

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra mikroelektroniky



Bakalářská práce

Programovatelná klávesnice

Šimon Honsů

Vedoucí: Ing. Tomáš Teplý

Studijní program: Elektronika a komunikace

Praha, Květen 2023

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Honsů** Jméno: **Šimon** Osobní číslo: **499147**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra mikroelektroniky**
Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Programovatelná klávesnice

Název bakalářské práce anglicky:

Programmable Keyboard

Pokyny pro vypracování:

Navrhněte a realizujte programovatelnou klávesnici, která se bude skládat z alespoň deseti programovatelných kláves, informačního displeje a ovládacích prvků pro snadnou změnu předdefinovaných funkcí kláves (profilů). Vytvořte alespoň tři profily, na kterých bude demonstrována funkce klávesnice. Pro připojení k PC zvolte standardní komunikační sběrnici USB.

Postup řešení:

- 1) Proveďte průzkum trhu v oblasti programovatelných klávesnic a na jeho základě definujte funkce, které považujete za důležité, aby vaše zařízení podporovalo.
- 2) Proveďte návrh obvodového zapojení a výběr vhodných součástek pro realizaci zařízení.
- 3) Proveďte návrh desky plošných spojů a zařízení realizujte.
- 4) Shrňte dosažené výsledky.

Seznam doporučené literatury:

- [1] Robert B. Reese: Microprocessors From Assembly Language to C Using the PIC18Fxx2, Da Vinci Engineering Press, Hingham, Massachusetts, 2005
- [2] Záhlava, V.: Návrh a konstrukce desek plošných spojů, BEN, Praha, 2011
- [3] Brian W. Kernighan, Dennis M. Ritchie: Programovací jazyk C, Computer Press, a.s., Brno 2006

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Tomáš Teplý katedra mikroelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **13.02.2023**

Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce: **22.09.2024**

Ing. Tomáš Teplý
podpis vedoucí(ho) práce

prof. Ing. Pavel Hazdra, CSc.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 26. května 2023

.....

Poděkování

Tímto bych chtěl poděkovat vedoucímu mé bakalářské práce Ing. Tomáši Teplému za cenné rady a konzultace, které mi během psaní práce poskytl. Dále bych také rád poděkoval své rodině, přítelkyni a přátelům za trpělivost a podporu v průběhu mého dosavadního studia.

Abstrakt

Tato práce se zabývá návrhem a realizací programovatelné klávesnice neboli macropadu, jejímž účelem je zefektivnění každodenní práce na počítači. V první řadě je proveden průzkum trhu, který představuje několik komerčně dostupných klávesnic a na jeho základě jsou definovány požadavky na mnou navrhované zařízení. Následně jsou v textu blíže popsány jednotlivé komponenty a použitá komunikační rozhraní, přičemž proveden je také výběr vhodných součástí pro mojí aplikaci. Výstupem práce je plně funkční programovatelná klávesnice, která implementuje mechanické spínače, rotační enkodéry, OLED displej a připojení pomocí sběrnice USB. Vytvořen je také doprovodný počítačový software, který slouží ke zpracování dat z klávesnice, její konfiguraci a vykonávání požadovaných funkcí.

Klíčová slova: programovatelná klávesnice, macropad, mechanické spínače, enkodér, OLED displej, USB

Abstract

This thesis focuses on the design and implementation of a programmable keyboard or macropad, which purpose is to streamline everyday computer work. Firstly, market research is conducted to present several commercially available programmable keyboards. Based on this research, the requirements for my device are defined. Subsequently, the individual components and the communication interfaces used are described in more detail and the selection of suitable components for my application is made. The output of the thesis is a fully functional programmable keyboard that implements mechanical switches, rotary encoders, an OLED display and USB bus connectivity. Additionally, accompanying computer software is created to process data from the keyboard, configure it and perform the required functions.

Keywords: programmable keyboard, macropad, mechanical switches, encoder, OLED display, USB

Obsah

Seznam použitých zkratek

1 Úvod	1
1.1 Cíl práce	1
2 Průzkum trhu	2
2.1 Adafruit MacroPad	2
2.2 Elgato Stream Deck	3
2.3 DuckyPad	4
2.4 Shrnutí	5
3 Požadavky na zařízení	7
4 Výběr součástek	8
4.1 Mikrokontrolér	8
4.1.1 Programování mikrokontroléru	8
4.1.2 Zvolený mikrokontrolér	8
4.2 Komunikace s počítačem	9
4.2.1 Řešení pomocí USB HID	9
4.2.2 Řešení pomocí sériového portu	9
4.3 Mechanické spínače	10
4.3.1 Typy spínačů	10
4.3.2 Srovnání s membránovými klávesnicemi	11
4.3.3 Hot-swappable patice	11
4.3.4 Zvolené spínače	12
4.4 Rotační enkodér	12
4.4.1 Typy enkodérů	12
4.4.2 Mechanismus enkodéru	13
4.4.3 Zvolené enkodéry	13
4.5 Displej	13
4.6 LED podsvícení	14
5 Použitá komunikační rozhraní	16
5.1 UART	16
5.1.1 Zapojení	16
5.1.2 Formát dat	16
5.1.3 Modulační rychlost	17
5.2 SPI	17
5.2.1 Zapojení	17
5.2.2 Formát dat	18
5.3 USB	18
5.3.1 Topologie	19
5.3.2 Enumerace	19
5.3.3 Konektor	20

6	Návrh zapojení	21
6.1	USB	21
6.2	Pull-up rezistory	22
6.3	Displej	22
7	Návrh desky plošných spojů	24
8	Program mikrokontroléru	26
8.1	Programový zavaděč	26
8.2	Cílová aplikace	27
8.2.1	Obsluha displeje	27
8.2.2	Obsluha kláves	28
8.2.3	Obsluha enkodérů	28
8.2.4	Příjem konfigurace	29
9	Počítačová aplikace	31
9.1	Grafické rozhraní	31
9.2	Funkce kláves a enkodérů	32
9.3	Detekce klávesnice	33
9.4	Nahrávání konfigurace	34
10	Realizace zařízení	35
10.1	Demonstrační profily	36
11	Zhodnocení dosažených výsledků	37
11.1	Kalkulace ceny	37
11.2	Srovnání s komerčně dostupnými zařízeními	37
11.3	Možná vylepšení	38
12	Závěr	39
	Literatura	40
	Přílohy	44
A	Obsah elektronické přílohy	44
B	Schéma zapojení klávesnice	45
C	Osazená deska plošných spojů	48
D	Nastavení demonstračních profilů	49

Seznam použitých zkratek

ASCII	American Standard Code for Information Interchange
CS	Chip Select
DC	Data/Command
DPS	Deska Plošných Spojů
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESD	Electrostatic Discharge
GIF	Graphics Interchange Format
GUI	Graphic User Interface
HID	Human Interface Device
I2C	Inter-Integrated Circuit
ID	Identifier
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
LSb	Least Significant Bit
MIDI	Musical Instrument Digital Interface
MISO	Master In, Slave Out
MOSI	Master Out, Slave In
MSb	Most Significant Bit
OLED	Organic Light-Emitting Diode
PBT	Polybutylentereftalát
PWM	Pulse Width Modulation
RC	Resistor-Capacitor
RES	Restart
RGB	Red-Green-Blue
SCK	Serial Clock
SCLK	Serial Clock
SD	Secure Digital
SDA	Serial Data
SPI	Serial Peripheral Interface
SS	Slave Select
UART	Universal Asynchronous Receiver/Transmitter
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

Kapitola 1

Úvod

Práce na počítači je pro většinu z nás každodenní záležitostí. Existuje mnoho klávesových zkratk, které mohou tuto práci zefektivnit, mnohdy je však obtížné je stisknout nebo si je vůbec zapamatovat. S tímto problémem nám může pomoci programovatelná klávesnice neboli macropad. Jedná se o malou počítačovou klávesnici, jejímž úkolem je poskytnout jednoduchý přístup k často používaným klávesovým zkratkám a funkcím.

Každá klávesa macropadu má přiřazené tzv. makro. Makro je obecně sekvence příkazů, ve spojení s programovatelnými klávesnicemi jde zejména o sekvenci stisků kláves. Díky tomu nám macropad umožňuje nahradit stisk více kláves na standardní klávesnici stiskem pouze jedné klávesy. S pomocí doprovodného softwaru spuštěného na počítači může macropad při stisku klávesy vykonávat i složitější funkce než vyvolávání klávesových zkratk. Příkladem může být otevírání a ovládání různých programů nebo spouštění uživatelem vytvořených skriptů.

Důležitou vlastností programovatelných klávesnic je také to, že uživatel má možnost si funkce kláves nadefinovat sám dle vlastních specifických potřeb a tak maximálně zefektivnit práci na počítači.

1.1 Cíl práce

Cílem mé práce je navrhnout a realizovat vlastní programovatelnou klávesnici. Mezi základní požadavky na zařízení patří alespoň deset programovatelných kláves, informační displej zobrazující funkce jednotlivých kláves a připojení k počítači pomocí sběrnice USB. Zároveň je mým úkolem vytvořit alespoň tři profily, které specifikují funkce kláves a přidat prvky pro jejich snadné přepínání.

Další funkce mého macropadu budou definovány na základě následujícího průzkumu trhu v oblasti programovatelných klávesnic.

V rámci práce bude proveden výběr vhodných součástek, bude navrženo obvodové zapojení a deska plošných spojů a v závěru bude zařízení realizováno.

Kapitola 2

Průzkum trhu

Na trhu je k dostání velké množství programovatelných klávesnic, ať už v podobě hotových výrobků nebo ve formě sad, které si zákazník sestaví sám. Na internetu lze také najít mnoho návodů na stavbu vlastního macropadu. V této kapitole bych rád představil několik komerčně dostupných zařízení, která jsou populární nebo mě při průzkumu jiným způsobem zaujala.

2.1 Adafruit MacroPad

Jedním z populárních zařízení je macropad vyráběný společností Adafruit [1], který je k vidění na obrázku 1. Ve své práci jsem ho rozhodl zmínit zejména kvůli jeho zajímavé výbavě spojené s příznivou cenou, která činí přibližně \$50.



Obrázek 1: Adafruit MacroPad [1]

Hlavní součástí tohoto macropadu je dvanáct kláves s mechanickými spínači a tzv. hot-swappable patičkami, které umožňují jednoduchou výměnu spínačů bez nutnosti pájení. Mechanické spínače i patice budou blíže popsány v podkapitole 4.3 dále v této práci. Mezi další zajímavé součásti macropadu patří černobílý OLED displej s řadičem SH1106 a rozlišením 128 x 64 pixelů, rotační enkodér s tlačítkem nebo RGB LED podsvícení. Pro připojení k počítači slouží konektor USB typu C. Zařízení také disponuje malým reproduktorem a speciálním konektorem pro připojení periférií po sběrnici I2C, osobně však považuji tyto komponenty na klávesnici za zbytečné.

O řízení macropadu se stará výkonný dvoujádrový mikrokontrolér RP2040 vyvinutý společností Raspberry Pi doplněný o 8 MB externí flash paměti. Jelikož výrobcem není dodáván žádný software, pomocí kterého by bylo možné zařízení konfigurovat, je nutné,

aby uživatel sám mikrokontrolér naprogramoval a definoval tak funkce jednotlivých periferií. Podporovanými jazyky jsou Arduino, MicroPython a CircuitPython. Díky tomu může klávesnice plnit širokou škálu funkcí. Například se může jevit jako standardní počítačová klávesnice a využívat tzv. HID ovladačů, které má většina moderních operačních systémů předinstalované nebo ji lze využít jako MIDI zařízení pro tvorbu hudby.

Výhodou je také to, že díky popularitě produktu je na internetu k dispozici mnoho již hotových programů a knihoven, které může uživatel využít.

2.2 Elgato Stream Deck

Bezpochyby nejprodávanějším zařízením v oblasti programovatelných klávesnic je Elgato Stream Deck [2] vyobrazený na obrázku 2. Jedná se o speciální klávesnici, jejíž hlavní předností jsou tlačítka se zabudovanými miniaturními LCD displeji. Jednotlivé displeje mohou zobrazovat text, obrázek nebo GIF a přehledně tak dávat najevo, jaká je funkce příslušného tlačítka. Stream Deck je dostupný v několika verzích – základní verze MK.2 s 15 tlačítky, verze Mini se 6 tlačítky a XL s 32 tlačítky. Nejnovějším modelem je Stream Deck +, jehož 8 tlačítek je doplněno o podlouhlý dotykový displej a 4 rotační enkodéry s tlačítkem. Všechny modely jsou vybaveny USB C konektory využívající rozhraní USB 2.0.



Obrázek 2: Elgato Stream Deck MK.2 [2]

Další předností těchto zařízení je propracovaný konfigurační software, který je dostupný k zařízení a je nutný pro jeho správný chod. Program nabízí přehledné GUI, kde může uživatel nastavit funkce tlačítek a obsah, který mají zobrazovat příslušné displeje. Mezi základní funkce kláves patří například stisk klávesové zkratky, ovládání multimedií nebo spouštění aplikací. Stream Deck zároveň nabízí vysokou integraci s mnoha

programy s pomocí dodatečných rozšíření. Tato rozšíření lze nainstalovat z virtuálního obchodu, který je součástí softwaru a lze díky nim klávesám přiřadit funkce specifické pro dané programy. V důsledku toho může Stream Deck značně urychlit například stříh videa, editaci fotek, tvorbu hudby a mnoho dalších. Mezi ostatní užitečné funkce patří také možnost využívat tlačítka jako složky pro další funkce nebo možnost vytvořit na zařízení více profilů, které se mohou automaticky přepínat na základě momentálně aktivní aplikace na uživatelském počítači.

Jak bylo popsáno, Stream Deck disponuje opravdu bohatou výbavou, avšak té odpovídá cena, která se u všech modelů pohybuje v řádech jednotek tisíc korun.

2.3 DuckyPad

Další programovatelnou klávesnicí, která mě zaujala svými parametry je DuckyPad. Zařízení lze zakoupit na webu [3] ve formě konfigurovatelného setu, který vyžaduje pouze minimální sestavení. Na rozdíl od předchozích zařízení však jde o open source projekt, tudíž veškerý kód a soubory potřebné pro vlastní realizaci jsou také dostupné na webu GitHub [4].

DuckyPad (obrázek 3) je podobně jako Adafruit MacroPad vybaven patnácti mechanickými klávesami s hot-swappable paticemi, USB C konektorem a 1,3“ OLED displejem s rozlišením 128 x 64 pixelů. Displej je zde použit pro zobrazování funkcí jednotlivých kláves. Řídícím čipem je mikrokontrolér STM32F072C8T6 od firmy STMicroelectronics. Jednotlivé klávesy jsou individuálně podsvíceny pomocí RGB LED SK6812MINI-E, které jsou pod tlačítka připájeny z druhé strany a svítí skrz otvory v DPS.



Obrázek 3: DuckyPad [3]

Výhodou oproti Stream Decku je to, že DuckyPad funguje samostatně a nevyžaduje pro svoji funkci žádný dodatečný ovladač nebo program. Toho je docíleno tím, že DuckyPad emuluje standardní klávesnici a myš a stejně jako Adafruit MacroPad využívá předinstalovaných HID ovladačů. Do počítače jsou při stisknutí tlačítka macropadu odeslány binární hodnoty reprezentující klávesy na standardní klávesnici. Tyto hodnoty se

nazývají HID Usage IDs a jsou specifikovány v dokumentu HID Usage Tables [5]. S využitím těchto kódů macropad emuluje stisky kláves na klávesnici. Zdrojem komplikací u macropadů může být to, že každé HID Usage ID odpovídá pouze fyzické pozici klávesy na klávesnici, a ne znaku nebo funkci, kterou představuje. Následně je až na operačním systému počítače, jak daný kód vyhodnotí na základě toho, jaké je zrovna nastavené rozložení klávesnice. Z toho důvodu DuckyPad umožňuje nastavit momentálně používané rozložení klávesnice, aby počítač stisky kláves správně vyhodnocoval. Díky tomu při přechodu např. z českého rozložení klávesnice na anglické stačí DuckyPadu přenastavit rozložení klávesnice a není nutné měnit všechny již nadefinované funkce kláves.

Pro konfiguraci tohoto produktu autor poskytuje doprovodnou aplikaci, která umožňuje zařízení nastavit s využitím GUI. Použití této aplikace však není nutností, jelikož DuckyPad poskytuje pro konfiguraci vlastní skriptovací jazyk DuckyScript. Uživatelé stačí s použitím tohoto jazyka vytvořit textové soubory, které popisují jednotlivé profily a funkce kláves a následně je uložit na SD kartu, která je umístěna v macropadu. Použití SD karty zajišťuje, že si zařízení pamatuje konfiguraci a také umožňuje její rychlý přesun mezi různými zařízeními.

Jak již bylo zmíněno, tento macropad také umožňuje uživateli vytvořit více profilů a zněkolikanásobit tak množství využitelných funkcí. Součástí zařízení jsou i dvě tlačítka pro rychlou změnu těchto profilů. Na GitHub stránce projektu [4] je zdarma dostupný program DuckyPad Profile Auto-switcher, který umožňuje automaticky měnit profily klávesnice podle právě používaného programu stejně jako je tomu u Stream Decku.

2.4 Shrnutí

Pro přehledné srovnání parametrů popsaných klávesnic jsem na závěr průzkumu trhu vytvořil tabulku 1. Do tabulky jsem zároveň zahrnul i další zařízení, která však nedisponují žádnými inovativními funkcemi, a tudíž nebylo potřebné věnovat jim vlastní podkapitulu.

Tabulka 1: Porovnání klávesnic

Název	Adafruit Macropad [1]	Elgato Stream Deck ¹ [2]	DuckyPad ² [3]	Mountain DisplayPad [6]	Pikatea Macropad R26 ² [7]	Moje zařízení
Klávesy	hot-swap mechanické	s LCD displeji	hot-swap mechanické	s LCD displeji	hot-swap mechanické	hot-swap mechanické
Enkodér	Ano	pouze model +	Ne	Ne	Ano	Ano
Samostatný displej	Ano	pouze model +	Ano	Ne	Ne	Ano
Podsvícení	Ano	-	Ano	-	Ano	Ano
Doprovodný software	Ne	Ano	Ano	Ano	Ano	Ano
Konektor	USB C	USB C	USB C	USB C	USB C	USB C
Cena	\$49,95	€79,99- €249,99	\$95,96- \$122,95	\$109,99	\$53,69- \$87,69	-

¹ Cena zařízení závisí na zvoleném modelu.

