České vysoké učení technické v Praze – Fakulta elektrotechnická
Katedra řídicí techniky

Diplomová práce

Implementace USB Interface pro počítačové periferie

Roman Bartosiński
2003

Abstrakt

Tato diplomová práce se zabývá implementací a programovou podporou rozhraní USB pro mikroprocesorové systémy.

Teoretická část obsahuje výtah ze specifikace USB verze 1.1 a částečně verze 2.0. Součástí je také přehled a popis jednotlivých typů obvodů určených pro implementaci rozhraní USB. Výsledkem rozboru dostupných materiálů o obvodech pro USB je přehledový katalog dostupných obvodů a podrobnější katalog řadičů a mikrokontrolérů s implementací USB. Další část popisuje navrženou vývojovou desku rozhraní USB s vybraným obvodem PDIUSBD12 firmy Philips.

Praktická práce zahrnuje vytvoření programové podpory pro realizované rozhraní v moderních 32 bitových systémech. V textu jsou uvedeny výsledky této práce ve formě popisu jednotlivých programů, které lze nalézt na přiloženém CD.

Abstract

This diploma thesis describes the implementation and software support of a USB interface for microprocessor systems.

The theoretical section contains a simplified description of the USB specification in version 1.1 and partly in version 2.0. As a component part of this is also a list and description of types of USB interface circuits for implementation. The results of the analysis are summarized in the catalogue of available USB circuits and detailed catalogue of microcontrollers with USB interface. The next section describes the proposed USB interface board with selected circuit PDIUSBD12 from Philips.

The practical section includes software package for the support of the USB interface into contemporary 32bits systems. Included in the thesis are all individual programs, which can also be found on the enclosed CD.

Prohlášení

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

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne …….. …..……

Poděkování

Na tomto místě bych rád poděkoval především vedoucímu mé diplomové práce Ing. Pavlu Píšovi, bez jehož rad a času, věnovaných mi při konzultacích, by tato práce nemohla vzniknout. Dále bych chtěl poděkovat svému nejbližšímu okolí za podporu při studiu.

Obsah

1. ÚVOD **1**

2. PRŮMYSLOVÝ STANDARD USB **3**

2.1 HISTORIE USB 3
2.2 SPECIFIKACE USB 4
2.2.1 Fyzická vrstva USB *4*
2.2.1.1 Architektura USB 4
2.2.1.2 Mechanické vlastnosti přenosového média USB 5
2.2.1.3 Elektrické vlastnosti sběrnicového napájení 6
2.2.1.4 Elektrické vlastnosti signálové sběrnice 7
2.2.1.5 Kódování dat 10
2.2.2 Linková vrstva *11*
2.2.2.1 Pakety a jejich pole 11
2.2.2.2 Typy a průběh přenosů 13
2.2.2.3 Přidělování kapacity šířky pásma 14
2.2.3 Řídící přenos *15*
2.2.3.1 Enumerace zařízení 15
2.2.3.2 Deskriptory zařízení 15
2.2.3.3 Struktura požadavku 19
2.2.3.4 Přehled standardních požadavků 19

3. INTEGROVANÉ OBVODY PRO USB **21**

3.1 PRVKY SBĚRNICE USB 21
3.1.1 Hostitel *21*
3.1.2 Rozbočovač *22*
3.1.3 Koncové zařízení *23*
3.2 TYPY OBVODŮ PRO USB 24
3.2.1 Obvody napěťových spínačů *24*
3.2.2 Obvody vysílače a přijímačů *25*
3.2.3 Obvody fyzické vrstvy *25*
3.2.4 Ostatní obvody *26*
3.3 ZVOLENÝ OBVOD PDIUSBD12 26

4. HARDWARE **28**

4.1 DESKA ROZHRANÍ ID12 28
4.1.1 Část CPU *28*
4.1.2 Rozhraní RS232/RS485 ..... *29*
4.1.3 Rozhraní USB *29*
4.1.4 Napájení, konektory a propojky *29*
4.2 SPI PROGRAMÁTOR CPU 30

5. SOFTWAROVÉ VYBAVENÍ **31**

5.1 VRSTVY SOFTWAROVÉHO VYBAVENÍ 31
5.1.1 Programové vybavení na straně zařízení *31*
5.1.2 Programové vybavení na straně hostitele *35*
5.1.3 Popis struktury ovladače pro USB zařízení. *36*

5.2 POUŽITÉ SOFTWAROVÉ NÁSTROJE 38
5.2.1 Nástroje pro programové vybavení zařízení 38
5.2.1.1 Deska iD12 s CPU AT89S5282 38
5.2.1.2 Deska MO_CPU s CPU M68376 39
5.2.2 Nástroje pro programové vybavení hostitele 39
5.2.2.1 Linux 39
5.2.2.2 Windows 2000 39
5.2.2.3 MathWorks MATLAB 39
5.3 VYTVOŘENÝ SOFTWARE 39
5.3.1 SPI (PC-DOS) 40
5.3.2 PDIUSB (iD12, M68) 40
5.3.3 SKELETON (iD12) 40
5.3.4 BOOTLOAD (iD12) 41
5.3.5 USB2COM (iD12) 42
5.3.6 USB (M68) 42
5.3.7 SEND2D12 (PC-WIN, LINUX) 42
5.3.8 UCOM (PC-LINUX) 43
5.3.9 BOOTLOAD (PC-WIN) 43
5.3.10 WLIBUSB (PC-WIN) 43
5.3.11 USB2COM (PC-WIN) 43
5.3.12 EVBOUSB (PC-WIN) 44
5.3.13 USBPORT (PC-WIN,LINUX) 44

6. ZÁVĚR 45

7. LITERATURA 46

8. OBSAH PŘILOŽENÉHO CD 47

9. PŘÍLOHY 1

A. KATALOG OBVODŮ S USB 1
A.1 Přehledový katalog obvodů pro implementaci USB 1
A.1.1 Spínače napětí 1
A.1.2 Blok vysílače a přijímačů 2
A.1.3 Blok fyzické vrstvy (PHY) 2
A.1.4 Řadič hostitele 2
A.1.5 Řadič rozbočovače 2
A.1.6 Koncové zařízení 3
A.1.7 Ostatní obvody 5
A.2 Katalog řadičů a mikrokontrolérů s implementací USB 6
A.3 Seznam výrobců obvodů uvedených v katalogu 18
B. KATALOGOVÝ LIST OBVODU PDIUSBD12 1
C. TECHNICKÁ DOKUMENTACE DESKY ROZHRANÍ USB - ID12 1
C.1 Schéma zapojení iD12 1
C.2 Deska plošných spojů 2
C.3 Rozložení součástek 3
C.4 Rozmístění a popis konektorů a zkratovacích propojek 3

Seznam obrázků

Obrázek 2.1 - Model sběrnice USB. 4
Obrázek 2.2 - Stromová struktura sběrnice USB. 5
Obrázek 2.3 - Průřez kabelem pro USB. 5
Obrázek 2.4 - Typy konektorů a zásuvek pro USB. 6
Obrázek 2.5 - Zařízení USB podle typu napájení. 7
Obrázek 2.6 - Úbytky napětí v nejhorším možném případu. 7
Obrázek 2.7 - Připojení jednotlivých prvků na sběrnici podle specifikace 1.1. 8
Obrázek 2.8 - Šíření signálu mezi zařízeními. 8
Obrázek 2.9 - Průběh signálů sběrnice při připojení/odpojení zařízení. 9
Obrázek 2.10 - Průběh signálů D+ a D- při přenosu paketu. 10
Obrázek 2.11 - Kódování dat pomocí NRZI. 10
Obrázek 2.12 - Příklad dat vysílací části přenosové cesty. 10
Obrázek 2.13 - Přenos po sběrnici v časových rámcích. 11
Obrázek 2.14 - Struktura některých polí, tvořící pakety. 11
Obrázek 2.15 - Složení paketů z jednotlivých polí. 11
Obrázek 2.16 - Příklad průběhu přenosu pro LS zařízení. 13
Obrázek 2.17 - Průběhy jednotlivých typů přenosu. 13
Obrázek 2.18 - Posloupnost operací při řídicím přenosu a přenosu objemných dat. 14
Obrázek 2.19 - Stavy zařízení z hlediska USB. 15
Obrázek 2.20 - Hierarchie deskriptorů – příklad popisu zařízení. 16
Obrázek 2.21 - Pořadí odesílaných deskriptorů spolu s deskriptorem konfigurace. 17
Obrázek 3.1 - Blokové schéma rozhraní USB hostitele s jedním výstupním portem. 21
Obrázek 3.2 - Blokové schéma rozhraní USB rozbočovače napájeného ze sběrnice. 22
Obrázek 3.3 - Blokové schéma rozhraní USB rozbočovače napájeného z externího zdroje. 23
Obrázek 3.4 - Blokové schéma rozhraní USB koncového zařízení. 23
Obrázek 3.5 - Blokové schéma napěťového spínače pro USB. 24
Obrázek 3.6 - Vnitřní schéma vysílače/přijímačů pro LS/FS zařízení. 25
Obrázek 3.7 - Blokové schéma SIE. 25
Obrázek 3.8 - Vnitřní schéma obvodu pro ochranu před elektrostatickým výbojem. 26
Obrázek 3.9 - Blokové schéma obvodu PDIUSBD12. 26
Obrázek 4.1 - Blokové schéma vývojové desky iD12. 28
Obrázek 4.2 - Mapa externího adresového prostoru CPU. 28
Obrázek 4.3 - Schéma zapojení rozhraní USB. 29
Obrázek 4.4 - Schéma zapojení programovacího budiče SPI rozhraní. 30
Obrázek 5.1 - Model sběrnice USB. 31
Obrázek 5.2 - Vrstvy části programu pro rozhraní USB. 31
Obrázek 5.3 - Část vývojového diagramu pro hlavní smyčku s prioritní obsluhou USB. 32
Obrázek 5.4 - Vývojový diagram obsluhy přerušení pro obvod PDIUSBD12. 33
Obrázek 5.5 - Vývojový diagram obsluhy vstupního kanálu řídicí brány. 34
Obrázek 5.6 - Vývojový diagram obsluhy výstupního kanálu řídicí brány. 34
Obrázek 5.7 - Vrstvy programového vybavení pro hostitelský počítač 35
Obrázek 5.8 - Základní struktura ovladače pro zařízení USB. 37
Obrázek C1 - Schéma zapojení desky rozhraní USB iD12. C-1
Obrázek C2 - Pohled ze strany spojů. C-2
Obrázek C3 - Pohled ze strany součástek. C-2
Obrázek C4 - Rozložení součástek na desce iD12. C-3
Obrázek C5 - Rozmístění konektorů, zkratovacích propojek a LED. C-3

Seznam tabulek

Tabulka 2.1 - Základní stavy sběrnice definované na straně zařízení. 9
Tabulka 2.2 - Typy paketů a jejich identifikace. 12
Tabulka 2.3 - Struktura deskriptoru zařízení. 16
Tabulka 2.4 - Struktura deskriptoru konfigurace. 17
Tabulka 2.5 - Struktura deskriptoru rozhraní. 18
Tabulka 2.6 - Struktura deskriptoru brány. 18
Tabulka 2.7 - Deskriptor podporovaných jazyků. 19
Tabulka 2.8 - Deskriptor textového řetězce. 19
Tabulka 2.9 - Struktura požadavku. 19
Tabulka 2.10 - Přehled standardních požadavků. 20
Tabulka A.1 - Přehled obvodů USB – Spínače napětí. A-1
Tabulka A.2 - Přehled obvodů USB – Blok vysílače a přijímačů. A-2
Tabulka A.3 - Přehled obvodů USB – Blok fyzické vrstvy. A-2
Tabulka A.4 - Přehled obvodů USB – Řadič hostitele bez společného zařízení. A-2
Tabulka A.5 - Přehled obvodů USB – Řadič hostitele se společným zařízením. A-2
Tabulka A.6 - Přehled obvodů USB – Řadič rozbočovače bez složeného zařízení. A-2
Tabulka A.7 - Přehled obvodů USB – Řadič rozbočovače se společným zařízením. A-3
Tabulka A.8 - Přehled obvodů USB – Obvod rozhraní mezi USB a MCU. A-3
Tabulka A.9 - Přehled obvodů USB – Mikrokontroléry s USB. A-3
Tabulka A.10 - Přehled obvodů USB – Převodníky USB. A-4
Tabulka A.11 - Přehled obvodů USB – Ostatní obvody. A-5
Tabulka A.12 - Řadiče rozhraní USB. (seřazeno podle výrobce) A-6
Tabulka A.13 - Mikrokontroléry USB. (seřazeno podle obvodu) A-7
Tabulka A.14 - Seznam výrobců uvedených v katalogu. A-18
Tabulka C1 - Popis konektorů na desce iD12. ..... C-4
Tabulka C2 - Popis zkratovacích propojek na desce iD12. ..... C-5

1. Úvod

Univerzální sériová sběrnice (USB – *Universal Serial Bus*) je od druhé poloviny 90.let minulého století novým standardem pro připojování zařízení k počítači a přenos dat mezi nimi. Sběrnice se na trhu prosadila díky vlastnostem, které znamenaly pro uživatele jednoduché použití a pro výrobce levnou a rychlou implementaci do nových i již hotových zařízení. Mezi tyto vlastnosti patří především možnost připojování zařízení za chodu počítače i zařízení, napájení zařízení s menším odběrem přímo ze sběrnice, rychlost přenosu až do 12Mb/s, univerzálnost použití a možnost připojení více zařízení k jedné sběrnici.

Další rozšíření USB na trhu umožnila nová verze specifikace, která dovoluje přenos až do rychlosti 480Mb/s a tím i připojení datově náročnějších zařízení. Navíc s rozvojem mobilní digitální techniky a uvedením dodatku *On-The-Go* k poslední verzi specifikace je možné propojovat některá zařízení přímo, bez potřeby počítače jako hostitele. Vzhledem k těmto vlastnostem rozhraní USB vytlačuje dosud používané starší paralelní a sériové rozhraní počítačů.

Výhodou USB je také standardizace základních tříd zařízení se stejným účelem, pro které jsou definovány přenosové protokoly. Podpora těchto tříd v operačních systémech pak představuje pro uživatele zjednodušení instalace nových zařízení, které patří do podporované třídy a které tak nepotřebují zvláštní programové vybavení.

Rozšíření sběrnice USB a použití i v nekomerčních projektech je umožněno širokou součástkovou základnou vyráběných elektronických obvodů s implementovaným rozhraním USB. Jejich přehled a srovnání je jedním z cílů diplomové práce.

Další z cílů diplomové práce je návrh a realizace vývojové desky s implementací USB pro připojení k mikroprocesorovým systémům. K tomu účelu byl vybrán obvod rozhraní USB PDIUSBD12 od firmy Philips. Součástí práce je také vytvoření programového vybavení pro obě strany sběrnice USB s využitím volně dostupných nástrojů. Na straně zařízení se jedná o jednoduchou aplikaci s podporou USB a její modifikace pro speciální účely. Na straně hostitele jsou to ovladače, knihovny a aplikace pro komunikaci s deskou. Motivací pro realizaci této desky je otestování možnosti využití sběrnice USB pro řídicí aplikace.

Pro některé negativní vlastnosti, mezi něž patří například topologie sběrnice, omezená maximální vzdálenost zařízení od počítače, nemožnost galvanického oddělení zařízení, se tato sběrnice nerozšířila do průmyslu, kde se i nadále používají vhodnější sběrnice založené například na standardu RS-485. Pro připojení takové sběrnice k počítači, který má jako rozhraní periferie pouze USB, je třeba převodník mezi těmito dvěma rozhraními. Proto je součástí praktické práce vytvoření takového převodníku na realizované vývojové desce. Protože vytvořený převodník nezapadá do žádné ze specifikovaných tříd podporovaných operačním systémem, jsou součástí práce také naprogramované ovladače pro 32 bitový operační systém Windows 2000.

Tato práce je rozdělena do samostatných kapitol podle vytyčených cílů. Kapitola 2 začíná přehledem historického vývoje standardu USB. Hlavní náplní této kapitoly je stručný výtah ze specifikace standardu USB verze 1.1 s některými rozšířeními z verze 2.0. Výtah ze specifikace je proveden s použitím literatury [1], [2] a částečně [3]. V kapitole 3 je provedeno rozdělení a přehled potřebných a dostupných typů integrovaných obvodů pro implementaci USB. Také je v této kapitole stručný popis vybraného obvodu PDIUSBD12. Přehled typů obvodů je doplněn přílohou A, v které je přehledový katalog integrovaných obvodů pro implementaci USB a podrobnější katalog řadičů rozhraní a mikrokontrolérů s implementací USB. Katalogové listy obvodů zařazených do katalogu jsou dostupné na přiloženém CD. Kapitola 4 popisuje hardware vytvořený v rámci diplomové práce. S touto kapitolou souvisí

příloha C, která obsahuje základní technickou dokumentaci k vytvořené vývojové desce. Softwarové vybavení pro vývojovou desku i hostitelský počítač je, stejně jako použité softwarové nástroje, popsán v kapitole 5. Na závěr kapitoly jsou popsány jednotlivé vytvořené programy. Zhodnocení výsledků práce a dosažených cílů je provedeno v kapitole 6. Součástí práce je přiložený kompaktní disk s katalogem, některými použitými softwarovými nástroji a vytvořeným programovým vybavením. Obsah a rozdělení CD je v kapitole 8.

2. Průmyslový standard USB

V této kapitole je stručně popsán standard USB verze 1.1. Popis sběrnice je doplněn o vývoj USB a některé vlastnosti sběrnice podle novější verze 2.0. Podrobnější informace lze najít především v [1], [2] a případně [3], odkud byly čerpány informace.

2.1 Historie USB

Sběrnice USB vznikla jako reakce na potřebu rychlé, univerzální a pro uživatele snadno obsluhovatelné propojení mezi počítačem a periferiemi, umožňující připojit více zařízení k jedné sběrnici za chodu počítače i zařízení.

Standard USB je vyvíjen od první poloviny devadesátých let minulého století zástupci firem z oblasti hardwaru i softwaru sdružených do organizace *USB Implementers Forum* (USB-IF). Největší podíl na něm mají firmy Compaq, Intel, Microsoft, NEC, Hewlett-Packard, Lucent Technologies a Philips.

Na počátku roku 1996 byl uveřejněn standard USB verze 1.0. Firma Intel začala vyrábět a implementovat řadiče USB do základních desek nových počítačů a zásuvných karet pro starší počítače. V té době nebylo rozhraní USB podporováno operačními systémy a specifikace nebyla dostatečně propracovaná, takže některá zařízení připojená k jedné sběrnici společně nepracovala.

Nedostatky specifikace z verze 1.0 byly odstraněny ve verzi 1.1 uveřejněné v polovině roku 1998. V počítačích bylo do té doby rozhraní USB implementováno jen výjimečně. Verze 1.1 byla již podporována novými operačními systémy, tím se rozrůstal i počet periferií připojovaných ke sběrnici USB. Také vznikly nové typy zařízení jako například přenosné paměťové moduly (*USB Flash Drive*).

Ve verzi 1.1 byly definovány dva základní typy rozhraní USB s různou přenosovou rychlostí. Prvním typem jsou nízkorychlostní zařízení (*low-speed devices*, dále jen LS), což jsou zařízení s maximální přenosovou rychlostí do 1,5Mbit/s a menším přenášeným objemem dat. Taková zařízení jsou levnější na výrobu a jsou proto specifikací podporována. Do této skupiny patří například počítačové myši, klávesnice a pákové ovladače. Druhým typem jsou plnohodnotná zařízení (*full-speed devices*, dále jen FS), která mají maximální přenosovou rychlost do 12Mbit/s a jsou primárně určena pro přenos dat jako je digitální zvuk, kompresované video, ISDN. V této verzi už bylo počítáno s dalším vývojem standardu pro vyšší přenosové rychlosti, jak se zvyšovaly nároky na přenosovou rychlost a zvětšoval se objem přenášených dat připojovaných zařízení.

Poslední verze 2.0, zveřejněná v dubnu 2000, rozšiřuje standard o třetí typ zařízení, kterým jsou vysokorychlostní zařízení (*high-speed devices*, dále jen HS) s přenosem do 480Mbit/s. Díky tomuto rozšíření lze pomocí USB připojovat k počítači také zařízení s velkým objemem přenášených dat, jako jsou například disky, připojení k síti, zařízení pro zpracování obrazu a další.

Rostoucí trh s mobilními zařízeními (digitální fotoaparáty, mobilní telefony, PDA a další) přinutil USB-IF vyvinout nový standard *On-The-Go* (dále jen OTG) rozšiřující verzi 2.0 pro tato zařízení. Standard OTG umožňuje propojovat některé zařízení pomocí USB přímo bez stolního hostitelského počítače.

Na závěr ještě zmínka o budoucnosti sběrnice USB, o té svědčí přes 1,1 miliardy vyprodukovaných zařízení s implementovaným rozhraním za prvních pět let od uvedení USB

na trh (údaj pochází ze zdrojů USB-IF). Tím má USB zajištěnu silnou pozici na trhu, kterou neohrozí ani nový standard bezdrátového přenosu *BlueTooth*.

2.2 Specifikace USB

Specifikace standardu USB definuje architekturu sběrnice, elektrické a mechanické vlastnosti jednotlivých prvků, přenosový protokol a datový tok na sběrnici.

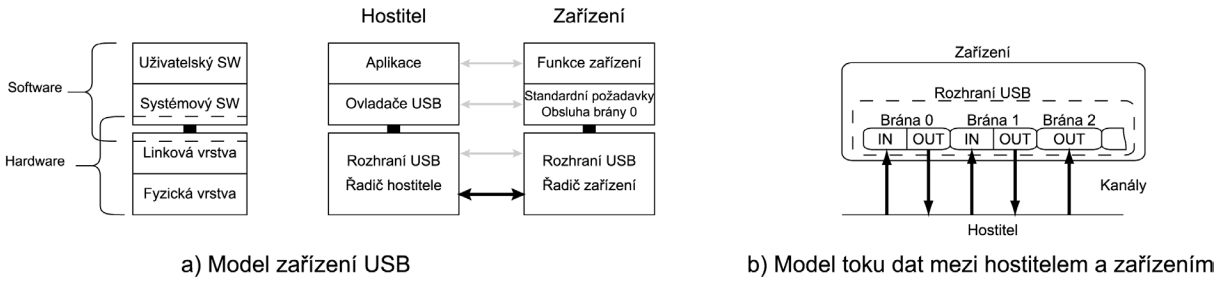
V této podkapitole jsou popsány základní vlastnosti fyzické a linkové vrstvy sběrnice USB podle specifikace verze 1.1.

2.2.1 Fyzická vrstva USB

Do fyzické vrstvy patří architektura sběrnice, její mechanické vlastnosti, elektrické vlastnosti napájecí a signálové části sběrnice a kódování dat. Každé z těchto částí je věnována samostatná podkapitola, podrobnější informace lze nalézt v kapitolách 4, 5, 6 a 7 specifikace USB verze 1.1 [1] a pro verzi 2.0 v [2].

2.2.1.1 Architektura USB

Sběrnice USB je složena z několika typů prvků. Jsou to hostitel (*HOST*), původně vždy pouze jako součást stolního počítače, rozbočovače (*HUB*) a koncová zařízení (*DEVICE*). Dále sběrnici tvoří propojení vždy mezi dvěma prvky typu *point-to-point*. Každý prvek se skládá z několika logických vrstev zobrazených na obrázku 2.1a. Komunikace mezi hostitelem a koncovým zařízením probíhá skrz logické komunikační kanály (*pipes*), které jsou zobrazené na obrázku 2.1b.



a) Model zařízení USB

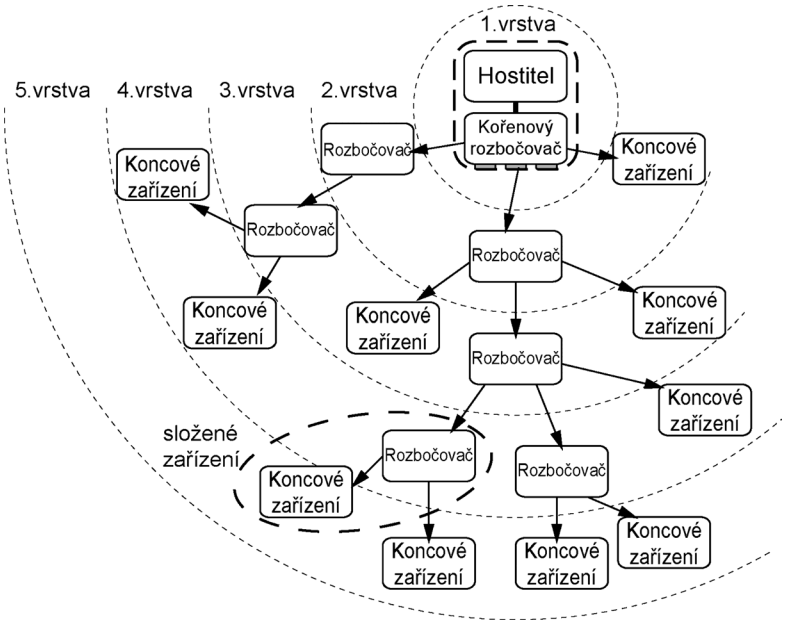
b) Model toku dat mezi hostitelem a zařízením

Obrázek 2.1 - Model sběrnice USB.

Topologie sběrnice má stromovou strukturu, jak je znázorněno na obrázku 2.2. Hostitel tvoří centrální jednotku každé sběrnice a zároveň je spolu s kořenovým rozbočovačem první vrstvou sběrnice. V každé sběrnici USB může být pouze jeden hostitel, který řídí přidělování přenosového média metodou výzvy (*polling*) v logickém kruhu, tj. všechna zařízení poslouchají na síti a pokud zachytí pověření (*token*) se svojí adresou, odpoví na něj. S použitým adresováním lze připojit až 127 nezávislých zařízení k jednomu hostiteli.

Stejně jako kořenový, tak i ostatní rozbočovače rozšiřují sběrnici o další přípojné body a zvyšují počet vrstev sběrnice. Většina rozbočovačů je součástí složeného zařízení, které má v jednom fyzickém pouzdře kromě rozbočovače i koncové zařízení, takové zařízení je připojeno ke svému rozbočovači a patří tak logicky do další vrstvy. Specifikací USB 1.1 je pro LS a FS zařízení povoleno nejvýše 5 vrstev včetně kořenového rozbočovače, pro USB 2.0 (FS, HS) je počet povolených vrstev roven 7, tyto hodnoty vychází z maximálních dovolených zpoždění signálu pro jednotlivé úseky sběrnice a z maximální doby na odpověď zařízení hostitelskému počítači.

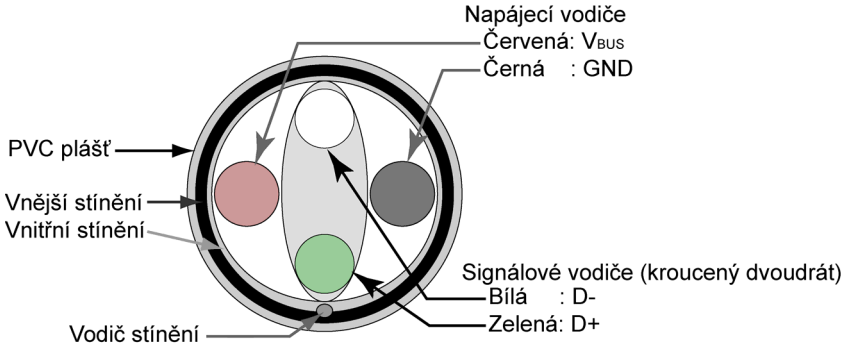
Koncové zařízení obsahuje několik logických bran (*endpoints*) s vlastní FIFO pamětí, přes které zařízení komunikuje s hostitelem. Jednotlivé brány jsou sdruženy do rozhraní (*interfaces*). Každé rozhraní představuje pro hostitele jednu nezávislou funkci zařízení. Například zařízení fungující jako scanner, tiskárna a kopírka má tři rozhraní viditelné pro uživatele hostitelského počítače jako tři samostatné zařízení.



Obrázek 2.2 - Stromová struktura sběrnice USB.

2.2.1.2 Mechanické vlastnosti přenosového média USB

Mechanické vlastnosti sběrnice definují povolené hodnoty parametrů a typy vodičů a konektorů.

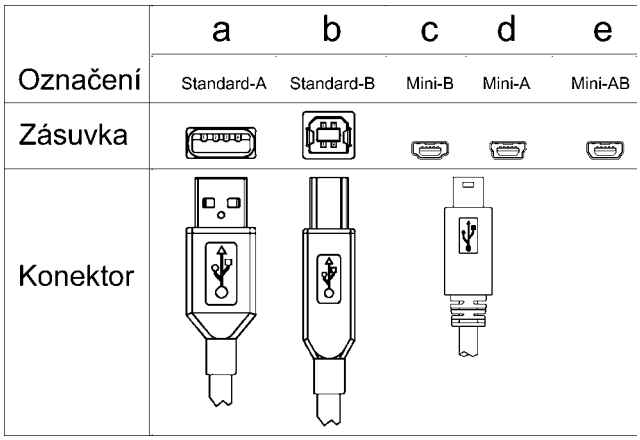


Obrázek 2.3 - Průřez kabelem pro USB.

Na obrázku 2.3 je znázorněn průřez propojovacím kabelem mezi prvky sběrnice USB. Kabel obsahuje 2 dvojice vodičů a stínění. Jedna dvojice vodičů je napájecí (GND, V_{BUS}) a druhá dvojice je datová (D+, D-). Pro FS a HS zařízení je pro datové vodiče předepsán kroucený dvoudrát, s vnějším stíněním. Pro LS zařízení je tento požadavek pouze jako doporučení. Takové kabely se používají pouze v koncových zařízeních s pevně připojeným kabelem, pro jejich nižší cenu a menší možný průměr.

Specifikací povolená délka kabelu je dána vlastnostmi použitých vodičů a požadovanými elektrickými vlastnostmi sběrnice, mezi které patří doba zpoždění, útlum, doba náběžné a sestupné hrany signálů pro datové vodiče a úbytek napětí pro napájecí vodiče. Tyto vlastnosti jsou popsány v následující podkapitole. Vzhledem k těmto parametrům je

maximální délka kabelu 5 metrů pro HS/FS zařízení a 3 metry pro LS zařízení. Specifikace také nepovoluje prodlužovací kabely, protože by nemusely být dodrženy definované parametry.



Obrázek 2.4 - Typy konektorů a zásuvek pro USB.

Každý kabel, s výjimkou kabelů pevně připojených k LS zařízením, je opatřen dvěma konektory s nezaměnitelným tvarem. Tím je zajištěno jednoduché použití a vyloučena možnost nesprávného připojení. Konektor typu "A" má obdélníkový průřez (obr.2.4a) a slouží k připojení kabelu k výstupu z hostitelského počítače nebo rozbočovače. Konektor typu "B" je čtvercového průřezu (obr.2.4b) a kabel je jím připojen k zařízení nebo na vstup rozbočovače. Vzhledem k novým menším zařízením připojovaným pomocí USB, byl specifikací 2.0 přidán nový typ konektoru označený jako "Mini-B" (obr.2.4c), který má přibližně poloviční výšku než konektor typu "B". Specifikací jsou povoleny dva propojovací kabely, první s konektory typu "A" a "B" a druhý s konektory "A" a "Mini-B".

Pro doplnění přehledu je na obrázku 2.4d nový konektor pro OTG zařízení označený "Mini-A" jako alternativa k původnímu konektoru nově označovanému "Standard-A". Kabely s tímto konektorem mohou být dlouhé maximálně 4,5 metru. Na obrázku 2.4e je zásuvka typu "Mini-AB" určená pro OTG zařízení, která mohou být připojena jako zařízení i jako hostitel.

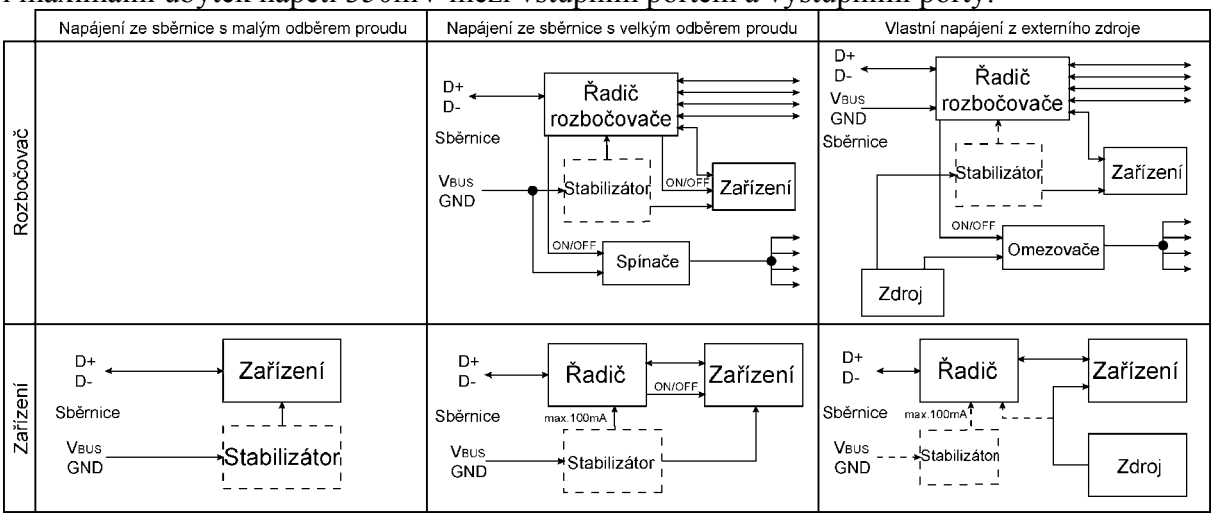
2.2.1.3 Elektrické vlastnosti sběrnicového napájení

Zařízení připojená ke sběrnici USB lze rozdělit z hlediska napájení na tři skupiny. Zařízení napájená ze sběrnice s malým odběrem proudu (*low-power bus-powered device*), zařízení napájená ze sběrnice s velkým odběrem proudu (*high-power bus-powered device*) a zařízení s vlastním napájením (*self-powered device*). Pro každou skupinu je definovaný maximálně možný odebíraný proud a minimální napětí při kterém musí být zařízení schopno řádně pracovat. Přehled zapojení napájecí části pro jednotlivé skupiny zařízení jsou na obrázku 2.5. Zařízení může být z hlediska napájení v jednom ze tří stavů. Prvním stavem je režim obecného zařízení. V tomto stavu se zařízení nachází po připojení ke sběrnici, dokud mu není přiřazena vlastní adresa. Dalším stavem je funkční režim zařízení. Třetí stav je režim pozastavení činnosti zařízení (*suspend mode*).

