

DUM č. 20 v sadě

29. Inf-5 RoboLab a Lego Mindstorms

Autor: Hana Křetínská

Datum: 25.06.2014

Ročník: 1AV, 2AV, 3AV, 4AV, 5AV

Anotace DUMu: Výuka robotiky a programování pomocí stavebnice Lego a programu Lego Mindstorms a programování v prostředí BricxCC. Sestavení robotků z Lega a jejich programování na počítači pro žáky 2. stupně ZŠ a první a druhý ročník čtyřletého gymnázia. Lego Mindstorms a funkce v NXC.

Materiály jsou určeny pro bezplatné používání pro potřeby výuky a vzdělávání na všech typech škol a školských zařízení. Jakékoliv další využití podléhá autorskému zákonu.



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

20. Lego Mindstorms a funkce v NXC

Funkce v NXC

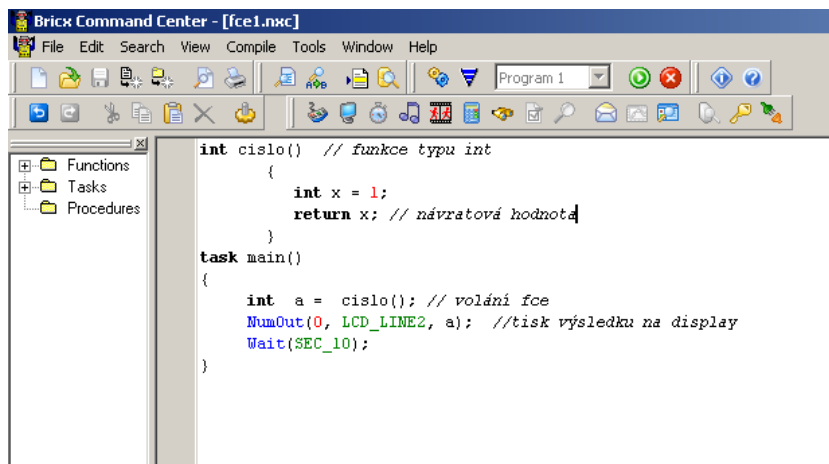
Funkce v NXC mají za úkol provést nějakou akci s návratovou hodnotou. Seznam argumentů může být prázdný nebo může obsahovat jeden a více argumentů, které jsou od sebe odděleny čárkou. Argument je představován typem a jménem proměnné. Proměnná může být typu bool, char, byte, int, short, long, unsigned int, float, strings, struct typy nebo array. Deklarace fce je před task main().

- Příklad funkce, která nemá žádný vstup a výstupem je proměnná typu int:

```
int cislo()
{
    int x = 1;
    return x;
}
```

Pro volání funkce použijeme příkaz:

```
int a = cislo();
```



Po spuštění programu v NXT kostce se provede volání fce cislo a výsledná hodnota se přiřadí do proměnné a. Hodnota proměnné a =1 se zobrazí na displeji na druhém řádku.

- Příklad funkce, která má dva vstupní parametry a výstupem je proměnná typu int:

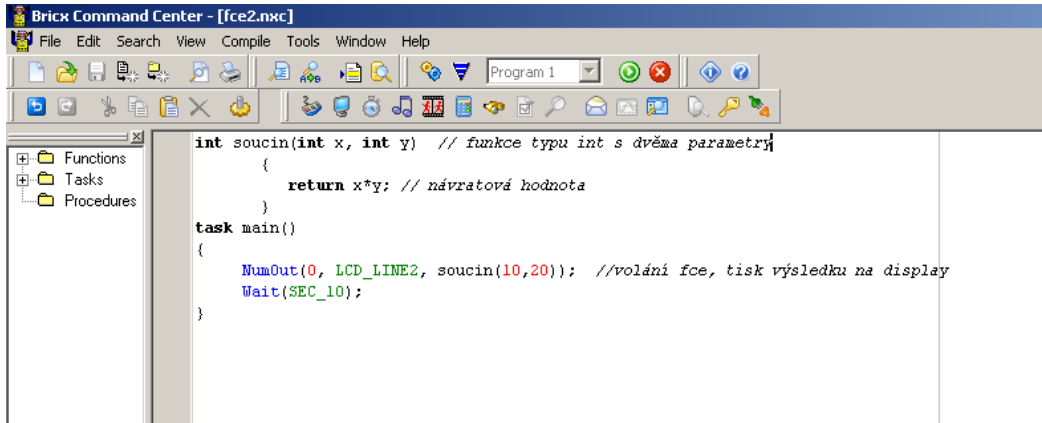
```
int soucin(int x, int y) // funkce typu int se dvěma parametry
```

```
{
```

```
    return x*y; // návratová hodnota
}
```

