

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**VZDUŠNÝ HOKEJ - REALIZACE A NÁVRH
AUTOMATICKÉHO HRÁČE**

AIR HOCKEY - CONSTRUCTION DESIGN AND CONTROL DESIGN OF ARTIFICIAL PLAYER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Dominik Jašek

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Radomil Matoušek, Ph.D.

BRNO 2020

Zadání diplomové práce

Ústav: Ústav automatizace a informatiky
Student: **Bc. Dominik Jašek**
Studijní program: Strojní inženýrství
Studijní obor: Aplikovaná informatika a řízení
Vedoucí práce: **doc. Ing. Radomil Matoušek, Ph.D.**
Akademický rok: 2019/20

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Vzdušný hokej – realizace a návrh automatického hráče

Stručná charakteristika problematiky úkolu:

Vzdušný hokej nebo Air hockey je společenská hra, která se vyvinula ze stolního hokeje. Je určena pro dva hráče, cílem je dostat kotouč pomocí speciální „pálky“ do soupeřovy branky. Obsahem diplomové práce bude realizace této hry s využitím automatizace, konkrétně fyzickou realizaci, vč. návrhu řízení.

Cíle diplomové práce:

- Rešerše hry Air Hockey v kontextu návrhu automatického hráče.
- Návrh konstrukce Air Hockey (zmenšená varianta).
- Návrh konstrukce automatického hráče (vestavěné řešení).
- Zprovoznění systému automatického hráče.
- Zhodnocení výsledků a tvorba prezentace.

Seznam doporučené literatury:

DENNIS, Andrew K. Raspberry Pi home automation with Arduino. Packt Publishing Ltd, 2013.

SPONG, Mark W. Impact controllability of an air hockey puck. Systems & Control Letters, 2001, 42.5: 333-345.

BISHOP, Bradley E.; SPONG, Mark W. Vision-based control of an air hockey playing robot. IEEE Control Systems Magazine, 1999, 19.3: 23-32.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2019/20

V Brně, dne

L. S.

doc. Ing. Radomil Matoušek, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Tato diplomová práce se zabývá návrhem a realizací automatického hráče vzdušného hokeje. Konkrétně je navržen herní stůl, kinematický typ robota H-bot a elektronické zapojení. Pohyb robota je zajištěn dvěma krokovými motory, které řídí Arduino Micro pomocí interních přerušení. Práce je doplněna dalšími prvky, jako je např. optická brána na detekci gólu a automatické vystřelování puku z branky. Ve spolupráci s Ondřejem Slámou [40] byla práce prakticky realizována.

Abstract

This diploma thesis deals with the design and implementation of an automatic Air hockey player. In particular, a gaming table and a kinematic type of H-bot robot and electronics are designed. The movement of the robot is ensured by two stepper motors, which are controlled by the Arduino Micro using timer interrupts. The thesis is improved by additional elements, such as goal detection using photoelectric sensors and automatic firing of the puck from the goal and others. The Air hockey robot was practically realized in cooperation with Ondřej Sláma [40].

Klíčová slova

Robotický vzdušný hokej, Arduino, H-bot, řízení krokových motorů

Keywords

Air hockey robot, Arduino, H-bot, stepper motor control

JÁŠEK, Dominik. Vzdušný hokej - realizace a návrh automatického hráče. Brno, 2020. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/125432>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky. Vedoucí práce doc. Ing. Radomil Matoušek, Ph.D.

Poděkování

Děkuji mému vedoucímu práce doc. Ing. Radomilovi Matouškovi, Ph.D. za vedení a připomínky k práci. Děkuji Ing. Tomáši Maradovi, Ph.D. a Ing. Zdeňkovi Cejpkovi za cennou pomoc při praktické realizaci.

Velmi děkuji mému kamarádovi Ondřeji Slámovi za ochotu a bezchybnou spolupráci při tvorbě robotického vzdušného hokeje.

Vřelé poděkování také patří mé přítelkyni a rodině za neutuchající morální podporu.

Čestné prohlášení

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením doc. Ing. Radomila Matouška, Ph.D a s použitím literatury uvedené v seznamu literatury.

V Brně dne 24. 6. 2020

.....

Bc. Dominik Jašek

Obsah

1	Úvod	15
2	Současný stav řešení robotického Air hockey	17
2.1	Průmyslový robot Nuvation	17
2.2	Realizace Jonathan Chiu, Daniel Kuo	18
2.3	JJrobots	18
2.4	Bakalářská práce Davida Kopeckého, ČVUT	20
3	Krokové motory	21
3.1	Typ motoru	21
3.2	Princip činnosti krokových motorů	21
3.3	RL obvod	22
3.4	Typy krokových motorů	23
3.4.1	Motor s proměnnou reluktancí (VR)	24
3.4.2	Motor s rotorem z permanentního magnetu (PM)	24
3.4.3	Hybridní motor	25
3.5	Druhy krokového motoru podle počtu fází	26
3.6	Zapojení fázových vinutí 2-fázového motoru	27
3.7	Buzení vinutí krokového motoru	28
3.7.1	Unipolární motor	28
3.7.2	Bipolární motor	29
3.8	Způsoby řízení 2-fázových krokových motorů	29
3.8.1	Čtyřtaktí s magnetizací jedné fáze (Wave Drive)	31
3.8.2	Čtyřtaktí s magnetizací dvou fází (Full Step Drive)	31
3.8.3	Osmiaktí (Half Step Drive)	32
3.8.4	Mikrokrokování (Microstepping)	33
3.9	Driver	34
3.10	Výběr vhodného krokového motoru	36
4	Řídicí členy	37
4.1	Raspberry Pi	37
4.1.1	Diplomová práce Bc. Ondřeje Slámy	37
4.2	Arduino	38
4.2.1	Přerušení	39
4.2.2	Hardwarové přerušení	40
4.2.3	Přerušení časovače	40

4.3	UART	41
5	Návrh herního stolu	43
5.1	Stůl z hliníkového profilu	44
5.2	Stůl ze dřeva	44
5.3	Ventilátory	44
6	Návrh automatického hráče	47
6.1	Kinematické typy	47
6.1.1	Kartézský typ	47
6.1.2	CoreXY	48
6.1.3	H-bot	50
6.1.4	Srovnání	51
6.2	Návrh kinematické konstrukce v 3D modeláři	52
7	Elektronika	55
7.1	Zdroje stejnosměrného napětí	55
7.2	Spínací prvky	55
7.3	Lokální kondenzátor pro solenoid	56
7.4	Zapojení	57
8	Řízení motorů	59
8.1	Matematický popis kinematického typu H-bot	59
8.2	S-křivka	60
8.3	Online generování S-křivek	60
8.4	P regulátor	62
9	Softwarová implementace	65
9.1	Struktura programu	65
9.2	Řízení signálů do driverů	67
10	Doplňující prvky	69
10.1	Optická brána na detekci gólu	69
10.2	Automatické vystřelování puku z brány	70
10.3	Automatické zastavení před mantinelem	71
10.4	Osvětlení	71
11	Závěr	73

Seznam obrázků	75
Seznam tabulek	76
Seznam algoritmů	76
Použité zkratky	77
Seznam použité literatury	78
Příloha	83

1 Úvod

Vzdušný hokej neboli Air hockey je hra pro dva hráče populární převážně v USA. Od roku 1978 se každoročně koná mistrovství světa, ve kterém reprezentanti USA jasně ukazují svou dominanci, když pouze dvakrát nevyhrál Američan. Česká republika zaznamenala svůj dosavadní největší úspěch na Mistrovství Evropy v Krakově v roce 2014, kdy soutěž vyhrál Čech Patrik Juchelka [1].

Cílem hráče je dostat puk do soupeřovy branky pomocí jednoruční hokejky. Puk klouže po hrací ploše díky malým otvorům v ní, kterými proudí vzduch vytlačovaný ventilátorem.

Tato diplomová práce se zabývá návrhem a realizací zmenšené varianty herního stolu a automatického hráče Air hockey. Práce je fakticky rozdělena do 9 kapitol. Kapitola 2 je rešerší současných řešení robotického Air hockey. Kapitola 3 zdůvodňuje, proč byly k řízení použity právě krokové motory a důkladně popisuje jejich princip. Kapitola 4 popisuje dvě hlavní použité řídicí jednotky - Arduino a Raspberry Pi.

Kapitoly 5 a 6 pojednávají o konstrukci a technické realizaci herního stolu a automatického hráče. Jsou představeny 3 kinematické typy a následně srovnány v kontextu robotického Air hockey. Následně je zdůvodněno, proč byl vybrán kinematický typ H-bot. Kapitola 7 popisuje použité elektrické zdroje a elektronické zapojení včetně použitých spínacích prvků.

Významnou částí je kapitola 8, ve které je popsán způsob řízení krokových motorů pomocí S-křivek a je představen řídicí algoritmus pro pohyb automatického hráče pomocí P regulátoru. Kapitola 9 je zaměřena na softwarovou implementaci řízení v Arduinu Micro, kdy jsou použita interní přerušení časovačů. Závěrečná kapitola popisuje doplňující prvky vylepšující herní možnosti, jako je detekce gólu, automatické vystřelování puku z branky a osvětlení pomocí LED pásků.

Práce vznikala v úzké spolupráci s Ondřejem Slámou, jehož diplomová práce [40] je zaměřena na detekci puku, herní strategii robotického hráče včetně počítačové simulace a tvorbu uživatelského rozhraní pro dotykový displej.

Celá práce byla prakticky realizována a video ze samotné hry lze vidět v příloze, případně jsou na konci práce přiloženy fotografie reálného modelu.

2 Současný stav řešení robotického Air hockey

Robotický Air hockey (česky robotický vzdušný hokej, dále jen RAH) není příliš rozšířený v porovnání s ostatními variantami robotických stolních her stylu člověk versus robot, jako je například robotický stolní fotbal nebo poměrně populární robotický stolní tenis.

Současný stav vývoje RAH se dá rozdělit do dvou směrů podle konstrukce manipulátoru. První z nich je dán použitím průmyslového robotu, kde největší předností je spolehlivost pohybu, rychlost a přesnost, značnou nevýhodou je pak jejich cenová dostupnost. Dostupnější se tak jeví druhý směr, kdy pohyb hokejky je zajištěn pomocí dvouosé kartézské konstrukce.

2.1 Průmyslový robot Nuvation

Za jeden z prvních projektů RAH můžeme považovat dílo firmy Nuvation, které vzniklo v roce 2008. Používá 32bitový Coldfire Freescale procesor a pro strojové vidění je použit kontroler Keyence CV 3000. Autoři uvádějí, že úspěšnost robota v blokování střel je až 98 % [3]. Při příležitosti Freescale Technology Forum získal tento projekt ocenění People's Choice Award.



Obrázek 1: Air hockey s průmyslovým robotem firmy Nuvation [3]

2.2 Realizace Jonathan Chiu, Daniel Kuo

Jonathan Chiu a Daniel Kuo v roce 2015 demonstrovali koncept použití operačního systému Windows 10 IoT Core, což není RTOS (real time operating system) na real time problému. Rozhodli se proto postavit právě RAH.

Windows 10 IoT Core běží na počítači MinnowBoard Max od firmy Intel. Detekce obrazu je zajištěna pomocí kamery Pixy CMUcam5, která zároveň provádí zpracování obrazu a do MinnowBoardu tak posílá přímo data o poloze puku. Je schopná pracovat až na 50 FPS¹. Interakce s uživatelem je zajištěna pomocí dotykové obrazovky. K pohonu se používají krokové motory s přírubou NEMA23 a přídržným momentem 1,9Nm. Pro pohyb os byl zvolen kartézský systém 2S2M, který bude popsán v kapitole 6.1. Pro kalibraci byly použity jazýčkové spínače, které jsou ovládané magnetickým tokem. Konstrukce se vyznačuje svou robustností a dobrým zpracováním pomocí hliníkových profilů. Řízení je naprogramováno v jazyce C#, který jakožto kompilovaný jazyk zajišťuje rychlost řízení.



Obrázek 2: RAH řízený počítačem MinnowBoard Max s Windows 10 IoT Core [3]

2.3 JJrobots

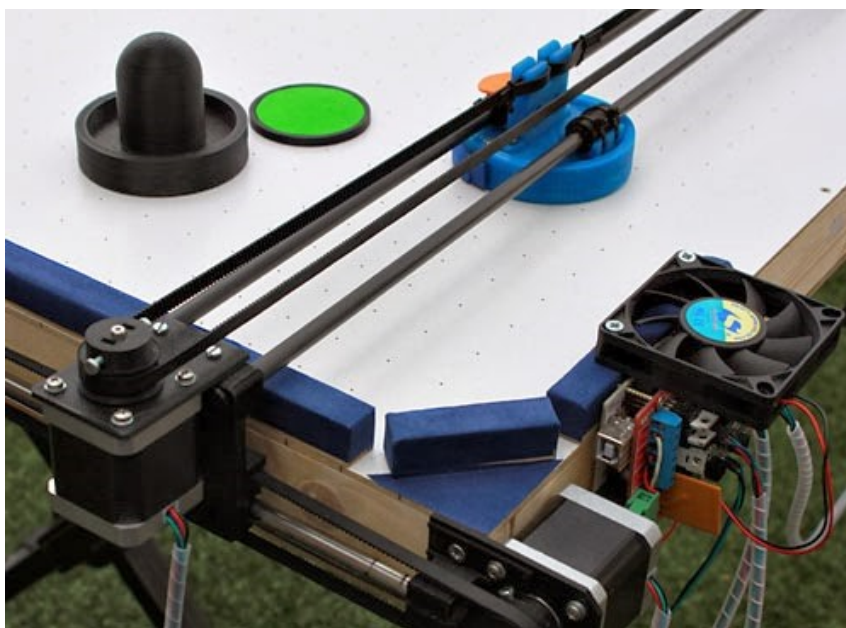
Společnost JJrobots sídlí v Kalifornii a zaměřuje se na nízkonákladové projekty s běžně dostupnými komponenty. Mezi jejich nejvýznamnější produkty patří i RAH, který se rozhodli prodávat pro veřejnost.

¹FPS - frames per second - počet snímků za vteřinu, používá se jako jednotka snímkovací frekvence

Ve svých projektech používají jednodeskový počítač Devia Robotics Control board s procesorem ARM Cortex M0 (ATSAMD21G18) s frekvencí 48 MHz, který je stejný jako v Arduino Zero, se kterým je tak kompatibilní. Po připojení jej vývojové prostředí Arduino IDE tedy rozezná právě jako Arduino Zero. Pro pohon jsou použity dva krokové motory s přírubou NEMA17, které jsou drivery A4988 poháněny zdrojem 12V (2A).

JJrobots vydali 2 verze RAH, které se nejvýrazněji liší svou konstrukcí, minoritně pak softwarově. Obě verze ke snímání obrazu používají přes Wi-Fi připojený telefon, kde si uživatel nainstaluje aplikaci, ve které se pak i následně manuálně kalibruje obraz.

První verze RAH od JJrobots byla založena na kartézském kinematickém typu, stejně jako výše uvedený RAH v kapitole 2.2. Použity jsou dva řemeny GT2 o průměru 6mm. [4]



Obrázek 3: První verze RAH od JJrobots s kartézským kinematickým typem [4]

Druhá verze RAH již používala kinematický typ H-bot. Přinesla plynulejší pohyby a byla jednodušší na sestavení pro uživatele. Na rozdíl od kartézského kinematického typu je použit pouze jeden řemen v celku. Výhodou této konstrukce je jiné rozložení pohybu motorů pro dosažení maximální rychlosti v osách x a y . Při kartézském typu se točil pouze jeden motor, v tomto případě se pro pohyb v osách otáčejí oba. Robot nemá koncové snímače ke zjištění polohy, tudíž poloha robota je zaznamenávána z kamery, což může způsobovat chyby a v krajním případě i rozbití robota, v případě, že by narazil do mantinelu.

Snímání obrazu je zajištěno mobilním telefonem, kde je patrná největší slabina, jelikož průměrně bylo dosaženo cca 30 FPS [6], což je poměrně málo pro tak rychlou hru, jako je Air Hockey.

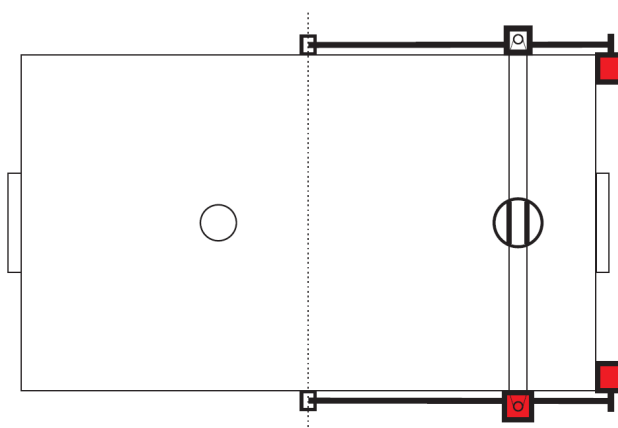


Obrázek 4: Druhá verze RAH od JJrobots s typem H-bot [6]

2.4 Bakalářská práce Davida Kopeckého, ČVUT

V České republice stojí za zmínku bakalářská práce Davida Kopeckého pod vedením Ing. Jiřího Zemánka na ČVUT v Praze. Autor se přímo nechal inspirovat projektem popisovaným v kapitole 2.3. V bakalářské práci využil herního stolu, který byl dříve postaven jako skupinový projekt studentů. Využívá kartézského kinematického typu 2S2M s motory Microcon Sx17-1003LQCEF s přírubou NEMA 17.

Pro snímání obrazu je použita USB webkamera a pro rozpoznávání knihovna OpenCV. Zpracování obrazu zajišťuje Raspberry Pi 3, které následně posílá data do Arduino Mega2560, ve kterém se vyhodnocuje strategie. Slabinou byl algoritmus detekující polohu puku, který výrazně zpomaloval rozhodovací strategii. [5]



Obrázek 5: Kartézské rozvržení motorů v práci Davida Kopeckého [5]

3 Krokové motory

Tato kapitola se bude zabývat výběrem vhodného typu motoru. Bude vysvětleno, proč byly v RAH použity krokové motory a následně detailně popsány.

3.1 Typ motoru

Při volbě vhodného typu motoru musíme zohlednit specifické požadavky projektu. Motory, které byly zvažovány pro řízení robota, byl kartáčový stejnosměrný motor (DC), bezkartáčový stejnosměrný motor (BLDC) a krokový motor. U prvních dvou zmíněných motorů je velkou výhodou jejich možnost vysokého zrychlení. Naopak nevýhodou je nutnost použití enkodéru pro zjištění aktuální polohy motoru. Pracovali bychom tak se servo mechanismem, tedy v uzavřené smyčce se zpětnou vazbou. Krokový motor již z principu své konstrukce (která bude popsána v kapitole 3.2) žádný enkodér nepotřebuje a pracuje tudíž bez zpětné vazby. Nedosahuje tak vysokého zrychlení jako DC a BLDC motory, ale pro účely RAH dosahuje zrychlení dostatečného. Významným faktorem při výběru vhodného motoru hrály pořizovací náklady, a tudíž byl zvolen motor krokový.

3.2 Princip činnosti krokových motorů

Základním elektrickým pohonem v této práci je tzv. krokový motor. V následující kapitole bude popsán jeho princip, typy a způsoby řízení.

Při konstrukci strojů často vyžadujeme přesné polohování pohyblivých částí. Použijeme-li k pohonu stejnosměrné nebo střídavé motory, je přesné polohování bez použití koncových spínačů a enkodérů značně obtížné. Proto se, zvláště v menších zařízeních, velmi často používají krokové motory.

Krokové motory jsou elektromechanická zařízení, která převádějí elektrické pulsy na diskrétní mechanické pohyby [8]. Tyto pohyby mohou být jak rotační (točivé krokové motory), tak lineární (lineární krokové motory). Pohyb je zajištěn pomocí magnetického pole, které vzniká v cívkách. Když totiž do dané cívky přivedeme elektrický proud, indukuje se magnetické pole a pokud je tento elektrický proud přiváděn cyklicky (ve smyslu pulsů), magnetické pole cívky se bude taktéž měnit. Jestliže do blízkosti této cívky přiblížíme magnet, natočí se do energeticky nejvýhodnější polohy, čímž je donucen k pohybu. Tento princip je použit v krokových motorech. Skládají se z nepohyblivé části (statoru) složené z cívek, které jsou cyklicky napájeny, a z pohyblivé části (rotoru), který je vyroben z feromagnetického materiálu nebo permanentního magnetu.

Rychlost otáčení rotoru tak je dána frekvencí, s jakou jsou přiváděny vstupní pulsy a počet diskrétních kroků přímo souvisí s počtem těchto pulsů.

Uved' me nyní hlavní výhody a nevýhody krokových motorů: [8, 9]

Výhody:

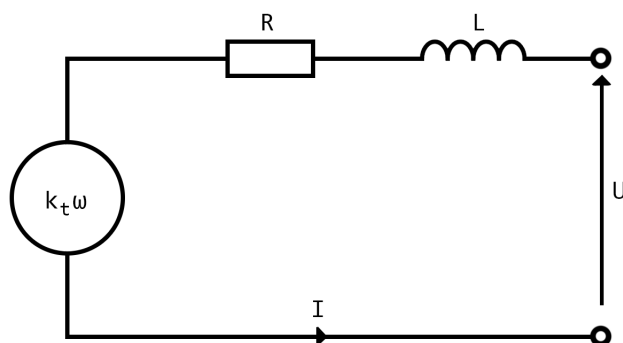
- Úhel natočení hřídele je přímo úměrný počtu vstupních impulsů.
- Rychlost otáčení je úměrná frekvenci vstupních impulsů.
- Řízení v otevřené smyčce bez nutnosti polohové zpětné vazby, což zajišťuje jednodušší ovládání.
- Rychlá a přesná reakce na zrychlení, zpomalení a změnu směru otáčení.
- Velký točivý moment při nízkých rychlostech otáčení; není potřeba používat převodovky.
- Krokový motor může být přetížen a zastaven bez poškození.
- Vyšší životnost, protože motor nemá kartáčky a komutátor.