Zařízení napájená ze sběrnice s malým odběrem proudu mohou odebírat maximálně 100mA, kromě režimu spánku, kdy mohou odebírat nejvýše 0,5mA. Dále musí být schopna pracovat již od napětí 4,4V. Toto napětí se měří na začátku přívodního kabelu u konektorem typu "A".

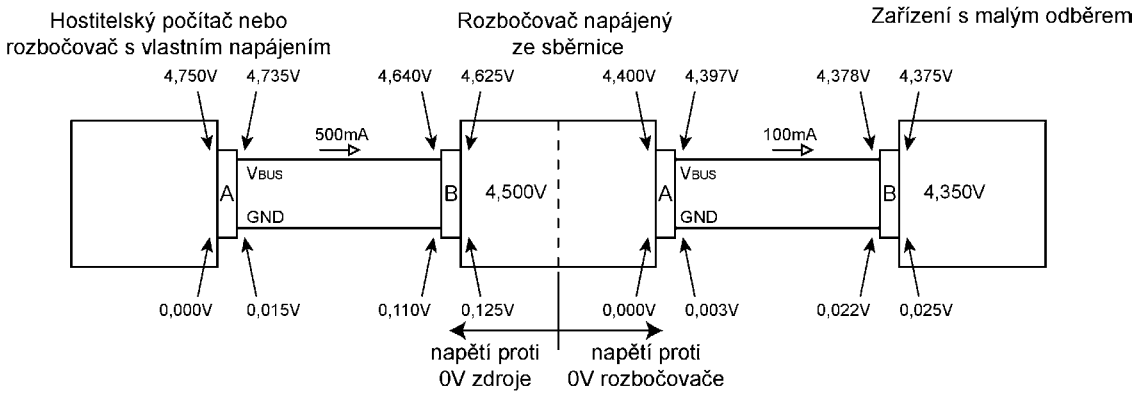
Zařízení, která ve funkčním režimu odebírají více než 100mA, patří do skupiny zařízení napájených ze sběrnice s velkým odběrem proudu. Jejich odběr nesmí překročit 100mA pro režim obecného zařízení a musí pracovat od napětí 4,4V. Ve funkčním režimu zařízení nesmí

odebírat více než 500mA a musí pracovat od napětí 4,75V, obě napětí jsou měřena na konci přívodního kabelu u konektorem typu "A". Pro režim spánku je povolen maximální odběr 2,5mA. Mezi tato zařízení patří rozbočovače napájené po sběrnici. Pro ně je předepsán i maximální úbytek napětí 350mV mezi vstupním portem a výstupními porty.



Obrázek 2.5 - Zařízení USB podle typu napájení.

Zařízení s vlastním napájením mohou mít USB řadič napájen buď z vlastního zdroje nebo ze sběrnice, v tom případě zařízení může odebírat maximálně 100mA. Sběrnicové napájení je většinou využito pro komunikaci zařízení s hostitelským počítačem při vypnutém vlastním napájení.



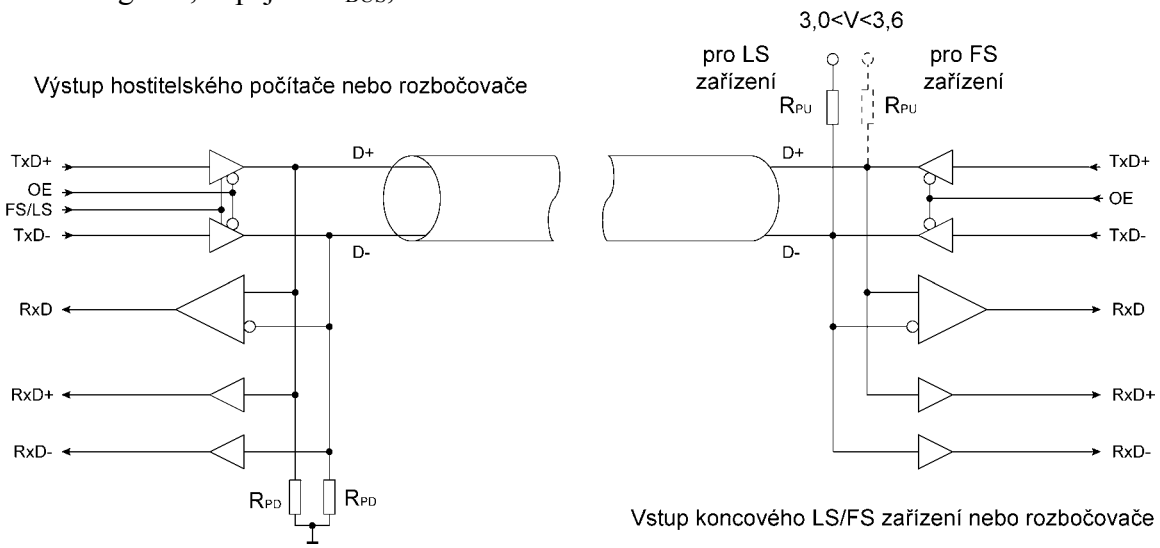
Obrázek 2.6 - Úbytky napětí v nejhorším možném případu.

Na obrázku 2.6 jsou znázorněny minimální napětí jednotlivých typů zařízení a zároveň nejhorší možný případ úbytků napětí na sběrnicovém napájení ve funkčním režimu všech znázorněných zařízení. Z obrázku je patrné, že největší možný úbytek napětí na propojovacím kabelu je 125mV, což je jeden z faktorů omezující největší povolenou délku propojovacího kabelu.

2.2.1.4 Elektrické vlastnosti signálové sběrnice

Přenos signálů po sběrnici USB je zajištěn dvojicí signálových vodičů *D+* a *D-* s rozdílovým signálem. Tolerované hodnoty výstupu budičů jsou pro nízkou úroveň signálu $V_{OL} \leq 0{,}3V$ při zátěži $R_{PU}=1{,}5k\Omega$ proti napětí 3,6V a pro vysokou úroveň signálu $V_{OH} \geq 2{,}8V$ při zátěži $R_{PD}=15k\Omega$ proti napětí 0V pro vysokou úroveň signálu.

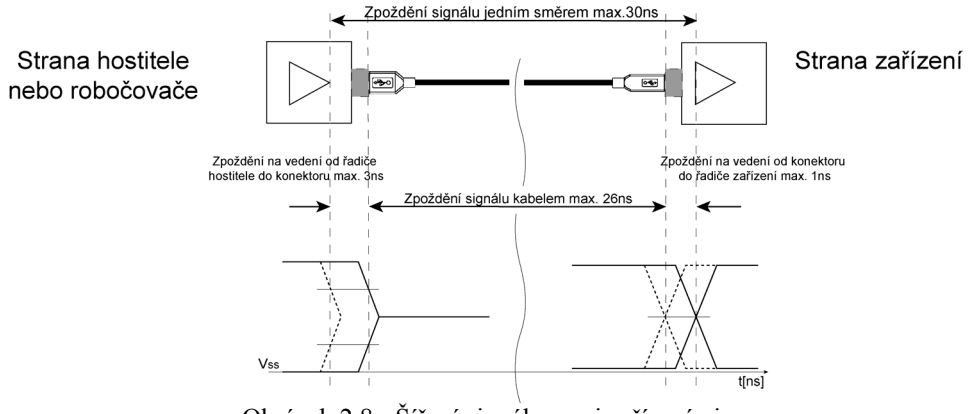
Budiče vysílačů jsou třístavové, aby mohl probíhat obousměrný poloduplexní přenos. Podle specifikace musí být oba budiče datových signálů schopny vydržet trvalý zkrat proti druhému signálu, napájení V_{BUS}, GND a stínění kabelu.



Obrázek 2.7 - Připojení jednotlivých prvků na sběrnici podle specifikace 1.1.

FS zařízení musí mít vysílače (obrázek 2.7) s předepsanou impedanci 28Ω až 44Ω, což je dáno definovaným spojením pomocí kabelu s impedancí Z_o=90Ω±15% a maximálním[1] zpožděním hrany signálu vyslaného k zařízení a jeho odrazu v rámci vysílání jednoho bitu. Do tohoto času se počítá 26ns na dobu šíření signálu kabelem, což má vliv na maximální možnou délku kabelu. Zbylé 4ns jsou rozděleny na maximálně 3ns zpoždění způsobené vyvedením signálu z čipu a desky hostitelského počítače nebo rozbočovače ke konektoru a 1ns zpoždění pro vedení signálu z konektoru na desce zařízení (obrázek 2.8). Maximální dobu šíření ovlivňují také povolené doby náběhu a sestupu hran signálu, které jsou maximálně 4-20ns±10%.

Pro LS zařízení je dáno maximální zpoždění na propojovacím kabelu 18ns, což převážně určují povolené doby náběhu a sestupu hrany signálu, které jsou 75-300ns±20%.

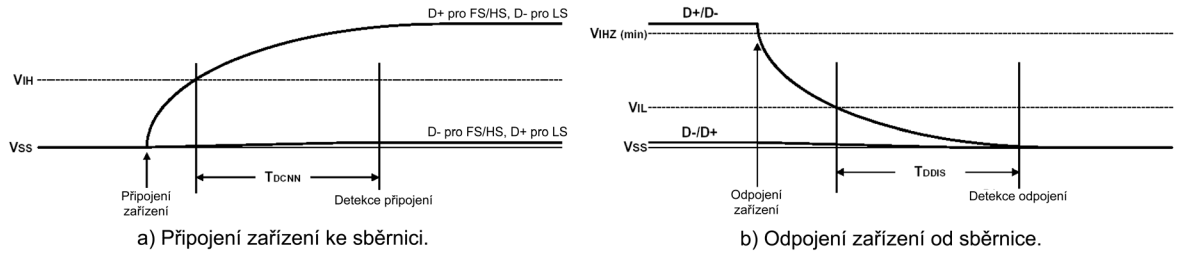


Obrázek 2.8 - Šíření signálu mezi zařízeními.

Zařízení používají pro příjem signálů dva druhy přijímačů (obrázek 2.7). Rozdílový přijímač a samostatné přijímače pro každý datový signál. Ty detekují nízkou úroveň vstupního signálu pro napětí V_{IL}≤0,8V a vysokou úroveň vstupního signálu pro napětí V_{IH}≥2,0V. Napětí obou

[1] Výraz maximální je použit jako specifikací povolená největší hodnota.

signálů *D+* a *D-* může být menší než V_{IH} maximálně 14ns pro FS a 210ns pro LS zařízení, aniž by byl stav sběrnice interpretován jako stav *SE0*, popsaný dále v této podkapitole.



Obrázek 2.9 - Průběh signálů sběrnice při připojení/odpojení zařízení.

Dokud není žádné zařízení ke sběrnici připojeno, jsou napětí na obou datových linkách *D+* a *D-* nulové. Po připojení zařízení, vlivem odporů R_{PD} a R_{PU} zapojených podle obrázku 2.7, dojde k nárůstu napětí na linkách, jak je znázorněno na obrázku 2.9a. Napětí na lince, která je připojena přes odpor R_{PU} k napětí 3,0V až 3,6V, překročí hodnotu V_{IH} a pokud tento stav vydrží déle než 2,5µs, je detekováno připojení zařízení. Je-li odpor R_{PU} připojen k lince *D-*, je detekováno zařízení typu LS. V druhém případě (R_{PU} je připojen k lince *D+*) je detekováno zařízení typu FS. Odpojení zařízení je detekováno v případě, že napětí na obou linkách pokleslo pod V_{IL} na dobu delší než 2,5µs.

Detekce zařízení typu HS je složitější, stejně jako struktura bloku HS vysílače a přijímače, proto je zde pouze naznačena. Podrobně je průběh popsán v kapitolách 7.1.5 až 7.1.7.5 specifikace [2]. Připojením odporu R_{PU} se detekuje zařízení jako typ FS, poté zařízení vysílá určitou dobu sled signálů tak, aby rozbočovač detekoval zařízení připojené k portu jako typ HS. Rozbočovač poté vyšle signál pro zapnutí funkce HS zařízení. Zařízení odpojí odpor R_{PU} a připojí k lince HS vysílač a přijímače a další komunikace probíhá v režimu HS zařízení.

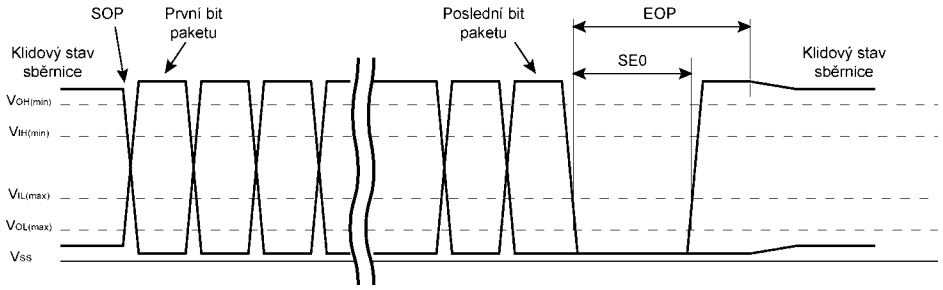
FS a LS zařízení využívají pro komunikaci tři základní typy úrovní signálu na linkách *D+* a *D-*. Jsou to "*rozdílová 1*", "*rozdílová 0*" a "*jednoznačná 0*" (*SE0 – single-ended 0*), jejich definice na straně přijímače je uvedena v tabulce 2.1. Definici na straně hostitele lze nalézt v kapitole 7.1.7 [1].

Tabulka 2.1 - Základní stavy sběrnice definované na straně zařízení.

Stav sběrnice	**Definice stavu**
Rozdílová "1"	$(V_{D+} - V_{D-}) > 200mV$ a $U_{D+} > V_{IH(min)}$.
Rozdílová "0"	$(V_{D-} - V_{D+}) > 200mV$ a $U_{D-} < V_{IL(max)}$.
Jednoznačná "0" (SE0)	V_{D+} a $V_{D-} < V_{IL(max)}$
Datový stav "J"	Pro FS zařízení : "Rozdílová 1". Pro LS zařízení : "Rozdílová 0".
Datový stav "K"	Pro FS zařízení : "Rozdílová 0". Pro LS zařízení : "Rozdílová 1".
Klidový stav (pro FS)	$V_{D+} > V_{IHZ(min)}$ a $V_{D-} < V_{IL(max)}$
Klidový stav (pro LS)	$V_{D-} > V_{IHZ(min)}$ a $V_{D+} < V_{IL(max)}$
Začátek paketu (SOP)	Přechod z klidového stavu do stavu "K".
Konec paketu (EOP)	SE0 na dobu přenosu dvou bitů následovaný stavem "J" na dobu přenosu jednoho bitu.
Připojení zařízení	Klidový stav nejméně po dobu 2ms.
Odpojení zařízení	SE0 nejméně po dobu 2,5µs.
Signál reset	V_{D+} a $V_{D-} < V_{IL(max)}$ po dobu nejméně 10ms

Přenos dat probíhá po paketech, které jsou ohraničeny událostmi "začátek paketu" (SOP) a "konec paketu" (EOP), jak je zobrazeno na obrázku 2.10. SOP je detekován přechodem ze stavu nečinné sběrnice do stavu K, který je součástí synchronizačního pole paketu, které je

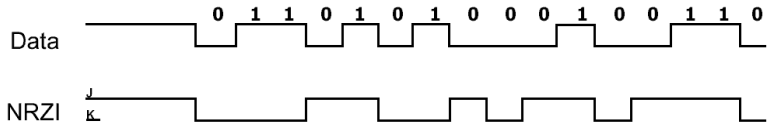
popsáno v následující podkapitole 2.2.2. EOP je definován jako přechod do stavu SE0 přibližně na dobu přenosu 2 bitů následovaný stavem J. Z něj přejde sběrnice buď do nečinného stavu nebo začne vysílání dalšího paketu.



Obrázek 2.10 - Průběh signálů D+ a D- při přenosu paketu.

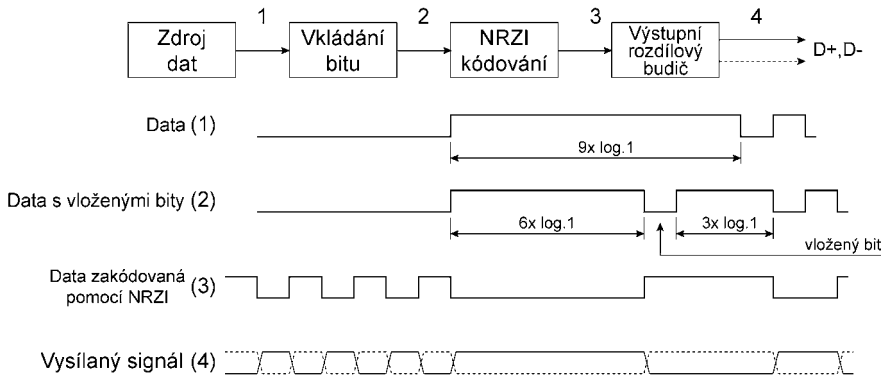
2.2.1.5 Kódování dat

USB používá pro přenos dat po sběrnici kódování NRZI (*Non Return to Zero Inverse*), při něm je datová logická jednička reprezentována jako neměnný stav signálů na vodičích sběrnice a datová logická nula jako změna z jednoho stavu do druhého, jak je znázorněno na obrázku 2.11. Na sběrnici se tak při přenosu dat objevují dva stavy J a K, které se pro LS a FS zařízení liší a jsou popsány v předchozí podkapitole 2.2.1.4.



Obrázek 2.11 - Kódování dat pomocí NRZI.

Z použitého kódování plyne, že sekvence logických nul v datech způsobuje změnu stavu na sběrnici při každém bitu, naproti tomu sekvence logických jedniček nechává po celou dobu sběrnici ve stejném stavu. Jelikož se přenos synchronizuje pomocí změny stavu na sběrnici je změna zajištěna i pro sekvenci logických jedniček vkládáním bitu (*bit stuffing*). Při něm se po každých vyslaných šesti datových jedničkách vloží navíc jedna nula, tím dojde minimálně jednou za sedm bitů ke změně stavu a tím i synchronizaci sběrnice (obrázek 2.12). Současně je vkládání bitu využito k zabezpečení přenosu.



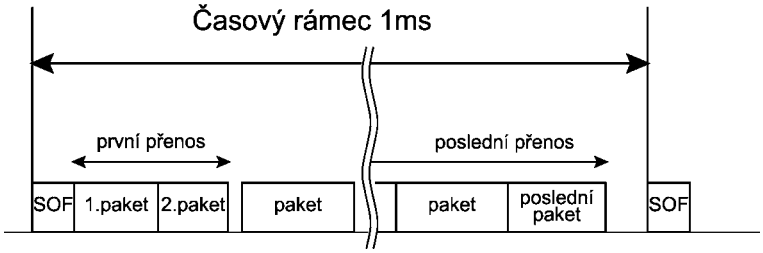
Obrázek 2.12 - Příklad dat vysílací části přenosové cesty.

Při dekódování se každá logická nula po šesti logických jedničkách vyřadí. Pokud je přijato sedm a více jedniček, nastaví se chyba a celý paket se ignoruje. Speciálním případem je poslední bit před koncem paketu, kdy může dojít v rozbočovači k prodloužení tohoto bitu,

který přijímač detekuje jako další bit před koncem paketu. Pokud je tento nadbytečný bit jako šestá logická jednička, kterou nenásleduje žádný datový přenos, je paket v pořádku přijat i bez předpokládané následující logické nuly.

2.2.2 Linková vrstva

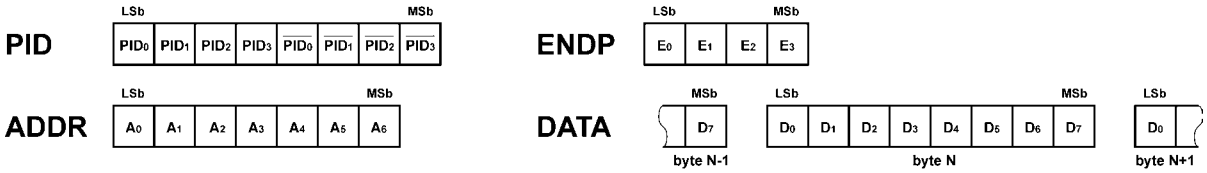
Přenos je na sběrnici organizován do časových rámců (*frames*). V nich se přenášejí informace v paketech seskupených do přenosů pro jednotlivé zařízení. Každý rámec je určen speciálním paketem označující začátek rámce (obrázek 2.13). Rámce mají pro FS a LS zařízení délku 1ms±500ns. HS zařízení používají mikrorámce (*microframes*), které jsou dlouhé 125µs±62,5ns. Řazení paketů v jednotlivých rámcích není pevně dané a určuje ho hostitel, jak je popsáno dále.



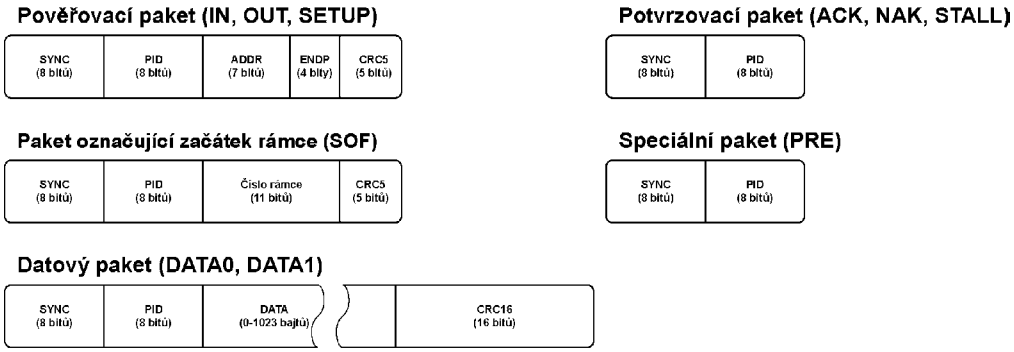
Obrázek 2.13 - Přenos po sběrnici v časových rámcích.

2.2.2.1 Pakety a jejich pole

Jsou definovány čtyři skupiny paketů – pověřovací (*Token*), datové (*Data*), potvrzovací (*Handshake*) a speciální (*Special*). Pakety a jejich rozdělení do skupin jsou znázorněny v tabulce 2.2. Každý paket je složen z několika různě dlouhých polí, ale celý paket je vždy dlouhý celočíselný počet bajtů. Bity jednotlivých bajtů jsou posílány od nejméně důležitého (*LSb*) po nejvíce důležitý (*MSb*). Následující obrázky 2.14 a 2.15, zobrazují jednotlivé pole a složení paketů z polí tak, jak jsou bity vysílány, tj. v pořadí zleva doprava.



Obrázek 2.14 - Struktura některých polí, tvořící pakety.



Obrázek 2.15 - Složení paketů z jednotlivých polí.

Každý paket začíná synchronizačním polem SYNC, které je dlouhé jeden bajt a má hodnotu 128. Po NRZI kódování je tak poslána sekvence stavů "KJKJKJKK" (popis stavu

"K" a "J" je v podkapitole 2.2.1.4). Toto pole zajišťuje synchronizaci přijímače na vyslaná data.

Další pole, které každý paket obsahuje, je identifikační pole PID (*Packet IDentifier field*) určující druh paketu. Pole PID je dlouhé osm bitů, z nichž čtyři bity nesou informaci a zbylé čtyři jsou jejich doplňkem. Pokud negace jedné čtveřice bitů neodpovídá druhé nebo je zjištěna neznámá kombinace, je celý paket zařízením ignorován. Přehled možných typů paketu je v tabulce 2.2, jejich kódy lze nalézt v kapitole 8.3.1 specifikace [1].

Tabulka 2.2 - Typy paketů a jejich identifikace.

Skupina paketů	Typ paketu	Popis
Pověřovací (*Token*)	OUT	Přenos dat od hostitele k zařízení.
	IN	Přenos dat ze zařízení k hostiteli.
	SETUP	Konfigurační přenos od hostitele.
	SOF	Označuje začátek rámce.
Datový (*Data*)	DATA0	Sudý datový paket (první v přenosu).
	DATA1	Lichý datový paket.
Potvrzovací (*Handshake*)	ACK	Potvrzení bezchybného příjmu dat.
	NAK	Data nebyla přijata nebo vyslána.
	STALL	Brána zařízení je pozastavena nebo konfigurační požadavek není podporován.
Speciální (*Special*)	PRE	Označuje LS přenos tak, aby rozbočovače zapnuly LS zařízení k nim připojené.

Pověřovací pakety typu OUT (vysílací), IN (přijímací) a SETUP (konfigurační – speciální případ vysílacího paketu) obsahují adresu složenou ze dvou polí. Prvním polem je adresa zařízení ADDR určující zdroj nebo cíl následujících datových paketů. Má délku 7 bitů, proto může být v jedné síti až 127 zařízení (adresa 0 je určena pro nenakonfigurované zařízení). Druhé pole ENDP obsahuje adresu USB brány (*endpoint*) v rámci jednoho zařízení a má délku čtyř bitů, což určuje maximální počet bran v zařízení. LS zařízení může obsahovat maximálně tři brány, na rozdíl od FS zařízení, které může mít až šestnáct bran. Všechna zařízení musí mít alespoň bránu 0 s řídícím typem přenosu, ostatní mohou být jakéhokoliv typu. Přehled typu bran a přenosů je v podkapitole 2.2.2.2. Pro vysílací a konfigurační typ určuje pole ADDR a ENDP příjemce datových paketů v přenosu, pro přijímací typ je polem ADDR a ENDP určeno, kdo bude následující datové pakety vysílat.

Adresní pole jsou chráněna pětibitovým polem CRC5 (*Cyclic Redundancy Check*) s generujícím polynomem $G(X) = X^5+X^2+1$.

Mezi skupinu pověřovacích paketů patří i paket začátku rámce (SOF – *Start of Frame*). Kromě pole PID obsahuje paket pole s číslem rámce a jeho ochranu pomocí pole CRC5. Pole s číslem rámce je dlouhé jedenáct bitů a jeho hodnota se s každým rámcem zvětšuje cyklicky mezi 0 a 2047. Standard nezajišťuje doručení tohoto paketu, protože zařízení na paket neodpovídají a pokud nepotřebují ke své činnosti synchronizaci sběrnice, pak tento paket ignorují.

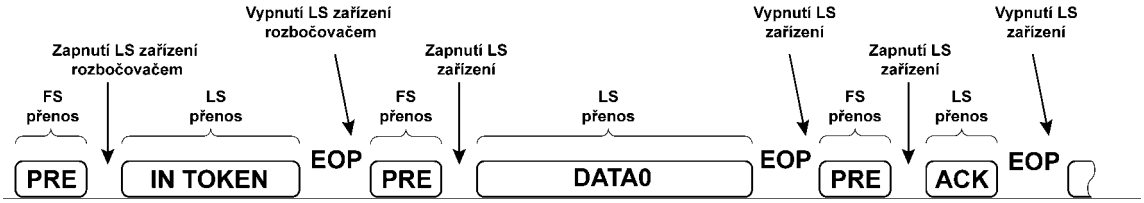
Pověřovací pakety smí vysílat pouze hostitel, který tak řídí přístup na sběrnici. Pověřovací pakety jsou zařízeními ignorovány, pokud nekončí značkou EOP a nemají délku tři bajty.

Další skupinou jsou datové pakety. V této skupině jsou dva typy paketů lišící se pouze polem PID, které určuje zda se jedná o sudý datový paket (DATA0) nebo lichý datový paket (DATA1). Při vysílání dvou a více datových paketů za sebou se oba typy střídají, čímž se synchronizuje datový přenos.

Hlavní částí datových přenosů je datové pole, které může být dlouhé 0 až 1023 bajtů. Délka je závislá na typu přenosu a na hardwarové velikosti paměti vstupní brány. Datové pole je chráněno dvoubajtovým polem CRC16 s generujícím polynomem $G(X) = X^{16}+X^{15}+X^{2}+1$.

Poslední základní skupinou jsou potvrzovací pakety. Obsahují pouze pole PID a slouží k informování o stavu předchozích přenosů. Do této skupiny patří tři pakety, prvním z nich je kladné potvrzení (ACK). Tento paket posílá hostitel jako reakci na přenosy zahájené paketem typu IN nebo zařízení jako reakci na přenosy zahájené pakety typu OUT nebo SETUP. Dalším typem je záporná odpověď (NAK), tu posílá zařízení jako odpověď, že nemůže dočasně přijmout data poslaná hostitelem (v rámci přenosu zahájeného paketem OUT) nebo že nemá žádná data k odeslání (přenos zahájený paketem IN). Zařízení pak může přijmout nebo odeslat data v rámci dalšího přenosu bez zásahu hostitele. Posledním paketem je záporná odpověď se zastavením (STALL). Zařízení posílá tento paket při nemožnosti přijmout nebo vyslat data bez dalšího zásahu hostitele. Navíc tento paket posílá zařízení jako reakci na nepodporovaný konfigurační přenos.

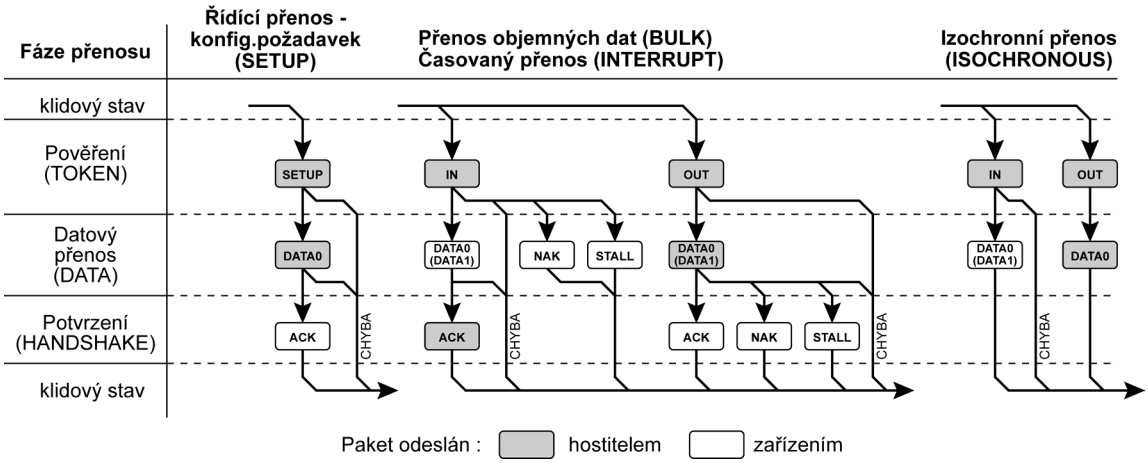
Speciální skupinou je paket PRE, kterým hostitel označuje přenos pro LS zařízení. Tento paket je určen speciálně pro rozbočovače. Po sběrnici probíhají zároveň FS a LS přenosy, proto musí být LS zařízení od sítě odpojena, dokud není přenos typu LS. Bez odpojování by LS zařízení přijímala i FS přenosy, které by špatně interpretovala. Po paketu PRE musí hostitel čekat nejméně po dobu 4 bitů FS přenosu než začne vysílat, aby měl rozbočovač čas na připojení LS zařízení. Na obrázku 2.16 je příklad takového přenosu.



Obrázek 2.16 - Příklad průběhu přenosu pro LS zařízení.

2.2.2.2 Typy a průběh přenosů

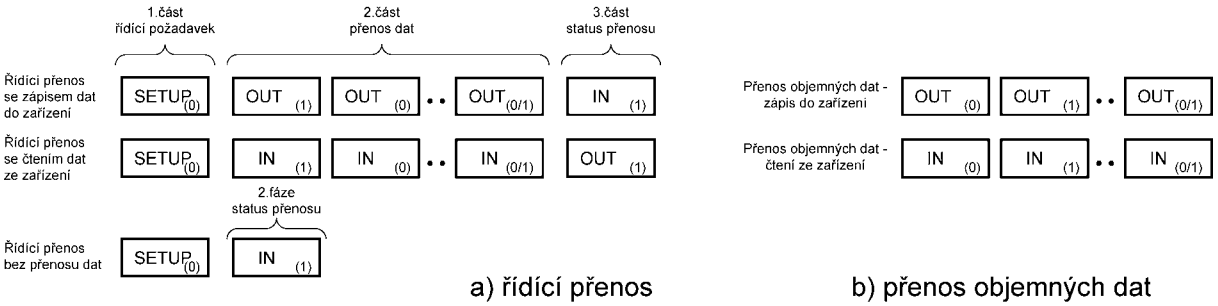
Typ přenosu a jeho formát je závislý na typu brány. Jsou definovány čtyři druhy bran a tím i čtyři druhy přenosu – řídící (*control*), časovaný[2] (*interrupt*), izochronní (*isochonous*) a přenos objemných dat (*bulk*). LS zařízení mohou využívat pouze řídící a časovaný přenos. Obecně je každý přenos složen z pověření, datového přenosu a potvrzení, jak je znázorněno na obrázku 2.17.



Obrázek 2.17 - Průběhy jednotlivých typů přenosu.

[2] Použitý výraz není přesným překladem, je však použit pro jeho větší výstižnost.

Řídící přenos má dvě nebo tři části v závislosti na potřebě přenosu dat. V první části je vyslán hostitelem paket s konfiguračním pověřením následovaný datovým paketem s požadavkem, popis požadavků je v podkapitole 2.2.3. Pokud je požadavek v pořádku přijat, zařízení vyšle potvrzení úspěchu. V požadavku je obsažena informace o případném přenosu dalších dat v rámci řídícího přenosu. Přenos dat je řízen podle stejných pravidel jako přenos objemných dat, všechny datové pakety mají vždy stejný směr přenosu. Řídící přenos ukončuje stavová část a pozná se opačným směrem toku dat než byla vysílaná data, jak je znázorněno na obrázku 2.18a.



Obrázek 2.18 - Posloupnost operací při řídicím přenosu a přenosu objemných dat.
Čísla v závorkách označují, který typ datového paketu je posílán (DATA0 nebo DATA1).

Při přenosu objemných dat je zaručeno jejich bezchybné doručení kontrolou chyb a případným opakováním paketů. Není však zaručeno zpoždění, s jakým jsou data doručena, viz.podkapitola 2.2.2.3. Přenos je třífázový s výjimkou nulové délky dat, kdy přenos neobsahuje datový paket. Při čtení dat ze zařízení vyšle hostitel IN paket na který zařízení zareaguje odesláním dat nebo odpovědí NAK, pokud nemá data připravena, ale může je odeslat později a nebo odpovědí STALL, pokud je brána pozastavena. Úspěšné obdržení dat hostitel potvrdí zprávou ACK. Jestliže hostitel nezašle žádné potvrzení, znamená to chybu přenosu. Při vysílání dat do zařízení je situace obdobná. Zařízení může odpovědět ACK, pokud data byla přijata v pořádku a hostitel může poslat další datový paket, NAK pokud data přišla v pořádku, ale zařízení je nezpracovalo a hostitel bude muset poslat paket znovu, STALL pokud je brána mimo provoz a hostitel by neměl ve vysílání paketů pokračovat nebo zařízení nepošle žádnou odpověď, pokud přišla data poškozená, v tom případě hostitel vyšle znovu stejný paket. Posloupnost přenosů je zobrazena na obrázku 2.18b.

