

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA PEDAGOGICKÁ
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

**VZDĚLÁVACÍ ÚLOHY PRO VÝUKU PROGRAMOVÁNÍ NA SŠ
POMOCÍ SYSTÉMU LEGO MINDSTORMS NXT A ROBOTC**
DIPLOMOVÁ PRÁCE

Bc. Ladislav Štrunc

Učitelství pro střední školy, obor INF-Ge

Vedoucí práce: Mgr. Tomáš Jakeš

Plzeň, 2014

Prohlašuji, že jsem diplomovou práci vypracoval samostatně s použitím uvedené literatury a zdrojů informací.

V Plzni, 30. června 2014

.....
vlastnoruční podpis

Poděkování

Poděkování patří především vedoucímu mé diplomové práce Mgr. Tomáši Jakešovi za vstřícný přístup, trpělivost, odborné rady a cenné připomínky.

OBSAH

Úvod	3
1 CHARAKTERISTIKA ROBOTICKÉ STAVEBNICE LEGO MINDSTORMS NXT	4
1.1 ŘÍDÍCÍ JEDNOTKA NXT	6
1.2 VÝSTUPNÍ MODULY	9
1.2.1 Interaktivní servomotory	9
1.2.2 Světelné kostky	10
1.3 VSTUPNÍ MODULY	10
1.3.1 Dotykový senzor	10
1.3.2 Zvukový senzor	11
1.3.3 Světelný senzor	11
1.3.4 Ultrazvukový senzor	12
1.3.5 Doplnňkové senzory	13
2 VYUŽITÍ STAVEBNICE NA STŘEDNÍCH ŠKOLÁCH	14
2.1 PROGRAMOVÁNÍ	14
2.2 ROBOTIKA	15
2.2.1 Soutěže	16
2.3 AUTOMATIZACE	17
2.4 VYUŽITÍ V DALŠÍCH PŘEDMĚTECH	17
3 POPIS ROBOTC.....	19
3.1 DOSTUPNOST	19
3.2 UŽIVATELSKÉ PROSTŘEDÍ	19
3.3 STRUČNÉ PŘEDSTAVENÍ JAZYKA ROBOTC.....	25
3.4 UŽIVATELSKÁ PODPORA	27
4 VÝUKA PROGRAMOVÁNÍ NA SŠ POMOCÍ STAVEBNICE LEGO MINDSTORMS	28
4.1 VÝUKA PROGRAMOVÁNÍ V RVP PRO SŠ.....	28
4.1.1 Gymnázia	28
4.1.2 Střední odborné vzdělávání bez zaměření na informační technologie	29
4.1.3 Střední odborné vzdělávání zaměřené na informační technologie	29
4.2 SOUHRN INFORMACÍ O VÝUCE PROGRAMOVÁNÍ NA SŠ	30
5 SADA ÚLOH.....	34
5.1 CÍLE PRO SADU ÚLOH.....	34
5.2 TVORBA ÚLOH	35
5.3 POUŽITÍ SADY ÚLOH	35
5.4 OSNOVA VYTVOŘENÉ SADY ÚLOH.....	37
5.5 ZADÁNÍ A METODICKÉ POKYNY K VYTVOŘENÝM ÚLOHÁM	40
5.5.1 Úloha 1: Zachraňte Jamese Bota	41
5.5.2 Úloha 2: Automatický autobus	42
5.5.3 Úloha 3: Zvedací most	44
5.5.4 Úloha 4: Adaptivní tempomat	46
5.5.5 Úloha 5: Budík	47
5.5.6 Úloha 6: Vrták.....	49
5.5.7 Úloha 7: Automatické světlo	51
5.5.8 Úloha 8: Robotický smetáček	53
5.5.9 Úloha 9: Sprinter „RoBolt“	55
5.5.10 Úloha 10: Jízda po čáře.....	57

5.5.11 Úloha 11: Trasa jízdy.....	60
5.5.12 Úloha 12:Třídíčka míčků	62
ZÁVĚR.....	64
RESUMÉ.....	66
SEZNAM LITERATURY	68
SEZNAM OBRÁZKŮ, TABULEK, GRAFŮ A DIAGRAMŮ	73
PŘÍLOHY	I

Úvod

Tato diplomová práce se zabývá tvorbou vzdělávacích úloh pro výuku programování na středních školách.

Cílem diplomové práce je přiblížit robotickou stavebnici LEGO Mindstorms a vymezit možnosti jejího využití v rámci výuky na středních školách. Dále se diplomová práce věnuje programovacímu prostředí RobotC. Primárním cílem je vytvoření vlastní sady úloh určených pro podporu výuky programování.

Dnešní studenti jsou vyspělou techniku zvyklí používat ve svém běžném životě a výuka programování na běžných počítačích pro ně tudíž není tolik atraktivní. Téma vzdělávání za pomoci stavebnice Lego Mindstorms jsem si zvolil, protože v této pomůcce vidím vhodný technický doplněk výuky. Studentům přináší potřebné zpestření a zároveň adekvátně aplikuje získané znalosti při praktických ukázkách.

V první kapitole je představena robotická stavebnice LEGO Mindstorms a moduly, které jsou obsaženy v základní soupravě určené pro výuku ve školách a které jsou využity při praktických úlohách.

Druhá kapitola se zabývá možným využitím stavebnice LEGO Mindstorms na středních školách. Detailněji se věnuje především využití stavebnice při výuce programování a robotiky. V rámci tématu jsou také zmíněny robotické soutěže, kterých se studenti mohou zúčastnit.

Vývojové prostředí RobotC je přiblíženo ve třetí kapitole. Důraz je kladen především na stručné představení a přiblížení uživatelského prostředí včetně dostupné podpory.

Čtvrtá kapitola se věnuje možnému využití stavebnice LEGO Mindstorms ve výuce programování na středních školách. Podkladem pro seznámení s rozsahem výuky programování jsou Rámcové vzdělávací programy určené pro zvolené typy vzdělání.

Tvorbě sady úloh se věnuje poslední kapitola, v rámci které jsou nejprve stanoveny cíle, které se pomocí sady mají plnit. Dále je popsán způsob tvorby jednotlivých úloh a jejich použití. Podstatnou součástí pak tvoří sada úloh. Kapitola informuje o způsobu tvorby a použití sady úloh. Součástí kapitoly je také výstup diplomové práce a to vypracované úlohy.

1 CHARAKTERISTIKA ROBOTICKÉ STAVEBNICE LEGO MINDSTORMS NXT

Společnost LEGO se zabývá vývojem robotických stavebnic již od konce 80. let 20. století. První robotická stavebnice LEGO Mindstorms Robotics Invention System (RIS) byla představena a uvedena do prodeje v roce 1998 [1]. Pro označení robotické stavebnice bylo vybráno jméno Mindstorms. Jako inspirace posloužila kniha Seymoura Paperta Mindstorms: Children, Computers, and Powerful Ideas [2] [3]. Kniha vyšla v roce 1980. Její hlavní myšlenkou bylo, že se děti mohou učit ovládat počítače a tím mohou změnit celý způsob, jakým se učí i v jiných předmětech [4]. První generace stavebnice je známá také pod názvem Lego Mindstorms RCX, který je odvozen od pojmenování řídicí kostky RCX (Robotic Command Explorers) [3].

V roce 2006 byla na trh uvedena nová generace robotické stavebnice LEGO Mindstorms NXT. Označení NXT je odvozeno z anglického slova next (další), které odkazuje na to, že se jedná o novou generaci stavebnice [5]. O tři roky později byla vydána aktualizovaná verze označená jako LEGO Mindstorms NXT 2.0.

V současnosti nejaktuálnější verze byla představena v roce 2013 a jmenuje se LEGO Mindstorms EV3. Zkratka EV3 (Evolution 3) připomíná, že se jedná již o třetí generaci robotických stavebnic [6]. Pro účely diplomové práce byla k tvorbě výukových úloh využita starší verze stavebnice LEGO Mindstorms NXT 2.0. Důvodem je fakt, že verze EV3 byla v České republice uvedena do prodeje v září 2013 [1], tedy až po zadání této práce. Navíc podpora pro zvolné vývojářské prostředí RobotC byla k dispozici až od jara 2014 [7],



Obrázek 1: Vývoj robotických stavebnic LEGO Mindstorms [1]

Robotické stavebnice Mindstorms patří k nejúspěšnějším a nejprodávanějším produktům společnosti LEGO. Stavebnice se těší velké popularitě, jsou organizovány výstavy a soutěže, publikuje se velké množství knih a návodů. Stavebnice dokázaly zaujmout i zákazníky,

nadšence do robotiky, kteří by o výrobky LEGO jinak neprojevovali velký zájem a naopak, poutavým a jednoduchým způsobem přiblížily robotiku širšímu spektru zákazníků [1] [8].

Praktická část této diplomové práce byla vytvořena s pomocí robotické stavebnice LEGO Mindstorms NXT, konkrétně druhou verzí této generace, která byla na trh uvedena v roce 2009 [1]. Specifikem použité stavebnice je to, že se jedná o speciální upravenou verzi určenou pro využití ve školách, s úplným označením LEGO Mindstorms Education NXT Base Set 9797 (Obrázek 2). Oproti standardní verzi určené k volnému prodeji se odlišuje v několika bodech:

- obsahuje rozdílné senzory (na rozdíl od komerční verze má navíc zvukový a světelný senzor, barevný senzor naopak chybí);
- neobsahuje programovací software;
- je dodávána v odlišném balení, vhodném pro snadnější uskladnění [9] [10] [11] [12].

Název také napovídá, že se jedná o základní soupravu (tzv. base set). K základní soupravě lze pořídit i soupravu doplňkových dílů, která dále rozšiřuje konstrukční možnosti stavebnice [13].



Obrázek 2: LEGO Mindstorms Education NXT Base Set 9797 [11]

Možnosti pro další rozšiřování robotické stavebnice LEGO Mindstorms NXT jsou značné, ale neznamená to, že by byl základní set vybaven nedostatečně. Balení obsahuje celkem 437 dílů, přičemž většinu tvoří konstrukční díly z řady LEGO Technic (Obrázek 3).



Obrázek 3: Ukázka dílů LEGO Technic

Kromě konstrukčních dílů jsou součástí stavebnice i aktivní robotické prvky. Jsou to:

- řídicí jednotka NXT
- výstupní moduly
 - tři interaktivní servomotory
 - tři světelné kostky
- vstupní moduly
 - dva dotykové senzory
 - zvukový senzor
 - světelný senzor
 - ultrazvukový senzor

V balení nechybí propojovací kabely různých délek určených k propojení senzorů s řídicí jednotkou NXT. Kabely tvoří nepostradatelnou část stavebnice, bez nichž by nebyl provoz robota možný. V přehledu byl představen obsah balení robotické stavebnice LEGO Mindstorms NXT. Budeme se věnovat jejím aktivním součástkám.

1.1 ŘÍDÍCÍ JEDNOTKA NXT

Řídicí jednotka NXT je nejdůležitějším aktivním prvkem celé stavebnice (Obrázek 4). Jedná se o programovatelný mikropočítač. Zajišťuje chod našich programů, řídí pohyby robota či spravuje ostatní senzory.



Obrázek 4: Řídící jednotka NXT [14]

Uvnitř kostky nalezneme 32bitový mikroprocesor ARM7 s 64Kb operační pamětí od společnosti Atmel. V řídicí jednotce NXT se vyskytuje ještě jeden pomocný 8bitový koprocesor AVR, který vypomáhá se správou senzorů. Pro programy je připravena 256Kb FLASH paměť [15]. Paměť je využívána i pro další soubory a zejména soubory zvuků ji mohou rychle zaplnit.

Kromě výpočetního výkonu se řídicí jednotka NXT také vyznačuje následujícími charakteristikami:

Napájení

Řídící jednotku NXT je možné napájet dvěma způsoby. Buďto pomocí šesti tužkových AA baterií, nebo pomocí nabíjecí lithiové baterie. Zde je nutno upozornit, že síťový adaptér, který slouží k dobíjení, není v balení a musí se zakoupit zvlášť [10]. Z baterií v kostce se napájí vstupní a výstupní moduly [16].

Výstupy

Na kostce jsou umístěny tři výstupní porty. Nacházejí se na horní straně kostky a jsou označeny písmeny A, B, C (Obrázek 5). Výstupy slouží pro zapojení motorů a světelných kostek. Zapojení motorů či světelných kostek je možné libovolně kombinovat. Avšak existuje výchozí zapojení, které ukazuje Tabulka 1.

Vstupy

Vstupní porty slouží pro zapojení senzorů. Na kostce se nalézají čtyři a jsou umístěny na dolní hraně kostky, mají číselné značení 1, 2, 3, 4 (Obrázek 5). Obdobně jako u výstupů zde nezáleží na tom, jaký senzor zapojíme do kterého portu, ale i zde se udává výchozí zapojení (Tabulka 1).

Tabulka 1: Výchozí zapojení výstupů a vstupů [12]

Výstupy	
Port A	Motor pro zvláštní využití
Port B	Levý pojezdový motor
Port C	Pravý pojezdový motor
Vstupy	
Port 1	Dotykový senzor
Port 2	Zvukový senzor
Port 3	Světelný senzor
Port 4	Ultrazvukový senzor

Možnosti připojení

Kostka nám nabízí připojení k PC pomocí USB 2.0 portu (Obrázek 5). V kostce se však nachází i bezdrátové Bluetooth rozhraní. Pomocí rozhraní Bluetooth může kostka komunikovat nejen s PC, ale také s dalšími kostkami. Pro komunikaci lze přednastavit až tři různé kostky, přenos dat však může probíhat vždy pouze s jednou z nich [15]. Je také možné komunikovat s mobilními zařízeními, díky čemuž můžeme např. ovládat pohyb robota pomocí telefonu vybaveného technologií Bluetooth a vhodnou aplikací.

Reproduktor

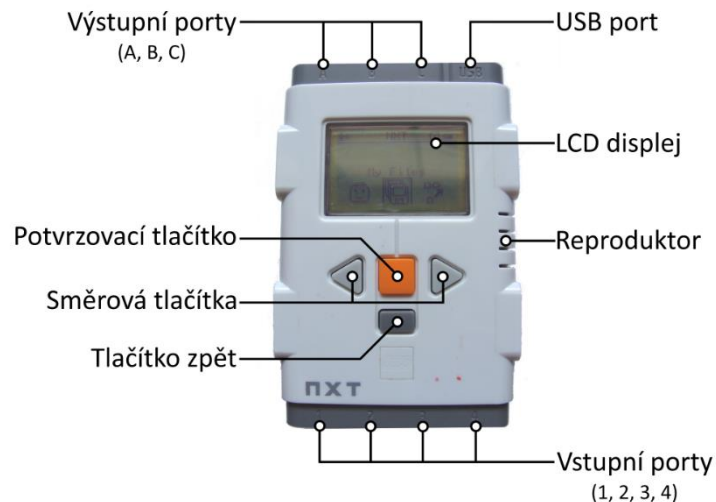
Jedná se spíše o doplňkový prvek NXT kostky. Reproduktor umožňuje uživatelsky příjemnější ovládání kostky, kdy je možné získávat také zvukové informace o průběhu programu. Zejména se jedná o pokyny „start“, „stop“ či „chyba“. Reproduktor se nachází pod mřížkou na pravé straně (Obrázek 5).

Tlačítka

Na kostce nalezneme čtyři ovládací tlačítka (Obrázek 5). Primárně slouží pro ovládání kostky a pohyb v menu. Středové oranžové tlačítko je potvrzovací a zároveň slouží pro zapnutí řídicí kostky NXT. Pod ním se nachází tlačítko, které slouží pro návrat zpět i pro vypnutí řídicí kostky. Vedle středového tlačítka jsou umístěna směrová tlačítka pro navigaci v menu. Tlačítka je možné přeprogramovat tak, aby během chodu programu plnila požadovanou funkci.

Displej

Pro lepší uživatelskou podporu je součástí řídicí jednotky NXT také černobílý maticový displej s rozlišením 100 x 64 pixelů. Slouží především pro zobrazení menu, uplatní se však i v průběhu programu, kdy je možné jej využít například pro vypsání textu či vykreslování jednoduché grafiky.

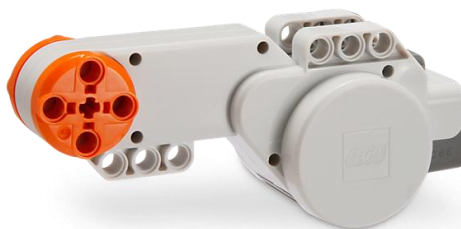


Obrázek 5: Popis řídicí jednotky NXT

1.2 VÝSTUPNÍ MODULY

1.2.1 INTERAKTIVNÍ SERVOMOTORY

Servomotory (Obrázek 6) slouží především k pohybu robota nezávisle na tom, zda má kola, pásy nebo nohy. Mohou mít ale také další uplatnění, například ovládání různých druhů ramen. Motory mají obousměrný chod a je možnost nastavit jejich výkon v intervalu od - 100% do + 100%. Pokud se nastaví nulový výkon, motory se zastaví.



Obrázek 6: Interaktivní servomotor [14]

Důležitou vlastností motorů je to, že jsou vybaveny vestavěnými rotačními senzory. Ty snímají s rozlišením 1° [15]. S jejich pomocí lze na základě úhlu natočení definovat posun robota o požadovanou vzdálenost. Softwarově lze také motory synchronizovat. Díky těmto vlastnostem je možné mnohem lépe ovlivňovat pohyb robota.

1.2.2 SVĚTELNÉ KOSTKY

Jedná se o kostky se zabudovanými žárovkami. Tyto kostky jsou pozůstatkem původní stavebnice LEGO Mindstorms RCX a proto jsou k nim dodávány speciální kabely s redukcí. Na žárovky lze nasadit průhledné barevné kostky, a tím docílit efektu barevného světla.

1.3 VSTUPNÍ MODULY

Senzory poskytují zpětnou vazbu o chování robota a informují ho o okolním prostředí. V základním balení jsou čtyři typy: dotykový, zvukový, světelný a ultrazvukový.

1.3.1 DOTYKOVÝ SENZOR

Jedná se o jednoduchý a užitečný senzor. Dotykový senzor reaguje na stisknutí oranžového tlačítka (Obrázek 7). Může tak nabývat pouze dvou stavů: uvolněno – stisknuto. Pomocí těchto stavů je možné sledovat tři události [15]:

- stisknutí (Pressed) – dlouhodobé stisknutí (držení), trvající většinou déle než půl vteřiny až jednu vteřinu;
- uvolnění (Released) – uvolnění senzoru, tlačítko není stisknuto;
- stisknutí a opětovné uvolnění (Bumped) – uvolnění musí přijít nejdéle jednu vteřinu, většinou však půl vteřiny po stisknutí [17].



Obrázek 7: Dotykový senzor [14]

Nejčastější využití dotykového senzoru je pro detekci nárazu robota. Časté využití je i pro ovládání funkcí robota, např. rozjetí robota po stisknutí tlačítka.