Nevýhody:

- Při určitých rychlostech otáčení jsou náchylné k rezonancím.
- Nejsou vhodné pro vyšší rychlosti otáčení.
- Pokud dojde ke ztrátě kroku, je ztracena poloha poháněného stroje a systém musí být znovu inicializován.
- Mají menší točivý moment než AC nebo DC motory srovnatelné velikosti. [9]
- Hlučnost

3.3 RL obvod

Na obrázku 6 můžeme vidět náhradu krokového motoru za obvod tvořeného s kompenzačním faktorem $k_t \omega$, což je koeficient závislý na kroutícím momentu k_t a úhlové rychlosti otáčení ω , dále rezistorem s odporem R a cívku s indukčností L . [8]



Obrázek 6: Náhrada krokového motoru RL obvodem [8]

Chování tohoto obvodu popisuje vztah (1):

$$U = RI + k_t \omega(t) + L \frac{dI}{dt} \quad (1)$$

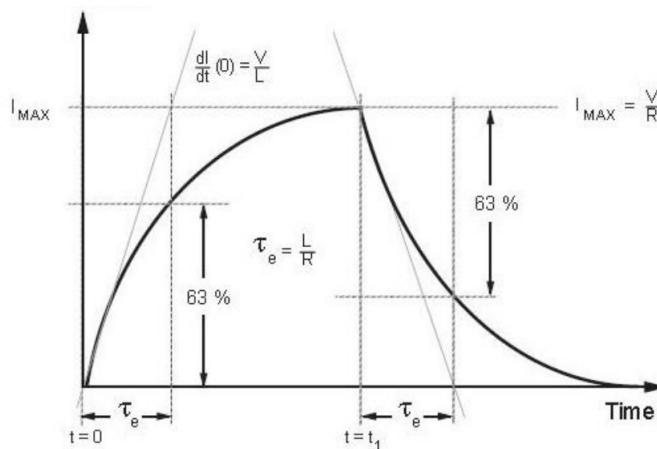
kde RI je napětí ovlivňující protékající proud, $k_t \omega$ je napětí kompenzující zpětnou elektromotorickou sílu². Řešením této diferenciální rovnice pro proud I dostáváme:

$$I = \frac{U - k_t \omega(t) - L \frac{dI}{dt}}{R} \Rightarrow I = \frac{U}{R} \left(1 - e^{-\frac{R}{L}t}\right) - \frac{k_t \omega(t)}{R} \quad (2)$$

Krouticí moment M je přímo úměrný protékajícímu proudu a je tedy možné jej vyjádřit vztahem (3):

$$M = k_t I \quad (3)$$

Jelikož má každá cívka nenulovou indukčnost, nárůst tekoucího proudu není okamžitý. Z pohledu automatizace se jedná o setrvačný člen 1. řádu. Odezva na skokovou změnu napětí je na obr. 7. V čase $t = 0$ byl obvod sepnut a v čase $t = t_1$ byl rozpojen. Elektromotorická síla je pro názornost nulová, ve vysokých otáčkách může však tento průběh výrazně pozměnit [8].



Obrázek 7: Proud v cívce při skokové změně napětí [8]

3.4 Typy krokových motorů

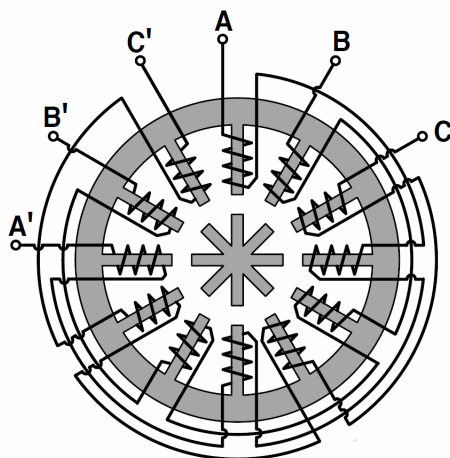
Krokové motory můžeme rozdělit na točivé a lineární. Lineární motory vykonávají neperiodický posuvný pohyb, stator je rozvinut do úsečky, a tedy pohyb do obou stran je konstrukčně omezený. Točivé rotory můžeme rozdělit na motory s proměnnou reluktancí, motory s permanentním magnetem a hybridní motory. Hybridní motory se vyrábějí ve standardizovaných velikostech podle doporučení NEMA

²anglicky back-EMF (electromotive force), odpovídá napětí, když se rotor pohybuje v proměnlivém magnetickém poli

(National Electrical Manufacturers Association). Velikost motoru se určuje podle velikosti strany příruby v palcích. Velikost NEMA23 tak znamená přírubu o rozměrech 2,3x2,3 palce, což je přibližně 57x57mm. Běžné velikosti jsou NEMA 8, 11, 17, 23, 34 a 42.

3.4.1 Motor s proměnnou reluktancí (VR)

Jedná se o nejstarší typ motoru, který je dnes již zřídka používán [9]. Alternativní názvy tohoto typu jsou reakční nebo reluktanční motor, případně motor s pasivním rotorem. Anglicky se tyto motory označují „variable reluctance motors“ (VR). Rotor je složen pouze z magneticky měkkých svazků plechů s pólovými nástavci, nalisovaný na hřídel [8]. Rotor je tedy bez vinutí. Stator je tvořen také svazkem plechů s pólovými nástavci, na nichž jsou uložena jednotlivá vinutí fází. Mezi pólovými nástavci statoru a rotoru je velmi malá vzduchová mezera (běžně 0,02 až 0,2mm [11]). Když těmito statorovými vinutími protéká proud, pólové nástavce se stávají elektromagnetem. Pólové nástavce rotoru jsou tímto magnetickým polem přitahovány a hřídel se tak dává do pohybu. Tento motor dosahuje menších točivých momentů (a tím i výkonů) než krokový motor PM popsáný v následující kapitole 3.4.2. Snadno jej poznáme tak, že při protáčení rukou se pohybuje plynule a bez znatelného „krokování“.

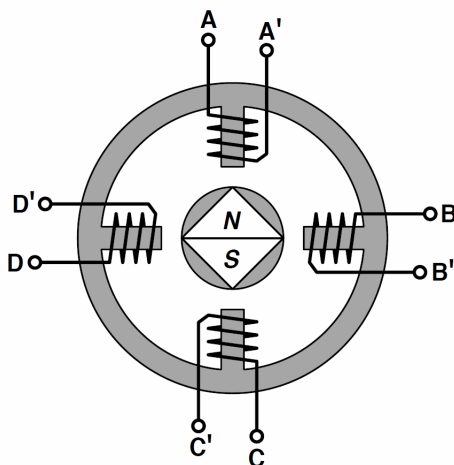


Obrázek 8: Krokový motor s proměnnou reluktancí [9]

3.4.2 Motor s rotorem z permanentního magnetu (PM)

V angličtině se těmto motorům často přezdívá „tin can“ (česky plechovka) [7]. Rotor není tvořen pólovými nástavci jako to bylo v případě VR motoru, ale je tvořen z radiálně pólovaného permanentního magnetu. Tyto póly střídají radiálně svou magnetickou orientaci mezi severním a jižním pólem. Tyto magnetické póly poskytují zesílené magnetické pole a díky tomu mají motory tohoto typu vyšší krouťací moment v porovnání s VR motory. Stator a jeho vinutí jsou tvořeny obdobným způsobem jako

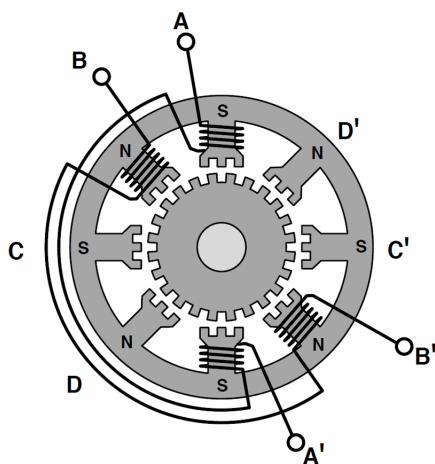
VR. Výhodou těchto motorů je také jejich jednoduchá konstrukce, a tedy nízká cena [9], dále pak nízké rozlišení kroků (od $7,5^\circ$ do 15° , tedy 48 - 24 kroků/otáčku) [7].



Obrázek 9: Krokový motor s rotorem z permanentního magnetu [9]

3.4.3 Hybridní motor

Jednoznačně nejpoužívanějším krokovým motorem je motor hybridní, protože vykazuje lepší momentové i dynamické parametry, vyšší rychlost a lepší krokové rozlišení (standardně od $0,9^\circ$ do $3,6^\circ$, což odpovídá od 400 do 100 kroků/otáčku [7]). V průmyslu tak získal své výhradní postavení i přes svou vyšší cenu než PM motor.



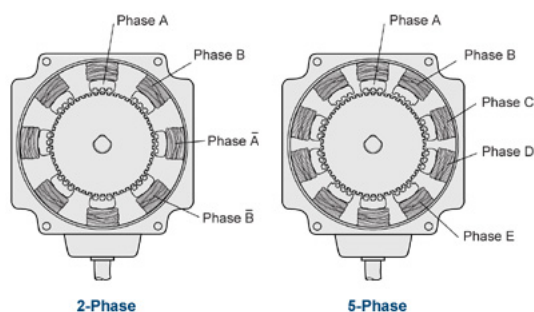
Obrázek 10: Krokový hybridní motor [9]

Hybridní motor kombinuje vlastnosti z obou typů motoru VR a PM. Rotor je tvořen z nejméně dvou nalisovaných pólových nástavců, které jsou z magneticky měkkého materiálu. Tuto vlastnost

přebírá od motoru VR. Tyto pólové nastavce jsou nasazeny na axiálně polarizovaný permanentní magnet, který zmagnetizuje každý z pólových nastavců na opačnou polaritu. Tuto vlastnost pak přebírá od PM motorů. Rotorové pólové nastavce pak mají na svém obvodu vyniklé zuby. Jejich počet určuje velikost kroku. Nejběžnější počet zubů je 50, což odpovídá velikosti kroku $1,8^\circ$ [9]. Konce statorových nastavců jsou pak taktéž opatřeny vyniklými zuby, které jsou na čele opatřeny drážkováním. Počet zubů statoru a rotoru není shodný, obvykle se volí počet rotorových zubů větší. Nevýhodou hybridních motorů je jejich vyšší cena, ale i přesto nalézají nejčastější uplatnění [10]. V RAH byly použity právě tyto hybridní krokové motory, proto již následující úvahy budou směřovány pouze k tomuto typu.

3.5 Druhy krokového motoru podle počtu fází

Krokové motory mají obvykle dvě fáze [9]. Fáze je označení pro počet magnetických obvodů statoru krokového motoru s příslušnými vinutími. Existují však i 3 a 5-fázové krokové motory. Vzájemně se liší dvěma zásadními rozdíly. První rozdíl je v samozřejmě v počtu fází. Druhý z nich je v jejich konstrukci. Dvoufázový motor je složen z 8 statorových nastavců, 3-fázový ze 6 a 5-fázový z 10.



Obrázek 11: Konstrukce 2-fázového a 5-fázového krokového motoru [12]

Počet kroků na jednu otáčku (při čtyřtaktním řízení s magnetizací jedné nebo dvou fází, viz 3.8) můžeme vyjádřit jako

$$p = 2 \cdot N_{ph} \cdot Ph, \quad (4)$$

kde N_{ph} je počet odpovídajících pólů na jednu fázi a Ph je počet fází [7].

Krokové rozlišení pak lze vyjádřit jako

$$\text{úhel kroku} = \frac{360^\circ}{p}, \quad (5)$$

Pro případ hybridního 2-fázového motoru s 50 zuby rotoru dostáváme (jak již bylo uvedeno v 3.4.3):

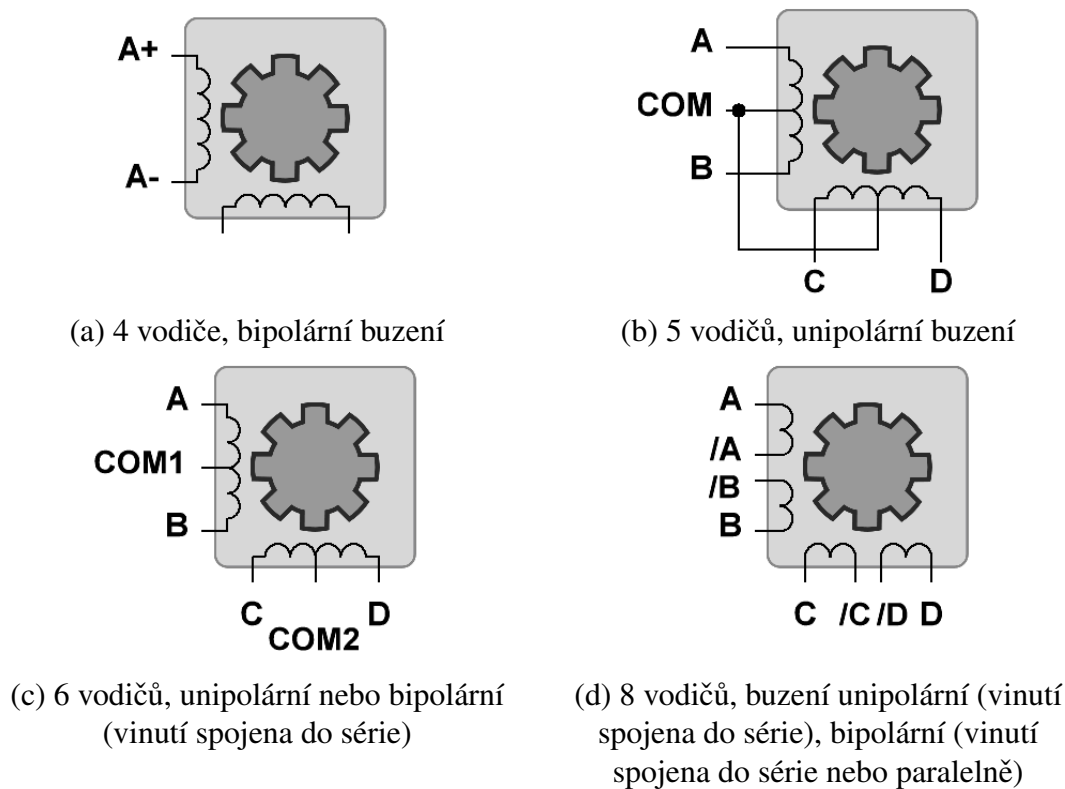
$$\text{úhel kroku} = \frac{360^\circ}{2 \cdot 2 \cdot 50} = 1,8^\circ$$

Výhoda 3-fázových a 5-fázových motorů je zřejmá; poskytují vyšší krokové rozlišení. Dále se vyznačují nižšími vibracemi, vyšším zrychlením a tišším chodem. Jsou vhodné pro aplikace s nárokem na velmi vysokou přesnost. Nevýhodou je jejich cena. [12]

3.6 Zapojení fázových vinutí 2-fázového motoru

Dvoufázové krokové motory patří mezi nejčastější motory používané v průmyslových aplikacích.

Vyrábějí se v několika variantách s různě zapojenými fázovým vinutími. Na první pohled je lze poznat podle počtu vyvedených vodičů [9]. Na obr. 12 jsou uvedeny nejběžnější z nich včetně možného buzení fázových vinutí (viz kap. 3.7).



Obrázek 12: Zapojení fázových vinutí 2-fázového motoru [9]

3.7 Buzení vinutí krokového motoru

V praxi se využívají dva základní druhy zapojení fází ve statoru ve 2-fázovém krokovém motoru: unipolární a bipolární.

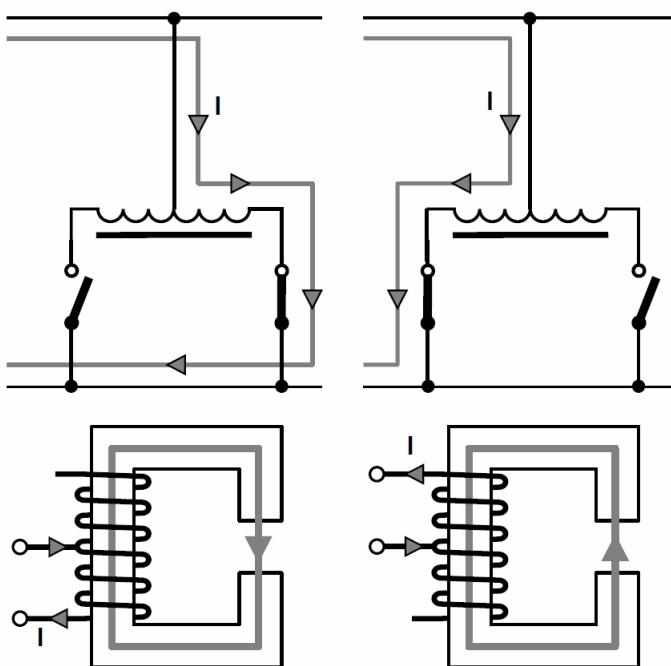
3.7.1 Unipolární motor

Princip unipolárního buzení protékajícího proudu objasňuje obr. 13, kdy spínač symbolizuje tranzistor. Každá fáze statoru je rozdělena na dva segmenty a jejich střídavým spínáním dosáhneme měnícího se magnetického toku [9].

Při buzení unipolárního motoru se používá neměnná polarita napětí. Jednotlivá vinutí se spínají tranzistory k zemi. Např. na obr. 3.6 bychom provedli zapojení připojením vodiče COM k napájecímu napětí +VCC a zbylé 4 vodiče přes řídicí tranzistory k zemi. Pro experimentální zjištění společného vodiče bez technické dokumentace můžeme měřit odpory mezi jednotlivými konci vodičů. Odpor mezi jedním koncem vinutí a středním vodičem bude poloviční jako odpor mezi konci vinutí.

Z obr. 13 je patrné, že poloviny vinutí jsou navinuty protisměrně. Proud tedy v každém segmentu protéká opačným směrem a vytváří opačné magnetické pole.

Každá fáze je vyvedena pouhým jedním koncem vinutí, ve kterém proud teče vždy stejným směrem, proto se této konstrukci říká „unipolární“.

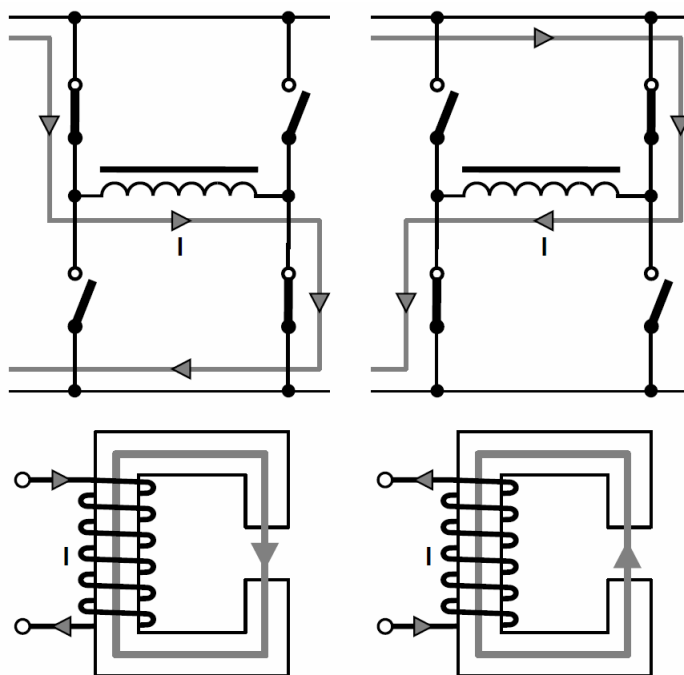


Obrázek 13: Unipolární buzení [9]

3.7.2 Bipolární motor

Při buzení bipolárního motoru potřebujeme polaritu napětí na vinutí měnit. Pro tento účel se často využívá H-můstek, což je obvod složený ze 4 tranzistorů. Jeho funkce je naznačena na obr. 14, kde opět spínače reprezentují řídicí tranzistory. Vzájemnou kombinací zapnutí/vypnutí jednotlivých tranzistorů lze docílit změny směru protékajícího proudu ve vinutí. Na rozdíl od unipolárního motoru, bipolární motor oba konce (póly) vinutí jsou vyvedeny. Z toho důvodu se nazývá „bipolární“ [9, 14].

Bipolární motory dosahují vyššího kroutícího momentu a jsou účinnější.



Obrázek 14: Bipolární buzení [9]

3.8 Způsoby řízení 2-fázových krokových motorů

Krokový motor může být považován ve své podstatě za střídavý synchronní motor, jelikož vinutími teče proud oběma směry. Pro řízení tak bylo vyvinuto několik technik.

K zobrazení proudů v cívkách se používá fázový diagram, který vykresluje proudy v obou cívkách na příslušné osy. Pro jednoduchost uvažujme kladnou osu I_A jako osu reprezentující úhel $\theta = 0$. S rostoucím úhlem θ se výsledný fázor pohybuje proti směru hodinových ručiček. Při úhlu $\theta = 90^\circ$ leží fázor na ose I_B a tudíž proud I_A tekoucí cívkou A je nulový.

Podle [16] pro úhel θ platí

$$\theta = \tan^{-1} \left(\frac{I_B}{I_A} \right). \quad (6)$$

Libovolný úhel θ ležící mezi 0° a 90° reprezentuje možnou pozici mikrokrokování (pozice mezi krajními pozicemi dosaženými při čtyřtaktním řízení s magnetizací dvou fází). Pro požadovanou jemnost mikrokrokování je třeba vygenerovat hodnoty proudů I_A , I_B tak, aby platila rovnice (6). Pokud je tedy požadováno dosažení např. 10x jemnějšího krokování, je třeba vygenerovat proudy I_A , I_B odpovídající úhlům $\theta = 0^\circ, 9^\circ, 18^\circ, 27^\circ, \dots, 81^\circ, 90^\circ$. Snadno nahlédneme, že existuje nekonečně mnoho řešení splňující tuto rovnici v závislosti na pevně zvoleném proudu I_A nebo I_B .

Hodnoty I_A , I_B se pak typicky volí podle požadavků na krouticí moment. Protože obě vinutí mají stejný odpor R , výkon P se určí jako:

$$P = I_A^2 R + I_B^2 R \quad (7)$$

Velikost fázoru pak můžeme vypočítat jako:

$$|I| = \sqrt{I_A^2 + I_B^2} = \sqrt{\frac{P}{R}} \quad (8)$$

Podle (7) je krouticí moment přímo úměrný velikosti proudu (za předpokladu, že není dosaženo magnetické saturace) a tím pádem je i přímo úměrný fázoru proudu (8). Také výkon souvisí podle (3) s velikostí fázoru proudu. Fázorový diagram nám tak dává dobrý přehled, jak se mění krouticí moment v závislosti na úhlu mikrokrokování θ . Mikrokrokování s konstantní délkou fázoru přináší výhodu konstantního krouticího momentu.

Všimněme si, že krouticí moment M je přímo úměrný fázoru, který je přímo úměrný \sqrt{P} . Tudíž krouticí moment M je přímo úměrný \sqrt{P} . Chceme-li proto např. 2x zvýšit krouticí moment, je zapotřebí zvýšit výkon 4x [16].

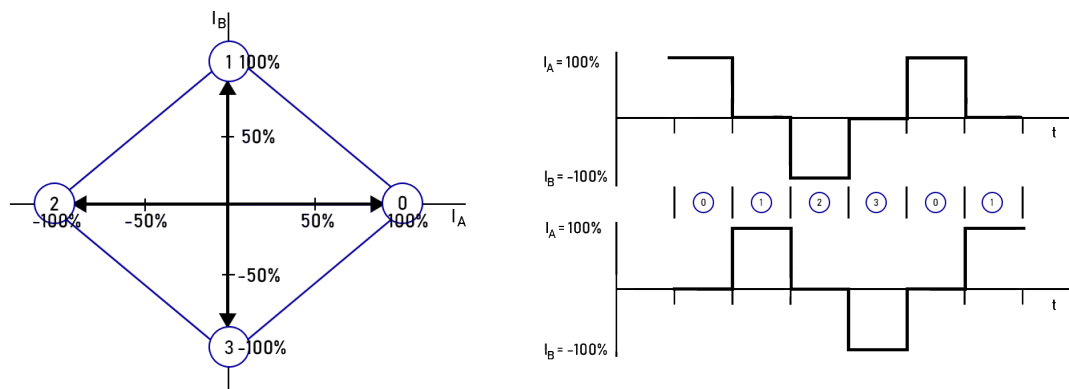
V následujících kapitolách budou ukázány jednotlivé druhy řízení.