Časovaný přenos má stejné fáze a pravidla jako přenos objemných dat. Rozdíl mezi nimi je v systému přidělování kapacity šířky pásma. U časovaných přenosů vysílá hostitel pověření s periodou, kdy je zařízení schopno vysílat či přijímat data, čímž se šetří přenosová kapacita média. Více o přidělování kapacity je v následující podkapitole.

Izochronní přenos má pouze dvě fáze, příjemce nepotvrzuje přijetí a proto není doručení dat zaručeno. Používá se k přenosu souvislého toku dat.

2.2.2.3 Přidělování kapacity šířky pásma

Přidělování kapacity sběrnice řídí výlučně hostitelský počítač. Rozvrhování přenosů v rámci není specifikováno standardem USB a je proto různé podle implementace hostitele. Jsou předepsány pouze limitující pravidla. Periodickým přenosům (izochronní a časované) může být přiděleno nejvýše 90% rámce (80% mikrorámce pro HS sběrnici). Zbylých 10% je primárně přiděleno řídícím přenosům. Pokud zbývá nějaká kapacita z rámce, je přidělena přenosu objemných dat.

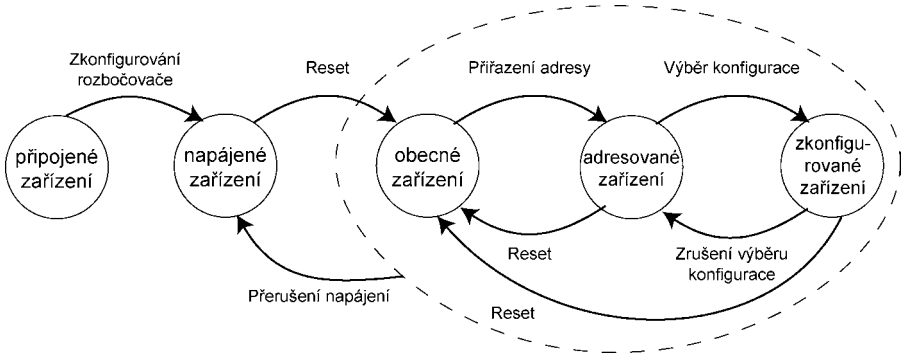
2.2.3 Řídící přenos

Každé zařízení musí mít nejméně jednu obousměrnou řídící bránu s adresou 0 (*endpoint 0*). Po připojení zařízení ke sběrnici je pro hostitele viditelná pouze tato brána a skrz ní může být zařízení přidělena adresa a provedena konfigurace. Tento proces se označuje jako *enumerace* zařízení a je popsán v následující podkapitole 2.2.3.1. Při enumeraci zasílá zařízení informace o sobě pomocí *deskriptorů*, jejich popis je v podkapitole 2.2.3.2.

Hostitel zahajuje řídící přenosy vysláním konfiguračního požadavku, jehož struktura je popsána v podkapitole 2.2.3.3. Všechny zařízení musí minimálně reagovat na sadu standardních požadavků, ta je popsána v podkapitole 2.2.3.4. Detailní informace lze získat v [1] nebo pro HS zařízení v [2].

2.2.3.1 Enumerace zařízení

Zařízení připojené ke sběrnici USB je až do odpojení v některém ze stavů znázorněných na obrázku 2.19. Enumerací přejde zařízení ze stavu připojení do stavu zkonfigurovaného zařízení.



Obrázek 2.19 - Stavy zařízení z hlediska USB.

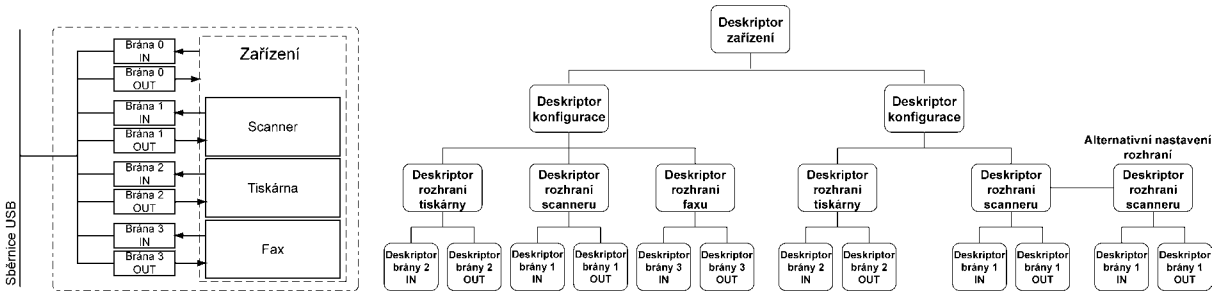
V první fázi je hostitel informován rozbočovačem, k němuž je zařízení připojeno, o změně stavu na portu. Rozbočovač zapne napájení portu. Hostitel počká nejméně 100ms, aby zařízení mělo čas na ustálení napájecího napětí a poté vyšle signál k povolení portu a signál reset. Tím se zařízení dostane do stavu obecného zařízení a je schopno komunikovat s hostitelem na adrese 0 pomocí řídící brány 0.

Hostitel zjistí informace přečtením deskriptoru zařízení a přidělí zařízení unikátní adresu v rámci sběrnice. Zařízení se tak dostane do adresovaného stavu. Od toho okamžiku hostitel komunikuje se zařízením na přidělené adrese. Hostitel přečte všechny konfigurace zařízení, což může trvat až několik milisekund.

Na základě získaných informací nastaví hostitel jednu konfiguraci a tím přejde zařízení do zkonfigurovaného stavu. V tomto stavu může zařízení používat všechny brány popsané deskriptory vybrané konfigurace.

2.2.3.2 Deskriptory zařízení

Univerzálnost sběrnice USB je také založena na možnosti připojovat k hostiteli zařízení různého charakteru. Tuto vlastnost umožňují deskriptory, kterými je zařízení popsáno pro potřeby sběrnice a toku dat. Jsou to datové struktury definovaného formátu, které zařízení vyšle jako reakci na požadavek hostitele. Je definováno několik typů deskriptorů – deskriptor zařízení, konfigurace, rozhraní, brány a textového řetězce. Pro HS zařízení jsou definovány i další typy popsané v kapitole 9 [2]. Deskriptory jsou hierarchicky zřetězeny, jak je zobrazeno na obrázku 2.20. Díky tomu lze využít možnosti více konfigurací, rozhraní a bran a tím i větší flexibility zařízení. Programově lze pak za běhu měnit vlastnosti zařízení změnou odesílaných deskriptorů.



Obrázek 2.20 - Hierarchie deskriptorů – příklad popisu zařízení.

Správné nastavení deskriptorů je jednou z podmínek úspěšné enumerace zařízení a proto zde budou jednotlivé deskriptory stručně popsány. Detailnější informace lze nalézt v kapitole 9.5 specifikace [1].

Každý deskriptor začíná dvoubajtovou hlavičkou, první bajt je délka deskriptoru a druhý je identifikátor typu deskriptoru.

Tabulka 2.3 - Struktura deskriptoru zařízení.

Jméno pole	Ofset[B]	Délka[B]	Popis
bLength	0	1	Velikost deskriptoru.
bDescriptorType	1	1	Kód deskriptoru zařízení (=1).
bcdUSB	2	2	BCD kódované číslo specifikace, se kterou jsou deskriptory kompatibilní.
bDeviceClass	4	1	Identifikátor třídy zařízení. Pokud je roven nule, každé rozhraní v konfiguraci má nastavenou třídu nezávisle na ostatních rozhraních. Pokud je roven 255, zařízení nepatří do žádné definované třídy.
bDeviceSubClass	5	1	Identifikátor podskupiny třídy zařízení.
bDeviceProtocol	6	1	Identifikátor protokolu závislý na podskupině a třídě zařízení.
bMaxPacketSize0	7	1	Maximální velikost dat při jednom přenosu řídící branou 0.
idVendor	8	2	Identifikátor výrobce, který přiděluje USB IF.
idProduct	10	2	Identifikátor výrobku, který si volí výrobce.
bcdDevice	12	2	BCD kódovaná verze výrobku.
iManufacturer	14	1	Index textového řetězce popisující výrobce (většinou jméno výrobce) nebo nula, pokud není výrobce popsán.
iProduct	15	1	Index textového řetězce popisující výrobek nebo nula.
iSerialNumber	16	1	Index textového řetězce se sériovým číslem nebo nula.
bNumConfiguration	17	1	Počet možných konfigurací zařízení.

Deskriptor zařízení mají všechna zařízení pouze jeden, jeho struktura je v tabulce 2.3. V něm jsou základní informace o zařízení a použité specifikaci USB, podle které se zařízení chová. Mezi základní informace patří kód třídy (definovaný organizací *USB-IF*), ke které zařízení patří a případně používaný protokol třídy, velikost paměti brány 0, která musí být minimálně 8 bajtů a maximálně 64 bajtů, indexy textových řetězců popisující výrobce a zařízení v textové podobě. Velice důležité jsou pole *idVendor* a *idProduct*, což jsou identifikátory výrobce VID (Vendor ID) a výrobku PID (Product ID). Na základě těchto dvou polí vybere hostitelský systém vhodný ovladač zařízení. Označení výrobku si určuje výrobce sám. Přidělení kódu výrobce provádí organizace *USB-IF*.

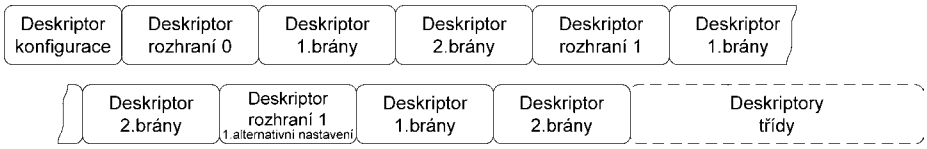
Získat VID lze buď jako součást členství v *USB-IF* za roční poplatek $2500 nebo jako součást licence na používání loga USB v souvislosti s vlastními výrobky za poplatek $1500 na dva roky. Poslední možností je získání VID za poplatek $1500 bez možnosti užívat logo USB.

Posledním polem v deskriptoru zařízení je počet možných konfigurací zařízení. Pro každou konfiguraci musí být samostatný deskriptor i se všemi hierarchicky svázanými deskriptory.

Tabulka 2.4 - Struktura deskriptoru konfigurace.

Jméno pole	Ofset[B]	Délka[B]	Popis
bLength	0	1	Velikost deskriptoru.
bDescriptorType	1	1	Kód deskriptoru konfigurace (=2).
wTotalLength	2	2	Celková délka všech deskriptorů posílaných spolu s deskriptorem konfigurace.
bNumInterfaces	4	1	Počet rozhraní v konfiguraci.
bConfigurationValue	5	1	Identifikátor konfigurace. Tato hodnotu je použita pro výběr konfigurace požadavkem *SetConfiguration*.
iConfiguration	6	1	Index textového řetězce popisující tuto konfiguraci nebo nula, pokud není konfigurace popsána.
bmAttributes	7	1	Vlastnosti zařízení v konfiguraci : D6: Zařízení má vlastní napájení D5: Zařízení může vzbudit hostitele
MaxPower	8	1	Maximální proud odebíraný zařízením ze sběrnice při použití této konfigurace. Jednotkou jsou 2mA.

Deskriptor konfigurace (tabulka 2.4) určuje počet rozhraní zařízení v konfiguraci. Každé zařízení může mít více konfigurací, například s různým druhem napájení zařízení nebo jiným rozložením bran pro jednotlivé rozhraní. Deskriptor obsahuje také informace o režimu napájení a maximální velikosti odebíraného proudu ze sběrnice. Tyto informace může hostitel využít pro řízení napájení sběrnice. Dále obsahuje celkovou velikost všech deskriptorů posílaných při požadavku o odeslání deskriptoru konfigurace. S ním se posílají deskriptory všech rozhraní konfigurace, deskriptory bran rozhraní a vlastní deskriptory třídy zařízení, jejich pořadí je jako příklad znázorněno na obrázku 2.21.



Obrázek 2.21 - Pořadí odesílaných deskriptorů spolu s deskriptorem konfigurace.
(Příklad pro 2.konfiguraci zařízení podle obrázku 2.20)

Deskriptor rozhraní (tabulka 2.5) popisuje rozdělení bran pro jednu funkci zařízení (například zařízení fungující jako fax, scanner a tiskárna má tři deskriptory rozhraní). Deskriptor obsahuje informace o počtu bran patřících k rozhraní a o třídě zařízení, ke které funkce zařízení představovaná rozhraním patří.

Každá brána může být využita pouze v rámci jednoho rozhraní. Výjimkou je brána 0, která je dostupná pro celé zařízení a proto se neuvažuje do počtu bran v deskriptoru rozhraní a není ani popsána vlastním deskriptorem brány. Pro větší flexibilitu, je možné definovat alternativní nastavení pro každé rozhraní, tím lze měnit vlastnosti bran za běhu, bez vlivu na ostatní rozhraní.

Tabulka 2.5 - Struktura deskriptoru rozhraní.

Jméno pole	Ofset[B]	Délka[B]	Popis
bLength	0	1	Velikost deskriptoru.
bDescriptorType	1	1	Kód deskriptoru rozhraní (=4).
bInterfaceNumber	2	1	Identifikátor rozhraní. První rozhraní v konfiguraci má identifikátor roven 0.
bAlternateSetting	3	1	Identifikátor alternativního nastavení rozhraní.
bNumEndpoints	4	1	Počet bran používaných rozhraním bez řídící brány.
bInterfaceClass	5	1	Identifikátor třídy zařízení. Hodnota 255 znamená, že rozhraní nepatří k žádné definované třídě.
bInterfaceSubClass	6	1	Identifikátor podskupiny třídy zařízení.
bInterfaceProtokol	7	1	Identifikátor protokolu závislý na třídě a podskupině třídy zařízení.
iInterface	8	1	Index textového řetězce popisující toto rozhraní nebo nula, pokud není rozhraní popsáno.

Deskriptor brány (tabulka 2.6) popisuje vlastní nastavení brány, tj. typ a adresu brány, největší možnou velikost datového paketu, který lze branou najednou přenést a pro časovaný typ brány také periodu přenosu. Pro izochronní typ brány hostitel používá hodnotu v poli maximální velikosti dat při časovém rozvrhování paketů v rámci.

Tabulka 2.6 - Struktura deskriptoru brány.

Jméno pole	Ofset[B]	Délka[B]	Popis
bLength	0	1	Velikost deskriptoru.
bDescriptorType	1	1	Kód deskriptoru brány (=5).
bEndpointAddress	2	1	Adresa brány v rámci zařízení : D3-D0: Číslo brány D7: Směr toku dat kanálem brány 0 = od hostitele do zařízení 1 = ze zařízení k hostiteli
bmAttributes	3	1	Vlastnosti brány : D1, D0: Typ brány 0 = Řídící 1 = Izochronní 2 = Pro objemná data 3 = Časovaná
wMaxPacketSize	4	2	Maximální velikost dat v jednom přenosu.
bInterval	6	1	Interval dotazování na data nebo zasílání dat pro časovaný typ brány. Pro izochronní typ brány musí být nastaven na 1. Hodnota je v milisekundách.

Deskriptor textového řetězce popisuje jednak podporované jazyky textů (tabulka 2.7), které lze ze zařízení získat požadavkem *GetDescriptor* a pak texty samotné (tabulka 2.8). Všechny texty jsou v kódování *UNICODE* definovaném v [4]. Texty mohou být libovolně využity v závislosti podle funkce zařízení, především jsou však určeny k přidání pro člověka srozumitelného popisu pro jednotlivé konfigurace a rozhraní. Počet podporovaných jazyků je omezen pouze maximální délkou deskriptoru a proto jich může být až 126.

Tabulka 2.7 - Deskriptor podporovaných jazyků.

Jméno pole	Ofset[B]	Délka[B]	Popis
bLength	0	1	Velikost deskriptoru.
bDescriptorType	1	1	Kód deskriptoru textu (=3).
wLANGID[0]	2	2	Identifikátor 1.podporovaného jazyka.
…	…	…	…
wLANGID[x]	N	2	Identifikátor x.podporovaného jazyka.

Tabulka 2.8 - Deskriptor textového řetězce.

Jméno pole	Ofset[B]	Délka[B]	Popis
bLength	0	1	Velikost deskriptoru.
bDescriptorType	1	1	Kód deskriptoru textu (=3).
bString	2	2*N	Textový řetězec dlouhý N znaků kódovaný UNICODE.

2.2.3.3 Struktura požadavku

Všechna zařízení musí reagovat na standardní řídící požadavky přijaté branou 0. Délka požadavku je 8 bajtů tak, aby mohl být vyslán hostitelem v rámci jednoho přenosu. Požadavek obsahuje informace pro koho je určen (zařízení, rozhraní, brána), o jaký typ požadavku se jedná (standardní, specifický pro třídu, specifický pro výrobek), kód požadavku, směr přenosu a délku dat a dva parametry závislé na konkrétním požadavku.

Tabulka 2.9 - Struktura požadavku.

Jméno pole	Ofset[B]	Délka[B]	Popis
bmRequestType	0	1	Vlastnosti požadavku : D7: Směr přenosu dat v rámci požadavku 0 = od hostitele k zařízení 1 = ze zařízení k hostiteli D6, D5: Typ požadavku 0 = standardní 1 = specifický pro třídu 2 = specifický pro výrobek D4-D0: Příjemce požadavku 0 = zařízení 1 = rozhraní 2 = brána 3 = ostatní
bRequest	1	1	Kód požadavku (pro standardní požadavky viz.tab.2.9)
wValue	2	2	1.parametr závislý na požadavku
wIndex	4	2	2.parametr požadavku, většinou index nebo ofset
wLength	6	2	Délka dat přenášených s požadavkem do/ze zařízení

Požadavky, jejich kódy a parametry jsou definovány specifikací pouze pro standardní požadavky. Jejich přehled je v následující podkapitole. Požadavky pro určitou třídu zařízení jsou definovány vždy v konkrétní specifikaci třídy.

Podle specifikace musí zařízení odpovědět na požadavek bez přenosu dalších dat do 50ms od přijetí. Pokud je součástí reakce na požadavek odeslání dat hostiteli, musí být první i každý další datový přenos uskutečněn do 500ms od předchozího přenosu. Pro požadavky s datovým přenosem do zařízení je limit pro celý přenos 5s. Pokud zařízení některý požadavek nepodporuje nebo je požadavek chybný, musí být vrácena odpověď STALL.

2.2.3.4 Přehled standardních požadavků

Standard USB definuje 11 základních požadavků, sloužících hostiteli k nastavení nebo zjištění nastavení zařízení, jejich přehled je v tabulce 2.10. Detailní přehled a popis parametrů je v kapitole 9.4 specifikace [1].

Tabulka 2.10 - Přehled standardních požadavků.

Požadavek	Kód	Popis
GetStatus	0	Zjištění stavu zařízení, rozhraní nebo brány.
ClearFeature	1	Vypnutí definované funkce zařízení.
SetFeature	3	Zapnutí definované funkce zařízení.
SetAddress	5	Nastavení adresy zařízení v rámci sběrnice.
GetDescriptor	6	Přečtení deskriptoru zařízení, konfigurace nebo textu.
SetDescriptor	7	Změna nebo přidání deskriptoru.
GetConfiguration	8	Zjištění vybrané aktuální konfigurace.
SetConfiguration	9	Nastavení aktuální konfigurace.
GetInterface	10	Zjištění výběru alternativního nastavení rozhraní.
SetInterface	11	Výběr alternativního nastavení rozhraní.
SynchFrame	12	Určení synchronizačního rámce.

Požadavkem *SetFeature*/*ClearFeature* lze zapnout/vypnout definované funkce zařízení. Pro USB verze 1.1 jsou to dvě funkce – možnost probuzení hostitele zařízením a trvalé zastavení zvolené brány.

Požadavek *GetConfiguration* vrací hostiteli číslo aktuálně nastavené konfigurace zařízení. Nastavení konfigurace provádí hostitel pomocí požadavku *SetConfiguration*, ten musí být vyslán minimálně při enumeraci, aby se zařízení dostalo do zkonfigurovaného stavu zařízení.

Požadavkem *SetInterface* lze vybrat jiné než základní nastavení určitého rozhraní, pokud to zařízení umožňuje. *GetInterface* vrací hodnotu vybraného alternativního nastavení pro určité rozhraní.

GetDescriptor vrací deskriptor zařízení, textového řetězce nebo konfigurace spolu s dalšími deskriptory patřícími hierarchicky pod zvolenou konfiguraci. *SetDescriptor* umožňuje nastavit jakýkoliv deskriptor nebo přidat nový. Tento požadavek většina zařízení nepodporuje, protože má pevně dané funkce a tím i své deskriptory.

Požadavek *GetStatus* vrací stav zařízení, rozhraní nebo brány v definovaném tvaru. Lze tak u zařízení zjistit, zda je napájeno ze sběrnice a zda může budit hostitele. Pro určitou bránu pak lze zjistit, zdali je aktivní.

Důležitý je požadavek *SetAddress*, kterým hostitel nastaví adresu zařízení v rámci sběrnice. Zařízení po potvrzení tohoto požadavku změní svoji adresu a další požadavky jsou přijímány na nové adrese. Tímto požadavkem se zařízení dostane ze stavu obecného zařízení do adresovaného stavu.

Posledním požadavkem je *SynchFrame*, ten je používán pro nastavení synchronizačního rámce pro izochronní přenos.

Všechna zařízení připojovaná ke sběrnici USB, zpracovávají pouze pro ně relevantní požadavky. Minimem, které je nezbytné pro každé zařízení, jsou požadavky *SetAddress*, *SetConfiguration*, *GetDescriptor* a případně *SetInterface*.

3. Integrované obvody pro USB

Nejčastěji je rozhraní zařízení a sběrnice USB spolu s dalšími funkcemi tvořeno jedním integrovaným obvodem. V některých případech je vhodné sestavit rozhraní z více obvodů, například s ohledem na vývoj nebo ostatní obvody v zařízení. Tato kapitola je proto věnována přehledu typů dostupných obvodů, z kterých se rozhraní USB skládá. Pro každý typ je v samostatné podkapitole stručně popsána funkce obvodu a jsou také uvedeni významnější výrobci zmiňovaného typu obvodu.

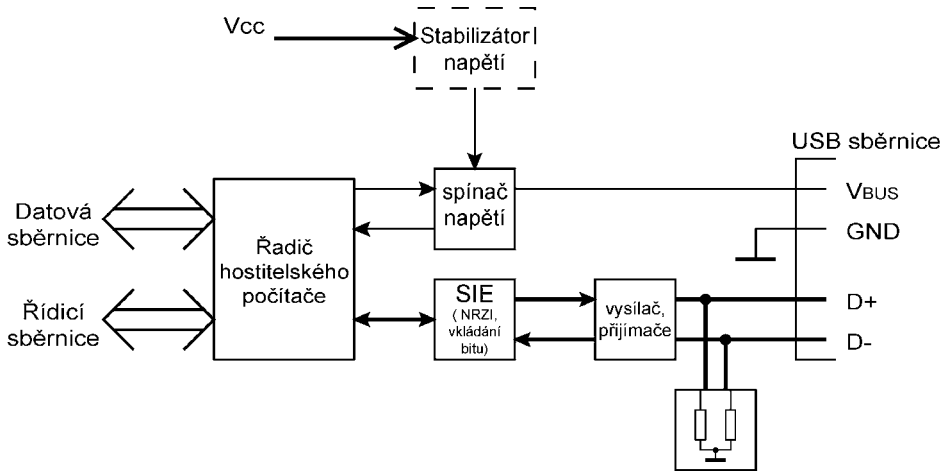
S touto kapitolou úzce souvisí příloha A, kde je jednak přehledový katalog jednotlivých obvodů pro implementaci USB a také podrobnější přehledový katalog řadičů a mikrokontrolérů pro USB. Vzhledem k velikosti trhu nejsou v katalogu uvedeny všechny obvody ani výrobci.

Podklady pro tuto kapitolu pocházejí z katalogových listů jednotlivých výrobců, jejichž seznam se nachází na konci přílohy A spolu s odkazy na jejich internetové stránky, kde lze získat více informací o uvedených obvodech.

3.1 Prvky sběrnice USB

Na sběrnici USB mohou být připojeny tři typy zařízení – hostitel, rozbočovač a koncové zařízení. Každý typ se skládá z několika bloků, které mohou být implementovány samostatně nebo dohromady v jednom integrovaném obvodu.

3.1.1 Hostitel



Obrázek 3.1 - Blokové schéma rozhraní USB hostitele s jedním výstupním portem.

Rozhraní hostitele je, z hlediska datové části sběrnice, většinou tvořeno jedním integrovaným obvodem zahrnujícím v sobě všechny potřebné bloky USB portu: blok vysílače a přijímačů (*Transceiver*), blok sériového rozhraní SIE (*Serial Interface Engine*) a blok řadiče (*Host Controller*), které znázorňuje obrázek 3.1. Napájení sběrnice je připojeno ke zdroji napětí přes napěťový spínač (*Power Switch*) řízený řadičem rozhraní. Popis jednotlivých bloků je v samostatných podkapitolách dále v textu.

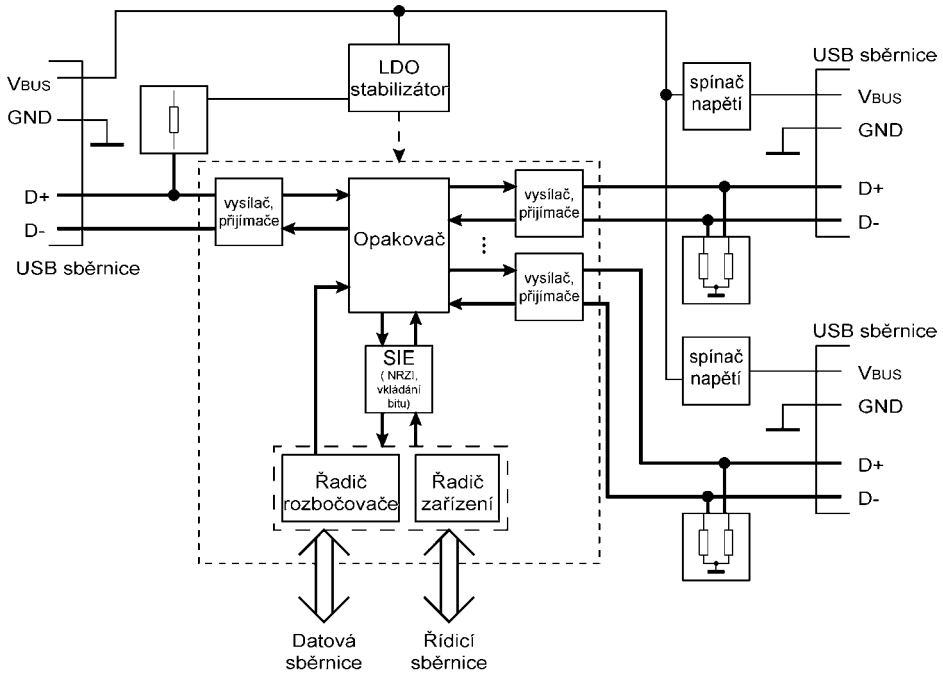
Řadič hostitele zajišťuje obsluhu sběrnice USB podle požadavků nadřízené vrstvy. Při detekci nového zařízení zajistí enumeraci zařízení a přidělení unikátní adresy, rozděluje přenosovou kapacitu sběrnice, detekuje chyby. Existují tři typy řadičů s odlišnou vnitřní

strukturou – *Open Host Controller*, *Universal Host Controller* a pro USB 2.0 *Enhanced Host Controller*.

Výrobci základních desek počítačů, kde je rozhraní hostitele spolu s dalšími funkcemi integrován do větších obvodů, používají převážně zakázkové obvody, které nejsou běžně samostatně dostupné. Proto je nabídka obvodů jen s řadičem hostitele USB malá. Většina takových obvodů tvoří rozhraní mezi sběrnicí PCI a USB, ty vyrábí například firmy ALi, OPTi, NEC, Philips. Jedinými nalezenými dostupnými obvody, které jsou určeny pro přímé připojení k mikroprocesorům jsou obvod ISP1160 firmy Philips a obvod SL811HS firmy Cypress. Firma Philips vyrábí ještě další obvod ISP1161, který má kromě hostitelského řadiče i druhý port s řadičem koncového zařízení.

3.1.2 Rozbočovač

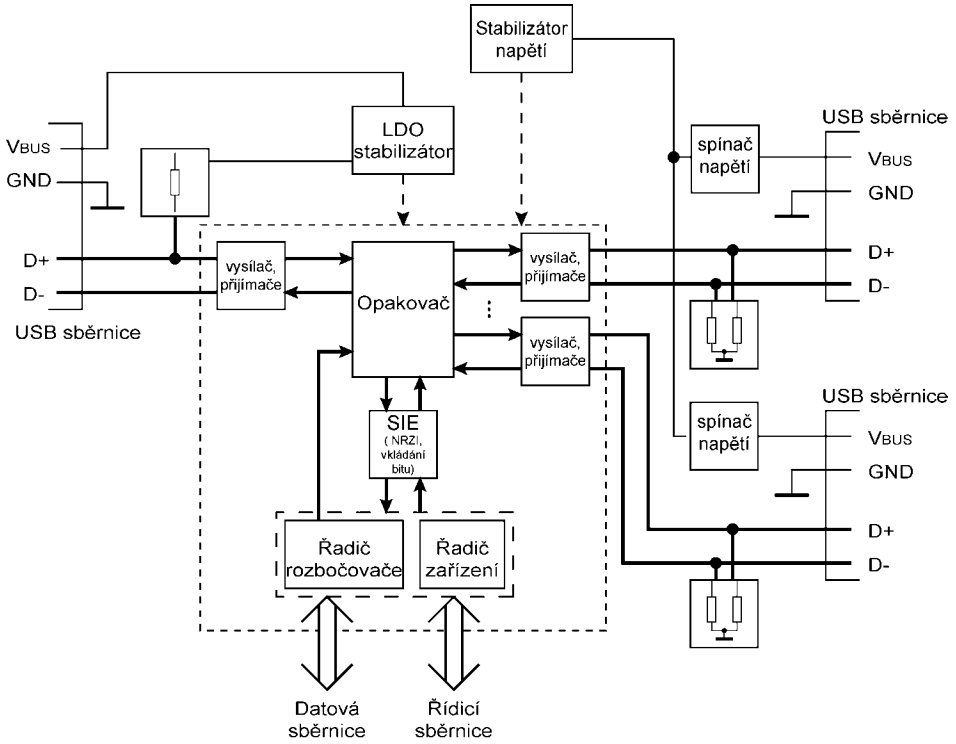
Typická struktura rozbočovače napájeného ze sběrnice je na obrázku 3.2 a napájeného z externího zdroje na obrázku 3.3.



Obrázek 3.2 - Blokové schéma rozhraní USB rozbočovače napájeného ze sběrnice.

Každý obvod s implementovaným rozbočovačem obsahuje blok vysílače a přijímačů pro připojení k vyšší vrstvě sběrnice, bloky vysílačů a přijímačů pro připojení zařízení z nižší vrstvy a mezi nimi důležitý blok, kterým je opakovač (*Repeater*). Ten se stará o připojování a odpojování výstupních portů pro LS zařízení, předávání paketů a správné chování připojených zařízení při zastavení sběrnice (*suspend*) a vzdáleném buzení hostitele zařízením. Obvod dále obsahuje blok sériového rozhraní SIE a řadič rozbočovače. Většina dostupných obvodů má implementováno i koncové zařízení, které je připojeno k jednomu z výstupů rozbočovače a tvoří tak složené zařízení.

Z hlediska napájení je pro každý výstupní port potřeba napěťový spínač. Pokud je rozbočovač napájen ze sběrnice, ale potřebuje nižší napětí, je použit stabilizátor napětí s malým úbytkem napětí (*Low DropOut regulator*).



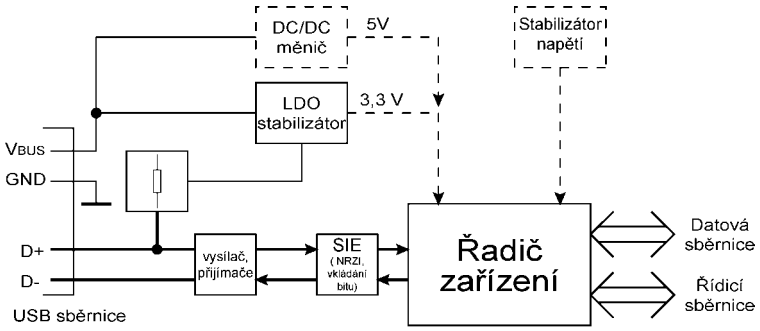
Obrázek 3.3 - Blokové schéma rozhraní USB rozbočovače napájeného z externího zdroje.

Nabídka dostupných obvodů na trhu je výrazně vyšší než je u obvodů s funkcí hostitele. Všichni výrobci, produkující obvody pro koncové zařízení, mají ve své nabídce alespoň jeden obvod s funkcí rozbočovače. V mnoha případech jde o obvody plnící stejnou funkci koncového zařízení, u nichž je navíc implementován rozbočovač. Nabídka takových obvodů je asi dvakrát větší než nabídka obvodů pouze s funkcí rozbočovače. Část dostupných obvodů má neměnnou funkci, například obvody pro klávesnice. Zbytek jsou mikrokontroléry, většinou osmibitové, s programovatelnou funkcí.

Většina obvodů má čtyři porty pro připojení zařízení z nižší vrstvy, ale jsou dostupné i obvody s dvěma, třemi a pěti porty. Sedm portů jako maximum u dostupných obvodů má TUSB2077A firmy Texas Instruments a FT8U100AX firmy FTDI, který navíc obsahuje složené zařízení. Mezi další výrobce patří firmy Atmel, Cypress, Mitsubishi, Motorola, NEC, Philips.

3.1.3 Koncové zařízení

Nejvíce dostupných obvodů pro USB jsou obvody koncového zařízení. Typická struktura rozhraní USB takového obvodu je na obrázku 3.4.



Obrázek 3.4 - Blokové schéma rozhraní USB koncového zařízení.

Blok řadiče zařízení se stará o příjem a dekódování paketů, zpracování řídících požadavků a extrakci dat z paketů, které ukládá do pamětí FIFO, odkud jsou čtena dalšími obvody koncového zařízení. Opačně zajišťuje řadič vysílání dat v rámci datového paketu.

Zařízení s jednou pevnou funkcí mají řadič většinou tvořen i s dalšími obvody hardwarově. Pokud je rozhraní USB implementováno v obvodu spolu s mikrokontrolérem je hardwarově tvořen pouze blok vysílače/přijímače, SIE a nejnižší vrstva řadiče. O zpracování řídících požadavků se stará program mikrokontroléru.

