



mpc
METODICKO-PEDAGOGICKÉ CENTRUM

**VZDELÁVANÍM
PEDAGOGICKÝCH ZAMESTNANCOV
K INKLÚZII MARGINALIZOVANÝCH
RÓMSKYCH KOMUNÍT**



Moderné vzdelávanie pre vedomostnú spoločnosť / Projekt je spolufinancovaný zo zdrojov EÚ

Kód ITMS: 26130130051

číslo zmluvy: OPV/24/2011

Metodicko – pedagogické centrum

Národný projekt

**VZDELÁVANÍM PEDAGOGICKÝCH ZAMESTNANCOV
K INKLÚZII MARGINALIZOVANÝCH RÓMSKYCH KOMUNÍT**

Mgr. Petra Komárová

Digitálny svet pre 9. ročník

Vydavateľ: Metodicko-pedagogické centrum,
Ševčenkova 11, 850 01
Bratislava

Autor UZ: Mgr. Petra Komárová

Kontakt na autora UZ: Školská 9, Lovinobaňa,
petra.komarova@centrum.sk

Názov: **Digitálny svet pre 9. ročník,**

Rok vytvorenia: 2014

**Oponentský posudok
vypracoval:** RNDr. Katarína Golianová

ISBN 978-80-8052-755-6

Tento učebný zdroj bol vytvorený z prostriedkov projektu Vzdelávaním pedagogických zamestnancov k inklúzii marginalizovaných rómskych komúnít. Projekt je spolufinancovaný zo zdrojov Európskej únie.

Text neprešiel štylistickou ani grafickou úpravou.

OBSAH

ÚVOD	5
1 PRÁCA S INFORMÁCIAMI	6
1.1 Tvorba viacstránkového dokumentu	6
1.1.1 Tvorba obsahu	7
1.2 Úprava grafickej informácie	8
1.3 Digitalizácia textu	9
1.4 Tvorba fotoalbumu	10
1.5 Úprava videa	11
2 KOMUNIKÁCIA A INTERNET	12
2.1 Download súborov	12
2.1.1 Sťahovanie cez webové stránky	12
2.1.2 Sťahovanie cez P2P siete	13
2.1.3 Sťahovanie pomocou aplikácií	13
3 IKT A PROGRAMOVÉ SYSTÉMY	14
3.1 Základné nastavenie operačného systému	14
3.2 Inštalácia softvéru	15
3.2.1 Kancelársky balík	15
3.2.2 Antivírusový program	15
3.2.3 Internetový prehliadač	16
3.3 Digitálne médiá	16
4 ALGORITMY A PROGRAMOVANIE	18
4.1 Robotika	18
4.2 Robotická stavebnica	18
4.3 Programovacie prostredie Lego Mindstorms Education NXT (jazyk NXT-G)	20

4.4 Námety	21
4.4.1 Jazdi do štvorca	21
4.4.2 Pasca na myš	22
4.4.3 Morzeovka	23
4.4.4 Blikajúce svetlo	23
4.4.5 Alarm	24
4.4.6 Škatuľa na lieky s časovačom	25
4.4.7 Automatická závora	27
4.4.8 Vypíš náhodné číslo	28
4.4.9 Náhodné pohyby (robotický vysavač)	29
4.4.10 Zastaň na čiernej	31
4.4.11 Zastavenie na tretej čiare	31
4.4.12 Počítanie čiar	32
4.4.13 Počítanie stisnutí	34
4.4.14 Pohyb po čiernej čiare	35
4.4.15 Pohyb po čiernej čiare s prekážkou	37
4.4.16 Nasleduj človeka	38
4.4.17 Sumo robot	40
5 SPOLOČNOSŤ A TECHNOLOGIE	42
5.1 Odvrátená strana virtuálneho sveta	42
ZÁVER	44
ZOZNAM POUŽITEJ LITERATÚRY	45
PRÍLOHY	47

ÚVOD

Tento učebný zdroj je určený pre žiakov deviateho ročníka, ako učebný materiál na hodiny novovytvoreného predmetu Digitálny svet. Učebný zdroj je zostavený podľa osnov predmetu Digitálny svet vytvoreného na základe školského vzdelávacieho programu.

Nachádzajú sa tu rozpracované témy z okruhu práca s informáciami. V tejto kapitole sa venujeme tvorbe viacstránkového dokumentu, úprave grafickej informácie, digitalizácii textu, tvorbe fotoalbumu a úprave videa.

V ďalšej kapitole sa nachádzajú témy z okruhu komunikácia a internet. Bližšie popisujeme riziká sťahovania z internetu.

V tretej kapitole IKT a programové systémy sa venujeme operačným systémom, základnému nastaveniu, inštalácii softvéru a digitálnym médiám.

Štvrtá kapitola Algoritmy a programovanie sa venuje tematike robotov. Obsahuje popis, návrhy konštrukcií a programov, pomocou ktorých by sa mohli žiaci rýchlejšie oboznámiť a lepšie pochopiť prácu s robotmi.

Piata kapitola Spoločnosť a technológie sa zaoberá odvrátenou stranou virtuálneho sveta.

1 PRÁCA S INFORMÁCIAMI

1.1 Tvorba viacstránkového dokumentu

Pri tvorbe dokumentov treba dodržiavať isté pravidlá, ktoré sú medzinárodne dodržiavané.

Norma STN ISO 690 je medzinárodný štandard, ktorý upravuje obsah, formu a štruktúru písomných dokumentov ako aj formu bibliografických citácií a odkazov.

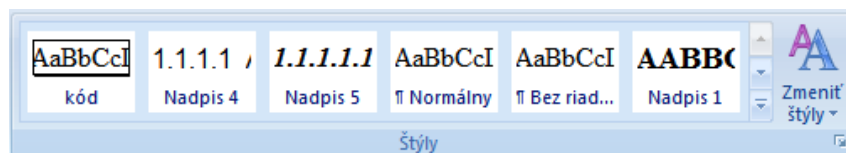
Štruktúra dokumentu obsahuje: titulný list, obsah, úvod - jadro (kapitoly) – záver, zoznam použitej literatúry, zoznam príloh a samotné prílohy.

Rozsah dokumentu sa počíta na autorské háčky. Jedna strana 30 riadkov x 60 znakov = 1800 znakov. Jeden autorský hárok je 20 strán.

Okraje na stránke sú 2,5 cm (sprava a zľava) a 3 cm (zhora a zdola).

Písmo: používame písmo typu Times New Roman (Arial), veľkosť 12 bodov a rez písma *Normálny*.

Hlavný text: priradíme mu štýl *Normálny*.



Obr. 1 Štýly

Číslovanie kapitol a podkapitol:

- Kapitoly (1 Úroveň) sa číslujú arabskými číslicami bez bodky, veľkosť písma 14 bodov, všetky písmená sú veľké a tučné. Každé kapitole priradíme štýl *Nadpis 1*. Úvod, záver a prílohy sa nečíslujú. (Nap.: 1 PRÁCA S INFORMÁCIAMI)
- Podkapitoly (2 Úroveň) sa číslujú arabskými číslicami, číslice sú oddelené bodkou, ale za posledným číslom ani za názvom bodka nie je. Veľkosť písma 12 bodov, všetky písmená sú tučné, nadpis začína veľkým písmenom. Každé podkapitole priradíme štýl *Nadpis 2*. (Nap.: 1.1 Viacstránkový dokument)

- Podkapitoly (3 Úroveň) sa používajú tie isté pravidlá, ako pri podkapitolách druhej úrovne, iba číslovanie je odlišné. Každý podkapitole 3 Úroveň priradíme štýl *Nadpis 3*. (Napr.: 1.1.1 Obsah dokumentu)

Odsek: pred odsekom 0pt, za 6pt riadkovanie je 1,5 riadka

Číslovanie strán: Strany sa čísloujú arabskými číslicami v dolnej časti strany, rovnakým písmom ako hlavný text.

Zoznam použitej literatúry: obsahuje údaje o použitej literatúre. Bibliografický údaj môže

obsahovať povinné a nepovinné údaje.

Bibliografický údaj úplný aj s nepovinnými údajmi:

[1] KALAŠ, Ivan. – WINCZER, Michal.: *Tvorivá informatika. Informatika okolo nás*. Bratislava: SPN - Mladé letá, 2007. 48. s. ISBN 978-80-10-00887-2

Bibliografický údaj s povinnými údajmi:

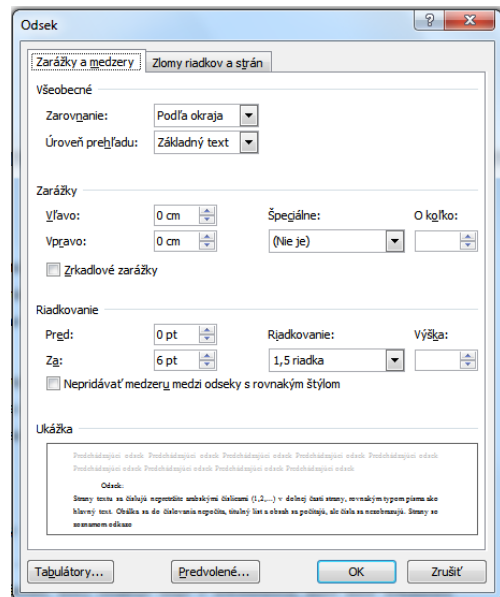
[1] KALAŠ, Ivan. – WINCZER, Michal.: *Tvorivá informatika. Informatika okolo nás*. 2007. ISBN 978-80-10-00887-2

1.1.1 Tvorba obsahu

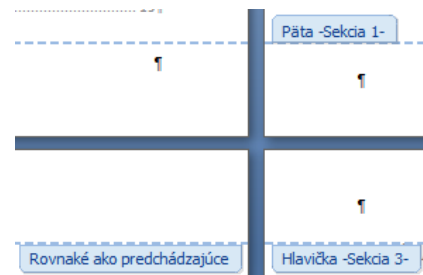
Aby sme mohli vygenerovať obsah automaticky je potrebné mať priradené štýly, a to hlavnému textu priradiť štýl Normálny, hlavným kapitolám štýl Nadpis 1 a podkapitolám postupne Nadpis 2, Nadpis 3, atď.

Ďalej si treba premyslieť, či chceme mať číslované všetky strany, alebo chceme začať iba od určitej strany. My chceme začať číslovať až od úvodu. Dokument si preto musíme rozdeliť na sekcie.

1. Presunieme sa na koniec strany pred úvodom a klikneme na *Rozloženie strany/Zlomy/Zlomy sekcií – Nasledujúca strana*. Pri dvojkliknutí na päťu stránky by sme mali vidieť niečo podobné:

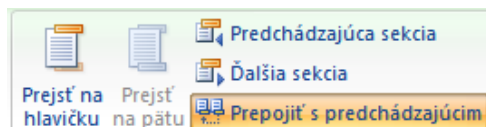


Obr. 2 Odsek



Obr. 3 Sekcie

2. Ak máme dokument rozdelený na sekcie, musíme zrušiť prepojenie s predchádzajúcou sekciou. Dvojklikneme na päťu strany a zobrazí sa panel *Navigácia*, v ktorom odklikneme *Prepojiť s predchádzajúcim*.



Obr. 4 Zrušenie prepojenia

3. Presunieme sa do sekcie (na stranu s úvodom), ktorú chceme číslovať a klikneme v paneli s nástrojmi na záložku *Vložiť/Číslo strany/Spodná časť strany* a vyberieme si, na ktorej pozícii chceme mať číslo strany.

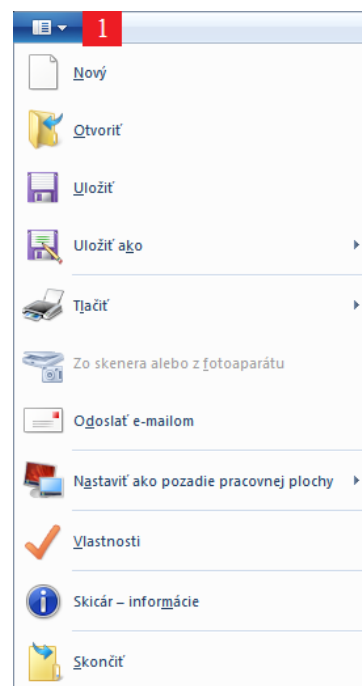
Číslovanie by malo začínať od úvodu a malo by tu byť číslo, ktoré prislúcha konkrétnej stránke.

Ak by sme chceli očíslovať celý dokument preskočíme prvý a druhý bod. Presunieme sa na hociktorú stránku a klikneme v paneli s nástrojmi na záložku *Vložiť/Číslo strany/Spodná časť strany* a vyberieme si, na ktorej pozícii chceme mať číslo strany.

1.2 Úprava grafickej informácie

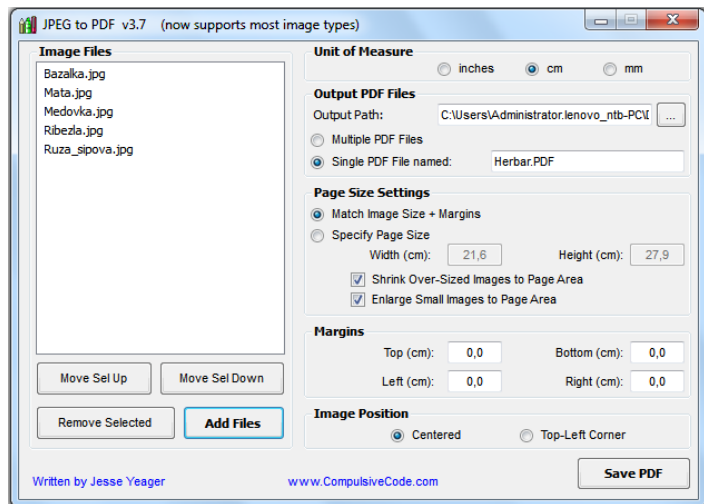
Použijeme program JPG to PDF a vytvoríme jednoduchý herbár.

1. Zozbierame obrázky, ktoré potrebujeme, uložíme ich do počítača. Potom postupne každý obrázok otvoríme v skicáry a nastavíme veľkosť papiera pre všetky obrázky. Stisneme **1**/Vlastnosti – Predvolené.
2. Keď máme nastavenú veľkosť papiera, presunieme obrázok podľa potreby, vložíme názov a nejaký popis a uložíme postupne každý obrázok.
3. Spustíme program JPEGtoPDF. Nie je potrebná inštalácia a program je free.
4. Načítame potrebné obrázky stlačením tlačidla Add Files. Zobrazí sa okno s adresármí a súbormí. Nájde požadované súbory a potvrdíme tlačidlom Otvoriť.
5. Po načítaní súborov typu .jpg nastavíme cestu pre umiestnenie vytvoreného súboru Herbar.PDF.



Obr. 5 Nastavenie papiera

6. Nastavíme vlastnosti, aké bude mať vytvorený dokument Herbar.PDF. To docielime podľa obrázka 6.



Obr. 6 Prostredie programu JPEGtoPDF

1.3 Digitalizácia textu

Digitalizácia je proces, pri ktorom prevedieme ľubovoľnú informáciu (text, obrázok, video, zvuk) do binárnej (dvojkovej) sústavy.

Pri digitalizácii textu sa používa kódovacia sada znakov ASCII tabuľka. Táto tabuľka obsahuje veľké, malé písmená anglickej abecedy, číslice, matematické znamienka, interpunkčné znamienka, špeciálne znaky a rôzne kódy. Pôvodná ASCII tabuľka bola 7-bitová, obsahovala teda 128 znakov a neskôr sa do nej pridali kódy znakov z rôznych krajín.

Optické rozpoznávanie znakov (OCR) je mechanický alebo elektronický preklad z naskenovaných obrázkov s vytlačeným textom.

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	=	=	M]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

Obr. 7 ASCII tabuľka

Textový znak po znaku je analyzovaný a prekladaný do znakového kódu ASCII. V naskenovanom obrázku sú analyzované svetlé a tmavé oblasti, ktoré sú identifikované s každým písmenom abecedy alebo číslicami. Ak je znak rozpoznávaný, bude prevedený do kódu ASCII. My na hodinách pri digitalizácii textu použijeme program ABBYY Fine Reader.

Inteligentné rozpoznávanie znakov (ICR) je moderné optické rozpoznávanie znakov rukopisu. Znaky rukopisu, ale obsahujú rôzne štýly písma, preto tento spôsob sa počas

spracovania učí lepšie, presnejšie rozpoznávať znaky. Napr.: program Neuroph OCR - Handwriting Recognition.

