

**UNIVERZITA PARDUBICE**  
Fakulta elektrotechniky a informatiky

**MOBILNÍ ROBOT VYKONÁVAJÍCÍ POSLOUPNOST  
POHYBOVÝCH PŘÍKAZŮ**

Lukáš Říha

Bakalářská práce

2023

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2021/2022

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Lukáš Říha**  
Osobní číslo: **I18088**  
Studijní program: **B2612 Elektrotechnika a informatika**  
Studijní obor: **Řízení procesů**  
Téma práce: **Mobilní robot vykonávající posloupnost pohybových příkazů**  
Zadávací katedra: **Katedra řízení procesů**

## Zásady pro vypracování

Cílem práce je sestavení jednoduchého mobilního robota s dvěma koly a příslušného řídicího systému, který bude vykonávat zaslou sekvenční příkazy, kde každý příkaz bude obsahovat informaci o translační a úhlové rychlosti pohybu a době pohybu. Příkazy pohybu bude možné zřetěžit do jediné trajektorie, která bude zaručovat spojitý průběh rychlosti. Bude využito dostupných komponent a vývojových modulů, jako např. Arduino, a příslušných periférií.

Rozsah pracovní zprávy: **min. 30 stran**  
Rozsah grafických prací:  
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

NOVÁK, P. *Mobilní roboty 1 – Pohony, senzory, řízení*. BEN, 2004.  
SICILIANO, B., KHATIB, O. (Eds.). *Springer Handbook of Robotics*. Springer-Verlag, 2008.  
CVEJN, J. *Průmyslové roboty* [online]. Pardubice: Univerzita Pardubice, FEI, 2021. Elektronický studijní materiál k předmětu Průmyslové roboty.

Vedoucí bakalářské práce: **doc. Ing. Jan Cvejn, Ph.D.**  
Katedra řízení procesů

Datum zadání bakalářské práce: **17. prosince 2021**  
Termín odevzdání bakalářské práce: **13. května 2022**

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

L.S.

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

V Pardubicích dne 7. ledna 2022

## **Prohlášení**

Prohlašuji:

Tuto práci 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., autorský zákon, 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. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 17.5.2023

Lukáš Říha

## **Poděkování**

Rád bych poděkoval mému vedoucímu doc. Ing. Janu Cvejnovi, Ph.D. za jeho cenné rady a připomínky, které mi pomohli při tvorbě této bakalářské práce. Dále bych rád poděkoval své rodině. Zvláště pak své sestře za morální podporu během doby mého studia.

V Pardubicích dne 17. 5. 2023

Říha Lukáš

## **ANOTACE**

*Cílem práce bylo sestrojít dvoukolový robot s příslušným řídicím systémem, který vykonává sekvenci pohybových příkazů. Každý příkaz zadává translační rychlost, úhel natočení, směr a dobu pohybu. Dále tyto funkce bude možné zřetězit do jedné trajektorie. Sestrojené zařízení se skládá z vývojového modulu Arduino Uno, dvou stejnosměrných motorů a jedním kolem volně, driveru DC motorů L293D a držáku baterií. Řídicí program je vytvořen v prostředí Arduino.*

## **KLÍČOVÁ SLOVA**

*mobilní robot, Arduino, plánování trajektorie, pohony robotů, kinematika dvoukolového robota.*

## **TITLE**

*MOBILE ROBOT EXECUTING A SEQUENCE OF MOVEMENT COMMANDS*

## **ANNOTATION**

*The aim of the work was to build a two-wheeled robot with the appropriate control system, which executes a sequence of movement commands. Each command assigns the translational velocity, steering angle, direction and time of motion. These commands can be chained into single trajectory. Constructed device consists of an Arduino Uno development module, two DC motors and one free wheel, the L293D DC motor driver and a battery holder. The control program is created in the Arduino environment.*

## **KEYWORDS**

*mobile robot, Arduino, trajectory planning, robot actuators, kinematics of a two-wheeled robot.*

# OBSAH

|  |    |
|--|----|
| Seznam zkratk a značek .....                       | 9  |
| Seznam symbolů veličin a funkcí .....              | 10 |
| Seznam ilustrací .....                             | 11 |
| Seznam tabulek .....                               | 12 |
| ÚVOD .....   | 13 |
| 1 ROBOTIKA .....                                   | 14 |
| 1.1 HISTORIE ROBOTIKY .....                        | 14 |
| 1.2 DĚLENÍ MOBILNÍCH ROBOTŮ PODLE KOSTRUKCE .....  | 15 |
| 1.3 AUTONOMNOST MOBILNÍCH ROBOTŮ .....             | 16 |
| 2 KOMPONENTY MOBILNÍCH ROBOTŮ .....                | 18 |
| 2.1 POHONNÝ SUBSYSTÉM ROBOTŮ .....                 | 18 |
| 2.1.1 Výkonové zesilovače .....                    | 18 |
| 2.1.2 Motory .....                                 | 19 |
| 2.1.3 Stejnoseměrný motor .....                    | 19 |
| 2.1.4 Krokový motor .....                          | 20 |
| 2.1.5 Převodovka .....                             | 22 |
| 2.2 SENZORY MOBILNÍCH ROBOTŮ .....                 | 23 |
| 2.2.1 Taktilní senzory .....                       | 23 |
| 2.2.2 Infračervené senzory .....                   | 23 |
| 2.2.3 Ultrazvukové senzory .....                   | 24 |
| 2.3 PRVKY POHYBOVÉHO APARÁTU .....                 | 25 |
| 2.3.1 Kolové podvozky .....                        | 25 |
| 2.3.2 Pásové podvozky .....                        | 26 |
| 2.3.3 Kráčeující podvozek .....                    | 26 |
| 3 Kinematika Dvoukolového robota .....             | 27 |
| 4 PLÁNOVÁNÍ TRAJEKTORIE .....                      | 29 |
| 4.1 NÁVRH TRAJEKTORIE METODOU POINT-TO-POINT ..... | 29 |
| 5 ZPŮSOBY KOMUNIKACE MOBILNÍCH ROBOTŮ .....        | 31 |
| 5.1 KOMUNIKACE PŘES SÉRIOVÝ PORT .....             | 31 |
| 5.2 IR OVLÁDÁNÍ .....                              | 31 |
| 5.3 BLUETOOTH .....                                | 32 |

|       |   |    |
|-------|---|----|
| 6     | Praktická část práce.....                   | 33 |
| 6.1   | ROBOTICKÁ PLATFORMA .....                   | 33 |
| 6.1.1 | Mechanická konstrukce mobilního robota..... | 33 |
| 6.1.2 | Klon Arduino Uno R3.....                    | 35 |
| 6.1.3 | Motor driver shield L293D .....             | 36 |
| 6.1.4 | Vývojové prostředí Arduino IDE.....         | 39 |
| 6.1.5 | Zapojení komponent .....                    | 40 |
| 6.2   | Ovládání robota.....                        | 44 |
| 6.2.1 | Programové řešení řízení robota .....       | 45 |
| 6.2.2 | Struktura řídicího systému .....            | 48 |
| 6.3   | EXPERIMENTY S ROBOTEM.....                  | 53 |
| 7     | ZÁVĚR .....                                 | 58 |
|       | POUŽITÁ LITERATURA .....                    | 59 |
|       | PŘÍLOHY .....                               | 61 |



## SEZNAM ZKRATEK A ZNAČEK

|        |   |
|--------|---|
| S.R.I. | Stanfordský výzkumný ústav                              |
| M.I.T. | Massachusettský technologický ústav                     |
| DC     | stejnoseměrný   |
| AC     | střídavý  |
| PWM    | pulzně šířková modulace                                 |
| IR     | infračervený  |
| LED    | světlo emitující dioda                                  |
| Wi-Fi  | bezdrátový přenos                                       |
| USB    | universální sériová sběrnice                            |
| SD     | zabezpečený digitální                                   |
| PC     | osobní počítač  |
| FLASH  | elektricky programovatelná paměť s libovolným přístupem |
| IDE    | integrované vývojové prostředí                          |
| A/D    | analogový signál/digitální signál                       |
| ISR    | integrovaný spínací regulátor                           |

## SEZNAM SYMBOLŮ VELIČIN A FUNKCÍ

|            |   |
|------------|---|
| M          | moment ozubených kol, Nm                            |
| $\omega$   | rychlost otáčení ozubených kol, otmin <sup>-1</sup> |
| n          | počet ozubení kol                                   |
| $v_k$      | rychlost kola, ms <sup>-1</sup>                     |
| $\omega_k$ | úhlová rychlost kola, rads <sup>-1</sup>            |
| $r_k$      | poloměr kola, m                                     |
| v          | rychlost robota, ms <sup>-1</sup>                   |
| $v_1$      | rychlost jednoho kola                               |
| $v_2$      | rychlost druhého kola                               |
| $\omega_l$ | úhlová rychlost levého kola, rads <sup>-1</sup>     |
| $\omega_r$ | úhlová rychlost pravého kola, rads <sup>-1</sup>    |
| r          | poloměr kol, m                                      |
| R          | vzdálenost osy otáčení                              |
| d          | průměr vzdálenosti kol                              |
| l          | délka, m  |
| $a_i$      | konstanta polynomu                                  |
| q          | referenční trajektorie                              |
| t          | čas, s  |
| $v_{max}$  | konstantní rychlost pohybu, ms <sup>-1</sup>        |
| a          | zrychlení, ms <sup>-1</sup>                         |

## SEZNAM ILUSTRACÍ

|   |    |
|---|----|
| Obrázek 1.1 – Rozdělení mobilních robotů .....  | 15 |
| Obrázek 1.2 – Dělení mobilních robotů podle autonomnosti .....                        | 16 |
| Obrázek 2.1 – Schéma stejnosměrného motoru s permanentním magnetem .....              | 20 |
| Obrázek 2.2 – Schéma krokového motoru.....  | 20 |
| Obrázek 2.3 – Řez krokového motoru s pasivním rotorem .....                           | 21 |
| Obrázek 2.4 – Infračervený senzor vzdálenosti GPY0A02YK0F .....                       | 24 |
| Obrázek 2.5 – Ultrazvukový senzor vzdálenosti HC-SR04 .....                           | 25 |
| Obrázek 3.1 – Ukázka robota v kartézské soustavě souřadnic .....                      | 27 |
| Obrázek 4.1 – Profil rychlosti s lineárním náběhem .....                              | 30 |
| Obrázek 5.1 – Bezdrátové ovládání pomocí IR ovladače.....                             | 32 |
| Obrázek 5.2 – UART Bluetooth modul .....  | 32 |
| Obrázek 6.1 – Schéma desky mobilního robota .....                                     | 34 |
| Obrázek 6.3 – Rejdovací kolečko .....   | 35 |
| Obrázek 6.2 – Kolo s DC motorem .....   | 35 |
| Obrázek 6.7 – Klon Arduino Uno R3 .....   | 36 |
| Obrázek 6.4 – Motor driver shield L293D .....   | 37 |
| Obrázek 6.5 – Schéma H-můstku .....   | 38 |
| Obrázek 6.6 – Ukázka nastavení pinů a základního nastavení v knihovně AFMotor.h ..... | 39 |
| Obrázek 6.8 – Vývojové prostředí Arduino.....   | 40 |
| Obrázek 6.10. zapojení mobilního robota zezdola .....                                 | 41 |
| Obrázek 6.9. zapojení mobilního robota seshora.....                                   | 41 |
| Obrázek 6.11. Schéma zapojení mobilního robota .....                                  | 43 |
| Obrázek 6.12. Graf úhlu v závislosti na čase.....                                     | 46 |
| Obrázek 6.13. Pokus č. 1 .....  | 54 |
| Obrázek 6.15. Pokus č. 2 .....  | 55 |
| Obrázek 6.15. Pokus č. 3 .....  | 56 |

## SEZNAM TABULEK

|   |    |
|---|----|
| Tabulka 6.1 – Parametry dvoukolového robota.....              | 34 |
| Tabulka 6.2 – Vstupy a výstupy driveru L293D .....            | 42 |
| Tabulka 6.3 – Parametry pro příkazový řádek.....              | 44 |
| Tabulka 6.4 – Ujetá vzdálenost mobilního robota na čase ..... | 45 |
| Tabulka 6.5 – Úhel v závislosti na čase.....                  | 46 |

## ÚVOD

Mobilní robotika nachází uplatnění v řadě oblastí. Setkáme se s ní třeba v domácnosti, jako je například mobilní robotický vysavač a v různém odvětví průmyslu, kde pomáhá vyrábět různé výrobky. Pro práci s těmito roboty je potřeba zkušených operátorů k jejich manipulaci nebo programování.

Cílem této práce je vytvoření jednoduchého dvoukolového mobilního robota, který bude schopen vykonávat určité povely, které se mu zasílají. Robot, na základě jeho naprogramování, pracuje s translační rychlostí, úhlem natočení a dobou pohybu. Má být také možné tyto pokyny zřetěžit do jedné trajektorie tak, aby mobilní robot byl schopen vykonat více pohybů za sebou.

Součástí řešení je i posouzení, zda tento robot bude schopen trajektorii vykonat a jestli odpovídá jeho pohyb zasílaným parametrům.

# 1 ROBOTIKA

Robotika je technický a vědecký obor, který má za úkol pomoci člověku řešit problémy s usnadněním jeho práce a zvyšovat produktivitu. Čerpá z umělé inteligence, ale i měřicí techniky, elektrotechniky, mechaniky a dalších vědních a technických disciplín. Vyžaduje vzdělání s širší perspektivou v daných oborech (Novák, 2005).

