



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## MIDI SEKVENCÉR PRO HUDEBNÍ NÁSTROJ

MIDI SEQUENCER FOR A MUSICAL INSTRUMENT

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Dominik Golian

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Zdeněk Bradáč, Ph.D.

BRNO 2024

# Diplomová práce

magisterský navazující studijní program **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

**Student:** Bc. Dominik Golian

**ID:** 220979

**Ročník:** 2

**Akademický rok:** 2023/24

**NÁZEV TÉMATU:**

## MIDI sekvencér pro hudební nástroj

### POKYNY PRO VYPRACOVÁNÍ:

Prozkoumejte a uveďte do funkčního stavu hudební nástroj metalofon. Navrhněte a zrealizujte paralelní elektronické řízení nástroje k současnému elektronickému řídicímu systému. Využijte již existující MIDI kontrolér pro foukací harmoniku. Navrhněte a zrealizujte elektronický MIDI sekvencér pro přehrávání MIDI souborů současně na foukací harmonice a metalofonu. Navrhněte a realizujte HW, oživte a otestujte. Vytvořte programové vybavení včetně implementace přehrávání MIDI souborů s nadřazeného systému. Demonstrujte správnou funkci.

1. Proveďte internetový průzkum a literární rešerši.
2. Navrhněte koncepci systému a zapojení jeho komponent.
3. Opravte a oživte metalofon. Připojte paralelní řídicí systém.
4. Navrhněte obvodové schéma MIDI sekvencéru.
5. Zrealizujte elektroniku MIDI sekvencéru, osadte a oživte.
6. Vytvořte programové vybavení.
7. Zhodnoťte dosažené výsledky.

### DOPORUČENÁ LITERATURA:

Patrick Hood-Daniel. Build Your Own CNC Machine. Apress, 2009. ISBN: 978-1430224891.

Dle pokynů vedoucího práce.

**Termín zadání:** 5.2.2024

**Termín odevzdání:** 15.5.2024

**Vedoucí práce:** doc. Ing. Zdeněk Bradáč, Ph.D.

**doc. Ing. Petr Fiedler, Ph.D.**  
předseda rady studijního programu

### UPOZORNĚNÍ:

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

## **ABSTRAKT**

Hudba, ako univerzálny jazyk ľudstva ktorým vyjadrujeme emócie, myšlienky, skúsenosti prechádza nepretržitou evolúciou vďaka technologickému pokroku. Táto práca sa venuje jednému z najväčších technologických milníkov, vzniku MIDI. Náplň tejto práce je tvorená vyhotovením paralelného MIDI systému, ktorý bude interagovať s jedným z hudobných nástrojov, metalofónom alebo harmonikou. Súčasťou práce je návrh a vyhotovenie takéhoto zariadenia.

## **KĽÚČOVÉ SLOVÁ**

MIDI, mikrokontrolér, zariadenie, komunikácia, byte, správa

## **ABSTRACT**

Music, as the universal language of humanity through which we express emotions, thoughts, experiences, undergoes continuous evolution, also thanks to technological advancement. This work focuses on one of the greatest technological milestones, the creation of MIDI. The scope of this work involves the development of a parallel MIDI system, which will interact with one of the musical instruments, either a metalophone or a mouth organ. The thesis includes the design and implementation of such a device.

## **KEYWORDS**

MIDI, microcontroller, device, communication, byte, message

GOLIAN, Dominik. *MIDI sekvencér pro hudební nástroj*. Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2024. Vedúci práce: prof. Ing. Zdeněk Bradáč, Ph.D.



## Vyhlásenie autora o pôvodnosti diela

**Meno a priezvisko autora:** Bc. Dominik Golian  
**VUT ID autora:** 220979  
**Typ práce:** Diplomová práca  
**Akademický rok:** 2023/24  
**Téma záverečnej práce:** MIDI sekvencér pro hudební nástroj

Vyhlasujem, že svoju záverečnú prácu som vypracoval samostatne pod vedením vedúcej/cého záverečnej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej záverečnej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno .....

.....

podpis autora\*

---

\*Autor podpisuje iba v tlačenej verzii.

## POĎAKOVANIE

Rád bych poděkoval vedoucímu semestrální práce panu prof. Ing. Zdeňkovi Bradáčovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci a panu Ing. Ondřejovi Baštánovi za odborné rady.

# Obsah

Úvod	11
<b>1 MIDI Systém</b>	<b>12</b>
1.1 Vznik MIDI	12
1.2 MIDI 1.0 štandard	14
1.2.1 MIDI správy	14
1.2.2 MIDI režimi	21
1.2.3 MIDI hardware	23
1.2.4 Dodatok normy k 3.3V systémom	24
1.2.5 Prepájanie MIDI systémov	26
1.2.6 MIDI 2.0 štandard	28
<b>2 Prieskum trhu</b>	<b>30</b>
<b>3 Návrh systému</b>	<b>32</b>
3.1 Hudobné nástroje	32
3.2 Požiadavky pre návrh systému	34
3.2.1 Ovládanie cez USB	34
<b>4 Návrh hardvéru</b>	<b>36</b>
4.1 Výber mikrokontroléru	36
4.2 Ovládanie solenoidov	37
4.3 Ovládanie regulátoru tlaku	38
4.4 Komunikačné rozhrania	40
4.4.1 MIDI	40
4.4.2 USB	42
4.5 Ovládacie prvky	44
4.6 Návrh dosky plošných spojov	46
<b>5 Implementácia softvéru</b>	<b>48</b>
5.1 MIDI subsystém	51
5.1.1 Trieda RunningStatus	51
5.1.2 Trieda MidiDevice	54
5.2 Ovládanie výstupov	54
5.3 Zobrazovanie a ovládanie	57
<b>6 Výsledky práce</b>	<b>60</b>
<b>Záver</b>	<b>63</b>

Literatúra	65
Zoznam symbolov a skratiek	67
Zoznam príloh	68
A Schéma zapojenia	69
B Doska plošných spojov	71
C Obsah elektronickej prílohy	73

# Zoznam obrázkov

1.1	Rozdelenie MIDI správ . . . . .	15
1.2	Štruktúra MIDI správ . . . . .	15
1.3	Rozdelenie MIDI bytov . . . . .	16
1.4	Režimi MIDI zariadenia . . . . .	22
1.5	Zapojenie MIDI rozhrania . . . . .	24
1.6	Zapojenie MIDI rozhrania pre 3.3V systémy . . . . .	25
1.7	Prepojenie MIDI zariadení . . . . .	27
1.8	Príklad chybného prepojenia zariadení . . . . .	28
3.1	Metalofón . . . . .	32
3.2	Fúkacia harmonika osadená na prípravku so solenoidmi . . . . .	33
3.3	Bloková schéma konceptu systému . . . . .	34
4.1	Zapojenie napájacej časti . . . . .	38
4.2	Obvod pre ovládanie regulátora tlaku . . . . .	39
4.3	Zapojenie MIDI rozhrania . . . . .	41
4.4	Zapojenie UART prevodníku . . . . .	44
4.5	Zapojenie konektoru pre display a rotačný enkodér . . . . .	45
4.6	Zapojenie napájacej časti . . . . .	46
4.7	Model dosky plošných spojov . . . . .	47
5.1	Topológia systému . . . . .	49
5.2	Výpis hlášok zo systému . . . . .	50
5.3	Diagram stavového automatu . . . . .	53
5.4	Rozdelenie intervalu pre noty . . . . .	57
6.1	Midi sekvencér . . . . .	61

# Zoznam tabuliek

1.1	Kanálové správy . . . . .	17
1.2	Spoločné systémové správy . . . . .	19
1.3	Správy reálneho času . . . . .	20
1.4	Zvláštne systémové správy . . . . .	21
4.1	Parametre tranzistoru pre spínanie solenoidov . . . . .	37

# Úvod

Hudba, ako neoddeliteľná súčasť ľudskej kultúry a histórie prešla dlhou cestou od prvých rytmov na bubnoch počas pradávnych rituálov až po sofistikované symfónie, ktoré vychádzajú z digitálnych zariadení dnešnej doby. Tento druh umenia predstavuje univerzálny jazyk a prechádza nepretržitou evolúciou aj vďaka technologickému pokroku.

Táto práca sa venuje jednej z najvýznamnejších technologických inovácií v hudbe - MIDI (Musical Instrument Digital Interface). MIDI, od svojho vzniku v 80. rokoch, hlboko ovplyvnilo spôsob komunikácie hudobníkov s nástrojmi, a ako sú nástroje navzájom prepojené. Tento štandard je nielen základnou súčasťou moderného hudobného výrobného priemyslu, stal sa aj kľúčovým nástrojom pre tvorivú expresiu v digitálnej ére.

Hlavným cieľom je navrhnuť zariadenie, ktoré bude komunikovať prostredníctvom MIDI a interagovať s hudobnými nástrojmi, harmonikou a metalofónom. Tento návrh sa nezaobíde bez kľúčových vedomostí z oblasti MIDI, a preto ako prvé budú vysvetlené dôležité aspekty tohoto štandardu pre návrh zariadenia. Ďalším bodom je urobiť prieskum trhu, ktorý môže poskytnúť podnetné body pre návrh zariadenia.

Na základe týchto poznatkov bude vypracovaný vhodný koncept zariadenia a jeho riešenia, čo predstavuje návrh hardvéru a softvéru. Výstupom práce by mal byť prístroj schopný hrať podľa povelov na jednom z hudobných nástrojov a tieto riadiace informácie zdieľať s ďalšími systémami.

# 1 MIDI Systém

Táto kapitola sa zaoberá vývojom a evolúciou pre-MIDI systémov a ich vplyvom na súčasnú hudobnú technológiu. Začína sa opisom systémov, ktoré boli používané pred zavedením MIDI, a ako tieto riešenia ovplyvnili potrebu štandardizovanejšieho komunikačného protokolu. Následne sa podrobne venuje samotnému MIDI štandardu, rozoberá technické špecifikácie, funkcionality a možnosti, ktoré MIDI ponúka a zameriava sa aspekty, ktoré sú dôležité pre návrh tejto práce. Záver kapitoly je venovaný súčasnému stavu MIDI štandardu, jeho nedávnym aktualizáciám a rozšíreniam.

## 1.1 Vznik MIDI

Prvé pokusy umožniť komunikáciu medzi viacerými elektronickými hudobnými zariadeniami a medzi počítačmi siahajú do polovice 50. rokov 20. storočia. Šlo hlavne o snahu paralelne prepojiť viacero hudobných nástrojov, hrajúcich niekoľko zvukových farieb súčasne, pričom by všetky zdieľali rovnaké riadiace informácie. Tieto rozhrania slúžili pre nahrávanie riadiacích dát do počítača v reálnom čase (editácia, prehrávanie, prevod na štandardnú notáciu) a automatizovanú hru (diaľkové riadenie nástrojov). [1]

Snahy o automatizovanú hru začali už v 16. storočí. Jednalo sa o zvonohry, flašiny, orchestrióny a iné automatofóny riadené dernými štítkami a kolíkovými bubnami, vo funkcii pamäti. V roku 1963 vznikali prvé analógové sekvencery riadené napäťovými signálmi (*Control Voltage – riadiace napätie* (CV) a Trig signálmi). Vďaka tomuto štandardu bolo možné nástroje riadiť a prepájať medzi sebou. Pozdejšie vznikali analógovo-digitálne sekvencery.[1]

Všetky tieto zariadenia boli pomerne prosté a neponúkali mnoho možností. Dali sa s nimi vytvárať len jednoduché, opakujúce sa rady hudobných motívov. Prepojením viacerých sekvencérov bola obmedzená možnosť polyfónie. Aj keď mali analógovo-digitálne sekvencery isté obmedzenie, boli veľmi často používané. [1]

Omnoho väčší význam mali sériové komunikačné systémy a mikroprocesory z 80. rokov. Štandardné sériové rozhrania a analógovo-digitálne prevodníky boli výbavou množstva mikroprocesorov a tak mohli vzniknúť hardwarovo nenáročnejšie systémy. Asi najvýraznejším pre-MIDI systémom bol *Digital Communication Bus – digitálne komunikačné rozhranie* (DCB) od firmy Roland. Zariadenia s DCB systémom sa prepájali 4 žilovým káblom pre jednosmernú komunikáciu a 7 žilovým káblom pre obojsmernú komunikáciu. 10 bitové slová (štartbit, 8 dátových bitov, stopbit a bit párnej parity) boli prenášané s kmitočtom 31,25 kBaudov. Ako prvý sa vysielal stavový byte, potom rôznych počet dátových bytov. [1]



Napriek tomu že tento systém prenášal len informácie o stlačení, pustení klávesy a zmene registru(programu), dochádzalo k určitým problémom s kompatibilitou rôznych DCB prístrojov. Niektoré zariadenia prenášali informácie plynule, ostatné ich mohli posielat prerušovane. Takisto existoval rôzny spôsob priradenia kláves hlasovým jednotkám, pričom niektoré zariadenia mohli prijať maximálne 8 hlasov naraz. Oproti ostatným systémom bol jediný z komerčne dostupných, ale nikdy sa neujal. Aj cez drobné nedostatky tohto systému, ukázal ďalšiu cestu pre MIDI. [1]

Veľké množstvo komunikačných rozhraní viedlo k nekompatibilitate medzi zariadeniami, kde každý výrobca používal svoj vlastný komunikačný protokol. To spôsobilo problémy nielen medzi prístrojmi rozličných firiem, ale aj medzi zariadeniami tej istej značky, ale inej rady. [1]

Nazvdory týmto problémom, v roku 1981 zasadli predstavitelia hudobných firiem (SEQUENTIAL CIRCUITS, OBERHEIM, ROLAND . . .), aby prejednali predbežné návrhy na univerzálny interface hudobných nástrojov. Vypracovali predbežné riešenia, ku ktorým sa neskôr pripojili aj ďalšie veľké firmy. Ešte ten istý rok vyšiel prvý ucelený návrh - *Universal Synthetizezer Interfase - univerzálne rozhranie syntetizátorov* (USI). Avšak nebol oň veľký záujem. [1]

Na základe pripomienok k USI, v roku 1982 prebehlo ďalšie zasadnutie, kde boli pridané nové úpravy a pripomienky. K tomuto zasadnutiu sa pripojilo mnoho amerických a japonských firiem (celkom 15) a koncept nového štandardu sa začal rysovať. Bolo dosiahnuté konsenzu medzi firmami a nový štandard dostal meno *Musical Instument Digital Interface - digitálne rozhranie hudobných nástrojov* (MIDI). Konečná verzia normy bola poskytnutá výrobcom v podobe MIDI normy 1.0, predstavenej 5.8.1983. Ešte v tomto roku vyšli prvé zariadenia s podporou tejto normy a čoskoro na to vypukla midimánia. [1]

Pretože bol štandard v niektorých oblastiach nejasný, výrobcovia si ho interpretovali rôzne. Kvôli finančným dôvodom, sa všetci výrobcovia snažili implementovať štandard čo najskôr. Znovu došlo k problémom s kompatibilitou, a tak vznikli normatívne orgány zodpovedajúce za dodržanie MIDI normy a jej vývoj (MMA a JMSC). [1]

V roku 1985 vyšla podrobná MIDI norma, odvtedy sú problémy s kompatibilitou skôr výnimkou. Prvé dodatky normy odrážajú vývoj v kontroleroch a vývoj v univerzálnych zväštných systémových dátach. Ďalšie dodatky opisovali *MIDI Time Code - MIDI časový kód* (MTC), Standart MIDI File Format - štandardné MIDI súbory (SMFF), prenos MIDI súborov a General MIDI (obecné MIDI). Jednou z oblastí ktorá nieje dopodrobna štandardizovaná, sú exkluzívne systémové dáta (SysEx). Aj napriek tomu sa MIDI systém stal priemyslovým štandardom, ktorý žiadny výrobca nemohol ignorovať. [1]

## 1.2 MIDI 1.0 štandard

Predstavuje dnes systém, ktorý sa stal svetovým priemyslovým štandardom. Aj napriek limitáciám a nedokonalostiam, pracuje veľmi dobre a ovplyvnil podobu dnešnej elektronickej hudby. Tvorí ho fyzické prepojenie, spôsob kódovania a prenos informácií v systéme. Vo svojej podstate, sa jedná len o upravené komunikačné rozhranie RS-232. [1]

MIDI býva niekedy nazývané aj sieťou, v skutočnosti sa o sieť nejedná. Norma údava, že sa jedná o jednosmerný sériový komunikačný protokol. Umožňuje prepojenie rôznych hudobných zariadení (syntetizátory, sekvencery, rytmy a iné). Hardvér je tvorený mikroprocesorom s interným alebo externým prevodníkom UART. [1]

Táto norma taktiež definuje konektory pre komunikáciu a samozrejme aj prepájacie káble. Jedná sa o konektory typu DIN a definuje:

- MIDI In - vstup dát
- MIDI Out - výstup dát
- MIDI Thru - bufferovaný prechod dát.

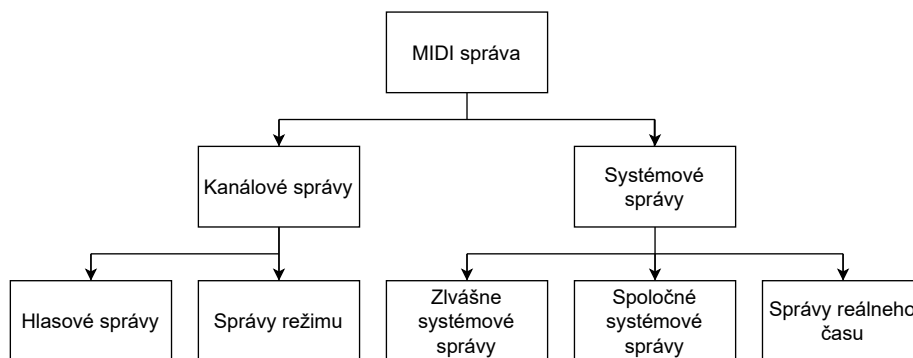
Tienenie káblu sa zapája len na jednej strane, aby nedochádzalo k zemným smyčkám medzi zariadeniami. Informácie sa odosielajú asynchrónne, pri prenosovej rýchlosti 31250 Baud/s (1MHz/32), a to v prúdovej slučke. Tečúci prúd signalizuje logickú nulu, vypnutý stav signalizuje logickú jednotku. Každé zariadenie má na svojom vstupe galvanicky oddeľujúci optoizolátor, ktorý taktiež zabraňuje prítomnosti zemných smyčiek. [1, 2]

Prístroje vybavené MIDI zvyčajne obsahujú prijímač a vysielač. Softwarové vybavenie je vytvorené dohodnutým komunikačným rozhraním. Toto vybavenie definuje, ako majú byť abstraktné MIDI dáta interpretované a aplikované na konkrétne možnosti prístroja. [1]

Komunikácia prebieha viacbajtovými správami. Začiatok správy je tvorený stavovým bytom. Po stavovom byte nasleduje jeden alebo dva dátavé bajty (až na výnimku dát reálneho času a zvláštnych systémových dát). Stavový bajt môže obsahovať informáciu o čísle kanálu (1-16). Užívateľ má možnosť nastaviť adresy zariadení na tieto kanály, pričom tieto zariadenia budú reagovať iba na správy prichádzajúce na príslušný kanál. [1, 2]

### 1.2.1 MIDI správy

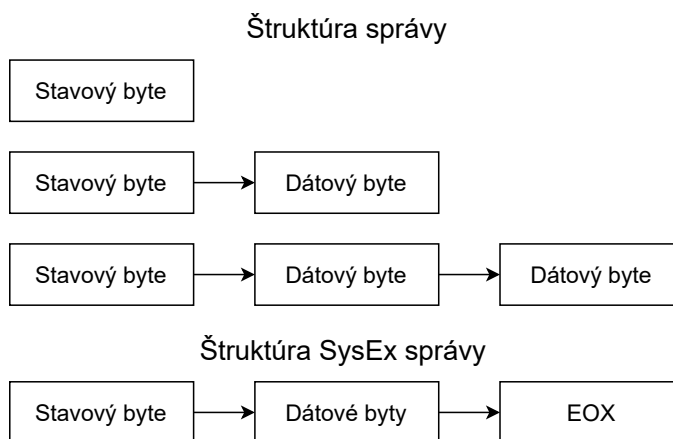
MIDI správy sa delia do dvoch skupín, na kanálové správy a na systémové správy. Do skupiny kanálových správ patrí napríklad informácia o zapnutí a vypnutí nôt, o ich rýchlosti stlačenia, vypnutia a o tlaku ich stlačenia. Tlaková citlivosť klaviatúry je vysielaná buď to individuálne, alebo spoločne pre celú klaviatúru, kde sa posiela



Obr. 1.1: Rozdelenie MIDI správ[2]

najvyššia hodnota. Do tejto skupiny patria aj dáta o polohe kontrolérov (potenciometrov, prepínačov), o voľbe programu, režime nástroju a odpojenia od hlavnej jednotky (zariadenie preruší spojenie medzi klaviatúrou a jeho hlasovými jednotkami). [1, 2]

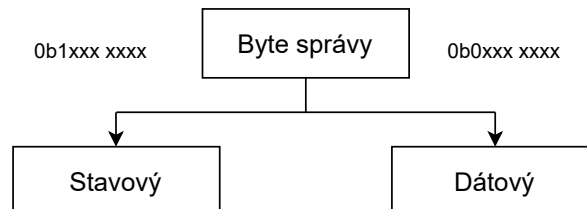
Ďalšie informácie sa vysielajú bez priradenia ku konkrétnemu kanálu, jedná sa o správy systémové. Sú to dáta reálneho času, povely, časovacie hodiny, ukázateľ pozície skladby a aktívne vnímanie (po odpojení káblu by zariadenie malo vypnúť svoje hlasy). Táto kategória obsahuje aj spoločné systémové správy, ako napríklad voľba skladby, žiadosť o naladenie, reset a zvláštne systémové dáta. Zvláštne systémové dáta sú štandardizované len čiastočne a patria sem napríklad informácie o zariadení. [1, 2]



Obr. 1.2: Štruktúra MIDI správ[2]

MIDI správa je zostavená z jedného stavového bytu a žiadneho až dvoch dátových bytov a prikazuje zariadeniu vykonať konkrétnu činnosť. Dátové byty majú na

