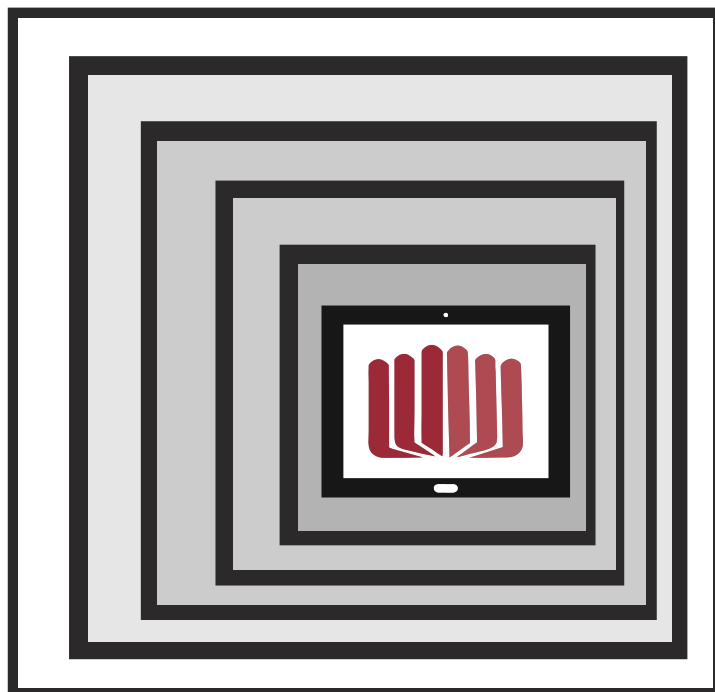
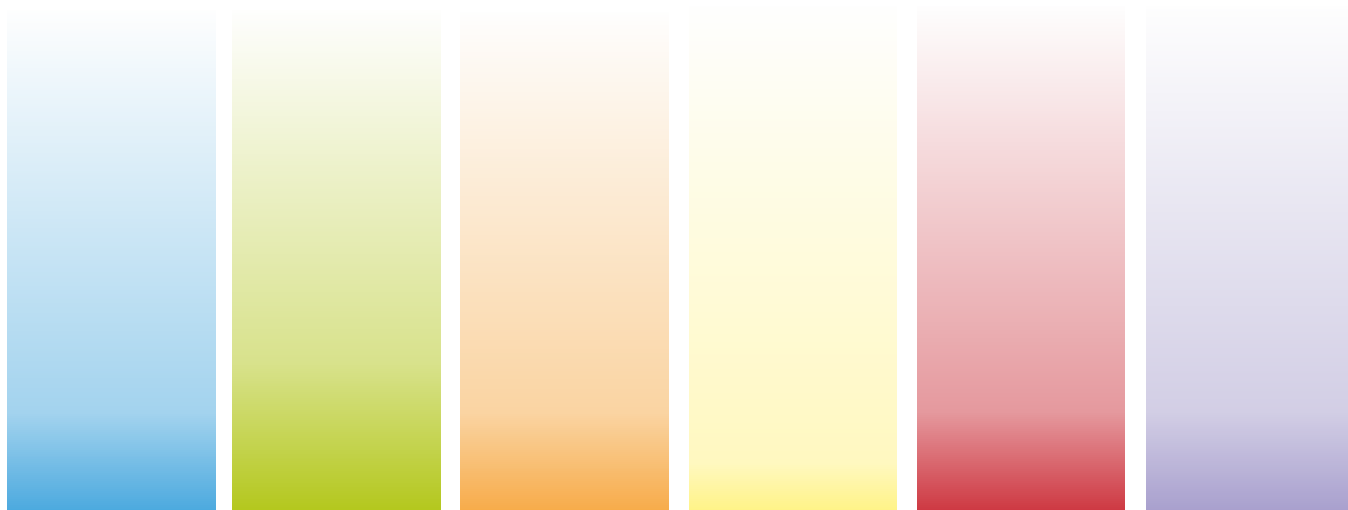


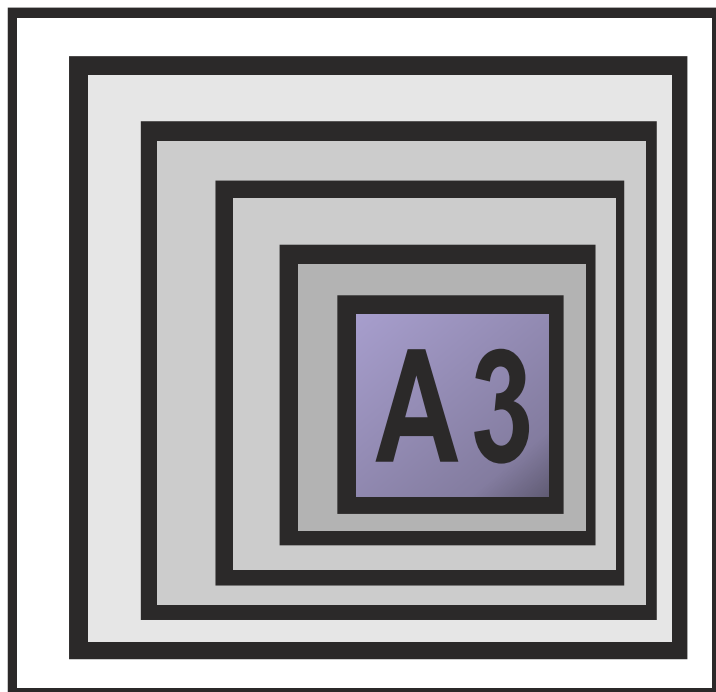
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ



Kantor Ideál

ROBOTICKÉ STAVEBNICE





ROBOTICKÉ STAVEBNICE

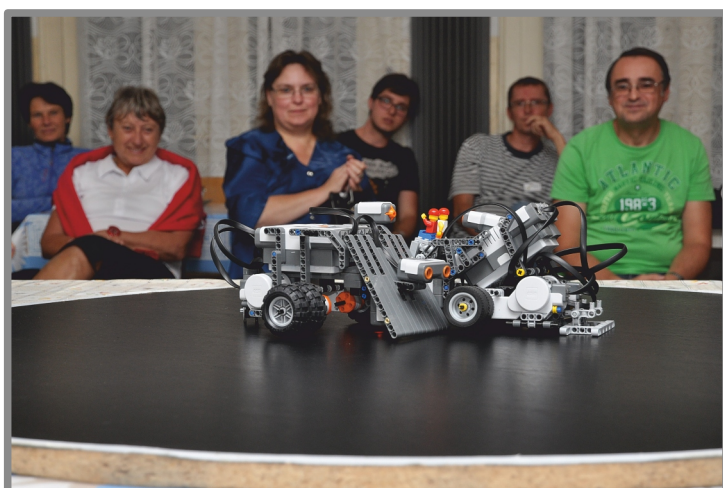
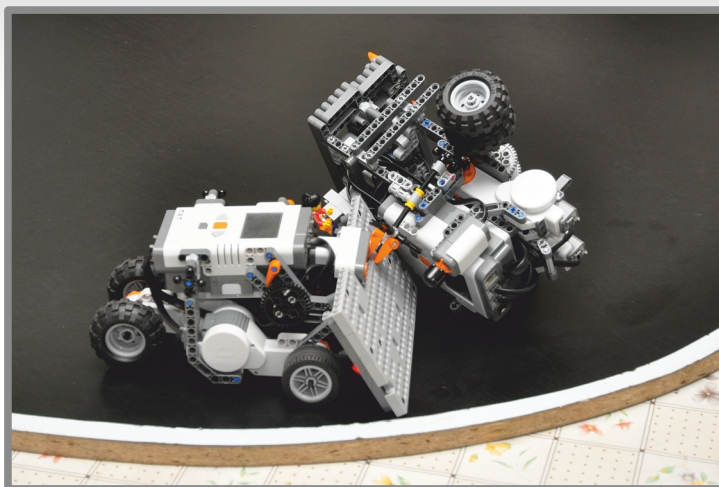


