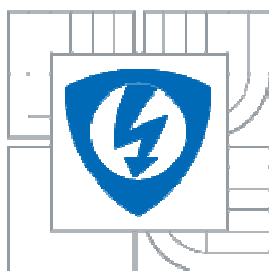


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

ROTAČNÍ OVLADAČ K POČÍTAČI

ROTARY CONTROLLER FOR PC

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

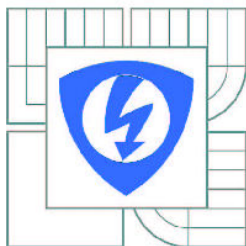
AUTOR PRÁCE
AUTHOR

LUKÁŠ OLIVÍK

VEDOUCÍ PRÁCE
SUPERVISOR

prof. Dr. Ing. ZDENĚK KOLKA

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Bakalářská práce

bakalářský studijní obor
Elektronika a sdělovací technika

Student: Lukáš Olivík
Ročník: 3

ID: 106682
Akademický rok: 2009/2010

NÁZEV TÉMATU:

Rotační ovladač k počítači

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se se sběrnici USB a vyberte vhodný typ mikrokontroléru (AVR, PIC, Freescale) s rozhraním USB, který umožní snímání úhlového natočení ovládacích prvků prostřednictvím inkrementálního snímače.

Vytvořte firmware, který bude realizovat periférii typu HID. Vytvořte příslušné deskriptory. Při návrhu dbejte na dodržení požadavků std. USB 2.0.

Zařízení realizujte po mechanické stránce se dvěma snímači, tlačítky a LED. Vytvořte demonstrační program pro PC pro ověření funkčnosti.

DOPORUČENÁ LITERATURA:

[1] AXELSON, J. USB Complete The Developer's Guide Fourth Edition, Madison: Lakewiev Research LLC, 2009, 504 s. ISBN13 978-1-931448-08-6

Termín zadání: 8.2.2010

Termín odevzdání: 28.5.2010

Vedoucí práce: prof. Dr. Ing. Zdeněk Kolka

prof. Dr. Ing. Zbyněk Raida
Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Tato práce popisuje sběrnici USB a konstrukci jednoduché HID periferie, připojující rotační enkodér jako ovládací prvek k počítači. Zahrnuje přehled vhodných mikroprocesorů, návrh zapojení, plošného spoje a mechanického uspořádání, vytvoření programu pro vybraný mikroprocesor, a programování ukázkového software na straně počítače.

KLÍČOVÁ SLOVA

USB, HID, rotační enkodér, PIC18

ABSTRACT

This thesis describes the Universal Serial Bus and the development of a simple HID peripheral device, connecting a rotary encoder as a control element to the host computer. This paper includes an overview of suitable microprocessors, circuit design, PCB and mechanical design, creation of a firmware for the selected microprocessor, and programming of demonstration software on the PC's side.

KEYWORDS

USB, HID, rotary encoder, PIC18

OLIVÍK, L. *Rotační ovladač k počítači*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 38 s., 3 s. příloh. Vedoucí bakalářské práce prof. Dr. Ing. Zdeněk Kolka.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma Rotační ovladač k počítači jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne 28. května 2010

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce prof. Dr. Ing. Zdeňku Kolkovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne 28. května 2010

.....

(podpis autora)

OBSAH

1	ÚVOD.....	1
2	POPIS SBĚRNICE USB.....	2
2.1	SPECIFIKACE USB.....	3
2.2	NAPÁJENÍ PŘES USB.....	4
2.3	TYPY PŘENOSŮ NA SBĚRNICI.....	4
2.4	TŘÍDY USB ZAŘÍZENÍ.....	6
3	REALIZACE.....	7
3.1	VÝBĚR MIKROPROCESORU.....	7
3.2	POPIS VYBRANÉHO MIKROPROCESORU.....	8
3.3	ROTAČNÍ ENKODÉRY.....	10
3.4	POPIS ZAPOJENÍ.....	11
3.5	NÁVRH PLOŠNÉHO SPOJE A MECHANICKÉ KONCEPCE.....	13
4	FIRMWARE PROCESORU.....	15
4.1	ZÁPIS PROGRAMU DO MIKROPROCESORU.....	15
4.2	DESKRIPTORY.....	15
4.3	OBSAH REPORTU.....	19
4.4	KONFIGURACE USB MODULU.....	19
4.5	KONFIGURACE PROCESORU.....	20
4.6	Hlavní funkce firmwaru (main), proměnné.....	21
4.7	TESTOVÁNÍ SPLNĚNÍ SPECIFIKACE USB 2.0.....	25
4.8	ALTERNATIVNÍ FIRMWARE.....	26
4.9	JINÉ MOŽNOSTI TVORBY FIRMWARE.....	27
5	PROGRAM NA STRANĚ PC.....	28
5.1	UKÁZKOVÉ KÓDY OD VÝROBCE PROCESORU.....	28
5.2	UŽIVATELSKÉ ROZHŘANÍ APLIKACE, PROMĚNNÉ.....	28
5.3	DETEKCE PŘIPOJENÍ ZAŘÍZENÍ.....	30
5.4	KOMUNIKACE PŘES SBĚRNICI USB.....	31
5.5	ZNÁMÉ PROBLÉMY, CHYBY A OMEZENÍ.....	32
6	ZÁVĚR.....	33
	SEZNAM POUŽITÉ LITERATURY.....	34
	SEZNAM SYMBOLŮ A ZKRATEK.....	36
	OBSAH PŘÍLOŽENÉHO CD.....	37
	SEZNAM PŘÍLOH.....	38

1 ÚVOD

Cílem práce je navrhnout a vyrobit ovladač se dvěma rotačními enkodéry a čtyřmi tlačítky, který se připojí k počítači. Ovladač bude sloužit například k ovládání procesů měření. Ovladač je doplněn o čtyři indikační LED diody, na kterých je také demonstrován přenos dat z počítače do ovladače.

Ovladač bude připojen k rozhraní USB. USB (Universal Serial Bus) je nejrozšířenější rozhraní pro připojení periférií k PC. Uvedením USB v roce 1998 bylo nahrazeno několik starších rozhraní, např. paralelní port LPT, sériový port RS-232, Gameport a PS/2. Výhodou USB oproti starším rozhraním je plná podpora Plug&Play. To umožňuje připojení nebo odpojení zařízení kdykoliv za chodu počítače, bez nutnosti restartu. USB je tedy uživatelsky přívětivé.

Uživatelská přívětivost je vykoupena výrazně vyšší náročností vývoje takového zařízení. Nutnou součástí každého USB zařízení je řídicí integrovaný obvod. Tím může být buď jednoúčelový speciální obvod, nebo univerzální procesor. Potřebám tohoto ovladače nevyhovuje žádný speciální obvod, a proto je nutné použít mikroprocesor. Na trhu je k dispozici celá řada procesorů lišících se výpočetním výkonem, velikostí paměti, množstvím periférií, velikostí pouzdra i cenou. Pro dané zařízení bude dostačují některý z osmibitových mikroprocesorů s integrovaným USB řadičem. Výběr mikroprocesoru je podrobněji popsán v kapitole 3.1. Pro vybraný procesor je nutné následně vytvořit firmware. Tomu je věnována kapitola 4. USB je velmi komplexní rozhraní, a proto i firmware procesoru musí obsahovat mnoho různých funkcí. Výrobci procesorů proto zpravidla dodávají předpřipravené zdrojové kódy alespoň pro základní funkčnost. Každé zařízení musí mít ve své paměti uloženy tzv. deskriptory. To jsou bloky dat popisující zařízení a jeho funkce, které si řídicí počítač vyžádá po připojení zařízení. Deskriptory musí mít přesně danou strukturu, kterou je nutné pro správnou funkci zařízení nutně dodržet. Proto je jejich vytvoření poměrně složité. Deskriptory jsou popsány v kap. 4.2.

USB zařízení se řadí do tříd. Tento ovladač bude zajišťovat interakci mezi uživatelem a počítačem, a proto se přímo vybízí zařadit jej do třídy HID (human interface device – zařízení pro styk s uživatelem). Výhodou této třídy je snadná instalace bez potřeby dodání ovladačů, jak je blíže popsáno v kapitole 2.4.3. Zařazení do této třídy však přináší další komplikaci ve vývoji zařízení, kterou je nutnost vytvoření dalšího speciálního, tzv. report deskriptoru. Ten popisuje strukturu následně přenášených dat, kterou je následně nutné dodržovat. Report deskriptor je stručně popsán v kap. 4.2.7

USB také dodává napájecí napětí 5V, s dovoleným odběrem až 500 mA. Tento ovladač bude mít odběr pouze několik desítek mA a proto bude napájen z USB (tzv. bus-powered). Více je o napájení napsáno v kapitole 2.2

Posledním krokem ve vývoji USB zařízení je vytvoření ovládacího programu na straně počítače, které je popsáno v kapitole 5.

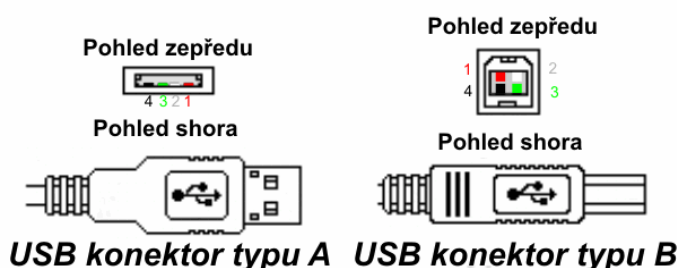
Cílem je navrhnout zařízení tak, aby plně vyhovovalo specifikaci USB v současné verzi 2.0.

2 POPIS SBĚRNICE USB

Řídícím prvkem na sběrnici je host. Na sběrnici je vždy jen jeden host. Hostem se obvykle rozumí PC, ale může jím být i např. autorádio s USB vstupem, které umí přehrávat hudební soubory z připojeného flash disku. K hostu se připojují zařízení (device).

K jednomu hostu lze připojit až 127 zařízení. Dostupný počet portů lze zvýšit rozbočovačem – hubem.

Host je vybaven alespoň jedním konektorem typu A (zásuvkou). Zařízení může mít zásuvku typu B, a připojovací kabel A-B, nebo napevno zabudovaný kabel se zástrčkou typu A. Oba typy konektorů jsou zobrazeny na Obr. 1.



Obr. 1: Konektory USB (převzato z [3])

Kabel se skládá z dvojice napájecích vodičů (VBUS a GND) a datového diferenciálního páru (D+ a D-). Maximální délka kabelu je 5 m. Zapojení kabelu je uvedeno v Tabulka 1.

Tabulka 1: Popis vodičů sběrnice USB 2.0

Pořadí vodiče	Název	Význam	Barva vodiče
1	VBUS	Napájení +5 V	červená
2	D-	Datový vodič -	bílá
3	D+	Datový vodič +	zelená
4	GND	zem	černá

USB nabízí možnost volby ze čtyř přenosových rychlostí: 1,5 Mb/s (Low speed), 12 Mb/s (Full speed), 480 Mb/s (High speed), a nově dle specifikace USB 3.0 i 5 Gb/s (Superspeed). Pro daný ovladač, kde není požadavek na přenos velkých objemů dat, postačuje i přenosová rychlost 1,5 Mb/s. Tato rychlost je často využívána u myši. Umožňuje použití nestíněného, tenčího a levnějšího kabelu. Přináší však i určitá omezení, například maximální velikost datového paketu pouze 8 bytů, jak je popsáno např. v [1]. Proto je zvolena rychlost Full speed, tedy 12 Mb/s. Z pohledu uživatele není při potřebě přenášet malé objemy dat mezi těmito dvěma rychlostmi žádný rozdíl.

Zařízení ohlásí svou přítomnost na sběrnici připojením zdvihacího (pull-up) odporu mezi napětí 3,3 V a jednu z datových linek – pomalá zařízení (Low speed) připojují pull-up na datovou linku D-, zařízení Full speed a High speed pak na D+. Po zjištění nového zařízení probíhá tzv. enumerace – to je proces, při kterém si host vyžádá informace uložené v deskriptorech zařízení. Po identifikaci zařízení host přiřadí zařízení adresu a načte příslušné ovladače, nebo si je vyžádá od uživatele. Zařízení je pak připraveno k použití.

Komunikace s Low speed a Full speed zařízeními probíhá ve framech s intervalem 1 ms, High speed zařízení komunikují v microframech s intervalem 125 μ s. Data skrz sběrnici tečou pomyslnými rourami (pipes). Rozlišují se dva typy rour – roury zpráv (message pipes), u kterých mají data danou strukturu, a proudové roury (stream pipes), u kterých není struktura dat přesně stanovena. Zdrojem nebo příjemcem dat v zařízení je endpoint (koncový bod). To je vyhrazená oblast paměti, ve které jsou uložena přijatá data nebo data k odeslání. Každý endpoint má daný směr buď od zařízení k hostu – IN, nebo od hosta k zařízení – OUT. Každé zařízení obsahuje alespoň jeden endpoint 0 v každém směru, a jednu výchozí rouru (default control pipe).

Komunikaci vždy zahajuje host, a zařízení musí odpovídat na jeho požadavky. Zařízení nesmí zahájit komunikaci bez požadavku od hosta.

USB používá systém kódování NRZI (Non Return to Zero Invert). To znamená že logickou 0 reprezentuje změna úrovně, logickou 1 pak reprezentuje signál stejné úrovně. Aby byla zachována synchronizace, je za každých 6 log. 1 v řadě vložena změna úrovně (bit stuffing). Tato změna úrovně je automaticky odstraněna při dekodování NRZI. Podrobnější popis je dostupný v literatuře [1],[2],[3],[4].

2.1 Specifikace USB

Aktuální specifikace je vždy dostupná online na stránkách USB-IF [2].

2.1.1 USB 1.0

Tato specifikace byla představena v lednu 1996. První operační systém s podporou USB byl Windows 95 OEM Service Release 2. Specifikace Definuje přenosové rychlosti Low speed (1,5 Mb/s) a Full speed (12 Mb/s).

2.1.2 USB 1.1

Specifikace USB 1.1 byla uvedena v září 1998 a nahradila předchozí verzi 1.0. Přidala podporu nového typu přenosu – Interrupt OUT (Přerušovací přenos směrem od hosta k zařízení).

2.1.3 USB 2.0

Specifikace USB 2.0 z dubna 2000 nahrazuje předcházející verze. Tato specifikace je i v dnešní době stále aktuální. Zachovává zpětnou kompatibilitu se staršími verzemi. Přináší novou přenosovou rychlost High speed (480 Mb/s). High speed zařízení po připojení na host USB 1.1 funguje na rychlosti Full speed.

2.1.4 USB 3.0

Specifikace USB 3.0 byla uvedena v prosinci roku 2008. Na rozdíl od předchozích nenahrazuje starší specifikace, ale doplňuje specifikaci 2.0. Přináší novou přenosovou rychlost SuperSpeed (5 Gb/s). Pro dosažení této rychlosti bylo nutné vyvinout nové kabely a konektory, byly přidány dva datové diferenciální páry pro obousměrný přenos dat. Pár vodičů pro USB 2.0 byl zachován, a tím i zpětná kompatibilita. Zařízení nepodporující Superspeed mají nadále splňovat specifikaci USB 2.0.

