

Univerzita Karlova  
Pedagogická fakulta

BAKALÁŘSKÁ PRÁCE

2022

Oliver Erbes

Univerzita Karlova

Pedagogická fakulta

Katedra informačních technologií a technické výchovy

## BAKALÁŘSKÁ PRÁCE

Edukační robotika při volnočasových aktivitách a v zájmových útvarech

Educational Robotics in leisure activities

Oliver Erbes

Vedoucí práce: PhDr. Daniel Toháček

Studijní program: B7507 Specializace v pedagogice

Studijní obor: B IT-M (7507R040, 7504R015)

2022

Prohlašuji, že jsem bakalářskou práci na téma Edukační robotika při volnočasových aktivitách a v zájmových útvarech vypracoval pod vedením vedoucího práce samostatně za použití v práci uvedených pramenů a literatury. Dále prohlašuji, že tato práce nebyla využita k získání jiného nebo stejného titulu.

V Praze 17. 4. 2022.

.....

podpis

Rád bych poděkoval vedoucímu mé práce PhDr. Danielovi Tocháčkovi za podstatné rady, příjemnou spolupráci a předané znalosti k této práci. Dále bych chtěl poděkovat ZŠ Na Lukách a ZŠ Pomezí za poskytnutí prostor k vyučování. V neposlední řadě rodině a kamarádům za podporu a pochopení.

## **ABSTRAKT**

Bakalářská práce se zaměřuje na přípravu a organizaci aktivit v zařízení zajišťující zájmovou činnost a vzdělávání mládeže v zájmových útvarech. Cílem práce je vytvořit sadu úloh týkajících se robotické sady VEX, následné ověření žáky v zájmovém útvaru s analýzou jejich výstupů a na jejich základě tyto úlohy modifikovat, či upravit. V teoretické části směřujeme k vhodnému vymezení základních pojmů robotiky, edukační robotiky, základních pedagogických metod pro tento typ robotické výuky, základů algoritmizace, seznámení s jinými robotickými sadami a popisem vybrané stavebnice.

## **KLÍČOVÁ SLOVA**

Edukační robotika, robotika, VEX, sada úloh, vzdělávání

## **ABSTRACT**

This bachelor's thesis focuses on the preparation and organization of activities in facilities providing leisure activities and youth education in hobby departments. The aim of the work is to create a set of tasks related to the robotic set VEX, which will then be verified by students in leisure activities and subsequently analyse their output.. In the theoretical part we aim for a suitable definition of basic concepts of robotics, educational robotics, basic pedagogical methods for this type of robotic teaching, basics of algorithms, familiarization with other robotic sets and a description of the selected kit.

## **KEYWORDS**

Educational robotics, robotics, VEX, set of tasks, education

## Obsah

1	Robotika .....	9
1.1	Historie robotiky .....	11
1.2	Edukační robotika a konstruktivismus.....	11
1.3	Didaktické metody využité pro výuku.....	13
1.4	Základy algoritmizace.....	15
1.4.1	Základní konstrukce algoritmů.....	15
2	Školní robotické sady .....	17
2.1	Bee bot .....	17
2.2	LEGO.....	17
2.3	OZOBOT .....	18
2.4	VEX .....	18
2.5	Popis VEX 123 .....	19
2.6	Popis VEX GO.....	22
2.7	Popis VEX IQ .....	25
2.8	Popis jazyka ROBOTC .....	25
2.9	Nabídka Robot .....	27
2.10	Pole Text Function .....	28
2.10.1	Math.....	28
2.10.2	Motors.....	28
2.10.3	Sensors.....	30
2.10.4	Sound .....	33
3	Sady úloh pro VEX .....	34
3.1	1. stupeň.....	34
3.1.1	Programování pomocí dotyku .....	34

3.1.2	Programování pomocí karet.....	38
3.1.3	Sada úloh v prostředí VEX code .....	43
3.2	2. stupeň.....	51
3.2.1	Sada úloh v prostředí VEX code .....	51
3.2.2	VEX IQ.....	64
3.2.3	Sada úloh v prostředí VEX code .....	65
4	Analýza výstupů .....	70
4.1	1. Stupeň .....	70
4.1.1	Kvantitativní výzkum .....	73
4.2	2. stupeň.....	73

## Úvod

V posledních několika letech vstupují do života většiny lidí, ať už v průmyslu, nebo i v domácích zařízeních, různá robotická zařízení. Odraz této „revoluce“ se samozřejmě objevuje i v oblasti vzdělávání, které by mělo připravit dorůstající generaci na nové možnosti a novou realitu. Stejně jako se vyučuje například fungování spalovacích motorů, tak dnes by měly být žákům vysvětlovány a osvětlovány základní principy práce s robotickými zařízeními. Výuka robotiky se navíc vyskytuje v novém RVP pro základní a střední školy, kdy se žáci v tomto oboru mají vzdělávat již od prvního stupně ZŠ. Na řadě škol a dalších zařízeních se navíc vyskytují různé volnočasové útvary, které mají za úkol představit tuto disciplínu zájemcům z řad dětí a mládeže. Vzdělávání dětí se věnuje relativně nový obor zvaný edukační robotika, který je tématem této bakalářské práce.

V současné době se objevuje celá řada různých robotů, robotických hraček a robotických vzdělávacích sad určených pro vzdělávání dětí a mládeže ve věku od 4 let do dospělosti. Úplnou novinkou v Čechách je nasazení výrobků kanadské firmy VEX Robotics, jejíž některé modely byly poprvé distribuovány v roce 2021.

Práce je tvořena teoretickou a praktickou částí. Cílem teoretické části je vysvětlení základních pojmů robotiky, edukační robotiky, pedagogických metod, základů algoritmizace a seznámení s robotickými stavebnicemi a jejich popis. Praktická část má následující cíle: sestavení sady úloh separátně pro 1. a 2. stupeň, realizace úloh s dětmi, provedení analýzy těchto aktivit a stanovení případných úprav.

Tato bakalářská práce je zaměřena na tvorbu ucelené sady úkolů týkajících se práce s výukovými robotickými sadami, konkrétně se třemi modely jmenované firmy, a to VEX 123, VEX GO a VEX IQ, jejich ověření se skupinami dětí a následnou analýzou tohoto procesu.

# 1 Robotika

Robotika v současné době prožívá bouřlivé období svého rozvoje. S jejími aplikacemi se můžeme setkat nejen v průmyslu, ale i v běžném životě. Nemalou měrou se tento obor objevuje také na středních, základních a dokonce i v mateřských školách.

Robotiku jako takovou lze rozdělit na dvě základní kategorie, jimiž jsou průmyslová a experimentální robotika. Průmyslovou využijeme při navrhování průmyslových robotů, k automatizaci výroby a tedy nahrazení práce člověka těmito roboty. Příkladem mohou být robotická ramena v automobilech a jiných pásových pracích. Tento typ robotů se vyznačuje následujícími vlastnostmi: manipulační schopnost – pomocí manipulačních ramen dovede zvedat, přenášet požadované předměty, které může upravit pomocí montážních úkonů. Automatická činnost a snadná změna programu – úkoly jsou plněny v postupných krocích automaticky, bez zásahu člověka, a zároveň, pokud je potřeba, se tak úkol robota může změnit a přepsáním programu robota změnit jeho funkci. Dále univerzálnost a zpětná vazba. Experimenty, ověřování principů a disciplínami jako kybernetika a umělá inteligence se zabývá experimentální robotika. První experimenty provedli v 60. letech vědci z USA na MIT a dalších univerzitách, kde se s experimenty později připojila Velká Británie a Japonsko.

Isaac Asimov robotiku<sup>1</sup> definuje jako „*disciplínu o vytváření inteligentních strojů integrující několik vědeckých a inženýrských oblastí*“. Exaktnější definicí je například podle Bradyho, robotika jako propojení mezi vnímáním a činností, nebo jako podle McKerrowa, který ji uchopil jako souhrn několika činností - navržení samotného robota a jeho programování, využití sestaveného funkčního robota k řešení problému, či různých úloh. Analyzování algoritmů, řídicích procesů a senzorů napříč lidmi, zvířaty a stroji, pomocí nichž dospěje k návrhu a využití robota. Zjednodušeně se jedná o obor, který se zajímá o stavbu robotů a o jejich studium. Propojuje mechaniku, elektrotechniku, umělou inteligenci a celou řadu dalších disciplín.

Co je to robot? Inteligentní stroj s částečnou samostatností, který dokáže splnit zadané úlohy s jistou interakcí s okolním světem. V tom případě se dá mluvit o autonomních nebo

---

<sup>1</sup> TOCHÁČEK, Daniel a Jakub LAPEŠ. *Edukační robotika*. s. 7.

také kognitivních robotech. Robot dokáže vnímat různé jevy z okolí a rovněž s nimi interagovat. K vnímání svého okolí využívá senzory, jako jsou optické snímače, teplotní čidla atd.

Slovo robot vymyslel malíř a spisovatel Josef Čapek, ve svém díle R. U. R. (Rossumovi univerzální roboti) jej použil jeho bratr Karel Čapek. Tímto se české slovo rozšířilo po celém světě. Definicí<sup>2</sup> podle Mezinárodní organizace pro standardizaci v normě ISO 8373 je robot definován jako „*automaticky řízený, opětovně programovatelný, víceúčelový manipulátor pro činnost ve třech nebo více osách, který může být buď pevně upevněn na místě, nebo mobilní k užití v průmyslových automatických aplikacích*“.

Jak uvedl Isaac Asimov v knize I Robot (1950) jsou Tři zákony robotiky<sup>3</sup>, které robot nesmí porušit, a pravidla, kterými se musí řídit. Jinak známé také jako Asimovy zákony.

1. Robot nesmí ublížit člověku nebo svou nečinností dopustit, aby člověku bylo ublíženo.
2. Robot musí uposlechnout příkazů člověka, kromě případů, když jsou tyto příkazy v rozporu s prvním zákonem.
3. Robot musí chránit sám sebe před zničením, kromě případů, kdy je tato ochrana v rozporu s prvním nebo druhým zákonem.

Roboty dělíme podle několika kategorií, podle generace, schopnosti se přemísťovat a také účelu, vzhledu a schopností. Roboti 1. generace fungují na pevně daném programu, oproti tomu 2. generace je vybavena senzory a disponuje možností reagovat na okolí. Podle možnosti přemístění dělíme na stacionární a mobilní. A třetí dělení účelu, vzhledu atd. dělíme na manipulátor - stroj bez vlastní inteligence, který je ovládán na dálku. Kuchyňský robot – kombinace několika kuchyňských spotřebičů, jako propojená jednotka s nástavci. Android – robot, který je podobný člověku (dále droidy a humanoidy). Kyborg – kombinace robotických součástí a přírodního biologického těla. Prvním kyborgem je Angličan Neil Harbisson, který se narodil se vzácnou formou barvosleposti. V hlavě má

---

<sup>2</sup> Kdo vymyslel slovo robot? Karel Čapek to nebyl!. *Factory Automation* [online]. Praha: FANUC Czech, 2014 [cit. 2022-03-16]. Dostupné z: <https://factoryautomation.cz/kdo-vymyslel-slovo-robot-karel-capek-to-nebyl/>

<sup>3</sup> TOCHÁČEK, Daniel a Jakub LAPEŠ. *Edukační robotika*. s. 9.

voperovanou anténu, kterou dokáže slyšet barvy. Anténa zachycuje světelné spektrum, kde se signál přenáší formou vibrací přes kost, a tím tak dokáže slyšet jednotlivé barvy. Toto odvětví může mít v budoucnu v medicíně vysoký potenciál ke zlepšování své výkonnosti různých částí lidského těla, nebo jako lepší verze protéz.

## **1.1 Historie robotiky**

Už od pradávna se lidstvo setkávalo s mírnou představou robotiky, po úmorné fyzické práci, kdy si ve svých fantaziích mohli představovat stroje, které odvedou práci za ně. Jak se společnost rozvíjela, tak se tyto sny mohly stát skutečností. V 18. st. vznikla první napodobenina člověka tzv. android (vypadá a působí jako člověk). Vynálezci byli Pier a Henry Droz pocházející ze Švýcarska. Výtvorem byl písář, automatizovaný stroj k psaní perem i několik vět. Dále se přešlo na mechanické napodobeniny zvířat tzv. zooidy. Kde je v dnešní době známým čtyřnohým psím robotem Lumos od společnosti Boston dynamics, který se dostane kamkoliv, kde člověk, a tím ho lze v praxi na potenciálně nebezpečných místech nahradit. Ve 40. letech se objevují teleoperátory, které měly první praktické využití pro manipulaci s radioaktivními materiály. Prvním průmyslovým robotem se stal UNIMATE v r. 1961. V 80. letech jsou do průmyslových robotů instalovány čidla hmatu, počítačové vidění s dalšími prvky. Robot Jojourner se stal prvním robotem vysazeným na Marsu v roce 1997. Tímto způsobem se jeví, že je opravdu snaha nahradit člověka strojem. To vede k úctyhodným objevům, jako jsou robotické protézy pro osoby s omezenou pohyblivostí. Vývoj v nemocničním prostředí, novým možnostem léčby a přístrojům ke složitým operačním zákrokům.

## **1.2 Edukační robotika a konstruktivismus**

Jak už jsme si řekli, robotika má široké využití ve všech možných vědních oborech a je tedy možné ji využít i ve vzdělávání. Tím není myšleno jen vysokoškolské studium robotiky a podobných oborů využívající robotiky. Obecně ji lze využít po celém vzdělávacím školním systému od mateřské, základní až po střední školy. Nemusí se jednat o samotné vyučovací hodiny informatiky, ale také o další předměty. Téměř na všech webových stránkách výrobců edukačních robotických sad jsem se setkal s pojmem STEM. Jedná se o nový vzdělávací koncept, který je zkratkou Science (přírodní vědy), Technology (technika), Engineering (technologie) a Mathematics (matematika). Cílem je propojit tyto

předměty a částečně je vyučovat společně a najít vzájemné prolnutí. Opírá se o řešení zadaných problémů z kombinace předmětů z reálného prostředí. K samotné výuce se využijí robotické pomůcky, nebo celé robotické sady jako Lego WeDO, Mindstorms, nebo v našem případě sada VEX. S nimi přichází i všelijaké programovací jazyky vytvořené primárně pro děti. Základním jazykem byl jazyk LOGO, na kterém se podílel Seymour Papert s dalšími vědci. Mezi další jazyky patří např. Karel, Scratch, Kodu Game LaB. Základní myšlenkou práce s roboty je, že pro žáky při interakci s fyzickými objekty se součástkami, které při správném zapojení, sestavení a naprogramování přinesou ihned výsledky, a výuka se tedy pro ně stává zábavnější, pestřejší a tedy i uchopitelnější. Oproti práci s programy v nějakém prostředí na počítačích nemusí být tak zábavná pro žáka, který si neosvojil znalosti ze základní školy potřebné k vyřešení složitějších a zajímavějších úloh.

Nyní bych se zmínil o dvou vzdělávacích teoriích, kterými jsou **konstruktivismus** a **konstrukcionismus**. Zakladatelem konstruktivismu byl švýcarský filozof Jean Piaget, který se zároveň zajímal o vývojovou psychologii, kterou inspiroval i pedagogy ve vzdělávání. Seymour Papert, jakožto americký profesor matematiky a informatiky, navázal na teorii Jeana Piageta a přišel s konstrukcionismem, který z konstruktivismu vychází.

Základní myšlenkou konstruktivismu je nepředávat myšlenky a znalosti žákovi samovolně, ale sám žák je staven do pozice zkoumání a objevování daného problému a pochopení problematiky zevnitř, kde se pedagog stává spíše žakovým průvodcem dané problematiky a pomocnou rukou, žák ale musí znát kompletní koncept a mít znalosti v určitém odvětví. Žáci tedy konstruují své znalosti v závislosti na svých zkušenostech a informacích, které nasbírali během svého života. Tuto teorii je možné využít právě v této oblasti edukační robotiky, kde většinou žáci čelí praktickým problémům, které v tomto případě představují samotní roboti s úkoly. Právě při tomto procesu žáci konstruují dle svých zkušeností a rozvíjí další.

Podle Jeana Piageta a jeho teorie konstruktivismu<sup>4</sup> „není správné vzdělávání chápat jako proces přenosu informací od učitele k žákovi nebo studentovi, ale jako proces aktivního

---

<sup>4</sup> TOCHÁČEK, Daniel a Jakub LAPEŠ. *Edukační robotika*. s. 22.

*vlastního konstruování znalostí založený na zkušenostech daných skutečným životem, které se nabalují na dříve získané znalosti a zkušenost“.*

Podle Seymoura Paperta teorie konstrukcionismu<sup>5</sup> „*doplňuje předchozí teze o tvrzení, že nejefektivnější metodou konstruování nových znalostí je zapojení žáka či studenta do takových činností, při nichž vytváří určitý konkrétní a pro něj osobně zajímavý produkt. Jde tedy o jisté rozšíření konstruktivismu zdůrazňující praktické řešení problémů s nasazením práce vlastní hlavy a rukou. Studující budují hmatatelné objekty ve vhodném konstruktivním prostředí. Úkolem učitele je vytvořit takové prostředí, v němž žáci nebo studenti tvoří smysluplné věci a prostřednictvím této práce se učí mnohem více než tradičním způsobem“.*

V 80. letech se objevuje technologie<sup>6</sup> LEGO/Logo, jedná se o první robotickou stavebnici, která byla dostupná širokému okolí. Kombinuje známou stavebnici LEGO k samotné konstrukci robotů a programovacího jazyka Logo. Do této sady přibyly součástky jako ozubená kola, motory a senzory. Jazyk logo byl především spojen s výukou programování dětí. Tento jazyk je známý želví grafikou, kde se po pláži pohybuje želva, která za sebou nechává stopu, pokud má ocas položený na zemi, ten může zvedat, a tím se stopou přestat. Tato sada Lego/Logo přesunula želvy z monitoru do praktického světa, konstrukcí želv pomocí stavebnice LEGO.

### **1.3 Didaktické metody využité pro výuku**

V této kapitole si přiblížíme pedagogické pojmy a metody, které využíváme ve výuce s robotickými sadami. Didaktika je součástí pedagogiky, zabývá se teorií vyučování. Pochází z řeckého didasko, jehož význam je „vyučuji“. Skalková didaktiku vymezuje jako „*teorii vzdělávání a vyučování. Zabývá se problematikou obsahů, které se jakožto výsledky společensko-historické zkušenosti lidstva stávají v procesu vyučování individuálním majetkem žáků. Zabývá se zároveň procesem, který charakterizuje činnosti učitele a žáků a v němž si žáci tento obsah osvojují, tedy vyučováním a učením“.* Pojem didaktika se

---

<sup>5</sup> TOCHÁČEK, Daniel a Jakub LAPEŠ. *Edukační robotika*. s. 22.

<sup>6</sup> ALIMISIS, Dimitris, ed. *Teacher Education on Robotics-Enhanced Constructivist Pedagogical Methods*. Athény: New Technologies Publications, 2009. ISBN 978-960-6749-49-0.

v některých proudech pedagogického myšlení téměř nepoužíval. Z počátku 20. století vznikla spousta návrhů, nových modelů a koncepcí s didaktickým charakterem (daltonský plán, projektové učení, nebo skupinové učení). Tyto aspekty rozvíjely podstatným způsobem problematiku učení a vyučování.<sup>7</sup>

Tímto se dostáváme k progresivní výchově, jejímž autorem je John Dewey. Jeho filosofie byla pragmatického rázu, základní myšlenkou a pojmem je životní zkušenost a učení prací (learning by doing). Z této koncepce vychází následující metody - problémové a projektové vyučování, které je pro práci s robotickými sadami mezi žáky zřejmě vhodné. Zároveň můžeme sledovat podobnost s konstruktivistickým pojetím výuky. Cílem **problémového vyučování** není podávat hotové poznatky, ale podat učivo, aby ho nutilo k zamyšlení, uvažování a vytváření vlastních závěrů. Problém stojí mimo žáka a má objektivní ráz. Pro **projektové vyučování** je typická možnost volby tématu žákem, a tedy jistý vliv na zainteresování do projektu. Učení vychází z prožitků bez vnější motivace žáků, problémy řeší z aktuálního podnětu.<sup>8</sup>

Další využitou metodou, nebo také organizační formou výuky je **kooperativní výuka**. Ta může být zaměňována s výukou skupinovou. Rozdíl nastává v prostředí skupinky, pokud by žáci řešili nějaký úkol a tím si opakovali látku, šlo by o výuku skupinovou. Jestliže se učitel soustředí na spolupráci ve skupině a získání nových poznatků, jednalo by se o formu kooperativní. Pokud se budeme bavit o kooperativním vyučování, základním prvkem je komunikace tváří v tvář, kdy je třída rozdělena do skupin (3 až 5 žáků), aby spolu mohli žáci komunikovat. Dalším vztahem je kooperace, aby uspěl jeden člen, musí uspět všichni. Úspěch jednotlivce se váže na úspěch zbylých členů. Dále si žáci obohatí sociální dovednosti prací ve skupině. Sledujeme, jakým způsobem jako jednotlivci přispívají skupině a co se žák naučil jako jedinec.

---

<sup>7</sup> SKALKOVÁ, Jarmila. *Obecná didaktika*. 2. vyd. Praha: Grada Publishing, 2007. ISBN 978-80-247-1821-7.

<sup>8</sup> PROKEŠOVÁ, Miriam. *Obecná pedagogika*. Ostrava: Ostravská univerzita, Pedagogická fakulta, 2018. ISBN 978-80-7599-056-3.

## 1.4 Základy algoritmizace

V publikacích zabývajících se algoritmizací pojem algoritmus není zaveden a předpokládá se, že čtenář ví, o co se jedná, a rozumí této problematice. V matematice existují základní matematické elementární pojmy, jako jsou bod, množina atd. a mezi ně patří i algoritmus. Nelze jej přímo definovat a musíme se přiklonit k opisu. Příkladem algoritmu ze života může být příprava ráno do školy, návod k sestavení nábytku atd. K vizualizaci algoritmu využijeme vývojových diagramů.