OBSAH

Robotické stavebnice a jejich využití ve výuce.	3
Lego Mindstorms a Merkur	4
Lego Mindstorms.	5
Kde soutěžit?	6
Robosoutěž ČVUT	6
Robotický den MFF UK	6
First Lego League.	7
Robotiáda	7
RoboRave	7
Motorola Diversity	8
RobotChallenge	8
Jde to i bez soutěží	8
Jak stavět?	9
Spojování kostiček	9
NXT kostka – základ stavebnice	10
Motor (servo motor).	10
Dotykový senzor (touch)	10
Ultrazvukový senzor (ultrasonic)	11
Světelný senzor (light)	11
Zvukový senzor (sound)	11
Další senzory	11
Jak programovat?	12
Grafické programování v EV3	12
Hlavní okno EV3	12
Ovládání kostky.	12
Výpis na displej (Display)	13
Čekání (Wait)	13
Zvuk (Sound)	13
Motor (Large Motor)	13
Pohyb (Move Steering)	14
Port View.	14
Cyklus (Loop)	14
Dotykový senzor (Touch).	14
Ultrazvukový senzor (Ultrasonic)	15
Světelný senzor (Light)	15
Paralelismus.	16
Sledování čáry (Line Follower).	16
Programování v NXC.	17
První program.	17
Motory = pohyb	17
Zastavení motorů	17
Wait.	17
Motory = zatačení	17
Motory – další příkazy	18
Senzory	18
Čekání na hodnotu senzoru	18
Řídící struktury – podmínky	18
Řídící struktury – cykly	19
Paralelní úlohy	19
Cvičení 09 – cik-cak jízda po čáře.	19
Použité zdroje	20

Robotické stavebnice a jejich využití ve výuce

- Kde všude se dá stavebnice využít
 - předmět informatika (RVP/ŠVP oblast algoritmizace)
 - robotický kroužek
 - volitelný předmět
 - studentské projekty
 - projektové dny/týdny



- Proč soutěžit?
 - Ukázat, že něco umím
 - Porovnat své schopnosti s ostatními
 - Zjistit, jak to dělají „jinde“
 - Naučit se prezentovat
 - Týmová spolupráce
 - Motivace pro další práci
 - Zábava, hra, ...

- S čím soutěžit?
 - stavebnice **Lego Mindstorms** (NXT nebo EV3)
 - stavebnice **Merkur**



obrázek převzat z [3]



obrázek převzat z [4]

Lego Mindstorms a Merkur

■ Lego Mindstorms

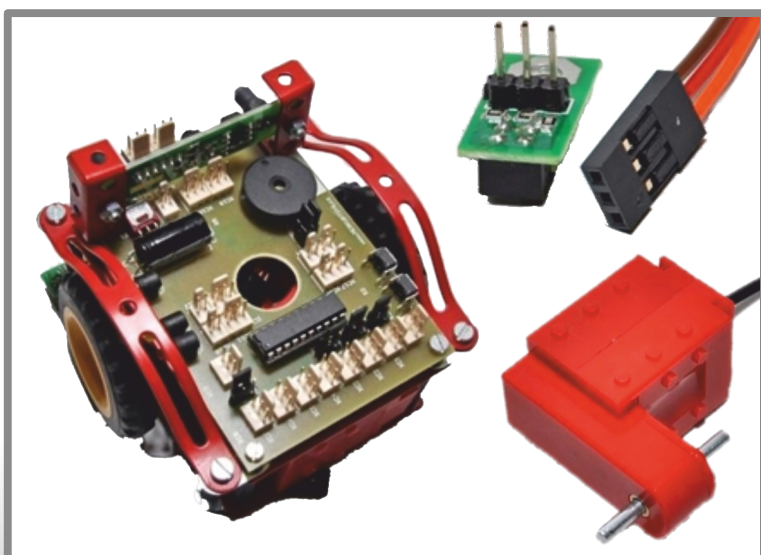
- Řídící kostka
- Motory
- Senzory (světelný, zvukový, dotykový, ultrazvukový, barevný, gyroskopický, GPS, ...)
- Různé možnosti programování



- ☺ snadná konstrukce, vhodné i pro začátečníky
- ☺ jednoduché programování – ikony, C, Java, Python
- ☺ velké množství senzorů
- ☺ množství návodů, metodik
- ☺ blbuvzdorné provedení
- ☹ omezený počet vstupů a výstupů
- ☹ originální senzory jsou drahé
- ☹ menší mechanická odolnost

■ Merkur

- Osazená řídící deska s procesorem
- Základem je procesor Atmel nebo PICAXE, resp. populární platforma Arduino
- Motory
- Světelná čidla, další průmyslově používaná čidla



- ☺ česká firma, má zájem dostat se do škol
- ☹ spíše pro zkušenější – hůře se programuje
- ☺ možno použít standardní průmyslové senzory
- ☹ méně návodů, metodik, ukázek
- ☹ není tolik blbuvzdorná
- ☺ možno libovolně rozšiřovat
- ☺ kvalitnější mechanické provedení

Lego Mindstorms

■ Historie

- 1988: počátky na MIT – první programovatelná kostka
- 1998: RCX
- 2006: NXT
- 2009: NXT 2.0
- 2013: EV2

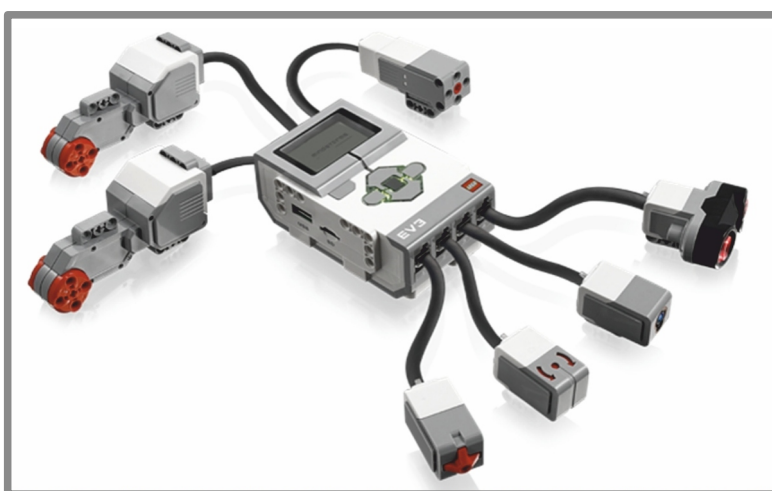
■ verze NXT



obrázek převzat z [3]

- 3 stejné motory
- 4 senzory (světelný, dotykový, ultrazvukový, zvukový)
- na řídicí kostce je Bluetooth modul + USB
- 32bit 48MHz procesor + 64kB paměti RAM
- 256kB flash paměti
- displej 100 x 64 pixelů

■ verze EV3



obrázek převzat z [3]

- 3 motory (2 stejné, jeden rychlejší)
- 4 senzory (světelný, dotykový, ultrazvukový, akcelerometr), lze využít senzory z NXT
- na řídicí kostce je Bluetooth modul + USB
- 300MHz procesor + 64MB paměti RAM
- 16MB flash paměti + volitelně SD karta
- displej 178 x 128 pixelů

Kde soutěžit?

