



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**  
DEPARTMENT OF INTELLIGENT SYSTEMS

**HARDWAROVÁ IMPLEMENTACE ŠACHŮ**  
HARDWARE CHESS IMPLEMENTATION

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**ROMAN ZAJÍC**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. JAROSLAV ROZMAN, Ph.D.**

**BRNO 2020**

## Zadání bakalářské práce



22812

Student: **Zajíć Roman**

Program: Informační technologie

Název: **Hardwareová implementace šachů**  
**Hardware Chess Implementation**

Kategorie: Teorie obvodů

Zadání:

1. Nastudujte problematiku hardwareových šachovnic. Nastudujte elektroniku a mikrokontroléry nutné pro tvorbu vlastní šachovnice.
2. Na základě získaných znalostí navrhněte vlastní šachovnici řízenou vybraným mikrokontrolérem. U návrhu se zaměřte na co nejlepší detekci a rozpoznání šachových figur. Při návrhu počítejte také se signalizace možných tahů a tvorbou vlastní UI implementované v mikrokontroléru.
3. Šachovnici, včetně šachových figur vytvořte.
4. Proveďte otestování funkčnosti vytvořené šachovnice. Zaměřte se především na detekci a rozpoznání šachových figur. Dále otestujte výdrž baterie a další...
5. Práci zhodnoťte, srovnajte se stávajícími řešeními a navrhněte možná vylepšení do budoucna.

Literatura:

- Podle domluvy s vedoucím.

Pro udělení zápočtu za první semestr je požadováno:

- první dva body zadání

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Rozman Jaroslav, Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2019

Datum odevzdání: 28. května 2020

Datum schválení: 31. října 2019

## **Abstrakt**

Tato práce se zabývá návrhem a následnou výrobou elektronické šachovnice. Důraz byl kladen na to, aby výsledná šachovnice byla co nejvíce podobná klasickým šachovým deskám. Zejména, aby co nejméně omezovala hráče při hře. K ovládání šachovnice slouží modul hodin, který kromě přepínače času obsahuje i displej a ovládací klávesy. Jako hlavní kontrolér této desky slouží WeMos R1 mini v kombinaci se třemi Arduiny Nano. Dvě z těchto Arduin slouží k detekci šachových figur a signalizaci na samotné šachovnici. Třetí Arduino slouží k ovládání šachových hodin a úpravám nastavení šachovnice.

## **Abstract**

This thesis deals with the design and subsequent production of electronic chessboard. The emphasis was placed so that the resulting chessboard was as similar as possible to a classic chessboard. In particular, to restrict players as little as possible when playing. The clock module is used to control the chessboard, which in addition to the time switch also contains a screen and keys. WeMos R1 mini, in combination with three Arduino Nano, serves as the main controller of this board. Two of these Arduinos are used to detect figures and signaling on the board itself. The third Arduino is used to control the chess clock and adjusting the chessboard settings.

## **Klíčová slova**

elektronické šachy, Wemos D1 mini, Arduino NANO, RFID, jazýčkové relé, RGB LED pásek WS2812B

## **Keywords**

electronic chess, Wemos D1 mini, Arduino NANO, RFID, reed switch, RGB LED strip WS2812B

## **Citace**

ZAJÍC, Roman. *Hardwareová implementace šachů*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Rozman, Ph.D.

# **Hardwareová implementace šachů**

## **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Rozmana Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....  
Roman Zajíc  
27. května 2020

## **Poděkování**

Rád bych poděkoval Ing. Jaroslavu Rozmanovi Ph.D. za vedení mojí bakalářské práce práce. Také bych chtěl poděkovat své rodině a přátelům, za podporu, nejen v době psaní této práce, ale i v době celého studia.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Stávající řešení</b>	<b>3</b>
2.1	Excalibur Saber II . . . . .	3
2.2	MillenniumChess . . . . .	4
2.3	Square off . . . . .	5
2.4	Raspberry Pi & Arduino Wooden Chess Computer . . . . .	6
2.5	IoT Connected Chess Boards . . . . .	6
2.6	Wireless Arduino Powered Chess . . . . .	6
<b>3</b>	<b>Šachy</b>	<b>8</b>
3.1	Pravidla šachu . . . . .	8
3.2	Šachová notace . . . . .	9
<b>4</b>	<b>Hardwareový návrh</b>	<b>11</b>
4.1	Řídící jednotky . . . . .	11
4.2	Detekce figur . . . . .	13
4.3	Signalizace a UI . . . . .	16
4.4	Ostatní použité součástky . . . . .	17
4.5	Funkční model šachovnice . . . . .	20
<b>5</b>	<b>Výroba šachovnice</b>	<b>25</b>
5.1	Návrh PCB v programu Eagle . . . . .	25
5.2	Výroba a testování PCB . . . . .	27
5.3	Výroba ostatních částí šachovnice . . . . .	30
<b>6</b>	<b>Implementace šachů</b>	<b>33</b>
6.1	Vývojové prostředí, komunikace a knihovny . . . . .	33
6.2	Slave pro řízení multiplexovací desky . . . . .	35
6.3	Hodiny . . . . .	35
6.4	Hlavní řídící jednotka . . . . .	35
<b>7</b>	<b>Testování a výroba prototypů</b>	<b>37</b>
<b>8</b>	<b>Možná vylepšení</b>	<b>40</b>
<b>9</b>	<b>Závěr</b>	<b>41</b>
	<b>Literatura</b>	<b>43</b>

# Kapitola 1

## Úvod

Šachy jsou velmi oblíbenou hrou, která v pohledu informační technologie již v roce 1997 byla „vyřešena“ počítačem, když tehdy počítač Deep Blue od firmy IBM porazil šachového velmistra Garriho Kasparova. Od té doby ale vývoj šachových motorů neustal a programátoři se stále snažili psát silnější motory pro hraní těchto her. Příkladem může být například Stockfish, který patří v současnosti mezi nejsilnější šachové motory, a je distribuován pod GPL licencí, tudíž je volně dostupný ke stažení a užití. Obecně vzato s vylepšováním postupů při tvorbě AI a rostoucím výkonem byly jednotlivé motory silnější a v současnosti pro moderní počítače není problém většinu z těchto motorů pustit.

V nedávné době došel svět šachů k dalšímu kroku v před. Poté, co společnost DeepMind pokročila svět hráčů hry Go, který si do té doby myslel, že nejlepší hráči této hry nebudu použití alespoň dalších 10 let, začala s vývojem motoru pro šachy. Přibližně po dvou letech zveřejnila svůj motor AlphaZero (Prosinec 2017), který na rozdíl od ostatních motorů využívá technologii neurálních sítí a procesu samovolného učení se této hry. Po svém vypouštění AlpaZero dominovala ostatním šachovým motorům a spustila vlnu nových postupů při vytváření opět silnějších programů pro šachy. Příkladem těchto motorů je Leela Chess Zero.

Tehdy jsem se rozhodl, že moje bakalářská práce se bude zabývat právě šachy. Původně jsem se chtěl zaměřit specificky na umělou inteligenci, ale prací s podobnou tématikou byl nespočet, a proto jsem se rozhodl pro vytvoření elektronické „hmatatelné“ šachovnice, protože vytváření podobných projektů mě tou dobou velmi začalo bavit a zajímat.

Tyto šachovnice samozřejmě také existují, příkladem lze zmínit šachovnice od firmy Millenium. Existuje i větší množství po domácku vyráběných šachovnic. Ty z velké části pracují pomocí jazýčkových relé a jako kontrolér obsahují některé Arduino v kombinaci s Raspberry Pi nebo novější z nich některou mutaci ESP826. Více se existujícími řešeními zabývá kapitola 2.

Kapitola 3 popisuje jednotlivá pravidla šachu a popis šachové notace. Kapitola 4 uvádí přehled zvažovaných komponent, jako jsou jednotlivé kontroléry a ostatní elektronika. Dále popisuje samotný proces návrhu a prototypování jednotlivých elektronických částí. Jednotlivé kroky výroby šachovnice jsou popsány v kapitole 5. V kapitole 6 jsou uvedené požadavky pro vývoj softwaru na těchto kontrolérech a stručně popsaná struktura jednotlivých programů pro kontrolu šachovnice.

## Kapitola 2

# Stávající řešení

Seznam elektronických šachovnic je velký a všechny zmínit nemohu. Některé se liší například použitým displejem a podobně (DGT Centeur používající e-paper display). Většina z nich používá k načítání tlakové senzory nebo jazýčková relé, a i signalizace tahů je většinou stejná jakou u níže zmiňovaných šachovnic.

### 2.1 Excalibur Saber II

Jedna z nejstarších elektronických šachovnic, kterou se mi podařilo dohledat (1995). Načítání figur je řešeno tlakovým senzorem a signalizace tahů počítače je řešená pomocí výpisu počáteční pozice a cílové pozice na malém LCD display (viz obr. 2.1). Šachovnice umožňuje ukládání her a různé módy výuky nových hráčů. Samotný herní motor má sílu 1650 ELO. Použitý procesor je Motorola MC68HC05 (8bit, 2MHz 176B RAM). Napájení je řešeno třemi AA bateriemi. Pozdější modely přinášeli zejména vylepšení motoru a herních módů.



Obrázek 2.1: Šachovnice Saber II Převzato z [http://www.spacious-mind.com/html/sabre\\_ii.html](http://www.spacious-mind.com/html/sabre_ii.html)

## 2.2 MillenniumChess

Tato firma s 35letými zkušenostmi ve výrobě elektronických šachovnic<sup>1</sup> je ukázkou, že podobné šachovnice se vyrábějí již dlouhou dobu, a zároveň toho, jak může taková šachovnice vypadat. Tato společnost nabízí velké množství šachovnic, z nichž nejnovější pracuje na bázi načítání dílků pomocí RFID. Starší varianty pracují s tlakovým senzorem. Dále tyto šachovnice obsahují nějakou formu indikace tahů, většinou pomocí LCD displeje. Šachovnice jsou ukázány na obr. 2.2.

### Chess Genius Exclusive

Jedná se o jejich nejlépe vybavenou šachovnici. Načítání figur je řešeno pomocí RFID. Každé políčko má v rozích malé LED diody, kterými se signalizuje pohyb figur, jejich špatná pozice a podobně. K šachovnici je ještě externí modul sloužící k ovládání, případně jako šachové hodiny a podobně. Velikost šachovnice zdá se odpovídá standardům turnajových šachových desek (2 – 2,5 palce na šachový dílek podle šachové federace spojených států – USCF). Pro běh programu je v šachovnici procesor ARM Cortex M7 (300MHz/2MB ROM/ 384KB RAM). Samotný šachový motor dosahuje síly přibližně 2300 ELO (ohodnocení výkonnosti). Cena této šachovnice je 16 500Kč<sup>2</sup>.

### Chess Genius PRO

Jedna z jejich šachovnic pracujících s načítáním dílků pomocí tlakových senzoru (tady pravděpodobně membránové tlačítko či podobné). Samotná šachovnice je spíše cestovních rozměrů. Signalizace tahů je pomocí LCD displeje a vestavěných tlačítek na jedné straně šachovnice. Šachovnice používá procesor ARM Cortex M4 (120MHz / 160KB RAM). Samotný šachový motor dosahuje síly přibližně 2200 ELO (ohodnocení výkonnosti). Cena této šachovnice je 4500Kč<sup>3</sup>.



Obrázek 2.2: MillenniumChess šachovnice - šachovnice Chess Genius Exclusive , šachovnice Chess Genius PRO - obrázky převzaty z <https://computerchess.com/>

<sup>1</sup><http://www.millennium2000.de/>

<sup>2</sup><https://computerchess.com/en/chessgenius-exclusive-2/>

<sup>3</sup><https://computerchess.com/en/chessgenius-pro-2/>

## 2.3 Square off

Je dalším výrobcem elektronických šachovnic<sup>4</sup>. Zmiňuji ho zde zejména kvůli jejich inovativním šachům, které jsou schopny automatického pohybu figur. Uvnitř šachovnice je mechanika na styl plotru, s magnetem na přichycení figur. Samotné načítání figur je řešeno tlakovým senzorem nebo tlačítkem pod jednotlivými políčky. Jejich šachovnice jsou ale bez externích hodin, případně vestavěného kontrolního panelu. Veškerá nastavení a další obsluha je prováděna pomocí externí mobilní aplikace.

Cena se pohybuje v rozmezí od 11 000Kč u jejich „vlajkové lodě“ Grand Kingdom Set po 4500Kč pro jejich novou šachovnici Neo, u té ovšem ještě není jasné její vybavení. Šachovnici můžete videt na obr. 2.3

Šachovnice Grand Kingdom používá jako motor program Stockfish 10, tudíž Elo se dle nastavení pohybuje v rozmezí 700-3300. Z dostupných informací nevím, jestli samotný motor běží v mobilní aplikaci, nebo v samotné šachové desce, stejně jako se mi nepodařilo zjistit bližší informace o kontroléru v samotné šachovnici.



Obrázek 2.3: Šachovnice Grand Kingdom převzato z <https://squareoffnow.com/product/gks>

---

<sup>4</sup><https://squareoffnow.com/product/gks>

## 2.4 Raspberry Pi & Arduino Wooden Chess Computer

Toto byla první elektronická šachovnice<sup>5</sup>, kterou jsem nalezl. Jedná se o domácí projekt stavěný na Arduinu UNO (obsluha šachovnice, LCD výstup apod.) a Raspberry Pi (běh šachového motoru – Stockfish a zpracování dat z Arduina). Šachovnice obsahuje matici jazýčkových relé pro detekci figur a druhou matrici LED diod pro signalizaci tahů. Samotná relé jsou napojena na I/O rozbočovač a ten je napojen na Arduino. Šachovnici ukazuje obr. 2.4.



Obrázek 2.4: Raspberry Pi & Arduino Wooden Chess - převzato z <https://www.youtube.com/watch?v=f4WUa8H0cPo>

## 2.5 IoT Connected Chess Boards

Tato šachovnice je projekt Sarah Al-Mutlaq<sup>6</sup>, (technik společnosti SparkFun). Jedná se o dvě vzájemně propojené šachové desky. Načítání pozice dílků je řešeno jazýčkovým relé v kombinaci s posuvným registrem. Každá polovina šachovnice pracuje pouze s polovinou figur (jedna pouze s bílými a druhá pouze s černými viz obr. 2.5). Uživatelské rozhraní je kombinace adresovatelných RGB LED diod a malého LCD displeje. LCD je součástí ovládacího panelu z boku šachovnice. Jelikož se jedná o IoT (internet of things) šachovnici, postavenou na kontroléru ESP8266, je připojena na logovací stránku SparkFunu. Veškerá schémata zapojení a ostatní soubory jsou dostupné online na jejím GitHubu<sup>7</sup>.

