

**Univerzita Pardubice**  
**Fakulta Elektrotechniky a informatiky**  
**Katedra řízení procesů**  
**Obor Automatizace**

# **Robo vysavač**

Bakalářská práce

Vypracoval: Bordovský Petr

Vedoucí: Ing Rozsival Pavel

Pardubice 2024

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2023/2024

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Petr Bordovský**  
Osobní číslo: **I21035**  
Studijní program: **B0714A150008 Automatizace**  
Téma práce: **Robo vysavač**  
Zadávací katedra: **Katedra řízení procesů**

## Zásady pro vypracování

Cílem práce je návrh platformy robotického vysavače, schopného autonomně projíždět prostor. Teoretická část práce popíše možnosti pohonů a konstrukce typů podvozku použitelných u robotických vysavačů s ohledem na složitost implementace řízení pohonů a manévrovatelnost zvolené konstrukce. Součástí teorie bude i rozbor algoritmů pohybu robota pro optimální pokrytí plochy s překážkami. Praktická část provede návrh, konstrukci a ověření modelu robotického vysavače, schopného samostatného pohybu v prostoru. Robot bude implementovat vybraný algoritmus pohybu. Součástí práce bude ověření funkčnosti vybraného algoritmu, případně srovnání více.

Rozsah pracovní zprávy: **30-50**  
Rozsah grafických prací:  
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

- [1] VÁŇA, V. Mikrokontroléry ATMEL AVR: popis procesoru a instrukční soubor. Praha: BEN technická literatura, 2003.336 s. ISBN 978-80-7300-083-0.  
[2] VÁŇA, V. Mikrokontroléry ATMEL AVR: programování v jazyce C. Praha: BEN technická literatura, 2003. 216 s. ISBN 978-80-7300-102-0.  
[3] VLACH, J. Řízení a vizualizace technologických procesů. Praha: BEN technická literatura, 2002. 160 s. ISBN 978-80-86056-66-X.  
[4] BRTNÍK, B. Základní elektronické obvody. Praha: BEN technická literatura, 2011. 156s. ISBN 978-80-7300-408-8  
[5] RIPKA, P.; TIPEK, A. Master Book of Sensors. Praha : BEN, 2003. ISBN 0-12-752184

Vedoucí bakalářské práce: **Ing. Pavel Rozsival**  
Katedra elektrotechniky

Datum zadání bakalářské práce: **15. prosince 2023**  
Termín odevzdání bakalářské práce: **10. května 2024**

**Ing. Zdeněk Němec, Ph.D.** v.r.  
děkan

L.S.

**Ing. Daniel Honc, Ph.D.** v.r.  
vedoucí katedry

V Pardubicích dne 12. ledna 2024

# Prohlášení

Prohlašuji:

Práci s názvem **Robo vysavač** jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 21.2. 2024

Jméno a příjmení autora v.r.

## **Poděkování**

Rád bych vyjádřil svou upřímnou vděčnost Ing. Pavlu Rozsivalovi za jeho vedení mé bakalářské práce. Jeho cenné rady a odborné vedení byly neocenitelnými příspěvky nejen při vypracování bakalářské práce, ale i během celých tří let strávených na půdě pardubické univerzity. Dále bych chtěl poděkovat Martinu Plíškovi za jeho neocenitelnou pomoc s 3D tiskem a za zapůjčení všech souvisejících komponentů. Na závěr bych rád poděkoval mé přítelkyni Anně Šmídové za její neúnavnou podporu a pomoc s korekturami.

## **Anotace**

Cílem této práce je navrhnout platformu pro robotický vysavač, která bude schopna autonomně navigovat a čistit prostor. Teoretická část práce se zaměřuje na popis možných pohonů a konstrukcí podvozků vhodných pro robotické vysavače, s důrazem na složitost implementace řízení pohonů a manévrovatelnost vybrané konstrukce. Součástí teoretické části je také analýza algoritmů pohybu robota pro optimální pokrytí plochy s překážkami.

## **Klíčová slova**

robotický vysavač, autonomní řízení, senzory a sensorické systémy, pohon a mechanismy pohybu, navigační algoritmy a mapování prostoru, bezpečnost a detekce překážek

## **Title**

Robotic vacuum cleaner

## **Annotation**

The aim of this work is to design a platform for a robotic vacuum cleaner capable of autonomously navigating and cleaning spaces. The theoretical part of the work focuses on describing possible drives and chassis constructions suitable for robotic vacuum cleaners, with an emphasis on the complexity of implementing drive control and maneuverability of the selected design. Also included in the theoretical part is an analysis of robot motion algorithms for optimal surface coverage in the presence of obstacles.

## **Keywords**

Robotic vacuum cleaner, sensors and sensory systems, propulsion and motion mechanisms, navigation algorithms and space mapping, safety and obstacle detectio.

# Obsah

Seznam použitých obrázků, grafů a tabulek .....	8
Seznam použitých zkratk.....	9
Úvod.....	10
1 Sestavení robotického vysavače a jeho mechanické části.....	11
1.1 3D tisk a modelovací program.....	11
1.2 Vzhled vysavače .....	15
1.2.1 Tvar vysavače.....	15
1.2.2 Typy podvozků.....	15
1.3 Propojení jednotlivých komponentů.....	16
2 Fyzické části robotického vysavače (Hardware).....	17
2.1 Řídící jednotka a pomocné ovládací systémy.....	17
2.1.1 Používané řídicí jednotky v robotickém vysavači.....	17
2.1.2 Arduino UNO.....	18
2.1.3 A4988 DVR8825 Adapter.....	21
2.1.4 DRV8825 driver pro krokové motory .....	22
2.2 Akční členy.....	23
2.2.1 Pohony.....	23
2.2.2 Krokový motor NEMA 17 17HS3401 0,28Nm .....	24
2.2.3 TT Motor s převodovkou .....	25
2.2.4 Sací motor .....	26
2.3 Senzory .....	27
2.4 Akumulátor a jeho nabíjení .....	29
3 Softwarová část robotického vysavače .....	30
3.1 Vývojový diagram programu.....	34
3.2 Program .....	37
4 Návrh desky plošných spojů .....	44
4.1 Použité komponenty na DPS .....	46
4.1.1 SRD-05VDC-SL-C .....	46
4.1.2 FINDER TYPE 44.52.....	46
4.1.3 Step down.....	46
4.1.4 Tlačítka.....	46
4.1.5 Rezistory 100 k $\Omega$ a 20 k $\Omega$ .....	47
5 Vysávání.....	48
Závěr .....	51
Literární zdroje.....	52
Přílohy k bakalářské práci.....	54

## Seznam použitých obrázků, grafů a tabulek

Obrázek 1: První robotický vysavač Trilobite+ .....	10
Obrázek 2: Jednotlivé komponenty vysavače .....	11
Obrázek 3: 3D výkres vysavače .....	12
Obrázek 4: 3D pohled spodní části vysavače .....	13
Obrázek 5: Zadní kolo vysavače s hřídelí a krokovým motorem .....	14
Obrázek 6: Kompletní schéma robotického vysavače .....	17
Obrázek 7: Arduino UNO .....	20
Obrázek 8: Zapojení Adapteru .....	21
Obrázek 9: DRV8825 driver .....	22
Obrázek 10: Servomotor .....	23
Obrázek 11: Bipolární krokový motor .....	24
Obrázek 12: Unipolární krokový motor .....	24
Obrázek 13: Princip IR senzoru .....	27
Obrázek 14: Senzor TMF8701 porovnání naměřené a skutečné vzdálenosti .....	28
Obrázek 15: Lokalizace .....	31
Obrázek 16: Aproximativní rozklad .....	32
Obrázek 17: Lichoběžníkový rozklad .....	33
Obrázek 18: Vývojový Diagram .....	34
Obrázek 19: Vývojový Diagram – loop() .....	35
Obrázek 20: Vývojový Diagram – Podprogramy .....	35
Obrázek 21: Vývojový Diagram – B .....	36
Obrázek 22: Vývojový Diagram – A .....	36
Obrázek 23: Schéma DPS .....	44
Obrázek 24: DPS .....	45
Obrázek 25: Zapojení rezistorů 100 kΩ a 20 kΩ .....	47
Obrázek 26: Místnost před vysáváním .....	48
Obrázek 27: Místnost po pěti minutách vysávání .....	49
Obrázek 28: Místnost po deseti minutách vysávání .....	49
Obrázek 29: Místnost po patnácti minutách vysávání .....	50
Graf 1: Závislost točivého momentu na otáčkách motoru značky NEMÁ .....	25
Graf 2: Proudová zatížitelnost .....	45
Tabulka 1: Specifikace Arduina UNO .....	19
Tabulka 2: Převody TT motoru s převodovkou .....	25

## Seznam použitých zkratk

SLAM (Simultaneous Localization and Mapping)

3D = trojdimenzionální

CAD (Computer Aided Design) = počítačem podporované projektování

CAM (Computer aided Manufacturing) = počítačem podporované obrábění

CAE (Computer aided Engineering) = počítačem podporované inženýrství

PLC (Programmable Logic Controller) = programovací logický automat

SBC (Single Board Computer) = jednodeskový počítač

DPS = Deska Plošných Spojů

CPU (Central Processing Unit) = centrální procesorová jednotka

GPU (Graphics Processing Unit) = grafická procesorová jednotka

RAM (Random Access Memory) = paměť s náhodným přístupem

ALU (Arithmetic Logic Unit) = aritmeticko-logická jednotka

I/O (Input/Output) = vstupně/výstupní.

USB (Universal Serial Bus) = univerzální sériová sběrnice

Wi-Fi (Wireless Fidelity)

HDMI (High-Definition Multi-media Interface)

UART (Universal Asynchronous Receiver-Transmitter) = univerzální asynchronní přijímač-vysílač

ADC (Analog to Digital Converter) = A/D převodník

TDC (Time to Digital Converter) = převodník času na digitální signál

PWM (Pulse Width Modulation) = pulsně šířková modulace

NO (Normal Open) = kontakt je v klidovém stavu rozepnut

NC (Normal Close) = kontakt je v klidovém stavu sepnut

DIP (Dual In-line Package) = vícenásobný přepínač

LIDAR (Light Detection and Ranging) = detekce a měření světla

ToF (Time of Flight)

I2C (Inter-Integrated Circuit) multi-masterová sériová sběrnice

LED (Light-Emitting Diode) = elektroluminiscenční dioda

SPDT (Single Pole Double Throw) = jeden přepínací kontakt

DPDT (Double Pole Double Throw) = dva přepínací kontakty

## Úvod

Úvodem je potřeba zmínit něco málo o historii robotického vysavače s autonomním řízením a jeho postupným vývojem, kterým prošel za posledních dvacet let. Následně bude také poukázáno na nedostatky, které robotický vysavač má oproti klasickému.

## Historie

Prvním robotickým vysavačem s autonomním řízením byl vysavač značky Trilobite, jak je patrné na *Obrázku 1: První robotický vysavač Trilobite+* vyrobený švédskou společností Electrolux v roce 2001. Autonomním řízením se rozumí řízení, které probíhá bez přítomnosti lidského faktoru, a vysavač pracuje sám díky řídicí logice. V prvním autonomním robotickém vysavači značky Trilobite byl autonomní systém řešen pomocí ultrazvukových senzorů a předem předpřipraveného programu. Tyto senzory sloužily k mapování prostoru, zatímco předem definovaný program řešil flexibilitu vysavače při reakci na neočekávané situace.

S postupem času byly navigační schopnosti vysavače vylepšeny pomocí technologie SLAM. Tato technologie umožňuje vytváření map na základě zmapovaného prostředí a systematické vysávání na základě těchto map.

Posledním vývojem bylo propojení vysavače s tzv. „chytrými telefony“, „chytrou domácností“.



*Obrázek 1: První robotický vysavač Trilobite+*

## Výhody a nevýhody

Moderní robotické vysavače umožnily zjednodušit úklid v domácnostech i ve firmách, což umožnilo mít více času v dnešní uspěchané době a zároveň šetří náklady ve firmách, kde může robotický vysavač nahradit lidskou práci. Výhody robotického vysavače mohou být zřejmě i v místech, která jsou nebezpečná nebo obtížně přístupná pro lidi. Příkladem mohou být vysavače do bazénu.

S postupem vývoje už většina nevýhod, který měl robotický vysavač oproti klasickému, zmizela. Klasický vysavač je chápán jako vysavač, který není autonomně řízen a vyžaduje lidského faktoru. Nicméně některé nedostatky robotického vysavače stále přetrvávají. Příkladem může být sací síla ve srovnání s klasickým vysavačem v poměru ceny a výkon.

# 1 Sestavení robotického vysavače a jeho mechanické části

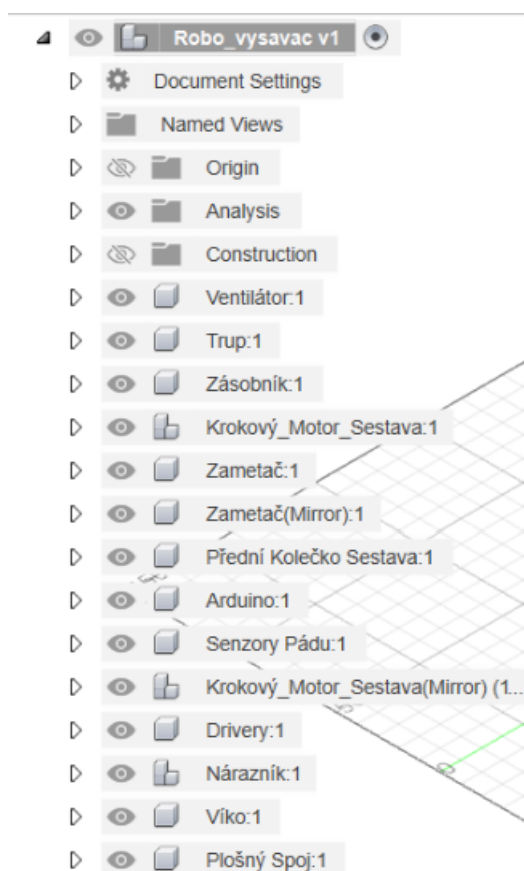
Tato kapitola je věnována sestavení robotického vysavače. Zabývá se procesem 3D tisku klíčových komponentů, výběrem vhodného tvaru vysavače a typem podvozku. Dále se zaměřuje na propojení jednotlivých komponentů, abychom vytvořili plně funkční zařízení.

## 1.1 3D tisk a modelovací program

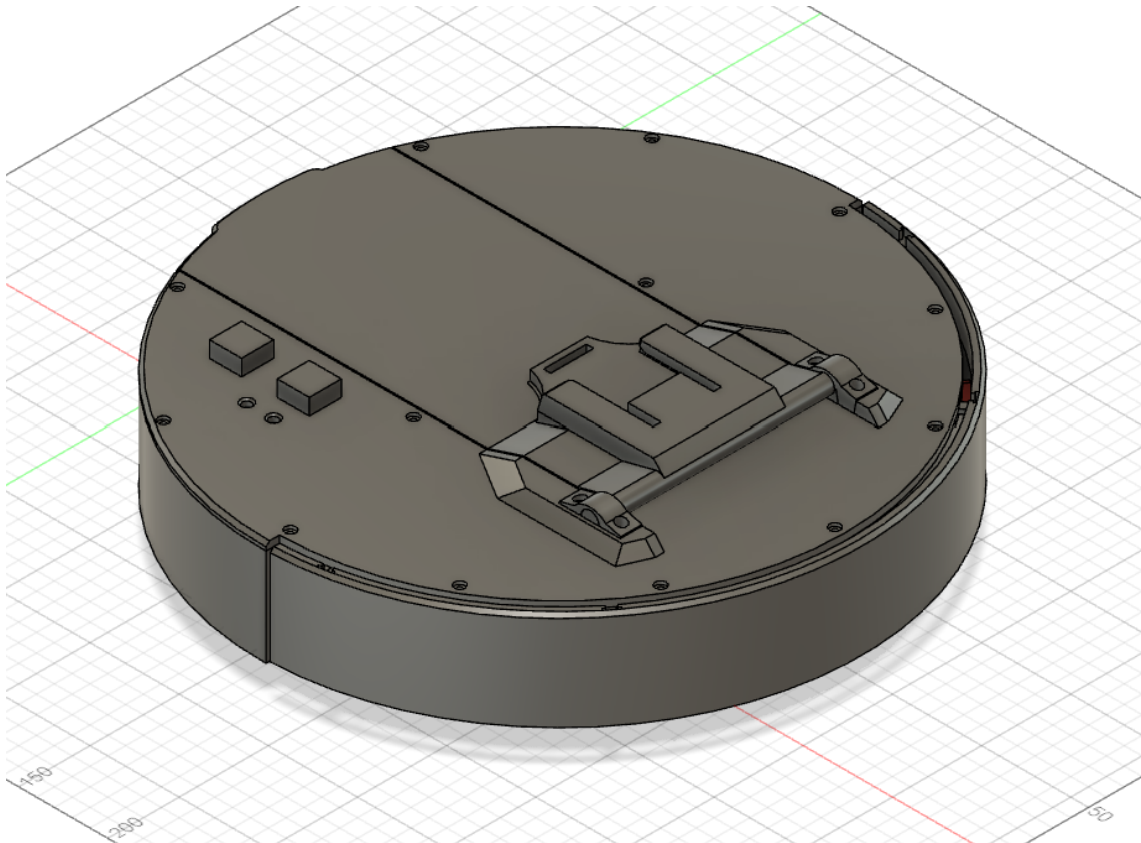
V praktické části byl model robotického vysavače vytvořen v modelovacím jazyce Fusion 360. Tento software nabízí širokou škálu nástrojů pro navrhování (CAD), jak pro strojírenské, tak pro 3D tisk, a také pro přípravu výrobních dat (CAM), a testování návrhů (CAE). Dále Fusion 360 umožňuje vytváření desek plošných spojů (PCB). V této práci byl využit nástroj pro navrhování (CAD) a generaci výrobních dat (CAM) .

Model byl vytvářen pomocí jednotlivých komponentů, jak je patrné na *Obrázku 2: Jednotlivé komponenty vysavače*. Každý komponent byl navržen s ohledem na jeden nebo více předchozích komponentech. Návrh probíhal od ventilátoru přes trup vysavače až po uložení desky plošných spojů. Do výkresu byly zahrnuty jednotlivé komponenty, které se budou do vysavače pouze vkládat, nikoliv tisknout na 3D tiskárně. Některé složitější části byly slepovány k sobě, protože nebylo možné je vytisknout jako jeden celek. Na *Obrázek 3: 3D výkres vysavače* již zobrazuje celý namodelovaný vysavač.

Jednotlivé komponenty jsou k dispozici v přílohách bakalářské práce.