Robosoutěž ČVUT

- Pořádá FEL ČVUT Praha, podzim (říjen – listopad), pro SŠ
- Každý rok jiný úkol → různorodé
 - balancování s míčkem
 - překonání řeky
 - sbírání míčků
 - prohledávání bludiště
- Pouze roboti ze stavebnice Lego Mindstorms
 - zapůjčení stavebnic ☺
- Středoškolského kola se v roce 2014 zúčastnilo cca 80 týmů z celé republiky
 - atraktivní ceny
 - nadstandardní občerstvení ☺
- Vítězové SŠ kola postupují do VŠ finále, kde změří své síly proti vysokoškolákům
 - vítězové SŠ kol jsou přijati na ČVUT ☺
- <http://www.robosoutez.cz>



Robotický den MFF UK

- Pořádá MFF UK Praha, červen, bez rozdílu věku
- Různé soutěžní disciplíny
 - Bear Rescue (záchrana medvěda z bludiště)
 - Ketchup House (sbírání plechovek)
 - Line Follower
 - Mini Sumo
 - Art, Robots & Entertainment (otevřená soutěž – umění/robotika/zábava)
 - RoboCarts (závody)
- Rok 2014 – 152 robotů a 235 účastníků z 5 zemí
- Kostrukce robota není omezena
 - ale je vždy speciální podkategorie stavebnice
- <http://www.roboticday.cz>



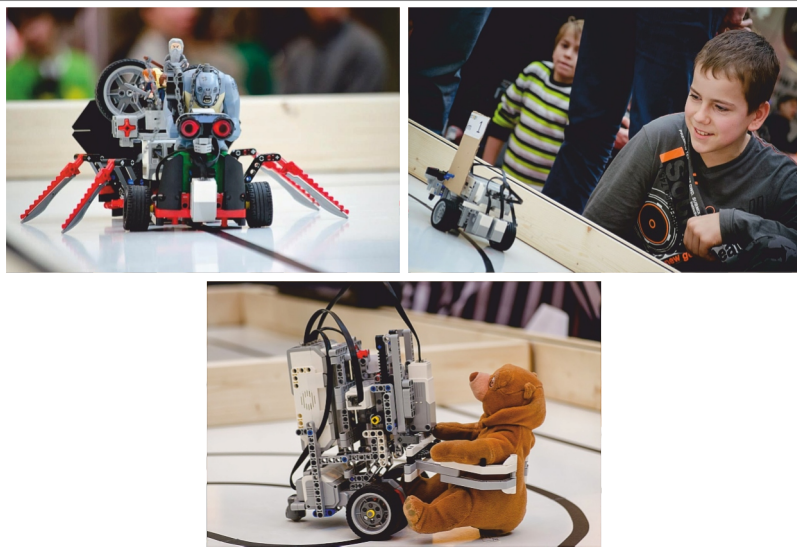
First Lego League

- Mezinárodní robotická soutěž, týmy ve věku 10–16 let, 3 až 10 členů
- Jednotné téma, každý rok jiné
 - Nature's Fury
 - Senior Solutions
- 4 soutěžní disciplíny:
 - Robot Game
 - Výzkumný úkol
 - Design robota
 - Týmová spolupráce
- <http://www.firstlegoleague.org>
<http://www.ceskaligarobotiky.cz>



Robotiáda

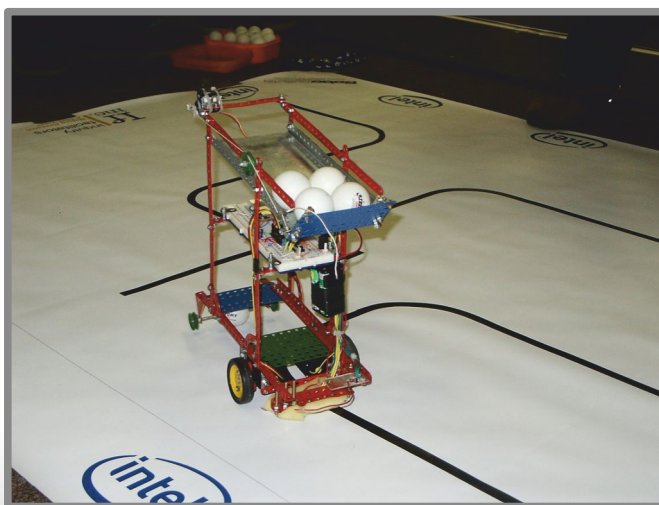
- Brno, prosinec a březen, pro 2. stupeň ZŠ
- Roboti ze stavebnice Lego Mindstorms
- Kategorie
 - sledování čáry
 - záchrana medvěda (dálkově řízená)
 - robopřehlídka (+ prostor pro prezentaci týmu, školy...)
- <http://www.robotiada.cz>



obrázky převzaty z [9]

RoboRave

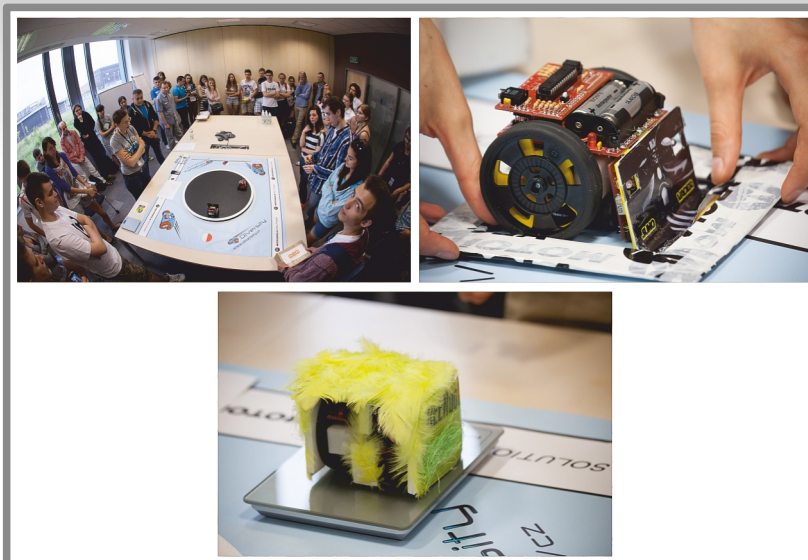
- Mezinárodní soutěž, 9–18 let, finále v květnu v Albuquerque, New Mexico
- národní kolo v březnu v Pardubicích
- V ČR kategorie
 - hašení ohně
 - doprava míčků
- <http://amavet.fottp.cz>
<http://roboquerque.org>



obrázek převzat z [10]

Motorola Diversity

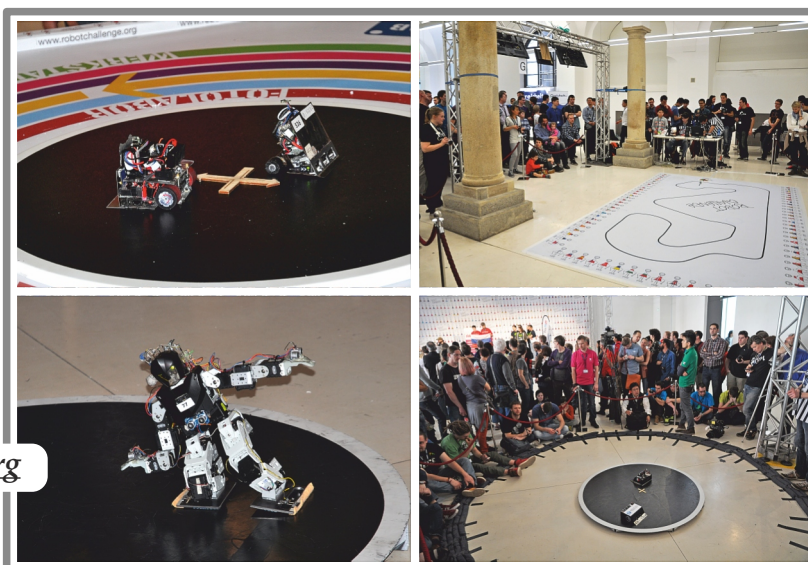
- Pro smíšené týmy (min 50% dívek)
- Soutěž mini-sumo robotů
 - předchází jí návrh algoritmu a jeho testování ve virtuálním prostředí, poté vybrané týmy dostanou soutěžní HW
 - součástí je video prezentace týmu a stavby robota
- <http://www.diversity.pl/en>



obrázky převzaty z [11]

