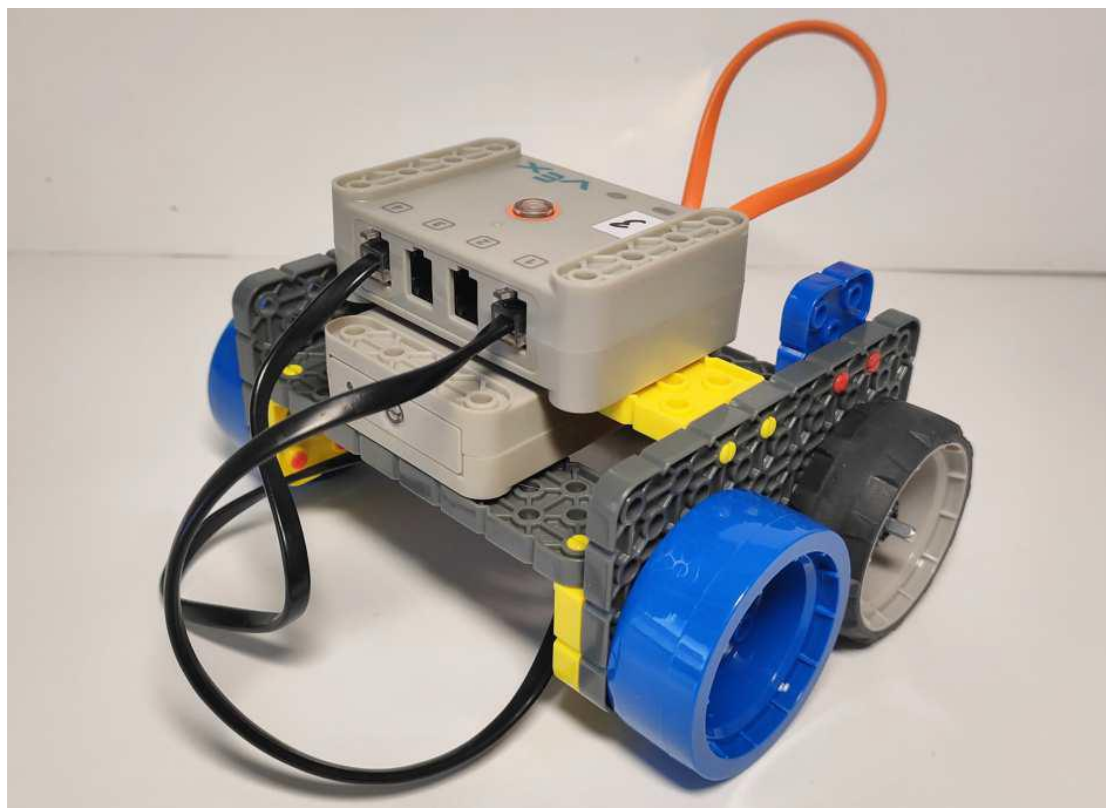


Úlohy pro robotickou stavebnici VEX GO



5 Praktická část

Praktická část práce se věnuje ověřování úloh z učebnice „Robotika s LEGO Mindstorms pro 2. stupeň základní školy“ [55] a tvorbě nových úloh pro robotickou stavebnici VEX GO. Nově navržené úlohy jsou zpracovány v rámci diplomové práce z důvodu, že nebylo možné všechny původní úlohy předělat pro práci se stavebnicí VEX GO.

Původní učebnice se skládá z 11 kapitol. Kapitoly jsou dále rozděleny na jednotlivé podúlohy, přičemž vybrané úlohy jsou dále rozčleněny na dílčí úkoly, viz obrázek č. 1 níže. V textu praktické části práce zůstala zachována původní struktura a dochází tedy k rozlišení mezi: úkoly, hádankami a nápovědami.



Obrázek 1: Ukázka rozčlenění úloh a úkolů v původní učebnici

5.1 Kapitola 1: Stavíme pojízdného robota

První kapitola je krátká, obecná a zabývá se stavbou základního pojízdného robota. Ačkoli se kapitola může jevit triviální, je důležitá především z důvodu další práce se základním robotem i v následujících kapitolách. Mimo jiné i kvůli seznámení se stavebnicí, rozvíjení konstrukčních dovedností a zručnosti.

Na kapitolu je dle původní učebnice stanovena jedna vyučovací hodina (45 minut). Vyjma stavění robota kapitola nastíní prvotní připojení, a pro rychlejší žáky je připraveno prozkoumání programovacího prostředí včetně prvních pohybů robota, například pohyb dopředu, dozadu či otočení do strany.

5.1.1 Stavba robota – původní učebnice

Stavba robota podle metodik uvedených u původní učebnice zabere nezkušenému uživateli přibližně 45 minut, zatímco zkušený stavitel ji zvládne dokončit zhruba

za 15 minut. Tento časový údaj odpovídá době sestavení robota, kdy byl v rámci praktické části robot postaven během 20 minut.

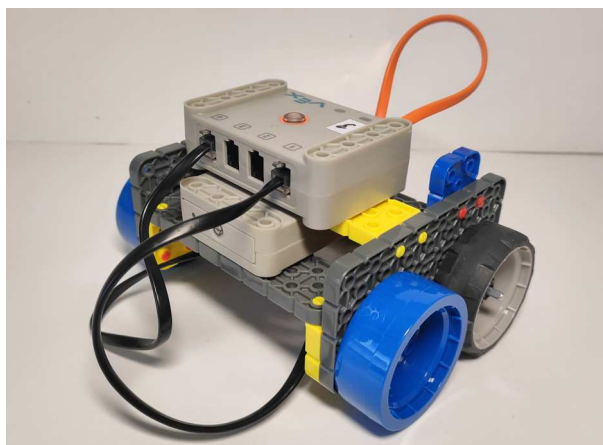


Obrázek 2: *Sestavený robot LEGO Mindstorms (již s připojenými motory k řídicí jednotce)*

Základního pojízdného robota ze stavebnice LEGO Mindstorms lze postavit dle manuálu. Tento návod se nachází v papírové podobě v krabici s dílky a také v původní učebnici, kde je odkaz na dokument ve formátu PDF, který obsahuje 2D manuál, jak postavit základního robota viz obrázek č. 2 výše.

5.1.2 Stavba robota – stavebnice VEX

Základní podoba pojízdného robota ze stavebnice VEX GO byla postavena během 10 minut. Ke kratšímu časovému údaji pravděpodobně přispěla minimalistická kostra robota, na kterou bylo potřeba využít méně dílků než u stavebnice LEGO.



Obrázek 3: *Sestavený robot VEX GO (již s připojenými motory k řídicí jednotce)*

VEX ke stavbě základního pojízdného robota, který je dle oficiálních stránek u stavebnice VEX GO nazvaný „Code Base“ nabízí dvě možnosti manuálu. První manuál je 2D a je podobný jako u LEGO Mindstorm. Oproti LEGO nabízí VEX manuál ve 3D. Tento virtuální manuál ukazuje krok po kroku, jak má stavitel pokračovat. Uživatel zde může v daném kroku otáčet vizualizací, přibližovat, oddalovat a na konci každého kroku se zobrazí animace připojování daných dílků. Tyto manuály jsou dostupné buď na stránkách „vexrobotics“, nebo se lze odkázat na manuály skrze programovací prostředí, kde se nachází záložka „konstrukce“.

5.1.3 Shrnutí kapitoly 1

V této kapitole nebylo potřeba provést žádné změny oproti původní učebnici, jelikož se kapitola primárně zabývá stavěním pojízdného robota. Samotná stavba robota je u obou stavebnic stejná, a proto lze tuto kapitolu zachovat i u stavebnice VEX GO. Jedinými rozdíly v této kapitole jsou možnosti výběru manuálu a rozdíl v době stavby základního robota. Což pravděpodobně vypovídá o tom, že stavebnice VEX GO je primárně určena pro první stupeň základní školy na rozdíl od stavebnice LEGO, která je určena na stupeň druhý.

5.2 Kapitola 2: Oživení pojízdného robota

Druhá kapitola plynule navazuje na předchozí úvodní kapitolu. V první kapitole bylo primárním cílem postavit základního pojízdného robota, se kterým se ve druhé kapitole dále pracuje. Na začátku se žáci seznámí s tím, jak připojit motory k řídicí jednotce a jak propojit robota s počítačem.

Následně se žáci seznámí se základními pohyby pojízdného robota včetně jeho řízení. Prozkoumávají blokově orientované programovací prostředí přizpůsobené přímo pro danou stavebnici. V tomto prostředí se seznamují s vytvářením základního kódu pomocí bloků (v původní učebnici řazených horizontálně), změnou parametrů a celkovou strukturou prostředí. Podle metodických listů původní učebnice je předpokládána časová dotace pro druhou kapitolu dvě vyučovací hodiny.

5.2.1 Oživení robota – původní učebnice

V první části druhé kapitoly, v úloze 2.1 žáci připojí motory pomocí kabelů k řídicí jednotce. Vstupní porty na řídicí jednotce jsou označeny písmeny A až D. Poskládaný a připojený robot je vyobrazen na fotografii v předchozí kapitole 5.1.1 Stavba robota – původní učebnice.

„Začínáme – první program“ je název následující úlohy, jež nese číslo 2.2 a je zaměřena na první pohyby robota. V této části se nachází již vytvořený kód od autorů učebnice, který je v původní učebnici ke stažení.



Obrázek 4: První program – úloha 2.2

V úloze 2.3 mají žáci první program z předchozí úlohy nahrát do robota. Zde si žáci prakticky sami vyzkouší, jak robota připojit k počítači a zkontrolují správně připojené motory a senzory. Dále následuje úloha 2.4 která je sérií dílčích úkolů zahrnující tři úkoly a jednu hádanku. Část s názvem „Učíme robota jezdit“ vychází z prvního programu od autorů. Tyto úkoly jsou postavené především na otázkách týkajících se funkčnosti kódu a úprav parametrů pro řízení robota.

Úkoly 2.4.1 a 2.4.3 vychází z kódu vyobrazeného výše, a proto není třeba jej znovu uvádět. V těchto úkolech se ke kódu vážou otázky: „Co se stane, pokud motory připojíte k daným portům dle obrázku, program nahrajete do řídicí jednotky a spustíte?“, „Co musíte v programu změnit, aby robot ujel pouze polovinu této vzdálenosti?“ Úkol 2.4.2 má mírně upravený předchozí kód a váže se k němu otázka: „V čem se tento program liší od předchozího a jaký pohyb robot vykoná nyní?“



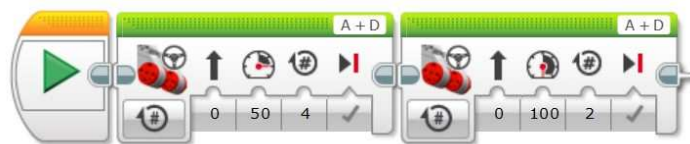
Obrázek 5: Kód k úkolu 2.4.2

V hádance 2.4.4 mají žáci přijít na to, proč robot po spuštění kódu vyobrazeného níže nevykoná žádný pohyb. Zde se nachází jistý chyták, který se skrývá v nastavení řízení pohybů robota. Je zde nastaveno, že kola robota mají provést otočení motoru o jeden stupeň (ačkoli robot příkaz vykoná, pohyb je pro lidské oko neviditelný).



Obrázek 6: Kód k hádance 2.4.4

V prostředí druhé kapitoly se nachází souhrnná úloha 2.5, kde mají žáci vytvořit vlastní kód. Robot musí ujet vzdálenost čtyř otáček kol na rychlost motoru 50 % a následně na další dvě otáčky zrychlí na 100 %.



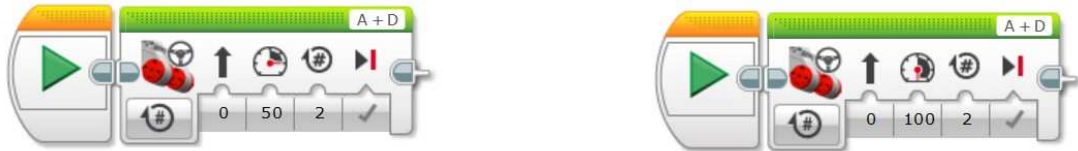
Obrázek 7: Řešení úlohy 2.5

Úloha 2.6 také obsahuje sérii dílčích úkolů a hádanek. Obsaženy jsou zde především otázky týkající se vyobrazených kódů, včetně úkolů obsahujících chytáky k zamyšlení. Úkol 2.6.1: „*Jaký je rozdíl mezi programy na obrázku?*“, „*Pomocí kterého programu robot ujede delší vzdálenost?*“ Především se klade důraz na pochopení hodnoty parametrů v blocích, v kombinaci s režimy řízení robota. Zde se nachází téměř totožné bloky, avšak s rozdílnou hodnotou parametru a ve dvou režimech pro řízení otáčení motoru. Nicméně oba kódy mají totožný význam.



Obrázek 8: Kódy k úkolu 2.6.1

Úkol 2.6.2: „*Ve kterém případě robot urazí delší vzdálenost? Dokážete najít odpověď jen na základě obrázku?*“, „*Jaký je rozdíl mezi oběma programy?*“ Kódy opět vypadají téměř totožně, tentokrát však s rozdílným parametrem u nastavené rychlosti motorů.



Obrázek 9: Kódy k úkolu 2.6.2

Hádanka 2.6.3: „Kolik centimetrů robot ujede, pokud spustíme program na obrázku?“ Výsledkem je vzdálenost 34-36 cm. To mohou žáci zjistit změřením ujeté vzdálenosti (např. pravítkem) nebo výpočtem z obvodu kola (obvod = 17,5 cm).



Obrázek 10: Kód k hádance 2.6.3 a 2.6.4

Hádanka 2.6.4 se váže ke kódu vyobrazenému nad textem: „Jak zajistíte, aby stejnou vzdálenost, jako v předchozí hádance, ujel robot zadáním ve stupních?“ Řešením je přepnutí režimu řízení a nastavení hodnoty 720°.

Úkol 2.6.5 se týká vytvoření vlastního jednoduchého programu, aby robot ujel co nejpřesněji 30 cm. Řešení tohoto úkolu:



Obrázek 11: Řešení úkolu 2.6.5

Úloha 2.7 v původní učebnici je označena jako náročnější. Žáci zde mají vytvořit kód na základě znalostí naučených v předchozích úlohách. Znění úlohy: „Vytvořte program, pomocí kterého robot ujede vzdálenost 5 otáček kol, poté se otočí o 90° vpravo a následně ujede dalších 5 otáček.“ Testováním úlohy v rámci praktické části bylo zjištěno, že pro otočení robota o 90° stačí jedna otočka jednoho z motorů.



Obrázek 12: Řešení pokročilejší úlohy 2.7

Úloha 2.8 je pro rychlejší žáky a je založena na třech úkolech. Úkol 2.8.1: „Zařídte, aby robot ujel 15 cm, pak na 3 vteřiny zastavil, poté se znovu rozjel a ujel 15 cm.“ Tato úloha je ve své podstatě jednoduchá, jelikož lze vycházet z úkolu 2.6.5, kde byl vytvořen kód, aby robot ujel vzdálenost 30 cm. V tomto případě stačí počet otáček vydělit dvěma a získáme tak vzdálenost 15 cm. Mezi bloky, které řídí motory, byl vložen blok s pauzou tří sekund.



Obrázek 13: Řešení úkolu 2.8.1

Úkol 2.8.2: „Zařídte, aby robot ujel 30 cm, následně zastavil a zacouval zpět na počáteční pozici.“ Kód pro tento úkol byl vytvořen principiálně totožně jako vyobrazený kód výše. Byla zde vložena jedna sekunda pro vyčkání, aby bylo patrné, že robot opravdu zastavil. Ve třetím bloku byla nastavena záporná hodnota, aby robot couval na původní pozici.



Obrázek 14: Řešení úkolu 2.8.2

V úkolu 2.8.3: „Zařídte, aby robot ujel 30 cm, otočil se o 180° a dojel zpět na výchozí pozici.“ Pro tento úkol bylo využito znalostí z předchozích úkolů, a proto nebyl problém tento kód sestavit. Pro otočení o 180° bylo po několika testováních zvoleno 2,1 otočky jednoho z motorů, aby úhel byl co nejpřesnější.



Obrázek 15: Řešení úkolu 2.8.3

Druhá kapitola je uzavřena závěrečnou úlohou 2.9. Zde žáci programují robota, aby ujel vyobrazenou dráhu, která je znázorněna v původní učebnici obrázkem. Slovně lze tuto dráhu popsat následovně: šest otáček dopředu na rychlost 100 %,

otočit se o 90° doleva, čtyři otáčky rovně na rychlost 60 %, otočit se o 90° doprava a dvě otáčky rovně na rychlost 30 %.



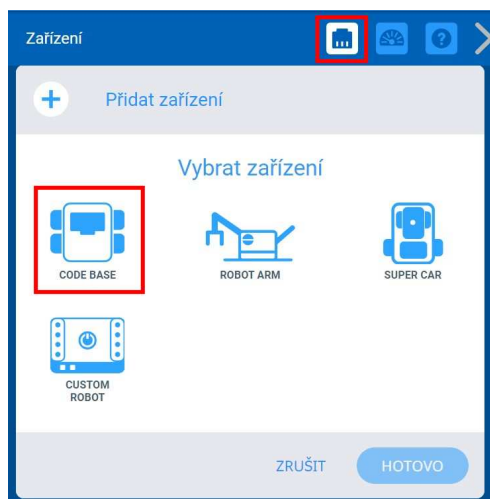
Obrázek 16: Řešení úlohy 2.9

5.2.2 Oživení robota – stavebnice VEX

Začátek kapitoly lze ztotožnit u obou stavebnic. Žáci mají v první řadě připojit motory k řídicí jednotce. Vstupní porty na řídicí jednotce jsou označeny čísly jedna až čtyři. Sestavený a připojený robot je vyobrazen v předchozí kapitole 5.1.2 Stavba robota – stavebnice VEX.

Vhodné je zmínit, že u stavebnice VEX GO je po připojení motorů k řídicí jednotce nutné připojit samotnou řídicí jednotku k baterii. Baterie je v konstrukci robota zakomponována jako samostatný komponent (není součástí řídicí jednotky).

Aby bylo možné robota začít programovat, bylo nutné v programovacím prostředí nastavit používané komponenty. Ve VEXcode GO se nachází tři ikony, přičemž kliknutím na první ikonu (připomínající lego kostky) se zobrazila nabídka „Přidat zařízení“. Poté bylo potřebné zvolit typ konstrukce. V první kapitole bylo sestaveno vozidlo zvané Code Base, a proto bylo vybráno stejnojmenné zařízení. To mělo za následek, že v panelu s programovacími bloky se automaticky přidaly nové bloky potřebné k řízení robota (které doposud v panelu nebyly).



Obrázek 17: Nabídka pro zvolení – zařízení, komponentů

V původní učebnici je první program z úlohy 2.2 sestaven v blokovém prostředí, kde se bloky skládají do řádku (horizontálně). VEXcode GO využívá skládání do sloupce (vertikálně). V tomto prostředí se používají odlišné bloky. Proto prvotní program od autorů by ve VEXcode GO mohl vypadat například takto:



Obrázek 18: První kód v podobě vertikálního blokového prostředí – úloha 2.2

V úloze 2.3 bylo cílem spustit prvotní program a oživit tak postaveného robota z první kapitoly. V následující úloze 2.4 se nachází sada tří úkolů a jedné hádanky. Úkol 2.4.1: „Co se stane, pokud motory připojíte k daným portům dle obrázku, program nahrajete do řídicí jednotky a spustíte?“ Otázka se vztahuje ke kódu vyobrazenému výše (kód z úlohy 2.2). U VEX GO však z kódu není zřejmé, ke kterým portům mají být motory připojeny. Proto je žádoucí v daných případech upravit zadání tak, aby bylo patrné ke kterým portům mají být motory připojeny.

Úkol 2.4.2: „V čem se tento program liší od předchozího? A jaký pohyb robot vykoná nyní?“ Kód je odlišný pouze v parametru rychlosti pohonu. Jelikož je uvedena záporná hodnota, robot pojedou dozadu.



Obrázek 19: Upravený parametr pohonu – úloha 2.4.2

Úkol 2.4.3: „Pomocí programu na obrázku ujede robot určitou vzdálenost. Co musíte v programu změnit, aby robot ujel pouze polovinu této vzdálenosti? Vymyslete

řešení a otestujte ho.“ Tato otázka se vztahuje ke kódu uvedenému pod textem a nabízí dvě možnosti řešení. Jednou z možností je změnit rychlost pohonu na polovinu anebo upravit čas tak, aby byl poloviční oproti původnímu času.



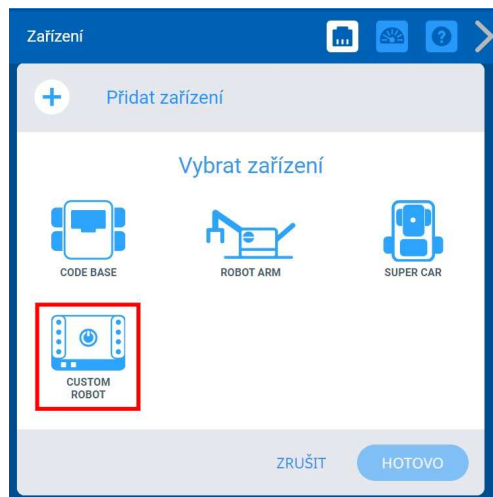
Obrázek 20: Upravený parametr času – úloha 2.4.3

Hádanka 2.4.4: „Nyní pro vás máme jednu hádanku. Proč se robot po spuštění tohoto programu nerozjede?“ Tato úloha byla zásadní pro další práci se stavebnicí VEX GO. Neboť přidané zařízení „Code Base“ neumožňuje řízení motoru pomocí otáček, ani v režimu řízení pomocí úhlových stupňů, jak je tomu u stavebnice LEGO v původní učebnici. Zařízení „Code Base“ nabízí v blocích pohybu motorů pouze tři režimy: nepřetržitý pohyb vpřed, pohyb do kontaktu s překážkou nebo ujetí zadané vzdálenosti v milimetrech či palcích.

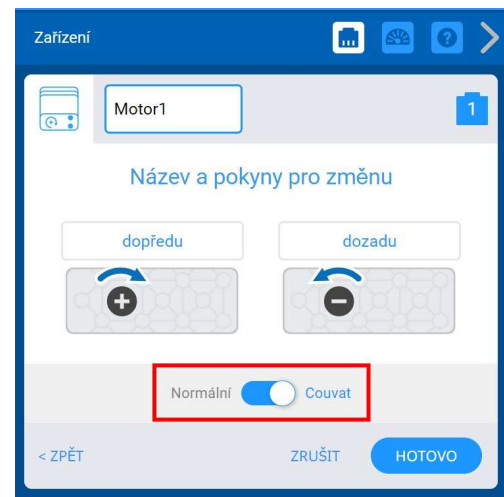
Bylo zjištěno, že při přidání zařízení „Custom Robot“ místo „Code Base“ lze ovládat oba motory nejen samostatně, ale také v požadovaných otáčkách a stupních. V levém panelu se automaticky zobrazily bloky pro ovládání jednotlivých motorů.

Posledním důležitým krokem při přidání „Custom Robot“ byla úprava směru otáčení motorů. Kvůli zrcadlové konstrukci motorů se jeden motor otáčel opačně, což způsobovalo otáčení robota na místě. Proto bylo nutné v aplikaci nastavit, aby se jeden motor pohyboval vpřed a druhý v opačném směru.

Nevýhodou přidáním zařízení „Custom Robot“ pro nastavení jednotlivých komponentů je, že nelze pracovat s oběma motory současně jedním blokem jako tomu bylo u zařízení „Code Base“. Tato problematika je podrobněji vyobrazena na další straně a v následujících řešeních úloh praktické části práce.

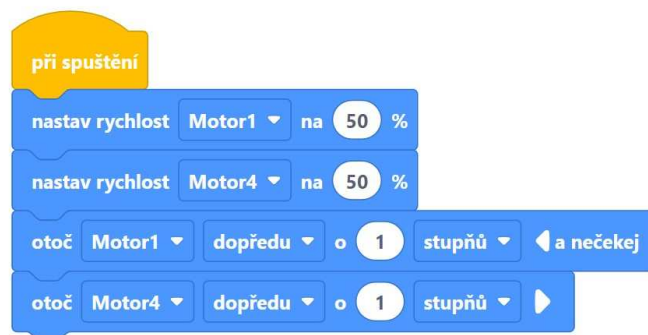


Obrázek 21: Přidání zařízení –
Custom Robot



Obrázek 22: Nastavení opačného
otáčení motoru

Níže na obrázku č. 23 je zpracování hádanky 2.4.4 pomocí otáčení motoru úhlovým stupněm. Vyjma toho je k povšimnutí, že každý motor je řízen vlastním blokem. Ovšem pro správnou funkčnost kódu a chod robota je důležité, aby u prvního bloku pohánějící jeden z motorů bylo zaškrtnuto „a nečekej“. Tím se zajistí, že otáčení dvou motorů proběhne současně.

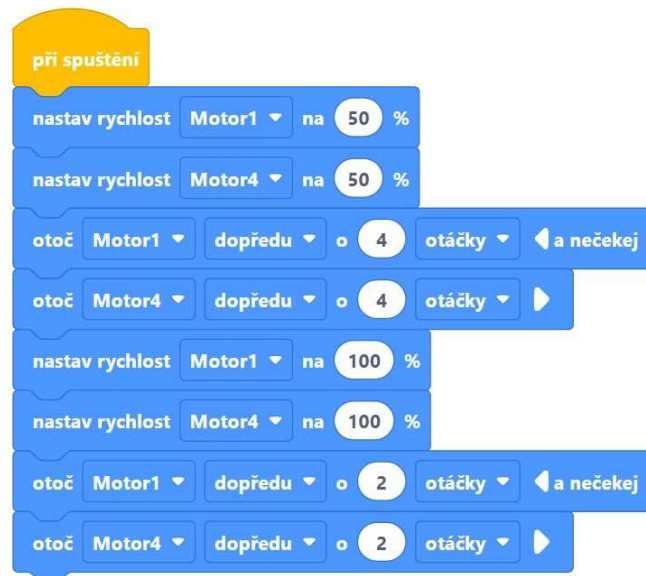


Obrázek 23: Kód k hádance 2.4.4

Uprostřed druhé kapitoly se nachází úloha 2.5 s názvem „střídáme rychlosti“. V LEGO Mindstorms lze úlohu vyřešit pomocí dvou bloků a může se jevit méně náročným. Každý z těchto bloků však obsahuje čtyři parametry a ovládá oba motory najednou, což znamená, že změna rychlosti se nastavuje pro oba motory současně v rámci jednoho bloku.

Naopak ve VEXcode GO je potřeba použít osm bloků, protože každý blok nastavuje pouze jeden parametr a ovládá pouze jeden motor. Pro každý motor jsou tedy potřebné čtyři bloky s různými parametry. Ve VEXcode GO je důležité dbát

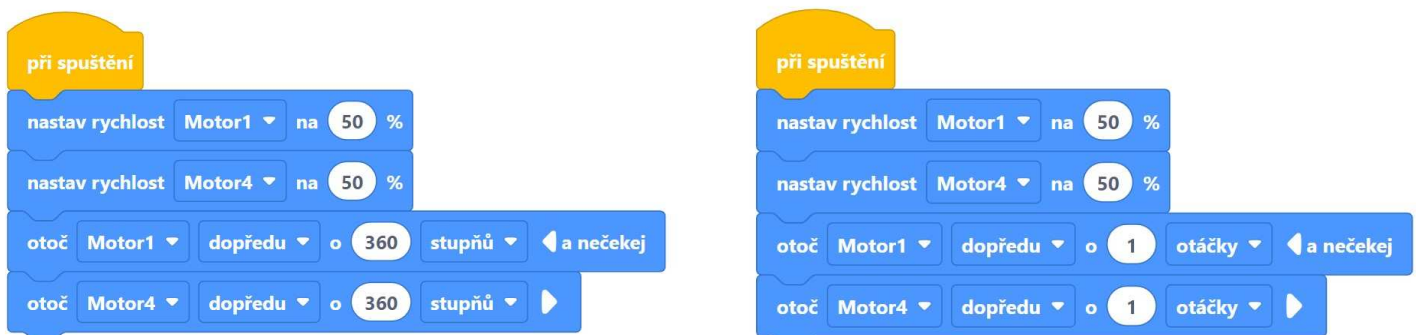
na to, aby vždy jeden blok ovládající jeden z motorů měl nastaveno „a nečekej“. To je nutné k zajištění toho, aby se robot pohyboval směrem vpřed a neotáčel se na místě.



Obrázek 24: Řešení úlohy 2.5

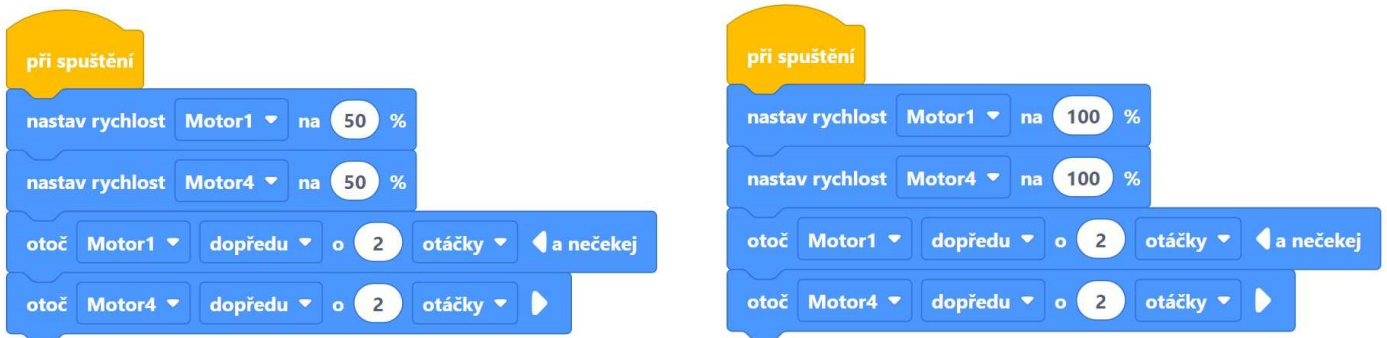
Série úkolů v úloze 2.6 „učíme robota vzdálenost“ obsahuje tři úkoly a dvě hádanky, přičemž jeden z úkolů je určen pro rychlejší žáky. Vzhledem k tomu, že autoři učebnice v této kapitole často využívají funkci rotací v otáčkách či stupních, jsou dále uváděny kódy v režimu přidaného zařízení „custom robot“.

Úkol 2.6.1: „Jaký je rozdíl mezi programy na obrázku?“, „Pomocí kterého programu robot ujede delší vzdálenost?“ I zde je nutné použít již zmíněný blok s nastavenou hodnotou „a nečekej“.



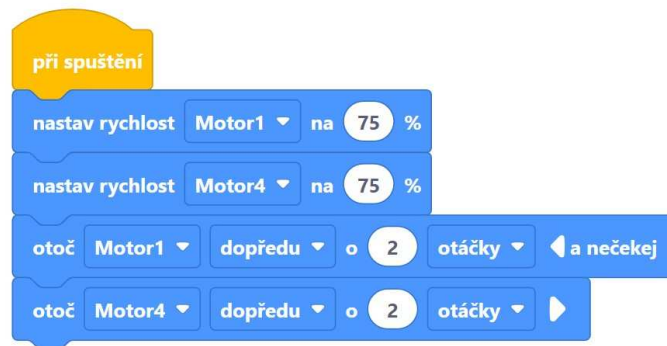
Obrázek 25: Kódy k úkolu 2.6.1

Úkol 2.6.2: „Ve kterém případě robot urazí delší vzdálenost? Dokážete najít odpověď jen na základě obrázku?“, „Jaký je rozdíl mezi oběma programy?“



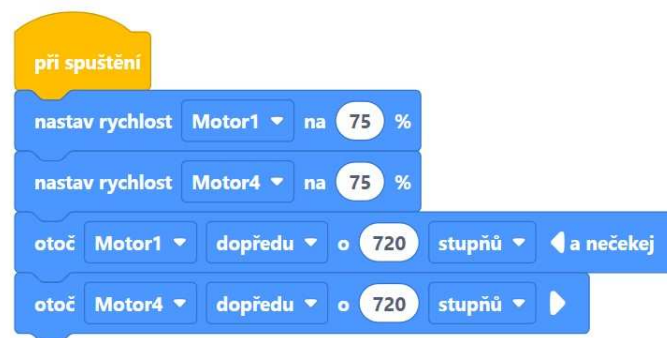
Obrázek 26: Kódy k úkolu 2.6.2

Hádanka 2.6.3: „Kolik centimetrů robot ujede, pokud spustíme program na obrázku?“ (Rozšiřující: „Jakým způsobem by se to dalo zjistit i bez jeho spuštění?“) Robot na dvě otáčky kol ujede vzdálenost cca 30 cm.



