



Programando Robôs Lego NXT usando NXC

Prof. Marcelo Santos Linder

Sumário

- Robô (kit) Lego NXT
- Linguagem NXC
 - Estrutura geral de um programa
 - Estruturas de controle de fluxo
 - Motores
 - Execução de códigos de exemplo em um robô móvel
 - Sensores
 - Execução de códigos de exemplo em um robô móvel
 - Apresentação de robô móvel programado com NXC executando tarefa proposta em trabalho de disciplina
 - Oficina Programando Robôs Lego NXT usando NXC
- Bibliografia

Robô (kit) Lego NXT

- Parceria LEGO MIT (1988) -> "tijolo inteligente"
- Objetivo trazer vida às criações LEGO através de programação em computadores.
- Em 1998 o tijolo inteligente Lego Mindstorms RCX é revelado à imprensa no Museu de Arte Moderna, em Londres.
- Em 2006, em Las Vegas, a LEGO apresentou ao público a nova geração do Mindstorms.
- O Mindstorms NXT, um kit de robótica direcionado para o público pré-adolescente (10+), composto por sensores e atuadores, que visa fomentar o estudo/interesse por ciências e engenharia.
- Em 2013 foi lançada a versão LEGO Mindstorms EV3.

Robô (kit) Lego NXT

- Sensores e atuadores presentes no kit Lego MindStorms NTX em sua primeira versão.



Robô (kit) Lego NXT

- Sensores e atuadores presentes no kit Lego MindStorms NTX em sua segunda versão.



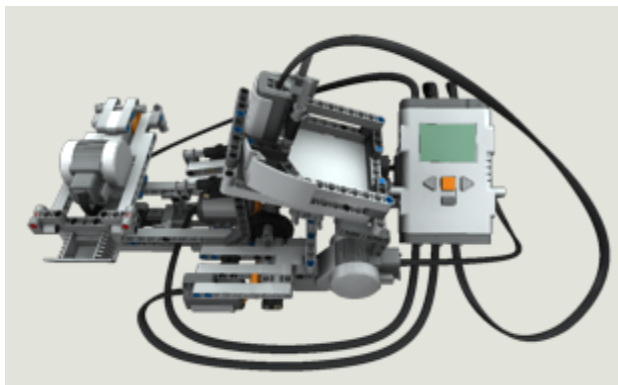
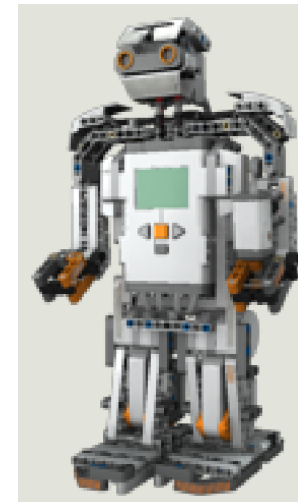
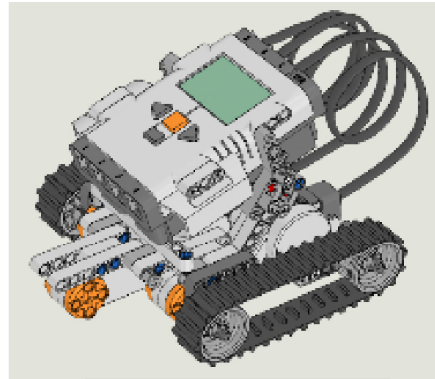
Robô (kit) Lego NXT

- O kit possui 619 peças



Robô (kit) Lego NXT

- O kit permite a construção de inúmeros robôs com configurações diferentes.
- Exemplos:



Robô (kit) Lego NXT

Como pode ser indicado para crianças?

- Apresentando tutoriais de como montar configurações de robôs.
- Utilizando o NXT-G que é uma linguagem gráfica similar ao National Instruments LabVIEW, que vem junto com o conjunto vendido.

Agora vamos explorar o aplicativo/ambiente...

Linguagem NXC

- Visando ampliar a capacidade de explorar a arquitetura disponibilizada pelo Mindstorms NXT foi proposta uma linguagem mais robusta/flexível que a NXT-G.
- Esta linguagem foi denominada NXC.
- NXC é uma linguagem de programação inventada por John Hansen que foi especialmente desenvolvida para os robôs Lego.

Linguagem NXC

- Visando facilitar a programação em NXC, foi criado o Brick Command Center (BricxCC – Centro de Comando Brick).
- BricxCC dá suporte a:
 - escrita de programas;
 - download de programas para o robô;
 - navegar na memória flash do NXT;
 - e muito mais.

