

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

SEMESTRÁLNÍ PRÁCE



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## MIDI PO ETHERNETU

### SEMESTRÁLNÍ PRÁCE

SEMESTRAL THESIS

### AUTOR PRÁCE

AUTHOR

Vojtěch Lukáš

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Ondřej Krajsa, Ph.D.

BRNO 2020

# Semestrální práce

bakalářský studijní program **Audio inženýrství**  
specializace Zvuková produkce a nahrávání  
Ústav telekomunikací

**Student:** Vojtěch Lukáš

**ID:** 211572

**Ročník:** 3

**Akademický rok:** 2020/21

**NÁZEV TÉMATU:**

## MIDI po Ethernetu

### POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte přípravek pro přenos protokolu MIDI po Ethernetu. Přípravek bude mít dva vstupní porty a dva výstupní porty s možností přepnutí do režimu Thru. Data přenášená po Ethernetu pak budou v dalším přípravku odeslána na patřičné MIDI porty. V rámci semestrální práce proveďte návrh zařízení a testování dílčích částí s využitím vývojového kitu.

### DOPORUČENÁ LITERATURA:

[1] LINSLEY HOOD, John. Audio Electronics. Kent: Elsevier Science, 1995. ISBN 9780750621816

[2] GUÉRIN, Robert. Velká kniha MIDI: standardy, hardware, software. Brno: Computer Press, 2004, 340 s. ISBN 80-7226-985-2

**Termín zadání:** 2.10.2020

**Termín odevzdání:** 11.12.2020

**Vedoucí práce:** Ing. Ondřej Krajša, Ph.D.

**doc. Ing. Jiří Schimmel, Ph.D.**  
předseda rady studijního programu

### UPOZORNĚNÍ:

Autor semestrální práce nesmí při vytváření semestrální 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.

## PROHLÁŠENÍ

Prohlašuji, že svou semestrální práci na téma „MIDI po Ethernetu“ jsem vypracoval samostatně pod vedením vedoucího semestrální 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 semestrální 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/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora

# Obsah

<b>Úvod</b>	<b>9</b>
<b>1 Protokol MIDI</b>	<b>10</b>
1.1 Hardwarová vrstva . . . . .	10
1.1.1 MIDI výstup . . . . .	10
1.1.2 MIDI vstup . . . . .	11
1.2 Softwarová vrstva . . . . .	12
1.2.1 Výjimky . . . . .	13
<b>2 Návrh přípravku</b>	<b>14</b>
2.1 Hardware . . . . .	14
2.2 Software . . . . .	15
2.2.1 Databáze spojení . . . . .	15
2.2.2 Formát zprávy . . . . .	16
2.2.3 Síťová komunikace . . . . .	16
<b>3 Průběh komunikace</b>	<b>17</b>
3.1 Připojení k lokální síti . . . . .	17
3.2 Upozornění na vlastní přítomnost . . . . .	17
3.3 Rozpoznání zařízení na síti . . . . .	18
3.4 Přijetí MIDI zprávy . . . . .	18
3.5 Přijetí UDP zprávy . . . . .	18
<b>Závěr</b>	<b>19</b>
<b>Literatura</b>	<b>20</b>
<b>Seznam symbolů, veličin a zkratk</b>	<b>22</b>
<b>Seznam příloh</b>	<b>23</b>
<b>A Tabulky</b>	<b>24</b>
A.1 MIDI STATUS BAJTY . . . . .	24
A.2 MoE značky . . . . .	25
<b>B Výpisy kódu</b>	<b>26</b>
B.1 Výpis kódu pro příjem a zpracování MIDI zprávy . . . . .	26

<b>C</b>	<b>Některé příkazy balíčku <code>thesis</code></b>	<b>28</b>
C.1	Příkazy pro sazbu veličin a jednotek . . . . .	28
C.2	Příkazy pro sazbu symbolů . . . . .	28
<b>D</b>	<b>Druhá příloha</b>	<b>29</b>
<b>E</b>	<b>Příklad sazby zdrojových kódů</b>	<b>30</b>
E.1	Balíček <code>listings</code> . . . . .	30
<b>F</b>	<b>Obsah přiloženého CD</b>	<b>34</b>

# Seznam obrázků

1.1	Schéma MIDI výstupu [1]. . . . .	11
1.2	Schéma MIDI vstupu [1]. . . . .	11
1.3	Bity obou druhů MIDI bajtů [1] . . . . .	12
1.4	Struktura zprávy <i>Změna kontroléru</i> [1]. . . . .	12
2.1	Schéma prototypu [3] . . . . .	14
2.2	Kódování zdrojového a cílového kanálu v bajtu <code>srcdstChannel</code> . . . .	15
D.1	Alenčino zrcadlo . . . . .	29

# Seznam tabulek

A.1	Tabulka MIDI STATUS BAJTŮ [1]. . . . .	24
A.2	Tabulka prvních bajtů MoE zpráv. . . . .	25
C.1	Přehled příkazů . . . . .	28



# Seznam výpisů

E.1	Ukázka sazby zkratk . . . . .	30
E.2	Příklad Schur-Cohnova testu stability v prostředí Matlab. . . . .	32
E.3	Příklad implementace první kanonické formy v jazyce C. . . . .	33

# Úvod

Protokol MIDI<sup>1</sup> je již dlouhá léta zavedeným standardem nejen pro komunikaci mezi elektronickými hudebními nástroji, ale také pro řízení studiové nebo scénické techniky, časovou synchronizaci dvou a více zařízení a podobně. V

Tato práce se věnuje oblasti MIDI (Musical Instrument Digital Interface – digitální rozhraní hudebního nástroje), zejména jevům, které nastanou při nedodržení Nyquistovy podmínky pro *vzorkovací kmitočet* ( $f_{\text{vz}}$ ).<sup>2</sup>

Šablona je nastavena na *dvoustranný tisk*. Pokud máte nějaký závažný důvod sázet (a zejména tisknout) jednostranně, nezapomeňte si přepnout volbu **twoside** na **oneside**!