## 1.1 HISTORIE ROBOTIKY

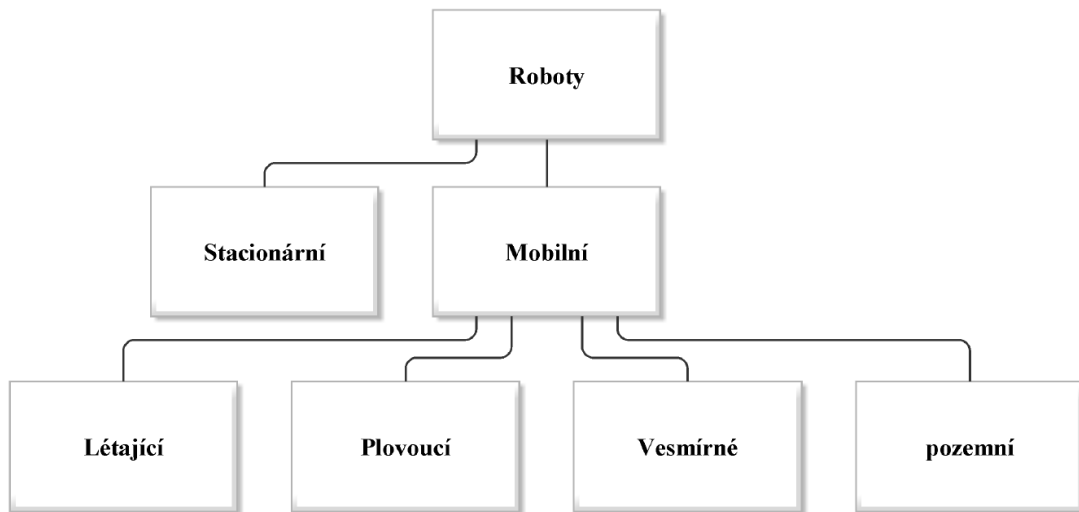
Po celá staletí bylo lidstvo nuceno provozovat celoživotní těžkou a namáhavou práci. Po náročné práci mohla být pro lidi útěcha fantazie, ve které si mohli vysnit stroje, které by za ně práci řešili automaticky. U většiny případů snaha o automatickém vykonávání práce směřovala ke konstrukci automatických zařízení, podobajících se lidem či zvířatům. Tato vize zřejmě bude lidstvo provázet ještě několik staletí. (Maršík, Kubičík, 1980).

V 2. polovině 20. stol. Došlo k značnému rozvoji praktických aplikací robotiky zejména v USA. Roku 1961 začíná provoz průmyslového robota UNIMATE z firmy General Motors. Rok 1964 byl rokem otevírání laboratoří zabývajících se umělou inteligencí na Stanford Research Institute (S.R.I.), Massachusetts Institute of Technology (M.I.T) a dalších institucí v USA.

V roce 1968 byl vytvořen v S.R.I. robot Shakey, který byl vybavený viděním a mohl se do určité míry orientovat v prostředí. V 70. letech modely robotů vstoupily do sériové výroby a našly uplatnění hlavně v automobilové výrobě. V 80. letech 20. století se do vedení s využitím robotů dostává Japonsko. V této době jsou průmyslové roboty vybavovány čidly hmatu a počítačovým viděním v oblastech výzkumu.

K robotice patří i výzkum dálkového řízení, kde operátor dostává všechny informace o průběhu robota, jeho souřadnice a prostředí, ve kterém se robot nachází. Operátor by měl uzpůsobovat dálkové ovládání tak, jako by byl na pozici robota. Takto řízený robot by pak mohl pomáhat jak záchranářům, tak i hasičům.

## 1.2 DĚLENÍ MOBILNÍCH ROBOTŮ PODLE KOSTRUKCE



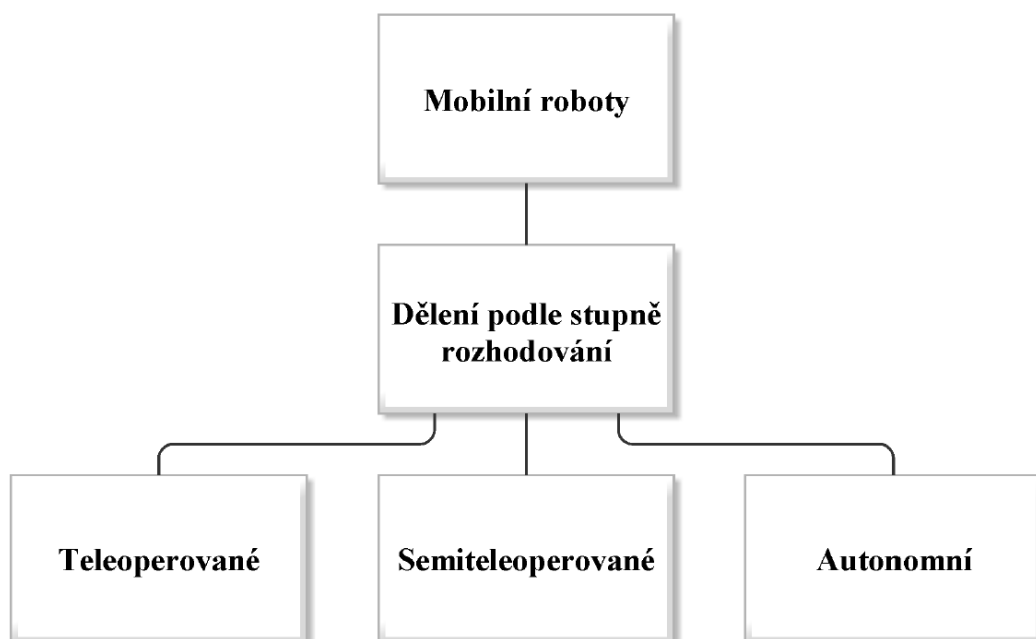
Obrázek 1.1 – Rozdělení mobilních robotů

Hlavní dělení robotů se dělí na dvě kategorie, a to stacionární a mobilní. Tato práce se dále bude věnovat pouze mobilním robotům. (Buchli, 2006)

Každý robot musí být navržen tak, aby odpovídal druhu použití, rychlosti a prostředí s pevnými nebo pohyblivými překážkami. Existují 4 hlavní kategorie:

- 1) Létající – robotické drony, letadla, helikoptéry, robotické ovládané padáky, či vzducholodě.
- 2) Plovoucí – operují na hladině, nebo pod ní. Využívají k tomu vodní trysky, nebo vrtule.
- 3) Vesmírné – jsou navrženy tak, aby mohli operovat v mikrogravitaci vesmíru. Běžně jsou používány na údržbu vesmírných stanic. Vesmírní roboti se pohybují buď šplhem, nebo nezávislým pohonem.
- 4) Pozemní – nejčastější druh robota. V této kategorii nalezneme roboty kolové, pásové, nebo kráčející.

### 1.3 AUTONOMNOST MOBILNÍCH ROBOTŮ



Obrázek 1.2 – Dělení mobilních robotů podle autonomnosti

Autonomnost lze chápat jako samostatnost. Mobilní roboty se dělí podle míry autonomního rozhodování do tří základních skupin:

- 1) Teleoperované (neautonomní) – roboty jsou plně ovládaný operátorem. Rozhodují se, jaké činnosti mají vykonávat na základě řídicích povelů ze strany operátora. Sám robot není schopen jakéhokoliv řízení.
- 2) Semiteleoperované (polo-autonomní) – roboty dostávají jen dílčí cíle od operátora. Jsou schopny vykonávat činnosti jen z části samostatně, z tohoto důvodu je robot nucen přijímat operátorské povely.
- 3) Autonomní – roboty jsou programovány tak, aby provedly určitý úkol bez potřeby zásahu operátora. Roboty se umějí rozhodnout, zda zvládnou a jsou schopny plnit činnosti. Aby autonomní roboty mohly samostatně řešit úlohy, tak musejí mít ve výbavě senzorický řídicí systém. (Havel, 1980).

Robot označený jako autonomní by se měl pohybovat samostatně, sám by měl dlouhodobě vykonávat aktivity a získávat údaje o pracovním prostředí bez nutnosti zásahu



programátora a měl by být naprogramován tak, aby nemohlo dojít ke zranění jeho obsluhy, nebo jeho destrukce.

## 2 KOMPONENTY MOBILNÍCH ROBOTŮ

Mobilní roboty se skládají z hardwarových a softwarových komponent, které jsou propojené, komunikují a spolupracují mezi sebou. Činnosti a struktury těchto subsystémů spočívá ve specifických úkonech, jež mobilní robot vykonává. Subsystémy mobilních robotů dělíme na:

- Pohonný subsystém – motory.
- Senzorický subsystém – taktilní, infračervené, ultrazvukové.
- Pohybový subsystém – kola, pásy, kráčedla.
- Řídicí (kognitivní) subsystém – Arduino.
- Komunikační subsystém – Bluetooth, Wi-Fi, IR ovládání.

### 2.1 POHONNÝ SUBSYSTÉM ROBOTŮ

Součástí pohybového subsystému je pohonný subsystém robota. K pohonu mobilních robotů je nejčastěji používáno rotačních elektromotorů. U elektromotorů z poměru výkonu a hmotnosti se nejvíce vyplatí stejnosměrné a střídavé motory. V oblasti školních mobilních robotů je nejvýhodnější používat krokové motory z důvodu nízké ceny a snadného ovládání (není zapotřebí převodovka). Systém pohonu se skládá z přívodu energie, výkonového zesilovače, motoru a převodovky. (Novák, 2005).

#### 2.1.1 Výkonové zesilovače

Princip výkonového zesilovače je převod řídicího signálu na veličinu pohánějící motor. U elektrických pohonů je přívod energie tvořen většinou můstkovým usměrňovačem a transformátorem. U DC motorů s permanentními magnety je výstupem stejnosměrné napětí, kdežto u bezkartáčových DC motorů je tímto výstupem střídavé napětí. Pro požadovaný rozsah výkonů se používají PWM zesilovače. U DC motorů s permanentními magnety se používá konverze typu DC-to-DC, tzv. choppers. Pro bezkartáčové DC motory se využívá převodu DC-to-AC, tzv. invertory. (Cvejn, 2021)

## 2.1.2 Motory

Typy motorů se dělí na:

- 1) Elektrické – převádí elektrickou energii na mechanickou práci.
- 2) Hydraulické – přetvářejí hydraulickou energii na mechanickou práci.
- 3) Pneumatické – transformují energii stlačeného vzduchu na mechanickou energii turbín, nebo pístů

V robotice musí mít motory vysoký poměr výkon/hmotnost. Dále musí mít schopnost vysokých zrychlení a momentálního přetížení. Důležitý požadavek je i na přesnost polohy a široký rozsah rychlosti (1-1000 ot/min).

Do elektrických motorů řadíme stejnosměrné motory s permanentními magnety, bezkartáčové stejnosměrné motory a krokové motory. (Cvejn, 2021).

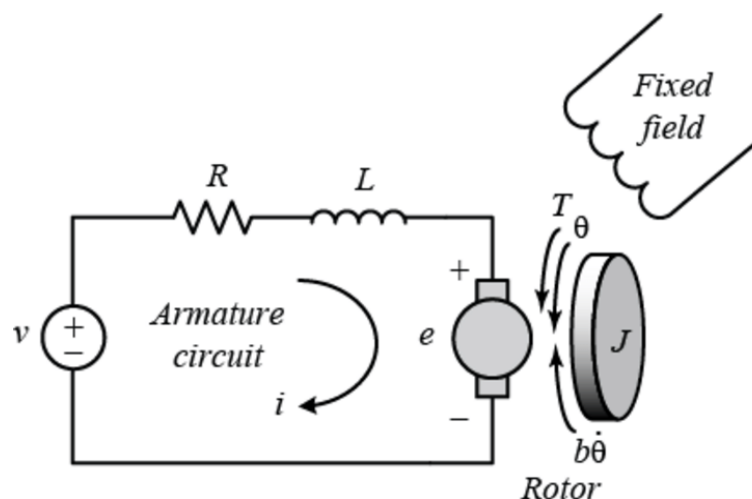
## 2.1.3 Stejnosměrný motor

Hlavní části stejnosměrného motoru tvoří rotor a stator. Díky komutátoru se polarita elektromagnetu rotoru mění dvakrát za jednu otáčku a když se souhlasné póly magnetů odpuzují, je rotor nucen k pohybu. (Novák, 2005).

Rozdíl mezi stejnosměrnými motory s permanentními magnety (schéma na obr. 2.1) a bezkartáčovými stejnosměrnými motory je v tom, že u DC motorů s permanentními magnety je stator z feromagnetických keramických materiálů a rotor s vinutím, kdežto u bezkartáčových DC motorů je rotor z permanentního magnetu a stator obsahuje vinutí pevně připojené k napájení. (Cvejn, 2021).

Výhodami stejnosměrného motoru je dobrý poměr hmotnosti a výkonu, dále pak dostupnost a široká nabídka. Další výhodou je lineární charakteristika závislosti otáček na budícím napětí rotoru. Jednoduchost je i v řízení směru otáček stejnosměrného motoru, kde se změna směru otáčení dosáhne změnou polarit napájecího napětí. (Novák, 2005).

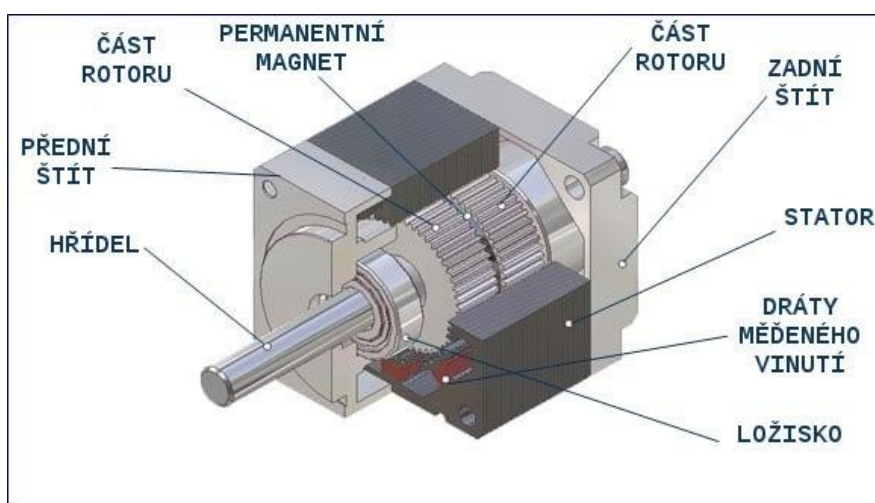
Nevýhodami u stejnosměrného motoru je složitější polohové a rychlostí řízení. Stejnosměrný motor pracuje na nízkém momentu a vysokých otáčkách, což je pro potřebu pohonů nevýhodné. Tento problém je potřeba řešit použitím mechanické převodovky.