Obrázek 27: Kód k hádance 2.6.3

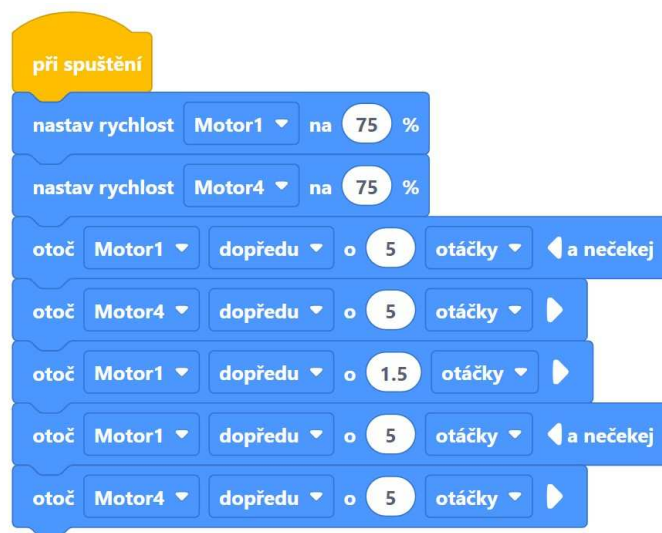
Hádanka 2.6.4 zadává úkol, jak by mohl vypadat kód, zadaný ve stupních, aby robot ujel stejnou vzdálenost jako v předchozím kódu.



Obrázek 28: Kód k hádance 2.6.4

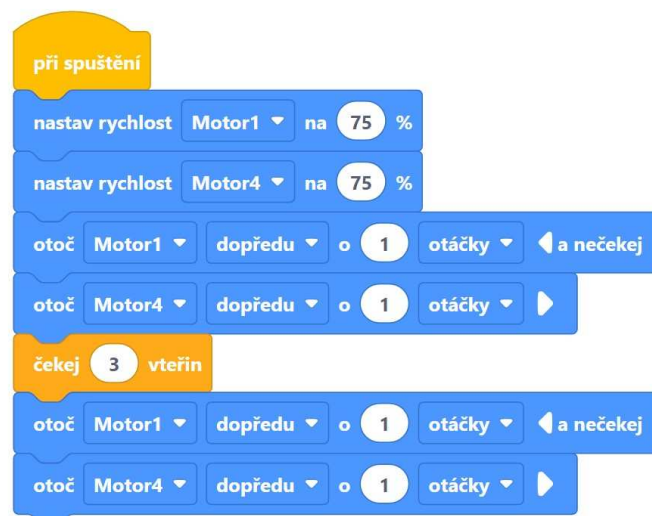
Úkol 2.6.5 pro rychlejší žáky: „Vytvořte program, pomocí kterého robot ujede co nejpřesněji 30 centimetrů.“ Řešení pro tuto úlohu není třeba uvádět. Pro stavebnici VEX GO odpovídá vzdálenost 30 cm dvěma otáčkám motorů, jelikož obvod kol robota je 15 cm. Kód pro tento program je totožný jako kód výše u hádanky 2.6.3 v otáčkách anebo u hádanky 2.6.4 ve stupních.

Kapitola pokračuje náročnější úlohou 2.7. „Vytvořte program, pomocí kterého robot ujede vzdálenost 5 otáček kol, poté se otočí o 90° vpravo a následně ujede dalších 5 otáček.“ Pro tuto úlohu byla zvolena strategie, kdy se po zastavení obou motorů následně začne otáčet pouze jeden, dokud robot není v požadovaném úhlu.



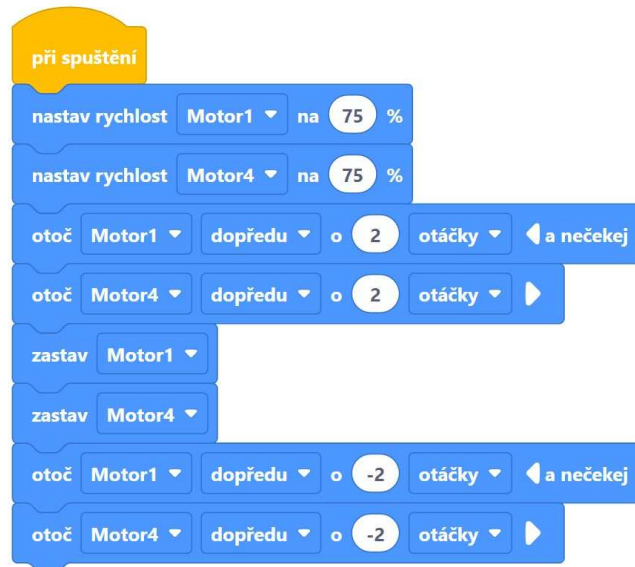
Obrázek 29: Řešení úlohy 2.7

Úloha 2.8 pro rychlejší žáky začíná úkolem 2.8.1: „Zařídte, aby robot ujel 15 cm, pak na 3 vteřiny zastavil, poté se znovu rozjel a ujel 15 cm.“



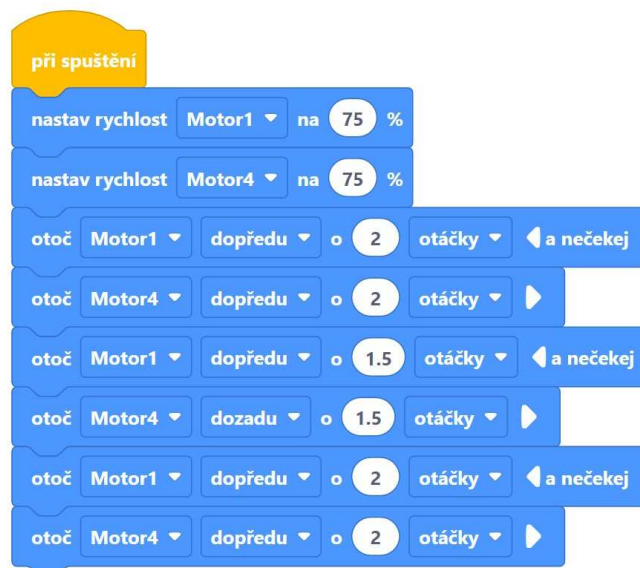
Obrázek 30: Řešení úlohy 2.8.1

Úkol 2.8.2: „Zařídte, aby robot ujel 30 cm, následně zastavil a zacouval zpět na počáteční pozici.“



Obrázek 31: Řešení úlohy 2.8.2

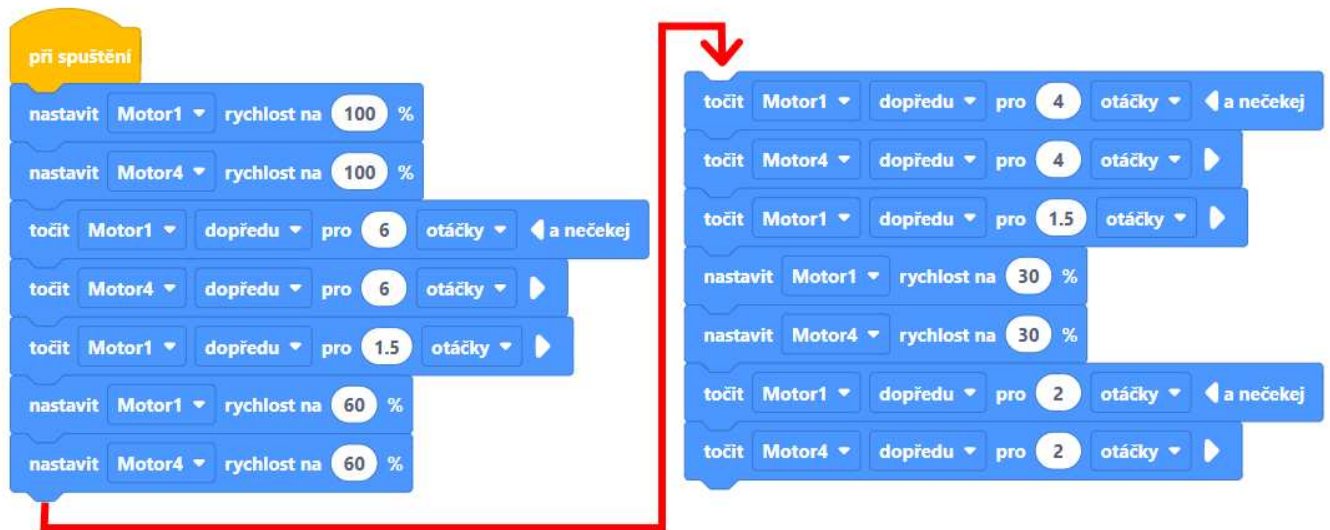
Úkol 2.8.3: „Zařídte, aby robot ujel 30 cm, otočil se o 180° a dojel zpět na výchozí pozici.“ V tomto úkolu byla zvolena strategie, kdy po ujetí 30 cm robot zastavil a otočil se oběma motory současně, čímž se robot otočil o 180° na místě.



Obrázek 32: Řešení úlohy 2.8.3

Úloha 2.9 závěrečné opakování: „Naprogramujte robota tak, aby projel následující dráhu, a to podle zadaných pokynů o délce i rychlosti pohybu.“ Vyobrazená dráha v původní učebnici by se slovně dala popsat takto: Šest otáček dopředu na

rychlost 100 %, otočit se o 90° doleva, čtyři otáčky na rychlost 60 %, otočit se o 90° doprava, dvě otáčky na rychlost 30 %. Ačkoli se kód v této úloze může jevit příliš zdlouhavý, nebylo nalezeno kratší řešení. Délku kódu zde ovlivňuje nutnost nastavování rychlosti a počtu otáček pro každý motor zvlášť.



Obrázek 33: Rozložené řešení úlohy 2.9

5.2.3 Shrnutí kapitoly 2

Celou kapitolu se podařilo převést z původní učebnice, a to i přes drobné obtíže s prvotním nastavením. Nebylo potřebné navrhovat žádné nové úlohy.

Oživení stavebnice VEX GO je zdlouhavý proces zahrnující několik kroků, než se může začít s robotem pracovat.

LEGO Mindstorms na prvotním stažení kódu z původní učebnice, při nastavené rychlosti motoru 75 % a době otáčení čtyř sekund ujel vzdálenost 137 cm. Zatímco VEX GO na stejnou rychlost motoru (75 %) a stejný čas (čtyři sekundy) ujel vzdálenost pouze 67 cm. To znamená, že VEX GO na stejné nastavení hodnot ujede přibližně poloviční vzdálenost oproti LEGO Mindstorms.

Kódy ve VEXcode GO působí zdlouhavým dojmem oproti stavebnici LEGO. Je to dáno především tím, že při tvorbě kódu u LEGA se využívá méně bloků, které mají více nastavitelných parametrů. Vezmeme-li v úvahu, že kdyby bylo možné parametry imaginárně rozdělit do jednotlivých bloků jako je tomu u stavebnice VEX, dostali bychom podobně dlouhé kódy u obou stavebnic. Dále pro možnost

práce u VEX GO s motory v režimu otáčení ve stupních či otáčkách, je potřebný blok pro každý motor zvlášť. U nastavení zařízení „Code Base“ lze ovládat oba motory současně, avšak za cenu nemožnosti pracovat s otáčkami a stupni otáčení.

Při práci s motory v nastavení „Custom Robot“ jak je dáno v kódech této kapitoly, je velkou nevýhodou nutnost správného nastavení bloků s prvkem „a nečekej“. Tento lehce přehlédnutelný fakt zásadně ovlivňuje funkčnost celého kódu, avšak pro jeho správné fungování je velmi podstatný.

V celkovém shrnutí je nutné podotknout, že kódy druhé kapitoly lze převést do stavebnice VEX GO, avšak nejedná se o nejvhodnější způsob, především z důvodu zbytečně zdlouhavých kódů.

5.3 Kapitola 3: Robot ve městě

Žáci v první kapitole postavili základní vozidlo, které se naučili ovládat v druhé kapitole. Ve třetí kapitole stále využívají stejného robota a zároveň aplikují naučené znalosti z druhé kapitoly. Jak už sám název kapitoly napovídá, žáci programují robota v několika úlohách tak, aby robot projížděl určené dráhy na plánku města. Tento plánek je možné vytisknout na velkoformátový papír z původní učebnice.

Žáci v této kapitole kombinují různé směry pohybu robota (dopředu, dozadu, otáčení o různé úhly na obě strany či pohyb po kružnici). Dle metodických listů je tato kapitola určena na dvě vyučovací hodiny.

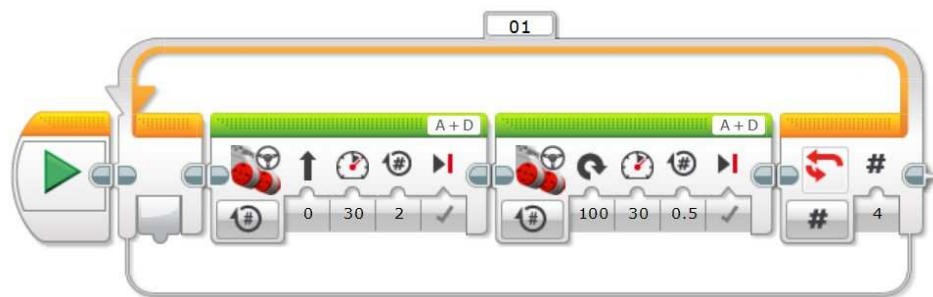
V kapitole se nachází úlohy A-H. Většina úloh je postavena na základních pohybech a kódech, které se objevily v druhé kapitole praktické části. Není proto třeba uvádět veškeré úlohy z této kapitoly. Pro ukázkou byla zvolena pouze jedna úloha, kdy žáci poprvé musí použít blok pro opakování.

5.3.1 Robot ve městě – původní učebnice

V této kapitole úloha C jako jediná nabízí nový poznatek, který žáci v minulé kapitole nevyužívali. Znění z původní učebnice této úlohy je: *„Při projíždění ulicí nebylo možné nikde zaparkovat. Před nemocnicí je jen jedno velmi malé parkovací místo. Objedte dokola celý areál nemocnice, třeba se vám místo k parkování podaří najít. Pokud ne, zaparkujte na místě C.“*

Jelikož budova nemocnice na plánu města je ve tvaru čtverce, mají žáci u této úlohy tip na řešení. Tím je, že některé kroky se při objíždění daného objektu opakují, a proto je možné využít blok pro opakování. Je tedy žádoucí, aby žáci naprogramovali kód tak, aby robot ujel trasu ve tvaru čtverce a to za předpokladu využití co nejméně bloků.

Vhodné pro tento kód je poznamenat způsob zadávání počtu opakování v původním softwaru. Počet opakování se nezadá v bloku do horní kolonky s číslem (na ukázce číslo 01), ale tuto hodnotu je potřebné nastavit v bloku vpravo dole (na obrázku č. 34 číslo čtyři vpravo dole).

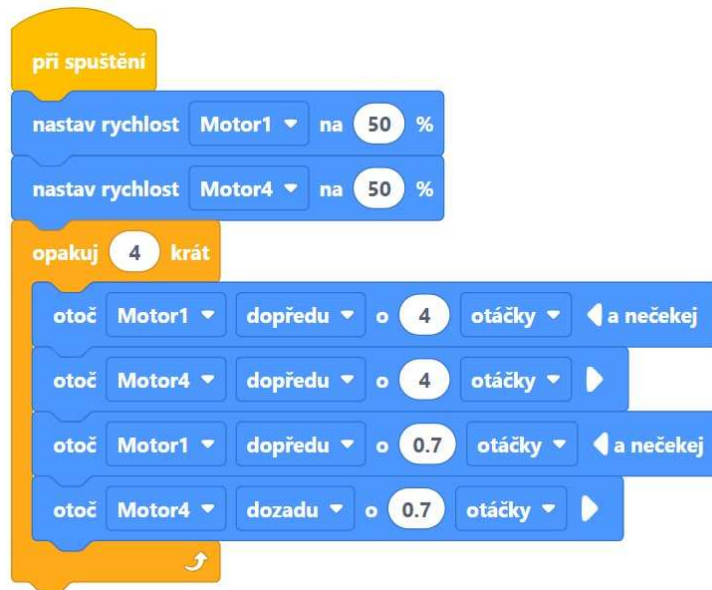


Obrázek 34: Řešení úlohy C u stavebnice LEGO

5.3.2 Robot ve městě – stavebnice VEX

Všechny úlohy ze třetí kapitoly není potřeba vyobrazovat, jelikož, jak bylo zmíněno na začátku této kapitoly, úlohy nepřinášejí nic nového oproti druhé kapitole. Níže je vyobrazena úloha C s ukázkou bloku opakování. Uvedena je především pro porovnání s původní aplikací pro stavebnici LEGO.

Pro tuto úlohu bylo zvoleno řešení otočení robota o 90° pomocí otáčení obou motorů současně v opačných směrech. Nejprve byla při vytváření úlohy testována strategie otáčení pomocí jednoho motoru. V tomto případě se řešení s jedním motorem ukázalo jako neefektivní. Jelikož na vícero površích, různého materiálu se stávalo, že robotovi se jedno dané kolo protáčelo na místě a robot se tak neotočil o požadovaných 90° . Různými typy povrchů, na kterých byla úloha testována, se myslí například lino, koberec, dřevěný stůl.



Obrázek 35: Řešení úlohy C u stavebnice VEX

5.3.3 Shrnutí kapitoly 3

Třetí kapitola byla v původní učebnici vytvořena především pro ukotvení znalostí z předchozí kapitoly.

V této práci byla pro ukázkou zvolena pouze jedna úloha, jelikož byla první úlohou, kde žáci mají za úkol využít blok opakuj. Dle ukázek je vidět, že obě programovací prostředí nabízí totožnou možnost využití tohoto bloku. Mezi několika poznatky lze zmínit odlišné nastavení opakování pro blok opakuj. Zatímco u stavebnice VEX je blok opakování intuitivní a lze zde nastavit pouze počet opakování, v bloku opakuj pro stavebnici LEGO lze pracovat s více možnostmi nastavení bloku pro opakování. Subjektivním poznatkem je, že v původní aplikaci blok opakuj zabírá zbytečně graficky více místa v programovacím prostředí.

Ačkoli je v programování žádoucí mít kód co nejkratší, pro kód u stavebnice VEX není zvolena nejkratší možnost. To je dáno tím, že otočení robota o 90° pomocí jednoho motoru, tzn. jednoho bloku, na površích jako lino, koberec anebo stůl nefungovalo zcela správně. Při zvolené strategii otáčení robota, kdy jeden motor stojí a otáčí se pouze druhý, se opakovaně stávalo, že se kolo protáčelo na místě a robot se tak neotočil do požadovaného úhlu. A to i přesto, že tato situace při testování úloh z první kapitoly nenastala. Nakonec bylo zvoleno otáčení dvou motorů současně v protichůdných pohybech.

5.4 Kapitola 4: Zvuk a displej

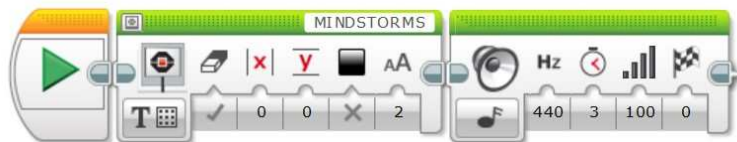
Kapitola čtyři se oprostuje od pojízdného robota a má za úkol naučit žáky pracovat s displejem a zvukem, které jsou součástí řídicí jednotky stavebnice LEGO. Podle metodických listů, je tato kapitola opět určena na dvě vyučovací hodiny.

Na začátku čtvrté kapitoly je detailně představena řídicí jednotka stavebnice LEGO Mindstorms. Žáci v této kapitole především testují, upravují již vytvořené kódy a v mnoha případech této kapitoly odpovídají na otázky týkající se těchto kódů. Taktéž mají za úkol kódy upravovat a prakticky testovat.

5.4.1 Zvuk a displej – původní učebnice

Úlohy 4.1 a 4.2 jsou spíše teoretickou částí kapitoly, kde je detailně popsána hlavní komponenta stavebnice, kterou je řídicí jednotka. Jsou zde popsána tlačítka, vstupy, baterie, displej, paměť na základní desce, reproduktor a procesor. Tyto části jsou doplněny fotografiemi rozebrané řídicí jednotky.

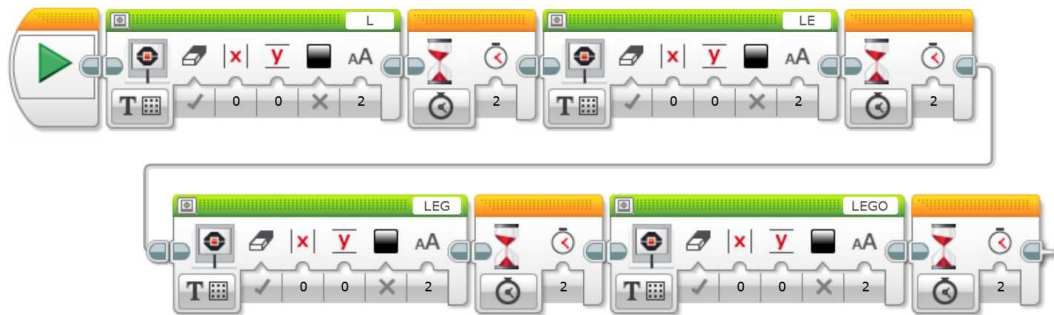
Úloha 4.3 je zaměřena na displej a začíná odkazem ke stažení kódu, se kterým žáci dále pracují.



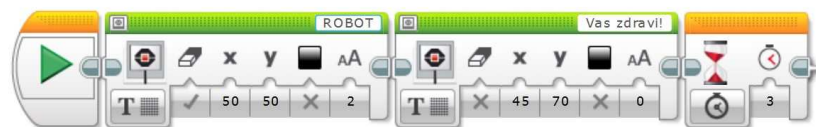
Obrázek 36: Kód k úloze 4.3

Úloha 4.3 obsahuje dílčí úkoly 4.3.1-4.3.5, které jsou spíše doplňujícími otázkami k výše vyobrazenému kódu na obrázku č. 36. V úkolech žáci odpovídají na otázky, testují své domněnky a je jim teoreticky představen blok „čkej“.

Kapitola pokračuje rozčleněnou úlohou 4.4, ve které žáci upravují předchozí program. Úkol 4.4.1 využívá nového bloku „čkej“, který byl představen v předchozí úloze: „Upravte program tak, aby text zůstal na displeji zobrazený po dobu 3 vteřin.“ Úkoly 4.4.2 a 4.4.3 se blíže zabývají výpisem textu na displej řídicí jednotky a tvorbou daného programu.



Obrázek 37: Řešení úkolu 4.4.2



Obrázek 38: Řešení úkolu 4.4.3

Úloha 4.5 je počáteční úlohou pro práci se zvukem. U této úlohy je v učebnici tlačítko ke stažení kódu, se kterým žáci dále pracují v úloze 4.6. Kód, jak již je napovězeno v zadání úkolu 4.6.1 („Spustte soubor. Měla by se vám přehrát stupnice, která je však pokažená.“), je vytvořen záměrně s chybami. Žáci mají za úkol program spustit a najít chyby v kódu. Následně mají v úkolu 4.6.2 nalezené chyby opravit. Úkol 4.6.3 se zabývá porozumění parametrům u bloku se zvukem.

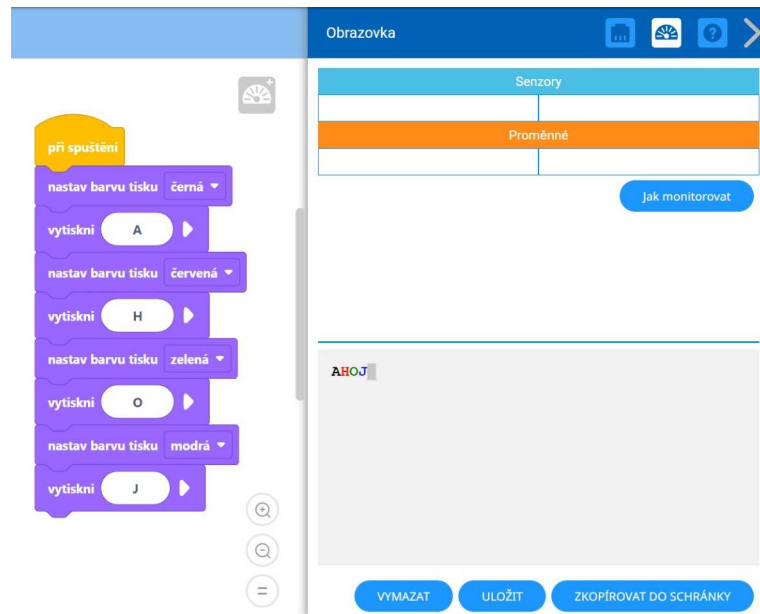
Kódy pro tyto úkoly nejsou dále vyobrazeny, protože stavebnice VEX GO neobsahuje reproduktor a nebyly vyhodnoceny jako podstatné pro tuto práci.

5.4.2 Výpis a gyroskop (alternativa) – stavebnice VEX

Tato kapitola musela být pro stavebnici VEX předělaná a musely být navrženy alternativní úlohy, jelikož stavebnice VEX GO není vybavena ani displejem, ani reproduktorem. VEX GO nabízí alternativu v podobě výpisu na obrazovku zařízení, na kterém uživatel programuje. Možnost výpisu na obrazovku počítače je dohledatelná ve VEXcode GO v záložce „obrazovka“. Mimo jiné řídicí jednotka VEX GO nabízí možnost práce s gyroskopem nebo dokáže snímat gravitační zrychlení.

Návrh první úlohy může být podobný té v původní učebnici, ovšem s rozdílem, že text se vypíše na obrazovku zařízení namísto displeje robota. K úloze stačí řídicí

jednotka s baterií. Úlohou může být jednoduchý výpis na obrazovku v prostředí VEXcode GO. Navržené slovní zadání pro úlohu: „Vytvořte program, který na obrazovku vypíše slovo AHOJ, přičemž každé písmeno bude jinak barevné.“



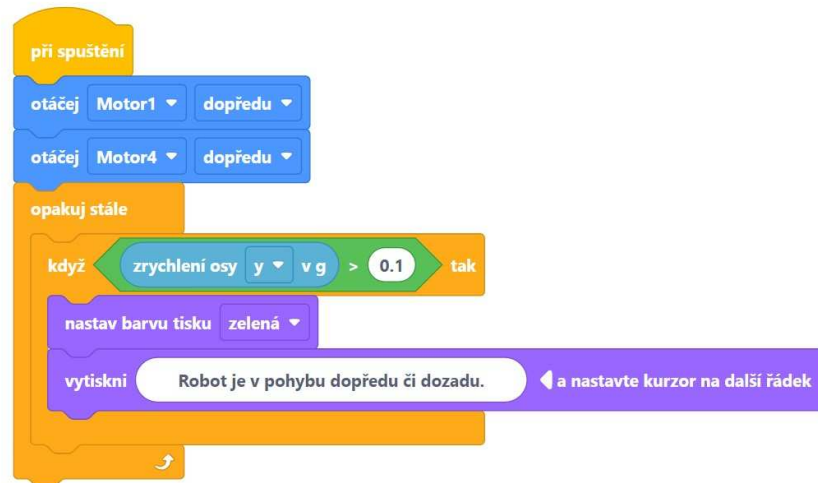
Obrázek 39: Navržené řešení první úlohy čtvrté kapitoly – výpis slova

Jelikož v předchozí úloze nejsou potřebné žádné senzory či další díly a úloha sama o sobě nerozšiřuje zkušenosti v robotice, další návrh je rozšiřujícím úkolem: „K řídicí jednotce připojte LED nárazník. Naprogramujte, aby se na obrazovku vypsalo slovo (např. z předchozího úkolu) poté, co stisknete LED nárazník. Zařídte, aby se po stisknutí, každé slovo vypsalo na nový řádek.“



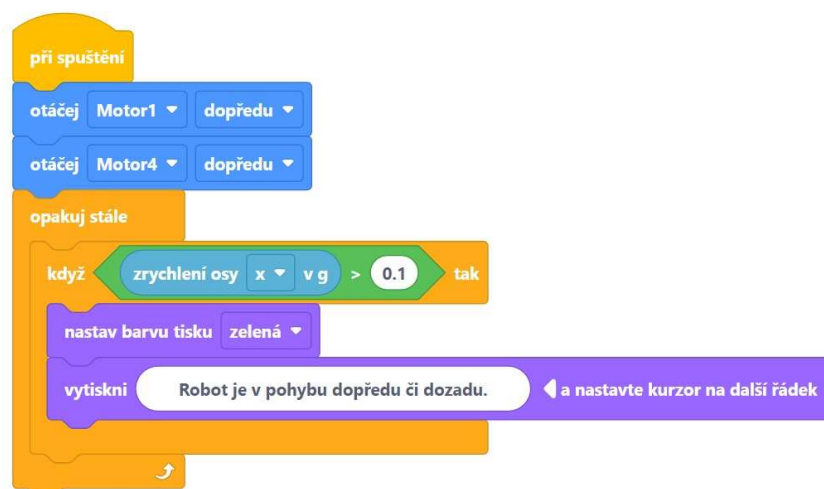
Obrázek 40: Rozšiřující úkol první úlohy čtvrté kapitoly – výpis slova

Druhá úloha čtvrté kapitoly je navržena k práci s pojezdným robotem z první kapitoly. Návrh slovního zadání: „Co se stane, pokud spustíte kód vyobrazený níže? Objevili jste nějakou chybu? Nejdříve o tom přemýšlejte a následně svoji domněnku otestujte.“



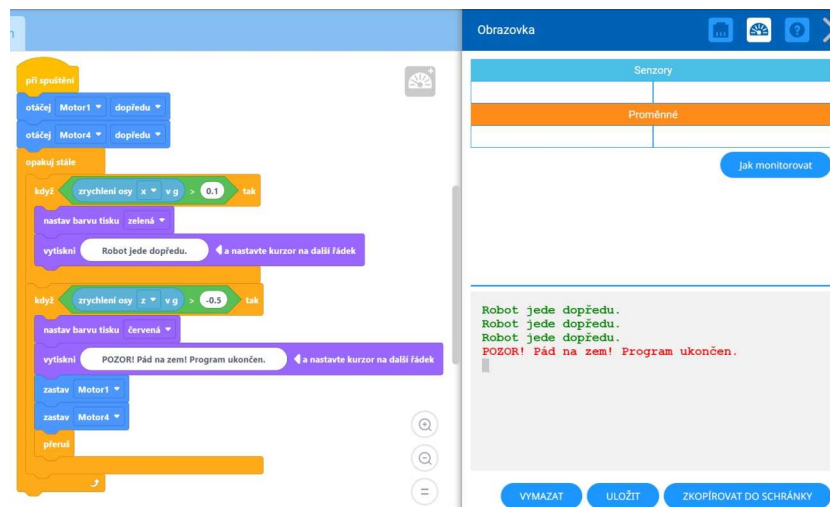
Obrázek 41: Chybný kód k druhé úloze čtvrté kapitoly – gravitační zrychlení

Návrh prvního dílčího úkolu k druhé úloze: „Kód vyobrazený výše obsahuje chybu. Opravte chybu v kódu a vyzkoušejte, zda program již správně funguje.“ Chyba v kódu se vyskytuje v podmínce, kdy se bere zrychlení osy „y“. To však zaznamenává pohyb vpravo či vlevo řídicí jednotky. Aby se vypsal text z fialového bloku, je nutné změnit zrychlení osy na písmeno „x“. Cílem úlohy je, aby žáci porozuměli kódu a přišli na to, že řídicí jednotka zaznamenává pohyb (gravitační zrychlení) třemi směry.



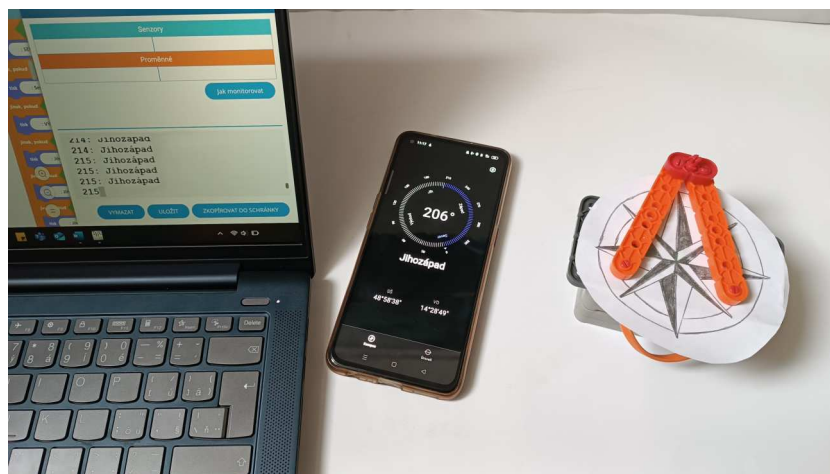
Obrázek 42: Správné řešení druhé úlohy čtvrté kapitoly – gravitační zrychlení

Druhý dílčí úkol vztahující se k druhé úloze: „Program upravte tak, aby při pohybu vpřed, vypisoval hlášku ‚Robot jede dopředu‘, a pokud gyroskop zaznamená prudký pohyb dolů (například pád ze stolu) vypíše se hláška ‚POZOR! Pád na zem! Program ukončen.‘ a celý program se přeruší, včetně zastavení motorů.“ V tomto případě je důležité upozornit, aby robot nebyl shozen na tvrdý povrch podlahy z velké výšky. Vhodné je před úkolem připravit prostředí či pomůcky, kdy se vedle stolu na zem položí molitan, silná deka nebo libovolný měkký povrch, aby se zabránilo poškození stavebnice.