---

<sup>1</sup>Musical Instrument Digital Interface – digitální rozhraní hudebního nástroje

<sup>2</sup>Tato věta je pouze ukázkou použití příkazů pro sazbu zkratk.

# 1 Protokol MIDI

Tento protokol umožňuje přenos zejména hudebních informací mezi dvěma (i více) elektronickými hudebními nástroji, sekvencery, počítači a dalšími přístroji. Původně byl zamýšlen pro použití „v reálném čase“, tedy při živé produkci. [1] S nástupem moderních DAW<sup>1</sup> ale přišla možnost povely také nahrávat, upravovat a znovu přehrávat.

MIDI se však netýká výhradně hudby a hudebních dat. Umožňuje též přenos kontrolních povelů a synchronizačních značek. Přirozeně se tedy rozšířilo i do nahrávacích studií, divadel a dalších zařízení, kde umožňuje globální řízení jednotlivých přehrávačů nebo vzdálené ovládání parametrů řídicích konzolí.

## 1.1 Hardwarová vrstva

Pro přenos MIDI dat se tradičně používá zásuvka a vidlice DIN 5<sup>2</sup>. Kabel mezi dvěma zařízeními by neměl být delší než 15 m a tvořit by jej měla stíněná kroucená dvojlinka. Toto stínění by mělo být připojeno k pinu 2 na obou koncích.

Výměna informací je realizována pomocí 5mA proudové smyčky. Při protékání proudu je zaznamenána logická 0, v opačném případě logická 1. [1]

Přesná specifikace obvodů pro zpracování příchozích a odchozích MIDI signálů upravuje kapitola *Hardware* z dokumentu [1]. Tato pasáž byla v roce 2014 aktualizována normou [2], která adaptovala protokol i pro zařízení s 3,3V logikou a přidává další prvky pro zamezení zejména RF<sup>3</sup> interferencí. V následujících schématech bude však zobrazeno originální schéma z původní normy,

### 1.1.1 MIDI výstup

Výstupní port MIDI sběrnice je ve své podstatě jednoduchý. Z UART<sup>4</sup> čipu jsou přes napěťové sledovače nebo tranzistory vedeny logické impulzy na pin 5, zatímco na pin 4 je přivedeno stálé napětí. Pin 2 je v tomto případě uzemněn. [1]

Podle aktualizací normy [2] je možné přidat za každý rezistor feritové jádro pro zamezení vysokofrekvenčních interferencí. V případě použití zařízení s 3,3V logikou jsou pak použity rezistory s menšími hodnotami odporů. Na obr. 1.1 je schéma k nahlédnutí.

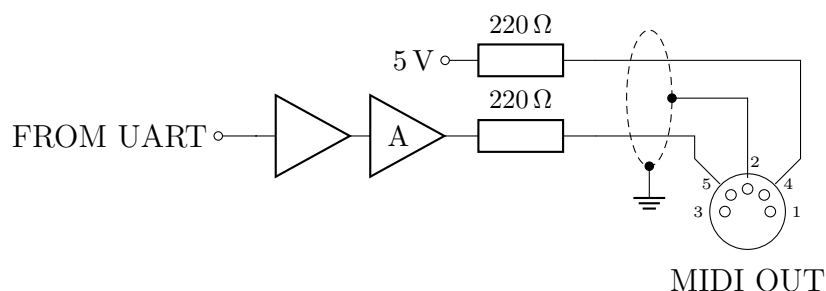
---

<sup>1</sup>Digital Audio Workstation – digitální pracovní stanice pro náběr a úpravu vícestopého záznamu.

<sup>2</sup>Dnes je častější využití USB sběrnice nebo technologie Bluetooth pro obousměrný přenos.

<sup>3</sup>Radio Frequency

<sup>4</sup>Universal Asynchronous Receiver/Transmitter – univerzální asynchronní přijímač/vysílač.



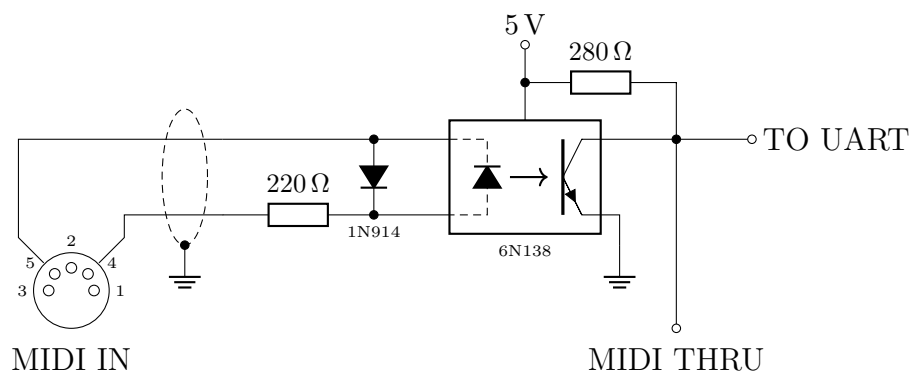
Obr. 1.1: Schéma MIDI výstupu [1].

### 1.1.2 MIDI vstup

Vstupní port MIDI sběrnice je mnohem složitější. Z obr. 1.2 je patrné, že s cílem eliminovat zemní smyčky, které jsou ve zvukové technice nepřijatelné, je vstup každého MIDI zařízení skrz optočlen galvanicky oddělen. Z téhož důvodu je také pin 2 – na rozdíl od výstupního portu – zapojen naprázdno (podle [1]).

Některá zařízení disponují také výstupem MIDI THRU. Ten „kopíruje“ data ze vstupu MIDI IN a tak umožňuje komplexní řetězení zařízení za sebou. Jeho schéma je totožné s tím na obr. 1.1.

