

ZÁPADOČESKÁ UNIVERZITA V PLZNI

---

Fakulta elektrotechnická  
Katedra elektroniky a informačních technologií

## BAKALÁŘSKÁ PRÁCE

Inteligentní metronom

Autor práce:

**Martin Kněžourek**

Vedoucí práce:

**Ing. Petr Weissar, PhD.**

---

2023

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická  
Akademický rok: 2022/2023

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Martin KNĚŽOUREK**  
Osobní číslo: **E19B0094P**  
Studijní program: **B2612 Elektrotechnika a informatika**  
Téma práce: **Inteligentní metronom**  
Zadávací katedra: **Katedra elektroniky a informačních technologií**

## Zásady pro vypracování

1. Navrhněte a realizujte metronom využívající řízení mikrokontrolérem.
2. Umožněte generování rytmu podle skladeb uložených v zařízení.
3. Využijte vhodné uživatelské rozhraní.

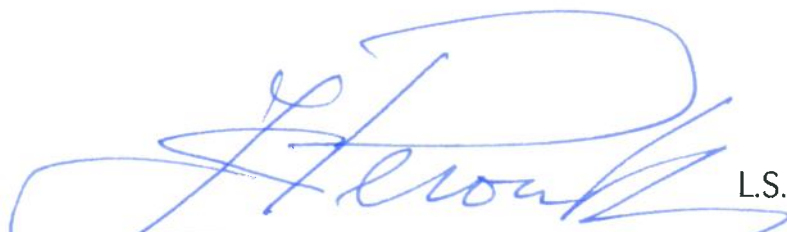
Rozsah bakalářské práce: **30 – 40**  
Rozsah grafických prací:  
Forma zpracování bakalářské práce: **elektronická**

Seznam doporučené literatury:

1. Yiu, Joseph. The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors, Elsevier Science & Technology, 2013.
2. Online dokumentace výrobců.

Vedoucí bakalářské práce: **Ing. Petr Weissar, Ph.D.**  
Katedra elektroniky a informačních technologií


Datum zadání bakalářské práce: **7. října 2022**  
Termín odevzdání bakalářské práce: **26. května 2023**



L.S.

---

**Prof. Ing. Zdeněk Peroutka, Ph.D.**  
děkan



---

**Doc. Ing. Jiří Hammerbauer, Ph.D.**  
vedoucí katedry

V Plzni dne 7. října 2022

# Abstrakt

Metronom je zařízení, které pomáhá hudebníkům, zejména bubeníkům, udržet stálé tempo skladby po celou dobu jejího trvání. Na trhu je k dostání celá řada metronomů v různých provedeních od mechanických po elektronické. Jejich funkce však většinou bývá prostá - pouze tikat v nastaveném rytmu. V žádném z dostupných metronomů není možnost předvolit si rychlosti skladeb tak, aby byla možnost pohotově nastavit a spustit metronom. Cílem této bakalářské práce bylo sestrojít metronom s možností uložení seznamu skladeb a jejich rychlostí. Pro nastavení metronomu uživatel tedy pouze zadá číslo skladby a metronom se sám nastaví. Hlavním požadavkem byla pokud možno co nejmenší cena zařízení. Z tohoto důvodu byl použit na dnešní dobu poněkud zastaralý, avšak cenově dostupný hardware, který svým výkonem plně dostačuje k sestrojení tohoto zařízení.

## Klíčová slova

Metronom, I2C, UART, HD44780, ATmega324A, uložení dat do paměti

# Abstract

A metronome is a device that helps musicians, especially drummers, keep a steady tempo throughout the duration of a song. A variety of metronomes are available on the market in different designs, from mechanical to electronic. However, their function is usually simple - just to tick according to a set rhythm. In none of the available metronomes is there an option to preset the tempos of the songs so that the metronome can be quickly set and triggered. The aim of this bachelor thesis was to build a metronome with the possibility to store a list of songs and their velocities. Thus, to set up the metronome, the user simply enters a song number and the metronome sets itself. The main requirement was to keep the cost of the device as low as possible. For this reason, somewhat outdated, but affordable hardware was used, which is fully sufficient to build this device.

## Keywords

Metronome, I2C, UART, HD44780, ATmega324A, storage of data in memory

## Poděkování

Tímto bych rád poděkoval vedoucímu práce Ing. Petru Weissarovi, PhD. za jeho cenné profesionální rady a připomínky, které mi pomohly k vypracování této práce.

# Obsah

Seznam použitých symbolů a zkratk	vi
Seznam obrázků	vii
Seznam tabulek	viii
Úvod	1
<b>1 Návrh hardwaru</b>	<b>2</b>
1.1 Blokové schéma . . . . .	2
1.2 Popis jednotlivých bloků . . . . .	3
1.2.1 Napájecí zdroj 5V . . . . .	3
1.2.2 Mikrokontrolér . . . . .	3
1.2.3 Displej . . . . .	4
1.2.4 USB/UART převodník . . . . .	4
1.2.5 EEPROM paměť . . . . .	5
1.2.6 Ostatní periferie . . . . .	5
1.3 Návrh plošného spoje . . . . .	6
<b>2 Návrh softwaru</b>	<b>7</b>
2.1 Popis způsobu SW ovládání jednotlivých komponent . . . . .	7
2.1.1 Obsluha displeje . . . . .	7
2.1.2 Obsluha EEPROM paměti . . . . .	9
2.1.3 Obsluha USB/UART převodníku . . . . .	12
2.1.4 Obsluha inkrementálního čidla . . . . .	14
2.1.5 Obsluha maticové klávesnice . . . . .	15
2.2 Popis uživatelského rozhraní . . . . .	16
2.2.1 Popis rozhraní metronomu . . . . .	16
2.2.2 Popis PC aplikace . . . . .	20
2.2.3 Popis způsobu uložení dat v paměti . . . . .	25
2.2.4 Popis architektury SW . . . . .	28
<b>3 Výroba</b>	<b>30</b>
3.1 Výroba plošných spojů . . . . .	30
<b>4 Závěr</b>	<b>32</b>
Seznam použité literatury	33

## Přílohy

**A**



# Seznam použitých symbolů a zkratek

<b>Značka</b>	<b>Popis</b>
<i>ASCII</i>	American Standard Code for Information Interchange
<i>Bd</i>	Baud
<i>BPM</i>	Beats Per Minute
<i>EEPROM</i>	Electrically Erasable Programmable Read-Only Memory
<i>HW</i>	Hardware
<i>I2C</i>	Inter-Integrated Circuit
<i>LSB</i>	Least Significant Bit
<i>MSB</i>	Most Significant Bit
<i>PC</i>	Personal Computer
<i>SCL</i>	Serial Clock
<i>SDA</i>	Serial Data Line
<i>SW</i>	Software
<i>TTL</i>	Transistor-Transistor-Logic
<i>UART</i>	Universal Asynchronous Receiver-Transmitter
<i>USB</i>	Universal Serial Bus

# Seznam obrázků

1	Blokové schéma metronomu. . . . .	2
2	Blokové schéma mikrokontroléru ATmega324A [2]. . . . .	4
3	Výsledná deska plošného spoje. . . . .	6
4	Vnitřní struktura zapojení pinů sběrnice [8]. . . . .	9
5	Průběhy na vodičích I2C sběrnice [8]. . . . .	10
6	Průběhy napětí na výstupech inkrementálního čidla. . . . .	14
7	Schéma zapojení maticové klávesnice. . . . .	15
8	Rozložení ovládacích prvků metronomu. . . . .	16
9	Hlavní obrazovka metronomu. . . . .	17
10	Režimy pro úpravu číselných hodnot. . . . .	18
11	Režim odeslání dat. . . . .	19
12	Režim příjmu dat. . . . .	19
13	Hlavní okno aplikace. . . . .	20
14	Soubor vytvořený aplikací. . . . .	21
15	Výsledek neúspěšné kontroly dat. . . . .	22
16	Pole znaků podporovaných displejem. . . . .	22
17	Okno výběru portu pro odeslání dat se signalizací stavu dokončeného přenosu. . . . .	23
18	Okno výběru portu pro příjem dat se signalizací stavu dokončeného přenosu. . . . .	24
19	Struktura dat v EEPROM paměti. . . . .	25
20	Formát jedné položky v paměti. . . . .	26
21	Výpis bloku paměti. . . . .	26
22	Fotografie výsledného hardwaru. . . . .	30
23	Leptání plošného spoje. . . . .	31

# Seznam tabulek

1	Tabulka vybraných instrukcí displeje a jejich adres [3]. . . . .	8
2	Konfigurace adresy zařízení [5]. . . . .	10
3	Vybrané registry pro obsluhu I2C sběrnice [2]. . . . .	11
4	Podaplikace a jejich funkce. . . . .	28
5	Seznam použitých knihoven. . . . .	29

# Úvod

Tato práce se zabývá výrobou metronomu s možností uložení seznamu skladeb a jejich rychlostí. Metronom s pamětí existuje pouze jako mobilní aplikace, která sice obsahuje vše potřebné, avšak použití mobilního telefonu během koncertu není příliš praktické, neboť viditelnost jeho displeje na přímém slunci je téměř nulová, zároveň mohou být rušivá i oznámení z ostatních aplikací. Z tohoto důvodu musí být náš metronom samostatně fungující zařízení s displejem, který je čitelný i na přímém slunci. Zároveň by měl mít dostatečně jednoduché a intuitivní uživatelské rozhraní, aby ho dokázal ovládat člověk bez jakéhokoliv zaškolení.

Pro realizaci metronomu byl proto zvolen alfanumerický LCD displej s velkými znaky. Metronom je řízen osmibitovým mikrokontrolérem z řady AVR. Skladby jsou uloženy ve vnější EEPROM paměti. Obsah paměti je možné vytvářet v PC aplikaci, data z této aplikace jsou přenášena do metronomu prostřednictvím UART sběrnice. Metronom musí samozřejmě plnit i základní funkce, kterými je akustická signalizace úderu a zobrazení čísla doby v taktu.

Zařízení vzniká na zakázku. Zadavatel vede kapelu a delší dobu sháněl výše popsané zařízení. Toto zařízení by mělo sloužit zastupujícím členům kapely, kteří se díky němu budou moci snaže zorientovat v seznamu skladeb kapely. Seznam skladeb bude vytvářen zpravidla ve zkušebně, v průběhu vystoupení by však mělo být možné upravit některé hodnoty. Z tohoto důvodu musí metronom umožňovat přímou změnu číselných hodnot. Naopak texty, uložené v zařízení, jako jsou například názvy skladeb a textové poznámky, není nutné upravovat přímo v metronomu. Mohou být upravovány pouze pomocí PC aplikace.

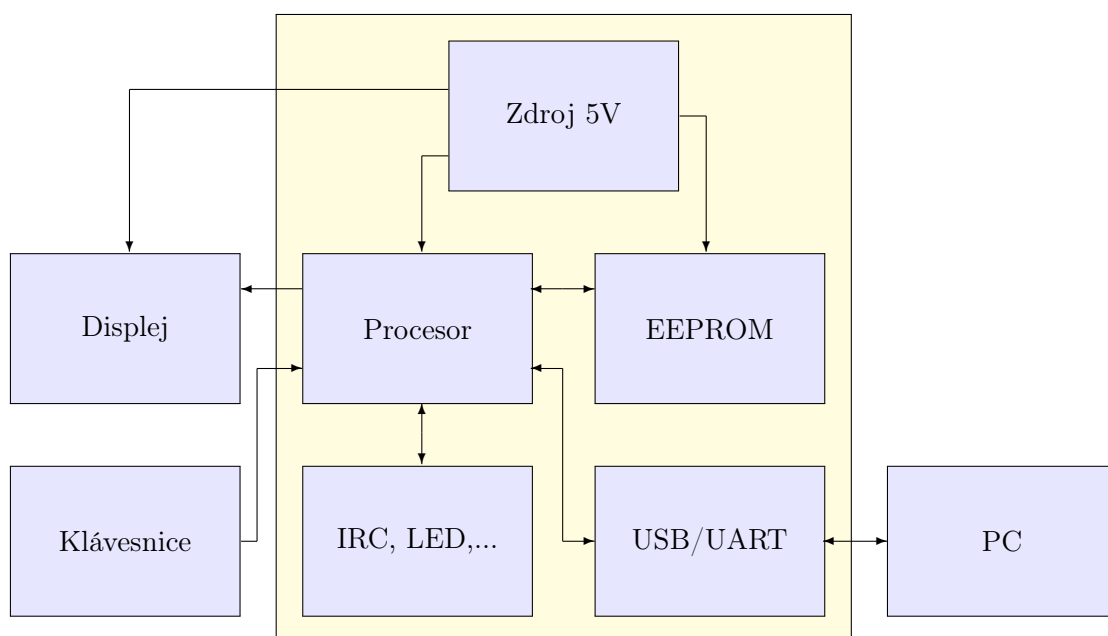
První kapitola této práce se zabývá návrhem hardwaru. Obsahuje popis jednotlivých použitých komponent a návrh plošného spoje. Druhá kapitola popisuje uživatelské rozhraní metronomu včetně jednotlivých režimů. Dále popisuje funkci PC aplikace, způsoby komunikace mezi aplikací a metronomem a způsob uložení dat v paměti. V závěru kapitoly je popsána architektura softwaru, jsou zde popsány jednotlivé podaplikace a použité knihovny. Poslední kapitola se zabývá samotnou výrobou metronomu. Je v ní popsán proces výroby plošného spoje fotocestou.

