



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**

**ÚSTAV TELEKOMUNIKACÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

# **GENERÁTOR SYNCHRONIZAČNÍHO KÓDU PRO HUDEBNÍ APLIKACE**

SYNCHRONIZATION SIGNAL GENERATOR FOR MUSICAL APPLICATIONS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**LUKÁŠ VÁCHA**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. JIŘÍ SCHIMMEL, Ph.D.**

BRNO 2010



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Bakalářská práce

bakalářský studijní obor  
**Teleinformatika**

**Student:** Lukáš Vácha

**ID:** 109739

**Ročník:** 3

**Akademický rok:** 2009/2010

## NÁZEV TÉMATU:

### Generátor synchronizačního kódu pro hudební aplikace

#### POKYNY PRO VYPRACOVÁNÍ:

Navrhněte hardwarový generátor synchronizačního kódu MIDI Clock s jedním vstupem a několika výstupy, který bude kromě generování časových značek Timing Clock v uživateli zadaném tempu umožňovat vysílání příkazů Start, Stop a Continue a zpráv Song Position Pointer. Generátor bude obsahovat displej pro zobrazení tempa, pozice ve skladbě a tlačítka pro jejich nastavení a tlačítka Start, Stop, Continue a tlačítka pro odeslání nastavené pozice pomocí MIDI zprávy SPP. Tempo bude možné také nastavit pomocí MIDI zpráv System Exclusive. Pokuste se také implementovat možnost definování smyčky v přehrávání skladby pomocí zpráv SPP.

Navrhněte schéma zapojení, desku plošných spojů a program pro řídicí mikroprocesor a zařízení realizujte.

#### DOPORUČENÁ LITERATURA:

- [1] The Complete MIDI 1.0 Detailed Specification, document version 96.1, MIDI Manufacturers Association, Japan MIDI Standard Committee, 1997.
- [2] Forró, D. MIDI - komunikace v hudbě. Grada, Praha 1993.
- [3] Matoušek, D., Práce s mikrokontroléry ATMELE AVR, 2. vydání. Nakladatelství BEN, Praha, 2005. ISBN 80-7300-209-4

**Termín zadání:** 29.1.2010

**Termín odevzdání:** 2.6.2010

**Vedoucí práce:** Ing. Jiří Schimmel, Ph.D.

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

*Předseda oborové rady*

#### UPOZORNĚNÍ:

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

## ANOTACE V ČESKÉM JAZYCE

Tato bakalářská práce se zabývá rozбором MIDI protokolu a především oblasti implementace synchronizačních metod. Cílem této práce je návrh a vytvoření prototypu zařízení, které vysílá jeden druh synchronizačního kódu, a umožňuje jeho přímé nastavení na vyrobeném zařízení.

První část řešení práce se věnuje MIDI obecně, jeho historii, hardwarové definici a rozboru protokolu. Zejména pak rozboru metod a druhů synchronizace. Hlavní pozornost je věnována kódu MIDI Click / SPP, který je také implementován ve vytvořeném prototypu.

Druhá část práce se zabývá návrhem samotného hardwarového řešení prototypu, ve které je podrobně rozebrán princip zapojení a použité součástky. Třetí část práce popisuje detailně softwarový návrh zařízení. Od nutných inicializací mikrokontrolérů až po operace s jednotlivými periferiemi. Taktéž je zde rozebrána uživatelská obsluha. Poslední, čtvrtá část práce pojednává o samotné realizaci prototypu, jeho oživení a testování. Na závěr je uvedeno zhodnocení vlastností prototypu, návrhy pro případné rozšíření a shrnutí dosažených cílů práce.

**Klíčová slova:** AVR, ATtiny2313, MIDI, MIDI Click, MIDI SPP, SysEx

## ABSTRACT

This bachelor thesis deals with analysis of MIDI's protocol and especially by the implementation of synchronization methods. The aim of this thesis is suggestion and creating of a prototype of equipment, which transmits one sort of synchronization code and enables its direct extension on produced equipment.

First part of solution of this thesis deals with MIDI in general, its history, hardware definition and protocol analysis, especially by analysis of methods and by types of synchronization. The main attention is devoted to MIDI Click/SPP code, which is also implemented in created prototype.

The second part of this thesis deals with proposal of hardware solution of prototype, where is parsed a principle of involvement and used components in detail. The third part describes software proposal of solving from necessary initializations of microcontrollers up to operations with individual circumferences. There is also analyzed a customer service. Last, fourth part of thesis deals with realization of prototype, its revival and testing. There is stated value of properties of prototype, some suggestions for possible extension and summary of achieved aims of this thesis in the end.

**Keywords:** AVR, ATtiny2313, MIDI, MIDI Click, MIDI SPP, SysEx

## **BIBLIOGRAFICKÁ CITACE PRÁCE**

VÁCHA, L. *Generátor synchronizačního kódu pro hudební aplikace*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 38 s. Vedoucí bakalářské práce Ing. Jiří Schimmel, Ph.D.

### Prohlášení

Prohlašuji, že svou bakalářskou práci na téma *Generátor synchronizačního kódu pro hudební aplikace* jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne .....

.....

(podpis autora)

# PODĚKOVÁNÍ

Zde bych rád poděkoval svému vedoucímu bakalářské práce panu Ing. Jiřímu Schimmelovi, Ph.D. za jeho ochotu, cenné rady a odbornou pomoc při tvorbě bakalářské práce.

V Brně dne .....

.....

podpis autora

# OBSAH

Seznam obrázků	vii
Seznam tabulek	viii
Úvod	1
1 MIDI	2
1.1 Historie.....	2
1.2 Hardware.....	3
1.3 MIDI Protokol.....	5
1.4 Kanálová MIDI data .....	7
1.5 Systémová data .....	8
1.5.1 Společná systémová data .....	8
1.5.2 Systémová data reálného času .....	9
1.5.3 Zvláštní systémová data.....	10
1.5.4 Univerzální systémová data .....	11
1.6 Synchronizace .....	11
1.6.1 Taktová synchronizace .....	12
1.6.2 Taktová synchronizace - MIDI Clock / SPP.....	12
1.6.3 Časová synchronizace.....	15
2 Obvodový návrh zařízení	15
2.1 Popis mikrokontroléru .....	15
2.1.1 Periferie USART.....	16
2.1.2 Rozhraní USI .....	16
2.1.3 16bitový čítač/časovač.....	16
2.1.4 Vektory přerušení ATtiny2313 .....	17
2.1.5 Port PD.....	17
2.2 Obvodový popis .....	18
2.2.1 Rozhraní MIDI.....	19
2.2.2 Výběr oscilátoru a jeho vliv na chybu UART .....	20
2.3 Sériový programátor .....	20
2.4 Zapojení pinů ATtiny 2313.....	21

2.5	Zapojení generátoru do MIDI systému .....	22
3	Softwarový návrh Zařízení .....	23
3.1	Popis činnosti programu .....	23
3.2	Inicializace mikrokontroléru .....	23
3.2.1	Inicializace proměnných .....	23
3.2.2	Inicializace UART .....	23
3.2.3	Inicializace a obsluha 16bitového čítače .....	24
3.3	Obsluha sedmissegmentového displeje .....	25
3.4	Obsluha změny BPM .....	25
3.5	Obsluha přijetí UART a SysEx .....	25
3.6	Nastavení FUSES .....	26
4	Realizace Prototypu .....	27
5	Zhodnocení a možnosti rozšíření .....	28
6	Závěr .....	30
	Literatura .....	31
	Seznam veličin a zkratk .....	32
	Seznam příloh .....	33

## SEZNAM OBRÁZKŮ

Obr. 1.1: Schéma rozhraní MIDI [1]. .....	4
Obr. 1.2: Příklad datového MIDI slova. ....	5
Obr. 1.3: Struktura datového a stavového bytu [4]. ....	6
Obr. 1.4: Struktura SysEx rámce [4]. ....	11
Obr. 1.5: Chování vysílače a přijímače při synchronizaci kódu MIDI Clock / SPP [4].	14
Obr. 2.1: Zapojení generátoru v MIDI systému. ....	22
Obr. 3.1: Struktura SysEx rámce pro generátor MIDI Click. ....	26
Obr. 3.2: Ukázka nastavení pojistek - Fuses. ....	26
Obr. 4.1 Prototyp generátoru MIDI Click / SPP. ....	28

## SEZNAM TABULEK

Tab. 1.1: Kanálová MIDI data [4]. .....	7
Tab. 1.2: Systémové MIDI data [4]. .....	8
Tab. 1.3: Distribuce identifikátorů výrobce [4]. .....	10
Tab. 2.1: Vektory přerušení ATtiny2313 [7]. .....	17
Tab. 2.2: ATtiny zapojení pinů. ....	21

# ÚVOD

Rozhraní MIDI (Musical Instruments Digital Interface) je dnes zřejmě jedno z nejpoužívanějších rozhraní pro komunikaci mezi hudebními nástroji. Protokol MIDI je i po jeho již 25 letech existence neustále ve vývoji, i když již přestává splňovat především rychlostní požadavky pro dnešní moderní studiovou techniku. Jako jediný standart je uznáván a dodnes implementován všemi výrobci klávesových nástrojů, zvukových a efektových modulů.

Od počátku digitalizace hudebních nástrojů vedlo úsilí k propojení více nástrojů za účelem znásobit paralelním propojením množství zvukových barev, které zní současně. Taktéž vznikala potřeba automatizace, která by omezila počet lidských úkonů při samotném živém hraní ale také při studiové tvorbě. MIDI taktéž umožnilo zadávání a editaci hudebních dat mimo reálný čas, ale také vzdálenou kontrolu parametrů nástroje.

S těmito požadavky však přišel problém synchronizace. Pro synchronizaci MIDI systémů bylo vyvinuto mnoho druhů synchronizačních kódů. Skrze tyto konkrétní synchronizační signály či jejich kombinacemi se synchronizují všechny přístroje, které jsou zapojeny v MIDI systému.

V dnešní době je dostupných mnoho různých MIDI kontrolérů umožňující vysílání synchronizačního kódu MIDI Clock /SPP. Či dokonce generátorů, které vysílají jen synchronizační kódy ať už MIDI Clock /SPP, MTC či SMPTE a další. Většinou to jsou ale velké a těžké rackové přístroje určené spíše do studií. Taktéž se jedná o zařízení velmi komplexní, sdružující mnoho funkcí, což jejich cenu jen navyšuje. Na trhu není dostupný žádný přístroj, který by generoval pouze synchronizační kód MIDI Click. Myšlenkou mého projektu je tedy navrhnout generátor synchronizačního kódu MIDI Clock / SPP, který by byl cenově dostupný a v kompaktních rozměrech.

# 1 MIDI

## 1.1 Historie

Pomineme-li předešlá analogová komunikační rozhraní, která byla založena na principu napěťového řízení, tak na počátku 70. let minulého století vznikala potřeba vzájemně propojit v té době rychle se rozvíjející a více dostupné syntezátory standardním digitálním komunikačním rozhraním. Data přenášená po digitálním rozhraní tedy nejsou audio signálem, ale pouze řídicí data.

Na počátku byly zaváděny provozy založené na paralelní komunikaci, ale nedošlo k jejich rozšíření. Na vině byl především drahý hardware (například až 24-vodičové propojovací kabely). Právě v 80. letech vznikaly první nenáročné systémy založené na sériové komunikaci. K nižší ceně taktéž napomohl rozmach a aplikace mikroprocesorové techniky. Výrobci se drželi zavedených standardů pro sériová rozhraní typu RS-232 nebo IEEE-488 (Digital Keyboards) nebo používali své vlastní (Sequential Circuits, Oberheim) [4].

Komerčně nejuspěšnějším se s postupem času stal systém navržený a realizovaný firmou Roland nazvaný DCB (Digital Communication Bus) neboli digitální komunikační sběrnice. Tento systém implementoval do svých nástrojů Jupiter JP-8, Juno JU-6, Juno JU-60 a v několika DCB rekordérech. Přístroje byly propojeny jedním jednosměrným čtyř žilovým (H172) nebo obousměrný sedmi žilový s 14pinovým konektorem (H165) kabelem. Rychlost tohoto rozhraní byla 31250 Baudů a vysílala se ve tvaru: startbit, 8 databitů, lichou paritou a jedním stopbitem. Protokol DCB taktéž začal rozlišovat datové a stavové byty (identifikátory). Pro stavové byty bylo vyhrazeno 15 kódů (\$F1 až \$FF), pro datové pak \$00 až \$F0. Ze stavových bytů se však využívaly jen tři, a to tyto [1], [4]:

- a) Patch Code \$FD - přepnutí rejstříku, následován jedním databytem, Patch code se však používal pouze pro nástroje JP-8 a OP-8.
- b) Key Code \$FE - data o stisknutých klávesách, následovaná různým počtem databytů podle hlasových jednotek nástroje. Nejvyšší bit určoval zapnutí / vypnutí klávesy (Gate On / Off), dalších sedm bitů určovalo číslo klávesy.
- c) End Mark \$FF - označoval konec datového bloku, který neměl pevně dán počet data bytů nebo pokud nebyl ukončen novým identifikátorem (poslední vyslaný datový blok)

I přes určité nedostatky a problémy kterým rozhraní DCB trpělo, se tento systém stal základem pro MIDI. Na červnové mezinárodní výstavě NAMM roku 1981 v Anaheimu, USA se sešli poprvé prezidenti firem Sequential Circuits, Oberheim

a Roland aby projednali a vypracovali předběžný návrh na univerzální rozhraní hudebních nástrojů. Později se ke spolupráci taktéž připojily firmy Yamaha, Korg a Kawai. Už v říjnu téhož roku předvedli Dave Smith a Chet Wood na konferenci Audio Engineering Society v New Yorku první ucelený návrh, takzvaný USI. O tento interface však nebyl očekávaný zájem, proto Dave Smith začal shromažďovat připomínky a další návrhy různých výrobců.