² Cena zařízení závisí na konfiguraci, přičemž minimální cena odpovídá konfiguraci, která již zahrnuje všechny komponenty potřebné pro okamžité používání.

Pro zobrazování funkcí klávesnice se jako velmi elegantní způsob jeví použití kláves se zabudovanými displeji jako je tomu u zařízení Elgato Stream Deck a Mountain DisplayPad. Tyto displeje jsou však náročné na implementaci, čemuž odpovídá vyšší cena klávesnic. Podstatně levnějším řešením, které poskytuje podobnou funkcionalitu je využití mechanických kláves a jednoho samostatného displeje jako je tomu u macropadů Adafruit a DuckyPad.

Jak je vidět v tabulce 1, žádné ze zařízení nenabízí mechanické vyměnitelné klávesy v kombinaci se všemi ostatními komponentami a konfiguračním softwarem. Toto je jedna z hlavních motivací pro návrh vlastní programovatelné klávesnice, která by všemi těmito vlastnostmi disponovala.

Návrh mého zařízení jsem pro srovnání také zahrnul do tabulky, přičemž jeho hlavní parametry a požadavky na něj budou blíže popsány v následující kapitole. Jeho finální cena bude uvedena v závěru práce.

Kapitola 3

Požadavky na zařízení

Dle zadání mezi základní požadavky na mnou navrhované zařízení patří minimálně deset kláves, možnost vytvoření více profilů, informační displej zobrazující momentálně aktivní profil a připojení k počítači pomocí sběrnice USB.

Na základě předchozího průzkumu trhu považuji za podstatné, aby moje programovatelná klávesnice dále disponovala následujícími komponentami. Pro plánovaných dvanáct kláves se jeví jako vhodné využít mechanické spínače a patice umožňující jednoduchou výměnu spínačů bez nutnosti pájení. Připojení k počítači bude provedeno prostřednictvím USB konektoru typu C, jelikož jde o odolný oboustranný konektor, který je v dnešní době velmi rozšířený, jak je vidět v tabulce 1. Další zajímavou součástí, kterou bych na svém zařízení rád využil je rotační enkodér. Ten může mít mnoho využití jako například ovládání hlasitosti počítače nebo přepínání profilů klávesnice. Zároveň bych rád přidal podsvícení kláves pomocí RGB LED, které ulehčuje použití ve tmě, může dávat uživateli zpětnou vazbu o stisknutí tlačítka a v neposlední řadě přispívá k celkovému estetickému dojmu klávesnice.

V rámci práce plánuji také vytvořit pro mojí klávesnici doprovodnou aplikaci ve stylu té, kterou využívá popsany Stream Deck. Ta bude sloužit k jednoduché konfiguraci zařízení, vytváření a správě profilů a především nastavování funkcí kláves. Hlavním důvodem pro tvorbu tohoto programu je to, aby nebylo třeba aktualizovat program nahraný v klávesnici v případě, že uživatel chce změnit její konfiguraci. Na druhou stranu však musí existovat jednoduchý způsob, jak klávesnici přeprogramovat a to převážně pro účely prvotního vývoje.

Kapitola 4

Výběr součástek

V této kapitole bude rozebrána problematika jednotlivých komponent mnou navrhované klávesnice a budou vybrány vhodné součástky pro moji aplikaci. Zmíněná komunikační rozhraní, která jednotlivé součástky využívají, budou blíže popsána v kapitole 5.

4.1 Mikrokontrolér

Řídicím prvkem mojí klávesnice bude mikrokontrolér, který bude realizovat obsluhu displeje, zpracovávání vstupů z periférií a následné odesílání dat do počítače.

Mikrokontrolér je integrovaný obvod obsahující mikropočítač. Podle [8] je primárně složený z mikroprocesoru a programové a datové paměti, přičemž důležitou součástí jsou také vstupně-výstupní porty umožňující komunikaci s externími zařízeními. Mezi další periferie, které mohou být na čipu integrovány patří např. oscilátory, časovače, převodníky nebo různá komunikační rozhraní jako UART, SPI či I2C.

Podle [9] na rozdíl od mikroprocesorů, které známe z počítačů a které vyžadují mnoho externích zařízení pro správnou funkci, mikrokontroléry integrují všechny potřebné komponenty na jednom čipu a jsou soběstačné. Mikrokontroléry se vyznačují podstatně nižší cenou, která ale odpovídá jejich nižšímu výkonu. Z těchto důvodů se mikrokontroléry hodí převážně pro jednoúčelové, výpočetně méně náročné aplikace, se kterými se můžeme setkat všude okolo nás. Jednou z nich může být právě obsluha programovatelné klávesnice.

4.1.1 Programování mikrokontroléru

Důležitou součástí práce s mikrokontroléry je jejich programování. Základem je napsání programu, který popisuje, jak má mikrokontrolér pracovat. Následnou kompilací programu je získán binární soubor, který je za pomoci programátoru nahrán do programové paměti mikrokontroléru. Další možností pro nahrání programu je tzv. programový zavaděč, jehož funkce bude blíže popsána v podkapitole 8.1.

4.1.2 Zvolený mikrokontrolér

Pro řízení mého macropadu jsem zvolil mikrokontrolér PIC18F26K42 od firmy Microchip [10]. Jedná se o 8bitový mikrokontrolér s pracovní frekvencí až 64 MHz. Nabízí 24 vstupně-výstupních pinů, přičemž pin 1 lze využít pro externí reset zařízení nebo jako digitální vstup. Pro program je vyhrazeno 64 kB programové paměti a součástí je i 1024 B paměti typu EEPROM, která je nevolatilní a uchovává si tak data i po vypnutí napájení. Tento PIC18 zároveň podporuje všechny základní komunikační rozhraní jako je UART, SPI či I2C.

Hlavními důvody pro výběr tohoto mikrokontroléru byla jeho příznivá cena, dobrá dostupnost v době psaní práce a naprosto dostačující výbava pro moji aplikaci. Dalšími faktory byla moje již předchozí zkušenost s mikrokontroléry řady PIC18 a také to, že tento integrovaný obvod je dostupný v pouzdře SOIC28, které bude možné pohodlně ručně připájet.

4.2 Komunikace s počítačem

4.2.1 Řešení pomocí USB HID

Při rozhodování, jakým způsobem by moje klávesnice měla komunikovat s počítačem, jsem z počátku zvažoval realizaci pomocí HID zařízení po vzoru popsaných zařízení Adafruit a DuckyPad. Pro tento způsob řešení jsem plánoval použít čip CH9329 značky WinChipHead [11]. Ten realizuje převod UART komunikace z mikrokontroléru na standard USB a po připojení k počítači je rozpoznán jako USB klávesnice, myš nebo vlastní HID zařízení dle nastaveného módu. Díky tomu není nutný žádný dodatečný software a macropad funguje ihned po připojení ke kterémukoliv počítači s využitím předinstalovaných HID ovladačů.

Problémem tohoto řešení je skutečnost, že mé zařízení by bylo omezeno na funkce standardní klávesnice, myši a podobných zařízení. Kvůli tomu by všechny funkce macropadu musely být realizovatelné pomocí kláves klasické klávesnice nebo tlačítek myši. Příkladem může být jednoduché spuštění aplikace. Pro provedení této funkce musí macropad nejprve emulovat stisknutí klávesové zkratky *Win + R*, která na operačním systému Windows otevře okno *Spusit*. Následně musí automaticky vypsát cestu k programu, který chceme spustit a stisknout klávesu *Enter*. Toto řešení se mi osobně jeví jako dosti nepraktické. Další nevýhodou by byla nutnost ošetření problémů spojených s nastaveným rozložením klávesnice, které byly popsány v podkapitole 2.3.

Způsobem, jak vyřešit tyto problémy a zároveň umožnit macropadu efektivně vykonávat další komplikovanější funkce, je vytvoření vlastního počítačového softwaru. Pro tento účel je třeba, aby macropad neemuloval klávesnici, ale aby do počítače pouze odesílal data informující o použití jednotlivých kláves nebo enkodérů. Tato data program zpracuje a na jejich základě vykoná odpovídající funkce.

Čip CH9329 a HID standard umožňují přenos vlastních dat, jejichž účelem není emulovat HID zařízení. Implementace tohoto přenosu by však byla komplikovanější a je třeba na místě hledat jiné řešení.

4.2.2 Řešení pomocí sériového portu

Z výše uvedených důvodů jsem se nakonec rozhodl využít velmi populární integrovaný obvod FT232RL [12] od společnosti FTDI. Ten stejně jako CH9329 zpracovává UART komunikaci z mikrokontroléru a odesílá ji do počítače v souladu se standardem USB. Na rozdíl od předchozího čipu je však FT232RL počítačem rozpoznán jako virtuální sériový port a poskytuje tak jednoduchý způsob, jak implementovat obousměrnou sériovou komunikaci.

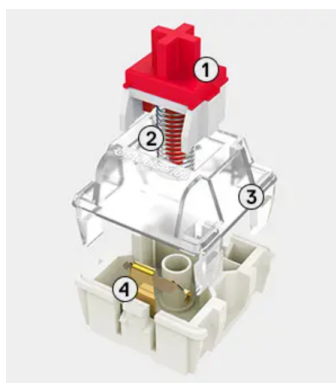
Program, který bude spuštěný v počítači na pozadí, se připojí na příslušnou sériovou linku, bude z klávesnice číst data a vykonávat požadované funkce. Ve výsledku půjde pouze o rozšíření již plánované aplikace pro konfiguraci zařízení. Komunikaci po sériové lince lze na počítači jednoduše implementovat například pomocí programovacího jazyka Python a pro něj již dostupných modulů. Python zároveň poskytuje v podstatě neomezené možnosti z hlediska funkcí, které mohou být při stisku klávesy vykonány, díky čemuž je tato metoda řešení velmi flexibilní.

Další výhodou použití virtuálního sériového portu je jednoduchá komunikace z počítače do macropadu. Ta je potřebná pro nastavení macropadu v případě, že nechceme využít SD karty nebo obdobného média pro přenos konfigurace.

V neposlední řadě bude také připojení klávesnice pomocí sériové linky velmi užitečné při jejím programování s využitím zavaděče.

4.3 Mechanické spínače

Mechanické spínače [13] jsou součástky, které se používají jako tlačítka v tzv. mechanických klávesnicích. Tyto součástky jsou umístěny pod jednotlivými klávesami, přičemž klávesou je v tomto případě myšlen vyměnitelný kryt tlačítka s popiskem. Vnitřní mechanismus spínače je k vidění na obrázku 4. Každý spínač v sobě obsahuje pružinku, která klade odpor při jeho stisku a následně jej vrací do původní polohy. Při stisku dochází uvnitř tlačítka ke stlačení pružinky a dotyku vodivých kontaktů, který je zaregistrován řídicím obvodem klávesnice.

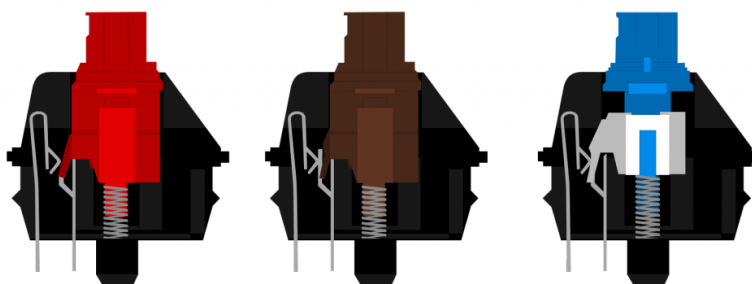


1. Stopka pro připevnění klávesy
2. Kovová pružinka
3. Kryt spínače
4. Kovové kontakty

Obrázek 4: Vnitřní mechanismus spínače [14]

4.3.1 Typy spínačů

Podle [15] existují tři základní typy mechanických spínačů, jejichž řez je zobrazen na obrázku 5.



Obrázek 5: Typy mechanických spínačů – lineární, taktilní a klikavé [16]

Lineární spínače jsou tím nejjednodušším typem a vyznačují se tím, že stisk klávesy je naprosto plynulý. Neposkytují uživateli žádnou zpětnou vazbu v momentě aktivace tlačítka, ke které dochází většinou přibližně v půlce hloubky stisku. Druhým typem jsou spínače taktilní, které již poskytují hmatovou odezvu. Při stisku tlačítka je v aktivačním

bodě cítit zvýšený odpor a tlačítko se v podstatě velmi nepatrně zasekne, což informuje uživatele o registraci stisku. Nejkomplikovanější konstrukcí disponují klikavé spínače, které fungují stejně jako ty taktilní, navíc je u nich však v okamžiku aktivace tlačítka slyšet výrazné kliknutí. To je způsobeno pohyblivou součástí uvnitř spínače (bílá část modrého spínače na obrázku 5), která v okamžiku registrace stisku naráží do pouzdra součástky.

Dále lze mechanické spínače rozlišovat podle několika parametrů [13]. Mezi ně patří například síla potřebná k aktivaci tlačítka, maximální hloubka stisku nebo již zmíněný aktivací bod, což je hloubka stisku, při které dojde k sepnutí spínače.

U většiny výrobců jednotlivým typům spínačů odpovídají různé barvy. V případě populárního výrobce Cherry [17] odpovídá lineárním spínačům červená, taktilním hnědá a klikavým modrá jako je tomu na obrázku 5. Mimo to existuje i mnoho dalších modelů, které se od základních liší ve zmíněných parametrech a barevném označení.

4.3.2 Srovnání s membránovými klávesnicemi

S membránovými klávesnicemi se můžeme setkat všude kolem nás, zejména pokud se jedná o levnější počítačové klávesnice nebo klávesnice v noteboocích. Podle [18] jsou membránové klávesnice tvořeny několika vrstvami. Vrchní vrstva je flexibilní a stiskem klávesy dochází k jejímu promáčknutí. V důsledku toho v klávesnici vznikne vodivý kontakt a stisk je zaregistrován.

Ve srovnání s membránovými nabízí mechanické klávesnice lepší odezvu, přesnost a delší životnost. Zároveň díky tomu, že je na trhu dostupné velké množství různých druhů mechanických spínačů, může si každý najít model, který mu bude nejvíce vyhovovat. Zejména z těchto důvodů jsem se rozhodl mechanické spínače ve svém zařízení použít.

Na druhou stranu nevýhodami mechanických spínačů jsou především jejich hlučnost a vyšší cena. Tyto vlastnosti však osobně nepovažuji za kritické.

4.3.3 Hot-swappable patice

Významným doplňkem k mechanickým spínačům jsou hot-swappable patice (obrázek 6). Tyto patice se připájí na desku plošných spojů a následně do nich lze spínače jednoduše zasunout. Díky tomu není třeba jednotlivé spínače pájet a lze je rychle vyměnit, ať už z důvodu poruchy nebo při přechodu na jiný typ.



Obrázek 6: Kailh hot-swappable patice [19]

4.3.4 Zvolené spínače

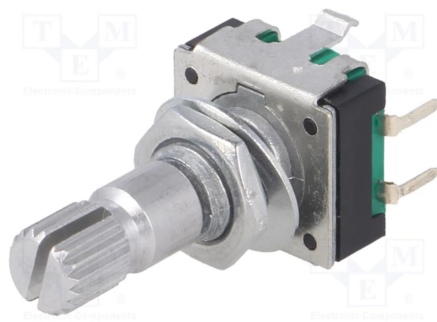
Pro mojí klávesnici jsem se rozhodl použít spínače od společnosti Gateron a to konkrétně typ G Pro 2.0 Brown [20]. Jedná se o hnědé taktilní spínače, jenž mají na spodní straně tři piny – dva kovové, které budou zasunuty do patice na DPS a jeden plastový určený pro usazení do DPS. Tato řada disponuje průhledným pouzdem, které poskytuje dobrý průchod světla z RGB LED, jež mohou být umístěny pod jednotlivé spínače. Zároveň jsou již z továrny tyto spínače uvnitř nalubrikované pro poskytnutí plynulejšího stisku.

Spínače doplním o patice značky Kailh [19] a černé klávesy bez popisků vyrobené z PBT plastu.

Důvodem pro výběr těchto součástek byla především jejich dobrá dostupnost a příznivá cena na asijských e-shopech. Zároveň již vlastním klávesnici s lineárními spínači značky Cherry a stavbu vlastního macropadu vnímám jako příležitost vyzkoušet nový typ spínačů.

4.4 Rotační enkodér

Rotační enkodér je elektromechanická součástka, která převádí informaci o úhlu natočení nebo rotačním pohybu hřídele na analogový nebo digitální signál. Jeho vlastnosti jsou zde popsány na základě informací ze zdrojů [21, 22, 23].



Obrázek 7: Mechanický inkrementální rotační enkodér [24]

4.4.1 Typy enkodérů

Existují dva základní typy rotačních enkodérů. Prvním jsou absolutní rotační enkodéry, které informují o absolutní poloze hřídele, tj. úhlu jejího natočení. Druhým typem jsou rotační enkodéry inkrementální, jejichž dva výstupní signály nesou informaci o směru a rychlosti rotace hřídele, nikoliv však o její absolutní poloze. Na tento typ se dále zaměřím, jelikož je použit na dříve popsaném zařízení od společnost Adafruit a zároveň se jeví jako vhodný pro mojí aplikaci. Tato součástka se běžně využívá jako vstupní zařízení, jehož rotace snižuje nebo zvyšuje hodnotu zvoleného parametru. V případě macropadu může jít například o hlasitost počítače.

