



System Design Projekt Einführung in NXC

Dorothea Simons, Thomas Lampe,
Maximilian Beinhofer

Albert-Ludwigs-Universität Freiburg

Übersicht

- Ausgabe der Leihverträge
- Einführung in NXC
- Ausgabe der LEGO Kästen

Ausgabe der LEGO Kästen

Nur an Gruppen, die sich per Email an lego@imtek.uni-freiburg.de registriert haben

- Betreff: Anmeldung
- Inhalt: Name, Email und Matrikelnummern aller Mitglieder im Format

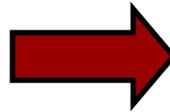
Max Mustermann ; max@mustermann.de ; 1234567890

Was ist NXC?

- Eine C-ähnliche Programmiersprache (Not eXactly C)
- Speziell zur Programmierung des Lego-NXT Microcontrollers

```
#define THRESHOLD 30 //percent

task main()
{
    SetSensorLight(IN_1);
    while(true)
    {
        ClearScreen();
        NumOut(60, LCD_LINE1, Sensor(IN_1));
        OnFwd(OUT_AC, 50);
        while(Sensor(IN_1) > THRESHOLD)
        {}
        Off(OUT_AC);
        OnRev(OUT_C, 100);
        Wait(800);
    }
}
```



Was Sie heute erwartet

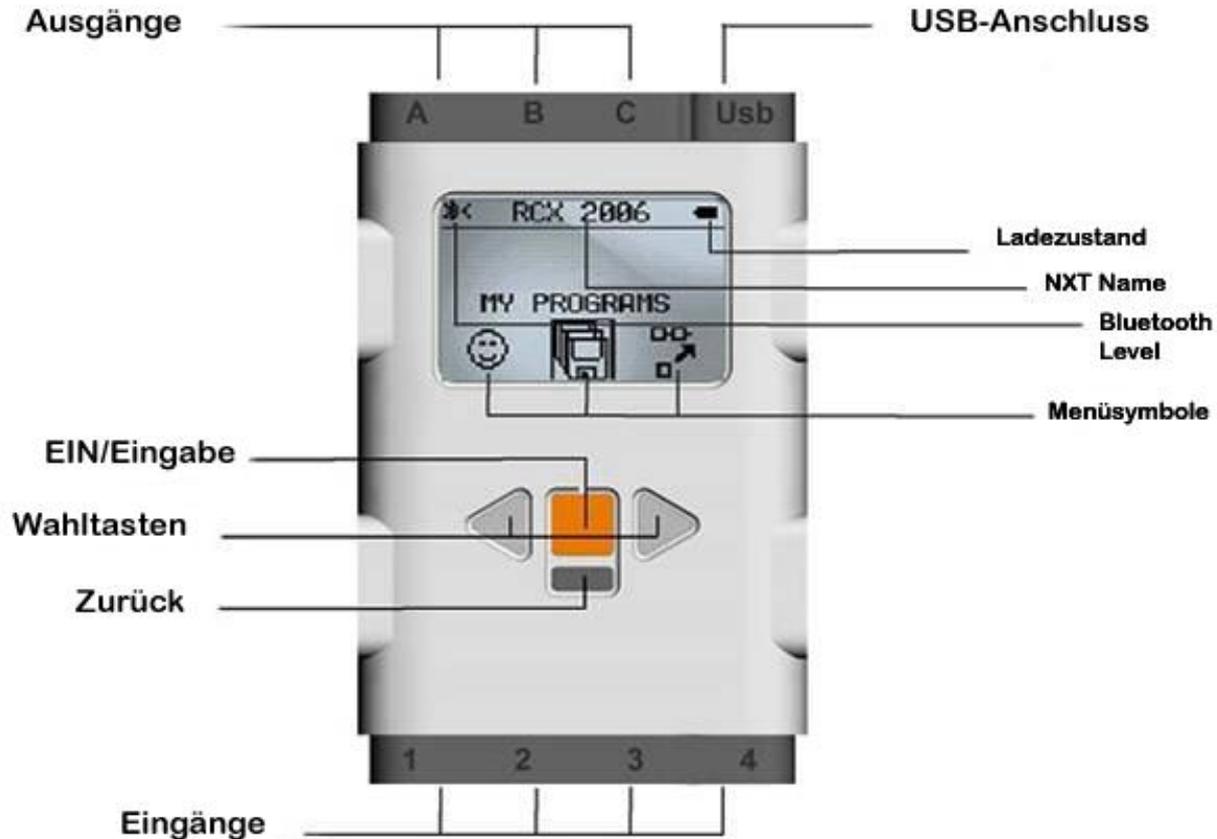
- Einführung in NXC anhand von Beispielen
- Folien mit den Beispielen gibt es unter <http://ml.informatik.uni-freiburg.de/teaching/ws1314/sdp>
- Dies ist kein umfassender Programmierkurs, das meiste müssen Sie sich selbst aneignen.

