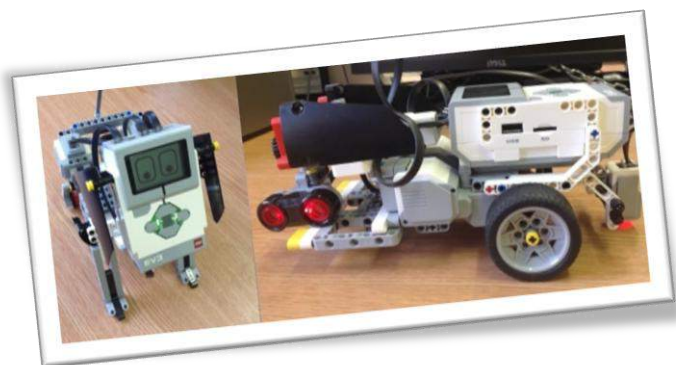


Popularizace vědy ve volnočasových aktivitách žáků SŠ - informatika



Tento modul obsahuje náměty aktivit, které jsou vhodné pro realizaci ve volnočasových kroužcích na střední škole. Jedná se především o aktivity s využitím informatiky a robotiky.

Obsah:

- Motivace k zájmu o studium informatiky
- Možnosti a typy popularizace informatiky
- Lego Mindstorms EV3
- Parkovací asistent
- Robopejsek



Tento materiál vznikl z finanční podpory Evropského sociálního fondu a státního rozpočtu České republiky v rámci projektu „Popularizace vědy a badatelsky orientované výuky“, registrační číslo CZ.1.07/2.3.00/45.0007.

Popularizace vědy ve volnočasových aktivitách žáků SŠ - informatika

Tento modul/kurz je zaměřen na následující témata v kontextu věkové skupiny žáků středních škol: motivace k zájmu o studium technických a přírodovědných oborů, možnosti a typy popularizace vědy, získávání informací z nejnovějších vědeckých výzkumů, náměty pro aktivity zájmového kroužku, náměty projektů, experimentů, tipy na exkurze apod.

Autor:

Mgr. Tomáš JAKEŠ, Ph.D.

Všechny uvedené texty, obrázky a videa jsou vlastní, není-li uvedeno jinak. Autory Youtube embed videí lze nalézt při kliknutí na znak Youtube ve videu během přehrávání.

K plnohodnotnému využití této studijní opory je nutný přístup k on-line zdrojům a materiálům.

Tento materiál vznikl z finanční podpory Evropského sociálního fondu a státního rozpočtu České republiky v rámci projektu „Popularizace vědy a badatelsky orientované výuky“, reg .č. CZ.1.07/2.3.00/45.0007.

1 Popularizace informatiky - úvodem

Kapitola se zaměřuje na popularizaci informatiky se zaměřením na středoškolské vzdělávání. Představuje možnosti uplatnění absolventů na trhu práce. Dále uvádí možnosti a typy popularizace v rámci mimoškolních aktivit. Navrhuje několik možností pro realizace exkurzí. Na závěr jsou zde představeny některé on-line informační zdroje pro získávání nejnovějších vědeckých a technických poznatků.



1.1 Motivace k zájmu o studium informatiky

Výpočetní technika ve společnosti

Role výpočetní techniky a informatiky v našich životech, vzdělání a povolání je nezpochybnitelná. Podíl techniky na různých činnostech je značný, průmyslovou výrobou počínaje a administrativou nebo například vedením záznamů konče. S počítači se dnešní děti setkávají již od brzkých let a s výukou informatiky se začíná již na prvním stupni základní školy.

Pokud se ale zaměříme na středoškolský stupeň vzdělávání, musíme rozlišovat dva typy studentů. První skupinu tvoří ti, kteří studují technicky zaměřený obor a setkají se tak s výukou informatiky ve větší míře. Druhou skupinu tvoří studenti, kteří studují například humanitně zaměřené obory. Vzhledem k postavení a důležitosti výpočetní techniky ve společnosti je ale důležité, aby si obě tyto skupiny studentů pěstovaly vztah k využití informačních a komunikačních technologií a získaly tak konkurenční výhodu na trhu práce.



Motivace ke studiu informatiky

Nyní se zaměříme na to, proč by měli být studenti motivováni ke studiu informatiky. Nejprve se budeme zabývat skupinou studentů s humanitním zaměřením studia. Ačkoliv se zdá, že pro ně využití počítačů v dalším studiu nebo zaměstnání není tolik podstatné, opak je pravdou. Také v humanitních vědách je důležité zpracovávat data, vyhodnocovat je a připravovat různé výstupy a přehledy. Bez počítačů se takováto činnost již v dnešní době neobejde. Ať již tedy budou pokračovat ve studiu na vysoké škole, nebo naleznou uplatnění na trhu práce, znalosti využití výpočetní techniky pro ně budou velice důležité.

Pro úplnost můžeme uvést několik příkladů odvětví, ve kterých je úloha informačních a komunikačních technologií nezpochybnitelná. Jedná se například o administrativní pozice, státní správu, zdravotnictví, vzdělávání nebo výzkum.

Studenti, kteří již na střední škole studují obory zaměřené na informatiku či informační a komunikační technologie, se dostávají do zcela jiné situace. Počítače tvoří jejich hlavní pole působnosti. Své znalosti mohou prohlubovat buďto při studiu na vysoké škole, nebo přímo po absolvování střední školy v zaměstnání. Uplatnění mohou nalézt v oblasti programování a vývoje aplikací, ve které je v současnosti po odborně vzdělaných pracovnících velká poptávka. Podobně je tomu i v oblasti správy firemních počítačových sítí. Další oblast, ve které mohou absolventi nalézt uplatnění, je oprava a údržba počítačů a dalších zařízení. Mnozí z nich se uplatní jako hardwaroví či softwaroví odborníci, jiní jako učitelé či odborní lektoři.

Z tohoto krátkého přehledu a výčtu možného uplatnění absolventů je zřejmé, že bez znalostí výpočetní techniky, její obsluhy a správy alespoň základních aplikací se absolvent střední školy neobejde. Studium informatiky a další i mimoškolní rozvoj v této oblasti je tak velice žádoucí.

1.2 Možnosti a typy popularizace informatiky

Exkurze

Během exkurzí je možné se seznámit jak s aktuálními technologiemi, tak i s jejich historií. Nalezneme mnoho tematických expozic. Mnohé instituce své programy a expozice často obměňují a nabízí aktuální programy jak pro jednotlivce, tak i pro skupiny (např. školní třídy).

- Praha [Národní technické muzeum](#)
- Linec [Ars Electronica Center](#)
- Mnichov [Deutsches Museum](#)
- [Fakulta informačních technologií VUT v Brně](#)
- [Moravské zemské muzeum Brno](#)
- [Techmania Science Center Plzeň](#)
- [IQpark Liberec](#)

Exkurze za účelem poznání principů řídicích systémů

- Návštěva [Rozhlasového muzea](#) v Plzni
- Návštěva některé z poboček [Českého hydrometeorologického ústavu](#)
- Seznámení s řídicím systémem pivovaru [Plzeňský Prazdroj](#)
- Seznámení s vývojem ve [Škoda Transportation](#).

Mimoškolní aktivity

Mimoškolní aktivity slouží k dobrovolnému rozvoji ve zvolené oblasti mimo rámec školního vyučování. Někdy jsou realizovány přímo školami, častěji externími institucemi.

- [Dům digitálních dovedností Plzeň](#) - centrum pro vzdělávání se zaměřením na ICT technologie.
- [Stanice mladých techniků Plzeň](#) (Středisko volného času Radovánek) - centrum provozující značné množství mimoškolních aktivit (programování, robotika a další).
- Návštěva plánovaného [robotického centra](#).

Odborné kurzy

Kurzy správy počítačových sítí - [Cisco Networking Academy](#).

odkazy viz. on-line kurz

1.3 Možnosti získávání informací z nejnovějších vědeckých výzkumů

Pro samostudium nebo vytváření semestrálních prací je potřeba využít co největší množství zdrojů. Pro usnadnění orientace v oblasti informatiky a techniky přikládáme přehled několika on-line zdrojů z různých oblastí. Jedná se jak o webové stránky knihoven, tak i internetové časopisy, portály a weby.

Knihovny a nakladatelství:

Na následujících odkazech naleznete webové stránky technicky zaměřených knihoven a také jednoho nakladatelství věnujícího se vydávání technických časopisů.

- [Národní technická knihovna](#)
- [Studijní a vědecká knihovna Plzeňského kraje](#)
- [Státní vědecká knihovna](#)
- [FCC Public](#) - nakladatelství technických časopisů

Webové stránky věnující se novinkám ze světa vědy a technologií:

Následující odkazy vedou na stránky technicky zaměřených webů představujících také novinky z oblasti technologií a vědy.

- Technet.cz
- [21. století](http://21.stoleti.cz)
- ScienceWeek.cz
- ScienceWorld.cz
- Grafen.cz

Webové stránky věnující se informatice a výpočetní technice:

Následující webové stránky se zaměřují na hardware, software, počítačové sítě, multimédia a jiné oblasti týkající se počítačů.

- CZC.cz
- [PC World.cz](http://PCWorld.cz)
- [Živě.cz](http://Zive.cz)
- JNP.cz
- CHIP.cz
- [Svět hardware](http://SvetHardware.cz)
- [PCTuning](http://PCTuning.cz)
- Root.cz
- [MS Virtual Academy](http://MSVirtualAcademy.cz)

odkazy viz. on-line kurz

2 Robotika v praxi, ve škole i ve volnočasových aktivitách

Roboti nalézají své uplatnění zejména v průmyslu, kde zastávají mnohé nebezpečné, člověkem nerealizovatelné či monotónní operace. Usnadňují nám práci, chrání naše zdraví a zvyšují naši produktivitu. Jedním z odvětví, kde se roboti hojně využívají, je i automobilový průmysl. Roboti provádí přesné svařování, upevňují a doplňují díly, lakují či pomáhají lidem (jejich činnost můžete vidět na videu - viz. on-line kurz).

V trochu jiné formě je však můžeme najít i u domácích spotřebičů (např. samostatných robotických vysavačů) či automobilů. Jejich moderní asistenti umějí díky sensorům samostatně parkovat, detekovat možné kolize či sledovat dopravní značky. V poslední době se experimentuje i s automatickým řízením automobilů (viz video) nebo

<https://youtu.be/uCezlCQNqJU?t=31>.

S rozvojem společnosti bude mít robotika čím dál tím větší význam. Bude potřeba lidí, kteří ji umějí nejen využívat, ale též i konstruovat a programovat. Jenže jak se tomu mají naučit? Určitě se shodneme, že začínat se učit na robotické ruce za desítky miliónu korun či vytvářet vlastní autonomní automobil by nebyla ta nejlepší varianta. Naštěstí existuje celá řada stavebnic či mikropočítačů, které nám pro začátky velmi dobře poslouží.

LEGO Mindstorms EV3



Robotická výuka je řešitelná přes různá zařízení, prostředí, systémy či stavebnice. Jedním z nich je i stavebnice LEGO Mindstorms EV3. Nenechte se mýlit názvem. Nejedná se o klasické kostičky určené ke stavbě domečků, ale o stavebnici LEGO Technics doplněnou o mikropočítač a senzory. Díky tomu lze vytvářet programovatelné roboty, kteří reagují na podněty ze svého okolí. Technické díly LEGO Technics zároveň poskytují možnost tvorby podpůrných konstrukcí a převodů. Stavebnice LEGO Mindstorms EV3 (2013) navazuje na úspěšné stavebnice RCX (1998), NXT (2006) a NXT 2.0 (2009).

Podporu pro stavebnici LEGO Mindstorms naleznete na následujících adresách.

- Kurz programování, moduly a náměty úloh: <http://lego.zcu.cz/>
- Stránky výrobce: <http://www.lego.com/en-us/mindstorms/>
- Stránky českého distributora: <http://www.eduxe.cz/les/robotika-ev3/>

V robotice se koná celá řada soutěží, kterých se mohou zúčastnit žáci základních i středních škol.

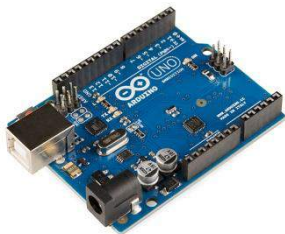
- FLL – First Lego League (EN): <http://www.firstlegoleague.org/>

- FLL – Robotická soutěž FLL (CZ): <http://www.ceskaligarobotiky.cz/item/22-first-lego-league>
- WRO – World Robot Olympiad: <http://www.wroboto.org/>
- Robosoutěž ČVUT: <http://www.robosoutez.cz/>

Alternativy

Pokud stavebnici nedisponujete, můžete si ji buďto zakoupit v některé ze specializovaných LEGO prodejen, či u výhradního českého distributora pro vzdělávání, firmy Eduxe <<http://www.eduxe.cz/>>. Můžete též zvolit i zcela jiný systém konkurenčního charakteru. Abyste se v systémech mohli zorientovat, nabízíme stručné informace o několika variantách.

Arduino



Arduino vzniklo pro jednoduché automatizační úlohy. Jedná se o mikropočítač, ke kterému se pomocí sběrnic připojují další senzory. Hodí se jak pro skutečné automatizační úlohy, tak i pro výuku programování a úvod do automatizace či robotiky. Velkou výhodou je univerzálnost a celá řada doplňků. Nevýhodou je pak nutnost dokoupení či vytvoření pojezdů, konstrukcí, uchycení, převodů a jiných mechanických částí, které nejsou, na rozdíl od stavebnic LEGO, součástí.

- Novinky ze světa Arduino v češtině: <http://arduino.cz/>
- Úvod do Arduina (EN): <http://www.arduino.cc/en/Guide/HomePage>

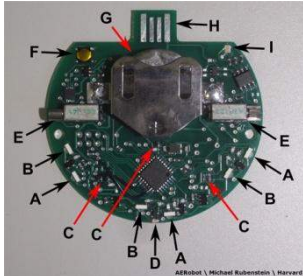
Raspberry Pi 2



Tento mikropočítač je do určité míry konkurencí pro Arduino. Zatímco Arduino je více oblíbenější pro řešení automatizačních úloh, Raspberry Pi sází na vyšší výkon procesoru a chová se více jako klasický počítač či mediální centrum pro televizi. Kromě sběrnic pro připojení externího hardware (senzorů, relé, displejů, ...) je vybaveno HDMI výstupem, microSD slotem, zvukovým výstupem, USB i ethernetem. Raspberry Pi je dodáváno s grafickým programovacím prostředím, které je více vhodné pro výuku žáků nižšího věku.

