

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra řídicí techniky

Využití robota LEGO MINDSTORMS ve výuce - návrh soutěžních úloh

Michal Urválek

Vedoucí: Ing. Martin Hlinovský Ph.D.
Obor: Systémy řízení
Studijní program: Kybernetika a robotika
Květen 2017

České vysoké učení technické v Praze
Fakulta elektrotechnická
katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Urválek Michal**

Studijní program: Kybernetika a robotika
Obor: Systémy a řízení

Název tématu: **Využití robota LEGO MINDSTORMS ve výuce - návrh soutěžních úloh**

Pokyny pro vypracování:

1. Navrhněte a realizujte dvě nové soutěžní úlohy pro LEGO Mindstorms NXT nebo EV3 s návrhem řízení ve Vámi vybraném programovacím prostředí.
2. Jedna soutěžní úloha bude robot sledující černou čáru a druhá soutěžní úloha s názvem 'Střelnice' bude robot, který bude rozpoznávat barevné míčky a následně je bude umísťovat do správných přihrádek.
3. Zpracujte dokumentaci konstrukce robotů (vytvořené v LEGO designéru) a způsobu jejich programování.
4. Vytvořte webové stránky k navrženým soutěžním úlohám a vypracujte podrobné návody ve formě pravidel k oběma soutěžním úlohám.

Seznam odborné literatury:

- [1] Tomáš Bělík: Využití robota Lego Mindstorms - Návrh a realizace speciálních úloh, Bakalářská práce, ČVUT FEL v Praze, 2010
[2] Dan Martinec: Využití robota Lego Mindstorms při výuce předmětu A3B99RO Roboti, Bakalářská práce, ČVUT FEL v Praze, 2010

Vedoucí: Ing. Martin Hlinovský, Ph.D.

Platnost zadání: do konce letního semestru 2017/2018

L.S.

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
Děkan

V Praze dne 6. 2. 2017

Poděkování

Tímto bych chtěl poděkovat vedoucímu bakalářské práce Ing. Martinu Hlinovskému Ph.D. za připomínky a cenné rady při řešení obou úloh a za zapůjčení LEGO setů. Dále bych chtěl poděkovat rodičům a přátelům za rady, které mi pomohly v řešení úloh.

Prohlášení

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

V Praze, 25. května 2017

.....

Abstrakt

Tato bakalářská práce se zabývá návrhem dvou soutěžních úloh s využitím stavebnice LEGO Mindstorms. Tyto úlohy mohou být použity jako zadání Robosoutěže a v rámci předmětu B3B35RO Roboti, který spadá do studijního programu Kybernetika a robotika. Zabývá se vybavením stavebnice LEGO Mindstorms NXT. Dále popisuje zadání a možnost řešení první úlohy sledování černé čáry a druhé úlohy střelnice. Závěrečná část se věnuje přípravě podkladů k webovým stránkám, které budou přidány k existujícím stránkám předmětu.

Klíčová slova: robot, LEGO, PID regulátor, NXC, střelnice, sledování černé čáry

Abstract

This bachelor thesis deals with the design of two competitive tasks using the LEGO Mindstorms kit. These tasks can be used as an assignment of the Robos Competition and within the A3B99RO Robots subject, which is part of the Cybernetics and Robotics study program. It deals with the LEGO Mindstorms NXT kit contents. It also describes the assignment and the possibility of solving the first task of tracking the black line and the second task of the shooting range. The final part is devoted to the preparation of documents for web pages that will be added to the existing pages of the subject.

Keywords: robot, LEGO, PID controller, shooting range, black line tracking

Title translation: Use of the LEGO MINDSTORMS robot in teaching - design of competitive tasks

Obsah

1 Úvod	1	3 Sledování černé čáry	13
2 LEGO Mindstorms	3	3.1 Popis zadání	13
2.1 Hardwarové vybavení stavebnice .	4	3.2 Speciální prvky černé čáry	14
2.1.1 Programovatelná NXT kostku	5	3.3 Konstrukce robota	15
2.1.2 Servomotor	5	3.4 Návrh regulátoru	16
2.1.3 Dotykový senzor	6	3.4.1 Obecný popis regulátoru	17
2.1.4 Ultrazvukový senzor	6	3.4.2 Ziegler-Nicholsova metoda	18
2.1.5 Světelný senzor	7	3.4.3 Vliv konstant k_p , k_i a k_d na chování robota	19
2.1.6 Senzor zvuku	7	3.5 Popis programu	20
2.1.7 Barevný senzor	8	4 Střelnice	21
2.1.8 Gyroskopický senzor	8	4.1 Popis zadání	22
2.2 Software, programovací možnosti	9	4.2 Měření barvy míčků	23
2.2.1 NXT-G	9	4.3 Konstrukce robota	25
2.2.2 NXC	9	4.3.1 Sběr míčků	26
2.2.3 leJOS	10	4.3.2 Převoz a střelba	27
2.3 LEGO Digital Designer	10	4.4 Popis programu	28

5	Tvorba webových stránek	31
6	Závěr	33
A	Literatura	35
B	Obsah CD	37
B.1	Sledování černé čáry	37
B.2	Střelnice	37
B.3	Webové stránky	37
B.4	BP-Urvalek	38
C	Zadání práce	39

Obrázky

2.1 LEGO Mindstorms RXC s motory, světelným a dotykovými senzory ...	3	3.5 Pseudokód pro sledování černé čáry	20
2.2 LEGO Mindstorms NXT s příslušenstvím	4	4.1 Uspořádání hracího hřiště.	21
2.3 Řídící kostka	5	4.2 Rozměry terče.	22
2.4 Servomotor	5	4.3 Způsoby měření	24
2.5 Dotykový senzor	6	4.4 Graf měření - senzor míří na střed míčku	24
2.6 Ultrazvukový senzor	6	4.5 Graf měření - senzor míří na spodek míčku	25
2.7 Světelný senzor	7	4.6 Celková konstrukce robota	26
2.8 Senzor zvuku	7	4.7 Možné sběrače míčků	27
2.9 Barevný senzor	8	4.8 Sběrač míčků typu "hrábě"	27
2.10 Gyroskopický senzor	8	4.9 Finální verze katapultu	28
2.11 Okno LEGO Digital Designeru	10	4.10 Podavač míčků	29
3.1 Soutěžní čára	14	4.11 Vývojový diagram střelnice ...	30
3.2 Speciální prvky černé čáry	14		
3.3 Pohled zespoda na podvozek robota.....	16		
3.4 Celková konstrukce robota	16		

Tabulky

2.1 Porovnání softwarů	9
3.1 Výpočet konstant pro jednotlivé typy regulátorů	19
3.2 Změny chování robota pro jednotlivé změny parametrů regulátoru	19
4.1 Měření barev světelným senzorem	24



Kapitola 1

Úvod

Tato práce se zabývá možnostmi využití stavebnice LEGO Mindstorms. Jedná se o návrh dvou soutěžních úloh, které mohou být využity v rámci předmětu B3B35RO Roboti studijního programu Kybernetika a robotika a nebo jako zadání Robosoutěže, kterou každoročně pořádá katedra řídicí techniky. Popisuje hardwarové a softwarové možnosti stavebnice.

První soutěžní úloha je sledování černé čáry, která obsahuje speciální prvky. Sledování černé čáry je jedna ze základních regulačních úloh, které lze se stavebnicí LEGO Mindstorms realizovat. Cílem je navrhnout a naprogramovat robota, který pojede kolem černé čáry.

Druhá soutěžní úloha se jmenuje střelnice. Cílem této úlohy je posbírat míčky umístěné na herní ploše, které jsou umístěné v předem daném prostoru, a umístit je do terče podle barev. Terč je tvořen laminátovými soustřednými kružnicemi, které dosahují výšky až 20 cm na herní plochu.

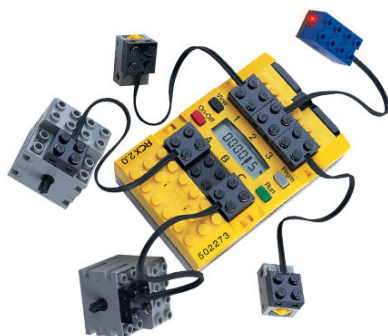
Poslední část se zabývá tvorbou webových stránek, které budou přidány k existujícím webovým stránkám předmětu a Robosoutěže.

Účelem obou návrhů soutěžních úloh je propagace elektrotechnické fakulty Českého vysokého učení technického v Praze.

Kapitola 2

LEGO Mindstorms

Jedná se o verzi LEGO stavebnice, která kromě standardních LEGO kostek obsahuje programovatelnou kostku, motory a senzory. Dnes se můžeme setkat se dvěma verzemi této stavebnice LEGO Mindstorms NXT, která se na trhu poprvé objevila v roce 2006, a LEGO Mindstorms EV3, která se začala prodávat v roce 2013. Historie LEGO robotů je ale podstatně starší. První počítačem řízení LEGO produkt vznikl už v roce 1986. O dva roky později ve spolupráci s Massachusetts Institute of Technology vznikla první inteligentní kostka programovatelná přes počítač. První inteligentní kostka ze série LEGO Mindstorms byla zveřejněna v roce 1998. Tato verze nesla označení LEGO Mindstorms RCX viz. Obrázek 2.1 a prodávala se pod názvem Robotics Invention System.



Obrázek 2.1: LEGO Mindstorms RCX s motory, světelným a dotykovými senzory

2.1 Hardwarové vybavení stavebnice

K realizaci řešení úloh jsem použil stavebnici LEGO Mindstorms NXT. Velmi podrobnému popisu této stavebnice se již věnovalo několik bakalářských prací (např. [1], [2]), ale pro úplnost zde uvedu alespoň stručný popis.



