

Univerzita Pardubice

**Fakulta ekonomicko-správní
Ústav systémového inženýrství a informatiky**

**Tvorba úloh pro předmět Základy algoritmizace
robotické úlohy**

Matej Polák

**Bakalářská práce
2013**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Matej Polák**
Osobní číslo: **E10566**
Studijní program: **B6209 Systémové inženýrství a informatika**
Studijní obor: **Informatika ve veřejné správě**
Název tématu: **Tvorba úloh pro předmět Základy algoritmizace - robotické úlohy**
Zadávací katedra: **Ústav systémového inženýrství a informatiky**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je tvorba úloh pro roboty, které budou využitelné v předmětu Základy algoritmizace.

- 1) Stavebnice Lego Mindstorm
- 2) Složení robotických stavebnic
- 3 Tvorba algoritmů pro roboty
- 4) Vytváření úloh pro předmět Základy algoritmizace

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

TAUFER I. Algoritmy a algoritmizace - vývojové diagramy. Pardubice:

Univerzita Pardubice, 2009. ISBN 978-80-7395-182-5.

PŠENČÍKOVÁ, J. Algoritmizace. 1. vyd. Kralice na Hané: Computer Media,
2007. ISBN 978-80-7402-034-6.

KNUTH, D. E., ERVIN, D. Umění programování I. Brno: Computer Press,
2008. ISBN 9788025120255.

Vedoucí bakalářské práce:


Ing. Jan Panuš, Ph.D.


Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce: **1. října 2012**

Termín odevzdání bakalářské práce: **30. dubna 2013**


doc. Ing. Renáta Myšková, Ph.D.
děkanka

L.S.


prof. Ing. Jan Šapek, CSc.
vedoucí ústavu

V Pardubicích dne 1. října 2012

PROHLÁŠENÍ

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 30. 4. 2013

Matej Polák

PODĚKOVÁNÍ:

Tímto bych rád poděkoval svému vedoucímu práce Ing. Janu Panušovi, Ph.D., za jeho odbornou pomoc, cenné rady a poskytnuté materiály, které mi pomohly při zpracování této práce.

ANOTACE

Tato práce bude sloužit studentům pro pochopení teorie, základů a činností algoritmů v předmětu Základy algoritmizace a měla by pomoci úspěšně poradit s naučením programů Enchanting a LEGO Mindstorms. Tištěná práce je doplněna o CD s kódy příkladů, které jsou zde zmíněny.

KLÍČOVÁ SLOVA

Lego Mindstorms, NXT-G, Enchanting, senzor Hitechnic, robot

TITLE

Creating excercises for the subject Basics of Algorithms - tasks for robots

ANNOTATION

This work will serve students for understanding of theory and activites of algorithm in subject Introduction to Algorithms and should help to successfully advise with learning of programs Enchanting and LEGO Mindstorms. Printed version is completed with the CD with programming codes which are here mentioned.

KEYWORDS

Lego Mindstorms, NXT-G, Enchanting, sensor Hitechnic, robot

OBSAH

ÚVOD.....	8
1 ROBOTICKÉ STAVEBNICE.....	9
1.1 PŘÍKLADY STAVEBNIC	9
1.1.1 Merkur	10
1.1.2 Fischertechnik.....	11
1.1.3 Robotis Bioloid.....	12
1.1.4 Lego Mindstorms.....	12
1.1.5 HiTechnic senzory.....	13
1.2 PROGRAMOVACÍ PROSTŘEDÍ	15
2 LEGO MINDSTORMS.....	17
2.1 PRVKY PROGRAMU	17
2.2 VÝBĚR POUŽITELNÝCH BLOKŮ	18
2.3 NAHRÁNÍ KÓDU DO ROBOTA	23
2.4 VYBRANÉ PŘÍKLADY V LEGO MINDSTORMS	24
2.4.1 Modulo.....	24
2.4.2 Strážce chodby (ultrazvukové senzory)	25
2.4.3 Najdi jih.....	26
2.4.4 Orientace v prostoru.....	27
2.4.5 Hlídaní prostoru.....	29
3 ENCHANTING	31
3.1 PRVKY PROGRAMU	32
3.2 PALETA BLOKŮ	33
3.3 NAHRÁNÍ FIRMWARU A KÓDU DO NXT JEDNOTKY	35
3.4 UKÁZKOVÉ PŘÍKLADY	36
3.4.1 Orientace v prostoru – 2x ultrazvukový senzor	36
3.4.2 Barevný senzor rozlišení barev	38
3.4.3 Barevný senzor - semafor	38
3.4.4 Vytlačení objektu z prostoru	40
3.4.5 Kompas – měření azimutu	41
3.4.6 Otočení robota směrem na sever.....	42
3.4.7 Kompas – měření rovnoběžnosti přímek	43
3.4.8 Hodnoty z gyroskopu	44
ZÁVĚR	46
POUŽITÁ LITERATURA.....	48

SEZNAM OBRÁZKŮ

Obrázek 1: Pásový podvozek Merkur s elektronikou a řídicí deskou	10
Obrázek 2: Stavebnice Fischertechnik robotická ruka	11
Obrázek 3: Ukázka kódu, který rozsvítí diodu po dobu 3 vteřin.....	12
Obrázek 4: Stavebnice Lego Mindstorms NXT 2.0 se senzory	12
Obrázek 5: Rozdělení os na senzoru zrychlení.....	14
Obrázek 6: Úvodní obrazovka LEGO MINDSTORMS NXT	18
Obrázek 7: Robot jede rovně, dokud nenajede na modrou barvu	20
Obrázek 8: Výpočet funkce MODULO v LEGO Mindstorms.....	25
Obrázek 9: Strážce chodby – využití ultrazvukových senzorů v LEGO MINDSTORMS	26
Obrázek 10: Najdi jih – kód pro využití kompasu v LEGO MINDSTORMS	27
Obrázek 11: Orientace v prostoru – ultrazvukový senzor první switch true	28
Obrázek 12: Orientace v prostoru – ultrazvukový senzor první druhý switch true.....	28
Obrázek 13: Orientace v prostoru – ultrazvukový senzor druhý switch false.....	28
Obrázek 14: Kód pro hlídání prostoru – více senzorů v LEGO MINDSTORMS	30
Obrázek 15: Rozložení prvků programu Enchanting	33
Obrázek 16: Kód pro orientaci v prostoru s využitím dvou ultrazvukových senzorů	37
Obrázek 17: Kód pro rozlišování barev pomocí barevného senzoru v Enchantingu	38
Obrázek 18: Kód pro jízdu po čáře a využití barevného senzoru pro semafor	40
Obrázek 19: Kód pro vytlačení objektu z prostoru v Enchantingu	41
Obrázek 20: Měření azimutu pomocí HiTechnic Kompas senzoru	42
Obrázek 21: Kód pro otočení robota směrem na sever v Enchantingu	43
Obrázek 22: Využití kompasového senzoru pro měření úhlů	44
Obrázek 23: Získávání hodnot z gyroskopu	45

ÚVOD

Cílem této bakalářské práce bylo navrhnout příklady programů pro roboty v LEGO Mindstorms NXT a Enchanting pro předmět Základy algoritmizace. Tyto programy, jejich kombinace nebo adaptace mohou být využity při výuce daného předmětu a při robotických ukázkách.

V první kapitole této práce jsou popsány vybrané robotické stavebnice, které jsou dostupné na trhu, včetně robotů LEGO Mindstorms, které jsou na půdě Fakulty Ekonomicko-správní Univerzity Pardubice. Dále jsou popsány doplňkové senzory od firmy HiTechnic, které se také nachází na akademické půdě a možné využitelné programovací jazyky.

Druhá kapitola je věnována programovacímu prostředí dodávaného se stavebnicí s názvem LEGO Mindstorms, známého také jako NXT-G. Je zde obsažen základní popis zmíněného programu a také několik ukázkových úloh, které jsem vymyslel.

Třetí kapitola se věnuje programovacímu prostředí Enchanting, který je nadstavbou programu Scratch. Je zde základní popis programu, jeho bloků a funkcí pro sestavování kódů. Následují vymyšlené ukázkové úlohy, z nichž většina využívá doplňkové senzory HiTechnic, které jsou v programu podporovány.

V závěru této práce jsou tyto dva programy porovnány.

1 ROBOTICKÉ STAVEBNICE

Již od dávných dob si člověk ulehčoval práci či výrobu. Slovo robot vzniklo v myslí a díle Karla Čapka R. U. R., ale první roboti existovali již dříve. Nejdříve to byly jednoduché mechanické stroje, později vznikla pásová výroba a zlomem bylo využívání automatických strojů, které jsou schopné konat zadanou práci. Ke strojům a jejich součástkám se v době rozmachu počítačů začali přidávat řídicí jednotky, integrované obvody s naprogramovanými kódy, které umožňují vykonat zadanou činnost. S rozvojem umělé inteligence jsou stroje schopné částečně myslet podle zadaných parametrů. Jistou definicí robota je, že roboti jsou stroje pracující samostatně nebo s určitou interakcí s okolím nebo uživatelem. Jejich různorodost dala vzniknout několika novým odvětvím a vědním disciplínám. Příkladem může být nanorobotika, která se zabývá vytvořením robotů o velikosti v řádech nanometrů. Dále také mikrorobotika, která má za kritérium limit menší než 1 milimetr. Bionika, která pro vyvíjení robotů a jejich vylepšení kopíruje a využívá věci z přírody a její evoluce. Swarm robotika zabývající se koordinací a kolektivním chováním mnoha robotů [5].

V dnešní době se roboti využívají v průmyslové výrobě, vojenství, průzkumech, lékařství, domácnostech, dopravě a dalších odvětvích. Robotické stavebnice jsou vytvořeny nejen pro zábavu, ale mohou sloužit i jako výukové pomůcky na školách a univerzitách. Dá se s nimi vytvořit robot, podle návodu nebo vlastních nápadů a pochopit základní mechanické funkce jako jsou otáčení, pohyb, chytání a další funkce s dostupnými vstupními senzory a výstupními zařízeními.

1.1 Příklady stavebnic

Konkurenční boj a vývoj v oblasti výpočetních technologií umožnil vytvoření mnoho typů robotických stavebnic. Některé z těchto stavebnic jsou odvozeny od stavebnic bez programovatelných jednotek a mechanických prvků (Lego, Merkur).

Na trhu je dostupných mnoho robotických stavebnic, rozšiřovacích setů a speciálních senzorů.

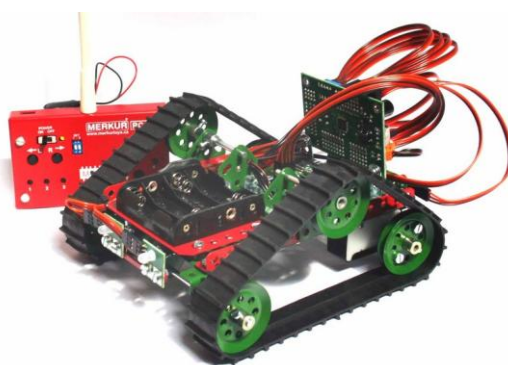
- stavebnice Lego Mindstorms,
- stavebnice Robotis Bioloid,
- stavebnice Fischertechnik,
- stavebnice Merkur,
- senzory HiTechnic.

1.1.1 Merkur

Tato tradiční česká firma vyrábí kovové stavebnice jak pro zábavu, tak i pro výuku fyziky, robotiky a mechatroniky.

Firma nabízí robotické stavebnice v kategoriích:

- kolové roboty – autíčka, kolová vozítka,
- kráčející roboty – brouci, pavouci,
- pásové roboty – pásová vozítka,
- ostatní roboti – robotické ruky, plotrový zapisovač, a další [16].



Obrázek 1: Pásový podvozek Merkur s elektronikou a řídicí deskou

Zdroj: [16]

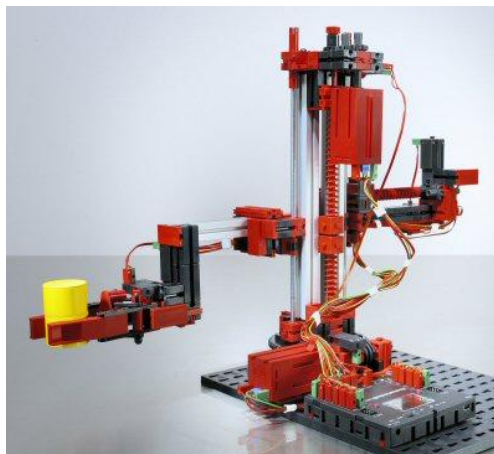
K daným setům se mohou dokoupit jednotlivé součástky, ať již jednotlivé dílky, tak i také servomotory různých typů a velikostí, elektrické moduly a senzory přímo určené pro robotiku. Jedná se o LED diody různých typů, rádiové přijímače a vysílače, žárovky, kabely, senzory na měření teploty, vzdáleností atd. Modely jsou taky vybaveny řídicí deskou typu Alfa ATMEL s 32 bitovým procesorem nebo řídicí deskou Alfa PICAXE. Lze také využít i jiné řídicí desky, například desky typu Arduino, nebo je možné si s patřičnou technickou zkušeností a znalostí vytvořit vlastní desku [16].

Většina robotických stavebnic je již předurčena pro daný úkol. Robot na řídicí desce má již uložen program, který vykonává. Další možností je upravit si robota podle svých možností za pomoci dalších součástek a přeprogramováním řídicí desky [16].

Jako programovací jazyk může být použit výrobcem určený programovací software, například PICAXE Programming Editor, nebo jiný objektově orientovaný jazyk jako C, BASIC, Visual Basic, a jiné [1], [2], [20].

1.1.2 Fischertechnik

Fischertechnik je robotická stavebnice vynalezená v Německu. Vyrábí se ve firmě fischertechnik GmbH. Firma vyrábí stavebnice pro děti, ale i robotické stavebnice [9].