Obrázek 2.1 – Schéma stejnosměrného motoru s permanentním magnetem  
(ResearchGate.net, 2017)

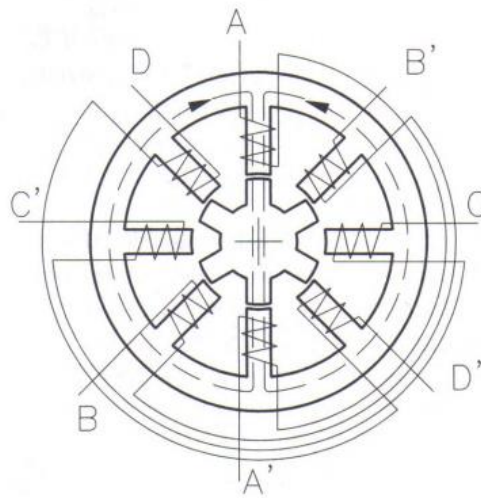
## 2.1.4 Krokový motor

Oproti stejnosměrného motoru je u krokového motoru (schéma na obr. 2.2) napájen stator. Rotor je buď z měkkého, nebo tvrdého feromagnetického materiálu. Motor je poháněn vhodnou posloupností poháněcích impulsů. Magnetické pole je generováno postupným napájením jednotlivých pólových dvojic. Krokové motory můžeme rozdělit na 3 skupiny. Krokový motor s pasivním rotorem, hybridní krokový motor a lineární krokový motor. (Novák, 2005).



Obrázek 2.2 – Schéma krokového motoru (eluc ikap.cz, nedatováno)

- 1) Krokový motor s pasivním rotorem – též označován jako reakční nebo reluktanční. Rotor je tvořen s feromagneticky měkkého materiálu a může být vytvořen jedním kusem ocele nebo svazkem ocelových plechů. Stator je tvořen (jako u rotoru) svazkem plechů s osmi vyniklými pólovými nástavci, která nesou vnutit jednotlivých fází. Na statoru se postupně spínají jednotlivé dvojice cívek, tím je tvořeno točivé magnetické pole, čímž je rotor tímto polem přitahován a začne se točit. Na obrázku 2.3 je vidět řez krokového motoru s pasivním rotorem



Obrázek 2.3 – Řez krokového motoru s pasivním rotorem (Novák, 2005)

- 2) Hybridní krokový motor – nejpoužívanější typ v průmyslu. Rotor se skládá s permanentních magnetů, které jsou namontovány axiálně, díky čemu mají pólové nástavce různou magnetickou polaritu. Pro provoz hybridního motoru je hlavní, aby zuby pólových nástavců byly navzájem natočeny tak, aby osy drážek jedno nástavce byly proti osám zubů druhého nástavce. U statoru jsou též na pólových nástavcích vytvořeny zuby. Spínáním cívek statoru se pak pohybuje celý motor.
- 3) Lineární krokový motor – zvláštní typ krokového motoru, který je díky své konstrukci schopen poskytovat lineární pohyb. Tento typ nachází uplatnění ve velkém množství strojů. Najdeme ho i v některých tiskárnách, plotrech, nebo u obráběcích strojů. Rotor není státorem obklopen. V tomto případě je stator rovný pás se zuby. Rotor je složen minimálně ze dvou střídavě napájených cívek a pomocí vytvářených magnetických sil se nad tímto pásem pohybuje.

### 2.1.5 Převodovka

Je nutností provést převod velkých otáček, které dodává motor na nižší otáčky a zároveň vyšší momenty na hnací hřídeli. Dále může být převod z rotačního pohybu na lineární. U převodovky se ale vždy musíme potýkat s určitými problémy, jako je pružnost převodových částí, nebo s vůlí (např. mezi zuby ozubených kol). Díky těmto problémům dochází k zhoršení mechanických vlastností pohonu. Současně dochází ke ztrátě výkonu, kvůli tření v převodech. Výhodou ale je, že z pohledu řízení částečně potlačují nelinearity dynamiky robota. Převodový poměr bývá z pravidla v rozsahu 1:20-1:200. (Cvejn, 2021).

Pro navýšení převodového poměru se může použít více převodových stupňů, kde výsledný převod násobíme. Tímto řešením ale dochází k zhoršení mechanických vlastností. V ideálním případě při převodu platí, že:

$$M_1\omega_1 = M_2\omega_2, \quad (2.1)$$

kde  $M_1$  – moment hnacího ozubeného kola, Nm,

$M_2$  – moment hnaného ozubeného kola, Nm

$\omega_1$  – rychlost otáčení hnacího ozubeného kola, otmin<sup>-1</sup>

$\omega_2$  – rychlost otáčení hnaného ozubeného kola, otmin<sup>-1</sup>.

Cílem u ozubených převodovek je dosažení nejvyššího převodového poměru při nejmenších rozměrech. Ozubené převodovky bývají většinou planetové, nebo harmonické. Pro základní typ ozubené převodovky platí:

$$\frac{\omega_2}{\omega_1} = \frac{n_1}{n_2}, \quad (2.2)$$

kde  $n_1$  – počet ozubení hnacího kola.

$n_2$  – počet ozubení hnaného kola

Tímto obvykle nedosáhneme dostatečně vysokého převodu.

U harmonické převodovky nalezneme 2 ozubené prstence, kde vnější je pevný a vnitřní je pružný a také má menší počet zubů než vnější. Pružný prsteneček je z vnitřní strany

přítlačován k vnějšímu prstenci hnacím kolem a při pohybu dochází k posunutí. Převodový poměr je pak:

$$\frac{\omega_2}{\omega_1} = \frac{n_1 - n_2}{n_1}. \quad (2.3)$$

Harmonické převodovky mají vysoký převodový poměr a jsou pružné.

## 2.2 SENZORY MOBILNÍCH ROBOTŮ

### 2.2.1 Taktilní senzory

Tento typ senzoru je nejjednoduššího provedení, který se realizuje kontaktním spínačem. Při zapnutí spínače dochází k sepnutí elektrického obvodu a změně logické úrovně. Toto připojení senzoru potřebuje napojení všech spínačů odděleně. Při aktivaci taktilního senzoru je na výstupu nízká logická úroveň. Pokud není spínač aktivován, tak je na příslušném výstupu tato logická úroveň vysoká. (Novák, 2005).

Spínače u taktilních senzorů lze také při omezeném počtu binárních vstupů připojit přes multiplexer. Jednotlivé senzory pak testujeme postupným adresováním jeho vstupů. Tímto zapojením můžeme rozlišovat současné aktivování více senzorů.

Mezi taktilní senzory patří i tenzometry, nebo senzory založené na piezoelektrickém jevu. Tyto typy zajišťují spojitý signál. Tenzometry se dají osadit nohou kráčejícího robota a lze detekovat její zatížení, nebo kolizi s danou překážkou.

### 2.2.2 Infračervené senzory

IR senzor (zobrazený na obr. 2.4) se používá pro detekování překážek v okolí robota. Tento typ senzoru je necitlivý na infračervené světelné záření jiných vlnových délek, ale v oblasti vlnových délek pod viditelným světlem ano. Princip IR senzorů je detekce odraženého infračerveného světla od určité překážky. Toto světlo je emitováno IR LED Diodou. Jako detektor se používá fototranzistor, který je citlivý v infračervené oblasti. Infračervený senzor poskytuje dvouhodnotový signál, což znamená, že buď detekuje

odražený IR signál, nebo nedetekuje odražený signál. Negativa u IR senzorů pracujících na principu detekce odraženého infračerveného světla je, že množství odraženého světla musí být závislé na barvě překážky. (Novák, 2005).

Ke konstrukci IR senzorů je potřeba několika součástek. Jedná se o IR LED vysílač, detektor a odpory. Podle specifikace detektoru je zapotřebí dále modulovaný signál ještě modulovat nižším kmitočtem kvůli správné funkčnosti senzoru. Perioda modulace bývá většinou větší než 1 ms. Senzor založený na tomto principu je vhodný k detekci blízkého okolí robota.



Obrázek 2.4 – Infračervený senzor vzdálenosti  
GPY0A02YK0F (Drátek.cz, 2022)

### 2.2.3 Ultrazvukové senzory

Dalším typem senzorů, které se využívají v robotice je ultrazvukový senzor vyobrazený na obrázku (2.5). Využívají zvuk o frekvenci od 29kHz až do 43kHz. Tento typ senzorů je tvořen přijímací a vysílací částí. Jsou vhodné pro měření objektů s mnohotvárným povrchem. Mají sice kratší dosah, ale hodí se pro průmyslové aplikace. Ultrazvukové senzory jsou schopny přesných měření výsledků vzdálenosti a často jsou využívány pro parkovací senzory. (Novák, 2005).





Obrázek 2.5 – Ultrazvukový senzor vzdálenosti HC-SR04  
(laskakit.cz, 2022)

## 2.3 PRVKY POHYBOVÉHO APARÁTU

### 2.3.1 Kolové podvozky

U kolových podvozků se musí počítat s uspořádáním a počtem kol, aby byl podvozek stabilní. Kolové podvozky se mohou dělit na stabilní a nestabilní. Do nestabilních podvozků patří jednokolové a dvoukolové podvozky a do stabilních patří tříkolové a vícekolové. (Kolíbal, 2016).

- 1) Jednokolové a dvoukolové podvozky – tento typ kolového podvozku je dynamicky stabilní a staticky nestabilní. Pohyb robotu s tímto podvozkem se realizuje naklopením v případě jednokolového provedení a změnou otáček jednoho z kol u dvoukolového provedení.
- 2) Tříkolové podvozky – jedná se o staticky stabilní podvozek s minimálním počtem kol. Dvě kola jsou hnací a třetí kolo je hnané a slouží ke statické stabilitě podvozku.
- 3) Čtyřkolové podvozky – dvounápravové uspořádání. Přední kola bývají hnací a zadní kola hnaná. Pro zvýšení rychlosti robota mohou být všechna kola hnací. Tento typ podvozku, vzhledem k rozložení kol, má kvalitní statickou stabilitu.

- 4) Šesti a vícekolové podvozky – tyto podvozky, vzhledem k počtu kol, se hodí do horších terénů a zajišťují možnost naprogramování libovolné trajektorie.

### **2.3.2 Pásové podvozky**

Pásové podvozky umožňují pohyb po vodorovné i šikmé ploše. Využívají se pro obtížně dostupné operační prostředí např. v oblasti vojenství a policie. Pásový podvozek je řízen diferenčně a může se pohybovat po různě členitém terénu kde by měli jiné druhy podvozku problémy. Délka pásů a šíře nápravy má důležitý vliv na manévrovatelnost robotu. (Kolibal, 2016).

### **2.3.3 Kráčejíci podvozek**

Podvozek tohoto typu se využívá v prostředí, kde není možné využití kolového nebo pasového podvozku. Jedná se například o schody. V průmyslu se dá využít ve stavebnictví v horském terénu, nebo pro práci na mělkém mořském dně. Důležitou předností kráčejíciho podvozku je diskrétní stopa. Tím, že je kontakt mezi stopou kráčedla a povrchem jenom v jednom bodě, umožňuje překračovat překážky. Nevýhodou je energetická náročnost pohybu. (AUTOMA.cz, 2000).

### 3 KINEMATIKA DVOUKOLOVÉHO ROBOTA

Pro získání modelu robota se využívá fyzikálních zákonů. Nejprve je potřeba si uvědomit, že rychlost pohybu tohoto robota v operačním prostoru je popsána translační rychlostí a úhlovou rychlostí rotace podvozku. Základní úlohou je z těchto rychlostí určit odpovídající rychlost rotace kol. Pro výpočet translační rychlosti robota postačí vycházet z výpočtu obvodové rychlosti kola:

$$v_k = \omega_k r_k, \quad (3.1)$$

kde  $v_k$  – obvodová rychlost kola,  $\text{ms}^{-1}$ ,

$\omega_k$  – úhlová rychlost kola,  $\text{rad}\cdot\text{s}^{-1}$ ,

$r_k$  – poloměr kola, m.

Translační rychlost robota pak bude:

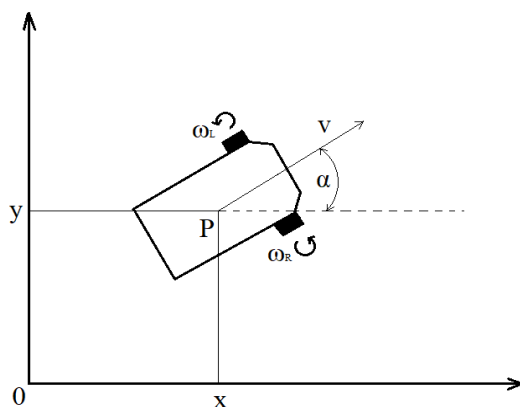
$$v = \frac{\omega_L + \omega_R}{2r}, \quad (3.2)$$

kde  $v$  – rychlost robota,  $\text{ms}^{-1}$ ,

$\omega_L$  – úhlová rychlost levého kola,  $\text{rads}^{-1}$ ,

$\omega_R$  – úhlová rychlost pravého kola,  $\text{rads}^{-1}$ ,

$r$  – poloměr kol robota, m.



Obrázek 3.1 – Ukázka robota v kartézské soustavě souřadnic

Na obrázku (3.1) je vyobrazen mobilní robot v kartézské soustavě souřadnic.

Pro úhlovou rychlost rotace podvozku mobilního robota se využije šířky robota, jednotlivé úhlové rychlosti obou kol a jejich poloměr. Vychází se ze vztahu pro obvodovou rychlost.

$$\omega = \frac{\omega_L - \omega_R}{l} r, \quad (3.3)$$

kde  $\omega$  – úhlová rychlost robota,  $\text{rads}^{-1}$ ,

$l$  – vzdálenost kol, m.

## 4 PLÁNOVÁNÍ TRAJEKTORIE

Pozici každého robota či objektu můžeme popsat v určitém prostředí, kde robot musí řešit dané situace nebo úkoly po dobu svého působení v tomto prostoru. Pro provedení pohybu robota je potřeba definovat trajektorii. Jedná se o předepsanou závislost kinematických veličin na čase. Plánování vychází ze zadaných požadavků na trajektorii.

### 4.1 NÁVRH TRAJEKTORIE METODOU POINT-TO-POINT

Pro plánování trajektorie touto metodou se zadává doba pohybu a první a poslední bod trajektorie. Jednou z možností je proložení interpolační funkce. V případě polynomu 3. řádu, je trajektorie ve tvaru:

$$q(t) = a_3t^3 + a_2t^2 + a_1t + a_0, \quad (4.1)$$

kde  $a_i$  – koeficienty polynomu,

$q(t)$  – trajektorie,

$t$  – čas, s.

