Tento vzdělávací materiál vznikl v rámci projektu
CZ.02.3.68/0.0/0.0/16_036/0005322 **Podpora rozvíjení informatického myšlení**.

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

Stavba LEGO robotů

Ing. Jana Kolaja Ehlerová, Ph.D.

2019

Obsah

Kapitola 1

Úvodem

Úlohy v této knize jsou určeny studentům vysokých škol, budoucím učitelům informatiky. Je počítáno s tím, že studenti stráví úlohami nejvýše 13 devadesáti-minutových bloků, proto je s ohledem na omezenou časovou dotaci potlačena stavebnicová část a je kladen důraz na část programovací.
Dospělí studenti nejsou cílovou skupinou robotických stavebnic a byla by škoda, kdyby na tyto hračky hleděli jako na pracovní pomůcku, která je v budoucnu čeká jako součást jejich povolání, možná i rodičovských starostí.
S nadějí, že v každém dospělém sedí jeho vnitřní dítě a to si chce hrát s roboty,

Jana Kolaja Ehlerová

Kapitola 2

Lekce robotiky, dlouhodobá struktura pro několik lekcí

1. Postavte robota, ať se hýbe, závodí a dělá efekty
2. Zamyslete se nad tím, k čemu jsou roboti lidstvu a k čemu obvykle slouží — automatizace, práce v extrémních podmínkách, přesná práce
3. Sestrojení robotů, kteří slouží
4. Radost z toho, že se hýbou (závodí a dělají efekty)

Kapitola 3

Robot WeDo 2.0

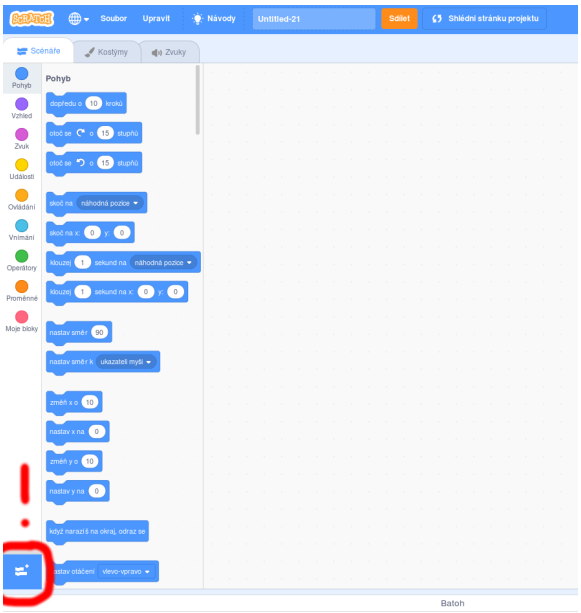
Stavebnice je určena dětem od sedmi let.
Pohyblivé/programovatelné součásti:

- Smart hub se připojuje přes bluetooth a umí svítit barevným světlem (hodnoty 0-100).
- Má motor s volitelnou rychlostí (0-9 v nativním prostředí, ve Scratchi procenta).
- Má senzor pohybu (0-100 mm, s přesností na 10 mm).
- Má senzor náklonu (ve Scratchi rozpoznává nahoru, dolů, vlevo, vpravo).
- Robot (smart hub) nemá paměť, program zprostředkovává počítač/tablet.

Nativní programovací prostředí

Nativní aplikace k WeDo 2.0 umožňuje zjednodušené programování bez jediného slova jen pomocí obrázků. Toto prostředí je intuitivní, vhodné pro děti prvního stupně i dokonce pro ty, které neumí číst.
K samotnému výkonu robota lze v aplikaci přidat zvuk (z tabletu, počítače) nebo obrázky (na obrazovku aplikace).
Programovací jazyk samotný nabízí dostatek funkčnosti, ale i značné omezení

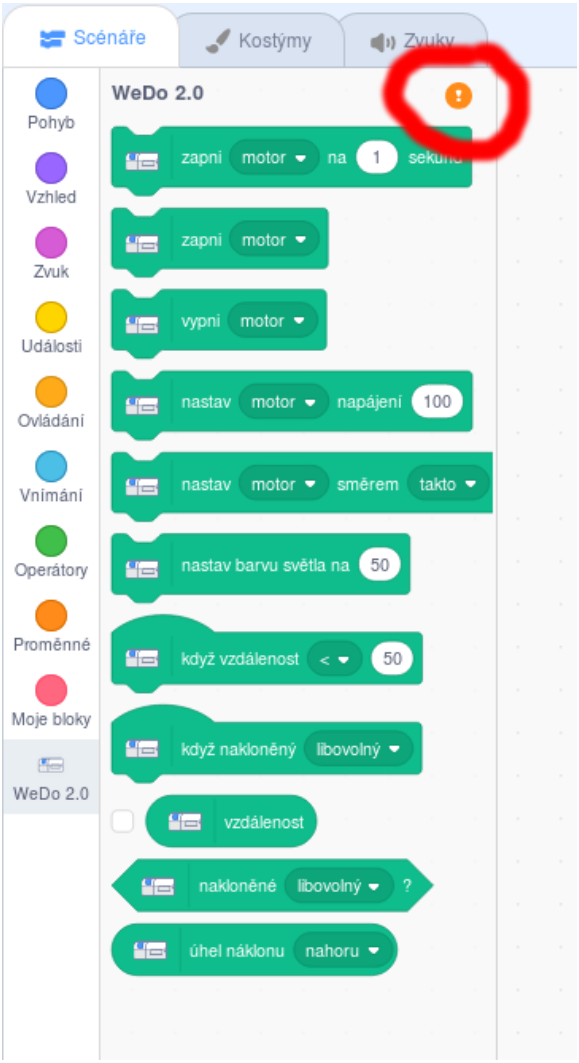
- Můžete použít cyklus, jak nekonečný, tak podmíněný (změna na senzoru) nebo s pevným počtem opakování.
- Nemůžete použít vnořený cyklus.
- Máte k dispozici jednu proměnnou, s kterou můžete provádět operace sčítání, odčítání, násobení a dělení (s desetinnou tečkou).
- S dostatkem fantazie můžete použít i větvení programu, pokud použijete funkci pošli zprávu s nějakým parametrem a zároveň nadefinujete chování programu po zachycení zprávy. Pro děti nevhodné.



Obrázek 3.1: Jak ve Scratchi přidat bloky pro ovládání WeDo robota.

Scratch

Ve spojení s programovacím jazykem Scratch může programátor použít pro WeDo roboty všechny běžné programovací postupy.
Ve Scratchi je potřeba přidat bloky pro programování WeDo. Ve sloupci Code je v prvním sloupci přehled nástrojů, zcela dole je možnost Add Extension. V ní najdete LEGO Education WeDo 2.0 Označená ikona vykřičníku naznačuje, že zrovna není připojený Smart Hub, pokud je připojený, je možné ho v tomto místě odpojit a vice versa.



Obrázek 3.2: Jak ve Scratchi připojit WeDo robota.

3.1 Lekce Zasviť

Co je potřeba

Pouze WeDo Smart Hub.

Cíl lekce

- Připojit se k hubu a postat mu instrukci (rozsvítit světlo v určité barvě).
- Seznámit se s programovacími prostředími pomocí nekomplikované úlohy.

Zadání

1. Naprogramujte postupnou změnu barvy z červené (ve Scratchi 0) až do fialové (ve Scratchi 100).
2. Dokážete změnu barvy zpomalit nebo urychlit?
3. Ukažte, co se stane při překročení 100 nebo jak uděláte nekonečný cyklus se změnou barvy.

Časová dotace

45 minut

Řešení Zasviť v nativním prostředí

Nejprve bude ukázána implementace řešení v nativním prostředí, která nevede k zadanému cíli, ale ukazuje možnosti prostředí.

Obrázek 3.3: Program posvítí krátce jednou jinou barvou.

Obrázek 3.4: Program vybírá náhodnou barvu a nechá ji sekundu svítit.

Obrázek 3.5: Program rozesvítí postupně 10 barev.

V nativním prostředí WeDo je možné použít jednu proměnnou (na počátku programu je nastavena na 0), v deseti opakování cyklu je proměnná inkrementována o jedna, posvíceno touto barvou a program 1 sekundu čeká. Proměnnou lze sice naplnit číslem s desetinnou tečkou, nicméně zelený příkazový blok na svícení bere jen celou část tohoto čísla. Výsledkem není spojitý přechod, barvy nejsou seřazeny jako na barevném kruhu či v duze.

Řešení Zasviť ve Scratchi

Oproti tomu ve Scratchi je možné nastavit barvu světla na číslo 0-100. Zpoma-

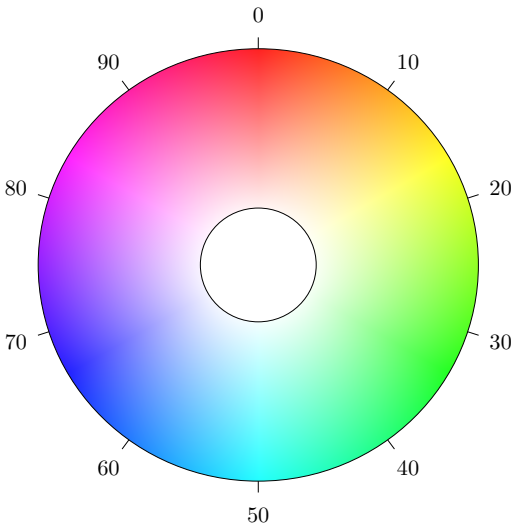
Obrázek 3.6: Plynulý přechod barev pomocí Scratche.

lení nebo zrychlení je možné pomocí „změň barva o jiné číslo“.

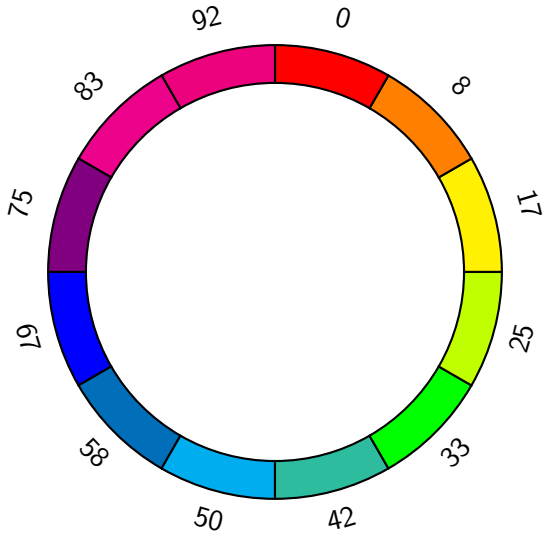
Debata

Po dokončení úkolu je vhodné se zmínit o barevném kruhu.

Barevný kruh 3.7 je znázornění, které odpovídá vnímání barev. Jeho diskretizované zobrazení je na obrázku 3.8. Velmi pěkný materiál o barvách: `https://blog.asmartbear.com/color-wheels.html`



Obrázek 3.7: Barevný kruh, jak odpovídá číslování ve Scratchi jednotlivým barvám



Obrázek 3.8: Barevný kruh,rozdělený diskrétně pro lepší rozpoznání barev.

Jak dlouho trvá Scratchi průchod jedním cyklem?

Uživatel Scratche si jistě povšimne, že program složený z čtyř stejných bloků za sebou trvá kratší dobu než stejný program tvořený pomocí cyklu 4x opakuj příkaz. Blíže se problematice věnuje následující odkaz:
`https://en.scratch-wiki.info/wiki/Single_Frame`

Jak dlouho to trvá jinému programovacímu jazyku?

Předchozí otázka byla bezelstně zodpovězena, ale v této otázce už není ostří skryto, tady je potřeba si přiznat, že odpověď zní: „A jakému cyklu, na jakém stroji, v jakém jazyce?“
Je možné sestavit jednoduchý výpočetní program s mnoha cykly, přepsat jej do několika různých jazyků, spustit a porovnat výkon. V případě, že do tohoto pokusu čtenář zahrne i Scratch, pak je třeba přihlédnout k faktu, že se nelze spolehnout na proměnnou „stopky“ či jiný nástroj v rámci Scratche, při zátěži uvádí hodnoty řádově nižší než reálné.

3.2 Lekce autíčko

Co je potřeba

WeDo Smart Hub + motor + senzor pohybu + kola...

Cíl lekce

- Seznámit se s konstrukčními postupy stavebnice Lego.
- Pomocí soutěživé úlohy najít vděčný cíl robotických stavebnic.
- Zjistit, že i jednoduché zadání může vést k matematickým problémům.