2.1.5 USB On-the-Go

Toto rozšíření specifikace USB 2.0 dovoluje implementovat do zařízení omezenou funkčnost hosta, a tím umožňuje vzájemnou komunikaci dvou zařízení. V praxi může například fotoaparát ukládat fotografie na připojený flash disk.

2.1.6 Wireless USB

Bezdrátová varianta USB byla uvedena již v roce 2005, stále se však nedočkala masového rozšíření. Používá nosný kmitočet v rozsahu 3,1 až 10,6 GHz a na vzdálenost do 3 metrů nabízí přenosovou rychlost 480 Mb/s, tedy stejnou jako High speed kabelové spojení. Na vzdálenost 10 m je pak garantována přenosová rychlost 110 Mb/s.

2.2 Napájení přes USB

Součástí USB je také rozvod napájecího napětí +5 V. Ihned po připojení má každé zařízení k dispozici až 100 mA.

Nutno podotknout, že napájecí specifikace nejsou výrobci příliš dodržovány, na trhu jsou běžně dostupné například různé vyhřívané podložky, které ihned po připojení do USB portu odebírají proud výrazně překračující dovolené maximum 500 mA. To je také případ mnoha externích 2,5“ disků. Naproti tomu někteří výrobci počítačových komponent dimenzují USB porty na základních deskách až na odběr 1,5 A.

Zařízení se z hlediska napájení dělí do dvou skupin: [1]

2.2.1 Zařízení napájené z USB (bus-powered)

Do této skupiny se řadí zařízení, která nemají vlastní napájecí zdroj, a berou energii z USB. Spotřeba zařízení hned po připojení nemá přesahovat 100 mA. Zařízení si může vyžádat až 500 mA prostřednictvím hodnoty bMaxPower v konfiguračním deskriptoru (viz dále). Tento proud však smí odebírat až po úspěšné konfiguraci. Spící zařízení (suspend mode) může odebírat max. 2,5 mA. Dovolená kapacita připojená na napájení z USB je 1 - 10 μ F. Napájení na straně zařízení (na konci kabelu) může kolísat v rozmezí 4,4 V až 5,25 V. SuperSpeed zařízení na sběrnici USB 3.0 může odebírat až 900 mA.

2.2.2 Zařízení s vlastním napájením (self-powered)

Do této skupiny patří zařízení, která musí být funkční i bez připojení k USB, potřebují větší napájecí proud než 500 mA nebo napětí větší než 5 V. Zařízení může ze sběrnice odebírat až 100 mA, tento proud je určen k napájení řídicího obvodu a pull-up odporu na datové lince. Ten musí být napájen ze sběrnice, jinak by host detekoval zařízení, ale nemohl s ním komunikovat. Zařízení nesmí nikdy napájet proudem sběrnici, ani detekční pull-up odpor nesmí být připojen dříve než je detekováno napětí ze sběrnice (VBUS).

2.3 Typy přenosů na sběrnici

Specifikace USB definuje 4 typy přenosů. Každý typ má specifické vlastnosti a je proto vhodný k jinému účelu [1],[5].

2.3.1 Řídící přenos (Control transfer)

Řídící přenosy slouží především k přenášení deskriptorů a konfiguračních informací. Mohou také přenášet libovolné informace specifikované určitou třídou nebo výrobcem. Jsou zabezpečené proti chybám a mají přesně definovanou strukturu – používají rouru zpráv (message pipe).

Tento typ přenosu musí podporovat každé zařízení. Maximální velikost datového paketu je pro Low speed zařízení 8 B, pro Full speed a High speed zařízení až 64 B, a pro SuperSpeed až 512 B.

2.3.2 Hromadný přenos (Bulk transfer)

Tento typ přenosu je vhodný pro přenos velkých objemů dat, u kterých však nezáleží na přesném časování. Má nižší prioritu než ostatní typy přenosu, využívá se volná přenosová kapacita sběrnice. Typické použití je posílání dat tiskárně nebo čtení a zápis na flash disk.

Na nevytížené sběrnici je tento typ přenosu nejrychlejší. Hromadné přenosy nejsou dostupné pro Low speed zařízení. Podpora tohoto typu přenosu není povinná, ale může být vyžadována určitou třídou, například Mass-storage zařízení musí mít alespoň jeden Bulk endpoint v každém směru.

Maximální velikost datového paketu je pro Full speed zařízení až 64 B, pro High speed zařízení 512 B a pro SuperSpeed 1024 B. Hromadné přenosy jsou rovněž zabezpečené proti chybám.

2.3.3 Přerušovací přenos (Interrupt transfer)

Přerušovací přenosy jsou užitečné pro data, která musí být přenesena bez zpoždění. Typicky se používají pro HID zařízení jako klávesnice a myš, ale také pro ohlášení stavu USB hubu. Low speed zařízení podporují pouze řídící a přerušovací přenosy. I přerušovací endpointy musí vyčkat až si host vyžádá data. Přerušovací přenosy jsou dostupné pro všechny rychlosti ale nejsou povinné. Pouze např. zařízení třídy HID musí mít jeden přerušovací IN endpoint pro reportování dat hostu.

Maximální velikost datového paketu je pro Low speed zařízení 8 B, pro Full speed zařízení 64 B, pro High speed a SuperSpeed zařízení až 1024 B. Přerušovací přenosy jsou zabezpečené proti chybám. Zařízení si určí maximální latenci pomocí hodnoty bInterval v deskriptoru endpointu.

2.3.4 Izochronní přenos (Isochronous transfer)

Tento typ přenosu je proudový, probíhá v reálném čase. Je vhodný, když data musí mít definovanou latenci a určitou přenosovou kapacitu i na obsazené sběrnici, ale jsou přípustné občasné chyby v přenosu. Typicky se používá pro přenos hudebního signálu nebo videa v reálném čase.

Izochronní přenosy nejsou podporovány Low speed zařízeními. Maximální velikost datového paketu je pro Full speed zařízení až 1024 B, pro High speed a SuperSpeed zařízení až 1024 B. Izochronní přenosy nejsou zabezpečené proti chybám.

2.4 Třídy USB zařízení

Většina USB zařízení má mnoho společného s jinými zařízeními plnícími podobnou funkci. Proto byly zavedeny různé třídy, které tato zařízení sdružují. Zařazení daného zařízení do třídy usnadňuje vývoj a umožňuje použití standardních ovladačů přeinstalovaných v operačním systému. Specifikace definují celou řadu různých tříd z mnoha oblastí, například Audio (přenos zvuku), Printer (tiskárny), Mass-storage (velkokapacitní paměťová zařízení) nebo i Personal healthcare (péče o zdraví). Některé třídy však nejsou podporovány v operačních systémech. V následujícím textu jsou stručně popsány pouze třídy použitelné pro připojení rotačního enkodéru k PC [1].

2.4.1 Communication class – CDC (komunikační třída)

Zahrnuje např. ISDN modemy, nebo převodníky USB/Ethernet. Nejčastěji se však používá virtuální COM port, který umožňuje snadnou náhradu zastaralého rozhraní RS-232 bez nutnosti změn v aplikaci v PC. Komunikace na straně PC aplikace i firmwaru zařízení je snadná. Nevýhodou je zejména nutnost vytvořit konfigurační .inf soubor, který je nutné načíst při instalaci.

2.4.2 Custom class (vlastní třída)

USB dovoluje definovat i vlastní třídu. Přenášená data pak mohou mít jakýkoliv formát, na rozdíl od HID nemusí mít stanovenou strukturu a délku reportu. Nevýhodou je opět nutnost vytvořit .inf soubor a potřeba jeho načtení při instalaci zařízení. Aplikace v PC může využívat standardní ovladač WinUSB.

2.4.3 Human interface – HID (zařízení pro styk s uživatelem)

Zahrnuje klávesnice, myši, nebo herní zařízení. HID třída však může přenášet i výrobcem definovaná data. Komunikace probíhá v tzv. reportech, kterým se musí přizpůsobit formát dat. Musí být definovány v report deskriptorech. Deskriptory jsou podrobněji popsány v kapitole 4.2.

Třída HID dovoluje použití pouze řídicích nebo přerušovacích přenosů. Každé HID zařízení musí mít jeden IN endpoint s přerušovacími přenosy. Komunikace ve směru od hosta k zařízení může probíhat pomocí endpointu 0 a řídicích přenosů. Výhodnější je však použít přerušovací OUT endpoint, který má garantovanou latenci. Při rychlosti komunikace Full speed je dosažitelná přenosová rychlost dat 64 kB/s v každém směru.

Tato třída má dobrou podporu i ve starších operačních systémech. Instalace HID zařízení probíhá na běžných operačních systémech zcela automaticky, bez zásahu uživatele. Zařízení je obvykle během několika sekund po připojení připraveno k použití.

Aplikace ve Windows mohou k HID zařízení přistupovat pomocí standardních funkcí ReadFile a WriteFile, nebo pomocí specifických API, jako je HidD_GetFeature. Další možností je přístup pomocí DirectInput nebo Raw Input. Tento se používá především u herních zařízení. Návrhu aplikace je věnována kapitola 5.

3 REALIZACE

3.1 Výběr mikroprocesoru

Řídicí obvod byl vybírán z nabídky firem Microchip a Atmel. Microchip nabízí širokou paletu obvodů s podporou USB, počínaje 8-bitovou řadou PIC18F, následuje 16-bitová řada PIC24F a nejvýkonnější 32-bitová řada PIC32F disponující i funkcí USB Host. Jak již bylo zmíněno v úvodu, pro tento ovladač vyhoví svým výkonem 8-bitový mikrokontrolér, proto bylo vybráno několik typů z rodiny PIC18F.

Atmel nabízí USB mikrokontroléry se dvěma odlišnými architekturami, AVR a 8051. Rodina AVR má vyšší výkon, a je celkově perspektivnější.

Pro AVR existuje i softwarová implementace USB firmy Objektive Developments. Výhodou pak možnost použít téměř jakýkoliv běžně dostupný procesor řady ATtiny nebo ATmega. Omezením je však podpora starší specifikace USB 1.1 s přenosovou rychlostí Low speed, a také chybějící ošetření chyb v komunikaci [6].

Byly vybírány obvody s velikostí programové paměti od 8 kB, alespoň 14 vstupně/výstupními piny, podporou USB 2.0 Full speed a cenou do 6 €. Rozhodující byla také dostupnost.

Základní vlastnosti vybraných procesorů jsou shrnuty v Tabulka 2.

Tabulka 2: Vlastnosti vybraných mikroprocesorů

Typ	Flash (kB)	I/O	SRAM (B)	ADC	Timer	Pouzdro	Cena (€)
Atmel AT90USB162	16	22	512	0 (1 komp.)	2	TQFP32	5,21
Atmel ATmega32U4	32	26	2560	12	5	TQFP32	3,70
Microchip PIC18F14K50	8	14	768	9	4	DIP20 SOIC20	2,36
Microchip PIC18F26J50	64	22	4096	10	5	DIP28 SOIC28	3,42
Microchip PIC18F2550	32	24	2048	13	4	DIP28 SOIC28	4,95

Vysvětlivky:

Flash: velikost programové paměti v kilobytech

I/O: počet vstupně/výstupních pinů

SRAM: velikost operační paměti v bytech

ADC: počet kanálů A/D převodníku, uvedeno pouze pro informaci, v této aplikaci není použit a proto nemá rozhodující vliv. Komp. = komparátor.

Timer: počet časovačů, rovněž není rozhodující

Pouzdro: výběr pouzder

Cena: Maloobchodní cena v € za 1 kus. Zdroj: Farnell, květen 2010

3.1.1 Atmel AT90USB162

Tento obvod je v nabídce již delší dobu, a proto je dobře dostupný. Podporuje USB 2.0 Full speed s maximálním počtem 4 endpointů. Zajímavostí tohoto obvodu je zabudované rozhraní PS/2 na společných pinech s USB, což je ideální pro návrh klávesnice nebo myši kompatibilní s oběma rozhraními. Nevýhodou tohoto obvodu je velikost operační paměti pouze 512 B, a vzhledem k poměrně chudé výbavě i vyšší cena. Vyrábí se pouze v pouzdrech pro povrchovou montáž [7].

3.1.2 Atmel ATmega32U4

Procesor ATmega32U4 je naopak novinkou. Oproti předchozímu obvodu přináší mnoho periférií navíc, například 12-kanálový A/D převodník, nebo vestavěné teplotní čidlo. Výhoda tohoto obvodu je z výroby předprogramovaný USB bootloader, odpadá tedy nutnost použití ISP programátoru. Firmware se do procesoru nahrává programem Flip. Vyrábí se pouze v pouzdrech pro povrchovou montáž. Tento procesor je bohužel naprosto nedostupný [8].

3.1.3 Microchip PIC18F14K50

Tento obvod vyniká nízkou cenou. Je dostupný v pouzdru SOIC20 nebo DIP20. Počtem vývodů je však na hranici použitelnosti pro danou aplikaci. Také paměť programu a RAM mají poměrně malou kapacitu [9].

3.1.4 Microchip PIC18F26J50

Nejnovější generace obvodů obsahuje interní oscilátor s přesností 0,15 %, který postačuje i pro použití s Low speed USB. Nevýhodou je maximální dovolené napájecí napětí 3,6 V, proto pro napájení z USB je nutné použít externí stabilizátor. Jádro procesoru pracuje při napětí 2,5 V z vestavěného LDO stabilizátoru. Pouze 6 I/O pinů je 5 V tolerantních. Vyrábí se v pouzdru DIL28 a SOIC28 [10]. Obvod umožňuje libovolné namapování mnoha periférií na zvolené I/O piny, což velmi usnadňuje návrh plošného spoje. Proto byl vybrán pro použití v ovladači. Dalšími důvody byly široký výběr pouzder, příznivá cena a poměrně dobrá dostupnost.

3.1.5 Microchip PIC18F2550

Tento procesor byl uveden jako první s vestavěným USB rozhraním z rodiny 18F. Vyrábí se v pouzdru DIL28 nebo SOIC28 a je vybaven mnoha funkcemi [11]. Díky dobré dostupnosti a snadné použitelnosti pouzdra DIL40 byla varianta procesoru PIC18F4550 použita pro počáteční vývoj aplikace v nepájivém kontaktním poli. Dalším důvodem pro použití tohoto typu byla i existence několika článků v češtině, které usnadnily počáteční kroky při vývoji [12],[13].

3.2 Popis vybraného mikroprocesoru

Procesor PIC18F26J50 je vybaven 64 kB programové paměti Flash s životností až 100 000 přepsání, podporou samo-programování a možností programování v obvodu ICSP (In-Circuit

Serial Programming). Operační paměť má kapacitu 4 kB. Procesor má dovolené napájecí napětí max. 3,6 V, jádro procesoru je napájeno napětím 2,5 V ze zabudovaného LDO stabilizátoru 34[10][11].