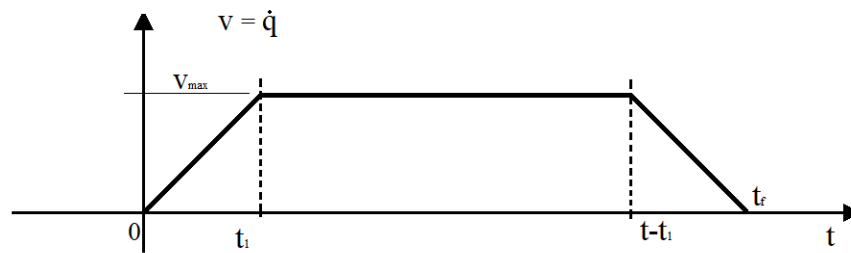
Kromě pozice jsou rovněž zadány hodnoty rychlostí  $\dot{q}(0)$  a  $\dot{q}(t_f)$ . Vztah pro rychlost lze získat derivací (4.1):

$$\dot{q}(t) = 3a_3t^2 + 2a_2t + a_1. \quad (4.2)$$

Rovnice (4.1) a (4.2) pro  $t=0$  a  $t=t_f$ , kde  $t_f$  je doba pohybu, umožňují určit koeficienty polynomu  $a_i$ .

U této metody však vznikají skokové změny zrychlení v okrajových bodech, které jsou nežádoucí. Je možné zvýšit počet parametrů tím, že se zvýší stupeň polynomu.

Další metoda, jak navrhnout trajektorii spočívá v dodržení určitého profilu rychlosti. Obvykle se používá lineární náběh s konstantním zrychlením, poté s konstantní rychlostí pohybu a následným lineárním poklesem rychlosti k nule – viz obr. (4.1). (Cvejn, 2021).



Obrázek 4.1 – Profil rychlosti s lineárním náběhem

Pro čas, kde začíná konstantní rychlost platí, že:

$$t_1 = \frac{v_{\max}}{a}, \quad (4.3)$$

kde  $v_{\max}$  – konstantní rychlost pohybu,  $\text{ms}^{-1}$ ,  
 $a$  – určité zrychlení,  $\text{ms}^{-2}$ .

Následně se určí doba pohybu, která vychází z předpokladu, že plocha pod rychlostním profilem se rovná rozdílu  $q(t_f) - q(0)$

$$q(t_f) - q(0) = (t_f - t_1)v_{\max}. \quad (4.4)$$

V případě, že  $t_1 > t_f/2$ , platí, jinak platí vztah 4.5.

$$q(t_f) - q(0) = a \left( \frac{t_f}{2} \right)^2. \quad (4.5)$$

Z těchto výpočtů se může určit zrychlení, nebo doba pohybu. Záleží na tom, co je zadáno.

## 5 ZPŮSOBY KOMUNIKACE MOBILNÍCH ROBOTŮ

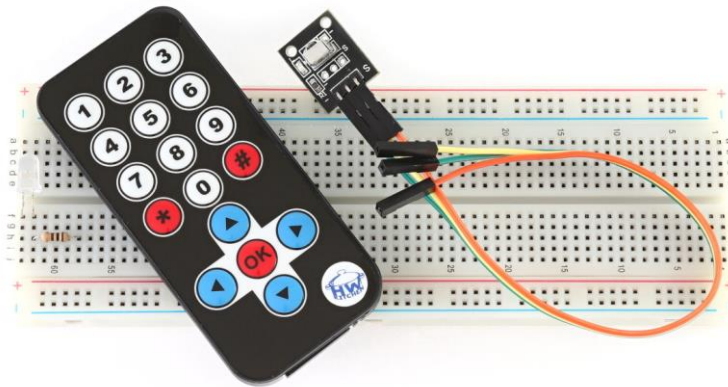
Komunikace mezi zařízeními může probíhat buď bezdrátově nebo přes kabeláž. Bezdrátová komunikace se hodí tam, kde není možné nebo kde se nechce použít k propojení zařízení kabeláž. Informace v bezdrátové komunikaci se přenáší rádiově, ultrazvukově, opticky či elektromagneticky. Mezi známější druhy bezdrátové komunikace patří například Wi-Fi. Dále pak Bluetooth nebo IR ovládání.

### 5.1 KOMUNIKACE PŘES SÉRIOVÝ PORT

U tohoto staršího druhu, který již dnešní počítače nevyužívají, se data přenáší přes kabel, bit po bitu. Při větší přenosové vzdálenosti klesá rychlost přenosu. Sériový port u počítače je odolný proti rušení a přenos může probíhat do vzdálenosti 15 metrů. U portu počítače posílají datové bity, následně přichází paritní bit, a nakonec je přenos ukončen stop-bitem, který produkuje zdržení ke zpracování dat přijímačem. Přenosová rychlost vysílače a přijímače musí být stejná. Většinou je nastavena na 9600 Bd. Pro komunikaci s počítačem přes sériovou linku je za potřebí nastavit určité parametry, jako je rychlost v Bd, počet datových bitů, parita a délka stop-bitu. (My Arduino.cz, 2022).

### 5.2 IR OVLÁDÁNÍ

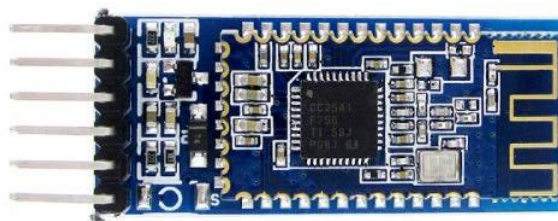
IR záření, jenž je zobrazené na obrázku (5.1), má větší vlnovou délku než viditelné světlo, a proto je neviditelné pro lidské oko. Jedná se o infračervené zařízení, které musí mít jak vysílač, tak i přijímač. Vysílačem je převážně jistý druh ovladače a slouží k převodu elektrického signálu na optický. přijímač je modul, který přijímá infračervené světlo, obsahuje obvod pro dekodování přijaté informace a převádí signál z optického zase zpátky na elektrický. Pro zamezení rušení od okolního infračerveného záření je potřeba zvolit správnou frekvenci. Nejčastěji se používá frekvence 38 kHz. IR vysílač má různý počet tlačítek. Každé toto tlačítko je naprogramováno hexadecimálním kódem, který je stejný i na straně přijímače. Dálkový ovladač používá ke komunikaci s přijímačem protokol RC5, který se však dá změnit vhodnou úpravou kódu v Arduinu. (SBProject.net, 2021).



Obrázek 5.1 – Bezdrátové ovládání pomocí IR ovladače  
(HWKITCHEN.cz, 2022)

### 5.3 BLUETOOTH

Bluetooth komunikace vznikla jako náhrada kabelové komunikace. Vzhledem k rozšíření Bluetooth na mobilní telefony, tablety nebo notebooky, mohou tyto zařízení sloužit jako vysílače. Aby komunikace mezi vysílačem a mobilním robotem fungovala, musí se na robota připojit Bluetooth modul. Komunikace je druhu master-slave, což znamená, že jedna ze dvou stran spojení řídí komunikaci. Nejprve je za potřebí spárování zařízení s řídicím modulem, poté nastává pevné spojení, a to už funguje automaticky. Oproti IR ovládání je Bluetooth o mnohem lepší, protože je nosičem rádiová vlna, místo světelného paprsku a díky tomu umožňuje komunikovat na delší vzdálenost. (Kubička, nedatováno). Na obr. (5.2) je vyobrazen Bluetooth modul typu AT-09.



Obrázek 5.2 – UART Bluetooth modul (blog.laskarduino.cz, 2021)



## 6 PRAKTICKÁ ČÁST PRÁCE

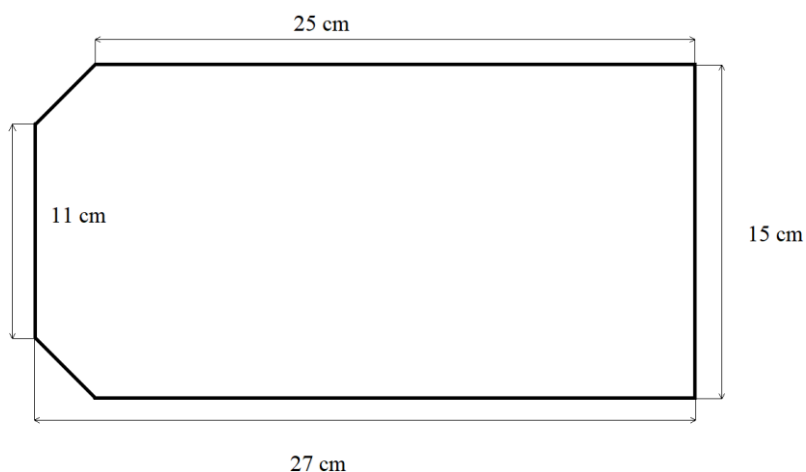
Cílem práce bylo nejen kompletní sestavení vlastního robota s využitím určitých komponent, ale rovněž i jeho softwarové řízení. Pro programování robota bylo zvoleno vývojové prostředí Arduino. Dvoukolový robot přijímá příkazy přes sériovou linku, kde se navolí směr pohybu, rychlost pohybu, stupně natočení, doba konání tohoto příkazu a okamžitá změna příkazu. Není nutné čekat na dokončení aktuálního příkazu, protože posloupnost příkazů je možné zřetěžit. Pro návrh trajektorie je využito plánování rychlostního profilu, kde je zadána doba pohybu a maximální rychlost.

Vozidlo je řízeno příkazy jízdy dopředu, dozadu, doleva a doprava. V programu je při rovném směru, na začátku i na konci jízdy, naimplementováno rozjíždění a zpomalení. K zatáčení byla použita odzkoušená rychlost motorku tak, aby odpovídalo realitě, s přihlédnutím na okolní vlivy a konstrukci vozidla. Povel řízení se budou ukládat do pole, které je založené staticky. Povel bude obsahovat také informaci o tom, zda bude příkaz proveden až na něj přijde řada, nebo se ostatní zadané příkazy vymažou a aktuálně prováděný povel bude zastaven. Následně začne neprodleně probíhat uživatelem momentálně zadaný povel. Elektrické napájení, včetně datové komunikace, bude zajištěno kabelovým propojením.

### 6.1 ROBOTICKÁ PLATFORMA

#### 6.1.1 Mechanická konstrukce mobilního robota

Základem konstrukce je deska z kuchyňského prkénka a následně za pomoci úhlové brusky vybroušena do tvaru viditelného na obrázku (6.1)



Obrázek 6.1 – Schéma desky mobilního robota

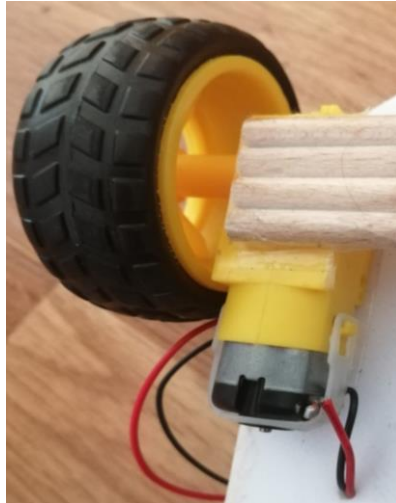
Pohyb robota zajišťují dva stejnosměrné motory s převodovkou vyobrazené na obrázku (6.2) Rovnováhu a pomocný pohyb robota obstarává rejdivací kolečko (viz obr. 6.3), které nemá motor a jeho pohyb je ovládán pomocí dvou motorů v předu. Jízda dopředu a dozadu je realizována společným pohybem dvou předních kol. Zatáčení mobilního robota je dosaženo rozdílnou rychlostí kol.

Tabulka 6.1 – Parametry dvoukolového robota

|                        |        |
|------------------------|--------|
| váha                   | 509 g  |
| výška                  | 80 mm  |
| šířka                  | 180 mm |
| délka                  | 270 mm |
| poloměr kola           | 33 mm  |
| převodový poměr motorů | 1:48   |

V tab. 6.1 jsou vypsány mechanické parametry robota.

Vzdálenost kol byla měřena od středu prvního kola, po střed druhého a poloměr kol byl měřen mezi středem kola a místem dotyku kola se zemí. Těchto parametrů bude zapotřebí v další části práce.



Obrázek 6.2 – Kolo s DC motorem



Obrázek 6.3 – Rejdivací kolečko

### 6.1.2 Klon Arduino Uno R3

Pro řízení robota je využito mikropočítačového modulu Arduino Uno. Tento modul je nejspíše nejpoužívanějším druhem mikropočítačového modulu. Má procesor ATmega328 a také obsahuje USB port. Většina projektů se dělá z tohoto modulu, protože co není na desce, se dá zařídit pomocí shieldů, které jsou vyráběny speciálně pro tento typ desky. Z této desky se vyvinuly i dvě specifické desky. Jedna z nich je Arduino Ethernet, která má místo USB portu Ethernet port a slot pro microSD karty. Druhá z nich je Arduino Bluetooth, jenž má místo USB zabudovaný bluetooth modul. (Voda, 2015).

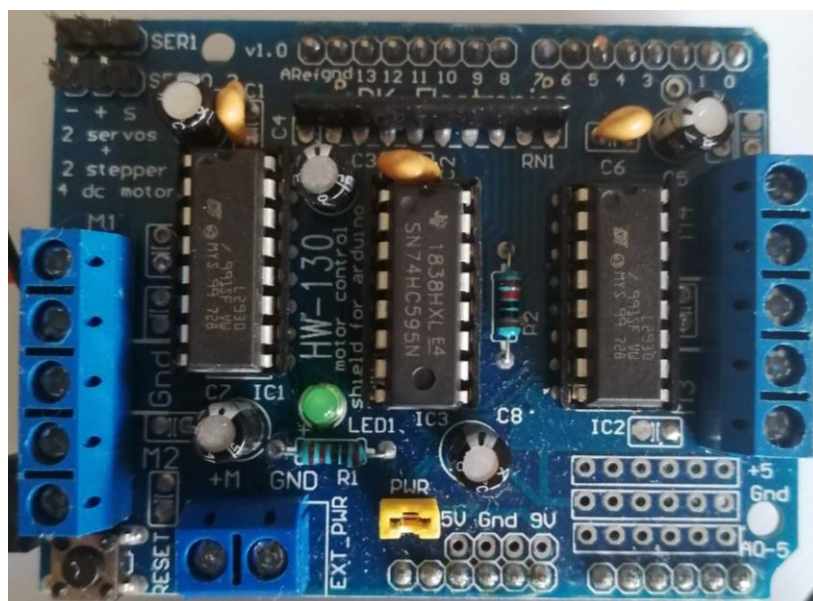
Pro tuto práci byla zvolena vývojová deska klon Arduino Uno R3 viditelné na obr. (6.7). Jedná se o klon, tudíž nebyla zaručena jeho plná funkčnost. Klon vychází z originální desky Arduino Uno Rev3. Modul obsahuje procesor ATmega328, který využívá 3 různé paměti. Vývojová deska obsahuje konektor micro USB, takže postačí k propojení s PC kabel, který je dosti využíván pro nabíjení mobilů. Obsahuje mikrokontrolér ATmega328P. Deska dále obsahuje 14 digitálních I/O pinů a 6 analogových výstupů. Provozní napětí je 5 V a kapacita FLASH paměti je 32 kB. Vstupní napětí je doporučováno na 12 V.



