

EV3

GRAFICKÉ PROGRAMOVÁNÍ

Robotický LEGO seminář, 27. 1. – 28. 1. 2017, ČVUT FEL, Praha

Štěpánka Baierlová
Jan Preclík

JAK PROGRAMOVAT?

- **V grafickém prostředí – pro začátečníky**
 - EV3 software → pro NXT i EV3, volné
(<https://www.lego.com/cs-cz/mindstorms/downloads/download-software>)
 - založeno na LabView
- „Klasickým“ programovacím jazykem
 - zápis příkazů v textové podobě
 - Robot C, komerční → C
 - Java + vývojové prostředí Eclipse, Netbeans

HLAVNÍ OKNO EV3

Startovní kostka

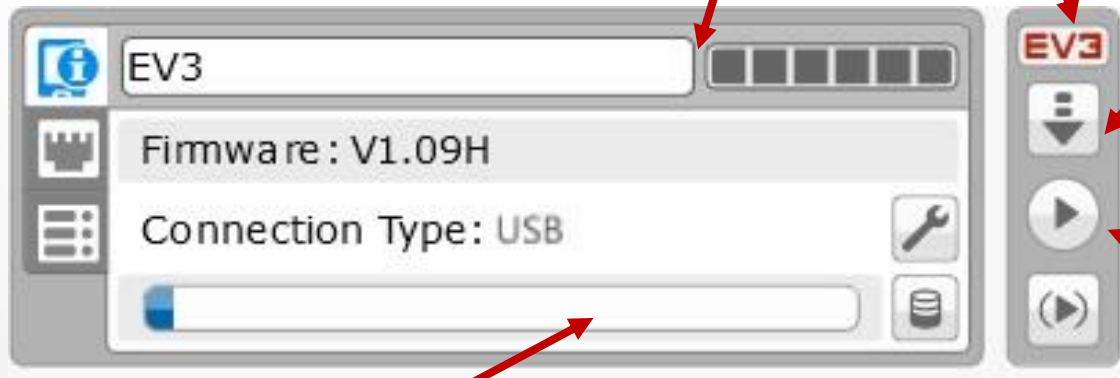
Palety kostiček = příkazů

- Action = výstup (pohyb, displej, zvuk)
- Flow Control = běh programu (cykly, podmínky)
- Sensor = senzory, vstupy
- Data Operations = proměnné, výpočty
- Advanced = soubory, Bluetooth...
- My Blocks = vlastní kostky (podprogramy)

OVLÁDÁNÍ KOSTKY

Typ připojené kostky (EV3, NXT)

Název kostky + stav baterie



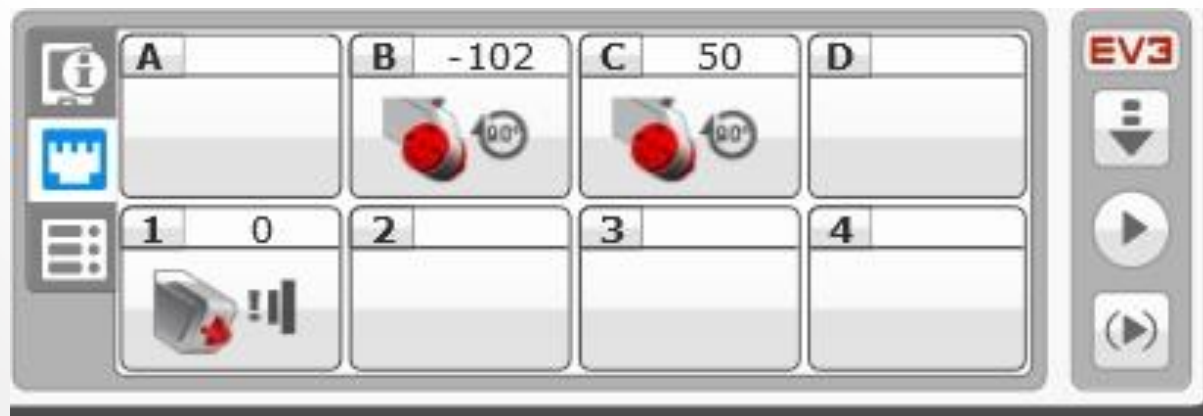
Nahrát program do kostky

Nahrát a spustit

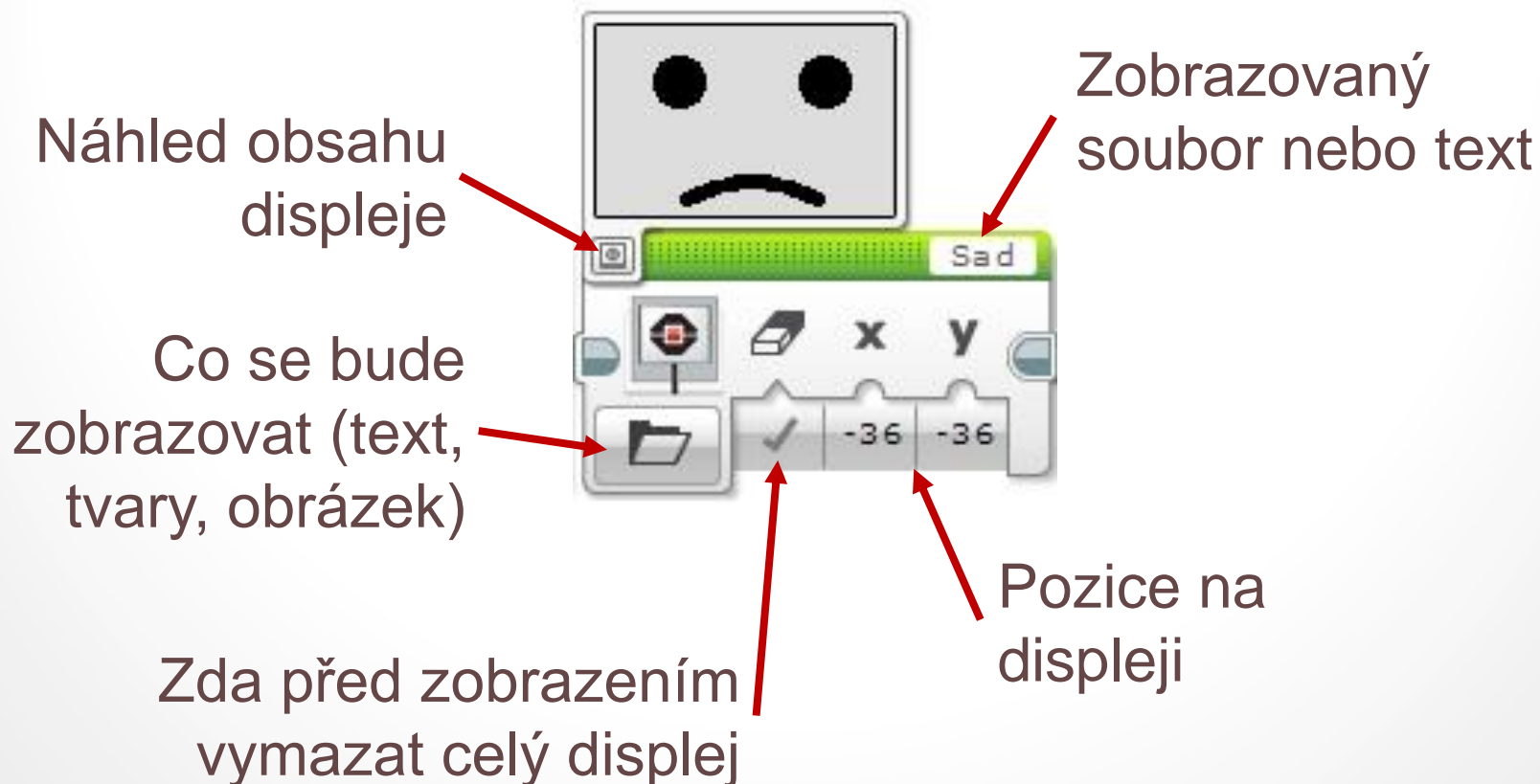
Zaplnění paměti kostky + možnost editace obsahu paměti

PORT VIEW

- Na kartě Port View můžeme velice snadno odečítat hodnoty ze vstupů i výstupů
- **Auto ID** = připojená zařízení jsou ihned automaticky rozpoznávána



VÝPIS NA DISPLEJ (DISPLAY)



VÝPIS NA DISPLEJ (DISPLAY)



- Po zobrazení je nutno počkat, jinak program skončí a nic neuvidíme

ČEKÁNÍ (WAIT)