Obrázek 43: Rozšiřující úkol druhé úlohy čtvrté kapitoly – gravitační zrychlení

Třetí alternativní úloha týkající se výpisu na displej, je naprogramování kompasu. Funkčnost naprogramovaného kompasu ve stavebnici VEX GO není založena na magnetické střelce, jako u reálného kompasu. Světové strany budou určovány dle úhlu otočení řídicí jednotky a nemusí tedy vždy odpovídat skutečným hodnotám.



Obrázek 44: VEX GO kompas vypisující světovou stranu na PC

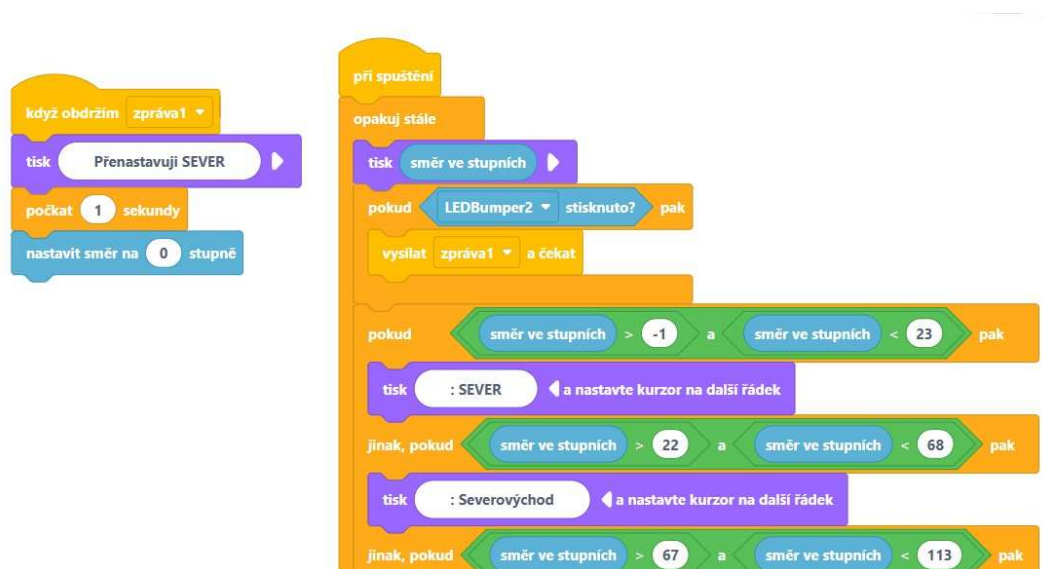
Základem úlohy je sestavit rozsáhlejší podmínku, která bude kontrolovat úhel otočení řídicí jednotky, podle které se budou na displej zařízení vypisovat světové strany. Hodnoty podmínek by si měli žáci zjistit sami experimentováním za pomoci buď reálného kompasu, buzoly nebo mohou využít chytrého mobilního zařízení, které též umožňuje funkci kompasu viz fotografie na předchozí straně (obrázek č. 44).



Obrázek 45: Řešení třetí úlohy čtvrté kapitoly – kompas

Po sestavení programu, který kontroluje v podmínkách úhel otočení u řídicí jednotky, je možné diskutovat o tom, zda skutečně tato pomůcka znázorňuje světové strany tak jako opravdový kompas. V případě, že tomu tak není, jak by bylo možné takový problém vyřešit.

Jedno z možných řešení je přidání tlačítka pro kalibraci světových stran. Do kódu byla přidána část pro tlačítko (LED nárazník), které po stisku přenastaví hodnoty směru řídicí jednotky a je tak možné změnit výchozí hodnotu, například pro sever. Pro synchronizaci lze využít opět reálný kompas nebo aplikaci. V takovém případě stačí seřadit jednu světovou stranu a pokud žáci nastaví správně rozsahy hodnot v podmínce, zbylé světové strany by měly být určovány správně.



Obrázek 46: Výňatek řešení dílčího úkolu třetí úlohy – kompas

5.4.3 Shrnutí kapitoly 4

Tato kapitola byla první, kterou nebylo možné předělat z původní učebnice do stavebnice VEX GO. Tento fakt byl předpokládán již od samotného začátku, jelikož po rozboru všech kapitol původní učebnice bylo zjištěno, že čtvrtá kapitola se zabývá zvukem a displejem. Těmito součástmi stavebnice VEX GO nedisponuje.

Původní kapitola byla podrobena analýze na jejímž základě byly pro stavebnici VEX navrženy alternativní úlohy. Pro práci se stavebnicí VEX GO byly zvoleny alternativy, jako jsou výpis na obrazovku (podobný výpisu na displej), a vzhledem k tomu, že zvuk nelze v tomto případě nijak nahradit, byla zvolena práce s gyroskopem a vnímáním gravitačního zrychlení řídicí jednotky. Je vhodné poznamenat, že ačkoli i stavebnice LEGO disponuje gyroskopem, funguje pouze jako externí senzor a nedokáže pracovat s gravitačním zrychlením. Struktura kapitoly byla přizpůsobena dle původní učebnice. Návrh úloh zahrnuje tvorbu vlastního kódu,

rozbor již vytvořeného kódu, úpravu kódu a doplňující otázky. V poslední úloze, jejímž cílem je napodobit realistickou funkčnost, je nutné mít k dispozici vhodné pomůcky, jako například kompas nebo aplikaci kompasu.

5.5 Kapitola 5: Mixér

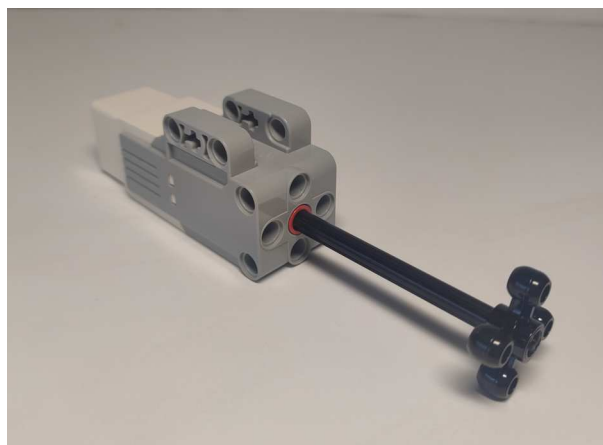
Pátá kapitola se, stejně jako kapitola předchozí, nezabývá pojízdným robotem. Na začátku kapitoly žáci postaví „mixér“, ke kterému jsou potřebné pouze řídicí jednotka, střední motor a drobné komponenty, aby finální stavba alespoň mírně připomínala vzhled tyčového mixéru.

Poté, co žáci splní konstrukční fázi, začínají programovat, aby uvedli mixér do pohybu. Úlohy začínají od základního rozpořbování až po mírně složitější úkoly, ve kterých žáci připojují tlačítko k mixéru. Kapitola se dostává až do fáze, kdy postavené a naprogramované zařízení připomíná miniaturu tyčového mixéru, včetně světelných a zvukových efektů.

5.5.1 Mixér – původní učebnice

Kapitola začíná úlohou 5.1, kde mají žáci za úkol postavit mixér podle vizuální ilustrace v původní učebnici. Po sestavení mají připojit motor k řídicí jednotce.

Úloha 5.2 „začínáme programovat mixér“, je rozdělena na dílčí úkoly. Úkol 5.2.1 je nejjednodušším úkolem, ve kterém mají žáci v programovacím prostředí objevit blok pro ovládání středního motoru. Tento blok je totožný s blokem pro velké motory a taktéž obsahuje tři parametry s možnostmi dalších nastavení pro pohyb.



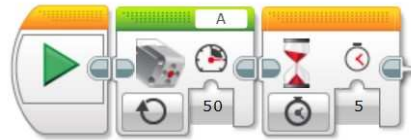
Obrázek 47: Sestavený mixér – LEGO Mindstorms

Záložka „5.2.2 Problém“ a „5.2.3 SOS“ se vztahují ke kódu z prvního úkolu 5.2.1. V těchto záložkách mají žáci odpovědět na otázky a zjistit, proč se mixér otočil velmi krátkou dobu: „Z jakého důvodu se mixér otočil jen velmi málo a poté se zastavil? Pokuste se na problém přijít.“



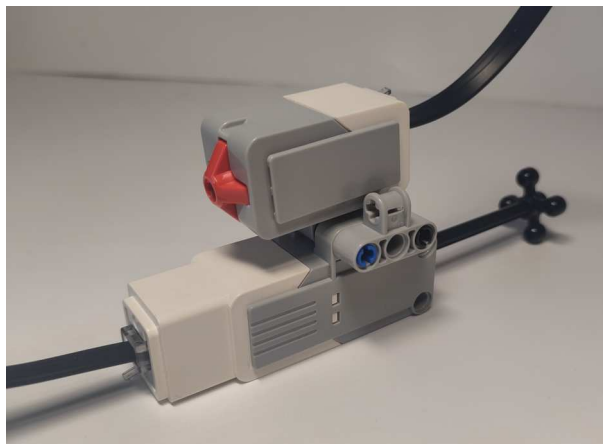
Obrázek 48: Kód k úkolu 5.2.1

Po zodpovězení otázky, proč se mixér spustil pouze na krátkou dobu (což je způsobeno tím, že v bloku je nastavena jen jedna otáčka), mají žáci v úkolu 5.2.4 upravit kód nebo vytvořit vlastní tak, aby mixér běžel po dobu pěti vteřin.



Obrázek 49: Řešení úkolu 5.2.4

Úloha 5.3: „Mixér, se kterým jsme nyní pracovali, je velmi nepraktický. Nedá se nijak ovládat ani zastavit.“ Jedná se o další konstrukční úlohu. Cílem je upevnit tlačítko k mixéru a následně jej připojit k řídicí jednotce. V následující úloze 5.4 se programuje mixér s tlačítkem, který je vyobrazen na fotografii níže (obrázek č. 50).



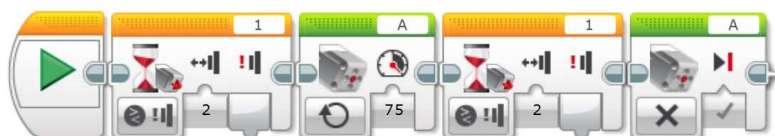
Obrázek 50: Sestavený mixér LEGO Mindstorms s přidaným tlačítkem

Pod záložkou 5.4.1 je tip pro žáky, jak pracovat s blokem „čkej“ i jaké možnosti tento blok nabízí. Následujícím úkolem 5.4.2 je upravit kód, kterým se žáci zabývali v předchozích úlohách. Cílem úkolu je zajistit, aby se mixér začal točit až po stisknutí tlačítka.



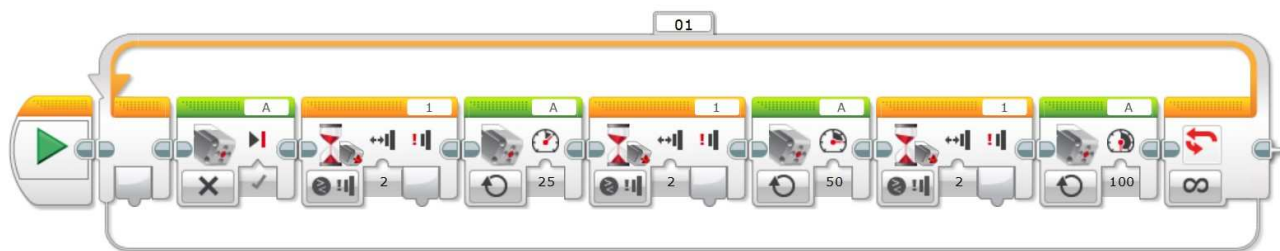
Obrázek 51: Řešení úkolu 5.4.2

Úkol 5.4.3 je rozšiřujícím úkolem k úkolu 5.4.2: „*Jak musíte program rozšířit, aby se mixér po stisknutí tlačítka spustil a po dalším stisknutí tlačítka znovu vypnul?*“



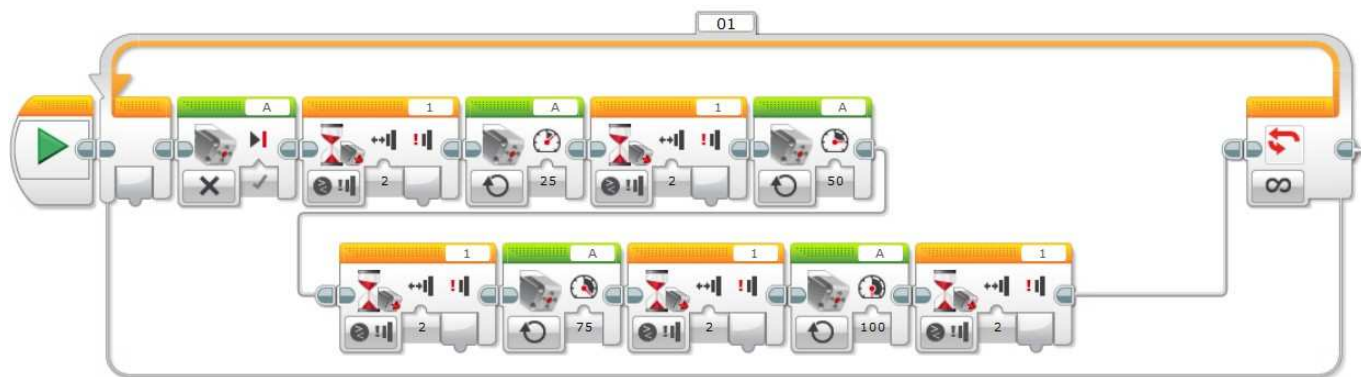
Obrázek 52: Řešení úkolu 5.4.3

V úloze 5.5 mají žáci již vytvořený kód od autorů učebnice, avšak se záměrně implementovanou chybou. Žáci zde mají vyjma odhalení chyby odpovědět na doplňující otázky. Mají se tak seznámit s blokem „opakuj“, který byl v původní učebnici k obeznámení zatím pouze jednou v kapitole tři.



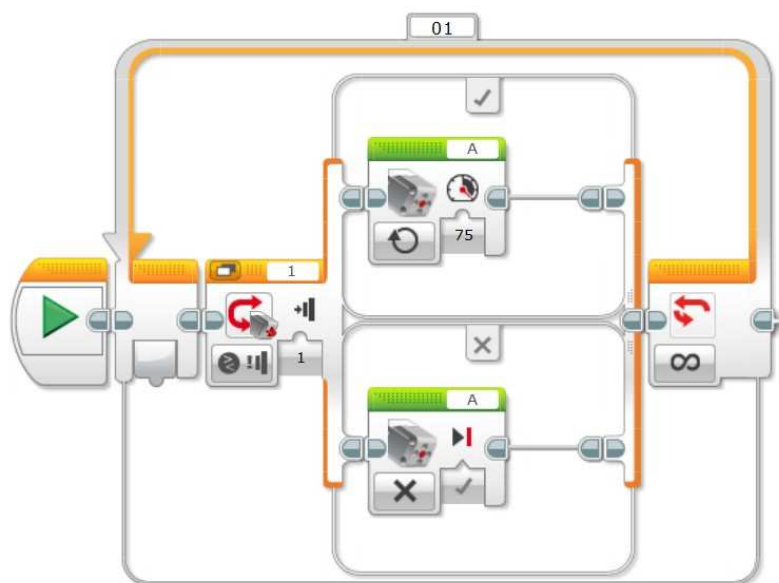
Obrázek 53: Kód k úkolu 5.5 obsahující chybu

Chyba v kódu se nachází na konci řádku, kde chybí jeden blok „čkej“. V takovém případě se nestihne vykonat rychlost na 100 %. Po odhalení a následném opravení této úlohy mají žáci navíc kód rozšířit, aby mixér měl přidanou rychlost 75 %, která v původním kódu chybí. Opravený a rozšířený kód je uveden na další straně (obrázek č. 54).



Obrázek 54: Opravený a rozšířený kód úkolu 5.5

Vyjma poslední úlohy pro rychlejší žáky, se kapitola uzavírá závěrečnou úlohou 5.6, která začíná kratší teoretickou částí o podmínce, kterou představuje blok „switch“ v původním softwaru. Úloha obsahuje úkol 5.6.2 pro modifikaci mixéru, ve kterém mají žáci naprogramovat, aby mixér fungoval pouze v případě, kdy uživatel drží stisknuté tlačítko, a po jeho uvolnění se mixér vypne.

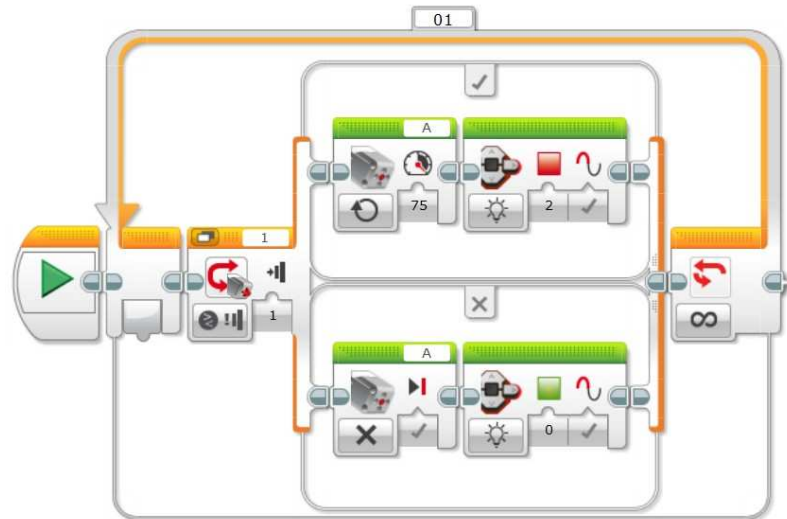


Obrázek 55: Kód k úkolu 5.6.2

Úkol 5.6.3 zadává modifikaci mixéru pro světelné efekty. Při stlačení spouštěcího tlačítka nebo obecně běhu mixéru mají tlačítka řídicí jednotky svítit červeně. Naopak, když mixér není v běhu, tlačítka mají svítit zeleně.

Úkol 5.6.4: „Náš mixér je velmi tichý. Vylepšete ho i o zvukovou signalizaci běhu.“ je téměř totožný jako kód předchozí úlohy, který je vyobrazen na další straně (obrázek č. 56). Vedle bloku pro podsvícení tlačítek by byl přidán blok pro

zvuky. Vzhledem k tomu, že tento kód nelze vytvořit ve stavebnici VEX, není kód s bloky pro zvuk vyobrazen. Kapitola je završena úlohou pro rychlé žáky, a je založena především na fantazii žáků, čímž se nechává volný tvůrčí prostor. Žáci zde mají upravit konstrukci mixéru.



Obrázek 56: Kód k úkolu 5.6.3

5.5.2 Mixér – stavebnice VEX

Na úvod kapitoly byl sestaven mixér z komponentů stavebnice VEX, která disponuje menší rozmanitostí součástí pro tuto úlohu. Zatímco LEGO obsahuje dva velké a jeden střední motor, VEX GO nabízí tři stejně velké a výkonné motory. Proto byl pro tuto úlohu zvolen jeden z těchto tří motorů. Úvodní úloha byla splněna bez komplikací, neboť mixér nevyžaduje žádné speciální komponenty.



Obrázek 57: Sestavený mixér VEX GO

Po sestavení následuje úloha 5.2. Prvotní program v úkolu 5.2.1 pro rozpočívání mixéru je velmi jednoduchý a je tedy možné jej sestavit i ve VEXcode GO. Program je však velmi krátký, obsahuje totiž pouze jednu otáčku motoru. Tato skutečnost je dána tím, že k tomuto kódu se v záložkách „5.2.2 Problém“ a „5.2.3 SOS“ vážou doplňující otázky. Zatímco v původním kódu učebnice byla nastavena rychlost na 50 %, u stavebnice VEX byla rychlost motoru nastavena na 70 %, aby byl výkon otáček zřetelnější a srovnatelný se stavebnicí LEGO.

V úkolu 5.2.4 mají žáci upravit předchozí kód tak, aby se místo jedné otáčky, mixér otáčel po dobu pět vteřin.



Obrázek 58: Řešení úkolu 5.2.1



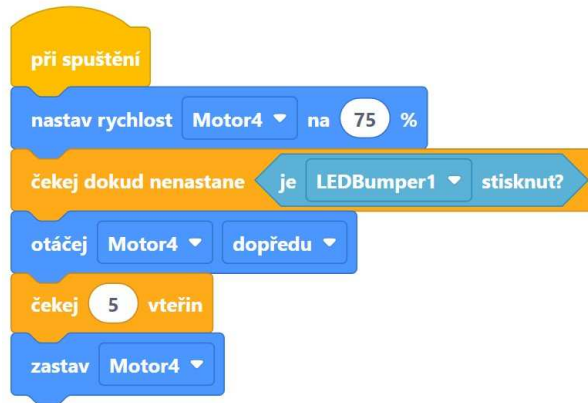
Obrázek 59: Řešení úkolu 5.2.4

Jelikož kódy úloh a úkolů doposud byly spuštěny pouze zahájením programu, je takový mixér velice neefektivní. Proto v úloze 5.3 mají žáci za úkol modifikovat mixér a konstrukčně k němu upevnit tlačítko. U stavebnice VEX GO byl pro tento případ využit LED nárazník.



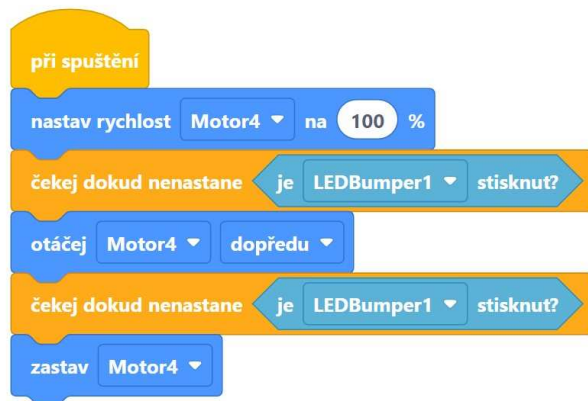
Obrázek 60: Sestavený mixér VEX GO s přidaným tlačítkem

Úloha 5.4 začíná úkolem 5.4.2 a v podstatě navazuje na předchozí úkol. Žáci mají naprogramovat mixér tak, aby se program uvedený v úkolu 5.2.4 spustil až poté, co uživatel stiskne tlačítko. Jako nevhodnější varianta pro tento kód se ukázalo použití bloku „čkej dokud nenastane“.



Obrázek 61: Řešení úkolu 5.4.2

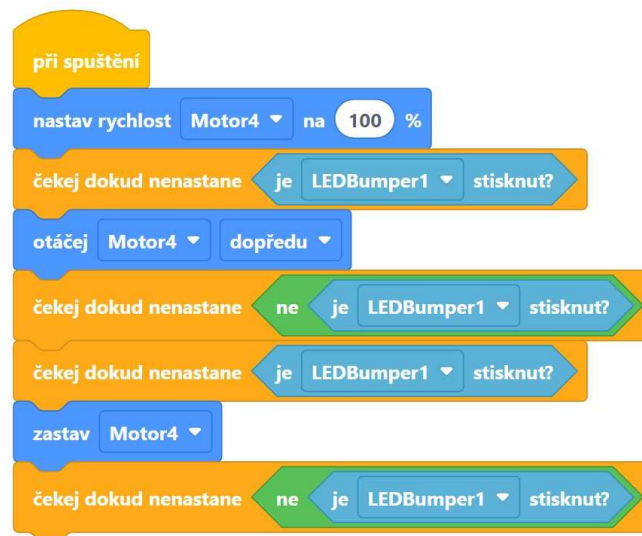
Úkol 5.4.3 je navazujícím úkolem a zároveň rozšiřující částí: „*Jak musíte program rozšířit, aby se mixér po stisknutí tlačítka spustil a po dalším stisknutí tlačítka znovu vypnul?*“ Níže je uveden kód, jak lze jednoduše program rozšířit záměnou jednoho bloku z předchozího programu.



Obrázek 62: Řešení úkolu 5.4.3

Ačkoli se výše zmíněný kód jeví jako logický a zcela funkční, je třeba zmínit důležitý aspekt jeho fungování. Pokud uživatel stiskne tlačítko poprvé, motor se začne otáčet, při druhém stisknutí se motor dle očekávání zastaví. Nicméně pokud by nastala situace, kdy uživatel bude od samotného začátku tlačítko držet,

podmínka v bloku „čkej dokud nenastane“ se splní v obou případech kódu. Což by způsobilo spuštění mixéru na velice krátkou dobu, následně okamžité zastavení a ukončení programu. Z tohoto důvodu je zde uveden i kód, který je vhodnější variantou a zabraňuje této skutečnosti. Do kódu bylo nutné přidat další bloky „čkej dokud nenastane“ s podmínkou, že tlačítko nesmí být stisknuté. Díky tomu bude LED nárazník skutečně fungovat jako tlačítko, které musí být stisknuto a následně uvolněno, aby kód pokračoval ve svém běhu. Teprve tak bude mixér fungovat jako v původní učebnici.

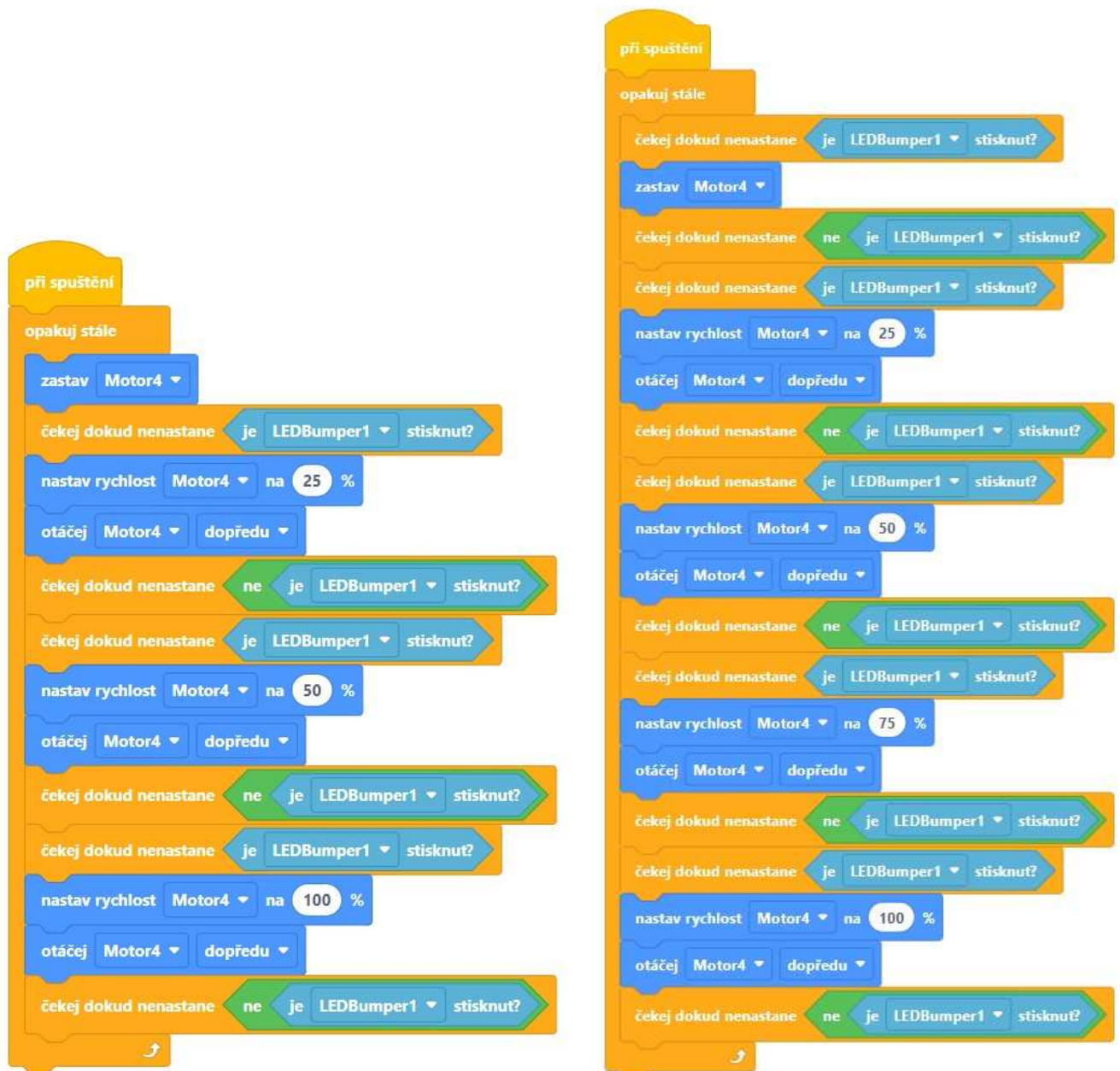


Obrázek 63: Upravené řešení úkolu 5.4.3

Kapitola pokračuje úlohou 5.5 s názvem „Dokážete opravit program?“ Zde mají žáci k dispozici kód od autorů, ve kterém se vyskytuje chyba. Cílem žáků je chybu nalézt, opravit ji a následně upravit kód tak, aby byla pro mixér přidaná rychlost 75 %. Zároveň zde žáci testují funkčnost bloku „opakuji“.

Ačkoli se podařilo sestavit úlohu i se záměrnou chybou, kterou autoři v kódu zanechali, kód je příliš zdlouhavý a opakující se. Pro zachování záměrné chyby v kódu se ve VEXcode GO nepodařilo vytvořit kratší verzi kódu.

Z důvodu rozsáhlosti kódu, je ilustrován samostatně na další stránce (obrázek č. 64).



Obrázek 64: Kódy k úloze 5.5 – nalevo kód se záměrnou chybou, napravo opravený kód úlohy s rozšířením o jeden stupeň rychlosti

Lepší a efektivnější variantou řešení tohoto úkolu je jeho zpracování pomocí proměnných. Toto řešení, které je vyobrazeno na další straně je přehlednější a stručnější než předchozí varianty. Avšak za cenu práce s proměnnými, které nemusí vyhovovat především žákům nižších ročníků druhého stupně. Nicméně v tomto řešení není zahrnuta potenciální chyba, kterou autoři učebnice zamýšleli. I přesto je vhodné jej uvést jako jednu z možných cest k optimalizaci původního, výrazně zdlouhavějšího kódu.



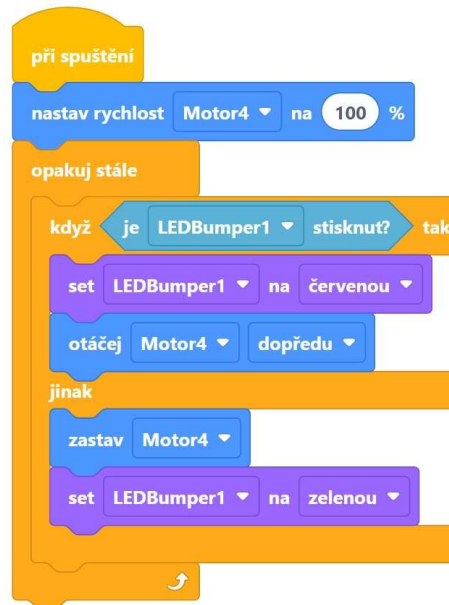
Obrázek 65: Řešení k úkolu 5.5 s proměnnou (bez záměrné chyby)

Závěrečná úloha 5.6 s úkolem 5.6.2: „Zařídte, aby mixér běžel jen v případě, že je tlačítko stisknuto. Po uvolnění tlačítka se vypne.“ Tento úkol byl ve VEXcode na sestrojení jednodušší než předchozí úkol, protože zde není potřeba řešit podmínky pro uvolnění tlačítka. V tomto případě lze úkol vyřešit pomocí bloku „když, jinak“ s neméně tak důležitým blokem „opakuj stále“ pro správné fungování programu.



Obrázek 66: Řešení úkolu 5.6.2

Úkol 5.6.3 modifikuje kód z předchozího úkolu. Modifikace se týká vizuálních efektů. Jedná se o lehkou úpravu, při níž během chodu mixéru svítí LED nárazník, který umožňuje podsvícení dvěma barvami. Pro splnění podmínek tohoto úkolu je potřebné do kódu, který je vyobrazený výše přidat pouze dva bloky z nabídky „vzhled“.



