

Programování robotů LEGO Mindstorms EV3 pomocí NXC

Martin Hlinovský¹

e-mail: martin.hlinovsky@fel.cvut.cz

¹ České vysoké učení technické v Praze, Fakulta elektrotechnická, katedra řídicí techniky

Klíčová slova

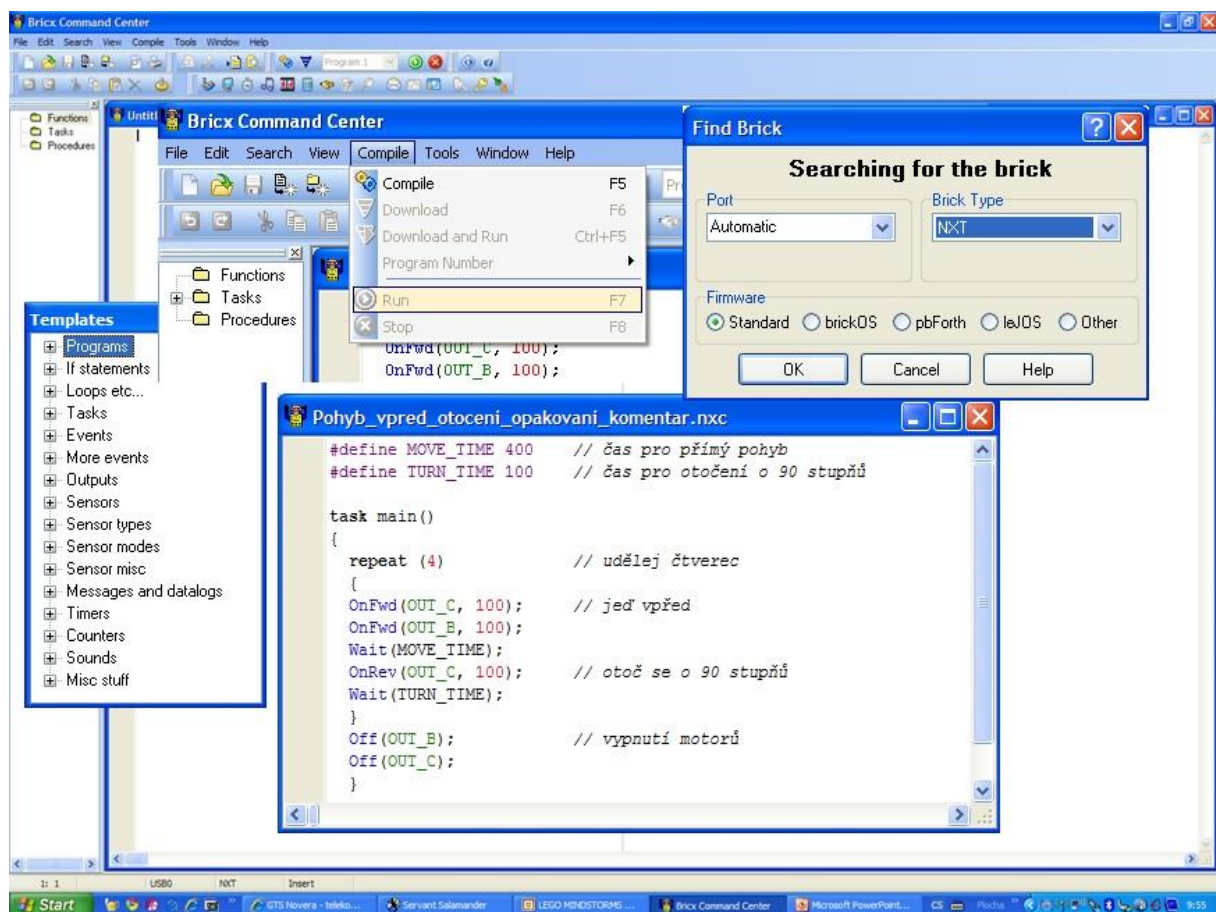
LEGO, Mindstorms, EV3, NXC, programování

