

Univerzita Pardubice
Fakulta ekonomicko-správní

Kooperace robotických systémů

Adam Babec

Bakalářská práce

2017

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Adam Babec**
Osobní číslo: **E13718**
Studijní program: **B6209 Systémové inženýrství a informatika**
Studijní obor: **Informatika ve veřejné správě**
Název tématu: **Kooperace robotických systémů**
Zadávající 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 vytvoření a naprogramování několika samostatných robotických jednotek, které spolu budou spolupracovat a reagovat na sebe. K vytvoření robotů bude použita stavebnice Mindstorm od firmy Lego.

Osnova:

1. Složení robotických stavebnic Lego Mindstorm
2. Tvorba algoritmů pro pohyb robota
3. Implementace algoritmů určených pro kooperaci robotů

Rozsah grafických prací:

Rozsah pracovní zprávy: **cca 60 stran**

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

Seznam odborné literatury:

1. TAUFER I. Algoritmy a algoritmizace - vývojové diagramy. Pardubice: Univerzita Pardubice, 2009. ISBN 978-80-7395-182
2. PŠENČÍKOVÁ, Jana. Algoritmizace. 1. vydání. Kralice na Hané : Computer Media, 2007. 128 s. ISBN: 80-86686-80-9.
3. MILKOVÁ, Eva. Algoritmy: objasnění, procvičení a vizualizace základních algoritmických konstrukcí. Praha : Alfa, 2008. 114 s. ISBN: 978-80-87197-10-3.
4. GRIFFIN, Terry. The art of LEGO Mindstorms NXT-G programming. San Francisco, CA: No Starch Press, 2010, xvii, 200 p. ISBN 1593272189.
5. KNUTH, Donald,Er. Umění programovat I: Základní algoritmy. 1. vyd. Brno: Computer Press, 2008. ISBN 978-80-251-2025-5.

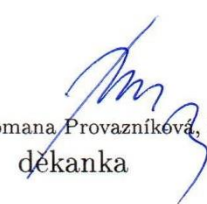
Vedoucí bakalářské práce:


Ing. Jan Panuš, Ph.D.


Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce: **4. září 2016**

Termín odevzdání bakalářské práce: **28. dubna 2017**


doc. Ing. Romana Provozničková, Ph.D.
děkanka

L.S.


doc. Ing. Pavel Petr, Ph.D.
vedoucí ústavu

V Pardubicích dne 4. září 2016

PROHLÁŠENÍ AUTORA

Prohlašuji, že jsem zadanou bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jana Panuše, Ph.D. Veškeré literární prameny a informace, které jsem pro práci použil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na mou 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 10. 8. 2017

Adam Babec

PODĚKOVÁNÍ

Rád bych poděkoval za odbornou spolupráci vedoucímu práce, panu Ing. Janu Panušovi Ph.D., paní Mgr. Evě Kousalové za pomoc s korekturami a svým rodičům za psychickou a finanční pomoc.

ANOTACE

Tato práce seznamuje s historií robotiky, s technologií Bluetooth, s popisem stavebnice Lego Mindstorm EV3, jejími částmi a základy sestavování algoritmů.

V praktické části demonstřuji možnosti spolupřáce více robotických jednotek, vytvořených z robotické stavebnice Lego Mindstorm EV3.

KLÍČOVÁ SLOVA

Kooperace robotických systémů, roboti, Lego Mindstorm EV3, Bluetooth, algoritmus

TITLE

Cooperative robotic systems

ANNOTATION

This work introduces the history of robotics, Bluetooth technology, the Lego Mindstorm EV3 kit, its parts and the basics of algorithm compilation.

In the practical part, I demonstrate the possibilities of co-operating with more robot units created from the Lego Mindstorm EV3 robotic kit.

KEYWORDS

Cooperative robotic systems, robots, Lego Mindstorm EV3, Bluetooth, algorithm

OBSAH

ÚVOD	10
1 HISTORIE ROBOTIKY	11
2 BLUETOOTH	12
2.1 HISTORIE	12
2.2 STANDARDY BLUETOOTH	12
2.3 VYUŽITÍ TECHNOLOGIE BLUETOOTH.....	13
2.4 JAK BLUETOOTH PRACUJE	14
2.5 TŘÍDY A DOSAH.....	14
3 LEGO MINDSTORMS	15
3.1 INTELIGENTNÍ KOSTKA EV3.....	15
3.1.1 <i>Popis</i>	15
3.1.2 <i>Vzhled</i>	15
3.2 EV3 SERVOMOTORY	16
3.3 SENZORY	16
3.3.1 <i>Dotykový senzor</i>	16
3.3.2 <i>Barevný senzor</i>	16
3.3.3 <i>Gyroskopický senzor</i>	17
3.3.4 <i>Ultrazvukový senzor</i>	17
3.3.5 <i>Senzory třetích stran</i>	17
3.4 PROPOJENÍ	18
4 ZÁKLADY ALGORITMIZACE	19
4.1 MOŽNOSTI ZÁPISU ALGORITMU	19
4.2 VÝVOJOVÉ DIAGRAMY	20
4.2.1 <i>Mezní značky vývojových diagramů</i>	20
4.2.2 <i>Sekvenční bloky vývojových diagramů</i>	20
4.2.3 <i>Větvění vývojových diagramů</i>	21
4.2.4 <i>Další značky</i>	22
5 TVORBA ROBOTICKÉ ÚLOHY	23
5.1 EV3 BASIC	24
5.2 SEZNÁMENÍ S PROGRAMOVACÍM JAZYKEM	25
5.3 PŘÍKAZY NA OBSLUHU MOTORŮ.....	27
5.4 PŘÍKAZY PRO KOMUNIKACI PŘES BLUETOOTH	27
5.5 PŘÍKAZY NA OBSLUHU SENZORŮ.....	28
5.6 DISPLAY A TLAČÍTKA	28
5.7 TEXTOVÉ FUNKCE	28
6 KONSTRUKCE ROBOTŮ	29
7 ROBOTICKÉ ÚLOHY	31
7.1 ZASLÁNÍ ZPRÁVY Z JEDNOHO ROBOTA DO DRUHÉHO	33
7.2 ROZPARSOVÁNÍ ZPRÁVY NA POKYNY PRO ROBOTA.....	33
7.3 SPOLEČNÉ ZVLÁDNUTÍ ZASTAVENÍ PŘED PŘEKÁŽKOU NEBO NA KONCI POLE	35
7.4 SPOLEČNÉ ZVLÁDNUTÍ PROJETÍ BLUDIŠTĚM	36
8 ZÁVĚR	40

SEZNAM ILUSTRACÍ A TABULEK

Obrázek 1 - Logo Bluetooth.....	12
Obrázek 2 - EV3 kostka	15
Obrázek 3 - Primární servomotor	16
Obrázek 4 - Sekundární servomotor.....	16
Obrázek 5 - Dotykový senzor.....	16
Obrázek 6 - Barevný senzor	16
Obrázek 7 - Gyroskopický senzor	17
Obrázek 8 - Ultrazvukový senzor.....	17
Obrázek 9 - Zapojení konektoru.....	18
Obrázek 10 - Příklad zapojení sběrnice.....	18
Obrázek 11 - Značka pro začátek a konec algoritmu	20
Obrázek 12 – Značka procesu zpracování dat.....	20
Obrázek 13 - Značka pro vstup / výstup	21
Obrázek 14 - Značka pro větvení programu.....	21
Obrázek 15 - Značka pro podproces	22
Obrázek 16 -Značka pro přípravu dat	22
Obrázek 17 - Značka pro spojku	22
Obrázek 18 - Prostředí Lego Mindstorm.....	23
Obrázek 19 - IDE Small Basic	24
Obrázek 20 - EV3 Explorer.....	25
Obrázek 21 - Zkušební pole a kalibrační čtvercová síť.....	29
Obrázek 22 - Robot EV3_1	30
Obrázek 23 - Robot EV3_2.....	30
Obrázek 24 - Obrazovka nastavení	31
Obrázek 25 - Aktivace Bluetooth.....	31
Obrázek 26 - Zadávání hesla	32
Obrázek 27 – Potvrzovací ikona	32
Tabulka 1 - Třídy Bluetooth.....	14

SEZNAM ZKRATEK A ZNAČEK

apod.	a podobně
BSIG	Bluetooth Special Interest Group
cm	centimetr
CNC	Computer Numerical Control (počítačové číslicové řízení)
EV3	Evolution 3
FHSS	Frequency Hopping Spread Spectrum
GB	Gigabyte
GHz	Gigahertz
HDD	Hard Disk (pevný disk)
HW	Hardware
I/O	Input/Output (vstup/výstup)
IC	Inter-Integrated Circuit (vnitřně integrovaný obvod)
IDE	Integrated Development Environment (vývojové prostředí)
IEEE	Institute of Electrical and Electronics Engineers (Ústav elektrotechnických a elektronických inženýrů)
in	inch (palec)
IoT	Internet of Things (internet věcí)
ISM	Industrial Scientific Medicine
kHz	Kilohertz
LCD	Liquid Crystal Display (displej z tekutých krystalů)
LED	Light_Emitting Diode (světlo emitující dioda)
Li-Ion	Lithium – Iont
m	metr
MB	Megabyte
MHz	Megahertz
Ncm	Newton-centimetr
obr.	obrázek
OS	Operační systém
PC	Personal Computer (osobní počítač)
PDA	Personal Digital Assistant (osobní digitální asistent)
R. U. R.	Rossumovi Univerzální Roboti
RAM	Random Access Memory (paměť náhodného přístupu)
RGB	Red Green Blue (červená zelená modrá)
SCL	Synchronous Clock (synchronizační hodinový pulz)
SDA	Synchronous Data (synchronizovaná data)
st. př. n. l	století před naším letopočtem
st. n. l.	století našeho letopočtu
str.	strana
SW	Software
tzv.	tak zvaný
USB	Univesal Serial Bus (univerzální sériová linka)
Wi-Fi	Wireless Fidelity (standart pro lokální bezdrátové počítačové sítě)
WPAN	Wireless Personal Area Networks (osobní bezdrátová síť malého dosahu)

ÚVOD

Cílem bakalářské práce je demonstrovat možnost spolupráce více robotických jednotek za pomoci stavebnice Lego Mindstorms EV3. V průběhu práce budou navrženi, sestaveni a naprogramováni dva roboti, kteří budou autonomně řešit zadanou úlohu a navzájem spolu budou na řešení úlohy spolupracovat.