## 2.6 Wireless Arduino Powered Chess

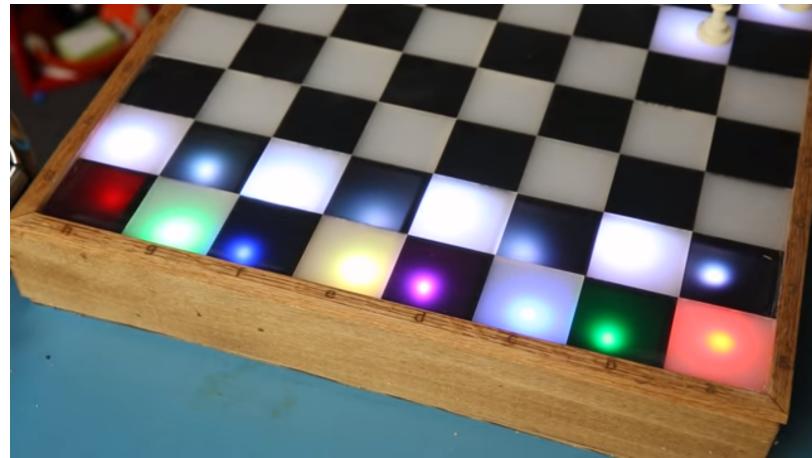
Tento pravděpodobně studentský projekt<sup>8</sup> vyžívá k pohybu figur podobně jako MillenniumChess mechanik plotru s magnetem na ramenu viz obr. 2.6. Samotné načítání dílků je řešeno pravděpodobně pomocí jazýčkových relé. Samotný projekt je z roku 2012, tudíž byl

<sup>5</sup><https://www.youtube.com/watch?v=f4WUa8H0cPo>

<sup>6</sup><https://www.youtube.com/watch?v=ta-q7Qbpj4Q>

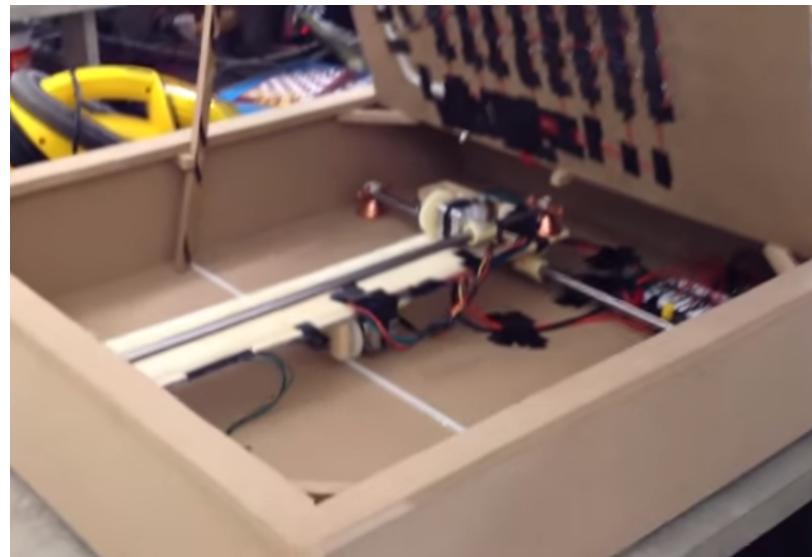
<sup>7</sup>[https://github.com/sparkfun/IoT\\_Connected\\_Chess\\_Boards](https://github.com/sparkfun/IoT_Connected_Chess_Boards)

<sup>8</sup><https://www.youtube.com/watch?v=dX37LFv8jWY>



Obrázek 2.5: Šachovnice s připojeník k IoT - signalizace různých figur z druhé šachovnice, převzato z <https://www.youtube.com/watch?v=ta-q7Qbpj4Q>

nejspíše první tohoto typu. Více informací jsem k této šachovnici nenašel, jelikož jsem mohl čerpat pouze ze tří videí od tvůrců. Zmiňuji ho zde proto, aby bylo vidět že pohyb figur není novinkou MillenniumChess, ale je již znám delší dobu.



Obrázek 2.6: Šachovnice s automatickým pohybem dílků - ukázka vnitřního mechanizmu pro pohyb dílků, převzato z <https://www.youtube.com/watch?v=XCp4M08IWnA>

# Kapitola 3

## Šachy

Jelikož se tato práce zabývá výrobou šachové desky, je vhodné zmínit pár věcí i k samotné hře.

Tato hra se objevila v Evropě již okolo 15.století. Již od vzniku této hry se v ní pořádají různé turnaje, ovšem profesionální šach se vyskytuje přibližně od poslední čtvrtiny 19. století, kdy proběhl první mezinárodní turnaj.

Šachy se hrají na desce rozdělené na 8x8 polí se střídajícím se tmavým a světlým polem (nejčastěji černá a bílá). Každý hráč má na začátku šestnáct figur s rozdílnými vlastnostmi. Těchto figur existuje šest variant - král, dáma, střelec, kůň, věž a pěšec.<sup>[6]</sup>

### 3.1 Pravidla šachu

Nejprve se musí jednotlivé šachové figury postavit do základního postavení tak, aby černý roh šachovnice byl po levé ruce bílého hráče. První řada obsahuje z levé strany věž, koně, střelce, dámu, krále, střelce, koně a věž. Druhá řada obsahuje osmu pěšců. Černý rozestaví své figury stejným stylem, s rozdílem prohození dámy a krále, k tomu může pomoci rčení „Dáma ctí barvu“ (bílá dáma na bílém políčku, černá na černém).

Poté se připraví šachové hodiny, bílý táhne některou svojí figurou a přepnutím tlačítka na hodinách na svoji stranu spustí měření času (u některých hodin stiskne tlačítko protihráče). Tím se zahájí hra a hráči se střídají v provádění svých tahu, než nastane jedna z podmínek nutných k ukončení hry.

Síla jednotlivých šachových figur se liší jejich možnostmi tahů, s výjimkou krále, který i přes svojí omezenou pohyblivost je nejdůležitější figurou. Pokud je král napaden, nastává **šach**. V tu chvíli musí hráč

- Uhnout s králem na pozici, kde není napaden
- Ochránit krále jinou figurou jejím položením mezi útočící figurou a svého krále (nelze použít v případě napadení koněm)
- Napadající figuru sebrat

V případě že takovýto tah není možný nastane **mat**, a partie končí vítězstvím pro hráče který takto krále napadl. Z těchto pravidel vyplývá, že ve skutečnosti není možné soupeřova krále sebrat.

Může nastat i situace, kdy král "hráče na tahu" není v šachu, ale nemůže provést žádný tah tak, aby šach nenastal, a ani jiná jeho figura nemůže provést žádný tah. V tom případě nastane **pat**. Tak případě končí partie nerozhodně.<sup>[5]</sup>

## Pohyb figur

Pro pohyb figur platí, že se mohou pohybovat pouze na pole, na kterých není již postavena figura stejného hráče. Dále nesmí ostatní figury přeskakovat. Každý figura může zabrat soupeřovu figuru postavením své figury na pole, kde se nachází daná figura soupeře. Soupeřovu figuru tak odstraní z herního pole. Některé figury mají různé výjimky, ty jsou popsány níže.

- *Král* - pohybuje se do všech osmi směrů, maximálně však na vzdálenost jednoho pole. Hráč ve svém tahu nemůže krále postavit na pozici pod soupeřovou kontrolou, protože by se vystavil šachu. Speciálním pohybem je **rošáda**. V tom případě se král posune o dvě pole směrem k jedné z věží, následně hráč danou věží krále přeskočí, a umístí ji na pole vedle něj. V tomto případě se ještě rozlišuje malá a velká rošáda (malá rošáda se provádí s věží na královské polovině šachovnice, velká s věží na dámské půlce). Rošáda se může provádět pouze v případě, že mezi králem a věží nejsou žádné další figury, s králem a danou věží se ještě netáhlo, a král před provedením, a ani po provedení není v šachu (nesmí ani procházet polem, kde by v šachu mohl být) viz obr. 3.1 .
- *Dáma* - stejně jako král se může pohybovat všemi osmi směry, s tím rozdílem, že nemá omezenou vzdálenost pohybu. To z ní činí nejsilnější herní figuru.
- *Střelec* - pohybuje se pouze po diagonálách o libovolný počet polí. Jelikož hráč má dva střelce, jeden z nich se může pohybovat pouze po bílých polích a druhý naopak pouze po černých.
- *Věž* - pohybuje se pouze po sloupcích a řádcích o libovolný počet polí
- *Kuň* - pohybuje se skokem ve tvaru písmene L (jedním směrem o jedno poličko a druhým o dvě). Díky tomu jako jediná figura může přeskakovat ostatní kameny.
- *Pěšec* - pohybuje se pouze směrem dopředu a to o jedno pole. Pokud je v základním postavení, může se posunout o dvě pole vpřed, pokud jsou obě pole prázdná. Soupeřovy figury může zabrat pohybem vpřed o jedno pole, ale úhlopříčně. Dále může brát **mimochodem**. To znamená zabrat soupeřova pěšce, který se ze základního postavení posunul o dvě pole vpřed a tím přeskočil ohrožené pole tímto pěšcem viz obr. 3.1. Toto braní lze provést pouze bezprostředně po pohybu braného pěšce. Poslední vlastnost pěšce je **proměna**, ta nastane pokud se pěšec dostane až na konec herní desky (osmá řada pro bílého nebo první řada pro černého hráče). V tom případě se nahradí za dámu, věž, střelce nebo koňe, podle volby hráče.

## 3.2 Šachová notace

Dříve se používalo více druhů šachové notace, nyní se nejvíce používá *algebraická šachová notace*. [4] Jednotlivá pole se v ní označují kombinací sloupce (hodnoty a-h) a řádku (hodnoty 1-8).

Popis samotných tahů se skládá z čísla tahu, označení figury, kterou se táhlo, a jeho cílového pole. Číslo tahu označuje jeden tah bílého i černého hráče (tah pouze jednoho je v této notaci tzv. *půl tah*). V případě, že by nešlo jednoznačně určit počáteční pozici figury, uvede se i označení sloupce počátečního pole, a to před označením cílového pole. Figury se



Obrázek 3.1: Speciální šachové tahy Postavení před rošádou, postavení po rošádě a příklad braní mimochodem

značí počátečním písmenem ( **K** pro krále, **D** pro dámu, **S** pro střelce, **V** pro věž). U pěšce se označení figury nepíše.

Pro vyznačení zabrání figury se uvede **x** mezi označení figury a cílové pole (bere-li pěšec, uvede se místo znaku figury počáteční sloupec). Pro značení **rošády** se používá **0-0** pro malou a **0-0-0** pro velkou rošádu. Pro braní mimochodem se jako cílové pole uvede to, kam se pěšec skutečně přesunul (za tah lze připsat **e. p.**).

Pokud je daným tahem **šach**, přidá se na konec zápisu **+**. Pokud je tahem **mat** přidá se na konec zápisu **#**. Při **proměně** pěšce se uvede označení figury, ve kterou se proměnil (opět na konec zápisu).

Samotná notace dále obsahuje značení výhry a remízy a povoluje značení komentářů (často pomocí zavedených značek např. pro sílu tahu).

Příklad zápisu matu čtvrtým tahem by mohl vypadat takto :

1. e4 e5
2. Dh5?! Jc6
3. Sc4 Jf6??
4. Dxf7# 1-0

Z tohoto zápisu lze vidět, že druhý tah bílého byl lehce pochybný, a třetí tah černého byl označen za hrubou chybu.

## Kapitola 4

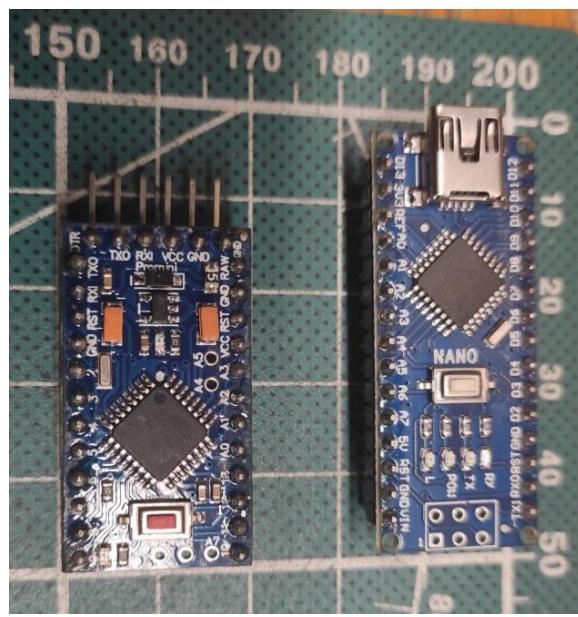
# Hardwareový návrh

### 4.1 Řídící jednotky

#### Arduino NANO

Je nízko úrovňová řídící jednotka založená na mikrokontroleru ATmega od firmy Atmel. Na rozdíl od jeho menší varianty Arduina Mini má tato deska na sobě i USB port a převodník, tím je rozměrově trošku větší, ale odpadá nutnost použití převodníku při nahrávání programu. Porovnání viz obr. 4.1. Procesor na této desce může být ve dvou provedeních, a to ATmega 328p (častější) a ATmega 168. Dále je třeba zkontolovat, zdali se jedná o 5V nebo 3,3V variantu.

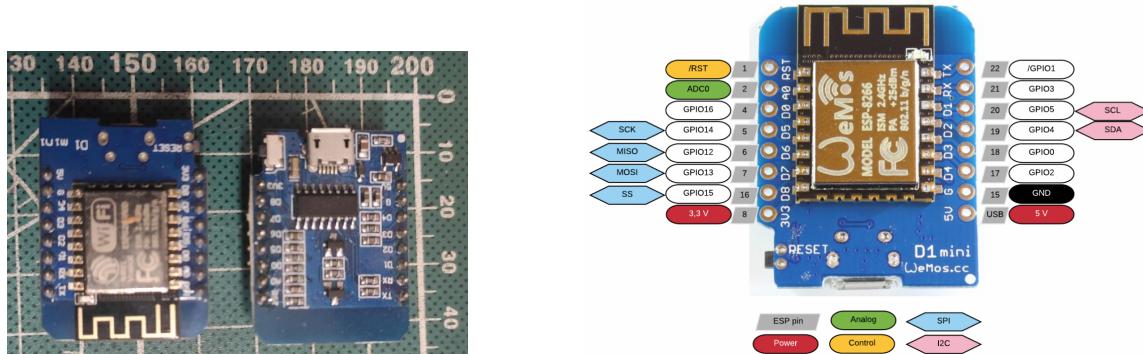
Jelikož je tento kontrolér plánován jako slave jednotka, tak jeho výkon je dostačující, a poskytuje i stejný počet datových kontaktů, proto není potřeba ani použít Arduino UNO ani Arduino Mega.



Obrázek 4.1: **Arduino Nano a Arduino Mini** - porovnání velikostí

## WeMos R1 mini

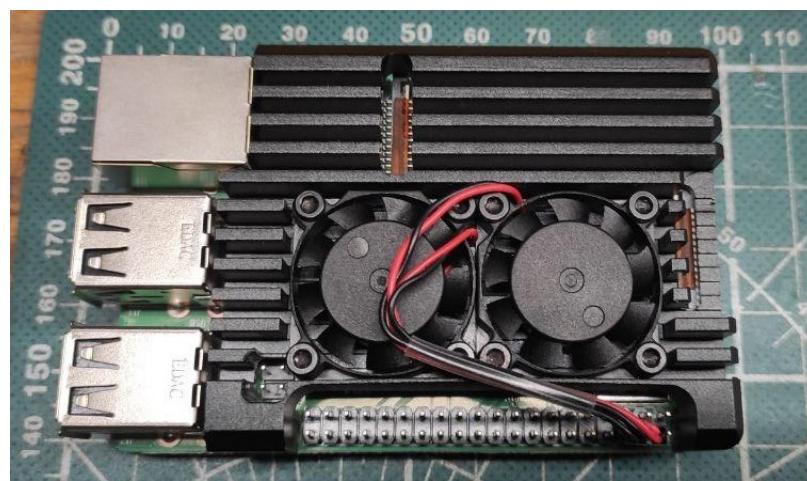
Oproti Arduinu je tato deska postavena na kontroléru ESP8266, který má 32-bit architekturu. Obsahuje vestavěnou Wi-Fi. Takt procesoru je ve variantách 80MHz (základní) nebo 160MHz. Modul je ukázán na obr. 4.2, včetně popisů funkcí jednotlivých kontaktů.



