

Řešené úlohy z programování malých robotů se stavebnicí Lego Mindstorms EV3

Bc. Lukáš Hrtoň

Diplomová práce
2016



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Lukáš Hrtoň**
Osobní číslo: **A14411**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Učitelství informatiky pro střední školy**
Forma studia: **prezenční**

Téma práce: **Řešené úlohy z programování malých robotů se stavebnicí Lego Mindstorms EV3**

Téma anglicky: **Resolved Tasks from the Programming of Small Robots with LEGO Mindstorms EV3**

Zásady pro vypracování:

1. Provedte literární rešerši tématu, zaměřte se na oblast projektové výuky a využití interaktivních stavebnic.
2. Prostudujte vlastnosti stavebnice LEGO Mindstorms a její didaktické možnosti, včetně možností jejího zařazení do výuky.
3. Navrhněte několik navazujících projektově řešených úloh využívajících stavebnici LEGO Mindstorms.
4. Jednotlivé řešené úlohy realizujte a ověřte jejich efektivnost z pohledu výuky.
5. Sestavte systém hodnocení jednotlivých úloh.
6. Zhodnoťte účelnost navržených úloh z pohledu rozvoje dovedností žáka.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. W. H. KILPATRICK, The Project Method: The Use of the Purposeful Act in the Educative Process, University of California, Eleventh Impression March, 1929.
2. KRATOCHVÍLOVÁ, J., Teorie a Praxe projektové výuky. Brno: MU 2006, ISBN: 89-210-4142-0.
3. NOVÁK, D., Elektrotechnické stavebnice v technické výchově, Praha, 1997, ISBN: 80-86039-37-4.
4. SKALKOVÁ, J., Obecná didaktika. Praha: ISV, 1999, ISBN: 80-85866-33-1.
5. PAVELKOVÁ, I., Motivace žáků k učení: perspektivní orientace žáků a časový faktor v žákovské motivaci. Praha: Universita Karlova, 2002, ISBN: 80-729-0092-7

Vedoucí diplomové práce:

Ing. Peter Janků

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

5. února 2016

Termín odevzdání diplomové práce:

20. května 2016

Ve Zlíně dne 5. února 2016



doc. Mgr. Milan Adámek, Ph.D.
děkan



doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití, jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval (v případě publikace výsledků budu uveden jako spoluautor);
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 19. 5. 2016



.....
podpis diplomanta

ABSTRAKT

Diplomová práce je zaměřena na praktické úlohy se stavebnicí LEGO Mindstorms EV3 využitelné pro kurzy programování a robotiky na základních i středních školách. Teoretická část představuje využití interaktivních stavebnic v projektové výuce a seznamuje s možnostmi stavebnice. Výstupem práce je soubor řešených na sebe navazujících úloh se stavebnicí LEGO Mindstorms EV3. Praktická část se věnuje realizaci úloh, systému hodnocení a vyhodnocení navržených úloh.

Klíčová slova: LEGO Mindstorms, EV3, řešené úlohy, projektová výuka, programování, robotické systémy

ABSTRACT

This thesis is focused on practical tasks with LEGO Mindstorms EV3. These tasks can be used for courses of programming and robotics on primary or secondary schools. The theoretical part introduces use of interactive building sets in project teaching. It also introduces what is possible to do with this set. The output of the thesis is the set of solved consequential tasks with LEGO Mindstorms EV3. The practical part is focused on realization and the evaluation system of suggested tasks.

Keywords: LEGO Mindstorms, EV3, solved tasks, project teaching, programming, robotic systems

Touto cestou bych chtěl poděkovat našemu garantovi doc. Mgr. Romanovi Jaškovi, Ph.D. za zajímavé téma diplomové práce a za cenné rady.

Dále děkuji vedoucímu práce Ing. Peteru Janků za věcné námitky a podněty ke zlepšení práce.

Mé poděkování patří i rodině za trpělivost a podporu během celého studia.

„If you can't explain it simply, you don't understand it well enough.“

Albert Einstein

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 INTERAKTIVNÍ STAVEBNICE V PROJEKTOVÉ VÝUCE	11
1.1 PROJEKTOVÁ VÝUKA.....	11
1.1.1 Hodnocení projektu.....	12
1.2 VYUŽITÍ INTERAKTIVNÍCH STAVEBNIC	12
1.3 PŘEHLED INTERAKTIVNÍCH STAVEBNIC	13
1.3.1 LEGO Mindstorms EV3	13
1.3.2 Merkur.....	13
1.3.3 Arduino	14
1.4 POROVNÁNÍ MOŽNOSTÍ STAVEBNIC	15
2 LEGO MINDSORMS EV3	16
2.1 DIDAKTICKÉ MOŽNOSTI STAVEBNICE	16
2.2 POROVNÁNÍ S DŘÍVĚJŠÍMI VERZEMI.....	16
2.3 POPIS STAVEBNICE	17
2.3.1 Řídicí jednotka	18
2.3.2 Servomotory	19
2.3.3 Senzory.....	20
2.4 PROGRAMOVÁNÍ ROBOTŮ.....	23
2.4.1 Programování pomocí aplikace LEGO MINDSTORMS EV3	23
2.4.2 Programování pomocí mobilního telefonu nebo tabletu.....	23
2.4.3 Programování v jiných programovacích jazycích.....	24
II PRAKTICKÁ ČÁST	25
3 POPIS PROJEKTU	26
3.1 CÍLOVÁ SKUPINA.....	26
3.2 ČASOVÁ NÁROČNOST	26
3.3 VYUŽITÍ POMŮCEK	26
3.4 PRÁCE VE SKUPINÁCH	26
3.5 CÍL PROJEKTU.....	27
3.6 UKONČENÍ PROJEKTU	27
3.7 NAVAZUJÍCÍ AKTIVITY.....	27
3.8 STRUKTURA ÚLOH PROJEKTU	27
4 POPIS PROGRAMOVACÍHO JAZYKA	28
4.1 VÝVOJOVÉ PROSTŘEDÍ	28
4.2 PROGRAMOVACÍ BLOKY	28
4.2.1 Akční bloky.....	28
4.2.2 Funkční bloky.....	29
4.2.3 Senzorové bloky.....	29
4.2.4 Operační bloky	30
4.2.5 Pokročilé bloky	30
5 PROJEKT: ROBOTICKÉ VOZÍTKO	31

5.1	ÚLOHA 1: SESTAVENÍ VOZÍTKA.....	32
5.1.1	Zadání.....	32
5.1.2	Komentované řešení.....	32
5.2	ÚLOHA 2: OVLÁDÁNÍ VOZÍTKA	33
5.2.1	Zadání.....	33
5.2.2	Komentované řešení.....	33
5.3	ÚLOHA 3: PRVNÍ PROGRAMOVÁNÍ.....	36
5.3.1	Zadání.....	36
5.3.2	Komentované řešení.....	36
5.4	ÚLOHA 4: SENZORY	39
5.4.1	Zadání.....	39
5.4.2	Komentované řešení.....	40
5.5	ÚLOHA 5: ZATÁČENÍ VOZÍTKA	47
5.5.1	Zadání úlohy.....	47
5.5.2	Komentované řešení.....	47
5.6	ÚLOHA 6: PROMĚNNÉ A OPERACE S ČÍSLY.....	51
5.6.1	Zadání.....	51
5.6.2	Komentované řešení.....	51
5.7	ÚLOHA 6: POKROČILÉ PROGRAMOVÁNÍ	55
5.7.1	Zadání.....	55
5.7.2	Komentované řešení.....	55
5.8	ÚLOHA 7: INTELIGENTNÍ CHOVÁNÍ VOZÍTKA.....	58
5.8.1	Zadání.....	58
5.8.2	Komentované řešení.....	58
6	SYSTÉM HODNOCENÍ ÚLOH.....	60
6.1	PRŮBĚŽNÉ HODNOCENÍ	60
6.2	ZÁVĚREČNÉ HODNOCENÍ.....	60
6.3	SEBEHODNOCENÍ ŽÁKŮ	60
7	ROZVOJ DOVEDNOSTÍ ŽÁKA.....	61
7.1	MECHANICKÁ ZRUČNOST	61
7.2	SCHOPNOST ŘEŠENÍ PROBLÉMŮ	61
7.3	PŘEDSTAVIVOST.....	61
7.4	LOGICKÉ UVAŽOVÁNÍ.....	61
7.5	PRÁCE V TÝMU	61
	ZÁVĚR	62
	SEZNAM POUŽITÉ LITERATURY.....	63
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	65
	SEZNAM OBRÁZKŮ	66
	SEZNAM TABULEK.....	69
	SEZNAM PŘÍLOH.....	70

ÚVOD

Ve školách dnes často převažuje slovně názorné a transmisivní pojetí výuky (přenos ~ transmise). Přeceňuje se výklad učitele, tedy předávání poznatků. Žák je zatlačen do pasivity – spíše poslouchá, zapisuje si, pozoruje, vštěpuje si poznatky do paměti a později je reprodukuje. Při takové výuce žák přichází o získání vlastní zkušenosti.

Projektové vyučování je cesta, jak zvýšit smysluplnost a efektivitu výuky. Žáci jsou vedeni a motivováni k aktivitě a kreativitě. Projekt by měl zaměstnávat nejlépe nejen mozek, ale i ruce a tím zapojit více smyslů do akce. Projektová výuka si klade za cíl odstranit nedostatky slovně názorné výuky více se zaměřit na praktické schopnosti a logické myšlení.

Tato práce se zaměřuje na rozvoj projektové výuky. Cílem je příprava zadání a komentovaného řešení úloh pro zájmový kroužek programování a robotiky. Předpokládaným výstupem je návrh úloh a realizace na sebe navazujících řešení, která rozvíjejí postupnou formou technické i programátorské schopnosti dětí primárně na základní škole. Nedílnou součástí řešení je i navrhnutá metodika hodnocení výstupů sestavená tak, aby mohla být součástí reálného hodnocení žáka nebo motivačního programu.

Při tvorbě práce se předpokládá využití rozsáhlých možností stavebnice Lego Mindstorms EV3 a odpovídajících programovacích jazyků a nástrojů.

I. TEORETICKÁ ČÁST

1 INTERAKTIVNÍ STAVEBNICE V PROJEKTOVÉ VÝUCE

1.1 Projektová výuka

Základ projektové metody otevřel v roce 1918 William Heard Kilpatrick, čerstvý profesor učitelské koleje Kolumbijské univerzity v New Yorku, svým článkem *The Project Method*. S jeho jménem je také nadále metoda spojována. Kilpatrick definuje projekt takto: „*Projekt jest určitě a jasně navržený úkol, který můžeme předložit žákovi tak, aby se mu zdál životně důležitým tím, že se blíží skutečné činnosti lidí v životě*“ [1,6]

Existují i další definice, například pojem projektová výuka dobře popsala v knize *Teorie a praxe projektové výuky* J. Kratochvílová: „*Podstatným znakem výukové metody je, že žáci projekt realizují od jeho plánování až po vytvoření konkrétního produktu a konkrétního výstupu projektu a svoje zkušenosti zprostředkovávají druhým.*“ [2]

Projektové vyučování je založeno na řešení komplexních teoretických nebo praktických problémů na základě aktivní činnosti žáků. Překonává nedostatky běžného vyučování, jeho izolovanost, odtrženost od životní praxe, odcizení od zájmu dětí a nízkou motivaci. Nechce však zcela odstranit běžné vyučování, spíše vyplňuje jeho mezery. [4]

Projektové vyučování se orientuje především na zkušenosti žáka. Vychází z předpokladu, že předměty získávají význam, pokud se včleňují do lidských zkušeností. V kontextu se životem, který je žákům blízký, vznikají otázky a probouzí se přirozený zájem o poznávání. Koncepce dále vychází z předpokladu, že od sebe nelze odtrhávat poznání a činnost, práci hlavy a rukou. Rozumí se činnost teoretická i praktická, individuální i sociální. Jde zároveň o takovou činnost, se kterou se děti ztotožňují. [4]

Postup při projektovém vyučování lze charakterizovat následujícími momenty:

- 1) Volí se situace, která představuje pro žáky skutečný problém.
- 2) Se žáky se diskutuje plán řešení zvoleného problému.
- 3) Rozvíjejí se činnosti vyžadující řešení tohoto problému.
- 4) Závěr projektu (zveřejnění výsledku práce, zhodnocení projektu). [4]

Než však začneme s projektovou výukou, měli bychom si uvědomit, že ta předpokládá u žáků určité dovednosti, které je musíme nejprve naučit. Je to především dovednost diskutovat, řešit problém, dovednost pracovat ve skupině, argumentovat a prezentovat výsledky své práce. Proto nemůžeme začít realizovat projekt se žáky prvních tříd prvního stupně ZŠ.

Ideální projekt by měl vycházet z potřeb a zájmů žáků. Iniciátory projektu by tedy měli být sami žáci. Učitel může původní nápad žáků rozšířit o další činnosti, pomocí nichž si osvojí důležité kompetence. [7]

Projekt by měl obsahovat činnosti skupinové i individuální. Za jeho úspěšnou realizaci jsou odpovědni všichni žáci. Každý za nějaký konkrétní krok. Zachována by měla být i svoboda, kterou konkrétní činnost si žák v rámci projektu vybere a udělá. [7]

Výhodou výstupu z projektu je jeho hmatatelnost. Žáci vidí, že něco konkrétního vytvořili, což je motivuje k další práci. Žáci mají neustále na mysli cíl a vidí smysl své práce. Postupnými kroky se k cíli přibližují. Výstupem z projektu se žáci mohou pochlubit svým rodičům a kamarádům. Důležité je, aby každý žák mohl říci: „Tuto část projektu jsem udělal já.“ [7]

Shrnutí základních znaků projektového vyučování:

- Zaměření na výsledný produkt
- Žáci mohou ovlivňovat podobu projektu
- Aktivní zapojení všech žáků podle jejich schopností
- Kooperace a individualizace
- Závěrečná reflexe

1.1.1 Hodnocení projektu

Při hodnocení projektu se jedná o hodnocení celého procesu – naplánování projektu, jeho průběhu i výsledku a to z pohledu žáků i učitele. Hodnocení projektu by se mělo opírat o kritéria, která jsou s žáky vytvořena nebo jsou jim předložena. Z hodnocení by měla vyplynout opatření do budoucna, a to v rovině dítěte i učitele. [2]

1.2 Využití interaktivních stavebnic

Všeobecné technické vzdělání má za cíl naučit žáky plánovat a promýšlet technickou práci, řešit technické problémy, pracovat s jednoduchými nástroji a překonávat překážky technické povahy. Významným didaktickým prostředkem k dosažení většiny uvedených cílů jsou elektrotechnické stavebnice. [3]

Elektrotechnické stavebnice jsou moderními učebními pomůckami, které jsou využívány ve výuce především technické výchovy a fyziky na základních i speciálních školách či ve výuce nejrůznějších odborných teoretických i praktických elektrotechnických předmětů na

středních a vysokých školách. Kvalitní elektrotechnická stavebnice není jen obyčejnou učební pomůckou pro žáka nebo hrou pro dítě, ale podporuje i samostatné technické myšlení, tvořivost a představivost, logické usuzování, schopnost seberealizace či pomáhá vytvářet názor na abstraktní jevy, fyzikální veličiny atd. [3]

Uplatněním stavebnic lze vhodně rozvíjet technologické vědomosti žáků, a přitom je naučit technikám jednoduché experimentální práce. Tím se intelektualizuje praktické vyučování technických předmětů. Technologické vědomosti nejsou vhodné jen pro technicky zaměřené žáky, ale mohou přinést a rozvést praktické zkušenosti do běžného života. [3]

Elektrotechnické stavebnice jsou různorodé a vyskytují se v mnoha konstrukčních provedeních. Jelikož při práci se stavebnicí dochází k její interakci se žákem, lze tyto stavebnice rovněž označit jako interaktivní. [3]

1.3 Přehled interaktivních stavebnic

Existuje více použitelných interaktivních stavebnic. V oblasti robotiky a programování se jedná převážně o produkty od výrobců LEGO, Merkur a Arduino.

1.3.1 LEGO Mindstorms EV3

Stavebnice LEGO Mindstorms kombinuje programovatelné kostky, elektrické motory a senzory s LEGO kostkami a díly z řady Lego Technic. Je optimalizována pro práci ve třídě. Žáci a studenti v ní najdou vše potřebné k modelování, programování a testování reálných robotických zařízení. [11, 12]

Programování je založeno na principu ukládání programovacích ikon do řetězce příkazů. Grafické programování umožňuje uživatelům jednoduše nastavit prahové hodnoty pro senzory tak, aby např. při dosažení určité nastavené hodnoty (zjištěné senzorem) vydala kostka výstražný zvuk. [11,16]

Stavebnice LEGO Mindstorms EV3 je podrobněji popsána v kapitole č. 2.