Obrázek 2: Jednotlivé komponenty vysavače



Obrázek 3: 3D výkres vysavače

Tvarem robotického vysavače, jak je již patrné z *Obrázku: 3 3D výkres vysavače*, je kruh. Důvodem zvolení nejen tohoto tvaru, ale i samotného podvozku, bude zmíněno v podkapitolách *1.2.1 Tvar vysavače* a *1.2.2 Typy podvozků*. Na *Obrázku 4: 3D pohled spodní části vysavače*, si lze všimnout některých vybraných kót. První kótou je velikost 335 mm a jedná se o celkový průměr vysavače. Tento rozměr není nějak důležitý, ale pomůže nám přiblížit velikost celého modelu. Druhou kótou, již významnější, je velikost 275 mm. Tento rozměr udává vzdálenost mezi dvěma koly, které mají, na rozdíl od toho třetího, svůj hnací pohon. Z tohoto rozměru lze vypočítat vzdálenost od osy otáčení, viz vzorec (1).

$$l = \frac{\text{vzdálenost kol od sebe}}{2} [\text{mm}] \quad (1)$$

Tuto vzdálenost označíme  $l$  a po dosazení 275 mm do vzorce bude její hodnota 137,5 mm.

Na *Obrázku 5: Zadní kolo vysavače s hřídelí a korkovým motorem* je naznačen průměr zadního kolečka. Průměr  $d$  je tedy 6 mm a poloměr kolečka  $r$  3 mm. Při dosazení této vzdálenosti do vzorce (2), který je vzorcem pro obvod kruhu, dostáváme, že při otočení kolečka o  $360^\circ$  ( $2\pi$ ) mu odpovídá vzdálenost  $O$  velikosti 18,85 cm.

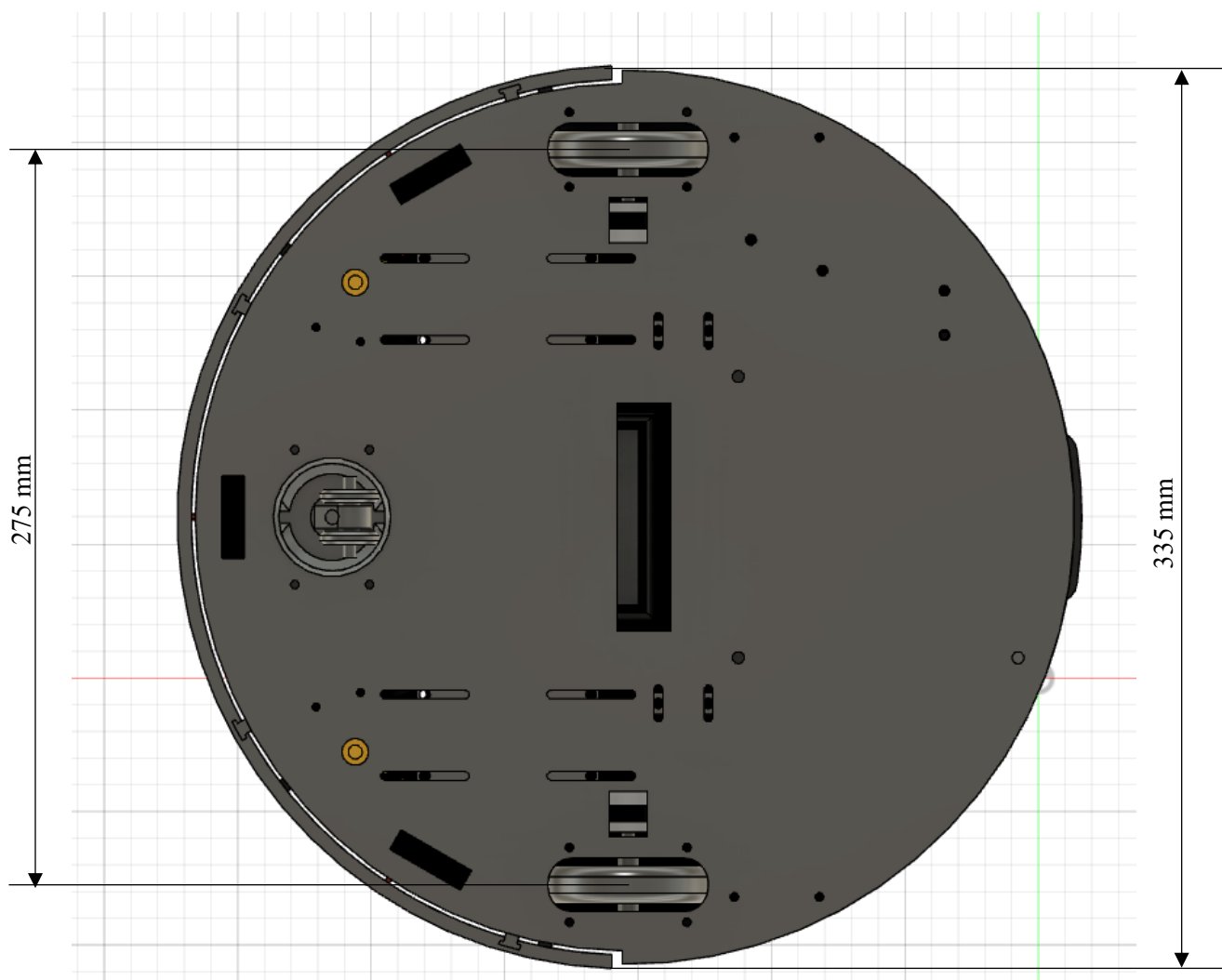
$$O = 2 * \pi * r [\text{cm}] \quad (2)$$

Na základě znalosti vzdálenosti od osy otáčení  $l$  a velikosti obvodu kruhu  $O$  můžeme podle vzorce (3) dopočítat úhel, o který se vysavač otočí.

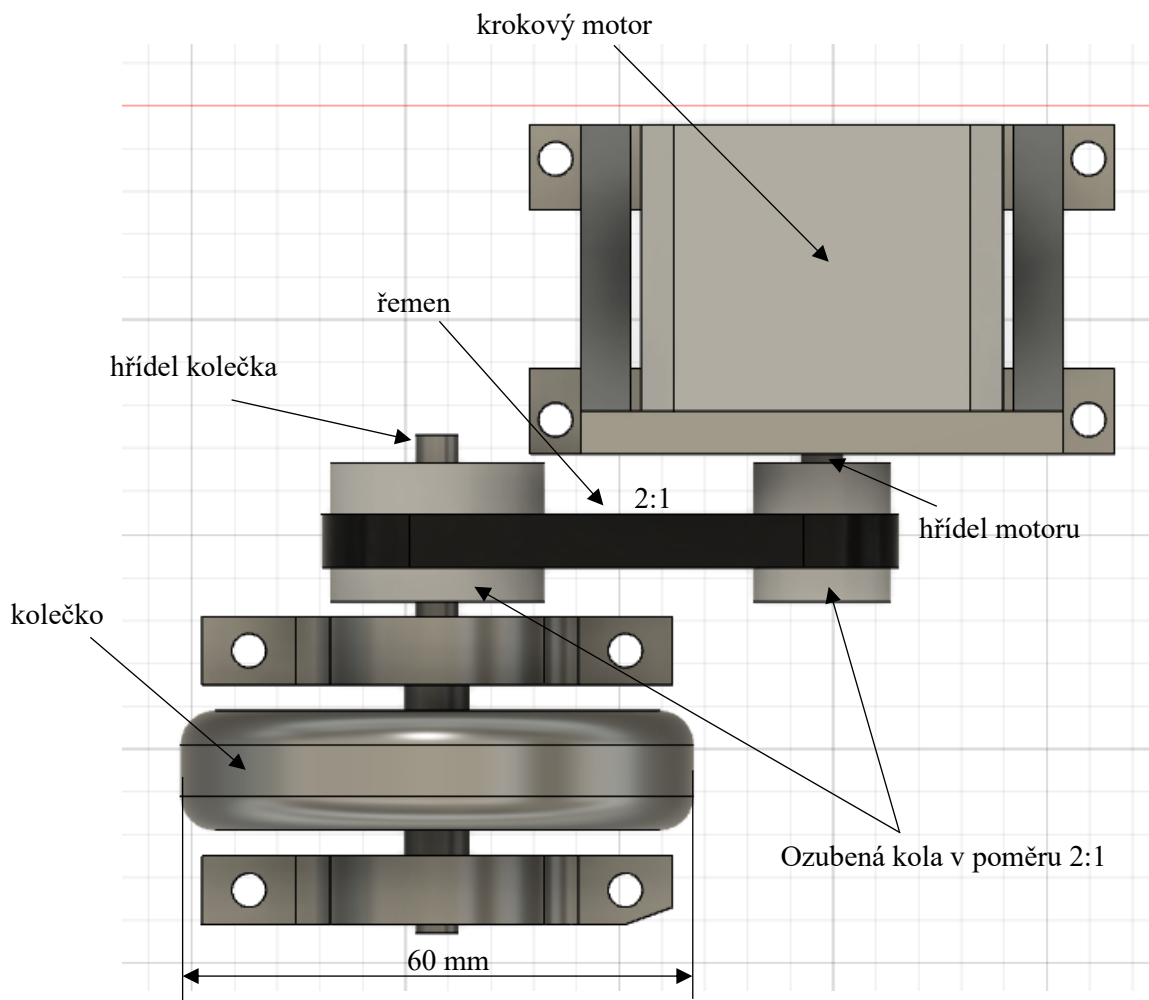
$$\alpha = \frac{O}{l} * \frac{180}{\pi} [^\circ] \quad (3)$$

Po dosazení do vzorce (3) dostáváme úhel  $\alpha = 68.76^\circ$ . O tento úhel se vysavač otočí, když se jednotlivá kolečka proti sobě otočí kolem své osy o  $360^\circ$ .

Na základě těchto velikostí se pak programuje velikost couvání a pootočení v podkapitole 3.2 *Program*. Velikosti jsou ale jiné, a to kvůli pootočení o úhel jiný než  $360^\circ$  a také kvůli jiným převodům koleček a krokových motorků zmíněných o odstavci níže.



Obrázek 4: 3D pohled spodní části vysavače



Obrázek 5: Zadní kolo vysavače s hřídelí a krokovým motorem

Na Obrázku 5: Zadní kolo vysavače s hřídelí a krokovým motorem si můžeme povšimnout, že poměr ozubených kol krokového motoru a kolečka se liší. Jedná se o poměr 2:1 a je realizován, aby se snížila rychlost otáčení motoru a kolečka. Rychlost otáčení kolečka bude korigována ještě v následujících kapitolách. Důvodem je, aby se zabránilo mechanickému poškození při nárazu vysavače do překážky, a také aby senzory, respektive cyklus programu, stíhaly dostatečně rychle reagovat.

## 1.2 Vzhled vysavače

Tvar vysavače je úzce spojený s typem podvozku a jeho použitím.

### 1.2.1 Tvar vysavače

#### **Kruhový**

Kruhový tvar vysavače je nejrozšířenějším tvarem. Jeho hlavní výhodou je, že kruhový tvar má prakticky nulovou šanci se zaseknout v rozích místností. U kruhového tvaru lze použít dotyková čidla pro detekci překážek při nárazu do nich, u ostatních tvarů by to bylo těžké, respektive je větší šance, že se vysavač nevymotá. Nevýhodou tohoto tvaru je, že se nedostane do rohů místností, kvůli tomu byl nahrazen půlkruhovým tvarem.

Tento tvar byl použitý v této bakalářské práci.

#### **Půlkruhový**

Půlkruhový tvar je novějším typem. Tento tvar pomalu vytlačuje kruhový tvar, a to hlavně kvůli možnému zajetí až do rohů místností. Tento tvar již nevyužívá dotykové senzory pro detekci překážek, ale používá bezdotykové senzory.

Oba tyto tvary se používají u robotických domácích i průmyslových vysavačů k vysávání interiérů.

#### **Čtvercový**

Čtvercový tvar je používán převážně u vysavačů na čištění bazénů.

### 1.2.2 Typy podvozků

Tato podkapitola, jak název naznačuje, je věnována typům podvozků v robotickém vysavači. Zaměřuje se převážně na **složitost řízení pohonů** a **manévrovatelnost**, a to z hlediska tvaru (typu) podvozku.

#### **Podvozek s pásy**

Tento typ podvozku se využívá u všech typů tvarů robotických vysavačů.

Řízení pásových podvozků obvykle probíhá pomocí dvou motorů umístěných na konci každého ze dvou pásů. Na podvozku se nachází turasové (ozubené) hnací kolo, ke kterému je připojen pohon podvozku. Dále je zde napínací mechanismus, který udržuje pásy napnuté, aby nedocházelo k jejich skluzu. Pohony těchto podvozků nejsou složité, používají se dva nezávislé řízené motory, jak je popsáno v podkapitole 2.2.1 *Pohony*. Otáčení vysavače je pak pomocí protichůdnému otáčení pásového podvozku, který probíhá smykem podvozku. Protože dotyková plocha mezi zemí a podvozkem je obdélníková, vysavač se otočí i na kluzkých, či nerovných podvozcích prakticky na místě, proto jeho manévrovatelnost je jedna z nejlepších z řady druhů podvozků.

## Podvozek se čtyřmi koly

Podvozek se čtyřmi koly se nejčastěji používá u čtvercových a půlkruhových tvarů.

Podvozek se čtyřmi koly u robotických vysavačů bývá bez natáčecí nápravy, proto manévrovatelnost tohoto vysavače bývá jen částečná a vzdálenost otočení oproti jiným podvozkům je mnohem delší.

Řízení se u těchto druhů podvozků provádí buď pomocí dvou pohonů, jeden pohon řídí levou stranu vysavače a druhý pravou, pak je zapotřebí pro otočení vysavače otočit smysl otáčení levé a pravé strany vysavače. Druhou možností je mít pohon na každé kolo zvlášť, a pak se robotický vysavač otáčí kombinací smyslu otáčení vždy protilehlých kol, proti ose otáčení.

Výběr pohonů může být libovolný, zmíněný dále v podkapitole 2.2.1 *Pohony*.

## Dvě pevná kolečka a jedno otočné o 360°

Tento nejrozšířenější typ podvozku se používá u robotických vysavačů s kruhovým tvarem, jako je i ten zvolený v této práci.

Ovládání pohonů je jednoduché, jako u předchozích příkladů. Je nutné mít pohon na každé kolo zvlášť, případně lze mít jeden pohon na zadní dvě kola a na otočné kolo mít jiný pohon, kterým by se řídila manévrovatelnost vysavače. V této práci byla zvolena první možnost, tedy zadní kola mají každý svůj pohon a přední kolo otočné o 360° je bez pohonu. Možnosti výběru pohonu jsou libovolné, a jak již bylo zmíněno, v předchozích dvou případech, lze ho najít v podkapitole 2.2.1 *Pohony*. V této práci byl zvolen pohon pomocí krokových motorů, o kterých bude zmíněno v dalších kapitolách. Otáčení vysavače je pomocí protichůdného smyslu otáčení levého a pravého zadního kola.

Manévrovatelnost tohoto typu podvozku je velmi dobrá, vysavač se otočí prakticky na místě. Oproti podvozkům pásovým, může dojít k nechtěnému smyku, ale při nízkých rychlostech je tato pravděpodobnost velmi nízká.

## 1.3 Propojení jednotlivých komponentů

Výpočet plochy průřezu vodiče je stanoven Normou ČSN 33 2130 ed.3 kvůli riziku ohřívání vodiče v důsledku průchodu elektrického proudu. Doporučené zatížení vodiče  $\sigma$  se pohybuje v rozmezí od 8 do 10 A/mm<sup>2</sup>. Tento rozsah je určen s ohledem na uložení vodiče, izolaci a počet vodičů. Pro zajištění bezpečnosti bylo zvoleno zatížení 8 A/mm<sup>2</sup>.

Minimální průřez vodiče byl vypočítán pomocí vzorce (4), kde maximální proud  $I$  je 2 A a zatížení vodiče  $\sigma$  je 8 A/mm<sup>2</sup>.

$$S = \frac{I}{\sigma} [\text{mm}^2] \quad (4)$$

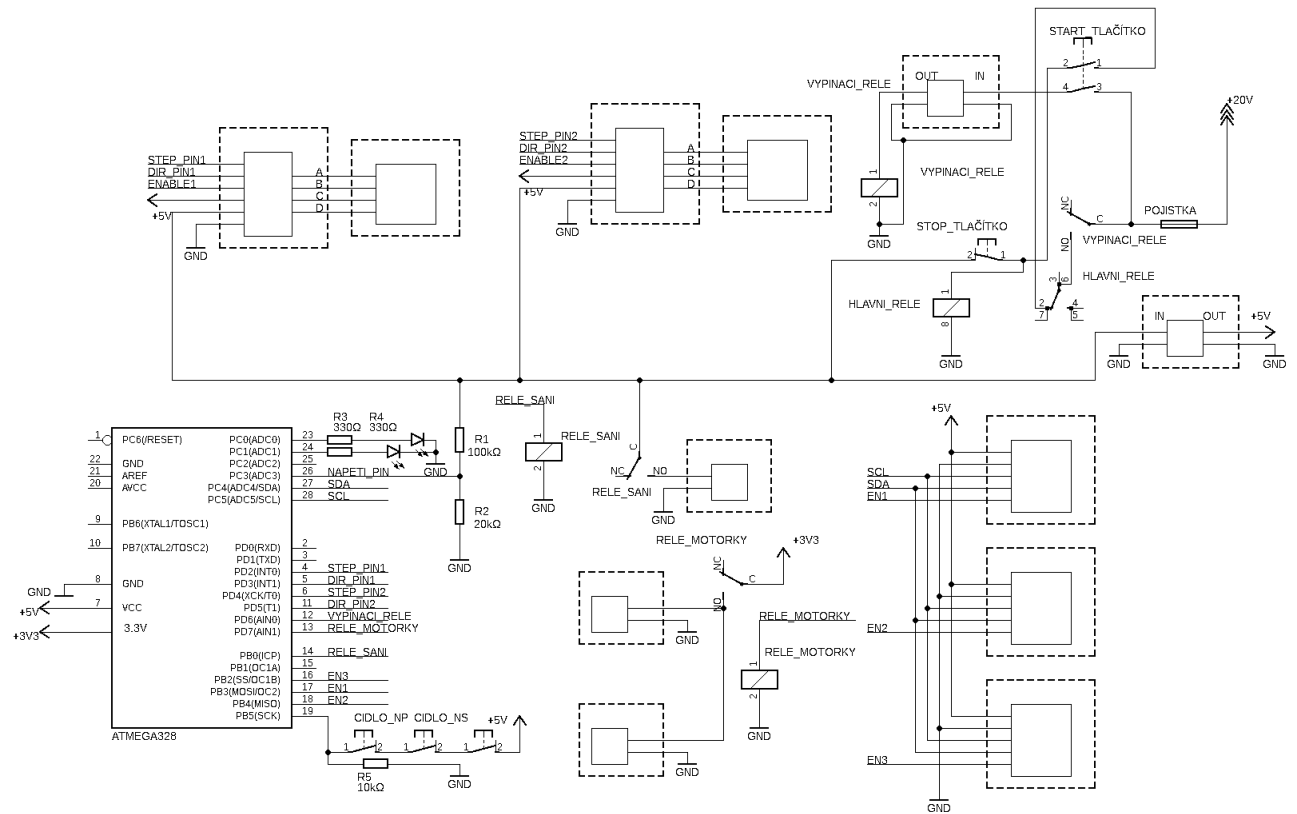
Po dosazení do vzorce získáme minimální použitelný průřez 0,25 mm<sup>2</sup>.