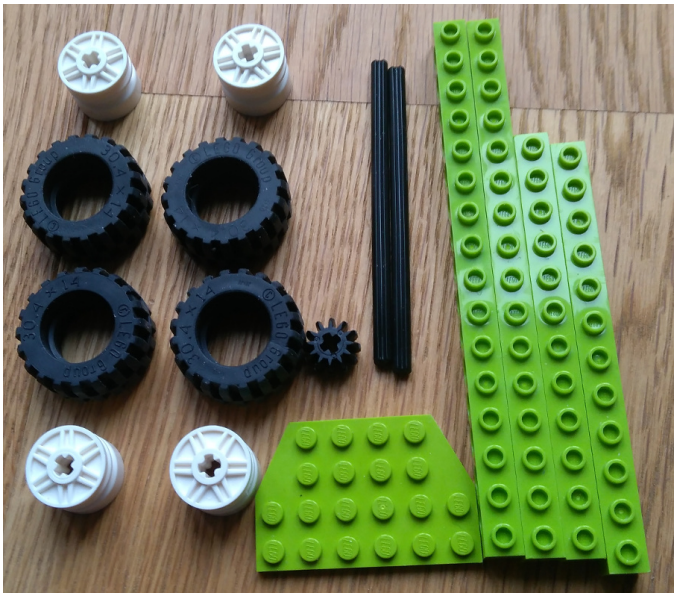
Zadání

1. Sestrojte autíčko.
2. Naprogramujte motor, aby jel.
3. Udělejte závody autíček, kdo nejdříve dojede ke zdi.
4. Auta naráží do zdi. Přidejte autu senzor pohybu. Modifikujte program, aby zvládl jet a nenabourat do zdi.
5. Uspořádejte závody, kde auto, které naboural do zdi, bude diskvalifikováno.
6. Pokuste se napsat program tak, aby auto dojelo právě 5 cm od stěny.

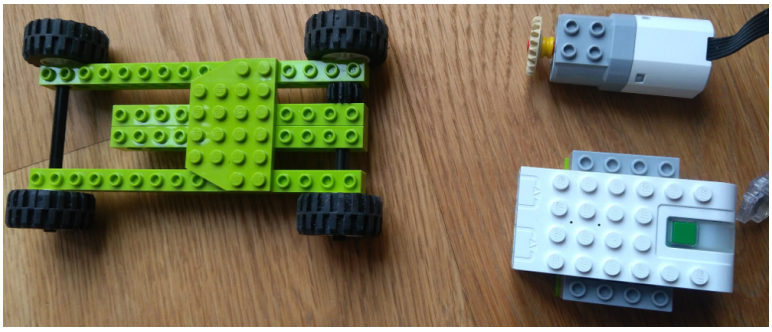
Časová dotace

135 minut

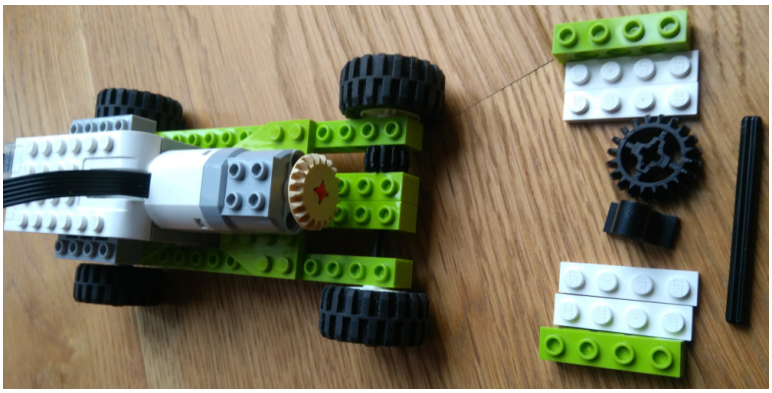
Jak postavit auto



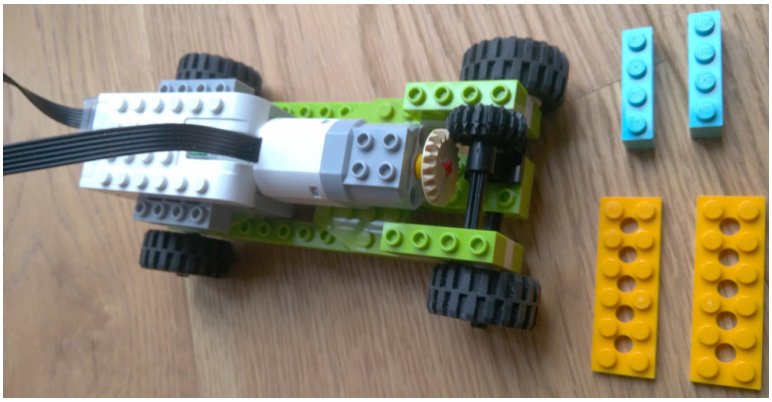
Obrázek 3.9: Přichystejte materiál na podvozek



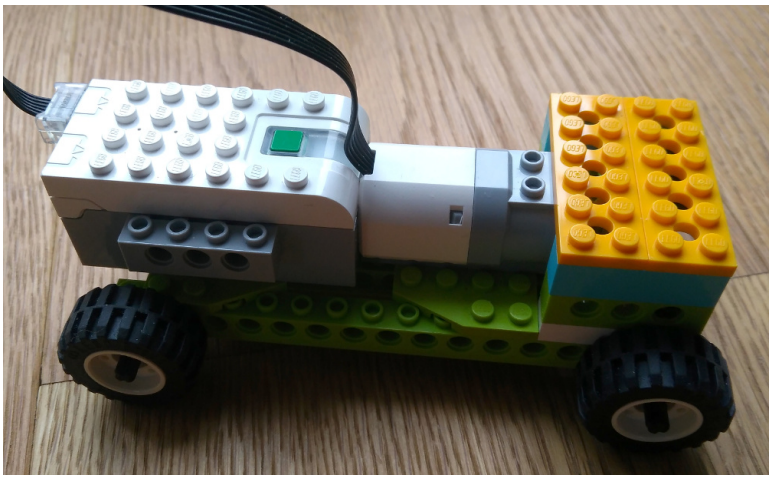
Obrázek 3.10: Sestavte podvozek a přichystejte blok a motor. V motoru je červená osa, nechte ji dostatečně vystrčenou ven, aby z ní nesklouzlo kolo.



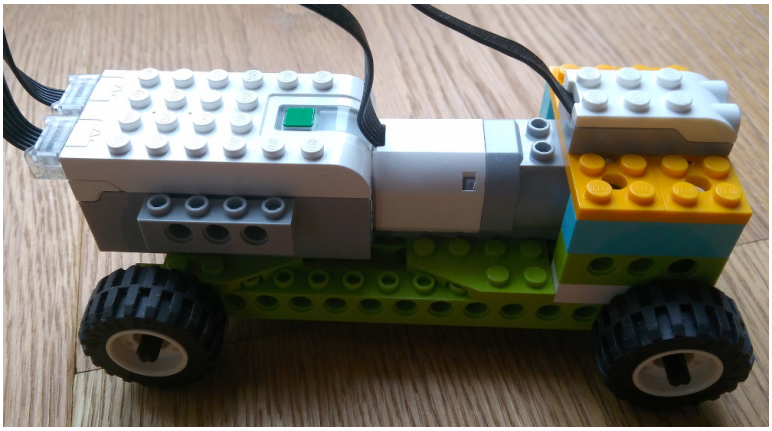
Obrázek 3.11: Umístěte blok a motor, přichystejte materiál na převodovou skříň.



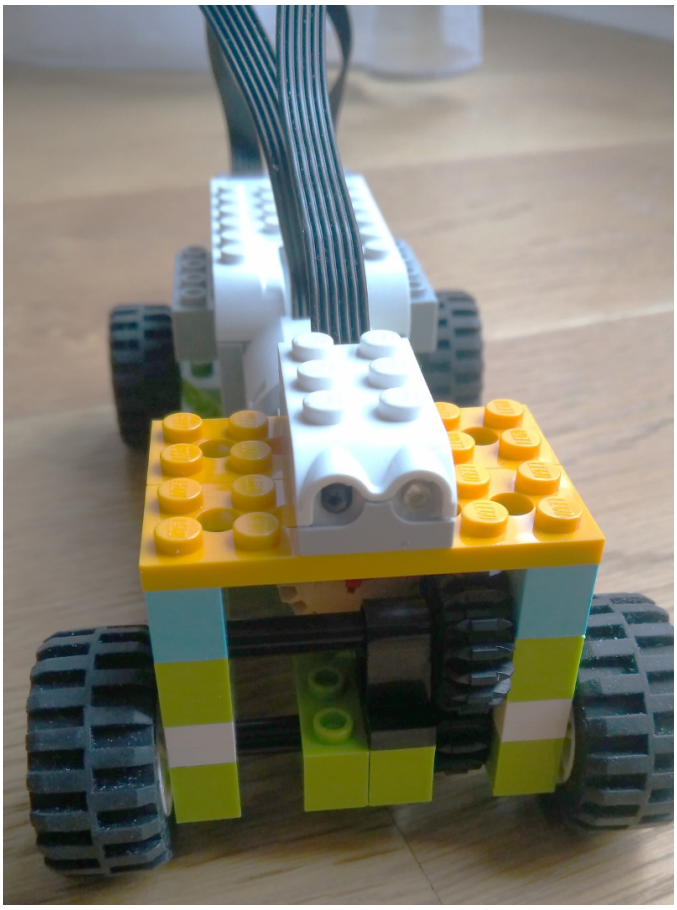
Obrázek 3.12: Sestrojte převodovku, která v této variantě má tendence se rozpadat, připravte se na její zpevnění.



Obrázek 3.13: Zakrytujte převodovku.



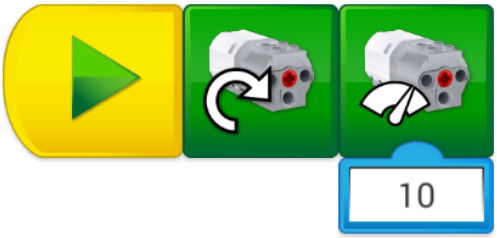
Obrázek 3.14: Přidejte senzor pohybu.



Obrázek 3.15: Pohled zepředu na převodovou skříň a senzor.

Řešení Autíčko v nativním prostředí

Jak rozjet auto, je na obrázku 3.16, první instrukce nastaví směr otáčení motoru a druhá motor zapne na plný výkon.



Obrázek 3.16: Rozjede motor na plné otáčky.

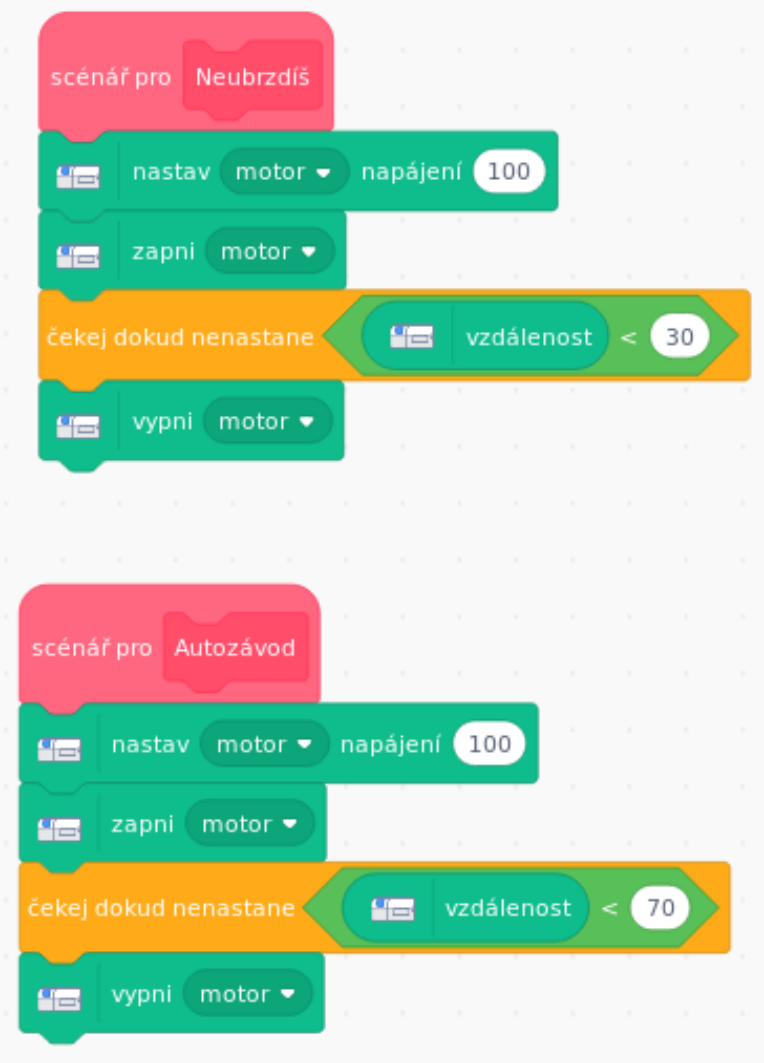
Úkol, jak se nevybourat, je na obrázku 3.17:

Obrázek 3.17: Rozjede motor na plné otáčky, po spatření překážky zastaví.