1.3.2 Merkur

Stavebnice robota Mini Sumo GT12 je konstruována jako základní stavebnice robota pro získávání základních znalostí a poznatků při konstrukci robotických modelů. Tuto stavebnici je možné rozšiřovat o již připravené a připravované moduly jako například zvuková čidla, čidlo rozpoznávající barvu. Jedná se o otevřený systém (open source), který lze vyu-

žít s dalšími moduly (ať již zakoupenými či vlastní konstrukce) nebo ve spojení s dalšími systémy pro komplexnější aplikace. [9]



Obrázek 1: Stavebnice robota Mini Sumo GT12 – Merkur [9]

Řídící deska robota je již z výroby naprogramována základním kódem pro využívání robota v soutěžních kláních Mini Sumo. Funkce tohoto programu spočívá v orientaci robota v ringu (černý podklad s ohraničením bílou čarou), vyhledání protivníka a snahou o jeho vytlačení z ringu. Stavebnice je dodávána kompletně včetně motorů, řídicí jednotky a procesoru PICAXE – 20M2. [9]

V základní sestavě stavebnice je infračervený modul, který je již v programovém kódu využit, ale lze jeho funkci samozřejmě dále libovolným způsobem měnit (vhodnou změnou zdrojového kódu). Programování řídicí desky (procesoru) je velmi jednoduché, přes konektor na řídicí desce se speciálním kabelem (součástí balení) propojí s klasickým PC a pomocí návodu dodaném na CD se seznámíte s principem programování a naučíte se samostatně programovat další činnosti robota. [9]

1.3.3 Arduino

Arduino je počítač, který pomocí různých senzorů dokáže vnímat vnější svět a reagovat na něj například pohybem motorků a svícením LED. Zároveň je to otevřená elektronická platforma, založená na uživatelsky jednoduchém hardware a software. Lze se setkat s označením Genuino, což je obchodní značka pro produkty prodávané mimo Spojené státy americké. [10]



Obrázek 2: Základní deska stavebnice Arduino [10]

Hlavními výhodami Arduino platformy je jednoduchost použití, obrovské množství kompatibilního hardware, dobrá cena (kolem 650 Kč za Arduino Uno) oproti kitům a stavebnicím, ale hlavně podpora obrovské komunity Arduino nadšenců. Díky této celosvětové skupině je možné nalézt Arduino návody téměř na cokoliv, od použití Arduina jako toustovače, přes stavbu 3D tiskárny a robota až po jeho vyslání do vesmíru jako satelitu. [10]

1.4 Porovnání možností stavebnic

V následující tabulce jsou porovnány vlastnosti jednotlivých stavebnic z hlediska jejich využití pro výuku.

	Merkur	Arduino	LEGO Mindstorms EV3
Dostupnost přípojných modulů	nutno dokoupit	nutno dokoupit	součástí sady
Možnost programování pomocí grafických bloků	ne	ne	ano
Zaměřené i pro žáky základních škol	ano	ne	ano
Rozhraní pro připojení	-	USB	USB, Bluetooth
Možnost programování přes mobil / tablet	ne	ne	ano

Tabulka 1: Porovnání stavebnic z hlediska využití pro výuku [9,10,11]

2 LEGO MINDSTORMS EV3

Pro hlavní část této práce byla zvolena stavebnice LEGO Mindstorms EV3, která je ve škole k dispozici. Je to jedním z hlavních požadavků zadání této práce.

Stavebnice se jeví jeho nejvhodnější ze všech popsaných alternativ. Sestava LEGO Mindstorms EV3 je optimalizovaná pro práci ve třídě. Nabízí univerzální řešení a bohaté možnosti pro výuku robotiky, programování i zájmové kroužky.

Tvorba programu probíhá pomocí bloků, které na sebe navazují. Jazyk umožňuje sestavovat jednoduché programy, rozvíjet kompetence žáků a přecházet ke složitějším algoritmům. Díky jeho jednoduchosti a přehlednosti je vhodný i pro žáky základních škol.

2.1 Didaktické možnosti stavebnice

Stavebnice LEGO Mindstorms EV3 je optimalizována pro práci ve třídě. Žáci a studenti v ní najdou vše potřebné k modelování, programování a testování reálných robotických zařízení. [13]

Software dodávaný ke stavebnici je optimalizován pro školské prostředí, vychází z moderních požadavků, je velmi intuitivní a uživatelsky přátelský. Programování je založeno na principu ukládání programovacích ikon do řetězce příkazů. Grafický jazyk umožňuje sestavovat jednoduché programy, rozvíjet kompetence žáků a přecházet ke složitějším algoritmům. [13]

V souladu s požadavky na vzdělávání v 21. století si práce se stavebnicí klade za cíl podněcovat kreativní myšlení k řešení problémových úloh, vytváří návyky pro práci v týmu, rozvíjí komunikační dovednosti a vše potřebné pro úspěšnost ve škole i mimo ni. Žáci při práci zastávají role skutečných konstruktérů a vědců. [13]

2.2 Porovnání s dřívějšími verzemi

EV3 je třetí generací robotické stavebnice LEGO Mindstorms. První verze byla dána na trh v roce 1998. Nejnovější verze Lego Mindstorms EV3 pochází z roku 2013. [12]

Na obrázku je vidět proměna řídicí jednotky od verze RCX, přes NXT až po EV3. Největší změna byla mezi prvním a druhým modelem. Verze EV3 je v mnohém srovnatelná s NXT a je např. zpětně kompatibilní s jejími senzory. [8, 14]



Obrázek 3: Porovnání kostek ze stavebnic LEGO Mindstorms RCX, NXT a EV3 [8]

Po stránce výkonu a možností jednotlivých verzí udělalo LEGO za patnáctiletou evoluci veliký krok vpřed. Kromě několikanásobně rychlejšího procesoru a vyšší paměti nové verze disponují např. technologií Bluetooth. EV3 verze navíc podporuje Apple zařízení, lze připojit microSDHC karta a pomocí přídatného modulu je možné obohatit řídicí jednotku o WiFi. Podrobné srovnání je uvedeno v následující tabulce. [12, 14]

	RCX	NXT	EV3
Rok vydání	1998	2006	2013
Displej	segmentový	100 x 64 px	178 x 128 px
Procesor	Hitachi H8/300	Atmel 32-Bit ARM AT91SAM7S256	ARM9
Frekvence procesoru	16 MHz	48 MHz	300 MHz
Paměť RAM	32 kB	64 kB	64 MB
Úložná paměť	16 kB (ROM)	256 kB (Flash)	16 MB (Flash) + microSDHC slot
WiFi	ne	ne	ano (jako přídatný modul)
Bluetooth	ne	ano	ano
Možnost propojení se zařízeními Apple	ne	ne	ano

Tabulka 2: Srovnání verzí RCX, NXT a EV3 [12]

2.3 Popis stavebnice

Pro práci je k dispozici stavebnice LEGO Mindstorms Education EV3 s číselným označením 45544. Obsahuje celkem 541 dílů. Cena této stavebnice se na českém trhu pohybuje v rozmezí 11 000 – 12 000 Kč (údaj platný k dubnu 2016). [17]



Obrázek 4: Stavebnice LEGO Mindstorms Education EV3 [6]

2.3.1 Řídicí jednotka

EV3 kostka slouží jako řídicí centrum a napájecí stanice pro robota. Obsahuje

- 4 vstupní porty: 1, 2, 3, 4 (k připojení senzorů na EV3 kostku);
- 4 výstupní porty: A, B, C, D (K připojení motorů k EV3 kostce);
- rozhraní (1 mini USB PC port, USB hostitelský port, port microSDHC karty);
- vestavěný reproduktor. [11]



Obrázek 5: Řídicí jednotka – kostka [6]

Programovatelná, inteligentní EV3 kostka představuje mozek i srdce robotů. Je opatřena šesti barevně podsvícenými tlačítky, indikujícími stav kostky, a černobílým displejem s vysokým rozlišením. Díky internímu programovacímu prostředí umožňuje programování a záznam dat přímo na kostce. Jedná se o kompatibilní mobilní zařízení napájené AA bateriemi nebo EV3 nabíjecí DC baterií. [11]

Podrobnější technická specifikace je uvedena v následující tabulce.

Procesor	ARM 9, frekvence 300 MHz
Operační systém	Linux
Vstupní porty	4 (s možností získávání dat s frekvencí až 1000 vzorků/sekundu)
Výstupní porty	4
Interní paměť	Flash paměť 16 MB
Čtečka karet	mini SDHC karty do 32 GB
RAM	64 MB
Displej	černobílý s rozlišením 178 x 128 px
Rozhraní	USB, Bluetooth, WiFi
Napájení	6 AA baterií nebo akumulátor 2050 mAh

Tabulka 3: Technická specifikace EV3 kostky [11,12]

2.3.2 Servomotory

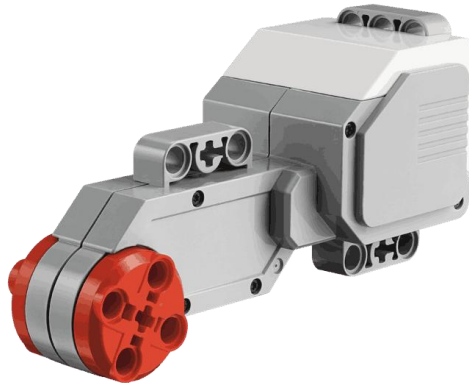
Velký motor

EV3 velký servomotor je výkonná pohonná jednotka využívající zpětnou vazbu z interního rotačního senzoru, což uživateli umožňuje ovládat otáčky motoru s přesností na jeden stupeň. Díky vestavěnému senzoru je možné programovat více motorů tak, aby se model pohyboval po zvolené trajektorii stejnou rychlostí. To je důležité například pro přesné načítání dat u experimentů. Servomotor obsahuje vlastní převodový mechanismus. [15]

Technická specifikace:

Otáčky motoru	160 – 170 ot/min
Krouticí moment	0,2 Nm
Maximální zátěž	0,4 Nm

Tabulka 4: Technická specifikace velkého motoru [15]



Obrázek 6: Velký motor [6]

Střední motor

Střední motor slouží k udržení přesnosti, přičemž určitý výkon se využívá k zachování kompaktní velikosti a rychlejších reakcí. EV3 střední servomotor je vhodný do modelů s menší zátěží, požadavkem na vyšší rychlost a rychlou odezvu. Motor je opatřen interním rotačním senzorem, který umožňuje jeho ovládání s přesností jednoho stupně. [11,15]

Technická specifikace:

Otáčky motoru	240 – 250 ot/min
Krouticí moment	0,08 Nm
Maximální zátěž	0,12 Nm

Tabulka 5: Technická specifikace středního motoru [15]



Obrázek 7: Střední motor [15]

2.3.3 Senzory

Dotykový senzor

Díky dotykovému senzoru může robot reagovat na dotyk. Analogový EV3 dotykový senzor svým čelním tlačítkem zjišťuje stisk či uvolnění, případně dokáže registrovat počet vícenásobných stisků a následných uvolnění. Umožňuje modely opatřit startovacími či stop

tlačítka, dotykovými čidly pro pohyb robotů v labyrintech, simulovat klávesy digitálních hudebních nástrojů, počítačové klávesnice, spínače kuchyňských spotřebičů aj. [11,15]



Obrázek 8: Dotykový senzor [15]

Ultrazvukový senzor

Digitální EV3 ultrazvukový senzor generuje svazek zvukových vln, načítá jejich odraz od vzdálených objektů a propočtem vzdálenost vyčíslí. Dokáže detekovat objekty (princip sonaru), případně čekat na příjem vln odražených anebo vyslaných z jiného senzoru. Díky senzoru je možné simulovat monitorování dopravy, dodržování vzdálenosti mezi vozidly aj. Umožňuje studium a testování zařízení, která známe z každodenního života, jako např. automatické otevírání dveří, parkovací senzory na autě nebo automatické výrobní linky. [11,15]

Technická specifikace:

Rozsah měření vzdálenosti	3 cm – 250 cm
Přesnost	± 1 cm

Tabulka 6: Technická specifikace ultrazvukového senzoru [15]



Obrázek 9: Ultrazvukový senzor [15]

Barevný senzor

Digitální EV3 senzor barvy rozlišuje osm různých barevných stavů (uvedených v tabulce níže). Může pracovat v režimu světelného senzoru, rozlišujícího intenzitu světla. V rámci školských projektů může být využit jako senzor třídící linky, pohybu robota po čáře, při experimentech s odrazivostí povrchů těles různých barev aj. Práci se senzorem získají žáci a studenti zkušenost s technologiemi uplatňovanými v rámci udržitelného rozvoje, v zemědělství nebo logistice. [11,15]

Technická specifikace:

Rozsah měření	od tmy až po přímé sluneční světlo
Rozlišení barev	barevná x černobílá; modrá, zelená, žlutá, červená, bílá a hnědá
Vzorkování	1 kHz

Tabulka 7: Technická specifikace barevného senzoru [15]



Obrázek 10: Barevný senzor [15]

Gyroskopický senzor

Digitální EV3 gyroskopický senzor měří rotační pohyb robota včetně změn v orientaci pohybu. Umožňuje záznam úhlu pohybu, udržení rovnovážné polohy, zkoumání působení síly na těleso a konstrukci reálných zařízení jako je Segway nebo např. herní ovladač. [11,15]

Technická specifikace

Přesnost měření úhlu	± 3 stupně
Maximální rychlost měření	440 stupňů / sekunda
Vzorkování	1 kHz

Tabulka 8: Technická specifikace gyroskopického senzoru [15]



Obrázek 11: Gyroskopický senzor [15]

2.4 Programování robotů

Existuje více způsobů, jak je možné výrobky ze stavebnice LEGO Mindstorms EV3 naprogramovat. Tato část kapitoly nabízí jejich základní přehled.

2.4.1 Programování pomocí aplikace LEGO MINDSTORMS EV3

Základní možnost, jak programovat roboty, je využití aplikace LEGO Mindstorms EV3 přímo od společnosti LEGO. Tato aplikace je ve verzi Home volně dostupná ke stažení z oficiálních webových stránek. Jedná se o programování pomocí grafických bloků, které se skládají za sebe a tvoří tím tok programu. [13]

Požadavky na hardware a software k programu jsou uvedeny v následující tabulce. Je vyžadováno, aby byl operační systém nainstalován včetně nejnovějších aktualizací. [13]

	Microsoft Windows	Apple Macintosh
Operační systém	Windows Vista (32/64 bitů) nebo novější	Mac OS 10.6 nebo novější
Procesor	dvoujádrový procesor s frekvencí 2 GHz nebo vyšší	
Paměť RAM	2 GB nebo větší	
Prostor na disku	2 GB	
Displej	alespoň XGA (1024 x 768)	
Rozhraní	1 volný USB port	

Tabulka 9: Technické požadavky aplikace LEGO MINDSTORMS EV3 [13]

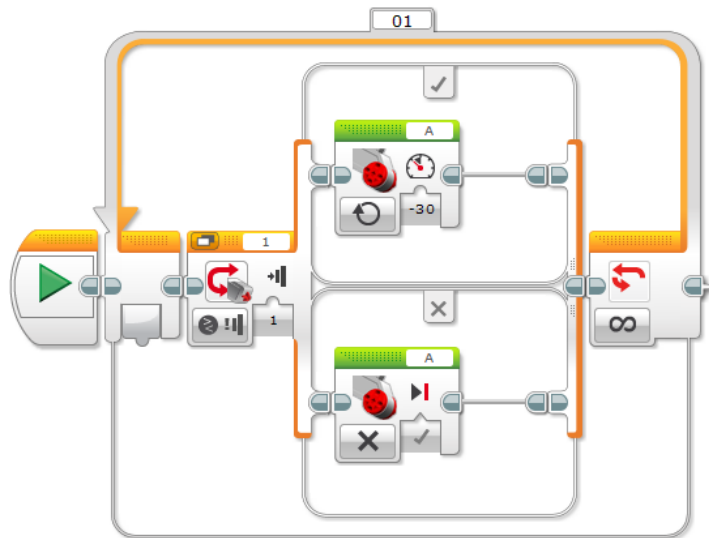
2.4.2 Programování pomocí mobilního telefonu nebo tabletu

Alternativou k programování robotů přes počítač je programování přes mobilní telefon nebo tablet. Je možné využít mobilní telefon nebo tablet s operačním systémem Android. EV3 již podporuje i Apple zařízení, takže lze využít i iPhone. [13]

Aplikace EV3 Programmer umožňuje programovat roboty stejným způsobem jako aplikace LEGO Mindstorms EV3 na počítači. Je optimalizována pro tablety a pro ovládání pomocí dotykového displeje. Je možné využít bloky akční, funkční a sensorové. Operační a pokročilé bloky nelze využít. [13,16]

Pomocí aplikace LEGO Mindstorms Commander je možné vytvořeného robota ovládat. Kromě ovládání přednastavených robotů lze nastavit i vlastní ovládání. Zvolí se ovládací prvek, např. horizontální nebo vertikální posuvník, tlačítko či joystick. Po nastavení pří-

slušných portů je možné takto ovládat řídicí jednotku. Aplikace využívá pro připojení technologii Bluetooth. [13,16]



Obrázek 12: Ukázka jednoduchého programu v LEGO Mindstorms EV3

2.4.3 Programování v jiných programovacích jazycích

Pro pokročilejší programátory existuje možnost programovat i v programovacích jazycích jako je Java nebo C. Tato možnost není k dispozici přímo od společnosti LEGO, je nutné využít software třetích stran. Alternativní vývojové prostředí nabízí LeJOS nebo Brick Command Center. [18,19]

II. PRAKTICKÁ ČÁST

3 POPIS PROJEKTU

V praktické části byl navržen projekt, ve kterém se žáci seznámí se stavebnicí LEGO Mindstorms a pochopí základní principy a možnosti této stavebnice.