Minimalbeispiel: Motoren

```
task main()  
{  
    OnFwd(OUT_A, 100);  
    Wait(5000);  
    Off(OUT_A);  
}
```

- **task** main(){}: Hauptfunktion. Hier kommen die Kommandos rein, die ausgeführt werden sollen
- **OnFwd(x,y)**: starte Vorwärtsgang auf Motor x mit y% der Maximalgeschwindigkeit
- **Wait(x)**: warte für x msec
- **Off(x)**: stoppe Motor x

Aufbau des NXT Microcontrollers



Motoren und Sensoren

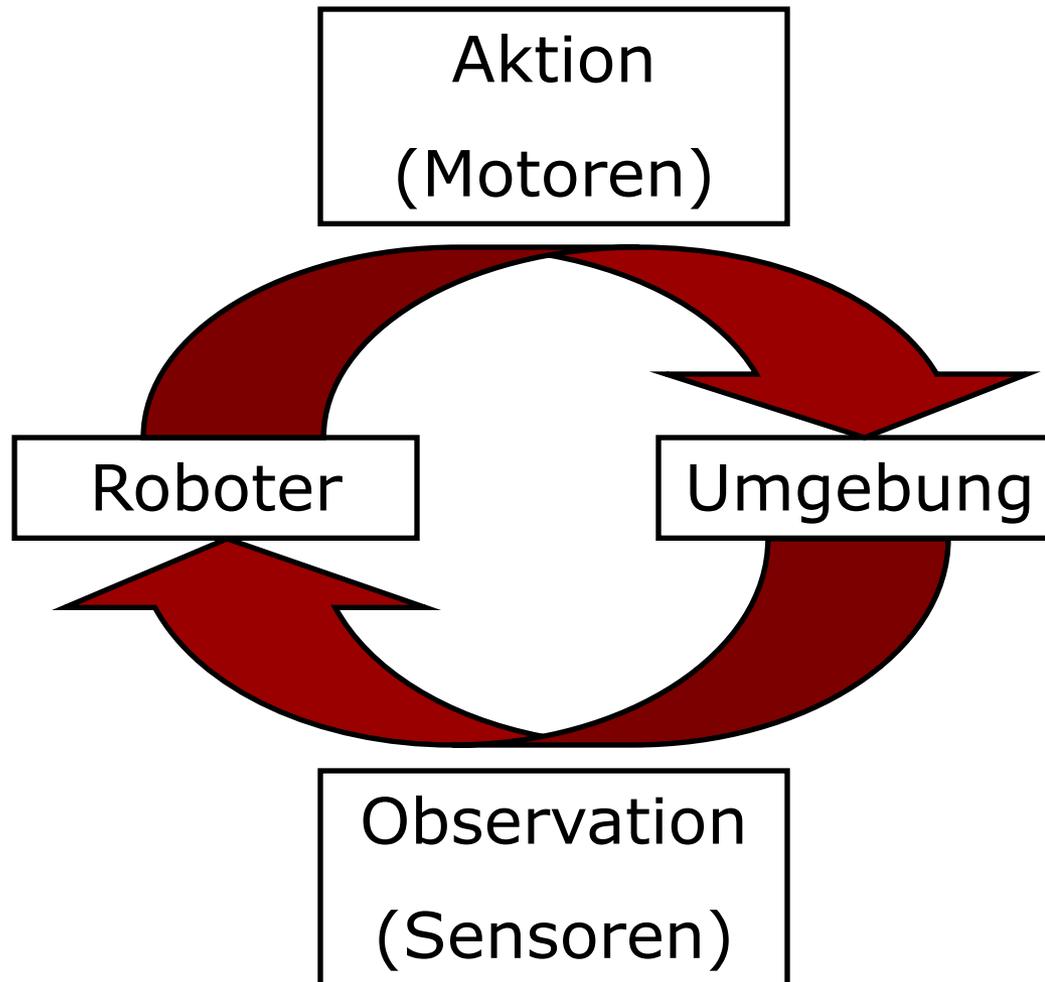


Minimalbeispiel: Sensoren

```
task main()
{
    SetSensorLight (IN_1);
    ClearScreen ();
    NumOut (60, LCD_LINE1, Sensor (IN_1));
}
```

- **SetSensorLight(x)**: definiere x als Lichtsensor
- **ClearScreen()**: lösche die Anzeige des NXT
- **NumOut(x,y,z)**: zeige Nummer z an Position (x,y) auf dem Display des NXT an
- **Sensor(x)**: Funktion, die von Sensor x gemessenen Wert zurückgibt

Funktionszyklus dynamischer Systeme



Beispiel: Aktion/Observation/Reaktion

```
task main()
{
    SetSensorTouch(IN_2);
    OnFwd(OUT_AC, 75);
    until (Sensor(IN_2) == 1);
    Off(OUT_AC);
}
```

- **SetSensorTouch(x)**: definiere x als Berührungssensor
- **until** (Bedingung): warte bis Bedingung erfüllt, fahre dann fort mit nächster Zeile

Zyklus

```
#define THRESHOLD 30 //Wert in Prozent

task main()
{
    SetSensorLight(IN_1);
    while(true)
    {
        ClearScreen();