Optické rozpoznávanie značiek (OMR)

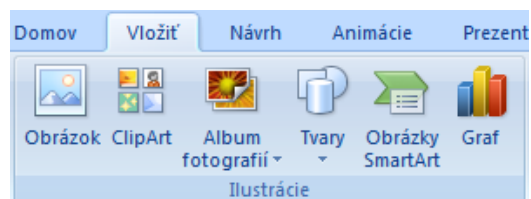
Rozpoznávanie čiarových kódov (OBR)

1.4 Tvorba fotoalbumu

Program PowerPoint je prezentačný program, ktorý umožňuje vytvoriť fotoalbum. Každá snímka môže mať iné rozloženie fotografií, môžeme sem pridávať popisy. Do albumu môžeme pridávať rôzne efekty, prechody snímok, farebné pozadia, motívy a mnoho ďalšieho. Album možno zdieľať, možno ho odoslať ako prílohu e-mailu, alebo publikovať na webe.

Fotoalbum vytvoríme tak, keď po otvorení programu PowerPoint klikneme na ikonu Microsoft/Nový. Zobrazí sa okno, kde vyberieme z ponuky *Nainštalované šablóny* a zvolíme konkrétnu šablónu *Moderný album fotografií*.

Obrázky do fotoalbumu môžeme pridávať po jednom alebo všetky naraz, a to tak, že na

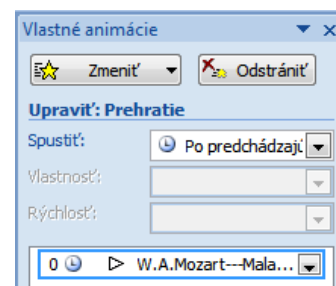


Obr. 8 Záložka vložiť

záložke Vložiť klikneme na šípku pod položkou

Album fotografií/Nový album fotografií. Klikneme na tlačidlo Súbor alebo disk, vyberieme obrázky, ktoré chceme vložiť do fotoalbumu, zadáme počet obrázkov na snímke, ich rozloženie a potvrdíme tlačidlom Vytvoriť. Toto je veľmi rýchly spôsob vytvorenia fotoalbumu, ale ja radšej volím pridávanie snímok po jednej, lebo si viem lepšie prispôbiť rozloženie jednotlivých snímok.

Pri prehrávaní fotoalbumu by nás mohla zaujímať hudba, ktorá by zážitok z pozerania fotografií umocnila. Pripravíme si vhodný hudobný súbor vo formáte wav. Premiestnime sa na prvú snímku fotoalbumu a klikneme na záložku Vložiť/Zvuk/Zo súboru. Klikneme na ikonu zvuku, ktorá sa objavila na prvej snímke. Potom na záložke Animácie/Vlastné animácie. Zobrazí sa okno s názvom Vlastné animácie. V tomto okne sa zobrazí hudobný súbor, na ktorý klikneme a zvolíme Možnosti efektu. Na karte Efekt zvolíme



Obr. 9 Vlastné animácie

Spustiť prehrávanie Od prvej a Zastaviť prehrávanie napíšeme číslo poslednej snímky a potvrdíme. Týmto zabezpečíme prehrávanie hudby počas celej prezentácie.

1.5 Úprava videa

Na úpravu a vytvorenie videa použijeme Windows Movie Maker, ktorý je k dispozícii zadarmo na stiahnutie: <http://www.windows-movie-maker.org/>. Program umožňuje strihať video, pridávať do videa špeciálne efekty, prechody, zvuk a titulky.

1. Import videa: V záložke Domov klikneme na Pridať videá a fotografie. Otvorí sa okno, v ktorom vyhľadáme video súbor, ktorý chceme upraviť a potvrdíme tlačidlom Otvoriť. Video súbor sa automaticky pridá na časovú os.
2. Úprava videa:
 - a. Strih videa: Na časovej osi sa presunieme na počiatočné miesto úseku, ktorý chceme odstrániť. V záložke Úprava klikneme na Rozdeliť. Potom sa presunieme na koncové miesto úseku, ktorý chceme odstrániť a v záložke Úprava klikneme na Rozdeliť. Na časovej osi sme rozdelili film na niekoľko častí. Teraz označíme začiatok a koniec a klikneme na tlačidlo Nástroj na vystrihnutie, čím odstránime nechcený úsek.
 - b. Pridanie titulkov: Môžeme pridať názov videa, popis (na konkrétne miesto na číselnej osi) a záverečné titulky.
 - c. Pridanie efektov a prechodov: Na časovej osi sa presuniem do bodu, kde chcem pridať animáciu, kliknem na záložku Animácie. Potom si už iba treba vybrať z rôznych efektov Prechody snímok a Posúvanie a približovanie.
 - d. Pridanie zvuku: V záložke Domov sa nachádza ikona Pridanie hudby, ktorým vieme pridať hudbu zo súboru, alebo nahrať hovorený komentár.
3. Uloženie videa: Dokončené video možno uložiť niekoľkými spôsobmi a je na každom z nás, ktorú možnosť si zvolí.

2 KOMUNIKÁCIA A INTERNET

2.1 Download súborov

Z internetu sa dá stiahnuť takmer všetko, záleží len na tom, či pri tom porušujeme zákon alebo nie. Najčastejšie je sťahovanie cez webové stránky (http protokol). Jedná sa o sťahovanie príloh v e- mailoch, ale aj o samotné načítanie nejakej webovej stránky. Z webových stránok môžeme sťahovať hry, hudbu, filmy ale aj programy.

2.1.1 Sťahovanie cez webové stránky

Upload server je server, na ktorý môžu užívatelia nahrať akýkoľvek softvér, aj nelegálny a iní užívatelia si tento softvér môžu zadarmo stiahnuť. Mená súborov na týchto serveroch neodpovedajú svojmu obsahu, aby súbory neboli zmazané kvôli nelegálnosti. Napr.: www.rapidshare.com, www.ulozto.cz, ...

Sťahovanie cez webové stránky:

- Sťahovanie z oficiálneho webu produktu. Väčšina programov a rôznych aplikácií má svoj vlastný webový priestor, kde je ich možné stiahnuť, a to buď v plnej verzii alebo v skúšobnej verzii, ktorej používanie si je potrebné neskôr zaplatiť.
- Sťahovanie z download centra. Download centrá sú servery, na ktorých je zhromaždený softvér. Softvér je v týchto download centrách triedený do kategórií. Na týchto stránkach sa nenachádza nelegálny softvér. Nachádza sa tu voľne dostupný softvér (freeware) a softvér, ktorý môžeme vyskúšať a neskôr si zakúpiť licenciu na jeho používanie. Napr.: www.stahuj.cz
- Sťahovanie cez blogy a diskusné fóra. V blogoch a diskusných fórach sa často nachádza odkaz na rôznyi softvér. Ale čo je horšie často tu autori článkov umiestňujú hudbu, filmy, softvér, ktorý je k dispozícii na stiahnutie. Pozor, to je nelegálne a porušujú tým zákon.
- Sťahovanie cez warez fóra. Warez fórum je internetová stránka, na ktorej sa nachádza množstvo odkazov na softvér (nelegálny), ktorý si je možné zadarmo stiahnuť z upload servera. Napr.: www.warforum.cz

2.1.2 Sťahovanie cez P2P siete

P2P (peer-to-peer) siete neobsahujú ani servery, ani klientov, ale iba rovnocenné sieťové uzly, ktoré plnia úlohu klienta aj servera. Tieto siete na uloženie súborov a dát nevyužívajú centrálné servery, ale používatelia siete zdieľajú časti svojich diskov pre ostatných užívateľov. Tento typ sťahovania je veľmi rýchly a spoľahlivý, ale nelegálny.

Komunikácia medzi počítačmi funguje tak, že každý počítač je zároveň serverom. Keď človek sťahuje nejaký súbor, sťahuje ho po častiach, ktoré sú skladané do jedného celku. Ale časti, ktoré už má v počítači stiahnuté, sú zároveň k dispozícii ďalším užívateľom. To znamená, kto sťahuje, zároveň aj ponúka.

Veľmi známe a obľúbené je sťahovanie cez BitTorrent. Preto si vysvetlíme základné pojmy.

Torrentový klient – je program, pomocou ktorého sťahujeme a zdieľame súbory.

Torrent – je súbor, ktorý po otvorení v klientovi začne sťahovať konkrétny súbor.

2.1.3 Sťahovanie pomocou aplikácií

Download manažér je nástroj na zvýšenie rýchlosti sťahovania dát z internetu. Vie naplánovať automatické sťahovanie a podporuje naviazanie na prerušený prenos. Podporuje množstvo populárnych webových prehliadačov, protokoly FTP, HTTP, HTTPS, sťahovanie Flash videa z portálu YouTube, Google Video alebo MySpaceTV. Napr.: Internet Download Manager

FTP klient umožní (aj hromadné) operácie so súbormi a adresármi, ako sú napr. kopírovanie, mazanie, editácia, premenovanie, uploadovanie dát na server a ich download zo serveru na svoj počítač.

Rozšírenia do webových prehliadačov sú malé programy, ktoré pridávajú nové funkcie do prehliadača ako napríklad ľahšie sťahovanie (hudby, filmov, atď.) z webu.

3 IKT A PROGRAMOVÉ SYSTÉMY

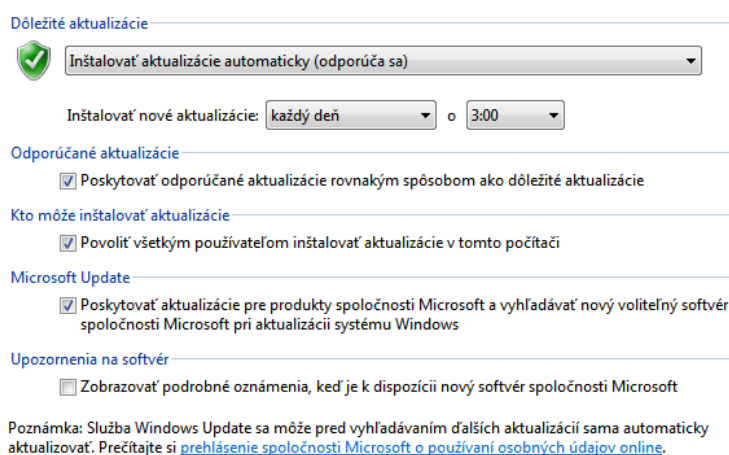
3.1 Základné nastavenie operačného systému

Brána Firewall je softvér alebo hardvér, ktorý kontroluje informácie prichádzajúce z Internetu. Podľa nastavenia brány firewall ich buď zablokuje, alebo im umožní vstup do počítača. Pomáha nám brániť sa pred hackermi alebo škodlivým softvérom. Okrem toho dokáže zastaviť odosielanie škodlivého softvéru z počítača.

Do nastavení brány Firewall sa dostaneme: *Štart/Ovládací panel/Systém a zabezpečenie/Windows Firewall*. Bránu Firewall možno vypnúť, zapnúť alebo nastaviť.

Aktualizácie OS Windows nastavíme: *Štart/Ovládací panel/Systém a zabezpečenie/Windows Update*.

Odporúčané nastavenie je prednastavené systémom, lebo je najbezpečnejšie. Dôležité aktualizácie sú inštalované hneď, keď sú k dispozícii.



Obr. 10 Zmeniť nastavenie

Nastavenie pracovnej plochy: Na pracovnej ploche klikneme pravým tlačidlom myši, aby sa rozbalila kontextová ponuka, z ktorej zvolíme Prispôbiť. Otvorí sa okno, v ktorom môžeme nastaviť pozadie pracovnej plochy, šetrič obrazovky (plán napájania) a ikony pracovnej plochy.

Nastavenie používateľských kont môžeme meniť v umiestnení: *Štart/Ovládací panel/Používateľské kontá*. Môžeme vytvoriť alebo zrušiť používateľské konto, vytvoriť heslo pre systém Windows, ale aj nastaviť rodičovskú kontrolu pre používateľské kontá.

3.2 Inštalácia softvéru

Po nainštalovaní operačného systému Windows je potrebné prispôbiť „pracovné“ prostredie operačného systému, doinštalovať potrebné programy.

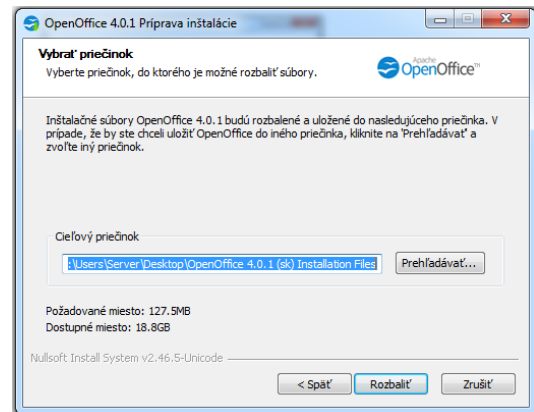
3.2.1 Kancelársky balík

Kancelársky balík OpenOffice si je možné stiahnuť na adrese: <http://www.openoffice.org/sk/download/>.

Inštaláciu spustíme dvojkliknutím myši na stiahnutý súbor:

Apache_OpenOffice_4.0.1_Win_x86_install_sk.

Najprv sa stiahnutý súbor rozbalí do umiestnenia, ktoré zvolíme pomocou tlačidla



Prehľadávať a stisneme tlačidlo Rozbaľiť.

Obr. 11 Inštalácia - rozbalenie s borov

Keď je s bor rozbaľen  treba pokra ovať stisnut m tlačidla Ďalej.

Ďalej treba postupovať podľa inštrukci . Pri inštalácii budeme vyzvan  pre zadanie mena pou ivateľa a organizacie. Nesk r si zvolíme typ inštalacie, m žeme zvoliť Typick  a potvrdiť tlačidlom Ďalej.

Po dokončení inštalacie spustíme OpenOffice dvojkliknut m na vytvoren  ikonu na pracovnej ploche. Program n s vyzve na zadanie svojho mena a priezviska. Potom sa zobraz  ponuka, z ktorej si m žeme vybrať typ dokumentu, ktor  chceme pou iavať.



Obr. 12 Ponuka OpenOffice

3.2.2 Antiv rusov  program

Antiv rusov  program Avast si je možné zadarmo stiahnuť na adrese: <http://www.avast.com/cs-cz/index>.

Inštaláciu spustíme dvojkliknut m myši na stiahnut  s bor: avast_free_antivirus_setup_online.

V ďalšom okne treba vybrať Obvyklá inštalácia (verzia Free). Budú sa zobrazovať ďalšie okná, kde treba čítať inštrukcie a potvrdzovať tlačidlom Pokračovať.

Po dokončení inštalácie treba program registrovať. Klikneme myšou na zelené tlačidlo Registrácia. Zobrazí sa ponuka, z ktorej vyberieme Free Antivirus a klikneme na tlačidlo Registrovať. Zobrazí sa registračný formulár, ktorý treba vyplniť.

3.2.3 Internetový prehliadač

Internetový prehliadač Google Chrome si možno stiahnuť z adresy: <http://www.google.com/intl/sk/chrome/>. Klikneme na modré tlačidlo Prevziať prehliadač Chrome. Potvrdiť tlačidlom Prijat' a inštalovať. Stiahol sa súbor ChromeSetup.exe, na ktorý dvojklikneme myšou. Rozbalí sa okno a spustíme inštaláciu tlačidlom Spustiť. Teraz treba už iba čakať, kedy bude prehliadač nainštalovaný.

3.3 Digitálne médiá

V minulosti sa textové a obrázkové informácie prenášali iba prostredníctvom kníh a časopisov. Ale v dnešnej dobe, plnej digitálnych technológií je to inak. S Nástupom multimédií sa zmenilo využívanie počítačov. Začala narastať popularita počítačov a tie sa začali masívne používať v domácnostiach, na školách, ale aj v takých odvetviach, o akých by sa nikomu ani nebolo nesnívalo.

Multimédiá sú informácie prezentované vo viacerých formátoch. Jedná sa o spojenie textu, obrazu, animácií, zvuku a videa za účelom sprostredkovania informácie prostredníctvom IKT. Pri ich použití musí byť používateľovi umožnené, aby sa interaktívne zúčastnil sprostredkovania.

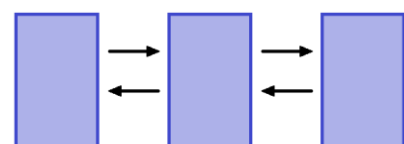
Multimédiá prezentujú informácie veľmi atraktívne, zaujímavo. Zapájajú do vnímania informácie viac zmyslov (zrak, sluch, hmat). Prinášajú názornosť, ktorá prináša lepšiu predstavivosť o prezentovanej informácii.

