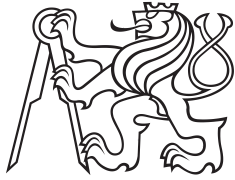


Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra řídicí techniky

## Využití robota LEGO Mindstorms - návrh soutěžní úlohy pro ROBOSOUTĚŽ

Josef Rastočný

Vedoucí: Ing. Martin Hlinovský, Ph.D.  
Obor: Kybernetika a robotika  
Květen 2024



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Rastočný** Jméno: **Josef** Osobní číslo: **499350**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra řídicí techniky**  
Studijní program: **Kybernetika a robotika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Využití robota LEGO Mindstorms - návrh soutěžní úlohy pro ROBOSOUTĚŽ**

Název bakalářské práce anglicky:

**Usage of the LEGO Mindstorms Robots - design of the competition task for ROBORACE**

Pokyny pro vypracování:

1. Seznamte se s možnostmi robota Lego Mindstorms EV3 (současný stav, HW a SW vybavení).
2. Navrhněte a realizujte soutěžní úlohu pro předmět ROBOTI nebo pro ROBOSOUTĚŽ pro LEGO Mindstorms EV3 s návrhem řízení ve Vámi vybraném programovacím prostředí.
3. Úlohou bude 'Satelit na geostacionární oběžné dráze' (robot musí udržovat správnou polohu na oběžné dráze)
4. Vytvořte webové stránky k realizované úloze (podrobný popis a zadání úlohy, návrh a realizace přípravku, principu činnosti, vysvětlení navrženého softwaru a fotogalerii včetně videí).

Seznam doporučené literatury:

- [1] James Floyd Kelly - LEGO MINDSTORMS NXT-G programming Guide, Second Edition
- [2] Daniele Benedettelli - Programming LEGO NXT Robots using NXC
- [3] <https://www.ist.uni-stuttgart.de/de/lehre/roboratione/aufgabe/rr2016aufgabe/>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Martin Hlinovský, Ph.D. katedra řídicí techniky FEL**

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

Datum zadání bakalářské práce: **19.01.2024**

Termín odevzdání bakalářské práce: **24.05.2024**

Platnost zadání bakalářské práce: **21.09.2025**

Ing. Martin Hlinovský, Ph.D.  
podpis vedoucí(ho) práce

prof. Ing. Michael Šebek, DrSc.  
podpis vedoucí(ho) ústavu/katedry

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

## III. PŘEVZETÍ ZADÁNÍ

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

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta



## Poděkování

Chtěl bych poděkovat svému vedoucímu panu Ing. Martinu Hlinovskému, Ph.D. za vypůjčení stavebnic EV3 a poskytnutí soutěžní desky. Dále bych chtěl poděkovat svému otci, za pomoc při výrobě konstrukce.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 24. května 2024

## Abstrakt

Bakalářská práce se zabývá návrhem a realizací soutěžní úlohy pro ROBOSOUTĚŽ. V práci je popsána výroba samotné konstrukce pro soutěžní úlohu, řízení motoru pomocí Arduina a H-můstku, využití stavebnice LEGO Mindstorms EV3 pro výrobu signalizačního obvodu. Dále je zde popsána konstrukce robota a jeho senzory, PID regulátor a rozbor řešení úlohy.

**Klíčová slova:** ROBOSOUTĚŽ, LEGO Mindstorms EV3, Arduino, MicroPython, PID regulátor

**Vedoucí:** Ing. Martin Hlinovský, Ph.D.

## Abstract

Bachelor's thesis deals with the design and implementation of a competition task for ROBOSOUTĚŽ. The thesis describes the construction of the competition task, motor control using Arduino and an H-bridge, and the use of LEGO Mindstorms EV3 for creating a signaling circuit. Furthermore, it describes the construction of the robot and its sensors, the PID controller, and the analysis of the task solution.

**Keywords:** ROBORACE, LEGO Mindstorms EV3, Arduino, MicroPython, PID controller

**Title translation:** Usage of the LEGO Mindstorms Robots - design of the competition task for ROBORACE

## Obsah

<b>1 Úvod</b>	<b>1</b>	3.3.3 PWM .....	17
<b>2 Zadání úlohy</b>	<b>3</b>	3.3.4 Zapojení manuálního ovládání rychlosti .....	18
2.1 Princip úlohy .....	3	3.3.5 Zapojení automatického ovládání rychlosti .....	19
2.2 Konstrukce úlohy .....	4	3.3.6 Program .....	20
2.3 Soutěž .....	5	<b>4 Řešení úlohy</b>	<b>23</b>
<b>3 Výroba úlohy</b>	<b>7</b>	4.1 Stavba robota .....	23
3.1 Konstrukce .....	7	4.1.1 Konstrukce .....	23
3.2 Signalizace .....	9	4.1.2 Ultrazvukový senzor .....	24
3.2.1 EV3 .....	9	4.1.3 Senzor barvy .....	25
3.2.2 Dotykový senzor .....	10	4.1.4 Programování EV3 v pythonu	26
3.2.3 Rozbočovač .....	11	4.2 PID regulátor .....	27
3.2.4 Zapojení .....	12	4.2.1 Proporcionální složka .....	27
3.2.5 Program .....	12	4.2.2 Integrovaná složka .....	27
3.3 Řízení motoru .....	13	4.2.3 Derivační složka .....	28
3.3.1 Arduino uno .....	13	4.3 Rozbor řešení .....	28
3.3.2 H-můstek .....	15	4.3.1 Sledování černé čáry .....	28

4.3.2 Detekování ramene a udržování vzdálenosti .....	29
4.3.3 Integrace obou úloh .....	29
<b>5 Závěr</b>	<b>31</b>
<b>A Literatura</b>	<b>33</b>



## Obrázky

2.1 Předloha úlohy z Univerzity ve Stuttgartu[14]. . . . .	4	3.14 Zapojení pro automatické ovládání rychlosti. . . . .	20
2.2 Rozměry kruhové desky[14]. . . . .	5	4.1 Konstrukce robota. . . . .	23
3.1 Základna s motorem. . . . .	8	4.2 Princip funkce ultrazvukového senzoru[10]. . . . .	24
3.2 Konstrukce úlohy. . . . .	8	4.3 Ultrazvukový senzor[8]. . . . .	24
3.3 Řídící kostka EV3[3]. . . . .	9	4.4 Senzor barvy[9]. . . . .	25
3.4 Dotykový senzor[6]. . . . .	11		
3.5 SPLIT-Nx-v2[7]. . . . .	11		
3.6 Zapojení signalizace. . . . .	12		
3.7 Program pro signalizaci. . . . .	13		
3.8 Arduino Uno[4]. . . . .	14		
3.9 Schéma H-můstku[11]. . . . .	15		
3.10 H-můstek MD25HV[5]. . . . .	16		
3.11 PWM signály a jejich zápis pro Arduino[12]. . . . .	17		
3.12 Schéma zapojení pro manuální ovládání rychlosti[1]. . . . .	19		
3.13 Schéma zapojení pro automatické ovládání rychlosti[2]. . . . .	20		





# Kapitola 1

## Úvod

Tato bakalářská práce s tématem využití robota LEGO Mindstorms se zabývá návrhem soutěžní úlohy a následnou realizací robota schopného udržovat svou polohu na rotující kruhové dráze. Zadání je inspirováno podobnou robosoutěží, která proběhla na Univerzitě ve Stuttgartu v roce 2016. Cílem práce je navigovat robota mezi rotujícími rameny a zůstat v bezpečné vzdálenosti, aniž by opustil určenou oblast nebo se přiblížil k rotujícím ramenům. Práce zahrnuje teoretické i praktické aspekty konstrukce úlohy a programování robota.

V první kapitole jsou podrobně popsány cíle a principy úkolu, inspirovaného satelity na geostacionární oběžné dráze. Úloha se soustředí na udržení polohy robota mezi otáčejícími se rameny na kruhové desce, přičemž je využit odstín šedé barvy na povrchu, sloužící pro orientaci robota.

Druhá kapitola se zaměřuje na konstrukci úlohy, včetně detailního popisu výroby otáčejících se ramen. Podrobně je vysvětlen proces výroby základny, instalace motoru a senzorů, a také výběr materiálů pro konstrukci ramen. Tato část poskytuje čtenáři návod, jak mechanicky realizovat podobnou úlohu, včetně řešení problémů spojených s pohonem a stabilitou konstrukce. Zároveň jsou detailně představeny i elektronické komponenty použité pro řízení motoru. Jsou zde představeny také základní principy fungování H-můstku, PWM signálu a role mikrokontroléru Arduino Uno v celém systému. Rovněž je i znázorněno zapojení a programování motorů pro manuální a automatické ovládání rychlosti a směru otáčení, což je klíčové pro správnou funkci otáčejících se ramen.