Aktualizační norma [2] pak umožňuje přidání feritových jader za piny 4 a 5, přes kondenzátor nízké kapacity uzemnění pinu 2 a při použití 3,3V logiky snížení odporu rezistoru na vstupu optočlenu.



Obr. 1.2: Schéma MIDI vstupu [1].

Bude-li do tohoto vstupu připojen výstup jiného zařízení, které vyšle logickou 1 objeví se na pinech 4 a 5 stejné napětí, v obvodu tedy neprochází žádný proud – LED<sup>5</sup> v optočlenu nesvítí. Vstupní UART čip přijímá logickou 1. Vyšle-li jiné zařízení logickou 0, na pinu 5 poklesne napětí oproti pinu 4, LED v optočlenu se rozsvítí, obvodem prochází proud a UART čip přijímá logickou 1.

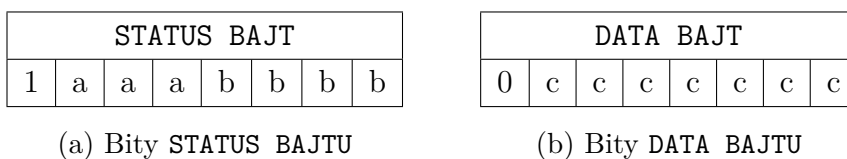
<sup>5</sup>Light-Emitting Diode – Dioda, která vyzařuje viditelné světlo.

## 1.2 Softwarová vrstva

MIDI protokol přenáší informace (příkazy) po sériové lince přenosovou rychlostí 31250 bps. Komunikace je jednosměrná, pro obousměrnou komunikaci je třeba využít dvou rozhraní (vstupu a výstupu) na všech zúčastněných zařízeních.

Příkazy se skládají z bajtů, jichž rozeznáváme dva typy:

- **STATUS BAJT** v sobě kóduje druh příkazu (*Nota stlačena*, *Změna kontroléru*. . .) a cílový kanál (1-16).
- **DATA BAJT** vyjadřuje hodnotu s jakou je příkaz posílán (0-127).

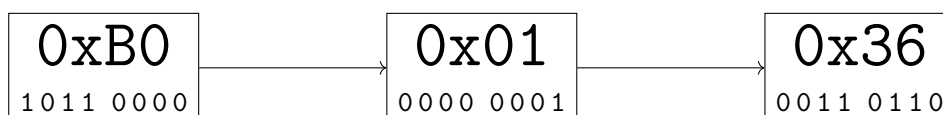


a	bits pro kódování druhu zprávy
b	bits pro kódování kanálu
c	bits pro kódování hodnoty

Obr. 1.3: Bity obou druhů MIDI bajtů [1]

Na obrázku 1.3 je patrná struktura obou bajtů. Za pozornost stojí především jejich MSB<sup>6</sup>: **STATUS BAJT** má MSB vždy s hodnotou 1. Naproti tomu MSB **DATA BAJTU** je vždy nulový.

Pomineme-li výjimky, každý příkaz začíná jedním **STATUS BAJTEM**, za nímž následuje jeden, nebo dva **DATA BAJTY**. Na obr. 1.4 je uvedena struktura ukázkové zprávy.



Obr. 1.4: Struktura zprávy *Změna kontroléru* [1].

Z prvního bajtu zprávy z obr. 1.4 lze dekodovat druh a cílový kanál příkazu. Jedná se o *Control Change* - *Změna kontroléru* na *kanálu 1*<sup>7</sup> Druhý bajt představuje číslo kontroléru (2 – *Modulation*) a třetí jeho hodnotu (54).

Pro potřeby tohoto semestrálního projektu bude nutné, aby každý přípravek dokázal rozeznat kanál k němu přichází MIDI zprávy, popř. jej pro další přenos pozměnil. **DATA BAJTY** budou jen „kopírovány“ a nebude se do nich nijak zasahováno.

<sup>6</sup>Most Significant Bit – Nejvýznamnější bit (většinou v bajtu).

<sup>7</sup>0000<sub>2</sub> ~ kanál 1 ... 1111<sub>2</sub> ~ kanál 16.

### 1.2.1 Výjimky

Je třeba alespoň okrajově zmínit výjimečné stavy, které v rámci MIDI protokolu mohou nastat. V této fázi semestrální práce tyto výjimky zatím nejsou ošetřeny.

#### Running Status

Pracuje-li MIDI vysílač v tomto stavu, neposílá **STATUS BAJT** v opakující se zprávě stejného druhu. Příjímač si tedy musí uložit poslední platný **STATUS BAJT** do paměti, kterou přemaže, obdrží-li zprávu s jiným, novým **STATUS BAJTEM**. Tento mód výrazně šetří přenesená data, zejména je-li použit při odesílání zprávy *Změna kontroly* kontinuálních ovladačů. [1]

#### Zprávy Real-Time

Tyto zprávy slouží pro synchronizaci MIDI zařízení, které pracují s časovou osou nebo časem obecně. Skládají se pouze z jediného bajtu a mají nejvyšší prioritu. Příjímač by měl být připraven i na to, že je obdrží uprostřed standardní zprávy nebo SysEx zprávy. [1]

#### Zprávy SysEx

SysEx (*System Exclusive*) zprávy se skládají z většího a libovolného počtu bajtů. Jsou víceúčelové a univerzální, používají se například pro posun na časové ose (komplementárně ke zprávám **Real-Time**), MSC<sup>8</sup> apod. [1]