        NumOut(60, LCD_LINE1, Sensor(IN_1));
        OnFwd(OUT_AC, 50);
        while(Sensor(IN_1) > THRESHOLD)
        {}
        Off(OUT_AC);
        OnRev(OUT_C, 30);
        Wait(2500);
    }
}
```

- **#define x y**: gib y den Namen x
- **//**: Kommentar, Rest der Zeile wird nicht als Programmcode interpretiert
- **while** (Bedingung){Befehle}: während Bedingung gilt, werden Befehle in einer Schleife ausgeführt
- **OnRev(x,y)**: wie **OnFwd**, nur rückwärts

Dasselbe als Statemaschine

```
#define THRESHOLD 30 //Wert in Prozent

#define STATE_INIT 0
#define STATE_SAFE_ON_TABLE 1
#define STATE_DETECTED_EDGE 2

int state=0;

task main()
{
    while(true)
    {
        if (state==STATE_INIT){
            SetSensorLight(IN_1);
            state=STATE_SAFE_ON_TABLE;
        }
        if (state==STATE_SAFE_ON_TABLE){
            OnFwd(OUT_AC,50);
            if (Sensor(IN_1)>THRESHOLD){
                // eigentlich überflüssig
                state=STATE_SAFE_ON_TABLE;
            } else {
                // Sensorwert kleiner oder gleich
                state=STATE_DETECTED_EDGE;
            }
        }
        if (state==STATE_DETECTED_EDGE){
            Off(OUT_AC);
            OnRev(OUT_C,30);
            Wait(2500);
            state=STATE_SAFE_ON_TABLE;
        }
        ClearScreen();
        NumOut(60, LCD_LINE1, Sensor(IN_1));
    }
}
```