V poslední kapitole je rozebrána samotná konstrukce robota, včetně popisu jednotlivých senzorů a jejich funkce. Dále je zde popsán PID regulátor a rozbor řešení úlohy.





## Kapitola 2

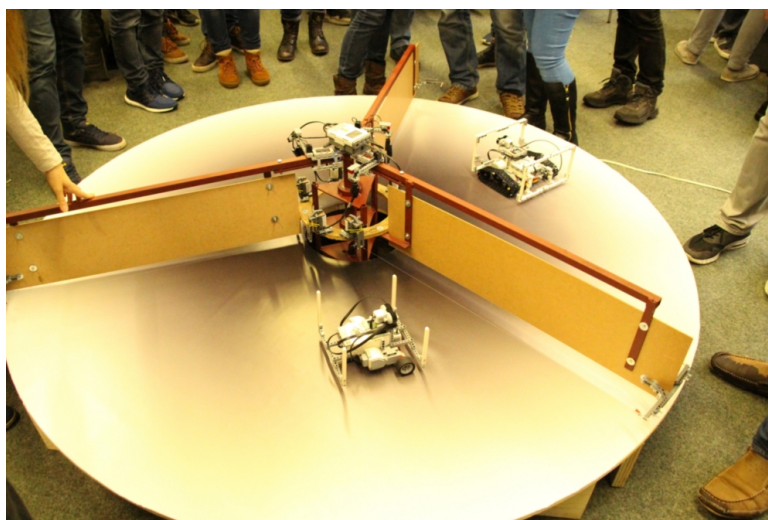
### Zadání úlohy



#### 2.1 Princip úlohy

Zadání je inspirováno satelity na geostacionární oběžné dráze, které musí udržovat svou orbitální polohu pro stabilní provoz. Tento zjednodušený design je koncipován tak, aby zobrazil komplexitu udržování polohy v oběžné dráze satelitu.

Hlavním úkolem je zůstat v oblasti mezi otáčejícími se rameny co nejdéle bez rizika přiblížení k ramenům nebo vyjetí ven z kruhové desky. Pro orientaci je využit odstín šedé barvy na povrchu, s tmavším středem a světlejším okrajem. Otáčení ramen je dynamické, může se měnit během závodu, ale nikdy nedochází k jejich zastavení či obratu.



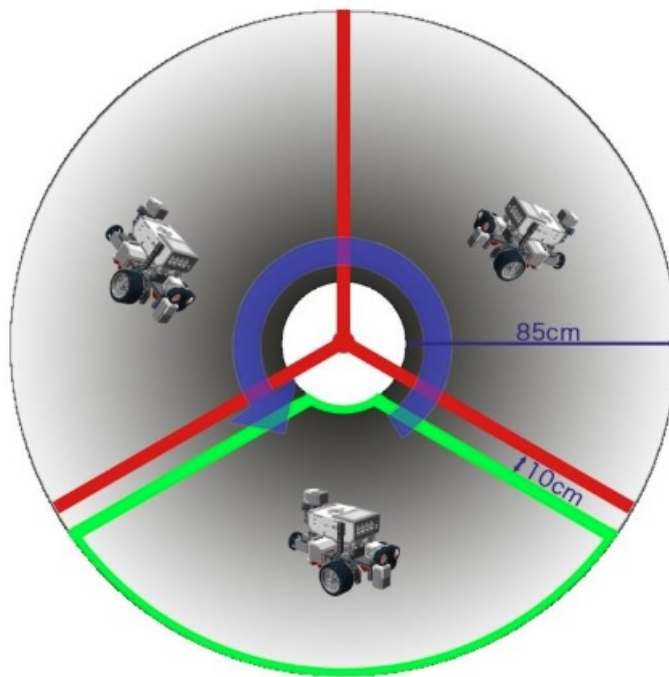
**Obrázek 2.1:** Předloha úlohy z Univerzity ve Stuttgartu[14].

## 2.2 Konstrukce úlohy

Konstrukce se skládá z desky o průměru 200 cm a ze tří otáčejících se ramen. Vnitřek desky o průměru 30 cm je vyříznut tak, aby do něj bylo možné usadit základnu s motorem pro ramena.

Ramena tedy rozdělují trať na tři samostatné sekce, v nichž je možné provozovat jízdu jednoho robota. To umožňuje současnou soutěž tří týmů. Kvůli vyříznuté části desky je efektivní šířka pro jízdu robota omezena na 85 cm. Díky třem ramenům má robot k dispozici oblast o šířce přibližně  $120^\circ$ , ve které může operovat, aniž by při tom s nimi přišel do kontaktu.

Na každém rameni je z obou stran natažené lanko 10 cm vzdálené, které napnutím a pomocí páky sepne senzor. Celkově je zde 6 senzorů, přičemž každému sektoru jsou přiděleny dva. Pod rameny je našroubována kruhová výseč pro umístění senzorů. Nad každým sektorem je umístěn jeden motor pro signalizaci vyřazení ze soutěže.



Obrázek 2.2: Rozměry kruhové desky[14].

## ■ 2.3 Soutěž

Pro určení vítěze je využíván čas, po který robot zůstane v povolené oblasti, aniž by ji opustil. Rozlišuje se mezi kontaktem s lopatkou a opuštěním směrem ven. Jakmile se otáčejí lopatky, začíná se měřit čas. Během celého závodu nesmí robot opustit trať směrem ven, ani dovnitř, a zároveň musí dodržovat minimální vzdálenost 10 cm od otáčejících se křídel. Pokud se robot přiblíží k ramenům blíže než 10 cm, je detekován napnutým lankem propojeným s čidlem, kterým je aktivován motor pro daný sektor. Měření času je tak zastaveno a běh je pro daný tým ukončen. Pokud robot opustí sektor směrem ven (opustí povolenou oblast radiálně), běh je považován za neplatný.





## Kapitola 3

### Výroba úlohy

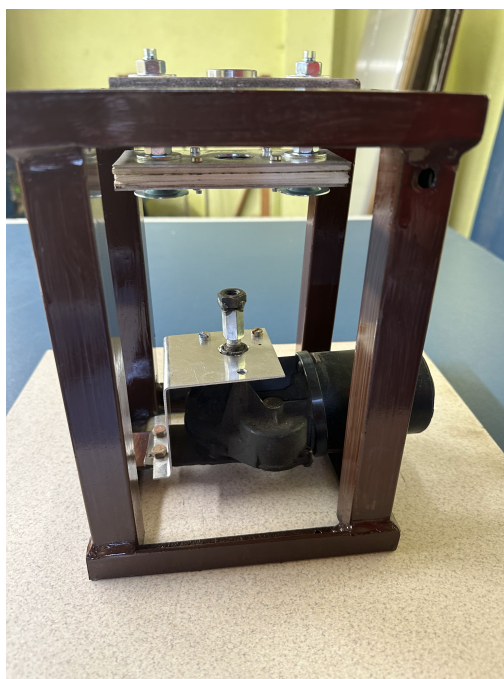
#### 3.1 Konstrukce

Pro vypracování této úlohy jsem měl k dispozici pouze video jako předlohu, pár fotografií a základní rozměry, jako např. průměr desky a poloměr lopatek.

První krok spočíval v návrhu základny pro upevnění motoru a hřídele, která bude otáčet lopatkami. Základna je svařena ze čtyř železných profilů ve tvaru písmene "L", které tvoří čtvercovou základnu pro uchycení motoru. Stojny základny sestávají ze čtyř hranolů z masivního železa. Vzhledem k délce ramen a vyššímu odporu vzduchu musí být základna připevněna k masivní dřevotřískové desce, aby nedošlo k převržení při prudkém startu nebo vypnutí motoru.

Pro pohon byl zvolen motor předních stěračů s převodovkou z vozu Škoda Favorit. Hnací hřídel o průměru 10 mm je uchycena na hřídel motoru pomocí šroubového spoje. K základně je upevněna dvěma ložisky kvůli stabilitě ramen při vyšších otáčkách. Na jejím konci se nachází příruba, na kterou jsou pod úhlem  $120^\circ$  namontována tři ramena.