Médium je prostriedok, ktorým sa informácia vníma, vyjadruje, ukladá alebo prenáša.

Hypertext je nelineárne (nesekvenčné) usporiadanie textu. Jednotlivé časti sú navzájom prepojené hypertextovými väzbami (odkazmi).

Hypertextový dokument môže mať rôznu štruktúru:

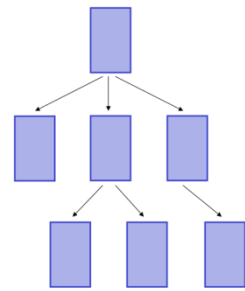
- Lineárna štruktúra – je najjednoduchšia.



Obr. 13 Lineárna štruktúra

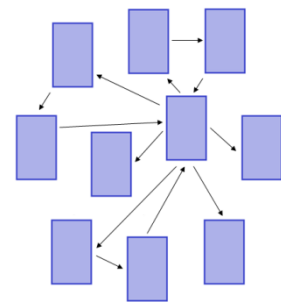
Skladá sa zo stránky, ktorá obsahuje odkaz na ďalšiu stránku a tá na ďalšiu. Ide vlastne o podobný princíp s akým sa stretneme pri listovaní knihy.

- Hierarchická štruktúra – je tvorená podstránkami s odkazmi na ďalšie stránky. Slúžia na rýchlejšie vyhľadávanie informácií.



Obr. 14 Hierarchická štruktúra

- Pavučinová štruktúra – je vlastne kombináciou predchádzajúcich dvoch štruktúr. Používa sa na vytváranie rôznych encyklopédií.



Obr. 15 Pavučinová štruktúra

Hypermédiá je to kombinácia multimédií a hypertextu.

Ľudia si často zamieňajú pojmy hypertext a hypermédiu. Pri hypertexte ide o prepájanie medzi časťami textu, ale pri hypermédiu ide o vytváranie spojení medzi textovými, grafickými (video, animácia), zvukovými informáciami.

4 ALGORITMY A PROGRAMOVANIE

4.1 Robotika

Robot je automatické zariadenie schopné reagovať na podnety okolia a na toto okolie spätne pôsobiť.

Slovo robot po prvýkrát použil český spisovateľ Karel Čapek vo svojom dramatickom diele R.U.R. (Rossum's Universal Robots).

Robotika je veda zaoberajúca sa robotmi a všetkého čo s nimi súvisí, o ich dizajne, výrobe a aplikáciách. Robotika úzko súvisí s elektronikou, mechanikou a softvérom.

Roboty sa dnes používajú najmä vo výrobe, montáži, transportoch, vesmírnom bádání, vojsku, lekárstve a bezpečnosti. S robotikou veľmi úzko súvisí aj niekoľko výskumov:

- Genetické programovanie
- Kybernetika
- Neuronové siete
- Umelá inteligencia
- Mobilná robotika

4.2 Robotická stavebnica

Robotická stavebnica Lego Mindstorms Education je nová generácia robotov od firmy Lego. Tieto robotické stavebnice je možné použiť na vyučovaní. Prvým krokom je zloženie robota a až potom môžeme začať vytvárať program, ktorý je ale potrebné prispôbiť jeho konštrukcii.

Robotická stavebnica Lego Mindstorms Educations obsahuje množstvo rôznych lego stavebných dielov, programovateľnú kocku NXT, niekoľko motorov a rôzne senzory.

Základným dielom tejto stavebnice je 32-bitový mikropočítač NXT Kocka so 4Kb Flash pamäťou a 512B pamäťou RAM. Komunikácia medzi počítačom a NXT



Obr. 16 NXT Kocka

Kockou je zabezpečená pomocou USB alebo Bluetooth pripojenia.

Na NXT Kocke sa nachádza monochromatický displej so štyrmi tlačidlami a 7 portov, ktoré slúžia na pripojenie motorov a rôznych senzorov.

NXT Kocka umožňuje tvorbu jednoduchých programov, ale toto programovanie je obmedzené iba na päť príkazov, pre zložitejšie programy je nutné použiť nejaké iné programovacie prostredie (napr.: NXC, ROBOTC, ...). My na hodinách programovania budeme používať programovacie prostredie Lego Mindstorma Education NXT (NXT - G).

Servomotor je výstupné zariadenie, umožňujúce pohyb. Má zabudovaný rotačný senzor, ktorý meria otočenie motoru v stupňoch alebo v počtoch celých otáčok. Pomocou neho môžeme regulovať rýchlosť a natočenie robota. Jedno otočenie je 360°. K jednej NXT Kocke možno pripojiť naraz tri servomotory.



Obr. 17 Servomotor

Senzory, ktoré zabezpečujú interakciu s prostredím:

- Svetelný senzor - meria úroveň prijímaného svetla v prostredí. Obsahuje aj zdroj svetla, pomocou ktorého môže zaznamenávať intenzitu odrazeného svetla, čím je možné určovať vzdialenosť, prípadne farby objektu.



Obr. 18 Svetelný senzor

- Tlakový senzor – reaguje na stlačenie tlačidla. Má priradené tri akcie:
 - pressed – zatlačenie tlačidla,
 - released – uvoľnenie tlačidla,
 - bumped – zatlačenie a uvoľnenie tlačidla.



Obr. 19 Tlakový senzor

- Ultrazvukový senzor - dokáže vyhľadať objekty, vyhýbať sa prekážkam, ale aj merať vzdialenosť a zaznamenať pohyb. Vzdialenosť meria v palcoch alebo v centimetroch s presnosťou ± 3 cm a v rozsahu od 0 do 255 cm. Senzor vyšle zvukový signál k zameriavanému objektu, ten sa vráti späť do senzora za istý časový interval, toto sa prepočíta a zistí sa vzdialenosť objektu od senzora.



Obr. 20 Ultrazvukový senzor

- Zvukový senzor – hlavnou úlohou je merať úroveň hluku v prostredí. Hluk meria v jednotkách dB (nízke aj vysoké frekvencie) a dBA (frekvencie, ktoré počuje človek). Pri čom úroveň hluku je vyjadrená v percentách (90dB = 100%).

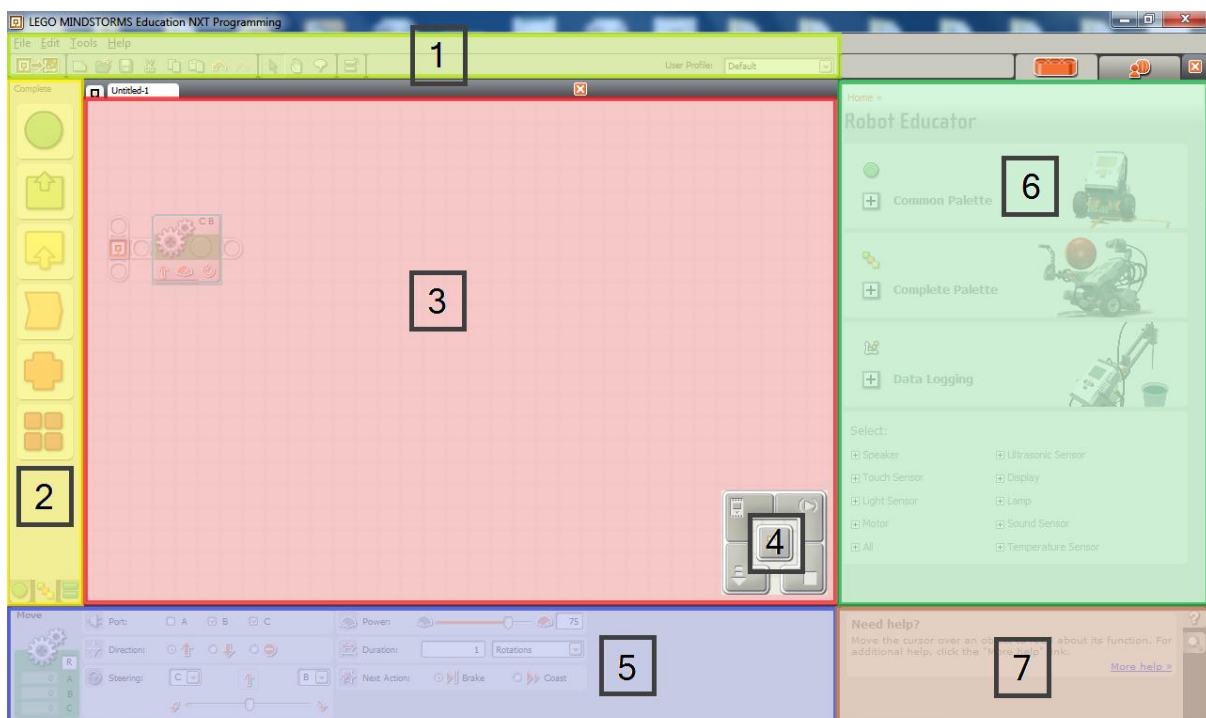


Obr. 21 Zvukový senzor

Senzory sa pripájajú na štyri vstupné porty NXT kocky a servomotory na tri výstupné porty A – C.

4.3 Programovacie prostredie Lego Mindstorms Education NXT (jazyk NXT-G)

Programovacie prostredie Lego Mindstorms Education vytvorila firma National Instruments v spolupráci s firmou Lego. V tomto prostredí sa využíva programovací jazyk NXT-G. Písmeno G je odvodené zo slova grafický. Tento programovací jazyk využíva pri programovaní skladanie programov pomocou programovacích blokov, ktoré sú navzájom prepájané. Je to jednoduchý programovací jazyk, s ktorým dokážu pracovať aj začiatočníci, ktorí nemajú žiadne skúsenosti s programovaním.



Obr. 22 Prostredie Lego Mindstorms Education NXT

- 1 Hlavné menu a nástrojová lišta – slúži na prácu so súborom.
- 2 Programovacie palety:

- a) Common palette – obsahuje najpoužívanejšie programovacie bloky,
 - b) Complete palette – obsahuje všetky programovacie bloky,
 - c) Custom palette – obsahuje vytvorené bloky a stiahnuté z internetu.
- 3 Pracovná (programovacia) plocha – určená na skladanie programov pomocou programovacích blokov.
 - 4 Controller – obsahuje tlačidlá pre export programov do NXT kocky a pre správu pamäti.
 - 5 Konfiguračný panel – slúži na nastavenie parametrov konkrétneho programovacieho bloku.
 - 6 Lego educator a My portal – programovacia a konštrukčná príručka. Obsahuje lekcie pre výučbu programovania v programovacom jazyku NXT – G.
 - 7 Okno s pomocnou nápovedou a mapa pracovnej plochy – klasická forma nápovedy.

4.4 Námety

V tejto časti sa pokúsime vypracovať niekoľko programov na lepšie pochopenie vytvárania programov v programovacom prostredí Lego Mindstorms Education NXT.

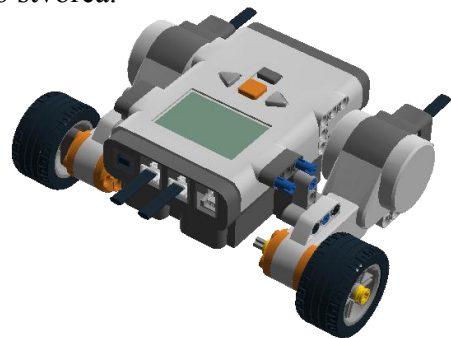
Na CD sa nachádzajú modely stavieb, aj programy, potrebné k naprogramovaniu týchto stavieb. V priečinku Lego modely sa nachádzajú modely jednotlivých stavieb, spolu s návodom, ako ich skonštruovať. Tieto modely aj postupy konštrukcií sú vytvorené v programe LEGO Digital Designer, ktorý je k dispozícii na stiahnutie: <http://ldd.lego.com/en-us/download/>. V priečinku Lego programy sa nachádzajú programy k jednotlivým konštrukciám. Programy sú vytvorené v programovacom prostredí Lego Mindstorms Education NXT-G.

4.4.1 Jazdi do štvorca

Jednoduchý program, v ktorom bude robot jazdiť do štvorca.

Stavba: NXT kocka, 2 servomotory

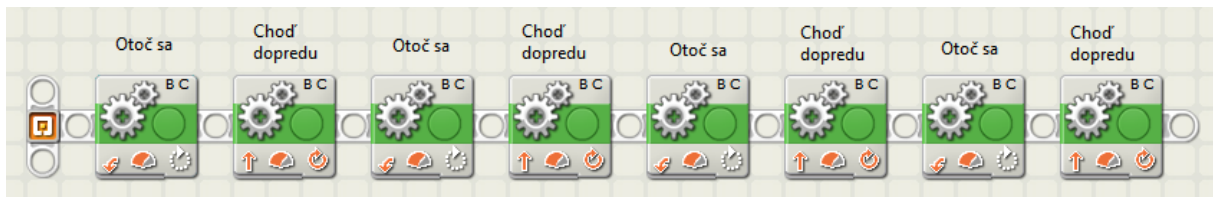
Jednoduchý pohybujúci sa robot na kolieskach, ktoré sú poháňané motormi. Robot sa vie otáčať a ísť rovno.



Obr. 23 Jednoduchý robot bez senzorov

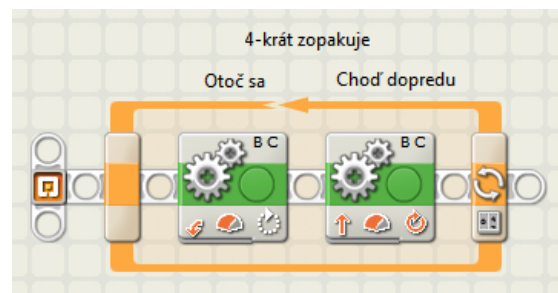
Program:

Najprv sa pokúsime vytvoriť program pomocou jednoduchého poskladania príkazov.



Obr. 24 Štvorec

Takéto skladanie príkazov je dosť zdĺhavé. Pozorný žiak si všimne, že sa v programe opakujú dva príkazy štyrikrát. Preto použijeme blok Loop (cyklus) z palety Common. Aby sa príkazy zopakovali štyrikrát, musíme tomuto cyklu nastaviť Control: Count (trvanie), Count: 4.



Obr. 25 Štvorec - cyklus

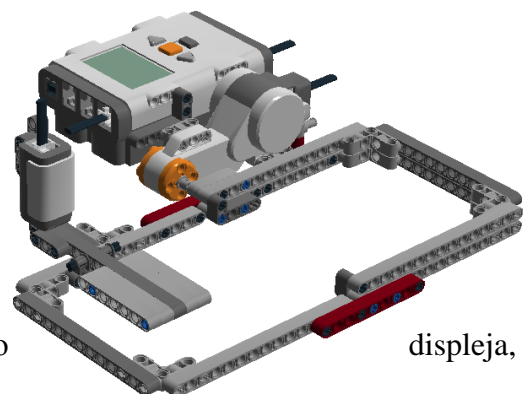
4.4.2 Pasca na myš

Pokúsime sa vytvoriť funkčnú pascu na myš.

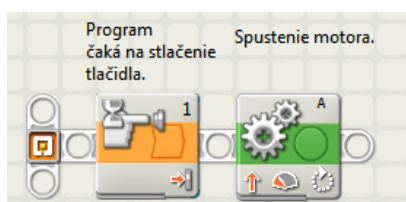
Stavba: NXT kocka, servomotor, dotykový senzor

Program:

Program čaká na príchod myši, ktorá vojde na plošinku v pasci, čím zatlačí tlačidlo dotykového



displeja,

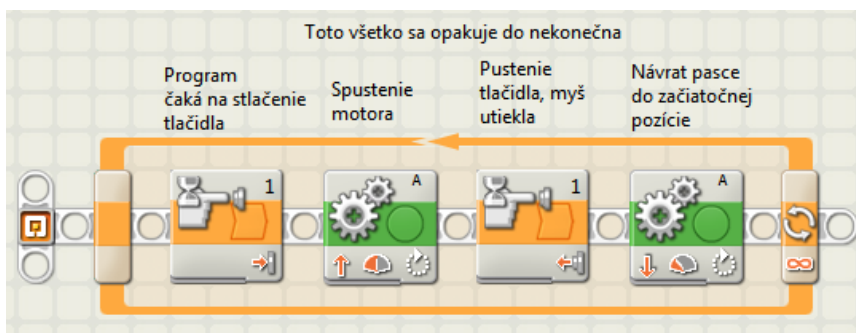


Obr. 27 Jednoduchá pasca

na základe čoho sa zaklapne pasca.

