

# **MIDI protokol a jevištní zařízení**

## **MIDI protocol and stage equipment**

Bc. Dalibor Slovák

---

Diplomová práce  
2008



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav aplikované informatiky

akademický rok: 2007/2008

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Dalibor SLOVÁK**  
Studijní program: **N 3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**  
Téma práce: **MIDI protokol a jevištní zařízení**

Zásady pro vypracování:

1. Rešerše stávajících dostupných řešení.
2. Výhody a nevýhody jednotlivých zařízení.
3. Návrh vlastního řešení a jeho zdůvodnění.
4. Popis použitých technologií a programového vybavení.
5. Praktická část.
6. Závěr a zhodnocení.



doc. Ing. Ivan Zelinka, Ph.D.  
vedoucí ústavu

prof. Ing. Vladimír Vašek, CSc.  
děkan

Rozsah práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. Popis procesoru PIC18F2550 [online]. Dostupný z WWW: "<http://www.microchip.com>"
2. Specifikace protokolu USB 2.0 [online]. Dostupný s WWW: "[www.usb.org/developers/usb\\_20.zip](http://www.usb.org/developers/usb_20.zip)"
3. Popis MIDI standardu [online]. Dostupný z WWW: "<http://www.midi.com>"
4. Nušl, J. Řízení techniky komunikující protokolem DMX512 pomocí dat získaných z hudby v reálném čase. Praha. České vysoké učení technické v Praze, Fakulta elektrotechnická, katedra počítačů, 2006. 58 s. Vedoucí diplomové práce Ing. Jiří Buček.
5. Mach, J. Firmware pro USB zařízení s mikropočítači PIC. Praha. České vysoké učení technické v Praze, Fakulta elektrotechnická, katedra počítačů, 2006. 50 s. Vedoucí diplomové práce Ing. Miroslav Skrbek, Ph.D.
6. Gardner N., An Introduction to programming the Microchip PIC in C. 2. vydání. Londýn: Nakladatelství Bluebird Electronics, 1998. ISBN 1-899013-04-0.

Vedoucí diplomové práce:

**prof. Ing. Vladimír Vašek, CSc.**

Ústav automatizace a řídicí techniky

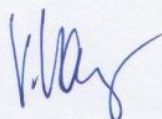
Datum zadání diplomové práce:

**22. února 2008**

Termín odevzdání diplomové práce:

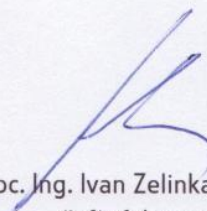
**19. května 2008**

Ve Zlíně dne 22. února 2008



prof. Ing. Vladimír Vašek, CSc.

*děkan*



doc. Ing. Ivan Zelinka, Ph.D.

*ředitel ústavu*

## **ABSTRAKT**

Tato diplomová práce se v první části zabývá popisem stávajících řešení koordinace světelných zařízení spolu s hudbou na bázi ovládání pomocí MIDI protokolu. Shrnuje vlastnosti jednotlivých řešení a popisuje jejich výhody a nevýhody. V další části se pak zabývá popisem řešení založeného na procesoru Microchip 18F2550 a jeho implementaci. Stěžejní problematikou řešenou v této diplomové práci je programování jednočipové aplikace v rámci protokolů USB a MIDI. Praktická část se zabývá implementací tohoto řešení do stávajícího modelu netriviální fontány dostupné v laboratořích FAI UTB. V závěru jsou pak shrnuty dosažené výsledky.

Klíčová slova: MIDI protokol, Microchip, Universal Serial Bus, pulzně-šířková modulace

## **ABSTRACT**

This diploma thesis in first part describes current solutions of the coordination lights equipment in conjunction with music based on control by MIDI protocol. This work summaries properties of particular equipment and describes their advantages and disadvantages. In other part describes solution based on processor Microchip 18F2550 and his implementation. Main problem solved in this work is single-chip application programming. Practical part describes implementation of this equipment to current non-trivial water fountain in the laboratory of Faculty of Applied Informatics in Tomas Bata University in Zlin. At the conclusion are summarized attained results.

Keywords: MIDI protocol, Microchip, Universal Serial Bus, pulse- width modulation

Na tomto místě chci poděkovat prof. Ing. Vladimíru Vaškovi, CSc. za odborné vedení, cenné rady a připomínky a také za možnost pracovat právě na tomto tématu. Dále chci poděkovat Ing. Davidu Příbylovi za spolupráci a přínosné podněty a připomínky při vytváření hardwarových součástí a jejich optimalizaci.

Děkuji paní Janě Slovákové a Daliboru Janu Slovákovi za trpělivost a inspiraci.

Prohlašuji, že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně, 1. září 2008

.....  
Podpis diplomanta

**OBSAH**

<b>ÚVOD.....</b>	<b>8</b>
<b>1 STÁVAJÍCÍ ŘEŠENÍ.....</b>	<b>11</b>
1.1 PROTOKOL DMX 512.....	11
1.1.1 Technická specifikace EIA485 (DMX512 protokolu).....	12
1.2 ŘEŠENÍ HARVEYE TWYMANA.....	12
1.3 FIRMA BOTEX A JEJÍ ŘEŠENÍ .....	13
1.4 MIDI HARDWARE TOMA SCARFFA .....	14
1.5 ŘEŠENÍ ING. NUŠLA.....	15
1.6 SYSTÉM SPOLEČNOSTI DARYS .....	16
1.6.1 Modul ML-80P2.....	16
1.6.2 Modul ML-640 PS .....	16
1.7 SPOLEČNOST MIDI LITE A JEJÍ ZAŘÍZENÍ.....	18
1.7.1 LanBox - LCX.....	18
1.7.2 LanBox - LCE .....	20
1.7.3 LanBox - LCM .....	20
1.7.4 MidiLite II Dimmer System.....	20
1.8 SHRNU TÍ.....	21
<b>2 VÝHODY A NEVÝHODY STÁVAJÍCÍCH ŘEŠENÍ.....</b>	<b>22</b>
2.1 TOVÁRNÍ, KOMERČNĚ DOSTUPNÁ ŘEŠENÍ .....	22
2.2 DALŠÍ ŘEŠENÍ .....	22
<b>3 NÁVRH VLASTNÍHO ŘEŠENÍ A JEHO ZDŮVODNĚNÍ .....</b>	<b>23</b>
3.1 VOLBA HARDWAROVÉHO ROZHRANÍ .....	23
<b>4 POPIS POUŽITÝCH TECHNOLOGIÍ A SYSTÉMOVÉHO VYBAVENÍ.....</b>	<b>24</b>
4.1 USB STANDARD .....	24
4.2 USB STANDARD PRO AUDIO ZAŘÍZENÍ .....	27
4.3 USB STANDARD PRO MIDI ZAŘÍZENÍ.....	27
4.3.1 Element .....	28
4.3.2 Vstupní a výstupní terminál .....	29
4.3.3 USB-MIDI Converter.....	29
4.3.4 Popis deskriptorů USB MIDI zařízení .....	29
4.4 MIDI PROTOKOL .....	32
4.5 USB MIDI EVENT PACKET .....	32
4.6 HARDWARE .....	35
4.6.1 Microchip PIC 18F2550.....	36
4.6.2 Zařízení USB MIDI Lights Device jako celek.....	37
4.7 VÝVOJOVÉ NÁSTROJE A POUŽITÝ FIRMWARE .....	39
4.7.1 MPLAB IDE .....	39
4.7.2 MCC C18 .....	40
4.7.3 Firmware – moduly a soubory .....	41



4.8	BOOTLOADER.....	45
4.8.1	Colt PIC18F Bootloader.....	46
4.8.2	Picdem FS USB.....	46
4.9	TESTOVACÍ NÁSTROJE .....	48
4.9.1	USB Trace .....	48
4.9.2	Virtual MIDI keyboard .....	49
4.9.3	Midi.exe .....	50
<b>5</b>	<b>PRAKTICKÁ ČÁST .....</b>	<b>52</b>
5.1	IDENTIFIKACE DO SYSTÉMU.....	52
5.2	PŘENOS DAT .....	52
5.3	VYUŽITÍ ZAŘÍZENÍ V RÁMCI MIDI (HUDEBNÍHO) SOFTWARE .....	53
5.4	ODEZVA KONCOVÉHO ZAŘÍZENÍ NA USB MIDI LIGHTS DEVICE.....	54
5.5	DALŠÍ MOŽNÉ CESTY VÝVOJE POPISOVANÉHO ZAŘÍZENÍ.....	55
5.5.1	Ovládání pomocí celé klaviatury .....	55
5.5.2	Využití jedné MIDI stopy pro více výstupních zařízení .....	55
5.5.3	Generátor světelné stopy s využitím Fourierovy transformace.....	55
5.5.4	Uživatelské rozhraní pro nastavení USB MIDI Lights Device.....	56
	<b>ZÁVĚR .....</b>	<b>57</b>
	<b>CONCLUSION .....</b>	<b>58</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>59</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>61</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>63</b>
	<b>SEZNAM TABULEK.....</b>	<b>64</b>

## ÚVOD

Využití počítačové techniky v hudbě je v současnosti standardem, jak pro oblast živého hraní, tak pro studiovou práci. V rámci živého hraní patří mezi nejvyužívanější technologické standardy z oblasti digitální techniky MIDI protokol. Je základním standardem pro nastavení, která souvisí především s nadefinovanými stavy elektronických nebo elektroniku využívajících hudebních nástrojů nebo také nastavení dalších jevištních zařízení, např. světelné techniky. Někteří hudebníci jej používají vědomě a cíleně, jiní naopak nevědomky a necíleně, neboť některá ze zařízení užívají protokol MIDI jako součást svého vnitřního vybavení a nevyžadují pro svou obsluhu MIDI událostí lidský zásah. MIDI protokol je dnes součástí klávesových nástrojů, strunných nástrojů, ale existují i MIDI kontroléry založené na principu dechových hudebních nástrojů. Tyto kontroléry vyrábí např. společnost Yamaha.

MIDI protokol je dnes také součástí efektových jednotek pro veškeré hudební instrumenty. Ať už se jedná o elektrické kytary nebo klávesové nástroje, ale také efektové jednotky pro zpěváky nebo pro akustické nástroje. MIDI protokol je také součástí dalších zařízení, která mají co dočinění s jevištěm a produkcí na něm. Tedy MIDI protokol je dnes součástí jevištního osvětlení a dalších efektových zařízení jeviště (umělé dýmy apod.). Současný vývoj a využití tohoto protokolu se zaměřuje především na koordinaci nastavení hudebních nástrojů a dalších zařízení na jevišti. Tedy především časová koordinace produkované hudby s dalšími vizuálními efekty na jevišti bez využití lidského faktoru. V rámci tohoto vývoje se MIDI protokol využívá v kombinaci s nejmodernějšími technologiemi. Velmi často je využíván spolu s USB.

V rámci protokolu MIDI existují různá řešení, která ve své podstatě nebyla zamýšlena pro ovládání světelné techniky, ale s úspěchem by se dala k tomuto účelu využít. Chtěl jsem ale, aby se tato práce opírala o mé vlastní řešení, a proto jsem se nezabýval možností použít některé ze stávajících řešení k danému účelu a klonovat je pro ovládání zařízení, které je dostupné v laboratořích FAI.

Cílem této diplomové práce je přesně koordinovat a propojit pomocí USB a MIDI protokolu vodní fontánu a světla, která jsou dostupná v laboratořích FAI UTB spolu s PC, na kterém je nainstalováno programové prostředí Cubase SX, který bude sloužit jako řídicí software. Toto téma plynule navazuje na bakalářskou práci mou, Petra Lukačoviče, Pavla Křeménka a Petra Hoffmanna. Výsledkem by mělo být nejen teoretické, ale především



v praxi proveditelné a použitelné řešení, poskytující možnost ovládat a obsluhovat pomocí dostupných prostředků daná zařízení.

Obsahem této diplomové práce je provést rešerži, která se zaměřuje na popis a zhodnocení stávajících zařízení, principů a postupů, pomocí kterých lze koordinovat hudbu a další jevištní zařízení. Dále pak navrhnout, podrobně popsat a aplikovat vlastní řešení. Hotový návrh pak otestovat, vyzkoušet a použít k ovládání dostupného zařízení. V průběhu celého vývoje dbát na to, aby výrobní a vývojové náklady byly co nejnižší a stejně tak, aby i provoz a údržba byly co nejlevnější. Dále, aby celé zařízení bylo maximálně univerzální a snadno nastavitelné, resp. naprogramovatelné.

## **I. TEORETICKÁ ČÁST**

## 1 STÁVAJÍCÍ ŘEŠENÍ

Společným znakem většiny stávajících řešení je protokol DMX512. Proto je první část této kapitoly věnována jeho stručnému popisu. Dále pak je třeba zmínit, že nalezená řešení nejsou popsána zcela do hloubky. Jedním z důvodů, proč se tak neděje, je omezený prostor této práce, a po té také malá nebo vůbec žádná dostupnost dokumentace k jednotlivým řešením.

### 1.1 Protokol DMX 512

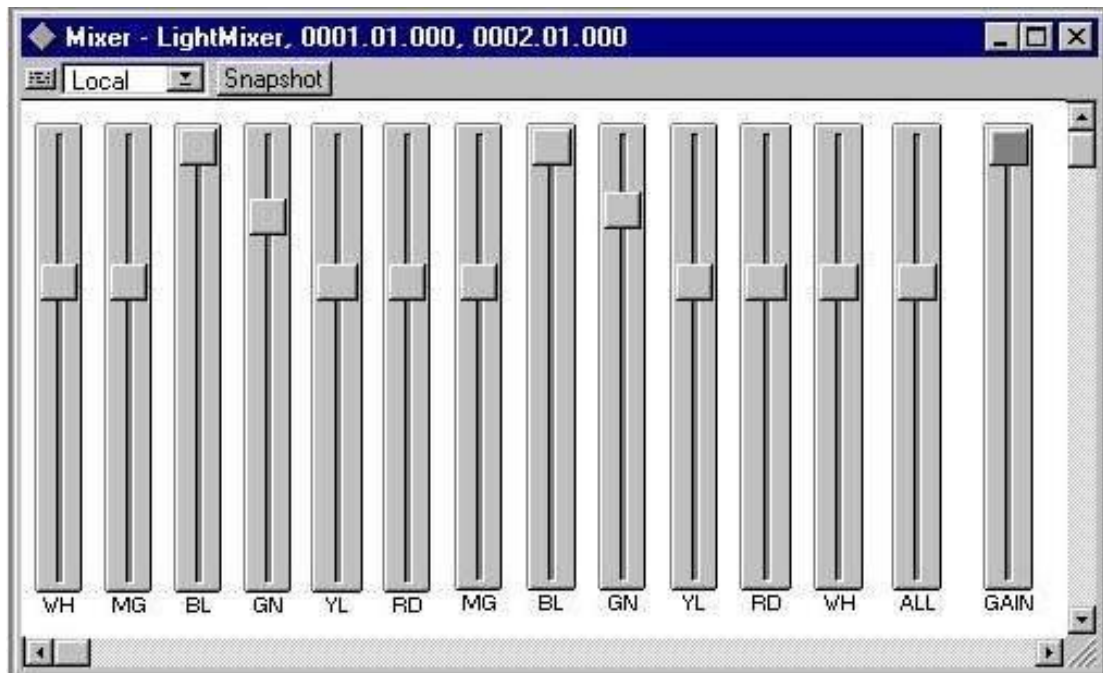
Stávající řešení využívají především výše jmenovaný protokol. Tento patří ve většině případů k základní výbavě všech profesionálních zařízení, která slouží k provozu světelné jevištní techniky. Je první digitální náhradou analogového řízení, kde základní řídicí veličinou je konkrétní hodnota napětí. Jeho specifikace vychází z průmyslového standardu EIA485, který byl dříve značen jako RS-485, případně RS485. Protokol DMX512 byl navržen v roce 1986 institutem USITT (United States Institute for Theatre Technology) pro řízení stmívačů a dalších speciálních efektů pomocí digitálního rozhraní. Měl nahradit do té doby používané analogové řízení, kde jako řídicí veličina sloužila výše zmiňovaná konkrétní hodnota napětí na řídicím kabelu. S tím bylo spojeno hned několik potíží. Pro každý řízený vstup (např. jeden kanál stmívače) byl potřeba jeden vodič, navíc nebyla vždy přesně dána řídicí veličina, neboť vlivem výrobních tolerancí mohla různá zařízení chápat či naopak vydávat mírně odlišné povely. Také analogový přenos byl samozřejmě náchylnější na rušení, což v blízkosti několika-kilowattových stmívacích jednotek nebylo zrovna velké plus. Stejně jako MIDI je i řídicí protokol DMX512 sériovým protokolem. Základem protokolu DMX512 je jeho elektrická specifikace. Ta vychází z již zmiňovaného, osvědčeného průmyslového standardu EIA485. Vzhledem k používání tohoto standardu v průmyslu jsou i technické prostředky pro jeho implementaci levné a přizpůsobené pro náročné podmínky. Pro přenos se využívá tří-pinový případně pěti-pinový konektor XLR. Jelikož data jsou přenášena po dvou vodičích a třetí vodič je zem, u pěti-pinového konektoru je čtvrtý a pátý pin nevyužíván. V praxi se používá především tří-pinový konektor. Pokud je v rámci jedné světelné sestavy využíváno více světelných jednotek, jsou zapojovány topologií sběrnice. Jednotlivá zařízení se určují adresně pomocí tabulek, které vytváří vždy výrobce daného zařízení. Vzhledem k impedančnímu přizpůsobení je nutno připojovat k poslední jednotce terminátor [2].