3.8.1 Čtyřtaktní s magnetizací jedné fáze (Wave Drive)

V jednom kroku je aktivována vždy pouze jedna fáze. Toto řízení není využíváno často kvůli svému nízkému krouticímu momentu. V případě unipolárního motoru je vinutí využíváno pouze 25 % času v jednom cyklu (složený ze 4 kroků), v případě bipolárního pak 50 % času.

Krok	Fáze A		Fáze B	
	Pin 1	Pin 2	Pin 3	Pin 4
0	+	–		
1			+	–
2	–	+		
3			–	+

Tabulka 1: Čtyřtaktní řízení s magnetizací jedné fáze [9]



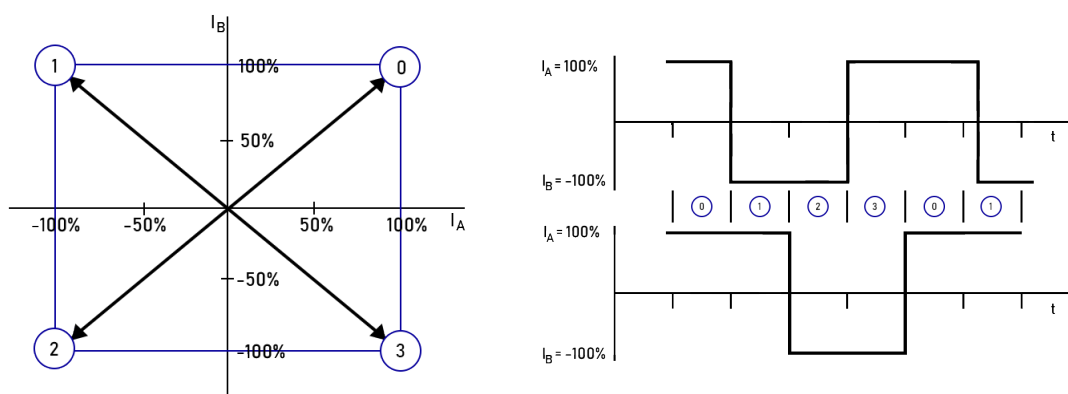
Obrázek 15: Fázorový diagram a časový průběh řízení wave drive [16]

3.8.2 Čtyřtaktní s magnetizací dvou fází (Full Step Drive)

Toto řízení se používá častěji než předchozí řízení. V každém kroku jsou aktivovány obě fáze, tudíž krouticí moment je vyšší. V unipolárním motoru se vinutí využívá 50 % času, u bipolárního motoru pak 100 % času.

Krok	Fáze A		Fáze B	
	Pin 1	Pin 2	Pin 3	Pin 4
0	+	–	+	–
1	–	+	+	–
2	–	+	–	+
3	+	–	–	+

Tabulka 2: Čtyřtaktní řízení s magnetizací dvou fází [9]



Obrázek 16: Fázorový diagram a časový průběh řízení full step drive [16]

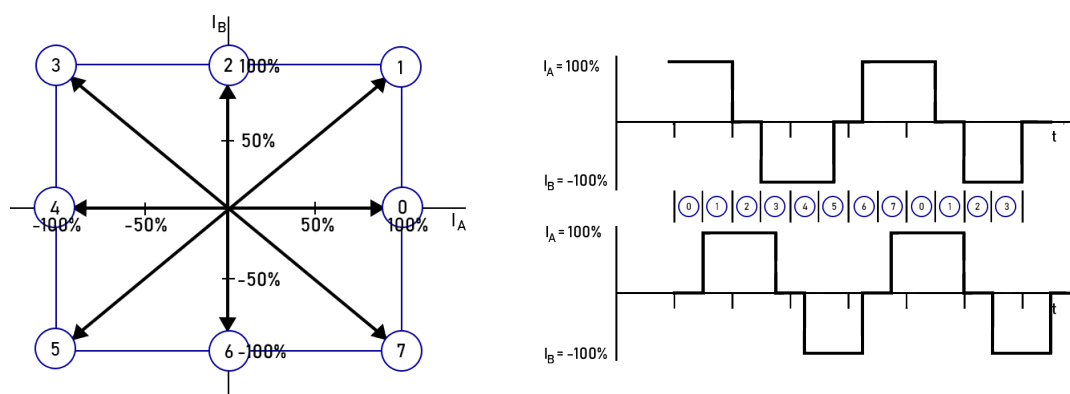
3.8.3 Osmiaktkní (Half Step Drive)

Osmiaktkní řízení kombinuje oba předchozí způsoby řízení. V sudých a lichých krocích se vzájemně střídají řízení s jednou aktivovanou fází a řízení s dvěma aktivovanými fázemi. Tím vzniká nekonzstantní kroutící moment, což je nevýhoda tohoto způsobu řízení. Výhodou je, že zjemňuje krokování o polovinu, tedy počet kroků potřebných k vykonání jedné otáčky (viz (5)) je dvojnásobný.

V případě unipolárního motoru je v lichých krocích vinutí využíváno 50 % času, v sudých 25 %. Průměrně se tak vinutí využívá na 37,5 %. V případě bipolárních motorů se pak vinutí v lichých krocích využívá 100 % času, v sudých 50 %. Průměrně se tak vinutí využívá na 75 %.

Krok	Fáze A		Fáze B	
	Pin 1	Pin 2	Pin 3	Pin 4
0	+	–		
1	+	–	+	–
2			+	–
3	–	+	+	–
4	–	+		
5	–	+	–	+
6			–	+
7	+	–	–	+

Tabulka 3: Osmiaktkní řízení [9]



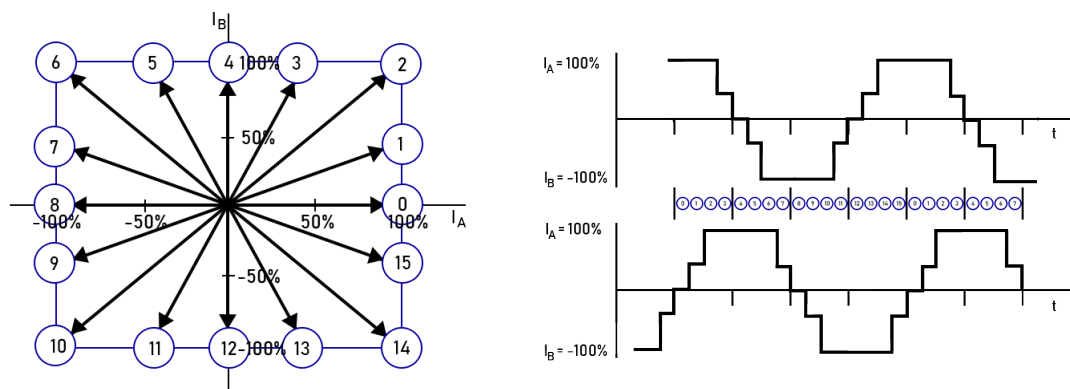
Obrázek 17: Fázorový diagram a časový průběh řízení half step drive [16]

3.8.4 Mikrokrokování (Microstepping)

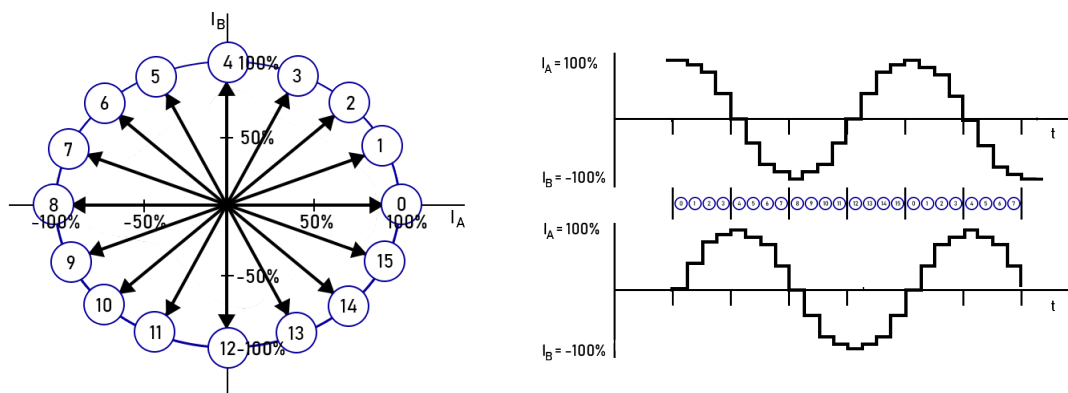
Mikrokrokování je řízení, při němž se proudy ve vinutích mění po menších krocích než v předchozích řízeních. V závislosti na fázorem obíhané trajektorii rozlišujeme dva druhy mikrokrokování: mikrokrokování s čtvercovou trajektorií a kružnicovou trajektorií. Čtvercová trajektorie dosahuje vyššího, avšak nekonstantního kroutícího momentu.

Čím je mikrokrokování jemnější, tím více se přibližuje ideálnímu sinusovému průběhu. Výhodou mikrokrokování je hladší chod motoru a tím pádem snížení intenzity vibrací. Mikrokrokování navíc umožní použití vyššího zrychlení při řízení motoru.

Tabulka řízení pro její zbytečnou složitost byla vypuštěna, řízení je patrné z obr. 18 a 19.



Obrázek 18: Fázorový diagram a časový průběh mikrokrokování s čtvercovou trajektorií [16]



Obrázek 19: Časový průběh a časový průběh mikrokrokování s kružnicovou trajektorií [16]

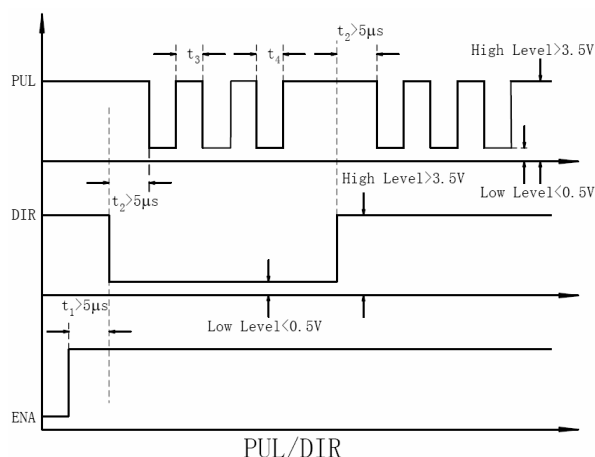
3.9 Driver

Nejjednodušší způsob, jak napájet krokové motory, je napájení jmenovitým napětím (udáno v technické dokumentaci motoru). Statorové cívky mají odpor řádově v jednotkách ohmů, tudíž pracovní proud dosáhne jmenovité velikosti při standardizovaných napájecích napětích, které je v praxi nejčastěji 5, 12 a 24 V [9]. Tento způsob však nedosahuje vyšší rychlosti otáčení ani větších točivých momentů. Uplatnění má zejména ve starších typech tiskáren, dále pak ve spotřebním a automobilovém průmyslu. [9]

Druhý, pokročilejší způsob je použití řídicího obvodu (tzv. driveru), který zajistí co rychlejší nárůst proudu a také mění směr proudu (tím pádem směr magnetického toku). Toto omezení proudu může být provedeno několika způsoby, uved' me např. řešení s předřadným rezistorem, dvouhladinové řízení proudu, či nejčastější pulsní řízení (chopper control). V RAH byl použit driver EM705 od výrobce Leadshine, který má napájecí napětí až do 70 V a špičkový proud 7 A. Pro jeho nastavení lze použít aplikaci SW ProTuner, která komunikuje pomocí standardu RS232. Driver má pokročilý algoritmus řízení proudu založený na DSP procesoru. Toto řízení probíhá pomocí PID regulátoru, kdy se proud reguluje na požadovanou maximální hodnotu pomocí akčního zásahu změny napětí. Hodnoty regulátoru K_P , K_I , K_D je možné manuálně měnit v aplikaci, či použít funkci Auto-tune, se aktivuje dvojitém přepnutím (v krátkém čase) polohy spínače SW4.

Maximální hodnoty protékajícího proudu se nastavují spínači na boční straně driveru, stejně tak mikrokrokování (které je s kružnicovou trajektorií).

Driver EM705 také disponuje funkcí bezsenzorové kontroly ztráty kroku. Snímáním napětí, proudu a zpětné elektromotorické síly dokáže driver rozpoznat ztrátu kroku bez použití enkodéru.



Obrázek 20: Napěťové signály pro řízení krokového motoru [19]

Pro komunikaci s nadřazeným systémem má driver 6 vstupních pinů: PUL+, PUL−, DIR+, DIR−, ENA+ a ENA−. Piny se záporným označením se připojují na společnou zem mezi řídicí jednotkou generující signál a driver. Signál pracuje na napěťové úrovni 5 V, tedy 4 - 5V je vyhodnoceno driverem jako hodnota HIGH, 0 - 0,5 V jako LOW.

Pin PUL+ detekuje náběžnou či sestupnou hranu (možné přizpůsobit v aplikaci ProTuner). Na tento pin se přivádí signál generující jednotlivé kroky. Pin DIR+ slouží pro určení směru otáčení. Výchozí hodnota pro signál LOW je proti směru hodinových ručiček, pro signál HIGH pak po směru. Tuto konfiguraci je možné opět změnit v aplikaci ProTuner. Pro časy t_3, t_4 na obr. 20 musí platit, že musí být delší než 7,5 μ s. [19]

V případě chyby (ať už ztráta kroku, nadproudové ochrany, či napěťové ochrany) driver indikuje chybu blikáním červené LED.

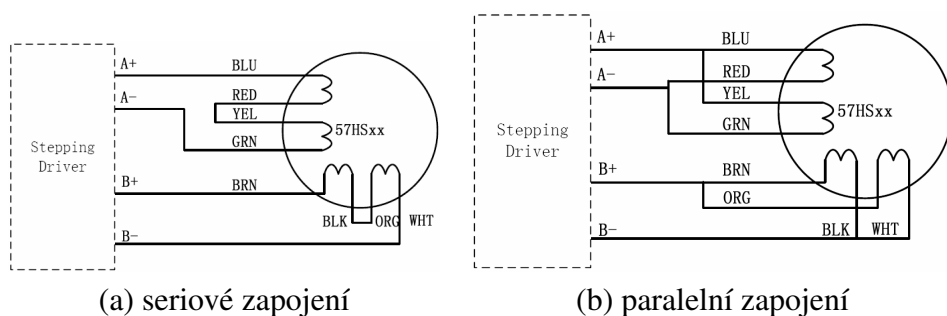


Obrázek 21: Driver EM705 [18]

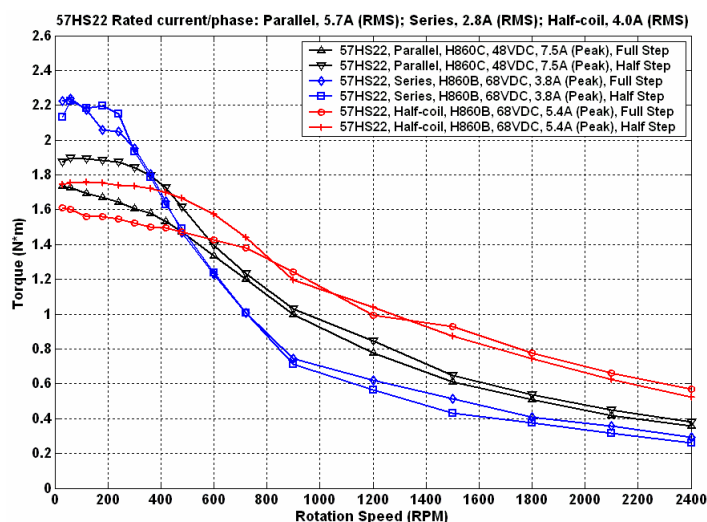
3.10 Výběr vhodného krokového motoru

Volbu vhodného modelu pro použití na RAH zásadně ovlivnily 3 faktory: cena, krouticí moment a velikost příruby. Požadovaná maximální rychlost byla stanovena na 2000 RPM³. Z důvodu složitě odhadovaného koeficientu tření a ostatních ovlivňujících faktorů bylo nesnadné počítat požadovaný krouticí moment. Byl tak otestován výkon motoru 42HS03 s přírubou NEMA17. Tento motor má přídržný moment 0,47 Nm. Při požadované maximální 2000 RPM je však hodnota kroutícího momentu přibližně 0,08 Nm, což není dostatečné pro pohyb robota.

Byl tak použit výkonnější a větší motor 57HS22 s přídržným momentem 2,2 Nm a přírubou NEMA23. Na obr. 23 můžeme vidět, že při zapojení vinutí jedné fáze do serie dosáhneme vyššího kroutícího momentu při nízkých otáčkách. V RAH je však spíše vyžadován vyšší krouticí moment při vyšších otáčkách, tudíž vinutí motoru byly zapojeny paralelně. Bylo použito bipolární řízení.



Obrázek 22: Zapojení fází motoru třídy 57SS [20]



Obrázek 23: Závislost momentu na rychlosti otáčení pro motor 57HS22 [20]

³RPM - rotates per minute (otáčky za minutu)

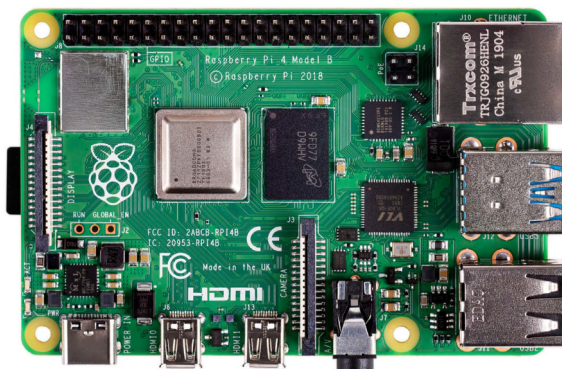
4 Řídicí členy

V této kapitole bude popsán RAH z hlediska jeho řízení. Budou představeny dva základní jednodeskové počítače použité v RAH - Raspberry Pi a Arduino. Druhý jmenovaný bude popsán detailněji.

4.1 Raspberry Pi

Raspberry Pi je jednodeskový počítač vyvinutý britskou nadací Raspberry Pi Foundation pod vedením Ebena Uptona na University of Cambridge. Cílem bylo přinést veřejnosti a studentům cenově dostupný mikropočítač, který poskytne shodné rozhraní pro programování a jednoduchou domácí automatizaci. [26]

Jako prozatím poslední verze bylo zveřejněno Raspberry Pi 4, které zvýšilo takt procesoru na 1,5 GHz. Oproti předchozím verzím je tato verze nabízena ve třech variantách podle velikosti RAM (1, 2, nebo 4 GiB). V RAH byla použita varianta s 1 GiB.



Obrázek 24: Raspberry Pi 4 Model B [27]

4.1.1 Diplomová práce Bc. Ondřeje Slámy

Diplomová práce Bc. Ondřeje Slámy je zaměřená právě na zpracování a vyhodnocení herních dat na Raspberry Pi. RPi zpracovává obraz z kamery a společně s přijatých dat z Arduina strategicky vyhodnocuje pohyb robota. K vizualizaci slouží dotykový displej s vlastním uživatelským rozhraním vytvořeným v jazyce Python pomocí knihovny *kivy*. Frekvence posílání příkazů do Arduina je konfigurovatelná, jako dostatečná se ukázala být hodnota 150 Hz. Pro detekci puku je použita Raspberry Pi kamera V1, která umožňuje 4x4 binning.

Jak již bylo zmíněno, obsahem práce Ondřeje Slámy je i strategie. Ta byla naprogramována v jazyce Python s využitím knihovny *pygame*. Samotnou práci a podrobnější popis lze nalézt v [40].

4.2 Arduino

Arduino je jedním z nejpopulárnějších open-hardware produktů na trhu. Bylo vyvinuto v Itálii v roce 2005 a jedná se o jednodeskový počítač založený na mikrokontroleru ATmega od firmy Atmel, který je nejčastěji používán jako počítač provádějící repetitivní činnost v nekonečné smyčce. [28]

Arduino bývá často voleno jako levnější alternativa pro úkoly, které by jinak vykonávalo PLC (Programovatelný logický automat). Na rozdíl od Raspberry Pi však nedokáže plnohodnotně nahradit klasický desktopový počítač, jelikož k němu nelze připojit monitor, klávesnici a myš. Nejčastěji se využívá pro ovládání externí elektroniky přes konektory (LED diody, servomotory, displeje, generátor signálu). Na Arduino je také možné umístit specifické shieldy modifikující funkci pro konkrétní činnost (typicky LCD shield nebo shield na řízení motorů).

Arduino kód se programuje v prostředí Arduino IDE a poté se nahraje do Arduina. Arduino IDE (*Integrated Development Environment*) je vývojové prostředí napsané v jazyce Java a vychází z výukového prostředí Processing a jazyka Wiring.

Jako alternativní přístup pro programování Arduina se nabízí firmware Firmata. Ten se nahraje do Arduina a stará se o veškeré jeho ovládání. Následně je pak pomocí importu knihovny Arduino do programu Processing možné ovládat Arduino přímo.

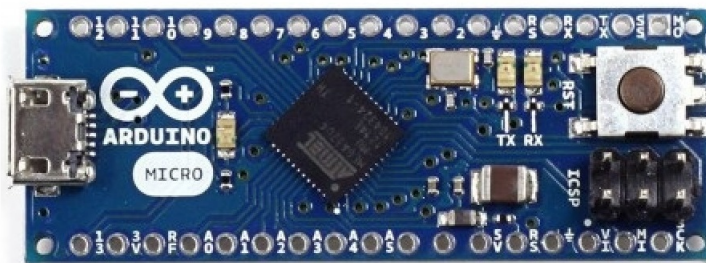
Arduino se vyrábí v několika různých modelech. Zmíníme nejpoužívanější z nich.

Arduino Mini patří k nejmenším nabízeným modelům Arduina a je navrženo pro úsporu místa. Kvůli malým rozměrům zde však chybí převodník, který umožňuje komunikaci mezi počítačem a čipem. Je tedy nutné použít externí USB-UART převodník. Pro své malé rozměry nachází využití ve vypínačích a podobných zařízeních vyžadující malé rozměry kontroleru.

V současné době nejpoužívanější a nejznámější typ Arduina je Arduino Uno. Na rozdíl od Arduina Mini již obsahuje zabudovaný převodník. Arduino Uno nabízí nespočet možností rozšíření pomocí shieldů, které jsou vyráběny často právě na tento model. Deska obsahuje procesor ATmega328. Arduino obsahuje 13 digitálních pinů (z toho dva se používají pro sériovou komunikaci) a 6 analogových pinů.

Pokud by tento počet digitálních/analogových pinů nebyl pro daný projekt dostatečný, nabízí se použití Arduina Mega2560, které připomíná prodlouženou verzi Arduina Uno. Nabízí výkonnější procesor ATmega2560 a má celkově 54 digitálních pinů (z nichž 15 mohou být použity jako výstupy PWM) a 16 analogových pinů.

Poslední Arduino, které zmíníme, je Arduino Micro. Obsahuje čip Atmega32u4, který se může chovat jako např. myš či klávesnice, a odesílat příkazy jako je stisk klávesy nebo tlačítka a posun myši. Oproti čipu na Arduinu Uno nabízí více časovačů, které jsou v RAH využívány.



