

Open Roberta Lab

Jan Preclík¹

e-mail: preclik@gymnachod.cz

¹ Jiráskovo gymnázium, Náchod, Řezníčkova 451

Klíčová slova

Open Roberta, výuková robotika, grafické programování, Blockly, LEGO Mindstorms, BBC micro:bit

I Open Roberta Lab

I.1 Historie

Open Roberta Lab (<https://lab.open-roberta.org>) je cloudové prostředí určené k programování nejrůznějších robotických stavebnic a výukových vývojových desek. Jde o Open Source projekt, prostředí je dostupné zcela zdarma a v češtině. Projekt vznikl v rámci německé vzdělávací iniciativy „Roberta— Learning with robots“ [1], kterou inicioval Fraunhofer Institute for Intelligent Analysis and Information Systems [2]. Cílem projektu je zpřístupnit žákům i učitelům vývojové prostředí, které na ně bude klást co nejmenší nároky (např. na nutnost instalace, znalost programování...). Projekt podporuje i firma Google.

I.2 Proč Open Roberta Lab?

V dnešní době máme k dispozici mnoho „hraček“ (= robotických stavebnic a vývojových desek), které můžeme v oblasti STE(A)M využívat. Každý výrobce většinou nabízí svoje vlastní softwarové řešení, jak jeho produkt programovat. Výhodou takového řešení je optimalizace pro konkrétní hardware a možnost využít všech možností, které nabízí. Nevýhodou může být nutnost se stále učit něco nového (filozofii ovládání, syntaxi...), snad nejhorší je situace, kdy se nové prostředí podobná jinému, ale v některých směrech se chová „trochu“ jinak.

Open Roberta Lab se pokouší nabídnout jednotné softwarové řešení pro různé HW platformy v čistě online podobě. Nevýhodou je horší spolupráce s hardwarem, kdy u některých platform nejsou podporovány všechny možnosti. Nevýhodou čistě online řešení (kromě nutnosti připojení k internetu) je také to, že pro spolupráci s některými platformami je nutno na PC nainstalovat komunikační software, který zajistí propojení mezi Open Roberta a připojeným HW.

2 Základy ovládání prostředí

Nejprve musíte zvolit, jaký hardware chcete programovat, buď v průvodci, který se objeví po spuštění prostředí, nebo v menu *roboti*. Pokud se zaregistrujete a přihlásíte (menu *uživatel*), budete moci využívat výhod cloudového úložiště pro ukládání vašich programů, sdílení nebo umístění do veřejně dostupné *galerie*.

Program (i bez přihlášení) můžete také exportovat/importovat do/z formátu XML (menu *úpravy*).

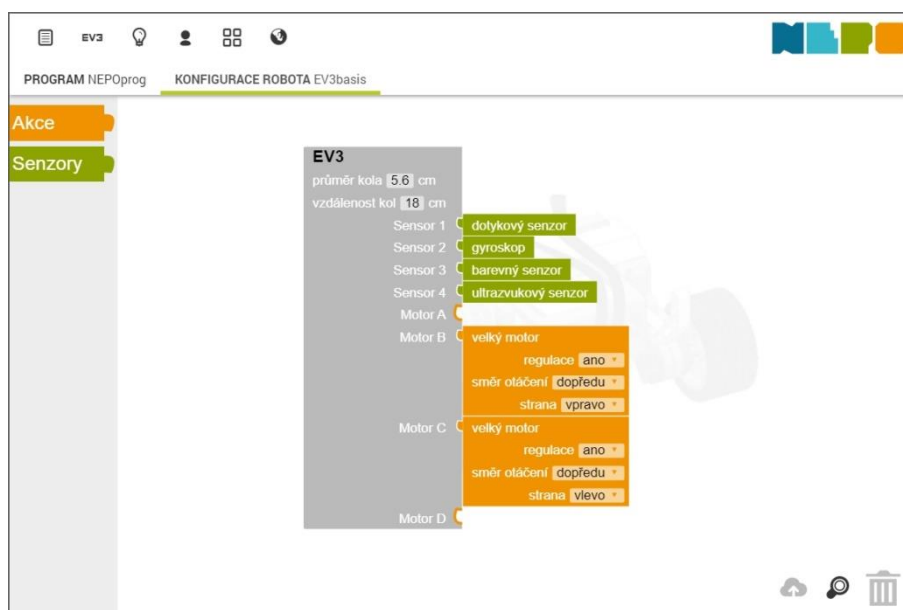
Programování je grafické pomocí **NEPO** bloků (**N**ew **E**asy **P**rogramming **O**nline), jde o kombinaci Blockly a Scratch. V menu *úpravy* nebo tlačítka nad panelem s bloky se můžete přepínat mezi NEPO bloky pro *začátečníka* nebo *pokročilého* uživatele. Bloky jsou uspořádány do kategorií (Akce, Senzory, Ovládání, Logické spojky, Matematika, Text, Seznamy, Barvy, Proměnné, Funkce, Zprávy) a barevně odlišeny. Pro každý hardware jsou pochopitelně k dispozici mírně odlišné bloky, ale řídicí struktury zůstávají stejné. Bloky tvořící váš program se přetahují do střední části pracovní plochy.



