



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**

**ÚSTAV TELEKOMUNIKACÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

## **NÁVRH ZAŘÍZENÍ S ROZHRANÍM USB**

THE DESIGN OF A USB DEVICE

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. TOMÁŠ KLÍR**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. ANTONÍN HUDEC**

BRNO 2011



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Diplomová práce

magisterský navazující studijní obor  
**Telekomunikační a informační technika**

**Student:** Bc. Tomáš Klír

**ID:** 98493

**Ročník:** 2

**Akademický rok:** 2010/2011

## NÁZEV TÉMATU:

### Návrh zařízení s rozhraním USB

## POKYNY PRO VYPRACOVÁNÍ:

Prostudujte komunikační protokol USB. Zaměřte se na rozšiřující specifikaci USB OTG umožňující propojení bod-bod. Navrhněte vhodnou demonstrační aplikaci USB OTG a vyberte vhodné obvodové řešení a mikroprocesor nejlépe přímo s podporou USB OTG pro realizaci. Vytvořte demonstrační přípravek s vámi zvolenými komponenty pro procházení adresářového systému FAT32 na USB flash paměti.

## DOPORUČENÁ LITERATURA:

- [1] AXELSON, J. USB complete: everything you need to develop custom USB peripherals. New York: Lakeview Research, 2005. 3rd ed., 572 s. ISBN 978-1-931448-02-4
- [2] KAINKA, B. USB - měření, řízení a regulace pomocí sběrnice USB. 1. vydání. Praha: BEN - technická literatura, 2002. 256 s. ISBN 80-7300-073-3
- [3] PINKER, J. Mikroprocesory a mikropočítače. 1. vydání. Praha: BEN - technická literatura, 2004. 159 s. ISBN 80-7300-110-1
- [4] PLÍVA, Z. EAGLE prakticky. 1. vydání. Praha: BEN - technická literatura, 2007. 181 s. ISBN 978-80-7300-227-5

**Termín zadání:** 7.2.2011

**Termín odevzdání:** 26.5.2011

**Vedoucí práce:** Ing. Antonín Hudec

**prof. Ing. Kamil Vrba, CSc.**

*Předseda oborové rady*

## UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **Abstrakt**

V teoretické části této diplomové práce je rozebráno rozhraní USB 2.0, včetně USB OTG. Jsou zde uvedeny především vlastnosti, které je vhodné znát pro návrh zařízení využívající USB. Stručně je zde uvedena i struktura souborového systému FAT. Praktická část obsahuje návrh OTG zařízení a jeho realizaci včetně implementace knihoven pro komunikaci přes USB a práci se souborovým systémem FAT.

## **Klíčová slova**

USB, OTG, bod–bod, konfigurace sběrnice, souborový systém, FAT, PIC24F, EEPROM, LCD.

## **Abstract**

In theoretical part of this master thesis is analyzed USB 2.0 interface, including USB OTG. Mainly are described here properties which are good to know for USB device design. In brief is given here a structure of the FAT file system. Practical part contains design of OTG Device and it's realization including implementation of libraries for communication via USB and for working with FAT file system.

## **Key words**

USB, OTG, On–The–Go, bus enumeratuion, file systém, FAT, PIC24F, EEPROM, LCD.

KLÍR, T. *Návrh zařízení s rozhraním USB*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 64 s. Vedoucí diplomové práce Ing. Antonín Hudec.

## Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Návrh zařízení s rozhraním USB“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny uvedeny v seznamu literatury na konci práce.

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

V Brně dne .....

.....  
podpis autora

## **Poděkování**

Děkuji svému vedoucímu diplomové práce Ing. Antonínu Hudcovi za vhodnou metodickou a odbornou pomoc při zpracování této práce.

V Brně dne .....

.....  
podpis autora

# Obsah

Úvod	11
1 Od USB 1.0 k WUSB	12
1.1 USB 1.0	12
1.2 USB 2.0	12
1.3 USB 3.0	12
1.4 WUSB	13
2 Rozbor USB 2.0	14
2.1 Vrstvový model	14
2.1.1 Komunikující zařízení	14
2.2 Topologie sběrnice	16
2.3 Fyzická vrstva	17
2.4 Robustnost sběrnice	19
2.5 Konfigurace sběrnice	20
2.5.1 Inventarizace sběrnice	20
2.6 Koncové body	23
2.7 Roury	23
2.8 Struktura komunikačního toku	25
2.8.1 Rámce a mikrorámce	25
2.8.2 Transakce a pakety	25
2.9 Typy přenosů	26
2.9.1 Řídící přenosy	26
2.9.2 Izochronní přenosy	27
2.9.3 Přerušovací přenosy	27
2.9.4 Hromadné přenosy	28
2.10 Deskriptory	28
2.10.1 Deskriptor zařízení	29
2.10.2 Deskriptor nastavení	30
2.10.3 Deskriptor koncového bodu	31
2.11 Třídy USB zařízení	32
2.11.1 Třída Mass Storage	33
3 Souborový systém FAT	35
4 USB OTG zařízení	38
4.1 Návrh zařízení	38
4.1.1 Blokové schéma zařízení	38
4.1.2 Volba součástek	38
4.2 Oživování zařízení	42

4.2.1	Kodér 1:N .....	42
4.2.2	Základní nastavení mikrokontroléru PIC24FJ64GB002 .....	43
4.2.3	Ovládání displeje .....	43
4.2.4	Komunikace s EEPROM .....	44
4.2.5	Programování komunikace přes USB.....	46
4.2.6	Práce se souborovým systémem .....	49
4.2.7	Uživatelské rozhraní .....	50
5	Závěr .....	54
	Seznam použitých zdrojů .....	55
	Seznam použitých zkratk a symbolů .....	57
	Seznam příloh .....	58
A	USB OTG zařízení .....	59
A.1	Schéma zapojení.....	59
A.2	Seznam součástí.....	60
A.3	DPS .....	61
A.4	DPS – osazovací plán.....	62
A.5	Fotografie OTG zařízení.....	63
B	Obsah přiloženého CD .....	64



## Seznam obrázků

Obr. 2.1: Vrstvový model [1].....	15
Obr. 2.2: Topologie USB.....	17
Obr. 2.3: Nejčastěji používané USB konektory: a) standardní A, b) standardní B, c) mini A, d) mini B, e) mikro AB, f) mikro B.....	18
Obr. 2.4: USB kabel.....	18
Obr. 2.5: Kódování NRZI.....	19
Obr. 4.1: Blokové schéma zařízení.....	38
Obr. 4.2: Posílení napájení sběrnice.....	40
Obr. 4.3: Procházení souborů.....	51
Obr. 4.4: Nabídka menu.....	51
Obr. 4.5: Vytváření nového adresáře.....	52
Obr. 4.6: Atributy souboru.....	52
Obr. 4.7: Režim B–device.....	53

## Seznam tabulek

Tab. 2.1: Barevné značení vodičů.....	18
Tab. 2.2: Standardní typy deskriptorů .....	29
Tab. 2.3: Deskriptor zařízení .....	30
Tab. 2.4: Deskriptor nastavení.....	30
Tab. 2.5: Deskriptor koncového bodu .....	31
Tab. 2.6: Třídy USB zařízení.....	32
Tab. 2.7: Podtřídy Mass Storage.....	33
Tab. 2.8: Protokoly třídy Mass Storage .....	34
Tab. 2.9: Povinné SCSI příkazy.....	34
Tab. 3.1: Rozdělení paměťového média podle FAT12 a FAT16 .....	35
Tab. 3.2: Hodnoty FAT tabulky pro FAT32.....	36
Tab. 3.3: Informace uvedené v položce .....	36
Tab. 3.4: Položka LFN.....	37
Tab. 4.1: Kódování tlačítek.....	41
Tab. 4.2: Nastavení nejdůležitějších položek v Configuration Word.....	43
Tab. 4.3: Pozice na displeji a odpovídající adresy DDRAM.....	44
Tab. 4.4: Nastavení I/O pinů pro komunikaci s EEPROM.....	45
Tab. 4.5: Registr SPI1CON1 .....	46
Tab. 4.6: Funkční vrstvy hostitelského zařízení dle [11] a odpovídající knihovny .....	48
Tab. 4.7: Nastavení FatFS.....	49
Tab. 4.8: Asociace knihoven pro práci s FatFS .....	49

## Úvod

Tato diplomová je zaměřena na USB OTG. USB je sériová sběrnice soužící převážně pro propojení počítače a jeho periferií. Těmi však nemusí být jen externí datová úložiště a polohovací zařízení, ale na trhu je možné nalézt například i zvukové karty. Dnes je však toto rozhraní využíváno ve velké míře i pro připojení periferií k PDA, mobilním telefonům či televizorům.

Teoretická část této práce obsahuje informativní vývoj tohoto standardu od USB 1.0 až po, dosud více nerozšířené, bezdrátové USB. Hlavní náplní teoretické části této práce je rozbor USB 2.0. Jsou zde uvedeny a rozebrány pojmy, které je vhodné znát při návrhu a pozdější realizaci zařízení s tímto rozhraním. Jsou zde rozebrány odlišnosti komunikujících stran, tedy hostitelského zařízení a koncového zařízení. USB 2.0 je zde rozebráno jak po stránce fyzických vlastností sběrnice, tak z pohledu logické komunikace. Z pohledu logické komunikace jsou zde rozebrány vlastnosti logického propojení včetně struktury komunikace, podporovaných typů přenosů a třída MassStorage, kterou využívá navržené demonstrační zařízení. Jsou zde uvedeny vlastnosti a omezení možností zařízení využívajících propojení bod–bod (On–The–Go, OTG) a protokolů, které byly specifikovány společně s USB OTG. Konec teoretické části je věnován souborovým systémům FAT12, FAT16 a FAT32.

Praktická část této práce se zabývá vývojem demonstračního zařízení. Je zde uveden postup návrhu zařízení, včetně jeho realizace a případných problémů. To zahrnuje ovládání periferií (LCD displeje, EEPROM paměti a napájecího obvodu pro USB) a způsob použití knihoven pro komunikaci přes USB a práci se souborovým systémem FAT.

# 1 Od USB 1.0 k WUSB

USB (Universal Serial Bus) je standardizovaná sériová sběrnice pro připojení periferních zařízení (Device) k hostitelskému zařízení (Host, hostitel). Byla navržena pro umožnění připojení zařízení pomocí jednotného standardizovaného rozhraní bez nutnosti restartovat počítač (Plug-and-Play). Díky této sběrnici je možné připojit prakticky jakoukoliv periférii od fotoaparátu, či klávesnice až po externí zvukovou kartu, čímž se stala standardním rozhraním pro připojení periférií.

## 1.1 USB 1.0

Specifikace verze 1.0 byla představena roku 1996 [25]. Byla navržena pro úsporu konektorů na zadní straně počítače a zjednodušení softwarové konfigurace komunikace se zařízením. Podporovány byly rychlosti 1,5 Mb/s (Low-Speed) a 12 Mb/s (High-Speed). Pozdější verze USB 1.1 opravovala jen drobné chyby a byla s předchozí verzí prakticky téměř shodná.

## 1.2 USB 2.0

Nová generace USB 2.0 byla vydána roku 2000 [24]. Umožnila především rapidní nárůst přenosové rychlosti až na 480 Mb/s (tzv. Hi-Speed). Později byly přidány a vylepšeny mnohé vlastnosti jako např. nový konektor, rozšíření řízení správy napájení, přidány nové deskriptory a další. Jako nejzajímavější se jeví zavedení podpory OTG (On-The-Go) [3]. Zařízení s podporou OTG může vystupovat jak hostitelské, tak periferní zařízení.

## 1.3 USB 3.0

Tento standart zavedený roku 2008 [25] opět zvyšuje přenosovou rychlost a to tentokrát na 5 Gb/s. Toho je dosaženo mimo jiné zvýšením počtu datových párů vodičů (sběrnice je označována jako SuperSpeed [24]). Jeden pár pro každý směr přenosu diferenčního signálu. Jedná se o Dual Simplex což umožňuje, na rozdíl od předchozích verzí USB podporujících Half Duplex, komunikaci v obou směrech najednou. Změněno je také kódování a další vlastnosti ztěžující zpětnou kompatibilitu. Ta je ve výsledku řešena jako „zdvojení“ konektorů a kabelů. Pokud zařízení pracuje v super-speed režimu, datové vodiče využívané pro komunikaci pomocí USB 2.0 se nevyužívají. Další velká změna se týká napájení. Minimální dodávaný proud byl zvýšen na 150 mA, maximální na 900 mA [25] a byly přidány nové možnosti správy napájení.

## **1.4 WUSB**

WUSB (Wireless USB) [27] je bezdrátová technologie krátkého dosahu pracující na frekvencích 3,1–10,6 GHz. Na rozdíl od drátové verze nevyžaduje hub a je možné připojit až 127 zařízení bez jeho použití. Verze 1.0 podporuje rychlost až 480 Mb/s do vzdálenosti 3 m a 110 Mb/s do vzdálenosti 10 m. V září r. 2010 byla vydána specifikace verze 1.1 s přenosovou rychlostí 1 Gb/s. Specifikace zahrnuje i zařízení dvojí role (Dual-Role Devices, DRDs), což je obdoba OTG.

## 2 Rozbor USB 2.0

Tato kapitola popisuje všechny potřebné vlastnosti USB pro jeho pochopení a návrh zařízení s tímto rozhraním. Bude se zde hovořit o zatím nejrozšířenějším USB 2.0.

### 2.1 Vrstvový model

Vrstvový model [1] rozděluje zařízení s USB rozhraním na několik částí, vykonávajících určitou specifickou část komunikace. Jeho zobrazení je na obrázku 2.1.

**Fyzická** vrstva zajišťuje skutečnou komunikaci mezi dvěma zařízeními. Jsou zde definovány elektrické a mechanické vlastnosti potřebné pro vlastní přenos datového toku. Na této vrstvě je také prováděno CRC.

Vrstva **zařízení** řídí komunikaci mezi zařízeními. Je zde také zahrnuta komunikace pomocí obecných USB operací.

Na **funkční** vrstvě probíhá komunikace mezi klientským SW hostitelského zařízení a funkcí poskytované periferním zařízením.

#### 2.1.1 Komunikující zařízení

Z obrázku 2.1 je vidět, že zařízení s rozhraním USB se dělí na dva základní typy a to na hostitelské zařízení (Host, hostitel) a periferní zařízení (Device).

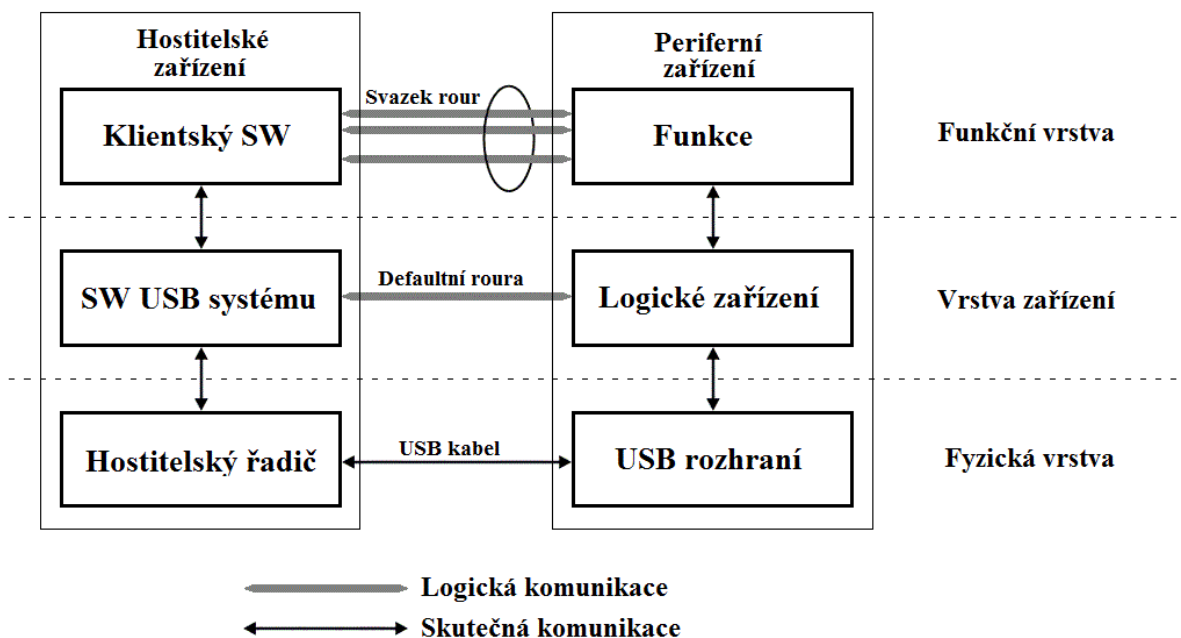
Hostitelské zařízení je hierarchicky nejvyšší bod v topologii. Jedná se o zařízení řídící komunikaci a využívající možné funkce koncového. Jeho základní části jsou:

- Klientský SW – tento SW je nejčastěji součástí operačního systému nebo je dodáván spolu s periferním zařízením; je to SW využívající funkce periferního zařízení.
- SW USB systému – jedná se o SW určený pro řízení sběrnice; je nezávislý na klientském SW a připojených zařízeních; je součástí jádra systému.
- Hostitelský radič – obsahuje HW a SW potřebný pro připojení periferních zařízení.