I Programování robotů LEGO Mindstorms NXT pomocí NXC

I.1 Programovací jazyk NXC

NXC [1] je textový jazyk odvozený od jazyka C, který běží v programovacím prostředí BricxCC (viz obr. 1) na standardním firmwaru LEGO Mindstorms. Tato skutečnost je velmi příjemná pro ty, kteří chtějí programovat jak v NXT-G, tak v NXC, protože s každou změnou programovacího prostředí nemusí do kostky nahrávat nový firmware. Práce s jazykem zkracujícím spojení „Not eXactly C“ je velmi příjemná a programátor alespoň trochu znalý jazyka C si díky téměř stejné sémantice v tomto prostředí zvykne programovat velmi snadno. Další výhodou je, že se jedná o freewarovou aplikaci. Jako nevýhodu bych uvedl někdy nepřliš snadné debuggování programů. Na rozdíl od NXT-G se jedná o čistě textové programování bez grafických prvků.

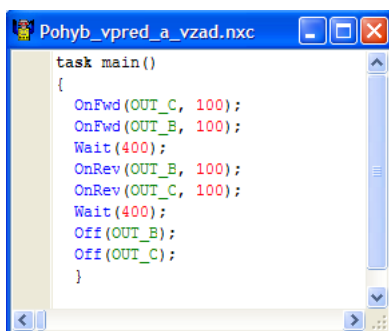
Programovací prostředí BricxCC bylo vyvinuto pro starší verzi LEGO Mindstorms NXT, bohužel však doposud neexistuje pro nejnovější verzi LEGO Mindstorms EV3 a to znemožňuje masívnějšímu rozšíření používání LEGO Mindstorms EV3 ve výuce (k dispozici je pouze grafický programovací jazyk, MATLAB, MonoBrick, ROBOTC (placená licence) nebo Java).



Obr. 1 Programovací prostředí BricxCC pro LEGO Mindstorms NXT

1.2 Příklad programů v NXC [1], [2]

Na obr. 2 je příklad programu pro pohyb robota vpřed a vzad. Na první pohled to vypadá velmi složitě, a tak se na něj podíváme zblízka. Program v NXC sestává z úloh (anglicky *task*). Náš program má jen jednu úlohu, nazvanou *main*. Každý program musí mít úlohu nazvanou *main* (anglicky hlavní), a tato úloha je zpracovávána při stisknutí tlačítka „RUN“. Úloha se skládá z množství příkazů, také nazývaných programové kroky (anglicky *statement*). Všechny programové kroky jsou pomocí složených závorek (tedy znaku { a }) uzavřeny do skupiny, aby bylo jasné, že všechny patří k této úloze. Každý programový krok je ukončen středníkem. Díky tomu je jasné, kde jeden programový krok končí a začíná druhý. Takže v podstatě každá úloha má strukturu jako v uvedeném příkladu.



```
Pohyb_vpřed_a_vzad.nxc
task main()
{
  OnFwd(OUT_C, 100);
  OnFwd(OUT_B, 100);
  Wait(400);
  OnRev(OUT_B, 100);
  OnRev(OUT_C, 100);
  Wait(400);
  Off(OUT_B);
  Off(OUT_C);
}
```

Obr. 2 Pohyb robota vpřed a vzad

Program je složen z několika programových řádků (kroků). Nyní je probereme jeden po druhém:

OnFwd(OUT_C, 100); Tento řádek říká robotu, aby zapnul výstup C (*On Forward Output C*), tedy motor připojený na výstup označený „C“ pro pohyb vpřed. Bude se pohybovat maximální rychlostí, nastaveno jako druhý parametr tohoto příkazu „100“.

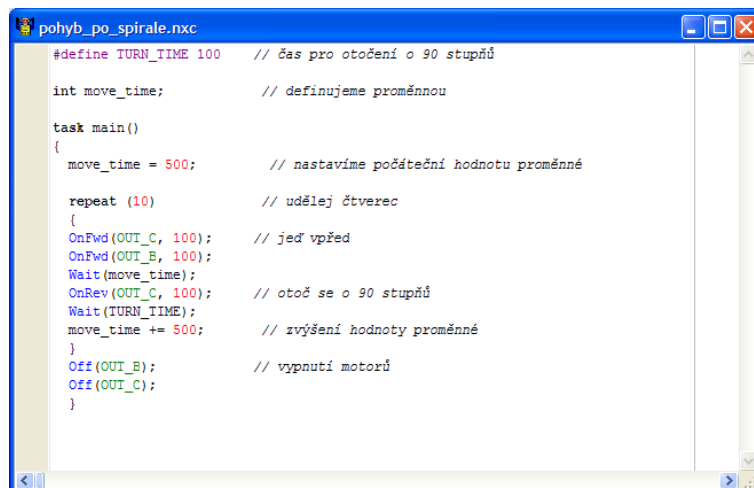
Wait(400); Nyní je čas chvíli počkat (*Wait*). Řádek říká, aby se počkalo 0,4 sekundy. Číslo mezi závorkami, udává počet „tiků“. Každý „tik“ trvá cca 1/1000 sekundy, takže můžete velice přesně určovat dobu čekání. Teď tedy po 0,4 sekundy program nic nedělá (tzv. „spí“) a robot proto pokračuje v pohybu vpřed.