Obr. 26 Pasca na myš

Program by sme mohli vylepšiť. Myš by mohla z pasce aj uniknúť, preto by sme mohli do programu doplniť ďalšie bloky. Po utečení myši sa tlačidlo dotykového senzora vráti do pôvodnej pozície, čím sa pasca nastaví na ďalšie zaklapnutie myši. Všetky tieto príkazy (bloky) sme dali do cyklu (Loop z palety Common). Cyklu nastavíme Control: Forever, čím zabezpečíme, že sa to bude opakovať do nekonečna, až kým pascu nevypneme.



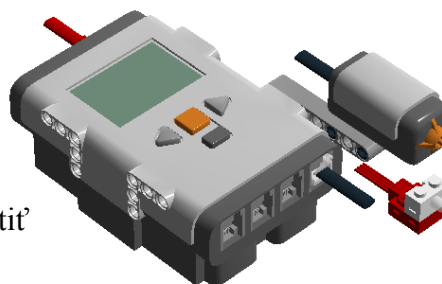
Obr. 28 Zložitejšia pasca

4.4.3 Morzeovka

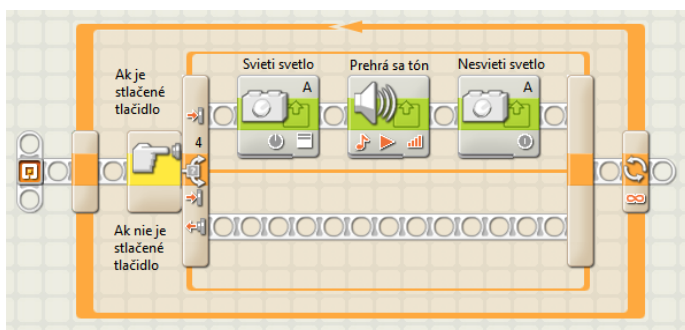
Jednoduchý program, ktorý napodobňuje morzeovku. Pri stlačení tlačidla svieti svetlo a prehrá sa hudobný tón, tak dlho, kým je zatlačené tlačidlo dotykového senzora.

Stavba: NXT kocka, dotykový senzor, lego kocka s diódou

Program: Využijeme blok Switch (vetvenie podľa dotykového senzora). Keď zatlačíme tlačidlo začne svietiť



Obr. 29 Konštrukcia morzeovky



Obr. 30 Morzeovka

svetlo diódy, prehrá sa tón a svetlo diódy vypne. Keď nie je tlačidlo dotykového senzora zatlačené, nerobí

sa nič. Aby program reagoval na ďalšie stlačenie a pridržanie tlačidla dotykového senzora, celé to dáme do nekonečného cyklu.

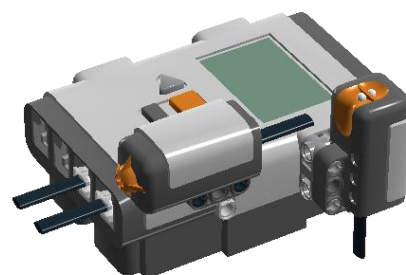
4.4.4 Blikajúce svetlo

Jednoduchý program, v ktorom bude blikáť svetelný senzor a prestane, až po zatlačení tlačidla na dotykovom senzore.

Stavba: NXT kocka, svetelný senzor, dotykový senzor

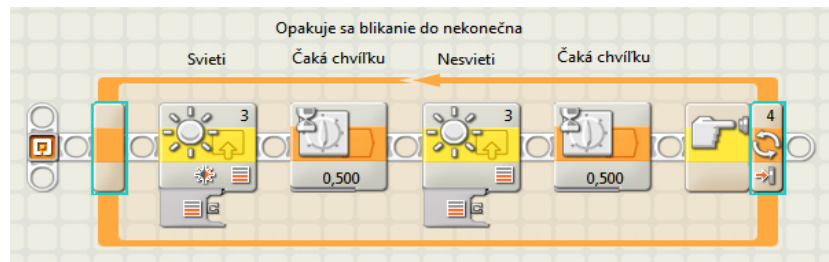
Program: V programe použijeme cyklus, ktorý sa bude opakovať až pokiaľ nebude zatlačené tlačidlo na

dotykovom senzore. Do cyklu vložíme dva bloky Svetelný senzor. Prvému bloku zaškrtneme Generate light (svieti) a tomu druhému odškrtneme



Obr. 31 Konštrukcia s blikaním

Generate light (nesvieti).
 V cykle sa bude opakovať
 svietenie senzora
 s nesvietením senzora. Aby

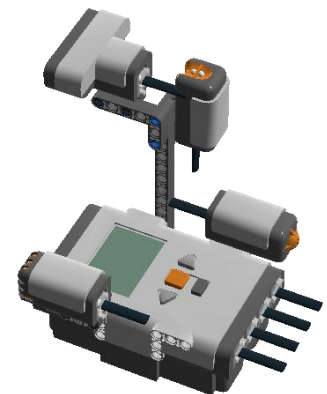


Obr. 32 Blikajúce svetlo

toto bolo funkčné, vložíme
 do cyklu ešte dva čakacie bloky Wait, ktorým nastavíme Control: Time (čas) a Seconds: 0,5.

4.4.5 Alarm

Zostrojíme jednoduchý alarm, ktorý bude reagovať na otvorenie dverí a rozbitie okna tak, že začne blikať červené svetlo a začne vydávať zvuk. Toto sa bude opakovať až do vypnutia alarmu (kým zatlačíme tlačidlo dotykového senzora). Keď sa nič nedeje, svieti červené svetlo neprerušene, až do aktivovania alarmu zlodejom.



Obr. 33 Alarm

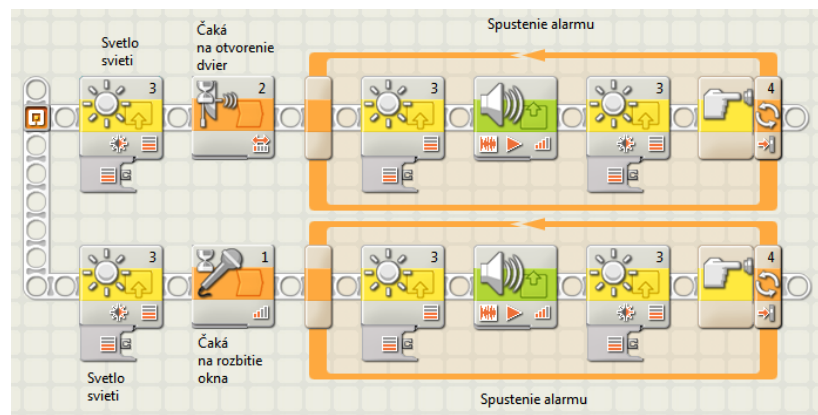
Stavba: NXT kocka, dotykový senzor, ultrasonický senzor, zvukový senzor a svetelný senzor

Program: V programe využijeme paralelné procesy. Súčasne budeme zisťovať, či sa dvere otvorili (ultrasonický senzor) a či zlodej nerozbil okno (zvukový senzor).

Ako zistíme, že sú dvere otvorené? Zistíme to tak, že postaveného robota (alarm) postavíme do určitej vzdialenosti od dvier. S robotom už nebudeme hýbať. Odmeriame vzdialenosť robota od zatvorených dvier, a to tak, že na NXT kocke v menu zvolíme View a vyberieme senzor Ultrasonic cm. Zvolíme port 2. Senzor odmeria vzdialenosť (80cm) od zatvorených dvier. Teraz už zostáva len zistiť, či sa dvere otvárajú smerom k robotovi alebo od robota.

Nám sa dvere otvárajú k robotovi, tak nastavíme ultrasonickému senzoru Distance < 75 centimeters.

Ďalej treba zistiť úroveň hluku a ticha. To zistíme na NXT kocke menu zvolíme View a vyberieme možnosť Sound dB. Podľa toho



Obr. 34 Alarm so zastavením

nastavíme úroveň zvuku, podľa ktorej alarm zistí, že bolo rozbité okno. My sme nastavili zvukovému senzoru Sound > 50.

4.4.6 Škatuľa na lieky s časovačom

Škatuľa s liekmi, ktorá má zabudovaný časovač. Vhodná pomôcka pre starších ľudí, ktorým už pamäť neslúži až tak dobre. Škatuľa je uzamknutá, nedá sa otvoriť, iba v určitých okamih sa odomkne, kedy ju môžeme otvoriť.

Do vytvorenej škatule z preglejky, ktorá sa zvrchu otvára, sme vmontovali lego súčiastky.

Stavba: škatuľa z preglejky, NXT kocka, servomotor, dotykový senzor, svetelný senzor

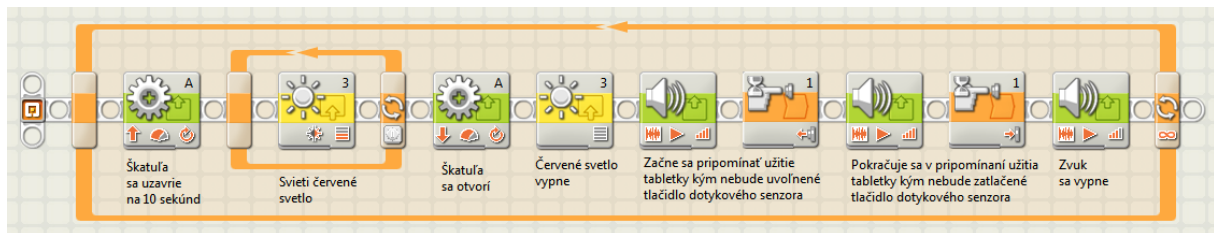
Program:

Na uzamknutie a odomknutie škatule používame servomotor a dotykový senzor. Svetelný senzor použijeme na



Obr 35 Škatuľa na lieky

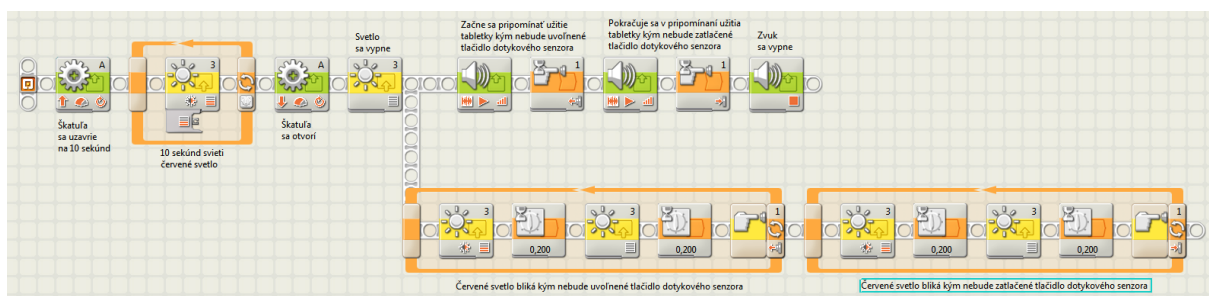
signalizáciu uzamknutia škatule. Keď je škatuľa uzamknutá, svieti červené svetlo. Keď sa škatuľa odokne, zo škatule sa začne šíriť zvuk, ktorý pripomína,




Obr. 36 Lieková škatuľa - jednoduchšia verzia

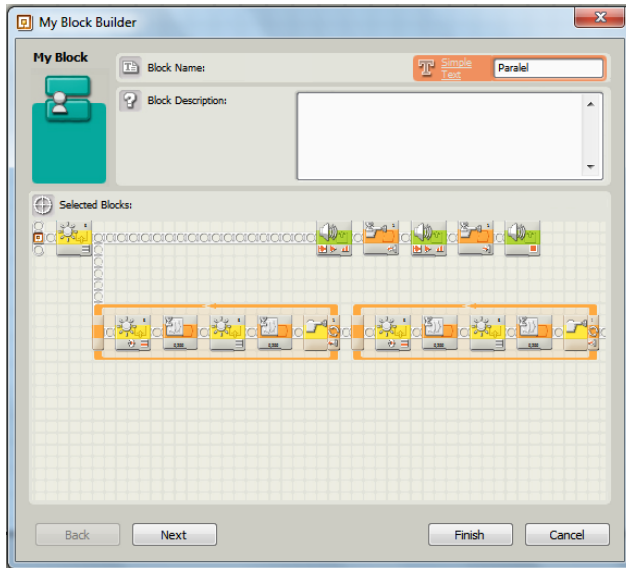
aby si človek zobral tabletku a zároveň sa vypne červené svetlo. Škatuľa prestane vydávať zvuk až po zavretí vrchného krytu, keď sa zatlačí tlačidlo dotykového senzora.

My by sme ale chceli, aby počas odomknutia škatule blikalo červené svetlo zo svetelného senzora. Budeme preto potrebovať paralelný proces.



Obr. 37 Lieková škatuľa - zložitejšia verzia bez cyklu

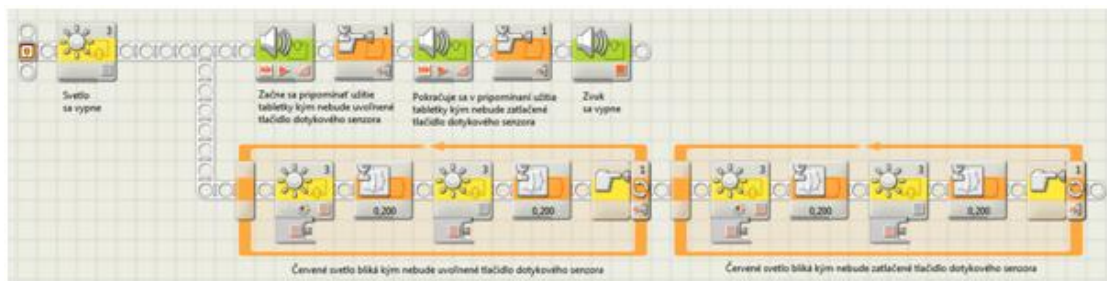
Program by bol celkom dokončený, ale treba, aby sa celý program opakoval. Potrebujeme ho zacykliť. Ale tu nastáva problém s paralelným procesom, preto vytvoríme Nový blok (podprogram). Časť programu, ktorú chceme umiestniť do Nového bloku označíme pomocou myši a klikneme na tlačidlo .



Nový blok treba pomenovať a kliknúť na tlačidlo Finish. Ak by sme chceli zmeniť obrázok na ikone Môj blok, treba kliknúť na tlačidlo Next. Zobrazí sa ďalšie okno, v ktorom si vybratý obrázok pretiahneme myšou do prázdneho poľa a zároveň sa zorazí náhľad ikony. Potvrdíme tlačidlom Finish.

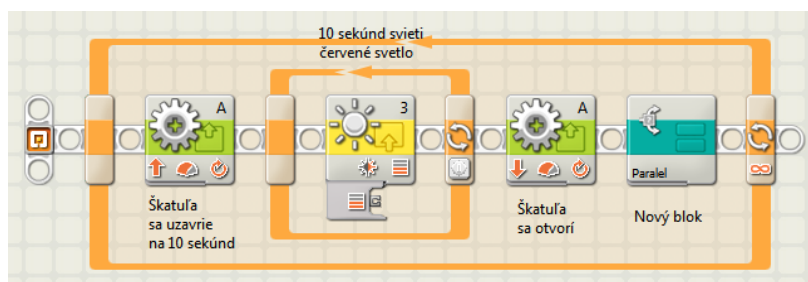
Obr. 38 Uloženie nového bloku

Nový blok s názvom Paralel:



Obr. 39 Nový blok - Paralel

Z programu sa presunul paralelný proces s potrebnými blokmi do Nového bloku s názvom Paralel. Za zostávajúce bloky pridáme novovytvorený blok Paralel. Všetky tieto bloky vložíme do nekonečného cyklu. Program je hotový, už ho iba stačí stiahnuť do NXT Kocky.



Obr. 40 Lieková škatuľa - zložitejšia verzia

Posledným problémom v tomto programe zostáva otázka: Ako vytvoriť vlastné zvuky?

Ak by sme chceli dostať vlastný zvuk do robota, musíme na to použiť niekoľko programov, lebo robot rozozná iba zvuk vo formáte *.rso.

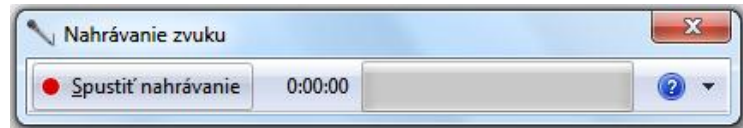
Najprv si nahráme zvuk cez Nahrávanie zvuku. Je to program, ktorý je súčasťou operačného systému Windows.