- Představení stavebnice na oficiálních stránkách (EN): <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>
- Představení na zive.cz: <http://www.zive.cz/clanky/raspberry-pi-2-sestkrat-vyssi-vykonve-steinem-baleni/sc-3-a-177031/default.aspx>

AERobot (Affordable Education Robot)



AERobot je malým jednoduchým robotem pro výuku programování u dětí. Jeho hlavní výhodou je cena, kdy celého programovatelného robota získáte za cca 250 Kč. Robot sice není dále rozšiřitelný, ale zvládá pohyb pomocí vibračních motorků do všech stran, umí měřit vzdálenost od objektů a opticky sledovat podložku a po vytvoření programu pak např. jezdit po čáře.

- Představení robota v EN: <http://wyss.harvard.edu/viewpage/539>
- Představení robota v CZ: <http://www.zive.cz/clanky/robot-za-230-korun-ktery-uci-deti-programovat/sc-3-a-176240/default.aspx>
- Kurz programování robota: <https://sites.google.com/site/aerobotcurriculum/home>

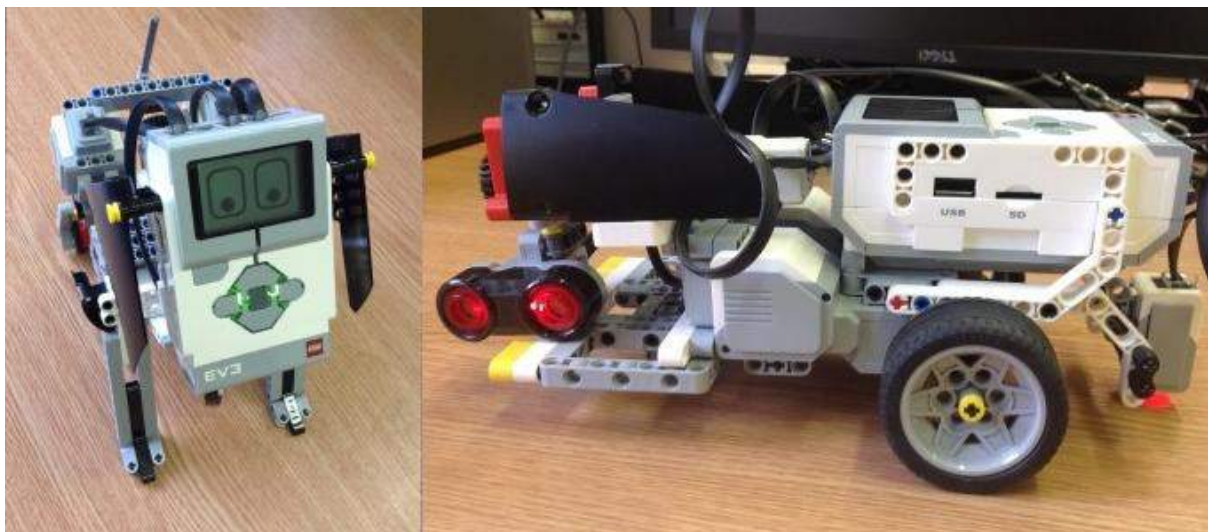
Chtěli jste se dozvědět, co je to robot, a stále to nevíte? Zkuste se podívat na Wikipedii <http://cs.wikipedia.org/wiki/Robot>. Přestože se jedná o terciální zdroj informací, s uvedenou definicí se dokážeme ztotožnit.

3 Náměty pro aktivity zájmového kroužku

Jednou z oblastí, kterou lze v rámci kroužků využít k popularizaci informatiky jako oboru, je i robotika. V rámci řešení úloh robotiky studenti sestavují konstrukci robota, vybavují ho senzory a z nich získané hodnoty dále programově zpracovávají. Informatiky většinou nejvíce zajímá poslední ze zmíněných oblastí, a to zpracování dat pomocí vlastního navrženého programu. Tvorbou programu žáci objevují základní konstrukce programovacích jazyků a prakticky přichází na jejich potřebu. Výhodou je možnost relativně rychlého praktického ověření výsledku.

Tím, že nepracujeme v umělém prostředí, ale v reálném světě, však narážíme na problémy vycházející z jeho podstaty, kdy zdánlivě dobře navržené řešení nemusí být kvůli rušivým vlivům prakticky realizovatelné. Je to nevýhoda i výhoda zároveň. Nutnost odstranění chyby nutí žáky chybu detekovat, přemýšlet nad příčinami jejího vzniku a navrhnout řešení vedoucí k bezproblémovému chodu. Někdy se nám dokonce stane, že původně zamýšlené řešení musíme zcela opustit a navrhnout nové. Řešení tak vyžaduje podstatně více času a dovedností než tvorba programů v umělém, např. počítačovém prostředí. Tento čas je však kompenzován lepším rozvojem schopnosti řešit praktické problémy.

V následujících úlohách budeme používat robotickou stavebnici LEGO Mindstorms EV3. Tato stavebnice dle našich zkušeností i přes její vyšší cenu poskytuje značnou variabilitu a rychlost stavby konstrukcí robota.



3.1 Projekt parkovací asistent

Parkování, ať již v podélné řadě stojících automobilů, nebo v příčném směru může činit mnohým z nás řadu obtíží. Musíme dbát na to, abychom nepoškodili vedle stojící automobily nebo další objekty, jako je osvětlení, obrubníky a další. Dále musí být parkování plynulé, abychom příliš nezdržovali provoz na frekventované ulici.

V dnešní době nám je značným pomocníkem pro tuto někdy zdánlivě obtížnou činnost parkovací asistent. Ten dokáže v okolí detekovat volný prostor a analyzovat, zda je dostatečně velký pro to, aby do něj mohlo vozidlo zaparkovat. Při parkování navíc detekuje předměty v okolí a snaží se předcházet kolizím.

Jak takový parkovací asistent funguje, si můžete prohlédnout na následujícím videu:

video viz. on-line kurz

Parkovací asistent s využitím LEGO Mindstorms EV3

Vytvořit si vlastní automobil obsahující parkovacího asistenta není příliš složité. Využít k tomu můžeme také robotickou stavebnici LEGO Mindstorms EV3. Bude se jednat o automobil opatřený zařízením pro detekci volného prostoru pro podélné parkování v řadě automobilů. Princip funkce bude spočívat v detekci volného prostoru při projíždění kolem řady zaparkovaných vozidel a v hledání dostatečného prostoru pro zacouvání do řady.

video viz. on-line kurz

Řešení této úlohy se může zdát složité, ovšem po splnění jednotlivých dílčích úkolů zjistíte, že vytvoření programu nebylo příliš náročné.

Vyžadované pomůcky:

Pro vytvoření modelu a následně také ovládacího programu budete potřebovat následující:

- základní sadu robotické stavebnice LEGO Mindstorms EV3,
 - řídicí jednotku EV3,
 - ultrazvukový senzor,
 - technické díly,
- programovací prostředí EV3.

Metodický list pro dlouhodobý badatelský projekt

Téma	Automatický parkovací asistent
Tematický celek	LEGO ROBOTIKA
Motivační rámec projektu	Pro řadu lidí není parkování příjemnou činností. Automobilky proto vybavují auta různými automatickými či poloautomatickými systémy, které tuto činnost zpříjemňují či zcela vykonávají místo člověka. Udělat si vlastní model automaticky parkovacího auta není se stavebnicí LEGO EV3 nikterak složité. Pojďme se do toho pustit.
Počet žáků	Ideálním počtem pro řešení této úlohy jsou dva studenti na jednu stavebnici LEGO Mindstorms EV3. Práce ve dvojici umožní optimální využití žáků a poskytne možnost konzultace i konfrontace různých způsobů řešení v rámci skupiny. V případě potřeby je však možno upravit počet na 1 až 3 studenty na jednu stavebnici. U většího počtu studentů často dochází k nevytížení některého z žáků, který si jen velmi těžko hledá možné uplatnění. Při řešení úlohy je možno si práci na vyřešení úlohy rozdělit (1 student pracuje na konstrukci robota, další programují ovládací program).
Věk žáků	15 +
Pomůcky	Pro komplexní realizaci úlohy je zapotřebí toto vybavení: <ul style="list-style-type: none"> robotická stavebnice LEGO Mindstorms EV3 (45544 EV3 Základní souprava), PC s programovacím prostředím LEGO EV3 (či alternativní), překážky vytvářející nevyhovující a dostatečná parkovací místa (úspěšně lze použít krabice od stavebnic). <p>Kompletní projekt vytvořený v programovacím prostředí EV3 ke stažení.</p> <p>Pozn: Úloha je s omezením řešitelná i za pomoci infračerveného senzoru, kdy ultrazvukový senzor může detekovat překážky před autem.</p>
Vhodná místa realizace projektu	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím EV3 (či alternativním) a dostatečným prostorem pro otestování automatického parkovacího systému.

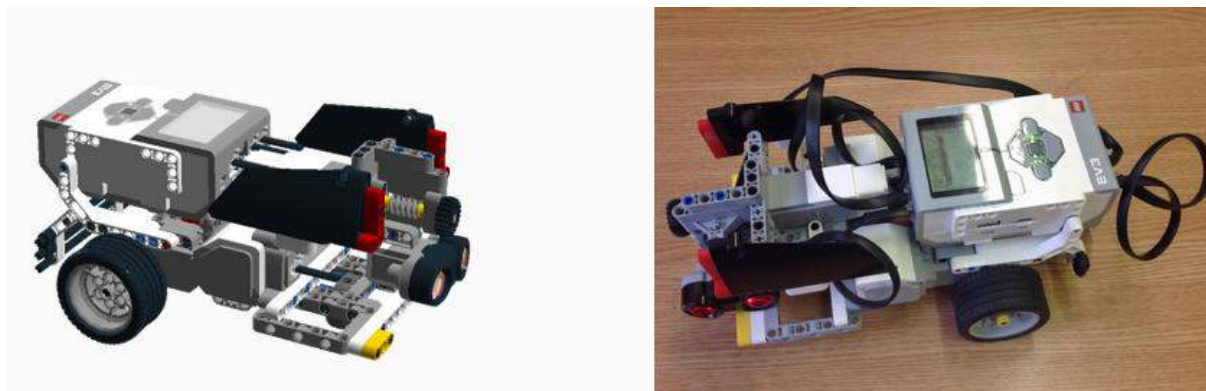
Cíle aktivit	<p>Dosažené cíle v rámci dlouhodobého badatelského projektu se značně liší dle způsobu vedení aktivity. Ta může být omezena na řešení dílčích úloh, zejména na programování zaměřených úloh či rozšířena o celkovou konstrukci robota a vysvětlení ultrazvukové detekce objektů.</p> <p>V případě řešení konstrukce by studenti měli dokázat:</p> <ul style="list-style-type: none"> • vytvořit pevnou samonosnou konstrukci umožňující pohyb a zatáčení auta, • navrhnout umístění a konstrukci otočného ultrazvukového senzoru. <p>Programováním robota by studenti zapojení do této činnosti měli zlepšit své dovednosti v oblastech:</p> <ul style="list-style-type: none"> • využití a ovládání servomotorů, • získávání zpětné vazby ultrazvukovým senzorem, • ovlivnění programu na základě vstupů, • uchování výchozích hodnot (např. směru parkování). <p>Řešení projektu je však do jisté míry závislé na návrhu studentů. Dá se však předpokládat, že po absolvování studenti dokáží:</p> <ul style="list-style-type: none"> • na displeji vykreslit informační texty, • přehrát vybrané zvuky, • realizovat řízení robota na základě hodnot zjištěných ultrazvukovým senzorem, • v rámci zaměření úlohy předcházet možným chybám, • uložit stav získaných hodnot a ten dále zpracovávat, • navrhnout systém zatáčení robota pro zaparkování.
Rozvíjené kompetence	Kompetence k řešení problémů.
Předchozí znalosti	Projekt je vhodný i pro úplné začátečníky se systémem LEGO, kdy je však nejprve vhodné konstrukci autíčka využít k získání dovedností ohledně pojezdu. Následně ho můžeme rozšířit o detekci překážky a teprve až poté přejít na automatické parkování. Výhodou konstrukce auta je jednoduchá možnost předělání např. na model určený pro pohyb po čáře.

Mezipředmětové vztahy	Automatizace, fyzika, matematika
Časový plán	Fáze projektu
	<p>Stanovovat přesný časový plán jednotlivých činností u projektu či badatelsky zaměřené činnosti učitelem není vhodné. Čas je velmi závislý na zkušenostech, znalostech a dovednosti jednotlivých skupin (jedinců) i na způsobu pomoci (či prozrazení podstaty problému) učitelem.</p> <p>Uvedeme proto jen seznam dílčích částí projektu s odhadovaným časovým rozsahem.</p> <ul style="list-style-type: none"> • Úvodní motivace - rozbor úlohy, rozdělení činností (15 min) • Tvorba konstrukce robota <ul style="list-style-type: none"> • podle vlastního návrhu (4 h); • podle schématu zapojení (7 - 25 minut) • Úlohy <ul style="list-style-type: none"> • přímočarý pojezd robota (2 - 10 min) • detekce začátku volného místa, výpisy na obrazovku (4 - 15 min) • pojezd, určení vzdálenosti a detekce konce volného místa (8 - 30 minut) • detekce problému, oprava na plynulou kontrolu volného místa (10 - 30 minut) • opakování režimu detekce při nedostatečné velikosti volného místa (10 - 25 minut) • provedení parkování (12 - 40 minut) • otestování celkové funkčnosti (3 - 10 min) • Presentace úlohy a řešených problémů, reflexe projektu a činnosti skupiny (dle počtu skupin cca 5 - 10 minut na skupinu + čas na závěrečnou reflexi)

Hodnocení	Hodnocení úlohy není vyžadováno. Podstatně větším přínosem je závěrečné představení hotového robota a jeho funkcí před ostatními skupinami. Tato forma více působí na vnitřní zpětnou vazbu a motivaci studentů.
-----------	--

Konstrukce automobilu s parkovacím asistentem

Pro vytvoření modelu automobilu s parkovacím asistentem máte dvě možnosti. Buďto můžete postupovat podle přiloženého návodu, který naleznete níže ke stažení, nebo se můžete pokusit vytvořit model vlastní, který vhodně opatříte potřebnými komponentami.