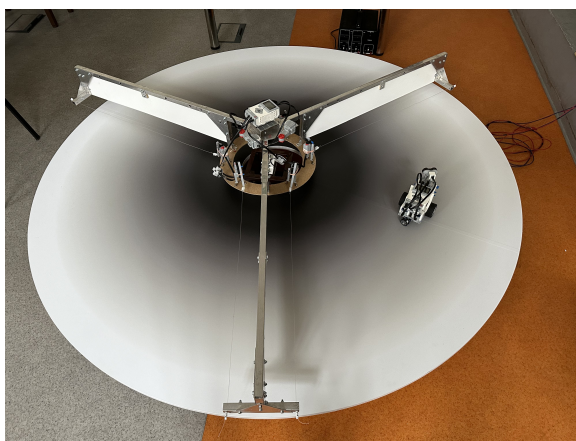
Konstrukce příruby a jednotlivých ramen je tvořena z hliníkového materiálu tak, aby byla minimalizována celková hmotnost a bylo dosaženo požadované maximální rychlosti. K ramenům jsou připevněny plastové desky ve tvaru obdélníku, aby roboti mohli detekovat svou vzdálenost od ramen. Na jejich vnitřním konci je upevněna kruhová výseč, na kterou jsou umístěna jednotlivá čidla s pákovým mechanismem, který je aktivován propnutím lanka, jehož



**Obrázek 3.1:** Základna s motorem.

druhý konec je připevněn na vnější konec ramen. Tento mechanismus čidla a lanka se nachází na obou stranách každého ramen. Celkově je použito šest senzorů, které slouží pro detekci pohybu robota ve správné výšce, a také pro detekci nízké nebo příliš vysoké rychlosti robota.

Řídící kostka EV3 je umístěna pomocí suchého zipu na horní straně příruby. Na okraji příruby jsou v každé výšce připevněny motory pro signalizaci napnutí jednoho z dvou lanek daného sektoru.



**Obrázek 3.2:** Konstrukce úlohy.

## 3.2 Signalizace

Tato podkapitola je zaměřena na popis veškerých komponent, které tvoří signalizační obvod mého projektu. Nejprve bude představena řídicí kostka EV3, která je vybavena procesorem ARM9 a poskytuje vysoký výkon pro programování a rychlé zpracování dat. Další část je věnována praktickému využití dotykového senzoru, který slouží k detekci fyzického kontaktu s robotem a je zásadní pro aktivaci motoru.

Problém s omezeným počtem portů na kostce EV3 je vyřešen pomocí rozbočovače, což umožňuje připojení více senzorů než je standardně dostupné. Kromě teoretického popisu jednotlivých prvků, bude podrobněji vysvětleno i jejich zapojení a programování.

### 3.2.1 EV3

Řídicí kostka EV3 je vybavena procesorem ARM9 s frekvencí 300 MHz, což je významné zlepšení oproti předchozí verzi NXT. S 16 MB vestavěné flash paměti a 64 MB RAM poskytuje kostka dostatek prostoru a výkonu pro složité programy a rychlé zpracování dat. Na přední straně kostky se nachází monochromatický LCD displej, který slouží pro zobrazování informací a umožňuje základní interakci s robotem bez nutnosti dalšího připojení. Zobrazuje informace o stavech programů, senzorová data a další užitečné informace pro ladění a interakci. Pod displejem se nachází řada tlačítek, která umožňují manuální ovládání a navigaci v menu.



Obrázek 3.3: Řídicí kostka EV3[3].

EV3 obsahuje čtyři vstupní a čtyři výstupní porty, které umožňují připojení různých senzorů a motorů. Kostka podporuje také USB port pro připojení Wi-Fi donglu nebo přímé spojení s počítačem, a rovněž je vybavena Bluetooth pro bezdrátové programování a ovládání. Díky této vysoké úrovni konektivity je kompatibilní s mnoha senzory a motory z předchozích verzí, jako je NXT, což usnadňuje integraci a rozšíření stávajících projektů.

LEGO Mindstorms lze programovat v několika prostředích, ale pro můj projekt bylo zvoleno LEGO Mindstorms EV3 a programování v Pythonu. Následně budou popsány obě tyto možnosti, které byly použity.

Oficiální programovací prostředí od LEGO pro EV3 je navrženo s ohledem na intuitivnost a přístupnost, což je ideální volba pro začínající programátory a vzdělávací účely. Umožňuje uživatelům vytvářet programy pomocí vizuálně orientovaného „drag-and-drop“ systému bloků. Každý blok představuje specifické funkce nebo příkazy, jako jsou řízení motorů, čtení senzorů, provádění logických operací a další. Tento způsob programování je nejen snadno pochopitelný, ale také umožňuje rychlé a efektivní prototypování, což je výhodné při výuce základních principů robotiky a programování.

#### ■ 3.2.2 Dotykový senzor

Senzor dotyku z LEGO Mindstorms EV3 funguje jako jednoduchý, ale efektivní mechanismus pro detekci fyzického kontaktu nebo tlaku. Jeho primární funkcí je informovat robota, že došlo k dotyku s objektem, což může být využito pro různé účely, jako je zastavení pohybu, změna trajektorie nebo aktivace nějaké programové sekvence. V mém případě slouží k detekci najetí robota do nataženého lanka, což aktivuje motor daného sektoru.

Senzor dotyku je konstruován s mechanickým tlačítkem na jednom konci, které lze stisknout. Když je tlačítko stlačeno, senzor tento stav zaznamená a vyšle signál do robotického řídicího systému EV3. Ve své podstatě je senzor dotyku binární zařízení, což znamená, že rozlišuje pouze mezi dvěma stavy: "stisknuto" a "nestisknuto".

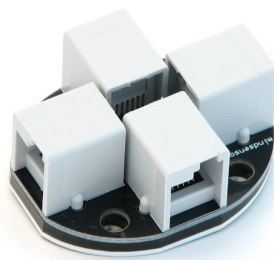


**Obrázek 3.4:** Dotykový senzor[6].

### ■ 3.2.3 Rozbočovač

Pro správnou signalizaci najetí robota do lanka bude zapotřebí 6 dotykových senzorů. Kostka EV3 má ale pouze 4 vstupní porty, proto byl použit rozbočovač. Split NX v2 zvyšuje počet dostupných portů na EV3 kostce, což umožňuje připojení více senzorů nebo motorů. Tento modul rozdělí jeden port na EV3 na 3 další porty, čímž rozšíří kapacitu pro připojení dalších komponent.

Rozbočovač je navržen pro kompatibilitu se všemi běžnými senzory a aktuá-



**Obrázek 3.5:** SPLIT-Nx-v2[7].

tory používanými v systémech LEGO Mindstorms EV3, včetně komponent od třetích stran. Kritickou součástí správy více senzorů na jednom portu je správná konfigurace I2C adres. Split NX v2 automaticky rozpoznává a

řídí I2C adresy připojených senzorů, což zajišťuje, že každý senzor může komunikovat s EV3 brickem bez konfliktů na sběrnici. Pro signalizaci ale nebylo této funkce využito. Bylo totiž dostačující použít pro každý sektor jeden rozbočovač, na který byly následně připojeny 2 dotykové senzory a nebylo tak rozlišováno, který z těchto dvou senzorů byl sepnut.

#### ■ 3.2.4 Zapojení

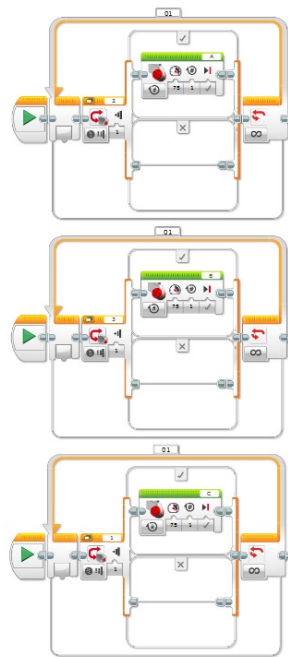
Zapojení je velice jednoduché, 3 motory jsou zapojeny do portů A,B a C. Dále do portů pro senzory 1, 2 a 3 jsou zapojeny rozbočovače, do kterých jsou zapojeny 2 senzory.



Obrázek 3.6: Zapojení signalizace.