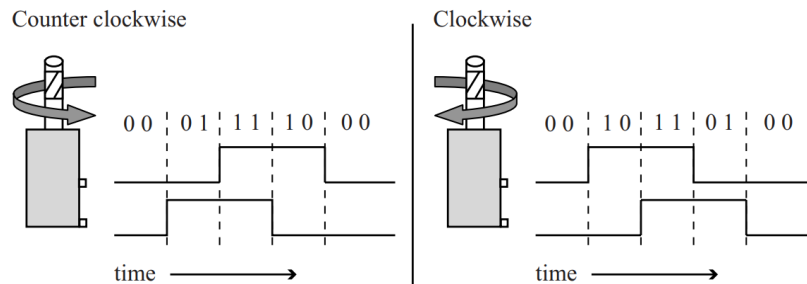
Za zmínku stojí i existence různých typů enkodérů z hlediska principu, který jejich vnitřní mechanismus využívá. Nejběžnějším typem jsou mechanické, dále však existují i optické či magnetické. Výhodou mechanických enkodérů je jejich nízká cena, avšak té

odpovídá i jejich nízká životnost a problémy spojené s nedokonalostí vnitřních vodivých kontaktů.

Příklad mechanického inkrementálního rotačního enkodéru je k vidění na obrázku 7.

4.4.2 Mechanismus enkodéru

Vnitřní mechanismus inkrementálního rotačního enkodéru by šlo popsat jako dvě tlačítka, které se díky rotaci hřídele spínají a rozepínají. V důsledku toho výstupní signály enkodéru alternují mezi stavy logické 1 a 0, jak je vidět na obrázku 8. Signály jsou vždy navzájem o 90 stupňů posunuté a na základě toho, který z nich změni logický stav dříve, lze určit směr otáčení hřídele. Výstupem enkodéru jsou sekvence Grayova kódu 00, 10, 11, 01 nebo 00, 01, 11, 10 v závislosti na směru rotace. Grayův kód [25] se vyznačuje tím, že sousední kombinace bitů se vždy liší pouze v jednom bitu z důvodu snížení výskytu chyby při přechodu mezi stavy. Průběh signálů zobrazený na obrázku 8 je měřitelný při každém kroku rotace hřídele, přičemž počet kroků na celou otáčku je jedním z hlavních parametrů rotačních enkodérů.



Obrázek 8: Výstupní signály inkrementálního rotačního enkodéru [22]

Součástí enkodéru může být také dodatečné tlačítko, které je sepnuto při stlačení hřídele.

4.4.3 Zvolené enkodéry

Pro mnou navrhované zařízení jsem se rozhodl použít dva mechanické inkrementální rotační enkodéry EC11 s tlačítkem. Oba disponují 20 kroky na otáčku, 15 mm dlouhou hřídelí a 5 piny. Dva piny tvoří tlačítko a zbylé tři jsou určeny pro snímání rotace enkodéru. Zároveň jsem k oběma enkodérům pořídil hliníkové hlavice, které se nasadí na hřídele pro pohodlnější úchop.

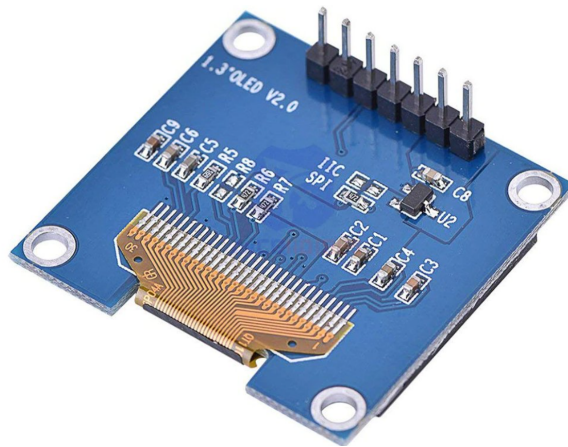
Důvodem pro využití těchto součástí byla opět jejich rozšířenost a nízká cena na asijských e-shopech.

4.5 Displej

Další důležitou součástí mojí programovatelné klávesnice bude displej, jehož účelem bude zobrazování funkcí přiřazených jednotlivým klávesám. Implementace displeje značně zpříjemní používání macropadu, jelikož nebude nutné, aby si uživatel pamatoval všechny funkce, které individuální tlačítka vykonávají.

Po vzoru zařízení Adafruit a DuckyPad jsem se rozhodl použít hojně dostupný 1,3" displej využívající technologii OLED. Podle [26] tato technologie ve srovnání s široce rozšířenou technologií LCD nabízí lepší kontrast, vyšší jas, nižší odezvu a širší pozorovací úhly. Zároveň panely typu OLED nevyžadují na rozdíl od LCD systém podsvícení, díky čemuž mohou být tenčí a lehčí.

Zmíněný displej jsem pořídil ve formě modulu, který je vidět na obrázku 9. Na malé DPS je napájen OLED panel spolu se všemi potřebnými součástkami a připojení probíhá prostřednictvím standardní kolíkové lišty. Díky tomu není třeba řešit zapojení samotného displeje a lze ho jednoduše připájet. Použitý panel disponuje rozlišením 128 x 64 pixelů a je pouze černobílý, což však bude naprosto dostačující pro zobrazování popisků kláves. K dispozici je i levnější 0,96" palcová varianta se stejným rozlišením, její menší rozměr by však pravděpodobně vedl k horší čitelnosti.



Obrázek 9: Modul použitého OLED displeje [27]

Komunikace displeje s mikrokontrolérem je realizována řadičem SH1106 [28]. Připojení k SH1106 lze na základě zapojení realizovat pomocí sběrnic I2C nebo SPI. Já jsem zvolil modul určený k připojení po SPI, jelikož se jeví jako vhodnější pro moji aplikaci. Podle [25] SPI sběrnice sice vyžaduje pro svojí funkci více vodičů, ale při komunikaci nedochází k přenosu adresy cílového zařízení jako je tomu u I2C. Důsledkem toho je možnost dosahovat vyšších přenosových rychlostí a jednodušší implementace zejména v případě připojení pouze jedné periferie.

4.6 LED podsvícení

Jako zdroj podsvícení kláves mého macropadu jsem se rozhodl využít barevné LED SK6812MINI-E [29]. Ty jsou použity na zmíněném zařízení DuckyPad a pravděpodobně i na klávesnici od společnosti Adafruit.

Každá z těchto součástek integruje červenou, zelenou a modrou LED a obvod, který je řídí pomocí PWM. Zapojení probíhá pomocí čtyř pinů – dva jsou napájecí a zbylé dva jsou určeny pro příjem a odesílání dat. Nastavení barvy probíhá pomocí 24 bitů, přičemž každé ze tří vnitřních LED odpovídá 8 bitů. Díky tomu je možné každé základní barvě nastavit 256 různých úrovní intenzity a získat tak libovolnou výslednou barvu.

Výraznou předností těchto LED je možnost řetězit je a ovládat jich větší množství pouze jedním pinem mikrokontroléru. To je možné díky schopnosti každé součástky přeposílat data. Například pro ovládání dvanácti LED je třeba do řetězce odeslat 12 x 24 bitů. První LED v řetězci zpracuje prvních 24 bitů, odpovídajícím způsobem nastaví vlastní barvu a zbylých 11 x 24 bitů přepoše dále. Tento proces se následně opakuje, dokud nejsou nastaveny všechny LED v řetězci.

Dalšími výhodami SK6812MINI-E jsou jejich malé rozměry a především provedení pájecích kontaktů. Na rozdíl od ostatních LED obvodů nabízejících podobnou funkcionality nemají tyto umístěny kontakty na spodní straně pouzdra ale po jeho stranách. V důsledku toho je možné připájet je z druhé strany DPS, než jsou umístěny klávesy a nechat světlo procházet otvory v desce. Díky tomu nehrozí případná kolize se spínači, pod kterými mají být LED umístěny.

Kapitola 5

Použitá komunikační rozhraní

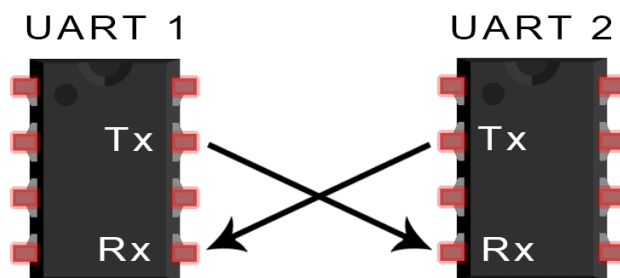
5.1 UART

UART je modul, který mikrokontroléry využívají pro sériovou asynchronní komunikaci. V mém zařízení bude použit pro výměnu dat mezi mikrokontrolérem a převodníkem FT232RL, který zprostředkovává komunikaci s počítačem.

Informace pro tuto podkapitulu byly čerpány z publikací [22, 30].

5.1.1 Zapojení

Základní zapojení UART komunikace vyžaduje pouze tři vodiče – dva datové a zem. Při propojování dvou zařízení je zejména důležité, aby jejich piny TX (vysílací) a RX (příjímací) byly propojené křížem jako je tomu na obrázku 10. Díky přítomnosti dvou datových vodičů je UART schopen realizovat full-duplex komunikaci, při které může docházet k přenosu dat v obou směrech zároveň.

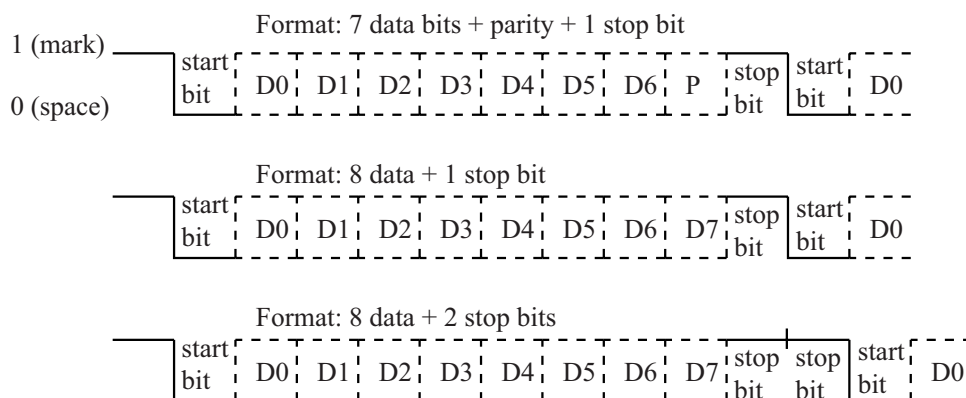


Obrázek 10: Propojení UART modulů [31]

Některé mikrokontroléry disponují pokročilejším modulem USART, který je schopen i synchronní komunikace. Pro tu však musí být přenášen synchronizační hodinový signál.

5.1.2 Formát dat

Pro UART modulem realizovanou asynchronní komunikaci je využíváno datových rámců, jejichž varianty jsou zobrazeny na obrázku 11. Před začátkem komunikace se linka nachází ve stavu logické 1. Začátek přenosu je indikován start bitem, kdy se linka dostane do stavu logické 0. Následuje osm datových bitů, které se posílají od nejméně významného (LSb) po nejvýznamnější (MSb). Přenos rámce je ukončen posláním jednoho nebo dvou stop bitů. Poslední osmý datový bit rámce může být případně nahrazen sudým nebo lichým parity bitem. Ten je vysílačem automaticky nastaven tak, aby celkový počet jedniček v rámci byl sudý, resp. lichý a příjemci tak může posloužit k detekci chyb vzniklých při přenosu.



Obrázek 11: Varianty datových rámců pro asynchronní komunikaci [22]

5.1.3 Modulační rychlost

Důležitým parametrem UART modulu je modulační rychlost neboli baud rate. Tu je pro správný přenos nutné korektně nastavit na obou zařízeních, která spolu mají komunikovat. Modulační rychlost je definována jako počet signálových prvků za sekundu. V našem případě sériové komunikace, kdy je vždy přenášen pouze jeden bit naráz, je modulační rychlost rovna přenosové rychlosti. Platí tedy vztah

$$v_{data} = v_{mod} = \frac{1}{t_b}, \quad (\text{bit/s, Bd, s}) \quad (1)$$

kde v_{data} je přenosová rychlost, v_{mod} je modulační rychlost a t_b je doba trvání jednoho bitu.

5.2 SPI

SPI je synchronní sériové rozhraní užívané zejména pro připojení senzorů a jiných periférií k mikrokontrolérům. V naší aplikaci bude použito pro komunikaci mikrokontroléru s displejem.

Zdrojem informací pro tuto podkapitolu byla publikace [25].

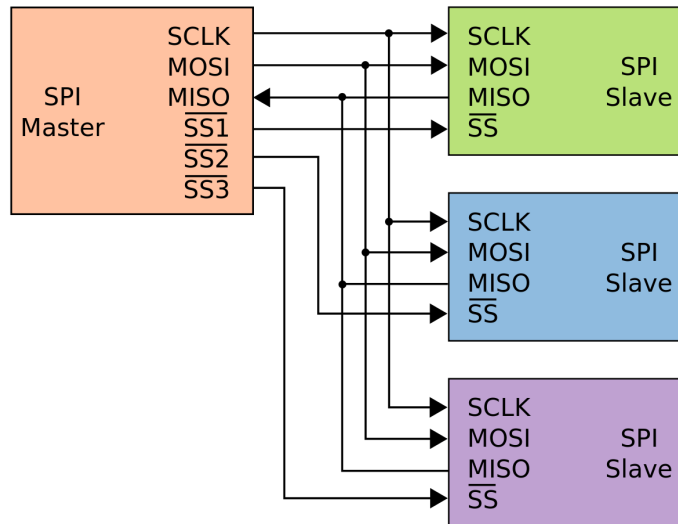
5.2.1 Zapojení

Sběrnice SPI funguje na tzv. principu master-slave, kdy master zařízení (v mém případě mikrokontrolér) řídí veškerý přenos dat.

Typické propojení zařízení pomocí SPI je vidět na obrázku 12. Pro full-duplex komunikaci jsou potřeba tři vodiče. Prvním z nich SCLK, který slouží k synchronizaci zařízení při odesílání a čtení dat. Dále je nutné zapojit vodiče MOSI pro přenos dat z master do slave zařízení a MISO pro přenos v opačném směru. V případě, že je na sběrnici připojeno více slave zařízení je nutné, aby ke každému také vedl samostatný SS (někdy také CS) vodič.

Z obrázku je patrné že datové a hodinové signály jsou sdíleny všemi připojenými zařízeními, a proto je nutné, aby mikrokontrolér řídil, které zařízení má být v daný

moment aktivní. Toho je docíleno pomocí SS vodičů, které jsou v klidu ve stavu logické 1 a pouze v případě, že má dané slave zařízení komunikovat, tak je příslušný SS signál mikrokontrolérem nastaven na 0.



Obrázek 12: Zapojení sběrnice SPI [32]

5.2.2 Formát dat

SPI nemá obecně specifikovaný formát a obsah posílaných dat, tj. kolik bitů je posíláno a jaký mají význam. Z toho důvodu je třeba pečlivě číst dokumentaci připojovaných periférií, která by měla specifikovat, jakým způsobem mají vypadat přenášená data. Zároveň je důležité zjistit, jaká je minimální a maximální frekvence hodinového signálu, která může nabývat hodnot od několika kHz do mnoha MHz.

5.3 USB

USB je v dnešní době široce rozšířená sběrnice pro propojení počítače či jiných zařízení s perifériemi jako jsou klávesnice, myši, tiskárny a mnoho dalších. Právě z těchto důvodů se jeví jako vhodná pro připojení mnou navrhovaného macropadu.

Obsah této kapitoly je založen na informacích z [33].

Existuje několik verzí USB standardu, které se liší především v maximální rychlosti přenosu dat a výkonu, jež jsou schopny připojenému zařízení dodávat. Integrovaný obvod FT232RL [12] od FTDI, který mám v plánu použít pro komunikaci mikrokontroléru s počítačem je kompatibilní s verzí USB 2.0 v režimu Full Speed. Proto se dále zaměřím na verzi 2.0, její provedení a možnosti.

Připojení USB periferie probíhá pomocí kabelů s pěti vodiči, jejichž přehled je v tabulce 2. Novější verze standardu přidávají další vodiče pro dosažení vyšších přenosových rychlostí a výkonů.

Napájení zařízení je zprostředkováno vodiči VBUS a GND a data jsou přenášena ve formě diferenciálního signálu pomocí krouceného páru D+ a D-. V důsledku toho

USB 2.0 realizuje half-duplex komunikaci, tj. v daném okamžiku dochází k přenosu dat v jednom nebo druhém směru, ale ne v obou zároveň.

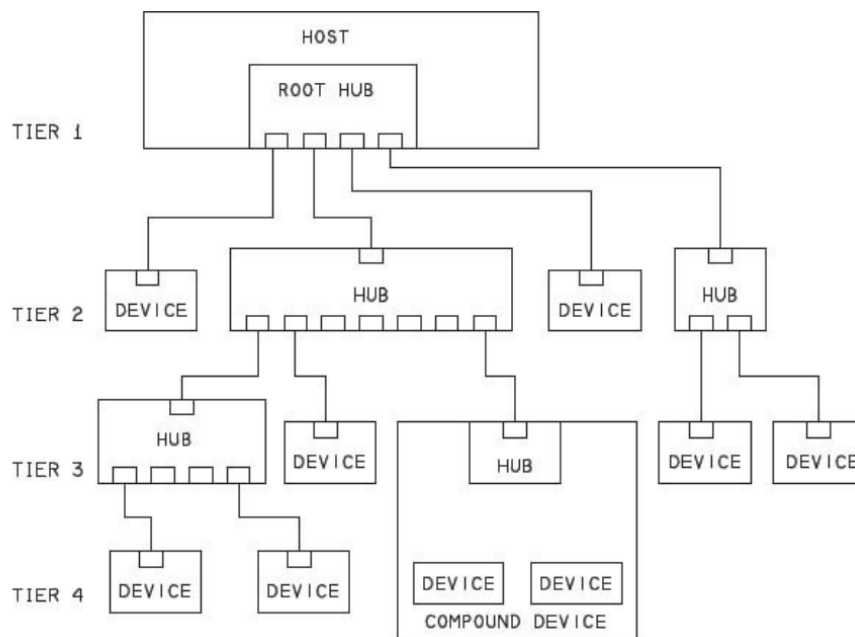
V režimu High Speed je tato sběrnice schopna přenosu rychlostí až 480 Mb/s a napájení proudem až 500 mA při 5 V.