# 1 Návrh hardwaru

Tato kapitola se zabývá návrhem hardwaru metronomu. Jsou v ní popsány jednotlivé vybrané komponenty a jejich funkce. Závěr kapitoly se zabývá návrhem plošného spoje.

## 1.1 Blokové schéma

Hardware je navržen podle blokového schématu, které je uvedeno na obrázku 1. Hardware je rozdělen na tři oddělené moduly - základní desku (v obrázku 1 vyznačena žlutě), dále pak na displej a modul s klávesnicí. Displej je použit alfanumerický 2004 s výškou znaku 9,22 mm. Modul klávesnice obsahuje matici tlačítek v rozložení 3 řady a 6 sloupců. Z důvodu snazšího zabudování je klávesnice na separátním plošném spoji. Základní deska na sobě nese mikrokontrolér, který má na starosti veškeré řízení metronomu. Kromě mikrokontroléru je na desce ještě několik pomocných obvodů a periférií, např. převodník USB-UART, EEPROM paměť, inkrementální čidlo, signalizační LED diody a samozřejmě nechybí ani zdroj 5V pro napájení procesoru a USB-UART převodníku.



**Obrázek 1:** Blokové schéma metronomu.

## 1.2 Popis jednotlivých bloků

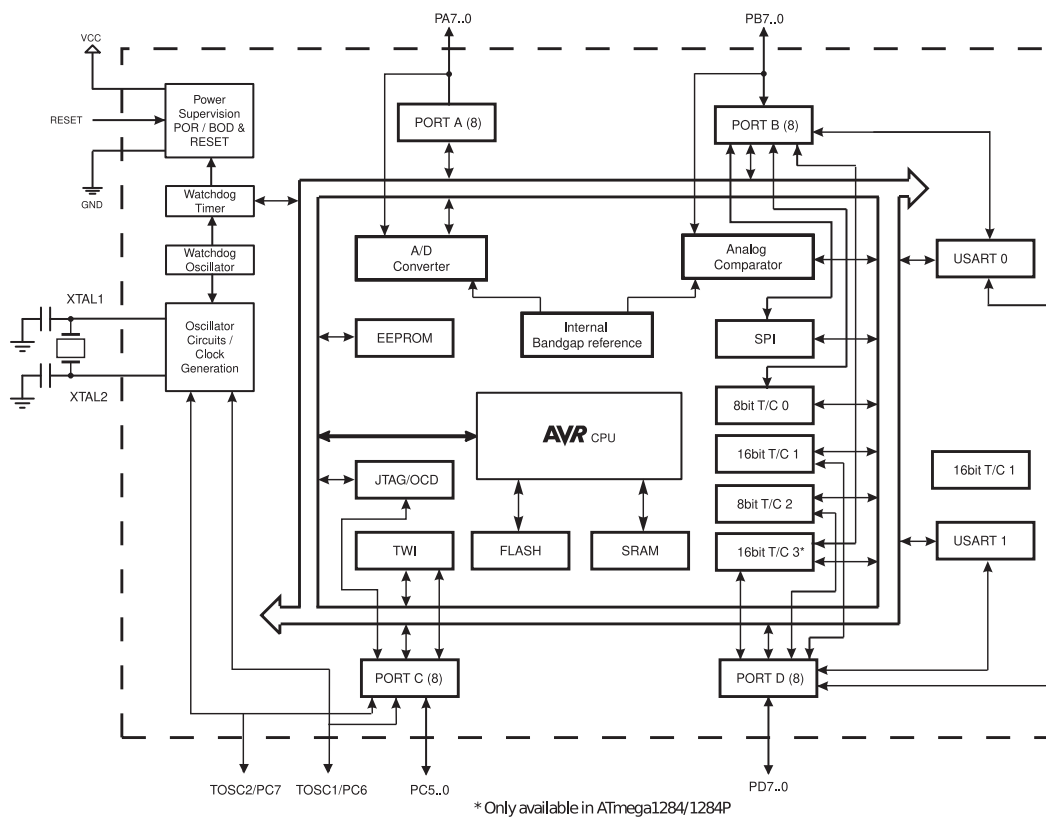
### 1.2.1 Napájecí zdroj 5V

Veškeré použité komponenty potřebují ke své práci napájení 5 V. Aby bylo zabráněno zničení obvodů při případném přepětí na vstupním napájení, je na vstupu zařazen lineární stabilizátor napětí 7805. Stabilizátor pracuje s rozsahem vstupního napětí od 7 V do 35 V.

Jelikož je stabilizátor lineární, ztrátový výkon na něm je přímo úměrný úbytku napětí a proudu tekoucího stabilizátorem. Z tohoto důvodu by hodnota vstupního napájecího napětí měla být co nejbližší hodnotě 7. Doporučená úroveň napájecího napětí je 7 až 12 V. [1]

### 1.2.2 Mikrokontrolér

Na výběr mikrokontroléru nebyly kladeny žádné speciální nároky. Vybraný mikrokontrolér musí mít možnost připojení I2C sběrnice a možnost komunikace pomocí UART. Zároveň musí mít dostatečný počet vstupně-výstupních pinů (vzhledem k většímu množství připojených periférií). Dále musí mít dostatečně velkou vnitřní flash paměť, aby do ní bylo možné zapsat středně rozsáhlý program. Pro tyto požadavky plně dostačuje mikrokontrolér typu ATmega324A z řady AVR. Blokové schéma mikrokontroléru je znázorněno na obrázku 2. Jedná se o 8-bitový procesor s RISC architekturou, 32 MB vnitřní flash paměť, 2 kB SRAM paměť, 1 kB vnitřní EEPROM paměť a čtyřmi vstupně-výstupními porty. Pro mikrokontrolér bylo zvoleno pouzdro typu DIL40, neboť v době vzniku HW nebyl tento procesor dostupný v jiném pouzdře. [2]



Obrázek 2: Blokové schéma mikrokontroléru ATmega324A [2].

### 1.2.3 Displej

Displej byl zvolen alfanumerický typu 2004 s řadičem HD44780. Displej má 4 řádky, na každém z řádků zobrazuje 20 znaků. Prodává se ve dvou velikostech, menší s výškou znaku 5,56 mm a větší s výškou znaku 9,22 mm. Pro tuto jednotku byl zvolen displej s výškou znaku 9,22 mm. Jeden znak zobrazovaný na displeji se skládá z matice 5x8 bodů. To, jaké znaky je možné zobrazit, popisuje ASCII tabulka v katalogovém listu. Prvních 8 pozic v ASCII tabulce displeje je určeno pro uživatelem definované znaky, ostatní znaky jsou napevno zapsány v paměti řadiče, není je tedy možné měnit. [3]

### 1.2.4 USB/UART převodník

Převodník USB/UART slouží pro komunikaci s počítačem, pomocí kterého se do metronomu zapisují seznamy skladeb. Jako převodník byl využit integrovaný obvod CH340G, který se používá např. u vývojových kitů Arduino. Pro správnou funkci převodníku je potřeba mezi piny 7 a 8 připojit krystal o frekvenci 12 MHz, včetně dvou zatěžovacích kondenzátorů proti zemi. K UART sběrnici mikrokontroléru je připojen piny 2 a 3. [2] [4]

### 1.2.5 EEPROM paměť

Paměť v metronomu slouží pro uchování seznamů skladeb. Je použita paměť 24C256 o velikosti 256 kbit, komunikující pomocí I2C sběrnice. Velikost paměti 256 kbit (32 kB) je pro toto zařízení plně dostačující, neboť dokáže pojmout data až 710 skladeb, bude-li průměrná délka textových řetězců 20 znaků. [5]

### 1.2.6 Ostatní periferie

K mikrokontroléru jsou kromě výše uvedených připojeny ještě následující periferie:

- **LED diody**, které ukazují uživateli aktuální dobu v taktu,
- **inkrementální čidlo**, které urychluje změnu číselných hodnot,
- **reproduktor**, který akustickým signálem udává rytmus,
- **maticová klávesnice**, umožňující volbu režimů, spouštění metronomu, případně zadání číselných hodnot.

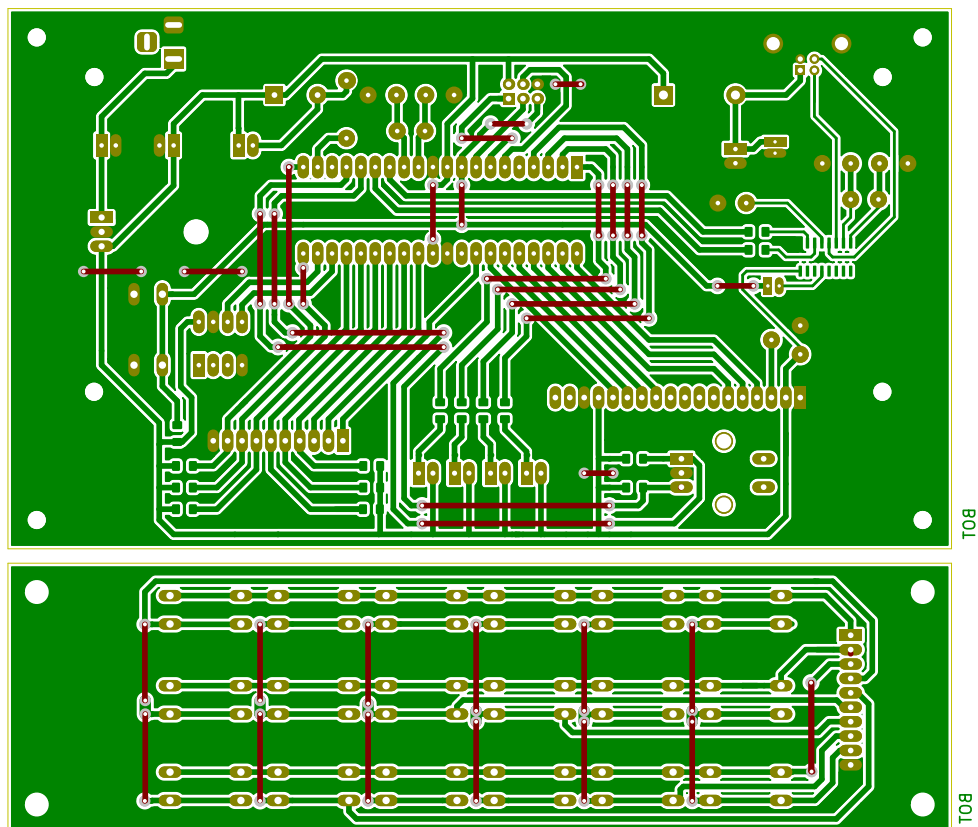


### 1.3 Návrh plošného spoje

Plošný spoj byl navržen pomocí prostředí KiCad, což je volně dostupné prostředí pro návrh desek plošných spojů od schéma přes desku plošného spoje až po 3D model výrobku. [6]

Zařízení je realizováno pomocí dvou desek plošných spojů. První z desek je deskou základní, která nese mikrokontrolér, displej a veškeré ostatní periferie. Na druhé desce je potom realizována maticová klávesnice se třemi řádky a šesti sloupci. Obě desky jsou propojeny desetižilovým plochým vodičem. Schéma zapojení je uvedeno v příloze A.

Celkově vznikly dvě verze plošných spojů. První verze byla oboustranná. V průběhu práce se však ukázalo, že výroba oboustranné desky plošného spoje je v domácích podmínkách poněkud komplikovaná. Hardware jednotky metronomu není natolik složitý, aby musel být na oboustranném plošném spoji. Díky velkým rozměrům displeje si můžeme dovést i rozměrnější základní desku. Vznikl tedy jednostranný plošný spoj, jehož horní vrstvu je možné nahradit drátovými propojkami.



Obrázek 3: Výsledná deska plošného spoje.

## 2 Návrh softwaru

Tato kapitola se zabývá návrhem softwaru pro metronom. Detailně popisuje ovládání jednotlivých použitých komponent a jednotlivé režimy metronomu. V další části je popsán návrh PC aplikace včetně způsobu uložení dat v paměti. Závěr kapitoly je věnován detailnějšímu popisu softwaru metronomu jako celku.