USB modul tohoto obvodu odpovídá specifikaci USB 2.0 a podporuje přenosové rychlosti Low speed a Full speed. Umožňuje použití až 32 endpointů (16 obousměrných) se všemi čtyřmi typy přenosů.

Architektura procesoru je optimalizována pro psaní programu v jazyce C. Obsahuje hardwarovou násobičku 8 x 8.

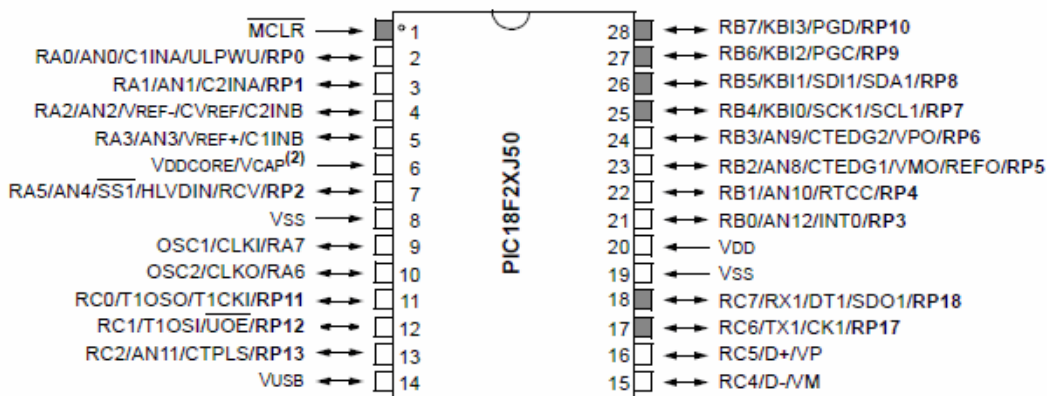
Zdrojem taktovacího kmitočtu může být u aplikací nenáročných na přesné časování interní RC oscilátor na kmitočtu 31 kHz až 8 MHz. Interní oscilátor s kmitočtem 6 MHz je dostatečně přesný pro Low speed USB. Pro přenosovou rychlost Full speed vyžaduje kmitočet 48 MHz. Díky vestavěným násobičkám a děličkám kmitočtu lze použít krystal nebo externí oscilátor s kmitočtem v rozsahu 4 MHz až 48 MHz. USB modul může pracovat na odlišném kmitočtu než jádro procesoru.

Procesor integruje celou řadu periférií: desetibitový A/D převodník s deseti vstupy, dva analogové komparátory, celkem 3 časovače, dva Capture/Compare/PWM moduly, rozšířené sériové rozhraní Enhanced USART, sériový port s podporou SPI a I2C a další. Zajímavou periférií je jednotka CTMU (Charge time measurement unit), která je určena pro implementaci dotykových kapacitních klávesnic.

Zapojení vývodů použitého pouzdra SOIC28 je na Obr. 2. Většina vývodů má několik různých možných funkcí. Jednotlivé funkce jsou popsány v Tabulka 3.

28-Pin SPDIP/SOIC/SSOP

■ = 5,5 V tolerantní piny



Obr. 2: Zapojení vývodů mikroprocesoru (převzato z [10])

Tabulka 3: Popis vývodů mikroprocesoru

Název	Funkce
VDD	Napájecí napětí v rozsahu 2 – 3,6 V
VCAP	Výstup LDO stabilizátoru 2,5 V pro jádro, nutné připojit kondenzátor
VSS	Zem (GND)
/MCLR	Reset procesoru, aktivní v log. 0, nutný pull-up odpor k VDD
PGD, PGM, PGC, VPP	Vývody určené k připojení programátoru (ICSP) nebo debuggeru (ICD)
OSC1, OSC2	Vývody pro připojení krystalu nebo externího oscilátoru

RA0 – RA6	Vstupně-výstupní port A, vyvedeno 5 bitů
RB0 – RB7	Vstupně-výstupní port B, vyvedeno 8 bitů, možnost externího přerušení při změně logické úrovně
RC0 – RC7	Vstupně-výstupní port C, vyvedeno 7 bitů, pin RC3 není implementován
AN0 – AN12	Vstupy A/D převodníku
INT0	Vstup externího přerušení, další 3 jsou mapovatelné
RX, TX	Vývody modulu EUSART, např. pro RS-232
SDO, SCI, SCK	SPI sběrnice
SCL, SDA	I2C sběrnice
D+, D-	Vývody pro přímé připojení k USB sběrnici
VUSB	Napájení USB modulu, nutné připojit napětí 3,3 V
RP0 – RP18	Remapovatelné piny

3.3 Rotační enkodéry

Rotační enkodéry nacházejí v elektronice stále širší uplatnění. Typickým použitím je regulace hlasitosti u audio zařízení, kde již téměř vytlačily dříve používané potenciometry. Ve spotřební elektronice se obvykle využívají mechanické enkodéry. Vyrábí se rovněž typy s axiálním tlačítkem, které jsou pak vhodné i pro pohyb v menu ovládaného přístroje. Mechanický enkodér se vzhledově téměř neliší od potenciometru. Na rozdíl od potenciometru nemá zarážky, je tedy možné s ním otáčet neomezeně. Má tři vývody, označované A,B,C. Vývod C (common – společný) se připojuje na zem. Vývody A a B se doplní pull-up rezistory a připojí na datové vstupy mikroprocesoru. Na těchto dvou vývodech jsou pak obdélníkové průběhy vzájemně fázově posunuté asi o 90°. Nejjednodušší metodou načítání stavu je připojit jeden vývod na vstup externího přerušení procesoru, nastavit přerušení např. při sestupné hraně, a při vyvolání tohoto přerušení načítat logickou úroveň druhého vývodu [14]. Tento postup je naznačen na Obr. 3.



Obr. 3: Princip rotačního enkodéru (převzato z [14])

Mechanické enkodéry obvykle dodávají několik desítek impulzů na každou otáčku. Jejich výhodou je především nízká cena, nevýhodami naopak omezená životnost a také generování záskritů.

Při vyšších nárocích na přesnost nebo rychlost snímání (např. měření otáček motoru) lze použít optické nebo magnetické enkodéry. Oba typy mají především výrazně delší životnost, která není negativně ovlivněna opotřebením kontaktů jako u mechanických enkodérů. Optické enkodéry pracují na principu odrazu světla, nebo změny propustnosti světla, magnetické enkodéry využívají Hallova jevu. Optické a magnetické enkodéry se vyrábějí i jako absolutní, které nacházejí uplatnění především v automatizaci a robotice, pro přesné snímání polohy motorů.

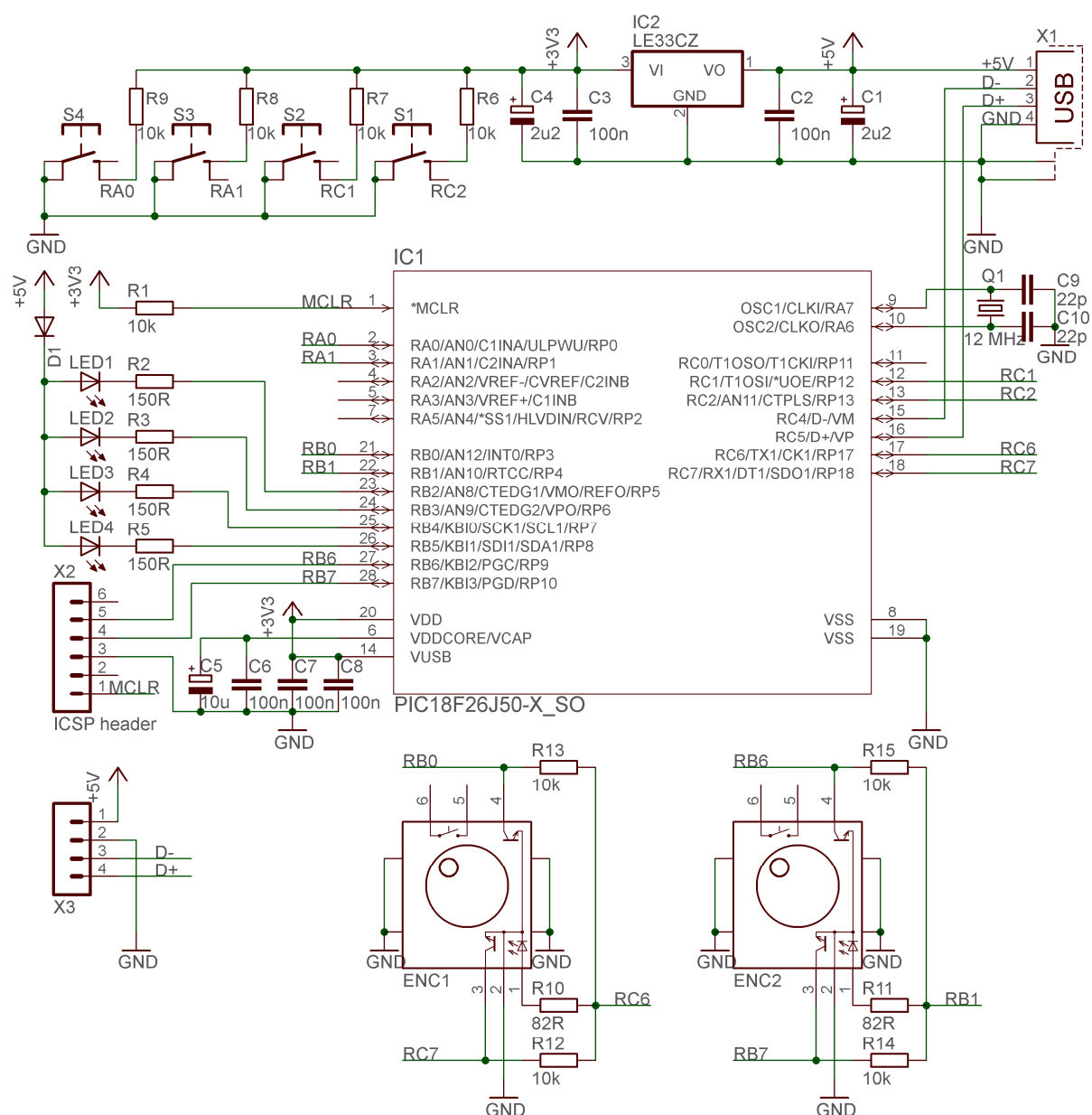
V ovladači jsou použity optické enkodéry Grayhill 62P22-L0 [15] s životností až milion

otočení. Generují 16 impulzů na otáčku. Nevýhodou tohoto typu je poměrně výrazná vůle hřídelky, a také její nestandardní průměr 4,75 mm. Problémem může být dlouhá doba přechodu mezi logickými úrovněmi (až 30 ms), řešením je připojit enkodér k pinům procesoru vybavenými Schmittovými klopnými obvody.

U optických enkodérů se při otočení o jeden krok změni logická úroveň pouze na jednom pinu, proto je nutné sledovat oba výstupní piny enkodéru a používat přerušení na sestupnou i náběžnou hranu.

3.4 Popis zapojení

Zapojení obsahuje díky použitému procesoru pouze minimum dalších součástek. Schéma zapojení je na Obr. 4.



Obr. 4: Schéma zapojení ovladače

Ovladač je napájen ze sběrnice USB, jedná se o tzv. bus-powered zařízení. Napětí 3,3V pro procesor zajišťuje nízkoubytkový stabilizátor LE33 doplněný filtračními a blokovacími kondenzátory. Kapacity C1 a C4 jsou voleny 2,2 μF , aby jejich součet nepřekročil maximální hodnotu dovolenou specifikací USB tj. 10 μF . Kondenzátor C5 filtruje napětí 2,5 V pro jádro procesoru, jeho kapacitu 10 μF předepisuje Microchip. Jako zdroj taktovacího kmitočtu slouží krystal Q1 s kondenzátory C9 a C10. Kmitočet 12 MHz je v procesoru pomocí PLL zvýšen na 48 MHz.

LED diody musí být připojeny na Port B nebo C, protože piny na Portu A mají dovolené zatížení jen 4 mA. Jsou připojeny anodami přímo k napájení 5 V z USB, katodami na I/O piny procesoru. Toto řešení nezatěžuje stabilizátor napětí. K rozsvícení LED diody dojde po zapsání log. 0 na příslušný pin. Pro zamezení prosvítání při log. 1 na výstupu je před LED diody zařazena ještě dioda D1. Proud LED diod je pro dostatečnou svítivost zvolen asi 15 mA. Tlačítka jsou připojena k různým pinům procesoru tak, aby se usnadnil návrh plošného spoje. Jsou zapojena proti zemi a doplněna pull-up rezistory s hodnotou 10 k Ω .

Optické enkodéry vyžadují napájení integrovaných LED diod, ty jsou připojeny také na I/O piny procesoru, aby se daly také ovládat programově a v případě potřeby snížit spotřebu ovladače. Hodnoty rezistorů okolo enkodérů jsou předepsány výrobcem [15]. Výstupní piny enkodérů jsou připojeny rovněž s ohledem na návrh plošného spoje, v procesoru jsou pak na tyto piny softwarově namapovány vstupy externích přerušení. Tím jsou také na příslušných vstupech aktivovány Schmittovy klopné obvody, které jsou u optických enkodérů nezbytné pro správnou funkci.

USB modul nevyžaduje žádné externí součástky, datové linky jsou z konektoru připojeny přímo na piny D+ a D- procesoru. Přestože procesor obsahuje omezenou ochranu proti elektrostatickým výbojům, byla později na datové linky doplněna integrovaná ESD ochrana PRTR5V0U4D [16].

3.5 Návrh plošného spoje a mechanické koncepce

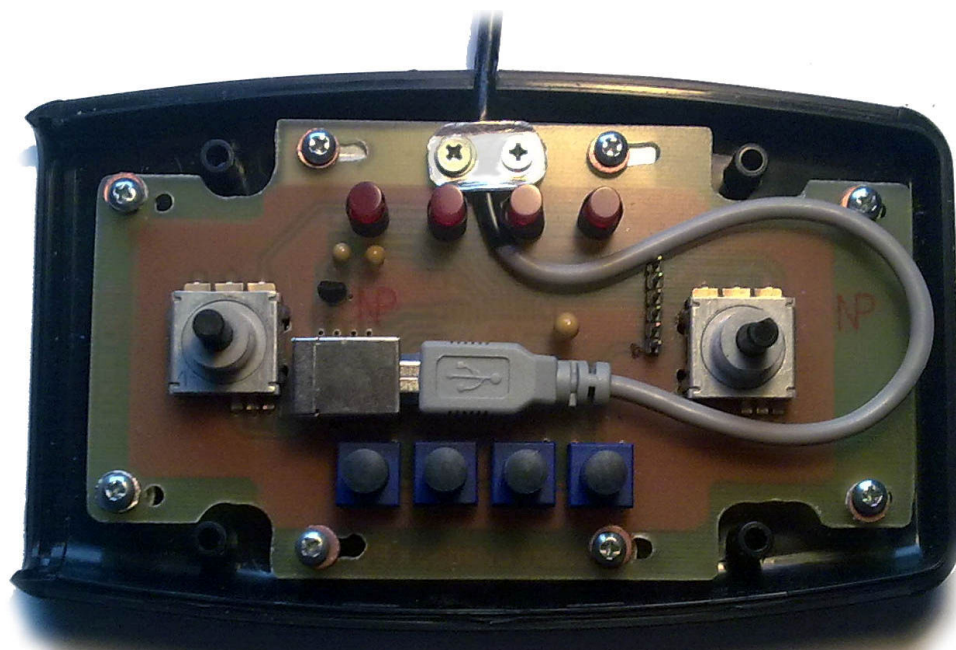
Pro ovladač byla vybrána kvalitní plastová krabička Hammond 1553DBKBBK [17]. Je přímo určena pro držení v ruce, na bočních stranách má příjemný pogumovaný povrch. Rozměry této krabičky jsou 147 x 87 x 24 mm. Rotační enkodéry Grayhill jsou doplněny hliníkovými knoflíky Mentor 513.61 o průměru 30 mm [18]. Jsou použita tlačítka Multimec 3E [19]. Celkové uspořádání je zřejmé z Obr. 5.