Obvody s rozhraním USB mají různé napájení, většinou je to 3,3V nebo 5V. Napětí 3,3V je vyžadováno i pro připojení pull-up rezistoru, jímž se detekuje připojení zařízení na sběrnici a určení jeho typu (LS/FS). Proto má většina obvodů implementovaný LDO stabilizátor s tímto napětím. Obvody s mikrokontrolérem jsou často vybaveny programově ovládaným spínačem pro připojení pull-up rezistoru R_{PU}, který je u mnoha obvodů také implementován. Pro obvod napájený 5V by měl být zapojen DC/DC měnič, ale většina takových obvodů může pracovat i s nižším napětím, které je zaručeno specifikací USB a proto se měnič nepoužívá.

Nabídka obvodů s implementovaným rozhraním USB koncového zařízení je v porovnání s obvody, které mají pouze rozhraní rozbočovače (bez složeného zařízení), asi osmkrát větší. Přibližně 90% těchto obvodů má implementovaný mikrokontrolér, zbytek jsou obvody mající pouze řadič a případně nezbytně nutnou část mikrokontroléru, ty pak slouží jako rozhraní mezi sběrnicí USB a paralelní nebo sériovou sběrnicí. Vyrábí je například firmy Ali, Epson, FTDI, NEC, Philips, SMSC.

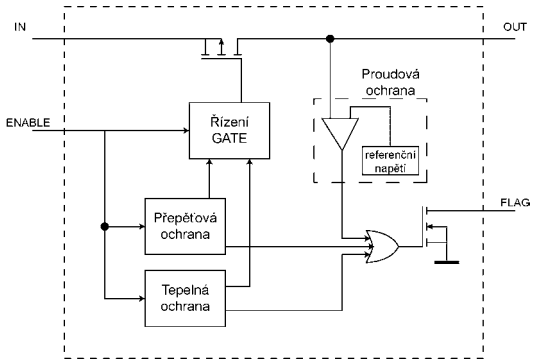
Obvody s mikrokontrolérem lze dále rozdělit na obvody s pevnou funkcí nebo jednorázově programovatelné obvody, nejčastěji určené pro počítačové myši, herní ovladače a klávesnice. Největším výrobcem takových obvodů je firma Cypress. Druhou skupinou jsou reprogramovatelné mikrokontroléry, které vyrábí například firmy Atmel, Cypress, Hitachi, Mitsubishi, Motorola a další.

3.2 Typy obvodů pro USB

Kromě obvodů, které integrují všechny bloky rozhraní USB, jsou dostupné obvody představující samostatné bloky. V následujících podkapitolách jsou stručně popsány základní funkce takových obvodů.

Nejsou zde zmíněny pouze LDO stabilizátory, které má většina řadičů zařízení integrována v jednom obvodu. Tento stabilizátor vytváří specifikací potřebné napětí 3,3V pro pull-up rezistor R_{PU} a budiče vysílačů.

3.2.1 Obvody napěťových spínačů



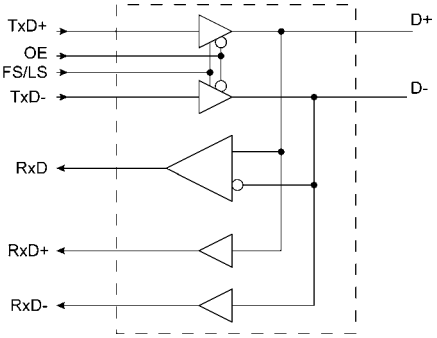
Obrázek 3.5 - Blokové schéma napěťového spínače pro USB.

Na obrázku 3.5 je zobrazena typická struktura napěťového spínače pro USB. Všechny dostupné obvody mají jako spínací prvek MOSFET tranzistor, většinou s kanálem N. Přechodový odpor je nejčastěji kolem 100mΩ, výjimečně i pod 40mΩ.

Tento spínací prvek je řízen vnějším signálem ENABLE z řadiče rozbočovače nebo hostitele a zároveň je podmíněn tepelnou a přepěťovou ochranou. Ty jsou spolu s proudovou ochranou zkombinovány a vyvedeny na výstup FLAG, kterým je řadič informován o chybě.

Mezi výrobce, kteří nabízejí tento typ obvodů výslovně pro USB, patří firmy Micrel, National Semiconductor a Texas Instruments.

3.2.2 Obvody vysílače a přijímačů



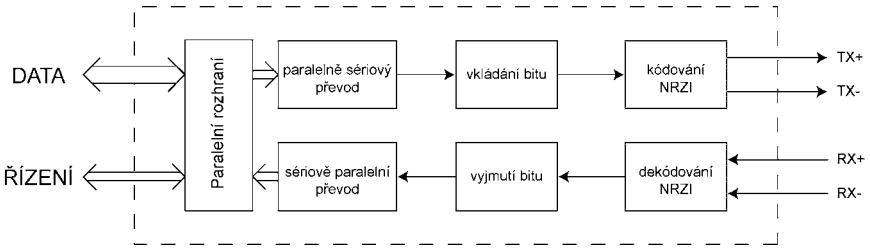
Obrázek 3.6 - Vnitřní schéma vysílače/přijímačů pro LS/FS zařízení.

Obvod s implementovaným blokem vysílače a přijímačů musí na straně sběrnice USB splňovat celou řadu parametrů, popsaných ve specifikaci USB (kapitola 2.2.1.4 nebo podrobněji kapitola 7.1 v [1]).

Na straně připojení k dalším blokům obsahují obvody přizpůsobení úrovní signálů tak, aby byly přímo připojitelné k mikrokontrolérům, signálovým procesorům (*DSP*), zákaznickým obvodům (*ASIC*) nebo programovatelným polím (*FPGA*), v kterých jsou implementovány zbylé bloky rozhraní USB.

Z výrobců produkujících samotný blok vysílače/přijímačů nabízí nejvíce typů firma Philips, dále jsou to například firmy Cypress, Fairchild, Maxim, Micrel, Sipex.

3.2.3 Obvody fyzické vrstvy

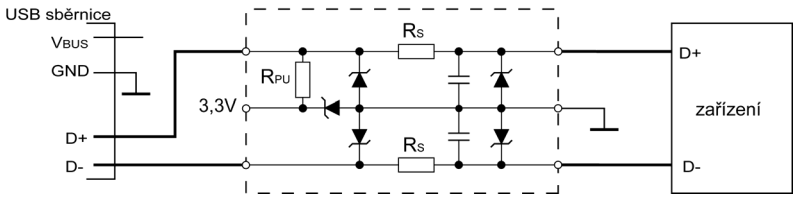


Obrázek 3.7 - Blokové schéma SIE.

Blok SIE (*Serial Interface Engine*) na obrázku 3.7, který plní funkci sério-paralelního rozhraní s vkládáním bitu a kódováním NRZI, se samostatně nevyrábí. SIE je buď součástí obvodu řadiče pro zařízení FS/LS nebo je v jednom obvodu spolu s blokem vysílače/přijímačů pro HS zařízení – pak se jedná o obvod fyzické vrstvy (PHY). Obvod má specifikované výstupní rozhraní nazvané UTMI (USB Transceiver Macrocell Interface).

Výrobcem tohoto obvodu je například firma Lucent Technologies nebo SMSC.

3.2.4 Ostatní obvody



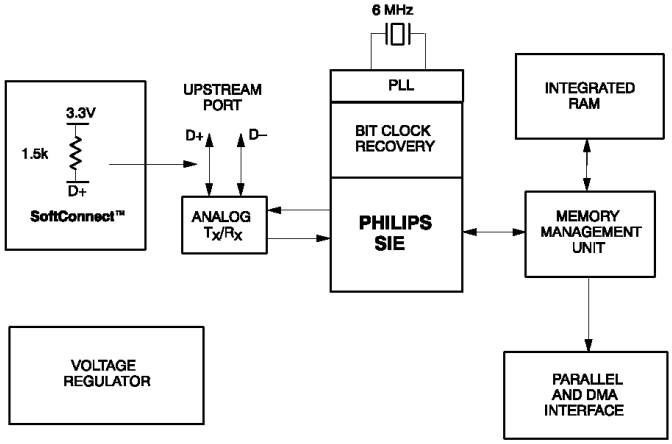
Obrázek 3.8 - Vnitřní schéma obvodu pro ochranu před elektrostatickým výbojem.

Mezi ostatní patří jednak speciální obvody připojované na sběrnici USB, jako je například obvod pro nabíjení Li-Ion baterií firmy Linear Technology. A pak především obvody ochrany před elektrostatickým nábojem. Na obrázku 3.8 je vnitřní schéma takového obvodu od firmy ON Semiconductor, který navíc obsahuje zakončovací rezistory, filtr a pull-up rezistor. Obvod se připojí přímo mezi obvod vysílače/přijímače a konektor sběrnice USB a tak nejsou potřeba žádné další pasivní součástky.

3.3 Zvolený obvod PDIUSBD12

Vzhledem k požadavkům v zadání, dostupnosti a svým možnostem byl zvolen obvod PDIUSBD12 firmy Philips. Jedná se o řadič USB určený pro FS zařízení, tvořící rozhraní mezi sběrnicí USB a paralelní sběrnicí mikrokontroléru s možným DMA přenosem dat.

Obvod obsahuje na straně připojení k USB tři brány a maximálně šest kanálů s několika možnými nastaveními. První brána s pevnou funkcí je dvoukanálová řídící brána s velikostí dat 16 byte. Druhá brána má dva kanály s velikostí dat 16 byte a programově volitelným typem brány mezi časovaným přenosem a přenosem objemných dat. Třetí brána může mít buď dva kanály s velikostí 64 byte pro obousměrný časovaný přenos, izochronní přenos nebo pro přenos objemných dat nebo může mít jen jeden kanál s velikostí 128 byte pro jednosměrný izochronní přenos.



Obrázek 3.9 - Blokové schéma obvodu PDIUSBD12.

Jak je patrné z obrázku 3.9, obvod obsahuje blok vysílače/přijímačů, blok SIE, FIFO paměť pro jednotlivé kanály USB, paralelní rozhraní, stabilizátor 3,3V pro připojení pull-up rezistoru R_{PU} a programově spínaný rezistor R_{PU}, umožňující softwarově řízené připojení a odpojení zařízení od sběrnice USB. Jeden z výstupů obvodu je určen k indikaci stavu připojení ke sběrnici USB pomocí LED. Dalším využitelným výstupem obvodu je

programovatelný hodinový signál až do frekvence 48MHz, vytvářený fázovým závěsem z externího hodinového signálu 6MHz.

Obvod lze připojit k mikrokontroléru multiplexovanou i nemultiplexovanou sběrnicí pomocí osmi datových signálů a řídících signálů A0, ALE, CS, WR, RD a INT. PDIUSBD12 vyvolává externí přerušení pomocí devíti zdrojů. Přerušení může způsobit odeslání dat, přijmutí dat, reset sběrnice, změna režimu sběrnice (*suspend mode*) a konec DMA přenosu.

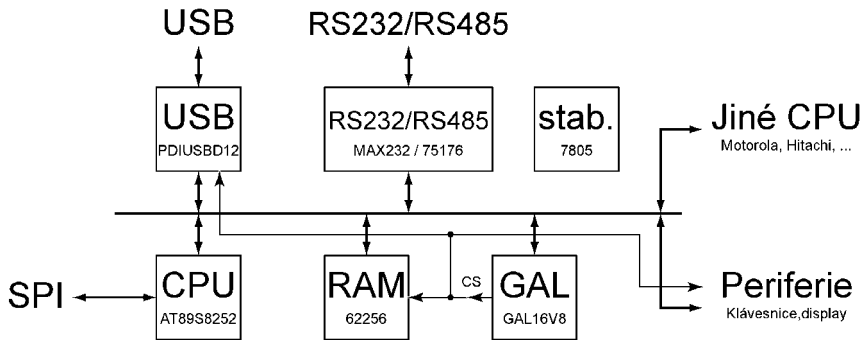
Podrobné informace lze nalézt v katalogovém listu obvodu, který je v příloze B nebo na internetových stránkách firmy Philips.

4. Hardware

V této kapitole jsou stručně popsány hardwarové prostředky vytvořené v rámci diplomové práce. Dokumentace k navrhnuté desce rozhraní iD12 je v příloze C.

4.1 Deska rozhraní iD12

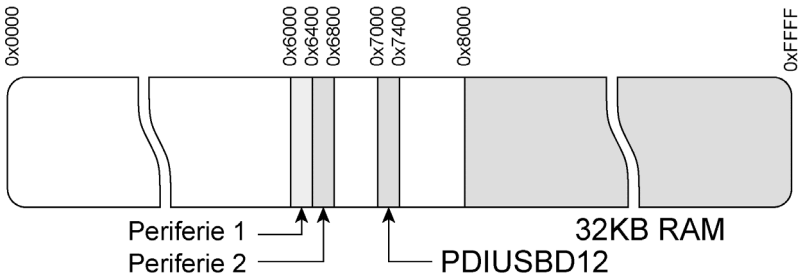
Deska rozhraní USB (dále jen iD12) byla navržena s ohledem na jednoduchý vývoj aplikací s rozhraním USB. Blokové schéma desky je na obrázku 4.1.



Obrázek 4.1 - Blokové schéma vývojové desky iD12.

4.1.1 Část CPU

Jako mikrokontrolér je použit obvod AT89S8252 firmy Atmel (dále jen CPU), který je kompatibilní s x52 a obsahuje 8KB Flash paměti reprogramovavatelné v systému pomocí SPI, 2KB paměti EEPROM pro uživatelská data a 256 byte paměti RAM. K CPU je připojena externí paměť RAM o velikosti 32KB. Tuto paměť lze využít jak pro programový kód, tak pro data, podle vyvíjené aplikace. CPU má datovou sběrnici multiplexovanou s dolní částí adresové sběrnice, proto je na desce obvod 74HC573 obsahující 8 záchytných registrů řízených signálem ALE, které slouží k zachycení spodní poloviny adresy. Aby mohly jednotlivé obvody komunikovat s CPU po společné sběrnici, je třeba generovat řídící signály pro výběr obvodu v adresovém dekodéru, vytvořeném funkcemi obvodu GAL16V8. Naprogramování obvodu určuje mapu externího adresového prostoru CPU znázorněnou na obrázku 4.2. Celou mapu lze jednoduše změnit přeprogramováním obvodu GAL.



Obrázek 4.2 - Mapa externího adresového prostoru CPU.

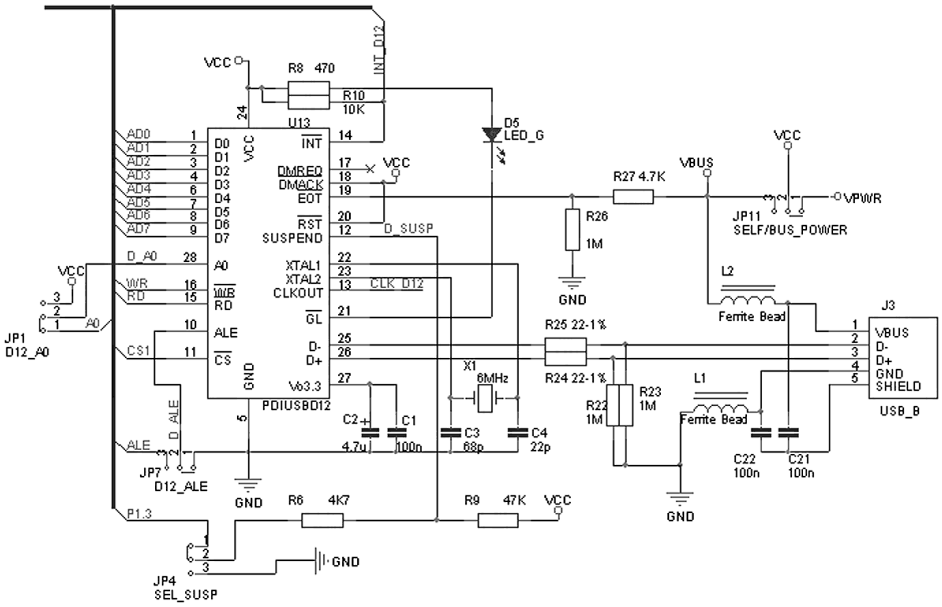
Kromě čtyř výstupů CPU jsou všechny využity pro datovou, adresovou a řídící sběrnici. Volné výstupy jsou připojeny na LED a mohou tak sloužit vyvíjeným aplikacím k vlastní indikaci.

4.1.2 Rozhraní RS232/RS485

Deska umožňuje zapojení obou rozhraní RS232 i RS485. Protože se však nepočítá s využitím obou rozhraní najednou, jsou vyvedena na společný konektor a jejich výběr se řídí vložením jednoho z budičů ICL232 pro RS232 a 75176 pro RS485 do příslušné patice na desce. Celá část rozhraní je od zbytku desky galvanicky oddělena pomocí optočlenů na signálových vodičích a pomocí DC/DC měniče na napájení.

4.1.3 Rozhraní USB

Rozhraní USB tvoří obvod PDIUSBD12 spolu s několika pasivními součástkami (viz.obrázek 4.3). Jeho zapojení na straně USB je převzato z výrobcem doporučeného zapojení obvodu. Signály D+ a D- ze sběrnice jsou připojeny přes zakončující rezistory přímo k obvodu. Pro zmenšení šumu při odpojené sběrnici jsou k oběma signálům připojeny pull-down rezistory na nulové napětí. Sběrnicové napájení je připojeno přes propojku JP1 k napájení desky a lze tak vyvíjet aplikace s vlastním i sběrnicovým napájením.



Obrázek 4.3 - Schéma zapojení rozhraní USB.

Ze strany mikrokontroléru je obvod připojen datovou sběrnicí a jedním adresovým vodičem pro připojení bez multiplexované datové sběrnice. Při připojení obvodu pomocí multiplexované datové sběrnice se využívá signál ALE namísto signálu A0. Ostatní řídící signály – CS pro výběr obvodu, RD pro čtení dat z obvodu, WR pro zápis dat do obvodu a INT pro požadavek obvodu o přerušení – jsou připojeny pro obě varianty k CPU.

U vyvíjených aplikací nebylo počítáno s využitím DMA přenosu, proto jsou k tomu určené vývody nevyužity. Naopak je využita možnost obvodu připojit indikační LED, ta informuje o připojení k USB. Pokud je zařízení úspěšně připojeno a zkonfigurováno, LED trvale svítí. Při enumeraci a přenosu paketů LED bliká.

4.1.4 Napájení, konektory a propojky

Deska je napájena buď ze sběrnice USB nebo z externího zdroje připojeného na konektor J6 přes stabilizátor 5V řady 7805, proto musí být přivedené napětí o něco vyšší. Součástí napájení je i sada blokovacích kondenzátorů pro jednotlivé integrované obvody.

Na desce se nachází šest konektorů rozmístěných po obvodu desky a označených J1-J6. Jeden pro přívod externího napájení, další pro připojení ke sběrnici USB, pro rozhraní RS232

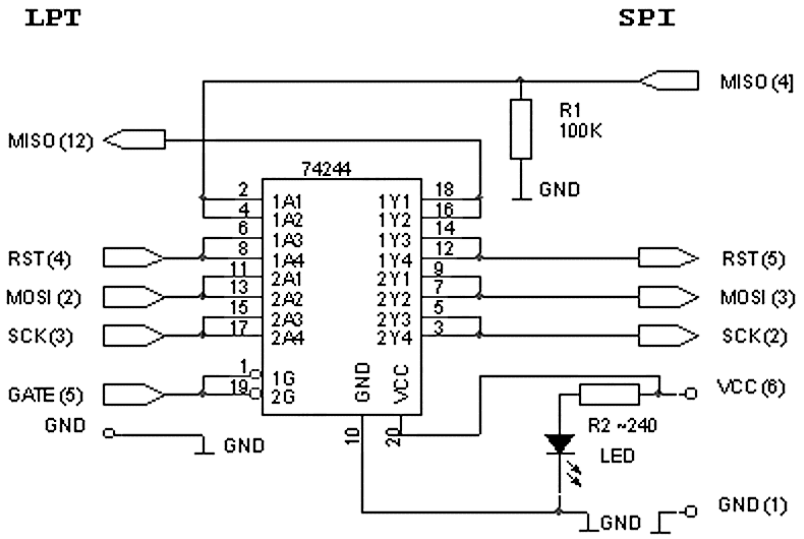
a RS485, pro připojení CPU přes SPI, pro připojení periferií a konektor pro připojení externích CPU k rozhraní USB. Popis vývodů jednotlivých konektorů je v příloze C. Konektor pro připojení periferií má dva signály CS pro výběr obvodu a umožňuje tak připojení dvou nezávislých zařízení – například klávesnice a zobrazovače.

Konektorem pro připojení externích CPU lze přímo k rozhraní USB připojit jakýkoliv jiný systém, který má osmibitovou datovou sběrnici, jeden adresní vodič a řídící signály RD, WR, CS a INT. Při připojení externího systému je nutné vyjmout z desky ostatní obvody CPU, GAL a externí RAM, které jsou pro ten účel osazeny na desce v paticích. Konektor sám je přímo navržen pro desku MO_CPU s procesorem Motorola 68376 firmy PiKRON. Její dokumentaci lze nalézt na internetové adrese URL:<http://www.pikron.com>.

Protože se jedná o vývojovou desku, je na desce 13 zkratovacích propojek označených JP1-JP13, které poskytují větší flexibilitu pro různé aplikace. Jejich popis a základní nastavení je také v příloze C.

4.2 SPI programátor CPU

K programování interní Flash a EEPROM paměti mikrokontroléru Atmel je využito rozhraní SPI připojené k paralelnímu portu počítače přes obvod budičů signálů. Takto zapojený budič umožňuje řízení průběhu programování z počítače. V rámci diplomové práce byl napsán i software pro programování mikrokontroléru Atmel, zmíněný na konci následující kapitoly. Programovací software používá programátor zapojený podle obrázku 4.4.



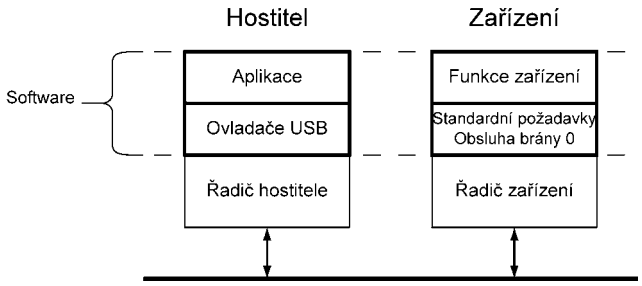
Obrázek 4.4 - Schéma zapojení programovacího budiče SPI rozhraní.

5. Softwarové vybavení

V této kapitole jsou nejprve popsány jednotlivé vrstvy programového vybavení na obou stranách sběrnice potřebného pro správnou funkci zařízení. Dále jsou zde uvedeny a stručně popsány použité softwarové nástroje. Na závěr kapitoly jsou popsány vytvořené jednotlivé části programového vybavení.

5.1 Vrstvy softwarového vybavení

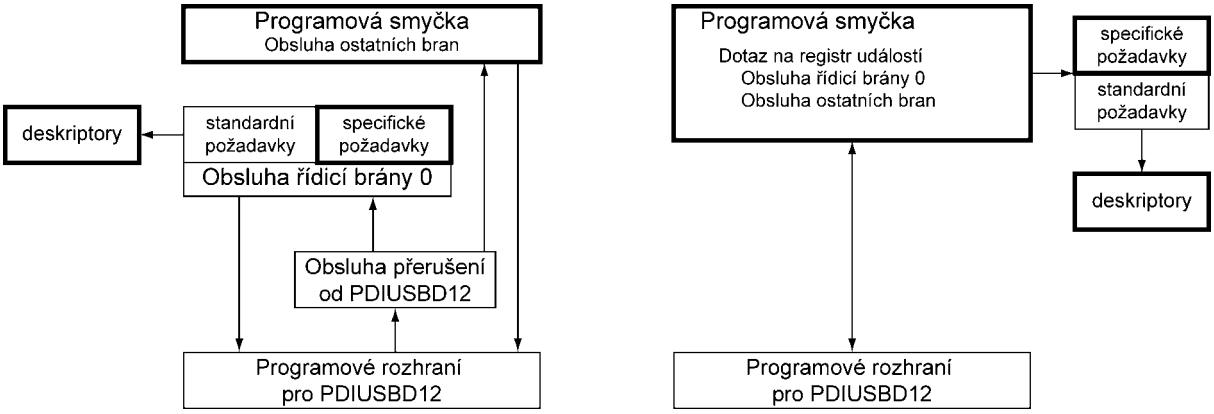
Tato podkapitola popisuje softwarové vrstvy modelu USB znázorněné a zvýrazněné na obrázku 5.1. Popis jednotlivých vrstev programového vybavení je proveden podle použitých součástí, na straně zařízení pro obvod PDIUSBD12 a na straně hostitele pro počítač s operačním systémem Microsoft Windows 2000.



Obrázek 5.1 - Model sběrnice USB.

5.1.1 Programové vybavení na straně zařízení

Celou část programu pro rozhraní USB lze rozdělit na několik úrovní, podle toho v jakém režimu komunikace s mikrokontrolérem obvod rozhraní využíváme. Jsou možné dva režimy. Prvním je využití přerušení, které obvod vyvolává při nějaké události na sběrnici USB. Druhým je režim dotazování (*polling*), kdy mikrokontrolér zjišťuje stav obvodu a sběrnice průběžným testováním registru událostí obvodu v programové smyčce.

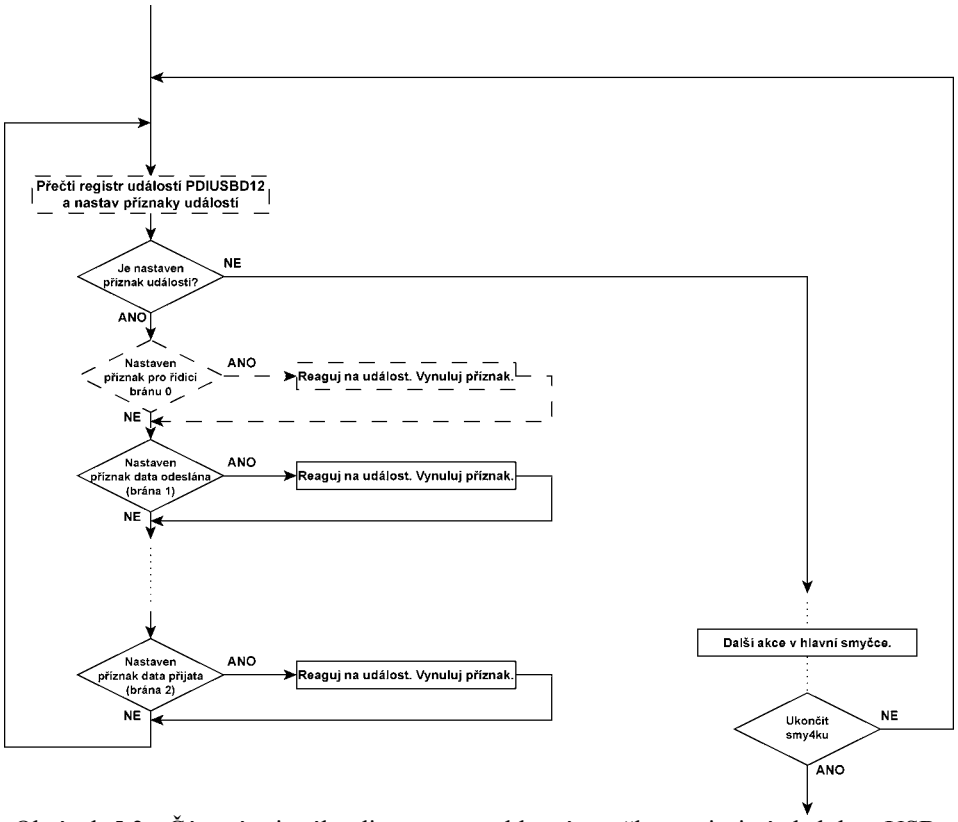


Obrázek 5.2 - Vrstvy části programu pro rozhraní USB.
a) V režimu s přerušením. b) V režimu dotazování.

Na obrázku 5.2 jsou naznačeny jednotlivé vrstvy části programu pro USB se zvýrazněnými vrstvami, které je nutné upravit nebo napsat znovu pro každé zařízení.

V obou režimech je nejspodnější vrstvou programové rozhraní pro základní komunikaci s obvodem rozhraní USB. Tato vrstva odděluje obvod od zbytku programu, který tak může být hardwarově nezávislý. Při změně obvodu rozhraní USB, stačí změnit pouze tuto vrstvu. Rozhraní zajišťuje pro vyšší vrstvy základní funkce pro práci s obvodem a sběrnicí USB. Funkce jsou složeny z elementárních příkazů pro komunikaci s obvodem podle specifikace zveřejněné výrobcem. Vrstva tak umožňuje využít funkce pro inicializaci a nastavení obvodu, jeho programového připojování a odpojování od sběrnice, výběr brány pro komunikaci, přijímání a vysílání dat z vybrané brány, zjišťování stavu brány a její nastavování a také pro čtení registru událostí obvodu.

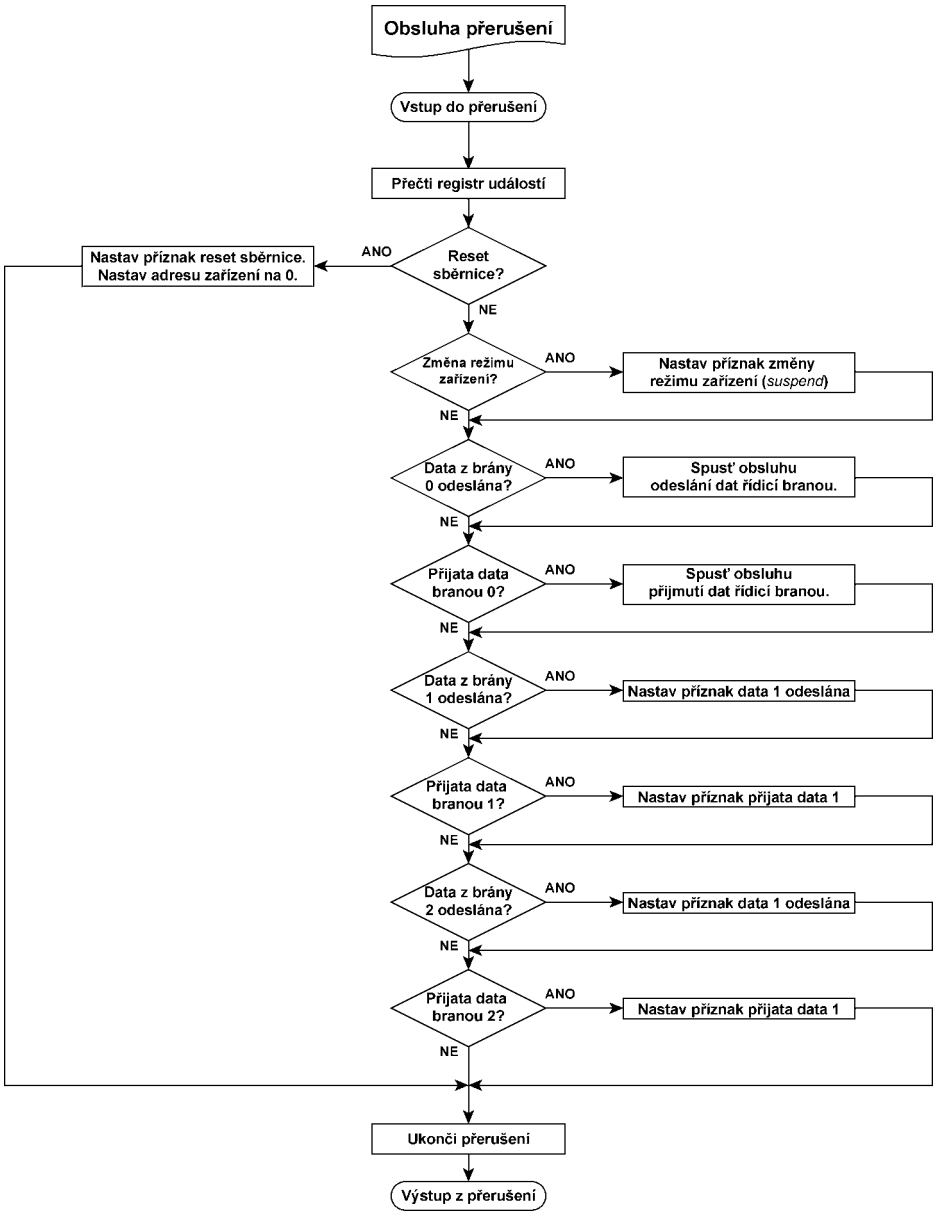
V režimu dotazování je celá obsluha komunikace USB v programové smyčce, na rozdíl od režimu s využitým přerušením. Na obrázku 5.3 je znázorněn vývojový diagram pro část smyčky programu s prioritní obsluhou rozhraní USB, kde jsou přerušovanou čarou zobrazeny kroky potřebné při dotazovacím režimu. Ve smyčce je nutné nejprve zjistit, zda došlo k nějaké události v obvodu rozhraní. Pokud ne, pokračuje program vykonáváním dalších kroků ve smyčce. Pokud ano, jsou vykonány jednotlivé kroky obsluhy pro všechny kanály, které měli nastaven příznak události, jinak program pokračuje vykonáváním dalších kroků ve smyčce. Obsluha jednotlivých bran je, kromě řídicí brány 0 popsané dále, závislá na konkrétní aplikaci, proto zde nebude uvedena. Po zpracování události od rozhraní USB program pokračuje novým testováním registru příznaků, čímž je zvýhodněna komunikace přes USB na úkor ostatních částí smyčky.



Obrázek 5.3 - Část vývojového diagramu pro hlavní smyčku s prioritní obsluhou USB. (Části kreslené přerušovanou čarou jsou pouze pro režim dotazování.)