1.3.2 ZVUKOVÝ SENZOR

Zvukový senzor měří úroveň hluku v jeho blízkosti (Obrázek 8). Senzor měří úroveň hluku v dB a dBA¹, a to až do hodnoty 90 dB [15], která odpovídá např. sekačce na trávu či mixéru [17]. Hodnoty jsou však vyjadřovány v procentech od 0 do 100. Jeho použití může být komplikované v rušných prostředích.



Obrázek 8: Zvukový senzor [14]

Zvukový senzor je často využíván pro roboty, kteří se reakcí na hlasitý zvuk rozjíždí a zastavují.

1.3.3 SVĚTELNÝ SENZOR

Světelný senzor (Obrázek 9) umožňuje robotovi vidět, tedy alespoň částečně. Detekuje intenzitu světla, která dopadá na fototranzistor. Senzor může pracovat ve dvou režimech: aktivním a pasivním.



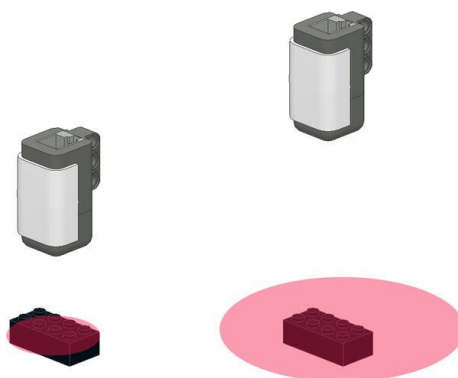
Obrázek 9: Světelný senzor [14]

V aktivním režimu je zapnuta přisvětlující LED dioda. Fototranzistor snímá intenzitu odraženého světla od povrchu, který LED dioda ozařuje. Senzor nám tyto hodnoty udává

¹ dB – jednotka akustického tlaku, dBA – označuje akustický tlak, který je přepočítán podle závislosti, které charakterizují vlastnosti lidského ucha.

v procentech od 0 do 100. V tomto režimu má senzor velmi malé zorné pole a je potřeba, aby byl umístěn blízko u měřeného povrchu (Obrázek 10).

Nejtypičtější činností senzoru v aktivním režimu je jízda po vodící čáře. Případně je možné senzor použít na rozpoznávání základních barev. Barvy však musí být dostatečně kontrastní.



Obrázek 10: Umístění světelného senzoru [18]

V pasivním režimu je přisvětlující LED dioda vypnuta. Senzor měří intenzitu světla v místnosti. Hodnoty se taktéž pohybují od 0 do 100%.

Oblíbené využití je reakce robota na rozsvícení či zhasnutí světla v uzavřené místnosti. Pro optimální reakci je zapotřebí dostatečného kontrastu mezi světelnými podmínkami při rozsvíceném a zhasnutém světelném zdroji.

1.3.4 ULTRAZVUKOVÝ SENZOR

Ultrazvukový senzor (Obrázek 11) tvoří další část zraku robota. Oproti světelnému senzoru neměří intenzitu světla, ale měří vzdálenost objektu od senzoru. Vzdálenost měří podobně jako např. netopýři. Senzor vyšle vysokofrekvenční zvukovou vlnu a měří, za jak dlouho se vrátí zpět. Z výpočtu poté určí vzdálenost od objektu.

Senzor má udávaný rozsah 0 až 255 centimetrů s přesností měření 3 centimetry [15]. Při používání více senzorů v jedné místnosti hrozí zkreslení výsledků, protože se mohou navzájem rušit. Ultrazvukový senzor může mít problémy s rozpoznáváním některých překážek. Je vhodné používat spíše větší a pravidelné předměty, než malé a zakulacené [16].



Obrázek 11: Ultrazvukový senzor [14]

Nejčastější využití senzoru je kontrola, zda robot nejede na překážku. Zajímavou možností může být i opačné nastavení, kdy robot bude vyhledávat objekt. Senzor může být také využit k detekci pohybu.

1.3.5 DOPLŇKOVÉ SENZORY

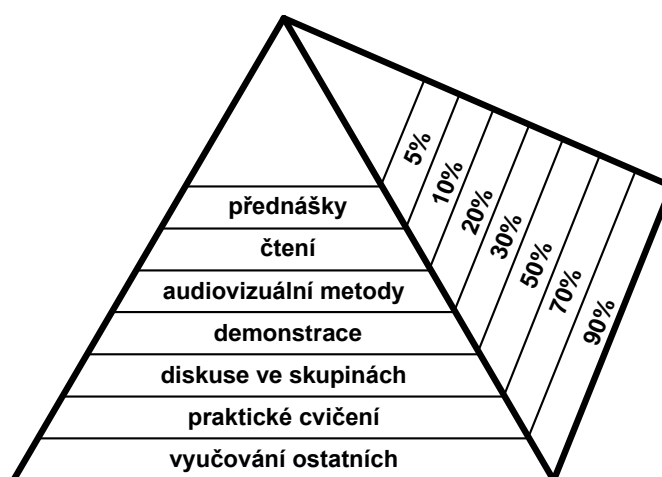
Senzory popsané v předešlé kapitole jsou k dispozici v balení LEGO Mindstorms Education NXT Base Set 9797. Avšak stavebnici lze také rozšířit o další senzory, které však již nedodává přímo společnost LEGO, ale jedná se o výrobky třetích stran. Známé jsou například výrobky firem HiTechnic, Vernier, DCP Microdevelopments (senzory LogIt) nebo Mindsensors. Důležité je, že velká část těchto rozšíření je certifikována přímo společností LEGO [19] [20].

2 VYUŽITÍ STAVEBNICE NA STŘEDNÍCH ŠKOLÁCH

V současné době lze zaznamenat nárůst využívání moderní didaktické techniky (např. interaktivní tabule, dataprojektory, vizualizéry či tablety). Důvodem je především snaha o zpestření výuky a větší aktivizaci studentů. Při vhodně zvolené metodě a formě výuky mohou tyto prostředky studentům usnadnit pochopení probírané látky.

Mezi moderní didaktickou techniku se řadí i robotická stavebnice LEGO Mindstorms. Stavebnice může nabízet jednoduché řešení, jak zpestřit a zatraktivnit běžnou výuku. Díky velkému množství dostupných doplňků ji lze využít také na specificky zaměřené aplikace a tím umožnit odbornou výuku. Během práce s robotickými stavebnicemi LEGO Mindstorms se studenti učí nejen potřebné dovednosti v daném oboru, ale mohou rozvíjet komunikaci, kreativitu a vzájemnou spolupráci [21].

Pomocí stavebnice LEGO Mindstorms lze snáze zapojit studenty do výuky praktickou formou. Díky praktické činnosti studentů je možné dosáhnout větší míry zapamatování látky. Neboť čím více je student zapojen do procesu výuky, tím více informací může získat. Schematické znázornění nabízí tzv. pyramida učení (Obrázek 12), kterou uvádí S. Shapiro. Procenta uvedená u jednotlivých metod mohou být variabilní, neboť výsledný efekt zapamatování není přímo úměrný zvolené metodě [22].



Obrázek 12: Pyramida učení dle S. Shapiro [22]

2.1 PROGRAMOVÁNÍ

Stavebnice LEGO Mindstorms umožňuje výuku programování zaměřenou na rozdílné úrovně odbornosti. Řídící jednotku je možné programovat za pomoci různých jazyků.

Existují jazyky zaměřené spíše na začátečníky, typickým příkladem je jazyk EV3. Dále je možnost řídicí jednotku programovat některým světově rozšířeným jazykem či jazyky od nich odvozenými. Jedná se např. o jazyky RobotC a NXC, které jsou odvozené od jazyka C nebo jazyk NXJ, který je založený na programovacím jazyku Java. I díky tomu se stavebnice stala možnou alternativou ve výuce programování.

Klasická výuka programování je spojena zejména s využitím běžného PC, kdy jsou pokyny zapsány kódem, a následný výstup se zobrazí na monitoru. Toto abstraktní smýšlení může být pro mnohé studenty těžce pochopitelné. Nedokáží si spojit programování s konkrétním výstupem, který využívají v běžném životě a nevidí tedy v programování žádný konkrétní přínos. Programování je díky tomu považováno za nepřiliš zajímavou oblast, ve které se v rámci třídy většinou orientuje pouze několik málo jedinců. Tímto přístupem se snižuje nejen motivace ke studiu, ale také možnost do tématu hlouběji proniknout a pochopit jej [23]. Díky využití stavebnice LEGO Mindstorms nejsou výstupy pouze abstraktní, ale lze je v reálném čase vidět, což je činí mnohem zajímavějšími [24].

Stavebnice LEGO Mindstorms se velmi hodí pro první seznámení studentů s programováním. Studenti při práci s ní získají základní znalosti a seznámí se s principy programování. Výhodou je, že pro úvodní výuku programování lze využít libovolný programovací jazyk [23]. Studentem nabyté dovednosti a návyky jsou z velké části přenositelné a následné učení některého dalšího programovacího jazyka tím bude výrazně ulehčeno. Stavebnici LEGO Mindstorms lze také využívat k pokročilejší výuce některého z programovacích jazyků. Taková výuka však již zpravidla neprobíhá na středních školách, ale spíše na školách vysokých, případně v rámci různých zájmových mimoškolních aktivit.

Konkrétní příklady využití LEGO Mindstorms ve výuce programování lze nalézt i na středních školách v České republice. Například úvod do programování se pomocí LEGO Mindstorms vyučuje na Gymnáziu a Obchodní akademii v Orlové [25]. Stavebnice pomáhá s výukou programování také na Gymnáziu v Jihlavě. Studenti zde mají k dispozici dvě sady videí nazvané „Programování pro neprogramátory“ [26].

2.2 ROBOTIKA

S robotikou se lze setkat na českých středních školách většinou jako s volitelným předmětem. Lze očekávat, že na volitelné předměty se hlásí studenti s větší vnitřní motivací

pro danou oblast. V rámci robotiky se může jednat o programování i techniku jako takovou. Předmět se tedy spíše zaměřuje na vytváření a řešení složitějších úloh. Školy, kde se volitelný předmět robotika těší větší oblibě, se také častěji zúčastňují robotických soutěží. Předmět tak může sloužit i jako příprava na ně.

Na českých středních školách jsou kroužky robotiky poměrně populární. Příkladem může být Gymnázium Sušice, kde provozují kroužky rovnou dva [27]. Studenti nižších ročníků mohou navštěvovat kroužek zaměřený na práci se stavebnicí LEGO Mindstorms. Věnují se především konstrukci a programování jednoduchých robotických modelů. Pro studenty vyšších ročníků je k dispozici kroužek zaměřený na základy programování. Jeho obsahem je jazyk C a programování složitějších robotických modelů. V obou kroužcích se také studenti připravují na různé robotické soutěže.

2.2.1 SOUTĚŽE

Robotické soutěže samozřejmě nejsou součástí standardní výuky na středních školách. Přesto považuji za důležité je v této práci zmínit především proto, že účastníci volitelných předmětů zaměřených na robotiku se na ně v těchto předmětech připravují a účastní se jich. Je tedy vhodné představit alespoň dvě nejvýznamnější robotické soutěže, které se v České republice pořádají.

Jediná mezinárodní soutěž, která má v České republice zastoupení, je FIRST LEGO League (FLL). Tato soutěž je celosvětově organizována již od roku 1998 [28], v České republice pak proběhl první ročník v roce 2006. Soutěž pořádá Česká liga robotiky za podpory Ministerstva školství, mládeže a tělovýchovy [29]. Soutěž je určena pro až desetičlenné týmy dětí ve věku od 10 do 16 let. Soutěží se ve čtyřech disciplínách: Robot Game, prezentace výzkumného úkolu, design robota a týmová práce. Celkovým vítězem se stává tým, který nasbírá největší počet bodů ze všech disciplín. Každý ročník je tematicky zaměřen, např. ročník 2013 se věnoval tématu přírodních katastrof [28].

Ve stejném ročníku byl úspěšný tým studentů sušického gymnázia, který obsadil na evropském semifinále celkové 6. místo a v disciplíně Robot Game dokonce skončil na druhém místě [27].

Pro studenty středních škol je soutěž FIRST LEGO League díky věkovému limitu přístupná pouze v prvním ročníku. Na všechny studenty středních školy, bez věkového omezení, se zaměřuje česká „ROBOSOUTĚŽ“, kterou na Fakultě elektrotechnické Českého vysokého učení technického v Praze pořádá Katedra řídicí techniky [30]. Soutěž je určena pro tříčlenné týmy. Na konci minulého roku (2013) proběhl již 5. ročník soutěže, do kterého se přihlásilo rekordních 64 středoškolských týmů. Šest z nich poustoupilo přímo do finálového univerzitního kola, kde soutěžily s týmy složenými ze studentů této katedry. Vítězem se stal tým pražského gymnázia Jana Keplera [31].

2.3 AUTOMATIZACE

Velmi blízko k robotice má obor automatizace a další obory v rámci strojírenství (mechatronika, CNC² technika). V rámci strojírenství se využití může rozšířit také na konstrukční stránku robota (nosnost, převody) a případně i na technický design (návrh robotů). Obor automatizace se zabývá řízením výrobních a jiných technologických procesů bez nutnosti zásahu lidské obsluhy. Samotný proces automatizace je prováděn pomocí regulátory řízenými akčními členy, které ovlivňují regulovanou soustavu na základě získaných dat ze snímačů a udržují ji tak v požadovaných mezích. Požadované meze či parametry regulované soustavy definuje programátor pomocí napsaného programu. LEGO Mindstorms tak může ulehčit přechod na skutečné robotické automatizační systémy. Na Střední průmyslové škole, střední odborné škole a středním odborném učilišti Nové Město nad Metují vznikl projekt Moderní výuka s podporou ESF³. Tento projekt zastřešuje využívání robotických stavebnic nejen pro výuku automatizace, ale také pro programování a kurzy robotiky [32].

2.4 VYUŽITÍ V DALŠÍCH PŘEDMĚTECH

Využití stavebnice LEGO Mindstorms není omezeno jen na předměty zabývající se programováním či robotikou. Stavebnici je možné využít také v dalších oborech, kde pro ni daný vyučující nalezne smysluplné uplatnění.

² CNC (Computer Numerical Control) – počítačem (číslicově) řízený stroj.

³ ESF – Evropský sociální fond.

Velké možnosti rozšíření stavebnice představují zajímavý způsob, jak pomocí stavebnice vyučovat elektrotechniku. Pro stavebnici existují elektrotechnické doplňky, se kterými je možné vytvářet elektronické obvody za pomoci nepájených spojů. Ke stavebnici lze připojit měřicí přístroje, jako jsou voltmetry a ampérmetry. K elektrotechnice má blízko obor zabezpečovací technologie, ve kterém lze stavebnici také využít. Je možné vytvářet různé varianty ochrany místnosti či objektů [33].

Stavebnice LEGO Mindstorms je možné částečně využívat i v přírodovědných předmětech. Velká část odborných aplikací vyžaduje doplňkové senzory, měřicí zařízení a další techniku. Ve fyzice se mohou za pomoci stavebnice LEGO Mindstorms provádět pokusy, zkoumat a ověřovat např. odraz světla, šíření zvuku či měření průtoku různých kapalin [21] [34]. Stavebnice LEGO Mindstorms může najít uplatnění i během pokusů a měření v chemii. Robot může být naprogramován tak, aby zajišťoval průběh měření teploty kapalin během pokusu a zároveň zpracovával a zobrazoval výsledky. Stavebnici LEGO Mindstorms lze ojediněle využít i při výuce biologie. Např. existují úlohy, které sledují chování jednoduchých organismů, kdy se zadá (naprogramuje) základní chování a poté se sleduje, co robot (organismus) dělá a jak reaguje na podněty (dotyk s překážkou) [35].

3 POPIS ROBOTC

RobotC je vývojové prostředí, které je přizpůsobeno pro ovládání robotických stavebnic. Ve vývojovém prostředí RobotC lze programovat nejen roboty řady LEGO Mindstorms, ale je také určeno pro další stavebnice, jako jsou TETRIX, MATRIX, CORTEX a Arduino [7]. RobotC byl vyvinut na půdě Carnegie Mellon University, konkrétně její Katedry robotiky. Na trh byl uveden ve spolupráci se společností Robomatter Inc, která se specializuje na prodej robotů vyvíjených ve spolupráci právě s touto univerzitou [36].

3.1 DOSTUPNOST

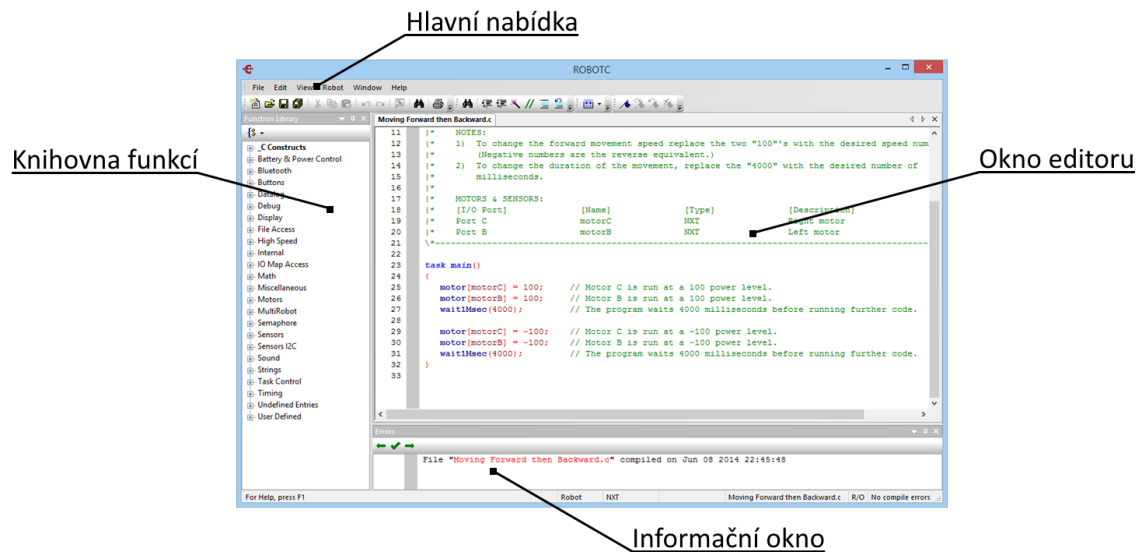
Vývojové prostředí RobotC není součástí základního balení stavebnice. Jedná se o placený software. Z oficiálních stránek www.robotc.net je možné zdarma stáhnout 30 denní zkušební verzi. Tuto verzi je pak možné aktivovat zakoupením licence. Software lze také zakoupit u českého distributora učebních pomůcek LEGO Education společnosti Eduxe [19].

Program je určený pro operační systémy Windows XP SP 2 a vyšší [7]. Jediným dostupným ovládacím jazykem je angličtina. Pro masové rozšíření v České republice by tento fakt mohl být překážkou. Pro využití programátory, ať už současnými či těmi budoucími, to však omezení není. Ve světě informačních technologií je totiž angličtina značně rozšířený jazyk. Ovládání programu je navíc srozumitelné a intuitivní.