---

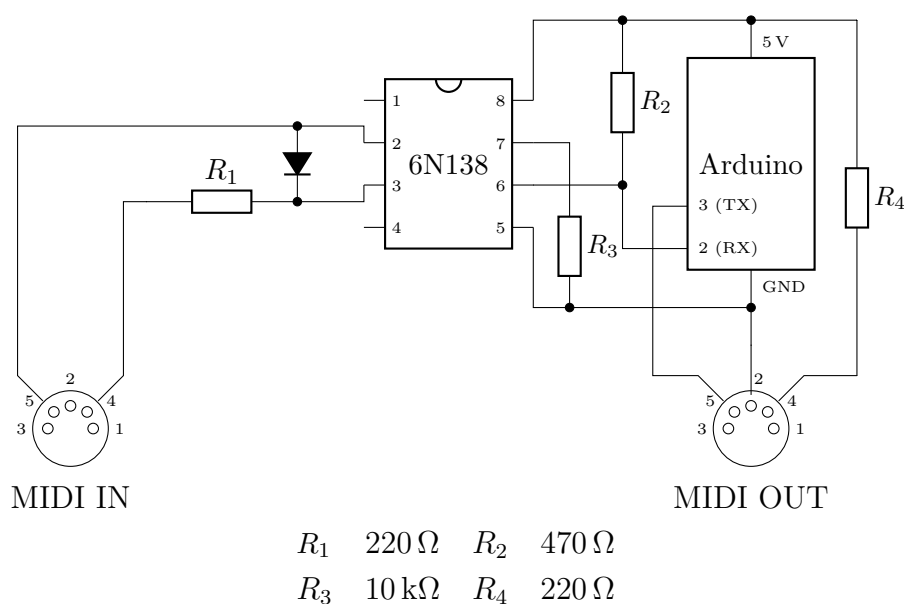
<sup>8</sup>MIDI Show Control – subprotokol pro ovládání scénické techniky, zejména pomocí příkazů GO, STOP, popř. RESUME.

## 2 Návrh přípravku

V rámci semestrální práce byl vytvořen prototyp zařízení na platformě Arduino UNO, která disponuje čipem ATMEL ATmega 328P. Arduino sekunduje Ethernet Shield V1, který je osazen čipem WizNet W5100.

Přípravek umožňuje výměnu MIDI příkazů přes lokální síť, využívá první čtyři vrstvy ISO/OSI modelu. Zároveň poskytuje pokročilé možnosti směrování jednotlivých MIDI kanálů.

### 2.1 Hardware



Obr. 2.1: Schéma prototypu [3]

Z obr. 2.1 je patrné zapojení přípravku. Je využito standardní zapojení MIDI vstupu a výstupu v souladu se schématy na obr. 1.1 a 1.2. Jak již bylo řečeno v kapitole 1, MIDI využívá pro komunikaci sériovou linku. V tomto případě je vstupní linka zapojena k Arduino na pinu 2 a výstupní k pinu 3<sup>1</sup>.

Ve schématu na obr. 2.1 je pro zjednodušení značka „Arduino“ chápána jako „Arduino + Wiznet Shield“.

Přípravek je v této fázi vývoje napájen skrz USB sběrnici Arduina napětím 5 V.

<sup>1</sup>Arduino umožňuje softwarové přepnutí pinů z režimu výstupu na režim vstup a naopak.

## 2.2 Software

Software přípravku byl vyvíjen za účelem maximalizace variability směřování MIDI příkazů. Na funkční rovině je inspirován systémem *Dante*, který se používá pro přenos zvukového signálu přes počítačovou síť. *Dante* umožňuje pokročilé směřování signálů napříč síťovými zařízeními, ovládané pomocí přehledné matice na kontrolním PC. Do této fáze by měl být vyvinut i systém MoE<sup>2</sup>.

### 2.2.1 Databáze spojení

S cílem docílit již zmiňované velké variability ve směřování zpráv napříč přípravky byla do zdrojového kódu implementována tzv. databáze spojení `subscriptions[]`:

```
typedef struct subscription
{
    byte srcdstChannel;
    byte dstIPnib;
} _subscriptions[MAX_SUBS];
```

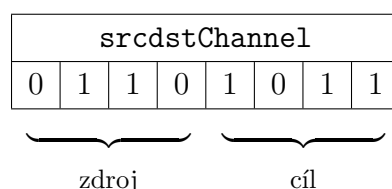
Toto pole struktur `subscription` je pro celý MoE protokol zásadní. Jedno „spojení“, tedy jeden prvek v tomto poli, propojuje jeden MIDI kanál lokálního zařízení s jedním MIDI kanálem cílového zařízení. Toto pole si lze představit jako virtuální patchbay. Jeden patch-kabel odpovídá jednomu prvku v poli (záznamu v databázi).

Každý záznam je tvořen dvěma bajty:

- `srcdstChannel` v sobě kóduje zdrojový MIDI kanál, který se týká lokálního zařízení, a cílový MIDI kanál – ten se týká zařízení cílového. „Kódování“ je uskutečněno způsobem vyobrazeným na obr. 2.2. Jeden kanál je vyjádřen pomocí čtyř bitů (viz kapitolu 1.2), jeden bajt tedy „pojme“ přesně dva kanály.
- `dstIPnib` vyjadřuje poslední bajt IP adresy cílového zařízení (viz též sekci 2.2.3)

---

<sup>2</sup>MIDI over Ethernet – MIDI po Ethernetu



Obr. 2.2: Kódování zdrojového a cílového kanálu v bajtu `srcdstChannel`



### 2.2.2 Formát zprávy

Pro optimální komunikaci zařízení mezi sebou bylo nutné zapouzdřit MIDI příkazy do MoE datagramu. Tento datagram se skládá ze *čtyř* bajtů, z čehož první je vyhrazen MoE značce a zbylé tři samotné MIDI zprávě nebo jiným datům. Tabulku všech dosavadně implementovaných MoE značek lze nalézt v příloze A.2.