OnRev(OUT_C, 100); Robot už ujel dost daleko a tak je čas říci mu, aby jel v opačném směru (*On Reverse Output C*), tedy zpět.

Off(OUT_C); Vypnutí (*Off*) motoru C.

Pravděpodobně jste si všimli barevného značení při psaní programu. Barvy se objevují automaticky. Vše co je v modré barvě, jsou příkazy pro robota, názvy motoru nebo jiných věcí které robot zná. Slovo *task* je vypsáno tučně, protože je důležitým (rezervovaným) slovem NXC. Barvy jsou užitečné, protože už při psaní uvidíme, že jsme neudělali chybu.

Na obr. 3 je příklad programu pro pohyb robota po spirále (s ukázkou definování konstant, proměnných a s využitím příkazu *repeat*).



```
pohyb_po_spirale.nxc
#define TURN_TIME 100 // čas pro otočení o 90 stupňů

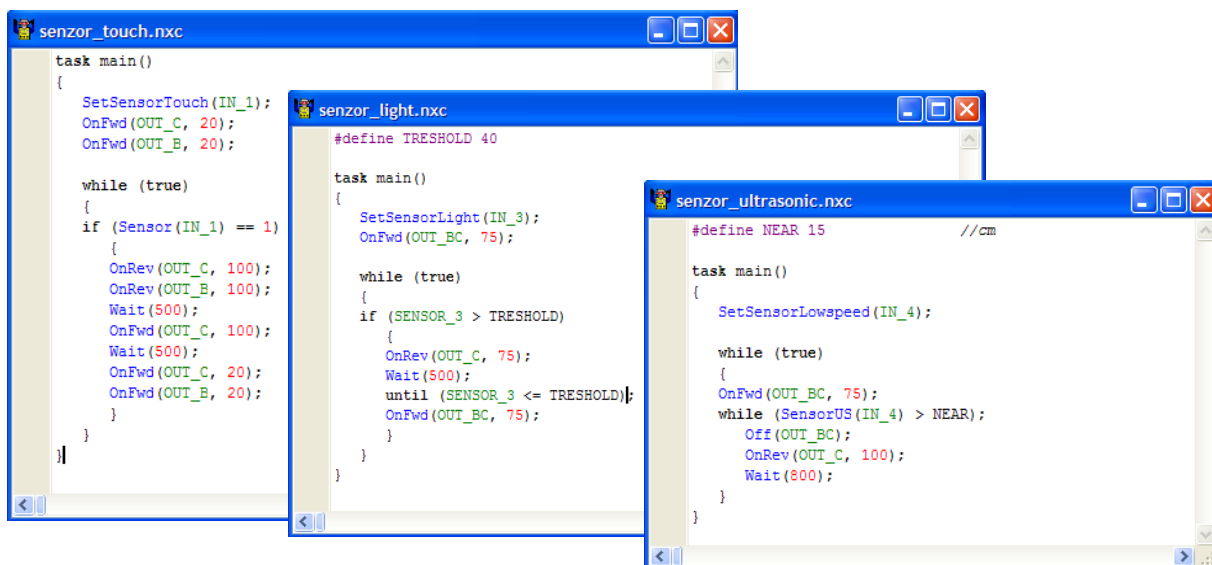
int move_time; // definujeme proměnnou

task main()
{
  move_time = 500; // nastavíme počáteční hodnotu proměnné

  repeat (10) // udělej čtverec
  {
    OnFwd(OUT_C, 100); // jed' vpřed
    OnFwd(OUT_B, 100);
    Wait(move_time);
    OnRev(OUT_C, 100); // otoč se o 90 stupňů
    Wait(TURN_TIME);
    move_time += 500; // zvýšení hodnoty proměnné
  }
  Off(OUT_B); // vypnutí motorů
  Off(OUT_C);
}
```