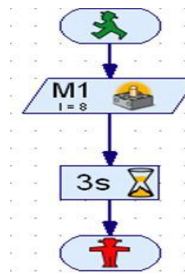


Obrázek 2: Stavebnice Fischertechnik robotická ruka

Zdroj:[9]

Stavebnice obsahují podle dané varianty výrobku řídicí jednotku s názvem ROBO TX Controller, dále určité vstupy a výstupy, jednotlivé stavební dílky, kabely a software ROBO Pro. Řídicí jednotka obsahuje 32 bitový procesor, 8 MB paměti RAM, Bluetooth zařízení, 8 vstupů, 4 výstupy, konektor pro USB 2.0 a baterie zajišťující stejnosměrnou elektrickou energii. Jako vstupy mohou být použity dotykové senzory, fototranzistory, ultrazvukový senzor, barevný senzor, teplotní senzor a další senzory dostupné pro rozšíření původní stavebnice. Výstupními prvky mohou být motory, světelné diody, zvuková zařízení, display řídicí jednotky a další [9].

Software s názvem ROBO Pro je dodáván společně se stavebnicí a obsahuje programovací prostředí s příkazy, které naprogramují daného robota, aby vykonával zadanou činnost. Dále lze využívat i objektově orientovaný jazyk C nebo C++. V programu ROBO Pro se využívá grafické programování pomocí vývojových diagramů, obsahujících různé příkazy, které ovlivňují danou činnost. Software obsahuje několik úrovní složitosti od začátečníků, až po profesionály [9]. Dané vývojové diagramy začínají a končí vždy určitými bloky, v tomto případě zeleným a červeným panáčkem (viz obrázek 2). Mezi nimi jsou ve směru postupu procesu kroky, které rozsvítí diodu na výstupu 8, a další krok ji nechá rozsvícenou 3 vteřiny, než se celý program vypne.



Obrázek 3: Ukázka kódu, který rozsvítí diodu po dobu 3 vteřin

Zdroj:[vlastní]

1.1.3 Robotis Bioloid

Robotis Bioloid je Jihokorejská firma vytvářející tyto roboty. Na trh je dodávána stavebnice v několika verzích. Ke stavebnicím je dodávána velká škála dodatečných součástí, jako gyroskopy, vzdálenostní senzory, IR senzory, servomotory různých typů, dálkové ovládání a jiné. Robot používá řídicí jednotku s názvem CM-530. Naprogramovat robota lze v programovacím jazyku C. Lze také využít programovací prostředí RoboPlus nebo jiné freewarové aplikace například Motion Editor nebo Behaviour Control [6], [21].

1.1.4 Lego Mindstorms

Lego Mindstorms NXT je robotická stavebnice od firmy LEGO. Firma již vydala několik verzí této stavebnice, z nichž využíváme verzi NXT 2.0, která byla vydána na trh v roce 2009 [26].



Obrázek 4: Stavebnice Lego Mindstorms NXT 2.0 se senzory

Zdroj:[14]

Dodávaná stavebnice obsahuje řídicí jednotku, jednotlivé dílky na stavbu, kabely, USB 2.0 kabel na propojení s počítačem, baterie, LED diody, 3 servomotory a základní senzory. Stavebnice obsahuje základní senzory, které jsou s ní dodávány:

- Dotykový senzor (Touch Sensor) – senzor funguje na principu pravda/nepravda (true/false), kdy ve stisknutém stavu vrací hodnotu 1 a v nestisknutém stavu hodnotu 0.
- Ultrazvukový senzor (Ultrasonic Sensor) – umožňuje měření vzdálenosti od předmětu před senzorem v rozsahu 0 až 255 centimetrů s přesností ± 3 centimetry.
- Zvukový senzor (Sound Sensor) – měří intenzitu zvuku v decibelech a vrací hodnoty mezi 0-100%.
- Světelný senzor (Light Sensor) – senzor detekuje intenzitu světla. Dokáže rozlišit světlo a tmu, ale i také intenzitu odraženého světla, který sám senzor pomocí LED diody vydává, od blízkého povrchu. Vrací hodnoty mezi 0-100 dle intenzity světla.

Mimo tyto senzory se mohou dokoupit další senzory ať již od firmy LEGO, nebo jiné firmy, která s ní spolupracuje a vyrábí senzory přímo pro LEGO Mindstorms NXT, jako je firma HiTechnic nebo Vernier [23], [13].

Řídicí jednotka obsahuje 32 bitový mikroprocesor, černobílý display, jeden konektor USB 2.0, bezdrátové zařízení Bluetooth, 3 výstupní porty pro servomotory, 4 vstupní porty pro senzory, místo pro baterie a konektor k jejich nabíjení [14].

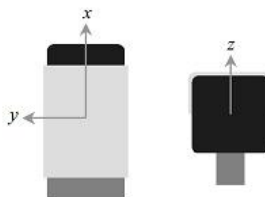
1.1.5 HiTechnic senzory

Firma HiTechnic se sídlem v USA, vyrábí širokou škálu robotických senzorů a příslušenství pro LEGO MINDSTORM NXT. Senzory s příslušenstvím tak doplňují základní senzory a dílky dodávané přímo s robotickou stavebnicí. Jednotlivé senzory mohou fungovat ve více režimech, podle nastavení v použitém programu a mohou tak vracet různé hodnoty podle použitého programovacího prostředí. Vybrané senzory, se kterými jsem se setkal na půdě Fakulty ekonomicko-správní [13]:

1.1.5.1. NXT Acceleration sensor (senzor měření zrychlení)

Senzor měření zrychlení měří pohyb z klidového stavu na třech osách. Osy jsou na senzoru rozděleny podle kartézského systému na osy x, y a z, jak je možno vidět na obrázku (viz

obrázek 5). Senzor vrací hodnoty v rozmezí od -2g do +2g a stupnice mezi jednotlivými g má 200 hodnot. Senzor lze také použít pro měření náklonu ve třech osách [13].



Obrázek 5: Rozdělení os na senzoru zrychlení

Zdroj:[13]

1.1.5.2. NXT EOPD

EOPD je zkratka z Electro Optical Proximity Detector, což v překladu znamená Elektrický optický detektor vzdálenosti. Senzor měří vzdálenost od objektů a je schopen detekovat i malé změny. Senzor má vlastní světelný zdroj a s jeho pomocí měří vzdálenost. Jeho základní vlastností je odfiltrování okolního rušivého světla, které by mohlo výpočet vzdálenosti ovlivnit. Senzor lze použít na měření krátké vzdálenosti (do 20 cm) s velkou přesností, nebo na delší vzdálenosti, ale s rostoucí odchylkou. Odchytky také mohou rovněž vznikat v záležitosti na velikosti, tvaru a světlosti měřeného objektu [13].

1.1.5.3. NXT Color sensor (barevný senzor)

Senzor, který je schopen rozlišit barvy. Škála barev, které je senzor schopný rozeznat, se pohybuje okolo 17 barev. Každá barva má na škále své vlastní číslo, které se vrací jako naměřená hodnota do řídicí jednotky, kdy hodnotu 0 má černá barva a hodnotu 17 má bílá barva. Senzor může také operovat v RGB módu, kde senzor odesílá do řídicí jednotky přímo naměřené hodnoty. Senzor obsahuje jednu bílou LED diodu na osvětlení povrchu objektu, takže se senzor může využít i za zhoršených světelných podmínek [13].

1.1.5.4. NXT Compass Sensor (kompas)

Senzor obsahuje digitální kompas, který měří magnetické pole země a vrací do řídicí jednotky naměřenou hodnotu směru, kam je senzor nasměrován. Hodnoty jsou měřeny v rozsahu od 0 do 359. Hodnota 0 znamená, že kompas je nasměrován na sever, hodnota 90 na východ, hodnota 180 na jih a hodnota 270 na západ. Senzor musí být při použití

v horizontální poloze, vzdálený od servomotorů a samotné řídicí jednotky, kvůli ovlivnění magnetického pole a tím měřených hodnot [13].

1.1.5.5. NXT Gyro sensor (gyroskopický senzor)

Senzor měří rychlost otáčení na jedné ose podle svého obvodu. Senzor musí být ve vertikální poloze. Při použití tohoto senzoru se do řídicí jednotky vrací hodnota otáčení senzoru ve stupních za sekundu. Maximální hodnota může být +/- 360 stupňů za sekundu [13].

1.1.5.6. Další senzory

- NXT Barometric Sensor (Barometrický senzor)
- NXT Angle Sensor (Úhlový senzor)
- NXT Force Sensor (Tlakový senzor)
- NXT IRReceiver Sensor (IR přijímač)
- NXT IRSeeker (IR vyhledávač)

1.2 Programovací prostředí

Na půdě Fakulty ekonomicko-správní se nachází robotické stavebnice LEGO MINDSTORMS NXT 2.0. Jako programovací jazyk může být použit se stavebnicí dodávaný program LEGO MINDSTORMS Software s názvem NXT-G, který je založen na grafickém programování programu LabVIEW. Je možné využít i jiné programovací jazyky a prostředí. Některé tyto programy jsou vytvořeny přímo pro LEGO NXT například [11] :

- brickOS – vytvořen na základě jazyka C/C++ a spíše pro starší verze NXT [8],
- programovací jazyk C v programu Not eXactly C [12],
- Enchanting – grafické programování pro jazyk leJOS NXJ [7],
- Lego.NET – alternativní operační systém využívající .NET [19],
- leJOS NXJ – Java pro stavebnice Lego Mindstorms [3], [4],
- NXT_Python – aplikace pro kontrolu robotů využívající programovací jazyk Python [10],

- Microsoft Robotics Developer Studio – využití jazyků .NET, Visual Basic, C# [17], [18],
- NXTGCC – nástroj pro vytváření kódu v jazyku C a C++ [25],
- MATLAB and Simulink – programovací jazyk a prostředí pro počítání, získávání dat, analýzu dat, modelování a simulování dynamických systémů. Mohou se zde vytvářet algoritmy, které se převedou do jazyka pro NXT roboty [15].

Programovací prostředí LEGO MINDSTORMS Software NXT-G a Enchanting jsou podrobně popsány v kapitolách níže v této práci. Tyto programy jsem vybral, protože se při jejich využití není potřeba učit programovací jazyky.

2 LEGO MINDSTORMS

LEGO MINDSTORMS je programovací prostředí pro psaní programů pro NXT roboty. Obsahuje všechny prvky pro tvoření kódu pro roboty. Obtížnost tohoto programu je klasifikována jak pro začátečníky, tak i pro středně pokročilé [14]. Program nenabízí při instalaci češtinu, takže ho využívám pouze v anglickém jazyce. NXT-G je programovací jazyk používaný tímto programem. V tomto jazyce se každá individuální část nazývá blok. Každý blok může obsahovat různé části kódu, a když se bloky spojí, ovládají konečné chování robota.

2.1 Prvky programu

Úvodní obrazovka programu a poté program samostatný se dá rozdělit na několik částí (viz obrázek 6). V části „A“ se nachází hlavní lišta s nabídkou, kde se dá zvolit volba založit nový, nebo otevřít program. Také ukončení samotného programovacího prostředí a jeho registrace u výrobce. Tvoří se zde nové bloky, definují proměnné, kalibrují senzory. Dále se zde nachází editory obrázků a zvuků, nahrání kódu do robota a nápověda.

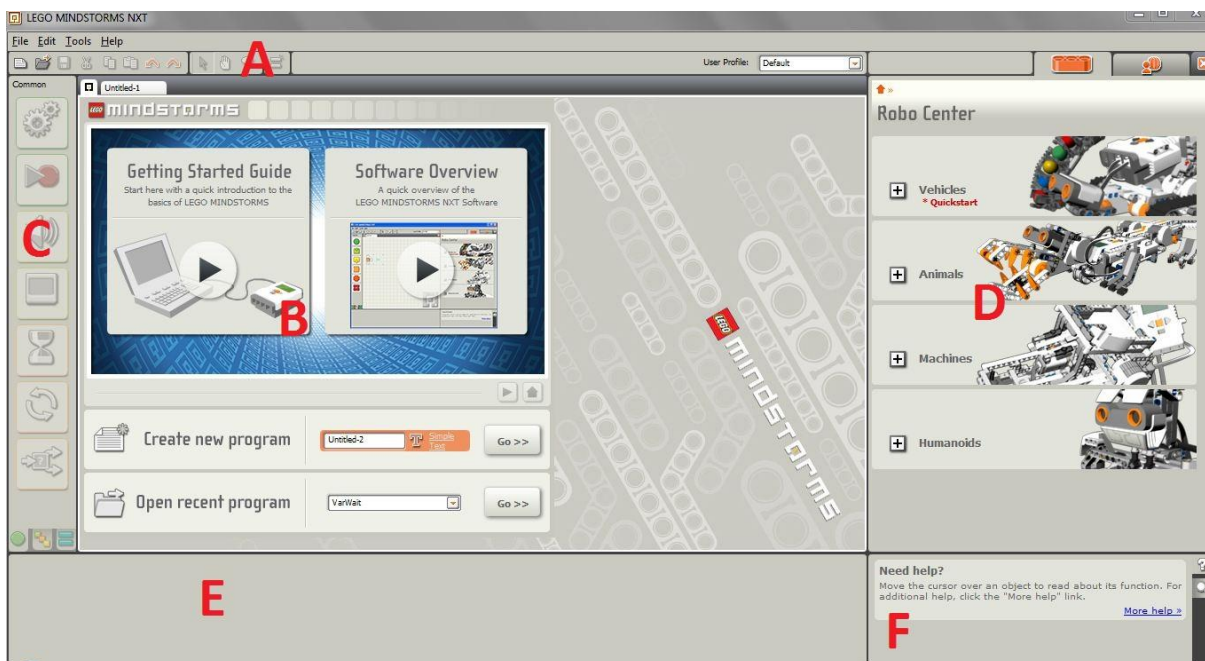
Uprostřed ve středu obrazovky je část „B“, kde je pracovní plocha na programování kódů. Přímo nad ní se nacházejí záložky, které umožňují přepínání mezi více otevřenými kódy.