3.1 Cílová skupina

Projekt je navržený pro kroužek robotiky a programování na druhém stupni základní školy. Jde o žáky ve věku 11 – 15 let. Jelikož jde o začátečníky, kteří se většinou setkají s programováním i stavebnicí LEGO Mindstorms poprvé, jsou úlohy sestavené od těch nejjednodušších. Ke složitějším zadáním se žáci dostanou po malých krůčcích. Komentovaná řešení vysvětlují postup krok po kroku a slouží jako vodítko pro učitele i pro žáky.

Úlohy lze aplikovat i na středních školách jako část výuky programování. V tomto případě je na učiteli, aby přizpůsobil řešení projektu požadavkům RVP a ŠVP.

3.2 Časová náročnost

Předpokládá se, že tento kroužek bude probíhat jednu hodinu týdně. U jednotlivých úloh není přesně vymezený čas na realizaci. Rychlost práce závisí na věku a šikovnosti žáků. Čas by si měl rozvrhnout učitel kroužku v závislosti na časové dotaci kroužku.

3.3 Využití pomůcek

Při práci na projektu budou žáci využívat tyto pomůcky:

- Stavebnice LEGO Mindstorms EV3
- Počítač
- Mobilní telefon se systémem Android nebo iPhone

3.4 Práce ve skupinách

Stavebnice LEGO Mindstorms nejsou levnou záležitostí, proto nelze předpokládat, že jich škola bude mít vždy dostatek. Projekt je postavený tak, aby žáci mohli pracovat ve skupinách. Každá skupina by měla mít 2-5 žáků v závislosti na dostupnosti stavebnic.

3.5 Cíl projektu

Během projektu se žáci naučí a pochopí základní principy programování, pochopí základy algoritmizace, a způsobem krok po kroku si osvojí metody práce se stavebnicí.

Motivací je pro žáky reálný model, který musí sami sestavit a naprogramovat. Mají dané jasné požadavky, ale samotné řešení je na nich, kreativitě se meze nekladou.

3.6 Ukončení projektu

Po ukončení projektu si žáci připraví prezentaci a výsledky svojí práce představí vedoucímu kroužku a zároveň ostatním skupinám. Prezentace bude obsahovat sebehodnocení žáků jednotlivých skupin. Zároveň proběhne zhodnocení celého projektu formou diskuse s vedoucím kroužku.

3.7 Navazující aktivity

Po úspěšném absolvování všech úloh si žáci osvojí metody programování a práce se stavebnicí LEGO Mindstorms. Otevírá jim to dveře k dalším aktivitám spojeným s touto stavebnicí. Žáci si mohou vymyslet a sestavit vlastní model inspirovaný reálným životem, například vysokozdvihný vozík nebo třídící linku ve výrobě. Žáci se zkušenostmi s programováním si mohou vyzkoušet naprogramovat model v programovacím jazyku jako je C nebo Java.

3.8 Struktura úloh projektu

Projekt je rozdělen na celkem 7 úloh od mechanického sestavení až po jeho pokročilé programování. Jednotlivé úlohy v tomto projektu jsou postaveny tak, že na sebe navzájem navazují, a díky tomu dává žákům celý projekt větší smysl. U každé úlohy je vypracováno obecné řešení.

4 POPIS PROGRAMOVACÍHO JAZYKA

Pro tvorbu úloh byl využit programovací jazyk EV3-G, který je součástí stavebnice LEGO MINDSTORMS. Jedná se o grafický programovací jazyk s vyšší mírou abstrakce. Díky jeho jednoduchosti a přehlednosti je vhodný i pro žáky základních škol. Tvorba programu probíhá pomocí bloků, které na sebe navazují. Při spuštění programu je možné sledovat, v jaké části se program nachází, a mít tak přehled o jeho funkčnosti a správnosti.

Programování je založeno na principu ukládání programovacích ikon do řetězce příkazů. Jazyk umožňuje sestavovat jednoduché programy, rozvíjet kompetence žáků a přecházet ke složitějším algoritmům.

Grafické programování umožňuje uživatelům jednoduše nastavit prahové hodnoty pro senzory tak, aby např. při dosažení určité nastavené hodnoty (zjištěné senzorem) vydala kostka výstražný zvuk.

4.1 Vývojové prostředí

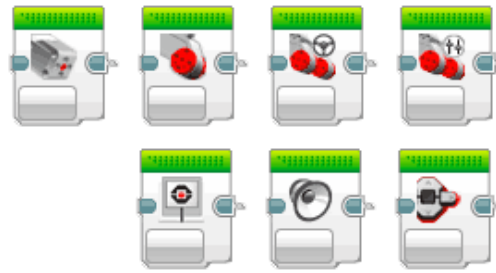
Práce probíhala ve vývojovém prostředí LEGO MINDSTORMS EV3 Home Edition. Tato verze je volně dostupná na internetu a pro účely této práce splňuje všechny požadavky. Přestože má škola licenci na LEGO MINDSTORMS EV3 Education, pro účely této práce byla plně vyhovující i základní verze. Placená verze má navíc pokročilý editor obsahu a možnost zaznamenávání dat.

Okno vývojového prostředí obsahuje prostor pro tvorbu programu a přednastavený startovací blok. V dolní části je možné vybírat požadované bloky, které jsou seskupené do kategorií označených barvami. V pravé dolní části je možné sledovat propojení s EV3 kostkou (řídící jednotkou). Pomocí tlačítek lze nahrávat vytvořené programy do kostky a spouštět je. Lze také sledovat aktuální hodnoty senzorů připojených ke kostce.

4.2 Programovací bloky

4.2.1 Akční bloky

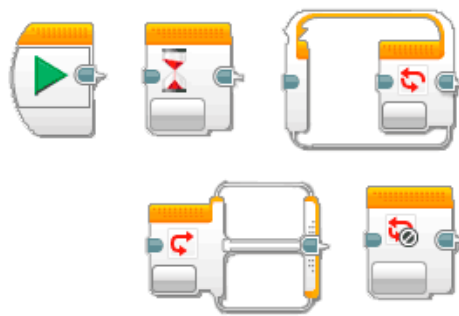
Akční bloky ovládají akce programu. Ovládají nejen otáčky motorů, ale také displej EV3 kostky. Pomocí těchto bloků se nastavuje i výstup na displej, reproduktor a podsvícení tlačítek na kostce.



Obrázek 13: Akční bloky

4.2.2 Funkční bloky

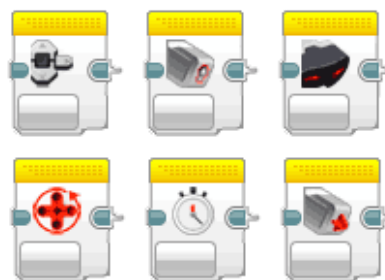
Funkční bloky ovládají funkci programu. Všechny vytvořené programy začnou se startovním blokem. Obsahují blok pro čekání, který lze nastavit na určitý čas nebo na jakýkoliv jiný moment, např. informace ze senzoru. Mezi funkční bloky je zařazena i smyčka pro opakování programu, rozhodovací prvek a blok pro opuštění smyčky.



Obrázek 14: Funkční bloky

4.2.3 Senzorové bloky

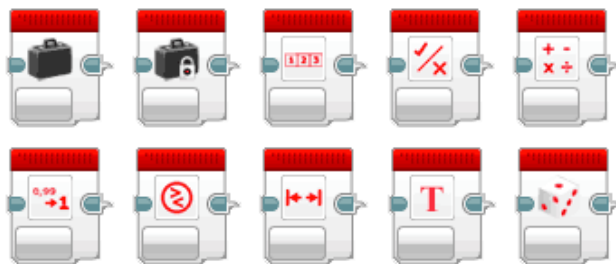
Senzorové bloky umožňují vytvářeným programům číst informace přicházející z barevných senzorů, IČ senzoru, dotykového senzoru a mnohých dalších. Takto přečtené informace lze využít k nastavení parametrů u dalších bloků nebo je např. uložit do proměnných. Mezi tyto bloky je zařazen i časovač.



Obrázek 15: Senzorové bloky

4.2.4 Operační bloky

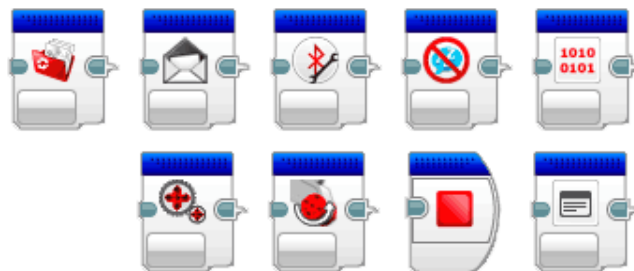
Operační bloky žákům umožňují pracovat s proměnnými. Lze je vytvářet, zapisovat do nich a číst jejich obsah. Pomocí operačních bloků je také možné porovnávat hodnoty, zaokrouhlovat nebo např. generovat náhodná čísla. Červené datové bloky jsou dostupné pouze pro programovací software pro PC/Mac a ne pro aplikaci EV3 Programmer, která je dostupná pro telefony a tablety.



Obrázek 16: Operační bloky

4.2.5 Pokročilé bloky

Pokročilé bloky umožňují spravovat soubory, ovládat připojení k Bluetooth a mnohé další věci. Modré pokročilé bloky jsou dostupné pouze pro programovací software pro PC/Mac a ne pro aplikaci EV3 Programmer.



Obrázek 17: Pokročilé bloky

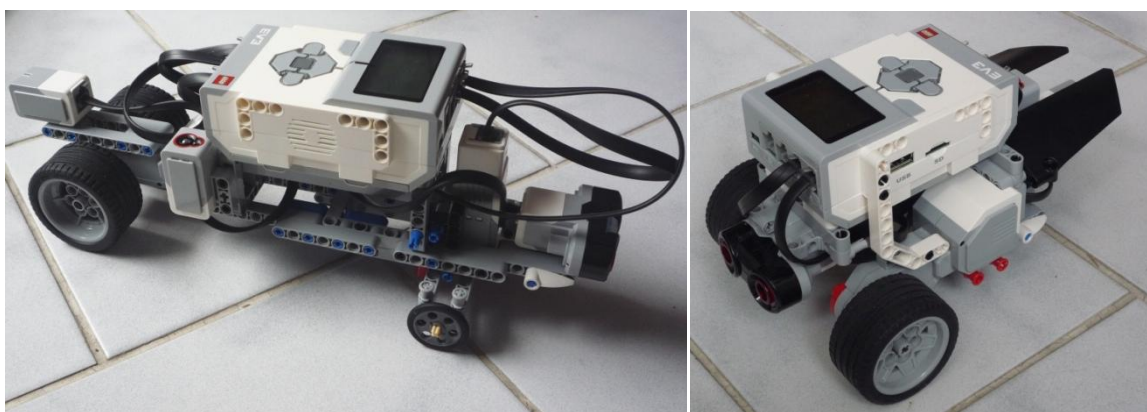
5 PROJEKT: ROBOTICKÉ VOZÍTKO

Tématem navrhnutého projektu (jehož návrh je popsán v kapitole č. 3) je robotické vozítko. Jde o předmět, který nabízí rozsáhlé možnosti programování. Lze na něm postupně realizovat široké spektrum úloh - od nejjednodušších až po ty náročné. Inspirací pro vozítko je reálný svět, kde může mít uplatnění jako dopravní prostředek, pracovní nástroj v průmyslu, průzkumník terénu apod.

Prostřednictvím projektu se žáci seznámí se stavebnicí LEGO Mindstorms EV3, mechanicky si vozítko sami sestaví a naučí se pracovat s motory a senzory. Formou postupných úkolů pochopí základní principy programování. Vytvořené programy budou na vozítku i sami testovat a ověřovat jejich správnou funkčnost.

V závěrečné fázi projektu se žáci dostanou i k pokročilejším úlohám, kde si vyzkouší naprogramovat vozítko tak, aby se chovalo inteligentně (např. reagovalo na překážku) nebo provádělo složitější úkony.

Hmatatelným výstupem projektu je robotické vozítko, které může mít mnoho podob v závislosti na fantazii a kreativitě žáků. Dva příklady, jak může vozítko vypadat, jsou zobrazeny na následujících fotografiích. Hlavní rozdíl mezi těmito modely je způsob konstrukce nápravy řízení. V prvním případě je zatáčení řešeno středním servomotorem, který pohybuje nápravou. Druhá verze vozítka je řešena dvěma poháněcími velkými servomotory a zatáčení je docíleno správným nastavením motorů.



Obrázek 18: Možné podoby výstupu projektu

5.1 ÚLOHA 1: Sestavení vozítka

5.1.1 Zadání

Pomocí stavebnice LEGO Mindstorms EV3 sestavte vozítko, které bude schopné jezdit dopředu a dozadu. Využijte servomotor ze stavebnice. K vozítku pevně připevněte i řídicí jednotku a správně jí propojte s motorem.

5.1.2 Komentované řešení

Cílem práce je vytvořit vozítko. Musí být sestaveno tak, aby byla konstrukce dostatečně pevná a odolná. Řídicí jednotku je vhodné umístit do vodorovné polohy, aby byla zachována dostatečná stabilita a nízké těžiště vozítka. Kola jsou napojena přímo na motor, tedy bez převodu. Lze sestavit i pohon s využitím převodu. Dostupná stavebnice nabízí pouze jeden pár větších kol, proto jsou pro druhou nápravu využita malá kolečka. Není vyžadováno, aby vozítko bylo schopné zatáčet, takže zatáčení nebudeme řešit.

Pro pohon vozítka zvolíme velký servomotor. Pro tento účel je vhodnější než střední servomotor z důvodu vyššího točivého momentu a výhodnější konstrukce.

Servomotor je pomocí kabelu propojen s řídicí jednotkou. Je možné využít porty A-D, což jsou výstupní porty. V tomto případě byl zvolen výstupní port A. V této úloze není vyžadováno připojení žádných dalších prvků kromě motoru, takže porty 1-4 sloužící pro senzory zůstanou nezapojené.

Vytvořené vozítko může mít mnoho podob, jedna z nich je na následujících obrázcích.



Obrázek 19: Konstrukce podvozku vozítka a montáž řídicí jednotky

5.2 ÚLOHA 2: Ovládání vozítka

5.2.1 Zadání

Vyzkoušejte ovládání vozítka přímo z řídicí jednotky EV3. Dále pomocí bluetooth propojte řídicí jednotku s mobilním telefonem či tabletem se systémem Android (lze využít i iPhone nebo iPad). Ovládejte vozítko pomocí telefonu.

5.2.2 Komentované řešení

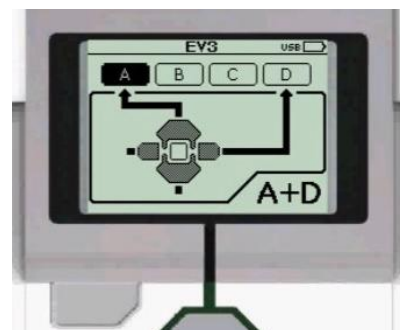
Sestavené vozítko je zatím schopné jezdit jen dopředu a dozadu. Cílem této úlohy je rozpohybovat motor a vyzkoušet si ovládání přímo z řídicí jednotky a také pomocí bluetooth.

Jednou ze schopností řídicí jednotky EV3 je ruční ovládání motorů připojených na výstupních portech A-D. Po zapnutí jednotky je tato možnost dostupná ve třetí záložce – volba „Motor Control“.



Obrázek 20: Volba ručního ovládání motorů

Pomocí tlačítek nahoru a dolů lze ovládat motor připojený na portu A. Pomocí tlačítek doleva a doprava lze ovládat motor B, který momentálně není zapojený. Motor je aktivní, jen když se tlačítko stiskne a drží.



Obrázek 21: Ruční ovládání motorů pomocí řídicí jednotky

Pro ovládání vozítka je tato varianta poněkud nepraktická. Další možností přímého ovládání je ovládání s využitím mobilního telefonu či tabletu s operačním systémem Android (lze

využit i iPhone nebo iPad). Pro následující ukázkou byl využit mobilní telefon Samsung Galaxy S3 Mini s operačním systémem Android.

Přes Obchod Play do telefonu stáhneme aplikaci LEGO Mindstorms Commander.