Vývojové prostředí RobotC je možné využít jak pro začínající, tak i pokročilé programátory. Vytvářet programy dokáží po seznámení i studenti bez předchozích programátorských zkušeností [23]. Z tohoto důvodu je doporučováno jeho využití na středních a vysokých školách [37].

3.2 UŽIVATELSKÉ PROSTŘEDÍ

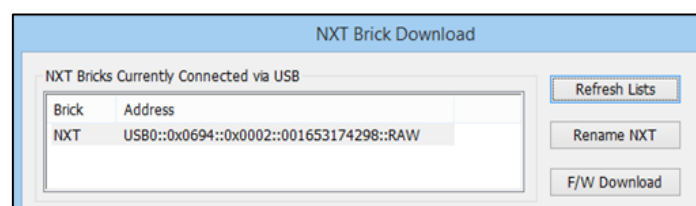
Okno programu RobotC je řešeno účelně a přehledně. Ve výchozím nastavení je možné okno programu rozdělit na čtyři části (Obrázek 13). V horní části se nachází hlavní nabídka a pod ní je panel nástrojů. Levá část okna je vyhrazena pro knihovnu funkcí. Největší plochu zabírá editor pro psaní kódu. V dolní části obrazovky se nalézá prostor pro informační okna. Pozice a typy zobrazených oken si každý uživatel může plně přizpůsobit.



Obrázek 13: Uživatelské prostředí RobotC

První spuštění

Před prvním použitím programu RobotC je potřeba přinstalovat originální firmware⁴ v řídicí jednotce NXT. Návod se uživateli zobrazí ihned po prvním zapnutí programu, takže není nutné vyhledávat další informace a stačí se držet postupu, který uživatele odkáže na okno s možností nahrání nového firmwaru (Obrázek 14). Stejně okno lze otevřít z hlavní nabídky (Obrázek 15) pomocí položky „Robot“ a příkazu „Download Firmware“. V tomto okně se zobrazí nalezená řídicí jednotka a stisknutím na tlačítko „F/W Download“ se firmware přehraje. Ač řídicí jednotka umožňuje bezdrátovou komunikaci s PC pomocí Bluetooth, pro nahrání firmware je nutné jednotku zapojit pomocí USB kabelu. Během nahrávání firmwaru se nesmí řídicí jednotka odpojit.

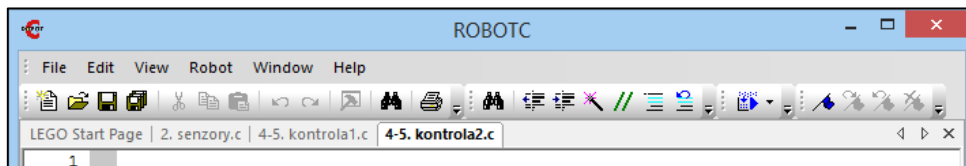


Obrázek 14: Okno pro přehrání firmwaru

⁴ Firmware je software, který řídí činnost a funkce řídicí jednotky. A je uložen na paměťovém čipu.

Práce se soubory

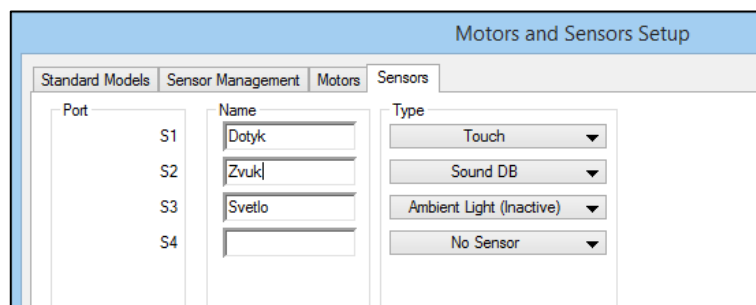
Jak je obvyklé, možnosti na otevření nového projektu, či načtení dříve uloženého, se nacházejí v hlavní nabídce v položce soubor (File). Zároveň mají umístěnou rychlou volbu na panelu nástrojů (Obrázek 15). Uživatel má také možnost rychlého načtení posledních otevřených dokumentů. Projekty se otevírají do záložek (Obrázek 15). To znamená, že lze zobrazit anebo editovat více projektů najednou a pohodlně mezi nimi přepínat.



Obrázek 15: Hlavní nabídka a panel nástrojů

Nastavení senzorů

Většina postavených robotů obsahuje nějaké senzory. Připojené senzory je nutné deklarovat, jinak k nim nelze programově přistupovat. Deklaraci je možné zapsat ručně, ale mnohem příjemnější je tak učinit pomocí okna „Motors a Sensors Setup“ (Obrázek 16), které lze najít v položce „Robot“. V okně se nachází více záložek, pro nastavení senzorů je nutné zvolit záložku „Sensors“. Senzory se zde přiřazují k jednotlivým portům S1 až S4. Toto označení je shodné s připojením senzorů na fyzické kostce. Z rozbalovací nabídky se poté vybere konkrétní typ senzoru. Pro lepší orientaci v kódu je vhodné senzor pojmenovat. Po potvrzení formuláře se automaticky vygeneruje potřebný kód, který se přidá do hlavičky programu (Obrázek 17). Hlavička musí vždy začínat na prvním řádku programu.



Obrázek 16: Okno pro nastavení senzorů


```

1  #pragma config(Sensor, S1, Dotyk, sensorTouch)
2  #pragma config(Sensor, S2, Zvuk, sensorSoundDB)
3  #pragma config(Sensor, S3, Svetlo, sensorLightActive)
4  /**!!Code automatically generated by 'ROBOTC' configuration wizard!!**/
5
6  task main()
7  {
8
9
10
11 }
12

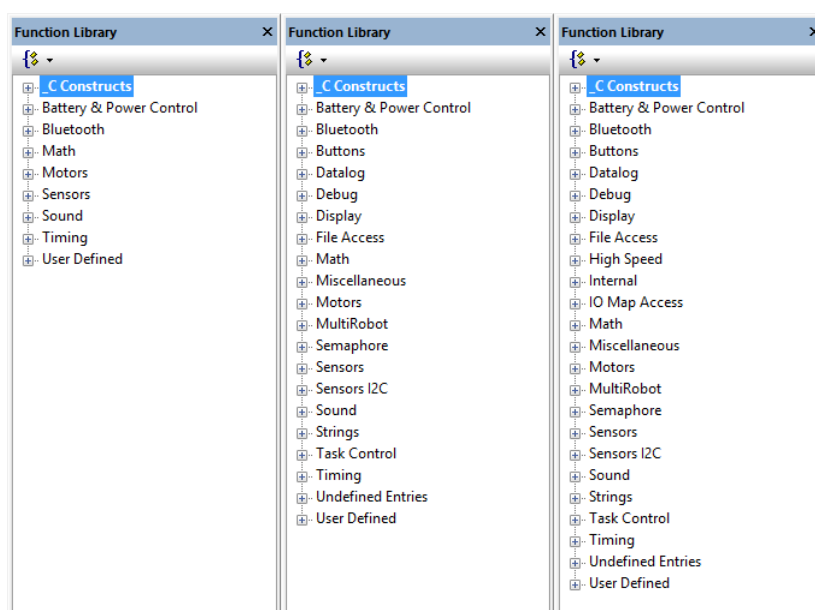
```

Obrázek 17: Deklarace senzorů v kódu a hlavní metoda „task main“

Psaní programu

Pro zápis programů je využíván jazyk, který je shodně jako program, pojmenován RobotC. Hlavní část programu je vždy umístěna v metodě „task main“ (Obrázek 17).

Editor umožňuje ruční zápis příkazů. Standardně je nutné si příkazy pro zápis kódu pamatovat. RobotC však disponuje pomocníkem ve formě knihovny funkcí. Knihovna zahrnuje všechny dostupné funkce a příkazy, které jazyk RobotC obsahuje. Příkazy není nutné do okna editoru ručně přepisovat. Vybraný příkaz stačí z knihovny do editoru jen přetáhnout a v editoru se vytvoří šablona daného příkazu, která se dále musí upravit, např. vložením konkrétních hodnot. Pokud uživatel nemůže v průběhu práce nalézt některé příkazy, je vhodné zkontrolovat nastavení uživatelské úrovně (Obrázek 18). Přepínání uživatelské úrovně je možné najít v hlavní nabídce v položce „Menu“ (Obrázek 15). Další příjemnou funkcí knihovny je možnost rychlého otevření nápovědy. Při dvojitým kliknutí na příkaz se rovnou otevře nápověda ke konkrétnímu tématu.



Obrázek 18: Obsah knihovny funkcí dle nastavení uživatelské úrovně (Zleva: Basic, Expert, Super User)

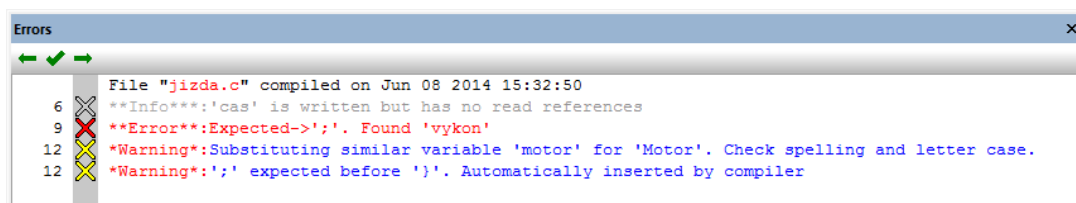
Samotný editor uživateli také usnadňuje práci se zápisem kódu. Barevně se zvýrazňují klíčová slova. Černou barvou zůstává text, který program nerozezná a je tedy možnost, že bude chybně zapsaný. Avšak černě značeny jsou i uživatelem vytvořené proměnné či funkce.

Překlad programu

Pro stažení programu do řídicí jednotky je potřeba jej nejprve přeložit. Uživatel má na výběr program jen přeložit (kompilovat) nebo ho rovnou přeložit a nahrát do řídicí jednotky. Během kompilace překladač ověřuje validitu kódu a upozorňuje na prohřešky proti syntaktickým pravidlům. V jazyku RobotC platí následující syntaxe:

- konec příkazu je označen středníkem,
- rozlišují se velká a malá písmena,
- všechny závorky jsou párové,
- bílé znaky (mezery, prázdné řádky) nemají efekt na funkčnost.

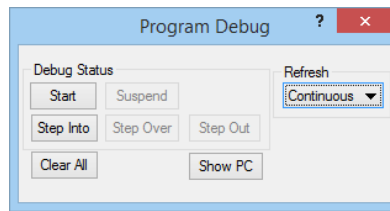
Pokud se v kódu budou vyskytovat chyby, tak se kompilace přeruší a chyby se vypíšou do informačního okna (Obrázek 19). Zde jsou varování a chybové hlášky rozděleny barevně podle závažnosti. Červeným textem jsou označeny ty chyby, které zabraňují kompilaci. Méně závažnější chyby neboli varování, jsou zvýrazněny žlutě. Tyto chyby buď program odstraní, případně je možné je při kompilaci ignorovat. Bíle označeny jsou zprávy informativního rázu, které nemají žádný dopad na kompilaci programu. Každý údaj je doplněn číslem řádku, kde se chyba vyskytuje, stejně tak se vyznačí přímo v editoru kódu, což usnadňuje jejich nalezení.



Obrázek 19: Informační okno

Po odstranění případných syntaktických chyb se program úspěšně nahraje do řídicí jednotky NXT a na obrazovce se objeví okno se základními možnostmi pro ovládání chodu programu (Obrázek 20). Uživatel zde může program rovnou spustit a nemusí tedy program

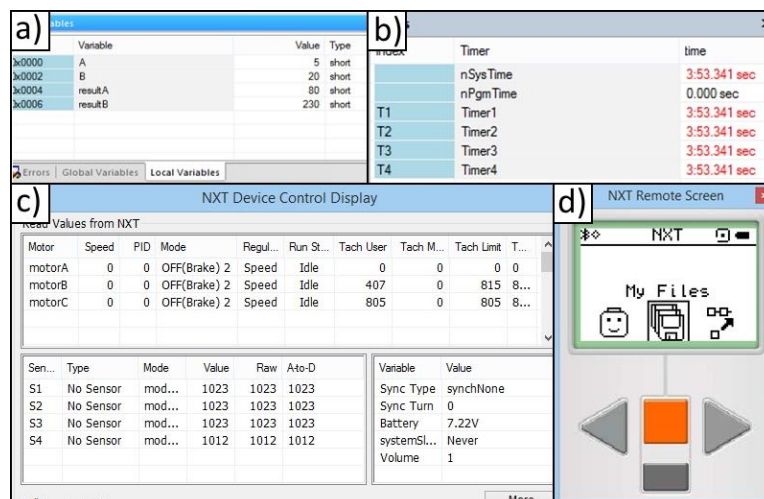
spouštět na řídicí jednotce. Je zde také možnost vstoupit do běhu programu a tím ho ručně krokovat, což je užitečné při ladění programu.



Obrázek 20: Okno pro základní ladění programu

Ladění programu

Laděním programu lze ověřovat činnost programu (případně i robotické sestavy). Za pomoci ladění se také mohou hledat případné chyby v programu. RobotC uživateli poskytuje velké možnosti, jak kontrolovat chod programu a hledat chyby. Umožňuje zobrazit několik ladících oken. Mezi nejzákladnější patří ty, které zobrazují aktuální hodnoty proměnných, časovačů a údajů spojených s motory a senzory (Obrázek 21a, b, c). Zajímavou vlastností, kterou běžné programovací prostředí nedisponují, je možnost zobrazit si virtuální řídicí jednotku (Obrázek 21d). Zde uživatel může vidět přesně to, co se děje na displeji jednotky. Navíc může jednotku ovládat pomocí aktivních tlačítek, stejně tak, jako kdyby tiskl tlačítka přímo na kostce. Tento způsob ovládání má význam především tehdy, pokud je řídicí jednotka připojena pomocí Bluetooth.



Obrázek 21: Náhled jednotlivých ladících oken – a) proměnné, b) časovače, c) motory a senzory, d) virtuální kostka

Hledání chyb velmi usnadňuje, pokud je kód zapsán přehledně. Na přehlednost má vliv především vhodné využívání tzv. bílých znaků v podobě vertikálních a horizontálních mezer.

Jako obecně doporučované pravidlo se bere umístění každého jednotlivého příkazu na nový řádek. Vnořené řádky by se měly odsazovat a jednotlivé bloky programu by se měly oddělovat prázdným řádkem. RobotC obsahuje možnost automatického formátu textu, který vhodně odsadí vnořené řádky. Pro snazší pochopení kódu je také vhodné využívat komentáře. Správně zapsaný kód má mnohem snazší čitelnost, díky níž se snáze hledají a opravují případné chyby (Obrázek 22).

<pre> task main() { // jizda vpred motor[motorC] = 100; motor[motorB] = 100; wait1Msec(4000); // jizda vzad motor[motorC] = -100; motor[motorB] = -100; wait1Msec(4000); } </pre>	<pre> task main(){motor[motorC] =100;motor[motorB]=100; wait1Msec(4000);motor [motorC] = -100; motor[motorB]=-100;wait1Msec (4000);} </pre>
--	---

Obrázek 22: Ukázka rozdílné přehlednosti kódu

3.3 STRUČNÉ PŘEDSTAVENÍ JAZYKA ROBOTC

Jazyk RobotC je založen na programovacím jazyku C [7]. Jedná se o strukturovaný procedurální programovací jazyk. Nepracuje tedy s objekty a třídami. Od standardní verze jazyka C se v některých ohledech odlišuje a není tak zpětně kompatibilní. Oproti běžnému programovacímu jazyku se RobotC vyznačuje specifickými příkazy pro ovládání robota (Tabulka 2).

Tabulka 2: Ukázka vybraných příkazů pro ovládání robota

Popis	Ukázka
Nastavení výkonu motoru	<code>motor[motorA] = 50;</code>
Čekání programu	<code>wait1Msec(1000);</code>
Vstupní hodnota ze senzoru	<code>SensorValue(ultrazvuk);</code>
Přehrání zvuku	<code>PlaySound(soundBeepBeep);</code>
Zobrazení textu na displej	<code>nxtDisplayTextLine(3, "Dobry den");</code>

V jazyce RobotC je možné průběh programu ovlivňovat větvením a cykly, neboli příkazy, které jsou souhrnně označeny jako řídicí struktury (Tabulka 3). Jazyk RobotC také umožňuje

využívat proměnné a konstanty, které je možné deklarovat v různých datových typech. Základní datové typy ukazuje Tabulka 4. Pro rozsáhlejší programy s opakujícím se kódem se hodí možnost vytvářet vlastní metody, které ukazuje Tabulka 5.

Tabulka 3: Řídící struktury

Popis	Ukázka
Neúplná podmínka	<pre>if(x <= 10) { //vnořený blok }</pre>
Úplná podmínka	<pre>if(x <= 10) { //vnořený blok } else { //vnořený blok }</pre>
Přepínač	<pre>switch(cislo) { case 1 { //vnořený blok } break; case 2: //příkaz break; default: //příkaz }</pre>
Cyklus s pevným počtem průchodů	<pre>for(int i = 0; i < 5; i++) { //vnořený blok }</pre>
Cyklus s podmínkou na začátku	<pre>while(x < 5) { //vnořený blok }</pre>
Cyklus s podmínkou na konci	<pre>do { //vnořený blok } while(x < 5);</pre>

Tabulka 4: Základní datové typy

Datový typ	Popis	Ukázka
bool (Boolean)	Logická hodnota – true, false	true
int (Integer)	Celé číslo od -32768 do 32767	0; 35; -117
float	Číslo s pohyblivou desetinnou částkou (reálné číslo)	0,14; 26,057
string	Textový řetězec	„Dobry den“

Tabulka 5: Vlastní metody

Popis	Ukázka
Vlastní metoda bez návratové hodnoty	<pre>void Metoda() { //vnořený blok }</pre>
Vlastní funkce s návratovou hodnotou	<pre>int Funkce(int x) { //vnořený blok return x; }</pre>
Vlastní dotaz (pro paralelní programování)	<pre>task Dotaz() { //vnořený blok }</pre>

3.4 UŽIVATELSKÁ PODPORA

V průběhu programování se lze snadno dostat do situace, kdy je potřeba využít nápovědu či vyhledat radu. Uživatelská podpora programu RobotC je jedním z důvodů, proč je tak úspěšný, neboť je opravdu precizní a propracovaná. Jako první se nabízí interní nápověda, která je již sama o sobě obsáhlá a zahrnuje jak obecné informace o softwaru RobotC, tak ukázky použití jednotlivých příkazů. Obdobná nápověda je dostupná také on-line [38]. K dispozici je dále množství ukázkových programů, které jsou již v rámci instalace nahrány do adresáře programu. Díky nim se lze například seznámit se základními principy ovládání robota.

Dalším velice užitečným pomocníkem je kurz "Teaching RobotC for LEGO Mindstorms". Kurz je vytvořen přímo vývojáři programu na Carnegie Mellon University, je ve své on-line verzi přístupný zdarma [39].