Nalevo je část „C“ s jednotlivými bloky pro pohyb, nahrávání a přehrávání zvuků, zobrazování na displeji, cykly, podmínky. Dále pak také bloky pro výstupní prvky robota, kterými jsou motory, displej, zvuk, výstupní signál Bluetooth a ovládání barvy vydávaným světelným senzorem. Vstupní prvky jsou zastoupeny mnoha senzory, vstupní částí ovládání Bluetooth, samotnými tlačítky na kostce NXT a časovači. Matematické prvky jsou zastoupeny bloky s názvy: logické prvky, porovnávací prvky, matematická znaménka, rozsah, náhodné číslo, proměnné a konstanty. Pokročilé prvky obsahují kalibraci senzorů, převod datových prvků mezi typy „text“, „logic“ a „number“, restartováním motoru a nastavením přímého Bluetooth ovládání.

Část „D“ s názvem Robo Center je rozdělena na 2 části. Jedna část obsahuje odkazy na web LEGO MINDSTORMS, kde je fórum pro uživatele, ukázky různých projektů a věci ke stažení. Druhá část se člení na vozidla (vehicles), zvířata (animals), stroje (machines) a humanoidy (humanoids). Každá tato část obsahuje ukázkový projekt stavebnice s návodem na stavbu včetně postupu jak vytvořit kód pro daného robota.

V dolní části programu je část „E“, která se nazývá konfigurační panel. Tato část se využívá k nastavování parametrů v jednotlivých blocích.

Ve spodní části vpravo je část „F“, tato část zobrazuje stručnou nápovědu ke každému bloku. Po kliknutí na „More Help“ se zobrazí internetový prohlížeč s rozsáhlou nápovědou pro celý program a všech jeho možností co obsahuje. Zde je také umístěn v této části navigační panel, který zobrazuje aktuální pozici pracovní plochy v programovacím kódu. Tato funkce pomáhá uživateli orientovat se v rozsáhlém kódu.



Obrázek 6: Úvodní obrazovka LEGO MINDSTORMS NXT

Zdroj: [Vlastní]

2.2 Výběr použitelných bloků

Část „C“, která obsahuje jednotlivé programovací bloky, je rozdělena na 3 části. Výběr běžných bloků (Common), kompletní výběr (Complete) a uživatelský výběr (Custom).

Výběr běžných bloků obsahuje bloky pro ovládání motorů (Move), nahrávání a přehrávání (Record/Play), zvuk (Sound), zobrazení na displeji (Display), podmínkový blok (Wait), cykly (Loop) a přepínač (Switch).

Uživatelská část obsahuje bloky vytvořené uživatelem pro jeho osobní projekty a nové bloky stažené z internetu. Tato část se spravuje pomocí možnosti Manage Custom Palette, po jejímž zvolení se zobrazí složka, kam lze nahrát soubory typu .rbt, které zastupují jednotlivé bloky. Bloky lze také do programu importovat tlačítkem Block Import and Export

Wizard v nástrojové liště. Při použití této metody je možné bloky přidávat do libovolného výběru.

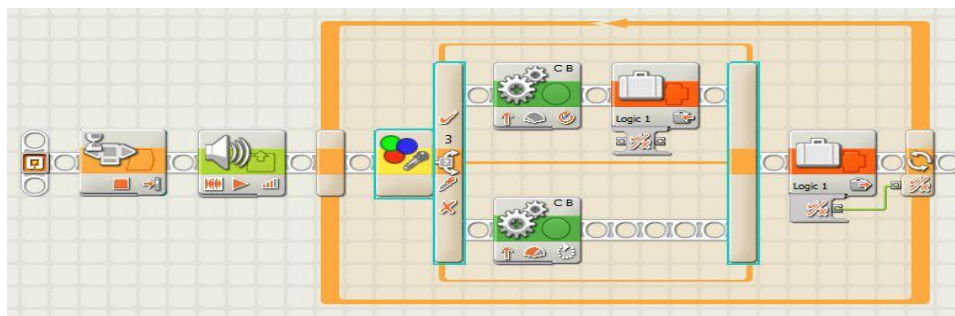
Kompletní výběr obsahuje, všechny bloky, které program obsahuje. Tento výběr se dělí na běžné (Common), které mají vlastní výběr, akční (Action), sensorové (Sensor), bloky procesu (Flow), datové bloky (Data) a pokročilé bloky (Advanced).

Běžné bloky (Common)

- Move – tento blok ovládá pohyb servomotorů. Lze použít i pro více servomotorů najednou. Specifikuje se zde zapojení servomotorů v konektorech, směr a rychlost jejich otáčení. Také se zde nastavuje doba otáčení servomotorů. Doba lze nastavit na nekonečně, určitý počet otáček, vteřiny akce nebo počet úhlů, o který se mají servomotory otočit.
- Record/Play – blok se dělí na dvě části. První část snímá a uloží pohyb z vybraných servomotorů po zadaný čas. Druhá část spustí akci, která tento pohyb zopakuje.
- Sound – tento blok ovládá vydávaný zvuk. Může vydávat určité tóny nebo zvukové soubory. Zvukové soubory mohou být vybrány z přednastavených zvuků nebo se může nahrát vlastní zvukový soubor. Nastavuje se zde také hlasitost, a zda se má zvuk opakovat nebo přehrát pouze jednou.
- Display – blok umožňuje zobrazení na displeji. Může se nastavit umístění a zobrazení textu, obrázku, jednotlivých bodů, linií a kruhů, nebo se může displej celkově resetovat.
- Wait – blok, který zastupuje podmínku IF. Když nastane určitá situace, je kód za touto podmínkou vykonán. Nastavení je zde pro všechny senzory, tlačítka a časovač.
- Loop – blok, který zastupuje cyklus. Nastavuje se zde počet opakování, nebo podmínka, podle které se cyklus spustí nebo zastaví.
- Switch – blok, který zastupuje podmínku IF THEN ELSE. Při splnění určité podmínky se provede určitá část kódu a při nesplnění podmínky se vykoná jiná část kódu. Podmínku lze nastavit na hodnoty logické, numerické nebo textové a také lze počkat na akci určitého senzoru nebo tlačítka. Při výběru numerických nebo textových typů lze udělat více podprogramů, které zastupují několik podmínek typu IF za sebou.

Využití několika bloků (viz obrázek 7), kde na začátku je podmínka pro stisknutí tlačítka Enter. Po stisknutí tohoto tlačítka je přehrána zvuková stopa a následuje cyklus. Cyklus

obsahuje podmínku Switch, která je nastavena na barevný senzor. Podmínka je, že senzor uvidí modrou barvu, při nesplnění se servomotory otáčí dopředu. Při splnění podmínky se servomotory zastaví a logická proměnná je nastavena na hodnotu True a po ukončení bloku Switch se celý cyklus zastaví.



Obrázek 7: Robot jede rovně, dokud nenajede na modrou barvu

Zdroj: [vlastní]

Akční bloky (Action)

Výběr bloků, které zastupují výstupní prvky.

- Motor – ovládání jednoho servomotoru. Specifikuje se zde konektor, směr, rychlost a doba otáčení.
- Sound – zvuky vydávané NXT jednotkou.
- Display – zobrazení obrázku, textu nebo kresby na displeji.
- Send Message – zaslání zprávy pomocí Bluetooth z NXT jednotky.
- Color Lamp – blok, který umožňuje měnit barvu světla vydávaným LED diodou na světelném senzoru. Lze měnit barvy mezi červenou, zelenou a modrou.

Senzorové bloky (Sensor)

Bloky, které zastupují vstupní prvky. Jsou zde obsaženy všechny senzory dodávané ke stavebnici, časovač a tlačítka na NXT jednotce. Po stažení určitých aktualizčních souborů jsou zde i bloky pro dodatečné senzory, například HiTechnic senzory [13].

- Touch sensor – blok zaznamenávající dotyk. Rozlišují se zde 3 stavy: stisknutí, uvolnění a ťuknutí (stisknutí a uvolnění).
- Sound sensor – zaznamenávání hladiny zvuku.
- Light sensor – měření intenzity světla. Povoluje se zde také spuštění LED diody na samotném senzoru.

- Ultrasonic sensor – blok zaznamenávající vzdálenost v jednotkách (centimetry, palce).
- NXT Buttons – blok, který snímá tlačítka na samotné NXT jednotce. Jde o tlačítka Enter, Left a Right. Princip funguje stejně jako na dotykovém senzoru.
- Timer – blok zaznamenávající čas od spuštění samotného programu. Je zde možnost využití 3 časovačů, jejich čtení nebo restartování.
- Receive Message – zajišťuje obdržení zprávy Bluetooth NXT jednotkou.
- Color Sensor – zjišťování barvy pomocí barevného senzoru.

Flow

Výběr Flow zahrnuje bloky, které ovlivňují tok programového kódu. Jsou zde bloky Loop, Stop, Wait a Switch, které jsou popsány v sekci Common.

Data

- Logic – blok provádějící logické operace z několika vstupů. Možné jsou operace OR, AND, NOT a XOR. Výstupem je vždy odpověď True nebo False.
- Math – ze vstupních hodnot se provádějí matematické operace, jakými jsou: sčítání, odčítání, dělení, násobení, absolutní hodnota a druhá odmocnina. Výstupem je hodnota po provedených matematických operacích.
- Compare – blok porovnává ze vstupů hodnoty a vrací True nebo False. Možnosti porovnávání jsou větší (>), menší (<) a rovná se (=).
- Range – vstupní hodnotou je číslo, které se porovnává se zadaným rozmezím, zda tam hodnota patří, nebo ne. Výstupem je hodnota True nebo False. Minimální a maximální zadané hodnoty rozmezí jsou – 2 147 483 648 až 2 147 483 647.
- Random – blok, který generuje náhodnou hodnotu v rozmezí od 0 do 32 767. Náhodná hodnota se dá využít při generování nepředvídatelného chování robota.
- Variable – zde se používají proměnné v daném programovém kódu. Proměnné se nastavují na hlavní liště programu v záložce Edit a Define Variables. Proměnné mohou být typu číselného, textového nebo logického.
- Constant – blok, který používá konstanty v programovém kódu. Konstanty se vytváří podobně jako proměnné, s tím rozdílem, že se klikne na Define Constant. Konstanty mohou být také typu logického, číselného nebo textového.

Advanced

Výběr Advanced obsahuje pokročilejší prvky pro programování.

- Number to text – vstupem do tohoto bloku je číslo, například hodnota získaná ze senzoru, a tato hodnota je změněna na text, který se může zobrazit na displeji.
- Text – blok, jehož vstupem jsou jakékoliv textové hodnoty, znaky i čísla. Tyto vstupy jsou spojeny dohromady do jedné věty, která je poté samotným výstupem z bloku a může být zobrazena na displeji.
- Keep Alive – tento blok zabraňuje NXT jednotce vstupu do režimu spánku. Když NXT jednotka je delší dobu bez vstupů, nebo se čeká na vykonání podmínky delší dobu, tak jednotka přejde do režimu spánku. Aby se zabránilo vstupu do spánku, tak se využívá tento blok, ve kterém se nastavuje čas v milisekundách, než se přejde do spánku.
- File Access – s využitím tohoto bloku se používají soubory uložené v NXT jednotce. Je zde možnost zápisu do souboru, čtení ze souboru, jeho zavření nebo smazání.
- Calibrate – blok, který kalibruje světelný senzor nebo zvukový senzor. Nastavuje se zde, v jakém portu se senzor nachází a zda chceme kalibrovat maximální, nebo minimální naměřenou hodnotu senzoru.
- Reset Motor – vypínání funkce, která měří o kolik se přesně servomotory otočí.
- Bluetooth – blok s funkcemi, které zapnou Bluetooth spojení, jeho vypnutí, inicializování spojení a spárování NXT jednotky s jiným zařízením, které také používá Bluetooth spojení. Jednotka se může najednou spojit až se třemi jinými zařízeními.

Image editor (Editor obrázků)

Program obsahuje jednoduchý editor na tvorbu či úpravu obrázků, které se mohou uložit do NXT jednotky a zobrazit na jejím displeji. Editor pracuje s plochou o velikosti 100x64 pixelů, které se zobrazují na NXT jednotce. Obrázky si lze vytvořit vlastní, nebo je možno si upravit již předpřipravené obrázky, které jsou uloženy již od instalace programu. Na tvorbu obrázků je možno použít tyto nástroje: tužku, přímka, čtyřúhelník, kyblík, elipsy, mazání, výběr a vložení textu. Jelikož NXT jednotka pracuje jen s jednobarevnými pixely, tak i editor pracuje jen se 2 barvami (černá, bílá).

Remote control (Vzdálená kontrola)

Vzdálená kontrola umožňuje ovládat NXT jednotku přímo z počítače. Spojení mezi počítačem a jednotkou je zajištěno USB kabelem nebo přes Bluetooth. Jsou zde tlačítka pro ovládání pohybů pomocí servomotorů, ovládání rychlosti, provedení akce a samostatné konfigurace servomotorů. Konfiguraci servomotorů je třeba provést před spuštěním programu a prováděním akce, aby počítač věděl, v kterých portech jsou servomotory zodpovědné za pohyb a jakým směrem se mají otáčet. Také se zde specifikuje akce třetího motoru, který není zodpovědný za pohyb robota. Touto akcí je myšleno jen otáčení zadaným směrem a rychlostí.