Obrázek 22: Instalace aplikace LEGO Mindstorms Commander

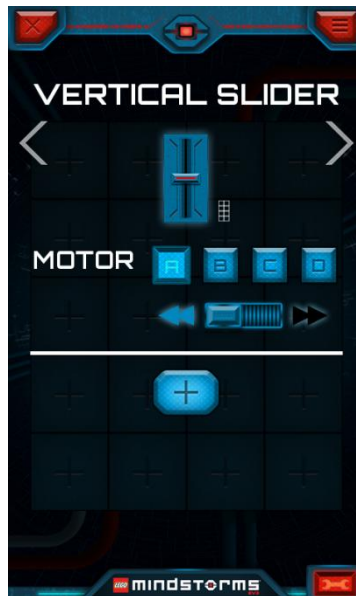
Na řídicí jednotce EV3 v nastavení zkontrolujeme, zda je zapnutý bluetooth. Spustíme aplikaci v mobilním telefonu a v hlavním menu vybereme možnost *Create & command your own robot*. Proběhne propojení telefonu s jednotkou EV3 a můžeme vytvořit vlastní program.



Obrázek 23: Instalace aplikace LEGO Mindstorms Commander

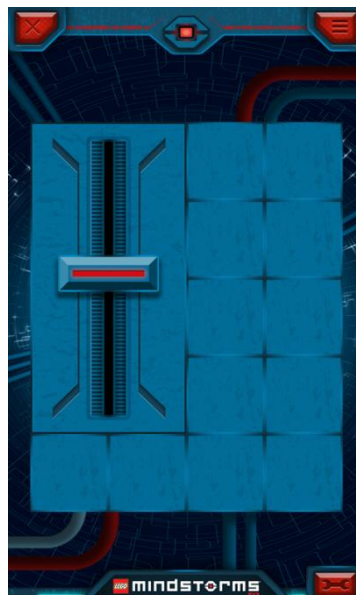
Pomocí znaku + ve čtvercové síti vložíme ovládací prvek. K dispozici jsou Joystick, Horizontal slider, Vertical Slider a Button. V našem případě se pro ovládání motoru nejlépe hodí slider. Rozdíl mezi horizontálním a vertikálním je pouze v umístění

ovládacího prvku na displeji, na funkčnost to nemá vliv. V nastavení slideru zvolíme motor A a můžeme změnit i směr otáčení.



Obrázek 24: Vložení prvku do vlastního programu

Po vložení prvku do čtvercové sítě máme možnost přidat další ovládací prvky, nebo ovládání spustit pomocí symbolu klíče v pravém dolním rohu.



Obrázek 25: Ovládání vozítka pomocí mobilního telefonu

5.3 ÚLOHA 3: První programování

5.3.1 Zadání

Seznamte se s prostředím programovacího softwaru Mindstorms EV3. Vytvořte programy, které budou splňovat následující požadavky:

Program A: Vozítko pojede plnou rychlostí vpřed po dobu tří sekund a poté se zastaví.

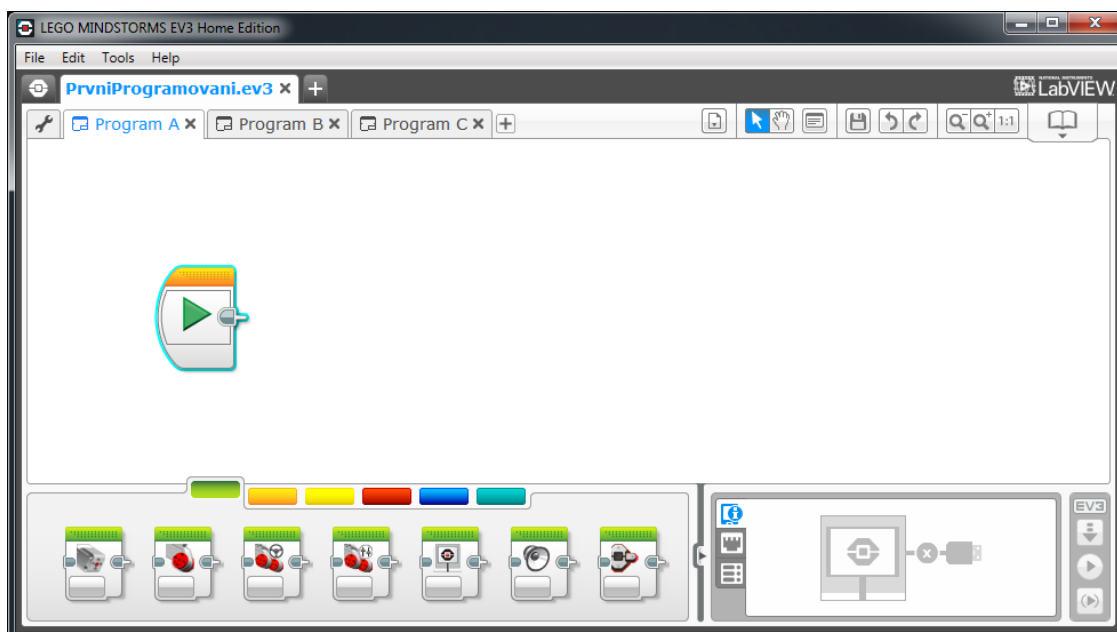
Program B: Vozítko pojede poloviční rychlostí vpřed po dobu dvou sekund, potom dvě sekundy počká a nakonec popojede zpět o tři otáčky motoru stejnou rychlostí.

Program C: Vozítko pojede libovolnou rychlostí vpřed po dobu dvou vteřin, poté zastaví bez brzdění motoru, počká jednu vteřinu, vrátí se zpět na původní místo, opět počká jednu vteřinu a takto se bude celý cyklus opakovat 5x. V programu využijte vhodný funkční blok pro opakování.

Propojte řídicí jednotku s počítačem, vytvořené programy do ní nahrajte a ověřte jejich funkčnost.

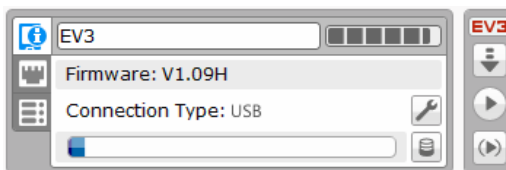
5.3.2 Komentované řešení

Po spuštění programu LEGO MINDSTORMS EV3 zvolíme nahoře v menu položku *File – New Project*. Tím se vytvoří nový prázdný projekt, který si pomocí *File – Save Project as* uložíme např. pod názvem *PrvniProgramovani*. Dvojklikem na první záložku *Program* ji můžeme přejmenovat na *Program A* a zároveň si nachystat i záložky pro program B a C.



Obrázek 26: Prostředí programu LEGO MINDSTORMS EV3

Po připojení řídicí jednotky k počítači a jejím zapnutí ji program rozpozná a v pravé spodní části ukazuje její stav. Jednotka EV3 je připojena pomocí kabelu USB – micro USB.



Obrázek 27: Informace o připojení řídicí jednotky

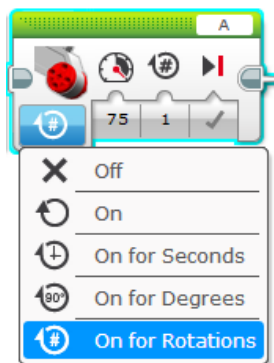
Programování pomocí tohoto softwaru funguje na principu skládání jednotlivých bloků. V tomto případě budeme využívat zelený akční blok pro ovládání motoru a oranžový funkční blok pro časovou prodlevu. V každém novém projektu je nachystaný startovací blok se zelenou šipkou, kterou je možné program spustit.

Řešení programu A:

Pro ovládání motoru slouží zelený akční blok s obrázkem motoru. V pravé horní části je přednastaven port A, ke kterému je motor připojený. Pro ovládání jiného motoru lze kliknutím port změnit. Pod obrázkem motoru je tlačítko s výběrem akce. Volby jsou:

- Off – zastavení motoru
- On – spuštění motoru
- On for Seconds – spuštění motoru na určitý počet sekund
- On for Degrees – spuštění motoru pro otočení o konkrétní úhel
- On for Rotations – spuštění motoru se stanoveným počtem otáček

Pro každou z těchto voleb se zobrazí odlišná nabídka parametrů. Vždy se nastavuje výkon v intervalu -100 až 100 (záporná hodnota spustí motor v opačném směru). V závislosti na nastaveném způsobu spouštění se nastavuje čas, stupně nebo otáčky. Poslední parametr umožňuje zapnout či vypnout zabrzdění při zastavení motoru.



Obrázek 28: Ovládací panel velkého motoru

Řešení programu A spočívá v nastavení jednoho akčního bloku velkého motoru. Musí být nastaveno spuštění na určitý čas a tento čas nastavit na 3 sekundy. Výkon nastavíme na hodnotu -100. Vozítko kvůli konstrukci jede vpřed tehdy, pokud je výkon záporný.



Obrázek 29: První programování - řešení programu A

Řešení programu B

Oproti programu A nejde o jeden příkaz (akci), ale o určitou sekvenci. Sekvence proběhne jen jednou. Bloky musíme seřadit za sebe. Pro časovou prodlevu využijeme oranžový funkční blok se symbolem přesýpacích hodin. Řešení programu je na následujícím obrázku:



Obrázek 30: První programování - řešení programu B

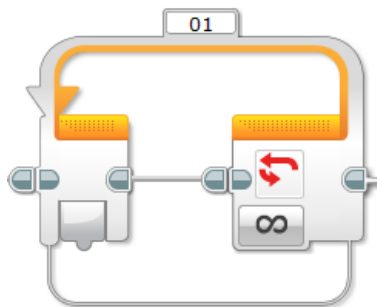
Na obou motorech nastavíme poloviční výkon, tedy 50. V prvním bude hodnota záporná, aby se vozítko pohybovalo vpřed. Na časové prodlevě nastavíme počet vteřin na 2. Ve třetím bloku vybereme možnost spuštění *On for Rotations* a nastavíme 3 otáčky.

Program můžeme spustit a sledovat pohyb vozítka. Zjistíme, že tři otáčky při polovičním výkonu zhruba odpovídají ujeté vzdálenosti při pohybu motoru po dobu dvou sekund. Vozítko se tedy vrátí zhruba na stejné místo, odkud startovalo.

Řešení programu C

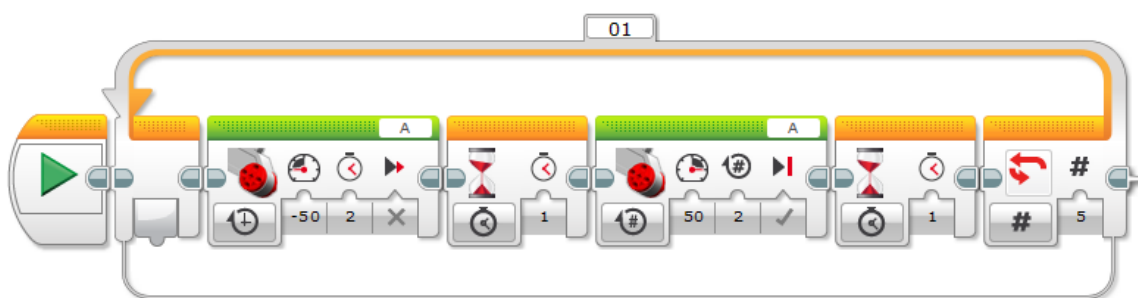
Úkolem je vytvořit sekvenci, která se bude pětkrát opakovat. Na základě znalostí z předchozích úkolů vytvoříme a otestujeme požadovanou sekvenci.

Pro opakování využijeme smyčku z řady oranžových funkčních bloků. Prvky, které vložíme dovnitř, se budou opakovat. Ve výchozím nastavení je smyčka nastavena na nekonečné opakování (Unlimited). Tuto možnost změňme na konkrétní počet opakování (Count). Ukončení smyčky je možné nastavit mnoha způsoby, například na základě senzorů, informací o otáčkách motorů nebo lze nastavit opakování po určitý časový limit.



Obrázek 31: Smyčka pro opakování programu

Požadovanou sekvenci vložíme do smyčky a program otestujeme. Řešení programu C je na následujícím obrázku.



Obrázek 32: První programování - řešení programu C

5.4 ÚLOHA 4: Senzory

5.4.1 Zadání

Na přední i zadní část vozítka doplňte senzory. Využijte dotykový a ultrazvukový senzor. Vytvořte programy, které budou splňovat následující požadavky:

Program A: Vozítko se rozjede po krátkém stisknutí dotykového senzoru a pojedje vpřed po dobu dvou sekund.

Program B: Po krátkém stisknutí dotykového senzoru vozítko počká jednu sekundu a rozjede se vpřed. Zastaví se až při opětovném krátkém stisknutí dotykového senzoru.

Program C: Vozítko pojedje vpřed jen při stisknutém dotykovém senzoru.

Program D: Po spuštění programu se vozítko rozjede vpřed rychlostí 20% z maxima. Při stisknutém dotykovém senzoru bude vozítko couvat. Když senzor nebude stisknutý, vozítko pojedje opět vpřed (bez časových prodlev). Program se ukončí po stisknutí prostředního tlačítka na řídicí jednotce.

Program E: Vozítko se po spuštění programu rozjede vpřed a ve vzdálenosti 5 cm před překážkou se na jednu sekundu zastaví. Potom se rozjede v opačném směru a pojede do té doby, než narazí na překážku. Poté jednu sekundu počká a celý cyklus se bude opakovat.

5.4.2 Komentované řešení

Na přední i zadní část vozítka mechanicky připevníme senzory. Na jednu stranu umístíme dotykový senzor, na druhou stranu ultrazvukový senzor.

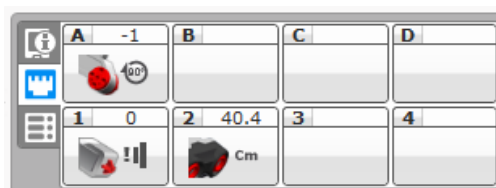


Obrázek 33: Vozítko doplněné o senzory

Senzory propojíme pomocí kabelů s řídicí jednotkou. Využijeme porty označené čísly 1-4. Pro pořádek si můžeme zvolit:

- Port 1: Dotykový senzor
- Port 2: Ultrazvukový senzor

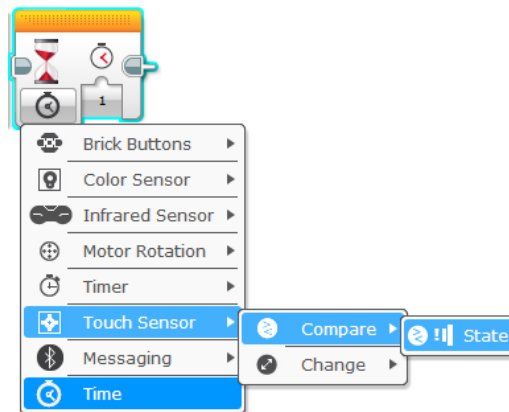
Pokud máme řídicí jednotku zapnutou a propojenou s počítačem, můžeme zkontrolovat stav připojení senzorů přes aplikaci LEGO MINDSTORMS EV3. V pravé dolní části lze pomocí prostředního malého čtvercového tlačítka přepnout zobrazení a vidíme připojené motory i senzory. Zároveň je zobrazena i aktuální hodnota stavu senzoru.



Obrázek 34: Připojené akční bloky a senzory

Řešení programu A

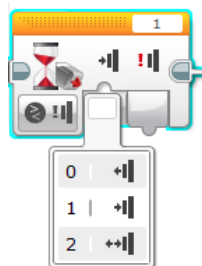
Cílem je, aby se vozítko rozjelo po krátkém stisknutí dotykového senzoru. Využijeme oranžový funkční blok pro časovou prodlevu, který lze přepnout na čekání na určitý stav senzoru. Pro dotykový senzor nastavíme hodnotu *Touch Sensor – Compare – State* dle obrázku.