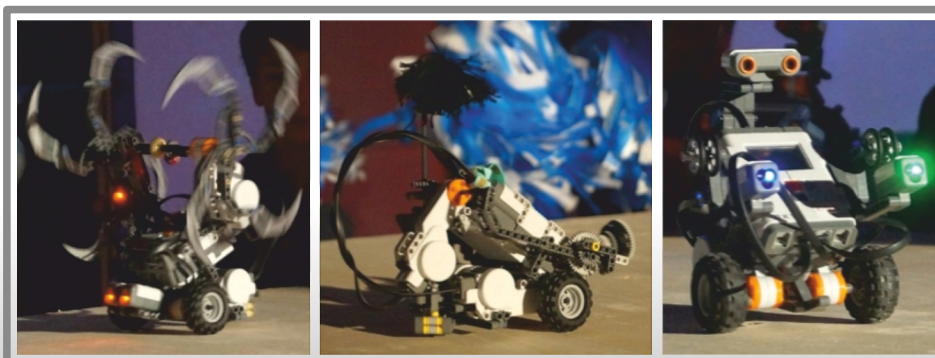
RobotChallenge

- Největší evropská soutěž robotů
 - v roce 574 robotů ze 46 zemí (Kanada, Katar, Taiwan, Mexiko, ...)
- koncem března ve Vídni
- 15 různých kategorií
- <http://www.robotchallenge.org>



Jde to i bez soutěží

- projektové dny
 - spojení napříč předměty (výtvarná výchova, divadlo...)
- přehlídka robotů

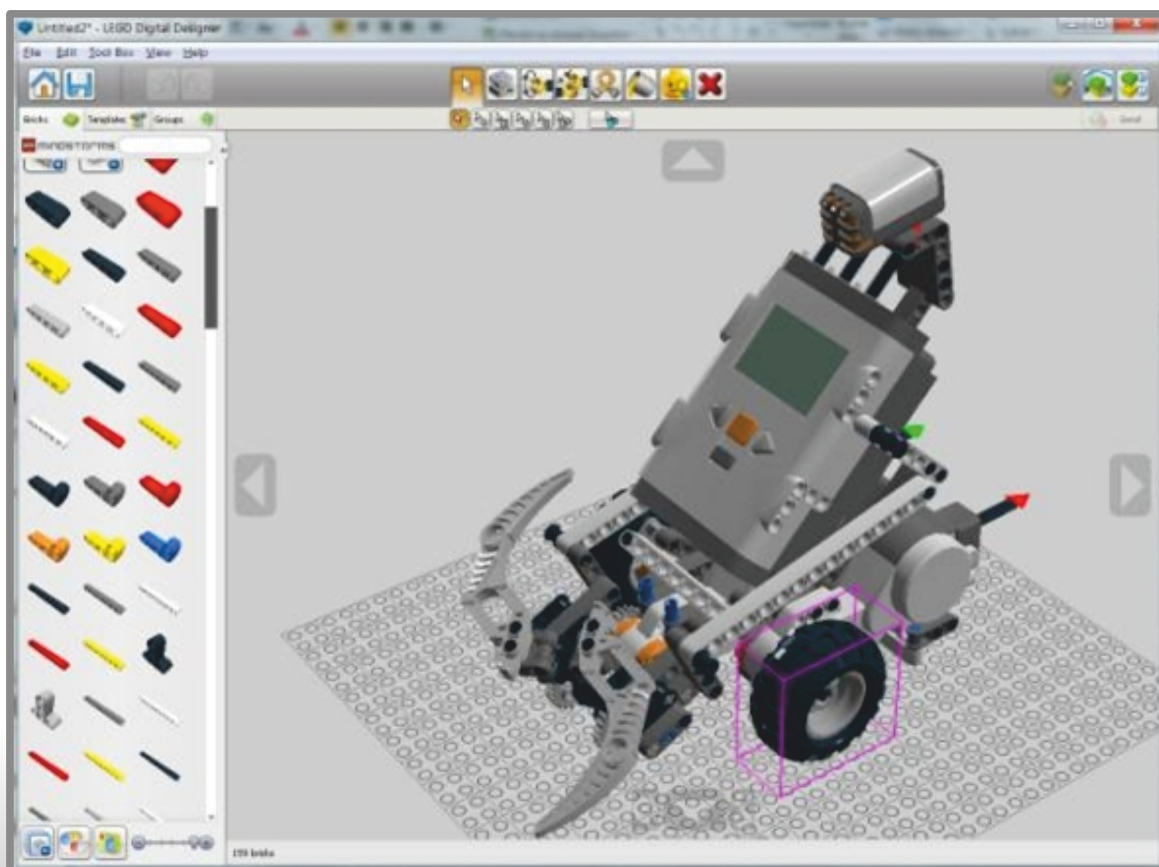
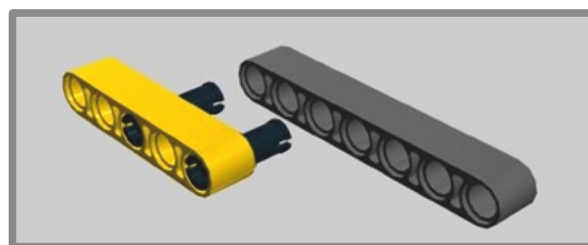
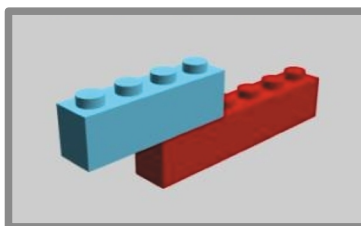


Jak stavět?

- Verze stavebnic NXT a EV3 se po HW stránce liší
 - verze EV3 může používat motory i senzory z NXT
 - naopak je to problém (motory z EV3 fungují s NXT, senzory ne)
 - NXT kostku je možno programovat softwarem pro EV3
- Dále bude vše uváděno pro verzi NXT.

Spojování kostiček

- kostičky jako v LEGO
 - menší pevnost,
lze využít díly z jiných stavebnic
- spojování pomocí čepů
 - pevnější
 - modré, černé, červené → pevné spojení
 - bílé → pohyblivé spojení
- Stavět lze i virtuálně pomocí **Lego Digital Designer**
(<http://ldd.lego.com>)



NXT kostka – základ stavebnice

- řídicí jednotka
- připojení k PC přes USB nebo Bluetooth
- dostupný i WiFi modul, lze ovládat přes chytrý telefon
- Kostka je schopna vyhodnocovat údaje, které jsou dodávány na vstupní porty různými senzory a podle nich řídit akce.
- do jisté míry otevřené řešení – lze nahrát jiný firmware a tím změnit funkčnost (leJOS, nxtOSEK, ...)
- programovat lze v
 - NXT-G (grafické prostředí, komerční)
 - NXT-G EV3 (grafické prostředí, Home verze volně)
 - RobotC (C, komerční)
 - NXC (Not Exactly C, OpenSource)
 - NXJ (Java, OpenSource)
 - ...
- Defaultní zapojení senzorů a motorů
- 4 vstupní porty (1, 2, 3, 4)
 - 1 → dotykový (touch) senzor
 - 2 → zvukový (sound) senzor
 - 3 → světelný (light), barevný senzor
 - 4 → ultrazvukový (ultrasonic) senzor
- 3 výstupní porty (A, B, C)
 - B, C → motory pro pohyb
 - A → motor pro speciální funkce, žárovka
- propojování pomocí kabelů s konektory RJ12 (telefonní konektor)