Sound editor (Zvukový editor)

Program také obsahuje zvukový editor, který umožňuje přidávání vlastních zvuků do programového kódu a jeho následném přehrání na NXT jednotce. Mohou se využít jakékoliv zvukové soubory z počítače, nebo nahrát vlastní zvukový soubor pomocí mikrofону počítače. Editace zvukového souboru je zde, však zastoupena jedinou funkcí a to stříháním zvukové stopy na kratší části. Po uložení zvukové stopy se v bloku Sound zobrazí tento zvukový soubor, který lze následně vybrat a přehrát.

2.3 Nahrání kódu do robota

Po sestavení programového kódu je nutné tento kód nahrát do NXT jednotky. Na začátku je nutné zapojit řídicí jednotku k počítači přes kabel USB nebo přes připojení BlueTooth. Poté je nutné vědět, zda je v řídicí jednotce uložen nejnovější firmware NXT. Když ne, tak přes volbu Update NXT Firmware se otevře nové okno, kde se tato aktualizace provádí. V horní části okna je napsána poslední stažená verze firmwaru, s možností stažení nové verze z internetu. Po vybrání dané verze NXT firmwaru se klikne na tlačítko Download, které spustí sekvenci tří kroků, které aktualizují firmware řídicí jednotky. Tyto kroky jsou: nalezení řídicí jednotky, její příprava na nahrání a samotné nahrání.

Když je firmware řídicí jednotky aktuální a není třeba ho měnit, tak následuje samotné nahrání programu do NXT jednotky. V pravém dolním rohu na pracovní ploše se po spuštění projektu zobrazí ovládací panel. Tento panel obsahuje pět tlačítek, které umožňují interakci mezi řídicí jednotkou a počítačem, včetně nahrání kódu do jednotky. Těmito tlačítky jsou:

- NXT Window – tlačítko zobrazuje připojené řídicí jednotky a základní údaje o nich. Mezi tyto údaje patří identifikační jméno jednotky, stav baterie, typ připojení k počítači a informace o volné nebo zaplněné kapacitě paměti.
- Download and run – umožňuje nahrání programového kódu do jednotky. Po dokončení nahrávání je program v jednotce spuštěn.
- Download and run selected – tlačítko, slouží jen k nahrání vybrané části kódu, po jehož nahrání je tato část ihned spuštěna.
- Download – ukládá programový kód do jednotky bez spuštění.
- Stop – zastaví činnost prováděného kódu.

2.4 Vybrané příklady v LEGO Mindstorms

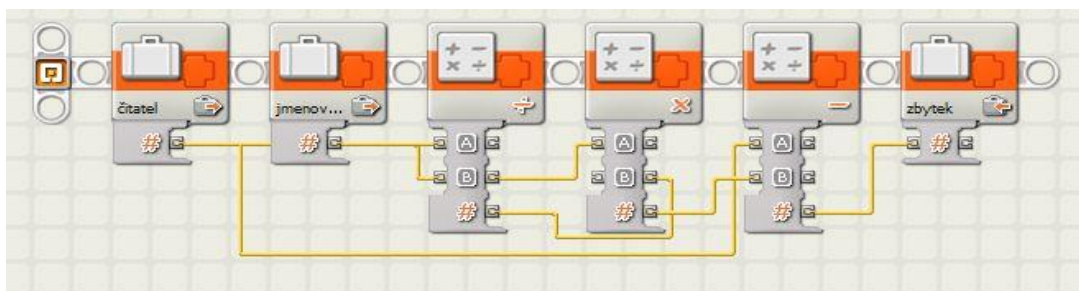
Tato kapitola je věnována ukázkovým příkladům vytvořeným v programu LEGO Mindstorms. Jsou to příklady jak vypočítat Modulo, Strážce chodby, Najdi jih, Orientace v prostoru a Hlídaní prostoru.

2.4.1 Modulo

Tato úloha řeší jak vypočítat zbytek po dělení, čili funkce modulo. Tato funkce není v matematických funkcích v programu zastoupena a je nutno si pro tuto funkci vytvořit vlastní kód (viz obrázek 8). Při tvoření této úlohy jsem využil z výběru Complete bloky typu Data. Přesněji jsem využil bloky Math a Variable.

Po spuštění programovacího prostředí LEGO Mindstorms je ihned na úvodní obrazovce zobrazena nabídka pro otevření nového programu. Po vytvoření programu je třeba vytvořit v „Define Variable“ proměnné, se kterými budeme pracovat a nastavit je na číselný typ. V tomto případě to jsou proměnné s názvy Čítatel, Jmenovatel a Zbytek. Do programu vložíme 2 bloky proměnných a nastavíme je na možnost čtení a vybereme proměnnou, kterou každý blok zastupuje. V tomto případě Čítatel a Jmenovatel. Dalším blokem co následuje je blok Math, který je nastaven na dělení. Následně spojíme čitatele se vstupem A na bloku Math, a jmenovatele s vstupem B na též bloku. Dalším krokem je vložení bloku Math, tentokrát s násobením. Vstupem do tohoto bloku na vstupu A je jmenovatel a na vstupu B do tohoto bloku je výstup z předchozího bloku, kterým jsme dělili. Poté je v programu blok Math nastavený na odečítání. Vstupem A je čítatel a vstupem B je výstup z předchozího

bloku. Výstup z bloku po odečítání je spojen se vstupem na posledním bloku proměnné Zbytek, který je nastaven na možnost zápisu.



Obrázek 8: Výpočet funkce MODULO v LEGO Mindstorms

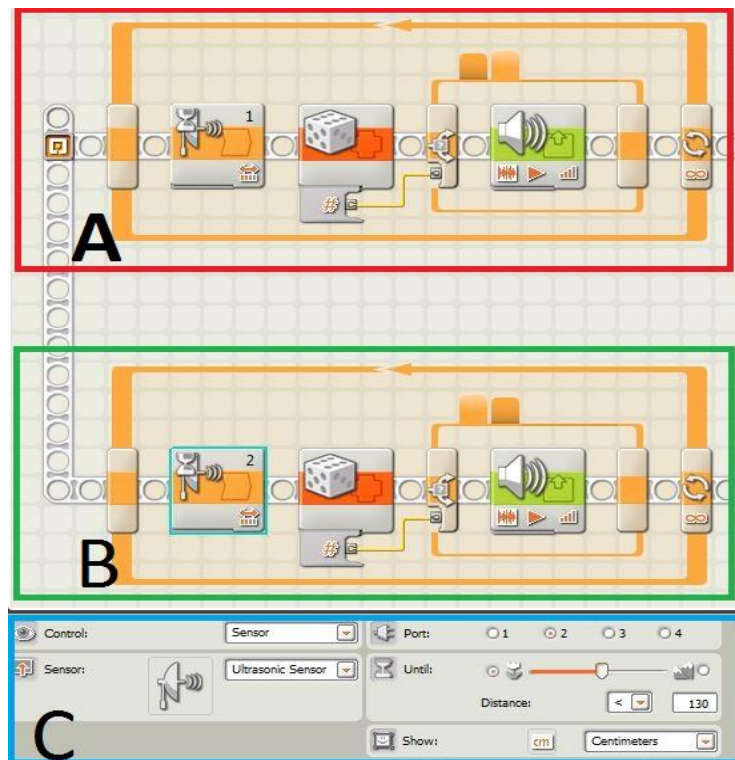
Zdroj: [vlastní]

2.4.2 Strážce chodby (ultrazvukové senzory)

Tento kód (viz obrázek 9) se dá využít při hlídání chodby. Když je robot umístěn v prostoru nebo na chodbě a senzory míří do prázdného prostoru, tak v případě umístění objektu před senzory (v tomto případě procházející osoba) je spuštěn alarm. Alarm bude opakován do té doby, dokud objekt před senzorem nebude odstraněn. V tomto příkladě jsou využity bloky z výběru Flow, Action a Data. Jedná se o bloky cyklu s názvem Loop, blok přepínání Switch, podmínka Wait, blok výstupu Sound a blok Random, který tvoří náhodné hodnoty.

Na řídicí jednotce NXT jsou umístěny 2 ultrazvukové senzory oba směřující jiným směrem. Jeden z nich je zapojen v portu číslo 1 a druhý v portu číslo 2. Programové kódy označené na obrázku písmeny A pro jeden senzor a písmenem B pro druhý senzor, běží paralelně na řídicí jednotce (viz obrázek 9). Označená část písmenem C je konfigurační panel pro ultrazvukový senzor, který je zapojen v portu číslo 2.

Programový kód označený písmenem A obsahuje blok pro nekonečně se opakující cyklus. V tomto cyklu jsou obsaženy všechny ostatní bloky. Prvním blokem v cyklu je blok Wait, který je nastaven na ultrazvukový senzor v portu 1. Kód za tímto blokem se bude vykonávat až po splnění podmínky, která je nastavena v tomto bloku na vzdálenost v rozmezí 0 až 130 centimetrů. Blok, který následuje po splnění podmínky je výběr náhodného čísla mezi čísly 1 a 2. Výstup, který obsahuje toto číslo je vstupem do podmínky Switch. Switch je nastaven na dvě možnosti numerického typu. Obě možnosti obsahují akční blok Sound a každý z nich je nastaven na jinou zvukovou stopu. Kód označený písmenem B je stejný s tím rozdílem, že je nastavený na druhý ultrazvukový senzor a blok Sound je nastavený na jiné zvukové stopy než v kódu A.



Obrázek 9: Strážce chodby – využití ultrazvukových senzorů v LEGO MINDSTORMS

Zdroj: [vlastní]

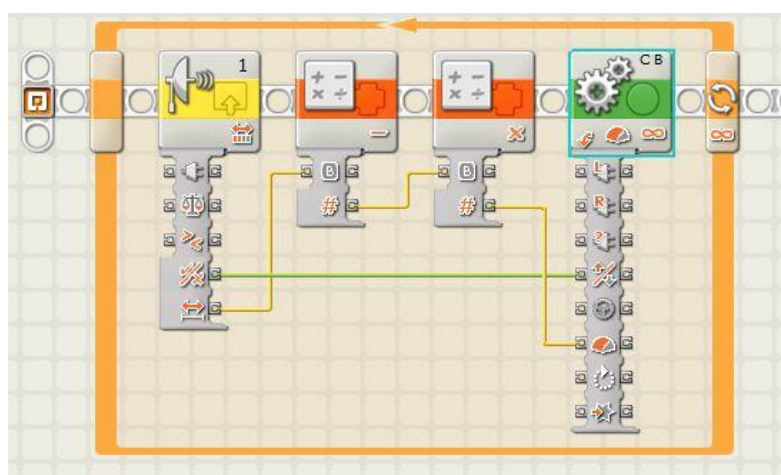
2.4.3 Najdi jih

V dalším příkladě je využit kompasový senzor od firmy HiTechnic. Robot s daným kompasem se točí dokola, dokud nenajede kompasem na jih a tam se má robot zastavit (viz obrázek 10).

Kompasový senzor není v základní výbavě stavebnice LEGO Mindstorms, a proto v softwaru od Lega pro tento senzor chybí blok. Blok pro kompasový senzor se může do programu dodatečně stáhnout jako aktualizace z oficiálních stránek firmy HiTechnic nebo se dá použít blok pro ultrazvukový senzor. V případě využití bloku pro ultrazvukový senzor není možné využít všechny funkce nabízené blokem přímo pro kompas a získáváme pouze hodnoty ze senzoru. V nastavení ultrazvukového bloku se musí přepnout režim na centimetry. Jelikož ultrazvukový senzor vrací hodnoty pouze od 0 do 250, nemůže plně nahradit kompasový senzor, který má hodnoty od 0 do 359. Hodnota získávána z kompasu je tedy dělena 2 a tím nám vzniká rozmezí od 0 do 179. Tímto způsobem znamená hodnota 0 sever, hodnota 45 východ, hodnota 90 znamená jih a 135 je západ [13].

Pro tento příklad jsou použity bloky z výběru Flow, Data, Common a Sensor. Jde o blok cyklu Loop, matematické bloky Math, senzorový blok pro ultrazvuk a blok pro pohyb více servomotorů Move.

Programový kód obsahuje nekonečný cyklus Loop. V něm se nachází všechny bloky pro daný program. Z výše uvedeného důvodu následuje blok pro ultrazvukový senzor, který je zapojen v portu číslo 1. Z něho jsou výstupem logické hodnoty, které ovlivňují funkci motorů. Dalším výstupem z tohoto bloku jsou samotné hodnoty získávané z kompasu dělené dvěma. Tyto hodnoty jsou vstupem do odčítacího bloku, kde se tyto hodnoty odečítají od čísla 90. Výstupní hodnota z tohoto bloku je v dalším bloku násobena číslem 2, pro rychlejší rotaci servomotorů. Čím nižší hodnota z kompasu bude, tím pomaleji se servomotory budou otáčet. Tato hodnota je nakonec vstupem do bloku Move, kde je spojena s ovládáním rychlosti motorů. V tomto bloku se musí nastavit doba otáčení na nekonečno a ve kterých portech jsou servomotory zapojeny.



Obrázek 10: Najdi jih – kód pro využití kompasu v LEGO MINDSTORMS

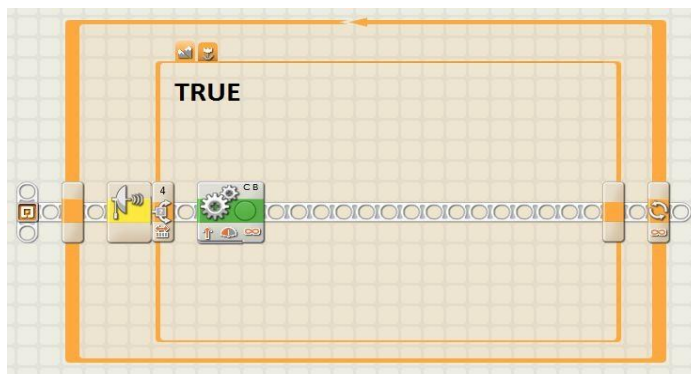
Zdroj: [vlastní]

2.4.4 Orientace v prostoru

Tento příklad ukazuje využití ultrazvukového senzoru pro základní orientaci v prostoru. Robot má tento senzor umístěný na sobě směřující dopředu, jezdí po prostoru a vyhýbá se překážkám. Využití bloků z výběru Common, Flow a Data.