- Čekat lze na různé hodnoty senzorů nebo po zadaný čas



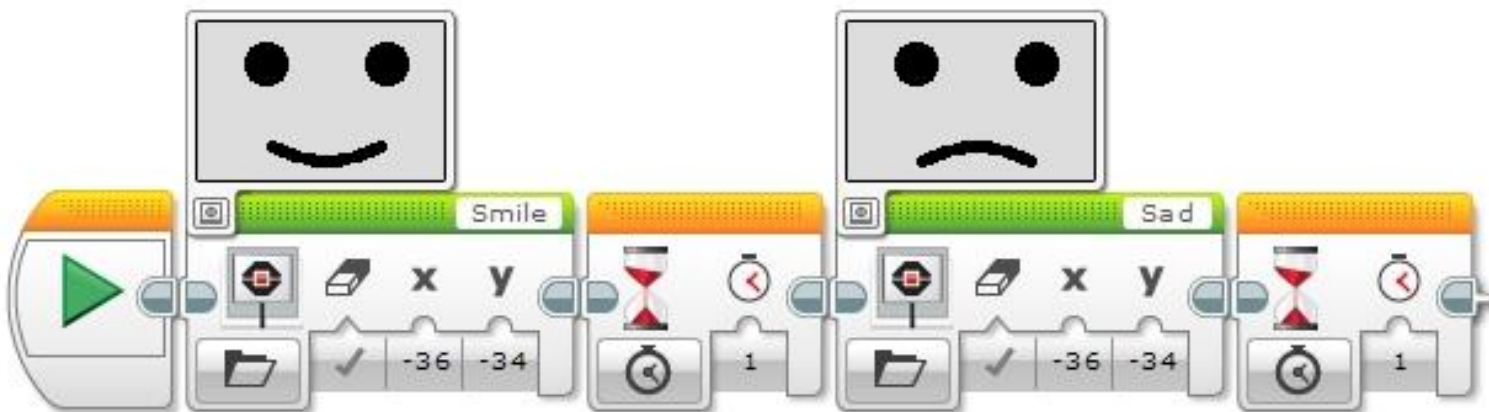
Na co se bude čekat

a jak dlouho nebo na jakou hodnotu

CVIČENÍ 01

- Ať se vaše kostička nejprve usměje (Smile),
- po 1 sekundě ji to přestane bavit,
- zamračí se (Sad),
- tak vydrží další sekundu
- a program skončí.

CVIČENÍ 01 – ŘEŠENÍ



ZVUK (SOUND)



Přehrávaný soubor

Co se bude přehrávat (soubor nebo tón)

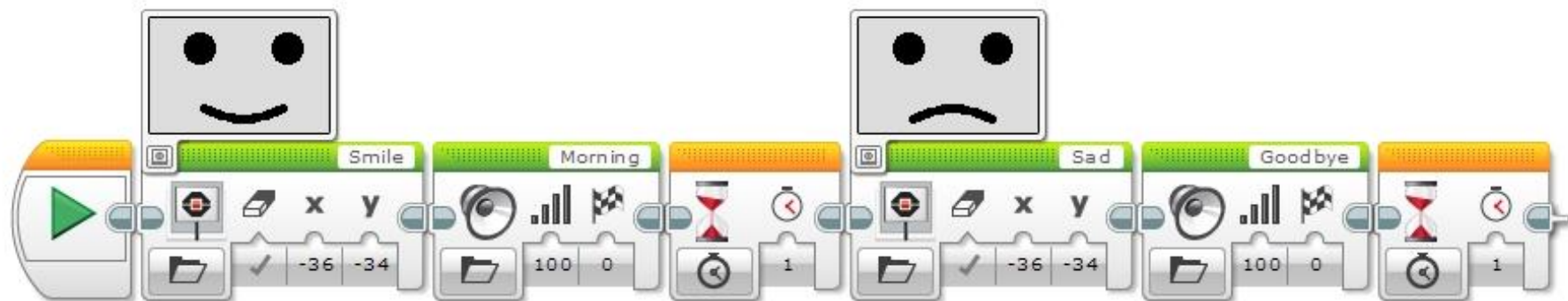
Hlasitost

Co se stane po přehrávání (0 = čeká na dokončení zvuku a pak program pokračuje, 1 = začne přehrávat zvuk a program běží dál, 2 = přehrává zvuk opakovaně a program běží dál)

CVIČENÍ 02

- Ať se vaše kostička nejprve usměje (Smile)
- a slušně pozdraví (Morning),
- po 1 sekundě ji to přestane bavit,
- zamračí se (Sad)
- a rozloučí se (Goodbye),
- tak vydrží další sekundu
- a program skončí.

CVIČENÍ 02 – ŘEŠENÍ



VÝPIS NA DISPLEJ A ZVUK – NÁMĚTY

- Komiks...
- Usmívat se dokud není stisknuto nějaké (zvolené) tlačítko na kostičce.
- Mračit se dokud někdo nepootočí levým kolem o celou otočku.

MOTOR (LARGE/MEDIUM MOTOR)



Akce

Který motor (resp. port, do kterého je zapojen)

Síla = „rychlost“

Počet (sekund, stupňů, otáček)

Co se stane po dokončení akce:
Brake = motor bude zabržděn,
Coast = „něžné zastavení“

MOTOR (LARGE/MEDIUM MOTOR)



Akce

- **Off** = motor bude zastaven
 - O způsobu zastavení rozhoduje Brake/Coast.
- **On** = motor bude zapnut a poběží tak dlouho, až ho jiná kostička vypne
 - Program mezitím poběží dál!
- **On for Seconds** = motor poběží uvedený čas
 - Na této kostičce se čeká, až běh motoru bude ukončen!
- **On for Degrees** = motor se otočí o zadaný počet stupňů, na této kostce se bude čekat tak dlouho, až se motor o zadaný úhel otočí, na to pozor například při zablokování motoru!
- **On for Rotations** = motor se otočí o zadaný počet otáček, pozor viz výše!

MOTOR (LARGE/MEDIUM MOTOR)

- Pokud je v daném portu zapojen jiný motor (místo Large Medium atp.), bude u kostičky výstraha!



POHYB (MOVE STEERING)



- Kostička ovládá dva motory a umožňuje jejich současný pohyb a zatačení změnou poměru jejich otáček
 - intuitivnější ovládání
- Podobnou funkci má i kostička **Move Tank**, kde lze přímo nastavit síly pro jednotlivé motory.
 - toto ovládání může být někdy přesnější

POHYB (MOVE STEERING)



Akce



Do kterých portů jsou motory zapojeny

Zatáčení = poměr mezi otáčkami obou motorů

POHYB (MOVE STEERING)



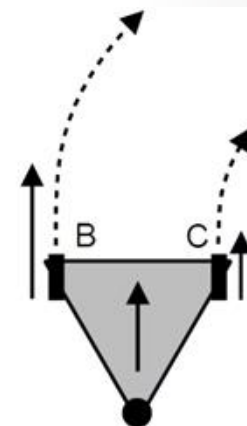
Jízda rovně



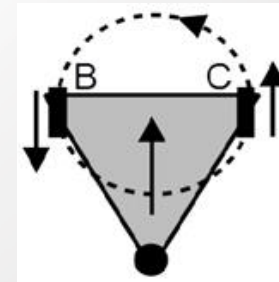
Pozvolné zatáčení v jednom směru



Zatáčení jedním kolem



Zatočení na místě



CVIČENÍ 03

- Naučte robota jezdit do čtverce.
 - Nejprve ho naučte se otočit přesně o 90° , to není tak jednoduché.
 - přesně 90° to nebude nikdy ☹ ... možná později při využití Gyro senzoru

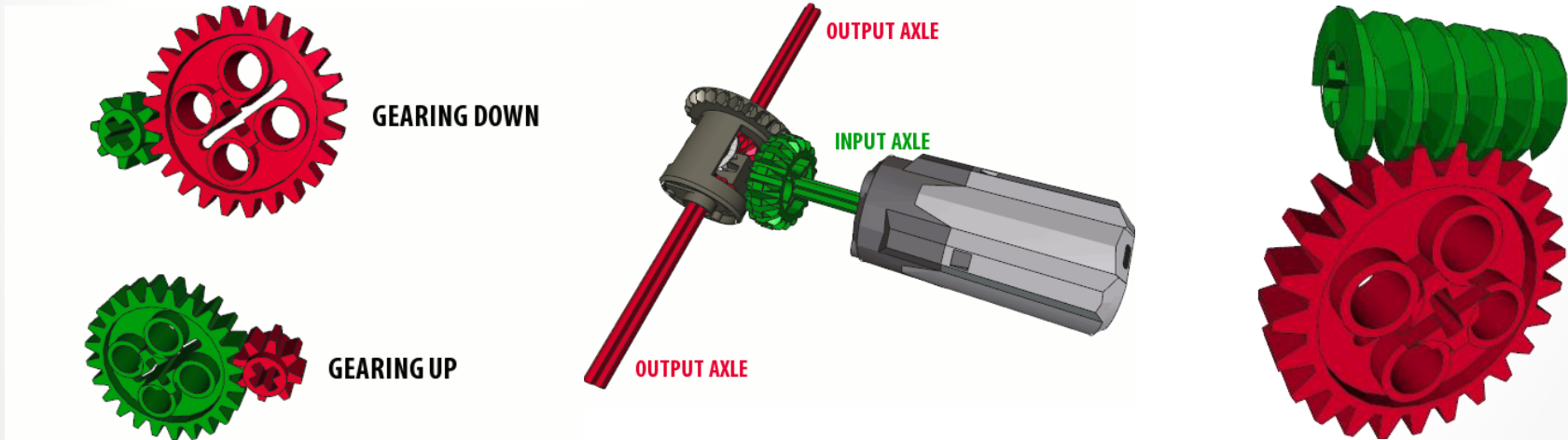
CVIČENÍ 03 – ŘEŠENÍ



- Na čem všem záleží, o kolik se robot skutečně otočí?
 - Na velikost kol?
 - Na vzdálenosti kol od sebe?
 - Na podkladu?
 - Na rychlosti?