Obrázek 2.2: LEGO Mindstorms NXT s příslušenstvím

Základní stavebnice LEGO Mindstorms obsahuje:

- Programovatelnou NXT kostku
- 3 servomotory s rotačními senzory
- 2 dotykové senzory
- Ultrazvukový senzor
- Světelný senzor
- Senzor zvuku
- Sedm spojovacích kabelů typu RJ-12

K řešení navrhovaných úloh lze také využít dalších senzorů:

- Barevný senzor
- Gyroskopický senzor

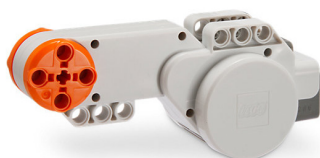
■ 2.1.1 Programovatelná NXT kostka



Obrázek 2.3: Řídící kostka

Programovatelná NXT kostka je základem celé stavebnice. Obsahuje 32-bitový ARM7 mikroprocesor, který běží na frekvenci 48 MHz s 256 KB Flash RAM a 64 KB RAM. Program do kostky nahrajeme pomocí USB kabelu nebo technologie Bluetooth. Bluetooth je možné také použít pro komunikaci mezi kostkami. Kostka obsahuje USB konektor typu B. Dále zde nalezneme 7 konektorů RJ-12. Jedná se o vstupní a výstupní porty. Porty označené písmeny A, B a C slouží pro připojení motorů. Motory jsou řízeny PWM signálem. Porty označené čísly 1 až 4 jsou vstupní porty a slouží pro připojení senzorů. Kostka také obsahuje 4 tlačítka, které slouží pro vypnutí a zapnutí kostky a pohyb v menu. Ale lze je naprogramovat a používat během běhu programu k interakci s uživatelem. Jako zobrazovací jednotka je použit maticový LCD display, který má 100 x 64 pixelů. Poslední věc, co kostka obsahuje je, reproduktor.

■ 2.1.2 Servomotor



Obrázek 2.4: Servomotor

Servomotor je jedinou periferií, která je pohonnou jednotkou robota, ale lze jej použít i jako senzor otáček. Servomotor obsahuje elektromotor, který je napájen 9 V. Výstup elektromotoru je na výstup servomotoru převodován

v poměru 48:1. Pro přesné otáčení slouží impulsní snímač otáček, který disponuje přesností $\pm 1^\circ$. Pro ovládání rychlosti motoru se používá pulzní šířková modulace (PWM).

■ 2.1.3 Dotykový senzor



Obrázek 2.5: Dotykový senzor

Dotykový senzor je v podstatě obyčejné tlačítko. Tento senzor umí rozeznat pouze dva stavy sepnuto a rozepnuto. Nejčastěji se používá pro detekci překážek nebo jako doraz, ale lze ho použít i jako tlačítko např. na spouštění programu.

■ 2.1.4 Ultrazvukový senzor



Obrázek 2.6: Ultrazvukový senzor

Ultrazvukový senzor vypadá jako oči robota. Ve skutečnosti jde o vysílač a přijímač, který funguje na frekvenci 40 kHz. Funguje na principu echolokace. Vysílač vyšle zvukový signál, který se odrazí od překážky a vrátí se zpět. Odečte se čas od vyslání do přijetí signálu, který se přepočítá na vzdálenost. Senzor je schopen měřit vzdálenost teoreticky od 0 do 255 cm s přesností ± 3 cm. Ve skutečnosti má problém změřit vzdálenost menší než 5 cm. Pro úplně malé vzdálenosti už není přijímač schopný zachytit vysílanou ultrazvukovou vlnu. Největší přesnosti dosahuje pokud měří vzdálenost od velkých objektů.

Malé objekty je velice obtížné tímto senzorem zachytit. Ultrazvukový senzor se používá na detekci překážek a orientaci v prostoru.

■ 2.1.5 Světelný senzor



Obrázek 2.7: Světelný senzor

Světelný senzor se používá na měření intenzity světla nebo měří barvy v odstínech šedi. V prvním případě využívá pouze fototranzistor a měří intenzitu okolního světla. Měření odstínu šedi funguje tak, že červená LED dioda, která je součástí senzoru, osvětluje měřený povrch, který je současně snímán fototranzistorem. Principem tohoto senzoru je, že každá barva pohlcuje jiné množství světla a zbytek se odrazí. Tranzistor pak následně zachytí množství odraženého světla. Výstupem senzoru mohou být hodnoty 0 až 100% nebo pokud chceme přesnější hodnoty, může nám senzor vracet hodnoty 0 až 1023. Nejčastěji se používá ke sledování černé čáry.

■ 2.1.6 Senzor zvuku



Obrázek 2.8: Senzor zvuku

Senzor zvuku měří akustický tlak. Detekuje zvuk v rozsahu 50 až 90 dB, ale hodnoty přepočítává a výsledek uvádí v procentech. Měří frekvence v

rozmezí 3 až 6 kHz, kde je lidské ucho nejcitlivější. Lze ho například použít pro spuštění robota tlesknutím.

■ 2.1.7 Barevný senzor



Obrázek 2.9: Barevný senzor

Tento senzor umí rozlišit základních šest barev: černá, modrá, zelená, žlutá, červená, bílá, které jsou reprezentovány čísly 1 až 6. Princip je stejný jako u světelného senzoru. Rozdíl je v tom, že k osvětlení se používá RGB dioda, která měřený povrch osvětluje současně červenou, zelenou a modrou barvou. Výhodou tohoto senzoru je, že přesně určí barvu a není potřeba to nijak softwarově převádět výstupní hodnoty senzoru. Nevýhodou je, že při špatném úhlu nebo velké vzdálenosti senzoru vůči povrchu bývá barva určena chybně.

■ 2.1.8 Gyroskopický senzor



Obrázek 2.10: Gyroskopický senzor

Jedná se o senzor, který snímá rychlost a směr otáčení v jedné ose. Senzor je schopen změřit maximální rychlost až 360° za sekundu. Senzor je od výrobce HiTechnic. Lze ho použít např. na segway nebo na jiné úlohy, kde je potřeba balancovat.

Vlastnosti	NXT-G	NXC	leIOS
Typ	Grafický	Not exactly C	Java
Firmware	Standartní	Standartní	Vlastní
Vícevláknové aplikace	Ano	Ano	Ano
Obsluha událostí	Ne	Ne	Ano
Podpora datového typu float	Ne	Ne	Ano
Podpora Bluetooth	Ano	Ano	Ano
Licence	Placená ve standardním setu	Freeware	Open-source

Tabulka 2.1: Porovnání softwarů

2.2 Software, programovací možnosti

Vzhledem k oblíbenosti LEGO Mindstorms vznikla spousta možností, jak programovat robota. Jedná se buď o grafické jazyky nebo o jazyky, které přejímají principy a syntax od standardních programovacích jazyků jako jsou Assembler, C a Java. Většina jazyků je popsána již v předchozích pracích. Proto tu zmíním ty, se kterými se lze často setkat, protože vychází z klasických programovacích jazyků: jedná se o grafický jazyk NXT-G, NXC založený na jazyku C a leIOS, základem je jazyk Java. Jejich porovnání je v tabulce 2.1

2.2.1 NXT-G

NXT-G je grafický jazyk, který je dostupný na LEGO stránkách. Jedná se o jazyk, kde dochází ke spojování bloků. Tento jazyk je vhodný pro ty, co začínají s programováním.

2.2.2 NXC

NXC neboli Not Exactly C je zjednodušená verze jazyka C, která slouží k programování robota. Hlavní výhodou je, že lze použít standardní firmware. Jedná se o freeware software volně dostupný na stránkách <http://bricxcc.sourceforge.net/>.

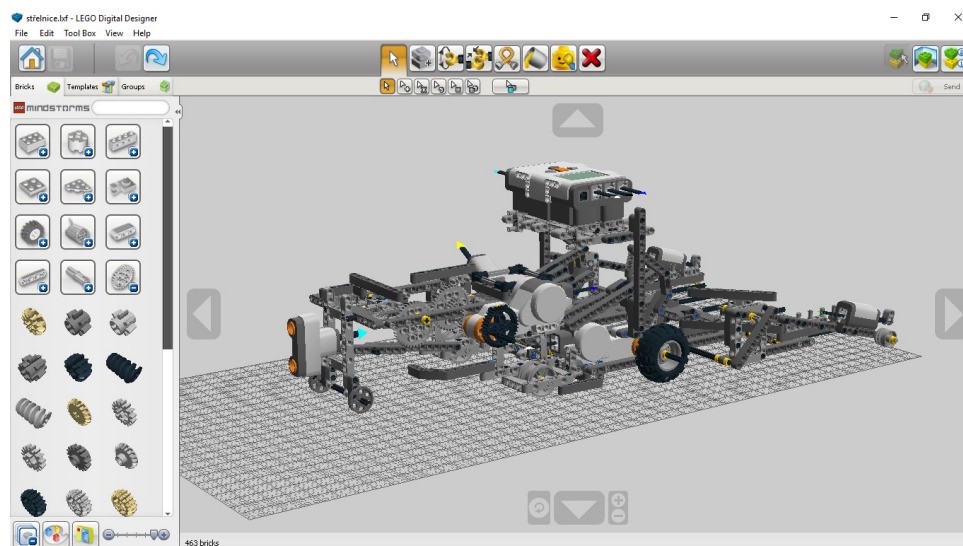
2.2.3 leJOS

LeJOS neboli LEGO Java OS je programovací jazyk, který umožňuje programovat robota v jazyce Java. Jedná se o open-source programovací jazyk. Hlavní výhody jsou například podpora objektového programování, podpora datového typu float a možnost obsluhy událostí pomocí JAVA events.

2.3 LEGO Digital Designer

Digital LEGO Designer je modelovací 3D program, který se odlišuje tím, že v něm lze používat pouze LEGO kostky. Obsahuje snad všechny dostupné typy kostek. Program je přímo od tvůrců LEGO a je zdarma ke stažení na stránkách LEGO[11]. Jeho hlavní výhodou je, že k postavenému modelu umí vygenerovat postup, podle kterého lze vytvořený model sestavit. Nejedná se úplně o dokonalý postup, ale lze ho podle návodu sestavit. Výhodou oproti fotkám je, že nezapomenete na dílky, který jsou umístěné uprostřed robota, nebo na spojovací prvky. Práce s programem je intuitivní, ale není moc uživatelsky příznivá. Návod je k dispozici ke stažení na stránkách LEGO Digital Designeru[14].