V první kapitole se zmíním o historii samotné myšlenky neživých pomocníků až po jejich realizace a následný vývoj.

V druhé kapitole představím technologii Bluetooth, díky které se budou roboti spolu dorozumívat, představím historii její technologie, princip fungování, standardy, její využití a zabezpečení.

Ve třetí kapitole seznámím podrobněji se stavebnicí Lego Mindstorms EV3, představím hlavní řídicí jednotku (dále jen „kostka“) a ostatní periferie a senzory, kterými stavebnice disponuje. Závěr kapitoly uvádí zmínky o propojení všech částí a způsob jejich komunikace.

Tématem čtvrté kapitoly jsou základy algoritmizace. Vysvětlím, co to vlastně algoritmus je, a popíši více jeho vlastnosti, některé formy zápisu a tu nejrozšířenější formu, což jsou vývojové diagramy.

Pátá kapitola obsahuje hlavní robotickou úlohu a její cíl. Také jsou v ní uvedeny možnosti programování Lego Mindstorm EV3 a seznámení s vybraným IDE a jazykem Microsoft Small Basic.

V šesté kapitole se budu zabývat návrhem a stavbou obou robotů, problémy, které se při tom musely řešit, a výslednými parametry jednotlivých robotů.

V poslední kapitole nejprve předvedu, jak roboty spárovat. Prioritně se budu zabývat jak vytvořením jednotlivých podúloh, tak i vytvořením hlavní komplexní úlohy pro sestavené roboty. V kapitole budou postupně řazeny úlohy podle stupně obtížnosti vytvoření.

1 HISTORIE ROBOTIKY

První myšlenku a vyobrazení uměle vytvořené bytosti najdeme již v řecké mytologii. Bůh kovářství Hefaistos si podle řecké legendy vyráběl mechanické sluhly, mezi kterými byly nejen služební, ale také užitkové třínožky na vlastní pohon nebo mechanická bytost Talos, který dohlížel na Krétu. Z hlíny prý uplácal také dívku Pandóru, která byla od živé k nerozeznání.

První návrhy, podobající se robotům, se objevují ve 4. st. př. n. l. Byl to mechanický holub od řeckého matematika Archyta z Tarentu, nebo o mnoho let později, kolem roku 1495, návrh prvního robota, podobajícího se člověku, Leonarda Da Vinci [1].

Slovo robot vymyslel až Josef Čapek v roce 1920 a bylo použito ve hře R.U.R. jeho bratra Karla. Název vstoupil do všech evropských i světových jazyků. Vývoj robotů, používaných v průmyslu, začal až ve druhé polovině 20. století.

Moderní historie robotiky začíná v Americe. Inženýři Georg Devol a Joseph Engelberger začali v roce 1956 pracovat na vývoji prvního průmyslového robota, kterého pojmenovali Unimate. O pět let později nahradil pracovníky továrny v Trentonu ve státě New Jersey. Společnost General Motors jej využila pro obsluhu strojů na lítí pod tlakem, robot zvedal a skladoval horké kusy kovů ze slévárny.

V současné době se počet robotů využívaných v průmyslu, ale i v jiných oborech stále navyšuje, například v roce 2011, tedy 50 let od využití prvního průmyslového robota ve výrobním procesu, vzrůstá množství kusů ve světě meziročně o 38 % na celých 166 028 kusů [1]. Tržby z prodeje robotů dosáhly ve světovém měřítku neuvěřitelných 8,5 miliard dolarů. Dnes jsou již robotická ramena, CNC, nebo jakýkoliv jiný druh programovatelných automatických pomocníků v každé továrně a firmě a postupně nahrazují lidskou práci, které do budoucna bude potřeba stále méně.

Důležitá data v historii robotiky

1920 – Poprvé použito slovo robot ve hře R.U.R.

1956 – Vývoj prvního průmyslového robota Unimate

1961 – Unimate nahradil dělníky v továrně v New Jersey

1967 – Anglie získala licenci pro výrobu robotů

1974 – První robot FANUC v Japonsku

1983 – Automatizace výroby v Anglii a Bulharsku s roboty FANUC

2 BLUETOOTH

V druhé kapitole je popsána technologie Bluetooth, její historie, standardy, využití a možnosti zabezpečení. Tato technologie je v práci použita na výměnu dat mezi roboty, konkrétněji na poslání zprávy z jednoho robota do druhého.

2.1 Historie

Bluetooth je standard pro bezdrátovou komunikaci, který umožňuje spojení a komunikaci dvou, point-to-point, a více, point-to-multipoint, elektronických zařízení. Vytvořen byl v roce 1994 firmou Ericsson jako bezdrátová náhrada za sériové drátové rozhraní RS-232.

Historie názvu sahá až do 10. st. n. l. Dánský král Harald Blatand, jehož jméno se překládá do angličtiny jako Bluetooth – Modrý zub –, sjednotil skandinávský lid a švédská firma Ericsson si název vybrala pro historickou paralelu [2].



Obrázek 1 - Logo Bluetooth [3]

2.2 Standardy Bluetooth

Technologie Bluetooth je definována standardem IEEE 802.15. Jejím cílem je umožnění komunikace různých platform a elektronických zařízení bez velkých nároků na spotřebu energie a konfiguraci. V roce 1998 vznikla skupina SIG, jejímž cílem bylo vytvořit univerzální standard pro tzv. WPAN. První hotovou specifikaci pak SIG uveřejnila ve verzi 1.0A v červenci roku 1999.

Další jsou verze 1.1 (2002), 1.2 (2003), 2.0 (2004), 2.1 +EDR (2007), 3.0 +HS (2009), 4.0 (2010), 4.2 (2014) a poslední, známá verze 5.0 (2016). Jejimi klíčovými vlastnostmi jsou až dvojnásobná rychlost se čtyřnásobným dosahem a osmkrát větší datovou vysílací přenosovou kapacitou v porovnání s Bluetooth 4.x. Hlavní záměr využitelnosti verze spočívá v Internetu věcí neboli IoT, fenoménu posledních několika let.

Měla by být schopná propojit různá zařízení skrz celý objekt, nejen v rámci přibližně jedné místnosti, jak tomu bylo doposud. Teoretický dosah v otevřeném prostoru má být až 400 m [4]. Každá nová verze je zpětně kompatibilní s verzemi předchozími.

2.3 Využití technologie Bluetooth

Prvotním úkolem vzniku bylo zjednodušení propojení, přenosu dat mezi přenosnými zařízeními, např.: v mobilních telefonech, PDA, různých databankách apod., jejich synchronizace, zálohování s PC a celkové oproštění od kabelů.

V dnešní době se Bluetooth používá pro odlišné typy využití a snad ve všech elektronických zařízeních, na které si je možné jen vzpomenout, a kde je tato technologie vhodná.

Uvádím typy využití [5]:

- **připojení doplňků**
klasické využití Bluetooth, např. spojení mobilního telefonu a handsfree. Jedno zařízení je „aktivní“ a víceúčelové, má vlastní procesor, paměť a displej, zatímco druhé zařízení je „pasivní“, většinou jednoúčelové a je jen rozšířením nebo doplněk „aktivního“;
- **vzájemný přenos dat**
tento druh využití slouží k propojení a výměně dat mezi jakýmkoliv zařízeními podporujícími Bluetooth;
- **přístupové body**
slouží, např. k připojení notebooku na internet přes mobilní telefon, který je pomocí integrovanému hardwarovému modemu připojený k internetu;
- **spojení více zařízení**
je určené k propojení jednoho nadřazeného zařízení (master) s více podřízenými zařízeními (slave) a nazývá se point-to-multipoint. Sestava více zařízení sdílející stejný přenosový kanál se nazývá piconet. V rámci piconetu, kde je jedno zařízení nadřazené (master) a ostatní jsou podřízená (slave), může být až sedm podřízených zařízení.

2.4 Jak Bluetooth pracuje

Bluetooth pracuje v tzv. nelicencovaném pásmu ISM s frekvencí 2,4 GHz, stejně jako Wi-Fi. K přenosu využívá metody FHSS, kdy během jedné sekundy je provedeno 1600 skoků (přeladění) mezi 79 frekvencemi s rozestupem 1 MHz [6]. Tento mechanismus má zvýšit odolnost spojení vůči rušení na stejné frekvenci.

Je definováno několik výkonových úrovní, s nimiž je umožněna komunikace do vzdálenosti 1 - 400 m. Těchto udávaných hodnot je ale možné dosáhnout pouze za ideálních podmínek a ve volném prostoru. Pokud jsou mezi komunikujícími zařízeními překážky, např. zdi nebo tělo uživatele, vzdálenost dosahu rychle klesá. Většinou ovšem nedochází ke skokové ztrátě spojení, ale postupně se zvyšuje počet chybně přenesených paketů.