O rok později (1982) se opět sešli představitelé hudebních firem na výstavě NAMM a ke konceptu USI byly přidány další úpravy a vylepšení. V projektu už bylo zapojeno celkem 15 amerických a japonských firem, firma Roland navrhla pro budoucí rozhraní název MIDI - Musical Instruments Digital Interface neboli digitální rozhraní hudebních nástrojů. V lednu roku 1983 na trh přišli první syntetizátory s MIDI rozhraním, a to SCI Proplet 600 a Roland Jupiter JP-6. Tyto nástroje si však vyměňovaly jen informace Note On / Note Off.

První verze MIDI normy byla nabídnuta výrobcům v podobě MIDI 1.0 Specification dne 5. 8. 1983. Byl to dokument o 15 stranách, avšak byl v konkrétních detailech velmi nejasný a tím vyvstaly problémy s kompatibilitou. Z tohoto důvodu byly v roce 1984 založeny normativní orgány odpovědné za dodržování normy MIDI a jejího dalšího vývoje. Byla to celosvětová MMA (MIDI Manufacturers Association) a JMSC (Japan MIDI Standard Committee). Členy těchto organizací jsou dnes firmy využívající MIDI rozhraní, vyrábějící audiotechniku a hudební elektroniku.

V září 1985 vyšla podrobná MIDI norma, která vyřešila mnohé nedostatky s kompatibilitou, ale vývoj MIDI si postupem času vynutil celou řadu dodatků a doplňků. Již v lednu roku 1986 byl vytvořen dodatek k používání kontrolérů a zvláštních systémových dat, taktéž zavedl MTC - MIDI Time Code. MIDI se i nadále neustále rozvíjí, avšak už díky svému zastaralému principu a také z rychlostních důvodů přestává splňovat požadavky kladené dnešní moderní studiovou a hudební elektronikou. [1], [2], [4], [6]

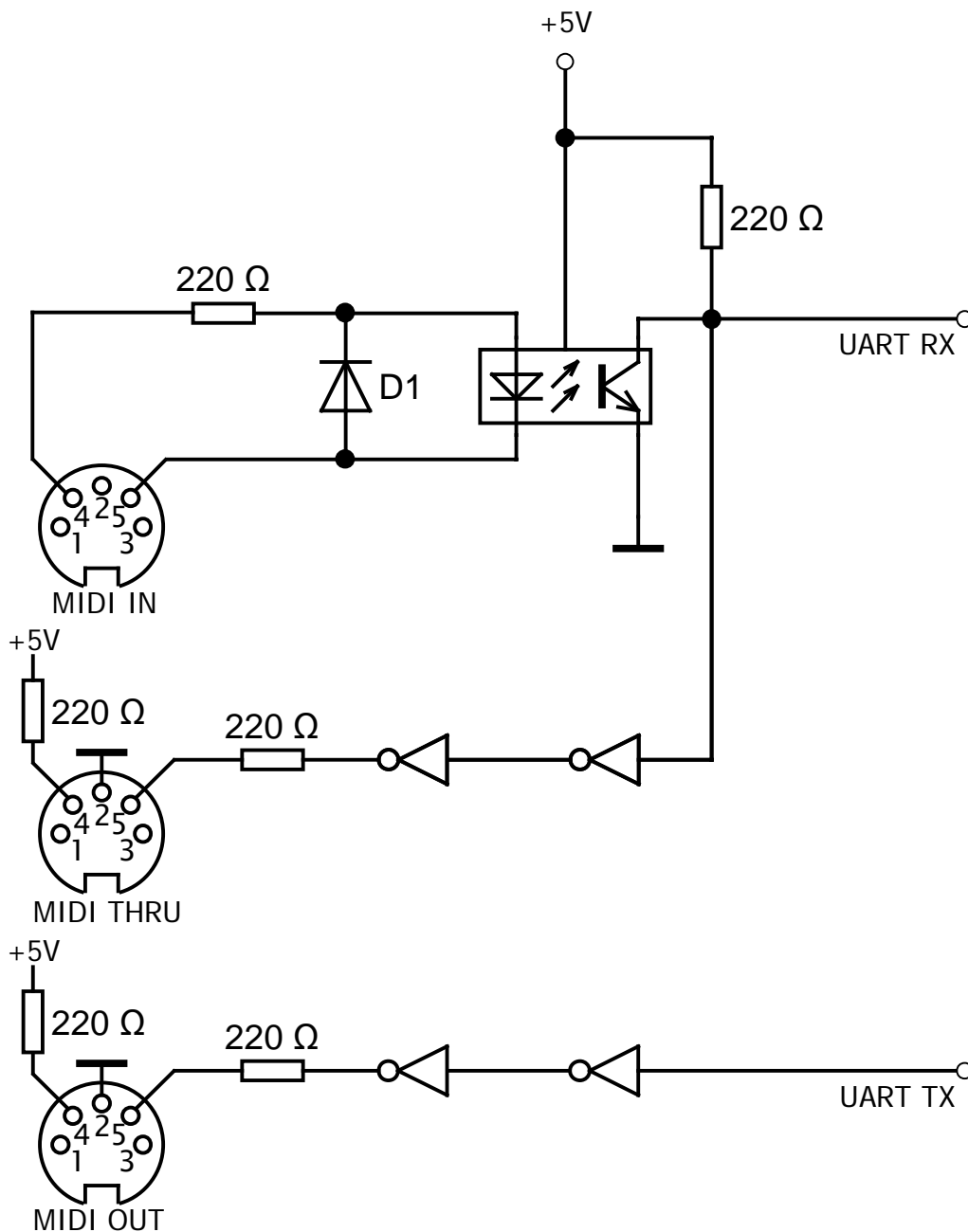
## 1.2 Hardware

MIDI sběrnice je proudová smyčka se jmenovitým proudem 5 mA, logické hodnotě nula odpovídá protékající proud, z toho vyplývá nulová hodnota protékajícího proudu pro logickou jedna.

Jelikož je MIDI sériový jednosměrný protokol, potřebuje mít oddělený vstup a výstup. Rozhraní MIDI je tvořeno trojicí 5 pinových konektorů, typu DIN 180° označených In, Out a Thru:

- a) In - vstupní, který přijímá do zařízení signály z vnějšího zdroje,
- b) Out - výstupní, ze kterého jsou odesílány signály generované zařízením,
- c) Thru - výstupní, propojený s konektorem In, umožňuje odesílání nezměněných

dat ze vstupu zařízení.



Obr. 1.1: Schéma rozhraní MIDI [1].

Konektor Thru nemusí být vždy použit, některá zařízení mají implementovanou funkci Soft Thru, která slučuje vstupní data s interně generovanými daty.

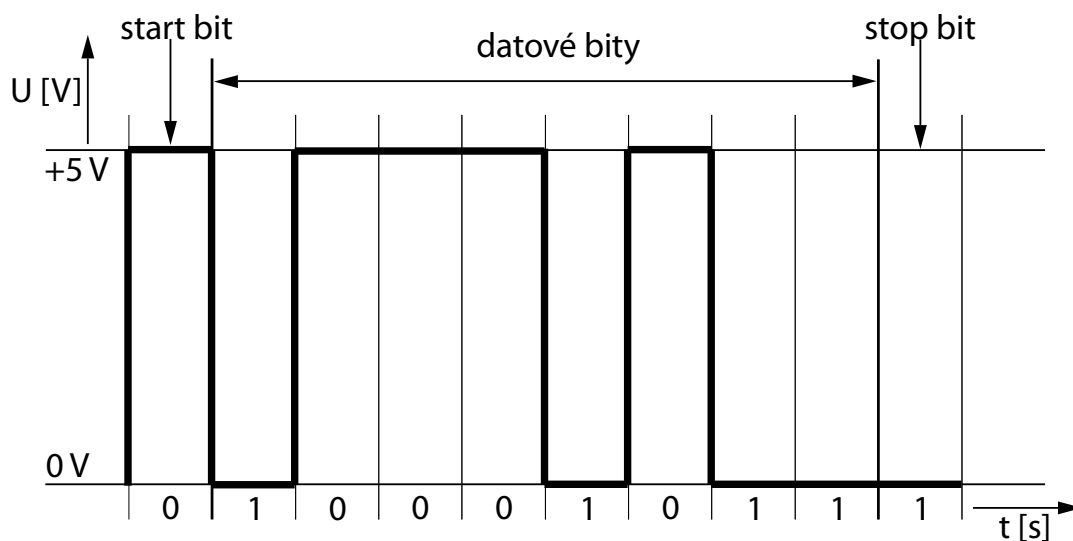
Z důvodu zemních smyček mezi jednotlivými přístroji se nesmí země na konektoru MIDI In připojovat k zemním svorkám přístroje. Z tohoto důvodu je navíc na MIDI In umístěn optoizolátor galvanicky oddělující vstup od vlastního přístroje. Optoizolátor na obr. 1.1, nebo také optočlen, který by neměl mít dle normy dobu reakce vyšší než 2  $\mu$ s. Z důvodu reakční doby optoizolátoru a také díky zpoždění způsobené reakční dobou

invertorů je připojování více přístrojů do řetězce přes MIDI Thru omezeno dle normy na čtyři. Při vyšším počtu by docházelo k velkému zpoždění, které by mohlo vést k nežádoucímu chování MIDI systému. Pro rozsáhlé MIDI systémy se proto využívá přídatného zařízení MIDI Thru Box, který obsahuje několik MIDI In konektorů a hlavně také MIDI Out konektory. Princip tohoto zařízení spočívá v přijímání dat ze všech MIDI In konektorů a následném vyslání na všechny MIDI Out konektory.

Propojovací MIDI kabely by neměly být delší než 15 metrů [1], avšak ze zkušeností vyplývá, že maximální délka MIDI kabelu může být jen 5 metrů. V současné době se využívají převodníky USB/MIDI pro propojení s PC, z důvodu absence MIDI portu na moderních počítačích. Dalšími alternativami pro přenos MIDI je například využití lokální sítě Ethernet nebo rozhraní IEEE-1394. Tato rozhraní už jsou i standardizována organizací MMA jako dodatky k MIDI normě 1.1. [1], [2], [4], [6]

### 1.3 MIDI Protokol

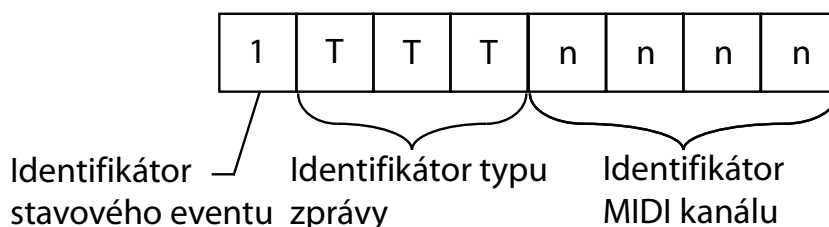
Po MIDI rozhraní se přenášejí jednotlivé datové bloky, které mají přesně daný tvar, tento blok se nazývá MIDI zpráva (MIDI message), nebo také MIDI událost (MIDI event). Používá se sériový asynchronní datový přenos s rychlostí 31250 Baudů ( $\pm 1\%$ ). Rámec je tvořen jedním start bitem (logická 1), osmi datovými bity a jedním stopbitem (logická 0), bez parity, viz obr. 1.2. Start a stop bit slouží k synchronizaci přijímače. Doba trvání rámce je 320  $\mu\text{s}$ , LSB bity se vysílají jako první. Samotná komunikace je založena na jednotlivých zprávách, které mají jedno nebo více datových slov (více bytů). Jako první byte je vždy stavový a za ním následují byty datové. Nejvýznamnější bit určuje, zda jde o stavový byte (MSB=1) nebo datový byte (MSB=0). Na obrázku obr. 1.3 je struktura stavového a datového bytu [1], [4].



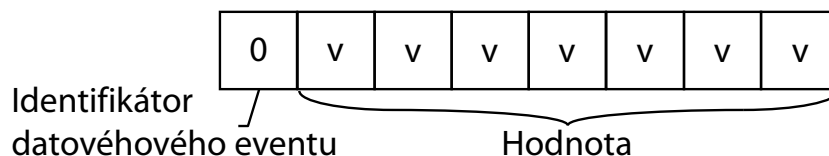
Obr. 1.2: Příklad datového MIDI slova.

MIDI právy se dělí na kanálové zprávy (Channel Message) a systémové zprávy (System Message). Dále se kanálová data dělí na hlasová data (Cannel Voice Message) a data režimu (Channel Mode Message), které přenášejí ve svém stavovém bytu informaci o virtuálním datovém kanále. Ve stavovém bytu jsou pro identifikaci kanálu vyhrazeny čtyři bity, viz obr. 1.3. Z tohoto vyplývá, že po jedné, fyzické MIDI sběrnici jsou přenášeny kanálové zprávy v 16 ( $2^4$ ) virtuálních kanálech. Systémové zprávy informaci o MIDI kanále nepřenáší, jsou totiž společné pro všechny kanály. Dolní čtyři bity stavového bytu slouží k identifikaci typu systémové zprávy, která se dělí na zvláštní systémová data (System Exclucive Mesage), která umožňují přenos větších specifických datových bloků. Dále to mohou být společná systémová data (System Common Message) a data reálného času (System Real Time Message), která slouží k vzájemné časové synchronizaci zařízení v MIDI systému. Počet datových bytů závisí na typu MIDI zprávy, neboli na stavovém bytu [1], [4].

Stavový byte:



Datový byte:



Obr. 1.3: Struktura datového a stavového bytu [4].