### 2.1 Popis způsobu SW ovládání jednotlivých komponent

Software pro metronom samotný je psaný v jazace C. Pro ovládání komponent byly většinou použity již vytvořené knihovny. Jedná se o knihovny pro ovládání alfanumerického displeje, I2C sběrnice a UART sběrnice. Ostatní použité knihovny byly napsány přímo pro tuto práci.

#### 2.1.1 Obsluha displeje

Alfanumerický displej s řadičem HD44780 komunikuje s mikrokontrolérem pomocí paralelní datové sběrnice. Sběrnice displeje je osmibitová, přičemž displej podporuje i režim, ve kterém jsou k mikrokontroléru fyzicky připojeny pouze čtyři datové piny. Ve čtyřbitovém režimu se do displeje přenáší nejprve horní polovina dat a s dalším "enable" impulzem pak dolní polovina dat. [3]

##### 2.1.1.1 Popis jednotlivých pinů displeje

Displej má celkem 16 využitých pinů, z nichž prvních 6 slouží pro napájení a řízení displeje, dalších 8 slouží pro paralelní přenos dat a zbylé dva slouží pro napájení podsvětlení displeje. [7]

- **Vss** - kostra napájení
- **Vdd** - +5V napájení displeje
- **V0** - nastavení jasu displeje
- **RS** - volba mezi zápisem do datového registru, nebo registru instrukcí ("1" znamená zápis do registru dat)
- **R/W** - volba mezi čtením nebo zápisem dat ("1" znamená čtení)
- **E** - enable bit, označující začátek zápisu nebo čtení
- **DB4 - DB7** - horní datové piny

- **DB0 - DB3** - dolní datové piny (pro čtyřbitový režim nezapojené)
- **LEDK** - katoda LED podsvícení displeje
- **LEDA** - anoda LED podsvícení displeje

### 2.1.1.2 Zápis dat do registru displeje

Jak bylo zmíněno v kapitole 2.1.1, zápis dat do displeje může probíhat ve dvou režimech, volba režimu se provádí odesláním příslušné řídicí instrukce v průběhu inicializace displeje. V osmibitovém režimu jsou všechna data odeslána v jedné fázi. Ve čtyřbitovém režimu je přenesena nejprve horní polovina dat, ve druhé fázi pak dolní polovina dat.[3]

Přenos dat je zahájen nastavením příznaku "E" do log. 1. Tento přenos je prováděn po celou dobu, po kterou je příznak "E" v log. 1. Minimální doba přenosu jedné sady dat by měla být větší než 41  $\mu$ s. Příznakem RS je možné zvolit mezi zápisem do registru dat a zápisem do registru instrukcí. Pomocí příznaku R/W je možné volit mezi čtením nebo zápisem do registrů displeje, přičemž čtením lze ověřit správnost zapsaných dat. Toto ověření dat se však příliš nepoužívá, vzhledem k relativně velké spolehlivosti přenosu. R/W pak bývá už při návrhu HW trvale nastaven do log. 0, tedy nastaven pouze pro zápis.[3]

### 2.1.1.3 Instrukce pro řízení displeje

Ovládání displeje se provádí pomocí sady osmibitových instrukcí, které jsou popsány v tabulce 1. Tyto instrukce se do registru displeje zapisují podobně jako data, avšak s tím rozdílem, že při zápisu instrukce musí být příznak RS nastaven do log. 0.[3]

Instrukce	DBx							
	7	6	5	4	3	2	1	0
Vymazání displeje	0	0	0	0	0	0	0	1
Návrat kurzoru do počátku	0	0	0	0	0	0	1	-
Zapnutí displeje	0	0	0	0	1	1	0	0
Zobrazení kurzoru	0	0	0	0	1	1	1	0
Nastavení 4-bitového režimu	0	0	1	0	0	-	-	-
Nastavení 8-bitového režimu	0	0	1	1	0	-	-	-

**Tabulka 1:** Tabulka vybraných instrukcí displeje a jejich adres [3].

## 2.1.2 Obsluha EEPROM paměti

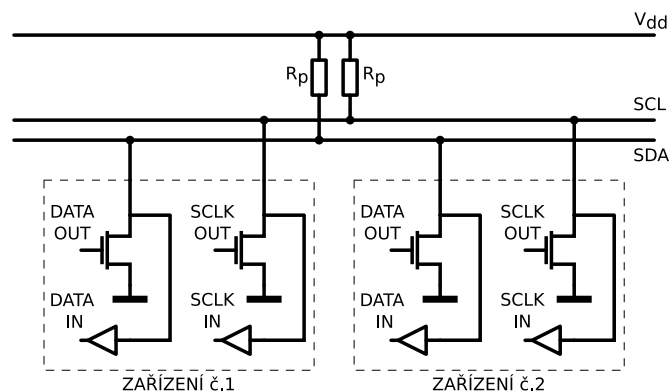
Paměť 24C256 je 256 kbit EEPROM paměť, ovládaná pomocí I2C sběrnice. Paměť má možnost volby tří posledních bitů její adresy na sběrnici tak, aby bylo možné na jednu sběrnici procesoru zapojit více pamětí a zvětšit tak celkovou kapacitu úložiště dat. [5]

### 2.1.2.1 I2C sběrnice

I2C sběrnice je synchronní datová sběrnice, která se využívá pro komunikaci mezi dvěma a více zařízeními na krátké vzdálenosti. Typická je zejména pro EEPROM paměti, displeje, nebo digitální čidla teploty. [8]

Sběrnice využívá dva vodiče. První z vodičů je vodič SCL, který nese hodinový signál. Druhý vodič SDA nese samotná data. V době, kdy neprobíhá žádný přenos dat, jsou oba vodiče sběrnice ve stavu vysoké impedance. Logická hodnota je na vodič přivedena pouze jeho připojením k zemi. Z tohoto důvodu musí být oba vodiče sběrnice připojeny pomocí pullup rezistorů, ke kladnému pólu napájení, jak je znázorněno na obrázku 4. Funkcí pullup rezistorů je přivést na vodiče sběrnice hodnotu log. 1 ve chvíli, kdy jsou ve stavu vysoké impedance. [8]

Přenos začíná vždy nastavením datového vodiče do log. 0 a následným spuštěním hodinového signálu. Poté jsou data přenášena vždy v blocích po osmi bitech. Přijetí bloku dat je vždy potvrzeno bitem ACK, který vyše zařízení přijímající data. Maximální frekvence hodinového signálu je specifikována pro každé zařízení, v mikrokontroléru se obvykle specifikuje v průběhu inicializace. [8]



Obrázek 4: Vnitřní struktura zapojení pinů sběrnice [8].

Zpráva se vždy skládá ze dvou částí. V první části zprávy je iniciátorem přenosu vyslána sedmibitová adresa zařízení, se kterým se pokouší komunikovat. Osmým bitem je pak sděleno, zda se bude v tomto přenosu číst či zapisovat (log. 0 znamená zápis). V následujících blocích jsou přenášena samotná data. Typicky bývají první datové bloky vyhrazené pro specifikaci adresy v paměti zařízení, ze které se má číst, nebo zapisovat. [8]



frekvencí signálu SCL, přenosovou rychlostí a prescalerem udává následující vzorec:

$$f_{SCL} = \frac{f_{CPU}}{16 + 2 \cdot TWBR \cdot 4^{TWPS}}, \quad (1)$$

kde  $f_{SCL}$  je frekvence signálu SCL,  $f_{CPU}$  je frekvence hodinového signálu mikrokontroléru (Hz),  $TWPS$  je hodnota bitů prescaleru z registru TWSR a  $TWBR$  je hodnota registru TWBR. Vyjádříme-li z výše uvedené rovnice  $TWBR$ , získáme následující vztah:

$$TWBR = \frac{f_{CPU}}{2 \cdot f_{SCL} \cdot 4^{TWPS}} - 8, \quad (2)$$

Úkolem prescaleru je dostatečně uměle snížit frekvenci hodinového signálu mikrokontroléru tak, aby bylo možné výsledný poměr mezi frekvencí hodinového signálu mikrokontroléru a frekvencí signálu SCL popsat osmibitovým číslem. Toto snížení je samozřejmě pouze pro potřeby výpočtu. Využití prescaleru má význam pouze pro velmi malé přenosové rychlosti, většinou tedy bývá nastaven na 0.[2]

Zahájení každého přenosu se provádí zápisem do registru TWCR. Nejprve je nutné povolit I2C sběrnici nastavením bitu TWEN a nastavit bit TWSTA pro odeslání startovací podmínky. Poté je potřeba nastavit bit TWINT a počkat, dokud nebude vnitřně nastaven do log. 1. Ve chvíli, kdy je bit TWINT nastaven do log. 1, je odeslána startovací podmínka a je možné odeslat adresu zařízení. Adresa se zapíše do registru TWDR, bit TWINT z registru TWCR opět signalizuje dokončení přenosu. Před i po odeslání adresy je vhodné zjistit stav sběrnice. Stav sběrnice je možné přečíst z horních pěti bitů registru TWST. Jednotlivé stavy jsou popsány v katalogovém listu mikrokontroléru.[2]

Po odeslání adresy už je možné odeslat samotná data. Princip je obdobný, tedy zapsat data do registru TWDR, pokusit se nastavit bit TWINT a počkat na jeho skutečné nastavení. V případě čtení ze sběrnice opět nastavíme bit TWINT a TWEA z registru TWCR. Bitem TWEA sdělujeme, že chceme po přijmutí dat odeslat potvrzení ACK, jak je znázorněno na obrázku 5. Ve chvíli, kdy je bit TWINT nastaven, jsou v registru TWDR dostupná přijatá data. [2]

Registr	Popis bitů registru								
	Bit	7	6	5	4	3	2	1	0
TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE	
TWST	TWS7	TWS6	TWS5	TWS4	TWS3		TWPS1	TWPS0	

**Tabulka 3:** Vybrané registry pro obsluhu I2C sběrnice [2].

### 2.1.3 Obsluha USB/UART převodníku

Samotný převodník USB/UART CH340 narozdíl od paměti není nutné nijak softwarově nastavovat, stačí jej pouze správně zapojit. Má-li převodník komunikovat s počítačem, potom je nutné nainstalovat ovladač pro CH340. Pro platformu Windows je možné nainstalovat ovladač přímo z \*.exe souboru. Na platformě Linux není nutné nic instalovat, stačí pouze přidat dané zařízení do skupiny "dialout". Zařízení v této skupině mají plný přístup k sériovému portu.

#### 2.1.3.1 UART sběrnice

Sběrnice UART je dvou vodičová sběrnice, která může pracovat jak v synchronním, tak i v asynchronním režimu. Její výhodou je fakt, že v jeden okamžik může probíhat jak zápis, tak i čtení. Sběrnice má totiž pro každý směr toku dat vyhrazen jeden vodič. Pro vyjádření log. 0 a log. 1 používá sběrnice úroveň dané TTL logikou, což znamená, že log. 0 je rovna 0 V, log. 1 je rovna 5 V, případně 3,3 V. Zvýšením těchto úrovní je však možné přeměnit UART sběrnici na RS232 či RS483, což umožní provozovat sběrnici i na delší vzdálenosti. [9] [2]

Datové vodiče mají v době, kdy neprobíhá přenos, hodnotu log. 1. Pro zahájení přenosu je odeslán jeden bit log. 0. V tuto chvíli je na straně příjemce vyvoláno přerušení od sestupné hrany a začíná se obsluhovat příjem dat. Po přijetí "start bitu" následuje příjem osmi datových bitů a jednoho paritního. Paritní bit vyjadřuje celkový součet všech bitů odeslaného bytu, majících hodnotu log. 1. Pomocí tohoto bitu je tedy možné zkontrolovat, zda nedošlo k poškození dat v průběhu přenosu. Přenos je ukončen odesláním jednoho až dvou bitů log. 1. Na rozdíl od I2C sběrnice musí být přenos ukončen po každém bytu. [9] [2]

Přenosová rychlost, udávaná jako "baudrate", musí být předem nastavena na každém ze zařízení. Standardní velikost přenosové rychlosti je 9600, požadujeme-li rychlejší přenos, potom je možné vynásobit základní rychlost číslem  $2^n$ , kde  $n$  je celé číslo. Vyšší přenosová rychlost však musí být podporována oběma zařízeními. [9] [2]

#### 2.1.3.2 UART sběrnice z pohledu mikrokontroléru

Ovládání UART sběrnice pomocí mikrokontroléru probíhá opět s využitím hardwarové jednotky mikrokontroléru, která je ovládaná zápisem do jejích řídicích registrů. Hlavním rozdílem oproti I2C je fakt, že pozice master a slave nejsou nijak odlišeny, tzn. master nežadá slave o data. Data jsou posílána nezávisle na tom, zda jsou právě požadována. Z tohoto důvodu musí být vytvořen kruhový buffer, který shromažďuje přijatá data. Hlavní program potom v případě potřeby pouze "odebírá" data z bufferu. Obdobný buffer se používá i pro odeslání dat, kdy by odeslání dat byte po byte příliš zpomalovalo chod hlavního programu. Proto se data připravená k odeslání zkopírují do kruhového bufferu, následně jsou odesílána "na pozadí" ve chvíli, kdy je právě volná sběrnice. [9] [2]

V průběhu inicializace je nastavena přenosová rychlost a přerušení od přijetí kompletní zprávy.