Obrázek 4.2: **WeMos R1 mini** - pohled zespodu a shora, popis kontaktů převzato z <https://www.esp8266.com/viewtopic.php?f=12&t=17077>

## Raspberry Pi

Tento jedno-deskový počítač je protipólem výše zmíněných kontrolérů. V závislosti na jeho verzi nabízí plný 32-bit nebo 64-bit operační systém a velký výpočetní výkon. Vnitřní úložiště (závisející na vložené microSD kartě) může být až 256GB. Díky těmto parametrům je vhodný zejména jako hlavní řídící jednotka. I přes vestavěné GPIO jsou tyto kontakty vodně spíše na komunikaci s jinými moduly než pro samotné řízení těchto IO pomocí Raspberry. Některé modely se také mohou při dlouhém běhu přehřívat, to dá vyřešit přidáním vetráčků pro lepší cirkulaci vzduchu viz obr. 4.3.



Obrázek 4.3: **Raspberry Pi 3b+** v pouzdru s externím chlazením

Mezi nevýhody Raspberry Pi patří jeho velká spotřeba tudíž by byl vhodný výkonnější zdroj energie. Ovšem i tento problém by šel vyřešit. Nevýhoda, která mě přiměla k tomu abych tento počítat nepoužil, je jeho čas spouštění systému. Z osobní zkušenosti vím že

Raspberry Pi 3b+ a Raspberry Pi 4b 4 GB RAM nabíhají alespoň 45 s, byť tato data mohou být zkreslena operačním systémem, případně programy, které na nich používám (zejména ty, co se pouští hned po nastartování). Tento problém lze redukovat, ovšem ne zcela eliminovat, tím, že by se zkompiloval vlastní minimalistický operační systém, nebo vypnutím některé služby. Toto jsem ve svých projektech ještě nepotřeboval, tudíž se musím spokojit s informacemi z internetových fór, kde lze dohledat dosažený čas spouštění systému kolem 15 s.

## Raspberry Pi Zero

Tento jednodeskový počítač je zmenšenou a méně výkonnou podobou Raspberry Pi 3. Ve verzi W/B má v základu i Wifi připojení a Bluetooth. Co se týče rozložení kontaktů tak je totožné s deskou Pi 3 / Pi 4. Velkou výhodou vůči klasické Raspberry Pi je jeho nízká spotřeba, a proto se často uplatní i v projektech stavěných na bateriovém napájení.

## 4.2 Detekce figur

U detekce figur jsem se zaměřil zejména na technologie, které hráče nebudou omezovat tím, že by vyžadovaly velmi přesné pozicování hracích figur (natočení vložení do slotu apod.). Samozřejmě sem se podíval i na technologie, které toto vyžadují, pro případ, že by ostatní možnosti byly nevyhovující.

Samotné testování spolehlivosti jednotlivých metod bylo poté hlavním kritériem pro výsledný výběr pro vytvoření prototypů finálního výběru metody, kterou použiji v samotné šachové desce (viz kapitola 7).

### RFID

Bezdrátová technologie, která slouží ke čtení hodnot na tagu (čip, který obsahuje 96 bitů dat s unikátním identifikátorem). V závislosti na použité čtečce (různé frekvence, rozdílné antény atd.) je schopen číst data ve vzdálenosti 1–10 cm, výjimkou jsou čtečky s vysokou frekvencí, kde je čtení možné až v řadu metrů. Standardní frekvence jsou 125 kHz, 134 kHz a 13,56 MHz. Tagy mohou být jak pouze pro čtení, tak i přepisovatelné.

Největší výhodou této technologie je schopnost detektovat nejen přítomnost figurky, ale i její typ.

Celkové hodnocení je tudíž výborné a s touto technologií bude dále experimentováno.

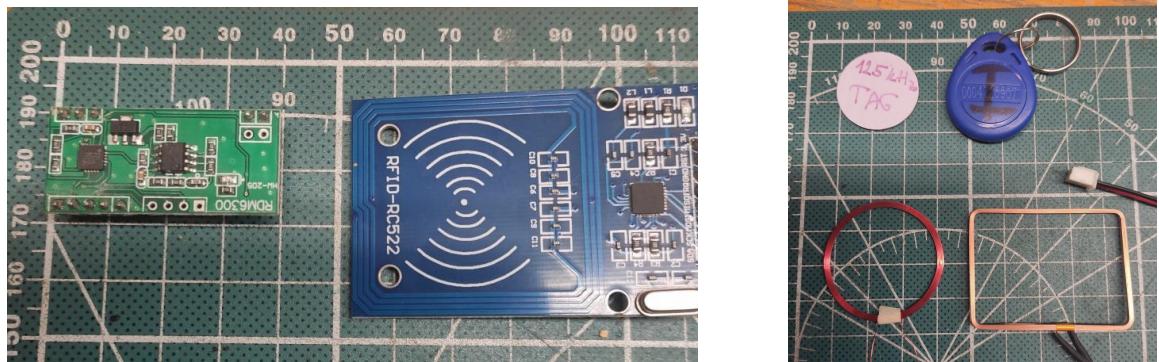
### NFC

Bezdrátová technologie podobná RFID s rozdílem frekvence čtečky - 13,56 MHz. Vzdálenost čtení je lehce menší než u RFID - obvykle do 4 cm. Velkou nevýhodou je, že jsem nenalezl čtečku s externí anténou, která je nutností pro použití v tomto projektu. Na obrázku 4.4 je porovnání RFID a NFC čtečky.

Celkové zhodnocení technologie je tudíž nevyhovující.

### Jazýčkové relé

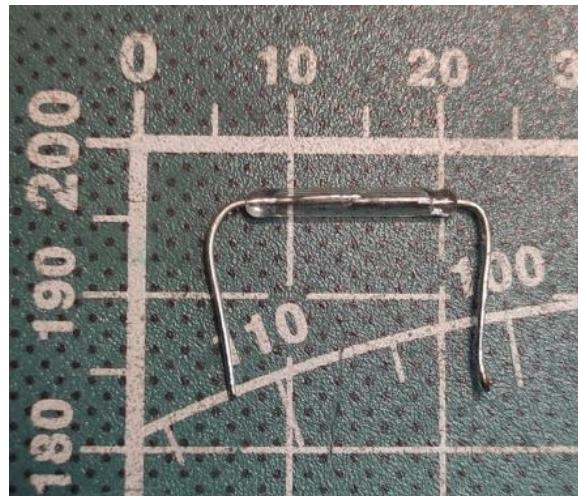
Elektromagnetický spínač tvořený dveřmi slabými kovovými proužky, které se v přítomnosti magnetů sepnou viz obr. 4.5. Jejich spolehlivá vzdálenost čtení je dána silou magnetu,



Obrázek 4.4: **Součástky pro použití RFID/NFC technologie** Na levé straně RFID čtečka pro připojení externí antény a NFC čtečka, vpravo externí antény a tagy

výrobní kvalitou, náklonem magnetu a vzdáleností magnetu od relé. Při základních testech jsem dosáhl spolehlivého čtení pro vzdálenosti 1–3 cm, testoval jsem i různé varianty naklonění relé vůči pozici magnetu viz obr. 4.6.

Celkové hodnocení je tudíž přijatelné a bude s touto technologií dále experimentováno.



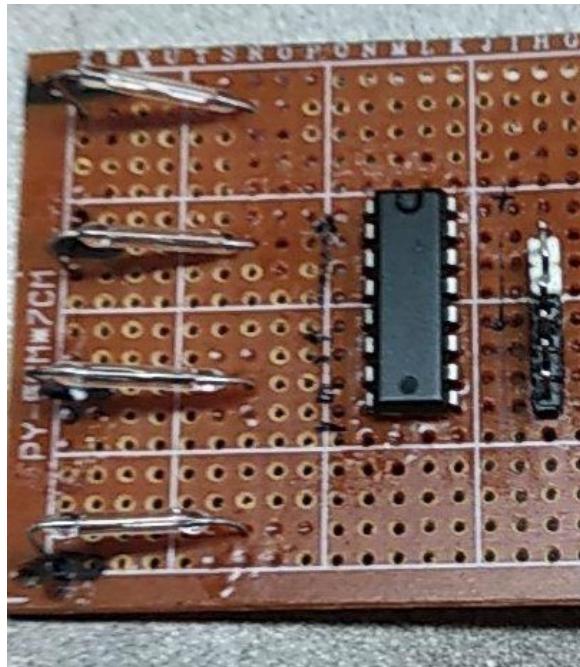
Obrázek 4.5: **Jazýčkový spínač ve skleněném pouzdru**

### Halova sonda

Technologie využívající tzv. Halova jevu. Tato elektronická součástka se používá pro měření magnetického pole. V této součástce je úzká polovodičová deska, která v magnetickém poli přeskujuje náboj na jednu stranu, a tím vzniká napětí.

Teoreticky by se za použití různě velkých magnetů a přesného pokládání figur na specifickou pozici na herním poli dalo detekovat kromě přítomnosti i typ figur, ale ladění pro přesné a opakovatelné čtení by bylo složité. V případě pouhé detekce přítomnosti by možná bylo toto řešení spolehlivější než jazýčkové relé, ale s ohledem na cenu a zapojení je výhodnější použití právě jazýčkového relé.

Celkové zhodnocení je tudíž nevyhovující.



Obrázek 4.6: **Test úhlu čtení** - ukázka testování čtecí vzdálenosti při náklonu  $45^\circ$  u horního relé až po  $0^\circ$  u spodního

## Kontakty

Použitím kontaktů v tomto projektu je myšleno buď použití nějakého konektoru (v rámci rozmýšlení byl zvažován například 3,5 Jack), nebo vytvoření nějakého systému, který by obsahoval dvě kovové nebo jinak vodivé plošky na hrací ploše, a ty by byly spojovány pomocí propojky v šachové figuře. Samotná šachová figura by obsahovala druhou část daného konektoru a odporu, který by pak pomocí čtení analogové hodnoty jednoznačně určil figurku.

Problémem této metody je jak to, že vyžaduje zvýšenou přesnost při pokládání figurek na herní pole, tak to, že výroba figur by byla řádově složitější a pravděpodobně by nebylo možné použítí kupovaných figurek.

Celkové zhodnocení technologie je tudíž nevyhovující.

## Tlakový senzor

Metoda velmi podobná předešlé, s tím rozdílem, že spínač se sepne přitlačením figur na spínač schovaný v herní desce, v závislosti na citlivosti spínače buď je zapotřebí figurku po odehrání na plochu přitlačit, a tím potvrdit pozici, nebo při použití citlivějších senzorů by mohla stačit samotná váha šachové figurky. Jelikož tlakové senzory, které jsem našel, stály přibližně kolem 200 Kč za kus, zkoušel jsem hledat alternativu. Jednou z nich bylo použití membránových tlačítek. Ta by ale požadovala daný tlak figury na políčko. Dále jsem nevěděl, jak bych realizoval políčka tak, aby byla dostatečně citlivá na sepnutí, a zároveň zamezil spínání i vedlejších polí.

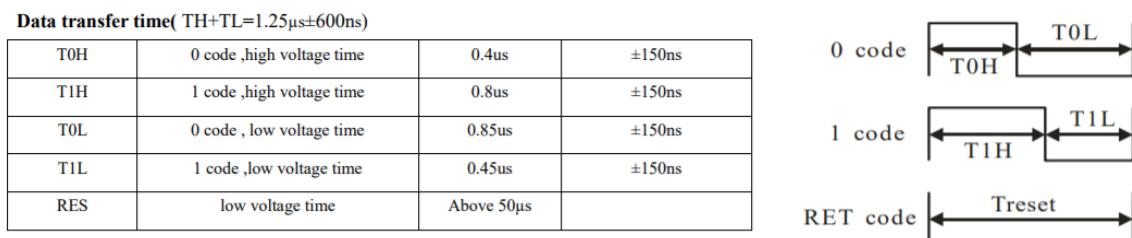
Celkové zhodnocení technologie je tudíž nevyhovující.

### 4.3 Signalizace a UI

Signalizací je v tomto projektu myšleno jak zobrazení jednotlivých tahů na herní desce, tak případné zaznamenání a zobrazení času, nastavení a podobně. Proto Základní myšlenou pro signalizaci bylo použití světelné signalizace přímo v desce, a dodatečného použití nějakého displeje na zobrazení ostatních dat.

#### RGB LED pásek WS2812B

Tyto RGB LED diody mají výhodu v tom, že k nastavování jejich barev stačí jeden datový kontakt. Po něm se přenáší 3B dat pro každou diodu. Z tohoto datového toku si každá načte data pro nastavení své barvy a zbytek posílá do dalších LED diod. Tato komunikace je postavená na přesném časování jednotlivých pulzů viz obr. 4.7. Konec datového toku signalizuje prodleva  $50\mu s$ . Při plném jasu všech tří barev je odběr až 60mA při napětí 5V.



Obrázek 4.7: Parametry časování půlzů LED diod WS2812B - převzato z [7]

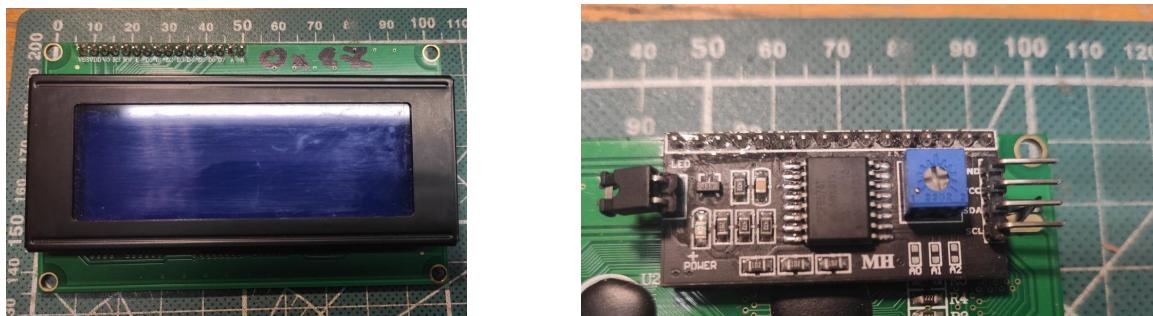
Pro lepší manipulaci jsem pásek rozdělil na jednotlivé diody viz obr.4.8



Obrázek 4.8: Dioda Ws2812b

#### LCD Display s I2C řadičem

Tento display (viz obr.4.9), který se dá sehnat především ve velikostech 16\*2 znaků a 20\*4 znaků, existuje ve dvou variantách, a to bez nebo s řadičem HD44780. Bez tohoto řadiče je výsledné schéma zapojení mírně složitější. Osobně používám Tento display pouze s tímto řadičem, protože umožňuje posílání dat do tohoto displeje pomocí komunikace I2C.



Obrázek 4.9: LCD display a řadič HD44780

### TFT Display

Jedná se o klasický pixelový LCD displej. Je dostupný ve velikém množství velikostí a rozlišení viz obr. 4.10. Samotná obsluha displeje je výpočetně složitější a náročnější na paměť (veškeré obrázky musí být uloženy v datových polích s informací o barvách a podobně).



Obrázek 4.10: LCD TFT displeje - ukázka různě velikých displejů

## 4.4 Ostatní použité součástky

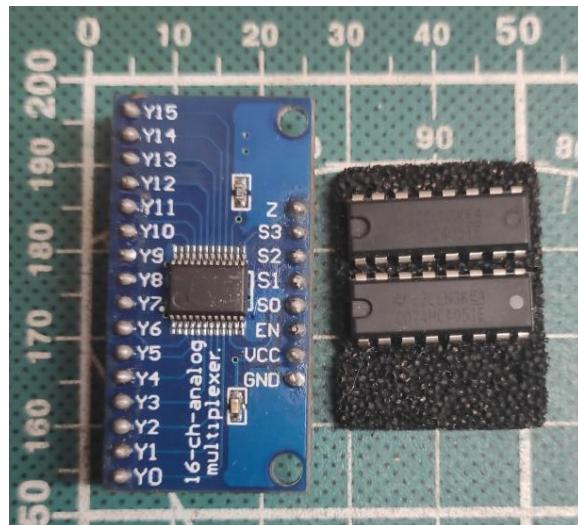
Níže je seznam a popis ostatních součástek, které byly použity k výrobě šachovnice. Jedná se o multiplexory viz. obr. 4.11, převodník logických úrovní a ostatní součástky pro napájení šachovnice obr. 4.14.

