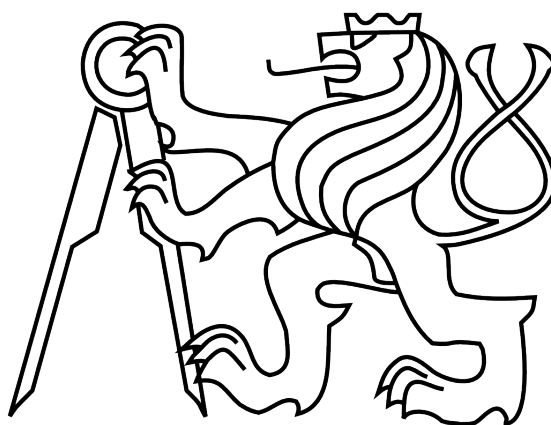


**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

**FAKULTA ELEKTROTECHNICKÁ**

**Katedra kybernetiky**



**Využití robota LEGO MINDSTORMS při výuce -  
návrh soutěžních úloh**

Bakalářská práce

Autor: Jakub Dubovic

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

Praha, 2015



## **Prohlášení autora práce**

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne .....

.....

Podpis autora práce





## Poděkování

Tímto bych chtěl poděkovat Ing. Martinu Hlinovskému, Ph.D., vedoucímu bakalářské práce, za podnětné připomínky, cenné rady a skvělou spolupráci.



## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

**Student:** Jakub Dubovic  
**Studijní program:** Kybernetika a robotika (bakalářský)  
**Obor:** Robotika  
**Název tématu:** Využití robota LEGO MINDSTORMS ve výuce - návrh soutěžních úloh

### Pokyny pro vypracování:

1. Navrhněte a realizujte dvě nové soutěžní úlohy pro robot LEGO Mindstorms s návrhem řízení v programovacích prostředích NXT-G, NXC nebo LeJOS-NXJ.
2. Jedna soutěžní úloha bude Robot přelézající schody ("Stair climber") a druhá Robot prohledávající plochu hracího hřiště a snaží se změnit barvu modulů, které nesvíí požadovanou barvou ("Dobývání území").
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 bakalářské práce:** Ing. Martin Hlinovský, Ph.D.

**Platnost zadání:** do konce letního semestru 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic  
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 9. 2. 2015

## Anotace

Tato práce pojednává o využití robotů LEGO Mindstorms při výuce. Součástí práce je seznámení s robotem LEGO Mindstorms, jeho současný stav, hardwarové a softwarové vybavení. Zabývá se také dvěma soutěžními úlohami. První úloha je "Robot přelézající schody". Cílem úlohy je postavit a naprogramovat robota, který v časovém limitu vyleze na co největší počet schodů. Druhá úloha je "Robot prohledávající plochu hracího hřiště a snažící se změnit barvu modulů, které nesvítí požadovanou barvou". Cílem druhé úlohy je v časovém limitu přebarvit co největší počet modulů na požadovanou barvu. Práce obsahuje teoretický i praktický rozbor úloh, jejich pravidla a možné konstrukční i programové řešení. Poslední část práce je věnována webovým stránkám k soutěžním úlohám.

Klíčová slova: Robot, LEGO, Mindstorms, NXT, schody, změna barvy

## Abstract

This bachelor thesis discusses using LEGO Mindstorms robots for education. The thesis includes the introduction of the LEGO Mindstorms robot, its current state, hardware and software equipment. It also describes two competitive assignments. The first assignment is "Stair climber". The objective of the assignment is to construct and program a robot that will climb the highest possible number of stairs in the time limit. The second assignment is "Robot exploring a playground and trying to change the color of modules emitting light of a different color than the required one". The objective of this assignment is to construct and program a robot that will change the highest possible number of lights to the required color in the time limit. The thesis includes the theoretical and practical analysis of the assignments, their rules and possible design and programming solutions. The last part of the thesis is dedicated to the web pages of the assignments.

Keywords: Robot, LEGO, Mindstorms, NXT, stairs, color change

# Obsah

<b>1</b>	<b>Úvod do bakalářské práce</b>	<b>1</b>
1.1	Cíl práce . . . . .	1
1.2	Obsah práce . . . . .	1
<b>2</b>	<b>Seznámení s robotem LEGO Mindstorms</b>	<b>2</b>
2.1	Současný stav . . . . .	2
2.2	Robot LEGO Mindstorms . . . . .	2
2.3	Hardwarové vybavení robota . . . . .	3
2.3.1	Inteligentní NXT kostka . . . . .	3
2.3.2	Motory . . . . .	4
2.3.3	Senzory . . . . .	5
<b>3</b>	<b>Softwarové vybavení robota</b>	<b>8</b>
3.1	Možnosti programování . . . . .	8
3.2	NXT-G . . . . .	8
3.3	NXC . . . . .	8
3.4	leJOS . . . . .	9
3.5	Závěr . . . . .	10
<b>4</b>	<b>Návrh soutěžních úloh</b>	<b>11</b>
4.1	Soutěžní úlohy . . . . .	11
4.2	Úloha č.1: Robot přelézající schody ("Stair climber") . . . . .	11
4.2.1	Cíl úlohy . . . . .	11
4.2.2	Pravidla . . . . .	12
4.2.3	Konstrukční řešení . . . . .	13
4.2.4	Programové řešení . . . . .	16
4.2.5	Závěr . . . . .	18
4.3	Úloha č.2: Robot prohledávající plochu hracího hřiště a snažící se změnit barvu modulů, které nesvítí požadovanou barvou ("Dobývání území") . . . . .	19
4.3.1	Cíl úlohy . . . . .	19
4.3.2	Pravidla . . . . .	20
4.3.3	Konstrukční řešení . . . . .	21
4.3.4	Programové řešení . . . . .	28
4.3.5	Závěr . . . . .	31
<b>5</b>	<b>Webové stránky</b>	<b>32</b>
<b>6</b>	<b>Závěr</b>	<b>33</b>

# 1 Úvod do bakalářské práce

## 1.1 Cíl práce

Cílem bakalářské práce je seznámení se s robotem LEGO Mindstorms a jeho potenciálem pro výuku studentů předmětu Roboti[10]. K tomu je nezbytné zjistit, jaké jsou hardwarové parametry robota a možnosti jeho softwarového programování. Dalším cílem je návrh a realizace dvou soutěžních úloh, které studentům umožní funkce robota poznat a naučit se s ním pracovat.

Tohle téma bakalářské práce jsem si vybral, protože předmět Roboti mě motivoval k dalšímu studiu a inspiroval k výběru studijního oboru robotika. Poprvé jsem si v něm vyzkoušel a osvojil základy programování. Domnívám se, že stavebnice LEGO Mindstorms je v současnosti jedním z nejzajímavějších výukových prostředků. V neposlední řadě mě zaujaly obě soutěžní úlohy. Prověří konstrukční i programátorské schopnosti studentů a jejich schopnost týmové práce. Zároveň jsou i divácky zajímavé.

## 1.2 Obsah práce

Práce je, nepočítám-li úvod a závěr, rozdělena na čtyři části. Druhá kapitola seznamuje s robotem LEGO Mindstorms a pojednává o jeho hardwarovém vybavení. Třetí kapitola popisuje softwarové vybavení a možné způsoby programování robota. Čtvrtá kapitola se věnuje jednotlivým soutěžním úlohám, jejich zadáním a pravidlům. Také je v ní vysvětlen postup konstrukčního i programového řešení úloh. Pátá kapitola se zabývá webovými stránkami k navrženým úlohám.

## 2 Seznámení s robotem LEGO Mindstorms

### 2.1 Současný stav

Stavebnice LEGO Mindstorms je v posledních letech velmi rozšířená po celém světě. O tom svědčí množství populárních soutěží, mimo jiné prestižní World Robot Olympiad (WRO)[6]. Také soutěž InfoMatrix[7] pro studenty středních škol, v jejímž letošním světovém finále získal tým pod mým vedením bronzovou medaili v kategorii sledování čáry.

Pozadu nezůstává ani využití ve školství. Školy po celém světě LEGO Mindstorms používají jako vynikající výukový prostředek. Pomáhá studentům osvojit si základy řízení, programování, robotiky a fyziky, a učí je pracovat v týmu.

O popularitě LEGO Mindstorms svědčí i nepřehledné množství videí na serveru Youtube.com. Klíčová slova LEGO Mindstorms zadaná do vyhledavače vrací přes 170 000 uživatelů nahraných videí.

### 2.2 Robot LEGO Mindstorms

Možnosti robota jsou v současnosti téměř neomezené. Existuje řada balíčků s přídatnými součástkami nebo senzory. Například firma HiTechnic produkuje Lego certifikované senzory zrychlení, síly, tlaku, gyroskopické, infračervené a magnetické senzory[8]. S takovou sensorickou výbavou může robot plnit nepřehledné množství různých funkcí, úkolů nebo měření. Záleží jen na fantazii uživatele.

K dispozici je zdarma také program LEGO Digital Designer, který umožňuje vytvářet ve virtuálním prostředí roboty z libovolného množství oficiálních součástek. Součástí programu je aplikace, která automaticky vygeneruje návod k sestavení robota.

Zároveň je ale tvorba robotů tak jednoduchá, že ji mohou zvládnout i děti (doporučená věková hranice je 10 let[9]). Pro ty je výhodou základní grafické vývojové prostředí NXT-G, které nevyžaduje znalosti programovacích jazyků. Základní balení také obsahuje několik návodů na konstrukci prvních robotů pro začátečníky.