Zápis hodnoty přenosové rychlosti se provádí zápisem do registrů UBRR0H (horní polovina hodnoty) a UBRR0L (dolní polovina hodnoty). Výpočet hodnoty pro zápis do těchto registrů je následující:

$$UBRR = \frac{f_{CPU}}{16 \cdot baudrate} - 1, \quad (3)$$

kde  $f_{CPU}$  je frekvence hodinového signálu mikrokontroléru (Hz),  $UBRR$  je vypočtená hodnota pro zápis do registrů UBRR a  $baudrate$  je požadovaná přenosová rychlost.

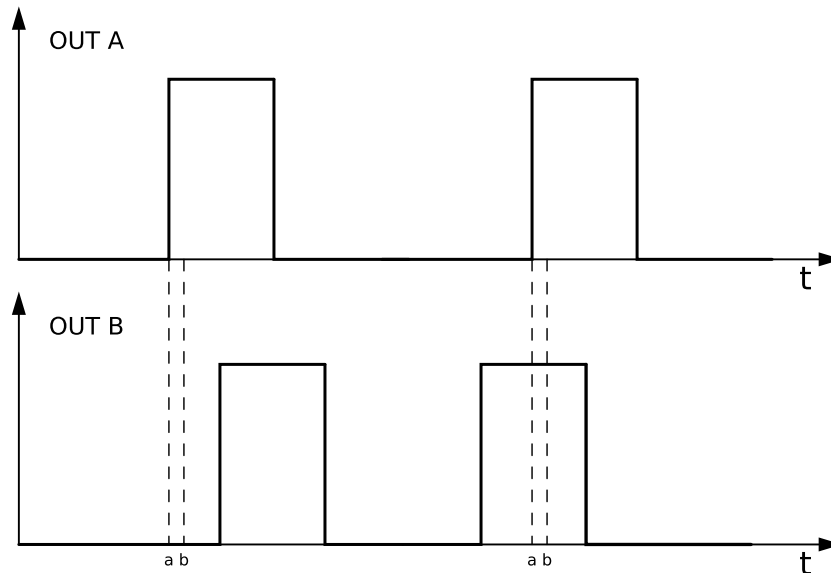
Dále je potřeba povolit přerušení od kompletní přijaté zprávy (zápisem log. 1 do bitu RXCIE registru UCSRnB) a povolit samotné piny sběrnice, zápisem log. 1 do bitů RXEN a TXEN registru UCSRnB. [2]

Jak bylo již zmíněno výše, příjem dat je obsluhován přerušením. V podprogramu tohoto přerušení jsou kopírována data z registru UDRn do kruhového bufferu a je kontrolován stav sběrnice v registru UCSRnA. Odesílání dat probíhá obdobně, po vyvolání přerušení se do registru UDRn zapíše data ze začátku bufferu. [2]



### 2.1.4 Obsluha inkrementálního čidla

Mechanické inkrementální čidlo se z pohledu mikrokontroléru chová prakticky jako dva spínače. Na jeho výstupu se po otočení hřídelí objeví dva impulzy, které jsou v závislosti na směru otočení fázově posunuty o  $90^\circ$ , případně  $-90^\circ$ , což je vidět na obrázku 6. Velikost impulzů je daná napětím, které je připojeno na společný pin čidla. [2]

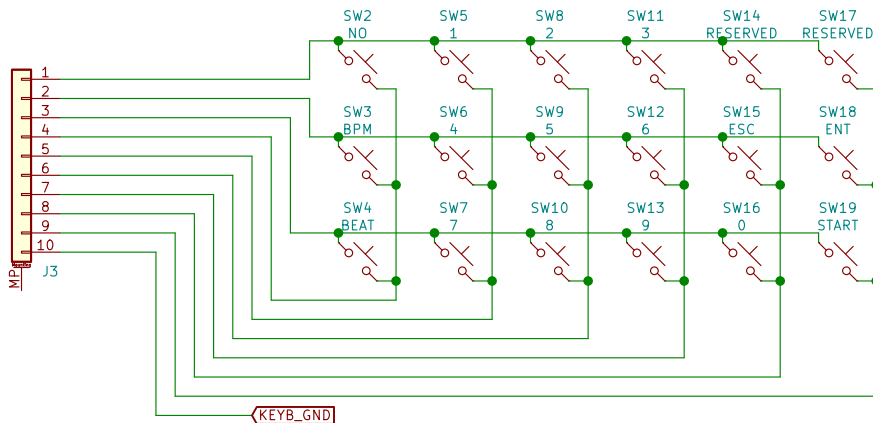


**Obrázek 6:** Průběhy napětí na výstupech inkrementálního čidla.

Jeden z výstupů čidla je připojen na pin mikrokontroléru, který umožňuje vyvolání vnějšího přerušení. Otočením inkrementálního čidla se na tomto výstupu objeví impulz, který vyvolá přerušení. Na obrázku 6 je tento okamžik znázorněn jako čas "a". V čase "b" je potom přečtena logická hodnota na druhém vstupu. Je-li tato hodnota rovna log. 1, potom bylo inkrementální čidlo otočeno proti směru hodinových ručiček. Je-li na tomto výstupu v čase "b" log. 0, potom bylo čidlo otočeno po směru hodinových ručiček. [2]

### 2.1.5 Obsluha maticové klávesnice

Kvůli úspoře pinů mikrokontroléru se využívá zapojení tlačítek klávesnice do tzv. matice. Schéma zapojení klávesnice je znázorněno na obrázku 7. Klávesnice má 3 řady a 6 sloupců, všechna tlačítka v dané řadě, případně sloupci jsou vzájemně propojena. Takto propojené řádky a sloupce jsou přivedeny do mikrokontroléru.



Obrázek 7: Schéma zapojení maticové klávesnice.

Čtení logické hodnoty z této klávesnice probíhá ve třech fázích. Ve výchozím stavu je na všech pinech klávesnice log. 1. V každé z fází je nejprve přiveden na jeden z řádků stav log. 0. Po uplynutí 1 ms je pak odečtena logická hodnota ze všech sloupců. Následně se na právě přečtený řádek přivede zpět log. 1 a na následující řádek se přivede log. 0. Tento postup se opakuje pro všechny řádky klávesnice.

Logickou hodnotu přečtenou z tlačítek je ještě potřeba tzv. ošetřit na zákmit. Toto ošetření se provádí z důvodu možného rušení z okolí, kdy by vlivem rušení mohlo dojít k nechtěnému sepnutí tlačítka. Zároveň může dojít ke stavu, kdy po stisknutí tlačítka uživatelem tlačítko "zakmitá", tzn. na vstupu mikrokontroléru se před sepnutím tlačítka objeví sekvence velmi krátkých impulzů. Každý z těchto impulzů by mikrokontrolér vyhodnotil jako jeden stisk tlačítka. Impulzy, jejichž původem je rušení, se vyznačují velmi malou šířkou (řádově maximálně jednotky ms). Impulzy, vytvořené stisknutím tlačítka, tedy ty které mají být zaznamenány, zpravidla nebývají kratší než 500 ms. Po první zaznamenané změně logické hodnoty na vstupu je proto potřeba počkat po dobu 200 ms. Po uplynutí této doby je znovu odečtena logická hodnota na vstupu. Jsou-li tyto dvě logické hodnoty shodné, je předána nadřazenému kódu informace, že bylo stisknuto tlačítko. V opačném případě je tento stisk tlačítka ignorován.

Kromě ošetření zákmitu je potřeba ještě ošetřit stavy kdy je v jeden okamžik stisknuto více tlačítek. Tyto stavy jsou nežádoucí, v případě stisknutí více tlačítek by mohlo v nadřazeném softwaru dojít k nepředvídatelným událostem, které by mohly skončit "zamrznutím" celého programu. Z tohoto důvodu je po stisknutí některého z tlačítek prováděn kontrolní součet. V případě, že tento součet vyhodnotí, že bylo stisknuto více tlačítek, je tento stisknutí ignorováno.

## 2.2 Popis uživatelského rozhraní

Uživatelské rozhraní metronomu se skládá ze dvou hlavních částí. První částí je rozhraní zabudované přímo v samotném metronomu. Druhou částí je pak PC aplikace, pomocí které uživatel vytváří seznamy skladeb a pomocí USB je ukládá do paměti metronomu.

### 2.2.1 Popis rozhraní metronomu

Jak již bylo zmíněno v kapitole 1, rozhraní metronomu se skládá ze čtyřřádkového alfanumerického displeje, klávesnice s 16 tlačítky, inkrementálního čidla, čtveřice signalizačních LED diod a reproduktoru. Rozložení ovládacích prvků je znázorněno na obrázku 8.

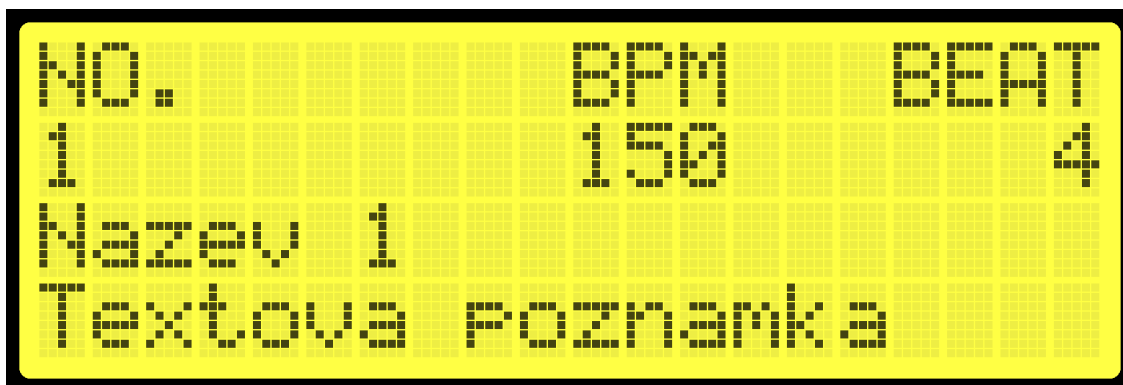


Obrázek 8: Rozložení ovládacích prvků metronomu.

Funkce metronomu je rozdělena do několika režimů, pro každý z režimů je specifické určité rozložení textu na displeji. Tyto režimy je možné měnit pomocí tlačítek "NO", "BPM" a "BEAT", případně pomocí inkrementálního čidla. Některé režimy však mohou být vyvolány pouze externě přes USB.