Obr. 3 Pohyb robota po spirále

Na obr. 4 jsou příklady programů použití dotykového, ultrazvukového a světelného senzoru (zkuste sami přijít na funkci programů).



```
task main()
{
  SetSensorTouch(IN_1);
  OnFwd(OUT_C, 20);
  OnFwd(OUT_B, 20);

  while (true)
  {
    if (Sensor(IN_1) == 1)
    {
      OnRev(OUT_C, 100);
      OnRev(OUT_B, 100);
      Wait(500);
      OnFwd(OUT_C, 100);
      Wait(500);
      OnFwd(OUT_C, 20);
      OnFwd(OUT_B, 20);
    }
  }
}

#define TRESHOLD 40

task main()
{
  SetSensorLight(IN_3);
  OnFwd(OUT_BC, 75);

  while (true)
  {
    if (SENSOR_3 > TRESHOLD)
    {
      OnRev(OUT_C, 75);
      Wait(500);
      until (SENSOR_3 <= TRESHOLD);
      OnFwd(OUT_BC, 75);
    }
  }
}

#define NEAR 15 //cm

task main()
{
  SetSensorLowspeed(IN_4);

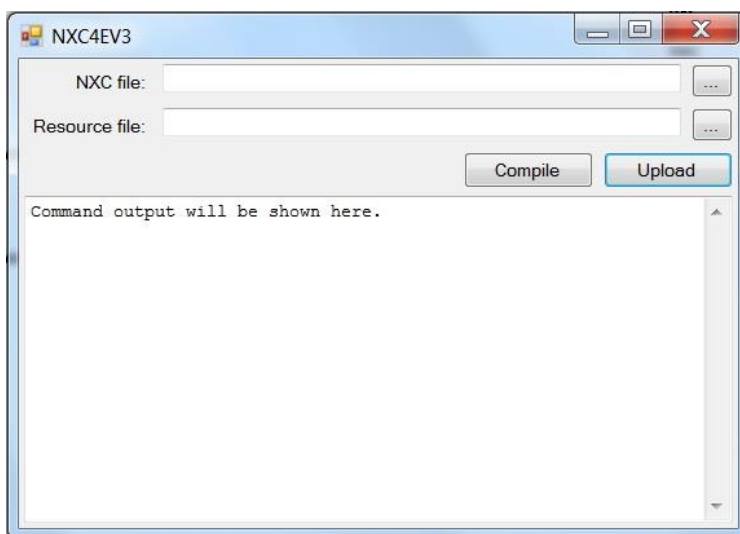
  while (true)
  {
    OnFwd(OUT_BC, 75);
    while (SensorUS(IN_4) > NEAR);
    Off(OUT_BC);
    OnRev(OUT_C, 100);
    Wait(800);
  }
}
```

Obr. 4 Příklady programů použití dotykového, ultrazvukového a světelného senzoru

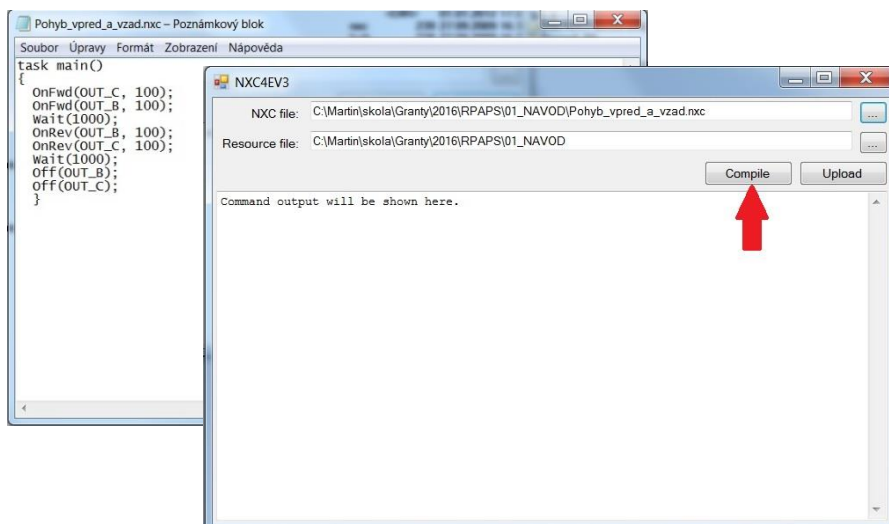
2 Programování robotů LEGO Mindstorms EV3 pomocí NXC – NXC4EV3

