

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta – Katedra fyziky

Vývoj převodníku USB na paralelní sběrnici

Bakalářská práce

Anotace:

Cílem této bakalářské práce je navrhnout zařízení, které převede signály z USB sběrnice na 16bit sběrnici. Následující text pojednává o jednotlivých problémech s komunikací na sběrnici USB a zvolených řešeních jejího převodu. Dále se pak zabývá konstrukcí převodníku samotného, základním softwarovým vybavením pro tento převodník, aby jej bylo možné ovládat pomocí PC.

Abstract:

The aim of this baccalaureate work is to develop device that converts the USB bus to 16 bit parallel bus. Following text deals with unique problems of communication on USB bus and select solutions of transmission. Furthermore it engages in construction of the own converter, and development of the basic software equipment for communication and control by the PC.

Prohlášení:

Prohlašuji, že předloženou práci jsem vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW) uvedené v příloze.

Prohlašuji, že v souladu s § 47 zákona č. 111/1998 Sb. V platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

V Českých Budějovicích dne 10. 10. 2008

.....

Poděkování:

Děkuji touto formou Doc. PaedDr. Petru Adámkovi, Ph.D. za cenné rady, odborné vedení a pomoc při vypracování této bakalářské práce.

Dále bych chtěl poděkovat mým blízkým za podporu.

OBSAH

1. Úvod a cíle práce	7
2. Úvod do USB.....	7
2.1 Základní vlastnosti USB	7
2.2 Konektory a kabely	7
2.3 Verze USB a přenosové rychlosti	8
2.4 Definice rychlosti podle zapojení a napájení.....	9
2.4.1 Napájení	10
2.4.2 Stínění	11
2.5 Architektura USB	11
2.6 Přenos dat.....	12
2.7 Typy přenosů dat na USB	13
2.8 Rozpoznání zařízení.....	13
2.9 Deskriptory zařízení.....	14
3. Analýza konverze USB na paralelní sběrnici	15
3.1 Mikrokontroléry s hardwarovou implementací USB.....	15
3.2 Softwarová implementace USB.....	15
3.2.1. Vlastnosti softwarové implementace USB	16
3.2.2 Hardware.....	16
3.2.3 Vzorové konstrukce	17
3.2.4. Další softwarové implementace	17
3.3 Samostatné USB řadiče	17
3.3.1 FTDI chip.....	18
3.3.2 Philips	19
3.3.3 Cypress.....	20
4. Návrh a realizace sériově-paralelního převodníku	20
4.1 Zvolené řešení převodníku.....	20
4.2 Konstrukce převodníku.....	21
4.2.1 Popis zapojení	21
4.3 Integrovaný obvod FT 232RL	23
4.3.1 Hardwarové vlastnosti	23
4.3.2 Blokové schéma obvodu	23
4.3.3 Popis vývodů FT 232RL.....	26

4.3.4 Režim BITBANG	26
4.4 Sběrnice I ² C	28
4.4.1 Struktura sběrnice	28
4.4.2 Přenos bitů	28
4.4.3 Start a Stop.....	28
4.4.4 Potvrzení přenosu ACK.....	29
4.4.5 Adresování na sběrnici I ² C	29
4.4.6 Adresy I ² C zařízení.....	30
4.4.7 Elektrické parametry sběrnice I ² C	30
4.5 Integrovaný obvod MCP 23016.....	31
4.5.1 Základní vlastnosti MCP 23016	31
4.5.2 Generátor hodin	31
4.5.3 Adresa obvodu	32
4.5.4 Instrukce.....	32
4.5.5 Registry	32
4.5.6 Logika přerušení INT.....	34
4.5.7 Zápis do registrů	35
4.5.8 Čtení z registrů.....	36
4.6 Realizace převodníku.....	37
5. Softwarové řešení komunikace s vyvinutým modulem.....	38
5.1 Naprogramování EEPROM obvodu FT 232R.....	39
5.2 Třída TFT232.....	39
5.3 Třída TI2C	40
5.4 Třída TMCP23016.....	41
5.5 Testovací aplikace DIPLOMKA 1	42
5.6 Testovací aplikace DIPLOMKA 2	43
6. Testování.....	44
6.1 Testy hardwaru	44
6.2 Testy komunikace	44
7. Možné praktické aplikace vyvinutého modulu.....	45
7.1 Příklady praktického využití modulu.....	46
7.1.1 Příklad 1 – řízení teploty v peci	46
7.1.2 Příklad 2 - Řízení trojosého obráběcího stroje.....	46
8. Závěr	48

Použitá Literatura.....	49
Příloha A – Přehled mikrokontrolérů s USB	50
Příloha B – Obrázky osazení plošného spoje.....	51
Příloha C – Seznam obrázků.....	52

1. Úvod a cíle práce

Hlavním cílem této bakalářské práce je vytvořit převodník z USB sběrnice na 16-ti bitovou (nadále 16bit) paralelní sběrnici. USB rozhraní se během posledních let stalo zcela běžnou součástí spotřební elektroniky připojitelné k počítači a již téměř vytlačilo klasický sériový port RS 232 a někde dokonce i paralelní port. Koupit dnes notebook s klasickým sériovým rozhraním začíná být problém, a proto jsou nuceni i výrobci a uživatelé speciálních aplikací postupně přecházet na USB. Nové typy rozhraní jsou lepší jak v rychlosti, tak i ve způsobu připojování. U rozhraní USB se již nemusí vypínat počítač, když potřebujeme připojit nové zařízení a díky technologii Plug&Play si operační systém sám najde potřebné ovladače. Toto nahrazování s sebou nese určité problémy při potřebě ovládat vlastní zařízení. Dříve bylo možné ovládat přímo jednotlivé vodiče na LPT portu, zatím co na USB je komunikace prováděna sériově po 2 vodičích.

Dalším cílem je osvětlení rozsáhlé problematiky převodu sběrnice USB na 16bit paralelní sběrnici. Finálním výstupem je realizace tohoto převodníku a jednoduché příklady softwarového řešení ovládání na osobním počítači PC.

2. Úvod do USB



USB – Universal Serial Bus – Univerzální sériová sběrnice. V této kapitole budou vysvětleny základní vlastnosti a pojmy spojené s USB.

2.1 Základní vlastnosti USB

- ✚ sériové rozhraní
- ✚ možnost napájení zařízení přímo z konektoru - normálně můžeme odebírat 100 mA, ve zvláštních případech možno odebírat až 500 mA
- ✚ přenosové rychlosti: 1,5 Mb/s, 12 Mb/s, 480 Mb/s
- ✚ velký počet zařízení připojitelných na jeden výstup PC – až 127 při použití HUB (rozbočovačů)
- ✚ podpora připojování a odpojování zařízení při provozu - Plug&Play
- ✚ velká podpora operačními systémy (Win, Linux, Mac.....)

2.2 Konektory a kabely

USB rozhraní používá dva typy konektorů. Plochý konektor „typ A“ (viz obr. 2.1) je dnes obsažen prakticky na každém PC v minimálně 2 konektorech (některé základní desky mají integrován rovnou USB hub, který obsahuje až 8 portů přímo v PC). Druhý konektor

„typ B“ (viz obr. 2.1) je určen pro periferní zařízení, čímž je zároveň definován standard propojovacího kabelu A-B. Menší zařízení jako jsou myši nebo klávesnice používají pevně připojený kabel zakončený konektorem typu A.

Kabel typu A-A se používá jen zřídka na propojení dvou počítačů (Laplink®). Vyžaduje to však speciální hardware mezi počítači.

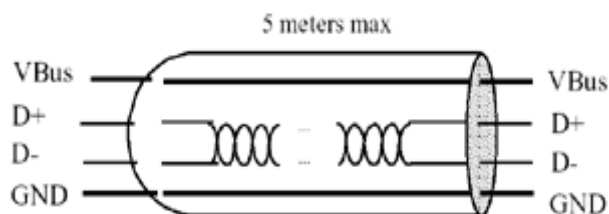


Obr. 2.1 USB konektory

Pro využití maximální přenosové rychlosti (12 Mbps) může být kabel dlouhý max. 5 metrů, přičemž musí být stíněný a kroucený. Pro přenosové rychlosti do 1,5 Mb/s nemusí být datové vodiče zkrouceny, nemusí být použito stínění a maximální délka takového kabelu je 3 m. Jednotková impedance kabelu je v obou případech 90 Ω .

Značení, popis a barvy jednotlivých vodičů jsou uvedeny v tabulce 2.1

Tab. 2.1 Kabel pro USB



Obr. 2.2 Kabel pro USB

Pin	Jméno	Barva	Popis
1	Vbus	Red	+5 V
2	-D	White	Data
3	+D	Green	Data
4	GND	Black	Ground

2.3 Verze USB a přenosové rychlosti

USB zařízení pracují ve verzi 1.1 a ve verzi 2.0. Tyto dvě verze se od sebe odlišují především přenosovými rychlostmi. Pro USB verze 2.0 byla doplněna nejrychlejší vrstva. Všechny verze propojení mohou být použity a provozovány současně pro připojení různých typů periférií k jednomu počítači. Uvedené verze se od sebe liší jak provedením kabelu, tak elektrickými parametry rozhraní připojeného zařízení (viz kapitola 2.4).

Přenosové rychlosti uvedené v tabulce 2.2 platí jen pro jedno zařízení. Je-li k počítači (na jeden kořenový HUB) připojeno více zařízení, rozděluje se přenosová rychlost mezi tato jednotlivá zařízení.

Tab. 2.2 přenosové rychlosti USB

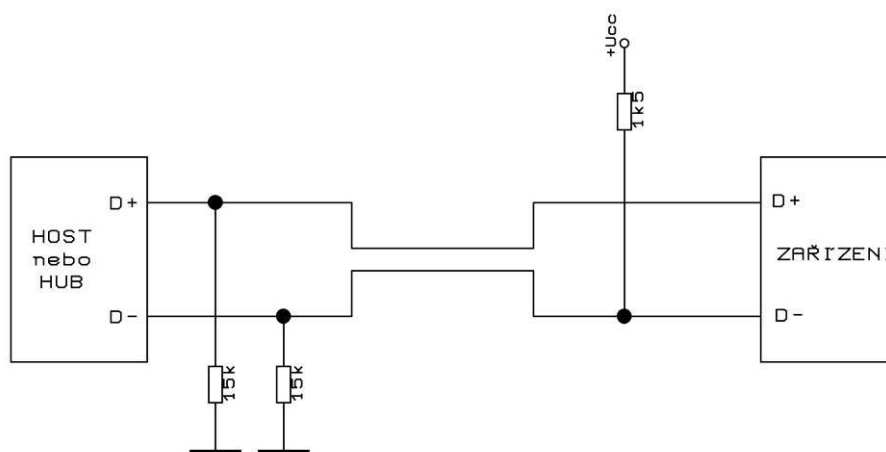
Rychlost	Přenosová rychlost	Standard
Low speed	1,5 Mb/s	USB 1.1/2.0
Full speed	12 Mb/s	USB 1.1/2.0
High speed	480 Mb/s	USB 2.0

2.4 Definice rychlosti podle zapojení a napájení.

Definici rychlosti určuje samo koncové zařízení, kdy je D+ nebo D- připojeno k 3,3 V.

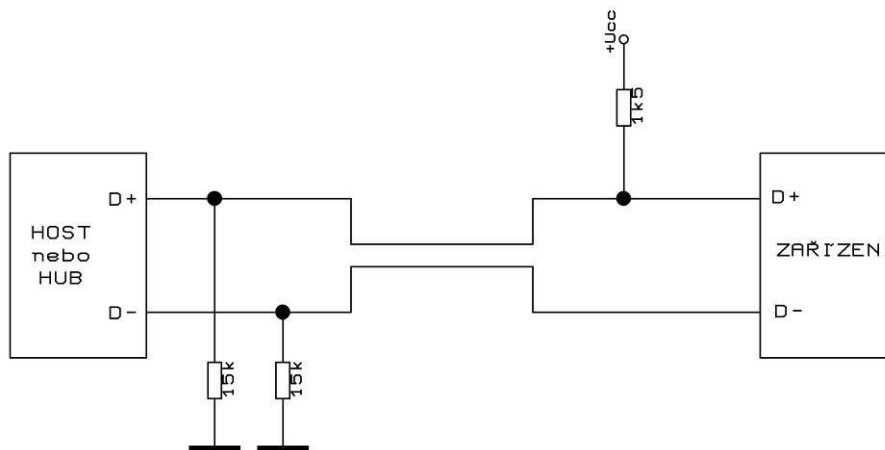
- Low speed zařízení připojí pull-up rezistor 1,5 k Ω mezi D- a +3,3 V. Viz obr. 2.3
- Full speed zařízení připojí pull-up rezistor 1,5 k Ω mezi D+ a +3,3 V. Viz obr. 2.4
- High speed zařízení připojí pull-up rezistor 1,5 k Ω mezi D+ a +3,3 V, stejně jako u Full speed, ale změna rychlosti se řeší programově.

Tyto rezistory jsou použity zároveň protistranou k detekci, zda je na protistraně připojeno USB zařízení, protože jinak jsou linky staženy k 0 V pomocí Pull down rezistorů o velikosti 15 k Ω .



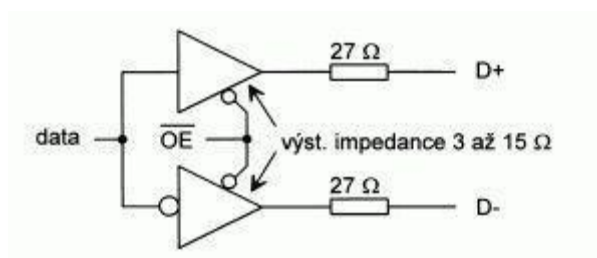
Obr. 2.3 Definice Low speed zařízení

Zařízení podle standardu USB 1.1 nemusejí podporovat Full speed, což umožňuje produkovat levnější rozhraní v aplikacích, kde není kritická rychlost. Totéž platí pro USB 2.0, jehož zařízení nemusejí podporovat High speed.



Obr. 2.4 Definice Full speed zařízení

Napěťové úrovně jsou přibližně TTL - logická nula je 0,3 V se zatížením do 1,5 kΩ proti kladnému napájení, logická jednička 2,8 V se zatížením do 15 kΩ proti zemi. Pro obě varianty je použito diferenční zapojení vysílačů viz obr. 2.5.



Obr. 2.5 Diferenční zapojení vysílačů

Minimální diferenční vstupní napětí přijímače je 200 mV, při souhlasném rušivém napětí až 2,6 V. Rozhraní USB zařízení jsou dále doplněna zakončovacími rezistory, podle výše uvedené definice rychlosti rozhraní.

2.4.1 Napájení

USB zařízení mohou být napájena přímo z USB sběrnice, pokud jejich odběr nepřekročí 100 mA, příp. 500 mA jen pro jedno zařízení na celé USB sběrnici. Napájecí napětí je 5 V. Pokud mají USB zařízení vlastní zdroj, je řízen USB sběrnici (zapínání, vypínání, SUSPEND mód atd.). Každý segment USB umožňuje omezený přenos výkonu pro napájení USB zařízení, přičemž zařízení může být současně napájeno z vlastního zdroje. HUB dodává pomocí napájecích pinů do USB zařízení napětí 4,75 V až 5,25 V. Maximální pokles napětí na HUBu je 0,35 V. Každý HUB musí být schopen poslat konfigurační data na napětí 4,4 V, ale jen "low-power" funkce musejí fungovat.

HUB napájený po sběrnici: odběr max. 100 mA při zapnutí a 500 mA průběžně.

HUB napájený ext. zdrojem: odběr max. 100 mA, musí dodávat 500 mA na každý port.

Zařízení "Low power": odběr max. 100 mA.

Běžná zařízení: odběr max. 100 mA.

"Spící" zařízení: odběr max. 0,5 mA.