### 2.2.1.1 Hlavní obrazovka

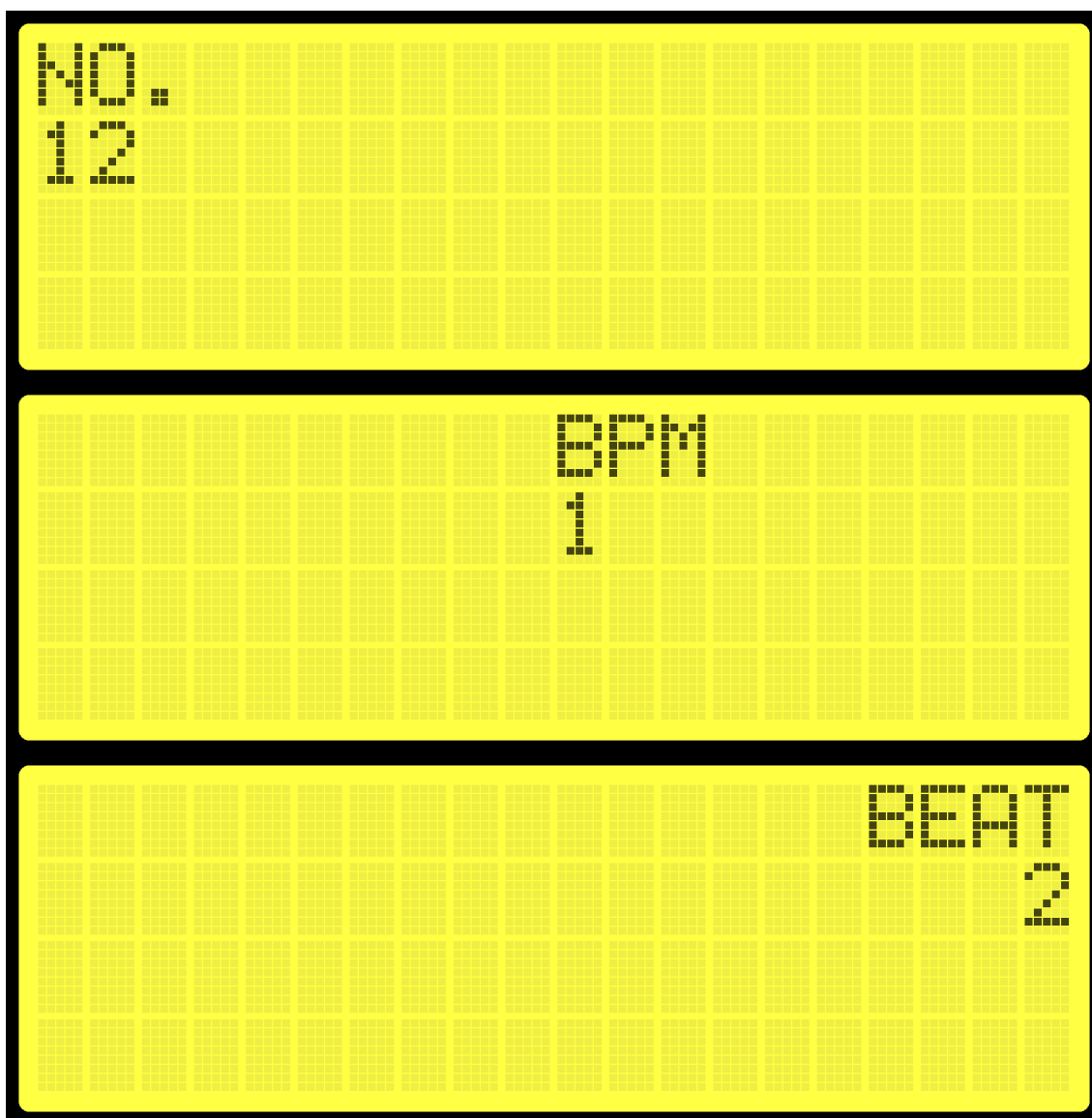
Na hlavní obrazovce jsou zobrazeny veškeré informace o skladbě právě načtené z paměti, tedy číslo skladby "NO", počet úderů za minutu "BPM", počet dob v taktu "BEAT", název skladby a textová poznámka. Rozložení textů na displeji je znázorněno na obrázku 9. Režim hlavní obrazovka je výchozím režimem a je zobrazen vždy po připojení metronomu k napájení, po návratu z jiného režimu a v době, kdy metronom "tiká".



Obrázek 9: Hlavní obrazovka metronomu.

### 2.2.1.2 Režim úpravy číselných hodnot

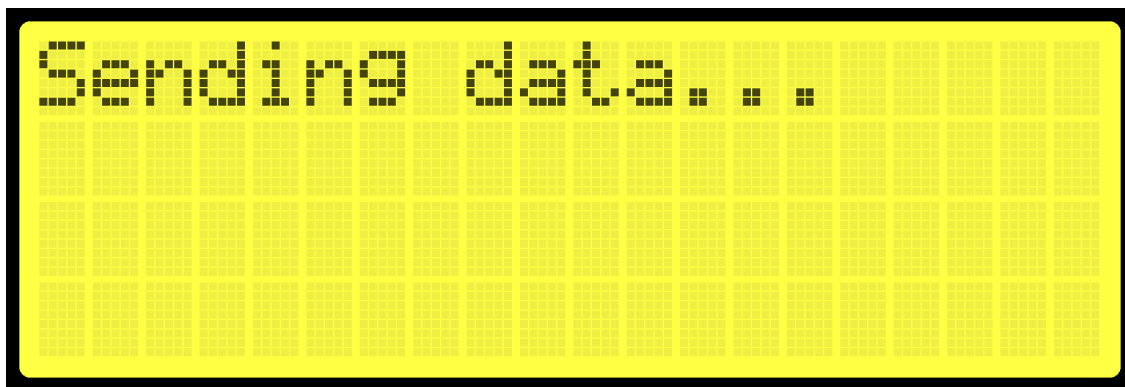
V režimu úpravy číselných hodnot je možné změnit číslo skladby, případně upravit hodnoty počtu úderů za minutu a počtu dob v taktu dané skladby. Na displeji je v tomto režimu zobrazena pouze upravovaná hodnota a její název, jak je možné vidět na obrázku 10. Zadání hodnoty se provádí pomocí numerické klávesnice, případně otáčením inkrementálního čidla. Tlačítko "ESC" slouží jako "backspace", tedy pro vymazání posledního zadaného znaku. Po stisknutí tlačítka "OK" je nejprve zkontrolována platnost zadané hodnoty (zda je zadaná hodnota v povoleném rozsahu). Je-li zadaná hodnota platná, uloží se do paměti. V opačném případě je v paměti ponechána původní hodnota. Zkontrolovat a uložit hodnotu do paměti je možné rovněž stisknutím tlačítka "START", ovšem pouze s tím rozdílem, že metronom začne okamžitě tikat. V případě, že uživatel během zadávání jedné hodnoty zvolí jiný režim, zadaná hodnota nebude uložena.



Obrázek 10: Režimy pro úpravu číselných hodnot.

### 2.2.1.3 Režim odeslání dat

Režim odeslání dat je vždy iniciován PC aplikací. V tomto režimu je přenesen veškerý obsah paměti pomocí sériového kanálu do aplikace. V aplikaci jsou potom data uspořádána do tabulky tak, aby je uživatel mohl jednoduše editovat. O skutečnosti, že právě probíhá přenos dat je uživatel metronomu informován vypsáním hlášky "Sending data..." na displej, jak je znázorněno na obrázku 11. Po ukončení přenosu dat se metronom opět vrátí do režimu "Hlavní obrazovka". Přenos dat je detailněji popsán v kapitole 2.2.2.



Obrázek 11: Režim odeslání dat.

#### 2.2.1.4 Režim příjmu dat

Režim příjmu dat je opět iniciován pouze PC aplikací. V tomto režimu jsou přenášena data z PC aplikace přímo do paměti a následně je vytvořen blok hlaviček, což je vlastně sada ukazatelů do paměti, z nichž každý ukazuje na první byte jedné skladby. Detailnější popis struktury paměti je v kapitole 2.2.3. O stavu přenosu dat je uživatel metronomu informován vypsáním hlášky "Receiving data..." a následně "Creating headers..." na displej, jak je znázorněno na obrázku 12. Po ukončení přenosu dat se metronom opět vrátí do režimu "Hlavní obrazovka".

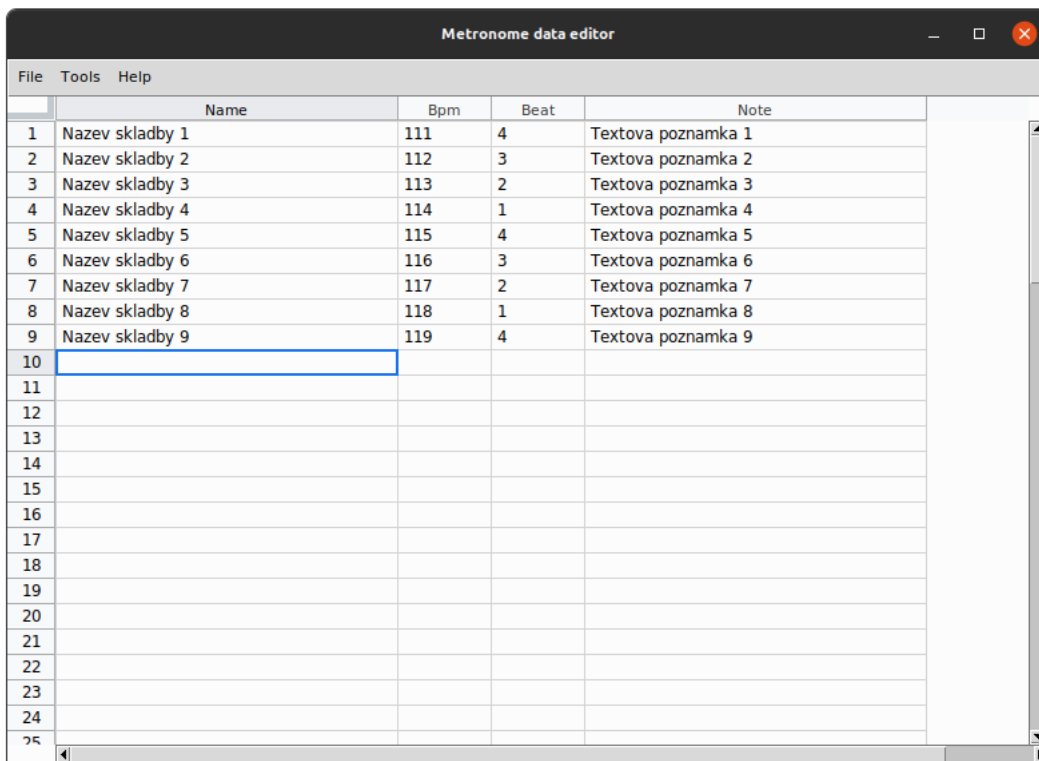


Obrázek 12: Režim příjmu dat.

## 2.2.2 Popis PC aplikace

Aplikace slouží k vytváření seznamů skladeb, které jsou následně pomocí sériového kanálu přenášeny do zařízení. Zároveň má aplikace za úkol zkontrolovat správnost uživatelem zadaných dat, kontroluje především existenci znaků, které nejsou podporovány displejem, dovolený rozsah číselných hodnot a maximální délku textových řetězců. Doplnkovou funkcí je možnost uložení seznamu do \*.csv souboru, případně načtení seznamu z \*.csv souboru.

Aplikace je vytvářena v jazyce Python pomocí vývojového prostředí PyCharm. Pro vytvoření grafického rozhraní byla využita knihovna tkinter, pro komunikaci se sériovým kanálem byla využita knihovna pyserial. Hlavní okno aplikace je zobrazeno na obrázku 13. Tabulkový editor tksheet není součástí základní knihovny tkinter, bylo nutné jej importovat samostatně.



	Name	Bpm	Beat	Note
1	Nazev skladby 1	111	4	Textova poznamka 1
2	Nazev skladby 2	112	3	Textova poznamka 2
3	Nazev skladby 3	113	2	Textova poznamka 3
4	Nazev skladby 4	114	1	Textova poznamka 4
5	Nazev skladby 5	115	4	Textova poznamka 5
6	Nazev skladby 6	116	3	Textova poznamka 6
7	Nazev skladby 7	117	2	Textova poznamka 7
8	Nazev skladby 8	118	1	Textova poznamka 8
9	Nazev skladby 9	119	4	Textova poznamka 9
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				

Obrázek 13: Hlavní okno aplikace.

### 2.2.2.1 Načtení dat ze souboru

Aby bylo možné jednoduše měnit celé seznamy skladeb v metronomu, pracuje aplikace s \*.csv soubory. V počítači je pak možné mít uloženo několik seznamů a pomocí aplikace nahrát do metronomu vždy právě ten, který je zrovna potřeba. Tyto soubory je možné vytvářet buď přímo v aplikaci, případně i v nějakém externím tabulkovém editoru. Je-li ovšem v aplikaci načten nějaký \*.csv soubor, jehož počet sloupců neodpovídá šabloně, tabulkový editor v aplikaci jej přesto načte, dokonce je možné jej i upravovat a znovu uložit. Kontrola obsahu probíhá pouze při pokusu odeslat data do metronomu.

Pro práci s \*.csv soubory je využito knihovny CSVParser, která je běžně dostupná ve správci balíčků pip. Pro práci se soubory byl využit modul filedialog z balíčku tkinter. Aplikace čte a ukládá \*.csv soubory výhradně s oddělovačem typu "středník". Přestože tento oddělovač není pro \*.csv soubory standardní, ve většině zemí se již stal jakýmsi standardem, neboť původní oddělovač "čárka" by mohl být zaměněn za desetinnou čárku, což by mohlo značně poničit strukturu \*.csv souboru. Zároveň většina tabulkových editorů, které jsou nastaveny do českého jazyka, mají pro \*.csv soubory "středník" jako výchozí oddělovač.