Modul Bluetooth se skládá ze tří základních částí [5]:

- **rádiového vysílače** – zajišťuje samotný rádiový přenos;
- **linkového ovladače** – řídí navázání spojení, identifikaci, přístup a samotnou komunikaci;
- **správce linky a I/O obvodů spoje** – zajišťuje komunikaci se zařízením.

2.5 Třídy a dosah

Jednotlivá zařízení vybavená technologií Bluetooth se řadí do čtyř tříd, podle vysílacího výkonu a dosahu (viz Tabulka 1).

Třída	Maximální povolený výkon		Dosah (přibližný)
	mW	dBm	
Třída 1	10	20	~100 metrů
Třída 2	2,5	4	~10 metrů
Třída 3	1	0	~1 metr
Třída 4	0,5	-3	~0,5 metru

Tabulka 1 - Třídy Bluetooth, upraveno podle [2]

S větším dosahem výrazně stoupá i energetická náročnost. Pozornost se musí věnovat faktu, v jakých třídách jsou jednotlivá zařízení schopna vysílat. Možná vzdálenost komunikace se vždy rovná jen maximální možné vzdálenosti komunikace slabšího zařízení.

3 LEGO MINDSTORMS

3.1 Inteligentní kostka EV3

3.1.1 Popis

Je to vlastně malý počítač, který hraje roli mozku stavebnice Lego Mindstorms. Kostku pohání procesor ARM9, taktovaný na 300 MHz, má 64 MB RAM a 16 MB úložné paměti, rozšiřitelné paměťovou kartou microSD až na 32 GB. Na tomto HW „běží“ upravený OS Linux [7].

Kostka umožňuje čtení dat z čidel, ovládání motorů, vykreslování na LCD, blikání dvoubarevnými LED diodami, přehrávání zvuků z reproduktoru, reakci na stisknutí tlačítek, zápis nebo čtení z microSD karty a přes USB komunikaci s PC. Kostka disponuje i rozhraním Bluetooth a po připojení USB Wi-Fi adaptéru rozhraním Wi-Fi. Přes tyto bezdrátové sítě může být kostka ovládána smartphony nebo komunikovat s PC a s dalšími EV3 kostkami.

3.1.2 Vzhled

Kostka (viz obr. 2) má rozměry přibližně 11 x 6,5 x 5 cm. Na přední straně se nachází monochromatický LCD displej s rozlišením 178x128 pixelů a pod ním šestice tlačítek pro ovládání. Z toho je, pro průsvit dvoubarevných LED diod (zelených a oranžových), pět tlačítek uprostřed průhledně orámovaných. Tlačítka slouží primárně pro zapnutí kostky, pohyb v menu a ovládání a programování kostky bez PC, nebo sekundárně pro ovlivňování chodu programu. Osamocené tlačítko vlevo pod LCD slouží pro vrácení v menu, ukončení běžícího programu, nebo vypnutí kostky.

Na levé straně je slot na microSD a USB pro připojení Wi-Fi. Na straně pravé je jen reproduktor. Na spodní straně se nachází čtveřice konektorů označených 1 až 4 pro připojení senzorů. Na straně horní se nachází mini USB konektor pro připojení k PC a čtveřice I/O konektorů označených A až D pro ovládání motorů a snímání jejich otáčení. Zespuhu je místo pro vložení velkého Li-Ion akumulátoru se vstupem pro nabíjecí konektor a indikačními LED diodami, nebo šestici tužkových baterií typu AA.



Obrázek 2 - EV3 kostka [8]

3.2 EV3 servomotory

Všechny servomotory obsahují senzor rotace s rozlišením jeden stupeň. Velký motor obsahuje převodovku, může dosáhnout až 160-170 otáček za minutu a maxima 20 Ncm točivého momentu (viz obr.4). Menší servomotor je připojen na přímo a dosahuje 240-250 otáček za minutu s 8 Ncm točivého momentu (viz obr. 3). Motory, pokrývají úchyty kompatibilní se zbytkem stavebnice Lego a zezadu motorů konektorem pro připojení ke kostce.



Obrázek 4 - Primární servomotor [9]



Obrázek 3 - Sekundární servomotor [10]

3.3 Senzory

3.3.1 Dotykový senzor

Má jedno červené tlačítko nabývajících hodnot 1 nebo 0 (viz obr. 5).



Obrázek 5 - Dotykový senzor [11]

3.3.2 Barevný senzor

Senzor, který dokáže rozlišovat barvy, má podobný tvar a velikost jako senzor dotykový, avšak místo červeného tlačítka je umístěna RGB LED dioda a pod ní samotný senzor snímající barvu.



Obrázek 6 - Barevný senzor [12]

3.3.3 Gyroskopický senzor

Dokáže měřit rotační pohyb a identifikovat změny v jeho natočení, dokáže měřit maximálně 440 otáček za sekundu a úhel otočení, měřit s přesností ± 3 stupně (viz obr. 7).



Obrázek 7 - Gyroskopický senzor [13]

3.3.4 Ultrazvukový senzor

Generuje ultrazvukové vlnění, které následně zas přijímá. Vyhodnocením času od vyslání signálu po jeho přijetí dokáže změřit vzdálenost od nejbližší překážky umístěné před senzorem. Zjištěná vzdálenost může být zaznamenána v centimetrech v rozmezí 1 - 255 cm, nebo v palcích (1 - 100 in) s přesností 1 cm (0,394 in). Dále umožňuje odesílat jednotlivé zvukové vlny, a tak dokáže fungovat jako sonar. Může fungovat také v módu čekání na zvuk, který spustí program (viz obr. 8). [14]



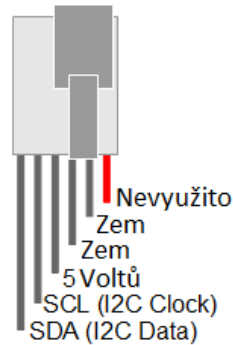
Obrázek 8 - Ultrazvukový senzor [15]

3.3.5 Sensory třetích stran

Stavebnice Lego Mindstorms EV3 udržuje zpětnou kompatibilitu s předchozí řadou NXT, je proto možné využít jak komponent stavebnice Lego NXT, tak i komponent externích producentů nabízených na minulou řadu kostek, např.: tlakový senzor (NXT Force Sensor), optický detektor vzdálenosti (NXT EOPD), magnetický senzor, barometrický senzor atd. [16]

3.4 Propojení

Všechny senzory i motory jsou připojeny stejným plochým šestižilovým kabelem zakončeným na obou stranách stejným šestipinovým konektorem kompatibilním i s NTX kostkami (viz obr. 9). Toto rozhraní obsahuje jak napájení senzorů a motorů tak komunikační dvoužilovou I²C sběrnici.



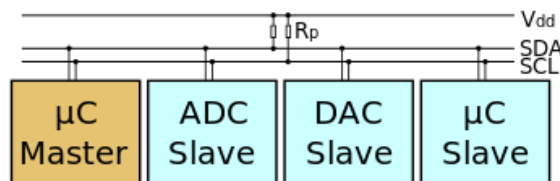
Obrázek 9 - Zapojení konektoru, upraveno podle [18]

I²C sběrnice rozděluje připojená zařízení na:

- řídicí (master) – zahajuje a ukončuje komunikaci; generuje hodinový signál SCL
- řízené (slave) – zařízení adresované masterem.

Vlastnosti sběrnice I²C:

- umožňuje propojení až 128 různých zařízení
- umožňuje propojení pomocí dvou obousměrných vodičů (jeden tvoří hodinový signál SCL a druhý datový kanál SDA)
- maximální frekvence signálu SCL je podle verze I²C 100 kHz nebo 400 kHz.
- neumožňuje duplexní přenos (v jednom okamžiku vysílá jen jedno zařízení)
- zařízení připojená na sběrnici musí mít individuální adresu o délce 7 nebo 10 bitů. [17]



Obrázek 10 - Příklad zapojení sběrnice [19]

4 ZÁKLADY ALGORITMIZACE

S algoritmy se například setkáváme při pročítání návodů, receptů z kuchařské knihy nebo při prostém popisu nějaké činnosti.

Pro správné pochopení a srozumitelnost musí algoritmy splňovat níže uvedené podmínky:

- **elementárnost** – algoritmus se skládá z konečného počtu jednoduchých, tzv. elementárních kroků;
- **konečnost** – každý algoritmus musí skončit v konečném počtu kroků, který může být libovolně velký;
- **obecnost (univerzálnost)** - algoritmus nesmí řešit jen jeden konkrétní problém, ale obecnou třídu obdobných problémů, kdy má širokou množinu možných vstupů;
- **determinovanost** – každý krok uvedený v algoritmu musí být jednoznačně a přesně definován. V každém kroku musí být jasné, co a jak se má provést a jak má provádění algoritmu pokračovat;
- **resultativnost** – algoritmus musí mít alespoň jeden výstup. Říkáme, že algoritmus vede od zpracování hodnot k výstupu;
- **opakovatelnost** – algoritmus musí při opakovaném zadání stejných hodnot dospět ke stejnému výsledku. [20]

4.1 Možnosti zápisu algoritmu

Při tvorbě algoritmu je třeba dbát na jeho srozumitelnost a přehlednost pro pozdější implementaci nebo případné úpravy. Je třeba si uvědomit, pro koho je zápis algoritmu určen, a podle toho správně vybrat způsob interpretace.