Pro použití byl zvolen průřez vodiče 0,5 mm<sup>2</sup>, což by mělo být více než dostatečné pro dané účely.

## 2 Fyzické části robotického vysavače (Hardware)

Tato kapitola bude věnována fyzickým částem robotického vysavače.

Na *Obrázku 6: Kompletní schéma robotického vysavače* je zobrazeno kompletní schéma robotického vysavače, které obsahuje všechny jeho důležité části a propojení mezi nimi.



Obrázek 6: Kompletní schéma robotického vysavače

### 2.1 Řídicí jednotka a pomocné ovládací systémy

Řídicí jednotka je základní prvek v odvětví řízení (Automatizace), který slouží k řízení celého systému. Příklady řídicích jednotek jsou jednodeskové počítače, mikrokontrolery, PLC a další.

#### 2.1.1 Používané řídicí jednotky v robotickém vysavači

V robotických vysavačích se jako řídicí jednotka běžně využívají jednodeskové počítače a mikrokontrolery.

Jednodeskové počítače (SBC) jsou počítače umístěny na jedné desce plošných spojů. Deska plošných spojů (DPS) je vyrobena převážně z laminátu a z jedné nebo více stran je nanosená měděná vrstva, na které se nachází elektronické součástky, které jsou mezi sebou propojeny vodiči, nebo vodivými cestami.

Hlavní komponentem jednodeskového počítače je mikroprocesor (CPU), který může být součástí celku v mikrokontroleru. Hlavní částí mikroprocesoru je aritmeticko-logická jednotka (ALU) sloužící k aritmeticko-logickým výpočtům. Dále se na jednodeskovém počítači nachází grafický procesor (GPU), který slouží ke grafickému zobrazování dat při připojení monitoru

k jednodeskovému počítači. Dalšími součástmi, které jednodeskový počítač má jsou Flash paměť neboli programová paměť a RAM paměť neboli datová paměť. V neposlední řadě se na desce nachází komunikační porty, jako je USB, Bluetooth, Wi-Fi, Ethernet, HDMI, SPI, I2C a další. Značky vysavačů, které požívají jako hlavní řídicí jednotku jednodeskový počítač jsou Neato Botvac D7 Connected, iRobot Roomba i7+ nebo Xiaomi Mi Robot Vacuum.

Druhou možností je použití mikrokontroleru. Mikrokontroler se používá spíše v levnějších typech robotických vysavačů. Jeho hlavní součástí je také mikroprocesor (CPU). Dále obsahuje také RAM paměť a Flash paměť. Oproti jednodeskovému zpravidla neobsahuje grafický procesor, protože se většinou používá v jednodušších zařízeních než jednodeskový počítač. Mikrokontroler obsahuje jako jednodeskový počítač I/O obvody a také komunikační porty, jako jsou USB, UART, I2C, SPI, Ethernet a další.

Mikrokontroler jako řídicí jednotku používá například vysavač značky iRobot Roomba 675.”

## 2.1.2 Arduino UNO

V sestaveném robotickém vysavači byla použita jako hlavní řídicí jednotka Arduino UNO. Jedná se tedy o jednodeskový počítač, jehož základem je mikrokontroler typu ATmega328 od firmy Atmel.

Řídicí jednotku od Arduina byla vybrána kvůli jednoduchosti a flexibilitě. Konkrétně vybraný typ byl pak Arduino UNO, protože vlastností a počet pinů této desky jsou dostačující. Všechny části řídicí logiky, které byly použity při sestavě robotického vysavače, jsou zde na jedné desce už připravené a naprogramované, proto odpadla práce s navrhnutím DPS a shánění komponentů, což by bylo jak těžší, tak dražší.

Arduino UNO má hodně částí, které se dají použít pro řešení různých úloh. Na Arduinu UNO byly použity některé z dále zmíněných částí.

Napájení Arduina bylo řešeno přivedením napětí ze step downu (zmíněn v podkapitole 4.1.3: *Step down*), na svorku Arduina 5V a GND.

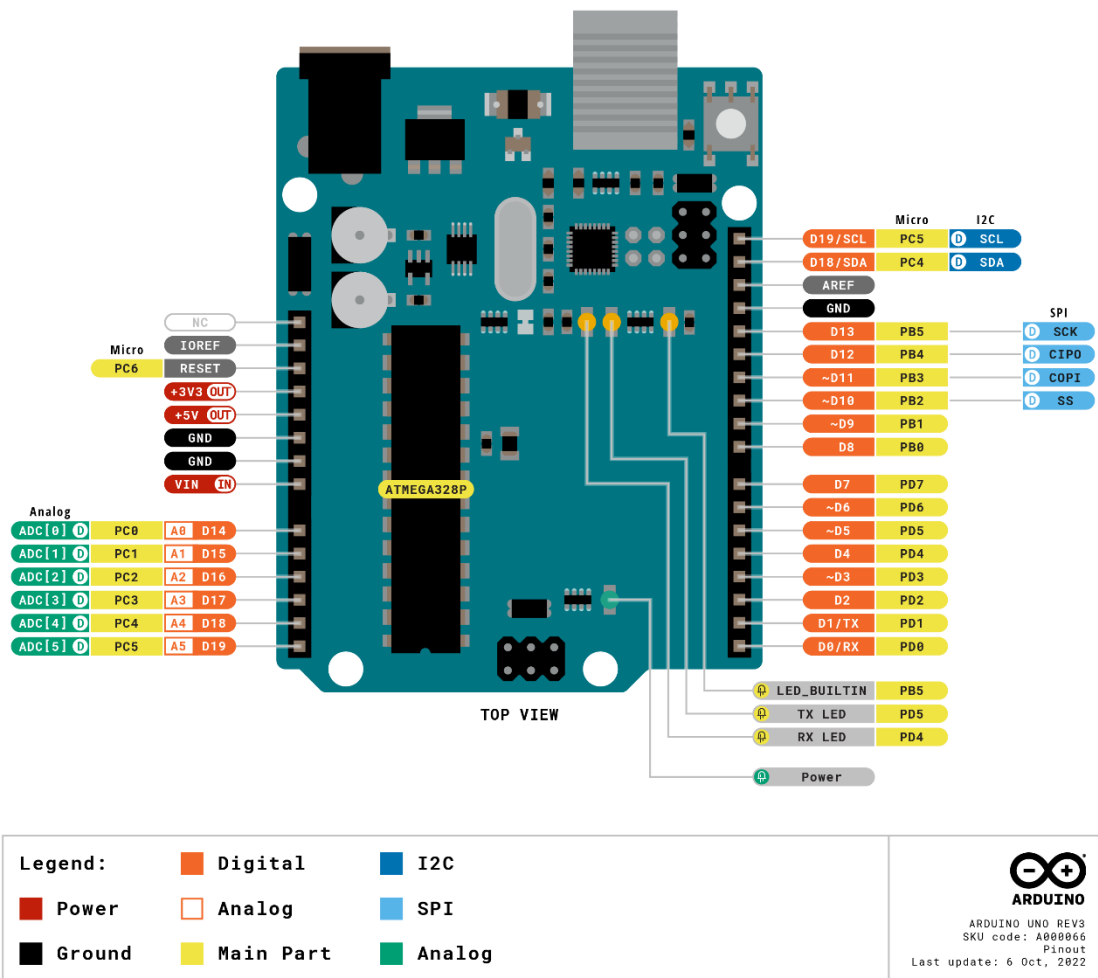
Dále byly využity analogové I/O. Jsou označeny jako „Ax“ viz *Obrázek 7: Arduino UNO* a budou dále zmíněny v podkapitole 3.2: *Program*. V této práci byly použity jak analogové vstupy, tak výstupy Arduina. V prvním případě byly z výstupu Arduina „A0“ a „A1“ připojeny LED diody. V druhém bylo na pin „A3“ připojen vstup z napěťového děliče, pro měření podpětí. U analogových vstupů se pak využívá ADC, který převede signál do digitální podoby, se kterou umí řídicí jednotka pracovat. V mikrokontroleru, který využívá Arduino UNO se nachází 10-bitový ADC, který odpovídá  $2^{10} = 1024$  hodnotám, které umí mikrokontroler rozlišit. ADC se dá programovat a nastavit na předem požadované použití. V této práci nebylo potřeba ADC programovat, byl pouze využit u analogových vstupů.

Analogové porty mají hojné využití v mnoha aplikacích. Analogové vstupy se nejčastěji využívají jako výstupy ze spojitých čidel a analogové výstupy k regulaci akčních členů k nimi připojených. Jako další byly využity digitální I/O, které jsou označeny jako číslice (0–13) viz *Obrázek 7: Arduino UNO*. Některé tyto I/O jsou s PWM. Tyto druhy výstupů byly použity ke změně smyslu otáčení krokových motorů. Jedná se o piny 3 a 5. Jako další piny, tentokrát už bez PWM, byly využity 2 a 4. Tyto piny slouží k pohybu krokových motorů. Dále jsou využity piny 6 až 8 a to k řízení relátek. Další tři piny 10 až 12 slouží k připojení enablů pinů. Poslední používaný pin je 13, na které jsou připojena dotyková čidla.

Další specifikace jsou uvedeny v tabulce *Tabulka 1: Specifikace Arduina UNO*.

Tabulka 1: Specifikace Arduino UNA

<b>Board</b>	<b>Name</b>	Arduino UNO R3
	<b>SKU</b>	A000066
<b>Microcontroller</b>	ATmega328P	
<b>USB connector</b>	USB-B	
<b>Pins</b>	<b>Built-in LED Pin</b>	13
	<b>Digital I/O Pins</b>	14
	<b>Analog input pins</b>	6
	<b>PWM pins</b>	6
<b>Communication</b>	<b>UART</b>	Yes
	<b>I2C</b>	Yes
	<b>SPI</b>	Yes
<b>Power</b>	<b>I/O Voltage</b>	5V
	<b>Input voltage (nominal)</b>	7-12V
	<b>DC Current per I/O Pin</b>	20 mA
	<b>Power Supply Connector</b>	Barrel Plug
<b>Clock speed</b>	<b>Main Processor</b>	ATmega328P 16 MHz
	<b>USB-Serial Processor</b>	ATmega16U2 16 MHz
<b>Memory</b>	<b>ATmega328P</b>	2KB SRAM, 32KB FLASH, 1KB EEPROM
<b>Dimensions</b>	<b>Weight</b>	25 g
	<b>Width</b>	53.4 mm
	<b>Length</b>	68.6 mm

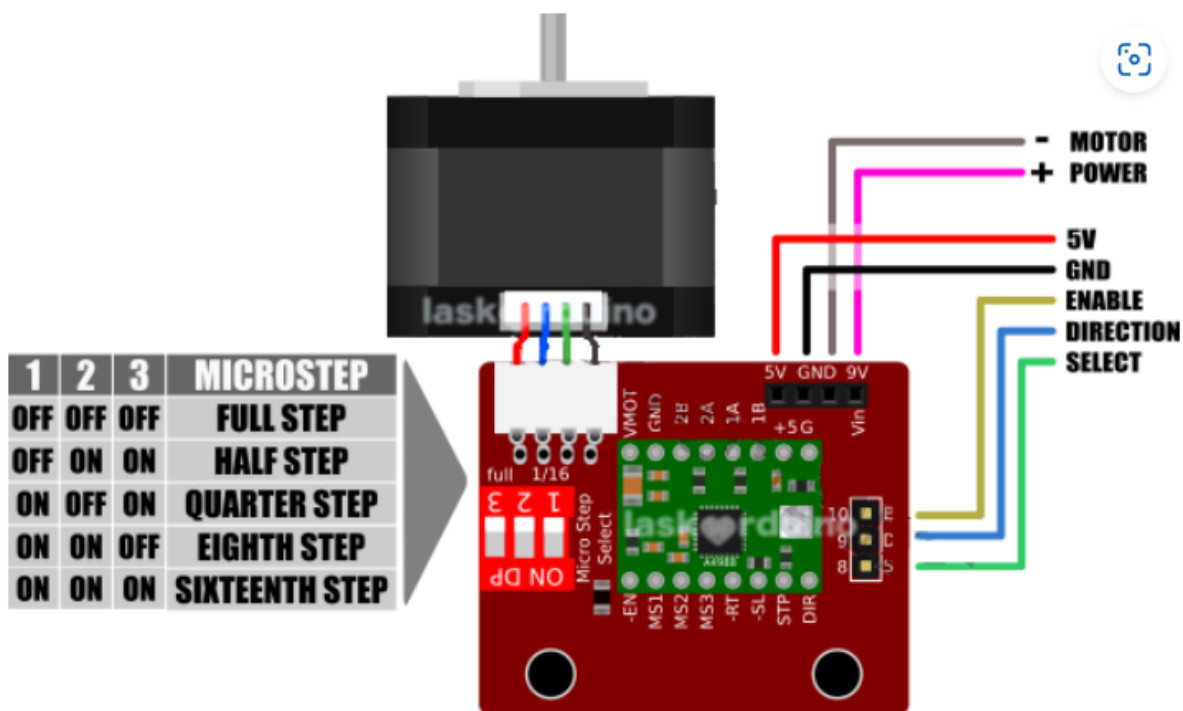


Obrázek 7: Arduino UNO

Arduino UNO má také 3.3 V výstup, který byl využit k napájení ToF senzorů 8701.

### 2.1.3 A4988 DRV8825 Adapter

Je to rozšiřující modul k Arduino. Využívá se na připojení driverů pro krokové motory. V této práci se jedná o DRV8825 driver zmíněný v podkapitole 2.1.4: *DRV8825 drivery pro krokové motory*. Lze na něj připojit i drivery s označením A4988.



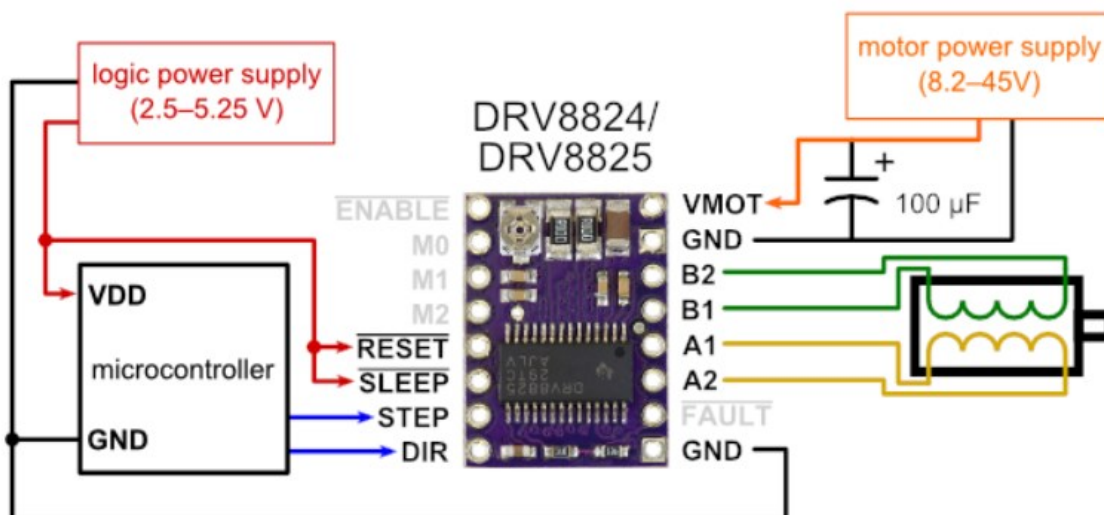
Obrázek 8: Zapojení Adapteru

A4988 DRV8825 Adapter má několik vstupů a výstupů. Na Adapter se dá připojit vnutí krokového motoru. Jedná se o konektor typu JST XH-4P. Tento konektor má čtyři piny a jeho hlavní výhodou je, že nejde zaměnit jejich pořadí. Dále se na adapteru nachází piny pro napájení. Jedná se o napájení logiky adaptéru a napájení motoru. Napájení logiky adaptéru odpovídá napětí 5 V, ve zhotovené práci je toto napětí přivedeno z výstupu step downu, dále zmíněno v podkapitole 4.1.3 *Step down*. Napájení krokových motorů odpovídá napětí 20 V. Toto napětí je přivedeno z Akumulátoru. Driver má tři piny na řídicí signály. Jedná se o pin „ENABLE“, „DIRECTION“ a „SELECT“. Tyto piny jsou připojené na výstupy Arduino UNA a slouží k ovládání korkových motorů, dále bude zmíněno v podkapitole 3.2: *Program*.

Na Adapteru se nachází DIP přepínač. Tímto přepínačem lze nastavit velikost kroku motoru od plného kroku až po šestnáctinu kroku. Tabulka nastavení mikrokroku je na *Obrázku 8: Zapojení Adapteru*. Ve zhotovené práci je poloha nastavena do polohy osminového kroku.

## 2.1.4 DRV8825 driver pro krokové motory

Drivery se používají pro řízení krokových motorů. V této práci se používají driver s označením DRV8825. Jedná se o vylepšenou verzi A4988 driveru. Tyto dva drivery jsou mezi sebou zaměnitelné. DRV8825 má několik pinů, od napájení logiky a krokového motoru, přes vynutí krokových motorů, až po piny řídicích signálů z Arduina. Tyto piny byly pouze zacvaknuté do patice na Adaptéru, který byl zmíněný v podkapitole 2.1.3: *A4988 DRV8825 Adapter*. Sled pinů a veškerých částí lze vidět na *Obrázku 9: DRV8825 driver*.



Obrázek 9: DRV8825 driver

Driver DRV8825 má několik nastavení. Hlavní nastavení je pomocí trimru. Lze si všimnout na *Obrázku 9: DRV8825 driver*. Tímto trimrem se nastavuje referenční napětí, z kterého se pak dále dopočítává velikost požadovaného maximálního proudu vynutím krokových motorů. Referenční napětí se měří pomocí voltmetru mezi svorkou GND a pouzdrem potenciometru. Vzorec pro vypočítání napětí, které máme nastavit, při námi zvoleném proudu je pak dán rovnicí (5).

$$U_{REF} = \frac{I_{MAX}}{2} [V] \quad (5)$$

V této práci je maximální možný proud tekoucí krokovým motorem 1,3 A. Při dosazení do vzorce nám vyjde velikost referenčního napětí 0,65 V. Není ale potřeba, aby krokovým motorem protékal maximální proud. Bylo tedy zvoleno, aby maximální proud procházející cívkou byl 0,6 A. Referenční napětí pak odpovídá 0,3 V.

Maximální proud, který může procházet driverem DRV8825 je 1,5 A a při aktivním chlazení až 2,2 A. V této práci je používáno aktivní chlazení, takže rezerva maximálního proudu je více než třínásobná.