Obr. 5: Pohled na hotový ovladač, rozmístění ovládacích prvků

USB kabel je u hotového ovladače neodělitelný. Uvnitř ovladače je však skryt standardní USB konektor typu B, což umožňuje použití běžně dostupného připojovacího kabelu a jeho snadnou výměnu, viz Obr. 6. Kabel je pomocí hliníkového pásku zabezpečen proti vytržení.

Návrh plošného spoje byl přizpůsoben umístění v krabičce. Většina součástek je v provedení SMD, což umožnilo vytvořit na horní straně prostor pro USB konektor s kabelem. Obě datové linky USB sběrnice mají na plošném spoji téměř stejnou délku, aby i zpoždění signálu bylo stejné. Použitý procesor obsahuje jednotku PPS (Peripheral Pin Select), která umožňuje mnoho periférií procesoru, včetně vstupů interních přerušení, napájet na téměř libovolný pin procesoru. Toho bylo při návrhu plošného spoje maximálně využito, kreslení plošného spoje probíhalo současně s návrhem schématu tak, aby se minimalizovalo křížení spojů. Podařilo se navrhnout jednostranný plošný spoj bez použití jediné drátové propojky. Výsledný obrazec o rozměrech 136 x 72 mm byl přizpůsoben pro ruční vrtání a ostřížení. Je obsažen v příloze A.1.



Obr. 6: Vnitřní uspořádání ovladače

4 FIRMWARE PROCESORU

Úkolem firmwaru procesoru je zajišťovat komunikaci přes sběrnici USB, a obsluhovat rotační enkodéry, tlačítka a LED diody připojené na piny procesoru.

Firmware byl vyvíjen v prostředí MPLAB IDE ve verzi 8.30. Aktuální verze prostředí je dostupná na stránkách firmy Microchip [20]. Pro vývoj USB zařízení je nutné dále nainstalovat Microchip USB Framework, jenž je součástí balíku Microchip Application Libraries [21]. Byla použita aktuální verze 2.6. Po nainstalování je vytvořena složka C:\Microchip Solutions. Součástí instalace jsou i vzorové projekty. USB Framework je sada několika zdrojových souborů v jazyce C. Soubory budou popsány dále, zejména ty, které vyžadují modifikaci. Firmware rotačního ovladače vychází ze vzorového projektu USB Device - HID - Simple Custom Demo. Je psán v jazyce C. Vytvořený firmware je datový soubor s příponou .hex. Zabírá 6 489 B, tedy asi 10 % z 64 kB programové paměti procesoru.

4.1 Zápis programu do mikroprocesoru

Pro nahrání .hex souboru do programové paměti procesoru přes ICSP rozhraní je nutné použít programátor, například PICkit2 [22]. K připojení programátoru slouží konektor X2. Napájení VDD není na konektor vyvedeno, ovladač je nutné vždy napájet z USB. Výhodou ICSP programování je přímé propojení cílového procesoru s vývojovým prostředím a možnost krokování programu. Tento způsob programování je potřebné provést pouze jedenkrát.

Do paměti procesoru byl nahrán USB HID bootloader. Zdrojový kód byl poskytnut firmou Microchip, stačilo upravit startovací podmínku. Aktivuje se resetováním procesoru při současném držení tlačítka S1, tedy log. 0 na pinu RC2 procesoru. Režim bootladeru je indikován střídavým blikáním LED1 a LED2. Program se následně nahrává přímo přes USB pomocí aplikace HIDBootLoader, která je také součástí instalace a je také přiložena na CD. Bootloader zabírá první 4 kB programové paměti procesoru.

4.2 Deskriptory

Deskriptory jsou datové struktury, které jsou uloženy v zařízení a po připojení jsou vyžádány hostem. Přenášejí informace o schopnostech a funkcích připojeného zařízení. Je definováno celkem 8 typů deskriptorů [1][11]. Každý deskriptor na svém počátku obsahuje údaj o velikosti, aby bylo jasné kolik dat bude následně přeneseno, následovaný konstantou definující typ deskriptoru. Konstanty jsou předdefinovány v hlavičkovém souboru usb_device.h, a proto jsou vyjádřeny slovně. Deskriptory se přenášejí řídicím přenosem přes výchozí endpoint 0.

Deskriptory jsou obsaženy v souboru usb_descriptors.c. Každá položka v deskriptoru je doplněna komentářem.

4.2.1 Deskriptor zařízení (device descriptor)

Poskytuje nejzákladnější informace o zařízení. Každé zařízení má pouze jeden deskriptor tohoto typu. Po standardních údajích o velikosti a typu deskriptoru je uvedena verze USB specifikace, současné verze 2.0. Následují kódy třídy, podtřídy a protokolu. Ty jsou důležité

především pro klávesnici a myš , které se používají při bootování počítače. U vlastního zařízení jsou zde ponechány nulové, a jsou definovány v deskriptoru rozhraní. Maximální velikost paketu endpointu 0 musí být zvolena 8, 16, 32 nebo 64 bytů. Pro dané zařízení postačuje nejmenší možná velikost. Další důležité údaje jsou identifikační čísla výrobce (Vendor ID) a zařízení (Product ID). ID výrobce bylo zachováno ze vzorového projektu, číslo zařízení bylo zvoleno symbolicky stejné jako zkratka naší fakulty. Tento postup je možný pouze pro demonstraci. Pro reálnou výrobu zařízení je nutné získat od USB Implementers Forum vlastní ID výrobce. Deskriptor zařízení dále obsahuje výrobní číslo, indexy string deskriptorů a počet možných konfigurací.

```
/* Device Descriptor */
ROM USB_DEVICE_DESCRIPTOR device_dsc=
{
    0x12,                // Velikost tohoto deskriptoru v Bytech
    USB_DESCRIPTOR_DEVICE, // Typ deskriptoru - DEVICE
    // všechny tyto konstanty jsou definovány v usb_device.h
    0x0200,              // Verze USB specifikace v BCD kódu
    0x00,                // Kód třídy (class)
    0x00,                // Kód podtřídy (subclass)
    0x00,                // Kód protokolu
    USB_EP0_BUFF_SIZE,  // Max. velikost paketu Endpointu 0,
                        // definováno v usb_config.h
    0x04D8,              // Vendor ID (Microchip)
    0xFEED,              // Product ID zvoleno...
    0x0001,              // Výrobní číslo
    0x01,                // Index string deskriptoru výrobce
    0x02,                // Index string deskriptoru produktu
    0x00,                // Index string deskriptoru sériového čísla
    0x01                 // Počet možných konfigurací
};
```

4.2.2 Deskriptor konfigurace (configuration descriptor)

Pro každou konfiguraci musí zařízení poskytnout odpovídající deskriptor. U tohoto ovladače postačuje pouze jedna konfigurace. Deskriptor konfigurace poskytuje informace o spotřebě a o typu napájení – v tomto případě napájení ze sběrnice (bus-powered). Dále může obsahovat informace o funkci vzdáleného probuzení (remote wake-up), která není u tohoto ovladače použita. Nastavení napájení je nutné provést také v souboru HardwareProfile – PIC18F26J50.h.

```
/* Configuration 1 Descriptor */
ROM BYTE configDescriptor1[]={
    /* Configuration Descriptor */
    0x09,                // Velikost tohoto deskriptoru v B
    USB_DESCRIPTOR_CONFIGURATION, // Typ deskriptoru - CONFIGURATION
    0x29, 0x00,          // Celková délka dat pro tuto konfiguraci
    1,                   // Počet rozhraní v této konfiguraci
    1,                   // Index této konfigurace
    0,                   // Index stringu popisujícího konfiguraci
    _DEFAULT,           // Atributy, více v usb_device.h
    50,                  // Max. spotřeba (x2 mA) = 100 mA
};
```

4.2.3 Deskriptor rozhraní (interface descriptor)

Zařízení musí poskytnout deskriptor pro každé rozhraní. Pro tento ovladač postačuje jedno rozhraní. Zde jsou definovány kódy třídy, podtřídy a protokolu, pokud již nebyly definovány v deskriptoru zařízení. Tento deskriptor dále udává počet endpointů pro dané rozhraní. Je

uložen ve společném poli s deskriptorem konfigurace.

```
/* Interface Descriptor */
0x09, // Velikost tohoto deskriptoru v Bytech
USB_DESCRIPTOR_INTERFACE, // Typ deskriptoru - INTERFACE
0, // Číslo rozhraní
0, // Počet alternativních nastavení
2, // Počet endpointů v tomto rozhraní - dva
HID_INTF, // Kód třídy - HID
0, // Kód podtřídy
0, // Kód protokolu
0, // Index stringu popisujícího rozhraní
```

4.2.4 HID deskriptor třídy (HID class-specific descriptor)

Tento deskriptor je specifický pro třídu HID. Obsahuje informace o verzi HID specifikace [23], počtu HID deskriptorů a velikosti report deskriptoru. Tyto dvě konstanty jsou definovány v souboru `usb_config.h`, a jejich hodnota musí odpovídat skutečnosti. Deskriptor je uložen ve společném poli s deskriptorem konfigurace.

```
/* HID Class-Specific Descriptor */
0x09, // Velikost tohoto deskriptoru v Bytech
DSC_HID, // Typ deskriptoru - HID
0x11, 0x01, // Verze HID specifikace 1.11
0x00, // Kód země (nepodporováno)
HID_NUM_OF_DSC, // Počet deskriptorů třídy
DSC_RPT, // Typ deskriptoru - Report
HID_RPT01_SIZE, 0x00, // Velikost report deskriptoru
```

4.2.5 Deskriptory endpointů (endpoint descriptors)

Tento ovladač používá mimo výchozí řídicí endpoint 0 ještě jeden přerušovací endpoint v každém směru. Přerušovací endpoint ve směru od zařízení k hostu (IN) je pro každé HID zařízení povinný. Pro přenos ve směru od hosta do zařízení by bylo možné použít výchozí řídicí endpoint 0, u kterého však není garantována latence. Proto byl použit také přerušovací endpoint. Firmware ovladače tedy obsahuje dva deskriptory endpointů. Každý obsahuje údaj o čísle endpointu a směru toku dat, a také informace o velikosti a obnovovacím intervalu. Velikost endpointů byla přizpůsobena skutečným potřebám, tedy 5 Bytů IN a 2 Byty OUT. Po změně velikosti endpointů je nutné změnit odpovídající konstanty v souboru `usb_config.h`, dále velikosti bufferů ve firmwaru i PC aplikaci, i parametry při volání funkcí pro čtení a zápis.

Obnovovací interval původního dema byl 1 ms pro dosažení teoretické max. přenosové rychlosti 64 kB/s. Pro tento ovladač byl interval prodloužen na 10 ms, což je stále dostatečná hodnota, která méně vytěžuje sběrnici a dává více času na zpracování dat aplikaci v počítači. Deskriptory jsou uloženy ve společném poli s deskriptorem konfigurace.

```
/* Endpoint Descriptor */
0x07, // Velikost tohoto deskriptoru v Bytech
USB_DESCRIPTOR_ENDPOINT, // Typ deskriptoru - Endpoint
HID_EP | _EP_IN, // Číslo a typ endpointu - EP1 IN
_INTERRUPT, // Typ přenosu
0x05, 0x00, // Velikost endpointu 5 Bytů
0x0A, // Interval 10 ms

/* Endpoint Descriptor */
0x07, // Velikost tohoto deskriptoru v Bytech
USB_DESCRIPTOR_ENDPOINT, // Typ deskriptoru - Endpoint
```

```

HID_EP | _EP_OUT,      // Číslo a typ endpointu - EP1 OUT
_INTERRUPT,           // Typ přenosu
0x02,0x00,            // Velikost endpointu 2 Byty
0x0A                   // Interval 10 ms

```

4.2.6 Textové deskriptory (string descriptors)

Obsahují popisující textové řetězce. Nejprve musí být uveden kód jazyka, nedoporučuje se používat jiný jazyk než angličtinu (0x0409). Další dva deskriptory obsahují název výrobce a název zařízení. V definici každého deskriptoru musí být uvedena skutečná délka textu.

```

// Deskriptor jazyka
ROM struct{BYTE bLength;BYTE bDscType;WORD string[1];}sd000={
sizeof(sd000),USB_DESCRIPTOR_STRING,{0x0409
}};

// Deskriptor - název výrobce
ROM struct{BYTE bLength;BYTE bDscType;WORD string[32];}sd001={
sizeof(sd001),USB_DESCRIPTOR_STRING,
{'L','u','k','a','s',' ','O','l','i','v','i','k',' ',' ',
' ','U','R','E','L',
' ','F','E','K','T',' ','V','U','T',' ','B','r','n','o'
}};

// Deskriptor - název zařízení
ROM struct{BYTE bLength;BYTE bDscType;WORD string[20];}sd002={
sizeof(sd002),USB_DESCRIPTOR_STRING,
{'R','o','t','a','c','n','i',' ','o','v','l','a','d','a','c',' ','k',' ',
'P','C'
}};

```

4.2.7 HID report deskriptor

Informuje hosta o formátu posílaných dat. Struktura je odlišná od jiných typů deskriptorů. K vytvoření, editaci, a kontrole tohoto deskriptoru se používá HID Descriptor Tool [24]. Sdílet hostu formát dat je důležité především u standardních systémových zařízení, nebo herních zařízení, aby fungovala přímo pod operačním systémem nebo podporovala získání dat nějakou standardní metodou. U vlastního zařízení si formát dat stanoví tvůrce, a následně ho respektuje při tvorbě aplikace pro počítač. Proto by bylo možné použít obecný report deskriptor ze vzorového projektu, případně jen změnit velikost přenášovaných dat.

