácie a nevyhnutné kroky k jej nadviazaniu medzi hostom a zariadeniami. Bolo navrhnuté zariadenie na zdieľanie viacerých USB tlačiarňí medzi viacerými počítačmi pomocou prepínača ovládaného mikrokontrolérmi.

Počas práce na semestrálnom projekte a následne bakalárskej práce sa vyskytli viaceré problémy. Ako prvý spomeniem nemožnosť osobnej konzultácie semestrálneho projektu kôli zahraničnej stáži môjho pôvodného vedúceho o ktorej som bol informovaný až po prihlásení sa k téme. Mailová komunikácia s ním bola značne zmätená a v odpovedi na môj podrobný mail týkajúci sa otázok ohľadne princípu pracovania celého prepínača mi napísal „Neznám USB specifikaci příliš podrobně.“

Po prepracovaní zapojenia prepínača ku ktorému mala komisia pri prezentácii semestrálneho projektu výhrady a jeho následnej konzultácii na ktorej sa na tri moje otázky týkajúce sa programovania mcu a vhodnosti použitých komponentov pán Hermany odpovedal „nevím“ a po dohode, že niečo zistí a mailom mi odpovie ma poslal za pánom Michalom Fuchsom, ktorý mi poradil, no po mojej druhej alebo tretej návšteve povedal že aby som niektoré veci konzultoval s pánom Hermanym a „ať sa Jirka taky snaží“. Pri otázke o pájkovaní TQFN púzdra, ktoré ako som písal vyššie je vcelku komplikované, som chcel vedieť či a kde je možnosť nechať si púzdro na dosku pripájkovať lebo sám som si na to netrúfal, keďže vraj s tým majú problém aj skúsenejší spolužiaci mi pán Hermany napísal „Co se týká připájení TQFN pouzdra, nemělo by to prý být o mnoho složitější, než pájení SMD.“ Na konzultáciach ktoré som s ním absolvoval sa viac pýtal ako odpovedal na moje otázky a mal som pocit, že ma skúša. Na moju otázku ako má niečo byť odpovedal „Jak by jste to udelal?“.

Pár dní pred tým ako som mal v pláne pána Hermanyho opäť navštíviť mi 30. dubna mailom oznámil, že končí svoje doktorské štúdium a moju bakalársku prácu bude ďalej viesť pán profesor Hanus. V tom čase som už tretí týždeň čakal na vzorky multiplexerov z Maximu (ktoré mi pomohol objednať pán Fuchs a jeho kolega). Po konzultácii s pánom profesorom Hanusom som dal dosku vyrobiť a osadiť multiplexermi. Po zakúpení súčiatoch kde bol najväčší problém zohnať mikrokontroléry ATmega8, ktoré nemal na sklade už dlhšiu dobu žiaden zo známych maloobchodných predajcov a ich následnom osadení som mal pripravený skúšobný program, ktorý sa mal prihlásiť ako USB myš Logitech. Keďže ale zo štyroch mcu s nahratým programom počítač rozpoznal len jeden, strávil som zvyšný čas do odovzdania práce hľadaním chyby v programe alebo zapojení.

10. Použité informačné zdroje

- [1] MRNÍK, Martin. Tiskový USB prepínač. Brno, 2010. 26 s. Semestrální práce. FEKT VUT v Brně.
- [2] KAINKA, B. USB. Měření, řízení a regulace pomocí sběrnice USB. Praha: BEN, 2002. 1. vyd. 247 s. ISBN 80-7300-073-3.
- [3] AXELSON, J. USB complete: everything you need to develop custom USB peripherals. New York: Lakeview Research, 2005. 3rd ed., 572 s. ISBN 978-1-931448-02-4.
- [3] OBJECTIVE DEVELOPMENT Software GmbH. V-USB is a software-only implementation of a low-speed USB device for Atmel's AVR® microcontrollers [online]. 2009 [cit. 2010-05-28]. Dostupné z WWW: <<http://www.obdev.at/products/vusb/index.html>>.
- [4] Embedded Systems Academy [online]. 2010 [cit. 2010-05-28]. I2C Bus Technical Overview and FAQ. Dostupné z WWW: <<http://www.esacademy.com/en/library/technical-articles-and-documents/miscellaneous/i2c-bus.html>>.
- [5] ČEŠKO, Ing. Igor. Igor Cesko's Web Page [online]. 2000-2007 [cit. 2010-05-28]. Implementácia USB do mikrokontroléra: IgorPlug-USB (AVR). Dostupné z WWW: <[http://www.cesko.host.sk/IgorPlugUSB/IgorPlug-USB%20\(AVR\).htm](http://www.cesko.host.sk/IgorPlugUSB/IgorPlug-USB%20(AVR).htm)>.
- [6] Maxim Integrated Products. USB 2.0 High-Speed, Fault-Tolerant 3:1, 4:1 Multiplexers : Datasheet [online], 2006 [cit. 2010-05-28]. Dostupné z WWW: <<http://datasheets.maxim-ic.com/en/ds/MAX4899AE-MAX4899E.pdf>>.
- [7] V-USB : wikidot [online]. 2009 [cit. 2010-05-28]. Dostupné z WWW: <<http://vusb.wikidot.com/start>>.

11. Obsah disku CD

Na kompaktnom disku ktorý je prílohou tejto práce sú súbory:

\eagle – zložka obsahujúca elektronickú verziu schémy zapojenia, dosky plošného spoja a vlastnú knižnicu pre obvod MAX4899AE

\libusb – zložka obsahujúca inštalačné súbory kompilátora

\vusb-20090822 – zložka obsahujúca súbory k open source projektu V-USB

\mousefw – zložka obsahujúca zdrojové aj skompilované súbory skúšobného firmware ktorý sa mal prihlásiť ako myš

mrnik_bp.pdf – táto bakalárska práca

Prílohy

Príloha č.1: Obvodové zapojenie navrhovaného prepínača