#### ■ 3.2.5 Program

Na obrázku je vidět jednoduchý program pro správnou signalizaci. Skládá se ze tří samostatných bloků, každý ovládá jeden motor, který je spuštěn pomocí jednoho ze dvou senzorů příslušného portu. Jakmile pojede robot příliš rychle nebo naopak příliš pomalu, narazí do nataženého lanka, čímž pomocí pákového mechanismu sepnou dotykové senzory, a tím aktivují motor pro daný sektor, který signalizuje konec daného robota v závodu.



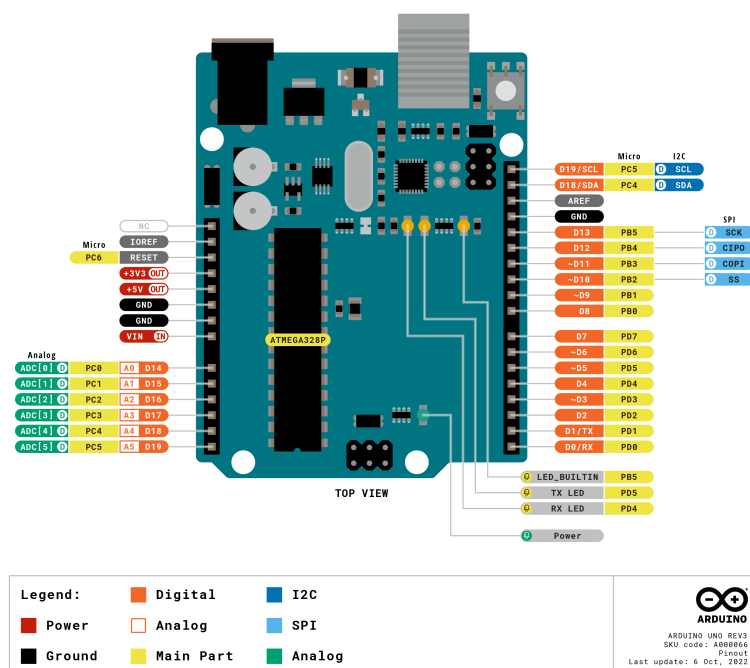
Obrázek 3.7: Program pro signalizaci.

## 3.3 Řízení motoru

Tato kapitola je zaměřena na jednotlivé komponenty použité pro řízení motoru. Budou podrobně popsány součástky, které umožňují efektivní manipulaci s pohybem a chodem motorů, a to jak v oblasti jejich rychlosti, tak směru otáčení. Poté bude představen mikrokontrolér Arduino Uno, který slouží jako mozek celého řídicího systému. Další podstatnými komponenty jsou H-můstek, který je nezbytný pro změnu směru otáčení motorů, a také principy PWM (Pulzně Šírková Modulace), které hrají klíčovou roli v regulaci rychlosti motorů. Kromě teoretického popisu těchto prvků, dojde i na praktické zapojení a programování, což umožňuje ověřit funkčnost a správný chod motoru pro soutěžní úlohu.

### 3.3.1 Arduino uno

Arduino Uno je otevřená hardwarová platforma založená na jednoduchém mikrokontroléru ATmega328 od společnosti Atmel. Je navržena s myšlenkou snadného použití pro amatéry i profesionály v oblasti elektroniky. Uno je nejpopulárnější model z řady Arduino a je často používán pro vzdělávací účely, prototypování a DIY (do-it-yourself) projekty.



Obrázek 3.8: Arduino Uno[4].

Arduino Uno je vybaveno mikrokontrolérem ATmega328P a nabízí 14 digitálních I/O pinů, z nichž 6 lze použít jako PWM výstupy pro řízení například motorů. K dispozici je také 6 analogových vstupů, které umožňují připojení různých senzorů. Celková paměť dosahuje 32 KB, z čehož 0,5 KB zabírá bootloader. Taktovací frekvence mikrokontroléru je 16 MHz, což zajišťuje dostatečný výkon pro většinu základních a středně náročných aplikací.

Arduino Uno se programuje prostřednictvím Arduino IDE (Integrated Development Environment), což je speciálně navržený software, který umožňuje psaní, testování a nahrávání programů na Arduino desku. Uživatelé píší kód v jazyce podobném C++, který IDE přeloží do strojového jazyka kompatibilního s mikrokontrolérem ATmega328P. Tento proces kompilace zahrnuje převod zdrojového kódu na spustitelný program, který je poté nahrán do paměti mikrokontroléru prostřednictvím USB rozhraní.

Jakmile je program nahrán, mikrokontrolér provádí napsané instrukce a interaguje s vnějším světem skrze své I/O piny. Tyto piny lze využít pro čtení vstupů z různých senzorů nebo pro ovládání externích zařízení, jako jsou motory, LED diody, a další. Díky své taktovací frekvenci 16 MHz je Arduino Uno schopné zpracovávat instrukce rychle a efektivně, což umožňuje jeho použití v široké škále aplikací.

Arduino Uno je oblíbený pro řízení DC motorů, což využívá jeho schopnost generovat PWM (Pulzně Šířková Modulace) signály, které efektivně



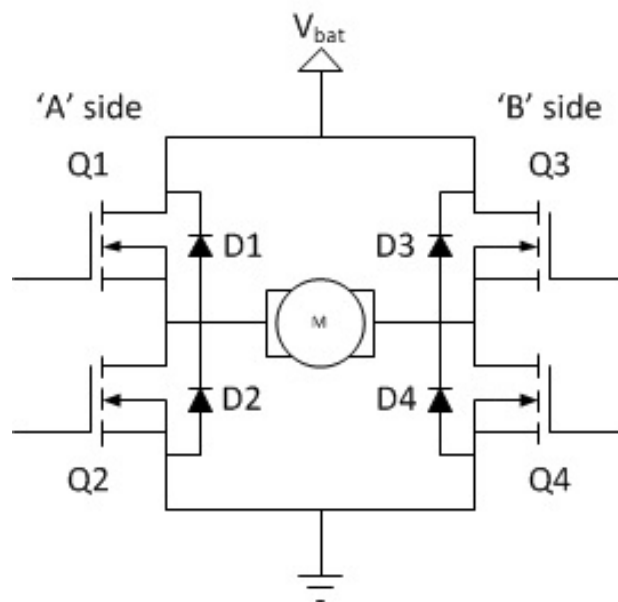
regulují rychlost motoru. Tyto signály umožňují kontrolovat množství energie dodávané motoru, což je klíčové pro aplikace, kde je zapotřebí přesné nastavení rychlosti nebo síly motoru.

Při použití Arduino k řízení DC motoru je obvykle nutné zapojit motorový ovladač, který funguje jako rozhraní mezi motorem a Arduino deskou, chrání desku před případnými škodami a zajišťuje dostatečný výkon pro motor. Proces řízení motoru začíná napsáním kódu, ve kterém uživatel definuje, jaké PWM signály mají být generovány v závislosti na požadované rychlosti motoru. Tento kód je poté nahrán do Arduina, které podle něj řídí motorový ovladač.

### 3.3.2 H-můstek

H-můstek je elektronický obvod používaný k řízení směru a rychlosti stejnosměrných (DC) motorů. Tento obvod je pojmenován podle svého schématu, které připomíná písmeno "H". H-můstek umožňuje motoru pracovat ve dvou směrech (vpřed a vzad) a řídí rychlost otáčení motoru pomocí modulace šířky pulzu (PWM).

Základní funkce H-můstku spočívá v tom, že umožňuje měnit směr proudu,



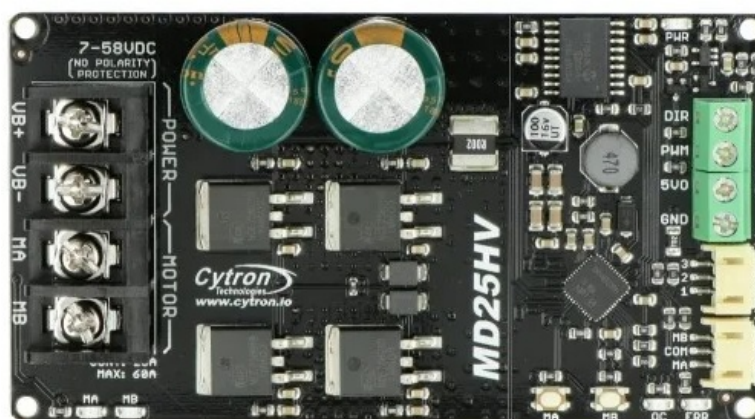
Obrázek 3.9: Schéma H-můstku[11].

který teče přes DC motor, což vede ke změně směru otáčení motoru. Tento

obvod využívá čtyř tranzistorů (nebo jiných spínacích prvků), které jsou spojeny tak, aby vytvářely můstkový obvod. Tímto způsobem je možné řídit, které dva tranzistory jsou otevřeny, a které jsou uzavřeny v závislosti na požadovaném směru otáčení motoru. Pro řízení rychlosti motoru H-můstek využívá PWM signály, které budou popsány v další podkapitole.