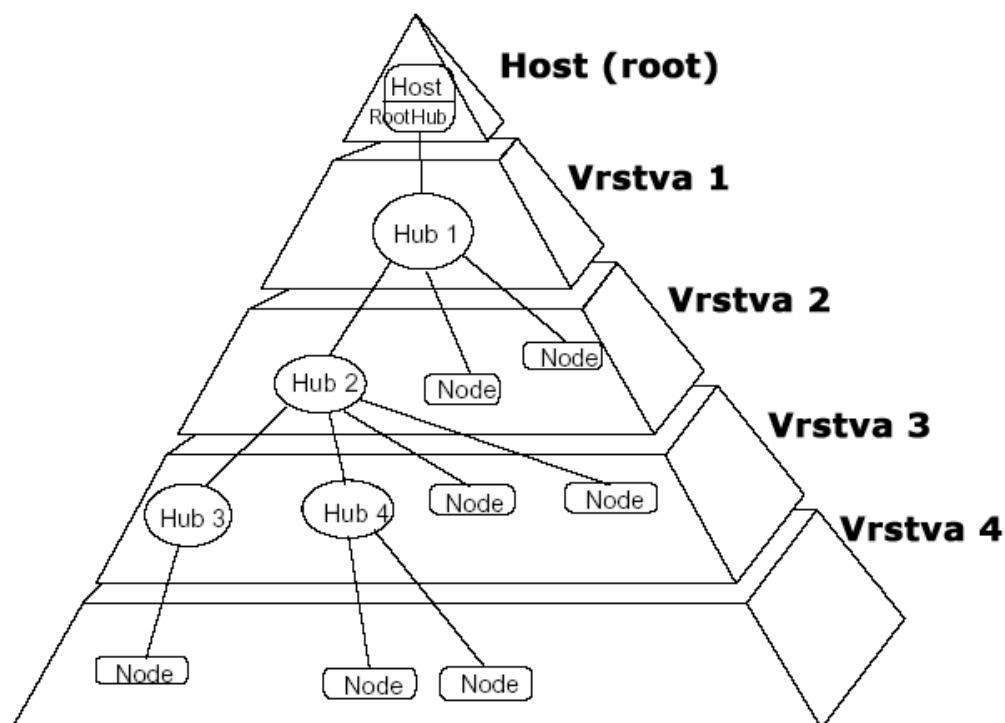
2.4.2 Stínění

Stínění USB kabelu musí být připojeno k pinu GND na straně "host". Žádné zařízení již nepřipojuje stínění k pinu GND.

2.5 Architektura USB

Zařízení vybavené rozhraním USB je buď rozdělovač (HUB – centrální jednotka hvězdicovité struktury), nebo funkční jednotka (periferní zařízení – např. myš, klávesnice, scanner, tiskárna, MP3 přehrávač, digitální audio vstup atd.). Propojení je řešeno pomocí víceúrovňové hvězdicové struktury.

Středem každého hvězdicového propojení je HUB a jednotlivé propojovací segmenty spojují buď počítač (USB host) obsahující centrální HUB s funkčními jednotkami a HUBy na vyšší úrovni, nebo HUB na vyšší úrovni s funkčními jednotkami a HUBy na nižší úrovni (viz obr.2.6).



Obr. 2.6 Hvězdicová struktura rozbočování USB

Každé USB zařízení má svoji USB adresu a podporuje jednu nebo několik koncových jednotek (end-points / nodes), se kterými může počítač komunikovat. Příklad jednoduché reálné struktury umožňující připojení běžných komerčních periférií je uveden na obrázku 2.7. Počítač může přes HUBy komunikovat se všemi zařízeními na tomto obrázku.



Obr. 2.7 Příklad jednoduché struktury

Každý HUB a jeho kabel zvyšuje zpoždění signálu, které nesmí překročit určitou maximální hodnotu. Proto lze zapojit maximálně 7 HUBů a maximálně 127 zařízení.

2.6 Přenos dat

Na USB sběrnici jsou všechny aktivity iniciovány z PC, jediného hostitelského zařízení typu "master". Žádné zařízení tedy nemůže na sběrnici vysílat data samovolně, vždy o to musí být požádáno ze strany PC. Datové pakety mají délku 8 až 256 bajtů a všechny putují v rámci o délce 1 ms. Uvnitř jednoho rámce mohou být pakety pro různá cílová zařízení a mohou mít i jiné rychlosti přenosu. O jejich směrování se stará HUB.

Rychlost je předepisována výhradně hostitelem. Všechna periferní zařízení jsou typu slave, a protože se hodinový signál nepřenáší po zvláštní lince, musí se slave synchronizovat na datový tok. K tomu se používá metoda NRZI technika *bit stuffing* (vsouvání bitů), která v případě delší neměnnosti posloupnosti bitů vloží za každých šest bezprostředně po sobě následujících jedniček jednu nulu. Tím je vynucena změna napěťové úrovně, zajišťující potřebnou synchronizaci komunikujících zařízení. Příjímač pak tuto nadbytečnou 0 odstraní. Kódování a dekodování signálů je čistě hardwarovou záležitostí.

Každý paket obsahuje za účelem synchronizace speciální bajt, tzv. sync bajt, podle kterého snadno zasynchronizuje. Během následujícího přenosu musí být synchronizace zachována. K adrese zařízení se přidává adresa tzv. endpointu (koncového zařízení). Tato adresa udává, kam se mají data v určitém zařízení uložit nebo vyzvednout. USB software si vytvoří logické datové cesty tzv. "pipes" (trubice) k jednotlivým endpointům konkrétních zařízení. Jedna pipe pak představuje logický kanál k jednomu endpointu v jednom zařízení. Fyzicky jsou data přenášena v paketech a ty v milisekundových rámcích. Pakety jsou pomocí hardwaru směřovány na určený endpoint. Jedno zařízení může využívat několik datových cest, čímž výrazně vzroste přenosová rychlost.

2.7 Typy přenosů dat na USB

Pro veškerou komunikaci mezi počítačem a funkční jednotkou jsou k dispozici tři typy paketů. Každá výměna dat začíná tím, že počítač vyšle tzv. token packet obsahující popis typu a směru výměny dat, adresu USB zařízení a číslo koncové jednotky (endpoint number). Pak zařízení, které má vysílat data, vyšle datový paket nebo indikuje, že žádná data nejsou k dispozici. Přijímací strana nakonec vyšle tzv. handshake packet, kterým informuje, zda přenos proběhl úspěšně.

Existují zde dva typy přenosového modelu:

Zpráva (message) využívající asynchronní přenos. Má přesnou strukturu:

- **Řídící přenos** (Control transfer) - určená pro konfigurování poprvé aktivovaného zařízení. Pracuje s vysokou prioritou a automatickým zabezpečením chyb.
- **Hromadný přenos** - zpráva obsahující větší objem dat se zabezpečením (např. pro tiskárnu nebo plotter), jež je většinou segmentována do více částí. Priorita přenosu je nízká, proto tento typ není vhodný na časově kritické operace.
- **Zpráva s přerušením** (obvykle několik bajtů), kterou spontánně vysílá zařízení, aby předalo zprávu o svém stavu (např. změna polohy myši).

Izochronní přenos je vhodný pro přenos velkých množství dat s definovanou rychlostí, bez jejich zabezpečení. Vhodné pro systémy, kde výpadek komunikace je horší než chyba v přenosu (např. zvukové karty).

2.8 Rozpoznání zařízení

Protože USB podporuje technologii Plug&Play, každé zařízení připojené k počítači je automaticky rozpoznáno operačním systémem. Po připojení zařízení se operační systém dotáže USB zařízení na parametry, ve formě tak zvaných deskriptorů. Podle získaných

informací se vybere odpovídající ovladač. Tyto operace provádí operační systém sám a uživatel se nemusí o nic starat.

HUB rozpozná nově připojené zařízení pomocí zvednutí D+ nebo D- linky pull-up rezistorem na +3,3 V. Potom se provedou následující kroky:

- ✚ HUB pošle hostitelskému počítači (dále jen Host) informaci, že je připojeno nové zařízení.
- ✚ Host pošle dotaz HUBu, na kterém portu je připojeno zařízení.
- ✚ Host vydá příkaz tento port aktivovat a provést reset USB sběrnice.
- ✚ HUB vyvolá reset USB sběrnice o délce 10 ms, poskytne zařízení proud 100 mA. Sériový interface v zařízení vyvolá reset mikroprocesoru a zařízení je připraveno.
- ✚ Host se obrátí na zařízení na jeho implicitní sběrnicové adrese 0. Přečte první bajty deskriptoru zařízení pro stanovení délky datových paketů.
- ✚ Host přiřadí zařízení jeho sběrnicovou adresu.
- ✚ Host si načte všechny informace deskriptoru zařízení.
- ✚ Host přiřadí zařízení konfiguraci, povolí odběr proudu, jaký je obsažen v deskriptoru. Zařízení je nyní připraveno.

2.9 Deskriptory zařízení

Podle informací obsažených v deskriptoru zařízení přiřadí host zařízení jednu z možných konfigurací. Kromě deskriptoru zařízení se používají i konfigurační deskriptory, které obsahují například informace o požadovaném odběru proudu. Následující tabulka 2.3 obsahuje pouze nejvýznamnější deskriptory zařízení.

Tab. 2.3 Deskriptory zařízení

Položka	Význam
VID (Vendor ID)	číselný identifikátor výrobce (16bit číslo, přiděluje organizace USB)
PID (Product ID)	číselný identifikátor výrobku (16bit číslo, přiděluje výrobce USB)
Manufactuer ID	řetězec identifikující výrobce
Manufactuer	řetězec popisující výrobce
Product	řetězec popisující výrobek
Serial number	řetězec sériového čísla

Podrobný popis celého komunikačního protokolu lze nalézt na oficiálních stránkách Universal Serial Bus [1].

3. Analýza konverze USB na paralelní sběrnici

V dnešní době je zařízení USB velice populární, zejména mezi uživateli, pro jeho jednoduchost připojení (plug and play). Pro vývojáře zařízení je zabudování komunikace USB komplikovanější než RS 232, protože je potřeba vyvinout firmware na straně zařízení a ještě programové podpory na straně PC (ovladače).

Zabudování USB rozhraní do zařízení je možné vyřešit následujícími způsoby:

1. Mikrokontroléry s hardwarovou implementací USB
2. Softwarová implementace USB
3. Samostatné USB řadiče

3.1 Mikrokontroléry s hardwarovou implementací USB

První možností je použití mikrokontroléru, který už má zabudované hardwarové USB rozhraní. V tom případě je nutné poznat, jak USB pracuje a podle toho napsat firmware do mikrokontroléru. Navíc je ještě nutné napsat ovládač (driver) na straně počítače (pokud ho už operační systém neobsahuje – zařízení třídy HID). Nevýhodou tohoto řešení, a to hlavně pro malé výrobce a amatéry, je malá dostupnost těchto mikrokontrolérů, jejich vysoká cena a také již zmíněný vývoj ovladačů.

Jistou kompenzací k tomu, že musíme psát firmware do mikrokontroléru je to, že například na stránkách firmy ATMEL [2] jsou ke každému mikrokontroléru uveřejněny aplikační poznámky. V těchto poznámkách jsou obsaženy knihovny pro komunikaci přes USB a různé konstrukce, které demonstrují komunikaci přes USB. Ke každé poznámce jsou pak většinou k dispozici i soubory s firmware pro mikrokontroléry.

V příloze A jsou uvedeny tabulky mikrokontrolérů některých výrobců, které mají HW podporu USB (hardwarovou implementaci USB).

3.2 Softwarová implementace USB

Druhou možností je použít mikrokontrolér, který nemá v sobě zabudovanou hardwarovou podporu USB a použít softwarovou implementaci. Výhodou tohoto řešení je možnost použití levného mikrokontroléru.

Dobrý příklad softwarové implementace USB je uveden na stránkách „Objective development“ [3]. Jedná se o čistě firmwarovou implementaci USB rozhraní, určenou pro většinu mikrokontrolérů Atmel AVR, se kterými lze takto postavit zařízení, i když

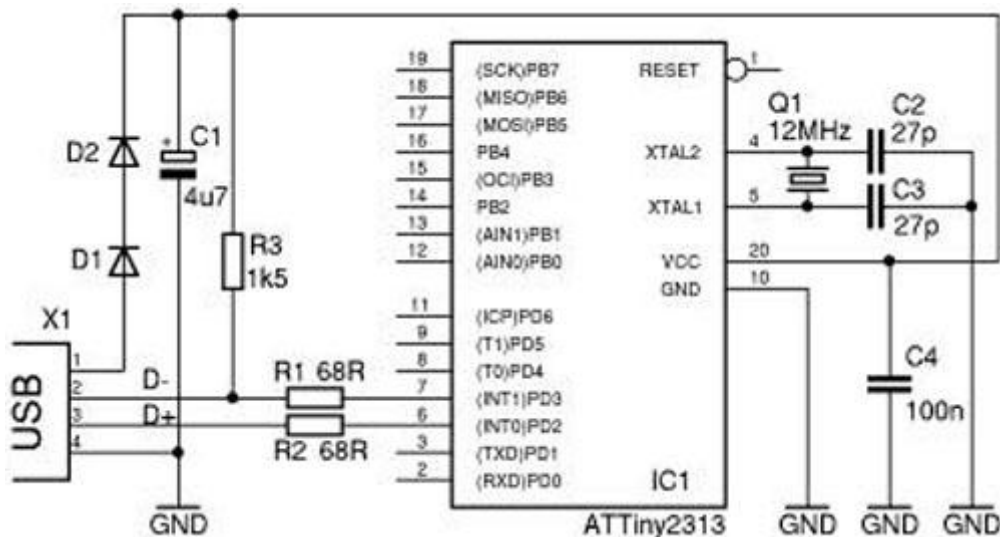
mikrokontrolér neobsahuje USB řadič a ani není třeba přidávat další obvod pro komunikaci přes USB.

3.2.1. Vlastnosti softwarové implementace USB

- ✚ Rozhraní je verze USB1.1 (low speed).
- ✚ Funguje na jakémkoliv AVR procesoru, který má alespoň 2 kB flash, 128 b RAM paměti a zvládne taktovací frekvenci 12 MHz.
- ✚ Obsahuje uživatelsky definovatelné identifikátory (vendor ID, product ID).
- ✚ Podporuje vícenásobný endpoint.
- ✚ Program obsluhy USB zabírá cca 1500 B paměti programu.
- ✚ Upravitelné části kódu jsou psány v ANSI-C a dobře okomentovány, lze je jednoduše modifikovat.
- ✚ Přenosové pakety až do velikosti 254 B.
- ✚ Ukázkové projekty demonstrují zařízení a implementaci ovladačů na operačních systémech Linux, Mac OS a Windows.
- ✚ Je možnost si zvolit licenci: open source nebo komerční.

3.2.2 Hardware

Na obrázku obr. 3.1 je znázorněno typické zapojení mikrokontrolérů k rozhraní USB, ze kterého je také napájen.



Obr 3.1 Typ zapojení na USB sběrnici

3.2.3 Vzorové konstrukce

Na těchto konstrukcích je demonstrováno, jak zapojit hardware a zároveň jsou ke každé konstrukci k dispozici soubory ke stažení, které obsahují softwarovou implementaci USB.

Power Switch

Jednoduchý příklad, který znázorňuje použití AVR-USB. Jedná se vlastně o jednoduchý převodník USB na 8bitovou paralelní sběrnici, která ovládá relé. K této konstrukci je také k dispozici software na PC (příkazový řádek), kterým se dá pomocí příkazů Power Switch ovládat. Power Switch je zároveň referenční konstrukcí pro použití AVR-USB.

HID Keys – USB klávesnice

Je to klávesnice se 17 klávesami, na které je demonstrováno použití HID třídy. HID (Human Interface Device) – zařízení pro komunikaci uživatele s PC. Ovladače hardwaru této třídy jsou již obsaženy v operačním systému.

RemoteSensor

Dálkový senzor, je bezdrátový termo-hydrometr s USB portem v přijímači. Je to pokročilejší projekt, protože používá přerušení koncového bodu k indikaci nově přichozích dat od senzorů.

Další příklady jsou na stránkách „Objective development“ [3].

3.2.4. Další softwarové implementace

Na internetových stránkách slovenského konstruktéra Ing. Igora Češka [4] je uveden další příklad softwarové implementace USB rozhraní pro mikrokontroléry.

Je zde uveden projekt: [základní USB modul IgorPlug-USB \(AVR\)](#) - dálkové ovládání počítače přes USB. Předložený projekt zahrnuje vývoj firmware a hardware na straně mikrokontroléru, vývoj ovladače na straně počítače (pro operační systém Windows), vývoj DLL knihovny na zpřístupnění funkcí zařízení jiným programům a vývoj demonstračního programu. Zařízení umožňuje příjem infračerveného kódu, ovládání 8bitové vstupně-výstupní brány, zápis do vnitřní EEPROM a vysílání znaků po sériové lince. Konstrukce USB-RS232 převodník + USB-8/16bit převodník + USB-EEPROM záznamník na bázi laciného AVR mikrokontroléru je pokračovatelem předešlého projektu. Konstrukce má vylepšené některé funkce, zejména příjem a vysílání po sériové lince.