MSB (Most Significant Bit - bit s najvyššou mocninou) vždy hodnotu 0, vyjadrujú teda hodnoty od 0-127 (0x00-0x7F). Naopak, stavové byty majú vždy na tomto mieste log. 1 a teda nadobúdajú hodnôt od 128-255 (0x80-0xFF). Počet databytov je presne definovaný podľa významu jednotlivých stavových bytov. Existujú tu ale určité výnimky. Napríklad, v režime priebežného stavového bytu stačí vyslať reťazec databytov, pokiaľ nie je potrebné zmeniť stavový byte. [1, 2]



Obr. 1.3: Rozdelenie MIDI bytov[2]

Z hľadiska systematického postupu, je oveľa lepšie rozložiť stavové byty na nibbly. Vzhľadom na to, že najvyšší bit pravého nibblu musí obsahovať 1 (platí pre stavové byty, inak by sa jednalo o dátový byte), pre ďalšie kombinácie ostávajú už len 3 bity (8 hodnôt). Existuje teda len 8 rôznych stavových bytov, ktorých prvý nibble môže mať jednu z hodnôt od 8 (0x8) do 15 (0xF). [1, 2]

Prvých sedem je určených pre kanálové dáta, kde druhý nibble určuje číslo kanálu (preto je možné komunikovať len na 16 kanáloch). Stavový byte začínajúci hodnotou 15 (0xF) je priradený pre systémové dáta. Druhý nibble ale v tomto prípade nevyjadruje číslo kanálu, ale rôzny typ správ. Do tejto skupiny správ patria zvláštne systémové dáta, spoločné systémové dáta a synchronizačné dáta reálneho času. [1, 2]

Dátové byty majú na mieste MSB vždycky hodnotu 0, tým pádom je jeden bit blokovaný a ostáva už len 128 kombinácií. Vzhľadom na to, že sa databyty používajú pre očíslovanie nôt, kontrolérov a programov, je toto číslo dostačujúce. [1, 2]

Pre ľahšiu orientáciu prístrojov v dátach, bola každá nota a prvok komunikácie označený číslom. Rytmus sa v systéme prenáša pomocou správy o vypnutej a zapnutej note, vysielanej v žiadanom časovom odstupe. [1, 2]

V MIDI systéme existuje priorita dát. Najvyššiu prioritu má resetovanie systému, potom povely reálneho času, zvláštne systémové dáta a až nakoniec kanálové dáta. [1, 2]

Ako bolo spomenuté, MIDI umožňuje riadiť len 16 nástrojov súčasne (alebo aj menej, ak jedno zariadenie prijíma na viacerých kanáloch). Táto nevýhoda sa rieši viacerými MIDI okruhmi, alebo nastavením viacerých nástrojov na jeden kanál (viacero nástroj hrá rovnakú notu). [1, 2]

Možnosti MIDI nástroja by mali byť popísané v jeho návode, a to pre príjem a aj pre vysielanie. Samotný nástroj v MIDI systéme môže vykonávať len povely, ktorým rozumie. [1, 2]

MIDI normou nieje definované, čo všetko by malo byť implementované v zariadení a ani neudáva spôsob ovládania. Určuje len relatívne vzťahy, oblasti dát a minimálne nutné vzťahy. Konkrétnym priradením týchto dát ku skutočným funkciám a možnostiam nástroja je na výrobcovi a preto vznikol aj štandard MIDI - General MIDI. [1, 2]

### Kanálové správy

Kanálové správy tvoria väčšinu prenášaných informácií medzi MIDI zariadeniami. Nesú informácie o stlačení a vypnutí noty, zmene programu, individuálnej a spoločnej tlakovej citlivosti, zmene kontroléru a ohýbaní tónu. [2]

Správa Nota zapnutá pozostáva z 3 bytov, čo na prenos vyžaduje 960 us. Vyslanie niekoľkých takýchto správ môže trvať aj niekoľko ms, čo má za následok predĺženie odozvy. Jedným z riešení je Running status. [2]

Hlasové správy nie sú výlučne určené pre klávesové nástroje a môžu byť prenášané na rôzne účely. Napríklad správy, ako Nota zapnutá, generované bežnou klávesnicou, možno použiť na spustenie bicieho nástroja alebo ovládača osvetlenia. [2]

<b>Kanálové správy</b>			
<i>Funckia</i>	<i>Stavový byte</i>	<i>1. Databyte</i>	<i>2. Databyte</i>
Nota vypnutá	0x8k	0-127 číslo noty	0-127 rýchlosť vypnutia
Nota zapnutá	0x9k	0-127 číslo noty	0 vypnutie, 1-127 rýchlosť zapnutia
Individuálna tlaková citlivosť	0xAk	0-127 číslo noty	0-127 hodnota
Kontrolér	0xBk	0-127 číslo kontroleru	0-127 hodnota
Voľba programu	0xCk	0-127 číslo programu	—
Spoločná tlaková citlivosť	0xDk	0-127 hodnota	—
Ohýbanie tónu	0xEk	0-127 hodnota LSB	0-127 hodnota MSB

Tab. 1.1: Kanálové správy (k predstavuje číslo kanálu 0-15)[1]

Stavový byte  $0x8k$  znamená Nota vypnutá a  $0x9k$  Nota zapnutá, kde  $k$  určuje MIDI kanál. Obe tieto informácie sú nutné, pretože priamo súvisia s kódovaním rytmu. Tónový generátor v nástroji, zvaný aj hlas, je spúšťaný príkazom Nota zapnutá, a v tomto stave zotrúva tak dlho, až pokým neprijme príkaz Nota vypnutá. Z toho vyplýva, že všetky príkazy Nota zapnutá by mali byť vyvážené príkazom Nota vypnutá. [1, 2]

Avšak to, že je nota zapnutá nemusí znamenať, že zvuk znie po celú dobu zapnutia. Nota môže prestať znieť skorej, ako bol prijatý príkaz na vypnutie, a po túto dobu ostáva blokovaná (máme o jeden hlas menej). Niektoré prístroje môžu príkaz Nota vypnutá odignorovať úplne, s tým, že sa po odoznení generátor vypína automaticky. [1]

Druhý byte správy, teda prvý databyte týchto povelov definuje číslo noty. Číslo tónu aktivuje priradený zvuk, pričom zariadenia môžu mať tieto abstraktné hodnoty ľubovoľne priradené k jednotlivým tónom. [1, 2]

Tretí byte prenáša rýchlosť stlačenia a pustenja noty. Táto informácia hovorí o hlasitosti tónu alebo slúži aj na ladenie parametrov zvuku. Midi norma doporučuje používať logartimickú mierku, aj keď výrobcovia túto rýchlostnú krivku väčšinou nedodržiavajú. Aby sa vystačilo len s jedným stavovým bytom pre Notu vypnutú aj zapnutú, prišlo sa na to, že stačí vyslať príkaz Nota zapnutá s rýchlosťou 0. [1, 2]

Individuálna tlaková citlivosť má podobu  $0xAk$  a je nasledovaná dvomi databytmi. Prvý z nich udáva číslo noty, druhý hodnotu tlaku. Najvhodnejšie praktické nasadenie tlakovej citlivosti je pri riadení hĺby frekvenčnej modulácie. Každý tón akordu môže byť behom znenia modulovaný. Avšak je treba dodať, že túto funkciu má implementovaných veľmi málo prístrojov a klaviatúr. Hlavným dôvodom je nedostatočná prenosová kapacita zbernice. [1, 2]

Stavový byte s hodnotou  $0xBk$  hovorí o zmene ovládania (Control change), za ktorým nasledujú dva databyty. Prvý hovorí o čísle kontroléru a druhý udáva jeho okamžitú hodnotu nastavenia. Tieto nastavenia platia tak dlho, pokiaľ nepríde nová informácia o nastavení (bez ohľadu naprepnutie MIDI kanálu a odpojenie káblu). [1] Samotné kontroléry nie sú určené normou a táto implementácia zostáva na výrobcovi. Avšak spodný rozsah je určený pre kontinuálne kontroléry, nasledovaný malým rozsahom pre spínače a prvky riadiace hĺbku modulácie. Rozsah 120 až 123 je určený pre špeciálne riadiace povelky (resetovanie, vypnutie všetkých zvukov, tónov a lokálne ovládanie). Za týmito povelmi vždy nasleduje databyte s hodnotou 0. Posledné hodnoty z rozsahu sú vyhradené pre prepínanie MIDI režimu (Omni, Poly a Mono). [1, 2]

Hodnota  $0xCk$  je charakteristická pre voľbu programu a hovorí o zmene znení zvuku. Voľba programu je nasledovaná jedným databytom, je teda možné prepínať medzi 128 programmi, ktoré norma nedefinuje. [1, 2]

<b>Spoločné systémové správy</b>			
<i>Funkcia</i>	<i>Stavový byte</i>	<i>1. Databyte</i>	<i>2. Databyte</i>
Voľba skladby	0xF3	0-127 číslo skladby	—
Žiadosť o naladenie	0xF6	—	—
Resetovanie systému	0xFF	—	—

Tab. 1.2: Spoločné systémové správy[1]

Stavový byte s hodnotou 0xDk je priradený spoločnej tlakovej citlivosti a je nasledovaný jedným databytom, ktorý nesie informáciu udávajúcu tlak. Je teda jedno ktorú klávesu budeme stláčať, vždy bude vysielaná najvyššia tlaková hodnota zo senzorov. [1, 2]

Poslednou správou zo skupiny kanálových správ je Pitch bend. Pitch bend, alebo aj ohýbanie tónu, reprezentuje stavový byte s hodnotou 0xEk, spolu s dvomi dátovými bytmi. Tieto dva byty dávajú dokopy 14 bitov čo tvorí 16384 kombinácií. V praxi sa dnes používa len 7 vrchých bitov. Ak je tento rozsah nastavovaný napríklad potenciometrom, dostávame rozlíšenie 16 krokov na jeden poltón. [1, 2]

## Systémové správy

Jedná sa o dáta, bez zakódovaného kanálu, týkajúce sa všetkých nástrojov v systéme. Medzi týmito dátami sa vyskytujú povely s rôznym a nepravidelným formátom. [1]

Je dobré podotknúť, že niektoré byty (0xF4, 0xF5 a 0xF9) niesú normou definované. Takýto nedefinovaný byte nesmie byť v žiadnom prípade používaný a vysielaný. Pretože sa ale takýto byte môže vyskytnúť (napríklad pri zapínaní prístroja), musí byť ignorovaný. [1]

Do skupiny systémových dát patria spoločné systémové dáta, dáta reálneho času a zvláštne systémové dáta. [1]

Medzi spoločné systémové dáta sa radí voľba skladby, žiadosť o naladenie systému a resetovanie. Voľba skladby bola zamýšľaná pre komunikáciu medzi dvomi sekvencermi alebo rytmermi. Tento povel vyžaduje mať uloženú tú istú skladbu pod rovnakým číslom naprieč prístrojmi. [1]

Žiadosť o naladenie bola často využívaná prístrojmi prvej generácie MIDI, z ktorých mnohé obsahovali napäťovo nastaviteľné oscilátory. Tieto obvody bolo potrebné ladiť, kvôli ich teplotnému driftu a zmenám spôsobených nestabilitou týchto analógových obvodov. [1]

Resetovanie nastavuje prístroj do predvoleného stavu, rovnakého, ako nastáva tesne po jeho zapnutí. V podstate sa jedná o soft-reset, a mal by byť používaný len

<b>Systémové správy reálneho času</b>			
<i>Funckia</i>	<i>Stavový byte</i>	<i>1. Databyte</i>	<i>2. Databyte</i>
MTC štvrtokienko	0xF1	0-119 typ a hodnota	—
Ukázateľ pozície skladby	0xF2	0-127 hodnota LSB	0-127 hodnota MSB
Časovacie hodiny	0xF8	—	—
Štart	0xFA	—	—
Pokračuj	0xFB	—	—
Stop	0xFC	—	—
Aktívne vnímanie	0xFE	—	—

Tab. 1.3: Správy reálneho času[1]

zriedka (napríklad pri zrušení systému, kedy ho nechceme kompletne vypínať). [1]

MTC štvrtokienko, ukázateľ pozície skladby, časovacie hodiny, povely štart, pokračuj, stop a aktívne vnímanie tvoria podskupinu dát reálneho času. MTC štvrtokienko predstavuje systémové hodiny a je odosielané v pravidelných periodických intervaloch. [1]

Ukázateľ pozície skladby udáva hodnotu od začiatku skladby. Je nasledovaný dvomi databytmi, ktoré skladajú maximálnu hodnotu na 16384. Hlavným účelom je, aby sa podriadené zariadenia orientovaly v skladbe a hrali vždy od rovnakého miesta, ako riadiaci prístroj. [1]

Aktívne vnímanie je reprezentované hodnotou stavového bytu FE. Použitie tohto povelu je nezáväzná, a to pre prijímač a vysielač. Tento stavový byte sa vysiela najpozdšie každých 300 ms, v prípade že sa na MIDI zbernici nevyskytuje žiadna aktivita. [1, 2]

Pokiaľ prístroj neobdrží daný bajt, bude pracovať normálne. V prípade príjmu bytu aktívneho vnímania, začne zariadenie očakávať tento stavový byte aspoň raz za 300 ms alebo akúkoľvek aktivitu na zbernici. Pokiaľ v tejto dobe nepríde žiadna informácia, prijímač vypne svoje hlasy a vráti sa do normálneho stavu. Jedná sa teda o bezpečnostný povel, ktorý zariadenie udržuje v bdelom stave a zabraňuje výskyt vysiaticich tónov v prípade odpojenia MIDI káblu. [1, 2]

Poslednou podskupinou systémových dát sú zvláštne systémové dáta. Táto podskupina dnes tvorí najrozsiahlejšiu oblasť MIDI sveta. [1] Zvláštne systémové dáta začínajú stavovým bytom 0xF0, za ktorým nasleduje identifikačný kód firmy (jedno až trojbitový), za ním ľubovoľný počet databytov. Koniec dát je značený EOX (End OF Exclusive). Všetky tieto dátové byty musia mať v prvom bite hodnotu 0. [1, 2]

Pretože SysEx sú väčšinou firemne a prístrojovo špecifické, obsahujú kód vý-



<b>Zvláštnne systémové správy (SysEx)</b>			
<i>Funckia</i>	<i>Stavový byte</i>	<i>1. Databyte</i>	<i>2. Databyte</i>
Začiatok zvláštnych systémových dát	0xF0	ii identifikačný kód výrobca	0-127 ľubovoľný počet databytov
Koniec zvláštnych systémových dát (EOX)	0xF7	—	—

Tab. 1.4: Zvláštnne systémové správy[1]

robcu, a po ňom často-krát nasleduje aj ID konkrétneho prístroju. Pokiaľ prijímač nerozumie ID, mal by dáta ignorovať. [1]

### Running status

V prípade prijatia a spracovania statusového bytu (platí len pre hlasové správy a správy módov), napríklad Nota zapnutá, zariadenie zostáva v tomto stave. Tento stav môže byť zmenený príjmom iného stavového bytu. V situácii, keď sa statusový bajt opakuje, môže byť úplne vynechaný, čo redukuje vyťaženosť zbernice a počet posielených dát. V tomto prípade správa pozostáva len z dátových bytov. [2]

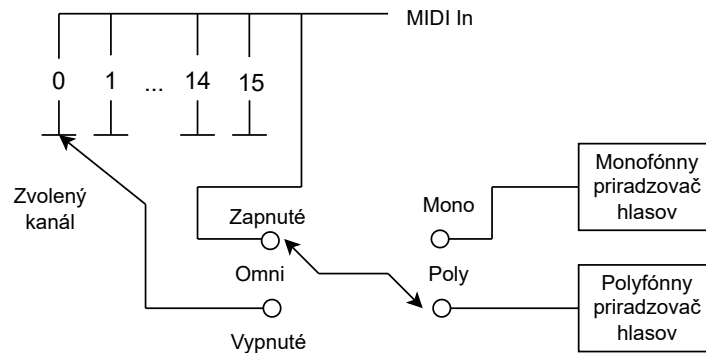
Táto funkcia je obzvlášť užitočná, keď sa napríklad posielajú dlhá postupnosť Nota zapnutá/vypnutá, kde sa pre vypnutú notu posielajú príkazy zapnutej noty s rýchosťou zapnutia 0 (reprezentácia vypnutej noty). V prípade že prijímač nepodporuje konkrétny stavový byte, prijatý príkaz by mal byť ignorovaný. [2]

### 1.2.2 MIDI režimi

Polyfónne zariadenia obsahujú jednotky, ktoré generujú zvuky, zvané hlasy. Hlas je zvuková jednotka alebo generátor zvuku v syntetizátore (alebo inom elektronickom hudobnom nástroji) generujúci tón. Každý nástroj by mal mať definovaný vzťah medzi 16 dostupnými kanálmi a priradením hlasov.[2]

Na základe tohto bolo určených niekoľko režimov zariadenia. Režim OMNI, ktorý môže byť vypnutý alebo zapnutý a režimy POLY/MONO, ktoré sa navzájom vylučujú. Ak je režim OMNI zapnutý, prírmač reaguje na dáta všetkých kanálov. Vo vypnutom móde zariadenie reaguje len na dáta na zvolenom kanály. V režime MONO, je priradený len jeden hlas zariadenia na kanál. Režim POLY priraduje viac hlasov na jeden kánal. [2] Z toho vyplýva, že sú celkovo možné 4 režimi nástroja pre prijímač a vysieláč:

- Režim 1 - Omni zapnuté, Poly - Hlasové dáta sú prijímané na všetkých dostupných kanáloch a priradené viachlasne k hlasom nástroja. Pre vysieláč platí, že



Obr. 1.4: Režimi MIDI zariadenia[2]

všetky dáta sú vysielané na zvolenom kanáli. Niektoré nástroje sa nastavujú do tohoto stavu po reštarte a spustení (hlavne staršie zariadenia). [1, 2]

- Režim 2 - Omni zapnuté, Mono - Dáta sú prijímané na všetkých kanáloch, s tým, že je riadený len jeden hlas, zariadenie teda pracuje jednohlasne. Priorita sa kladie na poslednú stlačenú klávesu. Vysielač vysielá všetky dáta na jeden kanál. Tento režim nieje veľmi používaný. [1, 2]
- Režim 3 - Omni vypnuté, Poly - Hlasové dáta sa prijímajú len na jednom zvolenom kanály a sú viachlasne priradené ku kanálom nástroja. Nápodobne, pre vysielač platí, že všetky dáta sú spoločne vysielané na nastavenom kanály. Tento režim je implicitne nastavený po zapnutí prístroja (novšie zariadenia). [1, 2]
- Režim 4 - Omni vypnuté, Mono - V tomto režime sa hlasové dáta prijímajú na rozsahu zvolených kanálov ( $k+m-1$ ) a sú jednohlasne priradené ku hlasom prístroja 1 až "m". Počet hlasov "m" je určený tretím bytom povelu režimu MONO. Toto číslo je v rozmedzí 1-16 a nepočíta sa s tým, že jeden nástroj bude mať viac než 16 kanálov. Ak je  $m = 0$ , jedná sa o špeciálny prípad, prikazujúci priradiť všetky hlasy, jeden na kanál, od základného kanálu "k". Zmysel tohto stavu bol hlavne v tom, že bolo možné nastaviť individuálnym kanálom rôzne zvukové farby. Tento postup platí obdobne pre vysielač, hlasové dáta sú vysielané na kanáloch  $k+m-1$ , s tým, že jeden hlas je priradený na jeden kanál. [1, 2]

MIDI prímáč alebo vysielač môže pracovať len v jednom stave v čase, obyčajne je vysielač aj prijímač nastavený na rovnaký režim. Nie je podmienkou, aby mal prijímač implementované všetky módy. V prípade požiadavky na zmenu nepodporovaného módu, by mal prijímač správu buď to odignorovať, alebo sa prepnúť do alternatívneho módu (Omni On, Poly). [2]

Norma odporúča, aby bol prístroj po zapnutí nastavený do Poly režimu so zapnutým Omni a nastaveným kanálom na adresu 1. Avšak toto nastavenie sa len odporúča, jeho implementácia závisí od výrobcu. [1, 2]

### 1.2.3 MIDI hardware

MIDI interface komunikuje rýchlosťou  $31,25 \pm 1\%$  kBaudov, asynchrónne, spolu so štartbitom, ôsmimi databitmi a s stopbitom, ako už bolo uvedené vyššie. Odoslanie týchto 10 bitov trvá 320 us, pre jeden sériový byte. Toto zapojenie funguje na princípe prúdovej smyčky (aspoň 5 mA), kde logická nula znamená zapnutý prúd. Dôležité je poznamenať, že jeden výstup môže riadiť len jeden vstup. [2]

Prijímač by mal byť oddelý optickou väzbou, a mal by vyžadovať menej než 5 mA k zapnutiu. Norma doporučuje použitie optoizolátorov PC-900 alebo HP-6N138, ktoré sú špecifické svojou vysokou rýchlosťou. Požiadavkam by mali vyhovovať aj optoizolátory, s časom reakcie menším ako 2 us. [2]

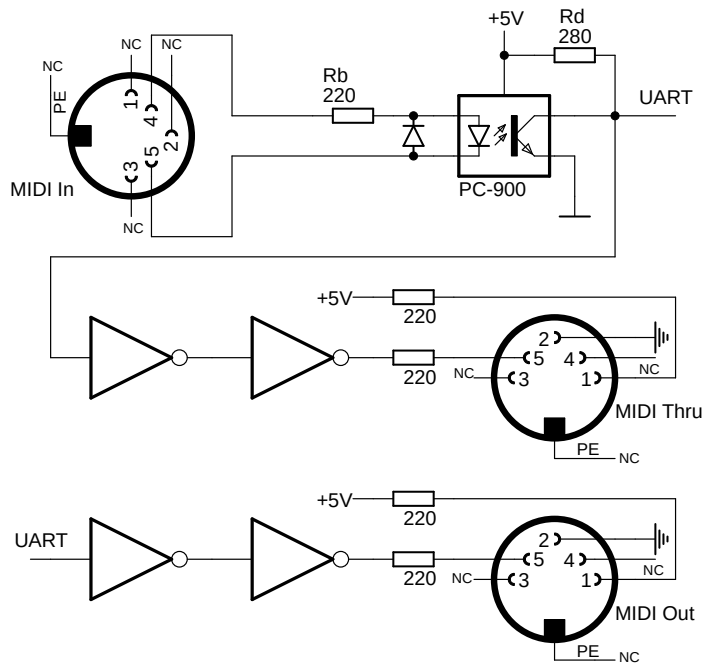
Pre konektory norma vymedzuje použitie DIN 5 panelového konektoru (uhol  $90^\circ$ ). Konektory by mali byť označené ako MIDI In alebo Out. Kontakty číslo 1 a 3 by mali ostať nepripojené ako na strane vysielajúcu, tak aj na strane prijímaču. [2]

Do zariadenie môže byť pridaný aj konektor MIDI Thru, na ktorom je priama bufferovaná kópia dát z MIDI In. Tento konektor slúži na zrefazovanie zariadení. Pri prepojení 3 a viac prístrojov by sa mali použiť optoizolátory s vyššou rýchlosťou, aby sa zabránilo vzniku časových chýb, ktoré môžu ovplyvniť šírku pracovného cyklu. [2]

Prepojovacie káble by mali mať maximálnu dĺžku 15 m a mali by byť zakončené na každej strane päť-kontaktným konektorom typu DIN. Kábel by mal mať dvojité tienenie, ktoré sa pripája na kolík číslo 2 na oboch stranách. Tienenie káblu a zástrčky by sa nemalo prepájať medzi sebou. [2]

Zo schémy zapojenia je možné vyčítať, že systém je napájaný 5 V (TTL logika). Dióda má funkciu ochrany pri prepólovaní a  $R_b$  obmedzuje vstupný prúd. Výstupné obvody optoizolátoru tvorí tranzistor, v tomto zapojení môžeme vraviť o zapojení so spoločným kolektorom. K tomuto zapojeniu sa pridáva pull-up rezistor, ktorý drží kludový stav v logickej jednotke. Norma doporučuje  $280 \Omega$  pre rezistor  $R_d$  (v prípade použitia PC-900), ale táto hodnota sa môže meniť v závislosti použitého optočlenu. [2]

Obvody pre MIDI Out a Thru sú identické, s tým rozdielom, že Thru sa pripája na výstup optoizolátoru (MIDI In) a Out na UART rozhranie riadiaceho členu. Rezistory  $R_r$  a  $R_f$  limitujú tečúci prúd bufferom (napríklad pri skrate). Norma doporučuje použiť výstupný buffer z logických hradiel alebo tranzistorov. Avšak, výstup MIDI Out je možné riadiť priamo kontrolérom, ak jeho pin znesie prúd aspoň 5 mA.