Vývojové prostředí RobotC je navíc rozšířené a oblíbené mezi uživateli. Díky tomu je k nalezení spousta návodů, tipů či videí. A využít lze i oficiální diskusní fórum [40].

4 VÝUKA PROGRAMOVÁNÍ NA SŠ POMOCÍ STAVEBNICE LEGO MINDSTORMS

Součástí diplomové práce je tvorba sady úloh pro podporu výuky programování na středních školách za pomoci robotické stavebnice LEGO Mindstorms NXT a vývojového prostředí RobotC. Samotné tvorbě sady úloh však musí předcházet stanovení jejího rozsahu a především cílů, které se za pomoci sady mají splnit. Nejprve je tedy nutné zjistit požadavky na výuku programování, kterou udávají rámcové vzdělávací programy (dále jen RVP). RVP jsou dokumenty vytvářené na státní úrovni, ve kterých jsou ustanoveny obecné cíle vzdělávání, vymezeny oblasti vzdělávání, jejich obsahy a očekávané výsledky vzdělávání.

4.1 VÝUKA PROGRAMOVÁNÍ V RVP PRO SŠ

V České republice nalezneme velké množství typů středních škol s různými zaměřenými: gymnázia, lycea, akademie, konzervatoře, odborné školy zakončené maturitní zkouškou a odborné školy s výučním listem. Tomu odpovídá i celkový počet vytvořených dokumentů RVP. Pro střední odborné vzdělávání bylo vytvořeno celkem 281 RVP, k tomu lze přičíst i dva RVP pro gymnázia a gymnázia se sportovní přípravou [41]. Cílem práce není provést analýzu všech variant RVP. Proto byl vybrán vzorek 52 RVP (Příloha 1), které byly prostudovány a poté rozděleny dle přístupu k výuce programování do tří kategorií: RVP pro čtyřletá gymnázia a vyšší stupně víceletých gymnázií, RVP pro střední odborné vzdělávání bez zaměření na informační technologie a RVP pro střední odborné vzdělávání zaměřené na informační technologie. Jednotlivé kategorie jsou dále přiblíženy.

4.1.1 GYMNÁZIA

V RVP pro čtyřletá gymnázia, vyšší stupně víceletých gymnázií a gymnázia se sportovní přípravou se výuka programování nachází ve vzdělávací oblasti Informatika a informační a komunikační technologie, konkrétně ve vzdělávacím obsahu Zpracování a prezentace informací, jehož součástí je učivo týkající se algoritmizace úloh [42]. Studenti by se měli seznámit se základními vlastnostmi algoritmu a jeho rolí v programování, věnují se také zápisu algoritmu a úvodu do programování.

Z celé koncepce vzdělávací oblasti je patrné, že na samotnou výuku programování není kladen hlavní důraz a tato disciplína je zařazena spíše s ohledem na komplexní prohloubení úrovně informační gramotnosti. Vzhledem k tomu, že obsah učiva v rámci úvodu

do programování není blíže specifikován, umožňuje tato formulace školám individualizovat výuku dle svých potřeb.

Na celkovou výuku informačních technologií je v RVP za celou dobu studia stanovena týdenní časová dotace 4 hodiny. Možnosti pro výuku programování tedy nejsou příliš široké. Lze však využít část z disponibilní týdenní časové dotace, která má souhrnnou výši 26 hodin. Na gymnáziích se lze tak setkat s programováním jako volitelným předmětem. Příkladem mohou být tyto školy: Gymnázium Brno třída Kapitána Jaroše [43], Gymnázium Sokolov [44] a Gymnázium Jaroslava Vrchlického Klatovy [45].

4.1.2 STŘEDNÍ ODBORNÉ VZDĚLÁVÁNÍ BEZ ZAMĚŘENÍ NA INFORMAČNÍ TECHNOLOGIE

RVP vytvořené pro střední odborné vzdělávání se liší s ohledem na jednotlivé obory vzdělání. Po prostudování několika odlišných RVP bylo zjištěno, že se jednotlivé obory v rámci výuky informačních technologií příliš neliší. Rozdílné jsou pouze různé časové dotace pro výuku. Týdenní hodinová dotace se pohybuje v rozmezí 2 až 8 hodin za celou dobu studia.

V RVP určených pro školy bez zaměření na informační technologie se nevyskytuje výuka samotného programování a převážně se objevuje učivo zaměřené na algoritmizaci. Cílem je, aby studenti ovládali principy algoritmizace úloh, sestavovali algoritmy a s jejich pomocí řešili konkrétní úlohy. V některých RVP pro střední vzdělání s výučním listem se algoritmizace ani nevyskytuje (Příloha 1).

Je však třeba upozornit na skutečnost, že dané RVP zahrnují různé vzdělávací obory a je tedy možné, že se na konkrétních školách výuka programování objevuje. V rámci oborových specializací se např. ve strojírenských oborech nachází programování CNC a CN techniky. Příkladem může být učební plán pro obor strojírenství na Střední průmyslové škole, střední odborné škole a středním odborném učilišti Nové Město nad Metují [46] nebo učební plán pro obor strojírenství na Střední průmyslové škole strojnické v Plzni [47]. S programováním se lze setkat i v rámci volitelných předmětů, např. na Obchodní akademii Plzeň [48].

4.1.3 STŘEDNÍ ODBORNÉ VZDĚLÁVÁNÍ ZAMĚŘENÉ NA INFORMAČNÍ TECHNOLOGIE

Jedná se o střední školy, kde je kladen větší důraz na výuku informačních technologií. Příkladem může být RVP pro obor Informační technologie [49]. V takto zaměřených školách

se zpravidla nevyučuje pouze jeden předmět zaměřený na informační technologie, ale nachází se zde více okruhů, které se věnují konkrétním oblastem. Například programové vybavení, hardware či sítě. Je zde možné najít také obsahový okruh programování a vývoj aplikací, kde je učivo zaměřeno na programování od algoritmizace přes strukturované programování. Dále se vyskytuje objektově orientované programování, jazyk SQL a tvorba statických a dynamických webových stránek.

Na výuku programování je vyhrazena týdenní časová dotace 8 hodin. Což je dvojnásobek časové týdenní dotace určené pro kompletní výuku informačních technologií na gymnáziích. Je tedy patrná velká diference. Při detailním pohledu na RVP určený pro střední odborné vzdělávání se zaměřením na informační technologie, se časová nevyváženost s ostatními typy škol projevuje ještě více. Celková týdenní časová dotace všech okruhů, které souvisí s informatikou, je 35 hodin během celé doby studia. V rámci volných disponibilních hodin je možné tento rozsah zvýšit až o dalších 39 hodin. Vzhledem k zaměření daných škol se minimálně část těchto hodin využívá pro další výuku informatiky či programování.

Příkladem mohu být učební plány pro studijní obor Informační technologie, které jsou zařazeny ve školních vzdělávacích programech těchto škol (čísla v závorkách představují počet hodin zaměřených na informační technologie v porovnání s celkovým počtem disponibilních hodin): Střední škola - Podorlické vzdělávací centrum (21/39) [50], Střední průmyslová škola elektrotechniky a informatiky Ostrava (21/33) [51] a Střední průmyslová škola Jihlava (13/33) [52].

Je třeba podotknout, že časová dotace v RVP určených školám se zaměřením na informační technologie je svým rozsahem ojedinělá a poskytuje maximální možný prostor pro výuku tohoto oboru. Programování se však vyskytuje také v RVP určených pro střední odborné školy s odlišným zaměřením, například technická lycea [53]. Byť samozřejmě v těchto případech je časová dotace i náročnost výuky nižší.

4.2 SOUHRN INFORMACÍ O VÝUCE PROGRAMOVÁNÍ NA SŠ

Výuka programování v RVP pro střední vzdělávání je v mnoha případech zredukována pouze na algoritmizaci. Pojem algoritmus se vyskytuje převážně v rámci programování, avšak principy algoritmizace jsou použitelné pro všechny oblasti lidské činnosti. Například

v matematice je algoritmizace využívána velmi často, jako postup při řešení početních příkladů. Zmínka o algoritmizaci se v oblasti matematického vzdělávání nachází jen v RVP pro gymnázia [42]. A to jen jako jedno z cílových zaměření vzdělávací oblasti Matematika a její aplikace: vytváření zásoby matematických pojmů, vztahů, algoritmů a metod řešení úloh a k využívání osvojeného matematického aparátu.

Využívání algoritmizace má také vliv na celkový rozvoj studenta. Schopnost algoritického myšlení rozvíjí představivost a abstraktní myšlení studentů, neboť akce „vydání pokynu“ nemusí být bezprostředně následována reakcí „splněním úkolu“ [54]. Algoritmizace nachází uplatnění napříč různými studijními obory a měla by tedy být zastoupena ve všech RVP. Proto může být překvapující úplné opomenutí algoritmizace v některých RVP.

Cíle výuky algoritmizace jsou koncipovány s ohledem na fakt, že základem výuky algoritmizace by mělo být samotné algoritické myšlení, neboli způsobilost řešit úlohy pomocí algoritmizace [42]. Konkrétní cíle vzdělávání jsou ve všech zkoumaných RVP podobné a je možné je shrnout cílem, který se vykytuje ve všech RVP pro odborné vzdělávání bez zaměření na informační technologie:

Student:

- ovládá principy algoritmizace úloh a sestavuje algoritmy řešení konkrétních úloh (dekompozice úlohy na jednotlivé elementárnější činnosti za použití přiměřené míry abstrakce).

Cíl poskytuje pouze nejasnou představu o tom, jakým způsobem by měla být vedena výuka. Není zde nijak upřesněno, zda by měli studenti využívat pro zápis algoritmů programovací jazyk. Lze tedy říci, že v těchto zkoumaných RVP se výuka programování nevyskytuje.

Ze vzorku zkoumaných RVP se programování nachází jen ve čtyřech z nich a to v RVP pro gymnázia, gymnázia se sportovní přípravou, technická lycea a informační technologie. V RVP pro gymnázia a gymnázia se sportovní přípravou se však vyskytuje pouze pojem v učivu: Úvod do programování. Nejsou zde uvedeny žádné další podrobnosti a nejsou zde vytyčeny ani vzdělávací cíle. Proto lze říci, že se zde programování vyskytuje, ale konkrétní struktura je v kompetenci dané školy a vyučujícího.

Výuka programování se podrobněji řeší pouze v RVP pro střední odborné vzdělávání v oborech Technické lyceum a Informační technologie. Programování je zde rozděleno na strukturované a objektově orientované. Nároky na studenty jsou však v obou případech rozdílné. RVP pro Technická lycea obsahují především požadavek na ovládnutí základů programování. Oproti tomu RVP pro Informační technologie se tématu věnuje mnohem detailněji.

Cíle učiva strukturovaného programování jsou:

Student:

- použije základní datové typy;
- použije řídicí struktury programu;
- vytvoří jednoduché strukturované programy.

Cíle učiva objektově orientovaného programování jsou:

Student:

- rozumí pojmům třída, objekt a zná jejich základní vlastnosti;
- použije jednoduché objekty.

Tyto cíle jsou již více informativní a jejich využitelnost pro tvorbu výuky je vyšší. Avšak ani v RVP pro Informační technologie není v cílech výuky zmínka o tom, že by student měl zapisovat algoritmy pomocí programovacího jazyka. Nachází se zde tak učivo algoritmizace a učivo programování, které ač jsou ve stejném okruhu, tak v důsledku chybí jejich propojení pomocí zapsaných cílů výuky.

Přestože je diplomová práce zaměřena na středoškolské vzdělávání, pro porovnání bylo prostudováno také RVP určené pro základní školy [55]. Cílem bylo nalézt výskyt algoritmizace tak, aby bylo možné zhodnotit, zda žáci přicházející na střední školu již mají její základy osvojeny. Na prvním stupni se v rámci vzdělávací oblasti Matematika a její aplikace nalézají učivo: písemné algoritmy početních operací v rámci matematiky. Ve vzdělávací oblasti Informační a komunikační technologie se vykytuje algoritmické myšlení v cílovém zaměření oblasti: schopnosti formulovat svůj požadavek a využívat při interakci s počítačem algoritmické myšlení.

Stav výuky algoritmizace a programování v českých RVP kontrastuje se Štátními vzdělávacími programy (dále uváděny jako ŠVP(SK)⁵), podle kterých se organizuje vzdělávání na Slovensku [56] [57] [58]. Předměty informatická výchova (informatika) a matematika jsou zde v jedné vzdělávací oblasti Matematika a práca s informáciami. V informatické výchově je přímo tematický okruh: Postupy, riešenie problémov, algoritmické myslenie. S algoritmickým přístupem se zde žáci taktéž prvně setkají na prvním stupni základní školy. Avšak ŠVP(SK) zaobírá se touto oblastí mnohem detailněji než český RVP.

Na prvním stupni základní školy žáci již řeší jednoduché algoritmy v dětském programovacím prostředí. Z hlediska tématu diplomové práce je zajímavé zařazení robotické stavebnice, s jejíž pomocí řeší žáci různorodé úlohy. Setkávají se tak nejen s okamžitým vykonáváním příkazů, ale také s vykonáváním posloupnosti dříve zadaných příkazů. Žáci si tak rozvíjejí schopnost abstrakce. Výuka algoritmů pokračuje také na druhém stupni, kde se již objevují např. proměnné, cykly a podmínky, řeší se míra efektivity různých způsobů řešení problémů.

Detailní popis výuky algoritmizace nalezneme také v ŠVP(SK) určených pro střední vzdělávání. Studenti se zabývají základy programování. Mimo znalostí příkazů, proměnných a řídicích struktur, je v základech programování kladen důraz i na ladění programů a hledání chyb.

Při prostudování ŠVP je na první pohled patrný rozdíl v přístupu k rozsahu vzdělávacích plánů. Slovenské programy jsou v oblasti výuky algoritmizace mnohem detailněji popsány. Algoritmizace a programování se v plánech vyskytují již od 1. stupně základní školy a dále v průběhu studia tak, aby na sebe jednotlivé okruhy navazovaly. Český systém je odlišný zejména v míře provázanosti algoritmizace na ostatní obory a celkové návaznosti jednotlivých okruhů. V českých RVP je celé pojetí výuky programování nedostatečně formulováno (výjimku částečně tvoří jen RVP pro informační technologie).

⁵ Označení ŠVP(SK) zvoleno z důvodu vyloučení záměny se školským vzdělávacím programem (ŠVP).

5 SADA ÚLOH

V návaznosti na teoretické poznatky získané z prostudovaných RVP bylo záměrem vypracovat sadu úloh tak, aby byla vhodná pro výuku úvodu do programování. Toto učivo je nejčastěji zahrnuto v učebním plánu pro první ročníky středních škol. Před tvorbou úloh je nutné stanovit cíle, kterých mají studenti za pomoci sady dosáhnout.

5.1 CÍLE PRO SADU ÚLOH

Podkladem pro vytvoření cílů sady úloh byly jak české RVP, tak slovenské ŠVP(SK). Důvodem využití i ŠVP(SK) byla nižší propracovanost výuky programování v RVP. Cíle jsou uzpůsobeny možnostem stavebnice LEGO Mindstorms a vývojovému prostředí RobotC. Cíle byly stanoveny se zaměřením na algoritmizaci a úvod do strukturovaného programování.

Nejprve byly stanoveny obecné cíle, ke kterým by měla sada úloh směřovat. Obecné cíle udávají, čeho se má při výuce dosáhnout v dlouhodobém horizontu. Jsou formulovány poměrně široce a všeobecně. Zpravidla také není možné dosáhnout těchto cílů stoprocentně [59].

Obecné cíle programování:

- rozvíjet schopnost algoritmického myšlení;
- zlepšit schopnost vyjadřování algoritmů pomocí programovacího jazyka;
- navrhovat a realizovat programy zadaných úloh.

Následně byly stanoveny konkrétní cíle výuky. Konkrétní cíle musí být naopak splnitelné a jejich výstup musí být ověřitelný. Nezaměřují se na učitele, ale na schopnost, kterou si studenti během výuky osvojí [59].

Konkrétní cíle výuky programování:

Algoritmizace

Student:

- navrhne algoritmus řešení daného problému;
- napíše algoritmus ve srozumitelné formální podobě;
- dokáže ověřit správnost algoritmu;
- řeší problémy pomocí algoritmizace;

- dokáže algoritmus zapsat pomocí programovacího jazyka;
- dokáže porozumět vytvořenému programu.

Řídící struktury, proměnné a datové typy

Student:

- řeší úlohy s použitím vhodných řídicích struktur, proměnných a datových typů;
- používá základní příkazy daného programovacího jazyka.

Ladění programu

Student:

- dokáže rozeznat a odstranit syntaktické chyby;
- opravuje chyby vzniklé za běhu programu;
- určí místa programu, na kterých může dojít k chybám za běhu programu.

5.2 TVORBA ÚLOH

Pro stanovené cíle byly následně vytvořeny praktické úlohy. Hned v počátku bylo potřeba se zamyslet nad motivací studentů. Faktory motivace k učení mohou být u studentů různé, nejsilnější však bývá pocit zadostiučinění při splnění obtížného úkolu a pocit úspěchu [59]. Stejný faktor však může být zároveň demotivační, pokud není dlouhodobě naplněn. Obecným pravidlem bývá, že jedince baví ty věci, ve kterých se mu daří a naopak. Při kontinuálním neúspěchu by tedy u studentů docházelo k podstatné ztrátě motivace.

Z toho důvodu byly úlohy v sadě tvořeny tak, aby úvodní úkoly byly jednoduché a zvládnutelné všemi studenty. Na druhou stranu vytvořené úlohy nesměly být příliš triviální, protože splněním příliš jednoduché úlohy se nedostaví pocit úspěchu a tím pádem se snižuje motivace pro další činnost. Postupně se náročnost jednotlivých úloh zvyšuje a na konci sady se vyskytují také úlohy, které již nemusí být splnitelné pro všechny studenty.

5.3 POUŽITÍ SADY ÚLOH

Při tvorbě úloh byla brána v potaz jejich posloupnost. To znamená, že jsou seřazeny od jednodušších až po ty náročnější. Dále se vyskytují rozšiřující úkoly, které mají za cíl především ukázat studentům další zajímavé možnosti programu a ještě více ztraktivnit výuku.

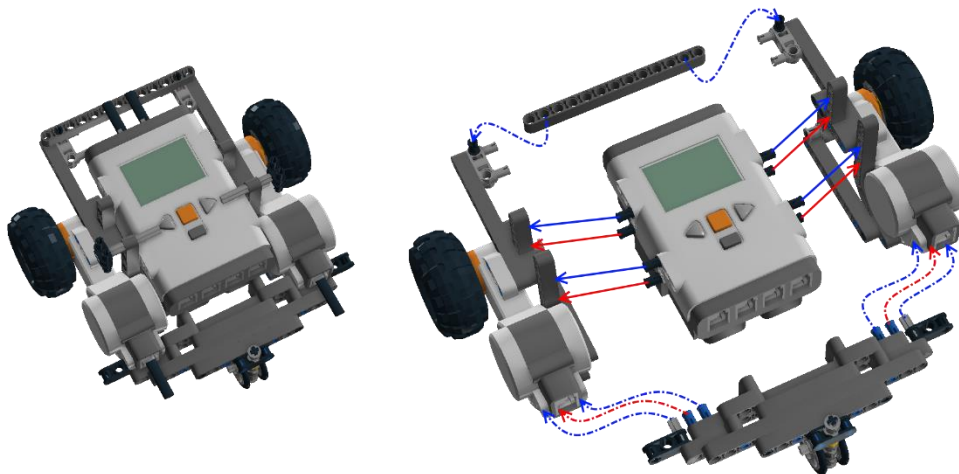
Úlohy jsou tvořeny tak, aby na jejich řešení mohla pracovat zhruba dvou až čtyřčlenná skupina studentů. Avšak velikost skupiny závisí především na množství stavebnic.