Protější stranou komunikace je periferní zařízení poskytující dodatečné funkce. To se dále rozděluje na koncová zařízení (Functions) a huby. Součástí periferního zařízení jsou:

- Funkce – jedná se o dodatečné funkce implementované sadou rozhraní periferního zařízení.

- Logické zařízení – je dáno sadou koncových bodů implementujících vlastní rozhraní.
- USB rozhraní – je obdoba hostitelského řadiče, ale na straně periferního zařízení.



Obr. 2.1: Vrstvový model [1].

Dodatečně bylo do specifikace přidáno zařízení OTG (dnes ještě občas označované podle staré terminologie Dual Role). Jeho základní vlastnosti zahrnují:

- Omezenou schopnost vystupovat jako hostitelské zařízení.
- Téměř plnohodnotná schopnost pracovat jako koncové zařízení.
- Dynamické přepnutí funkce hostitelského a koncového zařízení.
- Malý konektor vhodný pro přenosná zařízení.
- Nízkou spotřebu pro vyšší životnost při běhu na baterie.

V roli koncového zařízení je nemožné využít jen režim low-speed. Jako hostitelské zařízení se však oproti plnohodnotnému hostiteli liší v:

- Volitelné možnosti komunikace v low a hi-speed režimu (full-speed režim musí být podporován).
- Volitelné podpoře hubů.
- Nutná podpora SRP a HNP (viz dále).

- Minimální dostupný proud na jeden port je 8 mA.
- Možnost odpojení napájení sběrnice.
- Možnost komunikace pouze se zařízeními v PTL (Peripheral Target List).

Jedná se o tzv. Embedded Host. PTL je seznam podporovaných zařízení, kde jsou zařízení identifikována podle kódu výrobce (VID) a produktu (PID) nebo podle třídy (Class), podtřídy (Subclass) a protokolu viz kap 2.10.

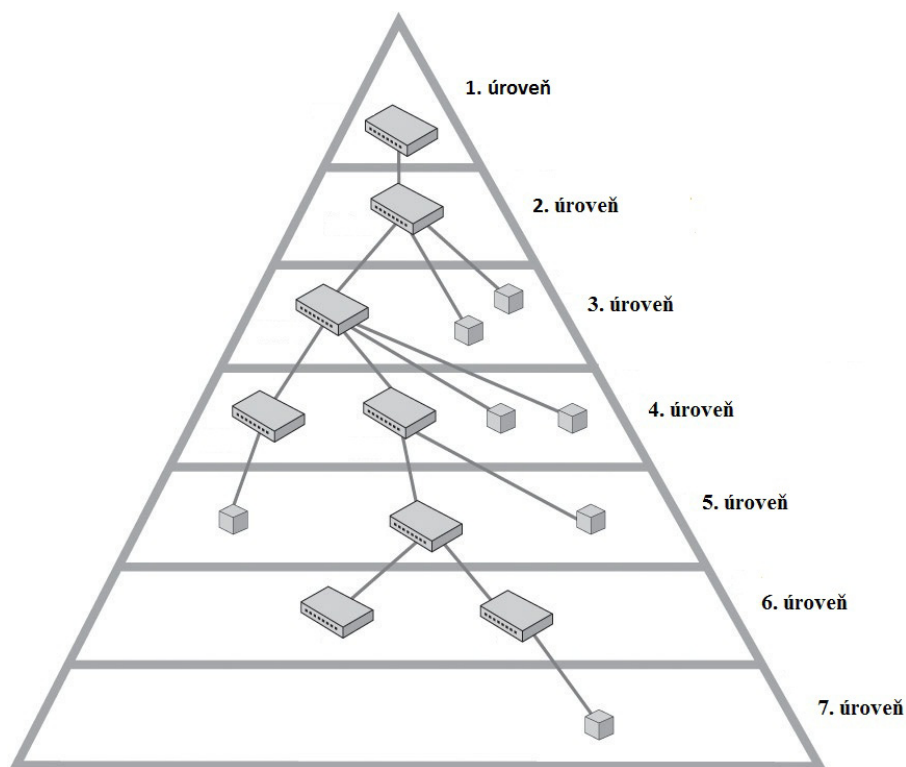
Spolu s režimem OTG byly specifikovány tři nové protokoly [25] – ADP (Attach Detection Protocol), SRP (Session Request Protocol) a HNP (Host Negotiation Protocol). ADP umožňuje detekci nově připojeného zařízení při odpojení napájení sběrnice pomocí měření kapacity na portu. SRP umožňuje řízení napájení sběrnice podle potřeb obou stran (hostitele i koncového zařízení). HNP je využito při propojení dvou OTG zařízení a umožňuje jim si vzájemně prohodit role hostitele a koncového zařízení.

## 2.2 Topologie sběrnice

USB sběrnice využívá stromovou fyzickou topologii, přičemž uprostřed se nachází kořenový rozbočovač (Root Hub), který je součástí hostitelského zařízení. K němu je možné připojit periferní zařízení (Device). To může být koncové zařízení či další hub, přičemž maximální počet úrovní je 7 [7]. To je dáno maximálním možným zpožděním. Topologii ukazuje obrázek 2.2, kde 1. úroveň obsahuje hostitelské zařízení spolu s kořenovým rozbočovačem. Poslední 7. úroveň už poté může obsahovat jen koncová zařízení.

V logické topologii jsou huby neviditelné a jedná se o hvězdicovou topologii. Vypadá to tedy, jako by byly všechny periferní zařízení připojené přímo k hostitelskému zařízení.





Obr. 2.2: Topologie USB

## 2.3 Fyzická vrstva

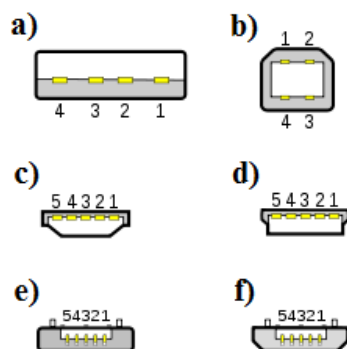
Pro propojení zařízení pomocí USB se využívá celá řada konektorů (obr. 2.3). Jejich základní dělení je na typ A a typ B [25]. Typ A využívá na straně hostitelského zařízení pro směr ke koncovému zařízení (Downstream–port). Naopak typ B se využívá na straně koncového zařízení pro směr k hostitelskému zařízení (Upstream–port), např. u tiskárny. Tím se mělo původně zabránit vytvoření elektrické smyčky na hubu. Dnes však již řada výrobců nabízí kabely se zástrčkami typu A na obou koncích. Dále jsou dostupné různé rozměry konektorů – standardní, mini a mikro. Velikosti mikro a mini byly zavedeny pro přenosná zařízení. U těchto se však může jednat o zařízení podporující OTG. Proto byl zaveden další typ konektoru – AB [26], který je pouze ve velikosti mini a mikro. Tento typ konektoru pro rozlišení, zda se jedná o typ A či B, obsahuje 5. pin (ID pin). Pokud je na připojeném kabelu tento pin uzemněn, OTG zařízení je v režimu hostitelského zařízení. V opačném případě je tento pin nezapojen a OTG zařízení je v režimu periferního zařízení. Podle konce kabelu, ke kterému je zařízení připojeno se označuje jako zařízení A (A–Device, v režimu host), či zařízení B (B–Device, v režimu device) [3]. Dále jsou specifikovány další tři stavy rozlišené připojenou zátěží:

- Je připojena samotná nabíječka, či se zařízením A nevyžadující napájení z USB konektoru; OTG zařízení se může nabíjet a využívat SRP, ale ne se připojit.
- Je připojena nabíječka a zařízení A vyžadující napájení z USB konektoru; USB zařízení je možné nabíjet a připojit, ale neumožňuje využívat SRP.
- Je připojena nabíječka a zařízení B; OTG zařízení je možné nabíjet a může se přepnout do režimu host.

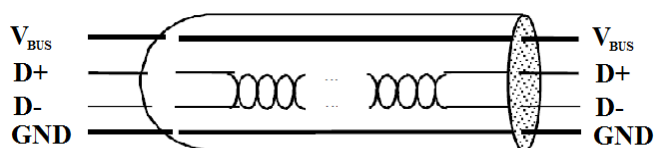
Kabel je čtyřvodičový stíněný (obr. 2.4). Jeden pár je použit pro napájení a druhý pro přenos dat. Datový pár navíc musí být kroucený pro full-speed a hi-speed. Pro low-speed je kroucený pár pouze doporučený. Maximální délka kabelu je 5 metrů [26].

Tab. 2.1: Barevné značení vodičů

Funkce vodiče	Barva vodiče	Číslo pinu
$V_{bus}$	Červená	1
D-	Bílá	2
D+	Zelená	3
GND	Černá	4



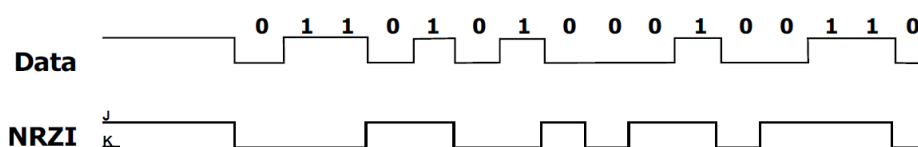
Obr. 2.3: Nejčastěji používané USB konektory: a) standardní A, b) standardní B, c) mini A, d) mini B, e) mikro AB, f) mikro B



Obr. 2.4: USB kabel

Napájecí vodiče poskytují napětí 5 V a proud minimálně 100 mA [24]. V případě potřeby je možné poskytnout až 500 mA [24].

Signál se přenáší po datových vodičích v diferenční podobě. Stav J odpovídá výchozímu stavu. Opakem je stav K. Kódování se používá NRZI (Non-Return-To-Zero-Inverted, obr. 2.5). U tohoto kódování vyjadřuje log. „0“ změnu stavu a log. „1“ setrvání v původním stavu. Při dlouhé posloupnosti jedniček by však hrozila ztráta synchronizace a proto se po každých šesti jedničkách vkládá nula.



Obr. 2.5: Kódování NRZI

Fyzická vrstva také určuje, zda připojené zařízení bude komunikovat v režimu low-speed či full-speed. To je určeno připojením 3,3 V na některou z datových linek přes pull-up rezistor (rezistor zvyšující napětí) [24]. Připojením tohoto rezistoru na linku D+ se nastavuje full-speed. Low-speed je naopak nastaven připojením rezistoru na D-. Rychlost hi-speed se nejprve inicializuje jako full-speed a poté je během resetování zařízení provedena chirp sekvence [3]. Jedná se tedy o SW nastavení. Tento pull-up rezistor také slouží pro detekci nově připojeného zařízení. Tímto rezistorem je také určeno, že se jedná o periferní zařízení. Hostitelské zařízení je pak poznáno podle připojení obou datových linek přes pull-down rezistory (rezistory snižující napětí) k zemi.

## 2.4 Robustnost sběrnice

Robustnost USB zajišťují tyto vlastnosti:

- Signálová integrita zajištěná použitím diferenčních vysílačů, přijímačů a stínění.
- Cyklický redundantní součet (CRC, Cyclic redundancy check) všech přenášených informací.
- Detekce připojování a odpojování zařízení a konfigurace zdrojů na systémové úrovni.
- Samoopravné prvky v protokolu využívající časovače pro ztracené či poškozené pakety.
- Řízení toku datových proudů.

- Rozdělení na datové a řídicí roury pro zajištění nezávislosti a případných interakcí zařízení.

Cyklický redundantní součet je schopný opravit dvě chyby. Odeslání poškozeného či ztraceného paketu je možné opakovat až třikrát.

## 2.5 Konfigurace sběrnice

Na sběrnici je možné připojit zařízení v jakoukoliv dobu. Hostitelské zařízení, včetně jeho softwaru, tedy musí být schopno reagovat na změny v topologii. Huby obsahují tzv. status bity [24], které se nastavují v závislosti na připojení či odpojení zařízení. Hostitelské zařízení pravidelně kontroluje tyto status bity a tím se dozví o připojení či odpojení zařízení.

V případě zjištění nově připojeného zařízení povolí hostitelské zařízení daný port na hubu a komunikuje se zařízením skrze řídicí rouru (Default control pipe) [1] [3]. Zařízení poslouchá na výchozí adrese. Následně dojde k přiřazení unikátní adresy v rámci topologie, zjištění typu zařízení a vytvoření řídicí roury (viz kapitola 2.7). V případě, že se jedná o nově připojený hub, se tento proces opakuje postupně pro každé zařízení připojené na tento hub. V případě připojení koncového zařízení je informován hostitelský software (ovladač) daného zařízení.

Po odpojení zařízení ze sběrnice zakáže hub daný port a informuje hostitelské zařízení o odpojení zařízení, kdy tuto zprávu musí zpracovat ovladač. Opět pokud se jedná o hub, ovladače všech zařízení musí správně zpracovat odpojení daných zařízení.

Tyto procesy (připojení a odpojení zařízení) se jednotně nazývají inventarizace sběrnice (Bus Enumeration) [24]. Pro sledování dynamických změn v topologii sběrnice je zapotřebí jí průběžně provádět.

### 2.5.1 Inventarizace sběrnice

Je definováno celkem šest možných stavů koncového zařízení [3]: *Napájeno* (Powered), *ve výchozím stavu* (Default), *s přiřazenou adresou* (Address), *nastaveno* (Configured), *připojeno* (Attached) a *pozastaveno* (Suspend). Každý z těchto stavů má specifické vlastnosti a chování. Během inventarizace sběrnice se zařízení vyskytne ve všech stavech kromě stavů *připojeno* a *pozastaveno*. Následující pořadí úkonů je nejběžnější, ale není však dané specifikací. Zařízení tedy musí být schopno reagovat na kterýkoliv řídicí přenos v jakémkoliv čase.

- 1 Připojení zařízení k USB portu** nebo také systém povolí napájení portu, kde je již zařízení připojeno. Nyní je zařízení ve stavu *napájeno* a port ke kterému je připojeno není u hubu povoleno.