Pro mou aplikaci byl zvolen můstek MD25HV. Tento můstek je navržen pro vysoké napěťové (7V-58V) aplikace, je vybaven čtyřmi výkonnými tranzistory a poskytuje robustní a spolehlivé řešení pro ovládání DC motorů.

Jednou z významných vlastností tohoto můstku je jeho schopnost ochrany



Obrázek 3.10: H-můstek MD25HV[5].

proti přetížení a přehřátí. Můstek je vybaven ochrannými mechanismy, které chrání tranzistory před poškozením v důsledku přetížení nebo nadměrného zahřátí, což zvyšuje jeho spolehlivost a životnost. Přepětová ochrana chrání motor před poškozením v případě jeho zastavení nebo při připojení motoru s nadměrným výkonem. V situaci, kdy by motor vyžadoval více proudu než je podporováno řídicí jednotkou motoru, je proud omezen na maximální možnou hodnotu. Dokáže nepřetržitě podporovat proud 25A (peak 60A) bez nutnosti přidání dalšího chladiče.

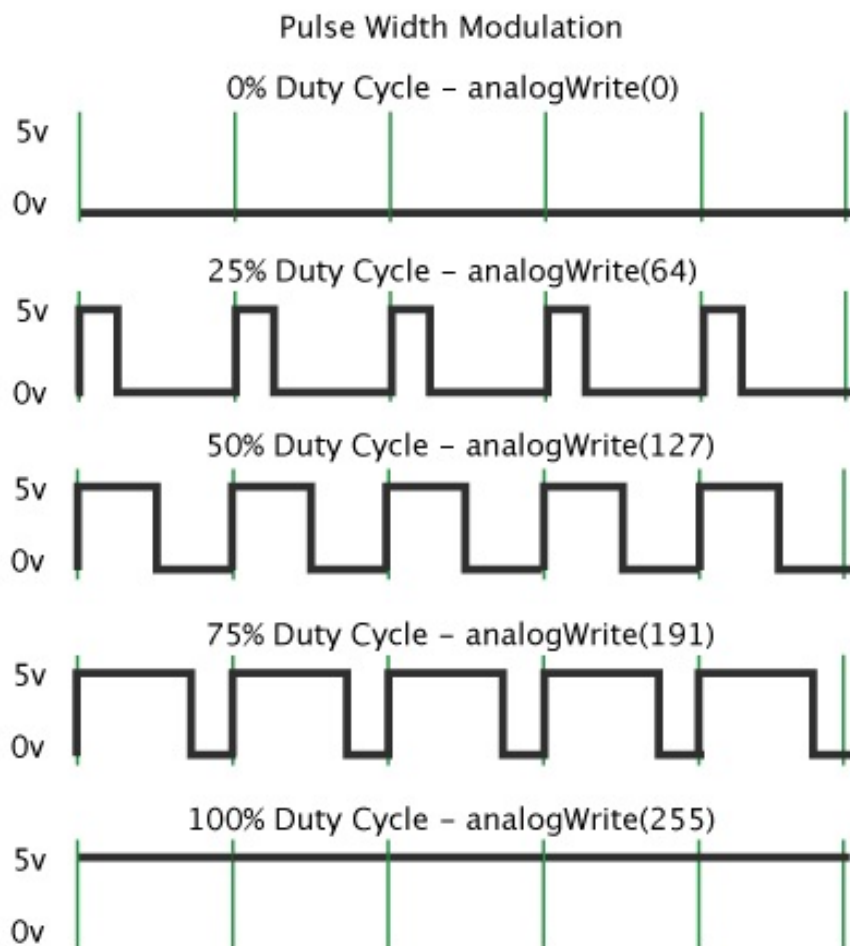
Dále můstek disponuje teplotní ochranou, což znamená, že maximální omezení proudu je upraveno v závislosti na teplotě desky. S rostoucí teplotou desky se snižuje maximální hodnota proudu, což umožňuje můstku optimalizovat své výkonné možnosti podle aktuálních podmínek a zabránit tak poškození MOSFET tranzistorů.

### 3.3.3 PWM

Princip PWM (Pulse Width Modulation) je metoda pro regulaci průměrného výkonu dodávaného elektrickému zařízení tím, že se periodicky mění šířka pulzů signálu. V PWM se využívá cyklického přepínání mezi dvěma stavy, "zapnuto"(HIGH) a "vypnuto"(LOW), což vytváří pulzní signál.

Základní princip PWM je založen na variaci šířky impulzů během jedné periody. Šířka impulsu je časový interval, po který je signál v "zapnutém" stavu během jednoho cyklu. Délka jednoho cyklu se nazývá perioda, a čím delší je šířka impulsu v rámci tohoto cyklu, tím vyšší je průměrná hodnota výstupního signálu.

PWM je velmi užitečná pro řízení stejnosměrných motorů. Při použití



**Obrázek 3.11:** PWM signály a jejich zápis pro Arduino[12].

PWM pro tyto motory se upravuje střída signálu, což efektivně mění napětí, které je aplikováno na motor. Tím se umožňuje regulace rychlosti motoru bez potřeby použití složitých a energeticky neefektivních analogových řídicích systémů.

Viz obrázek zobrazuje PWM signály v různých střídacích. Signály jsou vyjádřeny v pěti různých střídacích: 0%, 25%, 50%, 75% a 100%. Pro Arduino můžeme tyto signály generovat použitím funkce `analogWrite` (`pin`, `value`), kde `pin` je číslo pinu, který podporuje PWM, a `value` je hodnota od 0 do 255, která určuje šířku pulsu. Hodnoty uvedené na obrázku představují:

- 0% střída signálu = `analogWrite(0)` - výstupní napětí je stále 0V.
- 25% střída signálu = `analogWrite(64)` - signál je LOW 75% času a HIGH 25% času.
- 50% střída signálu = `analogWrite(127)` - signál je LOW 50% času a HIGH 50% času.
- 75% střída signálu = `analogWrite(191)` - signál je LOW 25% času a HIGH 75% času.
- 100% střída signálu = `analogWrite(255)` - výstupní napětí je stále 5V.

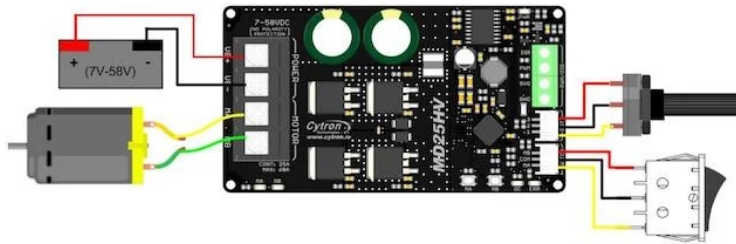
Tento způsob regulace rychlosti motoru je velmi efektivní, protože minimalizuje ztráty energie, které jsou typické pro rezistorové nebo tranzistorové regulátory napětí, jež generují značné množství tepla. PWM tak poskytuje efektivní a energeticky úsporné řešení pro ovládání stejnosměrných motorů v širokém spektru aplikací, od malých hobby projektů až po velké průmyslové stroje.

#### ■ 3.3.4 Zapojení manuálního ovládání rychlosti

Nejprve byla provedena zkouška manuálního ovládání rychlosti motoru pomocí potenciometru, aby byla ověřena funkčnost motoru. Zároveň je možné toto zapojení využít pro testování jízdy robota.

Následující obrázek znázorňuje zapojení H-můstku, potenciometru a tlačítka, které umožňuje manuální ovládání motoru.

### Mode 2: Potentiometer Control



**Obrázek 3.12:** Schéma zapojení pro manuální ovládání rychlosti[1].

Potenciometr je zde využit k manuálnímu nastavení rychlosti motoru, kde se jeho otáčením mění odpor. Třípolohové tlačítko umožňuje volbu mezi směrem otáčení motoru dopředu, vypnutím motoru a otáčením dozadu.