Driver DVR8825 dále pak obsahuje tepelnou a proudovou ochranu, proto není třeba se obávat o jeho zničení.

## 2.2 Akční členy

Akční členy představují jednu z hlavních skupin v automatickém řízení nebo regulaci, protože nastavují akční veličinu. Tyto členy lze rozdělit do dvou hlavních podskupin: pohony a regulační orgány. V této kapitole budou především zmíněny pohony, neboť na úrovni mechaniky robotického vysavače nejsou používány regulační orgány a regulace je zajištěna pomocí řídicí jednotky.

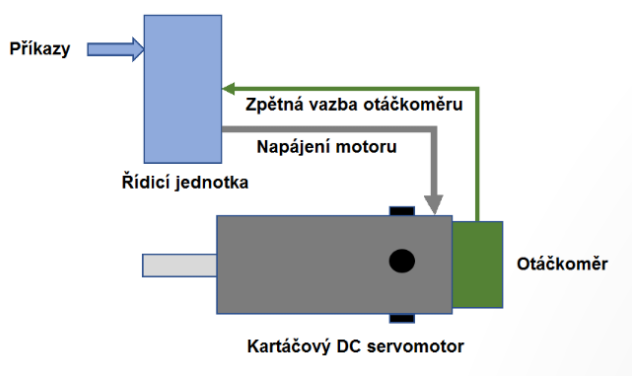
### 2.2.1 Pohony

Pohony lze rozdělit na pneumatické, hydraulické a elektrické. V této práci se nacházejí pouze elektrické pohony. Elektrické pohony lze dále rozdělit podle druhu energie na stejnosměrné a střídavé. Nejběžnějšími elektrickými pohony jsou motory. Ty se dále rozdělují na stejnosměrné a střídavé a následně synchronní, asynchronní a speciálně na krokové.

V robotickém vysavači se nejčastěji používají servomotory, krokové motory a stejnosměrné motory.

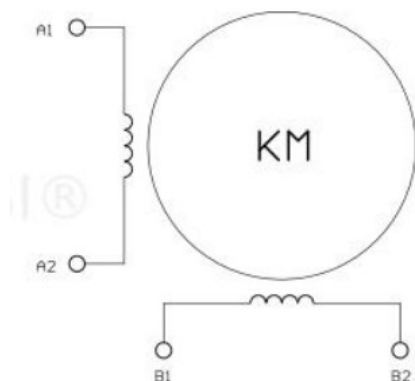
Servomotory jsou řízeny pomocí zpětnovazební smyčky, což umožňuje dosáhnout přesné polohy natočení. Poloha natočení je určena vestavěným senzorem polohy a zpětnou vazbou. Servomotory obvykle dosahují menších rychlostí otáčení než jiné typy motorů.

Na *Obrázku 10: Servomotor* je znázorněn servomotor s řídicími signály, které do něj vstupují. Dále je zobrazena zpětná vazba a měření polohy natočení.

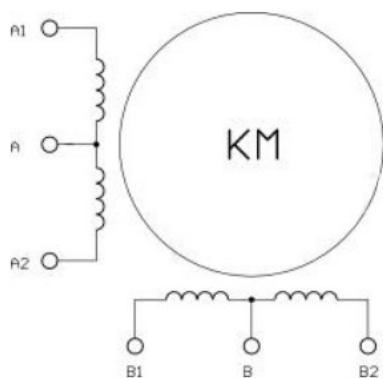


Obrázek 10: Servomotor

Při přivedení pulzu na vinutí se motor otočí o definovaný úhel. Tento úhle obvykle činí  $1,8^\circ$  nebo  $0,9^\circ$  a lze ho dále dělit pomocí driveru. Oproti servomotorům se krokový motor otočí pouze o předem definovaný úhel, což znamená, že servomotor je přesnější. Krokové motory lze rozdělit na bipolární viz *Obrázek 11: Bipolární krokový motor*, který má čtyři vývody a unipolární viz *Obrázek 12: Unipolární krokový motor*, který má šest vývodů.



Obrázek 11: Bipolární krokový motor



Obrázek 12: Unipolární krokový motor

Oba tyto druhy motorů se dají využít v robotickém vysavači, a to kvůli snadnému určení polohy hřídele.

V bakalářské práci byly tyto motory zvoleny kvůli jednodušší manipulaci oproti servomotorům. I když jsou méně přesné než servomotory, jsou pro toto konkrétní použití jsou dostačující.

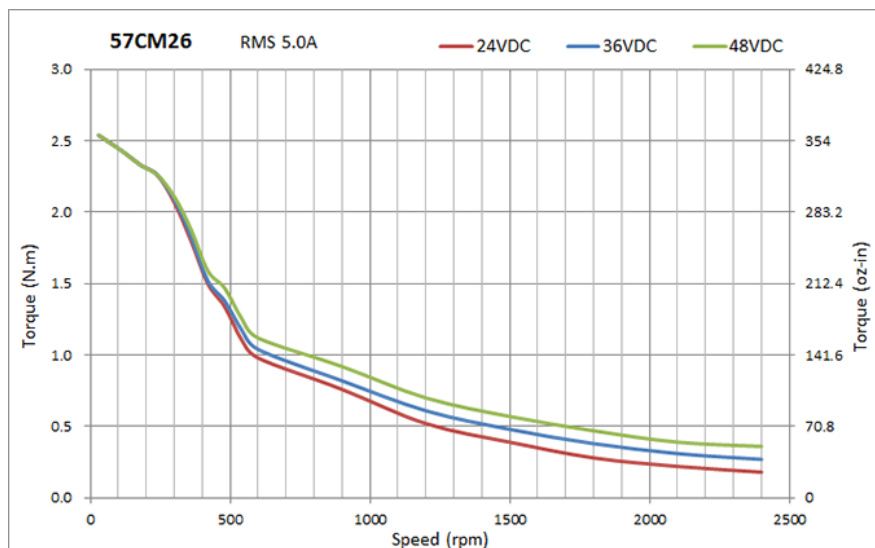
V robotickém vysavači se dále využívají jiné typy motorů, které nebyly zmíněny v předchozím textu. Jedná se však o pomocné motory, které jsou využívány například k pohonu vysávacích kartáčů nebo v mechanismu sání.

### 2.2.2 Krokový motor NEMA 17 17HS3401 0,28Nm

Jedná se o bipolární krokový motor s přídržným momentem 28 N.cm. Přídržný moment je moment, který je potřeba překonat k udržení motoru v klidu, jinými slovy je to síla, kterou motor dokáže udržet, než je násilnou silou pootočen. Je důležité, aby byl dostatečný, jinak by se mohla ztratit informace o poloze. Tento moment není lineární a závisí na rychlosti otáček a velikosti napětí. Se snižujícím napětím klesá moment, zatímco s klesajícími otáčkami roste. Dalším parametrem je maximální jmenovitý proud na vinutí. U tohoto typu motorů je maximální jmenovitý proud 1,3 A. Omezení na maximální velikost proudu je řešeno v podkapitole 2.1.4

*DRV8825 driver pro krokové motory.* Velmi důležitým parametrem je úhel jednoho kroku. U krokového motoru NEMA 17 17HS3401 je tento úhel  $1,8^\circ$  na jeden krok, což znamená, že je potřeba 200 kroků nebo pulzů pro otočení motoru kolem celé své osy.

V *Grafu 1: Závislost točivého momentu na otáčkách* zobrazuje grafickou reprezentaci tohoto závislého momentu na rychlosti otáčení. Tento graf je specifický pro krokové motory značky NEMA. Z grafu je patrné, že nejlepší moment je dosažen při otáčkách do 500 za minutu. Při otáčkách vyšších než 500 za minutu se moment snižuje na polovinu, což znamená, že při těchto vyšších rychlostech je motor méně účinný.



Graf 1: Závislost točivého momentu na otáčkách motoru značky NEMÁ

### 2.2.3 TT Motor s převodovkou

Jedná se o stejnosměrný motor s vestavěnou převodovkou. Ozubená kola v převodovce jsou vyrobena z plastu; lepší variantou by byla kovová, ale pro bakalářskou práci postačí plastová. Samotná převodovka je integrovaná a má poměr 48:1, což znamená, že se vnitřní hřídel otáčí 48krát rychleji, než je výstupní hřídel motorku. Rychlosti otáčení lze přepínat podle přiložené tabulky, viz *Tabulka 2: Převody TT motoru s převodovkou*. Z hodnoty napájecího napětí dostaneme rychlost otáčení za minutu. V bakalářské práci je TT motor napájen z Arduina, který poskytuje výstupní napětí 5 V. Při této hodnotě napětí dosahuje 197 otáček za minutu a odebírá proud o velikosti 100 mA. Zvolení napětí 5 V bylo z důvodu, že při 3,3 V byly otáčky příliš nízké kvůli tření kartáčů s podlahou.

Tabulka 2: Převody TT motoru s převodovkou

Napájení:	DC 3V	DC 5V	DC 6V
Proud:	100 mA	100 mA	120 mA
Převod:	48:1		
Výstupní otáčky:	114ot/min	197ot/min	240ot/min

Hlavní mechanismus je již zmíněný stejnosměrný motor.

Princip fungování stejnosměrného motoru spočívá v tom, že když proud protéká vodičem (vinutím), které je umístěno v magnetickém poli, začne samovolně rotovat. Rotor je tvořen vinutím, čímž se stává elektromagnetem. Magnetické pole je vytvořeno statorem, který obsahuje severní a jižní pól. Aby se rotor mohl otáčet nepřetržitě, musí se v průběhu otáčení měnit směr proudu a tím i magnetická polarita jednotlivých pólů. Tato změna je řízena komutátorem.

Síla, která pohání otáčení rotoru, je určena Lorenzovým vztahem. (6)

$$F = B * I * l * \sin\varphi [N] \quad (6)$$

Kde  $B$  odpovídá velikosti magnetické indukce ze statoru.  $I$  je velikost protékajícího proudu rotorovým vinutím a  $l$  je délka vodič kolmé na směr šíření indukčních čar, nebo zjednodušeněji na stator. Úhel  $\varphi$  je pak odchýlení od kolmosti.

Stejnosemřný motor má, jak už bylo zmíněno, tři hlavní části.

První je rotor, což je pohyblivá část motoru, tvořená elektromagnetem, komutátorem a může obsahovat i uhlíky.

Druhou částí je stator, což je nepohyblivá část motoru, která může být tvořena permanentním magnetem nebo elektromagnetem.

Poslední hlavní částí je komutátor, což je mechanický usměrňovač a měnič kmitočtu.

Stejnosemřné motory mohou mít různá provedení a zapojení, jako jsou sériové, derivační a kompaundní. Každé z těchto zapojení má specifické charakteristiky, chování a použití.

V bakalářské práci byly použity dva stejnosměrné motorky, jako pohon pro zametací systém, proto není nutné detailně rozebírat chování jednotlivých zapojení a příkladů.

#### 2.2.4 Sací motor

V robotických vysavačích se nejčastěji používají stejnosměrné sací motory. Princip tohoto motoru spočívá ve vytvoření podtlak, který slouží k nasávání nečistot. Mezi motorem a úložným prostorem pro nečistoty bývá obvykle umístěna přepážka, aby se nečistoty nedostávaly přímo dovnitř motoru. Motor je pak dobře izolován proti nečistotám.

V práci byl použit Dmyhadloví ventilátor XH2.54, který se obvykle používá k foukání vzduchu než k jeho nasávání. I přesto, že je tento ventilátor konstruován pro foukání vzduchu, jeho nasávací vlastnosti by neměly být zásadním problémem (v kapitole *Závěr* pak bude rozebráno). Dmyhadloví ventilátor je navržen pro napájecí napětí 20 V, což je v našem případě ideální. Jeho otáčky činí 3200 ot/min. Regulace otáček je možná pomocí změny napětí přivedeného na ventilátor, ale vzhledem k očekávanému výkonu sacího ventilátoru byla regulace otáček zanedbána.

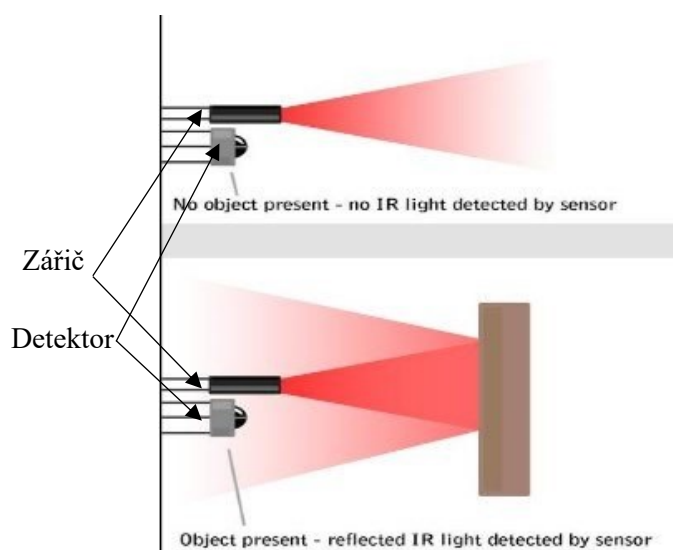
## 2.3 Senzory

Předchozí kapitola byla věnována akčním členům. Tato kapitola se bude věnovat měřícím členům, a to konkrétně sensorům a čidlům.

Úkolem měřících členů je získat měřená data pro ústřední člen. V této práci je ústředním členem Arduino UNO a měřícími členy jsou ToF senzory a koncové senzory. U robotického vysavače jsou senzory velmi důležitou částí, protože na základě jejich informací se získávají údaje o poloze a také zajišťují bezpečnost pohybu vysavače, aby vysavač nespadol ze schodů, nezasekl se a nenastali podobné situace.

V robotickém vysavačích se používá velká řada druhů sensorů a čidel. Některé principy budou zmíněny a na základě toho budou vybrány pro sestavení robotického vysavače.

**Infračervená čidla** jsou nespojitá i spojitá bezkontaktní čidla. Princip spočívá v emitaci infračerveného záření ze zářiče a vyhodnocení času odezvy do detektoru, princip je dále znázorněn na *Obrázku 13: Princip IR senzoru*. Infračervená čidla lze použít jako laserová a ultrazvuková čidla v následujících odstavcích, ale většinou se v robotických vysavačích používají k měření vzdálenosti vysavače od objektů, aby vysavač nemusel přímo nebořit do překážky a mohl se včas vyhnout. Výhodou těchto čidel je, že se na vzájemně neovlivňují, jako je to v případě ultrazvukových.



Obrázek 13: Princip IR senzoru

**Ultrazvuková čidla** lze označit jako levnější alternativou laserových čidel. Jedná se také o spojitě nebo nespojitě, bezkontaktní měření. Rozdílem je, že se nevysílá a nepřijímá paprsek, ale ultrazvuk. Na rozdíl od laserových čidel se mohou ultrazvuková čidla navzájem ovlivňovat, proto je třeba klást obezřetnost při jejich instalaci. Další nevýhodou je, že ultrazvuková čidla měří v menším rozsahu vzdálenosti než laserová, a navíc neměří od nulové vzdálenosti, což může být problém při jejich použití.

V bakalářské práci byly zvoleny senzory na následujících dvou principech.

## Laserové čidla

Laserová čidla pracují na principu LIDAR. Tato metoda může být použita ve více principech. Nejčastější princip používaný v robotickém vysavači je pomocí triangulace. Princip spočívá v tom, že čidlo vysílá světelné paprsky a na základě úhlu dopadu na detektor mapuje vzdálenost, neboli prostor.

Druhou možností je vysílání laserového paprsku a na základě délky odezvy se dopočítává vzdálenost. Tato metoda je využívána i u ultrazvukových čidel zmíněných v odstavci **Ultrazvuková čidla**. Tento princip je využíván ve zhotovené práci, a to u senzorů ToF značky DFRobot typu TMF8701.

## DFRobot TMF8701

Senzor TMF8701 pracuje na principu ToF, který byl zmíněn v předchozím odstavci.

Senzor dokáže měřit vzdálenost od 1mm až po 60 mm. Rozsah vzdáleností je pak vybírána mezi následujícími třemi režimy. První je „ePROXIMITY“, tento mód je určen k měření malých vzdáleností od 1 mm až po 10 mm. Druhý je „eDISTANCE“, tento režim je určený pro větší vzdálenosti od 10 mm do 60 mm. Posledním módem je „eCOMBINE“, který je kombinací předchozích dvou a měří od 1 mm až po 60 mm. Nevýhodou je větší chyba měření oproti předchozím dvěma. Tyto režimy se nastavují softwarově.

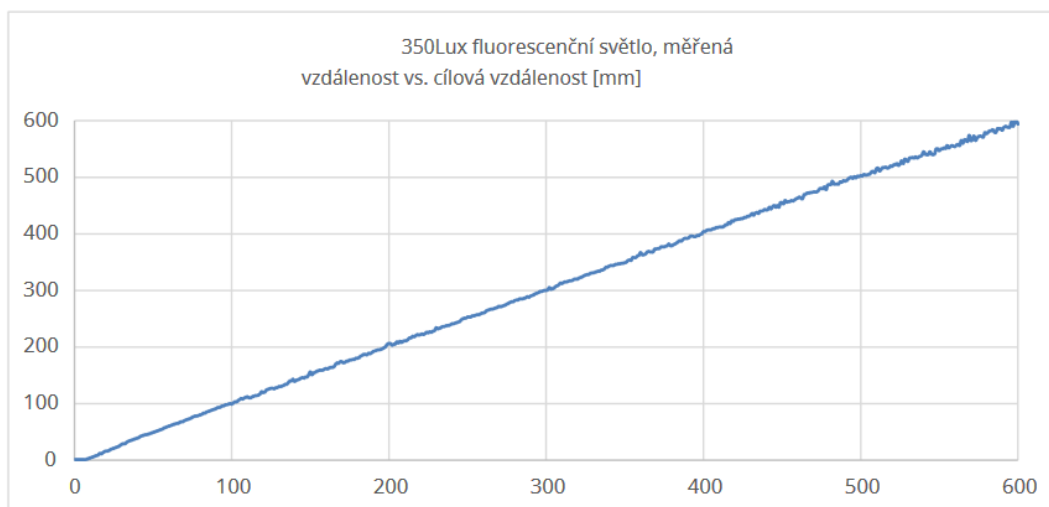
Připojení senzoru je pomocí I2C sběrnice. Adresa senzoru má defaultní hodnotu 0x41. Při práci s více zařízeními, které mají stejnou adresu, se používá ENABLE pin, kterým se vždy zapne pouze jedno zařízení, s nímž se pak dále pracuje, Dále je zmíněno v podkapitole 3.2 *Program*.

Senzor je vybaven TDC převodníkem.

Tento senzor je používán k měření vzdálenosti robotického vysavače od podlahy. Jinými slovy, tento senzor je používán k detekci možných pádů vysavače ze schodů rozdílných výšek vysávaných ploch.