Obrázek 6.7 – Klon Arduino Uno R3

### 6.1.3 Motor driver shield L293D

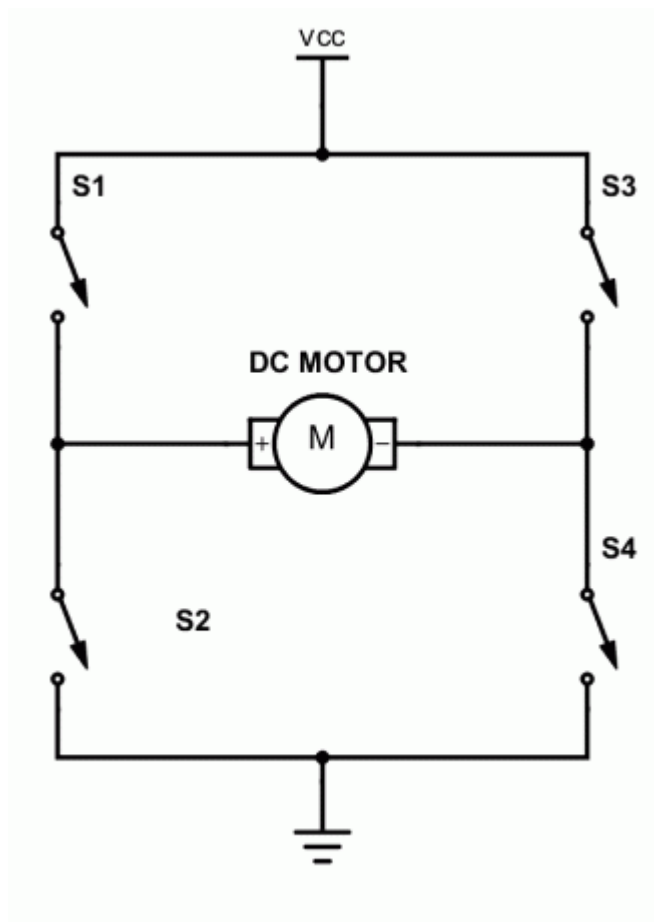
Řízení motorů zajišťuje Arduino shield L293D (viz obr. 6.4). Jedná se o dvojité bipolární H-můstek a připojit se na něho mohou buď 4 stejnosměrné motory nebo 2 servo motory či 2 krokové motory. Výhodou tohoto modulu je tlačítko reset a přesné pasování do vývojové desky Arduino Uno, díky čemuž není potřeba modul propojovat pomocí drátků.



Obrázek 6.4 – Motor driver shield L293D

H-můstek neboli h-bridge je důležité zařízení, které slouží na spínání součástek s různým napětím. Jedná se o elektrický obvod sestavený ze čtyř tranzistorů (viz obr. 6.5), který je určen na ovládání stejnosměrného motoru s permanentními magnety. Tyto čtyři tranzistory jsou otevřeny do kříže, díky čemuž mohou posílat opačně orientovaný stejnosměrný proud, bez změny polarity na zdroji. Motor se pak může začít otáčet na druhou stranu. (Paštika, 2017).

Při sepnutí spínače S1 a S4 se motor bude otáčet na jednu stranu. Další možnost nastává při sepnutí spínačů S3 a S2 a ta odpovídá reversnímu směru otáčení. Aby H-můstek fungoval, tak nesmí být v jednu chvíli sepnuty oba spínací prvky na jedné straně, což je například S1 a S2. Mohlo by pak dojít ke zkratu a proud by zničil sepnuté tranzistory.



Obrázek 6.5 – Schéma H-můstku (cs.technolong.com, 2022)

Pro programování tohoto modulu byla použita Knihovna AFMotor.h (dostupné z <https://github.com/adafruit/Adafruit-Motor-Shield-library>) od Adafruit, která byla vytvořena pro tento typ shieldu. Knihovna obsahuje základní nastavení pro práci s motory viditelné na obrázku (6.6). Nastavení pinů pro stejnosměrné motory a základní funkce pro směr dopředu, dozadu či zastavení.

```

// Bit positions in the 74HCT595 shift register output
#define MOTOR1_A 2
#define MOTOR1_B 3
#define MOTOR2_A 1
#define MOTOR2_B 4
#define MOTOR4_A 0
#define MOTOR4_B 6
#define MOTOR3_A 5
#define MOTOR3_B 7

// Constants that the user passes in to the motor calls
#define FORWARD 1
#define BACKWARD 2
#define BRAKE 3
#define RELEASE 4

// Constants that the user passes in to the stepper calls
#define SINGLE 1
#define DOUBLE 2
#define INTERLEAVE 3
#define MICROSTEP 4

```

Obrázek 6.6 – Ukázka nastavení pinů a základního nastavení v knihovně AFMotor.h

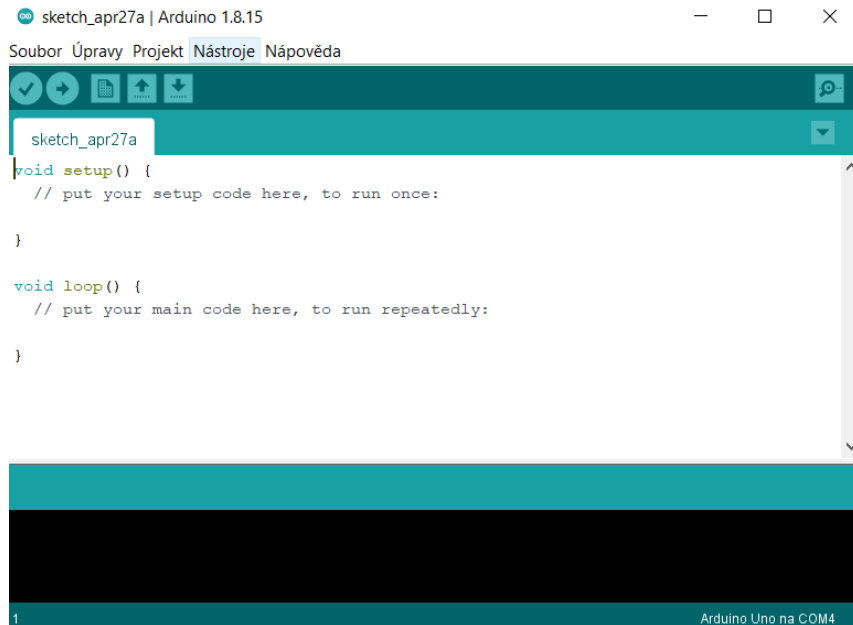
### 6.1.4 Vývojové prostředí Arduino IDE

Arduino IDE, viditelné na obrázku (6.8), je integrované vývojové prostředí, které je napsáno v jazyce Java. Jedná se o software, který vznikl z výukového prostředí Processing, následně byl upraven o určité funkce a v neposlední řadě má podporu jazyka Wiring. (Voda, 2015).

Základem pro práci s programem je rozbalovací nabídka nástroje (tools), zde nalezneme nastavení pro programování desky. Zde nalezneme funkci sériový monitor. Ta otevře ovládací panel pro psaní příkazů, které se naprogramovaly a následně jsou tyto příkazy posílány do vývojové desky. Dále nás bude zajímat ikona ověřit označená "fajfkou", která slouží pro kontrolu programu. Pokud nalezne nějakou chybu, pak je tato chyba zvýrazněna červeně. Vedle ikony ověření je ikona nahrát. Tato ikona slouží pro nahrání programu do připojené vývojové desky Arduina. Arduino je možné programovat v jazyce C nebo C++.

Balíčky, které obsahují nové funkce, které lze využít pro programování se nazývají knihovny. Není to nic jiného než dvojice souborů s koncovkou .h. V jednom programu lze

použít libovolné množství knihoven, ale jelikož každá knihovna zvětší výsledný program, tak jsme omezeni vnitřní pamětí Arduina. Pro vzor knihovny vycházíme z: `#include <jméno_knihovny.h>`.

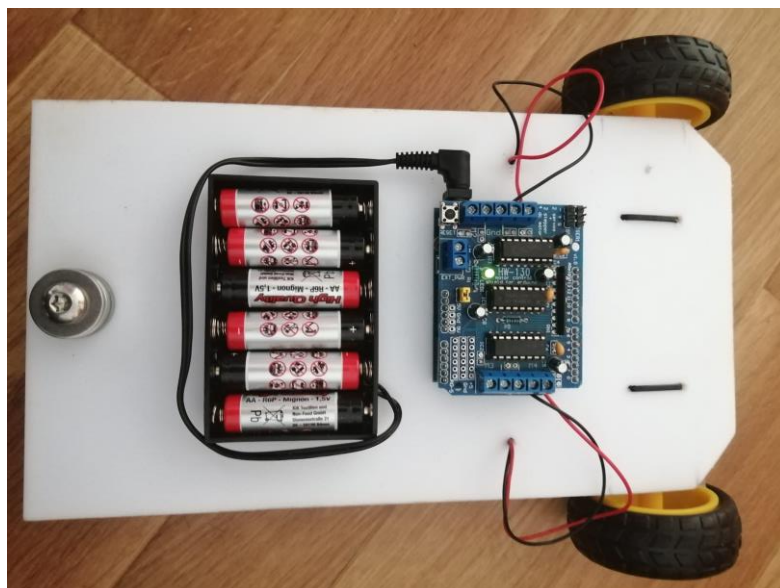


Obrázek 6.8 – Vývojové prostředí Arduino

### 6.1.5 Zapojení komponent

Základ hardwarových komponent je deska, ke které je šrouby připevněna vývojová deska Arduino Uno R3 a na ni připevněný shield L293D. DC motory jsou připevněny pomocí oboustranné lepicí pásky, podepřené dřívky a zajištěny stahovacími páskami. Zadní rejdovací kolečko a krabička na baterie jsou zajištěny pomocí šroubového spojení. Na obrázku (6.9) (6.10) je vyobrazeno zapojení celého robota. Pro napájení je použito 6 baterií AA.





Obrázek 6.9. zapojení mobilního robota seshora



Obrázek 6.10. zapojení mobilního robota zezdola

Schéma zapojení mobilního robota, vyobrazeno na obrázku (6.11), zobrazuje dva DC motory zapojené do driverů L293D. jedná se o dvoukanálový ovladač motoru s napájecím napětím 36 V a průměrný proud na kanál je 600 mA. Systém je řízen čtyřmi směrovými vstupy INPUT a dvěma aktivačními vstupy ENABLE. Frekvence řídicích signálů (při PWM modulaci) by neměla překročit 5 kHz. (botland.cz, 2023).

Tab. 6.2 obsahuje seznam vstupů a výstupů driveru L293D.

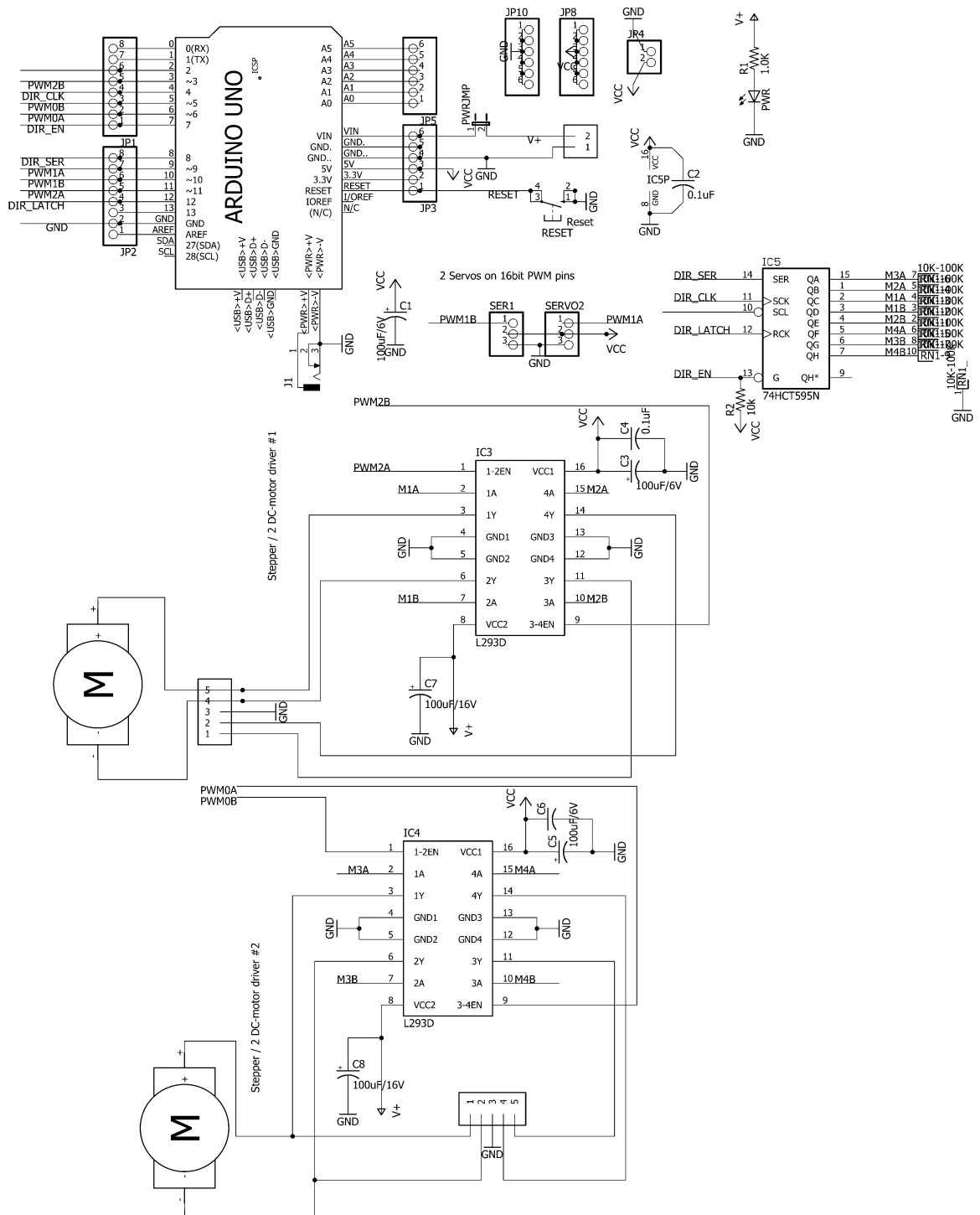
Tabulka 6.2 – Vstupy a výstupy driveru L293D

|                   |                        |
|-------------------|------------------------|
| 1-2EN (ENABLE 1)  | PWM prvního kanálu     |
| 1A (INPUT1)       | směr prvního kanálu    |
| 1Y (OUTPUT1)      | výstup prvního kanálu  |
| GND1              | uzemnění               |
| GND2              | uzemnění               |
| 2Y (OUTPUT2)      | vystup prvního kanálu  |
| 2A (INPUT2)       | směr prvního kanálu    |
| VCC2 ( $V_s$ )    | napájení motorů        |
| VCC1 ( $V_{ss}$ ) | napájení logické části |
| 4A (INPUT4)       | směr druhého kanálu    |
| 4Y (OUTPUT4)      | výstup druhého kanálu  |
| GND3              | uzemnění               |
| GND4              | uzemnění               |
| 3Y (OUTPUT3)      | výstup druhého kanálu  |
| 3A (INPUT3)       | směr druhého kanálu    |
| 3-4EN (ENABLE2)   | PWM druhého kanálu     |