Obr. 1.5: Zapojenie MIDI rozhrania[2]

Naopak, MIDI Thru by sa malo vždy pripájať cez zosilňovač, aby nedochádzalo k degradácii signálu z MIDI In. [2]

### 1.2.4 Dodatok normy k 3.3V systémom

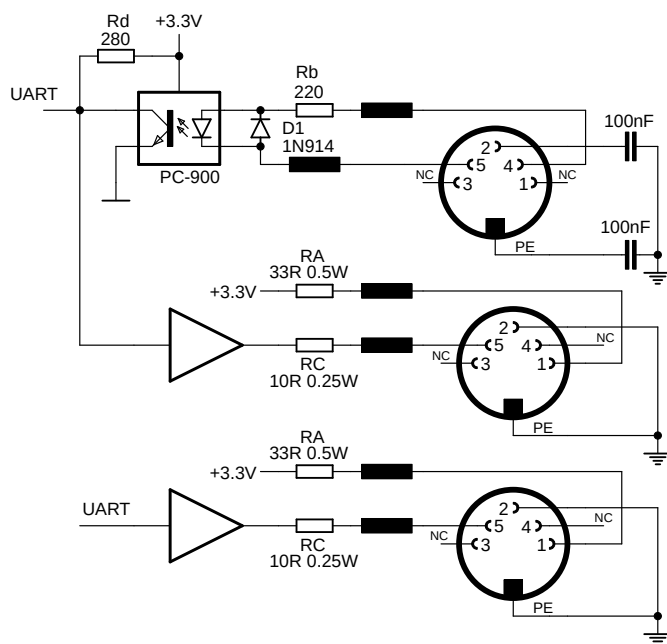
V čase založenia MIDI normy, bola 5V TTL logika priemyselným štandardom (TTL), preto aj MIDI vysielač vyžaduje 5V napájanie. Avšak, dnes sa štandardne používajú 3.3V systémy, preto tu existuje niekoľko dôvodov (hlavne redukcia výrobných nákladov) na prispôbenie tohto systému na 3.3V logiku. [3]

Štandardné zapojenie MIDI tvorí prúdová slučka ktorá ovláda opto-izolátor nachádzajúci sa v prijímači. Z pohľadu vysielača, tento optoizolátor tvorí sériové zapojenie LED spolu s  $220\Omega$  rezistorom. Optoizolátory, s ktorými sa môžeme stretnúť v MIDI prístrojoch, majú obvykle 1.9V ubýtok v priepustnom smere a potrebujú minimálne 5 mA pre správne fungovanie. Aby bolo možné do tohoto obvodu dodávať 5 mA, musí byť splnená nasledujúca podmienka:

$$V_{TX} \geq V_{RXDROP} = 0.005 * 220 + 1.9 = 3.0V \quad (1.1)$$

[3]

Ďalej je vyžadované prúdové obmedzenie MIDI výstupu. Toto obmedzenie je docielené pripojením sériových rezistorov k výstupným pinom. Aby obvodom mohlo



Obr. 1.6: Zapojenie MIDI rozhrania pre 3.3V systémy[3]

tieč 5 mA, výstupné rezistory by mali spĺňať túto podmienku:

$$(R_A + R_C) \leq \frac{V_{TX} - V_{RXDROP}}{0.005} \quad (1.2)$$

[3] Je doporučené použiť hodnoty  $33 \Omega$  a  $10 \Omega$  pre rezistory  $R_A$  a  $R_C$ , spolu  $43 \Omega$ . Za predpokladu zohľadnenia tolerancie rezistorov a napájacieho napätia, môže v najhoršom prípade obodom tiecť len 4.47 mA. Ukázalo sa, že tento prúd je dostatočný, pre vytvorenie 1.9V úbytku na LED optoizolátoru. [3]

Rezistor  $R_A$  je priamo pripojený na napájaciu vetvu, a preto by skrat mohol spôsobiť výkonovú strátu (+5% pre  $U_{RX}$  a -5% pre  $R_A$ ) podľa nasledujúceho vzťahu:

$$P = \frac{U_{RXMAX}^2}{R_A} = \frac{3.465^2}{31.35} = 0.384W \quad (1.3)$$

Pri skrute by pretekajúci prúd vytvoril na  $R_A$  výkonovú strátu 0.38 W, a preto je odporúčené použiť 0.5W rezistor. Pre ochranu digitálneho buffera, v prípade skratu prepojovacieho kábla, by mal byť použitý 0.25W rezistor. [3]

Pôvodná špecifikácia vyžaduje, aby bol pin č.2 MIDI In odpojený od signálovej zeme. Tento prístup zabraňuje zemným slučkám, ale je skôr vhodný pre audio signál, ake pre redukcii vysokofrekvenčného rušenia. Jedným z riešení je pripojiť tento pin MIDI In konektu na zem zariadenia prostredníctvom kondenzátoru s malou kapacitou. Toto riešenie zachová oddelenie zemnej slučky na audiofrekvenciách, zatiaľ čo tvorí nízkoimpedanšnú cestu pre vysoké frekvencie. [3]

Ďalej, pre splnenie noriem elektromagnetického rušenia a elektromagnetickej kompatibility sa môže vyžadovať RF filtrovanie signálových pinov MIDI konektorov. Tento dodatok odporúča pridanie feritových jadier ako RF filtrov pre signálové piny. Tieto ferity by mali byť umiestnené čo najbližšie ku konektorom, avšak nemusia byť použité, ak zariadenie nevyžaduje EMI/EMC certifikácie. Taktiež odporúča pripojiť tienenie konektoru k zemi prístroju. [3]

Tento nízko-napäťový obvod je kompatibilný so všetkými MIDI prijímačmi, prísne dodržiavajúcimi štandard MIDI 1.0. Produkty, ktoré sa odchyľujú od špecifikácie normy alebo sa ju pokúšajú rozšíriť, nemusia byť kompatibilné. Jedná sa o zariadenia ktoré buď to nepoužívajú oddelovací optoizolátor, alebo sú napájané priamo z pinu číslo 4 MIDI In vstupu. [3]

### 1.2.5 Prepájanie MIDI systémov

Keďže existuje množstvo kombinácií, ako prepájať MIDI zariadenia, budú popísané len základné spôsoby prepojenia. Aj v tejto oblasti existujú isté pravidlá, ktoré je vhodné dodržiavať. Hoci by nemalo dochádzať k poškodeniu prepájaním zariadení v zapnutom stave, zariadenia sa doporučuje prepájať vypnuté. [1]

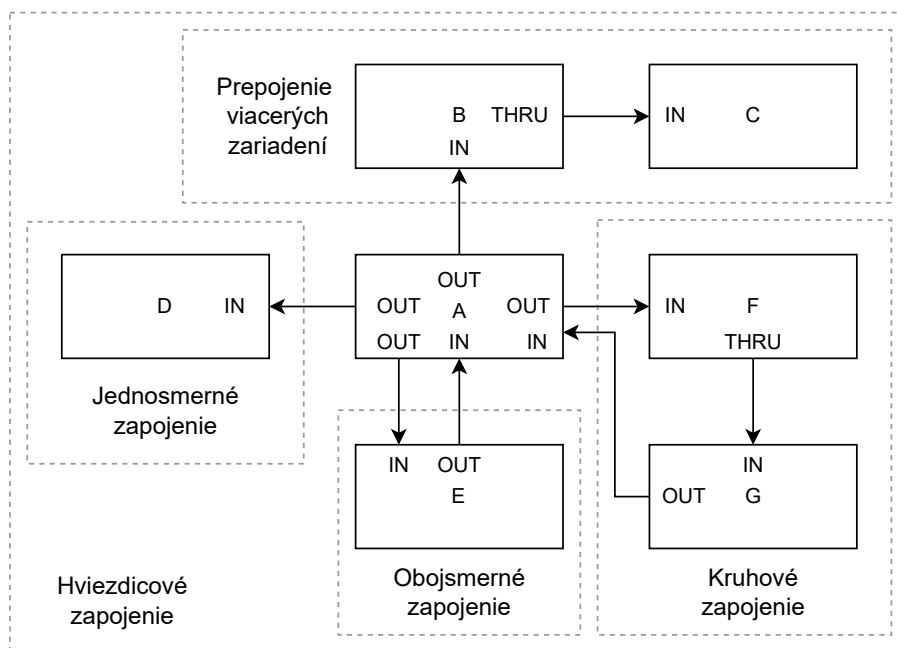
Ako prvé by sa mali zapínať riadené zariadenia (v prípade reťazového zapojenia by sa malo postupovať smerom od najvzdialenejšieho zariadenia) a až následne môžu byť zapnuté riadiace prístroje. Dôvodom pre to je, že obvykle po zapnutí riadiaci prístroj vysiela resetovací povel alebo konfigurácie pre nastavenia kontrolerov MIDI režimu. [1] MIDI nástroje môžu byť vybavené týmito DIN konektormi:

- MIDI In - vstup dát
- MIDI Out - výstup dát
- MIDI Thru - prechod dát

Prístroje bežne disponujú minimálne konektorom MIDI In a v prípade, že sú schopné vysielať, aj MIDI Out. Naopak, konektor MIDI Thru nie je považovaný za zásadný, a preto ho mnoho prístrojov nezahŕňa. [1, 2]

Niektoré hudobné nástroje, ako napríklad klávesy, majú integrovaný MIDI vysielač. Tento vysielač je priamo prepojený s klaviatúrou a rôznymi ovládacími prvkami. Okrem toho, tieto nástroje majú tiež MIDI prijímač, ktorý je spojený s hlasovými jednotkami na generovanie zvukov. Vysielač samozrejme posiela informácie aj do vlastných hlasových jednotiek prístroja. Tieto dve sekcie sú obvykle prepojené medzi sebou, aj keď pracujú nezávisle na sebe a na väčšine prístrojov sa dá toto spojenie zrušiť (Local Control). [1]

Nástroje je možné prepájať jednosmerne, obojsmerne, do kruhu a hviezdice, a je ich tiež možné reťaziť za sebou. V prvom prípade sa jedná o najjednoduchšie zapojenie medzi dvomi prístrojmi. Prepája sa MIDI Out nadriadeného zariadenia s MIDI In



Obr. 1.7: Prepojenie MIDI zariadení[2]

na riadenom zariadení. [1]

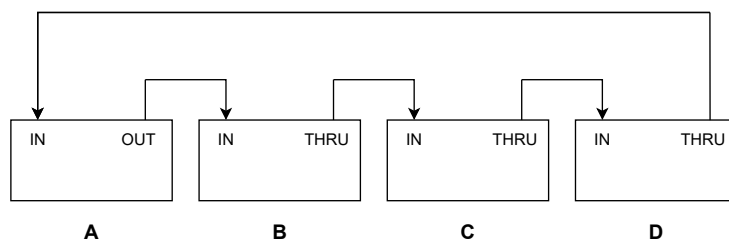
Podriadený prístroj by mal mať vhodne nastavený kanál na ktorom prijíma a zároveň rovnaký MIDI režim. Nástroj môže byť aj súčasne naďalej riadený z svojej klaviatúry, potom ale dochádza k zdvojovaniu tónov. V tomto prípade sa môže stať, že niektoré tóny zostanú zniet, dôsledkom porušenia následnosti informácií (nota zapnutá/vypnutá). [1]

Druhý prípad hovorí o obojsmernom zapojení, kde sa zariadenia prepájajú dvomi káblami. Takéto obojsmerné prepojenie je nutné v prípade vymieňania SysEx dát pomocou komunikácie s potvrdovaním. Ak bude takto prepojených viacero zariadení, bude sa jednať o kruhové zapojenie. [1]

Posledný prípad, hviezdicové zapojenie, je tvorené kombináciou predchádzajúcich zapojení. Pozostáva z jedného riadiaceho člena a niekoľko podriadených zariadení, pričom môžu komunikovať jednosmerne alebo obojsmerne. Toto zapojenie sa používa pre rozsiahle MIDI systémy s hlavným riadiacim členom v podobe počítaču alebo sekvenceru. [1]

Postupným vývojom vznikali v praxi ďalšie typy MIDI konektorov. Niektoré sekvencery kombinujú funkciu Out a Thru. Tieto prístroje kombinujú vlastné dáta s dátami prijatými na MIDI In a posielajú cez výstup nazývaný Mix (alebo Merge). Ďalšie zariadenia môžu byť vybavené konektorom Ext, slúžiacim na priame riadenie (neberie sa ohľad na momentálne nastavené zariadenia). [1, 2]

Pri prepojovaní zariadení je potrebné zabezpečiť, aby nedošlo k vytvoreniu uzav-



Obr. 1.8: Príklad chybného prepojenia zariadení[1]

retej smyčky. Táto smyčka by zapríčinila cirkuláciu dát v systéme. Ďalej MIDI norma zakazuje rozvdojovať MIDI prepojovacie káble, vzhľadom na to, že sa jedná o prúdovú smyčku a dochádzalo by k preťažovaniu (výstupných) hradiel. Miesto toho sa používa hviezdicové zapojenie v kombinácii s Thru Boxom (MIDI rozdelovač). [1, 2]

### 1.2.6 MIDI 2.0 štandard

Po desaťročiach úspechu pôvodného MIDI 1.0, bol predstavený nový MIDI 2.0, ktorý prináša významné inovácie a rozšírenia. Nový štandard predstavuje rozšírenie pôvodného MIDI 1.0 a pôvodný štandard nenahrádza, ale je postavený na jeho základných princípoch, architektúre a sémantike. MIDI 1.0 má určité obmedzenia pre ďalší vývoj, ktoré dneska rieši MIDI 2.0. [5]

Hlavnou inováciou je obojsmerná komunikácia, slúžiaca napríklad pre automatickú konfiguráciu zariadení. Na tejto inovácii stavia hlavná funkcia MIDI CI (MIDI Capability Inquiry), ktorá umožňuje vzájomnú identifikáciu zariadení. Identifikáciou sa myslí, že si zariadenia vymenia informácie o podporovaných funkcionalitách, čím sa výrazne zlepšuje ich vzájomná kompatibilita. [5, 4]

Ďalšou inováciou je univerzálny MIDI packet, ktorý definuje rozšírený dátový formát na 32 bitov. MIDI 2.0 zahŕňa štyri hlavné dokumenty, ktoré rozširujú funkcionality MIDI 1.0 a sú to:

- MIDI Capability Inquiry (MIDI-CI)
- Common Rules for MIDI-CI Profiles
- Common Rules for MIDI-CI Property Exchange
- Universal MIDI Packet (UMP) Format and MIDI 2.0 Protocol

Dokument MIDI CI opisuje mechanizmus, ktorý slúži na identifikáciu prepojených zariadení a ich možností tak, aby mohli efektívne využívať rozšírené funkcie MIDI 2.0. Tento mechanizmus je základom pre automatickú konfiguráciu a zvýšenú interoperabilitu medzi rôznymi zariadeniami. [4]

Profily predstavujú dôležitú súčasť automatickej konfigurácie v rámci MIDI-CI. Profil v MIDI-CI tvorí definovanú sadu MIDI správ a implementačných pravidiel,



ktoré sú určené na dosiahnutie konkrétneho cieľa alebo prispôsobenie sa špecifickým aplikáciám. Okrem definovania reakcií na MIDI správy môže profil voliteľne definovať aj ďalšie funkčné požiadavky na zariadenie. [4]

Common Rules for MIDI-CI Property Exchange popisuje mechanizmy na získavanie a nastavovanie dát vo formáte JSON, prenášaných univerzálnymi SysEx (system exclusive) správami. Universal MIDI Packet (UMP) definuje nový formát údajov pre správy protokolu MIDI 1.0, správy protokolu MIDI 2.0 a základ pre rozšírenie MIDI. [4]

Ako už bolo spomenuté, nejedná sa o náhradu, a tak musí byť zabezpečená spätná kompatibilita. Zariadenia by mali v základe komunikovať podľa MIDI 1.0, avšak pomocou SysEx správ sa môžu dohodnúť a prepnúť na štandard 2.0. [5, 4]

Táto norma sama o sebe nedefinuje žiadne špecifické fyzické rozhranie, avšak najčastejšie je implementovaná prostredníctvom USB. To znamená, že hoci stanovuje štandardy pre prenos a spracovanie MIDI dát, konkrétny spôsob, akým sú tieto dáta fyzicky prenášané od zariadenia k zariadeniu, môže byť rôznorodý. [5, 4]

USB sa vďaka svojej širokej dostupnosti a kompatibilite stalo bežným a preferovaným spôsobom implementácie tejto normy. Pre túto implementáciu existuje špeciálna norma - USB Class Definition for MIDI Devices v2.0, ktorá stanovuje štandardy pre prenos a spracovanie MIDI dát cez USB. To umožňuje širokú kompatibilitu a efektívne využívanie v rôznych zariadeniach. [4, 8]

Použitie USB pre MIDI prináša niekoľko dôležitých výhod. Prvou z nich je rýchlejší prenos dát, čo je kľúčové pre zabezpečenie plynulého a bezproblémového prenosu aj pri zložitých MIDI dátach. Vďaka väčšej kapacite a efektivite USB môžu MIDI zariadenia podporovať vyšší počet kanálov. [5]

Ďalšou výhodou je univerzálna kompatibilita USB. Toto rozhranie je široko podporované v mnohých zariadeniach, čo uľahčuje pripájanie a používanie MIDI zariadení s rozličnými počítačmi a mobilnými zariadeniami bez potreby špeciálnych adaptérov alebo rozhraní. Jednoduchosť konfigurácie a inštalácie je tiež dôležitým faktorom. Väčšina USB MIDI zariadení nevyžaduje inštaláciu špeciálnych ovládačov, čo umožňuje rýchlu a jednoduchú konfiguráciu. [5]

## 2 Prieskum trhu

Vzhľadom na to, že hlavným cieľom tejto práce je výroba zariadenia schopného hrať na hudobných nástrojoch, je potrebné vhodne analyzovať trh zariadení s podobnou funkcionalitou.

Výskyt takýchto zariadení je veľmi zriedkavý a v komerčnom smere takmer žiadny. Dôvodom je obrovská diverzita v hudobných zariadeniach, každé s unikátnymi ovládacími prvkami a špecifickými hudobnými charakteristikami, čo komplikuje vývoj univerzálneho zariadenia, ktoré by dokázalo uspokojiť všetky tieto požiadavky.

Avšak existuje plno nadšencov MIDI technológie, ktorí využívajú hlavne známe a cenovo dostupné mikrokontroléry s podporou USB (napríklad Arduino), čo spolu s jednoduchosťou MIDI USB značne pomohlo rozšíreniu týchto technológií. Sú to napríklad MIDI klávesnice alebo bicie, alebo si tieto nástroje upravujú podľa vlastných potrieb.

Tieto prístroje sú zvyčajne zostavené z rôznych tlačidiel, otočných a posuvných potenciometrov a prepínačov. Vo väčšine prípadov sa jedná o zariadenia, ktoré generujú rôzne MIDI udalosti.

Niektorí odborníci zaujímajúci sa o MIDI, upravujú rôzne hudobné nástroje ako sú klavíry a iné klávesové nástroje, čím rozširujú ich možnosti automatizácie hry a ovládania nástrojov cez rozličné komunikačné zariadenia.

Napriek vzácnosti výskytu vlastnoručne vyrobených MIDI zariadení v bežnom použití, sa podobné prístroje dajú nájsť napríklad v kostoloch. V týchto miestach sa používajú rozsiahle systémy na ovládanie veľkých organov, kde musí byť hrací stôl oddelený od píšťal organu.

Organy patria do skupiny najkomplexnejších hracích nástrojov, skladajúc sa z píšťal, vzduchového zásobníku a hracieho mechanizmu (klaviatúry). Hrací mechanizmus tvorí viacero klávesových klaviatúr (nazývané aj registre - sady píšťal), pedálová klaviatúra a rôzne tlačítka s tiahkami.