Prostředí Open Roberta Lab s popisy důležitých prvků

Pro některé roboty je na kartě *Konfigurace robota* k dispozici podrobnější nastavení – do kterého portu je připojen jaký senzor nebo motor, jaký je průměr kol robota atp.

V pravé části obrazovky můžete váš kód podrobněji komentovat, zobrazit zdrojový kód (který bude nahrán do vybraného hardwaru) nebo zobrazit simulátor (pokud je pro daný HW k dispozici).

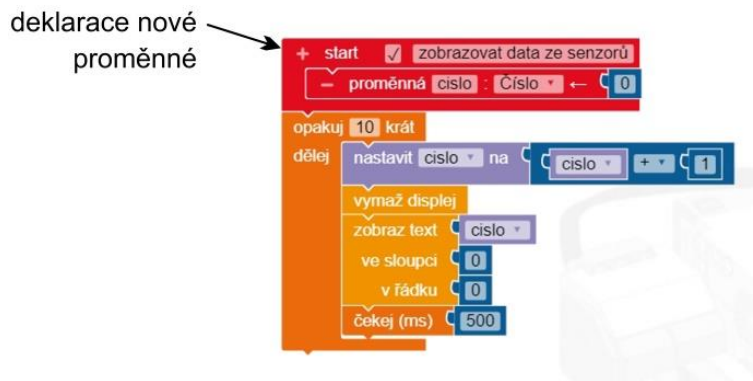


Prostředí Open Roberta Lab – ukázka konfigurace robota LEGO Mindstorms EV3

3 Programování

Nebudu zde podrobně popisovat, jak pomocí NEPO bloků programovat (kdo někdy pracoval s Blockly nebo Scratch, rychle se zorientuje) – jsou k dispozici všechny obvyklé programátorské konstrukce a datové struktury (podmínky, cykly, proměnné, pole...), zmíním jen pár bodů:

Proměnné je nutno nejprve deklarovat, to je děje kliknutím na + ve startovním bloku programu. Názvy proměnných v aktuální verzi nemohou obsahovat diakritiku. Deklarované proměnné uvidíte v sekci *Proměnné* a můžete je používat. Všechny takto deklarované proměnné jsou globální, tj. dostupné v celém programu.



Ukázka programování – deklarace a použití proměnné

Podprogramy mohou být procedury (bez návratové hodnoty) a funkce (s návratovou hodnotou), mohou mít i lokální proměnné, které také mohou sloužit jako vstupní parametry, bohužel ne jako výstupní (volané referencí).

Je škoda, že celá filozofie programování odpovídá spíše strukturovanému programování (od shora dolů...) než modernímu programování řízenému událostmi. To považuji za největší nedostatek prostředí, na nápravě se ale prý už pracuje.