POHYB – NÁMĚTY

- Ozubená kola a převody



- kalkulátor/simulace na <http://gears.sariel.pl>

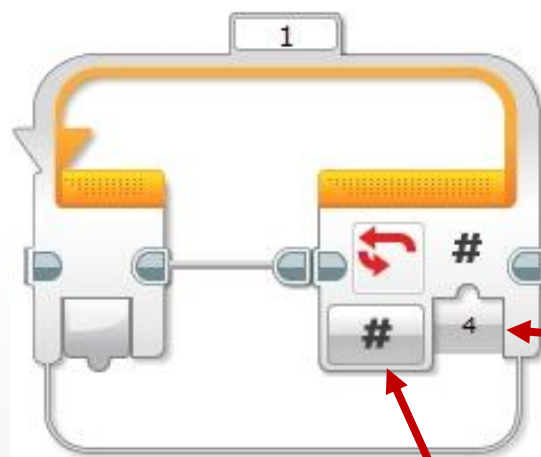
POHYB – NÁMĚTY

- Rychlostní závody (sice jen jízda rovně, ale to není tak jednoduché) + třeba zastavení na určitém místě
- Závody do vrchu (nakloněná plošina) + případně robot silák (táhne zátěž, vozík, pet láhev s vodou...)
- Přetahovaná/přetlačovaná robotů...

CYKLUS (LOOP)



- Opakování
 - předem zadaný počet opakování
 - po zadaný čas
 - dokud nebude zvolená hodnota senzoru



Hodnota (podmínky,
počet opakování...)

Podmínka cyklu

CVIČENÍ 03 – ŘEŠENÍ S CYKLEM



DOTYKOVÝ SENZOR (TOUCH)



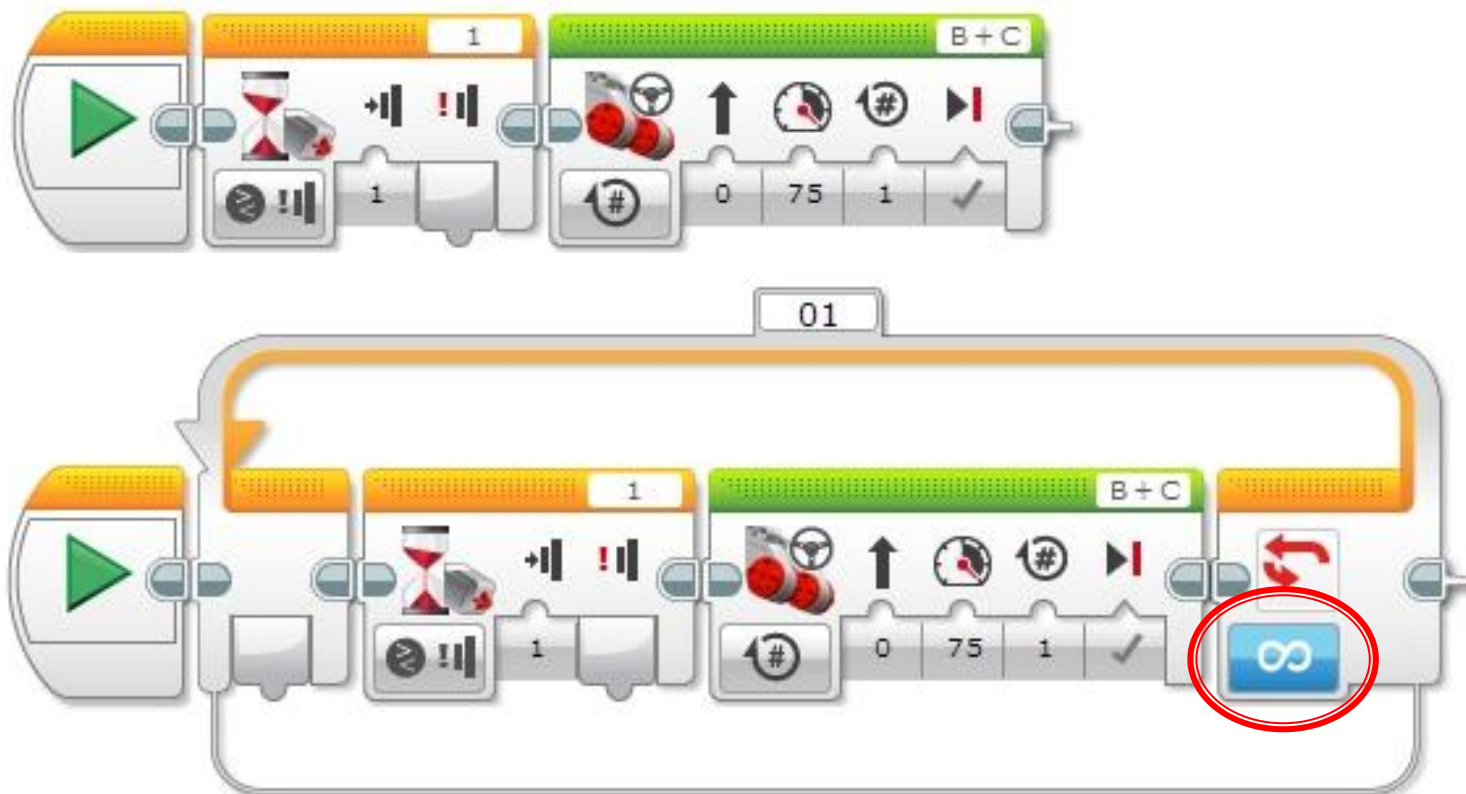
- „tlačítko“
- Lze použít v cyklu, čekání a podmínce
- Lze testovat
 - uvolnění (Released)
 - stisknutí (Pressed)
 - stisknutí a uvolnění (Bumped)

CVIČENÍ 04 – DOTYKOVÝ SENZOR



- Robot čeká na stisk tlačítka,
- po stisku tlačítka popojede
- a zase čeká...

CVIČENÍ 04 – ŘEŠENÍ



- Nekonečný cyklus musíme ukončit ručně!

DOTYKOVÝ SENZOR – NÁMĚTY

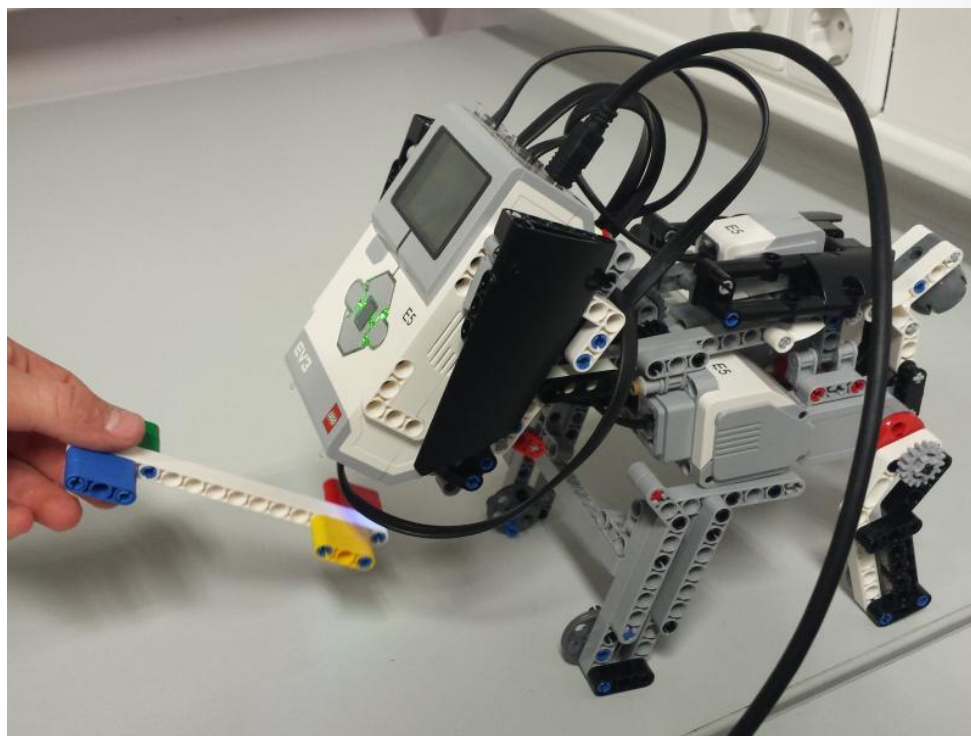


- Robot se po stisku tlačítka zasměje, zamračí, vydá zvuk...
- Nárazník – robot jede rovně tak dlouho, až narazí na překážku (tlačítko stisknuto), pak se zastaví... couvne, otočí...