Obrázek 67: Řešení úkolu 5.6.3

Zatímco podsvícení tlačítka u VEX působí vizuálně atraktivněji než u LEGO, úkol 5.6.4 nelze ve stavebnici VEX GO splnit, protože není vybavena reproduktory.

5.5.3 Shrnutí kapitoly 5

Konstrukčně je možné mixér bez problémů sestavit i ve stavebnici VEX GO. Avšak u stavebnice VEX GO nepůsobí mixér na pohled tak efektivně jako u LEGO, z důvodu menší rychlosti otáček motorů. Při stejně nastavené rychlosti 50 % u obou stavebnic, působí VEX pomalým pohybem a nepředstavuje tak model realistického mixéru jako u LEGO. O podobném problému již bylo pojednáno ve shrnutí druhé kapitoly.

Kódy pro stavebnici VEX vypadají komplikovaně, protože je v nich obsažena podmínka pro uvolnění tlačítka. Slovy: „čkej, dokud není tlačítko uvolněno“ ve své podstatě přeměňuje LED nárazník na tlačítko. Jedním stiskem a následným uvolněním program zapne a druhým stisknutím program vypne či přepne do

dalšího kroku programu. Nefunguje tedy stejně jako v úkolu 5.6.2, kdy mixér běží pouze tehdy, když je tlačítko drženo.

Bohužel není možné vytvořit poslední úkol původní učebnice, který se týká zvukové modifikace mixéru. Tato stavebnice nejen že nenabízí možnost práce se zvukem, ale nenabízí ani žádnou alternativní náhradu. I přes tento mírný nedostatek lze téměř celou kapitolu převést do stavebnice VEX GO.

5.6 Kapitola 6: Závora

Tato kapitola začíná konstrukcí robotické závory, při které se mohou žáci inspirovat realistickými fotografiemi nebo ji postavit dle své fantazie a kreativity. Hlavním cílem kapitoly je sestavit a naprogramovat minimalistickou verzi závory, kterou žáci mohou vidat v reálném světě a rozumí jejímu praktickému využití. Se sestavenou závorou žáci pracují i v následující sedmé kapitole.

Ačkoli je podle metodických listů kapitola rozplánována na dvě vyučovací hodiny, oproti předchozím se jeví jako kratší. Kromě stavění a programování ji opět doprovázejí dílčí otázky, na které žáci v průběhu odpovídají. Několik dílčích otázek se nachází především na začátku kapitoly. Žáci mají za úkol formulovat své odpovědi na otázky týkající se především fungování závor na parkovištích či u parkovacích domů.

Kapitola šestá se zaměřuje především na propojení znalostí z předchozích kapitol do nové podoby robota. Zároveň by se taktéž měli žáci v této kapitole naučit rozdělit složitější celek na dílčí problémy a řešit je postupně.

5.6.1 Závora – původní učebnice

Kapitola začíná obecným textem o závoře – jak může jednoduše fungovat, k čemu slouží a kde je možné se s ní setkat.

První část, označená jako úloha 6.1, je fotogalerie obsahující několik realistických snímků závor. Slouží jako vizuální doprovod k úvodnímu textu.

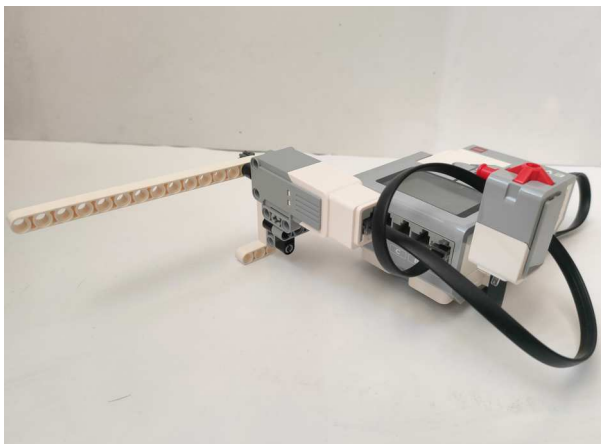
Úloha 6.2 „jak závory fungují“ obsahuje tři otázky vztahující se obecně k problematice závor: „*Jak závory fungují?*“, „*Jak mohou být závory ovládány?*“, „*Které problémy a chyby mohou při provozu závory nastat?*“ Je důležité o těchto aspektech komunikovat, aby žáci pochopili souvislosti mezi skutečnou závorou a závorou

jimi vytvořenou ze stavebnice. Jde o provázání senzorů a mechanismů v reálném světě s těmi na pracovním stole. Zároveň se jedná o obecné seznámení s touto problematikou, kterou se zabývají dvě kapitoly původní učebnice.

V úloze 6.3 je cílem konstrukčně sestavit jednoduchou závoru, kterou žáci následně budou programovat a modifikovat v dílčích úkolech. Na začátku úlohy mají žáci odpovědět na základní otázky týkající se procesu tvorby vlastní závory: „Co by námi vytvářená závara měla umět“, „Ve kterých stavech se může nacházet? Co je ovlivňuje?“, „Jak bude závara ovládána?“ Jmenované otázky slouží žákům jako prvotní plány, aby si uvědomili, jak bude jejich závara vypadat, fungovat a jak ji budou ovládat.

Úkol 6.3.1 je přímým zadáním pro tvorbu vlastní závory s případným tipem pro stavbu. Úkol 6.3.2 je nápovědou, jak by si měli žáci rozvrhnout práci a jak mají při práci postupovat. Hlavní myšlenkou této záložky je, aby si žáci uvědomili, že složitý celek mohou rozvrhnout na menší části.

Úkol 6.3.3 se týká zvoleného motoru v konstrukci vytvořené závory. Jedná se opět o malý návod či radu, jaký blok v SW využít pro ovládání středního motoru, a upozornění na to, že bloky mají více režimů ovládání. Jak v původní učebnici, tak v rámci praktické části byl pro závoru zvolen střední motor.



Obrázek 68: Sestavená závara z LEGO Mindstorms – již s přidaným tlačítkem pro následující úkoly

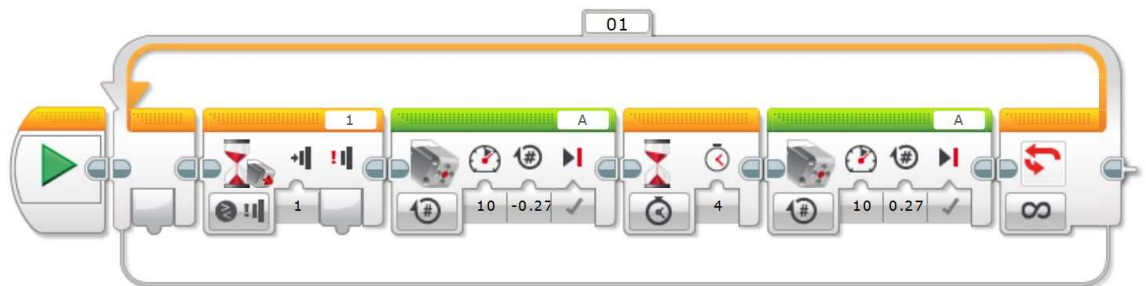
Úkol 6.3.4 vede žáky k tomu, aby se zamysleli nad ovládáním postavené závory. Tento úkol je ze všech jmenovaných této kapitoly takřka jediným, ve kterém žáci přímo vytvářejí kód. Mimo jiné je součástí zadání úkolu, přidání tlačítka k závoře,

kterým bude závora ovládána. Cílem tohoto úkolu je vytvoření kódu, který zajistí, že po stisknutí se závora otevře (respektive motor vykoná pohyb o určitý úhel), vyčká čtyři vteřiny a zavře se.



Obrázek 69: Řešení úkolu 6.3.4

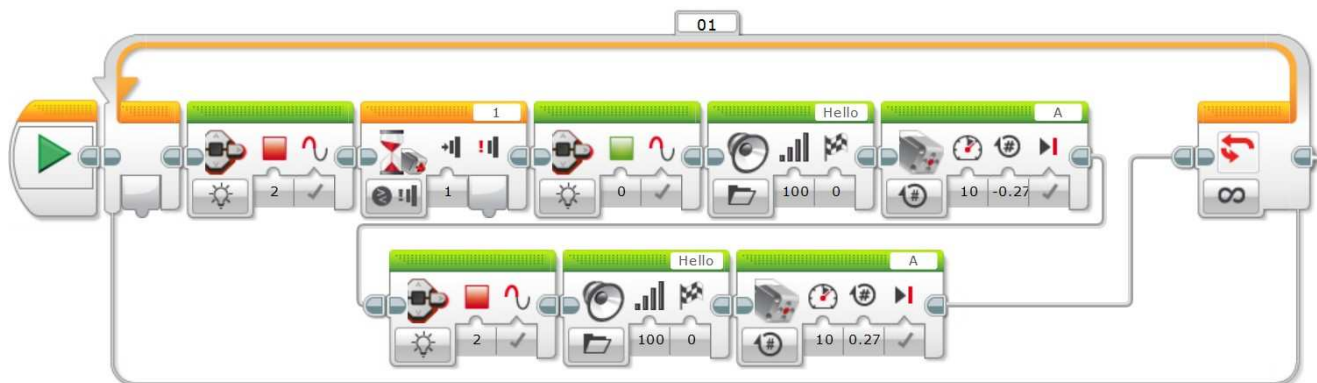
Jelikož by bylo nutné spouštět program neustále dokola po každém kliknutí na tlačítko, úkol obsahuje rozšiřující zadání. Kód má být upraven tak, aby se nemusel program neustále spouštět, ale aby bylo možné jej spustit jednou a tlačítko fungovalo na vícero stisknutí v rámci běhu jednoho programu.



Obrázek 70: Úprava řešení úkolu 6.3.4

V závěru této úlohy je úkol 6.3.5, který obsahuje otázky, jež mají opět přimět žáky zamyslet se nad realistickou funkčností závor, především, kdy a jak se závora zavírá. Žádoucí je těmito otázkami přimět žáky, aby se zamysleli nad využitím senzoru.

Úloha 6.4 je klasifikována jako úloha pro rychlejší žáky. Je rozdělena na dva dílčí úkoly, přičemž úkol 6.4.1 modifikuje závora o zvukovou signalizaci a úkol 6.4.2 doplňuje závora o světelnou vizualizaci. Jedná se o drobnou úpravu kódu z úkolu 6.3.4, jež je vyobrazený nad tímto textem.



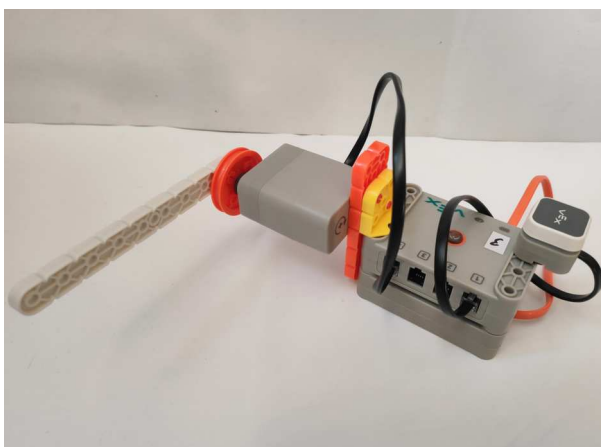
Obrázek 71: Řešení úlohy 6.4

5.6.2 Závora – stavebnice VEX

Jak bylo zmíněno na začátku, kapitola začíná dvěma teoretickými odstavci, ve kterých je úvodní text pro uvedení dané problematiky. Text popisuje, jak může závora jednoduše fungovat a kde se s ní můžeme běžně setkat.

V úloze 6.1 původní učebnice jsou fotografie zobrazující realistické závory. Úloha 6.2 obsahuje otázky týkající se funkčnosti závor v reálném světě. Tyto otázky jsou uvedeny v textu o závoře z původní učebnice. Není proto potřebné tyto části znovu dokládat.

Po zformulování odpovědí na otázky v úloze 6.3, jejichž cílem je dovést žáky k prvotním plánům pro sestavení závory, následují úkoly 6.3.1 a 6.3.2. Tyto úkoly vyzývají žáky ke konstrukci vlastní závory a poskytují drobné rady k úspěšnému sestavení. Přestože LEGO nabízí více dílků a možností pro kreativní vytvoření závory, taktéž je možné sestavit závora i ze stavebnice VEX.



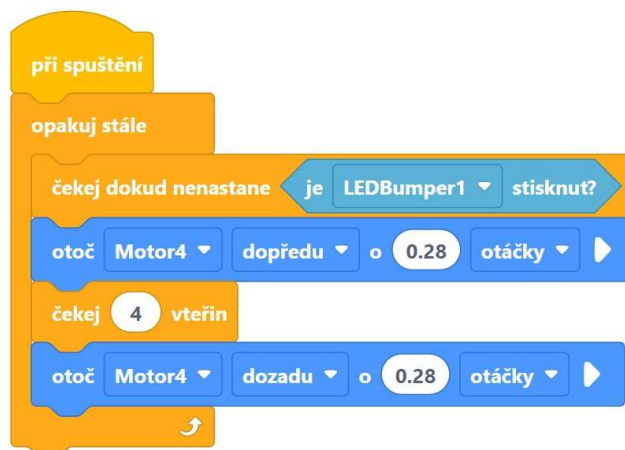
Obrázek 72: Sestavená závora ve VEX GO – již s přidáním tlačítkem

Úkol 6.3.4: „K závoře přidejte tlačítko, které ji otevře. Aby mohlo auto projet, závora musí zůstat 4 sekundy otevřená. Poté se sama zavře.“ Závora s tlačítkem je již vyobrazena na obrázku č. 72. Úkol se kromě konstrukčního kroku zabývá programováním závory. Tento úkol se podařilo bez obtíží vytvořit s použitím bloku „čkej dokud nenastane“. Pro otevírání a zavírání závory bylo zvoleno 0,28 otáčky, což je téměř shodná hodnota jako u stavebnice LEGO.



Obrázek 73: Řešení úkolu 6.3.4

Pro lepší a praktičtější funkčnost obsahuje úkol druhou rozšiřující část. V té je nutné zajistit, aby se nemusel program neustále spouštět dokola a přesto se závora zvedla po každém stisknutí tlačítka. V tomto případě stačí do kódu přidat pouze blok „opakuji stále“, čímž se program bude opakovat a nepřeruší se tak podmínka pro tlačítko.



Obrázek 74: Úprava úkolu 6.3.4

Šestá kapitola se uzavírá úlohou 6.4 pro rychlejší žáky, která obsahuje dva dílčí úkoly 6.4.1 a 6.4.2. Ty zadávají úpravu kódu z předchozí úlohy tak, aby byla závora modifikována jak o zvukovou, tak o světelnou signalizaci. Jak bylo již

v této práci několikrát zmíněno, stavebnice VEX GO nedisponuje reproduktory pro přehrávání zvuku. Z těchto důvodů není možné úkol 6.4.1 realizovat. Úkol 6.4.2, přidání světelného doprovodu lze bez komplikací splnit.



Obrázek 75: Řešení úkolu 6.4.2

5.6.3 Shrnutí kapitoly 6

Vyjma jednoho dílčího úkolu z celé kapitoly bylo možné vše naprogramovat a realizovat i u VEX GO. Také bylo možné konstrukčně sestavit závoru, přestože se na začátku objevily drobné komplikace, jak vhodně připojit závoru k motoru, aby držela ve vodorovném stavu.

Stejně tak i v této kapitole se již opakovaně objevila problematika s reproduktory, kterými VEX bohužel nedisponuje. Nebyla nalezena vhodná alternativa, jež by nahradila zvuk.

5.7 Kapitola 7: Automatická závora

V předchozí kapitole, za pomoci doplňujících otázek, by si měli žáci uvědomit, že spouštění závory skrze tlačítko na pevně stanovenou dobu je neefektivní a závora je možné zautomatizovat. Na konci předešlé kapitoly byli žáci již otázkami navedeni na myšlenku využití senzorů, čímž se zabývá sedmá kapitola, která má právě za cíl zautomatizovat závora pomocí ultrazvukového senzoru.

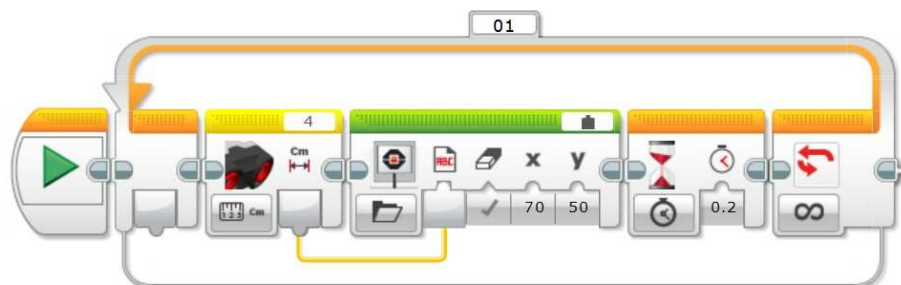
Žáci se zde naučí pracovat jak se senzorem vzdálenosti, tak se senzorem barev. Dále svou závoru, kterou sestavili v minulé kapitole modifikují. Kromě automatického zavírání a otevírání žáci naprogramují závoru, která bude rozpoznávat barvy čímž se rozliší typ vozidla, který bude skrze závoru projíždět.

Jako téměř veškeré kapitoly původní učebnice, je i tato podle metodických listů doporučena na dvě vyučovací hodiny. Žáci by se měli v této kapitole mimo jiné naučit uvažovat nad situacemi, které by mohly nastat při chodu závory, a navrhnout vhodná řešení.

5.7.1 Automatická závora – původní učebnice

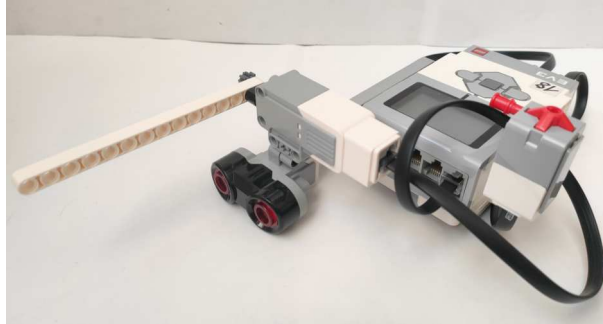
„Co by se ale stalo, kdyby auto muselo pod závorou náhle zastavit? Neměl by na to její tvůrce myslet?“ Toto jsou úvodní slova sedmé kapitoly původní učebnice, které slouží k uvedení žáků do problematiky daného tématu. Na začátku se v úloze 7.1 nachází dva odstavce textu, které teoreticky popisují ultrazvukový senzor stavebnice LEGO.

Úloha 7.2 je pro žáky seznamovací úlohou s novým senzorem. Po připojení ultrazvukového senzoru k řídicí jednotce je možné přejít k záložce 7.2.1. Ta obsahuje otázku, na kterých místech se nachází blok pro ovládání senzoru v programovacím prostředí. Záložka 7.2.2 poskytuje řešení s dalšími informacemi na předchozí otázku. Úkol 7.2.3 obsahuje již vytvořený kód, který mají žáci otestovat v programovacím prostředí a pochopit, jak senzor vzdálenosti funguje. Kód vypisuje hodnoty (vzdálenost) z ultrazvukového senzoru a zobrazuje je na displej řídicí jednotky. Záložka „7.2.4 SOS“ je pomocnou nápovědou pro pochopení níže uvedeného kódu.



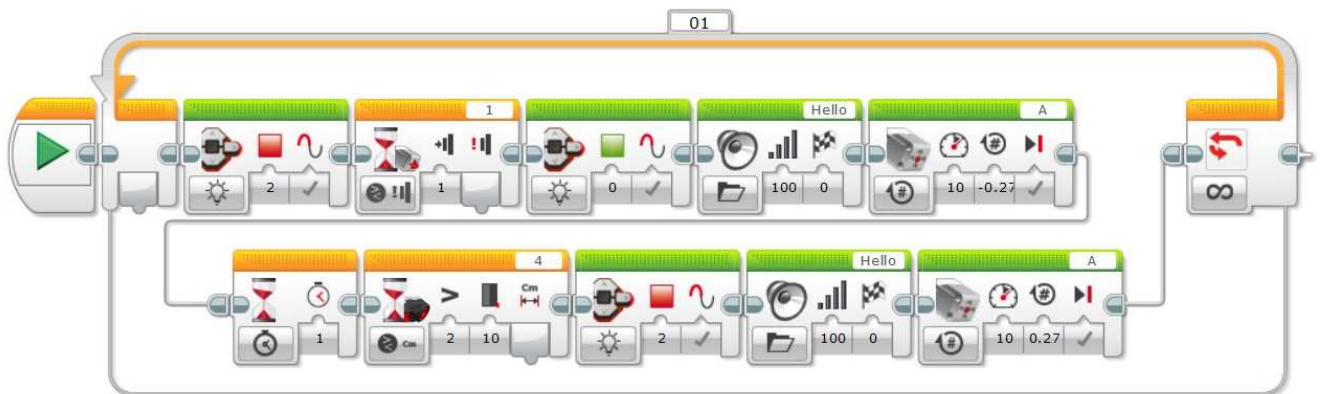
Obrázek 76: Kód k úkolu 7.2.3

Úloha 7.3 pobízí žáky k diskusi ve skupinách nad otázkou, jaké je nejvhodnější místo pro umístění senzoru. Po konstrukčním zakomponování ultrazvukového senzoru do robotické závory následuje úkol, jehož cílem je otestovat nejvhodnější možnou pozici pro umístění senzoru.



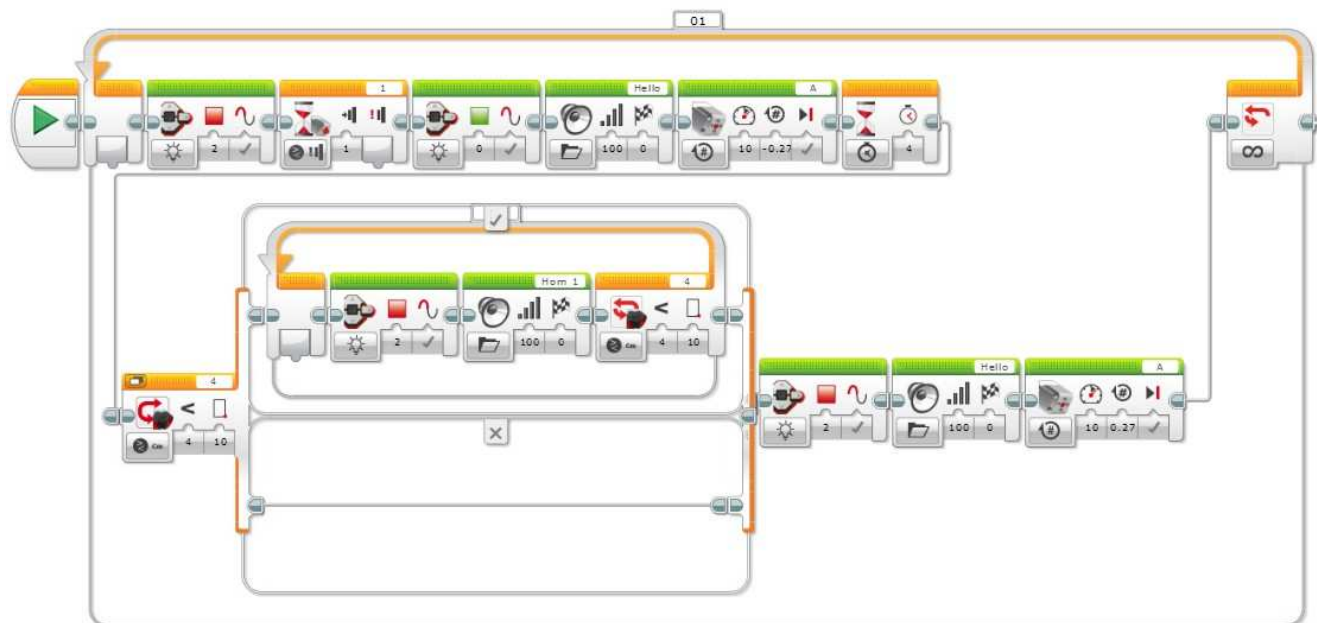
Obrázek 77: Sestavená závora LEGO Mindstorms s umístěným očním senzorem

V předchozí kapitole byla naprogramována závora, která se otevírala pomocí tlačítka a zavírala po pevně stanovené době. Úloha 7.4 tuto závoru z předchozí kapitoly upravuje. Úloha začíná úkolem 7.4.1, který zní: „Rozšířte program o bezpečnostní funkci, která nedovolí sklopení závory v době, kdy se pod ní nachází automobil.“ Stačí tak do kódu z posledního úkolu šesté kapitoly přidat jeden blok „čkej, dokud není vzdálenost větší než“ pracující se senzorem. Docílí se toho, že se závora nesklopí dolů, pokud se před senzorem nachází objekt.



Obrázek 78: Řešení úlohy 7.4

Úloha 7.5 je určena rychlejšími žákům. Tato úloha je pouze modifikací kódu nad tímto textem, ve kterém mají rychlejší žáci za úkol naprogramovat, že když vozidlo neopustí prostor závory před senzorem do čtyř vteřin, ozve se zvukové a světelné znamení.



Obrázek 79: Řešení úlohy 7.5

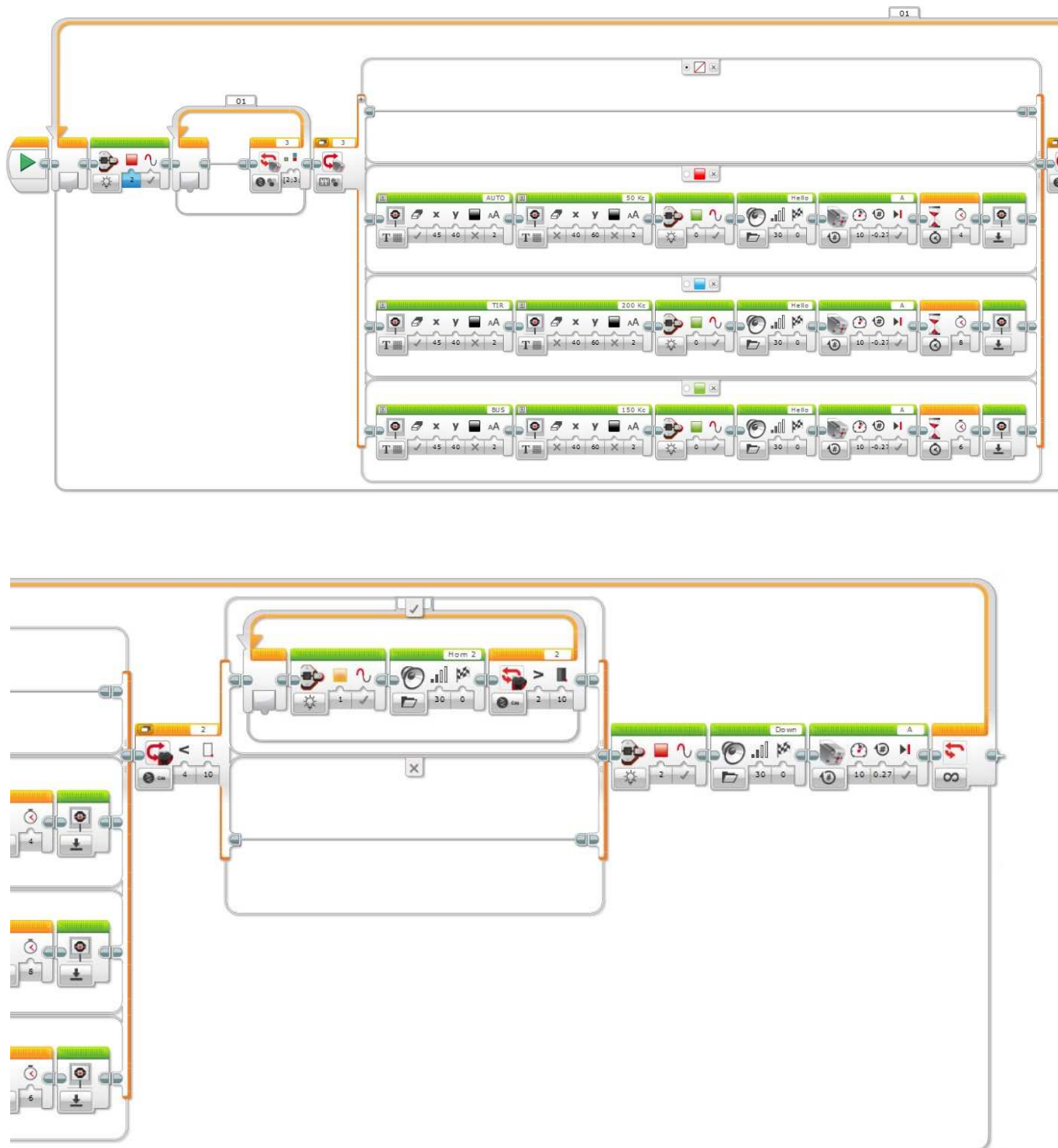
Úloha 7.6 „mýtná brána“ ke konstrukci závory připojuje barevný senzor. Ten slouží k rozpoznání barev, které budou sloužit jako identifikátory pro typ vozidla projíždějícího závorou. Úloha 7.7 obsahuje text popisující barevný senzor.

V úloze 7.8 „testujeme barevný senzor“ žáci zkoumají funkci barevného senzoru a jeho výstupy. Žáci mají za úkol do políček v učebnici doplnit hodnoty k určitým barvám, které senzor u barev oznamuje. Taktéž je zde tip, kde lze nalézt blok pro barevný senzor v programovacím prostředí.

Rozsáhlejší úloha 7.9 uzavírá dvojitou kapitolu zabývající se robotickou závorou. Je rozdělena do tří dílčích úkolů, z nichž jeden je opět určen pro rychlejší žáky. V úkolu 7.9.1 jsou uvedeny informace k jednotlivým barvám (červená = osobní automobil, zelená = autobus, modrá = nákladní automobil) a vyzývá k samotnému konstrukčnímu upevnění barevného senzoru. Kromě toho mají žáci naprogramovat, aby se závora automaticky otevírala po přiložení barevné karty. Z důvodu délky kódu je ukázka na další straně rozdělena do dvou částí.

Druhý rozšiřující úkol 7.9.2 zadává úpravu předchozího úkolu, který na displej řídicí jednotky vypíše, dle barev, jaké vozidlo skrze závoru projíždí s doplněním číselného poplatku za průjezd. Řešení je téměř totožné s kódem předchozího úkolu, proto je na další straně pouze výňatek vybrané části.

Poslední úkol pro rychlejší žáky, 7.9.3, zadává úkol, ve kterém musí být počítáno s tím, že závorou budou projíždět nákladní automobily, ty jsou však delší a potřebují více času, než se závora uzavře. Jedno z možných řešení této úlohy je vyobrazeno ve dvou částech pod tímto textem.



Obrázek 82: Řešení úkolu 7.9.1

5.7.2 Automatická závora – stavebnice VEX

V původní učebnici kapitola začíná krátkým popisem ultrazvukového senzoru. Stavebnice VEX GO však obsahuje odlišný senzor s jinými funkcemi a možnostmi než senzor stavebnice LEGO. Oční senzor stavebnice VEX, byl podrobněji popsán v teoretické části práce v kapitole 2.2.1 Funkce a komponenty.

Úloha 7.2 obsahuje dílčí úkoly, které žáky seznamují se základními operacemi při práci se senzorem. Znění úkolu 7.2.1 po drobné formulační úpravě: „Kde lze v programovacím prostředí najít bloky pro práci s očním senzorem?“ Odpověď na tuto otázku se nachází na vedlejší záložce pod číslem 7.2.2. V prostředí VEXcode GO lze programovací bloky pro oční senzor nalézt na levém bočním panelu v záložce „vnímání“.

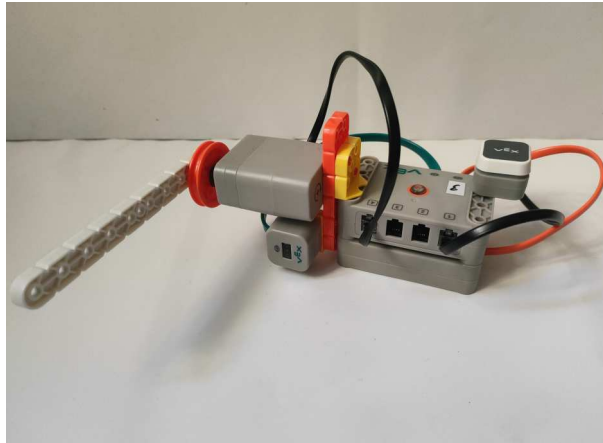
Úkol 7.2.3 v původní učebnici obsahuje hotový kód, ten mají žáci otestovat a zjistit jaké hodnoty se vypisují. Tento kód však nelze ve VEX GO přesně replikovat, protože oční senzor neumí měřit vzdálenost, jako tomu je u stavebnice LEGO. Oční senzor VEX pouze detekuje přítomnost objektu před sebou. Další drobnou komplikací tohoto úkolu je absence displeje. Vhodnou alternativou je vypisování hodnot jasu v procentech, vnímaných očním senzorem. Tyto hodnoty se zobrazují na obrazovce zařízení, ze kterého je robot programován a ke kterému je připojen. Ačkoli se kód může jevit jednoduchý, plní podobnou funkci jako kód z původní učebnice s tím rozdílem, že na místo vzdálenosti vypisuje hodnoty jasu. Tento úkol lze využít u stavebnice VEX jako přípravu k řešení jedné z úloh v osmé kapitole.