## 2.3 Hardwarové vybavení robota

Přehled vybavení robota:

- Inteligentní NXT kostka
- 3x servomotor s rotačními snímači
- 2x Dotykový senzor
- Mikrofon
- Optický senzor
- Ultrazvukový dálkoměr

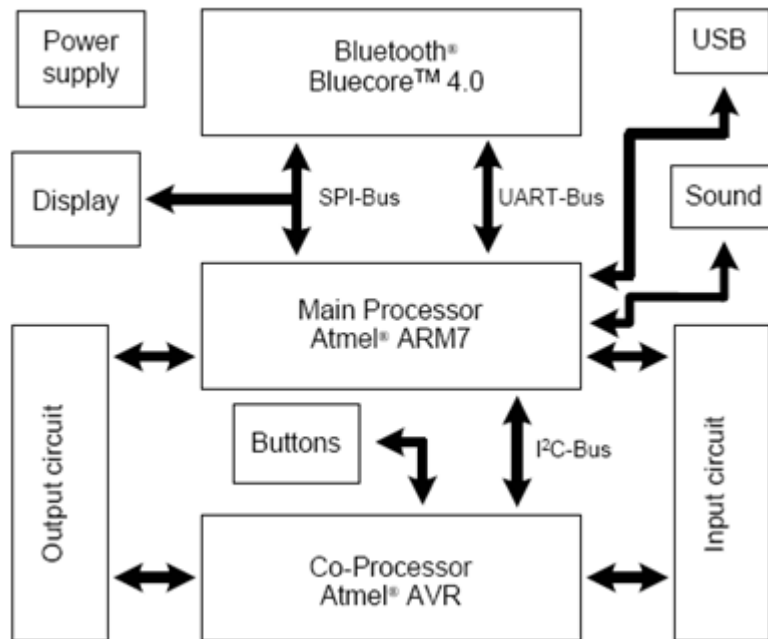
### 2.3.1 Inteligentní NXT kostka

Řídicí jednotka obsahuje 32-bitový procesor Atmel ARM7 běžící na frekvenci 48Mhz s 256kB flash paměti a 64kB RAM[11].

S počítačem kostka komunikuje prostřednictvím portu USB 2.0 nebo bezdrátově pomocí technologie Bluetooth.

Vstupní porty kostky (označené 1-4) umožňují připojení senzorů, výstupní porty (označené A-C) ovládají motory.





Obrázek 2.1: Blokové schéma NXT kostky[12].

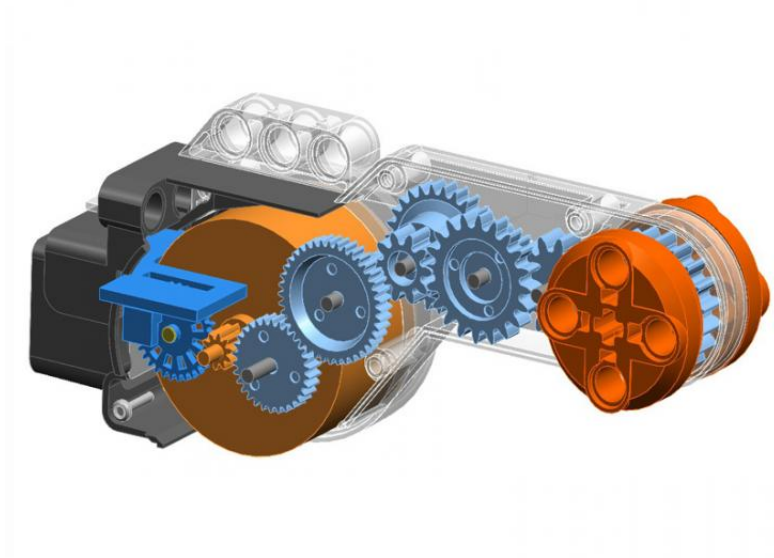
Kostka je napájena akumulátorem Li-ion nebo šesti AA bateriemi.



Obrázek 2.2: Akumulátor Li-ion[9].

### 2.3.2 Motory

Stavebnice obsahuje tři stejnosměrné elektromotory s napájením 9V. Motory mají zabudovaný optický inkrementální snímač s rozlišením 360 dílů na jednu otáčku. Díky tomu lze poměrně přesně měřit i řídit jejich rychlost[13].



Obrázek 2.3: Motor[13].

### 2.3.3 Senzory

#### Světelný senzor

Světelný senzor slouží k detekci intenzity světla. Můžeme měřit odražené světlo z přisvětlovací LED diody nebo úroveň okolního světla. Fototranzistor je nejcitlivější na infračervené záření.



Obrázek 2.4: Světelný senzor[9].

## Ultrazvukový senzor

Ultrazvukový senzor je digitální. Funguje pracuje na bázi echolokace. Z vysílací část senzoru je vypustí ultrazvukovou vlnu, která se odrazí od překážky, a je zachycena přijímací částí senzoru. Vysílání a příjem vln ovládá mikroprocesor. Senzor pracuje s frekvencí 40 kHz a je schopen měřit vzdálenosti do 2,5 m s přesností  $\pm 3$  cm. Detekce malých nebo kulatých objektů je ale velmi nepřesná. Když senzor nezachytí odraženou vlnu, vrací hodnotu 255.



Obrázek 2.5: Ultrazvukový senzor[9].

### **Dotykový senzor**

Dotykový senzor umožňuje reagovat na kontakt s překážkou. Stisknutím tlačítka se spojuje a rozpojuje jednoduchý obvod a senzor zaznamenává změnu napětí.



Obrázek 2.6: Dotykový senzor[9].

### **Zvukový senzor**

Zvukový senzor slouží k měření hladin intenzity zvuku. Rozlišuje hodnoty až do 90 dB.



Obrázek 2.7: Zvukový senzor[9].

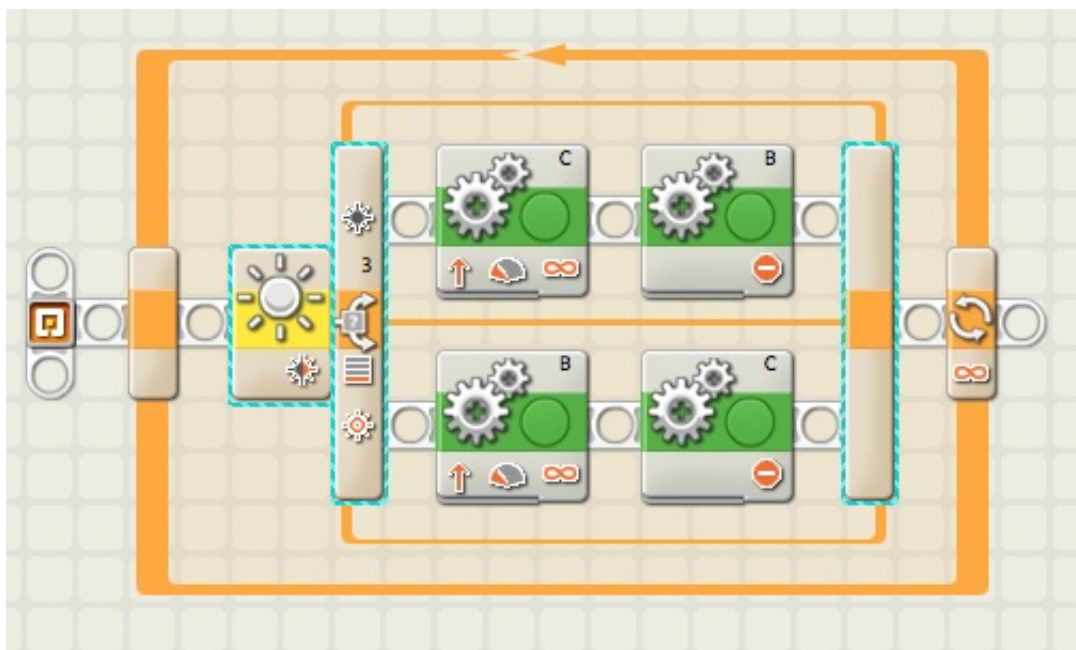
## 3 Softwarové vybavení robota

### 3.1 Možnosti programování

Inteligentní NXT kostku je možné programovat v mnoha různých prostředích nebo ovládat na dálku. Já se zaměřím na 3 nejpoužívanější programovací jazyky: NXT-G, NXC a leJOS-NXJ.

### 3.2 NXT-G

NXT-G je základní programovací prostředí, které poskytuje výrobce. Písmeno G značí, že jde o grafický jazyk. Programuje se spojováním grafických bloků dohromady a nastavováním jejich parametrů. Prostředí je primitivní a pro rozsáhlejší projekty se nehodí, hlavně kvůli časové náročnosti a nepřehlednosti.

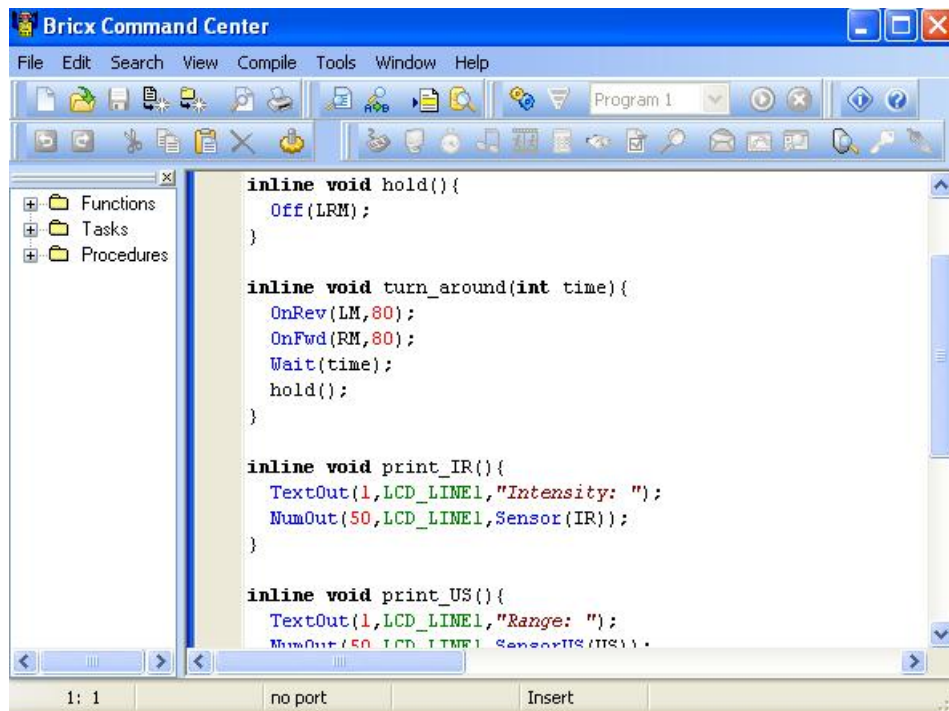


Obrázek 3.1: NXT-G - základní sledování čáry[14].

### 3.3 NXC

Prostředí NXC (Not eXactly C) je založeno na programovacím jazyku C. Svůj vlastní firmware nepotřebuje, ale jeho instalace umožňuje např. použití vícerozměrných polí a datového typu float.

NXC používá freewarové vývojové prostředí Bricx Command Center[15].