### 1.1.1 Technická specifikace EIA485 (DMX512 protokolu)

- Diferenciální (symetrický) napěťový přenos po kroucené dvojlince se schopností pracovat od napájecího napětí 5 V.
- Rozsah přípustného napětí na sběrnici od 7 V do +12 V .
- Možnost připojení až 32 přijímačů v jednom segmentu (existují budiče umožňující použít až 256 přijímačů). Impedanční přizpůsobení linky – odporem (terminátor) - 120 Ohmů. Minimální zatěžovací impedance vysílače je 60 Ohmů. Maximální zkratový proud vysílače je 150 mA proti zemi, 250 mA proti 12 V.
- Maximální délka kabelu je 4000 stop (přibližně 1200 m) při přenosové rychlosti max. 400 kBit/s z rozsahu 0 až 10 V.
- Počet segmentů není limitován.
- Budič musí být schopen dodat na sběrnici rozdílové vstupní napětí od 1,5 do 5 V.
- Budič musí mít ochranu pro případ, že by se více budičů pokoušelo vysílat na sběrnici.
- Přijímač by měl mít minimální vstupní impedanci 12 k $\Omega$ .
- Přijímač musí mít minimální vstupní citlivost +-200 mV.
- Toto vše musí být přijímač schopen zpracovat i při souhlasném stejnosměrném napětí na sběrnici od -7 V do +12 V [2], [4].

## 1.2 Řešení Harveye Twymana

Hardwarový modul Harveye Twymana sestává z integrovaného obvodu HC11, který vyrábí Motorola. Tento obvod vysílá MIDI zprávy do dalšího zařízení, kterým je Altera 8254 FPGA ( Field Programmable Gate Array). Tento modul poskytuje celkem 12 kanálů. Jmenovitý výkon jednotlivých kanálů může dosáhnout hodnoty až 300W. Takto koncipované ovládání poskytuje 128 světelných úrovní pro jednotlivá světla. Nastavení těchto úrovní je pak možno provádět přímo z editačních programů, jako je např. Cubase. K tomuto nastavení se dá konkrétně využít Key Graphics Editor, List Editor, případně Mixer Maps Editor. Autor využívá poslední ze jmenovaných editorů. Tyto editory jsou součástí hudebního softwaru Cubase.



Obr. 1. Nastavení barevné intenzity v MixerMaps editoru v programu Cubase

Každý ze světelných kanálů má tři parametry, kterými je možno kanál ovládat. Jsou jimi úroveň kanálu (Channel Level), kanálový zisk (Channel Gain) a celková úroveň (Master Gain):

- Úroveň kanálu (Channel Level) – rozsah hodnot je od nuly až do 127. Nastavení tohoto parametru ale úzce souvisí s parametrem celkové úrovně (viz. dále).
- Kanálový zisk (Channel Gain) – slouží jako nastavení světlosti daného kanálu, protože ty se, dle autora, jeví často světlejší, než je třeba.
- Celkový zisk (Master Gain) – slouží k nastavení celkové světlosti všech kanálů [5].

### 1.3 Firma Botex a její řešení

Mezi další zařízení, která používají MIDI protokol pro ovládání jevištního osvětlení, patří zařízení společnosti Botex. Tato zařízení ale kombinují MIDI s DMX512 protokolem. Z MIDI protokolu využívají především událost Note On (nota zapnuta) a Note Off (nota vypnuta). Jako ukázkový model jsem vybral čtyřkanálový dimmer Botex MPX – 405. Mezi jeho schopnosti patří:

- ovládání pomocí MIDI a DMX,
- infračervené dálkové ovládání,

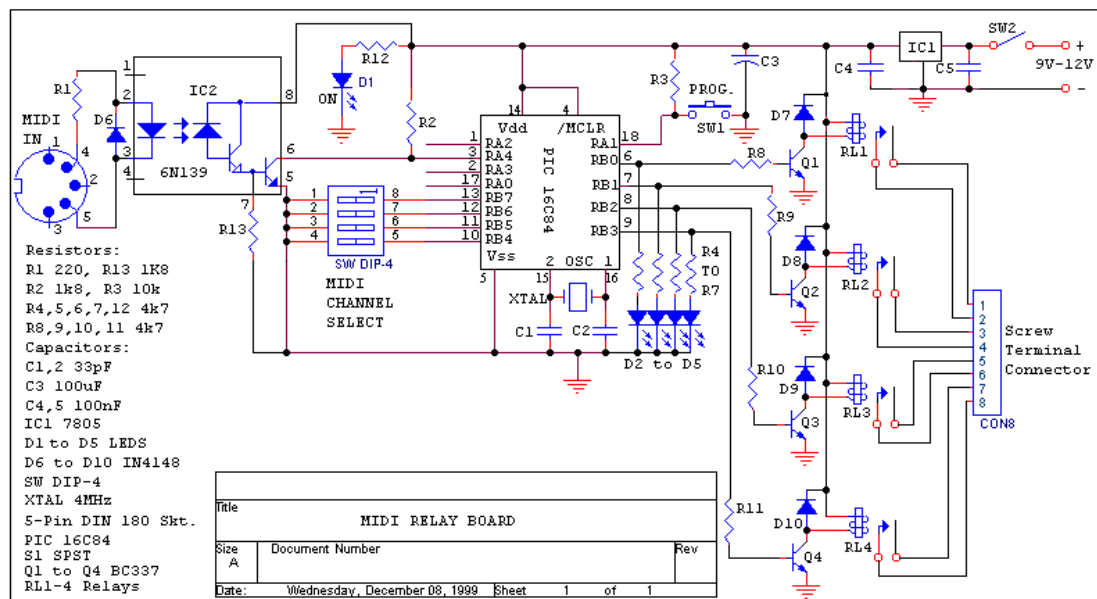
- 4 kanálový výstup pro dimmer,
- možnost propojení více jednotek (Link Up),
- paměť posledního nastavení [6].



Obr. 2. Botex MPX – 405

#### 1.4 MIDI Hardware Toma Scarffa

Autor tohoto řešení pochází z Dubliny, kde také působí a pracuje na dublinském technologickém institutu. Tom Scarff používá podobnou technologii, jako je využita v rámci této diplomové práce. Deska je osazena čipem, MIDI konektorem, potřebnými LED diodami a také v neposlední řadě konektorem pro devítí-voltovou baterii, která slouží jako napájení. Srdcem celého modulu je Microchip PIC16C84 spolu s přepínačem pro výběr daných kanálů. Tedy výběr výstupního signálu z daného MIDI kanálu není řešen pomocí naprogramované utility, ale je řešen napevno, hardwarově. Jedná se o nejjednodušší případ řízení světel, potažmo jiných elektrických zařízení, pouze pomocí příkazu Note On, resp. zapnuto, vypnuto. Ovládaná světla jsou pouze spínána. Nijak není ovládaná jejich intenzita záření [7].



Obr. 3. Schéma zapojení MIDI hardwaru Toma Scarffa

## 1.5 Řešení Ing. Nušla

I když se nejedná přímo o zařízení, které by využívalo MIDI protokol, považuji za nutné jej tady zmínit, protože se jedná o nejucelnější a nejkomplexnější hardwarově-sofwarovou soustavu pro „živé“ svícení. Již podle příjmení řešitele je patrné, že se jedná o tuzemský, tedy český produkt. Samotný Ing. Nušl celé toto řešení prezentoval jako svou vlastní diplomovou práci, na které vystavěl pak existenci své firmy zabývající se právě osvětlením různých kulturních akcí. Jeho řešení je založeno na analýze audio signálu v reálném čase, na základě které je pak generováno osvětlení jeviště. Základním stavebním prvkem je rozhraní mezi USB a DMX512 protokolem, které má na starosti především převod napěťových úrovní mezi TTL logikou a logikou DMX512. Další, velmi podstatnou součástí tohoto řešení, je analýza audio signálu v reálném čase. Děje se tak pomocí výpočtu hlasitosti ze signálu vyjádřeného v časové oblasti. Z tohoto je pak dále vytvořena sada několika dílčích příznaků, na základě kterých jsou již vytvářeny jednotlivé scény nasvícení. Mezi základní příznaky patří samozřejmě analýza hlasitosti pro různá frekvenční pásma. Ke korekci vstupní hlasitosti je po té využívána Rychlá Fourierova Transformace. Autor pro řízení světelných scén navrhl více než dvacet algoritmů. Některé z nich jsou popsány v jeho diplomové práci. Velmi podrobná vizualizace umožňuje nejen detailní nastavení audio vstupního signálu, ale také velmi podrobné přiřazení výstupních



DMX kanálů a nastavení světelných scén, které je možno i ukládat pro pozdější použití. Také je zde pomocí vizualizace monitorována aktivita na jednotlivých DMX kanálech [2].

## 1.6 Systém společnosti Darys

Polská společnost, která se zabývá výrobou světelných ovladačů komunikujících pomocí MIDI.

### 1.6.1 Modul ML-80P2

. Jejich produkt ML-80P2 je ovládán pomocí MIDI zpráv Note On, Note Off, Program Change a Control Change. Je schopen ovládat až osm světelných kanálů. Výstupní napěťový signál je potom v napěťovém rozmezí 0 - 10 V. Je možno nastavit intenzitu svícení v rozmezí 127 hodnot. Pro úplnost hmotnost zařízení je 0,4 kg a jeho rozměry jsou 190x130x40 mm. K danému zařízení je nutný externí napájecí zdroj (12V) [8].



Obr. 4. MIDI kontroler ML-80P2

### 1.6.2 Modul ML-640 PS

Vlajkovou lodí této společnosti je ale zařízení s typovým označením ML-640 PS. Toto zařízení jako výstup plně využívá DMX512 protokol, tedy je nutné vlastnit světelná zařízení, která umí komunikovat a především přijímat příkazy tohoto protokolu. Cena tohoto modulu je 400,- € [8]. Zařízení umí pracovat ve čtyřech módech. V osmi, šestnácti,

třiceti-dvou nebo šedesáti-čtyř kanálovém módu. Počet programů, které lze použít, vzrůstá s klesajícím číslem počtu DMX kanálů, které chce uživatel programovat. Viz. tabulka:

Mód	Počet programů	Počet scén	Čísla programů
mód 8	256	2048	A01-A64, B01-B64, C01-C64, D01-D64
mód 16	128	1024	A01-A32, B01-B32, C01-C32, D01-D32
mód 32	64	512	A01-A16, B01-B16, C01-C16, D01-D16
mód 64	32	256	A01-A08, B01-B08, C01-C08, D01-D08

Tab. 1. Přehled jednotlivých módů zařízení ML 640-PS

Zařízení je rozděleno do čtyř programovatelných bank: A, B, C a D. Každá banka obsahuje až osm programovatelných scén. Každá z těchto scén může být uživatelem programována dle jeho vlastních potřeb. Daná nastavení je pak možno ukládat do sekvencí. Maximální počet sestavených sekvencí je 16. Každou sekvenci lze sestavit maximálně s osmi programů. Dále pak zařízení využívá dva módy pro sepnutí světelných jednotek nebo světelných skupin:

1. Mód 1 – První z událostí Note On rozsvěcí světlo, následující Note On světlo zhasíná
2. Mód 2 – Událost Note On světlo rozsvěcí, událost Note Off světlo zhasíná.



Obr. 5. Modul ML-640 PS

## 1.7 Společnost MIDI Lite a její zařízení

Jedná se o zámořskou společnost, která se zabývá vývojem světelných zařízení ovládaných nejen pomocí MIDI, ale také DMX protokolem. Mezi její světelná zařízení využívající MIDI protokol patří např:

### 1.7.1 LanBox - LCX

Toto zařízení má ve své výbavě pro komunikaci se světelnou technikou kromě USB portu také Ethernet, MIDI In/Out a DMX In/Out porty. Součástí zařízení je i 37 pinový D sub konektor pro připojení externích zařízení. Tyto porty lze využívat všechny najednou. LCX zařízení lze užívat v kombinaci s dalšími LCX boxy nebo s menšími modely, které nesou označení LCE, a vytvořit až 6 systémů využívajících zároveň DMX protokol a získat tak 3072 DMX kanálů. Spolu se zařízením je dodávám software LCedit a grafický editor světelných scén pro MacOS Mac OS X a Windows98SE/2000/XP. Mezi součásti dodávaného zařízení patří i ukázkové příklady. Toto zařízení a všechna další od společnosti MIDI Lite jsou dostupná pouze na americkém a kanadském trhu.

Souhrn obecných parametrů zařízení LanBox – LCX:

- možno využít až 512 DMX kanálů,
- v rámci propojení je možno získat ve spojení s dalším LCX boxem případně s pěti boxy řady LCE až 3072 kanálů,
- tři různé rychlosti přenosu,
- diody indikující aktivitu na výstupních kanálech,
- DMX vstup může být sloučen s DMX výstupem a použit jako vstup pro světelné scény.

Parametry týkající se přenosu po síti:

- přenos DMX informací je realizován pomocí UDP protokolu,
- podpora protokolu AppleTalk a kompatibilních zařízení,
- podpora protokolu TCP/IP se statickými IP adresami,
- podpora připojení dalších LCX boxů,
- podpora platform Mac i PC jak pro editaci a monitoring, tak i pro řízení.

Standardy pro komunikaci zařízení s okolím:

- AppleTalk (ADSP), TCP/IP pro editaci a ovládání externích zařízení prostřednictvím ethernetu,
- USB,
- MIDI,
- překlad MIDI událostí Note On/Off na příslušné hodnoty DMX protokolu,
- nativní spouštění a výběr jednotlivých scén pomocí Go, Next/Prev,
- plná editace pomocí MIDI SysEx zpráv.

Vstupy a výstupy:

- 8 analogových vstupů napětíové úrovně 0-5V s rozlišením 10 bitů.
- 8 digitálních výstupů, maximální výstupní proud 20mA [9].



Obr. 6. LanBox - LCX

### 1.7.2 LanBox - LCE

Tento model je zmenšenou verzí řady LCX. Může pracovat jako samostatné stand alone zařízení nebo může komunikovat s dalšími zařízeními pomocí Ethernetu. Ke komunikaci využívá stejnou sadu příkazů stejně jako modul LCX. Zařízení je možno také doplnit dálkovým ovládním. Ostatní technické parametry související s ovládním tohoto zařízení jsou shodné jako u modelu LCX. Pouze velikost paměti Flash je poloviční [9].



Obr. 7. LanBox - LCE

### 1.7.3 LanBox - LCM

Opět zmenšená verze řady LCX. Parametry se opět ve většině shodují jak s model LCX, tak s modelem LCE. Základním rozdílem je protokol pro komunikaci. V tomto případě se jedná především o sériový a MIDI protokol, jenž využívá toto zařízení pro příjem daných povelů. Tak jako předchozí modely transformuje příchozí MIDI zprávy do DMX příkazů, které jsou potom použity pro ovládní světelných sestav [9].

### 1.7.4 MidiLite II Dimmer System

Jedná se o systém, který je pro velké nízko-výkonové světelné sestavy, kdy umožňuje ovládat a spouštět až 1 350 kanálů. Jeho využití je spojeno především se všemi možnými druhy MIDI zařízení, která jsou schopna vysílat MIDI zprávy, které MIDI Lite II System využije k ovládní světel. Jelikož se jedná o rackové zařízení, je možno jej přizpůsobit individuálním potřebám uživatele. Tento systém komunikuje s jakýmkoliv MIDI softwarem nebo hardwarem.