DOTYKOVÝ SENZOR – NÁMĚTY



- Zvířátko, které čeká na pohlázení, pak zavrčí, pohne se...
- Pasticčka na myši

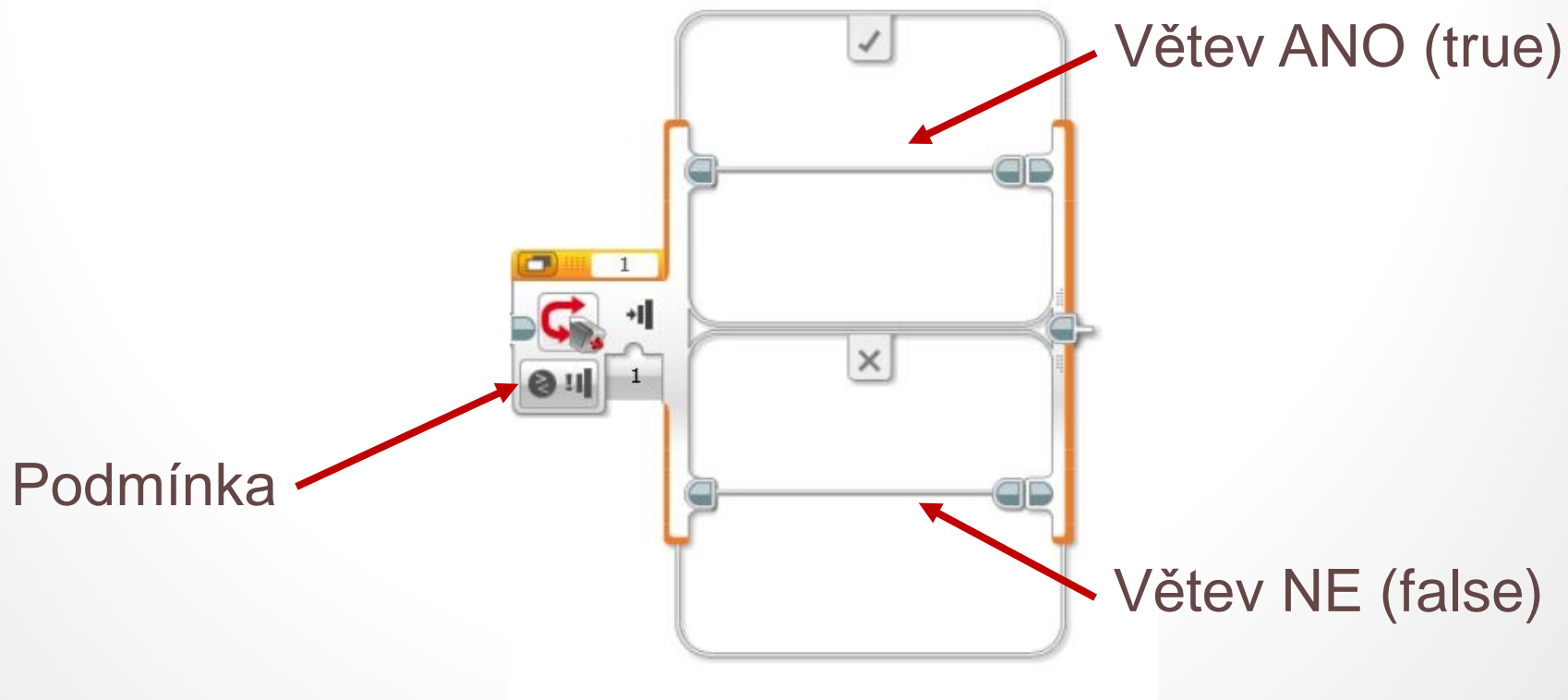


KELLER, Klaus-Dieter. *Tag A3 Puppy (Hund)* [online]. In: . [cit. 2017-02-05]. Dostupné z: https://commons.wikimedia.org/wiki/File:Tag_A3_Puppy.jpg, licence Creative Commons Attribution-Share Alike 4.0 International

PODMÍNKA (SWITCH)



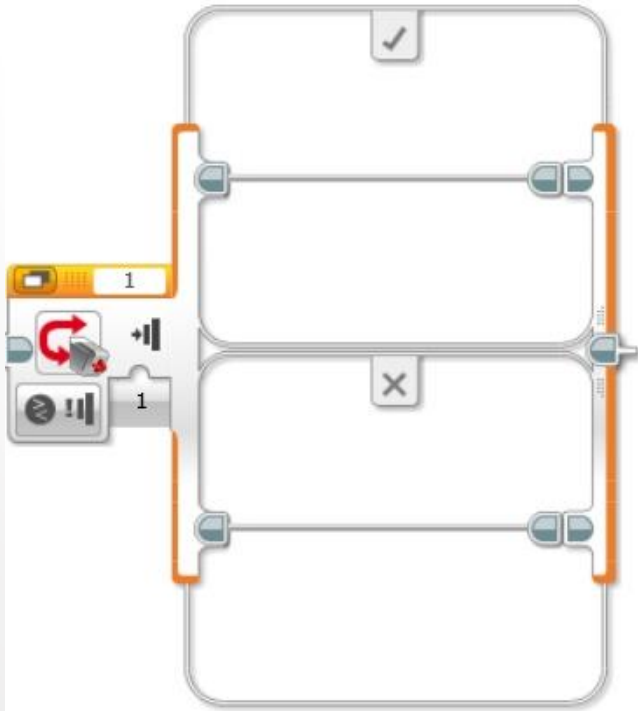
- Podmínka



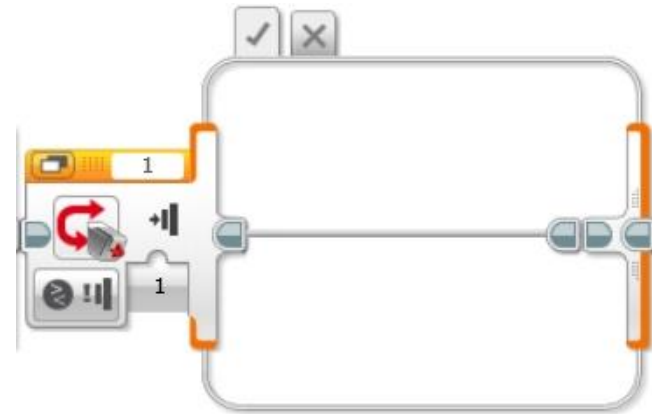
PODMÍNKA (SWITCH)



- Zobrazení **Tabbed View** lze využít pro úsporu místa



Flat view

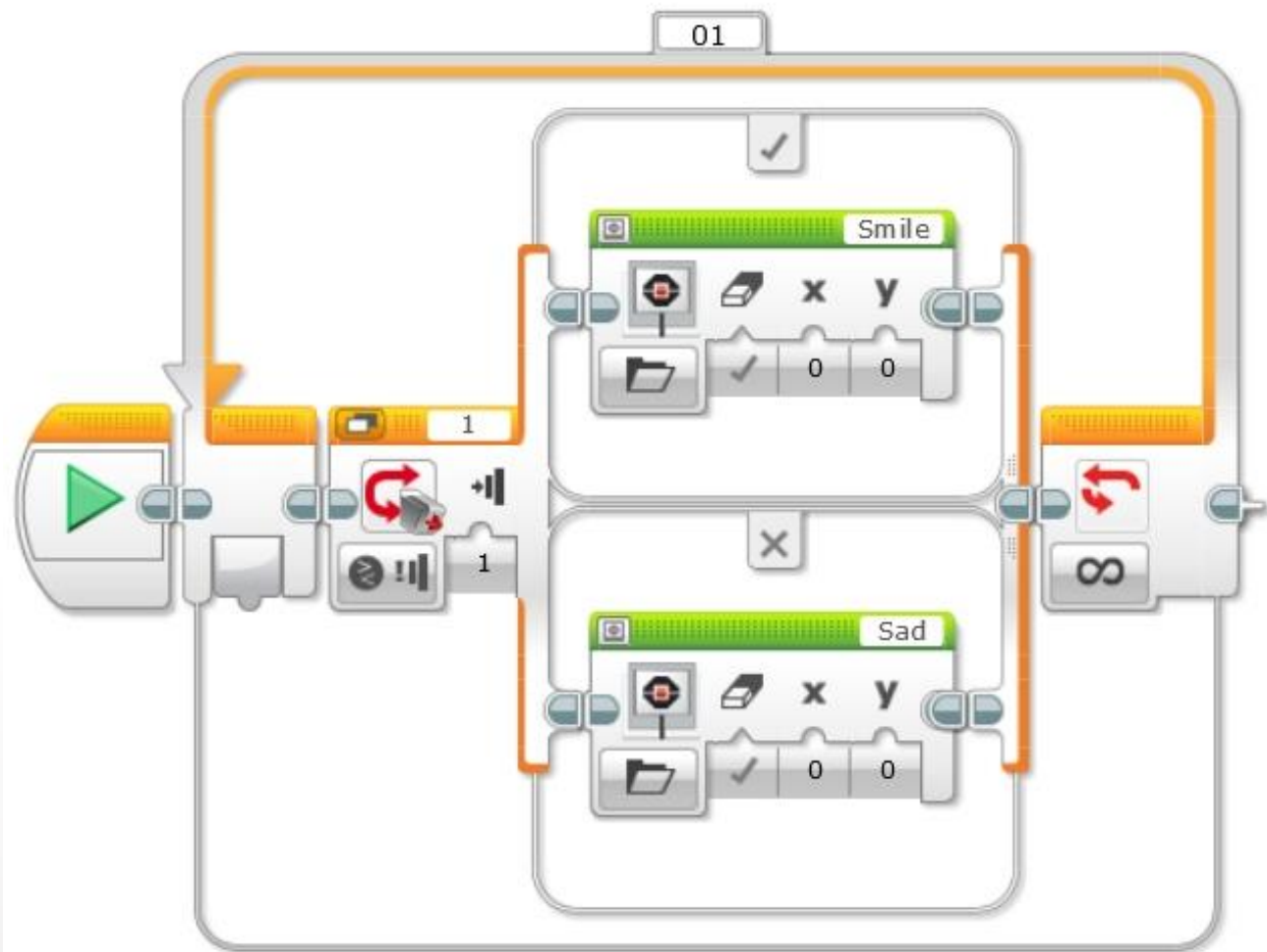


Tabbed View

CVIČENÍ 05 – PODMÍNKA

- Když je tlačítko v portu 1 stisknuto, robot se usmívá, jinak se mračí.