Samotné okno designeru lze rozdělit na tři části viz. Obrázek 2.11. Nástrojovou lištu, modelovací prostor a záložku na výběr kostek.



Obrázek 2.11: Okno LEGO Digital Designeru

Kostky lze vybírat hledáním v seznamu nebo pokud známe číslo dílu nebo název, je možné ho napsat do vyhledávací lišty. Výběr pomocí čísla nefunguje

vždy. Některé kostky mají jiné označení ve starších stavebnicích oproti novějším a některé to nenajde vůbec. Hledání v seznamu usnadňuje rozdělení kostek do několika kategorií. Obsah seznamu lze měnit pomocí View > New themes, kde si lze vybrat ze třech možností: LDD, LEGO MINDSTORMS a LDD Extended. První možnost zahrnuje většinu kostek včetně kostek stavebnice LEGO MINDSTORMS. LEGO MINDSTORMS obsahuje dílky z obou verzí stavebnice, jak EV3, tak NXT, a to včetně kabeláže. U kabelů se neřeší kudy daný kabel vede, ale z jakého senzoru vede do kterého výstupu kostky. Jediný dílek, který jsem nenašel v předchozích dvou sekcích, přestože je součástí stavebnice LEGO MINDSTORM NXT, je kolo o průměru 61 mm včetně příslušné pneumatiky. Tyto specifické dílky lze najít přepnutím do LDD Extended. Zde jsou k nalezení všechny díly, které lze v tomto programu použít. Nevýhodou volby LDD Extended je, že všechny kostky, které tam lze dohledat, mají nastavenou červenou barvu.

Nástrojová lišta je umístěná v horní části obrazovky. Nástroje popisují zleva doprava. První nástroj je nástroj na výběr kostek. Pomocí tohoto nástroje lze vybírat, přesouvat a sestavovat kostky nebo sestavy kostek. Další nástroj slouží ke klonování kostek. Třetí nástroj slouží na otáčení kostek. Lze otáčet pouze jednou kostkou a je nutné mít jeden pevný bod jako osu otáčení. Nástroj Hinge Align tool slouží na spojení dvou spojovacích bodů, následuje nástroj Flex tool, u obou z nich jsem nenašel využití a nepřišel jsem na to, jak je použít. Další nástroj slouží na změnu barvy kostek. Barvy si lze vybírat z předem dané palety barev. Poslední dva nástroje jsou na dočasné skrývání kostek a mazání kostek. Na pravé straně v liště nástrojů je možnost přepínat mezi třemi módy: modelovací/stavící mód, mód na prohlížení, kde si lze vybrat pozadí asi z pěti možností a poslední mód slouží na vygenerování manuálu. Manuál lze taky vygenerovat jako HTML soubor.

Poslední částí obrazovky je modelovací prostor. Při stavění lze vyjít z nějakého příkladu nebo začít úplně od nuly. Samotný model lze otáčet a přibližovat. Stavba modelu tu funguje principem "tap and build". Ze seznamu vyberete kostku, kterou pomocí šipek otočíte a následně spojíte z ostatními kostkami. Místo spojení kostek je označeno zeleným čtvercem.

Jako užitečný považuji nástroj, který najdeme v hlavní nabídce vlevo nahoře v záložce Tool Box > Take a screenshot. Tento nástroj vytvoří png obrázek modelu s průhledným pozadím. Stále se jedná o verzi printscreen a velikost a kvalita obrázku záleží na velikosti monitoru.

Obecně platí že, to co je postaveno pomocí LEGO designeru, jde vždy postavit i ve skutečnosti. Naopak to neplatí. Vlivem hmotnosti celého modelu a pružnosti jednotlivých kostek lze postavit věci, které nelze následně modelovat.

Kapitola 3

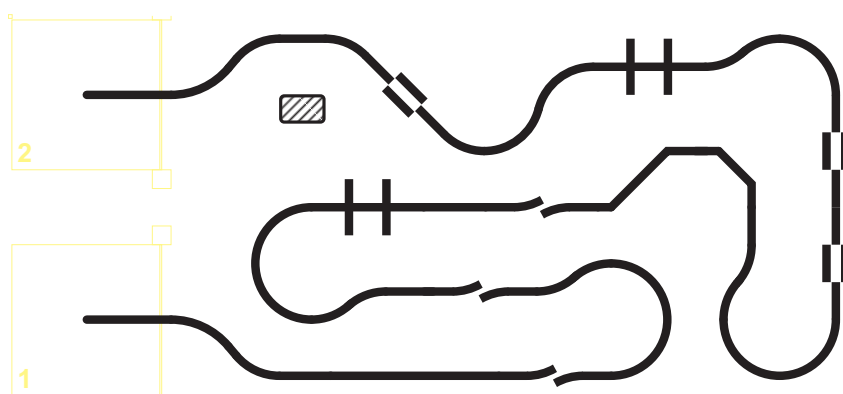
Sledování černé čáry

3.1 Popis zadání

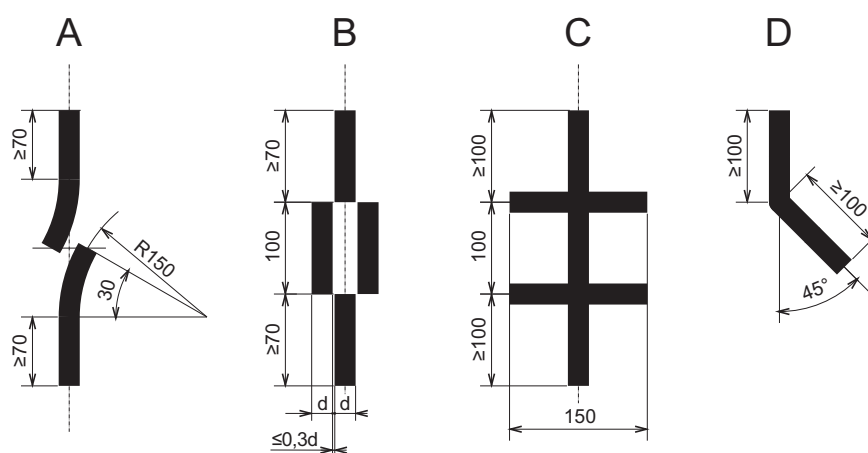
Cílem této úlohy, je sestavit robota, který bude sledovat černou čáru a projede danou čáru v nejkratším možném čase. Robot startuje v jednom ze žlutých čtverců označenými čísly 1 a 2. Druhý z nich je cílem. Do tohoto čtverce umístí soutěžící robota. Robot musí být schopný projet čárou v obou směrech viz Obrázek 3.1. Půdorys robota musí být menší než 36 x 36 cm, aby se vešel do startovního čtverce. Dále není povoleno používat kluzný podvozek. Za podvozek či jeho část je považována jakákoliv součást robota, která je při pohybu robota po rovině ve styku s pojezdovou plochou (i dočasně) a přenáší na pojezdovou plochu část váhy robota. Za kluzný podvozek je považován takový podvozek, jehož libovolná část se po pojezdové ploše neodvaluje, ale klouže. Během celé jízdy se robot nesmí od sledované čáry odchýlit o více jak 20 cm. Zkracování cesty je též zakázáno. Maximální doba na projetí čáry jsou 2 minuty. K řešení této úlohy je doporučeno zde použít PID regulátor, kterým lze při vhodně zvolených konstantách dosáhnout nejlepších výsledků.

Během soutěže bude mít každý soutěžící k dispozici 3 pokusy. Vítězem soutěže se stane ten, kdo projede čáru v nejkratším čase. V případě rovnosti časů se pojede 4. rozhodující jízda nebo porovná druhý nejrychlejší čas obou soutěžících. Případně se rozhodne hodem kostkou.

V tomto případě se jedná o černou čáru, která sice neobsahuje žádné výškové rozdíly, ale místo toho obsahuje 4 nové prvky viz Obrázek 3.2. Délka černé čáry je 7,5 m.



Obrázek 3.1: Soutěžní čára



Obrázek 3.2: Speciální prvky černé čáry

3.2 Speciální prvky černé čáry

Prvek, který je na obrázku označený písmenem A, je zvláštní přerušení čáry. Tento prvek také jednoznačně určuje, po které straně černé čáry robot musí jet. Dále testuje, zda byl správně nastaven regulátor, jestli po rychlém návratu k čáře nedojde k jejímu přejetí na druhou stranu. S tím také souvisí, že nelze použít integrační složku PID regulátoru, protože rozdíl od referenční hodnoty je příliš velký a doba než se robot vrátí k čáře je i několik vteřin. To způsobí, že i pro malé hodnoty integrační konstanty regulátoru dojde k načítání velké odchylky. Důsledkem toho potlačí integrační složka všechny ostatní složky regulátoru po návratu robota na černou čáru.

Prvek B je reverzní prvek, prohazuje černou za bílou a naopak. Jedná se o nejnáročnější prvek na průjezd i vzhledem k tomu, že na řešení této úlohy je povoleno používat pouze jeden světelný senzor. Více světelných senzorů není v jedné stavebnici k dispozici. Dalším úskalím tohoto prvku jsou přechody mezi normální a reverzní částí. Při nájezdu na tento prvek vzniká problém

způsobený tím, že se černá čára úplně nedotýká. Při hledání řešení jsem našel dvě možnosti, jak lze daný prvek projet viz videa v příloze. V obou případech záleží na úhlu nájezdu robota na přechod černá bílá u čáry. Ani v jednom případě se nejedná o způsob, který bude fungovat na 100%. První varianta průjezdu je o něco rychlejší, ale je méně spolehlivá. Druhá varianta, která je pomalejší a spolehlivější, se může časem lišit i o více než 5 vteřin mezi jednotlivými průjezdy celé tratě.