Přesto byl pro demonstraci vytvořen obsáhlejší deskriptor. Vyjadřuje, že zařízení může posílat hostu 0 až 5 Bytů. Z nich první tři Byty jsou absolutní neznaménkové, které budou použity pro posílání stavu tlačítek a dvou absolutních čísel vyjadřujících pozice enkodérů. Následují dva relativní znaménkové Byty, které budou vyjadřovat relativní změny poloh enkodérů. V další části deskriptoru je uvedeno, že host může poslat zařízení 0 až 2 Byty. První Byte se využívá pro ovládání LED diod. Druhý Byte je nevyužit, ukázalo se však že po nastavení velikosti OUT endpointu, odpovídajících bufferů a konstant v deskriptorech na 1 Byte byla komunikace nespolehlivá a také byla patrná velká latence v ovládání. Proto byla ponechána velikost endpointu 2 Byty. Problém je zřejmě na straně aplikace v PC. Ovladač akceptuje i posílání 1 Bytu.

Skutečná velikost deskriptoru musí odpovídat konstantě `HID_RPT01_SIZE` definované v `usb_config.h`, v tomto případě 43 Bytů.

Výsledný obsah reportu je popsán v následující kapitole 4.3.


```
//Class specific descriptor - HID
ROM struct{BYTE report[HID_RPT01_SIZE];}hid_rpt01=
{
    0x06, 0x00, 0xff,           // USAGE_PAGE (Generic Desktop)
    0x09, 0x01,                 // USAGE (Vendor Usage 1)
    0xa1, 0x01,                 // COLLECTION (Application)
    0x19, 0x01,                 //   USAGE_MINIMUM (0)
    0x29, 0x05,                 //   USAGE_MAXIMUM (5)
    0x15, 0x00,                 //   LOGICAL_MINIMUM (0)
    0x26, 0xff, 0x00,           //   LOGICAL_MAXIMUM (255)
    0x75, 0x08,                 //   REPORT_SIZE (8)
    0x95, 0x03,                 //   REPORT_COUNT (3)
    0x81, 0x02,                 //   INPUT (Data,Var,Abs)
    0x19, 0x01,                 //   USAGE_MINIMUM (0)
    0x29, 0x02,                 //   USAGE_MAXIMUM (2)
    0x15, 0x80,                 //   LOGICAL_MINIMUM (-128)
    0x25, 0x7f,                 //   LOGICAL_MAXIMUM (127)
    0x75, 0x08,                 //   REPORT_SIZE (8)
    0x95, 0x02,                 //   REPORT_COUNT (2)
    0x81, 0x06,                 //   INPUT (Data,Var,Rel)
    0x19, 0x01,                 //   USAGE_MINIMUM (0)
    0x29, 0x02,                 //   USAGE_MAXIMUM (2)
    0x91, 0x02,                 //   OUTPUT (Data,Var,Abs)
    0xc0                         // END_COLLECTION
};
```

4.3 Obsah reportu

Obsah reportu je odpovídá report deskriptoru a je přehledně shrnut v Tabulka 4. Význam proměnných je popsán dále v kapitole 4.6.

Tabulka 4: Obsah reportu

IN	Bit								Popis, hodnota
index Bytu	7(MSB)	6	5	4	3	2	1	0(LSB)	
0	enc1_abs								0 – 255
1	enc2_abs								0 – 255
2	-	-	-	-	S4	S3	S2	S1	0 – 15
3	enc1_rel								-127 – 127
4	enc2_rel								-127 – 127
OUT	Bit								Popis, hodnota
index Bytu	7(MSB)	6	5	4	3	2	1	0(LSB)	
0	-	-	-	-	LED4	LED3	LED2	LED1	0 – 255
1	-								Nevyužito

4.4 Konfigurace USB modulu

Konfigurace je obsažena v souboru `usb_config.h`. Tento soubor obsahuje nastavení počtu velikostí endpointů, volbu přenosové rychlosti (Full speed), připojení interního pull-up odporu na sběrnici, a dále některé interní konstanty nutné pro funkci USB Framework. Obsahuje také volbu zpracování USB komunikace. Jsou dvě možnosti: `USB_POLLING` znamená nutnost obsluhy komunikace v hlavní smyčce, což je zajištěno opakovaným voláním

funkce USBDeviceTasks(). USB_INTERRUPT znamená obsluhu pomocí přerušení. Byla zvolena první možnost.

```

/** DEFINITIONS *****/
#define USB_EP0_BUFF_SIZE      8    // Platné hodnoty: 8,16,32,64 B.
#define USB_MAX_NUM_INT        1
#define USB_MAX_EP_NUMBER      1

#define USB_USER_DEVICE_DESCRIPTOR &device_dsc
#define USB_USER_DEVICE_DESCRIPTOR_INCLUDE extern ROM
USB_DEVICE_DESCRIPTOR device_dsc

#define USB_USER_CONFIG_DESCRIPTOR USB_CD_Ptr
#define USB_USER_CONFIG_DESCRIPTOR_INCLUDE extern ROM BYTE *ROM
USB_CD_Ptr[]

// Volba režimu komunikace
#define USB_POLLING
// #define USB_INTERRUPT

// Definice parametrů je v usb_device.h

// připojení interního pull-up odporu ke sběrnici
#define USB_PULLUP_OPTION USB_PULLUP_ENABLE
// #define USB_PULLUP_OPTION USB_PULLUP_DISABLED

#define USB_TRANSCEIVER_OPTION USB_INTERNAL_TRANSCEIVER
// #define USB_TRANSCEIVER_OPTION USB_EXTERNAL_TRANSCEIVER

// volba přenosové rychlosti, slouží pouze k připojení odpovídajícího
pull-up odporu, odpovídající nastavení oscilátoru musíme nastavit
pojistkami (fuses)
#define USB_SPEED_OPTION USB_FULL_SPEED
// #define USB_SPEED_OPTION USB_LOW_SPEED

#define USB_SUPPORT_DEVICE
#define USB_NUM_STRING_DESCRIPTOR 3
#define USB_ENABLE_ALL_HANDLERS

/** DEVICE CLASS USAGE *****/
#define USB_USE_HID

/** ENDPOINTS ALLOCATION *****/
/* HID */
#define HID_INTF_ID            0x00
#define HID_EP                 1    // číslo HID endpointu
#define HID_INT_OUT_EP_SIZE    2    // velikost HID endpointu OUT
#define HID_INT_IN_EP_SIZE     5    // velikost HID endpointu IN
#define HID_NUM_OF_DSC         1    // počet HID deskriptorů
#define HID_RPT01_SIZE         43   // velikost HID report deskriptoru
// HID report deskriptor je definován v usb_descriptors.c

```

4.5 Konfigurace procesoru

Funkce pro inicializaci a přístup k LED diodám, tlačítkům, a enkodérům jsou definovány v samostatném souboru HardwareProfile – PIC18F26J50.h. Například pro rozsvícení LED1 je nutné zapsat log. 0 na pin RB2 příkazem

```
TRISBbits.TRISB2 = 0;
```

Proto je v tomto souboru definováno makro:

```
#define mLED_1 TRISBbits.TRISB2
#define mLED_1_On() mLED_1 = 0;
```

V programu se pak rozsvícení LED1 dosáhne zápisem

```
mLED_1_On();
```

Tento postup umožňuje snadnou změnu zapojení nebo typu procesoru bez nutnosti pracné editace hlavního kódu. Stačí editovat pouze tento soubor.

V souboru jsou dále nastaveny některé potřebné konstanty, např. frekvence procesoru 48 MHz, a také definice použitého bootloaderu, která je důležitá pro správné přemapování adres resetu a přerušení. Tento soubor také obsahuje nastavení napájení zařízení ve vztahu k USB.

4.6 Hlavní funkce firmwaru (main), proměnné

Je obsažena v souboru main.c. Původní myšlenkou bylo využít dvouúrovňový systém přerušení daného mikroprocesoru, a obsluhovat USB přerušením s vysokou prioritou, otáčení enkodéry přerušením nízkou prioritou a pouze zbývající funkce v hlavní smyčce. Posléze se ukázalo jako výhodnější provádět obsluhu USB v hlavní smyčce. Obsluhu USB zároveň není příliš vhodné přerušovat jinými procesy, proto byly pro obsluhu enkodérů použity pouze příznakové bity externích přerušení. Tento postup výrobce procesoru také připouští. Ve firmwaru tedy není možné používat blokující funkce nebo čekací smyčky.

Pozice enkodérů jsou uloženy v globálních proměnných. Jsou reprezentovány dvěma způsoby. Proměnné `enc1_abs` a `enc2_abs` jsou absolutní čísla v rozsahu 0 až 255, s počáteční hodnotou 127. Tyto hodnoty jsou počítány nezávisle na spuštění obslužné aplikace v PC. Druhým způsobem je relativní reprezentace proměnnými `enc1_rel` a `enc2_rel`. Tyto hodnoty vyjadřují změnu polohy od poslední komunikace s USB, po každém odeslání jsou vynulovány. Záleží tedy jen na aplikaci v PC, které hodnoty bude výhodnější použít. Dalšími významnými proměnnými jsou buffery pro příjem a odesílání dat.

```
unsigned char ReceivedDataBuffer[2];
unsigned char ToSendDataBuffer[5];
unsigned char enc1_abs = 127;
unsigned char enc2_abs = 127;
signed char enc1_rel = 0;
signed char enc2_rel = 0;
```

Proměnné začínající písmeny `sw_` slouží pro obsluhu a odstranění zámků tlačítek. V proměnných jejichž označení končí `_old` jsou uloženy předchozí stavy enkodérů a tlačítek.

V úvodní části souboru main.c jsou také definována makra pro přičítání a odčítání proměnných při otáčení enkodérů.

```
#define Minus1(){enc1_rel--; if(enc1_abs>0) enc1_abs--;}
#define Plus1() {enc1_rel++; if(enc1_abs<255) enc1_abs++;}
#define Minus2(){enc2_rel--; if(enc2_abs>0) enc2_abs--;}
#define Plus2() {enc2_rel++; if(enc2_abs<255) enc2_abs++;}
```

Funkce main tedy vypadá následovně:

```
void main(void)
{
    InitializeSystem();
    while(1)
    {
```

```

        USBDeviceTasks();
        ProcessIO();
    } //end while
} //end main

```

Funkce `InitializeSystem()` zajišťuje inicializaci procesoru a USB modulu. Součástí této funkce je i volání funkce `UserInit()`, kam je možné přidat vlastní inicializační kód. Po dokončení inicializace se neustále opakuje provádění funkcí `USBDeviceTasks()` a `ProcessIO()`. Tyto funkce budou popsány v následujících podkapitolách.

4.6.1 Funkce `UserInit()`

V této funkci je provedena inicializace pinů pro LED diody, tlačítka a enkodéry. Dále obsahuje nastavení přerušení od obou enkodérů. Rodina procesorů PIC18 nepodporuje externí přerušení na obě hrany, proto se musí při inicializaci zjistit logická úroveň daného pinu a nastavit přerušení pro odpovídající hranu – pro log. 0 náběžnou (rising edge), pro log. 1 sestupnou (falling edge). Nastavením vstupů přerušení jsou také aktivovány Schmittovy klopné obvody na příslušných pinech.

```

void UserInit(void)
{
    // Inicializace enkodérů
    enc_init();
    // Inicializace LED diod
    mInitAllLEDs();
    // Inicializace tlačítek
    mInitAllSwitches();

    // Ošetření nezapojených pinů mikrokontroléru RA2, RA3, RA4, RC0
    TRISA &= 0B11100011;
    LATA  &= 0B11100011;
    TRISC &= 0B11111110;
    LATC  &= 0B11111110;

    // Inicializace přerušení, mapování pinů
    RPINR1 = 18;
    RPINR2 = 9;
    RPINR3 = 10;
    if (enc1) INTCON2bits.INTEDG0 = 0; // INT0 falling edge
    if (enc11) INTCON2bits.INTEDG1 = 0; // INT1 falling edge
    if (enc2) INTCON2bits.INTEDG2 = 0; // INT2 falling edge
    if (enc22) INTCON2bits.INTEDG3 = 0; // INT3 falling edge
    INTCONbits.INT0IF = 0; // vymazat Flag INT0
    INTCON3bits.INT1IF = 0; // vymazat Flag INT1
    INTCON3bits.INT2IF = 0; // vymazat Flag INT2
    INTCON3bits.INT3IF = 0; // vymazat Flag INT3

    // Inicializace proměnných pro informaci o předchozím přenosu
    USBOutHandle = 0;
    USBInHandle = 0;
} //end UserInit

```

4.6.2 Funkce `USBDeviceTasks()`

Tato funkce je obsažena v souboru `usb_device.c`. Zajišťuje vlastní komunikaci přes sběrnici. Pro zajištění spolehlivé funkce zařízení je nutné volat tuto funkci pravidelně, doporučený interval je cca 100 μ s. Jestliže tato podmínka nebude splněna, host zařízení odpojí. Vykonání funkce údajně trvá 50 až 100 cyklů.