3.3 Samostatné USB řadiče

Třetí možností je použít některý univerzální převodník USB na jiné rozhraní. Pro konstrukci jednoduchých řídicích nebo měřicích zařízení není vždy nutné sáhnout po

mikrokontrolérech. Nemá cenu studovat celou USB specifikaci, nebo se zabývat vhodným mikročipem, jeho konstrukcí a programováním. Při použití řadiče odpadá psaní speciálního firmware a nemusíme dokonce ani vědět jak USB pracuje, protože ovladače výrobce obvodu většinou zdarma poskytuje. Nevýhodou tohoto řešení je zvýšení ceny a rozměrů zařízení z důvodů zabudování jednoho obvodu navíc (a jeho externích součástek). Převážně se používá převod z USB na RS232, RS485, na 8bitovou datovou sběrnici, I²C sběrnici.

Příklady univerzálních USB převodníků od některých firem:

3.3.1 FTDI chip

Obvody FTDI Chip [5] jsou v současnosti pravděpodobně nejznámější a nejdostupnější USB rozhraní na českém trhu (zřejmě díky podpoře firmy ASIX [6]). FTDI nabízí značné množství softwarových doplňků a online manuálů. Rozhraní mezi hardwarem a aplikací zajišťují Virtual COM Port (VCP) drivery nebo WDM drivery. Drivery jsou poměrně často revidovány.

3.3.1.1 VINCULUM

Obvod Vinculum je z rodiny USB Host kontrolérů. Není to jen USB Host, který ovládá přenos dat, ale jeho vnitřní zabudovaný 8/32-bitový procesor a flash paměť dělá z obvodu Vinculum univerzální zařízení. Když například přistupujeme do zařízení „mass storage“, jako jsou USB Flash disky, můžeme transparentně ovládat FAT strukturu souborů prostřednictvím jednoduchých implementovaných příkazů přes sériový UART, SPI nebo paralelní FIFO interface.

3.3.1.2 FT 232R - USB UART IC

FT 232R je poslední generace (třetí) převodníků USB UART z produkce firmy FTDI Chip. Je to převodník z USB na sériovou UART s výstupem vnitřních hodin a unikátním sériovým číslem (FTDI Chip-IDTM). Byl přidán synchronní a asynchronní BitBang mód. Obvod byl také zjednodušen o externí EEPROM, zdroj hodin a rezistor pro určení rychlosti sběrnice.

3.3.1.3 FT245R - USB FIFO IC

FT 232R je poslední generace (třetí) převodníků USB FIFO z produkce firmy FTDI Chip. Je to převodník z USB na paralelní FIFO interface a unikátním sériovým číslem (FTDI Chip-IDTM). Byl přidán synchronní a asynchronní BitBang mód. Obvod byl také zjednodušen zabudováním externí EEPROM, zdroje hodin a rezistor pro určení rychlosti USB přímo do obvodu.

3.3.1.4 FT2232C FT2232L FT2232D - Dual USB UART/FIFO IC

FT2232 je třetí generací rodiny obvodů USB UART/FIFO. Tento obvod se vyznačuje dvěma multi-účelovými UART/FIFO kontroléry, které mohou být konfigurovány individuálně do několika různých módů. FT2232 nabízí množství nových módů provozu používajících Multi-Protocol Synchronous Serial Engine (MPSSE) interface, který je navržen speciálně pro synchronní sériové protokoly, jako jsou I²C, JTAG a SPI.

3.3.1.5 FT232BM, FT232BL, FT232BQ - USB UART IC

FT232BM je druhou generací rodiny obvodů USB UART (FT232BL je vyroben technologií bez použití olova). Tento obvod je nástupce FT8U232AM, nevyznačuje se jen dalšími funkcemi oproti předchůdci, ale také redukuje počet vnějších součástek. Zachovává také kompatibilitu vývodů z pouzdra. FT232BQ je verzí obvodu FT232BL (bez olova) v patici QFN-32.

3.3.1.6 FT245BM, FT245BL, FT245BQ - USB FIFO IC

FT245BM je druhou generací rodiny obvodů USB paralelní FIFO (FT245BL je bezolovnatá verze). Tento obvod je nástupce FT8U245AM, nevyznačuje se jen dalšími funkcemi oproti předchůdci, ale také redukuje počet vnějších součástek. Zachovává také kompatibilitu vývodů z pouzdra. FT245BQ je verzí obvodu FT245BL v patici QFN-32.

3.3.2 Philips

3.3.2.1 PDIUSB11

Obvod vyrobený firmou Philips Semiconductors [7] pracuje jako převodník USB (podporuje USB standart ver.1.1) sběrnice na I²C sběrnici. Maximální rychlost komunikace na sběrnici I²C je 1 Mbit/s a chová se jako zařízení slave. Obvod je napájen napětím 3,3 V, ale jeho vývody jsou tolerantní k 5 V logice.

3.3.2.2 PDIUSB12

Obvod pracuje jako převodník USB (podporuje USB standart ver.2.0) na paralelní sběrnici. Datová propustnost obvodu je velmi vysoká 2 Mbit/s.

3.3.3 Cypress

3.3.3.1 SL811HS

Řadič sběrnice USB SL811HS od firmy Cypress [8] může pracovat ve dvou režimech HOST a SLAVE (zařízení). Je navržen přímo pro použití s klasickými mikrokontroléry, podporuje programovaný nebo mapovaný vstup/výstup do paměti. Obvod podporuje i DMA přístup do paměti a generování požadavku o přerušení a je vybaven příslušnými vývody.

Další vlastnosti obvodu SL811HS:

- ✚ Full speed a Low speed USB datové přenosy
- ✚ USB specifikace verze 1.1
- ✚ Osmibitová datová sběrnice
- ✚ 3,3 V logika, tolerantní s 5 V logikou
- ✚ 256 kB vnitřní paměti RAM
- ✚ SMD TQFP pouzdro

4. Návrh a realizace sériově-paralelního převodníku

4.1 Zvolené řešení převodníku

Po prostudování problematiky převodu, respektive samotného dekodování USB komunikace, jsem se přiklonil k použití samostatného USB řadiče. Jako nejvhodnější typ byl vybrán FT 232R od firmy FTDI Chip [5], který provádí převod USB na sériovou komunikaci.

Pro volbu integrovaný obvod FT 232R hovořilo několik pozitivních faktorů:

- ✚ Integrovaný obvod je snadno dostupný
- ✚ Malé množství externích součástek potřebných k obvodu
- ✚ Velké množství dokumentace včetně spousty praktických konstrukcí
- ✚ Knihovny k obvodu (VCP a D2xx) – jsou volně ke stažení, dobře komentovány a jsou s nimi vytvořeny vzorové příklady.

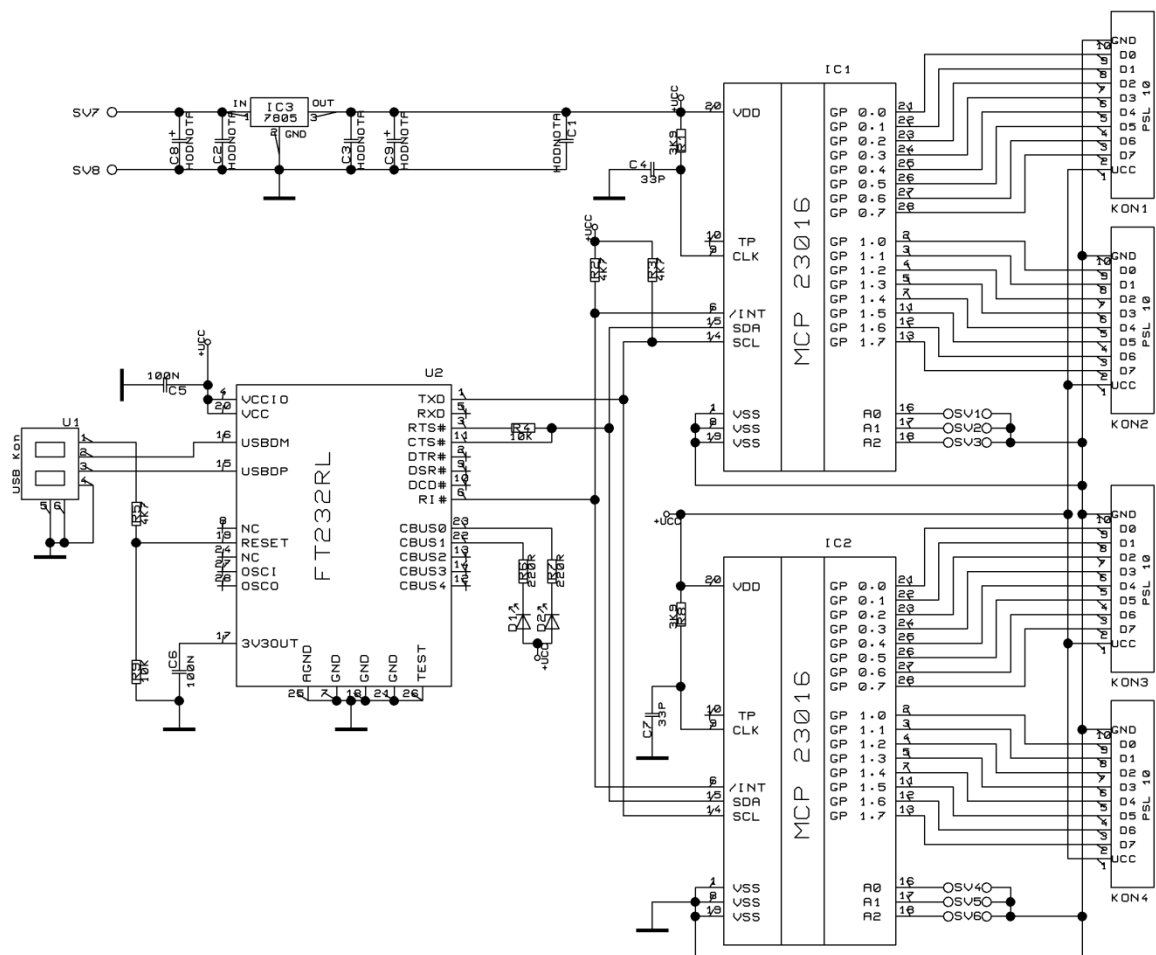
Převod na paralelní sběrnici provádí integrovaný obvod MCP 23016. Přestože obvod FT 232R může sám pracovat jako převodník na paralelní sběrnici ve speciálním módu tzv. „BitBang mód“, nemá dostatečný počet vývodů a jejich proudové zatížení je také malé. Z tohoto důvodu převod provádí obvod MCP 23016, který má 16 vstupně/výstupních linek, které lze libovolně konfigurovat jako vstup nebo výstup s proudovým zatížením až 25 mA/vývod. Zvolené řešení má další výhodu a to, že obvod MCP 23016 má 3 vývody pro nastavení adresy, počet obvodů lze tedy rozšířit až na 9, tedy 144 vývodů. Navíc, protože se

s obvodem MCP 23016 komunikuje po sběrnici I²C, lze k zařízení přímo připojit další integrované obvody komunikující přes I²C, jako jsou například A/D převodníky, D/A převodníky.

4.2 Konstrukce převodníku

4.2.1 Popis zapojení

Díky integrovaným obvodům se zapojení převodníku značně zjednodušilo jen na pár integrovaných obvodů a pár pasivních součástek. Zapojení celého převodníku je zobrazeno na obr. 4.1



Obr 4.1 schéma převodníku

Signál z konektoru USB je veden do integrovaného obvodu U2, který provádí komunikaci po USB sběrnici a dekódování signálu. Obvod U2 (FT 232RL) je zapojen podle doporučeného katalogového zapojení s napájením z vlastního zdroje. Napájení ze sběrnice USB je použito

k řízení vstupu RESET, aby bylo možno navázat komunikaci s USB Hostem nebo HUBem. Když je USB Host nebo HUB připojen na napájení, je interní rezistor 1,5 k Ω v FT 232RL na vývodu USBDP je připojen na napětí +3,3 V, a to identifikuje, že je na sběrnici připojeno Full Speed zařízení. Když je USB Host nebo HUB odpojen od napájení, je vývod RESET „stažen k zemi“ a zařízení je drženo v resetu. Když je RESET „stažen k zemi“, není interní rezistor připojen k +3,3 V, proto žádný proud neteče přes interní rezistor do vývodu USBDP a následně do vypnutého USB Hostu nebo HUBu.

Led diody D1 a D2 indikují příjem a vysílání dat na USB sběrnici. D1 indikuje příjem a D2 indikuje vysílání.

Pro samotný převod na paralelní sběrnici jsou použity dva integrované obvody IC1 a IC2 (MCP 23016). Podrobně je obvod popsán v kapitole [4.5](#). Vývody vstupně výstupních linek integrovaných obvodů (GP1 a GP2) jsou přímo připojeny na konektory KON1 až KON4, na které je také vyvedeno napájení pro externí obvody. Součástky R1, C4 a R8, C7 slouží integrovaným obvodům k získání hodinového signálu a jsou zapojeny podle doporučeného katalogového zapojení.

Komunikace mezi FT 232RL a MCP 23016 probíhá po sběrnici I²C, dvěma vodiči SDA a SCL. Zapojení převodníku je koncipováno tak, že na sběrnici je jen jeden MASTER (FT 232RL), proto linku SCL řídí jen on. Linka je připojena přes rezistor R3 na +5 V. Linku SDA může řídit jak MASTER tak i SLAVE, protože obvod FT 232RL není přímo určen pro komunikaci po sběrnici I²C, je zde použito netradiční řešení komunikace. Při zápisu na SDA se data zapisují na linku z vývodu RTS# přes rezistor R4. Při čtení z SDA se linka se trvale napájí opět z vývodu RTS# přes rezistor R4 na +U_{cc}, ale data se linky čtou vývodem CTS# (více o komunikaci po I²C v kapitole [4.4](#)). Adresu integrovaných obvodů IC1 a IC2 můžeme měnit pomocí propojek JP1-JP6.

Při generování přerušení obvodem IC1 nebo IC2 je linka INT připojena na zem přes vnitřní tranzistory obvodu MCP 23016 s otevřeným kolektorem. Tento signál je přiveden na vstup RI obvodu U2. To lze využít k jednoduché kontrole vstupů na MCP 23016, jestli na nich nedošlo k nějaké změně. Místo hlídání každého jednotlivého vstupu, kterých může být až 32, v našem případě, stačí hlídat jen jeden vývod. V případě, že dojde ke změně na jakémkoliv vstupu, vyvolá to přerušení a tranzistor zapojený na vývodu INT přitáhne linku na zem, což lze jednoduše sledovat řídicím systémem. Toto řešení podstatně ušetří řídicímu systému čas na kontrolu vstupů.

Integrovaný obvod IC3 slouží ke stabilizaci napájecího napětí a kondenzátory C2, C3, C8 a C9 k jeho filtraci.

4.3 Integrovaný obvod FT 232RL

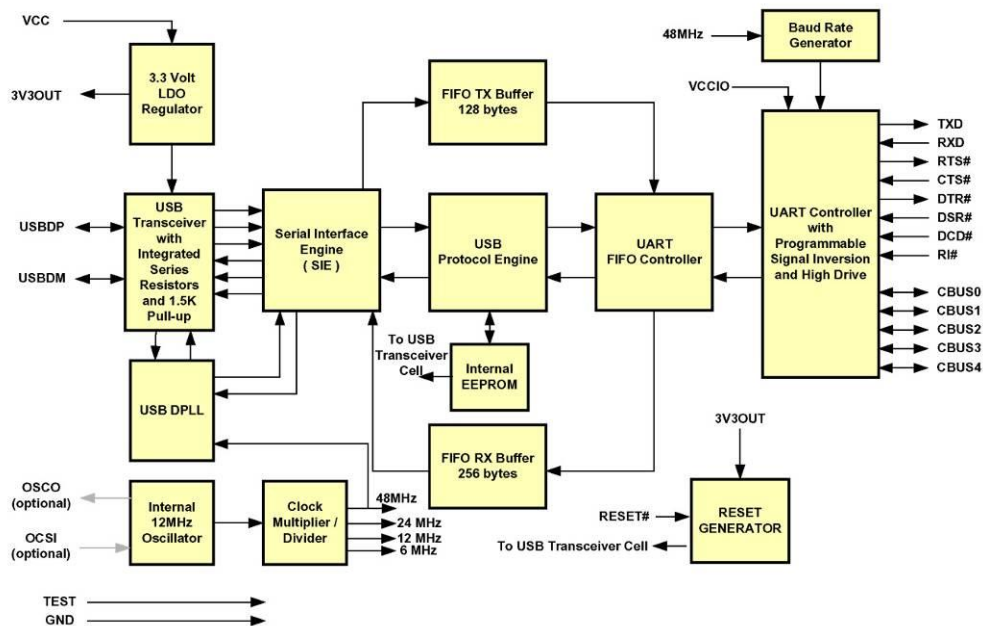
FT 323R je poslední generace převodníků USB UART z produkce firmy FTDI Chip [5]. V čipu FT 232R byly přidány nové funkce oproti jeho předchůdci, které z něj dělají čip „3 v 1“ pro široký rozsah aplikací. Výstup vnitřních hodin (6 MHz, 12 MHz, 24 MHz a 48 MHz) je možné vyvést z čipu a řídit s ním externí logiku. Unikátní číslo (FTDI Chip-ID™) je vypáleno již při výrobě přímo do čipu a je možné jej číst přes USB, což je možné využít při ochraně uživatelského software proti kopírování. Převodník byl dále zjednodušen zabudováním EEPROM a USB rezistoru 1,5 kΩ přímo do čipu.