### 2.2.3 Síťová komunikace

Pro síťovou komunikaci byl vybrán protokol UDP<sup>3</sup>. Jeho „peer2peer“ architektura je k tomuto projektu mnohem vhodnější než ta, kterou poskytuje protokol TCP. Komunikace probíhá na portu 50 000. MoE zařízení jsou schopna se do lokální sítě připojit s pevně nastavenou IP adresou, nebo požádat o přidělení IP adresy DHCP server. Tato síť však musí mít masku 255.255.255.0 – zařízení se z úsporných důvodů navzájem rozlišují *pouze* posledním bajtem IP adresy (jak je popsáno v sekci 2.2.1).

---

<sup>3</sup>User Datagram Protocol

## 3 Průběh komunikace

V této kapitole bude popsán průběh připojení přípravku k síti, automatické rozpoznání a zacházení s přijatými MIDI a UDP zprávami.

### 3.1 Připojení k lokální síti

Každému přípravku byla do prvních šesti bajtů EEPROM paměti uložena unikátní MAC adresa. Při zapnutí a následné inicializaci je adresa načtena a přiřazena. Nejprve proběhne pokus o získání IP adresy pomocí DHCP serveru sítě. V případě neúspěchu použije přípravek uživatelem napevno nastavenou IP adresu.

Díky knihovně `EthernetUdp.h` jsou výše zmíněné operace otázkou pouze několika málo řádků:

```
if (Ethernet.begin())
{
    //DHCP server zdárně přidělil zařízení IP adresu
    Serial.println(Ethernet.localIP());
}
else
{
    Ethernet.begin(_myMac, _userIP);
    //Přidělení IP adresy napevno
    Serial.println(Ethernet.localIP());
}
```

### 3.2 Upozornění na vlastní přítomnost

Okamžitě po zdárném připojení k lokální síti je vyslána zpráva `beacon`, jejíž účel je upozornit všechna zařízení v lokální síti na vlastní přítomnost. Zpráva se skládá ze čtyř bajtů:

`0xFF, 0xFF, 0xFF, 0xFF`

a je poslána na broadcast adresu sítě.

```
_broadcastIP = Ethernet.localIP();
_broadcastIP[3] = 255;
EthernetUdp eUDP;
eUDP.begin(MOE_PORT);
eUDP.beginPacket(_broadcastIP, MOE_PORT);
```

```
eUDP.write(_beacon, sizeof(_beacon));  
eUDP.endPacket();
```

### 3.3 Rozpoznání zařízení na síti

V případě přijetí zprávy `beacon` je rozpoznána IP adresa odesílatele a vytvořen nový záznam do databáze spojení `subscriptions`. V této fázi projektu přípravek napevno přiřazuje kanál 1 lokální příchozí MIDI sběrnice kanálu 1 výstupní MIDI sběrnice cílového zařízení. V další fázi bude operace přiřazení uskutečňována pravděpodobně přes kontrolní SW na PC připojeného do též sítě.

```
switch(_incomingUDP[0])  
{  
    //...  
    case 0xFF:  
        addSubscription(1, eUDP.remoteIP()[3], 1);  
        break;  
    //...  
}
```

### 3.4 Přijetí MIDI zprávy

Při přijímání MIDI zpráv je nejdříve třeba rozlišit, jedná-li se o zprávu dvoubajtovou, nebo třibajtovou. Toto lze vyřešit jednoduchým větvením `switch`. V dalším kroku je nutné dekodovat kanál MIDI zprávy z `DATA BAJTU` – tuto „extrakci“ lze vyřešit pomocí bitového maskování. Takto získaný bajt je pak porovnáván s prvním nibblem bajtu `srcdstChannel` každého záznamu databáze `subscriptions`. Dojde-li ke shodě, začíná konstrukce odchozí zprávy podle dalších informací v odpovídajícím záznamu databáze spojení. Výpis kódu je k dispozici jako příloha B.1

### 3.5 Přijetí UDP zprávy

Díky architektuře odesílané zprávy (viz podkapitolu 3.4) je veškeré směřování dokončeno již na straně odesílatele. Přijímač může obdrženou zprávu tedy rovnou poslat do MIDI výstupu.

# **Závěr**

Shrnutí studentské práce.

# Literatura

- [1] THE MIDI MANUFACTURERS ASSOCIATION. *MIDI 1.0 Detailed Specification*. Document Version 4.2.1. Los Angeles, CA: MMA, 1996.
- [2] MMA TECHNICAL STANDARDS BOARD/AMEI MIDI COMMITTEE. *MIDI 1.0 Electrical Specification Update*. 2014.
- [3] GHASSAEI, Amanda. *Send and Receive MIDI With Arduino* [online]. [cit. 1. 12. 2020]. Dostupné z URL: <<https://www.instructables.com/Send-and-Receive-MIDI-with-Arduino/>>.
- [4] VUT v Brně: *Úprava, odevzdávání a zveřejňování vysokoškolských kvalifikačních prací na VUT v Brně* [online]. Směrnice rektora č. 2/2009. Brno: 2009, poslední aktualizace 24. 3. 2009 [cit. 23. 10. 2015]. Dostupné z URL: <<https://www.vutbr.cz/uredni-deska/vnitрни-predpisy-a-dokumenty/smernice-rektora-f34920/>>.
- [5] ČSN ISO 690 (01 0197) *Informace a dokumentace – Pravidla pro bibliografické odkazy a citace informačních zdrojů*. 40 stran. Praha: Český normalizační institut, 2011.
- [6] ČSN ISO 7144 (010161) *Dokumentace – Formální úprava disertací a podobných dokumentů*. 24 stran. Praha: Český normalizační institut, 1997.
- [7] ČSN ISO 31-11 *Veličiny a jednotky – část 11: Matematické znaky a značky používané ve fyzikálních vědách a v technice*. Praha: Český normalizační institut, 1999.
- [8] BIERNÁTOVÁ, O., SKŮPA, J.: *Bibliografické odkazy a citace dokumentů dle ČSN ISO 690 (01 0197) platné od 1. dubna 2011* [online]. 2011, poslední aktualizace 2. 9. 2011 [cit. 19. 10. 2011]. Dostupné z URL: <<http://www.citace.com/CSN-ISO-690.pdf>>
- [9] *Pravidla českého pravopisu*. Zpracoval kolektiv autorů. 1. vydání. Olomouc: FIN PUBLISHING, 1998. 575 s. ISBN 80-86002-40-3.
- [10] WALTER, G. G.; SHEN, X. *Wavelets and Other Orthogonal Systems*. 2. vyd. Boca Raton: Chapman & Hall/CRC, 2000. 392 s. ISBN 1-58488-227-1
- [11] SVAČINA, J. Dispersion Characteristics of Multilayered Slotlines – a Simple Approach. *IEEE Transactions on Microwave Theory and Techniques*, 1999, vol. 47, no. 9, s. 1826–1829. ISSN 0018-9480.