Využívá celé spektrum MIDI zpráv, mezi které patří jak příkazy Note On, Note Off, tak také příkazy Program Change, Poly pressure nebo zprávy typu System Exclusive. Pro

každý stmívač, případně sadu stmívačů, je využit jeden MIDI kanál. Prostřednictvím jednoho MIDI kabelu tak může být ovládáno až 15 stmívačů, resp. 15 sad stmívačů. Šestnáctý kanál je využit pro řídicí informace celého systému. Zařízení dokáže komunikovat s jakýmkoliv MIDI hardwarem nebo softwarem. V rámci využití tohoto zařízení je také specifikace dimerů (stmívačů). Jejich maximální výkon by neměl překročit 600 watt na kanál. Doporučen je poloviční výkon, tedy 300 watt. Počet stmívačů v jedné sadě by neměl přesáhnout počet 90 [9].

## 1.8 Shrnutí

V této kapitole byly v rámci rešerše vybrány moduly pro ovládání světel pomocí protokolu MIDI, které ve své podstatě představují vždy jednu z možných cest, jak řešit problém výsledného osvětlení, které bude v úzké součinnosti spolu s hudbou. Snahou této kapitoly bylo alespoň částečně zmapovat dostupná řešení. Jejich výhody a nevýhody jsou součástí dalšího oddílu této práce.

## 2 VÝHODY A NEVÝHODY STÁVAJÍCÍCH ŘEŠENÍ

### 2.1 Tovární, komerčně dostupná řešení

Mezi výhody komerčně dostupných zařízení patří především jejich otestovaný provoz a také, v rámci možností, dostupný servis v případě poruchy zařízení.

Mezi nevýhody lze pak zařadit DMX512 protokol, který prodražuje světelná zařízení a snižuje jejich dostupnost pro široké masy uživatelů, kteří by rádi obohatili svou produkci i o vizuální efekty. Další nevýhodou související s protokolem DMX512 je nutnost vybavit mateřské zařízení modulem, který dokáže transformovat audio signál, případně MIDI povely do příkazů protokolu DMX512. To samozřejmě v případě, kdy toto neovládá námi zakoupené světelné zařízení. Tato funkčnost se bohužel také odráží v ceně zařízení.

Tomuto se z továrních zařízení vymyká pouze zařízení od polské společnosti Darys a některá ze zařízení americké společnosti MidiLite. U společnosti Darys se jedná jmenovitě o model ML-80P2, který převádí hodnoty dané MIDI zprávou na napěťové úrovni. Jelikož využívá z MIDI zpráv i zprávy typu Control Change, je možné nastavovat i intenzitu svícení. Koncepce tohoto zařízení se nejvíce podobá koncepci řešení, které je využito v rámci této diplomové práce.

### 2.2 Další řešení

Z nekomerčních projektů patří mezi nejzdařilejší sestava zařízení navržená Harveyem Twymanem. Jedinou nevýhodou je pouhých 300 Watt na kanál, které nejsou zcela ideální pro současné osvětlení. Lepší by bylo zařízení, které má výkonové stupně někde úroveň jednoho kilowattu na kanál. Řešení Harveye Twymana bylo jednou z inspirací pro vytvoření tohoto projektu. Mezi zdařilé součásti lze zařadit využití procesoru HC11 a také koordinaci s prostředím Cubase. Mezi nevýhody patří to, že právě takto vypracované řešení není komerčně dostupné. Lze říci, že pokud by se tohoto postupu ujala např. firma Steinberg, která vytváří prostředí Cubase, mohlo by se jednat o velmi zdařilý produkt, který by možná i rozšířil schopnosti hudebního programového prostředí Cubase.



### 3 NÁVRH VLASTNÍHO ŘEŠENÍ A JEHO ZDŮVODNĚNÍ

Řešení, které bylo zvoleno pro realizaci této diplomové práce vychází ze zařízení, která byla k dispozici. Tedy především vybavení laboratoře D 314 a vybavení laboratoře D 303. Výchozím hardwarovým zařízením je tedy sestava popsaná a také zprovozněná v diplomové práci Petra Lukačoviče [8] v laboratoři D 303 a sestava navržená a zprovozněná Pavlem Křeménkem [18] v laboratoři D 314.

Cílem bylo navrhnout co nejjednodušší a finančně nejméně náročné zařízení. Tedy modul, který je jednoduše programovatelný a samozřejmě vysoce variabilní. Variabilní tak, aby jej bylo možné bez velkých zásahů použít i u jiných světelných sestav. Při návrhu byla brána v potaz i úvaha, aby dané zařízení bylo využíváno při živé hudební produkci. Tedy bylo nutno řešit i zpoždění při provádění požadovaných operací převodu MIDI signálu na USB protokol, který pak byl dále upravován pro potřeby pulzně-šířkové modulace. Zjednodušeně řečeno, aby vizuální efekty úzce korespondovaly s hudební produkcí. Od počátku bylo zřejmé, že se bude jednat o jednočipovou sestavu, která bude mít jak malé napájecí požadavky, tak minimální nároky na údržbu a prostor, tak samozřejmě co možná nejnižší stupeň poruchovosti. Tedy zařízení s minimem hardwarových komponent.

#### 3.1 Volba hardwarového rozhraní

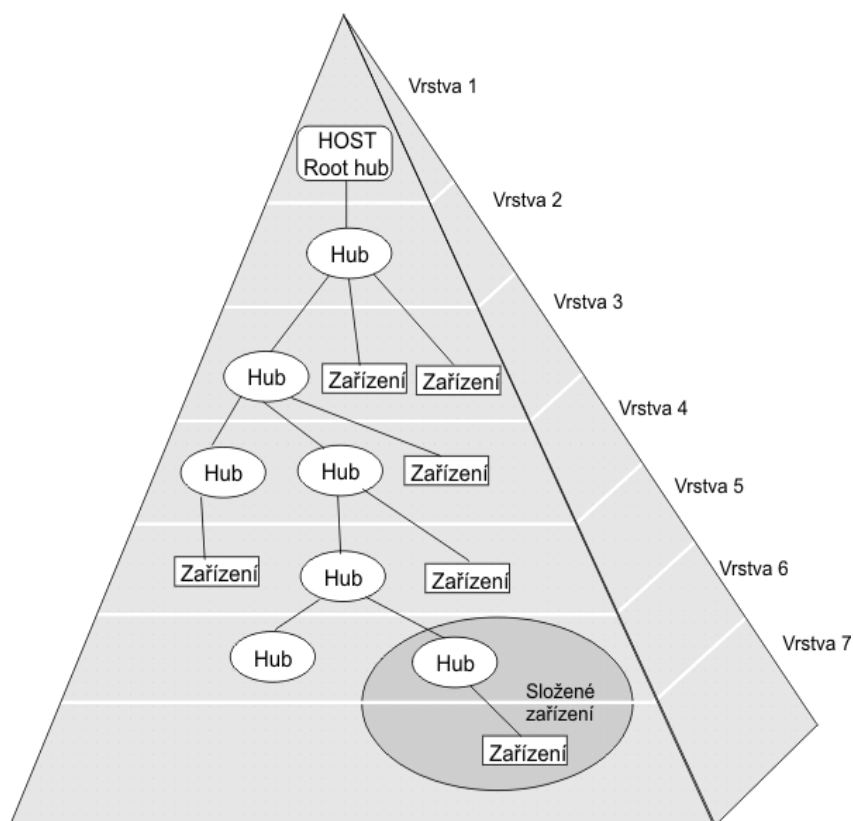
Hned od počátku vývoje ovládání pro danou světelnou soustavu bylo hlavním úkolem vytvořit co možná nejuniverzálnější zařízení. Bylo tedy zřejmé, že bude třeba využít řešení, které bude z části nebo úplně využívat USB protokol tak, aby bylo možné se připojovat z jakéhokoliv počítače. Řešením by mohlo být i využití Gameportu nebo sériového portu. Gameport je poněkud zastaralá technologie a často se již tento jack nevyskytuje na počítačových skříních. Sériový port je na tom podobně, navíc se s úspěchem imituje právě pomocí USB. Podmínkou je instalovaný MIDI software, který má potom přístup přes toto zařízení k výkonovým stupňům světelných ovladačů. Ve finále zvítězilo řešení společnosti Microchip, konkrétně jejich rozhraní s modelovým označením 18F2550. Na základě tohoto rozhodnutí musela být také specifikována norma pro přenos dat pomocí USB.

## 4 POPIS POUŽITÝCH TECHNOLOGIÍ A SYSTÉMOVÉHO VYBAVENÍ

### 4.1 USB standard

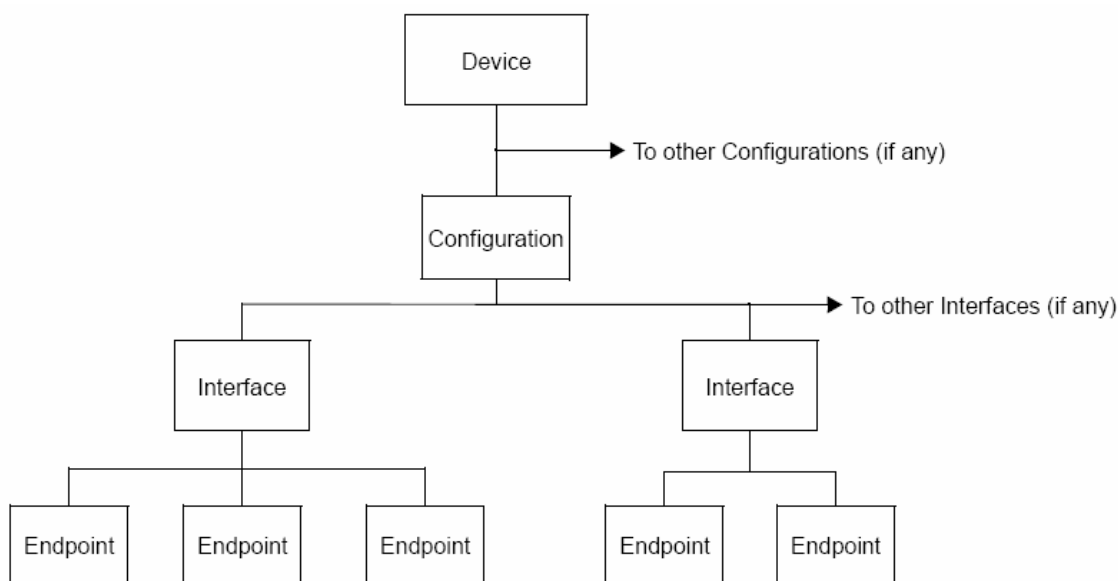
Základem pro tuto specifikaci je samozřejmě obecná norma USB verze 2.0, na kterou navazuje norma pro USB Audio zařízení a z ní po té vychází samotná USB MIDI norma. Tato je v současnosti hojně využívána, protože přenos MIDI informací přes USB je efektivnější, než přenos MIDI informací pomocí klasických DIN jacků. DIN zabírají příliš mnoho místa jak na skříních PC, tak na externích zvukových kartách, které se využívají nejčastěji v kombinaci s notebooky. Umístění DIN konektorů na notebooky je kvůli velikosti zcela vyloučeno.

USB (Universal Serial Bus) má hvězdicovou charakteristiku o maximálně sedmi úrovních. Základem komunikace je Root USB hub, který vyzývá všechny kolem sebe ke komunikaci. Může být připojeno max. 127 zařízení v sedmi úrovních. Na Root USB Hub můžou být připojeny další huby anebo samotná zařízení, která se ovšem v anglosaské terminologii nazývají *functions* [10].



Obr. 8. Struktura USB stromu

V rámci USB propojení jsou definovány tzv. endpoints – koncové body, které mají v každém USB zařízení svou funkčnost. Celkový maximální počet endpointů je 15. Endpoint zero – výchozí koncový bod je pro defaultní rouru (propojovací kanál klienta a USB hosta), která se propojuje, když se dané USB zařízení identifikuje do sestavy. Jsou vysílány požadavky IRP (I/O Request Packet), které se provádějí v sériovém pořadí. Pokud dojde k chybě při provádění nějakého IRP, ostatní IRP jsou odstraněny z fronty a dochází potom k odstranění chyby. A znovunačtení IRP požadavků jednotlivých zařízení USB. Roury jsou dvojího druhu. Zprávy a Proudů.



Obr. 9. Vrstvení jednotlivých endpointů v rámci struktury USB

Obr. č. 9 znázorňuje obecnou hierarchickou strukturu sestavy endpointů, pomocí které je možno definovat obecné USB zařízení. Dále je možno pomocí tohoto schématu definovat také příslušné deskriptory (popisovače, neboli struktury) endpointů daného zařízení:

1. **Device descriptor** – deskriptor první úrovně – je programově jakousi samostatnou součástí, která slouží v průběhu enumerace (zavádění zařízení do systému) k identifikaci zařízení.
2. **Configuration descriptor** – struktura, která popisuje zařízení do nejnižší možné identifikační úrovně v rámci endpointů. V této struktuře jsou obsaženy všechny ostatní podřízené struktury (deskriptory endpointů) a určují tak veškerou funkčnost daného USB zařízení.
3. **Interface descriptor** – Interface deskriptor nemusí být vždy jen jeden.

4. **Deskriptory endpointů** – těchto deskriptorů je vždy několik. Jejich počet vychází z příslušné funkčnosti USB zařízení. Jejich pojmenování je přímo závislé na druhu a typu daného USB zařízení.

Veškerá komunikace na sběrnici je časově multiplexována do tzv. rámců (frames) dlouhých 1 milisekundu. Každý rámeček může obsahovat mnoho transakcí různých zařízení a různých endpointů. Transfery dat po sběrnici se dají rozdělit na 4 základní typy:

1. **Izochronní transfery** - Transport velkého množství dat (až 1023 B) - je zajištěna doba dodání, ale není zajištěna integrita dat.
2. **Bulk transfery** – Transport velkého množství dat se zajištěnou integritou, avšak doba doručení zajištěna není.
3. **Interrupt transfery** – Pro malý objem dat, zajištěna integrita i včasné doručení.
4. **Control transfery** – Použit při úvodním setupu zařízení (enumerace).

Napájení připojených zařízení je možné ze samotné sběrnice. USB specifikace přesně stanovuje maximální možný odběr proudu zařízením na 100mA. Energeticky náročnější zařízení musí používat externí zdroj napájení [11].

Ve chvíli, kdy je zařízení připojeno ke sběrnici, USB host zahájí proces zvaný enumerace. Během tohoto procesu host identifikuje zařízení, získá potřebné konfigurační informace vyžádáním deskriptoru zařízení, konfigurace, rozhraní a endpointů a nakonec nastaví požadovanou konfiguraci zařízení. Průběh enumerace může vypadat například takto:

1. USB reset,
2. vyžádání deskriptoru zařízení – pouze základní část identifikace zařízení,
3. USB reset,
4. nastavení unikátní adresy zařízení,
5. vyžádání deskriptoru zařízení – tentokrát celý, včetně všech základních deskriptorů,
6. vyžádání deskriptoru konfigurací,
7. vyžádání dalších deskriptorů,
8. nastavení požadované konfigurace.

Z pohledu zařízení definuje USB specifikace konkrétní stavy, ve kterých se zařízení během enumerace může nacházet, a povolené přechody mezi nimi (DETACHED \_ATTACHED \_

POWERED \_ DEFAULT \_ ADDRESS\_PENDING \_ ADDRESSED\_ CONFIGURED \_ READY) [10].

Zařízení připojená ke sběrnici mohou být zcela specifická a vyžadující specifické ovladače, nebo mohou patřit do třídy USB zařízení (HID, MSD, CDC). Tyto třídy definují některé vlastnosti a chování zařízení, takže pro různá zařízení hlásící se ke stejné třídě, může systém použít stejný ovladač.

Zprávy jsou přesně specifikovány a mají předem danou velikost a určují požadavek zařízení, případně hosta. Roury zprávy dodržují určitou strukturu dat, která umožňuje identifikaci příkazů a požadavků. Roury zpráv jsou obousměrné a vždy používají dva endpointy se stejným číslem

Proudy jsou roury, které obsahují přenos samotných dat. Jsou jednosměrné a data v nich jsou přenášena za sebou, paketově. Pokud nedojde k naplnění celého paketu, není nutno jej doplňovat do plné velikosti [11].