K zápisu algoritmů je možné použít některou z možností:

- **slovní** – jakýkoliv slovní popis jakékoliv činnosti nebo úkonu, který splňuje výše uvedené podmínky. Výhodou je, že se tak lze domluvit v daném oboru i s laikiem. Nevýhodou je, že není moc přehledná;
- **matematická** – hodí se tam, kde je možné řešenou problematiku popsat pomocí matematických vztahů. Výhodou je jednoznačnost. Nevýhodou je potřeba znalosti matematických zápisů a jejich malá podrobnost;

- **pomocí rozhodovacích tabulek** – jsou velmi vhodné v případech, kdy se v úloze vyskytuje několik možností a řešení pro každou možnost je jednoduše popsitelné, např. školní rozvrh. Výhodou je jednoznačný, přehledný a snadno pochopitelný zápis, nevýhodou vhodnost jen pro určitý typ úloh;
- **použití vývojových diagramů** – jsou popsána v další podkapitole. [21]

4.2 Vývojové diagramy

Jde o symbolický algoritmický jazyk, který se používá pro názorné zobrazení algoritmu. Je to jedna z nejdokonalejších a nejrozšířenějších forem zápisu algoritmů vhodná pro vývoj SW, hlavně jako komunikační prostředek při týmové spolupráci analytiků s programátory a k dokumentačním účelům.

4.2.1 Mezní značky vývojových diagramů

Mezní značka představuje vstup z vnějšího prostředí do programu nebo naopak výstup z něj do vnějšího prostředí. Jednoduše řečeno *začátek* a *konec* programu (viz. obr. 11).

Začátek / konec



Obrázek 11 - Značka pro začátek a konec algoritmu, upraveno podle [22]

4.2.2 Sekvenční bloky vývojových diagramů

Sekvenční bloky se vyskytují uvnitř programu, mezi začátkem a koncem. Označují sekvenční postup algoritmu a v jejich průběhu nesmí dojít k rozvětvení algoritmu;

- **zpracování**

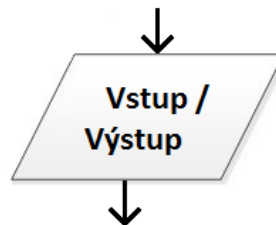
znázorňuje nějakou činnost programu, během níž dochází k transformaci dat, např. sečtení dvou čísel (viz. obr. 12);



Obrázek 12 – Značka procesu zpracování dat, upraveno podle [22]

- **vstup nebo výstup**

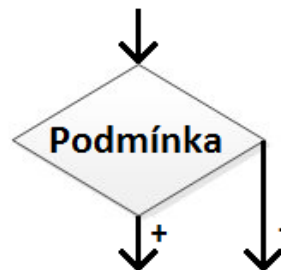
slouží ke komunikaci s počítačem nebo uživatelem v běhu programu. Vstup může například načítat data z klávesnice nebo datového souboru uloženého třeba na HDD. Výstup nám naopak může zobrazit výsledky zpracování programu, např. na monitoru, nebo výsledek může zase třeba uložit do souboru na HDD (viz. obr. 13).



Obrázek 13 - Značka pro vstup / výstup, upraveno podle [22]

4.2.3 Větvení vývojových diagramů

Sekvenční bloky nám, až na nějaké triviální operace, stačit nebudou. Je potřeba, aby mohl program reagovat na měnící se hodnoty a dále se větvit. Větvení však nemůže probíhat nahodile, ale vždy pouze na základě nějaké podmínky. Při plnění podmínky se program ubere cestou + a v případě nesplnění cestou - (viz. obr. 14).

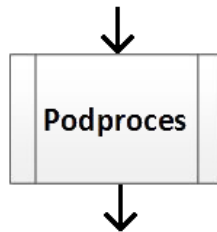


Obrázek 14 - Značka pro větvení programu, upraveno podle [22]

4.2.4 Další značky

- **podproces**

znázorňuje samostatnou část programu, která může obsahovat větší množství kroků. Používá se hlavně v částech, které by se musely několikrát zbytečně opakovat, a tím by se diagram a program staly nepřehlednými (viz. obr. 15);



Obrázek 15 - Značka pro podproces, upraveno podle [22]

- **příprava**

označuje přípravnou fázi programu, např. se užívá pro zahájení cyklů o známém počtu opakování (viz. obr. 16);



Obrázek 16 - Značka pro přípravu dat, upraveno podle [22]

- **spojka**

umožňuje spojit dvě části vývojového diagramu, které nebylo možné napsat souvisle (viz. obr. 17).

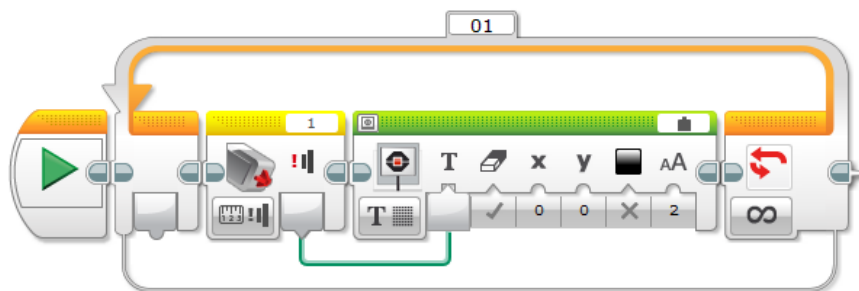


Obrázek 17 - Značka pro spojku, upraveno podle [22]

5 TVORBA ROBOTICKÉ ÚLOHY

Před samotnou stavbou a programováním robotů je důležité si vymežit problém zpracovávaný úlohou. V hlavní robotické úloze jde konkrétně o to, aby robot EV3_2 projel ohraničeným bludištěm a do ničeho nenarazil. Problém je v tom, že je „slepý“, čili že nedisponuje žádnými čidly, která by mu v tom pomohla. Proto mu robot EV3_1 musí pomoci. Bludištěm za pomoci čidel, kterými disponuje, nejprve sám projede a pak robotu EV3_2 přes Bluetooth zašle jakousi „mapu“ neboli zprávu s instrukcemi, podle které bludištěm robot EV3_2 již snadno projede.

Po seznámení se stavebnicí a použitými technologiemi je nutné vymyslet, jakým způsobem a v jakém prostředí bude nejlepší a nejsnazší roboty programovat. Jako první se nabízí možnost programování v prostředí, které je k této stavebnici oficiálně vydáno od Lega na jeho stránkách; je to grafické IDE, v kterém se program skládá pomocí různých bloků rozdělených do různých kategorií (viz. obr. 18). Úloha by v tomto prostředí šla jistě také sestavit, ale z důvodu nepřehlednosti a neohrabanému způsobu pohybu nad otevřeným projektem jsem tuto možnost zavrhl a rozhodl jsem se podívat po jiné alternativě.



Obrázek 18 - Prostedí Lego Mindstorm [23]

Našel jsem plugin do IDE Microsoft Visual Studio. Programování bylo sice již značně příjemnější a přehlednější, ale problém nastal v tom, že se program po stisku tlačítka kompilace přímo odeslal do kostky a zrovna provedl, bez možnosti jeho uložení do souboru, který se pak da spustit přímo z kostek. Pro 2 roboty bych tedy potřeboval 2 počítače, což by bylo zbytečně složité.

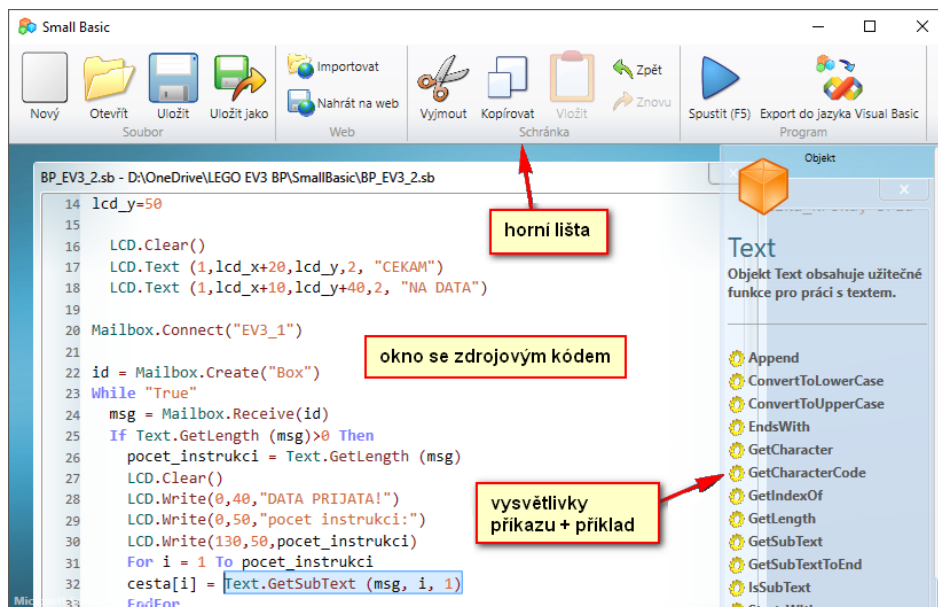
Dále jsem našel ještě několik dalších možností, od kompilátorů pro Java, C++ a nebo možnosti naboťování upraveného firmwaru, který pak zvládal kompilaci různých jazyků přímo v kostce apod. Nejvíce mě ale zaujal projekt EV3 Basic.