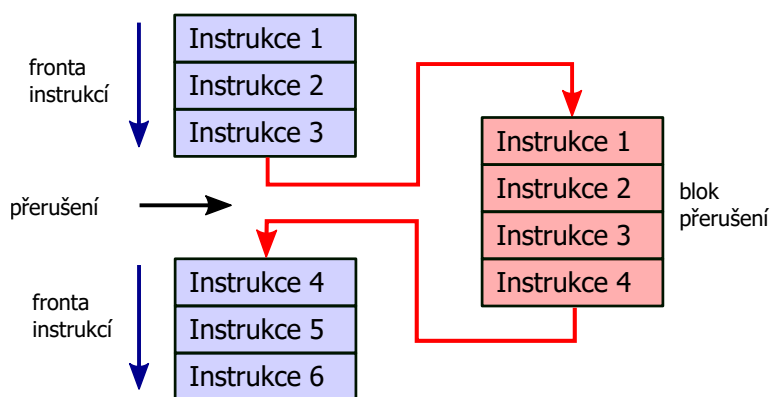
Obrázek 25: Arduino Micro [28]

4.2.1 Přerušení

Ve skalárních procesorech, jako je Atmega32u4 v Arduino Micro, může probíhat v jeden časový okamžik pouze jedna instrukce. Při normálním stavu jsou jednotlivé instrukce zpracovávány po sobě a jsou řazeny do fronty.

V informační technice pojem *priorita* znamená potřebu vykonat instrukci ihned. Instrukce, která má danou prioritu, může pozastavit vše ostatní. Poté je provedena požadovaná instrukce a následně po jejím dokončení se program vrací zpět na původní místo. Způsob vykonání těchto instrukcí se nazývá *interrupt* (přerušení). Interrupty jsou zpravidla obsluhovány v tzv. ISR funkcích (Interrupt Service Routine), což jsou zvláštní druhy funkcí, které se liší od klasických funkcí tím, že nemohou mít žádné vstupní parametry a nevrací návratovou hodnotu. Při práci s ISR je nutné definovat proměnné jako *volatile*, aby se zabránilo kompilátoru v optimalizování této proměnné [31].

Ilustraci interruptu lze vidět na obr. 26. K přerušení může dojít dvěma způsoby - tzv. vnější přerušení (hardwarové přerušení) a vnitřní přerušení (časovač nebo softwarové přerušení). Priorita přerušení se pak řídí tzv. interrupt vektorem, který upřesňuje priority jednotlivých přerušení.



Obrázek 26: Ilustrace přerušení

4.2.2 Hardwarové přerušení

Konfigurací hardwarového přerušení na určitý pin (definovaný jako vstup) lze dosáhnout spuštění daného ISR kdykoliv je splněna požadovaná podmínka na hodnotu pinu. Tou může být jedna z těchto možností:

- LOW - kdykoliv je hodnota pinu LOW
- CHANGE - kdykoliv se hodnota pinu změní
- RISING - kdykoliv je detekována náběžná hrana
- FALLING - kdykoliv je detekována sestupná hrana

V Arduino Micro lze hardwarové přerušení použít pro piny 0, 1, 2, 3 a 7, avšak piny 0 a 1 se používají pro sériovou komunikaci s Raspberry Pi, tudíž je není možné využít. V kapitole 10.1 je však ukázáno, jak lze pomocí AVR příkazů tento počet pinů na hardwarové přerušení rozšířit.

4.2.3 Přerušení časovače

Časovač (timer) je integrovaná hardwarová část v čipu, pomocí níž lze měřit časové události. Při každé změně hodinového signálu je inkrementována hodnota registru TCNTn n-tého časovače o hodnotu 1. Maximální hodnota časovače je dána na velikostí registru TCNTn. Čip Atmega32u4 má celkově čtyři časovače o různých velikostech registru TCNTn. Timer0 má velikost 8 bitů. Timer1 a Timer3 mají shodně velikost 16 bitů a vysokorychlostní Timer4 má velikost 10 bitů.

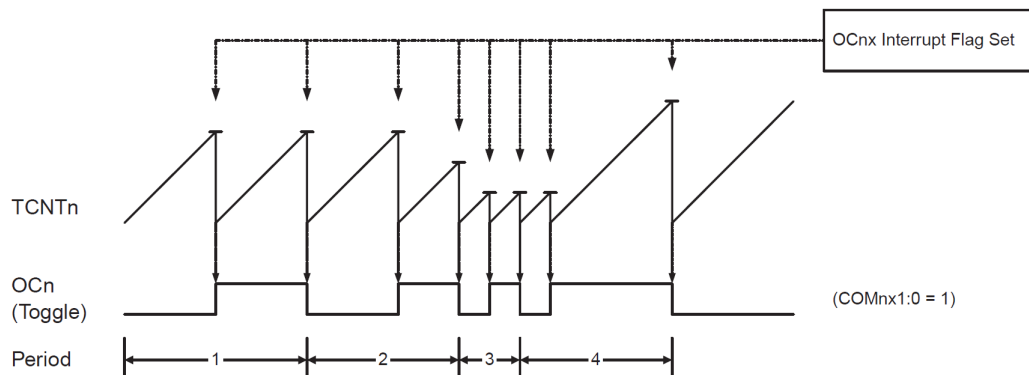
Každý časovač může mít nastavený tzv. *prescaler*, který zpomaluje inkrementování hodnoty registru TCNTn. Hodnoty prescaleru mohou být 1, 8, 64, 256 nebo 1024 a nastavují se pomocí registrů TCCRnB.

Časovač může pracovat v několika režimech: [33]

1. Normální režim (přetečení) - funkce ISR se spustí, pokud hodnota TCNTn přeteče maximální hodnotu danou velikostí časovače, např. pro 16-bitový časovač Timer1 je tato hodnota $2^{16}=65536$.
2. Záznam vstupu - umožňuje zápis současné hodnoty TCNTn do jiného registru při externí události na některém z ICP pinů (Input Capture Pin).
3. Pulzně-šířková modulace - umožňuje použití pulsně-šířkové modulace na výstupu pinu.
4. Porovnávací režim - funguje na podobném principu jako režim přetečení, ale dovoluje nám nastavit hodnotu, při kterém se vyvolá funkce ISR. Pro zápis této hodnoty slouží registry OCRn. Na obr. 27 můžeme vidět ilustraci funkce časovače v režimu CTC, který zajistí resetování TCNTn na nulovou hodnotu po vykonání ISR. Frekvence OCn je závislá na porovnávací hodnotě OCRn a zvoleném prescaleru. Platí: [32, str. 99]

$$f_{OCn} = \frac{f_{clk_I/O}}{2 \cdot N \cdot (1 + OCRn)}, \quad (9)$$

kde $f_{clk_I/O}$ je frekvence čipu (v případě Atmega32u4 je $f_{clk_I/O} = 16 \text{ MHz}$) a N reprezentuje hodnotu zvoleného prescaleru.



Obrázek 27: Časovač v režimu CTC [32]

4.3 UART

UART (Universal Asynchronous Receiver-Transmitter) je počítačová součástka sloužící pro asynchronní sériovou komunikaci. Sériová komunikace se vyznačuje tím, že data putují v podobě bitů seriově za sebou po jednom komunikačním kanálu (na rozdíl od paralelní komunikace, kdy se používá více komunikačních kanálů - kabelů).

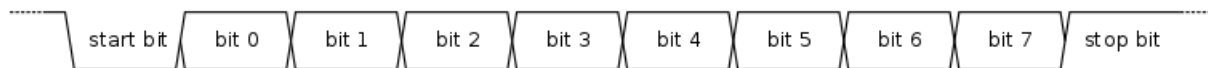
Na rozdíl od komunikačních protokolů jako např. SPI nebo I²C se v případě UART nejedná o protokol, nýbrž o fyzický integrovaný obvod (IO), který může být součástí jiného IO [30].

Pomocí UART spolu mohou komunikovat právě dvě zařízení. Označme je zařízení A a zařízení B. Každé zařízení, které má UART, obsahuje přijímací pin Rx (receive) a vysílací pin Tx (transmit). Spojení těchto zařízení provedeme propojením pinu Rx zařízení A a pinu Tx zařízení B. Taktéž je nutné propojit piny GND obou zařízení a tím zajistit jednotnou napěťovou úroveň pro komunikaci. Tímto jsme dosáhli tzv. simplexního spojení, kdy můžeme komunikovat pouze jedním směrem (v našem případě ze zařízení B do zařízení A). Abychom obdrželi duplexní spojení, je třeba propojit pin Rx zařízení B a pin Tx zařízení A.

Duplexní komunikace se dále dělí na tzv. poloviční duplex, kdy komunikace je možná současně jen jedním směrem, nebo na plný duplex, kdy je možné současně komunikovat oběma směry. [30]

Jelikož se jedná o asynchronní komunikaci, musí mít vysílací i přijímací zařízení shodně nastavený tzv. baud rate, což odpovídá počtu přenesených symbolů za sekundu. Bit rate pak znamená počet přenesených bitů za vteřinu. Obecně platí, že bit rate je vždy větší nebo roven baud ratu.

UART posílá data po bajtech. Výchozí hodnota je HIGH kvůli detekování mechanického přerušení vodiče. Délka odeslání jednoho bitu je závislá na bit ratu. Nejprve se zahájí komunikace odesláním signálu LOW (neboli odesláním start bitu). Odesílaný bajt, který se skládá z 8 bitů a z případného parity bitu, je pak vzápětí odeslán přes pin Rx do druhého zařízení na pin Tx. Odeslání bajtu je zakončeno stop bitem s hodnotou HIGH. Druhé zařízení si tyto bity dekoduje a zaznamená do čtecího bufferu.



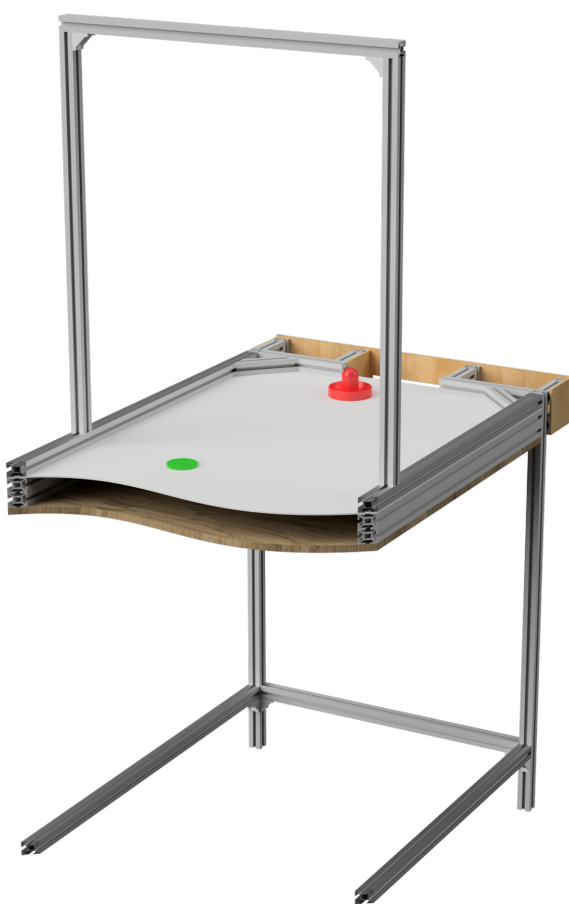
Obrázek 28: UART komunikace [29]

5 Návrh herního stolu

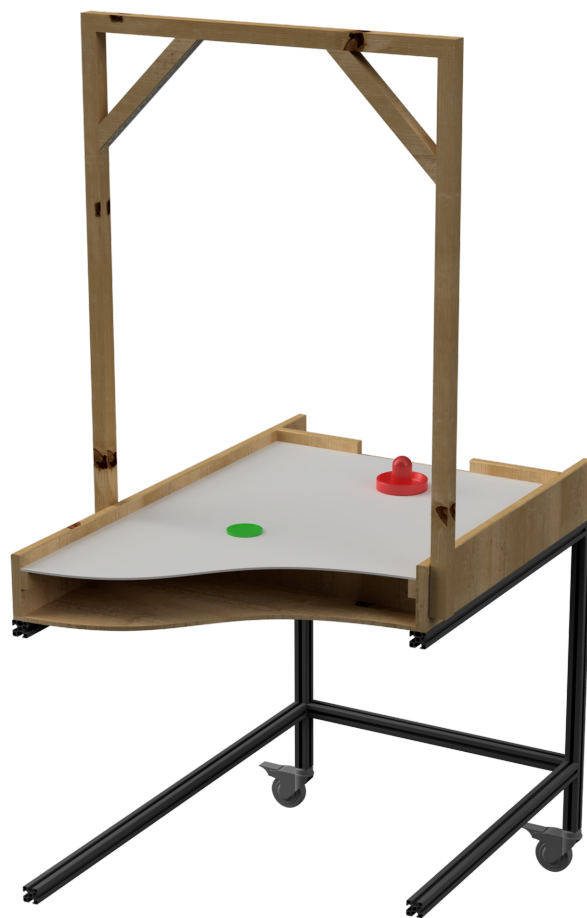
V této kapitole budou představeny 2 návrhy stolu na RAH.

Pro modelování byl použit software Autodesk Fusion 360, který nabízí cloudové uložení, což je jeho největší předností. V posledních letech si získává na popularitě zvláště mezi komunitou 3D tiskařů. Všechny popisované modely je možné si interaktivně prohlédnout online, viz [41].

Jako podstavec herního stolu byl navržena konstrukce z hliníkových profilů stavebnicového systému Minitec od firmy Opticontrol. Byly použity profily s čtvercovým průřezem o velikosti 30x30 mm. Na vsazení stolu jsou na profilech umístěny hliníkové L profily. Pro pohyb stolu jsou připevněny na konstrukci otočná kolečka s brzdou.



(a) stůl z hliníkových profilů



(b) stůl ze dřeva

Obrázek 29: Řez stolů RAH

5.1 Stůl z hliníkového profilu

Původní koncept stolu na Air hockey byl navržen z hliníkových profilů systému Kombi 30x30. Bok herního stolu je tvořen třemi profily systému Kombi 30x30. Výhodou tohoto konceptu je jeho robustnost a odolnost vůči neopatrné manipulaci či nárazům hokejky do mantinelu. Nevýhodou je pak nesnadná možnost utěsnění vzduchové komory a vyšší cena. Návrh je možné vidět na obr. 29(a).

5.2 Stůl ze dřeva

Jako levnější alternativa konstrukce stolu byla navržena konstrukce dřevěná. Jako materiál byla použita prkna ze smrkového dřeva o šířce 110 mm a tloušťce 18 mm. S ohledem na oficiální rozměry Air hockey byly rozměry hrací plochy zvoleny 1000x600 mm.

Jako materiál pro hrací desku, po které bude klouzat puk, byl zvolen kompozitní panel alubond.

Pro snadnou manipulaci s elektrickými komponenty umístěnými za brankami jsou zadní překližky připevněny ke stolu pomocí několika magnetů (viz obr. 53). Návrh stolu je je zobrazen na obr. 29(b).

5.3 Ventilátory

K zajištění dokonalého klouzání puku po povrchu se používají ventilátory umístěné na spodní desce stolu. Ty vhánějí do vzduchové komory vzduch, kde vzniká přetlak.

Do alubondové hrací desky byla vyvrtána čtvercová síť tvořená celkově 920 dírami (40x23) o průměru 1 mm. Vzdálenost děr 25 mm byla volena s ohledem na oficiální rozměry sportovního Air hockey stolu a průměrem puku.

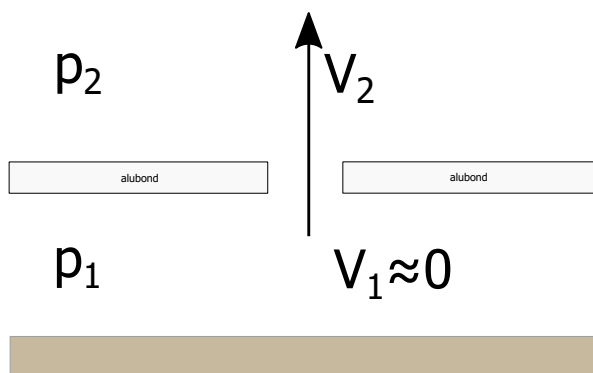
Puk o průměru 63 mm tak leží vždy na nejméně 4 dírách. Jeho hmotnost je $m = 12$ g. Aby puk klouzal po povrchu, musí platit:

$$F_g = F_{vz} \quad (10)$$

kde F_g je gravitační síla působící na puk a F_{vz} je síla způsobená proudícím vzduchem z děr, která působí proti síle $F_g = mg$, kde g je gravitační zrychlení $g = 9,81 \text{ m/s}^2$.

Puk leží vždy minimálně na 4 dírách, které mají obsah $S = \frac{\pi}{4}d^2$, kde $d = 0,001$ m. Působící sílu proudícího vzduchu, který protéká čtyřmi dírami vytváří tlak:

$$p = \frac{F_{vz}}{4S} = \frac{F_g}{4S} = \frac{mg}{4S} \quad (11)$$



Obrázek 30: Proudění vzduchu otvorem na hrací ploše

Pro místa 1 a 2 na obr. 30 platí podle Bernoulliho rovnice:

$$p_1 + \frac{1}{2}\rho v_1^2 = p_2 + \frac{1}{2}\rho v_2^2 \quad (12)$$

kde ρ je hustota vzduchu přibližně $\rho = 1,225 \text{ kg/m}^3$. Z (12) můžeme vyjádřit rozdíl tlaků a po dosazení dostáváme:

$$\Delta p = p_1 - p_2 = \frac{1}{2}\rho v_2^2 \quad (13)$$

Vyjádřením rychlosti v_2 z (13) a dosazením rozdílu tlaků (11) získáváme rychlost vzduchu proudící skrz díru v hrací desce:

$$v_2 = \sqrt{\frac{2 \cdot \Delta p}{\rho}} \quad (14)$$

Pro počet děr N pak požadovaný průtok vzduchu můžeme vyjádřit jako:

$$Q = v_2 \cdot S \cdot N = \sqrt{\frac{2 \cdot \frac{mg}{4S}}{\rho}} \cdot S \cdot N = \sqrt{\frac{2 \cdot \frac{0,012 \cdot 9,81}{4 \cdot \frac{\pi}{4} \cdot (0,001)^2}}{1,225}} \cdot \frac{\pi}{4} \cdot (0,001)^2 \cdot 920 = 0,1787 \text{ m}^3/\text{s} = 643,4 \text{ m}^3/\text{h}$$

Ve sportovních Air hockey stolech jsou použity ventilátory radiální. Kvůli nedostatku místa bylo však nutné použít ventilátor axiální. Z dostupných ventilátorů se nejbližší této hodnotě jeví použití dvou ventilátorů SUNON PMD2412PMB1-A(2).GN s průtokem vzduchu $322,8 \text{ m}^3/\text{h}$ [21]. Napájecí napětí tohoto ventilátoru je 24 V a příkon má 18,2 W. Použitý ventilátor ukazuje obr. 31.



Obrázek 31: Ventilátor SUNON PMD2412PMB1-A(2).GN [21]

Celkově je tak vháněno do vzduchové komory $645,6 \text{ m}^3/\text{h}$ vzduchu, což po utěsnění komory zajišťuje dostatečně plynulý a hladký pohyb puku po alubondové hrací ploše.

6 Návrh automatického hráče

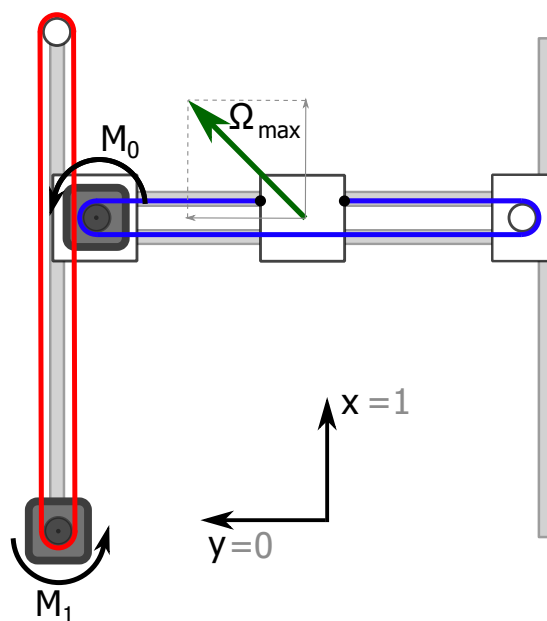
V této kapitole bude popsána konstrukce automatického hráče. Všechny díly byly vytvořeny v programu Autodesk Fusion 360 a byly vytisknuty z filamentu typu PLA na 3D tiskárně Original Prusa i3 MK3S. Bylo čerpáno z [22, 23].

6.1 Kinematické typy

Při návrhu automatického hráče byly zvažovány známé kinematické typy používané pro konstrukce 3D tiskáren. V následujících podsekcích budou porovnány jednotlivé typy. Obrázky byly vytvořeny v programu Inkscape, který podporuje práci s vektorovými formáty. Při výběru vhodného typu byly zvažovány tři kinematické typy, které budou na následujících stránkách popsány. Na závěr bude provedeno zhodnocení a odůvodnění vybraného typu.

Na obrázcích 32, 33, 35 představuje bílý čtverec jádro, se kterým bude pohybováno. Budeme používat dvě souřadnicové osy: šedé pro motory $(0, 1)$ a černé pro hrací plochu (x, y) . Počet otáček motoru M_i budeme značit q_i . Rychlost i -tého motoru M_i v jednotkách krok/s budeme značit ω_i . Maximální rychlost motoru budeme značit ω_{max} . Konstanta α je dána počtem zubů řemenice N , roztečí zubů na řemeni d a rozlišením mikrokrokování n . Platí $\alpha = \frac{N \cdot s}{n}$.

6.1.1 Kartézský typ



Obrázek 32: Kartézský kinematický typ

Kartézský kinematický typ (někdy také nazývaný zjednodušeně 1S1M) je nejprimitivnější kinematický typ. Je zapotřebí použít dva oddělené řemeny (na obr. 32 znázorněny červeně a modře). Matice transformace je

$$\begin{bmatrix} x \\ y \end{bmatrix} = \alpha \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ q_1 \end{bmatrix} \quad (15)$$

Pro přepočet rychlosti motorů na rychlost v souřadnicích x, y slouží následující matice transformace:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \alpha \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix} \quad (16)$$

Maximální rychlosti hokejky dosáhneme při souběžném pohybu motorů při jejich maximální rychlosti. Dosazením $\omega_0 = \omega_1 = \omega_{max}$ dostáváme rychlost

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \alpha \begin{bmatrix} \omega_{max} \\ \omega_{max} \end{bmatrix} = \Omega_{max} \quad (17)$$

Tato rychlost je na obr. 32 vyznačena zeleně. Velikost této rychlosti je pak

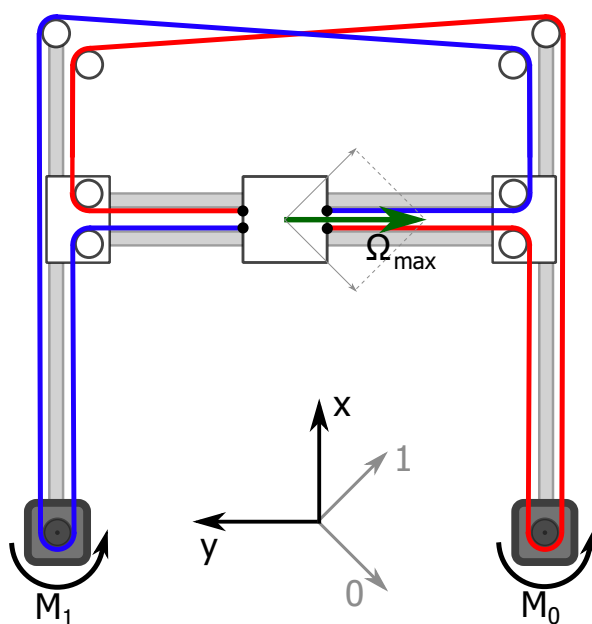
$$\|\Omega_{max}\| = \alpha \sqrt{(\omega_{max}^2 + \omega_{max}^2)} = \alpha \sqrt{2} \omega_{max} \quad (18)$$

Snadno také nahlédneme, že velikost maximální rychlosti v osách x a y je $\alpha \omega_{max}$. Maximální pohyb je tedy v úhlopříčných směrech na souřadnicové osy x, y .