Dalšími parametry jsou napájecí napětí, které je v rozmezí 2,7 V až 3,3 V, pracovní proud, který odpovídá 1,5 mA a pracovní teplot, která je v rozmezí -30°C až 60°C.

Tyto senzory jsou přímo napájeny z výstupu Arduina UNA 3,3 V.



Obrázek 14: Senzor TMF8701 porovnání naměřené a skutečné vzdálenosti

**Koncová čidla** jsou mechanická a nespojitá. Jejich princip je velmi jednoduchý: při zmačknutí mechanického kontaktu se buď sepnou nebo rozeprnou kontakt. Toto čidlo je v robotickém vysavači nejčastěji využíváno při detekci objektů; jinými slovy, při naražení čidla do objektu se sepnou nebo rozeprnou kontakt. V práci bylo toto čidlo použito pro detekci nárazu do objektu a při nárazu do stěny, kdy čidlo detekuje konec místnosti. Důvodem zvolení čidel je jednoduchost.

Robotický vysavač může obsahovat gyroskop nebo zařízení jim podobná. Gyroskop je zařízení, které si umí uchovat informaci o své poloze, i když se zařízení pohybuje. Proto se v robotickém vysavači používá k orientaci v prostoru.

## 2.4 Akumulátor a jeho nabíjení

Zdrojem napájení robotického vysavače je dvacetivoltový LI-ION (Lithium-Ionový) akumulátor značky „PARKSIDE“.

Tento zdroj byl zvolen kvůli vysoké kapacitě a spolehlivosti akumulátoru. Tento zdroj napětí je také velmi flexibilní - stačí pouze zasunout akumulátor na vrchní straně víka vysavače.

U tohoto zdroje napětí nemusíme také řešit stabilitu napětí a odpadnou také problémy s realizací nabíjecí části vysavače. Nabíjecí zařízení je součástí balení při koupi baterie.

Jedna z mnoha výhod je také rozšířenost těchto akumulátorů na dnešním trhu. Při poškození nebo opotřebení akumulátoru stačí pouze jednoduše koupit nový a vyměnit ho za starý.

Ve většině robotických vysavačů jsou akumulátory přímo vestavěné uvnitř zařízení, zpravidla se také jedná o LI-ION akumulátory.

Robotické vysavače se mohou dále lišit principem nabíjení akumulátorů.

V této práci, jak již bylo zmíněno, je akumulátor nabíjen pomocí nabíjecího zařízení, které se musí přímo připojit do baterie. V dnešní době se u robotických vysavačů častěji používá bezdrátové nabíjení, které funguje na jednoduchém principu elektromagnetické indukce.

Výhody bezdrátového nabíjení jsou, že se akumulátor nezahřívá při napájení a není zapotřebí fyzická nabíječka. Nevýhodou je, že je nabíjení pomalejší a neekonomičtější.

Hlavním rozdílem je, že při bezdrátovém nabíjení se v robotickém vysavači používá bezdrátová nabíjecí stanice, kterou robotický vysavač při poklesu napětí vyhledá a sám se nabije. Toto usnadní při napájení, proto vyžaduje složitý program, který musí mapovat prostor vysávání, proto byla zvolena jednodušší alternativa.

### 3 Softwarová část robotického vysavače

Softwarová část jsou všechny programy, instrukce a data, na základě, kterých se řídí činnost všech částí počítače. Na základě provedení softwaru je úzce spojená volba hardwarových částí a naopak. Čím lepší provedení softwaru v robotickém vysavači je, tím lepší vlastnosti a možnosti pak robotický vysavač může nabízet. Příkladem může být připojení k mobilnímu telefonu, samonabíjecí obsluha vysavače nebo systematickosti vysávání.

Zde jsou zmíněny některé algoritmy na jejichž základě se pak sestavuje celkový software robotického vysavače.

#### Algoritmus pro řízení a ovládání

Algoritmus pro řízení a ovládání je primárním algoritmem. Jeho realizace je úzce spojena s dvěma komponentami. První komponentou, jak již bylo zmíněno v podkapitole 2.2.1 *Pohony*, je pohon. Druhou je pak řídicí jednotka pro vytvoření právě již zmíněného algoritmu. Řídicí jednotka byla zmíněna v podkapitole 2.1 *Řídicí jednotka a pomocné ovládací systémy*.

Samotný algoritmus ovládá zmíněný pohon a následně mechanismus otáčení robotického vysavače. Algoritmem je pak soubor instrukcí na jejichž základě se pak řídí.

#### Algoritmus detekce překážek

Tento algoritmus je také velmi důležitý a bez jeho přítomnosti by robotický vysavač daleko nedeje. Pro funkci tohoto algoritmu musí být přítomen algoritmus zmíněný v předchozím odstavci. Dále musí být přítomny senzory, čidla, které byly již zmíněny v podkapitole 2.3 *Senzory*. Hlavním úkolem je pomocí řídicí jednotky na základě dat ze senzorů ovládat

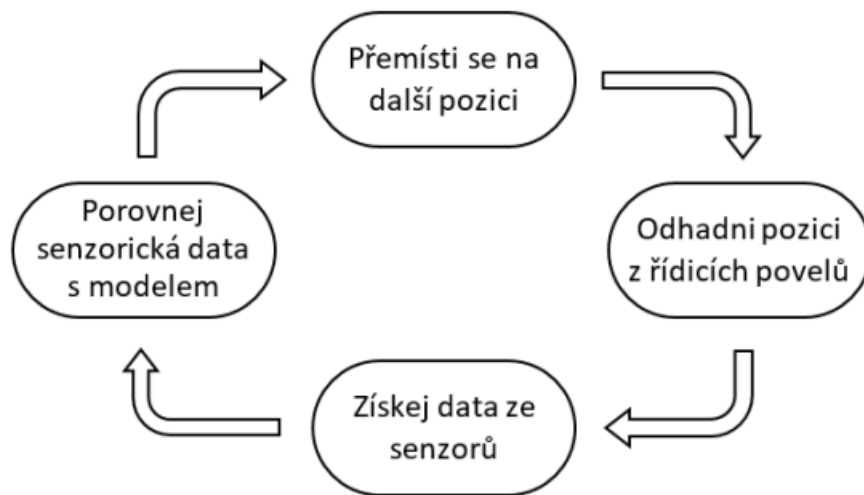
#### Algoritmus pro řízení a ovládání.

Následující dva zmíněné algoritmy již nemusí být v robotickém vysavači přítomny, Tyto algoritmy pouze vylepšují kvalitu vysávání.

#### Algoritmus lokalizace a mapování prostoru

##### Lokalizace

Lokalizace je zjišťování polohy robota vzhledem k předchozí poloze. Nejlépe je znázorněno na *Obrázku: 15 Lokalizace*. V prvním kroku se vykonají řídicí povely, tyto povely jsou instrukce pro řízení pohonů zmíněných v **Algoritmu pro řízení a ovládání**. Na základě těchto povelů lze odhadnout následující polohu, kam se robotický vysavač dostal. Následně proběhne měření uražené vzdálenosti pomocí senzorů. V posledním čtvrtém kroku proběhne porovnání dat ze senzorů a z povelů.



Obrázek 15: Lokalizace

K lokalizaci se používají následující senzory.

a) **Odometry**

Tyto senzory měří počet otočení kol. Jedná se o nejjednodušší způsob lokalizace. Nevýhodou těchto sensorů je, že se špatně detekuje chyba při měření.

**Akcelerometry**

Tyto senzory měří absolutní zrychlení  $\mathbf{a}$ . Na základě zrychlení je pak pomocí vzorce (7) dopočítána poloha  $\mathbf{x}$ . Senzory umožňují měřit ve více než jedné ose.

$$x = \iint a [mm] \quad (7)$$

**Sonary**

Tento druh práce na principu ultrazvukových sensorů. Princip spočívá k vyslání ultrazvukového paprsku k pevnému bodu (stěně), který se následně odrazí a vrátí se zpět k senzoru. Na základě lze počítat dobu trvání letu a dopočítávat vzdálenost.

**Laserová senzory**

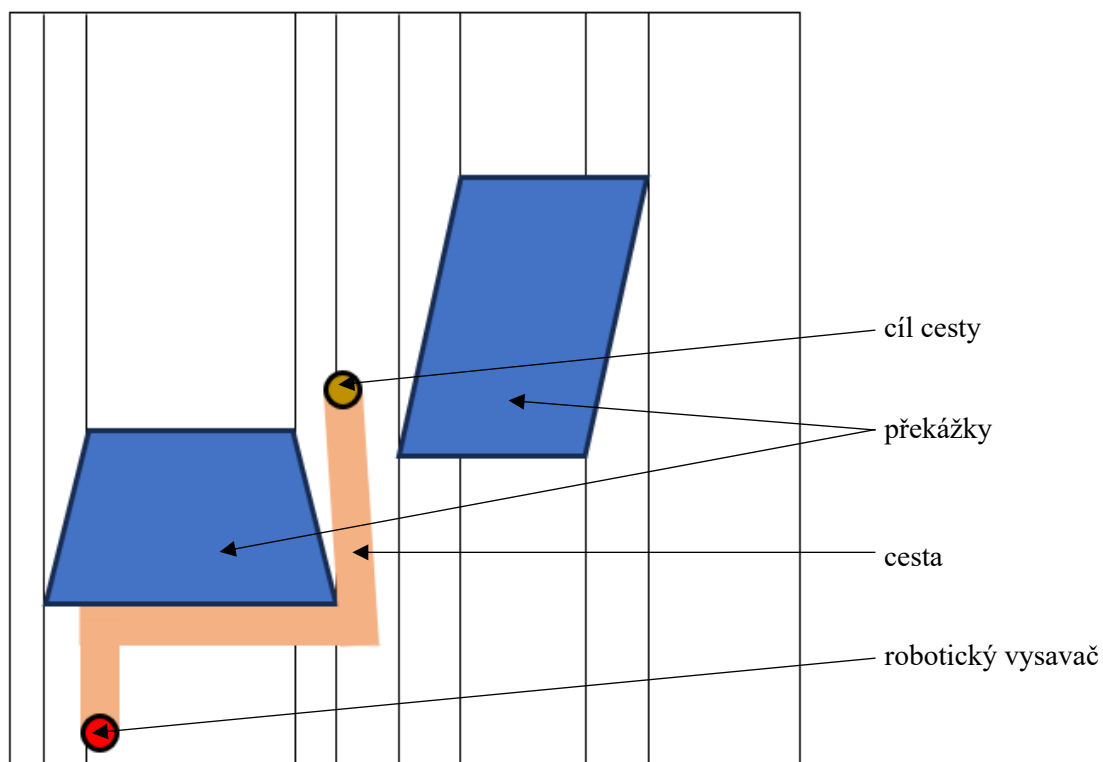
Jedná se o nejpoužívanější senzory. Princip je založen na LIDAR, to je jako u předchozího senzoru měření dobu letu, ale místo doba ultrazvuku se měří doba letu paprsku.

**Mapování**

Tento proces je již uložení lokalizovaných dat a na základě toho vytvoření map.

Využívají se zde sensorické mapy, které se vytváří na základě map ze sensorů. Tyto mapy se ukládají nejčastěji do matic, kde jsou uloženy souřadnice v bodě  $x$ ,  $y$  a případně i  $z$ . Ukládání dat nemusí být neustálé, a to kvůli ušetření místa v paměti, proto nastává aproximační chyba.





Obrázek 17: Lichoběžníkový rozklad

Mapování tras se pak realizuje na základě místa s vyšším výskytem nečistot. Zjištění místa s vyšším výskytem nečistot se pak provádí pomocí měření průtoku nečistot do plnicí násypky.

V robotickém vysavači, který byl vytvořen jsou zhotoveny pouze první dva zmíněné algoritmy, a to **Algoritmus pro řízení a ovládání** a **Algoritmus pro detekci překážek**. Realizace posledních dvou algoritmů by byla složitá, proto bylo upuštěno od jejich realizace.

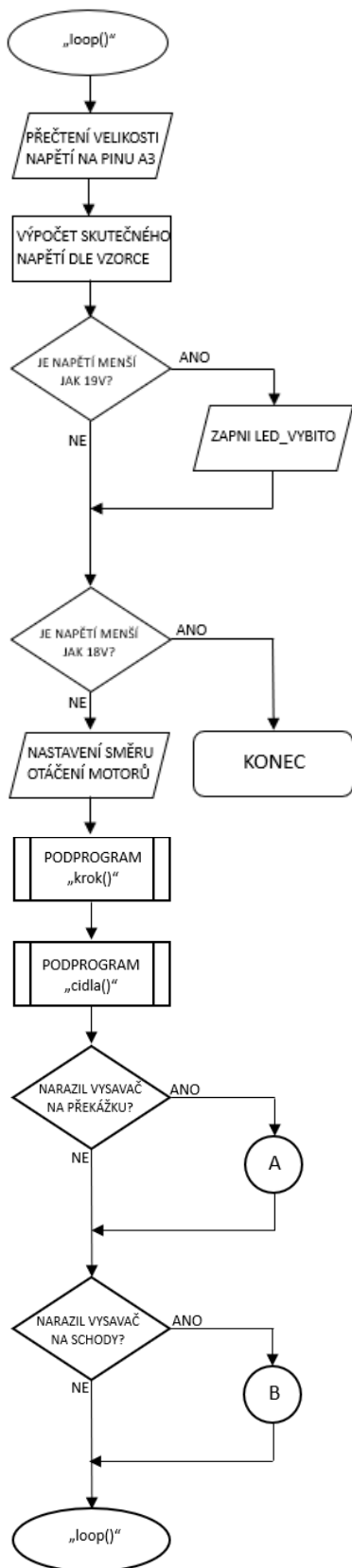
### 3.1 Vývojový diagram programu

Vývojový diagram složí ke grafickému znázornění jednotlivých kroků algoritmů, procesů, operací. Nejčastěji se používá k vizualizaci toku dat v programování.

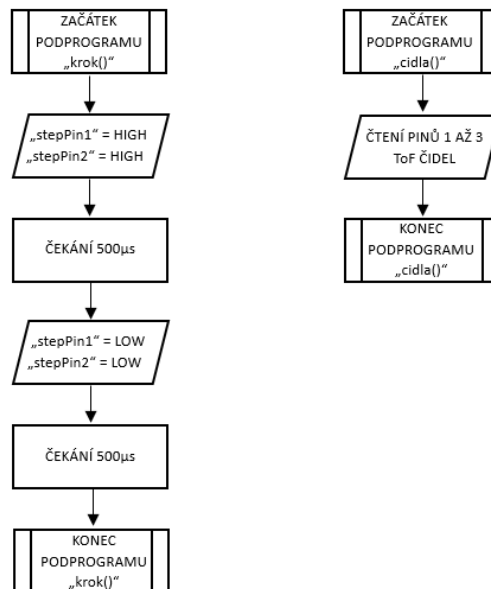
Následující obrázky budou věnovány vývojovému diagramu, který zobrazuje jednotlivé kroky programu zmíněný v podkapitole 3.2 *Program*.



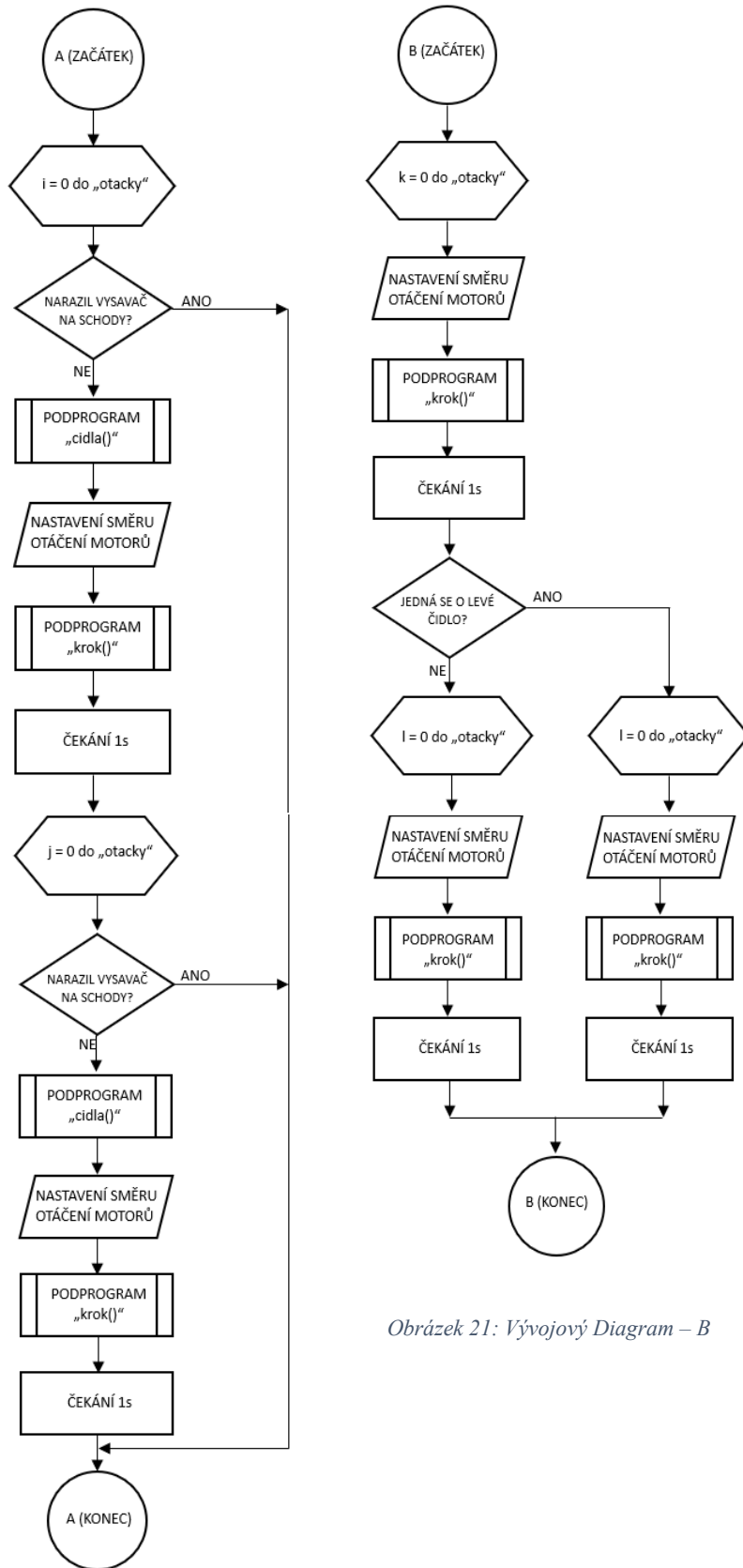
Obrázek 18: Vývojový Diagram



Obrázek 19: Vývojový Diagram – loop()



Obrázek 20: Vývojový Diagram – Podprogramy



Obrázek 21: Vývojový Diagram – B

Obrázek 22: Vývojový Diagram – A

## 3.2 Program

Tato podkapitola je věnována použitému programu v robotickém vysavači. Program bude bod po bodu vysvětlen vždy pod jeho částí. Jako programovací prostředí bylo použito již zmíněné Arduino IDE. Programovací prostředí Arduino IDE pracuje s programovacím jazykem Arduino Programming Language, který je založen na programovacím jazyce C++.

```
#include "DFRobot_TMF8x01.h"

// DIGITÁLNÍ
#define step_pin1 2
#define dir_pin1 3
#define step_pin2 4
#define dir_pin2 5
#define vypinaci_rele 6
#define rele_motoriky 7
#define rele_sani 8
#define EN3 10 // enable, senzor 3 vpravo
#define EN1 11 // enable, senzor 1 vlevo
#define EN2 12 // enable, senzor 2 střed
#define koncova_cidla 13

// ANALOGOVÉ
#define led_zapnuto A0
#define led_vybity A1
#define napeti_pin A3

#define INT -1 //Zakázání pinu přerušení na ToF senzorech
```

V této části programu jsou definovány názvy používaných pinů, aby se s nimi lépe pracovalo a aby jejich názvy byly podobné funkci, kterou vykonávají. Všechny analogové i digitální piny jsou deklarovány pomocí příkazu `#define`. Na prvním řádku je použita jediná knihovna „DFRobot\_TMF8x01.h“ pro práci s ToF senzory, které budou dále zmíněny v další části programu.