5.1 EV3 Basic

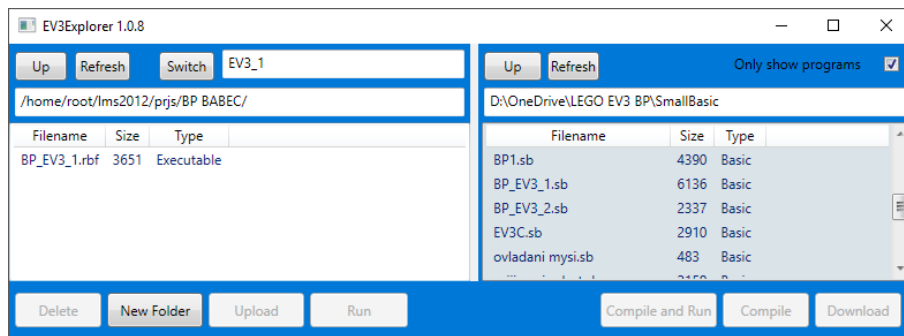
EV3 Basic je plugin pro IDE a stejnojmenný jazyk Small Basic od Microsoftu. Small Basic je velice podobný ostatním programovacím jazykům, stačí se naučit jen několik příkazů, jinak konstrukce a programování obecně jsou skoro stejné (viz. obr. 19).

Má velice jednoduchou syntaxi, nerozlišuje malá a velká písmena, nepoužívá se moc speciálních znaků a proměnným se nemusí přiřazovat datové typy, ty si program přiřadí sám, až podle obsahu v proměnné. IDE také práci velice usnadňuje aktivní nápovědou a našeptáváním při psaní kódu. Jedním stiskem je také možno zálohovat program do jakéhosi cloudu a získat na něj jedinečný identifikátor a odkaz, kterým může vývojář lehce sdílet svůj program třeba se svým kolegou.

Své aplikace můžete také později exportovat do pokročilejšího VB.NET. Po doinstalování pluginu nám v prostředí přibudou třídy a funkce pro obsluhu Lego EV3 kostky, možnost spuštění programu, jak přes Bluetooth, Wi-Fi, a nebo přes kabel USB, přímo z IDE a také přibude nejdůležitější vlastnost, a to miniaplikace EV3Explorer (viz. obr. 20, str. 25), v které se dá spravovat obsah v kostkách, a hlavně kompilovat soubory *.sb* (Small Basic) do souborů *.sbf*, které se následně uloží do paměti EV3 a jsou z kostek i následně spustitelné.



Obrázek 19 - IDE Small Basic, vlastní



Obrázek 20 - EV3 Explorer, vlastní

5.2 Seznámení s programovacím jazykem

I když je jazyk Small Basic poměrně jednoduchý, rozhodl jsem se jeho jednotlivé části, a hlavně ty, které byly mnou použity k vytvoření robotických úloh, popsat a ukázat jejich použití na jednoduchých příkladech. V místech, kde to mělo jakousi vypovídající hodnotu, jsou příklady doplněny o komentáře s vysvětlivkami (text napsaný kurzívou za apostrofem ');

- **proměnné a pole**

```
a = 5 'přiřazení hodnoty do proměnné a
a = a + 1 'inkrementace hodnoty proměnné a
a = a - 1 'dekrementace hodnoty proměnné a
B[1] = 55 'přiřazení hodnoty 55, do pole B pod index 1
x = "hi" 'uložení textu do proměnné x
```

- **sub (procedura)**

procedura či podprogram je ohraničený kód, který se vykoná, když funkci někde v programu zavoláte. Používá se při opakování kódu na více místech:

```
Sub idPodprogramu 'definování podprogramu
'kód podprogramu
EndSub 'ukončení podprogramu
```

```
idPodprogramu() 'volání podprogramu
```

- **cyklus For**

For je cyklus s pevně stanoveným počtem opakování:

```
For i=1 to 5 'začátek cyklu For, který se provede 5 krát
TextWindow.WriteLine("I:"+I) 'vypsání I:1 až I:5
EndFor 'konec cyklu For
```

- **cyklus While**

cyklus `While` nám opakuje kód, dokud platí podmínka; je s podmínkou na začátku.

Tento programovací jazyk ani žádné jiné cykly nepodporuje:

```
while i>3 'cyklus s podmínkou na začátku
i=i+1
TextWindow.WriteLine("I:"+i)
endwhile 'ukončení cyklu s podmínkou na začátku
```

- **větvení za pomoci If, Then, ElseIf, Else**

podmínky zapíšeme mezi příkazy `If` a `Endif`. Pro vícenásobné větvení s více podmínkami se používá `ElseIf`. Dále můžeme použít příkaz `Else`, který se provede, není-li splněna ani jedna z podmínek:

```
If i = 5 Then 'první podmínka
    TextWindow.WriteLine("i = 5")
ElseIf i > 5 Then 'druhá podmínka
    TextWindow.WriteLine("i > 5")
ElseIf i < 5 Then 'třetí podmínka
    TextWindow.WriteLine("i < 5")
Else 'provede se když předchozí nejsou splněny
    TextWindow.WriteLine("Nezadal jsi číslici. ")
EndIf 'konec podmínek
```

v podmínce můžeme používat následující operátory:

```
=      'rovná se
<>    'nerovná se
>      'větší než
<      'menší než
>=    'větší, nebo rovno
<=    'menší, nebo rovno
```

logické výrazy můžeme v podmínkách skládat pomocí operátorů `AND` a `OR`. Ty označují

"a zároveň" a "nebo", např. `If a > 5 AND a < 13 Then`

- **skok Goto**

```
idSkoku: 'deklarace skoku
TextWindow.WriteLine("Toto je nekonečný cyklus")
Program.Delay(40)
Goto idSkoku 'volání skoku
```

[24]

5.3 Příkazy na obsluhu motorů

Na ovládání motorů je mnoho příkazů a možností, já ve své práci využívám jen jeden, a to `Motor.MoveSync("porty", rychlost_1, rychlost_2, stupně, "brzda")`. Za `porty` se do uvozovek dosadí dvě písmena portů, do kterých jsou zapojeny hlavní motory, za `rychlost_1` dosadíme rychlost prvního motoru a za `rychlost_2` rychlost druhého. Při rovném pohybu kupředu nebo vzad zvolíme stejné hodnoty a při otáčení se na místě dáme před jednu z hodnot znaménko mínus. Poslední parametr `brzda` nám ovlivňuje, zda se motory po dokončení úkonu uvedou do klidového stavu. Nabývá logické hodnoty *true* nebo *false*. [25]

5.4 Příkazy pro komunikaci přes Bluetooth

Příkazem `Mailbox.Connect("bleetooth_identifikátor_druhé_kostky")` se kostka pokusí o navázání spojení s druhou kostkou, s kterou chceme komunikovat. Je k tomu jen zapotřebí znát název zařízení Bluetooth druhé kostky a tento název zadat do parametru funkce. Zprávy je možné posílat, jen pokud bylo navázáno připojení, buď výše uvedenou funkcí, nebo ručně v nastavení kostky.

Dalším příkazem `Mailbox.Create("název_schránky")` si vytvoříme schránku, do které pak chodí zprávy poslané druhou kostkou. Bez vytvoření schránky zprávy přijímat nelze. Na jedné kostce lze vytvořit až 30 schránek a návratová hodnota funkce je identifikační číslo schránky, které jí je přiděleno. Je potřeba si ho uložit do proměnné, protože ho dále budeme ještě potřebovat (např.: `id=Mailbox.Create("název_schránky")`).

Příkaz `Mailbox.IsAvailable(id)` zkontroluje, zda schránka obsahuje nějakou zprávu. Parametr obsahuje proměnnou `id` s identifikačním číslem schránky, kterou jsme si vytvořili o krok výše. Návratová hodnota funkce, pokud zpráva existuje, je *true*. Jinak vrací *false*. V případě došlé zprávy je návratová hodnota funkce `Mailbox.Receive(id)` text přijaté zprávy.

Parametr obsahuje proměnnou `id` jako již v předcházejícím odstavci. Poslední funkcí v této kategorii je funkce `Mailbox.Send("Bluetooth_id_druhé_kostky", "název_schránky", "zpráva")`, která odešle textovou zprávu do schránky jiné kostce, nebo kostkám, s kterými byla navázána komunikace. První parametr je Bluetooth identifikátor kostky, na kterou chceme zprávu odeslat. V případě, že je zadán prázdný text, tak se zpráva odešle všem kostkám, s kterými je zrovna navázané připojení. Druhý parametr označuje cílovou schránku a v posledním je samotný text zprávy. [25]

5.5 Příkazy na obsluhu senzorů

Nejprve je potřeba senzory deklarovat a nastavit. Úkon se dělá hned na začátku příkazem `Sensor.SetMode(číslo_portu, režim_senzoru)`, kde parametr `číslo_portu` označuje číslo fyzického portu na kostce, do kterého je zapojen, a za `režim_senzoru` dosadíme číslo podle režimu, v jakém ho chceme používat. V případě ultrasonického dálkoměru je to 0 pro vzdálenost v mm, tu jsem použil, a 1 pro vzdálenost v palcích. U senzoru rozpoznávání barev je 0 pro zjištění hodnoty odraženého světla od podložky, 1 pro intenzitu okolního osvětlení, 2 pro přímé rozpoznání standardních barev (0=neznámá, 1=černá, 2=modrá, 3=zelená, 4=žlutá, 5=červená, 6=bílá, 7=hnědá), tu jsem použil, a 4 pro hodnotu barvy v RGB.