Niektoré z týchto prístrojov poskytujú nielen lokálne ovládanie, ale aj možnosť automatizovanej hry alebo diaľkového ovládania prostredníctvom MIDI. Príkladom takéhoto systému je organ v Fourth Presbyterian Church v Chicagu. [7] Tento systém je vytvorený firmou Allen Organ Company, špecializujúcou sa na výrobu automatizovaných organov pre kostoly, nahrávacie štúdiá a divadlá.

Existujú spoločnosti zaoberajúce sa vývojom systémov, ktoré zaznamenávajú pohyby a dynamiku hry skutočných hudobníkov a prekládajú ich na digitálne informácie vhodné pre záznam v *Digital Audio Workstation* (DAW) softvare.

Jedným z príkladov je firma Bechstein Digital. Táto spoločnosť sa zaoberá vytváraním systémov, ktoré transformujú bežné akustické klavíry a piana na ich digi-

tálne verzie. Ich systém detailne zaznamenáva všetky aspekty hry hudobníka, vrátane tlaku na klávesy, rýchlosti stlačenia a uvoľnenia kláves. Toto umožňuje presný a komplexný digitálny záznam, ktorý možno následne upravovať, analyzovať alebo prehrávať pomocou rôznych hudobných softwarov.

MIDI bol pôvodne navrhnutý ako digitálny komunikačný systém pre hudobné nástroje. Tento systém sa rozšíril aj do iných oblastí, hlavne vďaka SysEx (System Exclusive) správam.

Jedným z príkladov, kde sa MIDI uplatňuje mimo tradičnej hudobnej sféry, je oblasť osvetlenia, najmä pri sprevádzaní hudby svetelnou šou. Existujú svetelné systémy a zariadenia, ktoré sú schopné komunikovať cez MIDI rozhranie. MIDI poskytuje jednoduchý a efektívny spôsob, ako koordinovať hudbu a svetelné efekty, čo je obzvlášť užitočné pri koncertoch, divadelných predstaveniach a iných podujatiach.

Vzhľadom na nespochybniteľné výhody USB, sa výrobcovia hudobného vybavenia sústreďujú predovšetkým na implementáciu tohto rozhrania. Napriek tomu, mnohé zariadenia stále poskytujú aj štandardné MIDI rozhranie. USB v súčasnosti predstavuje systém "plug and play", ktorý zaručuje kompatibilitu a vysokú rýchlosť komunikácie s rozličnými operačnými systémami ako Windows, MacOS, Linux, Android a iOS. Príkladom môžu byť klávesové kontroléry, syntetizátory, samplery a elektronické bicie, ktoré sú týmito rozhraniami aj napájané. [8]

Aj napriek vysokej rýchlosti USB, môže niekedy dochádzať k oneskoreniam. V dôsledku toho môže byť v niektorých situáciách uprednostnené štandardné MIDI rozhranie. Často sa môžeme stretnúť s kombináciou USB a MIDI zariadení, kde sú oba typy zariadení prepojené prostredníctvom USB-MIDI hubov a konvertorov. Tento spôsob umožňuje využívať výhody oboch rozhraní - flexibilitu a jednoduchosť USB pri súčasnom zachovaní spoľahlivosti štandardného MIDI. [8]

## 3 Návrh systému

Témou tejto práce je navrhnúť spôsob a technické riešenie pre ovládanie fúkacej hramoniky a metalofónu. Ovládanie týchto hudobných nástrojov zahŕňa použitie elektrických solenoidov na generovanie tónov, pričom každý nástroj má svoj vlastný špecifický spôsob ovládania solenoidov pre generovanie zvukov. Zariadenie by malo byť schopné prijímať riadiace povely prostredníctvom MIDI a USB, vhodne ich spracovať a následne priradiť výstup tak, aby boli zahrané požadované noty.

### 3.1 Hudobné nástroje

#### Metalofón

Prvým z riadených hudobných nástrojov je metalofón. Metalofón je nástroj pozostávajúci z naladených kovových dielov (tyče, rúry alebo taniery), niekedy nazývaných aj kameňov, na ktoré sa udiera buď to kladivkami, ručne alebo klaviatúrou.



Obr. 3.1: Metalofón

Ako je možné vidieť na obrázku, tento metalofón má 25 kovových dosiek (2 oktávy plus jeden tón z vyššej oktávy). Tento systém je ovládaný solenoidmi s pružinou. Jedná sa o 5W príťažný solenoid na 24V od firmy BLP Components. Pri

zapnutí solenoidu dôjde k pritiahnutiu dreveného kladivka. Odpojením sa uvoľní nahromadená energia v pružine úderom kladivka o kovový kameň. Frekvencia znejúceho tónu je daná na základe rozmerov a hmotnosti kovových dosiek. Tieto diely mali byť umiestnené na vhodnom materiály, aby nedochádzalo k rýchlemu tlmeniu vibrácií.

Jednou z úloh bolo vymeniť gumičky, na ktorých tieto kamene ležali. Degradácia materiálu spôsobila, že guma stratila svoje špecifické vlastnosti a jednotlivé tóny znely nenaladeno a rýchlo doznievali. Namiesto tohto dielu bola použitá silikónová trubička, ktorá prístroju vrátila jeho hudobné vlastnosti.

### **Fúkacia harmonika**

Druhým riadeným nástrojom je fúkacia harmonika Hohner Melodica Student 32 (označovaná aj HM-32). Jedná sa o klaviatúru pozostávajúcu z 32 klávesov.

Ako v prípade metalofónu, aj tento prístroj je ovládaný solenoidmi. Tieto solenoidy ovládajú klávesnice mechanickými tiahľami. Oproti metalofónu, znenie tónu začína hneď po aktivácii solenoidu, a s deaktiváciou sa klávesa púšťa.

Fúkacia harmonika patrí do skupiny dychových hudobných nástrojov. Pre správnu funkčnosť fúkacej harmoniky je nevyhnutné zabezpečiť dostatočný prísun vzduchu na jej vstup. Tento prívod vzduchu je kľúčový pre generovanie zvuku, keďže harmonika vytvára zvuk vibráciou vzduchu vo vnútri svojich píšťal alebo jazýčkov. Intenzita znejúcich tónov sa bude riadiť regulátorom vzduchu, čo je aj jedna z požiadaviek, ktorú bude treba zohľadniť pri návrhu systému.

Stláčanie kláves je realizované pomocou elektromagnetických solenoidov. Pre túto funkciu boli vybrané príťažné solenoidy od spoločnosti SAIA-BURGESS. Jedná sa o 7W modely, bez pružiny, schopné vyvinúť silu 12 N.



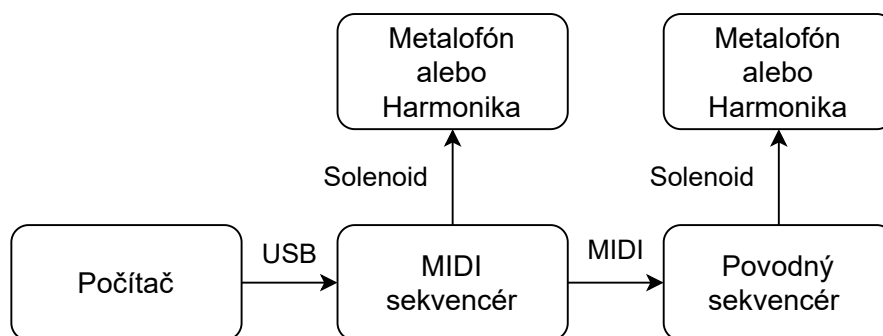
Obr. 3.2: Fúkacia harmonika osadená na prípravku so solenoidmi

## 3.2 Požiadavky pre návrh systému

Cieľom práce je vytvoriť zariadenie, ktoré dokáže ovládať hudobný nástroj pomocou solenoidov a prípadne regulovať tlakový ventil. Systém musí byť schopný nielen spracovávať riadiace informácie, ale aj ich preposielať do iných MIDI prijímačov, ako je napríklad ďalší MIDI sekvencér ovládajúci hudobný nástroj. Zariadenie by malo obsahovať rozhranie pre USB komunikáciu s nadriadeným systémom a schopnosť preposielať dáta cez MIDI rozhranie.

Pre konfiguráciu a ovládanie systému je dôležité pridať ovládacie prvky, ako sú zobrazovací displej a fyzické ovládacie rozhranie. Ovládacie rozhranie by malo zahŕňať prvky ako tlačidlá alebo otočné ovládače, ktoré umožnia používateľom rýchlo a presne nastaviť parametre systému, ako sú výstupné nastavenia solenoidov, regulácia tlaku a výber MIDI kanálov.

Pri navrhovaní PCB je potrebné brať ohľad na spotrebu zariadenia, zvlášť pri zapínaní viacerých solenoidov naraz.



Obr. 3.3: Bloková schéma konceptu systému

### 3.2.1 Ovládanie cez USB

Jednou z požiadaviek je, aby bol systém ovládateľný cez USB rozhranie. V podstate sa ponúkajú dve možné riešenia.

Prvá možnosť zahrňa využitie mikrokontroléru s integrovaným USB rozhraním, ako je napríklad AtMega32U4. Tento prístup využíva štandardizáciu USB zariadení podľa ich tried, čím umožňuje jednoduchšie využitie univerzálnych ovládačov. V prípade MIDI zariadení je možné využiť špecifickú USB MIDI triedu. Aj keď táto metóda môže poskytnúť vyššie prenosové rýchlosti oproti tradičnému MIDI, pre tento typ aplikácie to nieje kľúčový faktor. Vzhľadom na zložitosť implementácie by bolo vhodné zvážiť použitie hotových riešení, ako sú napríklad Arduino Pro alebo Leonardo.

Druhé možné riešenie zahŕňa komunikáciu prostredníctvom sériového komunikačného portu. Táto metóda sa bežne využíva na komunikáciu a niekedy aj na programovanie zariadení. Vyžaduje pridanie hardvérového prevodníka, ktorý vytvorí spojenie medzi USB a sériovou linkou. Toto zariadenie sa potom prezentuje ako komunikačný port s priradeným číslom.

DAW software však počíta so zariadením typu MIDI port. Sériový komunikačný port nemôže byť prepojený ani konvertovaný na tento druh rozhrania, preto je potrebné vytvoriť virtuálny MIDI port v programe loopMidi. Tento softwarový nástroj slúži na vytváranie virtuálnych MIDI portov, pre komunikáciu a smerovanie dát naprieč rôznymi programami.

Ďalším krokom je premostiť tieto dáta z virtuálneho MIDI portu do sériového portu. Na tento účel slúži aplikácia Hairless MIDI to Serial Bridge. Prostredníctvom tejto aplikácie je možné komunikovať medzi sériovým a MIDI zariadením zároveň v oboch smeroch. Je dobré poznamenať, že tento software ponúka len predefinované komunikačné rýchlosti, pričom neumožňuje výber štandardnej rýchlosti 31,25 kBaud, ktorá je pre MIDI typická.

Táto konfigurácia programov spolu s DAW sa ukázala ako funkčná a bola otestovaná na vývojovom kите pre mikroprocesor STM32F407.

## 4 Návrh hardvéru

Návrh hardvéru je stanovený zároveň podľa MIDI normy a aj požiadavkami samotného výrobku, ktoré preddefinujú nutnosť zahrnúť niektoré nevyhnutné komponenty. Dodržanie návrhu vstupných a výstupných obvodov MIDI zabezpečí hardverovú kompatibilitu naprieč rôznymi zariadeniami.

Prijaté informácie z USB a MIDI je potrebné vhodne vyhodnocovať a spínať výstupy, čo zabezpečí mikrokontrolér s dostatočným počtom rozhraní a výstupov. Výstupy by mali byť spínané pomocou vhodného prvku, pretože priame spínanie solenoidov pomocou mikroprocesoru nieje vhodné a pravdepodobne ani funkčné riešenie.

Užívateľ by mal vedieť vhodne interagovať so systémom a do určitej miery si ho pripôsobiť potrebám aplikácie. Na tento účel bol zvolený maticový displej s rozlíšením 128x64 pixelov a ako vstupná periféria rotačný enkodér.

Celý systém bude napájaný 24V zdrojom, kvôli správnej funkcionalite a dynamike solenoidov. Z toho plynie potreba vytvoriť vhodné napájacie vetvy pre logickú časť a pre riadenie tlakového regulátoru. Pri návrhu je dobré brať ohľad na celkovú spotrebu pri súčatnej aktivácii viacerých výstupov, a výrobok vhodne kompenzovať a chrániť .

### 4.1 Výber mikrokontroléru

Medzi hlavné požiadavky vhodného mikrokontroléru patrí dostatočný počet výstupov a aspoň dve komunikačné rozhrania UART. Pôvodne sa uvažovalo použitie mikrokontroléru AtMega2560 od firmy Atmel (dnes už Mikrochip). Tento čip bohate vyhovoval požiadavkam, má dostatočný počet výstupov, komunikačných rozhraní, výpočetný výkon a pamäť. Pôvodný systém bol navrhovaný s predpokladom že sa použije tento typ procesoru.

Kvôli vysokej cene a nedostupnosti bola následne vybratá náhrada. Bol vybratý procesor ST32F407, ktorý je narozdiel od 8-bitovej AtMega2560 postavený na 32-bitovej architektúre.[9]

Jadro je postavené na *Cortex<sup>TM</sup> – M4* architektúre, s maximálnym taktom 168 MHz.[9] Tento mikrokontrolér má k dispozícii rozsiahlu škálu komunikačných rozhraní a časovačov, čo bude veľmi užitočné pre následný dizajn systému.

Okrem toho obsahuje mikrokontrolér aj analógovo-digitálne prevodníky ako ADC a DAC.[9] Tieto prvky budú veľmi výhodné pre ovládanie elektrického tlakového ventilu.

Významnou výhodou mikrokontroléru je tiež veľká kapacita pamäti - 512 KB flash a 192 KB RAM, ktorá spolu s jeho výkonom poskytuje ideálne podmienky



AO3400 parametre		
<i>Parameter</i>	<i>Symbol</i>	<i>Maximálna hodnota</i>
Napätie Drain-Source	$V_{DS}$	30 V
Napätie Gate-Source	$V_{GS}$	$\pm 12$ V
Odpor v zapnutom stave $V_{GS} = 2.5$ V	$R_{DS}$	48 m $\Omega$
Prahové napätie Gate-Source	$V_{GS(th)}$	1.45 V
Trvalý prúd Drainom	$I_D$	4.7 A

Tab. 4.1: Parametre tranzistoru pre spínanie solenoidov[10]

pre použitie operačného systému, ako je napríklad FreeRTOS.[9]

Aby mikrokontrolér fungoval správne, je dôležité vybrať komponenty v súlade s odporúčaniami výrobcu, vrátane filtračných kondenzátorov, zdroja hodín a bootovacích nastavení.

Zdrojom hodín pre systém je 8 MHz kryštál, pričom oscilátor pre hodiny reálneho ostane nepoužitý. Mikrokontrolér poskytuje viacero možností bootovacích nastavení - pre štart z internej flash pamäte musia byť piny BOOT0 a PB2 uzemnené a pridané LED diódy budú signalizovať stav systému. [9]

## 4.2 Ovládanie solenoidov

Pre spínanie solenoidov bolo vybrané zapojenie s unipolárnym tranzistorom vo funkcii spínaču. Oproti bipolárnemu, sa tento druh riadi napätím medzi elektródami Gate a Source. Aby bolo možné tieto tranzistory riadiť priamo mikrokontrolérom, musia mať  $V_{GS(th)}$  nižšie, ako hodnota logickej jednotky na výstupe portu. Jedným z kritérií výberu bol aj parameter popisujúci odpor v zopnutom stave. Nižšia hodnota tohto odporu znamená nižšiu výkonovú strátu.

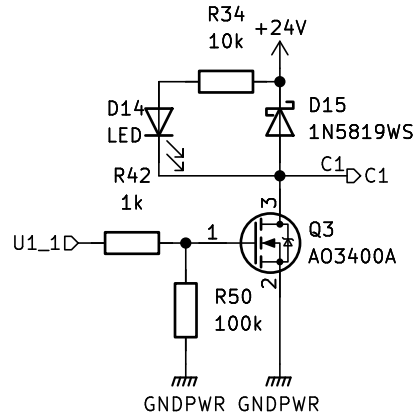
Na túto úlohu bol vybraný tranzistor AO3400A, s vhodným  $V_{GS(th)}$ , malým prechodovým odporom. Aby sa nedošlo k poškodeniu, tento tranzistor má maximálne napätie medzi Drain a Source 30 V.[10]

Tranzistor sa začína otvárať približne pri 1 V, a pri 2.5 V vykazuje odpor 50 m $\Omega$ , čo môžeme brať ako najhorší prípad.[10] Takto nízky odpor, v porovnaní s odporom solenoidu, sa javí ako dokonalý spínač.

Solenoid sa skladá z cievky navinutej okolo feromagnetického jadra, ktoré je pri tečúcom elektrickom prúde cievkou vťahované dovnútra. Keď je cievka vypnutá, dochádza k uvoľneniu tejto energie vo forme napäťovej špičky s opačnou polaritou.

Na elimináciu týchto špičiek bola pridaná zhášacia dióda, ktorá chráni tranzistory a ďalšiu elektroniku.

LED diódy indikujú zopnutie tranzistoru, pričom sériový rezistor obmedzuje prúd na približne 2.2 mA.



Obr. 4.1: Zapojenie napájacej časti

Aby sa zabránilo náhodnému aktivovaniu tranzistorov v dôsledku indukovaného napätia pri zapínaní, bol medzi Gate a zem pridaný rezistor s hodnotou  $100K\Omega$ . Tento rezistor v kombinácii s rezistorom riadiacim Gate tvorí napäťový delič. Pre udržanie funkčnosti oboch rezistorov je dôležité zvoliť veľký pomer ich nominálnych hodnôt.

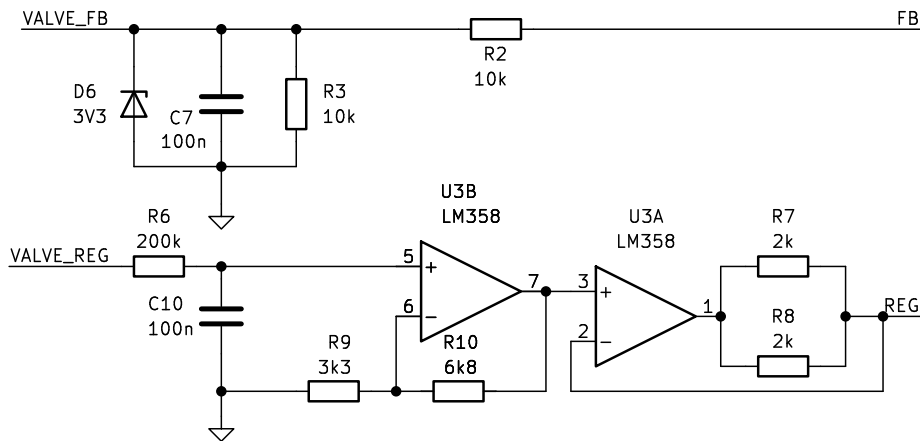
Pull-down rezistor zabezpečuje, že pri zapínaní nedôjde k nechcenému aktivovaniu tranzistoru kvôli náhodnému šumu v čase, keď výstupné porty mikrokontroléra nie sú inicializované.

Funkcia druhého rezistoru sa prejaví po inicializácii výstupných portov MCU, pričom vďaka veľkému pomeru hodnôt rezistorov bude na Gate dosiahnuté logické napätie z MCU.

### 4.3 Ovládanie regulátora tlaku

Na regulovanie tlaku pre fúkáciu harmoniku bol zvolený ventil ITV1030 od firmy SMC. Jedná sa 24V o regulátor tlaku s rozsahom od 0.005 do 0.5 MPa, riadený napätím v rozsahu 0 až 10 V. Tento ventil disponuje kontrolným výstupom s rozsahom 0 - 5 V.[11]

DAC prevodník vybraného mikrokontroléra pracuje v rozsahu od 0 do 3,3 V, čo je nedostatočné pre ovládanie tohto konkrétneho ventilu. Pre zosilnenie tohto rozsahu bola použitá kombinácia operačných zosilňovačov. Pre túto funkciu bol zvolený obvod



Obr. 4.2: Obvod pre ovládanie regulátora tlaku

LM358. Jedná sa o veľmi populárny integrovaný obvod, obsahujúci dva zosilňovače. Prvý zosilňovač zosilňuje vstupné napätie podľa nasledujúceho vzťahu:

$$A = 1 + \frac{R_{10}}{R_9} = 1 + \frac{6800}{3300} = 3 \quad (4.1)$$

Rezistory R9 a R10 boli zvolené tak, aby pri maximálnom napätí mikrokontroléra bolo na výstupe 10 V.

Druhý operačný zosilňovač tvorí buffer. Medzi zápornou spätnou väzbou a výstupom operačného zosilňovaču boli pridané rezistory R7 a R8. Tieto rezistory v paralelnom zapojení dávajú 1 k $\Omega$  a slúžia na limitovanie výstupného prúdu, ktorý je udavaný výrobcom na 20 mA.

Tento obvod je napájaný z 15V zdroja. Dôvodom tohto napájania je, že použitý operačný zosilňovač nie je typu Rail-to-Rail. Pre tento obvod je charakteristický výstupný rozsah od hodnoty záporného napájacieho napätia až po hodnotu VCC zníženú o 2 V.

Pred zosilňovačom je umiestnený dolnopriepustný filter. Úlohou tohto filtra je spomaliť náhle zmeny úrovne signálu pochádzajúce z DAC prevodníka. Toto spomalenie dáva zosilňovaču dostatočný čas na adekvátnu reakciu na konkrétne zmeny v signále, čím sa znižujú prekmity.

Sptaná väzba ventilu je privedená na vstupný pin ADC prevodníku mikrokontroléra. Tento signál je pred vstupom ešte upravený, aby nepresiahol medze snímacieho obvodu. Napätovým deličom je tento signál znížený z pôvodných maximálnych 5 V na 2.5 V. Zvyšok rozsahu prevodníku slúži len ako rezerva, a v prípade jej prekročenia bola pridaná zenerova dióda. Táto dióda chráni vstup mikrokontroléru jednak

pri prekročení limitu, a jednak pri prepolarizovaní. Kondenzátor C7 filtruje šum, a slúži ako antialiasingový filter pre ADC prevodník.