- 2 **Hub detekuje připojené zařízení.** Detekce probíhá na základě zvýšení napětí na jedné z datových linek přes pull-up rezistor na koncovém zařízení. V této době hub stále nezahajuje žádnou komunikaci.
- 3 **Host se dozví o novém zařízení.** Hub použije přerušovací koncový bod k oznámení nově připojeného zařízení. Informace v tomto oznámení je pouze, jaký typ zařízení byl připojen (další hub či koncové zařízení). Následně host požádá hub o další informace. Tím se host dozví, zda je zařízení nově připojeno.
- 4 **Určení rychlosti připojeného zařízení.** Ještě před resetováním zařízení hub zjistí jakou rychlostí je zařízení schopno komunikovat (low či full-speed). To platí pro verzi USB 2.0 (ve verzi 1.x je zařízení nejdříve resetováno). Je to z důvodů, aby hub věděl, zda má po resetu detekovat high-speed zařízení. Tato informace je opět odeslána hostovi.
- 5 **Hub resetuje zařízení.** Jakmile se host dozví o novém zařízení, pošle žádost o jeho resetování. To probíhá připojením obou datových linek přes pull-down rezistory na zem po dobu minimálně 10 ms. Resetováno je pouze nově připojené zařízení.
- 6 **Zjištění zda zařízení podporuje high-speed.** Během detekce podpory high-speed komunikace se využívá dvou stavů. Chirp K, kdy je řízena D- linka, a chirp J, kdy je naopak řízena D+ linka. Během resetu vyšle zařízení podporují high-speed signál chirp K. Pokud hub a host podporují tuto rychlost, odpoví kombinací KJKJKJ. Pokud zařízení detekuje tuto posloupnost, odpojí svůj pull-up rezistor a dále komunikuje high-speed rychlostí.
- 7 **Vytvoření signálové cesty mezi zařízením a sběrnicí.** Host pravidelně kontroluje, zdali je zařízení stále v resetu. Jakmile hub uvolní reset, zařízení se nachází ve *výchozím stavu* a je schopné reagovat na řídicí přenosy skrze koncový bod nula. Zařízení nyní naslouchá na výchozí adrese 00h. Tato adres je v rámci sběrnice jedinečná, jelikož host nemůže konfigurovat více zařízení najednou. V tomto stavu nesmí být odběr proudu zařízením větší než 100 mA.
- 8 **Zjištění maximální velikosti paketu výchozí řídicí roury.** Zařízení odesílá na žádost hosta deskriptor zařízení (Device Descriptor, viz kap. 2.10), ve kterém je, mimo jiné, maximální velikost paketu podporovaného koncovým bodem 0. Po obdržení osmého bytu (ve kterém je požadovaná maximální velikost paketu) je možné opětovné resetování zařízení. Tím se docílí, že zařízení bude určitě ve *výchozím stavu*. Není to však nezbytně nutné.
- 9 **Přiřazení adresy.** V tomto kroku je zařízení přiřazena nová unikátní adresa, kterou dále používá. Tato adresa mu zůstane do doby, kdy dojde k odpojení či resetování, kdy probíhá celý proces inventarizace sběrnice znovu.
- 10 **Host zjistí možnosti zařízení.** Host odešle znova žádost o deskriptor zařízení. Nyní jej přijme celý. Deskriptor obsahuje základní informace o zařízení včetně řady

nastavení, které zařízení podporuje. Poté host zažádá o deskriptor(y) nastavení (Configuration Descriptor) a deskriptor nastavení rozhraní (Configuration's Interface Descriptor(s)), který je následován deskriptorem s nastavením endpointů pro dané rozhraní. Některá nastavení vyžadují také deskriptory závislé na třídě či výrobci. Takové jsou odeslány následně.

- 11 Přiřazení na nahrání ovladačů zařízení.** Podle informací získaných v předchozím bodě se snaží host najít odpovídající ovladač pro správnou komunikaci se zařízením. Obvykle jsou nejdříve hledány shody u výrobce (Vendor ID) a jeho zařízeními (Product ID), následně se hledá shoda se třídou (Class), podtřídou (Subclass) a protokolem. Po nahrání vhodného ovladače může být potřeba zaslání dalších specifických deskriptorů podle dané třídy. Výjimku v tomhle bodě tvoří tzv. složená zařízení (Composite Devices). U těch může být pro každé rozhraní jiný ovladač, který je možný nahrát až po povolení daného rozhraní. K jeho povolení však musí být zařízení ve stavu *nastaveno*.
- 12 Nastavení zařízení.** Ovladač zašle žádost o nastavení zařízení. Zařízení může podporovat více možností nastavení a proto je zasláno i číslo dané konfigurace. Zařízení aplikuje požadovaná nastavení a povolí dané rozhraní. Nyní se nachází ve stavu *nastaveno*. V případě složeného zařízení se dále nahrají ovladače daných rozhraní podle předchozího bodu.

Pokud je zařízení ve stavu *připojeno*, znamená to, že hub nenapájí sběrnici. To může nastat z důvodu příliš velkého odběru proudu či host zažádal o odpojení napájení. Pokud je zařízení napájeno ze sběrnice, nemůže s hostem komunikovat a výsledkem je stav, jako když je zařízení úplně odpojeno.

Do stavu *pozastaveno* se dostane zařízení, jestliže neprobíhá žádná komunikace (včetně značek signalizujících počátek rámce) po dobu delší jak 3 ms. V tomto stavu by mělo zařízení omezit odběr proudu ze sběrnice na minimum. Tento stav musí být schopno zařízení využívat jak v *nastaveném* stavu, tak v *nenastaveném*.

## 2.6 Koncové body

Hostitelské zařízení komunikuje s periferním zařízením pomocí rour (Pipes, kapitola 2.7) [3]. Na konci roury se na straně periferního zařízení nachází koncový bod (Endpoint) [4]. Každé logické zařízení, s jedinečnou adresou v rámci sběrnice, je složeno z několika nezávislých koncových bodů. Ty mají již od výrobce přiřazené číslo koncového bodu v rozsahu 0–15. Koncové body jsou jednosměrné, přičemž směr toku dat je také pevně dán. Koncový bod je jednoznačně identifikován pomocí adresy zařízení, čísla koncového bodu a směru toku dat. Z toho tedy vyplývá, že je možné mít dva koncové body se stejným číslem, pokud budou mít opačný směr toku dat.

Typ přenosů mezi koncovým bodem a klientským SW popisuje jeho charakteristika. Ta popisuje koncový bod pomocí:

- Požadavků na frekvenci a latenci přístupů ke sběrnici.
- Požadavkem na šířku přenosového pásma.
- Čísla koncového bodu.
- Požadavků na chování obsluhy chyb.
- Maximální velikost paketu, kterou je koncový bod schopen přijmout či odeslat.
- Typem přenosu.
- Směrem přenosu.

Koncové body jsou před konfigurací zařízení v neznámém stavu a hostitelské zařízení by je nemělo využívat. Výjimkou je koncový bod s číslem nula. Jedná se o dva body (výstupní a vstupní), avšak pro zjednodušení se označují jako jeden bod. Ten je určen jako výchozí bod. Pomocí něho je umožněná základní manipulace s periferním zařízením, jako je jeho konfigurace či stav zařízení. Tento bod podporuje pouze řídicí přenosy (viz kap. 2.9).

Ostatní koncové body, především jejich počet, jsou určeny požadavky funkcí poskytovaných periferním zařízením. Zařízením podporující low-speed je umožněno použít pouze dva další koncové body [24]. Zařízení ve full-speed a hi-speed režimu mohou využít až maximálních 15 dalších koncových bodů.

## 2.7 Roury

Reprezentují schopnost přenosu dat mezi SW na straně hostitele, jeho vyrovnávací pamětí, a koncovým bodem na straně periferního zařízení. Roura může být jak jednosměrná, tak obousměrná. Může tedy využívat dvou koncových bodů se stejným číslem, ale opačným směrem toku dat. Jsou definovány dva typy rour [9]:

- Proudová roura (Stream).
- Roura zpráv (Message).

Proudové roury slouží pro přenos dat, jejichž struktura není nijak určená. Jedná se o sekvenční přenos dat. Proudové roury jsou jednoznačně určeny koncovým bodem a jeho směrem. Jedná se tedy vždy o jednosměrné roury. Stejný koncový bod s opačným směrem přenosu pak může být využit pro další rouru. Tyto roury nejsou synchronizovány pro přístup více klientů k jedné rouře. Tyto je možné využít pro hromadné, izochronní a přerušovací přenosy (viz kapitola 2.9).

Na rozdíl od proudové roury, roura zpráv má pro přenos dat přesně danou strukturu. Obsah datové části je ale také nespecifikován. Roura zpráv komunikuje s koncovým bodem ve třech krocích. Nejprve je zaslána hostitelem žádost. Tu následuje přenos požadovaných dat v požadovaném směru a na konec je přenesen status požadavku. Z toho vyplývá, že tento typ roury umožňuje obousměrnou komunikaci, přičemž komunikační tok je převážně jednosměrný. Je možné také využít rouru zpráv více klienty. Opět ale není možné vyřizovat jejich požadavky současně a jsou vyřizovány postupně v pořadí jejich obdržení. Pokud dojde ke zrušení přenosu zprávy ze strany hostitele, může být rourou odeslána další zpráva v pořadí. Ta je, ze strany periferního zařízení, předčasná (ještě nebyla dokončena předchozí). Tím dojde z pohledu SW obsluhujícího rouru k chybě, tento požadavek je ukončen a další odloženy. Tato roura podporuje pouze řídicí přenos dat a defaultní roura (Default Control Pipe) je vždy jejího typu.

S každou rourou je spjat:

- Požadavek na přístup a šířku pásma sběrnice;
- Typ přenosu;
- Charakteristické vlastnosti koncového bodu (např. maximální velikost paketu).

Roura spojující koncové body nula je označována jako výchozí. Je k dispozici hned po připojení napájení a obdržení resetovacího signálu.

Komunikace pomocí rour vzniká na základě odeslání paketu s požadavkem na vstup či výstup dat (I/O Request Packet, IRP) [1] od klientského SW. Po zpracování požadavku je klientský SW informován o výsledku jeho provedení, tedy i v případě neúspěšného provedení. V případě, že je roura nevyužita, tj. neprovádí žádný IRP ani žádný IRP na provedení nečeká, je ve stavu nečinnosti (idle state). V takovémto případě není dané rouře přidělena žádná šířka přenosového pásma.

V případě neisochronní roury, kdy nastane situace vedoucí s vyslání signálu STALL [26] (koncový bod je zastaven nebo roura nepodporuje daný požadavek) či



pokud dojde ke třem po sobě jdoucím chybám sběrnice u nějakého paketu v rámci IRP, je tento IRP přerušen. Následné IRP ve frontě jsou odloženy a příjem dalších je zastaven, dokud není závada klientským SW odstraněna. Pokud je koncový bod zaneprázdněn, nemusí to být důvod k oznámení chyby. Stačí, že koncový bod na paket odpoví signálem NAK [25] (Not AcKnowledge), který pouze způsobí pozastavení probíhajícího požadavku. Maximální počet těchto signálů není omezen.

Je možné, že je požadován přesun dat, jehož objem je větší než maximální velikost paketu. Poté dochází k rozdělení dat do více paketů. V případě posledního paketu se může USB systém zachovat dvěma způsoby, v závislosti na očekávání klienta:

- Klient očekává proměnné množství dat.
- Klient očekává předem určené množství dat.

V prvním případě je neúplný paket brán jako signál konce dat, IRP je dokončen bez chybového hlášení a je možné provést další požadavek. V druhém případě neúplný paket, tím pádem menší objem přijatých dat než byl očekáván, znamená chybu. Ta má za následek zastavení roury a následující požadavky jsou odloženy.

## **2.8 Struktura komunikačního toku**

### **2.8.1 Rámce a mikrorámce**

Rámec (Frame) [3] dle standardu USB je časový interval o velikosti jedné milisekundy. Mikrorámec je definován jako časový interval o velikosti 125 mikrosekund. Rámce jsou využívány při rychlostech low a full-speed a při hi-speed je využíván mikrorámec. Platí, že (mikro)rámec může obsahovat i několik transakcí. Jaké transakce jsou povoleny v daném rámci určuje typ přenosu (kap. 2.9). Koncové body využívající izochronní a přerušovací přenos mohou využívat jen každý N-tý rámec (N se může rovnat jedné).

### **2.8.2 Transakce a pakety**

Transakce [3] je komunikace s daným zařízením. Je složena ze dvou až tří paketů. Komunikace je na USB vždy zahájena hostem, který jako první vyšle token paket, tzv. peška (jako u kruhové topologie v PC síti). Tento paket určuje konkrétní zařízení, se kterým se bude komunikovat a jeho požadovaný koncový bod. Následně je vyslán datový paket s požadovanými daty. Tento paket může vyslat jak hostitelské, tak periferní zařízení. Poslední možný paket transakce je handshake paket. Ten slouží pro

potvrzení úspěšnosti přenosu. Vyslat jej můžou opět obě strany. Logicky lze odvodit, že potvrzovací paket odesílá opačná strana než datový paket.

## 2.9 Typy přenosů

Jak již bylo řečeno, konkrétní formát ani význam přenášených dat, tedy požadovaných informací, není nijak určen. Přesto USB poskytuje různé typy přenosů, které jsou co nejvíce optimalizovány pro poskytnutí nabízených služeb koncového zařízení. Pro každý typ přenosu jsou charakteristické určité vlastnosti, které jsou pevně dané při vzniku roury a neměnné do jejího zrušení. Některé vlastnosti však můžou být změněny i v průběhu života roury. Mezi vlastnosti komunikačního toku určené typem přenosu jsou [24]:

- Formát dat předepsaný specifikací USB.
- Směr komunikačního toku.
- Omezení velikosti paketu.
- Omezení přístupu na sběrnici.
- Omezení zpoždění.
- Požadovaný sled dat.
- Řešení chyb.

Především tyto vlastnosti odlišují typy přenosů, které se dělí na [24]:

- Řídící (Control Transfers).
- Izochronní (Isochronous Transfers).
- Přerušovací (Interrupt transfers).
- Hromadné (Bulk Transfers).

### 2.9.1 Řídící přenosy

Pomocí řídicích přenosů je možný přístup k různým částem periferního zařízení. Jsou využívány pro přenos konfiguračních, řídicích a stavových dat. Tento typ přenosu je zahájen transakcí SETUP od hostitelského zařízení. Ta je následována různým počtem transakcí (může být i nulový) přenášející dodatečná data a nakonec je vyslána stavová transakce STATUS pro informování hostitele o úspěchu požadované operace.

Jak již bylo zmíněno výše, řídicí přenosy je možné přenášet pouze přes rouru zpráv a podporují obousměrný přenos dat. Omezení velikosti datového paketu následující SETUP paket jsou 8 bajtů pro low-speed zařízení; 8, 16, 32, a 64 bajtů pro full-speed zařízení a 64 bajtů pro hi-speed [24]. Dále je určeno, že pokud řídicí přenosy

nezabírají více než 10 % rámce či 20 % mikrorámce, může být zbývající přenosová rychlost využita pro hromadné přenosy [24]. Naopak pokud je zapotřebí pro řídicí přenosy větší kapacity než je dostupná a současně jsou k dispozici nevyužité rámce pro izochronní nebo přerušovací přenosy, je možné je využít. V případě chyby během přenosu zjištěné hostitelským zařízením je činnost koncového bodu obnovena obdržetím dalšího setup paketu. Není tedy k obnově zapotřebí jiné roury. Pokud se ale jedná o defaultní kontrolní rouru a nedojde k obdržetím dalšího setup paketu, dojde k resetování koncového zařízení.

### **2.9.2 Izochronní přenosy**

Tento typ přenosu je využíván zejména pro přenos dat v reálném čase (např. videa či audia) požadujících konstantní šířku přenosového pásma a zpoždění. V případě chyby by opakovaně odeslaná data již byla neaktuální a tak se opětovné vyslání neprovádí. Tím ztrácí význam handshake paket, který se u tohoto typu přenosu nevyskytuje. To má za následek, že endpoint nemůže být nikdy zastaven.

Jedná se vždy o jednosměrný přenos dat podporující maximální velikost paketu 1023 bajtů pro full-speed a 1024 bajtů při hi-speed [24]. Low-speed zařízení tento typ přenosu nemohou využívat. Při vytváření izochronní roury je kontrolována dostupná přenosová rychlost. Pokud není možné poskytnout dostatečnou šířku přenosového pásma, není roura vytvořena. Během inicializace USB rozhraní není zapotřebí konfigurace izochronního koncového bodu. Ta je možná kdykoliv stejně tak jako možnost změnit šířku potřebného přenosového pásma. Je požadováno, aby tento typ přenosu zabíral maximálně 90 % rámce a 80 % mikrorámce [24]. Izochronní koncový bod musí specifikovat požadovaný interval přístupu ke sběrnici. Musí být také schopen odesílat data s nižší periodou. Ovšem v případě, že nemá k dispozici požadované data, odesílá prázdný (mikro)rámec. To je důsledek chybějícího handshake paketu.

### **2.9.3 Přerušovací přenosy**

Tyto přenosy jsou využívány u koncových bodů nevysílajících data často, ale vyžadující zaručenou maximální periodu vyřízení požadavku a opětovné vyslání paketu v případě chyby.

Jedná se opět o jednosměrný typ přenosu s maximální velikostí paketu 1024 bajtů pro hi-speed. Pro full-speed je velikost snížena na 60 bajtů a low-speed podporují maximální velikost paketu 8 bajtů [24]. Velikost přenášených dat je nastavena během inicializace a je neměnná. V případě nedostatku požadované přenosové rychlosti není roura vytvořena. Omezení přístupu je shodné s izochronními přenosy, tedy maximálně 90 % rámce či 80 % mikrorámce [24]. Opět může nastat

případ, kdy nejsou připravena požadovaná data k odeslání. Zde však nechybí handshake paket a je vyslán signál NAK.