Čtení hodnot ze senzorů se provádí za pomoci příkazu `Sensor.ReadRawValue(číslo_portu, index)`, kde `číslo_portu` již je dobře známé a `index` je zpravidla až na výjimky v dodatcích u senzorů 0. V určitých případech je možné použít příkaz `Sensor.ReadPercent(číslo_portu)`, která vrací hodnotu ze senzoru v procentech v rozmezí 0 až 100. [25]

5.6 Display a tlačítka

Na čtení stavu stisknutí tlačítek mně stačila funkce `Buttons.Current`, která vrací písmeno/a (U=horní, D=dolní, L=levé, R=pravé, E=prostřední) označující stisknutou/té klávesu/y ve formě textového řetězce.

Na obsluhu LCD stačila funkce `LCD.Clear()`, která smaže celý display a funkce `LCD.Text(barva, x, y, velikost, "text")`. U parametru `barva` je 1 pro černý text na světlém pozadí a 0 pro invertované zobrazení. Další dva parametry `x` a `y` znamenají souřadnice, odkud se začne text vypisovat, `velikost` určuje velikost fontu a jako poslední je samotný text zavřený do uvozovek. [25]

5.7 Textové funkce

Poslední podkapitolou jsou části kódu, které mně pomohly k sestavení jednotného řetězce a následného rozparsování (cílený převod z textové podoby na nějaký specifický typ) do instrukcí pro druhého robota. První z těchto funkcí je `Text.Append("text_1", "text_2")`, která má za úkol skládání dvou textových řetězců do jednoho. Další je funkce `Text.GetLength("text")`, která vrací počet písmen textového řetězce a jako poslední je textová funkce `Text.GetSubBluetoothext("text", pozice, délka)`, která nám z textového řetězce dokáže vrátit jen požadovaný počet písmen od stanovené pozice v textu, v mém případě se pozice vždy při opakování cyklu o jedno zvýšila a délka byla nastavena na 1. [25]

6 KONSTRUKCE ROBOTŮ

Zkonstruování hlavního sensorického robota, o kterém se dále budu zmiňovat jen jako o EV3_1 (viz obr. 22, str. 30), se ukázalo poněkud složitější, než jsem původně zamýšlel. Tato, i když vospělá robotická stavebnice, je ale stále jen hračka. Snažil jsem se všemožnými způsoby chyby co nejvíce eliminovat, avšak byl to „boj s větrnými mlýny“. Rád bych popsal některé „slepé uličky“, do kterých jsem se dostal:

Základní stavebními prvky přesnosti úspěchu je co nejrovnější přímočarý pohyb a co nejvíce kolmé zatáčení. Má první konstrukce hlavního robota řešila zatáčení přes gyroskopický senzor a měla jeden senzor vzdálenosti na své přední straně. Konstrukce se ukázala jako nešťastná z důvodu, že se robot musel často otáčet, aby zjistil, jak vypadá svět kolem něj. Přesnost kolmosti zatáčení se po každém natočení zmenšovala a záznam trasy byl po delší jízdě nepoužitelný.

Nepřesnosti vznikaly především chybou gyroskopu a vůlí převodovky v hlavních motorech. Pokusil jsem se tedy co nejvíce eliminovat otáčení robota umístěním ultrazvukového senzoru na další motor, který s ním podle potřeby otáčel a mapoval tak terén kolem. To již situaci značně zlepšilo, ale robot byl příliš dlouhý pro mé zkušební pole (viz obr. 21).



Obrázek 21 - Zkušební pole a kalibrační čtvercová síť, vlastní

Stejně tak se pro pohyb ukázala lepší kolečka umístěné co nejbližší středu robota. Na otáčení je tedy zapotřebí menší síly a také je i menší riziko proklouznutí. Ze stejného důvodu jsem nezvolil pásový podvozek. Navíc se s pásovým podvozkem šířka robota výrazněji zvětší.



Obrázek 22 - Robot EV3_1, vlastní

Robot je také vybaven barevným senzorem umístěným několik mm nad povrchem. Výsledné rozměry EV3_1 jsou na délku 14 cm, na šířku 12 cm a na výšku 12 cm s tím, že překážky před sebou zvládne zachytit již od 3 cm a po stranách od 6 cm nad povrchem.

U druhého robota EV3_2 (viz obr. 23) již nároky na konstrukci tak velké nebyly. Byl vlastně jediný, měl mít podobné rozměry jako EV3_1. Jeho rozměry jsou na délku 14 cm, na šířku 11 cm a na výšku 14 cm, což jsem vcelku zachoval.



Obrázek 23 - Robot EV3_2, vlastní

7 ROBOTICKÉ ÚLOHY

Hlavní úlohu jsem se rozhodl rozdělit do několika menších podúloh. Ve stejných blocích jsem tvořil i samotný algoritmus a zdrojový kód hlavní úlohy. Hlavní úloha i podúlohy budou vždy obsahovat tyto části:

- **popis**
- **moduly**
- **program**
- **shrnutí**

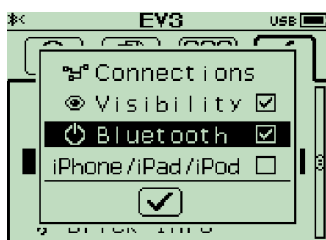
Nejprve jsem musel oba roboty spárovat přes Bluetooth, což se provede následujícím způsobem:

přejdeme do nastavení na ikonu klíče a otevřeme nastavení Bluetooth (viz. obr. 24);



Obrázek 24 - Obrazovka nastavení [26]

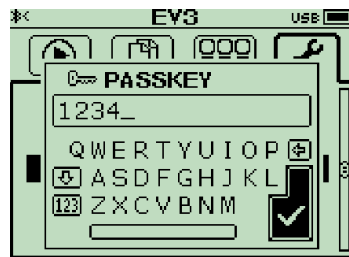
v tomto nastavení se zaškrtnou první dvě zaškrťovací políčka (viz. obr. 25), to první nám umožní zařízení navzájem vyhledat a druhé samotné Bluetooth aktivuje;



Obrázek 25 - Aktivace Bluetooth [27]

po provedených dvou krocích práce na obou robotech přejdeme na položku *Connections*, stačí jen na jednom robotu, kde klikneme na *Search* a spustíme vyhledávání ostatních zařízení. Po chvíli hledání se načte seznam, na kterém uživatel vybere název druhého zařízení, potvrdí, že

se k němu chceme připojit, a to samé provedeme i na druhém zařízení. Dále již jen stačí na obou zařízeních zadat stejné heslo do okna které se objeví (viz. obr. 26);



Obrázek 26 - Zadávání hesla [28]

podle ikony, která se objeví v levém horním rohu (viz. obr. 27), poznáme, že párování je zdárně u konce.



Obrázek 27 – Potvrzovací ikona , upraveno podle [28]

Na str. 29 jsem se zmiňoval, že u těchto úloh je velmi důležité, aby se roboti pohybovali co nejvíc přímočaře a jejich zatačení bylo co nejménější. Proto bylo potřeba pro každého robota vytvořit jakousi kalibraci jeho jízdnicích vlastnosti a v kódu co nejvíce vyladit parametry odpovědné za tyto pohyby. Především jde o parametry, které ovlivňují o kolik stupňů se jednotlivá kolečka v jednom kroku pootočí. Přesnost ovlivňuje i rychlost, jakou se tento krok vykoná, nebo zda je zapnutá nebo vypnutá „brzda“ na konci kroku. Kalibraci jsem prováděl pomocí čtvercové sítě, kterou jsem si narýsoval z druhé strany zkušebního pole (viz. obr. 21, str. 29).

Příklad nastavení a subprocedury pro jeden krok pohybu vpřed u EV3_1:

```
pohon_rychlost = 10  
delka_kroku = 30  
brzda = "true"
```

```
Sub pohon_vpřed  
Motor.MoveSync(pohon_porty, pohon_rychlost, pohon_rychlost, delka_kroku,  
brzda)  
EndSub
```


7.1 Zaslání zprávy z jednoho robota do druhého

popis: Úkolem je navázání bezdrátové komunikace a odeslání zprávy z kostky EV3_1 do EV3_2;

moduly: v této úloze není zapotřebí žádného modulu, technologie Bluetooth je již integrována;

program: výsledkem jsou dva programy pro dva roboty. Program pro robota EV3_1 se jen pokusí o navázání připojení k robotu EV3_2 a následně odešle krátkou zprávu: „Ahoj, světe!“. Zatímco program pro robota EV3_2 má na starosti přijetí zprávy. Vytvoří schránku „BOX“ a následně se aktivuje nekonečná smyčka, která neustále maže display a vzápětí zobrazuje obsah schránky. Je důležité, aby byl program na robotu EV3_2 spuštěn dříve než na robotu EV3_1, jinak se příklad nezdaří;

shrnutí: v této triviální úloze jsem ukázal práci s propojením více robotů a jejich jednoduchou komunikaci. Důležité je dodržet postup a zkontrolovat Bluetooth připojení na obou robotech.

Příklad pro odeslání zprávy:

```
Mailbox.Connect("EV3_2")
Mailbox.Send("EV3_2", "Box", "Ahoj světe!")
```

Příklad pro přijetí zprávy:

```
id = Mailbox.Create("Box")
While "True"
  LCD.Clear()
  LCD.Text(1,0,0,1Mailbox.Receive(id))
EndWhile
```

7.2 Rozparsování zprávy na pokyny pro robota

popis: Po přijetí zprávy přes Bluetooth se přijatá zpráva rozloží na jednotlivá písmena, které znamenají jednotlivé kroky pro pohyb robota EV3_2;

moduly: v úloze jsou použity dva moduly, a to motory zapojené na porty A a D;

program: program je z důvodu místa ořezán jen na nejdůležitější části a ani zde již neopakují kód pro odeslání zprávy, lišil by se pouze ve zprávě, kterou bych pro tento příklad odesílal, což by mohla být jakákoliv dlouhá kombinace písmen V, L a P. Část pro vytvoření schránky a smyčky také znám. Přibyla v ní podmínka, zda schránka obsahuje zprávu funkcí Mailbox.IsAvailable.