Linguagem NXC

- O BricxCC pode ser visualizado como um ambiente de desenvolvimento integrado (IDE - Integrated Development Environment).
- É possível fazer o download gratuito do BricxCC no link:

<http://bricxcc.sourceforge.net/>

Linguagem NXC

- BricxCC roda em PCs com Windows (95, 98, ME, NT, 2000, XP, Vista).
- A linguagem NXC também pode ser utilizada em outras plataformas.
- Sendo possível fazer o download do compilador no link:

<http://bricxcc.sourceforge.net/nxc/>

Linguagem NXC

- A estrutura básica de um programa em NXC é:

```
task main()  
{  
    ...  
}
```

Linguagem NXC

- Seu tipos são:
 - bool
 - byte
 - char
 - int
 - short
 - long
 - unsigned
 - float
 - mutex
 - string
 - Structures
 - Arrays
- Exemplos de declarações de variáveis

```
task main()
{
    int x;
    bool y, z;
    long a=1,b;
    float f=1.15, g;
    int data[10];
    bool flags[] = {true, true, false, false};
    string msg = "hello world";
    ...
}
```

Linguagem NXC

- Possui estruturas de controle de fluxo como:
 - if else
 - Sintaxe exatamente igual a da linguagem C
 - while
 - Sintaxe exatamente igual a da linguagem C
 - do while
 - Sintaxe exatamente igual a da linguagem C
 - for
 - Sintaxe exatamente igual a da linguagem C

Linguagem NXC

- Atuadores (motores)

Os atuadores podem ser associados aos periféricos de saída.

São conectados às saídas do console NXT.

Estas saídas estão associadas aos identificadores OUT_A, OUT_B e OUT_C.

Três funções básicas (existem outras) estão associadas à manipulação dos motores:

- OnFwd (saída(s), velocidade);
- OnRev (saída(s), velocidade);
- Off (saída(s)).

Linguagem NXC

- Acionamento dos motores (atuadores)

Exemplo 1:

```
task main()
```

```
{
```

```
    OnFwd(OUT_A, 75); //75% da velocidade máxima
```

```
    OnFwd(OUT_C, 75);
```

```
    Wait(4000); //valor em milissegundos
```

```
    OnRev(OUT_AC, 75);
```

```
    Wait(4000);
```

```
    Off(OUT_AC);
```

```
}
```

Linguagem NXC

- Sensores

Podem ser associados aos periféricos de entrada.

Os sensores do NXT permitem que o robô reaja a estímulos externos.

As entradas onde podem ser conectados os sensores estão associadas aos identificadores IN_1, IN_2, IN_3 e IN_4.

Começaremos nossos estudos pelo **sensor de toque**.

Para uma melhor compreensão vamos observar um exemplo de utilização do sensor de toque (considerando que este está conectado à entrada 1 do NXT).

Linguagem NXC

```
//Exemplo2: movimentação dos motores para
// quando o sensor de toque é pressionado
task main()
{
    SetSensor(IN_1, SENSOR_TOUCH);
    OnFwd(OUT_AC, 25);
    until (SENSOR_1 == 1);
    Off(OUT_AC);
}
```

Linguagem NXC

//Exemplo3: sensor de toque utilizado como um para-choque

task main()

{

SetSensorTouch(IN_1);

 OnFwd(OUT_AC, 75);

 while (true)

 {

 if (**SENSOR_1 == 1**)

 {

 OnRev(OUT_AC, 75); Wait(300);

 OnFwd(OUT_A, 75); Wait(300);

 OnFwd(OUT_AC, 75);

 }

 }

}

Linguagem NXC

Além do sensor de toque, você também recebe um sensor de luz, um sensor de som e um sensor digital ultrassônico com o sistema Mindstorms NXT (em sua primeira versão).

Os sensores que acompanham o kit NXT 2.0 são: o de toque (duas unidades), o de ultrassom e o de cor/luz .

Sensor de cor/luz

O sensor de luz pode ser acionado para emitir luz ou não, então você pode mensurar a quantidade de luz refletida ou de luz ambiente em uma direção particular.

Medir a luz refletida é particularmente útil quando, por exemplo, se deseja fazer o robô seguir uma linha no chão.

Linguagem NXC

Isso é o que vamos fazer no próximo exemplo, onde conectaremos o sensor de cor à entrada 3 e o utilizaremos com sensor de luz (disponível na primeira versão do kit NXT).

```
//Exemplo4
```

```
#define THRESHOLD 40
```

```
task main() {
```

```
    SetSensor(IN_2, SENSOR_COLORRED);
```

```
    OnFwd(OUT_AC, 35);
```

```
    while (true) {
```

```
        if (Sensor(IN_2) > THRESHOLD) {
```

```
            OnRev(OUT_C, 35);
```

```
            Wait(100);
```

```
            until(Sensor(IN_2) <= THRESHOLD);
```

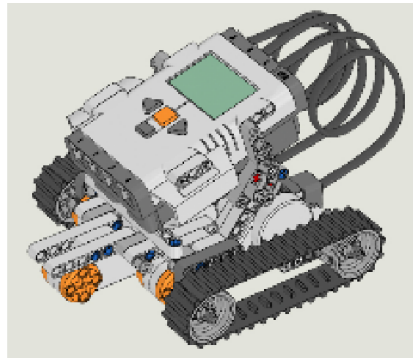
```
            OnFwd(OUT_AC, 35);
```

```
        }
```

```
    }
```

Linguagem NXC

O exemplo anterior representa uma implementação bem simples de um robô seguidor de linha com a configuração abaixo:



Onde consideraremos que o motor ligado na saída C move a esteira mais abaixo na imagem.