Před začátkem práce na řešení úlohy je potřeba, aby vyučující uvedl studenty do dané problematiky. V průběhu práce je poté jeho role především v tom, aby průběžně nastiňoval cestu vedoucí ke správnému řešení a konzultoval se studenty jejich postupy.

Konstrukce robota

Výchozí konstrukcí pro většinu úloh je robot označen jako „Express-Bot“ (Obrázek 23). Konstrukce je volně dostupná ze stránek: www.nxtprograms.com [60]. Oproti původní konstrukci byla provedena změna uchycení podvozku a zadního kola. Návod na stavbu upravené konstrukce lze najít na přiloženém DVD. Sestavení robota by nemělo trvat déle než 20 minut.

Úpravy pro jednotlivé úlohy následně spočívají zejména v zapojení senzorů a jen několika málo konstrukčních dílů. Konstrukce dovoluje rychlé a jednoduché nasazování senzorů. Výhodou také může být to, že lze v případě nutnosti jednoduše vyjmout celou řídicí jednotku. Rozebrání robota lze provést čtyřmi jednoduchými kroky, stejně tak opětovné složení (Obrázek 23).

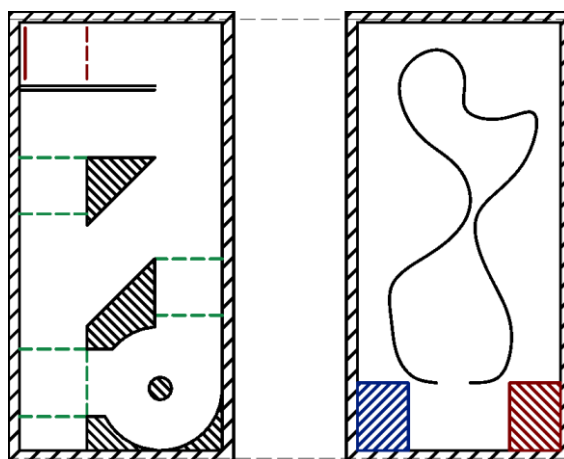


Obrázek 23: Rozložení a složení základního robota

Pracovní prostor

Pro správnou činnost robota je nezbytné připravit jeho pracovní prostor, ve kterém se bude pohybovat a kde bude vykonávat naprogramované úlohy. Pracovní prostor by měl být jednobarevný s hladkým povrchem, na který se pro jednotlivé úlohy umístí potřebné

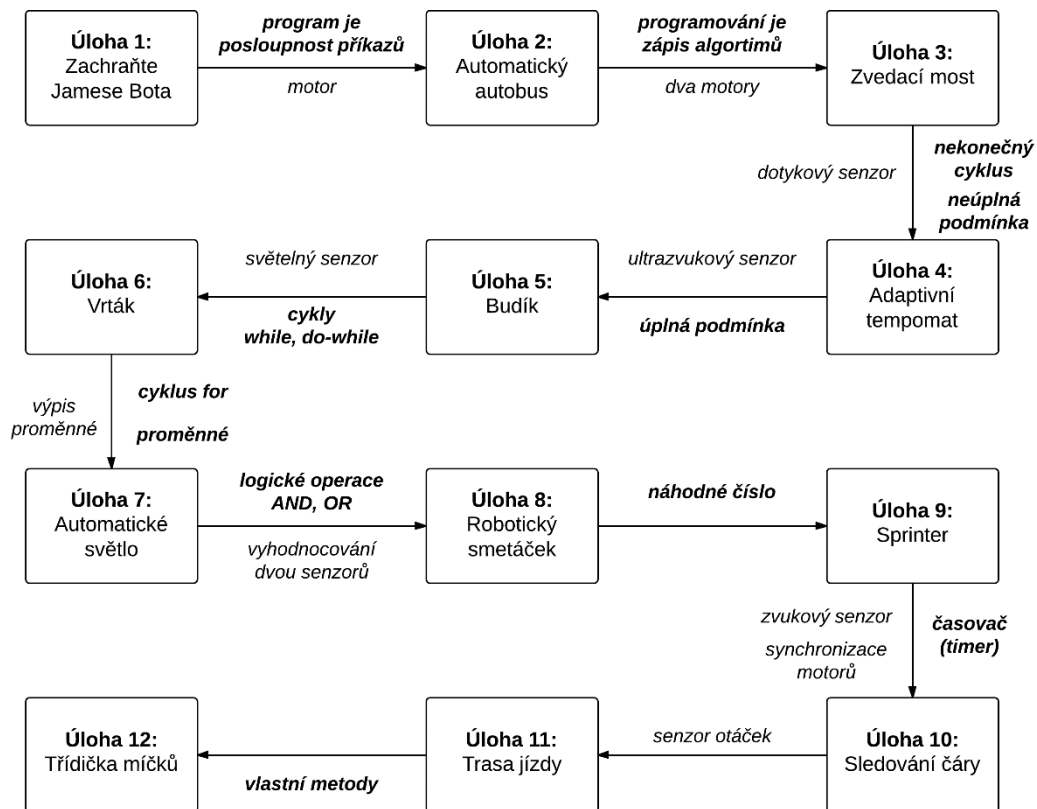
značky. Pro potřeby některých úloh mohou být vyžadovány stěny pro ohraničení pracovního prostoru. Není-li k dispozici pracovní prostor dle výše uvedené specifikace, je nutné některé řešení úloh upravit. Pro potřeby této diplomové práce byla použita deska s bílým hladkým povrchem z obou stran o rozměrech 1x2m s následujícím uspořádání vodících čar a překážek (Obrázek 24).



Obrázek 24: Schéma pracovní plochy

5.4 OSNOVA VYTVOŘENÉ SADY ÚLOH

Posloupnost jednotlivých úloh byla stanovena tak, aby s každou další vyřešenou úlohou studenti prohlubovali své znalosti. Stanovené cíle zahrnují dosažené znalosti z předešlých úloh a dochází k rozvoji požadovaných dovedností. Sada obsahuje 12 úloh zaměřených na rozvoj algoritmického myšlení a úvodu do programování s pomocí stavebnice LEGO Mindstorms. Pořadí jednotlivých úloh je odvozeno od jejich obtížnosti a použitých doplňkových senzorů (Obrázek 25). Časová náročnost celé sady úloh se pohybuje v rozmezí 12 až 15 běžných školních vyučovacích hodin (45 min). Při obvyklé dvou hodinové týdenní dotaci je potřebný čas na vypracování všech úloh 6 až 7 týdnů.



Obrázek 25: Struktura sady úloh

Úloha 1: Zachraňte Jamese Bota

První úkol je zaměřen především na to, aby se studenti seznámili s prostředím RobotC, konkrétně se způsoby zápisu a nahrávání programů do řídicí jednotky. Úloha byla zařazena na úvod, protože příkazy na ovládání motoru jsou jednoduché, snadno pochopitelné a názorně ukazují skutečnost, že program je tvořen posloupností příkazů.

Úloha 2: Automatický autobus

Druhá úloha rozšiřuje téma ovládání motorů. Studenti již musí být schopni ovládat dva motory, které slouží k pohybu a otáčení robota. Úloha zahrnující ovládání motorů je záměrně zařazena hned v úvodu sady, neboť tuto dovednost využijí studenti i nadále. Úloha se také soustřeďuje na rozvoj algoritmického myšlení, kdy studenti musí plánovat trasu autobusu, kterou následně zapisují do programu.

Úloha 3: Zvedací most

Třetí úkol seznamuje studenty jednoduchým stylem hned s několika důležitými prvky v ovládání robotů. Jsou jimi: nekonečný cyklus, deklarace připojeného senzoru a příkaz

pro získání hodnot ze senzoru. Proto je úloha zařazena na začátek sady. Jako první senzor byl záměrně vybrán dotykový, neboť jeho fungování je pro studenty snadno pochopitelné. Pro detekci stisku je v úloze využita neúplná podmínka.

Úloha 4: Adaptivní tempomat

V předchozí úloze se studenti setkali s neúplnou podmínkou, tato úloha tedy doplní látku také o podmínku úplnou. Příklad adaptivního tempomatu byl vybrán právě pro ideální možnost, jak předvést rozdíl mezi oběma typy podmínek. Pro vyhodnocování vzdálenosti je použit ultrazvukový senzor.

Úloha 5: Budík

Studenti se již seznámili s využitím nekonečného cyklu zapsaného pomocí cyklu s podmínkou na začátku. V této úloze studenti pozmění zápis příkazu tak, aby se cyklus ukončoval. V úloze se také vyskytuje cyklus s podmínkou na konci. Jejich rozdíl je v úloze názorně ukázán. Studenti se také poprvé setkají se světelným senzorem.

Úloha 6: Vrták

V minulé úloze se studenti setkali s podmíněnými cykly. Aby se ucelilo téma cyklů, byla zařazena také úloha, ve které pracují s cyklem s pevným počtem průchodů. V úloze studenti využijí i vnořený cyklus a vytváří celočíselné proměnné a vypisují je na displej.

Úloha 7: Automatické světlo

Studenti využívají logické operátory (AND a OR) pro správu vstupních hodnot ze dvou senzorů najednou. Také zde pracují s proměnou typu boolean. Zadání úlohy opět pracuje s podmínkami a navíc je rozšiřuje o vícenásobné vnoření. Důvodem zařazení byla především snaha seznámit studenty s prací se dvěma senzory najednou.

Úloha 8: Robotický smetáček

Úloha shrnuje a propojuje předešlé poznatky a využívá znalosti, které studenti získali v předchozích úkolech. Studenti mohou rozvinout dovednosti týkající se ovládání a pohybu robota a dozvědí se více o možnostech generování náhodného čísla.

Úloha 9: Sprinter

Je zde použit zvukový senzor, se kterým se zatím studenti neseťkali. V důsledku se však nejedná o žádnou novinku, neboť jeho ovládání se v podstatě neliší od ostatních senzorů. Podmínkou správného použití je pouze znalost měřené veličiny. Úloha je delší, ale je rozdělena na menší části, které studenty postupně směřují k výsledku. Úloha prohlubuje znalost pokročilejších funkcí RobotC, jako je časovač a synchronizace motorů.

Úloha 10: Sledování čáry

Sledování stanovené trasy je způsob, jakým jsou roboti i v praxi často využíváni. V úloze si tedy studenti mohou otestovat, jestli daný úkol dokážou naprogramovat. Zařazení v druhé polovině osnovy je dáno zejména využitím senzoru otáček pro určení délky jízdy robota. Vyšší obtížnost může být způsobena větší mírou abstrakce a řešením chování robota v průběhu vykonávání úkolu.

Úloha 11: Trasa jízdy

Předposlední úloha rozvíjí téma senzoru otáček. V předchozí úloze studenti využívali senzor otáček jen pro měření vzdálenosti přímého pohybu. V této úloze ho využijí pro zadávání úhlu natočení. Důležitým prvkem úlohy je tvorba vlastních metod. Jedná se již o pokročilejší programování, které vyžaduje větší představu o tom, jak program funguje. Z toho důvodu byla úloha zařazena na závěr sady.

Úloha 12: Třídička míčků

Vzhledem ke své náročnosti je úloha zařazena na závěr, přestože se zde již nevyskytuje žádný nový prvek. Obtížnost spočívá zejména v pochopení a posloupnosti jednotlivých úkonů. Úloha má svůj význam ve schopnosti otestovat porozumění studentů vytvořenému programu.

5.5 ZADÁNÍ A METODICKÉ POKYNY K VYTVOŘENÝM ÚLOHÁM

Zadání jednotlivých úloh obsahuje popis, klíčové pojmy, strukturu úkolů, předpokládanou časovou náročnost a pokyny pro učitele. Na přiloženém DVD jsou dále k dispozici zadání pro studenty, listy pro učitele, konstrukce a řešení všech úloh.

5.5.1 ÚLOHA 1: ZACHRAŇTE JAMESE BOTA

V úloze studenti použijí příkazy na ovládání motoru. Vytvoří jednoduchou sekvenci, která umožní otevřít dveře a po 3 vteřinách je opět zavřít. Dosáhnou toho pomocí kladného, nulového a záporného výkonu.

Cíle úlohy

Student:

- napíše kód pro ovládání motoru v prostředí RobotC;
- dokáže kompilovat program a nahrát jej do řídicí jednotky NXT;
- řeší program jako posloupnost příkazů.

Klíčové pojmy, příkazy v RobotC

- nastavení výkonu motoru – motor[];
- čekání programu – wait1Msec().

Zadání

Záchranná mise

Agent Jamese Bota zajali zrádci a uvěznili ho ve vězení. Klíč k jeho cele neexistuje, pro vstup je potřeba přijít na tajný kód. Použijte vašeho robota, abyste mu pomohli ze zajetí. Na motor, který ovládá mechanismus otevírání dveří, se můžete napojit pomocí kostky NXT. V programu RobotC zapište požadovaný kód a otevřete dveře.

Utajení útěku

Upravte program tak, aby zajatci nemohli tak snadno přijít na to, že James Bot byl osvobozen. Pokusíte se je obelstít tím, že dveře otevřete, necháte je otevřené pouze tři vteřiny a následně je zase zavřete.

Předpokládaná časová náročnost

- konstrukce (vězení): 10 min (zajistí učitel);
- Záchranná mise: 10 min;
- Utajení útěku: 5 min;
- **celkem: 15 min.**

Pokyny pro učitele

Konstrukci k první úloze vytvoří učitel a studentům ji představí společně se zadáním úkolu.

Jedná se o první ukázkovou úlohu, při níž se studenti poprvé seznámí s prostředím RobotC i s celou stavebnicí LEGO Mindstorms. V počátku je třeba studentům ukázat, jak pracovat s programem RobotC.

Pozornost je potřeba věnovat především směru otáčení motoru, neboť při špatném směru otáčení motoru se dveře zaklíní o konstrukci a bude tak zablokován chod motoru. Proto je dobré zadávat nízký výkon. Hodnoty v řešení úlohy jsou pro zvednutí ze zcela zavřené pozice následující: výkon 20%, čas 2000ms. Pro zavření se musí jen otočit směr (znaménko při zadání výkonu).

Konstrukce vězení neobsahuje žádné díly, které jsou potřeba pro stavbu základní konstrukce pojízdného robota, a proto mohou být postaveny současně.

5.5.2 ÚLOHA 2: AUTOMATICKÝ AUTOBUS

Studenti ovládají pohyb robota pomocí dvou motorů (jízda v přímém směru i zatáčení). Vytvoří a zapíše algoritmus pro průjezd městem včetně zastavování ve stanicích.

Cíle úlohy

Student:

- používá základní příkazy pro ovládání motorů tak, aby s robotem mohl pohybovat v přímém směru i zatáčet;
- navrhne a popíše algoritmus pro jízdu automatického autobusu;
- navržený algoritmus zapíše pomocí programovacího jazyka RobotC.

Klíčové pojmy, příkazy v RobotC

- nastavení výkonu motoru – motor[];
- čekání programu – wait1Msec().

Zadání

Slavnostní první jízda

Slavnostní otevření nové plně automatické autobusové linky se blíží. Autobusy nepotřebují řidiče, ale potřebují naprogramovat svoji cestu. A to bude váš úkol. První slavnostní jízda bude ze stanice Hlavní nádraží do zastávky Na Růžku. V zastávce Na Růžku se slavnostní jízda ukončí. Zajistěte cestujícím komfortní jízdu a zvolte přiměřenou rychlost jízdy.

Celá trasa

První jízdu si všichni pasažéři pochvalovali, proto jste byli pověřeni vytvořením kompletního jízdního řádu pro celou zelenou linku. Autobus musí navštívit všechny tři stanice (Hlavní nádraží, Na Růžku, Úzká), navíc musí skončit na stanici, ze které vyjel. Vyhodnoťte, jakou trasou a v jakém pořadí stanic bude linku nejvýhodnější vést. Návrh musí projít schválení představitelů města, zpracujte jej a poté jim ho představte. Až následně jej zapište do programu a zrealizujte

Slepá ulice (rozšiřující úkol)

Vznikla nová stanice ve Slepé ulici. Je tedy nutné, abyste program pro řízení autobusu upravili. Autobus bude muset udělat odbočku do stanice. Po naložení cestujících se opět napojí na zelenou linku. Jak autobus vyjede ze stanice je jen na vás.

Předpokládaná časová náročnost

- konstrukce (základní robot): 20 min;
- Slavnostní první jízda: 10 min;
- Celá trasa: 25 min;
- Slepá ulice (rozšiřující úkol): 15 min;
- **celkem: 70 min** (bez rozšíření: 55 minut).

Pokyny pro učitele

Dle návodu dostupného na přiloženém DVD nejprve studenti postaví základního robota.

Před začátkem prvního úkolu je užitečné studentům poskytnout čas na to, aby si mohli otestovat, jak daleko robot pojedí při zadání konkrétních hodnot (například při 50% výkonu po dobu jedné vteřiny robot ujede přibližně 24 cm). Pokud si hodnoty poznamenají, usnadní

se jim další práce. Do stejného programu je poté možné otestovat i hodnoty potřebné pro otáčení robotem. Například otočka o 180° kolem své osy při výkonech motorů 50 a -50% trvá jednu vteřinu. Otočka o stejný úhel kolem zastaveného kola trvá při výkonu aktivního motoru 50% dvojnásobně dlouho, tedy dvě vteřiny.

Studenty je vhodné upozornit, že pohyb robota není nikdy naprosto přesný. A proto nemusí své jízdy propočítávat na milimetry, ale stačí přibližně dodržovat vymezenou trasu. V našem případě tedy není velká chyba, pokud robot lehce vyjede z ohraničené trasy.

Zatáčení robota je přesnější, pokud se před otočkou zastaví (především to platí pro otáčení kolem vlastní osy). Délka kódu tím sice vzroste, ale jízda bude přesnější.

V úloze mají studenti za úkol jezdit s robotem po ulicích v naznačeném městě. Řešení úkolů platí pro plán města, který byl vytvořen pro účely dané práce. Plány města však mohou být libovolné a upravené podle možností a prostoru v konkrétních podmínkách výuky. Samotné ulice je možné vyznačit různorodě, mohou posloužit knížky, přepravky od stavebnic a další předměty. Praktickým řešením je na vyznačení cest použít černou elektrikářskou pásku. Úkoly je možné dále modifikovat a lze například umístit na nejjednodušší trasu uzavírku.