Na schématu je dále vidět posuvný registr 74HC595, který je určen pro převod dat ze sériového vstupu (v nejjednodušší variantě jeden datový a jeden hodinový signál) na osm paralelních výstupů. Jedná se o soustavu klopných obvodů, kde se logická informace posouvá dále pomocí hodinového impulsu. Registr 74HC595 má 3 vstupy (SER – sériový vstup dat, SCK – vstup hodin posuvného registru, RCK – hodinový vstup uložiště registru) a 8 výstupů ( $Q_a$  až  $Q_h$ ). Z toho vyplývá, že k ovládní 8 výstupů je zapotřebí pouze 3 piny na Arduino. (arduino8.webnode.cz, 2023).

SER1 a SERVO2 jsou konektory pro připojení servo motorů, které se nacházejí v levém horním rohu desky motor shieldu.

Vrchní část schématu zabírá Arduino. A0–A5 jsou A/D převodníky a piny s “vlnkou” zobrazují PWM signály. Piny 12, 4 a 8 slouží pro ovládání posuvného registru 74HC595. Pro připojení prvního driveru jsou užity piny PWM signálu 5 a 6 a pro připojení druhého driveru piny 3 a 11.



Obrázek 6.11. Schéma zapojení mobilního robota

## 6.2 Ovládání robota

Komunikaci s robotem zajišťuje kabel, kde jsou příkazy posílány robotu po sériovém portu. Jako první je potřeba nastavit rychlost přenosu. Sériový port je nastaven na přenosovou rychlost 9600 b/s společně se sériovým monitorem. Dále je možné posílat příkazy přímo ze sériového monitoru Arduino IDE Pro port se nastavuje časové okno čtení a zároveň zápisu. Čtení a výpis byly nastaveny na 5 ms s předpokladem rezervy.

Pro komunikaci s robotem je potřeba ve funkci `setup()` nastavit rychlost přenosu dat v bitech za sekundu pro sériový přenos dat a to pomocí funkce:

```
Serial.begin(9600);
```

Dále se nastaví časové okno [ms] pro čtení a výpis sériového portu pomocí funkce:

```
Serial.setTimeout(5);
```

Každý zadávaný parametr programu je separován dvojtečkou. Separace v příkazovém řádku je nutná pro rozdělení příkazu na jednotlivé části. Po sériovém portu se posílají následující příkazy:

Tabulka 6.3 – Parametry pro příkazový řádek

| zadávané parametry              | možné varianty                   |
|---------------------------------|----------------------------------|
| směr pohybu                     | dopředu, dozadu, doleva, doprava |
| rychlost                        | 0-120                            |
| úhel                            | 0-320                            |
| čas                             | 0.00-neomezeno                   |
| následující příkaz <sup>o</sup> | p, z                             |

- p – následující příkaz, který se naváže na probíhající příkaz.
- z – okamžitá změna, která hned ukončí probíhající příkaz a nahradí ho.

Pro ukázkou formátu příkazů je použit následující příklad:

dopredu:20:000:1.20:p

První část příkazu se píše bez háčku a čárek. Je možné psát jak velkými písmeny, tak malými, protože program je nastaven tak, že všechna velká písmena přepíše na malé.

Druhá, třetí a čtvrtá část příkazu je zadávána číselně, s tím že u druhé části musí být tři číslice a u čtvrté části musí být dvě desetinná místa.

Poslední část je už jen volba mezi p (zřetězený příkaz) a z (okamžitá změna)

Při chybném zadání příkazu posílá mobilní zařízení po sériové lince informace o chybě do osobního počítače. Při nedodržení syntaxe vypíše robot zprávu o tom, že zadaný řetězec neobsahuje správnou strukturu zápisu. Zařízení dále vypisuje různé informace, jež byly nastaveny ve formě ladění. Z těch důležitějších to je primárně informace o čase, zrychlení a zpomalení s krokem 5 a typ směru jízdy. Pro menší kapacitu paměti a pro rychlejší zpracování příkazu zůstala však pouze informace o čase. Ostatní funkce ladění lze zprovoznit odkomentováním jednotlivých ladění.

### 6.2.1 Programové řešení řízení robota

Robot byl testován na plovoucí podlaze a Linoleu. Praktickým testem byl odzkoušen rozsah rychlosti od 120 do 255, kde je (u rychlosti 120) 5% šance, že se motor neroztočí. Rychlost byla přenastavena na rozsah od 0 do 125, kde nula odpovídá hodnotě 130. Při plném nabití baterií bylo testováno, jakou vzdálenost ujede mobilní robot, při konstantní rychlosti, za 1 s. Tomu odpovídá následující tabulka:

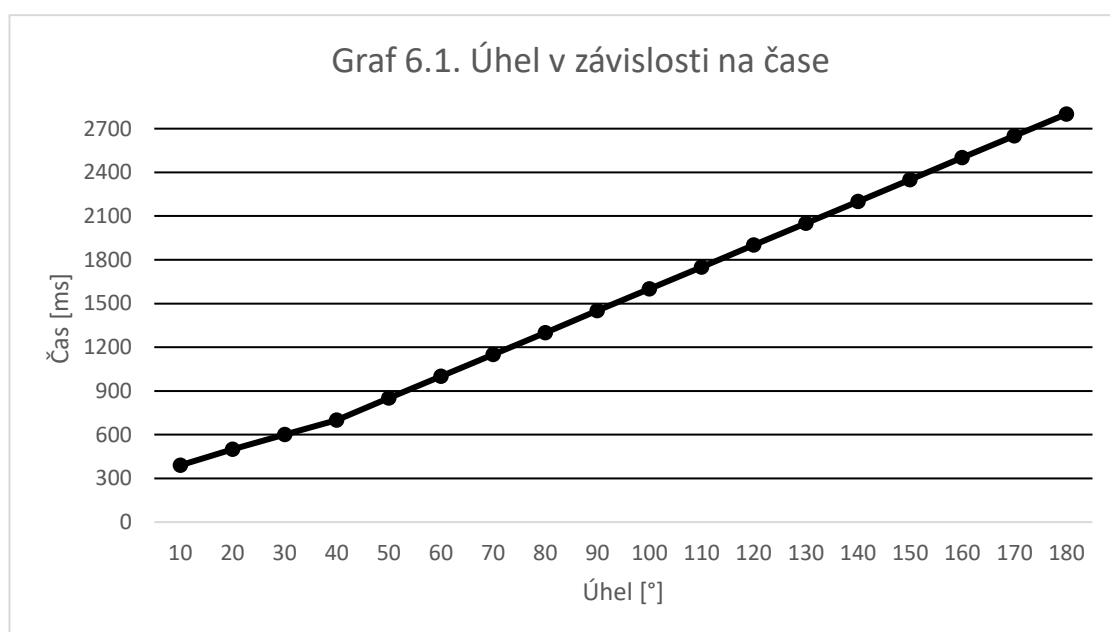
Tabulka 6.4 – Ujetá vzdálenost mobilního robota na čase

| hodnota rychlosti | ujetá vzdálenost |
|-------------------|------------------|
| 0 (130)           | 0,32 m           |
| 20 (150)          | 0,39 m           |
| 70 (200)          | 0,5 m            |
| 120 (250)         | 0,57 m           |
| 125 (255)         | 0,59 m           |

Vzhledem k délce poloosy nefunguje otáčení nastavením různých rychlostí kol. Optimálním provedením otáčení vozidla je zastavení jednoho kolečka a otáčení pouze kolečka druhého. Praktickým testováním bylo dosaženo závěru, že rychlost 130 po dobu 390 ms je optimální a vozidlo se otočí přibližně o 10°, což je krokové nastavení pro otáčení vozidla vzhledem k prokluzu při roztočení kolečka. Při roztočeném kolečku již nedochází k tak markantnímu protočení. Testováním byla zjištěna závislost úhlu otočení v závislosti na čase, která je uvedena v tabulce (6.5).

Tabulka 6.5 – Úhel v závislosti na čase

| úhel | čas     |
|------|---------|
| 20°  | 500 ms  |
| 30°  | 600 ms  |
| 40°  | 700 ms  |
| 90°  | 1450 ms |
| 180° | 2800 ms |



Obrázek 6.12. Graf úhlu v závislosti na čase

Na obrázku (6.12) je vidět graf úhlu v závislosti na čase.

z čehož je možné určit dobu otočení o daný úhel:

$$\frac{1400 \text{ ms}}{90^\circ} = 15,56 \text{ ms/stupeň.} \quad (6.1)$$

Pro výpočet ostatních úhlů natočení je možné použít výpočet:

$$\text{Doba otáčení [ms]} = 700 + 15 \cdot (\text{úhel } [^\circ] - 40), \quad (6.2)$$

Pro úhel 180° pak platí že:

$$\text{úhel} = 700 + 15 \cdot 140 = 2800. \quad (6.3)$$

Na základě testu a přesnosti otočení bylo rozhodnuto o kroku 10° s tím, že řád stupňů bude vždy zaokrouhlen dolů – takže od 0° až do 9° bude výsledek 0°. V případě rozmezí 10–19° bude výsledek 10°. Test byl proveden také na otočení vozidla. Testy se však nezdařily, neboť kolečko otáčející se ve zpětném směru nedokáže vyvinout potřebnou sílu na to, aby došlo k roztočení a tím zatačení vozidla. Obě kolečka byla schopna se otáčet s hodnotou 255 v dopředném směru a 200 ve zpětném směru, což je velice malý rozsah na otočení.

Postupné zrychlování se řeší v přerušení, kde se přičte konstantní hodnota, což znamená, že se rychlost mění lineárně s konstantním zrychlením pomocí časového přerušení. Například pro rychlosti 170, 210 a 255, kde u rychlosti 170 bude doba zrychlování okolo 0,5 s, u 210 pak 1 s a u rychlosti 255 to bude 1,5 s. Rychlost se bude navyšovat o 5 až do maxima. Když bude robotu zadán příkaz, kde bude například zadaná rychlost 70 a čas 1 sekunda, tak to neznamená že robot bude vykonávat trajektorii 1 s. nejdříve bude zrychlovat 0.5 s, pak pojedou konstantní rychlostí 1 s a pak bude zpomalovat 0.5 s, což znamená, že výsledný čas vykonávaného příkazu bude 2 s.

Výpočet úhlové rychlosti v této práci nebyl aplikován, protože při vyšší rychlosti docházelo k prokluzu koleček a úhel otočení nebyl nikdy totožný. Při nastavení úhlové rychlosti jednotlivých motorků by se vycházelo ze vztahů:

$$v_1 = v \left( 1 + \frac{d}{R} \right), \quad (6.3)$$

$$v_2 = v \left( 1 - \frac{d}{R} \right), \quad (6.4)$$

kde  $v$  – celková rychlost pohybu mobilního robotu,  $\text{ms}^{-1}$ ,

$v_1$  – rychlost jednoho kola,  $\text{ms}^{-1}$ ,

$v_2$  – rychlost druhého kola,  $\text{ms}^{-1}$ ,

$R$  – vzdálenost osy otáčení, m.

$d$  – průměr vzdálenosti kol, m.

Vztah platí pouze v případě, že  $R \geq d$ .

Zvolená varianta natočení robota byla taková, že se robot bude otáčet konstantní nejnižší rychlostí. Jedno kolo je nehybné a druhé nastaveno na rychlost 130.