Algoritmus<sup>9</sup> je návod, jak provést určitou činnost. Ovšem ne každý návod představuje algoritmus. Algoritmem označíme návod splňující jednotlivé vlastnosti:

1. Je elementární. Skládá se z konečného počtu činností, které označíme jako kroky.
2. Je determinovaný. Po každém kroku lze určit, zda proces skončil, a pokud neskončil, kterým krokem má pokračovat.
3. Je konečný. Počet opakování jednotlivých kroků je vždy konečný. Algoritmus končí po konečném počtu kroků.
4. Je rezultativní. Vede ke správnému řešení.
5. Je hromadný. Algoritmus lze využít k řešení skupiny podobných úloh.

**Metoda shora dolů** je způsob zápisu algoritmu, pokud známe řešení nějaké třídy problémů, algoritmus můžeme sepsat. Při této metodě postupujeme tak, že řešení rozkládáme na jednodušší operace, až se dostaneme k elementárním krokům. Další metodou je zdola nahoru, kde se vychází z elementárních kroků k procesům, které vyřeší tížený problém.

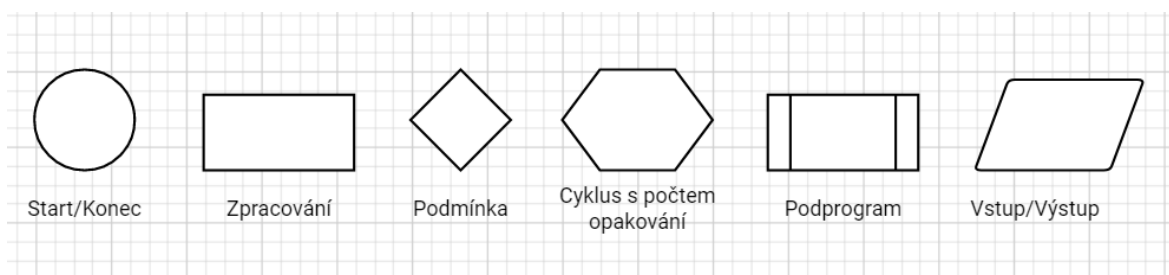
### 1.4.1 Základní konstrukce algoritmů

Každý algoritmus by měl mít začátek a konec a tři základní konstrukce, kterými jsou posloupnost příkazů, cyklus (iterace) a podmíněnou operaci (selekcí). Teoretickou konstrukcí jsou také vstupy a výstupy, které se načtou a jsou potřebné k programu, nebo jako zobrazení výstupů programu. Posloupnost příkazů je tvořena jednotlivými kroky

---

<sup>9</sup> ZNÁM, Štefan. *Pohľad do dejín matematiky: celoštátna vysokoškolská príručka*. Bratislava: Alfa, 1986. Edícia matematicko-fyzikálnej literatúry.

programu tzv. zpracování (sečtení dvou čísel, jízda dopředu). Cykly jsou složeny z těla cyklu, do kterého jsou vloženy jednotlivé příkazy a podmínky pro opakování cyklu. Setkáváme se se třemi typy s podmínkou na začátku, na konci a s počtem opakování. Podmíněné operace využijeme k selekci, větvení programu. Pokud je splněna některá podmínka, můžeme si v tom případě zvolit, jaká část programu by se měla provést.



*Obr. 1 - Základní prvky vývojových diagramů*

## **2 Školní robotické sady**

### **2.1 Bee bot**

Tento programovatelný robot je vhodný pro výuku základů programování pro děti již od 3 let. Tato možná pomůcka ve výuce může rozvíjet logické myšlení, prostorovou představivost a základní matematické dovednosti. Využití robota se tedy nabízí již v mateřských i základních školách. Součástí a potřebnou pomůckou je podložka se sítí čtverců. Tímto robotem je možné děti seznámit se základy algoritmizace a mohou si vymyslet své algoritmy či programy. Možnost pohybu včelky je ve čtyřech směrech (doleva, doprava, dozadu a rovně) pomocí tlačítek, která jsou umístěna nahoře na zádech robota. Dalšími tlačítky jsou stop, tlačítko pro smazání programu a tlačítko OK pro jeho spuštění. Sekvencí jednotlivých směrových tlačítek lze vytvořit program, který má maximální kapacitu až 200 kroků. Otočka na obě strany je do pravého úhlu o 90 stupňů. Robot je napájen baterií, kterou je možné nabíjet pomocí USB kabelu. Pokud je baterie nabita, včelka vydrží až 6 hodin standardní práce, jestliže není robot po dvě minuty používán, automaticky se přepne do úsporného režimu. Ten se deaktivuje zmáčknutím jakéhokoli tlačítka. Bee-Bot je vybaven těmito senzory: infračervený senzor pro komunikaci s dalším Bee-Botem, mikrofón, reproduktor, LED diody. Tento robot najde uplatnění v předmětech STEM, kterými jsou přírodní vědy, technologie, technika a matematika.

### **2.2 LEGO**

Lego je mezi robotickými sadami tou nejznámější s jistou kvalitou. Mezi nejznámější sady patří LEGO Mindstorms, které prošlo několika generacemi postupně RCX, NXT a nejnovější verzi EV3. Stavebnice jsou podobné, ale s postupem času a vývojem technologií přichází nové prvky a také senzory. Verze NXT obsahovala senzor světla, dotyku, zvuku a ultrazvukový senzor vzdálenosti. K tomu motory s řídicí kostkou. K verzi EV3 přibyl IR přijímač (senzor vzdálenosti) a gyroskop.

Sada LEGO Education WeDO 2.0 byla navržena, jak už název napovídá, pro edukaci jako nástroj do výuky pro pedagogy. Tuto stavebnici je možné využít na prvním stupni základních škol, kde si žáci postaví svého robota, s kterým řeší všelijaké problémy

a vytváří projekty. Cílem je rozvíjet žákovy znalosti pomocí teorie STEM k rozvinutí zvědavosti žáka v rozličných oborech.

Na stránkách výrobce prezentuje mindstorms jako 5 osobitých robotů, kteří mají různé vlastnosti a jiný účel. Programuje se přes Mindstorms aplikaci, která využívá dvou jazyků. Buď blokový Scratch, nebo textový Python. Roboti jsou dálkově ovládáni pomocí Bluetooth. Tato nová verze obsahuje dobíjecí HUB vybavený LED displejem a 6 input/output porty, výkonné motory, senzory barev a vzdálenosti.

### **2.3 OZOBOT**

Ozobot je malý robot o velikosti pingpongového míčku, který slouží k seznámení s programováním a robotikou. Robota lze využít ve výuce pro žáky starší 8 let. Existují dvě verze Ozobot bit a Ozobot evo, kde verze evo je dražší, komplexnější a inteligentnější verzí robota. Ozobot bit je vybaven 5 světelnými čidly, RGB LED diodou a motory pro pohyb. K programování lze využít 2 způsoby barevným kódováním, nebo pomocí OzoBlockly. Robot dokáže snímat povrch pod sebou a dokáže rozpoznat červenou, modrou, zelenou a černou barvu, pomocí kterých lze vytvářet trasy, po kterých robot pojedí. Sekvencí několika barev za sebou lze robota programovat např. otočení vlevo. Pro příklad pokud bych měl trasu černou barvou a na dráze by byl blok modrá, černá a opět modrá, tak robot zrychlí. Těchto příkazů je samozřejmě více. Druhou možností je OzoBlockly, což může připomínat prostředí Scratch, a ten je tedy na blokovém principu. Verze bit a evo jsou druhou a třetí generací původního ozobota, který Ozoblockly nepodporoval.

Nedokážu říct, zda další funkce, o které se budeme bavit, je výhodná, nebo zbytečná a zda měla by fungovat automaticky. Robot totiž může jezdit na displeji tabletu nebo na papíře či další bílé struktuře. Pokud bych robotovu podložku vyměnil, pokaždé musím provést kalibraci barev. Na papír, nebo tablet nakreslím černý bod, podle kterého ozobot provede kalibraci pro černou barvu, a následně ho přesunu na prázdné bílé místo pro barvu bílou. Po bílém probliknutí by měla být kalibrace úspěšně splněna.

### **2.4 VEX**

Sada robotů VEX je poměrně rozsáhlý systém robotů a robotických stavebnic.

Nejjednodušším robotem je VEX123, umožňující programování pomocí tlačítek, kartiček nebo blokové počítačové aplikace. Pro komunikaci s okolím robot používá bluetooth. Je určen pro nejmenší děti, ačkoliv s ním lze provádět i složitější algoritmy. Robot má senzor barev a vzdálenosti, gyroskop, reproduktor a programovací tlačítka.

Poněkud pokročilejším je robot s názvem VEX GO. Jedná se již o stavebnici, pomocí které můžeme variabilně stavět celou řadu modelů od autíčka na gumu, přes různé varianty programovatelného vozítka až po robotickou ruku. K počítači je připojen pomocí bluetooth. Co se senzorů týče, obsahuje stavebnice senzor vzdálenosti, barvy, gyroskop a senzor dotyku.

Kvalitnější variantou je VEX IQ, který se od VEX GO liší především masivnější a spolehlivější konstrukcí a větším počtem senzorů, motorů a tlačítek. K počítači je připojen pomocí rozhraní USB. Další změnou je možnost programování pomocí modifikovaného jazyka C s názvem RobotC.

VEX Robotics dodávají i další pokročilejší modely lišící se především kvalitou provedení a větší flexibilitou, jedná se kovové stavebnice EXP a V5. Oproti předchozím verzím se liší také cenou, kdy vybavení třídy je téměř 4 x dražší než sada VEX GO. Stejně jako model IQ je možno je programovat také pomocí jazyka RobotC či Pythonu.

## 2.5 Popis VEX 123

Robot 123 je programovatelný robot, který přináší předškolákům nebo žákům prvního stupně zkušenosti z informatiky a výpočetního myšlení se základy logiky. Robota je možné programovat a ovládat třemi způsoby. Za prvé pomocí dotyku, za druhé pomocí kodéru a kódovacích karet, které se do kodéru vkládají postupně za sebe a třetí možnost je programování pomocí softwarového prostředí VEXcode123. Toto



Obr. 2 - Robot VEX 123

prostředí vychází z prostředí Scratch, kde se jednotlivé příkazy vkládají pomocí blokových příkazů. Nedílnou součástí je hřiště, ve kterém robot může plnit různé úlohy. Pomocí

čtvercových dlaždic můžeme hřiště postavit do jakýchkoli tvarů, jak se nám zlíbí. V hřišti jsou čtverce, které mají délku strany stejnou, jako je délka robota.

Jak s robotem začít a jak ho zprovoznit? Po rozbalení robota je potřebné ho nabít. K nabíječce se připojí pomocí kabelu USB – C. Nabíjet robota můžeme i pomocí USB portu počítače. Po připojení a při samotném nabíjení se průhledné tlačítko s nápisem VEX rozblíká červeně. Pokud je robot zcela nabit, tak se tlačítko zbarví z červené na zelenou. Robot se zapne jednoduchým přejetím kolečky po povrchu.

Robot má na horní straně 5 tlačítek, kterými se dotykem programuje. Startovní tlačítko, které je uprostřed, které zároveň informuje o stavu nabití baterie, funguje při stlačení několika vteřin jako vypínací tlačítko robota, kde kontrolka zabliká žlutě a po cvaknutí se robot vypne. Robot se automaticky vypne sám, pokud je přibližně 5 minut neaktivní. Okolo startovního tlačítka jsou 4 další, kterými už lze robota programovat. Šipka dopředu jako jízda o jeden krok vpřed (o jednu délku robota, což je 7,5 cm), šipka vpravo slouží k rotaci robota doprava o 90 stupňů. Stejně šipka vlevo k rotaci o 90 stupňů vlevo. Posledním tlačítkem je obrázek „reproduktoru“, který slouží k zatroubení např. po splnění úkolu, nebo jako varianta počkej několik vteřin. Po zvolení námi potřebných příkazů stačí stlačit tlačítko uprostřed robota světelnou kontrolku a tím program spustit. Pokud byl robot naprogramován k nějakému úkolu a dále již příkazy nepotřebujeme, stačí s robotem zaklepat. Tím se příkazy vymažou z robotovy paměti.

Pro školní využití je potřeba aplikace Vex Classroom, která neslouží k programování robota, ale jako informační aplikace o stavu robota a jeho možném nastavení. Aplikaci je možné si stáhnout na Google Play nebo na App Store, a lze tedy využít pouze na mobilních zařízeních či tabletu. V aplikaci se dá robot lokalizovat tak, že vydá zvuk přejmenovat nebo updatovat firmware. Dále lze změnit jeho hlasitost a čas spánkového režimu, tedy vypnutí. K aplikaci lze připojit více robotů naráz.

Druhou možností jak robota ovládat je, jak již bylo zmíněno, pomocí kodéru a kódovacích karet. Kodér je napájen dvěma AAA bateriemi. Jednotlivé karty fungují jako příkazy, které vkládáme do kodéru, který má 11 slotů. Vždy prvním příkazem, kterým se projekt začíná, musí být bez výjimky „when start 123“. Karty jsou členěny dle funkce do kategorií. Celkově jich je 7 a dělí se na pohybové – řídí pohyb a rotaci robota, zvukové – ovládají

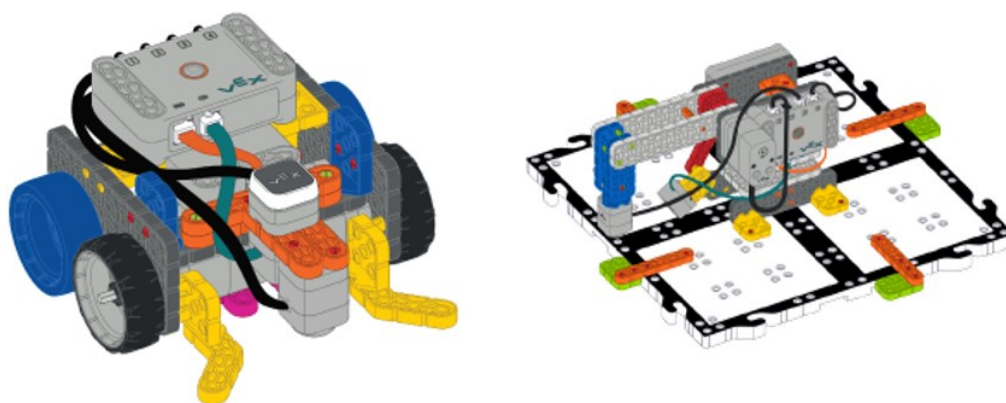
zvuky robota, vzhledové – ovlivní barvu kontrolky uprostřed robota, ovládací – mezi ně patří podmínky k rozhodování atd., události – pouští, opakují a stopují program, akce – změny robotovy emoce a čas – určují, po jakou dobu má robot čekat. Na horní straně kodéru jsou tři tlačítka - zleva zelené tlačítko play pro spuštění, oranžové pro spuštění programu po krocích a červené tlačítko stop pro vypnutí. Zároveň fungují ke spuštění a ukončení programu. Při zmáčknutí startovacího tlačítka se v pravém rohu rozsvítí zelená dioda, čímž je kódér spuštěn. Spárování robota s kódérem se provede podržením play a stop tlačítek na kodéru a tlačítek na robotovi pro otočení doprava a doleva po dobu delší 5 vteřin. Po propojení s kódérem robot zahraje melodii. V tuto chvíli se mohou jednotlivé karty s příkazy zasouvat do slotů (drážek) kodéru. Po zmáčknutí tlačítka play na kodéru se program zkompiluje, uploaduje a spustí.

Třetím a posledním způsobem jak robota programovat je v prostředí VEXcode123, které je postavené na blokovém schématu příkazů. Aplikace běží na webovém prohlížeči zdarma. Jinou variantou je si aplikaci stáhnout. Příkazy jsou seřazeny do kategorií stejně jako pro kódovací karty, ale přibylo 5 dalších kategorií a těmi jsou vnímání – detekce nehody, gyroskop, oční senzor atd., operátory – logické a aritmetické operace, proměnné – možnost vytvoření vlastní proměnné, moje bloky – možnost vytvoření funkce a v neposlední řadě komentáře - popis a poznámky k zapsanému kódu. Na rozdíl od kodéru se ztrácí problém s maximálním počtem příkazů, kterých bylo deset. Zde jsme v podstatě neomezeni.

Propojení robota s prostředím na počítači probíhá bezdrátově přes Bluetooth. Robota propojíme tak, že spustíme prostředí VEXcode123, po kliknutí na tlačítko Robot zvolíme Connect, kde se objeví okno s názvem zapnutých robotů a silou Bluetooth signálu. Po výběru robota je proces propojení dokončen. Samotné programování probíhá tedy metodou „Drag and drop“. Prvním blokem každého programu je blokový příkaz when started. Na plochu k sestavení programu jednoduše příkaz přetáhneme a položíme. Nami vytvořená sekvence příkazu se spustí tlačítkem start, kde se program zkompiluje a pošle do robota. Tlačítko Step využijeme v případě, pokud chceme projít program v jednotlivých krocích a optimalizovat nějakou část programu, která může být problematická. Program lze ukončit tlačítkem stop, nebo vložením příkazu stop project.

## 2.6 Popis VEX GO

Tuto robotickou sadu je možné začít používat ve výuce s žáky staršími 8 let. Při koupi Go Education Kit dostaneme dva kufry se všelijakými součástkami k stavbě robota, motory, senzory a další elektroniku. Na rozdíl od sady VEX 123 přichází další možnosti pro rozvoj kreativity žáků z pohledu konstrukce robota. S použitím součástek lze postavit Super Car, Super Code Base Car, robotické rameno a cokoliv, co si žák vymyslí. Na webu je možné dohledat přehledné postupy k sestavení robotů v několika verzích.



10

*Obr. 3 - Super Code Base Car a Code Robot Arm*

Jak začít s robotickou sadou. Od věci nemusí být celkové zorganizování a celkový pořádek. Pokud bude ve třídě 20 žáků, kteří jsou po dvojicích, tak by každá dvojice měla mít tyto dva kufry. Žáci pak budou mít práci jednodušší, budou součástky snáze hledat a nedojde k smíchání součástek v jednotlivých kufrech. Tudiž popsat každou stavebnici a k ní stejným způsobem baterie, mozek robota a senzory. Z počátku se může zdát, že je stavebnice komplikovaná, ale po pár stavbách a vyzkoušení si všech různých pinů a součástek dojde k osvojení. Po uspořádání součástek nabijeme baterie pomocí USB – C kabelu. Na boční straně baterie jsou dvě diody, které informují o stavu nabití baterie po zmáčknutí tlačítka vedle diod. Pokud svítí obě zeleně, tak je baterie plně nabitá, když svítí jedna zeleně, tak je baterie částečně nabitá (25-75 %). Jestliže svítí pouze jedna červeně, tak to znamená, že je baterie téměř vybitá, a je tedy nutné ji dobít. Po zapojení nabíječky

---

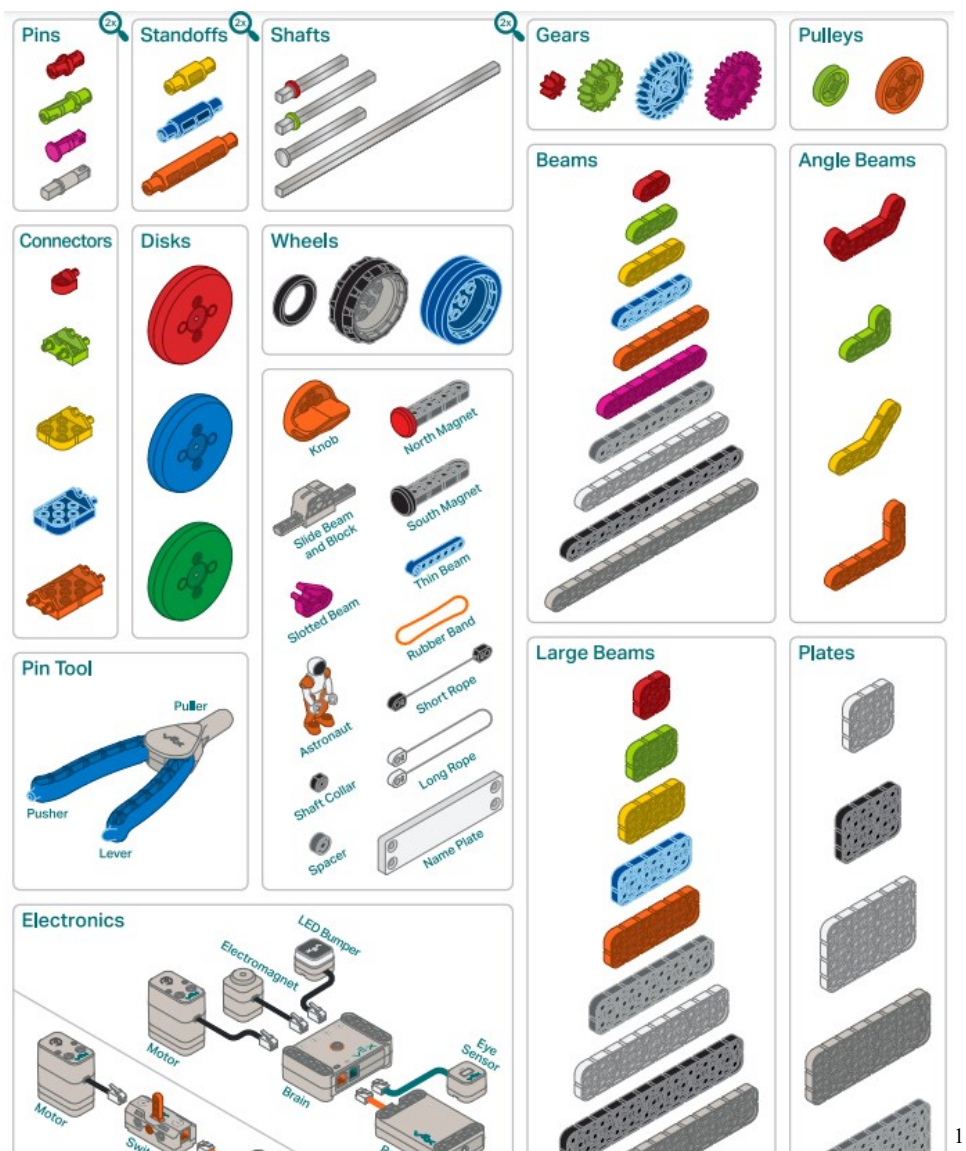
<sup>10</sup> Super Code Base Car a Code Robot Arm. In: *VEX* [online]. Greenville: VEX Robotics, 2022 [cit. 2022-04-10]. Dostupné z: <https://www.vexrobotics.com/go/downloads/build-instructions>

a při samotném nabíjení svítí dioda červeně. Po kompletním nabití svítí obě dvě zeleně. Baterie využijeme k propojení a napájení mozku, motorů a senzorů.