#### **2.9.4 Hromadné přenosy**

Poslední typ přenosu je využíván pro přenos dat, které nejsou kritické, tzn. nepožadují garanci přenosového pásma, periody přístupu či konstantního zpoždění. Současně ale požadující co největší možné přenosové pásmo. V případech málo využívané sběrnice je tedy možné dosáhnout velmi vysokých přenosových rychlostí.

Maximální velikost paketu je zde omezena na 64 bajtů pro full-speed a 512 bajtů pro hi-speed [24]. Ani tento typ přenosu není možné použít pro low-speed zařízení. Během inicializace zařízení je zajištěno, aby nebyl překročen objem dat, který je koncový bod schopen zpracovat. Maximální šířka přenosového pásma zde není nijak omezena. Je pouze stanovena nejnižší priorita pro hromadné přenosy. To znamená, že je možné obsadit pouze zbývající přenosovou kapacitu. V případě výskytu chyby je požadovaný přenos opakován.

#### **2.10 Deskriptory**

Jsou to datové struktury (nebo bloky informací v daném formátu), ze kterých se host dozvídá informace o připojeném zařízení. Mohou obsahovat informace o zařízení jako takovém, či jen jeho části. Veškerá zařízení musí být schopna odpovědět na žádosti o standardní USB deskriptory. Standardní typy deskriptorů jsou uvedeny v tabulce 2.2, kde *bDescriptorType* je kód deskriptoru. Lze říci, že s rostoucí úrovní deskriptoru obsahuje deskriptor specifitější údaje. Každé zařízení má jeden či více deskriptorů nastavení. Ty říkají, jak zařízení využívá napájení sběrnice a počet rozhraní. Každý deskriptor rozhraní specifikuje žádný či více deskriptorů koncového bodu obsahující informaci pro komunikaci skrze daný koncový bod. Pokud není specifikován deskriptor koncového bodu, rozhraní musí využít ke komunikaci řídicí koncový bod. Jestli zařízení obdrží žádost o deskriptor nastavení, odesílá za ním také deskriptory rozhraní, koncového bodu a případné podřaděné (Subordinate Descriptors) specifické deskriptory, dokud neodešle požadovaný počet bytů. Jelikož neexistuje požadavek o zaslání deskriptoru koncového bodu, musí být odeslány vždy deskriptory od deskriptoru nastavení.

Následně budou popsány vybrané deskriptory, ostatní obsahují stejné nebo podobné položky a je možné je nalézt v [24].

Tab. 2.2: Standardní typy deskriptorů

bDescriptorType	Typ deskriptoru	Vyžadován?
01h	Zařízení	Ano
02h	Nastavení	Ano
03h	Řetězce (string)	Ne
04h	Rozhraní	Ano
05h	Koncového bodu	Pouze pokud zařízení využívá jiné koncové body než 0
06h	Kvalifikace zařízení (device_qualifier)	Pouze pokud zařízení podporuje full i high-speed.
07h	Nastavení jiné rychlosti (other_speed_configuration)	Pouze pokud zařízení podporuje full i high-speed.
08h	Řízení napájení rozhraním (interface_power)	Ne
09h	OTG	Pouze OTG zařízení
0Ah	Ladící (debug)	Ne
0Bh	Přiřazení rozhraní (interface_association)	Pouze složená zařízení

### 2.10.1 Deskriptor zařízení

Deskriptor zařízení obsahuje základní informace o zařízení. Obsahuje celkem 14 polí zahrnujících informace o samotném deskriptoru, zařízení, jeho konfiguraci a třídu, do které spadá.

Pokud je kód třídy v rozsahu 1–126, jedná se o třídu pro celé zařízení. Hodnou 127 se poté značí třída závislá na výrobci. Hodnota 0 značí, že každé rozhraní má jinou třídu, která je určena v deskriptoru daného rozhraní. V tomto případě je i následná hodnota podtřídy rovna nule.

Obdobná situace je při určování protokolu. Pokud je hodnota v poli nula, zařízení jak celek nevyužívá protokol závislý na třídě zařízení, avšak jednotlivá rozhraní je využívat mohou. Opět pokud je hodnota pole rovna 127, zařízení využívá protokol specifikovaný výrobcem.

Identifikační číslo (ID) výrobce je unikátní a je přidělováno za poplatek organizací USB-IF. Platné identifikační číslo výrobce musí obsahovat každá deskriptor zařízení. Následné identifikační číslo produktu je unikátní v rámci ID výrobce.

Poslední tři údaje jsou ukazatele na deskriptor výrobce, produktu či sériového čísla, který obsahuje řetězec znaků popisující daného výrobce, produkt nebo sériové číslo.

Tab. 2.3: Deskriptor zařízení

Odsazení [byte]	Pole	Velikost [byte]	Popis
0	bLength	1	Velikost deskriptoru
1	bDescriptorType	1	Typ deskriptoru
2	bcdUSB	2	Verze specifikace USB v BCD kódu
4	bDeviceClass	1	Kód třídy
5	bDeviceSubclass	1	Kód podtřídy
6	bDeviceProtocol	1	Kód protokolu
7	bMaxPacketSize()	1	Maximální velikost paketu pro koncový bod 0
8	idVendor	2	ID výrobce
10	idProduct	2	ID produktu
12	bcdDevice	2	Verze produktu v BCD kódu
14	iManufacturer	1	Index pro deskriptor řetězce výrobce
15	iProduct	1	Index pro deskriptor řetězce produktu
16	iSerialNumber	1	Index pro deskriptor řetězce sériového čísla produktu
17	bNumConfigurations	1	Počet možných konfigurací

### 2.10.2 Deskriptor nastavení

Tento deskriptor obsahuje informace o využití napájení ze sběrnice a počet podporovaných rozhraní. Velké množství zařízení podporuje pouze jedno možné nastavení. Existuje však také řada více účelových zařízení nebo zařízení podporujících více režimů, kdy každý režim musí mít svůj deskriptor. Celkem deskriptor nastavení obsahuje osm polí (tab. Tab. 2.4).

Tab. 2.4: Deskriptor nastavení

Odsazení [byte]	Pole	Velikost [byte]	Popis
0	bLength	1	Velikost deskriptoru
1	bDescriptorType	1	Typ deskriptoru
2	wTotalLength	2	Počet bytů deskriptoru nastavení a jeho podřaděných deskriptorů
4	bNumInterfaces	1	Počet rozhraní v nastavení
5	bConfigurationValue	1	Identifikátor nastavení
6	iConfiguration	1	Index pro deskriptor řetězce s nastavením
7	bmAttributes	1	Nastavení pro samostatné napájení či ze sběrnice a vzdáleného probuzení
8	bMaxPower	1	Požadavky na napájení ze sběrnice

Pole *bConfigurationValue* určuje možná nastavení. Počet nastavení musí být vždy nenulový. V případě, že zařízení obdrží požadavek, aby nastavilo nastavení nula, dostane se do stavu *nenastaveno*.

Z pole *bmAttributes* se host dozví, zda je zařízení napájeno ze sběrnice a nebo má vlastní napájení. Dále je zde uvedeno, jestli je podporováno vzdálené probuzení (remote wakeup). To umožňuje zařízení ve stavu *pozastaveno* oznámit hostovi, že chce obnovit komunikaci.

Hodnota uvedená v posledním poli je polovina maximálního požadovaného odběru proudu ze sběrnice. Pokud není možné požadované množství proudu poskytnout, host zamítne konfiguraci zařízení. Ovladač může následně vybrat nějaké další podporované nastavení.

### 2.10.3 Deskriptor koncového bodu

Každý použitý koncový bod zařízení musí mít svůj deskriptor (tab.2.5). Výjimku tvoří pouze koncový bod nula, který musí podporovat všechny zařízení a jeho maximální možná velikost je uvedena v deskriptoru zařízení. Zbylá nastavení koncového bodu 0 jsou pevně daná. Pro získání tohoto deskriptoru musí host zažádat o deskriptor nastavení, jak bylo popsáno výše.

Tab. 2.5: Deskriptor koncového bodu

Odsazení [byte]	Pole	Velikost [byte]	Popis
0	bLength	1	Velikost deskriptoru
1	bDescriptorType	1	Typ deskriptoru
2	bEndpointAddress	1	Číslo a směr koncového bodu
3	bmAttributes	1	Typ podporovaného přenosu
4	wMaxPacketSize	2	Maximální podporovaná velikost paketu
6	bInterval	1	Interval přenosu dat

*bEndpointAddress* určuje číslo koncového bodu a jeho směr. Zařízení typu low-speed mohou využívat maximálně 3 koncové body. Ostatní zařízení jich může využít všech 16.

Bitsy v poli *bmAttributes* mají význam podle verze USB. Ve verzi USB 1.1 je využit nultý a první bit k určení typu přenosu (řídící, isochronní, hromadný, přerušovací) a zbylé bity jsou rezervovány. Ve verzi USB 2.0 se dále využívá u isochronního přenosu těchto bitů (v případě jiného typu přenosu jsou rovny nule). Druhý a třetí bit určuje typ synchronizace (žádná, asynchronní, adaptivní a synchronní). Další dva bity určují typ použití (data, zpětná vazba, implicitní zpětná vazba dat).

Pole *wMaxPacketSize* je opět využíváno různě pro USB 1.1 a USB 2.0. Předchozí verze využívá pouze deset bitů pro určení maximální velikosti paketu. Pozdější verze navíc využívá zbylé dva bity pro zjištění, kolik podporuje dodatečných transakcí (0–3) během jednoho mikrorámce.

Význam bitů v *binterval* se liší nejenom podle specifikace, ale také v závislosti na typu přenosu a rychlosti. Může se jednat buď o maximální latenci pro přenos dat, nebo maximální frekvenci NAK signálů odesílaných koncovým bodem. Rozsah hodnot tohoto pole a jejich význam je možné nalézt v [23].

## 2.11 Třídy USB zařízení

Aby nebylo nutné ke každému zařízení dodávat ovladač, byly zařízení rozděleny do tříd (Class) podle jejich funkce. Host poté má jeden ovladač pro všechny podobné zařízení. Většina těchto tříd má následně podtřídu (Subclass) a podtřídy mohou mít několik komunikačních protokolů. Definice třídy se může nacházet na dvou místech. V deskriptoru zařízení a deskriptoru rozhraní (může se nacházet i na obou) viz tabulka 2.6.

Tab. 2.6: Třídy USB zařízení

Kód třídy	Třída definována v deskriptoru	Název třídy
00h	Zařízení	Třída definována v deskriptoru rozhraní
01h	Rozhraní	Audio
02h	Obou	Communications Device and CDC Control
03h	Rozhraní	HID (Human Interface Device)
05h	Rozhraní	Physical
06h	Rozhraní	Image
07h	Rozhraní	Printer
08h	Rozhraní	Mass Storage
09h	Zařízení	Hub
0Ah	Rozhraní	CDC-Data
0Bh	Rozhraní	Smart Card
0Dh	Rozhraní	Content Security
0Eh	Rozhraní	Video
0Fh	Rozhraní	Personal Healthcare
DCh	Obou	Diagnostic Device
E0h	Rozhraní	Wireless Controller
EFh	Obou	Miscellaneous
FEh	Rozhraní	Application Specific
FFh	Obou	Vendor Specific



Dále bude probrána pouze třída Mass Storage . Informace o ostatních třídách je možné najít v [1].

### 2.11.1 Třída Mass Storage

Tato třída byla vytvořena pro zařízení využívající přenos v jednom nebo obou směrech jako jsou disketové a optické mechaniky, pevné disky či paměťové karty. Jelikož tyto zařízení využívají různé skupiny standardních sad příkazů, jsou pro ně vytvořeny podtřídy (viz tabulka 2.7). Vlastní přenos dat pak určuje protokol (tab. 2.8). Může se jednat o CBI (Control/Bulk/Interrupt transfers), BBB (Bulk-only transfers) a UAS (nebo také UASP). Protokol CBI se využívá výhradně pro full-speed disketové mechaniky a BBB protokol se využívá ve většině ostatních případů. Ačkoliv je UAS, který vychází z BBB, specifikován již v USB 2.0, využívá se až v USB 3.0.

Tab. 2.7: Podtřídy Mass Storage

Kód podtřídy	Specifikace skupiny příkazů
00h	SCSI, sada příkazů není hlášena
01h	RBC
02h	MMC – 5 (ATAPI)
03h	Bylo vyhrazeno pro QIC – 157 (již se nepoužívá)
04h	UFI
05h	Bylo vyhrazeno pro SFF – 8070i (již se nepoužívá)
06h	SCSI, transparentní sada příkazů
07h	LSD FS
08h	IEEE 1667
09h–FEh	Rezervováno
FFh	Závislé na výrobci

V případě BBB protokolu probíhá přenos dat ve třech fázích: přenos příkazu, dat a stavu. V první fázi odesílá host strukturu zvanou CBW (Command Block Wrapper), následuje přenos vlastních dat a ve třetí fázi zařízení odešle strukturu CSW (Command Status Wrapper). Některé příkazy neobsahují druhou fázi přenosu dat. Z čeho se tyto struktury skládají je možno nalézt opět v [1].

Ovladače Mass Storage pracují s logickou jednotkou v zařízení. Pokud má zařízení více datových úložišť stejného typu, je každému přiřazeno jiné číslo logické jednotky (LUN, Logical Unit Number) a host je může ovládat nezávisle na sobě.

Tab. 2.8: Protokoly třídy Mass Storage

Kód Protokolu	Protokol
00h	CBI (příkazy soutěží o přerušovací přenos)
01h	CBI (příkazy nesoutěží o přerušovací přenos)
02h	Již se nepoužívá
03h–04h	Rezervováno
50h	BBB
21h–61h	Rezervováno
62h	UAS
63h–FEh	Rezervováno
FFh	Závislé na výrobci

Dnešní USB Flash disky a externí pevné disky využívají ke komunikaci SCSI příkazy [23]. Těch je poměrně velké množství a není zapotřebí je všechny implementovat. Musí být implementovány minimálně příkazy uvedené v tabulce 2.9. Příkaz INQUIRY slouží ke zjištění informací o zařízení, jako jsou např. podporované příkazy, zda je paměťové médium vyjímatelné, řetězec s názvem výrobce a další. REQUEST SENSE se využívá pro zjištění případné chyby. Použití ostatních názvů vyplývá z jejich názvu. Více o SCSI příkazech je možné zjistit na [23].

Tab. 2.9: Povinné SCSI příkazy

Název příkazu	Kód příkazu
FORMAT UNIT	04h
INQUIRY	12h
READ (10)	28h
READ CAPACITY (10)	25h
REPORT LUNS	A0h
REQUEST SENSE	03h
SEND DIAGNOSTIC	1Dh
TEST UNIT READY	00h
WRITE (10)	2Fh

### 3 Souborový systém FAT

Jedná se o systém organizace dat na paměťovém médiu. Není tedy závislý na fyzickém typu datového úložiště ani jeho způsobu komunikace. Jelikož adresovat každý byte by při dnešních kapacitách vyžadovalo extrémně dlouhé adresy a byla by zapotřebí značná režie během přenosu, jsou datové média rozdělena do sektorů, obvykle o 512 bytech. Tato hodnota je používána z historických důvodů. Sektory jsou následně sloučeny do clusterů a až ty jsou adresovány. Jeden cluster se může skládat z jednoho až 128 sektorů. Podle množství bitů použitých pro adresaci se rozlišuje FAT12, FAT16 a FAT32 [21]. Dále existují verze VFAT, FAT+ a exFAT. Jejich podpora nebude však v zařízení implementována a proto nejsou dále popisovány. U FAT32 je omezení na použití u disků, které mají více clusterů než je možné adresovat při použití FAT16.

Systém souborů FAT (File Allocation Table) rozděluje disk na specifické části, jak je znázorněno v tabulce 3.1. V případě FAT32 je pouze jedna odlišnost a to, že kořenový adresář nemá vyhrazen svůj vlastní prostor, ale nachází se mezi ostatními adresáři (viz dále).

Master Boot Record obsahuje informace o jednotlivých oddílech. Ty mohou být až čtyři. Jednotlivé oddíly se řadí za sebe a každý obsahuje části od Boot Sectoru až po prostor pro data. Použitím oddílů se docílí rozdělení jednoho fyzického disku na více logických. Na začátku sektoru s MBR je ještě část zvaná „Executable Code“. Pomocí té se určí, ze kterého oddílu bude počítač bootovat.