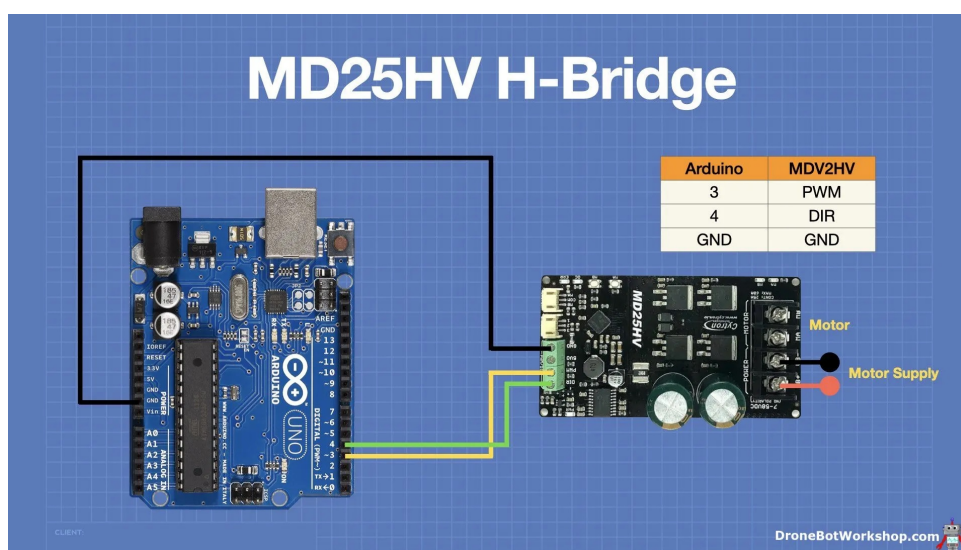
Tímto zapojením bylo otestováno, zda při startu nedochází k zásekům, a zda je zrychlování motoru plynulé. Současně bylo využito příležitosti za účelem prověření maximální rychlost motoru.

### 3.3.5 Zapojení automatického ovládání rychlosti

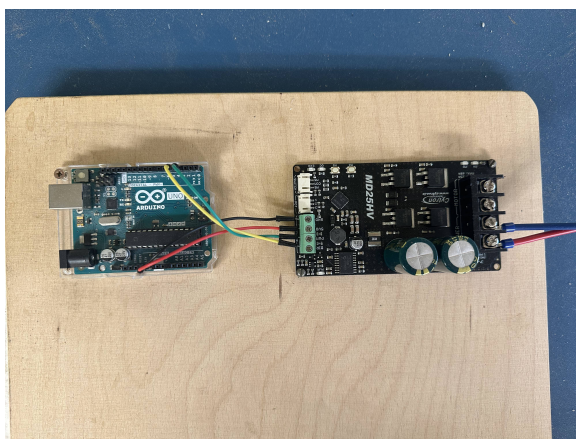
Na obrázku je patrné schéma zapojení, které spojuje Arduino Uno, MD25HV H-můstek a motor. Jedná se o typickou aplikaci v robotice pro řízení rychlosti a směru otáčení motoru.

Arduino Uno je mikrokontrolér, který slouží jako mozek celého projektu, jelikož poskytuje nezbytné řídicí signály pro modul H-můstku. Arduino je schopné generovat PWM signály pro regulaci rychlosti motoru a digitální signály pro určení směru otáčení.

V zapojení, pin 3 na Arduino je připojen na PWM vstup H-můstku. Tento výstup z Arduino je odpovědný za šířku pulsu signálu, která určuje rychlost motoru — delší pulsy znamenají vyšší rychlost. Pin 4 na Arduino je použit pro připojení na DIR (Direction) vstup na modulu H-mostu, který řídí směr otáčení motoru podle toho, jestli je signál v logické jedničce (HIGH) nebo v logické nule (LOW). Společná zem (GND), spojující Arduino a H-můstek, je



Obrázek 3.13: Schéma zapojení pro automatické ovládání rychlosti[2].



Obrázek 3.14: Zapojení pro automatické ovládání rychlosti.

důležitá pro stabilní a spolehlivé fungování systému, neboť zajišťuje, že obě zařízení mají stejný referenční bod pro všechny signály.

Celkově toto zapojení umožňuje efektivně ovládat jak rychlost, tak směr otáčení motoru, což je zásadní pro moji aplikaci.

### ■ 3.3.6 Program

Pro programování v Arduino IDE prostředí byla použita knihovna Cytron, která je navržena speciálně pro ovládání různých motorů s použitím Cytron motorových ovladačů, jako je MD25HV. Tato knihovna poskytuje jednoduchý

a efektivní způsob, jak implementovat ovládání rychlosti a směru motoru pomocí funkcí, jako je `setSpeed()`, což umožňuje nastavit rychlost motoru v rozmezí od -255 do 255, kde kladné hodnoty způsobují otáčení motoru vpřed a záporné vzad.

```

1 #include "CytronMotorDriver.h"
2
3 // Configure the motor driver.
4 CytronMD motor(PWM_DIR, 3, 4); // PWM = Pin 3, DIR = Pin
   4.
5
6 // The setup routine runs once when you press reset.
7 void setup() {
8   // Initialization code here
9 }
10
11 // The loop routine runs over and over again forever.
12 void loop() {
13   for (int i = 30; i < 120; i += 3) {
14     motor.setSpeed(i);
15     delay(5000);
16   }
17   for (int i = 120; i > 30; i -= 3) {
18     motor.setSpeed(i);
19     delay(5000);
20   }
21   motor.setSpeed(0)
22 }

```

**Listing 3.1:** Arduino Motor Control

Na začátku kódu je ovladač motoru konfigurován s použitím dvou pinů Arduina - pin 3 pro PWM signál, který řídí rychlost motoru, a pin 4 pro DIR signál, který řídí směr otáčení motoru. Tato konfigurace je definována vytvořením instance motor objektu `CytronMD` s parametry (PWM DIR, 3, 4). To znamená, že motor bude ovládán pomocí PWM a směru DIR.

Funkce `setup()` se spustí jednou po resetování nebo zapnutí Arduina. V tomto příkladu je tato funkce prázdná, protože veškerá konfigurace byla provedena již při inicializaci.

Funkce `loop()` obsahuje kód, který se opakuje neustále. V této smyčce je definovaný cyklus `for`, který postupně zvyšuje rychlost motoru od 0 do 150 v krocích po 10 jednotkách. Pro každou nastavenou rychlost motor pracuje po dobu 5 sekund (`delay(5000)`) než se rychlost zvýší o dalších 10 jednotek.





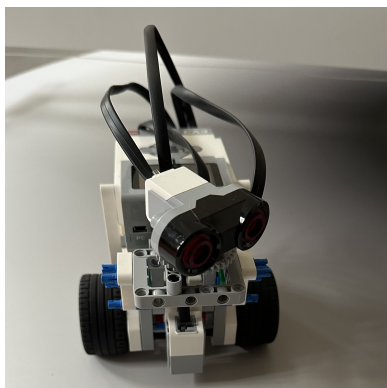
## Kapitola 4

### Řešení úlohy

#### 4.1 Stavba robota

##### 4.1.1 Konstrukce

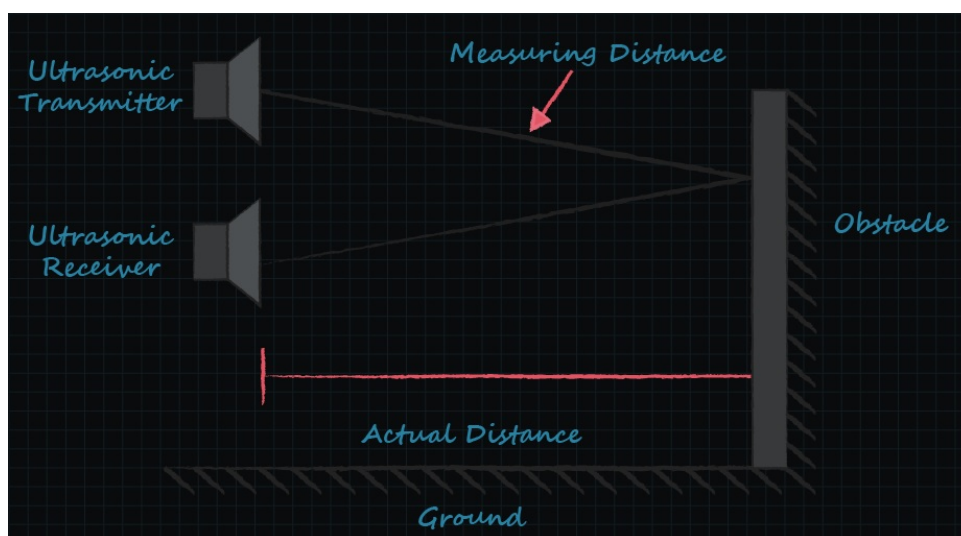
Konstrukce postaveného robota je velice jednoduchá. Řídící kostka je připevněna na podvozku, který se skládá ze dvou větších motorů, které pohání přední kola. Dále je vzadu nepoháněné vlečné kolečko. Robot je vybaven ultrazvukovým senzorem pro sledování vzdálenosti od ramene a senzorem barev pro správnou navigaci po kruhové desce.