Při využití přerušení je součástí programové výbavy vrstva obsluhy přerušení. Vývojový diagram obslužné rutiny je na obrázku 5.4. Z něj je patrné, že všechny události, kromě události resetu zařízení ze sběrnice a obsluhy brány 0, pouze nastavují svůj příznak a jejich obsluha je v programové smyčce. To je dáno univerzálností programu tak, aby nebylo nutné

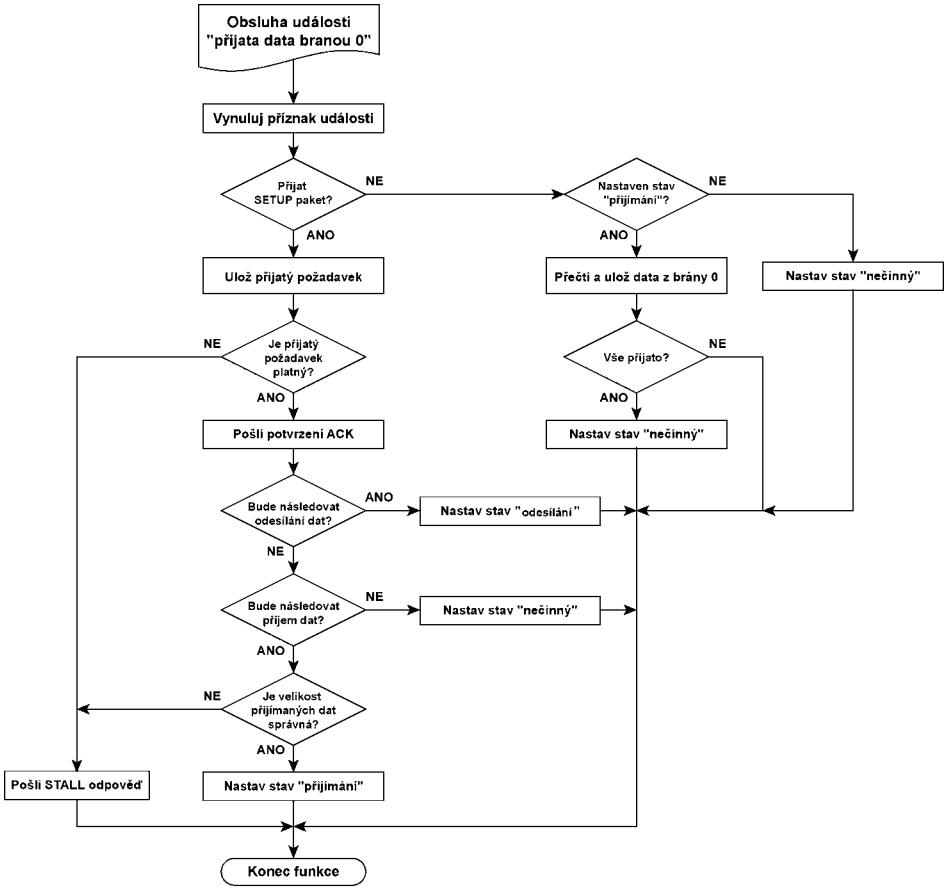
zasahovat do programu obsluhy přerušení. Také se tím předchází problémům, které mohou nastat při příchodu asynchronního přerušení a práci se sdílenými daty, které by přerušení zpracovávalo.



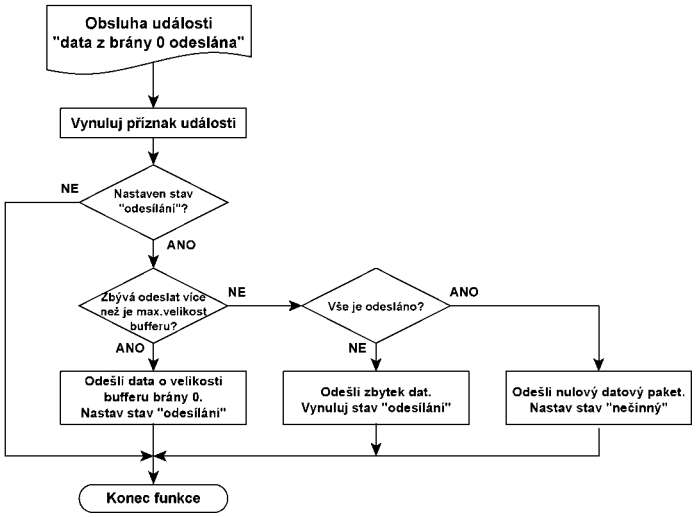
Obrázek 5.4 - Vývojový diagram obsluhy přerušení pro obvod PDIUSBD12.

Při resetu sběrnice je podle specifikace zařízení nastaveno do režimu obecného zařízení s adresou 0 a je nutná jeho nová konfigurace hostitelem. Tím se zároveň zastaví všechny probíhající přenosy přes USB.

Obsluha řídicí brány 0 je v samostatné vrstvě, protože zpracovává řídicí požadavky zaslané hostitelem a je stejná pro všechny aplikace, s výjimkou požadavků specifických pro aplikaci. Vývojový diagram pro přijímací kanál řídicí brány 0 je na obrázku 5.5 a pro vysílací kanál na obrázku 5.6. Protože může být řídicí přenos dvoufázový nebo jednofázový, je přenos řízen jako stavový automat. Z nečinného stavu se brána 0 dostane do stavu přijímání nebo vysílání dat po příjmu konfiguračního požadavku s nastaveným požadavkem na přenos dat.



Obrázek 5.5 - Vývojový diagram obsluhy vstupního kanálu řídicí brány.



Obrázek 5.6 - Vývojový diagram obsluhy výstupního kanálu řídicí brány.

Po přijmu dat je proto nejprve testováno, zda se jedná o konfigurační požadavek. Pokud ano, je požadavek uložen a otestován, zda ho umí zařízení zpracovat. Jde-li o neznámý požadavek, brána 0 je pozastavena a hostiteli tak přijde odpověď STALL. Umí-li zařízení požadavek zpracovat je odesláno potvrzení a je testováno zda je součástí požadavku datová část a případně v jakém směru a podle toho je nastaven stav brány. Poté se přijmou nebo odešlou všechna data, která jsou součástí požadavku. Nakonec je brána vrácena do nečinného stavu.

Protože jsou standardní požadavky pro většinu aplikací stejné, až na různé deskriptory, je obsluha brány 0 rozdělena na tři části. Samotná obsluha a reakce na standardní požadavky je v první části, která se mění pouze u speciálních aplikací. Mezi standardní požadavky, na které zařízení nepoužívající izochronní přenos reaguje jsou požadavky GetDescriptor, SetFeature, ClearFeature, GetConfig, SetConfig, GetInterface, SetInterface, GetStatus, SetAddress.

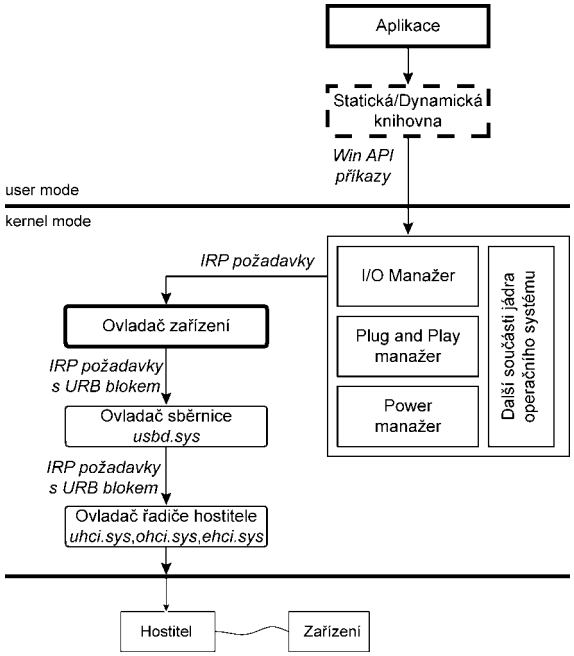
Další částí jsou deskriptory, které jsou odesílány jako reakce na požadavek GetDescriptor. Tyto deskriptory se mění s každou aplikací, proto jsou v samostatné části jako datové struktury. Poslední částí je zpracování požadavků specifických pro aplikaci (*vendor request*). Tuto část některé aplikace neobsahují, nepotřebují-li využívat vlastních řídicích požadavků.

5.1.2 Programové vybavení na straně hostitele

Programové vybavení na straně hostitele je podstatně složitější než na straně zařízení. V moderních 32bitových operačních systémech (Windows, Linux) je základní podpora USB sice zabudovaná, ale pro jednotlivé zařízení jsou zapotřebí vlastní ovladače a knihovny rozhraní mezi aplikacemi a ovladačem.

Protože je praktická práce zaměřena především na operační systém Microsoft Windows 2000, bude zde popisováno potřebné vybavení pro tento systém, zvláště pak vrstva ovladače. Vzhledem k rozsahu této problematiky, je zde celý systém pouze načrtnut, detailní informace lze nalézt v [8] a [10], odkud byly také čerpány pro vlastní práci. Především vlastní ovladače nelze bez zdroje [8] vytvářet, obsahuje popis nezbytných funkcí nabízených jádrem systému.

Programové vybavení na straně hostitele je rozděleno systémem do dvou částí, podle režimu v kterém běží. Do uživatelského režimu (*user mode*) patří, z pohledu vybavení pro USB, aplikace a případně knihovny které oddělují aplikace od přímého přístupu k ovladačům. Do chráněného režimu jádra systému (*kernel mode*) patří ovladače zařízení, ovladače pro hostitelský řadič a ovladače pro řízení sběrnice USB.



Obrázek 5.7 - Vrstvy programového vybavení pro hostitelský počítač

Na obrázku 5.7 jsou znázorněny jednotlivé vrstvy a části systému starající se o funkci USB zařízení. Zvýrazněné části je třeba pro nové zařízení napsat znovu, pokud zařízení nepoužívá univerzální ovladače. Univerzální ovladače jsou použity, pokud patří zařízení do

některé ze systémem podporovaných tříd. U systému Windows 2000 jsou to tyto standardy tříd: HID (*Human Interface Devices*), zvuková zařízení (*USB audio*) a uložiště dat (*USB mass storage devices*).

Zařízení USB patří mezi zařízení použitelná okamžitě po připojení (*Plug and Play*, dále jen PnP), proto se o události týkající se připojení a odpojení stará PnP manažer jádra. Po připojení zařízení zahájí PnP manažer enumeraci zařízení a podle polí *idVendor* a *idProduct* v získaném deskriptoru zařízení vybere vhodný ovladač z již nainstalovaných nebo ve spolupráci z dalšími moduly systému vyzve uživatele k instalaci nového. Po výběru ovladače dojde k jeho spuštění, pokud již není aktivní, nakonec PnP zašle ovladači požadavek na zahájení funkce zařízení.

Při odpojení zašle PnP manažer ovladači požadavek na zastavení funkce zařízení a pokud ovladač není nadále využíván, také zastaví ho.

Aplikace běží v *user-mode* režimu operačního systému a pro přístup k ovladačům využívají služeb I/O manažera. Systém se tak chrání a snižuje riziko chyby a následného kolapsu, který by mohla způsobit chyba v ovladači běžícím v *kernel-mode* režimu. Pomocí modulu I/O manažera mohou aplikace přistupovat k ovladači zařízení jako k pojmenovanému souboru a také s ním tak mohou pracovat. Vrstva ovladače zařízení bude popsána samostatně v další podkapitole.

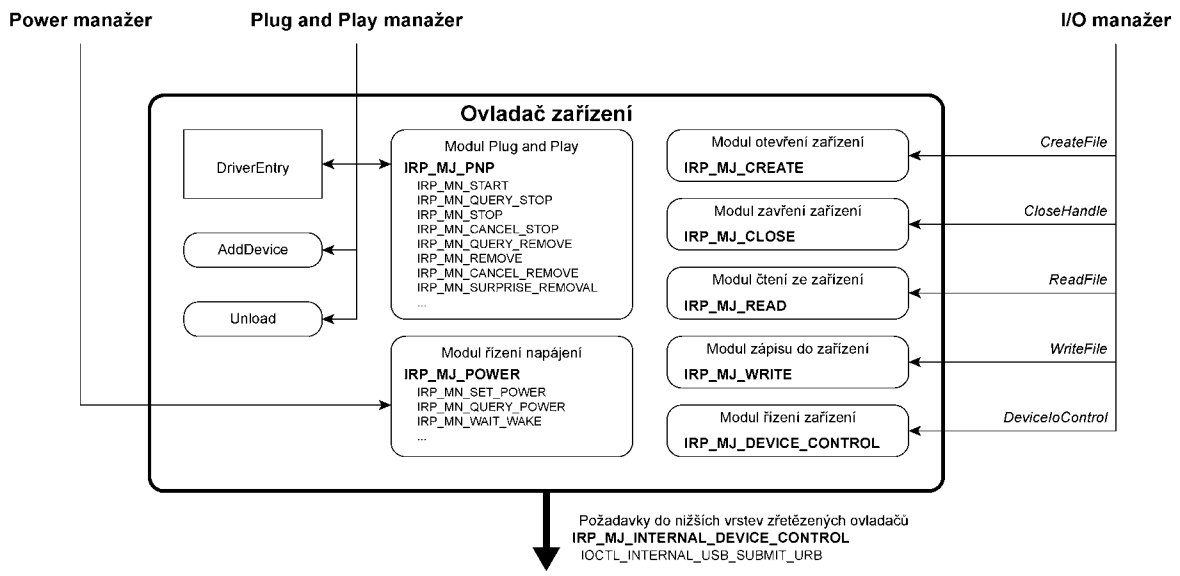
Aby jednotlivé vytvářené aplikace pro stejné zařízení nemusely používat elementární příkazy a potřebné konstanty (například jméno ovladače) pro komunikaci s ovladačem, je často mezi vrstvou aplikace a ovladače vložena vrstva knihovny, která může být dynamicky nebo staticky připojovaná k aplikaci. K překladu zdrojových kódů, využívajících funkce pro přístup k ovladačům, stejně jako pro vlastní ovladače, je potřeba speciální balík knihoven Windows *Driver Development Kit* (DDK). To je také dalším důvodem pro použití oddělující knihovny mezi aplikací a ovladačem, aplikace tak potřebují pouze soubory pro připojení knihovny.

Vlastní ovladač zařízení je zřetězen do fronty dalších ovladačů zajišťující jednotlivé funkce pro hardwarové prostředky. Na obrázku 5.7 jsou zobrazeny navíc jen dva ovladače: ovladač sběrnice a ovladač hostitele USB. Navíc jsou ve frontě i další ovladače, například ovladač převaděče PCI-USB, ovladač rozbočovače a mezi nimi i ovladače filtrů. Komunikace mezi ovladači probíhá pomocí IRP požadavků (*I/O Request Packet*). IRP je struktura s různými proměnnými položkami podle typu požadavku. Existuje 11 tříd požadavků a kromě několika základních, záleží na typu zařízení, které bude jeho ovladač zpracovávat. Pokud aplikace zavolá funkci WinAPI (*Windows Application Programming Interface*) pro práci se souborem se jménem zařízení (nejčastěji otevření, zavření, čtení a zápis dat a řídící požadavek), příkaz zavolá I/O manažera, který převede požadovaný příkaz na IRP požadavek ovladači, který obsluhuje zařízení s uvedeným jménem. Ovladač pak vytvoří a inicializuje vlastní IRP požadavek s URB strukturu (*USB request block*), který je poslán do nižších vrstev fronty ovladačů, až je nakonec transformován v ovladači hostitele na USB přenos po sběrnici. URB struktury jsou ovladačem hostitele podporované požadavky s definovaným formátem.

Skupina dokumentovaných USB požadavků je v systému Windows redukována na ty nejběžnější požadavky, některé další lze poslat zařízení pomocí nedokumentovaných funkcí ovladače hostitele.

5.1.3 Popis struktury ovladače pro USB zařízení.

Zjednodušeně je celá struktura ovladače zobrazena na obrázku 5.8. Protože je struktura jednotlivých bloků ovladače závislá z velké části na funkci zařízení, jsou zde popsány pouze obecné funkce bloků.



Obrázek 5.8 - Základní struktura ovladače pro zařízení USB.
(S označenými požadavky, které každý modul zpracovává)

Vstupním bodem každého ovladače je funkce DriverEntry, kterou PnP manažer spustí při inicializaci ovladače. V této funkci nastaví do datové struktury ovladače adresy funkcí pro zpracování jednotlivých typů IRP, jak jsou znázorněny na obrázku 5.8. Systém pak volá tyto funkce pokud přijde IRP požadavek pro ovladač. Navíc se nastaví adresy funkce pro přidání ovladače do řetězce ovladačů (*AddDevice*) a funkce pro uvolnění zdrojů při ukončení činnosti ovladače (*Unload*).

Když PnP manažer provede inicializaci ovladače zařízení, poté zavolá funkci *AddDevice*. Touto funkcí se přidá nové propojení ovladače na zařízení, s kterým je tak možno komunikovat skrz ovladač pomocí IRP.

Kromě základní inicializace ovladače a jeho ukončení, zajišťuje PnP manažer vysílání IRP požadavků ovladači podle stavu zařízení a případného zásahu uživatele. PnP modul ovladače tak musí umět reagovat na požadavky: spuštění funkce zařízení (*START_DEVICE*), dotaz na možnost zastavit funkci zařízení (*QUERY_STOP_DEVICE*), zastavení funkce zařízení (*STOP_DEVICE*), přerušení zastavování (*CANCEL_STOP_DEVICE*), dotaz na možnost odpojení zařízení (*QUERY_REMOVE_DEVICE*), softwarové odpojení zařízení (*REMOVE_DEVICE*), přerušení odpojování (*CANCEL_REMOVE_DEVICE*) a neočekávané odpojení (*SURPRISE_REMOVAL*). Zvláště neočekávané odpojení je pro USB důležité, protože může dojít k přerušení sběrnice například vytažením konektoru za zásuvky USB, v tom případě musí ovladač korektně přerušit probíhající komunikaci, aby nezapříčinil chybu operačního systému.

Modul ovladače pro řízení napájení vyřizuje požadavky od systémového Power manažeru, který se stará o správu výkonu počítače, přepíná režimy napájení systému. Při změně provozního režimu posílá Power manažer všem ovladačům IRP požadavky s upozorněním na přepnutí režimu, díky tomu mohou zařízení změnit podle možností svůj režim. Také posílá požadavky na vzbuzení systému nebo zařízení vnějším signálem (například modem na příchozí volání). Tento modul proto reaguje na události: čekání na vzbuzení (*WAIT_WAKE*), dotaz a nastavení režimu zařízení (*QUERY_POWER*) a (*SET_POWER*).

Zbylých pět modulů je důležitých pro komunikaci s aplikacemi a zařízením. Jednotlivé moduly vyřizují IRP požadavky od I/O manažera, který je posílá v reakci na WinAPI funkce pro práci se soubory.

Modul otevření zařízení zpracovává požadavek IRP_MJ_CREATE, který je vyslán při pokusu o otevření zařízení aplikací pomocí funkce *CreateFile*. Zpracování tohoto požadavku

obsahuje inicializaci potřebných datových struktur pro případné následující čtení nebo zápis. Na ovladači také záleží, zda povolí vícenásobné otevření a tím i sdílení zařízení. Je-li akce úspěšná, ovladač vrátí I/O manažeru informaci o úspěchu a manažer vrátí aplikaci handle otevřeného zařízení. Ostatní WinAPI funkce pak jako jeden ze svých parametrů posílají tento handle.

Modul zavření zařízení reaguje na požadavek transformovaný I/O manažerem z WinAPI funkce *CloseHandle*. Při pokusu o zavření zařízení je třeba přerušit všechny probíhající přenosy a uvolnit všechny alokované zdroje při otevření tak, aby nebyl ohrožen operační systém.

Modul řízení zpracovává požadavky IO řízení zařízení, ty jsou vysílány jako reakce na funkci *DeviceIoControl* v *user-mode* části systému. Jednotlivé požadavky jsou rozlišeny IOCTL kódy požadavků. Pomocí těchto požadavků lze řídit a nebo číst stav zařízení a ovladače. Pro zařízení jsou většinou transformovány na URB požadavek, který je následně v ovladači hostitelského řadiče transformován na USB požadavek specifický pro aplikaci (*vendor request*).

Moduly čtení dat ze zařízení a zápisu dat do zařízení obsluhují požadavky vyvolané WinAPI funkcemi *ReadFile* a *WriteFile*. Funkce mají jako parametry velikost požadovaných dat a jejich umístění, to je v IRP požadavku předáno ovladači. Pokud je požadovaná velikost přenosu větší než zařízení dovoluje, je požadavek rozdělen na několik menších, které jsou následně vyslány jako samostatné požadavky URB ovladači hostitele.

Protože příchod jednotlivých požadavků do ovladače je asynchronní, musí být zajištěna synchronizace přístupu ke společných strukturám, aby nedošlo k jejich změně pokud jsou využívány.

5.2 Použité softwarové nástroje

Při vývoji jednotlivých částí programového vybavení byla snaha o využití co nejvíce volně dostupných nástrojů.

5.2.1 Nástroje pro programové vybavení zařízení

5.2.1.1 Deska iD12 s CPU AT89S5282

Vývoj programového vybavení pro desku iD12 byl prováděn v jazyce C, proto byl potřeba překladač z jazyka C do strojového kódu. K tomu účelu byl použit *Small Device C Compiler* (SDCC). Jedná se o volně dostupný ANSI-C překladač určený pro 8bitové mikroprocesory založené na architektuře Intel 8051 a Zilog Z80. Autoři překladače pracují i na podpoře pro architektury Atmel AVR, Dallas DS390 a Microchip PIC. SDCC podporuje základní datové celočíselné typy *char* (1 byte), *short* (2 byte), *long* (4 byte) a typ s plovoucí řádovou čárkou *float* (4 byte ve formátu IEEE). SDCC také umožňuje přímé vkládání příkazů v jazyku symbolických adres - assembleru. SDCC je srovnatelný například s komerčním překladačem KeilC51. Podstatným rozdílem je metoda ukládání dat v paměti, zatímco překladač KeilC51 ukládá nejvíce významný byte jako první a nejméně významný byte jako poslední (big-endian), překladač SDCC ukládá jako první nejméně významný byte (little-endian). Navíc SDCC produkuje hůře optimalizovaný výsledný kód, který tak potřebuje více paměti.

Vývoj programů pro desku iD12 byl prováděn pomocí SDCC verze 2.3.0. Poslední verzi kompilátoru a další informace lze získat na adrese URL:<http://sdcc.sourceforge.net>.

Překlad jednotlivých programů byl proveden automaticky pomocí utility *make*, která není součástí SDCC.

5.2.1.2 Deska MO_CPU s CPU M68376

Pro desku MO_CPU byl dopsán pouze modul USB k již hotovým částem programů, které byly dodány spolu s deskou. Proto byly při vývoji použity stejné nástroje jako pro zbývající části. Mezi tyto nástroje patří jednak volně dostupné GNU nástroje zkompilované pro cílovou platformu *m68k-coff* a spouštěné v systému Linux, kde byl vývoj programového vybavení pro CPU M68376 prováděn. Pro ladění aplikace byl využit režim BDM (*Background Debug Mode*) mikrokontroléru.

5.2.2 Nástroje pro programové vybavení hostitele

5.2.2.1 Linux

Pro vývoj aplikací v operačním systému Linux byla využita volně dostupná knihovna LIBUSB. Domovská stránka projektu LIBUSB je na URL:<http://libusb.sourceforge.net>, kde lze získat poslední verzi knihovny. Tato knihovna zajišťuje přístup k zařízením USB v uživatelském módu bez vlastních ovladačů.

Jednotlivé aplikace byly přeloženy překladačem GCC, který je spolu s dalšími nástroji součástí volně dostupného balíku nástrojů GNU Tool Chain. Tento balík je také součástí systému Linux.

5.2.2.2 Windows 2000

K vývoji programového vybavení pro operační systém Windows 2000 byl použit program *Microsoft VisualC++* 6.0 (dále jen MSVC), který je součástí vývojových prostředků sdružených do Microsoft Visual Studio 6.0. Pomocí MSVC byly napsány všechny programy a knihovny jako samostatné projekty s výjimkou ovladačů pro stranu hostitele USB. Při jejich vývoji byla využívána nápověda MSDN [10].

Tento prostředek není volně dostupný, ale bylo nutné ho použít minimálně pro vývoj ovladačů, protože byl vyžadován balíkem *Microsoft Windows 2000 Driver Development Kit* (dále jen DDK), bez kterého nelze ovladače pro systém Windows vyvíjet.

DDK obsahuje hlavičkové soubory a statické knihovny s funkcemi z jádra systému Windows, které jsou nezbytné pro překlad vytvořených ovladačů. V DDK je dále mnoho příkladů ovladačů pro různé druhy zařízení a také obsáhlá nápověda [8] nejen k samotnému psaní ovladačů, ale i k jejich distribuci a instalaci. Při vlastním vývoji ovladačů bylo vycházeno z příkladů DDK pro zařízení USB (v adresáři NTDDK\src\wdm\usb).

5.2.2.3 MathWorks MATLAB

Tento program, pro který byla vytvořena knihovna pro práci s USB zařízením je v samostatné podkapitole, protože je dostupný jak pro systém Windows, tak pro systém Linux. Vývoj vlastních MEX-funkcí je možný buď v jazyce C nebo v jazyce Fortran, pro jejich překlad jsou všechny nutné nástroje součástí instalace programu Matlab s výjimkou překladače, který je závislý na operačním systému. Vytvořené funkce jsou přeloženy jako dynamicky připojované knihovny, které interpret Matlabu automaticky načte a spustí, pokud je vytvořená funkce v programu volána. Potřebné informace lze nalézt v [11].

5.3 Vytvořený software

V této podkapitole jsou stručně popsány jednotlivé programy vytvořené v rámci práce. Rozsáhlejší popis je na přiloženém CD u jednotlivých programů v souboru *readme.txt* nebo přímo jako komentáře v jednotlivých programech.

5.3.1 SPI (PC-DOS)

Program slouží k naprogramování FLASH paměti kódu a EEPROM paměti dat CPU přes rozhraní SPI. Program je určený pouze pro obvod AT89S8252 použitý na desce iD12 a při použití s jinými obvody je nutné změnit konstanty velikosti pamětí uvnitř programu a případně i způsob programování obvodu, který probíhá podle doporučeného postupu uvedeného v [7].

Tento program byl napsán v jazyce C pro operační systém DOS. Protože používá k přístupu do programátoru (uvedeného v kapitole 4.2) přímý přístup na paralelní port, nelze jej využít v nových 32bitových systémech bez použití speciálních ovladačů.

Program může pracovat ve dvou režimech, buď přímé programování souboru zadaného na příkazové řádce nebo v interaktivním módu. V tom lze kromě programování také číst data z paměti obvodu a ukládat je do souboru, mazat paměť a provádět jednoduchou editaci dat mezi pamětí kódu, pamětí dat a načteným souborem pro paměť kódu a pro paměť dat. Program umožňuje zobrazit data jako adresovaný hexadecimální a ASCII výpis nebo jako výpis v jazyce symbolických adres. Jsou podporovány soubory ve formátu Intel HEX (přípona HEX nebo IHX) a soubory se syrovými daty bez adresace (jakákoliv jiná přípona).

Ovládání programu je intuitivní a popsané nabídkou funkčních kláves zobrazených v programu.

5.3.2 PDIUSB (iD12, M68)

Modul PDIUSB je řešením nejspodnější vrstvy programového vybavení zařízení. Je napsán v jazyce C pro překladač SDCC, ale bez úprav je použitelný i pro další překladače, například KeilC51, m68-coff-gcc. Jedná se o univerzální modul pro celou rodinu PDIUSB obvodů rozhraní USB firmy Philips (obvody PDIUSBD11, PDIUSBD12, PDIUSBH11 a PDIUSBH11A v obou režimech).

Při použití je třeba vybrat několik nastavení pro překladač. Všechna nastavení se provádí v souboru PDIUSB.H „zakomentováním“ nebo „odkomentováním“ definic jednotlivých nastavení. Podrobně je vše popsáno u jednotlivých nastavení v souboru PDIUSB.H. Tímto způsobem je třeba vybrat použitý obvod. Dále povolit nebo zakázat klíčové slovo *using* pro změnu banky registru u funkcí pro proceduru přerušení. Modul umožňuje použít vlastní procedury pro komunikaci s obvodem nebo výběr adres pro komunikaci pomocí procedur, které jsou součástí modulu.

Jednotlivé funkce v modulu odpovídají funkcím obvodů dokumentovaných v katalogových listech (pro PDIUSBD12 v [6]), zároveň jsou v modulu definice všech potřebných konstant pro nastavení obvodů.

Všechny programy psané pro iD12 i MO_CPU využívají tento modul pro komunikaci s obvodem PDIUSBD12.

5.3.3 SKELETON (iD12)

Program představuje kostru aplikace pro desku iD12 s rozhraním USB. Program je napsán v jazyce C pro překladač SDCC a skládá se z několika samostatných modulů tak, aby byl umožněn co nejjednodušší vývoj dalších aplikací, vycházejících z této kostry. Protože se jedná o vývojovou desku, je program určen pro překlad a nahrání do externí paměti desky (více v další podkapitole o programu BOOTLOAD).

Program využívá přerušení od obvodu PDIUSBD12, ale na rozdíl od obrázku 5.4 zpracovává události z řídící brány mimo přerušení. Díky tomu lze zapnou výpis ladících informací, které je možné nechat vypisovat za běhu programu na sériové rozhraní. To tvoří další modul kostry. Skrze sériovou linku je možné komunikovat s deskou iD12 pomocí

několika základních příkazů, jejich úplný přehled program vypíše po příkazu „help“. Důležité příkazy programu jsou popsány i s parametry v následujícím přehledu.

Všechny příkazy jsou psány malými písmeny a parametry jsou psány podobně jako v jazyce C a odděleny mezi sebou čárkami nebo mezerami. Program zná následující výrazy: desítkové číslo (např.123), hexadecimální číslo (např.0x12AB), znak (např.‘i’) a řetězec (např.“zyxvw“).

Základní příkazy:

- *help* – Vypíše přehled známých příkazů.
- *ver* – Vypíše verzi jednotlivých modulů programu.
- *echo* – Bez parametrů vypíše stav funkce echo. S parametrem *1* zapne funkci echa, která vrací všechny přijaté znaky sériovou linkou. S parametrem *0* funkci vypne.
- *debug* – Bez parametrů vypíše stav proměnné debug. S parametrem *1* zapne funkci a s parametrem *0* vypne funkci výpisu ladících informací.
- *swreset* – Provede softwarový reset CPU, pokud je zadán parametr *i* je proveden reset programu z interní FLASH paměti mikrokontroléru.
- *usbon* – Připojení rozhraní USB, vyslání konfigurace obvodu PDIUSBD12 pro připojení R_{PU} ke sběrnici.
- *usboff* – Odpojení rozhraní USB od sběrnice.
- *get* – Funkce přečte hodnotu bajtu na adrese zadané druhým parametrem z paměti určené prvním parametrem (E –XDATA, C – CODE, R –RAM).
- *set* – Funkce nastaví hodnotu zadanou třetím parametrem na paměťové místo určené prvními dvěma bajty jako u příkazu *get*.
- *dump* – Funkce vypíše část paměti o délce určené třetím parametrem od místa zadaného prvními dvěma parametry jako u příkazu *get*.

Při použití pro většinu programů stačí dopsat zpracování událostí pro USB brány 1 a 2 v programové smyčce, definice deskriptorů a případně obsluhu požadavků specifických pro zařízení a třídu.

5.3.4 BOOTLOAD (iD12)

Program je vytvořen ze základní kostry programu, popsané výše, s několika speciálními úpravami, protože se jedná o zaváděcí program určený pro nahrání do interní FLASH paměti CPU. Vzhledem k velikosti interní paměti a použití SDCC k překladu programu, je vypnutý modul příkazové řádky a výpis ladících informací.

Program umožňuje komunikovat s deskou po sběrnici USB pomocí speciálního ovladače a programu pro nahrávání aplikací do iD12, popsaných dále.

Nahrávání a spouštění aplikací z externí paměti je umožněno zavedením vektorů přerušení na začátku externí paměti, tam jsou přesměrovány všechny obsluhy přerušení z interní paměti. Po zapnutí desky se spustí program v interní paměti, který nejdříve zkontroluje poslední dva bajty v externí paměti a pokud obsahují definovanou hodnotu, pokračuje program skokem na začátek externí paměti, kde je nahrán spouštěný program. Pokud je na posledních dvou bajtech jiná kombinace, CPU pokračuje ve vykonávání interního programu, kterým je BOOTLOAD. Tato metoda není stoprocentně bezpečná, protože stav externí paměti po zapnutí je náhodný. Vzhledem k účelu desky je ale dostačující (Pravděpodobnost skoku do externí paměti bez nahraného programu je 1/65536.).

Program zpracovává řídící požadavky specifické pro zařízení pro: spuštění nahrávání externího programu pomocí brány pro přenos objemných dat, softwarový reset desky s možností vypnutí USB a nastavení příznaku spuštění externího programu, nastavení jednoho

bajtu paměti, spuštění čtení paměti přes bránu pro přenos objemných dat a čtený jednoho bajtu paměti.

5.3.5 USB2COM (iD12)

Ukázková aplikace je vytvořená z kostry programů pro iD12. Program funguje jako převodník mezi rozhraním USB a rozhraním RS232 (RS485).

Na přiloženém CD je program přeložen tak, aby se po zapnutí připojila sběrnice USB a spustila se funkce převodníku. Při překladu lze změnit nastavení programu, aby byl po zapnutí přístupný příkazový řádek. Z něj je pak možné zapnout funkci převodníku příkazem „usb2com“, jinak jsou dostupné stejné funkce jako u základní kostry programů, tedy i výpis ladících informací, který je směrován na sériovou linku a neovlivňuje nijak normální funkci převodníku.

Program je rozšířen vzhledem k základní kostře o modul *u2c*, který ukládá příchozí data ze sběrnice USB do jedné vyrovnávací paměti a ze sériové linky do druhé vyrovnávací paměti, z nich pak data vysílá na opačné sběrnice. Program tak pracuje v duplexním režimu přenosu dat.

Pro správnou funkci desky jako převodníku je potřeba na straně hostitele USB ovladač vytvořený také jako součást diplomové práce a popsaný dále v kapitole.

5.3.6 USB (M68)

Jedná se o programový modul obsluhy sběrnice USB pro desku MO_CPU s využitím vytvořené knihovny PDIUSB. Modul obsahuje základní funkce pro rozhraní USB stejně jako aplikace kostry programů pro iD12 popsaná výše. Navíc obsahuje zpracování událostí brány pro přenos objemných dat tak, aby byl umožněn přístup k příkazové řádce přes USB shodný s již vytvořenou příkazovou řádkou pro sériový kanál.

Modul je s drobnými modifikacemi použit v ukázkových programech, dodaných spolu s deskou MO_CPU.

5.3.7 SEND2D12 (PC-WIN, LINUX)

Program je „protikusem“ pro hostitelský počítač k aplikaci BOOTLOAD pro desku iD12 a slouží k nahrávání a spouštění programů z externí paměti desky iD12. Program je konzolovou aplikací, která využívá knihovny LIBUSB pro Linux nebo vytvořené knihovny a ovladače WLIBUSB pro Windows.

Aplikace umožňuje načtení souborů ve formátu Intel HEX a nahrání jejich obsahu do paměti desky iD12. Pokud je zadáno více souborů, jsou odesílány ve stejném pořadí, v jakém byly zadány. Pro každý soubor lze specifikovat, jak velká část souboru a do jakého typu paměti má být nahrán. Program obsahuje dále funkci softwarového resetu desky s výběrem, odkud bude program po resetu spuštěn, funkci vynulování externí paměti a nastavení adresy externího programu.