Prvek C je dvojité překřížení, simuluje dvě křížení černé čáry. Na jeho bezproblémové překonání stačí mít dostatečnou rychlost a dobře nastavený regulátor, aby nedošlo k překmitu přes černou čáru. Ale pokud robot zvládne prvek A bez problému, průjezd tímto prvkem je v podstatě formalitou.

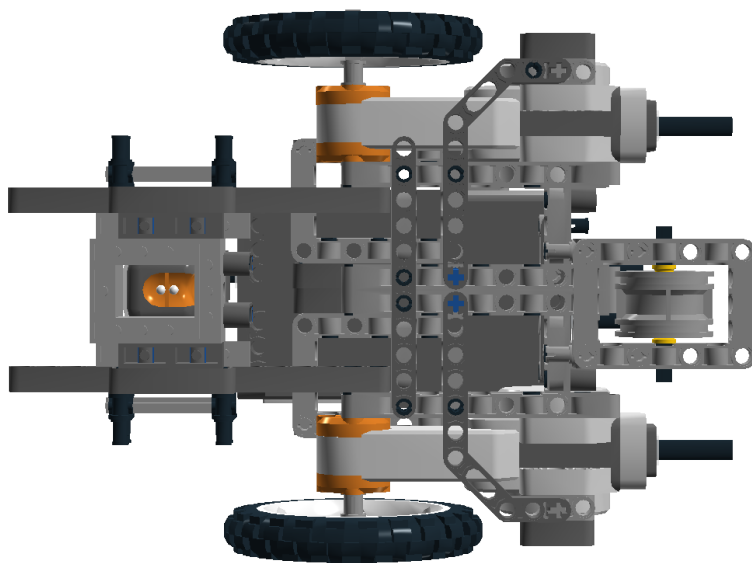
Posledním prvkem je prvek D, což je lomená zatáčka po úhlem 45° . Pokud robot jede z vnější strany, tak se jeho chování výrazně neliší od průjezdu klasickou zatáčkou. Ale v případě, že jede po vnitřní straně, testuje tento prvek vhodnost zvolené rychlosti a velikosti proporcionální složky regulátoru. Pokud je příliš vysoká rychlost nebo je nedostatečná velikost proporcionální složky, robot přejede černou čáru.

3.3 Konstrukce robota

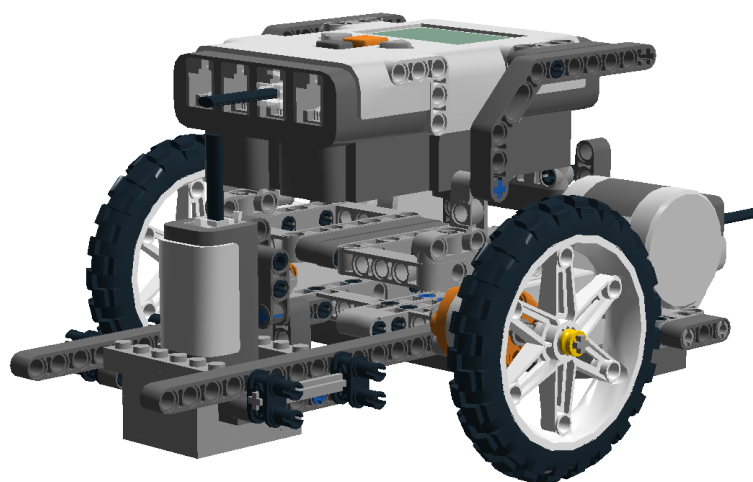
V této úloze je konstrukce robota jednoduchá. Z elektrického vybavení jsem použil dva motory, optický senzor a řídicí jednotku. Vzhledem ke zkušenostem získaným v předmětu Roboti jsem se rozhodl použít největší kola. Jejich hlavní výhodou je, že už okolo 50% rychlosti motoru dosahují dostatečné rychlosti a usnadní to regulaci.

Pro umístění senzoru je nutné brát v potaz několik kritérií. Senzor musí být umístěn před osou otáčení pohonných kol, ale neměl by být ve větší vzdálenosti než je rozteč kol, nebo přímo v ose otáčení. Pokud to zadání umožňuje, tj. není potřeba řešit nájezdy, je vhodné senzor umístit před osu otáčení. V opačném případě je vhodnější umístit senzor na osu otáčení, jak z hlediska programátorského, tak i konstrukčního, aby se senzor nezasekl o nájezdové rampy. Jak jsem již zmínil v úvodu kapitoly, v tomto případě se jedná o čáru, která neobsahuje žádné nájezdy. Proto jsem zvolil umístění senzoru před osou otáčení. Dále jsem zvažoval negativní okolní světelné vlivy na výstupní hodnoty světelného senzoru. Vytvořil jsem kolem senzoru krytí viz obrázek 3.4, které spolehlivě omezilo vliv vnějšího světla. Mezera mezi spodní částí krytí senzoru a podložkou byla menší než 1 cm. Senzor včetně krytí byl umístěn na dvou "lištách", které umožňovaly přesouvat senzor během ladění PID regulátoru. Chování robota sledujícího černou čáru bylo otestované i v místnosti, kde byla tma a bylo stejné jako za normálního osvětlení. Jako zadní kolečko byl v původním řešení použit míček. Ten byl nahrazen kolečkem z důvodu lepšího chování a robustnosti viz obrázek 3.3. Uvádím to z důvodu,

že ve videích v příloze jsou vidět obě varianty.



Obrázek 3.3: Pohled zespoda na podvozek robota



Obrázek 3.4: Celková konstrukce robota

3.4 Návrh regulátoru

Nastavení regulátoru je jádrem celé této úlohy. Je důležité si uvědomit, že robot nejede po černé čáře, ale po rozhraní mezi černou a bílou. Obecně si

lze vybrat, po jaké straně čáry robot pojede. V tomto případě lze jet pouze po levé straně černý čáry (z pohledu startu).

3.4.1 Obecný popis regulátoru

Nejjednodušším regulátorem je proporcionální regulátor (P regulátor). V našem diskretním případě lze fungování regulátoru popsat následujícím vztahem:

$$u = k_p \cdot (y - y_{ref}) \quad (3.1)$$

kde u je akční zásah nebo-li vstup do soustavy, k_p je proporcionální konstanta, y je výstupní hodnota senzoru a y_{ref} je požadovaná hodnota. Z toho vyplývá, že se počítá pouze rozdíl mezi naměřenou a požadovanou hodnotou, který se následně vynásobí konstantou. Zadanou čáru lze projet za použití proporcionálního regulátoru, ale při průjezdu zatáčkami bude docházet ke kmitání robota. Kmitání lze potlačit přidáním derivační složky. Derivační složka slouží k předvídání nejbližší budoucnosti na základě minulosti. Obecně to pro spojité případ popisuje následující vztah:

$$u(t) = k_p \cdot e(t) + k_d \cdot \dot{e}(t) \quad (3.2)$$

kde $e(t)$ je odchylka naměřené hodnoty od požadované a $\dot{e}(t)$ je derivace odchylky. V našem diskretním případě bude mít rovnice následující tvar:

$$u = k_p \cdot (y - y_{ref}) + k_d \cdot ((y - y_{ref}) - (y_{-1} - y_{ref})) \quad (3.3)$$

kde k_d je derivační konstanta a y_{-1} je hodnota naměřená senzorem v předchozím měření. Tento regulátor se nazývá PD regulátor. V našem případě se jedná o nejvhodnější regulátor.

Pro úplnost ještě uvedu možnost přidání integrační složky, kterou v našem případě nelze použít viz kapitola 3.2. Integrační složka zrychluje ustálení odchylky, ale může způsobit rozkmitání robota (soustavy). V diskretním případě funguje tak, že sčítá všechny odchylky od naměřené hodnoty za celou dobu běhu programu. Výsledný regulátor se nazývá PID regulátor a pro akční veličinu platí:

$$u = k_p \cdot (y - y_{ref}) + k_d \cdot ((y - y_{ref}) - (y_{-1} - y_{ref})) + k_i \cdot (I + (y - y_{ref})) \quad (3.4)$$

kde k_i je integrační konstanta a I je suma všech odchylek naměřené hodnoty od referenční. V programu je nutné nastavit počáteční hodnotu rovnu nule.

3.4.2 Ziegler-Nicholova metoda

Dále popíšu jednu z možností, jak určit hodnoty konstant regulátoru: k_p , k_d a k_i . Vzhledem k charakteru úlohy jsem se rozhodl použít Ziegler-Nicholovu metodu nastavení regulátoru. Jedná se o experimentální metodu, která nám nedá přesné hodnoty konstant regulátoru, ale určí nám přibližné hodnoty, které podle chování systému upravíme tak, aby chování robota odpovídalo požadavkům.

Nejdříve si zadefinujeme pojem kritické zesílení označované k_c . Jedná se o velikost konstanty proporcionální složky regulátoru k_p , při které je systém na mezi stability. Pokud to převedeme do našeho případu robota sledující černou čáru, jedná se o situaci, kdy robot viditelně kmitá okolo hrany čáry. Pokud by jsme dále zvyšovali k_p , oscilace by se zvětšovaly, až by robot přešel černou čáru a ztratil by se.