## 6.2.2 Struktura řídicího systému

Základními definicemi programu jsou:

- nastavení řetězce, jeho časování a aktivace:

```
#define VELIKOST_POLE_RJ 3           Velikost pole řídicí jednotky s hodnotou 3.
#define DOBA_ZATACENI_10 390       Čas, za který se robot otočí o 10°.
#define ODDELOVACI_ZNAK_PRIKAZU ':' Znak separace.
#define ULOZENI_NASTAVENI_RJ_DO_POLE 'p' Další příkaz na uložení do zásobníku.
#define OKAMZITE_PRENASTAVENI_RJ 'z' Okamžitá změna.
#define KONEC_RETEZCE '\0'        Znak pro ukončení příkazu (enter).
#define DOSTUPNOST_RETEZCE_BUFFERU 0 Zásobník.
#define DELKA_PRERUSENI_MS 30      Délka přerušení nastavená na 30 ms
#define DOBA_AKTIVACE_SERIOVEHO_PORTU_MS 1000
```

- délky částí řetězců pro příkaz sloužící pro nastavení vozidla:

```
#define DELKA_PRVNI_CASTI_RETEZCE 8
#define DELKA_DRUHE_CASTI_RETEZCE 4
```



```
#define DELKA_TRETI_CASTI_RETEZCE 4
#define DELKA_PATE_CASTI_RETEZCE 2
#define DELKA_POMOCNEHO_RETEZCE DELKA_PRVNI_CASTI_RETEZCE
#define DELKA_TEXTOVEHO_POVELU DELKA_PRVNI_CASTI_RETEZCE
```

Je také třeba deklarovat třídy motorů a nastavit rychlost PWM signálu, IO pracuje do 5 kHz. Pro společné nastavení motorů byl z definovaných možností vybrán 1 kHz.

```
volatile AF_DCMotor motor_levy(1, MOTOR12_1KHZ);
volatile AF_DCMotor motor_pravy(4, MOTOR34_1KHZ);
```

Pro využití minimální kapacity paměti byly logické příznaky deklarovány jako bitové pole.

```
typedef struct {
volatile unsigned stav_vozidla           :3;
volatile unsigned motorky_vypnuty       :1;
volatile unsigned povoleno_zpracovani_retezce :1;
volatile unsigned povoleno_prenastaveni_v_preruseni :1;
volatile unsigned dalsi_prikaz_na_zpracovani :1;
volatile unsigned okamzita_zmena_rj     :1;
volatile unsigned rezim_motorku_nastaven :1;
volatile unsigned casovac_zapnut        :1;
volatile unsigned povel_dokoncen        :1;
volatile unsigned povel_nacten          :1;
volatile unsigned zmenit_rezim_motorku   :1;
volatile unsigned pole_bitu             :3;
} BITOVE_POLE;
```

Řídicí jednotka je v řídicím systému reprezentována následující strukturou:

```
typedef struct {
char textovy_povel[DELKA_TEXTOVEHO_POVELU];
byte rezim_levého_motorku;
byte rezim_praveho_motorku;
volatile byte rychlost_levého_motorku;
```

```

volatile byte rychlost_praveho_motorku;
volatile unsigned int doba_povelu;
volatile int uhel_zatoceni;
} RIDICI_JEDNOTKA_VOZIDLA;

```

Z hlediska sekcí v programu mobilního robotu je program rozdělen do pěti základních skupin:

- 1) funkce setup() – vstupní nastavení parametrů programu.

Zde se nastavuje přerušení (viz kapitola 6.2.3), přenosová rychlost pro port a sériový monitor [b/s] a nakonec nastavení časového okna pro čtení a výpis [ms.]

- 2) funkce loop() – nekonečný neustále běžící cyklus.

Nastaví se pole řídicích jednotek, a struktury pole poté na výchozí nastavení, které je rozděleno do tří částí, jimiž jsou – výchozí nastavení databáze povelu, výchozí nastavení pole k provádění povelů a výchozí nastavení dokončeného povelu. Nakonec je třeba nastavit cyklus while k řízení vozidla.

- 3) ISR – definice funkce přerušení a nastavení stavů vozidla.

Přerušení se definuje pomocí funkce:

```

ISR (TIMER1_COMPA_vect)
{
}

```

kde TIMER1 je šestnácti bitový časovač. V přerušovací rutině se inkrementují proměnné, a to počet přerušení (úprava proměnné pro stanovení nové rychlosti) a násobek dvou (když uběhly dvě přerušení pro navýšení rychlosti). Stav mobilního robota jsou rozděleny pomocí funkce case (Ukázka části kódu v kapitole 6.2.2 níže).

- 4) úprava databáze – přesunutí příkazu, okamžitá změna.

V této části se přesouvají povely z databáze zaslaných povelů. Například pokud je v databázi alespoň jeden povel, přesune se do kopie aktuální jednotky a následně se nahraje povel nový. Když není zadána okamžitá změna, přesune se povel z druhého místa na místo první a první povel na pozici s indexem nula. Když nastane okamžitá změna, databáze se vyprázdní a s časovou rezervou 1 s se začne vykonávat příkaz nový.

5) zpracování řetězce – separace, zadávané parametry.

Separace slouží k rozdělení příkazu. Je nutné nalézt oddělovací znaky. Dále je třeba uložit index pro další pokračování zpracování řetězce a nakonec nahrazení znaku ukončovacím znakem. Zadávané parametry jsou rozděleny do pěti částí. První část je směr jízdy, kde se rozdělují zadané povely a nastavují se odpovídající režimy motorků. V druhé části se řeší rychlost, u které zadávaný znak musí být číslo. Třetí částí je řešení úhlu natočení. Když je zadaný povel dopředu, nebo dozadu, úhel se vynuluje. V případě zadání povelu pro zatáčení se převede řetězec příkazu na číslo a rychlost ani čas nehrají důležitou roli a nejsou brány v potaz. Čtvrtá část příkazu řeší čas. Ten je zadán s přesností na dvě desetinná místa. V páté, poslední části příkazu se nastavuje okamžitá změna a další příkaz do zásobníku. V případě okamžité změny se překopíruje uložená struktura do databáze s indexem 0 a v databázi zůstane pouze tento jeden povel.

Pohyb robota je realizován ve čtyřech stavech, jimiž se rozumí zrychlování, zpomalování, konstantní jízdu a stání. Zde je navíc ještě jeden stav a to zatáčení. Kvůli zvolenému řešení tohoto pohybu má zatáčení svůj vlastní stav a nejde tedy pouze o směr jízdy. Robot se chová jako konečný automat, kde se tyto stavy mezi sebou přepínají podle potřeby. Každý stav má určité přechodové podmínky, například pro zrychlení se řeší, zda se vozidlo rozjíždí ze stání, nebo zda již bylo dosaženo požadované rychlosti. V tom případě se robot přepne ze stavu zrychlení do stavu konstantní jízdy. Nakonec se řeší možnost, že se už vozidlo pohybuje a má zrychlit. V následujícím kódu, který se nachází v bloku ISR, je část programu pro stav zrychlení kdy se vozidlo rozjíždí ze stání za pomocí funkce case.

```
break;
```

```
case ZRYCHLOVANI:
```

```
if(nasobek_dvou == 2) //Když uběhly 2 přerušení pro navýšení rychlosti.
```

```
pocet_preruseni = pocet_preruseni / 2; //Úprava proměnné pro stanovení nové rychlosti.
```

```
if(predchozi_provadena_jednotka.rychlost_leveho_motorku == 0 AND
```

```
kopie_aktualni_jednotky.rychlost_leveho_motorku > 0) //Když se vozidlo rozjíždí ze stání.
```

```
if((MINIMALNI_RYCHLOST + ZMENA_RYCHLOSTI * (pocet_preruseni - 1)) <=
kopie_aktualni_jednotky.rychlost_leveho_motorku) //Když je aktuální rychlost menší než
požadovaná.
```

```
motor_levy.setSpeed((byte)(MINIMALNI_RYCHLOST + (ZMENA_RYCHLOSTI * (pocet_preruseni -
1))));
```

```
motor_pravy.setSpeed((byte)(MINIMALNI_RYCHLOST + (ZMENA_RYCHLOSTI * (pocet_preruseni -
1)))); //Nastavení rychlostí
```

Přerušeni slouží k změnám jednotlivých stavů mobilního robota. Níže je ukázáno nastavení přerušeni ve funkci setup().

```
cli(); //globální zakázání přerušeni.
```

```
TCCR1A = 0; //zakázání změn stavu výstupního portu B; nastaveni bitu WGM11+10 pro režim CTC.
```

```
TCCR1B &= B11100000; //první 3 bity se ponechají ve výchozím nastaveni; zbylé bity se vynulují pro
další nastaveni CTC.
```

```
TCCR1B |= (1 << WGM12) | (1 << CS11); //nastaveni bitu WGM12 pro CTC, WGM13 se ponechá v 0;
nastaveni předděličky pomoci bitu CSXX na clk/8.
```

```
OCR1AH = B11101010; //vlození hodnoty pro režim CTC komparátor; první se nastavuje horni Byte
MSB.
```

```
OCR1AL = B01011100; //vlození hodnoty pro režim CTC komparator; druhý se nastavuje dolni Byte
LSB.
```

```
TIMSK1 &= B00000000; //zakázání všech přerušeni pro C/C1.
```

```
TIMSK1 |= (1 << OCIE1A); //povoleni přerušeni komparátoru A C/C1; obsluha přerušeni funkci ISR.
```

```
sei(); //globalni povoleni preruseni.
```

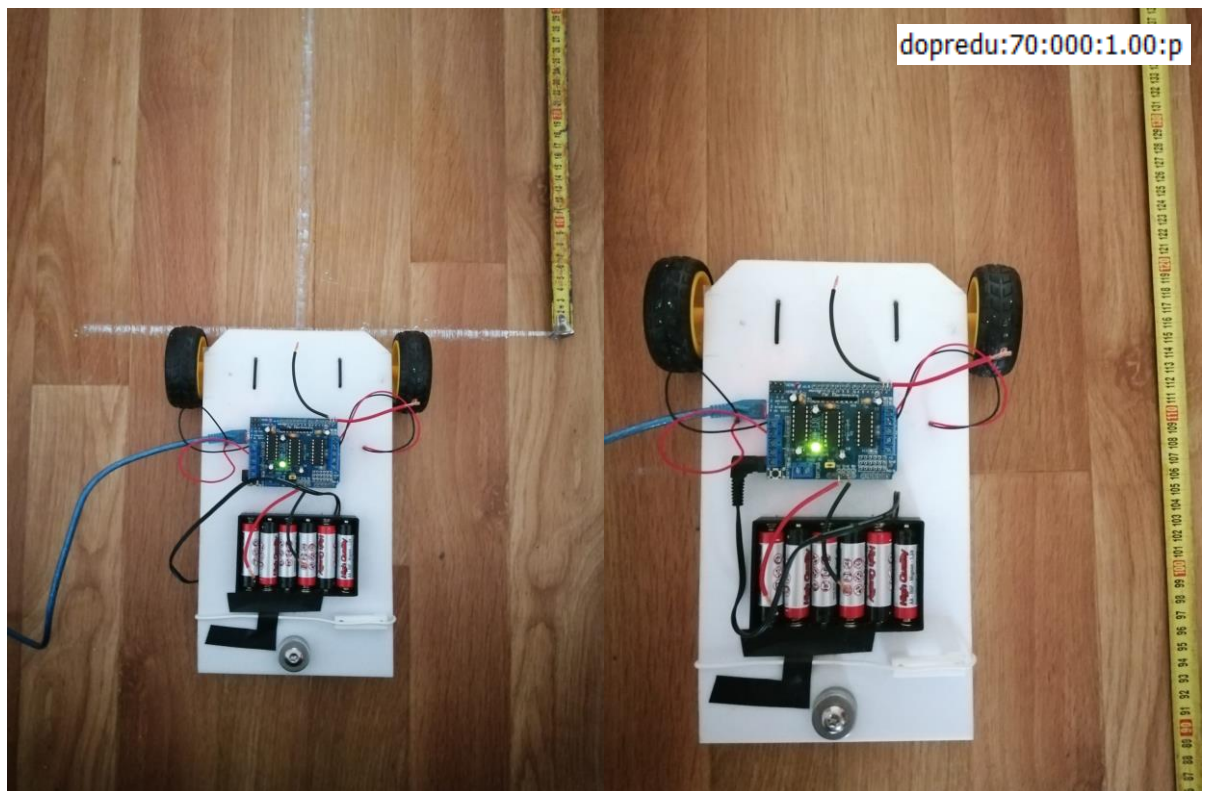
Pro využití časového přerušeni je nutné nastavit následující registry mikropočítače:

- TCCR1A (Timer/Counter1 Control Register A) – Řídící registr A časovače/ čítače, jedna se o první registr periferie Timer 1, který je 16ti bitovým volně běžícím časovačem
- CTC (Clear Timer on Compare) – Režim časování
- OCR1Ah a OCR1AL (Output Compare Register 1 A) – Výstupní porovnávací registry, které mají 16ti bitovou hodnotu. Tato hodnota je porovnávána s hodnotou čítače TCNT1, shoda se dále používá pro generování přerušení.
- TIMSK1 (Timer/Counter1 Interrupt Mask Register) – maska přerušení časovače. Obsahuje řídicí bity přerušení

Dále je nutné definovat obsluhu přerušení v bloku ISR. Přerušení se používá například pro okamžitou změnu příkazu, nebo změnu stavů vozidla.

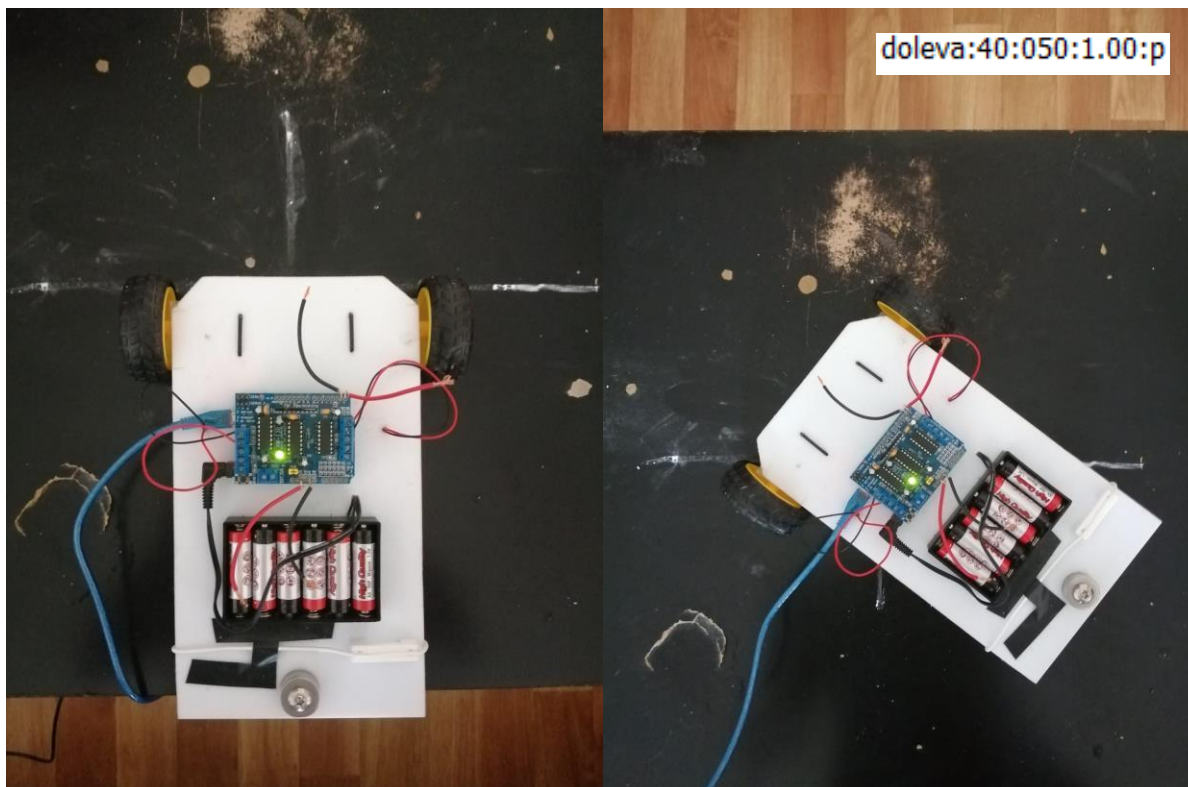
### **6.3 EXPERIMENTY S ROBOTEM**

Experimenty byly prováděny na linoleu, s výjimkou úhlu natočení, pro který byl vhodnější dřevěný podklad. Otestováno bylo zrychlení, úhel natočení, zpomalení, jízda dopředu a jízda dozadu. Pro první experiment byla zvolena jízda dopředu. Na obrázku (6.13) je nalevo stav robota před zadáním příkazu a napravo po jeho vykonání. Pro tento experiment byl zadán příkaz, který je zobrazen v pravém horním rohu. Tímto experimentem bylo zjištěno, že mobilní robot ujede s rychlostní hodnotou 200 120 cm za 1 s.