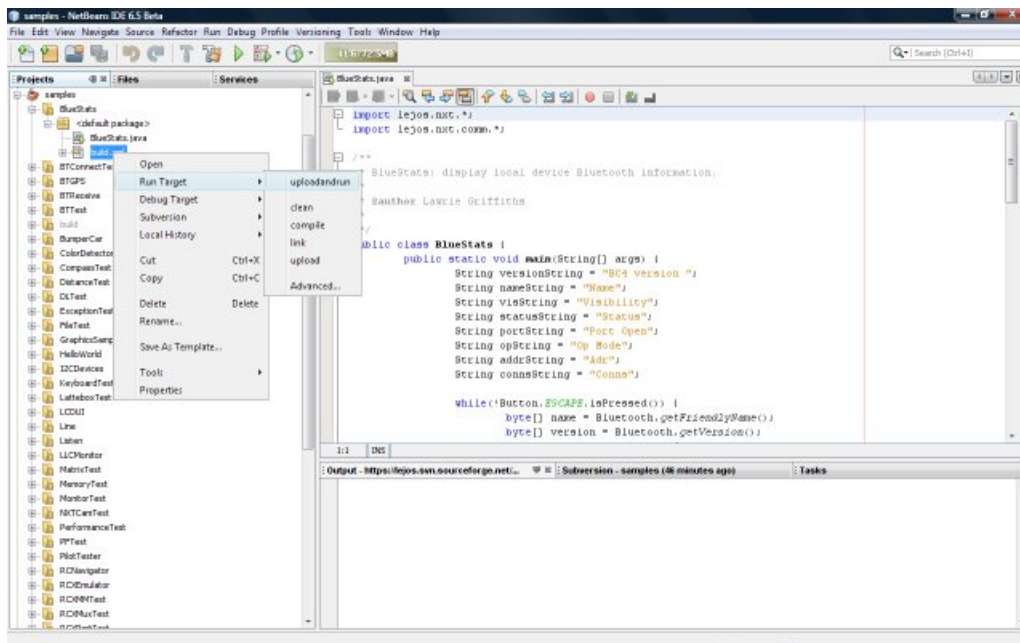


Obrázek 3.2: Bricx Command Center[15].

### 3.4 leJOS

Prostředí leJOS-NXJ (Java Operating System - Not eXactly Java) má vlastní firmware, kterým nahradí původní. Tento firmware obsahuje Java Virtual Machine, který umožňuje přímý přístup ke všem vstupům i výstupům. leJOS nemá vlastní vývojové prostředí, ale je možné programovat v IDE Netbeans nebo Eclipse.

Největší výhodou prostředí leJOS je objektově orientované programování, které díky implementovaným uživatelským třídám a knihovnám umožňuje velmi snadnou a přesnou práci se senzory i motory robota, včetně zabudovaného zpětnovazebního řízení, PID regulátoru a Kalmanova filtru. Dalšími výhodami jsou možnost rekurzivního programování, používání vícerozměrných polí a vícevláknových aplikací. Součástí je také několik testovacích programů[16].



Obrázek 3.3: leJOS v prostředí Netbeans[17].

### 3.5 Závěr

Prostředí NXT-G se k programování zadaných úloh nehodí, proto jsem se rozhodoval mezi leJOS a NXC. Vzhledem k tomu, že leJOS je určen spíše pokročilým programátorům, a pro úplné začátečníky se nehodí, zvolil jsem prostředí NXC. Je přehledné a programování v něm je pohodlné. Nepotřebuje ani vlastní firmware a Brick Command Center se dá jednoduše stáhnout a nainstalovat. Proto věřím, že ani pro studenty, kteří ještě nikdy neprogramovali, by neměl být větší problém se v NXC rychle zorientovat. Zároveň má ale NXC (obzvláště s dostupným volitelným firmwarem umožňujícím používat pokročilé programovací techniky) obrovský potenciál i pro zkušené programátory.

## 4 Návrh soutěžních úloh

### 4.1 Soutěžní úlohy

V rámci předmětu Roboti budou studenti plnit následující dvě soutěžní úlohy. Ty mají za úkol je seznámit s robotem LEGO Mindstorms, jeho základními způsoby programování, funkcí jednotlivých senzorů a s možnostmi jeho řízení. Úkolem studentů bude analyzovat daný problém, nalézt co nejlepší řešení a to potom realizovat.

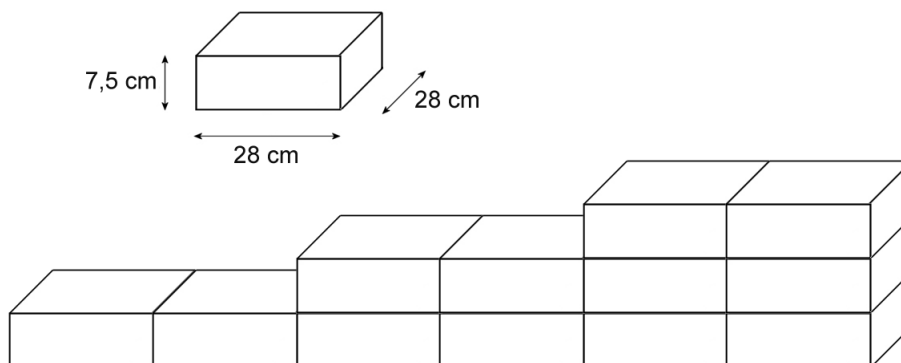
### 4.2 Úloha č.1: Robot přelézající schody ("Stair climber")

#### 4.2.1 Cíl úlohy

Úkolem první úlohy je sestavit a naprogramovat robota tak, aby v časovém limitu samostatně bez jakékoliv další pomoci (ovládání robota pomocí hlasu, bluetooth či jiných komunikačních kanálů) vylezl na co největší počet schodů. O vítězství rozhoduje počet schodů, na které robot kompletně vyleze.

Každý schod se skládá ze dvou modulů tvaru kvádrů položených za sebou. Každý modul je vysoký 7,5 cm a jeho podstava je čtverec o straně 28 cm. Rozměry každého schodu jsou tedy: délka 56 cm, šířka 28 cm a výška 7,5 cm.

Začáteční pozice robota je čelem k prvnímu (nejnižšímu) schodu, vzdálenost robota od schodu je 10 cm.



Obrázek 4.1: Schody poskládané z jednotlivých modulů.





Obrázek 4.2: Fotografie jednoho modulu[18].

## 4.2.2 Pravidla

### Vybavení

Každému týmu bude zapůjčena souprava LEGO Mindstorms Education (9797) , rozšiřující souprava dílů (9648) a síťový adaptér (9833), které vrátí po skončení soutěže organizátorovi v kompletním a neporušeném stavu. Inteligentní NXT kostku lze programovat v libovolném programovacím jazyce, ale při odevzdání musí obsahovat standardní firmware. Žádné jiné součástky ani spojovací prostředky (lepidlo, šrouby, gumičky), než zde uvedené, nesmí být při konstrukci robota použity.

### Soutěž

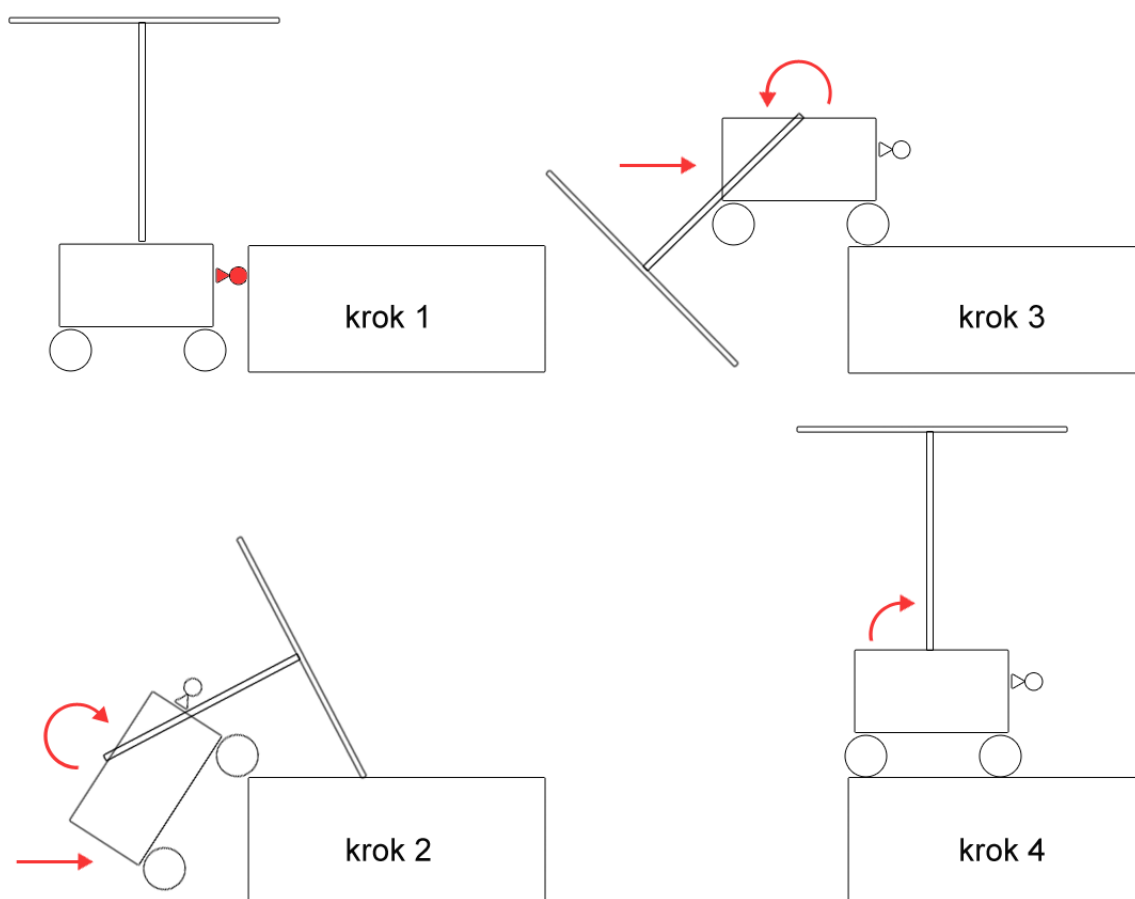
- Počáteční pozice robota je 10 cm od prvního schodu.
- Měření času začíná v momentě spuštění programu robota.
- Po skončení časového limitu je zaznamenána pozice robota a pokus končí.
- Při pádu robota ze schodiště pokus končí a počítá se nejvyšší schod, na kterém robot kompletně stál.
- Každý tým bude mít na úlohu 2 pokusy. Počítá se pokus s větším počtem zdolaných schodů.
- Mezi jednotlivými pokusy je možné vyměnit akumulátor, změnit polohu senzorů a provést úpravy v programu robota.
- Robot musí být plně autonomní a není povoleno ho v průběhu soutěže jakkoli na dálku ovládat.