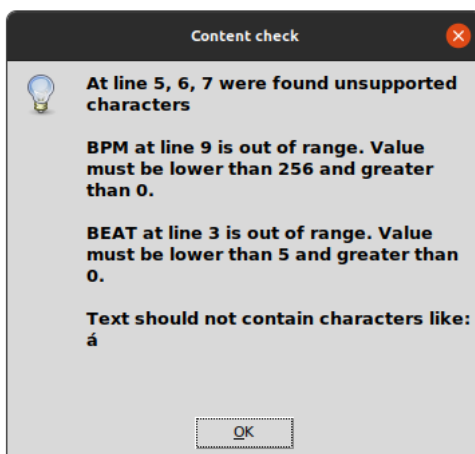
```
1 |Nazev skladby 1;111;4;Textova poznamka 1
2 |Nazev skladby 2;112;3;Textova poznamka 2
3 |Nazev skladby 3;113;2;Textova poznamka 3
4 |Nazev skladby 4;114;1;Textova poznamka 4
5 |Nazev skladby 5;115;4;Textova poznamka 5
6 |Nazev skladby 6;116;3;Textova poznamka 6
7 |Nazev skladby 7;117;2;Textova poznamka 7
8 |Nazev skladby 8;118;1;Textova poznamka 8
9 |Nazev skladby 9;119;4;Textova poznamka 9
```

Obrázek 14: Soubor vytvořený aplikací.



### 2.2.2.2 Kontrola správnosti zadaných dat

O kontrolu správnosti zadaných dat se stará funkce `check_data()` z třídy `Validator`. Funkce je volaná vždy před zavoláním funkce, která odesílá data do zařízení. Vstupem pro funkci `check_data()` je seznam polí, jehož každá položka obsahuje jeden řádek tabulky dat. Podle výsledku kontroly potom funkce vrací logickou hodnotu nesoucí informaci o výsledku kontroly. V případě, že kontrola proběhla s negativním výsledkem, funkce zobrazí okno s nalezenými chybami. Toto okno je znázorněno na obrázku 15. Uživateli není umožněno odeslání dat do metronomu, dokud neopraví nalezené chyby.



Obrázek 15: Výsledek neúspěšné kontroly dat.

Podmínky pro úspěšně provedenou kontrolu dat jsou následující:

1. texty nesmí obsahovat nepovolené znaky (seznam povolených znaků je uveden na obrázku 16),
2. číselné hodnoty nesmí obsahovat žádné znaky kromě číslic,
3. zadané hodnoty počtu úderů za minutu musí být větší než 0 a menší než 253,
4. zadané hodnoty počtu dob v taktu musí být větší než 0 a menší nebo rovny 4.

```
# characters supported by device's display
self.supported_chars = \
    ['!', '\\', '#', '$', '%', '&', '\\', '(', ')', '*', '+', ',', '-', '.', '/', '{', '|', '}', ' ',
    '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', ':', ';', '<', '=', '>', '?', '&',
    'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T',
    'U', 'V', 'W', 'X', 'Y', 'Z',
    'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't',
    'u', 'v', 'w', 'x', 'y', 'z', ]
```

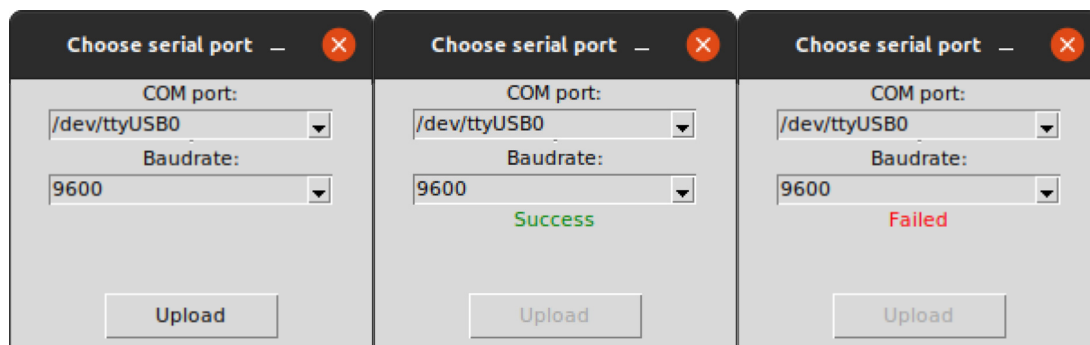
Obrázek 16: Pole znaků podporovaných displejem.

### 2.2.2.3 Odesílání dat do zařízení

Jak již bylo zmíněno v kapitole 1.2.4, odeslání dat do zařízení probíhá pomocí USB/UART převodníku, připojeného k počítači pomocí USB. Přenosová rychlost je pevně nastavena na 9600 Bd/s.

Před započítím samotného přenosu má aplikace za úkol zkontrolovat, zda jsou data zobrazitelná na displej. Ve chvíli, kdy kontrola nezjistí žádné chyby, je zobrazeno okno, které vyzve uživatele, aby vybral sériový port, na kterém je připojen metronom. Toto okno je znázorněno na obrázku 17 vlevo. Ve chvíli, kdy uživatel klikne na tlačítko "Upload", aplikace poskládá data za sebe tak, jak budou uložena v paměti a zahájí přenos.

Každý přenos dat je uvozen inicializační zprávou. Tuto zprávu odesílá PC aplikace, metronom s periodou 1 ms kontroluje, zda se ve vstupním bufferu UART sběrnice nenachází nějaká část inicializační zprávy. Inicializační zpráva pro přenos dat z aplikace do metronomu je 0xFEFEFE. Přijme-li metronom celou tuto zprávu, přepne se do režimu příjmu dat. Aplikace odesílá vždy jeden řádek tabulky, další řádek tabulky je odeslán až ve chvíli, kdy od metronomu do aplikace přijde byte 0xFF. Příjem tohoto bytu znamená, že metronom již dokázal zapsat do paměti předchozí řádek a požaduje další data. Přenos je ukončen odesláním bytu 0xFF ve směru od aplikace do metronomu. Stav přenosu je poté vypsán v okně pro výběr portu. V případě úspěšného přenosu je zobrazena zelená hláška "Success". Není-li přenos z nějakého důvodu dokončen, je zobrazena červená hláška "Failed". Tyto stavové hlášky jsou znázorněny na obrázku 17 uprostřed a vpravo.



**Obrázek 17:** Okno výběru portu pro odeslání dat se signalizací stavu dokončeného přenosu.

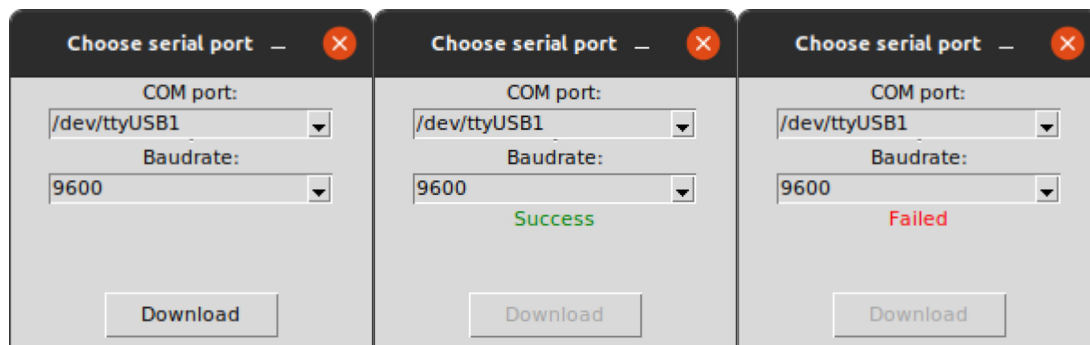
Z pohledu metronomu přenos probíhá obdobně. Metronom s periodou 1 ms odebere jeden byte z vstupního kruhového bufferu. V případě, že přijme sekvenci bytů inicializační zprávy, přepne se do režimu příjmu dat. V tomto režimu odebírá s periodou 1 ms ze vstupního kruhového bufferu jeden byte, který okamžitě zapíše pomocí I2C sběrnice do EEPROM paměti. Přijme-li oddělovač skladeb, odešle aplikaci byte 0xFF, kterým si vyžádá další data. Přijme-li metronom byte 0xFF, potom ukončí přenos a vrátí se do režimu "Hlavní obrazovka".

### 2.2.2.4 Příjem dat ze zařízení

Režim přenosu dat ze zařízení do aplikace umožňuje uživateli pouze upravovat data v metronomu bez nutnosti vytvářet celý seznam od začátku.

Proces příjmu dat z metronomu je mnohem jednodušší, než proces odeslání. Důvodem je fakt, že velikost operační paměti počítače je znatelně větší, než velikost operační paměti mikrokontroléru, můžeme si proto dovolit uložit celý řetězec dat z EEPROM paměti do operační paměti počítače a teprve po dokončení přenosu tento řetězec zpracovat. Zároveň v tomto procesu není nutné ověřovat platnost přijatých dat, neboť počítač nemá omezené možnosti, co se týká zobrazení znaků, na rozdíl od alfanumerického displeje.

Požádá-li uživatel v aplikaci o data z metronomu, otevře se opět okno, ve kterém uživatel vybere sériový port, na kterém je připojen metronom. Toto okno je znázorněno na obrázku 18 vlevo. Kliknutím na tlačítko "Download" je započat příjem dat z metronomu.



**Obrázek 18:** Okno výběru portu pro příjem dat se signalizací stavu dokončeného přenosu.

Proces příjmu dat z metronomu je tedy opět iniciován aplikací. Proces je zahájen odesláním inicializační sekvence 0xFFFFFFFF. Po přijetí této sekvence se metronom přepne do režimu odeslání dat a s periodou 1 ms odesílá vždy jeden byte z paměti zařízení. V počítači se data ukládají do operační paměti. Přenos je ukončen odesláním bytu 0xFF, který je v paměti uložen vždy na konci bloku paměti. Přenos tedy ukončí sám metronom, po ukončení přenosu automaticky přejde do režimu "Hlavní obrazovka" (viz. kapitola 2.2.1.1).

Po ukončení přenosu je v okně pro výběr portu napsáno, zda byl přenos úspěšný či nikoliv. V případě úspěšného přenosu je zobrazena zelená hláška "Success". Není-li přenos z nějakého důvodu dokončen, je zobrazena červená hláška "Failed". Tyto stavové hlášky jsou znázorněny na obrázku 18 vpravo a uprostřed. Po uzavření okna pro výběr portu potom aplikace vypíše do tabulky přijatá data.

## 2.2.3 Popis způsobu uložení dat v paměti

Data jsou v metronomu uložena v 256 kbit EEPROM paměti. Tato velikost paměti umožňuje uložit seznam skladeb dlouhý až 710 položek s průměrnou délkou textu 20 znaků. Tato délka seznamu je plně dostačující, neboť průměrná délka playlistu kapel bývá do 300 skladeb.

Při návrhu struktury paměti se nabízely 3 možnosti uložení dat do paměti:

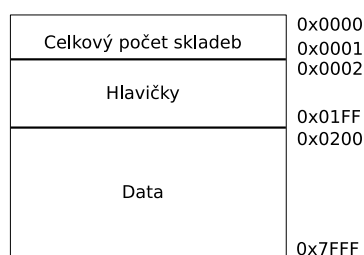
1. skladby uložené v paměti tak, jak přicházejí z aplikace (vyhledávání jedné skladby by znamenalo přečíst prakticky celou paměť),
2. skladby uložené v matici s pevně danými délkami textu (počáteční adresa každé skladby by byla pevně nastavená a bylo by možné ji dopočítat dosazením do vzorce),
3. skladby uložené v paměti tak, jak přicházejí z aplikace (navíc je zařazen blok ukazatelů do paměti, každý z nich odkazuje na začátek dat jedné skladby).

### 2.2.3.1 Struktura paměti

Z výše uvedených možností je třetí možnost tím nejlepším kompromisem mezi předchozími dvěma body. Struktura paměti je znázorněna na obrázku 19. Paměť se skládá ze tří částí.

První dva byty paměti nesou informaci o celkové délce seznamu, tedy nejvyšší číslo skladby.

Druhá část, tedy blok paměti s indexy od 0x0002 do 0x01FF je vyhrazen pro hlavičky, tedy ukazatele na data do bloku dat. Každá z hlaviček zabírá dva byty v paměti. V těchto dvou bytech je uveden index prvního bytu dané skladby v bloku paměti. Index hlavičky je pak pouze dvojnásobkem čísla skladby.