## 4.4 Komunikačné rozhrania

Na programovanie a ladenie bolo vybrané SWD (Serial Wire Debug) rozhranie. Toto rozhranie je štandardné pre procesory typu STM32. Na programovanie zariadenia bude použitý klon ST-Link. Protokol SWD využíva na komunikáciu dva vodiče:

- SWDIO - prenos dát
- SWCLK - prenos hodinového signálu

Toto riešenie uľahčí ladenie programu a pomôže pri hľadaní chýb.

Na výpis diagnostických a kontrolných hlášok bol z mikrokontroléru vyvedený UART port. Predpokladá sa, že zaintegrovateľný UART USB bridge bude primárne použitý na MIDI komunikáciu, a preto ho nebude možné použiť na tieto účely.

Konektor pre UART, bol zapojený tak, aby mal identické rozloženie pinov s komunikačným Bluetooth modulom HC-05. Tento Bluetooth modul plní rovnakú funkciu ako USB na UART prevodník, avšak poskytuje bezdrôtový prenos informácií. Na zabezpečenie komunikácie s týmto modulom je potrebné najprv povoliť tento typ komunikácie a tiež virtuálne porty zariadenia.

V budúcnosti by mohlo byť toto rozhranie využité aj na priamu MIDI komunikáciu. Ak by nebolo potrebné použiť tento Bluetooth modul, môže sa na tento konektor pripojiť štandardný UART prevodník alebo môže ostať odpojený.

### 4.4.1 MIDI

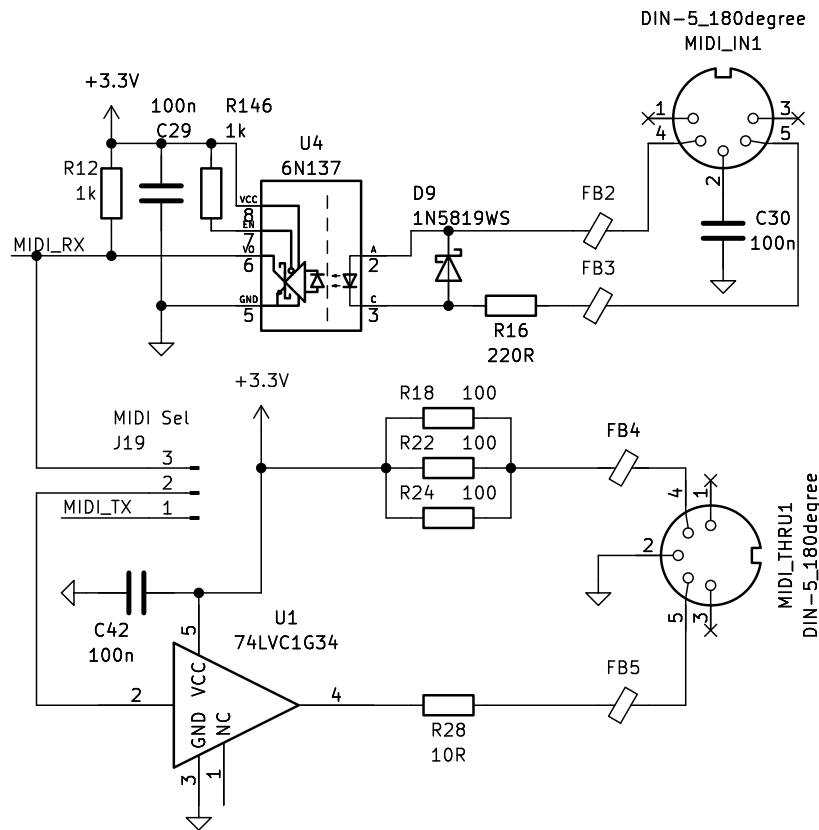
MIDI rozhranie je jedným z hlavných rozhraní implementovaných v tomto systéme. Jeho komponenty a hodnoty sú stanovené normou, ktorá musí byť dodržaná pre zachovanie kompatibility.

Pôvodne bol v systéme použitý 5V zdroj a obvod bol navrhnutý podľa pôvodnej MIDI normy. Vstup, teda MIDI In, bol izolovaný pomocou optoizolátora, ktorý bol zapojený antiparalelne k dióde a sériovo s rezistorom. Kvôli nízkej rýchlosti tohto oddelovacieho obvodu bol neskôr nahradený komponentom 6N137. Hoci norma MIDI špecifikuje použitie obvodu 6N138, v praxi sú tieto obvody takmer identické.[3] Rozdiel je v tom, že použitý komponent má hradlový výstup, zatiaľ čo odporúčaný funguje na princípe otvoreného kolektora.[12]

Vzhľadom na dostupnosť 5V napájania bol na výstup použitý invertujúci logický obvod 74H04, ktorý obsahuje šesť invertujúcich hradel. Pri jeho použití však zostali štyri hradlá nevyužitá.

Pre možnosť výberu medzi MIDI Thru a MIDI Out bola do zapojenia pridaná prepodka, ktorá umožňuje výber tejto funkcie. Je možné vybrať buď zrkadlenie signálu z MIDI In portu alebo odosielanie dát z mikrokontroléru. Toto riešenie rozširuje funkčnosť zariadenia tak, že môže byť použité na ovládanie iných MIDI zariadení.

### MIDI rozhranie podľa dodatku štandardu k 3,3V systémom



Obr. 4.3: Zapojenie MIDI rozhrania

S rozvojom a následnou štandardizáciou systémov s napätím 3.3 V bol k MIDI norme vydaný dodatok, ktorý sa týka práve týchto systémov. Tento dodatok hovorí o obvodovej úprave pri zachovaní kompatibility s ostatnými MIDI zariadeniami. Úpravy sa týkajú hlavne výstupnej, vysielačnej časti štandardizovaného zapojenia. Ostatné zmeny prevažne slúžia na zlepšenie vlastností EMI a EMC.

Príjímacia časť ostala zachovaná, a navyše s tým, že na koomunikačné piny MIDI In vstupu je možné pridať feritové jadrá. Pin č.2 ostal podľa pôvodnej schémy nezapojený. Podľa dodatku normy sa doporučuje uzemniť malou kapacitou (100 nF).

Rovnako ako v prípade prijímača, aj pre vysielač sa odporúča použitie feritových jadier umiestnených pred výstupným konektorom. Tieto jadrá slúžia na filtrovanie

špičiek, ktoré môžu vzniknúť pri odpojení alebo pripojení kábla, na ktorom je naindukované napätie.

Tento dodatok upravuje hodnoty rezistorov výstupnej časti tak, aby bola zachovaná kompatibilita s pôvodnými systémami pri použití 3.3V napájania. Toto riešenie je vhodné pre ušetrenie finančných prostriedkov, kedy odpadá potreba použitia 5V vetvy.

Oproti pôvodnej norme, kde výstupný buffer nebol definovaný konkrétnym obvodom ani zapojením, tento dodatok odporúča použiť tranzistor, logický obvod alebo ich kombináciu.

Na tento účel bol vybraný neinvertujúci buffer (74LVC1). Tento obvod je určený na prúdové zosilnenie signálu, vďaka čomu nebude preťažovaný výstup mikrokontroleru alebo oddelovacieho členu. Zároveň spĺňa časové limity vyhradené normou, a tak by nemalo dochádzať k oneskоровaniu signálu.[13]

Hodnoty rezistorov výstupnej časti sa zmenili z  $220\Omega$  na  $33\Omega$  a  $10\Omega$ . V sériovom zapojení so vstupným obvodom MIDI prijímaču spĺňajú definované prúdové požiadavky, vďaka čomu je tento systém kompatibilný aj s 5V systémami. Je dobré podotknúť, že by mal byť použitý 0.5W a 0.25W rezistor, kvôli výkonovým stratám pri skrate. Prvý rezistor bol zložený z troch  $100\Omega$  štvrt-watových rezistorov v paralelnom zapojení. Druhý rezistor pozostáva z paralelnej kombinácie dvoch  $20\Omega$  rezistorov.

## 4.4.2 USB

USB sa stalo veľmi populárnym rozhraním vďaka svojej univerzálnosti a širokej podpore zo strany hardwaru aj softwaru. V tomto prípade bude komunikácia vyriešená prevodníkom USB na UART.

UART patrí do základnej výbavy mikroprocesorov, s pomerne jednoduchou obsluhou. Zabudovanie USB prinesie výhody kompatibility s rôznymi systémami.

UART alebo aj sériová komunikácia je komunikačné rozhranie pracujúce asynchrónne. Jedná sa o duplexnú komunikáciu, kde sa jednotlivé bity vysielajú postupne. Oproti paralelnej komunikácii sa používajú len dva vodiče, na vysielanie a na príjem. Zdroj hodín tvorí vnútorný generátor, ktorým sa nastavuje baud-rate, teda prenosová rýchlosť. Hodnota baud-rate hovorí o tom, koľko logických zmien môže na zbernici nastať za jednotku času. Táto hodnota musí byť nastavená rovnako pre obe prepojené zariadenia, inak by dochádzalo k chybným rámcam a iným poruchám.

Jeden prenosový rámec tvorí start bit, 5 až 9 databitov, voliteľná alebo žiadna parita a stop bity (1, 2 niektoré zariadenia umožňujú 1.5). Prenos sa zahajuje stiahnutím zbernice z kludovej úrovne na logickú nulu, a spúšťa sa proces vzorkovania

v daných intervaloch. Jednotlivý bit je zaznamenaný viac-krát, na základe čoho sa potom určuje výsledná hodnota tohto bitu. Prevzorkovaním sa zvyšuje robustnosť komunikácie a kompenzuje sa odchylka v nastavených prenosových rýchlostiach. Komunikácia sa ukončuje uvoľnením zbernice do kludového stavu (po odvysialní stop bitu).

Na umožnenie komunikácie bol zvolený USB prevodník CP2102. Tento prevodník, ktorý podporuje USB 2.0, je určený pre prevod USB signálov na sériovú UART komunikáciu. Vyznačuje sa širokou škálou komunikačných rýchlostí a umožňuje rôzne nastavenia prenosového rámca.[14]

Po pripojení CP2102, sa na počítači vytvorí virtuálny COM port, cez ktorý môžu komunikovať rozličné aplikácie. Pre bežné použitie postačujú štandardné vstavané ovládače. V prípade potreby rozsiahlejšej konfigurácie hardvéru poskytuje výrobca špeciálne ovládače, ktoré umožňujú prispôbiť fungovanie zariadenia podľa konkrétnych potrieb.[14]

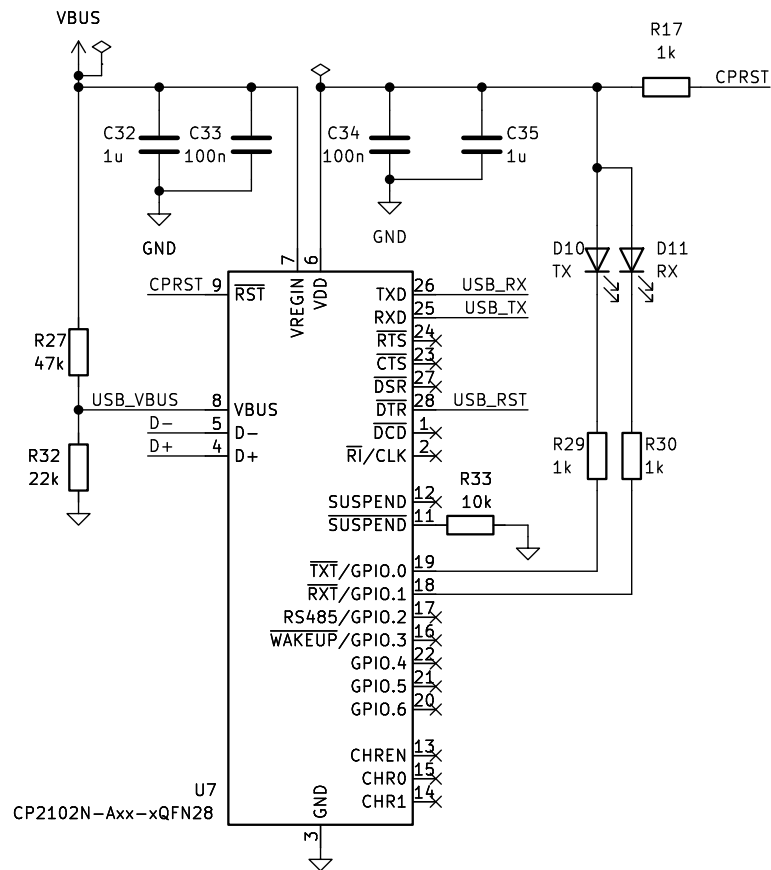
Prevodník môže byť napájaný priamo z USB s využitím interného regulátoru napätia. Tento regulátor môže zároveň tvoriť aj napájací modul zariadenia, ale vzhľadom na jeho parametre nebol použitý.

Obvod bol zapojený podľa technickej dokumentácie. Kondenzátory plnia úlohu filtrov pre napájacia napätia. Aby nedošlo k prekročeniu maximálnej úrovne vstupného napätia, pin VBUS je pripojený prostredníctvom deliča napätia. Resetovací pin je vybavený pullup rezistorom, čím sa zabráni nechcenému udržiavaniu obvodu v resetovanom stave. Na pin Suspend je pripojený pulldown rezistor, ktorý umožní fungovanie v idle režime. LED diódy signalizujú aktuálnu aktivitu na linke RX a TX.

Komunikačné piny boli pripojené priamo ku konektoru a mikrokontroléru podľa dokumentácie. Výstup DTR slúži na resetovanie mikrokontroléru a prepája sa prepojkou JP1. Táto funkcionálna nie je vyžadovaná, ale bola pridaná, v prípade potreby programovať MCU týmto rozhraním.

Ďalšou z možností je USB priamo prepojiť s natívnym USB rozhraním zvoleného mikrokontroléru. Spoločnosť STMicroelectronics poskytuje rozsiahly súbor knižníc pre svoje riešenia, ktoré uľahčujú vývoj s ich mikrokontrolérmi a inými komponentmi. Tieto knižnice pokrývajú rôzne funkcionality a protokoly, čo umožňuje rýchlejšie a efektívnejšie vývoj aplikácií. Avšak, knižnica pre MIDI protokol cez USB nie je zahrnutá v tejto ponuke.

Pre implementáciu MIDI rozhrania cez USB by bolo potrebné vyvinúť vlastné riešenie alebo hľadať knižnice tretích strán. To môže zahŕňať napísanie vlastného kódu pre spracovanie MIDI správ a ich prenos cez USB, alebo adaptáciu existujúcich open-source riešení, aby vyhovovali špecifikám. Toto prispôbenie implementácie môže byť časovo náročné, a tak sa zostalo pri použití prevodníku CP2102.



Obr. 4.4: Zapojenie UART prevodníku

## 4.5 Ovládacie prvky

Na zobrazovanie aktuálneho stavu zariadenia a jeho konfiguráciu bol vybraný 0,96-palcový OLED displej s rozlíšením 128x64 pixelov. Jedná sa o dvojfarebný display ovládaný radičom SSD1306.

Tento radič ovláda jednotlivé pixely, má zabudovanú potrebnú operačnú pamäť a informácie dostáva prostredníctvom paralelnej alebo sériovej komunikácie. Obvod je v prevedení modulu, s potrebnými komponentami pre jeho funkciu, s vyvedením komunikačným rozhraním  $I^2C$  (s adresou 0x78) alebo SPI.

Z dôvodu jednoduchosti a efektivity bola pre tento účel zvolená  $I^2C$  komunikácia, ktorá pre synchronnú komunikáciu využíva dva vodiče. Výstupné obvody sú zapojené v prevedení otvoreného kolektora, pričom je zbernica napájaná cez pullup rezistory.

Tento fakt je treba zohľadniť pri dlhších komunikačných vodičoch, kedy narastá kapacita a zvyšuje sa časová konštanta tvoriaca tento RC článok. Časová konštanta sa dá znížiť zmäňšením hodnoty pullup rezistoru (súčasne so zvýšením spotreby obvodu).



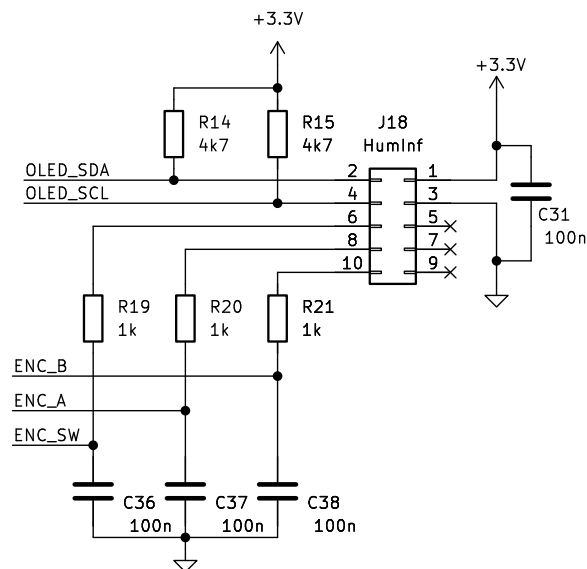
Kedže sa jedná o synchronne rozhranie, jeden z vodičov je využitý pre posielanie hodín. Štandardný takt pre toto rozhranie je 100 KHz, 400 KHz a v niektorých prípadoch až 1 MHz. Druhý vodič je využitý pre obojsmernú výmenu dát.

Systém  $I^2C$  je založený na hierarchickom princípe, kde nadriadené zariadenie (Master) iniciuje komunikáciu s podriadenými obvodmi (Slave). Každý z týchto podriadených zariadení má nastavanú unikátnu adresu v rozsahu od 1 do 127.

Jedno komunikačné slovo je tvorené deviatimi bitmi, pričom prvých 8 tvorí dáta a zvyšný bit slúži na potvrdzovanie prijatia správy.

Vysielanie sa začína štart podmienkou s adresou, pričom sa toto číslo posúva do-  
lava a zvyšný bit dáva informáciu o čítaní alebo zápise. Ak táto správa nebude potvr-  
dená, komunikácia sa ukočuje ako v bežnom prípade, stop podmienkou. V opačnom  
prípade, môže byť vysielaný ľubovoľný počet bytov, pokiaľ ich prijímač potvrdzuje.

Jednotlivé nastavenia, voľby a potvrdzovanie bude umožnené rotačným enkodérom. Rotačný enkodér funguje na princípe detekcie pohybu osy, ku ktorej je pri-  
pojený. Tento typ snímača je relatívny, čo znamená, že dokáže zaznamenať pohyb  
ale nie absolútnu polohu. Enkodér funguje na princípe disku, ktorý pôsobí ako spí-  
nač, generujúci dva obdĺžnikové signály posunuté o  $90^\circ$  elektrických. Tieto signály  
sú následne interpretované na určenie smeru a množstva pohybu.



Obr. 4.5: Zapojenie konektoru pre display a rotačný enkodér

Existujú rôzne spôsoby interpretácie signálov z enkodéra, závisiace od jeho kon-  
krétneho typu a nastavení. V niektorých prípadoch môže byť jeden z výstupov pou-  
žitý ako hodinový signál (clock), kde zmena logickej úrovne indikuje pohyb, a hod-  
nota na druhom výstupe potom určuje smer tohto pohybu. Alternatívne, môže byť

potrebné monitorovať zmeny na oboch výstupoch súčasne, pričom smer pohybu sa určuje na základe ich logickej zhody.

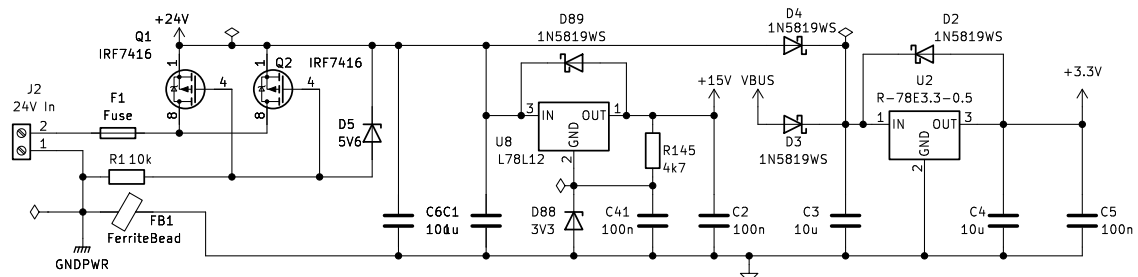
V prípade použitého mechanického enkodéra, rotácia generuje signály so zákmity, ktoré môžu spôsobiť nepresnosti v detekcii polohy. Preto je dôležité tieto zákmity ošetriť, buď na úrovni hardwaru pridaním filtrov, alebo v softvary. V tomto prípade boli na dosku pridané RC články, ktoré pomáhajú kompenzovať tieto zákmity, s časovou konštantou 100 ms, čím sa zvyšuje presnosť a spoľahlivosť símania.

## 4.6 Návrh dosky plošných spojov

Pred samotným návrhom boli najskôr jednotlivé periférie a zapojenia otestované na vývojovej doske. Po overení funkčnosti zapojenia a zistení kompatibility mikrokontroléru s nevyhnutnými knižnicami bolo nasledujúcim krokom vytvorenie schémy celého zapojenia.

Na tvorbu schémy a návrh dosky plošných spojov bol použitý softvér KICAD. Proces návrhu začína kreslením schémy a následným priradením púzdiar jednotlivým komponentom. Tento návrh je potom prenesený do prostredia pre návrh dosky.

Pri samotnom návrhu boli dodržané štandardné princípy. Konektory boli umiestnené na okraj dosky, aby bol zabezpečený ľahký prístup, a dbalo sa o vhodné rozmiesnenie komponentov, pre lepšie elektrické vlastnosti dosky.



Obr. 4.6: Zapojenie napájacej časti

Na napájanie bol zvolený 12V lineárny regulátor typu 7812. Tento regulátor má posunuté referenčné napätie o úbytok na zenerovej dióde, čo umožňuje vytvorenie 15V zdroja pre modul ovládania ventilu. Hodnoty potrebných komponentov boli určené na základe technickej dokumentácie tohto regulátora.