4.6.3 Funkce ProcessIO()

Zajišťuje vlastní obsluhu uživatelských úkolů, v tomto případě snímání otáčení enkodérů a stisků tlačítek. Dále zajišťuje zpracování přijatých dat a přípravu dat k odeslání. Funkce BlinkUSBStatus() se dá využít k indikaci stavu USB zařízení pomocí dvou LED diod. Ve firmwaru je tato funkce zakázána. Obsluha enkodérů je řešena načítáním příznakových bitů přerušení v hlavní smyčce. Použité optické enkodéry nevyžadují žádné zvláštní rutiny pro odstraňování zákmitů (debouncing). Při načítání tlačítek jsou uloženy čtyři předchozí stavy. Nový stav tlačítek se reportuje až když jsou načteny 4 shodné po sobě následující stavy. Za další opatření proti zákmitům tlačítek lze považovat samotnou USB komunikaci, jejíž interval 10 ms je výrazně delší než doba zákmitu tlačítka (max. 2 ms [19]). Reporty jsou odesílány pouze pokud dojde ke změně stavů ovládacích prvků. Po navázání komunikace s PC aplikací je pak po každém přijetí dat z PC také odeslán report.

```
void ProcessIO(void){
    if((USBDeviceState<CONFIGURED_STATE)|| (USBSuspendControl==1))return;

    //Blink the LEDs according to the USB device status
    if(blinkStatusValid)
    {
        BlinkUSBStatus();
    };
    // Obsluha enkodérů
    if (INTCONbits.INT0IF)          // přerušení od pinu enc1 - INT0
    {
        if (!enc1 && !INTCON2bits.INTEDG0)//sestupná hrana
        {
            if (enc1) Plus1()          // zjištění směru otočení
            else Minus1();
            INTCON2bits.INTEDG0 = 1;    // nastavit přerušení na
                                        // náběžnou hranu
        }
        else if (enc1 && INTCON2bits.INTEDG0)//náběžná hrana
        {
            if (!enc1) Plus1()          // zjištění směru otočení
            else Minus1();
            INTCON2bits.INTEDG0 = 0;    // nastavit přerušení na
                                        // sestupnou hranu
        }
        INTCONbits.INT0IF=0;           // vymazat Flag
    };//end if (INTCONbits.INT0IF)

    if (INTCON3bits.INT1IF)          // přerušení od pinu enc11 - INT1
    {
        if (!enc11 && !INTCON2bits.INTEDG1)//sestupná hrana
        {
            if (!enc1) Plus1()          // zjištění směru otočení
            else Minus1();
            INTCON2bits.INTEDG1 = 1;    // nastavit přerušení na
                                        // náběžnou hranu
        }
        else if (enc11 && INTCON2bits.INTEDG1)//náběžná hrana
        {
            if (enc1) Plus1()           // zjištění směru otočení
            else Minus1();
            INTCON2bits.INTEDG1 = 0;    // nastavit přerušení na
                                        // sestupnou hranu
        }
        INTCON3bits.INT1IF=0;          // vymazat Flag
    }
```

```

}; //end if (INTCON3bits.INT1IF)

if (INTCON3bits.INT2IF)          // přerušení od pinu enc2 - INT2
{
    if (!enc2 && !INTCON2bits.INTEDG2) // sestupná hrana
    {
        if (enc22) Plus2()          // zjištění směru otočení
        else Minus2();
        INTCON2bits.INTEDG2 = 1;     // nastavit přerušení na
                                     // náběžnou hranu
    }
    else if (enc2 && INTCON2bits.INTEDG2) // náběžná hrana
    {
        if (!enc22) Plus2()          // zjištění směru otočení
        else Minus2();
        INTCON2bits.INTEDG2 = 0;     // nastavit přerušení na
                                     // sestupnou hranu
    }
    INTCON3bits.INT2F=0;              // vymazat Flag
}; //end if (INTCON3bits.INT2IF)

if (INTCON3bits.INT3IF)          // přerušení od pinu enc22 - INT3
{
    if (!enc22 && !INTCON2bits.INTEDG3) // sestupná hrana
    {
        if (!enc2) Plus2()           // zjištění směru otočení
        else Minus2();
        INTCON2bits.INTEDG3 = 1;     // nastavit přerušení na
                                     // náběžnou hranu
    }
    else if (enc22 && INTCON2bits.INTEDG3) // náběžná hrana
    {
        if (enc2) Plus2()             // zjištění směru otočení
        else Minus2();
        INTCON2bits.INTEDG3 = 0;     // nastavit přerušení na
                                     // sestupnou hranu
    }
    INTCON3bits.INT3IF=0;             // vymazat Flag
}; //end if (INTCON3bits.INT2IF)
// Konec obsluhy enkodérů*/

sw_3 = sw_2;
sw_2 = sw_1;
sw_1 = sw_0;
if (!sw1) sw_0 |= 0B00000001; else sw_0 &= 0B11111110; // Načtení
if (!sw2) sw_0 |= 0B00000010; else sw_0 &= 0B11111101; // tlačítek
if (!sw3) sw_0 |= 0B00000100; else sw_0 &= 0B11111011;
if (!sw4) sw_0 |= 0B00001000; else sw_0 &= 0B11110111;

if ((sw_3==sw_2)&&(sw_2==sw_1)&&(sw_1==sw_0)) sw_debounced = sw_0;

if(!HIDRxHandleBusy(USBOutHandle)) // Zkontrolovat jestli byla
                                     // přijata data
{
    // Nastavit LED podle přijatých dat
    if (ReceivedDataBuffer[0] & 1) mLED_1_On() else mLED_1_Off();
    if (ReceivedDataBuffer[0] & 2) mLED_2_On() else mLED_2_Off();
    if (ReceivedDataBuffer[0] & 4) mLED_3_On() else mLED_3_Off();
    if (ReceivedDataBuffer[0] & 8) mLED_4_On() else mLED_4_Off();

    if(!HIDTxHandleBusy(USBInHandle)) // jako odpověď odeslat data

```

```

{
// Připravit data pro odeslání:
ToSendDataBuffer[0]= enc1_abs;    // Absolutní pozice Enkodéru1
ToSendDataBuffer[1]= enc2_abs;    // Absolutní pozice Enkodéru2
ToSendDataBuffer[2]= sw_debounced; //Tlačítka
ToSendDataBuffer[3]= enc1_rel;    // Relativní pozice Enkodéru1
ToSendDataBuffer[4]= enc2_rel;    // Relativní pozice Enkodéru2
USBInHandle = HIDTxPacket(HID_EP, (BYTE*)&ToSendDataBuffer, 5);
sw_old = sw_debounced; // uložení předchozích hodnot
enc1_old = enc1_rel;
enc2_old = enc2_rel;
enc1_rel = 0;                // po odeslání vymazat relativní hodnoty
enc2_rel = 0;
}
USBOutHandle = HIDRxPacket(HID_EP, (BYTE*)&ReceivedDataBuffer, 2);
}; //end if (!HIDRxHandleBusy(USBOutHandle))

// Odeslat data pouze při změně
if (!HIDTxHandleBusy(USBInHandle) &&
((enc1_rel!=enc1_old) || (enc2_rel!=enc2_old) || (sw_debounced!=sw_old)))
    // pokud není odesílací rutina zaneprázdněna,
    // odeslat data
    {
        ...
    }
} //end ProcessIO

```

4.7 Testování splnění specifikace USB 2.0

K testování slouží program USB 2.0 Command Verifier dostupný přímo na stránkách USB Implenenters Forum [25]. Při testování je nutné ovladač připojit k počítači přes USB 2.0 hub. Po spuštění programu se nainstaluje dočasný testovací ovladač USB hubu. Následně je možné vybrat typ testu zařízení pro test. Byly provedeny celkem 4 testy, které budou níže stručně popsány. Záznamy o průběhu testů ve formátu HTML generované testovací aplikací jsou rovněž přiloženy na CD.

4.7.1 Current measurement tests

K tomuto testu je potřebná také kabelová redukce umožňující zapojení ampérmetru do obvodu. Po spuštění test nastaví program testované zařízení nejdříve do nekonfigurovaného stavu a vyzve uživatele ke změření odebíraného proudu. Poté je nastavena první konfigurace a opět se zobrazí výzva ke změření odběru. Takto se změří všechny dostupné konfigurace. V nekonfigurovaném stavu nesmí zařízení odebírat více než 100 mA. V konfigurovaných stavech nesmí proud překročit hodnotu uvedenou v příslušném deskriptoru konfigurace, která je u tohoto ovladače také 100 mA. Naměřené hodnoty proudu jsou shrnuty v Tabulka 5. Naměřené hodnoty jsou menší než dovolené, test je tedy splněn úspěšně.

Tabulka 5: Měření proudového odběru ovladače

Stav ovladače	Proud. odběr
Nekonfigurovaný	47 mA
Konfigurace 1	47 mA
Max. odběr (všechny LED rozsvíceny)	88 mA
Min. odběr (pouze běžící procesor, napájení enkodérů vypnuto)	14 mA

4.7.2 Chapter 9 tests

Tyto testy ověřují dodržování Kapitoly 9 USB specifikace, ověřuje se především obsah deskriptorů zařízení a konfigurace a jejich vyžádání ve stavu adresovaném a konfigurovaném. Na závěr testu je provedeno 150 enumerací zařízení. Test byl úspěšný.

4.7.3 HID tests

Tyto testy ověřují obsah report deskriptoru, odezvu na nastavení nečinnosti (GetIdle) a další. Test byl úspěšný.

4.8 Alternativní firmware

Funkce HID zařízení je z velké části dána report deskriptorem. Vytvořením nového deskriptoru dosáhneme zcela jiné funkce zařízení, s nutností pouze minimálních úprav ostatních částí firmwaru. Pro demonstraci byly vytvořeny dva nové firmwary:

4.8.1 Myš

V tomto příkladu se levým enkodérem ovládá pohyb kurzoru svisle (v ose Y), pravým enkodérem pak pohyb vodorovně (v ose X). S1 má funkci levého tlačítka myši, S4 funguje jako pravé tlačítko, S3 jako střední tlačítko. Tlačítko S2 není využito. Ovládání myši rotačními enkodéry by se dalo využít pro přesné ovládání kurzoru v pravoúhlých souřadnicích např. při rýsování, pro skutečné použití by však bylo vhodné použít enkodéry generují několikanásobně více impulzů na otáčku.. Report deskriptor má tento obsah:

```
0x05, 0x01, /* Usage Page (Generic Desktop) */
0x09, 0x02, /* Usage (Mouse) */
0xA1, 0x01, /* Collection (Application) */
0x09, 0x01, /* Usage (Pointer) */
0xA1, 0x00, /* Collection (Physical) */
0x05, 0x09, /* Usage Page (Buttons) */
0x19, 0x01, /* Usage Minimum (01) */
0x29, 0x03, /* Usage Maximum (03) */
0x15, 0x00, /* Logical Minimum (0) */
0x25, 0x01, /* Logical Maximum (0) */
0x95, 0x03, /* Report Count (3) */
0x75, 0x01, /* Report Size (1) */
0x81, 0x02, /* Input (Data, Variable, Absolute) */
0x95, 0x01, /* Report Count (1) */
0x75, 0x05, /* Report Size (5) */
0x81, 0x01, /* Input (Constant) ;5 bit padding */
0x05, 0x01, /* Usage Page (Generic Desktop) */
0x09, 0x30, /* Usage (X) */
0x09, 0x31, /* Usage (Y) */
0x15, 0x81, /* Logical Minimum (-127) */
0x25, 0x7F, /* Logical Maximum (127) */
0x75, 0x08, /* Report Size (8) */
0x95, 0x02, /* Report Count (2) */
0x81, 0x06, /* Input (Data, Variable, Relative) */
0xC0, 0xC0} /* End Collection,End Collection */
```


4.8.2 Multimediální ovladač

Funkce tohoto ovladače jsou běžně implementovány v multimediálních klávesnicích. Přesto by ovladač s tímto firmwarem mohl být použit např. k zabudování do čelního panelu počítače sloužícího jako centrum domácího kina (Home Theater PC). Podpora pro tento ovladač je integrována ve všech moderních operačních systémech, funguje tedy ihned po připojení k PC. Levým enkodérem se ovládá posun ve skladbě (Rewind/Fast Forward), tato funkce však zřejmě funguje pouze v přehrávači Windows Media Player. Pravým enkodérem se ovládá systémová hlasitost (Volume). Funkce tlačítek jsou následující:

S1 – přehrávání/pauza (Play/Pause)

S2 – předchozí skladba (Scan prev track)

S3 – následující skladba (Scan next track)

S4 – umlčení zvuku (Mute)

Funkce ovladače byla ověřena na operačních systémech Windows XP a Vista. Report deskriptor multimediálního ovladače je zde:

```
0x05, 0x0c, // USAGE_PAGE (Consumer Devices)
0x09, 0x01, // USAGE (Consumer Control)
0xa1, 0x01, // COLLECTION (Application)
0x19, 0x00, // USAGE_MINIMUM (Unassigned)
0x2A, 0xFF, 0x03, // USAGE_MAXIMUM (1023)
0x95, 0x01, // REPORT_COUNT (1)
0x75, 0x10, // REPORT_SIZE (16)
0x15, 0x00, // LOGICAL_MINIMUM (0)
0x26, 0xff, 0x03, // LOGICAL_MAXIMUM (1023)
0x81, 0x00, // INPUT (Data, Ary, Abs)
0xc0 // END_COLLECTION
```

4.9 Jiné možnosti tvorby firmware

V předchozích kapitolách byla popsána tvorba firmwaru úpravou vzorového projektu poskytnutého výrobcem procesoru. To však není jediná možnost.

Firma Trace Systems nabízí aplikaci HIDmaker FS, která generuje kompletní kód pro procesor PIC18F4550 a několik dalších typů, a to podle požadavku buď v Assembleru, v jazyce C nebo v Basic. Také vytváří kompletní aplikaci na straně počítače, a to pro celou řadu vývojových prostředí, včetně použitého MS Visual C++ Express Edition [26].

Další možností je volně šiřitelný program EasyHID firmy Mecanique. Tento program je součástí vývojového prostředí MicroCode Studio. Umožňuje vygenerovat kód pro procesor v jazyce Basic, a kostru PC aplikace pro Delphi 5, Visual Basic 5.0 nebo Visual C++ 6.0 [27].

5 PROGRAM NA STRANĚ PC

Program v počítači musí nejprve detekovat přítomnost zařízení na sběrnici a následně navázat spojení. Po úspěšném navázání spojení je možné přijímat a posílat data přes sběrnici USB.

Program vychází z ukázkové aplikace obsažené v balíku Microchip USB Framework [21], konkrétně z aplikace obsažené ve složce USB Device - HID - Custom Demos\Generic HID - PnP Demo - PC Software. Je vyvíjen v prostředí Microsoft Visual C++ 2008 Express Edition, které je volně dostupné [28]. Hlavní část kódu je obsažena v souboru Form1.h. Ukázkový kód byl vytvořen ve starší verzi 2005, ale toto prostředí při otevření nabízí jednoduchou konverzi. Výsledný program je spustitelný .exe soubor. Drobnou nevýhodou programů vyvinutých v tomto prostředí je nutnost mít pro spuštění programu nainstalovaný balík Microsoft .NET Framework ve verzi 2.0 nebo novější. Ten je standardně obsažen ve Windows Vista, ve Windows XP je nutné jej doinstalovat manuálně.

5.1 Ukázkové kódy od výrobce procesoru

Microchip poskytuje tři různá demo komunikující se stejným zařízením, a funkčně stejné – umožňují načtení stavu tlačítka a ovládání LED diody.

5.1.1 Simple Demo

Nejjednodušší je Simple demo. Ukazuje komunikaci s USB zařízením použitím standardních metod ReadFile a WriteFile. Počáteční vyhledání zařízení je však řešeno pouze ručním kliknutím na tlačítko. Program neposkytuje informace o odpojení zařízení, a pro reálné použití tak není příliš vhodný.