### 4.2.3 Konstrukční řešení

#### Teoretický rozbor

Pro řešení úlohy jsem se rozhodl zkonstruovat robota, který jezdí na dvou párech kol. Aby dokázal vyjet na schod, zvedá se dlouhou pákou umístěnou na horní části.

Teoretické řešení je následující: Robot s pákou ve svislé poloze se pomalu pohybuje dopředu. K nalezení schodu použije dotykový senzor (krok 1). Pomocí páky zvedne svou přední část. Tou potom pomocí zadních kol, která jsou stále na zemi, najede na schod (krok 2). Páku přemístí dozadu, zvedne zadní část (krok 3). Nakonec pomocí předních kol přesune i zadní část na schod a páku opět zvedne do svislé polohy (krok 4). Tuto sekvenci akcí potom od začátku opakuje pro další schody.



Obrázek 4.3: Teoretický plán pohybu robota.

Na konstrukci robota potřebuju 3 motory (náhon na přední i zadní kola a ovládání páky). Každý pár kol tedy řídí jeden motor a robot nebude moct zatáčet. Dále budu používat dotykový senzor k detekci schodu.

### **Praktické řešení**

Robota jsem podle požadavků na motory a senzory sestavil. Motory řídící kola jsem umístil před a za inteligentní NXT kostku. Abych omezil délku robota, motory mají opačnou orientaci. Proto při pohybu dopředu musím jedním motorem otáčet v kladném směru a druhým motorem ve směru opačném - záporném.

Horní motor ovládající páku jsem upevnil k více různým částem robota, aby se při vibracích nebo působení velkých sil co nejméně pohyboval vůči zbytku robota. Při testování páky na rovině jsem zjistil, že motor nemá dost velkou sílu, aby přední nebo zadní část robota zvedl. Proto jsem mezi motor a páku umístil převod 1:5 tvořený ozubenými koly. Potom už zvedání probíhalo bez problému.

Na přední část robota jsem nakonec umístil 2 dotykové senzory po stranách předního motoru. To jsem udělal kvůli omezení délky robota a souměrnosti.

V teoretickém nákresu je vidět, že páka v některých momentech prochází tělem robota. To jsem vyřešil tak, že používám dvě rovnoběžné páky spojené v místě převodu a na opačném konci. Jejich vzájemná vzdálenost je větší, než šířka robota, takže se kolem něj můžou volně pohybovat.

Testování ukázalo, že konec páky se po povrchu schodu a podložky pohybuje trhaným neplynulým pohybem. To je patrné hlavně při současném pohybu robota a tlaku páky. Proto jsem na konce páky, které s jejím tělem tvoří písmeno 'T', upevnil kola. Ta pohyb usnadňují a experiment ukázal, že jejich hmotnost robotovi nebrání páku zvednout.



Obrázek 4.4: Hotový robot pro první úlohu.

#### 4.2.4 Programové řešení

##### 1. Nalezení schodu

Robot používá senzory jen v jedné části programu - k detekci schodu. Robot jede dopředu, dokud nezaznamená zmáčknutí jednoho z dotykových senzorů.

##### 2. Přemístění páky na schod

Po stisku dotykového senzoru přesune páku dopředu na schod. Kvůli hmotnosti a délce páky používám funkci, která pákou pohybuje rovnoměrně zpomalenou rychlostí.

$$pwr(t) = 100 - \alpha(t) \cdot \frac{100}{\varphi}$$

$pwr(t)$  je hodnota 0-100 řídicí rychlost otáčení motoru.

$\alpha(t)$  je okamžitá velikost úhlu, o který se už motor otočil.

$\varphi$  je celkový úhel, o který se má motor otočit.

To znamená, že motor se otáčí největší rychlostí, když je páka ve svislé poloze ( $\alpha(0) = 0$ ). S pohybem páky dopředu a dolů směrem ke schodu se rychlost motoru postupně snižuje. Zároveň se ale těžiště páky přesouvá před robota a na páku působí tíhové zrychlení. Kdyby rychlost motoru zůstala konstantní, páka by do schodu narazila zbytečně velkou rychlostí a robot by se mohl poškodit. Náraz by také vyvolal velké vibrace a pravděpodobně i nepředvídatelně změnil polohu robota.

##### 3. Přemístění předních kol na schod

Potom, co páku na schod položím, opět motoru páky udělím maximální rychlost. Páka už se vzhledem ke schodu nepohybuje, ale zvedne přední část robota. Současně otočím motor, který řídí zadní kola, o takový úhel, aby se přední kola dostala na schod. Úhel jsem hledal experimentálně.

##### 4. Přemístění páky na podložku

V dalším kroku potřebuju dostat páku za robota. Proto opět použiju funkci, která tam páku rovnoměrně zpomalenou rychlostí přemístí. V tomto případě páka nezačíná ve svislé poloze, ale můžu ji do ní dostat maximální rychlostí. Dál je funkce podobná, ale s významným rozdílem.

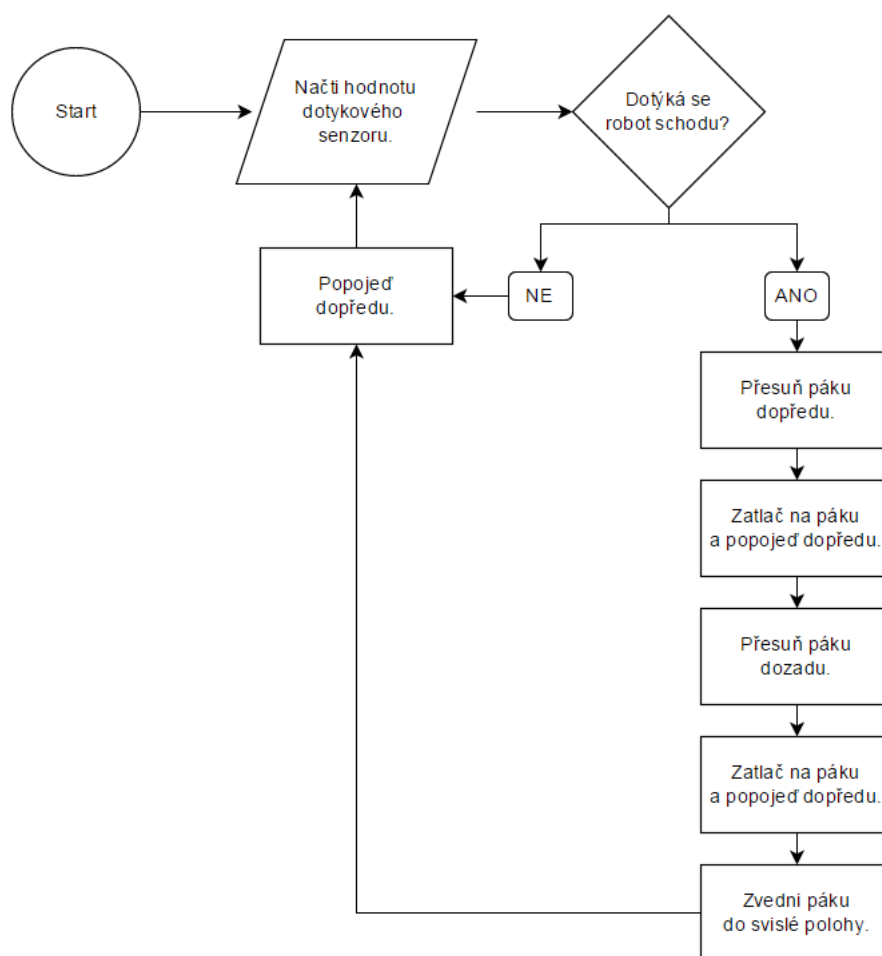
Páka se musí dostat až na zem, takže její těžiště bude ve větší vzdálenosti, než když ji robot pokládá na schod. Když se snaží ji na zem pomalu položit, převrátí se, protože nemá dost velkou hmotnost a je v nestabilní poloze s předními koly nahoře. Proto páku ještě nad zemí přestane ovládat - motor, kterým ji přemísťuje, se v určitém úhlu vypne a páku nechá spadnout. To sice robota také začne převracet, ale páka dopadne dřív, než robot spadne. Úhel, ve kterém robot vypne motor řídicí páku, jsem opět určoval experimentálně a bude rozdílný pro různé konstrukce robota.

## 5. Přemístění zadních kol na schod

Potom, co se páka dostane na zem, provede robot krok podobný kroku číslo 3. Páku tlačí na opačnou stranu, takže zvedne zadní kola. Současně se pohybuje dopředu pomocí předních kol, dokud se zadní kola nedostanou na schod.

## 5. Zvednutí páky

Robot páku maximální rychlostí zvedne (výsledné vibrace na robota mají pro náš úkol zanedbatelný vliv) a nachází se ve stejné poloze, jako je poloha výchozí. Proto může pokračovat od bodu číslo 1.



Obrázek 4.5: Diagram programu robota pro první úlohu.

#### 4.2.5 Závěr

Úloha byla na konstrukci i programování poměrně jednoduchá. Jediné problémy souvisely s dynamikou páky.

První byl převod, bez kterého motor řídící páku, robota nezvedne. Ten ale není náročný na návrh ani konstrukci.

Druhý problém byl složitější - pohybovat pákou rovnoměrně zpomaleným pohybem. V tom konstrukce nepomůže, takže to bylo nutné vyřešit programově. V jazyce NXC je to relativně jednoduché a elegantní.

V mém řešení jsem mezi jednotlivé kroky robota přidal vteřinové přestávky. Eliminuje to některé problémy s dynamikou a pro diváky je snazší sledovat jednotlivé kroky odděleně. Mé řešení není z hlediska rychlosti optimální. Pravděpodobně by pomohl rychlejší pohyb po rovině mezi jednotlivými schody. A studenti během soutěže možná vymyslí i rychlejší způsob, jak se dostat na schod. Je ale spolehlivé a funkčnost při závěrečném testování byla 100%.