Pro zvýšení průchodnosti MIDI sběrnice je zaveden režim průběžný stav (Running Status), který definuje, že v případě kanálových dat není nutné vždy vysílat úplné MIDI zprávy. Pokud se totiž nezmění typ informace určený stavovým bytem (např. stavové slovo, číslo kontroléru), můžeme dále vysílat jenom další databyty. Za kompletní zprávu lze považovat pouze odpovídající počet data bytů. Tímto způsobem se dá ušetřit asi třetina kapacity sběrnice. Tento stav platí do té doby, než je přijat jiný stavový byte [4].

Jakýkoliv stavový byte, který není definován či implementován, musí nástroj komunikující pomocí MIDI rozhraní ignorovat, stejně tak k němu příslušející datové byty. Pokud není sběrnice v průběžném stavu, musí ignorovat datové byty bez příslušejícího stavového bytu. Priorita dat v MIDI systému [4]:

1. resetování systému,
2. data reálného času,
3. zvláštní systémová data,
4. společná systémová data,
5. kanálová data.

## 1.4 Kanálová MIDI data

Podle svého názvu se jedná o data, která se vztahují pouze k určitému MIDI kanálu, jehož číslo je zakódováno a přenášeno v dolních čtyřech bitech stavového bytu. Čtyřmi bity lze vyjádřit 16 čísel, z toho vyplývá možnost adresovat 16 MIDI kanálů. Pro identifikátor kanálových dat jsou ve stavovém bytu vyhrazeny tři bity, k dispozici je tedy sedm identifikátorů, osmý je vyhrazen pro systémová data. Přehled kanálových dat je uveden v tabulce tab. 1.1.

Tab. 1.1: Kanálová MIDI data [4].

Tvar stavového	ID	Počet data	MIDI zpráva	Význam
\$8n	0	2	Note Off	Nota vypnuta
\$9n	1	2	Note On	Nota zapnuta
\$An	2	2	Polyphonic Key	Individuální tlaková
\$Bn	3	2	Controllers Change	Změna kontroléru
\$Cn	4	1	Program Change	Změna programu
\$Dn	5	1	Channel Pressure	Společná tlaková citlivost
\$En	6	2	Pitch Bend Change	Ohýbání tónu

Kde n označuje čtyř bitové číslo udávající číslo MIDI kanálu

Kanálová data jsou přijímána různě, přijímač MIDI může pracovat v jednom ze čtyř režimů, které určují, jak jsou kanálová MIDI data zpracována. Tyto čtyři režimy se rozlišují podle nastavení parametrů OMNI ON/OFF a MONO/POLY. A to v těchto kombinacích režimů:

1. Omni On, Poly,
2. Omni On, Mono,
3. Omni Off, Poly,
4. Omni Off, Mono.

Parametr MONO/POLY definují přiřazení jednotlivých hlasů přijímače při přijetí více not. V režimu MONO hraje každý hlas přijímače monofonně noty přijaté na patřičném kanálu. V režimu POLY hrají hlasy přijímače přijaté noty polyfonně. Parametr OMNI definuje, jestli bude nástroj přijímat data ze všech kanálů, (ON) anebo

pouze na zvoleném kanále (OFF). Ve výchozím nastavení každého nástroje by měl být zapnutý režim OMNI a POLY, pokud se ovšem jedná o vícehlasý nástroj [1], [4].

## 1.5 Systémová data

Jedná se o data, ve kterých se neuvádí číslo MIDI kanálu, protože jsou společná pro všechny kanály v systému, viz tab. 1.2. Všechny typy systémových dat mají horní čtyři bity (horní nibbl) stavového bytu s hodnotou \$F. Zbylé 4 bity (dolní nibble) můžou určovat až 16 možných typů systémových dat, z nichž je zatím však definováno celkem 12 systémových zpráv, zbylé čtyři nejsou definovány.

Tab. 1.2: Systémové MIDI data [4].

ID	Název	Počet data	Význam
\$F0	System Exclusive	neomezen	Začátek zvláštních systémových dat
\$F1	MTC Quarter Frame	1	Čtvrtinový rámec synchronizace MIDI Time
\$F2	Song Position Pointer	2	Ukazatel pozice skladby
\$F3	Song Select	1	Volba skladby
\$F4	nedefinováno	-	
\$F5	nedefinováno	-	
\$F6	Tune Request	0	Žádost o naladění
\$F7	End of System Exclusive	0	Konec zvláštních systémových dat
\$F8	Timing Clock	0	Časovací impuls synchronizace MIDI Clock
\$F9	nedefinováno	-	
\$FA	Start	0	Příkaz Start synchronizace MIDI Clock
\$FB	Continue	0	Příkaz Continue synchronizace MIDI Clock
\$FC	Stop	0	Příkaz Stop synchronizace MIDI Clock
\$FD	nedefinováno	-	
\$FE	Active Sensing	0	Aktivní vnímání
\$FF	System reset	0	Reset systému

### 1.5.1 Společná systémová data

Do společných, nebo také obecných systémových dat, která jsou určena pro všechna zařízení v MIDI systému, patří ukazatel pozice skladby (Song Position Pointer), synchronizační značka MTC Quarter Frame, volba skladby (Song Select), žádost o naladění nástroje (Tune request), a konec zvláštních systémových dat (End Of System Exclusive - EOX). Taktéž musí být tato data standardizována MIDI normou, protože jsou společná pro všechna MIDI zařízení.

1. Ukazatel pozice skladby - viz kapitola 1.6.2.
2. MTC Quarter Frame, se používá pro synchronizaci MIDI zařízení s audio

a video technikou. Kód MIDI Time Clock (MTC), vychází z obecně používaného kódu SMPTE/EBU. MTC používá dva základní typy zpráv, a to MTC Quarter Frame a MTC Full Frame. MTC Quarter Frame má definován svůj vlastní stavový byte \$F1, který ve čtyřech dvojicích MIDI zpráv přeneše informaci o čase, a to ve tvaru hodiny, minuty, sekundy a rámce. MTC Full Frame patří do zvláštních systémových dat a zvládne přenést celou informaci o čase v jedné MIDI zprávě.

3. Volba skladby - Song Select (\$F3) se používá výhradně v sekvencerech a slouží pro zvolení skladby, která má být přehrávána. Z master sekvenceru se vyšle požadavek na změnu skladby a ostatní ji zpracují. Patří k ní jeden databyte, z čehož plyne 128 možností pro volbu skladby.

4. Žádost o naladění - Tune request (\$F6), pozůstatek z dob analogových syntezátorů s číslicovým řízením. Při přijetí této zprávy se provedla kontrola naladění oscilátorů, byla-li implementována.

5. Konec zvláštních systémových dat - End Of System Exclusive (\$F7), určuje konec bloku SysEx zpráv, tvoří dvojici s MIDI zprávou SysEx.

### 1.5.2 Systémová data reálného času

Tato data nenesou žádnou časovou informaci, jejich princip spočívá v přijetí ve správný okamžik, neboli v přesně definované časové pozici. Z tohoto důvodu mají nad všemi ostatními MIDI zprávami prioritu, a proto musí být okamžitě po přijetí zpracovány MIDI přijímačem.

1. Synchronizační kód MIDI Clock - viz kapitola 1.6.2.

2. Aktivní vnímání, Active Sensing (\$FE) je volitelné. Tento stavový byte je vyslán maximálně každých 300 ms, pokud se nevysílají jiné MIDI zprávy. Po přijmutí Active Sensing zařízení předpokládá každých 300 ms přichozí MIDI zprávu. Pokud však během této doby nepřijde žádná zpráva, zařízení předpokládá ztrátu spojení, a proto vypne všechny hlasy a vrátí se do počátečních podmínek. MIDI norma však doporučuje, vysílání aktivního vysílání v intervalech do 270 ms a přijímače aby rozhodly o přerušení spojení až po 330 ms [4].

3. Resetování systému (System reset), má hodnotu \$FF, po přijetí by měly všechny nástroje v MIDI systému uvedeny do výchozího stavu a provést následující operace [4]:

- 1) zapnout režim Omni a Poly (zda jsou implementovány),
- 2) zapnout lokální řízení,
- 3) vypnout všechny hlasy,
- 4) všechny kontroléry uvést do výchozího stavu,
- 5) nastavení pozice skladby na pozici 0,
- 6) zastavit přehrávání,

- 7) zrušit průběžný režim,
- 8) nástroj nastavit do počátečních podmínek.

### 1.5.3 Zvláštní systémová data

Zvláštní systémová data (System Exclusive, zkráceně SysEx) jsou nejuniverzálnější zprávy v celém MIDI systému, které jsou určeny pro specifické potřeby daného zařízení, či skupiny zařízení. Jejich využití a formát záleží zcela na výrobci daného zařízení. Blok SysEx zpráv má pevně danou strukturu, která musí být dodržena. SysEx začíná vždy stavovým bytem \$F0, následuje záhlaví SysEx dat (viz dále) a libovolný počet databytů, ukončený stavovým bytem \$F7 (End Of SysEx). Blok SysEx dat nesmí být přerušen žádnou jinou MIDI zprávou s výjimkou zpráv reálného času, přijetí jakéhokoliv jiného stavového bytu je považován za ukončení bloku SysEx.

Protože se v MIDI systému může vyskytovat více MIDI zařízení, je nutné zavést identifikaci jednotlivých zařízení. Proto MIDI norma definuje záhlaví SysEx dat, které identifikuje zařízení podle dvou identifikátorů [4]:

1. ID Number je číslo identifikující výrobce zařízení, aby se předešlo konfliktům, jsou tato čísla přidělována a spravována organizacemi MMA a JMSC. Původně jednobytový identifikátor byl rozšířen na tříbytový. Pokud zařízení při přijetí rozpozná na místě identifikátoru hodnotu \$00, znamená to použití tříbytového identifikátoru a přečte tedy dva následující byte jako identifikátor výrobce. Identifikátor \$7D je určen pro nekomerční účely, \$7E a \$7F pro tzv. univerzální zvláštní systémová data. V Tab. 1.3 je uveden systém přidělování identifikátorů pro výrobce dle geografického rozprostření.

Tab. 1.3: Distribuce identifikátorů výrobce [4].

	Americká	Evropská	Japonská	Ostatní	Speciální
1 bytový ID	01 - 1F	20 - 3F	40 - 5F	60 - 7C	7D - 7F
3 bytový ID	00 00 01	00 20 00	00 40 00	00 60 00	-
	00 1F 7F	00 3F 7F	00 5F 7F	00 7F 7F	-

2. Device ID, tento identifikátor rozlišuje více zařízení stejného výrobce zapojených v jednom MIDI systému. Protože má podobnou funkci jako MIDI kanál, je také označován jako subkanál, avšak s rozsahem 0 až 127. Na MIDI zařízeních lze tento identifikátor uživatelsky nastavit, nebo je odvozen od čísla základního MIDI kanálu [4]. Dle normy je možno ještě rozšířit tento identifikátor o další dva datové byty, sub-ID#1 a sub-ID#2, viz Obr. 1.4. V tomto případě se jsou většinou dva byty vyhrazeny pro identifikátor výrobku a jeden pro subkanál. Hodnota \$7F je určena jako identifikátor pro všechna zařízení.

\$F0	ID	Device ID	sub-ID#1	sub-ID#2	----- datablock -----	checksum	\$F
------	----	-----------	----------	----------	-----------------------	----------	-----

Obr. 1.4: Struktura SysEx rámce [4].

### 1.5.4 Univerzální systémová data

Poskytují ještě větší aplikační možnosti než SysEx. Obdobně jako pro SysEx je definováno jen jejich záhlaví, a obsah je již jen na výrobci zařízení. V každé nové verzi MIDI specifikace se objevují nové standardizované aplikace zvláštních systémových dat. V podstatě by se dali rozdělit na univerzální SysEx data mimo reálný čas (identifikátor \$7E) a na univerzální SysEx data v reálném čase (identifikátor \$7F). V současnosti norma například obsahuje tyto standardy [4]:

- Přenos zvukových nahrávek - Sample Dump,
- Přenos souborů - File Dump,
- Mikroladění nástrojů - MIDI tuning,
- Řízení scénické techniky - MIDI Show Control,
- Řízení studiové techniky - MIDI Machine Control,
- Ovládání záznamových zařízení - Real Time Cueing,
- Notační informace - Notation Information,
- Zprávy systému General MIDI.

## 1.6 Synchronizace

Synchronizace zaručuje pro různá zařízení v jednom systému souběžný běh. To však nemusí znamenat, že zařízení začnou přehrávat na absolutně stejné pozici a s absolutně stejnou rychlostí. To by vyžadovalo nekonečnou přesnost, čehož nelze v praxi dosáhnout. Synchronizace běžících zařízení v systému spíše spočívá v zajištění spojení definovaných bodů na jejich časových osách [4].

Pro synchronizaci je nutné, aby vždy jedno zařízení sloužilo jako řídicí (master). Master zařízení určuje informaci o aktuální pozici, přičemž všechna ostatní zařízení připojená k němu jsou jím řízena (slave). Podřízená zařízení trvale sledují aktuální pozici, s přesností jak jsou jen schopné. Dle použitých zařízení lze dělit dva typy synchronizace:

1. V zařízení jako bicí automaty či sekvencery, je informace o pozici vyjádřena v taktech, neboli používají taktovou synchronizaci (viz kapitola 1.6.1). Protože se tempo a rytmus během skladby může měnit, jsou jednotlivé doby jedinými přesnými zachytnými body.
2. V zařízeních pro přehrávání analogového či digitálního zvukového signálu, jako

např. systémy HDR, páskové magnetofony nebo páskové magnetofony pro video signál (VCR), je obvykle používána Časová synchronizace (viz kapitola 1.6.31.6.1). Informace o pozici je udávána ve formě časové informace, která je nazývána časovým kódem (Time Code). Tento druh synchronizace není závislý na rychlosti posuvu pásku či kmitočtu, vůči tempu skladby. Časové kódy se dají dále dělit podle toho, zda obsahují konkrétní pozici (časovou nebo taktovou) na:

- a) Relativní časové kódy - které neobsahují informaci o aktuální pozici, je tedy nutné je vždy spouštět od začátku.
- b) Absolutní časové kódy - které obsahují informaci o pozici, a lze je tedy spouštět z libovolného místa.