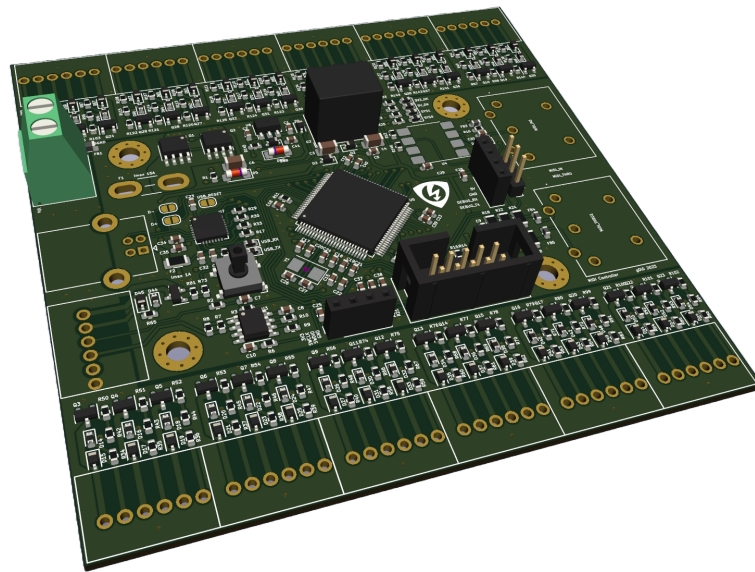
Napájaciu časť pre riadenie systému tvorí 3.3V spínaný zdroj. Pre tento účel bol vybraný regulátor R-78E3.3-0.5, ktorý je hotovým riešením celého zdroja v podobe malého modulu. K tomuto zdroju je však potrebné pripojiť filtračné kondenzátory podľa odporúčaní v technickej dokumentácii.[15]

Pri zapnutí solenoidu prúdi približne 300 mA, čo pri súčasnom zapnutí všetkých solenoidov dáva hodnotu až 11 A. Tento fakt je dôležité zohľadniť pri výbere vhodného napájacieho zdroja.

Napájacie cesty v zariadení je potrebné adekvátne dimenzovať, aby zvládali takéto prúdy. Preto boli použité širšie cesty a na niektorých miestach bol odstránený ochranný lak, aby sa ich vodivosť mohla zvýšiť pospájkovaním.

V prípade prepólovania napájania obvod chráni sústava dvoch unipolárnych P-kanálových tranzistorov. Keď je napätie privedené v správnej polarite, obvod sa najprv premostí cez zabudované diódy v tranzistoroch. Postupným nábehom sa pomocou rezistoru R1 začnú tieto tranzistory otvárať. Zenerova dióda D5 chráni Gate vstup tranzistoru pred prekročením maximálneho napätia  $V_{GS_{MAX}}$ . Ak je napätie priložené v opačnej polarite, diódy tranzistorov sú v reverznom smere a tranzistory zostávajú zatvorené, keďže medzi  $V_{GS}$  je nulový potenciál.

Ako ochrana pri skrate boli pridané tavné poistky, jedna je umiestnená za hlavným napájacím vstupom a druhá za USB konektorom.



Obr. 4.7: Doska plošných spojov

## 5 Implementácia softvéru

Táto kapitola sa zaoberá implementáciou softvéru do zariadenia s využitím editora kódu Visual Studio Code (VS Code). VS Code je bezplatný open-source editor navrhnutý ako univerzálny nástroj pre vývoj softvéru, podporujúci mnoho programovacích jazykov a technológií.

Samotný editor slúži primárne na úpravu textu, avšak pre programovanie je potrebné použiť vhodné moduly. S ohľadom na širokú podporu Arduina bol vybraný doplnok PlatformIO, ktorý je postavený na tejto platforme a rozširuje ju o možnosti ladenia a debugovania kódu. PlatformIO poskytuje podporu pre mnoho mikrokontrolérov a obsahuje rozsiahlu knižnicu na ovládanie periférií a externých modulov. Vďaka rozsiahlej komunite Arduina a PlatformIO je možné nájsť riešenia problémov na internete pomocou diskusných fór, dokumentácií a tutoriálov.

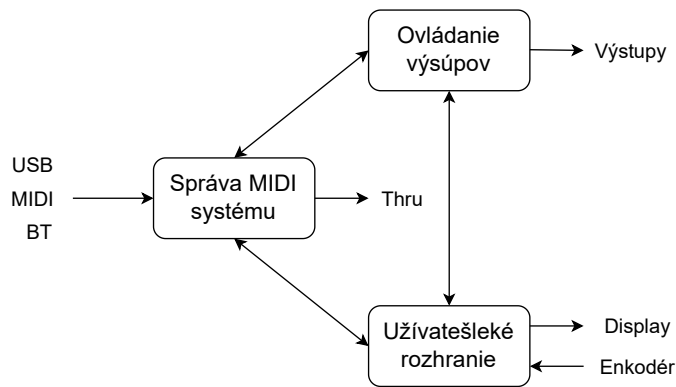
Zvolený mikrokontrolér disponuje pomerne vysokým výkonom, čo prináša možnosť využitia operačného systému pre efektívnu správu úloh a procesov. Implementácia operačného systému umožňuje rozdelenie programu na menšie moduly a vlákna, čo uľahčí ich správu a vykonávanie. Pre túto úlohu bol vybratý FreeRTOS, open-source operačný systém v reálnom čase, navrhnutý pre Embedded aplikácie. FreeRTOS poskytuje mechanizmy pre plánovanie úloh, správu pamäti a synchronizáciu procesov. Jednou z hlavných výhod použitia FreeRTOS je jeho nízka pamäťová náročnosť a škálovateľnosť.

Arduino program sa skladá z dvoch funkcií, `void setup()`, ktorá sa zavolá raz po reštarte alebo nabežnutí programu a `void loop()`, ktorá sa volá cyklicky. Systém FreeRTOS sa spúšťa zavolaním funkcie `vTaskStartScheduler`, čo predstavuje supersmyčku pre správu systémových prostriedkov, a tým pádom sa už systém k volaniu `void loop()` nedostane.

### Koncept systému

Samotný systém pozostáva z 3 funkčných celkov, subsystém pre správu MIDI, ovládanie výstupov a grafický výstup pre display. Tieto jednotlivé moduly sú prepojené a pomocou funkcií si navzájom predávajú dáta.

MIDI subsystém sa primárne zaoberá spracovaním správ z rôznych vstupných rozhraní, ako sú MIDI, USB a Bluetooth. Po prijatí dát sú tieto byty spracované pomocou objektu `RunningStatus`, ktorý identifikuje stavové a dátové byty a z nich vytvára kompletne MIDI správy. Následne sú tieto správy odoslané do objektu `MidiDevice`, ktorý sa stará o volanie callback funkcií a rieši rôzne módy zariadenia (napríklad Omni, Poly, Mono). Podľa nastaveného módu a príslušného kanála sú správy buďto spracované alebo ignorované, čím sa zabezpečuje správna distribúcia MIDI informácií.



Obr. 5.1: Topológia systému

Po prijatí kanálovej správy súvisiacej s generovaním tónov sa volajú funkcie ovládania výstupov. Nastavený MIDI mód a kanál priamo ovplyvňujú presmerovanie týchto správ, ktoré sú potom adresované jednotlivým výstupom alebo skupinám výstupov. Výstupy môžu byť aktivované okamžite po prijatí správy, alebo s oneskorením, nastaveným pomocou konfigurácie.

Výstupy pracujú v móde Duration alebo On/Off. V móde Duration je výstup automaticky vypnutý po uplynutí stanovenej doby a nemusí sa čakať na explicitný povel na vypnutie výstupu. Mód On/Off výstup aktivuje po prijatí povelu na aktiváciu a ostáva v tomto stave, kým nie je prijatý povel na jeho vypnutie.

Posledným modulom je správa zobrazenia, ktorý má dve hlavné funkcie. Zabezpečuje zobrazovanie aktuálneho stavu na displeji a zároveň spracováva informácie z rotačného enkodéra. Tento modul sa primárne skladá z dvoch základných zobrazení.

Po spustení systému sa na displeji zobrazuje základná obrazovka, ktorá poskytuje informácie o aktuálnom stave zariadenia. Táto obrazovka zobrazuje údaje o aktívnom MIDI móde, kanáloch, aktivovaných klávesách a ďalších relevantných informáciách. Druhé zobrazenie, nastavenia, slúži na nastavenie pracovného módu a konfiguráciu kanálov. To umožňuje používateľovi prispôsobiť fungovanie systému podľa zvoleného hudobného nástroja alebo inej výstupnej periférie.

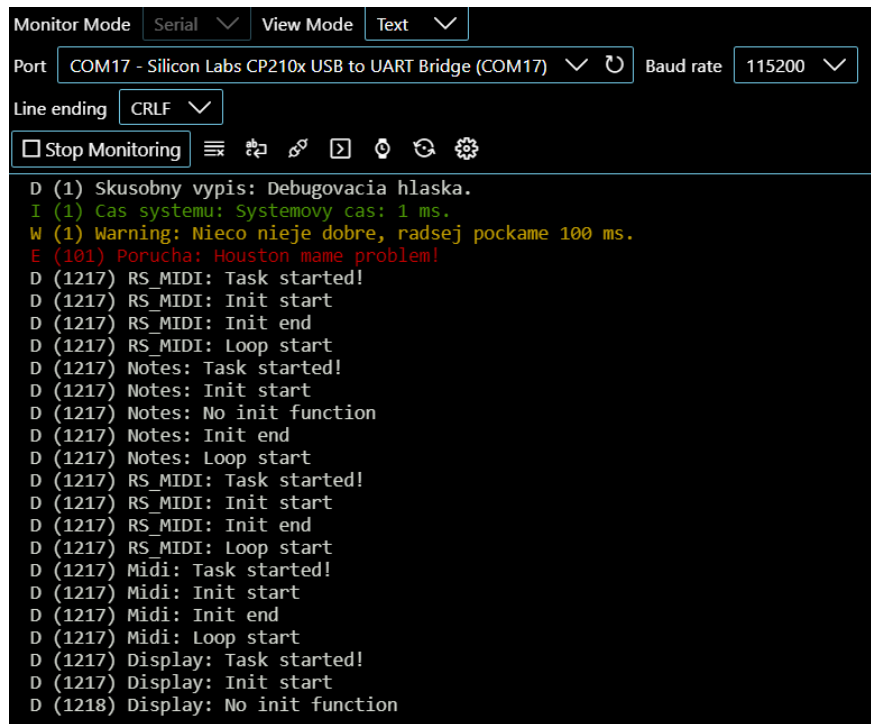
### Logovanie správ pre ladenie

Jednou z prvých vytvorených knižníc pre toto zariadenie bola knižnica s názvom Log. Jej úlohou je posielat ladiace správy na jeden zo sériových výstupov. Okrem toho slúžila aj na testovanie práce so systémom FreeRTOS.

Cieľom bolo vytvoriť logovacie rozhranie s podobnou funkcionalitou ako v ESP-IDF, kde sú logovacie správy rozdelené do štyroch úrovní: Error, Warning, Info a Debug. Každá úroveň má svoju priradenú farbu textu, ktorá je reprezentovaná ako

escape sekvencia, špeciálna postupnosť znakov, ktorú podporuje ANSI terminál.

Štruktúra každej logovacej správy pozostáva z farby textu (Escape sekvencia určujúca farbu textu pre danú úroveň správy), času (Aktuálny čas v milisekundách od spustenia programu), Tag (Názov modulu alebo komponentu, ktorý vyvolal danú logovaciu správu) a Text správy (Samotný obsah správy). Takto štruktúrované logovacie správy poskytujú prehľadný a farebne odlišený výstup, ktorý je možno nastaviť podľa úrovne logovania.



```
Monitor Mode Serial View Mode Text
Port COM17 - Silicon Labs CP210x USB to UART Bridge (COM17) Baud rate 115200
Line ending CRLF
[ ] Stop Monitoring
D (1) Skusobny vypis: Debugovacia hlaska.
I (1) Cas systemu: Systemovy cas: 1 ms.
W (1) Warning: Nieco nieje dobre, radšej pockame 100 ms.
E (101) Porucha: Houston mame problem!
D (1217) RS_MIDI: Task started!
D (1217) RS_MIDI: Init start
D (1217) RS_MIDI: Init end
D (1217) RS_MIDI: Loop start
D (1217) Notes: Task started!
D (1217) Notes: Init start
D (1217) Notes: No init function
D (1217) Notes: Init end
D (1217) Notes: Loop start
D (1217) RS_MIDI: Task started!
D (1217) RS_MIDI: Init start
D (1217) RS_MIDI: Init end
D (1217) RS_MIDI: Loop start
D (1217) Midi: Task started!
D (1217) Midi: Init start
D (1217) Midi: Init end
D (1217) Midi: Loop start
D (1217) Display: Task started!
D (1217) Display: Init start
D (1218) Display: No init function
```

Obr. 5.2: Výpis hlášok zo systému

Konštruktor triedy má parametre pre objekt sériovej linky a úroveň výpisu. Pre prípad, že by sériová linka nestihla odoslať predošlú správu alebo by funkcia na zápis bola zavolaná na viacerých miestach súčasne, je potrebné použitie vyrovnávacej pamäte.

Na tento účel bola použitá fronta, implementovaná priamo v knižniciach FreeRTOS, obsahujúca všetky potrebné synchronizačné mechanizmi. Fronta funguje na princípe FIFO(first in, first out). To znamená, že správy sú odoberané v poradí, v akom boli vložené. Dátový typ xQueue má pevne definovanú dĺžku, ktorá je určená pri jej vytváraní a je optimalizovaná pre situácie, kedy je potrebné zachovať poriadie správ. Používa sa napríklad pri zdieľaní dát medzi úlohami alebo pri predávaní jednotlivých udalostí.

Po zavolaní funkcie na výpis sa ako prvé spočíta potrebná veľkosť pamäte, ktorá

bude alokovaná pre správu. Ak alokácia prebehne úspešne, pointer správy s nakopírovanými dátami sa vloží do fronty.

Stav tejto fronty sa cysklicky kontroluje v osobitnom tasku. Ak je fronta prázdna, task sa uspáva, aby sa uvoľnili výpočetné prostriedky pre ďalšie úlohy. V opačnom prípade sa z fronty vyčíta správa, ktorá je poslaná na výpis sériovou linkou. Sériová linka obsahuje vnútorný buffer kde sa správa prekopíruje, a tak potrebná pamäť, po zavolaní zápisu, môže byť ihneď dealokovaná.

## 5.1 MIDI subsystém

Časť programu ktorá sa stará o MIDI správy pozostáva z dvoch celkov, RunningStatus a MidiDevice. RunningStatus sa stará o prijímanie bytov po sériovej komunikácii, spracovávanie a vyhodnocovanie týchto bytov. MidiDevice slúži na spracovanie kompletnej správy poskladanej RunningStatusom, jej presmerovanie a volanie callbackov.

Running status je spôsob optimalizácie prenosu MIDI správ, ktorý umožňuje znížiť dátový tok v MIDI sériových prenosoch. Napríklad, ak sa posielajú po sebe správy NoteOn s rovnakým kanálom a rovnakou notou, môže sa statusový bajt vynechať, a posielajú sa len dáta pre notu a intenzitu zvuku. Keď sa pošle iný typ správy, zmení sa kanál alebo nota, musí sa znova poslať plný formát správy s kompletným statusovým bajtom.

MidiDevice tieto kompletne správy spracováva, a ak sa jedná o kanálové dáta, vyhodnocuje číslo kanálu podľa pracovného módu Omni, Poly a Mono. Po spracovaní správy je zavolaný callback, volanie funkcie cez jej pointer. Callbacky sú rozdelené podľa dĺžky správ a podľa toho či sa jedná o správu kanálovú, systémovú alebo správu reálneho času.

### 5.1.1 Trieda RunningStatus

Funkcionalita RunningStatusu sa uplatňuje v situáciách, keď je MIDI sieť vyťažená alebo keď sa prenášajú veľké množstvá údajov naraz. Pri štandardizovanej rýchlosti 31,5 kBaudov sa môžu vyskytnúť oneskorenia, ktoré môžu ovplyvniť plynulosť hudobného výkonu.

Oneskorovanie prenosu správ môže mať nepriaznivý vplyv na reakčný čas zariadení, čo môže viesť k nesynchronizovaným alebo nekonzistentným hudobným prejavom.

Tento typ optimalácie neslúži len pre kanálové správy, ale obecne pre správy ktoré majú aspoň jeden dátový byte. Pre správy, ktoré pozostávajú len zo statusového bytu, sa neuplatňuje (bez statusového bytu by takáto správa nemohla existovať).

Pre tieto účely bola vytvorená trieda s názvom `RunningStatus`. Základným prvkom je dátový typ `MidiByte`. Tento typ má potrebné metódy na vyhodnotenie, či sa jedná o stavový alebo dátový byte a priradenie čísla kanálu. Z tohoto dátového typu je poskladaná správa, nesúca potrebné informácie, rozsah správy (počet dátových bytov), a ak sa jedná o kanálovú správu, aj informáciu o adrese.

Spracovanie prichádzajúcich bytov prebieha v samotnom tasku, jeden task na každé rozhranie (Bluetooth, MIDI a USB). Ak je v buffery sériovej komunikácie nahromadených viacero bytov, toto vyčítanie prebieha v jednom cykle tasku, aby sa predošlo stratám údajov v dôsledku pretečenia buffera a zbytočnému oneskoreniu správ.

Prijatý byte sa ďalej posiela ako vstupný parameter do metódy `ProcessMidiByte`. Metóda `ProcessMidiByte` je tvorená stavovým automatom, kde sa na začiatku rozhoduje, či sa jedná o stavový alebo dátový byte.

V prípade stavového bytu sa kontroluje, či stavový automat nebol nastavený na príjem špeciálnych systémových dát. Ak áno, stream špeciálnych systémových dát je ukončený. Následne sa aktualizuje stav automatu podľa prijatého bytu a v ďalšom cykle, na základe typu bytu, sa buďto očakáva príjem dátových bytov, alebo ďalšieho stavového bytu.

Prijatím dátového bytu sa stav automatu neaktualizuje, len sa kontroluje počet potrebných bytov (funkcionalita `RunningStatus`). Po prijatí potrebného počtu bytov sa vytvorí správa, ktorá je ďalej poslaná do buffera `xMessageBuffer`.

Datová štruktúra `xMessageBuffer`, umožňuje asynchrónnu komunikáciu medzi úlohami, podobne ako fronta. Narozdiel od fronty, je jeho použitie vhodné na situácie, kde je potreba flexibilného posielania a prijímania, bez potreby udržiavať poradie alebo obmedzenie veľkosti fronty. Taktiež poskytuje metódy pre príjem a zasielanie správ o ľubovolnej dĺžke, čo umožňuje dynamické prenosy a preto je jeho použitie na tento problém vhodné, keď sa správy môžu líšiť svojou veľkosťou.

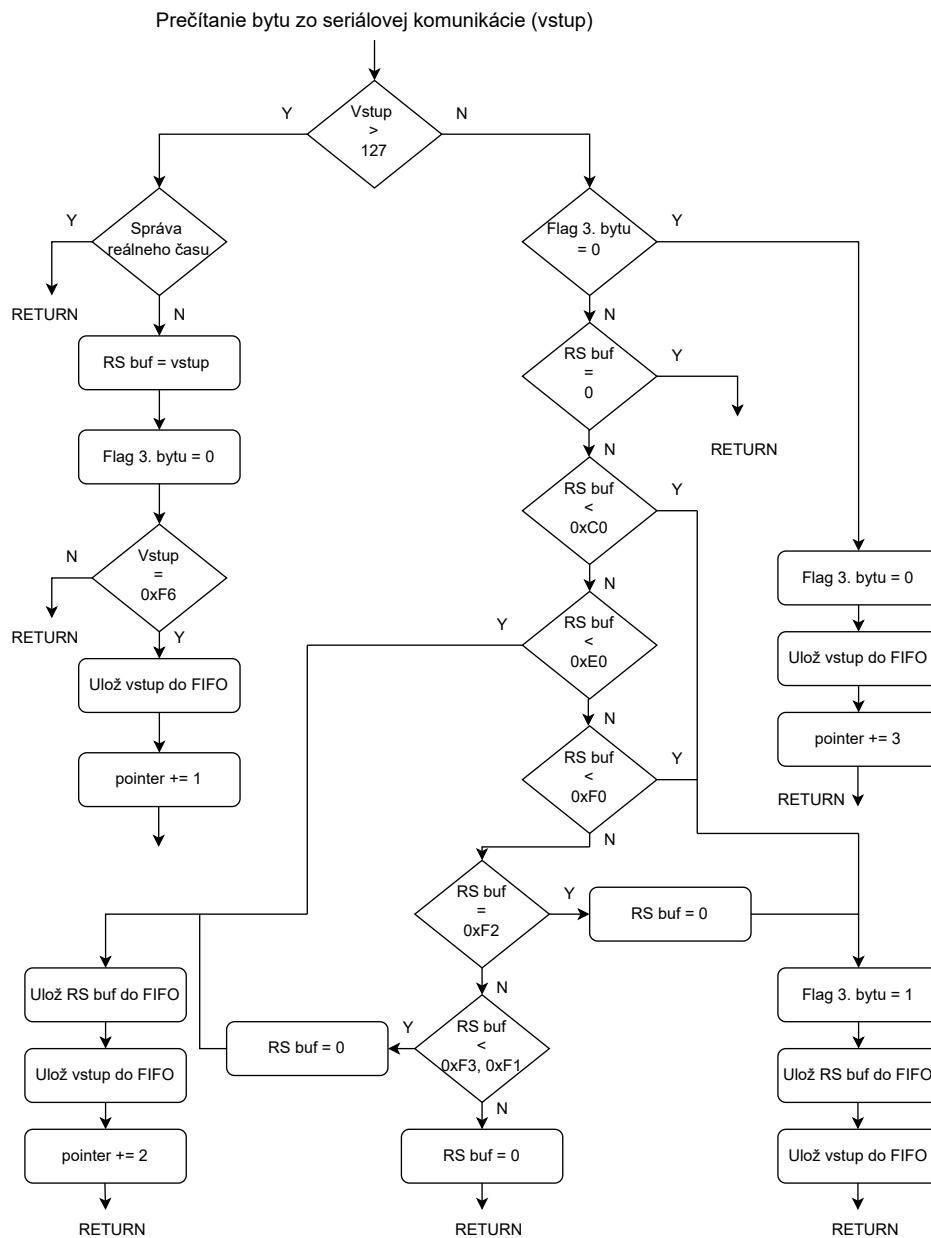
Špeciálne systémové dáta sú tvorené začiatkom zvláštnych systémových dát nasledované ľubovoľným počtom dátových bytov a ukončené špeciálnym statusovým bytom.