- [12] RAJMIC, P.; SYSEL, P. Wavelet Spectrum Thresholding Rules. In *Proceedings of the International Conference Research in Telecommunication Technology*, Žilina: Žilina University, 2002. s. 60–63. ISBN 80-7100-991-1.

# Seznam symbolů, veličin a zkratek

<b>MIDI</b>	Musical Instrument Digital Interface – digitální rozhraní hudebního nástroje
<b>DAW</b>	Digital Audio Workstation – digitální pracovní stanice pro náběr a úpravu vícestopého záznamu.
<b>USB</b>	Universal Serial Bus – univerzální sériová sběrnice pro komunikace hosta s jeho periferiemi.
<b>RF</b>	Radio Frequency
<b>UART</b>	Universal Asynchronous Receiver/Transmitter – univerzální asynchronní přijímač/vysílač.
<b>LED</b>	Light-Emitting Diode – Dioda, která vyzařuje viditelné světlo.
<b>MSB</b>	Most Significant Bit – Nejvýznamnější bit (většinou v bajtu).
<b>MSC</b>	MIDI Show Control – subprotokol pro ovládání scénické techniky, zejména pomocí příkazů GO, STOP, popř. RESUME.
<b>MoE</b>	MIDI over Ethernet – MIDI po Ethernetu
<b>UDP</b>	User Datagram Protocol
<b>Šířka levého sloupce Seznamu symbolů, veličin a zkratek</b> je určena šířkou parametru prostředí <b>acronym</b> (viz řádek 1 výpisu zdrojáku na str. 30)	
<b>KolikMista</b>	pouze ukázka vyhrazeného místa
<b>DSP</b>	číslicové zpracování signálů – Digital Signal Processing
$f_{vz}$	vzorkovací kmitočet

# Seznam příloh

<b>A</b>	<b>Tabulky</b>	<b>24</b>
A.1	MIDI STATUS BAJTY . . . . .	24
A.2	MoE značky . . . . .	25
<b>B</b>	<b>Výpisy kódu</b>	<b>26</b>
B.1	Výpis kódu pro příjem a zpracování MIDI zprávy . . . . .	26
<b>C</b>	<b>Některé příkazy balíčku thesis</b>	<b>28</b>
C.1	Příkazy pro sazbu veličin a jednotek . . . . .	28
C.2	Příkazy pro sazbu symbolů . . . . .	28
<b>D</b>	<b>Druhá příloha</b>	<b>29</b>
<b>E</b>	<b>Příklad sazby zdrojových kódů</b>	<b>30</b>
E.1	Balíček listings . . . . .	30
<b>F</b>	<b>Obsah přiloženého CD</b>	<b>34</b>



# A Tabulky

## A.1 MIDI STATUS BAJTY

Název		Hex. hodnota	Bin. hodnota
Note-Off	Nota vypnuta	0x8n	1000 nnnn
Note-On	Nota zapnuta	0x9n	1001 nnnn
Poly Key Pressure	Polyfonický tlakový ovladač	0xA n	1010 nnnn
Control Change	Změna kontroléru	0xB n	1011 nnnn
Program Change	Změna programu	0xC n	1100 nnnn
Channel Pressure	Monofonický tlakový ovladač	0xD n	1101 nnnn
Pitch Bend	Ohyb výšky tónu	0xE n	1110 nnnn

Tab. A.1: Tabulka MIDI STATUS BAJTŮ [1].

## A.2 MoE značky

Bajt	Popis
⋮	
0x08	Broadcast zpráva od řídicího PC, která sonduje účastníky sítě
⋮	
0x0F	Unicast zpráva od řídicího PC, která zařízení vkládá nový záznam do databáze <b>subscriptions</b> , který je obsahem datagramu
⋮	
0x80	Unicast odpověď na sondáž PC, součástí datagramu je i jeden záznam databáze <b>subscriptions</b>
⋮	
0xA2	Dvoubajtová MIDI zpráva pro port A
0xA3	Třibajtová MIDI zpráva pro port A
0xB2	Dvoubajtová MIDI zpráva pro port B
0xB3	Třibajtová MIDI zpráva pro port B
⋮	
0xFF	Zpráva <b>beacon</b> sloužící pro upozornění na vlastní přítomnost

Tab. A.2: Tabulka prvních bajtů MoE zpráv.