### **1.6.1 Taktová synchronizace**

Neboli také označována jako přímá synchronizace, je založena na použití relativního časového kódu. Jehož hustota se mění s tempem skladby. Nejsou-li doplněny navíc informací o pozici skladby (SPP), je vždy nutné startovat synchronizaci všech zařízení od začátku [1].

1. Pulse Clock - jedná se o původní metodu synchronizace analogových sekvencí nebo bicích automatů. Je založena na periodickém vysílání elektrického impulsu (obdoba hodinového signálu), dle normy v rychlosti 24 PPQN (1/96 noty). Dříve někteří výrobci používali rychlosti 48, 64, 384 PPQN. Zdokonalené kódy tohoto typu jako (Sync, DIN Sync). Měli taktéž navíc signál pro Start/Stop přehrávání, a signál Fill In, neboli jednotaktový hudební přechod.

2. FSK - Zakódování impulsů Pulse Clock kmitočtovým klíčováním mezi dvěma frekvencemi. Přičemž se používalo harmonického nosného signálu o kmitočtu 1 kHz a 2 kHz. Byl nahráván na magnetický pásek jako zvukový signál, a používal se především k synchronizaci bicích automatů spolu s páskovými magnetofony. FSK patří taktéž do relativních kódů. Proto pokud byl pásek poškozen, došlo vynecháním několika pulsů ke ztrátě synchronizace. Toto napravily až zdokonalené verze FSK2, Smart FSK a Tape Sync II. Tím že zakódovaly navíc i informaci o pozici ve skladbě.

### **1.6.2 Taktová synchronizace - MIDI Clock / SPP**

Při příchodu MIDI standardu byl zahrnut v protokolu jako ekvivalent hodinových impulsů, a to jako synchronizační kód MIDI Clock. Má vyhrazeny hned čtyři ze šesti zpráv dat reálného času, viz Tab. 1.3.

Jedná se o jednoduchý relativní časový kód, který přenese puls 24krát během jedné čtvrté doby, což znamená s rozlišením 24 PPQN, neboli 1/96 noty. Tímto pulsem je zpráva MIDI Timing Clock (\$F8) která nemá žádný databyte. Taktéž bez databytů jsou další používané zprávy tohoto synchronizačního kódu, a to Start (\$FA), Continue (\$FB) a Stop (\$FC). Protože však sekvencery používají časové rozlišení 960 PPQN a vyšší, je

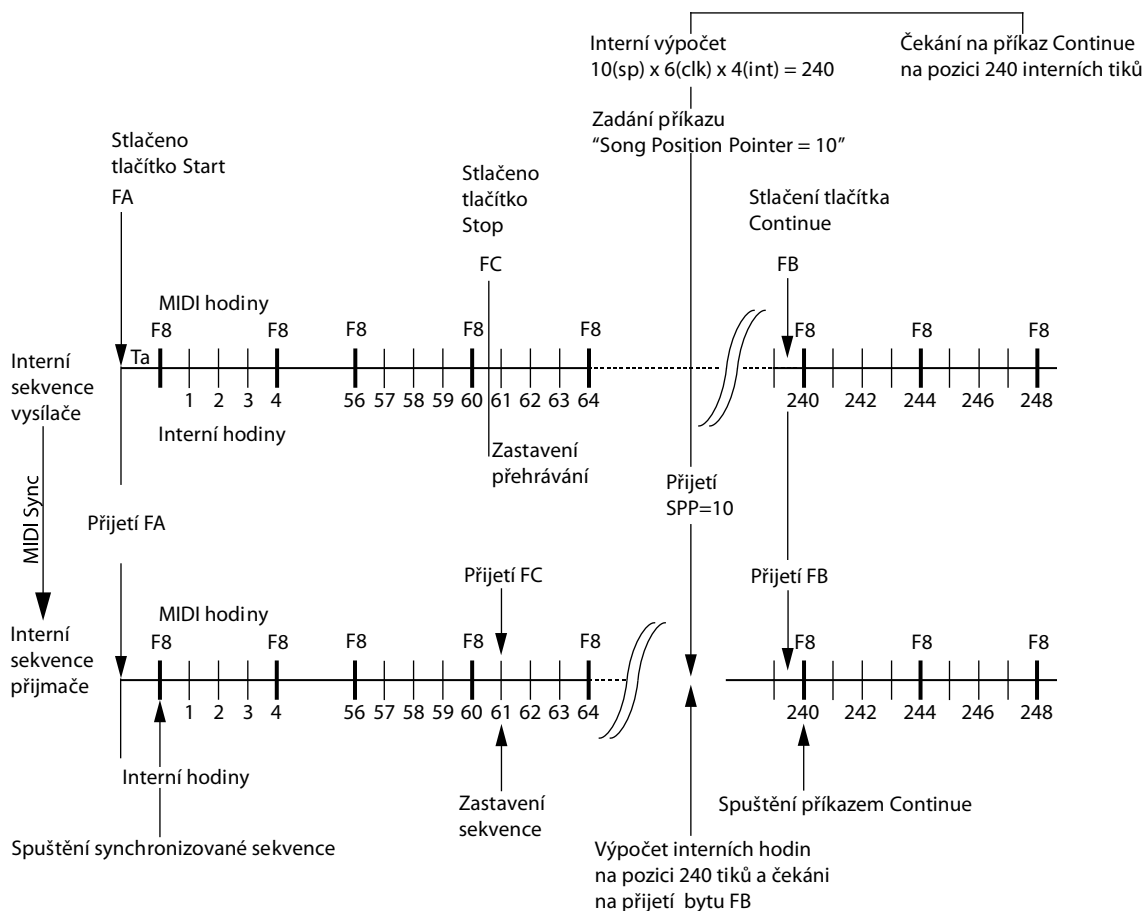
nutné, aby si časové pozice mezi synchronizačními značkami Timing Clock interpolovaly. Povel Start, Stop a Continue slouží pro spuštění a zastavení přehrávání. Dle normy by mělo master zařízení vysílat zprávy Timing Clock i při zastaveném přehrávání. Toto slouží pro synchronizaci zařízení podřízených, aby měla zasynchronizován interní časový generátor, a tak mohla po přijetí zprávy Start či Continue nastavit správné tempo přehrávání.

Song Position Pointer SPP (\$F2): tento stavový byte určuje, kolik taktů od začátku skladby je aktuální pozice přehrávání. Za tímto stavovým bytem jsou dva následující databyty, které udávají konkrétní počet šestnáctinových not od počátku přehrávání. Z použití dvou data bytů plyne maximální hodnota 16384 ( $2^{14}$ ) šestnáctinových not. Ve 4/4 metru to je 1024 taktů, neboli cca 34 minut při tempu 120 BPM, což pro běžné účely zcela dostačuje.

Informace o pozici ukazatele skladby se vysílá pouze při skokové změně ukazatele, a to při přehrávání smyčky či při spuštění přehrávání z jiného místa než ze začátku skladby. Může být také vyslána při přetáčení skladby, neboli nastane-li rychlá změna pozice skladby při zastaveném přehrávání. Při kontinuálním přehrávání se počítá aktuální pozice skladby podle příchozích synchronizačních značek MIDI Click.

Samotný lokátor je vnitřní registr v sekvencerech, který počítá počet šestnáctinových dob od začátku spuštěné skladby. Chování přijímače a vysílače synchronizačního kódu MIDI Click / SPP je znázorněno na Obr. 1.5. Obě zařízení používají interní rozlišení 96 PPQN a proto jsou intervaly zpráv Timing Clock interpolovány 4krát.

Pokud je v MIDI systému více záznamových zařízení, tzv. MIDI sekvencerů, musí vždy být jeden nastaven jako řídicí (Master) a ostatní jako podřízené (Slave). Při přetáčení / spuštění přehrávání vysílá řídicí sekvencer MIDI zprávu SPP a řízené sekvencery si podle ní upraví svoji aktuální pozici skladby.



Obr. 1.5: Chování vysílače a přijímače při synchronizaci kódu MIDI Clock / SPP [4].

Po odeslání zprávy Start, musí čekat vysílací zařízení minimálně dobu intervalu  $Ta = 1$  ms z důvodu poskytnutí přijímacímu zařízení čas pro nachystání přehrávání (např. přesun interního ukazatele na začátek skladby). Poté začne vysílací zařízení vysílat synchronizační zprávy Timing Clock dle svého nastaveného tempa. Na Obr. 1.5 je znázorněna situace kdy je po 60. vyslaném pulsu Timing Clock vyslána zpráva Stop, následována zprávou SPP a Continue. Řízené zařízení po přijetí zprávy Stop ukončí přehrávání na následujícím bodu své interní časové osy (pozice 61).

Poté uživatel posune na řídicím zařízení pozici skladby a řídicí zařízení vyšle zprávu SPP s hodnotou 10. Protože je  $1/16$  nota 4krát menší, než interval  $1/4$  noty, odpovídá intervalu 4 PPQN. Hodnota desíti  $1/16$  not od začátku skladby proto odpovídá 60 intervalům kódu MIDI Clock a 240 intervalům interního časového generátoru sekvenceru s rozlišením 96 PPQN. Řízené zařízení tedy po přijetí této zprávy SPP provede přesun svého interního ukazatele na pozici skladby 240. Po vyslání či spíše přijetí příkazu Continue pokračují obě zařízení v přehrávání od své interní pozice 240. Při vyslání příkazu Start namísto Continue, by došlo k vynulování obou interních ukazatelů a přehrávání by začalo od začátku skladby [1], [2], [4].

### 1.6.3 Časová synchronizace

Je odvozena z rychlosti střídání obrazových políček nebo také rámců (frame), filmového pásu. Proto nemá sekundy rozdělené na desetiny a setiny, ale na rámce. Dříve byl jeden rámeček doba, za kterou prošlo jedno okénko filmu kamerou či projektorem. Kromě video signálu se ale taktéž používá pro zvukové signály. Počet rámců za sekundu se však dle různých norem a použití liší. Pro film se stalo mezinárodní normou 24 rámců za sekundu (fps). V USA se však původní černobílá televize vysílala s 30 fps. Po zavedení barevné TV se používalo 29,97 fps, a to kvůli kompatibilitě s původní černobílou TV. V Evropě se však již od počátku používala 25 fps, a po zavedení barevné televize se stala tato rychlost součástí standardu PAL. Důvod k odlišným hodnotám fps byl způsoben použitím odlišných frekvencí proudu v rozvodných sítích (Evropa 50 Hz, USA 60 Hz), z které si v té době televizní přijímače určovaly hodinové impulsy [1], [2], [4]. Patří sem tyto kódy:

1. Synchronizační kód SMPTE / EBU,
2. Synchronizační kódy LTC, VITC,
3. Synchronizační kód MIDI Time Code (MTC).

## 2 OBVODOVÝ NÁVRH ZAŘÍZENÍ

### 2.1 Popis mikrokontroléru

Pro zařízení byl vybrán jako řídicí obvod mikrokontrolér firmy ATMEL a to ATtiny2313. Jedná se sice o téměř nejmenší mikrokontrolér firmy ATMEL, avšak pro toto použití zcela dostačuje. Jedná se o 8bitový mikrokontrolér, s 32 obecně použitelnými registry. Jako hlavní programovou paměť má 2 KB Flash, dále má k dispozici 128B paměti SRAM, a také 128B EEPROM paměti. Dále má integrovány různé periferie, uvedeny jsou však jen ty, které budou použity pro návrh a realizaci generátoru a to především hardwarový USART který je použit pro příjem a vysílání samotných MIDI rámců. Jako další důležitá periferie je zde 16bitový čítač/časovač a pro komunikaci se sedmsegmentovým displejem je využito sériové rozhraní USI. Dále disponuje paralelními vstupně/výstupními branami (porty), z nichž jeden zajišťuje funkčnost tlačítek. Veškeré tyto a další periferie mají své příslušné konfigurační registry. Jednotlivé součástky byly vybrány cenově nenáročné, pro snazší montáž v provedení DIL. Pro návrh schématu a DPS byl zvolen program EAGLE verze 5.8.0.

### 2.1.1 Periferie USART

Umožňuje jak asynchronní tak synchronní přenos s možností volby počtu stopbitů, parity a počtu databitů z rozsahu 5-9. Periferie USART je řízena třemi vstupně/výstupními registry a jedním pro vstup/výstup dat:

- UCR - řídicí registr, konfigurace příjmu a vysílání, masky přerušení,
- USR - stavový registr, příznaků přenosu,
- UBRR - řídicí registr přenosové rychlosti,
- UDR - datový registr, obsahuje vyslanou či přijatou hodnotu.

### 2.1.2 Rozhraní USI

Zajišťuje základní hardwarové prostředky potřebné pro sériovou komunikaci. Jeho výhoda spočívá v minimu nutného kontrolního softwaru, taktéž má implementováno hardwarové přerušení. Je typu Master-Slave, hodiny generuje Master. Po dokončení vysílání je hardwarově požadováno přijetí vyslaných dat pro kontrolu. Periferie USI je řízena dvěma vstupně/výstupními registry a jedním pro vstup/výstup dat:

- SPCR - řídicí registr, který určuje rychlost a formát přenosu,
- SPSR - stavový registr, příznaky dokončení a kolize přenosu,
- SPDR - datový registr, vstupní a výstupní data.

Rozhraní USI má více operačních režimů, v tomto případě bylo zvoleno zapojení v třídrátovém režimu (Three-wire Mode). Jednotlivé vývody mají následující význam:

- DO - Master Out, výstupní i vstupní pin, ze kterého se data vysílají ven z posuvného registru, na stejném pinu jsou pak i přijímána,
- OC1A - výstup 4bitového čítače USI,
- UCSK - hodinový signál generovaný USI Mastrem.