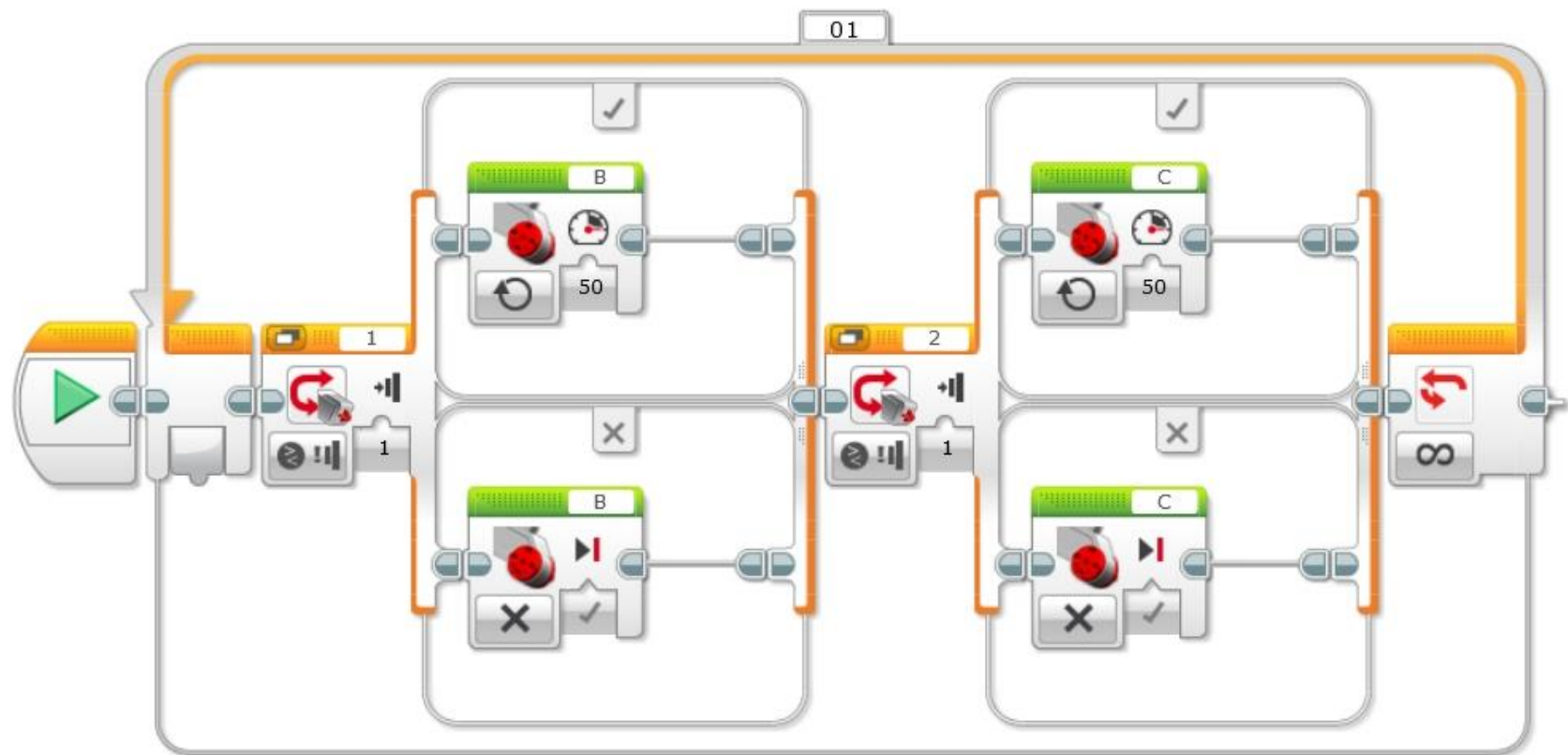
CVIČENÍ 05 – ŘEŠENÍ



CVIČENÍ 06 – DÁLKOVÉ OVLÁDÁNÍ

- Tlačítka v portech 1 a 2,
- každé ovládá jeden motor – když je stisknuto, motor se točí.

CVIČENÍ 06 – ŘEŠENÍ



ULTRAZVUKOVÝ SENZOR (ULTRASONIC)



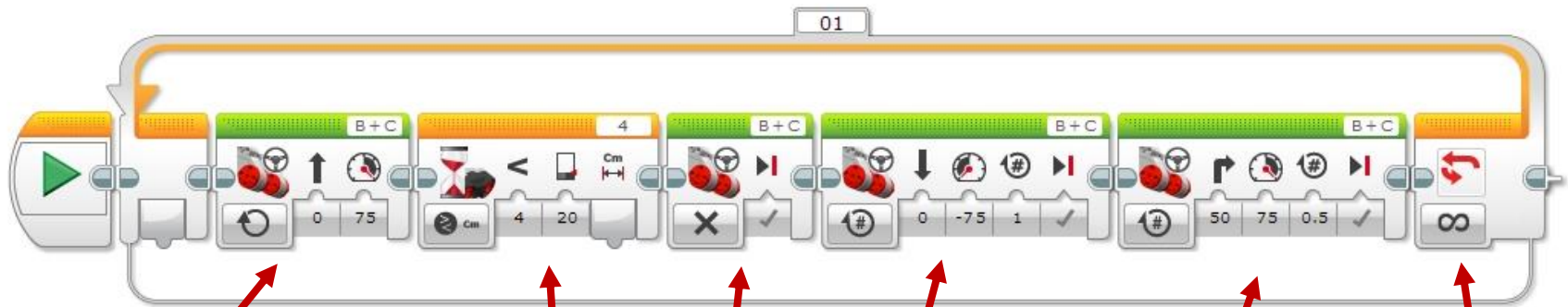
- svítí = **Measure mode** = měří vzdálenost
 - vysílá i přijímá
- bliká = **Presence mode** = detekce jiného senzoru
 - jen poslouchá, nevysílá
 - vhodné třeba na robotické sumo = pozná, že ho protivník sleduje

CVIČENÍ 07 – ULTRAZVUKOVÝ SENZOR



- Robot jede dopředu,
- jakmile se přiblíží k překážce,
- couvne, pootočí se
- a zase pojede dopředu.

CVIČENÍ 07 – ŘEŠENÍ



1. Spustíme motory (B+C), necháme je běžet a program pokračuje dál.

2. Čekáme (a motory běží), až bude vzdálenost menší než 20 cm,

3. poté motory zastavíme,

4. couvneme

5. a pootočíme se.

6. A opakujeme opět od začátku (nekonečný cyklus).

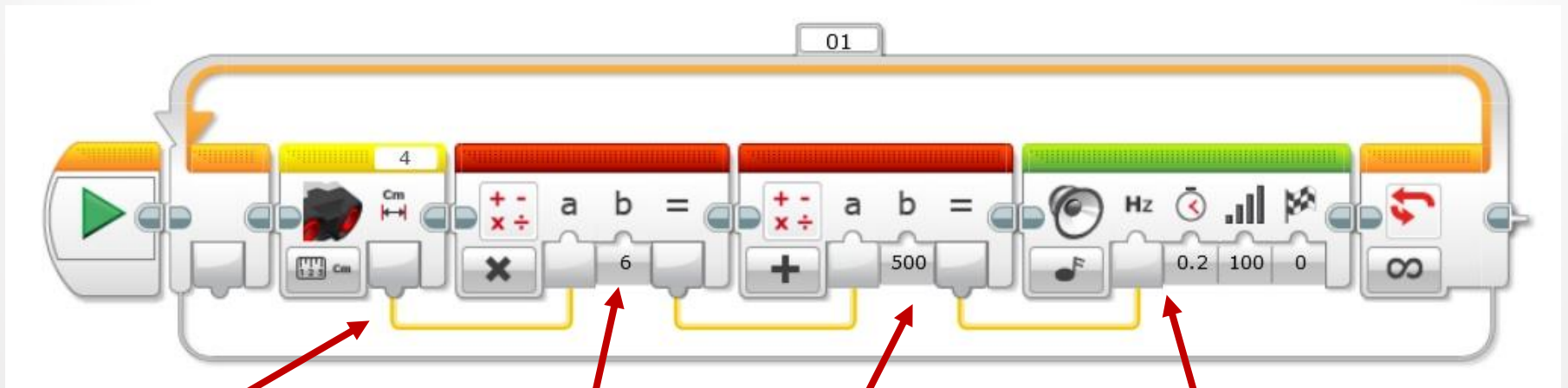
ULTRAZVUKOVÝ SENZOR – NÁMĚTY



- Když se někdo přiblíží, zasmát se
- Pasticčka na myši, hlídací pes...
- Hledání dveří, hledání cesty v bludišti (zed' po pravé ruce)
- Hledání protivníka – jen poslouchám ostatní UZ senzory..., ty případně mohou sloužit jako majáky
- něco jako „theremin“

VÝPOČTY – NĚCO JAKO „THEREMIN“

- Podle vzdálenosti od UZ senzoru hrajeme určitý tón



1. Vzdálenost od UZ senzoru v cm...

2. vynásobíme 6, ...