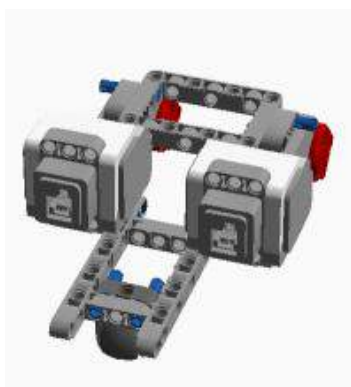
Námi vytvořený soubor konstrukce si můžete stáhnout [<zde>](#). K prohlížení je zapotřebí mít nainstalovaný program LEGO Digital Designer. Pokud tuto možnost nemáte, můžete si vyexportovaný návod s omezenými funkcemi prohlédnout na [této adrese](#).

Pozn.: Program LEGO Digital Designer potřebný k prohlížení a editaci souboru konstrukce je ke stažení na adrese <http://ldd.lego.com/cs-cz/download>.

Problémy řešené při vytváření konstrukce

Základní konstrukce automobilu

Základem vytvoření robustního a stabilního modelu je dobře vytvořená základní konstrukce. V úloze s parkovacím asistentem jsme propojení motorů a vytvoření základní nosné konstrukce realizovali pomocí plošných dílů EV3. Tyto díly nám umožnily vytvořit nosnou plochu pro další prvky konstrukce.



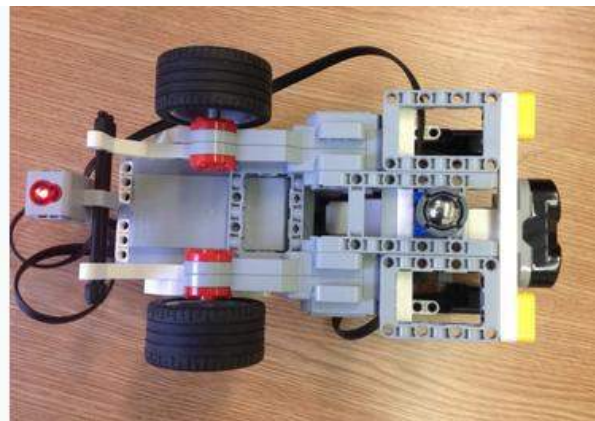
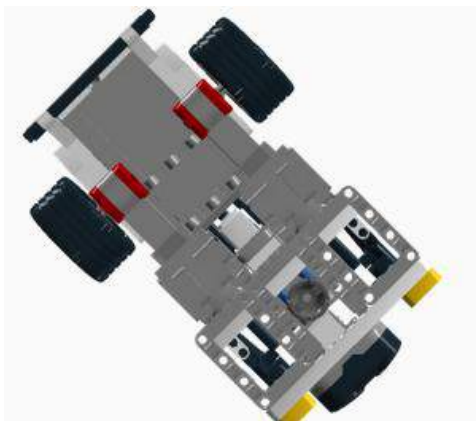
Umístění a možnosti připojení řídicí jednotky

Druhým problémem, který jsme museli řešit, bylo umístění řídicí jednotky na modelu robota. Musíme dbát na to, abychom mohli bezproblémově k jednotce připojit jak vodiče připojených motorů a senzorů, tak také USB vodič pro komunikaci řídicí jednotky s počítačem (pokud nepoužíváme Bluetooth komunikaci). Jelikož model využívá k pohonu dva vpředu umístěné servomotory, je dobré řídicí jednotku umístit do jeho přední části, čímž docílíme zatížení přední nápravy modelu, a tedy jeho lepší pohyblivosti (eliminujeme možnost podklouznutí předních pneumatik).



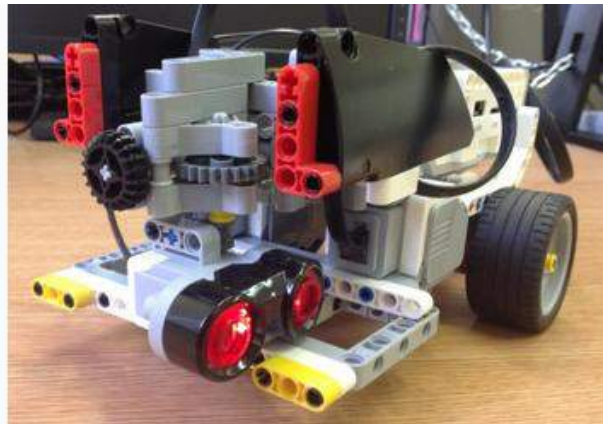
Manévrovatelnost modelu

Jelikož automobil se musí během jízdy natáčet, aby co nejlépe zaparkoval na volné, detekované místo, musíme co nejvíce zlepšit jeho manévrovatelnost. K tomu nám napomůže ocelová rejdivací kulička, kterou umístíme na podvozek na zadní části auta.



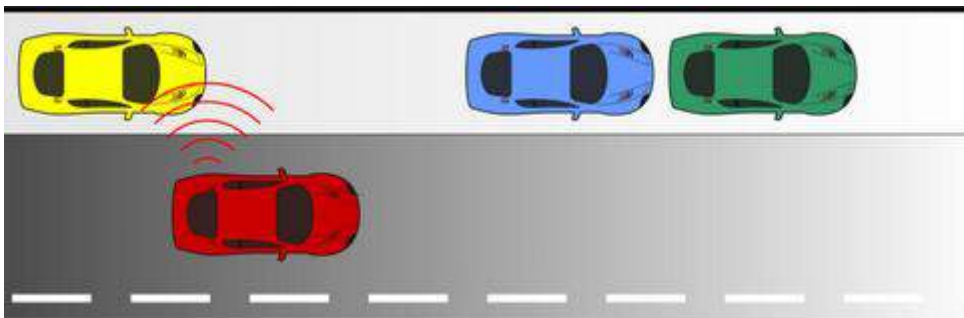
Umístění a natáčení ultrazvukového senzoru

Nejsložitější částí robota je umístění ultrazvukového senzoru. Abychom zajistili variabilitu modelu, potřebujeme vytvořit konstrukci, která nám umožní parkovací senzor (ultrazvukový senzor) natáčet všemi směry. Vytvořili jsme proto za pomoci ozubených kol mechanismus, který tuto funkci umožňuje. Jeho podrobnější rozbor naleznete v příloženém souboru s kompletní realizací modelu v LEGO Digital Designer.



Nalezení volného místa

Prvním krokem, který musíme při vytváření ovládacího programu pro parkovacího asistenta vykonat, je detekce volného místa na zaparkování. Jak jsme již předeslali v úvodu, bude se tak dít pomocí ultrazvukového senzoru umístěného na boku modelu.



Popis programu:

Představíme si programový zápis tohoto jednoduchého úkonu pro lepší pochopení. Robot se bude pohybovat podél pomyslné řady zaparkovaných automobilů a pomocí ultrazvukového senzoru bude detekovat, jak velká je naměřená vzdálenost od jeho boční strany. Pokud bude nižší než 25 cm, bude hodnota vyhodnocena tak, že se v řadě nachází automobil nebo jiná překážka, a není tak možné zastavit. Jakmile se ale vzdálenost zvětší nad 25 cm, robot okamžitě zastaví. Detekoval totiž místo, kde by mohl začínat prostor, který je potenciálně dostatečně velký pro zaparkování.

Pro lepší názornost jsme program ještě doplnili o zvukové a vizuální prvky signalizace. Před rozjetím robota se ozve signál Searching, který nás informuje o započetí snímání. Zároveň se na displeji zobrazuje informace VYHLEDÁVÁM MÍSTO. Jakmile je detekován prostor pro parkování, vypíše se na displej informace NALEZENO a zazní informační hláška Detected.



Doporučení:

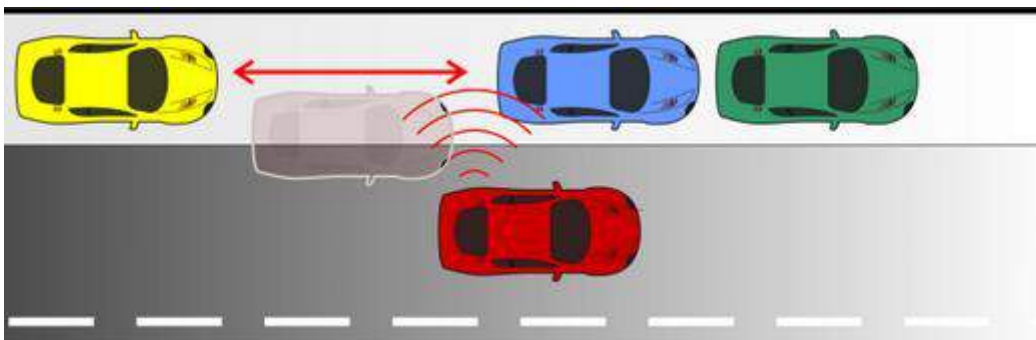
Pro zpřehlednění programu si můžeme pro některé z částí programu vytvořit vlastní bloky, které obsahují část zdrojového kódu programu (několik programových bloků). Na obrázku můžete vidět tentýž program po vytvoření dvou nových bloků. První pro vyhledávání s

informačním výpisem a druhý pro situaci, kdy je volné místo již nalezeno. V dalších částech projektu tak již můžeme využívat tento zjednodušený programový konstrukt.



Měření potřebné vzdálenosti

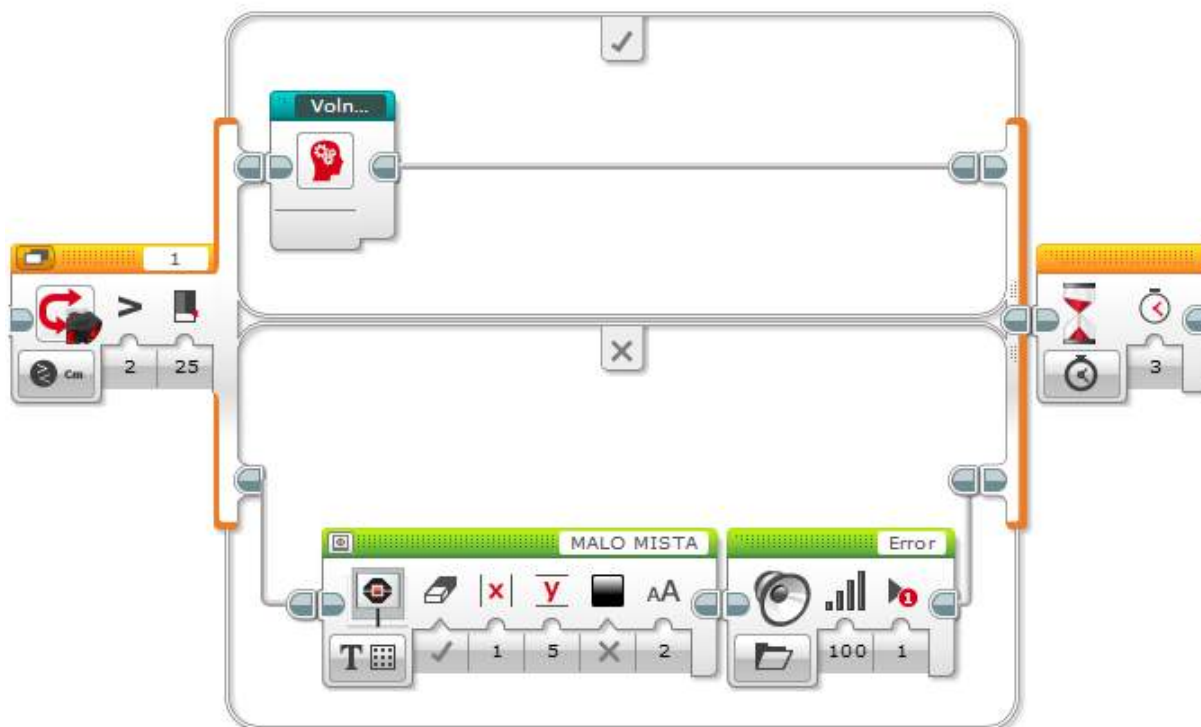
Jakmile náš automobil dokáže zjistit, kde začíná volný prostor, musíme ho naučit změřit, zda je prostor dostatečně velký pro zaparkování. Nejjednodušší metodou pro tento úkon je ta, že si změříme, jak dlouhé naše vozidlo je a necháme robota tuto vzdálenost urazit. Když ji ujede, znovu pomocí ultrazvukového senzoru zjistíme, zda se na místě, kde zastavil, nachází stále mezera vhodná pro zaparkování.



Na obrázku vidíte program z úlohy [Nalezení volného místa](#) rozšířený o pojezd o určitou délku. Pro svůj model jsme jako vhodnou délku zvolili dvě otáčky motoru. Čítač otáček musíme nejprve vynulovat a až následně dvě otáčky provést. Po ujetí vzdálenosti se robot zastaví.

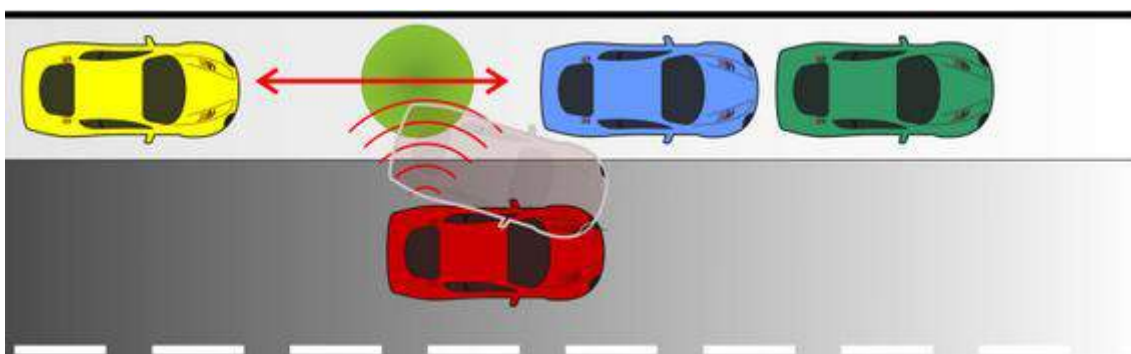


Robot sice od okamžiku, kdy detekoval volné místo pro zaparkování, urazil vzdálenost potřebnou pro zaparkování, nevím ovšem, zda i na tomto místě je stále volné parkovací místo. Opět tedy musí změřit, zda se zde nachází dostatečně velká mezera (hodnota > 25). Pokud zde prostor pro zaparkování je, na displej se vypíše informace NALEZENO a zazní informační hláška Detected. Pokud zde ale již prostor není, což znamená, že od místa, kde jsme detekovali začátek mezery, až na toto místo není dostatečně velký prostor pro zaparkování, vypíše se displej informace MALO MISTA a zazní informační signál Error představující chybu.