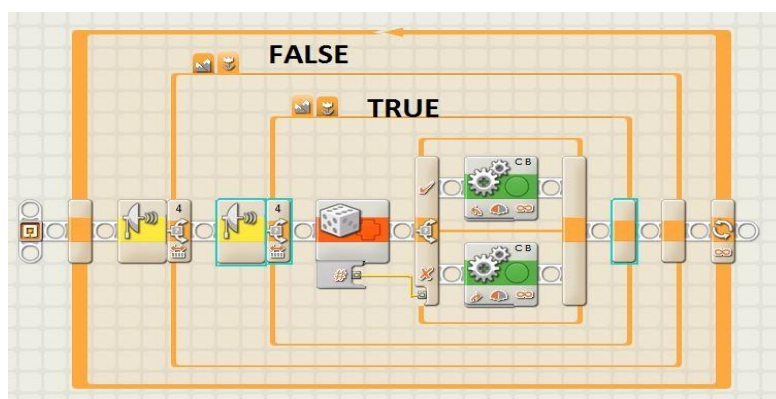
Po spuštění programu se spustí nekonečně opakující cyklus (viz obrázek 11). Prvním blokem co následuje je Switch, který je nastaven na ultrazvukový senzor v portu 4. Podmínkou je vzdálenost větší než 75 centimetrů. Při splnění této podmínky následuje blok Move ovládající servomotory, které rozjedou robota směrem vpřed. Při nesplnění této podmínky následuje další Switch (viz obrázek 12) s ultrazvukovým senzorem a podmínkou větší jak 10 centimetrů. V této situaci robot detekuje ve vzdálenosti mezi 10 až 75 centimetrů překážku a snaží se ji vyhnout. Blok Random tvoří dvě náhodná čísla, která ovlivňují výběr v dalším bloku Switch. Tento třetí Switch obsahuje na obou možnostech bloky Move

s nastavením jízdy vpřed, akorát s rozdílem, že jeden se točí doleva, druhý doprava. Pokud ve druhém bloku Switch není podmínka splněna, spustí se bloky Move s pohybem robota zpět a zvukový soubor (viz obrázek 13).



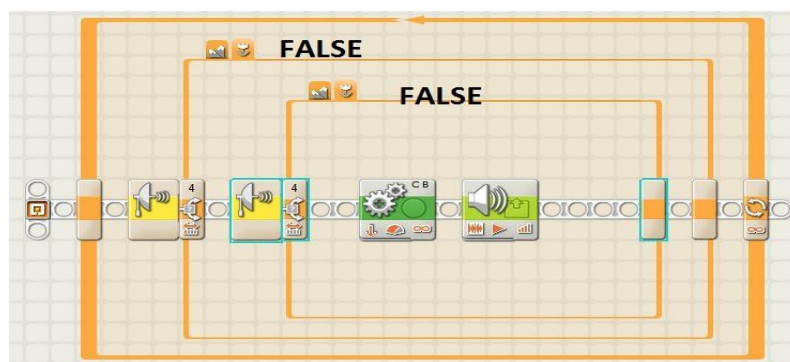
Obrázek 11: Orientace v prostoru – ultrazvukový senzor první switch true

Zdroj: [vlastní]



Obrázek 12: Orientace v prostoru – ultrazvukový senzor první druhý switch true

Zdroj: [vlastní]



Obrázek 13: Orientace v prostoru – ultrazvukový senzor druhý switch false

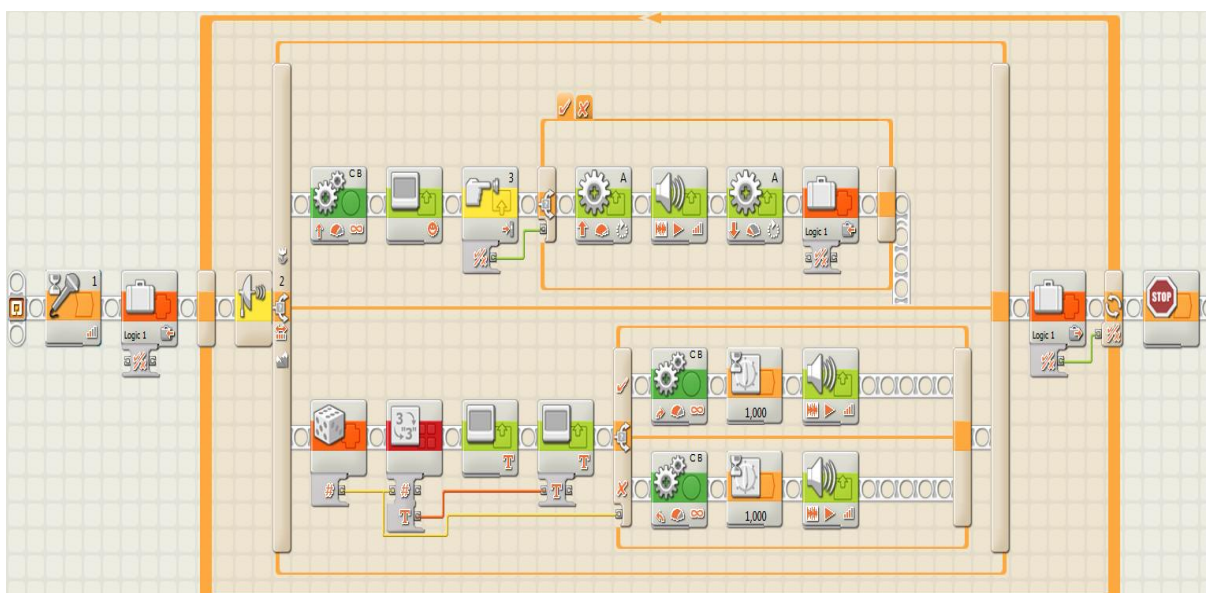
Zdroj: [vlastní]

2.4.5 Hlídaní prostoru

Robot v tomto příkladě hlídá prostor okolo něj. Robot čeká na místě do překročení určité zvukové hladiny, která označuje vetřelce. Poté se robot rozjede dokola a hledá ultrazvukovým senzorem tohoto narušitele. Po jeho nalezení se k němu rozjede a po sepnutí kontaktu na dotykovém senzoru se spustí třetí servomotor, který je umístěný na horní části robota a udeří vetřelce kladívkem. Poté se robot vypne. K tělu robota je připojeno více senzorů a tři servomotory. Servomotory v portech B a C zajišťují pohyblivost robota a servomotor v portu A má na sobě kladívko. Sensory jsou v portech s čísly 1 až 3. Jedná se o zvukový senzor, ultrazvukový senzor a dotykový senzor.

Programový kód (viz obrázek 14) obsahuje bloky ze všech částí výběru Complete. Je definována jedna logická proměnná s názvem Logic1. Po spuštění robota se aktivuje podmínkový blok Wait, nastavený na zvukový senzor, který je nastaven na hodnotu větší než 50. Po překročení hladiny zvuku je v následujícím bloku Variable nastavena logická proměnná na False. Poté se spouští cyklus, který se opakuje do té doby, dokud není proměnná Logic1 nastavena na True. V cyklu je obsažen Switch, který je nastavený na ultrazvukový senzor s hodnotou menší než 150 centimetrů. Při vzdálenosti větší než 150 centimetrů, je aktivována druhá možnost, kdy se vybere náhodné číslo v bloku Random mezi hodnotami 1 a 0. Tato hodnota je v následujícím bloku Number to Text změněna na typ text a pomocí dalšího bloku Display zobrazena na displeji řídicí jednotky. Náhodná hodnota je také vstupem do dalšího bloku Switch. Ten obsahuje v obou možnostech blok Move nastavený pro jízdu dokola, blok Wait nastavený na jednu sekundu a vydávání zvukového signálu označující hledání. Obě možnosti v bloku Switch se liší pouze směrem, do kterého se robot otáčí. Pokud je nalezena ultrazvukovým senzorem vzdálenost menší než 150 centimetrů, je aktivována první možnost v prvním bloku Switch. Robot se pomocí Move rozjede dopředu a blok Display zobrazí na displeji řídicí jednotky zobrazení příslušného obrázku. Robot momentálně jede vpřed za svým cílem. Pokračuje blok snímající hodnoty dotykového senzoru. Při stisknutí dotykového senzoru je poslána hodnota ze senzoru do bloku Switch. Je zde obsažena podmínka, při hodnotě False blok neobsahuje nic, a při hodnotě True je vykonána činnost určité série bloků. První z nich je spuštění servomotoru s kladívkem, který se otočí vpřed maximální rychlostí o 180°. Poté je vydán zvukový signál a servomotor s kladívkem se za pomoci dalšího bloku vrací zpět třetinovou rychlostí. Po dokončení této činnosti je v bloku Variable nastavena proměnná Logic1 na hodnotu True. Po vyskočení z bloku Switch je načtena proměnná Logic1 a její hodnota poslána do podmínky, která opakuje cyklus.

V případě, že hodnota proměnné je True, je cyklus ukončen a s pomocí bloku Stop se program v robotu vypne.



Obrázek 14: Kód pro hlídání prostoru – více senzorů v LEGO MINDSTORMS

Zdroj: [vlastní]

3 ENCHANTING

Enchanting je programovací prostředí pro robotické stavebnice LEGO Minstorms NXT. Byl vytvořen pro poskytnutí jednodušší cesty pro programování NXT řídicích jednotek než poskytují jiné jazyky. Program je založen na programovacím prostředí Scratch, jazyku Snap/BYOB a využívá také programovací jazyk leJOS NXJ [7].

Scratch

Scratch je programovací jazyk používající vlastní program, který využívá metody Drag and drop. Jeho pochopení je jednoduché a je vhodný pro začátečníky v programování, kteří zde mohou vytvářet vlastní animace, hry, muziku a realizovat i jiné nápady. Je zde velká podpora online komunity a diskusních fór, kde účastníci mohou nahrávat, stahovat či debatovat o svých programech a nápadech. Scratch byl vyvinut společností Lifelong Kindergarten Group v MIT Media Lab [22].

Snap (BYOB)

Snap (dříve pojmenovaném BYOB) je programovací jazyk využívající také metodu Drag and drop. BYOB je zkratka pro Build Your Own Blocks. Je rozšiřujícím doplňkem pro Scratch nebo Enchanting, který umožňuje vytvářet vlastní bloky a skripty. Program také může běžet v prostředí internetového prohlížeče. Snap je prezentován jako výtvar Univerzity v Kalifornii v Berkeley [24].

leJOS NXJ

Tento programovací jazyk na bázi Javy je využíván pro kompilaci algoritmů z Enchantingu do řídicí jednotky robota. Při každém uploadu do robota se kód převede do tohoto jazyka a nahraje se do paměti NXT.

Enchanting používá metodu Drag and drop, kde si potřebné prvky pouze přetáhneme z palety bloků do námi tvořených algoritmů, takže nemusíme psát, kompilovat a opravovat chyby ve složitých programových kódech. Program je takzvaný Otevřený software (Open source), takže je možné jakkoliv program využívat, prohlížet nebo upravovat [7]. Pro úpravu samotného programu je po instalaci Enchantingu, také ve zdrojové složce zobrazen soubor, který umožňuje úpravy v programu.

Program je od jeho vydání v roce 2011 neustále vyvíjen, doplňován novými možnostmi a jsou opravovány chyby, proto je třeba kontrolovat, zda nebyla vydána jeho nová verze. Pro svoji práci a tvorbu využívám verzi aplikace Enchanting 0.2.3. Jelikož je Enchanting stále

ve vývoji, tak nejde využít všechny funkce, co jsou nabízeny a byly převážně zkopírovány ze Scratche. Dále se jedná o nepřesné časování, kdy časovač v programu je pomalejší než má být, špatný přenos informací z řídicí jednotky zpět do počítače, občasné nepropojení řídicí jednotky s programem v počítači a další chyby, na kterých se pracuje, nebo byly zredukovány [7]. Dále je zde možnost využití spojení Bluetooth, ale tato možnost není přímo spojena s Enchantingem a je třeba použít jinou aplikaci, která je nainstalována společně s programem. Zde je, ale potřeba pokročilá znalost jazyků Java a leJOS a spojení je velmi pomalé.¹

3.1 Prvky programu

Tato část je věnována popisu zobrazovaného programu v okně a jeho základním částem. Enchanting lze rozdělit po spuštění na čtyři části (viz obrázek 15).