Program sa nachádza:

Štart/Všetky programy/

Príslušenstvo/ Nahrávanie zvuku.

Zvuk si uložíme na pracovnú plochu, priradíme mu názov. Napr.: Halo.wma.



Obr. 41 Nahrávanie zvuku

Ale formát .wma musíme prerobiť na formát .wav. To by sa nám mohlo podariť pomocou online konvertora, napr.: <http://www.online-convert.com/>. Teraz máme náš zvuk uložený ako Halo.wav.

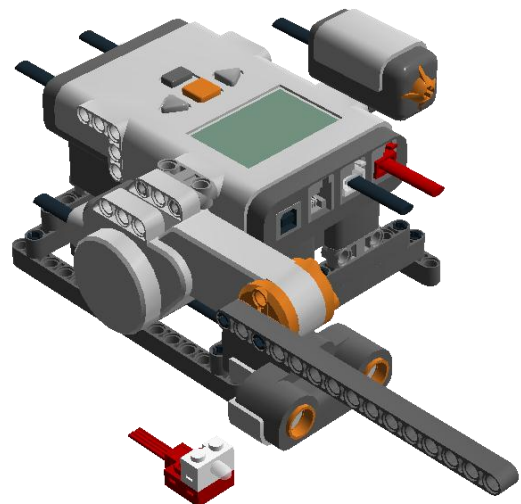
Teraz už stačí použiť program wav2rso.exe, ktorý prerobí zvukový súbor Halo.wav na Halo.rso a tento súbor už stačí prekopírovať na miesto, kde sa nachádzajú aj ostatné zvukové súbory: C:\Program Files\LEGO Software\LEGO MINDSTORMS Edu NXT\engine\Sounds. Zvukový súbor je pripravený na použitie v programe, teraz sa už zobrazí aj v ponuke Sound file v programovacom prostredí NXT edu.

4.4.7 Automatická závora

Vytvoríme jednoduchú závora, ktoré sa dnes používajú na parkovisku. Pri závore je snímač, ktorý sníma, či sa pod zavorou náhodou nenachádza auto, aby sa závora v tomto momente nezavrela.

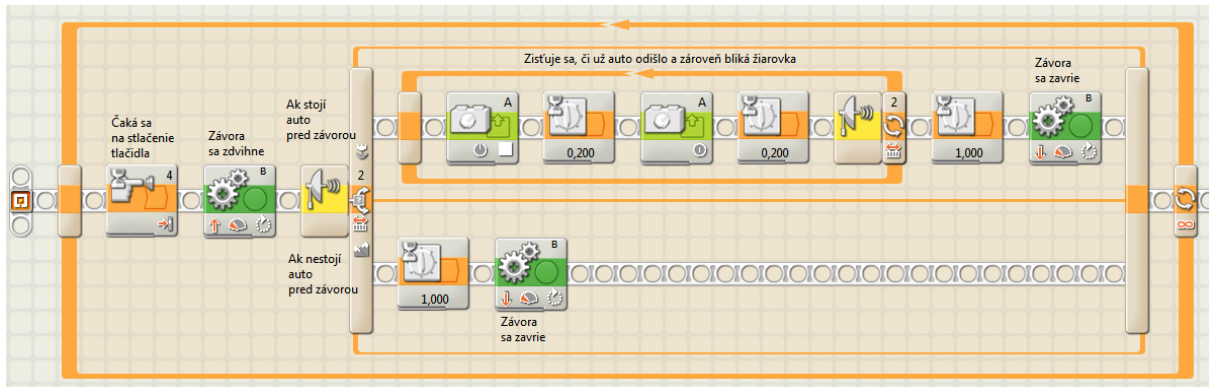
Stavba: NXT kocka, servomotor, ultrasonický senzor, dotykový senzor, lampa

Program: Závora sa otvorí po stisnutí tlačidla dotykového senzora. Ak sa pred zavorou nenachádza auto, alebo nejaká prekážka (ultrasonický senzor nezameria predmet), závora



Obr. 42 Automatická závora - konštrukcia

sa zavrie pomocou pripojeného servomotora. Ak by ale pred závorou stálo auto, alebo nejaká prekážka (ultrasonický senzor zameria predmet), závora sa nezavrie kým auto neodíde (ultrasonický senzor nezaznamená žiadny predmet pred závorou). Pokiaľ je závora otvorená bliká svetlo.



Obr. 43 Automatická závora - program

4.4.8 Vypíš náhodné číslo

Vytvoríme program, ktorý bude generovať náhodné číslo, ktoré sa zobrazí na displeji.

Stavba: NXT kocka

Program: Aby sme vygenerovali náhodné číslo, použijeme blok Random z palety Data. Nastavíme mu interval (Minimum a Maximum), v ktorom sa bude číslo nachádzať.

Aby sa číslo zobrazilo na displeji, použijeme blok Number to Text z palety Advanced. Tento blok prevedie číslo na text, lebo na displeji sa zobrazí iba text alebo obrázok.

Ďalším blokom bude blok Display z palety Action. Musíme nastaviť, čo sa má zobraziť na displeji (text) a kde (pozícia). Pozícia by mohla byť x: 32, y: 31 a riadok (line): 5.

V tomto programe sme použili špeciálne drôty, ktoré prenášajú informáciu (dáta) medzi jednotlivými blokmi.



Obr. 44 Náhodné číslo

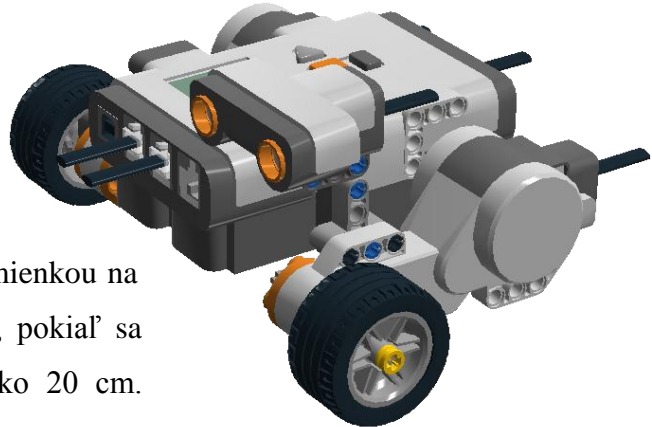
Tieto drôty nemôžeme spájať hociako, ako sa nám zachce. Môžeme prepájať číselné údaje s číselnými údajmi a textové údaje s textovými údajmi.

Aby sme číslo na displeji videli, do programu pridáme čakací blok Wait z palety Flow, ktorému nastavíme Time (čas) a Seconds: 3.

4.4.9 Náhodné pohyby (robotický vysavač)

V tomto programe sa pokúsime pozrieť na to, ako by mohol fungovať robotický vysavač, ktorý sa zapne v izbe a on sa pohybuje náhodnými pohybmi po celej miestnosti, pričom vysáva podlahu.

Stavba: NXT kocka, 2 servomotory, ultrasonický senzor



Obr. 45 Robot

Program: Najprv vytvoríme cyklus, s podmienkou na konci. Robot sa bude pohybovať dopredu, pokiaľ sa nepriblíži k nejakej prekážke na menej ako 20 cm.

Keď sa priblíži na menej ako 20 cm, kúsok cúvne (Degrees: 50).

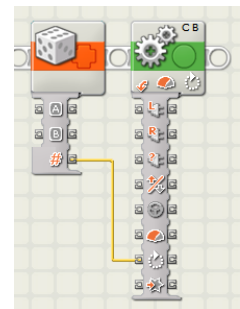
Ale my by sme chceli docieľiť, aby sa otáčal o náhodnú hodnotu. Použijeme blok Random z palety Data a nastavíme interval Minimum: 1 a Maximum: 360.



Obr. 46 Chod' dopredu, ale nenaraz

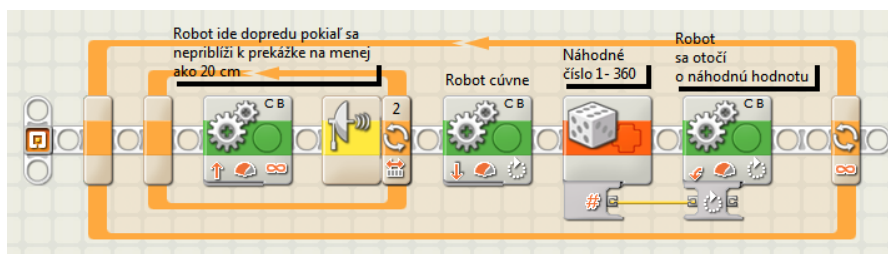
Bloku Move (motory) z palety Common nastavíme otáčanie na jednom mieste.

Ďalej musíme zabezpečiť veľkosť otočenia motora, ktoré má byť závislé od náhodného čísla, a to dosiahneme prepojavacími drôťmi medzi týmito dvoma blokmi.



Obr. 47 Náhodné otáčanie

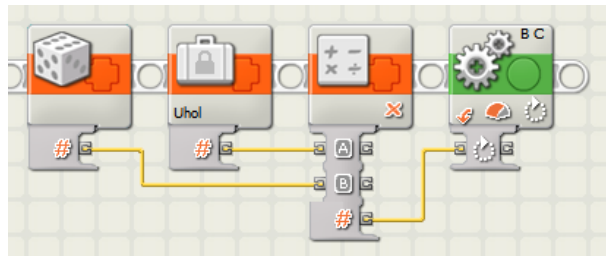
Celý poskladaný program by mohol vyzerat' takto:



Obr. 48 Robot sa otočí o náhodný uhol

Predchádzajúci program by sme mohli ešte vylepšiť tak, aby sa robot otáčal iba o 90° , 180° , 270° alebo 360° . Aby sme to dosiahli, musíme preprogramovať druhú časť programu.

Program by mal fungovať tak, že vygeneruje náhodné číslo z intervalu 1 – 4, ktoré vynásobí s konštantou $Uhol = 90^\circ$ a výsledok sa priradí otáčkam motorov.

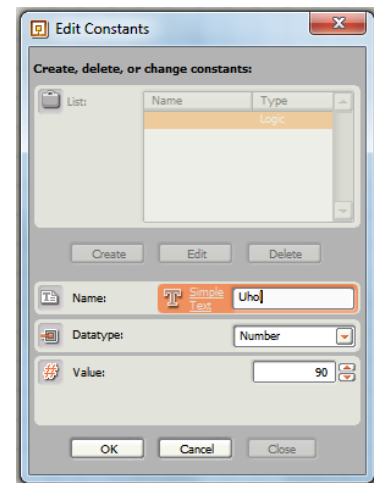


Obr. 49 Uhol 90° , 180° , 270° alebo 360°

Použijeme tri bloky z palety Data, a to Random, Constant a Math. Bloku Random nastavíme interal Minimum: 1, Maximum: 4.

S blokom Constant sme sa ešte nestretli, preto si podrobnejšie vysvetlíme, ako vytvoriť novú konštantu $Uhol = 90^\circ$.

V hlavnom menu klikneme na záložku Edit/Define Constants. Zobrazí sa okno, v ktorom môžeme vytvoriť konštantu Uhol pomocou tlačidla Create. Do poľa Name vpíšeme jej meno Uhol a do poľa Datatype zvolíme dátový typ Number (číslo), nastavíme Value (hodnota): 90 a stlačíme tlačidlo OK. Nová konštanta Uhol je vytvorená.

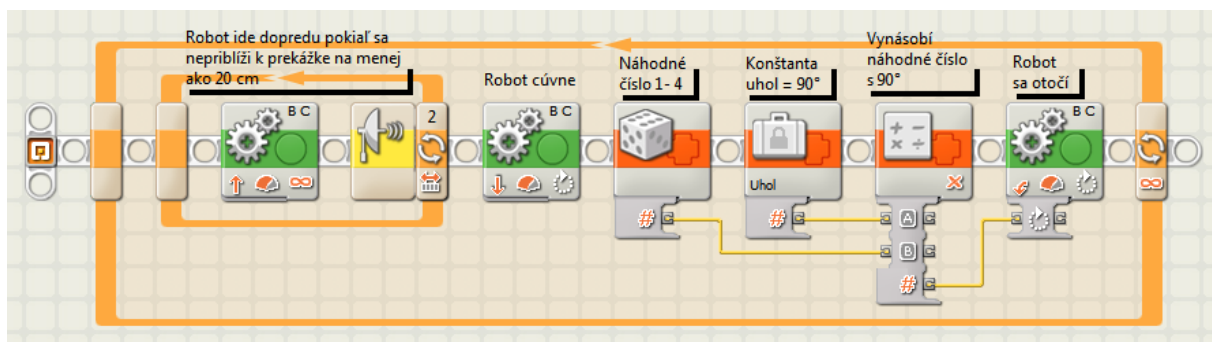


Obr. 50 Vytvorenie konštanty

Tretím blokom je blok Math, ktorému nastavíme Operation: Multiplication (násobenie).

Keď správne poprepájame drôty medzi týmito tromi dátovými blokmi a otáčkami motorov (duration), program bude dokončený.

Prerobný program:



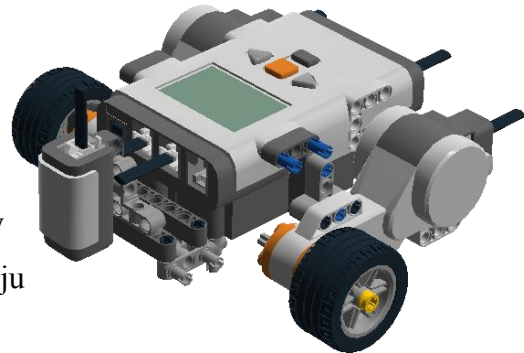
Obr. 51 Robot sa otočí o náhodný uhol (90° , 180° , 270° alebo 360°)

4.4.10 Zastaň na čiernej

Jednoduchý program, v ktorom robot hľadá čiernu farbu na podložke. Keď ju nájde, zastane.

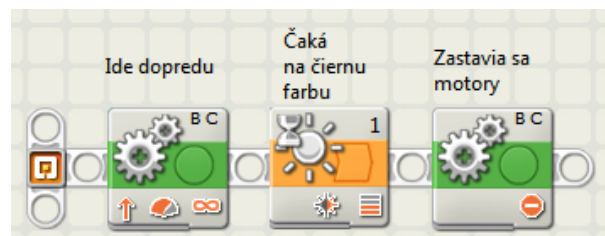
Stavba: NXT kocka, 2 servomotory, svetelný senzor

Program: Robot sa pohybuje dopredu, kým svetelný senzor nezameria čiernu farbu na podložke. Keď ju



Obr. 52 Robot so svetelným senzorom

zameria, zastane. K tomu použijeme blok čakanie, ktorému nastavíme Light sensor. Aby svetelný senzor rozoznal čiernu farbu pomocou NXT kocky zistíme intenzitu odrazeného svetla od čiernej farby. Na NXT kocke v menu View vyberieme Reflected light a môžeme začať s meraním. Svetelnému senzoru nastavíme pre čiernu farbu $\text{Light} < 40$.



Obr. 53 Zastaň na čiernej

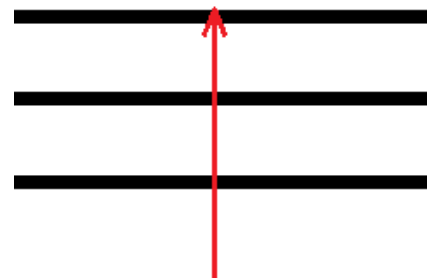
4.4.11 Zastavenie na tretej čiari

Jednoduchý program, v ktorom sa robot pohybuje vpred a počíta čierne čiary. Keď napočíta tri čierne čiary, zastaví sa.

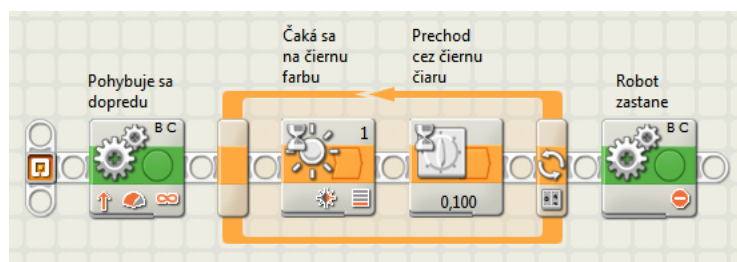
Stavba: NXT kocka, 2 servomotory, svetelný senzor

Použijeme koštrukciu robota z predchádzajúcej časti.