Obrázek 83: Řešení navrženého úkolu 7.2.3

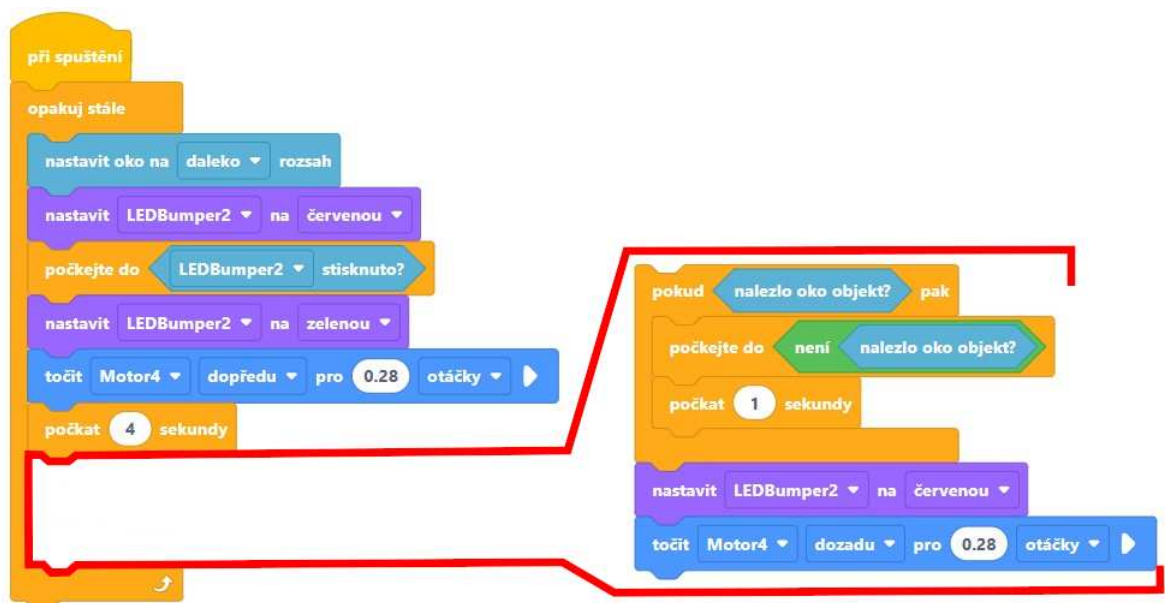
Úloha 7.3: „Na které místo je nejlepší umístit senzor, aby plnil svou funkci a brána fungovala bezproblémově?“ Konstrukčně byl tento úkol řešen téměř totožně jako v původní stavebnici. VEX sice nabízí méně rozmanité dílky, nakonec však

taktéž umožňuje umístění senzoru pod úroveň závory. Ačkoli umístění senzoru u této stavebnice může být obecně komplikované, především kvůli jeho kombinaci se senzorem barev (více ve shrnutí této kapitoly).



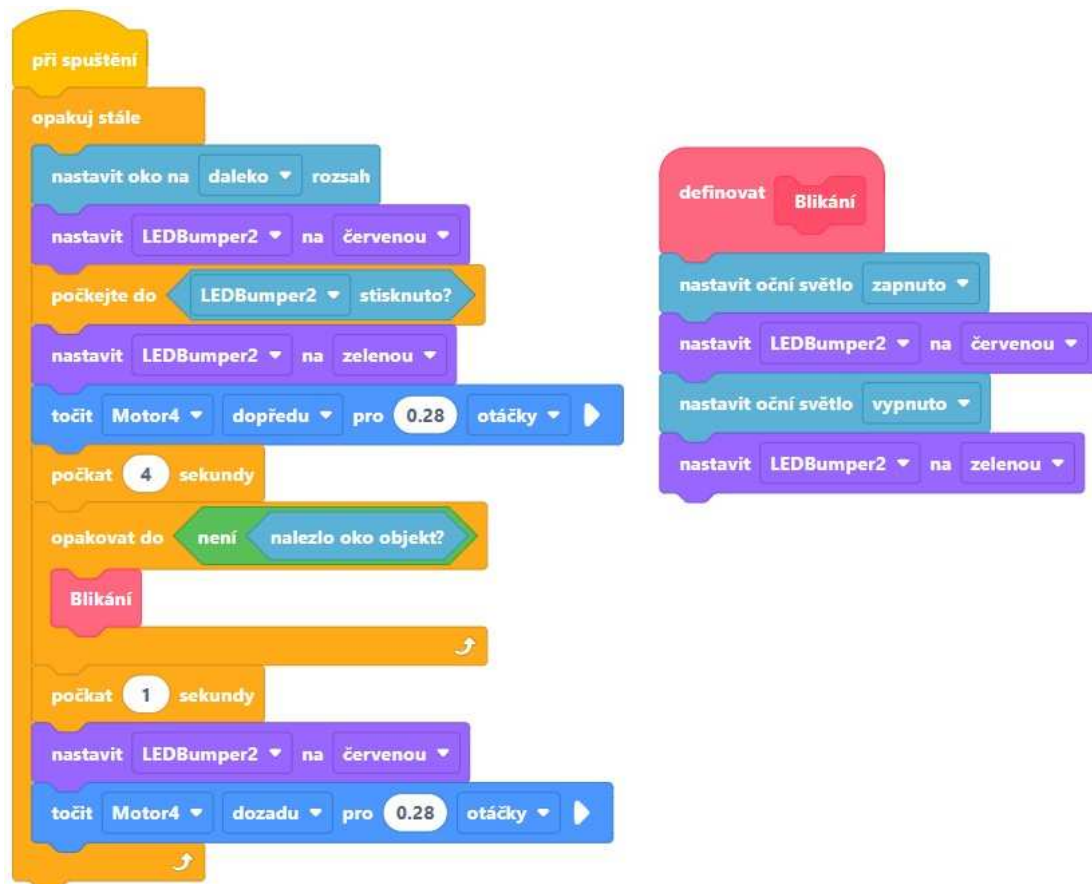
Obrázek 84: Sestavená závora VEX GO s umístěným očním senzorem

V úloze 7.4 je cílem vytvořit kód, který zajistí, že se závora nesklopí do doby, dokud se pod ní, respektive před senzorem, bude nacházet jakýkoliv objekt. Jde tak o realistické zpracování automatické závory, se kterou se žáci mohou setkat v běžném životě. Do části kódu s podmínkou pro oční senzor bylo nutné přidat blok „čkej jednu vteřinu“. V nepřítomnosti tohoto bloku se závora sklopila ihned poté, co senzor nezaznamenal žádný objekt. Při testování ke sklopení docházelo velmi rychle, a proto bylo nutné přidat jmenovaný blok pro plynulejší chod závory.



Obrázek 85: Řešení úlohy 7.4

Úloha 7.5 určená pro rychlejší žáky, zadává modifikaci kódu vyobrazeného na předchozí straně. Cílem je přidání podmínky, která kontroluje, zdali se předmět nenachází před senzorem příliš dlouhou dobu. Jinými slovy, po uplynutí čtyř vteřin řídicí jednotka začne vydávat zvukové a světelné signály. Z důvodu nepřítomnosti reproduktoru u stavebnice VEX byla přidána pouze světelná signalizace. Světelným upozorněním je blikání osvětlení senzoru a LED nárazníku. Pro realizaci kódu byl vytvořen vlastní blok, který spouští samotné blikání senzoru a LED nárazníku.

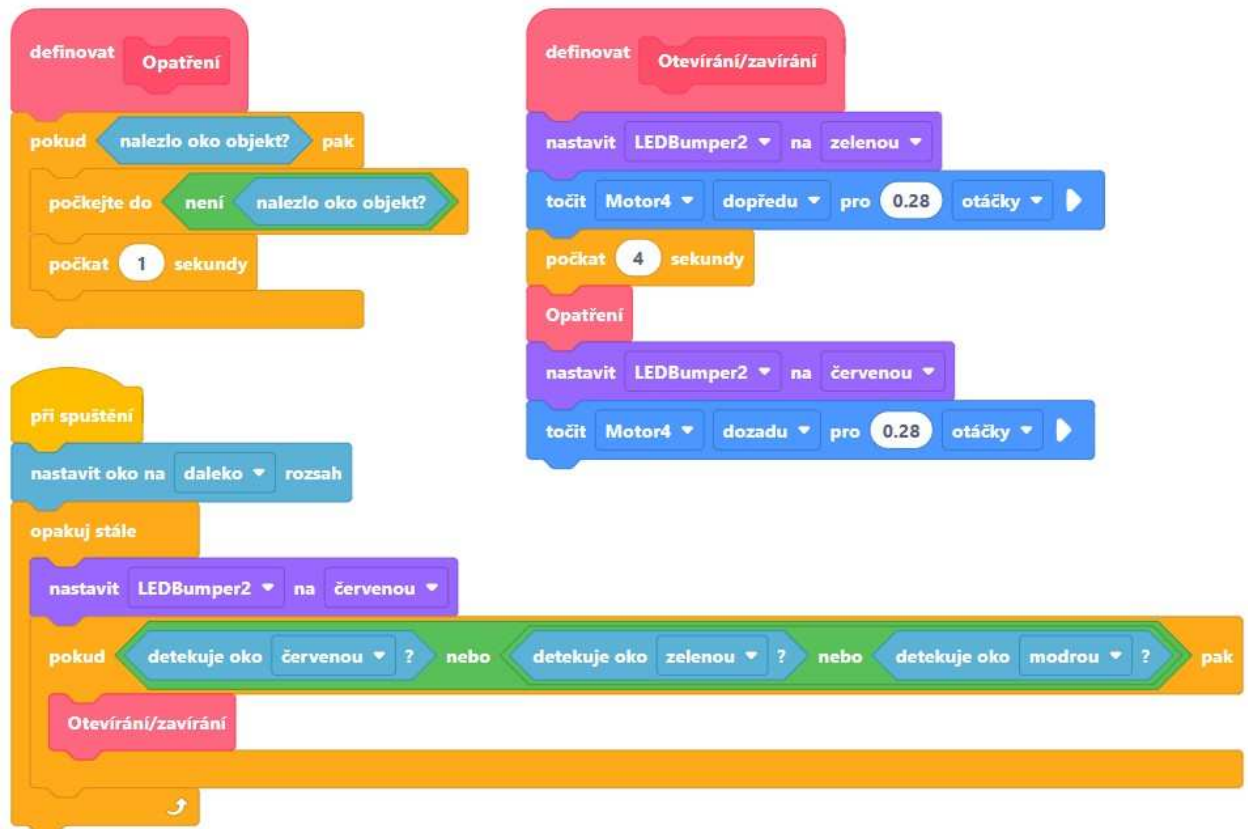


Obrázek 86: Rozšíření kódu z úlohy 7.4 – řešení úlohy 7.5

Úloha 7.6 je informační částí a počátkem práce věnované barevnému senzoru. Ve stavebnici VEX je barevný senzor integrován společně v kombinaci s očním senzorem (oproti stavebnici LEGO, kde je barevný senzor samostatně). Úloha 7.7 obsahuje pouze teoretické informace o senzoru.

V úloze 7.8 žáci testují barevný senzor, prozatím nepřípojený k závoře. U stavebnice VEX GO oční/barevný senzor v základu rozeznává pouze tři základní barvy (červená, zelená a modrá). Závěrečná úloha 7.9 „mýtná brána“, je roz-

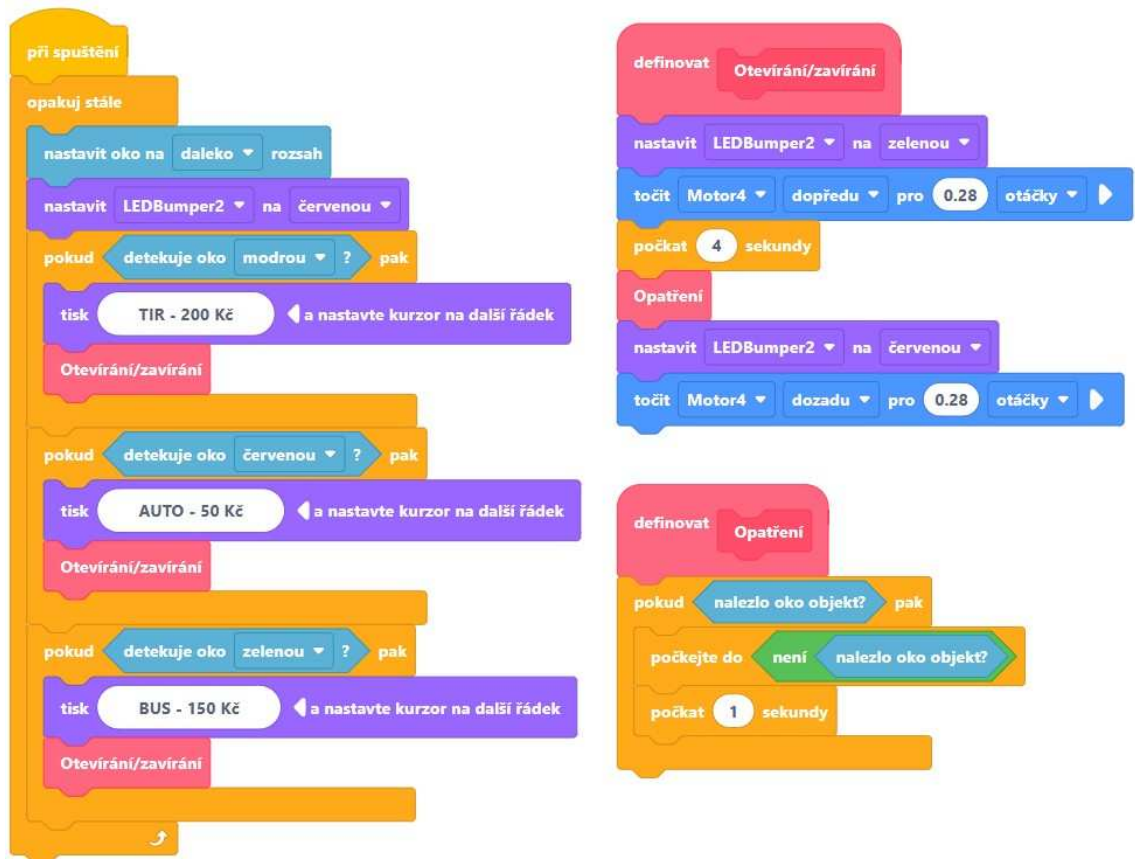
dělena do tří úkolů, které na sebe plynule navazují. V úkolu 7.9.1 je cílem plně zautomatizovat závoru, která od tohoto úkolu nebude ovládána tlačítkem, nýbrž automaticky se bude otevírat za pomoci očního senzoru. V úkolu je potřeba vytvořit podmínku, která bude závoru otvírat, jakmile bude před senzor umístěna barevná karta (červená, zelená nebo modrá). K realizaci byly využity dva vlastní bloky pro lepší přehlednost a zkrácení kódu.



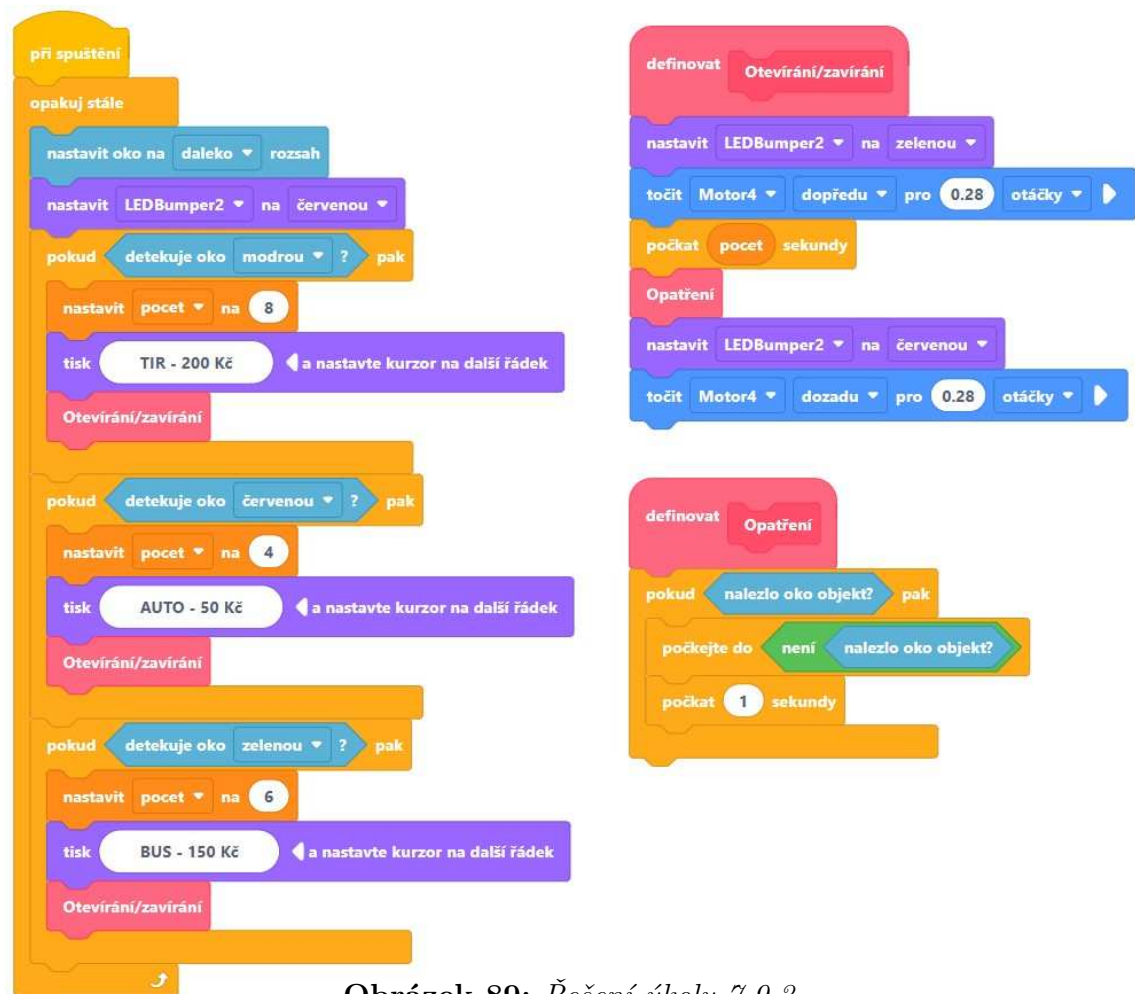
Obrázek 87: Řešení úkolu 7.9.1

Úkol 7.9.2 (obrázek č. 88) rozšiřuje kód z úkolu 7.9.1 tak, aby po předložení barevné karty před senzor byla vypsána částka a typ vozidla. Úprava se týká hlavního spouštěcího kódu, zatímco dva vlastní bloky zůstaly zachovány. Výpis je opět realizován na obrazovce zařízení, na kterém uživatel pracuje.

Poslední úkol 7.9.3 (obrázek č. 89) pro rychlejší žáky má efektivnější a přehlednější řešení u stavebnice VEX, kde se mnohem lépe pracovalo s proměnnými. Tato skutečnost kód zkracuje a činí jej přehlednějším oproti původnímu softwaru. Bylo tak docíleno toho, že dobu sklopení závory nastavuje proměnná dle předložené karty pro daný typ vozidla.



Obrázek 88: Řešení úkolu 7.9.2



Obrázek 89: Řešení úkolu 7.9.3

5.7.3 Shrnutí kapitoly 7

Po spuštění kódu z programu 7.9.1 u stavebnice VEX se závora začala sama opakovaně otevírat a zavírat. Po několika pokusech, kdy závora stála na bílém papíře a doplňkové světlo senzoru svítilo, senzor vyhodnocoval, že vidí jednu ze tří barev. Jakmile byla závora přesunuta na dřevěný stůl, závora začala správně fungovat dle kódu.

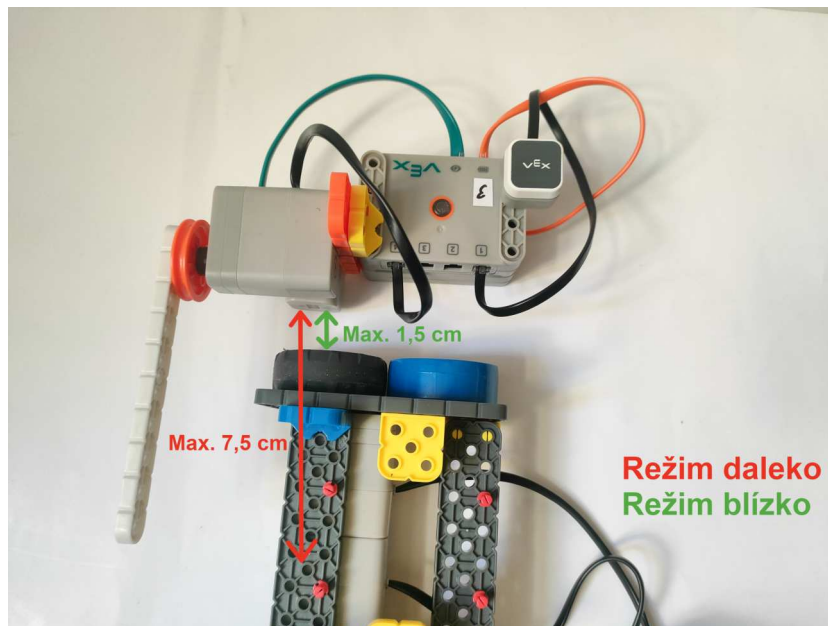
Tento problém se opětovně vyskytl v následujícím úkolu 7.9.2, kdy se závora opět sama otevírala a zavírala. Tentokrát se problém objevil i tehdy, když závora byla na dřevěném stole, nestála na bílém papíře a doplňkové světlo senzoru bylo vypnuto. Bylo zjištěno, že tento problém nastává při snížených světelných podmínkách v testovaném prostředí. Když byla pracovní plocha se závorou osvětlena stolní lampou nebo byla závora přesunuta k oknu s více dostupným denním světlem, začala opět fungovat správně podle naprogramovaného kódu.

Kapitolu bylo možné přepracovat, ačkoli v jistých úlohách s mírnými obtížemi. V prvním úkolu 7.2.3 nelze replikovat kód, který je založený na využití ultrazvukového senzoru. U stavebnice LEGO tento senzor dokáže měřit vzdálenost, avšak VEX GO touto funkcí nedisponuje. Oční senzor VEX dokáže rozlišovat pouze zdali se objekt nachází před senzorem, případně určit jednu ze tří barev anebo určit intenzitu jasu.

S tím je spjatá celková funkcionálnost očního senzoru, která byla dlouho nejasná. Senzor začal správně fungovat až po diskusi s pracovníky vexrobotics na veřejném fóru, kteří po určité době zařídili aktualizaci programovacího prostředí. Teprve následně bylo docíleno správné funkčnosti. K povšimnutí v kódech této kapitoly je blok v přesném znění „nastavit oko na daleko rozsah“, který ovlivňuje, jak daleko je senzor schopný vnímat předměty před sebou samým. V programech kapitoly byl zvolen rozsah „daleko“, jelikož to umožňuje senzoru vnímat objekty do maximální vzdálenosti cca 7,5 cm. V opačném případě režim „blízko“, umožňuje vnímat objekty do maximální vzdálenosti cca 1,5 cm.

Další nevýhodou je kombinace senzoru barev a senzoru pro rozpoznání objektu. Tato nevýhoda se týká úloh se závorou, kdy je potřebné mezi objekt a senzor umístit barevnou kartu, či barevný předmět pro rozpoznání typu vozidla. V případě, že by objekt stál blízko senzoru, je prostor pro umístění karty omezený, například na

místo označené zelenou šipkou, jak je vidět na obrázku pod tímto textem. Taktéž barevná karta nesmí být před senzorem umístěna příliš dlouho, aby neovlivnila dobu, po kterou je závora zvednuta.



Obrázek 90: *Vzdálenosti režimů očního senzoru VEX GO*

Možným řešením může být přidání dalšího senzoru, který by fungoval samostatně pro rozlišování barev. Ovšem řídicí jednotka stavebnice VEX GO umožňuje připojit pouze jeden oční senzor, pro který je vyhrazen jen jeden port.

V této kapitole je již patrné, že programování do řádku není nejvhodnější variantou při rozsáhlejších programech. Úlohy a úkoly této kapitoly byly naprogramovány s kratším a přehlednějším kódem ve stavebnici VEX GO.

5.8 Kapitola 8: Adaptivní tempomat, detekce překážky

Osmá kapitola opouští úlohy a problematiku týkající se závory a vrací se k práci s pojízdným robotem, se kterým žáci pracovali v prvotních kapitolách učebnice. Cílem kapitoly je naučit žáky řídit pojízdného robota s pomocí ultrazvukového senzoru v rámci předpokládaných dvou vyučovacích hodin.

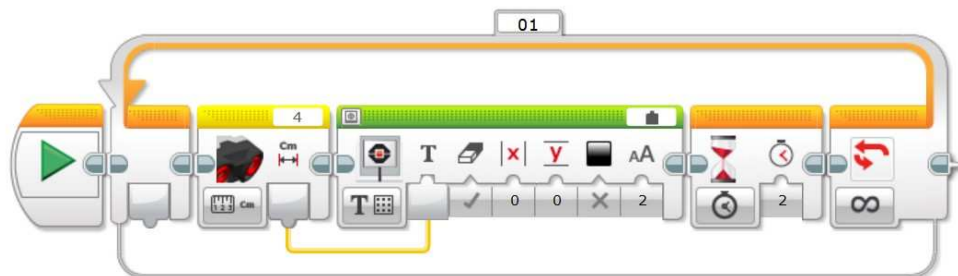
Žáci se s ultrazvukovým senzorem setkali poprvé v předchozí sedmé kapitole. Tato kapitola však obsahuje úlohy zaměřené výhradně na tento senzor, aby usnadnila plynulejší pochopení jeho použití. Dále pracují s displejem a hodnotami, které se na něj vypisují. Primárně by se žáci měli naučit robota zastavit před

překážkou, upravovat rychlost robota podle pohybující se překážky a stabilizovat robota tak, aby se vyhnuli trhavým a prudkým pohybům.

Kapitola je řazena mezi náročnější, proto se v ní objevuje mnoho otázek pro žáky týkající se daných úkolů. Ty mají vést především k zamyšlení nad praktickou funkčností, realizací a samotnou funkčností kódů. V této kapitole mají žáci taktéž předpřipravené jisté programy, které mají otestovat a identifikovat, který kód je nejvhodnějším řešením.

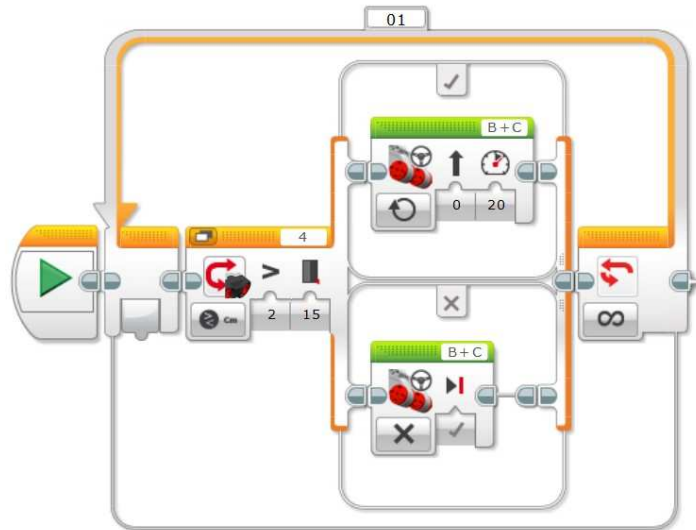
5.8.1 Adaptivní tempomat, detekce překážky – původní učebnice

Kapitola začíná úlohou, která má seznámit žáky s programováním ultrazvukového senzoru. Jelikož s tímto senzorem budou pracovat po celou kapitolu, je nutné, aby pochopili, jak senzor reaguje, jak funguje a jakou má přiměřenou přesnost měření. V úkolu 8.1.1 mají žáci naprogramovat metr, který bude vypisovat naměřené hodnoty na displej. V záložce 8.2.1 je představen nový blok pro spojení výpisu např. hodnoty ze senzoru a napsaného textu. Záložky 8.1.3 a 8.1.4 obsahují otázky týkající se naprogramovaného metru. Jedna sada otázek se zaměřuje na samotné měření a druhá na funkčnost senzoru.



Obrázek 91: Řešení úkolu 8.1.1

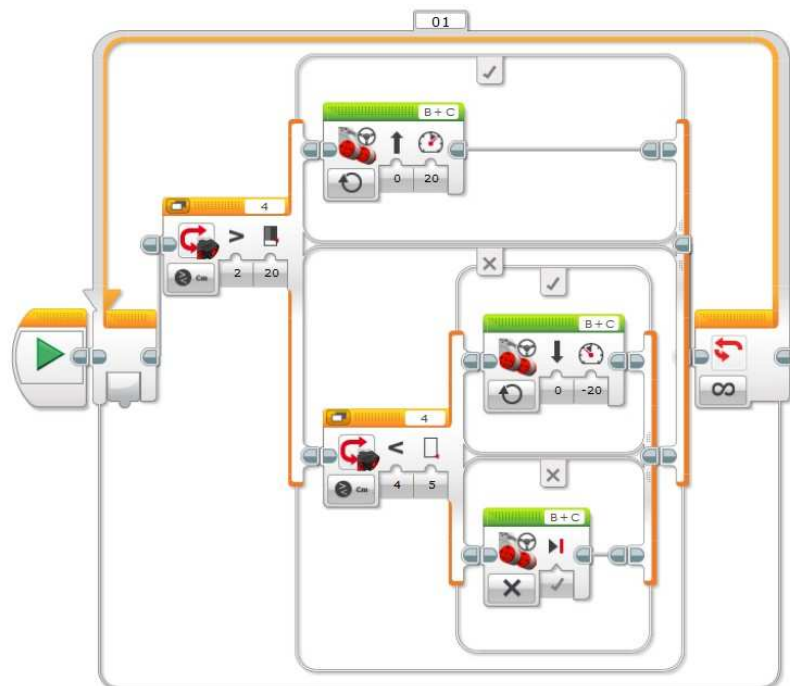
Poté, co se žáci v první úloze seznámí s ultrazvukovým senzorem a pochopí jeho funkcionalitu, mohou přejít na druhou úlohu 8.2, která zadává aktivitu 8.2.1: „Vytvořte program, díky kterému Váš robot pojede neustále rovně vpřed, a pokud se před ním objeví překážka, zastaví od ní v bezpečné vzdálenosti 15 cm.“ V tomto úkolu by se mělo jednat o opakování z předchozí kapitoly, kdy téměř stejnou podmínku na podobném principu využili již v kapitole o automatické závoře.



Obrázek 92: Řešení úkolu 8.2.1

V záložkách 8.2.2 a 8.2.3 jsou otázky vztahující se ke kódu výše (obrázek č. 92). Tyto otázky žáci zodpovídají až po otestování programu, který sami vytvořili.

Začátkem kapitoly 8.3 mají žáci již naprogramovaného robota s bezpečnostním prvkem, tedy robota, který umí zastavit před překážkou. V této úloze mají žáci upravit program robota tak, aby udržoval od překážky vzdálenost 20 cm. V úkolu 8.3.1 je zadáno, že robot má splňovat funkci stálého udržování 20 cm od překážky. V případě, že se překážka začne vzdalovat od robota směrem dopředu, robot ji dojede a v opačném případě přibližující se překážky, robot začne couvat.



Obrázek 93: Řešení úkolu 8.3.1

V úkolu 8.3.2 se mají žáci zamyslet, zdali na něco nezapomněli při vytváření předchozího kódu. Včetně toho jejich program testují a odpovídají na doplňující otázky. Úkol 8.3.3 je volnější a zadává úkol více skupinám ve třídě současně. Žáci mají vzít své naprogramované pojízdné roboty, dát jich několik za sebe a vytvořit tak vláček. Tímto úkolem žáci zjistí, zda své roboty naprogramovali správně. V záložce 8.3.4 je položena otázka týkající se chování robotů. Zdali se žákům u jejich robotů stává, že robot má sekavé a trhané pohyby. V této záložce je přiložený snímek kódu, na kterém mají žáci přijít na to, proč má robot takové pohyby. V záložce 8.3.5 mají žáci objasnění, proč se tato skutečnost u robotů objevuje.