Tab. 3.1: Rozdělení paměťového média podle FAT12 a FAT16

Master Boot Record	
Nevyužité místo na disku	
První oddíl	Boot Sector
	Nevyužité místo na disku
	FAT 1
	⋮
	FAT n
	Kořenový adresář
	Oblast dat

Boot Sektor obsahuje základní informace o oddílu, mezi které například patří: název operačního systému na oddílu, velikost sektoru a clusteru, název souborového systému, název disku (Disk Volume) a v případě FAT32 hlavně pozici prvního clusteru kořenového adresáře.

Oblast FAT slouží k navigaci v oblasti dat. Skládá se z částí stejně dlouhých jako adresa. Význam jednotlivých hodnot pro FAT32 je uveden v tabulce 3.2. Při uložení souboru na disk je tento soubor rozdělen do jednotlivých clusterů a jejich návaznost je uložena právě zde. Navazující části souboru tedy nemusí být v clusterech za sebou. Jelikož chyba ve FAT tabulce by měla fatální následky, vytvářejí se její kopie (FAT 1–n).

Tab. 3.2: Hodnoty FAT tabulky pro FAT32

Hodnota	Význam
0x00000000	Nevyužitý cluster
0x00000001	Rezervovaná hodnota
0x00000002–0x0FFFFFFEF	Ukazatel na navazující cluster
0x0FFFFFFF0–0x0FFFFFFF6	Rezervované hodnoty
0x0FFFFFFF7	Vadný sector clusteru nebo rezervovaný cluster
0x0FFFFFFF8–0x0FFFFFFF	Poslední cluster souboru

Prostor pro kořenový adresář je pevně daný a jeho velikost je uvedena v Boot Recordu. Obvykle je to hodnota 512 položek. Položky (Entries) jsou informace o jednotlivých souborech nebo adresářích (viz tabulka 3.3). Jeden vstup je veliký 32 bytů. Tyto položky mohou mít ve jménu maximálně osm znaků velkých písmen nebo číslic. To se stalo později velkým omezením a byla specifikována položka pro dlouhý název (Long File Name, LFN). Ta je uvedena v tabulce 3.4. Pole atributy a první cluster jsou zde zachovány z důvodu zpětné kompatibility (díky jejich hodnotám jsou LFN položky ignorovány). V případě výskytu dlouhého jména je pořadí od poslední položky LFN do první položky LFN a nakonec je položka souboru. Položky LFN a souboru, které spolu souvisí již musí být umístěny za sebou.

Tab. 3.3: Informace uvedené v položce

Odsazení [byte]	Popis
00h–07h	Jméno souboru
08h–0Ah	Přípona
0Bh	Atributy
0Ch–0Fh	Rezervováno
10h–15h	Rezervováno
16h–17h	Čas vytvoření
18h–19h	Datum vytvoření
1Ah–1Bh	První cluster
1Ch–1Fh	Délka

Poslední částí je oblast dat. V jednotlivých clusterech jsou data souborů nebo obsah adresáře (Directory Table). Obsah adresáře obsahuje opět položky, stejně jako kořenový adresář.

Tab. 3.4: Položka LFN

Odsazení [byte]	Popis
00h	Část názvu (v případě poslední části je hodnota zvětšena o 40h)
01h–0Ah	10 znaků jména
0Bh	Atributy
0Ch	Rezervováno
0Dh	CRC
0Eh–10h	12 znaků jména
1Ah–1Bh	První cluster (vždy 0)
1Ch–1Fh	4 znaky jména

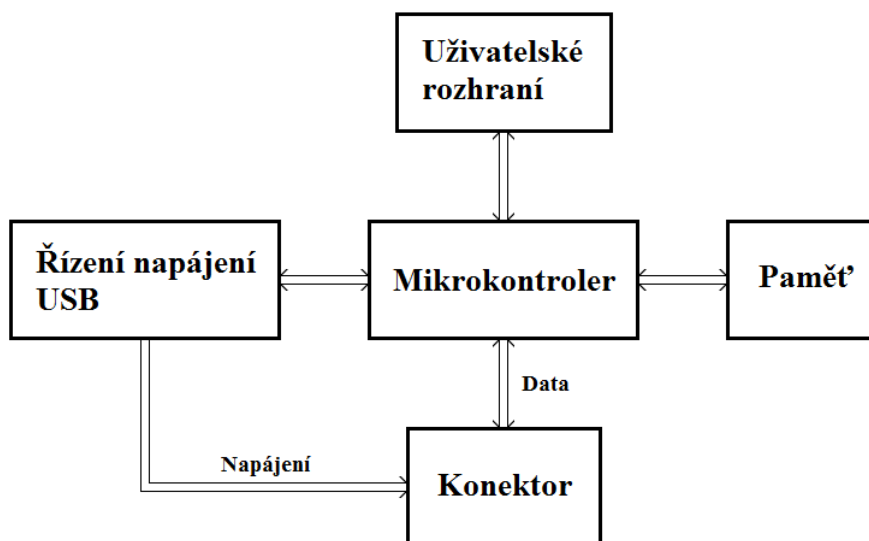
## 4 USB OTG zařízení

### 4.1 Návrh zařízení

Cílem je navrhnout zařízení s rozhraním USB schopných pracovat jak v režimu koncového zařízení, kdy bude připojeno k PC, tak v režimu hostitelského zařízení, kdy k němu bude možné připojit periferní zařízení. Jako funkce vhodná implementaci se nabízí správa souborů pomocí protokolu Mass Storage, což je u USB OTG velmi často využíváno.

#### 4.1.1 Blokové schéma zařízení

Blokové schéma je uveden na obrázku 4.1. Pro správnou funkci koncového zařízení je potřeba pouze mikrokontroler a paměť. V případě hostitelského zařízení je zapotřebí přidat ještě napájení sběrnice, které je možné odpojit, bude-li zařízení v režimu koncového zařízení. Další blok, z pohledu USB komunikace zbytečný, pro zařízení však nezbytný, je uživatelské rozhraní pro ovládání zařízení.



Obr. 4.1: Blokové schéma zařízení

#### 4.1.2 Volba součástek

##### Výběr mikrokontroléru

Jako nejdůležitější je výběr mikrokontroleru. Jako kritéria, samozřejmě včetně podpory USB OTG, jsou dostupnost, pouzdro a cena, která zde také hraje nezanedbatelnou roli.

Co se týče dostupnosti v ČR, jsou v nabídce desítky mikrokontrolerů od různých výrobců. Při výběru pouze mikrokontrolerů, které jsou skladem, se výběr značně zúží. Při osazování součástek je mikrokontroler v SMD pouzdře velikou překážkou (mnohdy nemožnou překonat) a seznam se zúží na jednotky položek. Jako nejlevnější ze zbylých je mikrokontroler PIC24FJ64GB002 [18] od Microchip. S nejmenší cenou nabízí, co se USB OTG týče, prakticky stejné možnosti jako dražší mikrokontrolery. Jeho nejdůležitější vlastnosti pro toto zařízení jsou:

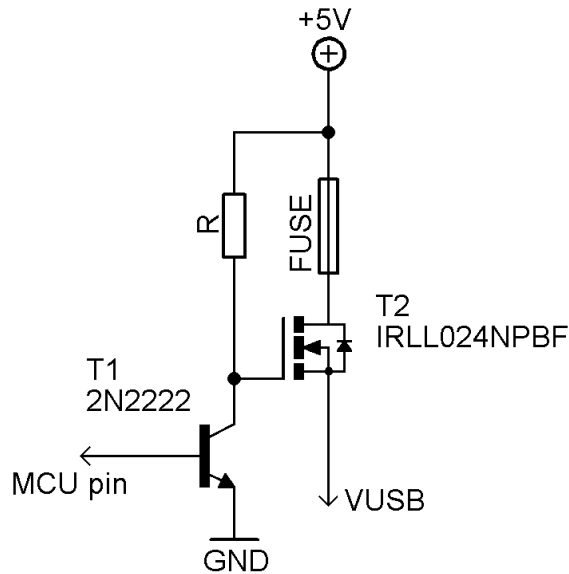
- Podpora USB 2.0, OTG.
- Podpora rychlostí low-speed a full-speed v režimu hostitele.
- Podpora rychlosti full-speed v režimu koncového zařízení.
- Až 16 obousměrných endpointů.
- Integrované rezistory pull-up a pull-down pro výběr rychlosti a funkce zařízení.
- Integrovaný 8MHz oscilátor s přesností 0,25 % (přesnost dostačující pro USB).
- SPI modul (pro komunikaci s pamětí).
- Pouzdro SPDIP28.

### **Součástky pro řízení napájení**

Způsob řízení napájení, se schopností dodat základní napájecí proud, je přímo doporučen v [18]. Jedná se o DC/DC měnič napětí, konkrétně MCP1253. Jeho zapojení, včetně připojení k mikroprocesoru, je také uvedeno [18]. Tento integrovaný obvod je schopný poskytnout napájení 5 V a 120 mA. Pro posílení napájecí části, tedy možnosti dodat až 500 mA, poslouží vratná pojistka a tranzistor (Obr. 4.2). Vratná pojistka bude použita MINISMDC050F jejíž odpor je maximálně 1 Ω. Nejvhodnější tranzistor je IRL024NPBF. Jedná se o výkonový unipolární tranzistor. V tomto případě hlavní roli nehrálo, zda je tranzistor výkonový, ale jeho vnitřní odpor při napětí na hradle 5 V. Ten je u tohoto typu menší než 0,1 Ω. Celkový maximální úbytek napětí tedy bude:

$$U = (R_T + R_p) \cdot I = (0,1 + 1) \cdot 0,4 = 0,44 \text{ V}, \quad (1)$$

kde  $R_T$  je odpor tranzistoru,  $R_p$  odpor pojistky a  $I$  proud procházející tranzistorem a pojistkou. Proud 400 mA je zvolen jako rozdíl maximálního dodávaného proudu do USB a proudu dodávaného MCP1253. Při napájecím napětí 5 V tedy bude na sběrnici USB napětí 4,56 V, což je stále v toleranci [24]. Mikroprocesor je však napájen 2,5 V. To na dostatečné sepnutí daného tranzistoru nestačí. Je tedy zapotřebí použít další tranzistor, kterým se potřebných 5 V připojí. Zde je již možné použít např. jakýkoliv NPN tranzistor. Mojí volbou je 2N2222.



Obr. 4.2: Posílení napájení sběrnice

### Volba paměti

Jako paměť je ideální použít paměťovou kartu, například typu SD, která komunikuje pomocí sériového rozhraní SPI. Zařízení je pak možné použít i jako čtečku SD karet. Komunikace s paměťovou kartou je ale omezena na rychlost full-speed, kde se pohybuje reálná přenosová rychlost lehce přes 1 MB/s. V případě kopírování souboru z SD karty na flash disk by se rychlost kopírování pohybovala na mnohem nižší hodnotě. Takové kopírování by bylo, při dnešních kapacitách paměťových karet a standardních rychlostech, velmi zdouhavé. Navíc jelikož se jedná o demonstrační zařízení, jako ukázka bohatě poslouží přenos malých dokumentů (například textových souborů). Byla tedy zvolena paměť EEPROM. Ta nabízí nesrovnatelně menší kapacitu paměti, ovšem za ani ne poloviční cenu v případě jedné z rychlejších pamětí. Vhodná volba je, s ohledem na kapacitu a rychlost, paměť 25LC512 [10], jejíž nejdůležitější vlastnosti jsou:

- Maximální hodinový kmitočet 20 MHz pro  $V_{CC} \geq 4,5$  V; 10 MHz pro  $V_{CC} \geq 4,5$  V.
- Kapacita paměti 512 kb.
- Doba zápisu stránky 5 ms a velikost stránky 128 B.
- Komunikace pomocí SPI.

Doba zápisu stránky značně snižuje rychlost zápisu přibližně na 25 kB/s, ale i přes to je možné zaplnit celou paměť za 2,5 s, což je přijatelná hodnota.



## Součástky pro uživatelské rozhraní

Pro uživatelské rozhraní je zapotřebí několik tlačítek a zobrazovací jednotka. Zobrazovací jednotka by měla být co největší pro přehlednější procházení souborů. Při zohlednění ceny a složitosti ovládání je vhodnou volbou prakticky jakýkoliv alfanumerický LCD displej s radičem, se čtyřmi řádky a dvaceti znaky na řádek. Čtyři řádky na procházení souborů není mnoho, ale alfanumerické displeje s více řádky nejsou v nabídce a větší grafické displeje jsou příliš drahé. Nabízených typů je mnoho a liší se minimálně. Konkrétní vybraný typ je RC2004AB [22]. Byl vybrán na základě přítomnosti podsvětlení a, dle mého názoru, uživatelsky příjemnější kombinace barev (modré a bílé). Displej je řízen pomocí šesti pinů. Čtyři jsou datové a zbylé slouží pro hodinový signál a výběr registru. U displeje je možné také číst status registr, ze kterého se dozvíme, zdali už je dokončená poslední instrukce. Pro přepínání směru komunikace slouží pin *RW*. Z důvodu ušetření pinu na mikrokontroléru je tento pin trvale uzemněn a z displeje není možné číst. Místo kontrolování status registru se bude čekat zaručeně dlouhou dobu, kdy už bude instrukce vykonána. Další důležitý pin displeje je *VO*, kterým se řídí kontrast ten je podle doporučení připojen přes 10 k $\Omega$  trimr mezi napájecí napětí a zem. Poslední z pinů slouží pro napájení podsvětlení. Pro možnost vypnutí podsvětlení je zde zapojen přepínač.

K ovládání zařízení je použito sedm tlačítek. Čtyři ve funkci kurzorů a zbylé tři ve funkci kláves *escape*, *enter* a *menu*. Zde nastává problém, jelikož už jsou volné jenom tři vstupní piny na mikrokontroléru. Je tedy zapotřebí použít další obvod, kódér 1:N. Ten podle toho, který ze vstupů je aktivní, vyšle na výstup požadovanou kombinaci (tabulka 4.1). Jelikož jsou volné tři piny na mikrokontroléru, je možné odlišit osm stavů. Osmý stav říká, že žádné z tlačítek nebylo stisknuto.

Tab. 4.1: Kódování tlačítek

Funkce tlačítka	Zakódovaná hodnota
nestisknuto	000
Nahoru	001
Dolu	010
Vpravo	011
Vlevo	100
Enter	101
Escape	110
Menu	111

Nyní je zapotřebí ještě připojit programovací piny. Všechny piny na mikrokontroléru jsou již obsazeny a musí být tedy sdíleny s nějakým periferním zařízením. V případě programování nebo ladění bude muset být vybrán periferní obvod odpojen od sdílených pinů. Jako nejméně důležité jsou tlačítka. To je v podstatě i nejlepší možnost. Místo hardwarového kodéru je zde použit další mikrokontroler, což značně zjednodušuje odpojení sdílených pinů během programování. Jednoduše se tyto piny přepnou na funkci vstupních pinů, kdy jsou ve stavu vysoké impedance. Třetí pin je stále výstupní a tak je možné jedno tlačítko použít pro ladění programu. Je zde použit mikrokontrolér PIC16F505 [17]. Kritéria pro výběr byly pouze dvě, interním oscilátor a požadovaný počet vstupních a výstupních pinů.

Celé schéma zapojení je uvedeno v příloze spolu s deskou plošného spoje (DPS). Během ožívování bylo zapotřebí mírně pozměnit schéma (viz dále). Jelikož většina součástek mohla být přidána jako SMD, nebylo nezbytně nutné navrhovat nový plošný spoj. Zbylé součástky jsou přiletovány „ze spodu“. Schéma je tedy aktuální, ale navržené PDS neobsahuje všechny jeho součástky. Dále jelikož displej je umístěn nad DPS, jsou na DPS pouze pájecí plošky. Ty jsou vodiči propojeny s odpovídajícími piny na displeji.

## 4.2 Ožívování zařízení

Pro programování mikrokontrolérů byl využit debugger ICD2 a vývojové prostředí MPLAB v8.66. Dále byl zapotřebí kompilér. Pro 16bitové procesory řady PIC24 má Microchip vlastní kompilér C30, poslední verze je 3.25. V případě 8bitových mikrokontrolérů je zapotřebí použít kompilér třetí strany. Použil jsem CC5X v3.4E.

### 4.2.1 Kodér 1:N

Zde se vyskytla první chyba v návrhu. Tlačítka jsou připojená k napájecímu zdroji (5 V) a mikrokontrolér je napájen 2,5 V. Ten obsahuje tzv. Clamp diody na vstupech [17], které umožňují připojení vstupů s vyšším napětím než je napájecí. Díky tomu se však zvýšilo napájecí napětí mikrokontrolérů na nepřijatelné 3 V. Bylo tedy zapotřebí připojit tlačítka pomocí napěťového děliče a tím snížit napětí na vstupech mikrokontroléru na 2,5 V.