4.3.1 Hardwarové vlastnosti

- ✚ Jednočipový převodník USB na sériový interface.
- ✚ Plné zabudování USB protokolu do čipu – není třeba vyvíjet firmware.
- ✚ Podpora 7/8bitového přenosu, 1/2 stop bitů, parita sudá, lichá, bez parity.
- ✚ Přenosová rychlost od 300 baud do 3 Mbaud.
- ✚ Přijímací buffer 384 B, vysílací buffer 128 B pro zajištění velké propustnosti dat.
- ✚ Ovladače pro VCP a D2xx volně ke stažení.
- ✚ Konfigurovatelné CBUS vývody.
- ✚ Možnost připojení LED pro indikaci vysílání nebo příjmu.
- ✚ Automatická kontrola vysílacího bufferu pro aplikace s RS 485.
- ✚ Možnost vyvést hodinový signál ven z čipu pro řízení externích součástek (6 MHz, 12 MHz, 24 MHz, 48 MHz).
- ✚ Synchronní a asynchronní BITBANG mód.
- ✚ Nastavitelný Time Out přijímače.
- ✚ Zabudovaná EEPROM pro uložení VID, PID, sériového čísla a konfigurace čipu.
- ✚ Podpora různých druhů napájení.
- ✚ Podpora sběrnice s napětím 3,3 V.
- ✚ Integrované USB rezistory, reset po zapnutí, interní oscilátor.

4.3.2 Blokové schéma obvodu

Na obrázku 4.2 je znázorněno blokové schéma celého obvodu FT232R. Funkce jednotlivých bloků je pak popsána níže.



Obr 4.2 Blokové schéma obvodu FT232R

3,3 V LDO Regulator

Generuje 3,3 V referenční napětí pro buzení USB vysílače. Vyžaduje vnější blokovací kondenzátor připojený na 3,3 V vývod obvodu. Také poskytuje napětí pro interní rezistor 1,5 k Ω připojený na USB DP. Slouží také jako zdroj 3,3 V pro externí logiku a odběrem až maximálně 50 mA.

USB Transceiver

USB vysílač/přijímač poskytuje USB 1.1/USB 2.0 fyzické rozhraní pro kabel.

USB DPLL

Detekuje přichodící NRZI USB signál a rozděljuje ho na hodinový signál a na datový signál pro SIE.

Internal 12 MHz Oscillator

Generuje signál 12 MHz pro interní 4x násobičku hodinového signálu. Poskytuje také referenční hodinový signál pro SIE, USB protokol engine a pro blok UART FIFO controller.

Clock Multiple/Divide (Násobička/dělička hodin)

Vezme signál 12 MHz z interního oscilátoru a generuje z něj signál 48, 24 a 6 MHz. Signál 48 MHz je dále použit pro USB DPLL a blok Baud Rate Generator.

SIE (Seriál Interface Engine – Sériové rozhraní)

Provádí paralelně sériovou a sériově paralelní konverzi USB dat. Se shodou s protokolem USB 2.0 zajišťuje vkládání/vyjímání synchronizačních bitů a CRC5/CRC16 generaci/testování v datovém toku USB.

UPE (USB Protokol Engine)

Spravuje datový tok z řídicího koncového bodu USB. Ovládá nízkourovňové požadavky USB protokolu, které vysílá USB Host a příkazy pro kontrolu funkčních parametrů UART.

FIFO Tx Buffer

Data z výstupního koncového USB bodu se ukládají do Tx bufferu, odkud jsou vyjímána vysílacím registrem UART, který spravuje UART FIFO.

FIFO Rx Buffer

Data z přijímacího UART registru se ukládají do Rx bufferu před tím, než jsou vyjmuta SIE při dotazu na data z koncového bodu.

UART FIFO Controller

Ovládá přenos dat mezi Tx/Rx buffery a vysílacím/přijímacím registrem UART.

UART Controller with Programmable Signal Inversion and High Drive

Spolu s UART FIFO kontrolerem a UART kontrolerem řídí přenos dat mezi UART Rx, UART Tx bufferem a UART vysílacím a přijímacím registrem. Zajišťuje asynchronní 7/8bitovou paralelně-sériovou a sériově paralelní konverzi dat na RS 232. Řídí signály RTS, CTS, DSR, DTR, DCD a TI. Poskytuje řídicí signály pro ovládání RS 485 vysílačů.

Baud Rate Generator – generátor přenosové rychlosti

Bere signál 48 MHz z násobičky a posílá ho do 14bitové děličky a 3bitového registru, aby byla možnost jemně doladit přenosovou rychlost. Přenosovou rychlost lze nastavit v rozmezí 300 baud až 3 Mbaud.

RESET generátor

Provádí reset obvodu po připojení napájecího napětí. Vývodem RESET můžeme provést reset obvodu z vnějšího zdroje.

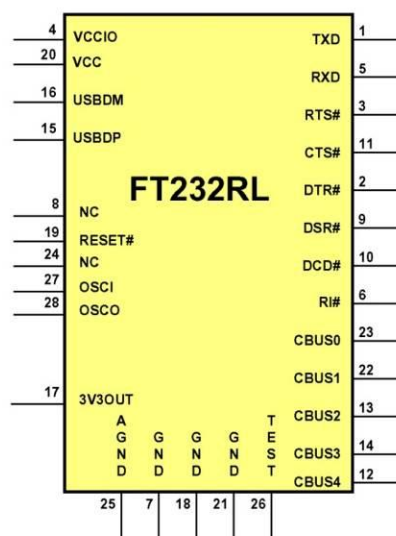
Internal EEPROM – Interní EEPROM paměť

Slouží k uložení VID, PID, sériového čísla, popisu zařízení a k různým nastavením obvodu. Používá se také k nastavení vývodů CBUS.

4.3.3 Popis vývodů FT 232RL

Obvod FT 232R se vyrábí ve dvou verzích pouzder SSOP-28 a QFN-32. Na obr. 4.3 je znázorněna schematická značka vývodů v pouzdru SSOP-28. V tabulce 4.1 je potom popis jednotlivých vývodů s vysvětlením jejich funkce a popisem, jestli jde o vstup nebo výstup.

Obr. 4.3 schematická značka FT 232RL



4.3.4 Režim BITBANG

Obvod FT 232R můžeme programově přepnout do speciálního režimu, tak zvaného „BitBang“, ve kterém můžeme vývody UART rozhraní a CBUS přímo řídit, a to každý zvlášť. Každý vývod lze samostatně konfigurovat jako vstup nebo výstup. Datové linky obvodu lze pak chápat jako 8bitovou vstupně/výstupní sběrnici.

Rychlost přenosu bajtů uložených do výstupního bufferu odpovídá 16 násobku zvolené přenosové rychlosti. Zvolená přenosová rychlost také určuje vzorkovací kmitočet, kterým se ukládá do vstupního bufferu. Například při přenosové rychlosti 9600 Bd je vzorkování 153600 B/s (jeden bajt za 6,5µs).

Tab. 4.1 vývodů obvodu FT 232RL v pouzdrů SSOP-28

Vývod	Signál	Typ	Popis
USB Rozhraní			
15	USBDP	Vstup/výstup	USB signál D+
16	USBDM	Vstup/výstup	USB signál D-
Napájení			
4	VCCIO	Napájení	Napájení pro UART a CBUS rozhraní v rozhraní 3 až 5V. Pokud pracujeme s externí logikou 3,3V zapojíme na vývod 3,3V zabudovaného regulátoru
7,18,21	GND	Napájení	Napájecí a signálová zem
17	3,3 V	Výstup	Výstup zabudovaného regulátoru 3,3V. Vývod musí být blokován kondenzátorem o kapacitě 100nF. Z obvodu může být odebíráno maximálně 50 mA pro externí logiku
20	VCC	Napájení	Napájení 5V jádra a zabudovaného regulátoru
25	AGND	Napájení	Napájení zabudované děličky hodin
Různé vývody			
8, 24	NC	NC	Nezapojeno
19	RESET#	Vstup	Reset obvodu vnějším zdrojem
26	TEST	Vstup	Test obvodu, používá výrobce.
27	OSCI	Vstup	Vstup hodin při použití externího zdroje hodin
28	OSCO	Výstup	Výstup hodin při použití externího zdroje hodin
UART rozhraní a CBUS			
1	TXD	Výstup	Výstup vysílaných asynchronních dat
2	DTR#	Výstup	Data terminál připravena kontrolní výstup (řídící linky modemu)
3	RTS#	Výstup	Požadavek na vysílání-kontrolní výstup (řídící linky modemu)
5	RXD	Vstup	Vstup přijímaných asynchronních dat
6	RI#	Vstup	Vyzvánění kontrolní vstup. Tímto vstupem můžeme budít USB host zařízení z režimu USB suspend
9	DSR#	Vstup	Data připravena kontrolní vstup
10	DCD#	Vstup	platná data kontrolní vstup
11	CTS#	Vstup	Připraven pro příjem
12	CBUS4	Vstup/výstup	Konfigurovatelný vývod CBUS. Funkce vývodu se nastavuje v interní EEPROM. Standardně je od výrobce nastaven jako SLEEP.
13	CBUS3	Vstup/výstup	Konfigurovatelný vývod CBUS. Funkce vývodu se nastavuje v interní EEPROM. Standardně je od výrobce nastaven jako TXDEN.
14	CBUS2	Vstup/výstup	Konfigurovatelný vývod CBUS. Funkce vývodu se nastavuje v interní EEPROM. Standardně je od výrobce nastaven jako PWREN.
22	CBUS1	Vstup/výstup	Konfigurovatelný vývod CBUS. Funkce vývodu se nastavuje v interní EEPROM. Standardně je od výrobce nastaven jako RXLED.
23	CBUS0	Vstup/výstup	Konfigurovatelný vývod CBUS. Funkce vývodu se nastavuje v interní EEPROM. Standardně je od výrobce nastaven jako TXLED.

4.4 Sběrnice I²C

Abychom mohli komunikovat s obvodem MCP 23016, je nutné pochopit principy komunikace po sběrnici I²C.

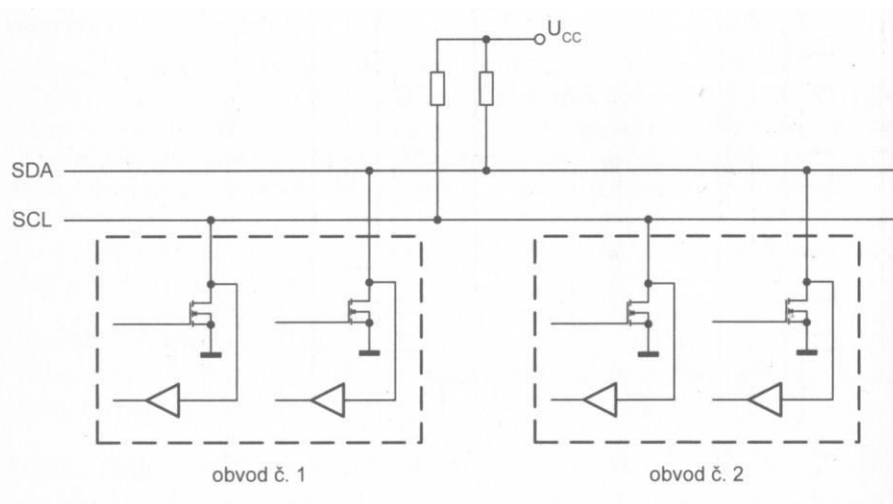
I²C je dvou vodičová obousměrná sériová sběrnice, kterou vytvořila firma Philips Semiconductors [7] a slouží ke komunikaci procesorů s periferními obvody.

Ke komunikaci slouží pouze dvě linky:

- SDA – slouží pro přenos dat
- SCL – slouží k přenosu hodinového signálu komunikace

4.4.1 Struktura sběrnice

Na obr. 4.4 je zobrazena struktura sběrnice I²C pro více master zařízení. Obě linky SDA a SCL musí být prostřednictvím pull up (zvedacích) rezistorů připojeny na kladný pól napájecího napětí. Obvod má na vstupu pro vysílání zapojen jen tranzistor s otevřeným kolektorem. Pokud obvod vysílá na sběrnici, přitáhne pouze linku k 0 V. Pokud by došlo ke kolizi komunikace, chtělo by vysílat více obvodů najednou, poškodí se pouze úroveň signálů a nikoli vysílací obvody. Na každý vývod SDA nebo SCL obvodu je připojen vstup pro příjem dat, tím je zajištěna obousměrná komunikace. Na obr. 4.4 je zobrazena dvou masterová komunikace. V případě, že je na sběrnici pouze jeden master (řídící obvod), řídí linku SCL pouze master.



Obr. 4.4 Struktura sběrnice I²C

4.4.2 Přenos bitů

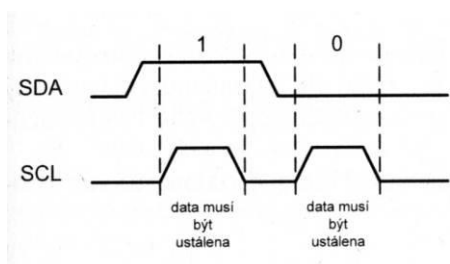
Přenos dat je řízen linkou SCL. Po dobu jednoho hodinového taktu se přenáší právě jeden datový bit a linka SDA nesmí být změněná po celou dobu trvání tohoto taktu, jinak by to bylo chápáno jako řídicí signál (viz dále). Na obr. 4.5 je znázorněn přenos dvou bitů.

4.4.3 Start a Stop

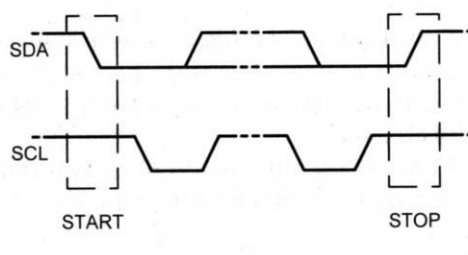
Protože je sběrnice I²C jen dvou vodičová musí se začátek a konec komunikace řešit speciálními stavy na sběrnici. Viz obr. 4.6.

- START (nebo S) – je oznámen sestupnou hranou linky SDA při SCL=log.1
- STOP (nebo P) – je oznámen náběžnou hranou linky SDA při SCL=log.1

Je-li sběrnice v klidovém stavu, jsou signály na SDA a SCL v logické 1.



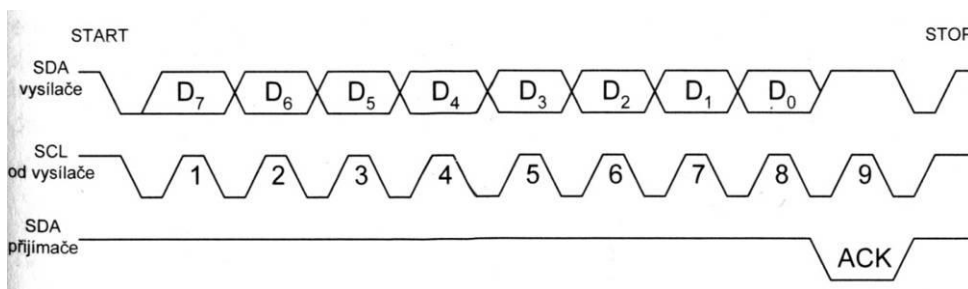
Obr. 4.5 Přenos bitů



Obr. 4.6 Řídící signály

4.4.4 Potvrzení přenosu ACK

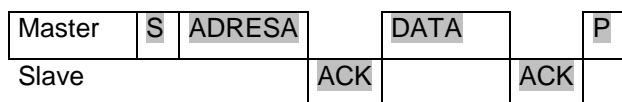
Protože počet přenesených datových bajtů z vysílače do přijímače mezi signály START a STOP není omezen, musí se zajistit zpětná kontrola, že byla data správně přijata. To se provádí tak, že přijímač vloží za každý datový bajt (8 bitů) jeden potvrzovací signál ACK (nebo A). ACK představuje log. 0 vloženu na linku SDA s potvrzovacím hodinovým signálem SCL. Viz obr. 4.7



Obr. 4.7 Přenos jednoho bajtu na I²C

4.4.5 Adresování na sběrnici I²C

Sběrnice I²C není vybavena adresovou sběrnici, ani výběrovým vodičem, adresa musí být vysílána stejným způsobem jako data. Příklad komunikace je na následujícím obrázku 4.8.