Tabulka 2: Vodiče rozhraní USB 2.0 [33]

Vodič	Název	Význam
1	VBUS	+5 V
2	D-	Záporný diferenční signál
3	D+	Kladný diferenční signál
4	GND	Zem
Shell	Shield	Stínění

5.3.1 Topologie

USB využívá hvězdicové topologie, která je znázorněna na obrázku 13. Centrálním členem je tzv. hostitel (v mém případě počítač) s rozbočovačem, který inicializuje veškerou komunikaci na sběrnici. K hostiteli lze s pomocí dalších rozbočovačů a jejich řetězení připojit až 127 periférií.



Obrázek 13: Topologie sběrnice USB [33]

5.3.2 Enumerace

Po připojení periferie k počítači dochází automaticky k mnoha procesům, které však nebudu všechny popisovat a zmíním pouze ty nejdůležitější.

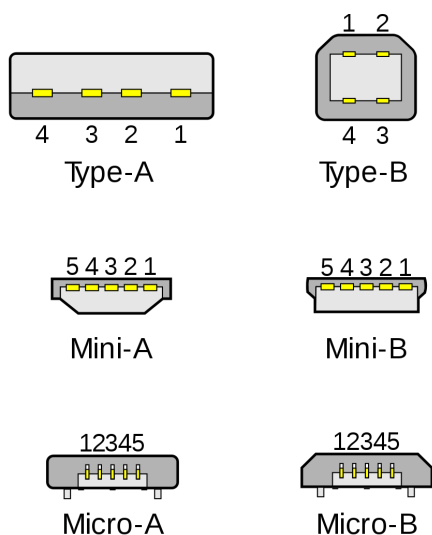
V první řadě je zařízení přidělena unikátní adresa.

Následně si počítač vyžádá tzv. deskriptory. To jsou pevně definované datové struktury uložené v paměti zařízení obsahující informace o jeho vlastnostech a možnostech. Tyto deskriptory mají hierarchickou strukturu. Hlavním povinným deskriptorem je deskriptor zařízení. Ten je vždy pouze jeden a obsahuje základní informace o zařízení jako je sériové číslo, kompatibilní verze USB, ID výrobce a produktu a další. Zároveň je v něm zapsán počet konfigurací zařízení, která každá vyžaduje vlastní deskriptor konfigurace. Těm jsou dále podřazeny deskriptory rozhraní, jenž popisují jednotlivé funkce zařízení a většinou informují o jeho třídě. Mezi tyto třídy patří např. audio/video, tiskárna, úložiště nebo HID, což je třída zahrnující klávesnice, myši a další zařízení určená pro interakci člověka s počítačem. Posledním ze základních deskriptorů jsou deskriptory brány.

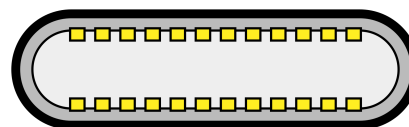
Nakonec je na základě informací z deskriptorů operačním systémem k zařízení přiřazen odpovídající ovladač.

5.3.3 Konektor

Pro standard USB 2.0 byly původně určeny konektory typu A na straně hostitele a různé verze konektoru B na straně zařízení. Tyto jednotlivé konektory jsou vyobrazeny na obrázku 14.



Obrázek 14: Konektory USB 2.0 [35]



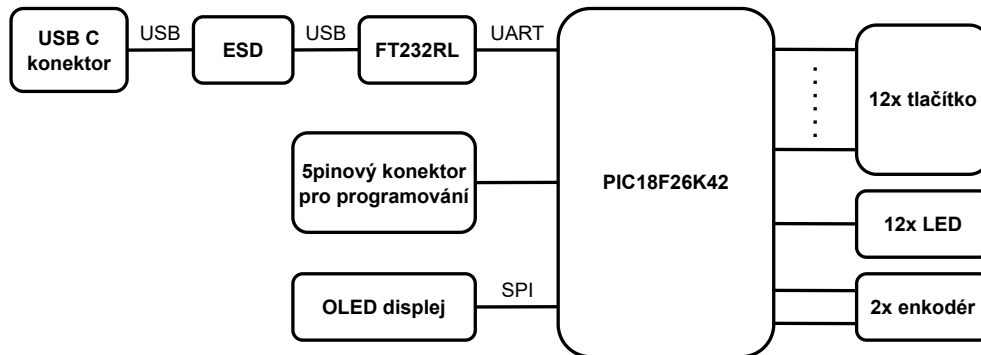
Obrázek 15: Konektor USB C [35]

S dalšími revizemi standardu přišel i dnes již velmi rozšířený konektor typu C [34], který je vidět na obrázku 15. Ten má oproti starším konektorům mnohem více pinů pro podporu novějších verzí USB, lze jej však využít i pro verzi 2.0. Jde o malý, robustní a především oboustranný konektor, což jsou hlavní důvody, proč ho chci ve svém zařízení využít.

Kapitola 6

Návrh zapojení

V této kapitole bych rád blíže popsal schéma zapojení méj klávesnice, pro jehož vytvoření jsem využil zdarma dostupného softwaru KiCad. Celé schéma je k nalezení v příloze B, přičemž zde na obrázku 16 je k vidění jeho velmi zjednodušená verze. Kompletní KiCad projekt je součástí elektronické přílohy.



Obrázek 16: Zjednodušené schéma zapojení mého zařízení

Stěžejní součástí zapojení jsou mikrokontrolér PIC18F26K42 a USB-UART převodník FT232RL, jejichž komunikace probíhá prostřednictvím dvou UART vodičů. Na základě jejich datových listů k nim byly přidány blokové kondenzátory C1-C5. Ty podle [36] slouží zejména jako lokální zdroje energie obvodů, mají pokrývat jejich impulzní spotřebu a předcházet zvlnění napájecího napětí.

Podle datového listu FT232RL byl na napájecí vodič přidán feritový korálek FB1, jehož účelem je redukce elektromagnetického rušení na vedení. Zároveň jsem k mikrokontroléru umístil pětipinový konektor pro připojení programátoru.

6.1 USB

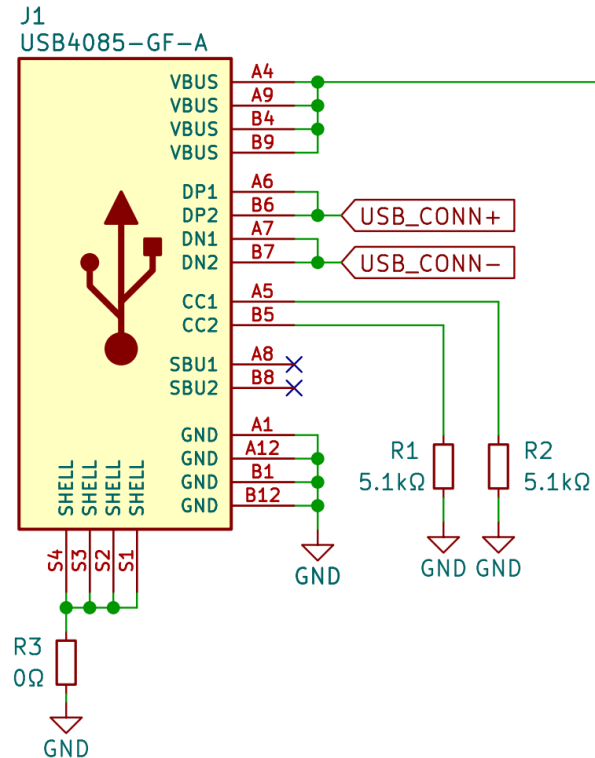
Důležitou komponentou je USB konektor typu C sloužící k připojení klávesnice k počítači. Zároveň je z něj celá klávesnice napájena pomocí 5 V. Zde jsem se rozhodl použít model USB4085 [37] od firmy GCT, který je přímo určený pro USB verze 2.0 a z toho důvodu eliminuje některé piny přítomné na plnohodnotném USB C konektoru. Jeho zapojení je vidět na obrázku 17.

Na základě [34, 38] jsem pro správnou funkci uzemnil oba CC piny konektoru pomocí 5,1k Ω rezistorů a piny SBU nechal plovoucí. Datové piny DP a DN, které se na konektoru nachází dvakrát jsem odpovídajícím způsobem spojil, aby byla zajištěna oboustrannost konektoru. Orientací zapojení kabelu je následně určeno, který ze dvou párů pinů je při přenosu dat aktivní.

Komplikace nastala v momentě, kdy jsem řešil zapojení krytu (SHELL pinu) konektoru. Dostupné zdroje se totiž rozcházejí v tom, zda pin nechat plovoucí nebo jej uzemnit, a to buď přímo nebo s využitím rezistoru. Z toho důvodu jsem se nakonec rozhodl přidat do zapojení rezistor s hodnotou 0 Ω , což mi umožní při realizaci klávesnice

vyzkoušet veškeré tyto možnosti.

Na USB sběrnici jsem zároveň umístil ESD ochranný obvod USBLC6 [39] od firmy STMicroelectronics, jehož účelem je ochrana mého zařízení před elektrostatickým výbojem.



Obrázek 17: Zapojení USB C konektoru

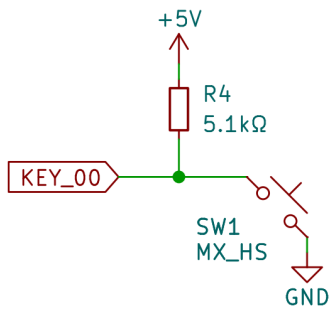
6.2 Pull-up rezistory

Jak je vidět na obrázcích 18 a 19, na vodiče připojující jednotlivé klávesy a enkodéry k mikrokontroléru jsem přidal tzv. pull-up rezistory. Úkolem každého z nich je zajistit jasný stav na vstupním pinu mikrokontroléru v době, kdy není příslušné tlačítko stisknuté. Bez jejich použití by vstupní piny byly plovoucí a nebylo by jasné, v jakém stavu se nachází.

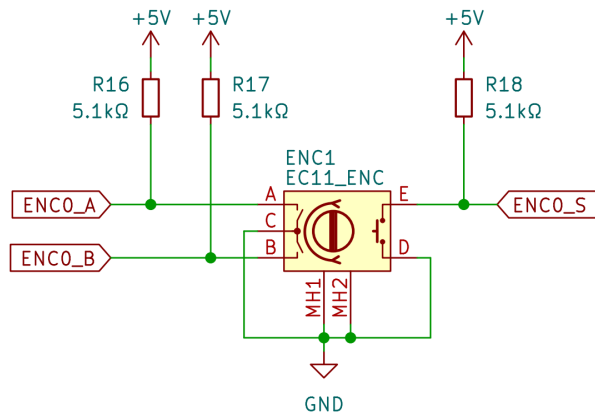
Podle [40] se pro pull-up rezistory běžně používají hodnoty 1-10 kΩ. Z toho důvodu jsem opět využil rezistory s hodnotou 5,1 kΩ, protože jsem je již pořizoval kvůli zapojení USB konektoru a při vlastním otestování bez problému plnily svoji funkci.

6.3 Displej

Displej je k mikrokontroléru připojen pomocí sběrnice SPI, disponuje však pouze jedním datovým pinem SDA a z toho důvodu může displej data pouze přijímat, ale ne je odesílat. Pin SCK je využit k přivedení hodinového signálu a pin RES pro restart displeje. Velice podstatný je kontakt DC, který slouží ke specifikaci toho, zda jsou přijímaná data určena k zobrazení nebo zda se jedná o příkaz určený ke konfiguraci



Obrázek 18: Zapojení spínače



Obrázek 19: Zapojení enkodéru

displeje. Přivedením nízké úrovně napětí na pin CS je displeji dáváno najevo, že má být v daný moment aktivní. Jelikož však není v mém zařízení na SPI sběrnici připojeno žádné další zařízení, rozhodl jsem se tento kontakt pevně připojit na zem.

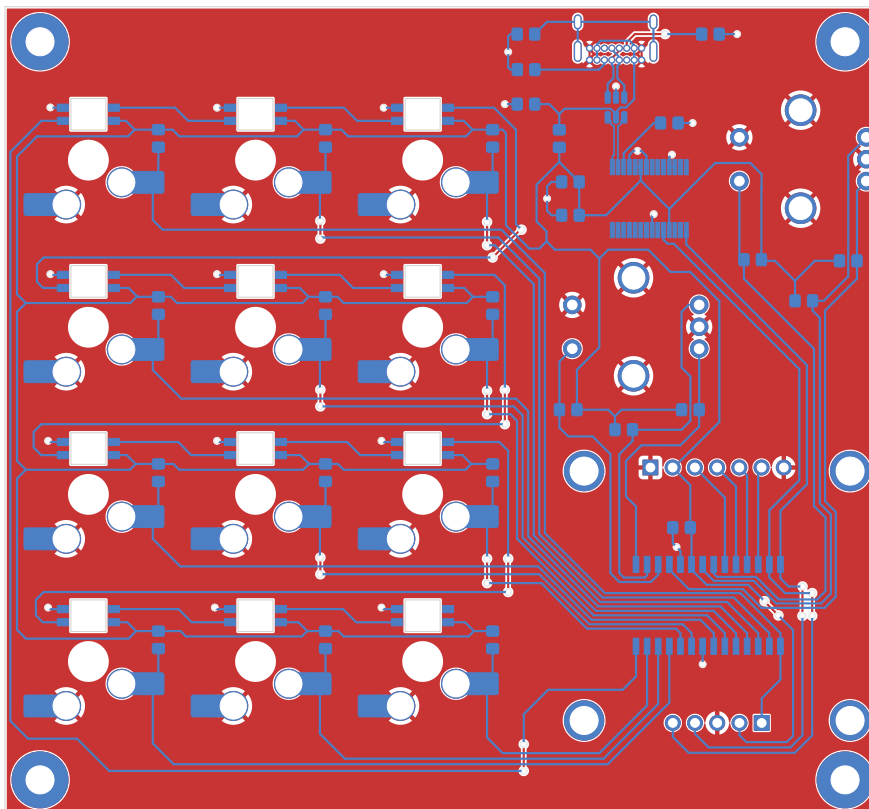
Kapitola 7

Návrh desky plošných spojů

Pro návrh DPS mojí klávesnice jsem stejně jako u schématu zapojení využil program KiCad. Hotový návrh desky je zobrazen na obrázku 20, přičemž pro přehlednost je na něm skryt potisk desky.

Jednotlivým součástkám ve schématu bylo v první řadě přiřadit tzv. footprinty, což jsou rozložení jejich pájecích plošek a poté schéma načíst do editoru DPS. Ten následně zobrazuje, jak mají být jednotlivé součástky na základě schématu propojeny. Většina footprintů, které jsem při návrhu využil byla dostupná na internetu nebo přímo v knihovnách KiCadu. Rozložení pájecích plošek pro enkodéry a displej jsem si však vytvořil sám.

Ke všem součástkám se mi zároveň podařilo opatřit 3D modely, jak je vidět na vizualizaci na obrázku 21. Kompletní 3D model zařízení může být užitečný například při návrhu krytu klávesnice.

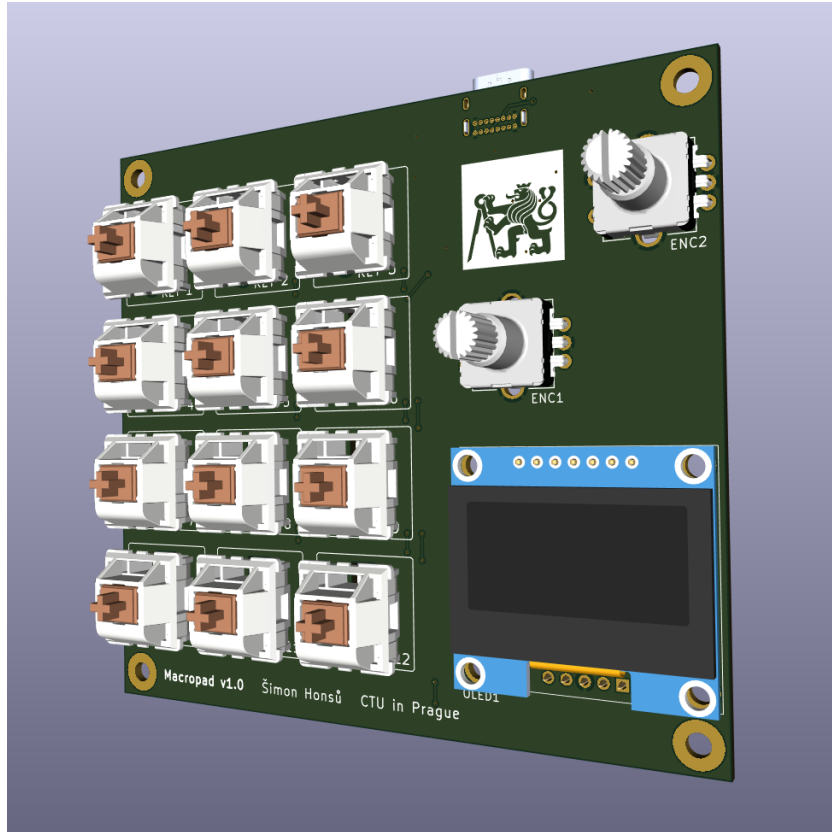


Obrázek 20: Návrh desky plošných spojů klávesnice

Prvním krokem návrhu samotné desky bylo rozmístění jednotlivých součástek. Primárně šlo o polohu kláves, displeje a enkodérů s cílem zajistit co nejpohodlnější používání. Omezením byl v tomto kroku rozměr desky, který nesměl překročit 100 x 100 mm, jelikož by pak její výroba u firmy JLCPCB byla nákladnější.

Vyzkoušel jsem vytvořit několik různých rozložení, nakonec jsem se však rozhodl pro

to na obrázku 21. Jelikož macropad plánuji umístit vedle své stávající klávesnice a ovládat jej levou rukou, jeví se toto rozložení jako nejvýhodnější. Umožňuje jednoduchý přístup ke klávesám a enkodérům a zároveň poskytuje dobrou viditelnost displeje.



Obrázek 21: 3D model macropadu

Mnou navržená deska je dvouvrstvá a podle doporučení v [36] jsem jednu vrstvu použil pro vedení signálů a napájení a druhou pro vodivou plochu připojenou na zem a případné krátké signálové můstky. Veškeré spoje na desce jsem dělal 0,25 mm široké, což by podle online kalkulačky [41] mělo být pro mojí aplikaci naprosto dostačující.