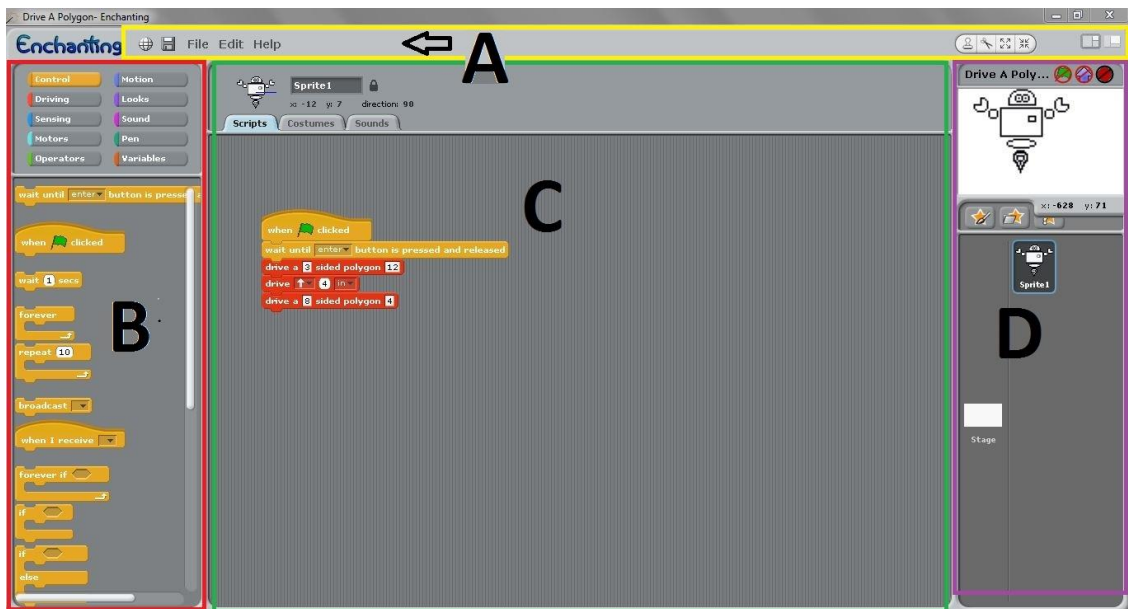
V první části na obrázku označené písmenem A je lišta nástrojů, která je obsažena ve většině programů. Úplně vlevo je tlačítko na přepínání jazyků. Je zde obsaženo přes 50 světových jazyků, včetně češtiny a slovenštiny, ale překlady těchto jazyků nejsou úplné a u nových verzí programu jsou některé nástroje stále v angličtině. Proto používám jen anglickou verzi tohoto programu. Dále tu jsou základní operace se soubory, možnost exportu kódu do jazyka Java, nahrání nového firmwaru do řídicí jednotky robota, zapnutí Bluetooth, poznámky k projektu a další. V pravé části jsou tlačítka na velikost zobrazení scény (část D).

V červeně označeném obdélníku s písmenem B je paleta bloků, které využíváme při sestavování algoritmů.

Třetí část s písmenem C je pracovní plocha. Sem se přetahují bloky z palety a tvoří se zde kódy. Pravým tlačítkem se mohou přidávat komentáře ke kódům. Je zde také možnost nahrávání zvuků a tvorba obrázků, které se zobrazí na displeji.

Vpravo je panel scény označený D a nachází se zde tlačítka pro nahrání kódu do robota, jeho spuštění a vypnutí. Dále tu je obrazovka, která zobrazuje obrázky a seznam spritů, pod kterými mohou běžet různé skripty.

¹ Tuto informaci mi poskytl přes email sám správce aplikace Clinton Blackmore <clinton.blackmore@gmail.com>



Obrázek 15: Rozložení prvků programu Enchanting

Zdroj: [vlastní]

3.2 Paleta bloků

Na paletě bloků je podle vzoru ze Scratche několik tlačítek, které se zobrazují ve výběru pod sebou v panelu nástroje a bloky. Tato tlačítka se dělí podle toho, jak ovládají chod algoritmu.

Control

Jsou zde obsaženy bloky, které ovlivňují samotný chod algoritmu a kódu. Jedná se o spuštění, pozastavení, vypnutí kódů. Dále různé podmínky typů IF, WAIT UNTIL, cykly s určitým počtem opakování, nekonečné cykly nebo cykly s podmínkou a bloky, které ovlivňují jiné skripty a podprogramy.

Driving

Tato část je pro robota se dvěma servomotory. Při jejich konfiguraci se volí průměr kol, jejich vzdálenost a porty v nichž jsou zapojeny. Po dokončení konfigurace se zobrazí nástroje pro pohyb robota. Jedná se o jízdu podle zadaného směru, vzdálenosti, rychlosti, otáčení robota a proměnné, se kterými se pracuje, jako jsou například: rychlost otáčení servomotoru, rychlost robota, pozice robota a další.

Sensing

Zde se nachází nástroje pro vstupní prvky do řídicí jednotky robota včetně časovače. Na začátku se musí provést jejich konfigurace, spočívající v určení o jaký senzor se jedná

a v jakém portu je zapojen. Poté se zobrazí všechny položky, které se daného senzoru týkají a jsou v tomto programu podporovány. Sensory zde nelze používat ve více módech, které jsou nabízeny v programu LEGO Mindstorms a vracejí pouze základní naměřenou hodnotu. Enchanting podporuje jen tyto senzory:

- všechny senzory dodávané se stavebnicí – ultrazvukový, světelný, dotykový, zvukový,
- HiTechnic Color sensor
- HiTechnic Compass sensor
- HiTechnic Gyro sensor

Motors

Používá se pro ovládání jednotlivých servomotorů. Po konfiguraci, při které se nastavuje o jaký servomotor se jedná a v jakém portu je zapojen, je na výběr z několika nástrojů pro jejich jednotlivé ovládání. Jde o spuštění otáčení daným směrem či zastavení, rotace o zadaný počet stupňů a nastavování rychlosti.

Operators

Jak již název napovídá, tak jsou zde na výběr matematické, logické a porovnávací operátory. Dále tu je také výběr z náhodného čísla, zaokrouhlení, funkce modulo, operace s textovými řetězci, goniometrické funkce, logaritmické funkce, absolutní hodnota a další operace a funkce.

Motion

Úprava a přesun obrázků, textů nebo jednotlivých bodů na displeji řídicí jednotky.

Looks

Vybrání obrázku, který bude zobrazen na displeji nebo ukázání nějakého textového řetězce.

Sound

Využívání a ovládání reproduktoru v řídicí jednotce. Lze využít některý z přednastavených zvukových stop nebo nahrát vlastní zvukový soubor.

Pen

Tato část se používá při tvorbě kreseb na displeji. Před kreslením se zapojí blok PEN DOWN, který zapne kreslicí čáru a poté pomocí bloků ze sekce Motion se tvoří kresba.

Variables

Zde se vytváří a mažou proměnné, také se zde nastavuje jejich hodnota a o kolik se má změnit. Při vytvoření se nespécifikuje, o jaký typ proměnné se jedná, ale program si s typovou konverzí poradí sám. Dále se zde vytváří seznamy, které zastupují funkci polí. V seznamech lze hodnoty přidávat, mazat, vkládat na určitou pozici nebo nahrazovat. Dalšími funkcemi je procházení prvků, zda obsahují zadanou hodnotu, ověřování prvků zda je to text a samotná délka seznamu. Dále je zde odkaz na začleněnou aplikaci BYOB pro tvorbu vlastních bloků.

3.3 Nahrání firmwaru a kódu do NXT jednotky

V této části bude popsána interakce programu s řídicí jednotkou, jako je nahrání firmwaru, nahrání kódu do robota, jeho spuštění a vypnutí.

Pokud máme novou robotickou stavebnici a ještě jsme nenahrávali nic přes Enchanting, je třeba nahrát do řídicí jednotky nový firmware, protože v jednotce je nahraný firmware pro LEGO Mindstorms. Verze firmwaru, se kterým Enchanting pracuje se nazývá leJOS 0.9.1. Instaluje se do jednotky buď aplikací přes nabídku Start, nebo přes lištu nástrojů v programu, kde se v obou případech klikne na tlačítko Flash Firmware a systém ho tam sám nahraje. Tento proces občas dělá chyby, kdy se nahrání zasekne a jednotka NXT vydává tiché klapání. V tomto případě je třeba jednotku restartovat a pokusit se o nové nahrání firmwaru. Proces přeinstalace firmwaru není nevratný a lze přes program LEGO Mindstorms nahrát zpět originální firmware.

Programový kód se do robota nahrává přes USB kabel při kliknutí na tlačítka v pravé horní části programu. Jsou zde 3 tlačítka. První z nich označené zelenou vlaječkou zkompiluje kód, nahraje ho do robota a ihned po dokončení procesu program spustí. Modrá šipka kód pouze zkompiluje a nahraje. Červené kolečko okamžitě zastaví spuštěný program. Program lze také spustit samostatně v robotovi. Nevýhodou Enchantingu je, že v jednotce NXT může být nahrán jen jeden kód, a proto je třeba programy zálohovat, než se nahraje nový kód. Dalším problémem je, když se při spuštěném Enchantingu hodnoty v kódu ukládají do seznamů nebo do proměnné a jednotka NXT je stále při spuštění spojena USB kabelem s počítačem. I když programový kód spustíme přes řídicí jednotku, tak se některé hodnoty přenášejí do počítače a zde se ukládají. Při opravě kódu a jeho následném nahrání se nahrají i tyto naměřené hodnoty a dělají chyby při opětovném spuštění programu v robotovi. Problém se dá vyřešit

kliknutím před každým novým nahráním programu na tlačítko Clear All Variables v liště nástrojů.

3.4 Ukázkové příklady

V této kapitole jsou ukázky mnou vytvořených příkladů v programu Enchanting. Příklady jsou převážně přizpůsobeny na využití senzorů HiTechnic, ale nechybí ani zapojení ostatních senzorů či použití servomotorů.

3.4.1 Orientace v prostoru – 2x ultrazvukový senzor

První příklad programu v Enchantingu. Robot se orientuje pomocí ultrazvukových senzorů a jezdí rovně po prostoru. Když uvidí v dálce nějakou překážku, začne zatáčet, aby se danému objektu vyhnul. Při bližší vzdálenosti se robot otočí nebo bude couvat, aby změnil směr jízdy (viz obrázek 16).

Na řídicí jednotce jsou umístěny po obou stranách servomotory, které jsou zapojeny v portech A a B. V přední části robota jsou po levé a pravé straně umístěny ultrazvukové senzory, které směřují oba vpřed. Senzor zapojený v portu číslo 3 je nazvaný LEFT ultrasonic sensor, dále jen levý senzor, a senzor zapojený v portu 4 má název RIGHT ultrasonic sensor 2, dále jen pravý senzor. Tento příklad je možné z důvodu nevyužitých 2 portů pro senzory a jednoho portu pro servomotory doplnit ještě o další funkce.

Samotný program je rozdělen do 6 částí označených písmeny A až E a esc. Část označená textem esc je kód, který se opakuje ve více příkladech. Tento kód umožňuje vypnutí spuštěného programu po stisknutí spodního tlačítka, který se nachází na řídicí jednotce. Po spuštění programu se spustí nekonečně opakující se cyklus, který obsahuje zbytek programu. Části obsažené v cyklu A až E jsou jednotlivé podmínky typu IF, které zastupují vybrané jednotlivé situace, které mohou vzniknout při pohybu robota. První podmínka s písmenem A nastává, když oba senzory naměří hodnotu větší jak 50 centimetrů. Při jejím splnění se nastaví rychlost obou servomotorů na otáčky 360°/s. Po nastavení rychlosti se oba servomotory spustí směrem vpřed. Druhá situace nastává v případě, že jeden ze senzorů zachytí vzdálenost menší než 5 centimetrů a druhý vzdálenost větší než 50 centimetrů. Toto nastává, když robot narazí stranou do překážky a nemůže pokračovat jízdou vpřed. Robot zastaví a rychlost servomotorů se nastaví na hodnotu 360°/s. Poté se oba motory otočí najednou o -1000°, což znamená o 1000° vzad a robot odjede od překážky. Nakonec se pravý motor otočí o 450°. Tím se robot otočí namísto přibližně o 90° stupňů doleva a senzory mohou registrovat nové hodnoty. V třetím případě s písmenem C robot narazí do překážky

přímo a oba senzory registrují hodnotu menší než 5 centimetrů. Je zde podmínka REPEAT UNTIL než levý senzor zachytí hodnoty větší než 50 centimetrů. Robot se rozjede dozadu a pravý servomotor se točí 2x rychleji než levý, tím se robot otáčí směrem vpravo. Části s písmeny D a E jsou pro případ, kdy robot jedoucí vpřed registruje jedním senzorem před sebou překážku ve vzdálenosti od 5 do 50 centimetrů a druhým senzorem vzdálenost větší než 50 centimetrů. V těchto případech se sníží rychlost servomotoru na opačné straně na rychlost 90°/s a robot začne měnit směr, tak aby se vyhnul překážce.

```

when clicked
  wait until [ESC] button is pressed and released
  stop all

  forever loop
    if distance from [left ultrasonic sensor] in cm > 50
      and distance from [right ultrasonic sensor 2] in cm > 50
        set left motor speed to 360 %/s for following commands
        set right motor speed to 360 %/s for following commands
        start turning left motor forward
        start turning right motor forward

    if distance from [left ultrasonic sensor] in cm < 50
      and distance from [left ultrasonic sensor] in cm > 5
      and distance from [right ultrasonic sensor 2] in cm < 50
      and distance from [right ultrasonic sensor 2] in cm > 5
        set right motor speed to 360 %/s for following commands
        set left motor speed to 360 %/s for following commands
        stop left motor by braking
        stop right motor by braking
        rotate right motor -1000 (just start it)
        rotate left motor 1000 (just start it)
        stop right motor by braking
        stop left motor by braking
        rotate right motor 450 (until done)

    if distance from [left ultrasonic sensor] in cm < 5 or
      distance from [right ultrasonic sensor 2] in cm < 5
      repeat until distance from [left ultrasonic sensor] in cm > 50
        set right motor speed to 720 %/s for following commands
        set left motor speed to 360 %/s for following commands
        start turning right motor backward
        start turning left motor backward

    if distance from [left ultrasonic sensor] in cm < 50
      and distance from [left ultrasonic sensor] in cm > 5
      and distance from [right ultrasonic sensor 2] in cm > 50
        set right motor speed to 90 %/s for following commands
        set left motor speed to 360 %/s for following commands
        start turning right motor forward
        start turning left motor forward

    if distance from [left ultrasonic sensor] in cm > 50
      and distance from [right ultrasonic sensor 2] in cm < 50
      and distance from [right ultrasonic sensor 2] in cm > 5
        set right motor speed to 360 %/s for following commands
        set left motor speed to 90 %/s for following commands
        start turning right motor forward
        start turning left motor forward
  
```

Obrázek 16: Kód pro orientaci v prostoru s využitím dvou ultrazvukových senzorů

Zdroj: [vlastní]

3.4.2 Barevný senzor rozlišení barev

Další příklad využívá barevný senzor od firmy HiTechnic (viz obrázek 17). S využitím daného senzoru je možné rozlišovat povrchové barvy objektů, které jsou nastaveny v senzoru. K tomuto programu je potřeba jen řídicí jednotka NXT a připojený barevný senzor, který je zapojen v portu 1. Poté ručně přejíždíme senzorem nad povrchy, které chceme snímat.