Program pro kódování stisknutého tlačítka je velmi jednoduchý. Celý program tvoří pouze dvě podmínky. První detekuje, zdali probíhá programování nebo ladění hlavního mikrokontroléru. To je možné detekovat pomocí signálu !MCLR, který vyvolá reset zařízení. V případě, že je tato podmínka splněna, nastaví se piny společně s programovacími na funkci vstupů. Druhá podmínka detekuje stisk tlačítka. Pokud je splněna, odešle se na výstup zakódovaná hodnota tlačítka. Následně se čeká na jeho uvolnění (nebo minimálně 70 ms) a pak se na výstup pošle kód odpovídající

nestisknutému tlačítku. Minimální čekací doba 70 ms je zde kvůli zákmitům při stisku tlačítka. Její hodnota byla odvozena empiricky.

#### 4.2.2 Základní nastavení mikrokontroléru PIC24FJ64GB002

V první řadě je zapotřebí nastavit Configuration Word 1, 2, 3 a 4 (CW1, CW2, CW3, CW4). Ty je možné nastavit v MPLABu v nabídce *Configure* → *Configuratarion bits...* nebo přímo v kódu. Já jsem zvolil druhou možnost a v tom případě je potřeba ve výše uvedené nabídce zaškrtnout položku *Configuration Bits set in code*. Nejdůležitější nastavení těchto registrů je v tabulce 4.2. Ostatní položky jsou pro konfiguraci vypnutých modulů nebo nemají na funkci tohoto zařízení vliv. Dále je nutné nastavit vnitřní oscilátor a děličky kmitočtu. Blokové schéma této části mikrokontroléru PIC24FJ64GB002 je již poměrně složité [18] [20]. Část nastavení obsahuje CW2 a další nastavení je CPDIV a RCDIV nastavené na nulu. Výsledkem je frekvence USB modulu 48 MHz a ostatní části mikrokontroléru pracují na frekvenci 16 MHz. Poslední částí základního nastavení je získat I/O porty „pod kontrolu“. Tedy ujistit se, že můžeme ovládat všechny požadované I/O piny. Mikrokontroléry PIC standardně mají přiřazený k pinům A/D převodník. Dané I/O piny přepneme do jejich primární funkce nastavením registrů A/D převodníku: AD1PCFG = 0xFFFFh a AD1CSSL = 0.

Tab. 4.2: Nastavení nejdůležitějších položek v Configuration Word

Configuration Word	Popis položky (název položky)	Nastavení
CW1	JTAG port (JTAGEN)	Zakázáno
	Výběr programovacích pinů (ICS)	PGEC3/PGED3
	Watchdog (FWDTEN)	Zakázáno
CW2	Dělič kmitočtu (PLLDIV)	Děleno 2
	Automatické zapnutí PLL po startu (PLL96MHz)	Povoleno
	Výběr vnitřního oscilátoru (FNOSC)	RC oscilátor s PLL
	Výstup oscilátoru na I/O pinu RA3 (OSCIOFCN)	Zakázáno
	Výběr hlavního oscilátoru (POSCMD)	Zakázáno
CW3	Výstup sekundárního oscilátoru na I/O pinu RA4 (SOSCSEL)	Zakázáno

#### 4.2.3 Ovládání displeje

Dalším krokem bylo napsání funkcí pro ovládání displeje. Po každé, když je připojeno napájení je zapotřebí provést inicializaci, ta se liší v závislosti na použitém řadiči v displeji. V mém případě je použit S6A0069. Jeho dokumentace je však na velmi

špatné úrovni a nebylo možné podle ní správně naprogramovat proces inicializace sestávající se z pěti instrukcí. Při hledání řešení se ukázalo, že používá stejné instrukce jako nejpoužívanější řadič HD44780 [5]. Potřeba zpoždění během inicializace a pro provedení jednotlivých instrukcí však shodné nebyly. Bylo zapotřebí je tedy odvodit empiricky.

K zobrazení znaku se využívá instrukce pro zápis do CGROM (Character Generator ROM) a instrukce pro zápis do DDRAM (Display Data RAM) [5]. V CGROM jsou uloženy znaky, které displej umí zobrazit. Paměť DDRAM obsahuje znaky, které jsou zobrazeny na příslušné pozici displeje (tabulka 4.3). Pro zobrazení znaku je zapotřebí nejdříve určit adresu v DDRAM a poté adresu CGROM.

Tab. 4.3: Pozice na displeji a odpovídající adresy DDRAM

		Sloupce				
		0	1	...	18	19
Řádky	0	00	01	...	18	19
	1	64	65	...	82	83
	2	20	21	...	38	39
	3	84	85	...	102	103

#### 4.2.4 Komunikace s EEPROM

Vybraná paměť EEPROM komunikuje pomocí rozhraní SPI, které využívá těchto vodičů:

- SCK – hodinový signál.
- SDI – datový vstup.
- SDO – datový výstup.
- SS (nebo CS) – výběr zařízení.

Datová komunikace má pro každý směr vlastní vodič. Vodiče SCK, SDI a SDO jsou společné pro všechna zařízení. Konkrétní zařízení, se kterým chceme komunikovat, se vybírá pomocí logické úrovně na vodiči SS (Slave Select nebo CS, Chip Select), který bývá jeden pro každé zařízení. Komunikace se zařízením se převážně, v závislosti na typu instrukce, skládá ze tří částí: odeslání instrukce, odeslání adresy, odeslání (přijetí) dat. Odeslání adresy se nemusí, v závislosti na instrukci, provádět.

Mikrokontrolér PIC24FJ64GB002 disponuje tzv. PPS (Peripheral Pin Select) [18]. Většina periferních modulů (jako např. SPI modul) není pevně přiřazena k pinům mikrokontroléru, ale je zapotřebí je pomocí registrů přiřadit ke zvoleným pinům. Jaké

registry s k tomu využijí, závisí na konečné funkci pinu, zdali se bude jednat o výstup či o vstup. Každý vstupní pin periferního modulu má svůj registr a každý výstupní pin má své číslo. Dále každý pin mikrokontroléru, ke kterému je možné tyto moduly připojit, má svůj registr a své číslo. Pro přiřazení vstupního pinu, se do registru pinu modulu zapíše číslo pinu mikrokontroléru. Pro přiřazení výstupního pinu, se do registru pinu mikrokontroléru zapíše číslo pinu modulu. I přesto, že pin modulu má daný směr toku dat, je zapotřebí nastavit požadovaný směr toku dat u pinů mikrokontroléru v registrech TRIS příslušných portů. Konkrétní nastavení ukazuje tabulka 4.4. Jelikož je možné přistupovat k jednotlivým bitům či skupinám bitů, jsou uvedeny jen ty. Pin SS se v případě zařízení řídicího komunikaci nenastavuje v PPS registrech, jelikož protější zařízení nevybírání SPI modul. K tomuto účelu je použita standardní funkce I/O portu.

Tab. 4.4:Nastavení I/O pinů pro komunikaci s EEPROM

Registr	Hodnota
TRISB13	1 (vstup pro SDI)
TRISB14	0 (výstup pro SDO)
TRISB15	0 (výstup pro SCK)
TRISB4	0 (výstup pro SS)
SDI1R (registr pinu SDI)	Dh (číslo volitelného pinu mikrokontroleru)
RP14R (registr volitelného pinu mikrokontroleru)	8h (kód pinu SDO)
RP15R (registr volitelného pinu mikrokontroleru)	7h (kód pinu SCK)

Nastavení SPI modulu se provádí pomocí tří registrů SPI1STAT, SPI1CON1 a SPI1CON2 [18]. SPI1CON2 obsahuje nastavení rozšířené vyrovnávací paměti SPI modulu, kterou nevyužívám. Je tedy nastaven na hodnotu 0x0000h. Z registru je zapotřebí nastavit pouze bit SPIEN, kterým se zapíná modul. Zbylé bity jsou příznaky či nastavení rozšířené vyrovnávací paměti. Registr SPI1STAT je nastaven na hodnotu 0x8000h. Nejdůležitější je registr SPI1CON1 (tab.4.5). Je nastaven na hodnotu 0x003Fh. Piny SDO, SCK jsou povoleny a pin SS je zakázán. Registr neobsahuje bit kontrolující funkci SDI. Tento pin se pouze přiřadí pomocí PPS. Bity SMP, CKE, CKP je zapotřebí nastavit podle funkce EEPROM. Jsou tedy nastaveny na vzorkování uprostřed impulsu, klidový stav hodinového signálu je během nízké úrovně a log „1“ odpovídá nástupné hraně. MSTEN je nastaven na Master mode. Poslední jsou předděliče hodinového signálu. V tomto zapojení (tedy při napájecím napětí 2,5 V) je dle manuálu paměť EEPROM pracovat maximálně na frekvenci 10 MHz. Nejbližší možná úroveň pod touto hodnotou je 8 MHz. Při této hodnotě však nebylo možné

komunikovat s pamětí. Ukázalo se, že pokud se využijí děličky kmitočtu SPI modulu, není možné komunikovat s pamětí. Pokud se nevyužijí, hodinový signál má frekvenci 16 MHz. Je to více než by paměť při daném napětí měla zvládnout, avšak komunikace fungovala. Po zobrazení průběhu komunikace na osciloskopu, se při nastavení na 16 MHz zobrazovaly pravoúhlé impulsy s frekvencí skutečně 16 MHz. V případě využití děliček, se zobrazovaly impulsy zdeformované, jako kdyby se nestačil nabít nějaký kondenzátor. Výskyt impulsů se navíc jevil jako náhodný. Důvod, proč děličky kmitočtu u SPI modulu nefungují tak jak by měly se mi nepodařilo zjistit. Nastavení obou děliček je tedy ponechal 1:1, kdy je výsledná frekvence 16 MHz. Paměť a komunikace s ní probíhá naprosto v pořádku a žádný problém se nevyskytl.

Tab. 4.5: Registr SPI1CON1

Bitý registru	Název	Popis
15–13		Neimplementováno
12	DISSCK	Zakázání SCK pinu
11	DISSDO	Zakázání SDO pinu
10	MODE16	Nastavení 8b nebo 16b komunikace
9	SMP	Vzorkování vstupních dat
8	CKE	Změna výstupu na sestupnou nebo nástupnou hranu
7	SSEN	Povolení SS pinu
6	CKP	Úroveň klidového signálu
5	MSTEN	Povolení Master mode
4–2	SPRE	Sekundární dělička; 1:1, ..., 1:8
1–0	PPRE	Hlavní dělička; 1:1, ..., 1:64

#### 4.2.5 Programování komunikace přes USB

Vytvoření kódu pro komunikaci pomocí USB, tak aby splňoval všechny požadavky specifikace, je náročné a tak Microchip zdarma poskytuje knihovny pro nekomerční účely. Jedná se o součást souboru knihoven, který obsahuje všechny knihovny poskytované firmou Microchip. Název balíku a použitá verze je *Microchip Solutions v2010-10-19*. Ke svým knihovnám poskytuje i patřičnou dokumentaci. Jako výchozí je možné použít například [11]. Zde je i uvedeno schéma jednotlivých funkčních vrstev pro hostitelské zařízení. To je uvedeno v tabulce 4.6 spolu s odpovídajícími zdrojovými soubory. Knihovny od Microchipu jsou využity pro spodní tři vrstvy. Knihovna pro práci se souborovým systémem je rozebrána v kapitole 4.3.6 a uživatelskému rozhraní se zabývá kapitola 4.3.7. Pro koncové zařízení jsou horní dvě vrstvy totožné a podpora SCSI příkazů je součástí ovladače MSD. Je zde tedy rozdíl pouze v použití knihoven *usb\_function\_msd.c* a *usb\_device.c*. Veškeré nastavení knihoven pro komunikaci přes

USB zajistí soubory vygenerované nástrojem *USBConfig.exe*, který je součástí souboru knihoven.

Konfigurace knihoven s využitím přiloženého nástroje je jednoduchá. V záložce *Main* je zapotřebí zvolit *PIC24/dsPIC* z nabídky *Target Device Family*, zakliknout *USB Dual Role* a *Ping-Pong on All Endpoints*. Ačkoliv USB specifikace zaměnila název *Dual Role* za *On The Go* v roce 2006 a již více se o *Dual Role* (vyjma *WUSB*) nikde nezmiňuje, tento nástroj umožňuje vybrat z obou. Jediný rozdíl v nastavení knihoven je u povolení *NHP* a *SRP* protokolů u volby *OTG*. Ty se však při komunikaci s *USB* diskem nebo počítačem nevyužívají. Proto zvolíme možnost *Dual Role*. *Ping Pong mode* je pouze záležitostí hardwarové funkce *USB* modulu u mikrokontrolérů od *Microchipu* zvyšující výkon tohoto modulu. Nesouvisí tedy nijak se specifikací *USB*. Další potřebné nastavení se nachází v záložce *TPL*. Zde je zapotřebí kliknout na *Support via Class ID* a zadat *Class ID=0x08*, *SubClass ID=0x06* a *Protocol=0x50*. Dále pak zvolit v *Client Driver* možnost *Mass Storage* a kliknout na *Add to TPL*. Poslední potřebnou záložkou je *Mass Storage*. Zde zaškrtneme možnosti *Mass Storage Function is used in Peripheral mode*, *Mass Storage Client is used in Host mode* a zvolíme *Maximum Logical Unit(LUN)=1*. Tím je nastavování ukončeno a kliknutím na *Generate* se vytvoří požadované konfigurační soubory. Pro správnou funkci je ještě zapotřebí vytvořit strukturu, která slouží „propojení“ knihoven *Microchipu* a mnou vytvořené knihovny pro práci s *EEPROM*:

```
LUN_FUNCTIONS LUN[MAX_LUN + 1] =
{
    {
        &EEPROM_MediaInitialize,
        &EEPROM_ReadCapacity,
        &EEPROM_ReadSectorSize,
        &EEPROM_MediaDetect,
        &EEPROM_SectorRead,
        &EEPROM_WriteProtectState,
        &EEPROM_SectorWrite
    }
};
```

Poté je ještě zapotřebí definovat odezvu na *INQUIRY* dotaz. Je možné použít vzorový příklad z [AN1189] s tím, že je zapotřebí změnit médium na nevyjímatelné a je vhodné vložit vlastní popis výrobce a zařízení. Tomu je dále zapotřebí upravit položku s velikostí odezvy a v hlavičkovém souboru *usb\_function\_msd.h* změnit velikostí polí s těmito popisy v definici struktury *InquiryResponse*. Výsledná struktura poté vypadá:

```

const ROM InquiryResponse inq_resp = {
    0x00,          // peripheral device is connected, direct access block device
    0x00,          // nonremovable media
    0x04,          // SPC-2 commands
    0x02,          // response is in format specified by SPC-2
    0x1C,          // InquiryResponse lenit n-4 = 34-4=28= 0x1C
    0x00,          // sccs etc.
    0x00,
    0x00,
    {'K','I','I','r',' ',' ','T',' ',' '},
    },
    {'M','a','s','t','e','r',' ','T','h','e','s','i','s'},
    },
    {'0','0','0','0'}
    }
};

```

Tab. 4.6: Funkční vrstvy hostitelského zařízení dle [11] a odpovídající knihovny

Funkční vrstva	Odpovídající knihovna
Aplikační vrstva	<i>main.c UserInterface.c</i>
Podpora souborového systému	<i>ff.c, diskio.c</i>
Podpora SCSI příkazů	<i>usb_host_msd_scsi.c</i>
Ovladač MSD	<i>usb_host_msd.c</i>
Ovladač USB	<i>usb_host.c</i>

To je vše, co je zapotřebí udělat pro použití knihoven od Microchipu. Jelikož výsledná velikost kódu již překračuje kapacitu paměti programu, je zapotřebí nastavit kompilér pro optimalizaci kódu z hlediska velikosti programu. To se provede v MPLABu v *Project*→*Build Options*...→*Project*. Nyní vybereme záložku *MPLAB C30, Categories: Optimization* a zvolíme v grafu možnost č. 5.