5.5.3 ÚLOHA 3: ZVEDACÍ MOST

Studenti využijí dotykové senzory k tomu, aby mohli ovládat zvedací most. Jeden senzor bude sloužit pro zvednutí mostu, druhý pro sklopení.

Cíle úlohy

Student:

- dokáže deklarovat zapojený senzor v prostředí RobotC;
- použije vhodný zápis podmínky, aby detekoval stisk dotykového senzoru;
- popíše činnost nekonečného cyklu.

Klíčové pojmy, příkazy v RobotC

- neúplná podmínka – if;
- nekonečný cyklus – while(true);
- vstupní hodnota ze senzoru – SensorValue().

Zadání

Zvedání mostu

V jednom českém městě se rozhodli inspirovat slavným londýnským mostem Tower Bridge a chtějí si postavit zvedací most. Aby se most mohl otevřít pokaždé, když pod ním pojede loď, musí mít obsluha možnost toto jednoduše provést, kdykoliv bude potřeba.

NXT tlačítka (rozšiřující)

Most také můžeme ovládat pomocí tlačítek umístěných přímo na kostce NXT. Upravte program tak, aby se most zvedal stiskem šipky doleva a pokládal šipkou doprava.

Předpokládaná časová náročnost

- konstrukce (most): 10 min;
- Zvedání mostu: 15 min;
- NXT tlačítka (rozšiřující): 5min;
- **celkem: 30 min** (bez rozšíření: 25 minut).

Pokyny pro učitele

Jedná se o rychlou a jednoduchou úlohu, která má studenty seznámit s možnostmi, jak využívat zpětnou vazbu od senzorů. Důležitou součástí jsou řídicí struktury, konkrétně větvení s neúplnou podmínkou a v úloze se také vyskytuje nekonečný cyklus. Nekonečné cykly jsou v ovládní robota často využívanou možností. Je tedy vhodné studentům tento cyklus představit. Mělo by zaznít také sdělení, že cyklus neustále ověřuje platnost zadané podmínky a je prováděn po celou dobu zapnutí řídicí jednotky.

Studenti se prvně setkají s příkazem na zjištění hodnoty ze senzoru „SensorValue“. Je důležité říci, že tento příkaz je stejný pro libovolné senzory. A pouze deklarací senzoru zaručíme, že budeme získávat odpovídající hodnoty.

Kostku není potřeba vyjmát ze základního konstrukce robota, dotykové senzory i třetí motor jdou zapojit do základní konstrukce.

Konstrukce mostu neobsahuje žádné díly, které jsou potřeba pro stavbu základního robota, a proto mohou být postaveny současně.

V úloze je důležitý směr otáčení motoru. Záporné hodnoty výkonu most zvedají, kladné pokládají. Vzhledem k relativně vysoké hmotnosti konstrukce mostu je nutné omezit výstupní výkon na hřídeli motoru z důvodu možného poškození ozubených kol. Při výkonu 10% se most zvedne do vertikální polohy za 4 vteřiny.

5.5.4 ÚLOHA 4: ADAPTIVNÍ TEMPOMAT

Studenti si vyzkouší napodobit funkci adaptivního tempomatu. Pomocí ultrazvukového senzoru budou kontrolovat vzdálenost od překážky, a pokud bude kratší než zvolená mez, tak robot zastaví.

Cíle úlohy

Student:

- pomocí praktického příkladu vysvětlí rozdíl mezi úplnou a neúplnou podmínkou;
- dokáže využít vstupní hodnoty ze senzoru pro ovládání výkonu motorů;
- zkontroluje průběh programu pomocí krokování a údajů o řídicí jednotce.

Klíčové pojmy, příkazy v RobotC

- úplná podmínka – if-else;
- vstupní hodnota senzoru ovládá výkon motoru.

Zadání

Zastavení a opětovné rozjetí

Moderní automobily jsou vybaveny systémem, který auto sám dokáže zastavit a opět rozjet. Je to šikovný pomocník především pro pohyb po městě a v kolonách. Vaším úkolem bude podobný systém nainstalovat do vašeho auta. Cílem je, aby při jízdě po městě, kdy jedete rychlostí 50 km/h (v našem případě 50% výkonem), auto zastavilo ve chvíli, kdy se před ním objeví překážka. Pokud překážka zmizí, auto se opět rozjede.

Plynulé brzdění a rozjezd

V současném stavu auto zastaví, ale velmi nepohodlně a prudce. Proto vylepšete funkci tempomatu tak, aby auto zpomalovalo průběžně. Čím blíže bude k překážce, tím pojede pomaleji.

Omezení rychlosti (Rozšiřující)

Upravte program tak, aby robot plnil následující podmínky:

1. přiblíží-li se k překážce na 10 cm a méně, zcela zastaví;
2. bude-li ho od překážky dělit více než 50 cm, jeho výkon bude konstantně na 50%;
3. pohybuje-li se v rozmezí 10-50 cm od překážky, jeho výkon bude závislý na hodnotě senzoru.

Předpokládaná časová náročnost

- konstrukce (základní robot + ultrazvukový senzor): 5 min;
- Zastavení a opětovné rozjetí: 15 min;
- Plynulé brzdění a rozjezd: 10 min;
- Omezení rychlosti (rozšiřující): 15 min;
- **celkem: 45 min** (bez rozšíření: 30 minut).

Pokyny pro učitele

Hlavní důraz je kladen na použití úplné podmínky. Rozdíl mezi úplnou a neúplnou podmínkou je možné názorně předvést právě na příkladu adaptivního tempomatu. Při použití neúplné podmínky nedojde k úplnému zastavení robota, což zapříčiní kolizi s překážkou. Studenti by měli umět vysvětlit, proč tomu tak je.

Je třeba studenty upozornit, že více aktivních ultrazvukových senzorů v jedné místnosti se může navzájem rušit.

Navázat na třetí úkol lze pomocí otázky: „Jak rychle jede vaše auto, když nevidí žádnou překážku?“. Studenti by měli prakticky otestovat, že auto jede rychleji. Dále lze pomocí ladících nástrojů zkontrolovat, jaký výkon se nastavuje na motor z ultrazvukového senzoru.

5.5.5 ÚLOHA 5: BUDÍK

Studenti si naprogramují budík, který bude zvonit (přehrávat zvuk), jakmile světelný senzor, pracující v pasivním režimu, zaznamená vyšší intenzitu světla. Budík půjde zastavit dotykovým senzorem.

Cíle úlohy

Student:

- vysvětlí rozdíl mezi cykly while a do-while;
- dokáže přehrát zvuk;
- objasní pojem prázdný programový blok (na příkladu přehrávání zvuku).

Klíčové pojmy, příkazy v RobotC

- cyklu s podmínkou na konci – do-while;
- přehrání zvuku – PlaySound;
- prázdný programový blok.

Zadání

Vstávání

Na internetu jste četli myšlenku, že zdravé vstávání je takové, kdy se člověk probouzí se svítáním a chcete si ji na sobě ověřit. Musíte si tedy sestavit takový budík, který vám s experimentem pomůže. Večer jej prostě zapnete a on bude čekat tak dlouho, dokud nebude dostatečné světlo na to, aby vás vzbudil. Intenzitu světla monitorujte světelným senzorem. Nastavte zároveň zvuk, kterým vás ráno bude budit a nezapomeňte ani na možnost vypnutí budíku pomocí stisknutí dotykového senzoru.

Grafika budíku (rozšiřující)

Chcete se svým novým budíkem pochlubit přátelům, takže byste na něm rádi ještě něco vylepšili. Co takhle zobrazit vhodný obrázek na displej? Zkuste na něm zobrazovat postavičku, která bude v noci mlčet a při probuzení vám popřeje dobré ráno. Obrázek postavičky naleznete v kostce NXT.

Předpokládaná časová náročnost

- konstrukce (základní robot + dotykový senzor): 5 min;
- Vstávání: 15 min;
- Grafika budíku (rozšiřující): 5 min;
- **celkem: 25 min** (bez rozšíření: 20 minut).

Pokyny pro učitele

Úloha se zaměřuje na využití cyklů s podmínkou na začátku a na konci. V programu jsou použity oba, aby byl patrný rozdíl. Oba cykly lze výhodně testovat jak v rámci ladících nástrojů, tak přímo za běhu programu. Stačí zkusit různé kombinace zakrytí světelného

senzoru a tisknutí dotykového senzoru. Studenti si ověří, že i při okamžitém stisknutí tlačítka pro vypnutí buzení dojde ke krátkému zvukovému signálu.

Rozšiřovací úloha přidává jen grafiku a textové informace na displej. Grafika je v příkladu řešena za pomoci vykreslení dvou ikon, které jsou součástí firmwaru RobotC. Zadání lze modifikovat a kreslit jen jednoduché obrazce.

5.5.6 ÚLOHA 6: VRTÁK

V úloze studenti na základní konstrukci robota přimontují vrták. Robot poté podle zadání vrtá na daných místech. Pohyb i práce robota jsou řízeny pomocí cyklů s pevným počtem průchodů.

Cíle úlohy

Student:

- popíše části, ze kterých je tvořen příkaz pro cyklus s pevným počtem průchodů;
- používá vnořené cykly;
- dokáže vytvořit proměnou datového typu integer;
- dokáže na displeji zobrazit celočíselnou proměnou.

Klíčové pojmy, příkazy v RobotC

- cyklus s pevným počtem průchodů – for;
- vnořené cykly;
- zobrazení textu a celočíselné hodnoty na displej;
- proměnná, celočíselný datový typ integer, inkrementace.

Zadání

Nový vrták

Firma vyrábějící vzduchotechnické zařízení potřebuje vyvrtat díry do plechových desek používaných pro vyústky vzduchotechnického vedení. Rozestup jednotlivých děr ležících v přímce je 24 cm a na jeden plech je jich potřeba vyvrtat vždy 10. Navrhněte program, který by samostatně prováděl požadovanou práci.

Informace o činnosti

Pro vizuální kontrolu činnosti robota při vrtání je potřeba zobrazovat informace o aktuálním průběhu vrtáku na displeji robota.

Nový výrobek

Firma rozšiřuje zakázkovou výrobu a zavádí výrobu čtvercového plechu s vrtanými kruhovými otvory po jeho obvodu. Každá strana čtverce o velikosti 72 cm obsahuje 4 otvory v pravidelných rozestupech. Rohové otvory jsou pro jednotlivé strany společné. Je tedy potřeba vyvrtat celkem 12 otvorů. Vrtání bude začínat vždy v jednom rohu čtvercového plechu a pokračovat bude ve směru hodinových ručiček. Na každé straně tak bude robot vrtat 3 otvory, které označíme souřadnicemi o dvou znacích XY. První číslice značí danou stranu čtverce a druhá číslice pořadí otvorů.

Zvuk vrtání (rozšiřující)

Během vrtání by přeci mělo být něco slyšet, přidejte zvuk vrtačky. Požadovaný zvuk je k dispozici v zadání úlohy na přiloženém DVD.

Předpokládaná časová náročnost

- konstrukce (základní robot + vrták): 10 min;
- Nový vrták: 15 min;
- Informace o činnosti: 5 min;
- Nový výrobek: 25 min;
- Zvuk vrtání (rozšiřující): 5 min;
- **celkem: 60 min** (bez rozšíření: 55 minut).

Pokyny pro učitele

Úloha je zaměřena na cyklus s pevným počtem průchodů. V úloze se pracuje i s proměnnými a s výpisem na displej.

Konstrukce vrtáku nedovolí ose vyjet příliš vysoko ani nízko. Proto pokud bude špatně nastavený výkon, tak se motor nebude moct točit. Díky dvojitému převodu jsou kladné výkony pro směr nahoru a záporné dolů. Vhodné hodnoty jsou např. 30% po dobu 500ms.

Pro správné řešení je potřeba, aby se program spustil v okamžiku, kdy je vrták v horní poloze. Konstrukce je k dispozici na přiloženém DVD.

Ve třetím úkolu mohou studenti přijít s řešením, ve kterém vytvoří čtyři samostatné cykly. V tom případě je nutné je upozornit na to, že se jim v programu vyskytují velké množství

stejného kódu, a proto je mnohem lepší vytvořit další cyklus s pevným počtem průchodů, který se bude 4x opakovat.

Pro rozšiřující úlohu byl připraven zvuk vrtačky, který je již ve správném formátu (.rso) k dispozici na DVD. Zvuk lze do kostky nahrát pomocí okna „File Manager“. V tomto okně stačí zvolit příkaz „Download“ a vybrat zvukový soubor a potvrdit.

5.5.7 ÚLOHA 7: AUTOMATICKÉ SVĚTLO

Úloha v první části kombinuje využívání světelného a ultrazvukového senzoru. Program rozsvěcuje žárovky podle okolní intenzity světla a reakcí na pohyb. Ve druhé části úlohy se využívá pouze ultrazvukový senzor. Světlo se s první detekcí pohybu rozsvítí a s druhou zhasne.

Cíle úlohy

Student:

- použije vhodné řídicí struktury pro stálou detekci a vyhodnocování hodnot ze senzorů;
- vysvětlí funkce logický součin a logický součet;
- zkontroluje optimální funkčnost programu.

Klíčové pojmy, příkazy v RobotC

- podmínka, vnořené podmínky;
- proměnná, datový typ boolean;
- operátory AND (logický součin) a OR (logický součet).

Zadání

Světlo na chodbě

Rozhodli jste si ušetřit námahu s rozsvěcováním světla na chodbě. Chcete, aby se světlo rozsvítilo, pokud někdo v noci půjde přes chodbu.

Vylepšené světlo na chodbě

Váš strýc má malé děti, které se bojí tmy. Vaše světlo se mu líbí, ale chtěl by, aby bylo na chodbě rozsvíceno i v případě, že nikdo nejde. V chodbě má světlo o třech žárovkách a jeho fungování by si představoval takto: při dostatku světla nesvítí žádná žárovka,

při úplné tmě svítí všechny tři žárovky a mezi tím se počet svítících žárovek mění společně s intenzitou světla. Co však určitě chce zachovat je to, aby se, pokud svítí alespoň jedna žárovka a venku tedy není zcela světlo, rozsvítily všechny tři žárovky, projde-li někdo chodbou.

Světlo ve spižírně

Kuchař ve vaší oblíbené restauraci neustále chodí do spižírny pro suroviny na vaření. Protože u toho má vždy plné ruce, byl by rád, kdyby nemusel pokaždé sám rozsvítit a zhasnout, ale kdyby se světlo rozsvítilo samo, jakmile vstoupí. Světlo musí zůstat rozsvícené po celou dobu, po kterou vybírá vhodné potraviny a pak se samo zhasnout ve chvíli, kdy místnost opustí.

Zlepšená detekce (rozšiřující)

Všechno, co jste si předsevzali zařídit, již funguje. Po nějaké době se vám ale ozval kuchař, že „něco nehraje“. Říkal, že když se zastaví ve dveřích, tak světlo bliká a navíc se mu už párkrát stalo, že zůstal ve spižírně po tmě. Zkuste nedostatky opravit.

Předpokládaná časová náročnost

- konstrukce (základní robot + žárovky, světelný a ultrazvukový senzor): 5 min;
- Světlo na chodbě: 10 min;
- Vylepšené světlo na chodbě: 20 min;
- Světlo spižírně: 15 min;
- Zlepšená detekce (rozšiřující): 15 min;
- **celkem: 65 min** (bez rozšíření: 50 minut).

Pokyny pro učitele

Je dobré zmínit potřebu zapsat ověřování vzdálenosti pomocí úplné podmínky. Tím se zabezpečí to, že světlo nebude problikávat, ale bude svítit po požadovanou dobu. V prvním úkolu studenti budou muset použít logický operátor AND a ve druhém úkolu logický operátor OR. Z praktického využití by tedy měli být schopni tyto operátory popsat.

Ve třetím úkolu existuje více řešení. Pokud studenti budou úkol řešit jen pomocí podmínek, tak program nebude fungovat zcela správně. Mohou se však vyskytnout studenti, kteří úkol již vyřeší pomocí cyklu while. Na ně by se poté nevztahoval rozšiřující úkol, místo toho

by mohli dělat poradce ostatním skupinám, a vysvětlit, z jakého důvodu úkol vyřešili pomocí cyklu a ne podmínky.

5.5.8 ÚLOHA 8: ROBOTICKÝ SMETÁČEK

Studenti naprogramují robota, který bude jezdit ve vymezeném prostoru a ultrazvukovým senzorem bude detekovat zeď. Protože detekce nemusí fungovat ve všech případech optimálně, studenti přidají na robota nárazník.

Cíle úlohy

Student:

- navrhne jízdu robota tak, aby se pohyboval po celé vymezené ploše;
- použije správný zápis podmínky pro ošetření situace, pokud by se robot ocitl ve slepém místě;
- popíše způsob zadávání náhodného čísla;
- navrhne algoritmus, který bude detekovat náraz do levého či pravého nárazníku.

Klíčové pojmy, příkazy v RobotC

- náhodné číslo – `random()`;
- úplná a neúplná podmínka;
- nekonečný cyklus.

Zadání

Úklid

Ach to uklízení..., je nutné, ale málokoho baví. Řekli jste si tedy, že si práci ulehčíte a pořídíte si RoboSmetáček od firmy Smeták a syn. Bohužel je třeba RoboSmetáček naprogramovat a s tím si nyní musíte poradit. Smetáček bude uklízet v ohraničené místnosti bez překážek. Měl by jezdit po ploše a uklízet ji tak, aby neboural do zdí.

Robot a úskalí

Jste připraveni na situaci, kdy RoboSmetáček zajede do slepé uličky či do rohu? Otestujte ho, a pokud ne, raději to napravte. Současně zvyšte jeho efektivitu tím, že nastavíte otáčení o náhodný úhel. Jeho pohyby po místnosti se tak nebudou opakovat a uklidí lépe.

„Ta technika“

Je váš RoboSmetáček také nepoučitelný a stále bourá? Smeták a syn vydali nový nárazník určený pro RoboSmetáček, který zajistí, že při nárazu robot okamžitě zastaví a couvne. Při pohledu na nárazník zjistíte, že ukrývá dokonce dva dotykové senzory. Díky tomu můžete RoboSmetáček naprogramovat tak, aby při nárazu na levou stranu couvnul a otočil se doprava a naopak.

Předpokládaná časová náročnost

- konstrukce (základní robot + ultrazvukový senzor): 5 min;
- Úklid: 20 min;
- Robot a úskalí: 10 min;
- „Ta technika“: 15 min;
- **celkem: 50 min.**

Pokyny pro učitele

Úlohu je vhodné studentům zadat bez dalších doplňujících informací. Sami by měli robota a jeho jízdu naprogramovat tak, aby se vyhnul zdím a dalším překážkám.

Vzhledem ke konstrukci a mírným nepřesnostem robota nemusíme trvat na tom, že robot musí uklízet přímo u stěn. Z důvodu dostatečného prostoru na otočení je možné nechat ultrazvukový senzor nastaven na větší vzdálenost.