Argumenty programu a syntaxe příkazové řádky jsou následující:

```
send2d12 [params] [filename1] [fileparams1] [filename2] [fileparams2] …

params              -- společné parametry programu
    -h              - vypíše nápovědu
    -z, -zT, -zF:T  - vymaže externí paměť (celou, do adresy T, od adresy F do adresy T)
    -v              - „upovídaný“ režim programu
    -u, -uX         - výběr používané brány USB
    -r, -ri, -re    - softwarový reset desky (-re nastaví spouštění externího programu)
filename            - jméno souboru ve formátu Intel HEX
fileparams          -- parametry pro jednotlivé soubory, vztažené vždy k poslednímu souboru
    -tX             - určení cílové paměti, pro data ze souboru (R pro RAM, E pro XDATA)
```

-sT, -sF:T	- poslání pouze části dat (od začátku do adresy T, mezi adresami F a T)
-i	- neodeslání části souboru, kde jsou obsluhy přerušení (0x8000-0x8032)
-g, -gA	- nastavení startovací adresy na začátek souboru nebo na adresu A

5.3.8 UCOM (PC-LINUX)

Program je jednoduchou USB verzí textového terminálu pro komunikaci pomocí sériové linky. Byl vytvořen pro otestování funkce USB modulu pro MO_CPU. Ke své funkci využívá knihovnu LIBUSB a měl by tak být s drobnými úpravami použitelný i pro systém Windows s využitím knihovny WLIBUSB, s kterou ale nebyl testován. Program umožňuje výběr VID a PID zařízení, se kterým bude komunikovat a možnost vyslání speciálního požadavku pro zapnutí a vypnutí funkce terminálu v zařízení.

5.3.9 BOOTLOAD (PC-WIN)

Jedná se o ovladač pro systém Windows 2000, založený na příkladu *bulkusb* z balíku W2000 DDK a pouze ho rozšiřuje o zpracování dalších IOCTL požadavků. Ovladač komunikuje se zařízením pomocí brány pro přenos objemných dat a řídící brány, která je využita pro zasílání řídících požadavků specifických pro zařízení.

K tomuto ovladači je vytvořena knihovna rozhraní mezi ovladačem a aplikacemi, které tak mohou k funkcím ovladače přistupovat voláním jednotlivých funkcí knihovny bez znalosti jména zařízení.

Tento ovladač byl vytvořen pro komunikaci se starší vytvořenou verzí programu BOOTLOAD a využívala ji starší verze aplikace send2d12.

5.3.10 WLIBUSB (PC-WIN)

Součástí tohoto balíku je jednak ovladač založený také na příkladu *bulkusb* z balíku DDK, který je ovšem doplněn o možnost vysílat všechny požadavky, které operační systém Windows umožňuje. Navíc jsou všechny přenosy vysílány s nastavitelným časovým omezením (*timeout*).

Další částí balíku WLIBUSB je knihovna zajišťující stejné chování ovladače jako má knihovna LIBUSB pro Linux. Výjimkou jsou funkce, které systém Windows neumožňuje. Popis původních funkcí z knihovny LIBUSB lze nalézt v dokumentaci ke knihovně, která je jako jeden z nástrojů na přiloženém CD. Popis funkcí knihovny WLIBUSB a rozdílů mezi těmito knihovnami je v souboru *readme.txt* balíku WLIBUSB.

5.3.11 USB2COM (PC-WIN)

Jedná se o ovladač pro systém Windows 2000, který umožňuje aplikacím přistupovat k zařízení USB jako k vysokorychlostnímu sériovému portu. Ovladač vychází ze dvou příkladů *bulkusb* a *serial* nalézající se v balíku DDK.

Na rozdíl od výše uvedených ovladačů je tento naprosto odlišný od původního ovladače založeného na příkladu *bulkusb*. To je způsobeno potřebou vyrovnávací paměti pro přenos mezi současně příchozími požadavky z aplikací a sběrnice USB. Zároveň ovladač využívá pro svou funkci tří samostatných vláken. Jedno pro čtení dat z USB sběrnice, další pro zápis dat na sběrnici a třetí pro čekání na událost nastavené z aplikace. Díky tomu umožňuje ovladač plně duplexní asynchronní komunikaci se zařízením.

Ovladač reaguje na všechny řídicí požadavky pro sériový port, popsané v dokumentaci k DDK (požadavky začínající prefixem IOCTL_SERIAL). Ovladač tak mohou využívat základní funkce WinAPI pro práci se sériovým rozhraním bez dalších knihoven. Tím mohou ovladač využívat i aplikace postavené na těchto funkcích. Aplikace využívající pro práci

přerušení nebo přímý přístup k I/O portům běžného sériového rozhraní s tímto ovladačem nefungují.

Ovladač vytvoří při startu zařízení odkaz se jménem „COM“ a číslem prvního volného portu. Toto jméno si ovladač zapamatuje v registru, čímž je zařízení přístupné pod stejným jménem i při dalším připojení. Aplikace pak mohou otevřít zařízení s tímto jménem. Jméno portu, stejně jako i další parametry, lze změnit na listu předvoleb pro zařízení využívající ovladač USB2COM. Tento list je přidáván pomocí knihovny EVBOUSB popsané dále.

Ovladač a jeho funkce byly zkoušeny programem MTTTY, který je přístupný jako příklad na stránkách firmy Microsoft.

Jednotlivé funkce pro komunikaci jsou popsány například v [10], zde je uveden pouze seznam funkcí, které vytvářejí ovladačem podporované požadavky:

CreateFile, *CloseFile*, *ReadFile*, *WriteFile*, *ClearCommBreak*, *ClearCommError*, *EscapeCommFunction*, *GetCommConfig*, *GetCommMask*, *GetCommModemStatus*, *GetCommProperties*, *GetCommState*, *GetCommTimeouts*, *PurgeComm*, *SetCommBreak*, *SetCommConfig*, *SetCommMask*, *SetCommState*, *SetCommTimeouts*, *SetupComm*, *TransmitCommChar*, *WaitCommEvent*.

Ovladač je pro přehlednost rozdělen do jednotlivých souborů podle zpracovávaných požadavků, jak je znázorněno na obrázku 5.8, kde ovšem nejsou zobrazeny všechny obsluhované typy požadavků.

5.3.12 EVBOUSB (PC-WIN)

K oddělení od ostatních zařízení je vytvořena vlastní třída zařízení, do které jsou zařazena všechna zařízení vytvořená v rámci DP. Třída je pojmenovaná „Vývojové desky USB“. Knihovna přidává nové listy s předvolbami (*property-pages*) pro všechna vytvořená zařízení, která je při instalaci vyžadují.

Tato knihovna vznikla z příkladu *pnpports* balíku DDK a je využita především pro nastavení zařízení používající ovladač USB2COM. Pro všechny ostatní obsahuje pouze list s informacemi o zařízení a ovladači.

5.3.13 USBPORT (PC-WIN,LINUX)

Tato knihovna obsahuje základní funkce pro práci s USB zařízeními v programu Matlab. Jedná se o MEX funkci, kterou lze přeložit podle popisu například v [11]. Dále jsou součástí knihovny skripty pro Matlab, které přistupují k funkcím společné MEX funkce. MEX funkce využívá pro přístup k USB knihovny LIBUSB/WLIBUSB a pro program Matlab zajišťuje základní funkce pro otevření a zavření zařízení, čtení a zápis dat do USB brány s přenosem objemných dat, vyslání řídicího požadavku a zvolení konfigurace, které jsou obdobou funkcí knihovny LIBUSB. Podrobněji jsou funkce popsány v souboru *readme.txt* na přiloženém CD u této knihovny.

6. Závěr

Hlavní náplní práce byla implementace rozhraní USB pro mikroprocesorové systémy a vytvoření programové podpory pro toto rozhraní.

Na úvod této diplomové práce byl proveden stručný výtah ze specifikace USB verze 1.1 a částečně i verze 2.0, který pomohl pochopit funkci jednotlivých částí a vrstev sběrnice. Výtah je proveden do větší hloubky než bylo nezbytně nutné pro řešení zadání tak, aby byl získán širší pohled na tuto relativně novou sběrnici. Komplexnější výtah ze specifikace byl vypracován také proto, že se jedná o perspektivní rozhraní, které stále více vytlačuje starší sériové rozhraní RS232 a to nejen pro počítačové periferie, ale i pro některé průmyslové systémy jako jsou systémy pro konfiguraci a sběr dat.

Protože se práce zabývá implementací rozhraní USB s využitím dostupného vyráběného obvodu, musel být takový obvod nejprve vybrán z rozsáhlé nabídky dostupných obvodů. K tomu účelu byly, nad rámec zadání této práce, vypracovány dva katalogy zpřehledňující částečně nabídku obvodů pro implementaci USB. Jedním z nich je přehledový katalog všech prozkoumaných obvodů pro USB a druhým je podrobnější katalog řadičů a mikrokontrolérů s implementovaným rozhraním USB. Z dostupných obvodů byl vybrán obvod řadiče rozhraní USB od firmy Philips PDIUSBD12.

S vybraným obvodem byla navržena a realizována deska s připojeným rozhraním USB k mikrokontroléru řady x51 AT89S8252 firmy Atmel. Deska obsahuje kromě rozhraní USB i rozhraní RS232 a RS485 a rozhraní pro připojení dalších mikrokontrolérů, uzpůsobené především pro připojení desky MO_CPU firmy PiKRON s mikrokontrolérem MC68376. Díky tomu umožňuje deska jednoduchý vývoj a testování aplikací využívajících rozhraní USB.

Největší podíl praktické části diplomové práce zaujímá vytvoření programového vybavení pro realizované rozhraní. Programovou výbavu lze rozdělit na dvě části, část pro stranu zařízení a část pro hostitelský systém. Pro zařízení byl vytvořen univerzální modul pro podporu vybraného obvodu PDIUSBD12 stejně jako i dalších obvodů ze skupiny obvodů PDIUSB firmy Philips. Tento modul byl napsán s ohledem na co nejjednodušší přenositelnost mezi různými architekturami řídicího mikrokontroléru. Dále bylo pro zařízení napsáno několik aplikací, které demonstrují použití rozhraní USB.

Jako hostitelský systém byl použit osobní počítač s operačním systémem Windows 2000 firmy Microsoft a také operační systém Linux. Pro systém Windows bylo vytvořeno několik ovladačů, které umožňují systému komunikovat se zařízením přes USB. Mezi nejrozsáhlejší patří ovladač USB2COM, díky jemu se zařízení chová pro 32 bitové aplikace jako vysokorychlostní sériový port. Pro systém Linux byla využita volně dostupná knihovna LIBUSB umožňující aplikacím přístup k USB zařízením bez speciálních ovladačů nebo podpory v operačním systému. Byla také vytvořena obdoba této knihovny pro operační systém Windows WLIBUSB, která má stejné funkce a parametry jako LIBUSB, až na odlišnou instalaci pro zařízení danou vlastnostmi systému Windows. Navíc byla vytvořena knihovna pro podporu zařízení USB v programu MATLAB firmy MathWorks pomocí LIBUSB/WLIBUSB. Pomocí této knihovny lze vytvářet programy pro komunikaci se zařízením pomocí sběrnice USB přímo v programu MATLAB.

7. Literatura

[1] *Universal Serial Bus Specification Revision 1.1*.
URL:<http://www.usb.org> [cit.2001-03-08]

[2] *Universal Serial Bus Specification Revision 2.0*.
URL:<http://www.usb.org> [cit.2002-02-06]

[3] *On-The-Go Supplement to the USB 2.0 Specification Revision 1.0*
URL:<http://www.usb.org> [cit.2003-03-20]

[4] *The Unicode Standard. Worldwide Character Encoding. Version 3.0.*
The Unicode Consortium, Addison-Wesley Publishing Company,
Reading, Massachusetts. URL:<http://www.unicode.com>

[5] Petr Kocourek. *Přenos Informace*. Vydavatelství ČVUT, Praha, 1999.

[6] *PDIUSBD12 Product data*. Koninklijke Philips Electronics, 2001.
URL:<http://www.semiconductors.philips.com>

[7] *AT89S8252 Product data*. Atmel Corporation, 2000.
URL:<http://www.atmel.com>

[8] *Microsoft Windows 2000 Driver Development Kit*.
Microsoft Corporation. [Verze z 2000-06-28]

[9] Johan Knol. *SDCC Compiler User Guide*. [2001-07-16]

[10] *Microsoft Developer Network*. URL:<http://msdn.microsoft.com>

[11] *MATLAB Application Program Interface Guide, version 5*.
The MathWorks, Inc., 1998.

8. Obsah přiloženého CD

Každý adresář na CD je opatřen souborem *index.htm* s odkazy a popisem podadresářů a souborů nacházejících se v adresáři.

Obsah kořenového adresáře přiloženého CD:

/bin	Zde jsou přeložené jednotlivé části vytvořeného programového vybavení.
/catalog	Elektronická podoba vytvořených katalogů součástek pro implementaci USB, součástí jsou jednotlivé katalogové listy uvedených obvodů.
/doc	Adresář obsahuje dokumenty použité při implementaci USB rozhraní a při vývoji programového vybavení, dále je zde elektronická podoba textu diplomové práce.
/src	Zdrojové soubory vytvořeného programového vybavení rozdělené podle platforem pro které jsou určeny.
/tools	Volně dostupné nástroje a balíky použité při vývoji.
index.htm	Soubor s popisem a odkazy na CD.

9. Přílohy

A. Katalog obvodů s USB

A.1 Přehledový katalog obvodů pro implementaci USB

Obvody jsou rozděleny do jednotlivých tabulek podle funkce. Všechny tabulky jsou abecedně seřazeny podle jména výrobce.

A.1.1 Spínače napětí

Tabulka A.1 - Přehled obvodů USB – Spínače napětí.

Device	Manufacturer	Description
A2525	Allegro	USB POWER CONTROL SWITCHES
A2526	Allegro	USB DUAL POWER CONTROL SWITCHES
A2535	Allegro	USB POWER CONTROL SWITCHES
A2536	Allegro	USB DUAL POWER CONTROL SWITCHES
ADM1072	Analog Devices	Dual, USB 2.0 Full/Standby Power Controller with Supply Steering
MIC2025	Micrel	Single-Channel Power Distribution Switch
MIC2026	Micrel	Dual-Channel Power Distribution Switch
MIC2027	Micrel	Quad USB Power Distribution Switch
MIC2075	Micrel	Single-Channel Power Distribution Switch
MIC2076	Micrel	Dual-Channel Power Distribution Switch
MIC2077	Micrel	Quad USB Power Distribution Switch
MIC2505	Micrel	Single 2A High-Side Switches
MIC2506	Micrel	Dual 1A High-Side Switches
MIC2514	Micrel	IttyBitty™ Integrated High-Side Switch
MIC2524	Micrel	Quad USB Power Control Switch
MIC2525	Micrel	USB Power Control Switch
MIC2526	Micrel	Dual USB Power Control Switch
MIC2527	Micrel	Quad USB Power Control Switch
MIC2536	Micrel	Dual USB Power Distribution Switch
MIC2537	Micrel	Quad Power Distribution Switch
MIC2544	Micrel	Programmable Current Limit High-Side Switch
MIC2545A	Micrel	Programmable Current Limit High-Side Switch
MIC2546	Micrel	Dual Programmable Current Limit Switch
MIC2547	Micrel	Dual Programmable Current Limit Switch
MIC2548	Micrel	Programmable Current Limit High-Side Switch
MIC2549A	Micrel	Programmable Current Limit High-Side Switch
LM3525	National Semiconductor	Single Port USB Power Distribution Switch and Over-Current Protection
LM3526	National Semiconductor	Dual Port USB Power Distribution Switch and Over-Current Protection
LM3543	National Semiconductor	Triple Port USB Power Distribution Switch and Over-Current Protection
LM3544	National Semiconductor	Quad Port USB Power Distribution Switch and Over-Current Protection
µPD16874	NEC	QUAD Pch HIGH-SIDE SWITCH FOR USB
µPD16875	NEC	DUAL Pch HIGH-SIDE SWITCH FOR USB
µPD16876	NEC	SINGLE Pch HIGH-SIDE SWITCH FOR USB
SP2026	Sipex	+2.7V to +5.5V USB Power Control Switch
SP2525	Sipex	+3.0V to +5.5V USB Power Control Switch
SP2526	Sipex	+3.0V to +5.5V USB Power Control Switch
TPS2070	Texas Instruments	4-port USB HUB power controllers
TPS2071	Texas Instruments	4-port USB HUB power controllers
TPS2074	Texas Instruments	4-port USB HUB power controllers
TPS2075	Texas Instruments	4-port USB HUB power controllers
TPS2110	Texas Instruments	Autoswitching power MUX
TPS2111	Texas Instruments	Autoswitching power MUX
TPS2112	Texas Instruments	Autoswitching power MUX
TPS2113	Texas Instruments	Autoswitching power MUX
TPS2114	Texas Instruments	Autoswitching power MUX
TPS2115	Texas Instruments	Autoswitching power MUX
TPS2140	Texas Instruments	Adjustable LDO and SWITCH with Dual current limit for USB High-power Peripheral
TPS2141	Texas Instruments	Adjustable LDO and SWITCH with Dual current limit for USB High-power Peripheral
TPS2148	Texas Instruments	3.3V LDO and Dual switch for USB Peripheral
TPS2149	Texas Instruments	3.3V LDO and Dual switch for USB Bus-Powered HUB
TPS2150	Texas Instruments	Adjustable LDO and SWITCH with Dual current limit for USB High-power Peripheral
TPS2151	Texas Instruments	Adjustable LDO and SWITCH with Dual current limit for USB High-power Peripheral
TPS2158	Texas Instruments	3.3V LDO and Dual switch for USB Peripheral
TPS2159	Texas Instruments	3.3V LDO and Dual switch for USB Bus-Powered HUB

A.1.2 Blok vysílače a přijímačů

Tabulka A.2 - Přehled obvodů USB – Blok vysílače a přijímačů.

Device	Manufacturer	Description	USB spec.	USB speed
CY7C68000	Cypress	TX2™ USB 2.0 UTMI Transceiver	2.0	Full,High
USB1T11A	Fairchild Semiconductor	Universal Serial Bus Transceiver	1.1	Low,Full
MAX3340E	MAXIM	ESD-Protected USB Translators in UCSP	1.1	Low,Full
MAX3341E	MAXIM	ESD-Protected USB Transceivers with USB Detect	2.0	Full
MAX3342E	MAXIM	ESD-Protected USB Transceivers with USB Detect	2.0	Full
MAX3343E	MAXIM	ESD-Protected USB Level Translators in UCSP	1.1	Low,Full
MIC2550	Micrel	Universal Serial Bus Transceiver	2.0	Low,Full
MIC2551	Micrel	Universal Serial Bus Transceiver	2.0	Low,Full
ISP1102	Philips	Advanced Universal Serial Bus transceiver	2.0	Full
ISP1104	Philips	Advanced Universal Serial Bus transceiver	1.1	Low,Full
ISP1105	Philips	Advanced Universal Serial Bus transceivers	1.1	Low,Full
ISP1106	Philips	Advanced Universal Serial Bus transceivers	1.1	Low,Full
ISP1107	Philips	Advanced Universal Serial Bus transceivers	1.1	Low,Full
PDIUSBP11	Philips	Universal Serial Bus Transceiver	1.1	Low,Full
SP5301	Sipex	USB Transceiver	1.1	Low,Full

A.1.3 Blok fyzické vrstvy (PHY)

Tabulka A.3 - Přehled obvodů USB – Blok fyzické vrstvy.

Device	Manufacturer	Description	USB spec.	USB speed
USS2X1	Lucent Technologies	8bit USB 2.0 PHY	2.0	Full,High
USS2X1W	Lucent Technologies	16bit USB 2.0 PHY	2.0	Full,High
ISP1501	Philips	Hi-Speed USB peripheral transceiver and SIE	2.0	Full,High
GT3200	SMSC	USB 2.0 PHY IC	2.0	Full,High

A.1.4 Řadič hostitele

Tabulka A.4 - Přehled obvodů USB – Řadič hostitele bez společného zařízení.

Device	Manufact.	Description	USB spec.	USB speed
M5271	Ali	PCI <-> P1394 / USB2.0 / MS / SD Controller (EHCI 2xOHCI)	2.0	Low,Full,High
M5273	Ali	PCI <-> USB2.0 / MS Controller (EHCI 2xOHCI)	2.0	Low,Full,High
µPD720100A	NEC	USB 2.0 HOST Controller (EHCI+2xOHCI)	2.0	Low,Full,High
µPD720101	NEC	USB2.0 HOST CONTROLLER (EHCI+2xOHCI)	2.0	Low,Full,High
82C861	OPTi	FireLink USB – PCI-to-USB Bridge (OHCI)	1.1	Low,Full
82C862	OPTi	FireLink USB - Dual Controller PCI-USB Host Bridge (2xOHCI)	1.1	Low,Full
82C863	OPTi	FireLink USB - Dual Controller PCI-USB Host Bridge (2xOHCI)	1.1	Low,Full
ISP1160	Philips	Embedded Universal Serial Bus Host Controller	2.0	Low,Full
ISP1561	Philips	Hi-Speed USB PCI host controller (EHCI+2xOHCI)	2.0	Low,Full,High

Tabulka A.5 - Přehled obvodů USB – Řadič hostitele se společným zařízením.

Device	Manufact.	Description	USB spec.	USB speed
SL811HS	Cypress	Embedded USB Host/Slave Controller	1.1	Low,Full
S1R72005	Epson	On-The-Go device controller	2.0	Full
ISP1161	Philips	Full-speed USB single-chip host and device controller (no OTG)	2.0	Low,Full
ISP1362	Philips	Single-chip Universal Serial Bus On-The-Go controller	2.0	Low,Full
LC77700B-IAPr	SANYO	IAPr™ (INTERNET PROCESSOR) (2-port host/1port device)		

A.1.5 Řadič rozbočovače

Tabulka A.6 - Přehled obvodů USB – Řadič rozbočovače bez složeného zařízení.

Device	Manufacturer	Description	USB spec.	USB speed
AT43301	Atmel	Low-cost USB Hub Controller	1.1	Full
AT43312A	Atmel	Self- and Bus-powered USB Hub Controller	1.1	Full
CY7C65100	Cypress	Four-Port USB Fixed-Function Hub Controller	1.1	Full
CY7C65640	Cypress	TetraHub™ High-speed USB Hub Controller	2.0	High
µPD720110A	NEC	USB2.0 HUB CONTROLLER	2.0	High
µPD72012	NEC	USB HUB CONTROLLER	1.1	Full
µPD720160A	NEC	USB2.0 HUB CONTROLLER	2.0	High
ISP1122	Philips	Universal Serial Bus stand-alone hub	1.1	Full
USB20H04	SMSC	Hi-speed USB HUB controller IC (OTG)	2.0	High
TUSB2036	Texas Instruments	2/3-port HUB with optional serial EEPROM inter.	1.1	Full
TUSB2046B	Texas Instruments	4-port HUB with optional serial EEPROM interface	1.1	Full
TUSB2077A	Texas Instruments	7-port HUB with optional serial EEPROM interface	1.1	Full
W81181	Winbond	USB HUB Controller	1.1	Full

Tabulka A.7 - Přehled obvodů USB – Řadič rozbočovače se společným zařízením.

Device	Manufacturer	Description	USB spec.	USB speed
AT43USB320A	Atmel	Full-speed USB µController with an Embedded Hub	1.1	Full
AT43USB326	Atmel	Multimedia USB keyboard Controller with Hub	1.1	Low,Full
AT43USB353M	Atmel	Full-speed USB µController with Embedded Hub, ADC and PWM	1.1	Full
AT43USB355	Atmel	Full-speed USB µController with Embedded Hub, ADC and PWM	1.1	Full
CY7C65013	Cypress	USB Hub with Microcontroller	1.1	Full
CY7C65113	Cypress	USB Hub with Microcontroller	1.1	Full
CY7C66013	Cypress	Full-Speed USB Peripheral Controller with Integrated Hub	1.1	Full
CY7C66113	Cypress	Full-Speed USB Peripheral Controller with Integrated Hub	1.1	Full
FT8U100AX	FTDI	7 Port USB Hub Controller with Embedded Peripheral Functions	1.1	Full
8x930Hx	Intel	USB HUB Peripheral Controller	1.0	Full
8x931HA	Intel	USB Peripheral Controller with HUB	1.0	Full
M38K27	Mitsubishi	Single-chip 8bit CMOS Microcomputer	2.0	Full
M38K29	Mitsubishi	Single-chip 8bit CMOS Microcomputer	2.0	Full
MC68HC08KH12	MOTOROLA	Low-cost, high performance 8-bit µController (family M68HC08)	1.1	Full
MC68HC708KH12	MOTOROLA	Low-cost, high performance 8-bit µController (family M68HC08)	1.1	Full
MC68HC908LD64	MOTOROLA	Low-cost, high performance 8-bit µController (family M68HC08)	1.1	Full
ISP1123	Philips	Universal Serial Bus compound hub	1.1	Full
PDIUSBH11	Philips	Universal Serial Bus Hub	1.0	Low,Full
PDIUSBH12	Philips	USB 2-port hub	1.1	Low,Full
USB97C102	SMSC	USB Peripheral Controller with 5 ports HUB	1.1	Full
TUSB2136	Texas Instrum.	USB Keyboard HUB Controller	1.1	Full
TUSB5052	Texas Instrum.	USB to 2-serial port controller with configurable optional HUB	1.1	Full
W81282	Winbond	USB Keyboard Controller with 4 ports HUB	1.1	Full

A.1.6 Koncové zařízení

Tabulka A.8 - Přehled obvodů USB – Obvod rozhraní mezi USB a MCU.

Device	Manufacturer	Description	USB spec.	USB speed
M5611B	Ali	Full-speed USB Device Controller	1.1	Full
M5622	Ali	High-Speed USB2.0 Interface Controller	2.0	High
CY7C68001	Cypress	EZ-USB® SX2™ High-Speed USB Interface Device	2.0	Full,High
SL811S/T	Cypress	USB Dual Speed Slave Controller	1.1	Low,Full
S1R72013	Epson	USB2.0 Device Controller LSI	2.0	Full,High
USS-820D	Lucent Technologies	USB Device Controller	1.1	Full
M66290A	Mitsubishi	USB DEVICE CONTROLLER	1.1	Full
USBN9602	National Semiconductor	Full Speed Node Controller with Enhanced DMA Support	1.1	Full
USBN9603	National Semiconductor	Full Speed Node Controller with Enhanced DMA Support	1.1	Full
USBN9604	National Semiconductor	Full Speed Node Controller with Enhanced DMA Support	1.1	Full
ML60851E	Oki Semiconductor	USB Device Controller	2.0	Full
ML60852A	Oki Semiconductor	USB Device Controller	1.1	Full
ISP1181	Philips	Full-speed Universal Serial Bus interface device	2.0	Full
ISP1581	Philips	Hi-Speed Universal Serial Bus interface device	2.0	High
PDIUSBD11	Philips	USB device with serial interface	1.1	Low,Full
PDIUSBD12	Philips	USB interface device with parallel bus	2.0	Low,Full

Tabulka A.9 - Přehled obvodů USB – Mikrokontroléry s USB.

Device	Manufacturer	Description	USB spec.	USB speed
M5617	Ali	Full-Speed USB Scanner Controller	1.1	Full
M5623	Ali	High-Speed USB2.0 Scanner Controller	2.0	High
M5625	Ali	High-Speed USB2.0 Integrated Scanner Controller	2.0	High
ADSP-2192M	Analog Devices	Dual core DSP Microcomputer	1.1	Full
AT43USB351M	Atmel	FS/LS USB Microcontroller with ADC and PWM	1.1	Low,Full
AT76C711	Atmel	AVR®-FS USB and Fast Serial Asynchronous Interfaces	1.1	Full
AT83C5132	Atmel	USB Microcontroller with 64K Bytes ROM	1.1	Full
AT89C5131	Atmel	8-bit Flash Microcontroller with Full Speed USB Device	1.1	Full
AT89C5132	Atmel	USB Microcontroller with 64K Bytes Flash	1.1	Full
AT90SC6464C-USB	Atmel	Secure Microcontroller	1.1	Full
AN2131	Cypress	EZ-USB Microprocessor	1.1	Full
CS5954AM	Cypress	USB Controller for NAND Flash	1.1	Full
CY7C63221A	Cypress	enCoRe™ USB Low-speed USB Peripheral Controller	2.0	Low
CY7C63231A	Cypress	enCoRe™ USB Low-speed USB Peripheral Controller	2.0	Low
CY7C63411	Cypress	Low-speed USB Peripheral Controller	2.0	Low
CY7C63412	Cypress	Low-speed USB Peripheral Controller	2.0	Low
CY7C63413	Cypress	Low-speed USB Peripheral Controller	2.0	Low
CY7C63511	Cypress	Low-speed USB Peripheral Controller	2.0	Low

CY7C63512	Cypress	Low-speed USB Peripheral Controller	2.0	Low
CY7C63513	Cypress	Low-speed USB Peripheral Controller	2.0	Low
CY7C63612	Cypress	Low-speed USB Peripheral Controller	2.0	Low
CY7C63613	Cypress	Low-speed USB Peripheral Controller	2.0	Low
CY7C63722	Cypress	enCoRe™ USB LS USB & PS/2 Peripheral Controller	2.0	Low
CY7C63723	Cypress	enCoRe™ USB LS USB & PS/2 Peripheral Controller	2.0	Low
CY7C63743	Cypress	enCoRe™ USB LS USB & PS/2 Peripheral Controller	2.0	Low
CY7C64013	Cypress	Full-Speed USB (12 Mbps) Function	1.1	Full
CY7C64113	Cypress	Full-Speed USB (12 Mbps) Function	1.1	Full
CY7C64613	Cypress	EZ-USB FX™ USB Microcontroller	2.0	Full
CY7C68013	Cypress	EZ-USB FX2™ High-speed USB Peripheral Controller	2.0	High
SL11R	Cypress	USB Controller/16-Bit RISC Processor	1.1	Full
HD6432210	Hitachi	High-performance microcomputer with H8S/2000 CPU	1.1	Full
HD6432211	Hitachi	High-performance microcomputer with H8S/2000 CPU	1.1	Full
HD6432215	Hitachi	High-performance microcomputer with H8S/2000 CPU	1.1	Full
HD6432217	Hitachi	High-performance microcomputer with H8S/2000 CPU	1.1	Full
HD64F2212	Hitachi	High-performance microcomputer with H8S/2000 CPU	1.1	Full
HD64F2215	Hitachi	High-performance microcomputer with H8S/2000 CPU	1.1	Full
HD64F2218	Hitachi	High-performance microcomputer with H8S/2000 CPU	1.1	Full
SH7705	Hitachi	32-bit RISC-type SuperH microprocessor	1.1	Full
HT48RB8	Holtek	8-Bit USB Type OTP MCU	1.1	Low
C165UTAH	Infineon Technologies	Embedded C166 with USB, HDLC, IOM-2	1.1	Full
SLE66CUX640P	Infineon Technologies	USB Security IC	1.1	Low,Full
79RC32355	Integrated Device Tech.	IDT Interprise Integrated Communications Processor	1.1	Full
8x930Ax	Intel	USB MICROCONTROLLER	1.0	Full
8x931AA	Intel	USB Peripheral Controller	1.0	Full
M37532	Mitsubishi	Single-chip 8bit CMOS Microcomputer	1.1	Low
M37534	Mitsubishi	Single-chip 8bit CMOS Microcomputer	1.1	Low
M37536	Mitsubishi	Single-chip 8bit CMOS Microcomputer	1.1	Low
M37640	Mitsubishi	Single-chip 8bit CMOS Microcomputer	1.1	Full
M37641	Mitsubishi	Single-chip 8bit CMOS Microcomputer	1.1	Full
M38K07	Mitsubishi	Single-chip 8bit CMOS Microcomputer	2.0	Full
M38K09	Mitsubishi	Single-chip 8bit CMOS Microcomputer	2.0	Full
MC68HC05JB3	MOTOROLA	8-bit microcontroller (family M68HC05)	1.0	Low
MC68HC08JB1	MOTOROLA	8-bit microcontroller (family M68HC08)	1.1	Low
MC68HC08JB8	MOTOROLA	8-bit microcontroller (family M68HC08)	1.1	Low
MC68HC908BD48	MOTOROLA	8-bit microcontroller (family M68HC08)	1.0	Low
MC68HC908JB8	MOTOROLA	8-bit microcontroller (family M68HC08)	1.1	Low
LM9701	National Semiconductor	Digital Image Processor w/ Integrated Pre-processor	1.1	Full
UDA1321	Philips	USB Digital-to-Analog Converter (DAC)	1.1	Full
UDA1325	Philips	USB Audio CODEC	1.1	Full
UDA1331H	Philips	Audio Playback Peripheral (APP)	1.1	Full
LC87F1564A	SANYO	USB Microcontroller	1.1	Full
LC87F5316A	SANYO	USB Microcontroller	1.1	Low
LC87F5664A	SANYO	USB Microcontroller	1.1	Full
USB97C210	SMSC	USB 2.0 Flash Media Controller	2.0	High
USB97C211	SMSC	USB 2.0 Flash Media Controller	2.0	High
USB97C242	SMSC	USB 2.0 Flash Drive Controller	2.0	High
73S1121F	TDK	Smart Card Terminal Controller	1.1	Full
TAS1020B	Texas Instruments	USB Streaming Controller (STC)	1.1	Full
TUSB3200	Texas Instruments	USB Streaming Controller (STC)	1.1	Full
TUSB3210	Texas Instruments	USB General Purpose Device Controller	2.0	Full
TMP92CH21	Toshiba	32bit Microcontroller	1.1	Full
W81281	Winbond	USB Keyboard/Device Controller	1.0	Low
W81E381	Winbond	FULL SPEED USB INTEGRATED MICROCONTROLLER	1.1	Full

Tabulka A.10 - Přehled obvodů USB – Převodníky USB.