## 4.3 Úloha č.2: Robot prohledávající plochu hracího hřiště a snažící se změnit barvu modulů, které nesvítí požadovanou barvou (”Dobývání území”)

### 4.3.1 Cíl úlohy

Úkolem druhé úlohy je sestavit a naprogramovat robota tak, aby prohledával dopředu známé bludiště, nacházel na podložce barevně svítící elektronické moduly a aby v časovém limitu samostatně bez jakékoliv další pomoci (ovládání robota pomocí hlasu, bluetooth či jiných komunikačních kanálů) přebarvil co největší množství modulů na zelenou barvu. O vítězství rozhoduje počet modulů, které na konci svítí zelenou barvou.

Jednotlivé moduly se nacházejí uprostřed segmentů ve tvaru čtverce o straně 28 cm. Segmenty mají rovný bílý povrch. Bludiště je složeno z 9 x 6 segmentů.

Každý modul má tvar kruhu o průměru 5 cm. Obsahuje 9 LED svítících červenou nebo zelenou barvou. V každém modulu je také umístěna Hallova sonda, která při detekci magnetického pole změní barvu světla modulu (ze zelené na červenou nebo z červené na zelenou).

Každý robot proto bude vybaven magnety. To umožňuje technický díl 3703, do kterého jsou vlisovány 3 magnety. Technický díl by měl být umístěn na robotu tak, aby jím robot mohl pohybovat nahoru a dolů a tím určovat, zda změní barvu každého konkrétního modulu. Pro změnu barvy by měl technický díl být ve výšce maximálně 10 mm nad elektronickým modulem.

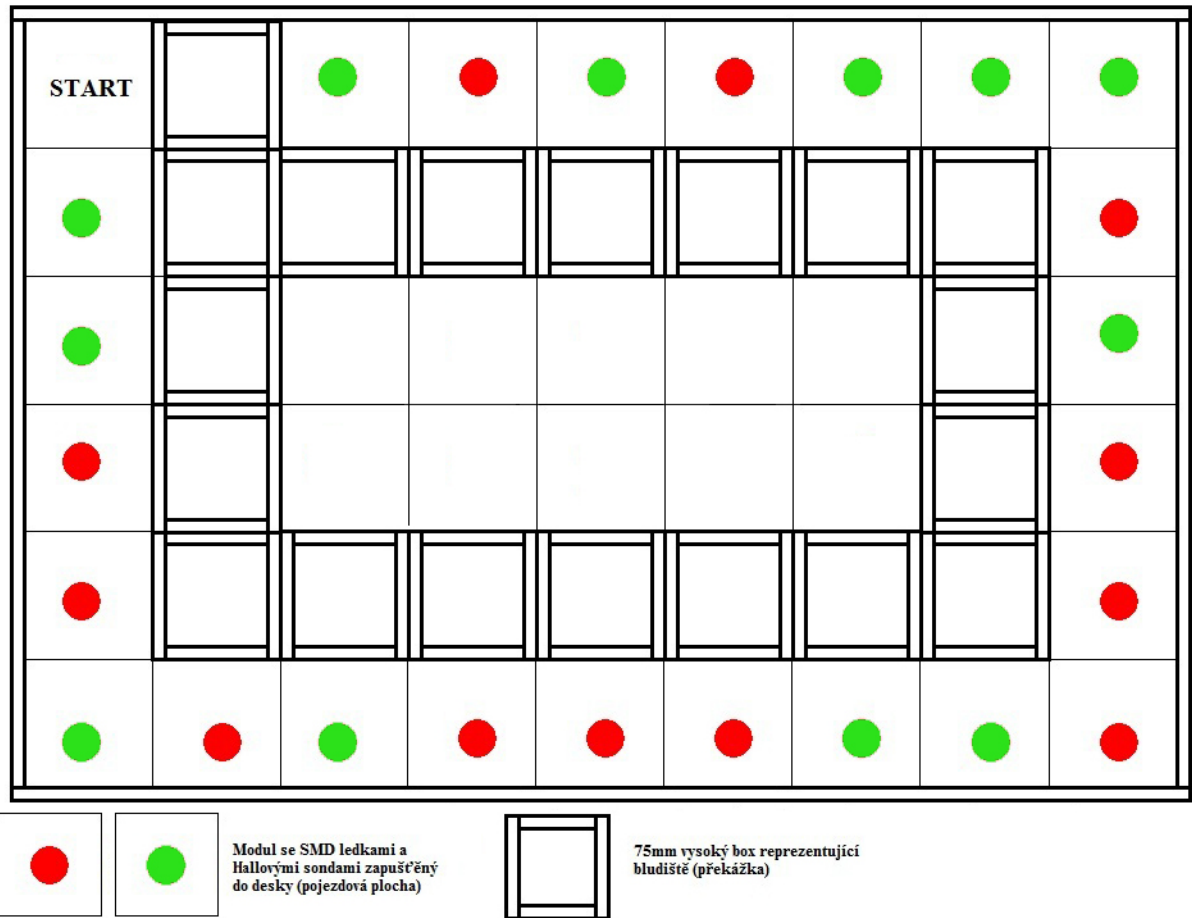


Obrázek 4.6: Technický díl 3703[18].

Na některých segmentech mapy jsou umístěny překážky. Jejich podstava má stejné rozměry jako segment a jejich výška je 7,5 cm. Překážky tvoří stejné moduly, ze kterých byly postaveny schody v předchozí úloze (Obr. 4.2).

Okraj ohraničující plochu bludiště má rovněž výšku 7,5 cm.





Obrázek 4.7: Schéma možné konfigurace bludiště[18].

### 4.3.2 Pravidla

#### Vybavení

Pravidla pro vybavení robota jsou stejná jako v předchozí úloze (4.2.2). Kromě toho bude každému týmu zapůjčen technický díl 3703, do kterého budou vlisovány 3 magnety.

#### Soutěž

- Robot je před začátkem úlohy umístěn na startovní pozici.
- Doba průjezdu začíná vyjetím ze startovní pozice a končí po uplynutí časového limitu.
- Jestliže se robot při průjezdu bludištěm zasekne, jeho tým ho může přesunout na začátek. Čas přitom dál běží a světla zůstávají ve stejném stavu.

- Každý tým bude mít na úlohu 2 pokusy. Počítá se pokus s největším počtem zelených světél.
- Robotu není dovoleno překonávat jakýmkoliv způsobem překážky umístěné uvnitř bludiště.
- Mezi jednotlivými pokusy je možné vyměnit akumulátor, změnit polohu senzorů a mostu, a provést úpravy v programu robota.
- Robot musí být plně autonomní a není povoleno ho v průběhu soutěže jakkoli na dálku ovládat.



Obrázek 4.8: Fotografie stolu s herní plochou bez rozmístěných překážek[18].

### 4.3.3 Konstrukční řešení

#### Teoretický rozbor

Pro řešení úlohy jsem se rozhodl zkonstruovat robota, který jezdí na dvou párech kol. Každé přední kolo je připojeno k vlastnímu motoru, aby robot mohl zatáčet. Třetí motor používám ke zvedání a snižování technického modulu s magnety, kterým robot dává signál Halloovým sondám. Tyto tři motory jsou pro funkčnost robota nezbytné.

Kromě toho musí být robot vybaven světelným senzorem na detekci svítících modulů.

Jestliže chceme prohledávat neznámé bludiště, vzniká problém: Každá stavebnice obsahuje jen jeden ultrazvukový senzor, pomocí kterého se robot může v bludišti orientovat.

Robot tento senzor může ovládat motorem a tak se uprostřed každého segmentu "rozhlízet". To ale v našem případě není možné, protože už všechny tři motory využíváme.

Další možnost je senzor na robota umístit napevno a nechat robota, aby se uprostřed každého segmentu celý otočil doleva a potom doprava. To je ale pro naše potřeby velmi pomalé a ubíralo by to celé soutěži na dynamice a atmosféře.

Poslední možností je nepoužívat motor pro ovládání technického modulu s magnety. To je ale nejméně efektivní řešení, protože robot při každém přebarvení musí vykonávat složité pohyby. A vzniká možnost, že i při malém odchýlení z trasy neúmyslně přebarví zelený modul, přes který přejíždí.

Z těchto důvodů jsem se rozhodl používat předem známá bludiště. Těch bude několik možných konfigurací a robot bude mít pro každou z nich speciální program. Ten jeho tým na místě vybere podle toho, které bludiště bude jejich robot procházet.

### **Praktické řešení**

Robota jsem podle požadavků na motory a senzory sestavil. Motory řídicí kola jsou po levé a pravé straně pod inteligentní NXT kostkou. Mezi nimi je motor řídicí technický modul s magnety.

Po testování robota jsem se rozhodl přidat ultrazvukový senzor, který měří vzdálenost robota od jedné stěny. Bez této zpětné vazby se robot často po několika desítkách centimetrů odchýlil z rovné trasy. Důvody mohly být různé: Nepoložil jsem robota na startovní čáru úplně rovně, jeden z motorů měl nepatrně nižší rychlost kvůli vybitým bateriím, nebo robot zatáčel vlivem nepřesného upevnění kol robota. Ultrazvukový senzor pomohl také při vyrovnávání robota po zatáčkách, které ne vždy zvládl přesně na 90°.

Dopředu jsem umístil světelný senzor. Abych našel jeho optimální polohu pro tuto úlohu, prostudoval jsem jeho výstup pro různé výšky a úhly. To jsem udělal pro oba módy - měření okolního (ambient) a odraženého (reflected) světla.

Stejná měření jsem pro srovnání zopakoval se světelným senzorem EV3. Ten má navíc mód umožňující rozpoznání barvy pomocí srovnání hodnot intenzity odraženého červeného, zeleného a modrého světla. Při testování nad barevným povrchem fungoval senzor správně. Při testování nad svítícími LED modulu ale naměřil pro všechny tři složky světla příliš vysokou intenzitu a hlásil detekci bílé barvy.

Z tohoto důvodu jsem stejně jako u NXT senzoru použil pouze módy měření intenzity okolního a odraženého světla.