## B Výpisy kódu

### B.1 Výpis kódu pro příjem a zpracování MIDI zprávy

```
void handleMIDI()
{
    int numIncBytes = midiSerial.available();
    if (numIncBytes)
    {
        switch (numIncBytes)
        {
            //...
            case 3:
                byte data0, data1, data2;
                data0 = midiSerial.read();
                data1 = midiSerial.read();
                data2 = midiSerial.read();
                sendUDP(data0, data1, true, data2);
                break;
            //...
        }
    }
}

void sendUDP(byte data0, byte data1, bool threeByte,
             byte data2)
{
    byte srcCh = data0 & 0x0F;
    for (byte i = 0; i < MAX_SUBS; i++)
    {
        if (srcCh == (_subscriptions[i].srcdstChannel
                     & 0xF0) >> 4)
        {
            data0 = data0 & 0xF0;
            data0 = data0 | (_subscriptions[i].srcdstChannel
                           & 0x0F);

            IPAddress remoteIP = Ethernet.localIP();
            remoteIP[3] = _subscriptions[i].dstIPnib;
            eUDP.beginPacket(remoteIP, MOE_PORT);
            eUDP.write(0xAA);
            eUDP.write(data0);
        }
    }
}
```

```
eUDP.write(data1);  
if (threeByte)  
    eUDP.write(data2);  
eUDP.endPacket();  
}  
}  
}
```

## C Některé příkazy balíčku thesis

### C.1 Příkazy pro sazbu veličin a jednotek

Tab. C.1: Přehled příkazů pro matematické prostředí

Příkaz	Příklad	Zdroj příkladu	Význam
<code>\textind{...}</code>	$\beta_{\max}$	<code>\$\beta_{\textind{max}}\$</code>	textový index
<code>\const{...}</code>	$U_{\text{in}}$	<code>\$\const{U}_{\textind{in}}\$</code>	konstantní veličina
<code>\var{...}</code>	$u_{\text{in}}$	<code>\$\var{u}_{\textind{in}}\$</code>	proměnná veličina
<code>\complex{...}</code>	$u_{\text{in}}$	<code>\$\complex{u}_{\textind{in}}\$</code>	komplexní veličina
<code>\vect{...}</code>	$\mathbf{y}$	<code>\$\vect{y}\$</code>	vektor
<code>\mat{...}</code>	$\mathbf{Z}$	<code>\$\mat{Z}\$</code>	matice
<code>\unit{...}</code>	kV	<code>\$\unit{kV}\$</code> či <code>\unit{kV}</code>	jednotka

### C.2 Příkazy pro sazbu symbolů

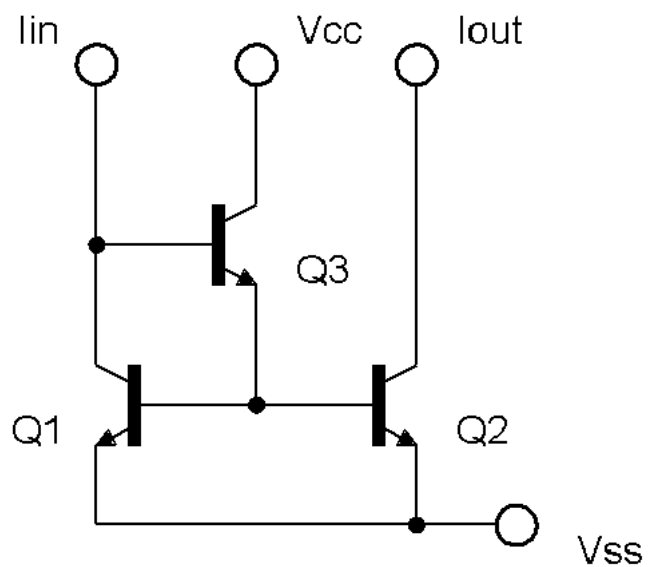
- `\E`, `\eul` – sazba Eulerova čísla:  $e$ ,
- `\J`, `\jmag`, `\I`, `\imag` – sazba imaginární jednotky:  $j$ ,  $i$ ,
- `\dif` – sazba diferenciálu:  $d$ ,
- `\sinc` – sazba funkce:  $\text{sinc}$ ,
- `\mikro` – sazba symbolu mikro stojatým písmem<sup>1</sup>:  $\mu$ ,
- `\uppi` – sazba symbolu  $\pi$  (stojaté řecké pí, na rozdíl od `\pi`, což sází  $\pi$ ).

Všechny symboly jsou určeny pro matematický mód, vyjma `\mikro`, jenž je použitelný rovněž v textovém módu.

---

<sup>1</sup>znak pochází z balíčku `textcomp`

## D Druhá příloha



Obr. D.1: Zlepšené Wilsonovo proudové zrcadlo.

Pro sazbu vektorových obrázků přímo v  $\text{\LaTeX}$ u je možné doporučit balíček `TikZ`. Příklady sazby je možné najít na `\TeX`ample. Pro vyzkoušení je možné použít programy `QTikz` nebo `TikzEdt`.

# E Příklad sazby zdrojových kódů

## E.1 Balíček listings

Pro vysázení zdrojových souborů je možné použít balíček `listings`. Balíček zavádí nové prostředí `lstlisting` pro sazbu zdrojových kódů, jako například:

```
\section{Balíček lstlistings}
Pro vysázení zdrojových souborů je možné použít
  balíček \href{https://www.ctan.org/pkg/listings}%
  {\texttt{listings}}.
Balíček zavádí nové prostředí \texttt{lstlisting} pro
  sazbu zdrojových kódů.
```

Podporuje množství programovacích jazyků. Kód k vysázení může být načítán přímo ze zdrojových souborů. Umožňuje vkládat čísla řádků nebo vypisovat jen vybrané úseky kódu. Např.:

Zkratky jsou sázeny v prostředí `acronym`:

```
6 \begin{acronym}[KolikMista]
```

Šířka textu volitelného parametru `KolikMista` udává šířku prvního sloupce se zkratkami. Proto by měla být zadávána nejdelší zkratka nebo symbol. Příklad definice zkratky  $f_{vz}$  je na výpisu E.1.