Postup nastavení konstant je následující:

- 1) Integrační a derivační konstantu nastavíme na nulu. Rychlost motorů nastavíme třeba na 25%, volíme menší rychlost, než kterou chceme mít jako finální.
- 2) Následně nastavujeme velikost proporcionální konstanty k_p , od velikosti 1 nebo i více záleží na tom, jak je napsaný program a sledujeme chování robota. Pro malé velikosti k_p má robot problém s průjezdem zatáček. Pro moc velké k_p robot osciluje okolo čáry. Proto začínáme od malých hodnot.
- 3) Postupně zvyšujeme k_p po rozumných hodnotách, to závisí na tom jak máme napsaný program, aby krok nebyl moc velký a sledujeme chování robota. Cílem je postupným zvyšováním najít takovou hodnotu k_p , při které začíná robot viditelně oscilovat. Tuto hodnotu k_p potom označíme k_c .
- 4) Pro nalezenou hodnotu k_c je následně potřeba určit periodu oscilací P_c . To je doba, za kterou se robot přesune z jedné strany od hrany černá-bílá na druhou stranu a následně se vrátí na první stranu. V případě LEGO robota se perioda oscilací pohybuje mezi 0,5 sekundou až 2 sekundami.
- 5) Poslední konstanta, kterou je potřeba určit před samotným výpočtem konstant regulátoru, je doba běhu cyklu dT . Pro PID regulátor napsaný v NXT-G se doba cyklu pohybuje v rozsahu od 0,015 do 0,020 sekundy/cyklus. Tyto hodnoty jsem použil i pro program napsaný v NXC, jehož pseudokód je uveden v Kapitole 3.5.
- 6) Získané hodnoty dosadíme do vzorečků v tabulce 3.1 a získáme konstanty pro požadovaný regulátor.
- 7) Nastavíme vypočítané hodnoty a zjistíme, jak se robot chová.
- 8) Podle chování robota upravujeme hodnoty konstant podle případů popsaných v tabulce 3.2 tak dlouho, než bude chování přijatelné.
- 9) Potom zvýšíme rychlost a znovu upravíme konstanty. To opakujeme dokud není chování robota akceptovatelné.

Ziegler-Nicholsova metoda: výpočet konstant			
Typ regulátoru	k_p	k_i	k_d
P	$0,5 \cdot k_c$	0	0
PI	$0,45 \cdot k_c$	$1,2 \cdot k_p dT / P_c$	0
PD	$0,8 \cdot k_c$	0	$k_p P_c / (8 \cdot dT)$
PID	$0,6 \cdot k_c$	$2 \cdot k_p dT / P_c$	$k_p P_c / (8 \cdot dT)$

Tabulka 3.1: Výpočet konstant pro jednotlivé typy regulátorů

3.4.3 Vliv konstant k_p , k_i a k_d na chování robota

V následující tabulce 3.2 je popsáno, jak jednotlivé změny konstant k_p , k_i a k_d ovlivňují chování robota, kde:

Doba náběhu (rise time) je definována jako doba, ze kterou se robot vrátí zpět do ustálené polohy po té, co je vnějšími vlivy nucen tento stav změnit. V našem případě je tato situace nejlépe vidět při průjezdu prvkem C. Jedná se o dobu, ze kterou se robot vrátí zpět ke hraně černé čáry.

Překmit (overshoot) je definován jako situace, kdy se robot přiblíží k hraně černá-bílá, nezastaví se před hranou, ale přejeđe na druhou stranu. Pokud by byl překmit moc moc velký, robot by přešel čáru.

Doba ustálení (setting time) je doba, za kterou se robot dostane zpět do ustálené polohy po té, co jí byl nucen vnějšími vlivy opustit. Vnějšími vlivy jsou myšleny veškeré změny směru jízdy způsobené zatačkou nebo jedním z prvků A až D.

Chyba v rovnováze (error at equilibrium) je chyba, která je způsobena tím, že se robot nikdy přesně neustálí na hraně čáry, ale stále kolem ní osciluje. Velikost oscilací je velmi malá. Pokud chceme chybu v rovnováze vynulovat, musíme použít integrační složku regulátoru.

Změny chování při zvýšení parametrů				
Parametr	Doba náběhu	Překmit	Doba ustálení	Chyba v rovnováze
k_p	snižování	zvyšování	malé změny	snižování
k_i	snižování	snižování	snižování	eliminuje
k_d	velmi malé změny	snižování	snižování	žádné

Tabulka 3.2: Změny chování robota pro jednotlivé změny parametrů regulátoru

3.5 Popis programu

V této úloze jsem si zvolil jako programovací jazyk NXC, jelikož je to nejrozšířenější jazyk na programování LEGO robota. Přestože jsem nepoužil integrační složku, nechal jsem výpočet integrační složky v programu. U výsledného programu stačí jenom správně nastavit rychlost a hodnoty konstant regulátoru. Lze s ním bez problému projet většinu černých čar stejného typu jako je v zadání. Vychází z rovnic které jsou popsány v kapitole 3.4.1.

Samotný program lze rozdělit na dvě části. První část slouží k interakci s uživatelem. Dotazuje se uživatele zda chce kalibrovat senzor nebo použít předem nastavenou hodnotu. Druhá část programu je vlastní sledování černé čáry. Je napsána podle pseudokódu který je na obrázku 3.5. Zde je nezbytné ošetřit podmínkami případy, kdy je výsledná hodnota rychlosti motoru udávaná v procentech větší než 100% nebo naopak byla menší než 0%.

Jako možné řešení uvádím dvě možné konfigurace viz videa v příloze.

V prvním případě byl použit PD regulátor. Rychlost pro pohyb na trati byla nastavena na 45%. Hodnoty konstant regulátoru byly následující: proporcionální konstanta 320 a derivační konstanta 1333.

Ve druhém případě byl použit P regulátor. Rychlost pro pohyb na trati byla nastavena na 50%. Hodnota proporcionální konstanty byla 370. V tomto případě se jednalo o systém, který se pohyboval blízko u hranice stability.

```

Kp = 1000           ! REMEMBER we are using Kp*100 so this is really 10 !
Ki = 100           ! REMEMBER we are using Ki*100 so this is really 1 !
Kd = 10000        ! REMEMBER we are using Kd*100 so this is really 100!
offset = 45       ! Initialize the variables
Tp = 50
integral = 0      ! the place where we will store our integral
lastError = 0    ! the place where we will store the last error value
derivative = 0   ! the place where we will store the derivative
Loop forever
  LightValue = read light sensor ! what is the current light reading?
  error = LightValue - offset    ! calculate the error by subtracting the offset
  integral = integral + error    ! calculate the integral
  derivative = error - lastError ! calculate the derivative
  Turn = Kp*error + Ki*integral + Kd*derivative ! the "P term" the "I term" and the "D term"
  Turn = Turn/100               ! REMEMBER to undo the affect of the factor of 100 in Kp, Ki and Kd!
  powerA = Tp + Turn            ! the power level for the A motor
  powerC = Tp - Turn            ! the power level for the C motor
  MOTOR A direction=forward power=PowerA ! actually issue the command in a MOTOR block
  MOTOR C direction=forward power=PowerC ! same for the other motor but using the other power level
  lastError = error             ! save the current error so it can be the lastError next time around
end loop forever                ! done with loop, go back and do it again.

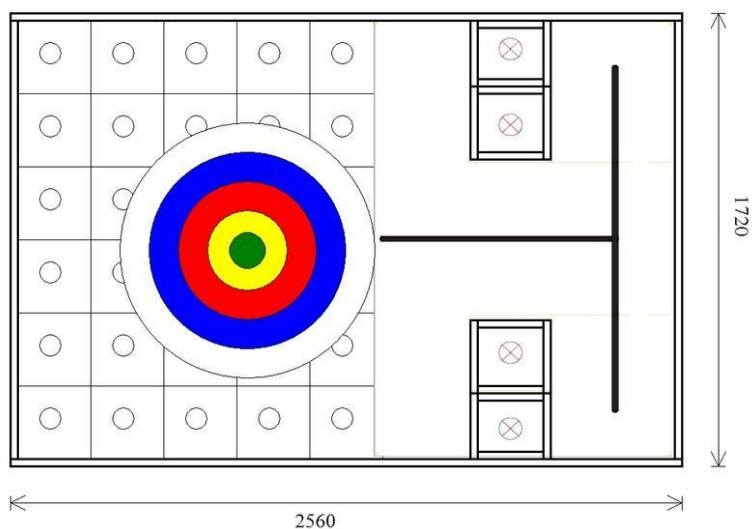
```

Obrázek 3.5: Pseudokód pro sledování černé čáry

Kapitola 4

Střelnice

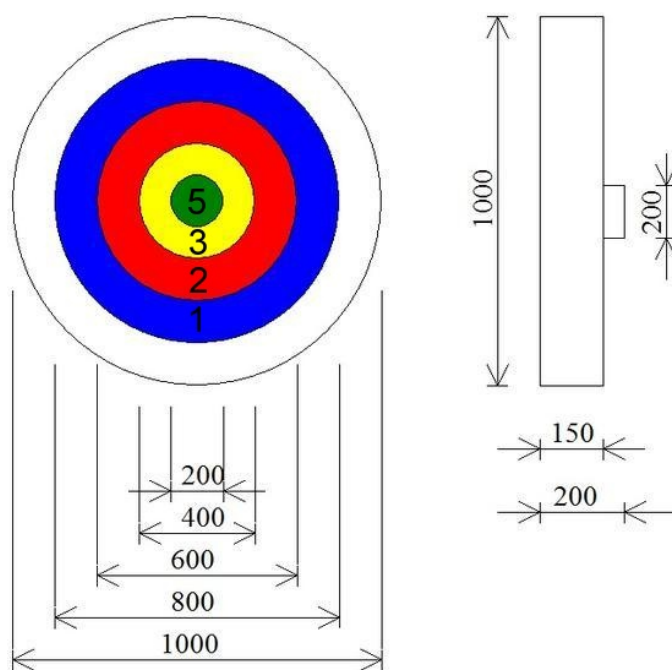
Cílem druhé úlohy je sestavit a naprogramovat robota tak, aby samostatně bez jakékoliv další pomoci (ovládání robota pomocí hlasu, bluetooth či jiných komunikačních kanálů není dovoleno) sebral z vyznačených "skladů" na hracím hřišti co nejvíce barevných míčků a umístil (vystřelil) je do cílových oblastí "Terče" podle barvy míčku. Za každý správně umístěný míček získá soutěžící příslušné bodové ohodnocení. Soutěž bude organizována jako vzájemný zápas dvou robotů na dvou samostatných identických hřištích. O vítězství rozhoduje větší počet získaných bodů.



Obrázek 4.1: Uspořádání hracího hřiště.