### Multiplexory

Protože jsem původně potřeboval multiplexovat RFID antény, potřeboval jsem analogový multiplexor, který by byl schopný pracovat na frekvenci 125kHz. Nejprve jsem našel mul-

tiplexor CD74HC4051[2]. Jedná se o osmi kanálový multiplexor v provedení DIP<sup>1</sup> a z datových listů se zdálo, že by měl být vhodný pro dané použití.

Následně jsem našel multiplexor CD74HC4067[3]. Tento multiplexor byl na rozdíl od prvního uvedeného šestnácti kanálový v provedení SOIC<sup>2</sup> a podle datových listů umožňoval práci na vyšších frekvencích. Později jsem našel i takový, který byl v provedení DIP (lépe řečeno, před připravenou desku v provedení DIP, na které byl osazen tento multiplexor).



Obrázek 4.11: Multiplexory - z levé strany multiplexor CD74HC4067 varianta SOIC a CD74HC4051 ve variantě DIP

Oba multiplexory se obsluhují obdobně, pomocí zadání požadované adresy na kontaktech S0-Sn. Přivedením napětí na daný kontakt nebo jeho uzemněním se zvolí daná adresa a na společném datovém kontaktu lze přečíst data. Při přivedení napětí na kontakt EN se na všech výstupech nastaví logická nula. Přehled výběru kanálu je na obrázku 4.12.

### Převodník logických úrovní

Tento převodník se používá pro bezpečné propojení s různým napětím (například komunikace zařízení pracujícího na 5V a zařízení pracujícího na 3,3V logice). V tomto projektu je použit čtyři kanálový převodník, ale existuje i osmi kanálový. Podporovaná napájení jsou 5V, 3,3V, 2,8V a 1,8V. Převodníky v obou variantách jsou videt na obrázku 4.13.

### Baterie 18650

Jedná se o nabíjecí baterie s různě velikou kapacitou (mnou použité Samsung ICR18650 mají kapacitu 2600mAh). Používají se zejména kvůli jejich velké kapacitě, životnosti a snadnému použití. Samotné tyto baterie nemají ochranu proti zkratu, přetížení ani přepětí. Proto se používají zejména v bateriových balíčcích (power-banky a podobně), ve kterých je obsažena elektronika, která tyto problémy hlídá.

<sup>1</sup>DIP- Dual in-line package (pouzdro se dvěma řadami pinů pro osazování skrz desku)

<sup>2</sup>SOIC- Small outline integrated circuit (v porovnání s DIP zabírá přibližně 30% velikosti, pájí se na desku, ne skrz ni jako u DIP)

TRUTH TABLE

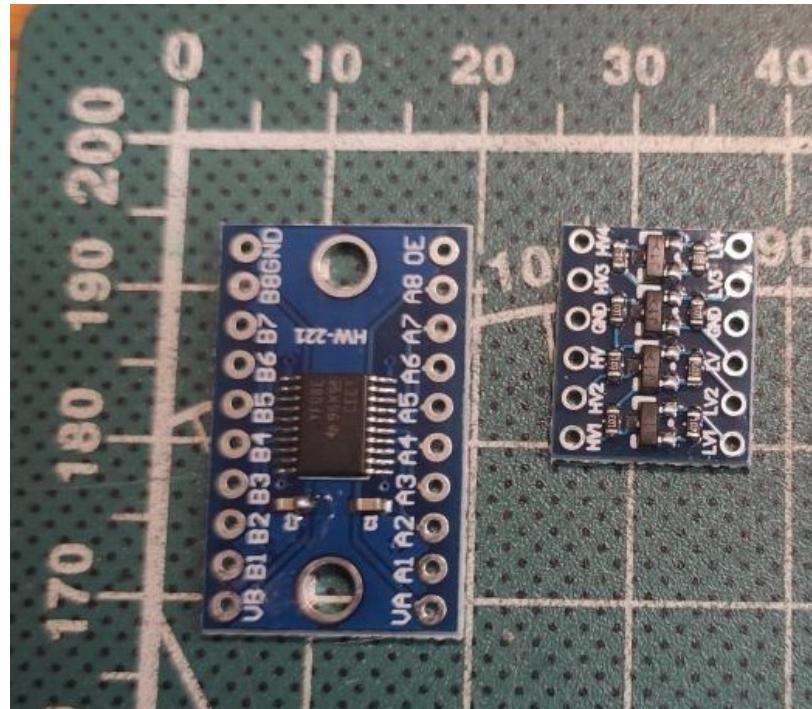
S0	S1	S2	S3	$\bar{E}$	SELECTED CHANNEL
X	X	X	X	1	None
0	0	0	0	0	0
1	0	0	0	0	1
0	1	0	0	0	2
1	1	0	0	0	3
0	0	1	0	0	4
1	0	1	0	0	5
0	1	1	0	0	6
1	1	1	0	0	7
0	0	0	1	0	8
1	0	0	1	0	9
0	1	0	1	0	10
1	1	0	1	0	11
0	0	1	1	0	12
1	0	1	1	0	13
0	1	1	1	0	14
1	1	1	1	0	15

H= High Level

L= Low Level

X= Don't Care

Obrázek 4.12: Pravdivostní tabulka nastavení adresy kanálu u multiplexoru CD74HC4067 převzato z [3]



Obrázek 4.13: Převodníky logických úrovní - vlevo osmi kanálový a vpravo čtyř kanálový

### Nabíječka Li-Pol baterií 1S

Tento modul slouží pro nabíjení jedné lithium-polymerové baterie. Obsahuje dvě indikační LED diody - červenou pro signalizaci nabíjení a modrou pro signalizaci plného nabití. Nabíjení probíhá skrze micro USB konektor nebo svorky (pod napětím 5V). Baterie se připojuje na svorky BAT+ a BAT-, výstup je na svorkách OUT+ a OUT-.

### Step-UP měnič MT3608 - nastavitelný

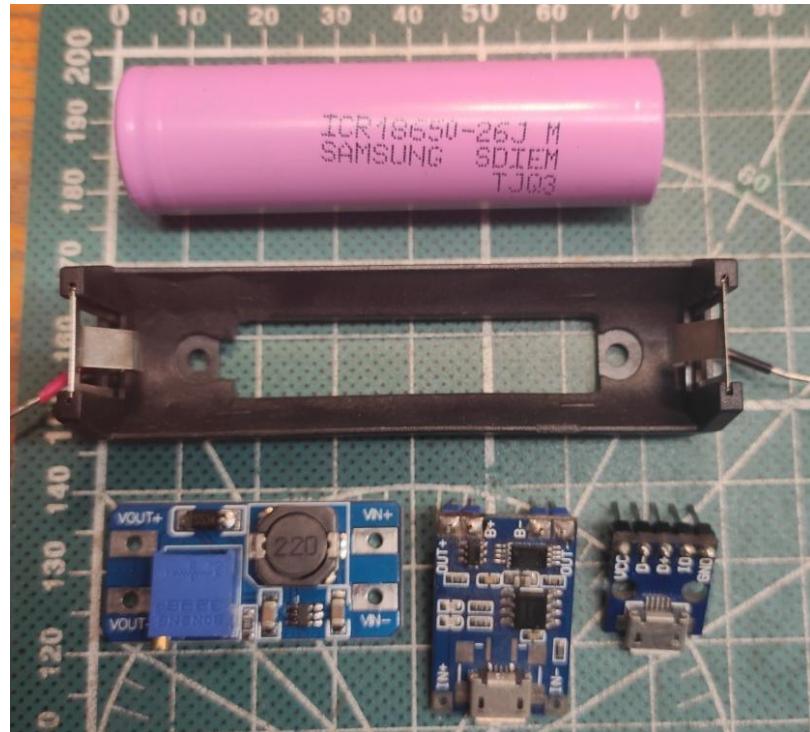
Modul, který mění nižší vstupní napětí na vyšší výstupní. V případě, že vstupní napětí je vyšší než napětí nastavené na výstupu, může dojít k poškození řídícího čipu. Na vstup lze přivést napětí v rozsahu 2-28V, na výstup lze nastavit až 28V. Maximální výstupní proud je 2A.

Kvůli některým speciálně spínaným zdrojům se doporučuje použít 100nF filtrační kondenzátor na vstupu i výstupu.

## 4.5 Funkční model šachovnice

### Přehled modelu

Při návrhu modelu jsem již byl rozhodnut o tom, že použiji jedno Arduino na ovládání šachových hodin, WeMos jako hlavní řídící jednotku, a jedno až dvě Arduina jako jednotky řídící čtení herní pozice a signalizaci tahu pro hráče. Z toho vyplývalo, že jednotlivé moduly spolu budou muset komunikovat. Až na samotné LED diody jsem použil všechny komponenty v provedení through-hole, neboli komponenty které prochází skrz desku, protože práce



Obrázek 4.14: Součástky použité pro napájení šachovnice - v horní polovině baterie 18650, spodní řádek zleva step-up modul, jedno-článková nabíječka Li-pol baterií a micro USB konektor

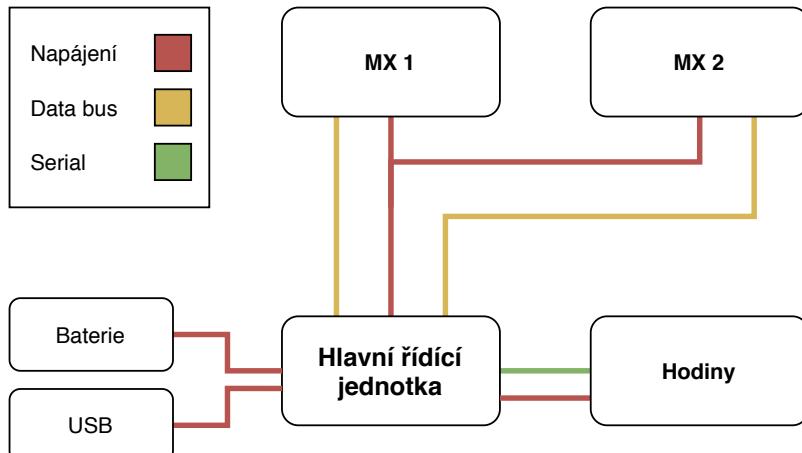
s nimi je jednoduší, byť mají většinou větší rozměr. Napájení je řešeno buď pomocí USB pro napájení ze sítě, nebo pomocí 18650 baterií. Model je zobrazen na obrázku 4.15.

### Základní řídící deska

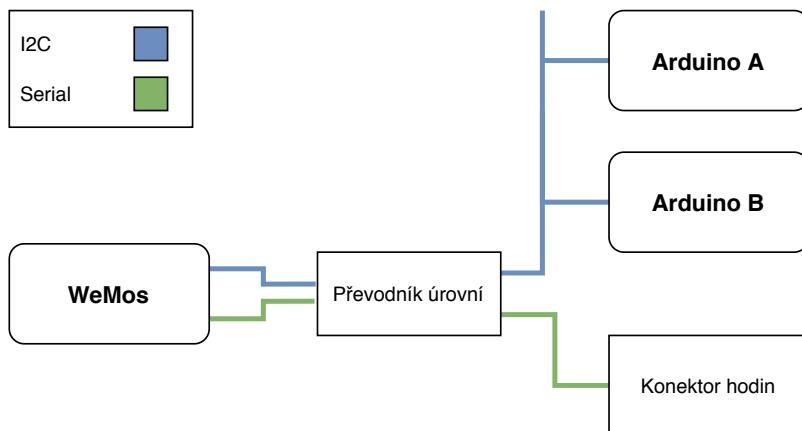
Kvůli omezené maximální velikosti desky, kterou jsem schopen doma vyrobit, jsem musel oba Arduino slavy umístit na hlavní řídící desku. Tudíž se tato deska lehce zkomplikovala, ale v celkovém důsledku se celý model zjednodušil. Dále na této desce je hlavní jednotka, převodník logických úrovní, kvůli převodu z 5 V na 3,3V pro komunikaci mezi jednotkami. Dále je pak na této desce pouze pár odporů (pull-up a pull-down) a konektory. Blokové schéma je na obrázku 4.16.

Z hlavní jednotky vedou pouze komunikační trasy, jedna pro obsluhu I2C s pull-up rezistory na obou vodičích a sériová komunikace. Tyto trasy vedou do převodníku napětí. Z něj vede I2C komunikace paralelně do obou Arduin, opět s přidanými pull-up odpory. Sériová komunikace vede na konektor pro hodiny.

Zapojení obou slayvů je totožné až na kontakt D12, které je u jednoho přivedeno k zemi a u druhého k napětí. Kontakty D7 a D8, které slouží jako společný kontakt z multiplexovací desky jsou pomocí pull-down rezistoru staženy k zemi, Arduino sice má na některých kontaktech tyto rezistory již umístěné, ale raději používám externí. Ostatní kontakty jsou zapojeny přímo na konektory pro multiplexovací desky. Kontakty pro adresování multiplexorů a pro povolující kontakt jsou pro oba multiplexory stejné, tudíž jsou napojeny na stejná I/O na Arduinu (D2-D6)



Obrázek 4.15: Blokový model šachovnice



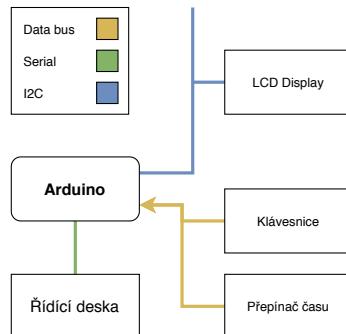
Obrázek 4.16: Blokový model řídící desky

### Desky pro obsluhu hodin a LCD

Modul hodin v minimální konfiguraci musí obsahovat některý z testovaných displejů, tlačítka na řízení tahu jednotlivých hráčů a ovládací tlačítka kvůli. Hlavní desku v hodinách tvoří rozbočovací deska s Arduinem a jednotlivými konektory. Dva jsou pro obsluhu tlačítek (šest tlačítek a napájení), další zajišťuje komunikaci s LCD displejem po I2C a poslední propojuje hodiny s hlavní deskou (Sériová komunikace a napájení). Dále je v hodinách deska s tlačítky namontovaná na přední panel vedle LCD. Deska s tlačítky přepínání hráčů je přichycena k horní straně šachových hodin. Blokové schéma je na obrázku 4.17.