Obr. 4.8 Komunikace po sběrnici

Po vyslání signálu START se nejdříve pošle platná adresa daného obvodu. Po adrese musí oslovovaný obvod vytvořit ACK. Nyní se posílá datový bajt zakončený opět ACK oslovovaného obvodu. Přenos se uzavírá signálem STOP.

4.4.6 Adresy I²C zařízení

Adresa I²C se skládá ze tří částí. Viz tab. 4.2.

Tab. 4.2 Adresování obvodu

A7	A6	A5	A4	A3	A2	A1	A0- R/W
0	1	0	0	X	x	x	0/1

- Pevná část – je zabudovaná hardwarově a mění se s typem obvodu. Jedná se převážně vždy o nejvyšší 4 bity adresy (A7-A4). Například pro obvod MCP 23016 je to 0100.
- Volitelná část – jedná se obvykle o 3 bity (A3-A1) označeny na obrázku jako x. Nastavuje se připojením adresovacích vstupů na log. 0 nebo 1. Potom lze připojit až 8 stejných obvodů na jednu sběrnici.
- R/W – jedná se o nejnižší bit (A0) a určuje směr přenosu. Pro R/W=1 je obvod vysílačem (čteme z něj data) a pro R/W=0 je obvod přijímač (zapisujeme do něj data).

4.4.7 Elektrické parametry sběrnice I²C

Pro praktickou konstrukci zařízení se sběrnici I²C musíme také znát její mezní hodnoty. Ty jsou uvedeny v tabulce tab. 4.3.

Tab. 4.3 Elektrické vlastnosti sběrnice I²C

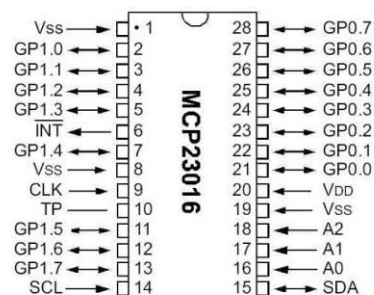
Parametr	Standard		Fast		jednotka
	min.	max.	min.	max.	
Vstupní log. 0	-0,5	1,5	-0,5	1,5	V
Vstupní log. 1	3	-	3	-	V
Výstupní log. 0	-	0,4	-	0,4	V
Vstupní odběr	-10	10	-10	10	μA
Kmitočet SCL	0	100	0	400	kHz
Doba trvání log. 0	4,7	-	1,3	-	μs
Doba trvání log. 1	4	-	0,6	-	μs

4.5 Integrovaný obvod MCP 23016



4.5.1 Základní vlastnosti MCP 23016

- ✚ 16bit vzdálený obousměrný I/O port
- ✚ Tři adresové piny
- ✚ Velké proudové zatížení na jeden I/O: 25 mA
- ✚ Rychlé rozhraní I²C (400kbit/s)
- ✚ Otevřený kolektor na výstupu přerušení
- ✚ Zachytávací registr přerušení
- ✚ Vnitřní reset po zapnutí
- ✚ Registr pro konfiguraci stavu výstupů (polaritu)



Obr. 4.9 vývody MCP 23016

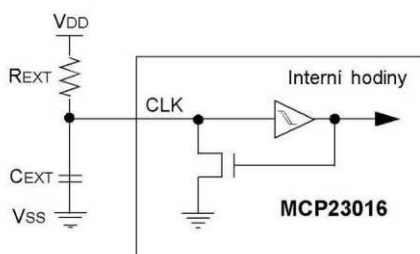
Integrovaný obvod vyrobený firmou Microchip [9] poskytuje 16 vstupně/výstupních linek a je určen pro I²C datovou sběrnici. Obvod má výstupy s velkým proudovým zatížením až 25 mA. Obvod obsahuje mnoho různých 8bit konfiguračních registrů pro vstup, výstup a volbu polarity. Je možné nastavit jednotlivé vývody obvodu jako vstup nebo výstup. Data pro každý vstup nebo výstup jsou uchovávaný v odpovídajícím vstupním nebo výstupním registru. Všechny registry jsou přístupné prostřednictvím sběrnice I²C.

Reset po zapnutí (POR) - Na čipu je umístěn obvod POR, který drží čip v RESETu, dokud napájecí napětí Vdd nevystoupí na dostatečnou úroveň, která deaktivuje POR.

Power up časovač - Power up časovač podrží čip po připojení napájecího napětí asi 72 ms ve stavu RESET, aby se mohlo Vdd napájecí napětí ustálit.

4.5.2 Generátor hodin

MCP 23016 používá externí RC obvod pro určení vnitřní rychlosti. Uživatel musí připojit rezistor a kondenzátor k obvodu, jak je to nakresleno na obr. 4.10. Pro správnou funkci obvodu je třeba frekvence typ. 1 MHz. Doporučené hodnoty R a C jsou v tabulce 4.4.



Tab. 4.4 Hodnoty RC

R _{ext}	C _{ext}
3.9 kΩ	33 pF

Obr. 4.10 Zapojení generátoru hodin

4.5.3 Adresa obvodu

Poslední tři bity ze 7 bitové adresy jsou volitelné uživatelem. Podle zapojení pinů A2-A0 můžeme definovat tyto bity. Viz tabulka 4.5

Tabulka 4.5 Adresa MCP 23016

Adresa	0	1	0	0	A2	A1	A0
Bit	b7						b1

4.5.4 Instrukce

Pomocí tohoto bajtu můžeme přistupovat k jednotlivým registrům obvodu MCP 23016. Viz tabulka 4.6. Funkce jednotlivých registrů bude vysvětlena dále.

Instrukce	Význam
00h	Přístup do GP0
01h	Přístup do GP1
02h	Přístup do OLAT0
03h	Přístup do OLAT1
04h	Přístup do IPOL0
05h	Přístup do IPOL1
06h	Přístup do IODIR0
07h	Přístup do IODIR1
08h	Přístup do INTCAP0 (jen pro čtení)
09h	Přístup do INTCAP1 (jen pro čtení)
0Ah	Přístup do ICON0
0Bh	Přístup do ICON1

Tab. 4.6 Instrukce MCP 23016

4.5.5 Registry

Legenda k tabulkám registrů:

R = bit je možno číst W = do bitu je možno zapisovat

U = nepoužitý bit, čte se jako 0

-n = hodnota po resetu

1 = bit je nastaven

0 = bit je smazán

x = neznámá hodnota

Registr portů

GP0 odpovídá portu GP0

GP1 odpovídá portu GP1

Čtením z těchto registrů zjistíme stav na jednotlivých vývodech těchto portů (vývodů integrovaného obvodu). Zápisem do těchto registrů změním stav záchytných registrů OLAT0, OLAT1 a stav na výstupech integrovaného obvodu.

Tab. 4.7 Registr portů

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GP0.7	GP0.6	GP0.5	GP0.4	GP0.3	GP0.2	GP0.1	GP0.0
GP1.7	GP1.6	GP1.5	GP1.4	GP1.3	GP1.2	GP1.1	GP1.0
bit7							bit0

Výstupní záchytný registr

OLAT0 je záchytný výstupní registr portu GP0

OLAT1 je záchytný výstupní registr portu GP1

Čtením z tohoto registru dostáváme jen aktuální stav výstupního záchytného registru, nikoliv stav výstupu. Zápisem do těchto registrů změníme stav výstupních záchytných registrů, které kontrolují výstup.

Tab 4.8 Výstupní záchytný registr

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OL0.7	OL0.6	OL0.5	OL0.4	OL0.3	OL0.2	OL0.1	OL0.0
OL1.7	OL1.6	OL1.5	OL1.4	OL1.3	OL1.2	OL1.1	OL1.0
bit7							bit0

Registr polarity vstupů

IPOL0 je registr polarity vstupů pro port GP0

IPOL1 je registr polarity vstupů pro port GP1

Tento registr dovoluje uživateli změnit polaritu (logickou úroveň) vstupních portů. Jestliže je bit v registru IPOL příslušného portu nastaven, je bit vstupního portu negován.

Tab 4.9 Registr polarity vstupů

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IGP0.7	IGP0.6	IGP0.5	IGP0.4	IGP0.3	IGP0.2	IGP0.1	IGP0.0
IGP1.7	IGP1.6	IGP1.5	IGP1.4	IGP1.3	IGP1.2	IGP1.1	IGP1.0
bit7							bit0

Registr směru vývodů

IOD0 registr kontroluje port GP0

IOD1 registr kontroluje port GP1

Jestliže je bit příslušného registru nastaven, je odpovídající vývod portu nastaven jako vstup, v opačném případě je výstup. Po resetu jsou všechny vývody nastaveny jako vstup.

Tab 4.10 Registr směru vývodů

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
IOD0.7	IOD0.6	IOD0.5	IOD0.4	IOD0.3	IOD0.2	IOD0.1	IOD0.0
IOD1.7	IOD1.6	IOD1.5	IOD1.4	IOD1.3	IOD1.2	IOD1.1	IOD1.0
bit7							bit0

Záchytný registr přerušení

ICP0 registr odpovídá portu GP0

ICP1 registr odpovídá portu GP1

V těchto registrech je obsažená informace generovaná při přerušení. Z těchto registrů lze jenom číst, zápis do nich je ignorován.

Tab. 4.11 Záchytný registr přerušení

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ICP0.7	ICP0.6	ICP0.5	ICP0.4	ICP0.3	ICP0.2	ICP0.1	ICP0.0
ICP1.7	ICP1.6	ICP1.5	ICP1.4	ICP1.3	ICP1.2	ICP1.1	ICP1.0
bit7							bit0

Kontrolní registr I/O expandéru

IARES bit kontroluje vzorkovací frekvenci na vývodech portů. Jestliže je bit v „0“, je doba pro zachycení změny na vstupu max. 32 ms, odběr proudu ze zdroje je menší. Je-li bit v „1“, doba pro zachycení změny na vstupu je max. 200μs, spotřeba je větší.

Tab. 4. 12 Kontrolní I/O registr

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
.	IARES
bit7							bit0

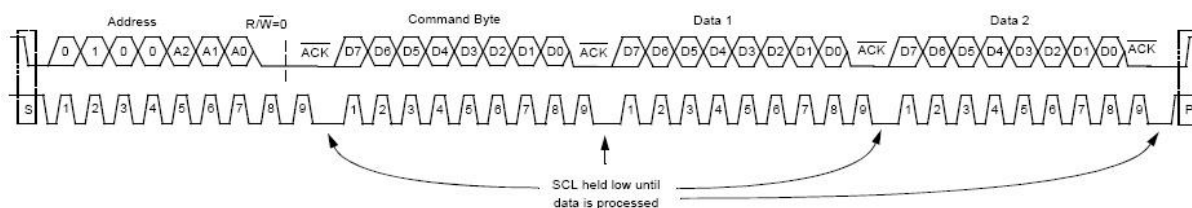
4.5.6 Logika přerušení INT

Otevřený kolektor tranzistoru na výstupu přerušení (INT) se aktivuje, když jakýkoliv vstupní pin změní svojí úroveň. To se používá pro oznámení řídicímu systému, že se stav na vstupu změnil. Jen vývody konfigurované jako vstupy mohou vyvolat přerušení. Vývody konfigurované jako výstup nemají na přerušení žádný vliv. Výstup přerušení zůstává aktivní do té doby, než čteme z portu (GPn), na kterém přerušení nastalo, nebo na příslušném registru INTCAP. Čtení stavu registrů GP0 nebo INTCAP0 nevymaže příznak přerušení vyvolaný GP1 nebo INTCAP1 a naopak. Změna vstupu na každém portu vyvolá přerušení a hodnota příslušného portu bude zachycena a překopírována do INCAP0/INCAP1. INCAPn registry jsou aktualizované jen v případě, když nastane přerušení. Tyto hodnoty zůstanou nezměněny, dokud nevymaže přerušení čtením příslušného portu nebo registru INCAPn.

Jestliže se hodnota portu vrátí do normálu ještě před tím, než je čteno přerušení INT, příznak přerušení se vymaže, nicméně registry INCAP0/INCAP1 obsahují hodnotu, která vyvolala přerušení. Jestliže se hodnota portu opět změní, je opět vyvoláno přerušení, hodnota portu je opět zachycena do INCAP.

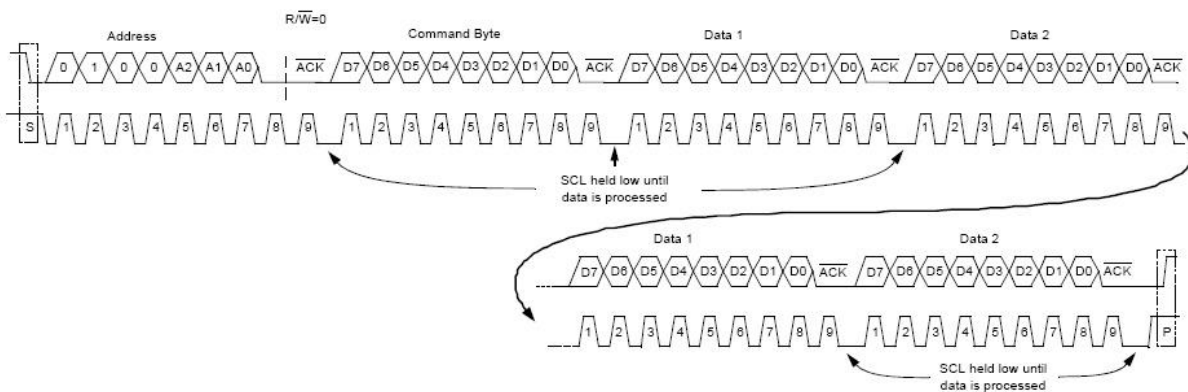
4.5.7 Zápis do registrů

Pro zapsání do registrů MCP 23016 musí master na sběrnici I²C provést následující kroky, jak jsou zobrazeny na obrázku obr. 4.11. Nejdříve musí být vybráno zařízení posláním slave adresy, která má bit R/W nastaven na log. 0. Po zaslání adresy následuje instrukce, která určí, do kterého registru se má zapisovat. MCP 23016 má dvanáct 8bitových registrů, které jsou konfigurovány, aby pracovaly jako šest 16bitových registrů. Tyto páry jsou založené na jejich funkci, například GP0 a GP1 jsou pár. První datový bajt, který následuje po instrukci, je zapsán do registru určeného instrukcí, zatím co druhý datový bajt je zapsán do druhého registru stejného páru. Například, je-li první bit poslán do OLAT1, druhý bajt bude zapsán do druhého registru stejného páru, OLAT0. Bude-li první bajt poslán do OLAT0, bude druhý bajt zapsán do OLAT1.



Obr. 4.11 Zápis do registru MCP 23016

Množství poslaných dat není ohraničeno na jeden zapisovací cyklus. Na obr. 4.12 je ukázáno vícenásobné zapsání dat do registrů na jeden zapisovací cyklus.



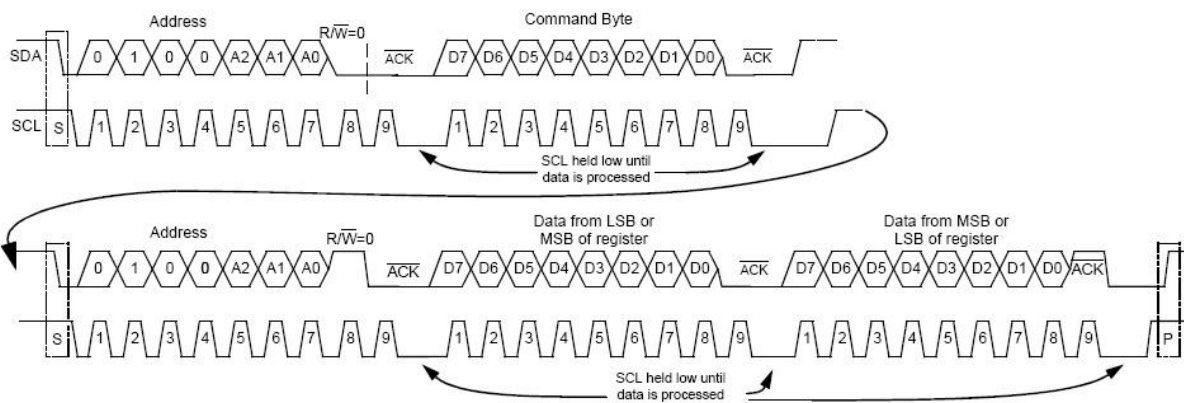
Obr. 4.12 Vícenásobný zápis do registru MCP 23016

4.5.8 Čtení z registrů

Aby bylo možné číst data z registrů MCP 23016, musí Master udělat následující kroky, které jsou zobrazeny na obr. 4.13. Nejdříve musí být obvod vybrán zasláním jeho adresy, bit R/W musí být nastaven do log. „0“. Po adrese se pošle příkaz, který určí, ze kterého registru se bude číst. Master vygeneruje START a znovu pošle adresu obvodu, bit R/W je nastaven na log. „1“. Registr definovaný příkazem v předchozím kroku čteme první a po něm následuje čtení druhého registru v páru. Logika výběru registrů je ta samá, jako při zápisu do registrů.