V prvním odstavci jsou definovány digitální piny. Na prvních čtyřech řádcích je definice pinů, které slouží k práci s krokovými motory. Piny „step\_pin1“ a „step\_pin2“ slouží k pootočení hřídelky krokových motorů. Piny „dir\_pin1“ a „dir\_pin2“ mají za úkol určit směr otáčení hřídelí krokových motorů.

Za piny, které slouží k ovládání krokových motorů, se nacházejí tři piny „vypinaci\_rele“, „rele\_motoriky“ a „rele\_sani“, které slouží k připojení vývodů na ovládací relé, kterými se dále ovládá zapnutí, či vypnutí, relé.

Posledními piny v sekci „DIGITÁLNÍ“, jsou čtyři piny, které slouží k práci se senzory. Na prvních třech pinech „EN1“ až „EN3“ jsou přivedeny přívody ze senzorů TMF8701, konkrétně přívody z enablů pinů, které slouží k zapínání (povolování) konkrétního senzoru. Na posledním, čtvrtém pinu, jsou pak sériově připojeny koncové senzory pro detekci překážky.

Ve druhé odstavci v sekci „ANALOGOVÉ“ jsou definovány tři piny. První dva piny „led\_zapnuto“ a „led\_vypnuto“ slouží k připojení LED diod, které signalizují zapnutí vysavače a nízkou úroveň napětí na akumulátoru. Poslední piny „napeti\_pin“ slouží k měření úrovně napětí na akumulátoru.

Poslední definovaný je pak pin „INT“, který zakazuje přerušování na senzorech TMF8701.

```
// POMOCNÉ PROMĚNNÉ
int stav_ToF_L = 0;
int stav_ToF_S = 0;
int stav_ToF_P = 0;
int hodnota_napeti_na_pinu = 0;
int i = 0;
float napeti = 0.0;
```

V této části programu jsou použity pomocné proměnné, které jsou dále využívány v programu. Všechny pomocné proměnné jsou inicializovány na hodnotu 0, aby se zabránilo náhodnému přiřazení jiné hodnoty a následnému selhání programu.

První tři pomocné proměnné „stav\_ToF\_x (L,S,P)“ slouží k ukládání stavu z tří senzorů TMF8701. Tento stav je vzdálenost naměřená zmíněnými senzory. Proměnné jsou datového typu integer, protože senzory vrací celočíselné hodnoty.

Další proměnou je „hodnota\_napeti\_na\_pinu“, která ukládá digitální reprezentaci naměřené hodnoty na pinu „napeti\_pin“, jak bylo zmíněno v předchozí části kódu.

Proměnná „i“ je klasická pomocná proměnná, která se používá ve *for* cyklech.

Poslední proměnou v tomto odstavci je „napeti“, která je jediná s datovým typem *float* a slouží k ukládání již přepočítané hodnoty digitálního napětí na analogový signál. Proto je hodnota uložena v neceločíselné proměnné.

```
DFRobot_TMF8701 sensor1(EN1, INT);
DFRobot_TMF8701 sensor2(EN2, INT);
DFRobot_TMF8701 sensor3(EN3, INT);
```

V této části kódu probíhá inicializace senzorů TMF8701. Jedná se o inicializaci pomocí funkce „DFRobot\_TMF8701 sensor()“, z jichž zmíněné knihovny „DFRobot\_TMF8x01.h“. Tato již předdefinovaná funkce potřebuje minimálně dva parametry. Prvním parametrem je pin, na který je připojen enable pin a druhým parametrem je pin, na který je připojen pin přerušování. Enable piny jsou již definované v první části kódu a ve stejné části kódu je i zakázaný pin pro přerušování.

```
void setup()
{
    Serial.begin(9600);

    // DIGITÁLNÍ
    pinMode(step_pin1, OUTPUT);
    pinMode(dir_pin1, OUTPUT);
    pinMode(step_pin2, OUTPUT);
    pinMode(dir_pin2, OUTPUT);
    pinMode(vypinaci_rele, OUTPUT);
    pinMode(rele_motorky, OUTPUT);
    pinMode(rele_sani, OUTPUT);
```

```

pinMode(EN1, OUTPUT);
pinMode(EN2, OUTPUT);
pinMode(EN3, OUTPUT);
pinMode(koncova_cidla, INPUT_PULLUP);

// ANALGOVÉ
pinMode(napeti_pin, INPUT);
pinMode(led_zapnuto, OUTPUT);
pinMode(led_vybito, OUTPUT);

// DIGITÁLNÍ
digitalWrite(dir_pin1, LOW);
digitalWrite(dir_pin2, LOW);
digitalWrite(vypinaci_rele, HIGH);
digitalWrite(rele_motorky, HIGH);
digitalWrite(rele_sani, HIGH);

// ANALGOVÉ
analogWrite(led_zapnuto, 255);
analogWrite(led_vybito, 0);

// Inicializace ToF senzoru 1
digitalWrite(EN2, LOW);
digitalWrite(EN3, LOW);
digitalWrite(EN1, HIGH);
sensor1.begin();
sensor1.startMeasurement(sensor1.eModeCalib, sensor1.eCOMBINE);

// Inicializace ToF senzoru 2
digitalWrite(EN1, LOW);
digitalWrite(EN3, LOW);
digitalWrite(EN2, HIGH);
sensor2.begin();
sensor2.startMeasurement(sensor2.eModeCalib, sensor2.eCOMBINE);

// Inicializace ToF senzoru 3
digitalWrite(EN1, LOW);
digitalWrite(EN2, LOW);
digitalWrite(EN3, HIGH);
sensor3.begin();
sensor3.startMeasurement(sensor3.eModeCalib, sensor3.eCOMBINE);
}

```

V této části kódu je zobrazená setupová funkce, která se provede pouze jednou při spuštění programu. Zde se inicializují a nastavují parametry zařízení.

Na prvním řádku je příkaz pro spuštění sériového monitoru, který zobrazuje úkony připojeného zařízení (mikrokontroleru) a počítače. 9600 (baud) určuje rychlost přenosu, tedy 9600 bitů za sekundu.

Ve dvou dalších odstavcích „DIGITÁLNÍ“ a „ANALOGOVÝ“ se nastavují piny (porty) mikrokontroleru, buď jako vstupu nebo jako výstupu . Nastavují se pomocí funkce *PINMODE()*. Tato funkce má dva parametry: první je název pinu, který chceme nastavit, a druhý určuje směr nastavení. U nastavení jako vstup lze také zapnout vnitřní pull-up rezistor, který má hodnotu v řádu několik desítek kilo ohmů. Tento rezistor se používá u pinů, na které jsou připojena čidla pro detekci nárazu. Při měření napětí na pinu „napeti\_pin“ se tento rezistor nepoužívá kvůli zkreslení měřeného napětí.

V dalších dvou odstavcích se pomocí funkcí *digitalWrite()* (pro digitální piny) či *analogWrite()* (pro analogové piny) nastavuje konkrétní hodnota na pinu, buď *LOW* (logickou 0, tedy 0 V) nebo *HIGH* (logickou 1, tedy 5 V). Tyto funkce mají dva parametry: první je název pinu, který chceme nastavit, a druhý určuje logickou úroveň.

V posledních třech odstavcích probíhá inicializace tří ToF senzorů TMF8701. Každý senzor je připojen na I2C sběrnici, a proto je nutné zapnout konkrétní senzor a ostatní zařízení vypnout. Zapnutí a vypnutí konkrétního zařízení probíhá pomocí pinu *enable*. Dále je pomocí funkce *sensor.begin()* zapnuta komunikace se senzorem a následně je nastaven režim senzoru funkcí *sensor.startMeasurement()*. Prvním parametrem je výběr, zda chceme použít kalibraci, a druhým je volba mezi režimy, které jsou dále popsány v podkapitole 2.3 *Sensory* v odstavci *DFRobot TMF8701*.

```
void krok()
{
    // Krok motoru
    for (int i = 0; i < 400; i++ )
    {
        digitalWrite(step_pin1, HIGH);
        digitalWrite(step_pin2, HIGH);
        delayMicroseconds(500);
        digitalWrite(step_pin1, LOW);
        digitalWrite(step_pin2, LOW);
        delayMicroseconds(500);
    }
}

void cidla() // Čtení ToF senzorů
{
    digitalWrite(EN2, LOW);
    digitalWrite(EN3, LOW);
    digitalWrite(EN1, HIGH);
    sensor1.begin();
    sensor1.startMeasurement(sensor1.eModeCalib, sensor1.eCOMBINE);

    if (sensor1.isDataReady())
    {
        stav_ToF_L = sensor1.getDistance_mm();
        Serial.print("Vzdálenost senzoru 1: ");
        Serial.print(stav_ToF_L);
        Serial.println(" mm");
    }
}
```

```

}

digitalWrite(EN1, LOW);
digitalWrite(EN3, LOW);
digitalWrite(EN2, HIGH);
sensor2.begin();
sensor2.startMeasurement(sensor2.eModeCalib, sensor2.eCOMBINE);

if (sensor2.isDataReady())
{
    stav_ToF_S = sensor2.getDistance_mm();
    Serial.print("Vzdálenost senzoru 2: ");
    Serial.print(stav_ToF_S);
    Serial.println(" mm");
}

digitalWrite(EN1, LOW);
digitalWrite(EN2, LOW);
digitalWrite(EN3, HIGH);
sensor3.begin();
sensor3.startMeasurement(sensor3.eModeNoCalib, sensor3.eCOMBINE);

if (sensor3.isDataReady())
{
    stav_ToF_P = sensor3.getDistance_mm();
    Serial.print("Vzdálenost senzoru 3: ");
    Serial.print(stav_ToF_P);
    Serial.println(" mm");
}
}

```

V této části programu jsou napsané dvě funkce, které usnadňují rutinní zápis.

První funkce „krok()“ slouží k pootočení oběma použitými krokovými motory o jeden krok (1,8°). Princip spočívá v tom, že se na pinu „step\_pin“, který slouží k otáčení krokového motoru, objeví stav *HIGH*. Poté se po krátkou dobu počká, v našem případě 0,5 ms. Tato velikost určuje rychlost otáčení motoru. Následně se pin nastaví na stavu *LOW*. V našem případě se jedná o *eight step*, kdy se motor při přivedení pulzu otočí o osminu kroku, což odpovídá 0,225°. Tento proces se opakuje 400krát. Úhel pootočení je tedy 45° místo očekávané velikosti 90°. Poloviční velikost je způsobena poměrem mezi hřídelí kolečka a hřídelí krokového motoru, který je 2:1, jak je zmíněno v podkapitole *1.1 3D tisk a modelovací program*.

Arduino IDE nabízí předem předefinovanou knihovnu, kde stačí jen zadávat parametry.

Druhá funkce „cidla()“ slouží k přečtení stavů ToF senzorů z analogového vstupu. Nejprve se provádí inicializace senzorů. Poté se zajišťuje pomocí příkazu *if* a funkce *sensor.isDataReady()*, zda senzor naměřil nová data. Pokud ano, data se přečtou pomocí funkce *sensor.getDistance()*, uloží do pomocné proměnné a zobrazí na sériovém monitoru.

```

void loop()
{

```

```

hodnota_napeti_na_pinu = analogRead(napeti_pin);
napeti = (hodnota_napeti_na_pinu * (5.0 / 1023.0) * 6);

if (napeti < 19.0)
{
    analogWrite(led_vybito, 255);
}

if (napeti < 18.0)
{
    digitalWrite(vypinaci_rele, LOW);
}

krok();
cidla();

```

V této části programu je nejprve měřeno napětí na pinu „napeti\_pin“. Pomocí funkce *analogRead()* se na tomto pinu přečte hodnota a uloží se do proměnné „hodnota\_napeti\_na\_pinu“. Tato hodnota je v digitální formě a má rozsah od 0 do 1023, což odpovídá 10bitovému analogově-digitálnímu převodníku (ADC). Proto se v dalším kroku tato hodnota vynásobí referenčním napětím 5 V a vydělí hodnotou 1023. Násobení šestkou je pak dáno poměrem rezistorů mezi pinem „napeti\_pin“ zmíněné dále v podkapitole 4.1.5 *Rezistory 100 kΩ a 20 kΩ*.

Dále v programu následují dvě podmínky. První podmínka je splněna, když napětí klesne pod hranici 19 V. Na základě tohoto faktu se rozsvítí LED dioda, která signalizuje nízké napětí. Druhá podmínka nastává při poklesu napětí na 18 V, což znamená, že se celý robotický vysavač vypne. Hodnota 18 V odpovídá 90 % z maximálního napětí akumulátoru. Tato podmínka zajišťuje ochranu akumulátoru, aby napětí nepokleslo pod hodnotu 85 % a akumulátor nebyl poškozen. Tato hodnota je dána výrobcem akumulátorů.

V poslední části programu jsou volány již zmíněné dva podprogramy.

```

if (stav_ToF_L > 100 || stav_ToF_S > 100 || stav_ToF_P > 100)
{
    digitalWrite(dir_pin1, HIGH);
    digitalWrite(dir_pin2, HIGH);
    for (i = 0; i < 15; i++)
        krok();
    digitalWrite(dir_pin1, HIGH);
    digitalWrite(dir_pin2, LOW);
    for (i = 0; i < 15; i++)
        krok();
    digitalWrite(dir_pin1, LOW);
    digitalWrite(dir_pin2, LOW);
}

if (digitalRead(koncova_cidla) == LOW)
{
    digitalWrite(dir_pin1, HIGH);
}

```

```

digitalWrite(dir_pin2, HIGH);
for (i = 0; i < 7; i++)
    krok();
digitalWrite(dir_pin1, HIGH);
digitalWrite(dir_pin2, LOW);
for (i = 0; i < 7; i++)
    krok();
digitalWrite(dir_pin1, LOW);
digitalWrite(dir_pin2, LOW);
    }
}

```

V této části programu jsou dvě podmínky.

První podmínka je pro detekci schodů. Při splnění podmínky, kdy výška senzoru od podlahy bude větší než 10 cm, se vykoná patřičná rutina. Prvně vysavač couvne o patřičný krok a poté se otočí o určitý úhel. Vzdálenost couvnutí je zhruba 33 cm. Úhel, o který se otočí, je  $137,5^\circ$ .

Druhá podmínka je pro detekci překážky. Při splnění podmínky, kdy robotický vysavač narazí na překážku, se vykoná patřičná rutina. Prvně vysavač couvne o patřičný krok a poté se otočí o určitý úhel. Vzdálenost couvnutí je zhruba 16,5 cm. Úhel, o který se otočí, je  $68,75^\circ$ .

Důvodem pro zvolení těchto konkrétních vzdáleností a úhlů byly zmíněny v podkapitole *1.1 3D tisk a modelovací program*. Úhly a vzdálenosti couvnutí jsou menší než v podkapitole *1.1 3D tisk a modelovací program*, a to kvůli *for* cyklům. U obou podmínek, se úhel pootočení liší od  $360^\circ$  ( $2\pi$ ). Za předpokladu, že úhle otočení je v podprogramu „krok()“ již zmíněný ve vyšším odstavci  $45^\circ$ , tak v případě prvního odstavce, kdy cyklus proběhne 15 krát dostáváme  $15 \cdot 45$ , to je  $675^\circ$  ( $15/16\pi$ ). V případě druhém je to  $7 \cdot 45$  to je  $315^\circ$  ( $7/8\pi$ ).

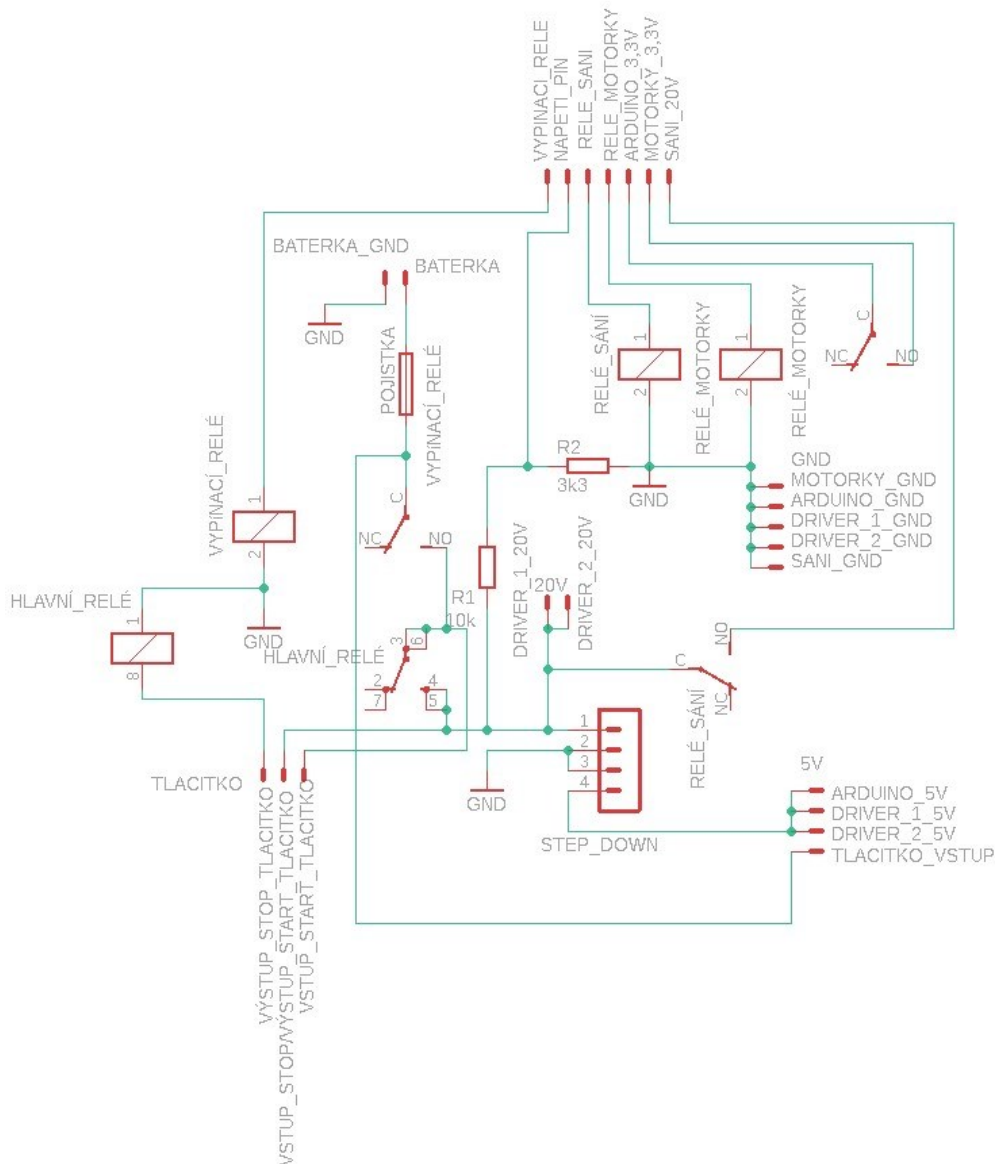
Těmito úhly byla záměrně změněna cesta pohybu, aby se robotický vysavač nevracel po stejné dráze.