4.1 Popis zadání

Soutěžní úloha se bude odehrávat na vodorovné ploše o rozměrech 2560 mm x 1720 mm - viz Obrázek 4.1. Hrací hřiště má šedý laminátový povrch. Stěny hřiště jsou vysoké 75 mm. Na části hřiště je upevněn bílý papír s vyznačenou černou čarou a prostorem "skladů". Sklady jsou ze dvou stran ohraničeny stěnami hřiště a ze třetí strany jsou umístěny dva čtvercové bloky u každého skladu. Blok má rozměry 280 x 280 mm a je vysoký 75 mm. Vnitřní rozměry skladů jsou 560 x 560 mm. V každém skladu je umístěno 12 míčků, 3 od každé barvy. Průměr míčku je 60 mm. Jedná se o míčky značky Mochtoys a je možné koupit zde . Tento typ míčků byl zvolen proto, že lze jednoznačně určit barvu míčku pomocí světelného senzoru viz Kapitola 4.2. Umístění míčků v prostoru skladu je náhodné. Pro usnadnění orientace v prostoru je na ploše vyznačená černá čára ve tvaru T - viz Obrázek 4.1.



Obrázek 4.2: Rozměry terče.

Na hrací ploše je umístěn přípravek "Terč". Terč má 4 cílové oblasti (modrá, červená, žlutá a zelená) - viz Obrázek 4.2. Z pohledu shora se jedná o soustředné kružnice. Podstava přípravku je tvořena laminátovou deskou o průměru 1000 mm a výšce 18 mm. V podstavě jsou v drážkách umístěny plastové stěny, které oddělují jednotlivé oblasti. Vnější stěny mají průměry 800 mm pro modrou barvu, 600 mm pro červenou barvu, 400 mm pro žlutou barvu. Výška těchto stěn je 150 mm. Poslední zelená oblast má vnější průměr

200 mm a výška této stěny je 200 mm. Pokud tým správně umístí míček dle barvy do příslušné oblasti, obdrží 1 bod za modrý míček, 2 body za červený míček, 3 body za žlutý míček, 5 bodů za zelený míček. Celkem je na hrací ploše 24 míčků (6 od každé barvy). Maximálně lze získat 66 bodů.

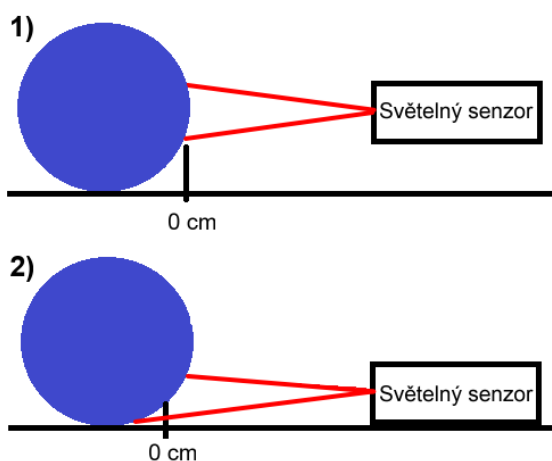
Samotná soutěž má dvě fáze. V první fázi mají soutěžící dva pokusy na to, aby získali co nejvíce bodů. Počet bodů získaných v obou kolech se sčítá. Do druhého kola postoupí 32 týmů, které získali nejvíce bodů. Počet postupujících týmu je dán celkovým počtem účastníků. Ve druhém kole jsou soutěžící rozlosováni do pavouka podle následujícího klíče: vítěz prvního kola bude nasazený proti třicátému druhému týmu z prvního kola, druhý tým z prvního kola bude nasazený proti třicátému prvnímu týmu atd.

Ve 2. kole (osmifinále, čtvrtfinále, semifinále, souboj o 3 místo a finále) budou spolu soupeřit vždy dvojice robotů. Dvojice se utká v jednom souboji. V daném souboji zvítězí ten robot, který získá větší počet bodů. Při rovnosti bodů zvítězí robot, který umístil do cílových oblastí "Terče" více barevných míčků (celkový počet míčků). V případě, že nedojde k rozhodnutí o vítězi na základě počtu bodů ani celkového počtu míčků v cílových oblastech terče, bude se souboj opakovat. Pokud ani druhý souboj nerozhodne, bude následovat hod kostkou. Zvítězí tým, který hodí vyšší číslo. Hod se může opakovat až do rozhodnutí.

4.2 Měření barvy míčků

Na určení barvy míčků, lze použít světelný senzor nebo RGB senzor. Jelikož v základní verzi stavebnice není k dispozici RGB senzor, rozhodl jsem se použít světelný senzor na rozlišování barev míčků. Následujícím měřením budu demonstrovat, jak je důležité správně nastavit vzdálenost a úhel senzoru vůči míčku. V případě míčků použitých pro tuto úlohu může dojít k záměně červené a žluté barvy nebo modré barvy a situace kdy není před senzorem žádný míček. Záměna modré a zelené barvy se může stát také, ale častěji jsem se setkal s ostatními záměnami.

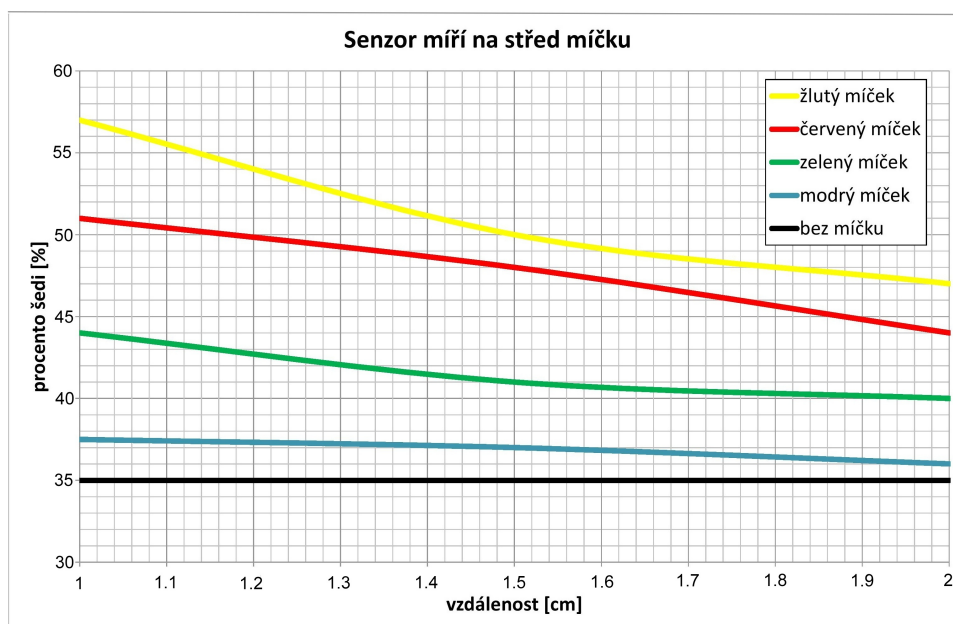
Samotné měření jsem prováděl ve dvou částech viz Obrázek 4.3. V první části jsem senzorem mířil na střed míčku. Ve druhé části jsem mířil na spodek míčku. Měření probíhalo na bílé podložce za denního světla bez žádného externího zdroje světla. Měřil jsem ve třech bodech mezi vzdálenostmi 1 až 2 cm. Výsledky měření jsou v následující tabulce 4.1.



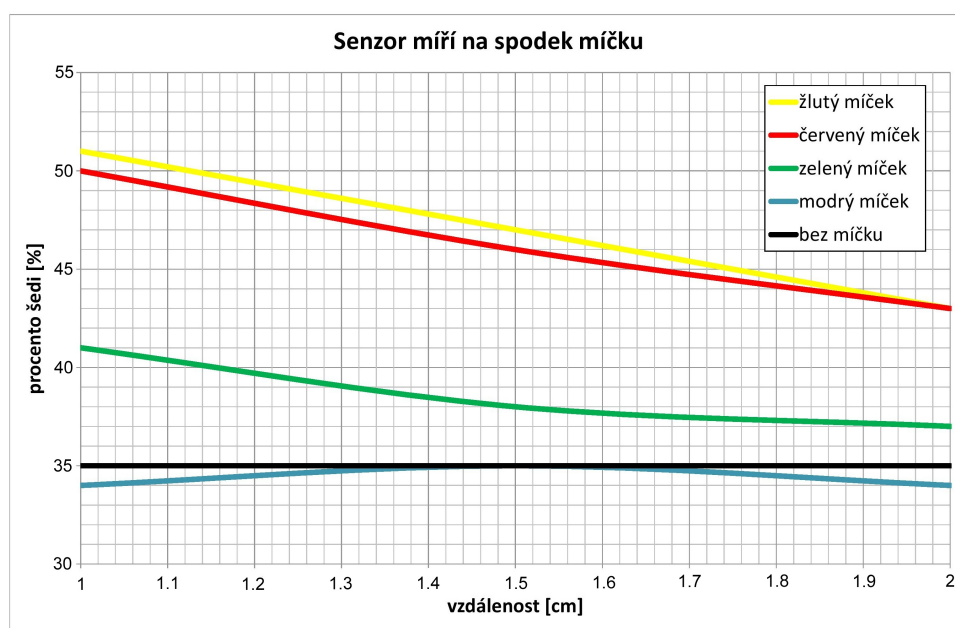
Obrázek 4.3: Způsoby měření

vzdálenost [cm]	Senzor míří na střed míčku			Senzor míří na spodek míčku		
	1	1.5	2	1	1.5	2
žlutý míček [%]	57	50	47	51	47	43
červený míček [%]	51	48	44	50	46	43
zelený míček [%]	44	41	40	41	38	37
modrý míček [%]	38	37	36	34	35	34
bez míčku [%]	35	35	35	35	35	35

Tabulka 4.1: Měření barev světelným senzorem



Obrázek 4.4: Graf měření - senzor míří na střed míčku



Obrázek 4.5: Graf měření - senzor míří na spodek míčku

Z výsledků měření, viz grafy 4.4 a 4.5, vyplývá, že ideální vzdálenost je přibližně 1 cm a senzor by měl směřovat na střed míčku. Jinak snadno dojde k záměnám, které jsem zmiňoval v úvodu kapitoly. Pokud je světelný senzor používán i k detekci míčku bez ohledu na barvu, je důležité správně nastavit hranici mezi modrou barvou a situací, kdy před senzorem není žádný míček. Bez ohledu na to, jestli je senzor nějak stíněn před okolním světlem, je vhodné před samotnou jízdou znova zkalibrovat barvy, protože míčky nejsou 100% neprůsvitné. Účelem tohoto měření bylo ukázat, že i světelný senzor je schopný jednoznačně rozlišit jednotlivé barvy.