K programování se opět využije VEX code GO, což je jako u VEX code 123 totožné prostředí s blokovými příkazy. Přibudou pouze příkazy propojené s novými senzory, které VEX 123 nevyužíval. Výhodou prostředí GO je, že nabízí jazyk čeština, pro verzi 123 je plánu v co neblíží době. Osobně nerozumím tomu, proč verze pro starší žáky má češtinu dříve než pro žáky mladší. K propojení postupujeme stejně jako u VEX 123 kliknutím na tlačítko brain v prostředí a stisknutím tlačítka na mozku robota, které se rozblíká zeleně. Následně pomocí Bluetooth zařízení propojíme, kliknutím na tlačítko connect. K updatu firmwaru stačí robota připojit k prostředí VEX code GO. Po připojení se robot automaticky updatuje, druhou možností je aplikace VEX Classroom, kde je možnost, a tedy výhoda, připojit více zařízení najednou a tím je i zároveň updatovat naráz.

Samotná stavba je další dovedností, kterou si žáci osvojí. Mezi základní prvky patří plastové části, jako jsou desky a nosníky, které tvoří konstrukční základnu většiny výtvorů. Tyto destičky mají různé délky a šířky. Nosníky nabývají zase několika úhlů, a to 45°, 60° a 90°. Tyto základní stavební prvky propojujeme a kombinujeme pomocí kolíků, sloupků a tzv. konektorů. Dalšími součástkami jsou ozubená kolečka, kterých jsou 4 typy velikostí. Ty u motorů využijeme k přenesení síly z jedné pozice do druhé. Pokud použijí kolečka o stejné velikosti, tak se síla bude přenášet vyváženě, při použití koleček různých velikostí můžeme autíčko zrychlit nebo zpomalit. Když jsme u pohybu, další součástí jsou samotná kola, pneumatiky a kladky, která nám mohou ovlivňovat trakci, a tedy jak se bude kolo prokluzovat. Kolo a krokový motor je propojen pomocí hřídelí, které mají tvar hranolu, pomocí kterých dojde k otáčení kola. Na závěr elektronika, která robota řídí, ovládá, ovlivňuje a pomocí které je napájen. Stavebním prvkem je mozek, který spouští program zkompileovaný v prostředí. Dále má další vstupy, do kterých lze připojit motory, nárazník a elektromagnet. Oranžový vstup využijeme pro baterii a modrozelený pro oční senzor. Pomocí krokového motoru dokážeme robota rozpohybovat. Může být zapojen přímo do mozku nebo do switche, kterým kontrolujeme směr jízdy dopředu nebo dozadu. Nárazník je v podstatě tlačítko, kterým můžeme ovládat cokoli, když bude zmáčknuto. V tomto tlačítku jsou zároveň LED diody, takže je možné ovlivnit barvu, kterou svítí.

Elektromagnet využijeme pro jakékoli sbírání barevných puků, které jsou také součástí sady. Využívá elektromagnetu, který mění mezi polohou on/off. Oční senzor dokáže vnímat barvy (červená, zelená, modrá) a také intenzitu světla (světlý, tmavý).



11

Obr. 4 - Součástky VEX GO

<sup>11</sup> VEX GO Parts poster. In: *VEX* [online]. Greenville: VEX Robotics, 2022 [cit. 2022-04-10]. Dostupné z: [https://www.vexrobotics.com/catalogsearch/result/?vex\\_site=downloads&q=\\_\\_empty\\_\\_](https://www.vexrobotics.com/catalogsearch/result/?vex_site=downloads&q=__empty__)

## 2.7 Popis VEX IQ

Na sdílení informací o tom, co řídí roboty, je kanadský výrobce vcelku skoupý. Žádný technický popis nenajdete ani v dodaném zboží, ba ani na stránkách výrobce se mi nepodařilo nalézt jakékoliv bližší informace. Na dotaz přímo v technickém oddělení mi nakonec bylo sděleno následující:

Hello Oliver Erbes,

Thank you for your email, I will be happy to help you.

Inside of the Microcontrollers we have a PCB board, usually being operated by a "One Cortex A9" processor or a Texas Instruments Tiva microcontroller.

Eli Osornio

VEX Robotics

Z této odpovědi tedy můžeme usuzovat na to, že v robotech najdeme, zřejmě jednojádrový Cortex A9 nebo Tiva, což je procesor Texas Instruments na bázi ARM procesorů. Oba jsou tedy z téže rodiny. Věnujme se dále procesoru Cortex A9.

Ten se vyskytuje prakticky ve dvou verzích. Jednak se jedná o jednojádrovou verzi a poté ještě existují verze až čtyřjádrové označované jako MPCore. Jak již bylo řečeno, v robotech se se vši pravděpodobností vyskytuje Cortex A9 s jedním jádrem a instrukční sadou ARMv7-A. Jedná se 32 bitový procesor taktovaný až do rychlosti 2MHz s rozšířením Jazelle, tedy s možností přímého spouštění osmibitového Java bytekódu. Velikost L1cache je 32 KB a L2 může mít 128 KB – 8 MB.

Zdroje: Cortex-A9 Technical Reference Manual, ARM instruction set

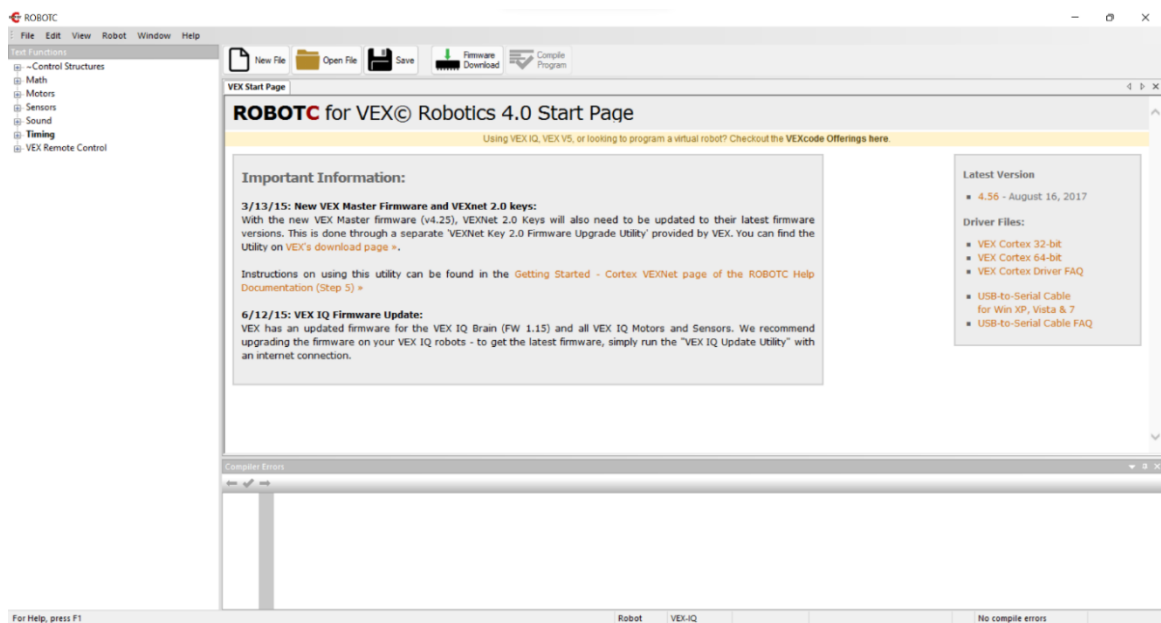
## 2.8 Popis jazyka ROBOTC

Dva druhy robotů řady VEX je možno programovat i pomocí textového editoru, tak jako píšeme aplikace ve vyšších jazycích. K tomu je určen jazyk a vývojové prostředí ROBOTC. Jak už sám název napovídá, je založen na jazyku C a je určen pro robotické systémy VEX IQ a VEX CORTEX.

Pro jeho užívání je třeba si nejdříve stáhnout a nainstalovat aplikaci VEX Code IQ, kterou najdeme v části Downloads při výběru karty IQ na stránkách vexrobotics.com. Poté stačí jen vybráním položky VEX Code software vybrat a následně stáhnout potřebnou variantu (Windows nebo Mac). Aplikaci je také možno stáhnout na klasických stránkách jako např. Google Play nebo App Store. Následná instalace, alespoň v mém případě, proběhla rychle a bez problémů.

VEX IQ se k počítači nepřipojuje pomocí bluetooth, ale prostřednictvím rozhraní USB. Při vlastním připojování je třeba mít se na pozoru, neboť aby spojení plnohodnotně fungovalo, je třeba provést download firmware.

Po startu se objeví obrazovka, kterou můžeme rozdělit do několika sektorů. Nahoře je klasické menu s položkami File, Edit, View, Robot, Window a Help. Jedinou netradiční položkou zde je Robot, ke které se později vrátíme. Pod tímto menu vlevo je část nazvaná Text function, věnující se některým příkazům, které rozšiřují C o další funkce určené na ovládání motorů, čidel a podobně. V pravé horní části je projektové menu sloužící k práci s editovaným souborem, stažením firmware, kompilací a spuštěním aplikace v robotovi. Pracovní plocha je určena pro psaní kódu vlastní aplikace a pod ní je kompilační okno, které ukazuje průběh kompilace včetně případných chyb v syntaxi kódu. Úplně dole je běžný stavový řádek. V dalším textu se blíže seznámíme s položkou Robot, Text function a ukážeme si několik jednoduchých ukázek.



Obr. 5 - Pracovní plocha ROBOTC

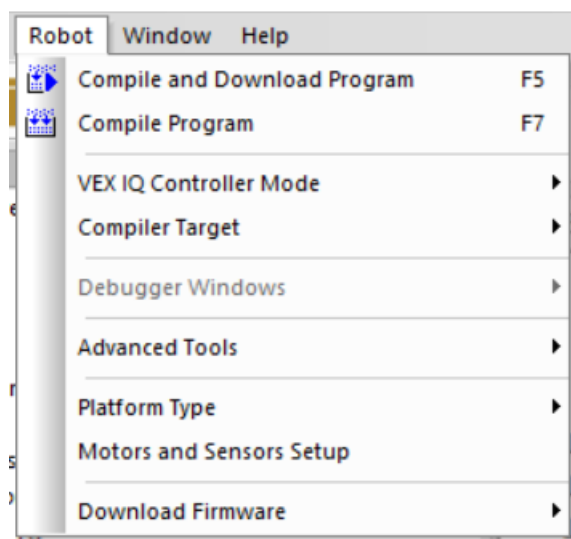
## 2.9 Nabídka Robot

V nabídce Robot jsou seskupeny příkazy sloužící ke komunikaci s robotem a pro kompilaci napsaného kódu.

První dva kompilují a případně hned odesílají zkompilovaný kód do robota. Další příkazy jsou určeny pro to, aby byl program ROBOTC informován a tom, co k sobě a v jaké konfiguraci má vlastně připojeno.

VEX IQ Controller mode umožňuje výběr ze dvou možných režimů TeleOP nebo autonomous. V prvním případě se jedná o řízení robota prostřednictvím dálkového ovladače nebo síťového připojení. V našem případě volíme autonomní režim, tedy režim, kdy se robot „řídí“ sám podle nahraného programu.

Compiler target je položka určující, kam se má zkompilovaný kód nahrát. V našem případě přímo do paměti robota.



Obr. 6 - Nabídka Robot pro program ROBOTC

V části Platform type je třeba vybrat patřičný typ robota, kterého používáme. V mém případě se jedná o již několikrát zmiňovanou VEX IQ.

Jedna z nejdůležitějších částí je pak Motors and Sensors Setup. Zde musíme nastavit buď přesný typ robota, kterého máme postaveného, a v případě netypizovaného stroje pak musíme zadat, jaké zařízení je na kterém portu. V případě, že jsme postupovali přesně podle sestavovacího návodu, stačí vybrat typ, a porty se nastaví automaticky. V opačném případě musíme ručně nastavit, co na kterém portu je, ať již se jedná o senzor, či o motorickou jednotku.

Neméně důležitá je i položka Download Firmware, protože před první kompilací musíme nahrát do robota firmware. Pokud tak neuděláme, budeme na tuto realitu upozorněni a kompilace nebude nahrána.

## 2.10 Pole Text Function

V této nabídce jsou obsaženy knihovny disponující některými klasickými a některými speciálními funkcemi určenými pro ovládání příslušných robotů jako například ty, které se týkají ovládání motorů, načítání dat ze senzorů a podobně.

### 2.10.1 Math

Math je klasická knihovna matematických funkcí a výpočtů, které se mohou používat k psaní aplikací bez připojování nějaké další knihovny funkcí. Jedná se například o funkce sinus, arcus sinus, odmocniny a o řadu dalších. Popis jednotlivých funkcí myslím není třeba provádět, neboť jejich užití je zřejmé.

### 2.10.2 Motors

- **moveMotorTarget(nMotorIndex, nPosition, nSpeed)** umožňuje otočit motorem o určitý počet stupňů zadanou rychlostí.
- nMotorIndex je proměnná typu tMotor, což v praxi znamená dvě možnosti. Známe-li port, na kterém je motor, pak stačí napsat slovo motor a za něj uvést patřičné číslo např.: motor1, což znamená, že myslíme motor na portu 1. Stejně tak je možno napsat přímo číslo portu, ovšem snížené o jedna. Tedy pro port1 musíme napsat číslo 0.

- `nPosition` udává, o kolik stupňů se má motor otočit. Kladná hodnota značí směr vpřed, záporná způsobí otáčení motoru zpět.
- `nSpeed` nastavuje rychlost motoru a může nabývat hodnot od -100 do +100. Zcela nečekaně záporné znaménko neznamena pohyb zpět, ale nastavení rychlosti např. na -20 má stejný efekt jako nastavení rychlosti na +20.
- **`setMotorTarget(nMotorIndex, nPosition, nSpeed)`** umožňuje otočit motorem na absolutní pozici udanou v proměnné `nPosition`. Proměnná `nMotorIndex` má stejný význam jako v předchozím příkazu. Proměnná `nPosition` se zde však liší v tom, že neudává, o kolik stupňů se má motor otočit, ale na jakou polohu se má motor otočit. Jako vysvětlující příklad uveďme dva příkazy:

```
moveMotorTarget(motor1,360,20);
```

```
setMotorTarget(motor1,0,20);
```

První příkaz způsobí popojetí robota o 360° vpřed a druhý příkaz jej vrátí zpět, odkud vyjel.

- **`waitUntilMotorStop(motor1)`** způsobí čekání aplikace až do zastavení motoru. Tento příkaz je vhodné použít vždy, když chceme, aby motor vykonal nějaký určitý pohyb. V případě, že bychom žádný čekací příkaz neuvedli, nemusel by se vůbec žádný pohyb motoru provést, ale ihned bychom skočili na další příkaz.
- **`setMotorSpeed(nMotorIndex, nSpeed)`** roztočí motor určený proměnnou `nMotorIndex` rychlostí `nSeed`. Rychlost může být nastavena v rozmezí -100 až 100, kde mínus znamená zpětný chod. U tohoto příkazu existuje jeden zásadní rozdíl oproti příkazu `moveMotorTarget`, a to ten, že směr vpřed je závislý na fyzickém uložení motoru. V případě robota, kde kola jsou ovládána motory, které jsou umístěny zrcadlově proti sobě, se bude jedno kolo otáčet vpřed a druhé tím samým příkazem vzad (levé vpřed a pravé vzad). Oproti tomu příkazy `moveMotorTarget`, resp. `setMotorTarget` točí oběma koly stejným směrem podle parametru `nPosition`. Pokud chceme tomuto zmatku zabránit, musíme nejdříve použít následující příkaz.
- **`setMotorReversed(nMotorIndex, bReversed)`** umožňuje nastavit směr otáčení kola příslušného motoru. Hodnota typu boolean je pak `bReversed`. Je-li nastavena na `true`, motor se bude otáčet v opačném směru. Prakticky toho využijeme tak, že

když motory ovládáme příkazem `setMotorSpeed`, tak u levého nastavíme hodnotu `bReversed` na `false` a u pravého na `true`. Pak se příkazem `setMotorSpeed` se stejným parametrem `nSpeed` budou kolečka točit stejným směrem, a neprojeví se tak „zrcadlový efekt“.

- **`resetMotorEncoder(nMotorIndex)`** nastaví počítadlo otáček zvoleného motoru na nula, neboli vytvoří počáteční či startovní bod pro další pohyby. Tento příkaz používáme tehdy, když dokončíme nějakou činnost a chceme začít nějakou další, jakoby od bodu 0. Pokud bychom chtěli mít pro případný pozdější návrat stav motoru uložený, museli bychom použít následující příkaz.
- **`getMotorEncoder(nMotorIndex)`** vrátí aktuální pozici otočení motoru ve stupních a je typu `float`.

### 2.10.3 Sensors

Tato část obsahuje funkce určené na načítání dat z jednotlivých senzorů, které jsou ve VEX IQ k dispozici. Jedná se o senzory vnímání barvy, vzdálenosti, dotyku či stisknutí a gyroskopu. Podívejme se nyní na jednotlivé senzory:

#### **Bumper sensor**

Bumper sensor je vlastně tlačítko, které má dva stavy: stisknuto – 1, nestisknuto 0, a právě tyto dvě hodnoty předává ve své jediné funkci.

Funkce `getBumperValue(nDeviceIndex)` je funkce, která vrací hodnotu 1, v případě, že je tlačítko stisknuto a 0 v opačném případě. Tlačítko nemá paměť a posílá aktuální stav. V robotu je použito pro zjištění, zda je zvedací zařízení „zaparkováno“.

## Color sensor

Color senzor se používá ke zjištění barvy předmětu, který se nachází v zorném poli senzoru. Výsledek se zobrazuje v závislosti na nastaveném barevném režimu.

- `setColorMode(nDeviceIndex, ColorMode)` nastaví příslušný senzor do jednoho z barevných režimů, které jsou uvedeny níže v tabulce.

Barevné módy:

<code>colorTypeUninitialized</code>	nedefinovaný režim senzoru
<code>colorTypeGrayscale_Ambient</code>	škála šedé barvy při vypnutém světle senzoru
<code>colorTypeGrayscale_Reflected</code>	škála šedé barvy při zapnutém světle senzoru
<code>colorTypeRGB_12Colors_Ambient</code>	jméno barvy s vypnutým světlem senzoru
<code>colorTypeRGB_12Colors_Reflected</code>	jméno barvy se zapnutým světlem senzoru
<code>colorTypeRGB_Hue_Ambient</code>	odstín barvy při vypnutém světle senzoru
<code>colorTypeRGB_Hue_Reflected</code>	odstín barvy při zapnutém světle senzoru

*Tabulka 1 - Barevné režimy pro senzory*

- `getColorName(nDeviceIndex)` vrátí načtenou barvu v jedné z dvanácti předdefinovaných barev, případně vrátí nerozpoznanou barvu hodnotou (`colorNone`, `colorRedViolet`, `colorRed`, `colorDarkOrange`, `colorOrange`, `colorDarkYellow`, `colorYellow`, `colorLimeGreen`, `colorGreen`, `colorBlueGreen`, `colorBlue`, `colorDarkBlue`, `colorViolet`). Barva pokusné kostky s rozměry 5 cm byla vnímána zhruba do vzdálenosti 8 cm od senzoru.

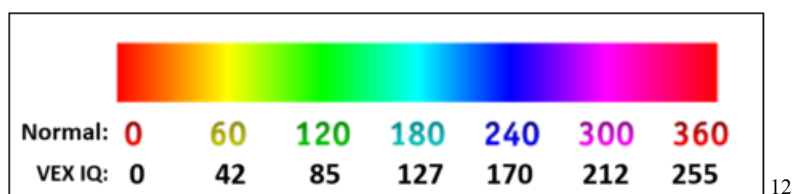
Kromě funkce `getColorName` můžeme požit také funkci `getColorGRayScale`, která vrací číselnou hodnotu 0-255, nebo funkci `getColorHue` vracející číslo v hodnotě 0-255, viz obrázek níže.

## TouchLED sensor

TouchLED senzor je tlačítko osvětlené LED diodami, jejichž barvu můžeme nastavit. Může tedy fungovat jednak jako jakýsi efektní prvek, nebo také informační, neboť svojí barvou může něco sdělovat.

`getTouchLEDValue(nDeviceIndex)` vrací stejně jako u tlačítka bumper hodnotu 1 nebo 0 v závislosti, zda tlačítko bylo stisknuto, či ne.

- `setTouchLEDRGB(nDeviceIndex,red,green,blue)` nastaví barvu příslušného tlačítka podle hodnot jednotlivých zvolených barevných složek. K nastavení barvy je možné použít jeden z příkazů `setTouchLEDColor (nDeviceIndex,nColor)` nebo `setTouchLEDHue (nDviceIndex, nHueValue)`. V prvním případě textovou hodnotou barvy (`colorNone`, `colorRedViolet`, `colorRed`, `colorDarkOrange`, `colorOrange`, `colorDarkYellow`, `colorYellow`, `colorLimeGreen`, `colorGreen`, `colorBlueGreen`, `colorBlue`, `colorDarkBlue`, `colorViolet`), ve druhém odstínem v číselném rozmezí 0 – 255 podle přiloženého obrázku převzatého z nápovědy aplikace ROBOTC.