Obrázek 35: Nastavení pro senzor

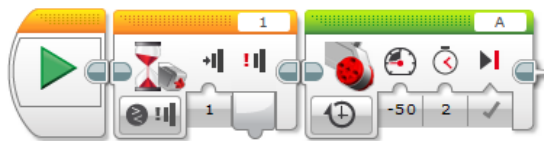
Jakmile se běžící program dostane k tomuto bloku, bude čekat, než se senzor dostane do stavu 1 (zmáčknuto). U dotykového senzoru lze nastavit akci při těchto událostech:

- 0 – uvolnění senzoru
- 1 – stlačení senzoru
- 2 - stlačení a uvolnění senzoru



Obrázek 36: Události dotykového senzoru

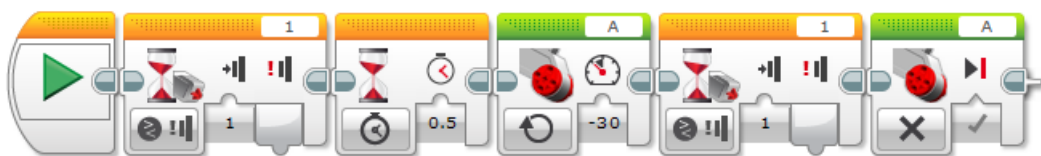
V našem případě nastavíme dotykový senzor na pouhé stlačení, tedy hodnotu 1. Jako následující prvek umístíme akční prvek pro ovládání motoru s nastavenými požadovanými parametry. Po spuštění bude program vyčkávat na událost stlačení dotykového senzoru. Až nastane stlačení, program bude pokračovat dalším blokem, kde dojde ke spuštění motoru. Řešení programu je následující:



Obrázek 37: Senzory - řešení programu A

Řešení programu B

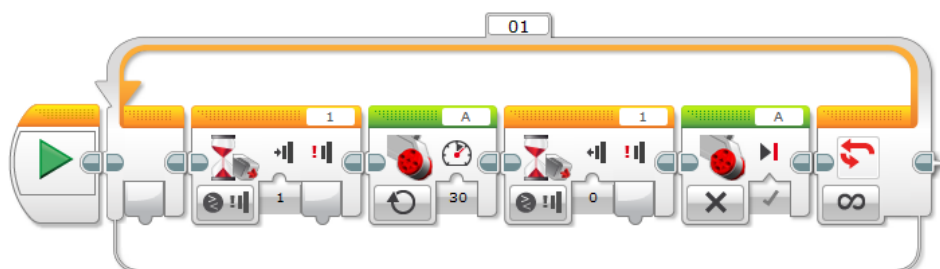
Oproti programu A musíme vytvořit delší sekvenci. Začne stejně – čekáním na stisknutí dotykový senzor. Jako další blok umístíme časovou prodlevu 0,5 sekundy (dle zadání). Po prodlevě je potřeba zapnout motor, typ spuštění tentokrát nastavíme pouze na *On*. Nastavíme jen jeho výkon. Motor poběží do té doby, než jej ručně zastavíme v další části programu. Zastavení nastane až po opětovném stisknutí dotykového senzoru. Takže následuje oranžový blok čekající na stisknutí dotykového senzoru a poté zastavíme motor přidáním prvku pro ovládání motoru a jeho nastavením do fáze *Off*.



Obrázek 38: Senzory - řešení programu B

Řešení programu C

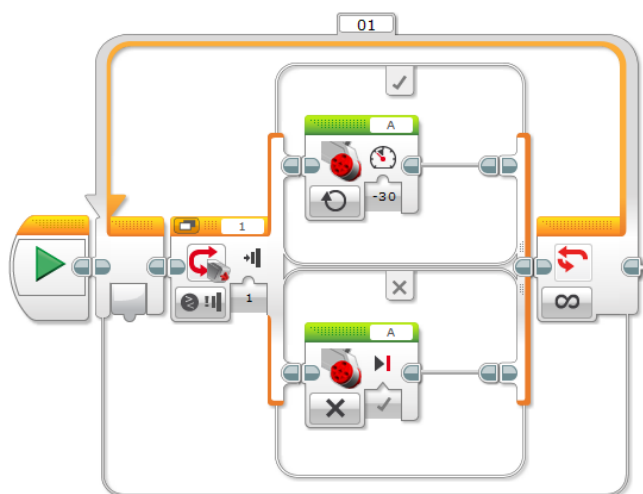
Vozítko musí jet jen v případě, že je stisknutý dotykový senzor. Využijeme možnosti nastavení události při čekání na stisknutí dotykového senzoru. Nejdříve budeme čekat na stisknutí tlačítka, potom zapneme motor a následně budeme čekat na uvolnění tlačítka. Po uvolnění tlačítka motor zastavíme. Celá tato sekvence se bude opakovat v cyklu a tím docílíme požadované funkčnosti. Po přechodu do dalšího cyklu se bude znovu čekat na stisknutí tlačítka dotykového senzoru. První možnost řešení tohoto programu je následující:



Obrázek 39: Senzory - řešení programu C

Alternativním řešením úlohy může být i využití rozhodovacího prvku *Switch*. Díky tomuto prvku je možné otestovat libovolnou hodnotu a na základě toho rozhodnout, co se bude dít. Horní větev se provede, pokud bude zadaná podmínka splněna. V opačném případě, pokud nebude splněna, se provede spodní větev.

Program běží neustále v nekonečné smyčce a v každém okamžiku testuje stav dotykového senzoru. Pokud je jeho hodnota 1 (je stlačený), dá se signál motoru ke spuštění. Pokud je hodnota senzoru jiná než 1 (je uvolněný), dá se signál motoru k zastavení. Algoritmus nemá žádné nastavené časové prodlevy (čekání), testování tedy probíhá neustále.

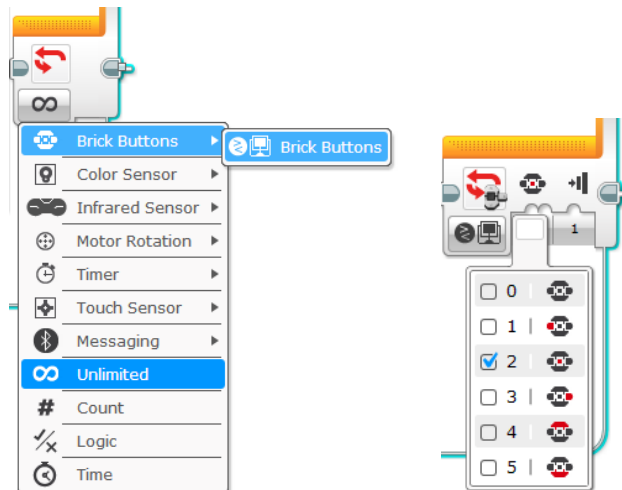


Obrázek 40: Senzory - alternativní řešení programu C

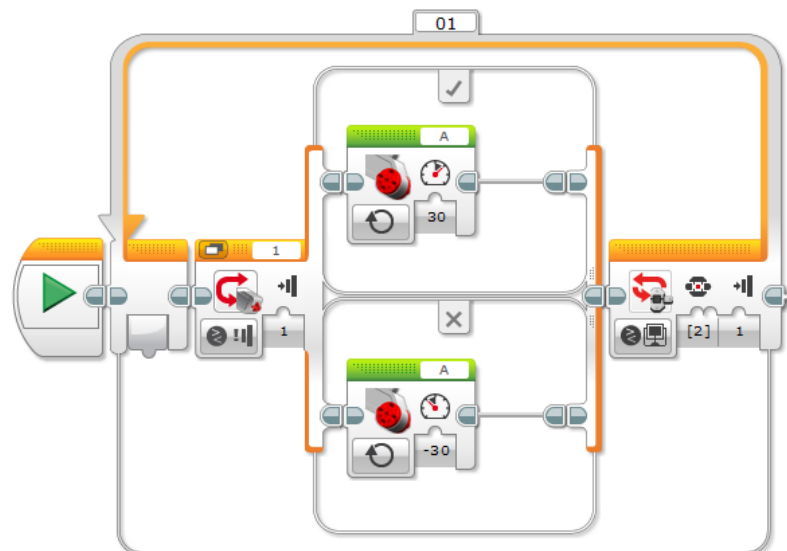
Řešení programu D

Program D je velice podobný programu C. Existují rovněž dvě varianty řešení, jako u předchozí úlohy. Řešení s využitím rozhodovacího prvku lze považovat za přehlednější z hlediska pochopení funkčnosti programu na první pohled. Podmínka pro rozhodování je stejná, testuje se stlačení dotykového senzoru. Liší se druhá (záporná) větev. Ta neprovádí zastavení motoru, ale spouští ho v opačném směru.

Další změnou je změna nastavené podmínky opakování smyčky (cyklu). Smyčka se bude opakovat do té doby, než bude stisknuto tlačítko na řídicí jednotce. To nastavíme změnou podmínky z *Unlimited* na *Brick Buttons – Brick Buttons*. Zkontrolujeme, že je vybráno tlačítko s číslem 2 (prostřední tlačítko na řídicí jednotce).



Obrázek 41: Nastavení ukončení programu při stisknutí tlačítka na jednotce



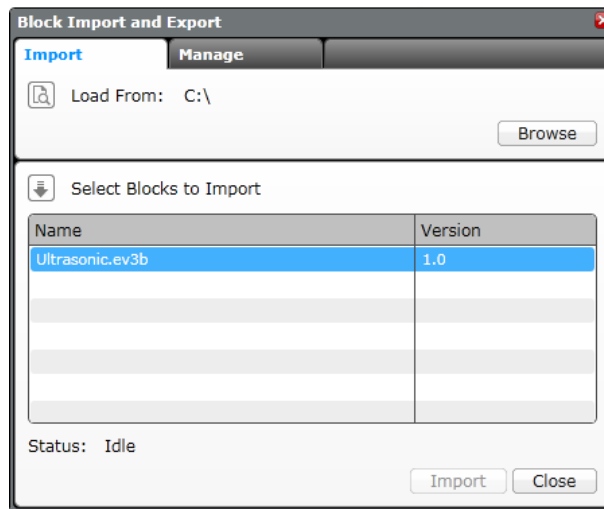
Obrázek 42: Senzory - řešení programu D

Řešení programu E

Cílem programu E je vytvořit sekvenci, aby vozítko jezdilo tam a zpět vždy do té doby, než narazí na překážku.

V tomto úkolu budeme využívat ultrazvukový senzor. Pokud ho v nabídce senzorů zatím nemáme, musíme ho tam přidat. Nejprve stáhneme potřebný soubor z oficiálního webu LEGO MINDSTORMS EV3 na adrese <http://www.lego.com/cs-cz/mindstorms/downloads> v sekci *Stážení softwaru EV3 (PC/MAC)*. Dále ho musíme nahrát do programu. Toho docílíme přes horní menu a položky *Tools – Block Import*.

Zobrazí se okno, v němž vybereme stažený soubor a potvrdíme tlačítkem Import. Po provedení importu je vyžadován restart programu LEGO MINDSTORMS EV3.

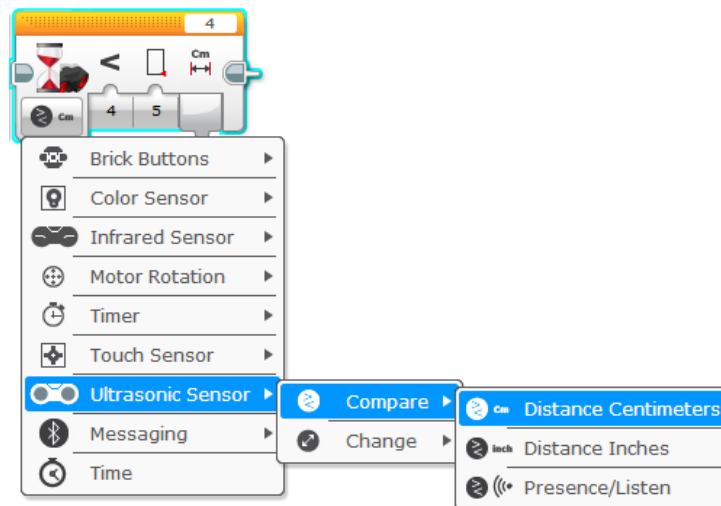


Obrázek 43: Import ultrazvukového senzoru do programu

Po úspěšném importu nyní v nabídce senzorů vidíme i ultrazvukový senzor. Využijeme ho pro měření vzdálenosti, prvek tedy nastavíme na *Ultrasonic Sensor – Compare – Distance Centimeters*. Můžeme si vybrat jeden z následujících způsobů porovnávání vzdálenosti:

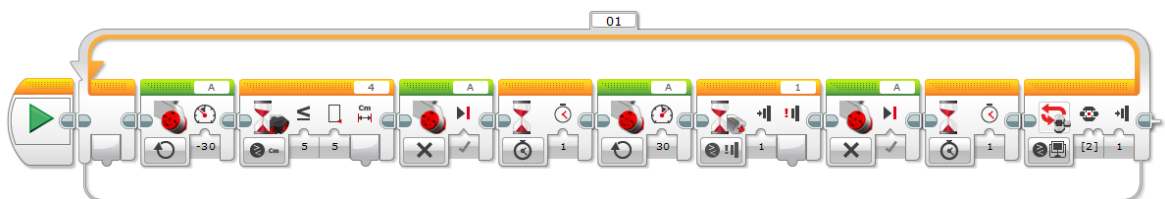
- 0 – je rovno
- 1 – není rovno
- 2 – je větší než
- 3 – je větší nebo rovno než
- 4 – je menší než
- 5 – je menší nebo rovno než

V dalším parametru zadáme hodnotu, která se bude porovnávat. V tomto případě nastavíme podmínku, která bude ověřovat, zda naměřená hodnota ze senzoru je menší nebo rovna než 5 cm.



Obrázek 44: Nastavení ultrazvukového senzoru

Vytvoříme sekvenci, která bude začínat spuštěním motoru pro pohyb vozítka vpřed. Dalším prvkem bude právě čekání na ultrazvukový senzor, než naměří hodnotu menší nebo rovno 5 cm. Pokud se vozítko přiblíží k nějaké překážce do uvedené vzdálenosti, program přejde do další fáze a tím bude zastavení motoru. Následně počkáme jednu sekundu a zahájíme opačný proces. Pomocí bloku pro ovládání motoru rozpohybujeme vozítko směrem vzad a čekáme na stisknutí dotykového senzoru. Až dojde ke stisknutí, zastavíme motor a počkáme znovu jednu sekundu. Celá tato sekvence bude uzavřena ve smyčce, kde ukončující podmínkou bude stisknutí prostředního ovládacího tlačítka na řídicí jednotce EV3. Řešení programu E je následující:



Obrázek 45: Senzory - řešení programu E

5.5 ÚLOHA 5: Zatáčení vozítka

5.5.1 Zadání úlohy

Proveďte přestavbu vozítka tak, aby bylo schopné zatáčet. Vytvořte a otestujte funkčnost těchto programů:

Program A: Vozítko pojede 3 sekundy rovně, poté se otočí a vrátí zpět na původní pozici.

Program B: Vozítko opíše kruh.

Program C: Vozítko opíše čtverec.

Program D: Vozítko opíše čtverec, otočí se a opíše čtverec znovu opačným směrem.

Program E: Vozítko opíše tvar osmičky.

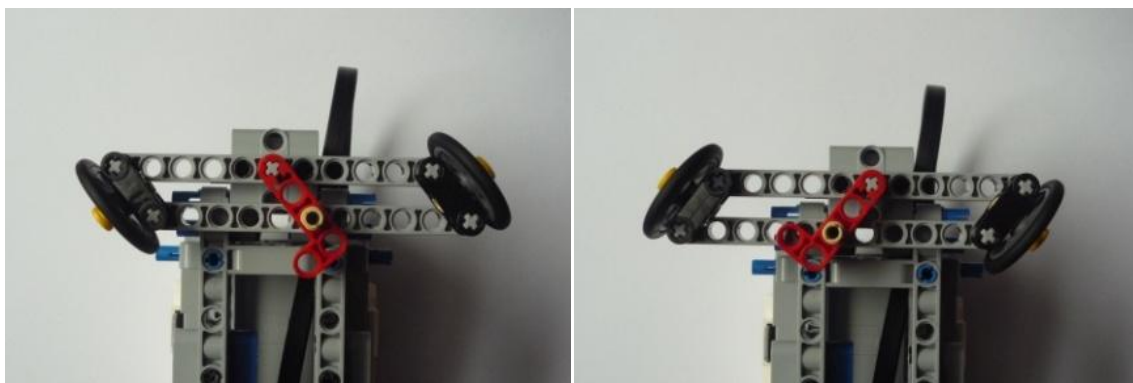
5.5.2 Komentované řešení

Existuje více způsobů, jak vyřešit problém řízení. V tomto řešení jsou popsány dva základní způsoby přestavby robotického vozítka tak, aby bylo schopné zatáčet.

1. Úprava přední nápravy na řízenou s využitím středního servomotoru
2. Přestavba vozítka s využitím dvou velkých servomotorů

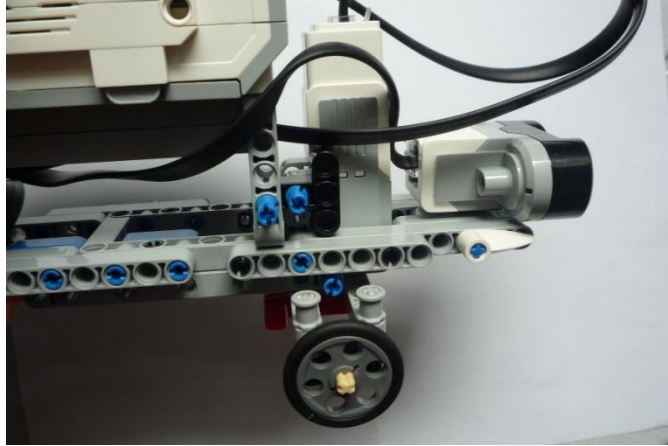
Možnost 1: Úprava přední nápravy

Na přední část vozítka připevníme střední servomotor. Umístíme ho směrem dolů a dle obrázků jej pomocí propojovací osy napojíme na zatáčecí nápravu. Využijeme malá kolečka, aby bylo možno zatáčet lehce. Střední motor nemá tak velký krouticí moment.