Device	Manufacturer	Description	USB spec.	USB speed
M5603	Ali	Full-Speed USB Video Controller	1.1	Full
M5603C	Ali	High-Speed USB2.0 Video Camera Controller	2.0	Full,High
M5615	Ali	Full-speed USB-based JPEG Controller	1.1	Full
M5619	Ali	Full-speed USB-to-IDE Interface Controller	1.1	Full
M5621	Ali	High-Speed USB2.0 IDE Device Controller	2.0	Full,High
M5632	Ali	USB2.0 Host-to-Host Networking Controller	2.0	Full,High
CS22210	Cirrus Logic	Wireless PCI/USB Controller	1.1	Full
AN2720SC	Cypress	Single-Chip USB-to-USB Networking Solution	1.1	Full
CY7C68300	Cypress	EZ-USB AT2™ USB 2.0 to ATA/ATAPI Bridge	2.0	Full,High
CYWUSB6942	Cypress	WirelessUSB™ Baseband IC		
ISD-300A1	Cypress	High Speed USB to ATA ASIC	2.0	Full,High
SL11RIDE	Cypress	USB to IDE/ATAPI Solution	1.1	Full
DS2490	Dallas Semicond.	USB to 1-Wire Bridge Chip	1.1	Full

S1R72003	Epson	USB2.0 <-> IDE Controller LSI	2.0	Full,High
FT232BM	FTDI	USB UART (USB - Serial) I.C.	2.0	Full
FT245BM	FTDI	USB FIFO (USB - Parallel) I.C.	2.0	Full
FT8U232AM	FTDI	U-UART - USB UltraBaud Data Transfer IC with RS232/RS422	1.1	Full
FT8U245AM	FTDI	USB FIFO - Fast Parallel Data Transfer IC	1.1	Full
USS-720	Lucent Technolog.	USB-to-IEEE 1284 (parallel) Bridge	1.0	Full
µPD720121	NEC	USB 2.0 to IDE bridge	2.0	High
µPD720130	NEC	USB 2.0 to IDE bridge	2.0	High
SAA8115HL	Philips	Digital camera USB interface		
SAA8116	Philips	Digital PC-camera signal processor and USB interface	1.1	Full
SAA8117HL	Philips	Digital camera USB interface IC		
SAA8122A	Philips	Digital Still Camera Processor (ImagIC family)	1.1	Full
USB97C201	SMSC	USB 2.0 ATA/ATAPI Controller	2.0	High
USB97C202	SMSC	USB 2.0 ATA/ATAPI Controller	2.0	High
USB97CFDC	SMSC	USB Floppy disk controller	1.1	Full
USB97CFDC2-1	SMSC	USB Floppy disk controller	1.1	Full
PCM2702	Texas Instruments	16-Bit Stereo digital-to-analog converter with USB interface	1.0	Full
TUSB3410	Texas Instruments	USB to serial port controller	1.1	Full
W81386	Winbond	USB Interface MS/SD/MMC reader	1.1	Full

A.1.7 Ostatní obvody

Tabulka A.11 - Přehled obvodů USB – Ostatní obvody.

Device	Manufacturer	Description
PMB 6811	Infineon Technologies	Power Management for Cellular Systems
LT1316	Linear Technology	Micropower DC/DC Converter with Programmable Peak Current Limit
LT1618	Linear Technology	Constant-Current/Constant-Voltage 1.4MHz Step-Up DC/DC Converter
LTC4053-4.2	Linear Technology	USB Compatible Lithium-Ion Battery Charger with Thermal Regulation
LTC4410	Linear Technology	USB Power Manager in ThinSOT
USB0803	Microsemi	Transient Voltage Suppressor
USB0805	Microsemi	Transient Voltage Suppressor
USB0812	Microsemi	Transient Voltage Suppressor
USB0815	Microsemi	Transient Voltage Suppressor
USB0824	Microsemi	Transient Voltage Suppressor
NUF2101W1T1	ON Semiconductor	USB Filter with ESD Protection
NUF2102UT1	ON Semiconductor	USB Filter with ESD Protection
NUF2221W1T2	ON Semiconductor	USB Upstream Terminator with ESD Protection
STF202-22	ON Semiconductor	USB Filter with ESD Protection

A.2 Katalog řadičů a mikrokontrolérů s implementací USB

Tabulka A.12 - Řadiče rozhraní USB. (seřazeno podle výrobce)

Device	Manufacturer	Description	Supply(V)	Interface	USB	speed	USB ports
M5611B	Ali	Full-speed USB Device Controller	5	8b PIO/DMA	1.1	Full	DEVICE: 1xBulk(64Bx2),2xInt/Bulk(8B)
M5622	Ali	High-Speed USB2.0 Interface Controller		8/16b PIO/DMA	2.0	Full,High	DEVICE: 2xBulk(512B),1xInt/Bulk(8B)
CY7C68001	Cypress	EZ-USB® SX2™ HS USB Interface Device	3.3	8/16b PIO	2.0	Full,High	DEVICE: up to 4xBulk with up to 1024B FIFO
SL811HS	Cypress	Embedded USB Host/Slave Controller	3.3	8b PIO/DMA	1.1	Low,Full	HOST: 1x | DEVICE: 6xBulk/Iso/Int
SL811S/T	Cypress	USB Dual Speed Slave Controller	3.3	8b PIO/DMA	1.1	Low,Full	DEVICE: 3xTx,3xRx
S1R72005	Epson	On-The-Go device controller	3.3,2.5	PIO/DMA	2.0	Full	HOST: 1xOTG | DEVICE: 5xBulk/Iso/Int
S1R72013	Epson	USB2.0 Device Controller LSI	3.3	8b PIO/(8/16b)DMA	2.0	Full,High	DEVICE: 3xBulk/Int
USS-820D	Lucent Technologies	USB Device Controller	3/5	8b PIO	1.1	Full	DEVICE: 14xBulk/Iso/Int(up to 64/1024)
M66290A	Mitsubishi	USB DEVICE CONTROLLER	3.3	16b PIO ,JTAG	1.1	Full	DEVICE: 5xBulk/Iso/Int (up to 1024)
USBN9602	National Semiconductor	FS Controller with Enhanced DMA Support	3.3	8b PIO/DMA	1.0	Full	DEVICE: 6xBulk/Int(4x32B,2x64B)
USBN9603	National Semiconductor	FS Controller with Enhanced DMA Support	5,3.3	8b PIO/DMA	1.1	Full	DEVICE: 6xBulk/Int/Iso(64B)
USBN9604	National Semiconductor	FS Controller with Enhanced DMA Support	5,3.3	8b PIO/DMA	1.1	Full	DEVICE: 6xBulk/Int/Iso(64B)
ML60851E	Oki Semiconductor	USB Device Controller	5,3.3	8/16b PIO/DMA	2.0	Full	DEVICE: 3xBulk(64B),1xInt(8B)
ML60852A	Oki Semiconductor	USB Device Controller	5,3.3	8/16b PIO/DMA	1.1	Full	DEVICE: 4xBulk/Int(64B),Bulk/Int(32B),4xBulk/Iso/Int(256B)
ISP1160	Philips	Embedded USB Host Controller	5,3.3	16b PIO/DMA	2.0	Low,Full	HOST: 2xOHCI
ISP1161	Philips	FS USB single-chip host and device controller	5,3.3	16b PIO/DMA	2.0	Low,Full	HOST: 2xOHCI compatible | DEVICE: 14xBulk/Iso/Int
ISP1181	Philips	FS USB interface device	5,3.3	8/16b PIO/DMA	1.1	Full	DEVICE: 14xBulk/Iso/Int(up to 1024)
ISP1362	Philips	Single-chip USB On-The-Go controller	3.3	16b PIO/DMA	2.0	Low,Full	HOST: 1OHCI+1xHOST/OTG/DEVICE | DEVICE: 14xBulk/Iso/Int
ISP1581	Philips	Hi-Speed Universal Serial Bus interface device	5,3.3	8/16b PIO/DMA	2.0	High	DEVICE: 7xTx,7xRx Bulk/Iso/Int
PDIUSBD11	Philips	USB device with serial interface	3.3	I2C	1.1	Full	DEVICE: 8xBulk/Int(8B)
PDIUSBD12	Philips	USB interface device with parallel bus	5,3.3	8b PIO/DMA	2.0	Full	DEVICE: 2xBulk/Int(16B)+(2xBulk/Int/Iso(64B) or Iso(128B))
PDIUSBH11	Philips	Universal Serial Bus Hub	3.3	I2C	1.1	Low,Full	HUB: 4x | DEVICE: 6xBulk/Int
PDIUSBH12	Philips	USB 2-port hub	3.3	I2C	1.1	Low,Full	HUB: 2x | DEVICE: 6xBulk/Int

Tabulka A.13 - Mikrokontroléry USB. (seřazeno podle obvodu)

DEVICE	Description								
	CORE	RAM [B]	ROM [B]	EPROM [B]	EEPROM [B]	FLASH [B]	Timer	ADC	PWM
Manufacturer		Freq. [MHz]	Supply [V]	Other interfaces					
	USB	specification	Speed	USB Ports					
73S1121F	Smart Card Terminal Controller for e-Transaction, Smart Card Reader Applications								
	8052	1K+4Kxram	4K			64K	3x16b		
TDK		48	3-3.3	WDT,I2C,UART,DMA,5x6KBD,LCD,SmC,RTC,DCDC					
	USB	1.1	Full	**4x**					
80930AD	USB Microcontroller								
	x251	1K					3x16b		
Intel		12	5	WDT,UART,PCA					
	USB	1.1	Low,Full	**6xBulk/Int/Iso(16B),2xIso(up to 1023)**					
80931AA	USB Peripheral Controller								
	x51	256					3x16b		
Intel		12	5	WDT,UART,KBD					
	USB	1.1	Low,Full	**4xBulk/Int/Iso(8/16B)**					
83930AD	USB Microcontroller								
	x251	1K	8K				3x16b		
Intel		12	5	WDT,UART,PCA					
	USB	1.1	Low,Full	**6xBulk/Int/Iso(16B),2xIso(up to 1023)**					
83930AE	USB Microcontroller								
	x251	1K	16K				3x16b		
Intel		12	5	WDT,UART,PCA					
	USB	1.1	Low,Full	**6xBulk/Int/Iso(16B),2xIso(up to 1023)**					
83931AA	USB Peripheral Controller								
	x51	256	8K				3x16b		
Intel		12	5	WDT,UART,KBD					
	USB	1.1	Low,Full	**4xBulk/Int/Iso(8/16B)**					
8x930Hx	USB HUB Peripheral Controller								
	x251	1K	0/8/16K				1x8b,2x16b		
Intel		12	5	UART,WDT					
	USB	1.0	Low,Full	**3x16B HUB: 3x**					
8x931HA	USB Peripheral Controller with HUB								
	x51	1K	0/8K				3x16b		
Intel		12	3.3	UART,WDT					
	USB	1.0	Low,Full	**1x8B,1x16B HUB: 4x**					
ADSP-2192M	Dual core DSP Microcomputer								
	2xADSP(320MIPS)	264K					3x32b		
Analog Devices				PCI,Sub-ISA,CardBus,JTAG,DMA					
	USB	1.1	Full	**3xBulk(64B)+8xBulk/Int/Iso**					

DEVICE	Description								
	CORE	RAM [B]	ROM [B]	EPROM [B]	EEPROM [B]	FLASH [B]	Timer	ADC	PWM
Manufacturer		Freq. [MHz]	Supply [V]	Other interfaces					
	USB	specification	Speed	USB Ports					
AN2131	EZ-USB Microprocessor								
	enhanced 8051	256+8Kcode					3x16b		
Cypress			3.3	2xUART,I2C,POR					
	USB	1.1	Full	**7xTx+7xRx Bulk/Int,8xTx+8xRx Iso**					
AN2135	EZ-USB Microprocessor								
	enhanced 8051	256+8Kcode					3x16b		
Cypress			3.3	2xUART,I2C,POR					
	USB	1.1	Full	**7xTx+7xRx Bulk/Int,8xTx+8xRx Iso**					
AN2136	EZ-USB Microprocessor								
	enhanced 8051	256+8Kcode					1x8b,2x16b		
Cypress			3.3	2xUART,I2C,POR					
	USB	1.1	Full	**7xTx+7xRx Bulk/Int/Iso**					
AT43USB320A	Full-speed USB Microcontroller with an Embedded Hub								
	AVR 8-bit RISC(83ns)	512					1x8b,2x16b		2x
Atmel			5	UART,SPI,WDT					
	USB	1.1	Low,Full	**2xBulk/Int HUB: 4x**					
AT43USB326	Multimedia USB keyboard Controller with Hub								
	AVR 8-bit RISC(83ns)	512	16K				1x8b		
Atmel			5	WDT,18x8 KBD,					
	USB	1.1	Low,Full	**2xBulk/Int HUB: 2x**					
AT43USB351M	Full-speed/Low-speed USB Microcontroller with ADC and PWM								
	AVR 8b RISC(83ns)	1K	24K				1x8b,1x16b	12ch x 10b	2x
Atmel		24	5	SPI,WDT					
	USB	1.1	Full	**2xBulk/Int/Iso(8B),2xBulk/Int/Iso(64B)**					
AT43USB353M	Low Cost Full-speed USB Microcontroller with Embedded Hub, ADC and PWM								
	AVR 8b RISC(41.5ns)	1K	24K				1x8b,1x16b	12ch x 10b	2x
Atmel		24	5	WDT					
	USB	2.0	Low,Full	**2x64B,1x8B HUB: 2x**					
AT43USB355	Full-speed USB Microcontroller with Embedded Hub, ADC and PWM								
	AVR 8b RISC(83ns)	1K	24K (24K SRAM)				1x8b,1x16b	12ch x 10b	2x
Atmel		12	5	SPI,WDT					
	USB	2.0	Low,Full	**2x64B,1x8B HUB: 2x**					
AT76C711	AVR®-based Bridge between Full-speed USB and Fast Serial Asynchronous Interfaces								
	AVR 8b RISC(83ns)	2K+2K+16Kcode	bootload				1x8b,1x16b		
Atmel		24	3.3	SPI,WDT,UART,IrDA,DMA					
	USB	1.1	Full	**6xBulk/Int/Iso(up to 64B)**					
AT83C5132	USB Microcontroller with 64K Bytes ROM								
	C51 8b (max.20MHz)	2304	64K				2x16b	2ch x 10b (8kHz)	
Atmel		24	3	SPI,WDT,UART,MultiMedia Card,KBD,I2S/PCM					
	USB	1.1	Full	**2xBulk/Int/Iso(64B)+1xBulk/Int/Iso(8B)**					

DEVICE	Description								
	CORE	RAM [B]	ROM [B]	EPROM [B]	EEPROM [B]	FLASH [B]	Timer	ADC	PWM
Manufacturer		Freq. [MHz]	Supply [V]	Other interfaces					
	USB	specification	Speed	USB Ports					
AT89C5131	8-bit Flash Microcontroller with Full Speed USB Device								
	C51 8b (max.20MHz)	256+1Kxram			4K	32K	3x16b	2ch x 10b (8kHz)	5ch
Atmel		48	3.3	SPI,WDT,eUART,KBD,PCA					
	USB	1.1	Full	**3x(32B),2x(2x64B),1x(2x512B)**					
AT89C5132	USB Microcontroller with 64K Bytes Flash								
	C51 8b (max.20MHz)	2304				64K+4K(boot)	2x16b	2ch x 10b (8kHz)	
Atmel		24	3	SPI,WDT,UART,MultiMedia Card,KBD,I2S/PCM					
	USB	1.1	Full	**2xBulk/Int/Iso(64B)+1xBulk/Int/Iso(8B)**					
AT90SC6464C-USB	Secure Microcontroller								
	AVR3 RISC	3K			64K	64K	2x16b	2ch x 10b (8kHz)	5ch
Atmel			3.3	DMA,RNG,DES,3DES,crypto					
	USB	1.1	Full	**2xBulk(2x64B),1xCtrl/Bulk/Iso/Int(64B)**					
C165UTAH	Embedded C166 with USB, HDLC, IOM-2								
	16/32b C166 (RISC)	256+3Kiram					5x16b		
Infineon Technologies		36	3.3	WDT,RTC,IOM-2,SSC,UART					
	USB	1.1	Full	**7xTx+7xRx Bulk/Int/Iso(up to 1023)**					
CS5954AM	USB Controller for NAND Flash								
	16b RISC	3K	6K				2x		
Cypress		48	3.3	WDT					
	USB	2.0	Full	**3xBulk/Int(up to 64B)or Iso(up to 1024B)**					
CY7C63221A	enCoRe™ USB Low-speed USB Peripheral Controller								
	8b RISC	96		3K			12b free-running		
Cypress		12	5	WDT,POR,PS/2-USB					
	USB	2.0	Low	**1xInt(8B)**					
CY7C63231A	enCoRe™ USB Low-speed USB Peripheral Controller								
	8b RISC	96		3K			12b free-running		
Cypress		12	5	WDT,POR,PS/2-USB					
	USB	2.0	Low	**1xInt(8B)**					
CY7C63411	Low-speed USB Peripheral Controller								
	8b RISC	256		4K			12b free-running (1us)		
Cypress		12	5	WDT,POR,PS/2-USB					
	USB	1.0	Low	**2xInt(8B)**					
CY7C63412	Low-speed USB Peripheral Controller								
	8b RISC	256		6K			12b free-running (1us)		
Cypress		12	5	WDT,POR,PS/2-USB					
	USB	1.0	Low	**2xInt(8B)**					
CY7C63413	Low-speed USB Peripheral Controller								
	8b RISC	256		8K			12b free-running (1us)		
Cypress		12	5	WDT,POR,PS/2-USB					
	USB	1.0	Low	**2xInt(8B)**					

DEVICE	Description								
	CORE	RAM [B]	ROM [B]	EPROM [B]	EEPROM [B]	FLASH [B]	Timer	ADC	PWM
Manufacturer		Freq. [MHz]	Supply [V]	Other interfaces					
	USB	specification	Speed	USB Ports					
CY7C63511	Low-speed USB Peripheral Controller								
	8b RISC	256		4K			12b free-running (1us)		
Cypress		12	5	WDT,POR,PS/2-USB,DAC(8b)					
	USB	1.0	Low	**2xInt(8B)**					
CY7C63512	Low-speed USB Peripheral Controller								
	8b RISC	256		6K			12b free-running (1us)		
Cypress		12	5	WDT,POR,PS/2-USB,DAC(8b)					
	USB	1.0	Low	**2xInt(8B)**					
CY7C63513	Low-speed USB Peripheral Controller								
	8b RISC	256		8K			12b free-running (1us)		
Cypress		12	5	WDT,POR,PS/2-USB,DAC(8b)					
	USB	1.0	Low	**2xInt(8B)**					
CY7C63612	Low-speed USB Peripheral Controller								
	8b RISC	256		6K			12b free-running (1us)		
Cypress		12	5	WDT,POR,PS/2-USB					
	USB	1.0	Low	**2xInt(8B)**					
CY7C63613	Low-speed USB Peripheral Controller								
	8b RISC	256		8K			12b free-running (1us)		
Cypress		12	5	WDT,POR,PS/2-USB					
	USB	1.0	Low	**2xInt(8B)**					
CY7C63723	enCoRe™ USB Combination Low-Speed USB & PS/2 Peripheral Controller								
	8b RISC	256		8K			12b free-running (1us)		
Cypress		12	5	WDT,POR,PS/2-USB,SPI,					
	USB	2.0	Low	**2xInt(8B)**					
CY7C63743	enCoRe™ USB Combination Low-Speed USB & PS/2 Peripheral Controller								
	8b RISC	256		8K			12b free-running (1us)		
Cypress		12	5	WDT,POR,PS/2-USB,SPI,					
	USB	2.0	Low	**2xInt(8B)**					
CY7C64013	Full-Speed USB (12 Mbps) Function								
	8b RISC	256	8K PROM				12b free-running (1us)		
Cypress		12/48	5	WDT,POR,I2C					
	USB	1.1	Full	**4xInt/Iso(2x8B+2x32B)**					
CY7C64113	Full-Speed USB (12 Mbps) Function								
	8b RISC	256	8K PROM				12b free-running (1us)		
Cypress		12/48	5	WDT,POR,I2C,DAC(8b)					
	USB	1.1	Full	**4xInt/Iso(2x8B+2x32B)**					
CY7C64613	EZ-USB FX™ USB Microcontroller								
	enhanced 8051	256+8Kcode	8K PROM				3x16b		
Cypress		24/48	3.3	2xUART,DMA,I2C					
	USB	2.0	Full	**14xBulk/Int(up to 64),16xIso(up to 1023)**					

DEVICE	Description								
	CORE	RAM [B]	ROM [B]	EPROM [B]	EEPROM [B]	FLASH [B]	Timer	ADC	PWM
Manufacturer		Freq. [MHz]	Supply [V]	Other interfaces					
	USB	specification	Speed	USB Ports					
CY7C65013	USB Hub with Microcontroller								
	8b optimized	256	8K				12b free-running (1us)	12ch x 10b	2x
Cypress		12	5	I2C,WDT,POR,					
	USB	1.1	Low,Full	**5x(max. 4x8B,2x32B) HUB: 7x**					
CY7C66013	Full-Speed USB Peripheral Controller with Integrated Hub								
	8b optimized	256	8K				12b free-running (1us)		
Cypress		12	5	I2C,WDT,POR					
	USB	1.1	Low,Full	**5x(max. 4x8B,2x32B) HUB: 4x**					
CY7C66113	Full-Speed USB Peripheral Controller with Integrated Hub								
	8b optimized	256	8K				12b free-running (1us)		
Cypress		12	5	I2C,WDT,POR,DAC					
	USB	1.1	Low,Full	**5x(max. 4x8B,2x32B) HUB: 4x**					
CY7C68013	EZ-USB FX2™ USB Microcontroller High-speed USB Peripheral Controller								
	enhanced 8051	512+8Kcode	8K PROM				3x16b		
Cypress		12/24/48	3.3	2xUART,DMA,I2C					
	USB	2.0	Full,High	**2xBulk/Int(64B),8xBulk/Int/Iso(512B)**					
FT8U100AX	FT8U100AX Seven Port USB Hub Controller with Embedded Peripheral Functions								
	8b optimized	256					12b free-running (1us)		
FTDI		48	3.3	RS232,IrDA,PS/2,I2C,					
	USB	1.1	Low,Full	**2xBulk/Int HUB: 7x**					
HD6432210	High-performance microcomputer with H8S/2000 CPU								
	H8/300	4K	32K				3x16b	4ch x 10b	7ch
Hitachi		24	3-3.3	DMA,WDT,RTC,SCI					
	USB	1.1	Full	**2xBulk(64)+1xInt(64)**					
HD6432211	High-performance microcomputer with H8S/2000 CPU								
	H8/300	8K	64K				3x16b	4ch x 10b	7ch
Hitachi		24	3-3.3	DMA,WDT,RTC,SCI					
	USB	1.1	Full	**2xBulk(64)+1xInt(64)**					
HD6432215	High-performance microcomputer with H8S/2000 CPU								
	H8/300	8/16K	64/128/256K				2x16b	6ch x 10b	7ch
Hitachi			3-3.3	DMA,WDT,RTC,SCI,DAC,					
	USB	1.1	Full	**2xBulk(64)+1xInt(64)**					
HD6432217	High-performance microcomputer with H8S/2000 CPU								
	H8/300	8K	64K				3x16b	4ch x 10b	7ch
Hitachi		24	3-3.3	DMA,WDT,RTC,SCI					
	USB	1.1	Full	**2xBulk(64)+1xInt(64)**					
HD64F2212	High-performance microcomputer with H8S/2000 CPU								
	H8/300	12K				128K	3x16b	6ch x 10b	7ch
Hitachi			3-3.3	DMA,WDT,RTC					
	USB	1.1	Full	**2xBulk(64)+1xInt(64)**					

DEVICE	Description								
	CORE	RAM [B]	ROM [B]	EPROM [B]	EEPROM [B]	FLASH [B]	Timer	ADC	PWM
Manufacturer		Freq. [MHz]	Supply [V]	Other interfaces					
	USB	specification	Speed	USB Ports					
HD64F2218	High-performance microcomputer with H8S/2000 CPU								
	H8/300	12K				128K	3x16b	6ch x 10b	7ch
Hitachi			3-3.3	DMA,WDT,RTC,SCI					
	USB	1.1	Full	**2xBulk(64)+1xInt(64)**					
HD6F32215	High-performance microcomputer with H8S/2000 CPU								
	H8/300	16K				256K	2x16b	6ch x 10b	7ch
Hitachi			3-3.3	DMA,WDT,RTC,SCI,DAC,					
	USB	1.1	Full	**2xBulk(64)+1xInt(64)**					
HT48RB8	8-Bit USB Type OTP MCU								
	8b RISC	160	4096x15b				1x8b,1x16b	6ch x 8b	
Holtek		12	5	WDT,PS/2					
	USB	1.1	Low	**3xInt**					
LC77700B	IAPr™ (INTERNET ACCESS PROCESSOR) (USB host:2ports,func:1port)								
	PowerPC405								
SANYO		192	3.3,2.5	LCD SVGA,mem.ctrl,PCMCIA,PS/2,IEEE1284,Keyb.,IrDA,					
	USB	2.0	Low,Full	**1x HOST: 2x**					
LC87F1564A	USB Microcontroller								
	??? RISC	2Kx9b				64K	2x16b/(2x8b)+4x8b	12ch	2x12b+1x16b(2x8b)
SANYO		12	3-5.5	WDT,SIO					
	USB	1.1	Full	**7xBulk/Iso/Int**					
LC87F5316A	USB Microcontroller								
	??? RISC	512x9b				16K	1x16b/(2x8b)	8ch	2x12b
SANYO		6	3-5.5	WDT,SIO					
	USB	1.1	Low	**1xInt**					
LC87F5664A	USB Microcontroller								
	??? RISC	3Kx9b				64K	2x16b/(2x8b)+4x8b	8ch	2x12b+1x16b(2x8b)
SANYO		12	3-5.5	WDT,SIO					
	USB	1.1	Full	**5xBulk/Iso/Int**					
LM9701	Digital Image Processor w/ Integrated Pre-processor								
	32-bit ARM7 RISC	16K+8Kcache					ARM		
National Semiconductor			1.8,3.3	2xUART,DMA,I2C,SPI,JTAG,RTC,POR					
	USB	1.1	Full						
M37532E8	Single-chip 8bit CMOS Microcomputer								
	8b 740 family	384	16K prom				3x8b	8ch x 10b	
Mitsubishi		6	5	WDT,UART					
	USB	1.1	Low	**1xInt**					
M37532M4	Single-chip 8bit CMOS Microcomputer								
	8b 740 family	256	8K				3x8b	8ch x 10b	
Mitsubishi		6	5	WDT,UART					
	USB	1.1	Low	**1xInt**					

DEVICE	Description								
	CORE	RAM [B]	ROM [B]	EPROM [B]	EEPROM [B]	FLASH [B]	Timer	ADC	PWM
Manufacturer		Freq. [MHz]	Supply [V]	Other interfaces					
	USB	specification	Speed	USB Ports					
M37532RSS	Single-chip 8bit CMOS Microcomputer								
	8b 740 family	384					3x8b	8ch x 10b	
Mitsubishi		6	5	WDT,UART					
	USB	1.1	Low	**1xInt**					
M37534E4	Single-chip 8bit CMOS Microcomputer								
	8b 740 family	256	8K prom				3x8b	8ch x 10b	
Mitsubishi		6	5	WDT,UART					
	USB	1.1	Low	**1xInt**					
M37534E8	Single-chip 8bit CMOS Microcomputer								
	8b 740 family	384	16K prom				3x8b	8ch x 10b	
Mitsubishi		6	5	WDT,UART					
	USB	1.1	Low	**1xInt**					
M37534M4	Single-chip 8bit CMOS Microcomputer								
	8b 740 family	256	8K				3x8b	8ch x 10b	
Mitsubishi		6	5	WDT,UART					
	USB	1.1	Low	**1xInt**					
M37534RSS	Single-chip 8bit CMOS Microcomputer								
	8b 740 family	384					3x8b	8ch x 10b	
Mitsubishi		6	5	WDT,UART					
	USB	1.1	Low	**1xInt**					
M37536E8	Single-chip 8bit CMOS Microcomputer								
	8b 740 family	384	16K prom				3x8b	8ch x 10b	
Mitsubishi		6	5	WDT,UART					
	USB	1.1	Low	**1xInt**					
M37536M4	Single-chip 8bit CMOS Microcomputer								
	8b 740 family	256	8K				3x8b	8ch x 10b	
Mitsubishi		6	5	WDT,UART					
	USB	1.1	Low	**1xInt**					
M37536RSS	Single-chip 8bit CMOS Microcomputer								
	8b 740 family	384					3x8b	8ch x 10b	
Mitsubishi		6	5	WDT,UART					
	USB	1.1	Low	**1xInt**					
M37640E8FP	Single-chip 8bit CMOS Microcomputer								
	8b 740 family	1K	32K prom				2x16b,3x8b		
Mitsubishi		12	5	WDT,2xUART,DMA					
	USB	1.1	Full	**8xBulk/Int/Iso(16/32/512B)**					
M37640E8FS	Single-chip 8bit CMOS Microcomputer								
	8b 740 family	1K		32K			2x16b,3x8b		
Mitsubishi		12	5	WDT,2xUART,DMA					
	USB	1.1	Full	**8xBulk/Int/Iso(16/32/512B)**					

DEVICE	Description								
	CORE	RAM [B]	ROM [B]	EPROM [B]	EEPROM [B]	FLASH [B]	Timer	ADC	PWM
Manufacturer		Freq. [MHz]	Supply [V]	Other interfaces					
	USB	specification	Speed	USB Ports					
M37640M8	Single-chip 8bit CMOS Microcomputer								
	8b 740 family	1K	32K				2x16b,3x8b		
Mitsubishi		12	5	WDT,2xUART,DMA					
	USB	1.1	Full	**8xBulk/Int/Iso(16/32/512B)**					
M37641F8	Single-chip 8bit CMOS Microcomputer								
	8b 7600 family	2.5K				32K	2x16b,3x8b		
Mitsubishi		24	5	WDT,2xUART,DMA					
	USB	1.1	Full	**8xBulk/Int/Iso(16/32/up to 2KB)**					
M37641M8	Single-chip 8bit CMOS Microcomputer								
	8b 7600 family	1K	32K				2x16b,3x8b		
Mitsubishi		24	5	WDT,2xUART,DMA					
	USB	1.1	Full	**8xBulk/Int/Iso(16/32/up to 2KB)**					
M38K07M4	Single-chip 8bit CMOS Microcomputer								
	8b 7600 family	1K	16K				3x8b	8ch x 10b	
Mitsubishi		8	5,3.3	WDT,2xUART,DMA					
	USB	2.0	Full	**4x**					
M38K09F8	Single-chip 8bit CMOS Microcomputer								
	8b 7600 family	2K				32K	3x8b	8ch x 10b	
Mitsubishi		8	5,3.3	WDT,2xUART,DMA					
	USB	2.0	Full	**4x**					
M38K27M4	Single-chip 8bit CMOS Microcomputer								
	740 family	1K	16K				3x8b	8ch x 10b	
Mitsubishi			5,3.3	UART,WDT					
	USB	2.0	Low,Full	**4x HUB: 2x**					
M38K29F8	Single-chip 8bit CMOS Microcomputer								
	740 family	2K				32K	3x8b	8ch x 10b	
Mitsubishi			5,3.3	UART,WDT					
	USB	2.0	Low,Full	**4x HUB: 2x**					
MC68HC05JB3	Low-cost, high performance 8-bit microcontroller (family M68HC05)								
	8b M68HC05 family	144	2.5K				1x16b	8ch x 10b	
MOTOROLA		3	5,3.3	WDT,2xUART,DMA					
	USB	1.1	Low	**2xInt(8B)**					
MC68HC08JB1	Low-cost, high performance 8-bit microcontroller (family M68HC08)								
	8b M68HC08 family	128	5632				2x16b		1x on timer
MOTOROLA		3	5,3.3	WDT,2xUART,DMA,PS/2,KBD,					
	USB	1.1	Low	**2xInt(8B)**					
MC68HC08KH12	Low-cost, high performance 8-bit microcontroller (family M68HC08)								
	MC68HC08 family	384	12K			32K	2x16b	8ch x 10b	2x
MOTOROLA		6	5,3.3						
	USB	1.1	Low,Full	**2x8B HUB: 4x**					

DEVICE	Description								
	CORE	RAM [B]	ROM [B]	EPROM [B]	EEPROM [B]	FLASH [B]	Timer	ADC	PWM
Manufacturer		Freq. [MHz]	Supply [V]	Other interfaces					
	USB	specification	Speed	USB Ports					
MC68HC708KH12	Low-cost, high performance 8-bit microcontroller (family M68HC08)								
	MC68HC08 family	384	12K (OTP)			32K	2x16b	8ch x 10b	2x
MOTOROLA		6	5,3.3						
	USB	1.1	Low,Full	**2x8B HUB: 4x**					
MC68HC908BD48	Low-cost, high performance 8-bit microcontroller (family M68HC08)								
	8b M68HC08 family	1K				48K	2x16b	6ch x 8b	16ch x 8b
MOTOROLA		6	5,3.3	WDT					
	USB	1.0	Low	**2xInt(8B)**					
MC68HC908JB8	Low-cost, high performance 8-bit microcontroller (family M68HC08)								
	8b M68HC08 family	256				8K	2x16b	6ch x 8b	2x
MOTOROLA		3	5						
	USB	1.1	Low	**3xInt(8B)**					
MC68HC908LD64	Low-cost, high performance 8-bit microcontroller (family M68HC08)								
	MC68HC08 family	2K				60K	2x16b	6ch x 8b	8ch x 8b
MOTOROLA		6	3.3	I2C					
	USB	1.1	Low,Full	**2x8B HUB: 4x**					
SH7705	32-bit RISC-type SuperH microprocessor								
	SuperH 16b RISC						3x32b	4ch x 10b	4ch
Hitachi		133.3	3.3	DMA,WDT,RTC,UART,IrDA,					
	USB	1.1	Full	**2xBulk(64)+1xInt(8)**					
SL11R	USB Controller/16-Bit RISC Processor								
	16b RISC	3K	6K				2x16b		4ch x 10b
Cypress		48	3.3	2xUART,DMA,WDT					
	USB	1.1	High	**6xBulk/Int/Iso**					
SLE66CUX640P	USB Security IC								
	enhanced 8051	256+700+4Kxram	128K+2K+6K		64K		2x16b		
Infineon Technologies		10	3.3	WDT,DES,3DES,EC2,RNG					
	USB	1.1	Low,Full	**4xTx,4xRx Bulk/Int/Iso(up to 64B)**					
TAS1020B	USB Streaming Controller (STC)								
	8052	256+5.8Kcode	8K				3x16b		
Texas Instruments		48	3.3	2xDMA,I2S,I2C,UART,AC97					
	USB	1.1	Full	**7xTx,7xRx Bulk/Iso/Int**					
TMP92CH21FG	32bit Microcontroller								
	32-bit CPU (900/H1)	16K	8K			20K	4x8b,1x16b	4ch x 10b	4x
Toshiba		40	3,3.3	DMA,4xCS,2xUART,IrDA,I2S,LCD,RTC,WDT,KBD					
	USB	1.1	Full	**2xBulk(64B),1xInt(8B)**					
TUSB2136	USB Keyboard HUB Controller								
	8052	256+8Kcode	6K				2x16b		
Texas Instruments		48	3.3	I2C,WDT					
	USB	1.1	Low,Full	**3Tx + 3Rx HUB: 2x**					

DEVICE	Description								
	CORE	RAM [B]	ROM [B]	EPROM [B]	EEPROM [B]	FLASH [B]	Timer	ADC	PWM
Manufacturer		Freq. [MHz]	Supply [V]	Other interfaces					
	USB	specification	Speed	USB Ports					
TUSB3200	USB Streaming Controller (STC)								
	8052	256+8Kcode	4K						1x
Texas Instruments		48	3.3	4xDMA,I2S,I2C,AC97					
	USB	1.1	Full	**7xTx,7xRx Bulk/Iso/Int**					
TUSB3210	USB General Purpose Device Controller								
	8052	256+8Kcode	6K				2x16b		
Texas Instruments		48	3.3	WDT,I2C					
	USB	2.0	Full	**3xTx,3xRx Bulk/Int**					
TUSB5052	USB to 2-serial port controller with configurable optional HUB								
	8052	256+16Kcode	6K				2x16b		
Texas Instruments		24/48	3.3	I2C,WDT,2x eUART,DMA					
	USB	1.1	Low,Full	**7Tx + 7Rx HUB: 5x**					
UDA1325	USB Audio CODEC								
	8051	256	12K				3x16b		
Philips		48	3.3	I2C,I2S,ADC,PGA,...					
	USB	1.1	Full	**2xInt,2xIso**					
USB97C102	USB Peripheral Controller with 5 ports HUB								
	8051	4K					2x16b		
SMSC		24	3.3	DMA(8237),ISA interface					
	USB	1.1	Low,Full	**up to 15Tx,15Rx with 0-1023B FIFO HUB: 4x**					
USB97C210	USB 2.0 Flash Media Controller								
	8051	768+12K							
SMSC		30	3.3	CF,SM,MS,SD/MMC					
	USB	2.0	Full	**2xBulk(512)**					
USB97C211	USB 2.0 Flash Media Controller								
	8051	768+12K							
SMSC		30	3.3	CF,SM					
	USB	2.0	Full	**2xBulk(512)**					
USB97C242	USB 2.0 Flash Drive Controller								
	8051	768+12K	48K						
SMSC		30	3.3	SM,NAND flash					
	USB	2.0	Full	**2x(64),2xBulk(512)**					
W81281	USB Keyboard/Device Controller								
	8051	256	6K				3x16b		
Winbond			5	WDT,PS/2,18x8KBD					
	USB	1.0	Low,Full	**4xInt(8B)**					
W81282	USB Keyboard Controller with 4 ports HUB								
	8052	256	6K						
Winbond			5	21x8KBD					
	USB	1.1	Low,Full	**HUB: 4x**					

DEVICE	Description								
	CORE	RAM [B]	ROM [B]	EPROM [B]	EEPROM [B]	FLASH [B]	Timer	ADC	PWM
Manufacturer		Freq. [MHz]	Supply [V]	Other interfaces					
	USB	specification	Speed	USB Ports					
W81E381	FULL SPEED USB INTEGRATED MICROCONTROLLER								
	8052	256				20K	3x16b		
Winbond		12	5	WDT,PS/2,18x8KBD,I2C					
	USB	1.0	Low,Full	**4xBulk/Int(16/8B)**					

A.3 Seznam výrobců obvodů uvedených v katalogu

Tabulka A.14 - Seznam výrobců uvedených v katalogu.