**Obrázek 19:** Struktura dat v EEPROM paměti.



### 2.2.3.2 Výběr oddělovačů

Výběr oddělovačů dat není náhodný. Při výběru oddělovačů bylo nutné počítat s tím, že se jejich hodnota nesmí objevit na žádné jiné pozici v datech. Všechny tři použité oddělovače, tedy 0xFD, 0xFE a 0xFF, jsou tři nejvyšší hodnoty, které je možné vyjádřit osmi bity. Použitý displej pracuje s ASCII tabulkou, jejíž každý znak je možné vyjádřit právě osmi bity. Znaky uvedené v ASCII tabulce pod kódy 0xFD, 0xFE a 0xFF, nejsou běžně používané znaky, jejich absence v textech tedy nijak neovlivní zobrazovaný obsah. Hodnota "BEAT" může nabývat maximální hodnoty 4, také tedy není nijak ovlivněna snížením rozsahu. Jediná hodnota, která je ovlivněna snížením rozsahu, je hodnota "BPM". Ta sice teoreticky může nabývat všech hodnot, které je možné vyjádřit osmi bity, rychlost větší než 0xFD úderů za minutu (dekadicky 253) však nemá reálné využití. Z tohoto důvodu byly vybrány právě hodnoty z konce řady osmibitových čísel.

Alternativně by bylo možné využít tři hodnoty umístěné na začátku řady osmibitových čísel. Například hodnoty 0x05, 0x06 a 0x07 by bylo možné též vypustit, co se týká hodnot "BPM" a "BEAT". Z pohledu řadiče displeje jsou však právě tyto hodnoty významné, neboť právě na prvních osmi pozicích ASCII tabulky displeje se nachází programovatelné znaky. Jak název napovídá, podobu těchto znaků může definovat sám uživatel. Obsazením těchto pozic unikátními znaky by byla znemožněno možné budoucí přidání podpory české diakritiky.

## 2.2.4 Popis architektury SW

Návrh SW probíhá v jazyce C, pomocí vývojového prostředí Eclipse IDE. Hlavní program je obsluhován ve třech časových smyčkách s různou periodou opakování (10 ms, 1 ms a 100  $\mu$ s). Priorita je nastavena tak, že nejkratší smyčka má nejvyšší prioritu a naopak nejdelší smyčka má nejnižší prioritu.

### 2.2.4.1 Časové smyčky

Ve smyčce 100  $\mu$ s je volána funkce pro obsluhu tikání metronomu. Tato funkce má za úkol v závislosti na zbytku programu aktivovat, případně deaktivovat akustický signál reproduktoru, kontrolovat délku jednotlivých akustických signálů a zobrazovat aktuální dobu v taktu pomocí signalizačních LED diod. Akustický signál je generován separátně pomocí čítače/časovače 1. Smyčka 100  $\mu$ s je volána vždy při přetečení čítače/časovače 0.

V 1 ms smyčce je obsluhována komunikace s počítačem (detailněji popsáno v kapitolách 2.2.2.3 a 2.2.2.4) a čtení řádků maticové klávesnice.

V 10 ms smyčce je pak obsluhováno čtení klávesnice jako celku, včetně ošetření zákmitu tlačítek. V návaznosti na stisk tlačítka je vždy zavolána příslušná funkce, která vykoná sekvenci operací, přiřazených k danému tlačítku.

Smyčky 10 ms a 1 ms jsou vytvořeny pomocí čítače, který se inkrementuje vždy ve 100  $\mu$ s smyčce.

### 2.2.4.2 Popis jednotlivých součástí aplikace

Aplikace metronomu se skládá z několika vzájemně propojených podaplikací. Seznam podaplikací a jejich funkcí je uveden v tabulce 4. Každá podaplikace obsluhuje jeden funkční celek.

Podaplikace	Funkce
main.c	Funkce main, Inicializace hardwaru
loops.c	Programové smyčky, obsluha externího přerušení, obsluha přerušení od čítačů
screen.c	Obsluha zobrazení na displej v různých režimech
comm.c	Obsluha komunikace s počítačem
mem.c	Obsluha komunikace s pamětí, inicializace paměti
buttonEvt.c	Události vyvolané stisknutím tlačítka
tick.c	Obsluha tikání metronomu, obsluha signalizačních LED, generování tónu

**Tabulka 4:** Podaplikace a jejich funkce.

### 2.2.4.3 Použité knihovny funkce

Pro obsluhu hardwarových periférií na nejnižší úrovni kódu byly použity již hotové knihovny. Seznam použitých knihoven je uveden v tabulce 5. Prakticky všechny knihovny bylo nutné mírně upravit, většinou z důvodu potřeby vyšší spolehlivosti kódu.

<b>Knihovna</b>	<b>Popis</b>
GPIO	Základní práce s vstupně výstupními porty
LCD	Obsluha alfanumerického displeje
Debounce	Ošetření zákmitu tlačítek
i2cmaster	Práce s I2C sběrnici
UART	Obsluha UART sběrnice
MatrixKeyboard	Čtení maticové klávesnice

**Tabulka 5:** Seznam použitých knihoven.



## 3 Výroba

Výroba celého zařízení proběhla v domácích podmínkách. Na obrázku 22 je zobrazena fotografie celého výrobku.



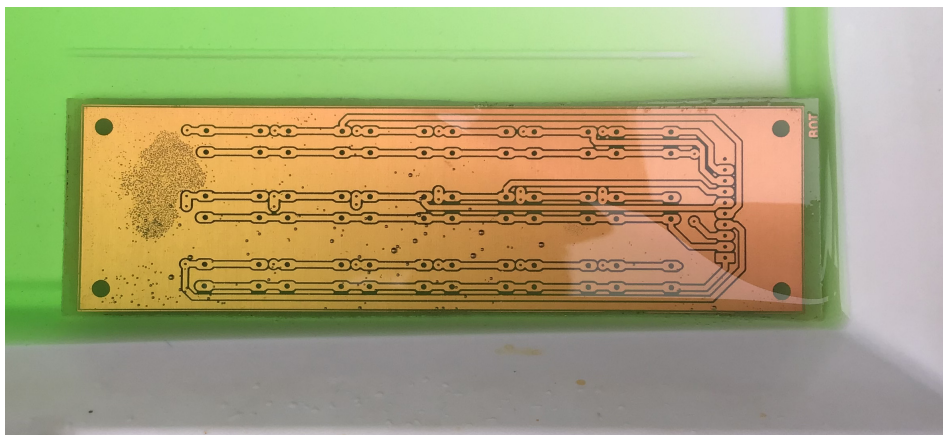
Obrázek 22: Fotografie výsledného hardwaru.

### 3.1 Výroba plošných spojů

Plošné spoje byly vyrobeny fotocestou. Pro výrobu byl použit jednostranný cuprextit (laminát FR4 s nanesenou vrstvou mědi), který má již z výroby nanesenou fotocitlivou vrstvu. Motiv plošného spoje byl vytištěn pomocí laserové tiskárny na průhlednou folii. Tato folie s motivem byla položena na fotocitlivou vrstvu cuprextitu, motiv byl následně přenesen na fotocitlivou vrstvu pomocí UV záření. Během výroby je nutné dodržet přesnou dobu expozice plošného spoje. Tuto délku je velmi obtížné vypočítat, vyplývá tedy z předchozích zkušeností. Při výrobě plošných spojů pro tuto práci byla zvolena délka expozice 23 minut. Osvětlovací jednotka, skládající se ze čtyř 14 W zářivek byla umístěna 34 cm nad povrchem plošného spoje.

Osvícený plošný spoj se následně vloží do vývojky. Ta má za úkol odplavit exponované části fotocitlivé vrstvy z povrchu tak, aby zůstaly kryté pouze ty vrstvy mědi, které mají na plošném spoji zůstat. Vývojka je 2 % roztok hydroxidu sodného. Dobu, po kterou je potřeba ponechat plošný spoj ve vývojce, není jednoduché přesně určit. Celý proces vyvolání je velmi závislý na teplotě roztoku, kvalitě expozice a tloušťce fotocitlivé vrstvy. Obecně je však potřeba ponechat plošný spoj ve vývojce, dokud není viditelná změna odstínu odplavovaných částí fotocitlivé vrstvy.

Z vyvolaného plošného spoje je možné odleptat nepotřebné části mědi. Odleptání se provádí reakcí kyseliny chlorovodíkové a peroxidu vodíku. Vyvolaný plošný spoj se nejprve vloží do 15 % roztoku kyseliny chlorovodíkové, následně se do misky s kyselinou postupně přilévá malé množství 30 % roztoku peroxidu vodíku. Peroxid vodíku je nutné postupně dolévat po celou dobu leptání. Čím více peroxidu vodíku obsahuje roztok, tím více bude roztok zbarven do zelené barvy, což je možné vidět na obrázku 23. V případě, že bude do nádoby dolito více peroxidu vodíku, může se stát, že se měď bude odplavovat příliš rychle a povrch plošného spoje začne šumět. V tuto chvíli je nutné okamžitě vyndat plošný spoj z leptacího roztoku a ponořit jej do nádoby s vodou. Pokračovat v leptání je možné až po zředění roztoku. Leptání probíhá až do chvíle, kdy jsou odleptány všechny nepotřebné vrstvy mědi.



**Obrázek 23:** Leptání plošného spoje.

## 4 Závěr

V této práci byl vytvořen metronom s možností uložení předvoleb do paměti. Tyto předvolby jsou v paměti uloženy ve formě seznamu. Uživatel vždy vybírá ze seznamu číslo skladby, podle čísla skladby metronom dokáže nastavit hodnoty počtu úderů za minutu a počet dob v taktu. Zároveň zobrazuje na displej název skladby a případnou textovou poznámku. Bylo vytvořeno uživatelské rozhraní, skládající se ze dvou částí. První částí je samotná jednotka metronomu, druhou částí je počítačová aplikace sloužící k vytvoření seznamu skladeb. Aplikace s metronomem komunikuje pomocí sériového kanálu.

Metronom se podařilo zprovoznit. Byl vytvořen první, plně funkční prototyp hardwaru metronomu, který byl předveden zadavateli. Tento prototyp je osazen na plošných spojích domácí výroby. V budoucnosti dojde k výrobě profesionálních plošných spojů, obsahujících nepájivou masku a potisk. V nové verzi plošného spoje, budou opraveny drobné chyby, které byly objeveny v průběhu "oživování". Zároveň bude pomocí 3D tiskárny vytvořeno pouzdro.

Software metronomu a počítačové aplikace jsou plně funkční. Metronom umožňuje práci ve všech režimech, uvedených v práci. Zároveň dokáže generovat akustický signál, signalizující doby v taktu. Počítačová aplikace umožňuje vytvoření seznamu skladeb, případně jeho načtení ze souboru a následné odeslání do zařízení. Seznam skladeb je možné načíst i přímo z metronomu.

Zadavatel měl k metronomu pouze drobné výhrady. Požadoval pouze přidání některých funkcí, spojených s inkrementálním čidlem. Metronom bude nově za poslední číslo v seznamu řadit číslo první a naopak. Dalším požadavkem byl požadavek na možnost změny hodnot "BPM" a "BEAT" pomocí inkrementálního čidla. Všechny tyto požadavky byly dodatečně splněny.