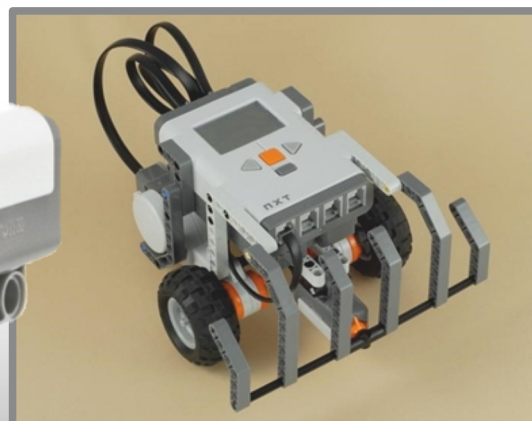
Motor (servo motor)

- volitelné úrovně výkonu (rychlosti otáčení)
 - snižuje se výkon
- vestavěný rotační senzor → přesnost $\pm 1^\circ$
 - podobný jako u kolečka myši
 - vhodný pro přesné řízení (ruka robota...)
- ochrana motoru proti přetížení



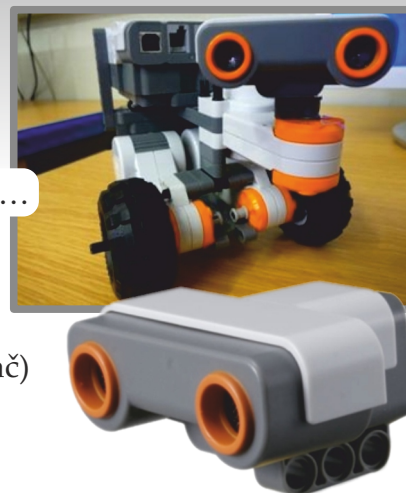
Dotykový senzor (touch)

- detekce stisku/uvolnění
- použití
 - tlačítko pro zapnutí/vypnutí
 - nárazník
 - nastavování hodnot opakovaným stiskem



Ultrazvukový senzor (ultrasonic)

- měří cca 0 – 255 cm s přesností ± 3 cm
- vyžaduje kvalitní odraz
 - princip podobný orientaci netopýra v prostoru, ozvěna...
 - ideální pro detekci velkých předmětů, pěnové materiály a zaoblené předměty působí problémy
- použití
 - oči robota (vyhýbání se překážkám → robotický vysavač)
 - měření vzdálenosti
 - hledání protivníka → robo sumo



Světelný senzor (light)

- rozlišuje světlo/tmu resp. měří intenzitu světla (LED vypnuta)
- měří intenzitu odraženého světla v procentech (LED zapnuta)
- použití
 - sledování čáry na kontrastním podkladě (černá lepicí páska...)
 - udržení robota v kruhu (ohraňováno čarou)



Zvukový senzor (sound)

- mikrofón
- měří intenzitu zvuku v dB nebo procentuální vyjádření hlasitosti (dBA)
 - 4-5 % → ticho
 - 5-10 % → vzdálený hovor
 - 10-30 % → hovor v blízkosti senzoru
 - 30-100 % → hlasitá hudba



Další senzory

- gyroskopický senzor
 - jednoosý gyroskopický senzor, hodnota reprezentující rotaci vyjádřenou v počtu stupňů za sekundu
 - použití → udržení míčku na plošině
- barevný světelný senzor
 - nastaven tak, aby rozpoznával hlavně barvy použité u dílků stavebnic, vrací čísla barev 0 – 17
 - použití
 - třídění lentilek
 - skládání Rubikovy kostky

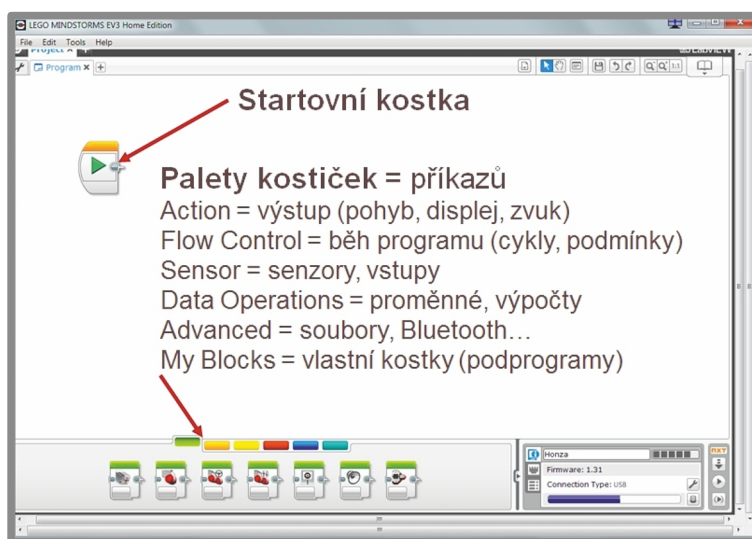
Jak programovat?

- V grafickém prostředí – pro začátečníky
 - NXT-G verze 2 → pro verzi NXT, komerční
 - EV3 Home Edition → pro NXT i EV3, volné
- „Klasický“ programovacím jazykem
 - zápis příkazů v textové podobě
 - Bricx Command Center, Open Source → NXC
 - Robot C, komerční → C
 - Java + vývojové prostředí Eclipse, Netbeans

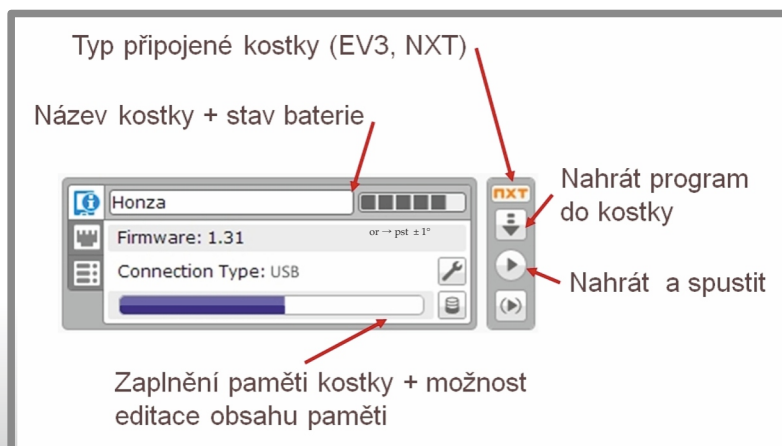
Grafické programování v EV3

- Založeno na „průmyslovém prostředí“ LabView
- Použitelné pro EV3 i NXT
 - v případě připojení kostky NXT nebudou některé volby dostupné
 - funkcionality kostky NXT se nerozšíří!
 - některé věci bohužel nefungují, ale dá se to vyřešit

Hlavní okno EV3



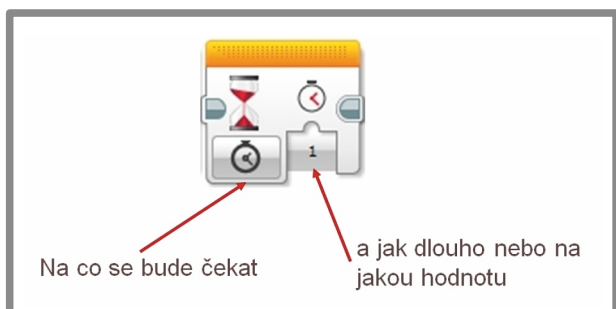
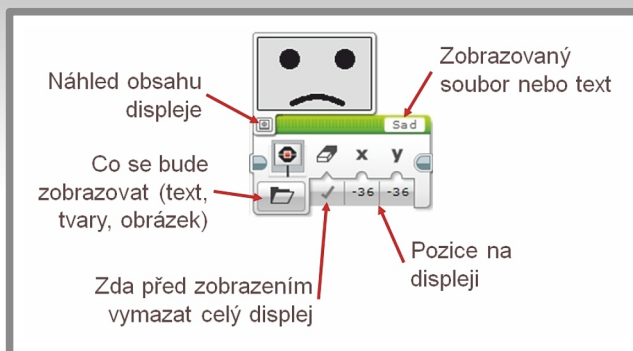
Ovládání kostky