## 4.2 USB standard pro Audio zařízení

Jak vyplývá ze zvolené tematiky, je USB svou přenosovou kapacitou zcela dostačující i pro přenos audio dat, potažmo MIDI informací. Audio zařízení mají v rámci protokolu USB svou vlastní specifikaci a odpovídající sadu deskriptorů popisujících požadované endpointy pro přenos audio dat. Ve většině případů jsou jednou ze specifikací daného zařízení, protože jsou vždy kombinovány se zařízeními jiného, než audio USB standardu. Mezi taková zařízení patří např. ovladač pro externí CD/DVD mechaniku, kdy je nutno popsat toto zařízení pomocí audio, video USB standardu a také jako velkokapacitní datové úložiště. Tedy v tomto případě, jako ve většině takových specifikací pro dané zařízení, je audio USB standard pouze jednou ze součástí daného zařízení. Poněkud jiná situace je v případě MIDI zařízení. MIDI USB standard je rozšířením Audio USB standardu. Popis audio zařízení vychází samozřejmě z příslušné normy [13].

## 4.3 USB standard pro MIDI zařízení

V prvopočátku je nutno říci, že typické USB MIDI zařízení patří do třídy CDC. Je tedy stejně jako audio zařízení považováno za komunikační rozhraní. USB popis třídy pro MIDI zařízení je jednou ze součástí USB Audio normy. Děje se tak, pokud je USB zařízení schopno přijímat, resp. odesílat MIDI zprávy. Jak již bylo zmíněno výše v kapitole 4.1, je

nutno specifikovat rozhraní na Interface úrovni daného zařízení. USB MIDI zařízení bude mít tedy dvě rozhraní (interface). Jedním rozhraním je audio, druhým pak MIDI. Toto je samozřejmě nutno popsat pomocí deskriptorů, tak, aby se zařízení bez problémů identifikovalo do systému a především bylo viditelné pro aplikace, které jsou schopny komunikovat přes MIDI protokol. V našem případě se jedná o aplikaci Cubase SX 2.

V rámci specifikace jsou deskriptory endpointů v případě MIDI zařízení označovány jako endpointy pro MIDI vstupní a výstupní jacky. Tyto jacky jsou dvojího druhu. Jedny jsou označovány jako External MIDI IN, resp. OUT jacky. Druhou skupinou jsou pak Embedded MIDI IN, resp. OUT jacky. Průchod MIDI dat od hostitele k MIDI zařízení a zpět je potom následující.

Informace jdoucí od hosta je adresována do zařízení pomocí External MIDI OUT jacku hosta, odtud pokračuje Embedded MIDI IN jackem daného zařízení, tím se dostane k USB zařízení a je zpracována. Takto zpracovaná informace je poslána zpět přes Embedded MIDI OUT jack daného zařízení zpět na External MIDI IN konektor hostitele. To celé je ale v podstatě pouze virtuální abstrakcí, kterou je nutno naprogramovat v rámci daného USB zařízení. Vše je pomocí deskriptorů očíslováno a spolu asociováno. Tedy příslušná položka daného deskriptoru, který konektor patří ke svému opačnému protějšku. Dále pak je také určena asociace propojení jednotlivých External MIDI IN (OUT) jacků s odpovídajícími Embedded MIDI OUT (IN) jacky. Vždy je propojen Embedded IN jack s External OUT jackem a obráceně. Stejně pak jednotlivé Piny MIDI jacků mají svá pořadová čísla a jsou také nastaveny spolu s číselnou informací, kolik je kterých pinů, a ke kterému zařízení jsou propojeny [12]. V rámci těchto norem se často jedná o multifunkční zařízení. Velmi často je takovým zařízením klávesový nástroj spadající do kategorie workstation – pracovní stanice. Daný klávesový nástroj může zároveň sloužit jako syntetizér, sampler, audio efektová jednotka a řídicí MIDI klávesnice v jednom. Potom je třeba v rámci řídicího softwaru specifikovat také pomocí deskriptorů i níže zmíněné logické úrovně.

#### 4.3.1 Element

Je jednou z určených logických úrovní, základní stavební prvek pro USB MIDI komunikaci. Jedná se o specifikaci zařízení, kterým může být například:

- syntezátor,
- externí MIDI časová synchronizace,

- efektné jednotky ovládané pomocí MIDI,
- atd.

Element je obvykle připojen k jednomu nebo více Embedded nebo External MIDI konektorům. Pokud se jedná o syntetizér, obvykle vytváří výstupní audio data vytvořená pomocí MIDI informace, které jsou posílány přes USB MIDI rozhraní. Audio je po té přenášeno přes audio zařízení skrz vstupní terminál syntetizéru [12].

#### 4.3.2 Vstupní a výstupní terminál

Vstupní terminál reprezentuje výstup daného zařízení (např. syntezátoru), vstupní terminál je připojen k zařízení a přes něj se dostává USB MIDI přenos do daného zařízení k jeho konfiguraci. Komunikace mezi MIDI zařízením a audio zařízením probíhá pomocí vstupního a výstupního terminálu. Tento je logicky vytvořen pomocí příslušných deskriptorů sloužících především pro přenos informací důležitých pro správné zpracování audia. Mezi takové údaje patří např. přenos nastavení efektných jednotek nebo údaje ekvalizérů a v neposlední řadě také např. nastavení hlasitosti pro výstupní signál [12].

#### 4.3.3 USB-MIDI Converter

USB-MIDI converter je srdcem každého MIDI zařízení, protože poskytuje propojení mezi hostem a USB-MIDI zařízením. Je to základní stavební kámen. Na straně jedné je to rozhraní s USB rourami, které jsou využívány k výměně MIDI dat mezi hostem a USB-MIDI endpointy daného zařízení. Na straně druhé pak je prezentován příslušným počtem Embedded MIDI jacků. Tyto Embedded jacky jsou logickým rozhraním prezentujícím skutečnou konektivitu uvnitř MIDI zařízení. USB-MIDI konverter poskytuje spojení mezi MIDI OUT endpointem a příslušným Embedded MIDI IN jackem. Podobně pak poskytuje spojení mezi Embedded MIDI OUT jackem a odpovídajícím MIDI IN endpointem [12].

#### 4.3.4 Popis deskriptorů USB MIDI zařízení

V návaznosti na předchozí text a dané normy bude mít pak typické MIDI zařízení komunikující pomocí USB tyto deskriptory:

##### 1. Device Descriptor

Jeho položky odpovídají standardnímu zařízení CDC třídy



## 2. Configuration Descriptor

Stejně jako device deskriptor

## 3. Standard AC Interface Descriptor

AudioControl rozhraní nemá žádný svůj endpoint. Používá default endpoint zero pro komunikaci. Class-specific AudioControl požadavky jsou posílány právě pomocí default kanálu. Neposkytuje žádné endpointy pro nastavení přerušení USB zařízení.

## 4. Class-specific AC Interface Descriptor

Je vždy spojen se Standard (header) deskriptorem, který obsahuje základní informace o Audio rozhraní. Obsahuje všechny pointery potřebné pro popsání sestavy audio rozhraní ve spojitosti s daným audio zařízením.

## 5. Standard MIDI Streaming Interface Descriptor

Standard interface descriptor charakterizuje zařízení jako takové. Pomocí tohoto deskriptoru je specifikována vnitřní struktura USB MIDI zařízení a jeho další podrobný popis je obsažen v deskriptorech, které jsou součástí této konfigurační struktury.

## 6. Class-specific MIDI Streaming Interface Header Descriptor

poskytuje další (upřesňující) informace vztahující se k vnitřní struktuře daného zařízení.

## 7. MIDI IN Jack Descriptor

Popisuje MIDI IN jacky, ať už se jedná o Embedded nebo o External jacky. Tento parametr je nastaven v hodnotě bJackType.

## 8. MIDI OUT Jack Descriptor

Popisuje MIDI OUT jacky, stejně jako MIDI IN deskriptor. V jeho struktuře jsou přidány další položky, které jsou nutné pro přesnou specifikaci propojení s odpovídajícím External, resp. Embedded MIDI IN deskriptorem. Tyto další položky specifikují především jednotlivé piny daného MIDI OUT konektoru a jeho stavu v případě přenosu dat.

## 9. Element Descriptor

Rozšiřuje strukturu MIDI OUT deskriptoru o položku vstupního a výstupního terminálu a ještě o nastavení případných dalších schopností USB MIDI zařízení.

Položka této struktury `bInTerminalLink` obsahuje identifikátor vstupního terminálu, ke kterému je připojen daný Element. Pokud není připojen žádný vstupní terminál, musí být toto pole nastaveno na nulu.

Položka `bOutTerminalLink` obsahuje identifikátor výstupního terminálu, ke kterému je připojen tento Element prezentující např. sampler nebo další zařízení ovládané pomocí MIDI. Pokud není použit výstupní terminál, musí být tato položka nastavena na nulu.

Položka `bElCapsSize` obsahuje bytovou velikost položky `bmElementCaps`. V současnosti je jeho velikost 1. Využití této položky bude sloužit pro budoucí další rozšíření.

Každý bit položky `bmElementCaps` reprezentuje nějakou schopnost, resp. Funkčnost USB-MIDI Elementu. Pokud jsou jednotlivé bity nastaveny, jsou podporovány příslušné schopnosti USB-MIDI zařízení. Alespoň jeden bit by měl být nastaven, vyjma bitu D7 [12].

## 10. Standard MIDI Streaming Bulk Data Endpoint Descriptor

Obsah tohoto deskriptoru se shoduje s obsahem Standard endpoint deskriptoru, který je popsán v kapitole 9.6.4 USB specifikace [10] a dále pak vysvětlen v USB Class specifikaci [14]. Parametr D7 z pole `bEndpointAddress` indikuje, jestli je endpoint vstupní nebo výstupní. Pole `bmAttributes` pak velikost endpointu. Synchronizace je potom indikována hodnotami bitů D3..2 a musí být nastavena na nulu.

## 11. Class-Specific MS Bulk Data Endpoint Descriptor

Pole `bNumEmbMIDIJack` obsahuje počet Embedded MIDI Jacků, spojených s tímto endpointem. V případě, že se jedná o vstupní endpoint, pak embedded jack by měl být MIDI OUT. Pokud se jedná o výstupní endpoint, mělo by se jednat o embedded MIDI IN jack. Pole `baAssocJacks` pole obsahuje potom ID těchto embedded jacků.

## 12. Standard MS Transfer Bulk Data Endpoint Descriptor

Tento deskriptor se také shoduje s popisem deskriptoru z kapitoly 9.6.4. USB specifikace, tedy se standardním Endpoint deskriptorem. Pole bEndpointAddress určuje pomocí parametru D7, jestli se jedná o vstupní transfer endpoint, nebo se jedná o výstupní transfer endpoint. Pole bmAttributes nastavuje parametr bulk typu daného endpointu. Typ synchronizace je indikován pomocí hodnot bitů D3..2 a musí být nastaven jako prázdný parametr [12].

## 13. Class-Specific MS Transfer Bulk Data Endpoint Descriptor

Ve specifikaci tohoto zařízení se nevyskytuje descriptor tohoto typu.

## 4.4 MIDI protokol

Základem MIDI komunikace je tzv. MIDI zpráva, která sestává ze tří bajtů. Každá MIDI událost (zpráva) je proto prezentována jako kombinace osmimístného čísla, které je tvořeno nulami a jedničkami. Každá MIDI zpráva může tedy obsahovat hodnotu od 0 do 255, tedy celkem 256 různých hodnot. Zprávy MIDI jsou rozděleny do dvou základních kategorií: zprávy Status a Data. Zpráva Status slouží k určení druhu informace, která je skrze MIDI posílána. Oznamuje zařízení, které zprávu přijímá, která událost náleží kterému kanálu MIDI a o jakou událost se jedná. Může jít např. o události: Note On, Pitch bend (změna výšky tónu), Program Change (změna patche) nebo aftertouch (poslední událost nastává v okamžiku, kdy je vyvinut další tlak na již stlačenou notu). Datové bajty obsažené ve zprávě zase zařízení informují o tom, jaké hodnoty jsou přiřazeny událostem, které v sobě nese Status Byte [21].

## 4.5 USB MIDI Event Packet

MIDI data jsou přes USB přenášena pomocí 32 – bitových MIDI Event paketů. Přenos dat je tak prováděn pomocí zpráv o standardní délce čtyři bajty. Pomocí těchto USB MIDI event paketů je vytvořeno virtuální propojení mezi endpointy USB hosta a USB MIDI zařízení. Tento způsob propojení je výhodný pro svou nenáročnost, kdy nepožaduje vysoký počet endpointů, jako jiné druhy USB zařízení. Každá MIDI událost má svůj vlastní USB MIDI packet, což omezuje vznik mnoha chyb.

Byte 0		Byte 1	Byte 2	Byte 3
CN – cable number Číslo kabelu	CIN – Code Index Number klasifikace jednotlivých druhů MIDI zpráv	MIDI zpráva první bajt	MIDI zpráva druhý bajt	MIDI zpráva třetí bajt

Tab. 2. Struktura USB MIDI Event paketu

První čtyři bajty začínající na MSB pozici obsahují informaci o čísle virtuálního MIDI kabelu, pro kterém má jít daná MIDI informace. Hodnota CN je údaj v rozmezí 0x0 až 0xF indikující číslo požadovaného Embedded jacku, přes který dochází ke spojení s příslušnou MIDI funkcí. Druhý nibl končící LSB obsahuje potom identifikační údaje o MIDI zprávě. Jednotlivá identifikační čísla a jejich odpovídající MIDI zprávy lze vidět v tabulce č. 2 [12].

CIN	MIDI_x Size	Popis
0x0	1, 2 or 3	Neobsazená hodnota, rezervována pro další rozšíření.
0x1	1, 2 or 3	Neobsazená hodnota, rezervována pro další rozšíření.
0x2	2	Dvoubajtové MIDI zprávy. Např. MIDI Time Clock nebo SongSelect
0x3	3	Třibajtové MIDI zprávy. Např. SPP
0x4	3	MIDI zprávy System Exclusive typu spustit nebo pokračovat.

0x5	1	Jednobajtové MIDI Systém Common zprávy nebo SysEx zprávy s příkazem end a následnou jednobajtovou informací.
0x6	2	SysEx zprávy s následnou dvojbajtovou informací.
0x7	3	SysEx zprávy s následnou trojbajtovou informací.
0x8	3	Nota vypnuta
0x9	3	Nota zapnuta
0xA	3	Společná tlaková citlivost
0xB	3	MIDI Control Change
0xC	2	Změna programu
0xD	2	Tlaková citlivost
0xE	3	Změna ladění
0xF	1	Single Byte

*Tab. 3. Code Index Number a odpovídající MIDI zprávy*

Druhý, třetí a čtvrtý bajt obsahují strukturu klasické MIDI zprávy tak, jak je zmíněno v předchozí kapitole. Neobsazené bajty musí být doplněny nulami do plné velikosti paketu.

V následující tabulce je patrné, jak se liší bajtový zápis MIDI zprávy od USB MIDI paketu, který je nutno poslat pokud se bude jednat o komunikaci a zasílání MIDI informací pomocí protokolu USB.