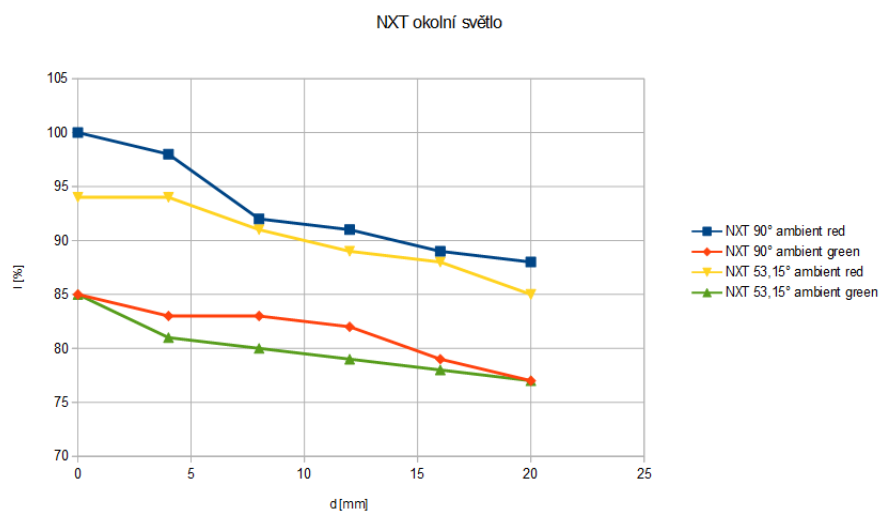
Měřil jsem pod úhly 90° a 53,13°, což je úhel, který s podložkou svírá většina úhlových nosníků ve stavebnici.

Krok měření jsem zvolil 4 mm, což je polovina délky jedné části lego nosníku (na každých 8 mm připadá 1 otvor, takže například nosník s 15 otvory má délku  $15 \times 8 = 120$  mm).

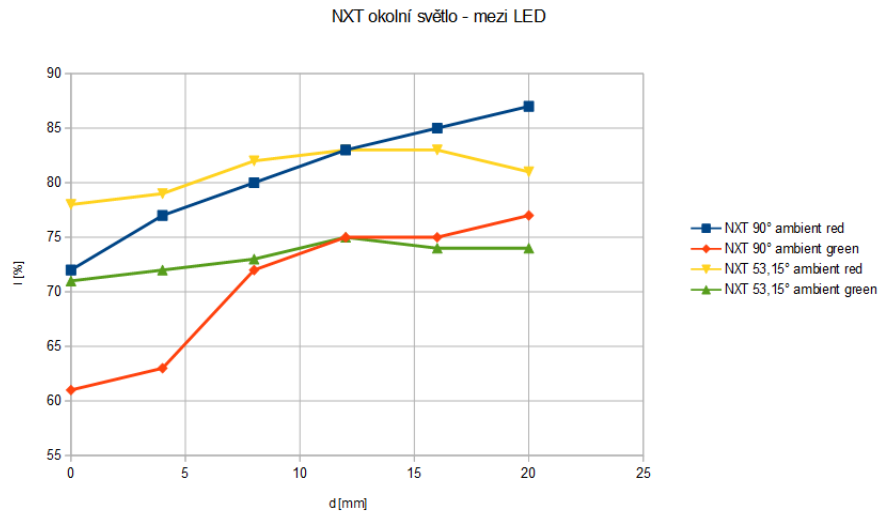
Intenzita LED		nad LED							mezi LED								
kostka	°	light	d [mm]	0	4	8	12	16	20	0	4	8	12	16	20		
EV3	90	ambient	red	56	55	54	53	52	51	40	43	45	47	49	50		
			green	48	47	47	47	46	45	34	36	39	41	43	43		
		reflected	red	41	53	33	23	15	12	39	48	32	22	14	11		
			green	41	52	36	24	16	12	39	44	34	22	15	11		
		53,13	ambient	red	54	52	52	51	50	48	42	48	48	48	47	45	
				green	50	49	46	44	44	43	37	41	42	42	41	41	
	reflected		red	30	18	12	8	5	4	26	16	9	7	4	3		
			green	31	21	11	8	5	4	26	17	8	7	4	3		
	NXT		90	ambient	red	100	98	92	91	89	88	72	77	80	83	85	87
					green	85	83	83	82	79	77	61	63	72	75	75	77
		reflected		red	100	98	94	91	91	90	78	82	83	87	88	88	
				green	88	87	85	82	80	80	69	74	74	78	79	78	
53,13		ambient		red	94	94	91	89	88	85	78	79	82	83	83	81	
				green	85	81	80	79	78	77	71	72	73	75	74	74	
		reflected	red	96	94	91	89	86	85	82	83	83	84	83	81		
			green	85	81	80	79	78	77	71	74	76	75	75	74		

Obrázek 4.9: Naměřené hodnoty intenzity světla.

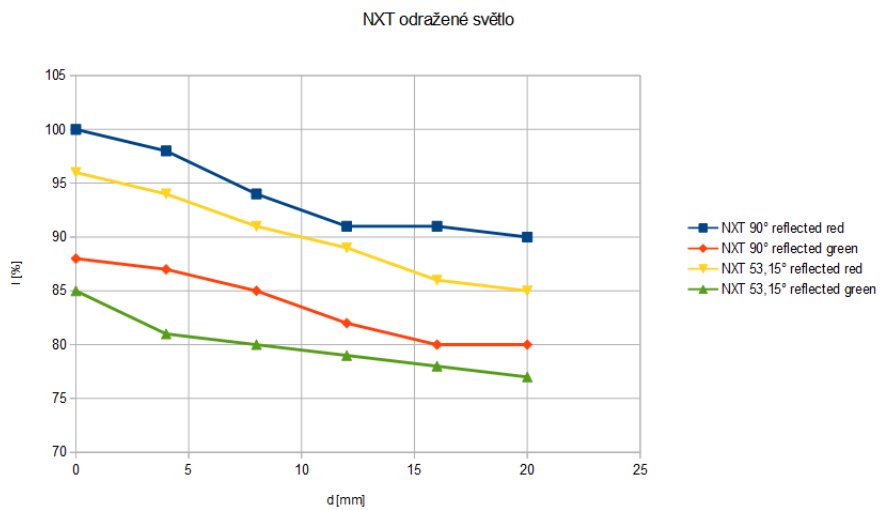
## Grafy intenzity světla pro NXT



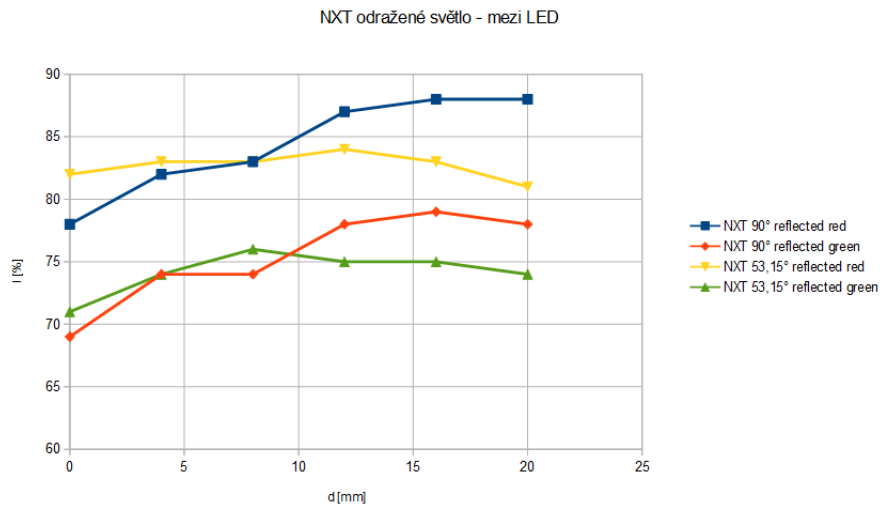
Obrázek 4.10: NXT - Intenzita okolního světla nad LED.



Obrázek 4.11: NXT - Intenzita okolního světla mezi LED.



Obrázek 4.12: NXT - Intenzita odraženého světla nad LED.



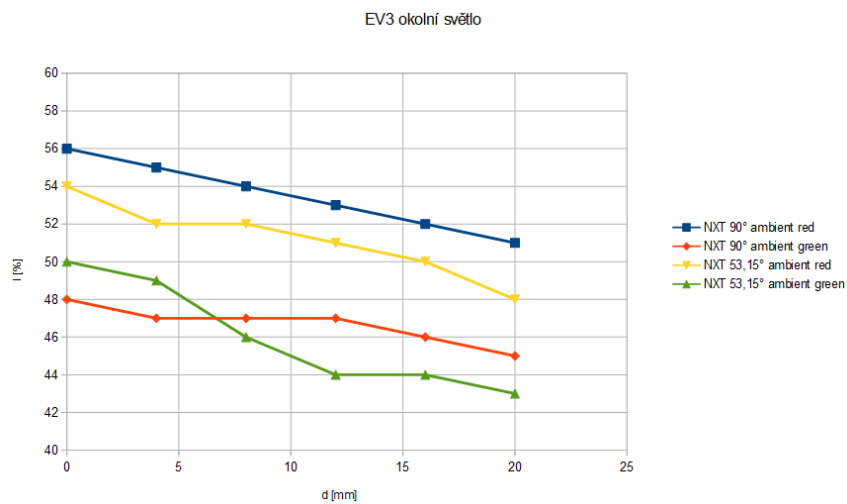
Obrázek 4.13: NXT - Intenzita odraženého světla mezi LED.

Jak je z grafů patrné, je lepší senzor neumístovat moc nízko. Pak se může stát, že robot mine LED a nesprávně naměří menší intenzitu světla (ta je například uprostřed modulu mezi jednotlivými LED). Tento problém se se vzdáleností od modulu zmenšuje, protože se jednotlivá světla překrývají.

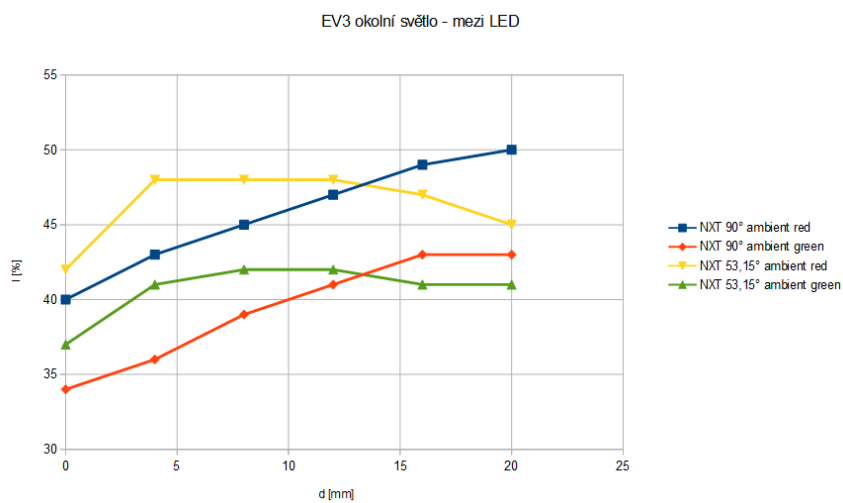
Měření jsem opakovat v horším osvětlení - šero a vypnutá umělá světla. Na naměřené hodnoty to nemělo znatelný vliv. V úplné tmě jsem měření neopakovat, protože si nemyslím, že v soutěži podobné podmínky nastanou.