### 2.1.3 16bitový čítač/časovač

Tento čítač/časovač disponuje rozlišením 16bitů, což je zvoleno kvůli přesnosti, protože je použit pro generování MIDI Click. Jako zdroj hodinového signálu je možné zvolit vnější signál, nebo hodinový signál mikrokontroléru CLK s možností aktivovat předděličku ( $f_0$ ,  $f_0/8$ ,  $f_0/64$ ,  $f_0/256$ ,  $f_0/1024$ ). Čítač obsahuje také obvod plnicí funkci Output Compare, který umožňuje průběžnou kontrolu aktuální napočítané hodnoty s hodnotou uloženou v komparačním registru (OCR1AH, OCR1AH). Při shodě umožňuje vyvolat hardwarové přerušení. S periferií 16bitového čítače/časovače jsou spojeny tyto řídicí a datové registry:

- TCCR1A - řídicí registr, konfigurac Output Compare,

- TCCR1B - řídicí registr, výběr hodinového zdroje a konfiguraci Output Compare,
- TIMSK - řídicí registr, povolení přerušování,
- TIFR - stavový registr příznaků čítače,
- TCNT1H, TCNT1L - obsahují načítanou hodnotu čítače,
- OCR1AH, OCR1AL - komparační registry, nastavení hodnoty Output Compare.

### 2.1.4 Vektory přerušování ATtiny2313

Tyto vektory jsou pevně, hardwarově dány. Jsou tedy neměnné, a proto z nich můžou plynout určitá omezení, viz Tab. 2.1: Vektory přerušování ATtiny2313 [7]. Z těchto vektorů jsou použity Timer/Counter1 Compare Match A pro generování MIDI Timing Clock. Který vyvolá přerušování při přetečení čítače. Pro obsluhu UART je použit vektor přerušování USART0, Tx Complete. Který je aktivován po úplném přijetí bytu na UARTu. Platí, že čím níže je daný vektor umístěn (má nižší hodnotu), tím má vyšší prioritu. Nastane-li případ dalšího přerušování při vykonávání právě přijatého, bude vykonáno po dokončení prvního.

Tab. 2.1: Vektory přerušování ATtiny2313 [7].

Číslo vektoru	Adresa programu	Definice
1	0x0000	External Pin, Power-on Reset, Brown-out Reset
2	0x0001	External Interrupt Request 0
3	0x0002	External Interrupt Request 1
4	0x0003	Timer/Counter1 Capture Event
5	0x0004	Timer/Counter1 Compare Match A
6	0x0005	Timer/Counter1 Overflow
7	0x0006	Timer/Counter0 Overflow
8	0x0007	USART0, Rx Complete
9	0x0008	USART0 Data Register Empty
10	0x0009	USART0, Tx Complete
11	0x000A	Analog Comparator
12	0x000B	Pin Change Interrupt
13	0x000C	Timer/Counter1 Compare Match B
14	0x000D	Timer/Counter0 Compare Match A
15	0x000E	Timer/Counter0 Compare Match B
16	0x000F	USI Start Condition
17	0x0010	USI Overflow
18	0x0011	EEPROM Ready
19	0x0012	Watchdog Timer Overflow

### 2.1.5 Port PD

ATtiny 2313 obsahuje celkem dva porty, port B a port D. Každý je možné je nakonfigurovat jako vstupní nebo výstupní a to dokonce i pro jednotlivé piny portu.

Každý port disponuje možností zapnutí interních pullup rezistorů. Pro zapojení tlačítek byl vybrán port D, tento port není úplný, což znamená, že má jen sedm využitelných bitů, avšak dva z nich jsou použity pro USART. Zbýlých pět pinů zajišťuje funkci tlačítek. Ošetření zákmitu tlačítka je řešeno softwarově. Každý port je řízen třemi vstupně/výstupními registry, toto platí pro port D:

- PIND - Pins Input D, je určen pro čtení a odpovídá hodnotě přečtené z jednotlivých pinů,
- DDRD - data direction register, určuje směr toku dat - vstup/výstup,
- PORTD - datový registr portu, odpovídá hodnotě zapsané na port.

## 2.2 Obvodový popis

Schéma zapojení je uvedeno v příloze A. Mikrokontrolér je zapojen v doporučeném zapojení, jako zdroj hodinového kmitočtu je použitý externí krystal Q1. Vývody oscilátoru jsou ošetřeny dvěma kondenzátory 22 pF dle doporučení, volba je uvedena dále.

Tlačítka jsou součástí desky plošných spojů, a jsou přímo připojeny k pinům mikrokontroléru. Elektricky jsou tlačítka zapojeny proti společnému vodiči GND, takže se sepnutím uzemní. Ošetření zákmitů tlačítek je řešeno programově, stejně tak zapnutí pullup rezistorů. Jako spínací prvky byly zvoleny mikrospínače.

Další částí elektrického obvodu je zabezpečení komunikace se sériovým portem počítače. Pro tuto komunikaci se používá sériového rozhraní RS 232. Pro převod úrovní mikrokontroléru na úroveň RS 232 je použit obvod firmy MAXIM [9]. Obvod je zapojen podle doporučeného elektrického zapojení. Je napájen z 5V a pro generování kladného a záporného napájecího napětí používá nábojovou pumpu. Pro správnou činnost této části obvodu jsou použity čtyři externí elektrolytické kondenzátory C7 až C10. Protože obvod poskytuje dva převodníky z úrovně RS-232 na úroveň TTL a dva převodníky z úrovně TTL na úroveň RS-232, pro správnou komunikaci chybí ještě konverze signálu RST, který je vysílán sériovým portem počítače. Konverze tohoto signálu je pro jednoduchost provedena zenerovou diodou VZ1. Jako ochranný odpor tohoto diodového omezovače je použit odpor R2.

Avšak ve výsledném prototypu bylo od tohoto zapojení s obvodem MAXIM a spojení skrze sériovou linku upuštěno. Byla zvolena metoda za použití paralelního portu, takže nebylo zapotřebí převádět napěťové úrovně, jako by tomu bylo při použití RS-232. Patříčné piny ISP rozhraní mikrokontroléru jsou přímo připojeny na piny portu LTP, a to piny:

- $\overline{RESET}$ , slouží pro resetování procesoru,
- SCK, synchronizační hodiny,

- MISO, MOSI, vodiče používané pro sériové čtení a sériový zápis obsahu interní paměti EEPROM a FLASH.

Tyto piny jsou taktéž vyvedeny pro zjednodušení odladění aplikace na DPS.

Pro zobrazování hodnoty tempa (BPM) či SPP se používá třímístný displej. Jako vlastní zobrazovače slouží diskrétní LED segmentovky, které jsou použity v uspořádání se společnou anodou. Buzení těchto segmentovek je provedeno z výstupu posuvných registrů, které jsou realizovány integrovanými obvody 74LS595.

Pro omezení proudů jednotlivými LED diodami je zapojeno pro každou katodu sériově  $270\Omega$  odpor. Zapojení jednotlivých vývodů segmentovek k jednotlivým výstupům posuvných registrů je optimalizováno s ohledem na jednoduchost návrhu plošného spoje. Toto zapojení potom odpovídá dekodovací tabulce pro zobrazení jednotlivých číslic, uložené v programové paměti mikrokontroléru. Samotné posuvné registry jsou zapojeny jako jeden 24 bitový posuvný registr a proto je vždy výstup předchozího obvodu (pin9) přiveden na vstup následujícího obvodu (pin14).

Na celé desce plošného spoje je dále blokováno napájecí napětí 5V kondenzátory C4, C5, C7 až C9. Tyto kondenzátory jsou rozmístěny v blízkosti jednotlivých integrovaných obvodů opět z důvodu potlačení rušení, které by se šířilo po napájecích vodivých cestách. Všechny zmiňované kondenzátory jsou v keramickém provedení.

Celý obvod lze napájet ze stejnosměrného bezpečného napětí 5V. Napájecí napětí se přivádí na svorkovnici XC1, přičemž kladný pól je na pinu dva. Dioda D2 slouží jako ochrana proti přepólování napájecího zdroje.

### 2.2.1 Rozhraní MIDI

Další částí obvodu je rozhraní MIDI, na vstupu je odizolováno optickým vazebním členem. Impedanční přizpůsobení je provedeno podle doporučení pro MIDI interface pomocí odporu R7 s hodnotou  $220\ \Omega$  [5]. Pro optické oddělení na vstupu je použitý optron 6N137 [5] a jeho výstup je přizpůsobený přímo TTL úrovní. Optron je tedy přímo připojen na pin mikrokontroléru RX. Dioda D1 slouží jako ochranný prvek optronu proti záporným špičkám, které mohou vznikat při použití dlouhého kabelu MIDI.

Výstup MIDI rozhraní je veden z pinu UART TX mikrokontroléru na schmittův klopný obvod, pro všechny výstupní konektory. Schmittův klopný obvod zajistí zlepšení tvaru hran signálu díky své hysterезi, proudové zesílení a dvojí negaci dle [5]. Odtud již jde výstupní signál pro každý konektor zvlášť na tři výstupní konektory MIDI. Výstupní konektory jsou impedančně přizpůsobeny odpory ze schmittova klopného obvodu. Na pátém pinu konektoru, a na čtvrtém pinu proti VCC. Na druhém pinu je konektor uzemněn, dle [5].

### **2.2.2 Výběr oscilátoru a jeho vliv na chybu UART**

Oscilátor byl vybrán s ohledem na pokud možno nulovou chybovost UART přenosu, avšak jen teoretickou. V podstatě mohou být dva možné zdroje chybovosti přijímače. Systém časování přijímače má vždy určité nestability způsobené napájecím napětí a provozní teplotou. Při použití krystalového oscilátoru pro generování hlavního hodinového kmitočtu mikrokontroléru se mnoha problémům předejde. Oproti použití např. rezonátoru kde může vznikat chyba více než 2 % z celkové hodnoty kmitočtu, v závislosti na provozní teplotě. Druhý zdroj chybovosti, který je však lépe kontrolovatelný, je to, že generátor hodinových impulsů pro taktování rozhraní nemůže vždy vydělit hodinový kmitočet na požadovanou rychlost.

Z tohoto plyne chyba vzniklá při špatné synchronizaci start bitu přijímaných a vysílaných rámců. Zejména může nastat problém při přijímání rámců do mikrokontroléru. Hodnota Baudové rychlosti, je totiž tvořena děličkou z hlavního hodinového kmitočtu mikrokontroléru. A je proto vhodné, aby dělitel byl ve tvaru celého čísla viz kapitola 3.2.2. Pro normální asynchronní přenos je udávána maximální tolerovaná chyba rychlosti na přijímači mikrokontroléru +3,78 / -3,83 %, přičemž doporučená je +/- 1,5 %. Tato doporučení vychází z úvahy, že přijímač i vysílač se podílejí na výsledné maximální chybovosti stejnou měrou [7].

## **2.3 Sériový programátor**

Pro programování se využije metody sériového programování, pomocí ISP rozhraní, což značně ulehčuje výběr vývojového kitu - programátoru. Tato metoda má sice svá omezení, ale pro toto použití zcela dostačuje. Navíc není potřeba vyrábět či kupovat žádný nákladný vývojový kit, ale stačí jen sériový (RS-232) či paralelní port (LPT). Jako programovací aplikace byl zvolen program PonyProg.

## 2.4 Zapojení pinů ATtiny 2313

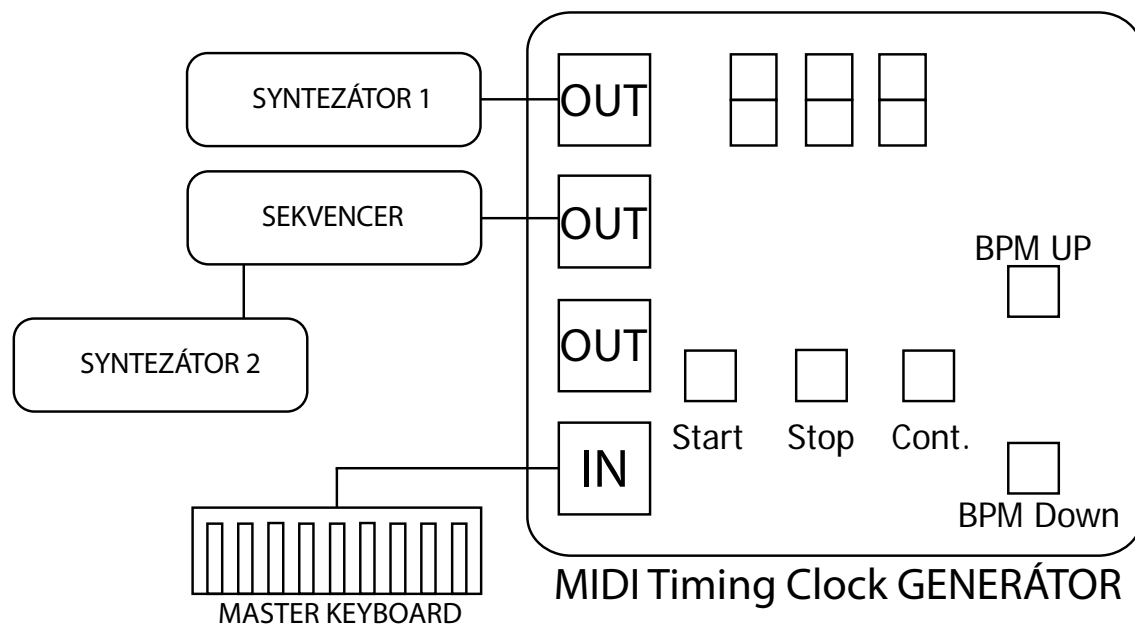
V tabulce 2.2 je uveden výčet a význam jednotlivých pinů mikrokontroléru.