Program otáčí motor, dokud nezjistí, že je před ním překážka. Nelze zajistit, aby brzdil později nebo zpomaloval, takto staví poměrně daleko před překážkou (vidí ji na 10 cm, zastavuje.) Je možné ukázat, že při opakovaném experimentu nikdy nezastaví stejně daleko od stěny, což není chyba programovacího prostředí, ale rychlosti odezvy.

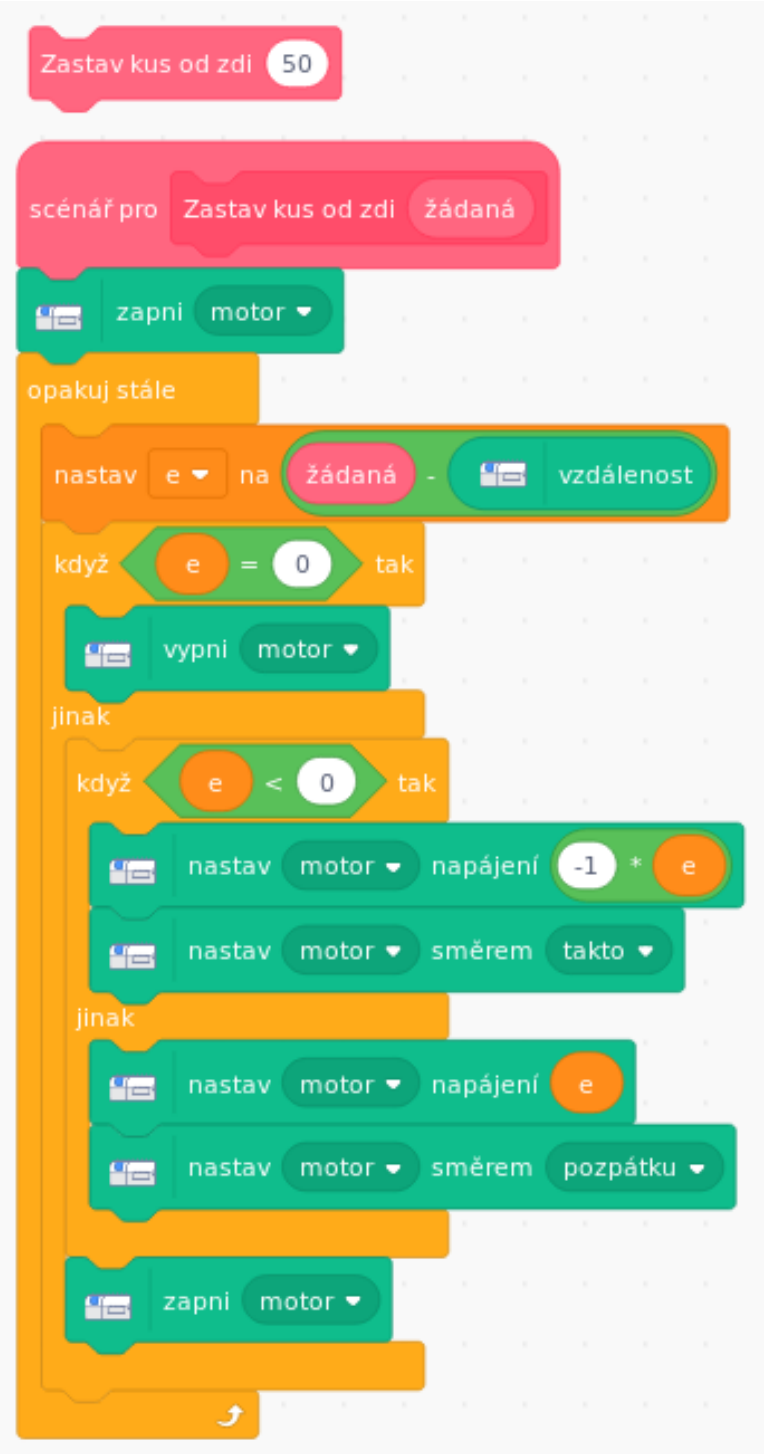
Řešení Autíčko ve Scratchi

První úkol rozjet auto je natolik snadný, že zde není uveden, stačí dát motor do provozu a případně odladit, aby auto necouvalo.
Úkol, jak auto rozjet a nevybourat, je řešitelný pomocí podmíněného cyklu, scénář Neubrzdíš 3.18. Schválně je nastavena vzdálenost 30 mm, až pak začíná auto brzdit a neubrzí. Reakční doba celého systému je na to příliš dlouhá.
Scénář Autozávod 3.18, kde je vzdálenost k překážce 6 cm a méně, už zvládá zabrzdit, ale ve výsledku je auto pokaždé v jiné vzdálenosti od zdi.
Je možné dosáhnout toho, aby dojelo do konkrétní vzdálenosti, ale vrhnout se do vlastního vytváření, je poměrně nevděčná a strastiplná cesta. Zde přiložený



Obrázek 3.18: Rozjede motor a zastaví po zaregistrování překážky. Jednoduchá varianta bez zpětnovazebního řízení.

algoritmus (Zastav kus od zdi, nastaveno na 5 cm) 3.19 využívá teorie z oboru automatizace řízení.



Obrázek 3.19: Zpětnovazební algoritmus pro zastavení auta v žádané vzdálenosti.

Debata

Diskuse k Autíčku se snadno může odvést od informatiky

Může auto i zatáčet?

Šlo by zpřevodovat, aby při zpětném chodu couvalo, ale bez přímého řízení to není radost. Vlastně je potřeba se smířit s tím, že výsledkem nemá být auto na dálkové ovládání, ale samořiditelné vozidlo, které si nicméně má student naprogramovat sám.

Šlo by autíčko přemluvit, ať přiveze někomu ke stolu vzkaz, počká minutu na odpověď a zacouvá zpět?

Ano, tento typ úlohy je pro něj jako dělaný.

Mohlo by auto vozit vzkaz (reklamu, nápis) a jet v nějakém úseku pomalu, pak zrychlit (tam, kde to nemá nikdo číst) a pak zase zpomalit?

Jistě.
Přiveďte debatu k tomu, že toto není auto na dálkové ovládání, ale automatický systém, který nemusí být dokonalý, ale je velmi trpělivý.
Jiná část debaty může dojít ke zpětnovazebnímu řízení a tomu, kde všude jde využít.

3.3 Lekce parkovací senzor

Co je potřeba

WeDo SmartHub + senzor pohybu + kola ...

Cíl lekce

- Seznámení se s funkcí senzorů a vyhodnocení jejich návratových hodnot.
- Programování pomocí větvení programu.

Zadání

1. Sestrojte auto bez motoru, pouze se SmartHubem a senzorem pohybu.
2. Naprogramujte parkovací senzor, který v případě překážky pípá, čím je blíž překážce tím častěji.
3. Otestujte.

Časová dotace

90 min

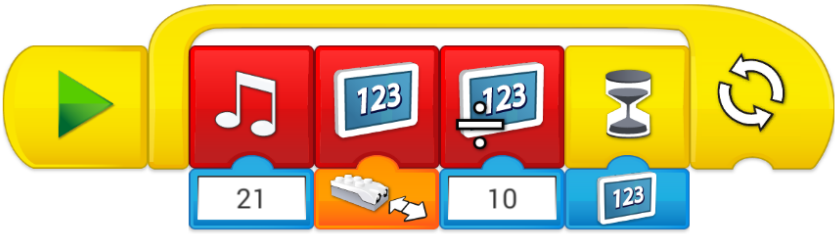
Jako postavit auto se senzorem

Teoreticky netřeba nic stavět, ale lépe se pracuje s kompaktním modulem+senzorem na kolech.

Řešení Auta se senzorem v nativním prostředí

Nativní prostředí WeDo dovoluje jen velmi omezené použití podmínky, je-li změna vzdálenosti viditelné senzorem.

Obrázek 3.20: Takto je možné senzor donutit, aby pípal, jakmile vidí změnu vzdálenosti.



Obrázek 3.21: Parkovací senzor pípá rychleji, když vidí překážku blíž.

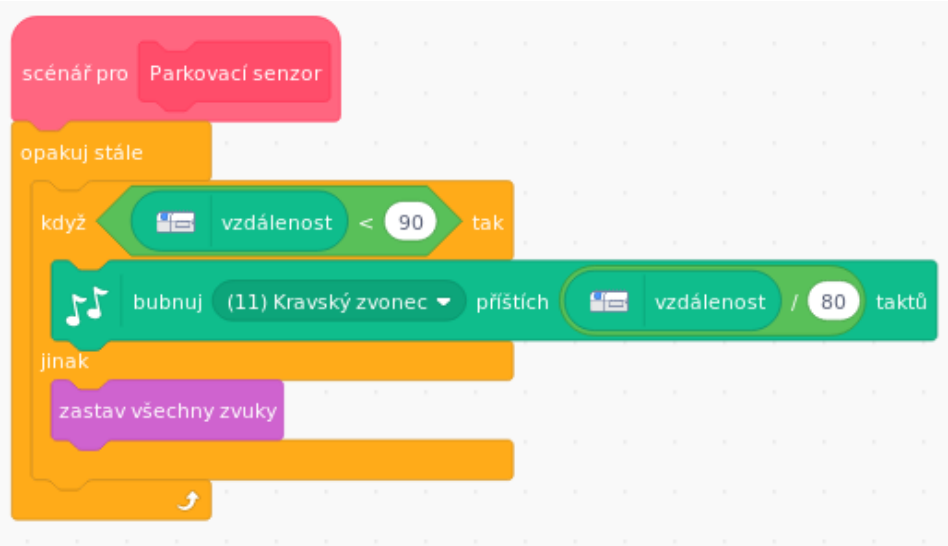
Do programu je možno přidat proměnnou:

- proměnná = vzdálenost viděná senzorem
- proměnná = proměnná / nějaká konstanta (zde 10)
- doba čekání = velikost proměnné

Tento senzor pípá, i když nic nevidí, tedy vzdálenost překážky je 10 a vyšší.

Řešení Auta se senzorem ve Scratchi

Ve Scratchi je možno využít buď funkce „Zvuk“ a blok „čekej“ nebo přidat rozšíření „Hudba“.
Scénář „Parkovací senzor“ využívá opět přímé úměry mezi vzdáleností k překážce a rychlostí pípání, jen je třeba najít ten vhodný poměr.



Obrázek 3.22: Parkovací senzor pomocí Scratche.

Debata

Když si necháte při couvání ukázat od člověka, jak daleko ještě máte do zdi, získáte jasnou představu, jestli 5 čísel nebo půl metru. Proč vám parkovací senzor neřekne vzdálenost a jenom pípá?

Při couvání nikoli kolmo k překážce není zřejmé, jestli uživatel chce znát vzdálenost průměrnou nebo tu nejmenší. Senzor také může zaregistrovat sloupek, který ale při přiblížení zmizí z jeho dosahu, takže pak podává mylnou informaci.

3.4 Lekce vrátný

Co je potřeba

WeDo SmartHub + senzor pohybu + senzor náklonu

Cíl lekce

- Naučit se pracovat se senzorem náklonu.
- Využití dvou senzorů najednou.
- Ukázání úlohy extrémně vhodné pro automatizaci — stroj se neunaví při čekání a vykonávání stále stejné činnosti.
- Úloha bez akčního zásahu, pouze s pozorovací funkcí.