## 4 Návrh desky plošných spojů

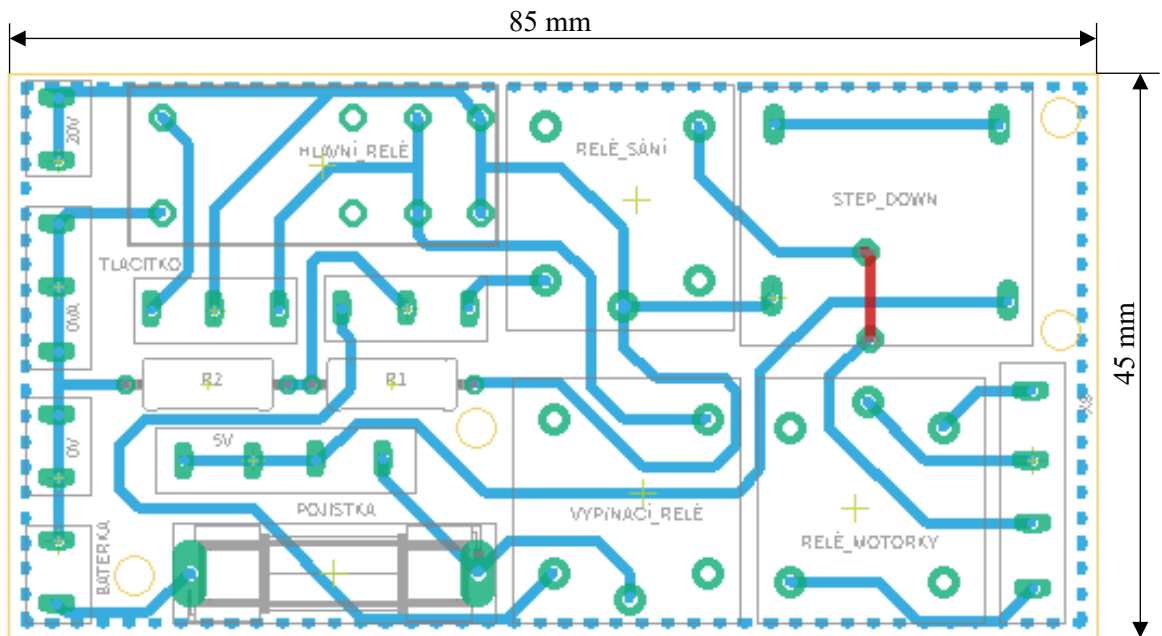
Deska plošných spojů DPS, jak už bylo zmíněno v podkapitole 2.1.1 *Používané řídicí jednotky v robotickém vysavači*, je deska vyrobená převážně z laminátu a z jedné nebo více stran je nanesená měděná vrstva, na které se nachází elektronické součástky, které jsou mezi sebou propojeny vodiči, nebo vodivými cestami.

Pro zjednodušení práce a zlepšení přehlednosti v robotickém vysavači bylo ve vývojovém softwarovém prostředí s názvem *Eagle* navrženo schéma desky plošných spojů, viz *Obrázek 23: Schéma DPS*, a na jeho základě byla navrhována samotná deska, viz *Obrázek 31: Deska plošných spojů*, podle předem zvolených součástek.

Na desce plošného spoje se nachází tavná pojistka s nominální hodnotou 2 A, tři výkonové relé Typu SRD-05VDC-SL-C a jedno elektromagnetické relé Typu 44.52 značky Finder. Dva rezistory, jeden o hodnotě 100 k $\Omega$  a druhý o hodnotě 20 k $\Omega$ , step down měnič napětí a několik terminálů.



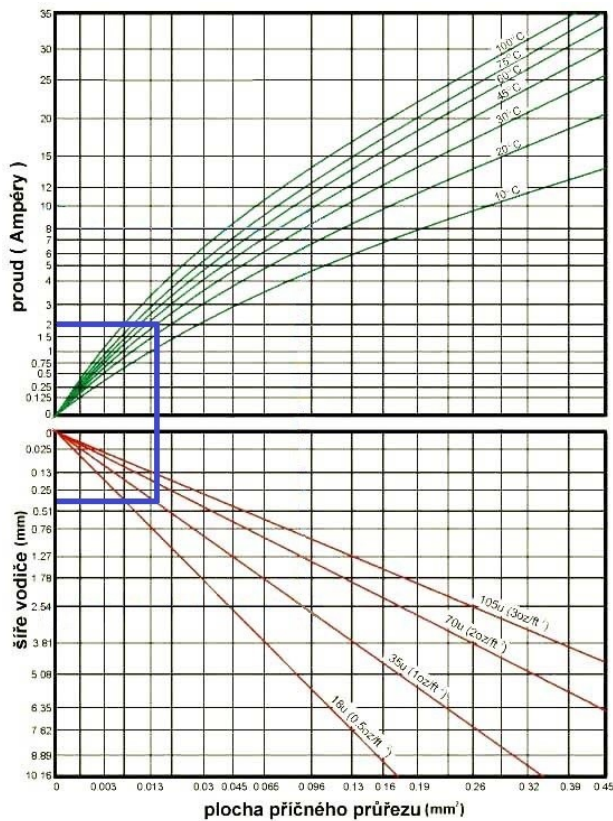
Obrázek 23: Schéma DPS



Obrázek 24: DPS

Vodivé cesty na desce mají šířku 1,27 mm. Deska je pokrytá měděnou folií o tloušťce 35  $\mu\text{m}$ . Podle Grafu 2: Proudová zatížitelnost DPS je tato šířka vodivých cest dostačující pro maximální proud 2 A.

Prokovy jsou voleny velikostí 0,55 mm a úchyt desky má šířku 3,2 mm.



Graf 2: Proudová zatížitelnost

## 4.1 Použité komponenty na DPS

### 4.1.1 SRD-05VDC-SL-C

Jedná se o výkonové relé s maximálním výkonem 0,45 W. Hlavní cívka má odpor 70  $\Omega$  a je spínána napětím 5 VDC. Relé má spínací/rozpínací kontakty typu SPDT, jeden spínací (NO) a jeden rozpínací (NC) pól, který maximálně spíná napětí 30 VDC a proud 10 A.

Na desce plošných spojů, která byla navržena, se nachází tři tato relé.

První, označené jako „VYPÍNACÍ\_RELÉ“ slouží k vypnutí celého obvodu vysavače při poklesu napětí pod bezpečnou mez, která je stanovena na 90 % z 20 VDC, tedy 18 VDC. Zde je použit kontakt typu NO. Lepším řešením by bylo použití kontaktu typu NC, který by umožnil snadné rozepnutí příkazem z Arduina. Avšak čas, který je potřeba k rozepnutí relé, je delší než čas, po který dojde k odpojení Arduina. Proto byly použity kontakty typu NO. Použitím kontaktů typu NO je vyžadován druhý napájecí okruh, aby Arduino stihlo naběhnout. Tento problém řeší „START\_TLAČÍTKO“ spolu s druhým „STEP\_DOWN“. Tlačítko, když je zmáčknuto, spíná napětí ze step downu a spíná pin pro sepnutí „VYPÍNACÍ\_RELÉ“, při puštění již drží relé sepnuté díky signálu z Arduina.

Druhé relé s označením „RELÉ\_SÁNÍ“ slouží k sepnutí obvodu sacího ventilátoru. Zde je použit kontakt typu NO, což umožňuje sepnutí obvodu sání pouze při signálu z řídicí logiky.

Poslední relé s označením „RELÉ\_MOTORKY“ slouží k sepnutí obvodu, ve kterém se nachází dva motory, které slouží jako pohon pro zametací zařízení. Zde je použit kontakt typu NO, což umožňuje sepnutí obvodu s motoriky pouze při signálu z řídicí logiky.

### 4.1.2 FINDER TYPE 44.52

Jedná se o elektromagnetické relé s maximálním výkonem 650 mW. Hlavní cívka má odpor o velikosti 900  $\Omega$  a je spínána napětím 24 VDC, avšak v našem případě pouze 20 VDC, což však nemá vliv na funkčnost relé. Relé disponuje spínacími/rozpínacími kontakty typu DPDT, jedná se tedy o dvou pólové relé, dva spínací a dva rozpínací kontakty. V práci jsou použity kontakty typu NO, ale lze použít i zapojení s kontakty NC. Tyto kontakty spínají/rozpínají maximální napětí 30 VDC a proud 6 A. Relé je použito k přepínání mezi tlačítky „STOP\_TLAČÍTKO“ a „START\_TLAČÍTKO“.

### 4.1.3 Step down

Step down, také nazývaný měnič napětí, slouží k regulaci velikosti napětí. Tento konkrétní step down mění napětí z 20 V na 5 V. Jeho výhodou je, že i při poklesu vstupního napětí v rozmezí 5,3 V až 26 V zůstane napětí na výstupu konstantní.

Tento Step down je využíván k napájení Arduina UNO, z něhož jsou pak napájena všechna čidla. Step down také slouží k napájení logiky Adaptérů A4988 DRV8825, na které jsou připojeny drivery DRV8825 pro řízení krokových motorů.

### 4.1.4 Tlačítka

Pro zapnutí a vypnutí vysavače jsou použita dvě tlačítka. Jedná se o tlačítka značky „EATON“ s označením ac15/230v/4a. Tlačítko pro zapnutí označované jako „START\_TLAČÍTKO“ je typu NC, tlačítko pro vypnutí, označované jako „STOP\_TLAČÍTKO“ je typu NC.

#### 4.1.5 Rezistory 100 kΩ a 20 kΩ

Rezistory 100 kΩ a 20 kΩ jsou zapojeny ke vstupu Arduina na pin A3 jako dělič napětí, jak je zobrazeno na *Obrázku 25: Zapojení rezistorů 100 kΩ a 20 kΩ*. Tento dělič napětí slouží k detekci poklesu napětí na akumulátoru. Na rozdíl od step downu konvertoru je u tohoto děliče žádoucí, aby se napětí lineárně měnilo na vstupu Arduina UNA v poměru napětí na akumulátoru.

Hodnota rezistorů je volena v řádech desítek kilo ohmů, aby byl proud protékající děličem co nejmenší. Vzorec nezátíženého děliče je pak dán rovnicí (8)

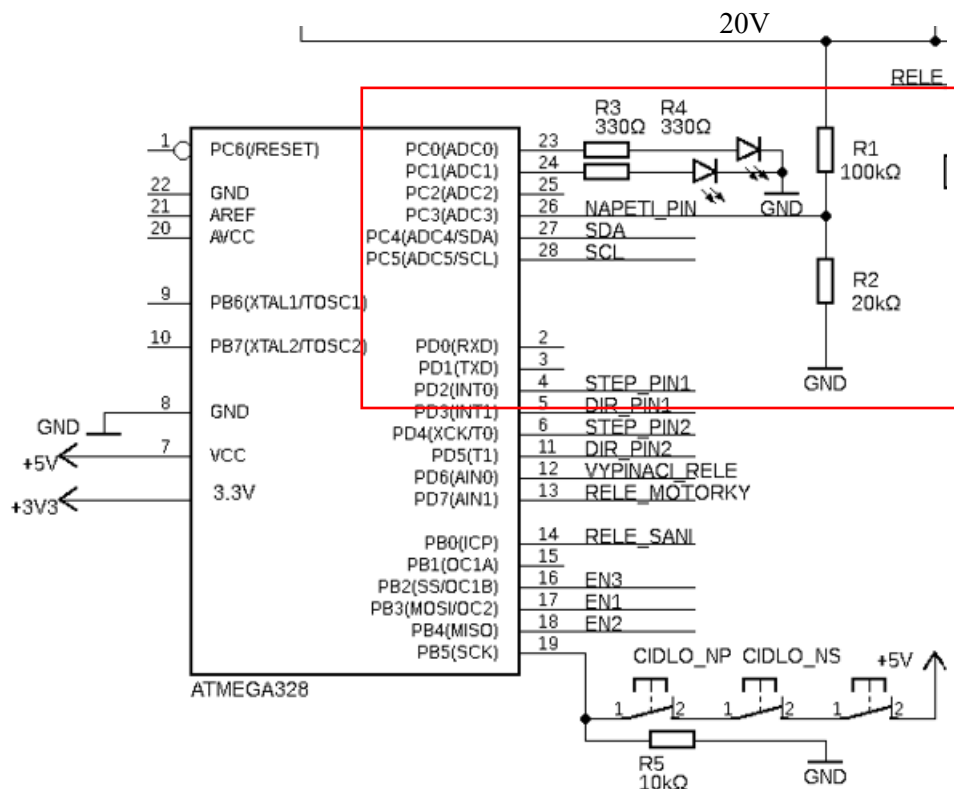
$$U_{arduino} = \frac{R_2}{R_1 + R_2} \times U_{20} \text{ [V]} \quad (8)$$

Velikost  $U_{arduino}$  je dán maximálním napětím na vstupu Arduina UNA, zatímco napětí  $U_{20}$  je určeno maximálním napětím, které může být na akumulátoru.

Poměr rezistorů je stanoven jako 1:5, což zajistí, že maximální napětí  $U_{arduino}$  nepřesáhne napětí 5 V.

Při dosazení hodnoty odpor  $R_2 = 20 \text{ k}\Omega$ , odporu  $R_1 = 100 \text{ k}\Omega$  a napětí  $U_{20} = 20 \text{ VDC}$  do vzorce (8) vyjde velikost napětí  $U_{arduino} = 3,3 \text{ VDC}$ . Tento výstupní napětíový rozsah odpovídá horní hranici, kterou dokáže Arduino UNO zpracovat.

Ideální poměr by byl 1:3, aby se napětí  $U_{arduino}$  rovnalo horní hranici, kterou dokáže Arduino UNO zpracovat, což by umožnilo využití celého rozsahu A/D převodníku a minimalizovalo by pásmo necitlivosti. Nicméně rezistory jsou zvoleny ve větším poměru kvůli riziku, že by se na baterii mohlo objevit napětí vyšší než 20 VDC, a také kvůli nedostatku rezistorů s jiným poměrem.