Výpis na displej (Display)



- Obrázky připraveny pro EV3
– mají pro NXT příliš velké rozlišení
 - nutno upravit jejich pozici na displeji
- Po zobrazení je nutno počkat, jinak program skončí a nic neuvidíme

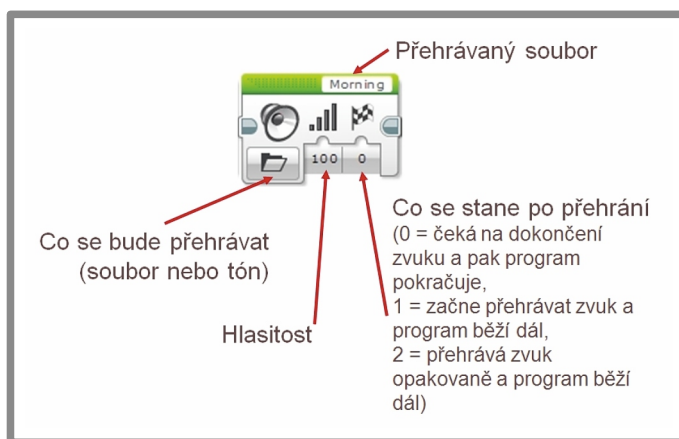


Čekání (Wait)



- Čekat lze na různé hodnoty senzorů nebo po zadaný čas

Zvuk (Sound)

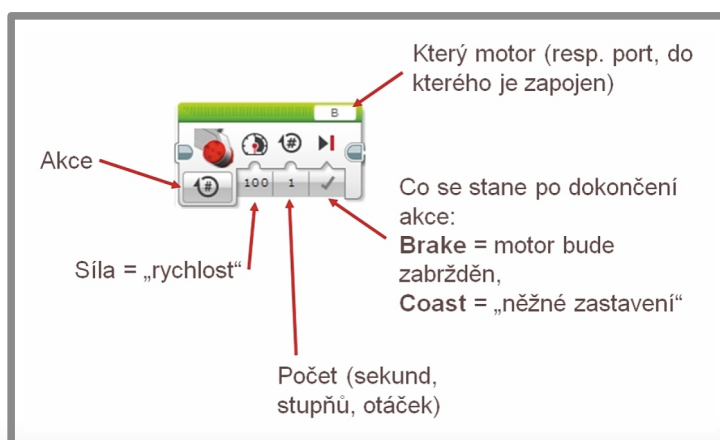


Motor (Large Motor)



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
- **On for Rotations** = motor se otočí o zadaný počet otáček



Pohyb (Move Steering)



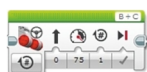
- Kostička ovládá dva motory a umožňuje jejich současný pohyb a zatáčení změnou poměru jejich otáček
 - intuitivnější ovládání

Akce



Do kterých portů jsou motory zapojeny

Zatáčení = poměr mezi otáčkami obou motorů



Jízda rovně



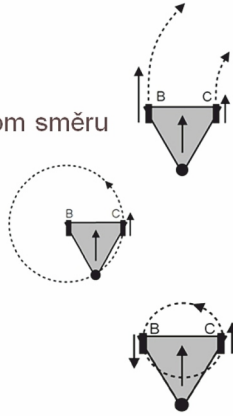
Pozvolné zatáčení v jednom směru



Zatáčení jedním kolem



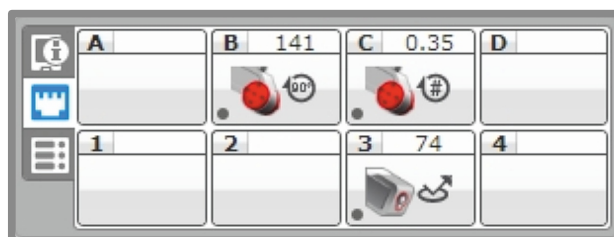
Zatočení na místě



- Podobnou funkci má i kostička **Move Tank**, kde lze přímo nastavit síly pro jednotlivé motory.
 - toto ovládání může být někdy přesnější

Port View

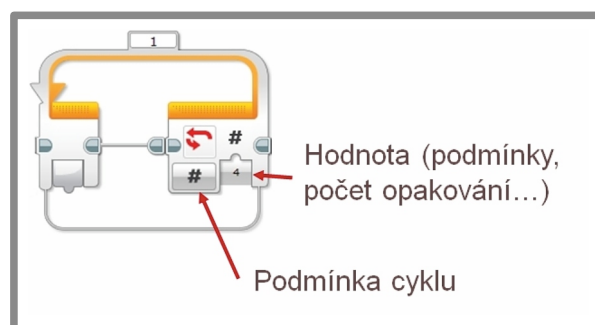
- Na kartě Port View můžeme velice snadno odečítat hodnoty ze vstupů i výstupů.
 - Například úhel otočení motoru nebo úroveň světla snímaného světelným senzorem.
 - Vhodné pro ladění.



Cyklus (Loop)



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



Dotykový senzor (Touch)

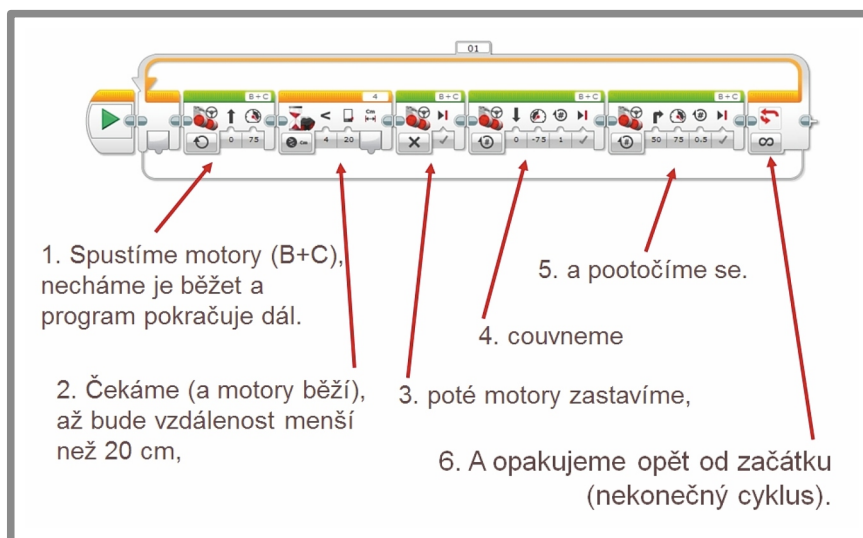


- „tlačítko“
- Lze použít v cyklu, čekání a podmínce
- Lze testovat
 - uvolnění (Released)
 - stisknutí (Pressed)
 - stisknutí a uvolnění (Bumped)

Ultrazvukový senzor (Ultrasonic)

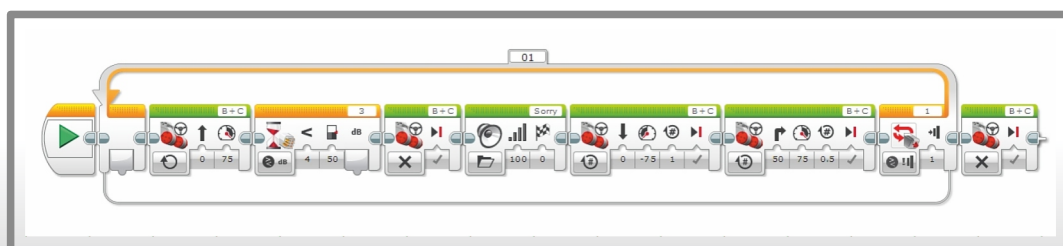


- Bohužel senzor z NXT není přímo v EV3 Home Edition podporován ☹ → musíme si kostičku pro něj doinstalovat ručně
 - *Tools/Block import*, soubor **Ultrasonic.ev3b**
 - Poté se příslušná kostička objeví mezi senzory a lze ji používat v podmínkách, čekání a cyklech.
- Příklad použití:
Robot jede dopředu, jakmile se přiblíží k překážce, couvne, pootočí se a zase pojede dopředu.