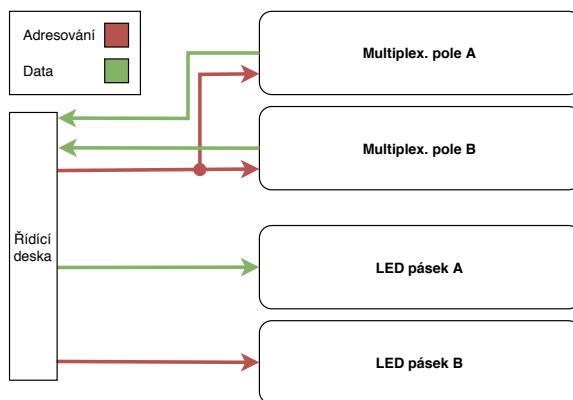
### Deska multiplexového pole

Při vytváření této desky jsem byl omezen nejen samotnými komponentami a jejich velikostí či počtem, ale i tím, že tato deska musela pozíci komponent kopírovat samotné šachové délky. Na začátku jsem tyto desky kreslil vycentrované vůči těmto délkám, ale kvůli šířce sběrnice od jazýčkových relé k multiplexoru jsem nakonec zvolil mírně nevycentrovanou verzí desky. Obě desky tedy obsahují třicet dva LED diod a jazýčkových relé. Dále pak konektory pro multiplexory a ostatní řídící spoje. Některé návody doporučují při zapojení



Obrázek 4.17: Blokový model hodin

více WS1285b použít kondenzátor pro kompenzaci změn v odběru proudu při změně baryvy. Ty jsem při návrhu na desku umístil, ale při vytváření desky jsem se rozhodl je nepoužít, protože při testování jsem je nepoužil a celý pás normálně fungoval (viz kapitola 7). Blokové schéma je na obrázku 4.18.



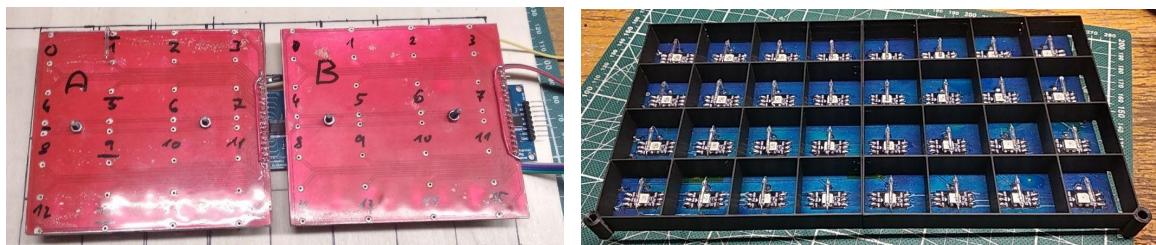
Obrázek 4.18: Blokový model multiplexového pole

Je nutné zmínit, že adresy relé a ledék jsou obrácené vůči sobě (relé A0 odpovídá LED diody A15). Toto přeadresování jsem vyřešil v kódu otočením zapisování barev pro jednotlivé LED diody.

Tato deska prošla největším počtem změn a několika vyrobenými návrhy (obr. 4.19). Většinou se jednalo o zvětšení tolerancí mezi trasami pro korektní výrobu desky. Ovšem výsledná deska se od původního návrhu liší tím, že ten používal samostatné desky pro LED diody (vždy po osmi kusech) a dvě desky pro multiplexory. Toto řešení sice umožňovalo vyrobit více desek a mít pár náhradních osazených desek v případě rozbití některého relé nebo LED diody, ale samotné adresování bylo více komplikované a desky se hůře připevňovaly k podložce, což vedlo k občasné horší detekci figur.

### Zapojení baterií

Nabíjení baterií je řešeno pomocí nabíjecího 1S modulu. Z něj je propoj do step-up konvertoru, protože deska je stavěná na 5V napájení a zvolené baterie 18650 pracují na 3,7V (nabité). Na spojích mezi šachovnicí a step-up modulem a mezi modulem a bateriemi jsou ještě tlačítka pro odpojení baterií od desky. To umožňuje jak desku vypínat, tak nabíjet



Obrázek 4.19: **Různé varianty multiplexovací desky** - z levé strany starší varianta s rozdělením načítání dílků a signalizací a novější deska kombinující obě části do jedné desky

baterie, jelikož by se z nich neměl v průběhu nabíjení odebírat proud, aby nedošlo k jejich poškození.

# Kapitola 5

## Výroba šachovnice

Jelikož většina této šachovnice je tvořena od základních komponent (pominu-li kupované součástky jako multiplexovací deska, a řídící jednotky), byla samotná výroba velmi náhylná na přesnost a kvalitu kvůli případným chybám ve výrobě. Toto komplikovalo zejména fázi výroby PCB, kdy jsem někdy nevěděl, jestli hledám chybu v návrhu nebo chybu výrobní.

### 5.1 Návrh PCB v programu Eagle

Jelikož jsem nechtěl výslednou desku mít na několika nepájivých polích nebo na velkém množství prototypovacích desek, musel jsem navrhnut samotné desky. K tomu jsem použil program Eagle, který je v současnosti vlastně společností Autodesk. Jeho základní neplacená verze je ovšem omezená ne velikost vytvářené desky na rozměry 100\*100 mm, což pro mou desku byl problém, protože i při použití čtyř desek na vytvoření detekce dílů bych potřeboval desky 120\*120 mm. Tady mě zachránila studentské licence, i když aktivace této licence byla mírně problémová (několikrát se nepodařilo k účtu připojit studentskou licenci ani s pomocí technické podpory).

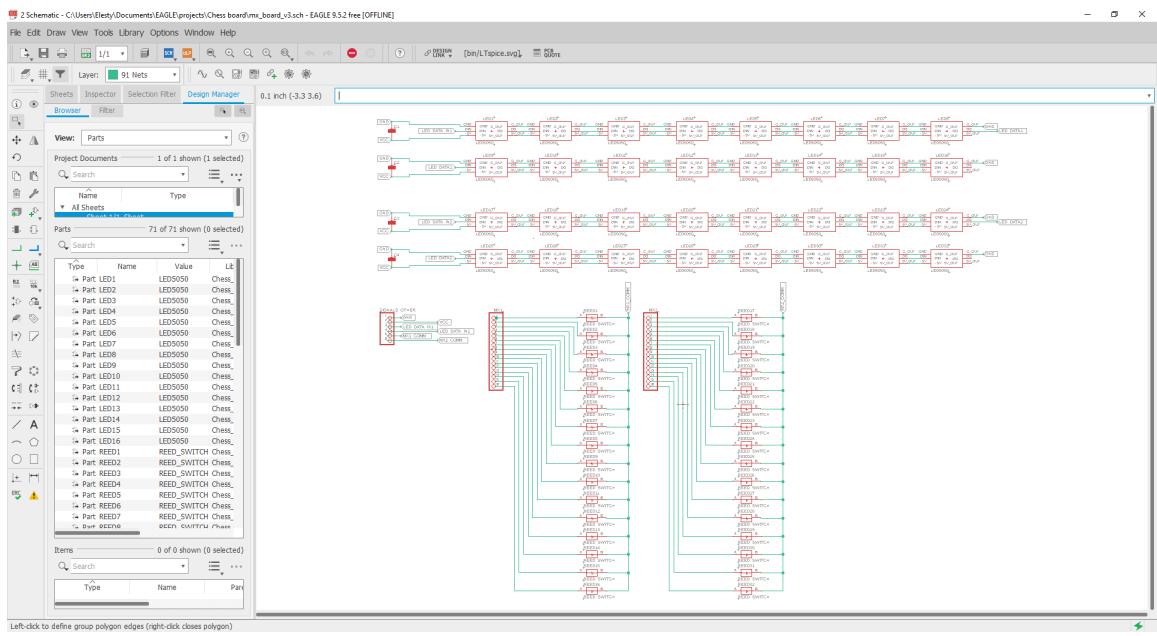
Samotný návrh desky se skládá ze dvou procesů

- Vytvoření schématu (obr. 5.1)
- Vytvoření desky (obr. 5.2)

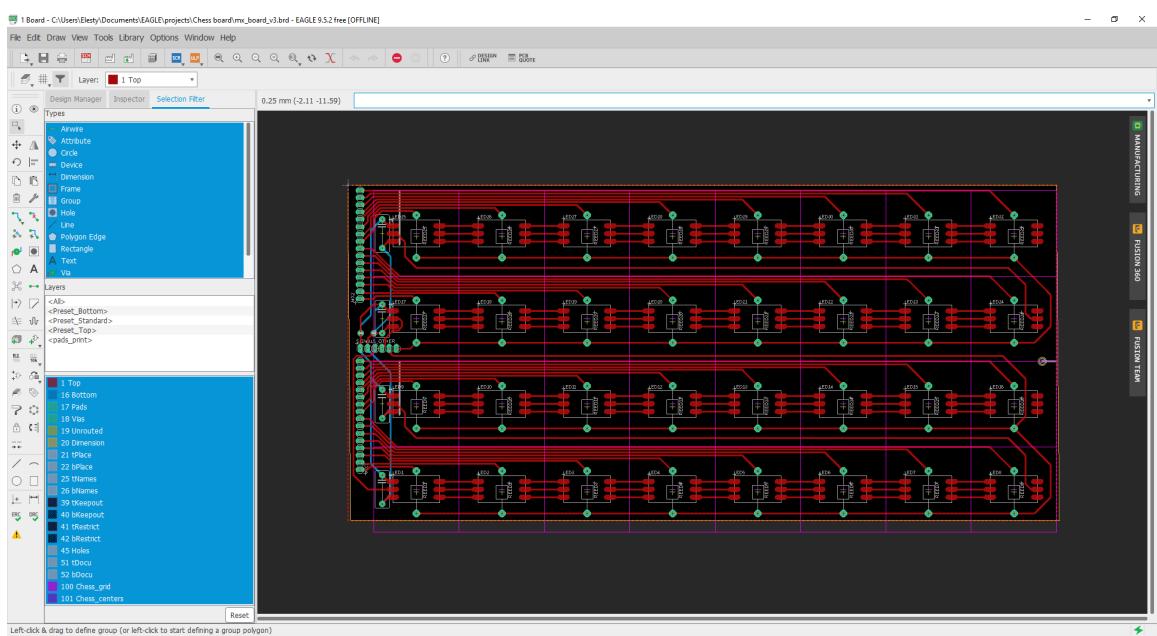
Při vytváření schématu se definují veškeré použité komponenty a propoje mezi jednotlivými kontakty. Tento proces je relativně jednoduchý, protože není potřeba definovat přesnou pozici jednotlivých komponent, ani se neřeší rozměry. Příklad schématu desky je na obrázku 5.4.

Pro snadnější práci se speciálními komponenty lze v tomto programu vytvořit knihovnu, která obsahuje definice jak jejich kontaktů, tak samotného vzhledu na desce. Toho jsem využil pro definici komponent jako jsou jazýčková relé, Arduino NANO a ostatní.

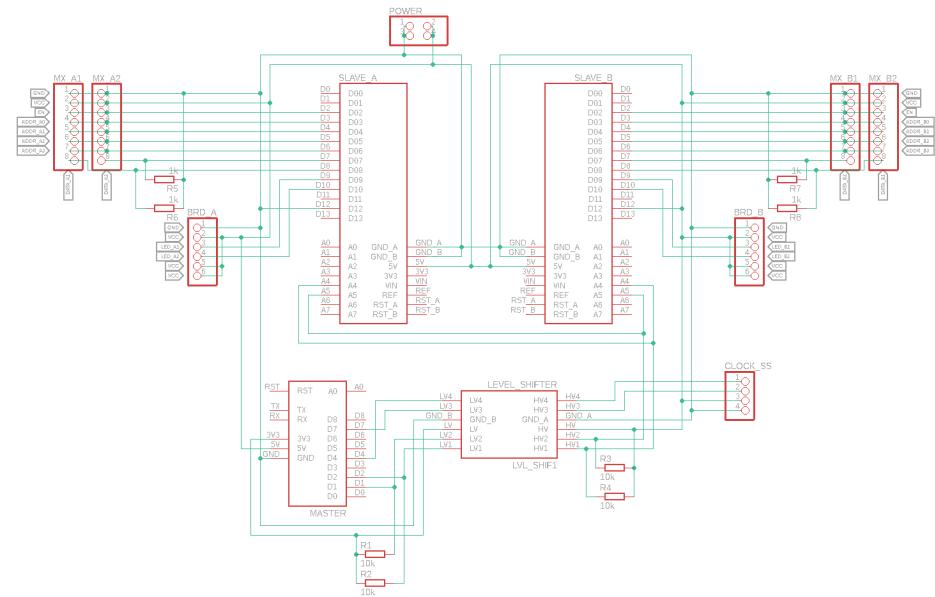
Ve druhé části jsem nejdříve přibližně rozmištěl komponenty, přičemž jsem se snažil co nejvíce eliminovat křížení tras. Dalším krokem bylo definovat jednotlivé tolerance (minimální šířka tras, minimální rozestupy tras, vzdálenosti průchodek atd.) Dále jsem začal kreslit jednotlivé trasy, většinou od nejkratších. V posledním kroku jsem upravil velikost desky, případně přidal díry pro uchycení, a vytvořil oblast pro tzv. Rastnet, který vytvoří na celé desce uzemněnou plochu. Příklad navržené desky je na obrázku 5.4.



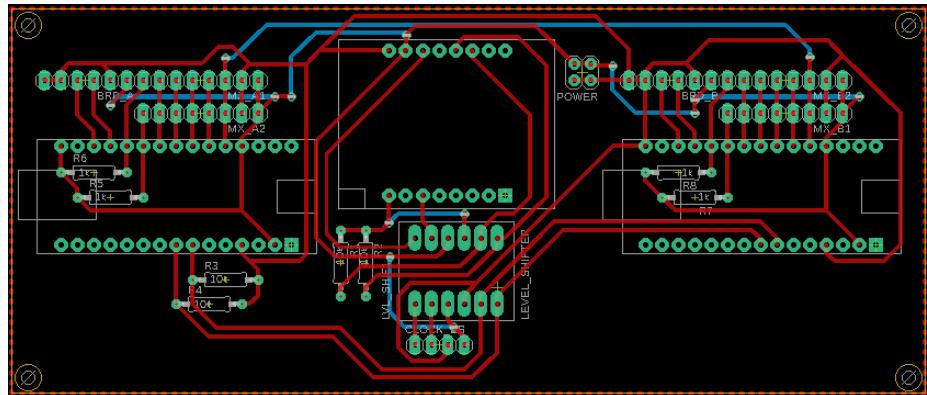
Obrázek 5.1: Program Eagle - režim vytváření schémat



Obrázek 5.2: Program Eagle - režim návrhu desky



Obrázek 5.3: Schéma zapojení hlavní řídící desky

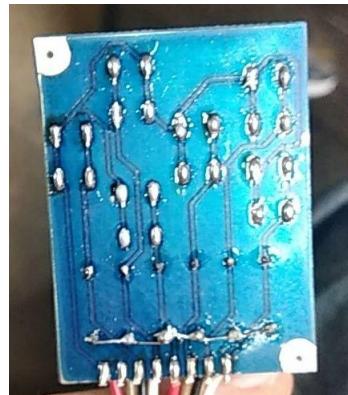


Obrázek 5.4: Výsledná podoba hlavní řídící desky

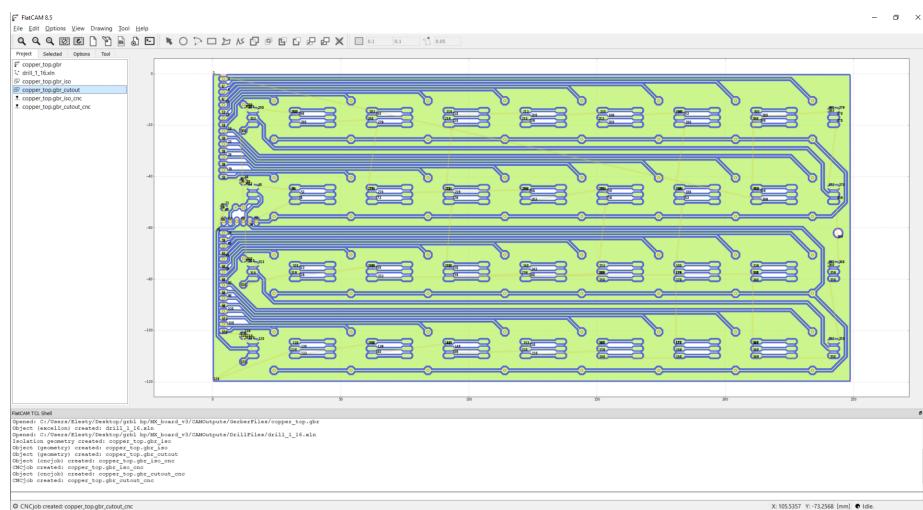
Toto jsou všechny kroky, které jsem dělal při navrhování jednotlivých desek. Poté se deska pouze vyexportovala (některé desky bylo třeba ještě zrcadlově otočit) a byla připravená k výrobě.