- **int** x=y: definiere x als integer-Variable und weise ihr den Wert y zu
- **if** (Bedingung){Befehle} **else** {andere Befehle}: falls Bedingung erfüllt, führe Befehle aus, falls nicht, führe andere Befehle aus
- =: weist linker Seite Wert von rechter Seite zu
- ==: checkt, ob linke und rechte Seite gleich

Statemachine mit Funktionen

```
#define THRESHOLD 30 //Wert in Prozent

#define STATE_INIT 0
#define STATE_SAFE_ON_TABLE 1
#define STATE_DETECTED_EDGE 2

int state=0;

int initialize()
{
    SetSensorLight(IN_1);
    return STATE_SAFE_ON_TABLE;
}

int drive()
{
    OnFwd(OUT_AC, 50);
    if (Sensor(IN_1) <= THRESHOLD) {
        return STATE_DETECTED_EDGE;
    }
    return STATE_SAFE_ON_TABLE;
}

int turn_around()
{
    Off(OUT_AC);
    OnRev(OUT_C, 30);
    Wait(2500);
    return STATE_SAFE_ON_TABLE;
}

void refresh_display()
{
    ClearScreen();
    NumOut(60, LCD_LINE1, Sensor(IN_1));
}
```

```
task main()
{
    while(true)
    {
        if (state==STATE_INIT){
            state=initialize();
        }
        if (state==STATE_SAFE_ON_TABLE){
            state=drive();
        }
        if (state==STATE_DETECTED_EDGE){
            state=turn_around();
        }
        refresh_display();
        // diese Funktion wird wahrscheinlich zu
        //oft ausgeführt ! (Performance !)
    }
}
```

- **int** Funktionsname (Input) {Befehle}: Funktion, die Befehle ausführt und ein int als Rückgabewert (**return**) hat
- **void**: Kennwort für Funktionen ohne Rückgabewert

Sensoren auslesen!

- Grenzwerte für Sensoren lassen sich auch durch die im NXT integrierten LEGO Sensor-Tests finden
- Das geht mit allen verfügbaren Sensoren:
 - Taster (Drucksensoren)
 - Lichtsensoren
 - Ultraschall zur Abstandsmessung
 - Mikrofon
- Möglichst versuchen gefundene Werte nicht im gesamten Programm zu „verteilen“ !
(z.B. Über Defines festlegen)

Programmierumgebungen

- Unter Windows: **bricxcc**
- Unter Linux: Texteditor + **NBC**
- Unter MacOS: Texteditor + **NBC** +evtl.
NeXT Tools

bricxcc

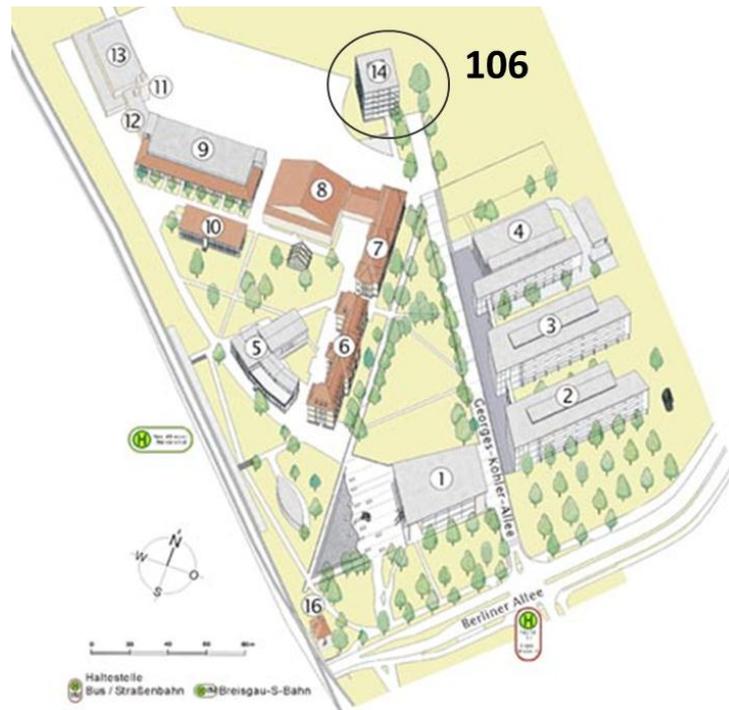
- Nützliche Programmierumgebung unter Windows für NXC
- Integrierter Texteditor und Compiler
- Kann per USB Programme auf den NXT übertragen
- Bluetooth ist möglich, wurde von uns nicht ausprobiert