Program: Robot sa rozbehne naprieč čiernymi čiarami. Aby však dokázal počítať čierne čiary, musíme vložiť blok s čakaním. Program čaká kým svetelný senzor zameria čiernu farbu. Keďže čierna farba je asi 2cm široká a chceme, aby ju započítal iba raz, vložíme ďalej blok čakaj, ktorému nastavíme do Control: Time Seconds: 0,1. Keďže čiary sú tri musíme to trikrát zopakovať, preto



Obr. 54 Plocha s tromi čiarami



Obr. 55 Zastavenie na tretej čiari

tieto dva bloky dáme do cyklu a nastavíme mu Count: 3. Aby sa robot zastavil na tretej čiernej čiare, za cyklus vložíme blok motory, pomou ktorého oba motory zastavíme.

4.4.12 Počítanie čiar

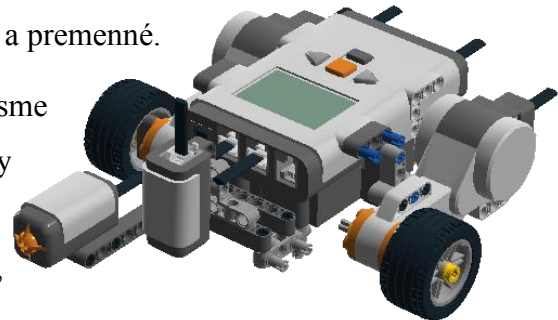
Program, pri ktorom sa robot hýbe vpred a pri tom počíta počet čiar. Tento počet zapisuje na displej. Robot sa zastaví, keď narazí na prekážku (zatlačený dotykový senzor).

Stavba: NXT kocka, 2 servomotory, svetelný senzor, dotykový senzor

Program: V programe sa používa paralelný proces a premenné.

Paralený proces využívame preto, aby sme dokázali naraz vykonávať dve veci (vetvy programu), a to:

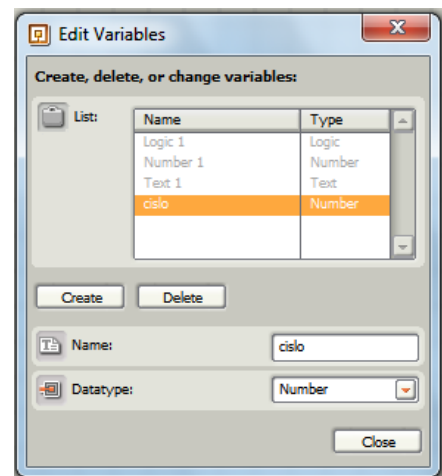
1. pohybovať sa vpred a súčasne počítať čiaary,
2. sledovať, či sme nenarazili na nejakú prekážku.



Obr. 56 Počítajúci robot

Prvá vetva:

Najprv si vytvoríme premennú číslo, a to tak, že v menu klikneme na záložku Edit/Define Variables. Zobrazí sa okno, v ktorom môžeme vytvoriť premennú číslo pomocou tlačidla Create. Do poľa Name vpíšeme jej meno číslo a do poľa Datatype zvolíme dátový typ Number (číslo) a stlačíme tlačidlo Close. Nová premenná číslo je vytvorená. Túto premennú využijeme na počítanie počtu prejdenných čiar.



Obr. 57 Tvorba premennej

Na začiatku programu bude blok premenná číslo, ktorej nastavíme Action: Write, Value: 0. Premennú číslo musíme vždy na začiatku programu vynulovať, lebo by mala v sebe zapísanú vždy inú hodnotu, ktorú by si pamätala z iného programu.

Potom vložíme blok Move z palety Common. Motorom B, C nastavíme Duration: Unlimited. Kolesá motora sa budú hýbať do nekonečna a pritom bude svetelný senzor zaznamenávať



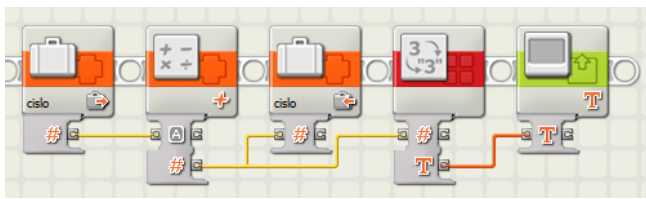
Obr. 58 Prechod medzi čiernou a bielou farbou

prechod medzi čiernou a bielou farbou, čo dáme do nekonečného cyklu, keďže sa očakáva, že čiar bude viac, ale my nevieme koľko.

V prvom bloku so svetelným senzorom zaznamenávame čiernu farbu a v druhom bielu farbu. Aby však jednu čiernu čiaru započítalo iba raz použijeme blok Wait z palety Flow (Control: Time, Seconds: 0,1).

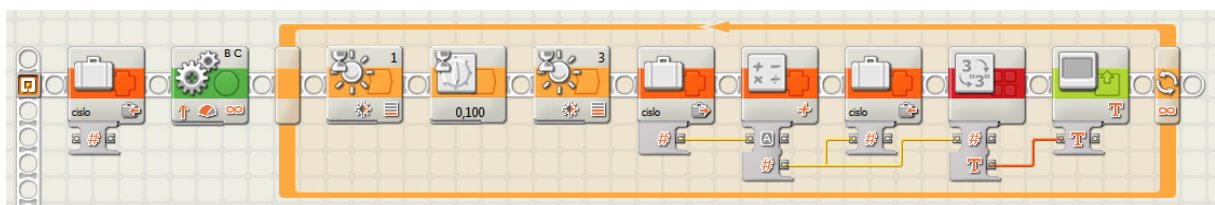
V tomto momente vie program načítať čiernu čiaru, ale nevie ich spočítať. Preto musíme do cyklu vložiť ďalšie bloky na počítanie čiar. Bloky z palety Data Variable, Math, blok Number toText z palety Advanced a Display z palety Action. Tieto bloky treba vhodne poprepájať drôťmi, aby sme zabezpečili funkčnosť programu.

Táto časť cyklu funguje tak, že sa najprv načíta obsah premennej číslo, pri zachytení čiernej čiary sa pripočíta 1, čím sa zvýši hodnota premennej číslo. Potom treba previesť číslo na text, aby sa počet čiar zobrazil na displeji.



Obr. 59 Počítanie čiar

Dokončená prvá vetva programu:



Obr. 60 Prvá vetva

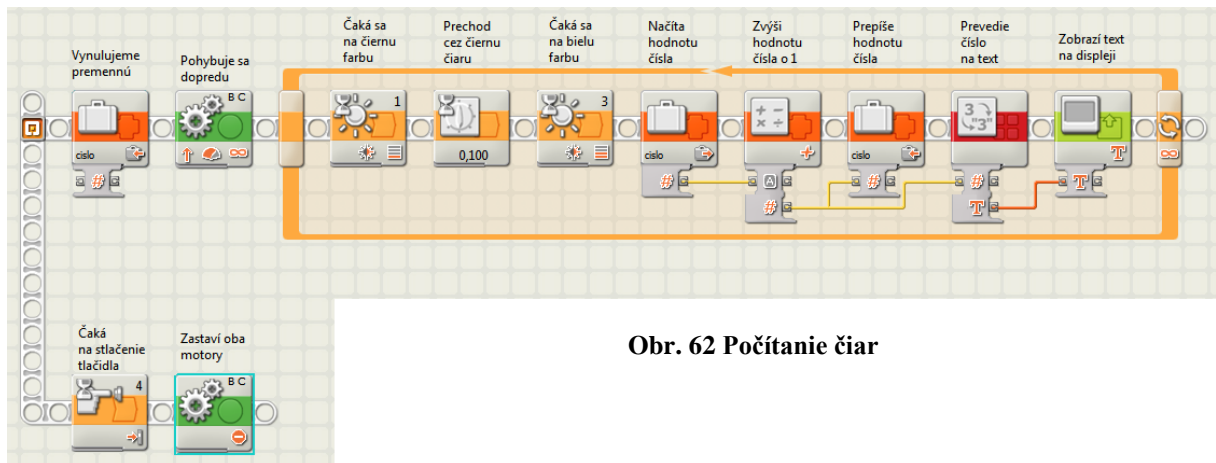
Druhá vetva:

Testuje, či robot nenarazil na nejakú prekážku. Ak narazil, robot zastane.



Obr. 61 Druhá vetva

Dokončený program s obidvomi vetvami (paralelnými procesmi):



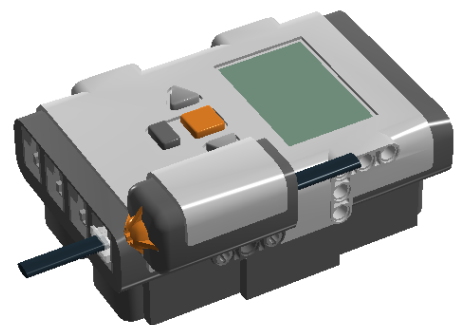
Obr. 62 Počítanie čiar

4.4.13 Počítanie stisnutí

Program, ktorý počíta počet stisnutí tlačidla dotykového senzora. Tento počet sa zobrazuje na displeji.

Stavba: NXT kocka, dotykový senzor

Program: V tomto programe použijeme premennú Poc, ktorá bude slúžiť na počítanie počtu stisnutí tlačidla dotykového senzora a vetvenie programu (Switch).

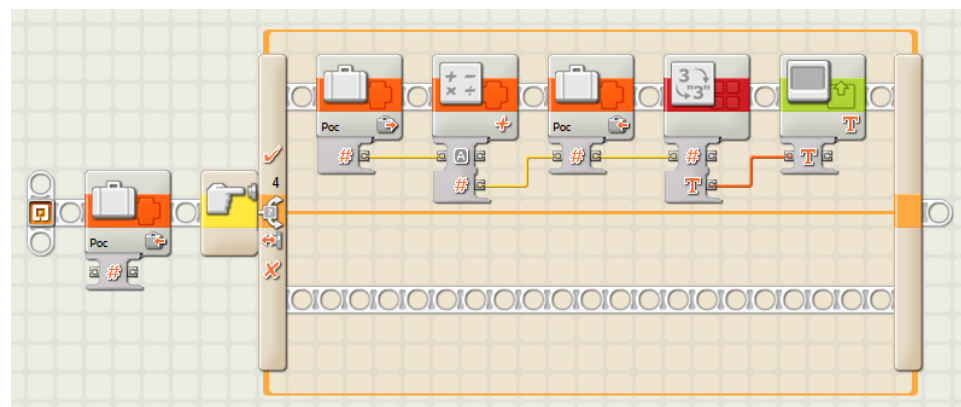


Obr. 63 Počet stisnutí

Ako prvé vytvoríme premennú Poc, ktorú vynulujeme. Potom vložíme do programu blok Switch z palety Common. Nastavíme vetvenie podľa toho, či bol zatlačený dotykový senzor. Zvolíme Control: Sensor, Sensor: Touch Sensor, Port: 4 a Action: Bumped.

Keď máme rozhodovací blok nastavený, do časti true (bumped) vložíme bloky

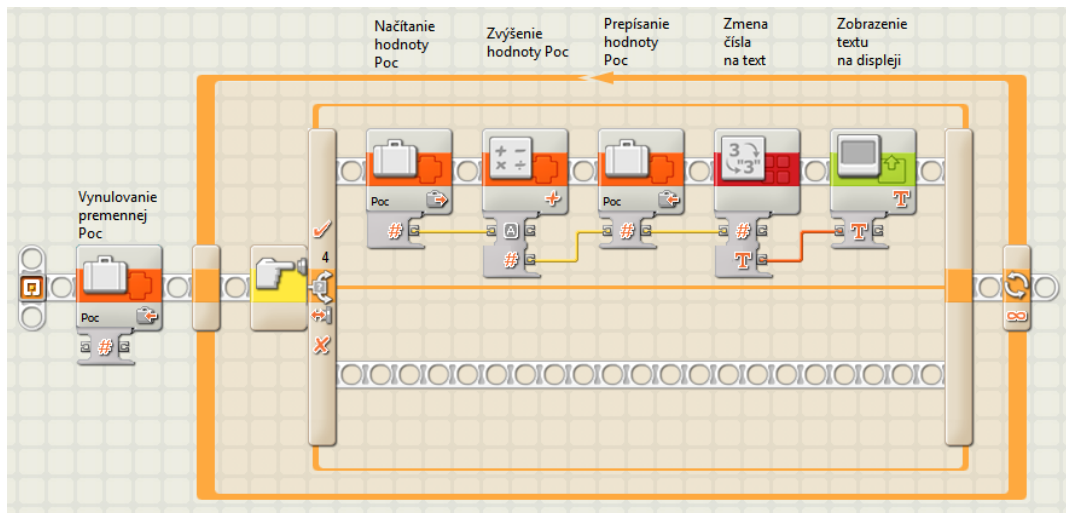
Variable
(premennú Poc)
z palety Data
Math (Addition -
pripočítavanie),
z Advanced
Number to Text a
z Action Display.



Obr. 64 Rozhodovací blok - vetvenie programu

Jednotlivé bloky vhodne poprepájame, aby sa počet stisnutí zobrazil na displeji. Do časti false nevložíme žiadne bloky, ostane prázdna.

Zdalo by sa, že program je dokončený, no v tomto momente sa započíta iba jedno stlačenie. Celý blok Switch musíme zacykliť – opakovať do nekonečna.



Obr. 65 Počítanie stisnutí

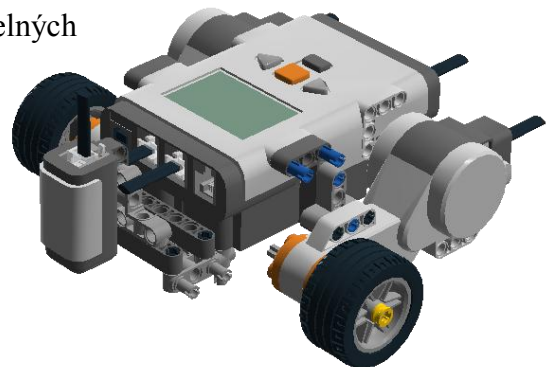
4.4.14 Pohyb po čiernej čiare

Program, pomocou ktorého sa robot snaží pohybovať po čiernej čiare. Táto úloha sa dá vyriešiť pomocou jedného alebo dvoch svetelných senzorov.

Konštrukcia s jedným svetelným senzorom:

Stavba: NXT kocka, 2 servomotory, svetelný senzor

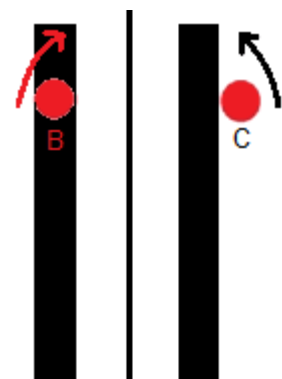
Program: V programe použijeme vetvenie programu



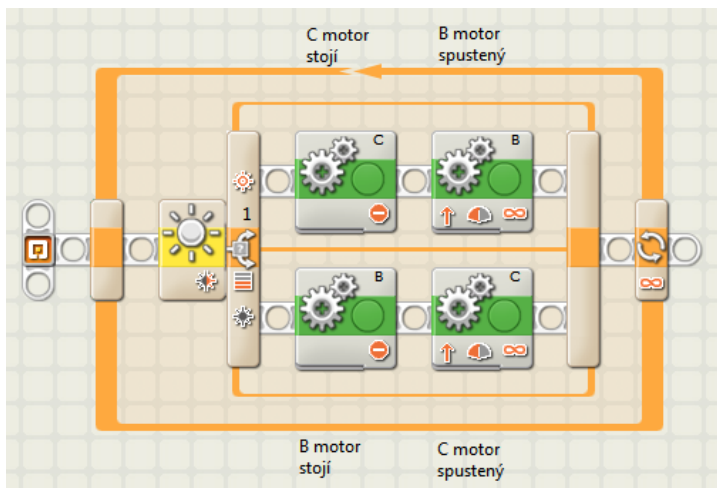
Obr. 66 Robot s jedným svetelným senzorom

(Switch) s rozhodovaním podľa svetelného senzora.

V programe budeme využívať vetvenie, preto vložíme do programu blok Switch z palety Common, ktorému nastavíme Control: Sensor, Sensor: Light Sensor, Port: 1 a Compare (true biela farba), $Light > 50$.