## 5.2 Výroba a testování PCB

Samotná výroba plošných spojů probíhala pomocí metody vyřezávání desky (vyrobená deska na obrázku 5.5). Pro leptání desek jsem neměl dostatek předešlých zkušeností a potřebné chemické prostředky pro samotný proces. Výroba desek na zakázku pomocí různých internetových služeb (příkladem JLC PCB) by byla pro jednotlivé prototypy finančně nákladná i pro pouhou výrobu finálních desek, protože desky multiplexovacích polí mají velký rozměr a cena pouze těchto dvou samotných desek by byla okolo 2000 Kč včetně poštovného (výroba je většinou minimálně po 5ks na typ, takže za jednu desku 400Kč s tím, že by se 3 pravděpodobně nepoužily). Další nevýhodou by pak bylo i prodloužení výrobního procesu.



Obrázek 5.5: Deska vyrobená pomocí metody vyřezávání spojů



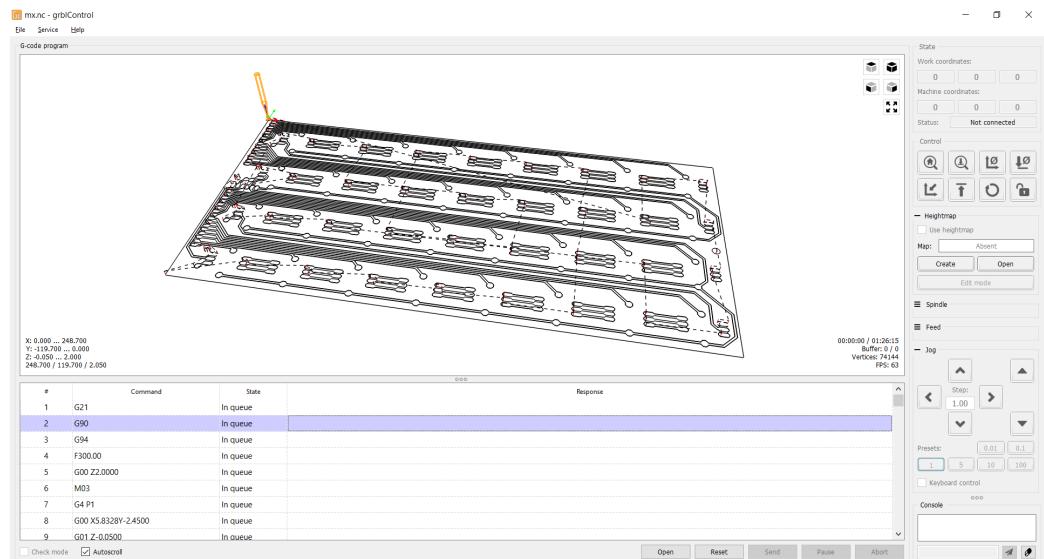
Obrázek 5.6: **FlatCam** - ukázka desky připravené exportu do programu GRBLControl

Pro použití CNC je potřeba z daného modelu desky vytvořit soubor instrukcí. K tomu používám program FlatCam, který umí vytvořit kolem jednotlivých tras dráhy, umí zpracovávat excelon soubory s definicí děr a generovat víceúrovňové dráhy na vyříznutí desky z opracovávaného materiálu (obr. 5.6). Pro každý z těchto kroků se generuje jeden grbl soubor (přípona .nc), který se následně nahrává do CNC.

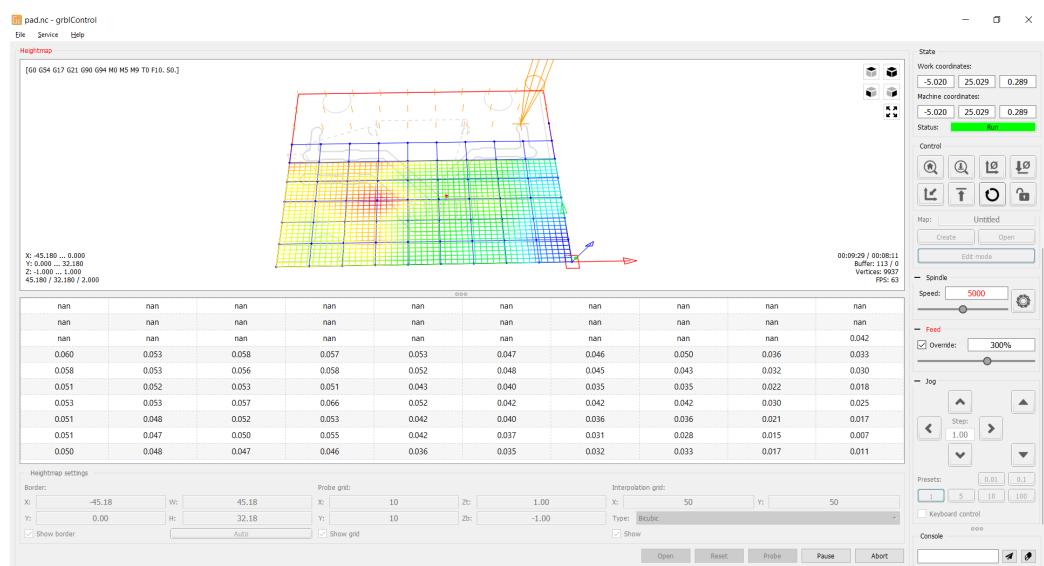
Pro nahrávání grbl dat do CNC používám GRBLControl (obr.5.7). V tomto programu také vytvářím výškové mapy cuprexitové desky, jelikož tyto desky jsou většinou nerovné a bez dostatečně přesného korigování těchto nepřesností jsou výsledky převážně nepoužitelné (obr.5.8).

Takto vyrobená deska následně projde první kontrolou tras. Dále je na ni nanesena ochranná UV maska, případně dojde k postříbření kontaktů (obr. 5.9). Před pájením je vhodné desku znova otestovat na případné spojené dráhy (vlivem stříbření), byť se mi to zatím nestalo.

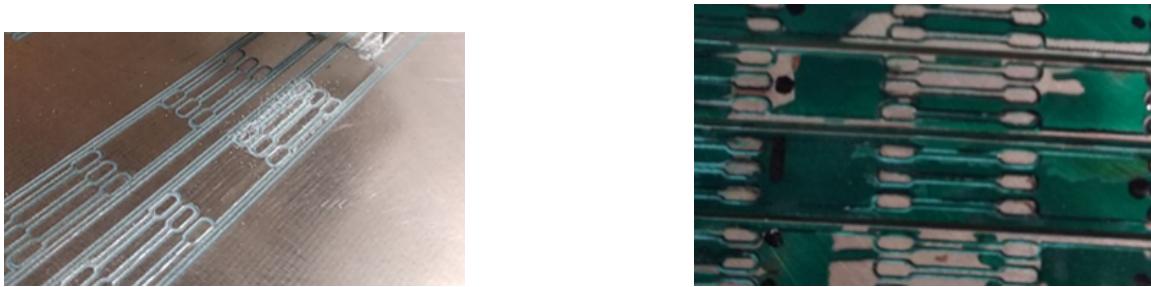
Po těchto krocích je deska připravena k samotnému napájení součástek. Celý proces výroby v závislosti na velikosti desky zabere přibližně 5 hodin u jednodušších a až den u složitějších desek.



Obrázek 5.7: GRBLControl - ukázka desky připravené k exportu do CNC



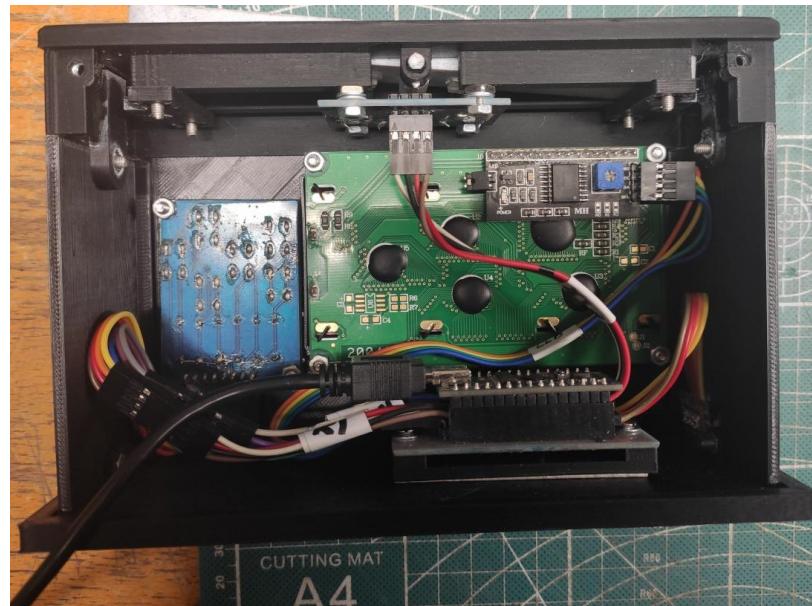
Obrázek 5.8: GRBLControl - vytváření výškové mapy



Obrázek 5.9: Jednotlivé kroky výroby plošného spoje z leva vyřezaná deska, deska před napájením komponent

### 5.3 Výroba ostatních částí šachovnice

Jelikož se výsledná šachovnice skládá ze dvou modulů a více menších PCB, musel jsem vytvořit na jednotlivé moduly obal, který je drží ve stabilním stavu (nejvíce patrné na modulu hodin, který se skládá z devíti tištěných částí viz obr 5.10). Pro modul šachovnice bylo třeba zajistit pevnou pozici pro multiplexovací pole kvůli zajištění přesnosti čtení a pro zafixování hlavní řídící jednotky. Základní desky jsou z překližkové desky, na kterou jsou následně přichyceny jednotlivé tištěné díly a plošné spoje. Překližkové desky jsou řezány ručně, proto není jejich přesnost nijak velká, ale po zajištění jednotlivými plastovými díly je to dostačující.



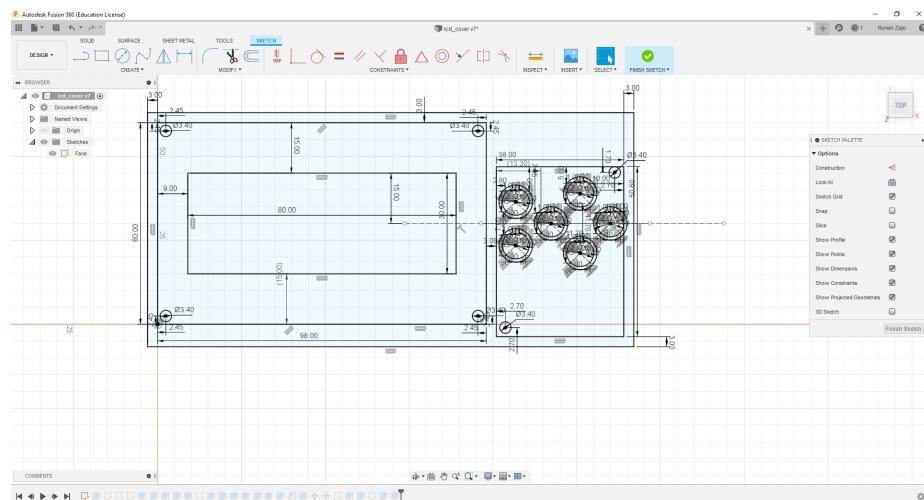
Obrázek 5.10: Modul hodin - bez zadního krytu

Plastové díly jsou 3D tištěny a veškeré 3D modely jsou vytvořeny specificky pro tuto šachovnici. Pro vytváření těchto modelů jsem použil program Fusion 360 od Autodesku, jelikož jako student mám nárok na výukovou licenci a oproti ostatním programům se mi s ním dělá nejlépe.

## Fusion 360

Tento program je celkově nazván „All in one“ program pro 3D modelování a přípravu modelů pro další použití, od 3D tisku (případně CNC obrábění), až po různé simulace namáhání jednotlivých součástek nebo jejich vizualizace. I přesto, že některé pokročilejší funkce používám v jiných osobních projektech, tady jsem ho použil především pro vytváření modelů na tisk. Pro přípravu těchto modelů musí člověk pamatovat na to, že 3D tisk je vlastně nanášení tenkých plastových čar na sebe, tudíž musí tomuto dané modely přizpůsobit. Hlavním pravidlem je snažit se vyvarovat velkým nebo složitým převisům, případně je upravit tak, aby šlo použít podpěry, které pak lze vyjmout (v některých modelech jsem na to zapomněl, a buď po vytisknutí podpěry nešlo vyjmout, čímž například blokovaly otvor na šrouby, nebo se model korektně nevytiskl a daný model bylo třeba předělat). Dále pak je třeba započítat i tolerance mezi jednotlivým díly, jelikož tisk je na úrovni desetin mm nepřesný (na mé tiskárně mám tyto tolerance odzkoušené na rozmezí 2-3 desetin mm).

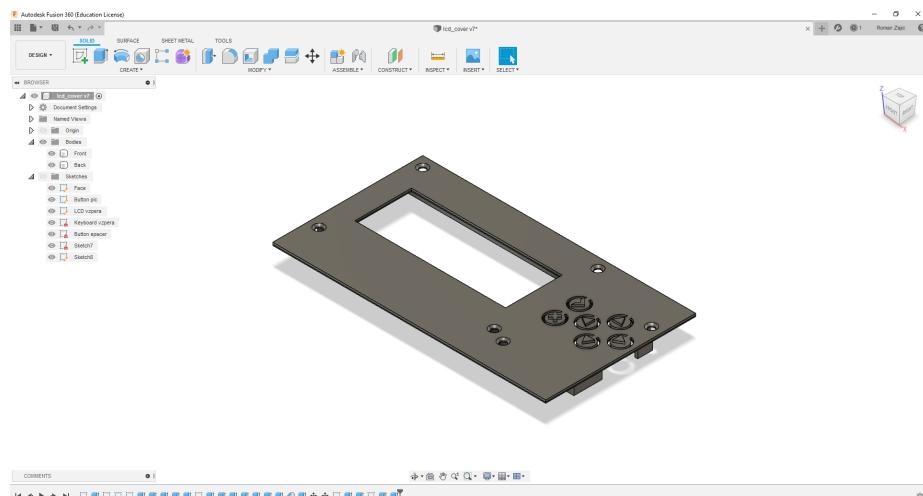
Pro samotný návrh modelu stačí v této práci zmínit, že nejprve se nakreslí nějaký základní tvar daného objektu (obr. 5.11, se kterým se dále pracuje a postupně se zesložuje) (obr. 5.12).