3. k výsledku přičteme 500...

4. a tón o této frekvenci přehrajeme.

SVĚTELNÝ SENZOR (COLOR)



- **Color Mode** = rozlišuje 7 barev
 - černá, bílá, zelená, modrá, červená, žlutá, hnědá, žádná
- **Reflected Light** = odražené světlo
 - 0 = černá, 100 = bílá
- **Ambient Light** = okolní světlo

SVĚTELNÝ SENZOR (COLOR)



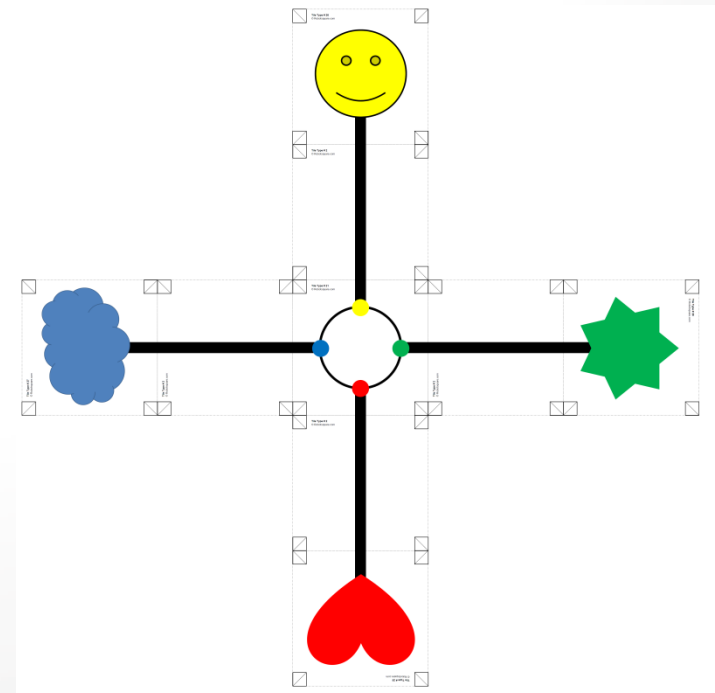
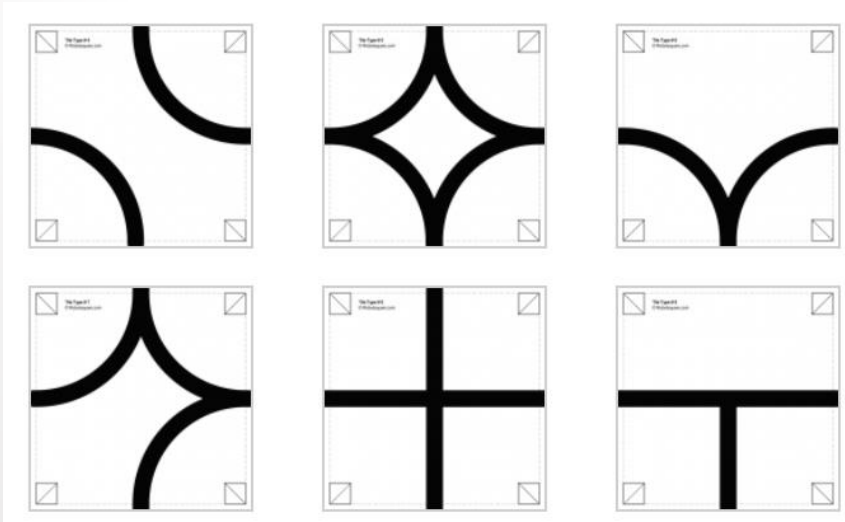
■ Color Mode

- Například orientace podle barev na křižovatkách ve městě: modrá → zahni doprava, žlutá → doleva, zelená → rovně atp.
- Tím, že robotu ukážeme papírek dané barvy, spustíme určitý program
- Třídíčka lentilek (kostiček Lega)

SVĚTELNÝ SENZOR (COLOR)



- pro tvorbu různých hříšť a čar
<http://robotsquare.com/2012/11/28/line-following/>



SVĚTELNÝ SENZOR (COLOR)



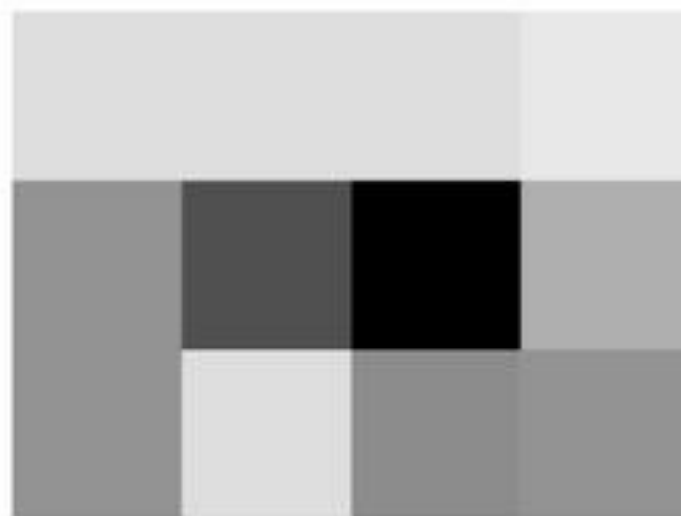
■ Ambient Light

- světelná závora, signalizace
- když se v místnosti rozsvítí/zhasne, robot něco udělá
- robot hledá nejtemnější kout
- robot jede za světlem



■ Reflected Light

- asi nejčastější použití, detekce černé čáry
- nutná kalibrace podle aktuálních podmínek
- námět – hledání „vody/ropy/nejvyššího vrcholu“ na mapě krajiny

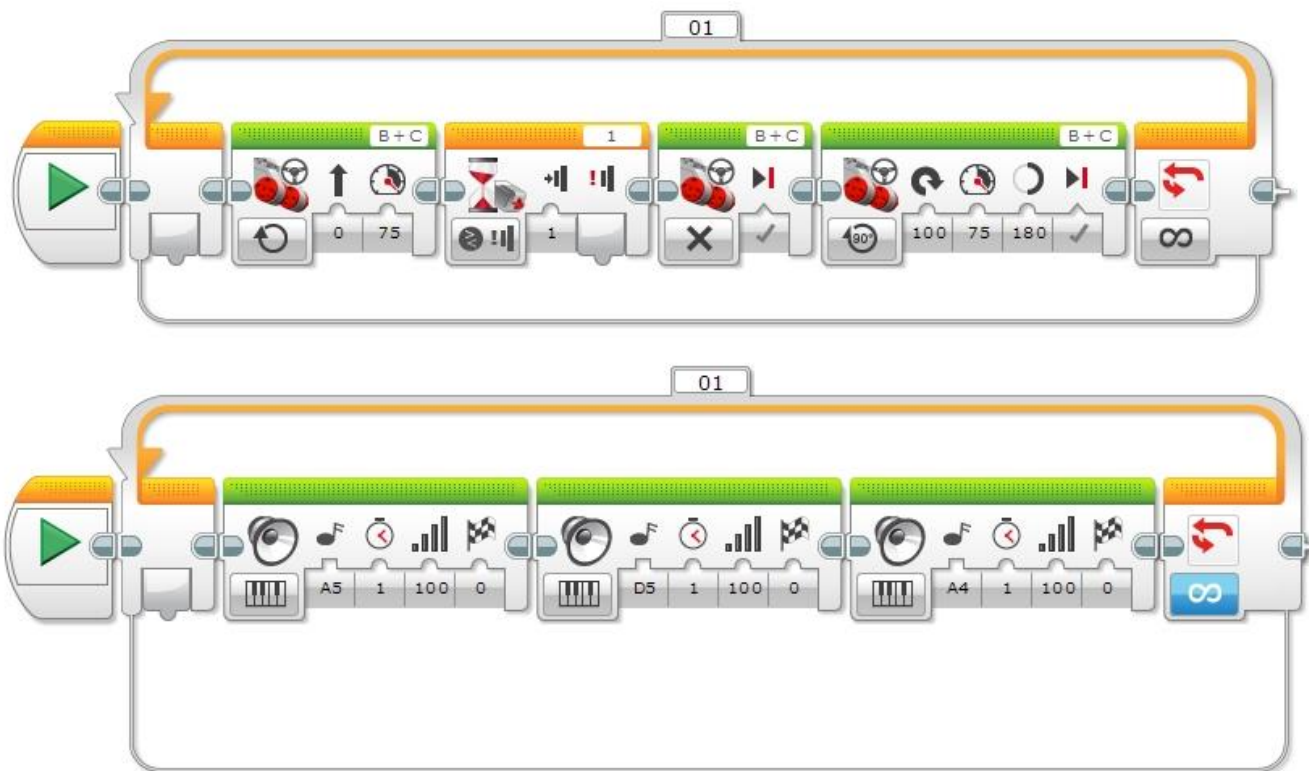


CVIČENÍ 08

- Robot se snaží udržet na bílém papíře,
 - jakmile uvidí černou čáru, „lekne se“ (zvuk...?),
 - couvne, pootočí se
 - a vyrazí jiným směrem.
 - Robot se vypne stiskem tlačítka.
-
- Pozor, před spuštěním musíte provést kalibraci světelného senzoru = zjistit, co je pro váš světelný senzor a vaše podmínky bílá a co černá!