Nyní když funguje komunikace s USB diskem, je možné otestovat proudové posílení napájení USB sběrnice. Pro větší odběr proudu byl mezi  $V_{USB}$  a  $GND$  připojen rezistor. Komunikace již však neprobíhala. Ukázala se zásadní chyby v návrhu schématu. Na vývodu *source* unipolárního tranzistoru je uvažováno nulové napětí. Ve skutečnosti je tam však  $V_{USB} = 5\text{ V}$  a tranzistor se nikdy neotevře. Jistým řešením tohoto problému je použití nábojové pumpy, která bude schopná dodat požadovaný proud. Kvůli tomu by však bylo zapotřebí vyrobit novou desku plošného spoje. Schéma tedy zůstalo nezměněné a zařízení je schopno dodat pouze základní napájecí proud 100 mA. Ten je však pro USB Flash disky dostačující.



#### 4.2.6 Práce se souborovým systémem

Knihovny pro práci jsou opět velmi obsáhlé a opět je nabízí Microchip. Poskytované knihovny mají však jednu velkou nevýhodu a tou je „neznalost“ LFN (Long File Name). Je zde tedy využita knihovna třetí strany a to, velmi často používaná, knihovna FatFS. Její výhoda oproti té od Microchipu je především v možnosti pracovat s LFN a při stejném nastavení zabírá podstatně méně paměti programu.

Oproti výchozímu nastavení je zde zapotřebí změnit jenom pár položek, které jsou v tab. 4.7. Položka `_MAX_LFN` definuje maximální velikost jména (specifikace dovoluje jména dlouhá až 255 znaků), `_VOLUMES` určuje počet médií, se kterými se bude pracovat (USB disk a EEPROM), a `_FS_SHARE` říká, kolik souborů je možné otevřít zároveň (pro kopírování je zapotřebí mít otevřený zdrojový i cílový soubor).

Dále je zapotřebí opět propojit knihovnu s nižšími funkčními vrstvami zajišťující vlastní přenos dat na médium. Knihovna volá funkce pro přístup k médiu, které obsahuje pouze konstrukci switch s parametrem požadovaného média. Musí se tedy doplnit pouze volání funkcí pro přístup k médiím (tab. 4.8).

Tab. 4.7: Nastavení FatFS

Položka	Nastavení
<code>_CODE_PAGE</code>	1250
<code>_USE_LFN</code>	1
<code>_MAX_LFN</code>	255
<code>_VOLUMES</code>	2
<code>_FS_SHARE</code>	2

Tab. 4.8: Asociace knihoven pro práci s FatFS

Funkce FatFS	Funkce pro přístup k USB disku, EEPROM
<code>disk_initialize</code>	USBHostMSDSCSIMediaInitialize, EEPROM se inicializuje při inicializaci celého zařízení
<code>disk_status</code>	USBHostMSDSCSIMediaDetect, EEPROM je stále připojená
<code>disk_read</code>	USBHostMSDSCSISectorRead, FatFS_EEPROM_SectorRead
<code>disk_write</code>	USBHostMSDSCSISectorWrite, FatFS_EEPROM_SectorWrite

Knihovny pro práci se souborovým systémem obvykle umožňují naformátování média. Při použití knihovny FatFS se ovšem nepodařilo vytvořit na EEPROM FAT12 ani FAT16. S použitím knihovny od Microchip se to podařilo. Ovšem jak se s postupem času měnil kód, tak již nadále nebylo možné naformátovat EEPROM ani pomocí

knihovny od Microchipu. Příčinu tohoto problému se nepodařilo najít a tak byl vytvořen program speciálně jen pro formátování paměti. V případě FAT12 se nejedná o mnoho hodnot a dají se celkem snadno vyčíst přímo z knihovny Microchipu. Pro naformátování EEPROM je tedy zapotřebí použít projekt na přiloženém CD.

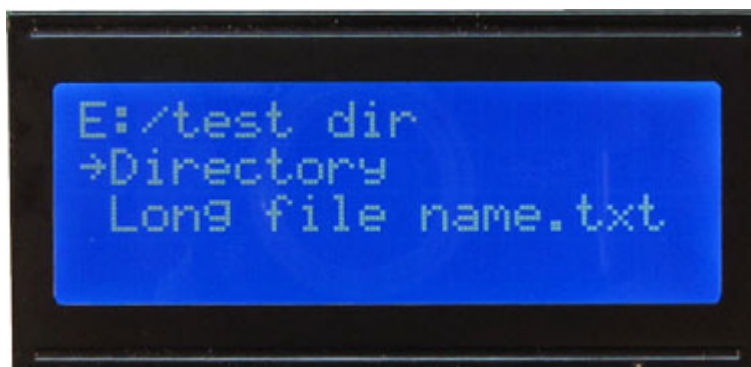
#### 4.2.7 Uživatelské rozhraní

U rozložení informací na displeji není moc možností. Na prvním řádku se nachází cesta k aktuálnímu adresáři a zbylé tři řádky slouží pro výpis jeho obsahu (obr. 4.3). Programování této části je již problematictější. Vlastní procházení položek adresáře se provádí opakovaným voláním funkce `f_readdir`. Ta vrací strukturu s informacemi o jednotlivých položkách. Problém je s jejich uchováním. Jsou tři možnosti, jak procházení souborů implementovat.

První byla načtení celého obsahu adresáře a jeho uložení do operační paměti. Pro tento způsob však není k dispozici dostatečná kapacita paměť dat.

Další možnost je vytvoření indexového listu. Ten by se vytvářel při každém vytvoření složky. Obsahoval by první dvě nebo tři první písmena jména, odsazení v sektoru a informaci zda se jedná o adresář či soubor. Při jeho vytváření by se nová položka porovnávala s již uloženými písmeny a v případě shody by se načetla přímo daná položka, se kterou se má porovnat. Ta se načte na základě uloženého odsazení. Když by bylo zapotřebí posunout zobrazovaný obsah, jednoduše by se opět na základě odsazení načetla další položka. Kámen úrazu byl však právě v odsazení, na kterém byla tato úvaha založena. Ukázalo se, že nefunguje správně. Při jeho zadání se začal číst obsah adresáře od začátku. Jako další problém této metody se ukázala stále velká náročnost na paměť a to tentokrát i na paměť programu.

Byla tedy zvolena třetí možnost. Je založená na prohledání celého obsahu adresáře při jeho posunutí na displeji. Jelikož uchovává pouze právě využívané informace o položkách, je nejméně náročná na paměť dat. Algoritmus v tomto případě je nejjednodušší a tak i náročnost na paměť programu je minimální. V případě mnoha položek ve složce však vyžaduje úměrné množství načítání a porovnávání. Je tedy časově nejnáročnější. V praxi se ukázalo, že do třiceti položek není zaznamenatelné zpoždění. Přibližně u sedmdesáti položek bylo již posouvání nepříjemně pomalé. Je to však jediná možnost, která splňuje požadavky na paměťovou náročnost.



Obr. 4.3: Procházení souborů

Další funkce jsou dostupné z menu (obr. 4.4) a jsou to:

- Přepnutí aktuální paměti (E:→U:).
- Vytvoření nové složky (Make dir).
- Zobrazení atribut aktuální položky (Show att.).
- Smazání aktuální položky (Delete).
- Kopírování souboru (Copy file).



Obr. 4.4: Nabídka menu

U přepínání paměti značí písmeno „E“ paměť EEPROM a „U“ USB disk. Zadávání jména nového adresáře (obr. 4.5) se provádí kurzorovými tlačítky. Jelikož se musí „proklikat“ v nejhorším případě (písmeno uprostřed abecedy) polovina abecedy (je možné procházet písmena i od konce), je zadávání písmen pro urychlení omezeno jen na velká písmena. Pokud se zvolí nabídka pro zobrazení atribut, zobrazí se obrazovka na obrázku 4.6. Jednotlivé položky značí:

- RDO – pouze pro čtení.

- HID – skrytý soubor.
- ARC – archiv.
- DIR – adresář.
- SYS – systémový soubor.



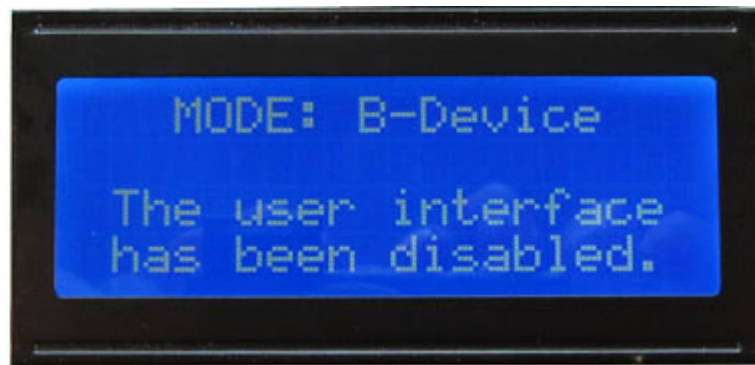
Obr. 4.5: Vytváření nového adresáře

Pokud se zvolí položka kopírování souboru, nabídka menu pak bude místo ní obsahovat položku vložit (Paste). Pokud chceme pro kopírování zvolit jiný soubor, podržíme tlačítko Enter nad položkou vložit po dobu 2 s a ta se změní zpátky na Copy file.



Obr. 4.6: Atributy souboru

Takto se chová uživatelské rozhraní, dokud není zařízení v režimu B-Device připojeno k hostitelskému zařízení. Pokud se tak stane je uživatelské rozhraní zakázáno (obr 4.7). To má jednoduchý důvod. Počítačové programy nepředpokládají, že by mohl někdo jiný měnit soubory a složky na připojeném disku. Pokud by došlo k nějaké změně, program to nezaregistruje.



Obr. 4.7: Režim B-device

## 5 Závěr

Výsledkem této diplomové práce je zařízení demonstrujících schopností USB OTG s využitím třídy MassStorage.

Pokud je toto demonstrační zařízení připojeno k hostitelskému zařízení (obvykle osobní počítač), funguje stejně jako flash disk a je možné procházet souborový systém na vestavěné paměti.

Pokud není demonstrační zařízení připojeno k hostitelskému zařízení, umožňuje, v rámci vestavěné paměti či připojeného flash disku, základní operace se souborovým systémem. Jedná se o vytváření a mazání složky, zobrazení atributů souboru či jeho kopírování.

Jako interní paměť zařízení byla vybrána paměť typu EEPROM. Ta má kapacitu pouze několik desítek kB. Oproti paměťovým kartám typu SD, jejichž implementace by byla stejně jednoduchá a mají mnohonásobně vyšší kapacitu, má EEPROM výhodu v ceně. Jelikož zařízení slouží pouze k demonstraci, malá kapacita není omezující.

Při návrhu a následné realizaci zařízení se vyskytlo několik problémů (např. dostupnost micro USB konektorů a potřebné kabeláže, či chyby mikrokontroléru). Všechny problémy se však podařilo vyřešit a zařízení funguje správně.

Ačkoliv zařízení demonstruje požadované schopnosti USB OTG, z praktického hlediska je využitelné jen jako součást většího celku, například mobilního telefonu, kde se také využívá. Jelikož je schválená univerzální nabíječka pro telefony, využívající právě micro USB konektor, má tento standard, dle mého názoru, možnost se dostat do podvědomí širší populace.

## Seznam použitých zdrojů

- [1] Anderson, D. *USB System Architecture (USB 2.0)*. 2001, ISBN: 0-201-46137-4
- [2] Anton, B. *A History of the USB Standard* [online]. 2009. URL:  
<http://ezinearticles.com/?A-History-of-the-USB-Standard&id=2532259>
- [3] Axelson, J. *USB komplete: Everything you need to develop custom USB peripherals*. 4. vydání. 2005, ISBN: 978-1-931448-02-4
- [4] Beyond Logic. *USB in a NutShell* [online]. 17. 9. 2010. URL:  
<http://www.beyondlogic.org/usbnutshell/usb1.shtml>
- [5] Hitachi. *HD44780 Data sheet* [online]. 2. 9. 2009. URL:  
<http://www.sparkfun.com/datasheets/LCD/HD44780.pdf>
- [6] Kolic, R. *Wireless USB Brings Greater Convenience and Mobility to Device* [online]. 21. 2. 2004. URL:  
<http://www.deviceforge.com/articles/AT9015145687.html>
- [7] Malý, M. *USB 2.0 – díl 1* [online]. 22. 2. 2005. URL:  
<http://hw.cz/Rozhrani/ART1232-USB-2.0---dil-1.html>
- [8] Malý, M. *USB 2.0 – díl 2* [online]. 10. 3. 2005. URL:  
<http://hw.cz/Rozhrani/ART1244-USB-2.0---dil-2.html>
- [9] Malý, M. *USB 2.0 – díl 3* [online]. 19. 4. 2005. URL:  
<http://hw.cz/Rozhrani/ART1264-USB-2.0---Typy-a-formaty-prenosu.html>
- [10] Microchip. *25LC512 Data sheet* [online]. 20. 6. 2010. URL:  
<http://ww1.microchip.com/downloads/en/DeviceDoc/22065C.pdf>
- [11] Microchip. *AN1140: USB Embedded Host Stack* [online]. 3.4.2008. URL:  
[http://ww1.microchip.com/downloads/en/AppNotes/USB\\_Host\\_Stack\\_01140a.pdf](http://ww1.microchip.com/downloads/en/AppNotes/USB_Host_Stack_01140a.pdf)
- [12] Microchip. *AN1142: USB Mass Storage Class on an Embedded Host* [online]. 2. 5. 2008 URL:  
<http://ww1.microchip.com/downloads/en/AppNotes/01142A.pdf>
- [13] Microchip. *AN1145: Using a USB Flash Drive with an Embedded Host* [online]. 30. 4. 2008. URL:  
<http://ww1.microchip.com/downloads/en/AppNotes/01145b.pdf>
- [14] Microchip. *AN1169: USB Mass Storage Class on an Embedded Device* [online]. URL:  
<http://ww1.microchip.com/downloads/en/AppNotes/01169a.pdf>
- [15] Microchip. *AN1189: Implementing a Mass Storage Device Using the Microchip USB Device Firmware Framework* [online]. 20. 4. 2008. URL:  
<http://ww1.microchip.com/downloads/en/AppNotes/01189a.pdf>
- [16] Microchip. *MCP1253 Data sheet* [online]. 15. 11. 2002. URL:

- <http://ww1.microchip.com/downloads/en/DeviceDoc/21752a.pdf>
- [17] Microchip. *PIC16F505 Data Sheet* [online]. 8. 3. 209. URL:  
<http://ww1.microchip.com/downloads/en/devicedoc/41236d.pdf>
- [18] Microchip. *PIC24FJ64GB004 Family Data Sheet* [online]. 15. 7. 2010. URL:  
<http://ww1.microchip.com/downloads/en/DeviceDoc/39940d.pdf>
- [19] Microchip. *PIC24F Family Reference Manual, Sect. 06 Oscillator* [online].  
URL:  
<http://ww1.microchip.com/downloads/en/DeviceDoc/39700c.pdf>
- [20] Microchip: *PIC24F Reference Manuals. Section 6: Oscillatori* [online].  
10. 9. 2009. URL:  
<http://ww1.microchip.com/downloads/en/DeviceDoc/39700c.pdf>
- [21] NTFS. *The FAT File Systems. FAT32 FAT16 FAT12* [online]. 20. 3. 1998. URL:  
<http://www.ntfs.com/fat-systems.htm>
- [22] Raystar Optronics. *RC2004a Data sheet* [online]. 12. 3. 2003. URL:  
[http://www.tme.eu/dok/06\\_optoelektronika/rc2004a-biw-esx.pdf](http://www.tme.eu/dok/06_optoelektronika/rc2004a-biw-esx.pdf)
- [23] Seagate Technology. *SCSI Commands Reference Manual* [online]. Rev. A. 5. 2.  
2006. URL:  
<http://www.seagate.com/staticfiles/support/disc/manuals/scsi/100293068a.pdf>
- [24] USB – IF. *Universal Serial Bus Specification 2.0* [online]. 27. 4. 2000. URL:  
[http://www.usb.org/developers/docs/usb\\_20\\_081810.zip](http://www.usb.org/developers/docs/usb_20_081810.zip)
- [25] USB – IF. *Universal Serial Bus Specification 3.0* [online]. 12. 11. 2008. URL:  
[http://www.usb.org/developers/docs/usb\\_30\\_spec\\_060910.zip](http://www.usb.org/developers/docs/usb_30_spec_060910.zip)
- [26] Wikipedia. *Universal Serial Bus* [online]. 30. 10. 2010. URL:  
[http://en.wikipedia.org/wiki/Universal\\_Serial\\_Bus](http://en.wikipedia.org/wiki/Universal_Serial_Bus)
- [27] Wikipedia. *USB On-The-Go* [online]. 15. 11. 2010. URL:  
[http://en.wikipedia.org/wiki/USB\\_On-The-Go](http://en.wikipedia.org/wiki/USB_On-The-Go)
- [28] Wikipedia. *Wireless USB* [online]. 20. 10. 2010. URL:  
[http://en.wikipedia.org/wiki/Wireless\\_USB](http://en.wikipedia.org/wiki/Wireless_USB)