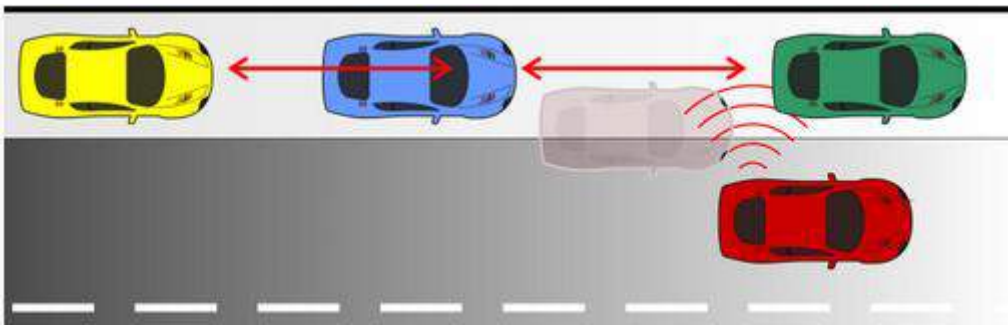


Fungoval by parkovací asistent bez problémů tímto způsobem v reálném světě?

Když parkovací asistent detekuje volné místo pro parkování, měří vzdálenost také pouze na začátku a na konci? Musíme počítat se situací, že mezi zaparkovanými vozidly sice volné místo bude, ale může se zde nacházet lampa veřejného osvětlení, popelnice, sloupek nebo zaparkovaný motocykl. Naučíme se proto ještě program rozšířit tak, aby parkovací asistent detekoval volné místo v celé jeho délce a ověřoval, že se zde žádná překážka nenachází. Více v kapitole [Měření s překážkou](#).

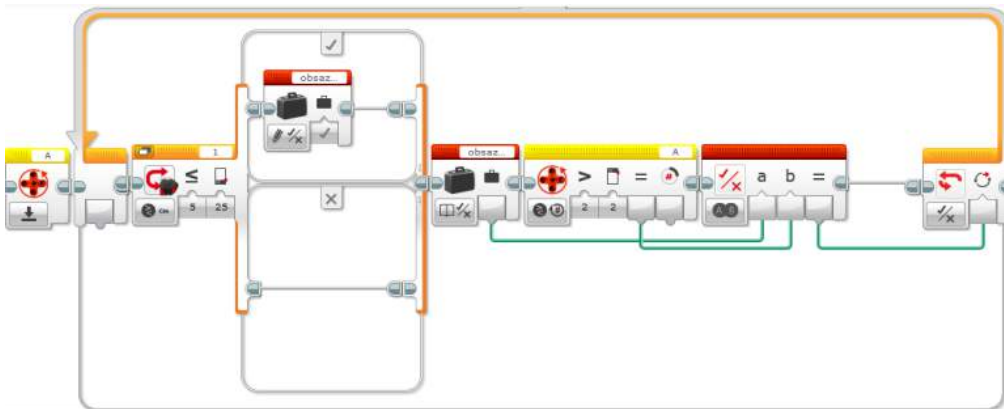


Měření s překážkou

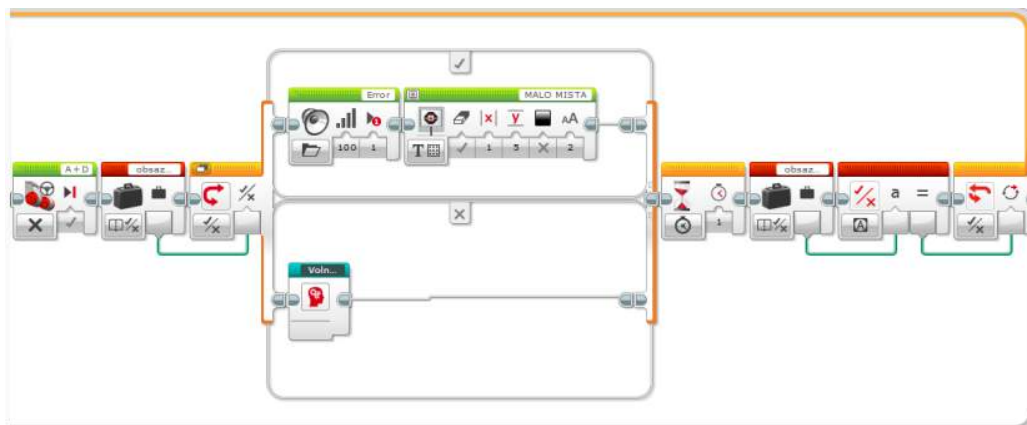


Při realizaci programu, který ověřuje, zda se v detekované mezeře mezi zaparkovanými automobily nenachází překážka, opět rozšíříme náš vytvářený program. Vytvoříme si proměnnou, která je logického datového typu (Logic). Na základě hodnoty, kterou si do ní budeme ukládat, budeme rozlišovat, zda je v detekovaném místě obsazeno, či nikoliv.

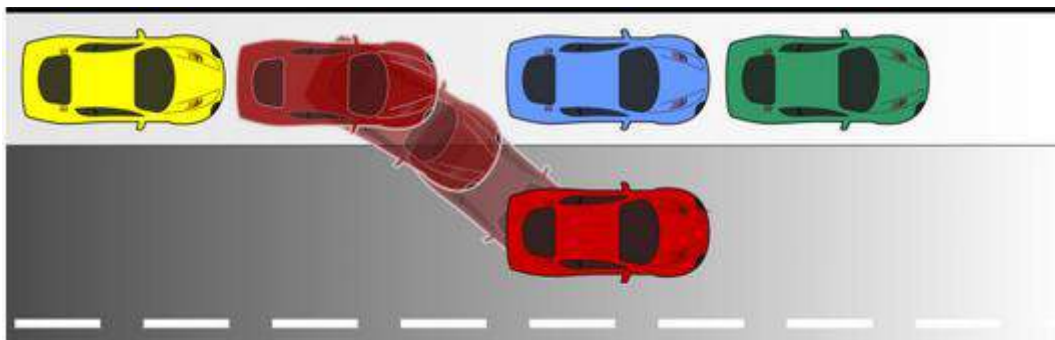
Na prvním obrázku můžete vidět cyklus, který nejprve ověří, zda je na daném místě dostatečný prostor pro zaparkování, nebo zda se v něm nenachází překážka. Pokud některá z těchto situací nastane, cyklus skončí.



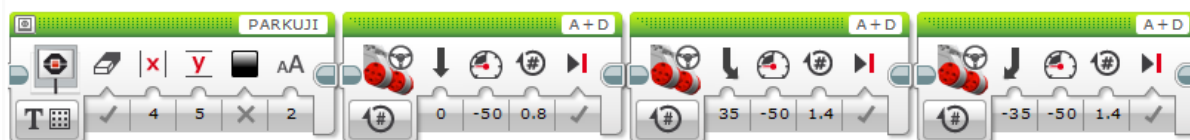
Dostaneme se v programu dále, kde znovu ověřujeme, zda je na měřeném místě obsazeno, či nikoliv. Pokud ano, je zhlášena chyba a vypsána informace, že se zde nachází málo místa. Pokud není obsazeno, je vypsáno na displej, že je zde volno pro zaparkování. Tyto kroky ověřování volného místa se provádí tak dlouho, dokud parkovací asistent nenalezne vhodné místo na zaparkování.



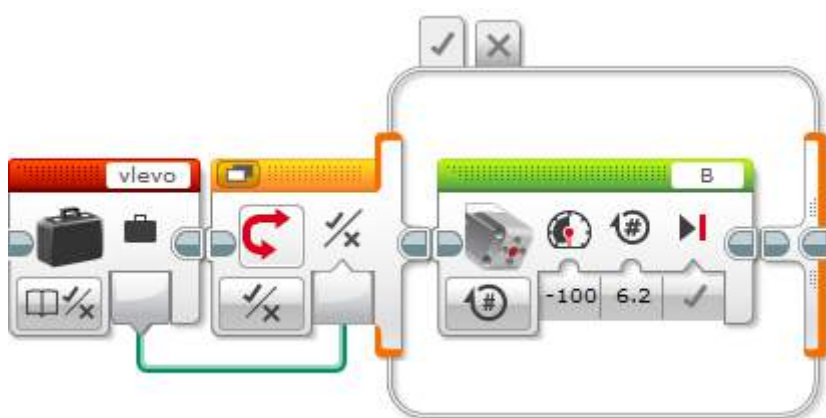
Volba strany pro parkování a parkování



Když náš robot úspěšně nalezne místo, do kterého může bezpečně zaparkovat, musíme ho tuto činnost naučit. Při couvání využijeme bloky pro synchronní řízení dvou motorů. Na obrázku můžete vidět část programového kódu, kterou je nutné přidat na závěr programu, aby robot do řady stojících vozidel zacouval. Hodnoty natočení motorů se mohou samozřejmě lišit případ od případu. Vše záleží na tom, jak máte připravenou dráhu, na které robot mezi dvě vozidla nebo jiné překážky couvá.



S naším programovým zápisem bychom ovšem dokázali parkovat pouze na jednu stranu. Konstrukci robota jsme ovšem opatřili otáčivým ultrazvukovým senzorem pro možnost parkovat jak vlevo, tak i vpravo. Ukážeme si proto, jak zrealizovat tento otáčivý pohyb senzoru. Navigaci můžeme řídit například tlačítky řídicí jednotky. Po stisku levého tlačítka se sensor natočí vlevo, při stisku pravého vpravo. Na obrázku můžete vidět příklad natáčení. Proměnná vlevo je logického datového typu. Jakmile její hodnota bude true, pomocí vestavěného motoru se ultrazvukový sensor natočí vlevo. Kompletní program s natáčením senzoru naleznete ke stažení v části [Metodický list pro dlouhodobý badatelský projekt](#).



3.2 Projekt RoboPejsek na dálkové ovládání

Ne vždy jsou roboti určeni jen pro práci. V Japonsku, v zemi, která je v oblasti robotiky průkopníkem, vznikli dokonce i robotičtí mazlíčci. Pomineme-li fakt, že skutečného živého mazlíčka snad nikdy nenahradí, jedná se o zajímavý koncept. Tyto hračky s názvem Aibo, vyvinuté firmou SONY, se umějí orientovat v prostoru, rozpoznávat barvy, vnímat směr zvuku, pohybovat se a simulovat 6 emocí. Na to, že vznikly v roce 1999, toho neumí vůbec málo. Cena však byla astronomická a dosahovala 45 000 dolarů.



zdroj: http://wallpapers.free-review.net/wallpapers/21/Entertainment_Robot_Aibo_-_Dogs.jpg

Vlastní RoboPejsek

Pokud máme k dispozici robotickou stavebnici LEGO, můžeme si postavit robotického pejska za zlomek ceny. Nejspíše nebude tak dokonalý, ale bude náš a i jeho chování si můžeme sami naprogramovat. A co by měl dělat? Má-li být pejskem, měl by určitě za námi chodit a měl by poslouchat některé příkazy, např. sedni a lehni. Můžete ale vymyslet a pejska naučit celou řadu dalších činností. Na ukázkou možného řešení se můžete podívat na níže umístěném videu. (video viz. on-line kurz)

Že nevíte, jak na to? Nevadí, společně si ukážeme, jak začít, a poradíme si i některé finty. Třeba u sledování a zadávání příkazů si pomůžeme infračervenou ovladačkou, ale to již předbíhám. ;-)

Vyžadované pomůcky

Pro komplexní realizaci úlohy je zapotřebí toto vybavení:

- robotická stavebnice LEGO Mindstorms EV3 ([45544 EV3 Základní souprava](#)),
- PC s programovacím prostředím LEGO EV3 (či alternativní),
- infračervený senzor [45509 EV3 IR senzor](#),
- infračervený ovladač [45508 EV3 IR Ovladač](#).

TIP: Úloha je s omezením řešitelná i bez infračerveného senzoru a vysílače. Příkazy lze zadávat tlačítky či pomocí barev vložených pod alternativně připojený barevný senzor (součástí sady). Pro řízení pohybu pejska je možno využít ultrazvukový senzor odhalující překážky. Pejsek se pak pohybuje, dokud je před ním volno, a v případě trvajících překážek se otočí.

Metodický list pro dlouhodobý badatelský projekt

Téma	RoboPejsek
Tematický celek	LEGO ROBOTIKA
Motivační rámec projektu	Úloha je postavena na tvorbě robotického "mazlíčka", který vykonává studentem naprogramované příkazy, jako je např. sedni, lehni, štěkej, hlídej či aby studenta sledoval jako pejsek na vodítku. K řízení je použit infračervený senzor a ovladač.
Počet žáků	Ideálním počtem pro řešení této úlohy jsou dva studenti na jednu stavebnici LEGO Mindstorms EV3. Práce ve dvojici umožní optimální využití žáků a poskytne možnost konzultace i konfrontace různých způsobů řešení v rámci skupiny. V případě potřeby je však možno upravit počet na 1 až 3 studenty na jednu stavebnici. U většího počtu studentů často dochází k nevytížení některého z žáků, který si jen velmi těžko hledá možné uplatnění. Při řešení úlohy je možno si práci na vyřešení úlohy rozdělit (1 - 2 studenti pracují na konstrukci robota, další programují ovládací program).
Věk žáků	15+
Pomůcky	Pro komplexní realizaci úlohy je zapotřebí toto vybavení: <ul style="list-style-type: none"> • robotická stavebnice LEGO Mindstorms EV3 (45544 EV3 Základní souprava), • PC s programovacím prostředím LEGO EV3 (či alternativní), • infračervený senzor 45509 EV3 IR senzor, • infračervený ovladač 45508 EV3 IR Ovladač. <p>Pozn: Úloha je s omezením řešitelná i bez infračerveného senzoru a vysílače. Příkazy lze zadávat tlačítky či pomocí barev vložených pod alternativně připojený barevný senzor (součástí sady). Pro řízení pohybu pejska je možno využít ultrazvukový senzor odhalující překážky (pejsek se pohybuje, pokud je před ním volno).</p>
Vhodná místa realizace projektu	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím EV3 (či alternativním) a dostatečným prostorem pro otestování řízení pejska ovladačem.