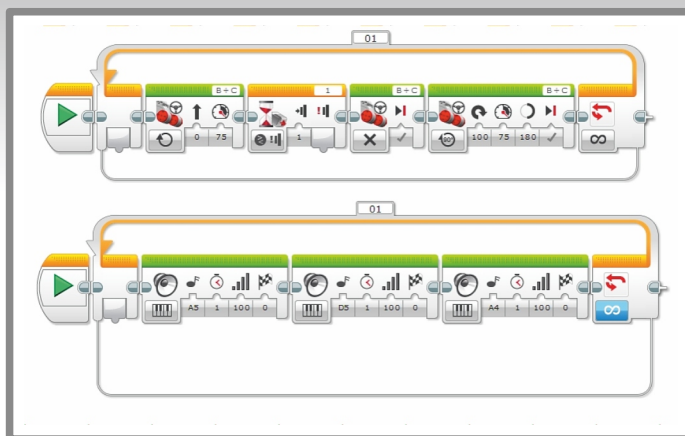
Světelný senzor (Light)

- Problém! Software EV3 nemá podporu pro Světelný senzor z NXT.
 - Pouze pro Color Sensor z EV3 ☹
- Musíme to obejít tak, že importujeme další kostičku pro Zvukový senzor z NXT a ten můžeme použít místo senzoru světelného.
 - *Tools/Block import*, soubor **Sound.ev3b**
 - Poté se příslušná kostička objeví mezi senzory a lze ji používat v podmínkách, čekání a cyklech
- Pro práci se světelným senzorem tedy budeme používat kostičku **NXT Sound Sensor**.
 - Je to nešikovné, ale funkční.
 - Hodnota dB udává hodnotu odraženého světla (Reflected Light)
 - Hodnota dBa udává hodnotu okolního světla (Ambient Light)
- Příklad použití: 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.



Paralelismus

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



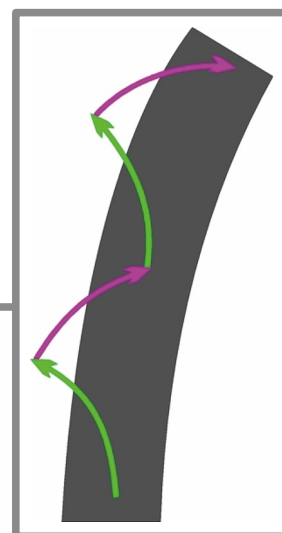
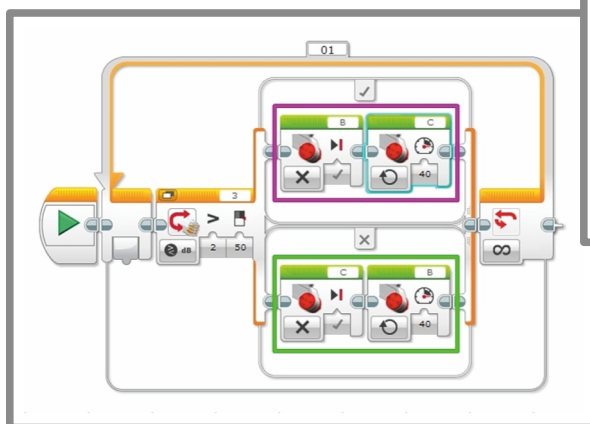
Sledování čáry (Line Follower)

- 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)

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

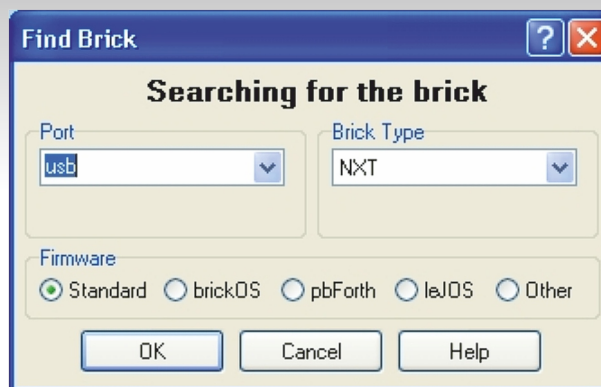
- jednoduchý
- stačí jeden světelný senzor
- robot stále zatáčí
 - neefektivní
 - náročné na motory
 - náročné na zdroje
- nemá rád ostré zatáčky

- 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?
 - dva senzory



Programování v NXC

- Bricx Command Center
 - Open Source, <http://bricxcc.sourceforge.net>
 - Jazyk NXC = Not eXactly C
- Připojení kostky



První program

- program se skládá z úloh (task), hlavní úloha **task main()** je spouštěna automaticky
- překlad programu (kompilace) – F5
- nahrání programu do kostky – F6
- nahrání a spuštění – Ctrl+F5

```
task main()
{
    TextOut(0, 0, "Ahoj!", 0);
    Wait(1000);
}
```

Motory = pohyb

- **OnFwd(OUT_BC, 100)** → příkaz pro otáčení motoru
 - OnFwd (port(y), síla motoru)
 - OUT_A(B,C), OUT_AB(AC,BC), OUT_ABC
 - síla motoru se udává v procentech maximálního výkonu, tj. -100 až 100
- **OnRev(OUT_B, 75)** → otáčení motoru na druhou stranu
 - OnRev(OUT_B, 75) = OnFwd(OUT_B, -75)

Zastavení motorů

- **Off(OUT_BC)** → zastavení motorů (s brzdou)
 - Off(OUT_BC) = OnFwd(OUT_BC, 0)
- **Float(OUT_BC)** → „něžné“ zastavení motorů (dojezd)

Wait

- **Wait(2000)** → pozastaví běh programu
 - Wait(počet milisekund)
 - nezastaví pohyb robota!, ale na tomto místě programu se čeká, až uplyne zadaný čas
- Příklad – dopředu a dozadu

```
task main()
{
    OnFwd(OUT_BC, 100);
    Wait(2000);
    OnRev(OUT_BC, 100);
    Wait(2000);
}
```

Motory = zatáčení

- **OnFwdSync(OUT_BC, 100, 50)** → síla rotace jednotlivých motorů
 - OnFwdSync(OUT_BC, 100, 50) = OnFwd(OUT_B, 100); OnFwd(OUT_C, 50)

Motory – další příkazy

- **RotateMotor(výstup, síla, úhel)**
- **RotateMotorEx(výstup, síla, úhel, zatačení, synchronizace, brzda)**
 - zatačení (-100, 100) → poměr otáček jednotlivých motorů, 0 = jízda rovně, 50 = jeden motor stojí, druhý se točí, 100 = motory se točí proti sobě
 - synchronizace (true, false) → synchronizovat otáčky motorů pro přesnější pohyb
 - brzda (true, false)

Senzory

- **SetSensorNázev(port)** → nastavení (inicializace) senzorů
 - IN_1(2, 3, 4)
- **Sensor(port)** → funkce, vrací hodnotu ze senzoru na daném portu
- **SENSOR_port** → proměnné s hodnotami ze senzorů
- Pozor – pro Ultrasonic je trošku jiné nastavení
 - SetSensorTouch(IN_1) Sensor(IN_1) SENSOR_1
 - SetSensorSound(IN_2) Sensor(IN_2) SENSOR_2
 - SetSensorLight(IN_3) Sensor(IN_3) SENSOR_3
 - **SetSensorLowspeed(IN_4) SensorUS(IN_4)**