Je-li naplněna, tak se změní délka zprávy funkcí Text.GetLength. Na ni navazuje cyklus For, který se provede tolikrát, kolik je znaků ve zprávě a jednotlivé znaky se pak znak po znaku textovou funkcí Text.GetSubBluetoothext uloží do pole cesta. Program dále přejde přes Goto na návěstí *start*, kde následuje další cyklus For, který celé pole znak po znaku přečte a pomocí vícenásobné podmínky If reaguje na přečtené znaky příslušnou akcí. V je posun vpřed, P je kolmé otočení doprava a L je kolmé otočení doleva;

shrnutí: tímto příkladem jsem demonstroval, jak je možné přijatý textový řetězec rozložit na jednotlivé úkony pro pohyb robota;

Příklad rozparsování zprávy:

```
Mailbox.Connect("EV3_1")
id = Mailbox.Create("Box")
While "True"
  If Mailbox.IsAvailable (id) = "true" Then
    msg = Mailbox.Receive(id)
    pocet_instrukci = Text.GetLength (msg)
    For i = 1 To pocet_instrukci
      cesta[i] = Text.GetSubBluetoothext (msg, i, 1)
    EndFor
    Goto start
  EndIf
EndWhile
```

start:

```
For i = 1 To pocet_instrukci
  If cesta[i] ="V" Then
    pohon_vpřed()
  ElseIf cesta[i] ="L" Then
    pohon_doleva()
  ElseIf cesta[i] ="P" Then
    pohon_doprava()
  EndIf
EndFor
```

7.3 Společné zvládnutí zastavení před překážkou nebo na konci pole

- popis:** Úkolem je, aby robot EV3_1 jel rovně tak dlouho, dokud se před ním neobjeví překážka ve vzdálenosti od předního senzoru menší než 4 cm, nebo okraj ohraničeného pole. Dále musí každý jednotlivý krok, který udělá, zaznamenat do textového řetězce a následně po zastavení a potvrzení klávesou Enter zašle druhému robotu;
- moduly:** v této úloze jsou použity již čtyři moduly, a to motorů zapojených na porty A a D, senzor vzdálenosti připojený na port 2 a barevný senzor na portu 3;
- program:** program začíná spuštěním procedury `vpredSub()`, ve které je podmínka reagující na přečtenou vzdálenost z předního ultrazvukového senzoru a barvy pod robotem. Je-li vzdálenost větší než 4 cm a zároveň barva pod robotem bílá, robot se o krok posune kupředu a zároveň procedura znovu sama sebe spustí. Operace se opakuje, dokud není jedna z podmínek porušena. V případě porušení podmínky se spustí procedura `konec()`, která pak smyčkou hlídá stav tlačítka Enter a po jeho stisku odešle obsah proměnné `zaznam_cesty` do robota EV3_2, který zadané instrukce provede. Samotný záznam cesty je pak řešen přímo v proceduře `pohon_vpred()`, jež vykoná samotný pohyb robota textovou funkcí `Text.Append` a jež skládá dva textové řetězce do jednoho;
- shrnutí:** příklad již demonstruje opravdovou kooperaci dvou zkušebních robotických jednotek, které až na potvrzení odeslání zprávy fungují již zcela autonomně.

Příklad zastavení před překážkou nebo na konci pole:

```
vpredSub()
Sub vpredSub
If Sensor.ReadRawValue(sensor_predek, 0)/10 > 4 OR
Sensor.ReadRawValue(sensor_barva, 0) = 6 Then
    pohon_vpred()
    vpredSub()
Else
    konec()
EndIf
EndSub

Sub konec
    promenna = 0
    While promenna = 0
        If Buttons.Current = "E" Then
            promenna = 1
        EndIf
    EndWhile
    Mailbox.Connect("EV3_2")
    Mailbox.Send("EV3_2", "Box", zaznam_cesty)
    Program.End()
EndSub

Sub pohon_vpred
    Motor.MoveSync(pohon_porty, pohon_rychlost, pohon_rychlost,
    delka_kroku, brzda)
    zaznam_cesty = Text.Append(zaznam_cesty, "V")
EndSub
```

7.4 Společné zvládnutí projetí bludištěm

- popis:** V této části se dostávám k hlavní komplexní úloze. Jejím cílem je projetí robota EV3_1 bludištěm za pomoci svých senzorů. Po dokončení úkolu uživatel potvrdí odeslání zprávy robotu EV3_2.
- moduly:** jsou použity již všechny moduly, a to moduly dvou motorů zapojených na porty A a D, tři senzory vzdálenosti připojené na porty 1,2 a 4 a barevný senzor připojený na port 3.
- program:** hned na začátku programu je cyklus, který usnadňuje start programu a umístění robota na startovní pozici v bludišti. Podmínka v něm kontroluje, zda je pod robotem bílá barva, tudíž se hlavní část programu spustí až poté, kdy je robot umístěn na podložku. Poté se přejde k již známé proceduře vpredSub(), která však byla od uvedených příkladů upravena, a hlídání vzdálenosti od překážky před robotem a barvy pod ním bylo rozděleno na dvě podmínky.

Tím je docíleno, že se ukončovací fáze spustí, jen když je robot natočen směrem, kterým vyjel, neboli je na konci bludiště. Jsou-li obě podmínky splněny, provede se posun robota o krok kupředu a volání procedury `vpredSub()` ze sebe samotné. Při porušení první podmínky, která signalizuje překážku před robotem, se spustí procedura `rozhodniSmerSub()`.

Tato procedura, jak její název napovídá, má na starost rozhodnout, kterým směrem je lepší překážku objet. Změří se nejprve, je-li vzdálenost od nějaké překážky na levé straně větší jak 19 cm. Jestliže ano, zapíše se tato vzdálenost do proměnné `max_vz_l` a to samé se provede na druhé straně do proměnné `max_vz_r`. Následně se obě naměřené hodnoty porovnají a podle toho, která je větší, se rozhodne směr objetí.

V případě, že na začátku procedury `rozhodniSmerSub()` podmínka splněna nebyla, přejde se rovnou na proceduru `objetiVpravoSub()` a překážka se objede vpravo. Samotné procedury pro objetí překážek, jak vpravo, tak vlevo, jsou řešeny tak, že se robot nejprve natočí na stranu, kudy má překážku objet a podmínkou se hlídá vzdálenost od překážky na straně robota, která je v tu chvíli právě natočena směrem na konec bludiště. Robot pak popojíždí tak dlouho, dokud není vzdálenost větší než 15 cm, což signalizuje, že je senzor již za překážkou.

Následně se provede cyklus pěti posunů, aby se tak vyrovnala šířka robota. Robot se pak natočí směrem ke konci bludiště a znovu se zavolá procedura `vpredSub()`. V průběhu procedur `objetiVlevoSub()` a `objetiVpravoSub()` se ještě kontroluje barva pod robotem, a to jak vždy v její hlavní části, tak i následně ve spuštěném cyklu na kompenzaci šířky robota. Děje se tak proto, aby se nestalo, že robot vyjede z pole. Zaznamená-li se vyjetí, tak se robot natočí ke konci a spustí se procedura objetí překážky na druhou stranu. Záznam cesty probíhá stejně jako v příkladu výše (*viz str. 36*) přímo v procedurách, které odpovídají za pohyb robota `EV3_1`;

shrnutí: jako již předešlé programy mé práce je i tento ořezán jen na ty nejdůležitější části, které jsou potřeba pro splnění zadaného úkolu, nebo o části, které by se zde již opakovaly, jako je kód pro druhého robota, procedury pro pohyb, procedura `konec()` a výpisy na LCD. Celé kódy jsou na přiloženém médiu.

Příklad společného zvládnutí projetí bludištěm:

```
start:
If Sensor.ReadRawValue(sensor_barva, 0) =6 Then
    Program.Delay(1000)
    vpredSub()
Else
    Goto start
EndIf

Sub vpredSub
    If Sensor.ReadRawValue(sensor_predek, 0)/10 > 4 Then
        If Sensor.ReadRawValue(sensor_barva, 0) = 6 Then
            pohon_vpřed()
            vpredSub()
        Else
            konec()
        EndIf
    Else
        rozhodniSmerSub()
    EndIf
EndSub

Sub rozhodniSmerSub
    If Sensor.ReadRawValue(sensor_leva, 0)/10 > 19 Then
        max_vz_l = Sensor.ReadRawValue(sensor_leva, 0)
    Else
        objetiVpravoSub()
    EndIf

    If Sensor.ReadRawValue(sensor_prava, 0)/10 > 19 Then
        max_vz_r = Sensor.ReadRawValue(sensor_prava, 0)
    EndIf

    If max_vz_l > max_vz_r Then
        objetiVlevoSub()
    Else
        objetiVpravoSub()
    EndIf

    If max_vz_l = max_vz_r Then
        objetiVlevoSub()
    EndIf
EndSub

Sub objetiVlevoSub
    pohon_doleva()
    startObVlevo:
    pohon_vpřed()
```

```

If Sensor.ReadRawValue(sensor_prava, 0)/10 > 15 Then
  For i = 1 To 5
    If Sensor.ReadRawValue(sensor_barva, 0) = 6 Then
      pohon_vpred()
    Else
      pohon_doprava()
      objektiVpravoSub()
    EndIf
  EndFor
  pohon_doprava()
  vpredSub()
EndIf

If Sensor.ReadRawValue(sensor_barva, 0) = 6 Then
  Goto startObVlevo
Else
  pohon_doprava()
  objektiVpravoSub()
EndIf
EndSub

Sub objektiVpravoSub
  pohon_doprava()
  startObVpravo:
  pohon_vpred()

If Sensor.ReadRawValue(sensor_leva, 0)/10 > 15 Then
  For i = 1 To 5
    If Sensor.ReadRawValue(sensor_barva, 0) = 6 Then
      pohon_vpred()
    Else
      pohon_doleva()
      objektiVlevoSub()
    EndIf
  EndFor
  pohon_doleva()
  vpredSub()
EndIf

If Sensor.ReadRawValue(sensor_barva, 0) = 6 Then
  Goto startObVpravo
Else
  pohon_doleva()
  objektiVlevoSub()
EndIf
EndSub

```

8 ZÁVĚR

Cílem mé bakalářské práce Kooperace robotických systémů bylo sestrojít dva roboty ze stavebnice Lego Mindstorm EV3 a demonstrovat na nich možnosti jejich komunikace a spolupráce.