Zadání

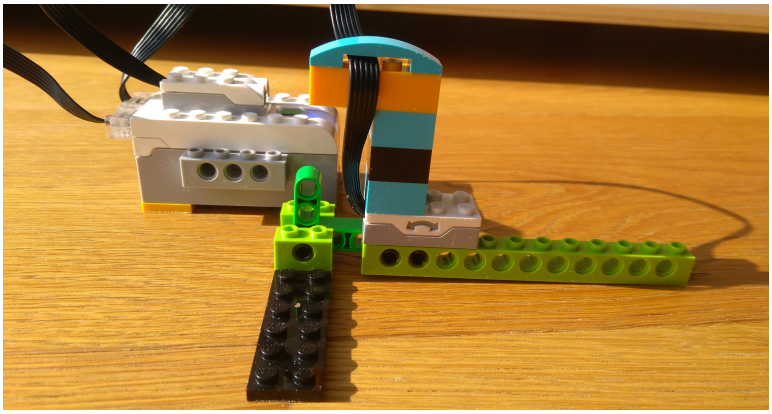
1. Postavte závoru, která bude mít na sobě senzor náklonu a zároveň bude sledovat, co je před závorou (jako když vjíždíte do podzemní garáže).
2. Napište program, který případnému vrátnému řekne, jestli může spát (závora je zavřená, nikdo před ní nestojí), jestli má otevírat (závora zavřená, před ní někdo stojí), jestli má zavírat (závora otevřená, nikdo před ní nestojí) a nebo jestli se má dívat, jak někdo projíždí (závora otevřená, někdo tam stojí).

Časová dotace

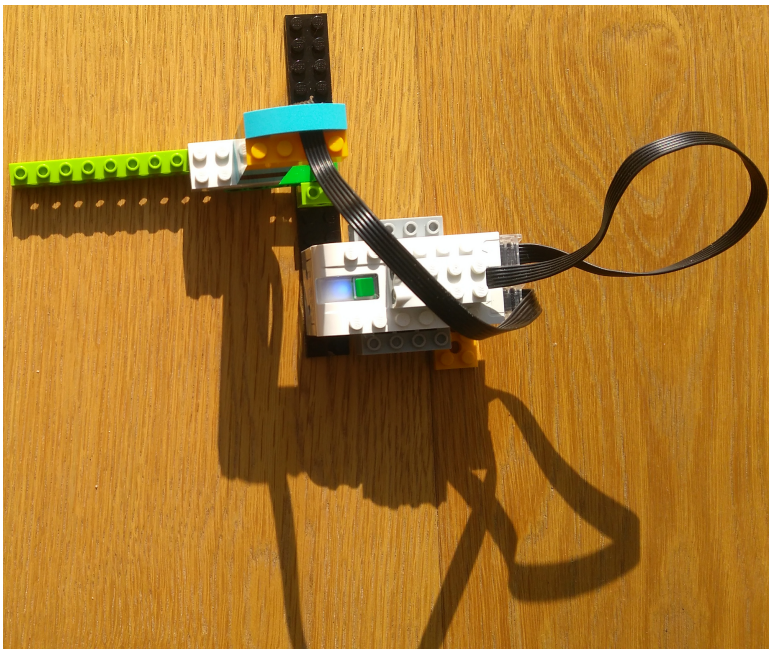
90 min

Jak postavit závoru

Toto je příklad závory, která dobře drží zavřená i otevřená.



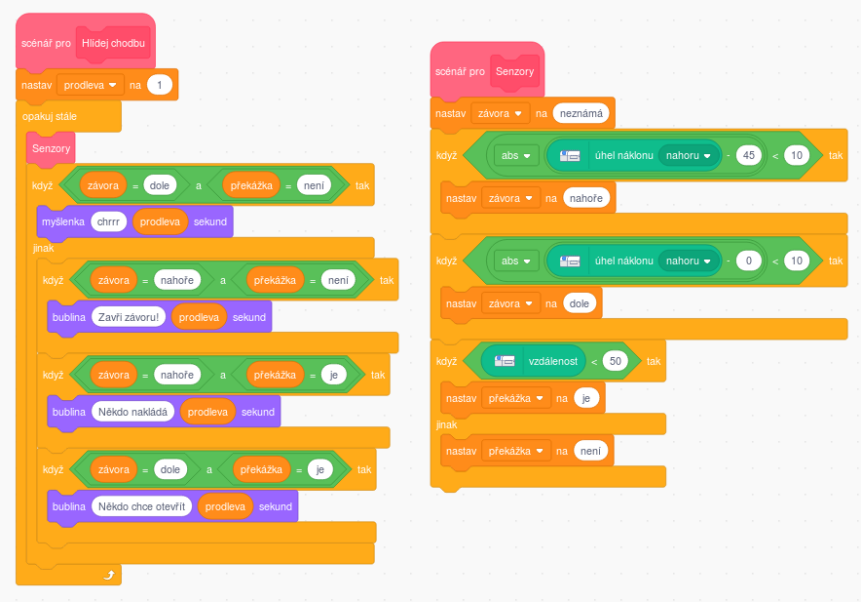
Obrázek 3.23: Ležící závora s připevněným senzorem náklonu.



Obrázek 3.24: Pohled shora na závoru, blok a černá podstava jsou pevné, zelená závora se otáčí kolem osy.

Řešení ve Scratchi

Toto cvičení doporučuji řešit pouze ve Scratchi. Jsou v něm hojně využívané podmínky. V navrženém řešení je oddělené čtení toho, co dávají senzory, a



Obrázek 3.25: Scénář pro vrátného ve Scratchi.

vyhodnocení, co s tím mám dělat. Není ošetřený případ, že je závora v neznámé pozici.

Debata

Zkuste v debatě vyřešit úlohu na vjezd do parkovacího domu.

WeDo má k dispozici příliš málo výstupů, ale je možné navrhnout teoretické řešení, které využije motor na zvedání závory, senzor pohybu, senzor náklonu (teoreticky, možno ho vynechat), tlačítko na tisk parkovacího lístku, světlo červené/zelené.

K čemu je možné využít systém, který jen sleduje, neprovádí žádný akční zásah (tedy pípá, bliká, ale nikam neodjíždí)?

Hledejte příklady ze života.

- Radar před obcí
- Parkovací asistent, jízdní asistenti (upozorňují na usínání řidiče, příliš malou vzdálenost k dalšímu autu, přejíždění z pruhu do pruhu)
- Alarm

3.5 Lekce zvědavé autíčko, bojácné autíčko

Co je potřeba

2x WeDo SmartHub + senzor pohybu + kola + motor + ...

Cíl lekce

- Návrat k soutěžení.
- Využití práce ve skupině, naprogramování navazujících úloh.

Zadání

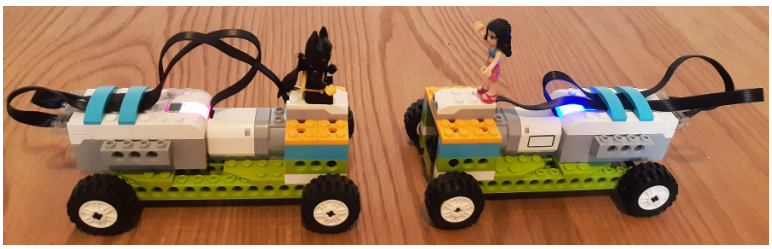
1. Postavte dvě auta.
2. Bojácné auto naprogramujte tak, aby se rozjelo v případě, že uvidí někoho blíže než 5 cm, jakmile bude objekt daleko, zastaví.
3. Zvědavé auto naopak jede ke každé překážce, kterou uvidí (blíže než 9 cm, například).
4. Dejte za sebe bojácné a zvědavé auto, aby se pronásledovala.
5. Sledujte dynamiku úlohy, navrhujte vylepšení.

Časová dotace

90 min

Jak postavit auta

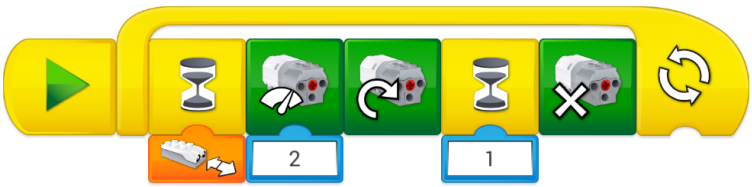
Návod na auto je v textu v kapitole 3.2.



Obrázek 3.26: Při větším množství autíček je důležité je dobře a zapamatovatelně označit.

Řešení Zvědavého a bojáckého autíčka v nativním prostředí

Úloha pro řešení není obtížná, spíš je náročnější zkoordinovat očekávání s tím, co obě autíčka umí, aby výsledek byl zajímavý.
Nativní prostředí umožňuje pouze sledovat, jestli proběhl nějaký pohyb nebo neproběhl. Proto zde Zvědavé auto po zaregistrování pohybu jede vpřed malou rychlostí po dobu jedné sekundy a Bojácné auto po zaregistrování pohybu couvá největší rychlostí a svítí náhodnou barvou.

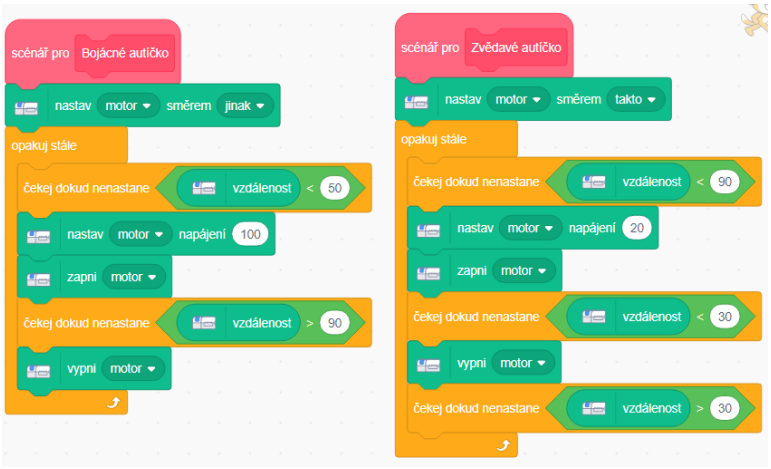


Obrázek 3.27: Program Zvědavé autíčko

Obrázek 3.28: Program Bojácné autíčko

Řešení Zvědavého a bojácného autíčka ve Scratchi

Jakmile je možné získat přesnější určení vzdálenosti ze senzoru, je možné lépe definovat podmínky, kdy má kdo kam odjet. Přiložené scénáře způsobují téměř plynulý pohyb Zvědavého autíčka vpřed a trhaný pohyb Bojácného autíčka vzad.



Obrázek 3.29: Program Zvědavé autíčko

Debata

Je možné nějak vyřešit, aby bojácné auto nenacouvalo do překážky?

Ne, WeDo sada umožňuje připojit jen motor + 1 senzor.

Navrhněte jinou variantu spolupracujících aut, například těch, které jedou v konvoji

Pokud budou všechna auta mít senzor vpředu, mělo by první auto jet podle sebe, další pak musí sledovat předchozí auto a udržovat setrvalý rozestup.

3.6 Lekce radar

Co je potřeba

WeDo SmartHub + motor + senzor pohybu + ramena ...

Cíl lekce

- Práce se systémem, který v reálném čase předává data k zobrazení.
- Nalezení hranic a omezení pro robotickou stavebnici.

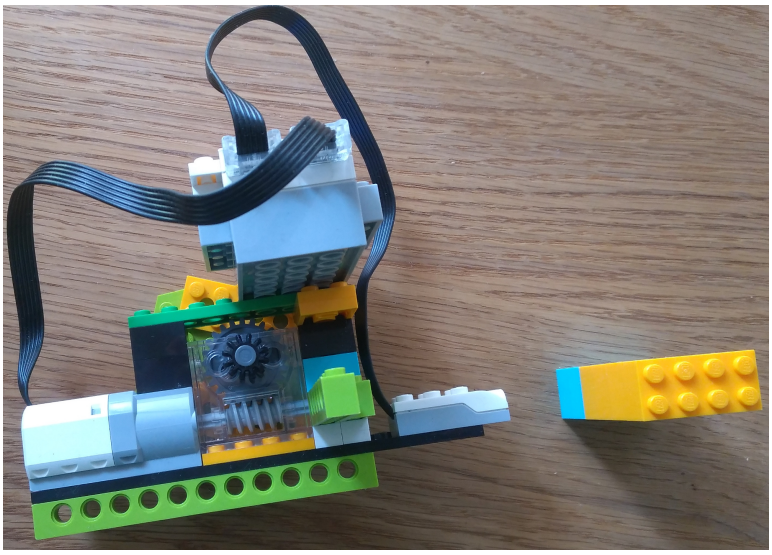
Zadání

1. Postavte radar, tedy něco, co stojí na místě, otáčí se kolem své osy a sleduje okolí.
2. Naprogramujte jej, aby zobrazoval scannovaný kruh a překážky, které v něm vidí.
3. Nakalibrujte systém, aby co nejlépe popisoval realitu.