4 Podporovaný hardware

Open Roberta Lab v současné době podporuje tento hardware (potřebné nastavení je popsáno na <https://jira.iais.fraunhofer.de/wiki/display/ORInfo>):

- robotická stavebnice **LEGO Mindstorms NXT**, stačí standardní firmware v řídicí kostce – Open Roberta generuje program v jazyce NXC [3],
- robotická stavebnice **LEGO Mindstorms EV3**, zde je nutno vložit do řídicí kostky SD kartu, na které je nainstalován operační systém buď
 - EV3 leJOS (verze 0.9.0 nebo 0.9.1 [4]) – Open Roberta generuje kód v jazyce Java, nebo
 - EV3dev [5] – Open Roberta generuje kód v jazyce Python,
- výukový robot **Bot'n Roll** [6], nutné jsou ovladače pro Arduino – Open Roberta generuje kód v jazyce C pro Arduino,
- humanoidní robot **NAO** [7] (jazyk Python),



Zleva LEGO Mindstorms NXT, EV3, Bot'n Roll, NAO (obrázky převzaty z [8], [9], [10], [11])

- výuková vývojová deska **BBC micro:bit** [12], Open Roberta generuje kód v jazyce Python, který je deskou přímo podporován,
- výuková vývojová deska **BOB3** [13] (s ní nemám žádné zkušenosti),
- výuková vývojová deska **Calliope** [14] je také přímo podporována přes jazyk C.



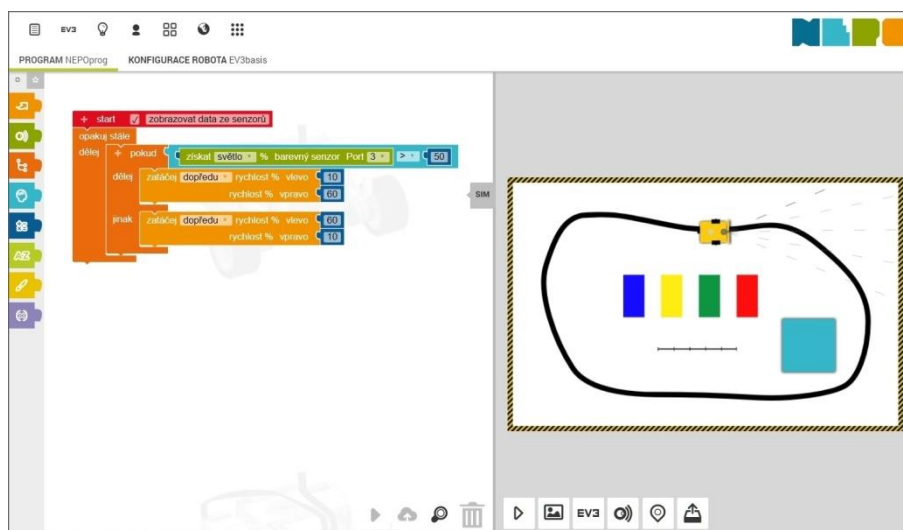
Zleva BBC micro:bit, BOB3, Calliope 2017 (obrázky převzaty z [15], [16], [17])

4.1 Simulátor

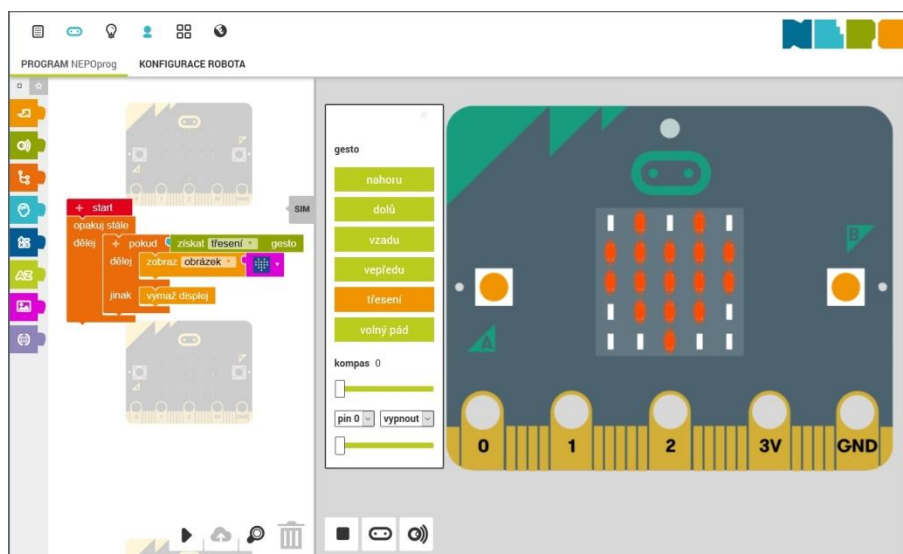
Pokud nemáte dostupný konkrétní hardware, můžete využít softwarového emulátoru, který je k dispozici pro LEGO Mindstorms NXT i EV3, BBC micro:bit a Calliope.