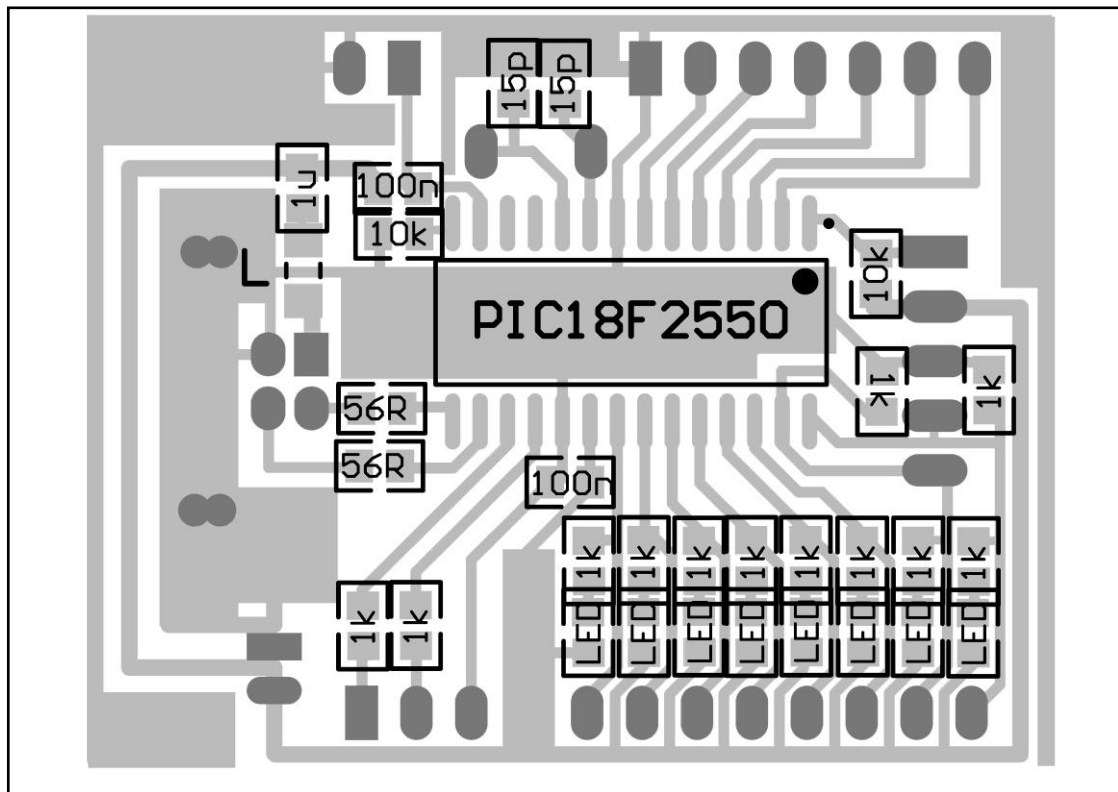
Popis MIDI zprávy	Bajtové vyjádření	USB MIDI paket
MIDI zpráva Nota On na virtuálním kabelu č. 1 (CN=0x1; CIN=0x9)	9n kk vv	19 9n kk vv
MIDI zpráva Control change skrze kabel č. 10 (CN=0xA; CIN=0xB)	Bn pp vv	AB Bn pp vv
System Exclusive Real-time zpráva F8 na kabelu č.3 (CN=0x3; CIN=0xF)	F8 xx xx	3F F8 xx xx
System Exclusive zpráva na kabelu p (CN=0xp). Označení počátku SysEx: CIN=0x4. Ukončení SysEx: CIN=0x5	F0 00 01 F7	p4 F0 00 01 p5 F7 00 00
System Exclusive zpráva prostřednictvím kabelu p (CN=0xp).Start SysEx: CIN=0x4. Konec SysEx: CIN=0x6	F0 00 01 02 F7	p4 F0 00 01 p6 02 F7 00
SysEx zpráva prostřednictvím kabelu p (CN=0xp).Start SysEx: CIN=0x4. Konec SysEx: CIN=0x7	F0 00 01 02 03 F7	p4 F0 00 01 p7 02 03 F7
Speciální případ: dvojbajtová SysEx Zpráva prostřednictvím kabelu p (CN=0xp; CIN=0x6)	F0 F7	p6 F0 F7 00
Speciální případ: trojbajtová zpráva SysEx prostřednictvím kabelu p(CN=0xp;CIN=0x7)	F0 mm F7	p7 F0 mm F7

Tab. 4. Popis MIDI zprávy a její bajtové vyjádření pomocí USB MIDI paketu

## 4.6 Hardware

Vzhledem k rozsahu práce není možno se zde zabývat popisem světelné sestavy a sestavy čerpadel, kterou vytvořil Petr Lukačovič. Podrobný popis tohoto systému je v jeho diplomové práci, která nese název Počítačové řízení fontány a je součástí podkladové literatury této práce [3]. Předmětem této kapitoly je pouze hardwarová sestava, která nese název USB MIDI Lights Device. Toto zařízení sestává z CPU, kterou je v tomto případě PIC 18F2550 od společnosti Microchip, 8 LED, krystal a nutné stabilizační kondenzátory

vyrovnávací odpory. Nedílnou součástí je také USB konektor typu B, sloužící k propojení s PC.



Obr. 10. Schéma jednotlivých součástí USB MIDI Lights Device

#### 4.6.1 Microchip PIC 18F2550

Srdcem celého modulu je mikrokontrolér americké společnosti Microchip s typovým označením 18F2550. Jedná se o jednočip, který je zcela kompatibilní s USB normou verze 2.0. Podporuje také oba druhy USB přenosu, jak Low Speed (1,5Mbit/s), tak Full Speed o rychlosti 12 Mbit/s. Umožňuje také veškeré druhy přenosů pomocí USB, tak aby byla zajištěna veškerá možná dostupná funkčnost v rámci USB protokolu. Procesor podporuje plný počet obousměrných endpointů (16).

V tzv. Run módu je kontrolér tehdy, pokud běží procesor i periferie. V Idle módu běží pouze periferie. Sleep mód je nastaven, pokud neběží ani procesor ani periferie. K zařízení je možno připojit dva externí oscilátory o max. frekvenci 48 MHz. Kontrolér má samozřejmě i svůj interní oscilátor. Uživatel si může vybrat celkem z osmi oscilačních kmitočtů z rozmezí 31 kHz až 8 MHz.

Na zařízení je také možno měnit polaritu, což se osvědčilo při vytváření výsledného projektu. První zařízení v sérii mělo totiž spínání LED diod v logické 1 a tím pádem potom docházelo k převrácení hodnot snímaných z parametru Velocity, který je obsahem třetího bajtu MIDI zprávy. Blíže bude upřesněno v dalších kapitolách. Processor má 100 000x přepisovatelnou paměť pro program a 1 000 000x přepisovatelnou paměť EEPROM. Jeho součástí je také 32 úrovnový stack a instrukční set, který obsahuje 105 systémových instrukcí. Funkce Code Protection zamezuje zápisu do vybraných oblastí paměti programu. Samozřejmostí je také podpora ICSP programování, která umožňuje programovat procesor zasazený v desce plošných spojů. Dalšími součástmi procesoru PIC18F2550 jsou 16 – bitový komparátor, SPI (Serial Peripheral Interface) a v neposlední řadě také EUSART. Tento sériový komunikační modul, vycházející ze standardu RS-232, poskytuje podporu pro protokol na sběrnici standardu LIN. Součástí EUSART je také automatická detekce přenosové rychlosti a 16 bitový generátor přenosové rychlosti. Pokud je v mikrokontroléru využíváno vnitřního oscilačního bloku, EUSART je využit na místě komunikace, kdy ošetřuje přístup na nevyužitý externí oscilátor a zamezuje tak chybám při požadavcích na přerušení [15].

#### 4.6.2 Zařízení USB MIDI Lights Device jako celek

Deska USB MIDI Lights Device byla navržena jako univerzální deska pro vývoj aplikací s mikrokontrolerem PIC18F2550 s důrazem na využití USB rozhraní toho mikrokontroleru.

Napájení desky je prováděno prostřednictvím USB portu, který poskytuje stabilizované napětí 5V. Pro filtraci tohoto napětí je přidána indukčnost (feritové jádérko) a elektrolytický kondenzátor 100 $\mu$ F. Samotný mikrokontroler dále obsahuje stabilizátor 3.3V, který pro stabilizování výstupu používá keramický kondenzátor 1 $\mu$ F. S napájením také souvisí blokovací kondenzátor 100nf umístěný mezi GND (uzemnění) a Vcc piny mikrokontroleru.

Zdrojem pro generování hodinového signálu je XTAL 20MHz s dvojicí keramických kondenzátorů 15pF. Hodnota tohoto krystalu byla vybrána z důvodu jeho dostupnosti, samotný mikrokontroler umožňuje použití krystalu v hodnotách (4, 8, 12, 16, 20, 24, 40, 48 MHz). Mikrokontroler je vybaven i vnitřním RC oscilátorem, ale pro zapojení využívající USB port je nutné použít přesný zdroj hodinového kmitočtu - krystal.

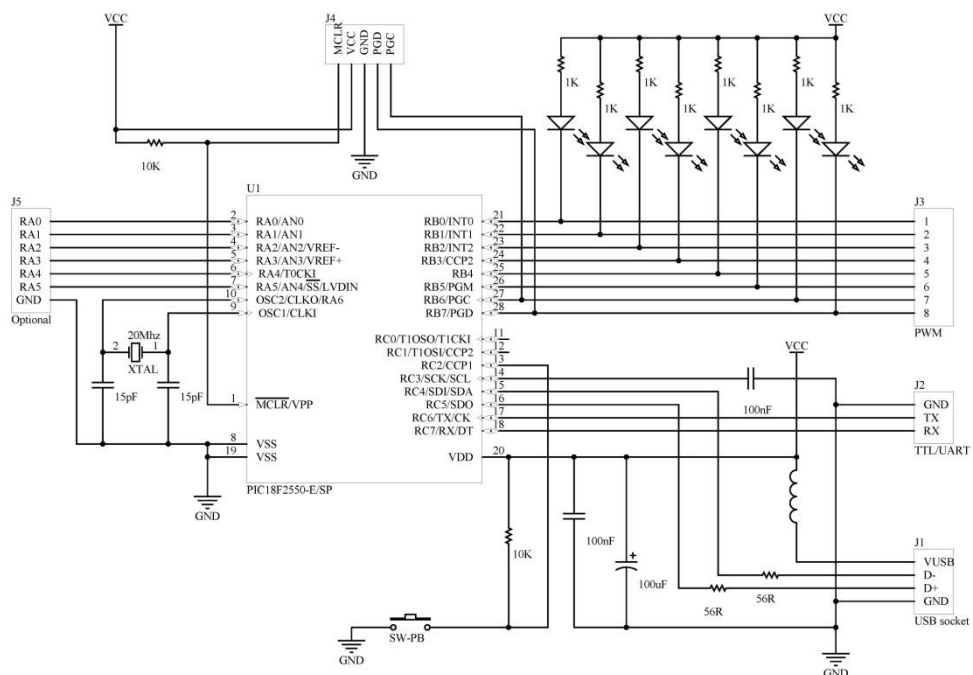


Na konektory J2, J3 a J5 jsou přivedeny vstupně/výstupní piny mikrokontroleru. Na konektor J2 jsou přivedeny piny RC6 a RC7 – sériový port mikrokontroleru a uzemnění. Na konektor J3 jsou připojeny piny brány B (RB0 – RB7), ke kterým jsou paralelně proti 5V připojeny LED se sériovým odporem.

Konektor J5 obsahuje piny brány A (RA0-RA5) a GND. Další konektor na desce je zkratovací propojka ve schématu označená SW-PB, která slouží pro aktivaci bootloaderu, v normálním módu je rozpojená v módu bootloaderu zkratovaná.

Základní program desky zavaděč – bootloader byl do desky naprogramován pomocí externího programátoru připojeného na konektor J4, který je na desce reprezentován ploškami pro připájení vodičů programátoru.

Reset mikrokontroleru je připojen pomocí odporu  $10k\Omega$  k napájecímu napětí. Jak je patrné ze schématu, zařízení má celkem osm funkčních výstupů (kanálů).

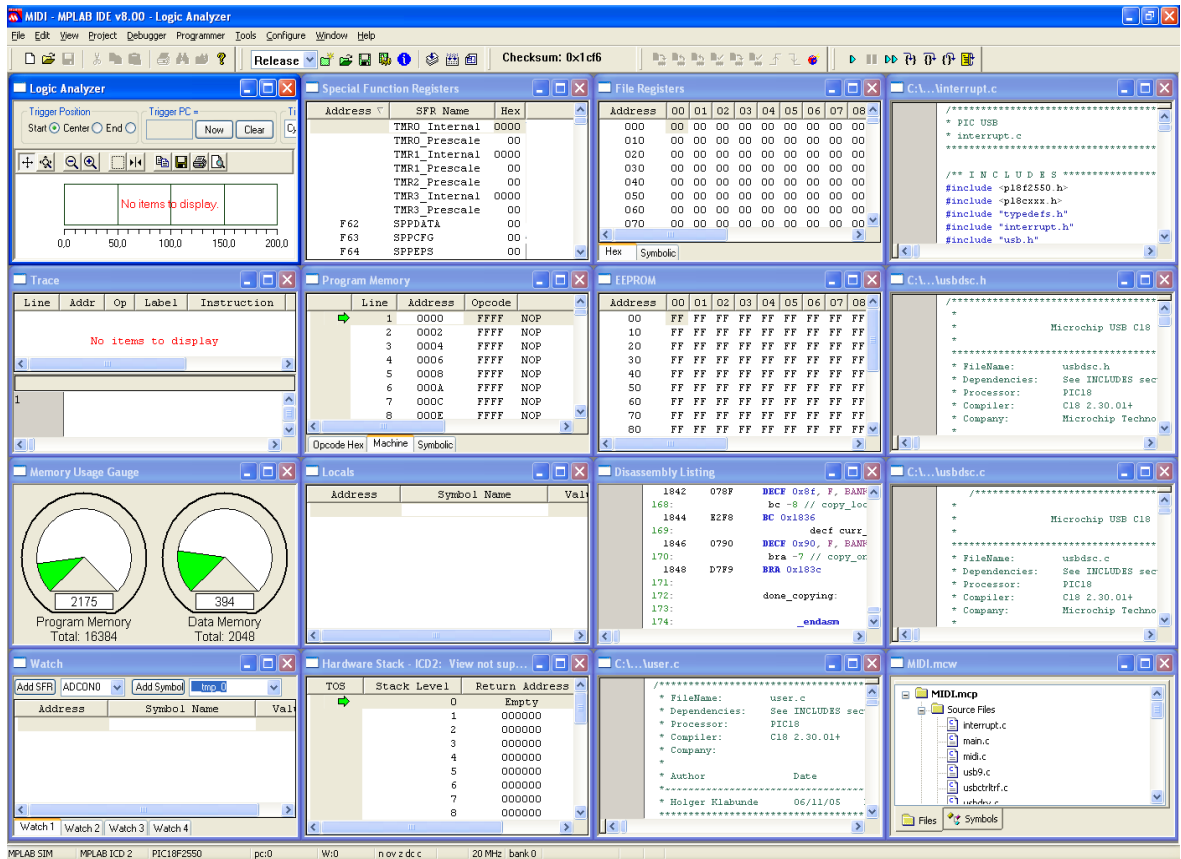


Obr. 11. Schéma zapojení USB MIDI Lights Device

## 4.7 Vývojové nástroje a použitý firmware

### 4.7.1 MPLAB IDE

K úpravám a vývoji zdrojového kódu pro USB MIDI Lights Device bylo využito vývojové prostředí, které společnosti Microchip dodává a také poskytuje jako shareware ke svým vývojovým deskám. Jedná se o IDE, které ve svém základu poskytuje podobné vývojové možnosti jako Visual Studio od Microsoftu, s tím rozdílem, že je zaměřeno na vývoj ovladačů a ovládacího software pro jednočipové aplikace. Samozřejmostí je debugger a možnost breakpointů, dále potom také zvýrazňování klíčových slov. Velkou výhodou je pak možnost zobrazení hodnot klíčových registrů, naplnění zásobníku, programové paměti, obsah jednotlivých proměnných, hodnot paměti EEPROM a také možnost trasování a monitoring datového průběhu při ladění programu. Poslední možnost je ale dostupná pouze s příslušným hardwarovým vybavením, které je také k dostání u společnosti Microchip. Prostředí je možno využívat jak při programování v Assembleru, tak při programování v C jazyce. Pro psaní zdrojových souborů v Assembleru je integrován nástroj MPASM. V rámci licenční politiky je toto prostředí zdarma k využití jako Student Edition po dobu šedesáti dní. Většinu vývojových a monitorovacích oken je možno vidět na obrázku č. 11 na následující straně. Součástí tohoto prostředí je i kompilátor MCC 18 a linker MPLINK. Oba jsou z produkce výše jmenované společnosti Microchip [11].



Obr. 12. Vývojové prostředí MPLAB IDE od společnosti Microchip

#### 4.7.2 MCC C18

Kompilační součást prostředí MPLAB IDE. Přesný obchodní název zní MPLAB® C18 C COMPILER. Tento kompilátor vychází z normy ANSI C '89. Tato norma není tímto kompilátorem zcela striktně dodržována z důvodu možných kolizí s optimalizačními procedurami pro mikrokontrolery společnosti Microchip. Jedná se o konzolovou aplikaci pro platformu Windows. Díky tomuto kompilátoru je také možno kombinovat části kódu psané v Assembleru a používat v souborech specifických pro C jazyk, direktivy psané v assembleru a takto kombinovaný kód bez problémů kompilovat. Použití assembleru je vhodné v případě obsluhy přerušení. Takové volání se podobá této direktivě:

```
_asm goto high_isr _endasm
```

Pomocí této direktivy je vykonávání programu přerušeno a je proveden sled instrukcí uvedený na adrese pro přerušování s vysokou prioritou.

Dále také snadňuje přístup do externí paměti pro čtení a zápis. Umožňuje uživateli plnou kontrolu nad přístupem do paměti tak, aby bylo ihned zřejmé, do kterých míst

paměti jsou jednotlivé proměnné adresovány a jaký je jejich způsob zpracování [16]. Pro tento kompilátor existuje ucelená sada knihoven, které jsou popsány v dokumentu [17].