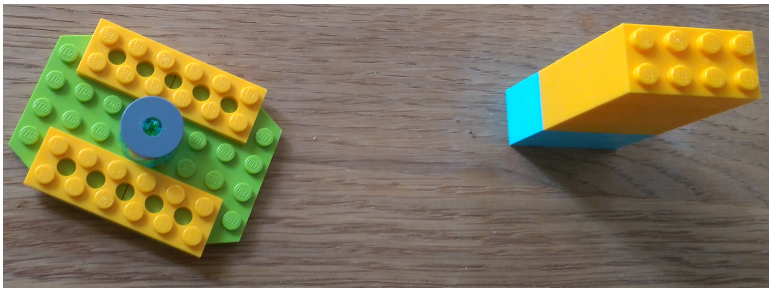
Časová dotace

180 min

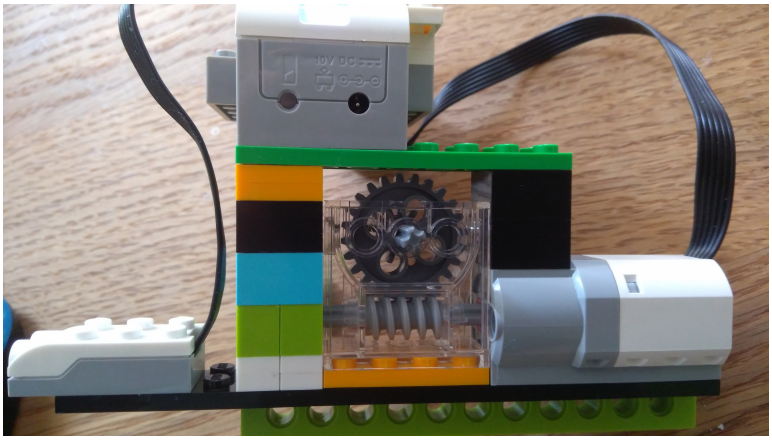
Jak postavit radar



Obrázek 3.30: Pohled shora na radar, jak kouká na překážku.



Obrázek 3.31: Podstava s překážkou.



Obrázek 3.32: Těleso radaru, pohled na šnekové ústrojí, středem ústrojí prochází osa, která pak stojí v podstavě.

Řešení Radaru v nativním prostředí

V nativním prostředí je omezena vizualizace, takže je možné rozpohybovat radar a pak použít stejný mechanismus jako pro parkovací senzor.

Obrázek 3.33: Radar pípá, vidí-li překážku. Bez vizualizace.

Řešení Radaru ve Scratchi

Scratch je sice pro tuto úlohu vhodnější, ale ani zde není implementace bezproblémová.
Navržené řešení počítá s postupem

1. Kalibrace radaru, jedno otočení kolem osy s jednou překážkou
2. Spuštění radaru

Po zkalibrování 3.34 je známá

- Délka cyklu: jak dlouho trvá otočení radaru
- Počet cyklů na jednu otočku: Cyklus ve Scratchi má nějaké zpoždění. Teoreticky by tedy bylo možné zjistit, kolik cyklů spotřebuje jedno otočení kolem své osy.

Tento program počítá s tím, že se radar otáčí konstantní rychlostí se stále stejnou délkou cyklu, Scratch postava se otáčí spolu s radarem a odchází do vzdálenosti, kterou vrací senzor. Tam ponechává svůj klon, který po uplynutí délky cyklu mizí.
Teoreticky byl tento systém mohl krásně vykreslovat, ovšem narážíme zde na limitaci především prostředí Scratch.

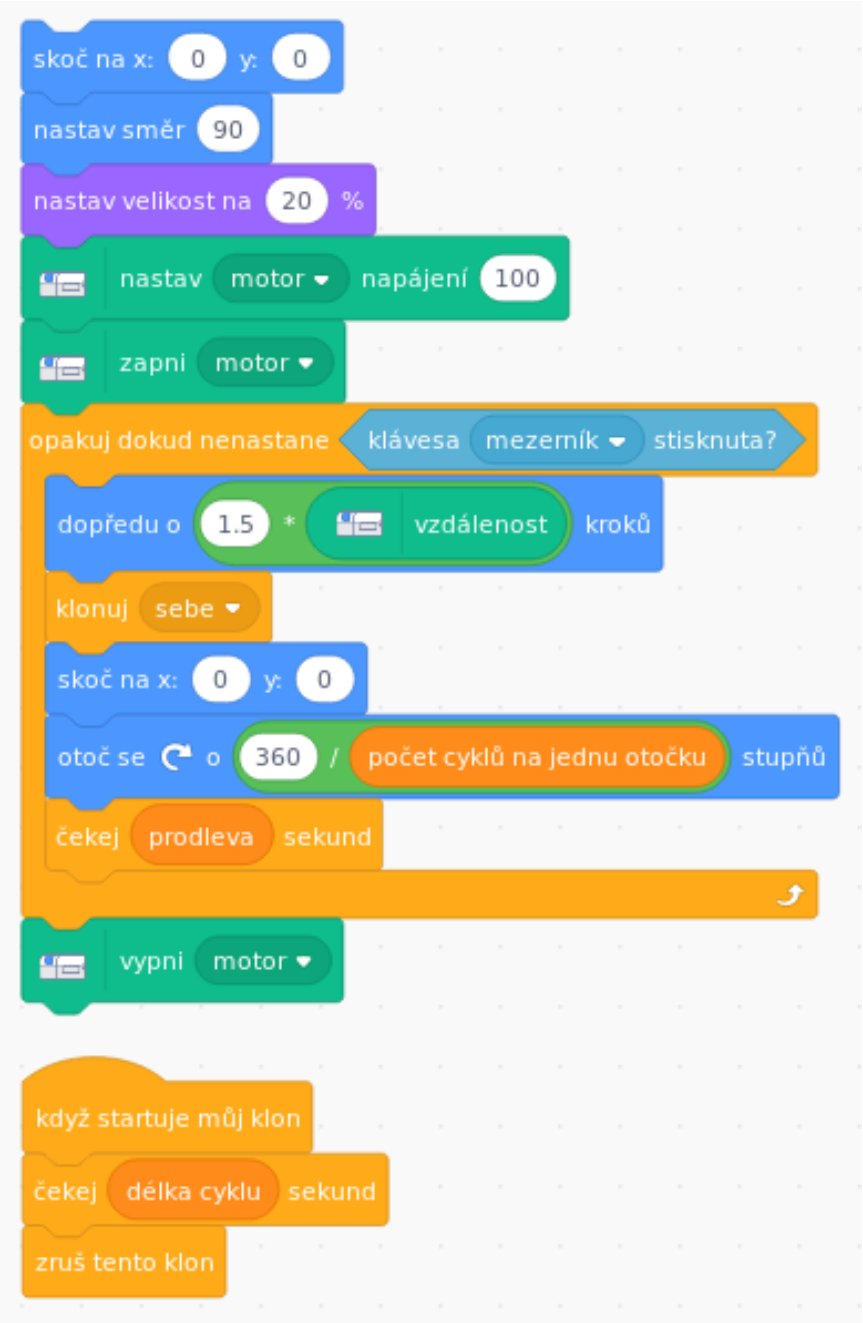
- Už při kalibraci je potřeba si povšimnout, že délka cyklu naměřená pomocí Scratchových stopek se dost liší od reálné délky cyklu
- Při kalibraci je v cyklu jedna iterace a jedno čekej. Ve skutečném měření je uvnitř cyklu pohyb a klonování, které zpomalují funkci Scratche.

Jedním z řešení je najít vlastní funkční hodnoty "Počet cyklů na jednu otočku". Ukažte pojmy

Obrázek 3.34: Kalibrace proměnné délka cyklu na jednu otočku

- validace (funguje to?)
- verifikace (ukazuje to, co má?)
- kalibrace (nalezení parametrů, po kterých to bude fungovat)

Kromě omezení Scratche je také omezení WeDo robota, jehož motor nemusí mít stále stejný výkon a senzor pohybu má zpoždění při hlášení vzdálenosti. Jiné řešení, jak dostat z celého soukolí přesnější údaje, je pouštět motor krokově, tedy spustit, zastavit, zpracovat, co dává senzor, a pak teprve pouštět dál.



Obrázek 3.35: Vizualizace radaru

Debata

Máte-li geniální myšlenku a nespolupracují-li nástroje, čí je to chyba?

Na nástroje je možné si stěžovat, ale nakonec musíte pracovat s materiálem, který máte k dispozici.

Máte trvat na to, aby se realita opravila do té správnější?

Návod, jak se srovnat s realitou, je pro tento text příliš ambiciózní.

Nebo je potřeba se smířit s nedokonalostmi a přistoupit k řešení z jiné strany?

Rozhodně ano!

Nebo se smířit s velmi nedokonalými výsledky?

Tomu se říká vzdát to. Vraťte se k předchozí otázce.

3.7 Lekce terč

Co je potřeba

WeDo SmartHub + motor + senzor náklonu + ramena ...

Cíl lekce

- Úloha, ve které je možné přímo interagovat s výslednou hračkou.
- Jednoduchá úloha, která umožňuje velké množství rozšíření.

Zadání

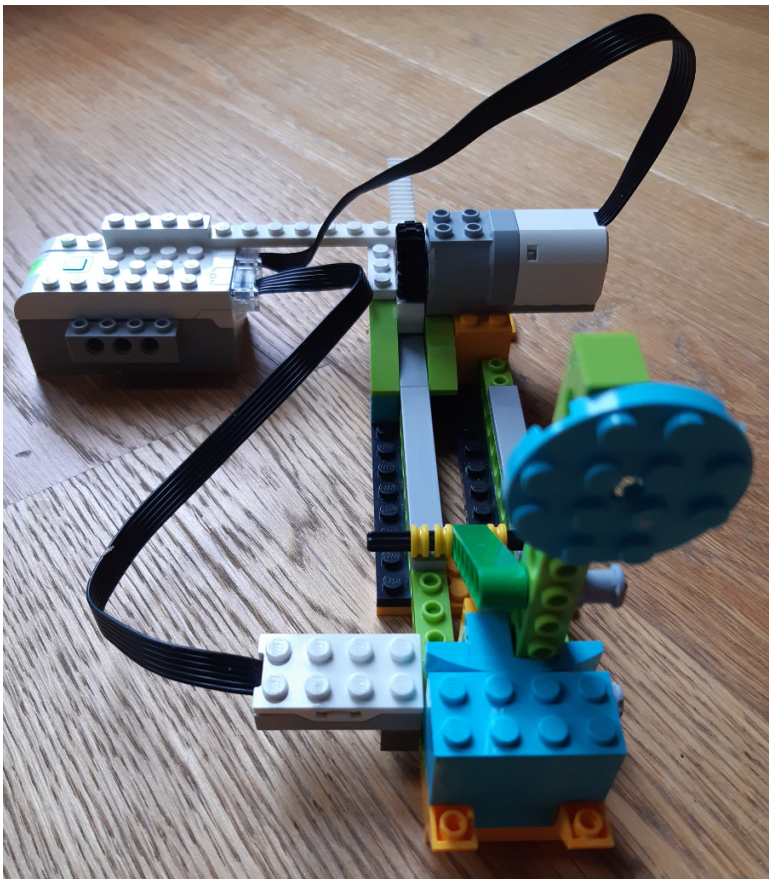
1. Vyrobte terč, který je půjde snadno převrhnout a bude mít zabudovaný senzor, který umí zaznamenat, jestli je terč nahoře nebo dole. Dále bude mít mechanismus, který terč znovu postaví.
2. Naprogramujte terč tak, aby po zasažení terče, terč padl a byl započítán bod, zahrána vítězná fanfára a terč se znovu zvedl.
3. Připrogramujte počítadlo na počet shození a k tomu stopky, které pak hráči oznámí, jak dlouho mu to trvalo. Po dlouhé nečinnosti může terč také upozorňovat (přehrávat „už mě shoď“ atp.).