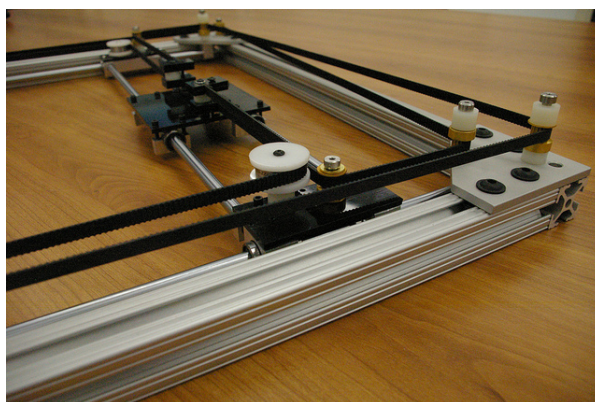
Kinematický typ 2S2M je obdobný typ s tím rozdílem, že pro pohyb v ose x jsou zapotřebí dva řemeny a dva motory. Tedy červený řemen s motorem na obr. 32 vlevo by u typu 2S2M byl také vpravo.

6.1.2 CoreXY

Typ CoreXY je velice oblíbený mezi konstruktéry 3D tiskáren, jelikož zajišťuje velice vysokou přesnost polohování manipulátoru [23]. Opět je zapotřebí použití dvou oddělených řemenů. Překřížení řemenů v horní části obr. 33 je řešeno vyvýšením řemenic příslušící jednomu z řemenů, viz obr. 34.



Obrázek 33: Kinematický typ CoreXY



Obrázek 34: Překřížení řemenů CoreXY [22]

Matice transformace pro typ CoreXY je:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{\alpha}{2} \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ q_1 \end{bmatrix} \quad (19)$$

Matice transformace rychlosti je pak obdobně:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \frac{\alpha}{2} \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix} \quad (20)$$

Maximální rychlost jádra je dosažením $\omega_0 = \omega_1 = \omega_{max}$ rovna:

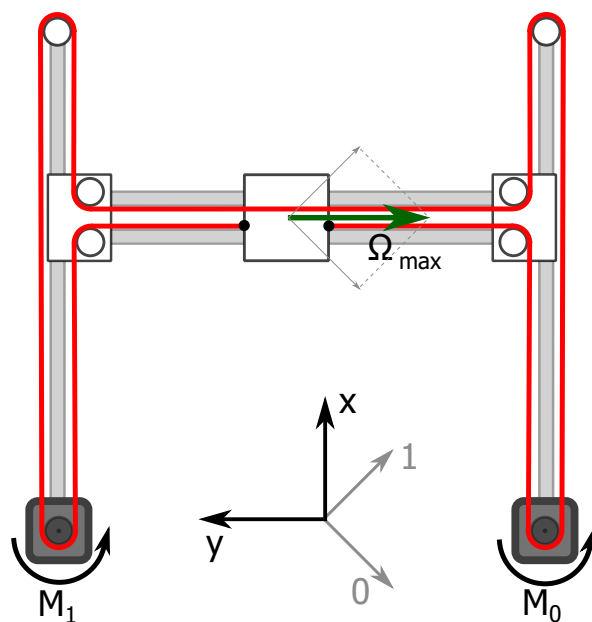
$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \frac{\alpha}{2} \begin{bmatrix} 0 \\ -2\omega_{max} \end{bmatrix} = \Omega_{max} \quad (21)$$

Její velikost je tedy

$$\|\Omega_{max}\| = \frac{\alpha}{2} \sqrt{4\omega_{max}^2} = \alpha\omega_{max} \quad (22)$$

Pohyb jádra při běhu pouze jednoho motoru má maximální velikost $\frac{\alpha}{2}\omega_{max}$, úhlopříčnou orientaci a směr je závislý na směru otáčení ω_{max} .

6.1.3 H-bot



Obrázek 35: Kinematický typ H-bot

Kinematický typ H-bot nabízí konstrukčně velice podobné řešení jako CoreXY, liší se však použitím jediného kus řemene v celku.

Matice transformace pro typ H-bot je totožná jako u CoreXY:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{\alpha}{2} \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ q_1 \end{bmatrix} \quad (23)$$

Matice transformace rychlosti je pak obdobně:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \frac{\alpha}{2} \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix} \quad (24)$$

Maximální rychlost jádra je shodná jako u CoreXY v (21). Její velikost je tedy taktéž:

$$\|\Omega_{max}\| = \frac{\alpha}{2} \sqrt{4\omega_{max}^2} = \alpha\omega_{max}$$

Úhlopříčný pohyb jádra má maximální velikost vektoru rychlosti $\frac{\alpha}{\sqrt{2}}\omega_{max}$. Můžeme formulovat obecný vztah pro rychlosti v_x, v_y (který ale platí i pro zrychlení):

$$|v_x| + |v_y| \leq \alpha\omega_{max} \quad (25)$$

6.1.4 Srovnání

Velkou výhodou kartézského kinematického typu oproti CoreXY a H-bot je teoretická schopnost dosáhnout 2krát vyšší rychlosti v úhlopříčných směrech (poměr $\frac{\sqrt{2}}{1}$). Předností je také snadnější přepočty v matici transformace a tím lepší intuitivní pohled na konstrukci. Velkou nevýhodou je však nestacionární umístění motoru M_0 . To zapříčiní vyšší moment setrvačnosti soustavy a tím pádem vyšší požadavky na odběr proudu. Není tak možné dosáhnout tak vysokého zrychlení jako u typu CoreXY nebo H-bot. Při použití většího motoru se pak tento typ může jevit i jako nebezpečný a může ohrozit člověka pohybujícího se v blízkosti stolu. Další nevýhodou je tzv. *racking*, což je známý pojem při konstrukci 3D tiskáren. Jedná se o vzpírání a vysování příčného ramene, na kterém se pohybuje jádro.

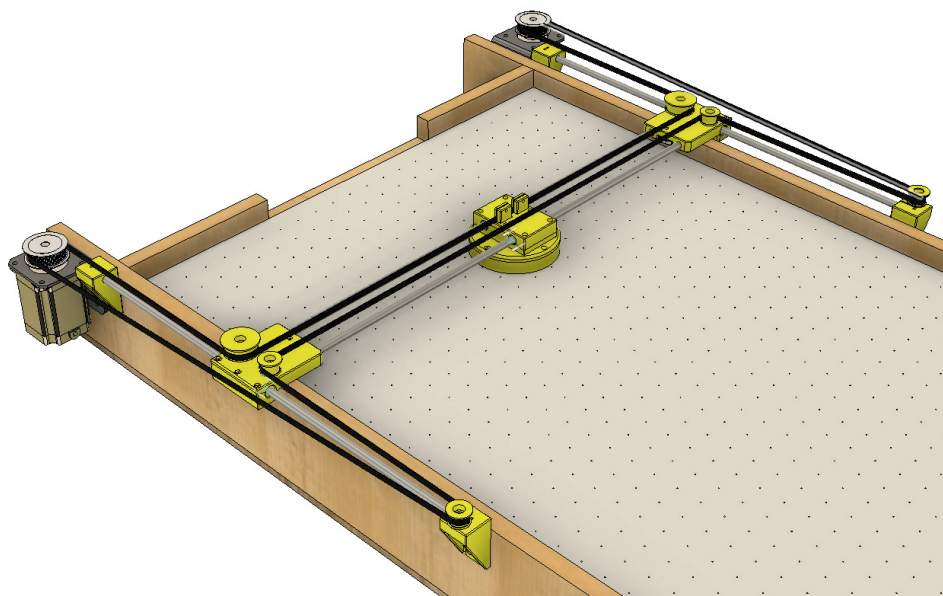
Z těchto důvodů nebyl zvolen kartézský typ.

CoreXY má oproti H-bot mnohem menší racking (ukázkou srovnání vytvořenou v simulačním prostředí Algodo0 lze vidět v [25]), což je hlavní důvod, proč převážná většina konstruktérů 3D tiskáren upřednostňuje tento typ. Pro RAH je výhodou (byť spornou) použití jednoho kusu řemene. Naopak jasnou výhodou typu H-bot je rozmístění řemenů, které nepřechází v polovině hrací plochy jako je tomu u typu CoreXY. Zamezí se tak nechtěnému kontaktu puku s řemeny.

Po zhodnocení těchto předností byl jako kinematický typ vybrán typ H-bot.

6.2 Návrh kinematické konstrukce v 3D modeláři

Nyní budou ukázány stěžejní díly v konstrukci H-bot navržené v programu Autodesk Fusion 360. Iterativní vývoj konstrukce prošel několika modifikacemi, prezentována bude však pouze ta koncová.



Obrázek 36: Model RAH v prostředí Fusion 360

Řemenice

Řemenice je jedním z klíčových dílů zajišťujícím plynulý chod. Sestava se skládá ze šroubu M3 s různými délkami v závislosti na daném místě, kde je řemenice umístěna, ze dvou matic a dvou kuličkových ložisek 623 2Z s vnějším průměrem 10 mm. Řemenice je opatřena na svých krajích lemy, které brání řemenu ve vyklouznutí.

Hokejka

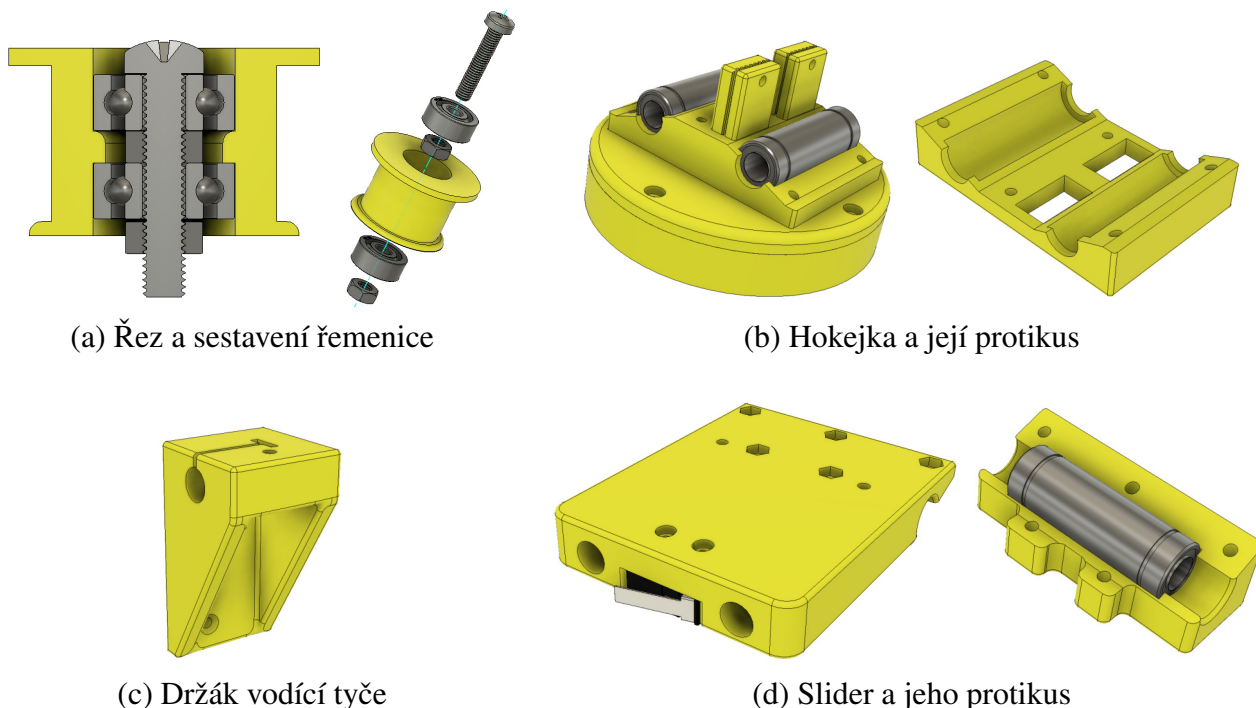
Hokejka se celkově skládá ze 3 dílů - spodní části, střední a protikusy. Střední část se spodní je volně spojena buď čtyřmi šrouby M3 nebo čtyřmi plastovými kolíky, které mohou sloužit jako pojistka při naražení hokejky do puku v blízkosti mantinelu. Je snadné je poté vyměnit za nové. S protikusem je hokejka spojena 2 šrouby. Hokejka má na vrchní části nástavce, do kterých se vsune řemen. Ten se následně zajistí stahovacími páskami, které se provlečou připravenými dírami. Hokejka se pohybuje v ose y po broušených indukčně kalených vodících tyčích osazených dvěma uzavřenými kuličkovými pouzdry LM8LUU.

Držák vodící tyče

Vodící tyče podél stolu jsou upevněny ve dvou držácích, z nichž jeden obsahuje i řemenici. Držáky jsou přichyceny ke stolu pomocí dvou vrtů, což se ukázalo jako dostatečně pevná vazba.

Slider

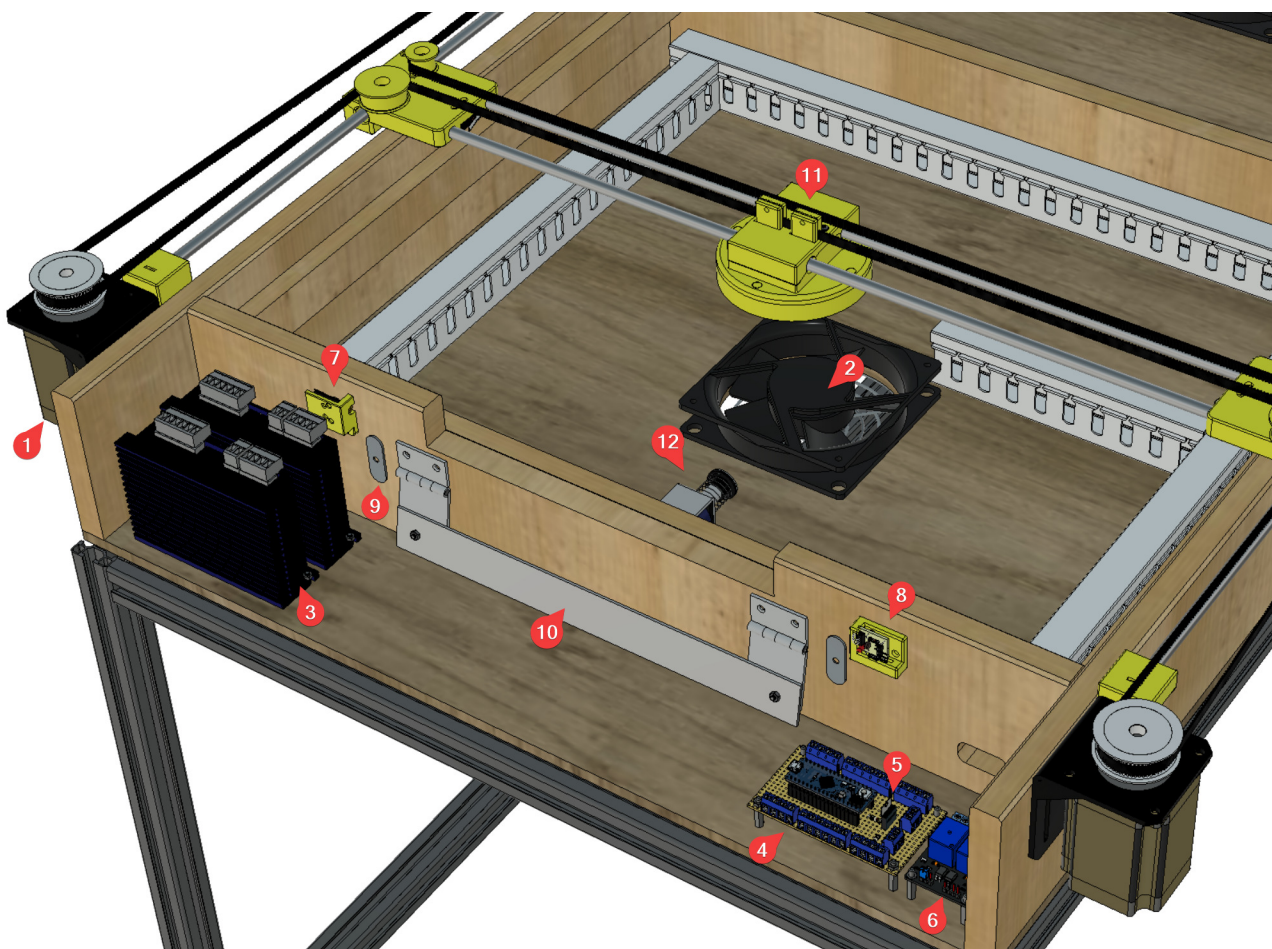
Slider slouží k uchycení vodících tyčí, na kterých se pohybuje hokejka, k tyčím podél stolu. Díry, do nichž jsou umístěny vodící tyče od hokejky, musí být co nejhlubší. Jejich hloubka totiž dost významně zmírňuje *racking*. Mezi dírami na vodící tyči se nachází prostor pro koncový mikrospínač, který slouží k *houmování*⁴, případně k hlídání krajních poloh pro pohyb hokejky. Spínač je přichycen dvěma šrouby M2,5. Dále jsou na slideru dvě díry pro řemenice, z nichž jedna má větší průměr z důvodu zachování souososti. Slider se pohybuje na ložisku LM8LUU, který obepíná protikus spojený 5 šrouby. V protikusu je prostor pro vývod kabelu od koncového spínače.



Obrázek 37: Základní funkční díly RAH

Obrázek 38 pak zobrazuje jednotlivé díly při kompletním zapojení. Na obrázku chybí branková konstrukce pro vyšší přehlednost. Podrobnější popis laserových modulů a odpalovače puku lze nalézt v kapitole 10.

⁴Houmování - proces, při kterém robot inicializuje pomocí dvou koncových spínačů svou polohu (v dané oblasti používané slovo, převedeno z anglického *homing*)



Obrázek 38: Detail dílů na straně robota s odkrytou hrací plochou

Popis dílů na obr. 10:

1. krokový motor
2. ventilátor
3. drivery
4. univerzální pájivé pole s Arduinem
5. MOSFET
6. relé modul
7. laser transmitter modul
8. laser receiver modul
9. kovová destička na přichycení magnetu
10. odpalovač puku
11. hokejka
12. solenoid

7 Elektronika

7.1 Zdroje stejnosměrného napětí

V RAH se nacházejí 3 zdroje stejnosměrného napětí. Ty jsou umístěny v rozvaděči, který je připevněn na hliníkové liště pod stolem. Výstupy zdrojů jsou vyvedena z rozvaděče pomocí 6-žilového kabelu přes mikrofonní konektor XLR189F do spodní desky stolu. Odtud vedou přes kolébkové spínače (umístěné na bočnici) dovnitř stolu, odkud jsou rozvedeny do dalších zařízení.

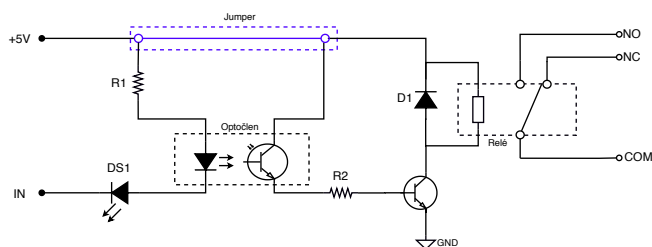
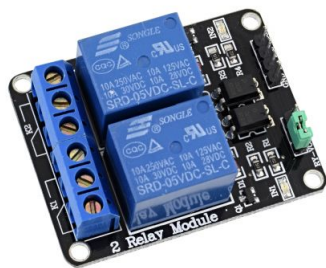
Prvním a také nejvýkonějším zdrojem je RPS369 s výstupním napětím 36 V a výkonem 350 W. Je využito pro napájení driverů krokových motorů.

Dalším zdrojem je TDK-Lambda DPP100-24 s napětíovou výstupní úrovní 24 V a výkonem 100 W. Tato napětíová úroveň slouží k napájení solenoidu, ventilátorů, LED pásků na osvětlení hřiště a osvětlení brankové konstrukce. Tyto prvky jsou podrobněji popsány v kapitole 10.

K napájení Raspberry Pi, ze kterého je dále napájen dotykový displej a Arduino, byl využit 5 V zdroj TDK-Lambda DPP25-5 o výkonu 25 W.

7.2 Spínací prvky

Jako spínací prvek byl pro solenoid a ventilátory využit dvoukanálový relé modul. Solenoid i ventilátory jsou připojeny na NO⁵ kontakty relé modulu.



Obrázek 39: Dvoukanálový relé modul [36]

Na obr. 40 lze vidět zapojení všech prvků na zdroj TDK-Lambda DPP100-24. Pro zjednodušení a přehlednost je relé modul nahrazen jednoduchým spínačem bez elektronického ovládání. Na zátěži indukčního charakteru (ventilátory, solenoid) byly připojeny antiparalelně tzv. *flyback diody*, která vyrovnávají náhlé napětíové špičky při rozepnutí relé.

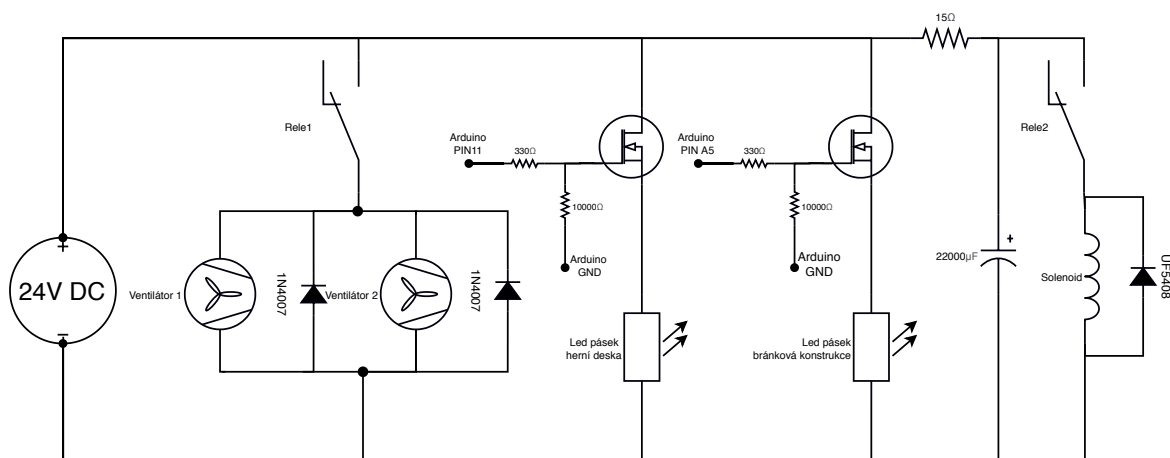
Pro spínání LED pásků na osvětlení stolu a brankové konstrukce byl použit N-kanálový MOSFET IRLZ44N. Elektroda Gate je umístěna do napětíového děliče mezi rezistory 330 Ω (slouží k omezení

⁵NO - Normally Open, při nesepnutém stavu jsou kontakty rozpojeny, při sepnutém stavu jsou spojeny

proudu) a $10\text{ k}\Omega$ (pull-down rezistor k zemi). Pro postupné rozsvěcování a zhasínání LED pásku je MOSFET spínán pomocí PWM⁶.