Úloha 8.4 je zaměřena na tzv. stabilizaci robota. Žáci si v záložce 8.4.1 stáhnou soubor s programy, kde se nachází celkem tři kódy, které mají žáci prozkoumat a otestovat na svém pojízdném robotovi. Následně odpovídají na otázky: „*Stabilizace spočívá v*“, „*Hlavní výhody*“ a „*Hlavní nevýhody*“. Tyto otázky jsou stejné pro každý kód ve staženém souboru. V záložce 8.4.2 mají žáci rozhodnout, který kód ze souboru se pro stabilizaci robota jeví jako nevhodnější řešení.

Poslední úloha 8.5 je vyznačena pro rychlejší žáky. Ti mají upravit rychlost robota tak, aby zrychloval, případně zpomaloval k blížící se či oddalující se překážce. Taktéž jej mají upravit, aby při couvání robot vydával zvukovou signalizaci.

5.8.2 Detekce překážky, skener (alternativa) – stavebnice VEX

Již při analýze původní učebnice bylo zjištěno, že tato kapitola bude vyžadovat vnoření návrhu minimálně jedné úlohy nebo alespoň úpravu zadání vybraných úloh. V této kapitole se pracuje s jednotkami vzdálenosti zjištěnými ze senzoru. Bohužel touto možností oční sensor stavebnice VEX GO nedisponuje, a není tak možné replikovat všechny úlohy dle původní učebnice se zachovaným zadáním.

První úloha 8.1 musela být nahrazena, jelikož není možné pracovat s jednotkami vzdálenosti naměřenými očním senzorem, a také proto, že oční sensor stavebnice VEX GO má velmi malý vzdálenostní rozsah. Podrobněji o tom bylo pojednáno ve shrnutí předchozí kapitoly. Žáky lze nechat, aby sami zjistili, do jaké vzdálenosti je možné využít oční sensor stavebnice VEX GO.

Alternativní návrh první úlohy: „Sestavte program, který bude na obrazovku monitoru vypisovat, zda oční sensor vidí, či nevidí objekt umístěný před senzorem. Důležité je na začátku programu vložit blok 'nastavit oko na rozsah', který bude měněn dle následujících úkolů.“

1. Nastavte režim rozsahu oka na „blízko“. Před oční sensor umístěte objekt do takové vzdálenosti, dokud se vypisuje, že oko vidí objekt. Tzn. najděte nejdelší možnou vzdálenost od oka a změřte, kolik centimetrů je mezi okem a objektem.
2. Ve druhém úkolu změňte rozsah oka na režim „daleko“. Opět umístěte objekt před oční sensor a pohybuje s ním, dokud nenajdete nejdelší možnou vzdálenost, při které je stále vypisováno, že objekt je zaznamenán. Změřte tuto vzdálenost v centimetrech.

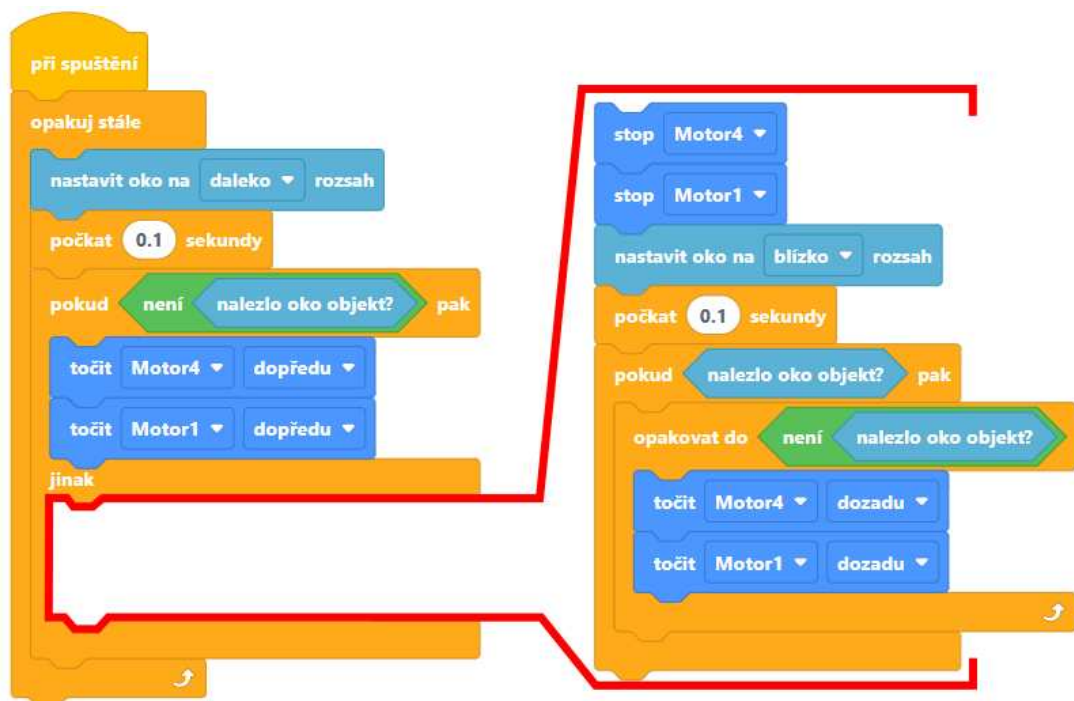
Úloha 8.2 s názvem „bezpečnostní pojistka“ je v principu jednoduchý úkol, ve kterém robot zastaví před překážkou. Původní učebnice obsahuje zadání, že má robot zastavit před překážkou ve vzdálenosti 15 cm. Toto konkrétní zadání není možné dodržet, úlohu lze realizovat pouze za předpokladu, že nebude definována vzdálenost zastavení robota od překážky. V případě, že by byl ve VEXcode GO nastaven rozsah oka na „daleko“, robot zastaví před překážkou přibližně sedm centimetrů od ní. V rozsahu „blízko“ by robot zastavil těsně před překážkou, zhruba ve vzdálenosti jednoho a půl centimetru od objektu. Podobné hodnoty by též měli žáci zjistit v předchozí navržené úloze.



Obrázek 94: Řešení navrženého úkolu 8.2.1

Úloha 8.3 se zadáním: „Vytvořte program, pomocí kterého si bude robot udržovat od překážky před ním (např. auta) odstup 20 cm. Pokud se bude vzdalovat, dojde ji, pokud se bude přibližovat, začne couvat.“ Úlohu je možné vypracovat s podobnou funkčností, avšak stejně jako u úlohy na předchozí straně zde nelze dodržet konkrétní vzdálenost (20 cm). Tuto úlohu lze realizovat za předpokladu, že se robot v režimu „daleko“ zastaví před překážkou ve vzdálenosti přibližně sedm centimetrů. Pro splnění zadání, že má robot začít couvat při přibližování se překážky, je však nutné v průběhu chodu programu přepnout na režim „blízko“. Tímto způsobem se podaří splnit podmínku, ovšem s tím, že předmět se musí přiblížit k očnímu senzoru do vzdálenosti zhruba necelých dvou centimetrů.

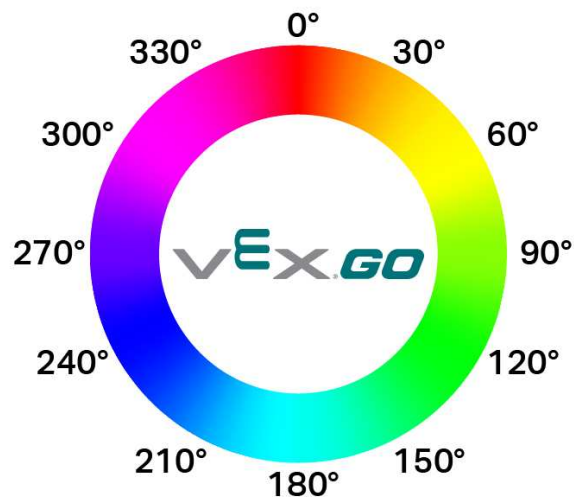
Pro správnou funkčnost bylo nutné do kódu přidat dva bloky „počkat 0,1 sekundy“, jelikož při testování bylo zjištěno, že senzor tak lépe zpracovává hodnoty, když se v průběhu kódu mění režim „blízko a daleko“. Také pro lepší plynulost robota bylo nutné do poslední podmínky přidat blok „opakovat do“. V případě, že byly dva bloky „točit Motor dozadu“ vloženy mimo tento blok, pouze v podmínce „pokud oko našlo objekt“, robot vykonával kmitavé a trhavé pohyby dopředu a dozadu.



Obrázek 95: Řešení úlohu 8.3.1

Mnoho úloh v původní učebnici se zaměřuje na realistické věci, které žáci mohou znát z reálného světa (viz kapitoly typu: mixér, závora nebo v učebnici též následující parkovací asistent). Následující navržená úloha má do jisté míry připomínat skener, který se v praktickém světě může využívat například k inventarizaci zboží ve skladech či obchodech.

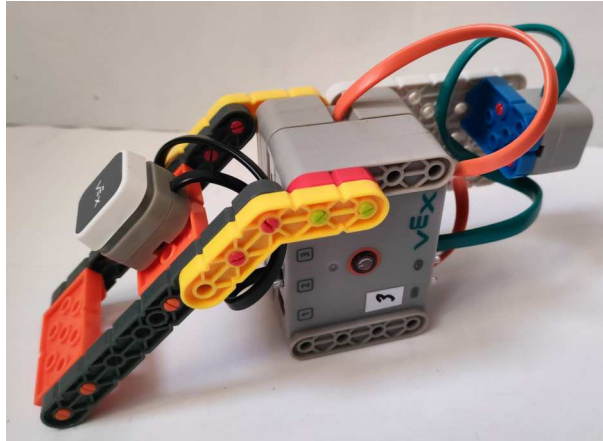
Pro zjednodušení mají žáci za úkol sestavit skener, který bude rozpoznávat jednotlivé barvy, jež budou následně vypisovány na displej monitoru. Přestože oční senzor stavby VEX GO rozpoznává v základu pouze tři barvy (červenou, zelenou a modrou), ve VEXcode GO se nachází blok „odstín oka ve stupních“. Tato možnost hlásí hodnoty mezi 0-359 stupni [56].



Obrázek 96: *Stupnice barev dle číselné hodnoty jasu z očního senzoru. Zdroj obrázku: VEX Library [56].*

Díky této stupnici mohou žáci naprogramovat skener tak, aby byl schopen rozpoznat i jiné než předem definované barvy. Vzhledem k tomu, že oční senzor ohlašuje hodnoty nepřetržitě, na monitoru by se vypisovalo příliš mnoho dat. Proto je úloha koncipována tak, aby skener hlásil hodnoty (barvy) pouze po stisknutí tlačítka.

Poté, co žáci sestaví skener, může následovat návrh zadání: „Naprogramujte skener barev, který po stisknutí tlačítka vypíše na obrazovce v programovacím prostředí název barvy, na kterou senzor ukazuje. K identifikaci jednotlivých barev využijte barevnou stupnici s číselnými hodnotami. Náповěda: V programu tohoto úkolu je nutné využít několik podmínek. Každá podmínka bude definovat rozsah hodnot jasu (tedy jednu konkrétní barvu), které zaznamenává oční senzor.“



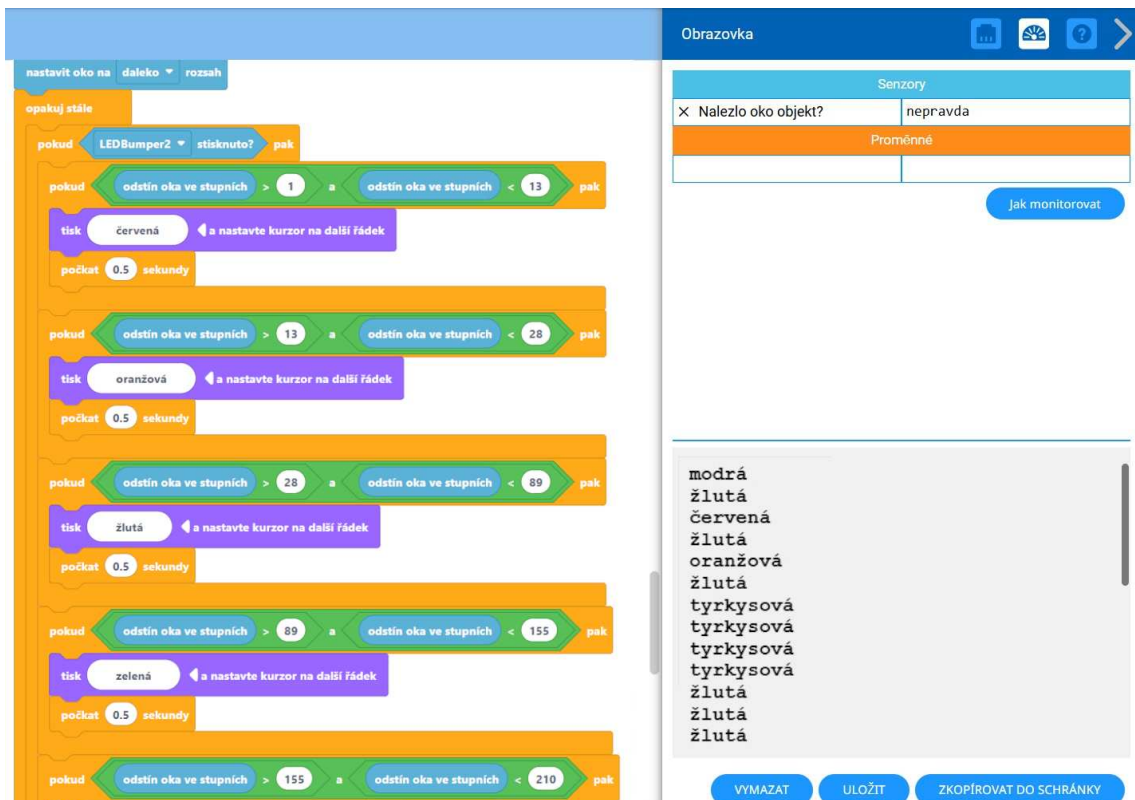
Obrázek 97: *Sestavený skener barev ze stavebnice VEX GO*

Vzhledem k situaci, že na stupnici mezi barvami jsou mnohdy mírné rozdíly (pár stupňů), a ne vždy je patrné, o jaké hodnoty se jedná, předpokládá se, že není možné identifikovat všechny barvy na stupnici. Není proto nutné dbát na to, aby žáci rozlišovali rozdíly například mezi světle zelenou, olivovou a tmavě zelenou (pro vytvoření takového kódu je potřebné dostatečné množství vzorků různých barev). Žádoucí naopak je, aby žákův sestrojený kód dokázal identifikovat i jiné než základní tři barvy. Příkladem tak může být žlutá, růžová nebo tyrkysová.

Žáci v této úloze experimentují se stupni a identifikují jednotlivé barvy na stupnici. Tato úloha může dále pokračovat diskusí o tom, jaké hodnoty stupňů pro jednotlivce představují jednotlivé barvy, jako například růžová či červená. Protože každý žák může mít rozdílný názor na číslo stupňů, které rozdělují pomyslnou hranici mezi růžovou a červenou barvou.

Mírnou nevýhodou této úlohy je osvětlení prostoru a skenovaných barev, jelikož množství světla ovlivňuje hodnoty očního senzoru. Z testování této úlohy je možné tvrdit, že přísun přirozeného denního světla z jednoho okna je pro tuto úlohu dostatečný.

Výňatek z rozsáhlého kódu, jenž se vztahuje k této úloze, je spolu s příslušným ilustrativním výpisem zobrazen na následující straně na obrázku č. 98.



Obrázek 98: Vypisované barvy, které snímá skener (oční senzor)

5.8.3 Shrnutí kapitoly 8

V této kapitole se opět jeví jako problematický oční senzor stavebnice VEX. Vzhledem k tomu, že není možné senzorem měřit vzdálenost v centimetrech, bylo nutné nahradit nebo alespoň upravit zadání každé úlohy v této kapitole.

Namísto elektronického metru, což je první úloha této kapitoly, byla navržena úloha, která seznamuje žáky s funkcí očního senzoru stavebnice VEX GO. Stejně tak s jeho možnostmi dvou režimů, respektive možnými vzdálenostmi, do kterých je možné senzor využít. Naměřené hodnoty mohou žáci brát v úvahu při tvorbě úlohy s názvem „bezpečnostní pojistka“.

Do této kapitoly byla přidána další alternativní úloha, jejímž cílem je zkonstruovat a naprogramovat skener barev. Při testování a tvorbě byla úloha překvapivě nad očekávání funkční. Tato úloha, vyjma programování nutí žáky k objevování a experimentování s tím, jaké hodnoty reprezentují danou barvu. Hodnoty mají žáci čerpat z teoretického obrázku barev, který byl převzat z oficiálních stránek vexrobotics. V samotném závěru úlohy je vhodné ladění programu, aby docházelo k co nejpřesnější identifikaci barev.

Tuto kapitolu lze zhodnotit jako jednu z nejvíce upravovaných, jelikož úlohy z původní učebnice neumožňují správnou funkčnost u stavebnice VEX GO. Alternativní úlohy byly zařazeny do této kapitoly, jelikož takto odpovídají konceptuálnímu schématu práce žáků, kdy v této kapitole detailněji pracují s očním senzorem. Není proto vhodné tyto úlohy zařadit do čtvrté kapitoly, kde se pracuje s displejem, protože žáci tam ještě nepracovali s právě zmiňovaným senzorem.

5.9 Kapitola 9: Inteligentní pojízdný robot

Devátá kapitola se opět inspiruje skutečným světem. Zatímco v šesté a sedmé kapitole měli žáci jisté představy o závoře, se kterou se mohou běžně setkat, tato kapitola se váže například k realistickým robotům, kteří se mohou specifickým způsobem pohybovat v průmyslových halách.

V této kapitole, která je doporučena na dvě vyučovací hodiny a je klasifikována jako náročnější, se mají žáci nejen naučit nové funkci, ale zároveň prohloubit znalosti o barevném senzoru. Vyjma rozpoznání barev mají žáci pracovat s intenzitou odraženého světla, kterou barevný senzor zaznamenává. Cílem kapitoly je, aby se žáci kromě seznámení s novou funkcí senzoru také naučili naprogramovat robota, který se bude pohybovat následováním vodící čáry.

Kapitola, vyjma samotných úkolů, obsahuje velkou část informací o barevném senzoru, otázky na přemýšlení o praktickém využití senzoru a jeho správné kalibraci. Ke splnění této kapitoly je vhodná předchozí příprava (plánek města z kapitoly tři, jiný plánek obsahující vodící čáru nebo připravenou čáru na podlaze).

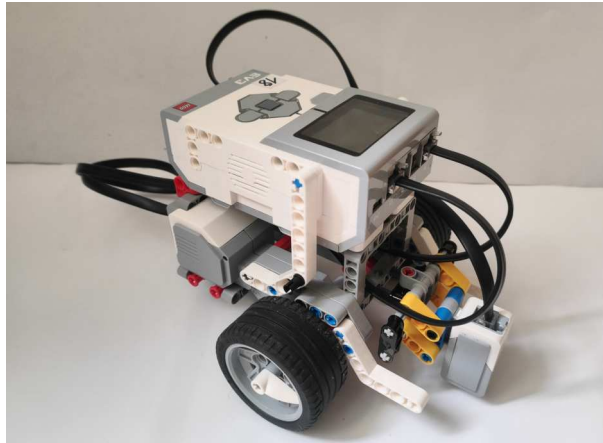
5.9.1 Inteligentní pojízdný robot – původní učebnice

Úvodní úlohou kapitoly je úloha 9.1 „rozpoznání čáry“, která začíná teoretickými informacemi o dostatečném kontrastu vodící čáry od okolního prostředí a uvažujícími otázkami nad využitím barevného senzoru.

Úloha 9.2 pokračuje dalšími informacemi, jak by se měl robot pohybovat po čáře, jak by měl daný pohyb vypadat a jaký směr by měl robot vykonat při zaznamenání určité barvy.

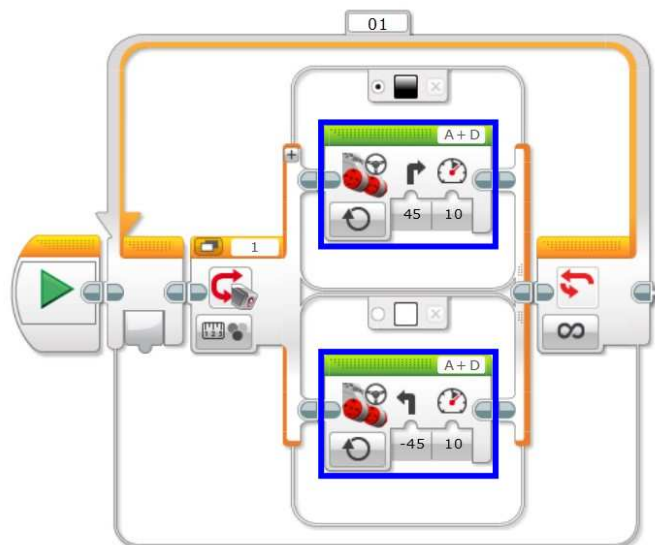
Úloha 9.3 zadává žákům pokyn k umístění senzoru na konstrukci pojízdného robota a nabízí možnost přiloženého návodu. Taktéž se zde objevuje zásadní

informace, aby žáci se senzorem experimentovali, v jaké poloze (náklon a výška) od vodící čáry se má nacházet, jelikož se může vyskytnout situace, kdy sensor bude špatně umístěn a kód nebude funkční i při správném naprogramování. Možnost tohoto ladění umožňuje pohyblivá konstrukce pro barevný sensor.



Obrázek 99: *Pojízdný robot z první kapitoly s přidaným barevným senzorem
LEGO Mindstorms*

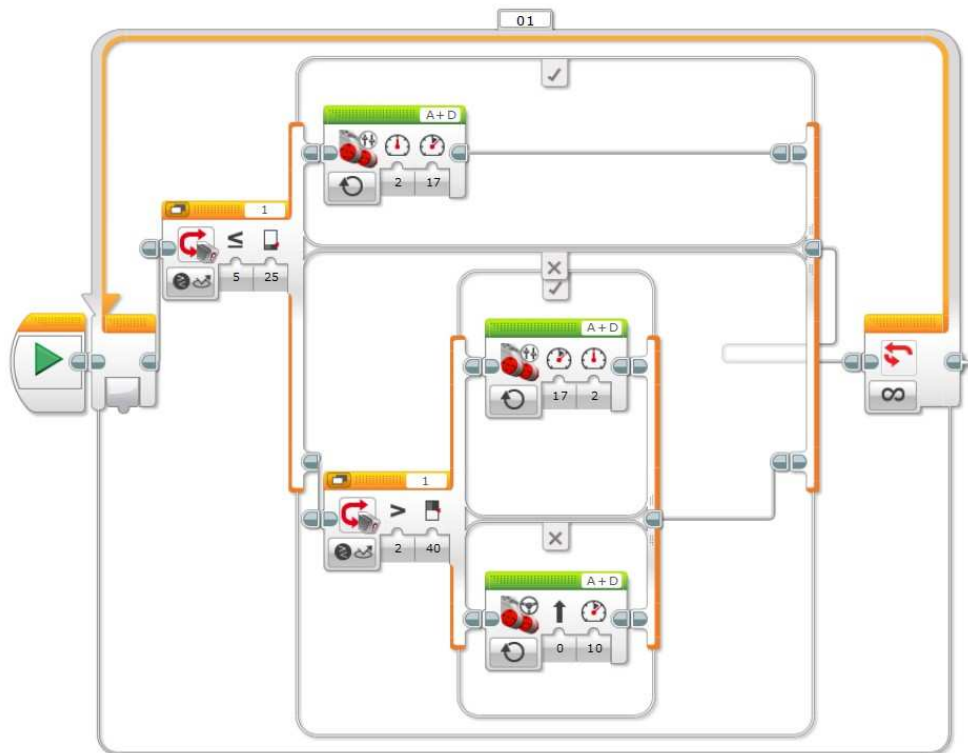
První programovací úlohou deváté kapitoly je úloha 9.4, která zadává žákům, aby doplnili část vyobrazeného kódu (původní kód, který mají žáci doplnit, je bez modře vyznačených částí na obrázku níže). Původní kód zahrnoval pouze spouštěcí tlačítko a jednu prázdnou podmínku. Žáci mají tuto podmínku doplnit (modře vyznačeno) tak, aby se robot pohyboval na základě vnímání černé a bílé barvy. Více o tomto pohybu byli žáci informováni v předchozích informačních úlohách této kapitoly.



Obrázek 100: *Řešení úlohy 9.4*

Jelikož se i sami žáci mohou setkat se skutečností, že sensor bude špatně vnímat intenzitu světla, je nutné jej správně zkalibrovat. Touto problematikou a správným nastavením senzoru se zabývá úloha 9.5 konkrétně v úkolech 9.5.1–9.5.5. Úkol 9.5.4 zadává následující úkol: „*Provedte kalibraci světelného senzoru. Nastavte minimální a maximální hodnotu pro snímání barvy bílého okolí a černé čáry. Nezapomeňte přidat interakci s uživatelem.*“ Z důvodu, že nebyla zjištěna podobná funkce a nastavení u stavebnice VEX, nejsou tyto úkoly dále detailněji popisovány.

S ohledem na to, že robot po spuštění programu vyobrazeného na předchozí straně (obrázek č. 100) vykonává kmitavé a trhané pohyby, úloha 9.6 se zaměřuje na plynulejší pohyb robota. Vzhledem k její náročnosti, je označena jako úloha pro rychlejší žáky. Plynulejšího pohybu lze docílit nikoli vnímáním barev, ale vnímáním intenzity odraženého světla, které dokáže barevný sensor zaznamenávat. Je nutné zmínit, že po následném vytvoření této úlohy je třeba sensor barev resetovat, jelikož po kalibraci z předchozí úlohy vrací jiné hodnoty než před kalibrací.

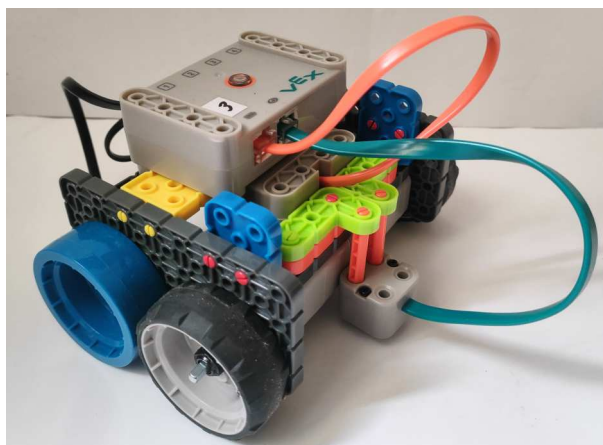


Obrázek 101: Řešení úlohy 9.6

5.9.2 Inteligentní pojízdný robot – stavebnice VEX

Úlohy 9.1–9.2 netřeba dále rozepisovat, jelikož jejich obsah byl popsán v odstavci o původní učebnici, a mimo jiné tato část kapitoly neobsahuje žádný kód či zadání pro programování.

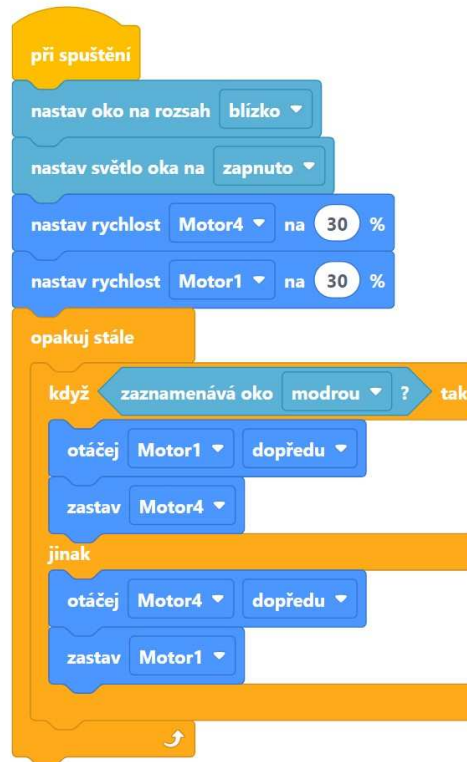
V úloze 9.3 mají žáci připevnit barevný senzor k pojízdnému robotovi, ale nejprve musí sestavit držák pro barevný senzor. Senzorem tak lze pohybovat více směry pro správné nastavení úhlu a pro správnou vzdálenost od čáry. Držák je důležitý, protože autoři původní učebnice v této úloze upozorňují na možný výskyt problému, ve kterém senzor bude špatně vnímat barvy, a je proto žádoucí nalézt vhodnou pozici pro senzor. Zde je nutné podotknout, že v rámci replikace úlohy, byl u stavebnice VEX GO sestaven držák na podobném mechanickém principu. Ovšem při testování bylo zjištěno, že tento držák nebyl v praxi zcela vyhovující. V mnoha ohledech byl senzor buď daleko anebo blízko vodící čáry, a ne vždy robot správně splňoval dané pokyny. V případě nalezení vhodné výšky senzoru od čáry, se při prudkých pohybech robota držák pomalu ohýbal k zemi a senzor opět špatně reagoval. Nakonec byl zvolen držák a upevnění senzoru dle oficiálního návodu od výrobce. Takové umístění senzoru je sice neměnné, tedy nelze se senzorem pohybovat nahoru a dolů jako v případě původní učebnice, avšak výška senzoru je nastavena v nejlepší možné pozici, kdy senzor správně funguje.



Obrázek 102: *Pojízdný robot z první kapitoly s přidaným senzorem VEX GO*

Zatímco v úloze 9.4 (obrázek č. 103) původní učebnice žáci využili podmínku založenou na dvou barvách (černá a bílá) u VEX GO není možné rozeznávat více než tři barvy (červená, zelená a modrá). Byla proto zvolena podmínka: „když oko

vidí modrou barvu tak... jinak..." a musí se tak brát v potaz, že vodící čára bude nakreslena modře. Dále v tomto kódu bylo nutné do podmínky přidat blok „zastav motor“, aby robot vykonával žádoucí kmitavé pohyby do stran a správně vnímal vodící čáru.



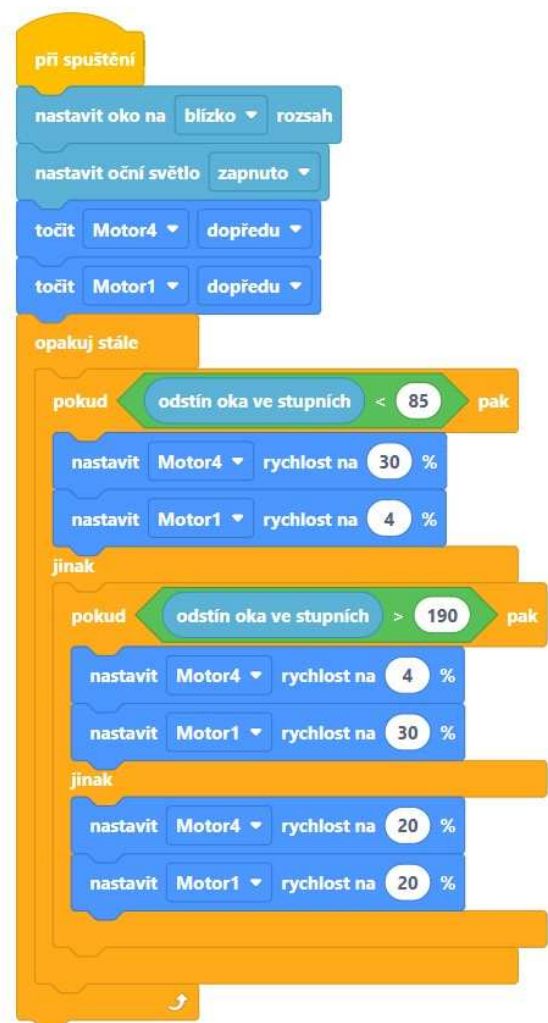
Obrázek 103: Řešení úlohy 9.4

Úlohu 9.5 a její úkoly pro kalibraci senzoru není u stavebnice VEX GO možné realizovat. Avšak nakonec se při zkoumání možností této úlohy ukázalo, že byla velmi nápomocná při zpracování následující úlohy 9.6. Právě v důsledku toho, že není možné žádným způsobem senzor zkalibrovat, se funkce robota při zpracování v programu pomocí bloku „jas oka v %“ ukázala být neefektivní. Přitom tento blok ve své podstatě funguje podobně jako v původním softwaru. Hodnoty jasu senzor ukazuje od nuly do sta, přičemž se velice často pohybovaly od 20 do 60 nebo se zobrazovala hodnota 100 %. Bylo proto náročné přijít na to, jaké hodnoty by byly nejvhodnější pro podmínky, aby robot následoval čáru. Ačkoliv se hodnoty vyobrazené na obrázku č. 104 jeví jako nejlepší varianta, mnohokrát se stávalo, že robot z vodící čáry odjel zcela mimo. Lze tvrdit, že toto řešení pro úlohu 9.6 není nejvhodnější variantou.