Obrázek 4.1: Konstrukce robota.

### 4.1.2 Ultrazvukový senzor

Ultrazvukový senzor z LEGO Mindstorms EV3 pomáhá robotům detekovat objekty ve svém okolí měřením vzdálenosti pomocí ultrazvukových vln. Má dosah detekce od 1 cm do 250 cm a poskytuje data s velmi nízkou latencí, což je ideální pro různé aplikace, jako je navigace, vyhýbání se překážkám nebo sledování pohybujících se objektů. Senzor pracuje na principu vysílání krátkých zvukových pulzů, které se odrazí zpět od překážky a vrátí se k senzoru. Čas, který trvá odrazu vlny zpět, se používá k výpočtu vzdálenosti od objektu.



Obrázek 4.2: Princip funkce ultrazvukového senzoru[10].



Obrázek 4.3: Ultrazvukový senzor[8].

Ultrazvukové senzory obecně obsahují dva hlavní komponenty: vysílač

a přijímač. Vysílač produkuje vysokofrekvenční zvukové vlny, které jsou směřovány do prostředí. Po nárazu na překážku se tyto vlny odrážejí zpět a jsou zachyceny přijímačem. Vzdálenost k objektu se poté určuje pomocí metody zvané "time of flight" (čas letu), která měří interval mezi vysláním ultrazvukového signálu a přijetím jeho odrazu. Spočítá se pomocí následujícího vzorce:

$$d = \frac{v \cdot t}{2} \quad (4.1)$$

kde  $v$  je rychlost paprsku a  $t$  je čas letu.

Odrzivost ultrazvukového signálu je klíčová pro efektivitu senzoru. Nejlepší odrazy poskytují materiály s hladkými, pevnými a nepropustnými povrchy, jako je kov nebo sklo. Tyto materiály umožňují vlnám odrážet se čistě zpět k senzoru bez rozptylu. Na druhou stranu, materiály, které jsou měkké, propustné nebo strukturované, jako je tkanina nebo pěna, mohou absorbovat ultrazvukové vlny nebo je rozptýlovat, což způsobuje slabší nebo rozptýlený odraz. To může vést k méně přesným měřením nebo dokonce k neschopnosti detekce objektu.

### ■ 4.1.3 Senzor barvy

Senzor barev z LEGO Mindstorms EV3 je multifunkční zařízení, které může rozpoznávat až sedm různých barev a zároveň měřit intenzitu světla. Senzor je schopen detekovat černou, modrou, zelenou, žlutou, červenou, bílou a hnědou barvu, což umožňuje robotům interagovat s barevně kódovaným prostředím. To je užitečné například pro sledování čar nebo rozpoznávání specifických objektů. Kromě rozpoznávání barev má senzor také režim detekce světla, kterým lze měřit úroveň osvětlení v okolí, což je výhodné pro úkoly, které se mění v závislosti na světelných podmínkách.



Obrázek 4.4: Senzor barvy[9].

Senzory barev, které jsou použity v EV3, pracují na principu emitování světla na objekt a následného měření intenzity a barvy světla odráženého zpět. Senzor je vybaven LED diodou, která osvětluje objekt před ním. Osvětlený objekt odráží světlo zpátky k senzoru, kde fotodioda nebo barevný senzor analyzuje světlo podle jeho barvy. Intenzita a barva odráženého světla jsou poté analyzovány na základě předem nastavených hodnot, díky kterým je senzor schopen rozlišit mezi různými barvami.

Senzor používá různé vlnové délky světla pro různé barvy, a tím zajišťuje přesnost v detekci. V režimu měření intenzity světla senzor pracuje podobně, ale je zaměřen na množství světla odráženého od objektu bez ohledu na jeho barvu. To je obzvláště užitečné pro aplikace, kde je důležitá změna osvětlení, nebo kde barva nemusí být klíčovým faktorem, ale spíše jas světla.

#### ■ 4.1.4 Programování EV3 v pythonu

MicroPython je úsporná verze Pythonu navržená pro běh na mikrokontrolérech a embedded systémech. Umožňuje uživatelům psát jednoduchý a efektivní kód, který přímo ovládá hardware.

Pro programování robota pomocí Pythonu musela být nejprve vytvořena bootovací microSD karta. Tento krok byl nezbytný, protože standardní firmware EV3 neumí běžet s Pythonem. Prvotní krok spočíval ve stažení firmware MicroPython z oficiálních stránek LEGO, který je speciálně navržen pro programování EV3 s Pythonem.

K vytvoření bootovací SD karty byl použit program Rufus, neboli nástroj, který umožňuje snadno přenést obraz operačního systému na USB nebo SD média. Po vložení prázdné microSD karty do počítače byl v programu Rufus vybrán stažený obraz firmware a zahájen proces zápisu. Rufus efektivně převedl obraz systému na kartu, čímž jsem ji připravil pro použití v mém EV3 Bricku.

Po vložení bootovací SD karty do slotu na EV3 Brick bylo zařízení restartováno. Při spuštění Brick byl automaticky načten nový MicroPython firmware z SD karty. Následně bylo možné připojit EV3 Brick k mému počítači pomocí USB kabelu a začít programovat robota pomocí Pythonu.

## ■ 4.2 PID regulátor

PID regulátor je zpětnovazební regulátor používaný v automatizaci a řízení systémů. PID regulátor se skládá ze tří základních složek: proporcionální (P), integrační (I) a derivační (D). Každá z těchto složek přispívá k výsledné regulační akci jiným způsobem.

PID regulátor lze popsat následující diferenciální rovnicí:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (4.2)$$

### ■ 4.2.1 Proporcionální složka

Proporcionální složka poskytuje regulační akci úměrnou aktuální chybě. Je navržena tak, aby okamžitě reagovala na změny chyby. Pokud je chyba velká, proporcionální složka vytvoří velkou regulační akci, a pokud je chyba malá, regulační akce bude malá.

$$u_P(t) = K_p e(t) \quad (4.3)$$

Proporcionální složka snižuje chybu, ale samotná nemusí být schopna odstranit trvalou ustálenou chybu (offset).

### ■ 4.2.2 Integrační složka

Integrační složka poskytuje regulační akci úměrnou akumulované chybě v čase. Je navržena tak, aby eliminovala trvalou ustálenou chybu, kterou proporcionální složka sama nemůže odstranit.

$$u_I(t) = K_i \int_0^t e(\tau) d\tau \quad (4.4)$$

Integrační složka pomáhá odstranit trvalou chybu, ale může způsobit pomalou reakci systému nebo oscilace, pokud je příliš vysoká

### ■ 4.2.3 Derivační složka

Derivační složka poskytuje regulační akci úměrnou rychlosti změny chyby. Tato složka je navržena tak, aby předpovídala budoucí chování chyby a podle toho upravila regulační akci.

$$u_D(t) = K_d \frac{de(t)}{dt} \quad (4.5)$$

Derivační složka zlepšuje stabilitu systému a tlumí oscilace, ale může způsobit šum a nestabilitu, pokud je příliš vysoká.

## ■ 4.3 Rozbor řešení

Pro řešení úlohy, ve které robot s dvěma motory, vybavený senzorem barvy a ultrazvukovým senzorem, musí jezdit po kruhové desce a zároveň “dohánět” točící se lopatku, je možné úlohu rozdělit na dvě dílčí části: sledování černé čáry a detekování ramene a udržování stabilní vzdálenosti od něj.

### ■ 4.3.1 Sledování černé čáry

První úloha zahrnuje sledování kružnice na desce, která je vyjádřena změnou intenzity světla zachyceného senzorem barvy. Protože deska je směrem dovnitř šedá a směrem ven bílá, může senzor barvy fungovat jako senzor světla. Hodnoty ze senzoru mohou být v rozmezí 0-100, kde 0 představuje nejtmaší hodnotu a 100 nejsvětlejší.