Nejprve jsem uvedl stručnou historii robotů a jejich postupný vývoj. V další části práce seznamuji čtenáře s historií, vývojem a principem fungování technologie Bluetooth, díky které je možná komunikace a spolupráce robotů. Následuje seznámení se stavebnicí Lego Mindstorms EV3, jež byla použita pro demonstraci úloh v mé bakalářské práci. Jsou popsány jednotlivé části stavebnice a především ty, z kterých byli roboti sestaveni. Také krátce vysvětluji, jakým způsobem spolu jednotlivé části stavebnice komunikují. Ve čtvrté části pojednávám o základech algoritmizace a o jejich hlavních prvcích, které jsou důležité při tvorbě úloh.

V páté blíže popisují hlavní robotickou úlohu a představují možnosti programování této robotické stavebnice. Mnou vybraná možnost programování, což je IDE a programovací jazyk Small Basic, je v kapitole i následně popsána. V předposlední části řeším návrh, konstrukce a výsledný vzhled obou robotů. Poslední část věnuji již samotné hlavní robotické úloze a jejím menším podúlohám. Demonstruji komunikaci robotů přes Bluetooth, rozparsování textového řetězce, reagování robotů na okolní prostředí a také demonstruji závěrečnou úlohu projetí bludištěm, ve které byly části všech podúloh použity.

Využití těchto principů se nachází nejen v různých průmyslových odvětvích ale i v jiných životních situacích lidstva, roboti mohou zastoupit člověka v nebezpečných pracích, např. natahování kabeláže skrz výrobní halu špatně přístupnými nebo nějak pro lidi nebezpečnými prostory. Byl by určen jen cílový bod, kam by bylo potřeba kabel natáhnout. Vyslaní roboti průzkumníci by prozkoumali cestu k cíli a ten, který by našel nejkratší průchodnou trasu, by ji zaslal robotu, ten by tam kabel dotáhl a třeba i někam zapojil.

Tento princip by šel použit i k nalezení cesty k raněným v zavaleném dole nebo ve zřícené budově. Nalezenou cestou by byl následně vyslán robot schopný zraněným poskytnout první pomoc a dopravit potřebné zásoby pro přežití, než se k nim dostanou záchranáři. Výsledky by se jistě velice zlepšily přidáním algoritmů, které by poznaly, že si průzkumný robot např. zbytečně zajel, a tyto slepé cesty by z výsledné trasy byly odstraněny.

9 SEZNAM LITERATURY A INTERNETOVÝCH ZDROJŮ

- [1] Factory Automation: Historie robotů? Sahá až do řecké mytologie! [online]. [cit. 2017-08-1]. Dostupné z: <https://factoryautomation.cz/historie-robotu-saha-az-do-recke-mytologie/>
- [2] In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-08-1]. Dostupné z: <https://cs.wikipedia.org/wiki/Bluetooth>
- [3] In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-08-1]. Dostupné z: <https://upload.wikimedia.org/wikipedia/commons/thumb/d/da/Bluetooth.svg/2000px-Bluetooth.svg.png>
- [4] Bluetooth: Bluetooth 5 [online]. [cit. 2017-08-1]. Dostupné z: <https://www.bluetooth.com/specifications/bluetooth-core-specification/bluetooth5>
- [5] TKÁČ, Josef. Jak na Bluetooth v rekordním čase. 1. vyd. Praha : Grada Publishing, a.s., 2005. 81 s. ISBN 80-247-1081-1.
- [6] TÁKAVCOVÁ, M. 2009. Tvorba bezdrátové domácí sítě Wi-fi a možnost sdílení pomocí technologie bluetooth. Bakalářská práce. Praha: Vysoká škola ekonomická v Praze. Fakulta informatiky a statistiky. 54 s. Dostupné z: <http://info.sks.cz/www/zavprace/soubory/68700.pdf>
- [7] In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-08-1]. Dostupné z: https://en.wikipedia.org/wiki/Lego_Mindstorms_EV3
- [8] 45500 [online]. In: . [cit. 2017-08-1]. Dostupné z: [https://sh-s7-live-s.legocdn.com/is/image/LEGO/45500?\\$PDPDefault\\$](https://sh-s7-live-s.legocdn.com/is/image/LEGO/45500?$PDPDefault$)
- [9] 45502 [online]. In: . [cit. 2017-08-1]. Dostupné z: [https://sh-s7-live-s.legocdn.com/is/image/LEGO/45502?\\$main\\$](https://sh-s7-live-s.legocdn.com/is/image/LEGO/45502?$main$)
- [10] 45503 [online]. In: . [cit. 2017-08-1]. Dostupné z: [https://sh-s7-live-s.legocdn.com/is/image/LEGO/45503?\\$main\\$](https://sh-s7-live-s.legocdn.com/is/image/LEGO/45503?$main$)
- [11] 45507 [online]. In: . [cit. 2017-08-1]. Dostupné z: [https://sh-s7-live-s.legocdn.com/is/image/LEGO/45507?\\$PDPDefault\\$](https://sh-s7-live-s.legocdn.com/is/image/LEGO/45507?$PDPDefault$)

- [12] rsz_legoek_f9udjoiifoidgrt-0ih0tf0001 [online]. In: . [cit. 2017-08-1]. Dostupné z: http://www.intorobotics.com/wp-content/uploads/2013/09/rsz_legoek_f9udjoiifoidgrt-0ih0tf0001.jpg
- [13] 45505 [online]. In: . [cit. 2017-08-1]. Dostupné z: [https://sh-s7-live-s.legocdn.com/is/image/LEGO/45505?\\$PDPDefault\\$](https://sh-s7-live-s.legocdn.com/is/image/LEGO/45505?$PDPDefault$)
- [14] PARK, Eun Jung. Exploring LEGO Mindstorms EV3: tools and techniques for building and programming robots. Indianapolis: John Wiley & Sons, Inc., 2014, xxvi, 376 pages. ISBN 11-188-7974-0.
- [15] 45504 [online]. In: . [cit. 2017-08-1]. Dostupné z: [https://sh-s7-live-s.legocdn.com/is/image/LEGO/45504?\\$PDPDefault\\$](https://sh-s7-live-s.legocdn.com/is/image/LEGO/45504?$PDPDefault$)
- [16] Hitechnic senzors. Hitechnic [online]. 2009-2012 [cit. 2017-08-1]. Dostupné z: <http://www.hitechnic.com/products>
- [17] In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-08-1]. Dostupné z: <https://cs.wikipedia.org/wiki/I%C2%B2C>
- [18] i2connections [online]. In: . [cit. 2017-08-1]. Dostupné z: <http://www.mindsensors.com/img/cms/i2connections.png>
- [19] In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-08-1]. Dostupné z: <https://cs.wikipedia.org/wiki/I%C2%B2C#/media/File:I2C.svg>
- [20] ALGORITMUS1-E-learning [online]. In: . [cit. 2017-08-1]. Dostupné z: https://moodle.sspbrno.cz/pluginfile.php/1951/mod_resource/content/0/Algoritmy/ALGORITMUS1-E-learning.pdf
- [21] PŠENČÍKOVÁ, Jana. Algoritmizace. Kralice na Hané: Computer Media, 2009. ISBN 978-80-7402-034-6
- [22] Vývojové diagramy značky [online]. In: . [cit. 2017-08-1]. Dostupné z: <http://robowiki.spsnome.cz/uploads/Programovani/vd-znacky.jpg>
- [23] Software Lego Mindstorms EV3 [online]. In: . [cit. 2017-08-1]. Dostupné z: <https://www.generationrobots.com/img/Software-Lego-Mindstorms-EV3.png>

- [24] IT network: Small Basic[online]. In: . [cit. 2017-08-1]. Dostupné z: <https://www.itnetwork.cz/programovani/small-basic>
- [25] EV3 Basic: EV3 Small Basic [online]. In: . [cit. 2017-08-1]. Dostupné z: <https://sites.google.com/site/ev3basic/>
- [26] Bluetooth_1 [online]. In: . [cit. 2017-08-1]. Dostupné z: https://www.mathworks.com/help/supportpkg/legomindstormsev3io/ug/Bluetooth_1.png
- [27] Bluetooth_3 [online]. In: . [cit. 2017-08-1]. Dostupné z: https://www.mathworks.com/help/supportpkg/legomindstormsev3io/ug/Bluetooth_3.png
- [28] Bluetooth_6 [online]. In: . [cit. 2017-08-1]. Dostupné z: https://www.mathworks.com/help/supportpkg/legomindstormsev3io/ug/Bluetooth_6.png