Výpis E.1: Ukázka sazby zkratek

```
40 \acro{symfvz} % název
41 [\ensuremath{f_{\text{vz}}}] % symbol
42 {vzorkovací kmitočet} % popis
```

Ukončení seznamu je provedeno ukončením prostředí:

```
26 {User Datagram Protocol}
27 \acro{zkTemp} % název
28 [Šířka levého sloupce Seznamu symbolů, veličin a zkratek]
29 {je určena šířkou parametru prostředí \texttt{acronym} (viz řádek~1
30 % rozvinutí zkratky
31
32 \acro{zkDummy}
33 [KolikMista]
34 {pouze ukázka vyhrazeného místa}
35
36 \acro{DSP} % název/zkratka
37 {číslicové zpracování signálů -- Digital Signal Processing}
```

```

38                                     % rozvinutí zkratky
39   %%% bsymfvz
40   \acro{symfvz}                    % název
41   [\ensuremath{f_\text{vz}}] % symbol
42   {vzorkovací kmitočet}          % popis
43   %%% esymfvz
44
45 \end{acronym}

```

#### **Poznámka k výpisům s použitím volby jazyka czech nebo slovak:**

Pokud Váš zdrojový kód obsahuje znak spojovníku -, pak překlad může skončit chybou. Ta je způsobená tím, že znak - je v českém nebo slovenském nastavení balíčku babel tzv. aktivním znakem. Přepněte znak - na neaktivní příkazem `\shorthandoff{-}` těsně před výpisem a hned za ním jej vraťte na aktivní příkazem `\shorthandon{-}`. Podobně jako to je ukázáno ve zdrojovém kódu šablony.



Na výpisu E.2 naleznete příklad kódu pro Matlab, na výpisu E.3 zase pro jazyk C.

Výpis E.2: Příklad Schur-Cohnova testu stability v prostředí Matlab.

```
1 %% Příklad testování stability filtru
2
3 % koeficienty polynomu ve jmenovateli
4 a = [ 5, 11.2, 5.44, -0.384, -2.3552, -1.2288];
5 disp( 'Polynom:'); disp(poly2str( a, 'z'))
6
7 disp('Kontrola pomocí kořenu polynomu:');
8 zx = roots( a);
9 if( all( abs( zx) < 1))
10     disp('System je stabilní')
11 else
12     disp('System je nestabilní nebo na mezí stability');
13 end
14
15 disp(' '); disp('Kontrola pomocí Schur-Cohn:');
16 ma = zeros( length(a)-1,length(a));
17 ma(1,:) = a/a(1);
18 for( k = 1:length(a)-2)
19     aa = ma(k,1:end-k+1);
20     bb = fliplr( aa);
21     ma(k+1,1:end-k+1) = (aa-aa(end)*bb)/(1-aa(end)^2);
22 end
23
24 if( all( abs( diag( ma.'))))
25     disp('System je stabilní')
26 else
27     disp('System je nestabilní nebo na mezí stability');
28 end
```

Výpis E.3: Příklad implementace první kanonické formy v jazyce C.

```
// první kanonická forma
short fxdf2t( short coef[][5], short sample)
{
    static int v1[SECTIONS] = {0,0}, v2[SECTIONS] = {0,0};
    int x, y, accu;
    short k;

    x = sample;
    for( k = 0; k < SECTIONS; k++){
        accu = v1[k] >> 1;
        y = _sadd( accu, _smpy( coef[k][0], x));
        y = _sshl(y, 1) >> 16;

        accu = v2[k] >> 1;
        accu = _sadd( accu, _smpy( coef[k][1], x));
        accu = _sadd( accu, _smpy( coef[k][2], y));
        v1[k] = _sshl( accu, 1);

        accu = _smpy( coef[k][3], x);
        accu = _sadd( accu, _smpy( coef[k][4], y));
        v2[k] = _sshl( accu, 1);

        x = y;
    }
    return( y);
}
```

## F Obsah přiloženého CD

Nezapomeňte uvést, co čtenář najde na přiloženém médiu. Je vhodné okomentovat obsah každého adresáře, specifikovat, který soubor obsahuje důležitá nastavení, který soubor je určen ke spuštění atd. Také je dobře napsat, v jaké verzi software byl kód testován (např. Matlab 2010b).

Pokud je souborů hodně a jsou organizovány ve více složkách, je možné pro výpis adresářové struktury použít balíček `dirtree`.

```
/ ..... kořenový adresář přiloženého CD
├── logo ..... loga školy a fakulty
│   ├── BUT_abbreviation_color_PANTONE_EN.pdf
│   ├── BUT_color_PANTONE_EN.pdf
│   ├── FEEC_abbreviation_color_PANTONE_EN.pdf
│   ├── FEKT_zkratka_barevne_PANTONE_CZ.pdf
│   ├── UTKO_color_PANTONE_CZ.pdf
│   ├── UTKO_color_PANTONE_EN.pdf
│   ├── VUT_barevne_PANTONE_CZ.pdf
│   ├── VUT_symbol_barevne_PANTONE_CZ.pdf
│   └── VUT_zkratka_barevne_PANTONE_CZ.pdf
├── obrazky ..... ostatní obrázky
│   ├── soucastky.png
│   ├── spoje.png
│   ├── ZlepseneWilsonovoZrcadloNPN.png
│   └── ZlepseneWilsonovoZrcadloPNP.png
├── pdf ..... pdf stránky generované informačním systémem
│   ├── student-desky.pdf
│   ├── student-titulka.pdf
│   └── student-zadani.pdf
├── text ..... zdrojové textové soubory
│   ├── literatura.tex
│   ├── prilozy.tex
│   ├── reseni.tex
│   ├── uvod.tex
│   ├── vysledky.tex
│   ├── zaver.tex
│   └── zkratky.tex
├── sablona-obhaj.tex ..... hlavní soubor pro sazbu prezentace k obhajobě
├── sablona-prace.tex ..... hlavní soubor pro sazbu kvalifikační práce
└── thesis.sty ..... balíček pro sazbu kvalifikačních prací
```