V případě nárazníku stačí detekovat levou a pravou stranu. Zadání, aby program hlídal také to, jestli došlo ke stisku obou senzorů naráz, je trochu zbytečné, protože během jízdy se vždy jeden ze senzorů sepne o malou chvíli dřív. Pokud bychom chtěli detekovat oba dotykové senzory najednou, bylo by potřeba do programu přidat zpoždění, při kterém by program čekal, zda do určité doby nesezne také druhý senzor.

Prostor určený k „úklidu“ je potřeba obklopit ze všech stran zdí, která by měla vydržet náraz robota. Vyhneme se tak časovým prodlevám při opakovaném stavění zdí po nárazu.

Jako rozšíření úlohy je možné do vymezeného prostoru umístit různé překážky a sledovat, které robot dokáže detekovat a které mu naopak dělají obtíže.

5.5.9 ÚLOHA 9: SPRINTER „ROBOLT“

Studenti naprogramují robota tak, aby odstartoval ve chvíli, kdy zvukový senzor detekuje tlesknutí. Zároveň díky světelnému senzoru dokáže určit cíl a zastavit zde. Robot si také měří čas jízdy.

Cíle úlohy

Student:

- vybere vhodnou řídicí strukturu pro rozpoznání tlesknutí a následného rozjezdu;
- navrhne algoritmus, který bude detekovat dvě tlesknutí;
- používá příkazy pro ovládání časovače (nulování a uložení do proměnné).

Klíčové pojmy, příkazy v RobotC

- synchronizace motorů;
- časovač (timer).

Zadání

Blíží se každoroční soutěž ve sprintu pro roboty, do které se přihlásíte a postavíte pro tyto účely vlastního robota-sprinterera. Jako opravdoví závodníci i ti robotičtí startují na výstřel (v našem případě na tlesknutí). Před velkým závodem však musí tvrdě trénovat.

Trénink startů

Začneme s tréninkem pár cvičných startů. Robot tedy bude nehybně na startu čekat na tlesknutí, a poté se rozjede. Zatím se nemusíte zabývat jeho zabrzděním, je tedy vhodné pověřit jednoho člena týmu, který jej včas zastaví.

Trénink rychlosti reakcí

Další trénink bude na hbitost, robot opět na tlesknutí odstartuje, ale na další tlesknutí se zastaví. Je tedy jen na vás, jako trenérech, jak dlouho robot „poběží“.

Trénink start-cíl

Poslední trénink před velkým závodem již bude plnohodnotný sprint od startu až do cíle. Protnutí cílové pásky snímejte světelným senzorem. To znamená, že robot odstartuje a pojedete tak dlouho, dokud nenarazí na cílovou pásku v podobě černé čáry.

Závod

Nastal čas závodu a proto je potřeba začít měřit čas. Zadržte, nemusíte si připravovat stopky, robot má svoje vlastní. Přidejte do programu časovač pro měření doby jízdy robota. Výsledný čas měřte férově, od startu až do cíle. Pro lepší kontrolu, i možnost uznání, nechte čas zobrazovat na displeji. Teď už jen zbývá si tipnout, za jak dlouho zvládne robot ujet celou trať.

Výsledky (rozšiřující)

Není příliš pěkné, že se výsledky zobrazují pouze v milisekundách. Program upravte tak, aby se výsledný čas zobrazoval přehledně a číslo bylo zobrazeno v sekundách se dvěma desetinnými místy.

Překonání rekordu (konstrukční úloha)

Jaký byl nejlepší výsledek dnešní soutěže? Dokázal někdo porazit loňského vítěze a rekordmana „SpeedyBota“, který dokázal zajet trať o délce 1,6 metrů za čas 1,21 sekund? Pokoříte ho? Možná budete muset lehce upravit konstrukci vašeho robota.

Předpokládaná časová náročnost

- konstrukce (základní robot + zvukový a světelný senzor): 5 min;
- Trénink startů: 15 min;
- Trénink rychlosti reakcí: 15 min;
- Trénink start-cíl: 10 min;
- Závod: 15 min;
- Výsledky (rozšiřující): 10 min;
- Překonání rekordu (konstrukční úloha): 40 min;
- **celkem: 105 min** (bez rozšíření: 60 minut).

Pokyny pro učitele

Celkově se jedná o dlouhou úlohu, která může být rozdělena do dvou hodin: na „tréninkové“ úkoly a poté závod.

Je dobré upozornit na skutečnost, že zvukový i světelný senzor potřebuje pro správnou funkci úvodní časovou prodlevu. Pokud by nebyl přidán příkaz na čekání programu, tak se může stát, že robot odstartuje ihned po zapnutí programu.

Studenti si zajisté všimnou, že robot sám od sebe mírně zatáčí. V takovém případě je vhodné jim vysvětlit, že je to způsobeno odlišným třením jednotlivých kol, vzorkem pneumatik nebo i samotnými motory. Ty mohou mít drobné odchylky již z výroby a každý může požadovaného výkonu dosahovat v jiném čase. V rámci toho se nabízí představit studentům možnost, jak to alespoň částečně potlačit pomocí příkazu k synchronizaci motorů.

Při plnění druhého úkolu může dojít také k situaci, kdy robot na tlesknutí vůbec nevyjede nebo pouze poskočí. Způsobeno je to chybějícím zpožděním za prvním cyklem, když zvukový senzor chybně vyhodnotí jedno tlesknutí jako dvě a robota rovnou vypne.

Ve čtvrtém úkolu je nutné nastavit, aby docházelo k nulování časovače a přiřazovat hodnotu do proměnné ve správný okamžik. Zobrazení desetinných míst lze vyřešit datovým typem float. Při prvním výpisu bude číslo obsahovat zbytečně velký počet desetinných míst. Jelikož se jedná jen o úkol pro rychlejší studenty, je možné je odkázat na nápovědu programu. Zde si studenti mohou najít informaci, jak docílit toho, aby byla zapsána jen dvě desetinná místa.

Poslední úkol nabízí možnost nechat studenty zkusit přijít se svojí konstrukční úpravou robota. Díky tomu je úloha časově náročná a navíc neobsahuje z programového hlediska nový obsah oproti předešlým úkolům. Proto zařazení úlohy je třeba zvážit dle konkrétních podmínek výuky. Na přiloženém DVD je umístěna možná konstrukce (SpeedyBot), kterou lze dát studentům k dispozici, a tím výrazně zkrátit vyhrazený čas pro úkol.

Závěr úlohy je možné doplnit výpočtem průměrné rychlosti jízdy. A například porovnáním s opravdovým Usainem Boltem.

5.5.10 ÚLOHA 10: JÍZDA PO ČÁŘE

Studenti naprogramují robota tak, aby pomocí světelného senzoru sledoval vodící čáru. Díky dotykovému senzoru zajistí, aby robot zastavil, jakmile narazí do překážky.

Cíle úlohy

Student:

- navrhne algoritmus pro jízdu robota po vodící čáře;
- použije vhodný způsob jak detekovat stlačení dotykového senzoru a následného zastavení robota;
- využije senzor otáček pro přerušení sledování čáry.

Klíčové pojmy, příkazy v RobotC

- podmínka;
- nekonečný cyklus;
- logické operátory;
- senzor otáček (encoder).

Zadání

Jízda po čáře

Vedoucí výrobní haly chce zefektivnit náklady a rozhodl se, že jim materiál budou rozvážet automatické vozíky. Vedení společnosti souhlasí, musí jim ale nejprve dokázat funkčnost nápadu v testovacím prostředí. Využijí k tomu zatím nevyužitou halu, kde vytvoří imaginární trasu, kterou by měl robot za pomoci světelného senzoru celou projet. Navrhněte a naprogramujte algoritmus, který robotovi umožní sledovat vodící čáru a projet celou trasu.

Detekce konce

Robot zdárně zdolal celou trať. Na konci se však začal zmateně točit, neboť vodící čára skončila a on se ji přesto pokoušel najít. Přidejte proto na robota nárazník ve formě dotykového senzoru, který při kontaktu s překážkou robota zastaví. Vytvoří se tím nejen možnost ukončení programu po projetí zadané trasy, ale také ochrana robota po celou dobu jízdy.

Délka trasy

Změřte, jak dlouhou trasu robot ujede. Protože vodící čára je celkem klikatá, využijte senzor otáček, který je umístěn v každém motoru. Hodnoty nezapomeňte zobrazit na displeji.

Zkratka

Pro některé situace by bylo šikovné, kdyby robot uměl přerušit sledování vyznačené trasy, vykonal požadovaný pokyn a následně opět našel a sledoval vodící čáru. Vytvořte tedy program, díky kterému robot ve vybraném místě opustí svojí čáru, přejede na jinou a vydá se opačným směrem.

Rychlejší průjezd (rozšiřující)

Je potřeba, aby byl robot rychlejší. Zkuste přijít s řešením, které robota zrychlí a stále bude schopný zdat celou trať. Znamená to, že je nutné najít způsob, jak naučit robota, aby na rovných úsecích jel rychleji a při průjezdu prudkých zatáček zpomalil.

Předpokládaná časová náročnost

- konstrukce (základní robot + světelný a dotykový senzor): 5 min;
- Jízda po čáře: 20 min;
- Detekce konce: 10 min;
- Délka trasy: 15 min;
- Zkratka: 10 min;
- Rychlejší průjezd (rozšiřující): 15 min;
- **celkem: 75 min** (bez rozšíření: 60 minut).

Pokyny pro učitele

V prvním úkolu je třeba studentům sdělit, že princip spočívá ve sledování přechodu mezi bílou a černou barvou, nikoliv v samotném sledování černé linky. Studenti by poté měli sami přijít na tom, že záleží, jakou stranu čáry si zvolí a podle toho musí přizpůsobit program. Studentům není třeba zadávat hodnoty pro výkon motorů, aby mohli sami otestovat, jaký výkon je optimální.

Studenti by již měli vědět, jakým stylem upravit kód, aby robot po celou dobu jízdy detekoval případné stisknutí tlačítka.

Třetí a čtvrtý úkol studentům odkrývá další možnost, jak řídit pohyb robota, a to pomocí senzoru otáček umístěného přímo v motoru. Studenti se nejprve seznámí pomocí měření trasy a přepočtu výsledné hodnoty otáček na cm. Výpočty možná některým studentům

budou dělat obtíže, a tak na nich nemusíme trvat a je možné poskytnou potřebné výsledky ($1^\circ = 0,05$ cm, neboli v programu musíme naměřenou hodnotu senzorem vynásobit 0,05).

V pátém úkolu se detekuje více stavů intenzity světla, musí být tedy více podmínek. Je možné vytvořit více neúplných podmínek, nebo využívat vnoření podmínek. Ukázkové řešení detekuje pět stavů a používá pět neúplných podmínek.

5.5.11 ÚLOHA 11: TRASA JÍZDY

Robot jezdí na vyznačené trase. Pro zadání jízdy jsou využity vlastní metody. V programu tak stačí zadávat rychlost a vzdálenost (případně úhel otočení).

Cíle úlohy

Student:

- vytváří vlastní metody bez návratové hodnoty;
- dokáže volat vlastní metody s parametry;
- popíše výhody používání vlastních metod.

Klíčové pojmy, příkazy v RobotC

- vlastní metody bez návratové hodnoty – void;
- parametry vlastní metod.

Zadání

Trasa

Připravte robota, který bude schopen projet znázorněnou trasu (Obrázek 26). Vytvořte program, ve kterém nebudete muset trasu předem naprogramovat, ale dokážete ji projet díky snadným změnám výkonu a času jízdy.

Využijte vlastní metody s parametry. Parametry vlastních metod upravte tak, aby se mohla zadávat přímo rychlost a vzdálenost u pohybu vpřed a rychlost a úhel zatočení u otáčení. Otáčení řešte jen kolem osy robota. Pro přesnější pohyby využijte pro jízdu synchronizaci motorů, a abyste jeli na přesnou vzdálenost, tak použijte rotační senzor (encoder).

Rozšiřující úkol slouží především jako ukázka dalších možností RobotC. Pro tento úkol je vyžadováno využití Bluetooth rozhraní, protože ovládání jedoucího robota tlačítky umístěnými přímo na kostce, je velmi nepraktické.

5.5.12 ÚLOHA 12: TŘÍDIČKA MÍČKŮ

Robot rozpozná barvu míčku (červená nebo modrá) a zaveze ho na určené místo, kde ho vyloží. Poté se vrátí zpět na startovní pozici a čeká na další míček.

Cíle úlohy

Student:

- rozčlení úlohu na menší části;
- navrhne algoritmus pro konkrétní část programu (např. spuštění a zvednutí ramena, ověření barvy míčku, pohyb robota dle barvy míčku);
- popíše fungování programu.

Klíčové pojmy, příkazy v RobotC

- vlastní metody;
- pohyb robota pomocí senzoru otáček.

Zadání

V továrně na míčky je třeba rozdělovat výrobky dle barev. Továrna vyrábí míčky ve dvou barvách: modré a červené. Každá barva má své místo a je třeba je roztřídit a odvézt. To přesně bude dělat váš robot. Robot musí naložit míček a podle barvy ho odvézt na určenou pozici. Na pozici míček vyloží a vrátí se zpět, kde bude připraven akci opakovat.

Předpokládaná časová náročnost

- konstrukce (třidička míčků): 10 min;
- celá úloha: 70 min;
- **celkem: 80 min.**

Pokyny pro učitele

Rozsáhlá a obtížná úloha, která se hodí na závěr úvodu do programování. Je nutné zvážit podle dovedností studentů, jestli ji zadat všem či jen zájemcům. Úlohu lze řešit i společně

se studenty. Učitel tedy bude mít funkci koordinátora myšlenek a studenti pomalými krůčky budou řešit úkol. Řešení úlohy nespočívá v programování složitých příkazů, ale naopak obsahuje převážně jednoduché základní pokyny, které jsou však nakumulovány ve větším množství. Od učitele je tedy vhodné, aby vedl studenty po jednotlivých krocích tak, aby se postupně dopracovali k celkovému řešení.

Prvním krokem je vytvořit část kódu, která bude zajišťovat ovládání ramena pro úvodní část úlohy. V řešeném příkladu se zvednutí ramene provádí pomocí ultrazvukového senzoru. Je však možné řešení upravit a rameno ovládat pomocí dotykového senzoru. Součástí úvodní části je i rozpoznání barvy míčku. Jakmile robot dokáže naložit míček a rozpoznat barvu, je možné postoupit a začít programovat pohyb robota.

Pro řešení úlohy je připravena konstrukce robota na DVD

ZÁVĚR

V diplomové práci byla představena robotická stavebnice LEGO Mindstorms NXT. Byly popsány aktivní díly stavebnice (řídící jednotka, vstupní a výstupní moduly), které obsahuje verze určená pro použití ve školách.

Dále byly představeny možnosti využití robotické stavebnice na středních školách. Nejširší využití má stavebnice v rámci programování a robotiky (automatizace). Přesto byly nastíněny také další možnosti využití v jiných oborech.

Dále bylo popsáno programovací prostředí RobotC. Popis byl zaměřen na možnosti a funkce programu. Byl také stručně představen programovací jazyk RobotC.

V diplomové práci bylo prezentováno postavení výuky programování v RVP pro střední školy. Byl vybrán vzorek 52 RVP tak, aby obsahoval různé typy středních škol v České republice. Na základě toho bylo zjištěno, že výuka programování se vyskytuje jen v RVP pro některé učební obory, jakými jsou například informační technologie či technická lycea. V ostatních RVP byla výuka programování nejčastěji zastoupena algoritmizací. Pro účely práce byly prostudovány také slovenské vzdělávací programy a následně provedeno jejich porovnání s českými. Bylo zjištěno, že programování a algoritmizací není v RVP věnován přílišný prostor.

Na základě poznatků získaných z prostudovaných RVP byly stanoveny cíle pro sadu úloh. Konkrétní úlohy byly vytvořeny a seřazeny podle obtížnosti a množství potřebné látky pro jejich splnění. Pro lepší přehled o struktuře učiva, které jednotlivé úkoly přináší, byla vytvořena osnova sady. Představena byla také konstrukce robota a publikovány nezbytné podmínky pro plnění zadaných úloh

Výstupem diplomové práce je sada 12 úloh určených pro podporu výuky programování na středních školách. Úlohy jsou složeny z dílčích podúkolů a doplněny o zadání a metodické pokyny. Pro všechny úlohy byla vytvořena řešení, která jsou dostupná na přiloženém DVD. Na tomto nosiči jsou zároveň k dispozici zadání pro studenty a metodické pokyny pro učitele.

Poznatky získané při tvorbě diplomové práce mi budou cenným zdrojem informací v budoucnosti při tvorbě přípravy na vyučovací hodiny, neboť zapojení moderních pomůcek do výuky je dle mého názoru vhodnou cestou k aktivizaci studentů při hodinách.

RESUMÉ

In the thesis the robotic construction set LEGO Mindstorms NXT was presented. The active components (regulation unit, input and output modules) contained in a suite prepared for schools were described.

Next were presented methods of use of the construction set at secondary schools. The most useful use of the construction set is for schooling of programming and robotics (automation). Although some additional uses for different branches of study were also presented.

After that the RobotC environment was described. The thesis is focused on options and functions of the program together with a brief introduction to the RobotC programming language.

In the thesis was presented function of schooling of programming in Framework Education Programme (FEP) for secondary schools. 52 FEP samples were chosen so they contain various types of secondary schools across the Czech Republic. From the comparison of the collected data was found out that schooling of programming is in FEP in several branches of study, for example information technologies and technical lyceums. In the other FEPs the schooling of programming was often substituted by algorithm development. For the purpose of the thesis the Slovak analogue of FEPs were also studied and subsequently compared with the Czech ones. The results of the comparison show that not much scope in FEP is given to the programming and algorithm development.

Based on the findings from studied FEP were defined goals for set of assignments. Specific assignments were created and sorted by difficulty and volume of subject matter needed for their completion. Additionally was created schema for better comprehension of the structure of the assignments. Essential conditions for the completion of the assignments were also published together with the construction of the robot.

The output of the thesis is the set of 12 assignments intended for schooling of programming at secondary schools. The assignments are composed of sub-assignments with general instructions. For all assignments were created answers that can be found at the attached

DVD. On the DVD can be also found instructions for the students and methodical instructions for teachers.