12

Obr. 7 - Hodnoty barev ve VEX IQ

## Gyro sensor

Gyro senzor umožňuje určit směr, ve kterém se právě robot nachází, a nalézá uplatnění v řadě pohybových úloh. Tomuto senzoru můžeme zadat nulový směr nebo z něho načíst okamžité natočení.

- `resetGyro(nDeviceIndex)` nastaví aktuální směr robota jako nulový, je třeba pouze zadat port, ve kterém je gyroskopický senzor zapojen.
- `getGyroDegrees(nDeviceIndex)`, resp. `getGyroFloat(nDeviceIndex)` vrátí hodnotu úhlu ve stupních, resp. jako desetinné číslo. Obě hodnoty mohou nabývat kladných i záporných hodnot. Senzor také umí vrátit úhel, který měl robot při posledním resetu. Tuto hodnotu lze načíst pomocí funkcí `getGyroHeading(nDeviceIndex)`, resp. `getGyroHedaingFloat(nDeviceIndex)`.

<sup>12</sup> ROBOMATTER INC. *ROBOTC for VEX Robotics 4.X - Users Manual* [software]. 2022 [cit. 2022-04-12]. Požadavky na systém: Win 7, Win8, Win10.

#### 2.10.4 Sound

Vydáním nějakého zvuku nás může robot upozornit na to, že skončil nějaký proces, a proto je dobré zvuky do aplikací vkládat, ať už jako ladící pomůcku, nebo jako upozorňovací komunikační nástroj. Část věnovaná zvukům obsahuje pouze tři příkazy, přičemž ten poslední se v případě mého robota ukázal jako nefunkční, respektive se mi jej nepodařilo jakýmkoliv způsobem zprovoznit.

- `playSound(sound)` přehraje některý z připravených zvuků, kterých je celá řada. Jako příklad si uveďme `soundTada`, `soundCarAlarm2`, `soundWave...`
- `playRepetitiveSound(sound, duration10MsecTicks)` – přehraje zvuk z knihovny výše uvedených zvuků, avšak můžeme zadat, jak dlouho se zvuk má přehrávat v trvání 10 milisekundových kliků (myšleno metronomu).
- `playNote(nNote, nOctave, duration10MsecTicks)` přehraje tón ze škály předdefinovaných tónů (`toneC`, `toneD`, `tone GSharp...`) v předdefinované oktávě (`octav1`, `octave2`, `octave3`) a po dobu 10 milisekundových kliků. Jak jsem již předeslal, tato funkce mi ani po řadě pokusů a ani po odzkoušení oficiální nápovědné ukázky nefungovala.

## **3 Sady úloh pro VEX**

### **3.1 1. stupeň**

Pro první stupeň využijeme sadu VEX 123, ke které budeme přistupovat třemi způsoby možného programování. Výuka probíhala na ZŠ Na Lukách v učebnách informatiky. Potřebným řešením bylo rozdělení žáků na 2 skupiny, kvůli vysoké účasti téměř čtyřiceti osob. Tyto hodiny probíhaly tedy dvakrát. Využili jsme kooperační metody výuky a žáky rozdělili do malých skupin, ve kterých řešili jednotlivé úlohy.

K jednotlivým úlohám jsou zdokumentovány videosekvence, na které směřují odkazy uvedené v příloze. Může se jednat o jinou formu zadání úlohy, zjistí podle videa, jakou činnost robot vykonává, a stejným způsobem ho taky naprogramuj.

#### **3.1.1 Programování pomocí dotyku**

##### **Úloha číslo 1**

Zadání: Dojeď s robotem 6 kroků rovně.

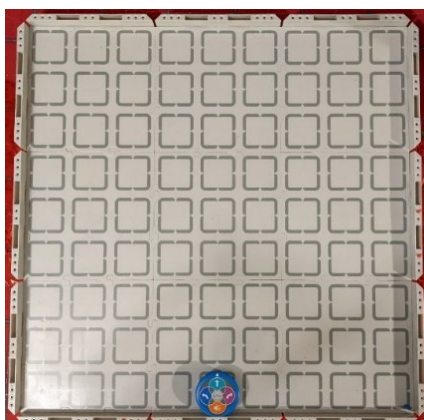
Popis: V této úloze se žáci seznámí s robotem 123, který má tvar puku. Zjistí, jak se robot zapíná a jak ho mohou rozpohybovat. Intuitivně pochopí, že šipka mířená směrem dopředu bude jízda dopředu, ale jestli je to krok, nebo kolikrát se musí tlačítko zmáčknout a jak program zapnout, zřejmě pro žáka být nemusí.

Pomocné otázky: Jak robota zapnu? Co dělají tlačítka na vrchu robota? Jak robot pojede rovně? Můžu kliknutí opakovat? Kolikrát musím tlačítko zmáčknout? Jak program zapnu, aby se robot rozjel a udělal, co jsme mu řekli?

Řešení: Zřejmě, šestkrát opakují stisknutí tlačítka se šipkou dopředu. Program se spustí delším podržením bílého tlačítka ve středu robota a následně zabliká.

Poznámky: Žáci mají rádi zvukové efekty, mohou si přidat na závěr příkaz zatroubení a tím úlohu zakončit. Jedinou potřebnou pomůckou je robot 123.

## Úloha číslo 2



*Obr. 8 – Zadání 2. úlohy s hracím polem*

**Zadání:** Jed' přes celé hřiště až na konec ke stěně a pak se vrať zpět do původní pozice (aby byl robot natočen v původním směru jízdy, když startoval).

**Popis:** Žáci si v této úloze vyzkouší a postaví herní pole (hřiště), kde si zjistí rozměry hřiště a tím zjistí, kolikrát musí robot jet. Jelikož jedno stisknutí posunu dopředu se rovná vzdálenosti jednoho čtverce.

**Pomocné otázky:** Jakou vzdálenost robot ujede při jednom stisknutí tlačítka dopředu? Dá se tato vzdálenost přirovnat k něčemu v hřišti? Jak velké hřiště je? Jak se vrátí robot zpět, když dojede do posledního políčka ke zdi? Jelikož je úloha zadána, vrať se do původní pozice, tak co musí robot ještě udělat?

**Řešení:** Délka hřiště v políčkách, následně dvě otočky vpravo/vlevo, délka hřiště v políčkách rovně a následné dvojitě otočení vpravo/vlevo.

**Poznámky:** Tato úloha může být zvolena i bez hřiště. Třeba tak, že si udělají dráhu po třídě, a mohou jezdit mezi lavicemi nebo židlemi. Potřebnými pomůckami jsou robot 123 a hrací pole.

## Úloha číslo 3

**Zadání:** Zjistí, jak je dlouhý jeden krok robota? Výsledek uveď v cm.

**Popis:** Primárním cílem úlohy je, aby se žák dozvěděl, jak dlouhý je jednotlivý krok. Přičemž si vyzkouší praktickou práci s metrem. Mohou udělat: např. deset kroků, změřit

ujetou vzdálenost od předku robota v startovní pozici do předku robota v cílové pozici. Následně změřenou hodnotu vydělit deseti, a tím získat vzdálenost jednoho kroku. Jelikož by se deset kroků na naše hrací pole nevešlo, tak jsme zvolili cestu úhlopříčně.

Pomocné otázky: Máš představu, jakou vzdálenost při jednom kroku robot ujede? V jakých jednotkách budeme měřit? Kolik kroků bude robot dělat? Jak, odkud až kam budeš měřit? Jak z naměřené ujeté dráhy vypočítáme hodnotu jednoho kroku?

Řešení: Přesun robota vpřed o deset kroků, změření ujeté vzdálenosti ze stejného bodu robota a vydělení 10.

Poznámky: Žáci mohou sledovat robota, jestli vznikají odchylky při pohybu a jaká je chybovost. Občas se totiž posune méně či více, začne zatáčet nebo se zadrhne. Mohou otestovat, na jakých materiálech robot chyboval více a na kterých méně. Potřebnými pomůckami jsou robot 123, metr, kalkulačka a hrací pole.

#### **Úloha číslo 4**

Zadání: Naprogramuj robota tak, aby jeho ujetá trasa byla ve tvaru schodů.

Popis: V dané úloze si žáci rozšíří pohled na věci z jiného úhlu, a to z bočního pohledu, v našem případě schody. Úloha může mít více výsledků, jelikož někdo si může představit schody jako čtverce, někdo jako obdélníky. Tudíž by robot jezdil v jiných vzdálenostech, ale princip schodů, dostat se do vyšší pozice, by byl zachován. Nezáleží ani na měřítku, pokud by žák chtěl schod vysoký dva, nebo tři posuny robota.

Pomocné otázky: Dokáže jet robot do schodů? Co to může ze zadání znamenat? Jak si dokážeš schody představit? Jaká bude výška a šířka jednoho schodu?

Řešení: Program začne krokem dopředu směrem nahoru, otočením vpravo, krokem dopředu, otočením vlevo atd. se program opakuje až do cílové pozice.

Poznámky: Úlohu lze modifikovat např. tak, že má robot nakreslit určitý tvar. Představitosti se meze nekladou. Může se jednat o půdorys jakékoli žákům známé budovy, např. na google mapách si mohou zvolit leteckou fotku školy a pomocí robota se jí pokusit nakreslit tím, že si pomocí konstrukce nebo izolepy připevní za robota tužku, který by pak za sebou dráhu kreslil. Možnou variantou úlohy je také nápis vlastního jména pomocí

robot. Překážkou by byl maximální počet kroků, které může robot pomocí tlačítek vykonat. Jelikož nápis svého jména by asi tuto kapacitu překonal. Můžeme říct, že se robot v podstatě pohybuje po čtvercové mřížce (pokud předpokládáme krok jako stranu čtverce a nedělá chyby), mohl by napsat i kolik je hodin v digitálním formátu. Potřebné pomůcky jsou robot 123, papír, tužka připevněná pomocí konstrukce k robotovi.

### **Úloha číslo 5**

Zadání: Naprogramuj, aby ujetá dráha robotu byla ve tvaru čtverce.

Popis: Úloha je na podobném principu jako předešlá. Může být jako příklad pro úvod pro cykly, jelikož žák bude provádět stejné operace 4 krát, což z programátorského hlediska není správně, ale v této úloze pomocí tlačítek jinak postupovat nelze. Žák si zopakuje základní vlastnosti platné pro čtverec, které jsou shodné délky stran a pravé úhly u vnitřních úhlů a zároveň jsou protilehlé strany rovnoběžné. Tudiž příkazy rovně a otoč se vpravo/vlevo, jsou dostačující a jediné potřebné.

Pomocné otázky: Jak vypadá čtverec? Jaké vlastnosti pro čtverec platí? Co se opakuje a kolikrát to nastane?

Řešení: Robot se přesune o daný počet kroků dopředu, otočí se vpravo/vlevo a tyto kroky opakuje celkem 4 krát včetně.

Poznámky: Jelikož není možnost nastavit jiný úhel než pravý úhel, tak jedinou jinou variantou úlohy je obdélník. Pokud má žák s úlohou problém a ztrácí se, může si čtverec nakreslit a po jednotlivých krocích si robůtka posouvat. Tím by se měla zvýšit pravděpodobnost správného počtu kroků.

## Úloha číslo 6



Obr. 9 - Zadání 6. úlohy s číselnou osou

Zadání: Ověř si na číselné ose pomocí pohybu robota, kolik je rozdíl  $8 - 5$ .

Popis: Na hrací pole připravíme osu přirozených čísel v našem případě od 0 do 8, jelikož nám hrací pole nedovolilo větší rozsah. Úloha má prověřit a prohloubit chápání rozdílu dvou čísel, součtu nebo stejných operací na celých číslech. Úloha může být koncipována pro více čísel, podmínkou je, aby byl výsledek mezi 0 až 8. Pro náš případ ujet 8 polí je zřejmé a rozdíl tedy bude vypadat tak, že se robot vrátí o 5 polí.

Pomocné otázky: Jak funguje operace sčítání a odčítání v porovnání s pohybem robota? Kde a na jakém čísle by se měl robot zastavit? Co musí udělat robot, aby začal odečítat? Souhlasí početní výsledek s pozicí?

Řešení: Robot se přemístí o 8 polí dopředu, otočí se a přesune se o 5 polí.

Poznámky: Úloha se může stanovit na jakoukoli operaci, teoreticky i násobení, které je komplikovanější. Pokud se postaví větší hřiště, tak se nemusíme omezovat čísly 0 až 8. Potřebnými pomůckami jsou robot 123 a hrací pole s vyznačenými čísly na polích.

### 3.1.2 Programování pomocí karet

#### Úloha číslo 1

Zadání: Seznam se s kartami základních příkazů a pomocí nich vytvoř taneční sestavu, ať jde o otočky, pohyb, chování robota šťastně, smutně apod.

Popis: Žák se v této úloze seznámí s novou možností, jak robota ovládat. Ne pomocí dotyku na tlačítka, ale pomocí kódovací karty. Do ní se vkládají jednotlivé příkazy, které

při této úloze objeví, jelikož se jedná o náhodnou „taneční“ sestavu. Žák vybírá z karet pohybových, ovlivňujících chování a zvukových.

Pomocné otázky: Prohlédl sis všechny příkazové karty? Máš představu, co bude robot dělat? Záleží na pořadí příkazů v kódovací desce? Jak udělá robot piruetu?

Řešení: Náhodná volba příkazů, připomínající tanec.

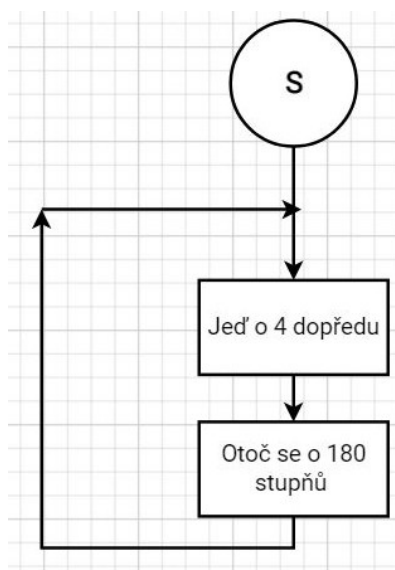
Poznámky: Pokud by žáci chtěli uspořádat taneční soutěž, mohou také své roboty pomocí papíru a různých dekorací ozdobit, a tím je k opravdovým tanečnickům přiblížit.

## Úloha číslo 2

Zadání: Jezdí neustále s robotem 4 kroky tam a zpátky.

Popis: Tato úloha je vytvořena jako seznámení se s cyklem a tím, aby robot neustále opakoval daný úkol, který nemusí být složitý. V tomto případě jde jen o 4 kroky dopředu, otočku směrem vzad, a tím je obsah cyklu uzavřen. Využije se příkazu go to start.

Algoritmizace:



Obr. 10 - Algoritmus 2. úlohy pro jízdu tam a zpět

Pomocné otázky: Jak provést 4 kroky dopředu a zpět je jasné, ale aby se daná činnost opakovala, to provedu jak? Jakého příkazu bych mohl využít? Znáš nějaké cykly? Jak bych mohl docílit, že se daný program zopakuje, a vrátí se tedy na začátek a postupně se příkaz po příkazu zopakuje?

Řešení: Vždy se začne příkazem when start 123, drive 4, turn around a nakonec go to start.

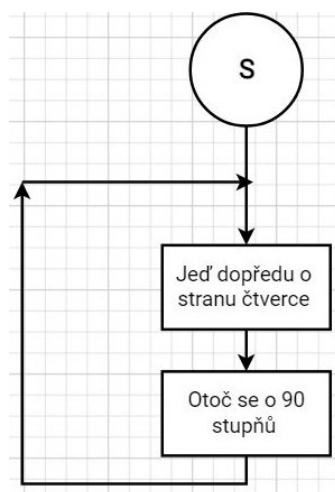
Poznámky: Předpokladem je, že žáci cyklus znát nebudou a nebudou vědět, jak způsobit opakovanou jízdu. Proto si myslím, že může být úloha důležitá. Potřebnou pomůckou je robot 123 a kódovací deska s kartami.

### Úloha číslo 3

Zadání: Nastav, aby robot neustále jezdil trasu ve tvaru čtverce.

Popis: Úloha už je známá, a to pomocí klikání na tlačítka. Žáci si mohou vyzkoušet stejný příklad jinou metodou, a to pomocí kódovacích karet. Úloha je vylepšena tím, že neobjedou čtverec jednou, ale nekonečně mnohokrát, neboť robot bude jezdit neustále do vypnutí programu. Využijeme tedy opět cyklus.

Algoritmizace:



Obr. 11 - Algoritmus 3. úlohy pro jízdu do čtverce

Pomocné otázky: Jak jsme postupovali při programování pomocí tlačítek? Co platí pro čtverec? Jak zaručíme, aby robot nepřestal jet, a jezdil tedy neustále?

Řešení: Opět příkaz when start 123, drive 2, turn right/left a go to start.

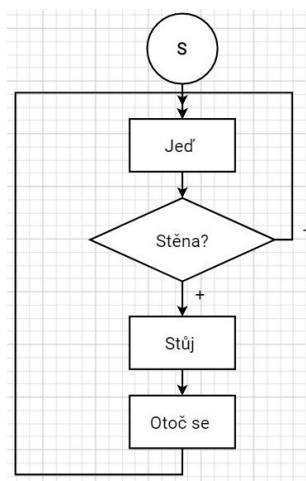
Poznámky: Variací na tuto úlohu s geometrickým pojetím je právě jedinou možností obdélník, jelikož máme stále pravé úhly. Potřebnou pomůckou je robot 123 a kódovací deska s kartami.

## Úloha číslo 4

Zadání: Objížděj celé hřiště dokola podél stěny, které se ale nesmíš nikdy dotknout.

Popis: Úloha je více méně podobná jako předešlá. Hlavní myšlenkou je, aby robot nenarazil do překážky, ale aby jel, a když detekuje překážku, tak zastavil, otočil se a pokračoval dál v jízdě. Tímto způsobem by robot měl objet celé hřiště kolem hranice hřiště. Reálnou představou a přirovnáním může být hlídač, který chodí kolem plotu a při dosažení překážky se otočí a půjde dál kolem plotu.

Algoritmizace:



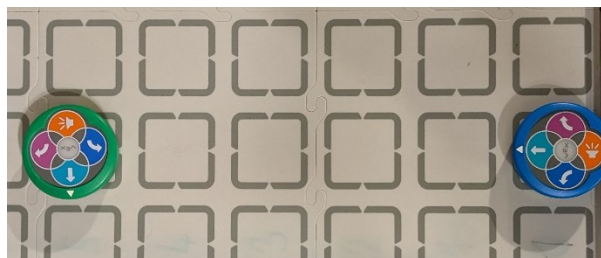
Obr. 12 - Algoritmus 4. úlohy pro objezd hřiště

Pomocné otázky: Jaký tvar má hřiště? Co se nesmí stát, a co proto robot musí udělat? Jak detekuje stěnu? Co bude robot dělat dál po detekci překážky? Bude se něco opakovat z daných příkazů? Je možnost využít cyklus?

Řešení: Ze začátku programu when start 123, drive until object, turn right/left a go to start.

Poznámky: Při úloze nesmí být robot v bezprostřední blízkosti zdi, jelikož by mohl ihned detekovat překážku a otočit se, a nesplnit tedy úlohu. Potřebné pomůcky jsou robot 123 a hrací pole.

## Úloha číslo 5

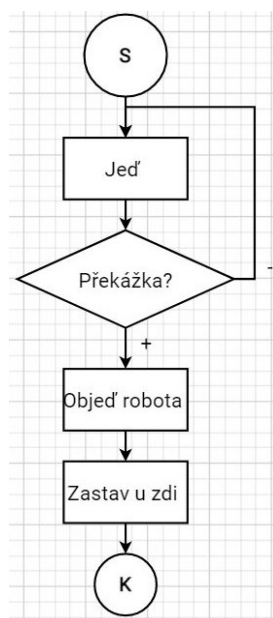


Obr. 13 - Zadání 5. úlohy

Zadání: Pokud se na trase vyskytuje překážka, kterou je zelený robot, tak ho objeddej a pokračuj ve své cestě až na konec hřiště ke stěně. Pokud je na cestě jiná než zelená překážka, zastav a ukonči činnost.

Popis: V této úloze robotovi v cestě překáží druhý zelený robot. Toho musí robot detekovat a bez jakéhokoli dotyku ho objet. Žáci využijí znalosti z předešlých úkolů, však navíc zde přibyla podmínka, co když detekuje zelenou barvu? V tu chvíli robot provádí sekvenci pohybů a otočení k překonání překážky. Následně stačí příkaz k dostání se robota ke stěně.

Algoritmizace:



Obr. 14 - Algoritmus 5. úlohy pro objetí robota

Pomocné otázky: Jak robot pojedě a nenarazí do překážky? Můžeš využít senzor k detekci zelené barvy a podle toho se dál řídit? Jak robota objedeš po jeho detekci? Jak se dostaneš po objetí na konec hřiště?

Řešení: Možným řešením je, aby jel robot rovně k objektu, kde, pokud detekuje zelenou, se zastaví a zeleného robota sekvencí posunů a otočení objede. Po objetí je jediná možná karta k dojetí do konce hřiště `drive until crash`. Tato možnost však zabere 11 příkazů a nevejde se do kódovací karty. Druhou variantou řešení je příkaz `drive until object`, kde se před objektem robot zastaví. Dál pokračuje jako v předešlém případě.

Poznámky: Omezující se stává počet možných příkazů v kódovací kartě, jelikož jich je pouze 10. Je tudíž možné, že nezbyde místo pro dokončení programu. Proto složitější úlohy nepřípadají v potaz. Ideálnější možností je prostředí VEX code.

### 3.1.3 Sada úloh v prostředí VEX code

#### Úloha číslo 1



Obr. 15 - Zadání 1. úlohy

Zadání: Dostaň se ze startovní pozice do cílové, nedotkni se překážek a zvol přitom nejkratší cestu. Až se dostaneš do cíle, rozsviť svého robota na modro. Popřemýšlej o dalším rozmístění překážek a variantách, jak se dostat do cíle, a najdi nejrychlejší variantu.