Po zavolání funkce se na displeji objeví součin dvou parametrů 10 a 20 v závorce funkce:

```
NumOut(0, LCD_LINE2, soucin(10,20));
```



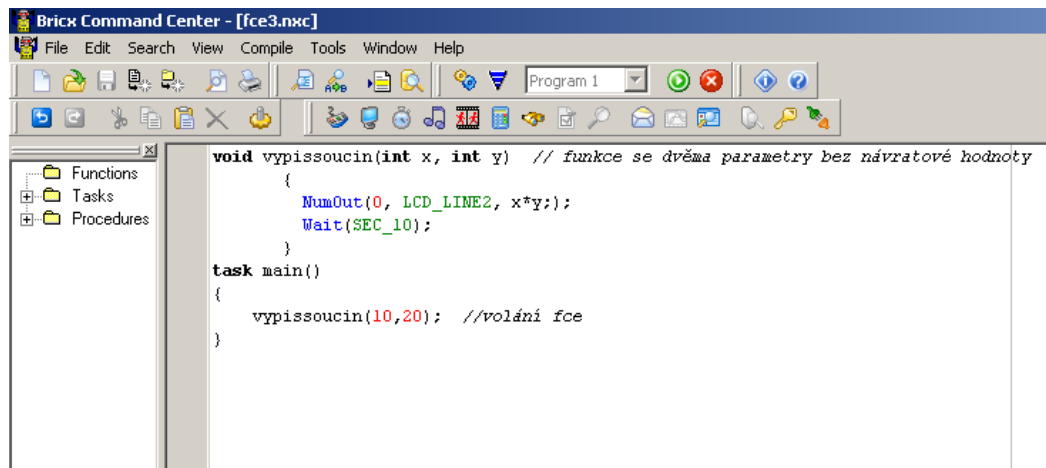
```
Brick Command Center - [fce2.nxc]
File Edit Search View Compile Tools Window Help
Program 1
Functions
Tasks
Procedures
int soucin(int x, int y) // funkce typu int s dvěma parametry
{
    return x*y; // návratová hodnota
}
task main()
{
    NumOut(0, LCD_LINE2, soucin(10,20)); //volání fce, tisk výsledku na display
    Wait(SEC_10);
}
```



- Příklad funkce, která má dva vstupní parametry a nemá žádnou návratovou hodnotu (procedura):

```
void vypissoucin(int x, int y) // funkce se dvěma parametry bez návrat. hodnoty
{
    NumOut(0, LCD_LINE2, x*y);
}
```

Volání funkce se dvěma parametry:
vypissoucin(10,20); //volání fce



Po spuštění programu a zavolání funkce se na displej vypíše hodnota 20.

Některé další příkazy Jazyka NXC

Příkaz SWITCH:

switch (proměnná)

{

case 5: //je-li hodnota proměnné 5, vykoná se následující příkaz

a = 5;

break;

case 1:

case 2: //je-li hodnota proměnné 1 nebo 2, vykoná se následující příkaz

a = 1;

break;

default: // je-li hodnota proměnné jiná než 5,1,2, vykoná se následující příkaz

a = 10;

break;

}

Okamžité zastavení motoru

příkazem Off(název konektoru):

```
OnFwd(OUT_A,100);
```

```
Wait(500);
```

```
Off(OUT_A); //motor okamžitě zastaví (brzda)
```

Pozvolné zastavení motoru

se samovolným dotočením příkazem Coast(název konektoru)

```
OnFwd(OUT_A,100);
```