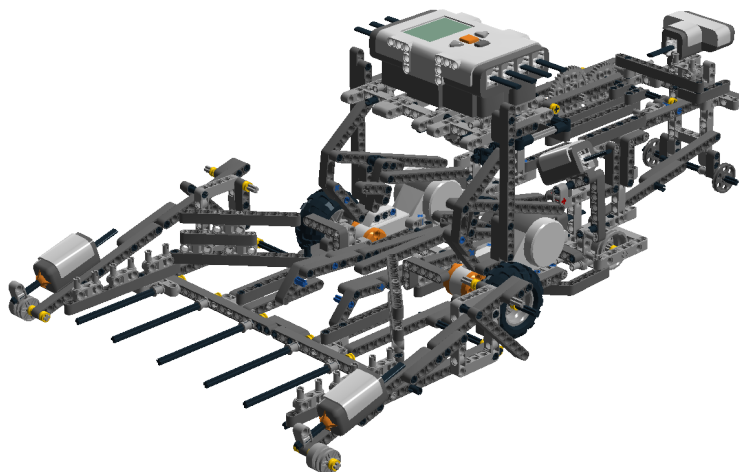
4.3 Konstrukce robota

Nejsložitější částí této soutěžní úlohy je konstrukce robota. Hlavní omezení spočívá v tom, že jsou k dispozici pouze tři motory. Ve většině konstrukčních řešení je nutné použít dva motory pro pohyb po hrací ploše a na všechny ostatní úkony zbývá pouze jeden motor.

V mém případě viz Obrázek 4.6 jsou dva motory použity na pohyb po hrací ploše. Poslední motor používám na střelbu míčků. Sensory jsem umístil následovně. Dva dotykové senzory vpředu slouží pro detekci stěny a srovnání robota o stěnu. Srovnání robota je nutné, jelikož otáčení není úplně přesné a robot by se ztratil. Ultrazvukový senzor je umístěn v zadní části a slouží

na detekci vzdálenosti od terče. Jako poslední jsem použil světelný senzor na detekci barvy míčku. Oproti předchozí úloze jsem zde nepoužil stínění senzoru před okolním světlem, protože vzdálenost senzoru od míčku je malá a stínění by blokovalo při střelbě míčků.

Samotnou konstrukci robota jsem rozdělil na dvě části, které na sebe navazují, ale je možné, řešit je samostatně. Jedná se o sběr míčků z herní plochy, převoz a následná střelba míčků na terč.



Obrázek 4.6: Celková konstrukce robota

4.3.1 Sběr míčků

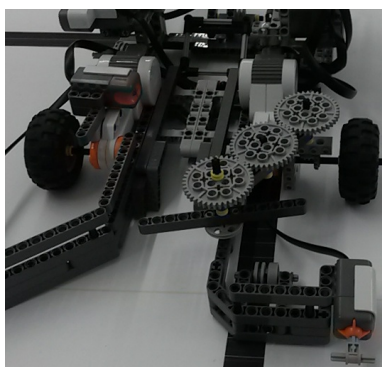
Jelikož jsem se rozhodl použít třetí motor na střelbu míčků, lze na sběr míčků použít motory, které používám současně i k pohybu robota nebo využít stěn hřiště. Nejdříve jsem zkusil několik řešení, které byly poháněny pomocí motorů přes převody viz Obrázek 4.7. První možnost řešení s použitím převodů byl otočný sběrač, který svíral s herní plochou úhel 90° a principem se podobá kombajnu. Druhou možností s využitím jednoho motoru je sběrač, který se podobá turniket. Zde byla použita 90° převodovka. Nevýhodou obou těchto řešení je, že míček musí být přesně proti sběrači, ale i přesto se může odrazit nebo zaseknout o otočnou část sběrače. Zaseknutí lze detekovat s využitím PID regulátoru na otáčení motorů. Vzhledem k malé úspěšnosti při sběru míčků jsem obě možnosti zamítnul jako nevhovující.

Pomocí teoretických pokusů jsem došel k závěru, že zasekávání míčků by šlo předejít použitím otočného zařízení, které by s podložkou svíralo úhel 45° . Bohužel se mi nepovedlo z dostupných kostek sestavit 45° převodovku.

Poslední a asi nejefektivnější způsob sběru míčků, který jsem nakonec použil, jsou tzv. "hrábě" viz Obrázek 4.8. Hrábě naberou několik míčků a následně se nárazem o stěnu hřiště zvednou a nasypou míčky do připraveného zásobníku ("kolejnice") vedoucí k dělu. Hrábě fungují na principu páky, která



(a) : Sběrač typu kombajn



(b) : Sběrač typu turniket

Obrázek 4.7: Možné sběrače míčků

vznikne mezi rameny sběrače a samotnými hráběmi. Na obou stranách tohoto sběrače míčků jsou umístěné dotykové senzory, které slouží k detekci stěny a následnému srovnání robota.



Obrázek 4.8: Sběrač míčků typu "hrábě"

4.3.2 Převoz a střelba

Nabrané míčky je potřeba nějakým způsobem dopravit ze skladu do míst, ze kterých lze střílet na terč. Vzhledem k tomu, že je 24 míčků, nabízí se několik řešení. Lze vozit míčky po jednom, ale toto řešení je časově velice neefektivní. Daleko výhodnější a celkově i rychlejší je vozit více míčků najednou. Neefektivnější by bylo sebrat všechny míčky z jednoho skladu najednou, ale tato možnost není realizovatelná při použití pouze jedné řídicí kostky. Velikost "zásobníku" je omezena rozměry robota a způsobem střelby. V mém

konstrukčním řešením se do zásobníku vejdu 3 míčky. Zásobník je tvořen dvěma kostkami, které tvoří "kolejnici" nakloněné směrem k dělu.

Na samotné dělo lze použít dva principy střelby buď katapult a nebo systém, který je podobný střelným zbraním, fungující na principu úderníku a hlavně. Zvolil jsem si střelbu pomocí katapultu. Na dostřel a trajektorii letu míčku mají vliv následující faktory: rychlost otáčení ramena katapultu, úhel otočení ramena a délka ramena. Snaha je dosáhnout co nejdelšího dostřelu. To ušetří čas, protože nebude nutné jezdit tak blízko k terči.

U katapultu platí, že čím je delší rameno, tím je delší dostřel. Další způsob, jak zlepšit dostřel, je použít převod do rychla. Případně lze přidat doraz na otáčení ramena katapultu. Přestože doraz zlepšuje přesnost střelby, není vhodné ho používat. Může dojít k poškození stavebnice, převodů nejen vnějších, ale i uvnitř servomotoru. Potencionální poškození může být způsobeno nárazem ve velké rychlosti (na poměry LEGO stavebnice). Při volbě délky ramene a pomocných převodů jsme omezeni silou použitého servomotoru a také prostorem herní plochy, ve které se má robot pohybovat. Proto jsem se snažil maximálně využít potenciál motoru. Finální verze katapultu je na Obrázku 4.9.



Obrázek 4.9: Finální verze katapultu

V případě katapultu bylo potřeba vyřešit podávání míček, aby při výstřelu nepropadávaly pod rameno katapultu. Tento problém jsem vyřešil pomocí kolébkového mechanismu, který je vidět na Obrázku 4.10.

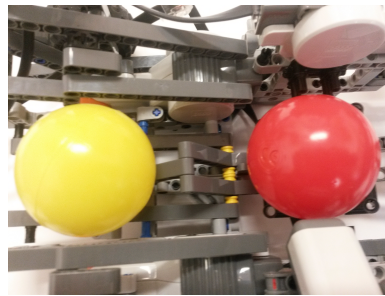
4.4 Popis programu

Stejně jako v předchozí úloze se program skládá ze dvou částí. Na vývojovém diagramu viz Obrázek 4.11 je červenou čárkovanou čarou znázorněna hranice mezi těmito částmi.

Část, která je napravo od červené čáry, vyžaduje interakci s uživatelem



(a) : Podavač bez míčků



(b) : Podavač s míčky

Obrázek 4.10: Podavač míčků

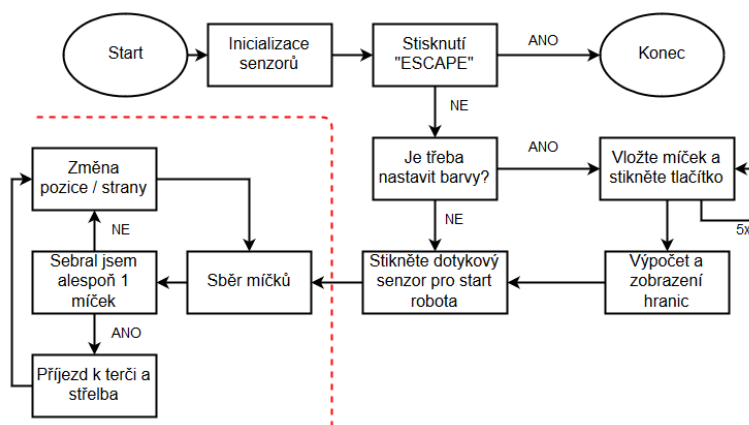
(soutěžícím). Tato část musí být provedena před samotným startem soutěžního kola. V této úloze je velice důležité správně určit barvu míčku. Hlavní funkcí této části je nastavení hranic k rozpoznávání barev míčků. Jelikož soutěž probíhá skoro celý den, mohou se měnit světelné podmínky v místnosti. Před každým kolem je tedy vhodné znovu nastavit hranice pro určování barev. O toto nastavení se starají dvě metody `setBallsColor()` a `calculate(int array[])`. Fungují tak, že se postupně vkládají míčky pod světelný senzor dle barev napsaných na display. Po vložení míčku se stiskne prostřední tlačítko na kostce, aby došlo k načtení hodnoty barvy míčku v odstínu šedi. Nakonec se ještě provede měření bez vloženého míčku, jelikož při sběru míčků detekují, zda jsem sebral nějaký míček (alespoň jeden), abych zbytečně nejezdil na střelbu a zpět.