Pro program je vytvořena proměnná s názvem barva. Po zapnutí programu je tato proměnná nastavena na text „neznama“ a spuštěna auto kalibrace senzoru. Dále pokračuje nekonečně se opakující cyklus, který má v sobě ostatní zbylé bloky. Patří sem několikrát za sebou podmínka typu IF, které jsou nastaveny na konkrétní barvy naměřené senzorem. Pokud je tato určitá podmínka splněna, je nastavená proměnná barva na text s názvem dané barvy. Nakonec je hodnota proměnné zobrazena na displeji jednotky pomocí bloku Print.



Obrázek 17: Kód pro rozlišování barev pomocí barevného senzoru v Enchantingu

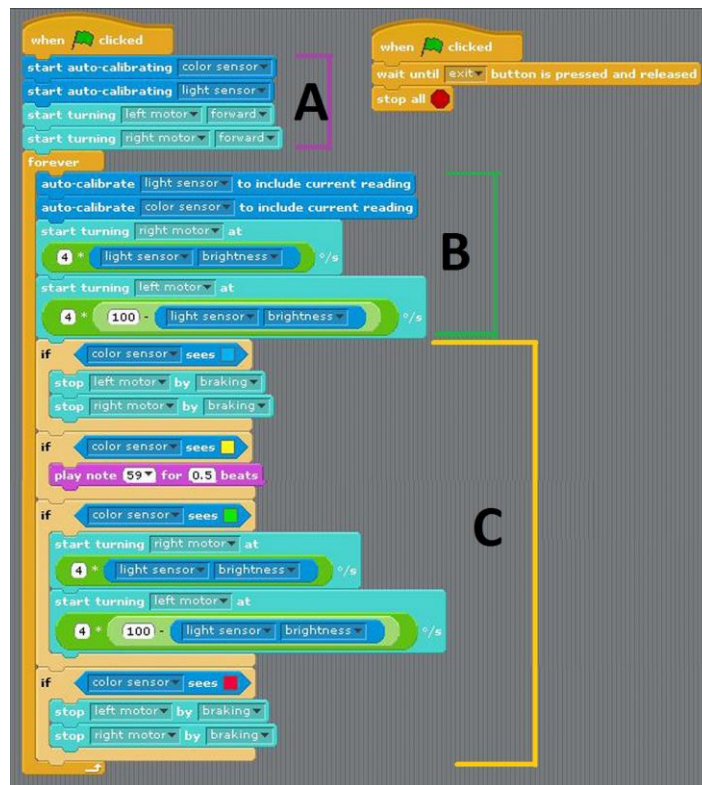
Zdroj: [vlastní]

3.4.3 Barevný senzor - semafor

Tento příklad také využívá barevný senzor HiTechnic. Na rozdíl od předchozího příkladu, kdy jsme s jednotkou a senzorem manipulovali ručně, tady má jednotka servomotory pro samostatný pohyb. Situace je taková, že robot se pohybuje po bílé podložce s černou čarou pomocí světelného senzoru. Někde vedle černé čáry je umístěn semafor. Robot po přijetí k semaforu registruje jeho barvu pomocí barevného senzoru, a buď zastaví, nebo pokračuje dál v jízdě.

Robot má na sobě dva servomotory po stranách, které mu umožňují jízdu. Dále obsahuje světelný senzor, který je umístěn v přední části robota, směřuje k zemi a je zapojen v portu 1. Barevný senzor, který je zapojen v portu 2. Tento senzor je umístěn na boku robota nebo vedle světelného senzoru, ale ne tak blízko aby se tyto dva senzory ovlivňovaly. Definitivní pozice tohoto senzoru je určena tím, jak je umístěn semafor, aby mohly být barvy registrovány.

Samotný kód (viz obrázek 18) příkladu je rozdělen na 4 části označené písmeny A, B, C a sekvence pro vypnutí spuštěného programu, která se nachází v pravé horní části obrázku. Pod písmenem A je spuštění auto kalibrace zapojených senzorů, což přivede tyto senzory k provozu. Dále se spustí oba servomotory směrem vpřed. Pokračuje nekonečně se opakující cyklus, který obsahuje zbytek kódu označený pod písmeny B a C. Písmeno B obsahuje auto kalibraci obou senzorů kvůli neustálým změnám měřených povrchů. Dále následují 2 bloky, které ovládají spuštění a rychlost servomotorů. První z nich násobí čtyřmi hodnotu získanou ze světelného senzoru a tato hodnota je nastavena jako rychlost pravého motoru. Druhý blok nastavuje rychlost tak, že odečítá od stovky hodnotu senzoru a poté ji vynásobí čtyřmi. Kód označený písmenem C je ovládán pomocí barevného senzoru. Když robot dojedez k semaforu nebo najede na kartičky simulující semafor, tak je pomocí několika IF podmínek ověřeno, zda je splněna zadaná podmínka a poté vykonán určitý kód. Při najetí na modrou nebo červenou barvu robot zastaví. Při zobrazení žluté barvy robot vydá zvukový tón a při zelené barvě se robot rozjede vpřed podle černé čáry, až se dostane mimo rozsah barev semaforu a pokračuje podle kódu z části B.



Obrázek 18: Kód pro jízdu po čáře a využití barevného senzoru pro semafor

Zdroj: [vlastní]

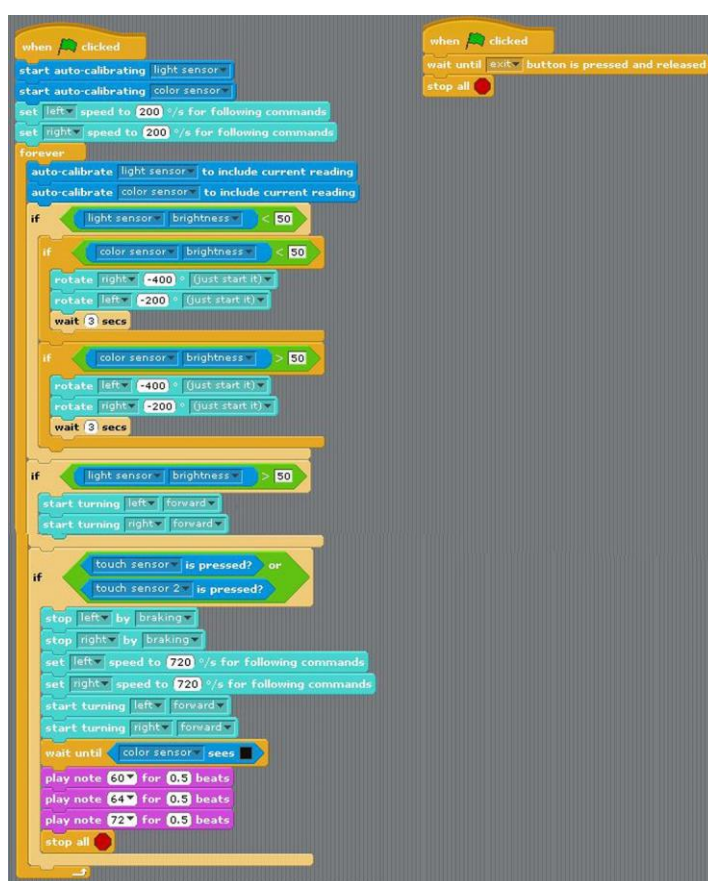
3.4.4 Vytlačení objektu z prostoru

Příklad, který znovu využívá barevný senzor a světelný senzor. Dále k nim jsou na robota přidány dva dotykové senzory a dva servomotory zajišťující pohyb. Situace je taková, že na podložce je vymezen černou čarou prostor pro pohyb robota. Někde v tomto prostoru je umístěn na druhé straně od robota objekt, který má být z prostoru odstraněn vystrčením. Po spuštění robot začne jezdit po prostoru, nesmí přejet černou čarou a po kontaktu s objektem a dotykovým senzorem ho odstraní.

V přední části robota po stranách jsou umístěny barevný senzor a světelný senzor směřující k zemi. Kousek před nimi jsou namontovány dva dotykové senzory s tyčkami, které jsou vedeny horizontálně před robotem, pomáhající stisknutí dotykových senzorů. Servomotory jsou připojené k řídicí jednotce, tak aby se robot mohl pohybovat.

Programový kód (viz obrázek 19) obsahuje část pro vypnutí robota a samotný kód ovládající jeho akce. Na začátku je auto kalibrace barevného senzoru, světelného senzoru a nastavení rychlosti obou servomotorů na 200°/s. Následuje opakující se nekonečně cyklus se třemi hlavními podmínkami typu IF. První z nich je nastavena na situaci, kdy světelný senzor zachytí hodnotu menší jak 50. Tato hodnota je pro přejetí černé čáry. Kód, který má

být vykonán, obsahuje další dvě podmínky IF. Tyto podmínky určují, zda robot najel na čáru zleva nebo zprava. Obě sekvence typu IF mají v podmínce hodnotu barevného senzoru, který momentálně pracuje jako světelný senzor. Rozdíl mezi nimi je buď větší, nebo menší než hodnota 50. V prvním případě se robot otočí o několik stupňů doprava a v druhém případě doleva. Druhá hlavní podmínka používá světelný senzor. Zajišťuje pohyb robota vpřed, když se nachází v prostoru na bílé podložce. Třetí hlavní podmínka nastává, když se narazí do objektu a je jeden nebo druhý z dotykových senzorů stisknut. V tomto případě se robot zastaví, nastaví se vyšší rychlost servomotorů na 720°/s a rozjede se vpřed, tlačící před sebou objekt. Následuje podmínka typu Wait, dokud barevný senzor nezachytí přejezd přes černou čáru. Poté se vydají tři zvukové signály a celý program se vypne.



Obrázek 19: Kód pro vytlačení objektu z prostoru v Enchantingu

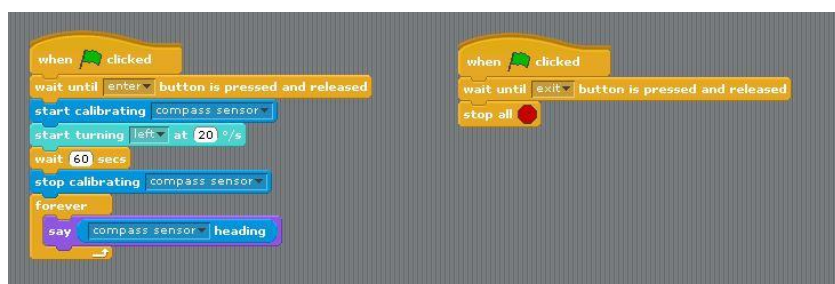
Zdroj: [vlastní]

3.4.5 Kompas – měření azimutu

Pro tento program je potřeba kompasový senzor od firmy HiTechnic. Robot má za úkol jezdit dokola a ukazovat na displeji azimut, kterým právě směřuje. Robot má na sobě dva servomotory, ale pohybuje se jen jeden a kompasový senzor, který je umístěn vysoko nad

robotem. Výšku doporučuji co nejvyšší, kam až dovolí připojovací kabel, z důvodu možného ovlivnění výsledku.

Kód (viz obrázek 20) obsahuje sekvenci pro vypnutí programu a samotný kód ovládající akce robota. Po spuštění programu v robotovi je třeba stisknout tlačítko enter. Poté se spustí kalibrace kompasu a levý servomotor s rychlostí 20°/s. Robot se při kalibraci senzoru musí otáčet pomalu a provést minimálně jednu otáčku. Proto následuje čekací blok Wait, který čeká, než uběhne 60 vteřin. Po tomto kroku se kalibrace zastaví, ale ne pohyb robota a následuje nekonečně se opakující cyklus, který obsahuje blok pro zobrazení dat na displeji. V tomto bloku je obsažena momentální hodnota azimutu, kterým stále otáčející robot směřuje.



Obrázek 20: Měření azimutu pomocí HiTechnic Kompas senzoru

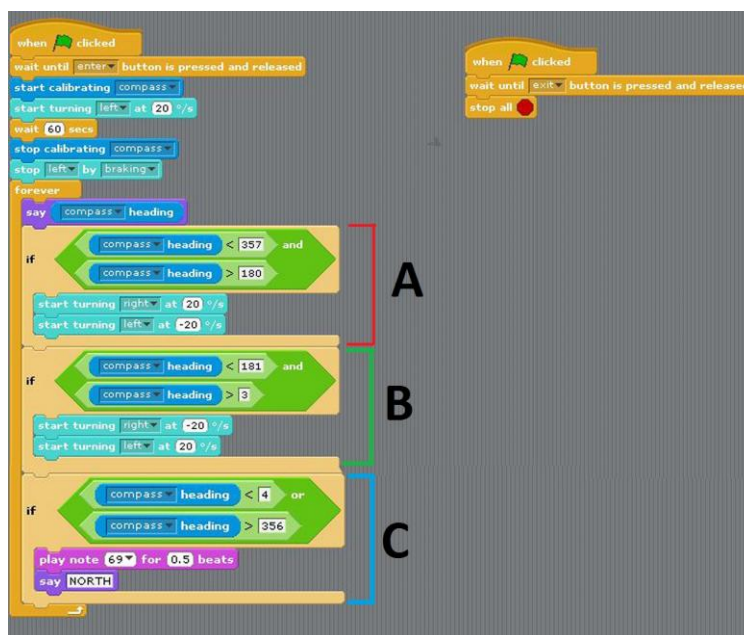
Zdroj: [vlastní]

3.4.6 Otočení robota směrem na sever

Robot obsahuje stejnou výbavu a má podobný úkol jako příklad 2.4.3 z programu LEGO Mindstorms. Na rozdíl od předchozího příkladu jsem tento příklad vyřešil pomocí intervalů, robot hledá sever a tento příklad je trochu nepřesný. Robot se v tomto případě pohybuje pomalu, kvůli tomu, že nestíhá číst hodnoty ze senzoru a docházelo by k chybám. S robotem lze i během otáčení manipulovat a robot bude stále hledat sever.