Cíle aktivit	<p>Dosažené cíle v rámci dlouhodobého badatelského projektu se značně liší dle způsobu vedení aktivity. Ta může být omezena na řešení dílčích zejména na programování zaměřených úloh či rozšířena o celkovou konstrukci robota a vysvětlení možnosti navigace v rámci prostoru.</p> <p>V případě řešení konstrukce by studenti měli dokázat:</p> <ul style="list-style-type: none"> • vytvořit pevnou samonosnou konstrukci umožňující pohyb pejska, • navrhnout a sestavit převod pro ovládání nohou motorem s větším počtem otáček, • vhodně uchytit senzory s ohledem na jejich použitelnost a design robota. <p>Programováním robota by studenti zapojení do této činnosti měli zlepšit své dovednosti v oblastech:</p> <ul style="list-style-type: none"> • využití a ovládání servomotorů, • získávání zpětné vazby (stisk tlačítek, měření vzdálenosti a zjištění segmentu) z infračerveného vysílače, • ovlivnění programu na základě různorodých vstupů. <p>Řešení projektu je však do jisté míry závislé na návrhu studentů. Dá se však předpokládat, že po absolvování studenti dokáží:</p> <ul style="list-style-type: none"> • na displeji vykreslit různé druhy obrázků dle vstupů (pohyb očí), • přehrát vybrané zvuky, • realizovat řízení robota s ohledem na možné chyby (dvojstisk tlačítka pro sedni s již vykonávaným příkazem sedni).
Rozvíjené kompetence	Kompetence k řešení problémů.
Předchozí znalosti	<p>Projekt není vhodný pro úplné začátečníky práce se systémem LEGO.</p> <p>Problematická je zejména konstrukce, která musí být samonosná, pohyblivá a ve které je potřeba vytvořit převod pro ovládání zadních nohou třetím motorem. Při poskytnutí návodu na sestavení je však konstrukce bezproblémová.</p> <p>Úlohu je možno řešit i s méně zkušenými studenty. V tom případě je pak vhodné žákům poskytnout dostatečnou oporu a úlohy společně rozdělit na dílčí snáze řešitelné části.</p>

Mezipředmětové vztahy	Automatizace, fyzika, matematika
Časový plán	Fáze projektu
	<p>Stanovovat přesný časový plán jednotlivých činností u projektu či badatelsky zaměřené činnosti učitelem není vhodné. Čas je velmi závislý na zkušenostech, znalostech a dovednosti jednotlivých skupin (jedinců) i na způsobu pomoci (či prozrazení podstaty problému) učitelem.</p> <p>Uvedeme proto jen seznam dílčích částí projektu s odhadovaným časovým rozsahem.</p> <ul style="list-style-type: none"> • Úvodní motivace - rozbor úlohy, rozdělení činností (30 min) • Tvorba konstrukce robota <ul style="list-style-type: none"> • podle vlastního návrhu (20 - 50 h); • podle schématu zapojení (0,25 - 1,5 h) • Úlohy <ul style="list-style-type: none"> • pozdrav při zapnutí (vykreslení očí, přehrání zvuku) (3 - 25 min) • štěkání na povel, využití cyklu a podmínky pro detekci povelů ovládačky (3 - 25 min) • posazení a narovnání pejska (5 min - 2 h: v závislosti na problémech s konstrukcí pejska) • lehnutí a narovnání pejska (5 min - 1 h: v závislosti na problémech s konstrukcí pejska) • režim hlídej - pojezd a vrčení pejska (8 min - 1,5 h: v závislosti na problémech s konstrukcí pohonu) • virtuální vodítko - sledování IR ovládačky <ul style="list-style-type: none"> • pojezd robota na určitou vzdálenost od ovládačky (15 - 45 min) • sledování segmentu - natočení očí (8 min - 1,5 h) • sledování segmentu - přizpůsobení pohybu - zatáčení (30 min - 3 h) • otestování celkové funkčnosti (5 - 10 min) • Prezentace úlohy a řešených problémů, reflexe projektu a činnosti

	skupiny (dle počtu skupin cca 5 - 10 minut na skupinu + čas na závěrečnou reflexi)
Hodnocení	Hodnocení úlohy není vyžadováno. Podstatně větším přínosem je závěrečné představení hotového pejška a jeho funkcí před ostatními skupinami. Tato forma více působí na vnitřní zpětnou vazbu a motivaci studentů.

Konstrukce RoboPejska

Úlohu RoboPejska lze řešit několika způsoby, konstrukcí na základě návodu a konstrukcí vlastního robota. Jsou odlišné zejména zaměřením, množstvím pomůcek, ale též i časovou náročností.

Konstrukce na základě návodu

Prvním jednodušším a časově méně náročným způsobem je nechat studenty sestavit RoboPejska podle poskytnutého návodu. Konstrukce podle návodu není časově tak náročná jako tvorba vlastní konstrukce, navíc vede k jistému výsledku, který v případě vlastní tvorby nemusí být vždy zaručen. Nemůžeme však očekávat, že se tímto způsobem budou u studentů rozvíjet schopnosti vytvářet vlastní robotické konstrukce. Účelem tohoto kroku je urychlit konstrukci a zbývající čas dát k dispozici pro tvorbu řídicího programu RoboPejska.



Námi vytvořený soubor konstrukce si můžete stáhnout [zde](#). K prohlížení je zapotřebí mít nainstalovaný program LEGO Digital Designer. Pokud tuto možnost nemáte, můžete si vyexportovaný návod s omezenými funkcemi prohlédnout na [této adrese](#).

Pozn.: Program LEGO Digital Designer potřebný k prohlížení a editaci souboru konstrukce je ke stažení na adrese <http://ldd.lego.com/cs-cz/download>.

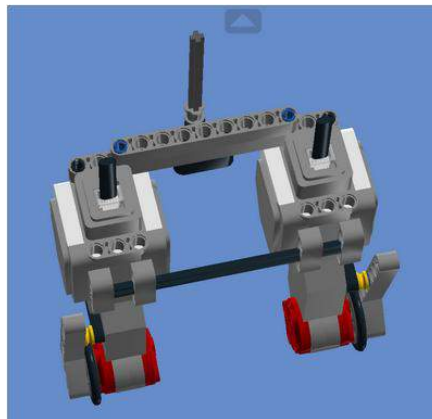
Tvorba vlastní konstrukce

Druhým postupem je možnost nechat studenty robotického pejska nejen naprogramovat, ale i vymyslet a sestavit jeho konstrukci. Pejsek by měl umět vykonat povely: sedni, lehni, štěkej a měl by se umět pohybovat za signálem infravysílače. Důležité je, aby si studenti uvědomili, co všechny tyto funkce znamenají pro konstrukci jejich robota, a domysleli, jakým způsobem pejsek dosáhne toho, aby si např. sedl či lehl.

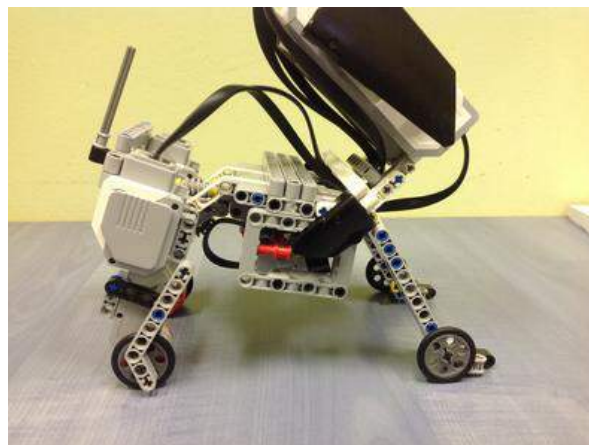
Nemáte-li nakoupeno dostatečné množství dalších technických dílů, studenti též brzy zjistí, že jsou konstrukčně limitováni a mohou používat jen díly, které jsou dostupné.

Pohyb pejska

Aby pejsek uměl jezdit za signálem, musí být vybaven motorkem. Pokud by se měl pohybovat pouze rovně, bylo by možno konstrukci vyřešit pomocí jediného motoru. Pro zatáčení pejska do stran je však potřeba konstrukci vybavit dvěma motory (možnost volby směru (rychlosti) do jednotlivých stran). Pokud by měl pejsek chodit pomocí posunu všech nožiček, byla by konstrukce ještě náročnější a nejspíše bychom potřebovali větší množství motorů, než je počet obsažený v základní sadě.



Při řešení pohybu robota je možno využít tvaru servomotorů tak, aby připomínaly nohy. Tíha motorů zároveň bude zvyšovat přítlak kolečka na podložku.



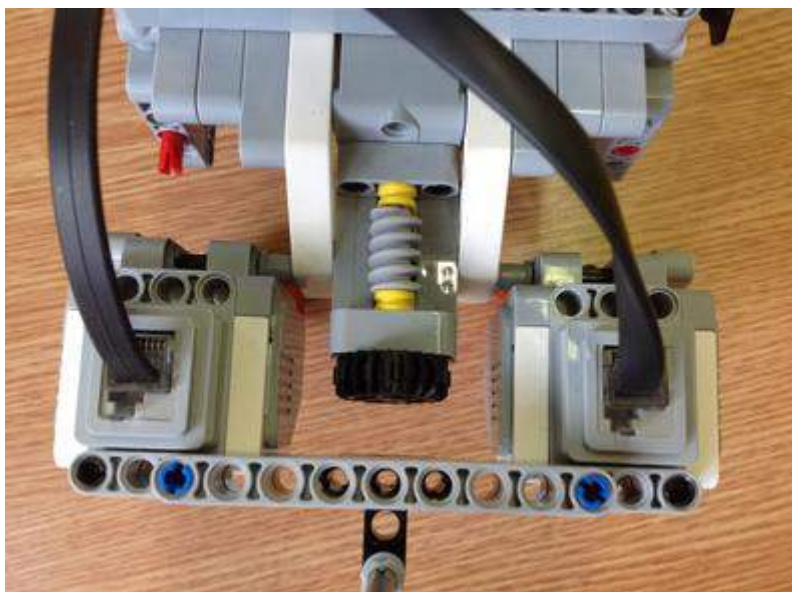
Prvotní myšlenkou bylo robotického pejska vybavit čtveřicí koleček pro snazší zatáčení. Jak se ukázalo, pejsek se skutečně snáze pohyboval směrem vpřed, dokonce si i lépe lehal a sedal. Problém však nastal při ovládní IR vysílačkou, kdy pryžová kolečka kladla v zatáčení značný odpor. Pejsek navíc nevypadal jako pejsek, a tak byla kolečka odstraněna a nahrazena kluzáky. Ty snáze mění směr, ale nehodí se pro nerovné či drhnuocí povrchy. Délka předních noh byla nastavena dle výšky zadní části pejska.

Sedání a lehání pejska

Sedání a lehání pejska je nejsnáze realizovatelné pohybem noh vůči tělu. Vytvořené zadní nohy jsme proto jako celek připevnili na osičku, se kterou jsme měli v úmyslu hýbat. Pohyb měl být realizován velkým motorem, který má průchozí uchycení a dostatečnou sílu. Na rozdíl od předchozí verze NXT je však stavebnice EV3 vybavena jen dvěma velkými motory, které jsme potřebovali již pro realizaci pohybu. Využít jsme proto museli střední motor s vyšším počtem otáček a neprůchozím kolmým uchycením.

Problém č. 1 - kolmé uchycení: Střední motor má řadu výhod, pro tuto realizaci se však příliš nehodil. Při přímém uchycení noh by vyčníval zcela mimo pejska. Bylo proto potřeba vymyslet takový převod, který by umožňoval podélné umístění a přichycení motoru v těle pejska a zároveň by ovládal k němu kolmě umístěnou osičku.

Problém č. 2 - výkon a rotace: Po úspěšné realizaci uchycení a kolmého převodu jsme očekávali, že vše již půjde bez dalších problémů. Pro převod jsme vymysleli dvě varianty. Jedna využívala stejně velká ozubená kolečka, která převáděla v poměru 1:1, druhá konstrukce měla z důvodu zmenšeného prostoru kolečko na motoru větší než kolečko na ose a tak pohyb osy oproti motoru urychlovala. Tato varianta se ukázala být zcela nepoužitelná a nedokázala s osou pohnout. Též varianta převodu 1:1 nebyla dobrá. Střední motor byl příliš rychlý a zadníma nohama hýbal příliš. Následnou zátěž při návratu nohou nezvládl a zasekl se, případně poškodil konstrukci. Řešením bylo pohyb motoru zpřevodovat do pomala a tím zvýšit ovladatelnost i sílu motoru. Z důvodu omezeného prostoru se jako ideální ukázalo využití šnekového převodu.



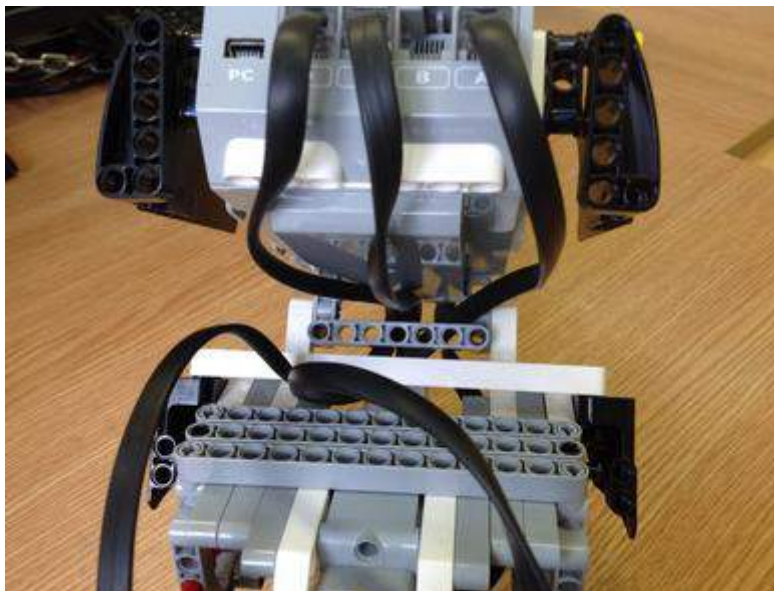
Problém č. 3 - zkrut osičky: Zadní nohy byly propojené jen pomocí křížové osičky. Jak se ukázalo, byla tato osička při pohybu motorů velmi namáhána, pejsek vstával nerovnoměrně a hrozilo její poškození. Pro rovnoměrné přenášení krutu byly obě části zpevněny a propojeny dvojicí rovných kostek, které můžete vidět na obrázku výše.

Problém č. 4 - nastavení nohou a překlápění robota: Až při tvorbě programu jsem zjistil, jak je otravné nastavovat nohy po nevydařeném pokusu do výchozí polohy. Model jsem proto mírně upravil a doplnil o ozubené kolečko pro možnost manuálního nastavení úhlu nohou vůči tělu při vypnutém robotu (viz černé ozubené kolečko na obrázku výše). Správné nastavení úhlu nohou bylo důležité zejména při návratu ze sedu, kdy se při špatném odhadu pejsek příliš zaklonil a převrátil se na záda. Ke zmírnění dopadu mimochodem sloužil také ocásek, který část energie pohltil a nasměroval pád robota do strany tak, aby primární pád nebyl na řídicí jednotku.