Časová dotace

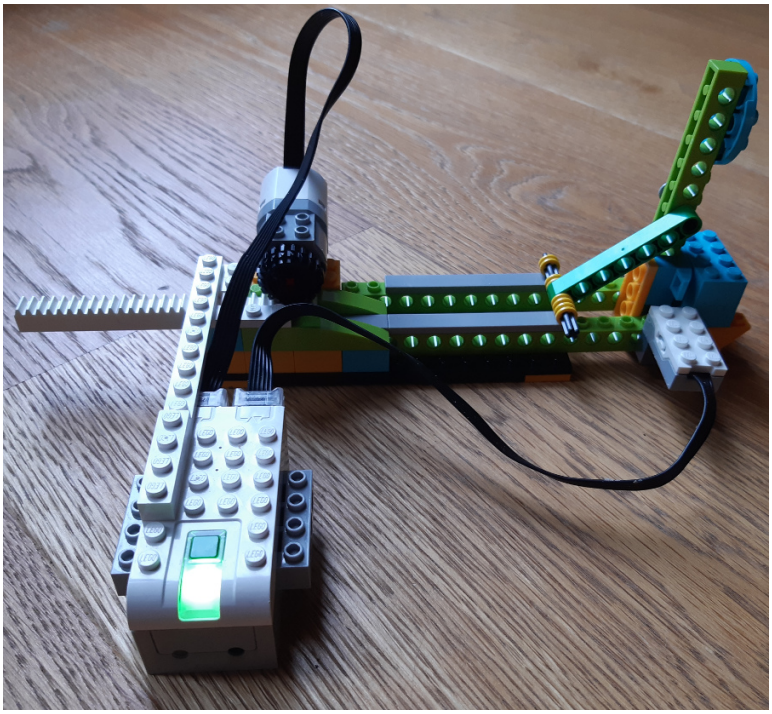
90 min

Jak postavit terč

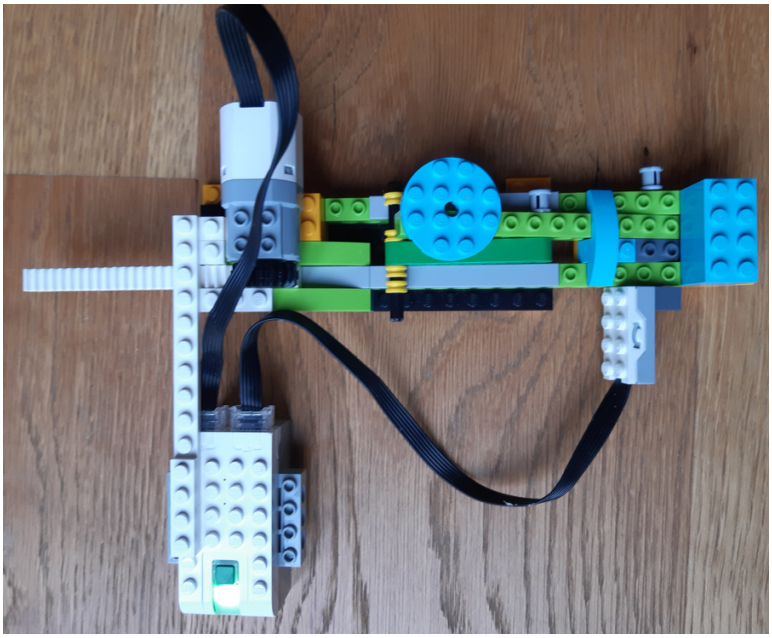
Základní konstrukce má dostatečně pevnou základnu a jedno kyvné rameno se senzorem náklonu. Ke konstrukci je připevněný motor, po lyžinách posílá bílý zubatý díl a staví rameno.



Obrázek 3.36: Pohled zepředu, modrý nezaostřený kruh je terč.



Obrázek 3.37: Pohled zboku, rameno vpravo je postavené, na něm je senzor náklonu. Motor posunuje bílý díl po šedých lyžinách.



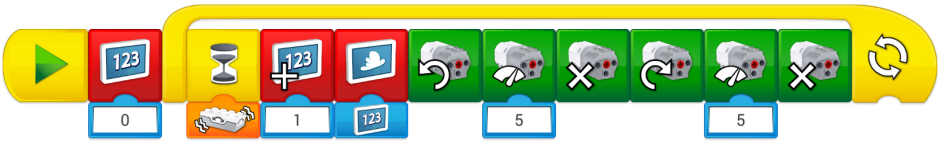
Obrázek 3.38: Pohled shora na sklopený terč.

Řešení Terče v nativním prostředí

Úloha terče je pro WeDo vhodná také proto, že její řešení je snadno implementovatelné jak v nativním prostředí tak ve Scratchi.
První řešení 3.39 napřímí terč a čeká, až dojde ke změně na senzoru náklonu.

Obrázek 3.39: Program postaví terč, čeká až spadne a znova staví.

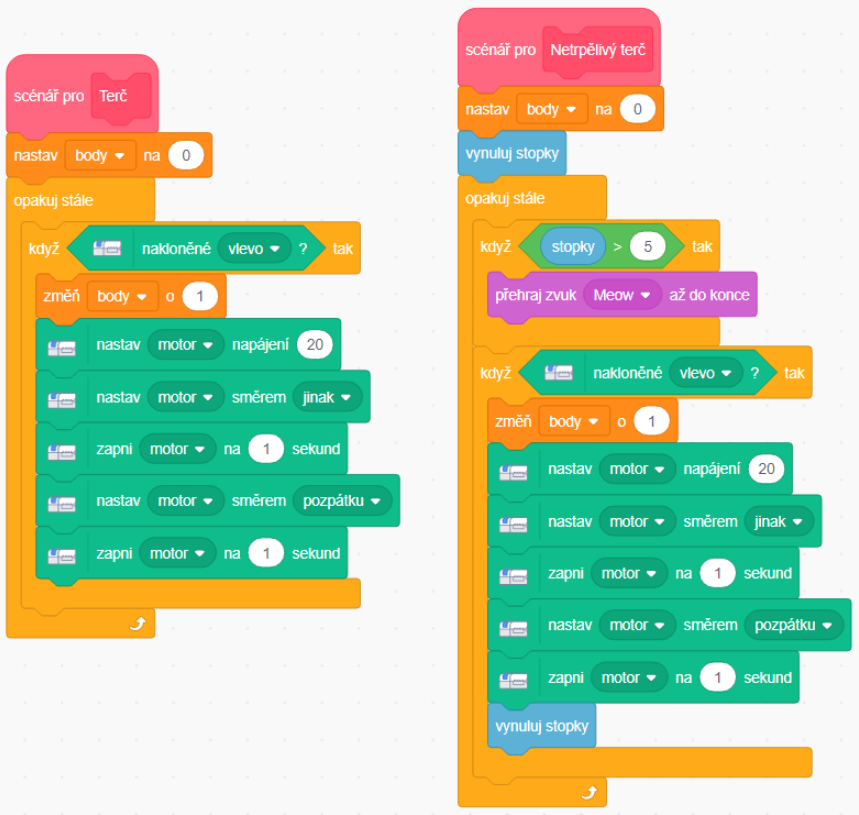
Druhé řešení 3.40 přidává do programu proměnnou, na počátku ji inicializuje na 0, při každé změně náklonu a napřímení ramene ji inkrementuje o 1 a zobrazí.



Obrázek 3.40: Terč s počítadlem

Řešení Terče ve Scratchi

Ve Scratchi 3.41 je pod scénářem „Terč“ uvedený program s počítadlem bodů, pod scénářem „Netrpělivý terč“ po pěti sekundách nečinnosti začíná program mňoukat.



Obrázek 3.41: Scénář Terč a Netrpělivý terč

Debata

Použití senzoru náklonu může působit problémy při konstrukci nebo programování.

Pokud vadí při konstrukci a působí jako nechtěné závaží, je možné jej vyměnit za senzor pohybu sledujícího, je-li terč vzpřímený nebo ne.
Při programování je potřeba dobře ověřit, jaké hodnoty senzor vrací.

Při vyčkávání na padnutí terče je použita proměnná stopky a podmínka, je-li větší než 5. Šlo by testovat, je-li rovna 5?

Testovat je to možné, ale s velkou pravděpodobností podmínka nebude splněna. Provádění cyklu trvá nějakou dobu a většinu času pouze testuje dvě podmínky. Proměnná stopky se ale neustále inkrementuje a to nikoli spojitě, ale po malých diskrétních dávkách. Dokonce není jisté, že vůbec někdy nabývá hodnoty 5.

Kapitola 4

Robot Mindstorms 2.0

Mindstorms je určeno pro děti od 10 let, má silnější motory, lepší senzory, možnost připojení více součástek a je doplněno stavebnicí LEGO Technic. To vše umožňuje stavbu komplexnějších strojů, lepší propojení s programátorským prostředím a pochopitelně i vyšší časovou dotaci na složitější úlohy.

Programování pomocí Scratche

Je možné, nicméně je velmi omezující. V tuto chvíli (srpen 2019) Scratch neumožňuje kontinuální ovládání motoru, má omezený vstup ze senzorů. Na vyzkoušení, že „něco“ funguje, je toto prostředí možné, pro seriózní programování není vhodné.

Programování pomocí nativního prostředí

Nativní nástroj pro EV3 vychází z LabView. Tentokrát nechybí podmínky ani proměnné, nicméně je potřeba věnovat čas k osahání prostředí.
Odkazy pro představení prostředí Mindstorms

- `https://www.lego.com/cs-cz/themes/mindstorms/learntoprogram`
  Návody na stránkách LEGO, vhodné i pro děti.
- `https://www.robotworld.cz/downloads/manual-lego-mindstorms-ev3-cs.pdf`
  Ucelenější text, vhodný i pro vysokoškolské studenty.

4.1 Lekce Zasviť

Co je potřeba

EV3 blok

Cíl lekce

- Připojit se k EV3bloku a postat mu instrukci.
- Seznámit se s programovacím prostředím.
- Naučit se práci s proměnnými a aritmetickými výrazy.

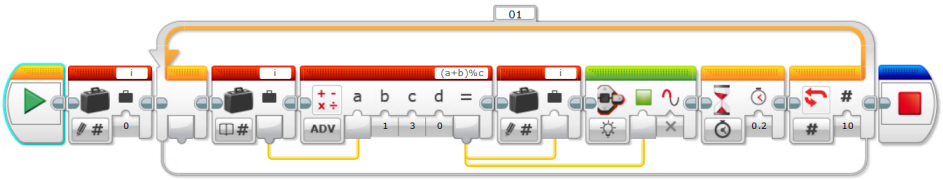
Zadání

Nechte cyklicky měnit barvy na bloku.

Časová dotace

90 min. Samotný program není náročný, ale pochopení programovacího prostředí nelze urychlit.

Řešení



Obrázek 4.1: Program pro Mindstorms na změnu barvy.

Struktura programu:

- Nastaví proměnnou i na 0
- V cyklu 10x
  - Načte proměnnou i
  - Předá ji výrazu, který ji inkrementuje o 1 a provede modulo 3
  - Předá výsledek jako novou hodnotu i a zároveň ji pošle na zobrazení
  - Zobrazí barvu podle hodnoty i
  - Čeká 0.2 sekundy

Hodnota proměnné i nabývá v cyklu hodnot: 1 2 0 1 2 0 1 2 0 1.

Debata

Jak moc se liší možnosti v programovacím prostředí Mindstorms od WeDo?

Porovnejte zkušenost s prostředím LabView s jinými programovacími jazyky, přičemž mějte na mysli, že toto prostředí je optimalizované pro práci měřících systémů a zařízení.

Jak se rozšiřují možnosti robota Mindstorm, když nahrává programy přímo do paměti bloku a nemusí být spojen při běhu s počítačem?

Zapřemýšlejte, jak by s výhodou šlo užít robota, kterého mohu přinést do výuky a využít předdefinovanou funkci. A nechat jej třeba zdánlivě bez dozoru projíždět chodby školy.

4.2 Lekce houpačka pro chytré hodinky

Co je potřeba

EV3 blok + motor/motory + dílky na stavění + chytré hodinky

Cíl lekce

- Propojení robota a jiné techniky.
- Seznámení se s konsktrukčními možnostmi v Lego Technic.
- Ozkoušení práce s motory Mindstorm.

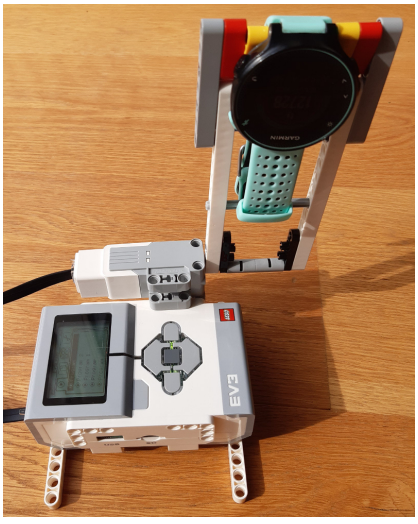
Zadání

Chytré hodinky načítají kroky uživateli. V některých zemích je dokonce zvykem hledět na výsledky kroků pro potřeby zdravotnictví či zaměstnavatele. Proto vznikly kolébky, které mohou kroky nasimulovat. Jejich simulace je jen naprogramování nepodmíněné sekvence pohybů s pevným počtem opakování.
Postavte houpačku, na kterou je možné umístit chytré hodinky, a naprogramujte ji, aby je houpala a tím se na hodinkách přičítaly kroky.