Tab. 2.2: ATtiny zapojení pinů.

Pin	Označení datasheet	Vstup/výstup, význam a funkce
1	PA2	( <i>RESET</i> ), resetovací při programování
2	PD0 (RXD)	Vstup UART
3	PD1 (TXD)	Výstup UART
4	PA1 (XTAL2)	Zapojení krystalu 12MHz
5	PA0 (XTAL1)	Zapojení krystalu 12MHz
6	PD2	Vstup, tlačítko Start
7	PD3	Vstup, tlačítko Continue
8	PD4	Vstup, tlačítko Stop
9	PD5	Vstup, tlačítko BPM DOWN
10	GND	Zem
11	PD6	Vstup, tlačítko BPM UP
12	PB0	-
13	PB1	-
14	PB2	-
15	PB3 (OC1A)	Vstup USI counter
16	PB4	-
17	PB5 (DI)	Výstup /Výstup, MOSI
18	PB6 (DO)	Vstup/Výstup, MISO a USI
19	PB7 (USCK)	Vstup/Výstup, SCK a USI clock
20	VCC	Napájení

## 2.5 Zapojení generátoru do MIDI systému

Toto je teoretický příklad zapojení Generátoru MIDI Clock v MIDI systému.



Obr. 2.1: Zapojení generátoru v MIDI systému.

## 3 SOFTWAREVÝ NÁVRH ZAŘÍZENÍ

Jako vývojové prostředí bylo zvoleno ATMEL AVR Studio 4.17, což je freeware poskytovaný stejnojmennou firmou. Z důvodu názornosti a pro lepší pochopení byl zvolen jako programovací jazyk assembler.

### 3.1 Popis činnosti programu

Nejdříve proběhne inicializace, která provede nezbytné úkony pro nastavení generátoru do implicitního stavu, a především povolí přerušení. Použity jsou dvě přerušení, první nastane při přetečení čítače jedna. Po němž je spuštěna obsluha přerušení, která vynuluje čítač a okamžitě se snaží vyslat byte MIDI Click \$F8. Druhé přerušení nastane po úplném přijetí bytu z UARTu, obsluha přerušení tento byte uloží a taktéž se snaží ihned vyslat tento byte UARTem ven. Podrobnější detaily jsou uvedeny dále.

Kromě obsluhy přerušení je program zacyklen ve smyčce obsluhy bpm. Což znamená, že neustále kontroluje tlačítka, a je tedy možné pomocí tlačítek BPM UP/DOWN upravovat tempo generovaných značek MIDI Click, které se generují neustále. Přičemž po stisku je zavolána obsluha změny BPM viz dále kapitola 3.4. Taktéž lze stiskem patřičných tlačítek poslat MIDI zprávu Start/Stop, které se cyklicky střídají, nebo zprávu Continue. Na displeji je zobrazena hodnota aktuálního tempa. Po stisku tlačítka BPM/SPP které slouží pro přepnutí do editace/vyslání MIDI SPP se program přepne do smyčky obsluhy SPP. To znamená, že na displeji je zobrazena aktuální nastavená hodnota dolního nibblu hodnoty MIDI SPP. Tlačítkem Continue se lze přepnout do editace horního nibblu MIDI SPP. Taktéž se může nastavená hodnota MIDI SPP vyslat tlačítkem Start/Stop, nebo upravit tlačítka BPM UP/DOWN. Opětovný stisk tlačítka BPM/SPP program uvede do smyčky obsluhy BPM.

### 3.2 Inicializace mikrokontroléru

#### 3.2.1 Inicializace proměnných

Zmíním především vlastní definovaný příznakový registr FLAGS, který je využit při příjmu SysEx, a dále vyhrazené místo v paměti SRAM jako zásobník ve kterém se ukládají vypočtené hodnoty offsetů použité při překreslování displeje. Dále jsou použity již jen běžné vstupně/výstupní registry.

#### 3.2.2 Inicializace UART

Rozhraní USART je nakonfigurováno pro vysílání a příjem UART rámců dle parametrů MIDI. A to jednoho start bitu, osmi datových bitů, jednoho stop bitu, bez parity. Pro

výpočet rychlosti v baudech platí následující vztah 3.1 [7]:

$$PR = \frac{f_{osc}}{16(UBRR + 1)}, \quad (3.1)$$

kde PR je výsledná přenosová rychlost v Bd,  $f_{osc}$  je kmitočet oscilátoru mikrokontroléru, a UBRR je hodnota registru UBRR. Z tohoto vztahu se určila výsledná hodnota 23, která se zapíše do registru UBRR, což odpovídá výsledné přenosové rychlosti 31250 baudů.

### 3.2.3 Inicializace a obsluha 16bitového čítače

Protože registry TCNT1L/H, a OCR1AL/H tvoří celkově 16bitový registr, ale je použit pouze 8bitový mikrokontrolér, je nutné k nim přistupovat po polovinách. Při zápisu musí být nejdříve zapsán horní byte H, který se uloží do dočasného registru. Poté proběhne zápis dolního bytu, při kterém se zároveň nakopíruje i byte horní. Při tomto zápisu musí být zakázáno přerušení. Pro zákaz se používá instrukce SEI a pro povolení CLI.

Pro nastavení masky přerušení (povolení) se povolí bit OCIE1A v registru TIMSK, dále se nastaví příznak přerušení nastavením bitu OCF1A v registru TIFR. Oba tyto registry slouží pro globální nastavení přerušení časovačů mikrokontroléru.

Jako kmitočet pro čítač je vybrán hodinový kmitočet s před děličkou 64, z čehož plyne frekvence jedné periody a to 187500 Hz, při vstupním kmitočtu 12 MHz. Z toho se určí čas jedné periody a to  $T_C = 5,3 \cdot 10^{-6}$  s. Pro příklad uvedu výpočet doby generování jedné periody časovače při 128 BPM, z vzorce 3.2 vyplývá že, perioda za jakou se má vyslat zpráva MIDI Click je  $T_B = 0,088$  s.

$$T_B = \frac{BPM / 60}{24} \quad (3.2)$$

Pro výpočet hodnoty konstanty do registru OCR1AL/H pro zvolené tempo BPM, se musí hodnota  $T_B$  vydělit konstantní hodnotou  $T_C$ . Z čehož plyne výsledná hodnota do registru čítače  $H_K \doteq 16667$ , zaokrouhlená na celé číslo. Pro celý rozsah BPM a to 50 - 250, jsou vypočítané hodnoty  $H_K$  uloženy v programové paměti. Hodnoty jsou převedeny do šestnáctkové soustavy, přičemž horní polovina je hodnota OCR1AH a spodní OCR1AL.

Přivedením zdroje hodin s předděličkou 64, což se provede zápisem do registru TCCR1B se nastaví zdroj hodinového kmitočtu pro čítač a tím se spustí.

### 3.3 Obsluha sedmisegmentového displeje

Pro zobrazení na sedmi segmentovém displeji je zavolána rutina DIVIDE, která má za úkol číslo BPM nebo SPP dekadicky rozdělit na jednotlivé znaky (offsety). Offset protože toto číslo je jen umístění v tabulce konkrétních znaků (závislé na elektrickém zapojení displeje). Tyto offsety, se po dekadickém rozdělení uloží do paměti SRAM. Rutina 8bitového dělení je převzata z [8]. Pro samotné zobrazení je zavolána rutina DISP, která načte hodnotu offsetu pro konkrétní sedmi segmentovku z paměti SRAM. Následovaná načtením znaku požadovaného zobrazovaného symbolu z tabulky znaků, a pošle jej přes USI. Začíná se posláním posledního znaku, z důvodu obvodového řešení.

### 3.4 Obsluha změny BPM

Po stisku tlačítka BPM nahoru či dolů, dojde k inkrementaci nebo dekrementaci proměnné BPM, která udává aktuální nastavenou hodnotu tempa. Následně dojde k ověření, zda hodnota BPM nepřesáhla 250 či není menší než 50, což by znamenalo návrat zpět do hlavního programu. Protože však hodnoty BPM uložené v programové paměti celkově zabírají  $200 \times 2 \text{byty} = 400\text{B}$ , představuje to pro adresaci menší problém, z důvodu zvolené osmibitové architektury. Je tedy nutné tabulku znaků rozdělit, přičemž hodnoty nad 178 BPM se načítají již z druhé tabulky. A to z důvodu že v první tabulce je uložených 128 hodnot BPM, což jak plyne z výše uvedené kapitoly 3.2.3 znamená 256 bytů, čímž je vyčerpán rozsah adresace jedním bytem. Následně jsou již jen načteny z tabulky konstanty do OCR1AH/L, vynulován aktuální stav čítače a překreslen display zavoláním rutiny DIVIDE.

### 3.5 Obsluha přijetí UART a SysEx

Je řešena přerušením, což znamená spuštění obsluhy přerušení při úplném přijetí bytu z UARTu. Obslužná rutina vykoná přečtení bytu z UART, ověření zda již není aktivován příjem SysEx (ověřením hodnoty bitu SERC v registru FLAGS) a případně provede kontrolu na SysEx (porovnání s \$F0) a okamžitě se snaží byte opět poslat ven.

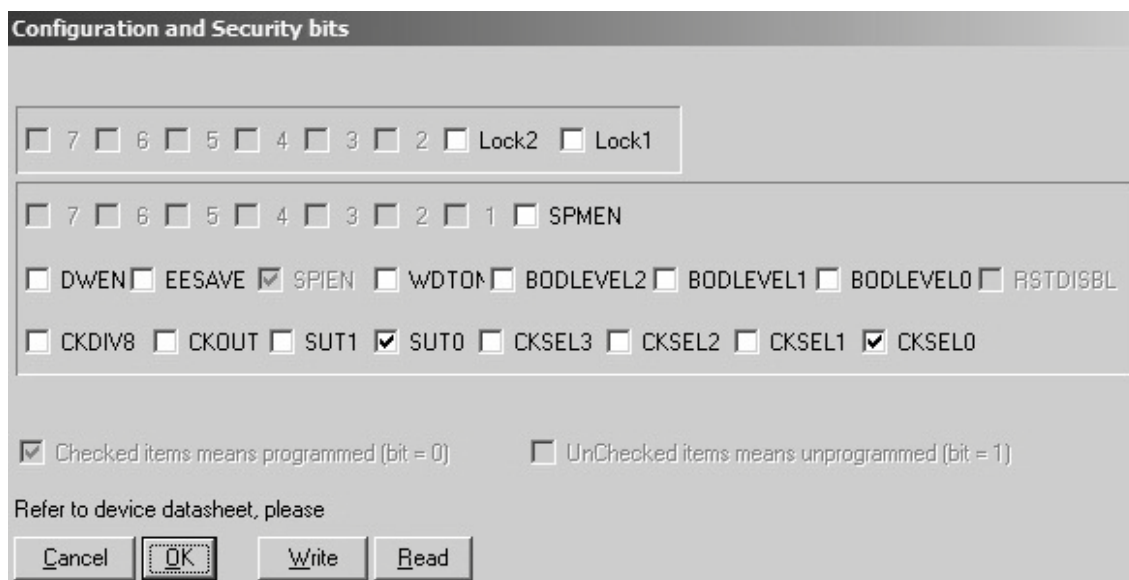
Nastane-li případ SysEx, nastaví se v příznakovém registru FLAGS aktivní příjem SysEx, což znamená, že při přijetí dalšího bytu z UART se již porovnává, zda se jedná o očekávané ID Number tohoto zařízení. Jestli ano, tak při dalším přijetí bytu z UART se ověří, zda je to pro toto konkrétní zařízení. Další dva přijaté byty z UART, již obsahují hodnotu BPM, přičemž první (BPM0) obsahuje spodní nibble a druhý (BPM1) horní nibble. Oba tyto byty se uloží do SRAM, a provede se jejich součet, který udává požadovanou hodnotu BPM, poté se zavolá rutina změny tempa a překreslení displeje. Jestli nastane případ, že při kterémkoliv porovnání identifikačních bytů dojde k neshodě, registr FLAGS se vynuluje.



Obr. 3.1: Struktura SysEx rámce pro generátor MIDI Click.

### 3.6 Nastavení FUSES

Fuses - pojistky, jsou velmi důležitou, užitečnou, ale také mohou být velmi nebezpečnou součástí většiny osmibitových mikrokontrolérů Atmel (můžou se lišit dle konkrétního modelu). Je zde totiž možnost neopatrným nastavením mikrokontrolér zcela zničit. Nastavení těchto konfiguračních bitů nelze provádět programově, ale pouze na programátoru. Například zaškrtnutím pojistky EESAVE se uzamkne paměť EEPROM. Či odškrtnutím pojistky SPIEN (SPI Enable) je možné zakázat programování skrze SPI rozhraní. Naštěstí lze některé "zámky" ještě odstranit, avšak již je nutné použít dražší a pokročilejší programovací kit. Pro tento případ aplikace jsou však důležité pojistky CKSEL3, CKSEL2, CKSEL1, CKSEL0, SUT1, SUT0. Tyto pojistky totiž volí zdroj hodinových pulsů, a to dle [7]. Naprogramováním pojistek CKSEL3..1 se volí rychlost vstupních hodinových impulsů na rychlost 8,0 - X MHz. Dále se kombinací pojistek SUT1.0 a CKSEL0 zvolil krystalový oscilátor, s rychlým náběhem napětí. Což znamená náběh mikrokontroléru od zapnutí po proběhnutí 16K hodinových cyklů = cca 1,3 ms. Při restartu je tato doba zkrácena na 14 hodinových cyklů + 4,1 ms.



Obr. 3.2: Ukázka nastavení pojistek - Fuses.