Jelikož při trhavých pohybech robota z úlohy 9.4 mohou nastávat problémy při navigaci podle čáry, je vhodné vytvořit program pro plynulejší pohyb. Tím se zabývá úloha 9.6, která je určena rychlejším žákům. O řešení úlohy je pojednáno v textu na konci předchozí strany, avšak s nedokončeným závěrem, že řešení skrze intenzitu odraženého světla není nejvhodnější variantou. Lepší plynulosti bylo dosaženo pomocí bloku „odstín oka ve stupních“, který se již objevil v návrhu osmé kapitoly se skenerem barev. Rozsah hodnot v podmínkách byl zjištěn experimentem na nakreslené čáře. Tyto hodnoty jsou určeny pro čáru modré barvy.



Obrázek 104: Nevhodné řešení úlohy 9.6 pomocí intenzity odraženého světla



Obrázek 105: Řešení úlohy 9.6 pomocí odstínu oka ve stupních

5.9.3 Shrnutí kapitoly 9

Kapitolu bylo možné realizovat u téměř všech úloh, vyjma jedné. To bylo důsledkem toho, že není možné barevný senzor (ve VEX GO oční senzor) kalibrovat tak, jako tomu bylo u LEGO. Právě to byl důvod, proč úloha 9.5 nebyla realizována u stavebnice VEX. Ovšem byla velmi přínosná pro realizaci úlohy 9.6.

U stavebnice VEX docházelo v této kapitole k dále jmenovaným problémům. Když se nastavila příliš vysoká rychlost (40 % a výše), stávalo se, že robot nezaznamenal vodící čáru a vyjel z vymezeného prostoru. V případě nastavení nízké rychlosti (např. 20 %) byl robot důkladnější a čáru mnohem lépe následoval, ačkoli značně trhanými pohyby. Obecně se při nastavení nízké rychlosti otáčení motorů u této stavebnice vyskytují nevhledné trhané pohyby robota.

Dalším často opakovaným problémem, který se objevoval při testování úloh této kapitoly bylo, že se robot z neznámých důvodů zasekl a prováděl pouze jednu vyhrazenou část kódu. Tedy například se opakovaně stávalo, že robot vyjel z vodící čáry, a v kódu ve VEXcode bylo stále znázorněno, že oční senzor vnímá modrou barvu (ačkoli již byl na bílém prostoru). Čímž se robot začal točit do kruhu a i když byl v průběhu chodu přesunut na libovolné místo, na dané situaci se nic nezměnilo a bylo nutné program restartovat.

5.10 Kapitola 10: Parkovací asistent

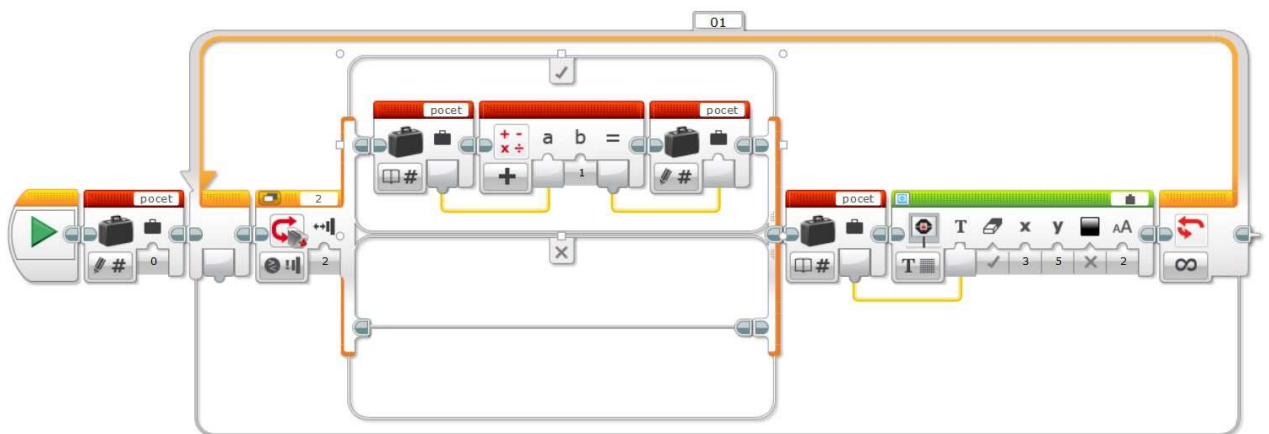
V předposlední, desáté kapitole se žáci zabývají parkovacím asistentem, se kterým se mohou setkat u modernějších automobilů. Jedná se tedy o další kapitolu, kde žáci mohou vidět reálné využití podobného mechanismu, jaký si sami vyzkouší naprogramovat. Navazují tak na získané zkušenosti z předchozích úloh a na práci s ultrazvukovým senzorem. Cílem kapitoly je, aby se žáci naučili kromě samotného vytváření kódů také analyzovat, pochopit a upravit již existující kód.

Mimo jiné mají žáci naprogramovat vlastní kód pro problematickou situaci, aby robot dokázal identifikovat volné místo k zaparkování mezi objekty. Postupnou gradací se žáci dopracují k naprogramování robota, který zaparkuje na dané volné místo. Mimo jiné je v této kapitole popsáno vytváření vlastních bloků pro lepší přehlednost kódů. Zatímco předchozí, devátá kapitola obsahuje více teoretických informací, v této kapitole se opět objevují dílčí otázky týkající se daných kódů

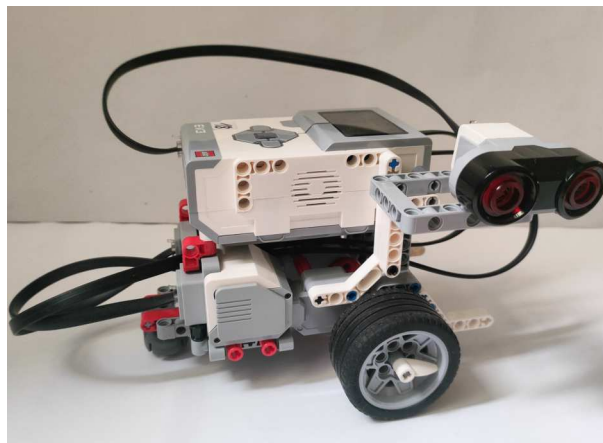
a otázky ohledně obecné problematiky parkovacího asistenta. V neposlední řadě má kapitola za cíl žákům ukázat vhodnost rozdělení velkého problematického celku na menší části.

5.10.1 Parkovací asistent – původní učebnice

Kapitola začíná úlohou 10.1, kde žáci jako první analyzují a popisují funkčnost kódu, který je vyobrazen v rámci úkolu 10.1.1 s názvem „poznáváme“. V tomto kódu je žákům představen nový blok, který může být užitečný v následujících krocích. Dále žáci popisují, k čemu slouží vyobrazený kód na obrázku č. 106. V úkolu 10.1.2 je stále vyobrazen ten samý kód jako v předchozí záložce, s tím rozdílem, že mají žáci odpovědět na otázku, co by se stalo, kdyby byla přepsána hodnota v prvním bloku pojmenovaném „počet“. Úkol 10.1.3 směřuje svými otázkami na zamyšlení se nad tím, jak by fungoval kód, kdyby byl přesunut první blok (blok s názvem „počet“) do bloku s opakováním.



Obrázek 106: Kód k úloze 10.1

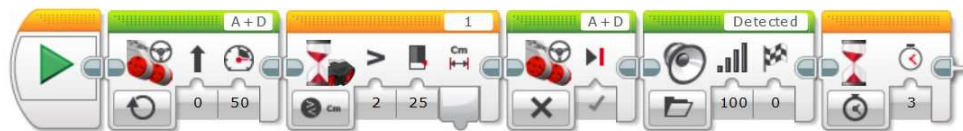


Obrázek 107: Robot z první kapitoly s přidáním ultrazvukovým senzorem

Úkol 10.1.4 taktéž vyobrazuje připravený kód, avšak odlišný oproti předchozím záložkám. Není dále detailněji popisován, jelikož, stejně jako v předchozí kapitole, pracuje s jednotkami vzdálenosti, které nelze replikovat či nahradit. Žáci zde mají opět popsat, jakou má program funkci a co se stane po jeho spuštění. V záložce s úkolem 10.1.5 si mají kód z předchozího úkolu stáhnout do zařízení, upravit jej a následně otestovat. Jelikož samotný kód není pro tuto kapitolu stěžejní (jedná se spíše o ukázkou práce s proměnnou) a o očním senzoru bylo v této práci pojednáno mnohokrát, nejsou dále detaily tohoto úkolu rozepisovány.

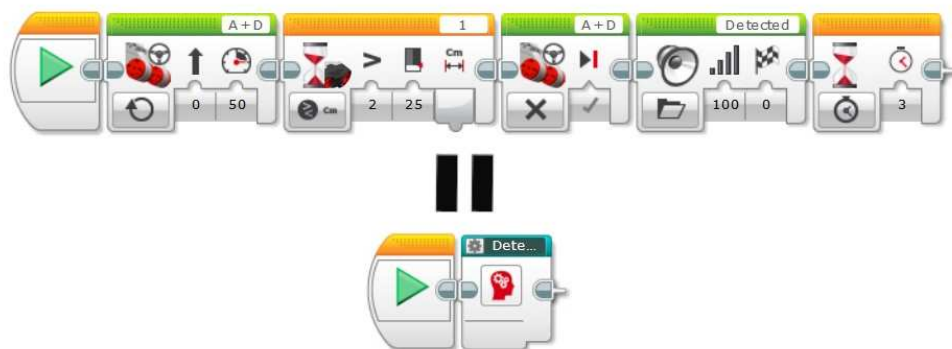
Úkoly 10.2.1-10.2.3, jež jsou součástí úlohy 10.2, plní funkci prvotních plánů před samotným začátkem programování parkovacího asistenta. Žáci si mají rozmyslet, jak bude jejich program fungovat a co bude stěžejní náplní. Především se tato úloha snaží žáky dovést k principu rozdělení velkého problému na menší části.

V první řadě, aby žáci dosáhli cíle této kapitoly, musí naprogramovat kód, který bude nejprve splňovat funkci detekce začátku volného místa. V záložkách 10.3.1-10.3.3 žáci naleznou zadání pro tento úkol, nápovědu a možné řešení.



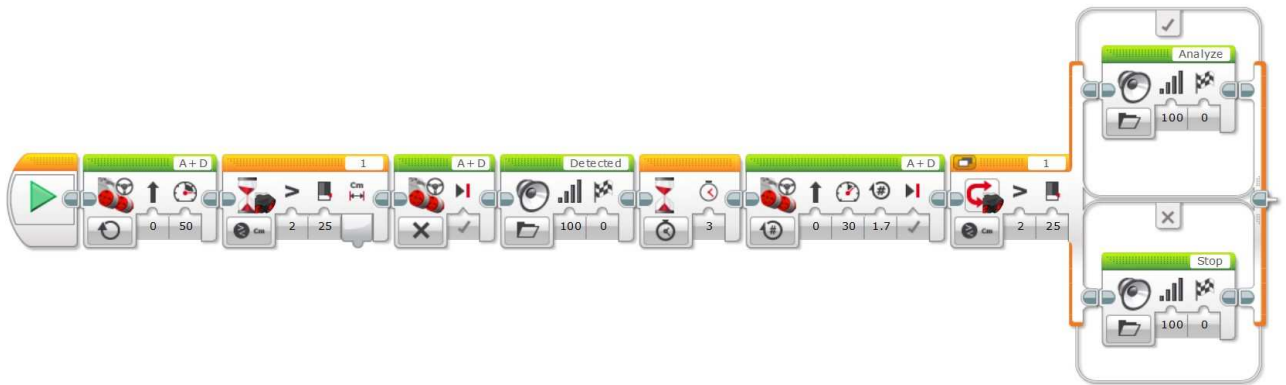
Obrázek 108: Řešení úkolu 10.3.1

Z důvodu rozsáhlosti finálního kódu této kapitoly se žáci v úloze 10.4 seznámí s tím, jak v programovacím prostředí pro LEGO Mindstorms vytvořit vlastní blok. Následně mají kód z předchozího úkolu vnořit do vlastního bloku, který pak zakomponují do svého hlavního programu, jenž budou v této kapitole rozšiřovat.



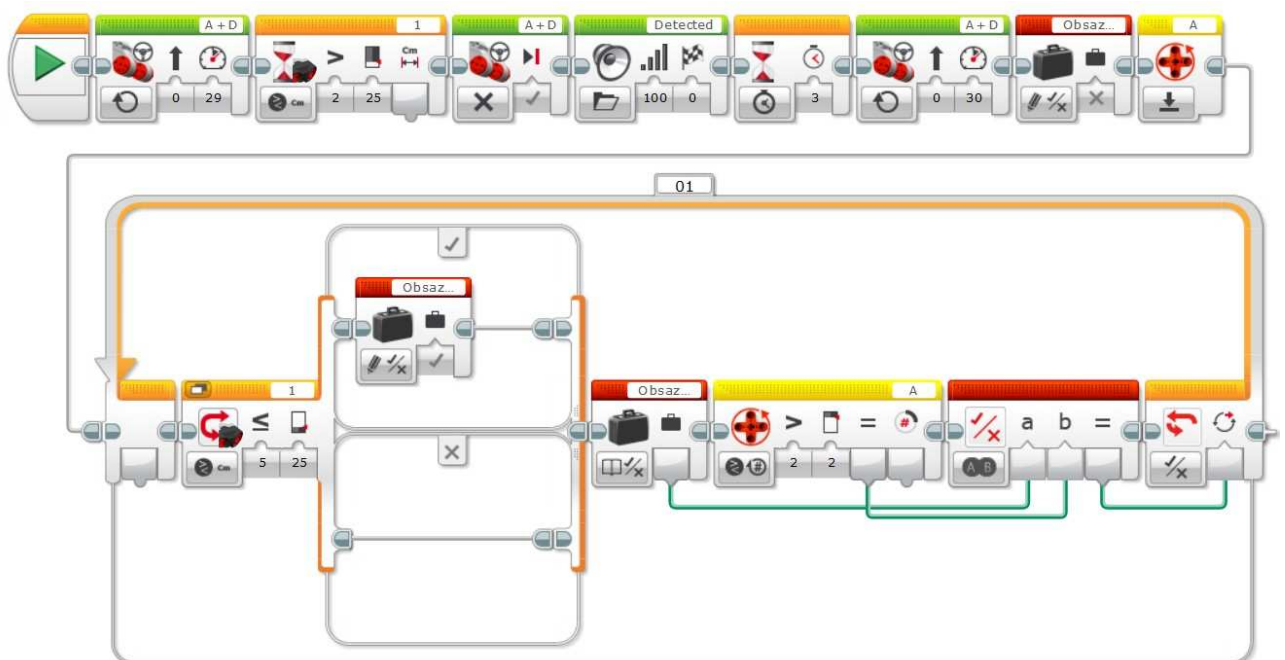
Obrázek 109: Řešení tvorby vlastního bloku v úloze 10.4

Druhým krokem k naprogramování parkovacího asistenta je úloha 10.5, kde je cílem vytvořit kód, který bude měřit mezeru mezi objekty, respektive mezeru vhodnou pro zaparkování robota. Žáci mají rozšířit předchozí program tak, aby po změření volného prostoru robot informoval, zda je mezera dostatečně velká pro zaparkování.



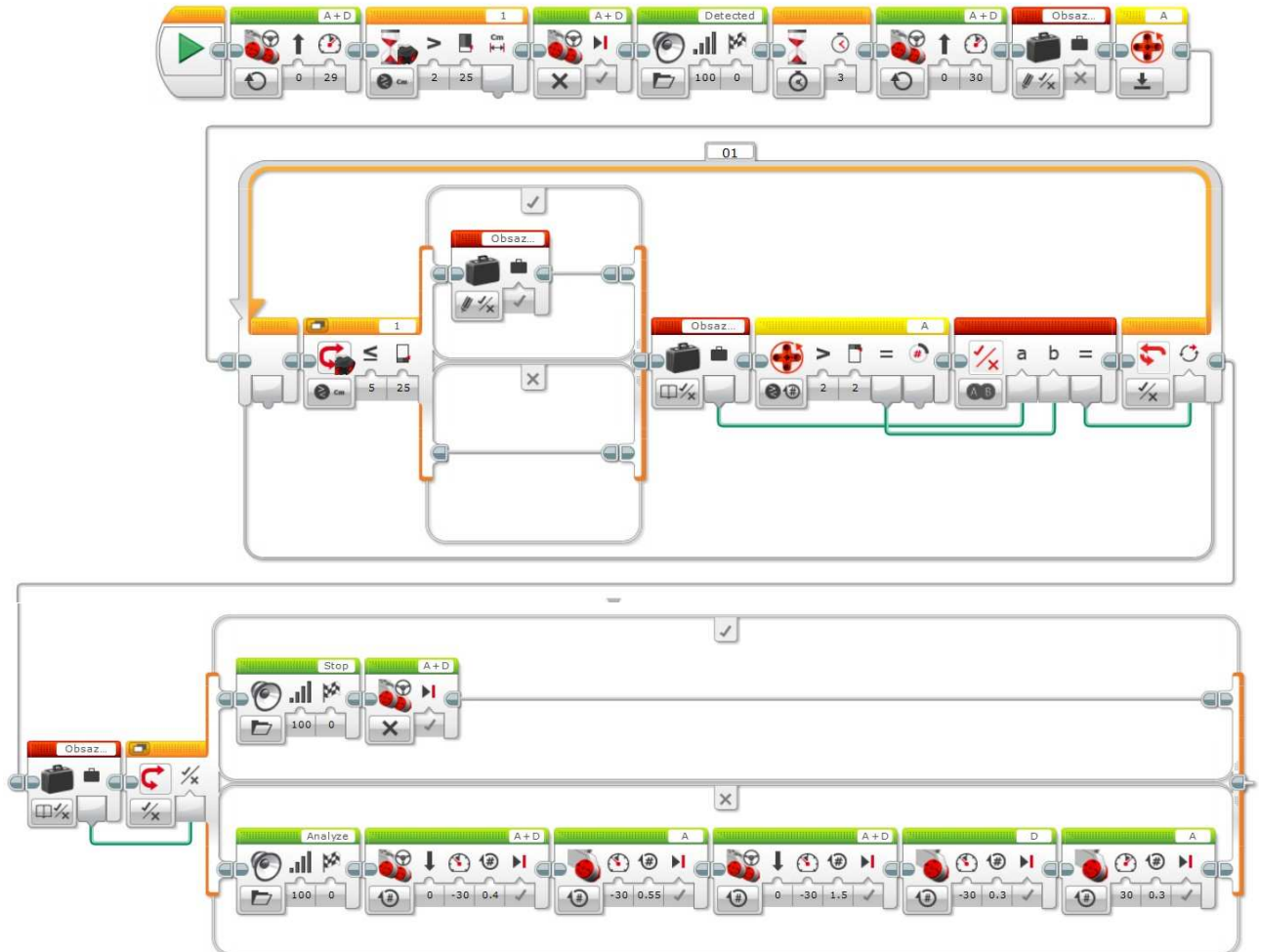
Obrázek 110: Řešení úlohy 10.5

Úloha 10.6 je rozšiřující úlohou pro rychlejší žáky, ve které je představen nový blok pro měření počtu otáček motoru. Úkolem žáků je analyzovat a pochopit funkcionalitu vyobrazeného kódu. Mimo to mají žáci do kódu zakomponovat funkci, která bude zajišťovat detekci menších překážek v místě pro zaparkování (pouliční osvětlení, popelnice...).



Obrázek 111: Řešení úlohy 10.6

Posledním krokem kapitoly je samotné zaparkování pojízdného robota na detekované volné místo. Tento krok má být naprogramován v poslední úloze 10.7. Žáci v tomto případě již nemusí řešit parkování za pomoci senzoru, ale pokud je detekováno volné místo, stačí, aby robot zaparkoval pomocí pevně definovaných pohybů, přesto s podmínkou, aby nenaboural okolní objekty.



Obrázek 112: Řešení úlohy 10.7

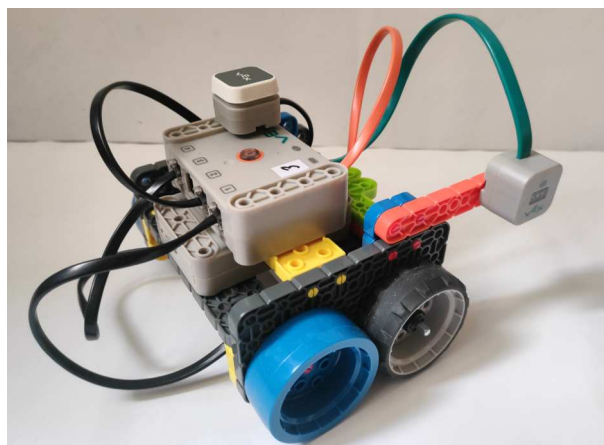
5.10.2 Parkovací asistent – stavebnice VEX

Na začátku kapitoly v úloze 10.1, přesněji v záložkách 10.1.1-10.1.3, je v původní učebnici vyobrazen kód, který mají žáci analyzovat a odpovědět na otázky týkající se kódu. Stručně mají odpovědět na to, co se vykoná po jeho spuštění. Po spuštění kódu se na displeji zobrazí hodnota nastavená v proměnné. Tato hodnota se zvyšuje o jedna, každým stisknutím tlačítka.

Kód z původní učebnice lze bez problému replikovat do stavebnice VEX GO s jeho stejnou funkčností. Jelikož v programovacím prostředí pro LEGO lze nastavit (přímo v nastavení bloku), aby se číslo jedna přičetlo k proměnné až po následném uvolnění tlačítka, byla tato funkce replikována i u kódu stavebnice VEX. Tuto funkci splňuje blok „čekej dokud nenastane“. Blíže o této problematice bylo pojednáno již v páté kapitole. Tímto řešením je zamezeno, neustálému přičítání při držení tlačítka. Jako výstup pro vypisování hodnot, byla opět zvolena „obrazovka“.



Obrázek 113: Řešení úkolu 10.1.1



Obrázek 114: Pojízdný robot s přidáním očním senzorem VEX GO

V úkolu 10.1.4 mají žáci vyobrazený odlišný kód. Opět v tomto případě mají vysvětlit, co se stane po jeho spuštění a jaká je jeho funkcionalita. Kód na displeji robota vypisuje největší naměřenou vzdálenost ultrazvukovým senzorem. Bohužel

tento kód není možné ve VEXcode GO replikovat, jelikož se zde nachází opětovný problém jako v předchozích kapitolách. Stavebnice VEX GO neumožňuje měřit vzdálenost pomocí očního senzoru.

Úloha 10.2 je především úlohou pro diskusi, obsahující otázky týkající se dané problematiky a slouží jako prvotní plán pro tvorbu parkovacího asistenta.

Prvním dílčím krokem k parkovacímu asistentovi je detekce volného místa mezi překážkami. Zadání pro tento dílčí úkol se v původní učebnici nachází v úloze 10.3. U stavebnice VEX lze tuto úlohu bez problémů splnit, ačkoli s mírnými praktickými nedostatky. V první řadě, jak už bylo zmíněno v kapitole o automatické závoře, oční senzor stavebnice VEX GO detekuje předmět pouze do vzdálenosti sedmi centimetrů. Samotný fakt nebrání vytvoření této úlohy, avšak může nastat situace, kdy robot pojede delší dobu podél objektu a začne směřovat šikmo od něj, senzor může vyhodnotit, že objekt se vedle robota již nenachází, ačkoli tomu tak není. Tedy omezujícím faktorem je rozsah měření vzdálenosti senzoru. Přesto při testování v rámci praktické části se tato skutečnost několikrát vyskytla. Jedním z mírných nedostatků je opět nepřítomnost reproduktoru, proto v tomto případě byl detekující zvuk nahrazen světelnou vizualizací (blikáním LED nárazníku).



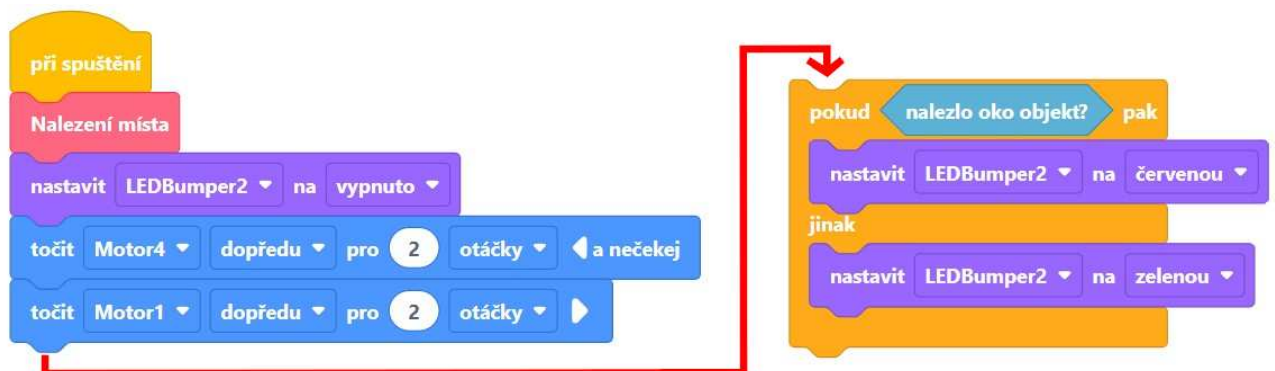
Obrázek 115: Řešení úkolu 10.3.1

Úloha 10.4 je teoretickou částí, která ukazuje jak vytvořit vlastní blok, čímž se dá zkrátit a zpřehlednit kód pro parkovacího asistenta. V některých kódech, předchozích kapitol vytvořené ve VEXcode GO byly již využívány vlastní bloky pro lepší přehlednost. Níže je přiložena ukázka, jak je možné upravit kód z předchozí strany na obrázku č. 115. Vytvoření bloku v tomto prostředí se provádí v levém bočním panelu, pod záložkou „moje bloky“.



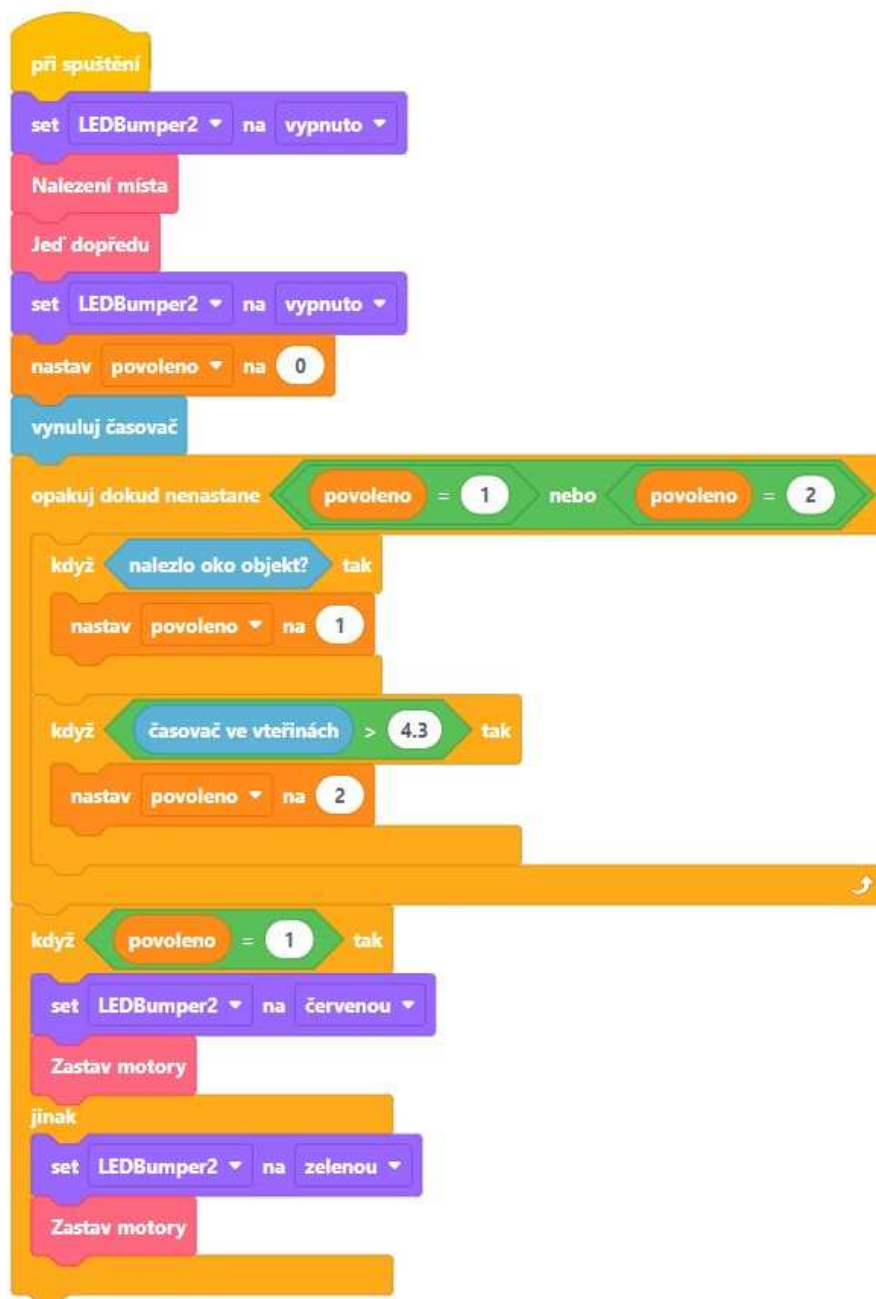
Obrázek 116: Řešení úlohy 10.4

Dalším krokem kapitoly je měření prostoru pro zaparkování. V úloze 10.5 mají žáci vytvořit kód, který identifikuje, zda se mezi objekty nachází dostatečná mezera. Po zjištění mezery vždy vydá vhodné znamení. V původní učebnici se využívalo oznámení o (ne)dostatečném prostoru pomocí zvuků. U stavebnice VEX GO byla zvolena světelná indikace. Pokud bude prostor dostatečně velký pro zaparkování robota, LED nárazník se rozsvítí zeleně, v opačném případě bude svítit červeně. V kódu je umístěn vlastní blok „Nalezení místa“, který pod sebou obsahuje rozsáhlejší kód z obrázku č. 116.



Obrázek 117: Řešení úlohy 10.5

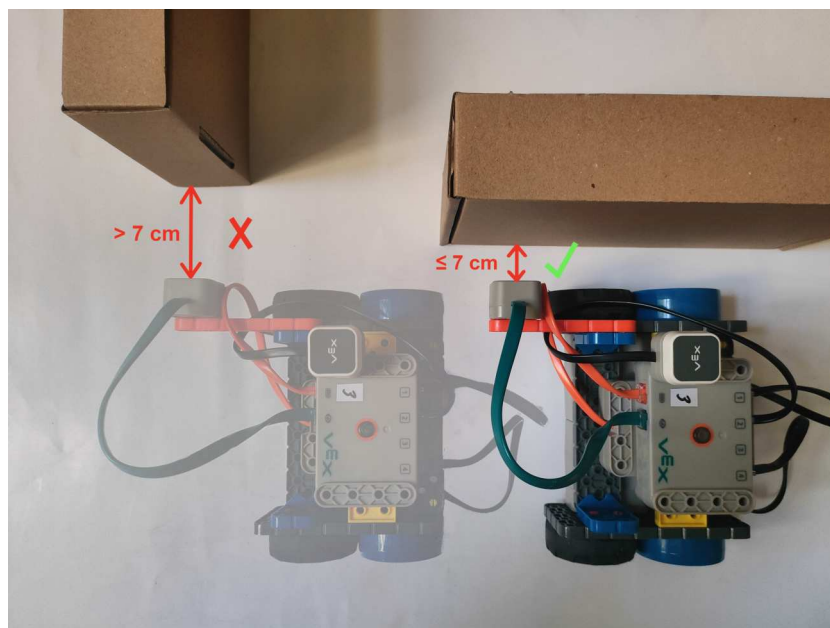
Zadání předchozí úlohy nemusí být ideálním z hlediska fungování robota. Může se stát, že mezi objekty bude překážka menších rozměrů (např. pouliční lampa, popelnice). Proto byla vytvořena úloha 10.6, která řeší tento problém, ale vzhledem k její náročnosti je určena pro rychlejší žáky. Ve verzi pro VEX GO se oproti původnímu kódu z učebnice pro LEGO nacházejí následující dva rozdíly: oční senzor neměří vzdálenost od objektu a namísto počtu otáček robot detekuje volné místo podle času. Protože VEXcode GO neumožňuje práci s počítadlem otočení motoru, bylo zvoleno měření času, jehož optimální hodnota byla určena experimentem.



Obrázek 118: Řešení náročnější úlohy 10.6

Ačkoli se podařilo s jistými úpravami kód vytvořit, tato úloha není nejvhodnější pro stavebnici VEX GO. Objevují se zde určité obtíže, jimiž se opět rozumí viditelnost očního senzoru, která je příliš krátká i v režimu očního senzoru „daleko“. Tato skutečnost byla již popsána ve shrnutí sedmé kapitoly.