Blokovací kondenzátory jsem se snažil umístit co nejbližší pinům příslušných integrovaných obvodů a dodržet tak doporučení výrobců. Musel jsem však zohlednit to, že bude veškeré součástky potřeba ručně připájet, a proto jsem kondenzátory neumístil do bezprostřední blízkosti pinů, abych předešel komplikacím při pájení. Stejným způsobem jsem na základě datového listu umístil ochranný ESD obvod blízko USB C konektoru pro zajištění nejlepší ochrany.

Dle [42] by USB datové diferenciální vodiče měly mít stejnou délku, využívat co nejméně prokovů a být co nejkratší. Z tohoto důvod jsem pro vodiče D+ a D- využil nástroj pro vedení diferenciálních párů a převodník FT232RL jsem stejně jako ESD ochranu umístil blízko USB konektoru.

Na desku jsem také přidal otvory pro upevnění modulu displeje a do rohů jsem umístil otvory pro případné upevnění desky do krytu nebo připevnění nožiček. Všechny tyto otvory jsou určeny pro standardní metrické šrouby s průměrem závitu 3 mm.

Kapitola 8

Program mikrokontroléru

V této kapitole bude popsán mnou vytvořený program, který specifikuje funkce použitého mikrokontroléru. V případě mé klávesnice se jedná o zpracování vstupů z tlačítek a enkodérů, obsluhu displeje a LED a odesílání dat do počítače.

Pro napsání programu v jazyce C [43] jsem využil software MPLAB X IDE od firmy Microchip ve spojení s kompilátorem XC8 [44]. Použil jsem také rozšíření MPLAB Code Configurator, které uživateli umožňuje generovat části kódu a ulehčuje tak práci.

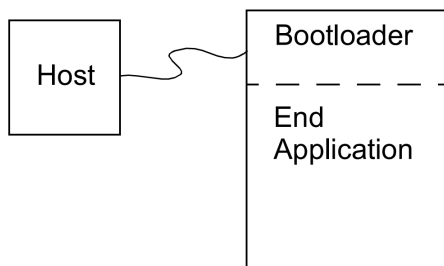
Napsaný program se skládá ze dvou samostatných částí, kterými jsou programový zavaděč a cílová aplikace. Obě budou popsány v následujících podkapitolách, přičemž kompletní zdrojový kód je součástí elektronické přílohy.

8.1 Programový zavaděč

Programový zavaděč neboli bootloader je program, jehož úkolem je správa cílové aplikace v mikrokontroléru. Informace o něm jsem čerpal z [45] a k jeho vygenerování jsem využil zmíněný MPLAB Code Configurator.

V první řadě je potřeba zavaděč nahrát do mikrokontroléru pomocí programátoru. Následně je při každém zapnutí mikrokontroléru nejdříve spuštěn zavaděč, který ověří, zda je v programové paměti přítomna validní cílová aplikace. Způsobů ověření je několik, nicméně jejich popis není pro tuto práci podstatný. V případě, že je přítomna validní aplikace, je jí předána kontrola a mikrokontrolér zahájí standardní funkci. Naopak pokud se v programové paměti nenachází validní aplikace, mikrokontrolér zůstane v režimu zavaděče a čeká na příjem dat.

V tento moment je možné do mikrokontroléru odeslat zkompilovaný binární *.hex* soubor cílového programu, který je zavaděčem zpracován a zapsán do zbylé programové paměti. V mém případě jsem pro příjem dat zvolil UART a pro jejich odesílání hostitelskou počítačovou aplikaci Unified Bootloader od Microchipu. Ta podporuje odesílání na sériový port a přesně z toho důvodu je výhodné, že jsem se rozhodl využít obvod FT232RL, který převádí virtuální USB sériovou linku na UART. Jednotlivé součásti popsaného procesu jsou vyobrazeny na obrázku 22.



Obrázek 22: Programování s pomocí zavaděče [45]

Užitečnou funkcí zavaděče, kterou jsem se rozhodl implementovat je tzv. indikační pin. Pro tento účel jsem využil pin 1, na který je již připojena klávesa. Pokud bude v momentě spuštění mikrokontroléru klávesa stisknuta, tak zařízení setrvá v zavaděči

i v případě, že již byl nahrán platný cílový program. Toto může být užitečné, zejména pokud by bylo třeba cílový program mikrokontroléru aktualizovat. Tímto způsobem lze aktualizace docílit prostřednictvím USB bez použití programátoru.

8.2 Cílová aplikace

Hlavní částí mého programu je cílová aplikace, která popisuje vlastní funkci mikrokontroléru.

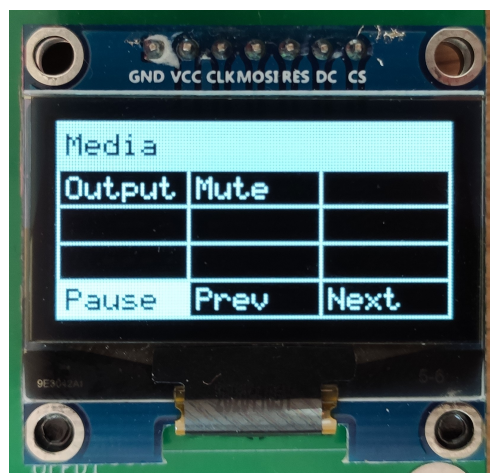
Po spuštění programu dochází nejdříve k nastavení samotného mikrokontroléru a jednotlivých periférií. V první řadě jsou spuštěny funkce pro nastavení pinů a oscilátoru mikrokontroléru. Frekvenci oscilátoru jsem nastavil na maximálních 64 MHz. Následuje inicializace SPI modulu pro ovládání displeje a UART modulu pro komunikaci s počítačem, přičemž jeho modulační rychlost jsem nastavil na 115 200 Bd. V dalším kroku je inicializován samotný displej a z paměti EEPROM jsou načteny jednotlivé profily. Načtena je také informace o tom, který profil má být právě aktivní a ten je zobrazen na displeji. Nakonec je rozsvíceno všech dvanáct RGB LED, které jsem nastavil na statickou červenou barvu.

Po počátečním nastavení je spuštěn nekonečný cyklus, ve kterém dochází ke neustálému snímání vstupů z klávesnice a enkodérů pomocí příslušných funkcí. Do počítače jsou odesílána data pouze oznamující, která klávesa nebo enkodér byly použity a jejich výsledná funkce je vykonána až počítačovou aplikací. Ta bude popsána v kapitole 9.

8.2.1 Obsluha displeje

Pro ovládání displeje prostřednictvím sběrnice SPI jsem vytvořil vlastní knihovnu sestávající se primárně ze souborů *sh1106.h* a *sh1106.c*.

Její základem jsou funkce pro odesílání příkazů a dat k zobrazení do displeje. O jaký typ dat se jedná je rozlišováno pomocí napěťové úrovně na DC pinu displeje. Další součástí jsou také funkce pro inicializaci SPI modulu mikrokontroléru a displeje samotného.



Obrázek 23: Zobrazení aktivního profilu na displeji

Podstatným prvkem knihovny je 128 x 64 bitů dlouhé pole, které reprezentuje jednotlivé pixely na displeji. Do něj jsou pomocí funkce pro vykreslení pixelu zapisována data. Prostřednictvím této funkce je realizováno vykreslení různých tvarů a především textu. Použitý font, který byl převzat z [46] je specifikován v souboru *sh1106_font.c*. Nejdůležitější součástí knihovny je funkce pro vykreslení profilů, která využívá data načtená z paměti EEPROM a vhodným způsobem je zapisuje do pole. V momentě, kdy je pole naplněno daty k zobrazení je celý jeho obsah odeslán do displeje.

Zobrazení jednotlivých profilů na displeji probíhá způsobem, který je k vidění na obrázku 23. V horní části displeje je umístěn název profilu a pod ním jsou v tabulce přehledně zobrazeny popisky dvanácti kláves v souladu s jejich fyzickým rozložením.

Pro poskytnutí zpětné vazby uživateli je při stisku klávesy vždy invertována odpovídající buňka na displeji, jak je také vidět na obrázku 23. Tato funkcionality byla jedním z hlavních důvodů pro využití zmíněného pole reprezentujícího jednotlivé pixely. Pro její realizaci je totiž nutné mít v mikrokontroléru přehled o tom, co je momentálně zobrazeno na displeji, přičemž samotný displej neumožňuje z něj zobrazovaná data vyčíst.

8.2.2 Obsluha kláves

Funkce zprostředkovávající obsluhu kláves je znázorněna diagramem na obrázku 24.

V prvním kroku je do pole načten stav jednotlivých pinů, které na sobě mají připojené spínače a pull-up rezistory. Následně je v cyklu kontrolováno, zda není některý z pinů připojený na zem, k čemuž dochází při stisku tlačítka.

V případě, že je některá z kláves stisknuta následuje tzv. debounce. To je v tomto případě 20ms prodleva, jejímž účelem je eliminace vlivu zákmitů, které vznikají při stisku spínače v důsledku nedokonalosti jeho vnitřních kontaktů. Dále je na displeji invertována buňka odpovídající stisknuté klávese a do počítače jsou prostřednictvím UART modulu odeslána data ve tvaru *K00-K11* informující o stisku dané klávesy. Nakonec je v momentě uvolnění klávesy buňka na displeji invertována zpět do původního stavu a opět následuje 20ms prodleva, jelikož nežádoucí zákmity vznikají i při uvolnění spínače.

8.2.3 Obsluha enkodérů

Pro zpracování vstupů z enkodérů jsem vytvořil algoritmus, který je vidět na obrázku 25.

Hlavní komplikací snímání enkodéru jsou opět zákmity, který vznikají nejenom při jeho stisku, ale i při otáčení hřídele, přičemž příčinou je opět nedokonalost kontaktů ve vnitřním mechanismu. Zákmity vznikají v okolí sestupných a vzestupných hran výstupních signálů enkodéru, které již byly popsány v podkapitole 4.4.2. V mém programu jsou tyto signály reprezentovány proměnnými *A0* a *B0*. Odstranění vlivu zákmitů lze docílit buď pomocí RC filtrů, kterými se signály vyhladí nebo pomocí programového debounce algoritmu. Ve svém zařízení jsem se rozhodl využít druhou z možností a ukázala se jako naprosto dostačující.

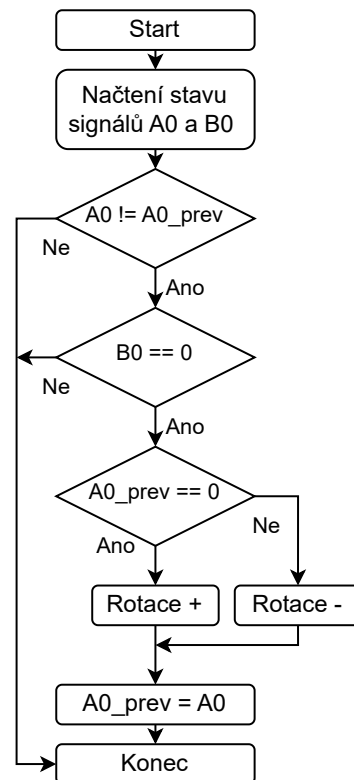
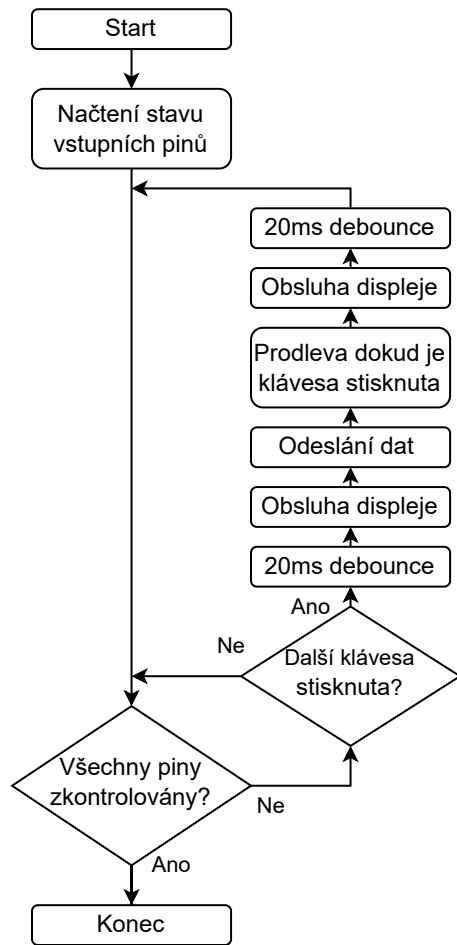
Základem mého algoritmu je porovnávání stavu signálu *A0* s jeho předchozím stavem uloženým v proměnné *A0_prev*. Jelikož jsou na všech výstupech enkodéru připojeny pull-up rezistory je při prvním běhu funkce hodnota *A0_prev* nastavena na 1. Jedná se o statickou proměnnou, tudíž je její hodnota zachována mezi různými voláními funkce.

V případě, že je zaznamenána změna stavu signálu *A0* je následně zkontrolováno, zda je signál *B0* ve stavu 0. Pokud je tato podmínka splněna, lze na základě předchozí

hodnoty signálu $A0$ rozpoznat, jakým směrem je enkodér otáčen. Následně jsou do počítače pomocí UART modulu odeslána data ve tvaru $E0+$ nebo $E0-$ podle směru rotace.

Při stisku tlačítka enkodéru je debounce vyřešen 20ms prodlevou stejně jako u kláves a do počítače je odeslán řetězec $E0S$. Funkce tlačítka enkodéru je programem pevně nastavena na změnu aktivního profilu.

Stejným způsobem je obsluhován i druhý enkodér, využívá však jiné názvy proměnných a do počítače odesílá data ve tvaru $E1+$, $E1-$ nebo $E1S$.



Obrázek 24: Algoritmus obsluhy kláves Obrázek 25: Algoritmus obsluhy enkodérů

8.2.4 Příjem konfigurace

Jednotlivé profily, tj. názvy profilů a kláves, které se mají zobrazit na displeji, jsou do klávesnice odesílány z počítačového softwaru. Komunikace je opět provedena pomocí UART modulu a převodníku FT232RL.

Použitý mikrokontrolér má 1024 bytů EEPROM paměti, což odpovídá 1024 ASCII znakům. Každý profil se skládá z dvanácti znaků pro jeho název a šesti znaků pro popis každé z dvanácti kláves. Dohromady je tedy pro uložení jednoho profilu potřeba 84 bytů.

Z tohoto důvodu jsem stanovil maximální počet profilů klávesnice na deset, což se jeví jako naprosto dostačující.

Příjem konfigurace začíná v momentě, kdy je UART modulem z počítače přijat první byte, který specifikuje aktivní profil. Druhý byte informuje o počtu profilů, které budou nahrány. Následně jsou přijaty jednotlivé profily a veškerá data jsou uložena do EEPROM. V případě využití všech deseti profilů je tedy do paměti zapsáno 842 bytů. Konec příjmu konfigurace je signalizován změnou barvy LED na zelenou a po krátké prodlevě je klávesnice restartována. Po zapnutí je nová konfigurace načtena z paměti a na displeji je zobrazen první profil.

Kapitola 9

Počítačová aplikace

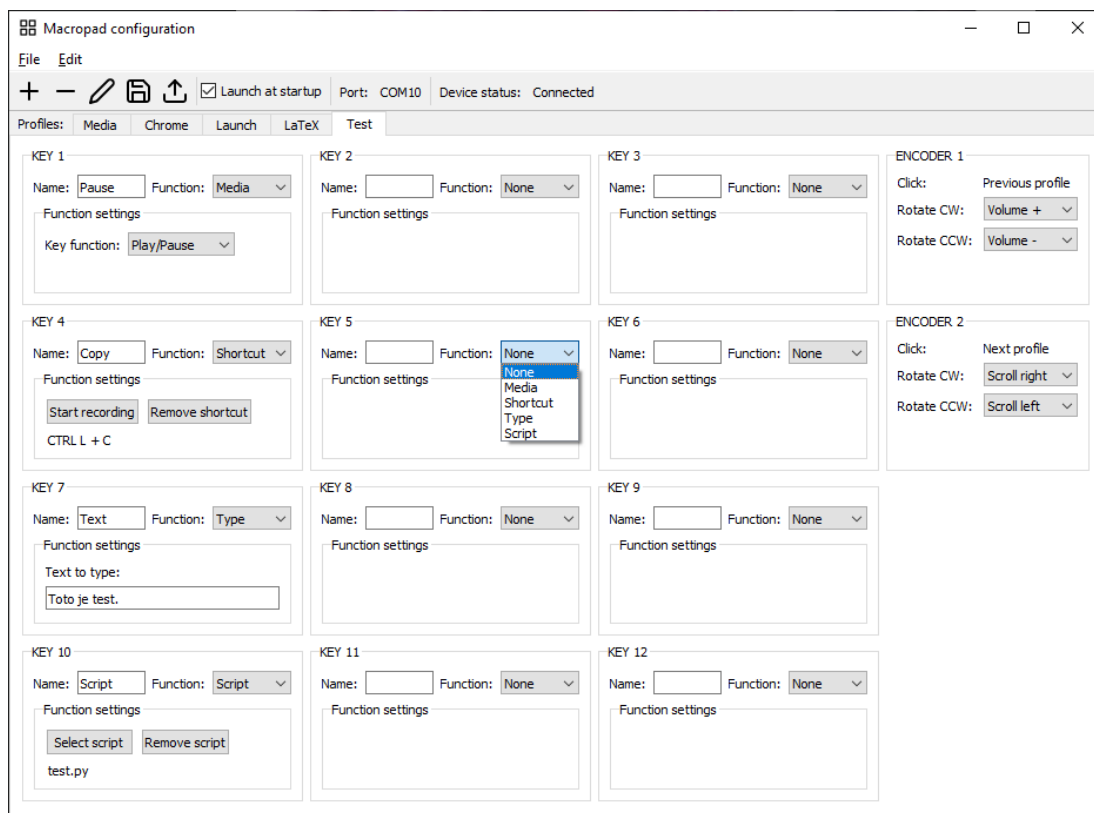
V následující části práce bude představen doprovodný počítačový software, který jsem vytvořil pro svoji klávesnici. Jeho hlavním účelem je konfigurace a zpracovávání vstupů z klávesnice a následné vykonávání uživatelem nastavených funkcí.