Jedným z prístupov spracovania týchto správ by bolo dopredu si naalokovať dostatočne veľký priestor a po prijatí ukončovacieho charakteru vytvoriť správu. Problémom je, že počet dátových bytov nie je dopredu známy a mohlo by sa stať, že dôjde k pretečeniu buffera.

Na tento problém bola použitá dátová štruktúra `xStreamBuffer`. Použitie tejto štruktúry umožňuje efektívny prenos dát medzi úlohami alebo úlohami a prerušeniami. Oproti `xQueue` a `xMessageBuffer`, je `xStreamBuffer` určený pre prenos dát vo forme streamu (kontinuálny tok dát) namiesto jednotlivých správ.

Prítomnosť a dostupnosť správ sa kontroluje metódou `available()`. Metódou `get()`





Obr. 5.3: Diagram stavového automatu

sa čítajú prijaté správy v takom poradí, v akom boli prijaté. Pre špeciálne systémové dáta existujú obdobné funkcie.

Pre prípad, že správy nechceme prijímať slúži metóda `SetActive()` a `SetActiveInactive()`. Zakázaním príjmu sa činnosť `RunningStatusu` neprerušuje, aby sa nestratila informácia pre stavový automat, zakazuje sa len vkladanie dát do front (xMessage-bufferu a xStreamBufferu).

## 5.1.2 Trieda MidiDevice

Spracovanie správ prebehia pomocou metód triedy MidiDevice. Táto trieda obsahuje všetky potrebné premenné pre správu aktuálneho módu, callbacky funkcií pre predanie dát a ich kontrolu.

Podobne ako pre RunningStatus, aj pre túto triedu bol vytvorený task. Task začína inicializáciou RunningStatusov pre Bluetooth, USB a MIDI následne sa prejde do supersmyčky. V supersmyčke sa cyklicky testuje dostupnosť správ z týchto zdrojov a volajú sa callbacky. Samotné riešenie dovoľuje prijímanie správ z viacerých periférii naraz, čo znamená že každé zariadenie má priradený svoj vlastný RunningStatus, ale vyhodnocovanie prebieha spoločne.

V supersmyčke sa volá metóda Loop() so vstupným parametrom Midi správy a dochádza k vyhodnoteniu druhu správy. Prvou podmienkou je, či sa jedná o správu kanálového módu.

Správy kanálového módu sú spojené so správaním a vyhodnocovaním kanálových správ. Týmito správami sa primárne nastavuje pracovný režim ako je Omni, Mono a Poly. Do tejto skupiny patria ďalej príkazy na vypnutie nôt a resetovanie zariadenia do základného stavu. Tieto správy zdieľajú spoločný stavový byte s kanálovou správou kontroléru (Controll change).

Ak sa jedná o kanálovú správu, je ďalej poslaná do metódy Filter(). Úlohou tejto metódy je odfiltrovanie správ podľa pracovného režimu, ktoré nie sú určené pre toto zariadenie. V režime Omni-Off Poly sa kontroluje zhoda nastaveného kanálu s kanálom správy, a móde Mono sa kontroluje kanál správy s nastaveným intervalom. Obsah validnej správy sa posiela ďalej do funkcie callbacku.

Spracovaním správy reálneho času a spoločnej systémovej správy sa ako prvé skontroluje pointer callbacku, ktorý je potom volaný. Samotná kontrola nieje potrebná, pretože tieto správy nenesú informáciu o priradenom kanály.

Trieda ďalej obsahuje potrebné metódy pre správu kanálov a nastavovanie režimov. Pre callbacky bol vytvorený spoločný dátový typ, ktorý je ukazovateľ na funkciu. Pred volaním sa vždy kontroluje, či ukazuje na validnú adresu (nerovná sa nullptr).

## 5.2 Ovládanie výstupov

Hlavnou úlohou tohto zariadenia je ovládať výstupy na základe prijatých MIDI povelov. Vzhľadom na veľkú diverzitu hudobných nástrojov je nevyhnutné, aby sa dal výstup konfigurovať podľa potrieb použitého hudobného nástroja. Každý hudobný nástroj má svoje špecifické požiadavky. Niektoré nástroje generujú tón počas celého

trvania aktivity (fúkacia harmonika), zatiaľ čo pre iné platí, že tón postupne sám doznie (metalofón).

Na základe toho bola vytvorená knižnica OutPP (output push-pull), ktorá zjednodušuje manipuláciu s výstupmi. Hlavným cieľom knižnice je rozšíriť základnú funkcionality `digitalWrite()` o možnosť invertovať výstup.

Inicializácia výstupného pinu prebieha v konšuktore, kde sa ako vstupné argumenty predáva číslo pinu a hodnota invertovania. V prvom kroku sa pin nastavuje ako vstupný, aby sa zabránilo nežiadúcemu impulzu v prípade, že by už bol nastavený ako výstupný. Po tomto nastavení sa hodnota invertovania použije na stanovenie počiatočného stavu výstupu, ale ešte sa neprejavuje na výstupe. Nakoniec sa pin nastaví ako výstup a je pripravený na používanie.

Jednou z ďalších požiadaviek bolo umožniť nastavenie oneskorenia pre zapnutie a vypnutie výstupu po jeho aktivácii. Na splnenie tejto požiadavky bola vytvorená trieda `OutPPAdv`, ktorá rozširuje triedu `OutPP` dedením a pridáva tieto oneskorenia, ako aj módy režimu. K dispozícii sú dva módy, `OnOff` a `Duration`.

V móde `OnOff` je výstup aktivovaný po prijatí príkazu a zostáva aktívny, kým nepríde príkaz na vypnutie. To znamená, že po aktivácii výstup pretrváva v tomto stave, kým nedostane príkaz na vypnutie.

V móde `Duration` je výstup aktivovaný po prijatí príkazu a zostáva aktívny počas určenej doby (periódy), ktorá je nastavená. Po uplynutí tejto doby sa výstup automaticky vypne. V tomto móde je možné vypnúť výstup okamžite pomocou príkazu na vypnutie, bez ohľadu na to, či perióda skončila.

Správne časovanie bolo dosiahnuté pomocou systémových programovateľných časovačov, ktoré sú integrované v knižniciach `FreeRTOS`. Tieto časovače sú schopné pracovať s presnosťou na úrovni milisekúnd (záleží od časovej základne systému), čo umožňuje presné riadenie časových udalostí v aplikácii. Pri nastavení časovača je možné definovať rôzne pracovné režimy, ako napríklad periodický časovač alebo jednorázový časovač.

Ukazovateľom na funkciu sa nastavuje callback, ktorý sa spustí po pretečení časovača. Tento mechanizmus umožňuje vykonávať určité akcie v aplikácii v pravidelných intervaloch alebo po uplynutí stanovenej doby. Avšak pri vysokom počte časovačov nastal problém s nesprávnym volaním callback funkcií, čo viedlo k nekonzistentnému správaniu aplikácie.

Problém nesprávneho volania callback funkcií pri veľkom počte časovačov bol vyriešený implementáciou cyklického kontrolovania v cyklickom tasku. Pri zavolaní funkcie `Set()` sa vyčíta aktuálny systémový čas a k nemu sa pripočíta perióda oneskorenia, obdobne aj pre funkciu `Reset()`.

Tieto metódy nevolajú aktivačné funkcie triedy `OutputPP`. Namiesto toho sa manipulácia s výstupmi vykonáva vo funkcii `Loop()`, ktorá je volaná z pomocného

tasku. V tejto funkcii prebieha jednoduchá kontrola, kde sa porovnáva nastavená hodnota z funkcií Set() alebo Reset() s aktuálnym systémovým časom. Ak je nastavená hodnota menšia ako aktuálny systémový čas, vykoná sa príslušná akcia s výstupom. Nakoniec sa porovnávaná premenná nastaví na maximálnu hodnotu aby nedošlo k chybnjej aktivácii v ďalšom cykle.

Na prispôsobenie triedy OutputPPAdv pre MIDI potreby, bola vytvorená trieda Note, ktorá rozširuje jej funkcionalitu a prispôsobuje ju pre potreby MIDI. Trieda Note poskytuje dodatočné metódy a atribúty, ktoré umožňujú lepšie reprezentovať hudobnú notu a pracovať s ňou v kontexte MIDI komunikácie. Dopĺňa ju o priradený kanál, prahovú hodnotu aktivácie a povolenie a zakázanie noty.

Priame rozhranie medzi MidiDevice a Note triedou tvorí NoteHandler. Pre využitie všetkých pracovných módov bola vytvorená štruktúra MidiGroup, združujúca noty s rovnakým kanálom.

Vytvorením objektu NoteHandler dochádza k alokácii NoteGroup, ktorá sa používa pri pracovnom móde Omni zapnuté, a alokácii poľa o veľkosti 15 prvkov (podľa počtu MIDI kanálov) pre mód Omni vypnuté. Na konci konštruktoru sa volá metóda Update(), ktorá prerozdelení noty podľa priradeného kanálu pre režim Omni vypnuté.

Vytvorením štruktúry NoteGroup sa ďalej alokuje pole pre uchovávanie priradených nôt. Metóda Get() vracia prvok z tohto poľa na základe vstupného indexu. Veľkosť poľa zostáva konštantná, avšak mení sa počet obsadených prvkov. Neobsadené prvky majú hodnotu nullptr. Vstupný index môže nadobúdať hodnoty v intervale 0 a 127 (podľa rozsahu dátového MIDI bytu).

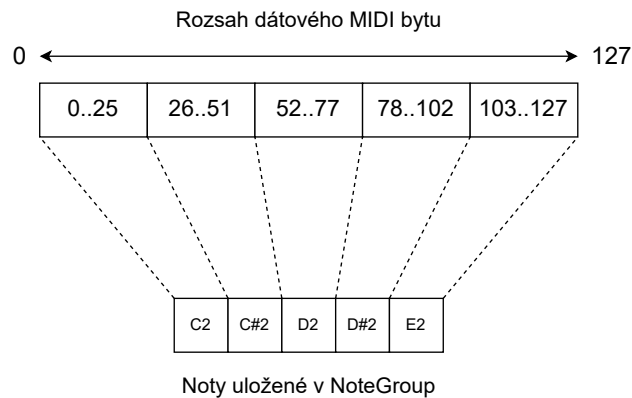
Výpis 5.1: Implementácia metódy Get

```
inline Note_c * Get(uint8_t idx) 1
{ 2
    // rovnomerne rozdelenie poľa, podľa jeho veľkosti 3
    return _notes [ (_used * (idx & 0b01111111)) >> 7]; 4
} 5
```

Hlavnou úlohou metódy Get() je transformovať vstupný index, ktorého rozsah je pevne daný, na interval poľa, ktorý sa môže dynamicky meniť. Veľkou výzvou bolo zabezpečiť, aby bol výpočet realizovaný len pomocou celočíselnej aritmetiky a aby bol čo najefektívnejší.

Na začiatku výpočtu sa na vstupný parameter, index, aplikuje maska 0x7F (127 v desiatkovej sústave), čo obmedzuje rozsah na interval MIDI dátového bytu. Tento upravený index sa potom vynásobí počtom použitých nôt a nakoniec sa vydelením číslom 128 (počet hodnôt od 0 do 127).

Samotný výpočet sa nejaví komplexne, ale zabraňuje pretečeniu poľa a zároveň vstupný interval rozdeľuje na interval 0 až po počet použitých nôt bez jednej. V



Obr. 5.4: Rozdelenie intervalu pre noty

celočísľelnej aritmetike sa delenie zaokrúhľuje vždy nadol. Miesto delenia číslom 128 bol použitý bitový posun o 7 bitov doprava, ktorý predstavuje efektívnejšie delenie v binárnej sústave.

### 5.3 Zobrazovanie a ovládanie

Pre pohodlné a intuitívne ovládanie bol použitý mechanický rotačný enkodér s kombináciou pixelového displeju. Po nabehnutí systému je na display zobrazená základná obrazovka, informujúca používateľa o základnom stave.

Na vrchnej časti sa nachádzajú základné informácie týkajúce sa MIDI, ako sú nastavené kanály a pracovný režim. Spodok displeja zobrazuje aktuálny stav klávesov, informácie týkajúce sa aktivácie nôt.

Stlačením enkodéru sa zobrazí sekundárna obrazovka pre nastavenia. Používateľ má možnosť si nakonfigurovať základné MIDI parametre a nastaviť výstupy podľa potrieb aplikácie.

#### Vykresľovanie informácií na displeji

Zobrazovacia jednotka je tvorená modulom obsahujúcim OLED displej s radičom SSD1306. Tento radič zabezpečuje konverziu prijatých príkazov z jedného z dostupných komunikačných rozhraní (I2C alebo SPI), a následne riadi zobrazenie pixelov na matrixovom OLED displeji s rozlíšením 128x64 pixelov. Radič má implementované len základné funkcie ako je napríklad skrolovanie a invertovanie farieb.

Nevýhodou je, že tento radič neposkytuje žiadnu vlastnú znakovú sadu ani textové fonty. Pre vyriešenie týchto nedostatkov bola využitá externá knižnica od Adafruit pre ovládanie SSD1306, ktorá má tieto funkcie zabudované. Knižnica implemen-

tuje celú správu zobrazenia pre SSD1306 vrátane inicializácie displeja, vykresľovania geometrických primitív, textu a bitových polí.

Pred použitím pre korektnú funkciu displeju je potrebné vedieť jeho základné parametre. Parametre I2C adresy a rozlíšenie displeja sa zadávajú v konštruktoře tejto triedy a sú nevyhnutné pre korektnú funkciu displeja.

Zobrazovanie informatívnych ikoniek sa realizuje zápisom bitového poľa do displeju. Jednoduché symboly je možné tvoriť aj priamo v programe, pomocou geometrických tvarov a funkcií na ich vyplnenie. Avšak táto cesta nieje veľmi efektívna, samotná príprava a výpočet trvá dlho, čo zabraňuje mikrokontroléru vykonávať iné dôležité úlohy.

Na základe týchto požiadaviek bola vytvorená nová trieda dedením, ktorá rozširuje bázu triedu AdafruitSSD1306 a umožňuje vykresľovanie bitových polí. Tieto ikonky boli nakreslené v bitmapovom formáte a následne pomocou vhodného nástroju skonvertované na bitové pole. Dáta potrebné na vykreslenie sú uložené v dátovom type Bitmap, ktorý obsahuje informácie pre zobrazenie a veľkosti obrázka.

Výpis 5.2: Príprava bitového poľa pre zobrazenie stavu klávesov

```
// pocet bitov na byty 1
size_t xMax = bitmap_midi_default.width >> 3; 2
size_t yMax = bitmap_midi_default.height; 3
for (size_t y = 0; y < yMax; ++y) 4
{ 5
    for (size_t x = 0, i = 0; x < xMax; ++x, i = y * xMax + x) 6
    { 7
        outbuf[y][x] = bitmap_midi_default.map[i]; 8
        outbuf[y][x] ^= bitmap_midi_mask.map[i] 9
                        & mask[(keys >> (x << 1)) & 0b11]; 10
        outbuf[y][x] ^= bitmap_midi_smask.map[i] 11
                        & mask[(skeys >> (x << 1)) & 0b11]; 12
    } 13
} 14
```

Použitím vyrovnávacej pamäte sa šetrí procesorový čas a znižuje sa priama manipulácia s displejom. Vyrovnávacia pamäť je najprv vyčistená funkciou Clear(). Grafickými funkciami sa do takto pripravenej pamäte zapisujú jednotlivé bity na predurčené pozície. Príkazom Display() je obsah odoslaný do displeja na následné zobrazenie. Tento prístup šetrí výpočetný výkon a predstavuje efektívnejšie spracovanie dát.

## Knižnica rotačného enkodéru

Mechanický enkodér obsahuje rotačný disk s výstupkami, ktorý pri rotácii premoštuje vstupy A a B so spoločnou zemou. Signáli generované na týchto výstupoch sú fázovo pootočené, vďaka čomu sa dá identifikovať relatívna zmena polohy.

Enkodér, ktorý bol vybraný, v jednom tik-u generuje dvojicu napätových zmien na oboch kanáloch. Tento enkodér umožňuje využitie nábežnej alebo dobežnej hrany jedného zo signálov na detekciu pohybu. Na základe logického stavu druhého signálu je následne možné presne určiť smer otáčania.

Spínaním mechanických kontaktov dochádza aj k nechcennému efektu, zakmitávaniu, čo môže viesť k nesprávnemu vyhodnoteniu polohy. Tento nechcenný efekt je ošetrený vstupnými filtrami, ktoré tento prechodový jav potláčajú.

Zmena polohy je zaznamenávaná v rutine prerušenia, ktoré je vyvovalé nábežnou hranou na kanáli A. Vyčítaním kanálu B sa aktualizuje počítadlo a zavolajú.

Pre tieto účely bola vytvorená trieda, ktorá obsluhuje činnosť tejto vstupnej periférie. V konštruktoe sa inicializujú vstupy a nastaví sa kludová úroveň pinov, podľa ktorej sa zapínajú buď pull-down alebo pull-up rezistory. Pre vstupy spínačov a kanálu A sa nakoniec povolia prerušenia. V týchto prerušeniach sa vyhodnocuje stláčanie tlačidla a poloha bežca enkodéru. Volajú sa príslušné callbacky funkcií a aktualizujú sa stavové premenné, ktoré slúžia na ďalšie vyhodnotenie programom.

## 6 Výsledky práce

Cielom práce bolo vytvoriť zariadenie, ktoré umožní ovládanie hudobného nástroja, metalofónu alebo fúkacej harmoniky. Úlohou bolo zariadenie pripraviť na ich odlišné metódy ovládania.

Harmonika generuje tón stlačením klávesy a tento tón zostáva aktívny počas celej doby držania klávesy. Pre metalofón platí, že aktiváciou sa natiahne kladivko, a pri deaktivácii vratná pružina udrie kladivkom po železnom kameni metalofónu. Úder kladivka spôsobí vybráciu železného kameňu, ktorý generuje trvajúci tón.

Degradácia materiálu uloženia kládívkov spôsobila falošné znenie tónov, a preto museli byť vymenené. Výmenou tohoto uloženia za silikónovú trubičku sa metalofónu vrátil jeho špecifický zvuk s dlhým doznievaním.

Tento výrobok mal vychádzať z pôvodného zariadenia pre ovládanie fúkacej harmoniky, mal opraviť jeho chyby a zlepšiť nedostatky. Ukázalo sa, že niektoré drobné závady by vyžadovali zásach do dosky plošných spojov a spolu v kombinácii s nedostatkom kremíkových čipov dali vzniknúť kompletne novému cenovo dostupnému hardvéru. Nový návrh opravuje problémy s priamou MIDI komunikáciou a riešenie dopĺňa o komunikačné rozhranie Bluetooth a širšie možnosti nastavení podľa potrieb použitia.

Návrh nového zariadenia začal stanovením jeho požiadaviek, definovaných zadáním a v súlade s MIDI normou. MIDI norma definuje aj obvodovú časť, ktorá musela byť zohľadnená hlavne v ranných fázach návrhu zariadenia.

Prvým krokom v procese testovania bolo overenie kompatibility potrebných knižníc na vývojovej doske DevEBox s osadeným procesorom STM32F407. Tento proces ukázal, že knižnice sú správne nainštalované a kompatibilné s danou hardvérovou platformou.

Po úspešnom overení kompatibility knižníc nasledovalo testovanie všetkých potrebných periférií a modulov na vývojovej doske. Toto zahŕňalo testovanie komunikácie, ako aj overenie funkčnosti jednotlivých periférií a externých modulov. Počas testovania sa ukázalo, že mikrokontrolér na vývojovej doske vyhovuje požiadavkám systému, čo viedlo k jeho výberu pre ďalšie použitie.

Následoval návrh a kreslenie elektrickej schémy. Elektrická schéma je organizovaná po funkčných celkoch, pre napájanie, komunikáciu, ovládanie a spínanie. Súčiastky boli primárne volené na základe technických dokumentácií, aby sa zaručila ich správna funkcionálnosť v systéme. Do obvodu boli pridané aj potrebné ochrany proti skratu, nadprúdu a prepólovaniu.

Takto nakreslená schéma sa následne preniesla do prostredia na tvorbu dosiek plošných spojov. Počas návrhu sa dbalo, aby boli dodržané základné princípi návrhu,



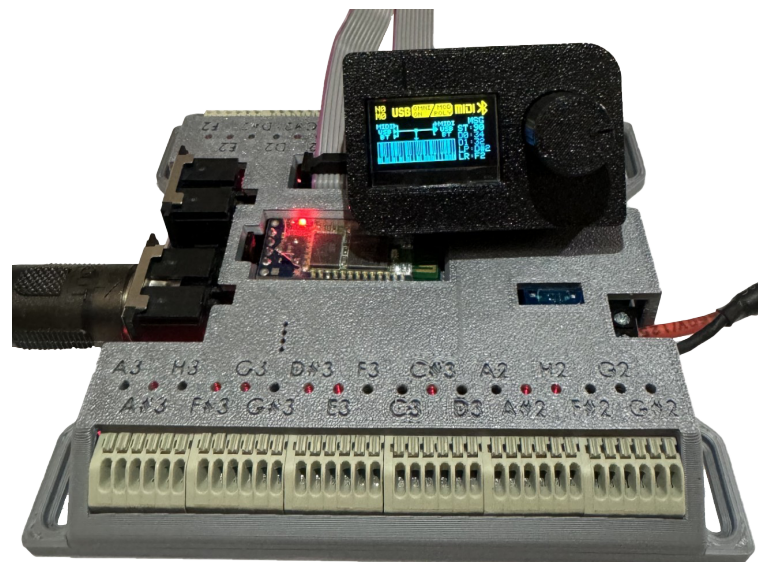
oddelenie výkonovej zeme, adekvátne šírky ciest, rozliatie zeme a logické rozmiestnenie komponentov.