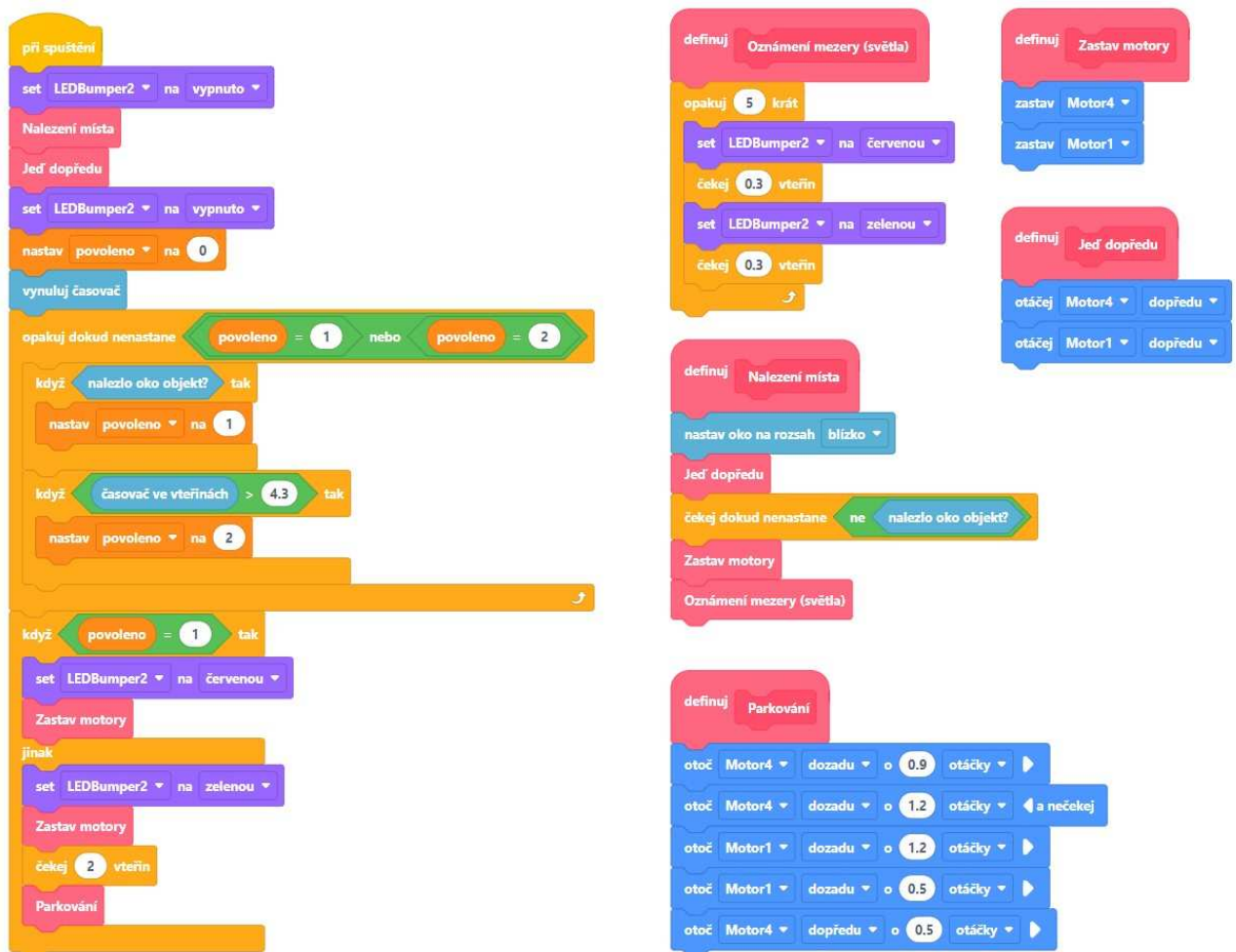
Robot začne u překážky se senzorem ve vzdálenosti maximálně sedmi centimetrů (na obrázku níže část se zelenou fajfkou). Následně jede rovně, dokud nezaznamená volné místo, respektive dokud senzor stále vnímá překážku. Tato část chování je totožná jako u stavebnice LEGO. Avšak v případě, že VEX GO pojede ve stanoveném čase rovně a překážka se bude nacházet ve vzdálenosti větší než sedm centimetrů, oční senzor ji již nezaznamená a robot oznámí, že je možné zaparkovat. Naopak pokud se bude nacházet překážka v pomyslné vodorovné čáře s první překážkou a bude blíže než sedm centimetrů, robot tak správně zareaguje a oznámí, že není možné zaparkovat.



Obrázek 119: *Problematika očního senzoru*

Poslední úsek kapitoly, úloha 10.7 automatického parkování, je ve své podstatě nejlehčí částí, jelikož ke kódu, který žáci vytvořili v předchozích úlohách, přidají pouze parkovací pohyby robota. Pro tuto část využívají pevně nastavených pohybů, které se naučili již v první kapitole učebnice a zúročí tak naučené znalosti. Robot po identifikaci volného prostoru zajede na dané volné místo nebo případně oznámí nemožnost zaparkování. Toho lze dosáhnout pomocí podmínky „když, jinak“.

Ve verzi VEXcode GO pro zaparkování robota byly opět využity vlastní bloky. Kompletní program nejen pro úlohu 10.7, ale pro celou kapitolu je vyobrazen zde:



Obrázek 120: Řešení úlohy 10.7 a zároveň řešení kapitoly 10

5.10.3 Shrnutí kapitoly 10

Před aktualizací programovacího prostředí byla tato kapitola zpracována s očním senzorem v režimu „blízko“. Ovšem to přinášelo mnohá úskalí při funkčnosti robota, jelikož vzdálenost vnímání předmětů byla velice malá. Nakonec k vyhotovení této kapitoly byl zvolen rozsah senzoru „daleko“.

V desáté kapitole byla potvrzena nezmíněná domněnka, že pojízdný robot stavebnice VEX GO neumí udržovat rovnou jízdu vpřed. Po daném úseku se trajektorie robota stáčí na jednu stranu. Opakovaně se stávalo, že robot do druhé překážky naboural, či odjížděl na druhou stranu mnohem dál od objektu. Tento fakt byl taktéž dohledán na oficiálním fóru vexrobotics, kde je v jednom případě

zmíněno, že daná stavebnice, se kterou byla testována praktická část, není jedinou stavebnicí, která se potýká s tímto problémem.

5.11 Kapitola 11: Hra

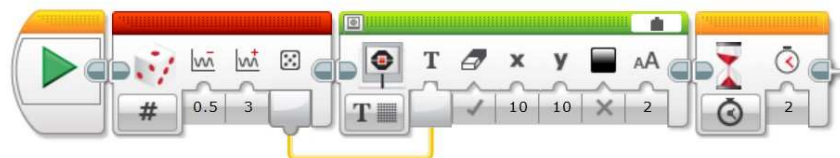
Kapitola jedenáct je poslední kapitolou původní učebnice. Žáci se zde oprostí od jezdících robotů, závor atd. K této kapitole je potřebná pouze řídicí jednotka s připojenými tlačítky (senzory dotyku).

Žáci v této kapitole programují a testují hru, která je vede nejen k prostému naprogramování, ale také k jejímu ladění, upravování a rozšiřování. V první fázi žáci naprogramují základ pro rychlostní postřehovou hru. Hra má jednoduchý princip – cílem je co nejrychleji stisknout tlačítko po zobrazení daného signálu. Žáci dále upravují hru například tak, aby se nezaznamenalo stisknutí tlačítka, pokud by bylo stlačeno dříve, než se zobrazí signál. V neposlední řadě též, jak efektivně ukazovat výsledek s případným uložením nejlepšího času.

Dalším z cílů je, aby se žáci naučili pracovat s blokem pro generování náhodného čísla a s časovačem. Kapitola je celkově klasifikována jako náročnější, jelikož vede k používání proměnných ve větší míře než předchozí kapitoly. V této kapitole je vhodné nechat žáky rozvinout svou fantazii a jejich nápady na vylepšení hry.

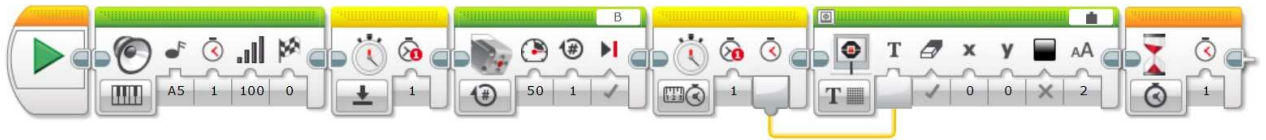
5.11.1 Hra – původní učebnice

První úloha 11.1 poslední kapitoly je rozdělena do tří záložek. První záložka, úkol 11.1.1, obsahuje kód, který si mají žáci přepsat do programovacího prostředí a několikrát tento program spustit. Následně mají odpovědět na otázku, co program po spuštění dělá, a tím zjistit, k čemu slouží blok „random“. Tímto způsobem se mají seznámit s blokem, který doposud nepoužívali.



Obrázek 121: Kód k úkolu 11.1.1

Ve druhé záložce, v úkolu 11.1.2, se nachází další kód, pomocí kterého mají žáci zjistit, jaké hodnoty se po spuštění vypisují na displeji robota. Kromě toho je nutné v tomto úkolu také připojit střední motor k řídicí jednotce.

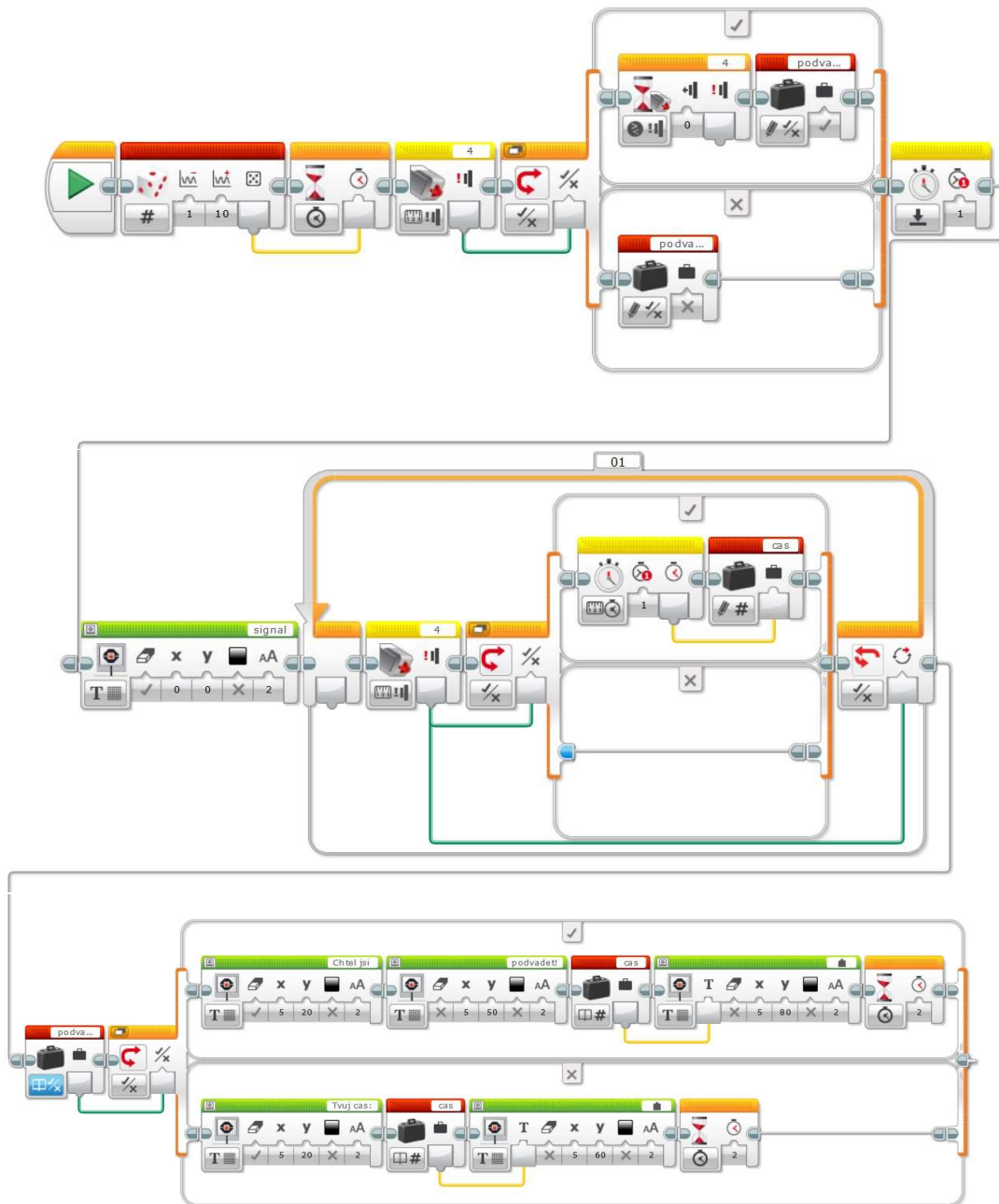


Obrázek 122: Kód k úkolu 11.1.2

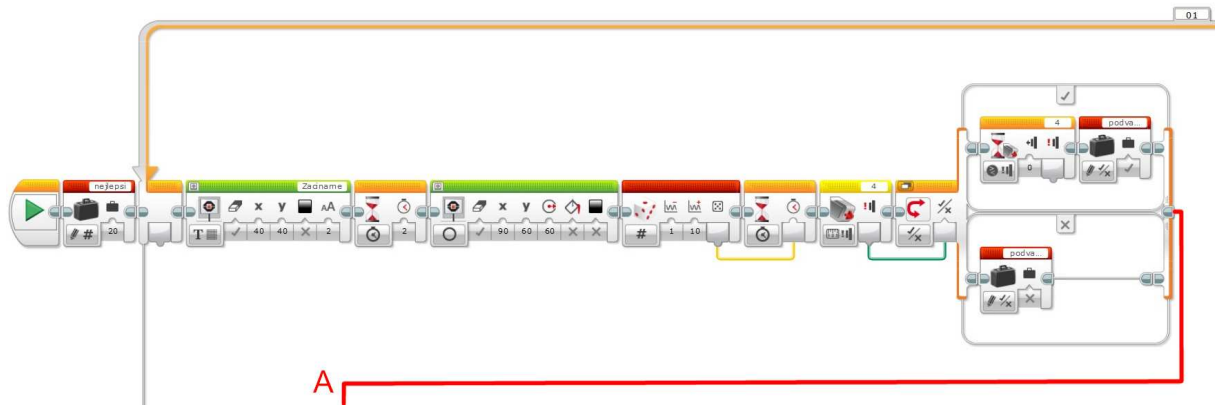
Třetí úkol 11.1.3 navazuje na předchozí kód, který je nad tímto textem. V této záložce je zadán úkol, aby žáci odstranili první žlutý blok se symbolem stopky (v kódu třetí blok zleva). Následně mají odpovědět na otázku, jak se změnila výsledná hodnota, která se vypisuje na displej. Jelikož je kód téměř totožný, vyjma jednoho chybějícího bloku, není třeba jej znovu vyobrazovat.

Úloha 11.2 nese název „Základní zadání hry Postřeh“ a jak je již podle názvu patrné, nachází se zde samotné zadání pro onu hru, která je cílem této kapitoly. Je nutné z předchozí úlohy odpojit motor a připojit tlačítka k řídicí jednotce. Popis principu hry, jež má připomínat zadání zní: hra funguje na principu, kdy se po náhodně dlouhé době zobrazí signál (v případě LEGO Mindstorms jde o text či obrázek na displeji mozku), po němž má být zmáčknuto tlačítko. Následně se zobrazí čas, za jak dlouho bylo tlačítko stisknuto. Žáci musí při programování počítat s tím, jak se hra zachová, když bude tlačítko stisknuto ještě před zobrazením signálu. Kód pro tuto úlohu je zobrazen na další straně na obrázku č. 123.

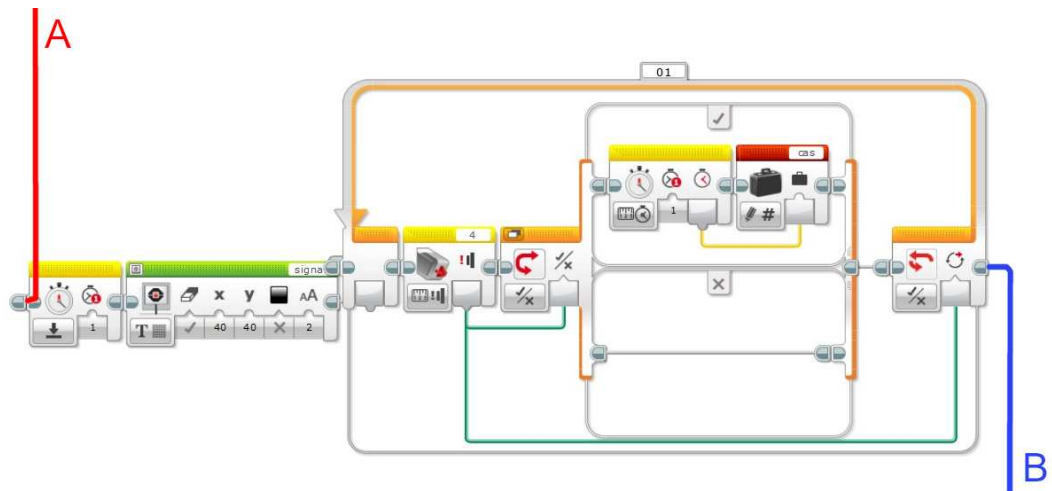
V úloze 11.3 je rozšíření základní hry z předchozí úlohy. Žáci mají program doplnit o zaznamenávání nejrychlejšího času. Následně by se po každé hře měl zobrazit aktuální čas a čas nejrychlejšího hráče. Vzhledem k rozsáhlosti programu jsou jednotlivé úseky kódu rozčleněny na dalších stranách, na obrázcích číslo 124-129. Pro lepší orientaci v kódu byly na sebe navazující úseky barevně a abecedně označeny.



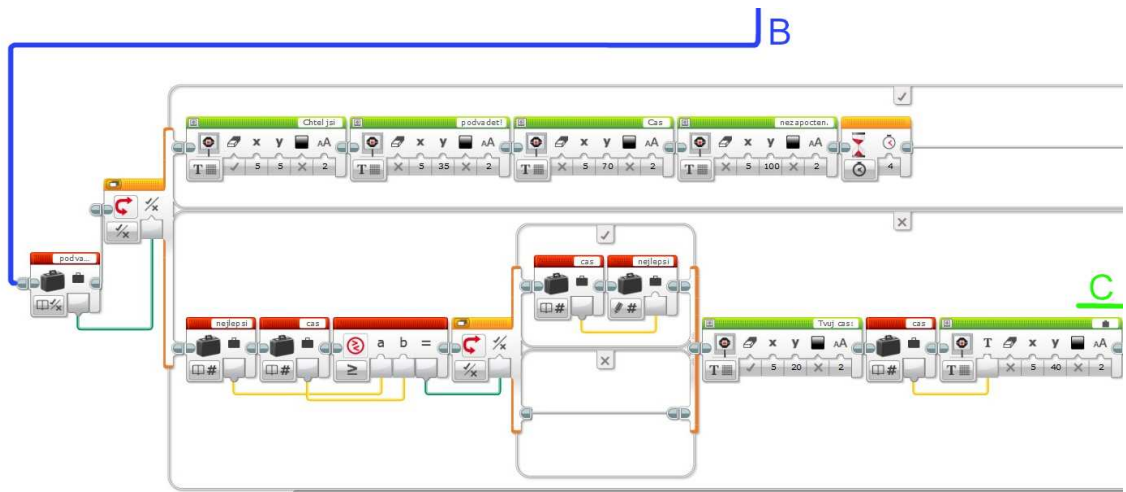
Obrázek 123: Řešení úlohy 11.2



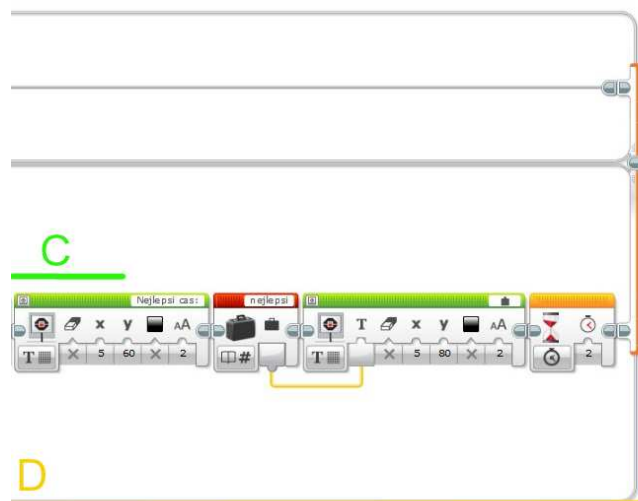
Obrázek 124: Řešení úlohy 11.3 – část A



Obrázek 125: Řešení úlohy 11.3 – část B



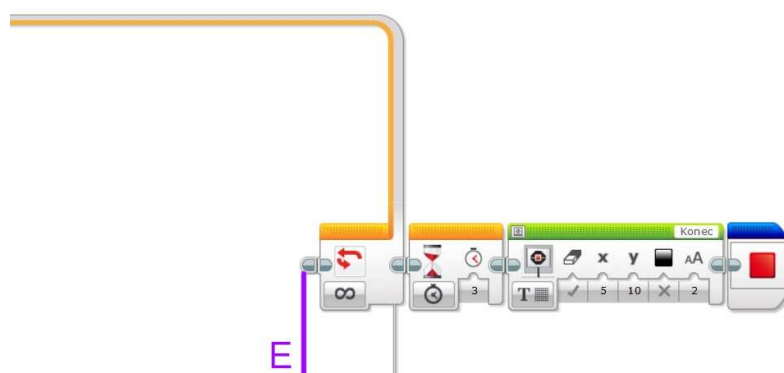
Obrázek 126: Řešení úlohy 11.3 – část C



Obrázek 127: Řešení úlohy 11.3 – část D



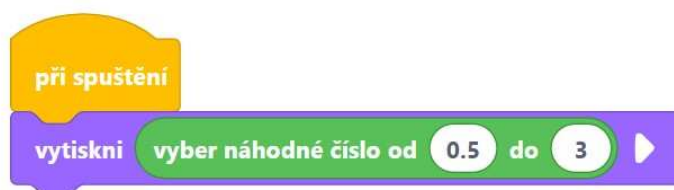
Obrázek 128: Řešení úlohy 11.3 – část E



Obrázek 129: Řešení úlohy 11.3 – část F

5.11.2 Hra – stavebnice VEX

V prvním úkolu 11.1.1 se žáci seznámí s novou funkcí v programovacím prostředí VEXcode GO. Jedná se o generování náhodného čísla v daném rozsahu. Zatímco v původní aplikaci a vyobrazeném kódu z původní učebnice je generování čísel vyobrazeno jako blok, ve VEXcode se jedná o operátor, který se vkládá do bloků či jiných operátorů. K výpisu náhodného čísla byla opět využita obrazovka zařízení, ze kterého žák programuje.



Obrázek 130: Řešení úlohy 11.1.1

Ve druhé záložce 11.1.2 mají žáci v první řadě za úkol připojit motor k řídicí jednotce a následně zjistit, co se vypisuje na obrazovce zařízení (ve VEXcode GO) po spuštění následujícího kódu.



Obrázek 131: Řešení úlohy 11.1.2

Žáci mají v rámci předchozí úlohy s kódem výše experimentovat a zkusit měnit parametry rychlosti a otáček motoru. Podobně je tomu ve třetí záložce 11.1.3, ve které mají žáci odstranit blok „vynuluj časovač“ z programu a vysvětlit, proč se změnila výsledná hodnota. Není nutné dokládat tento kód, protože je téměř totožný s kódem na obrázku č. 131, vyjma odstraněného čtvrtého bloku.

Úloha 11.2 je zadáním samotného cíle celé kapitoly, tedy zadáním pro sestavení kódu rychlostní postřehové hry. Po připojení tlačítka k řídicí jednotce mají žáci při vlastní tvorbě kódu předem zohlednit, jak zabránit započítání pokusu, v případě kdy bylo tlačítko stisknuto ještě před daným signálem (u stavebnice VEX GO byl signál zvolen změnou barvy LED nárazníku).

Autoři původní učebnice vyobrazují připojená dvě tlačítka k řídicí jednotce, avšak dále neupřesňují, jak a proč pracovat se dvěma tlačítky (s výjimkou návrhu rozšiřující úlohy pro dva hráče). Následující úlohy pro stavebnici VEX GO využívají pouze jedno tlačítko, a to i proto, že se v sadě této stavebnice nachází pouze jedno tlačítko (respektive LED nárazník, se kterým se dá pracovat jako s tlačítkem).

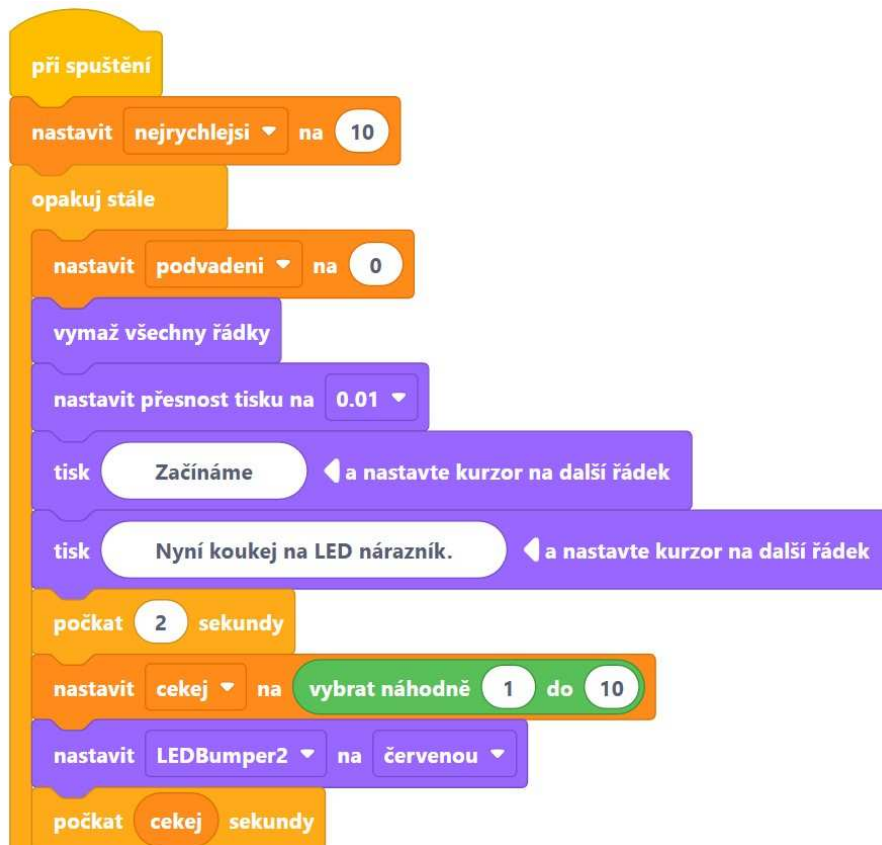
Žáci v úloze 11.2 vytváří kód, který definuje, že na začátku se má vydat signál, aby hráč věděl, kdy může tlačítko stisknout. Ve stavebnici VEX GO byla zvolena varianta změny barvy LED nárazníku z červené na zelenou, protože zde není přítomen ani reproduktor, ani displej. Počet vteřin, za jak dlouhou dobu bylo tlačítko stisknuto, se vypíše na obrazovku zařízení, ze kterého uživatel stavebnici programuje, a využije k tomu prostředí VEXcode GO. Výsledné číslo by mělo být zaokrouhleno na setiny, což umožňuje fialový blok „nastav přesnost tisku“.

Výše popsaný program, zobrazený na obrázku č. 132, obsahuje více proměnných. Prvním poznatkem je možnost vynechání proměnné „cekej“ a vložit zelený parametr „vyber náhodné číslo“ přímo do bloku „čekej“. Kód dále obsahuje blok „čekej dokud nenastane nestisknuto“, který zajišťuje, aby LED nárazník fungoval jako tlačítko, tedy aby byl stisknutý a následně uvolněný. Zároveň je tím ošetřena podmínka, že kdyby uživatel držel tlačítko ještě před daným signálem, program se neukončí a čas se nevypíše.

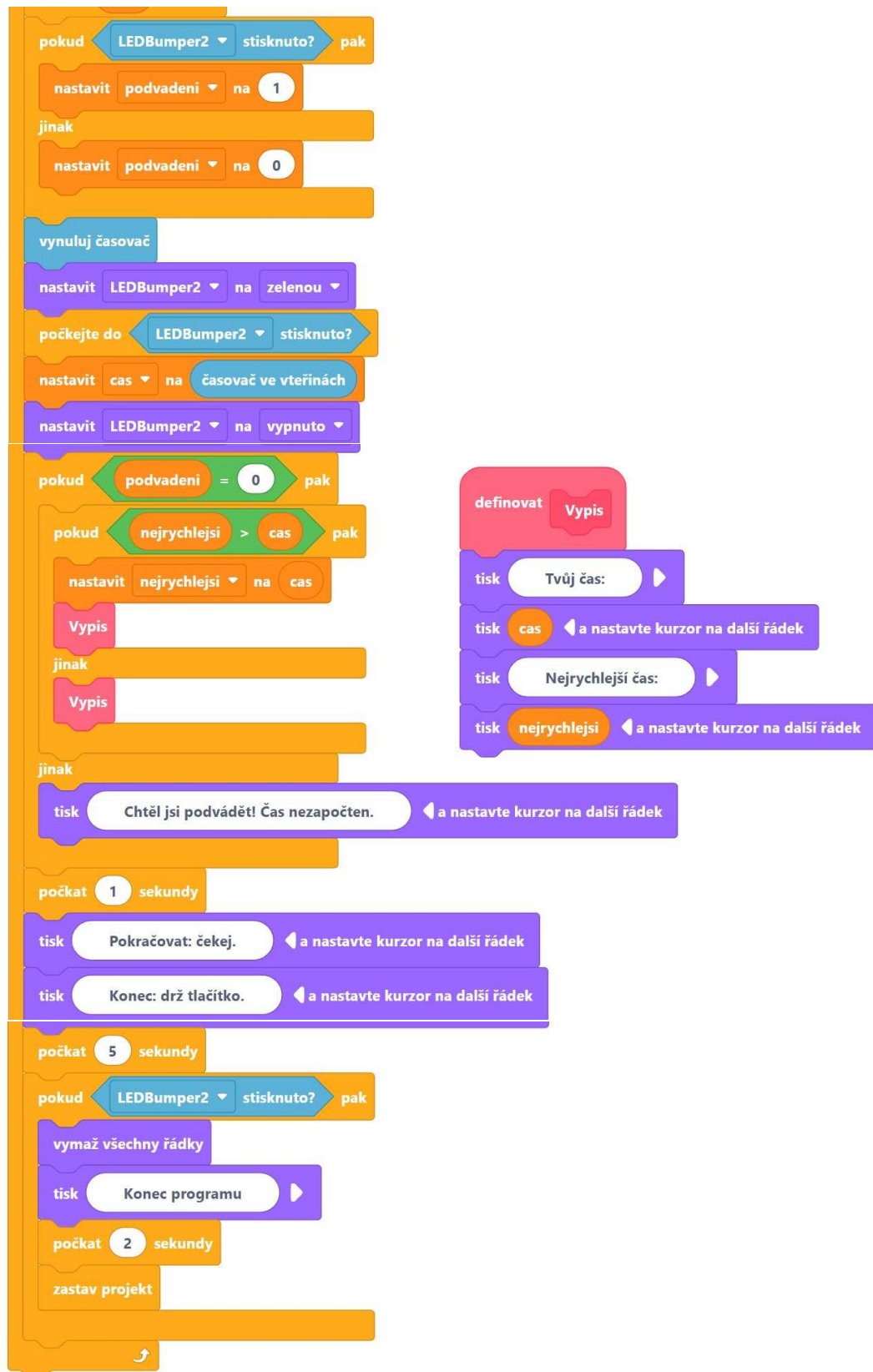
V úloze 11.3, rozšíření o nejlepší čas, mají žáci za úkol upravit předchozí program tak, aby zaznamenával nejlepší čas a zároveň po každé hře vypsal i čas aktuální hry a čas nejlepšího hráče. V tomto kódu byla pro zajištění detekce předčasně stisknutého tlačítka vytvořena proměnná s funkcí výpisu zprávy, že hráč chtěl podvádět.



Obrázek 132: Řešení úlohy 11.2



Obrázek 133: Řešení úlohy 11.3 první část (pokračuje na obrázku č. 134)



Obrázek 134: Řešení úlohy 11.3 druhá část