Pro napsání aplikace jsem využil programovací jazyk Python. GUI jsem vytvořil s pomocí knihovny PyQt, která umožňuje prostřednictvím Pythonu využít framework Qt [47]. Nakonec jsem pomocí nástroje PyInstaller celou aplikaci zabalil do jediného spustitelného *.exe* souboru určeného pro operační systém Windows.

Veškerý zdrojový kód i výsledná aplikace jsou součástí elektronické přílohy práce.

9.1 Grafické rozhraní

Rozhraní aplikace je tvořeno pouze jedním oknem, které je vidět na obrázku 26. Veškeré použité ikon jsem převzal z webu [48], přičemž hlavní ikoně samotné aplikace jsem doplnil bílé pozadí pro lepší viditelnost.



Obrázek 26: Grafické rozhraní počítačové aplikace

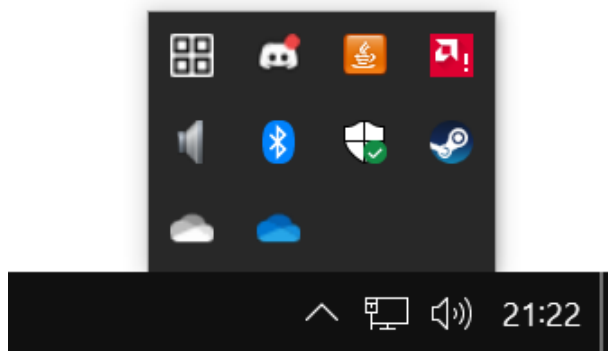
V horní části okna se nachází nástrojová lišta, která obsahuje tlačítka pro všechny důležité funkce. Mezi ně patří přidání, odebrání a přejmenování profilů a dále uložení konfigurace nebo její nahrání do klávesnice. V liště je také zobrazen sériový port, na

kterém je klávesnice připojena a její momentální stav. Poslední součástí nástrojové lišty je zaškrťávací políčko, jehož zaškrtnutí způsobí, že se aplikace spouští automaticky po zapnutí počítače. Toho je docíleno přidáním záznamu do Windows registrů.

Převážná část okna aplikace je tvořena záložkami, které reprezentují jednotlivé profily. Jak již bylo zmíněno v podkapitole 8.2.4, maximální počet profilů je nastaven na deset. Každý profil má svůj název a uvnitř jeho záložky jsou přehledně zobrazena nastavení jednotlivých kláves a enkodérů. Každé klávese lze také přiřadit název a po výběru její funkce z menu je zobrazeno nastavení dané funkce. Enkodérům lze nastavit odlišné funkce pro otáčení po a proti směru hodinových ručiček, jejich stisk je však pevně určen pro změnu aktivního profilu.

Všechny profily jsou při stisku tlačítka pro uložení nebo ve chvíli vypínání aplikace uloženy včetně všech nastavených názvů a funkcí. Pro uložení veškerých dat je využíváno registrů Windows, přičemž při spouštění aplikace jsou z nich profily opět načteny.

Důležitou vlastností aplikace je to, že je spouštěna na pozadí. Díky tomu není uživatel aplikací rušen v případě, že nechce klávesnici konfigurovat. Pro otevření okna je potřeba kliknout na malou ikonu, která je standardně umístěna na hlavní liště v pravém dolním rohu obrazovky, jak je vidět na obrázku 27.



Obrázek 27: Umístění ikony aplikace na hlavní liště Windows

Naopak k úplnému ukončení aplikace je třeba využít tlačítko *Exit* v menu *File*. Po stisknutí křížku v pravém horním rohu okna totiž aplikace není ukončena, ale je opět skryta do pozadí.

9.2 Funkce kláves a enkodérů

Jednotlivé nastavené funkce jsou aplikací vykonány v momentě, kdy jsou z klávesnice přijata data oznamující použití odpovídající klávesy nebo enkodéru. Formát těchto dat byl popsán v podkapitolách 8.2.2 a 8.2.3. Pro komunikaci s klávesnicí po sériovém portu jsem využil Python modul `Pyserial` [49].

Základními funkcemi, které lze klávesám přiřadit je ovládání médií (*Media*), stisk klávesové zkratky (*Shortcut*) a výpis nastaveného textu (*Type*). Pro implementaci těchto funkcí jsem využil modul `Pynput` [50], který umožňuje kontrolu a monitorování klávesnice a myši. Další dostupnou funkcí je spouštění vlastního Python skriptu (*Custom*

script), kterou jsem realizoval pomocí modulu Runpy [51]. Nastavení jednotlivých funkcí je vidět na obrázku 26, přičemž samozřejmostí je i možnost klávesy ponechat neaktivní a nepřisuzovat jim žádné funkce.

Mezi možnosti ovládání médií patří přehrávání a pozastavení spuštěného média, přeskokování na následující nebo předchozí skladbu a ovládání hlasitosti včetně možnosti úplného ztlumení.

Funkce *Shortcut* umožňuje zaznamenat libovolnou klávesovou zkratku a následně jí vyvolávat stiskem jediné klávesy na programovatelné klávesnici. Proces nahrávání je velice jednoduchý. Stačí kliknutím na tlačítko *Start recording* spustit záznam a požadovanou zkratku stisknout. Po uvolnění všech kláves je záznam automaticky ukončen a funkce je připravena k použití.

Nastavení výpisu textu je obdobně jednoduché. Do textového pole je nutné pouze zadat požadovaný text a ten je následně při stisku klávesy celý vypsán do textového pole, které je na počítači právě aktivní. Tato funkce může být užitečná například k vypisování e-mailové adresy nebo jiného často používaného textu.

Poslední dostupná funkce *Custom script* umožňuje uživateli stiskem klávesy spouštět libovolný Python skript. Pro její nastavení je třeba skript pouze zvolit. Toho je docíleno pomocí standardního dialogového okna pro výběr souboru, které je otevřeno po stisknutí tlačítka *Select script*. Okno umožňuje zvolit pouze Python soubory s příponou *.py*. Tato funkce je velmi flexibilní a může být vhodná například pro spouštění programů na počítači nebo jiné mnohem komplikovanější úkoly. Jediným problémem, na který je třeba dát si pozor, je to, aby daný skript netrval příliš dlouho nebo neuvízl v cyklu. Aplikace totiž skript nespouští v dalším vlákne a z toho důvodu při jeho průběhu nereaguje.

V neposlední řadě lze přiřadit funkce i rotaci enkodérů. Mezi ně momentálně patří pouze ovládání hlasitosti a vertikální nebo horizontální rolování.

9.3 Detekce klávesnice

Při spuštění aplikace je v nástrojové liště stav klávesnice v první řadě nastaven na *Disconnected* a sériový port na *None*. Následně je v dalším vlákne spuštěna funkce, která klávesnici vyhledává.

Vyhledávání probíhá tím způsobem, že jsou pomocí modulu Pyserial v intervalu 0,1 s načteny všechny aktivní sériové porty a jsou kontrolována ID výrobce, ID produktu a sériové číslo připojených zařízení. Pokud tyto hodnoty odpovídají použitému čipu FT232RL, tak je spuštěno další vlákno, ve kterém se aplikace na odpovídající sériový port připojí. Proces připojování je signalizován změnou stavu klávesnice na *Connecting...* a zobrazením příslušného portu. V momentě úspěšného připojení je stav nastaven na *Connected*. Následně v tomto vlákne aplikace čeká na data z klávesnice a případně vykonává požadované funkce.

Pokud je kdykoliv při běhu aplikace klávesnice odpojena, dojde k chybě při čtení ze sériového portu. V tu chvíli je ukončeno vlákno obsluhující sériový port, stav klávesnice je nastaven na *Disconnected* a je opět spuštěno vlákno, které skenuje aktivní porty a čeká na připojení klávesnice.

Díky těmto procesům je možné zařízení libovolně připojovat či odpojovat a počítačová aplikace je schopna se na něj vždy automaticky připojit.

9.4 Nahrávání konfigurace

Nahrávání konfigurace do klávesnice je zahájeno při stisku klávesy pro nahrávání v nástrojové liště. V tu chvíli je ve vláknech obsluhujícím klávesnici spuštěna funkce, která realizuje odesílání dat. Její start je signalizován zobrazením *Uploading...* u stavu klávesnice.

V první řadě je odeslán znak 0 popisující aktivní profil. Proměnná udržující informaci o aktivním profilu v aplikaci je následně také nastavena na 0, čímž je docíleno synchronizace klávesnice a aplikace z hlediska aktivního profilu. Dále je odeslán znak značící celkový počet profilů a nakonec jsou nahrány samotné profily. Jak již bylo popsáno v podkapitole 8.2.4, každý z profilů se skládá z dvanácti znaků pro jeho název a z šesti znaků pro popis každé klávesy, což celkem odpovídá 84 bytům.

Do klávesnice tedy nejsou odesílána žádná data popisující funkce kláves nebo enkodérů. Z toho důvodu není třeba provádět nahrávání konfigurace v případě, že je v aplikaci změněna pouze některá z funkcí. Naopak je konfiguraci žádoucí nahrát v momentě, kdy je změněn některý z popisků nebo je upraven počet, popř. pořadí profilů.

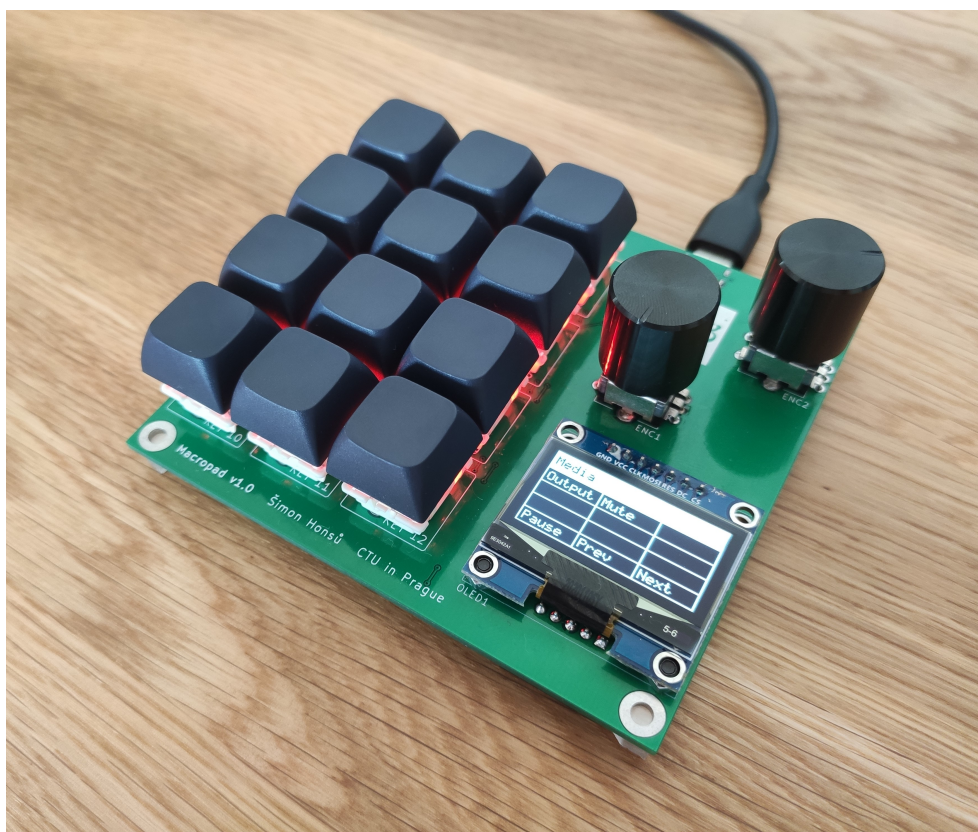
Kapitola 10

Realizace zařízení

K fyzické realizaci mojí klávesnice jsem využil vybavení dostupné ve školní IoT laboratoři.

V první řadě bylo nutné odpájet čip FT232RL z malé desky plošných spojů, na které byl z důvodu nízké ceny zakoupen. K tomu jsem použil horkovzdušnou pájecí stanicí. Následně jsem většinu součástek osadil ručně pomocí standardní páječky, přičemž k osazení integrovaných obvodů jsem opět využil horkovzdušnou pájecí stanicí. USB konektor, jehož zapojení bylo popsáno v podkapitole 6.1, jsem se prozatím rozhodl nechat plovoucí. Do rohů desky jsem nalepil provizorní gumové nožičky a pod modul displeje jsem přidal šroubky a matice pro jeho upevnění. Fotografie osazené desky jsou dostupné v příloze C. Nakonec jsem mechanické spínače s klávesami zasunul do patič a na enkodéry umístil hliníkové hlavičky.

Program mikrokontroléru byl do klávesnice nahrán ve dvou krocích. Prvním z nich bylo nahrání bootloaderu s pomocí vedoucím zapůjčeného programátoru PICkit 3. V druhém kroku byl do klávesnice prostřednictvím zavaděče odeslán zkompileovaný cílový program ve formě binárního *.hex* souboru. K tomu bylo využito počítačové aplikace Unified Bootloader od společnosti Microchip, která již byla zmíněna v podkapitole 8.1. Finální sestavené a naprogramované zařízení je k vidění na obrázku 28.



Obrázek 28: Realizovaná programovatelná klávesnice

10.1 Demonstrační profily

Pro demonstraci funkce klávesnice jsem v doprovodném počítačovém softwaru vytvořil čtyři profily, které využívají veškeré dostupné funkce a lze mezi nimi jednoduše přepínat stiskem enkodérů. Nastavení jednotlivých profilů je zobrazeno v příloze D.

První profil *Media* je určen pro ovládání médií. Z toho důvodu jsou klávesám přiřazeny funkce pro kontrolu přehrávání a enkodéru 1 ovládání hlasitosti počítače. Významná je zejména funkce klávesy 1, která s využitím externího softwaru SoundSwitch [52] umožňuje jednoduše přepínat mezi výstupními audio zařízeními.

Následující profil *Chrome* je určen pro zjednodušení práce ve webovém prohlížeči Chrome. Pomocí vyvolávání klávesových zkratk lze klávesnicí v prohlížeči otevírat a zavírat karty nebo mezi nimi přepínat. Stiskem klávesy 4 dochází k výpisu testovací e-mailové adresy a enkodéry umožňují vertikální a horizontální rolování.

Účelem třetího demonstračního profilu *Launch* je spouštění nejrůznějších programů. Mezi ty patří například Průzkumník souborů, webový prohlížeč Chrome nebo hudební přehrávač Spotify. Této funkcionality je docíleno tím způsobem, že stisky jednotlivých kláves spouští jednoduché Python skripty, které s využitím modulu Subprocess [53] otevírají dané programy.

Nakonec jsem vytvořil profil s názvem *LaTeX*, jehož účelem je zefektivnění práce v sázečím systému \LaTeX , který byl použit pro napsání této práce. Jednotlivým klávesám jsem přiřadil funkce umožňující rychlý výpis mnou často používaných příkazů, díky čemuž mi klávesnice již při tvorbě práce značně pomohla.

Kapitola 11

Zhodnocení dosažených výsledků

11.1 Kalkulace ceny

Kalkulace ceny mnou navržené klávesnice včetně cen jednotlivých komponent je uvedena v tabulce 3. Celková cena zařízení vychází na 712 Kč, což odpovídá přibližně \$32 nebo €30. Součástí kalkulace je však pouze cena součástí a není v ní zahrnuto mnoho dalších nákladů, kvůli kterým je reálná cena klávesnice podstatně vyšší. Mezi ty patří například náklady na dopravu, cena energií, potřebné vybavení a moje práce.

Tabulka 3: Kalkulace ceny navržené klávesnice

Položka	Množství	Prodejce	Cena v Kč
Deska plošných spojů	1	JLPCB	16
Mikrokontrolér PIC18F26K42	1	TME	62
Převodník FT232RL	1	AliExpress	27
Modul OLED SPI displeje	1	LaskaKit	178
ESD ochrana USBLC6-2SC6	1	LCSC	3
Kondenzátory, rezistory, feritový korálek	27	LCSC	8
Spínače Gateron G Pro 2.0 Brown	12	AliExpress	90
PBT klávesy	12	AliExpress	39
Hot-swappable patice Kailh	12	AliExpress	41
SK6812MINI-E RGB LED	12	LCSC	43
Rotační enkodéry EC11	2	AliExpress	14
Hliníkové hlavice na enkodér	2	AliExpress	49
5pinový programovací konektor	1	LCSC	2
USB-C konektor	1	TME	20
USB-C na USB-A kabel 2m	1	Alza	109
Šroubky, matky, nožičky	-	-	11
Celková cena			712

Výraznou položkou v seznamu součástí je modul displeje, který tvoří čtvrtinu celkové ceny zařízení. Totožný displej lze například na e-shopu AliExpress zakoupit za podstatně nižší cenu, z časových důvodů jsem však byl nucen zakoupit jej v tuzemském obchodě LaskaKit.

11.2 Srovnání s komerčně dostupnými zařízeními

Ve srovnání s komerčně dostupnými zařízeními, která byla popsána v kapitole 2, moje klávesnice disponuje jedinečnou kombinací komponent a funkcí, což byl jeden z hlavních důvodů pro její návrh. Mezi ty patří mechanické klávesy, dva enkodéry, displej zobrazující funkce kláves, LED podsvícení, USB C konektor a v neposlední řadě doprovodný software. Velkou výhodou je zároveň fakt, že je zařízení navrženo na míru a zejména jeho rozložení odpovídá mým specifickým potřebám.

Z hlediska pořizovací ceny se moje zařízení jeví podstatně levnější než všechna ostatní, a to řádově o stovky korun. Porovnání na základě ceny však nelze řádně provést,

jelikož součástí kalkulace ceny mé klávesnice nejsou výše zmíněné náklady ani žádná přírážka, jejímž účelem by byla generace zisku.