Obr. 67 Jeden svetelný senzor



Obr. 68 Program a jedným svetelným senzorm

Robot pomocou svetelného senzora sleduje „pravý okraj“ čiernej farby. Ak sa svetelný senzor nachádza nad čiernou farbou, zapne sa motor B a robot sa otočí vpravo, ak sa svetelný senzor nachádza nad bielou farbou, zapne sa motor C a robot sa otočí vľavo.

Blokom Move (motory), ktoré sú v pohybe, nastavíme

Power: 50, Duration: Unlimited.

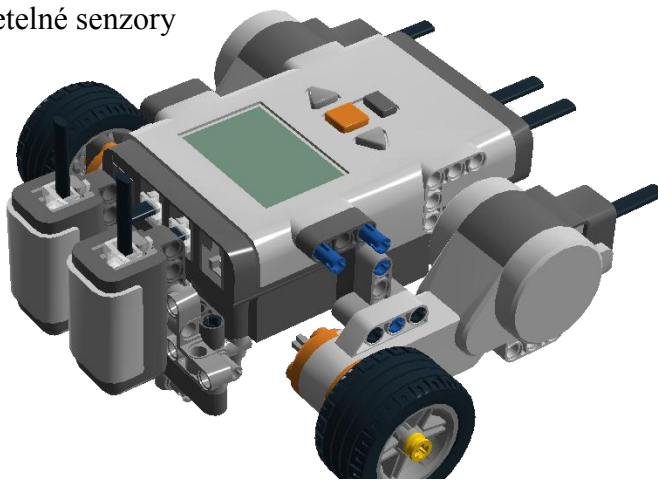
Nesmieme však zabudnúť na zacyklenie celého programu, aby sa sledovanie čiary opakovalo do nekonečna.

Konštrukcia s dvoma svetelnými senzormi:

Stavba: NXT kocka, 2 servomotory, 2 svetelné senzory

Program: V tomto programe využijeme o niečo zložitejšie vetvenie programu, keďže pri konštrukcii robota s dvoma svetelnými senzormi máme 4 možnosti, ktoré môžu nastať.

Do programu vložíme blok Switch z palety Common. Vznikli dve



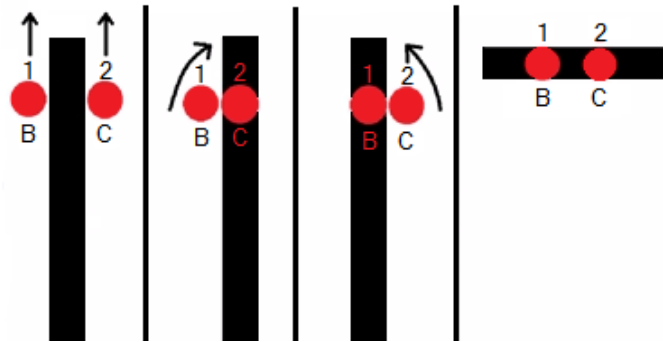
Obr. 69 Robot s dvoma svetelnými senzormi

vetvy programu. Do každej vetvy (bloku Switch) pridáme ešte po jednom bloku Switch. Pozri obr. 71. Všetkým trom blokom Switch nastavíme Control: Sensor, Sensor: Light Sensor a Compare (true biela farba), Light > 50. Obom senzorom treba nastaviť porty, a to Port 1 a 2.

Ďalej si popíšeme možnosti vetvenia programu:

- oba senzory zaznamenajú bielu, potom robot musí ísť dopredu.

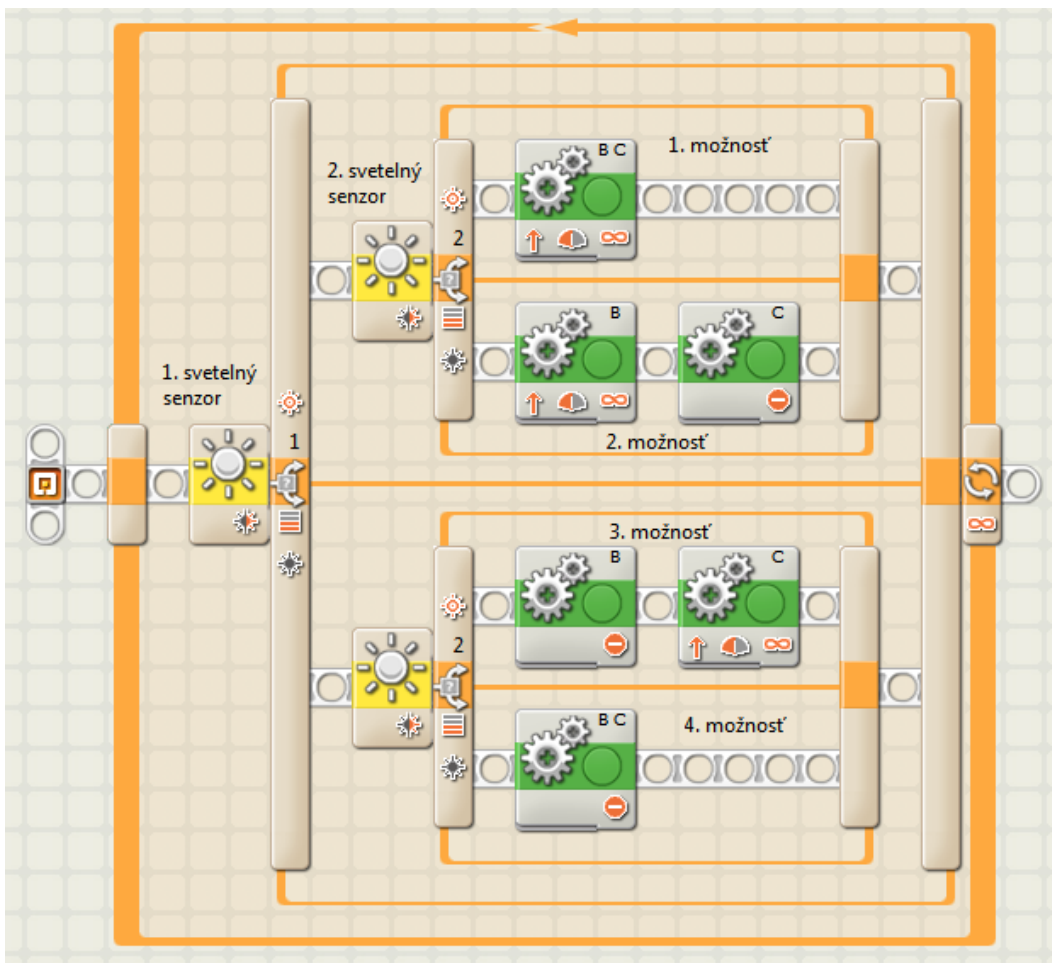
- 1 bielu, 2 čiernu, A motor ide, B motor stojí.
- 1 čiernu, 2 bielu, A motor stojí, B motor ide.
- oba senzory zaznamenajú



čiernu, robot zastane.

Obr. 70 Dva svetelné senzory

Bloky Move (motory), ktoré sa hýbu, nastavíme Power: 50, Duration: Unlimited.



Obr. 71 Program s dvomi svetelnými senzormi

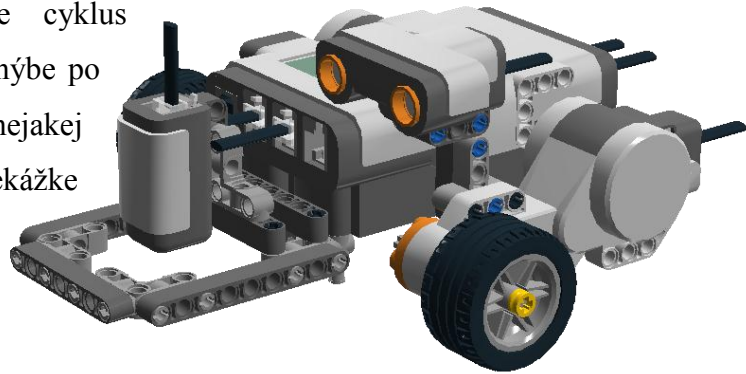
Celý program treba zacykliť, aby sa zisťovanie farby čiary opakovalo donekonečna.

4.4.15 Pohyb po čiernej čiare s prekážkou

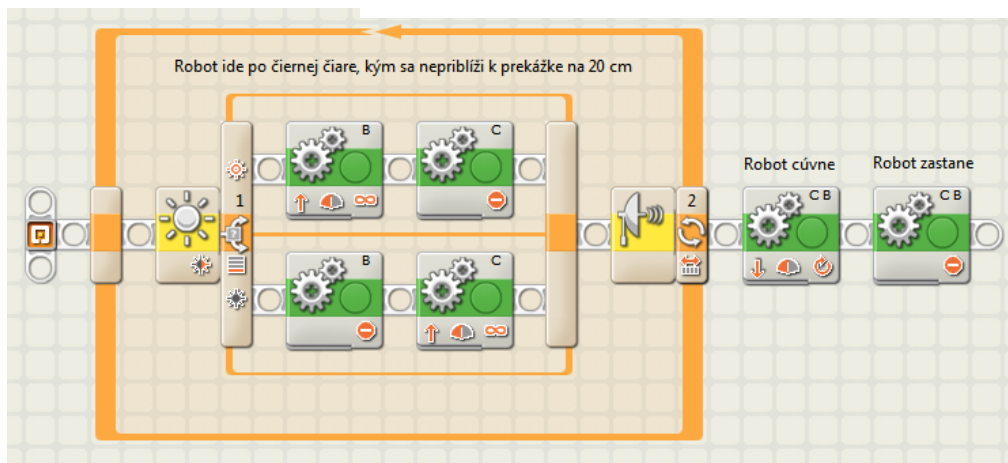
Program, pomocou ktorého sa robot pohybuje po čiernej čiare, pri čom sa na čiare nachádza nejaká prekážka.

Stavba: NXT kocka, 2 servomotory, svetelný senzor, ultrasonický senzor

Program: V programe použijeme cyklus s podmienkou na konci. Robot sa hýbe po čiernej čiare pokiaľ sa nepriblíži k nejakej prekážke na 20 cm. Ak sa k prekážke priblíži na 20 cm trochu cúvne a zastane.

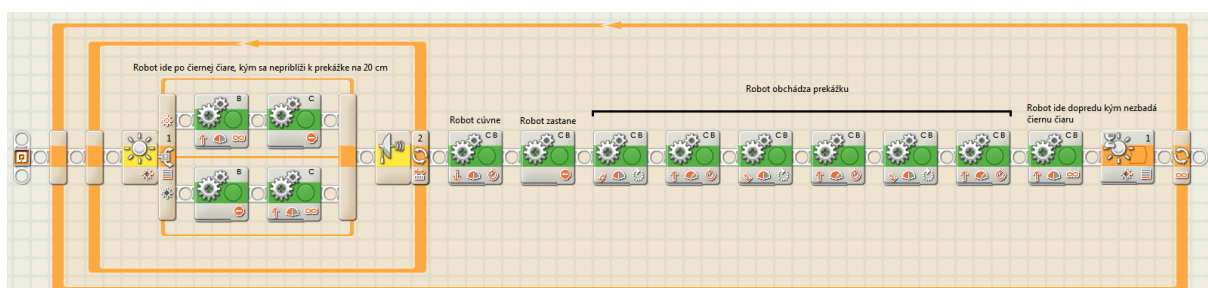


Obr. 72 Robot obchádzajúci prekážky



Obr. 73 Pohyb po čiernej čiare do zastavenia

Do programu ešte treba dorobiť obídenie prekážky. Robot by mohol obchádzať prekážku sprava a vrátiť sa na čiernu čiaru. Pri obídení prekážky sa bude pohybovať vpred, až kým nezameria čiernu čiaru. Keďže nevieme koľko prekážok sa nachádza na čiare, celý program dáme do nekonečného cyklu.



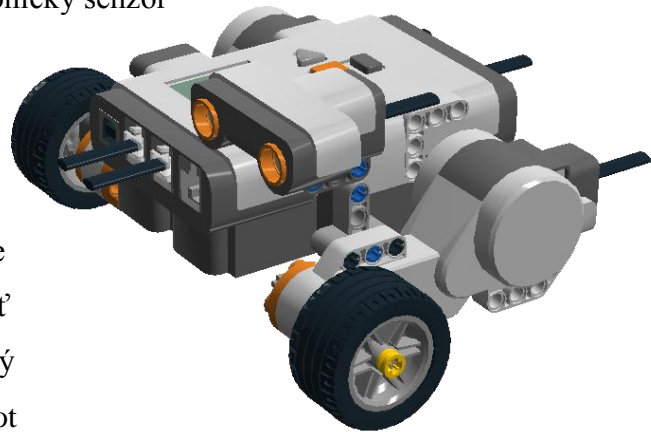
Obr. 74 Pohyb po čiernej čiare s obídením prekážky

4.4.16 Nasleduj človeka

Robot bude nasledovať človeka, ale bude si udržiavať odstup na 30 cm. Ak sa človek bude vzdďalovať, robot sa bude hýbať dopredu, kým sa nepriblíži na 30 cm, ak sa človek začne približovať k robotovi, tento sa začne pohybovať vzad.

Stavba: NXT kocka, 2 servomotory, ultrasonický senzor

Program: Vytvoríme program, v ktorom využijeme vetvenie. Budeme používať ultrasonický senzor na meranie vzdialenosti. Robot sa bude pohybovať vpred a vzad podľa toho, či je vzdialený viac ako 30 cm alebo menej. Aby sa robot



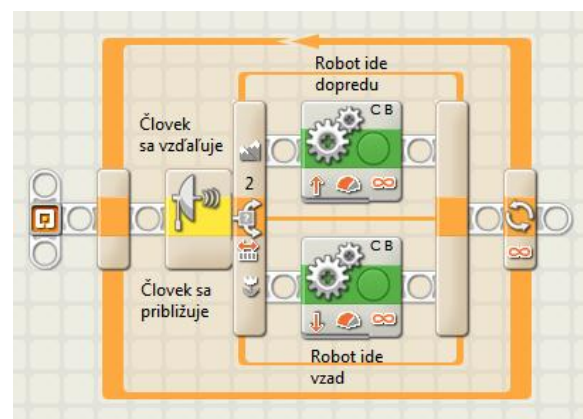
Obr. 75 Robot prenasledovateľ

pohyboval vpred alebo vzad, použijeme blok Move (motory), ktorému musíme nastaviť Duration: Unlimited, Power: 75 (rýchlosť je konštantná).

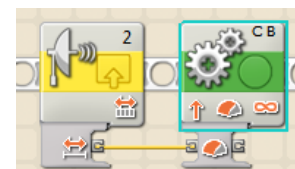
Ďalej sa pokúsime tento jednoduchý program prerobiť tak, aby sa menila rýchlosť v závislosti na vzdialenosti.

Prvá vetva: Čím viac sa bude človek

vzďaľovať robotovi, tým rýchlejšie sa bude pohybovať vpred. Keď sa robot bude približovať človekovi, bude spomaľovať. Použijeme prepájacie drôty, ktorými prepojíme Distance



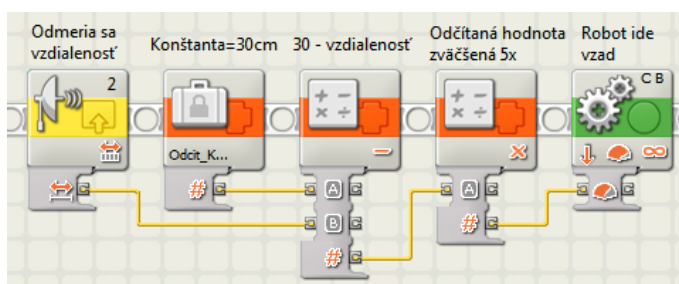
Obr. 77 Prenasledovanie



Obr. 76 Prvá vetva

(vzdialenosťou) s Power (rýchlosťou motora).