#### 4.7.3 Firmware – moduly a soubory

Software, který byl využit k ovládní USB MIDI Lights Device, vychází z ukázkového příkladu CDC RS – 232, tedy softwaru, který přes USB emuluje sériový port. Tento software byl modifikován a upraven tak, aby mohl obsluhovat osvětlení na základě obdržných MIDI příkazů. Základními a nepostradatelnými soubory jsou:

- **main.c** - Funkce main() obsahuje nekonečnou smyčku while(1) programu. V této smyčce jsou obslouženy prostřednictvím procedur USBTasks(void) a void ProcessIO(void) všechny požadované úkoly v programovém sledu, který je dán instrukcemi zdrojového kódu. Předtím je samozřejmě systém inicializován pomocí procedury InitializeSystem().
- **typedefs.h** - V tomto souboru jsou definovány jednotlivé datové typy.
- **interrupt.c a .h** - Tento modul obsahuje implementaci obsluh přerušení vysoké i nízké priority. Je zde definováno makro mEnableInterrupt(), které aktivuje propagaci USB přerušení do hlavního přerušovacího systému mikrokontroléru. Vlastní obsluha přerušení je pak implementována ve funkcích void low\_isr(void) a void high\_isr(void).
- **usb.h** – Tento soubor zajišťuje vkládání potřebných hlavičkových souborů do celého programu. Je zde třeba poznamenat, že výchozí nastavení není zcela kompletní a je nutno je doplnit podle potřeb jednotlivých zařízení a jejich funkčností.
- **usbdefs\_ep0\_buff.h** – V souboru je obsažena definice struktury CTRL\_TRF\_SETUP a datapaketu pro odpověď CTRL\_TR\_DATA pro Control transfer.
- **usbdefs\_std\_dsc.h** – Obsahem souboru je popis deskriptorů zařízení a také definice hodnot pro vstupní a výstupní endpointy. Tento soubor obsahuje všechny proměnné pro kompletní struktury endpointů.
- **usb\_compile\_time\_validation.h** – kontrola velikosti endpoint deskriptoru dle standardu USB. Tedy dosahovaná velikost tohoto deskriptoru může být buď 8, 16, 32 nebo 64 bajtů.

- **usbcfg.h** – Pomocí tohoto souboru je provedena konfigurace endpointů zařízení. Tedy je určena hodnota a výchozí nastavení pro endpointzero a dále pak přiřazení endpointu pro konfigurační deskriptor a dále pak také hodnoty pro interface deskriptory a jejich endpointy.
- **usbdsch a .c** - Tento modul obsahuje informace o USB deskriptorech. V souboru usbdsch jsou obsaženy definice struktur konfigurací a globalizují se zde deskriptory pro viditelnost i v ostatních modulech prostřednictvím klíčového slova `extern`. Jestliže uživatel přidá nebo odebere deskriptor z hlavního deskriptoru konfigurace, je zde třeba také změnit jeho strukturu. V souboru `usbdsch.c` je provedena inicializace konstant deskriptoru zařízení, deskriptoru konfigurací a požadovaných USB stringů. Důležité je poté naplnění pole `ClassReqHandler[x]` ukazateli na obslužné rutiny standardních požadavků třídy implementovaného zařízení. Obsluha paketů přichozích přes endpointzero probíhá tak, že hlavní obslužná rutina pro setup transferu z modulu `usbctrltrf.h` dá při obdržení paketu šanci k obslužení nejprve rutině `USBCheckStdRequest()` a neumí-li ho tato obsloužit, jedná se buď o chybu nebo o standardní request třídy a na řadu tedy přijdou všechny uživatelské rutiny uvedené právě v poli `ClassReqHandler` [11].
- **usbmap.h a .c** - Tento modul představuje USB paměťový manažer. Alokace USB endpointu a jejich buffer deskriptorů probíhá dynamicky během kompilace s využitím některých parametrů definovaných v `usbcfg.h`. Každý endpoint vyžaduje pro jeden směr transferu svůj Buffer Deskriptor (BD) uložený ve struktuře Buffer Descriptor Table (BDT). BD je čtyři bajty dlouhý a má pro daný endpoint pevně přiřazenou RAM adresu. BDT pro endpoint 0 se nachází na adresách `0x400 – 0x403`, endpoint 1 na adrese `0x404 – 0x407` atd. Tyto adresy jsou již hardwarově zadrátovány a nelze je změnit. Endpointy jsou jednoznačně definovány použitím čísla endpointu a směru transferu. Pro jednoduchost však `usbmap.c` používá pouze čísla endpointů, automaticky se pak alokují dva BD, jeden pro transfer IN a druhý pro OUT. Pro alokaci BD endpointu je použita hodnota `MAX_EP_NUMBER` definovaná v `usbcfg.h`. Je-li např. `MAX_EP_NUMBER` 1, jsou automaticky vytvořeny čtyři BD: `ep0 in` a `out`, `ep1 in` a `out`. Jelikož je tato alokace pevně daná, nastavení zbytečně vysokého `MAX_EP_NUMBER` může vést k neefektivnímu využití paměti. Např. Pokud zařízení používá EP0 a EP4, `MAX_EP_NUMBER` je 4 a paměť zabraná alokací BD pro EP1, EP2 a EP3 (24 bajtů) je alokována

zbytečně. Vynechávat endpointy nemá žádný smysl, na funkčnost to však nemá žádný vliv a rozhodující to může být pouze v aplikacích kritických na paměť. Konečné rozhodnutí je pak na programátorovi. Dalším krokem, který je zde proveden, je alokování paměti pro samotné endpoint buffery všech použitých endpointů. Endpoint buffer každé USB funkce se musí nacházet v dual port RAM oblasti a deklarace musí přijít po tom, co byly alokovány všechny Buffer Deskriptory.

- **usbdrv.c a .h** - Tento modul USB komunikaci a funkční propojení ostatních modulů. Nejdůležitější funkcí je void `USBDriverService(void)`, která se stará o všechna USB přerušení a volá ostatní funkce postupně tak, jak zařízení prochází jednotlivými stavy enumerace. Důležité je také makro void `mUSBBufferReady(buffer_dsc)`, které přepne vlastnictví bufferu endpointu PC pro obsluhu. Volá se po naplnění bufferu dat pro transfer ven, nebo po připravení bufferu pro data dovnitř. Služeb tohoto makra využívají všechny vyšší moduly. Pomocí dalších funkcí se zde pak ovládá aktivace a deaktivace modulu, uspává a probouzí zařízení (SUSPEND), reaguje na chyby atd. [11].
- **usb9.c a .h** - Tento modul obsluhuje standardní USB požadavky přicházející přes EP0 dle kapitoly 9 specifikace USB 2.0 [10]. Srdcem modulu je rutina void `USBCheckStdRequest(void)`, která otestuje setup paket a pokud ví, jak ho obsloužit, volá příslušné další rutiny:  
`void USBStdGetDscHandler(void)`  
`void USBStdSetCfgHandler(void)`  
`void USBStdGetStatusHandler(void)`  
`void USBStdFeatureReqHandler(void)`
- **usbctrltrf.h a .c** - Srdcem tohoto modulu je rutina void `USBCtrlEPService(void)`, která obsluhuje pouze následující tři transakce - EP0 SETUP, EP0 OUT, EP0 IN a volá příslušné rutiny. Nejdůležitější z nich je funkce void `USBCtrlTrfSetupHandler(void)`, která slouží jako dispečer. Při obdržení setup paketu postupně volá jednotlivé moduly (USB9,HID), zda ho neumějí obsloužit. Obsluhu poté dokončují rutiny void `USBCtrlEPServiceComplete(void)` a void `USBPrepareForNextSetupTrf(void)`. Komunikaci přes EP0 realizují funkce void `USBCtrlTrfOutHandler(void)` a void `USBCtrlTrfInHandler(void)` [11].

V případě běžného programování zařízení kterékoliv ze tříd zařízení (HID, MSD, CDC) nevyžadují žádný zásah. Bez hlubšího studia naopak jakékoliv zásahy mohou být na škodu. V případě programování USB MIDI Lights Device byly některé zásahy nutné a budou popsány v oddíle zabývající se praktickou stránkou věci. Níže popsané moduly se vztahují k jednotlivým třídám. Jejich programové využití závisí na tom, do jaké kategorie je programované zařízení zařazeno.

- **hid – hid.c a .h** – Již podle názvu je patrné určení tohoto modulu. Základní rutinou je void USBCheckHIDRequest(void) volaná z funkce USBCtrlEpService(). Pro inicializaci endpointu je zde rutina void HIDInitEP(void), která je volána po obdržení standardního USB požadavku SET\_CONFIGURATION. Pro vlastní transport dat jsou připraveny rutiny void HIDTxReport(char \*buffer, byte len) a byte HIDRxReport(char \*buffer, byte len), které přepokopírují data z uživatelského bufferu do bufferu endpointu a pro transport dat přes sběrnici využívají služeb makra mUSBBufferReady(). Stav transferu ve smyslu probíhá×hotovo lze průběžně testovat makry mHIDRxIsBusy(void) a mHIDTxIsBusy(void), které testují, zda buffer endpointu vlastní počítač nebo mikrokontrolér (bit UOWN BDnSTAT registru). Tento modul řeší pouze standardní HID požadavky. Vlastní funkčnost zařízení je implementována v modulu user/user\_mouse, který obsahuje implementaci funkce ProcessIO(). Ta obsahuje vlastní výpočty a pro odesílání připravených dat přes USB sběrnici používá služeb funkcí HIDTxReport() a HIDRxReport(). Tímto je oddělena rovina mezi skutečnou implementací a třídou zařízení. Při vývoji HID zařízení tedy stačí naimplementovat vlastní funkčnost v modulu user\_mouse a modul HID nevyžaduje buďto žádné, nebo pouze malé změny [11].
- **msd.c a .h** – Obsahem modulu je implementace funkcí rozhraní zařízení třídy MSD. Hlavičkový soubor obsahuje definice implementovaných SCSI příkazů a jejich kódů, definice chybových kódů apod. Standardní MSD požadavky přes EP0 obsluhuje funkce USBCheckMSDRequest(). O inicializaci příslušných endpointů se podobně jako v případě HID zařízení stará funkce MSDInitEP(). Vlastní funkčnost zařízení je opět implementována ve funkci ProcessIO(). Firmware implementuje zařízení standardu SBC-2 se SCSI transparentním příkazovým setem. Komunikace probíhá podle Command/Data/Status protokolu [11].

- **cdc.c a .h** – je posledním v řadě modulů, které specifikují jednotlivé třídy zařízení v rámci USB protokolu. Obsahem modulu jsou funkce void USBCheckCDCRequest(void), která se stará o obsluhu setup paketu, pokud má k tomu příslušné informace. Dále pak void CDCInitEP(void). Tato funkce inicializuje endpointy zařízení a buffer deskriptoru, zjišťuje vnitřní stav zařízení a nastavení jednotlivých proměnných. Měla by být vždy volána až po obdržení informací získaných ze SET\_CONFIGURATION požadavku. Funkce void CDCTxService(void) řídí komunikaci ze zařízení do počítače. Měla by být volána pouze jednou v rámci hlavního cyklu. S ní souvisí několik dalších funkčních volání, která mají na starosti přenos dat ze zařízení do počítače a obráceně. Pro snadnou obsluhu jsou v hlavičkovém souboru definována makra, která ve většině případů slouží jako podmínkové stavy pro volání funkcí přenosu dat.
- **user.c a h.** – Modul obsahující uživatelské funkce, proměnné, nastavení a makra.

Všechny výše uvedené soubory jsou součástí MCHPFSUSB. Jedná se o balíček vývojových součástí, které usnadňují programování jednočipových kontrolerů od společnosti Microchip. Je určen pro procesory řady 18F2455, 18F2550, 18F4455 a 18F4550, které mají zabudovaný USB modul. Tento balíček je poskytován zdarma. MCHPFSUSB je bezplatně poskytován na webových stránkách mateřské společnosti Microchip. Po instalaci vzniknou v počítači adresáře, z nichž tři nejdůležitější jsou popsány níže:

C:\MCHPFSUSB\FW\Cdc\ - obsahuje firmware (dále jen „FW“) třídy CDC pro procesor

C:\MCHPFSUSB\FW\Boot\ - obsahuje Bootloader pro procesor

C:\MCHPFSUSB\PC\ - obsahuje ovládací program Bootloaderu pro PC a některé ovladače.

## 4.8 Bootloader

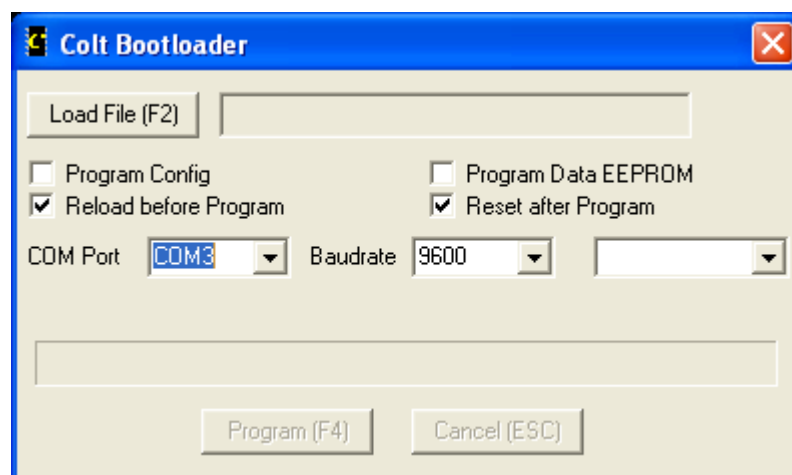
Softwarová součást, která nám umožňuje přehrávat programovou paměť bez poškození registrů, které nesmí být z důvodu správné funkčnosti přepsány. Dále nám nahráný program v bootloaderu umožňuje identifikovat zařízení v systému v době nahrávání



softwaru, tedy když je zařízení přepnuto do režimu přepisu uživatelské části programové paměti.

#### 4.8.1 Colt PIC18F Bootloader

Tato utilita slouží pro naplnění paměti programovým vybavením pro bootloader. Po instalaci je pomocí této utility možno přehrávat programy, které slouží jako bootloader. Colt PIC18F automaticky detekuje přenosovou rychlost na příslušném portu a následně je schopen naprogramovat bootovací sektor procesoru. Po nabootování je potom zařízení identifikováno jako nové USB zařízení a je třeba ještě doplnit .inf soubor s názvem mchpcdc.inf jako ovladač do systému pro toto USB zařízení, které v tomto stavu emuluje sériový port a díky tomuto bootlooaderu je možno nahrávat uživatelský kód přímo do procesoru přes USB.