Na začátku kódu (viz obrázek 21) je stejná sekvence bloků pro kalibraci kompasového senzoru jako v příkladě 3.4.5, ale s tím rozdílem že se robot po dokončení kalibrace zastaví. Poté následuje nekonečně se opakující cyklus, který obsahuje blok say pro zobrazení hodnoty azimutu na displeji a tři podmínky IF označené písmeny A až C. První z nich označená písmenem A je nastaven na interval hodnot z kompasu mezi hodnotami větší než 180 a menší než 357. V tomto i v následujících intervalech jsou použita jen znaménka větší (>) nebo menší (<), a proto se některé hodnoty překrývají. V tomto případě, se robot začne otáčet namísto směrem doprava. V druhé podmínce s písmenem B je interval hodnot mezi menší než 181 a větší než 3. Spojením těchto dvou intervalů nám vznikne třetí a zbývající interval s hodnotami mezi menší než 4 nebo větší než 356. Při dosažení tohoto intervalu je vydáván

zvukový tón a na displeji se zobrazí pomocí bloku say text NORTH. Robot se, ale při dosažení severu nezastaví a neustále přepíná mezi podmínkami A a B.



Obrázek 21: Kód pro otočení robota směrem na sever v Enchantingu

Zdroj: [vlastní]

3.4.7 Kompas – měření rovnoběžnosti přímek

Pomocí kompasového senzoru lze také měřit rovnoběžnost přímek. K řídicí jednotce je připojen kompas, který je ale upevněn jen v první části programu. Ke kompasu je připojena vodorovně podél jeho délky tyčka, s jejíž pomocí se měří přímky. Samotné měření má určitou toleranci a není úplně přesné.

Samotný kód (viz obrázek 22) jsem rozdělil na tři části označené písmeny A až C. Pod písmenem A se skrývá kalibrace kompasu. V této části musí být kompas upevněn na robotovi ve vhodné vzdálenosti. Po spuštění programu je třeba spustit tlačítko enter na robotovi. Spustí se kalibrace daného senzoru a robot se za pomocí bloků z části driving otočí pomalu dokola o 720°. Poté přecházíme do části B. Kompas už zde není upevněn na robotovi, ale měříme s ním dané přímky. Kompas přiložíme k první přímce a stiskneme tlačítko enter. Čeká se 3 vteřiny a hodnota kompasu se uloží do proměnné A. Po další vteřině zazní zvukový signál, kompas přiložíme k další přímce, se kterou porovnáváme první přímku a stiskneme tlačítko enter. Robot znovu vyčká tři vteřiny, uloží směr do proměnné B a vydá jiný zvukový signál. Následuje část označená C, ve které je podmínka typu IF ELSE, která zajišťuje měření úhlů v rozmezí mezi 0° až 180°. Když je hodnota B minus A větší než 180° tak se od odečtené hodnoty odečte 180 a tato nově získaná hodnota je zobrazena pomocí

bloku say na displeji. Jinak je pouze od B odečteno A a také zobrazeno na displeji. Program vyčká do opětovného stisknutí tlačítka enter a vypne se.



Obrázek 22: Využití kompasového senzoru pro měření úhlů

Zdroj: [vlastní]

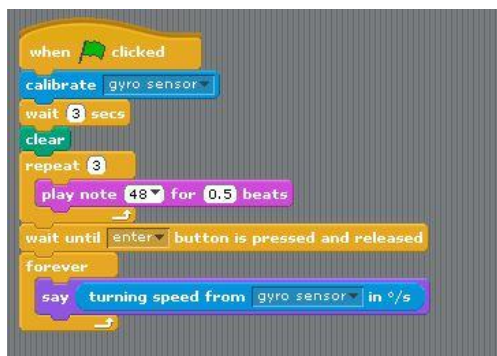
3.4.8 Hodnoty z gyroskopu

Jak již bylo řečeno tak gyroskop vrací hodnotu rychlosti otáčení v úhlech za vteřinu. Tento senzor se hodí pro vytváření robotických vozítek typu segway, které se ale z důvodu pomalejšího zpracování kódu v Enchantingu zde nedají vytvořit. Proto zde dávám alespoň příklad toho, jak lze zobrazit dané hodnoty na displeji.

K řídicí jednotce je připojen servomotor, který ale budeme ovládat ručně a jen v rozmezí 180°. Na pohyblivé části servomotoru je připojen gyroskop, který se také pohybuje. Při manipulaci s gyroskopem nebo samotným servomotorem se získávají hodnoty ze senzoru a ty se zobrazí na displeji.

Samotný kód pro tento program (viz obrázek 23) není nijak dlouhý. Po spuštění se spustí kalibrace gyroskopu a vyčká se tři vteřiny. Poté se pomocí bloku clear vymaže obrázek z displeje řídicí jednotky a pomocí cyklu, který je nastaven na tři opakování se spustí zvukový signál. Následuje blok, který čeká na stisknutí tlačítka enter a spustí se nekonečně opakující cyklus, který pomocí bloku say zobrazuje aktuální hodnoty ze senzoru na displeji. Pokud

je třeba hodnoty ukládat stačí vytvořit pole a vkládat hodnoty, ale toto již není součástí tohoto programu.



Obrázek 23: Získávání hodnot z gyroskopu

Zdroj: [vlastní]

ZÁVĚR

Hlavním cílem této bakalářské práce bylo vytvořit ukázkové příklady v programech LEGO Mindstorms a Enchanting. Tyto příklady jsou poté nahrány do řídicí jednotky NXT a ze stavebnice je postaven robot, který vykonává dané činnosti.

V první části je samotný úvod do robotiky a robotických stavebnic. Dále jsou popsány vybrané robotické stavebnice. Je k nim přiřazen jednoduchý popis od koho pocházejí, toho co obsahují a možné programovací jazyky, které se dají k nim využít. Ke stavebnicím LEGO Mindstorms je výběr programovacích jazyků rozšířený. Jsou zde také popsány vybrané doplňkové senzory HiTechnic, z nichž některé jsou využívány. Tyto jazyky a stavebnice jsem vybral podle svého uvážení.

Druhá část této bakalářské práce je věnována programu LEGO Mindstorms, známého také jako NXT-G. Je zde základní popis programu a funkce, se kterými jsem se setkal, a jsou v nabídce. Pokračuje postup, jak se tvoří programy a jejich nahrání do řídicí jednotky robota. Tyto zkušenosti jsem získal z praxe. Na konec této části je popsáno několik vytvořených příkladů. První z nich s názvem Modulo se nehodí přímo pro robota, ale dá se využít při tvorbě jiných kódů. V dalších příkladech jsou využity převážně ultrazvukové senzory, kompasový senzor a jiné senzory. Zde jsem chtěl dát kód pro segway, ale kvůli jeho složitosti se mi ho nepodařilo sestavit.

Třetí část bakalářské práce patří programu Enchanting. Jsou zde základní popis programu, funkcí a možností nahrání. Pokračují vymyšlené ukázkové příklady. První z nich využívá 2 ultrazvukové senzory pro orientaci po prostoru. Ostatní příklady používají doplňkové senzory od firmy HiTechnic, které jsou v tomto programu podporovány. Nepodařilo se mi vytvořit lepší příklady pro gyroskop z důvodu pomalejšího zpracování kódu v řídicí jednotce a nedostatku nápadů jak problém vyřešit.

Samotné rozdíly mezi použitými programovými prostředími, se kterými jsem se setkal, jsou ty, že program od LEGA je výkonnější, umožňuje více možností a funkcí, podporuje i dodatečné aktualizace pro nové senzory. Jeho nevýhodou může být poněkud větší složitost na začátku, kdy se uživatel snaží tento program naučit. Enchanting je oproti prvnímu programu intuitivnější a jelikož umožňuje jen některé funkce tak i přehlednější. Znalost programu Scratch je při použití tohoto programu výhodou. Samotnými nevýhodami je delší doba při zpracování a vykonávání kódu v řídicí jednotce, některé funkce programu jsou nedostupné nebo omezené. Enchanting samotný stále prochází určitým vývojem, jsou

vydávány nové verze, proto je třeba kontrolovat jejich aktuálnost a od začátku své práce jsem se setkal minimálně se 4 novými verzemi.

K práci je přiloženo CD s programovými kódy obou programů.

POUŽITÁ LITERATURA

- [1] Atmel AVR32850: ATSAM4L-EK User Guide. In: ATMEL microcontroller [online]. 2013 [cit. 2013-02-12]. Dostupné z: <http://www.atmel.com/images/doc42026.pdf>
- [2] AXE005 PICAXE-20M starter pack. In: PICAXE microcontroller [online]. 2008 [cit. 2013-02-12]. Dostupné z: <http://www.picaxe.com/docs/axe005.pdf>
- [3] BAGNALL, Brian. Intelligence unleashed: creating LEGO NXT robots with Java. Winnipeg, Manitoba: Variant Press, 2011. ISBN 978-098-6832-208.
- [4] BAGNALL, Brian. Maximum Lego NXT: building robots with Java Brains. Winnipeg, Man.: Variant Press, c2007, iv, 505 s. ISBN 978-097-3864-915.
- [5] BENEŠ, Michal. Aplikace Microsoft Robotics pro simulaci procesů. Pardubice, 2011. Dostupné z: <http://hdl.handle.net/10195/39595>. Bakalářská práce. Univerzita Pardubice. Vedoucí práce Ing. Panuš Jan, Ph.D.
- [6] Bioloid. ROBOTIS [online]. 2013 [cit. 2013-03-02]. Dostupné z: http://www.robotis.com/x/BIOLOID_main_en
- [7] BLACKMORE, Clinton. Enchanting [online]. 2011-2013 [cit. 2013-02-13]. Dostupné z: <http://enchanting.robotclub.ab.ca/tiki-index.php>
- [8] BrickOS Home Page. BrickOS [online]. 2004 [cit. 2013-02-13]. Dostupné z: <http://brickos.sourceforge.net/>
- [9] Fischertechnik GmbH [online]. 2012 [cit. 2013-02-12]. Dostupné z: <http://www.fischertechnik.de/en/Home.aspx>
- [10] Getting started. Nxt-python [online]. 2012 [cit. 2013-02-13]. Dostupné z: <http://code.google.com/p/nxt-python/>
- [11] GRIFFIN, Terry. The art of LEGO Mindstorms NXT-G programming. San Francisco, CA: No Starch Press, c2010, xvii, 200 p. ISBN 15-932-7218-9.
- [12] HANSEN, John C. Lego mindstorms NXT power programming: [robotics in C]. 2nd ed. Winnipeg: Variant Press, 2009. ISBN 978-097-3864-977.g
- [13] Hitechnic [online]. © 2001-2012 [cit. 2013-02-13]. Dostupné z: <http://www.hitechnic.com/>
- [14] Lego Mindstorms NXT [online]. 2012 [cit. 2013-02-13]. Dostupné z: <http://mindstorms.lego.com/en-us/default.aspx>

- [15] LEGO MINDSTORMS NXT Support from MATLAB and Simulink. THE MATHWORKS, Inc. MathWorks [online]. © 1994-2013 [cit. 2013-03-02]. Dostupné z: <http://www.mathworks.com/academia/lego-mindstorms-nxt-software/>
- [16] MERKUR: Robotika a mechatronika [online]. 2013 [cit. 2013-02-12]. Dostupné z: <http://www.merkurtoys.cz/vyrobky/robotika-a-mechatronika>
- [17] Microsoft Robotics Development Studio. Microsoft Robotics [online]. 2013 [cit. 2013-02-13]. Dostupné z: <http://www.microsoft.com/robotics/>
- [18] MORGAN, Sara. Programming Microsoft Robotics studio. Microsoft Press, xxviii, 253 p. ISBN 07-356-2432-1.
- [19] NXT.NET for LEGO Mindstorms. CodePlex [online]. 2010 [cit. 2013-02-13]. Dostupné z: <http://nxtnet.codeplex.com/>
- [20] Programování PICAXE 08M. Programujte.com [online]. 2010 [cit. 2013-02-12]. Dostupné z: <http://programujte.com/clanek/2010071900-programovani-picaxe-08m/>
- [21] ROBOTIS. BIOLOID Premium Kit. Jižní Korea, 2012. Dostupné z: http://www.megarobot.net/cj/manualy/robotis/bioloid/Bioloid%20Premium__.pdf
- [22] Scratch [online]. 2012 [cit. 2013-03-08]. Dostupné z: <http://scratch.mit.edu/>
- [23] Sensors. Vernier [online]. 2013 [cit. 2013-02-13]. Dostupné z: <http://www.vernier.com/products/sensors/>
- [24] SNAP! (BYOB) [online]. 2011 [cit. 2013-03-08]. Dostupné z: <http://snap.berkeley.edu/>
- [25] STALLMAN, Brian Gough. Forew. by Richard M. An introduction to GCC: for the GNU compilers gcc and g. 2. print., rev. and updated. Bristol: Network Theory Ltd, 2005. ISBN 978-095-4161-798
- [26] Tutorial: Understanding the difference between NXT set versions. Robot Square [online]. 2012 [cit. 2013-02-13]. Dostupné z: <http://robotsquare.com/2012/02/18/understanding-nxt-versions/>