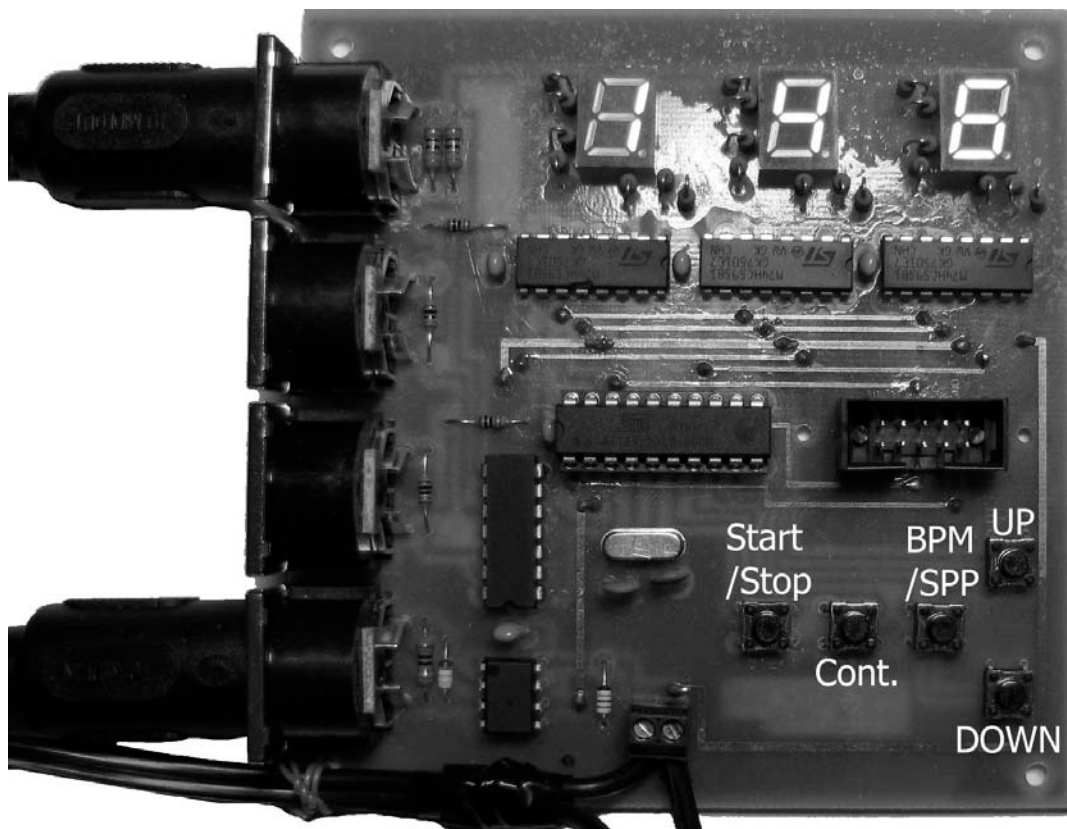
## 4 REALIZACE PROTOTYPU

Nejdříve byl celý obvod postupně sestaven a ožívován po jednotlivých funkčních blocích na nepájivém poli. Jako tester komunikace sloužil převodník MIDI do portu USB, M-Audio - USB Uno MIDI Interface. Vyskytly se však problémy, a to že z počátku zcela nefungoval UART, a to jak příjem, tak i vysílání. Po zkontrolování obvodového zapojení, proměření výstupního schmittova klopného obvodu, a taktéž vstupního optronu bylo usouzeno, že závada bude způsobena mikrokontrolérem.

Po důkladném pročení datasheetu se zjistilo, že v nastavení pojistek mikrokontroléru je taktéž navíc zaškrnut bit CKDIV8, viz Obr. 3.2. Zapnutím tohoto bitu však má jediný účel, a to je vydělení vstupního kmitočtu mikrokontroléru předděličkou 8krát. Z tohoto důvodu byla ve výsledku zcela špatně nastavena Baudová rychlost, protože mikrokontrolér nepracoval na předpokládaných 12 MHz, ale pouze na 1,5 MHz. Po odstranění této "závady" již fungovalo vysílání UART. Avšak stále nefungoval příjem UART, což bylo odstraněno zapnutím interního pull up odporu na pinu UART RX.

Jakmile vše fungovalo, DPS byla vyleptána. Po vyleptání se vyvrtaly díry pro osazení součástek a taktéž pro jednotlivé prokovy. Prokovy byly nutné kvůli propojení posuvných registrů, které mají hodně pinů propojených paralelně. Nejdříve se připájely prokovy, po čemž se zkontrolovalo správné propojení a případné studené spoje prokovených cest. Poté byla připájena patice mikrokontroléru, spolu se všemi pasivními R, C prvky, spínači, sedmissegmentovkami, krystalem a také veškerými konektory (součástky méně citlivé na teplotní poškození). Na závěr byly zapájeny veškeré integrované obvody. Po proměření napájecích pinů v patici byl osazen mikrokontrolér. Poté proběhl test naprogramování mikrokontroléru, čímž byl výrobek kompletně oživen.

Proudový odběr při testování při napájení ze stejnosměrných 5V činil maximálně 0,025 A. Programová paměť flash je zaplněna z 56,3 %, a to konkrétně 1154 B, z kterých však minimálně 400 B zabírají hodnoty konstant. Přesnost intervalů generovaných značek MIDI Timing Clock byla ověřena na logickém analyzátoru. Funkčnost přijetí SysEx a jejich zpracování byla ověřena programem SysEx 97. Výsledný prototyp je vidět na Obr. 4.1.



Obr. 4.1 Prototyp generátoru MIDI Click / SPP.

## 5 ZHODNOCENÍ A MOŽNOSTI ROZŠÍŘENÍ

Řídicí program by bylo vhodné prakticky otestovat a ověřit jeho chování v reálném MIDI systému. Protože v simulaci nelze přesně určit, k jakému zpoždění může docházet při generování MIDI Click, v závislosti na počtu přijímaných bytů. Teoreticky může nastat největší zpoždění jen díky čekání na odvysílání předešlého rámce, neboli 320  $\mu$ s. Avšak tato hodnota je i při nastaveném nejnižším BPM zanedbatelná.

Taktéž chování generátoru MIDI Click při příjmu SysEx zpráv může být problematické - nepřesné, z důvodu zpoždění při analýze SysEx zpráv. Omezení však plynou i pro použití SPP. Tato omezení jsou dána osmi bitovou architekturou mikrokontroléru, a tudíž nemožností uchování vyšší hodnoty SPP než 255. Proto je obsluha SSP řešena ne příliš přátelsky pro uživatele.

Dále bych uvedl obecné priority chování tohoto MIDI Click / SPP generátoru, které jsou však dokonce dány i zvoleným hardwarem - prioritami přerušení. Jako nejprioritnější je generování a vyslání kódu MIDI Click (nejnižší vektor přerušení), poté příjem dat (další vektor přerušení) a zároveň vysílání přijatých dat. Jako poslední

je řešena obsluha tlačítek.

Jako rozšíření, či spíše zdokonalení by bylo vhodné lépe implementovat nastavení SPP, ať už rozšířením displeje na možnost zobrazení čtyř číslic, nebo jen softwarově zdokonalit zadávání požadované hodnoty SPP.

## 6 ZÁVĚR

Výstupem této bakalářské práce je prototyp DPS Generátoru kódu MIDI Click a zpráv SPP, který lze zároveň použít jako MIDI Thru Box. Jeho výhoda především spočívá v nedostupnosti podobného jednoúčelového levného zařízení pro generování synchronizačního kódu MIDI Click.

Přínos této práce je pro mě jednoznačně v seznámení se s problematikou mikrokontrolérů a jejich programování. V tomto případě jsem zvolil assembler, neboť jsem s programováním mikrokontrolérů neměl žádné zkušenosti. Pro osvojení a pochopení vlastností mikrokontroléru je myslím velmi vhodný. Avšak pro další projekt bych již raději přešel k vyššímu programovacímu jazyku, např. jazyku C. Ať už z důvodu pohodlnosti při zápisu kódu. Ale hlavně kvůli výhodám, kterými jsou např., že je již většina potřebných funkcí vytvořena formou knihoven. A zejména pak jsou v těchto knihovnách ošetřeny stavy, při kterých můžou nastat v assembleru problémy (nefunkčnost), které nejsou pro neznalého jen tak zřejmé. Taktéž jsem si osvojil práci v návrhovém systému plošných spojů Eagle, praktickou výrobu a oživení DPS.

# LITERATURA

- [1] The Complete MIDI 1.0 Detailed Specification, document version 96.1, MIDI Manufacturers Association, Japan MIDI Standard Committee, 1997.
- [2] FORRÓ, D. *MIDI – Komunikace v hudbě*. Praha: Grada Publishing, spol. s r.o., 1993. 267s. ISBN 80-85623-56-0.
- [3] MATOUŠEK, D. , *Práce s mikrokontroléry ATMEL AVR*, 2. vydání. Nakladatelství BEN, Praha, 2005. ISBN 80-7300-209-4
- [4] KÁŇA, L., SCHIMMEL J. *Studiová a hudební elektronika* [online elektronická skripta]. 2009 [cit. 2009-12-10] . Dostupné z URL: <[https://www.feec.vutbr.cz/et/skripta/utko/Studiova\\_a\\_hudebni\\_elektronika\\_S.pdf](https://www.feec.vutbr.cz/et/skripta/utko/Studiova_a_hudebni_elektronika_S.pdf)>.
- [5] MIDI Manufacturers Association [online]. [cit. 2009-12-10] Dostupné z URL: <<http://www.midi.org>>
- [6] *Wikipedia: Musical Instrument Digital Interface* [online]. 2009 [cit. 2009-12-10]. Dostupné z URL: <<http://en.wikipedia.org/wiki/Midi>>.
- [7] Datasheet ATMEL ATtiny2313 [online] [c04/2006] [cit. 2009-12-10] Dostupné z URL: <[http://www.atmel.com/dyn/resources/prod\\_documents/DOC2543.PDF](http://www.atmel.com/dyn/resources/prod_documents/DOC2543.PDF)>
- [8] Datasheet ATMEL AVR200: Multiply and Divide Routines [online] [c09/2009] [cit. 2009-12-10] Dostupné z URL: <[http://www.atmel.com/dyn/resources/prod\\_documents/doc0936.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc0936.pdf)>
- [9] Datasheet MAXIM 232 [online] [c01/2006] [cit. 2009-12-10] Dostupné z URL: <<http://datasheets.maxim-ic.com/en/ds/MAX220-MAX249.pdf>>
- [10] Datasheet HP 6N137 [online] [c05/1996] [cit. 2009-12-10] Dostupné z URL: <<http://www.datasheetcatalog.org/datasheet/hp/HCNW2611.pdf>>
- [11] Datasheet HEF 40106B [online] [c01/1995] [cit. 2009-12-10] Dostupné z URL: <<http://www.datasheetcatalog.org/datasheet2/d/0jz6jjihcz5tpa7s77xog94uofy.pdf>>