Čekání na hodnotu senzoru

- **until(SENSOR_1 == 1)** → na tomto místě programu se bude čekat tak dlouho, dokud nebude podmínka splněna
 - Jakmile bude podmínka splněna, běh programu pokračuje dalším příkazem.
- Příklad – dotykový senzor

```
task main()
{
    SetSensorTouch(IN_1); //nastavení dotykového senzoru
    OnFwd(OUT_BC, 75);    //pohyb dokud není senzor stisknutý
    until(SENSOR_1 == 1);
    Off(OUT_BC);          //zastavení pohybu
}
```

Řídící struktury – podmínky

- porovnání: <, >, <=, >=, ==, !=
 - **if** (podmínka)
 {příkazy}
 - **if** (podmínka)
 {příkazy}
 - **else**
 {příkazy}

Řídící struktury – cykly

- **while** (podmínka)
{příkazy}
- **for** (inicializace, podmínka, inkrementace)
{příkazy}
- **repeat** (počet)
{příkazy}
- Příklad – jízda do čtverce

```
task main()

repeat(4)           // opakuj 4krát
{
    OnFwd(OUT_BC, 100); // jízda vpřed
    Wait(1000);
    Off(OUT_C);
    OnFwd(OUT_B, 75); // otočení,
    Wait(800); // hodnotu nutno vyzkoušet
}
```

Paralelní úlohy

- **Precedes** (úloha_1, úloha_2, ...) → paralelní běh několika úloh
- **start jméno_úlohy**
→ spustí danou úlohu
- **stop jméno_úlohy**
→ ukončí běh dané úlohy
- Příklad
– robot jede a bliká

```
task svetlo() // blikající žárovka na portu A
{
    repeat(5)
    {
        OnFwd(OUT_A, 75); // rozsvítím žárovku
        Wait(1000);
        Off(OUT_A); // zhasnu žárovku
        Wait(1000);
    }
}

task jizda() //jízda vpřed
{
    OnFwd(OUT_BC, 75);
    Wait(13000);
}

task main() //spuštění obou úloh paralelně
{
    Precedes(svetlo, jizda);
}
```

Cvičení 09 – cik-cak jízda po čáře

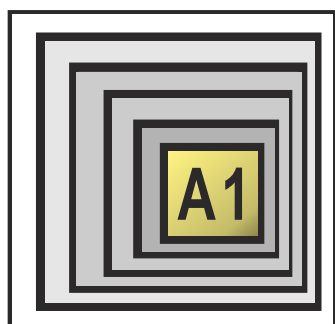
```
task main()
{
    SetSensorLight(IN_3, true); //nastavení světelného senzoru
                                // na port 3
                                // nekonečný cyklus
    while(true) {
        if (SENSOR_3 > 50) {
            // vidím bílou
            OnFwd(OUT_B, 0);
            OnFwd(OUT_C, 40);
        } else {
            // vidím černou
            OnFwd(OUT_B, 40);
            OnFwd(OUT_C, 0);
        }
    }
}
```

Použité zdroje

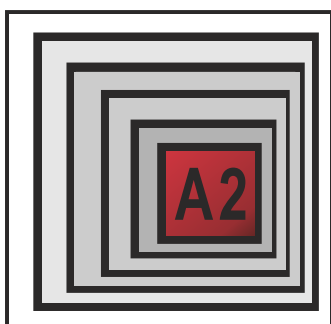
- [1] REFSDAL, Eirik. Golf bot. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-01-11]. Dostupné z: http://en.wikipedia.org/wiki/Lego_Mindstorms_NXT#mediaviewer/File:Lego_Mindstorms_Nxt-FLL.jpg. Licence CC BY 2.0.
- [2] Lego NXT in a common configuration. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-01-11]. Dostupné z: http://commons.wikimedia.org/wiki/File:Lego_NXT.jpg. Licence PD.
- [3] EDUXE S.R.O. – stránky společnosti. [online]. [cit. 2015-01-11]. Dostupné z: <http://www.eduxe.cz>
- [4] MERKUR TOYS s.r.o. – stránky společnosti. [online]. [cit. 2015-01-11]. Dostupné z: <http://www.merkurtoys.cz>
- [5] Lego Mindstorms RCX. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-01-11]. Dostupné z: <http://commons.wikimedia.org/wiki/File:LegoMindstormsRCX.jpg>. Licence CC BY-SA 3.0.
- [6] Mindstorms NXT. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-01-11]. Dostupné z: <http://commons.wikimedia.org/wiki/File:Nxt-brique.jpg>. Licence CC BY-SA 3.0.
- [7] Lego mindstorms ev3. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-01-11]. Dostupné z: <http://commons.wikimedia.org/wiki/File:Lego-mindstorms-ev3.jpg?uselang=cs>. Licence CC BY-SA 3.0.
- [8] ČVUT - KATEDRA ŘÍDICÍ TECHNIKY. Robosoutěž [online]. [cit. 2015-01-18]. Dostupné z: <http://www.robosoutez.cz/>
- [9] Robotiáda [online]. [cit. 2015-01-18]. Dostupné z: <http://www.robotiada.cz>
- [10] Gymnázium, Pardubice. [online]. [cit. 2015-01-18]. Dostupné z: <http://www.gypce.cz>
- [11] Motorola Diversity. [online]. [cit. 2015-01-18]. Dostupné z: <http://www.diversity.pl/en/>
- [12] RoboParade 2013 at Macomb Community College. [online]. [cit. 2015-01-18]. Dostupné z: https://www.youtube.com/watch?v=C_smGQ_aN3c
- [13] LEGO.COM. Uživatelská příručka EV3 Software [online]. [cit. 2015-01-27]. Dostupné z: <http://www.lego.com/cs-cz/mindstorms/downloads>
- [14] BENEDETTELLI, Daniele. Programovanie robotov LEGO NXT pomocou NXC [online]. [cit. 2015-01-26]. Dostupné z: http://www.robosoutez.cz/files/NXT_tutorial_sk.pdf
- [15] NXC Documentation [online]. [cit. 2015-01-26]. Dostupné z: http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC_Guide.pdf
- [16] MOC, Ivan. LEGO Mindstorms – návrh a realizace úloh, návod pro programování v NXC. Dostupné z: http://www.robosoutez.cz/files/BP_Ivan_Moc.pdf. Bakalářská práce. ČVUT Praha.



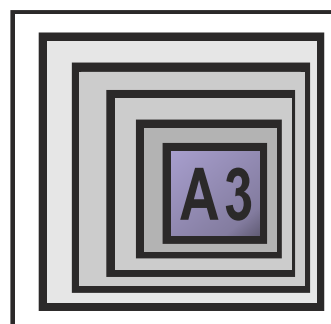
Kantor Ideál



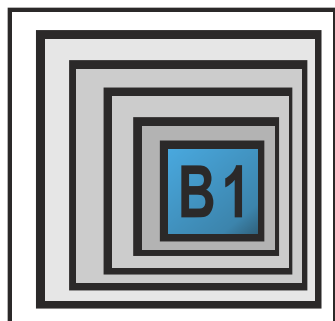
VZDĚLÁVÁNÍ ŘEDITELŮ



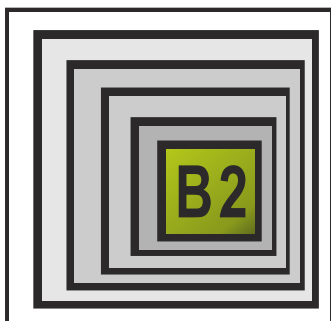
MENTORING



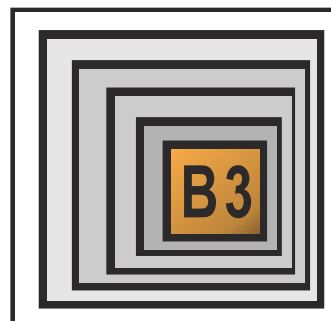
METODIK ICT VE ŠKOLE



CO UŽ MÁME



CO CHCEME



OBOROVÉ DIDAKTIKY