7.3 Lokální kondenzátor pro solenoid

Solenoid na vystřelení puku je spínán po dobu 50 ms. Protože je však konstruován na nominální hodnotu proudu 8 A a zdroj stejnosměrného napětí 24 V dodá pouze 4,2 A, pro rychlé odpálení puku je tento proud příliš nízký. Navíc LED pásky a ventilátory by neprocházelo po tuto dobu dostatek proudu, tudíž by bylo patrné pohasnutí LED pásku. Toto řeší RC článek s lokálním kondenzátorem, jak lze vidět na obr. 40. Slouží jako lokální zdroj energie pro solenoid.



Obrázek 40: Paralelní zapojení na zdroj 24V DC

Hodnota rezistoru $15\ \Omega$ byla stanovena pomocí Ohmova zákona podle vztahu (26) tak, aby odebíraný proud nepřekročil nominální hodnotu použitého zdroje. Ventilátory mají odběr každý 750 mA a LED pásek má odběr 1000 mA. Zdroj napětí 24 V je tedy schopný dodat pro větev kondenzátoru proud $I = 4,2 - 2 \cdot 0,75 - 1 = 1,7\text{ A}$.

$$R = \frac{U}{I} = \frac{24}{1,7} = 14,1\ \Omega \quad (26)$$

Pro výpočet kapacity kondenzátoru byl použit vztah (27) s poměrně nepřesným předpokladem, že vybíjecí proud kondenzátoru v LC okruhu je konstantní.

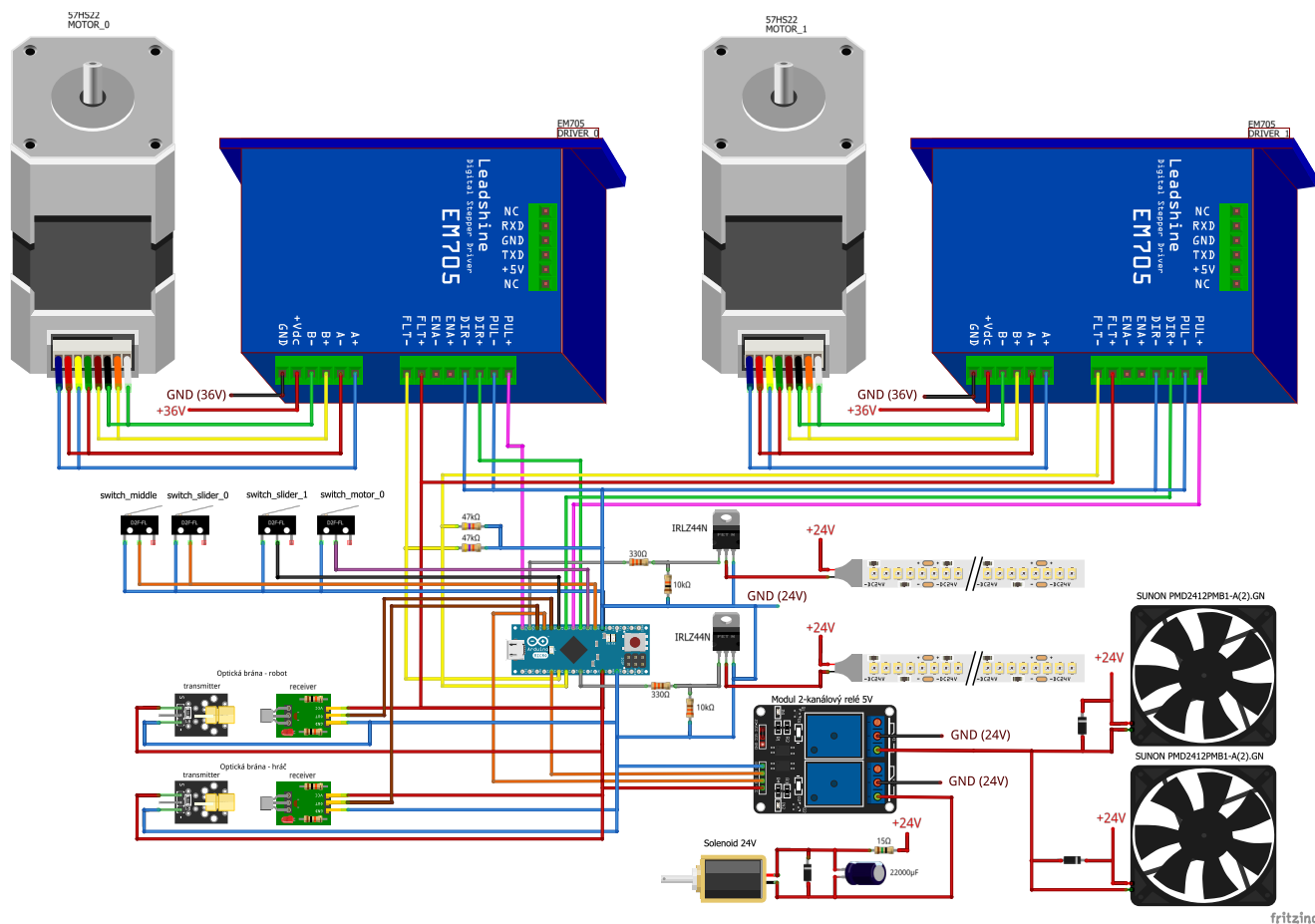
$$C = \frac{I \cdot t}{U} = \frac{8 \cdot 0,05}{24} = 0,0167\text{ F} \quad (27)$$

S ohledem na tuto hodnotu byl použit kondenzátor o kapacitě $22000\ \mu\text{F}$, jež se při praktickém pokus osvědčil (při aktivaci solenoidu nedošlo k pohasnutí světel a puk vystřelil z brány).

⁶PWM - pulzně-šířková modulace

7.4 Zapojení

Na obrázku 41 lze vidět schéma elektronického zapojení RAH. Schéma bylo vytvořeno v programu Fritzing.



Obrázek 41: Schéma elektronického zapojení RAH

Arduino používá interní *pull-up* rezistory na vstupních pinech pro koncové mikrospínače, aby se zamezilo problémům s nedefinovaným napětím na vstupních pinech.

Podle dokumentace výrobce [18] je při bezchybovém stavu je odpor mezi FLT+ a FLT- nízký, v případě chyby pak vysoký. Na vstupní pin Arduino je připojen *pull-down* rezistor k zemi, což tvoří napěťový dělič mezi +5 V a vstupním pinem, a je tak zajištěno správné čtení napětí.

Arduino je z důvodu snadné výměny osazené na dutinkové liště napájené na univerzálním plošném spoji. Na tomto plošném spoji se nachází také oba tranzistory a všechny potřebné rezistory (s výjimkou 15 Ω rezistoru na solenoid). Fotografii této desky lze vidět v příloze na obr. 51.

8 Řízení motorů

V této kapitole bude popsán způsob řízení motorů. Bude představen matematický popis kinematického typu H-bot, základní S-křivka, poté bude popsáno její generování. Následně bude toto řízení modifikováno použitím P regulátoru pro hladší průběh zastavení a snazší implementaci v simulaci.

Pro Arduino existuje řada knihoven pro řízení driverů krokových motorů, uved' me například hojně využívanou knihovnu AccelStepper [34]. Žádná z nich však neprovádí časování kroků pomocí interních časovačů, ale provádí je pomocí aktivního čekání (pomocí funkce *delayMicroseconds*), což je pro účely RAH velmi nevhodné (ovládání jediného motoru zcela vyčerpává veškerý výpočetní výkon Arduina). Z toho důvodu jsou motory řízeny s využitím časovačů.

8.1 Matematický popis kinematického typu H-bot

Důležitými vztahy jsou převody jednotek z kroků motoru na milimetry a naopak, jež jsou dány geometrií a parametry typu H-bot a popisuje je vztah (28).

$$1 \text{ mm} = 1 \text{ krok} \cdot \alpha, \quad (28)$$

kde α je H-bot konstanta, pro kterou platí vztah (29):

$$\alpha = \frac{N \cdot s}{n}, \quad (29)$$

kde N je počet zubů řemenice, s je rozteč zubů v jednotkách mm a n je počet kroků na jednu otáčku, který je dán zvoleným mikrokrokováním. V případě RAH byla použita řemenice s 60 zuby s roztečí 2 mm a rozlišení mikrokrokování 400 kroků na jednu otáčku. Pro výpočet polohových souřadnic x, y pak podle (23) a (24) platí:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{\alpha}{2} \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ q_1 \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}, \quad (30)$$

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \frac{\alpha}{2} \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix}, \quad (31)$$

kde q_0, q_1 jsou počty kroků motorů M_0, M_1 , x_0, y_0 je pevný offset, v_x, v_y jsou rychlosti v osách x, y a ω_0, ω_1 jsou rychlosti motorů v jednotkách krok/s. Inverzním vyjádřením vztahu (31) dostáváme (32).

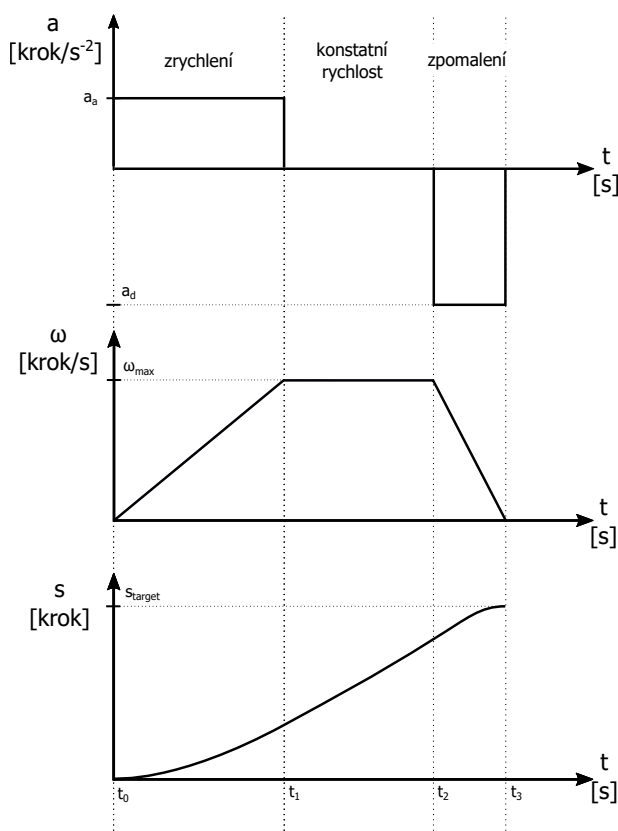
$$\begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix} = \frac{2}{\alpha} \begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad (32)$$

8.2 S-křivka

Kvůli fyzikálním omezením není možné dosáhnout žádané rychlosti okamžitě, nýbrž je zapotřebí této hodnoty dosáhnout daným zrychlením, které je omezeno maximálním točivým momentem motoru, pasivními odpory a setrvačností pohybujícího se mechanismu.

Z tohoto důvodu je běžné generování žádané polohy pomocí tzv. S-křivky. Ta se skládá ze dvou polynomiálních křivek druhého řádu (fáze zrychlení a zpomalení) a (nepovinně) polynomiální křivky prvního řádu (fáze udržení konstantní rychlosti). Tomuto průběhu žádané polohy odpovídají průběhy rychlosti a zrychlení znázorněné na obr. 42.

Tento způsob „řízení“ (kdy regulátor je principiálně zahrnut v motoru) byl využit i v RAH. Pro dosažení maximálních výkonů bylo rozlišováno mezi zrychlením a zpomalením. Zpomalení může totiž díky působení tření dosahovat vyšších hodnot.



Obrázek 42: S-křivka pro spojité řízení

8.3 Online generování S-křivek

Online generování S-křivek v RAH spočívá v nalezení optimálního časového okamžiku, ve kterém je třeba zahájit brzdění tak, aby bylo při zastavení dosaženo cílové polohy. Tento algoritmus sice

vykazuje překmity polohy od úsečky spojující aktuální a žádanou polohu, takže může nastat kolize s mantinelem. Tuto problematiku však řeší kapitola 10.3. Slovo online v názvu kapitoly vyjadřuje fakt, že se S-křivky neustále přepočítávají v závislosti na nové požadované poloze přijaté z RPi.

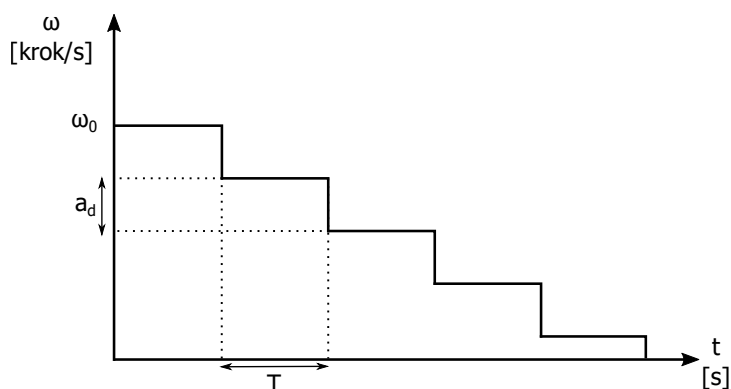
Výše zmiňovaný nejpozdější časový okamžik pro zahájení brzdění je na obr. 42 znázorněn jako čas t_2 . Rychlosti, kterými se motory otáčejí v tomto čase budeme nazývat *kritickým stavem*.

Obdržená poloha z RPi se nastaví jako požadovaná poloha. Připomeňme, že RPi posílá data do Arduina s frekvencí 150 Hz. V každém cyklu se pak vyjádří vektor posunutí od požadované polohy $v = [v_x, v_y]$, který je dále uvažován jako vektor rychlostí. Z rovnice (32) se následně přepočítá na požadované rychlosti motorů ω_0, ω_1 . Jak již bylo zmíněno, toto je nutné provádět v každé smyčce `loop()` protože z předchozích pohybů má robot často nenulovou rychlost, tudíž je třeba upravovat vektor rychlostí $[v_x, v_y]$ tak, aby vždy směřoval k žádané poloze. Důsledkem toho je, že dochází k polohovým překmitům. Z toho důvodu je implementováno automatické brzdění před mantinelem, které je popsáno v kapitole 10.3.

Motory pak poměrově⁷ zrychlují do doby, kdy jeden z motorů dosáhne maximální rychlosti ω_{max} (viz obr. 42), případně do dosažení *kritického stavu*.

V každé smyčce `loop()` je zároveň ověřeno, zda je dosaženo *kritického stavu*. Je tedy vypočítána teoretická brzdná dráha a pokud je dostatečně blízko odchylce od požadované veličiny, motory začnou synchronně poměrově brzdit. Následující řádky se budou zabývat právě oním výpočtem brzdné dráhy.

Pro odvození brzdné dráhy využijeme čas jedné smyčky `loop()`, který označíme jako T a zvolenou hodnotu decelerace jako a_d . Aktuální reálnou rychlost, kterou se motor otáčí, označíme ω_0 . Na obrázku 43 můžeme vidět ilustraci zastavení motoru celkem v 5 cyklech funkce `loop()`.



Obrázek 43: Graf otáček při zastavení motoru

⁷poměrové zrychlování/brzdění motorů - takové hodnoty zrychlení/zpomalení motorů, při kterých se robot pohybuje ve směru vektoru $[v_x, v_y]$ k požadované poloze

Obecně lze počet požadovaných smyček k zastavení c vyjádřit podle vztahu (33)

$$c = \left\lceil \frac{\omega_0}{a_d} \right\rceil \quad (33)$$

V k -té smyčce je reálná rychlost motoru ω_k a tedy se vykoná $\omega_k T$ kroků. Počet vykonaných kroků do úplného zastavení motoru označíme jako s a určí se podle vztahu (34). Zde je pro zjednodušení použita aproximace rychlosti v poslední smyčce $\omega_c = \omega_0 - a_d(c - 1) \approx a_d$.

$$s = \underbrace{a_d T + 2a_d T + 3a_d T + \dots + ca_d T}_{c\text{-krát}} = a_d T \sum_{k=1}^c k \quad (34)$$

Dostáváme tak výraz obsahující součet c členů aritmetické posloupnosti. Použitím známého vzorce na sumu těchto c členů dostáváme vztah (35).

$$s = \frac{a_d T c (1 + c)}{2} \quad (35)$$

Tímto jsme vyjádřili počet vykonaných kroků jednotlivých motorů do úplného zastavení. Využitím vztahu (30) pak lze vypočítat brzdnu dráhu, která se poměří s odchylkou od požadované hodnoty a tím se ověří dosažení *kritického stavu*.

8.4 P regulátor

Diplomová práce Ondřeje Slámy [40] se zabývá strategií automatického hráče a pro testování byla vytvořena počítačová simulace hry. Programové zreplicování výše popsaného online generování S-křivek je v simulaci zbytečně obtížné, proto byl navržen modifikovaný algoritmus řízení, který je snazší na implementaci. Navíc vyhlazuje křivky rychlosti při fázi brzdění.

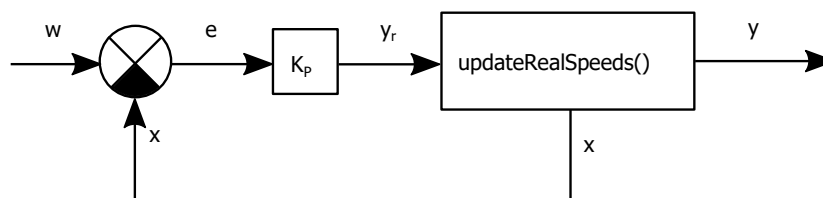
Prostý P regulátor by však nebylo možné použít. Bylo zapotřebí omezit maximální akční veličinu (rychlost nebo-li frekvenci kroků motoru). Toto omezení rychlosti vedlo na chování shodné s druhou fází S-křivky - udržováním konstantní rychlosti. Další nezbytnou úpravou regulátoru bylo přidání omezení strmosti hran (tzv. *rate limiting*), přesněji pouze „zrychlujících“ hran. To je dáno maximální silou, kterou motor dokáže v vyvinout. Tato úprava vede na chování odpovídající první fázi S-křivky - fázi konstantního zrychlování. V poslední fázi, fázi konstantního zpomalování, již přebírá kontrolu P regulátor. Tato fáze nastává na obr. 45 v čase t_{REG} . Průběh této fáze se oproti ideální S-křivce odlišuje.

Akční veličinu rychlosti lze regulátorem vypočítat pomocí vztahu (36).

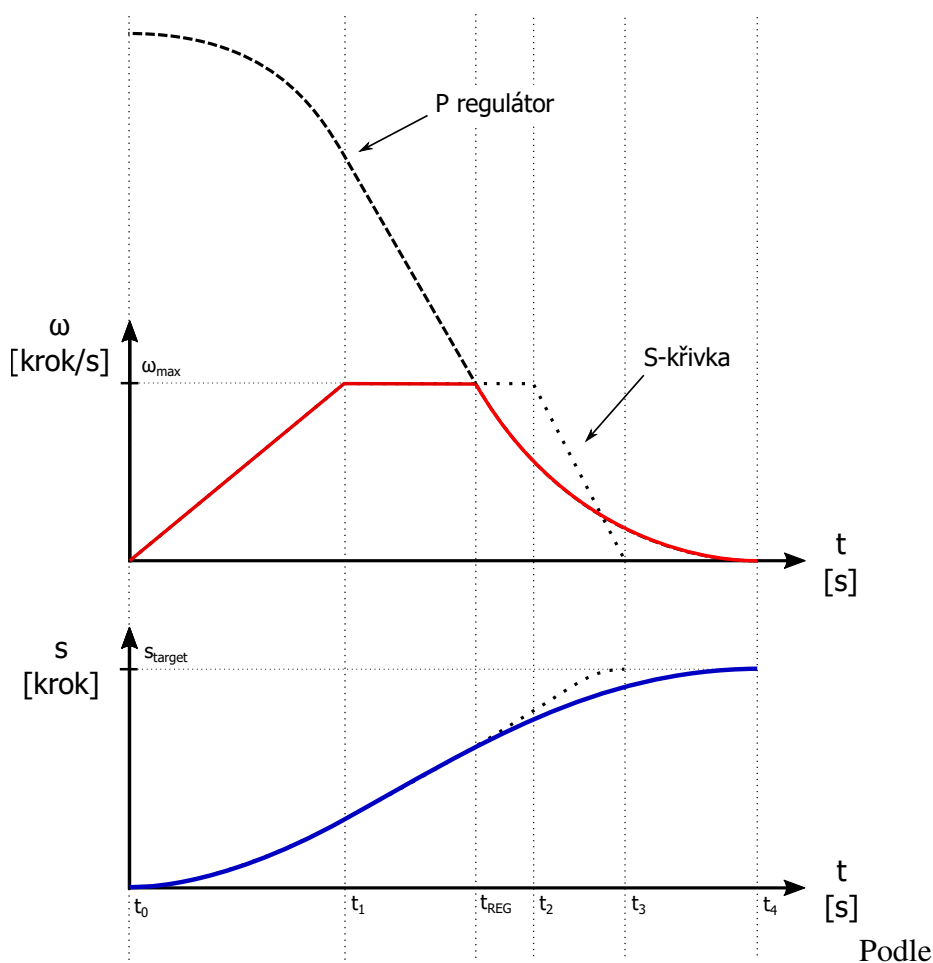
$$\omega = K_P \cdot (s_{\text{target}} - s) \quad (36)$$

Schéma regulátoru je pak možné vidět na obr. 44, kde K_P je gain, ω značí žádanou polohu a podle

(36) odpovídá s_{target} , e je regulační odchylka a odpovídá $(s_{\text{target}} - s)$, y_r je akční veličina regulátoru, tedy požadovaná rychlost motoru ω . Tato hodnota vstupuje do funkce `updateRealSpeeds()`, ve které je aplikována výše popsaná saturace a rate limiting. Tato funkce taktéž zajistí výpočet aktuální polohy x (odpovídá hodnotě s), která slouží jako zpětná vazba. Výstupní veličina y je reálná rychlost motoru. V kapitole 9.2 je pak ukázáno, jak je tato reálná rychlost převedena na odesílání signálů do driveru. Výsledné křivky rychlosti a polohy jsou na obr. 44 vyznačeny barevně.