Obrázek 46: Ukázka otáčení přední nápravy

Při úpravě počítáme s montáží původního ultrazvukového senzoru. Motor i senzor správně propojíme kabelem s řídicí jednotkou. Pro motor využijeme jeden z volných portů A-D a pro senzor port z řady 1-4.

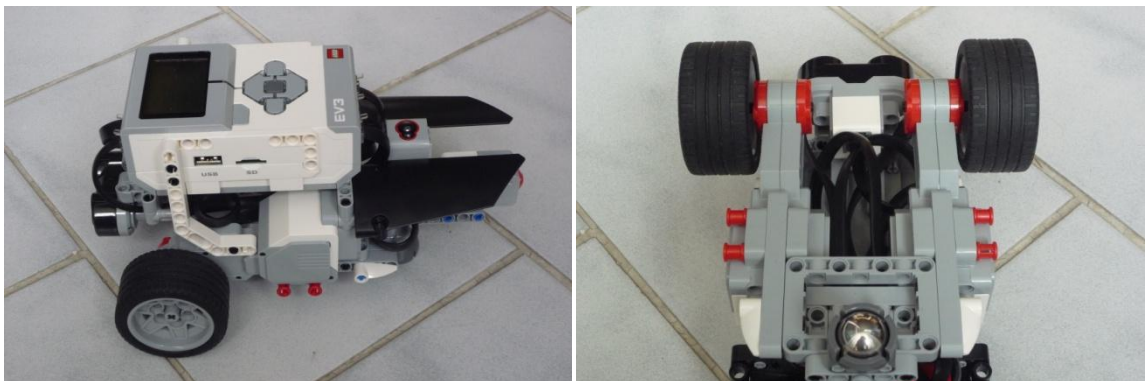


Obrázek 47: Konstrukce vozítka s řízenou nápravou včetně senzoru

Možnost 2: Přestavba vozítka s využitím dvou velkých servomotorů

Zatáčení vozítka můžeme vyřešit i pomocí dvou velkých servomotorů. Je potřeba provést větší přestavbu, protože jde o zásadní změnu konstrukce. Dá se říct, že nám tím může vzniknout úplně jiné robotické vozítko. Můžeme si stanovit, že poháněná náprava bude přední. Místo zadní nápravy je nejvhodnější použít kuličku, která nebude bránit např. v otáčení vozítka na malém prostoru.

Vozítko rozšíříme i o senzory. V rámci přestavby je možné se zaměřit i na estetickou úpravu modelu, např. využití černých krytů. Jedna z možných podob vozítka po přestavbě je uvedena na následujících obrázcích.



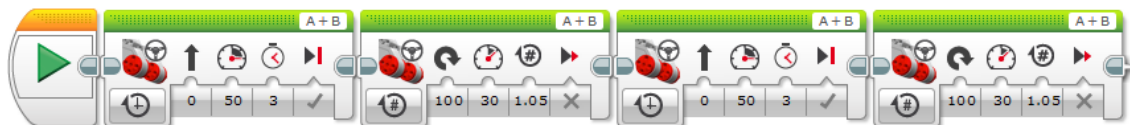
Obrázek 48: Konstrukce vozítka s řízenou nápravou včetně senzoru

Pro řešení programů v další části projektu budeme využívat takto přestavěné vozítko kvůli jeho lepším vlastnostem pro zatáčení.

Řešení programu A

Ke splnění prvního úkolu využijeme speciální blok pro ovládání dvou motorů současně. Slouží k zatáčení vozítka právě v tom případě, když jsou pro pohon využity velké servo-motory. V bloku se nastavuje směr jízdy, výkon a parametry závislé na typu spouštění. Typy spouštění jsou stejné jako u ovládání samostatného motoru.

Vozítko má jet tři sekundy rovně. To zajistí první blok, kde směr necháme nastavený na hodnotu 0 (šipka nad parametrem ukazuje přibližný směr pohybu). Výkon si zvolíme sami – např. 50. Dobu jízdy nastavíme na 3 sekundy a můžeme ponechat aktivní zabrzdění motoru. Po těchto třech sekundách je potřeba vozítko otočit, k tomu slouží stejný akční blok s odlišným nastavením. Spustí se jen na jednu otáčku, s menší rychlostí pohybu a směrem 100, což znamená otočení o 180°. Pro návrat zpět a otočení vozítka do původní pozice využijeme stejné bloky se stejným nastavením jako první dva. Po otestování zjistíme, že otáčení vozítka není úplně přesné (je závislé na povrchu). Otáčení vozítka můžeme upravit např. na 1,05 otáčky místo 1, abychom docílili lepšího výsledku. Řešení programu může být následující:



Obrázek 49: Zatáčení - řešení programu A

Řešení programu B

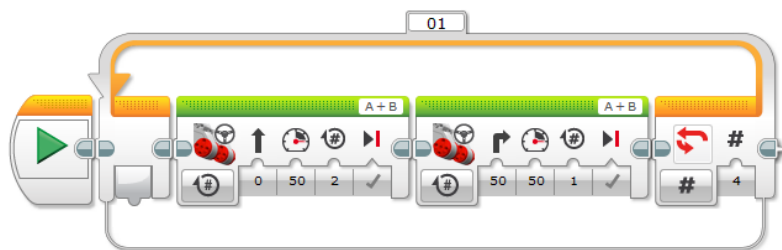
Cílem je, aby vozítko opsalo kruh. Klíčem ke správnému řešení je využití a správné nastavení jediného akčního bloku pro pohon motorů. Nejlepší metodou je program testovat a tím dojít k úspěšnému řešení. V tomto případě se osvědčilo nastavit směr na hodnotu 24, poloviční výkon a dobu jízdy 6 sekund. Vozítko se zastaví na původní pozici. Nastavení se může lišit v závislosti na konstrukci vozítka.



Obrázek 50: Zatáčení - řešení programu B

Řešení programu C

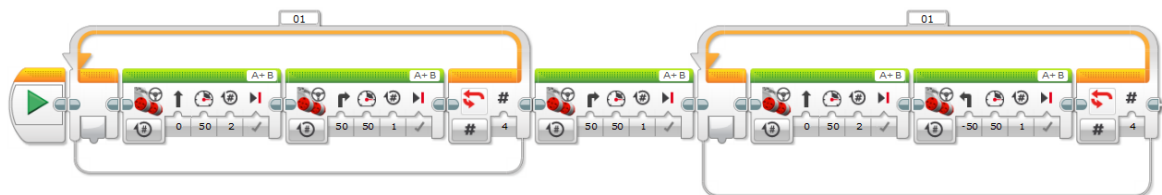
Je nutné se zamyslet, co vlastně pro vozítko znamená opsat čtverec. Zjistíme, že rozhodující je pohyb vpřed po určitou vzdálenost a poté otočení o 90° doprava nebo doleva. Abychom docílili čtverce, musí se tato činnost zopakovat čtyřikrát. Jedno z možných řešení je znázorněno na obrázku. Je využita smyčka pro opakování, která má počet opakování nastavený na konkrétní hodnotu 4. Vozítko jede vždy rovně po dobu dvou sekund a otočení je nastaveno směrem vpravo.



Obrázek 51: Zatáčení - řešení programu C

Řešení programu D

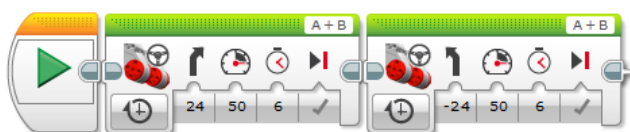
Tato úloha navazuje na předchozí, kde jsme docílili opsání čtverce. Tuto část můžeme beztestně využít. Následovat bude blok pro ovládání motoru, pomocí kterého zajistíme otočení o 90° vpravo. Tím se vozítko nasměruje do zpětného směru. Aby opsalo stejný čtverec, opět můžeme využít část s opakováním, ale musíme upravit směr zatáčení – změnit na otočení o 90° vlevo. Řešení programu je znázorněno na obrázku:



Obrázek 52: Zatáčení - řešení programu D

Řešení programu E

Pro opsání elipsy je potřeba objet vozítkem kruhový oblouk v jednom směru a poté v opačném směru. Klíčové je v tomto případě správné načasování, které musíme postupně odladit. Výsledkem může být následující program:



Obrázek 53: Zatáčení - řešení programu E

5.6 ÚLOHA 6: Proměnné a operace s čísly

5.6.1 Zadání

Využijte vozítko netradičně a pomocí něj se seznámte s funkcí proměnných a prací s nimi. Vytvořte následující programy a ověřte jejich funkčnost:

Program A: Uložte číselnou hodnotu do proměnné, poté jí přečtěte a zobrazte na displeji řídicí jednotky.

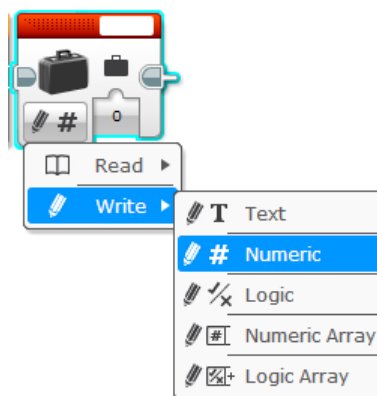
Program B: Uložte číselnou hodnotu do proměnné, přičtěte k ní jinou libovolnou hodnotu a výsledek zobrazte na displeji řídicí jednotky.

Program C: Vygenerujte náhodné číslo v rozmezí od 1 do 10 a ověřte, jestli je hodnota větší než 5. Výsledek oznamte prostřednictvím zvukového signálu na EV3 kostce.

Program D: Vytvořte program fungující jako počítadlo stisknutí dotykového senzoru na vozítku. Počet stisknutí se bude průběžně zobrazovat na displeji řídicí jednotky.

5.6.2 Komentované řešení

Proměnná slouží k uložení nějaké hodnoty (např. číslo nebo text) v nějaké části programu tak, aby šla později během programu použít. Lze do ní zapisovat a číst z ní. Blok pro práci s proměnnou se nachází mezi červenými operačními bloky. Po kliknutí na jeho tlačítko lze nastavit čtení nebo zápis (Read/Write).



Obrázek 54: Zatačení - řešení programu E

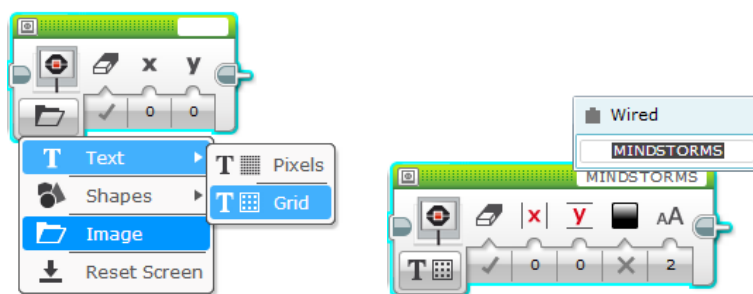
Proměnných lze v programu využít více. Kliknutím na bílý obdélníček v pravém horním rohu vybereme požadovanou proměnnou. Pokud tam žádná není, pomocí *Add Variable* přidáme novou a zadáme její název.



Obrázek 55: Zatáčení - řešení programu E

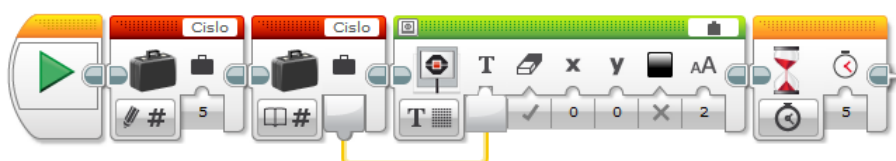
Řešení programu A

Prvním blokem bude blok pro proměnnou. Pro zapsání číselné hodnoty jej nastavíme na *Write - Numeric* a do vedlejšího chlívěčku zadáme požadovanou hodnotu 5. Název proměnné můžeme zvolit např. *Cislo*. Následovat bude stejný blok, ale s nastavením pro čtení. Program přečte hodnotu proměnné a přetažením jí můžeme nastavit jako parametr u jiného bloku. Je možné takto nastavit např. rychlost nebo dobu otáčení motoru. V tomto případě výsledek zobrazíme na displeji řídicí jednotky. Pro zobrazení využijeme k tomu určený blok z řady zelených (akčních). Pomocí tlačítka u něj nastavíme režim *Text - Grid*. Dále pomocí obdélníčku v pravé horní části zvolíme *Wired* pro napojení na proměnnou.



Obrázek 56: Nastavení bloku pro výstup na displej

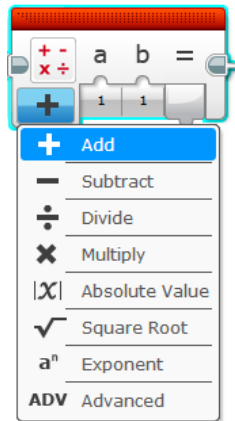
Přečtenou hodnotu z proměnné přetáhneme do chlívěčku parametru textu na displeji (pod písmenkem T). Další parametry necháme stejné. Je přednastavené vymazání displeje a šla by změnit např. pozice textu. Jako poslední prvek umístíme funkční blok pro čekání. Nastavíme ho např. na hodnotu 5 sekund. Po uplynutí této doby se program ukončí a hodnota proměnné z displeje zmizí. Bez tohoto bloku nejsme schopni ověřit funkčnost programu.



Obrázek 57: Proměnné - řešení programu A

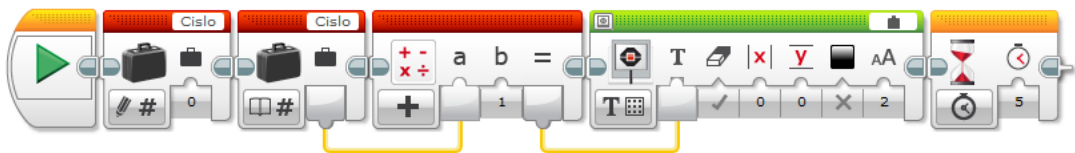
Řešení programu B

Práce s proměnnou je v tomto programu stejná jako v předchozí úloze. Přečtenou hodnotu nebudeme nastavovat na displej, ale do dalšího prvku pro operace s čísly. V tomto prvku necháme přednastavenou první možnost – sčítání. Do prvního chlívěčku přetáhneme přečtenou hodnotu z proměnné. Do druhého nastavíme jedničku, kterou chceme přičíst. Lze to udělat i obráceně. Výsledek operace je možné přečíst z chlívěčku pod znaménkem =.



Obrázek 58: Možnosti operace s čísly

Výsledek operace zobrazíme na displeji jako v předchozí úloze. Rovněž nastavíme čekací blok na určitou dobu, po kterou bude výsledek viditelný na displeji EV3 kostky.

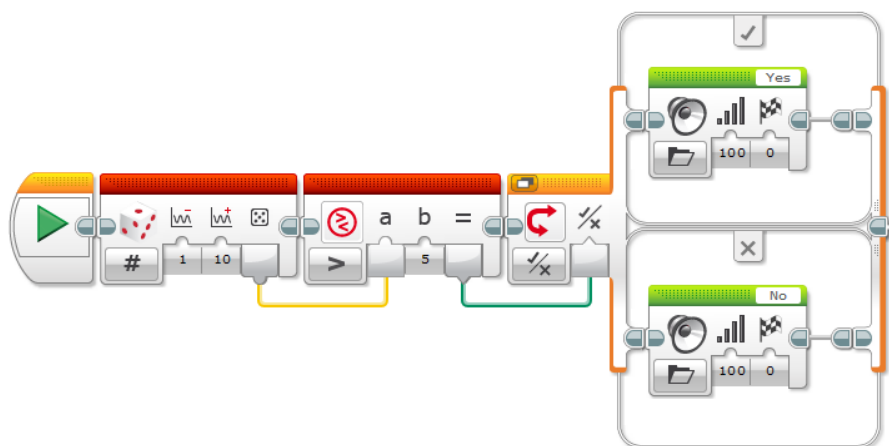


Obrázek 59: Proměnné - řešení programu B

Řešení programu C

Generování náhodného čísla zajistí operační blok k tomu určený. Poznáme ho podle symbolu hrací kostky. Jeho ovládání je intuitivní – nastavuje se rozmezí, ve kterém bude generovat náhodné číslo. Dolní hranici nastavíme na 1 a horní na 10. Vygenerovaná hodnota je v posledním chlívěčku. K porovnání využijeme porovnávací blok. Slouží k porovnání dvou hodnot za vybrané podmínky. Podmínku nastavíme na > (je větší než). První porovnávanou hodnotou bude vygenerované číslo, toho docílíme propojením chlívěčků přetažením. Bude se porovnávat s hodnotou 5, kterou zadáme ručně. Následovat musí rozhodovací prvek, ten musíme přepnout na logickou hodnotu (Logic). Tu získáme opět správným propojením – požadovaná logická hodnota je výsledkem porovnávacího bloku.