Nützliche Links:

- Zu NXC an sich:
 - Hier vorgestellte Befehle reichen NICHT, Rest im Selbststudium mit
 - http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC_tutorial.pdf
 - http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC_Guide.pdf
- USB-Treiber für NXT (Driver 1.02)
 - <http://drivers.softpedia.com/get/Other-DRIVERS-TOOLS/Others/LEGO-Mindstorms-NXT-Fantom-Driver-102.shtml> oder als Teil der kompletten LEGO .EV3 Software unter
 - <http://www.lego.com/de-de/mindstorms/downloads/software/ddsoftwaredownload/>
- Download **bricxcc**:
 - <http://sourceforge.net/projects/bricxcc/files/bricxcc/>
(nicht auf download, sondern auf bricxcc 3.3.8.9 klicken)
- Nützliches zu NXC unter Linux:
 - <http://www.eggwall.com/2011/07/setting-up-lego-programming-environment.html>

Betreuungstermine

- Vorerst in Gebäude 106, Raum 01-007
- Voraussichtlich ab 2.12.2013 in 106, UG



Betreuungstermine

- Vorerst in Gebäude 106, Raum 01-007
- Voraussichtlich ab 2.12.2013 in 106, UG

- Termine:
 - Di, 10:30-12:00 Florian Wolling
 - Di, 12:30-14:00 Bernhard Link
 - Mi, 16:00-17:30 Johan Vertens
 - Do, 14:30-16:00 Hanna Kuhn
 - Fr, 12:30-14:00 Kerstin Thiemann

- Betreuung startet am 5.11.2013

Zwischenbericht

- Fällig bis spätestens 13.12.2013
- Abgabe per Ticketsystem ab 06.12.2013 (Erinnerungsmail)
- Ticket zum Hochladen wird in der Erinnerungsmail stehen
- 2-3 Seiten als pdf, Layout beliebig

Kasteninhalt

- Bitte Elektronikteile im Kasten auf Vollständigkeit prüfen:
 - 1 NXT, 1 Akku, 1 Ladegerät
 - 3 Motoren
 - 3 Lichtsensoren
 - 2 Berührungssensoren
 - 1 Ultraschallsensor
 - 1 Mikrofon
- Sonstige Legoteile und Kabel können zu den Betreuungszeiten ergänzt werden

Kästenausgabe

- Kastenummer = Gruppennummer
= Startnummer beim Wettbewerb
- Kästen werden in zufälliger Reihenfolge ausgegeben
- Lagerung der Kästen in Eigenverantwortung, Schließfächer gibt es im Keller von Geb. 102
- Unvollständige/unregistrierte Gruppen bitte warten

Kästenausgabe

- Ausgabe nur gegen Vorlage aller **Studentenausweise** und des vollständig unterschriebenen **Leihvertrags**
- Bitte nur **eine** Person pro Gruppe zur Kästenausgabe kommen
- Diese sollte die Emailadresse der Gruppenanmeldung kennen