Pomocí stisku dotykového senzoru přejdeme do druhé části, která je zcela autonomní. Záměrně se robot spouští stisknutím dotykového senzoru a ne pomocí tlačítek na kostce. Důvodem je, že soutěž probíhá jako souboj dvou robotů a jde tam o čas. Tlačítka na kostce občas nezareagují napoprvé a někdy o výsledku rozhodují poslední vteřiny. Tato část kódu obsahuje metody, které se slouží k otáčení robota, sběru míčků, střelbě a metody, které sdružují jednotlivé sekvence pohybů, aby byl kód přehledný.

Sběr míčků probíhá podle následujícího mechanismu. Najedu do levého skladu a pokusím se sebrat míčky. Míčky sbírám pomocí zvednutí "hrábí". Hrábě zvedám dvakrát. Pokud žádný neseberu, zkusím tam najet ještě jednou. Mohlo se stát, že byly míčky špatně nabrány. Pokud ani podruhé nic neseberu, změním pozici. Takto zkusím do skladu najet celkem na tři různé pozice. Potom změním sklad z levého na pravý. Princip se opakuje. A po třech změnách pozice se znovu vracím do levého skladu. Přestože je ve skladu 12 míčků a robot na tři naložení uveze maximálně 9 míčků, je výhodnější najet do druhého skladu, kde je více míčků, než hledat poslední tři. Míčky se snadno odrážejí od stěn a robota, takže se může stát, že se mohou pohybovat libovolně po herní ploše. Počty opakování jednotlivých kroků byly určeny tak, aby se dosáhlo největší efektivity ze nejkratší čas.

O samotnou střelbu míčků se starají dvě metody `shot()` a `moveShot(int curPosition, int newPosition)`. První zjišťuje barvu míčku a střílí tzn. otáčí ramenem katapultu o pevně dané úhly. Druhá metoda posouvá celým robotem

podle barvy míčků. Katapult je nastaven tak, aby střílel na jednu vzdálenost a podle barvy míčku se mění vzdálenost mezi terčem a robotem.



Obrázek 4.11: Vývojový diagram střelnice



Kapitola 5

Tvorba webových stránek

Poslední částí této práce je tvorba webových stránek k navrhovaným soutěžním úlohám. Tyto stránky budou součástí webu Robosoutez.cz. Webové stránky budou sloužit k prezentaci předmětu B3B35RO Roboti a Robosoutěže pořádané na fakultě elektrotechnické ČVUT v Praze.

Vzhledem k tomu, že byly v letošním roce vytvořené nové stránky, které jsou spravovány pomocí redakčního systému, příprava podkladů spočívá v přípravě samotného textu a obrázku včetně popisků. Není třeba řešit formátování ani psát žádné skripty. Součástí textu je podrobný popis navrhované soutěžní úlohy a popis pravidel. Při přípravě podkladu pro webové stránky jsem vyšel ze struktury současných stránek. Podklady jsou k dispozici na přiloženém CD.

Kapitola 6

Závěr

V práci se mi podařilo splnit všechny body ze zadání. V úvodní části jsem zmínil možnost stavebnice a senzory, které je možné použít na řešení mnou navrhovaných úloh. Dále jsem zmínil možnosti programování, které považuji za nejčastěji používané. Přestože je stavebnice LEGO Mindstorms EV3 ve všech ohledech lepší a výkonnější, rozhodl jsem se realizovat navrhované úlohy ve starší verzi stavebnice LEGO Mindstorms NXT. Z důvodu, že pokud lze navrhovanou úlohu splnit se starší verzí stavebnice, tak s novou verzí ji lze splnit také. I když se na Robosoutěžích pomalu rozšiřuje nová verze stavebnice, mnoho studentů používá stále starší verzi stavebnice. Dle výsledků minulých soutěží je vidět, že při vhodně navrženém řešení jsou si obě stavebnice v podstatě rovné.

Při volbě programovacího jazyka jsem vycházel ze znalostí z předchozích předmětů. Protože jsem nepotřeboval k řešení použít výhod, které přináší leIOS, zvolil jsem si jako programovací jazyk NXC. Jeho výhodou je přehledný a jednoduše čitelný kód.

V první navrhované úloze jsem řešil sledování černé čáry, která obsahuje speciální prvky. V konstrukční části jsem pozornost věnoval hlavně stínění samotného senzoru. Na sledování černé čáry byl použit PD regulátor, který byl navržen pomocí Ziegler-Nicholsovy metody. V případě této čáry by k výraznému zrychlení a zvýšení spolehlivosti při průjezdu pomohlo použití tří světelných senzorů umístěné v řadě vedle sebe kolmo k černé čáře.

V druhé navrhované úloze, která se nazývá střelnice, jsem řešil způsob jakým sebrat míčky z herní plochy a dopravit je do terče. Metoda sběru míčků pomocí "hrábí" se ukázala jako nejefektivnější metoda a konstrukčně nejjednodušší. Místo komplikovaných převodů bylo lepší používat stěny hřiště. Samotná střelba je velice obtížná část, protože je třeba trefit míčkem o průměru 6 cm, který se snadno odráží, do mezery o velikosti 10 cm. Moje řešení střelby pomocí katapultu zasáhne terč s pravděpodobností větší než 90%. Zhruba 70-80% vystřelených míčků zůstane v terči a neodrazí se ven. Přibližně 40-60% míčků je trefeno do správné barvy. Rozptyl posledního

číslo je daný tím, že na přesnost střelby má vliv napětí akumulátoru, který se docela rychle vybíjí. To může být způsobeno jeho opotřebením. Způsob, jakým je možné vylepšit současné řešení, by byl rozšířit kapacitu zásobníku na míčky v robotovi ideálně na 12 míčků. Výrazně by se zkrátila doba, kdy robot jezdí po herní ploše a byla by větší šance, že posbírá všechny míčky. Nejsem si jistý, zda by toto bylo realizovatelné za pomoci pouze jedné řídicí jednotky.

K navrhovaným úlohám jsem vytvořil webové stránky, které obsahují popis úloh a pravidla doplněné o videa s možným způsobem řešení.

Příloha A

Literatura

- [1] BĚLÍK, Tomáš, 2010. Využití robota Lego Mindstorms - Návrh a realizace speciálních úloh. ČVUT FEL v Praze. Bakalářská práce.
- [2] MARTINEC, Dan, 2010. Využití robota Lego Mindstorms při výuce předmětu A3B99RO Roboti. ČVUT FEL v Praze. Bakalářská práce.
- [3] History of LEGO Robotics. [online], [cit. 2017-04-22]. Dostupné z: <https://www.lego.com/en-us/mindstorms/history>
- [4] ASML Techno Challenge 2013 [online], [cit. 2017-02-22]. Dostupné z: https://www.youtube.com/watch?feature=player_embedded&v=Dys5XvUXB8k
- [5] Robosoutěž [online], [cit. 2017-05-22]. Dostupné z: <https://robosoutez.fel.cvut.cz/>
- [6] A PID Controller For Lego Mindstorms Robots [online], [cit. 2017-04-22]. Dostupné z: http://www.inpharmix.com/jps/PID_Controller_For_Lego_Mindstorms_Robots.html
- [7] NXT Sound sensor [online], [cit. 2017-05-10]. Dostupné z: http://www.convict.lu/htm/rob/NXT_sound_sensor.htm
- [8] Robotické vzdělání LEGO Mindstorms, Gyroskopický senzor [online], [cit. 2017-04-22]. Dostupné z: <https://lego.zcu.cz/web/rozsirujici-moduly-nxt/gyroskopicky-senzor>
- [9] NXC: Version 1.2.1 r5 [online], 2011. [cit. 2016-11-23]. Dostupné z: http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC_Guide.pdf
- [10] BENEDETTELLI, Daniele, 2007. Programming LEGO NXT Robots using NXC: (beta 30 or higher) [online]. [cit. 2016-11-23]. Dostupné z: http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC_tutorial.pdf

- [11] LEGO Digital Designer [online], [cit. 2017-02-23]. Dostupné z: <http://ldd.lego.com>
- [12] BAUER, Jiří, 2016. Využití robota Lego Mindstorms - návrh a realizace speciálních projektu. ČVUT FEL v Praze. Bakalářská práce.
- [13] LEGO Engineering [online], [cit. 2017-04-30]. Dostupné z: <http://www.legoengineering.com/nxt-sensors/>
- [14] LEGO® Digital Designer 4.3 User Manual [online], [cit. 2017-05-15]. Dostupné z: cache.lego.com/r/ldd/-/media/digital%20designer/program/version%204%203/ldd%204%203%20en-manual.pdf?l.r2=1042722492
- [15] History of LEGO. [online], [cit. 2017-04-22]. Dostupné z: <http://bzpowercircle.wikia.com/wiki/LEGO>



Příloha B

Obsah CD

B.1 Sledování černé čáry

Adresář obsahuje program pro řízení robota, jeho model v LEGO Digital Designeru a videa ukazující funkčnost řešení úlohy sledování černé čáry.

B.2 Střelnice

Adresář obsahuje program pro řízení robota, jeho model v LEGO Digital Designeru a videa ukazující funkčnost řešení úlohy střelnice.

B.3 Webové stránky

Adresář obsahuje textové podklady, obrázky a video, které zobrazuje možné řešení, pro tvorbu webových stránek.

■ B.4 BP-Urvalek

Soubor BP-Urvalek.pdf je elektronická verze tohoto dokumentu.