TIP pro verzi NXT: Velké servomotory sady NXT nemají kříže pro uchycení osičky pro náklon nohou. Tento problém můžete částečně obejít konstrukcí využívající zpevněné křížové T. Konstrukci naleznete na [této adrese](#). U verze NXT je potřeba též, s ohledem na chybějící horní díry pro upevnění, upravit zpevnění konstrukce nohou.

Upevnění hlavy

Upevnění hlavy se zdálo být jednoduché. Opak byl však pravdou. Hlava tvořená řídicí jednotkou je nejtěžší částí robota, a proto ji bylo nutno umístit nejen tak, aby připomínala pejska, ale též i tak, aby se model při pohybu a sedu nepřeklápěl. Ani v našem modelu není vyvážení zcela optimální a při přílišném sedu způsobuje překlopení, ale po správném nastavení nedělá větší problémy. Hlava je umístěna pod úhlem, přední nohy jsou předsazeny.

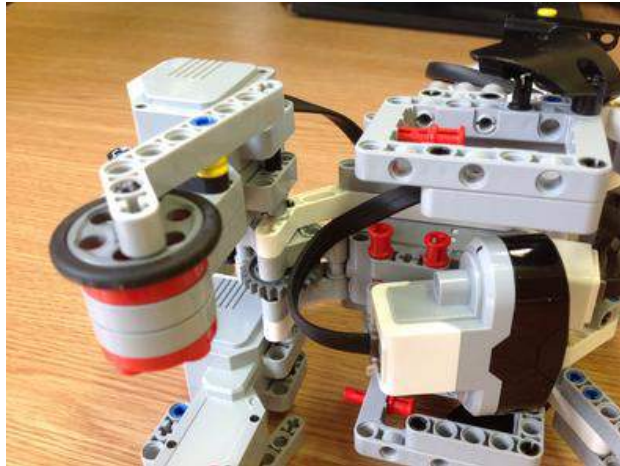


Umístění senzoru

Ideálním místem pro umístění infračerveného senzoru je hlava. Senzoru nic nepřekáží, má plný rozhled a je navíc umístěn na nejvyšším místě, takže robot lze ovládat i ze stoje. Robot fungoval velmi dobře, jenže jako pejsek nevypadal. Rozhodli jsme se proto senzor umístit tak, aby nebyl příliš vidět a nekazil estetický dojem. Z technických důvodů bylo vybráno

místo pod tělem (sedacím motorem) pejska. Nevýhoda tohoto řešení se objevila při ovládní. Díky nohám, které částečně zakrývají přímý výhled a odrážejí infračervené paprsky, se někdy stane, že snímač špatně vyhodnotí vzdálenost a polohu vysílače v určitém segmentu. Pejsek se pak nechová zcela podle našich očekávání, ale zase vypadá jako pejsek ;-).

Pro lepší ovládní z výšky jsme senzor mírně natočili směrem vzhůru. Přední červená spojka senzor držela, druhá zadní ho jen nadzvedávala.



Dokončení pejska

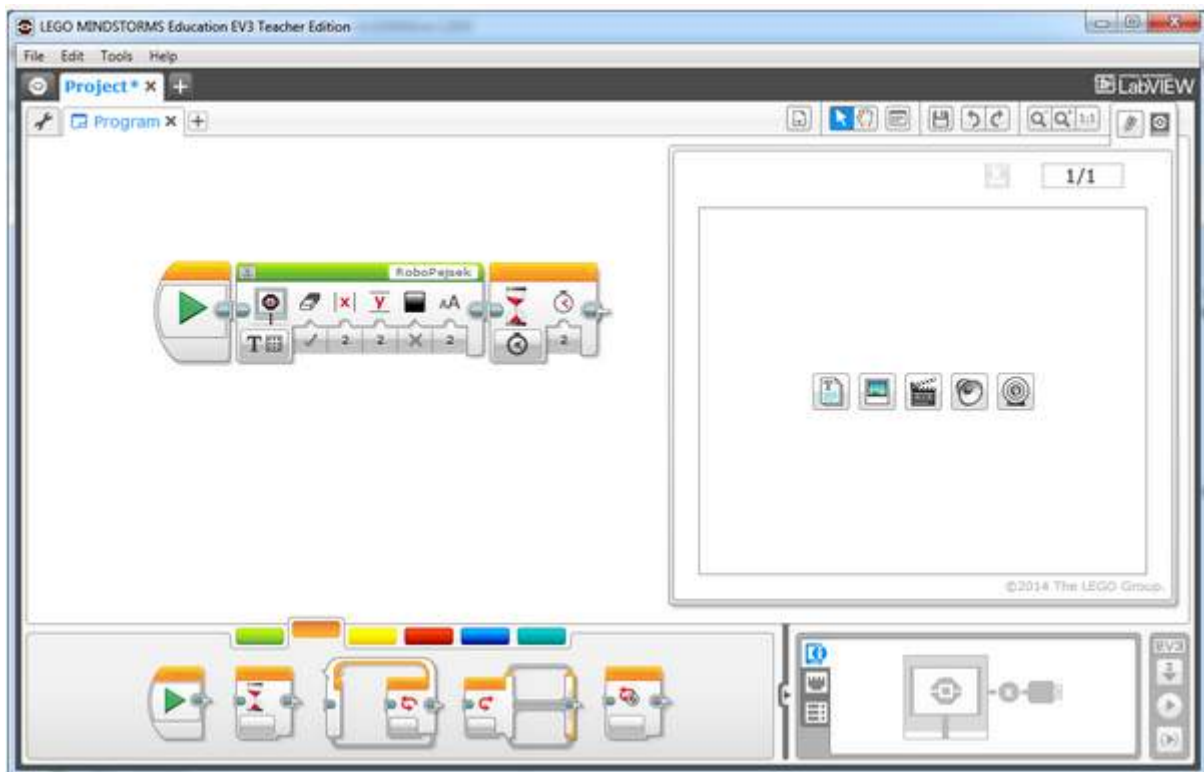
Na závěr už zbývalo dokončit pejska tak, aby vypadal jako pejsek. Přidali jsme plandající uši, bočnice a příčné spojky připomínající trup, drápky i dekoraci nožiček částečně zakrývající kolečka a přichytili kabely.



Programové řešení RoboPejska

Hlavní konstrukci robotického pejska máme již za sebou. Můžeme sice ještě objevit nedokonalosti a konstrukci upravit, ale i pro objevení těchto chyb potřebujeme pejska naprogramovat. Nyní nás proto čeká tvorba programu.

V rámci projektu RoboPejsk budeme používat grafické programovací prostředí LEGO Mindstorms Education EV3, jehož výhodou je relativní jednoduchost. Prostředí je vhodné zejména pro výuku začátečníků, ale poskytuje i některé rozšiřující funkce pro pokročilejší uživatele. Plnohodnotné prostředí je sice placené, ale existuje i zdarma dostupná omezená varianta [Home](#), která by nám mohla postačovat.



Rozdělení programového řešení

Řešit celý program najednou by bylo značně obtížné a z výukového pohledu nevhodné. Méně zkušení studenti by se v programu a při jeho řešení ve vzniklých problémech ztráceli a nezažívali by pocit úspěchu z vyřešení dílčích úloh.

Program je proto vhodné rozdělit na jednotlivé dílčí činnosti vedoucí k tvorbě komplexního programu. Jejich řazení jsme se pokusili sestavit podle předpokládané náročnosti a nově využívaných znalostí a dovedností.

1. Pozdrav při zapnutí
2. Štěkání pejska na povel
3. Povel sedni a lehni
4. Sleduj mě očima

5. Pojeď ke mně

Pozn.: Témata jsou uvedena jako pomůcka pro vyučujícího. V rámci řešení programu je vhodné, aby na ně, např. v rámci diskuse, přišli sami studenti. Úlohou učitele je žáky při diskusi směřovat k cíli a poskytnout jim zpětnou vazbu, např. v podobě náročnosti jednotlivých kroků.

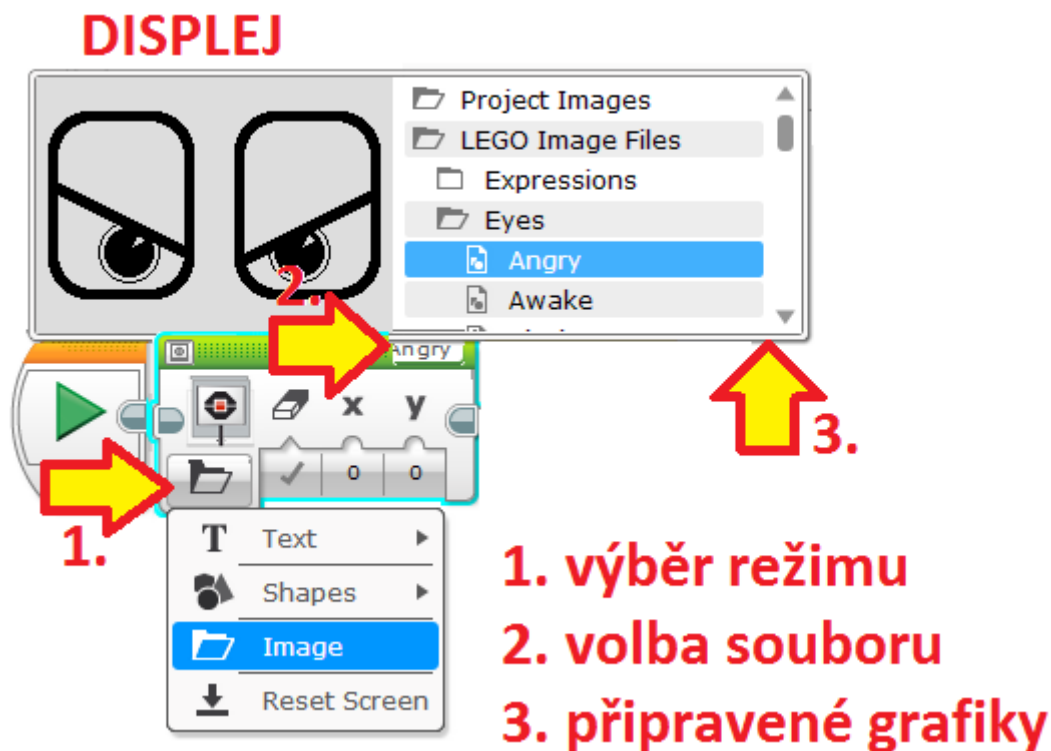
1. Pozdrav při zapnutí

Na úvod je potřeba realizovat některé z jednoduchých úloh, které povedou k odbourání prvotního strachu a zároveň zvýší motivaci studentů. Úloha by měla též představit základní zvyklosti ve tvorbě programu v daném programovacím řešení. Jak pěkné by bylo, kdyby pejsek vypadal jako pejsek a při zapnutí nás pozdravil.

V rámci úlohy proto vytvoříme první program, který pejskovi přidá oči a zašteká.

Práce s displejem

Displej pracuje v několika režimech. Pro nás je nejdůležitější režim obrázku (Image), který na celém displeji zobrazí předpřipravený obrázek. Ten je možno si vybírat z galerie prvků, kde se nachází i různé varianty očí (Eyes).



Spuštění programu a zjištění první nefunkčnosti

Pokud student takto vytvořený program spustí, zjistí, že RoboPejsek nic neudělá a program nefunguje. Není to však pravda. Program udělá přesně to, co se po něm chce. Vykreslí očička, a hned jak může, skončí. Prvním "řešením problému" je vložit za vykreslení blok čekání, druhým pak vložit prázdný cyklus.

PŘIDANÝ BLOK ČEKÁNÍ (s)



Poznámka: Před spuštěním programu je samozřejmě nutno řídicí jednotku zapnout a připojit (kabel či přes bluetooth) k počítači.

Štěkáni pejska

Řídicí jednotka EV3 je vybavena reproduktorem a umí přehrávat zvuky. Pro přehrání zvuků máme v zelené sekci Action k dispozici blok Sound s obrázkem reproduktoru. Blok se ovládá obdobně jako blok displeje s tím rozdílem, že volíme režim Play File a následně zvukový soubor z galerie.

ZVUK ŠTĚKEJ

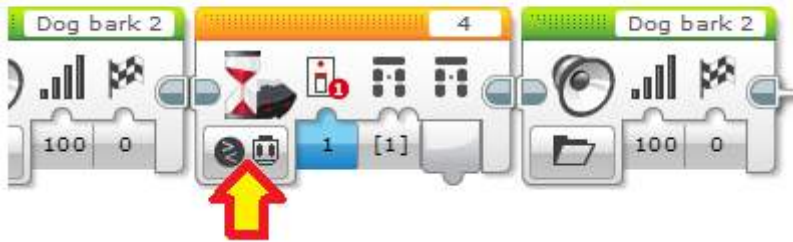


2. Štěkání pejska na povel

V dalším kroku, který je jen mírným rozšířením, můžeme chtít, aby pejsek štěkal na povel. Tím povelům může být stisk tlačítka na infračervené ovládače či řídicí jednotce. Obě dvě varianty jsou typově velmi podobné.

První řešení, na které by studenti mohli přijít, by mohlo spočívat ve změně typu čekání. Nečekalo by se na určitý čas, ale na stisk tlačítka. Poté by se přehrál zvuk.

ZMĚNA TYPU ČEKÁNÍ



Toto řešení však do budoucna není zcela vhodné, jelikož nepodporuje volbu jednotlivými tlačítky, i když studenty bychom za ně měli určitě pochválit.

3. Povel sedni a lehni

Povel sedni a lehni je vhodné vymyslet a otestovat nejprve ve zcela samostatné části programu, poté až přidat spuštění na základě stisku příslušného tlačítka.

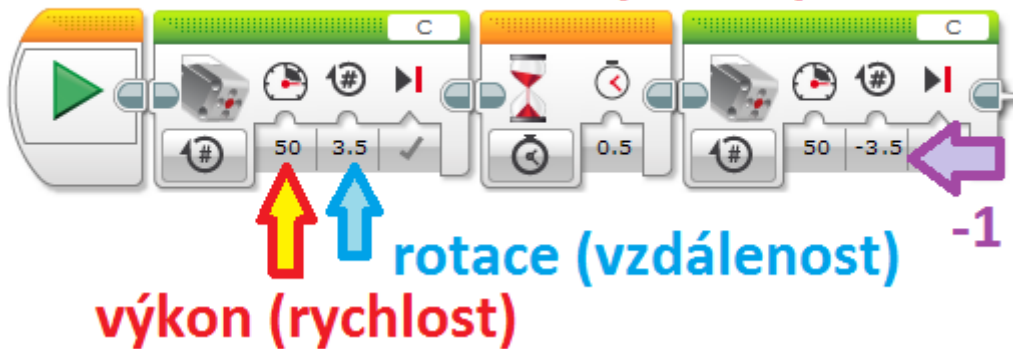
Jak povel sedni, tak i lehni, bude pracovat s natočením středního motoru upravujícího polohu nožiček vůči tělu. Správný rozsah je potřeba otestovat. Při příliš velkém rozsahu hrozí překlopení pejska a poničení modelu, při malém pak pejsek nedělá dobře, co má.