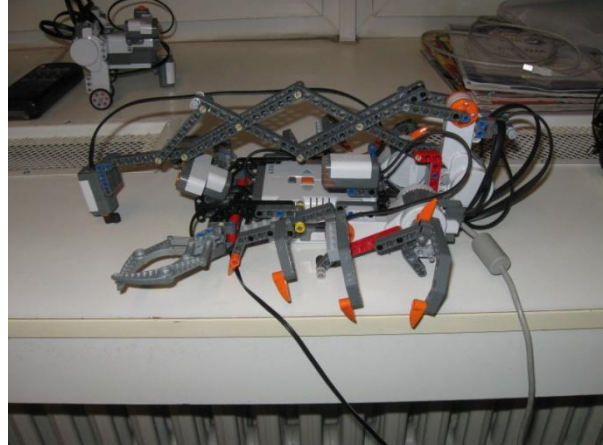
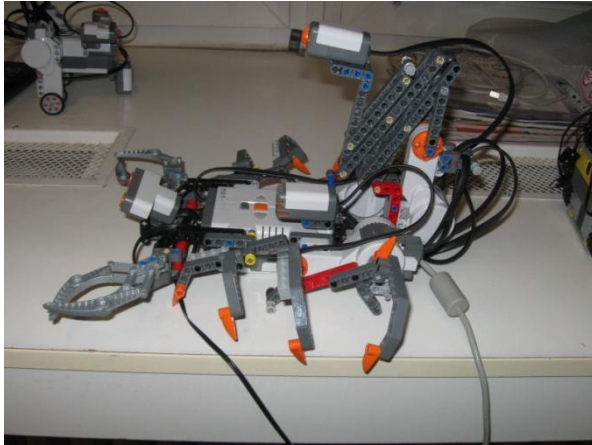
```
Wait(500);
```

```
Coast(OUT_A); //motor pozvolna zastaví
```

Pootočení motoru

Přesně specifikovaný pohyb pootočením motoru o určitý úhel příkazem RotateMotor(název konektoru ,síla,úhel). Motor připojený na konektor se bude otáčet silou o úhel.

```
RotateMotor(OUT_A,60,90);
```



1 Využití pootočení motoru s mechanickou rukou

Synchronizace motorů

Pokud aplikace vyžaduje souběžný chod motorů a každá baterie je jinak nabitá, je nutné motory synchronizovat příkazem OnFwdSync(název konektoru,síla,rozdíl), Vstupní parametr rozdíl definuje rozdíl mezi jednotlivými motory.

```
OnFwdSync(OUT_AB,75,-100); //robot by zatáčet vpravo
```

```
OnRevSync(OUT_AB,75,-100); // robot by zatáčet vlevo
```

Snímač otáček

Motor může fungovat jako snímač otáček příkazem

```
MotorTachoCount(název konektoru)
```

Pro vymazání příkaz

```
ResetTachoCount(název konektoru)
```

```
int pocitadlo;
```

```
ResetTachoCount(OUT_A);
```

```
OnFwd(OUT_A);
```

```
Wait(1000);
```

```
pocitadlo = MotorTachoCount(OUT_A); // pootočení
```

Vnořování příkazů

Robot objedná 10 čtverců (o čtyřech stranách)

```
// 10 čtverců
```

```
#define MOVE_TIME 500
```

```

#define TURN_TIME 360
task main()
{
repeat(10) // 10 čtvrců
{
repeat(4) // 4 strany
{
OnFwd(OUT_AC, 75);
Wait(MOVE_TIME);
OnRev(OUT_C, 75);
Wait(TURN_TIME);
}
}
Off(OUT_AC); // Motory se zastaví
}

```

Příklady použití zvukového a ultrasonického senzoru:

Zvukový senzor

```

#define THRESHOLD 40
#define MIC SENSOR_2
task main()
{
SetSensorSound(IN_2);
while(true){
until(MIC > THRESHOLD);
OnFwd(OUT_AC, 75);
Wait(300);
until(MIC > THRESHOLD);
Off(OUT_AC);
Wait(300);
}
}

```

Ultrasonic senzor

```

#define NEAR 15 //cm
task main(){
SetSensorLowspeed(IN_4);
while(true){
OnFwd(OUT_AC,50);
while(SensorUS(IN_4)>NEAR);
Off(OUT_AC);
OnRev(OUT_C,100);
}
}

```

```
Wait(800);  
}  
}
```

Úkol : Naprogramujte funkce na aritmetické operace přirozených čísel a výsledky vytiskněte na display.

Zdroj obrázků: Vlastní tvorba na PC s programem Mindstorms, Bricx Command Center a vlastní fotografie.