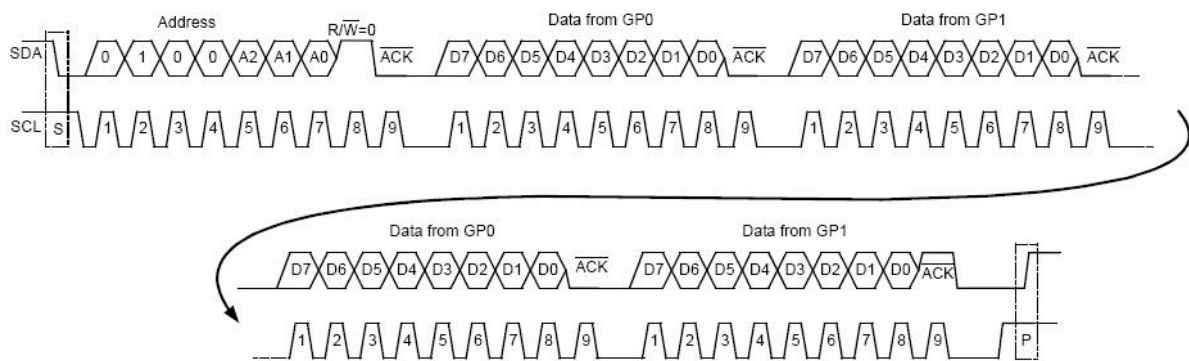
Sestupná hrana devátého hodinového impulsu inicializuje čtení z registrů. Obvod MCP 23016 drží vodič SCL hodinového impulsu „dole“, dokud nebudou data přečtena z registru a poslána do I²C kontrolního bloku.

Množství poslaných dat není ohraničeno na jeden čtecí cyklus. Na obr. 4.14 je ukázáno čtení dat z registrů.



Obr. 4.13 čtení z registrů

Na obr. 4.14 je příklad čtení ze vstupů. Příkazový bajt je vždy při této operaci nastaven na 00



Note: It is assumed that command byte is already set to 00.

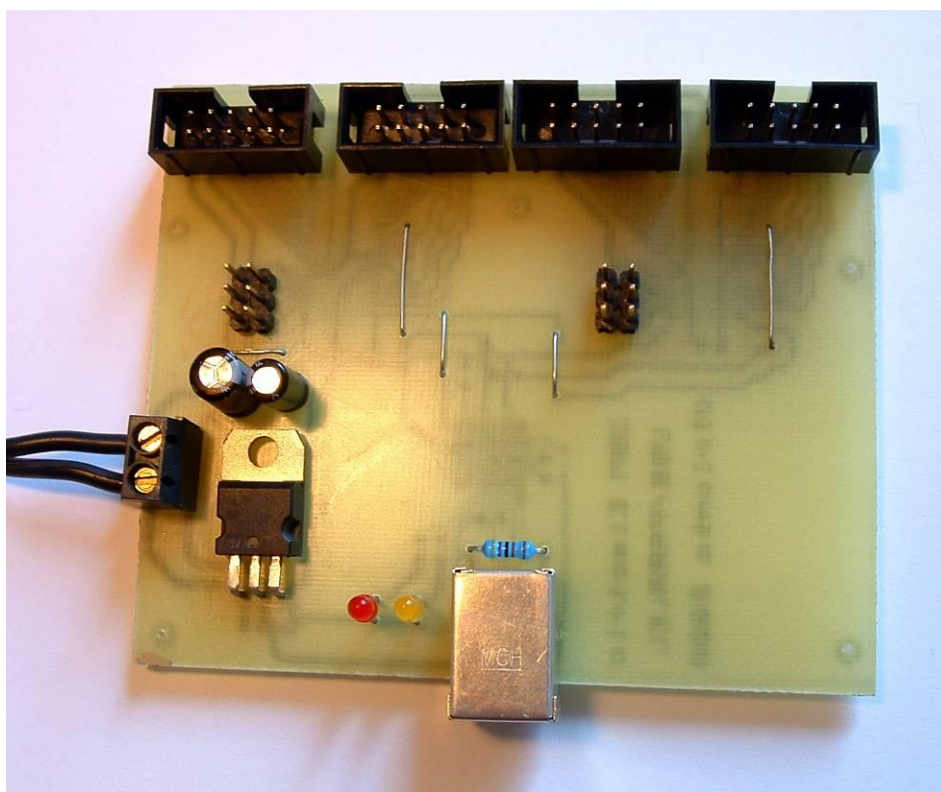
Obr. 4.14 Vícenásobné čtení z registrů (v tomto případě ze vstupů)

4.6 Realizace převodníku

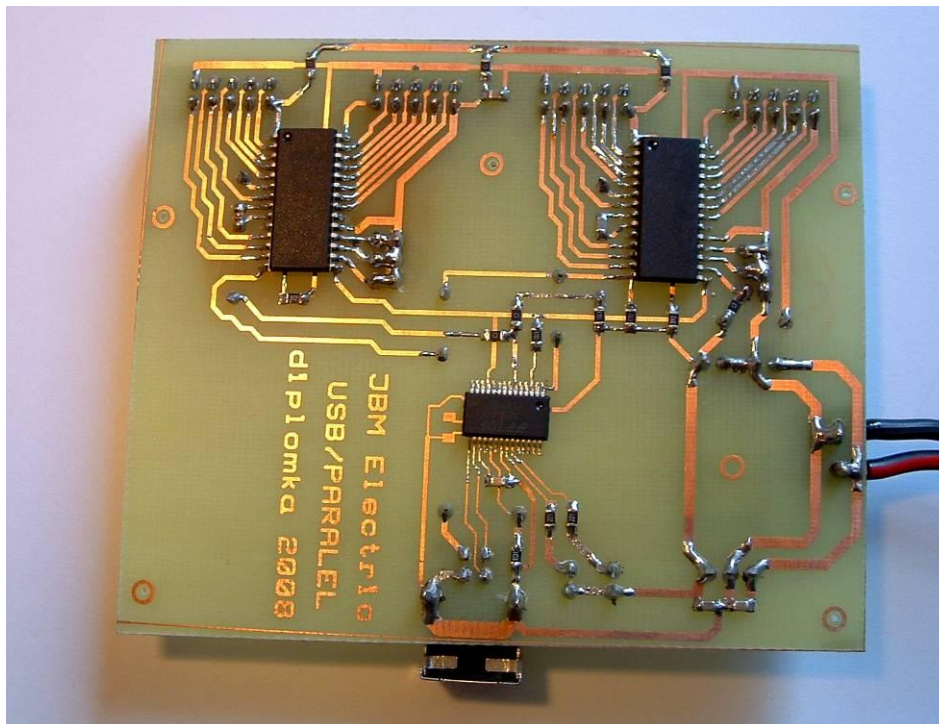
Samotný převodník je realizovaný na jednostranné desce plošných spojů obr. 4.15 a obr. 4.16. Při pohledu ze strany součástek je na desce umístěn konektor USB typu B pro připojení počítače. Vedle tohoto konektoru jsou led diody D1 a D2, které indikují příjem a vysílání dat na USB sběrnici. Dále je zde umístěna šroubovací svorka pro připojení napájecího napětí. Dále pak 4 ks konektorů PSL10, na které jsou vyvedeny jednotlivé porty GP0 a GP1 obvodů IC1 a IC2 (MCP 23016). Na straně součástek jsou dále umístěny piny pro zvolení I²C adresy daného obvodu.

Většina součástek, jak integrované obvody, tak i některé pasivní součástky jsou typu SMD (pro povrchovou montáž), a proto jsou umístěny na desce plošných spojů ze strany spojů.

Rozměry integrovaných obvodů pro povrchovou montáž jsou značně malé a zvláště pak i rozteče jejich vývodů. Z důvodu malých roztečí vývodů u integrovaného obvodu FT 232R, kde síla vývodu je 0,3 mm a rozteč 0,6 mm, se v amatérských podmínkách obtížně pájí.



Obr. 4.15 Přípravek převodníku (strana součástek)



Obr. 4.16 Přípravek převodníku (strana spojů)

Na tomto modulu byly piny pro zvolení I²C adresy propájeny ze strany plošných spojů. U obvodu IC2 jsou všechny piny pro určení adresy spojeny se zemí (GND), tudíž je jeho adresa pro zápis 0100 0000b=40hex. U obvodu IC1 je pin A1 připojen na U_{cc} napájení, piny A2 a A3 na zem (GND), tudíž je adresa obvodu pro zápis 01000010b=42hex.

Veškerá schémata a návrh plošných spojů byly realizovány v programu LSD 2000 ver.6. [10]. Soubory s návrhem schémat a spojů převodníku jsou umístěny na doprovodném CD v adresáři: \LSD\

5. Softwarové řešení komunikace s vyvinutým modulem

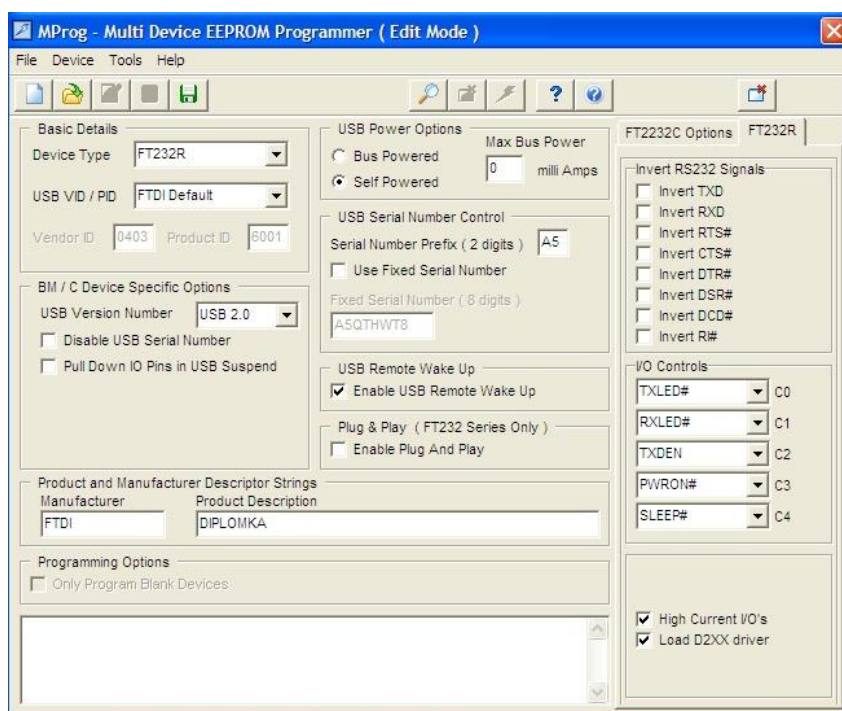
V této kapitole je popsán vývoj software pro ovládání modulu převodníku USB-parallel pomocí PC. Vyvinuté programy slouží pro názornou demonstraci funkčnosti modulu převodníku a ukázkou, jak komunikovat s modulem.

Následující software je napsán v programovacím jazyce DELPHI[®]. Tento programovací jazyk byl zvolen, protože pro obvod FT 232R je v něm napsáno mnoho praktických příkladů (jak už od výrobce obvodu, nebo v různých publikacích) a navíc vývojové prostředí Borland Delphi Personal 7[®] je zdarma.

5.1 Naprogramování EEPROM obvodu FT 232R

Protože USB podporuje technologii Plug&Play, každé zařízení připojené k počítači je automaticky rozpoznáno operačním systémem. Toto zařízení je nutné jednoznačně identifikovat. Je třeba identifikovat jeho jméno, sériové číslo, VID, PID, atd. Obvod FT 232R má tyto informace uložené v paměti EEPROM, která je již zabudovaná v obvodu. V paměti EEPROM jsou nejen uloženy informace identifikující obvod, ale je zde také nastavení obvodu.

Na obrázku 5.0 je zachyceno správné nastavení EEPROM obvodu FT 232R na převodníku. Nastavení obvodu se provádí pomocí programu MPROG verze 3.0a, který je volně ke stažení na stránkách firmy FTDI Chip [5].



Obr. 5.0 – nastavení EEPROM Obvodu FT 232R

5.2 Třída TFT232

Třída byla vytvořena pro jednodušší ovládání obvodu FT 232R. Tato třída zpřístupňuje některé funkce ovládacího rozhraní dynamické knihovny ftd2xx.dll. Úplný popis všech funkcí je uveden v dokumentu „D2XX Programmer's Guide“, který obsahuje asi padesát funkcí a nalezneme ho na stránkách výrobce FTDI Chip [5]. Popis těchto funkcí je také [11]. Třída TFT232 obsahuje jen asi dvacet funkcí a metod, které úplně postačí pro ovládání modulu převodníku. Třída byla převzata (se souhlasem autora) z [12].

Metody

- **Constructor Create(DeviceName:String)** - konstruktor, zkusí otevřít zařízení jménem **DeviceName**
- **Destruktor Destroy** – destruktork
- **Procedure SetTimeouts(ReadTimeout:DWord;WriteTimeout:DWord)** – nastaví komunikační timeouty pro příjem a pro vysílání.
- **Function ReadByte(var AByte:Byte):Boolean** – přijme jeden bajt do proměnné **AByte**, při úspěchu vrací true.
- **Function WriteByte(var AByte:Byte):Boolean** – pošle jeden bajt proměnné **AByte**, při úspěchu vrací true.
- **Function SetBitMode(Mask:Byte;Enable:Boolean):Boolean** – přepne obvod do režimu BitBang. Proměnná **Mask** je bitové pole, které určuje směr 8 vývodů (0=vstup,1=výstup). **Enable** zapíná (true) nebo vypíná (false) režim BitBang.
- **Function GetBitMode(var AByte:Byte):Boolean** – čte aktuální stav linek, nikoli obsah přijímacího bufferu. Výhodné pro čtení linek v režimu BitBang.

Vlastnosti

- **Handle:FT_HANDLE** – Handle zařízení. Lze používat i pro přímé volání funkcí z rozhraní FTD2xx.
- **Baud Rate:DWord** – Nastavuje se (i nestandardní) baudová rychlost pro komunikaci.
- **AmountInRxQueue:DWord** – Počet znaků v přijímacím bufferu.
- **AmountInTxQueue:DWord** – Počet znaků ve vysílacím bufferu.
- **CTS:Boolean** - čtení stavu linky CTS#.
- **DSR:Boolean** - čtení stavu linky DSR#.
- **RING:Boolean** - čtení stavu linky RI#.
- **RLSD:Boolean** - čtení stavu linky DCD#.
- **DTR:Boolean** - zápis stavu linky DTR#.
- **RTS:Boolean** - zápis stavu linky RTS#.
- **TxD:Boolean** - zápis stavu linky TXD#.

5.3 Třída TI2C

Jedná se o jednoduchou třídu, která umí jen posílat data na sběrnici I²C. Neobsahuje žádné speciální metody pro komunikaci, takže si programátor musí vytvořit sám sled příkazů pro daný obvod (Start, SendByte, Stop).

Třída TI2C vychází z třídy TFT232, přes kterou komunikuje s modulem. Využívá její vlastnosti pro přímé řízení linek TxD, RTS a čtení z linky CTS.

Metody

- **constructor Create(DeviceName:String)** – konstruktor, otevře zařízení jménem **DeviceName**, se kterým chceme pracovat.
- **destructor Destroy** – destruktork.
- **procedure Start** – vyšle na sběrnici signál Start.
- **procedure Stop** – vyšle na sběrnici signál Stop.
- **procedure SendByte(AData:Byte)** – vyšle jeden byte **AData** na sběrnici.

5.4 Třída **TMCP23016**

Tato třída je již vytvořená speciálně pro komunikaci s obvodem MCP 23016. Umí data do obvodu zapisovat i z něj data přečíst. Jednotlivé metody pro příjem a vysílání již v sobě obsahují správný sled příkazů pro komunikaci s obvodem (podle datasheetu), takže programátor se již nemusí zabývat programováním jednotlivých příkazů. Stačí zadat adresu obvodu a registr, ze kterého chceme číst nebo do něj zapisovat.