Neste caso, o programa apresentado serve apenas para seguir trilhas no sentido horário.

Linguagem NXC

No exemplo anterior utilizamos o sensor para emitir luz e verificar a intensidade da luz refletida.

Para ler a intensidade da luz ambiente com o LED desligado, configure o sensor da seguinte forma:

Para o sensor de cor (kit NXT 2.0)

```
SetSensorType(IN_3, IN_TYPE_COLORNONE);  
SetSensorMode(IN_3, IN_MODE_PCTFULLSCALE);  
ResetSensor(IN_3);
```


Linguagem NXC

Sensor de cor/luz

Quando explorado como sensor de cor pode ser utilizado para identificar seis cores básicas especificadas pelos identificadores:

- INPUT_BLACKCOLOR
- INPUT_BLUECOLOR
- INPUT_GREENCOLOR
- INPUT_YELLOWCOLOR
- INPUT_REDCOLOR
- INPUT_WHITECOLOR

Para entender como utilizar o sensor de cor para identificar as cores mencionadas veremos um exemplo.

```

task main() { //Exemplo5
    SetSensorColorFull(S3);
    while (true) {
        ClearScreen();
        TextOut(0,LCD_LINE1,"Saida do sensor");
        if (Sensor(S3) == INPUT_BLACKCOLOR) {
            TextOut(0,LCD_LINE2,"Cor preta");
        }else{
            if (Sensor(S3) == INPUT_BLUECOLOR){
                TextOut(0,LCD_LINE2,"Cor azul");
            }else{
                if (Sensor(S3) == INPUT_GREENCOLOR){
                    TextOut(0,LCD_LINE2,"Cor verde");
                }else{
                    if (Sensor(S3) == INPUT_YELLOWCOLOR){
                        TextOut(0,LCD_LINE2,"Cor amarela");
                    }else{
                        if (Sensor(S3) == INPUT_REDCOLOR){
                            TextOut(0,LCD_LINE2,"Cor vermelha");
                        }else{
                            if (Sensor(S3) == INPUT_WHITECOLOR){
                                TextOut(0,LCD_LINE2,"Cor branca");
                            }else{
                                TextOut(0,LCD_LINE2,"Cor indefinida!");
                            }
                        }
                    }
                }
            }
        }
    }
}
Wait(200);
}

```

Linguagem NXC

Sensor Ultrassônico

O sensor ultrassônico funciona como um sonar: falando grosseiramente, ele envia uma rajada de ondas ultrassônicas e mede o tempo necessário para que as ondas sejam refletidas de volta pelo objeto em vista.

Com esse sensor você pode fazer um robô enxergar e evitar um obstáculo antes de necessariamente atingi-lo (como no caso de um sensor de toque).

Linguagem NXC

//Exemplo6 - utilização do sonar

```
#define NEAR 15 //cm
```

```
task main(){
```

```
    SetSensorLowspeed(IN_3);
```

```
    while(true){
```

```
        OnFwd(OUT_AC,50);
```

```
        while(SensorUS(IN_3)>NEAR); //ou until
```

```
        Off(OUT_AC);
```

```
        OnRev(OUT_C,100);
```

```
        Wait(800);
```

```
    }
```

```
}
```

Linguagem NXC

Mais sobre motores

Existe um número de comandos adicionais para motor que podem ser usados para controlar os motores mais precisamente.

Linguagem NXC

- Parando suavemente

Quando você usa o comando `Off()`, o servo-motor para imediatamente, freando o eixo e mantendo a posição.

Também é possível parar os motores de um modo mais suave, não usando os freios.

Para isso usa-se o comando `Float()` ou `Coast()` (não há distinção), que simplesmente corta a força do motor.