Mnou vytvořený doprovodný software klávesnice není zdaleka tak pokročilý jako například ten dodávaný k zařízení Elgato StreamDeck, které bylo popsáno v podkapitole 2.2. Je však plně funkční a nabízí všechny důležité základní funkce jako je ovládání médií, vyvolání klávesové zkratky a výpis textu. Zároveň díky zvolenému způsobu komunikace a funkci pro spouštění vlastních skriptů moje klávesnice nabízí rozsáhlé možnosti z hlediska dalšího využití. V neposlední řadě mám také přístup ke kompletnímu zdrojovému kódu, což mi umožňuje software i program mikrokontroléru dále zdokonalovat.

11.3 Možná vylepšení

V průběhu kompletace a testování mojí klávesnice jsem úspěšně ověřil a vyzkoušel veškeré požadované funkce. Zároveň se ukázalo i několik možností, jak zařízení vylepšit. Tato vylepšení však nejsou pro plnou funkci klávesnice podstatná a z časových důvodů již nebyla v rámci práce realizována.

První z možností vylepšení je přidání dalších funkcí klávesám a enkodérům. Příkladem může být spouštění programů, které jsem doposud realizoval pomocí externího skriptu. Pro uživatele klávesnice by však bylo pohodlnější mít možnost tuto a další funkce nastavit přímo v grafickém rozhraní aplikace.

Dále by bylo užitečné mít možnost jednoduše měnit barvu RGB LED, které jsou umístěny pod klávesami, a to buď na klávesnici samotné nebo prostřednictvím počítačové aplikace. Momentálně je barva LED pevně nastavená a pro její změnu je třeba aktualizovat program mikrokontroléru s využitím zavaděče.

Během testování se také ukázalo, že klávesnice neumožňuje registraci stisku více kláves zároveň. Tento jev však neznepříjemňuje její používání a jeho odstranění je pouze další z cest, jak zařízení zdokonalit.

V neposlední řadě by bylo vhodné navrhnout pro klávesnici pouzdro a ideálně s pomocí 3D tisku jej zhotovit. Pouzdro by zlepšilo celkový vzhled klávesnice, zajistilo by lepší usazení spínačů na desce a zároveň by poskytlo ochranu použitým součástkám.

Kapitola 12

Závěr

Cílem této práce bylo navrhnout a realizovat vlastní programovatelnou klávesnici neboli macropad pro zefektivnění práce na počítači. Základní požadavky na zařízení zahrnovaly alespoň deset programovatelných kláves, informační displej zobrazující jejich funkce a připojení pomocí sběrnice USB.

V první řadě byl proveden průzkum trhu, ve kterém bylo popsáno a porovnáno několik komerčně dostupných programovatelných klávesnic. Na jeho základě byly definovány další požadavky na moje zařízení jako implementace mechanických kláves a rotačního enkodéru nebo vytvoření doprovodného počítačového softwaru.

Následně byla rozebrána problematika jednotlivých komponent tvořících klávesnici a byly vybrány vhodné součástky pro mojí aplikaci. Klíčové bylo zejména rozhodnutí použít pro komunikaci mezi klávesnicí a počítačem UART-USB převodník FT232RL, který je počítačem rozpoznán jako virtuální sériový port. Ten umožňuje obousměrný přenos dat a jejich jednoduché zpracování počítačovou aplikací. Blíže byla popsána také použitá komunikační rozhraní a s využitím softwaru KiCad bylo navrženo obvodové zapojení mojí klávesnice a odpovídající deska plošných spojů.

V rámci práce byl dále vytvořen program pro mikrokontrolér řídící samotnou klávesnici, který se skládá ze dvou částí. Jednou z nich je programový zavaděč a druhou je cílová aplikace, která zajišťuje snímání vstupů z klávesnice, obsluhu displeje a odesílání dat do počítače. Zároveň byl vyvinut doprovodný počítačový software, jehož účelem je konfigurace klávesnice, zpracování příchozích dat a následné vykonávání nastavených funkcí. Mezi dostupné funkce patří ovládání médií, vyvolávání klávesových zkratk, výpis textu nebo spouštění uživatelských Python skriptů.

Následně byla klávesnice fyzicky realizována a dle zadání byly vytvořeny čtyři profily pro demonstraci její funkce.

V závěru práce byla mnou realizovaná klávesnice stručně porovnána s komerčně dostupnými produkty a bylo představeno několik způsobů, jak by bylo možné její používání zpříjemnit. Z hlediska funkčnosti je s popsánými alternativami srovnatelná a jako jediná disponuje mechanickými spínači v kombinaci s dvěma enkodéry, displejem, LED podsvícením a doprovodným softwarem. Klávesnice byla zároveň navržena v souladu s mými specifickými potřebami a je tak pro mne ideálním pomocníkem při práci na počítači.

Literatura

- [1] Adafruit MacroPad RP2040 Starter Kit. In: *Adafruit* [online]. Dostupné z: <https://www.adafruit.com/product/5128>
- [2] Stream Deck MK.2. In: *Elgato* [online]. Elgato, c2023 [cit. 2023-05-20]. Dostupné z: <https://www.elgato.com/us/en/p/stream-deck-mk2-black>
- [3] DuckyPad: Do-It-All Mechanical Macropad. In: *Tindie* [online]. Tindie, c2023 [cit. 2023-05-20]. Dostupné z: <https://www.tindie.com/products/dekunukem/duckypad-do-it-all-mechanical-macropad/>
- [4] DuckyPad. In: GitHub [online]. GitHub, c2023 [cit. 2023-05-20]. Dostupné z: <https://github.com/dekuNukem/duckyPad>
- [5] HID Usage Tables for Universal Serial Bus (USB). In: *USB* [online]. USB Implementers Forum, 01/27/2023 [cit. 2023-05-20]. Dostupné z: <https://www.usb.org/document-library/hid-usage-tables-14>
- [6] Mountain DisplayPad. In: *Mountain* [online]. Mountain, c2023 [cit. 2023-05-20]. Dostupné z: <https://mountain.gg/keypads/displaypad/>
- [7] Pikatea Macropad R26. In: *Pikatea* [online]. Pikatea, c2023 [cit. 2023-05-20]. Dostupné z: <https://www.pikatea.com/products/pikatea-macropad-r26-desktop>
- [8] WILMSHURST, Tim. *Designing embedded systems with PIC microcontrollers: Principles and applications*. Second Edition. Kidlington: Newnes, 2010. ISBN 9781856177504.
- [9] SUBERO, Armstrong. *Programming PIC Microcontrollers with XC8*. Moruga: Apress, 2017. ISBN 9781484232729.
- [10] Datasheet PIC18F26K42. In: *Microchip* [online]. Microchip Technology, c1998-2023 [cit. 2023-05-20]. Dostupné z: <https://www.microchip.com/en-us/product/PIC18F26K42>
- [11] Datasheet CH9329. In: *WCH* [online]. Nanjing Qinheng Microelectronics, c2002-2023 [cit. 2023-05-20]. Dostupné z: http://wch-ic.com/downloads/CH9329DS1_PDF.html
- [12] Datasheet FT232RL. In: *FTDI Chip* [online]. Future Technology Devices International, c2023 [cit. 2023-05-20]. Dostupné z: <https://ftdichip.com/products/ft232rl/>
- [13] CUEVAS, Zackery. Click Clack: Your Ultimate Guide to Mechanical Key Switches. In: *PCMag* [online]. Ziff Davis, c1996-2023, April 29, 2022 [cit. 2023-05-20]. Dostupné z: <https://www.pcmag.com/how-to/guide-to-mechanical-key-switches-which-kind-is-right-for-you>
- [14] BARR, Benjamin. Obrázek mechanického spínače. In: *High Ground Gaming* [online]. Top Shelf Media, c2023, Jul 10, 2018 [cit. 2023-05-20]. Dostupné z: <https://www.highgroundgaming.com/mechanical-vs-membrane-keyboard/>

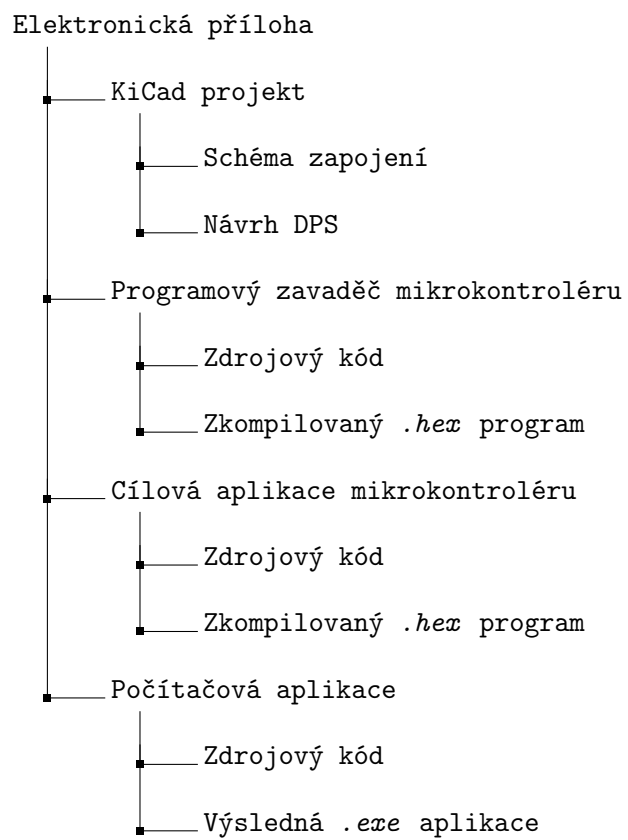
- [15] BERGER, Christoph. Switch Types. In: *Mechanical Keyboard* [online]. March 15, 2021 [cit. 2023-05-20]. Dostupné z: <https://www.mechanical-keyboard.org/switch-types/>
- [16] Obrázek typů mechanických spínačů. In: *Fantech World* [online]. Fantech World, c2023, August 6, 2021 [cit. 2023-05-20]. Dostupné z: <https://fantechworld.com/hot-swappable-switches-what/>
- [17] Spínače Cherry MX. In: *Cherry MX* [online]. Cherry, c2021 [cit. 2023-05-20]. Dostupné z: <https://www.cherrymx.de/en/cherry-mx.html>
- [18] DOBBIN, Jolene. Are Mechanical Keyboards Better for Gaming?. In: *HP* [online]. HP Development Company, c2020, August 11, 2018 [cit. 2023-05-20]. Dostupné z: <https://www.hp.com/us-en/shop/tech-takes/are-mechanical-keyboards-better-gaming>
- [19] Kailh hot-swappable patice. In: *Kailh* [online]. Dongguan Kaihua Electronics [cit. 2023-05-20]. Dostupné z: <http://www.kailh.com/en/Products/Ks/HPC/883.html>
- [20] Gateron G Pro 2.0 Switch Set. In: *Gateron* [online]. Gateron, c2023 [cit. 2023-05-20]. Dostupné z: <https://www.gateron.co/products/gateron-g-pro-2-0-switch-set?variant=40036010557529>
- [21] MURRAY, Mike. How rotary encoders work. In: *The Geek Pub* [online]. The Geek Pub [cit. 2023-05-20]. Dostupné z: <https://www.thegeekpub.com/245407/how-rotary-encoders-work-electronics-basics/>
- [22] REESE, Robert Bryan. *Microprocessors: From Assembly Language to C Using the PIC18FXX2*. Hingham: Da Vinci Engineering Press, c2005. ISBN 1584503785.
- [23] GAY, Warren. *Custom Raspberry Pi Interfaces: Design and build hardware interfaces for the Raspberry Pi*. St Catharines: Apress, c2017. ISBN 9781484224069.
- [24] Obrázek rotačního enkodéru. In: *TME* [online]. TME, c2023 [cit. 2023-05-24]. Dostupné z: <https://www.tme.eu/cz/details/pec11r-4220k-s0024/inkrementalni-enkodery/bourns/>
- [25] HOROWITZ, Paul a Winfield HILL. *The Art of Electronics*. Third Edition. New York: Cambridge University Press, 2015. ISBN 9780521809269.
- [26] KODEN, Mitsuhiro. *OLED Displays and Lighting*. Chichester: Wiley-IEEE, c2017. ISBN 9781119040453.
- [27] Obrázek modulu OLED displeje. In: *LaskaKit* [online]. LaskaKit, c2023 [cit. 2023-05-20]. Dostupné z: <https://www.laskakit.cz/oled-displej-bily-128x64-1-3--spi/>
- [28] Datasheet SH1106. In: *Pololu* [online]. Pololu, c2001-2023 [cit. 2023-05-20]. Dostupné z: <https://www.pololu.com/product/3760/resources>
- [29] Datasheet SK6812MINI-E. In: *Adafruit* [online]. [cit. 2023-05-20]. Dostupné z: <https://www.adafruit.com/product/4960>

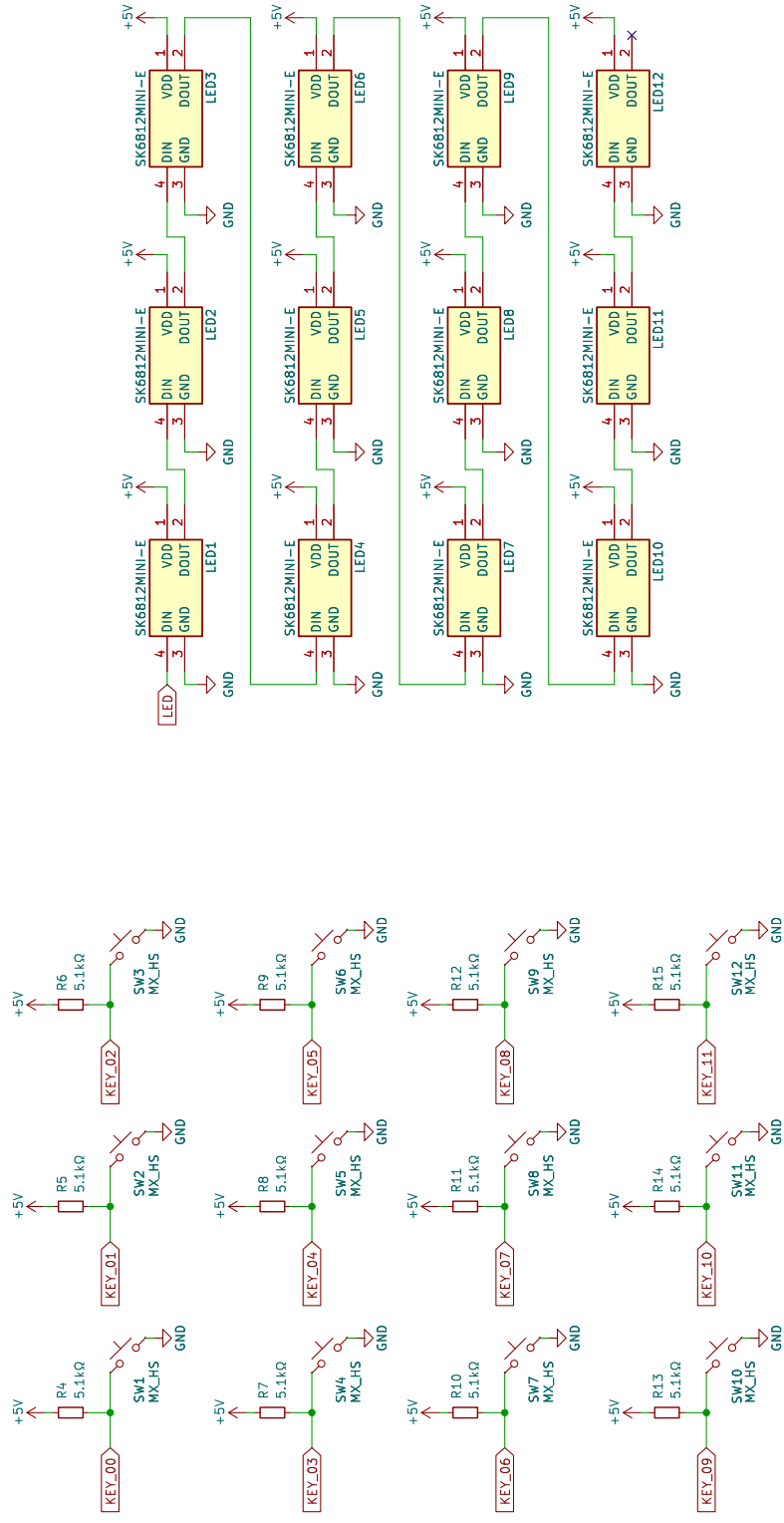
- [30] DAWOUD, Dawoud Shenouda a Peter DAWOUD. *Serial Communication Protocols and Standards: RS232/485, UART/USART, SPI, USB, INSTEON, Wi-Fi and WiMAX*. Gistrup: River Publishers, 2020. ISBN 9788770221542.
- [31] Obrázek propojení UART modulů. In: *Circuit Basics* [online]. [cit. 2023-05-20]. Dostupné z: <https://www.circuitbasics.com/wp-content/uploads/2016/01/Introduction-to-UART-Basic-Connection-Diagram.png>
- [32] Obrázek zapojení sběrnice SPI. In: *Wikipedia* [online]. [cit. 2023-05-21]. Dostupné z: https://cs.wikipedia.org/wiki/Serial_Peripheral_Interface
- [33] AXELSON, Jan. *USB Complete: The Developer's Guide*. Fifth Edition. Madison: Lakeview Research, 2015. ISBN 9781931448284.
- [34] Universal Serial Bus Type-C Cable and Connector Specification. In: *USB* [online]. USB Implementers Forum, 10/18/2022 [cit. 2023-05-21]. Dostupné z: <https://www.usb.org/document-library/usb-type-cr-cable-and-connector-specification-release-22>
- [35] Obrázky konektorů USB. In: *Wikipedia* [online]. [cit. 2023-05-21]. Dostupné z: https://en.wikipedia.org/wiki/USB_hardware
- [36] ZÁHLAVA, Vít. *Návrh a konstrukce desek plošných spojů: Principy a pravidla praktického návrhu*. Praha: BEN - technická literatura, 2010. ISBN 9788073002664.
- [37] Datasheet a výkres USB4085. In: *GCT* [online]. Global Connector Technology, c2023 [cit. 2023-05-21]. Dostupné z: <https://gct.co/connector/usb4085>
- [38] AN1953: Introduction to USB Type-C. In: *Microchip* [online]. Microchip Technology, c1998-2023, 06/24/2015 [cit. 2023-05-21]. Dostupné z: <https://www.microchip.com/en-us/application-notes/an1953>
- [39] Datasheet USBLC6. In: *STMicroelectronics* [online]. STMicroelectronics, c2023, December 2021 [cit. 2023-05-21]. Dostupné z: <https://www.st.com/en/protecti ons-and-emi-filters/usblc6-2.html>
- [40] PARKS, Michael. Pull Up Your Pins: How to Size Pull-up Resistors. In: *Mouser* [online]. Mouser Electronics, c2023, October 31, 2018 [cit. 2023-05-24]. Dostupné z: <https://www.mouser.com/blog/pull-up-your-pins-how-to-size-pull-up-resistors>
- [41] PCB Trace Width Calculator. *4PCB* [online]. Advanced Circuits, c2023 [cit. 2023-05-21]. Dostupné z: <https://www.4pcb.com/trace-width-calculator.html>
- [42] ROBLEDO, Edwin. PCB Routing Requirements for USB 2.0: Ultimate Guide. In: *Autodesk* [online]. Autodesk, c2022, February 3, 2023 [cit. 2023-05-21]. Dostupné z: <https://www.autodesk.com/products/fusion-360/blog/pcb-routing-requirements-usb-ultimate-guide/>
- [43] KERNIGHAN, Brian W. a Dennis M. RITCHIE. *The C Programming Language*. Second Edition. New Jersey: Prentice Hall, 1988. ISBN 9780131103702.