Obrázek 44: Zjednodušené schéma interního P regulátoru



Obrázek 45: Srovnání spojitého řízení polohy P regulátorem s S-křivkami

Využitím tohoto přístupu bylo jednodušší implementovat v simulaci pohyb robota (tuto část řeší práce Ondřeje Slámy [40]) a také se posun na požadovanou polohu stal plynulejším, díky plynulejšímu úbytku rychlosti motorů (viz obr. 45). Je zřejmé, že se robot na požadovanou polohu dostane za delší čas. Vzhledem k dosahovaným maximálním rychlostem a ve srovnání s výše zmíněnými benefity to však to není považováno za zásadní překážku.

9 Softwarová implementace

RAH je řízen pomocí mikrokontroleru ATmega32u4 na desce Arduino Micro. Tento typ byl zvolen kvůli požadavku na dva 16-bitové časovače [32, str. 92 -176], kdy každý časovač bude běžet pro jeden motor. Arduino Uno má například takový časovač pouze jeden, na druhou stranu má dva 8-bitové časovače, které by šly taktéž využít, ale nepřesnost časování odesílání signálů do driverů krokových motorů by byla vyšší (tedy rozlišení reálné rychlosti by bylo nižší).

Tato kapitola se bude věnovat zevrubnému popisu softwarové implementaci řízení. Program byl napsán v jazyce C/C++, který se jakožto kompilovaný jazyk vyznačuje svou rychlostí. Bylo využito vývojové prostředí Arduino IDE. Kompletní kód lze nalézt v příloze.

9.1 Struktura programu

Struktura kódu v Arduinu se skládá ze dvou funkcí - `setup()` a `loop()`. Funkce `setup()` je zavolána pouze poprvé po spuštění Arduina a před zavoláním funkce `loop()`. Po vykonání funkce `setup()` je volána funkce `loop()` v nekonečné smyčce.

Ve funkci `setup()` se definují jednotlivé piny jako vstupy nebo výstupy a jejich výchozí hodnoty. Dále se nastaví piny PE6, PD0 a PD1 (odpovídá digitálním pinům 7, 2, 3) jako piny externího přerušování, které zavolají funkci `checkSwitch()` při hodnotě pinu LOW. Tyto piny jsou interně připojené přes tzv. *pull-up* rezistor k napětí +5V, což zabráni tomu, aby se pin choval jako tzv. *floating pin*, což by znamenalo, že se na něm snadno indukuje okolní šum a neočekávaně je na něm možné měřit logickou hodnotu 1.

Dále jsou ve funkci `setup()` nastaveny 16-bitové časovače motorů Timer 1 a Timer 3 s hodnotami prescaleru 8, což znamená, že časovač se inkrementuje 8x pomaleji (viz kap. (4.2.3)). Dále je pak nastaven 10-bitový časovač Timer 4 s hodnotou prescaleru 512 a porovnávací hodnotou 800. Tato porovnávací hodnota se na rozdíl od časovačů 1 a 3 nemění a odpovídá frekvenci 125 Hz. Funkce zavolaná tímto časovačem obstarává blikání LED pásku v brankové konstrukci a odesílání dat to RPi. Tato data jsou ve tvaru (37):

$$[p_x], [p_y]; [v_x], [v_y], \quad (37)$$

kde $[p_x]$ resp. $[p_y]$ jsou polohové souřadnice robota a $[v_x]$ resp. $[v_y]$ jsou rychlosti robota.

Po vykonání funkce `setup()` je volána funkce `loop()` v nekonečné smyčce. Zjednodušený vývojový diagram této funkce lze vidět na obr. 46. Nejdříve je zkontrolována dostupnost nových dat z RPi po sériové lince. Příkazy se řídí jednoduchým protokolem popsáným v tabulce 4. V případě dostupných dat je proveden daný příkaz. Následně se provede výpočet aktuální polohy robota. Dále jsou ověřeny hodnoty pinů pro signál chyby z driverů.

Chyba (error) může vzniknout třemi způsoby:

1. nezhroumovaný stav
2. nezapojený driver; ztráta kroku; nadproudová ochrana driveru
3. sepnutí koncového spínače.

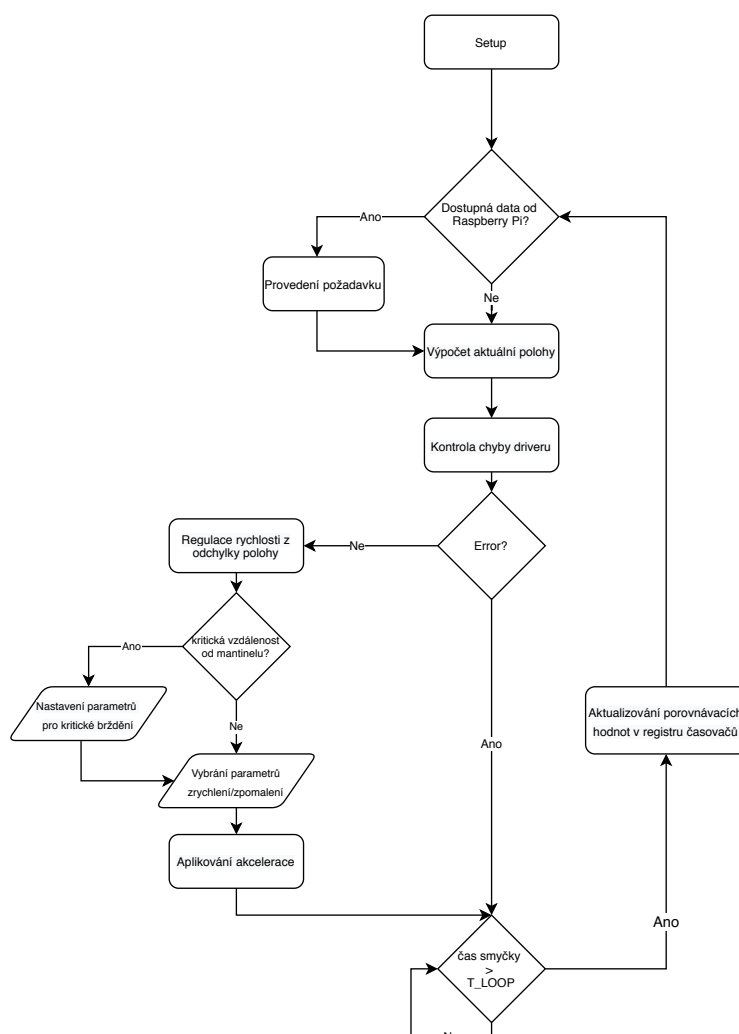
V takovém případě je vyžadován příkaz na houmování robota. Houmování je provedeno ve dvou krocích, kdy se aktivují jednotlivé koncové spínače a tím robot zjistí svou polohu.

V případě neexistence chyby se vypočítá požadovaná rychlost motorů a poté se aktualizuje proměnná reálné rychlosti. Následně program vyčká do uplynutí pevně stanoveného času jednoho cyklu funkce `loop()` kvůli časově ekvidistantním výpočetním krokům. Poté přepočítá reálnou rychlost motorů na porovnávací hodnoty časovačů, pomocí nichž se odesílají signály do driverů (viz kap. 9.2).

V případě interního přerušení způsobené časovačem dojde k vykonání funkce ISR, poté se program vrací zpět na místo, kde byl přerušen (viz obr. 26). Přerušení může nastat buď časovačem 4, který slouží primárně k posílání dat do RPi, nebo časovači 1 a 3, které slouží k řízení rychlostí motorů a posílají signály do driveru (viz 9.2). Tento TTL signál o logické hodnotě 1 musí mít podle [18] dobu trvání alespoň 5 μ s. K tomu je vícenásobně využit prázdný příkaz *nop* pro mikročipy AVR, který zajistí krátký delay 62,5 ns.

příkaz	popis
home	provede houmování robota
p, [x], [y]	nastaví žádanou polohu $x = [x], y = [y]$ v jednotkách mm/s
v, [x], [y]	nastaví žádanou rychlost $v_x = [x], v_y = [y]$ v jednotkách mm/s
m, [x], [y]	nastaví žádanou rychlost motorů $\omega_0 = [x], \omega_1 = [y]$ v jednotkách krok/s
setaccel, [x]	nastaví akceleraci na hodnotu [x]
setdecel, [x]	nastaví deceleraci na hodnotu [x]
setmaxspeed, [x]	nastaví maximální rychlost na hodnotu [x]
kpgain, [x]	nastaví parametr regulátoru K_P na hodnotu [x]
leds, [x]	nastaví jas LED pásků na osvětlení hřiště na hodnotu [x], kde $x \in [0, 255]$
fans, [x]	zapne nebo vypne ventilátory, x je logická hodnota 0 nebo 1
blink	zabliká LED pásy v brankové konstrukci
solenoid	vystřelí solenoid

Tabulka 4: Konvence příkazů pro ovládání Arduina



Obrázek 46: Stručný vývojový diagram funkce loop()

9.2 Řízení signálů do driverů

V předchozí kapitole 8 bylo popsáno jakým způsobem je počítána reálná rychlost motorů. V této kapitole bude na toto navázáno a bude popsáno, jakým způsobem se z reálné rychlosti vypočítá hodnota registru OCRnA, který ovlivňuje to, s jakou frekvencí se posílají signály do driverů na provedení kroku pro jednotlivé motory.

Frekvence časovačů Timer1 a Timer3 je

$$\tau = \frac{f_{CPU}}{\text{prescaler}} = \frac{16\text{MHz}}{8} = 2\text{MHz}$$

Z žádané rychlosti ω_i pro i -tý motor se pak požadovaný počet inkrementujících ticků λ_i určí jako:

$$\lambda_i = \frac{\tau}{\omega_i} \quad (38)$$

Maximální hodnota časovače je $2^{16} - 1$. Pro případ, kdy je hodnota λ_i větší než tato maximální hodnota (to nastane tehdy, když je rychlost motoru dostatečně malá), definujeme podle vztahu (39) proměnnou m_i , která vyjadřuje počet prázdných smyček časovače bez provedení odeslání signálů do driveru. V praktické implementaci je pak použita proměnná N , která inkrementuje svou hodnotu při každém takovém prázdném cyklu. V cyklu, ve kterém se odešle signál, se tato hodnota resetuje.

$$m_i = \left\lfloor \frac{\lambda_i}{(2^{16} - 1)} \right\rfloor \quad (39)$$

Porovnávací hodnota se pak určí podle vztahu (40).

$$\eta_i = \lambda_i \bmod (2^{16} - 1) \quad (40)$$

A tedy platí vztah (41).

$$\lambda_i = m_i \cdot (2^{16} - 1) + \eta_i \quad (41)$$

Díky zavedení těchto proměnných je možné i s využitím prescaleru řídit motor libovolně malou rychlostí (to například v práci Davida Kopeckého na ČVUT [5] ani u firmy JJRobots [6] možné není). Popis funkce řízení signálů popisuje následující pseudokód. Pro zjednodušení vynecháváme i -té indexy. Signály do motoru 0 řídí Timer3, motor 1 pak naopak Timer1, tedy $n = 3$ odpovídá index $i = 0$ a $n = 1$ odpovídá $i = 1$.

Algoritmus 1 ISR funkce n -tého časovače pro konkrétní motor

```

1: function ISR()
2:   if směr otáčení == 0 or error then
3:     return
4:   if  $N == m$  then
5:     PUL PIN driveru HIGH
6:     delay 3 $\mu$ s
7:     PUL PIN driveru LOW
8:     if  $m == 0$  then registr OCRnA =  $\eta$  else registr OCRnA =  $2^{16}$ 
9:     N=0
10:  else
11:    N++
12:    if  $N == m$  then
13:      register OCRnA =  $\eta$ 

```

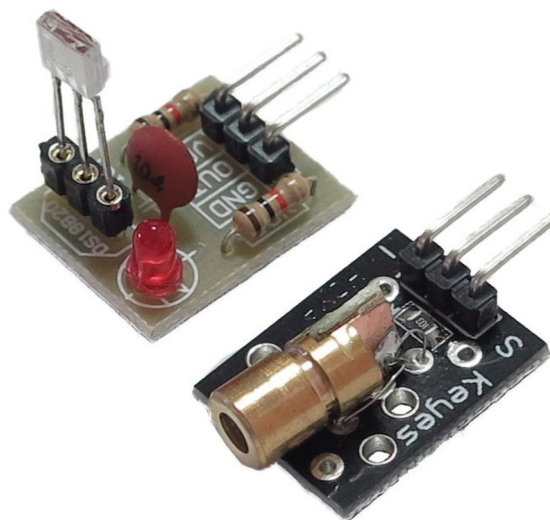
10 Doplnující prvky

Ve výše zmíněných kapitolách byla zmíněna optická brána na detekci gólu, solenoid a osvětlení. V této kapitole bude těmto prvkům věnován větší prostor a budou popsány některé další prvky vylepšující herní možnosti RAH.

10.1 Optická brána na detekci gólu

V RAH jsou použity na detekci gólu na obou stranách hrací plochy optické brány, tvořené Arduino laserovými moduly Keyes KY-008. Ty pracují při napětí 5 V, tudíž není potřeba nijak konvertovat signál a bývají často používány v kombinaci s Arduinem. Princip je jednoduchý, pokud puk přeruší laserový paprsek, výstupní napětí na přijímači je LOW, jinak HIGH.

Výhodou těchto modulů je jejich cena řádově v desítkách korun, malé rozměry a snazší montáž z důvodu viditelného paprsku (na rozdíl od infračerveného záření). Jsou umístěny za brankou v úrovni výšky hrací plochy. S ohledem na málo volného prostoru za brankou byl navržen a vytisknut držák na oba moduly tak, aby nebránil v umístění ostatním komponentům v RAH (displej, drivery, konstrukce branky).



Obrázek 47: Laserové moduly - receiver a transmitter [37]

Ačkoliv byl standardní počet pinů externího přerušení v Arduinu Micro vyčerpán (koncovými spínači), byly konfigurovány piny detekující změny laserového přijímače jako piny externích interruptů pomocí AVR příkazů:

```
PCICR |= 0b00000001; // turn on port b
```

```
PCMSK0 |= 0b01010000; // turn on pins PB6 and PB4
```

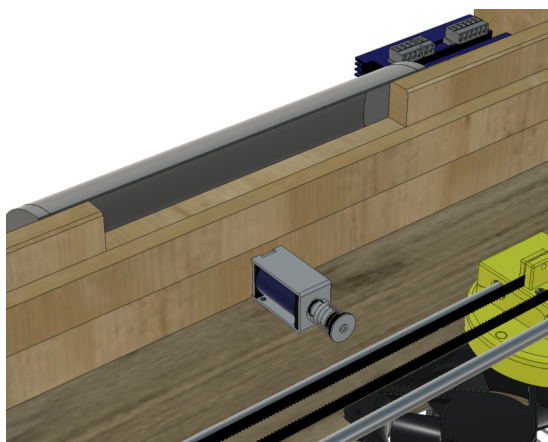
Použitý laser v transmitteru Keyes KY-008 má výkon 5 mW a vysílá záření o vlnové délce 650 nm (červená) [37].

Základem přijímacího modulu je integrovaný obvod „ISO203 Laser receiver“. Ten obsahuje fotodiodu se zesilovačem, tranzistor a stabilizátor napětí. Přijímač je nutné používat ve stinném prostředí, protože reaguje na viditelné světlo. Podrobnější popis tohoto modulu lze nalézt v [38].

Tyto moduly je možné vidět na obr. 38 jako díly 7 a 8.

10.2 Automatické vystřelování puku z brány

Automatické vystřelení puku z branky robota je důležitou funkcí RAH. Přináší komfortnější podmínky pro hru, protože není nutné obcházet stůl při každém vstřeleném gólu a taktéž není nutné mít pro hru více puků.



Obrázek 48: Solenoid uvnitř stolu

Vystřelování puku je zajištěno solenoidem na 24 V s odběrem 8 A. Dostatek energie zajišťuje lokální kondenzátor s kapacitou 22000 μF , jehož funkce je popsána v kapitole 7.3. Na solenoid je antiparalelně připojena dioda UF5408. Pro odpálení puku spíná Arduino relé modul po dobu 50 ms.

Branková konstrukce je vyrobena z trubky z materiálu PVC-U a její boční stěny jsou vyrobeny z překližky. Na vnějších bocích jsou připevněny magnety, pomocí nichž se lze tuto konstrukci přichytit ke kovovým destičkám, které jsou připevněny na zadní desce stolu (viz obr. 38 díl 9).

Solenoid je umístěn uvnitř stolu pod brankou a jeho feromagnetické jádro směřuje ven. Zastává funkci lineárního aktuátoru. Při sepnutí relé začne cívkou téct proud a začne se vytvářet magnetické pole. Tím se feromagnetické jádro uvede do pohybu směrem k odpalovači, který je vidět na obr. 38. Poté udeří odpalovač, který přes rotační vazbu v pantech vystřelí puk zpět na hrací plochu. Protože branková konstrukce má větší délku než samotná branka (z důvodu umístění LED pásek na osvětlení branky, viz kapitola 10.4), jsou do brankové konstrukce na úrovni konců branky přilepeny zábrany z plexiskla.

Mírnou nevýhodou je vyšší hlučnost samotného solenoidu, což by šlo eliminovat vložením tlumících podložek na konec jeho jádra.

Video ukazující automatické vystřelení puku z branky je k nalezení v příloze.

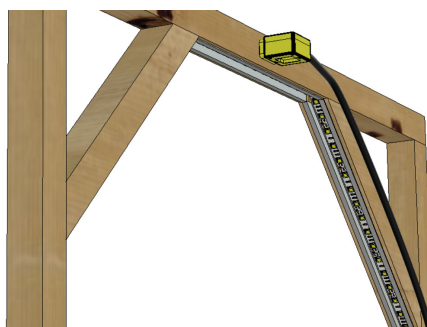
10.3 Automatické zastavení před mantinelem

V kapitole 8.4 o řízení pohybu P regulátorem byl jako jedna z nevýhod uveden možný překmit. Z toho důvodu je možné v Arduino aktivovat tzv. *caution mode*. Ten zajistí to, že Arduino v každém cyklu smyčky `loop()` spočítá průsečík aktuálního vektoru rychlosti s mantinely, následně vypočítá teoretickou brzdnu dráhu a porovná, zda je větší. Tato brzdna dráha je počítána pomocí algoritmu popsaneho v kapitole 8.3. V kladném případě zahájí Arduino nouzové brzdění, aby předešlo nárazu do mantinelu. Pro případ náhlého selhání je konstrukce robota je vybavena 4 koncovými mikrospínači, které plnily svou funkci převážně ve fázi testování robota.

10.4 Osvětlení

Pro zajištění dobrých světelných podmínek pro kameru byly do dřevěné stojné konstrukce pro kameru nainstalovány lišty s LED pásky o výkonu 24 W s barvou denního světla. Ty jsou dostatečně výkonné, takže i při úplné tmě je možné kamerou rozpoznat puk. Barevná korekce je vyladěna pomocí automatické vyvážení bílé barvy (detailní popis algoritmu hledání shluků bílých bodů používající K-means lze nalézt v práci Ondřeje Slámy [40]). Pomocí MOSFETů a PWM lze řídit intenzitu osvětlení.

Druhé využití LED pásek je na signalizaci vstřeleného gólu hráčem. Na kovové plíšky (obr. 38 díl 9) je přivedeno stejnosměrné napětí 24 V. Přes kovové plíšky na magnetu je toto napětí přivedeno na šroub držící magnet na boku konstrukce a odtud již na LED pásky, které jsou umístěny na vnitřní straně brankové konstrukce. Na druhém kovovém plíšku je pak přivedena elektroda Drain řídicího MOSFETu. Bezspornou výhodou tohoto zapojení je odnímatelná branková konstrukce. V případě, že by se v ní puk neočekávaně vzpříčil, je velmi snadné konstrukci odejmout a poté zpětně připevnit na magnety. Spínání řídicího MOSFETu je řešeno uvnitř ISR funkce pro odesílání dat to RPi.



Obrázek 49: Osvětlení stolu LED pásky

11 Závěr

Obsahem diplomové práce byl návrh a realizace robotického hráče Air hockey. Nejdříve byly představeny dosavadní nejvýznamnější projekty v této oblasti. Teoretická část byla majoritně zaměřená na popis krokových motorů včetně způsobů jejich řízení, minoritně pak na popis Arduina a UART komunikace s Raspberry Pi.

V praktické realizaci byl pak navržen v prostředí Fusion 360 model robotického Air hockey, který je možné si interaktivně prohlédnout online na odkazu [41]. Byly navrženy dva typy herního stolu, z ekonomických důvodů byl realizován stůl dřevěný. Návrh automatického hráče prošel několika iteracemi, prezentován byl pouze ten konečný.

Pro konstrukci robotického hráče byl vybrán kinematický typ H-bot, potřebné nestandardní díly byly vytisknuty na 3D tiskárně. Možné vylepšení konstrukce automatického hráče spatřuji ve využití tužších vodících tyčí s průměrem alespoň 12 mm (současně použit průměr 8 mm).

Krokové motory jsou buzeny drivery EM705. Pro řízení polohy byl navržen modifikovaný P regulátor doplněný o saturaci a rate limiter, jež vykazuje chování podobné řízení pomocí S-křivek. K těmto výpočtům slouží vývojová deska Arduino Micro. Řídicí signály jsou naprogramovány pomocí interních přerušení časovačů, ovládaných přímou manipulací uživatelských funkcí s příslušnými registry časovačů, protože uvažované dostupné knihovny na řízení krokových motorů se ukázaly být nevyhovující. Arduino je společně s ostatními elektronickými komponenty vsazené do univerzálního pájecího pole. Ventilátory a solenoid jsou spínány pomocí relé modulu, LED pásky pak pomocí N-kanálových MOSFETů. Robotický Air hockey je napájen třemi zdroji s výstupními napětími 36 V, 24 V a 5 V, které jsou umístěny v rozvaděči pod stolem.

Arduino komunikuje po sériové lince s Raspberry Pi, jehož úkolem je snímat polohu puku a následně vyhodnocovat vhodný pohyb. Touto problematikou se zabývá práce Ondřeje Slámy [40].

Pro komfortnější hru byly přidány doplňující prvky - optická brána na detekování vstřeleného gólu, automatické vystřelování puku z branky a osvětlení stolu a brankové konstrukce.

Při vysokých hodnotách zrychlení vyhodnocoval driver pomocí bezsenzorové detekce ztráty kroku chybu. Proto by bylo vhodné použít silnější motory, případně použít řemenici s 80 zuby (namísto současných 60). Použitím výkonnějšího počítače by bylo také možné přepočítávat požadovanou rychlost motorů častěji, tudíž pracovat s nižší hodnotou zrychlení mezi dvěma aktualizacemi požadované rychlosti (tedy plynulejším průběhem rychlosti), což sníží riziko ztráty kroku. Jako vhodný počítač malých rozměrů považuji Arduino kompatibilní vývojovou desku Teensy 3.6 s procesorem 180 MHz ARM Cortex-M4 (cena 1066 Kč [39]).

Časově nejnáročnější částí této diplomové práce byla praktická realizace celého projektu včetně odladění. Důraz byl kladen především na plug&play bezobslužnou funkcionalitu. Výsledný produkt je snadné rozložit či složit, převézt a pohodlně jej může ovládat i laik.

Robotický hráč má herní schopnosti na poměrně dobré úrovni a je schopen úspěšně konkurovat

průměrnému hráči (to dokazuje video, které lze nalézt v příloze). Vyniká zvláště v defenzivní stránce hry a často donutí lidského oponenta vstřelit si vlastní gól. V porovnání s ostatními projekty RAH, které byly v práci zmíněny, se výsledný RAH jeví jako velmi úspěšný.