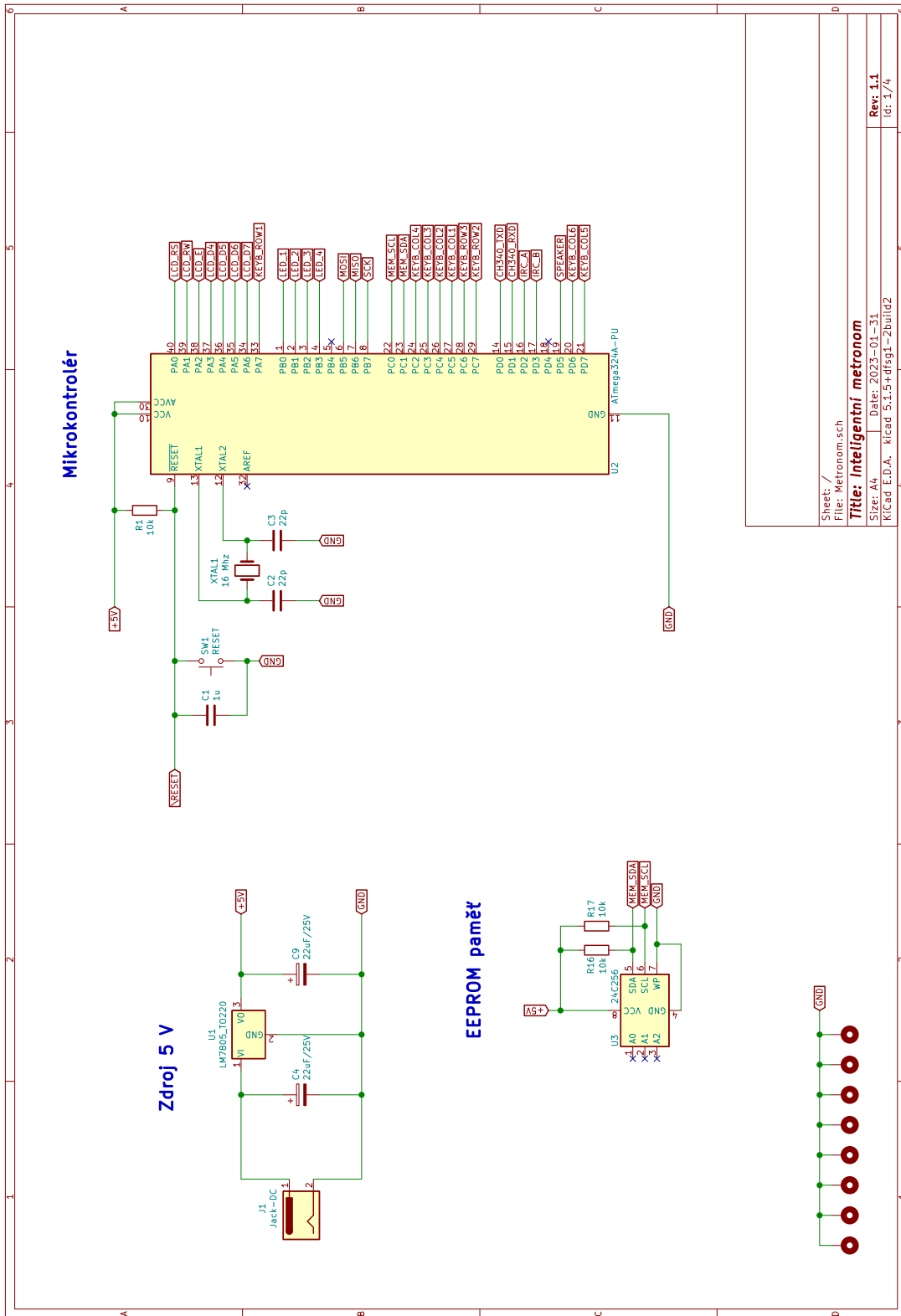
Software se bude nadále rozvíjet. Zadavatel navrhl nový projekt, ve kterém by měla být realizována výroba čtecího zařízení. Požaduje, aby čtecí zařízení komunikovalo s metronomem, tedy aby se číslo skladby na metronomu synchronizovalo s číslem skladby ve čtecím zařízení. Ke komunikaci mezi čtecím zařízením a metronomem bude využito stávajícího sériového kanálu, který v době, kdy neprobíhá přenos dat do zařízení, zůstává nevyužitý.

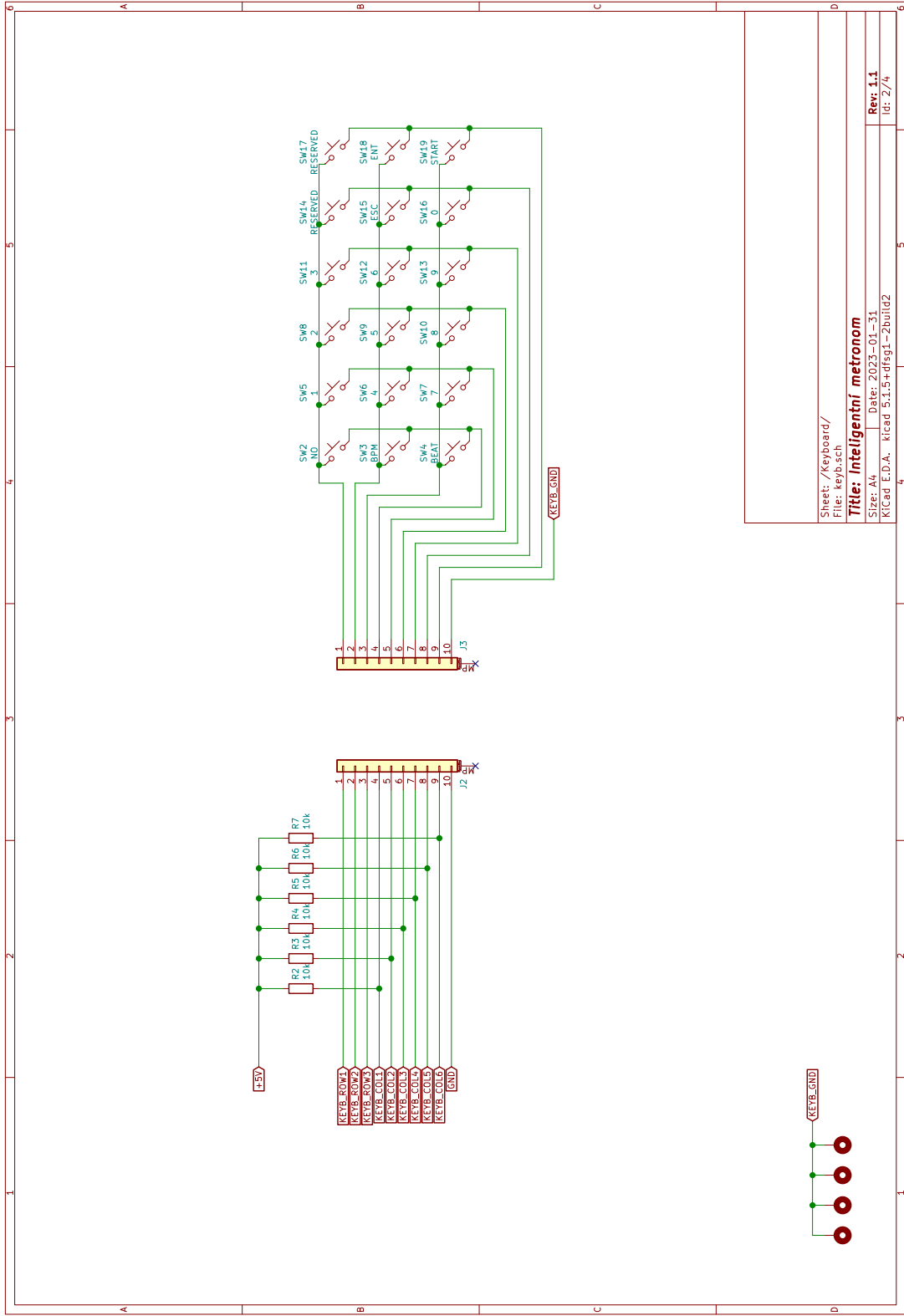
Jedná se pouze o drobnou změnu softwaru, výkon stávajícího mikrokontroléru bude i nadále plně dostačovat. Už od začátku návrhu softwaru však bylo počítáno s možností jeho přenosu na jiný mikrokontrolér s vyšším výkonem. Software má proto oddělené moduly komunikující s mikrokontrolérem na nejnižší úrovni od zbytku programu. V případě přenosu softwaru by tedy bylo nutné vytvořit pouze tyto moduly.

# Seznam použité literatury

- [1] *LM340, LM340A and LM78xx Wide VIN 1.5-A Fixed Voltage Regulators, Datasheet*, 2000. URL: <https://www.alldatasheet.com/datasheet-pdf/pdf/838007/TI1/LM7805.html>.
- [2] „ATmega164A/PA/324A/PA/644A/PA/1284/P, megaAVR® Data Sheet.“ (2020), URL: [https://ww1.microchip.com/downloads/aemDocuments/documents/MCU08/ProductDocuments/DataSheets/ATmega164A\\_PA-324A\\_PA-644A\\_PA-1284\\_P\\_Data-Sheet-40002070B.pdf](https://ww1.microchip.com/downloads/aemDocuments/documents/MCU08/ProductDocuments/DataSheets/ATmega164A_PA-324A_PA-644A_PA-1284_P_Data-Sheet-40002070B.pdf) (cit. 09. 05. 2023).
- [3] „HD44780 Datasheet (PDF) Download - Hitachi Semiconductor.“ (), URL: <https://pdf1.alldatasheet.com/datasheet-pdf/download/63673/HITACHI/HD44780.html> (cit. 29. 03. 2023).
- [4] „CH340G USB to UART Interface Datasheet.“ (), URL: <https://www.alldatasheet.com/datasheet-pdf/pdf/1132618/ETC2/CH340G.html> (cit. 09. 05. 2023).
- [5] „I<sup>2</sup>C-Compatible (Two-Wire) Serial EEPROM 256-Kbit (32,768 x 8).“ (), URL: <https://ww1.microchip.com/downloads/aemDocuments/documents/MPD/ProductDocuments/DataSheets/AT24C256C-1%5C%2%5C%B2C-Compatible-%5C%28Two-Wire%5C%29-Serial-EEPROM-256-Kbit-%5C%2832%5C%2C768x8%5C%29-Data-Sheet-DS20006042.pdf> (cit. 09. 05. 2023).
- [6] „KiCad EDA, A Cross Platform and Open Source Electronics Design Automation Suite.“ (), URL: <https://www.kicad.org/>.
- [7] „JHD762-004 Datasheet.“ (), URL: <https://www.alldatasheet.com/datasheet-pdf/pdf/276111/JHD/JHD762-204.html> (cit. 09. 05. 2023).
- [8] „Understanding the I2C Bus.“ (2015), URL: [https://www.ti.com/lit/ml/slva704/slva704.pdf?ts=1680108161191%5C&ref\\_url=https%5C%253A%5C%252F%5C%252Fwww.google.com%5C%252F](https://www.ti.com/lit/ml/slva704/slva704.pdf?ts=1680108161191%5C&ref_url=https%5C%253A%5C%252F%5C%252Fwww.google.com%5C%252F) (cit. 26. 04. 2023).
- [9] *UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter*, 2020. URL: <https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html>.

# Příloha A





KEYB\_GND

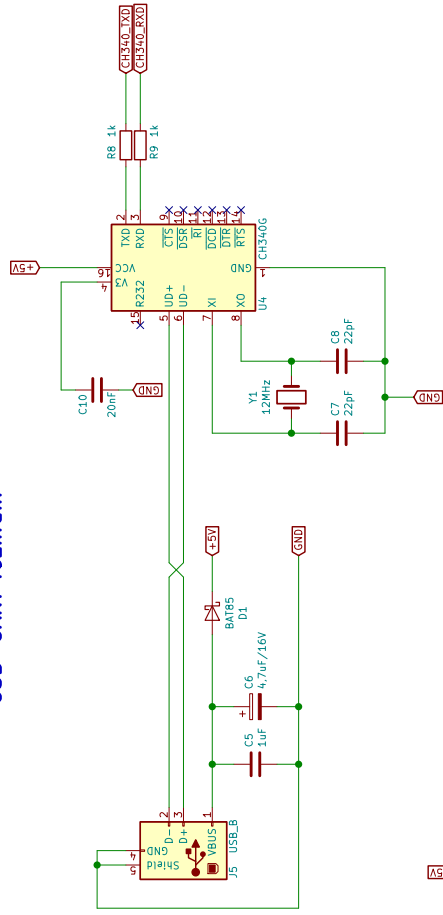
Sheet: /Keyboard/  
 File: keyb.sch

**Title: inteligentní metronom**

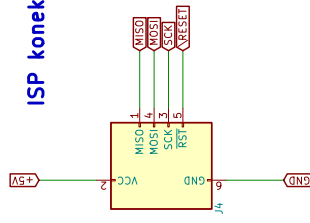
Size: A4 Date: 2023-01-31  
 KiCad E.D.A. kicad 5.1.5+dfsg1-2build2

Rev. 1.1  
 Id: 2/4

## USB-UART rozhraní



## ISP konektor



Sheet: /Communication/  
File: comm.sch

Title: **inteligentní metronom**

Size: A4

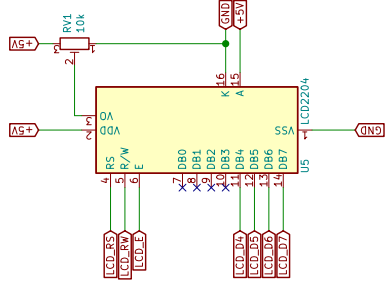
Date: 2023-01-31

KiCad E.D.A. kicad 5.1.5-rdfsg1-2build2

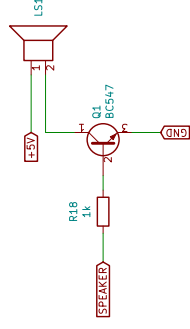
Rev. 1.1

Id: 3/4

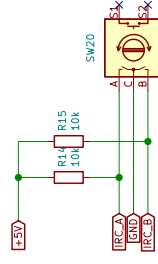
### Displej



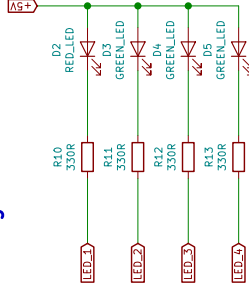
### Reproduktor



### Inkrementální čidlo



### Signalizační LED



Sheet: //IO peripherals/

File: ioper.sch

Title: **intelligentní metronom**

Size: A4

Date: 2023-01-31

KiCad E.D.A. - KiCad 5.1.5+dfsg1-2build2

Rev. 1.1

Id: 4/4