Senzor jsem se rozhodl umístit do výšky 8 mm pod úhlem 78° (kvůli konstrukci robota - inteligentní NXT kostka není ve vodorovné poloze) a měřím odražené světlo. Testy ukázaly, že pro tuto konfiguraci jsou mezi červenou a zelenou barvou rozdíly dostatečné na to, aby je robot bezpečně rozlišil. Zároveň jsou nad modulem i mezi LED hodnoty intenzity světla dostatečně vysoké, takže je robot rozezná od bílé podložky. Funkční jsou ale samozřejmě i jiné konfigurace, jak je vidět z grafů.

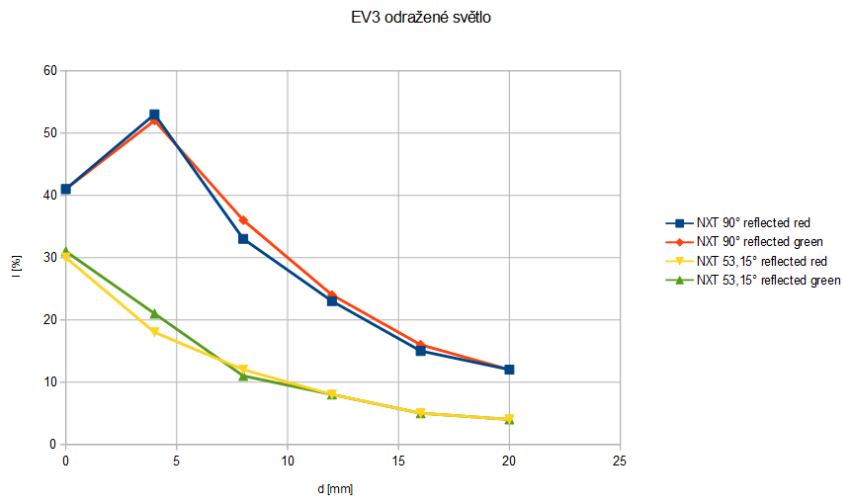
## Grafy intenzity světla pro EV3



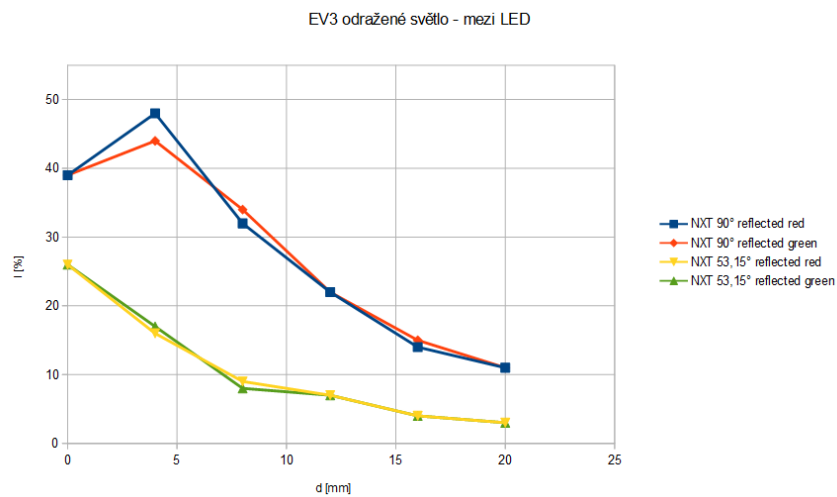
Obrázek 4.14: EV3 - Intenzita okolního světla nad LED.



Obrázek 4.15: EV3 - Intenzita okolního světla mezi LED.



Obrázek 4.16: EV3 - Intenzita odraženého světla nad LED.



Obrázek 4.17: EV3 - Intenzita odraženého světla mezi LED.

Z grafů pro EV3 je patrné, že pro rozlišení červeně a zeleně svítících LED nemůžeme použít mód měření intenzity odraženého světla. Rozdíl mezi intenzitami je velmi malý a pravděpodobně byl způsoben chybami měření.

V módu měření intenzity okolního světla ale můžeme EV3 senzor použít stejně efektivně jako NXT.





Obrázek 4.18: Hotový robot pro druhou úlohu.

#### 4.3.4 Programové řešení

Robot z výše uvedených důvodů (4.3.3) prozkoumává bludiště známých rozměrů. To jsem vyřešil tak, že program robota pro danou konfiguraci bludiště obsahuje počet hran (rovných úseků) bludiště, počet modulů na každé hraně bludiště a směr zatáček mezi jednotlivými hranami.

Robot při jízdě bludištěm počítá světla. Jejich počet si pamatuje a průběžně ho porovnává s počtem modulů dané hrany. Když dosáhne posledního modulu na hraně, zatočí na předem určenou stranu. Počet světél vynuluje, začne počítat od začátku a hodnotu srovnává s počtem modulů nové hrany. Na posledním světle poslední hrany zůstane stát.

##### 1. Nalezení světla

Robot jede dopředu. Ultrazvukový senzor měří vzdálenost od stěny a robot počítá odchylku očekávané a reálné vzdálenosti od stěny. Rychlost kol upravuje proporcionálně k odchylce.

$$dif(t) = (US(t) - d) \cdot k$$

$dif(t)$  je hodnota, o kterou robot upravuje rychlost otáčení motoru.

$US(t)$  je okamžitá hodnota reálné vzdálenosti od stěny, kterou vrací ultrazvukový senzor.

$d$  je očekávaná vzdálenost od stěny.

$k$  je proporcionální konstanta.

Hodnoty 0-100 řídící rychlosti otáčení motorů potom pro jednotlivé motory nastavím jako

$100 + dif$ , popřípadě  $100 + dif$  a  $100$ , jestliže je hodnota  $dif$  záporná.

To znamená, že čím větší odchylka, tím menší hodnotu rychlosti program předá motoru na příslušné straně. Druhý motor se otáčí maximální rychlostí, robot zatočí směrem k pomalejšímu motoru a odchylku tím vyrovná. V případě, že je odchylka nulová, pokračuje robot oběma motory maximální rychlostí dopředu.

Proporcionální konstantu  $k$  jsem určil experimentálně. Robot při ní jede dostatečně rychle (neosciluje doprava a doleva kolem ideální hodnoty vzdálenosti od stěny), ale zároveň odchylku vyrovnává tak rychle, že se celou dobu pohybuje uprostřed trasy a nemine žádná světla.

## 2. Měření intenzity světla

V okamžiku, kdy světelný senzor zaznamená větší hodnotu, než je hodnota světla odraženého bílou podložkou, našel robot svítící modul. Program přejde na smyčku, která trvá, dokud se intenzita odraženého světla opět nesníží na úroveň bílé podložky. V každé iteraci robot přečte hodnotu intenzity světla a pokud je větší, než současné maximum, zaznamená ji jako nové maximum. Po skončení smyčky přečte robot 1 k počtu světél, která našel.

## 3. Vyhodnocení světla

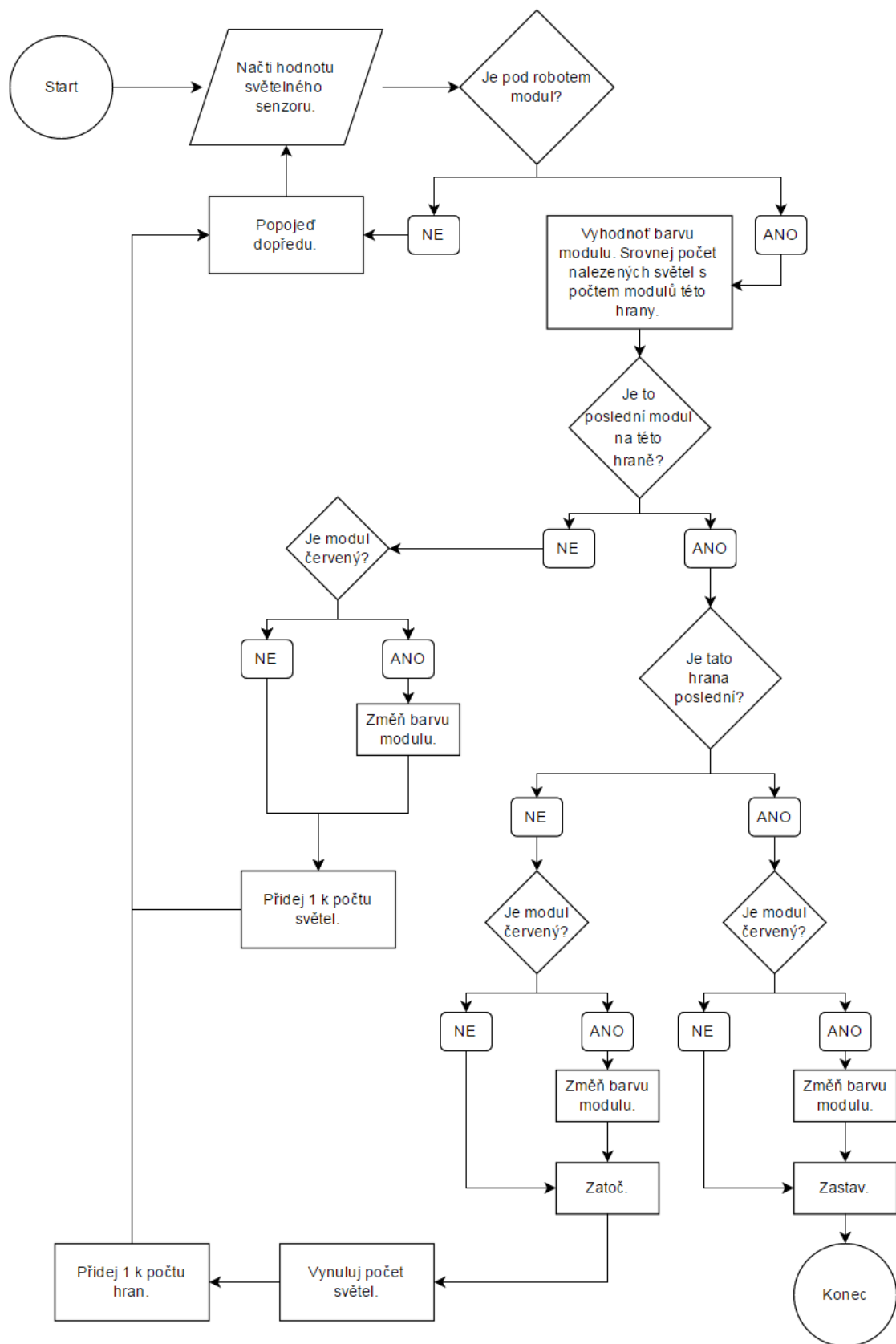
Další akce robota záleží na výsledku předchozího kroku. Nejdříve zkontroluje počet světél. Jestliže dosáhl posledního světla na hraně, zatočí na příslušnou stranu podle dané konfigurace mapy. V opačném případě pokračuje rovně.