Popis: Zadání se odehrává na hracím poli, na jedné desce, tvořené 3 x 3 políčky. Kde se má robot dostat z bodu A do bodu B, v cestě stojí překážky. Úloha má žáka seznámit s programovacím prostředím Vex Code 123 a se základními příkazy.

Pomocné otázky: Jakou trasu robot musí zvolit?

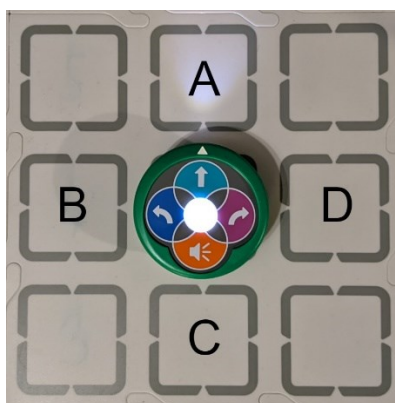
Řešení:



Obr. 16 - Řešení 1. úlohy

Poznámky: V této úloze můžeme vymyslet spoustu variant, např. kdyby dal prostřední překážku do levého horního rohu. Robot by mohl jet po polích, otáčet se o 90 stupňů a dojet do cíle, ale je možnost rychlejšího, s kterou může přijít druhý žák, že se otočí o pouhých 45 stupňů, pojedou krok rovně a pak se dotočí o zbylých 45 stupňů a zase o krok rovně, tak se dostane rychleji do cíle. Tímto se mohou vymýšlet složitější úlohy a stopovat, kdo se co nejrychleji dostane do cíle. Zadání se může rozšířit od pole 3 x 3 na větší i na celé hrací pole, kde se mohou vytvořit další body, které musí postupně navštívit. Potřebné pomůcky jsou robot 123, hrací pole a překážky.

## Úloha číslo 2



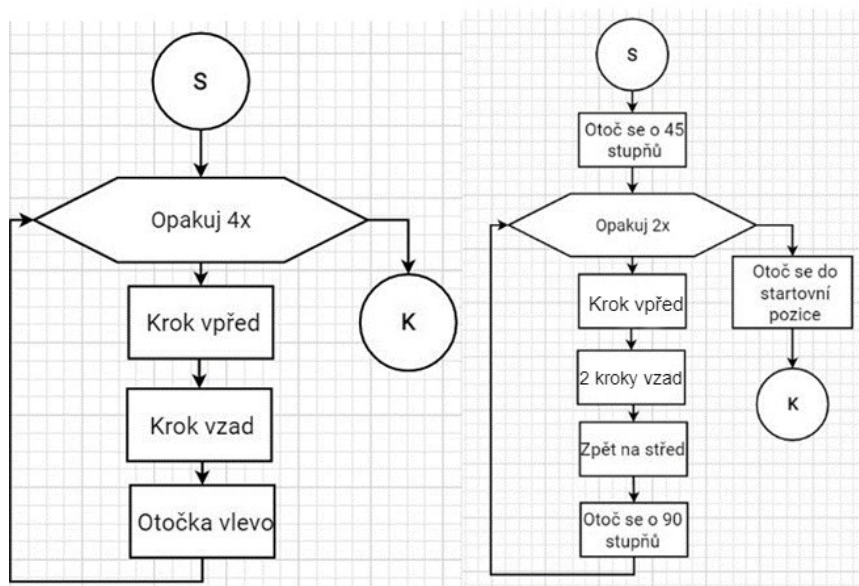
Obr. 17 - Zadání 2. úlohy

Zadání: V úloze jsou 4 body ABCD v tomto pořadí. Každý z nich postupně v daném pořadí musí robot navštívit a vždy se vrátit do startovní pozice do středu (tedy prvně

navštíví bod A, vrátí se do středu a pak bod B atd.). Druhou částí úkolu, která je bonusová, je navštívení všech 4 rohů, ale zase musí přejet přes startovní pozici.

Popis: Robot má ze všech stran body, do kterých se musí postupně po jednom dostat. V reálu by se robot dal využít v roli rozřídovače, kde na startovní pozici něco vezme a co má odnést do bodu A, vezme do A atd. pro ostatní. Úloha využije cyklu a to se opakuje tolikrát, kolik je bodů, proto v našem případě 4 krát. Opakovat bude krok tam a zpět s následovným otočením vlevo o 90 stupňů. V druhé části, která je jako bonus, robot navštíví neoznačená pole, kterými jsou rohová políčka. Dá se říct, že zadáním je, aby jel robot do kříže, když se budeme snažit trasu provést co nejefektivněji a nejrychleji.

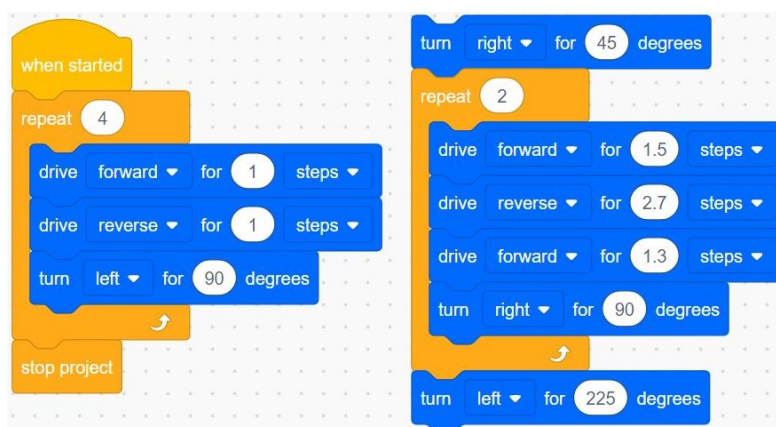
Algoritmizace:



Obr. 18 - Algoritmus 2. úlohy a bonus

Pomocné otázky: Jak bude teoreticky trasa robota vypadat? Pokud dojedu do bodu A, musím se otáčet, nebo mám jinou možnost, jak se vrátit na start? Jak bude vypadat trasa do bodu B? Opakuje se program, pokud ano, co to pro tebe znamená? Kolikrát se tedy cyklus bude opakovat? Jak se vyplatí začít pro bonusový úkol teoreticky a jak by trasa měla vypadat? O jaký úhel se robot natočí, aby mířil směrem na roh?

Řešení:



Obr. 19 - Řešení 2. úlohy

Poznámky: Úloha se dá lehce zaměnit pořadím bodů, čímž se změní robotova trasa, nebo přesunutím se na větší plochu hřiště s náhodným rozházením bodů po herní ploše. Potřebné pomůcky jsou robot a hrací pole.

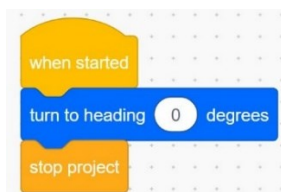
### Úloha číslo 3

Zadání: Sežeň si papír a podlož s ním robota. Následně vlož do programu příkaz „turn to heading 0 degrees“. Sleduj, co se bude dít a jak program ovlivní, když budeš s papírem, na kterém je robot, otáčet. Co si vypočítal, když byl program vypnutý, a jaká součástka v robotovi zajišťuje výsledek?

Popis: Tato úloha není náročná na rozvoj žákovy kreativního myšlení, ale jedná se o hříčku s gyroskopem, kde žák může nakouknout do robota, pozorovat, co je uvnitř a jak gyroskop funguje. Když podložím pod robota papír a začnu s ním otáčet, tak je jasné, že se bude otáčet i robot. Jenže když program zapnu a žák točí s papírem, robot si na začátku zapamatoval pozici jako 0 stupňů, do které se neustále snaží vrátit, jestliže se stále otáčí s papírem. Což pro žáky může být zajímavé a zábavné zjištění.

Pomocné otázky: Co pozoruješ za rozdíl po spuštění programu? Kam se robot neustále otáčí? Víš, díky čemu je toho schopen?

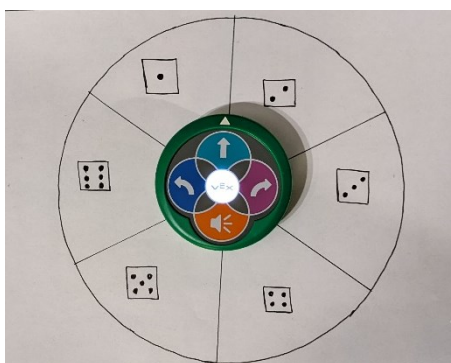
Řešení:



Obr. 20 – Řešení 3. úlohy

Poznámky: Zvolený úhel nemusí být 0 stupňů, žáci si mohou zvolit jiný a snažit se zabránit robotovi se do daného úhlu dostat. Potřebné pomůcky jsou robot 123 a papír.

#### Úloha číslo 4



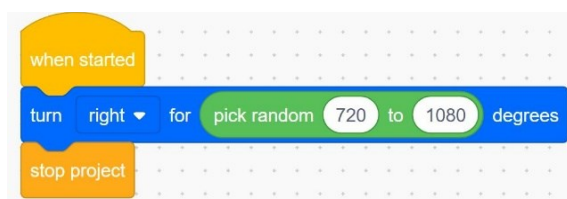
Obr. 21 - Zadání 4. úlohy

Zadání: Nemáš kostky, tvým úkolem je, si je zahrát pomocí robota. Vytvoř si dle obrázku kartu, po které se robot bude náhodně točit a tím zvolí číslo, které padne na kostce.

Popis: V této úloze žáci mohou rozvíjet svoji kreativní složku, i když se také dozví něco nového, jelikož se objevuje nový příkaz pick random. Pomocí něho dostaneme potřebný náhodný jev pro vylosování čísla z kostek. Robot se otočí na náhodně zvolený úhel. Pro přidání napětí je možné přidat jednu, dvě otočky navíc, tudíž by se nevolil úhel od 0 do 360 stupňů.

Pomocné otázky: Co má robot dělat? Můžeš provést, aby se robot otáčel náhodně? Jaký příkaz pro to využiješ? Umíš přidat otočku navíc pro pocit napětí? Kolik stupňů je jedna otočka?

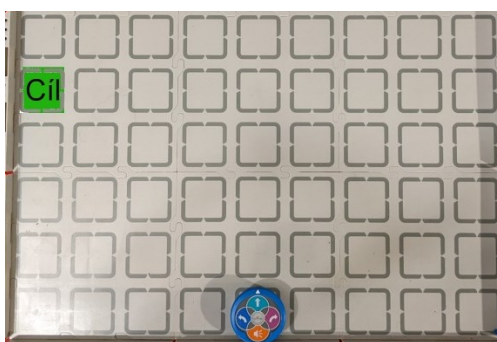
Řešení:



Obr. 22 - Řešení 4. úlohy

Poznámky: Žáci mohou výrobek a robota využít při hraní jakýchkoli deskových her. Pro starší lze využít i k hazardu jako např. využít robota jako ruletu, kde bude vybírat číslo jak u kostek. Potřebné pomůcky robot 123, papír, propiska, kružítko, pravítko, fix a jakékoli barevné pastelky pro dekoraci.

### Úloha číslo 5

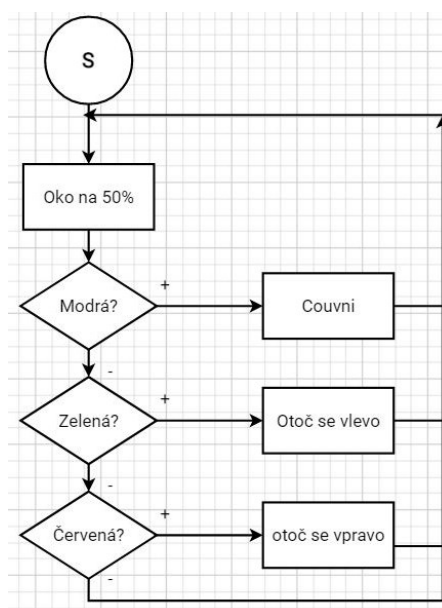


Obr. 23 - Zadání 5. úlohy

Zadání: Zaparkuj ze startovní pozice do cílové. Řid' se podle barev. Když robot detekuje červenou barvu, tak se robot otočí vpravo. Pokud bude detekovanou barvou modrá, robot bude couvat, a zelená barva robota otočí vlevo.

Popis: Daná úloha může působit lehce, ale při úloze mohou žáci narazit na několik překážek. Prvním problémem může být samotná detekce barev, jelikož oko snímá barvy z dálky. Proto se musí snížit jeho intenzita. Druhým problémem může být, že robot detekuje barvu hned dvakrát, i když chceme pouze jednou. Toho se zbavíme způsobem, že robot počká 1 vteřinu.

Algoritmizace:



Obr. 24 - Algoritmus 5. úlohy

Pomocné otázky: Jak se dostanu do cíle, pokud můžu jenom couvat a jsem nasměrován špatným směrem? Jak se můžeš otočit? Jaké příkazy využiješ? Jaký typ podmínky využiješ? Co musí robot dělat, když nebude detekovat žádnou barvu? O jaký úhel je výhodné, aby se robot otočil? Co když bude detekovat robot barvy příliš daleko? Jak zabrániš situaci, kde se robot otočí o 180 stupňů?

Řešení:



Obr. 25 - Řešení 5. úlohy

Poznámky: V dané úloze se mohou zaměnit podmínky podle barev, nebo pozice, do kterých má robot přijet. Zároveň si žáci mohou vytvořit bludiště s překážkami a tímto způsobem ho projet.

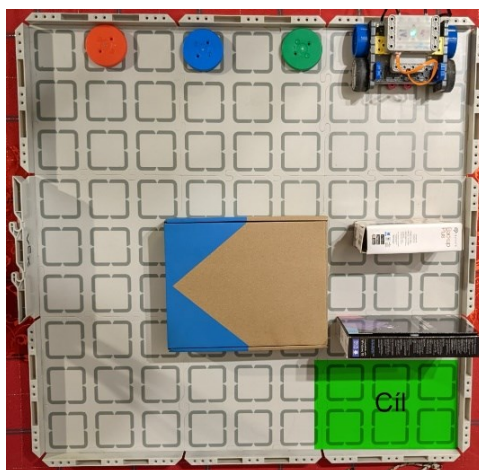
## 3.2 2. stupeň

Pro druhý stupeň využijeme sadu VEX GO a po zvládnutí úloh z této sady předvedeme pokročilejší variantu VEX IQ s různými metodami programování. Výuka probíhala na ZŠ Pomezí v učebnách informatiky. Na rozdíl od 1. stupně nebyla potřeba výuku dělit na více skupin, jelikož zájemců bylo přibližně patnáct a hodiny proběhly pouze jednou. Opět jsme využili kooperační metody výuky a žáky rozdělili do skupinek po třech. V těch následně roboty konstruovali a programovali.

K úlohám jsou opět zdokumentované videosekvence, na které směřují odkazy v příloze.

### 3.2.1 Sada úloh v prostředí VEX code

#### Úloha číslo 1



Obr. 26 - Zadání 1. úlohy VEX GO

Zadání: Projed' od startu k cíli celým bludištěm bez jakéhokoli dotyku.

Popis: Tato úloha slouží k seznámení se základními příkazy. Na pravé straně autíčka jsou v linii vyskládané disky a hned naproti autíčku je stěna, kterou když objede, tak se dostane do cílové rovinky. Tato úloha, řekněme, je nastavena na jedno řešení. Žáci si vyzkouší měřit vzdálenost pomocí metru, a tím zjistí jakou vzdálenost má autíčko jet. Musí si uvědomit, že je výhodné měřit od předních kol autíčka. Jde tedy o testování, správné měření, nebo také zkusit možnost pokus omyl. Stěny nesmí být bliž než samotná šířka autíčka, aby byla úloha splnitelná. Za prvé autíčko ujede naměřenou vzdálenost dopředu,

následně se otočí doprava o 90 stupňů, jede opět rovně a odbočí vlevo o 90 stupňů. Následně rovně, doleva a rovně.

Pomocné otázky: Jak můžeš přistupovat k řešení úlohy? Jaké příkazy můžeš využít? Kdy se má autíčko zastavit a otočit? Jak určíš vzdálenost, kterou má autíčko ujet? Odkud budeš vzdálenost měřit? V kolika krocích minimálně zapíšeš tento program?

Řešení: Tato úloha má právě jedno řešení, pokud neuvažují malé odchylky ve vzdálenostech.



Obr. 27 - Řešení 1. úlohy VEX GO

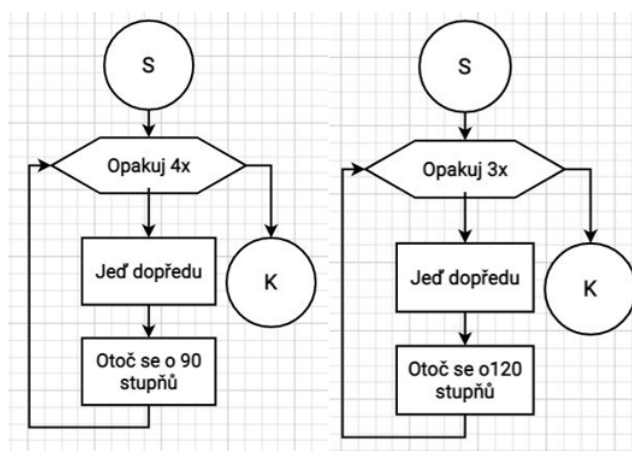
Poznámky: K zmenšení odchylek při pohybu je výhodné nastavit rychlost pohonu na 50 %, nebo i méně. Zároveň ono hřiště pro VEX 123 je celkem kluzké, což vede k prosmeknutí kol, a tudíž k ujetí nesprávné vzdálenosti. Potřebné pomůcky jsou autíčko Super Code Base a jakékoli překážky k vytvoření stěn bludiště.

## Úloha číslo 2

Zadání: Naprogramuj, aby autíčko jezdilo do tvaru čtverce a poté do rovnostranného trojúhelníku.

Popis: Tato úloha nevyžaduje žádné překážky, ale mohou se klidně přidat čtyři sloupy ve tvaru čtverce, které ve stejném tvaru objede. Autíčko jede dopředu o vzdálenost délky strany čtverce, následně se otáčí o 90 stupňů. Žák by si měl uvědomit, že by měl využít cyklu. Obdobně pro trojúhelník, když dojedu do vrcholu trojúhelníka, otočím se o rozdíl  $180 - 60$  stupňů, tedy o 120 a pokračuji v délce jeho strany.

Algoritmizace:



Obr. 28 - Algoritmy 2. úlohy VEX GO

Pomocné otázky: Jaké jsou vnitřní úhly čtverce? Co vše platí pro čtverec? Jak vypadá načrtnutí jedné strany s otočením a kolikrát se bude opakovat? Jaký cyklus využiji? Co platí pro rovnostranný trojúhelník? O jaký úhel se budu otáčet? Kolikrát cyklus proběhne?

Řešení:



Obr. 29 - Řešení 2. úlohy VEX GO

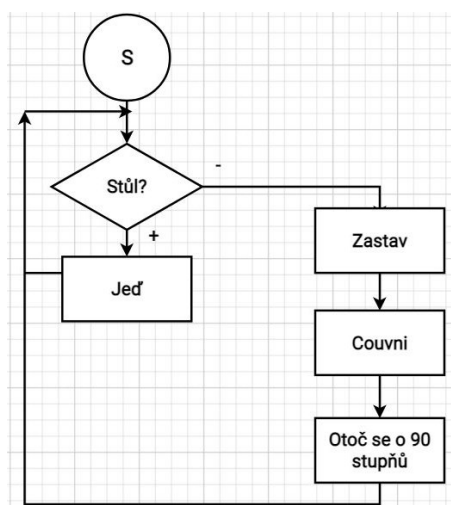
Poznámky: Autíčko dělá často chyby jak při otočení, tak i při posunu. Velmi záleží na materiálu, po kterém jezdí. Lepších výsledků se dostalo na koberci a hrubších površích. Potřebnou pomůckou je samotné autíčko Super Code Base.

### Úloha číslo 3

Zadání: Naprogramuj, aby robotické autíčko jezdilo po stole a nikdy z něho nespadlo dolů na zem.

Popis: K této úloze budeme potřebovat vyvýšený povrch, ze kterého by autíčko mohlo teoreticky spadnout. Naším úkolem je opak, a to, aby autíčko nespadlo. K samotnému autíčku budeme potřebovat oční senzor, který detekuje propast. Robota tedy postavíme směrem k hraně stolu a necháme ho vyjet. Při detekci srážu se autíčko otočí o libovolný úhel, nejlépe takový, aby se odtočil od srážu a nemusel ho detekovat znovu. Po detekci se autíčko zastaví a zacouvá o určitou vzdálenost, aby při otáčení nespadlo, a poté se otočí. Tato činnost se teoreticky může opakovat do nekonečna.

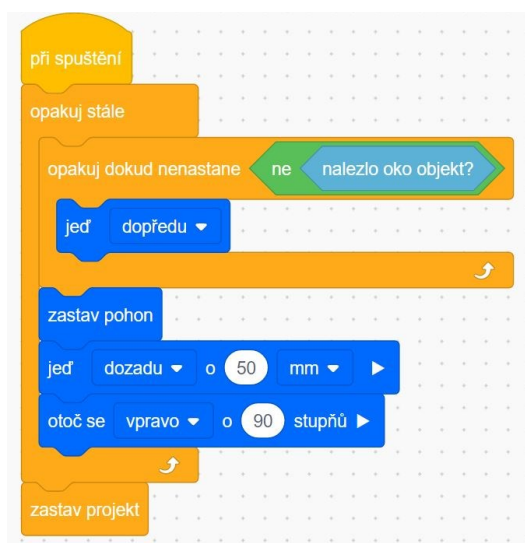
Algoritmizace:



Obr. 30 - Algoritmus 3. úlohy VEX GO

Pomocné otázky: Jak autíčko zastaví? Jaký senzor využiji pro detekci stolu a srážu? Bude lepší mít senzor od autíčka dál, nebo blíž? Co se má dít po detekci srážu?