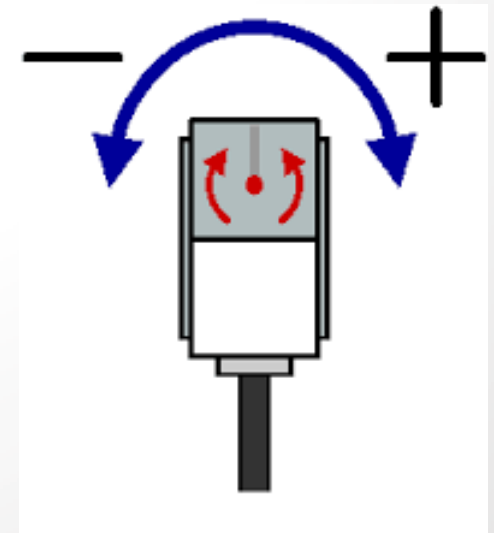
PARALELISMUS – CVIČENÍ 9

- Více úloh může běžet zároveň
 - v jednom vláknu například ovládání motorů
 - a ve druhém hraje nějakou melodii...





- jednoosý gyroskop
 - míra otáčení ve stupních za sekundu ve směru šipek (**Measure Rate**)
 - také hodnoty nasčítává, takže je možno použít pro detekci úhlu natočení (**Measure Angle**)
 - kostičku může být nutno do prostředí doinstalovat (Tools/Block Import, Gyro.ev3b)





- možno využít pro „přesné zatačení“ o zadaný úhel nebo pro jízdu rovně



1. Reset hodnoty gyra.

2. Otáče se...


3. tak dlouho, až bude nasčítaná hodnota (úhel) aspoň 45° .


PODPROGRAMY – MY BLOCKS

- Z kostiček pro „přesné“ otáčení můžeme udělat kostičku novou (podprogram) a tu pak dále využívat (volat).
- Tools/My Block Builder
- Vlastní kostičku můžeme pojmenovat, nastavit ikonu a nastavit jaké bude mít parametry → **jde o skutečné podprogramy včetně parametrů!**

PODPROGRAMY – MY BLOCKS

My Block Builder [Close]



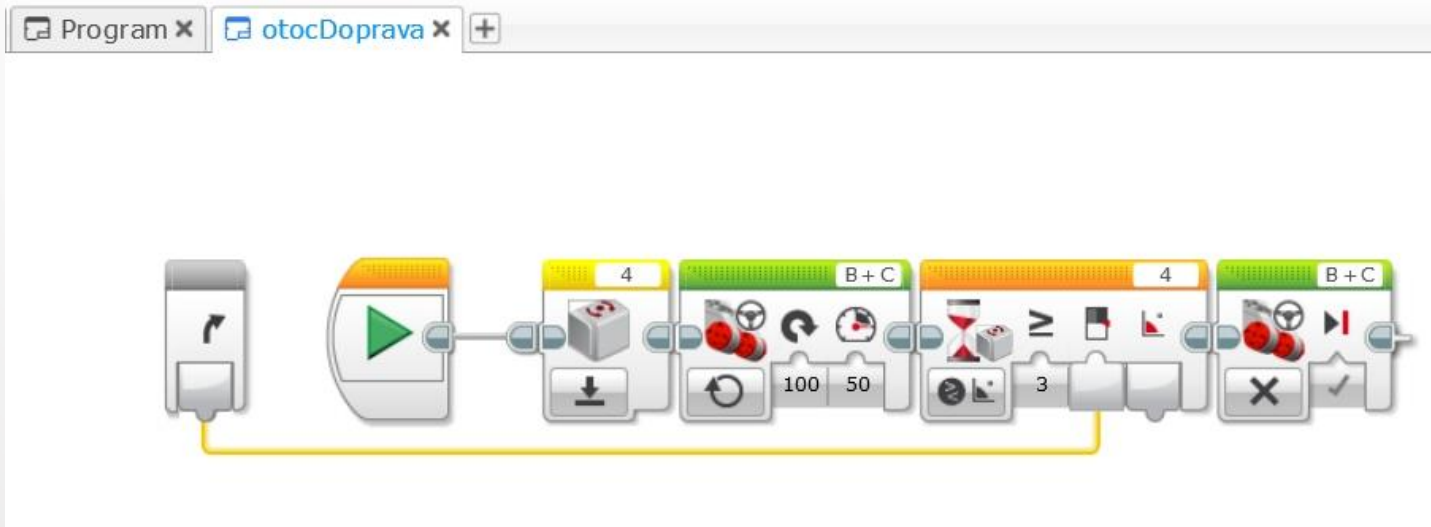
Click the button to add or edit parameters. 

Name: Description:

My Block Icons | **Parameter Setup** | **Parameter Icons**

Finish Cancel

PODPROGRAMY – MY BLOCKS

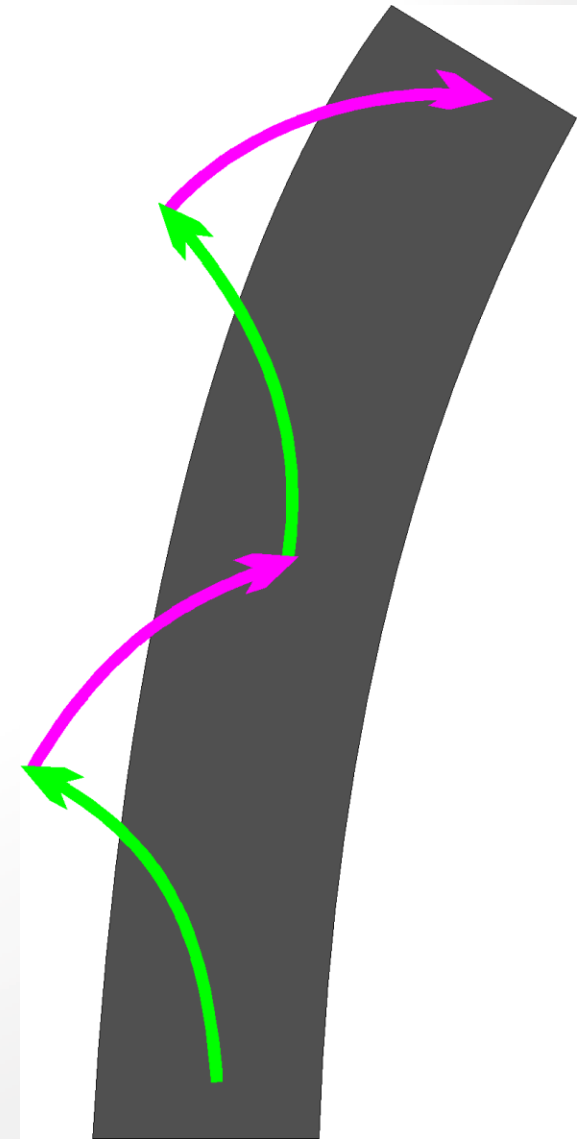


SLEDOVÁNÍ ČÁRY (LINE FOLLOWER)