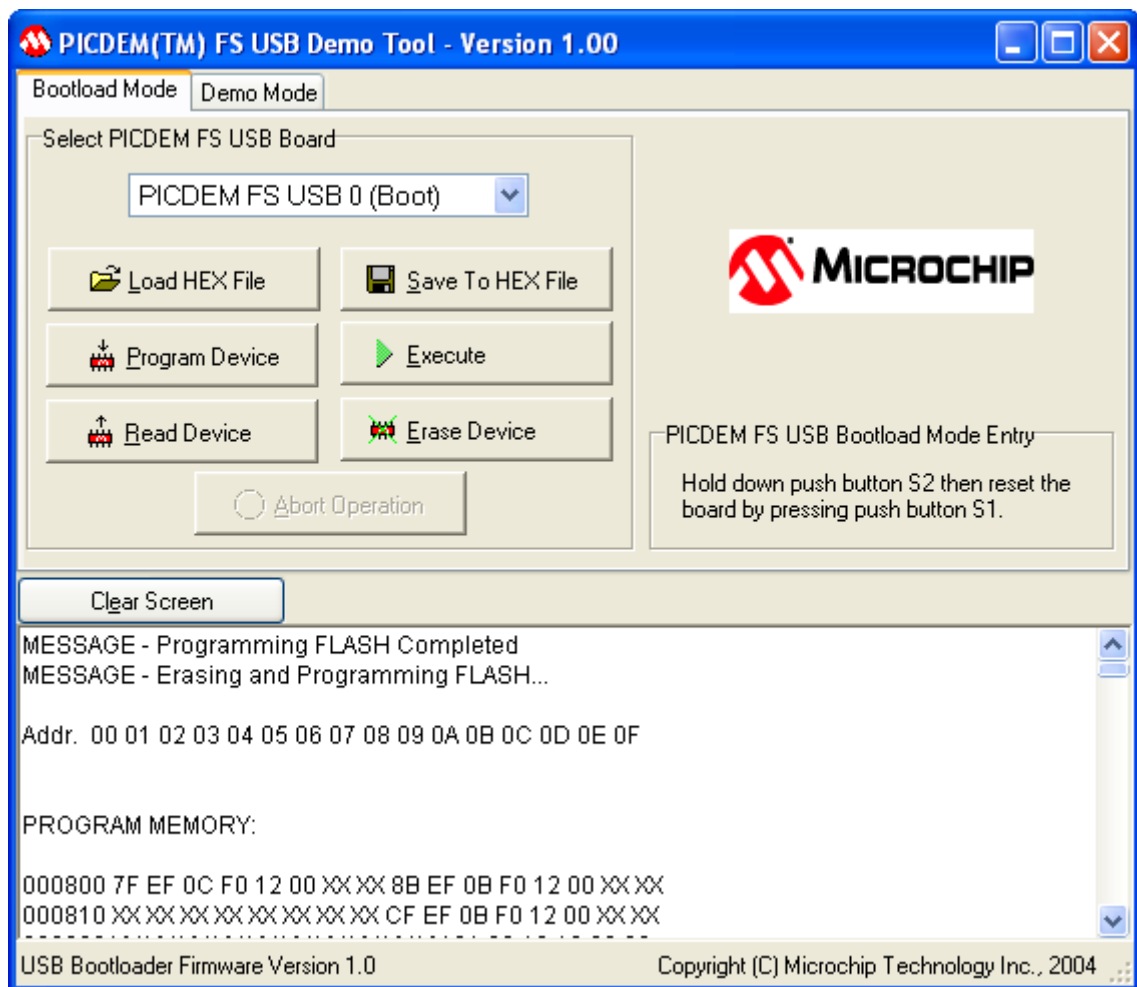


Obr. 13. Konfigurační okno Colt PIC 18F

#### 4.8.2 Picdem FS USB

Bootloader je tedy program, který je zapsán do horní části paměti programu mikroprocesoru. Tato část je pak zablokována (Code Protection). Pokud je aktivován, umí přepisovat nezablokované části paměti programu. Tzn. data přicházející po USB jsou nahrávána do procesoru -> procesor je programován. PICDEM FS USB je softwarová utilita, která přehrává uživatelský program v podobě hex souboru, který je výsledkem kompilace a linkování v prostředí MPLAB. Tato utilita je poskytována zdarma firmou Microchip, přímo na stránkách společnosti. Automaticky detekuje připojené zařízení. Pokud nějaké najde, zobrazí jej a nabídne k přepisu uživatelského programu. Od uživatele

je očekáván výběr příslušného hex souboru. Pokud je tento vybrán, je po stisku tlačítka *Program Device* nahrán do paměti procesoru. Po té je třeba zařízení přepnout do uživatelského režimu, např. restartovat odpojením od PC, pomocí jumperu přepnout a znovu připojit.



Obr. 14. Konfigurační okno PICDEM FS USB bootloaerů

Vlivem použití daného bootloaerů jsou změněny adresy v registrech, a to následujícím způsobem:

- Reset vector 0x800,
- High Interrupt vector 0x808,
- Low Interrupt vector 0x818.

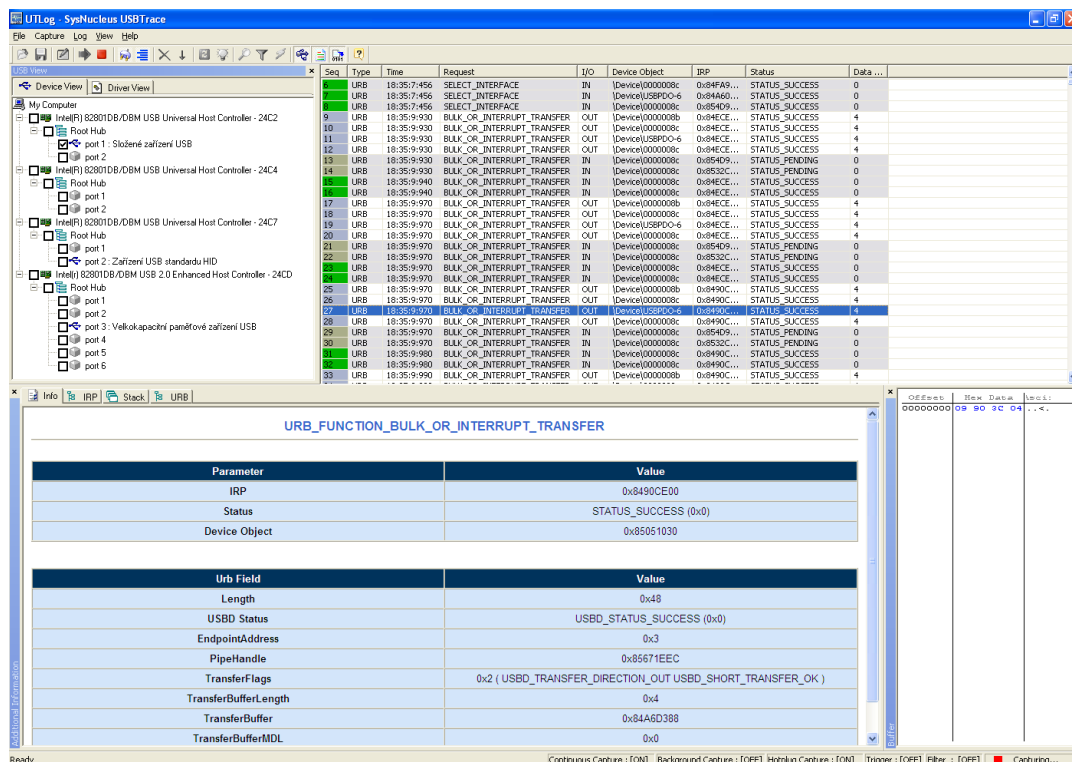
Původní hodnoty jsou o 0x000 pro Reset vektor, pro High Interrupt vektor je to potom 0x008 a nakonec původní hodnota Low Interrupt vektoru je 0x018.

Na tomto místě je třeba uvést pro úplnost, že vývoj USB MIDI Lights Device probíhal střídavě na dvou operačních systémech tak, aby bylo možno prokázat jeho univerzálnost alespoň v rámci platformy Windows. Jak programové prostředí MPLAB s kompilátorem MCC C18, tak bootloader PICDEM FS USB se ukázali zcela bezproblémovými co se týče ovladatelnosti na počítači jak se systémem XP Professional Service Pack 2, tak na Windows Vista Business. Veškerý programový kód byl generován bez nejmenších problémů a výsledný hex soubor byl bez problémů implementován do zařízení. Toto bylo oběma systémy identifikováno bez chyb, jen s tím rozdílem, že Windows XP neviděli úplný název a zařízení identifikovali jako Zvukové zařízení USB. Naopak systém Windows Vista Business byl schopen vyzískat i úplný název zařízení a tento poté bez problémů pro lepší přehlednost zobrazovat v aplikacích.

## **4.9 Testovací nástroje**

### **4.9.1 USB Trace**

Tato utilita od společnosti Sysnucleus identifikuje USB zařízení a také dokáže monitorovat přenos dat z počítače do zařízení a zpět. V rámci vývoje USB MIDI Lights Device byla využita především při nastavení příslušných endpointů pro identifikaci zařízení. V rámci identifikace bylo důležité, aby se zařízení hlásilo prvně jako Audio rozhraní a po té bylo nutné, aby se dále identifikovalo jako MIDI interface. USB Trace zobrazoval skutečné čísla endpointů, na kterých zařízení komunikuje. Toto zařízení je dostupné jako shareware na stránkách <http://www.sysnucleus.com>.



Obr. 15. Diagnostické okno USB Trace

#### 4.9.2 Virtual MIDI keyboard

Utilita, která usnadnila generování MIDI zprávy Note On a Note Off. Díky této utilitě vytvořené Wouterem van Beekem nebylo nutné připojovat skutečnou MIDI klávesnici a vývoj tím byl značně urychlen.

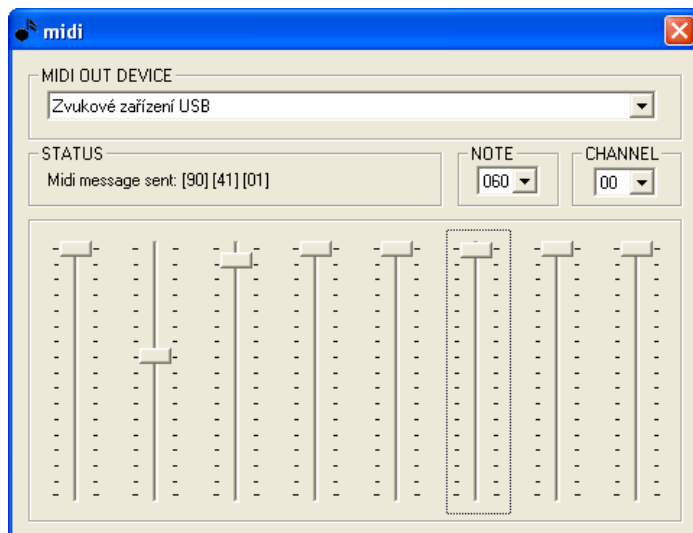


Obr. 16. Virtual MIDI keyboard

Jelikož nebylo možné zcela přesně kontrolovat hodnotu Velocity ve vyslané MIDI zprávě, byla ve spolupráci s Ing. Davidem Příbylem vyvinuta utilita, která umožňovala generovat parametr Velocity a tento přesně kontrolovat a zobrazovat jeho hodnotu.

### 4.9.3 Midi.exe

Tato aplikace sestává z jednoho dialogového okna, ve kterém uživatel zvolí zařízení, do kterého jsou směřovány MIDI zprávy. Událost Note On je generována v okamžiku, kdy příslušné táhlo není na nulové hodnotě Velocity parametru, který se zobrazuje nad táhly. Takto je možno generovat přesnou hodnotu Velocity parametru a kontrolovat reakci zařízení, případně celého systému.



Obr. 17. Konfigurační dialog aplikace midi.exe

## **II. PRAKTICKÁ ČÁST**

## 5 PRAKTICKÁ ČÁST

### 5.1 Identifikace do systému

Zařízení, už podle názvu, se identifikuje pomocí svého endpointzero do systému jako USB zařízení. Pomocí svých interface deskriptorů je pak po úvodní enumeraci identifikováno jako Audio zařízení s upřesňující specifikací MIDI. Na audio interface deskriptor nejsou vázány další endpointy. Audio rozhraní je zde jen a pouze pro potřeby operačního systému, aby ten bez problémů rozlišil a rychle identifikoval, jaký má použít ovladač. V zařízení jsou dále použity všechny deskriptory pro popis endpointů, které jsou v případě MIDI zařízení nahrazeny a označeny jako deskriptory pro MIDI IN resp. OUT Embedded, resp. External konektory (jacky). Tedy, aby mělo zařízení celkem čtyři obousměrné endpointy. Tyto mají ještě své Class – specific deskriptory, které určují především, jakým způsobem budou přenášena data. V případě MIDI zařízení se jedná o Bulk přenosy.

### 5.2 Přenos dat

MIDI zprávy byly v průběhu vývoje generovány jedním ze zařízení, které byly popsány v předchozí kapitole. Tato data je možno zpracovávat několika způsoby. Aby nedocházelo ke zbytečným komplikacím, a protože je ve výsledku zařízení osmi-kanálové, reaguje pouze na osm MIDI not. Jsou to noty s čísly 60-67. Hudebně (muzikantsky) se jedná o tyto noty:

- Note On číslo 60 – nota jednočárkované c,
- Note On číslo 61 – nota jednočárkované cis, resp. des,
- Note On číslo 62 – nota jednočárkované d,
- Note On číslo 63 – nota jednočárkované dis, resp. es,
- Note On číslo 64 – nota jednočárkované e,
- Note On číslo 65 – nota jednočárkované f,
- Note On číslo 66 – nota jednočárkované fis, resp. ges,
- Note On číslo 67 – nota jednočárkované g.

Pokud uživatel stiskne notu s jinou hodnotou v události MIDI On, zařízení nebude reagovat, i když je výstup MIDI události směřován právě do něj. Jedna nota tedy

představuje jeden ovládaný kanál, tedy jedno zařízení připojené na jednom kanále výkonového stupně, např. jedno čerpadlo, nebo jedno světelné zařízení. Záměrně je zde uváděno jedno světelné zařízení, nikoliv jedno světlo nebo dokonce jeden reflektor. Světelným zařízením může být celá sestava jednotlivých reflektorů, které budou připojeny k jednomu, výkonostně odpovídajícímu kanálu a budou společně – stejně reagovat na příkaz vyslaný právě na tomto kanále. Dále jsou v rámci uživatelského programu ošetřeny hodnoty Velocity, které jsou obsahem čtvrtého (posledního) bajtu USB MIDI Eventu. V rámci kódu je spuštěn timer, který pak na základě parametrů nastavených uživatelem vyvolá přerušení s vysokou prioritou. Parametr Velocity je klíčovým pro ovládání všech zařízení, která budou svou funkčností závislá na napětových úrovních na výstupu USB MIDI Lights Device.

### 5.3 Využití zařízení v rámci MIDI (hudebního) softwaru

Každý hudební software, který je nastaven ke komunikaci s MIDI zařízeními, je schopen identifikovat USB MIDI Lights Device na základě informací, kterými se zařízení prokázalo do operačního systému při enumeraci. V našem případě se jedná o Cubase SX 2 od společnosti Steinberg. Tento software slouží jako nahrávací a editační centrum pro video, audio a MIDI záznamy.

V prostředí Cubase SX 2 je možno vytvořit jak „ryzí“ MIDI projekt, tak i projekt kombinovaný, tedy sestávající z audio i MIDI stop. Ukázkový příklad je takovou kombinací. Vytvoříme projekt, který bude obsahovat audio stopu s písní, a pod ním vytvoříme MIDI stopu, která bude obsahovat ovládání světel a čerpadel. Jak přesně vytvořit audio resp. MIDI stopy a jak s nimi přesně pracovat je podrobně popsáno v bakalářských pracích Pavla Křeménka, Petra Hoffmanna a také v bakalářské práci autora [18], [19], [20].

K ovládání světel a čerpadel nám tedy na základě specifikace hardwaru a softwaru USB MIDI Lights Device bude stačit pouze osm not v rozmezí c1 – g1. Naše zadané noty nebudou nijak přehrávány, a tudíž nebudou nijak zvukově kolidovat s vloženou audio stopou. Tyto noty budou totiž pomocí USB MIDI Lights Device transformovány na povely pro spuštění čerpadel a rozsvícení světel.



Stejný postup je potom i v MIDI projektu, který neobsahuje žádné audio stopy. Jedna z MIDI stop bude pojmenována např. světla a budou v ní zaznamenány příslušné noty daného rozsahu, tak jak bylo specifikováno výše. Opět noty nebudou nijak zvukově kolidovat s dalšími MIDI stopami z důvodu, který byl zmíněn v předchozích větách.

Nejdůležitější pro správnou funkčnost zařízení je toto specifikovat jako výstupní zařízení ve stopě, kterou chceme ovládat čerpadla a světla. Je tedy nutné mít určeno v položce MIDI Out naše zařízení, tedy USB MIDI Lights Device. Pokud tak neučiníme a bude vybrána např. zvuková karta, tak budou noty původně určené pro světla skutečně přehrávána jako další audio, resp. MIDI stopa.

V neposlední řadě je také nutno si vybrat, která nota bude spouštět čerpadla a která nota rozsvítí světla. Toto už je ale otázkou především uživatelskou. Některý z uživatelů může disponovat pouze osmi kanály pro osm čerpadel a určí, že nota c1 sepne čerpadlo číslo 1, nota cis1 sepne čerpadlo číslo dvě, a ostatní noty budou přiřazeny v dalším logickém sledu až do čerpadla číslo osm. Jiný uživatel provede to samé, ale protože bude mít k dispozici osm světel o dvou barvách, tak přiřadí každou sudou notu první barvě a každou lichou notu druhé barvě. Samozřejmě není určeno, že na jednom kanálu musí být jedno zařízení, ale jak již bylo zmíněno výše, toto již není otázkou popisovaného zařízení, ale uživatelskou konfigurací příslušného systému.

Jedním z možných, ale ne zcela žádaných způsobů využití tohoto zařízení, je i možnost směřovat do tohoto zařízení některou audio (MIDI) stopu, která přehrává hudební nástroj. Výsledkem by bylo neuspořádané blikání světel a spínání čerpadel. Zařízení by totiž reagovalo jen na noty, které jsou specifikované v jeho rozsahu (c1 – g1).