Obrázek 5.11: Fusion 360 - náčrt přední strany hodin

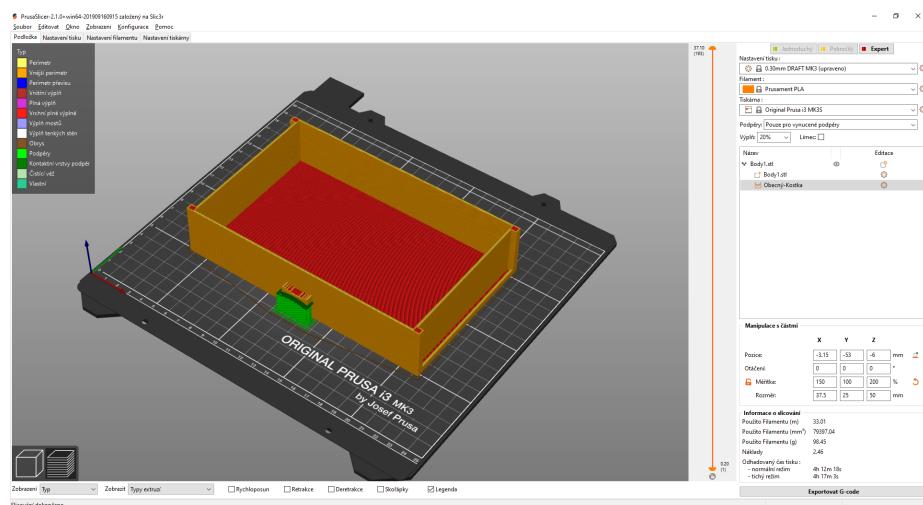
## PrusaSlicer a nastavení tisku

Pro přípravu vymodelovaných součástek k tisku je potřeba program nazývaný slicer, který model rozdělí na jednotlivé tiskové vrstvy (dle nastavené výšky, která ovlivňuje velikost detailu, pevnost a další parametry výsledného modelu). Jelikož vlastním tiskárnou Prusa i3 MK3s, tak používám jejich předpřipravený slicer PrusaSlicer (ve verzi 2.1.0 pro win 64) na obrázku 5.13 je vidět model připravený k tisku. Jednotlivá nastavení se mírně liší pro jednotlivé modely a materiál použití k tisku daného modelu (většina dílů je z PLA, páry z PETG kvůli lepším vlastnostem). Samotné ovládání je vcelku intuitivní a neuděláte velkou chybu při použití přednastavených profilů, které jsou součástí sliceru. Osobně tisknu při vyšších teplotách podložky kvůli přilnavosti jednotlivých dílů k podložce. Některé profily mám upraveny na rychlejší tisk (byť se ztrátou detailů a drobných tiskových nepřesností). Pouze páry modelů v této práci potřebují jiné nastavení. U modelu rámu kolem desek multiplexovacích polí je potřeba přidat v rozích modifikátor vynucení podpěr kvůli převisu



Obrázek 5.12: **Fusion 360** - výsledná podoba předního krytu hodin ve 3D náhledu

u držáku matky na připevnění k podkladové desce. U některých modelů dále doporučuji zapnout možnost límce (brim) pro zvětšení kontaktní plochy s tiskovou podložkou (rám hodin a podobné). Model rámu hodin doporučuji tisknout naležato, pro mírné zvětšení kontaktní plochy a pro eliminaci vysoké tenké stěny (ve vyšších rychlostech může docházet k tiskovým artefaktům v ose x nebo y a tím k případnému problému při sestavování modulu hodin).



Obrázek 5.13: **PrusaSlicer** - ukázka modelu připraveného k tisku (úložný prostor na figury)

# Kapitola 6

## Implementace šachů

### 6.1 Vývojové prostředí, komunikace a knihovny

#### ArduinoIDE

ArduinoIDE je víceplatformní vývojové prostředí určené pro práci s Arduino kompatibilními kontroléry. Umožňuje komplikaci a nahrávání programů do daných desek, případně vypalování zavaděče na tyto desky. Dále obsahuje správce knihoven pro jednoduché použití vestavěných a uživatelských knihoven a manažera desek, který umožnuje stáhnout aktualizované definice Arduino desek nebo balíčky s jinými deskami. Takto se dá stáhnout právě definice desek na bázi ESP8266 nebo ESP32 desek. Mezi další funkce patří sériový monitor sloužící ke čtení zpráv z jednotlivých desek a případné zaslání příkazu z počítače.

Pro získání definic desek na bázi ESP8266 je potřeba tyto definice stáhnout. Pro to je potřeba ArduinoIDE říct, kde dané definice hledat. V záložce Soubor > Vlastnosti > Správce dalších desek URL je potřeba vložit:

- [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

Poté samotné desky lze stáhnout v manažeru desek (Nástroje > Vývojová deska > Manažer desek) vyhledáním "ESP8266". V projektu se použily definice desek verze 2.6.3.

Jednotlivé knihovny se dají aktualizovat nebo stáhnout v manažeru knihoven (Nástroje > Spravovat knihovny). V projektu byly použity tyto externí knihovny:

- **FastLed** - autor Daniel Garacia<sup>1</sup>, verze 3.3.2
- **LiquidCrystal I2C** - autor Frank de Brabander<sup>2</sup>, verze 1.1.2

#### Wiring

Je programovací jazyk podobný C/C++ uzpůsobený pro jednoduchou práci s kontroléry na bázi Arduina.

Každý program používá dvě hlavní funkce a to setup() a loop(). Funkce *setup()* se volá na začátku programu a slouží k přípravě Arduina před během hlavního programu. Funkce *loop()* se cyklicky volá a většinou tvoří hlavní tělo programu.

<sup>1</sup><https://github.com/FastLED/FastLED>

<sup>2</sup>[https://github.com/johnrickman/LiquidCrystal\\_I2C](https://github.com/johnrickman/LiquidCrystal_I2C)

## I2C a knihovna Wire

Je multi-másterová synchronní komunikace vyvinutá firmou Philips Semiconductor a používá se pro připojování nízkorychlostních zařízení k základní desce na krátké vzdálenosti, a to až pro 128 zařízení. Samotná komunikace používá dva vodiče SDA (datový signál) a SCL (časový signál). Pro oba datové vodiče je potřeba použít  $10\text{ k}\Omega$  pull-up rezistor (tato hodnota postačí, protože velikost rezistoru se odvíjí od frekvence -  $F_{req}$  a od celkové kapacity na dané lince -  $C_{bus}$ ). Ovšem je lepší použít vysokého odporu kvůli ztrátám. V datovém list k čipu ATmega168 [1] lze dohledat vzorec pro přesný výpočet minimální a maximální hodnoty odporu (proměnné  $R_{min}$  a  $R_{max}$ ). Jelikož je odpor závislý na frekvenci komunikace má dvě varianty, pro pomalou(6.1) a rychlou(6.2) komunikaci. Proměnná  $V_{cc}$  udává provozní napětí.

$$Freq < 100\text{kHz} \implies R_{min} = \frac{V_{cc} - 0.4\text{V}}{3\text{mA}}, R_{max} = \frac{1000\text{ns}}{C_{bus}} \quad (6.1)$$

$$Freq > 100\text{kHz} \implies R_{min} = \frac{V_{cc} - 0.4\text{V}}{3\text{mA}}, R_{max} = \frac{300\text{ns}}{C_{bus}} \quad (6.2)$$

Samotná implementace je taková, že master jednotka buď zasílá data jednotlivým slayvů nebo požaduje zaslání dat určité velikosti. Na straně slayva se definují funkce, které tuto požadavky zpracovávají (definuje se funkce pro oba typy požadavků, která se automaticky volají při daném požadavku)

## UART a knihovna Serial

Jak z názvu vypovídá jedná se o knihovnu pro sériovou komunikaci mezi dvěma zařízeními. Pro Arduino je připravena stejně pojmenovaná knihovna, která se převážně používá pro komunikaci s PC (debug apod.). Arduino i WeMos mají přímo připravené kontakty s hardwarovou podporou pro tuto komunikaci, případně lze použít USB na desce. Ovšem tyto kontakty a USB jsou propojeny. Proto se často používá i knihovna SoftwareSerial, která umožňuje použití i jiných kontaktů za cenu zmenšení paměti RAM (protože každá komunikace si zabere určité veliký prostor pro implementaci zásobníku na příchozí data). V případě použití více SoftSerial na více kontaktech (každá z nich potřebuje dva kontakty) je potřeba přepínat mezi aktivní linkou, jelikož tato knihovna podporuje aktivní naslouchání pouze na jedné z nich. Nakonec jsem nakonec zvolil, že tato komunikace bude použita pouze mezi WeMosem a Arduinem v hodinách, a to v provedení SoftwareSerial.

## Knihovna FastLed

Tato knihovna umožňuje jednoduchou manipulaci s různými typy adresovatelných LED diod. Protože LED diody WS1284b jsou řízeny protokolem pracujícím s délkou jednotlivých signálů, samotná implementace je tvořena použitím vloženého assembleru a používá zejména instrukci NOP<sup>3</sup>. Knihovna má několik definic barev podle užitých LED diod (WS2812B používá definici GRB<sup>4</sup>)

<sup>3</sup>NOP - No operation (prázdná instrukce)

<sup>4</sup>GRB - pořadí barev pro LED diody WS2812B

## Knihovna LiquidCrystal I2C

Jedná se o vylepšení knihovny LiquidCrystal<sup>5</sup>. Na rozdíl od ní probíhá komunikace po I2C, jinak je práce s ní téměř totožná.

## 6.2 Slave pro řízení multiplexovací desky

Tento modul cyklicky načítá z multiplexovaacího pole pozice figur. Jelikož jeden modul pracuje pouze s polovinou desky a je schopen pouze detekovat přítomnost, tak vnitřní reprezentace je pomocí pole čtyř bytů, ve kterém každý bit představuje jednu pozici.

Podobná reprezentace je i pro řízení LED diod. Kromě pole RGB hodnot (tří bytová struktura definovaná knihovnou FastLed), kam se ukládá aktuální barva pro daný dílek, používá i stejná čtyř bytová pole pro ukládání požadavku na vykreslení barvy. Tato pole jsou plněna daty z master jednotky, z výjimkou pole pro ukládání přítomnosti figur. Příchozí komunikace je ve formátu 1B pro určení typu dat a 4B pro jednotlivé pozice na desce. Odchozí komunikace má stejný formát.

## 6.3 Hodiny

Jelikož hlavní funkce hodin je hlídání času, tak je informace o tom, kolik ho jednotlivým hráčům zbývá neustále ukazována. K tomu používám první řádek LCD displeje.

Druhý a třetí řádek má více funkcí. V základní slouží pro zobrazení herních tahů (používám lehce upravenou verzi šachové notace viz 6.4). Modul si drží informaci vždy maximálně o čtyřech tazích a v případě potřeby podá požadavek na další tahy masterovi. Dalším módem je menu a nastavení. Mezi jednotlivými módy se přepíná pomocí příznakové proměnné, která se přepisuje na základě stisklých tlačítek na modulu. Podobně jako u multiplexovacího modulu je komunikace ve formátu 1B pro určení typu dat, poté 1B pro určení délky a nB pro samotná data (parametr n je totožný s druhým parametrem a je spíše pro kontrolu). Odchozí komunikace má opět stejný formát.

Poslední řádek slouží pro výpis ostatních informací (příkladem je potvrzení proměny pěsce a podobně).

Kvůli korektnímu zpracování komunikace a uživatelských vstupů jsem se snažil psát program neblokující formou (tzn. snažil jsem se vyhnout použití dlouhých cyklů, vestavěné funkce delay() apod.)

## 6.4 Hlavní řídící jednotka

Při startu šachovnice nahraje nastavení z paměti a tato data pošle do modulu hodin. Dále si připraví zahajovací rozložení figur, a podle dat z hodin a multiplexovacích modulů zjistí, jestli byla zahájena hra (přitom signalizuje, jestli detektuje přítomnost figur na zahajovacích pozicích).

Hlavní jednotka v pravidelných intervalech načítá data z multiplexovacích modulů. Tato data zpracuje (detekuje, jestli některé pole změnilo svůj stav) a případně generuje možné tahy figury na dané pozice. Poté posílá zpět data o barvách na jednotlivých LED světlech.

<sup>5</sup>Vestavěná knihovna - pro ovládání LCD používá přímo datové kontakty na displeji

V případe proměny pěšce zažádá pomocí modulu hodin o zadání figury, ve kterou byl pěšec proměněn. Pokud hráč figuru před překlopením nezadá, předpokládám, že proměna byla v dámu.

Bohužel, celou logiku hlavní jednotky jsem nezvládl dopsat. Chybí implementovat herní logiku (detektovat figuru se kterou se táhne, vykreslit její možné pohyby atd), s tím, že generování tahů je hotové (s výjimkou rošády). Dále není hotové generování a zápis upravené šachové notace do modulu hodin.

Aktuální program na začátku vykresluje, požadavek na základní rozestavení figur a když přijde příznak o začátku hry, tak pouze vykresluje umístění figur. Vykreslování možných tahů je v kódu vypnuto.

### Upravená šachová notace

Oficiální šachovou notaci (popsána v kapitole 3) jsem mírně zjednodušil pro jednodušší použití. K označení figur používám první písmena anglických názvů figur, a to **K** pro krále, **Q** pro královnu, **B** pro střelce, **N** pro koně, **R** pro věž a přidávám ještě označení **P** pro pěšce. Následuje standardní příznak zabrání figury v podobě **x**, poté je uvedena jak původní, tak cílová pozice figury. Ostatní zápisu zůstávají stejné s výjimkou malé a velké rošády. Ty se zapisují pomocí **O** a **o** jako příznak na konci pohybu krále.

## Kapitola 7

# Testování a výroba prototypů

Samotné testování bych rozdělil do několika kategorií

- Základní testování jednotlivých metod
- Testování vyrobených desek - popsáno v kapitole 5.2
- Testování funkčnosti jednotlivých modulů
- Celkové testování všech modulů

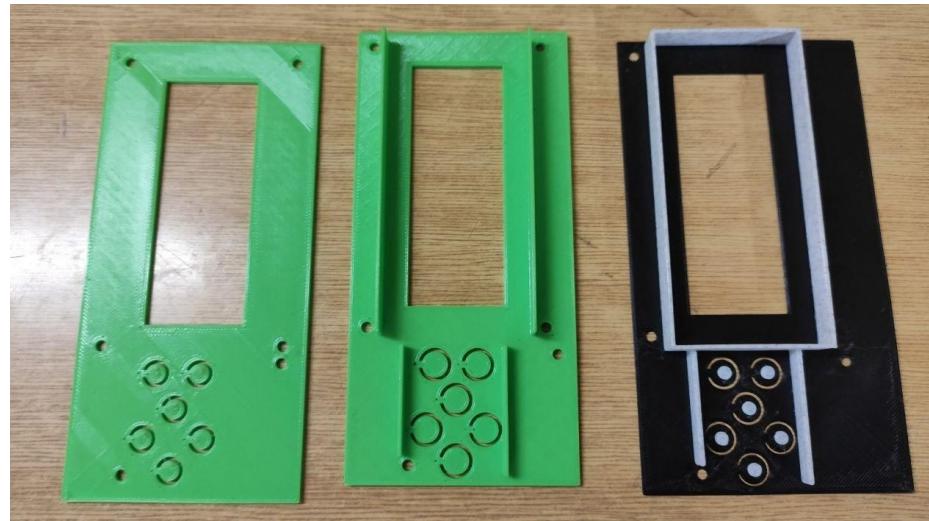
### Základní testování jednotlivých metod

Toto testování probíhalo v průběhu výběru komponent a jeho cílem bylo zvolit takové komponenty, které půjde použít v dalším návrhu, parametry byly cena, spolehlivost, hardwarové nároky na obsluhu, a spotřeba energie. Prototypy v této fázi byly stavěny převážně na nepájivém poli. Při testování metody detekce pomocí RFID nebyl problém načíst data při normálním zapojení modulu, data jsem byl schopen získat na vzdálenost několika centimetrů a každý tag se načetl pouze jednou při změně jeho pozice – což bylo ideální chování pro použití v této desce. Problémová situace nastala až ve chvíli, kdy se mezi čtečku a anténu připojil multiplexor (v tu chvíli jsem měl přístup pouze k multiplexoru CD74HC4051). Tento problém se projevoval tím, že čtečka neustále načítala data z tagu (při pozdějším hledání jsem zjistil, že se přepnula do módu kontinuálního čtení dat) a jeden tag se detekoval na několika herních pozicích naráz. Nepomohlo ani nastavení větších rozestupů mezi načítáním na různých pozicích, ani až nadbytečné čištění zásobníku s daty. Hlavní problémem však byla vzdálenost čtení, která se snížila na několik milimetrů.