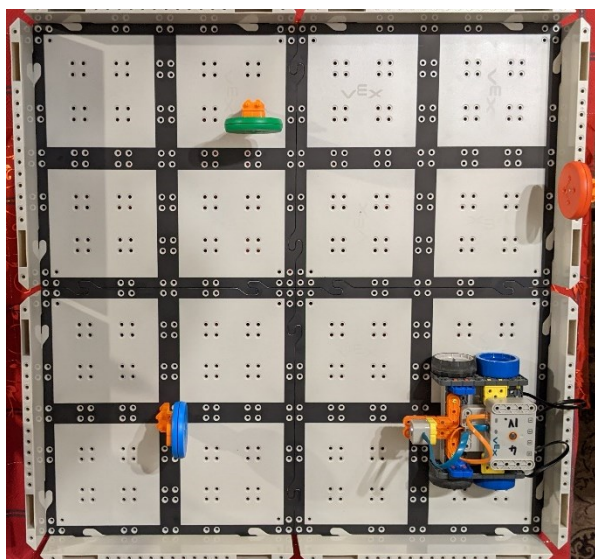
Řešení:



Obr. 31 - Řešení 3. úlohy VEX GO

Poznámky: Možným problémem tohoto úkolu mohou být rohy stolu, kde má autíčko větší pravděpodobnost pádu než na rovné hraně. Proto se senzor umístil na delší blok, aby oko dohlédlo co nejdále a mohlo na autíčko včas zareagovat. Potřebné pomůcky jsou robot Super Code Base, oční senzor a blok pro prodloužení konstrukce, aby oko dohlédlo dále.

#### Úloha číslo 4

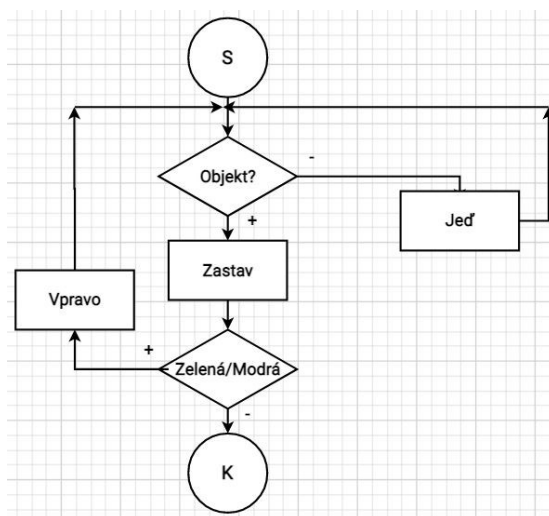


Obr. 32 - Zadání 4. úlohy VEX GO

Zadání: Autíčko musí projet po silnici do cíle k červené značce. Řid' se v dopravě podle barevných značek. Modrá a zelená značka značí odbočení vpravo.

Popis: V této úloze postavíme autíčko do pozice, kde bude mít před sebou modrou nebo zelenou značku. Značky musí být postaveny tak, aby se finálně autíčko zdárně dostalo do cíle. Proto je druhá značka po pravé straně a cílová červená taktéž, tedy vpravo od druhé značky. Jakmile auto detekuje značku, tak odbočí. Toto se opakuje do doby, než robot nalezne cílový objekt, kterým je červená značka.

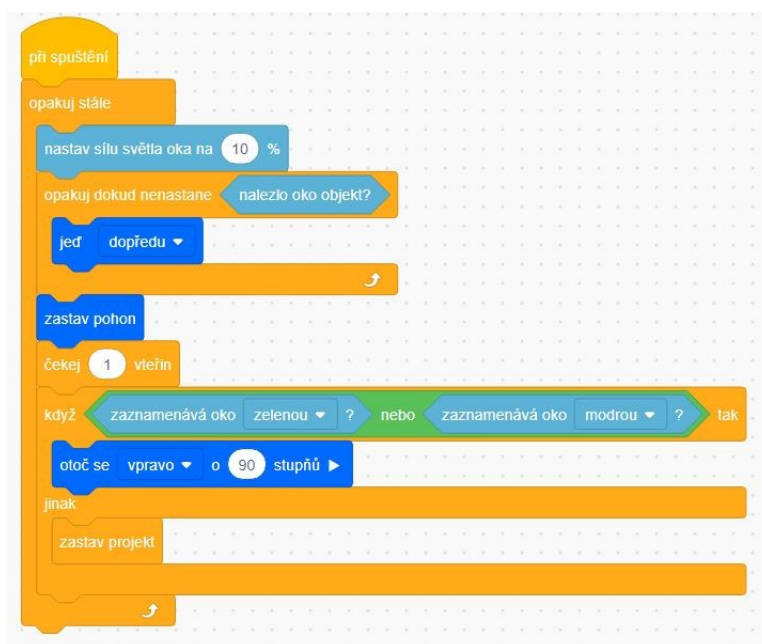
Algoritmizace:



Obr. 33 - Algoritmus 4. úlohy VEX GO

Pomocné otázky: Jak robot detekuje dané značky? Jaké senzory využije? Jaký cyklus použiješ pro jízdu robota a odbočení? Pokud oko značku detekuje příliš brzo, jaký příkaz použiješ, aby ji detekovalo později? Jakých chyb se vyvarovat? Jak autíčko zastaví?

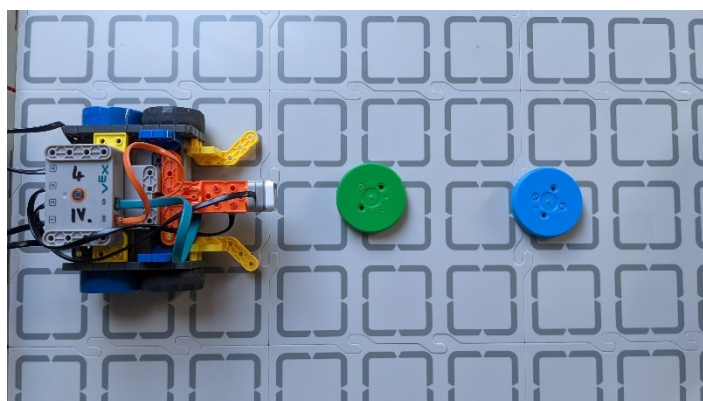
Řešení:



Obr. 34 - Řešení 4. úlohy VEX GO

Poznámky: Značky musí být ve správné výšce, aby senzor objekt a jeho barvu správně detekoval. Pokud byla moc nízko, vůbec ji nebral v potaz. Problém mohl nastat, pokud autíčko jelo přímo na střed značky, kde je kov pro magnet a senzor, tedy také by značku nepostřehl. Potřebné pomůcky jsou autíčko Super Code Base, disky, herní pole, bloky jako držák cedule.

## Úloha číslo 5

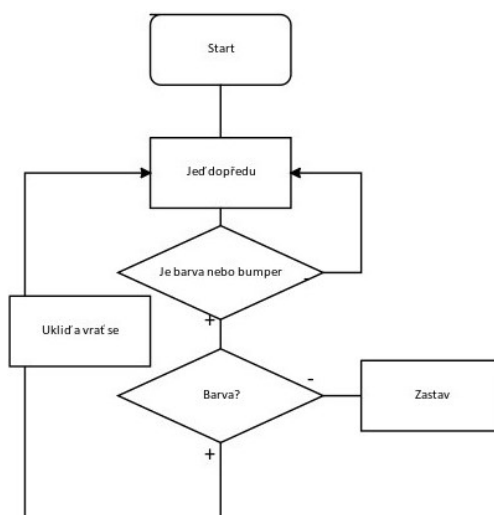


Obr. 35 - Zadání 5. úlohy VEX GO

Zadání: Uklid' cestu před sebou od barevných disků a každý disk přesuň na stranu. Zastav u protější strany.

Popis: V tomto úkolu se robotické autíčko se senzory postaví na začátek rovné dráhy, do které se umístí libovolný počet barevných disků. Disky nesmí být těsně u sebe, aby nedošlo k jejich náhodnému přemístění. Jakmile je detekována přítomnost barevného disku, musí být sepnut magnet a disk libovolným způsobem přenesen mimo dráhu. Dále se autíčko vrátí na místo detekce, natočí se ve směru jízdy a pokračuje dál. Tato činnost je opakována do doby, než je nalezena zeď. V tom okamžiku se musí robot zastavit a ukončit celý program.

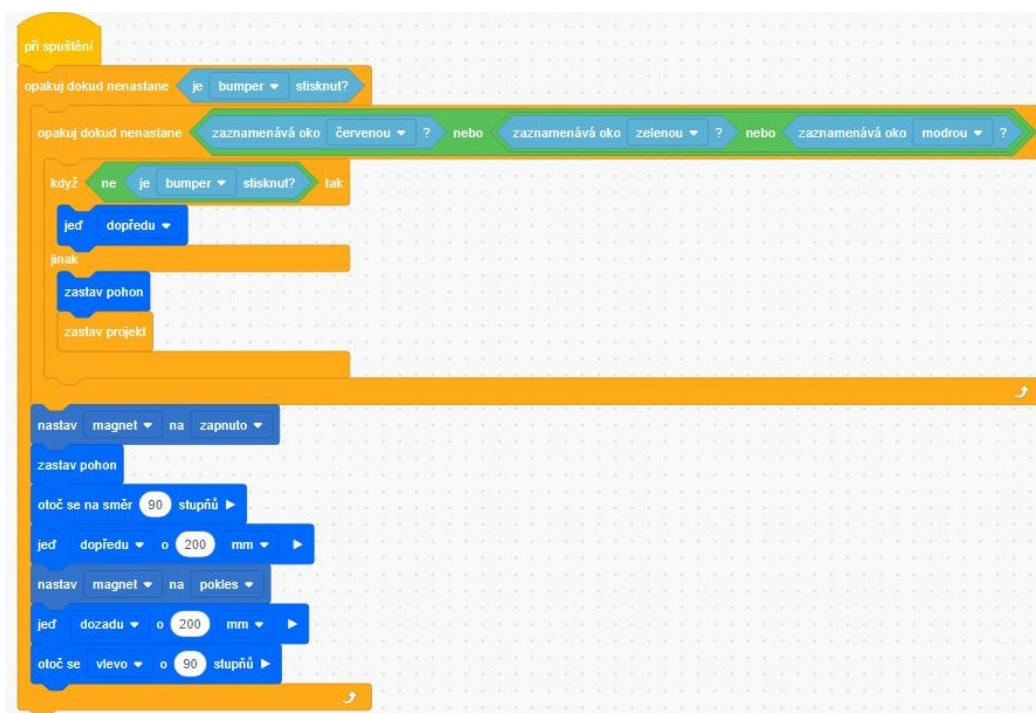
Algoritmizace:



Obr. 36 - Algoritmus 5. úlohy VEX GO

Pomocné otázky: Jak můžeš detekovat disky? Jak zvedneš a odložíš disky? Které senzory na to využiješ? Které události mají způsobit přerušeni jízdy robota? Jaký cyklus vybereš k jízdě autíčka a k jejímu přerušeni? Co vše musíš udělat, aby autíčko disk odklidilo z cesty? Jak můžeš zjistit mantinel? Jaké příkazy existují pro bumper a jaké využití má? Jaké případné chyby mohou nastat?

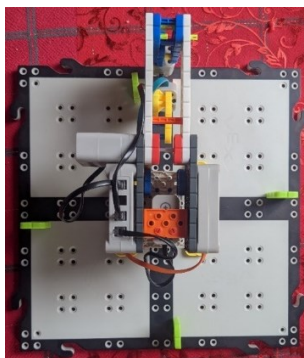
Řešení: Příklad možného řešení.



Obr. 37 - Řešení 5. úlohy VEX GO

Poznámky: Možnou variantou tohoto úkolu je hledání jednoho disku v herním poli a přenesení k mantinelu nebo přenos disků podle barvy např. k levému nebo pravému mantinelu. Potřebné pomůcky jsou herní pole se zvýšeným mantinelem, postavené autíčko Super Code Base a barevné disky.

## Úloha číslo 6



Obr. 38 - Zadání 6. úlohy VEX GO

Zadání: Postav robotické rameno a vyčisti desku od daných překážek. Následně rameno zvedni vzhůru a otoč se do pozice vpravo o 90 stupňů a rameno polož dolů.

Popis: V této úloze využijeme další typ robota. Bude jím robotické rameno, které se otáčí jak horizontálně, tak vertikálně a je připevněné v herní desce. V úloze jsou rozmístěny 4 překážky, které má rameno shodit. Výchozí poloha je taková, že je rameno nízko téměř položené při zemi. Žáci se seznamují se základními příkazy a zjišťují, že mohou otočit s base nebo arm.

Algoritmizace:



Obr. 39 - Algoritmus 6. úlohy VEX GO

Pomocné otázky: Jak se může robot pohybovat a jakým směrem? Jak pomocí ramene překážky odstraníš?

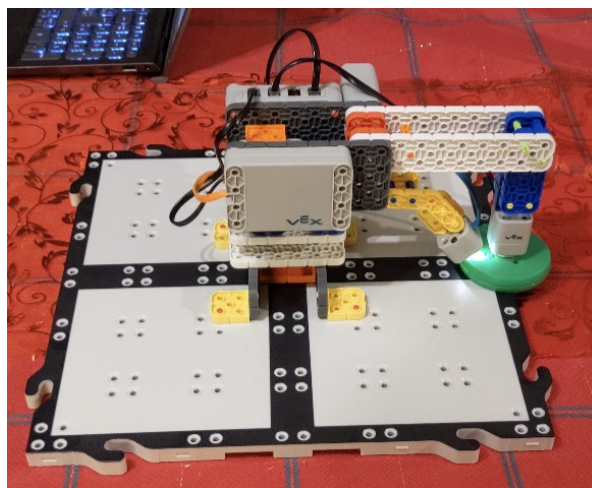
Řešení:



Obr. 40 - Řešení 6. úlohy VEX GO

Poznámky: Rameno při otočení vzhůru má odchytku, nedodrží úhel přesně 90 stupňů. Další variantou může být domino. Žáci si postaví standardní domino a pomocí ramene, mohou shodit první článek domina. Potřebné pomůcky jsou daný počet překážek v našem případě čtyři, robotické rameno s plochou.

### Úloha číslo 7



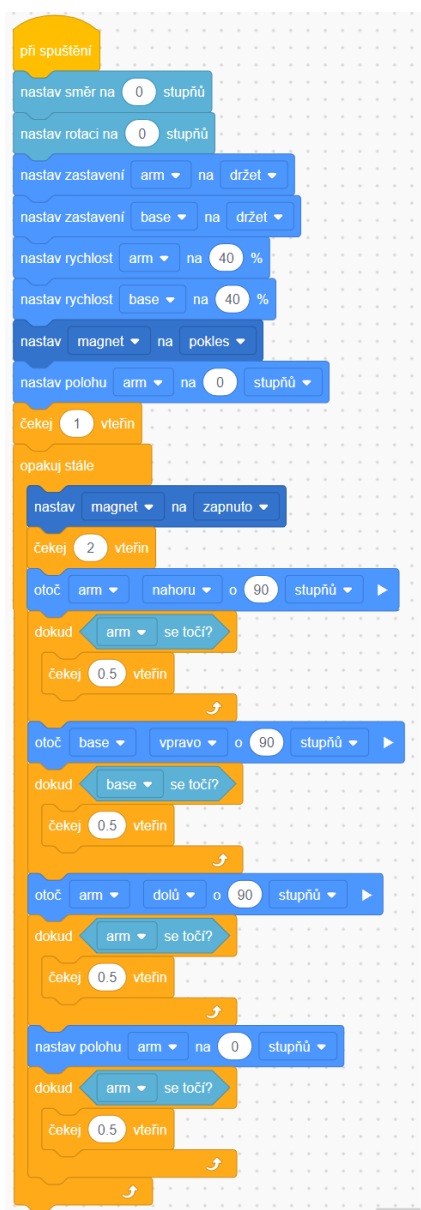
Obr. 41 - Zadání 7. úlohy VEX GO

Zadání: Opakuj zvednutí disku a jeho přesunutí vpravo o 90 stupňů.

Popis: Úloha měla být koncipována tak, že disk magnet nabere, následně se přesune o daný úhel a disk odloží. Jenže rameno se neotočí o správný úhel. Pouštělo disky občas z výšky a ty se odkutálely pryč, proto úlohu nebylo možné správně provést. Disk se tedy ve změněné úloze nepouštěl, ale byl stále přichycený na magnetu. Opět si žák prověří všechny pohybové příkazy a navíc si zopakuje příkazy pro magnet, které využil u robotického autíčka.

Pomocné otázky: Jaké činnosti rameno vykonává? Jak by ses snažil zabránit tomu, že rameno přejede námi požadovaný bod? Jaké části robota bys snížil rychlost? Co musíš nastavit, aby pouštěl a zvedal disky? Napadá tě způsob, jak by rameno mohlo čekat a nedělat chyby?

Řešení:



Obr. 42 - Řešení 7. úlohy VEX GO

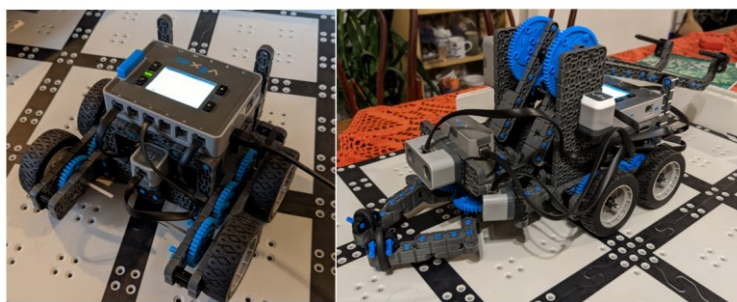
Poznámky: V tomto úkolu už byly téměř nepoužitelné chyby, které robot dělal. Proto se na začátku programu nastavila rychlost otáčení base i ramene na 40 %. Došlo k částečnému zlepšení, nicméně robot se např. nedotočil o správný úhel, v některých segmentech se choval zvláště sekaně, dopočítával úhly, a proto jsem zkusil čekání několik sekund.

### 3.2.2 VEX IQ

Tato robotická sada je na výrobcových webových stránkách uváděna pro žáky starší 11 let. Ačkoli si myslím, že už stavebnice VEX GO se plně uplatní pro žáky 9. tříd, a tedy by tato stavebnice mohla zasahovat až do středoškolského studia. Co se týče stavebnice, tak ta je velmi podobná verzi GO, akorát profesionálnější, stabilnější a celkově s menší chybovostí. V předešlé verzi se např. často protáčela kola a s IQ jsem podobné problémy neměl. Stavebnice VEX IQ tedy dokáže řešit stejné problémy jako verze GO. Odlišností je propojení robota s počítačem, ten se nepřipojuje pomocí bluetooth, ale přes USB kabel přímo do mozku robota k nahrání programu. Po nahrání programu je možné robota odpojit, jelikož vše zůstane nahráno v jeho paměti, a spustit program externě přímo tlačítky na robotovi.

Tato část práce je odlišná od ostatních sad, jelikož byla pouze jedna, a žáci neměli možnost s ní pracovat. Tudíž jde o zkompletování a ukázkou možných staveb a programů, nikoli vyloženě o otestování úkolů na samotných žácích. Zároveň konstrukce jednotlivých robotů je o dost časově náročnější a komplikovanější, a mohl by tedy nastat problém i z časových důvodů.

Tuto sadu lze využít ke stejným úlohám jako ve verzi GO a je možné postavit podobné autíčko s názvem Autopilot, které jsem také využil pro ukázkou jazyku ROBOTC. Dalším robotem je Clawbot IQ, který pomocí svých klepet dokáže zvedat různé předměty a nakládat je na svou rampu.



*Obr. 43 - Autopilot a Clawbot*

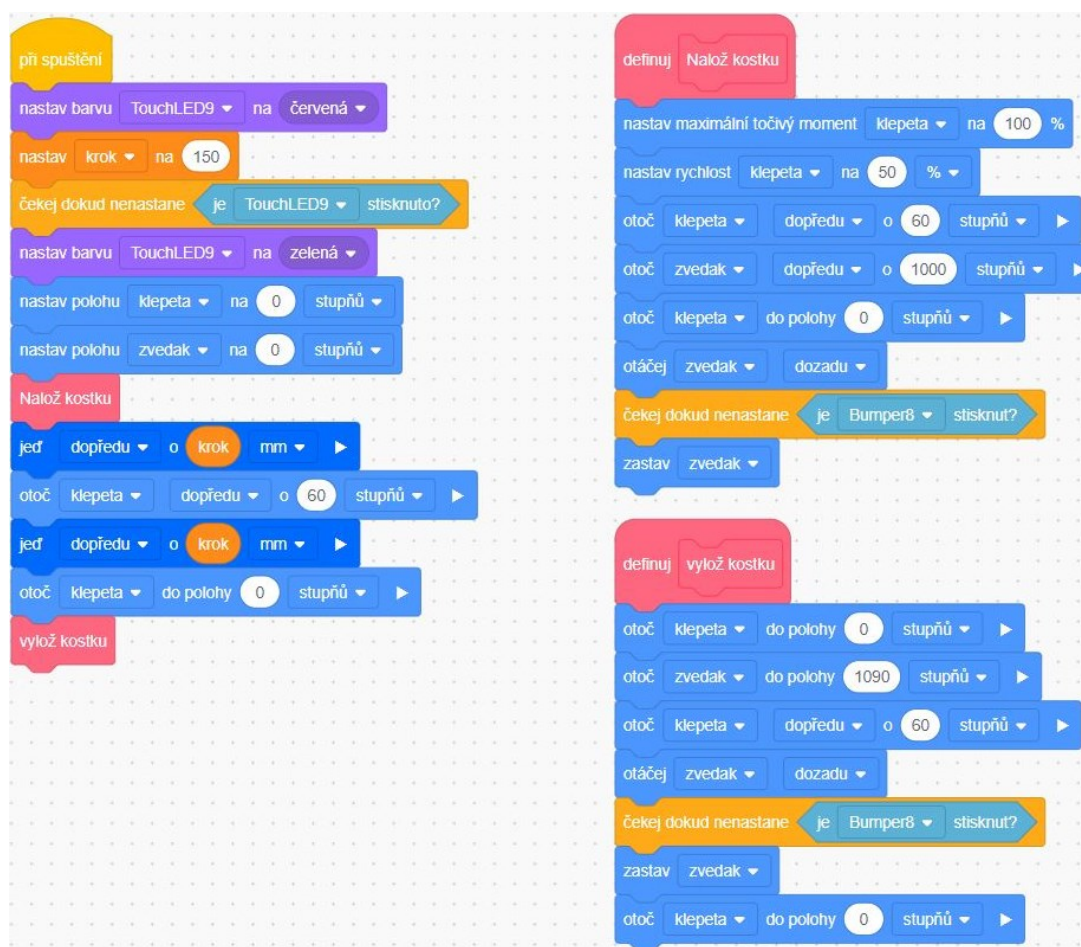
### 3.2.3 Sada úloh v prostředí VEX code

#### Úloha číslo 1

Zadání: Odstraň kostky, které jsou od sebe stejně vzdáleny, z cesty a přesuň je do propasti. První kostku naber na rampu a druhou kostku odlač do propasti. První kostku z rampy opět vezmi a také ji vyhod' za kostkou druhou.