Simulátor pro LEGO Mindstorms umožní simulovat pohyb robota na různých „hřištích“, můžete i nahrát svůj vlastní obrázek hřiště, v průběhu simulace lze sledovat hodnoty z jednotlivých senzorů. Konfigurace robota pro simulátor je pevně daná, nemůžete tedy testovat různá konstrukční řešení. Ale na odladění základních algoritmů (zastavení před překážkou, objíždění překážky, jízda po čáře) je to postačující.

Simulátor desek BBC micro:bit a Calliope umožní simulovat stisk tlačítek a různá gesta (třesení...).



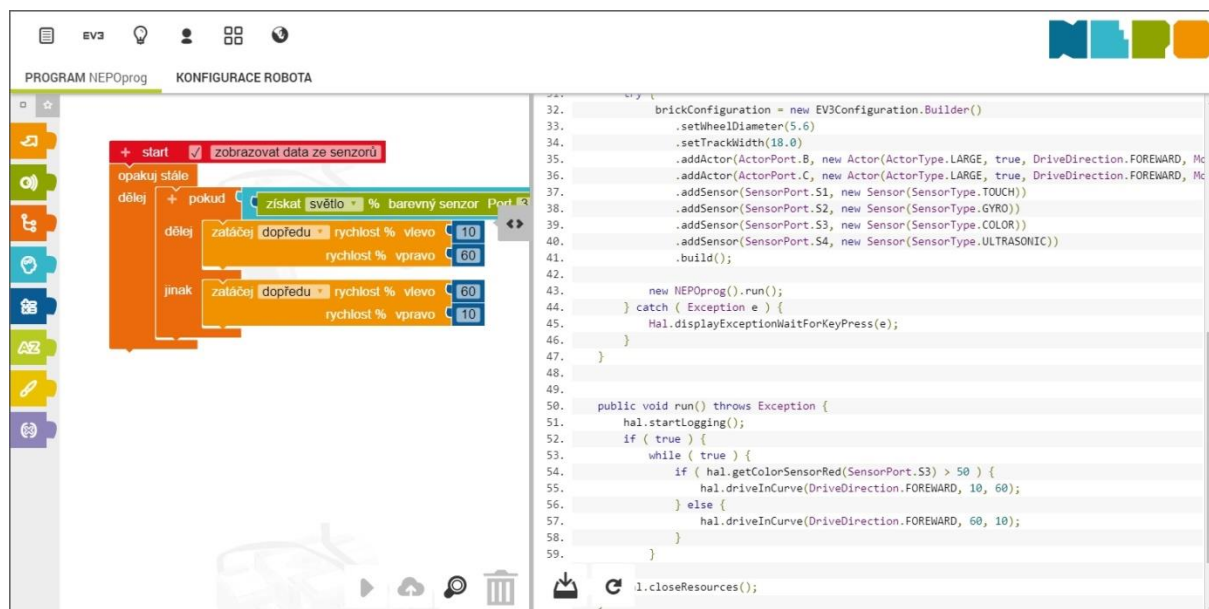
Simulátor EV3 – cik-cak algoritmus sledování čáry



Simulátor BBC micro:bit – reakce na zatřesení

4.2 Zdrojový kód

Open Roberta Lab funguje tak, že program z grafických NEPO bloků přeloží do zdrojového kódu, který je vhodný pro konkrétní hardware, a ten potom do něj (přímo nebo pomocí nějakého pomocného programu) nahraje. Zdrojový kód si můžete také stáhnout na disk a upravit, jen ale pozor, že Open Roberta využívá své vlastní knihovny.