5.1.2 HID DLL demo

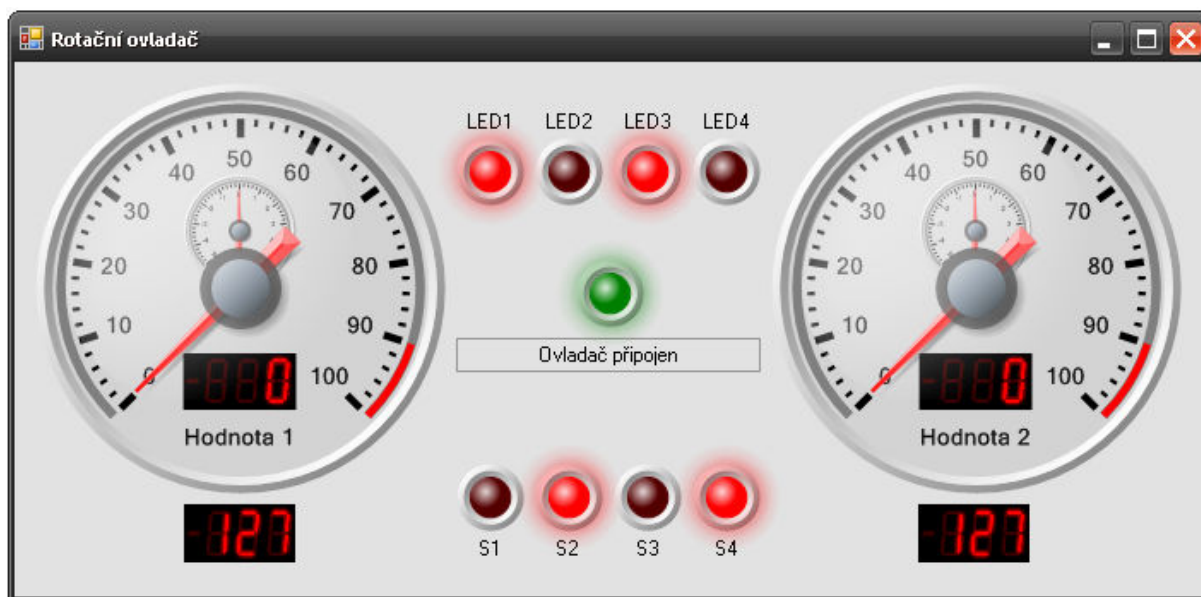
Propracovanější je HID DLL demo. V tomto programu již detekce připojeného zařízení probíhá automaticky v pravidelných intervalech. Nevýhodou je nutnost přiložit k výslednému programu knihovnu HID class.dll. Cílem bylo vytvořit plně samostatný exe soubor, a proto toto demo také nebylo použito.

5.1.3 PnP Demo

Nejdokonalejší z těchto tří je PnP Demo. Jak již název napovídá, obsahuje plnou podporu Plug&Play. Významné části kódu budou popsány v následujících kapitolách.

5.2 Uživatelské rozhraní aplikace, proměnné

Tvorba uživatelského rozhraní je v použitém vývojovém prostředí snadná. Pro přehledné zobrazení dat přijímaných z ovladače byly použity grafické komponenty z balíčku InstrumentLab 4.5 od Mitov Software [29]. Tento balíček obsahuje velké množství zdařilých indikačních prvků a je k dispozici pro celou řadu vývojových prostředí – nejen použitý MS Visual C++, ale i Borland Delphi a C++ Builder a další. Uživatelské rozhraní s připojeným ovladačem je na Obr. 7.



Obr. 7: Uživatelské rozhraní ovládacího programu

Připojení ovladače je indikováno svitem zelené LED diody uprostřed okna programu, a dále také textovou informací pod ní.

V horní části okna jsou umístěny čtyři červené LED diody označené LED1 až LED4. Kliknutím se ovládá svit těchto LED diod a také odpovídajících LED diod na ovladači. Stav LED diod se ukládá do 4 nejnižších bitů proměnné `leds`.

Čtyři červené LED diody označené S1 až S4 zobrazují aktuální stav tlačítek – při stisknutí tlačítka na ovladači svítí příslušná LED. Stav tlačítek je uložen ve 4 nejnižších bitech proměnné `buttons`.

Stavy enkodérů jsou vyjádřeny proměnnými pojmenovanými stejně jako ve firmwaru procesoru. I jejich význam je naprosto stejný a je popsán v kap. 4.6. Pomocí relativních hodnot z enkodérů `enc1_rel` a `enc2_rel` jsou měněny dvě uživatelské celočíselné proměnné, `my_val1` a `my_val2` v rozsahu 0 až 100. Tyto dvě proměnné jsou zobrazeny přehlednými ručičkovými ukazateli, které jsou doplněny ještě digitálními displeji. Přijímané relativní hodnoty jsou zobrazeny pomocí malých ručiček uvnitř hlavních ukazatelů. Níže jsou pomocí digitálních displejů vypsány absolutní hodnoty `enc1_abs` a `enc2_abs`, tyto proměnné mají po připojení ovladače hodnotu 127 a mění se otáčením enkodérů v rozsahu od 0 do 255.

Důležitou proměnnou je `AttachedState`, vyjadřující stav připojení ovladače.

Uživatelské rozhraní je periodicky obnovováno časovačem `FormUpdateTimer`. Kromě aktualizace zobrazovaných údajů získaných z globálních proměnných tento Timer zajišťuje také zablokování uživatelského rozhraní při odpojení ovladače, resp. odblokování po připojení ovladače. Zkrácený a komentovaný zdrojový kód obsluhy `FormUpdateTimeru` je uveden zde:

```
Void FormUpdateTimer_Tick(System::Object^sender, System::EventArgs^ e)
{
    // Zkontrolovat, jestli je zařízení připojeno
    if(AttachedState == TRUE)
    {
        // Připojeno, povolit uživatelské rozhraní
        textBox1->Text = "Ovladač připojen";
    }
}
```

```

        led9->Value = true;
        ...
    }
    if((AttachedState == FALSE) || (AttachedButBroken == TRUE))
    {
        // Nepřipojeno, zablokovat uživatelské rozhraní
        textBox1->Text = "Ovladač odpojen";
        ...
    }

    // Pokud je připojen ovladač, aktualizovat uživatelské rozhraní
    // nejnovějšími daty získanými z ReadWriteThread
    if(AttachedState == TRUE)
    {
        // Aktualizovat zobrazení uživatelských proměnných
        angularGauge1->Value= my_val1;
        angularGauge2->Value= my_val2;
        externalSegmentGaugeElement1->Value = my_val1;
        externalSegmentGaugeElement2->Value = my_val2;

        // Dekódovat stisky tlačítek a zobrazit pomocí CheckBoxů
        if (buttons & 1) led5->Value = true; else led5->Value = false;
        if (buttons & 2) led6->Value = true; else led6->Value = false;
        if (buttons & 4) led7->Value = true; else led7->Value = false;
        if (buttons & 8) led8->Value = true; else led8->Value = false;

        // Zakódovat stav LED diod
        leds_temp = 0;
        if (led1->Value) leds_temp += 1;
        if (led2->Value) leds_temp += 2;
        if (led3->Value) leds_temp += 4;
        if (led4->Value) leds_temp += 8;
        leds = leds_temp;
        // Zobrazit absolutní a relativní hodnoty přijaté z ovladače
        segmentGauge1->Value = enc1_abs;
        segmentGauge2->Value = enc2_abs;
        externalAngularGaugeElement1->Value = enc1_rel;
        externalAngularGaugeElement2->Value = enc2_rel;
    }

} //end Timet_Tick

```

5.3 Detekce připojení zařízení

Detekce připojení a odpojení byla beze změny převzata ze vzorového projektu. Je založena na principu čtení zpráv od Windows přijímaných aplikací. Z mnoha zpráv jsou využívány WM_DEVICECHANGE, které aplikaci informují o změně v zařízeních připojených k počítači. Zdrojový kód je poměrně obsáhlý a proto jej nemá smysl uvádět. Příslušné bloky jsou ve zdrojovém kódu výrazně označeny komentáři BEGIN CUT AND PASTE BLOCK a END CUT AND PASTE BLOCK, což usnadňuje jejich nalezení a použití. Stav zařízení je vyjádřen logickou globální proměnnou AttachedState, podle jejíž hodnoty se dále řídí chování celé aplikace. Pokud je zařízení úspěšně připojeno a komunikace navázána, má tato proměnná hodnotu true.

Připojené zařízení je vyhledáváno podle identifikačních čísel VID a PID, které je nutné definovat stejně jako ve firmwaru zařízení:

```
#define MY_DEVICE_ID "Vid_04d8&Pid_feec"
```

5.4 Komunikace přes sběrnici USB

Komunikaci zajišťuje komponenta `BackgroundWorker` pojmenovaná `ReadWriteThread`. Výhoda takového řešení je, že uživatelské rozhraní je dále funkční, i když dojde k problému v USB komunikaci. Tato funkce předává data v obou směrech mezi globálními proměnnými a příslušnými endpointy. Součástí této funkce je také obsluha dvou uživatelských proměnných. Zdrojový kód je komentován a je ve zkrácené podobě uveden níže:

```
Void ReadWriteThread_DoWork(System::Object^ sender,
System::ComponentModel::DoWorkEventArgs^ e) {

    unsigned char OUTBuffer[3]={0,0,0};    //Alokovat paměť velkou
                                           //jako OUT endpoint + 1
    unsigned char INBuffer[6];             //Alokovat paměť velkou
                                           //jako IN endpoint + 1

    DWORD BytesWritten = 0;
    DWORD BytesRead = 0;

    while(true)    // nekonečná smyčka
    {
        if(AttachedState == TRUE) //Nepokoušet se číst/zapisovat, dokud není
zařízení připojeno a připraveno
        {
            OUTBuffer[0] = 0;            // První byte je "Report ID" a nepřenáší se
                                           // přes USB, vždy nastavit 0
            OUTBuffer[1] = leds;         // Zapsat požadovaný stav LED diod

            if(WriteFile(WriteHandleToUSBDevice,&OUTBuffer,3,&BytesWritten, 0))
            {
                INBuffer[0] = 0;
                // Získat odpověď od firmware
            }//end writefile
            if(ReadFile(ReadHandleToUSBDevice, &INBuffer, 6, &BytesRead, 0))
            {
                //INBuffer[0] je Report ID, nestaráme se o něj
                enc1_abs = INBuffer[1]; // Abs. hodnota Enkoderu1
                enc2_abs = INBuffer[2]; // Abs. hodnota Enkoderu2
                buttons = INBuffer[3]; // Hodnota tlačítek,
                                           // 0.bit = SW1,..., 3. bit = SW4
                                           // 1 = tlačítko stiknuto
                enc1_rel = INBuffer[4]; // Rel. hodnota Enkoderu1
                enc2_rel = INBuffer[5]; // Rel. hodnota Enkoderu2

                //Ukázka použití relativních hodnot enkodérů
                //pro nastavení dvou uživatelských proměnných
                if (first_connected)
                {enc1_rel=0;                //po spuštění aplikace
                  enc2_rel=0;                //vymazat předchozí hodnoty
                  first_connected=false;}
                my_val1 += enc1_rel;
                if (my_val1 < my_val1_min) my_val1 = my_val1_min;
                if (my_val1 > my_val1_max) my_val1 = my_val1_max;
                my_val2 += enc2_rel;
                if (my_val2 < my_val2_min) my_val2 = my_val2_min;
                if (my_val2 > my_val2_max) my_val2 = my_val2_max;
            }
            //}end writefile - původní se závislou komunikací
        } //end of: if(AttachedState == TRUE)
    }
    else
    {
```

```
Sleep(5);    // Vložit malé zpoždění, jinak bude nekonečná
//smyčka příliš vytěžovat procesor PC bez přínosu pro aplikaci
}

} //end of while(true) loop
```

5.5 Známé problémy, chyby a omezení

Ke spuštění aplikace je nutné mít nainstalovaný balík Microsoft .NET Framework verze 2.0 nebo vyšší – to je dáno použitým vývojovým prostředím.

Při připojení ovladači nejdou spustit hry založené na DirectX – jedná se zřejmě o chybu v Microchip USB Framework, nebo rozhraní DirectInput, které je součástí DirectX. Stejnou chybu vykazují i ukázkové kódy. Řešením důsledku je připojit zařízení až po spuštění hry.

Ovladač má ve spícím režimu (suspend mode) spotřebu výrazně vyšší než dovořených 2,5 mA – při návrhu zapojení bylo počítáno s možností odpojení napájení optických enkodérů ve firmwaru, ale při probuzení ze spícího režimu docházelo ke generování náhodných impulzů, proto bylo rozhodnuto tuto funkci raději neimplementovat.

6 ZÁVĚR

V práci byl popsán kompletní návrh rotačního ovladače jako USB periferie ve třídě HID. Ovladač je založen na procesoru za rodiny PIC18F firmy Microchip, který v sobě integruje USB řadič a další potřebné periferie. Pro tento procesor byl vytvořen firmware. Stěžejním bodem návrhu firmware je konstrukce deskriptorů, zejména pak report deskriptoru. Díky tomu je ovladač správně rozpoznán operačním systémem a je možné s ním komunikovat.

V další části práce je popsán návrh obvodového zapojení, návrh plošného spoje a mechanického uspořádání. Ovladač díky použité krabičce dobře padne do ruky a je dostatečně robustní.

Pro demonstraci komunikace byla vytvořena aplikace pro operační systémy Windows. V aplikaci je použita poměrně robustní detekce připojení ovladače a díky tomu je možné ovladač připojovat a odpojovat při spuštěné aplikaci. Aplikace disponuje efektním grafickým uživatelským rozhraním, které přehledně zobrazuje hodnoty získané z ovladače.

V závěru bylo ověřeno splnění specifikace USB 2.0. Ovladač prošel všemi provedenými testy.

Nad rámec zadání byl vytvořen firmware umožňující použít tento ovladač jako myš, a dále firmware ovládající hudební přehrávač v PC.