SEZNAM LITERATURY

1. LEGO GROUP. Historie LEGO Robotics - robotického programu. *LEGO.com Mindstorms* [online]. 2014 [cit. 2014-06-07]. Dostupné z: <http://www.lego.com/cs-cz/mindstorms/gettingstarted/historypage>
2. BUMGARDNER, J. The Origins of Mindstorms. *Wired.com: GeekDad* [online]. 29032007 [cit. 2014-06-07]. Dostupné z: http://archive.wired.com/geekdad/2007/03/the_origins_of_/
3. Mindstorms RCX. *Brickipedia: the LEGO Wiki* [online]. [cit. 2014-06-07]. Dostupné z: http://lego.wikia.com/wiki/Mindstorms_RCX
4. LAPLANT, W. Lego Robots Project - History. *Lego Robotics in Education* [online]. 2006 [cit. 2014-06-07]. Dostupné z: <http://whipplefamily.com/bridgew/portfolio/INST525/history.htm>
5. Mindstorms NXT. *Brickipedia: the LEGO Wiki* [online]. [cit. 2014-06-07]. Dostupné z: http://lego.wikia.com/wiki/Mindstorms_NXT
6. LEGO GROUP. LEGO MINDSTORMS EV3: Education Core Set For Primary and Secondary Schools. *LEGO Education* [online]. 2013 [cit. 2014-06-07]. Dostupné z: <http://shop.legoeducation.com/gb/product/lego-mindstorms-education-ev3-core-set-45544-198/>
7. ROBOMATTER, INC. *ROBOTC.net: Home of the best robot programming language for Educational Robotics* [online]. 2005-2014 [cit. 2014-06-08]. Dostupné z: <http://www.robotc.net>
8. BAICHTAL, J. a J. MENO. *The cult of LEGO*. San Francisco: No Starch Press, 2011. ISBN 978-1-59327-391-0.
9. GRIFFIN, T. *The art of LEGO Mindstorms NXT-G programming*. San Francisco: No Starch Press, 2010. ISBN 978-1-59327-218-0.
10. EDUXE: distributor učebních pomůcek LEGO Education. *9797 MINDSTORMS Education NXT Základní souprava* [online]. [cit. 2014-06-07]. Dostupné z: <http://www.eduxe.cz/product/9797-mindstorms-education-nxt-zakladni-souprava-332/>
11. LAURENS. Robot Square - Tutorial: Understanding the difference between NXT set versions. *Robot Square - (Mindstorms) Robot design and development* [online]. 2012 [cit. 2014-06-07]. Dostupné z: <http://robotsquare.com/2012/02/18/understanding-nxt-versions/>
12. BLACKMORE, C. NXT 1.0, 2.0, and Education Inventory Comparison. *SABRE: The Southern Alberta Robotics Enthusiasts* [online]. [cit. 2014-06-07]. Dostupné z: http://static.robotclub.ab.ca/pages/nxt/InventoryComparison/nxt_1_vs_2_vs_edu.html#Electronics
13. EDUXE: distributor učebních pomůcek LEGO Education. *9695 Souprava doplňkových dílů* [online]. [cit. 2014-06-07]. Dostupné z: <http://www.eduxe.cz/product/9695-souprava-doplňkovych-dilu-333/>

14. LEGO GROUP. MINDSTORMS Robot, Robot Toy, Robotic Toys. *LEGO Shop* [online]. 2014 [cit. 2014-06-11]. Dostupné z: <http://shop.lego.com/en-US/MINDSTORMS-ByTheme>
15. LEGO GROUP. *LEGO MINDSTORMS Education NXT: NXT 2.0 User Guide*. 2008 [cit. 2014-06-07]. Dostupné z: http://cache.lego.com/downloads/education/9797_LME_UserGuide_US_low.pdf
16. PERDUE, D. J. a L. VALK. *The unofficial LEGO Mindstorms NXT 2.0 inventor's Guide*. 2nd Edition. San Francisco: No Starch Press, 2010. ISBN 9781593272159.
17. Levels of Noise. *American Academy of Audiology* [online]. 2014 [cit. 2014-06-11]. Dostupné z: <http://www.audiology.org/practice/resources/PublishingImages/NoiseChart24x36.pdf>
18. TROBAUGH, J. J. a M. LOWE. *Winning Lego Mindstorms Programming*. New York: Apress, 2012. ISBN 978-1-4302-4537-7.
19. EDUXE: distributor učebních pomůcek LEGO Education. *Robotika LEGO MINDSTORMS Education NXT* [online]. [cit. 2014-06-07]. Dostupné z: <http://www.eduxe.cz/les/mindstorms/>
20. LEGO GROUP. LEGO MINDSTORMS Education: Partners. *LEGO.com Education* [online]. 2014 [cit. 2014-06-07]. Dostupné z: <https://education.lego.com/en-us/preschool-and-school/secondary/11plus-mindstorms-education/partners>
21. CHURCH, W. et al. Physics With Robotics Using LEGO® MINDSTORMS® in High School Education. In: *2010 AAAI Spring Symposium Series*. Palo Alto: Association for the Advancement of Artificial Intelligence, 2010, s. 47 - 49 [cit. 2014-06-12]. Dostupné z: <http://www.aaai.org/ocs/index.php/SSS/SSS10/paper/viewFile/1062/1398>
22. KALHOUS, Z. a O. OBST. *Školní didaktika*. Praha: Portál, 2002. 80-7178-253-X.
23. BUTTERWORTH, D. T. Teaching C/C++ Programming with Lego Mindstorms. In: *RIE. 3rd International Conference on Robotics in Education*. Prague: RiE, 2012, s. 61-65 [cit. 2014-06-08]. Dostupné z: <http://www.ksi.mff.cuni.cz/rie2012/proceedings/2012RiE-08.pdf>
24. KIM, S. H. a J. W. JEON. Educating C Language Using LEGO Mindstorms Robotic Invention System 2.0. In: *IEEE International Conference on Robotics and Automation*. Orlando: IEEE, 2006, s. 715 - 720. ISSN 1050-4729.
25. SULÍR, M. *Úvod do programování robotů 1*. Orlová: Obchodní akademie Orlová, 2012 [cit. 2014-06-12]. 978-80-87477-04-5. Dostupné z: <http://scholanova.obaka-orlova.cz/kurz.html>
26. TALÁČEK, M. Praktická cvičení z informatiky. *Gymnázium Jihlava - VY_32_INOVACE_7-3* [online]. [cit. 2014-06-14]. Dostupné z: <http://www.gymnaziumjihlava.cz/projekty/1953.html?task=view>
27. Gymnázium Sušice. *Robotika* [online]. [cit. 2014-06-15]. Dostupné z: <http://www.gymsusice.cz/predmety/robotika.html>

28. FLL: *FIRST LEGO League* [online]. [cit. 2014-06-15]. Dostupné z: <http://www.firstlegoleague.org>
29. *Česká liga robotiky* [online]. 2013 [cit. 2014-06-15]. Dostupné z: <http://www.ceskaligarobotiky.cz>
30. ROBOTI. *ČVUT - Katedra řídicí techniky* [online]. 2010 [cit. 2014-06-12]. Dostupné z: <http://www.robosoutez.cz/>
31. ŠANDOVÁ, H. Univerzitní finálová Robosoutěž FEL ČVUT. *INFORMATIKA* [online]. 2011 [cit. 2014-06-12]. Dostupné z: http://195.113.84.161/~sandova/201314_r5a/robosoutez_VS.php
32. SPŠ, SOŠ a SOU Nové Město nad Metují. *Robotika - Automatizace* [online]. 2014 [cit. 2014-06-13]. Dostupné z: <http://robotika.spsnome.cz/automatizace>
33. ŽIDLÍK, M. *Využití stavebnice Lego Mindstorms NXT při výuce bezpečnostních technologií*. Zlín: 2010. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky [cit. 2014-02-25]. Dostupné z: <http://hdl.handle.net/10563/12031>
34. WILLIAMS, K. et al. Enriching K-12 Science and Mathematics Education Using LEGOs. In: *Advances in Engineering Education*. ASEE: American Society for Engineering Education, 2012, s. 1-27 [cit. 2014-06-10]. Dostupné z: http://advances.asee.org/wp-content/uploads/vol03/issue02/papers/aee-vol03-issue02-p10_correct.pdf
35. ALBRECHT, M. The use of the LEGO MINDSTORMS® System in Modeling the Foraging Behavior and Strategies of Simple In: *Tested studies for laboratory*. Association for Biology Laboratory Education (ABLE), 2002, s. 145-54 [cit. 2014-06-12]. Dostupné z: <http://www.ableweb.org/volumes/vol-23/9-albrecht.pdf>
36. ROBOMATTER, INC. *ROBOMATTER: We simply teach the complex* [online]. 2014 [cit. 2014-06-08]. Dostupné z: <http://robomatter.com>
37. CARNEGIE MELLON UNIVERSITY. Robotics Academy. *The Robotics Academy: Your source for Robotics Education materials and training* [online]. 2014 [cit. 2014-06-08]. Dostupné z: <http://www.education.rec.ri.cmu.edu/content/lego/start/>
38. *ROBOTC API Guide* [online]. verze 30 January 2014 [cit. 2014-06-08]. Dostupné z: http://www.robotc.net/wiki/Main_Page
39. CARNEGIE MELLON UNIVERSITY. ROBOTC Curriculum for LEGO MINDSTORMS. *The Robotics Academy: Your source for Robotics Education materials and training* [online]. 2009 [cit. 2014-06-08]. Dostupné z: http://www.education.rec.ri.cmu.edu/previews/robot_c_products/teaching_rc_lego_v2_preview/
40. *ROBOTC.net forums - Index page* [online]. [cit. 2014-06-08]. Dostupné z: <http://www.robotc.net/forums/>
41. Národní ústav pro vzdělávání. *RVP pro střední odborné vzdělávání* [online]. 2011-2014 [cit. 2014-06-17]. Dostupné z: <http://www.nuv.cz/ramcove-vzdelavaci-programy/rvp-os>

42. *Rámcový vzdělávací program pro gymnázia*. Praha: Výzkumný ústav pedagogický v Praze, 2007 [cit. 2014-06-02]. ISBN 978-80-87000-11-3. Dostupné z: <http://www.nuv.cz/file/159>
43. Gymnázium, Brno, třída Kapitána Jaroše 14. *Nabídka volitelných předmětů* [online]. 2014 [cit. 2014-06-19]. Dostupné z: <http://www.jaroska.cz/node/21>
44. Gymnázium Sokolov. *Volitelné předměty* [online]. 2012 [cit. 2014-06-19]. Dostupné z: <http://www.gymso.cz/volitelne-predmety.php>
45. Gymnázium Jaroslava Vrchlického Klatovy. *Informatika a programování* [online]. [cit. 2014-06-19]. Dostupné z: <http://www.klatovynet.cz/gymkt/informatika.asp>
46. SPŠ, SOŠ a SOU Nové Město nad Metují. *Učební plán oboru Strojírenství* [online]. 2011 [cit. 2014-06-20]. Dostupné z: <http://spsnome.cz/obory/studijni-obory/strojirenstvi>
47. *Učební plán: Strojírenství*. Plzeň: Střední průmyslová škola strojnická a Střední odborná škola profesora Švejcara, Plzeň, 2013 [cit. 2014-06-20]. Dostupné z: <http://www.spstrplz.cz/admin/files/File/Obory/up-strojirenstvi.pdf>
48. Obchodní akademie Plzeň. *Volitelné předměty* [online]. [cit. 2014-06-20]. Dostupné z: <http://www.oa.pilsedu.cz/volitelne-predmety>
49. *Rámcový vzdělávací program pro obor vzdělání 18-20-M/01 Informační technologie*. Praha: Ministerstvo školství, mládeže a tělovýchovy, 2008 [cit. 2014-06-05]. Dostupné z: <http://zpd.nuov.cz/RVP/ML/RVP%201820M01%20Informacni%20technologie.pdf>
50. *Školní vzdělávací program pro obor Informační technologie se zaměřením: Programování, vývoj* Dobruška: Střední škola – Podorlické vzdělávací centrum, 2012 [cit. 2014-06-20]. Dostupné z: http://www.sspvc.cz/Data/_web/ItRobotika.pdf
51. *Školní vzdělávací program pro obor Informační technologie*. Ostrava: Střední průmyslová škola elektrotechniky a informatiky, Ostrava, 2012 [cit. 2014-06-20]. Dostupné z: http://www.spseiostrava.cz/obory_2009/svp/svp_informacni_technologie_2012.pdf
52. *Školní vzdělávací program pro obor Informační technologie*. Jihlava: Střední průmyslová škola Jihlava, 2009 [cit. 2014-06-20]. Dostupné z: <http://www.sps-jia.cz/blue/PDF/ITsvp.pdf>
53. *Rámcový vzdělávací program pro obor vzdělání 78-42-M/01 Technické lyceum*. Praha: Ministerstvo školství, mládeže a tělovýchovy, 2007 [cit. 2014-06-06]. Dostupné z: <http://zpd.nuov.cz/RVP/ML/RVP%207842M01%20Technicke%20lyceum.pdf>
54. VANÍČEK, J. Programování. In: *Přednášky z didaktiky výpočetní techniky*. Jihočeská univerzita v Českých Budějovicích, Pedagogická fakulta, 2004 [cit. 2014-06-13]. Dostupné z: http://eamos.pf.jcu.cz/amos/kat_inf/externi/kat_inf_0548/7_programovani.pdf

55. *Rámcový vzdělávací program pro základní vzdělávání*. Praha: MŠMT ČR, 2013 [cit. 2014-06-15]. Dostupné z: <http://www.nuv.cz/ramcove-vzdelavaci-programy/upraveny-ramcovy-vzdelavaci-program-pro-zakladni-vzdelavani>
56. *Štátny vzdelávací program, Informatická výchova, Príloha ISCED 1*. Bratislava: Štátny pedagogický ústav, 2008 [cit. 2014-06-15]. Dostupné z: http://www.statpedu.sk/files/documents/svp/1stzs/isced1/vzdelavacie_oblasti/informaticka_vychova_isced1.pdf
57. *Štátny vzdelávací program, Informatika, Príloha ISCED 2*. Bratislava: Štátny pedagogický ústav, 2008 [cit. 2014-06-15]. Dostupné z: http://www.statpedu.sk/files/documents/svp/2stzs/isced2/vzdelavacie_oblasti/informatika_isced2.pdf
58. *Štátny vzdelávací program, Informatika, Príloha ISCED 3*. Bratislava: Štátny pedagogický ústav, 2008 [cit. 2014-06-15]. Dostupné z: http://www.statpedu.sk/files/documents/svp/gymnazia/vzdelavacie_oblasti/informatika_isced3a.pdf
59. PETTY, G. *Moderní vyučování*. Praha: Portál, 2008. ISBN 978-80-7367-427-4.
60. NXT Programs - Fun Projects for your LEGO Mindstorms NXT. *NXT Express-Bot* [online]. 2007-2011 [cit. 2014-03-02]. Dostupné z: <http://nxtprograms.com/9797/express-bot/index.html>

SEZNAM OBRÁZKŮ, TABULEK, GRAFŮ A DIAGRAMŮ

Obrázek 1: Vývoj robotických stavebnic LEGO Mindstorms [1].....	4
Obrázek 2: LEGO Mindstorms Education NXT Base Set 9797 [11].....	5
Obrázek 3: Ukázka dílů LEGO Technic.....	6
Obrázek 4: Řídící jednotka NXT [14]	7
Obrázek 5: Popis řídicí jednotky NXT	9
Obrázek 6: Interaktivní servomotor [14]	9
Obrázek 7: Dotykový senzor [14]	10
Obrázek 8: Zvukový senzor [14]	11
Obrázek 9: Světelný senzor [14]	11
Obrázek 10: Umístění světelného senzoru [18].....	12
Obrázek 11: Ultrazvukový senzor [14]	13
Obrázek 12: Pyramida učení dle S. Shapiro [22]	14
Obrázek 13: Uživatelské prostředí RobotC	20
Obrázek 14: Okno pro přehrání firmwaru	20
Obrázek 15: Hlavní nabídka a panel nástrojů.....	21
Obrázek 16: Okno pro nastavení senzorů.....	21
Obrázek 17: Deklarace senzorů v kódu a hlavní metoda „task main“	22
Obrázek 18: Obsah knihovny funkcí dle nastavení uživatelské úrovně (Zleva: Basic, Expert, Super User)	22
Obrázek 19: Informační okno.....	23
Obrázek 20: Okno pro základní ladění programu	24
Obrázek 21: Náhled jednotlivých ladících oken – a) proměnné, b) časovače, c) motory a senzory, d) virtuální kostka.....	24
Obrázek 22: Ukázka rozdílné přehlednosti kódu	25
Obrázek 23: Rozložení a složení základního robota	36
Obrázek 24: Schéma pracovní plochy	37
Obrázek 25: Struktura sady úloh	38
Obrázek 26: Trasa jízdy.....	61
Tabulka 1: Výchozí zapojení výstupů a vstupů [12]	8
Tabulka 2: Ukázka vybraných příkazů pro ovládání robota.....	25
Tabulka 3: Řídící struktury.....	26
Tabulka 4: Základní datové typy	26
Tabulka 5: Vlastní metody	27

PŘÍLOHY

Příloha 1: Přehled použitých RVP

	Kód	Obor vzdělání		Kód	Obor vzdělání
RVP pro gymnázia			27	4141M01	Agropodnikání
1	RVP G	Gymnázia	28	4157H01	Zpracovatel dřeva
2	RVP GSP	Gymnázia se sportovní přípravou	29	4341M01	Veterinářství
			30	5341M01	Zdravotnický asistent
RVP SOV, bez zaměření na informační technologie (s algoritmizací)			31	5343M01	Laboratorní asistent
			32	6341M01	Ekonomika a podnikání
3	1601M01	Ekologie a životní prostředí	33	6341M02	Obchodní akademie
			34	6542M02	Cestovní ruch
4	1602M01	Průmyslová ekologie	35	6843M01	Veřejnosprávní činnost
5	2142M01	Geotechnika	36	7241M01	Informační služby
6	2152H01	Hutník	37	7842M04	Zdravotnické lyceum
7	2341M01	Strojírenství	38	8241M04	Průmyslový design
			39	8251H09	Umělecký rytec
8	2344L01	Mechanik strojů a zařízení	40	8244M01	Hudba
9	2345M01	Dopravní prostředky	41	8247M01	Hudebně dramatické umění
10	2356H01	Obráběč kovů			
11	2362L01	Optik	RVP SOV, bez zaměření na informační technologie (bez algoritmizace)		
12	2641L01	Mechanik elektrotechnik	42	2857E01	Keramická výroba
13	2641M01	Elektrotechnika	43	2951E02	Potravinářské práce
14	2651H01	Elektrikář	44	3157E01	Textilní a oděvní výroba
15	2659H01	Spojový mechanik	45	3457E01	Knihářské práce
16	2842L01	Chemik operátor	46	4152E01	Zahradnické práce
17	3141M01	Textilnictví	47	6351J01	Obchodní škola
			48	6551E01	Stravovací a ubytovací služby
18	3343M01	Výroba hudebních nástrojů	49	8244J01	Ladení klavíru a kulturní činnost
19	3441M01	Polygrafie			
20	3453H01	Reprodukční grafik	50	3667E02	Stavební práce
21	3646M01	Geodzie a katastr nemovitosti	RVP SOV, se zaměřením na informační technologie		
22	3647M01	Stavebnictví	51	1820M01	Informační technologie
23	3741M01	Provoz a ekonomika dopravy	52	7842M01	Technické lyceum
24	3752H01	Železničář			
25	3908M01	Požární ochrana			
26	3941L02	Mechanik instalatérských a elektrotechnických zařízení			

Seznam příloh na DVD

- text diplomové práce v elektronické podobě
- využití dokumenty RVP pro střední vzdělávání
- zdrojové kódy k vyřešeným úlohám
- manuály pro sestavení robotických konstrukcí
- zadání pro studenty a metodické pokyny pro učitele