Časová dotace

90 min

Jak postavit houpačku

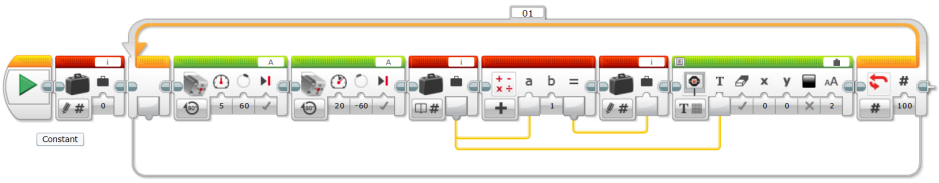


Obrázek 4.2: Blok s jedním motorem na pohyb hodinek.

Řešení

Struktura programu:

- Nastaví proměnnou i (počítadlo kroků) na nulu.
- Cyklus 100x opakuje
  - Otočí motor na jednu stranu (málo síly, pomáhá mu gravitace)
  - Otočí motor na druhou stranu
  - Načte proměnnou i
  - Inkrementuje i o 1
  - Uloží i
  - Vypíše i



Obrázek 4.3: Program pro pohyb hodinek.

Tip: výkon motoru je potřeba snížit, aby nedocházelo k překmitům.

Debata

Zobrazovací blok v navrženém řešení zobrazuje počet „nahoupaných“ kroků na bloku Mindstorms. Jaká je výhoda zobrazovat kroky na bloku a nikoli na obrazovce?

Blok Mindstorms nahraje celý program a může fungovat nezávisle na počítači — takže ačkoli je počítač vypnutý, je možné spustit úlohu přímo na bloku a sledovat, jak inkrementuje počet houpnutí.

4.3 Lekce autonomní vozidlo

Co je potřeba

EV3 blok + motor/motory + senzory + kola + ...

Cíl lekce

- Pokročilé stavební konstrukce Lego Technic.
- Zapojení senzorů a práce s nimi.
- Hraní si s autem.

Zadání

Postavte auto. Naprogramujte úlohy

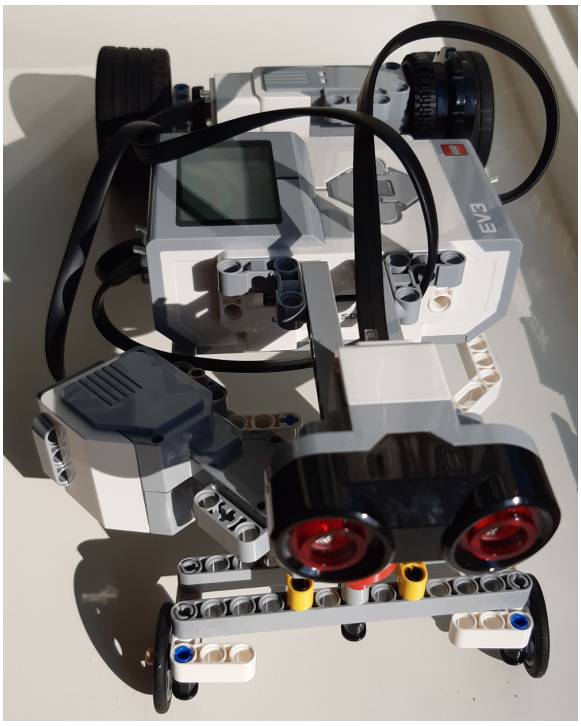
- Nenaraz do zdi (dojeď 5cm od zdi)
- Jeď jako had (ozkoušejte, jestli umíte zatáčet)
- Nenaraz do zdi - před zdí vycouvá a zabočí.
- Semafor (na červenou stojí, na zelenou jede)

Časová dotace

180 min

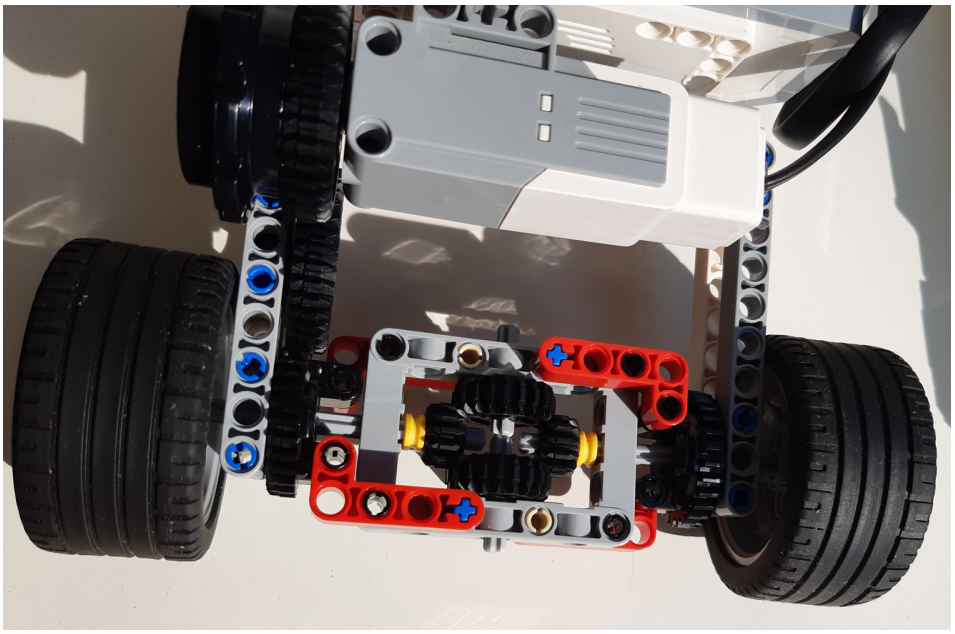
Jak postavit auto

U stavby auta z LEGO Mindstorms je potřeba si ujasnit, co vše nazýváme autem a jak moc chápeme konstrukci auta. Mindstorms má v sadě celkem 3 motory, dva stejně silné, jeden slabší. Napodobením klasické konstrukce auta se spalovacím motorem by mohl vzniknout robot s jednou hnanou osou a přídavným motorem na natáčení předních kol. Ale dva stejné motory v sadě nabízí možnost mít dvě hnaná kola, zatáčet pomocí pohonu jednoho či druhé kola — či ještě lépe pohonu jednoho kola a opačného pohonu druhého kola.
Druhý model má bohatou podporu v návodech LEGO Mindstorms, kde je popsán jako Driving Base, případně doplněné senzorem podle vlastního uvážení.
`https://education.lego.com/en-us/support/mindstorms-ev3/building-instructions`
První model auta, tedy osazeného jedním motorem na pohon kol a jedním motorem na zatáčení předních kol a osazeného ultrazvukovým senzorem na určení vzdálenosti a senzorem na určení barev najdete v tomto textu:

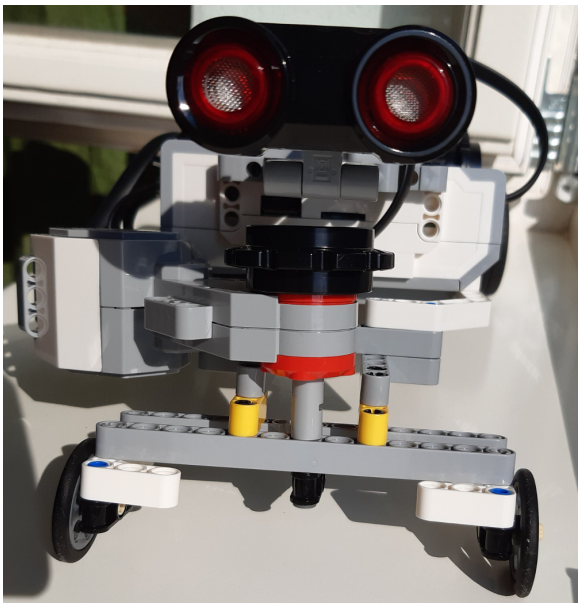


Obrázek 4.4: Pohled zepředu na ultrazvukový senzor a druhý motor k ovládání předních kol.

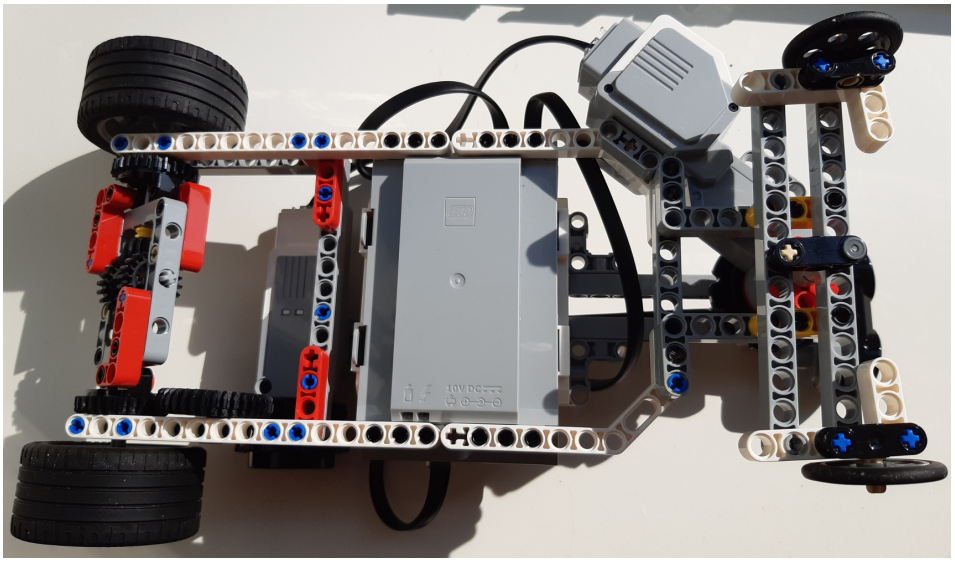
Obrázek 4.5: Pohled shora, viditelné oba motory.



Obrázek 4.6: Detail diferenciálu na zadních kolech. Není nutné, autor se chtěl vytáhnout.



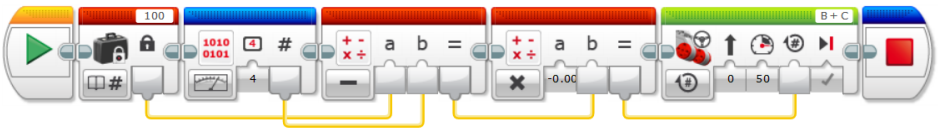
Obrázek 4.7: Detail vedení předních kol.



Obrázek 4.8: Pohled zespodu.

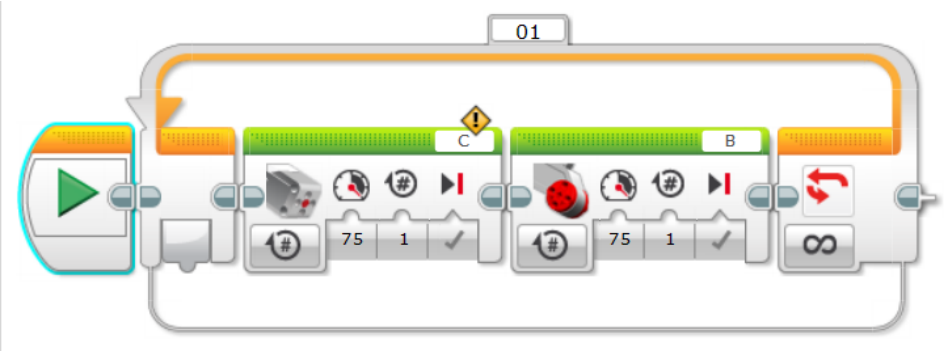
Řešení

Následující řešení jsou uvedeny pro model auta postaveného podle návodu na Driving base, tedy každé kolo je ovládáno svým motorem. Pro model s jedním motorem hnacím a druhým pro zatáčení, se programy změní v použití bloku na ovládání motoru určeného pouze pro jeden motor a ne pro dva. Paradoxně, u pohybu „Jako had“, se kód téměř nezmění. Bude jen uveden dvakrát za sebou, aby jednou zatáčel vlevo a pak vpravo.
Úloha Nenaraz do zdi je řešena jinak než při práci s WeDo: S WeDo uživatel může spustit motor a nechat senzory, ať vrací hodnoty a přizpůsobuje tomu práci motoru. Senzor na určení vzdálenosti pro Mindstorms dokáže daleko větší vzdálenosti, v tomto případě se tedy robot nejprve podívá, jak je překážka daleko, podle toho naplánuje, kolik otočení motoru/kol to dělá a jede. Logika takového program 4.9 je natolik jednoduchá, že nejsou potřeba podmínky ani cykly.



Obrázek 4.9: Úloha Nenaraz do zdi.