Popis: V této úloze se seznámíme s Clawbotem. Tuto úlohu lze rozdělit na dvě části, zaprvé sebrání a naložení kostky na rampu robota. Odtlačení druhé kostky není nijak složité, a druhou částí je nabrání kostky z rampy a následné zahození. K tomu si vytvoříme vlastní bloky a zmenšíme tím velikost a počet opakování daných příkazů. Při propojení robota s prostředím si pojmenujeme jednotlivé motory na klepeta, pomocí kterých svírá objekty, a zvedák, jež využije pro zvednutí kostky na rampu. Výhodným prvkem je tlačítko TouchLED. Program se nespustí do té doby, než je toto tlačítko zmáčknuto, i přesto že je program v prostředí spuštěn. Při klesání zvedáku zjistíme, že je dole pomocí bumperu, který zvedák stiskne, a tím se přestane rameno pokládat.

Řešení:



Obr. 44 - Řešení 1. úlohy VEX IQ

Poznámky: V úloze je zásadní před spuštěním programu připravit klepeta do polohy nesevřeno a tu brát jako polohu 0 stupňů. Jinak by program nefungoval. Vysoká čísla u otočení zvedáku mohou být překvapivá. Nejedná se o úhel v prostoru, ale o úhel, o který se otáčí motor.

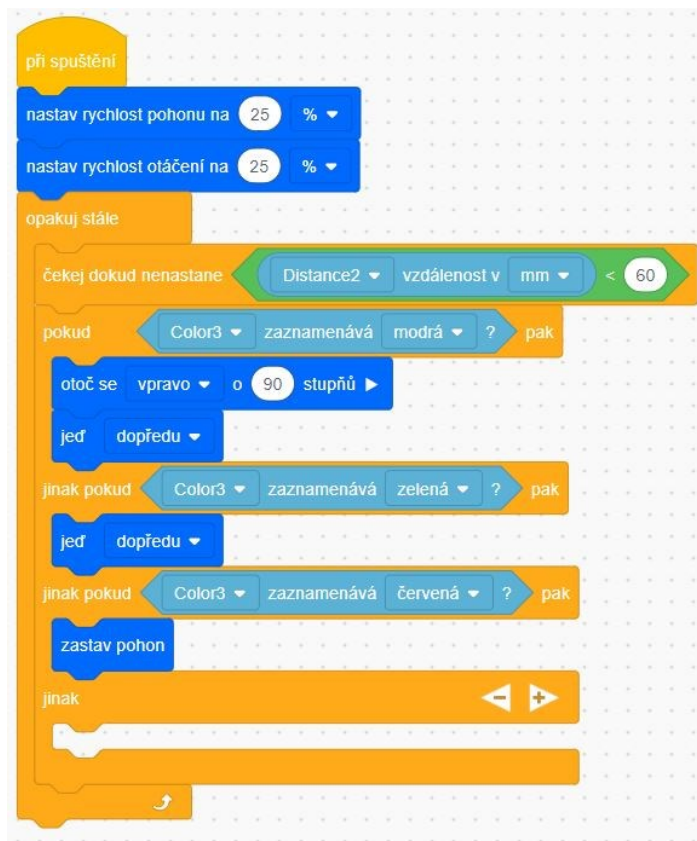
## Úloha číslo 2

Zadání: Ovládej robota pomocí barevné kostky. Zelená barva spustí jízdu dopředu, červená barva robota zastaví a modrá barva robota otočí doprava a následně spustí jízdu rovně.

Popis: V úloze využijeme robota Autopilot, který je lepší verzí autíčka Codebase ze stavebnice VEX GO. V úloze využíváme dva senzory, jeden pro vzdálenost, kterým určujeme, kdy má druhý senzor zjišťovat barvu na kostce a jestli je kostka v blízkosti

autíčka. Pomocí podmínek rozhodneme, jakou akci má robot provést. Možnou modifikací úlohy je upřesnit zadání tak, že musí autíčko někam dojet, vyhnout se překážce pomocí kostky.

Řešení:



Obr. 45 - Řešení 2. úlohy VEX IQ

Poznámky: Zásadním problémem může být světlo a vymalování místnosti. To může ovlivňovat námi nechtěným způsobem snímání barev na kostce.

### Úloha číslo 3

Zadání: Jezdí s robotem po kružnici. K programování využij jazyk RobotC.

Popis: Touto úlohou představíme kódování robota pomocí textového jazyka, kde můžeme např. ovládat rychlost jednotlivých motorů, a tím vytvářet křivky, nebo v našem případě kružnici. Jelikož jsou u robota motory v opačném směru, využijeme příkaz `setMotorReversed()`, kde jednomu motoru přiřadíme hodnotu `true` a druhému `false`, čímž oba pojedou ve stejném směru. Následně nastavíme různé rychlosti samotných motorů

příkazem `setMotorSpeed()`. Příkazem `wait1msec(10000)` se bude robot otáčet po dobu 10 vteřin.

Řešení:

```
#pragma config(StandardModel, "Standard Drive Base")
/**!!Code automatically generated by 'ROBOTC' configuration wizard      !!**//

task main()
{
  setMotorReversed(motor1, false);
  setMotorReversed(motor6, true);
  setMotorSpeed(motor1, 20);
  setMotorSpeed(motor6, 50);
  wait1Msec(10000);
}
```

*Obr. 46 - Řešení 3. úlohy VEX IQ*

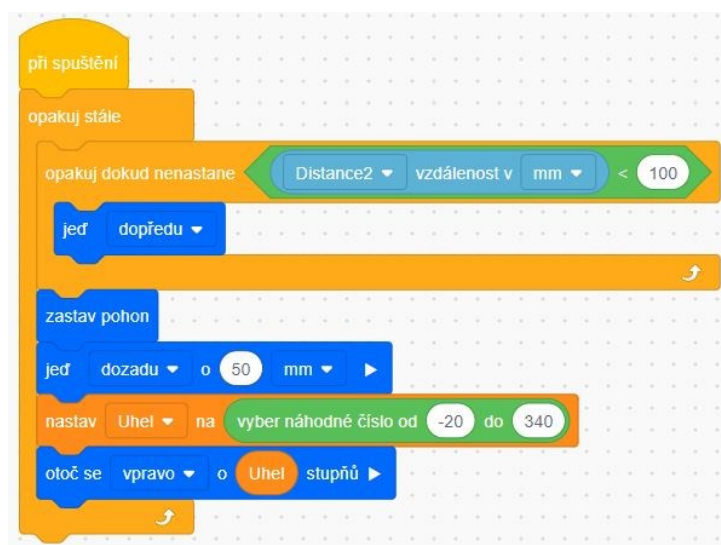
Poznámky: Žáci si mohou měnit rychlosti a zjišťovat, jak robot pojede, popřípadě udělat delší sekvenci příkazů a vymyslet taneční sestavu, nebo kreslit křivky.

Úloha číslo 4

Zadání: Naprogramuj robota tak, aby jezdil náhodně po místnosti a nezastavoval se u překážek.

Popis: V této úloze si ukážeme, jak robota naprogramovat, aby stále jezdil po místnosti náhodně zvoleným úhlem. Využijeme senzoru vzdálenosti, v našem případě jsem zvolil, jestli bude překážka blíže jak 100 milimetrů, tak zastaví, couvne a otočí se. Pro ukázkou vlastní proměnné jsem využil úhel, a tu jsem si také tak pojmenoval. Výběrem od -20 do 340 stupňů jsem se snažil snížit šanci identického otočení.

Řešení:



Obr. 47 – Řešení 4. úlohy VEX IQ

## 4 Analýza výstupů

### 4.1 1. Stupeň

#### Programování pomocí dotyku

S tímto typem kódování neměli žáci jakékoliv problémy. 90 % žáků vždy vědělo, jak k dané úloze přistupovat. Tedy vždy byla jedna skupina, která měla problémy, ale s dohledem a radami dosáhli správného výsledku. První úloha sloužila opět k seznámení se základním ovládáním, které je velmi intuitivní. V druhé úloze nastal problém ve špatném spočítání kroků k přejetí hřiště, nebo že se robot neotočil o 180 stupňů, ale o pouhých 90. Žáci se vždy opravili. Paradoxně nejtěžší úlohou byla úloha č. 3, kde bylo úkolem změřit, jakou vzdálenost ujede robot v jednom kroku a jak souvisí délka robota s délkou kroku. Žáci absolutně nevěděli, jak s metrem měřit od jakého místa kam. V této úloze jsme strávili nejvíce času. A po otázce, jestli spolu nějak souvisí délka robota a kroku, většinou přišla odpověď že „ano“, ale v některých případech „ne“ a to nebyla správná odpověď. Se čtvrtou úlohou opět měla vždy problém jenom jedna skupina, kde pomocnou radou bylo, aby si robotem po jednotlivých krocích posouvali a otáčeli a tím ho správně naprogramovali. Zajímavým pozorováním bylo, že jsem očekával, že zvolí schod o velikosti jednoho kroku, a tím se přesunou dál a vytvoří delší schodiště. Všechny skupiny zvolily větší měřítko, a tím vytvořily maximálně tři schody, ale zadání splnily. V 5. úloze nastal problém při vytvoření tvaru čtverce tak, že žáci zapomněli počet opakování jízdy a otočení. Čtverec tedy nebyl kompletní. Po splnění zadaného úkolu bylo druhým zadáním tvar obdélníku, kde většinou 3 skupiny nepochopily, jak volit délky obdélníku i po nakreslení tvaru a uvědomění si jeho vlastností. V 6. úloze neměli žáci jakékoliv problémy, jelikož se na škole používají frausovské učebnice, které rozvíjí počítání na ose podobným způsobem, jako je tato úloha. Předpokládal jsem, že úlohu vyřeší 8 kroky dopředu a 5 zpět. Žáci přišli s novým řešením a vlastně i jednodušším. Řekli si, že před robotem je 8 polí, takže pokud robot pojede 5 polí dopředu, tak zbydou tři políčka.

#### Programování pomocí karet

První skupina pracovala pomaleji a k řešení úloh došli s pomocí. Nikdo sám nepřišel s tím, jak nahradit cyklus příkazem go to start. Druhá skupina pracovala samostatně, po

některých otázkách došli k tomu, že se některé příkazy opakují, a mohou tedy využít cyklu. První úloha nedělala problémy, jelikož se jedná o pouhé seznámení s příkazy. Ve 2. úloze se poprvé měli seznámit s cykly, o kterých jsem mluvil na začátku hodiny. Nevím, jestli výklad nebyl dostatečný, nebo jestli jsem pro druhou skupinu problematiku vysvětlil lépe, ale mám pocit, že první skupina nedávala pozor, oproti tomu druhá byla potichu a poslouchala, a úlohu všichni smírnou radou zvládli. Opět někomu nebylo zpočátku jasné, jak je to s úhlem otočení. U 3 skupinek jsem se setkal s otočkou doprava a ihned doleva a předpokládali otočení, zvláštní pro mě bylo, že se tato chyba objevila u více skupin. Se 3. úlohou se žáci setkali již pomocí programování dotykem a samotnou sekvenci příkazu měli tedy správně. Po otázce, jestli se jim některé příkazy neopakují a jestli by nebylo možné použít cyklus, tedy příkaz `go to start`, tak většinou ihned přišla odpověď „ano“ a program si opravili. Skupiny, které úlohu zvládly bez problému, řešily trasu pro obdélník. Víceméně opět bez problému všichni přišli s řešením. U 4. úlohy při objíždění hřiště podél stěny jsem očekával, že použijí příkaz `drive until object`. Všichni si napočítali, že robot pojede 7 bloků dopředu a pak odbočí. Úkol jsem bral tedy jako správně, ale upravil jsem tuto úlohu, aby to platilo pro libovolný čtyřúhelník. Po otázce jak pojedou a nenarazím do stěny, jestli ji mohou nějakým způsobem detekovat, jsme došli k řešení. S pátou úlohou přišlo hned několik problémů. Žáci nechápali smysl podmínek, až na jednu skupinu nikdo nepřišel sám na řešení. Tudíž první modifikací úlohy by byla snazší úloha, např. pokud modrá barva jeď rovně, kde by se seznámili se základní podmínkou bez více variant. Druhou překážkou byl samotný senzor robota, který reagoval pomalu a také nepřesně. Modrou barvu vzal jako zelenou atd. Skoro všem skupinkám se robot z ničeho nic začal otáčet a snímat zelenou barvu. Předpokládám, že to bylo tím, že jsme byli v zeleně namalované místnosti a tato barva se odrážela. Tudíž úlohu provádět v kvalitně osvětleném prostoru bez okolních barev.

### **Programování v prostředí VEX code**

Hlavním problémem byla technologie Bluetooth, která nám práci s 1. a 2. úlohou mírně zkomplikovala. Žáci programovali na stolních počítačích. Dosah byl tak krátký, že došlo k odpojení i při menším posunu po stole. Využít hřiště tedy nebylo možné, práce tedy zůstala u stolů a pouze si představovali situaci na hřišti. Zadání bylo promítáno na plátno.

K programování v prostředí VEX code proběhla dvě setkání, jelikož vysvětlit jednotlivé příkazy, popsat prostředí, propojení robota a lehce vysvětlit ostatní cykly bylo časově náročnější než ostatní možnosti programování. Překvapením pro mě bylo, že už nastal problém při zapsání URL adresy této aplikace do webového prohlížeče. Žáci se dostali do jiného prostředí nebo na stránky výrobce, nebo úplně jinam. Obecně se mi po několika setkáních potvrdilo, že druhá skupina pracovala lépe. Problémem může být žák, který jako malý prožil domácí násilí a někdy měl problémy jakkoliv vnímat, neposlouchal a byl roztěkaný. Nebylo to však pravidlem, žák je velmi šikovný na matematiku, a pokud se k jeho stavům nedostalo, pracoval velmi dobře. Tímto se jedním žákem rozpadala pozornost celé skupiny, nebo aspoň tak to působilo. První úlohu většina skupin zvládla na první pokus téměř ihned, problém byl onen žák, který nebyl ani schopný podívat se na plátno a nevěděl, kam má robot jet. Druhá úloha se mi velmi líbila a řekl bych, že i žákům. Bystřejší skupinky se také musely zamyslet a popřemýšlet. Opět jako vždy polovinu skupin nenapadlo použít opakování, a tedy vypisovaly celý program po jednotlivých příkazech za sebou. Po otázce co se opakuje? Prošel jsem program, “jede dopředu, pak zpět, doleva atd.“, už si žáci všimli, které příkazy se opakují a kolikrát. Zajímavým pozorováním bylo, že jenom jednoho žáka napadlo couvat, ale většina žáků se otáčela o 180 stupňů a pak jeli rovně a většinou pak nedokázali s cyklem dojít k správnému řešení. Po možnosti couvat úlohu zvládli bez problému. S první skupinou jsme bonus s navštívením rohů nestihli, ale druhá skupina úlohu zvládla bez problému a přišli s novým řešením.

Na druhém setkání první skupina dodělala bonus druhé úlohy. Následně jsem dodal úlohu na čtverec a obdélník, aby byla kompletní pro všechny možnosti programování. Ta už nedělala problémy, v některých případech byl problém počet opakování pro obdélník. Prošli jsme 3. a 4. úlohu, ale 5. úlohu jsme nestihli a přesunul jsem ji na ještě jedno setkání s malým testem. Třetí úlohu jsem pojal jiným způsobem, žákům jsem rozdál papíry a ukázal kód, který si mohli opsat, a sledovat, jak ovlivňuje robota po spuštění, pokud budu s papírem otáčet. Všichni si všimli, že se robot otáčí, někteří dodali v protisměru a také, že se snaží vrátit do pozice, v které začínal. Po otázce, jakou součástku robot v tu chvíli využívá, tak gyroskop nikoho nenapadl a nikdo slovo ani neslyšel. Čtvrtá úloha proběhla kreativně. Žáci si postavili své herní pole k hraní kostek a následně hráli kostky ve

dvojicích, až jsme postupně našli pavoukovým stylem výherce. Bohužel úroveň angličtiny nebyla dostačující a blok pro náhodný výběr nikdo nenašel. Docházelo k volení různých čísel např. od 360 do 890, což vlastně nebylo špatně, jen asi byla narušená pravděpodobnost výhry, a tak by měl být rozdíl těchto dvou čísel 360 stupňů jako jedna otočka.

#### **4.1.1 Kvantitativní výzkum**

Součástí poslední hodiny byl nevelký kvantitativní výzkum, který měl zjistit prekoncepty u žáků 1. stupně před zahájením práce s roboty, jejich zkušenosti, postoj k výuce s robotickými sadami a jejich zájem o danou problematiku. Dotazník byl vytvořen v aplikaci Google Forms, k němuž se žáci dostali pomocí jim dostupného odkazu a dotazník vyplnili. Celkem bylo z obou skupin 38 respondentů. Dotazník s grafy viz příloha.

První otázkou zhodnot' na škále 1 – 5 obtížnost úloh, bylo 16 odpovědí pro snadné, 11 pro středně obtížné, 8 pro velmi snadné, 2 pro velmi obtížné a 1 pro obtížné. Druhou otázkou, jestli se někdy setkali s nějakými roboty, jich celkově 29 vybralo možnost ano a 9 ne. Třetí otázkou jaký typ programování byl pro ně složitý, 22 žáků odpovědělo v prostředí VEX CODE na počítači, 11 mělo problém s programováním pomocí dotyku, což pro mě bylo zajímavé zjištění. Očekával bych, jak je sada úloh sestavena, že programování pomocí dotyku je nejjednodušší variantou. Zbýlých 5 odpovědělo pomocí kódovací desky. Čtvrtou otázkou, jestli se má podle tebe ovládání roboto ve škole vyučovat, z nich 28 souhlasilo s včleněním do výuky, 10 žáků bylo proti. Pátá a poslední otázka zjišťovala jejich zájem o pokračování v práci s roboty, kde 29 chce nadále pokračovat a 9 pokračovat nechce.

## **4.2 2. stupeň**

První úloha začala komplikacemi týkajícími se udržení připojení Bluetooth na větší vzdálenost robota od počítače a nedostatkem hracích polí. V učebně bylo pouze jedno hrací pole, které nemohli používat všichni naráz. Pokud by ho používali postupně, došlo by k velkému časovému zpoždění. Úloha byla modifikována na přesně definovaný slalom na čas mezi nohami židle, kterou měla každá skupina k dispozici. Tím se z úlohy stal závod na čas a vyhrál ten, kdo našel nejefektivnější cestu a našel nejlepší rychlost motorů. Při

stoprocentní rychlosti se na linoleu kola robota protáčela. Kromě jedné skupiny úlohu splnili všichni.

Ve druhé úloze (objed' čtverec a rovnostranný trojúhelník) nebyla ani jedna skupina schopna určit správný úhel otočení u trojúhelníku. Někteří volili 60 stupňů, jiní zadávali úhel odhadem a zkoušeli, jestli došli k správnému výsledku. Správnou velikost úhlu jsme určili během společné diskuse spojené s nákresem. Možnosti cyklu si nevšimla žádná skupina, pravdou je, že s touto možností nebyli seznámeni. Nicméně si povšimli, že se některé příkazy několikrát opakují. Po upozornění na možnost použití cyklu, všichni tuto možnost aplikovali. Možnou modifikací by bylo užití vlastních bloků k zjednodušení programu, nebo kombinací sekvencí bloků.

Úlohu č. 3 nebyli žáci bez vytvoření společného algoritmu schopni naprogramovat. Druhým problémem bylo nastavení délky ramene s optickým senzorem. Téměř všichni zvolili příliš krátkou délku, která nestačila k tomu, aby senzor zaregistroval konec stolu. Jeden z robotů se zřítil a rozpadl se.

Ačkoliv se ve čtvrté úloze jednalo o hru s hřištěm, nebylo nutné, aby měli všichni hřiště k dispozici. Dokud neměli úlohu naprogramovanou. V této úloze se občas vyskytl problém se správnou identifikací barvy. Modifikací úlohy je odbočení vpravo na modrou barvu a vlevo na zelenou.

V páté úloze bylo nutné sestavit společně algoritmus, neboť žáci nebyli schopni samostatně tuto úlohu vyřešit. Dvě skupiny byly schopny splnit úlohu s velkou výpomocí a pouze dvě ji vyřešily samostatně. Možnou modifikací je přesouvání kotoučů na různá místa podle barev.

Pro šestou úlohu museli žáci postavit druhý typ robota a tím bylo robotické rameno. Stavba byla o dost náročnější jak obtížností, tak časově. Pomocí této úlohy se seznámili se základními ovládacími příkazy a úlohu zvládli bez problému.

Jak se ukázalo, sedmou úlohu nebylo možné zrealizovat vůbec, z důvodu řady chyb otočného ramene. Jednalo se o to, že rameno se výškově nevracelo směrem k disku a stejně tak stranově. Disk při brzdění občas ulétnul. Během žákovských pokusů se potvrdila velká chybovost ramene.

## **Závěr**

Se vstupem výuky robotiky do vzdělávacího procesu se otevřela celá řada metodicko-didaktických problémů spojených s tímto inovativním oborem. Tato bakalářská práce se snaží malým dílem přispět k řešení této problematiky.