Proto jsem přešel k použití jazýčkových relé. Tady problém při multiplexování nenastal, a proto jsem se zaměřil na jednotlivé faktory, které ovlivňují čtecí vzdálenost. Hlavním faktorem byla pochopitelně síla použitého magnetu, ale i jeho natočení vůči relé. Nejlepší vlastnosti čtení byly při kombinaci vodorovně umístěného relé a vertikálně postaveného magnetu. V tu chvíli bylo náhodné čtení na okolních polích minimální a při použití menší vzdálenosti od relé a menšího magnetu problém zcela zmizel.

### Testování funkčnosti jednotlivých modulů

V této fázi se testovalo, jestli jednotlivé moduly pracují korektně. Zejména jestli jednotlivá relé správně detekují délky. V této fázi jsem několikrát upravoval tištěné díly a řada z nich prošla výraznými úpravami (obrázek 7.1). V této fázi také vznikla finální podoba šachových



Obrázek 7.1: Vývoj předního panelu hodin

figur. Pro optimální čtení pozice je v nich magnet umístěn horizontálně vůči herní desce. Toto řešení přineslo malý problém se čtecím úhlem, proto všechny figury mají na sobě značku, která musí směřovat směrem k danému hráči (obrázek 7.2). Také se jednotlivé figury mohou mírně přitahovat, proto byly vyplňeny tavnou pistolí, aby se jim zvětšila váha (i přes toto řešení nebyl problém zcela eliminován a občas dochází k samovolnému posouvání nebo natáčení figur).

### Celkové testování všech modulů

Při testování celkové funkčnosti byl nejčastější problém ve správném propojení modulů, protože některé propojovací vodiče nebyly správě připojeny k jedné z desek. Toto byl velký problém u modulu hodin, kde LCD display při ztrátě spojení přestal komunikovat s Arduinem a modul se musel restartovat.

Další problém byl v odběru proudu LED diod. Jelikož jedna dioda může odebírat až 60mA, tak jen samotné LED diody mají odběr až 3,8A. Reálný odběr je samozřejmě menší, protože všechny LED diody většinou naráz nesvítí. Tento problém se částečně eliminoval maximálním jasem LED diod v kódu. Celkový odběr ostatních komponent se pohybuje kolem 300 mA a s předpokladem maximálního svitu diod se dostaneme tedy na spotřebu až 4,1A. Při použití bateriového napájení (dvě baterie zapojené paralelně, každá s kapacitou 2600mAh pro výstup 3,7V - přibližně 10Wh) by měla šachovnice vydržet zapnutá asi 30 minut. Reálně tato hodnota bude vyšší. Při testování jsem jí napájal z powerbanky (10Ah pro výstup 3,7V), podle předchozího výpočtu by šachovnici měla vydržet napájet necelé dvě hodiny, ale po několika dnech testování nekleslo nabití pod 75%. Z toho usuzuji, že na bateriích by šachovnice měla pár hodin vydržet.



Obrázek 7.2: **Ukázka šachových figur**

## Kapitola 8

# Možná vylepšení

K hlavním vylepšením, co se hardwaru týče, patří zprovoznění detekce figur pomocí RFID. Toho by šlo docílit buď použitím většího počtu RFID čteček, nebo zprovozněním jejich multiplexování. Použití šedesáti čtyř čteček je ale finančně a prostorově náročné. Pravděpodobněji tak může vylepšení nastat v multiplexování. Jednou z variant je použití jiných typů analogových multiplexorů, protože prototyp byl testován pouze pomocí CD74HC4051. Multiplexor, na kterém stavím současnou šachovnici, jsem našel v DIP variantě až později, proto nebyl v tomto použití testován. Také jsem zvažoval připojení multiplexoru na obě větve RFID antény, ale to by výsledná šachovnice (s tehdy používaným osmi kanálovým multiplexorem) potřebovala šestnáct těchto součástek. Což by bylo opět prostorově velmi rozměrné.

Pokud bych zůstal u varianty s jazýčkovými relé, tak bych přešel k variantě, která nemá skleněné pouzdro. Manipulace se skleněnými relé musí být velmi opatrná a než jsem přišel na způsob, jak ohýbat kontakty a umisťovat relé do desky na pájení, relativně dost mi jich popraskalo, či se jinak znehodnotilo. Poslední změnu v designu bych udělal v použití jiných konektoru pro propojovací vodiče, s menší velikostí a lepším ukotvením kabelu v konektoru (příkladem uvedu flex kabel s FFC/FPC konektorem).

Samotné plošné spoje by také mělo být možné zmenšit (s výjimkou multiplexovací desky, která je ovlivněna velikostí šachovnice) a některé komponenty by šly nahradit za jejich SMD varianty. Ovšem tyto změny by možná nešlo vytvořit doma a bylo by nutné nechat si tyto desky vyrobit na zakázku.

Samotná signalizace by také mohla dostat vylepšení, a to v podobě například automatického pohybu délku, jak bylo vidět v kapitole 2.

Mezi možnými softwarovými vylepšeními bych uvedl připojení šachovnice k internetu pomocí Wi-Fi v hlavní řídící jednotce, přidání různých herních variant (Fischerovy šachy<sup>1</sup> apod.) a přidání umělé inteligence, případně vytvoření rozhraní pro připojení externího šachového motoru.

Samotná AI by měla být implementovatelná relativně snadno (v závislosti na požadované síle), protože Flash paměť je dostatečná na použití knihovny zahájení, případně ukončení partie, pokud by byly rozumně velké. Výkon je také dostačující k tomu, aby bylo možné použít Alfa-beta prořezávání s dostatečnou hloubkou v případné kombinaci s řešením problematiky efektu horizontu. Samotná reprezentace šachových figur by měla jít implementovat pomocí bitboardu (i přes to že ESP8266 je 32bitový kontrolér).

---

<sup>1</sup>Varianta šachů s rozdílným počátečním rozestavěním figur

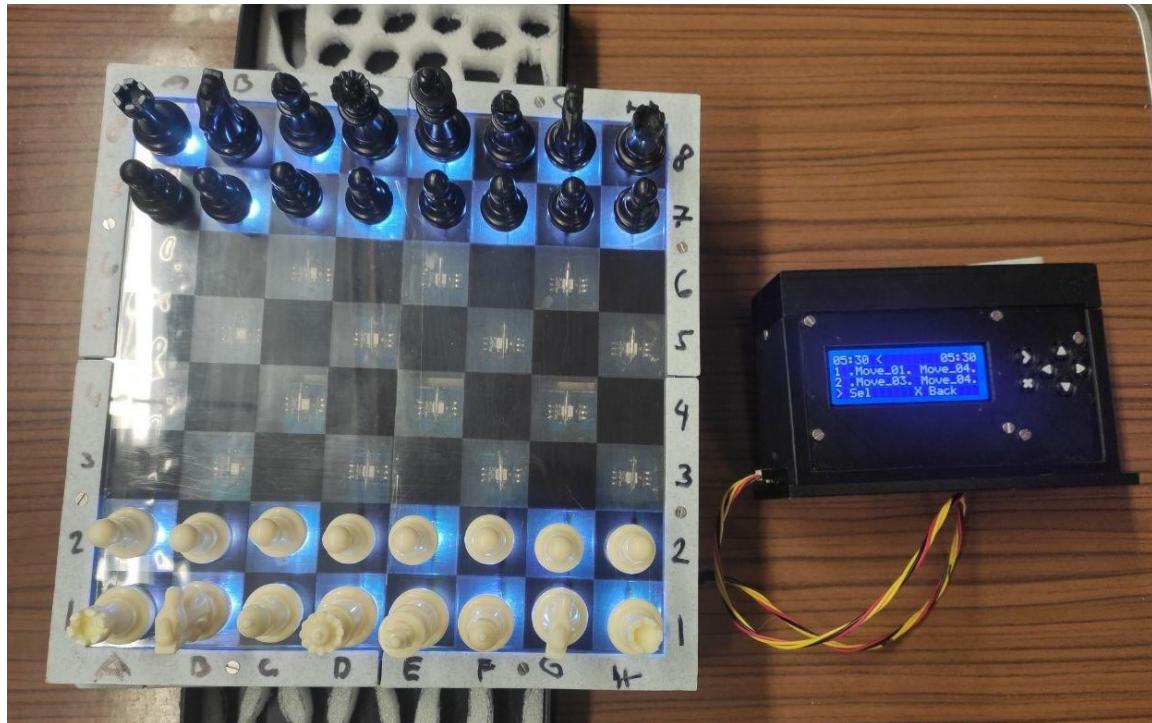
# Kapitola 9

## Závěr

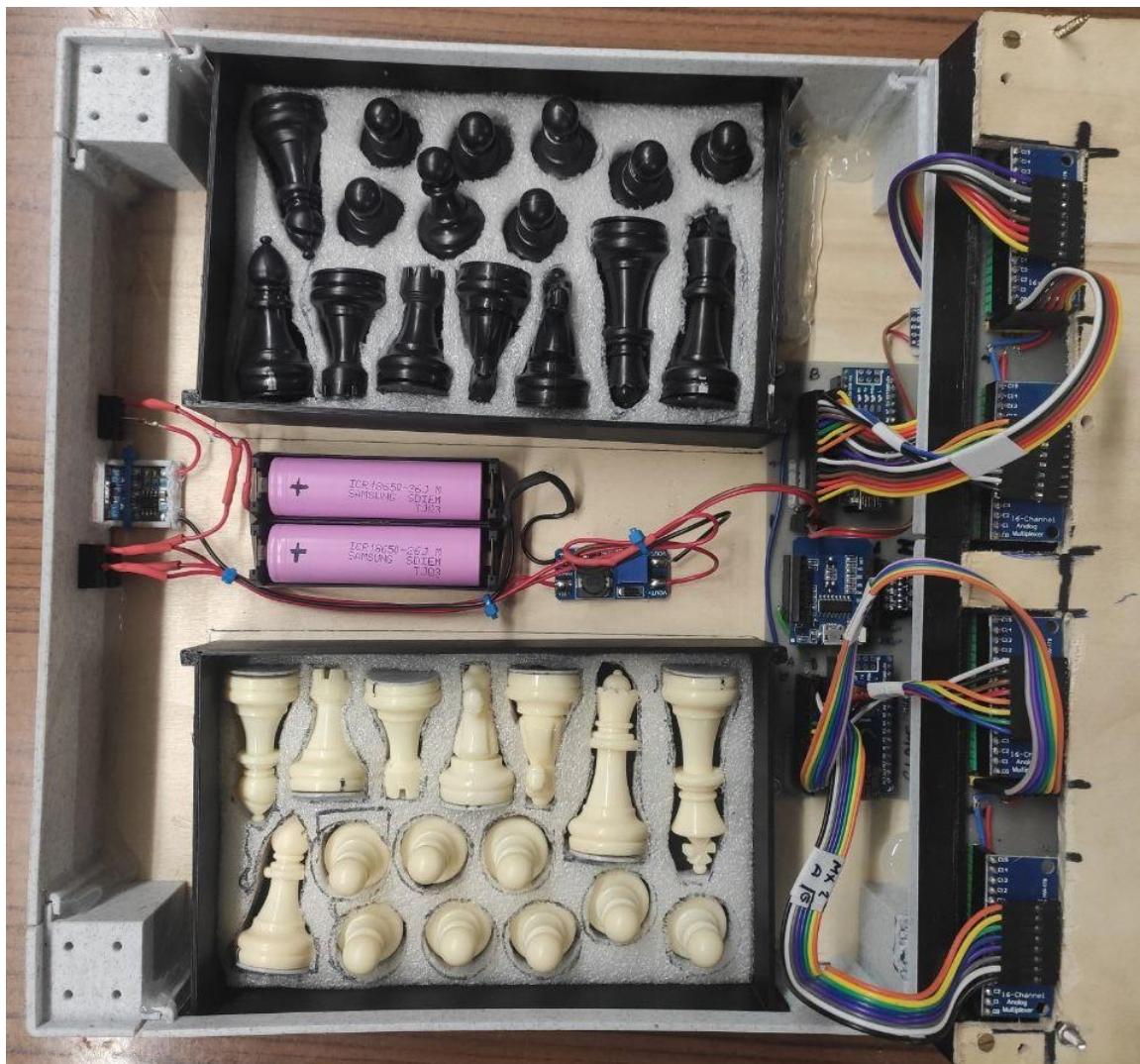
Cílem této práce bylo porovnat existující elektronické šachovnice a na základě těchto informací navrhnut a vyrobit vlastní šachovnici s hodinami. Šachovnice je na obrázcích 9.1 a 9.2.

Danou šachovnici se mi i přes drobné problémy v průběhu výroby povedlo úspěšně sestavit dle požadavků v zadání. Pro samotnou detekci dílků byla vybrána metoda čtení figur pomocí jazýčkových relé, která v testech vykazoval nejmenší omezení při samotné hře. Výsledné použití této metody je omezující pouze v natočení šachových dílků, aby bylo docíleno spolehlivé detekce.

Je jasné že s touto šachovou deskou nemohu konkurovat firmám, které je vyrábějí na profesionální úrovni, jak na poli elektroniky, tak i v oblasti softwaru. Výhodu bych našel pravděpodobně v nižší ceně a velké modifikovatelnosti, která u komerčních šachovnic není většinou možná.



Obrázek 9.1: Výsledná podoba šachovnice



Obrázek 9.2: Vnitřní rozmístění komponent

# Literatura

- [1] *8-bit Atmel Microcontroller with 4/8/16K Bytes In-System Programmable Flash.* ATmel Corporation, květen 2011. Rev. 2545T-AVR-05/11.
- [2] *CDx4HC405x, CDx4HCT405x High-Speed CMOS Logic Analog Multiplexers and Demultiplexers.* Texas Instruments, listopad 1997. SCHS122M Rev.M.
- [3] *High-Speed CMOS Logic 16-Channel Analog Multiplexer/Demultiplexer.* Texas Instruments, červenec 1998. SCHS209C.
- [4] WIKIPEDIE, P. *Šachová notace* [online]. Wikipedie: Otevřená encyklopedie., 2019. 2019-06-17 [cit. 2020-05-20]. Dostupné z: [https://cs.wikipedia.org/w/index.php?title=%C5%A0achov%C3%A1\\_notace&oldid=17376790](https://cs.wikipedia.org/w/index.php?title=%C5%A0achov%C3%A1_notace&oldid=17376790).
- [5] WIKIPEDIE, P. *Pravidla šachů* [online]. Wikipedie: Otevřená encyklopedie., 2020. 2020-04-21 [cit. 2020-05-19]. Dostupné z: [https://cs.wikipedia.org/w/index.php?title=Pravidla\\_%C5%A1ach%C5%AF&oldid=18419083](https://cs.wikipedia.org/w/index.php?title=Pravidla_%C5%A1ach%C5%AF&oldid=18419083).
- [6] WIKIPEDIE, P. *Šachy* [online]. Wikipedie: Otevřená encyklopedie., 2020. 2020-05-17 [cit. 2020-05-21]. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=%C5%A0achy&oldid=18521461>.
- [7] *Intelligent control LEDintegrated light source.* Worldsemi Co.