SEZNAM POUŽITÉ LITERATURY

- [1] AXELSON, J. *USB Complete The Developer's Guide Fourth Edition*, Madison: Lakewiev Research LLC, 2009, 504 s. ISBN13 978-1-931448-08-6
- [2] USB IMPLEMENTERS FORUM. *USB 2.0 Specification*, [online] cit. 22. listopadu 2009. Dostupné na WWW: <http://www.usb.org/developers/docs/>
- [3] MALÝ, M. *USB 2.0 – díl 1*, [online] cit. 2. prosince 2009. Dostupné na WWW: <http://hw.cz/Rozhrani/ART1232-USB-2.0---dil-1.html>
- [4] MALÝ, M. *USB 2.0 – díl 2*, [online] cit. 2. prosince 2009. Dostupné na WWW: <http://hw.cz/Rozhrani/ART1244-USB-2.0---dil-2.html>
- [5] MALÝ, M. *USB 2.0 – Typy a formáty přenosů*, [online] cit. 4. prosince 2009. Dostupné na WWW: <http://hw.cz/Rozhrani/ART1264-USB-2.0---Typy-a-formaty-prenosu.html>
- [6] OBJECTIVE DEVELOPMENTS SOFTWARE GMBH. *V-USB - Virtual USB port for AVR microcontrollers*. [online] cit. 26. prosince 2009. Dostupné na WWW: <http://www.obdev.at/products/vusb/index.html>
- [7] ATMEL CORPORATION. *AT90USB82/162 Summary*, [online] cit. 10. listopadu 2009. Dostupné na WWW: http://www.atmel.com/dyn/resources/prod_documents/7707S.pdf
- [8] ATMEL CORPORATION. *ATmega16U4 ATmega32U4 Preliminary Summary*, [online] cit. 10. listopadu 2009. Dostupné na WWW: http://www.atmel.com/dyn/resources/prod_documents/7766S.pdf
- [9] MICROCHIP TECHNOLOGY INC. *PIC18F13K50/14K50 20-Pin USB Flash Microcontrollers with nanoWatt Technology*, [online] cit. 10. listopadu 2009. Dostupné na WWW: <http://ww1.microchip.com/downloads/en/DeviceDoc/41350C.pdf>
- [10] MICROCHIP TECHNOLOGY INC. *PIC18F46J50 Family Data Sheet*, [online] cit. 10. listopadu 2009. Dostupné na WWW: <http://ww1.microchip.com/downloads/en/DeviceDoc/39931c.pdf>
- [11] MICROCHIP TECHNOLOGY INC. *PIC18F2455/2550/4455/4550 Data sheet*, [online] cit. 10. listopadu 2009. Dostupné na WWW: <http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>
- [12] TOMÍČEK, P. *USB mikroprocesory PIC a jejich použití - 1. díl*, [online] cit. 10. listopadu 2009. Dostupné na WWW: <http://www.mcu.cz/news.php?extend.623.2>
- [13] TOMÍČEK, P. *USB mikroprocesory PIC a jejich použití - 2. díl*, [online] cit. 10. listopadu 2009. Dostupné na WWW: <http://www.mcu.cz/news.php?extend.624.2>
- [14] MAŠLAŇ, S. *Použití rotačních enkodérů*, [online] cit. 23. listopadu 2009. Dostupné na WWW: <http://elektronika.kvalitne.cz/ATMEL/necoteorie/tutorial/RotaryEncoder/RotaryEncoder.html>
- [15] GRAYHILL INC. *Series 62E Optical Encoders*, [online] cit. 23. prosince 2009. Dostupné na WWW: <http://lgrws01.grayhill.com/web/images/ProductImages/Series62PEncoder.pdf>
- [16] NXP SEMICONDUCTOR. *Integrated quad ultra-low capacitance ESD protection*, [online] cit. 10. května 2010. Dostupné na WWW: http://www.nxp.com/documents/data_sheet/PRTR5V0U4D.pdf
- [17] HAMMOND MFG. *Soft Sided Hand Held Enclosures (1553 Series)*, [online] cit. 11. května 2010. Dostupné na WWW: <http://www.hammondmfg.com/pdf/1553D.pdf>
- [18] MENTOR. *Aluminium Turning Knobs with Setscrew Fixing*, [online] cit. 11. května 2010. Dostupné na WWW: <http://www.mentor-bauelemente.de/db/pdf/k/24.pdf>
- [19] MEC A/S. *Multimec 3C/3E*, [online] cit. 11. května 2010. Dostupné na WWW:

<http://mec.live.inlead.dk/files/webCE.pdf>

- [20] MICROCHIP TECHNOLOGY INC. *MPLAB Integrated Development Environment*. [online] cit. 23. prosince 2009. Dostupné na WWW: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002
- [21] MICROCHIP TECHNOLOGY INC. *Microchip Application Libraries*. [online] cit. 23. prosince 2009. Dostupné na WWW: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2680&dDocName=en537044
- [22] MICROCHIP TECHNOLOGY INC. *PICkit 2 Programmer/Debugger User's Guide* [online] cit. 11. května 2010. Dostupné na WWW: <http://ww1.microchip.com/downloads/en/DeviceDoc/51553E.pdf>
- [23] USB IMPLEMENTERS FORUM. *Device Class Definition for HID 1.11*, [online] cit. 30. listopadu 2009. Dostupné na WWW: http://www.usb.org/developers/devclass_docs/HID1_11.pdf
- [24] INTEL CORP. *HID Descriptor Tool*, [online] cit. 2. prosince 2009. Dostupné na WWW: http://www.usb.org/developers/hidpage/dt2_4.zip
- [25] USB IMPLEMENTERS FORUM. *USB 2.0 Command Verifier*, [online] cit. 16. května 2010. Dostupné na WWW: <http://www.usb.org/developers/tools/>
- [26] TRACE SYSTEMS INC. *HIDmaker FS*, [online] cit. 26. prosince 2009. Dostupné na WWW: http://www.tracesystemsinc.com/usb_tools.ivnu
- [27] MECANIQUE. *EasyHID*, [online] cit. 26. prosince 2009. Dostupné na WWW: <http://www.mecanique.co.uk/code-studio/index.html>
- [28] MICROSOFT CORPORATION. *Visual C++ 2008 Express Edition*. [online] cit. 26. prosince 2009. Dostupné na WWW: <http://www.microsoft.com/Express/VC/>
- [29] MITOV SOFTWARE. *InstrumentLab 4.5*, [online] cit. 16. května 2010. Dostupné na WWW: <http://www.mitov.com/html/instrumentlab.html>

SEZNAM SYMBOLŮ A ZKRATEK

b	bit, nejmenší jednotka informace nebo kapacity paměti, může nabývat logické hodnoty 0 nebo 1
B	Byte [bajt], jednotka množství dat, platí že $1\text{ B} = 8\text{ b}$
k	kilo, předpona mocniny 10^3 , při použití s b nebo B pak 2^{10}
M	mega, předpona vyjadřující mocninu 10^6
s	sekunda, jednotka času
Hz	Hertz, jednotka kmitočtu, počet cyklů za sekundu
F	Farad, jednotka elektrické kapacity
PWM	Pulse Width Modulation, pulzně šířková modulace
GND	Ground, elektrická zem, nulový potenciál
SMD	Surface Mounted Devices, součástky pro povrchovou montáž.
LDO	Low Drop-Out, lineární stabilizátor napětí s nízkým úbytkem

Případné další zkratky a symboly jsou vysvětleny v textu.

OBSAH PŘILOŽENÉHO CD

- elektronická verze této práce ve formátu PDF
- ukázková aplikace pro PC (Rotacni_ovladac.exe), zdrojové kódy pro Microsoft Visual C++ 2008 Express Edition nebo vyšší
- firmware pro mikroprocesor PIC18F26J50 (Rotacni_ovladac.hex), zdrojové kódy pro MPLAB IDE 8.30 nebo vyšší
- bootloader pro tento mikroprocesor, obslužná aplikace
- schéma zapojení, návrh plošného spoje a knihovny s enkodéry pro Eagle verze 4.60 nebo vyšší
- katalogové listy vybraných mikroprocesorů
- katalogové listy rotačního enkodéru a krabičky
- alternativní firmware – myš a multimediální ovladač
- záznamy z testování ovladače programem USB 2.0 Command Verifier ve formátu HTML

SEZNAM PŘÍLOH

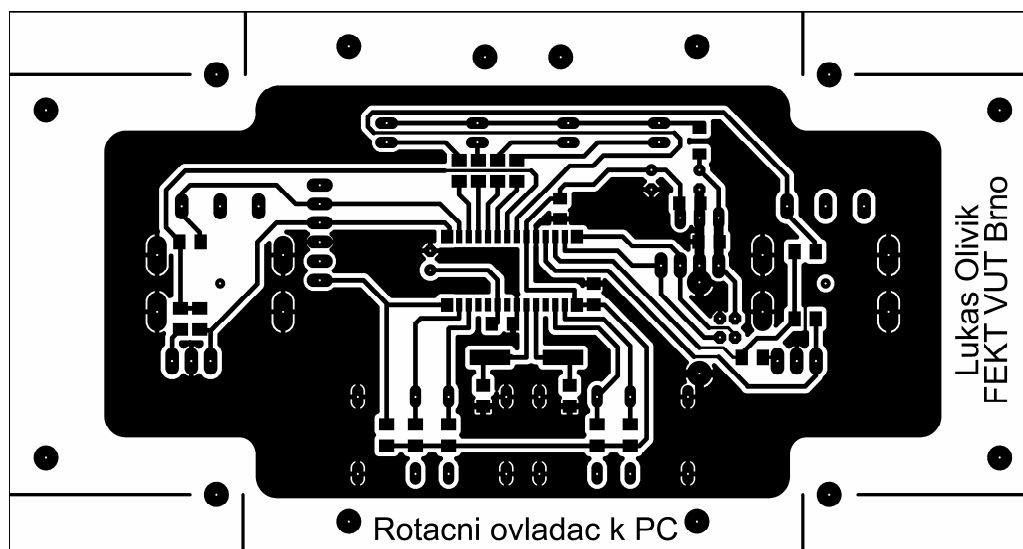
A	NÁVRH ZAŘÍZENÍ.....	39
A.1	DESKA PLOŠNÉHO SPOJE (BOTTOM).....	39
A.2	OSAZOVACÍ PLÁN.....	40
A.3	SEZNAM SOUČÁSTEK.....	41

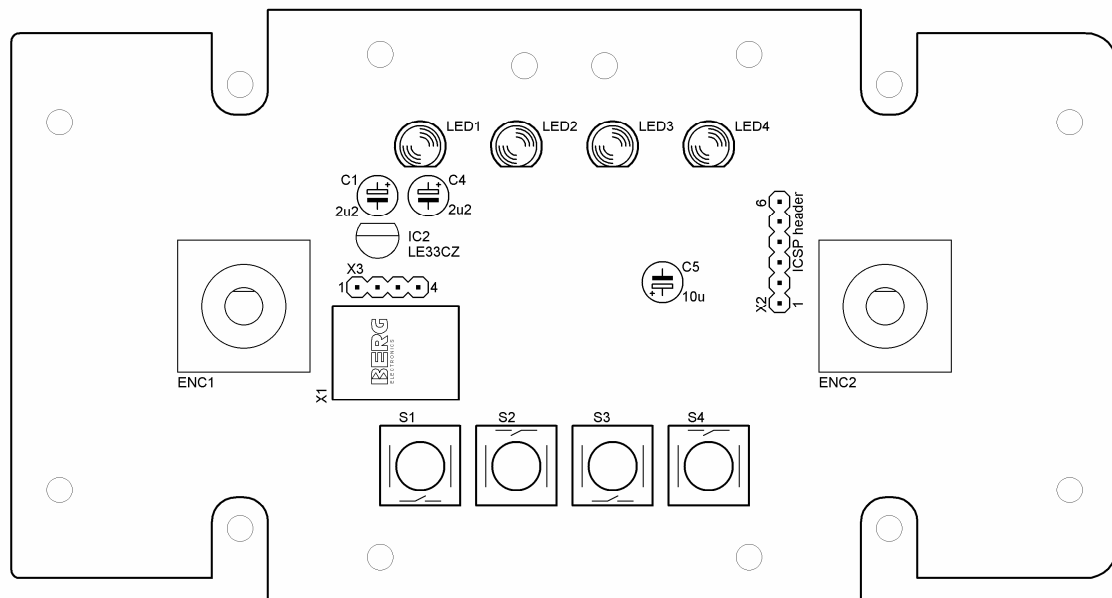
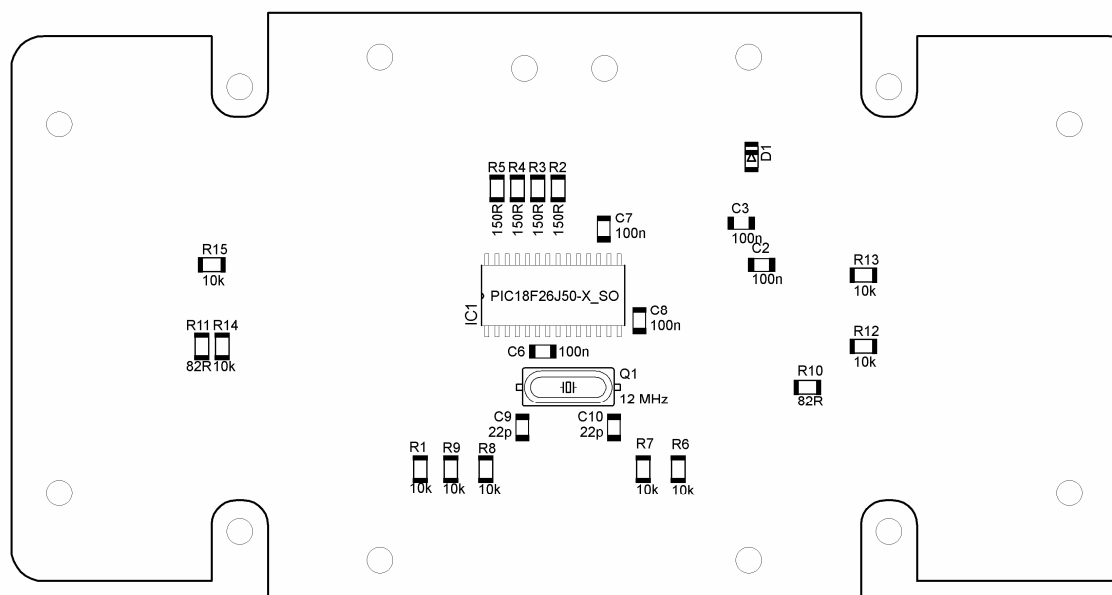
Pozn.: Schéma zapojení je obsaženo v textu na straně 11.

A NÁVRH ZAŘÍZENÍ

A.1 Deska plošného spoje (bottom)

Měřítko 1:1, rozměry 136 x 72 mm, horizontálně převráceno



Strana TOP:Strana **BOTTOM**

A.3 Seznam součástek

Označení	Hodnota/typ	Pouzdro	Popis
IC1	PIC18F26J50	SOIC28	Mikrokontrolér Microchip
IC2	LE33CZ	TO-92	LDO stabilizátor 3,3 V
Q1	12 MHz	HC49UP	Krystal SMD
D1	1N4007	MINIMELF	Dioda
LED1		LED5MM	LED dioda 5 mm červená
LED2		LED5MM	LED dioda 5 mm červená
LED3		LED5MM	LED dioda 5 mm červená
LED4		LED5MM	LED dioda 5 mm červená
ENC1	62P-22L0		Rotační enkodér Grayhill
ENC2	62P-22L0		Rotační enkodér Grayhill
S1	3ETL9-15.0		Tlačítko Multimec
S2	3ETL9-15.0		Tlačítko Multimec
S3	3ETL9-15.0		Tlačítko Multimec
S4	3ETL9-15.0		Tlačítko Multimec
X1		PN61729-S	USB konektor typ B do DPS
X2	ICSP header	MA06-1	Pinová lišta 2,54 mm
X3		MA04-1	Pinová lišta 2,54 mm
C1	2u2	E2,5-5	Tantalový kondenzátor
C2	100n	C1206	Keramický kondenzátor
C3	100n	C1206	Keramický kondenzátor
C4	2u2	E2,5-5	Tantalový kondenzátor
C5	10u	E2,5-5	Tantalový kondenzátor
C6	100n	C1206	Keramický kondenzátor
C7	100n	C1206	Keramický kondenzátor
C8	100n	C1206	Keramický kondenzátor
C9	22p	C1206	Keramický kondenzátor
C10	22p	C1206	Keramický kondenzátor
R1	10k	M1206	Rezistor
R2	150R	M1206	Rezistor
R3	150R	M1206	Rezistor
R4	150R	M1206	Rezistor
R5	150R	M1206	Rezistor
R6	10k	M1206	Rezistor
R7	10k	M1206	Rezistor
R8	10k	M1206	Rezistor
R9	10k	M1206	Rezistor
R10	82R	M1206	Rezistor
R11	82R	M1206	Rezistor
R12	10k	M1206	Rezistor
R13	10k	M1206	Rezistor
R14	10k	M1206	Rezistor
R15	10k	M1206	Rezistor