EV3 – cik-cak algoritmus slevování čáry a zdrojový kód v Javě pro firmware LeJOS

5 Závěr

Open Roberta Lab určitě stojí za vyzkoušení, může být velkým pomocníkem především při začátcích s programováním. Nedostatek dostupného hardware může vyřešit softwarový simulátor, ten mohou žáci využívat například při domácí přípravě. Prostředí se stále vyvíjí, a proto doufám, že některé nedostatky (např. nepřítomnost programování řízeného událostmi) budou časem odstraněny.

Citace

- [1] *Roberta – Lernen mit Robotern* [online]. [cit. 2018-03-04]. Dostupné z: <https://www.open-roberta.org>.
- [2] *Fraunhofer IAIS* [online]. [cit. 2018-03-04]. Dostupné z: <https://www.iais.fraunhofer.de/en.html>.
- [3] *Bricx Command Center - NXT Information* [online]. [cit. 2018-03-04]. Dostupné z: <http://bricxcc.sourceforge.net>.
- [4] *LeJOS, Java for Lego Mindstorms / EV3* [online]. [cit. 2018-03-17]. Dostupné z: <http://www.lejos.org/ev3.php>.
- [5] *Ev3dev Home* [online]. [cit. 2018-03-17]. Dostupné z: <http://www.ev3dev.org>.
- [6] *Bot'n Roll ONE A Robot* [online]. [cit. 2018-03-04]. Dostupné z: http://botnroll.com/onea_en.
- [7] *Find out more about NAO* [online]. [cit. 2018-03-04]. Dostupné z: <https://www.ald.softbankrobotics.com/en/robots/nao/find-out-more-about-nao>.
- [8] REITTER, Christian. Lego mindstorms nxt main brick.jpg. In: *Wikimedia Commons* [online]. [cit. 2018-03-04]. Dostupné z: https://commons.wikimedia.org/wiki/File:Lego_mindstorms_nxt_main_brick.jpg. Licence CC-BY-SA 3.0.
- [9] Lego-mindstorms-ev3.jpg. In: *Wikimedia Commons* [online]. [cit. 2018-03-04]. Dostupné z: <https://commons.wikimedia.org/wiki/File:Lego-mindstorms-ev3.jpg>. Licence CC-BY-SA 3.0.
- [10] One_A.jpg [online]. [cit. 2018-03-04]. Dostupné z: http://botnroll.com/onea/img/One_A.jpg.
- [11] Softbank Robotics Europe. NAO Evolution.jpg. In: *Wikimedia Commons* [online]. [cit. 2018-03-04]. Dostupné z: https://commons.wikimedia.org/wiki/File:NAO_Evolution_.jpg. Licence CC-BY-SA 3.0.

- [12] *The Micro:bit Foundation* [online]. [cit. 2018-03-04]. Dostupné z: <http://microbit.org>.
- [13] *B·O·B·3* [online]. [cit. 2018-03-04]. Dostupné z: <https://www.bob3.org/en>.
- [14] *CALLIOPE* [online]. [cit. 2018-03-04]. Dostupné z: <https://calliope.cc/en>.
- [15] Gareth Halfacree. BBC micro:bit.jpg. In: *Flickr* [online]. [cit. 2018-03-04]. Dostupné z: <https://www.flickr.com/photos/120586634@N05/26212930836>. Licence CC-BY-SA 3.0.
- [16] bob-mit-helm.jpg. In: *Gymnasium Waldstrasse* [online]. [cit. 2018-03-04]. Dostupné z: <https://www.gy-waldstrasse.de/files/daten/schulprofil/mint/bob-mit-helm.jpg>.
- [17] it-2405954_1920. In: *Pixabay* [online]. [cit. 2018-03-04]. Dostupné z: <https://pixabay.com/cs/to-technologie-vzd%C4%9Bl%C3%A1v%C3%A1n%C3%AD-deska-2405954/>. Licence CC0 Creative Commons.