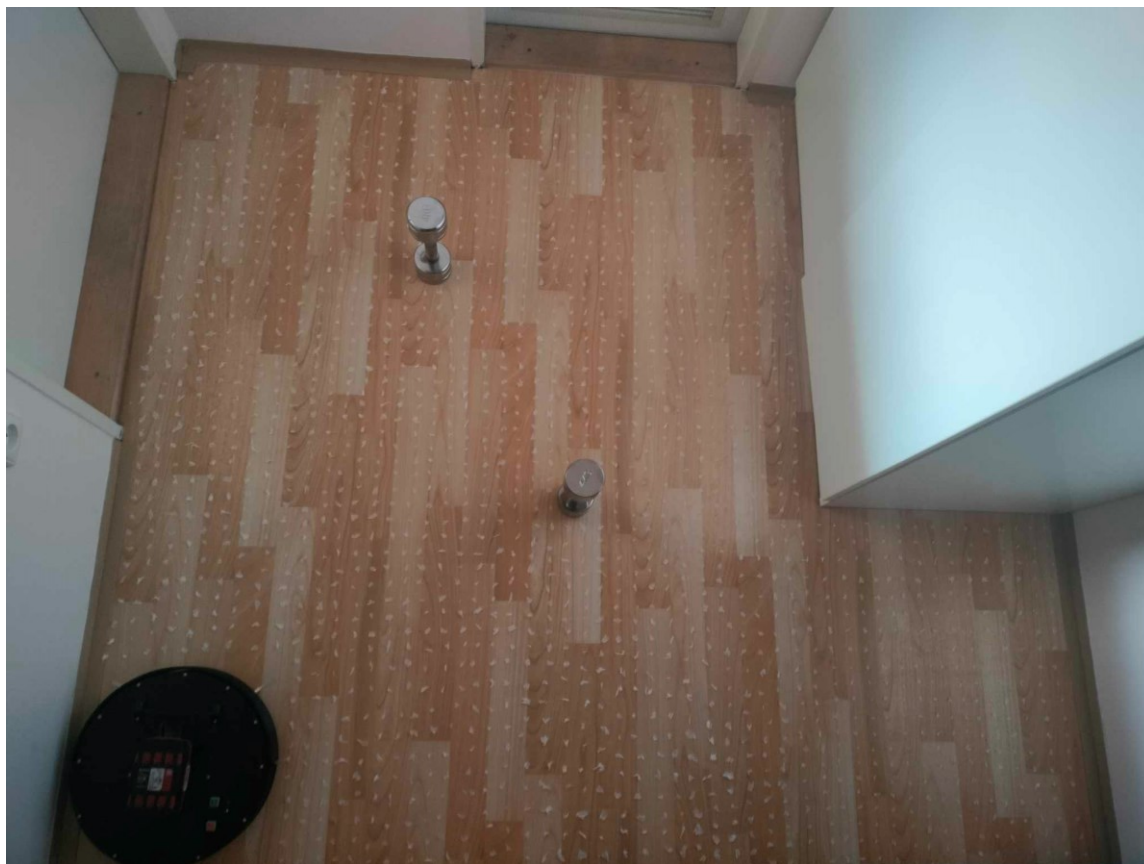


Obrázek 25: Zapojení rezistorů 100 kΩ a 20 kΩ

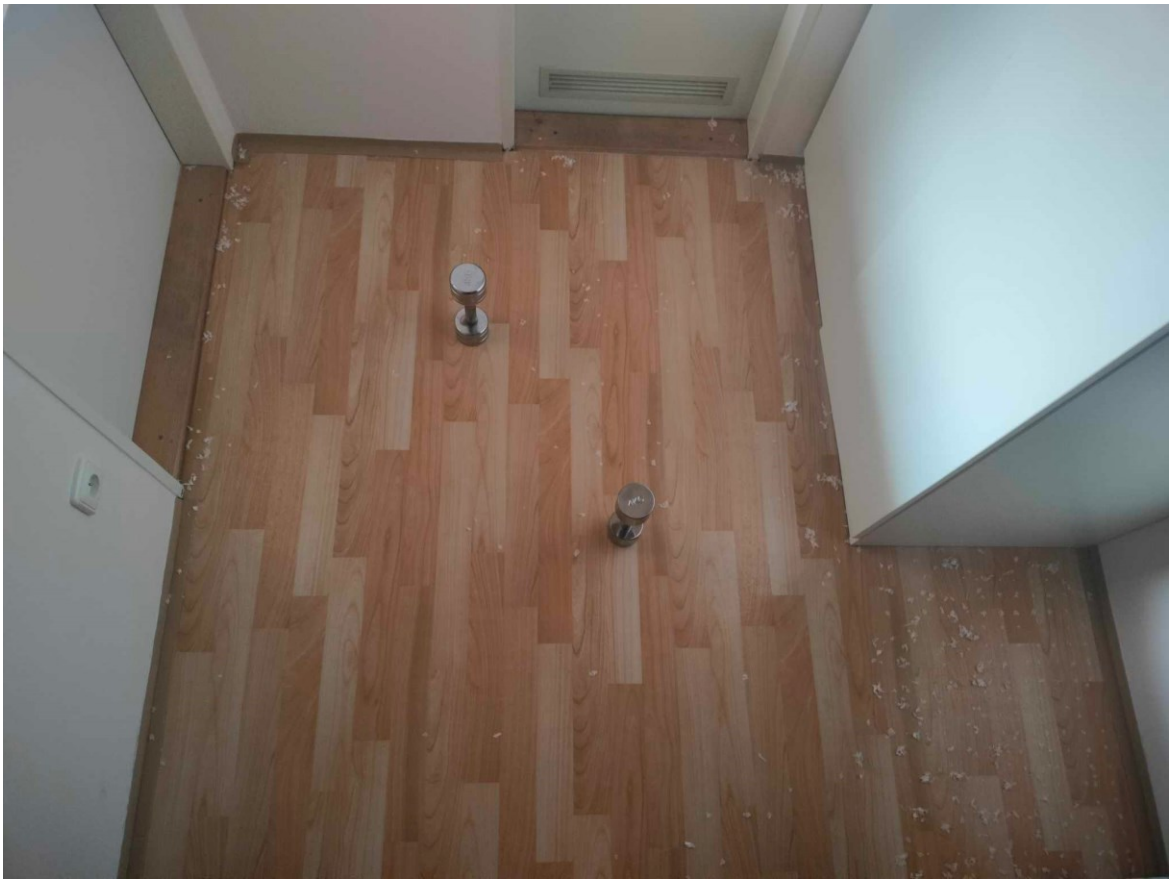
## 5 Vysávání

V této kapitole jsou zdokumentovány výsledky vysávání.

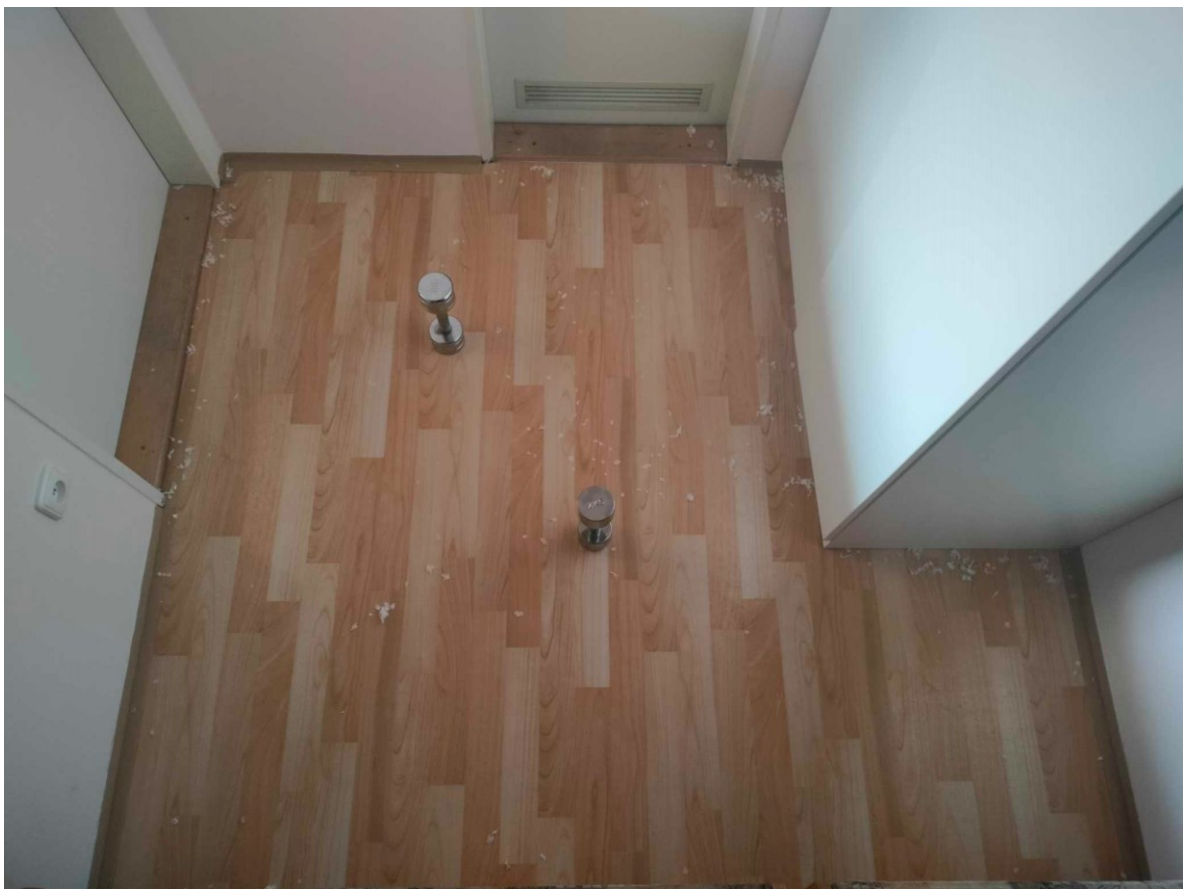
Bylo rozmístěno cirká 2500 útržků papírového kapesníku a dvě překážky. Samotný test byl proveden v místnosti ve tvaru L. Test probíhal po dobu patnácti minut, přičemž každých pět minut byla pořízena fotografie výsledku vysávání.



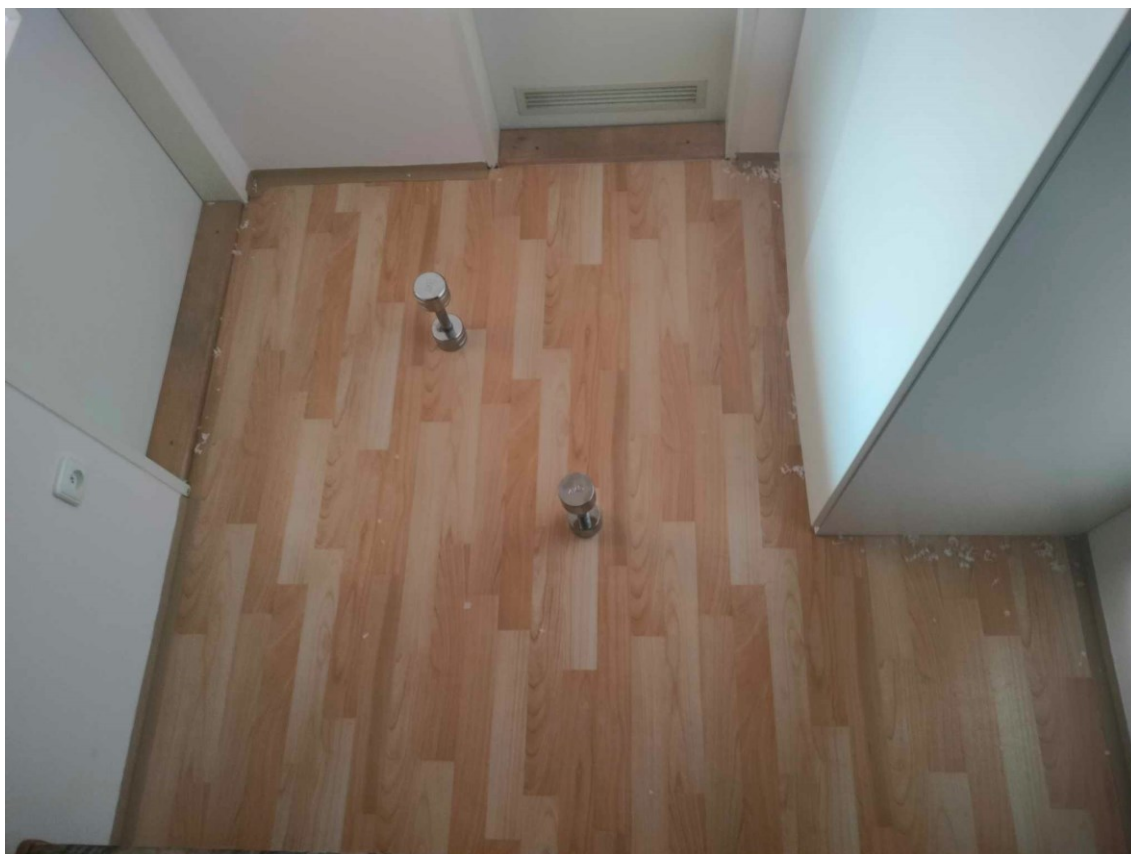
Obrázek 26: Místnost před vysáváním



*Obrázek 27: Místnost po pěti minutách vysávání*



*Obrázek 28: Místnost po deseti minutách vysávání*



*Obrázek 29: Místnost po patnácti minutách vysávání*

Z *Obrázku 27: Místnost po pěti minutách vysávání* lze vypořovat, že vysavač do pěti minut nenavštívil pravý horní roh místnosti. Dále je vidět, že místnost po deseti minutách byla již celkem dobře vysáta, jak lze vidět na *Obrázku 28: Místnost po deseti minutách vysávání*. Z posledních dvou obrázků *Obrázek 28: Místnost po deseti minutách vysávání* a *Obrázek 29: Místnost po patnácti minutách vysávání*, je vidět, že rozdíl v kvalitě vysátí již není velký.

## Závěr

V první řadě je potřeba říci, že se realizace robotického vysavače zdařila a vysavač je plně funkční k vysávání domácích ploch.

Některé věci na robotickém vysavači lze provést lépe.

První věcí je samotné sání, které nešlo realizovat jinak, a to kvůli neumožnění získání informací o konkrétních sacích motorech. Na druhou stranu, i pomocí zmíněného sání a využití jak zametacích smetáků, tak planžety, se dosáhlo uspokojivých výsledků. Vysavač dokáže vysát drobných nečistot, vlasů a chlupů.

Druhou věcí je snímání možnosti pádu ze schodiště a vyšších míst. Senzory pracují v delším časovém intervalu, a to kvůli opakované inicializaci, která je na I2C sběrnici nutná. Řešením tohoto problému je práce senzorů bez knihovny, čímž se sníží obsazenost paměti a tím se zvýší rychlost práce mikroprocesoru. Tento problém nastává při velké rychlosti pojezdu robotického vysavače. Rychlost vysavače byla snížena jak kvůli zmíněnému problému, tak i kvůli mechanickému opotřebením při nárazu nárazníkem do překážky.

Při porovnání zkonstruovaného vysavače a vysavače na trhu bylo zjištěno, že sestrojený vysavač projíždí prostorem velmi dobře, i když nemá algoritmy na mapování a plánování trasy. Cestu si volí na základě pevně stanoveného algoritmu, který je určen podle nastalé situace. Dále bylo pozorováno, že sestrojený robotický vysavač má lepší mechanismu pro detekci překážek.

Při pokusu byl jako znázorňující odpad použit papírový kapesník. Příhodnější by bylo asi použít věc, která nemá snahu uletět, protože vzduch vycházející z ventilátoru pohyboval přiložené kapesníky.

## Literární zdroje

- [1] ČSN 33 2130 ED.3.,2015 *Elektrické instalace nízkého napětí – Vnitřní elektrické rozvody. 332130. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví*
- [2] VTM, 2018. Průvodce džunglí robotických vysavačů . Jak vybrat ten správný? [online]. [cit. 2024-02-27]. Dostupné z: <https://vtm.zive.cz/clanky/pruvodce-dzungli-robotickych-vysavacu-jak-vybrat-ten-spravny/sc-870-a-195617/default.aspx#part=10>KILIÁN, Karel, 2018. Průvodce džunglí robotických vysavačů . Jak vybrat ten správný? *VTM*, č. 32, s. 1-10. ISSN 0322-9017
- [3] AUTODESK, *Fusion 360* [online]. [cit. 2024-02-27]. Dostupné z: <https://www.fusion360.cz/>
- [4] ARDUINO, *Arduino Reference* [online]. [cit. 2024-02-27]. Dostupné z: <https://docs.arduino.cc/hardware/uno-rev3/>
- [5] ARDUINO, *Product Reference Manual* [online]. [cit. 2024-02-27]. Dostupné z: <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>
- [6] LASKAKIT, *A4988 DVR8825 Adapter*. [online]. [cit. 2024-02-29]. Dostupné z: <https://www.laskakit.cz/a4988-drv8825-adapter/>
- [7] LASKAKIT, *DVR8825 driver pro krokové motory*. [online]. [cit. 2024-02-29]. Dostupné z: <https://www.laskakit.cz/drv8825-driver-pro-krokovye-motory/>
- [8] POLOLU, *DVR8825 Stepper Motor Driver Carrier, High Current*. [online]. [cit. 2024-02-29]. Dostupné z: <https://www.pololu.com/product/2133>
- [9] LASKAKIT, *TT motor s převodovkou – plastové motory*. [online]. [cit. 2024-02-29]. Dostupné z: <https://www.laskakit.cz/tt-motor-s-prevodovkou-plastove-prevody/>
- [10] RAVEO, *Jak funguje servomotor a zpětnovazební řízení*. [online]. [cit. 2024-02-29]. Dostupné z: <https://www.raveo.cz/aktualita/jak-funguje-servomotor-a-zpetnovazebni-rizeni/>
- [11] EUFACTORY, *Krokový motor – jak vybrat ten správný*. [online]. [cit. 2024-02-29]. Dostupné z: [https://shop.eufactory.com/blog/13\\_krokovy-motor-vyber](https://shop.eufactory.com/blog/13_krokovy-motor-vyber)
- [12] STOMISI, *Jak se řídí krokové motory*. [online]. [cit. 2024-02-29]. Dostupné z: <https://stomisi.cz/jak-se-ridi-krokovye-motory.html>
- [13] JANOVSKEÝ, Jan, 2018. *Koncepční návrh laserového snímače vzdálenosti s webovou kamerou pro detekci překážek v dráze robota*. Praha, Bakalářská práce. České vysoké učení technické v Praze, Fakulta strojní, Ústav přístrojové a řídicí techniky
- [14] AMS datasheet, 29. June 2020. *TMF Time-of-Flight Sensor*. [online]. [cit. 2024-02-27]. Dostupné z: <https://dfimg.dfrobot.com/nobody/wiki/2e8974e9dd0008f278e3d1031d289170.pdf>

- [15] Bc. HAAS, Filip, 2019. *Návrh podvozku pro zařízení určené na mobilní vsakovací zkoušky*. Praha, Diplomová práce. České vysoké učení technické v Praze, Fakulta strojní, Dopravní, letadlová a transportní technika
- [16] NELVSON, S., WIBISONO, F. A., ANGGADJAJA, E., ROCHADIANI, T., 2018. Food Products Monitoring Machine Using Combinations of Multiple Autonomous Sensors 2018: *10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Bali, 24.-26. July 2018: Institute of Electrical and Electronics Engineers (IEEE). [cit. 2024-02-27]. Dostupný z WWW: [https://www.researchgate.net/publication/328993022\\_Food\\_Products\\_Monitoring\\_Machine\\_Using\\_Combinations\\_of\\_Multiple\\_Autonomous\\_Sensors](https://www.researchgate.net/publication/328993022_Food_Products_Monitoring_Machine_Using_Combinations_of_Multiple_Autonomous_Sensors)
- [17] Ing. KUPKA Ph.D., Libor, 2023. *Mechatronika a robotika – Mapy prostředí a plánování dráhy*. Pardubice. Přednáškové materiály. Univerzita Pardubice, Fakulta elektrotechniky a informatiky
- [18] Bc. HAAS, Filip, 2019. *Návrh podvozku pro zařízení určené na mobilní vsakovací zkoušky*. Praha, Diplomová práce. České vysoké učení technické v Praze, Fakulta strojní, Dopravní, letadlová a transportní technika
- [19] LIM, Henry, August 31, 2011. *Introduction to SLAM (Simultaneous Localization and Mapping)*. [online]. [cit. 2024-02-27]. Dostupné z: <https://ouster.com/insights/blog/introduction-to-slam-simultaneous-localization-and-mapping>
- [20] PragoBoard, *Proudová zatížitelnost vodiče* [online]. [cit. 2024-02-27]. Dostupné z: <https://www.pcb.cz/node/34>
- [21] DRATEK, *Songle relay* [online]. [cit. 2024-02-27]. Dostupné z: [https://dratek.cz/docs/produkty/0/919/songle\\_srd.pdf](https://dratek.cz/docs/produkty/0/919/songle_srd.pdf)
- [22] TME, *44.52.9.060.0000 Finder* [online]. [cit. 2024-02-27]. Dostupné z: [https://www.tme.eu/cz/details/44.52.9.060.0000/elektromagneticka-rele-miniaturni/finder/?brutto=1&currency=CZK&gad\\_source=1&gclid=CjwKCAiA0Pu uBhBsEiwAS7fsNddPPXQSEfDxVuKrdaPP-j8ZFrO9lqokDAJTkeGlGYJNFLH0ceVvCRoCbJcQAvD\\_BwE](https://www.tme.eu/cz/details/44.52.9.060.0000/elektromagneticka-rele-miniaturni/finder/?brutto=1&currency=CZK&gad_source=1&gclid=CjwKCAiA0Pu uBhBsEiwAS7fsNddPPXQSEfDxVuKrdaPP-j8ZFrO9lqokDAJTkeGlGYJNFLH0ceVvCRoCbJcQAvD_BwE)

## **Přílohy k bakalářské práci**

Přílohy k bakalářské byly vloženy do STAGU

Seznam příloh:

- Program v programovém prostředí Arduino IDE
- Schéma a board desky v návrhovém prostředí Eagle
- Schéma robotického vysavače v návrhovém prostředí Fusion 360.
- Samostatný text bakalářské práce v elektronické podobě