Kromě toho se rozhoduje podle intenzity světla - jestliže intenzita překročila úroveň červené barvy, robot přejde k přebarvení světla. V opačném případě pokračuje krokem číslo 1.

## 4. Přebarvení světla

Světelný senzor je na přední části robota, technický modul s magnety na zadní. Proto se robot musí před přebarvením přesunout. To udělá tak, že jede dopředu stejným způsobem jako v kroku 1 po určitou dobu. Tu jsem určil experimentálně. Pro cestu po rovné trase je čas delší, než když robot v předchozím kroku zatočil. Potom robot zastaví a technickým modulem s magnety pohne pomocí příslušného motoru dolů a zpět nahoru tak, aby se magnety přiblížily k modulu na cca 0,5 cm a Hallova sonda je zaznamenala. Dál robot pokračuje krokem číslo 1.

Pohyb po určitou dobu může být pro přesun o danou vzdálenost nepřesný (například kvůli vybitým bateriím - robot pak za stejnou dobu ujede menší vzdálenost). Proto jsem experimentoval i s možností otočit místo toho oběma motory o určitý úhel. Potom sice robot ujede vždy přesnou vzdálenost, ale celou dobu drží počáteční směr a nemůže použít zpětnou vazbu z ultrazvukového senzoru. To může (například těsně po velké korekci ultrazvukovým senzorem nebo po zatáčce) vést ke značným odchylkám od středu trasy a robot může i minout modul. Proto používám méně přesnou, ale v tomto případě spolehlivější metodu jízdy po určitý čas. Jen je potřeba hlídat stav baterií.



Obrázek 4.19: Diagram programu robota pro druhou úlohu.

#### 4.3.5 Závěr

Úloha byla kvůli náročnosti na počet senzorů a motorů komplikovanější, než se na první pohled zdálo. Vzhledem k nepraktičnosti neustálého zastavování a otáčení se robota, aby prozkoumal okolí, jsem se rozhodl úlohu přepracovat na prozkoumávání bludiště známých rozměrů. Průzkum se potom omezuje na určování barvy světla modulů.

Po měření intenzity zeleného a červeného světla pro různé úhly, vzdálenosti a módy senzoru jsem našel konfiguraci, která je pro účely úlohy vyhovující. Robot je s ní schopen rozeznat barvu ve 100% případech.

Poslední problém - odchylování robota ze středu trasy, kde se nacházejí moduly - jsem vyřešil porporcionálním řízením pomocí ultrazvukového senzoru.

Své řešení jsem testoval na dočasné mapě, protože v době psaní této práce nebyl soutěžní stůl pro úlohu naprogramován. Ověřil jsem ale všechny klíčové funkce robota - pohyb ve středu trasy, ignorování zelených světél a přebarvování (v tomto případě vypínání) červených, zatáčení bez přebarvení i s přebarvením a počítání světél před zatáčkou.

## 5 Webové stránky

Webové stránky pro předmět Roboti popisují cíle a pravidla obou soutěžních úloh. Jsou na nich také nastíněny způsoby řešení a odkazy na videa k oběma vyřešeným úlohám. Obsahem stránek je výňatek z kapitol 4.2 a 4.3.

Při tvorbě webových stránek jsem se v rámci konzistence snažil zachovat formát i znění pravidel a obecných ustanovení z dřívějších soutěží. Také jsem použil původní fotografie soutěžní plochy a modulů bludiště[18].

## 6 Závěr

V jednotlivých kapitolách jsem shrnul hlavní aspekty stavebnice LEGO Mindstorms. Od její narůstající popularity, přes hardwarové možnosti a stručný popis jednotlivých součástí, až po výhody a nevýhody tří nejpoužívanějších programovacích prostředí.

Hlavně jsem se věnoval dvěma novým soutěžním úlohám, z nichž první procvičí obzvláště konstrukční schopnosti studentů, zatímco ve druhé se budou věnovat spíše programování. Dohromady se úlohy dotýkají oblastí fyziky, robotiky, řízení a programování. Zároveň studentům pomohou zdokonalit práci v týmu.

## Reference

### Bakalářské práce

- [1] Trojánek, Pavel. Využití robota LEGO Mindstorms při výuce, Praha 2009, bakalářská práce (Bc.). České vysoké učení technické v Praze. Katedra řídicí techniky.
- [2] Moc, Ivan. Využití robota LEGO Mindstorms – Návrh a realizace úloh, návod pro programování v NXC, Praha 2010, bakalářská práce (Bc.). České vysoké učení technické v Praze. Katedra řídicí techniky.
- [3] Bělík, Tomáš. Využití robota LEGO Mindstorms – Návrh a realizace speciálních úloh, Praha 2010, bakalářská práce (Bc.). České vysoké učení technické v Praze. Katedra řídicí techniky.
- [4] Martinec, Dan. Využití robota LEGO Mindstorms při výuce předmětu A3B99RO Roboti, Praha 2010, bakalářská práce (Bc.). České vysoké učení technické v Praze. Katedra řídicí techniky.
- [5] Mrva, Jakub. Využití robotů LEGO Mindstorms při výuce – Návrh soutěžních úloh, Praha 2011, bakalářská práce (Bc.). České vysoké učení technické v Praze. Katedra řídicí techniky.

### Internetové zdroje

- [6] World Robot Olympiad [online], (2015-05-15),  
<http://www.wroboto.org/>
- [7] InfoMatrix [online], (2015-05-15),  
<http://www.infomatrix.ro/>
- [8] HiTechnic [online], (2015-05-15),  
<https://www.hitechnic.com/sensors>
- [9] Lego Shop [online], (2015-05-15),  
<http://shop.lego.com/en-US/LEGO-MINDSTORMS-NXT-2-0-8547>
- [10] Roboti [online], České vysoké učení technické v Praze (2015-04-10),  
<http://support.dce.felk.cvut.cz/roboti>
- [11] Atmel products [online], Atmel (2015-04-10),  
<http://www.atmel.com/devices/sam7s256.aspx>
- [12] What's Inside the NXT Brick? [online], The Mindstorms NXT Review (2015-04-10),  
<http://mindstormsnxt.blogspot.cz/2006/08/whats-inside-nxt-brick.html>

- [13] NXT Motor internals [online], (2015-04-10),  
<http://www.philohome.com/nxtmotor/nxtmotor.htm>
- [14] Lego MINDSTORMS NXT Basic Line Follow Step By Step Directions [online], NeboMusic (2015-04-10),  
<http://neboMusic.net/NXT-G-BasicLineFollow.html>
- [15] Bricx Command Center [online], (2015-05-15),  
<http://bricxcc.sourceforge.net>
- [16] LeJOS, Java for Lego Mindstorms [online], NXJ Downloads (2015-05-15),  
<http://lejos.sourceforge.net/nxj-downloads.php>
- [17] Java for LEGO Mindstorms [online], (2015-05-15),  
[http://www.diy-robots.com/resources/LejosNxtTutorial\\_cn/preliminaries/usingnetbeans.php](http://www.diy-robots.com/resources/LejosNxtTutorial_cn/preliminaries/usingnetbeans.php)
- [18] Robosoutěž [online], A3B99RO Roboti (2015-05-15),  
<http://www.robosoutez.cz>
- [19] LEGO Engineering [online], Tufts university (2015-05-15),  
<http://legoengineering.com>



## Seznam obrázků

2.1	Blokové schéma NXT kostky[12]. . . . .	4
2.2	Akumulátor Li-ion[9]. . . . .	4
2.3	Motor[13]. . . . .	5
2.4	Světelný senzor[9]. . . . .	5
2.5	Ultrazvukový senzor[9]. . . . .	6
2.6	Dotykový senzor[9]. . . . .	7
2.7	Zvukový senzor[9]. . . . .	7
3.1	NXT-G - základní sledování čáry[14]. . . . .	8
3.2	Bricx Command Center[15]. . . . .	9
3.3	leJOS v prostředí Netbeans[17]. . . . .	10
4.1	Schody poskládané z jednotlivých modulů. . . . .	11
4.2	Fotografie jednoho modulu[18]. . . . .	12
4.3	Teoretický plán pohybu robota. . . . .	13
4.4	Hotový robot pro první úlohu. . . . .	15
4.5	Diagram programu robota pro první úlohu. . . . .	17
4.6	Technický díl 3703[18]. . . . .	19
4.7	Schéma možné konfigurace bludiště[18]. . . . .	20
4.8	Fotografie stolu s herní plochou bez rozmístěných překážek[18]. . . . .	21
4.9	Naměřené hodnoty intenzity světla. . . . .	23
4.10	NXT - Intenzita okolního světla nad LED. . . . .	23
4.11	NXT - Intenzita okolního světla mezi LED. . . . .	24
4.12	NXT - Intenzita odraženého světla nad LED. . . . .	24
4.13	NXT - Intenzita odraženého světla mezi LED. . . . .	25
4.14	EV3 - Intenzita okolního světla nad LED. . . . .	26
4.15	EV3 - Intenzita okolního světla mezi LED. . . . .	26
4.16	EV3 - Intenzita odraženého světla nad LED. . . . .	27
4.17	EV3 - Intenzita odraženého světla mezi LED. . . . .	27
4.18	Hotový robot pro druhou úlohu. . . . .	28
4.19	Diagram programu robota pro druhou úlohu. . . . .	30

## Elektronická příloha

- Elektronická verze bakalářské práce ve formátu pdf
- Modely robotů pro LEGO Digital Designer včetně návodů
- Zdrojové kódy použitých programů
- Video vyřešených soutěžních úloh
- Webové stránky ve formátu php