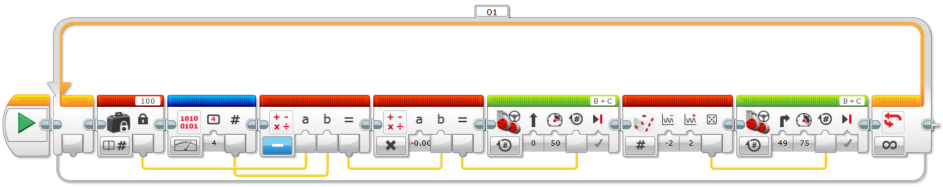
Zelený blok na ovládání motorů je použit ve variantě, kdy ovládá oba motory zároveň, první parametr je nulový, tedy oba motory jsou zatěžovány stejně.
Nenaraz do zdi je také možné řešit pomocí dotykového senzoru, jakmile dojde k jeho stisknutí, dojde k vypnutí motorů. Není sice zcela splněno zadání nenarazit do zdi, ale je to dost blízko řešení.
Jízda klikatě 4.10 je provedena střídavým zapínáním motorů.



Obrázek 4.10: Úloha Jeď jako had.

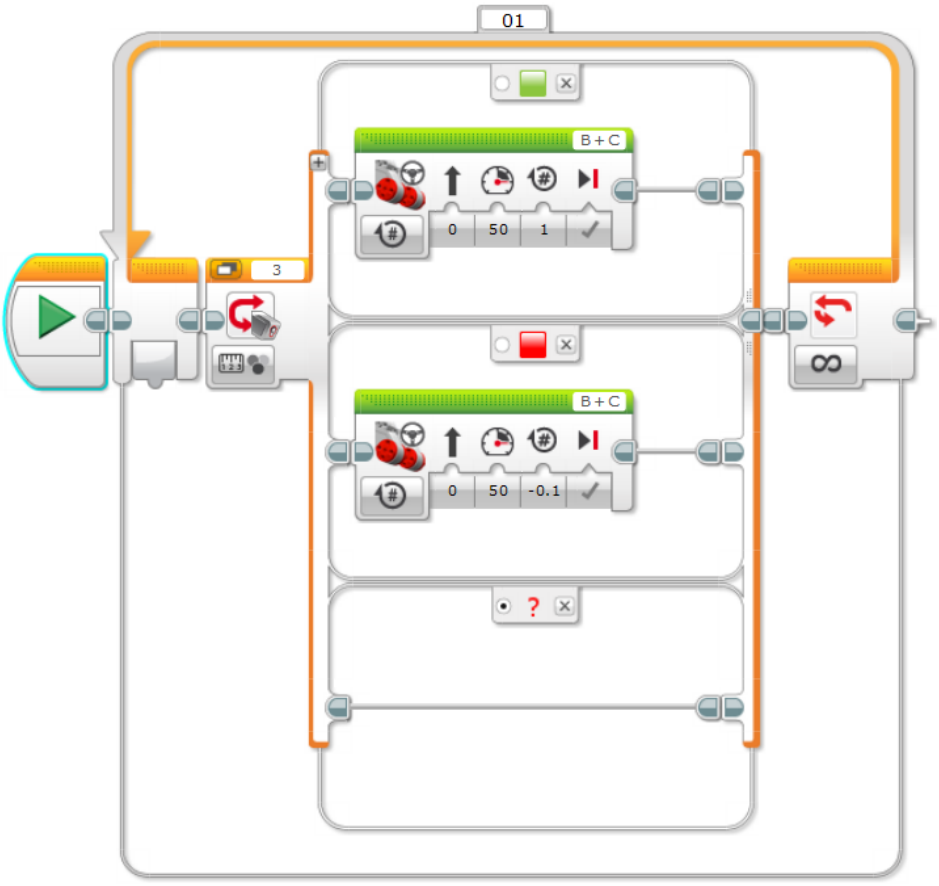
Jeď náhodně a nenaraz do zdi je částečně inspirováno pohybem robotického vysavače nebo robotické sekačky. V případě, že je vyřešena úloha Nenaraz do zdi, je možné použít předchozí program a vložit jej do nekonečného cyklu, kdy

po dojezdu ke zdi se robot otočí náhodně 4.11. Poslední blok cyklu otáčí oběmi motory v opačném směru.



Obrázek 4.11: Úloha Jeď náhodně a nenaraz do zdi.

Úloha s jízdou podle semaforu, je zde 4.12 implementována tak, že robot po zjištění zelené barvy jede vpřed, při zjištění červené bravy mírně couvne a nevidí-li ani červenou nebo zelenou, pak nedělá nic.



Obrázek 4.12: Úloha Jeď podle semaforu.

Debata

V řešení typu Nenaraz do zdi je cesta naplánována před tím, než robot vyrazí v daném směru. Nereaguje tedy na překážky, které se v cestě objeví až po začátku cesty.

Reagovat na podněty již při běhu motoru, je možné při využití paralelismu, tedy možnosti mít spuštěné dva příkazy (více příkazů) v jedné chvíli. Tématu se věnuje materiál
`http://www.legoengineering.com/design-patterns-in-ev3/`

Zapřemýšlejte nad úlohami, kde by více podobných aut mohlo spolu interagovat soutěživým způsobem.

Nabízí se závody vozů v rychlosti, přesnosti projetí dané dráhy, také je možné jít i do více agresivních úloh jako je přetlačování nebo demoliční derby.

4.4 Lekce Nejzbytečnější stroj na světě

Co je potřeba

WeDo SmartHub + motor + senzor pohybu + ramena ...

Cíl lekce

- Seznámení se s klasickou úlohou robotiky.
- Nalezení pamarametrů úlohy, tj. kalibrace.

Zadání

Koncept nejzbytečnějšího stroje (anglicky the most useless machine) je popsán zde: `https://en.wikipedia.org/wiki/Useless_machine`
Jde o stroj, který má jeden mechanický přepínač on/off. Jakmile jej přepnete na „on“, sám se přepne do polohy „off“.

1. Sestavte stroj s mechanickým přepínačem, senzorem, který zjišťuje, ve které pozici je přepínač, a mechanismum, který navrátí přepínač do polohy vypnuto.
2. Naprogramujte stroj tak, aby po přepnutí přepínače do polohy zapnuto, se přepínač přepnul zpět na vypnuto.

Časová dotace

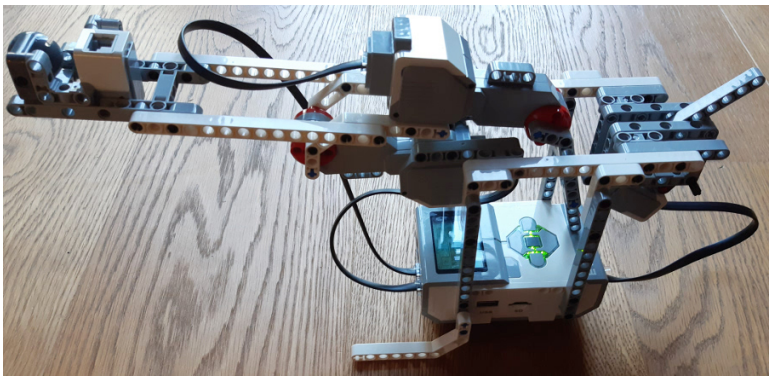
90 min

Jak postavit nejzbytečnější stroj na světě

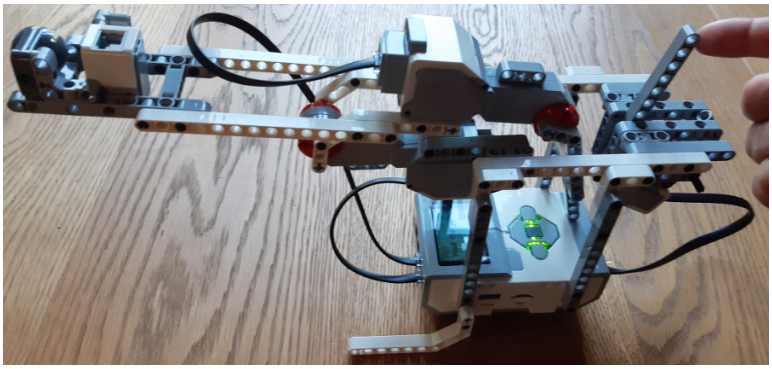
V následujícím řešení je postaven stroj, který přepínačem při překnutí na „on“ stiskne dotykový senzor. Ten pomocí dvou motorů rozpohybuje rameno s prstíkem, přepne přepínač zpět do polohy „off“ a vrátí se na své místo.

Řešení

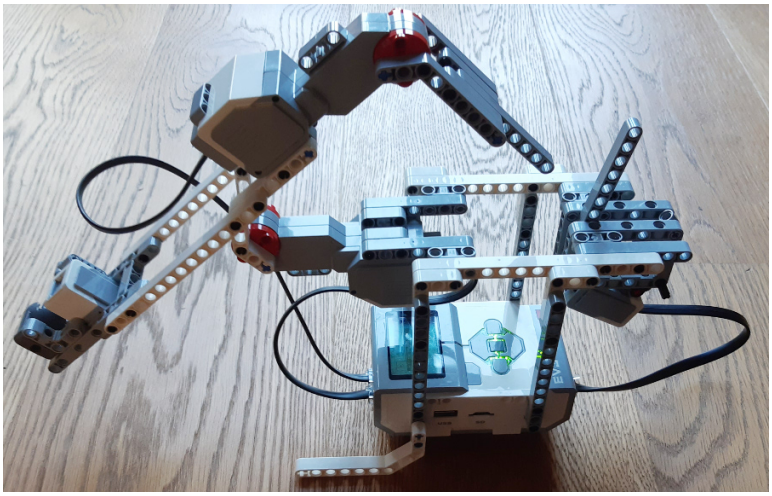
Stroj v nekonečné smyčce sleduje, jestli je zapnuto dotykové čidlo. Úkolem pro programátora je pak najít takovou sekvenci spouštění motorů, aby došlo k žádaného pohybu prstíku.
Navržený robot má na zadní části připravené čidlo polohy pro případ, že by se dostal do nějaké nestandardní polohy.



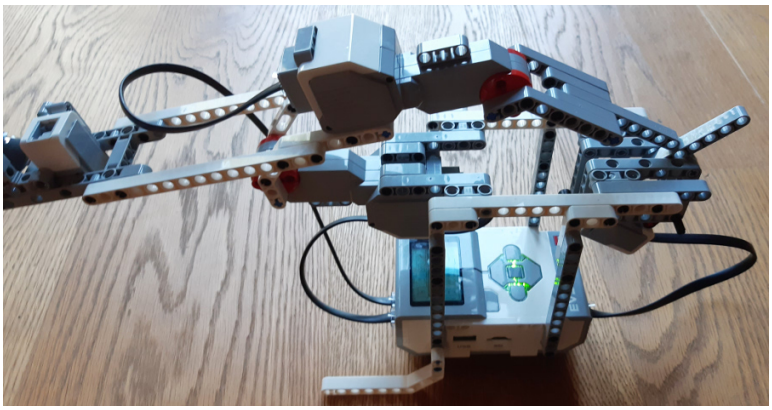
Obrázek 4.13: Zbytečný stroj, poloha off, stroj je v klidu.



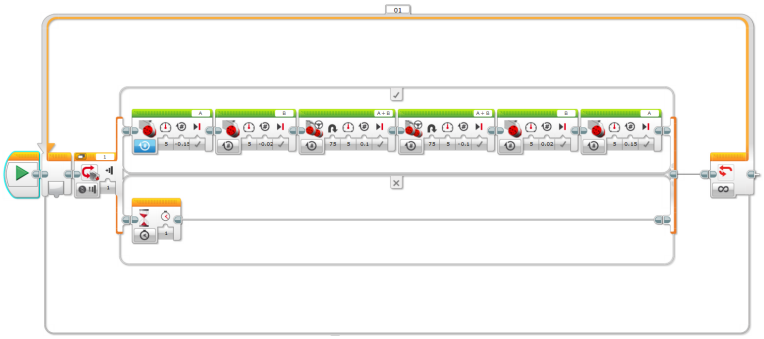
Obrázek 4.14: Zbytečný stroj, přepnutí do polohy on.



Obrázek 4.15: Zbytečný stroj, je přepnutý do on, zvedá rameno.



Obrázek 4.16: Zbytečný stroj, přepíná se do off.



Obrázek 4.17: Program pro zbytečný stroj

Debata

K čemu je tento stroj?

Na nic. Ale to nic je sofistikované nic, ve kterém se snoubí vtip, lenost, vychytralost, chuť nacházet problémy tam, kde původně vůbec nebyly, a touha je řešit. Je to celé tohle hraní s roboty v kostce.