Linguagem NXC

```
task main() //exemplo7
```

```
{
```

```
    OnFwd(OUT_AC, 75);
```

```
    Wait(500);
```

```
    Off(OUT_AC);
```

```
    Wait(1000);
```

```
    OnFwd(OUT_AC, 75);
```

```
    Wait(500);
```

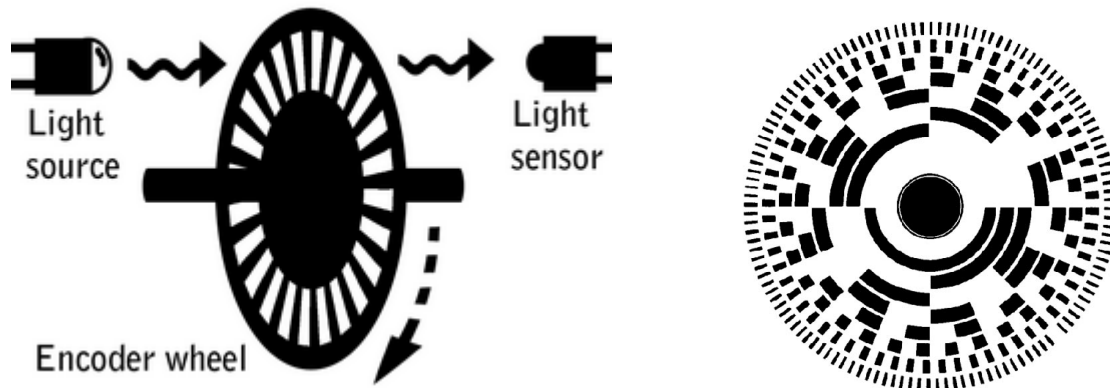
```
    Float(OUT_AC);
```

Linguagem NXC

- Movendo o eixo do motor um determinado ângulo

Os comandos `OnFwd()` e `OnRev()` são as mais simples rotinas para se mover motores.

Os servo-motores do NXT possuem um codificador interno (encoder) que permite que você controle precisamente a posição do eixo e sua velocidade.



Programando Robôs Lego NXT usando NXC

Linguagem NXC

- Movendo o eixo do motor um determinado ângulo

Com o uso dos encoders é possível, utilizando a função `RotateMotor()`, controlar o ângulo de rotação dos motores.

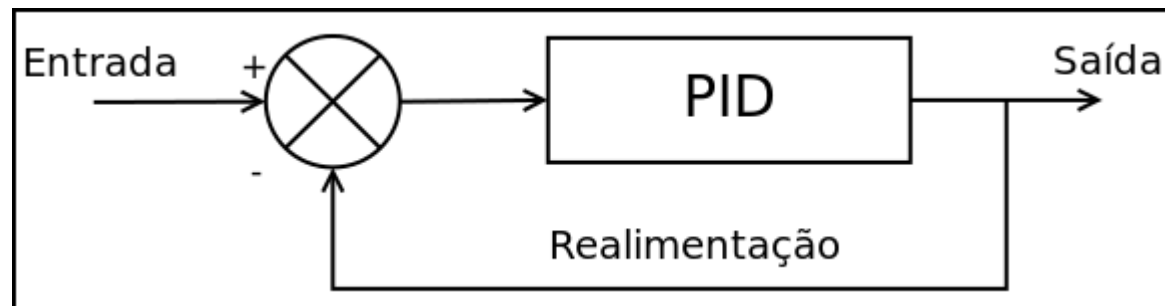
Por exemplo:

```
RotateMotor(OUT_AC,35,-360);
```

Linguagem NXC

- Comandos avançados

O Firmware do NXT implementa um controlador PID em malha fechada para controlar a posição dos motores e suas velocidades usando a informação dos encoders como realimentação (feedback).



Linguagem NXC

- Comandos avançados

Por exemplo, se você quer que seu robô se mova perfeitamente em linha reta, você pode usar um recurso de sincronização que faz com que dois motores selecionados rodem juntos e esperem um pelo outro caso um deles fique lento ou bloqueado.

Existem muitos comandos para liberar todo o potencial dos servo-motores, mas isto será assunto de um outro encontro...

Linguagem NXC

- Apresentação de robô móvel programado com NXC executando tarefa proposta em trabalho de disciplina
- Ver enunciado e código
- Demonstração...

Linguagem NXC

- Oficina Programando Robôs Lego NXT usando NXC

A apresentação de hoje está disponível no link:

http://www.univasf.edu.br/~marcelo.linder/MiniCursoS_cientex2015.pdf

Bibliografia

http://pt.wikipedia.org/wiki/LEGO_Mindstorms_NXT, link visitado em 08 de novembro de 2014.

<http://bricxcc.sourceforge.net/>, link visitado em 08 de novembro de 2014.

Apostila Programming LEGO NXT Robots using NXC (beta 30 or higher)

Autor: Danielle Benedettelli

Revisão: John Hansen

Versão 2.2, 7 de junho de 2007.

Tradução: Rafael Bermudes

Revisão: Felipe Nascimento Martins

NER@ – Núcleo de Estudos em Robótica e Automação –

<http://nera.sr.ifes.edu.br>

IFES – Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo

Edição: Janeiro de 2012.