#### **5.4 Odezva koncového zařízení na USB MIDI Lights Device**

Realizované koncové zařízení se nachází na Fakultě aplikované informatiky Univerzity Tomáše Bati ve Zlíně v laboratoři D 303. Pro prvotní testy bylo využito jedno z běžných PC dostupných v této laboratoři. Enumerace proběhla bez problémů a USB MIDI Lights Device se identifikovalo ihned po připojení. Dále bylo toto zařízení na svých vstupech propojeno pomocí konektorů přímo s koncovým stupněm ovládajícím čerpadla a světla.

Pro další testy, kdy bylo třeba testovat koordinaci světel čerpadel a hudby, byl využit notebook autora spolu s externí zvukovou kartou M-Audio modelové řady Fast Track Pro a reproboxy. Zvuková karta a repro soustava jsou majetkem univerzity.

Lze říci, že i přes počítačnické obavy při vývoji softwaru byla odezva použitých čerpadel a světel velmi plynulá a nedocházelo ke skokovým změnám. Takto kladný výsledek byl docílen právě využitím pulzně-šířkové modulace a snahy nijak neomezovat rozpětí hodnot parametru Velocity v MIDI zprávě. Rozsah parametru Velocity je 0 - 127. Vlivem pulzně-šířkové modulace byl k dispozici pro ovládání celý rozsah, tedy 127 hodnot. Tyto byly beze zbytku využity. Pouze v krajních mezích byla patrná malá prodleva průtoku čerpadel a také intenzity světel.

## **5.5 Další možné cesty vývoje popisovaného zařízení**

V následujících odstavcích jsou na úplný závěr zmíněny některé další možné způsoby ovládání a nastavení USB MIDI Lights Device.

### **5.5.1 Ovládání pomocí celé klaviatury**

V rámci ovládacího softwaru by bylo možné v tomto případě nerozlišovat čísla not a pouze reagovat na prvním bajtu na hodnotu 0x90 (144) – Note On, tedy sepnuto a následně pak na základě dostupného koncového zařízení pouze rozlišovat např. pomocí generátoru náhodných čísel, které ze světel může být rozsvíceno. V úzké návaznosti by bylo nutné také rozlišovat hodnotu 0x80 – Note Off, tedy klávesa vypnuta. Sepnutá zařízení není možné odpojit bez použití tohoto parametru.

Možným dalším řešením je opatřit si 88 světelných zdrojů a ke každému přiřadit jednu notu a využít tak plný rozsah klaviatury. Je možno také do počtu 88 kláves využít i jiná zařízení – např. umělý dým.

### **5.5.2 Využití jedné MIDI stopy pro více výstupních zařízení**

Dalším možným způsobem je směřovat stopu nástroje zároveň i do USB MIDI Lights Device. Například part basu směřovat na výstupu nejen do zvukové karty, ale zároveň i ke světlům. Nastavení našeho zařízení by pak odpovídalo popisu v předchozím článku tak, abychom nebyli omezeni žádným notovým rozsahem.

### **5.5.3 Generátor světelné stopy s využitím Fourierovy transformace**

Toto řešení už není založeno jen na USB MIDI Lights Device. Tento generátor by především usnadnil vytvoření světelné stopy tak, aby došlo k ušetření uživatelského času při tvorbě světelného partu. Jednalo by se o softwarovou automatizační utilitu, která by z audio vzorku na základě definovaných frekvencí vytvářela MIDI stopu světel. Inspirací

pro tuto myšlenku byla diplomová práce Ing. Nušla, který tento princip využíval, ale pro ovládání světel, které pro komunikaci používají DMX 512 protokol. Toto řešení by mohlo pak existovat ve dvou variantách. Jednak jako aplikace, která pracuje reálnodobě, ale také jako aplikace, která generuje MIDI stopu až po vlastním spuštění uživatelem. Tedy aplikace by se dala využívat jak při studiové práci, tak při živém hraní.

#### **5.5.4 Uživatelské rozhraní pro nastavení USB MIDI Lights Device**

Aby mohlo být zařízení „uživatelsky přátelské“, v dalším vývoji by bylo možné vytvořit GUI jak pro jednotlivé způsoby řešení, tak GUI pomocí kterého by bylo možné přepínat mezi jednotlivými módy a využívat tak popisované zařízení v co největší možné míře.

Dalšími možnostmi této obslužné aplikace by mohla být vizualice nastavených parametrů. Uživatel by pak pomocí obrazové simulace mohl na monitoru svého PC sledovat, jaká provedl světelná nastavení.

Toto GUI by bylo pak dále možné upravit tak, aby se potom výsledná aplikace dala použít jako plug-in nebo stand-alone aplikace. Tato aplikace by poté mohla pracovat samostatně. V lepším případě by byla zásuvným modulem MIDI softwaru.

## ZÁVĚR

Na začátku této diplomové práce bylo stanoveno několik dílčích cílů. Lze říci, že se podařilo je naplnit všechny. Byl realizován jednočipový vysoce univerzální kontroler, který je možno zapojovat do jakéhokoliv koncového zařízení a koordinovat tak vizuální efekty na jevišti spolu s hudbou. Zařízení je koncipováno tak, že je možno programově ošetřit nejen klávesy (čísla not), na které bude zařízení reagovat, ale také je možno přes výkonový stupeň připojit nejen světelná zařízení, ale jakákoliv kompatibilní elektrická zařízení, která jako vstup používají TTL logiku. Na vstupu výkonových stupňů pro jevištní zařízení musí být napěťová úroveň odpovídající TTL. Tedy např. motory čerpadel vodních fontán nebo zařízení vytvářející umělý dým.

Mezi základní výhody zařízení patří jeho výrobní cena. Tato cena se pohybuje v řádu stovek korun za jeden kus oproti několika tisícovým cenám za komerčně dostupná zařízení. Další výhodou je jeho snadná implementace do koncových stupňů světelných aplikací. Ze zařízení bude dostupný pouze USB konektor a případně přepínač jumperu pro snadné přeprogramování samotné aplikace v případě změn ve světelné sestavě. Významným přínosem je možnost využít jakákoliv efektová zařízení bez ohledu na to, jakým protokolem disponují či nedisponují v rámci své softwarové výbavy.

Jediným nutným předpokladem je ošetření výstupů pro světelnou sestavu pomocí tlumivek v případě velkých příkonů z důvodů ovládní intenzity pomocí pulzně šířkové modulace.

Zařízení je jistě dále konfigurovatelné a programovatelné. Tedy pokud by se dále pokračovalo ve vývoji je možno vytvořit několik dalších principiálních funkcí, které by umožnily lépe a pohodlněji ovládat příslušná jevištní zařízení.

## CONCLUSION

At the beginning this diploma thesis were specified several particular targets. Is it possible to tell, all was executed trust. It was realized single-chip universal controller, which is possible to connect any end lights device and coordinate visual stage effect conjunction with music. In the device is possible to program numbers of music notes for lights devices and it is possible connect end devices, that use input TTL logic devices. On enter must be TTL voltage level. For example water fountains engines or devices, which create synthetic smoke.

Elementary advantage of this application is cost price. This price is about a few hundreds of crowns against commercial prices by other appliances. Other advantage is easy implementation this application. This application will fully implemented inside ending lights levels. From ending lights levels will only accessible USB connector and eventually jumper for easy re-programming, if user will make some changes in luminous group. Meaningful contribution is option to use any equipment uncared-for software facilities of lights equipments.

Only one presumption is necessary for smooth functioning. On the out of lights equipment must any voltage suppressors for high voltage input. It is necessary through pulse width modulation using on out of our application.

Described USB MIDI Lights Device is surely further configurable and programmable. So in other development is possible to create other fundamental functionalisms for more comfortable control lights devices. Meaningful contribution is possibility

## SEZNAM POUŽITÉ LITERATURY

- [1] *Přehled standardu MIDI* [online]. [cit. 2008-05-15]. Dostupný z WWW:  
<<http://www.midi.cz>>
- [2] Nušl, J. *Řízení techniky komunikující protokolem DMX 512 pomocí dat získaných z hudby v reálném čase*. Praha, 2006. 62 s. Diplomová práce na Fakultě elektrotechnické na Českém vysokém učení v Praze. Vedoucí diplomové práce Ing. Jiří Buček.
- [3] Lukačovič, P. *Počítačové řízení fontány*. Zlín, 2004. 53 s. Diplomová práce na Fakultě technologické Univerzity Tomáše Bati na Institutu řízení procesů a aplikované informatiky. Vedoucí diplomové práce prof. Ing. Vladimír Vašek, CSc.
- [4] *Popis protokolu DMX512* [online]. [cit. 2008-04-25]. Dostupný z WWW:  
<<http://www.soundlight.de/techtips/dmx512/dmx512.htm>>
- [5] *MIDI zařízení Harveye Twymana* [online]. [cit. 2008-04-28]. Dostupný z WWW:  
<<http://www.twyman.org.uk/midilite.htm>>
- [6] *Parametry zařízení Botex MPX405* [online]. [cit. 2008-05-05]. Dostupný z WWW:  
<<http://eshop.prodance.cz/Files/MPX405%20MultiPak%20cz.pdf>>
- [7] *Ovládání světel pomocí MIDI* [online]. [cit. 2008-05-05]. Dostupný z WWW:  
<<http://tomscarff.tripod.com>>
- [8] *Popis produktů společnosti Darys* [online]. [cit. 2008-05-07]. Dostupný z WWW:  
<<http://www.konin.pl/darys/ainfo.htm>>
- [9] *Společnost Midilite a její produkty* [online]. [cit. 2008-05-08]. Dostupný z WWW:  
<<http://www.midilite.com>>
- [10] *Universal Serial Bus specification.pdf* [online]. [cit. 2008-04-08]. Dostupný z WWW:  
<<http://www.usb.org>>
- [11] Mach, J. *Firmware pro USB zařízení s mikro počítači PIC*. Praha, 2006, 62 s.  
Diplomová práce na Fakultě elektrotechnické Českého vysokého učení technického.  
Vedoucí diplomové práce Ing. Miroslav Skrbek, PhD.
- [12] *Universal Serial Bus Device Class Definition for MIDI Devices.pdf* [online].  
[cit. 2008-04-18]. Dostupný z WWW:  
<<http://www.usb.org>>

- [13] *Universal Serial Bus Device Class Definition for Audio Devices.pdf* [online].  
[cit. 2008-05-20]. Dostupný z WWW:  
<<http://www.usb.org>>
- [14] *Universal Serial Bus Class Definition for Communication Devices.pdf* [online].  
[cit. 2008-04-01]. Dostupný z WWW:  
<<http://www.usb.org>>
- [15] *PIC18F2455/2550/4455/4550 Data Sheet.pdf* [online]. [cit. 2008-05-20].  
Dostupný z WWW:  
<<http://www.microchip.com>>
- [16] *MPLAB C18 C Compiler User's guide.pdf* [online]. [cit. 2008-05-20].  
Dostupný z WWW:  
<<http://www.microchip.com>>
- [17] *MPLAB C18 C Compiler Libraries.pdf* [online]. [cit. 2008-05-20].  
Dostupný z WWW:  
<<http://www.microchip.com>>
- [18] Křemének, P. *Vybavení laboratoře multimediální techniky*. Zlín: Univerzita Tomáše Bati, Fakulta technologická, Institut řízení a aplikované informatiky, 2005. 71 s. Vedoucí diplomové práce prof. Ing. Vladimír Vašek, CSc.
- [19] Hoffmann, P. *Využití Cubase pro zpracování audio – signálů*. Zlín: Univerzita Tomáše Bati, Fakulta technologická, Institut řízení a aplikované informatiky, 2005, 81 s. Vedoucí diplomové práce prof. Ing. Vladimír Vašek, CSc.
- [20] Slovák, D. *Protokol MIDI a jeho implementace v programovém prostředí Cubase SX 2 s využitím přídavných VST instrumentů*. Zlín: Univerzita Tomáše Bati, Fakulta aplikované informatiky, Ústav aplikované informatiky, 2006, 79 s. Vedoucí diplomové práce prof. Ing. Vladimír Vašek, CSc.

## SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

<b>AppleTalk</b>	síťový protokol vyvinutý firmou Apple pro počítačové sítě.
<b>CDC</b>	Communication Device Class. Všechny druhy komunikačních zařízení, které jsou obsluhovány prostřednictvím USB protokolu.
<b>CIN</b>	Code Index Number – identifikace MIDI zprávy pro komunikaci přes USB
<b>EUSART</b>	Enhanced Universal Synchronous Asynchronous Receiver Transmitter
<b>FAI UTB</b>	Fakulta aplikované informatiky Univerzity Tomáše Bati ve Zlíně.
<b>GUI</b>	General User Interface – Uživatelské rozhraní aplikace
<b>HID</b>	Human Interface Device. Zařízení (počítačové vybavení), které reaguje na podněty a chování uživatele. Jedna ze tříd zařízení specifikace USB
<b>ICSP</b>	In-Circuit Serial Programming – Procesory takto označené je možno programovat bez fyzického odpojování a opětovného připojování od sestavy.
<b>IDE</b>	Integrated Development Environment – integrované vývojové prostředí
<b>IRP</b>	I/O Request Packet. Vstupně – výstupní požadavek pro přesun dat mezi hostem a zařízením.
<b>LIN</b>	Sběrnice LIN je nová sériová asynchronní sběrnice používající ke komunikaci jednovodičové spojení připojených zařízení. Nástupce sběrnice CAN.
<b>LSB</b>	Least signification bit – nejméně významý bit
<b>MIDI</b>	Musical Interface for Digital Instrument – norma popisující komunikaci mezi hudebními nástroji a počítačem
<b>MSB</b>	Most signification bit – nejvýznamnější bit



<b>MSD</b>	Mass Storage Device. Všechny druhy paměťových a diskových zařízení, které jsou specifikovány pomocí normy USB v 2.0
<b>RS - 232</b>	Rozhraní pro přenos informací vytvořené původně pro komunikaci dvou zařízení do vzdálenosti 20 m.
<b>SBC-2</b>	Jeden ze standardů blokových příkazů SCSI
<b>SCSI</b>	Small Computer System Interface – rozhraní pro připojení dalších zařízení do počítače
<b>SPI</b>	Serial Peripheral Interface – sériové rozhraní, které je využíváno při komunikaci mezi řídicími mikrokontrolery a jejich periferiemi.
<b>TTL</b>	Transistor – transistor logic – standard pro implementaci digitálních integrovaných obvodů.
<b>USART</b>	Universal Synchronous Asynchronous Receiver Transmitter
<b>USB</b>	Universal Serial Bus
<b>USITT</b>	United States Institute of Theatre Technology.
<b>XLR</b>	Nejčastěji třípinový konektor pro přenos audio signálu. Může být i pětipinový.

**SEZNAM OBRÁZKŮ**

Obr. 1.	Nastavení barevné intenzity v MixerMaps editoru v programu Cubase .....	13
Obr. 2.	Botex MPX – 405.....	14
Obr. 3.	Schéma zapojení MIDI hardwaru Toma Scarffa.....	15
Obr. 4.	MIDI kontroler ML-80P2.....	16
Obr. 5.	Modul ML-640 PS .....	17
Obr. 6.	LanBox - LCX.....	19
Obr. 7.	LanBox - LCE .....	20
Obr. 8.	Struktura USB stromu .....	24
Obr. 9.	Vrstvení jednotlivých endpointů v rámci struktury USB.....	25
Obr. 10.	Schéma jednotlivých součástí USB MIDI Lights Device .....	36
Obr. 11.	Schéma zapojení USB MIDI Lights Device.....	38
Obr. 12.	Vývojové prostředí MPLAB IDE od společnosti Microchip .....	40
Obr. 13.	Konfigurační okno Colt PIC 18F.....	46
Obr. 14.	Konfigurační okno PICDEM FS USB bootloaderu .....	47
Obr. 15.	Diagnostické okno USB Trace .....	49
Obr. 16.	Virtual MIDI keyboard.....	49
Obr. 17.	Konfigurační dialog aplikace midi.exe.....	50

**SEZNAM TABULEK**

Tab. 1.	Přehled jednotlivých módů zařízení ML 640-PS .....	17
Tab. 2.	Struktura USB MIDI Event paketu .....	33
Tab. 3.	Code Index Number a odpovídající MIDI zprávy .....	34
Tab. 4.	Popis MIDI zprávy a její bajtové vyjádření pomocí USB MIDI paketu.....	35