Posledním krokem je nastavení dvou větví v rozhodovacím prvku. Využijeme reproduktor ze zelených akčních bloků. V pravém horním rohu tohoto prvku můžeme zvolit soubor, který se má přehrát. Pro tento příklad vybereme z řady přednastavených zvuků – *Yes* a *No*. Po spuštění programu nám EV3 kostka anglicky odpoví, jestli je vygenerovaná hodnota větší než 5. Řešení programu je na následujícím obrázku:

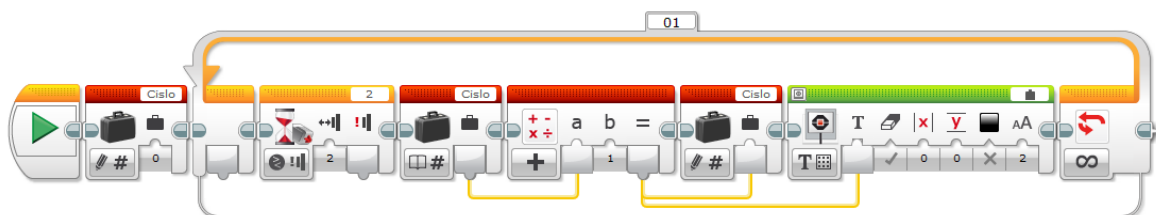


Obrázek 60: Proměnné - řešení programu C

Řešení programu D

Při řešení posledního úkolu této části využijeme prvky, které už známe. Jen je správně zvolíme do nekonečné smyčky. Jako první musíme založit proměnnou a nastavit do ní hodnotu 0. Poté může následovat smyčka, ve které první blok slouží k čekání na požadovaný signál z dotykového senzoru. Pro správnou funkčnost je důležité nastavit událost na stisknutí i uvolnění tlačítka. Když tato událost nastane, dojde k přečtení hodnoty z proměnné, přičtení jedničky a uložení proměnné. Kromě uložení je hodnota použita i pro výstup na displej řídicí jednotky. Následuje opětovné provedení smyčky, kde se opět čeká na stisknutí dotykového senzoru.

Řešení programu by bylo možné vylepšit tak, aby se místo nekonečné smyčky jednalo o smyčku trvající přesný počet sekund. Dalo by se například soutěžit, komu se podaří v omezeném čase provést největší počet stisknutí.



Obrázek 61: Proměnné - řešení programu D

5.7 ÚLOHA 6: Pokročilé programování

5.7.1 Zadání

Program A: Nastavte pohyb vozítka vpřed po neomezenou dobu. Pomocí dalšího procesu (druhé větve ze startovacího bloku) zajistěte, aby se vozítko zastavilo v případě nedostatečného osvětlení (např. když jej zakryjeme) a oznámilo to i zvykovým signálem. Až se hodnota osvětlení dostane na zvolenou dostatečnou hodnotu, bude vozítko pokračovat.

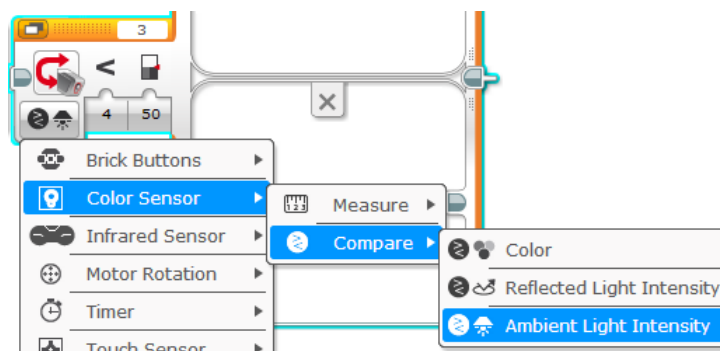
Program B: Vozítko pojede rovně po dobu 30 sekund a zastaví se. Pokud ale během jízdy detekuje překážku nacházející se méně než 40 cm, zastaví se dříve a oznámí to zvukovým signálem. Po zastavení se na displeji zobrazí doba, po kterou bylo vozítko v pohybu.

Program C: Vozítko se rozjede velmi pomalu a bude postupně zrychlovat. Na displeji EV3 kostky se budou průběžně ukazovat stupně otočení motoru. Po dosažení plné rychlosti se vozítko zastaví a zazní zvukový signál oznamující dokončení programu.

5.7.2 Komentované řešení

Řešení programu A

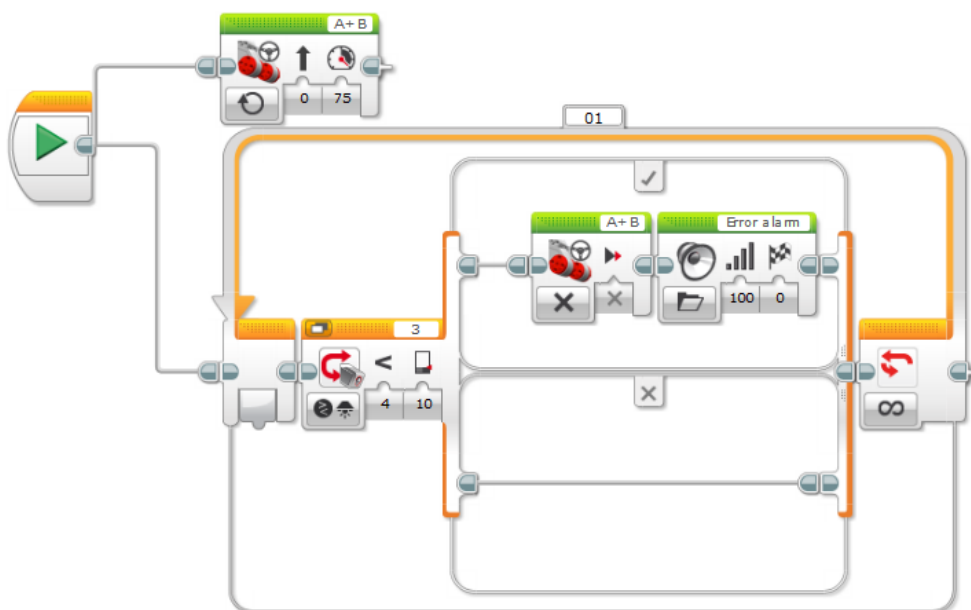
První větev programu zajišťuje spuštění motorů pro pohyb vozítka vpřed (na základě zadání). Ve druhé větvi se bude v nekonečně stále se opakující smyčce testovat, zda neklesla hodnota osvětlení. Toho docílíme připojením barevného senzoru a změnou nastavení jeho režimu – *Color Sensor – Compare – Ambient Light Intensity*.



Obrázek 62: Nastavení barevného senzoru pro měření intenzity osvětlení

Dovnitř smyčky umístíme rozhodovací prvek, u kterého provedeme výše uvedené nastavení. Podmínkou tedy bude právě hodnota osvětlení, kterou můžeme nastavit např. na 10. Pokud senzor vyhodnotí příliš slabé osvětlení, dojde k zastavení motoru a spuštění zvuku, např. předvolený soubor *Error alarm*. Smyčka neustále testuje stav osvětlení, až bude při-

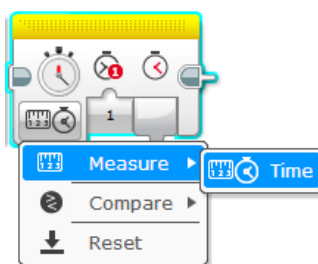
jatelný, přestane se motor blokovat a první proces dále zajistí jeho chod. Existuje více možností řešení této úlohy, jedno z řešení s využitím dvou větví je na obrázku.



Obrázek 63: Pokročilé programování - řešení programu A

Řešení programu B

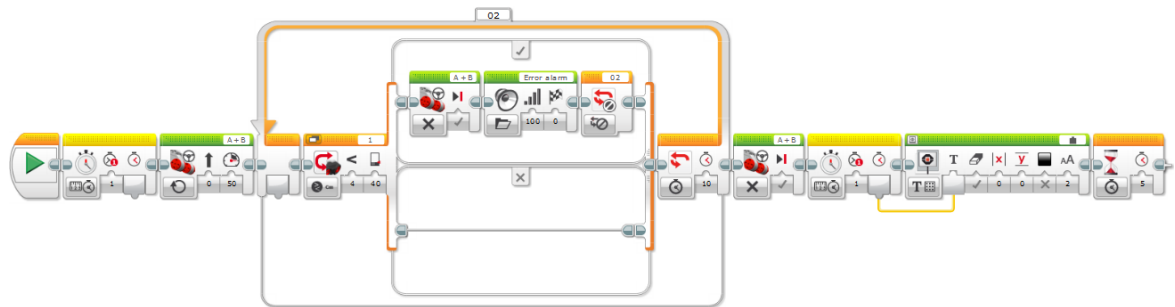
K řešení programu B lze opět přistoupit více způsoby. Je možné využít více procesů, v tomto popsaném řešení se zaměříme na řešení bez využití dalších procesů. Jelikož je potřeba měřit čas, využijeme nový prvek – časovač. Jeho nastavení ponecháme na *Measure – Time* pro měření času. ID časovače ponecháme na hodnotě 1. Prvním vložením tohoto prvku do programu se časovač spustí. Během programu jej využijeme pro zjištění hodnoty.



Obrázek 64: Nastavení časovače

Po nastavení časovače spustíme motor na neomezenou dobu pro pohyb vpřed. Hodnotu 30 sekund nastavíme jako dobu trvání smyčky, která bude následovat. Uvnitř smyčky bude rozhodovací prvek, který bude měřit, zda hodnota z ultrazvukového senzoru nerozpozná překážku v zadané vzdálenosti. Pokud ano, zastaví se motor, zazní zvukový signál a další prvek zajistí opuštění smyčky. Za smyčkou dojde k zastavení motoru, což se využije v případě uplynutí doby 30 sekund. Následuje přečtení hodnoty z časovače a zobrazení uplynu-

té doby na displeji EV3 kostky. Na závěr je nutné vložit čekací blok, aby byla hodnota na displeji nějakou dobu viditelná. Řešení je na následujícím obrázku:



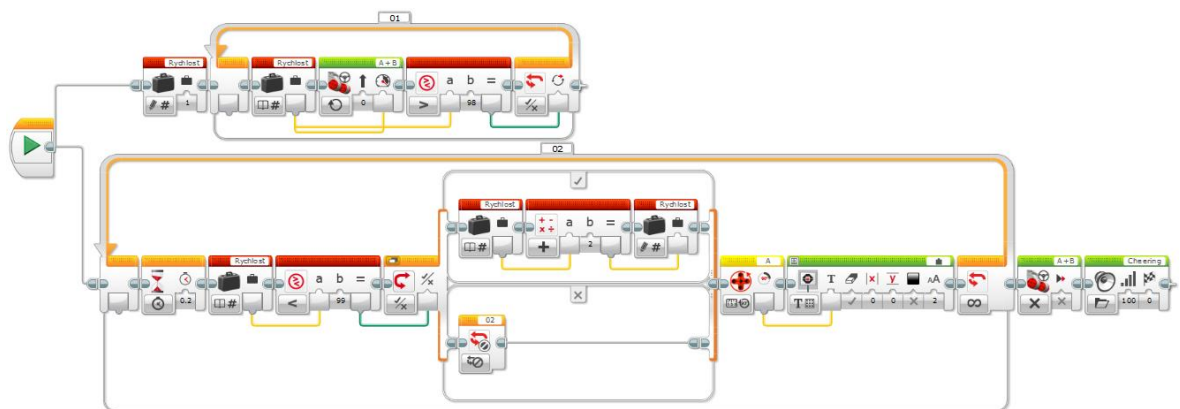
Obrázek 65: Pokročilé programování - řešení programu B

Řešení programu C

Cílem je, aby se vozítko rozjíždělo pomalu. Parametr rychlosti se musí průběžně měnit a to musíme zajistit pomocí proměnné. Rozdělíme si program na dva procesy (větve). V prvním se provede založení proměnné *Rychlost* s nulovou počáteční hodnotou. Ve smyčce se bude hodnota proměnné číst a přenášet do parametru rychlosti motoru. Ukončující podmínkou smyčky bude případ, kdy hodnota proměnné bude větší než 98.

Druhá větev programu bude zajišťovat nárůst proměnné. Rovněž bude tento proces probíhat v nekonečné smyčce, na jejímž začátku bude časová prodleva se zvolenou hodnotou 0,2 sekund. Každou sekundu se tedy zvýší hodnota proměnné pětkrát. Přečte se aktuální hodnota proměnné - v případě hodnoty 99 a vyšší se cyklus opustí. Pokud je menší než 99, přičte se jednička a pokračuje se dál přečtením hodnoty rotace motoru a zápisu na displej. Po opuštění smyčky dojde k zastavení motoru a oznamujícímu zvukovému signálu.

Náhled jednoho z možných řešení programu je na následujícím obrázku. V plné kvalitě a funkční podobě je program dostupný na příloženém CD.



Obrázek 66: Pokročilé programování - řešení programu C

5.8 ÚLOHA 7: Inteligentní chování vozítka

5.8.1 Zadání

Vymyslete, jak by se vozítko mělo chovat, aby bylo samostatné a rozhodovalo se inteligentně (např. způsob jeho reakce na překážku).

Vytvořte program pro inteligentní chování vozítka. Využijte pokročilé možnosti programování – smyčky, rozhodovací prvky a proměnné.

5.8.2 Komentované řešení

Vozítko by se mělo samo rozhodovat na základě informací, které má dostupné, tedy ze senzorů. Musí vyhodnocovat své bezprostřední okolí. Při určitých situacích by mělo vozítko oznamovat události zvukovým signálem. Při detekci překážky by mělo couvnout a objet ji. Při couvání by si mělo dávat pozor, aby do něčeho nenarazilo. Pokud ano, motor se musí ihned zastavit a nesmí do překážky tlačit. Vše by mělo být proloženo vhodnými časovými prodlevami, aby bylo možné rozeznat správnou funkci vozítka. Program bude ukončen stisknutím tlačítka na EV3 kostce.

Je možné vozítko obohatit o funkcionalitu, že při nedostatečném osvětlení v místnosti se sníží jeho rychlost a tento stav nouze bude signalizován i zvukově.

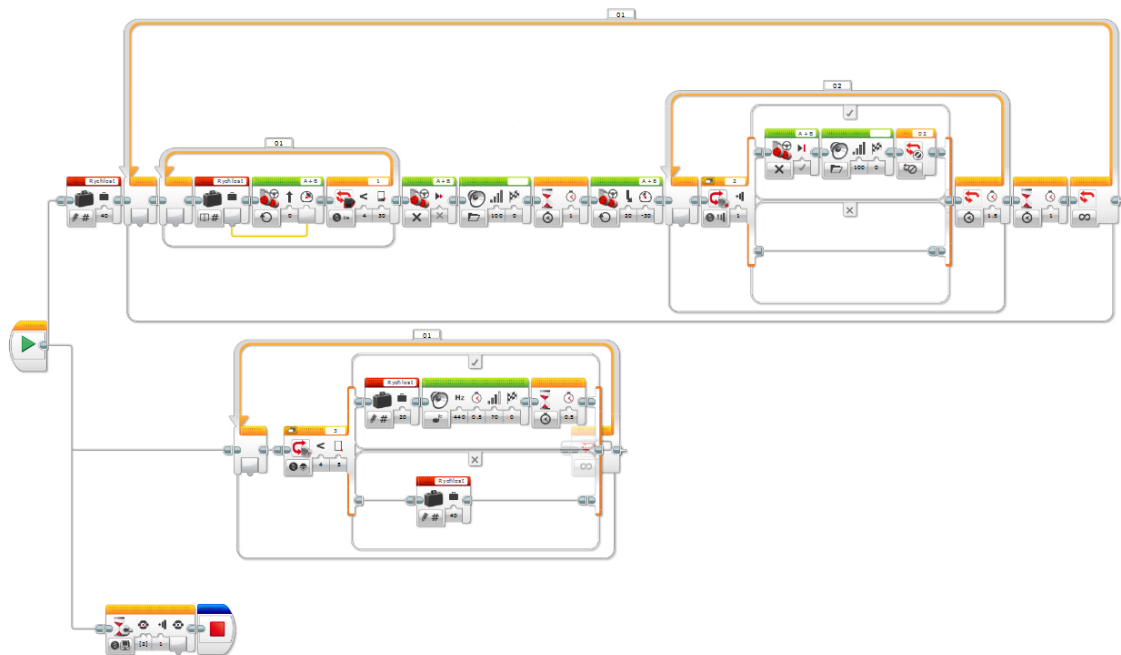
Program vozítka je řešen třemi větvemi. První větev zajišťuje pohyb vozítka. Nejprve dojde k založení proměnné Rychlost a následuje nekonečná smyčka. Na jejím začátku se v další smyčce spouští jízda vpřed. Rychlost se neustále načítá z aktuální hodnoty proměnné. Pokud vozítko před sebou detekuje překážku ve vzdálenosti 30 cm, zastaví se, oznámí to zvukovým signálem a počká jednu sekundu. Poté se spustí couvání vozítka a následuje další smyčka obsluhující couvání. Probíhá pouze 1,5 sekundy, ale i během této doby se musí neustále testovat dotykový senzor, zda nedošlo k nárazu. Pokud ano, couvání se ihned ukončí a zvukovým signálem se oznámí náraz. V tomto případě dojde k opuštění smyčky dříve než po nastavené době. Po této akci nastává znovu jednosekundová prodleva. Poté se opakuje pohyb vpřed.