Dokončený návrh bol vyrobený externou firmou JLCPCB, a čiastočne osadený SMD komponentami. Tento proces ušetril množstvo času pri manuálnom osádzaní THT komponentov, ako boli hlavne konektory.

Zhotovená doska bola potom oživená pomocou laboratórneho zdroja. Limitácia prúdu bola nastavená tak, aby v prípade chýb nedošlo k poškodeniu komponentov a aby sa zabezpečil priestor pre ďalšie opravy. Oživenie prebehlo bez väčších problémov, a doska bola pripravená na implementáciu programu.

Program bol vytváraný postupne, a rozdelený do funkčných celkov. Ako prvé boli vytvorené pomocné triedy a funkcie slúžiace na lepšie ladenie kódu. Postupne bolo implementované rozhranie MIDI, triedy pre ovládanie výstupných periférií a nakoniec užívateľské rozhranie. Veľkou výhodou je bohatá konfigurácia systému, ktorá dovoľuje nastavovať rôzne pracovné módy, konfigurovať výstupné kanály, a tým pádom prispôbiť zariadenie potrebám.

Sprvu bolo zariadenie otestované na fúkacej harmonike. Šlo hlavne o vyskúšanie rôznych konfigurácií výstupov na solenoidoch, a záťažové tesky, či je zariadenie prúdovo kompenzované na aktiváciu všetkých výstupov súčasne. Pre testovanie bolo vstupné napätie znížené na približne 20 V a následne bolo odskúšané s nominálnou hodnotou 24V.



Obr. 6.1: Midi sekvencér

Nakoniec bol výrobok pripojený k metalofónu. Terminály solenoidov boli usporiadané na pripojenie do svorkovnic na zadnej strane výrobku, a preto museli byť

predĺžené. Ako testovací zdroj MIDI správ poslužil program REAPER, v ktorom bola vytvorená testovacia skladba obsahujúca noty v celom rozrahu.

Tieto inštrukcie boli prevedené prevodníkom USB na MIDI a posielené do zariadenia pre vyskúšanie MIDI komunikácie. Následne pre test USB rozhrania sa v aplikácii loopMIDI vytvoril virtuálny MIDI port. Informácie z tohoto virtuálneho zariadenia ďalej putovaly cez program Hairless MIDI, ktorý ich odosielen cez sériové rozhranie priamo do zariadenia.

Zariadenie tieto informácie spracovalo a následne transformovalo na riadiace povely pre solenoidy, ktoré na mentalofóne rozohrali príslušné noty. Tieto informácie boli ďalej preposielené prostredníctvom výstupu MIDI Thru do sekvencéru fúkacej harmoniky.

Ukázalo sa, že sekvencér fúkacej hramoniky tieto povely prijímal, ale pravdepodobne dochádzalo k ich nesprávnemu interpretovaniu, a tým pádom k nesprávnemu stláčaniu klávesov. Miesto softverového MIDI Thru sa na výrobku prepojila prípojka pre priame zrkadlenie MIDI In, ktorá by mohla byť jedna z príčin, ale ani to nepomohlo s chýbným vyhodnocovaním informácií.

Navzdory týmto problémom sa nakoniec MIDI Thru vyskúšalo pripojením kanálu IN z predovníku MIDI na USB, a sledovala sa zhoda vyslaných a prijatých informácií. Prijaté povely sa presne zhodovali s vyslanými, čo potvrdilo správnu funkcionálnu hardverového dizajnu a činnosť pre softvérové Thru a priame MIDI Thru.

## Záver

Hlavnou úlohou bolo vytvoriť prístroj, ktorý dokáže ovládať hudobný nástroj, konkrétne fúkaciu harmoniku alebo metalofón. Toto zariadenie je koncipované ako MIDI sekvencér, ktorý by mal spĺňať základné požiadavky MIDI 1.0 komunikácie stanovené normou.

Seqvencér túto normu implementuje podľa popisu v prvej kapitole. Táto kapitola rozoberá vývoj štandardu od jeho založenia až po súčasné trendy, pričom sa kladie dôraz na aspekty, ktoré sú nevyhnutné pre ďalší návrh. Dôkladne opisuje fyzické rozhranie a štruktúru MIDI správ, s osobitným zameraním na kanálové správy, ktoré tvoria hlavnú komponentu tejto komunikácie. Záver kapitoly je venovaný súčasnému stavu MIDI, vrátane jeho integrácie s USB rozhraním.

Návrh zariadenia sa začal stanovením požiadaviek, ktoré by mal spĺňať. K ukázaní niektorých kľúčových aspektov mal poslúžiť prieskum trhu. Z prieskumu vyplynulo, že zariadenia podobného druhu sú na trhu skôr vzácnosťou než pravidlom, ale aj napriek tomu existujú obdodobné prístroje. Príkladom môžu byť komplexné systémy riadenia organov používané v kostoloch.

Zariadenie by malo byť primárne schopné ovládať jedno z dostupných hudobných zariadení, ovládané solenoidmi, ktoré sú popísané v následovnej kapitole. Pre vrátenie pôvodného prejavu metalofónu bola na tomto hracom nástroji vykonaná oprava a degradovaný materiál pod hracími železnými doskami bol vymenený za nový, s lepšími vlastnosťami.

Ddosiahnutie súčasného ovládania oboch nástrojov bol koncept systému vymyslený tak, aby zariadenie správu preposlalo naprieč celým systémom pri jej spracovaní. Na tieto účely slúži primárne MIDI a USB rozhranie, jednak pre vstup dát a jednak pre ich ďalšiu distribúciu. Do tejto skupinky rozhraní bola neskôr pridaná aj sériová komunikácia cez Bluetooth, ktorá však nieje taká stabilná ako pôvodné riešenia a pripájanie trvá pomerne dlho.

Stanovením požiadaviek sa postupne začalo výberom a testovaním komponentov, ktoré sú nevyhnutné pre jeho správnu funkcionálnosť. Zariadenie je postavené na mikrokontroléry STM32 ktorý sa stará o celú správu zariadenia, a osadené potrebnými komunikačnými a ovládacími modulmi. Výroba dosky plošných spojov prebehla pomocou externej firmy, ktorá sa čiastočne postarala aj o jej osadenie a zaspájkovanie.

Implementácie softvéru bola rozdelená do 3 hlavných celkov, pre príjem správ a ich spracovanie, obsluhu výstupov a nakoniec užívateľské rozhranie. Pre obsluhu systému bol použitý FreeRTOS, systém určený pre vstavané aplikácie, ktorý poskytuje plánovanie týchto úloh.

Táto práca opisuje nevyhnutné informácie pre návrh sekvencéru, a jej výstupom je funkčné zariadenie schopné obsluhovať hudobný nástroj. Sekvencér podporuje

príjem informácií prostredníctvom MIDI, USB a Bluetooth a ďalej ich preposiela podľa nastavenej konfigurácie Thru. Tieto povely mení na signály a príslušne ovláda jednotlivé výstupy podľa nastaveného pracovného módu MIDI a kanálu. Na hlavnej obrazovke sú zobrazené informácie týkajúce sa stavu systému, aktívnych nôt, režimu a poslednej prijatej správy. Obsluhou enkodéru je možné prejsť do menu a systém nakonfigurovať podľa potrieb. Návrh bol zakončený vymodelovaním a vytlačením krabičky.

Sekvencér nemusí slúžiť len na ovládanie hudobného nástroju. Svoje použitie by našiel aj pri svetelnej šou sprevádzanou hudbou, a v budúcnosti by mohol byť doplnený o prehrávanie skladieb z prenosného média.

# Literatúra

- [1] FORRÓ, Daniel, 1993. MIDI-komunikace v hudbě. 1. Praha: Grada. ISBN 80-85623-56-0.
- [2] ÚŘAD PRO TECHNICKOU NORMALIZACI, METROLOGII A STATNÍ ZKUŠEBNICTVÍ. MIDI (digitální rozhraní hudebních nástrojů) specifikace 1.0 (zkrácené vydání, 2015). 01.01.2018.
- [3] MMA TECHNICAL STANDARDS BOARD / AMEI MIDI COMMITTEE. (CA-033) MIDI 1.0 Electrical Specification Update. 2014
- [4] THE MIDI ASSOCIATION AND ASSOCIATION OF MUSICAL ELECTRONICS INDUSTRY (AMEI). MIDI 2.0 Specification Overview with Minimum Requirements. 01.01.2018
- [5] Introducing MIDI 2.0 [online], 2020. 01.01.2018. [cit. 2023-11-28]. Dostupné z: <https://www.soundonsound.com/music-business/introducing-midi-20>
- [6] USB Implementers Forum, Inc., USB Device Class Definition for MIDI Devices, Version 2.0. 5.5.2020
- [7] August 2016 TAO Cover Feature Article, Fourth Presbyterian Church [online], 2021. 01.01.2018. [cit. 2023-12-28]. Dostupné z: <https://www.agohq.org/august-2015-tao-feature/>
- [8] What is MIDI USB? A Beginner's Guide [online], 2023. 01.01.2018. [cit. 2023-12-26]. Dostupné z: <https://www.ac3filter.net/what-is-midi-usb/>
- [9] DS8626-STM32F405xx STM32F407xx [online], 2020. [cit. 2023-12-26]. Dostupné z: <https://www.st.com/resource/en/datasheet/stm32f407ve.pdf>
- [10] Alpha & Omega Semicon AO3400A [online], 2011. [cit. 2023-12-26]. Dostupné z: [https://datasheet.lcsc.com/lcsc/2311091734\\_UMW-Youtai-Semiconductor-Co---Ltd--A03400A\\_C347475.pdf](https://datasheet.lcsc.com/lcsc/2311091734_UMW-Youtai-Semiconductor-Co---Ltd--A03400A_C347475.pdf)
- [11] Compact Electro-Pneumatic Regulator Series ITV0000 [online], 2011. [cit. 2023-12-26] Dostupné z: <https://www.smc pneumatics.com/pdfs/ITVop.PDF>
- [12] Everlight Elec 6N137-V [online], 2013. [cit. 2023-12-26]. Dostupné z: [https://datasheet.lcsc.com/lcsc/2312271409\\_Everlight-Elec-6N137-V\\_C7032603.pdf](https://datasheet.lcsc.com/lcsc/2312271409_Everlight-Elec-6N137-V_C7032603.pdf)

- [13] DIODES INTERCORPORATED 74LVC1G34-SINGLE BUFFER GATE [online], 2016. [cit. 2023-12-26]. Dostupné z: [https://datasheet.lcsc.com/lcsc/1811061209\\_Diodes-Incorporated-74LVC1G34SE-7\\_C94111.pdf](https://datasheet.lcsc.com/lcsc/1811061209_Diodes-Incorporated-74LVC1G34SE-7_C94111.pdf)
- [14] SILICON LABS CP2102N-A02-GQFN28R [online], 2016. [cit. 2023-12-26]. Dostupné z: [https://datasheet.lcsc.com/lcsc/2304140030\\_SILICON-LABS-CP2102N-A02-GQFN28R\\_C964632.pdf](https://datasheet.lcsc.com/lcsc/2304140030_SILICON-LABS-CP2102N-A02-GQFN28R_C964632.pdf)
- [15] RECON R-78E-0.5 [online], 2021. [cit. 2023-12-26]. Dostupné z: <https://recom-power.com/pdf/Innoline/R-78E-0.5.pdf>

## Zoznam symbolov a skratiek

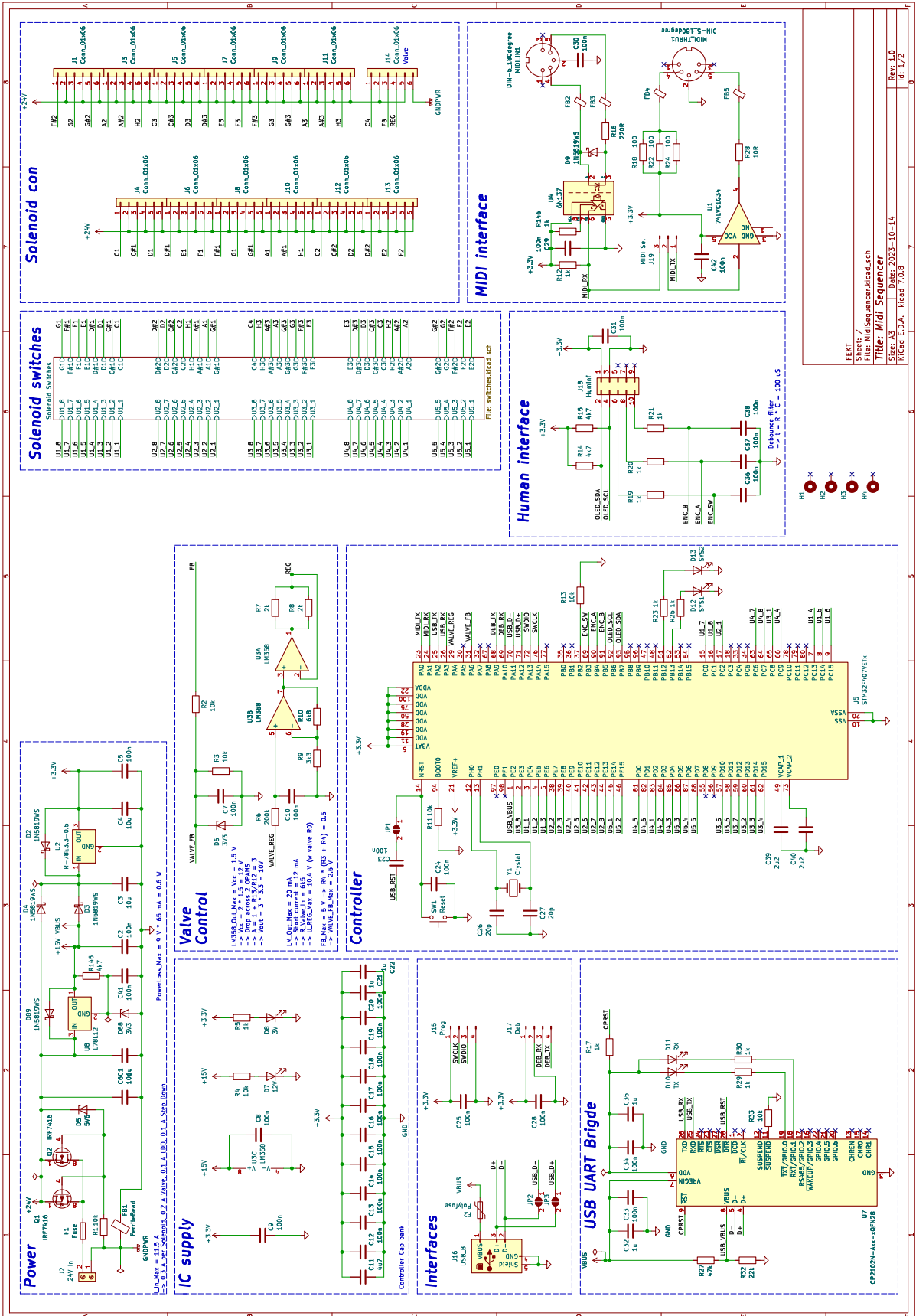
<b>CV</b>	Control Voltage – riadiace napätie
<b>DAW</b>	Digital Audio Workstation
<b>DCB</b>	Digital Communication Bus – digitálne komunikačné rozhranie
<b>MIDI</b>	Musical Instrument Digital Interface – digitálne rozhranie hudobných nástrojov
<b>MTC</b>	MIDI Time Code – MIDI časový kód
<b>UART</b>	Universal asynchronous receiver-transmitter
<b>USB</b>	Universal Serial Bus – univerzálna sériová zbernica
<b>USI</b>	Universal Synthesizer Interface – univerzálne rozhranie syntetizátorov

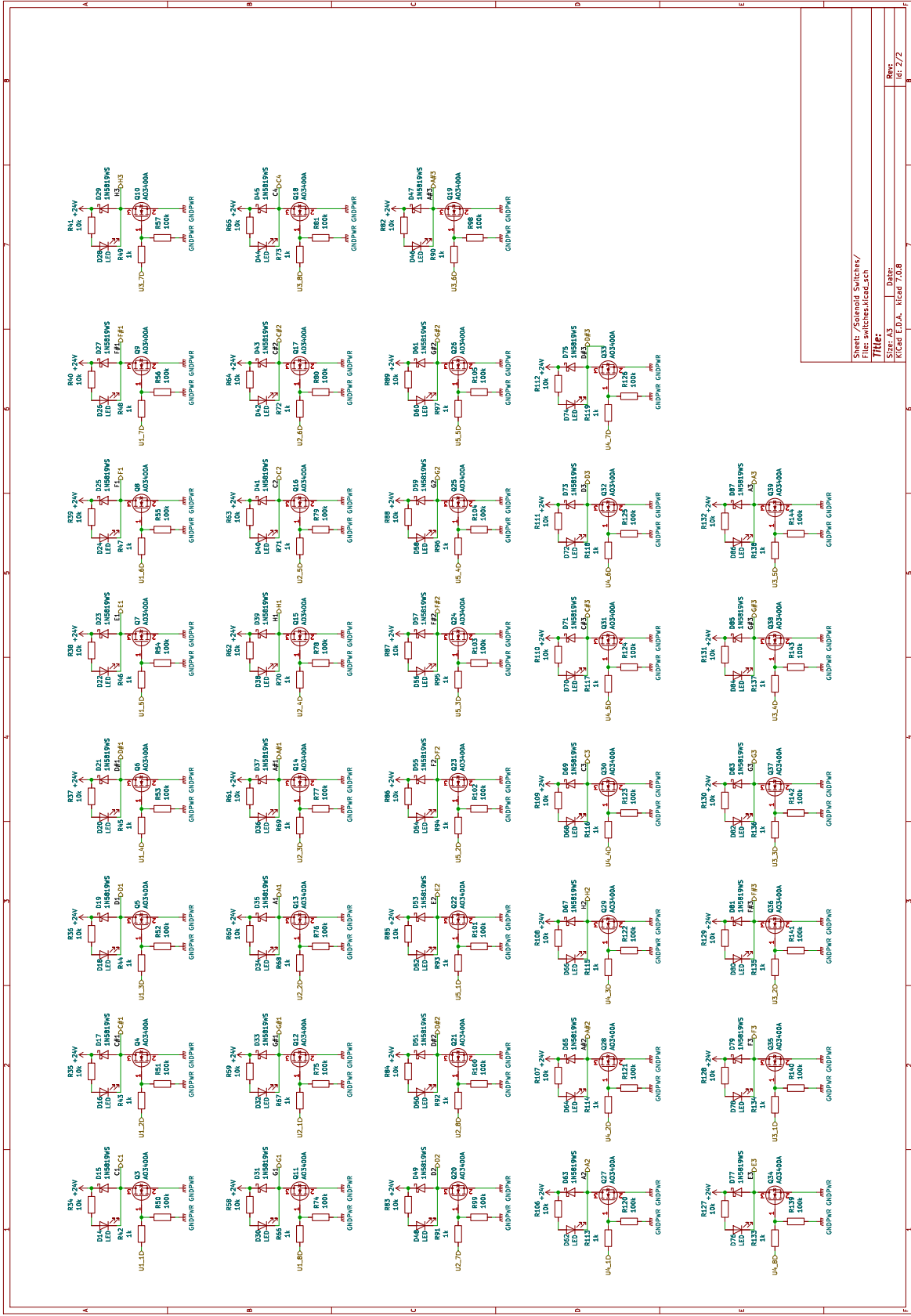
# Zoznam príloh

A Schéma zapojenia	69
B Doska plošných spojov	71
C Obsah elektronickej prílohy	73



# A Schéma zapojenia





Sheet: Switches Switches/  
File: switches.licad.sch

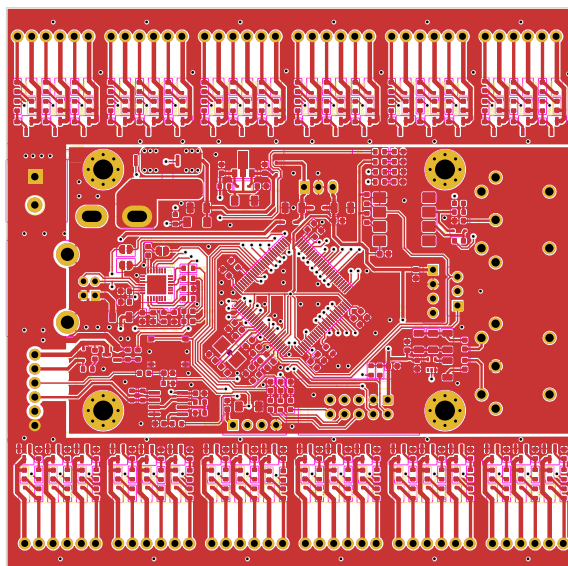
Title:

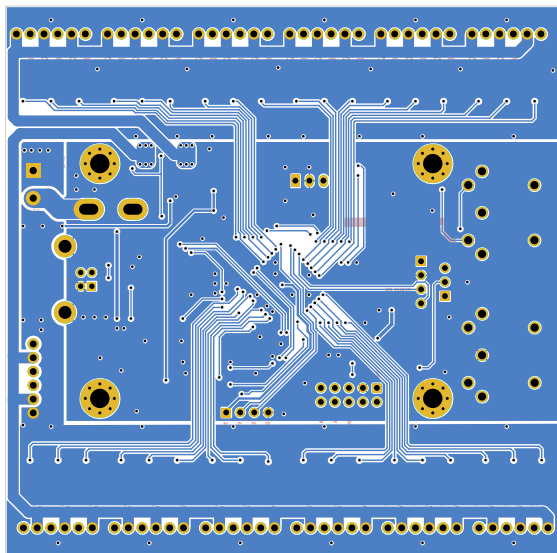
Size: A3  
Kicad E.D.A. licad 7.0.8

Rev: 2/2

Date:

## B Doska plošných spojov





## C Obsah elektronickej prílohy

/	
	— MidiSequencerDesign.zip ..... modely krabičiek, SolidWorks 2023
	— MidiSequencerHw.zip ..... schéma a dizajn pcb, KiCad7
	— MidiSequencerSw.zip ..... program, PlatformIO