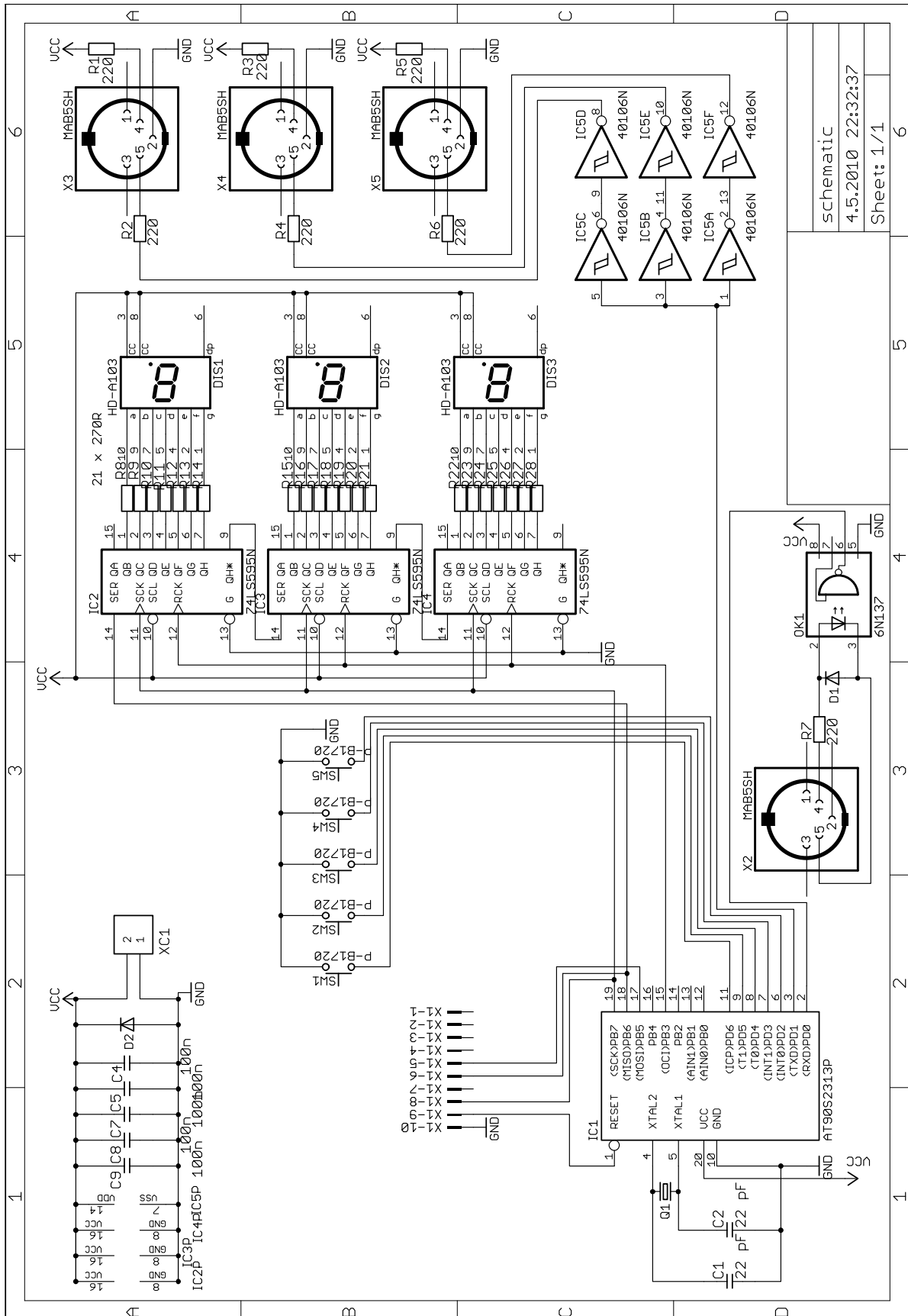
# SEZNAM VELIČIN A ZKRATEK

MIDI	Musical Instruments Digital Interface
MTC	MIDI Time Clock
SMPTE	Society of Motion Picture and Television Engineers
EBU	European Broadcasting Union
DCB	Digital Communication Bus
NAMM	National Association of Music Merchants
USI	Universal Synthesizer Interface / Universal Serial Interface
MMA	MIDI Manufacturers Association
JMSC	Japan MIDI Standard Committee
EOX	End Of System Exclusive
BPM	Beats per Minute - čtvrtových dob za minutu
SysEx	System Exclusive
LSB	Least Significant Bit, nejméně významný bit
MSB	Most Significant Bit, nejvíce významný bit
nibble	Polovina bytu
CLC	Clock, hodiny
B	Byte
DPS	Deska plošných spojů
SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
RS232	Sériová linka
LPT	Line Printer Terminal
VCC	Napájecí napětí
$\Omega$	Ohm, jednotka elektrického odporu
ISP	In System Programing
VCR	Video Cassete Recorder
PPQN	Pulses Per Quarter Note, impulsů za čtvrtovou notu
FPS	Frames Per Second
PAL	Phase Alternating Line
USB	Universal Serial Bus, universální seriová sběrnice
Tc	Doba periody hodinového kmitočtu s predděličkou 16
Tb	Doba periody generace MIDI Click pro zvolené BPM
DPS	Deska Plošných Spojů
FSK	Frequency Shift Keying
USART	Universal Synchronous and Asynchronous serial Receiver and Transmitter

# SEZNAM PŘÍLOH

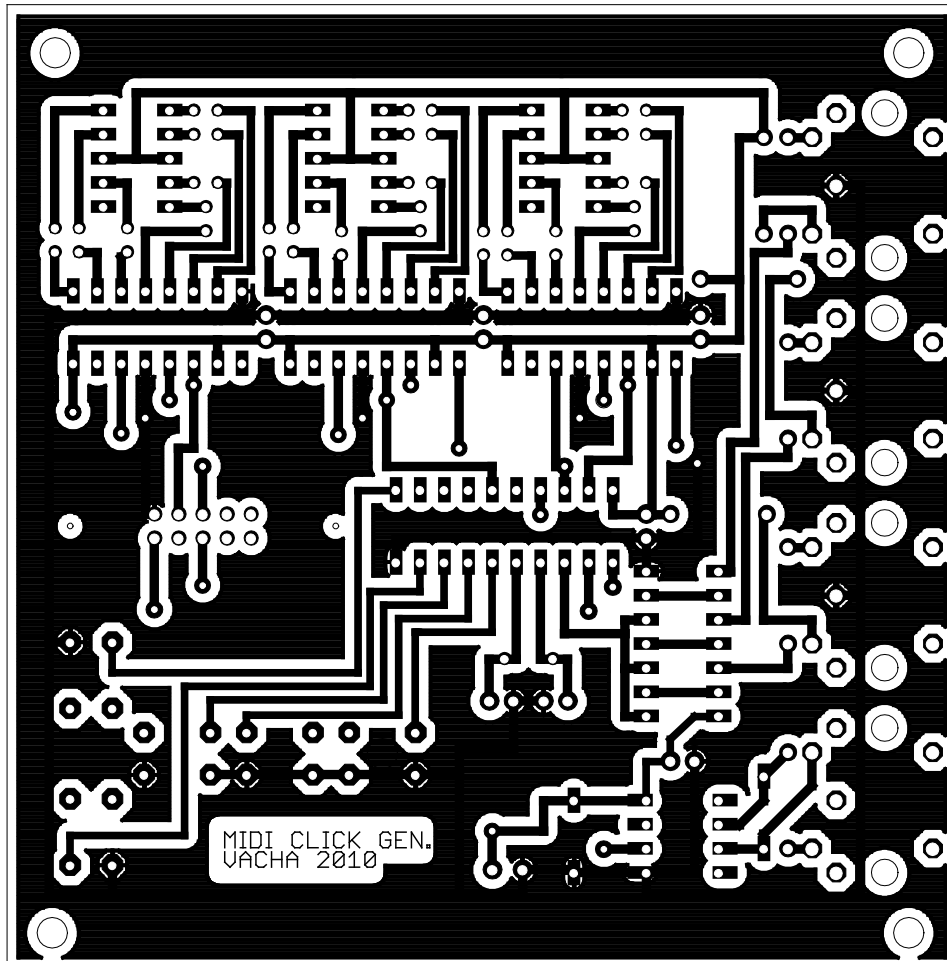
A	Schéma zapojení	34
B	Deska plošného spoje ze strany spojů	35
C	Deska plošného spoje ze strany součástek	36
D	Osazení desky plošného spoje	37
E	Seznam součástek	38

# A SCHÉMA ZAPOJENÍ



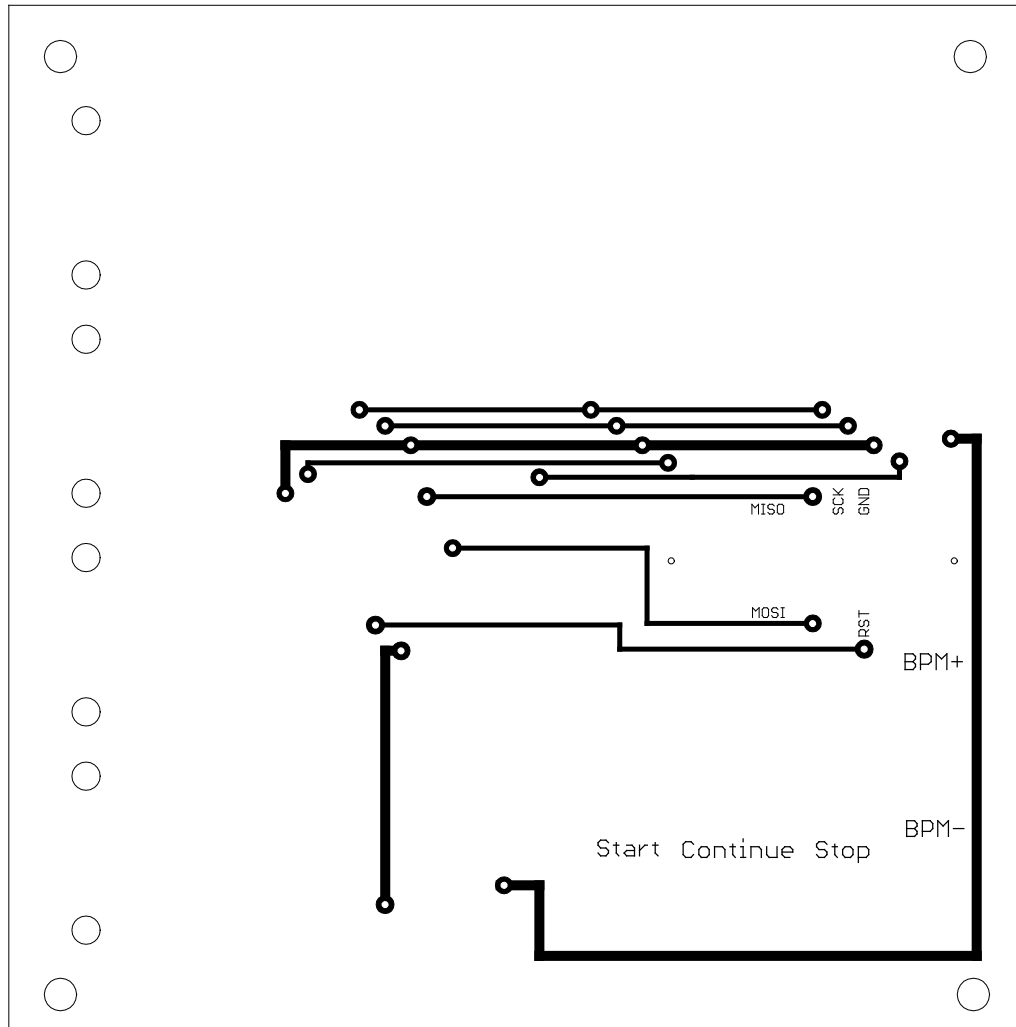
schematic  
4.5.2010 22:32:37  
Sheet: 1/1

## B DESKA PLOŠNÉHO SPOJE ZE STRANY SPOJŮ



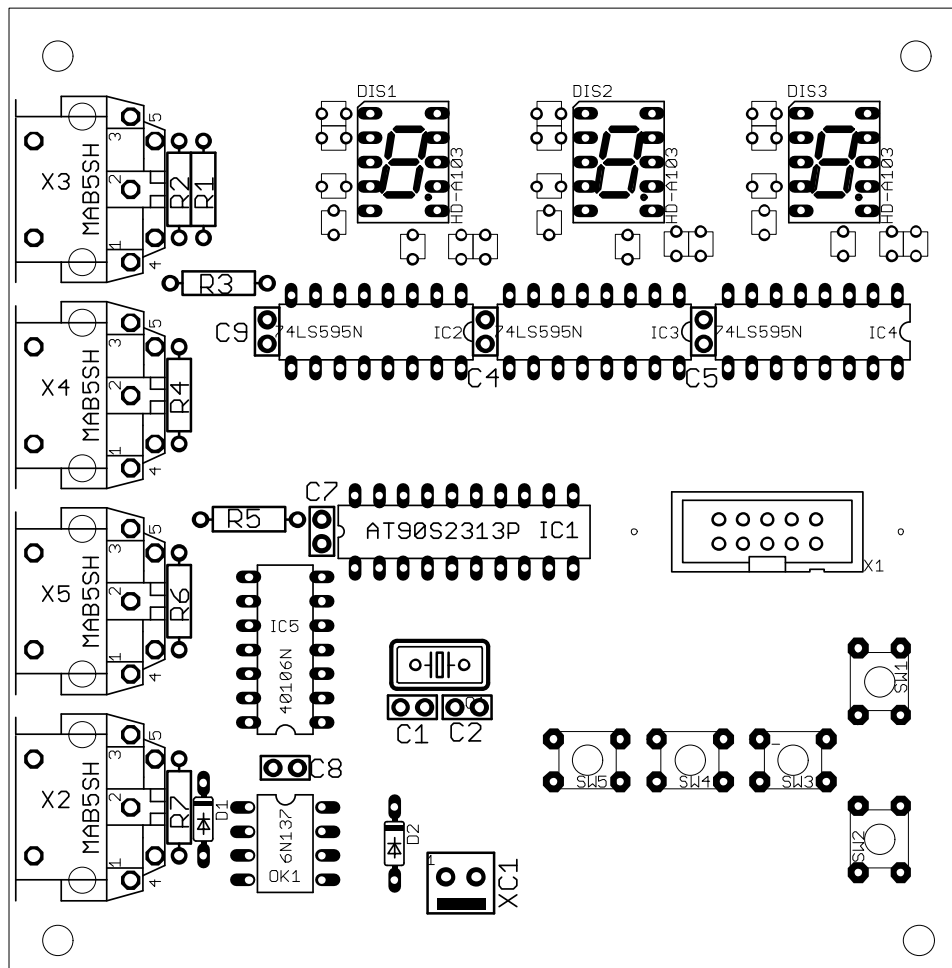
Pohled ze strany spojů, měřítko 1:1.

## C DESKA PLOŠNÉHO SPOJE ZE STRANY SOUČÁSTEK



Pohled ze strany součástek - propojky, měřítko 1:1.

## D OSAZENÍ DESKY PLOŠNÉHO SPOJE



Pohled ze strany součástek, měřítko 1:1.

## E SEZNAM SOUČÁSTEK

Počet	Hodnota	Označení	Součástka	Popis
2	-	1N4148DO35-7	D1, D2	Dioda
1	-	2510	X1	Konektor
1	-	CRYSTALHC18U	Q1	Krystal
1	-	PSH02P	XC1	Konektor - Svorkovnice
1	6N137	6N137	OK1	Optron
2	22 pF	CF-100N/40	C1, C2	Kondenzátor keramický
3	74LS595N	74LS595N	IC2, IC3, IC4	Integrovaný obvod
5	100n	CF-100N/40	C4, C5, C7, C8, C9	Kondenzátor keramický
1	40106N	40106N	IC5	Integrovaný obvod
1	AT90S2313	AT90S2313P	IC1	Integrovaný obvod
3	HD-A103	HD-A103	DIS1, DIS2, DIS3	Sedmi segmentovka
4	MAB5SH	MAB5SH	X2, X3, X4, X5	Konektor DIN5
5	P-B1720	P-B1720	SW1, SW2, SW3, SW4, SW5	Mikrospínač
7	220	TR296	R1, R2, R3, R4, R5, R6, R7	Odpor
21	270R	TR296	R8, R9, R10, R11, R12, R13, R14, R15, R16, R17, R18, R19, R20, R21, R22, R23	Odpor