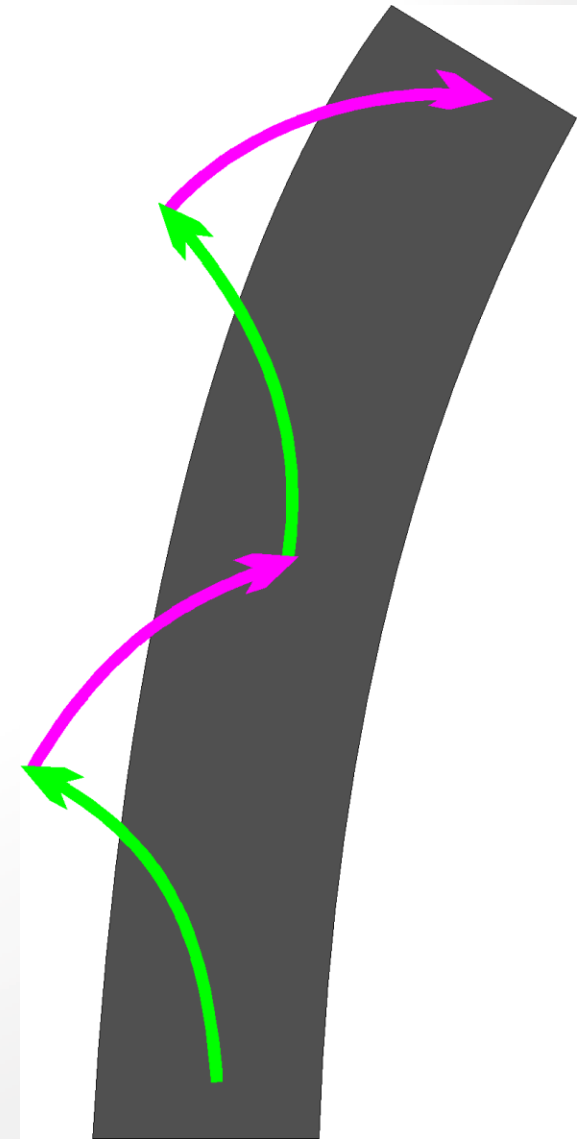
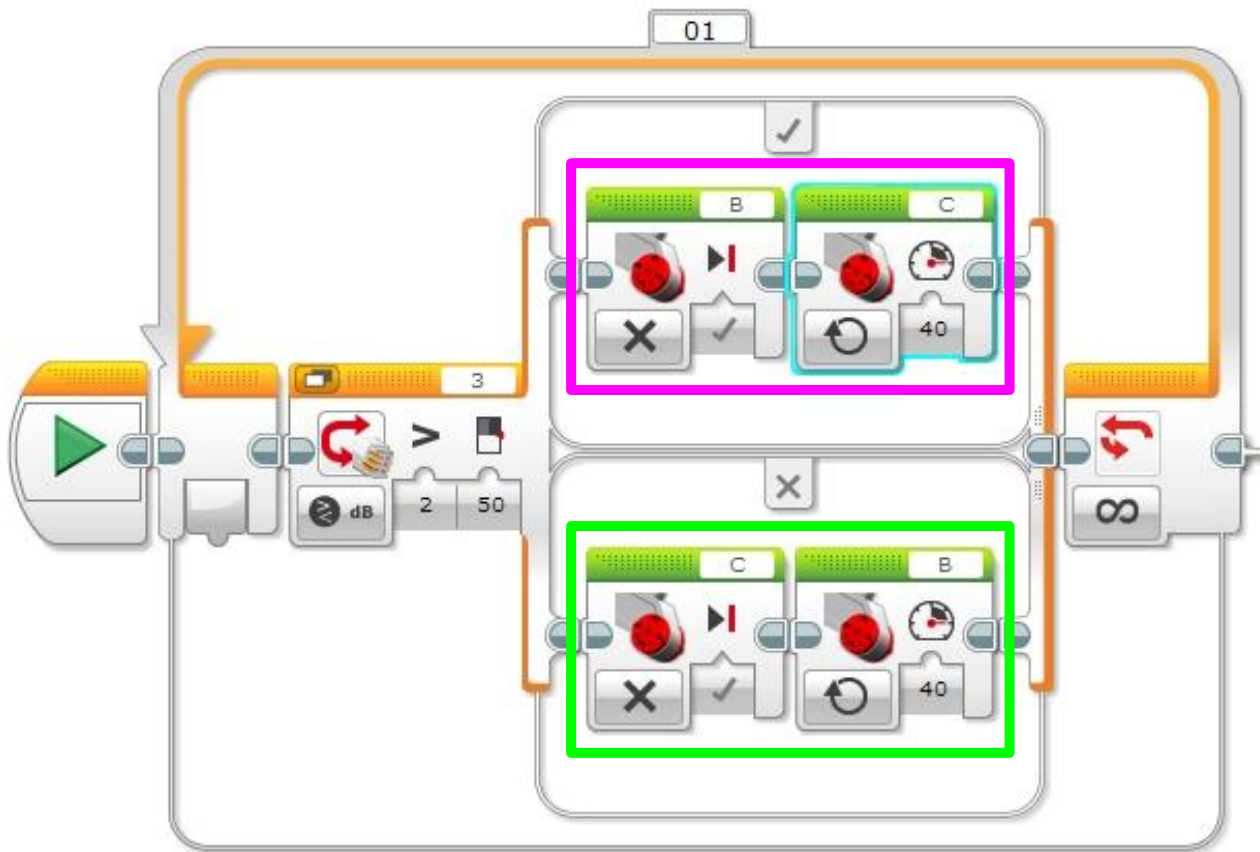
- Autonomní robot má projet co nejrychleji celou dráhu tak, že bude sledovat černou čáru.
- Závodí se na čas, resp. ve dvojicích každý s každým.
- Pokud robot opustí černou čáru, rozhoduje ujetá vzdálenost od počátku.
 - umístí se až za všemi, kteří dojeli do cíle
 - může čáru znovu najít a navázat (pokud to zvládne)

SLEDOVÁNÍ ČÁRY – CIK-CAK ALGORITMUS

- **cik-cak algoritmus (zig-zag)**
 - **zatačej doleva, dokud vidíš černou (čáru)**
 - až uvidíš bílou (tj. opustíš čáru), zastav
 - **zatačej doprava, dokud vidíš bílou (tj. dokud jsi mimo čáru)**
 - až uvidíš černou (čáru), zastav
 - opakuj od začátku



CIK-CAK ALGORITMUS



CIK-CAK ALGORITMUS

- jednoduchý
- stačí jeden světelný senzor
- robot stále zatačí
 - neefektivní
 - náročné na motory
 - náročné na zdroje
- nemá rád ostré zatačky

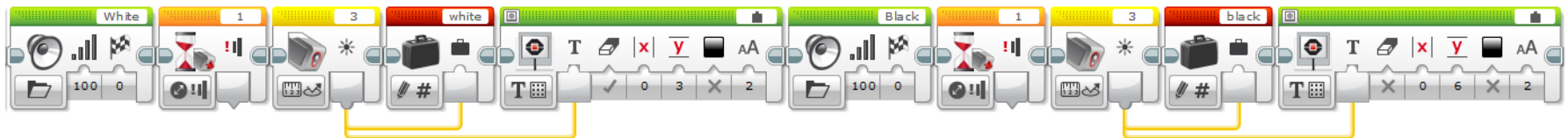
PROBLÉMY A NÁPADY

- světelný senzor nesmí být příliš blízko země
 - potom nefunguje odraz světla od diody
 - reaguje i na zvlněný papír
 - ideální vzdálenost 0,5 až 1,5 cm
- kolečka na papíře prokluzují
 - vybrat jiná?
 - zpomalit?
 - pásy?

NÁZNAK PID LINE FOLLOWERU

- Kalibrace hodnot černé a bílé barvy

```
function CALIBRATE  
  
  print "WHITE?"  
  Wait for Touch Sensor to change  
  white = Read Light Sensor  
  
  print "BLACK?"  
  Wait for Touch Sensor to change  
  black = Read Light Sensor  
  
done function
```



P REGULÁTOR

```
program LINE FOLLOWING
```

```
white = 0, black = 0  
CALIBRATE()
```

```
midpoint = ( white - black ) / 2 + black  
kp=1;
```

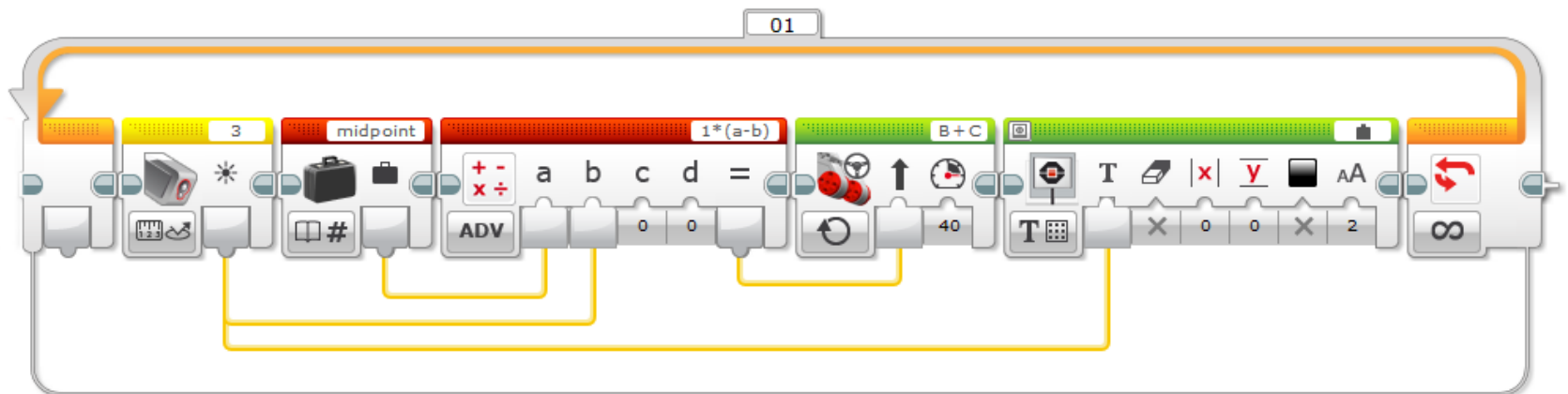
```
repeat
```

```
value = Read Light Sensor  
correction = kp * ( midpoint - value )
```

```
Turn B+C Motors by correction
```

```
done repeat
```

```
done program
```



ZDROJE

- [1] *LEGO® MINDSTORMS® - LEGO.com - Mindstorms LEGO.com* [online]. [cit. 2017-02-05]. Dostupné z: <https://www.lego.com/en-us/mindstorms>
- [2] *LEGO™ Gear Ratio Calculator* [online]. [cit. 2017-02-05]. Dostupné z: <http://gears.sariel.pl/>
- [3] *Customizable Line Following Tracks that you can print yourself - Robotsquare* [online]. [cit. 2017-02-05]. Dostupné z: <http://robotsquare.com/2012/11/28/line-following/>

Pokud není uvedeno jinak, všechny obrázky jsou vlastní tvorbou autorů.