Hodnoty počtu otáček či stupňů je dobré postupně navyšovat tak, aby nedošlo k poškození konstrukce pejska. V případě problémů lze běh programu kdykoli nouzově zastavit stiskem šedého tlačítka umístěného vlevo hned pod displejem.

Pro rychlejší testování doporučujeme motor následně programově vrátit do předchozí polohy (počet otáček či výkonu vynásobíme -1).

Chceme-li zadat např. 3,5 otáčky motoru, musíme zadat číslo s desetinnou tečkou (např. 3.5). Upozorňujeme, že otáčky jsou vázány na pohyb motoru, nikoli pohyb nožiček. Ten je dán převodem.

POHYB S NÁVRATEM (SEDNI)

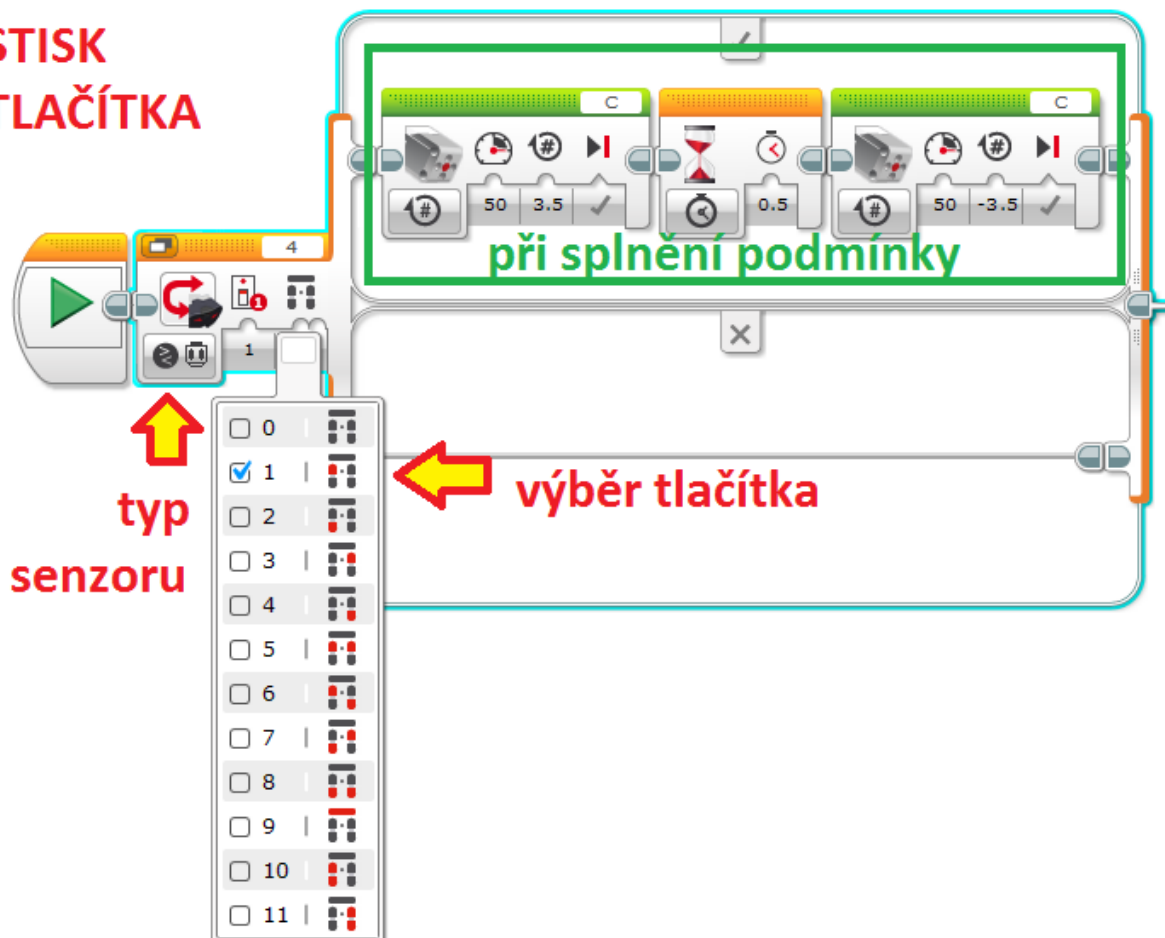


Chyby: Velmi častou chybou žáků je záměna portu. Té lze předcházet připojením robota v době návrhu (mění automaticky písmenko portu dle připojených zařízení, např. C).

Reakce na stisk tlačítka

Vytvořenou, otestovanou a funkční část kódu by bylo potřeba vykonávat jen v případě, že bylo stisknuto příslušné tlačítko. Toho lze dosáhnout větvením programu na základě podmíněného výrazu (podmínky). Bloky pro podmíněné vykonání se nacházejí v oranžové sekci prostředí Flow Control pod názvem bloku Switch.

STISK TLAČÍTKA



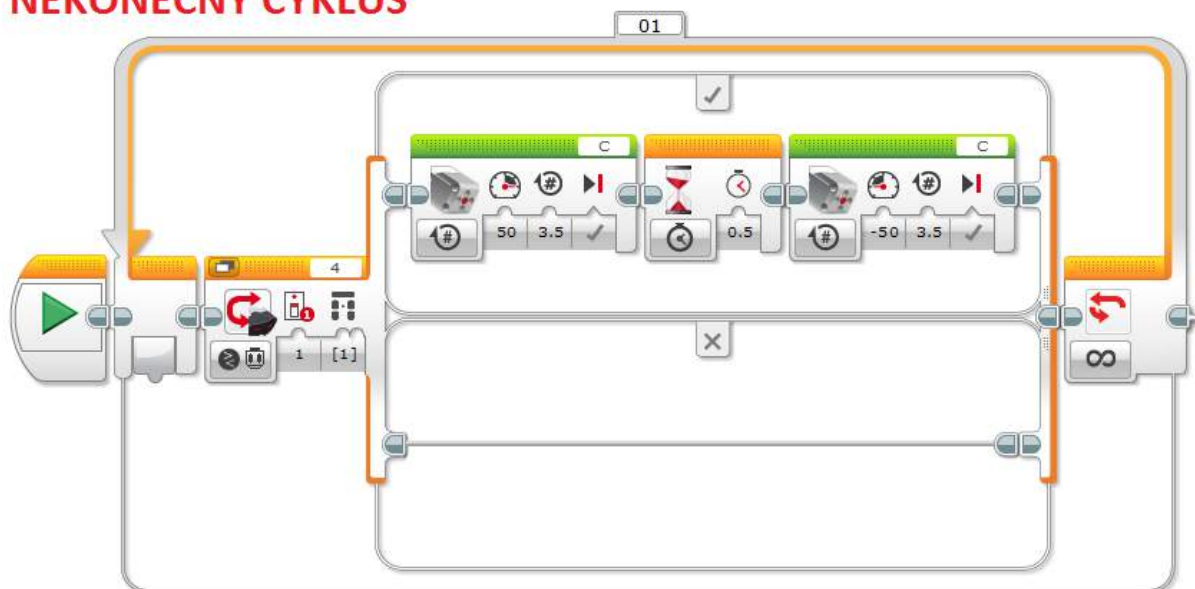
Tento blok může být řízen buďto externí podmínkou, nebo může být napojen např. na senzor a následně vyhodnocovat splnění očekávaného stavu. V našem případě se hodí blok napojit na infračervený senzor a režim porovnání - vzdálený vstup (Infrared sensor > Compare > Remote), kde si v dalším parametru zvolíme, která tlačítka budou znamenat splnění podmínky.

Pokud je takto uvedená podmínka splněna, vykonává se část s fajfkou (zaškrtnutím), pokud není, je proveden obsah v části s křížkem.

Chyba číslo dvě aneb ono mi to zase nefunguje

Pokud bychom spustili takto vytvořený program, zjistíme, že nám nebude fungovat. Je dobré se žáků zeptat, jak program probíhá a co má dělat, a případně je později upozornit na to, že program skončí (mohou vidět na displeji) a již opětovně netestuje stisk tlačítka na ovládači. Program nám nyní jen ihned po startu zkontroluje, zda senzor nezachytil signál o stisku požadovaného tlačítka, a skončí.

NEKONEČNÝ CYKLUS



Pokud chceme, aby program toto testoval opakovaně, musíme použít cyklus. V programovacím prostředí EV3 máme k dispozici pouze dva typy cyklů, s pevným počtem průchodů a s podmínkou na konci. Oba jsou reprezentovány jedním blokem Loop, v němž si volíme typ cyklu. Chybí nám cyklus s podmínkou na začátku a též cyklus s pevným počtem průchodů vyžaduje, aby proběhl alespoň jednou.

Podmínka pro ukončení cyklu může být různá, od počtu průchodů, porovnání hodnoty vstupu senzoru po časový limit. Velmi specifickým prvkem, který se v programování v rámci LEGO robotů používá velmi často, je nekonečný cyklus. Tento cyklus spolu s během celého programu přerušujeme vypnutím pomocí tlačítka umístěného vlevo pod displejem jednotky.

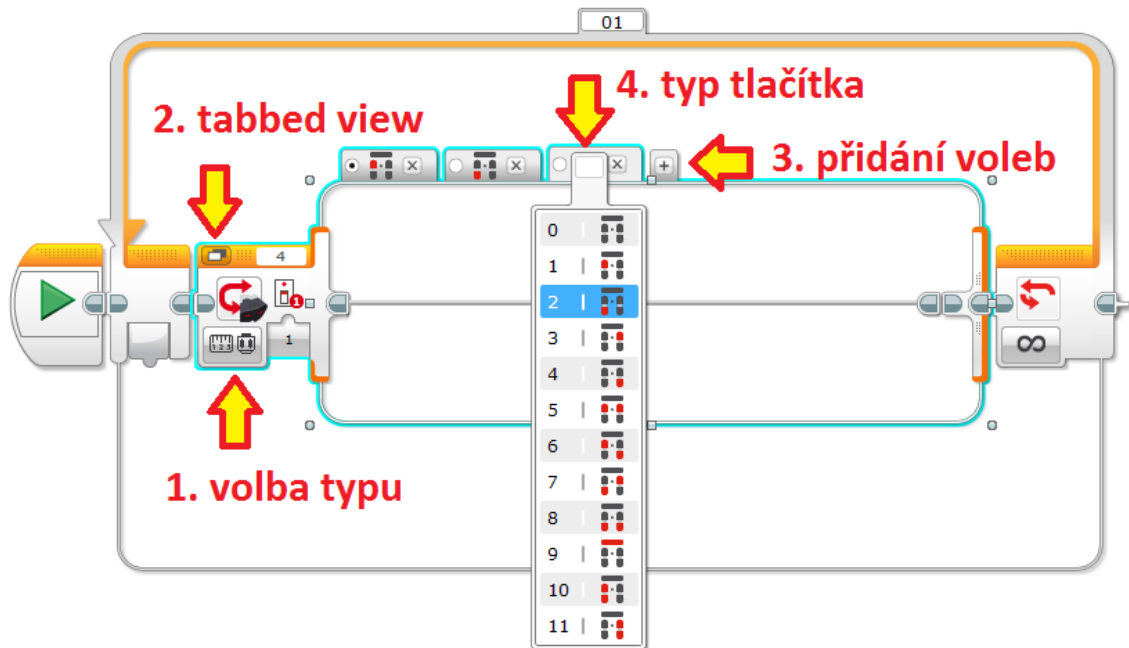
Protože chceme, aby náš RoboPejsek testoval signály z IR vysílačky opakovaně, použijeme nekonečný cyklus, do kterého zařadíme dříve vytvořené testování.

Rozlišení příkazů sedni a lehni (přidání více příkazů)

Pejsek si již na vyžádání sedne. Pokud chceme, aby si místo sednutí lehnul, stačí jen upravit směr prvotního natáčení a návratu a zjistit a nastavit míru (úhel) natočení.

Protože chceme, aby pejsek reagoval na více příkazů, musíme mu přidat další povely. Možnosti jsou následující. Buďto blok testování podmínky do cyklu umístíme ještě jednou, nebo původní přepneme do režimu měření (Infrared Sensor > Measure > Remote) a aktivujeme pohled více voleb (Tabbed View). Ten poskytuje i na místo úspornější grafické zobrazení.

REŽIM MĚŘENÍ SE ZOBRAZENÍM TABBED VIEW PRO VÍCE VOLEB

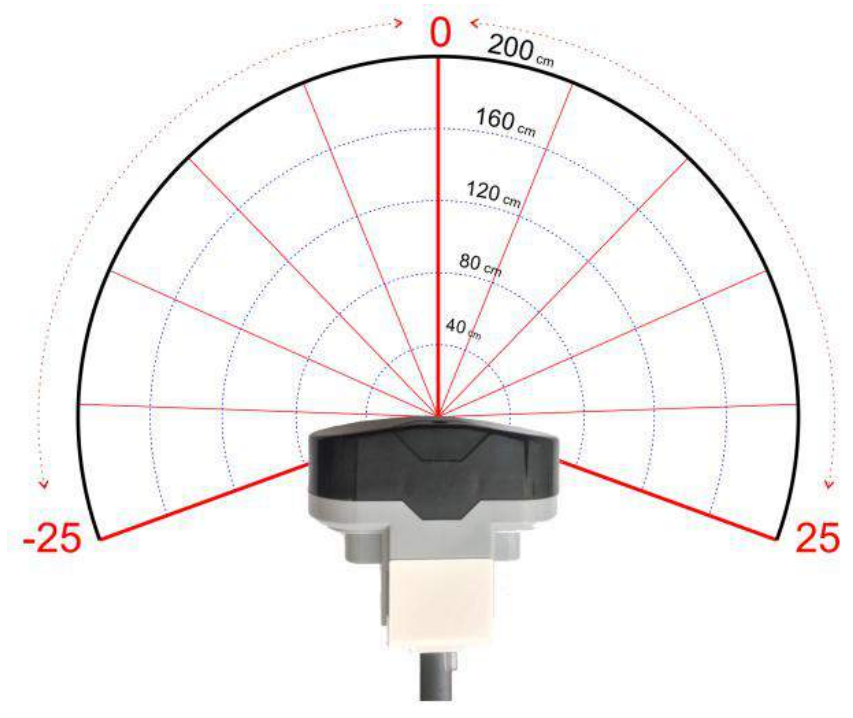


UPOZORNĚNÍ: Položka záložek označená tečkou je brána za výchozí, pokud nebude nalezena žádná z uvedených voleb tlačítek. V důsledku by to pro nás znamenalo, že by si pejsek pořád dokola sedal, i když by nebylo stisknuto žádné tlačítko. Je proto vhodné, abychom zařadili prázdnou položku nestisknuto žádné tlačítko a dali ji jako výchozí pro všechny ostatní kombinace.