Třída **TMCP23016** vychází z třídy **TFT232**. Na doprovodném CD jsou dvě verze tříd:

- 1) V adresáři **..\Aplikace\Diplomka 2V1** je třída, která komunikuje s modulem pomocí vlastností **TFT232** – přímé řízení linek TxD, RTS a čtení z linky CTS.
- 2) V adresáři **..\Aplikace\Diplomka 2V2** je třída, která komunikuje s modulem pomocí režimu **BitBang**.

Obě verze tříd **TMCP23016** mají shodné metody.

Metody

- **constructor Create(DeviceName:String)** - konstruktor, otevře zařízení jménem **DeviceName**, se kterým chceme pracovat.
- **destructor Destroy** – destruktork.
- **procedure Send8Byte(Adress:Byte;Command:Byte;AData1:Byte)** – pošle jeden byte **AData1** na adresu obvodu **Adress** a do registru určeného **Command**.
- **procedure Send16Byte(Adress:Byte;Command:Byte;AData1:Byte;AData2:Byte)** - pošle byte **AData1** a **AData2** (druhý registr v páru) do registru určeného **Command**, na adresu obvodu **Adress**.
- **procedure Read8Byte(Adress:Byte;Command:Byte;var AData1:Byte)** - přečte jeden byte **AData1** z registru určeného **Command**, adresy obvodu určeného **Adress** (adresa obvodu se zadává stejná jako pro zápis, program si jí sám upraví pro čtení).
- **procedure Read16Byte(Adress:Byte;Command:Byte;var AData1:Byte;var AData2:Byte)** - přečte byte **AData1** a **AData2** (druhý registr v páru) z registru určeného **Command**, adresy obvodu určeného **Adress** (adresa obvodu se zadává stejná jako pro zápis, program si jí sám upraví pro čtení).

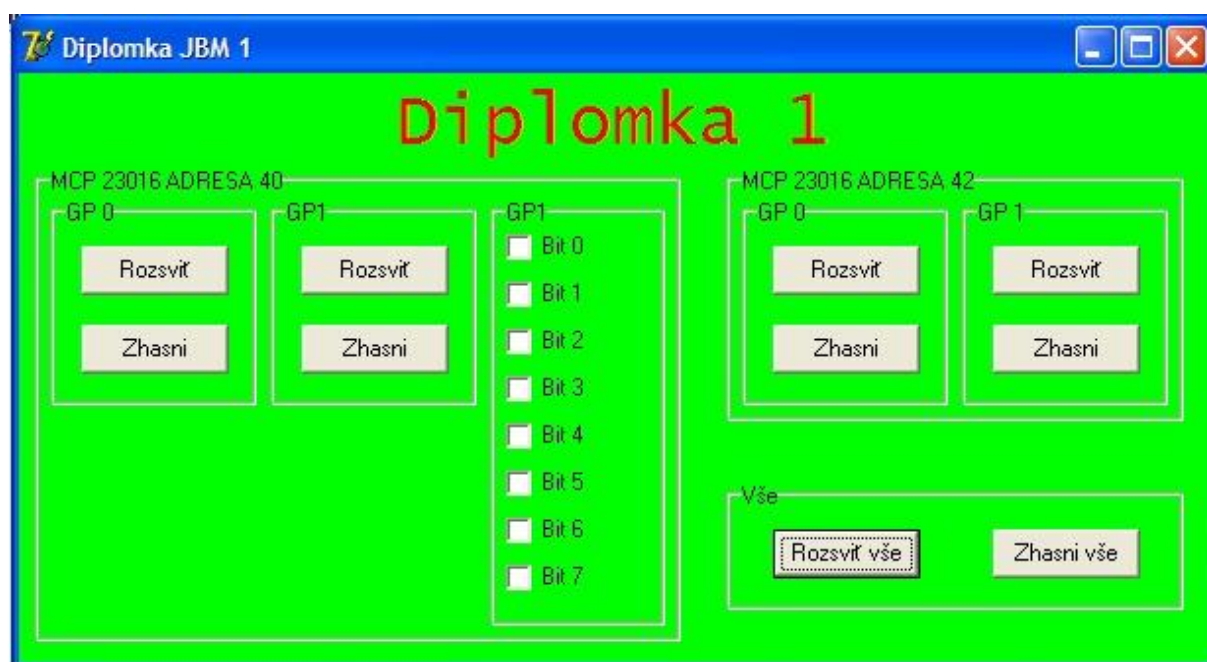
5.5 Testovací aplikace DIPLOMKA 1

Aplikace DIPLOMKA 1 ukazuje použití třídy TI2C, umí tedy pouze data do obvodu zapsat. Cílem této aplikace je ukázat ovládání vývodů GP0 a GP1 obou obvodů IC1 a IC2. Pro lepší demonstraci napěťové úrovně na výstupu obvodů IC1 a IC2 (logická 0 nebo 1), jsou na všechny výstupy připojeny přípravky, které zobrazují pomocí 8 LED stav na jednotlivých vodičích. Tyto přípravky byly publikovány v [13], jako AT8LED, odkud jsou také převzaty. V přípravku je použit pro buzení LED integrovaný obvod 74HCT245, který má negované výstupy. Svit LED diod je tudíž obrácený, přivedeme-li na vstup přípravku log. 0, LED svítí, přivedeme-li log. 1, LED zhasne.

Aplikace po spuštění vytvoří instanci třídy TI2C a ta zase instanci třídy TFT232 pro zařízení pojmenované „Diplomka“ (pokud zařízení není připojeno, zobrazí se messagebox podle obr. 5.2). Dále se pak nastaví všechny vývody na IC1 a IC2 jako výstupy a zapíše na všechny výstupy log. 1 (tedy zhasne všechny LED). Zmáčkneme-li například tlačítko „rozsvít“ v rámečku GP0, ve skupině MCP 23016 ADRESA 40, pošle aplikace do obvodu s adresou pro zápis 40h (obvod IC2), do registru GP0 data s hodnotou 00h. Na výstupech GP0.0-GP0.7 obvodu IC2 se objeví log. 0, LED na přípravku se rozsvítí.

U výstupu GP1 ve skupině MCP 23016 ADRESA 40 lze nastavit i jednotlivé bity výstupů pomocí zaškrtačacích políček, na které reaguje událost CheckBoxClick. V případě chyby komunikace se zařízením se zobrazí messagebox podle obr. 5.3.

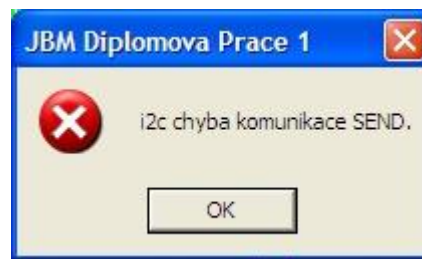
Na obrázku 5.1 je zobrazena aplikace DIPLOMKA 1.



Obr. 5.1 – Aplikace Diplomka 1



Obr. 5.2 – zařízení není k dispozici

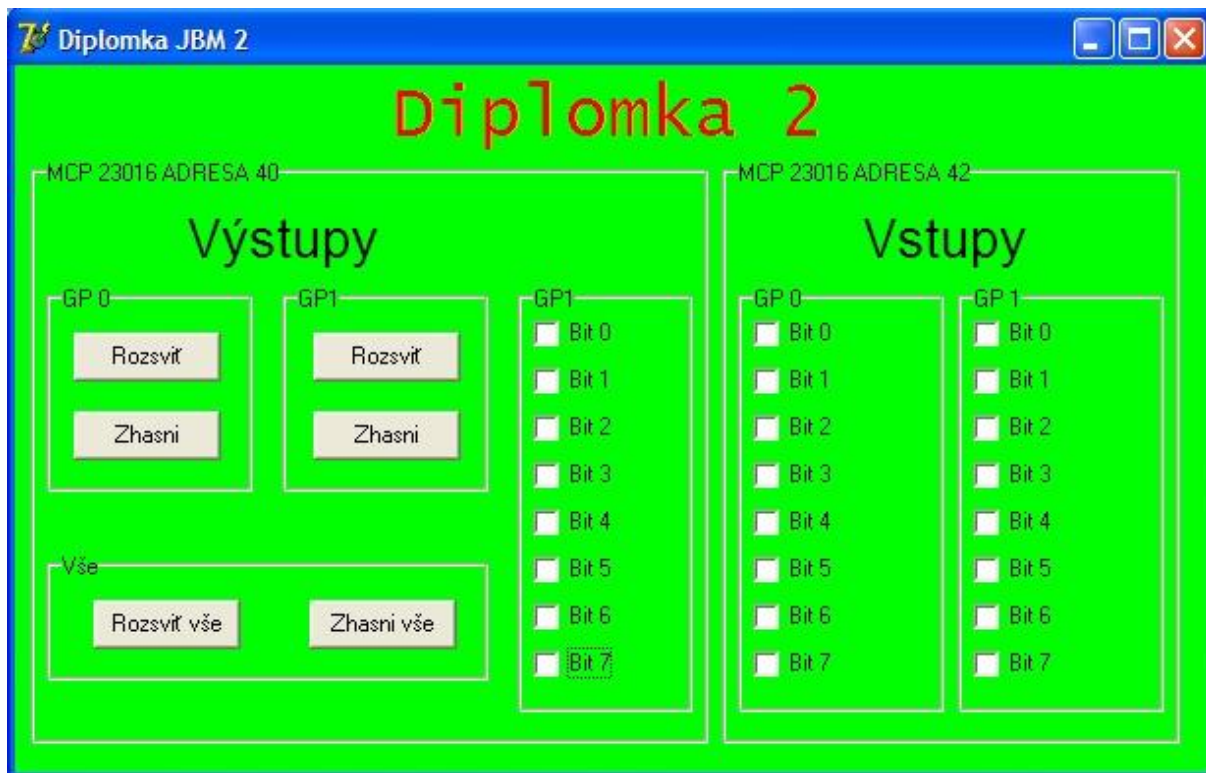


Obr. 5.3 – Chyba komunikace I²C

5.6 Testovací aplikace DIPLOMKA 2

Aplikace DIPLOMKA 2 ukazuje použití třídy TMCP23016. Umí tedy data z obvodu číst i do něj zapisovat. Pro lepší demonstraci napět'ové úrovně na výstupu obvodu IC2 (adresa 40h) je opět použit přípravek s 8 LED jako v aplikaci DIPLOMKA1. Vývody obvodu IC1 jsou tentokrát nakonfigurovány jako vstupy. Je na ně připojen přípravek ATDIPSW publikovaný v [13], který obsahuje 8 přepínačů pro volbu úrovně napětí (log. úrovně) na jednotlivých vstupech. Proti původní konstrukci jsou přepínače doplněny o pull-up rezistory, které „zvednou“ vstupy v případě rozpojení spínačů na +Ucc napětí (log. 1).

Aplikace po spuštění vytvoří instanci třídy TMCP23016 a ta zase instanci třídy TFT232 pro zařízení pojmenované „Diplomka“ (pokud zařízení není připojeno, zobrazí se messagebox podle obr. 5.2). Dále pak se nastaví vývody na IC2 jako výstupy a vývody na IC1 jako vstupy. Funkce aplikace DIPLOMKA 2 jsou podobné jako v aplikaci DIPLOMKA 1. Pro obvod IC2 (adresa 40h) jsou zde zase tlačítka pro zapnutí a vypnutí všech vodičů na GP0 nebo GP1. Pro GP1 jsou zde také Checkboxy pro nastavení jednotlivých vodičů. Obvod IC1 (adresa 42h) má vývody nastaveny jako vstupy a aplikace pouze periodicky monitoruje úrovně na jednotlivých vstupech. Aplikace obsahuje komponentu Timer, která každých 100 ms spustí proceduru **Timer1Timer**. Tato procedura přečte hodnotu na vstupech GP0 a GP1, dekoduje a zobrazí. V případě chyby komunikace se zařízením se také zobrazí messagebox podle obr. 5.3 chyba komunikace I²C. Na obrázku 5.4 je zobrazena aplikace DIPLOMKA 2.



Obr. 5.4 – Aplikace DIPLOMKA2

6. Testování

Funkčnost zařízení jsem kontroloval po jednotlivých krocích fyzické realizace, tzn. nejdříve v průběhu osazování desky, potom jsem testoval komunikaci s řadičem FT 232R a nakonec komunikaci s obvodem MCP 23016 přes I²C.

6.1 Testy hardwaru

Prvním krokem při osazování desky plošných spojů bylo oživení zdroje. Po napájení všech nezbytných součástí jsem modul připojil na stabilizovaný zdroj napětí, na kterém bylo nastaveno napětí 9 V a proud omezen na 100 mA. Deska nevykazovala žádný zkrat a napětí za regulátorem bylo 5 V. Po kontrole zdroje byl osazen zbytek desky s následným opětovným měřením zkratů v napájení a odběru proudu ze zdroje. Poté jsem ještě změřil voltmetrem úroveň všech zbývajících signálů SDA, SCL a INT, které mají mít definovanou úroveň 5 V.

6.2 Testy komunikace

Po připojení modulu na počítač jsem nejdříve musel nainstalovat správné ovladače a naprogramovat EEPROM v obvodu FT 232R. Poté jsem si vytvořil jednoduchou aplikaci, se kterou jsem v režimu BitBang ovládal jednotlivé vodiče SDA a SCL. Tím jsem si potvrdil funkčnost obvodu FT 232R a komunikaci s ním. Dále jsem napsal třídu TI2C a pomocí jednoduché aplikace poslal sled příkazů na obvod IC1. Během testování komunikace

s obvodem IC1 byla objevena chyba v zapojení. Při návrhu převodníku jsem mylně předpokládal, že pro volbu adresy obvodu MCP 23016 musí být vývody A0-A2 pevně spojeny se zemí GND (pro volbu log. 0), nebo ponechány nepřipojené (pro volbu log.1). Správné zapojení pro volbu log. 1 je propojit vývod A0-A2 přímo s napájením +Ucc. Na desce plošných spojů je proto jen možnost propojení vývodů se zemí GND, v případě, že chceme propojit vývod s +Ucc, musíme to provést pomocí drátové propojky. Po úpravě zapojení (připojení vývodu A1 obvodu IC1 na +Ucc) je již možné adresovat i obvod IC1. Změřením úrovně napětí na výstupech IC1 jsem si potvrdil správnost komunikace s tímto obvodem.

Při testech komunikace jsem si všiml, že rychlost s jakou se rozsvěčují diody LED, není velická. Již pouhým okem bylo poznat zpoždění v rozsvěcení jednotlivých portů při zmáčknutí tlačítka „rozsviť vše“. Zkoušel jsem tento nedostatek odstranit vymazáním čekacích procedur z programu, které měly za úkol chvíli počkat s posíláním dalších dat, než se ustálí napětí na vývodech FT 232R. Rychlost převodníku se touto úpravou zvýšila, ale pořád je patrné malé zpoždění při rozsvěcení LED. K dalšímu zlepšení rychlosti komunikace již nepomohlo ani zvyšování rychlosti sériové komunikace obvodu FT 232R pomocí vlastnosti **TFT232.BaudRate**.

Práce na těchto úkolech pak potvrdila funkčnost celého zařízení.

7. Možné praktické aplikace vyvinutého modulu

Vyvinutý převodník se spíše hodí použít na časově nenáročné aplikace. Rychlost komunikace modulu je ovlivněna dvěma faktory:

1. Přenosová rychlost modulu – závisí na maximální rychlosti komunikace s obvodem FT 232R, následným převodem na sériovou komunikaci I²C a v poslední řadě také přenosovou rychlostí obvodu MCP 23016.
2. Přidělený čas operačním systémem (OS) – OS přiděluje každé aplikaci určitý časový úsek, a pokud bude OS zaneprázdněn nějakou jinou činností, odloží OS i komunikaci s modulem.

S přihlédnutím k výše zmíněným omezením závisí použití tohoto převodníku prakticky už jen na fantazii návrháře. Uplatnění najde především tam, kde je potřeba k počítači připojit nějaké nestandardní zařízení na paralelní port a na počítači již není tento port k dispozici.