Manufacturer	Internet homepage
ALi Corporation	URL:<http://www.ali.com.tw>
Allegro MicroSystems, Inc.	URL:<http://www.allegromicro.com>
Analog Devices, Inc.	URL:<http://www.analog.com>
Atmel Corporation	URL:<http://www.atmel.com>
Cirrus Logic	URL:<http://www.cirrus.com>
Cypress Semiconductor	URL:<http://www.cypress.com>
Dallas Semiconductor	URL:<http://www.maxim-ic.com>
Fairchild Semiconductor Corporation	URL:<http://www.fairchildsemi.com>
Future Technology Devices Intl. Ltd.	URL:<http://www.ftdichip.com>
Hitachi Semiconductor	URL:<http://www.hitachi.com>
Holtek Semiconductor Inc.	URL:<http://www.holtek.com>
Infineon Technologies	URL:<http://www.infineon.com>
Integrated DeviceTechnology, Inc.	URL:<http://www.idt.com>
Intel Corporation	URL:<http://www.intel.com>
Linear Technology Corporation	URL:<http://www.linear.com>
Lucent Technologies Inc.	URL:<http://www.lucent.com>
Maxim Integrated Products	URL:<http://www.maxim-ic.com>
Micrel, Inc.	URL:<http://www.micrel.com>
Microsemi	URL:<http://www.microsemi.com>
Mitsubishi Electric Corporation	URL:<http://www.mitsubishichips.com>
Motorola Semiconductors	URL:<http://www.mot-sps.com>
National Semiconductor Corporation	URL:<http://www.national.com>
NEC Electronics	URL:<http://www.nec.com>
Oki Semiconductor	URL:<http://www.oki.com>
ON Semiconductor	URL:<http://www.onsemi.com>
OPTi Inc.	URL:<http://www.opti.com>
Philips Electronics	URL:<http://www.semiconductors.philips.com>
SANYO Electric Co., Ltd	URL:<http://www.sanyo.com>
Seiko Epson Corporation	URL:<http://www.epsondevice.com>
Sipex Corporation	URL:<http://www.sipex.com>
Standard Microsystems Corporation	URL:<http://www.smsc.com>
TDK Semiconductor Corp.	URL:<http://www.tsc.tdk.com>
Texas Instruments Incorporated	URL:<http://www.ti.com>
Toshiba Corporation	URL:<http://www.toshiba.com>
Winbond Electronics	URL:<http://www.winbond.com>

B. Katalogový list obvodu PDIUSBD12

V této příloze je uveden zkrácený katalogový list obvodu PDIUSBD12 firmy Philips. Celý list se nachází na přiloženém CD. Nejnovější verzi listu lze nalézt na internetových stránkách firmy PHILIPS URL:<http://www.philips.com>.

1. Description

The PDIUSBD12 is a cost and feature optimized USB device. It is normally used in microcontroller based systems and communicates with the system microcontroller over the high-speed general purpose parallel interface. It also supports local DMA transfer.

This modular approach to implementing a USB interface allows the designer to choose the optimum system microcontroller from the available wide variety. This flexibility cuts down the development time, risks, and costs by allowing the use of the existing architecture and minimize firmware investments. This results in the fastest way to develop the most cost effective USB peripheral solution.

The PDIUSBD12 fully conforms to the *USB specification Rev. 2.0 (basic speed)*. It is also designed to be compliant with most device class specifications: Imaging Class, Mass Storage Devices, Communication Devices, Printing Devices, and Human Interface Devices. As such, the PDIUSBD12 is ideally suited for many peripherals like Printer, Scanner, External Mass Storage (Zip Drive), Digital Still Camera, etc. It offers an immediate cost reduction for applications that currently use SCSI implementations.

The PDIUSBD12 low suspend power consumption along with the LazyClock output allows for easy implementation of equipment that is compliant to the ACPI™, OnNOW™, and USB power management requirements. The low operating power allows the implementation of bus powered peripherals.

In addition, it also incorporates features like SoftConnect™, GoodLink™, programmable clock output, low frequency crystal oscillator, and integration of termination resistors. All of these features contribute to significant cost savings in the system implementation and at the same time ease the implementation of advanced USB functionality into the peripherals.

2. Features

- Complies with the *Universal Serial Bus specification Rev. 2.0 (basic speed)*
- High performance USB interface device with integrated SIE, FIFO memory, transceiver and voltage regulator
- Compliant with most Device Class specifications
- High-speed (2 Mbytes/s) parallel interface to any external microcontroller or microprocessor
- Fully autonomous DMA operation
- Integrated 320 bytes of multi-configuration FIFO memory
- Double buffering scheme for main endpoint increases throughput and eases real-time data transfer
- Data transfer rates: 1 Mbytes/s achievable in Bulk mode, 1 Mbits/s achievable in Isochronous mode
- Bus-powered capability with very good EMI performance
- Controllable LazyClock output during suspend
- Software controllable connection to the USB bus (SoftConnect™)
- Good USB connection indicator that blinks with traffic (GoodLink™)
- Programmable clock frequency output
- Complies with the ACPI, OnNOW and USB power management requirements
- Internal Power-on reset and low-voltage reset circuit
- Available in SO28 and TSSOP28 pin packages
- Full industrial grade operation from –40 to +85 °C
- Higher than 8 kV in-circuit ESD protection lowers cost of extra components
- Full-scan design with high fault coverage (>99%) ensures high quality
- Operation with dual voltages:
  3.3 ±0.3 V or extended 5 V supply range of 4.0 to 5.5 V
- Multiple interrupt modes to facilitate both bulk and isochronous transfers.

3. Pinning information

3.1 Pinning

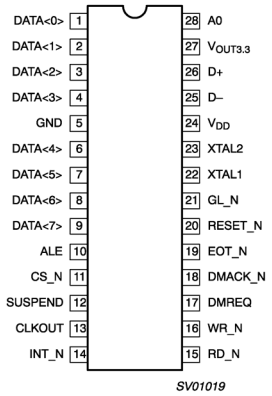
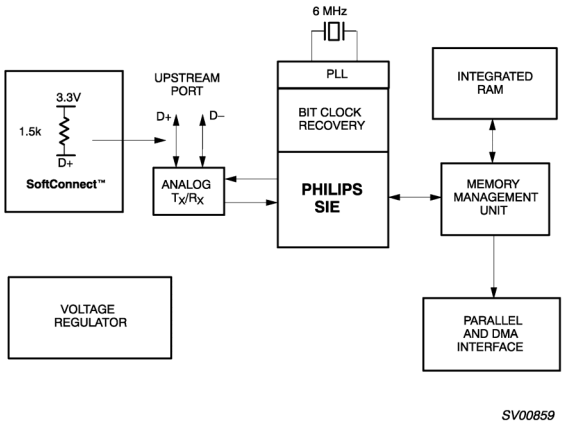


Fig 1. Pin configuration.

5. Block diagram



This is a conceptual block diagram and does not include each individual signal.

Fig 2. Block diagram.

6. Functional description

6.1 Analog transceiver

The integrated transceiver interfaces directly to the USB cables through termination resistors.

6.2 Voltage regulator

A 3.3 V regulator is integrated on-chip to supply the analog transceiver. This voltage is also provided as an output to connect to the external 1.5 kΩ pull-up resistor. Alternatively, the PDIUSBD12 provides SoftConnect technology with an integrated 1.5 kΩ pull-up resistor.

6.3 PLL

A 6 MHz to 48 MHz clock multiplier PLL (Phase-Locked Loop) is integrated on-chip. This allows for the use of a low-cost 6 MHz crystal. EMI is also minimized due to the lower frequency crystal. No external components are needed for the operation of the PLL.

6.4 Bit clock recovery

The bit clock recovery circuit recovers the clock from the incoming USB data stream using 4× oversampling principle. It is able to track jitter and frequency drift specified by the USB specification.

6.5 Philips Serial Interface Engine (PSIE)

The Philips SIE implements the full USB protocol layer. It is completely hardwired for speed and needs no firmware intervention. The functions of this block include: synchronization pattern recognition, parallel/serial conversion, bit stuffing/de-stuffing, CRC checking/generation, PID verification/generation, address recognition, and handshake evaluation/generation.

6.6 SoftConnect

The connection to the USB is accomplished by bringing D+ (for high-speed USB device) HIGH through a 1.5 kΩ pull-up resistor. In the PDIUSBD12, the 1.5 kΩ pull-up resistor is integrated on-chip and is not connected to V_{CC} by default. The connection is established through a command sent by the external/system microcontroller. This allows the system microcontroller to complete its initialization sequence before deciding to establish connection to the USB. Re-initialization of the USB bus connection can also be performed without requiring to pull out the cable.

The PDIUSBD12 will check for USB V_{BUS} availability before the connection can be established. V_{BUS} sensing is provided through pin EOT_N. See Section 3.2 “Pin description” for details. Sharing of V_{BUS} sensing and EOT_N can be easily accomplished by using V_{BUS} voltage as the pull-up voltage for the normally open-drain output of the DMA controller pin.

It should be noted that the tolerance of the internal resistors is higher (25%) than that specified by the USB specification (5%). However, the overall V_{SE} voltage specification for the connection can still be met with good margin. The decision to make sure of this feature lies with the users.

6.7 GoodLink

Good USB connection indication is provided through GoodLink technology. During enumeration, the LED indicator will blink ON momentarily corresponding to the enumeration traffic. When the PDIUSBD12 is successfully enumerated and configured, the LED indicator will be permanently ON. Subsequent successful (with acknowledgement) transfer to and from the PDIUSBD12 will blink OFF the LED. During suspend, the LED will be OFF.

This feature provides a user-friendly indicator on the status of the USB device, the connected hub and the USB traffic. It is a useful field diagnostics tool to isolate faulty equipment. This feature helps lower field support and hotline costs.

6.8 Memory Management Unit (MMU) and Integrated RAM

The MMU and the integrated RAM buffer the difference in speed between USB, running in bursts of 12 Mbits/s and the parallel interface to the microcontroller. This allows the microcontroller to read and write USB packets at its own speed.

6.9 Parallel and DMA Interface

A generic parallel interface is defined for ease-of-use, speed, and allows direct interfacing to major microcontrollers. To a microcontroller, the PDIUSBD12 appears as a memory device with 8-bit data bus and 1 address bit (occupying 2 locations). The PDIUSBD12 supports both multiplexed and non-multiplexed address and data bus. The PDIUSBD12 also supports DMA (Direct Memory Access) transfer which allows the main endpoint (endpoint 2) to directly transfer to and from the local shared memory. Both single-cycle and burst mode DMA transfers are supported.

6.10 Example of parallel interface to an 80C51 microcontroller

In the example shown in Figure 3, the ALE pin is permanently tied LOW to signify a separate address and data bus configuration. The A0 pin of the PDIUSBD12 connects to any of the 80C51 I/O ports. This port controls the command or data phase to the PDIUSBD12. The multiplexed address and data bus of the 80C51 can now be connected directly to the data bus of the PDIUSBD12. The address phase will be ignored by the PDIUSBD12. The clock input signal of the 80C51 (pin XTAL1) can be provided by output CLKOUT of the PDIUSBD12.

8. Endpoint description

The PDIUSBD12 endpoints are sufficiently generic to be used by various device classes ranging from Imaging, Printer, Mass Storage and Communication device classes. The PDIUSBD12 endpoints can be configured for 4 operating modes depending on the Set mode command. The 4 modes are:

Mode 0 Non-isochronous transfer (Non-ISO mode)

Mode 1 Isochronous output only transfer (ISO-OUT mode)

Mode 2 Isochronous input only transfer (ISO-IN mode)

Mode 3 Isochronous input and output transfer (ISO-I/O mode).

Endpoint number	Endpoint index	Transfer type	Direction[1]	Max. Packet size (bytes)
Mode 0 (Non-ISO mode)				
0	0	Control	OUT	16
	1		IN	16
1	2	Generic[2]	OUT	16
	3		IN	16
2	4	Generic[2][3]	OUT	64 [4]
	5		IN	64 [4]
Mode 1 (ISO-OUT mode)				
0	0	Control	OUT	16
	1		IN	16
1	2	Generic[2]	OUT	16
	3		IN	16
2	4	Isochronous[3]	OUT	128 [4]
Mode 2 (ISO-IN mode)				
0	0	Control	OUT	16
	1		IN	16
1	2	Generic[2]	OUT	16
	3		IN	16
2	5	Isochronous[3]	IN	128 [4]
Mode 3 (ISO-I/O mode)				
0	0	Control	OUT	16
	1		IN	16
1	2	Generic[2]	OUT	16
	3		IN	16
2	4	Isochronous[3]	OUT	64 [4]
	5		IN	64 [4]

[1] IN: input for the USB host; OUT: output from the USB host.

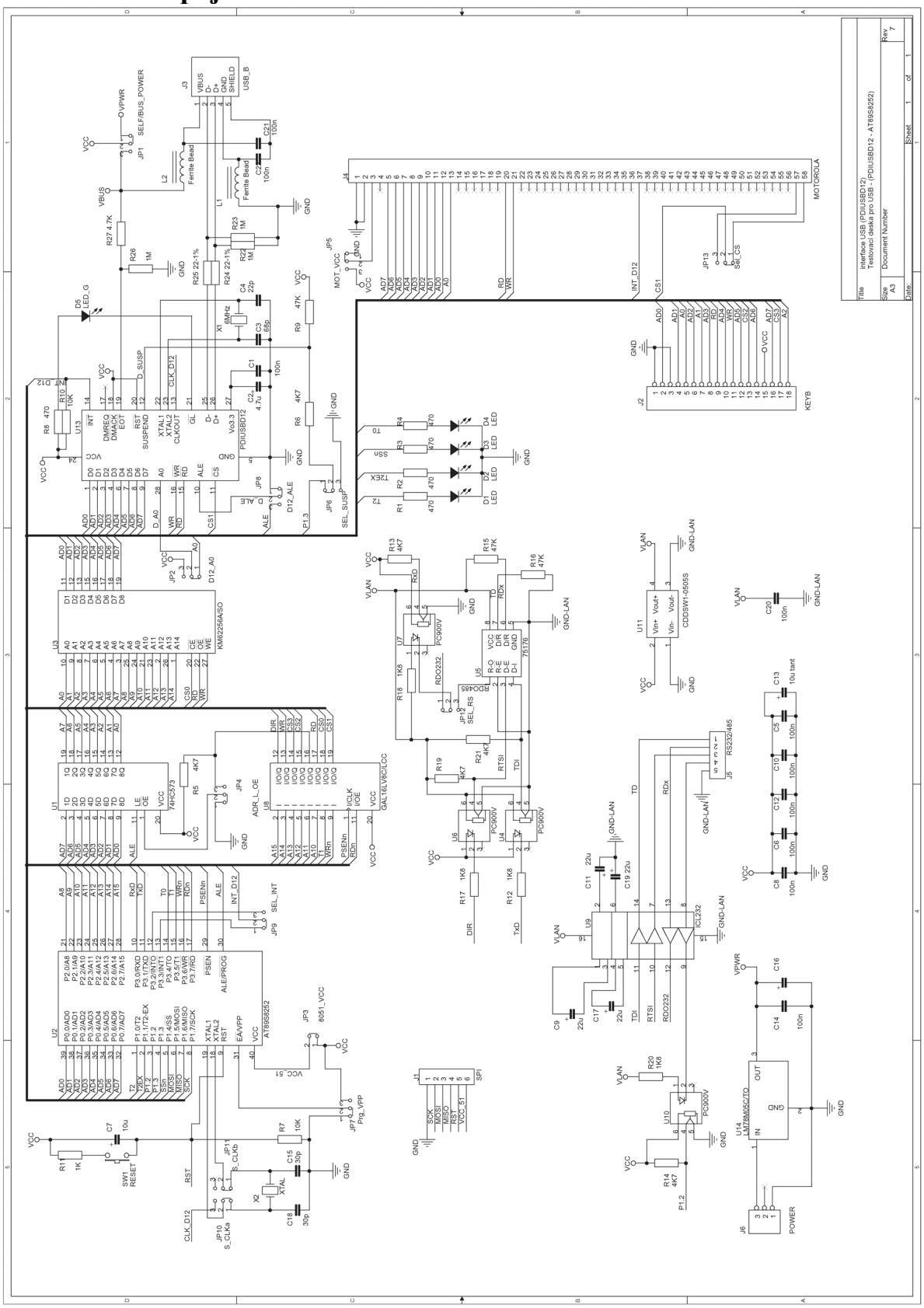
[2] Generic endpoints can be used either as Bulk or Interrupt endpoint.

[3] The main endpoint (endpoint number 2) is double-buffered to ease synchronization with the real-time applications and to increase throughput. This endpoint supports DMA access.

[4] Denotes double buffering. The size shown is for a single buffer.

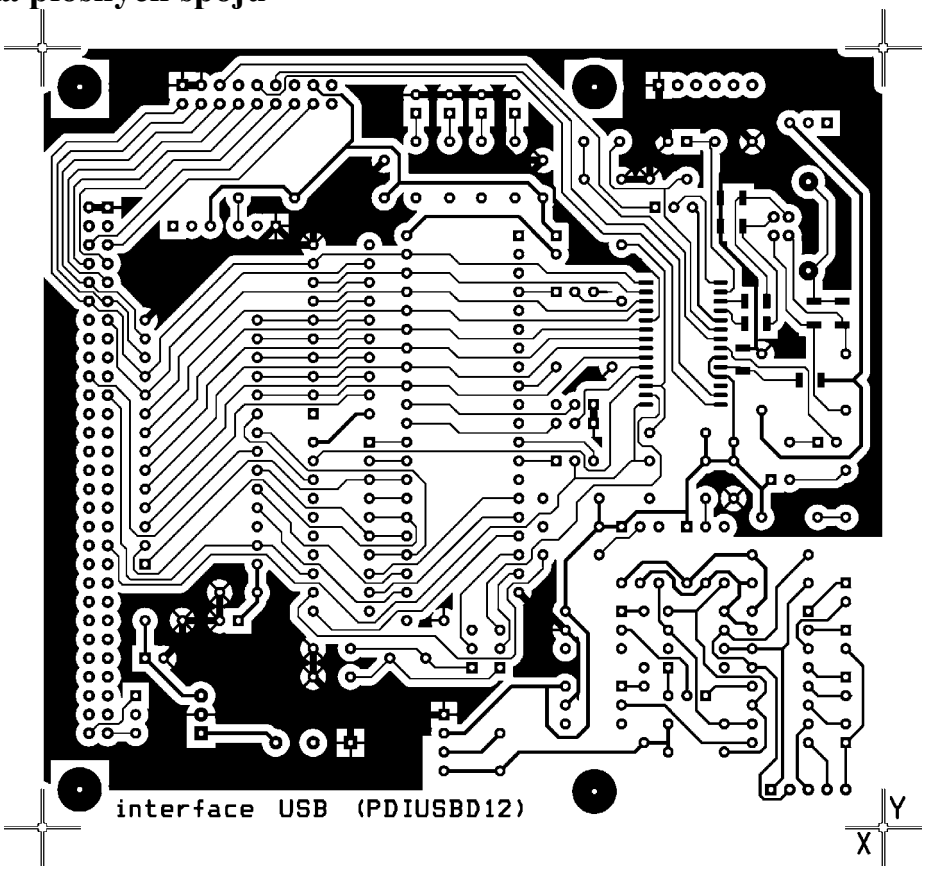
C. Technická dokumentace desky rozhraní USB - iD12

C.1 Schéma zapojení iD12

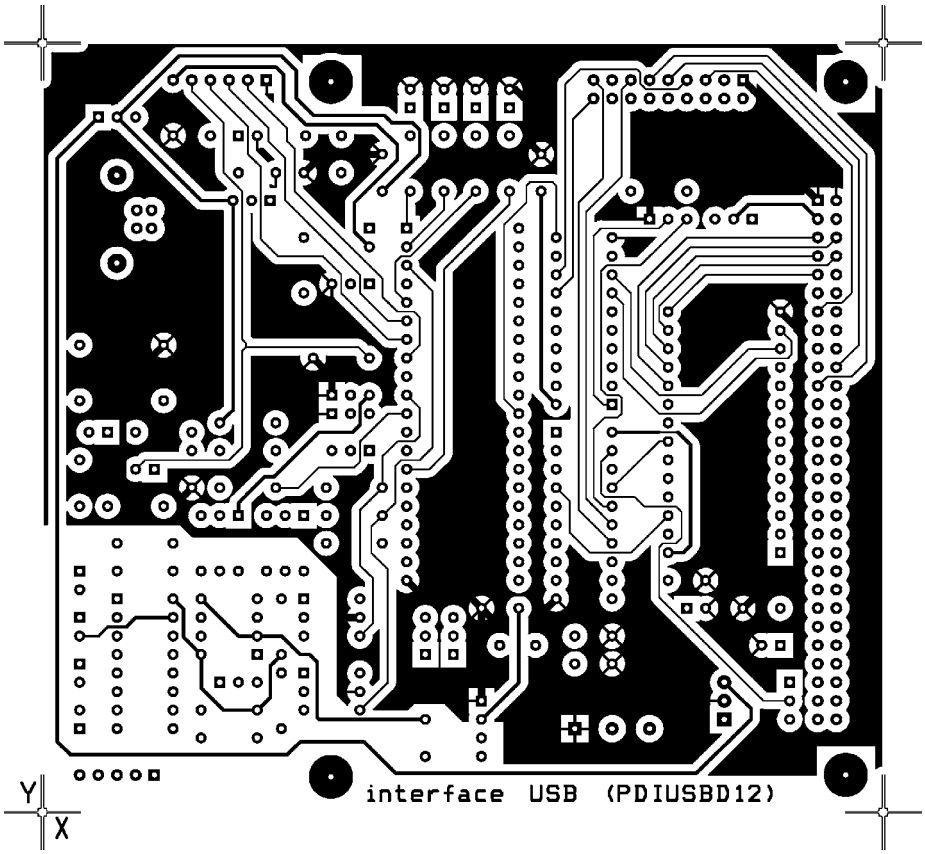


Obrázek C1 - Schéma zapojení desky rozhraní USB iD12.

C.2 Deska plošných spojů

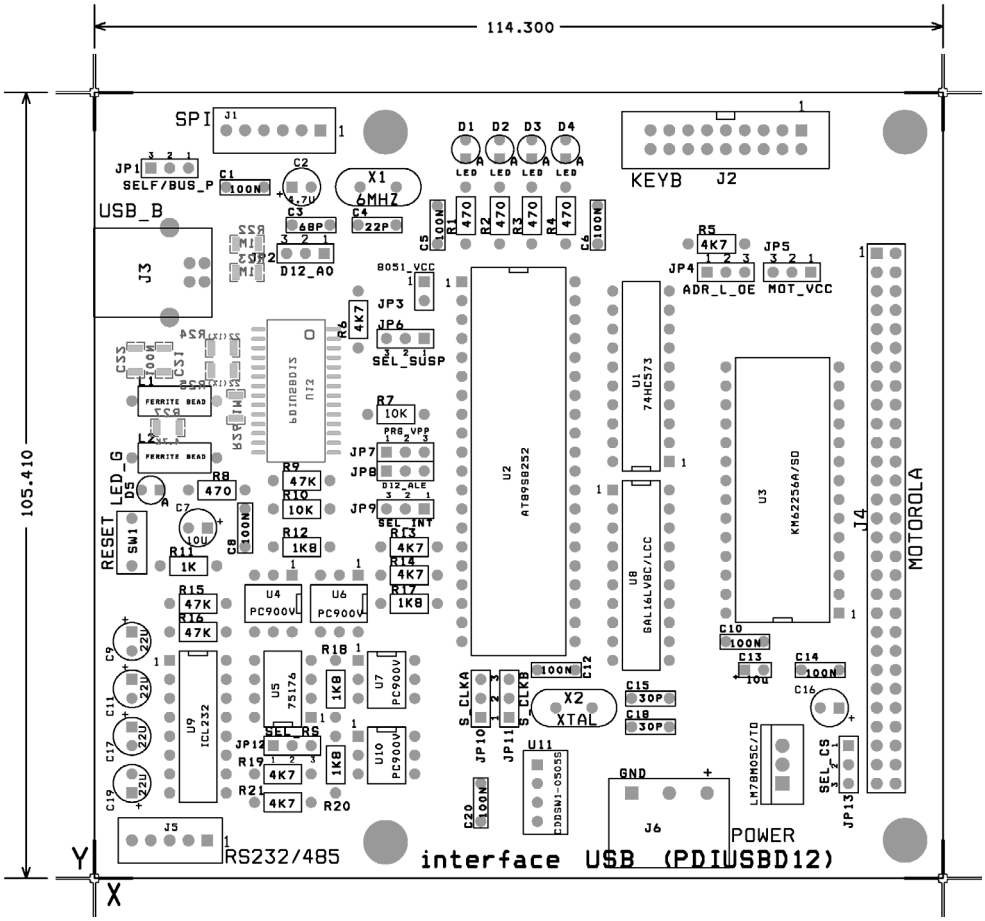


Obrázek C2 - Pohled ze strany spojů.



Obrázek C3 - Pohled ze strany součástek.

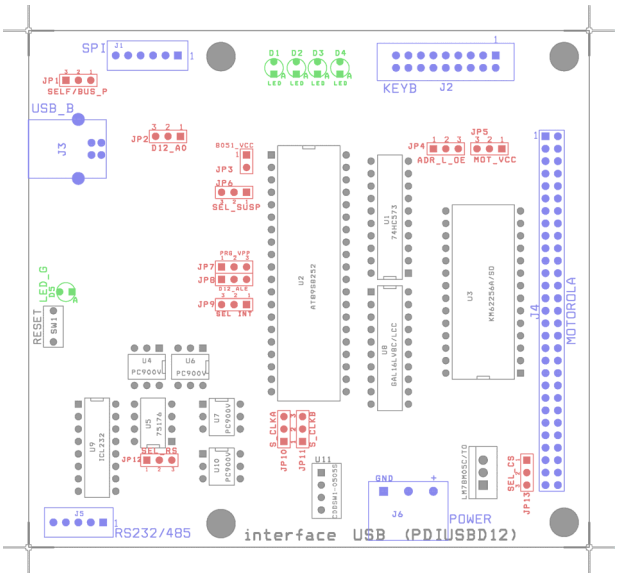
C.3 Rozložení součástek



Obrázek C4 - Rozložení součástek na desce iD12.

SMD součástky jsou na straně spojů, proto jsou na obrázku zrcadlově otočené a pro odlišení jsou nakresleny šedou barvou.

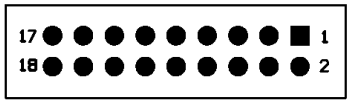
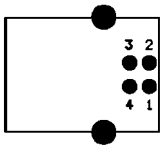
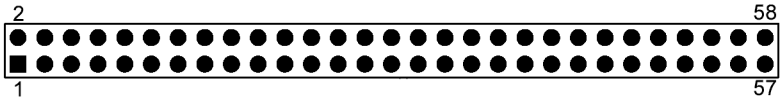
C.4 Rozmístění a popis konektorů a zkratovacích propojek



Obrázek C5 - Rozmístění konektorů, zkratovacích propojek a LED.

Na desce je 6 konektorů rozmístěných po okrajích desky, na obrázku C5 zobrazených modře a popsaných v tabulce C1. Třináct zkratovacích propojek je zobrazeno červeně a popsáno v tabulce C2. Na desce je také pět LED na obrázku nakreslených zelenou barvou.

Tabulka C1 - Popis konektorů na desce iD12.

J1 - SPI 	Připojení CPU ke sběrnici SPI. 1 – GND 2 – SCK 3 – MOSI 4 – MISO 5 – RST 6 – V_{CC}
J2 – KEYB  	Připojení dalších periferií k desce (klávesnice, display,…). 1 – GND 2 – D0 3 – GND 4 – D1 5 – A0 6 – D2 7 – A1 8 – D3 9 – RD 10 – D4 11 – WR 12 – D5 13 – CS2 14 – D6 15 – V_{CC} 16 – D7 17 – CS3 18 – A2
J3 - USB  	Připojení desky ke sběrnici USB. 1 – V_{BUS} 2 – D- 3 – D+ 4 – GND_{BUS}
J4 - MOTOROLA  	Připojení rozhraní USB k jiným desce MO_CPU s procesorem MOTOROLA. 1 – GND 2 – GND 3 – V_{CC} 5 – D7 6 – D6 7 – D5 8 – D4 9 – D3 10 – D2 11 – D1 12 – D0 13 – A0 20 – RD 21 – WR 37 – INT 58 – CS

J5 – RS232/RS485	Společný konektor pro sériové rozhraní RS232 a RS 485. 1 – TxD 2 – RxD 3 – RTS 4 – CTS 5 – GND
J6 - POWER	Napájení desky. 1 – GND 2 – nezapojen 3 – V_{CC}

Tabulka C2 - Popis zkratovacích propojek na desce iD12.

JP1 - SELF/BUS POWERED	Výběr napájení desky – ze sběrnice nebo externí. Deska je napájena z vlastního zdroje. Deska je napájena po sběrnici USB.
JP2 - D12_A0	Výběr typu sběrnice pro PDIUSBD12. Oddělená datová sběrnice. A0 = 1 na datové sběrnici je instrukce A0 = 0 na datové sběrnici jsou data (Propojka JP8 musí být v pozici 1-2) Multiplexovaná sběrnice. Sudé adresy znamenají data (A0=0) a liché znamenají instrukce (A0=1) (Propojka JP8 musí být v pozici 2-3)
JP3 - V_{CC}_8051	Připojení CPU a SPI konektoru ke společnému napájení. Napájení procesoru je připojeno ke společnému napájení V_{CC}. Napájení procesor je připojeno pouze k SPI konektoru.
JP4 - ADR_L_OE	Řízení povolení výstupu záchytného registru adresy. Záchytný registr je aktivně připojen ke sběrnici. /OE = 0. Záchytný registr nemá výstupy připojen ke sběrnici (případně je připojení řízeno signálem z obvodu GAL).

JP5 - MOT_VPWR	Spojení napájení s napájením z konektoru MOTOROLA. Napájení není připojeno ke konektoru MOTOROLA. Napájení je připojeno ke konektoru MOTOROLA.
JP6 - SEL_SUSP	Výběr ovládání signálu SUSPEND obvodu PDIUSBD12. Standardní nastavení - SUSPEND propojen s P1.3. Připojení SUSPEND k logické 0. Připojení SUSPEND k logické 1.
JP7 – EA/VPP	Povolení spouštění programu z externí paměti. Program je možno spouštět z externí paměti.
JP8 - D12_ALE	Výběr řízení ALE signálu u PDIUSBD12. Oddělená sběrnice ALE připojeno k log.0, instrukce a data jsou odlišeny signálem A0. (Propojka JP2 musí být v poloze 1-2). Multiplexovaná sběrnice ALE připojen k CPU, instrukce a data jsou odlišeny adresou. (Propojka JP2 musí být v poloze 2-3).
JP9 - SEL_INT	Výběr externího přerušení CPU pro PDIUSBD12. Požadavek od PDIUSBD12 je připojen k INT1 procesoru. Požadavek od PDIUSBD12 je připojen k INT0 procesoru.
JP10, JP11 - S_CLK	Výběr hodinového signálu pro CPU. Hodinový vstup CPU je připojen k vlastnímu krystalu. (Obě propojky musí být v poloze 1-2). Hodinový vstup CPU je připojen k výstupu D12. (Obě propojky musí být v poloze 2-3)
JP12 - SEL_RS	Výběr vstupu sériového rozhraní. RS232. RS485.
JP13 - SEL_CS	Výběr připojení signálu CS pro iD12 z desky MO_CPU. CS signál obvodu D12 připojen k signálu CS10 konektoru MOTOROLA. CS signál D12 připojen k CS9 konektoru MOTOROLA.