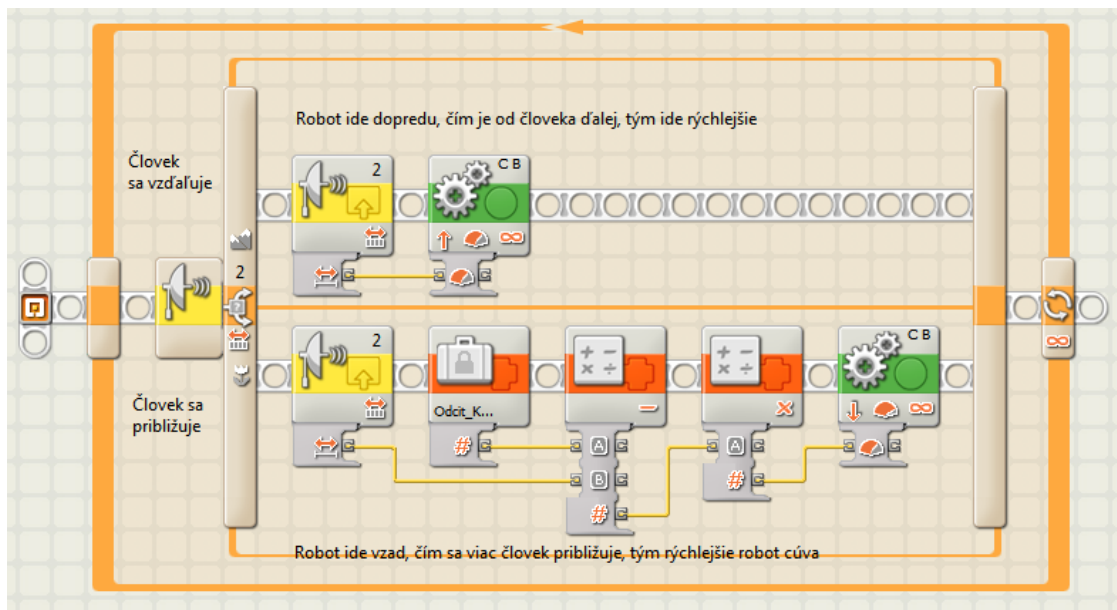
Druhá vetva: Čím viac sa bude človek približovať robotovi, tým rýchlejšie sa bude robot pohybovať vzad (cúvať).



Obr. 78 Druhá vetva

V tejto vetve nastáva problém, keby sme postupovali tak ako v prvej vetve, robot by sa hýbal presne naopak, čím by bol človek bližšie, tým by robot cúval pomalšie. Aby program fungoval ako má, použijeme vzorec: $(30 - \text{vzdialenosť}) \times 5 = \text{Power}$. Hodnotu 30 vložíme do konštanty Odcit_Konst, od ktorej budeme odčítavať vzdialenosť získanú z ultrasonického senzora. Aby bola rýchlosť dostatočná, túto hodnotu vynásobíme piatimi a priradíme drôtom rýchlosti motora.

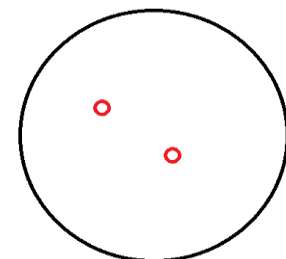
Dokončený program:



Obr. 79 Prenasledovanie s premenlivou rýchlosťou

4.4.17 Sumo robot

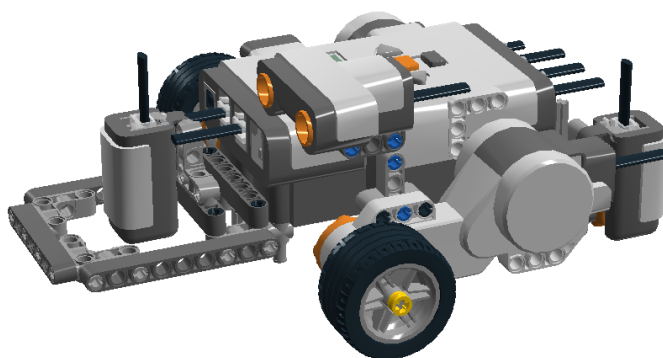
Pokúsime sa zostrojiť sumo robota. Je to taký robot, ktorý sa pohybuje v kruhovej aréne a snaží sa z nej vytlačiť druhého sumo robota. Vyhráva ten robot, ktorý vytlačí toho druhého a sám ostane v aréne.



Obr. 80 Aréna

Stavba: NXT kocka, 2 servomotory, 1 svetelný senzor

Program: Robot sa bude pohybovať vpred pokiaľ nezameria čiernu farbu. Keď zameria čiernu farbu, vráti sa vzad a otočí sa. Toto celé sa

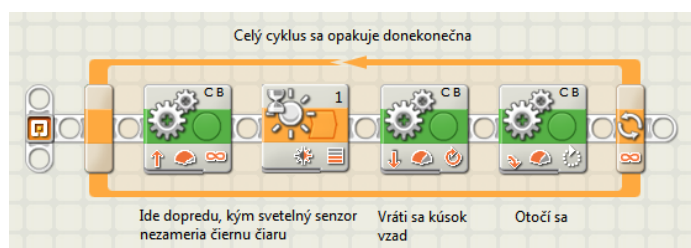


Obr. 81 Sumo robot

bude opakovať donekonečna.

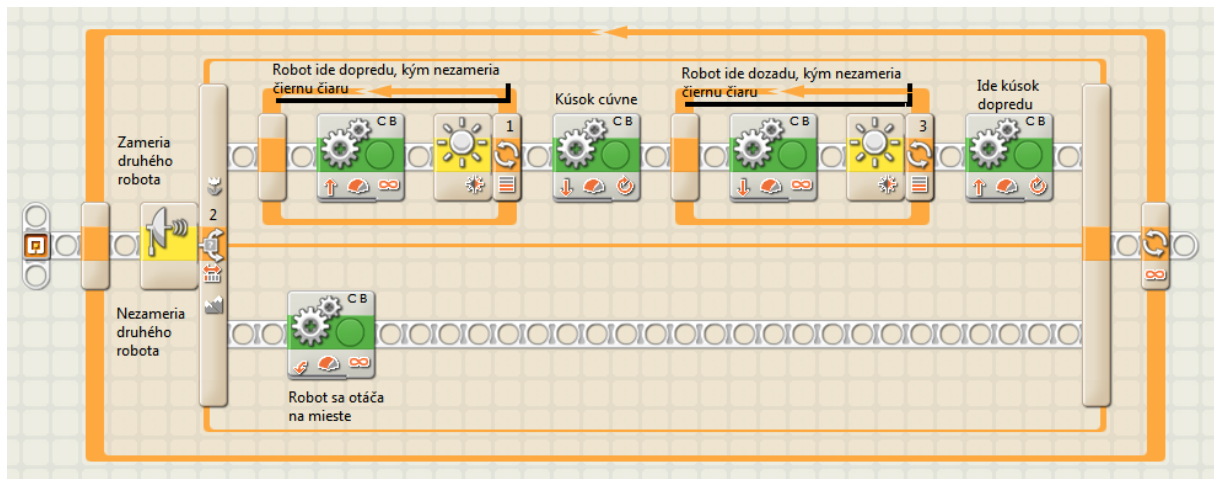
Stavba: NXT kocka, 2 servomotory, 2 svetelné senzory, ultrasonický senzor

Program: V programe použijeme vetvenie. Robot najprv zameria či sa



Obr. 82 Sumo robot s jedným svetelným senzorom

pred ním nenachádza druhý sumo robot, ak nie tak sa začne otáčať kým ho nezameria. Ak robota zameria, začne sa pohybovať dopredu (vytláčať druhého sumo robota), pokiaľ nezameria čiernu čiaru prvým svetelným sensorom vpredu. Ak čiernu čiaru zameria, začne cúvať kým nezameria čiernu čiaru druhým svetelným sensorom vzadu. Toto celé treba opakovať donekonečna.



Obr. 83 Sumo robot s využitím dvoch svetelných senzorov

5 SPOLOČNOSŤ A TECHNOLOGIE

5.1 Odvrátená strana virtuálneho sveta

Počítačové pirátstvo je nepovolený prístup do cudzieho počítača.

Softvérové pirátstvo je nezákonné používanie softvéru, ktorý je chránený autorským zákonom. K pirátstvu dochádza pri kopírovaní, zdieľaní, sťahovaní, pri samotnom predaji a inštalácii softvéru na viac počítačov, ako je uvedené v licenčnej zmluve.

Softvérové pirátstvo je druh počítačovej kriminality, kedy sa jedná o neoprávnené používanie, šírenie, upravovanie softvéru. Aj vtedy keď ide o legálne získaný softvér.

Licenčná zmluva obsahuje podmienky používania konkrétneho softvéru. Rôzne programy majú rôzne podmienky používania.

Autorské právo (copyright) je právo rozhodovať o vlastnom diele. Je garantované zákonmi krajiny, v ktorej bolo dielo vydané autorom. Medzinárodné dodržiavanie autorských práv je umožnené medzinárodnými zmluvami.

Na označenie svojho diela môže autor použiť zápis typu: © dátum_vzniku meno_autora. Značku © môže autor použiť bez registrácie svojho diela.

Business Software Alliance (BSA) neziskové združenie, ktoré sa snaží bojovať proti softvérovým pirátom. Chráni práva duševného vlastníctva autorov softvéru, presadzuje zákony v oblasti softvérových autorských práv.

Porušiteľom autorských práv a duševného vlastníctva hrozia sankcie – odobratie nelegálnych kópií, finančné pokuty, i väzenie. V trestnom procese ide o možnosť odňatia slobody až na osem rokov, alebo vysoký finančný trest, prepadnutie veci, povinnosť nahradiť spôsobenú škodu. V občianskom procese ide o súdny príkaz zastaviť akékoľvek ďalšie používanie softvéru, náhrada škôd a aj uhradenie zisku z nelegálneho používania či distribúcie, úhrada súdnych trov.

Ako sa líši legálny softvér od toho nelegálneho:

- Legálny softvér sa predáva v originálnych, neporušených obaloch s ochrannými prvkami. Je pri ňom priložená licenčná zmluva, certifikát autenticity a riadny daňový doklad.

- Nelegálny softvér je šírený bez originálneho obalu, býva napálený na obyčajnom médiu, kde sa môže nachádzať aj viac programov od viacerých výrobcov. Obvikle chýba licenčná zmluva, daňový doklad a cena býva nezvyčajne nízka.

ZÁVER

Tento učebný zdroj vznikol z nedostatku materiálu na novozavedený predmet Digitálny svet. Pomocou tohto učebného zdroja by sa mohli žiaci lepšie pripravovať na vyučovanie, získali by lepší prehľad o preberaných témach.

Učebný zdroj môže poslúžiť nielen žiakom, ale aj učiteľom ako vhodná učebná pomôcka. Nachádzajú sa tu rôzne metodické materiály na postup práce z rôznych tematických okruhov predmetu Digitálny svet.

Učebný zdroj obsahuje priložené CD s vytvoreným materiálom, ktorý je možné využiť priamo na vyučovacích hodinách ako vzorovú ukážku pri vysvetľovaní nového učiva, ale aj vytvorené modely robotov spolu s postupom konštrukcie a hotové programy vytvorené v programovacom prostredí Lego Mindstorms Educations NXT.

Časť učebného zdroja Algoritmy a programovanie by sa dala ďalej využiť aj pri príprave žiakov na robotickú súťaž.

Informačné a komunikačné technológie sú to, čo baví dnešné deti a mnohé z nich sa s robotom, ktorého môžu ovládať samy, nestretli.

Dúfam, že tento materiál zaujme a vhodne motivuje aj tie deti, ktoré nemajú až toľko príležitostí na stretnutie sa s modernými digitálnymi technológiami.

ZOZNAM POUŽITEJ LITERATÚRY

- [1] KALAŠ, Ivan – WINCZER, Michal: *Tvorivá informatika. Informatika okolo nás*. Bratislava: SPN - Mladé letá, 2007. 48. s. ISBN 978-80-10-00887-2
- [2] MŠ SR: *Štátny vzdelávací program pre 2. stupeň základnej školy v Slovenskej republike ISCED 2 – nižšie sekundárne vzdelávanie* [online]. 2008. [cit. 27. 12. 2013]. Dostupné na internete: < <http://www.noveskolstvo.sk/article.php?254> >
- [3] ŠNAJDER, Ľubomír – KIREŠ, Marián: *Informatika pre stredné školy. Práca s multimédiami*. Bratislava: SPN – Mladé letá, 2005. 48. s. ISBN 80-10-00422-7
- [4] HRABČÁK, Marek: *Multimédiá na PC*. Prešov: Prešovská Univerzita, 2006. 99. s. ISBN 978-80-8068-777-9
- [5] *Prehľad normy STN ISO 690*. [online]. 2010. [cit. 27. 12. 2013]. Dostupné na internete: <http://www.fem.uniag.sk/ksov/predmety/kvantitativna_makro/RefPopis/norma.doc>
- [6] BITZ Fits.: *OCR vs. ICR What's the difference?* [online]. 2010. [cit. 27. 12. 2013]. Dostupné na internete: <<http://thepaperlesstimes.com/theblog/2010/05/ocr-vs-icr-whats-the-difference/>>
- [7] PETROVIČ, Pavel: *NXT Projekty*. [online]. RoboWiki: 2006. [cit. 11. 1. 2014]. Dostupné na internete: <<http://virtuallab.kar.elf.stuba.sk/robowiki/index.php?title=Special:AllPages>>
- [8] JAKEŠ, Tomáš: *Robotické vzdelávaní*. [online]. Plzeň: 2010. [cit. 10. 1. 2014]. Dostupné na internete: <<https://www.lego.zcu.cz/web/moduly-systemu-lego-nxt>>
- [9] ČVUT: *Referenčná príručka ku grafickému jazyku NXT-G*. [online]. Praha: 2010. [cit. 17. 1. 2014]. Dostupné na internete: <<http://www.robosoutez.cz/files/nxtg.pdf>>
- [10] ROSENBERG, Neil: *Workshop – NXT Programming For Beginners*. [online]. Weaverville, NC: 2012. [cit. 12. 1. 2014]. Dostupné na internete: <http://www.rocwnc.org/Beginning_NXT_Programming_Workshop.pdf>
- [11] ČVUT: *NXT-G*. [online]. Praha: 2010. [cit. 19. 1. 2014]. Dostupné na internete: <support.dce.felk.cvut.cz/roboti/index.php?sekce=home&id=nxtg>

- [12] PETROVIČ, Pavel: *Výuka programovania pomocou grafických robotických programovacích jazykov pre začiatočníkov a pokročilých*. [online]. Bratislava: 2012. [cit. 3. 1. 2014]. Dostupné na internete: <http://dai.fmph.uniba.sk/~petrovic/pub/didinfo2012/didinfo_petrovic_2012.pdf>
- [13] CEEO: *NXT-G examples: Motors*. [online]. Boston, Massachusetts: 2013. [cit. 3. 1. 2014]. Dostupné na internete: <www.legoengineering.com/nxt-g-examples-motors/>
- [14] CEEO: *NXT-G examples: Sounds, displays, and delays*. [online]. Boston, Massachusetts: 2013. [cit. 3. 1. 2014]. Dostupné na internete: <<http://www.legoengineering.com/nxt-g-examples-sounds-displays-and-delays/>>
- [15] WIKIMEDIA FOUNDATION, INC.: *Copyright*. [online]. USA: 2013. [cit. 17. 1. 2014]. Dostupné na internete: <<http://en.wikipedia.org/wiki/Copyright>>
- WIKIMEDIA FOUNDATION, INC.: *Copyright*. [online]. USA: 2013. [cit. 15. 1. 2014]. Dostupné na internete: <<http://sk.wikipedia.org/wiki/Hypermédium>>
- [16] PALÁSTHY, Juraj: *Multimédiá I*. [online]. Orange Slovensko: 207. [cit. 10. 1. 2014]. Dostupné na internete: <http://www.oskole.sk/?id_cat=1008&clanok=17577>
- [17] TA SR: *Softvérové pirátstvo aj naďalej veľkým problémom*. [online]. Bratislava: 2013. [cit. 19. 1. 2014]. Dostupné na internete: <<http://www.aktuality.sk/clanok/224398/softverove-piratstvo-aj-nadalej-velkym-problemom/>>
- [18] SCHIKOR, Daniel: *Herbár*. [online]. 2012. [cit. 9. 1. 2014]. Dostupné na internete: <<http://www.byliny.sk/index.php>>
- [19] KAČINOVÁ, V.: *Mediálna výchova. Experimentálne učebné texty pre 9. ročník ZŠ*. [online]. Bratislava: ŠPÚ, 2007. [cit. 19. 1. 2014]. Dostupné na internete: <http://www.statpedu.sk/files/documents/svp/2stzs/isced2/ine_dokumenty/medialna_vychova_exp_ucebne_texty_9_rocnik.pdf>

PRÍLOHY

Zoznam príloh záverečnej práce:

- Príloha A – CD médium:
 - Herbár
 - Fotoalbum
 - Video
 - Lego modely
 - Lego programy