

## Lego-Mindstorms

### Linien folgen

Einer Linie zu folgen stellt eine interessante, relativ einfache Möglichkeit für einen Roboter dar, sich entlang eines definierten Weges fortzubewegen. Mit einer auf den Boden gemalten Linie kann man ganze Straßen für Roboter aufbauen (wie bei dem Ausschnitt aus *Wall-E* rechts). Und eigentlich ist das auch kein all zu schwer zu lösendes Problem.



*Wall-E*, Pixar 2008

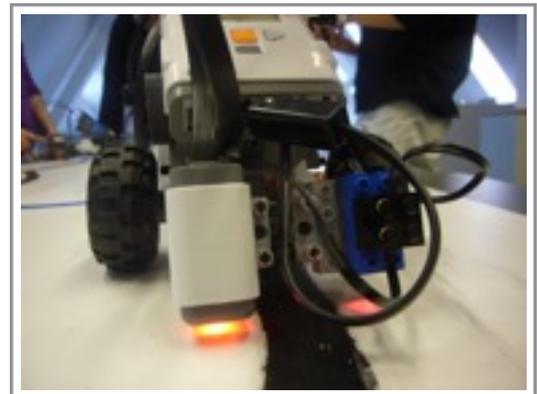
Benötigt wird im Grunde nur ein Lichtsensor, der auf den Boden zeigt. Noch viel einfacher geht es mit zwei Lichtsensoren, wie wir im folgenden zeigen.

Zum Anfang des Programms bestimmt der Roboter die Helligkeitswerte der (weißen) Umgebung. Nun wird eine gewisse Fehlertoleranz dadurch erreicht, dass von diesem Wert 10% heruntergerechnet werden. So schlägt der Roboter nicht bei jedem minimalen Messfehler aus.

```
lightLeft = Sensor(IN_1) * 90 / 100;
```

Der Hauptteil unseres Programms führt dann folgende Aktionen in einer Endlosschleife durch:

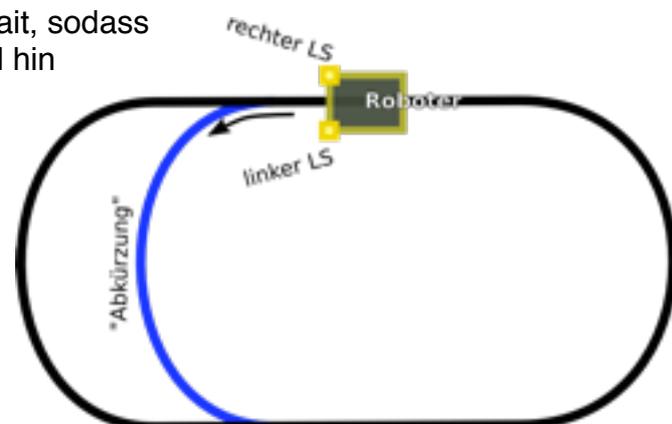
1. Generell geradeaus fahren
2. Wenn linker Lichtsensor die Linie sieht, etwas nach links drehen
3. Wenn rechter Lichtsensor die Linie sieht, etwas nach rechts drehen



Unser NXT-Roboter

Feintuning kann man dann noch erreichen, indem man den Abstand der Lichtsensoren optimiert und vor allem die Geschwindigkeiten zum Drehen möglichst gut einstellt. Wir haben damit gute Ergebnisse erzielt, dass sich zum Drehen ein Motor etwas langsamer vorwärts bewegt und der andere, noch langsamer, rückwärts. Abgerundet wird die Bewegung mit einem kurzen Wait, sodass der Roboter auf geraden Strecken nicht allzu schnell hin und her „zappelt“ und so Tempo verliert.

Ein Problem ergibt sich allerdings, wenn es mehrere mögliche Wege gibt (z.B. Abkürzungen): Dadurch, dass zuerst der linke und dann der rechte Lichtsensor überprüft wird, wird in jedem Fall der linke Weg einer Abzweigung bevorzugt. Im Bild rechts, **gegen den Uhrzeigersinn**, funktioniert das unten stehende Programm, anders herum nicht.



Ein Workaround besteht darin, das vorliegende Programm einfach **nur** für die Fahrrichtung gegen den Uhrzeigersinn zu verwenden, und für die entgegengesetzte Fahrrichtung die beiden unteren **IF-Abfragen zu tauschen**. Dann wird der rechte Sensor bevorzugt.

Somit haben wir zwei kompilierte Programm auf dem NXT – keine all zu schöne Lösung, jedoch wie wir finden eine sehr einfache.

Es folgt der komplette, kommentierte Sourcecode in NXC.

```
/* Informatik-Profilkurs Paul-Natorp-Schule
   Paul Grau / Michael Decker / Jan Corsten
   19-03-2009

   Aufgabe
   Roboter folgt einer dunklen Linie

   Ports
   OUT_A ~ /
   OUT_B ~ Motor 1
   OUT_C ~ Motor 2
   IN_1 ~ Lichtsensor (links)
   IN_2 ~ /
   IN_3 ~ Lichtsensor (rechts)
   IN_4 ~ /
*/
task main() {
    int lightLeft,lightRight,speed;

    // Sensoren initialisieren
    SetSensorLight(IN_1); // links NXT-Sensor
    SetSensorType (IN_3, SENSOR_TYPE_LIGHT); // rechts RCX-Sensor

    // geschwindigkeit einstellen
    speed = 100;

    // Roboter bittet um Bestätigung der korrekten Startposition
    PlayToneEx(262,200,3,0);
    TextOut(0,LCD_LINE2,"Position ok?");
    TextOut(0,LCD_LINE6,"links"); TextOut(0,LCD_LINE7,"rechts");
    ButtonCount(BTNCENTER, true); // auf Drücken der mittleren, orangenen Taste warten
    while(!ButtonCount(BTNCENTER, true)) {
        NumOut(40,LCD_LINE6,Sensor(IN_1));
        NumOut(40,LCD_LINE7,Sensor(IN_3));
    }

    // Speichern der hellen/weißen Farbe
    // und davon 10% runterrechnen, zur Fehlertoleranz beim späteren Messen
    lightLeft = Sensor(IN_1) * 90 / 100;
    lightRight = Sensor(IN_3) * 90 / 100;

    // gespeicherte Sensorwerte anzeigen
    NumOut(40,LCD_LINE6,lightLeft); NumOut(40,LCD_LINE7,lightRight);

    while(1) { // Endlosschleife
        // aktuelle Sensorwerte anzeigen
        NumOut(60,LCD_LINE6,Sensor(IN_1));
        NumOut(60,LCD_LINE7,Sensor(IN_3));

        // generell vorwärts fahren
        OnFwdReg(OUT_BC,speed,1);

        // wenn linker LS dunkler geworden ...
        if ( Sensor(IN_1) < lightLeft ) {
            OnFwd(OUT_C,speed/2); // C etwas langsamer vor, B langsamer zurück
            OnFwd(OUT_B,-speed/4);
        }
        // wenn rechter LS dunkler geworden ...
        if ( Sensor(IN_3) < lightRight ) {
            OnFwd(OUT_B,speed/2); // B etwas langsamer vor, C langsamer zurück
            OnFwd(OUT_C,-speed/4);
        }

        Wait(5);
        Off(OUT_BC);
    } // end while(1)
}
```