NXC4EV3 je nástroj pro spuštění programů napsaných v programovacím jazyce NXC na EV3 kostce. A jak program NXC4EV3 použít? Je to velice jednoduché:

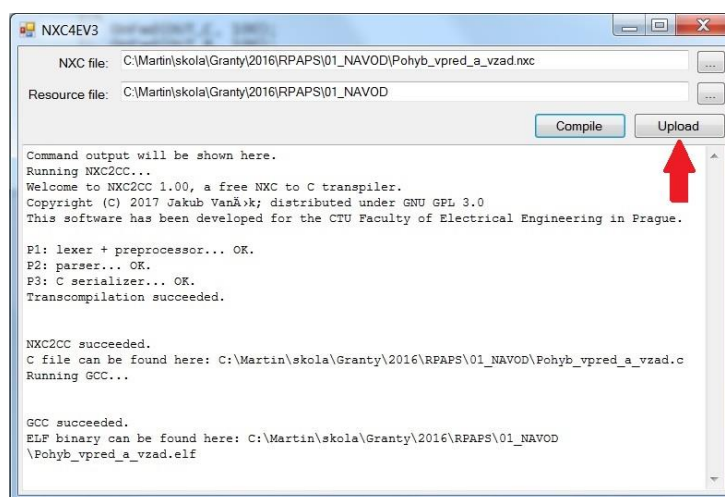
1. Otevřete program NXC4EV3 (obr. 5)
2. Vyberte zdrojový soubor v jazyce NXC pomocí tlačítka vedle pole pro „NXC file“ (Obr. 6 – použijeme příklad z předchozí kapitoly pro pohyb vpřed a vzad). „Resource file“ se doplní automaticky. Pro editaci zdrojových kódů třeba používat např. původní [Bricx Control Center](#), NXC4EV3 se stará pouze o kompatibilitu s EV3.
3. Pro sestavení klepněte na „Compile“ (Obr. 6). Pokud překlad proběhl bez problémů, můžete pokračovat dále.
4. Pro nahrání na EV3 kostku klepněte na „Upload“ (Obr. 7). Kostka musí být k počítači připojena přes USB. Pokud nahrání bylo úspěšné, program naleznete na kostce v adresáři, který se jmenuje dle vstupního NXC souboru.



Obr. 5 Základní okno programu NXC4EV3



Obr. 6 Vybrání zdrojového souboru a jeho sestavení



Obr. 7 Nahrání programu do EV3 kostky

3 Závěr

Cílem vývoje vlastního softwaru NXC4EV3 bylo umožnit programování robotů LEGO Mindstorms EV3 pomocí programovacího jazyka NXC a tím umožnit masívnější používání těchto stavebnic ve výuce. Tento software je ještě zajímavější díky tomu, že můžeme použít programy napsané pro LEGO Mindstorms NXT a rovnou je nahrávat do EV3 kostky bez jakýchkoliv úprav. Program je volně k dispozici ke stažení na webových stránkách ROBOSOUTĚŽE [5] včetně návodu na jeho použití.

Citace

- [1] BENEDETTELLI, Daniel. *Programming LEGO NXT robots using NXC* [online]. 2007. 51 s. Dostupné z WWW: <http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC_tutorial.pdf>
- [2] MOC, Ivan. *Využití robota LEGO Mindstorms – návrh a realizace úloh, návod pro programování v NXC* [online]. Bakalářská práce, 2010. 66 s. Dostupné z WWW: <http://www.robosoutez.cz/files/BP_Ivan_Moc.pdf>
- [3] <http://bricxcc.sourceforge.net/>
- [4] <http://www.robosoutez.cz/index.php?sekce=home&id=nxc>
- [5] <http://www.robosoutez.cz/index.php?sekce=nxc4ev3&id=nxc4ev3>