4. Sleduj mě

Pro řešení pohybu RoboPejska za ovládačkou by bylo vhodné zjistit vzdálenost IR ovládačky a úhel, kam je potřeba robopejska natočit. Zjišťování bude realizováno IR senzorem. Ten funguje ve dvou režimech Proximity Mode (měření vzdálenosti optickým senzorem), který nyní nepotřebujeme, a Beacon Mode (režim navigace).

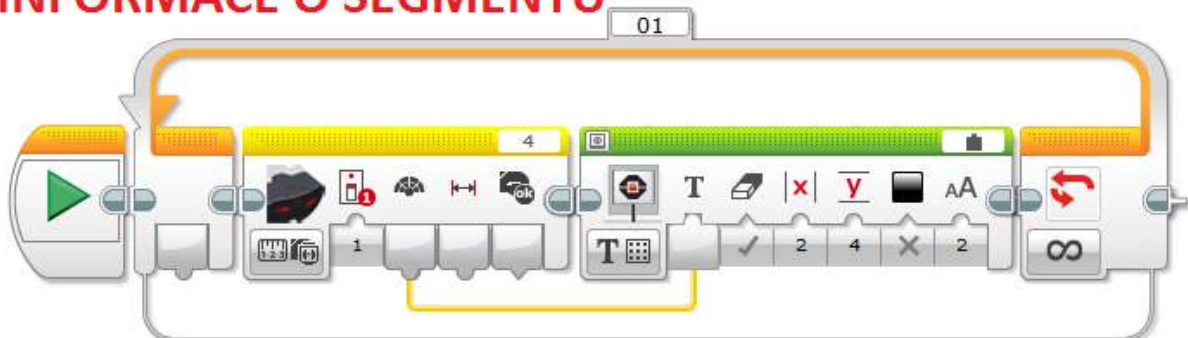
Režim navigace informuje o vzdálenosti IR vysílače od ovladače (v rozsahu 0 - 100) a o segmentu (bezrozměrné číslo -25 až 25), v němž se vysílač nachází. Pro lepší pochopení funkce doporučujeme následující obrázek. Upozorňujeme, že vracená vzdálenost senzorem není v centimetrech a vyjadřuje spíše, zda je objekt blízko či daleko.



Vypisování segmentu na displeji

Informace o segmentech jsou velmi orientační. Při praktickém použití se mnohokrát ukázala jejich nestálost a nelineárnost. Není proto vůbec od věci, když si senzorem vracené hodnoty otestujeme informativním zobrazením na displeji.

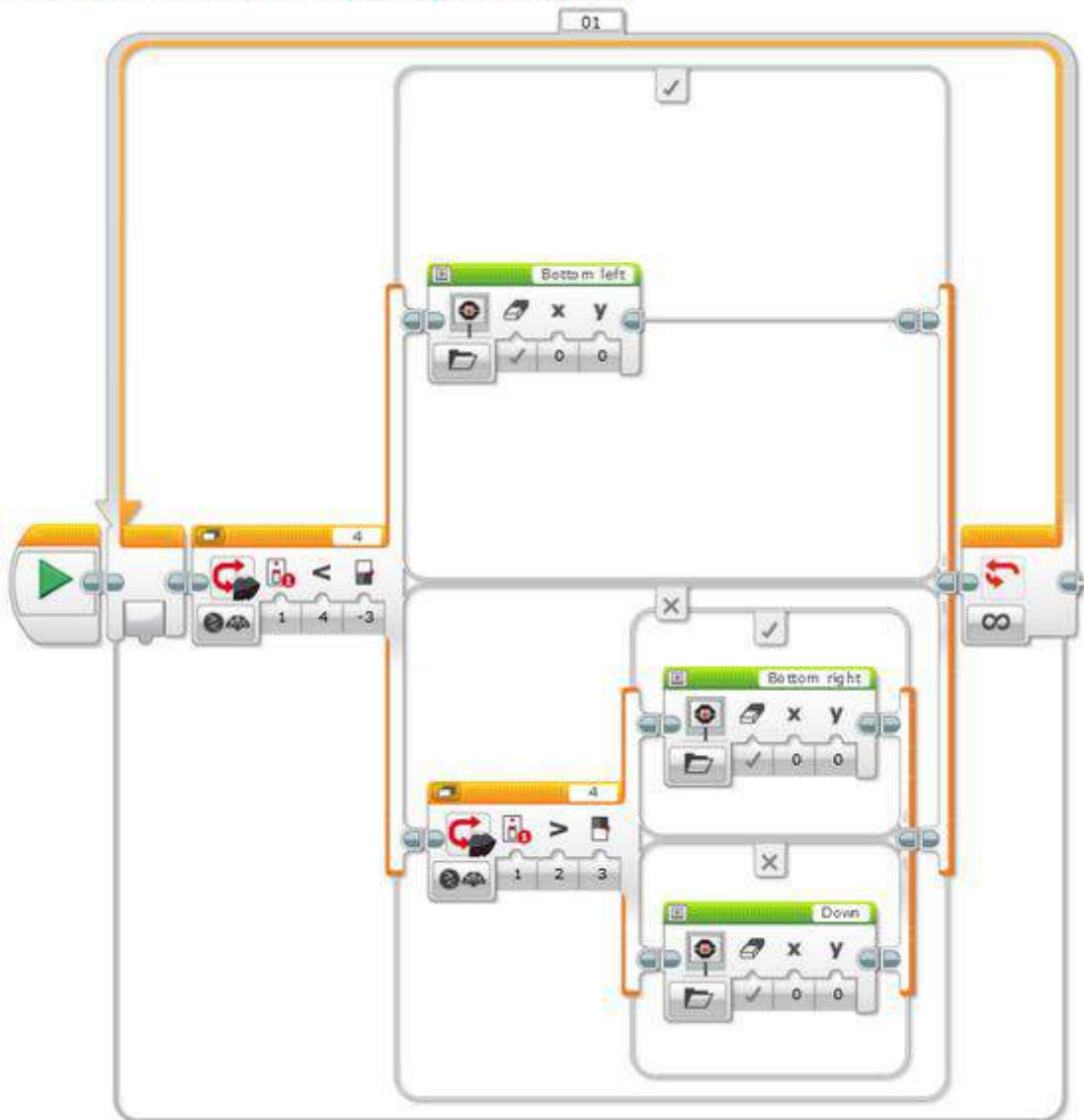
INFORMACE O SEGMENTU



Sledování segmentu očima

Práci se segmentem můžeme nacvičit ovládáním očí pejska. Ty budou ovládačku zdánlivě sledovat. Program lze realizovat kombinovanou podmínkou, kterou lze pro zmenšení prostoru přepnout do režimu Flat view.

SLEDOVÁNÍ VYSÍLAČKY OČIMA

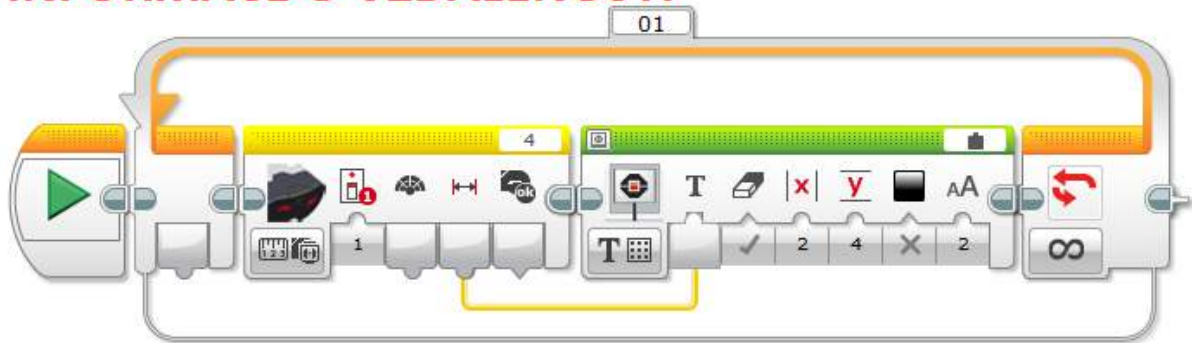


5. Pojeď ke mně

Má-li robotický pejsek jezdit za ovládačkou, potřebujeme, aby uměl zjistit nejen směr, ale i její vzdálenost. Protože se jedná o bezrozměrné číslo, je ho vhodné nejprve otestovat, obdobně jako u segmentu.

Informace o vzdálenosti

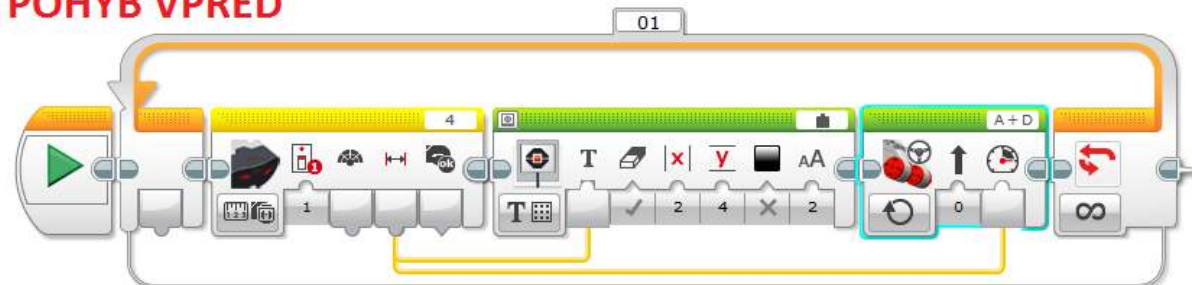
INFORMACE O VZDÁLENOSTI



Pohyb vpřed

Nyní můžeme napojit zjištěnou vzdálenost na výkon motoru. Pokud bude nulový, motor zastaví.

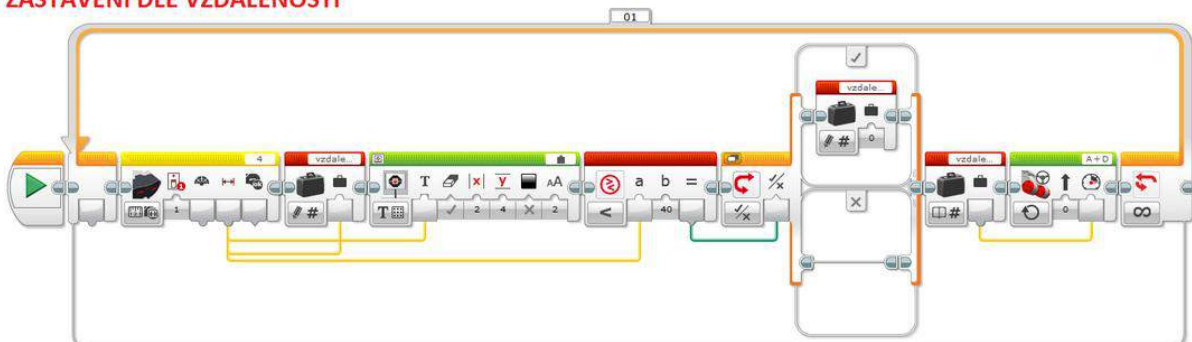
POHYB VPŘED



Zastavení dle vzdálenosti

Robotický pejsek by pro věrnější chování nemusel jezdit až k ovládačce, ale mohl by se zastavit v určité vzdálenosti od ní. Toho dosáhneme omezením výkonu. Můžeme si pomoci větvením programu i proměnnou.

ZASTAVENÍ DLE VZDÁLENOSTI

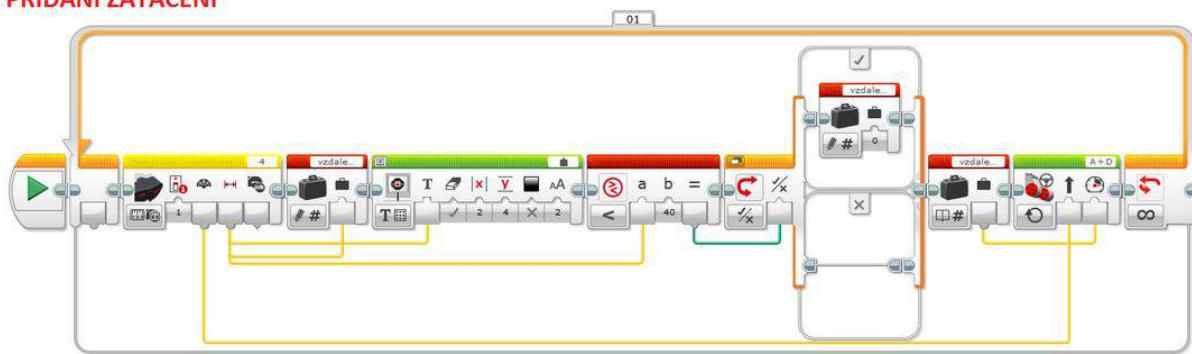


TIP: Aby se pejsek nezastavoval skokově, můžete vzdálenost upravovat změnou jejího výkonu.

Přidání segmentu

Pro zatáčení robota již v jednoduché variantě stačí přidat segment jako parametr zatáčení robota. Tento způsob řešení je závislý na pojezdu a pejsek se tak nebude otáčet, budeme-li s ovládačkou příliš blízko.

PŘIDÁNÍ ZATÁČENÍ



6. Další funkce

Rozšíření RoboPejska se meze nekladou. Lze přidat např. povely pro hlídání, aport či kombinaci naučených pohybu. Pokud by nám základní tlačítka ovládačky nepostačovala, můžeme využít tlačítkové kombinace. Nezapomeňte však, že stisk prvního tlačítka může být vyhodnocen jako stisk samostatného tlačítka, a je proto vhodné s tím počítat v programu.

Další funkcí, kterou lze robotického pejska vybavit, je detekce překážek. IR senzor v sobě obsahuje optický modul, který detekuje překážky do maximální vzdálenosti 75 cm.

Pokud nemáte IR vysílač či ovladač, můžete robotického pejska vybavit ultrazvukovým a optickým senzorem. Ovládání pak můžete provádět barvami.