Nejprve je nutné kalibrovat senzor tak, aby správně rozpoznával přechod mezi šedou a bílou částí desky. To znamená zaznamenání referenčních hodnot pro šedou a bílou oblast.

Robot by měl neustále monitorovat hodnotu světla ze senzoru. Pokud hodnota klesne pod určitý práh (odpovídající šedé oblasti), znamená to, že se robot nachází uvnitř kružnice. Pokud hodnota překročí tento práh, robot se nachází mimo kružnici.

Na základě hodnoty senzoru může robot upravovat rychlost a směr jednotlivých motorů tak, aby se udržoval na hranici mezi šedou a bílou oblastí.

### ■ 4.3.2 Detekování ramene a udržování vzdálenosti

Druhá úloha zahrnuje detekci točícího se ramene pomocí ultrazvukového senzoru a udržování stabilní vzdálenosti od něj. Rameno se točí po kružnici, a robot se musí pohybovat tak, aby udržoval určitou vzdálenost od ramene.

Ultrazvukový senzor bude neustále měřit vzdálenost k nejbližšímu objektu. Pokud detekuje rameno, zaznamená tuto vzdálenost. Na základě změřené vzdálenosti od ramene robot upraví svou rychlost. Pokud je vzdálenost příliš malá (robot je příliš blízko ramene), zpomalí. Pokud je vzdálenost příliš velká (robot zaostává), zrychlí. Rychlost obou motorů bude upravována synchronizovaně, aby robot mohl zrychlovat nebo zpomalovat bez vychýlení ze své trasy.

### ■ 4.3.3 Integrace obou úloh

Obě úlohy je třeba integrovat do jednoho systému, který bude schopen současně sledovat černou čáru a udržovat správnou vzdálenost od ramene. Robot bude muset paralelně zpracovávat data ze senzoru barvy i ultrazvukového senzoru. To znamená, že řídicí systém musí být navržen tak, aby dokázal dynamicky reagovat na změny v obou hodnotách.







## Kapitola 5

### Závěr

Tato práce se zaměřila na návrh soutěžní úlohy pro Robosoutěž, inspirované úlohou z Roborace Univerzity ve Stuttgartu z roku 2016. Cílem bylo vytvořit úlohu, která by prověřila schopnosti účastníků v oblasti robotiky a automatizace. Smyslem této úlohy bylo simulovat výzvy spojené s udržováním dráhy geostacionární družice, kde je klíčové udržet stabilní polohu a trajektorii v dynamickém prostředí.

V rámci této práce byla podrobně popsána konstrukce soutěžní úlohy, zahrnující nejen mechanické aspekty, jako je výroba a montáž otáčejících se ramen a základny, ale také elektronické komponenty pro řízení motoru a signalizační obvod pomocí LEGO Mindstorms EV3. Detailní popis zahrnuje také zapojení Arduina a H-můstku pro režim manuálního a automatického ovládání rychlosti. V práci je také popsána integrace senzorů a jejich propojení s řídicí jednotkou EV3, což umožnilo spolehlivé detekování příliš velké nebo nízké rychlosti robota.

Dále je v práci popsána konstrukce samotného robota, včetně použitých senzorů, jako jsou ultrazvukový senzor pro měření vzdálenosti a senzor barev pro navigaci. Byl představen PID regulátor, který umožňuje optimalizaci pohybu robota. Na závěr je v práci uveden rozbor řešení úlohy.



# Příloha A

## Literatura

- [1] RPishop.cz, Accessed on: 21.5.2024 [Online]. Available: <https://rpishop.cz/ridici-desky-robotu/1778-cytron-25amp-7v-58v-high-voltage-dc-motor-driver.html>
- [2] dronebotworkshop, Accessed on: 21.5.2024 [Online]. Available: [https://dronebotworkshop.com/dc-motor-drivers/#Cytron\\_MD25HV\\_Single\\_H-Bridge\\_Controller](https://dronebotworkshop.com/dc-motor-drivers/#Cytron_MD25HV_Single_H-Bridge_Controller)
- [3] LEGO, Accessed on: 21.5.2024 [Online]. Available: [https://www.lego.com/cdn/cs/set/assets/bltbef4d6ce0f40363c/LMSUser\\_Guide\\_LEGO\\_MINDSTORMS\\_EV3\\_11\\_Tablet\\_ENUS.pdf](https://www.lego.com/cdn/cs/set/assets/bltbef4d6ce0f40363c/LMSUser_Guide_LEGO_MINDSTORMS_EV3_11_Tablet_ENUS.pdf)
- [4] Arduino, Accessed on: 21.5.2024 [Online]. Available: <https://store.arduino.cc/products/arduino-uno-rev3>
- [5] botland.cz, Accessed on: 21.5.2024 [Online]. Available: <https://botland.cz/ovladace-motoru-highpower/14813-cytron-md25hv-jednokanalovy-budic-pro-stejnosmerne-motory-7v-58v-25a-590442234.html>
- [6] ruzovka.cz, Accessed on: 21.5.2024 [Online]. Available: <https://ruzovka.cz/cs/2-stupen-zs-vxs/16974-lego-mindstorms-45507-ev3-dotykovy-sensor.html>
- [7] mindsensors.com, Accessed on: 21.5.2024 [Online]. Available: <http://www.mindsensors.com/ev3-and-nxt/52-port-splitter-for-nxt-digital-sensors>

- [8] ruzovka.cz, Accessed on: 21.5.2024 [Online]. Available: [https://ruzovka.cz/cs/2-stupen-zs-vxs/16971-lego-mindstorms-45504-ev3-ultrazvukovy-sensor.html?gad\\_source=1&gclid=Cj0KCQjw\\_-GxBhC1ARIsADGgDjusQUIRgX4bA-CiMAGoSQGQWjPpAGV1TwL8FWPOB8aEk2dZ4aAqoUEALw\\_wcB](https://ruzovka.cz/cs/2-stupen-zs-vxs/16971-lego-mindstorms-45504-ev3-ultrazvukovy-sensor.html?gad_source=1&gclid=Cj0KCQjw_-GxBhC1ARIsADGgDjusQUIRgX4bA-CiMAGoSQGQWjPpAGV1TwL8FWPOB8aEk2dZ4aAqoUEALw_wcB)
- [9] ruzovka.cz, Accessed on: 21.5.2024 [Online]. Available: [https://ruzovka.cz/cs/2-stupen-zs-vxs/16973-lego-mindstorms-45506-ev3-color-sensor.html?gad\\_source=1&gclid=Cj0KCQjw\\_-GxBhC1ARIsADGgDjuixbq-SDIY28XYV40VYxevr4YEUL\\_wBhEDUaYo2oEDBVB48Wh9e4aAoxeEALw\\_wcB](https://ruzovka.cz/cs/2-stupen-zs-vxs/16973-lego-mindstorms-45506-ev3-color-sensor.html?gad_source=1&gclid=Cj0KCQjw_-GxBhC1ARIsADGgDjuixbq-SDIY28XYV40VYxevr4YEUL_wBhEDUaYo2oEDBVB48Wh9e4aAoxeEALw_wcB)
- [10] cuidevices.com, Accessed on: 21.5.2024 [Online]. Available: <https://www.cuidevices.com/blog/the-basics-of-ultrasonic-sensors>
- [11] modularcircuits.com, Accessed on: 21.5.2024 [Online]. Available: <https://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/>
- [12] Arduino, Accessed on: 21.5.2024 [Online]. Available: <https://docs.arduino.cc/learn/microcontrollers/analog-output/>
- [13] LEGO, Accessed on: 21.5.2024 [Online]. Available: <https://education.lego.com/en-us/product-resources/mindstorms-ev3/teacher-resources/python-for-ev3/>
- [14] ist.uni-stuttgart.de, Accessed on: 21.5.2024 [Online]. Available: <https://www.ist.uni-stuttgart.de/de/lehre/roborace/aufgabe/rr2016aufgabe/>
- [15] integrasources.com, Accessed on: 21.5.2024 [Online]. Available: <https://www.integrasources.com/blog/basics-of-pid-controllers-design-applications/>
- [16] MD25HV Datasheet, Accessed on: 21.5.2024 [Online]. Available: <https://docs.google.com/document/d/1xJHVG2dc3aEtedCHf3L9NzUy5KqpxeWjeS9Lfh9XuqA/edit>
- [17] control.com, Accessed on: 21.5.2024 [Online]. Available: <https://control.com/technical-articles/understanding-the-basics-of-pulse-width-modulation-pwm/>