## Seznam použitých zkratek a symbolů

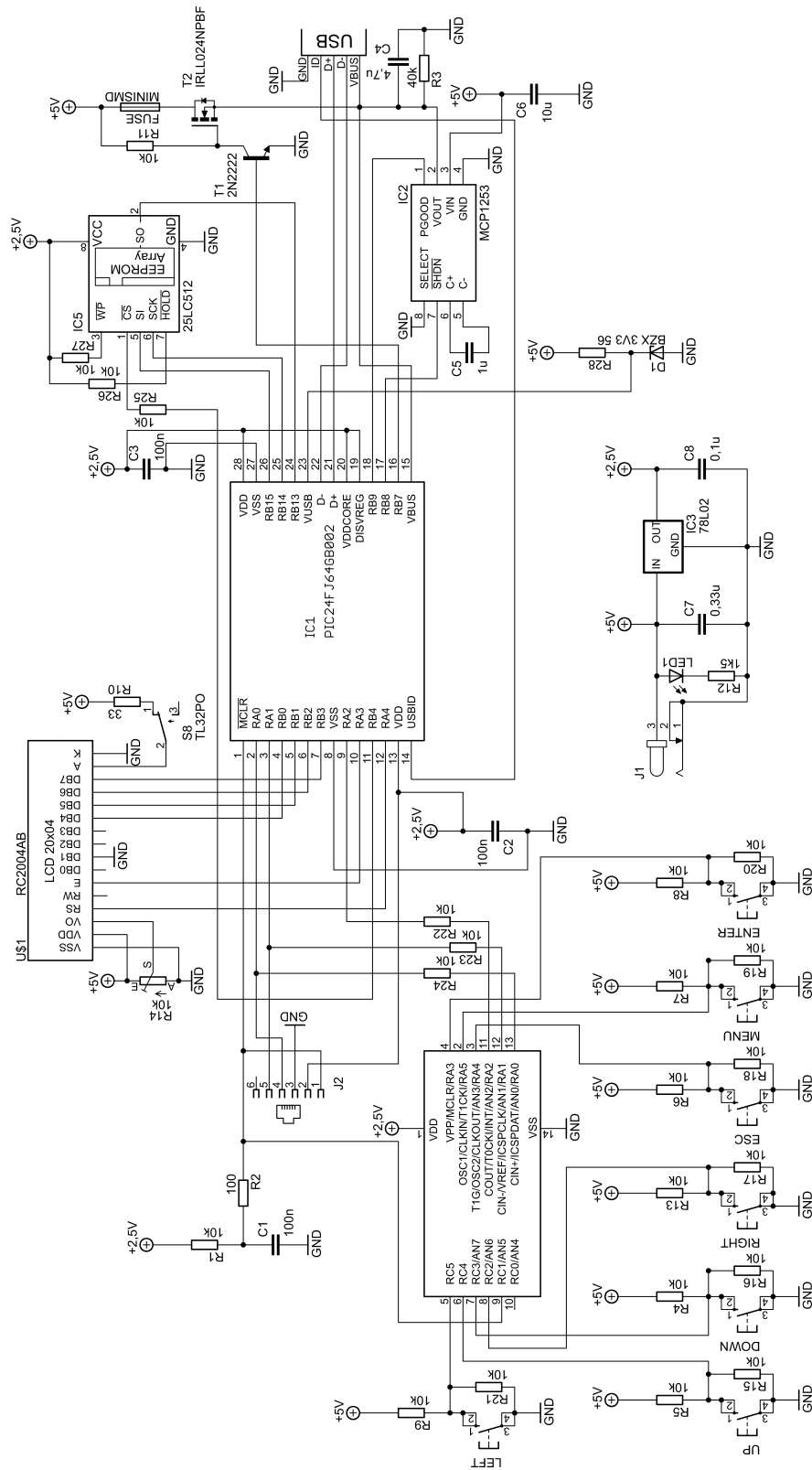
ADP	protokol detekce nově připojeného zařízení
ARC	atribut souboru – archiv
BBB	pouze hromadné přenosy
CBI	řídící, hromadné a přerušovací přenosy
CBW	blok příkazů
CGRAM	paměť znaků displeje
CRC	cyklický redundantní součet
CSW	výsledky přenosu
D+	kladný datový vodič
D-	záporný datový vodič
DIR	atribut souboru – adresář
DRD	zařízení dvojí role
DDRAM	paměť dat displeje
HID	atribut souboru – skrytý
HNP	protokol vyjednávání funkce zařízení
I/O	vstupně výstupní
ID	identifikační číslo
IRP	paket s požadavkem na vstup či výstup dat
LFN	dlouhé jméno
LUN	číslo logické jednotky
NPN	bipolární struktura tranzistoru
NRZI	invertované kódování bez návratu k nule
OTG	bod–bod
RDO	atribut souboru – pouze pro čtení
SCK	hodinový výstup pro sériový přenos
SDI	vstup sériových dat
SDO	výstup sériových dat
SMD	povrchová montáž
SPI	typ sériového přenosu
SRP	protokol řízení napájení sběrnice
SYS	atribut souboru – systémový
USB	univerzální sériová sběrnice
WUSB	bezdrátová univerzální sběrnice

## **Seznam příloh**

- A USB OTG zařízení
  - A.1 Schéma zařízení
  - A.2 Seznam součástí
  - A.3 DPS
  - A.4 DPS – osazovací plán
  - A.5 Fotografie OTG zařízení
- B Obsah přiloženého CD

# A USB OTG zařízení

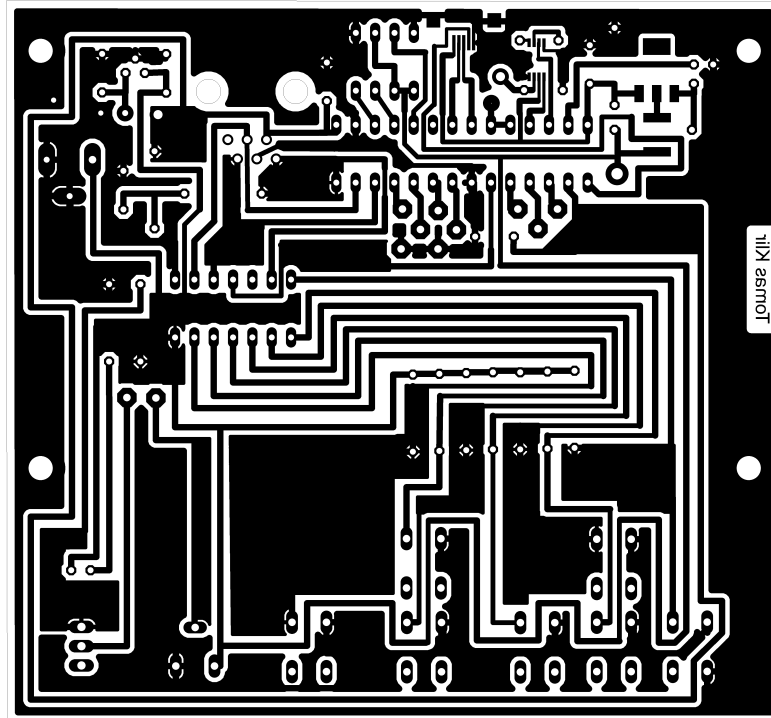
## A.1 Schéma zapojení



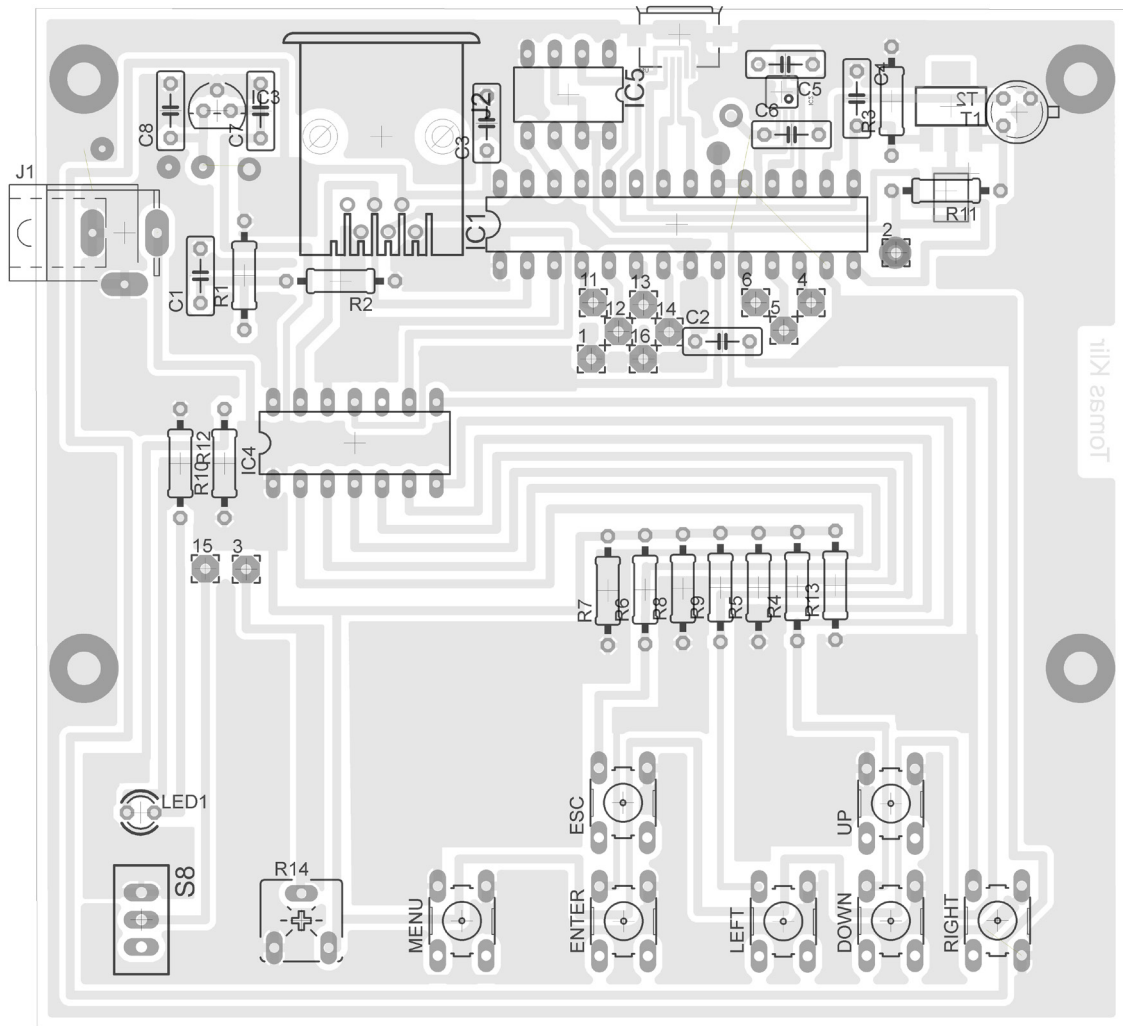
## A.2 Seznam součástek

Označení	Kusů	Hodnota/Název	Typ součástky (pouzdro)
C1, C2, C3, C8	4	100 nF	Kondenzátor (RM 5 mm)
C4	1	4,7 uF	Kondenzátor (RM 5 mm)
C5	1	1 uF	Kondenzátor (RM 5 mm)
C6	1	10 uF	Kondenzátor (RM 5 mm)
C7	1	330 nF	Kondenzátor (RM 5mm)
D1	1	BZX 3V3	Zenerova dioda
DOWN, UP, RIGHT, LEFT, ENTER, ESC, MENU	7	TC-0104	Tlačítko
FUSE	1	MINISMDC050 F-2	PTC pojistka
IC1	1	PIC24DJ64GB002	Mikrokontrolér
IC2	1	MCP1253-33X50IS	Nábojová pumpa
IC3	1	78L02	Stabilizátor napětí (TO92)
IC4	1	PIC16F505	Mikrokontrolér
IC5	1	25LC512	EEPROM (DIL8)
J1	1	FC68148	Napájecí konektor (5,5/2,1 mm)
J2	1	RJ12GLP-AMP	Konektor (RJ12)
LED1	1	Zelená	LED dioda (3 mm)
R1, R4, R5, R6, R7, R8, R9, R11, R13, R15, R16, R17, R18, R19, R20, R21, R22, R23, R24, R25, R26, R27	22	10 k $\Omega$	Rezistor
R14	1	10 k $\Omega$	Trimr (ležatý, 6 mm)
R2	1	100 $\Omega$	Rezistor
R3	1	40 k $\Omega$	Rezistor
R10	1	33 $\Omega$	Rezistor
R12	1	1,5 k $\Omega$	Rezistor
R28	1	56 $\Omega$	Rezistor
S8	1	P-ESP101	Přepínač
T1	1	2N2222	Tranzistor (TO92)
T2	1	IRLL024NPBF	Tranzistor
U $\S$ 2	1	USB micro AB	USB konektor

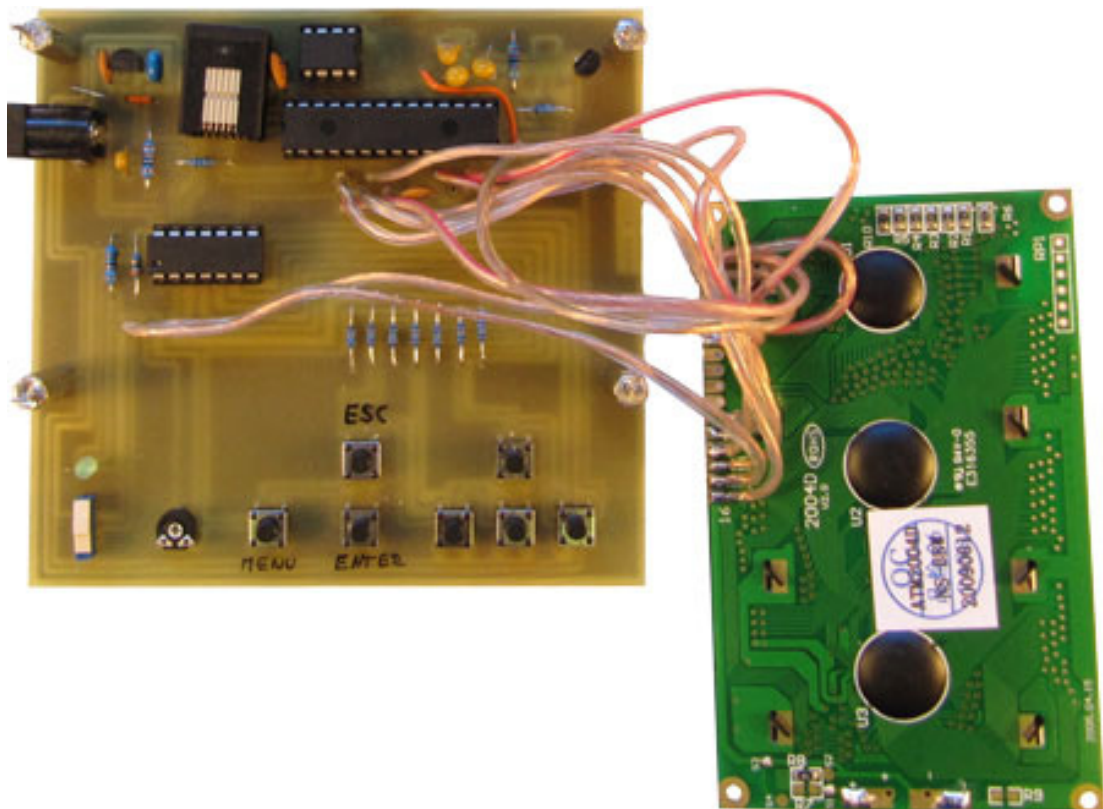
### A.3 DPS



## A.4 DPS – osazovací plán



## A.5 Fotografie OTG zařízení



## **B Obsah přiloženého CD**

/source projects.zip	zdrojové projekty pro MPLAB
/xklirt00.pdf	elektronická verze diplomové práce