Pořizovací náklady na RAH jsou 20000 Kč. V této částce jsou zahrnuty všechny komponenty včetně podstavného stolu z hliníkových profilů.

Fotografie reálného modelu robotického Air hockey lze nalézt v příloze na konci práce, odkaz na video zachycující samotnou hru je pak v přiložených souborech práce.

Seznam obrázků

1	Air hockey s průmyslovým robotem firmy Nuvation [3]	17
2	RAH řízený počítačem MinnowBoard Max s Windows 10 IoT Core [3]	18
3	První verze RAH od JJrobots s kartézským kinematickým typem [4]	19
4	Druhá verze RAH od JJrobots s typem H-bot [6]	20
5	Kartézské rozvržení motorů v práci Davida Kopeckého [5]	20
6	Náhrada krokového motoru RL obvodem [8]	22
7	Proud v cívce při skokové změně napětí [8]	23
8	Krokový motor s proměnnou reluktancí [9]	24
9	Krokový motor s rotorem z permanentního magnetu [9]	25
10	Krokový hybridní motor [9]	25
11	Konstrukce 2-fázového a 5-fázového krokového motoru [12]	26
12	Zapojení fázových vinutí 2-fázového motoru [9]	27
13	Unipolární buzení [9]	28
14	Bipolární buzení [9]	29
15	Fázorový diagram a časový průběh řízení wave drive [16]	31
16	Fázorový diagram a časový průběh řízení full step drive [16]	32
17	Fázorový diagram a časový průběh řízení half step drive [16]	33
18	Fázorový diagram a časový průběh mikrokrokování s čtvercovou trajektorií [16]	33
19	Časový průběh a časový průběh mikrokrokování s kružnicovou trajektorií [16]	34
20	Napěťové signály pro řízení krokového motoru [19]	35
21	Driver EM705 [18]	35
22	Zapojení fází motoru třídy 57SS [20]	36
23	Závislost momentu na rychlosti otáčení pro motor 57HS22 [20]	36
24	Raspberry Pi 4 Model B [27]	37
25	Arduino Micro [28]	39
26	Ilustrace přerušení	39
27	Časovač v režimu CTC [32]	41
28	UART komunikace [29]	42
29	Řez stolů RAH	43
30	Proudění vzduchu otvorem na hrací ploše	45
31	Ventilátor SUNON PMD2412PMB1-A(2).GN [21]	46
32	Kartézský kinematický typ	47
33	Kinematický typ CoreXY	49
34	Překřížení řemenů CoreXY [22]	49
35	Kinematický typ H-bot	50

36	Model RAH v prostředí Fusion 360	52
37	Základní funkční díly RAH	53
38	Detail dílů na straně robota s odkrytou hrací plochou	54
39	Dvoukanálový relé modul [36]	55
40	Paralelní zapojení na zdroj 24V DC	56
41	Schéma elektronického zapojení RAH	57
42	S-křivka pro spojitě řízení	60
43	Graf otáček při zastavení motoru	61
44	Zjednodušené schéma interního P regulátoru	63
45	Srovnání spojitěho řízení polohy P regulátorem s S-křivkami	63
46	Stručný vývojový diagram funkce loop()	67
47	Laserové moduly - receiver a transmitter [37]	69
48	Solenoid uvnitř stolu	70
49	Osvětlení stolu LED pásky	71
50	Robotický Air hockey	83
51	Arduino na pájecí desce a relé modul	83
52	Pohled na robotickou stranu bez brakové konstrukce s odkrytou horní deskou	84
53	Pohled na odkrytou stranu hráče s RPi a dotykovým displejem	84
54	Pohled na odkrytý rozvaděč se zdroji napětí 36 V, 24 V, 5 V	85
55	Pohled do vnitřních prostor stolu	85
56	Logo VUT FSI ÚAI	86
57	Uživatelské rozhraní na displeji pro nastavení parametrů RAH [40]	86

Seznam tabulek

1	Čtyřtaktní řízení s magnetizací jedné fáze [9]	31
2	Čtyřtaktní řízení s magnetizací dvou fází [9]	31
3	Osmiaktáktní řízení [9]	32
4	Konvence příkazů pro ovládání Arduina	66

Seznam algoritmů

1	ISR funkce n-tého časovače pro konkrétní motor	68
---	--	----

Použité zkratky

IDE Integrated Development Environment (Vývojové prostředí)

RAH Robotický Air Hockey

RPi Raspberry Pi

Seznam použité literatury

- [1] Vzdušný hokej. Wikipedia. [online]. [cit. 2020-03-20]. Dostupné z: https://cs.wikipedia.org/wiki/Vzdušný_hokej
- [2] Nuvation Engineering. Air Hockey Robot. [online]. [cit. 2020-03-22]. Dostupné z: <https://www.nuvation.com/air-hockey-robot>
- [3] Jonathan Chiu; Daniel Kuo. Air Hockey Robot Windows 10 IoT Core. [online]. [cit. 2020-03-22]. Dostupné z: <https://www.hackster.io/windowsiot/air-hockey-robot-7d7a24>
- [4] JJrobots. Air Hockey Robot (a 3D printer hack). [online]. [cit. 2020-03-22]. Dostupné z: <https://www.jjrobots.com/air-hockey-robot-a-3d-printer-hack/>
- [5] KOPECKÝ, David. Automatický vzduchový hokej. Praha, 2017. Bakalářská práce. Vedoucí práce Zemánek Jiří. České vysoké učení technické.
- [6] JJrobots. Air Hockey Robot EVO. [online]. [cit. 2020-03-22]. Dostupné z: <https://www.jjrobots.com/the-open-source-air-hockey-robot/>
- [7] Stepper Motor Basics. Industrial Circuits Application Note. [online]. [cit. 2020-03-22]. Dostupné z: <http://users.ece.utexas.edu/~valvano/Datasheets/StepperBasic.pdf>
- [8] Stepper motor basics, Application Note 001, FAULHABER [online]. [cit. 2020-03-22]. Dostupné z: https://www.faulhaber.com/fileadmin/user_upload_global/support/MC_Support/Motors/AppNotes/Faulhaber_AN001_EN.pdf
- [9] Krokové motory. Hobbyrobot, 2013. [online]. [cit. 2020-03-22]. Dostupné z: <http://profirobot.cz/wp-content/uploads/2018/03/Krokov%C3%A9-motory.pdf>
- [10] ACARNLEY, P. P. Stepping motors: a guide to theory and practice. 4th edition. United Kingdom: The Institution of Engineering and Technology, 2007. ISBN 978-0-85296-417- 0.
- [11] NOVÁK, P. Mobilní roboty - pohony, senzory, řízení. BEN, Praha, 2005. 256 s. ISBN 80-7300-141-1
- [12] 2 phase vs 5 phase stepper motors. Oriental motor. [online]. [cit. 2020-03-24]. Dostupné z: <https://www.orientalmotor.com/stepper-motors/technology/2-phase-vs-5-phase-stepper-motors.html>
- [13] FORMÁNEK, J. Tvorba grafické vizualizace principu krokového motoru a jeho řízení. Výukový text.

- [14] The difference between unipolar and bipolar stepper motors. Tech explorations. [online]. [cit. 2020-03-24]. Dostupné z: <https://techexplorations.com/blog/arduino/blog-the-difference-between-unipolar-and-bipolar-stepper-motors>
- [15] Stepper motor. Wikipedia. [online]. [cit. 2020-03-24]. Dostupné z: https://en.wikipedia.org/wiki/Stepper_motor
- [16] Zaber. Microstepping tutorial. [online]. [cit. 2020-03-24]. Dostupné z: <https://www.zaber.com/technical-articles/microstepping-tutorial>
- [17] W. SPONG, Mark. Impact controllability of an air hockey puck. Systems & Control Letters [online]. Elsevier B.V, 2001, 42(5), 333-345 [cit. 2020-03-25]. DOI: 10.1016/S0167-6911(00)00105-5. ISSN 0167-6911.
- [18] Leadshine. [online katalogový list]. EM705. [cit. 2020-03-25]. Dostupné z: http://www.cncshop.cz/EM/EM705d_P.pdf
- [19] Leadshine. [online katalogový list]. DM422C. [cit. 2020-03-25]. Dostupné z: <http://www.cncshop.cz/PDF/DM422C.pdf>
- [20] Leadshine. [online katalogový list]. 57HS Series Hybrid Stepping Motors. [cit. 2020-03-25]. Dostupné z: <http://www.cncshop.cz/PDF/57HS09-22.pdf>
- [21] GM Electronics. Ventilátor Sunon PMD2412PMB1-A. [online]. [cit. 2020-04-05]. Dostupné z: <https://www.gme.cz/ventilator-sunon-pmd2412pmb1-a>
- [22] CoreXY [online]. [cit. 2020-04-07]. Dostupné z: http://www.doublejumpelectric.com/projects/core_xy/2014-07-15-core_xy/
- [23] YIN, M., CHEN, Y., LEE, K.H., FU, D., TSE, Z., KWOK, K.W., Dynamic Modeling and Characterization of the Core-XY Cartesian Motion System. ISBN: 978-1-5386-6869-6
- [24] WEIKERT, S., RATNAWEERA, R., ZIRN, O., WEGENER, K., Modeling and measurement of H-bot kinematic systems. Inst. of Machine Tools and Manufacturing. Švýcarsko. 2011
- [25] FABERANT 3D-принтеры., CoreXY против H-Bot. Сравнение кинематики в физической симуляции. In: Youtube [online]. Zveřejněno 04. 09. 2019 [vid. 2020-04-07]. Dostupné z: <https://www.youtube.com/watch?v=-HLCfR7mnVA>
- [26] DENNIS, A., Raspberry Pi Home Automation With Arduino. Second ed. 2015. Birmingham, UK: Packt Publishing. ISBN 978-1-78439-920-7

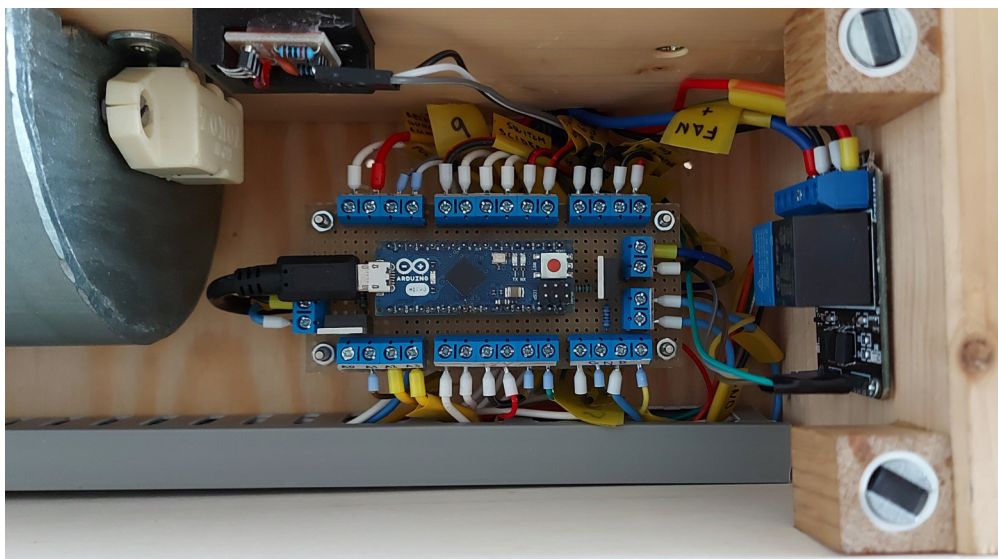
- [27] Raspberry Pi 4 Computer Model B. [online]. [cit. 2020-04-09]. Dostupné z: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf>
- [28] VODA, Z., Tým HW kitchen. Průvodce světem Arduina. Nakladatelství Martin Stříž, Bučovice. 240 s. 2015. ISBN: 978-80-87106-90-7
- [29] Universal asynchronous receiver-transmitter. Wikipedia. [online]. [cit. 2020-04-09]. Dostupné z: https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter
- [30] Basics of UART communication. Circuit Basics. [online]. [cit. 2020-04-09]. Dostupné z: <https://www.circuitbasics.com/basics-uart-communication/>
- [31] MICROCONTROLLERS INTERRUPTS and ACCURATE TIMING I. Portland State University. [online]. [cit. 2020-04-09]. Dostupné z: https://www.pdx.edu/nanogroup/sites/www.pdx.edu/nanogroup/files/2017%20Lab%20Interrupt%20and%20accurate%20timing_1.pdf
- [32] ATMEL. [online katalogový list]. ATmega16U4/ATmega32U4. [cit. 2020-04-10]. Dostupné z: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf
- [33] Timers. AVR Beginners. [online]. [cit. 2020-04-10]. Dostupné z: <http://www.avrbeginners.net/architecture/timers/timers.html>
- [34] AccelStepper library for Arduino. [online]. [cit. 2020-05-20]. Dostupné z: <https://www.airspayce.com/mikem/arduino/AccelStepper/>
- [35] Nguyen, Kim-Doang & Ng, Teck & Chen, I-Ming. (2008). On Algorithms for Planning S-curve Motion Profiles. International Journal of Advanced Robotic Systems. 5. 10.5772/5652.
- [36] GM Electronics. Relé modul 2x 10A/250V. [online]. [cit. 2020-05-25]. Dostupné z: <https://www.gme.cz/rele-modul-2x-10a-250v>
- [37] KY-008 Laser Diode Transmitter & Receiver Module. [online katalogový list]. [cit. 2020-05-28]. Dostupné z: <https://www.lelong.com.my/ky-008-laser-diode-transmitter-receiver-module-sainapse-io-F822529-2007-01-Sale-I.htm>
- [38] HAREENDRAN, T. K., The Mysterious Laser Receiver Sensor Module. [online]. [cit. 2020-05-29]. Dostupné z: <https://www.codrey.com/electronic-circuits/the-mysterious-laser-receiver-sensor-module/>
- [39] Vývojová deska Arduino Joy-it TEENSY 3.6 TEENSY 3.6, ATmega32. Velkoobchod Conrad. [online]. [cit. 2020-06-12]. Dostupné z: <https://velkoobchod.conrad.cz/vyvojova-deska-arduino-joy-it-teensy-3-6-teensy-3-6-atmega32.k1656380>

- [40] SLÁMA, Ondřej. Vzdušný hokej - strojové vidění a herní strategie. Brno, 2020. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/125433>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky. Vedoucí práce Radomil Matoušek.
- [41] JAŠEK, Dominik. Online model robotického Air hockey. [online]. [cit. 2020-06-12]. Dostupné z: <https://a360.co/2I2ok0s>
- [42] JAŠEK, Dominik. Arduino kód RAH. In: GitHub [online]. [cit. 2020-06-24]. Dostupné z: <https://github.com/holyduke/AirHockey>

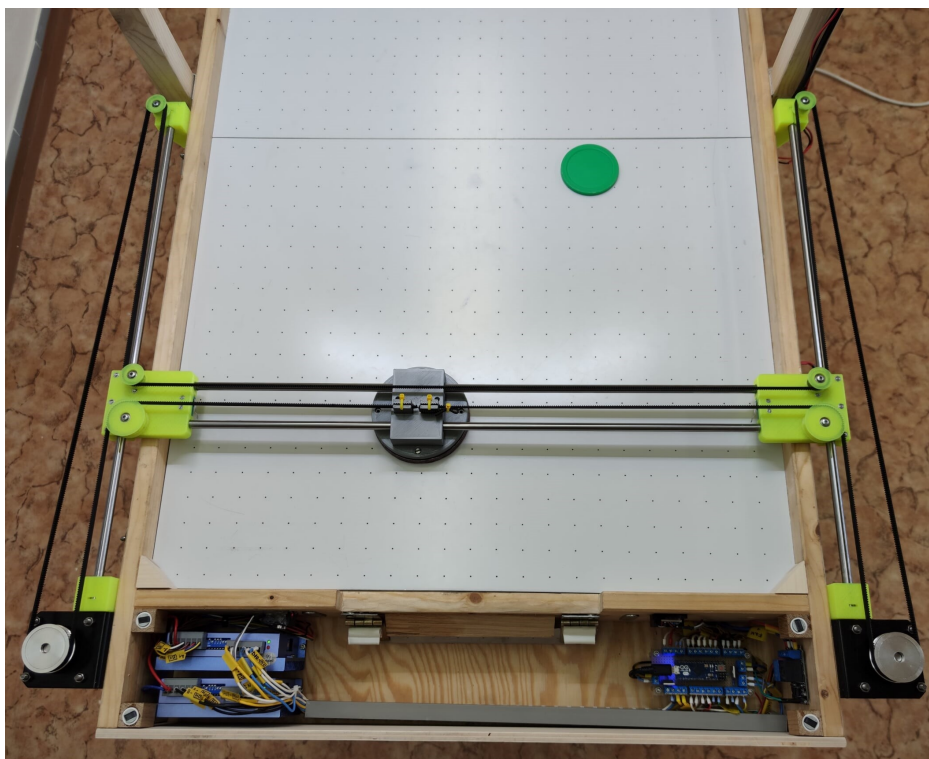
Příloha



Obrázek 50: Robotický Air hockey



Obrázek 51: Arduino na pájecí desce a relé modul



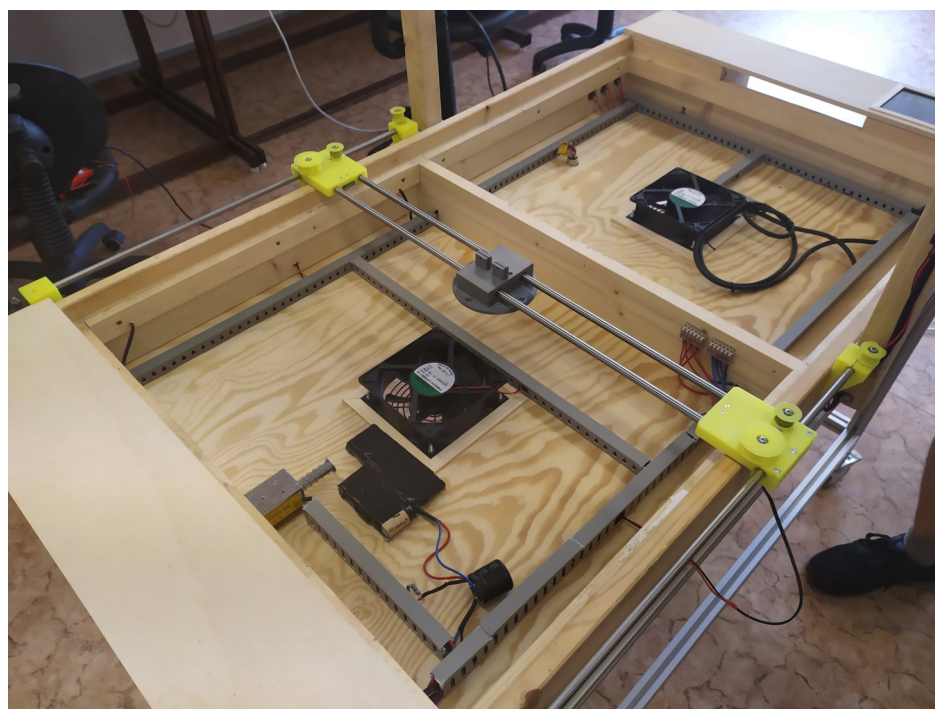
Obrázek 52: Pohled na robotickou stranu bez brakové konstrukce s odkrytou horní deskou



Obrázek 53: Pohled na odkrytou stranu hráče s RPi a dotykovým displejem



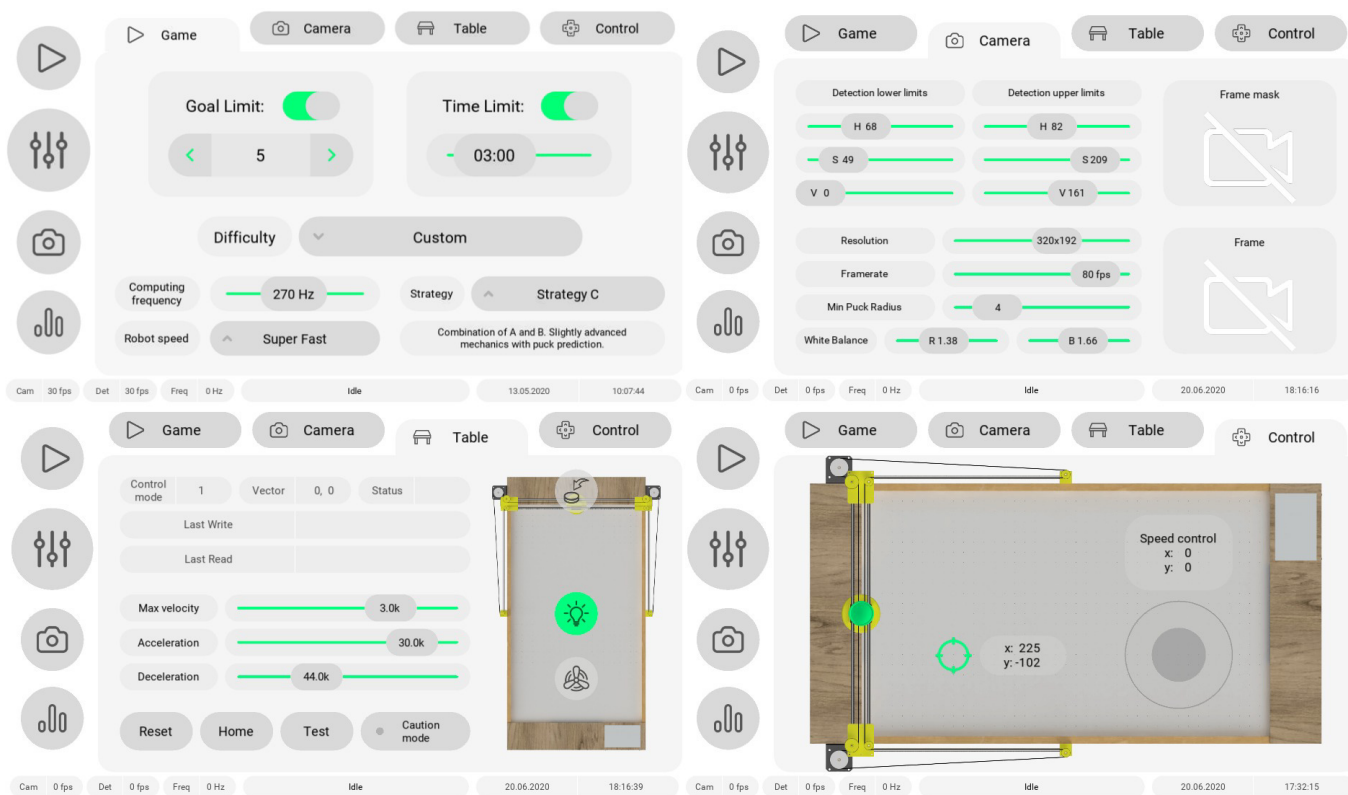
Obrázek 54: Pohled na odkrytý rozvaděč se zdroji napětí 36 V, 24 V, 5 V



Obrázek 55: Pohled do vnitřních prostor stolu



Obrázek 56: Logo VUT FSI ÚAI



Obrázek 57: Uživatelské rozhraní na displeji pro nastavení parametrů RAH [40]