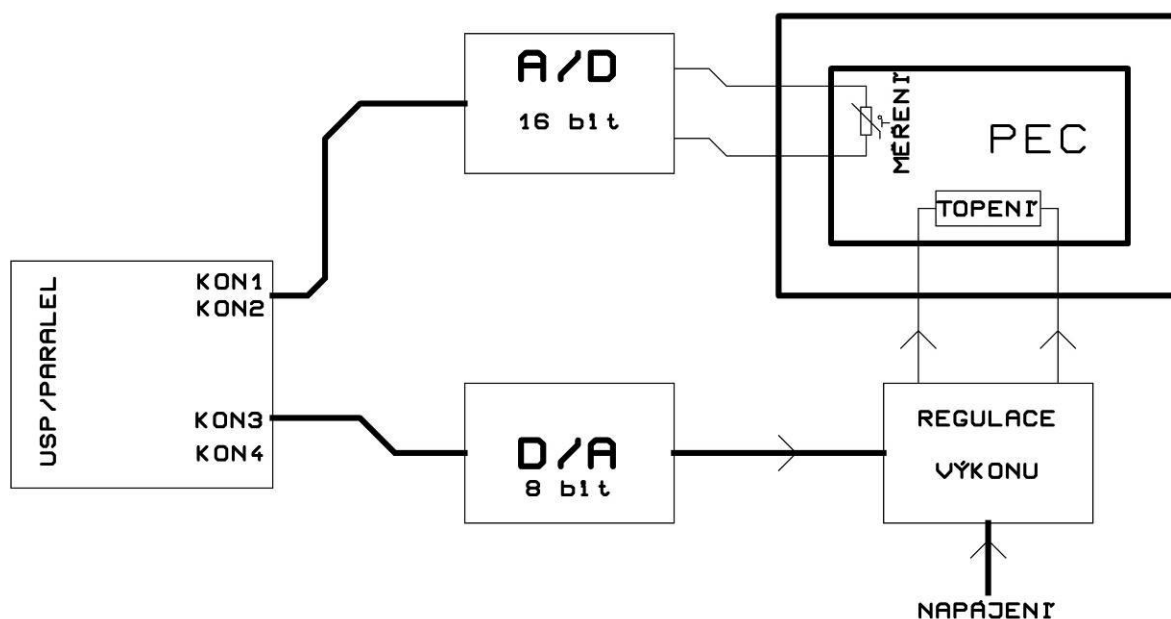
7.1 Příklady praktického využití modulu

7.1.1 Příklad 1 – řízení teploty v peci

Na obrázku 7.1 je blokově znázorněn jednoduchý příklad řízení teploty v peci. Obvod IC2 má vývody nakonfigurovány jako vstupy. Na tyto vstupy je připojen 16bit A/D převodník, který převádí signál z teplotního čidla v peci. Obvod IC1 má nakonfigurované vývody jako výstupy, na které je připojen D/A převodník. Výstup z tohoto převodníku je připojen na výkonovou řídicí část, která řídí vytápění pece (dodaný výkon). Řízení vytápění by mohlo být plynulé, jako v tomto případě, nebo skokové. Pokud by bylo skokové, řídilo by se přepínání jednotlivých článků topení podle potřeby.

V případě tohoto řešení se může využít řízení počítačem například k:

- ✚ Zaznamenávání průběhu teploty při vytápění nebo chladnutí pece.
- ✚ Vytápění nebo chladnutí podle dané křivky.
- ✚ Zapínání a vypínání v předem navolený čas.



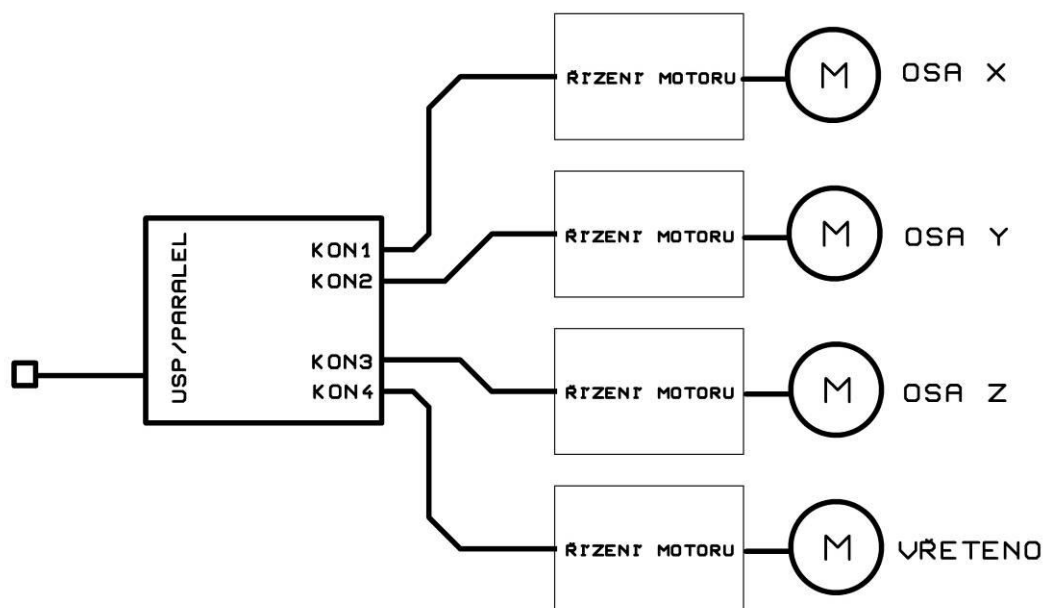
Obr. 7.1 Řízení vytápění pece

7.1.2 Příklad 2 - Řízení trojosého obráběcího stroje

Protože má převodník dostatečný počet vývodů, může být použit i na řízení trojosého obráběcího stroje, který je blokově znázorněn na obrázku 7.2. Jednotlivé porty převodníku jsou použity na řízení krokových motorů. Tyto porty mohou být využity i pro vstup snímačů

krajních poloh a referenčního bodu jednotlivých os stroje. Poslední port převodníku je využit pro regulaci otáček obráběcího nástroje (vřetene).

Samotné řízení stroje je potom otázkou vhodného počítačového programu.



Obr. 7.2 Řízení trojosého obráběcího stroje

8. Závěr

Jedním z cílů této bakalářské práce bylo seznámení s problémem USB interface. Byla provedena analýza problému konverze z USB sběrnice na 16-ti bitovou paralelní sběrnici. Na základě této analýzy byl navržen modul sériově-paralelního převodníku. Tento vyvinutý modul byl kompletně zdokumentován, realizován a oživen, což bylo hlavním cílem práce.

Konstrukce převodníku vychází z integrovaného obvodu FT232R, který provádí vlastní konverzi z USB sběrnice na sériovou I²C. Jako budiče výstupů bylo použito obvodu MCP 23016, který převádí signál ze sériové sběrnice I²C na paralelní. Použitím 2ks obvodu MCP23016 bylo získáno 32 datových vodičů. Zvýšení počtu vývodů na 32 a možnost programově si navolit, který vývod bude vstup nebo výstup, rozšířilo okruh uplatnění modulu převodníku.

Dále byl popsán vývoj software pro ovládání modulu převodníku pomocí PC. Vyvinuté programy slouží pro názornou demonstraci funkčnosti modulu převodníku a ukázkou, jak komunikovat s modulem.

Na závěr práce jsou uvedeny možné praktické aplikace tohoto převodníku. V prvním příkladu je znázorněno jednoduché řízení teploty v peci. V druhém příkladu je znázorněno řízení trojosého obráběcího stroje.

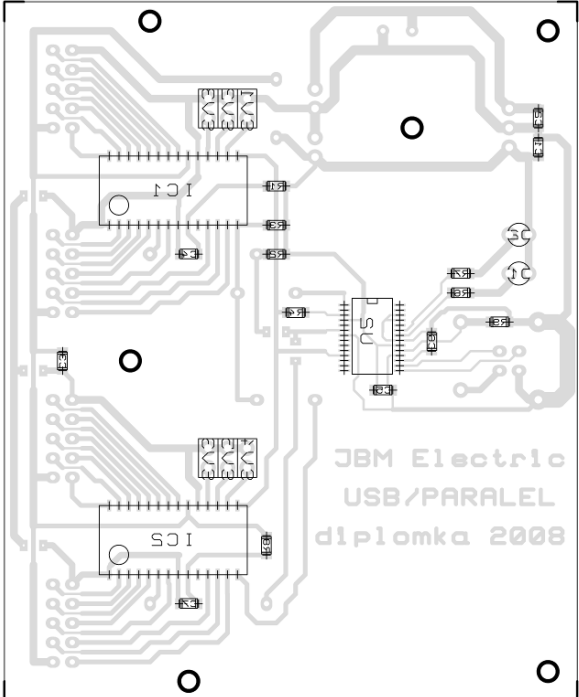
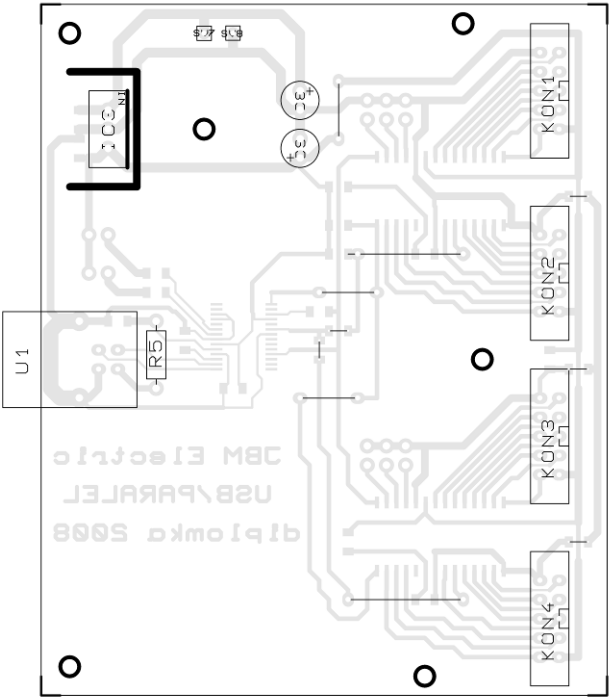
Použitá literatura

- [1] <http://www.usb.org> – oficiální stránky Universal Serial Bus
- [2] <http://www.atmel.com>, stránky výrobce Atmel
- [3] <http://www.obdev.at/products/avrusb/index.html>, USB Driver for Atmel AVR Microcontrollers
- [4] [http://www.cesko.host.sk/IgorPlugUSB/IgorPlug-USB%20\(AVR\).htm](http://www.cesko.host.sk/IgorPlugUSB/IgorPlug-USB%20(AVR).htm), Implementácia USB do mikrokontroléra: IgorPlug-USB (AVR)
- [5] <http://www.ftdichip.com>, Future technology international Ltd.
- [6] <http://www.asix.cz>, Asix s.r.o.
- [7] <http://www.nxp.com/>, Philips Semiconductors / NXP Semiconductors
- [8] <http://www.cypress.com/>, Cypress Semiconductor
- [9] <http://www.microchip.com>, Microchip technology Inc.
- [10] <http://www.lsd2000.cz/>, český návrhový systém pro elektroniku.
- [11] David Matoušek, USB prakticky s obvody FTDI – 1. díl, BEN – technická literatura, Praha 2003
- [12] David Matoušek, Udělejte si z PC v Delphi – 1. díl, BEN – technická literatura, Praha 2003
- [13] David Matoušek, Práce s mikrokontroléry ATMEL AT89C2051, BEN – technická literatura, Praha 2002

Příloha A – Přehled mikrokontrolérů s USB

Procesor	Výrobce	Popis	USB spec.	USB Rychlost
M5617	Ali	Full speed USB Scanner controller	1.0	Full
M5623	Ali	High-speed USB 2.0 scanner controller	2.0	High
M5625	Ali	High-speed USB 2.0 integrated scanner controller	2.0	High
ADSP-2192M	Analog Devices	Dual core DSP Microcomputer	1.1	Full
AT90USB1286	Atmel	USB Function Microcontroller	2.0	Full
AT90USB1287	Atmel	USB On-The-Go (OTG) Microcontroller	2.0	Full
AT90USB162	Atmel	USB Microcontroller	2.0	Full
AT90USB646	Atmel	USB Function Microcontroller	2.0	Full
AT90USB647	Atmel	USB On-The-Go (OTG) Microcontroller	2.0	Full
AT90USB82	Atmel	USB Microcontroller	2.0	Full
ATmega16U4	Atmel	MCU 2.0 Full-speed/Low Speed Device Module	2.0	Full
ATmega32U4	Atmel	2.0 Full-speed/Low Speed Device Module	2.0	Full
AT83C5134	Atmel	USB C51-based Microcontroller	2.0	Full
AT83C5135	Atmel	USB C51-based Microcontroller	2.0	Full
AT83C5136	Atmel	USB C51-based Microcontroller	2.0	Full
AT89C5130A-M	Atmel	USB C51-based Microcontroller	2.0	Full
AT89C5131A-L	Atmel	USB C51-based Microcontroller	2.0	Full
AN2131	Cypress	EZ-USB Microcontroller	1.1	Full
CY7C63221A	Cypress	enCoRe USB Low speed peripheral controller	2.0	Low
CY7C63411	Cypress	Low speed peripheral USB controller	2.0	Low
CY7C63412	Cypress	Low speed peripheral USB controller	2.0	Low
CY7C63413	Cypress	Low speed peripheral USB controller	2.0	Low
CY7C63511	Cypress	Low speed peripheral USB controller	2.0	Low
HD6432210	Hitachi	High performance microcontroller with H8S/2000 CPU	1.1	Full
HD6432211	Hitachi	High performance microcontroller with H8S/2000 CPU	1.1	Full
HD6432215	Hitachi	High performance microcontroller with H8S/2000 CPU	1.1	Full
HD64F2212	Hitachi	High performance microcontroller with H8S/2000 CPU	1.1	Full
HD64F2215	Hitachi	High performance microcontroller with H8S/2000 CPU	1.1	Full
HD64F2217	Hitachi	High performance microcontroller with H8S/2000 CPU	1.1	Full
8x930Ax	Intel	USB Microcontroller	1.1	Full
8x931AA	Intel	USB peripheral microcontroller	1.1	Full
PIC18F13K50	Microchip	USB Flash Microcontrollers with nanoWatt Technology	2.0	Full
PIC18F14K50	Microchip	USB Flash Microcontrollers with nanoWatt Technology	2.0	Full
PIC18F2553	Microchip	USB 8-bit PIC® Microcontrollers	2.0	Full
PIC18F4450	Microchip	USB 8-bit PIC® Microcontrollers	2.0	Full
PIC18F4455	Microchip	USB 8-bit PIC® Microcontrollers	2.0	Full
PIC18F4458	Microchip	USB 8-bit PIC® Microcontrollers	2.0	Full
PIC18F4550	Microchip	USB 8-bit PIC® Microcontrollers	2.0	Full
UDA 1321	Philips	USB analog to digital converter	1.1	Full
UDA1325	Philips	USB audio codec	1.1	Full
UDA1331H	Philips	Audio playback peripheral	1.1	Full

Příloha B – Obrázky osazení plošného spoje



Příloha C – Seznam obrázků

Obr. 2.1 USB konektory.....	8
Obr. 2.2 Kabel pro USB.....	8
Obr. 2.3 Definice Low speed zařízení.....	9
Obr. 2.4 Definice Full speed zařízení.....	10
Obr. 2.5 Diferenční zapojení vysílačů.....	10
Obr. 2.6 Hvězdicová struktura rozbočování USB.....	11
Obr. 2.7 Příklad jednoduché struktury.....	12
Obr 3.1 Typ zapojení na USB sběrnici.....	16
Obr 4.1 schéma převodníku.....	21
Obr 4.2 Blokové schéma obvodu FT232R.....	24
Obr. 4.3 schematická značka FT 232RL.....	26
Obr. 4.4 Struktura sběrnice I ² C.....	28
Obr. 4.5 Přenos bitů Obr. 4.6 Řídící signály.....	29
Obr. 4.7 Přenos jednoho bajtu na I ² C.....	29
Obr. 4.8 Komunikace po sběrnici.....	29
Obr. 4.10 Zapojení generátoru hodin.....	31
Obr. 4.9 vývody MCP 23016.....	31
Obr. 4.11 Zápis do registru MCP 23016.....	35
Obr. 4.12 Vícenásobný zápis do registru MCP 23016.....	35
Obr. 4.13 čtení z registrů.....	36
Obr. 4.14 Vícenásobné čtení z registrů (v tomto případě ze vstupů).....	36
Obr. 4.15 Přípravek převodníku (strana součástí).....	37
Obr. 4.16 Přípravek převodníku (strana spojů).....	38
Obr. 5.0 – nastavení EEPROM Obvodu FT 232R.....	39
Obr. 5.1 – Aplikace Diplomka 1.....	42
Obr. 5.2 – zařízení není k dispozici Obr. 5.3 – Chyba komunikace I ² C.....	43
Obr. 5.4 – Aplikace DIPLOMKA2.....	44
Obr. 7.1 Řízení vytápění pece.....	46
Obr. 7.2 Řízení trojosého obráběcího stroje.....	47