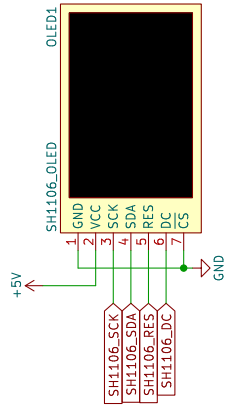
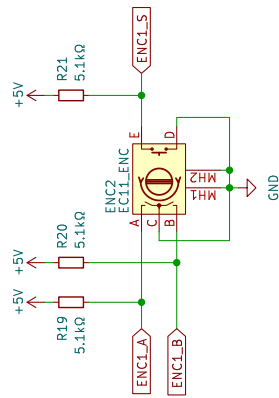
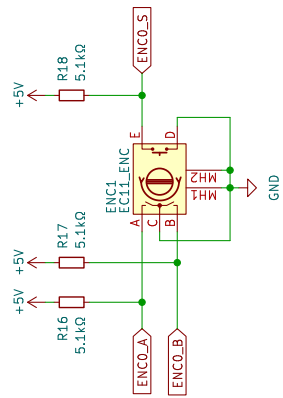
- [44] MPLAB XC8 C Compiler User's Guide for PIC MCU. In: *Microchip* [online]. Microchip Technology, c1998-2023 [cit. 2023-05-21]. Dostupné z: <https://www.microchip.com/en-us/tools-resources/develop/mplab-xc-compilers/downloads-documentation#XC8D>
- [45] Bootloader Generator User's Guide. In: *Microchip* [online]. Microchip Technology, c1998-2023 [cit. 2023-05-21]. Dostupné z: <https://www.microchip.com/en-us/tools-resources/develop/libraries/microchip-bootloaders/8-bit>
- [46] BOYANOV, Neven. *Standard ASCII 6x8 font* [online]. [cit. 2023-05-21]. Dostupné z: https://lexus2k.github.io/ssd1306/ssd1306_fonts_8c_source.html
- [47] *Qt Documentation* [online]. The Qt Company, c2023 [cit. 2023-05-21]. Dostupné z: <https://doc.qt.io/>
- [48] *Sada ikon Feather* [online]. [cit. 2023-05-21]. Dostupné z: <https://feathericons.com/>
- [49] *Dokumentace modulu Pyserial* [online]. Chris Liechti, c2001-2020 [cit. 2023-05-21]. Dostupné z: <https://pyserial.readthedocs.io/en/latest/>
- [50] *Dokumentace modulu Pynput* [online]. Moses Palmér, c2015-2022 [cit. 2023-05-21]. Dostupné z: <https://pynput.readthedocs.io/en/latest/>
- [51] *Dokumentace modulu Runpy* [online]. Python Software Foundation, c2001-2023 [cit. 2023-05-21]. Dostupné z: <https://docs.python.org/3/library/runpy.html>
- [52] *SoundSwitch* [online]. Antoine Aflalo [cit. 2023-05-21]. Dostupné z: <https://soundswitch.aafalalo.me/>
- [53] *Dokumentace modulu Subprocess* [online]. Python Software Foundation, c2001-2023 [cit. 2023-05-21]. Dostupné z: <https://docs.python.org/3/library/subprocess.html>

Přílohy

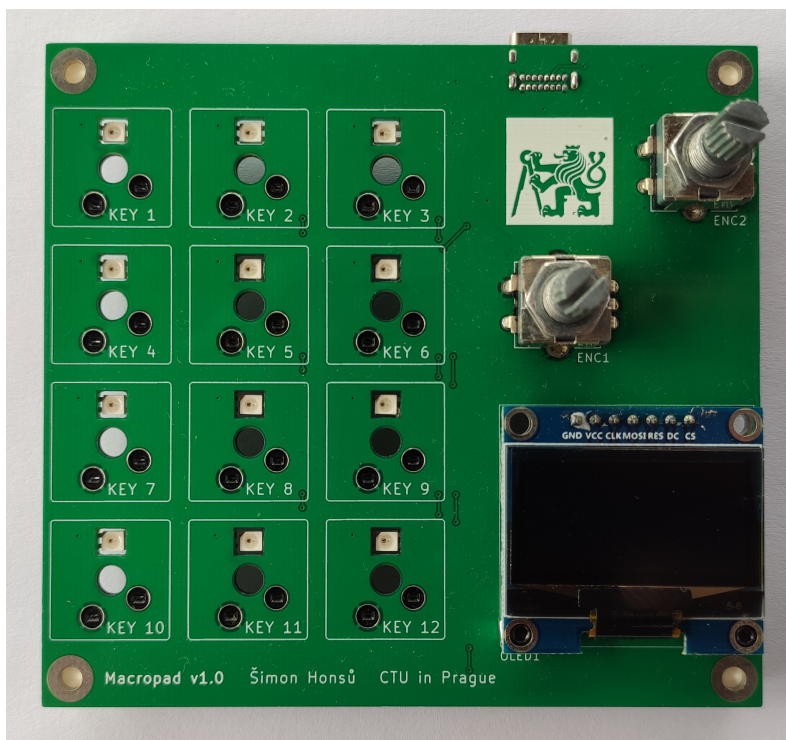
A Obsah elektronické přílohy



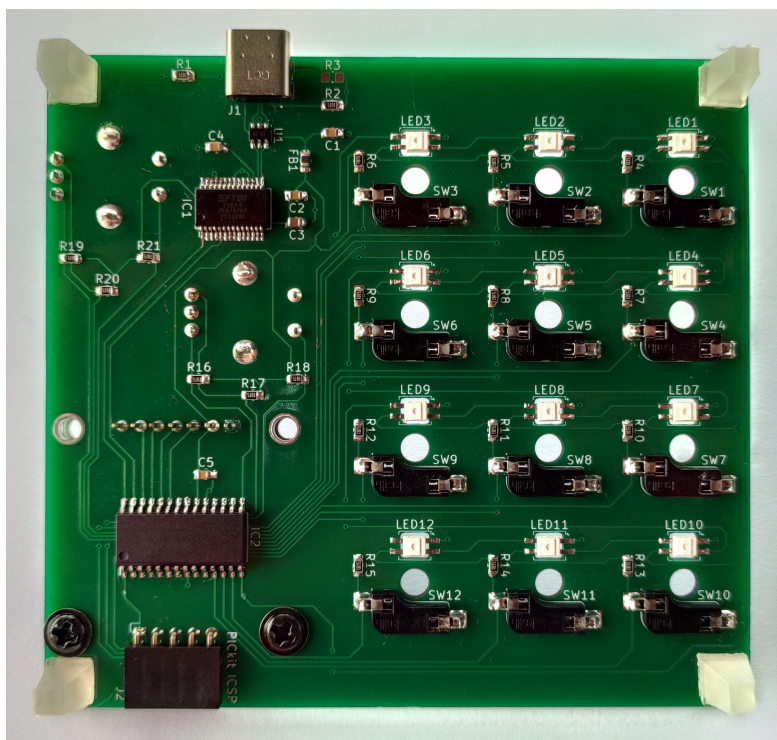




C Osazená deska plošných spojů

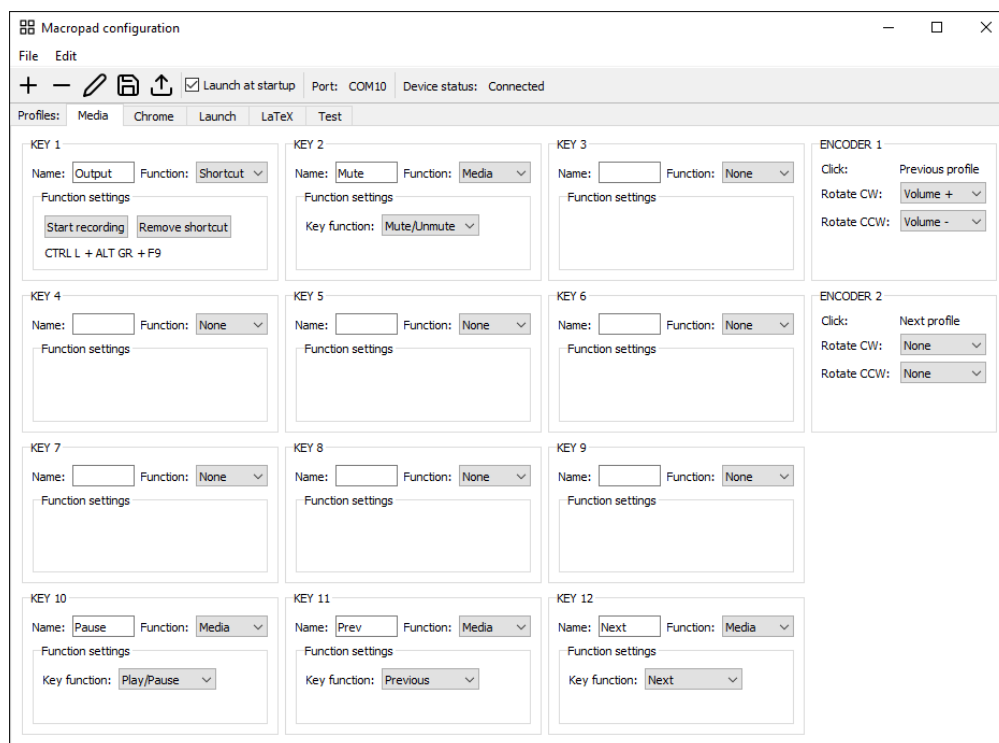


Obrázek C.1: Osazená deska plošných spojů – pohled shora

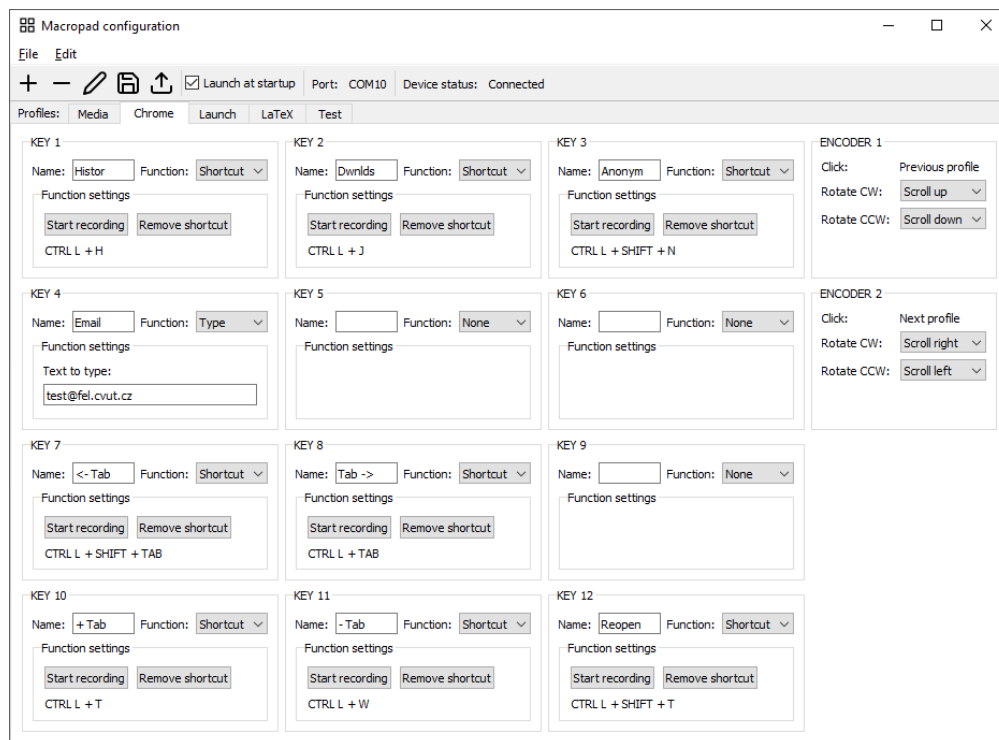


Obrázek C.2: Osazená deska plošných spojů – pohled zdola

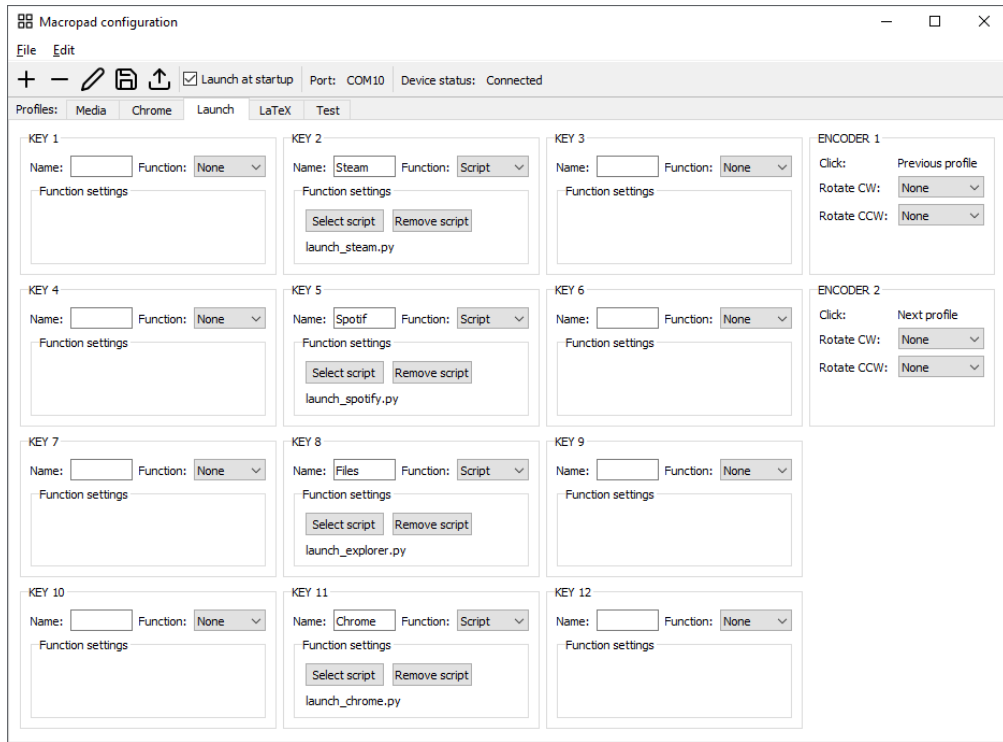
D Nastavení demonstračních profilů



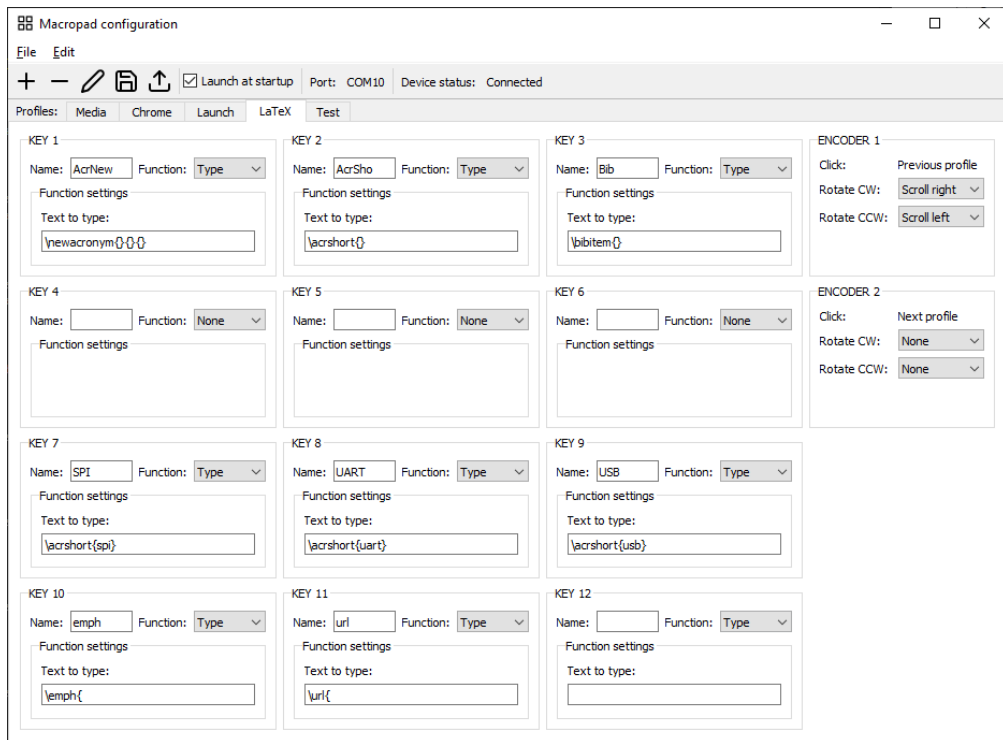
Obrázek D.1: Nastavení profilu *Media*



Obrázek D.2: Nastavení profilu *Chrome*



Obrázek D.3: Nastavení profilu *Launch*



Obrázek D.4: Nastavení profilu *LaTeX*