Obrázek 6.13. Pokus č. 1

Dalším pokusem, provedeným tentokrát na dřevěném stolku, byl odzkoušen úhel natočení, v tomto případě byl zvolen úhel  $50^\circ$ . S výjimkou určitých drobných odchylek, jež byly způsobeny nerovnostmi a děrami ve stolku, byl zadaný příkaz v pořádku úspěšně proveden. Robot se, s rychlostní hodnotou 130, otočil za 850 ms o zhruba  $50^\circ$ . Na obrázku (6.14) je opět vidět start a stav vozidla po vykonání příkazu.



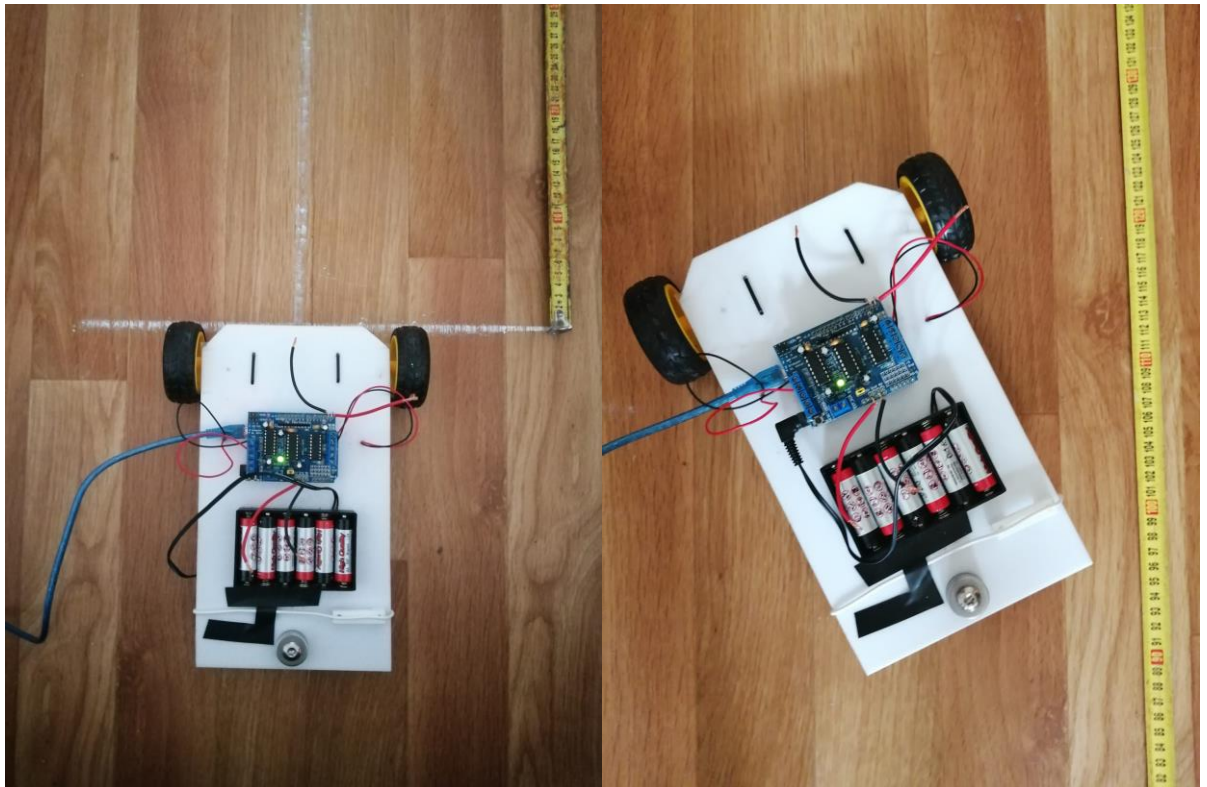
Obrázek 6.15. Pokus č. 2

Poté byla vyzkoušena varianta řetězení příkazů za sebou, kde první prováděný příkaz byla jízda dopředu a poté úhel natočení. Příkazy pro tento test byly následující:

dopředu:70:000:1.00:p

doleva:50:050:1.00:p

Robot zrychloval po dobu 0.5 s, 1 s jel konstantní rychlostí, poté 0,5 s zpomaloval a nakonec se, s časovou prodlevou, otočil o téměř 40°. Vzhledem k povrchu v podobě linolea byl úhel lehce nepřesný, protože se kolečko zadržovalo. Na obrázku (6.14) je možné vidět stav vozidla před vykonáním řetězce příkazů a po jeho vyplnění.



Obrázek 6.15. Pokus č. 3

Poslední ukázkou řetězce příkazů je, v příloze nahrané, video. Řetězec obsahuje šest zadaných příkazů které by měl robot dle svého naprogramování vykonat. Zadány byly následující příkazy:

dozadu:40:000:1.00:p

dopředu:70:000:1.00:p

dopředu:125:000:4.00:p

doleva:50:090:1.00:z

dozadu:125:000:1.50:p

dozadu:40:000:1.00:p

Jako první byla otestována jízda dozadu, robot zde začal zrychlovat, jel konstantní jízdou a následně zpomalil na stání. Poté se začal rozjíždět dopředu, jel po dobu jedné sekundy konstantní rychlostí 200 a poté se přepnul na následující příkaz, kde začal zrychlovat



na hodnotu 255. Po posláni příkazu na zatáčení se předchozí příkaz okamžitě vypnul a byl nahrazen následujícím, tedy čtvrtým, příkazem. Nepodařilo se plně dosáhnout cílového úhlu, neboť kolečko nevyvinulo kvůli linoleovému povrchu dostatečnou sílu na otočení vozidla. Po vykonání tohoto příkazu se autíčko začalo rozjíždět dozadu a s posledním příkazem zpomalilo na hodnotu 170. Po 1 s počalo zpomalovat až do stavu stání.

## 7 ZÁVĚR

Cílem této bakalářské práce bylo vytvoření mobilního robota a příslušného řídicího systému. Pro řízení bylo využito Arduino Uno, které komunikovalo s osobním počítačem skrze USB připojení.

Kompletní programové řešení bylo vytvořeno ve vývojovém prostředí Arduino IDE a za pomoci knihovny, jež sloužila k programování modulu motor driver shield L293D. Program obsahuje potřebné nástroje pro nastavování parametrů jízdy, ovšem až na úhlovou rychlost. Jako alternativní řešení tohoto problému bylo zvoleno úhlové natočení při konstantní rychlosti jednoho kolečka. To se ukázalo jako vhodnější řešení, neboť při úhlové rychlosti docházelo k prokluzu a úhel z tohoto důvodu neodpovídal zadanému příkazu. Další součástí řešení bylo také testování komunikace a funkčnosti spojení mezi modulem Arduina UNO a osobním počítačem. To proběhlo v pořádku a bylo vyhodnoceno jako plně funkční.

V testovací části byla provedena série pokusů, jejichž obsahem byly veškeré možné nastavitelné parametry trajektorie. Program zvládá zřetězení příkazů jakožto i okamžitou změnu příkazu stávajícího. Zjištěno bylo díky pokusům také to, že nejvhodnějším podkladem pro jízdu motorového robota je podklad. Odkoušena byla taktéž jízda na linoleu a dlažbě, zde však docházelo k prokluzu anebo zadrhávání kol robota a tím k jeho neúplnému výkonu.

Možným vylepšením by bylo vybavení senzory, které by umožnily měřit vzdálenosti k překážkám a vyhýbat se jim. Ke komunikaci s robotem by bylo možné využít také komunikaci bezdrátovou, například pomocí Bluetooth

## POUŽITÁ LITERATURA

BUCHLI, J. 2006. *Mobile Robotics, Moving Intelligence*. London: IntechOpen, 588 s. ISBN 3-86611-284-X.

*Bezdrátové ovládání pomocí IR ovladače*. 2022. [online]. HWKITCHEN. [cit. 2022-04-29].  
Dostupné na: <https://www.hwkitchen.cz/navody-hwkitchen/bezdratove-ovladani-pomoci-ir-ovladace-arduino-navody/>.

*Co je H-bridge, jak to funguje a jak je používat?* 2022. [online]. technolong. [cit. 2022-04-29]. Dostupné na: <https://cs.technolong.com/advice/4207-what-is-an-h-bridge-how-it-works-and-how-to-use>.

CVEJN, J. 2021. *Průmyslové roboty*. [online]. Pardubice: Univerzita Pardubice, FEI, Elektronický studijní materiál k předmětu Průmyslové roboty.

HAVEL, I M. 1980. *Robotika: Úvod do teorie kognitivních robotů*. Praha: SNTL, 279 s.  
*Infračervený senzor vzdálenosti 20-150 cm GP2Y0A02YK0F*. 2022. [online]. Drátek. [cit. 2022-04-29]. Dostupné na: <https://dratek.cz/arduino/7724-infracervený-senzor-vzdálenosti-20-150-cm>.

*IR Remote Control Theory*. 2021. [online]. SB-Projects. [cit. 2022-04-29]. Dostupné na: <https://www.sbprojects.net/knowledge/ir/index.php>.

*Komunikace*. 2022. [online]. My Arduino. [cit. 2022-04-29]. Dostupné na: <https://www.myarduino.cz/komunikace-rf-wifi-serial-bluetooth>.

KOLÍBAL, Z. 2016. *Roboty a robotizované výrobní technologie*. vyd. Brno: VUTIUM. ISBN 978-80-214-4828-5.

KUBIČKA, M. Nedatováno. *Bluetooth v robotice I*. [online]. Matejk. [cit. 2022-04-29].  
Dostupné na: <http://www.matejk.cz/zdroje/Dil-I-Bluetooth-a-robotika.pdf>.

*L293D - dvoukanálový ovladač pro motory 36V / 0,6A - 5 ks*. 2023. [online]. Botland.cz [cit. 2023-05-16]. Dostupné na: <https://botland.cz/ovladace-stejnosmerneho-motoru/176-l293d-dvoukanalovy-ovladac-pro-motory-36v-06a-5-ks-5904422350253.html>

*Lekce 12 - posuvný registr 74Hc595 :: Arduinio*. Botland. 2023 [online]. Webnode, [cit. 2023-05-15]. Dostupné na: <https://arduino8.webnode.cz/news/lekce-12-posuvny-registr-74hc595/>

MARŠÍK, A., KUBIČÍK, 1980. *M. AUTOMATIZACE – automatické řízení ve strojírenství*. vyd. Praha, SNTL. typové číslo L26-C2-I-01/55536.

- NOVÁK, P. 2005. *Mobilní roboty – pohony, senzory, řízení*. vyd. Praha: BEN – technická literatura, 250 s. ISBN 80-7300-141-1.
- PAŠTYKA, J. 2017. *DC motory řízené pomocí H můstku*. vyd. Praha: České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra měření. 5.s.
- RAJKUMAR, S. CHAKRABORTY, S. DEY, R. 2017. *Development of Embedded Speed Control System for DC Servo Motor using Wireless Communication*.  
10.13140/RG.2.2.13401.06245.
- ŘEŘUCHA, V. KRUPKA, Z. 2000. *Kráčející roboty pro vojenství, průmysl a ekologii*. vyd. Automa – časopis pro automatizační techniku, s. r. o. [online]. [cit. 2022-04-29]. Dostupné z: [https://automa.cz/cz/casopis-clanky/kracejici-roboty-pro-vojenstvi-prumysl-a-ekologii-idea-historie-stav-reseni-a-perspektiva-2000\\_01\\_27547\\_279/](https://automa.cz/cz/casopis-clanky/kracejici-roboty-pro-vojenstvi-prumysl-a-ekologii-idea-historie-stav-reseni-a-perspektiva-2000_01_27547_279/).
- UART Bluetooth moduly*. 2021. [online]. laskarduino. [cit. 2021-11-20]. Dostupné na: <https://blog.laskarduino.cz/uart-bluetooth-moduly/>.
- Ultrazvukový měřič vzdálenosti HC-SR04*. 2022. [online]. Laskakit. [cit. 2022-04-29]. Dostupné na: <https://www.laskakit.cz/arduino-ultrazvukovy-meric-vzdalenosti-hc-sr04>.
- VODA, Z. 2015. *Průvodce světem Arduina*. Vyd. Nakladatelství Martin Stříž, Bučovice. 268 s. ISBN 978-80-87106-90-7.

# **PŘÍLOHY**

**A – CD**

**Příloha k diplomové bakalářské práci**

Mobilní robot vykonávající posloupnost pohybových příkazů

Lukáš Říha

**CD**

## **Obsah**

- 1 Text bakalářské práce ve formátu PDF.
- 2 Úplný zdrojový kód.
- 3 Video jízdy robota.