Druhá větev v nekonečné smyčce hlídá intenzitu okolního osvětlení. Pokud je nedostatečná, sníží se hodnota proměnné Rychlost a tím pádem se ihned ovlivní rychlost pohybu, který zajišťuje první větev a využívá k tomu aktuální hodnotu proměnné. Dochází rovněž k aktivaci zvukového alarmu.

Třetí větev neustále čeká na stisknutí prostředního tlačítka na EV3 kostce. Pokud k němu dojde, program je ihned ukončen.

Po otestování funkčnosti programu můžeme dojít k novým nápadům a vylepšením, které lze aplikovat. Jedno z nich je např. reagovat na překážku dříve, než je 30 cm před vozítkem. Důvodem je to, že senzor snímá hodnotu přímo před ním. Pokud vozítko najíždí k překážce zešikma, není schopné dostatečně rychle zareagovat a může dojít k jeho najetí na překážku a převrácení.

Náhled řešení programu je na následujícím obrázku. V plné kvalitě a funkční podobě je program dostupný na příloženém CD.



Obrázek 67: Inteligentní chování vozítka - řešení programu

6 SYSTÉM HODNOCENÍ ÚLOH

Ke každému projektu patří jeho zhodnocení. V případě zájmového kroužku nejde o stěžejní věc a není na hodnocení kladen takový důraz jako při běžné výuce. Přesto se podíváme, jak je možné práci žáků hodnotit.

6.1 Průběžné hodnocení

Vedoucí kroužku by měl průběžně hodnotit práci svých žáků verbální i neverbální formou. Jde o důležitou věc pro správnou motivaci žáků k dosažení lepších výsledků. Jednoduché přikývnutí, gesto nebo zavrtění hlavou může žákovi pomoci. Slovní průběžné hodnocení může mít podobu krátkých vět jako např. *„Ano, to je správně. Dnes jsi mě potěšil.“* nebo *„Není to ono, zkus to ještě nějak jinak.“*

6.2 Závěrečné hodnocení

Závěrečné hodnocení může být v kroužku robotiky a programování realizováno různými způsoby. Jako nejvhodnější varianta se jeví společná diskuse se žáky, ve které zhodnotí, co všechno se naučili, co se jim líbilo a co se jim naopak nelíbilo, co bylo pro ně jednoduché nebo naopak příliš složité. Poznatky z diskuse lze využít jako zpětnou vazbu pro vylepšení projektu. Závěrečné slovní hodnocení žáků ze strany vedoucího kroužku je závislé na přípravě žáků, průběžné činnosti, rychlosti práce a správnosti řešení.

6.3 Sebehodnocení žáků

Na výsledek projektu se mohou kriticky podívat i sami žáci. Mohou zhodnotit, co se jim při práci povedlo a co se jim naopak nedařilo, s čím se trápili. Ze sebehodnocení žáků mohou také plynout návrhy na zlepšení pro práci na podobných projektech v budoucnu.

7 ROZVOJ DOVEDNOSTÍ ŽÁKA

Úlohy jsou navrženy tak, aby postupnou formou rozvíjely níže popsané dovednosti žáka.

7.1 Mechanická zručnost

Mechanickou zručnost rozvíjí část zaměřená na stavbu a přestavbu vozítka. Stavba je založena na volbě konstrukce, žáci využívají jemnou motoriku a postupně si tím vylepšují mechanické dovednosti.

7.2 Schopnost řešení problémů

Žáci jsou nuceni při práci překonávat překážky, tím se u nich zlepšuje schopnost řešení problémů. V nesnázích jim může dát vedoucí kroužku určité indicie a rady, ale řešení problému zůstává stále na žácích.

7.3 Představivost

V jedné z úloh musí žáci přestavět vozítko tak, aby bylo schopné zatáčet. Žáci při práci uplatní a zlepšují svoji představivost. Ta je neméně využitelná i při tvorbě a úpravě programů.

7.4 Logické uvažování

Při programování jsou žáci vedeni k zamyšlení nad zadáním úloh a tím se zvyšuje jejich schopnost logicky uvažovat a kriticky přemýšlet.

7.5 Práce v týmu

Práce na projektu probíhá ve skupinách a tím se žáci učí základům týmové spolupráce, která může být vyžadována v jejich budoucím pracovním procesu. Musí být schopni se domluvit, jak si práci rozdělí a kdo má za co zodpovědnost.

ZÁVĚR

Cílem této diplomové práce byla příprava zadání a komentovaného řešení úloh pro zájmový kroužek programování a robotiky. Výstupem je návrh úloh a realizace na sebe navazujících řešení tvořících projekt, který postupnou formou rozvíjejí technické i programátorské schopnosti dětí na základní škole. Úlohy jsou účelně navrženy tak, aby rozvíjely dovednosti žáka.

Při tvorbě práce byly využity možnosti stavebnici LEGO Mindstorms EV3. Byly vytvořeny pracovní listy pro učitele s podrobným řešením úloh krok po kroku. Jsou popsány překážky a úskalí, která mohou v průběhu práce překvapit nejen žáky, ale i učitele. Komentovaná řešení jsou vhodně doplněna o obrázky, díky kterým je pochopení pracovního postupu mnohem jednodušší a efektivnější.

Podnikavost, aktivita a spoluzodpovědnost žáků je základem pro využitou projektovou metodou. Znakem této výukové metody je, že žáci projekt realizují od jeho plánování až po vytvoření konkrétního produktu.

Vytvořený materiál si klade za cíl podněcovat kreativní myšlení k řešení problémových úloh, vytváří návyky pro práci v týmu, rozvíjí komunikační dovednosti a vše potřebné pro úspěšnost ve škole i mimo ni. Žáci při práci zastávají role skutečných konstruktérů a vědců, rozvíjejí dovednost řešit problémy a komunikovat.

Jako přílohu přikládám CD s textem práce a s vytvořenými programy v aplikaci LEGO Mindstorms EV3. Přílohou je i obrázek závěrečného programu pro inteligentní chování vozítka.

SEZNAM POUŽITÉ LITERATURY

- [1] W. H. KILPATRICK, *The Project Method: The Use of the Purposeful Act in the Educative Process*, University of California, Eleventh Impression March, 1929.
- [2] KRATOCHVÍLOVÁ, Jana, *Teorie a Praxe projektové výuky*. Brno: MU 2006, ISBN: 89-210-4142-0.
- [3] NOVÁK, Daniel, *Elektrotechnické stavebnice v technické výchově*, Praha, 1997, ISBN: 80-86039-37-4.
- [4] SKALKOVÁ, Jarmila, *Obecná didaktika*. Praha: ISV, 1999, ISBN: 80-85866-33-1.
- [5] PAVELKOVÁ, Isabella, *Motivace žáků k učení: perspektivní orientace žáků a časový faktor v žákovské motivaci*. Praha: Universita Karlova, 2002, ISBN: 80-729-0092-7.
- [6] VALENTA, Josef. et al., *Pohledy: Projektová metoda ve škole a za školou*. Praha: Ipos Artama, 1993, 61 s.
- [7] Projektové vyučování – 1. díl – Charakteristika. *Čtenářská gramotnost a projektové vyučování* [online]. 2010 [cit. 2016-04-28]. Dostupné z: <http://www.ctenarska-gramotnost.cz/projektove-vyucovani/pv-tipy/projektove-vyucovani-1>
- [8] EV3 and NXT: Difference and Compatibility. *Robot Square* [online]. 2013 [cit. 2016-04-27]. Dostupné z: <http://robotsquare.com/2013/07/16/ev3-nxt-compatibility/>
- [9] MERKUR Robotika a mechatronika. *MERKURTOYS* [online]. [cit. 2016-04-24]. Dostupné z: <http://www.merkurtoys.cz/vyrobky/robotika-a-mechatronika>
- [10] Arduino Zero & Genuino Zero. *Arduino* [online]. [cit. 2016-04-24]. Dostupné z: <https://www.arduino.cc/en/Main/ArduinoBoardZero>
- [11] LEGO MINDSTORMS Education EV3 Core Set. *LEGO Education* [online]. 2013 [cit. 2016-04-28]. Dostupné z: <https://education.lego.com/en-us/products/lego-mindstorms-education-ev3-core-set-/5003400>
- [12] Lego Mindstorms EV3. *Wikipedia* [online]. 2016 [cit. 2016-04-28]. Dostupné z: https://en.wikipedia.org/wiki/Lego_Mindstorms_EV3

- [13] LEGO MINDSTORMS Education EV3 Support. *LEGO Education* [online]. [cit. 2016-04-28]. Dostupné z: <https://education.lego.com/en-us/support/mindstorms-ev3>
- [14] NXT vs EV3. *LEGO Education* [online]. [cit. 2016-04-28]. Dostupné z: <https://education.lego.com/en-gb/lesi/middle-school/mindstorms-education-ev3/nxt-vs-ev3>
- [15] *LEGO Shop* [online]. [cit. 2016-05-14]. Dostupné z: <http://shop.lego.com/>
- [16] Nauč se programovat. *LEGO* [online]. [cit. 2016-04-28]. Dostupné z: <http://www.lego.com/cs-cz/mindstorms/learn-to-program>
- [17] *EDUXE s.r.o. - distributor učebních pomůcek LEGO Education* [online]. [cit. 2016-04-28]. Dostupné z: <http://www.eduxe.cz/>
- [18] *LEJOS - Java for LEGO Mindstorms* [online]. [cit. 2016-04-28]. Dostupné z: <http://www.lejos.org/>
- [19] *Bricx Command Center 3.3* [online]. [cit. 2016-04-28]. Dostupné z: <http://bricxcc.sourceforge.net/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

EV3 Označení stavebnice LEGO Mindstorms (EV – „evolution“, 3 – třetí generace)

RVP Rámcový vzdělávací program

ŠVP Školní vzdělávací program

RAM Operační paměť (Random Access Memory)

SEZNAM OBRÁZKŮ

<i>Obrázek 1: Stavebnice robota Mini Sumo GT12 – Merkur [9]</i>	14
<i>Obrázek 2: Základní deska stavebnice Arduino [10]</i>	15
<i>Obrázek 3: Porovnání kostek ze stavebnic LEGO Mindstorms RCX, NXT a EV3 [8]</i>	17
<i>Obrázek 4: Stavebnice LEGO Mindstorms Education EV3 [6]</i>	18
<i>Obrázek 5: Řídicí jednotka – kostka [6]</i>	18
<i>Obrázek 6: Velký motor [6]</i>	20
<i>Obrázek 7: Střední motor [15]</i>	20
<i>Obrázek 8: Dotykový senzor [15]</i>	21
<i>Obrázek 9: Ultrazvukový senzor [15]</i>	21
<i>Obrázek 10: Barevný senzor [15]</i>	22
<i>Obrázek 11: Gyroskopický senzor [15]</i>	22
<i>Obrázek 12: Ukázka jednoduchého programu v LEGO Mindstorms EV3</i>	24
<i>Obrázek 13: Akční bloky</i>	29
<i>Obrázek 14: Funkční bloky</i>	29
<i>Obrázek 15: Sensorové bloky</i>	29
<i>Obrázek 16: Operační bloky</i>	30
<i>Obrázek 17: Pokročilé bloky</i>	30
<i>Obrázek 18: Možné podoby výstupu projektu</i>	31
<i>Obrázek 19: Konstrukce podvozku vozítka a montáž řídicí jednotky</i>	32
<i>Obrázek 20: Volba ručního ovládní motorů</i>	33
<i>Obrázek 21: Ruční ovládní motorů pomocí řídicí jednotky</i>	33
<i>Obrázek 22: Instalace aplikace LEGO Mindstorms Commander</i>	34
<i>Obrázek 23: Instalace aplikace LEGO Mindstorms Commander</i>	34
<i>Obrázek 24: Vložení prvku do vlastního programu</i>	35
<i>Obrázek 25: Ovládní vozítka pomocí mobilního telefonu</i>	35
<i>Obrázek 26: Prostředí programu LEGO MINDSTORMS EV3</i>	36
<i>Obrázek 27: Informace o připojení řídicí jednotky</i>	37
<i>Obrázek 28: Ovládací panel velkého motoru</i>	37
<i>Obrázek 29: První programování - řešení programu A</i>	38
<i>Obrázek 30: První programování - řešení programu B</i>	38
<i>Obrázek 31: Smyčka pro opakování programu</i>	39
<i>Obrázek 32: První programování - řešení programu C</i>	39

<i>Obrázek 33: Vozítko doplněné o senzory</i>	40
<i>Obrázek 34: Připojené akční bloky a senzory</i>	40
<i>Obrázek 35: Nastavení pro senzor</i>	41
<i>Obrázek 36: Události dotykového senzoru</i>	41
<i>Obrázek 37: Senzory - řešení programu A</i>	42
<i>Obrázek 38: Senzory - řešení programu B</i>	42
<i>Obrázek 39: Senzory - řešení programu C</i>	42
<i>Obrázek 40: Senzory - alternativní řešení programu C</i>	43
<i>Obrázek 41: Nastavení ukončení programu při stisknutí tlačítka na jednotce</i>	44
<i>Obrázek 42: Senzory - řešení programu D</i>	44
<i>Obrázek 43: Import ultrazvukového senzoru do programu</i>	45
<i>Obrázek 44: Nastavení ultrazvukového senzoru</i>	46
<i>Obrázek 45: Senzory - řešení programu E</i>	46
<i>Obrázek 46: Ukázka otáčení přední nápravy</i>	47
<i>Obrázek 47: Konstrukce vozítka s řízenou nápravou včetně senzoru</i>	48
<i>Obrázek 48: Konstrukce vozítka s řízenou nápravou včetně senzoru</i>	48
<i>Obrázek 49: Zatačení - řešení programu A</i>	49
<i>Obrázek 50: Zatačení - řešení programu B</i>	49
<i>Obrázek 51: Zatačení - řešení programu C</i>	50
<i>Obrázek 52: Zatačení - řešení programu D</i>	50
<i>Obrázek 53: Zatačení - řešení programu E</i>	50
<i>Obrázek 54: Zatačení - řešení programu E</i>	51
<i>Obrázek 55: Zatačení - řešení programu E</i>	52
<i>Obrázek 56: Nastavení bloku pro výstup na displej</i>	52
<i>Obrázek 57: Proměnné - řešení programu A</i>	52
<i>Obrázek 58: Možnosti operace s čísly</i>	53
<i>Obrázek 59: Proměnné - řešení programu B</i>	53
<i>Obrázek 60: Proměnné - řešení programu C</i>	54
<i>Obrázek 61: Proměnné - řešení programu D</i>	54
<i>Obrázek 62: Nastavení barevného senzoru pro měření intenzity osvětlení</i>	55
<i>Obrázek 63: Pokročilé programování - řešení programu A</i>	56
<i>Obrázek 64: Nastavení časovače</i>	56
<i>Obrázek 65: Pokročilé programování - řešení programu B</i>	57

Obrázek 66: Pokročilé programování - řešení programu C 57
Obrázek 67: Inteligentní chování vozítka - řešení programu 59

SEZNAM TABULEK

<i>Tabulka 1: Porovnání stavebnic z hlediska využití pro výuku [9,10,11]</i>	15
<i>Tabulka 2: Srovnání verzí RCX, NXT a EV3 [12]</i>	17
<i>Tabulka 3: Technická specifikace EV3 kostky [11,12]</i>	19
<i>Tabulka 4: Technická specifikace velkého motoru [15]</i>	19
<i>Tabulka 5: Technická specifikace středního motoru [15]</i>	20
<i>Tabulka 6: Technická specifikace ultrazvukového senzoru [15]</i>	21
<i>Tabulka 7: Technická specifikace barevného senzoru [15]</i>	22
<i>Tabulka 8: Technická specifikace gyroskopického senzoru [15]</i>	22
<i>Tabulka 9: Technické požadavky aplikace LEGO MINDSTORMS EV3 [13]</i>	23

SEZNAM PŘÍLOH

P I Řešení závěrečného programu

P II CD s textem práce a vytvořenými programy

PŘÍLOHA P I: ŘEŠENÍ ZÁVĚREČNÉHO PROGRAMU