Hlavním cílem bylo sestavení sady úloh, týkajících se seznámení s edukačními robotickými sadami, s jejich prozkoumáním a objevováním v prostředí volnočasových útvarů. Vzhledem k možnostem byly zvoleny stavebnice VEX Robotics, které jsou v českém prostředí nové a neodzkoušené. Dalším cílem bylo realizovat jednotlivé úlohy s dětmi. Tato část byla rozdělena na dvě kategorie 1. stupeň a 2. stupeň. Třetím cílem bylo provést analýzu robotických aktivit a na jejím základě stanovit vhodné úpravy, nebo varianty zadání.

Celkem bylo vytvořeno 27 úloh, z nichž bylo 16 pro 1. stupeň a 11 pro druhý. První stupeň využíval robotickou stavebnici VEX 123, která je navržena pro žáky mladšího školního věku. Úlohy jsou koncipovány tak, aby se žáci postupně seznamovali se základními příkazy k ovládání robota. Další fází bylo objevování možnosti náhrady více stejných kroků cyklem. Poslední oblastí, kterou děti zkoumaly, byly senzory a větvení programů. Pro žáky 2. stupně kromě předchozího, úlohy obsahovaly i stavbu robota, tvorbu vlastních bloků, příkazy k ovládání pohybu robotické ramene a programování robota v textovém režimu jazyku RobotC.

Pokud by se měla shrnout analýza výstupů, tak z dotazníku pro první stupeň můžeme poznat, že si žáci výuku užili a byli spokojeni. Obtížnost pro ně nebyla nezvladatelná, ba naopak snadná. Pokud přihlížíme na způsob výuky, který jsem vedl. Skupinky jsem nechal pracovat a postupně je obcházel a sledoval chyby jako nevyužití cyklu, a s otázkou, jestli se něco v programu neopakuje, přišla většinou oprava. Úlohy splnili všichni žáci, některé skupiny bez pomoci. Pro druhý stupeň můžeme spekulovat, zda úlohy nebyly zvoleny správně. Byly úlohy, které nikdo nevyřešil, a až po společném sepsání algoritmu došli k výsledku. Samotné hodiny byly časově vytiženější kvůli náročnosti úloh i stavbě robotů.

Pokud bych měl shrnout modifikace jednotlivých úloh, tak pro 1. stupeň bych zásadní změny nedělal, jelikož výuka a seznámení proběhlo podle mých představ a úlohy dle mého

splnily svůj účel seznámení s daným problémem. Podle vyjádření žáků v dotazníku týkající se obtížnosti úloh se ukázalo, že považovali úlohy za jednoduché. Proto můžeme zvážit přidání několika složitějších zadání. Naopak na druhém stupni byly některé úlohy příliš složité a žáci je nebyli schopni bez větší pomoci vyřešit. Zřejmě by byla potřeba přidat několik snazších zadání.

Pokud bychom mluvili o rozšíření této práce, jednalo by se především o doplnění úloh pro VEX IQ, především o programování ROBOTC, které bylo v této práci zmíněno marginálně. Pro nadanější jedince by nebylo od věci přidat složitější úlohy typu procházení bludištěm a takových, u kterých je vyžadováno nejen řešení, ale i jeho optimalizace.

## Seznam použitých informačních zdrojů

ALIMISIS, Dimitris, ed. *Teacher Education on Robotics-Enhanced Constructivist Pedagogical Methods*. Athény: New Technologies Publications, 2009. ISBN 978-960-6749-49-0.

Bee-Bot Včelka. *Vyuka-vzdelavani* [online]. Brno: MORAVIA Consulting spol. s r.o., 2022 [cit. 2022-03-10]. Dostupné z: <https://www.vyuka-vzdelavani.cz/bee-bot-vcelka.html>

Bee-Bot. *Moravia.education* [online]. Brno: MORAVIA Consulting spol. s r.o., 2022 [cit. 2022-03-10]. Dostupné z: <http://www.moravia.education/cz/clanek/17-bee-bot>

Co je STEM?. *JEDUEDU!* [online]. Libeznice: STEM Education, 2022 [cit. 2022-03-29]. Dostupné z: <https://www.jeduedu.cz/stem/>

Coding with the Touch Buttons on the 123 Robot. *VEX* [online]. Greenville: VEX Robotics, 2022 [cit. 2022-03-04]. Dostupné z: <https://kb.vex.com/hc/en-us/articles/360055747492-Coding-with-the-Touch-buttons-on-the-123-Robot>

Get Started with a GO Kit (with Storage). *VEX* [online]. Greenville: VEX Robotics, 2022 [cit. 2022-03-04]. Dostupné z: <https://kb.vex.com/hc/en-us/articles/360049144612-Get-Started-with-a-GO-Kit-with-Storage->

Get Started with a GO Kit. *VEX* [online]. Greenville: VEX Robotics, 2022 [cit. 2022-03-04]. Dostupné z: <https://kb.vex.com/hc/en-us/articles/360049144232-Get-Started-with-a-GO-Kit>

Kdo vymyslel slovo robot? Karel Čapek to nebyl!. *Factory Automation* [online]. Praha: FANUC Czech, 2014 [cit. 2022-03-16]. Dostupné z: <https://factoryautomation.cz/kdo-vymyslel-slovo-robot-karel-capek-to-nebyl/>

Key Ideas for Building with VEX GO. *VEX* [online]. Greenville: VEX Robotics, 2022 [cit. 2022-03-06]. Dostupné z: <https://kb.vex.com/hc/en-us/articles/360049869692-Key-Ideas-for-Building-with-VEX-GO>

Kooperativní výuka. *Metodický portál RVP* [online]. Praha: Národní pedagogický institut České republiky, 2022 [cit. 2022-04-11]. Dostupné z: [https://wiki.rvp.cz/Knihovna/1.Pedagogick%C3%BD\\_lexikon/K/Kooperativn%C3%AD\\_v%C3%BDuka](https://wiki.rvp.cz/Knihovna/1.Pedagogick%C3%BD_lexikon/K/Kooperativn%C3%AD_v%C3%BDuka)

MIROSLAV, VIRIUS. *Základy algoritmizace* [online]. Vyd. 2. Praha: Česká technika - nakladatelství ČVUT, 2008 [cit. 2022-04-05]. ISBN 978-80-01-04003-4. Dostupné z: <http://www.jaderny-prvak.8u.cz/wp-content/uploads/2013/02/Z%C3%A1klady-algoritmizace-skripta-21.pdf>

New LEGO® MINDSTORMS® Robot Inventor lets creators build and bring to life anything they can imagine. *LEGO* [online]. Billund: The LEGO Group, 2022 [cit. 2022-04-01]. Dostupné z: <https://www.lego.com/en-us/aboutus/news/2020/june/lego-mindstorms-robot-inventor/>

Organizační formy výuky - Kooperativní výuka. *Škola Populo* [online]. Ostrava: Vzdělávací centrum Populo, 2022 [cit. 2022-04-11]. Dostupné z: <https://www.skolapopulo.cz/blog/organizacni-formy-vyuky-kooperativni-vyuka>

Ozobot bit. *Ozobot ve výuce* [online]. Lipnice: Jednota školských informatiků, 2022 [cit. 2022-03-19]. Dostupné z: <http://ozobot.sandofky.cz/zaciname/ozobot-bit/>

Ozobot evo. *Ozobot ve výuce* [online]. Lipnice: Jednota školských informatiků, 2022 [cit. 2022-03-19]. Dostupné z: <http://ozobot.sandofky.cz/zaciname/ozobot-evo/>

Pieces in the VEX GO Kit. *VEX* [online]. Greenville: VEX Robotics, 2022 [cit. 2022-03-05]. Dostupné z: <https://kb.vex.com/hc/en-us/articles/360047976292-Pieces-in-the-VEX-GO-Kit>

PROKEŠOVÁ, Miriam. *Obecná pedagogika*. Ostrava: Ostravská univerzita, Pedagogická fakulta, 2018. ISBN 978-80-7599-056-3.

ROBOMATTER INC. *ROBOTC for VEX Robotics 4.X - Users Manual* [software]. 2022 [cit. 2022-04-12]. Požadavky na systém: Win 7, Win8, Win10.

SKALKOVÁ, Jarmila. *Obecná didaktika*. 2. vyd. Praha: Grada Publishing, 2007. ISBN 978-80-247-1821-7.

ŠOLC, František a Luděk ŽALUD. *Robotika* [online]. Brno: FEKT, 2002 [cit. 2022-03-12]. Dostupné z: <http://media1.wgz.cz/files/media1:5100dca52f8f1.pdf.upl/Robotika.pdf>

TOCHÁČEK, Daniel a Jakub LAPEŠ. *Edukační robotika*. Praha: Pedagogická fakulta Univerzity Karlovy, 2012. ISBN 978-80-7290-577-5.

Using the VEX 123 Robot. *VEX* [online]. Greenville: VEX Robotics, 2022 [cit. 2022-03-04]. Dostupné z: <https://kb.vex.com/hc/en-us/articles/360055392011-Using-the-VEX-123-Robot>

Using the VEX Coder. *VEX* [online]. Greenville: VEX Robotics, 2022 [cit. 2022-03-04]. Dostupné z: <https://kb.vex.com/hc/en-us/articles/360055037612-Using-the-VEX-Coder>

VEX Coder Card Reference Guide. *VEX* [online]. Greenville: VEX Robotics, 2022 [cit. 2022-03-04]. Dostupné z: <https://kb.vex.com/hc/en-us/articles/360055264691-VEX-Coder-Card-Reference-Guide>

ZNÁM, Štefan. *Pohľad do dejín matematiky: celoštátna vysokoškolská príručka*. Bratislava: Alfa, 1986. Edícia matematicko-fyzikálnej literatúry.

## **Obrázky**

Super Code Base Car a Code Robot Arm. In: *VEX* [online]. Greenville: VEX Robotics, 2022 [cit. 2022-04-10]. Dostupné z: <https://www.vexrobotics.com/go/downloads/build-instructions>

VEX GO Parts poster. In: *VEX* [online]. Greenville: VEX Robotics, 2022 [cit. 2022-04-10]. Dostupné z: <https://content.vexrobotics.com/vexgo/poster-translations/EnglishPoster092120.pdf>

## Seznam obrázků

Obr. 1 - Základní prvky vývojových diagramů .....	16
Obr. 2 - Robot VEX 123.....	19
Obr. 3 - Super Code Base Car a Code Robot Arm .....	22
Obr. 4 - Součástky VEX GO .....	24
Obr. 5 - Pracovní plocha ROBOTC .....	27
Obr. 6 - Nabídka Robot pro program ROBOTC .....	27
Obr. 7 - Hodnoty barev ve VEX IQ.....	32
Obr. 8 – Zadání 2. úlohy s hracím polem .....	35
Obr. 9 - Zadání 6. úlohy s číselnou osou .....	38
Obr. 10 - Algoritmus 2. úlohy pro jízdu tam a zpět.....	39
Obr. 11 - Algoritmus 3. úlohy pro jízdu do čtverce .....	40
Obr. 12 - Algoritmus 4. úlohy pro objezd hřiště .....	41
Obr. 13 - Zadání 5. úlohy .....	42
Obr. 14 - Algoritmus 5. úlohy pro objetí robota.....	42
Obr. 15 - Zadání 1. úlohy .....	43
Obr. 16 - Řešení 1. úlohy.....	44
Obr. 17 - Zadání 2. úlohy .....	44
Obr. 18 - Algoritmus 2. úlohy a bonus .....	45
Obr. 19 - Řešení 2. úlohy.....	46
Obr. 20 – Řešení 3. úlohy .....	47
Obr. 21 - Zadání 4. úlohy .....	47
Obr. 22 - Řešení 4. úlohy.....	48
Obr. 23 - Zadání 5. úlohy .....	48
Obr. 24 - Algoritmus 5. úlohy .....	49
Obr. 25 - Řešení 5. úlohy.....	50
Obr. 26 - Zadání 1. úlohy VEX GO .....	51
Obr. 27 - Řešení 1. úlohy VEX GO.....	52
Obr. 28 - Algoritmy 2. úlohy VEX GO .....	53
Obr. 29 - Řešení 2. úlohy VEX GO.....	53

Obr. 30 - Algoritmus 3. úlohy VEX GO .....	54
Obr. 31 - Řešení 3. úlohy VEX GO.....	55
Obr. 32 - Zadání 4. úlohy VEX GO .....	55
Obr. 33 - Algoritmus 4. úlohy VEX GO .....	56
Obr. 34 - Řešení 4. úlohy VEX GO.....	57
Obr. 35 - Zadání 5. úlohy VEX GO .....	57
Obr. 36 - Algoritmus 5. úlohy VEX GO .....	58
Obr. 37 - Řešení 5. úlohy VEX GO.....	59
Obr. 38 - Zadání 6. úlohy VEX GO .....	59
Obr. 39 - Algoritmus 6. úlohy VEX GO .....	60
Obr. 40 - Řešení 6. úlohy VEX GO.....	61
Obr. 41 - Zadání 7. úlohy VEX GO .....	61
Obr. 42 - Řešení 7. úlohy VEX GO.....	63
Obr. 43 - Autopilot a Clawbot.....	64
Obr. 44 - Řešení 1. úlohy VEX IQ .....	66
Obr. 45 - Řešení 2. úlohy VEX IQ .....	67
Obr. 46 - Řešení 3. úlohy VEX IQ .....	68
Obr. 47 – Řešení 4. úlohy VEX IQ .....	69

# Přílohy

## Videa k jednotlivým úlohám

Programování pomocí dotyku

1. [https://drive.google.com/file/d/1F2ykn8jTIZ3-8Q9hLh\\_QSJloy471mJ42/view?usp=sharing](https://drive.google.com/file/d/1F2ykn8jTIZ3-8Q9hLh_QSJloy471mJ42/view?usp=sharing)
2. [https://drive.google.com/file/d/17-uSG\\_oUJBpeu-EmsSP-YSe9zraxiETC/view?usp=sharing](https://drive.google.com/file/d/17-uSG_oUJBpeu-EmsSP-YSe9zraxiETC/view?usp=sharing)
3. <https://drive.google.com/file/d/1CEaM-86yuCp4XdpRO0t2MzstZIKHVGlg/view?usp=sharing>
4. [https://drive.google.com/file/d/19-c\\_SdEB5t8aFHNgnD40R\\_pAjUME35ET/view?usp=sharing](https://drive.google.com/file/d/19-c_SdEB5t8aFHNgnD40R_pAjUME35ET/view?usp=sharing)
5. <https://drive.google.com/file/d/1BZhcVhfZiJKNWhb0SSXYczmkIjSun2L6/view?usp=sharing>
6. [https://drive.google.com/file/d/1N\\_rvdocuYC1vedIcW0VCc7iUZxjR0V3b/view?usp=sharing](https://drive.google.com/file/d/1N_rvdocuYC1vedIcW0VCc7iUZxjR0V3b/view?usp=sharing)

Programovací karta

1. [https://drive.google.com/file/d/1FZ97NzBbDU2ylb-Nw\\_IOnlXHCqhWg6sj/view?usp=sharing](https://drive.google.com/file/d/1FZ97NzBbDU2ylb-Nw_IOnlXHCqhWg6sj/view?usp=sharing)
2. <https://drive.google.com/file/d/1TM3imdaPtmGZBR9pBOiS6fhs8yBI1AmD/view?usp=sharing>
3. [https://drive.google.com/file/d/1TO0lIaSExZj2CLsFjBkCZl1\\_5Wcb3ekj/view?usp=sharing](https://drive.google.com/file/d/1TO0lIaSExZj2CLsFjBkCZl1_5Wcb3ekj/view?usp=sharing)
4. [https://drive.google.com/file/d/1416DKA087LgxOXJZWOXwSryKd\\_qBJFrP/view?usp=sharing](https://drive.google.com/file/d/1416DKA087LgxOXJZWOXwSryKd_qBJFrP/view?usp=sharing)
5. <https://drive.google.com/file/d/1RmaMqtbNWg1g4rx6xuq3iE1rv1bCmfV/view?usp=sharing>

VEX code

1. <https://drive.google.com/file/d/1SHXlp28FCI44CIBnigMYRmBbF-oKVhqP/view?usp=sharing>
2. <https://drive.google.com/file/d/1RIJ2x9RbikHJ44QikY6OM4imSdtyG5i2/view?usp=sharing>
3. <https://drive.google.com/file/d/1JNsc6sLvU9xB7t7YBAx9LM640RnAtueK/view?usp=sharing>
4. <https://drive.google.com/file/d/1Vs9QQb-0NwsCuL4kfhf0GrxTp-RLnXcv/view?usp=sharing>
5. [https://drive.google.com/file/d/1mA\\_NW\\_th5-73vl-sCGL7jdR7zGoMRXwB/view?usp=sharing](https://drive.google.com/file/d/1mA_NW_th5-73vl-sCGL7jdR7zGoMRXwB/view?usp=sharing)

VEX GO

1. [https://drive.google.com/file/d/1eWXmO6XOv8\\_i7SfXlJcm3yNCcpN5yy2s/view?usp=sharing](https://drive.google.com/file/d/1eWXmO6XOv8_i7SfXlJcm3yNCcpN5yy2s/view?usp=sharing)
2. [https://drive.google.com/file/d/1zIFK4nH88K5RgOv89F0EnRdOv4FbmH\\_3/view?usp=sharing](https://drive.google.com/file/d/1zIFK4nH88K5RgOv89F0EnRdOv4FbmH_3/view?usp=sharing)
3. [https://drive.google.com/file/d/1M1j1C\\_inXznVZICFVRaAu9yF5a\\_g893e/view?usp=sharing](https://drive.google.com/file/d/1M1j1C_inXznVZICFVRaAu9yF5a_g893e/view?usp=sharing)

4. <https://drive.google.com/file/d/1usFlzdh6pUIPNPKgNkKCnnosnMtTfYEN/view?usp=sharing>
5. [https://drive.google.com/file/d/1jZayQ8fJbSVzHJn\\_g2GFf0n9vd7U-17W/view?usp=sharing](https://drive.google.com/file/d/1jZayQ8fJbSVzHJn_g2GFf0n9vd7U-17W/view?usp=sharing)
6. [https://drive.google.com/file/d/1hXo7kxCtJEPB5YA\\_hk62DX2cbtyr7w1W/view?usp=sharing](https://drive.google.com/file/d/1hXo7kxCtJEPB5YA_hk62DX2cbtyr7w1W/view?usp=sharing)
7. <https://drive.google.com/file/d/1WF1Lj2rVio4y10v8ELLEWROwK3Cqosqh/view?usp=sharing>

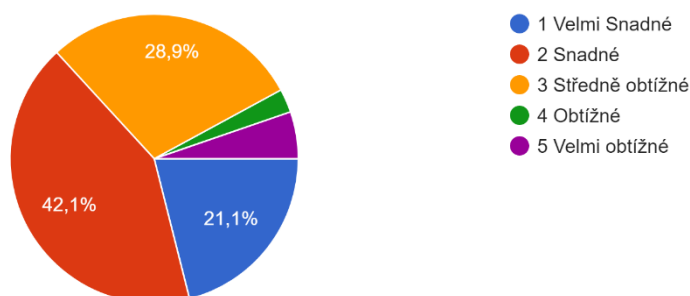
VEX IQ

1. <https://drive.google.com/file/d/1hbw7tjo-bmNyrID-k16SX5-fBzsMhApk/view?usp=sharing>
2. [https://drive.google.com/file/d/1AT7UH-YNrNqBz9-XSOHcF\\_c4qIx6e3W3/view?usp=sharing](https://drive.google.com/file/d/1AT7UH-YNrNqBz9-XSOHcF_c4qIx6e3W3/view?usp=sharing)
3. [https://drive.google.com/file/d/1Jlxa0wgDLLrmPBaKtQ1t\\_IXLeWPC5r1n/view?usp=sharing](https://drive.google.com/file/d/1Jlxa0wgDLLrmPBaKtQ1t_IXLeWPC5r1n/view?usp=sharing)
4. <https://drive.google.com/file/d/1Gm8xs7wgkINTTeQAptfZBPcMbd-Y10HG/view?usp=sharing>

## Výsledky z dotazníku pro 1. stupeň

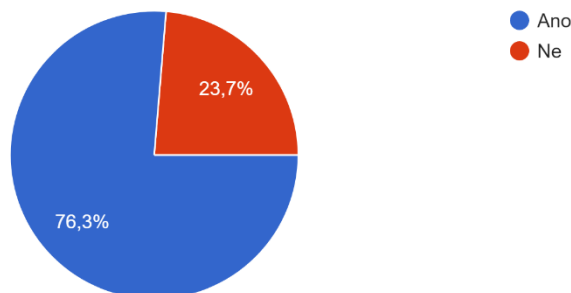
Na škále od 1 - 5 zhodnot, jak obtížné úlohy pro tebe byly?

38 odpovědí



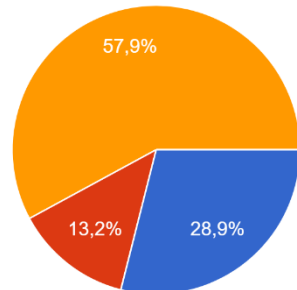
Setkal ses už někdy s nějakými roboty?

38 odpovědí



Jaký typ programování byl pro tebe složitý?

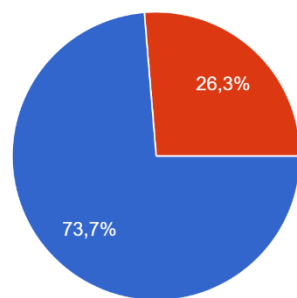
38 odpovědí



- Pomocí dotyku
- Pomocí kódovací desky
- Pomocí prostředí VEX CODE na počítači

Má se podle tebe vyučovat ovládání robotů ve škole?

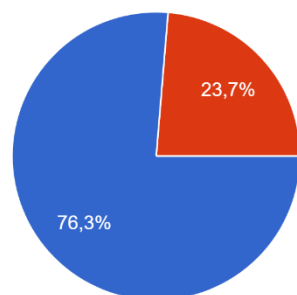
38 odpovědí



- Ano
- Ne

Chtěl bys nadále pracovat s roboty?

38 odpovědí



- Ano
- Ne