

# Roboter, Labyrinth und Lego

## – 2. Teil: Programmierung der Roboter –

### Lego Mindstorms Schulprojekt

### der Technischen Universität Dänemark

JENS STARKE

Technische Universität Dänemark  
Institut für Mathematik

Januar 2008

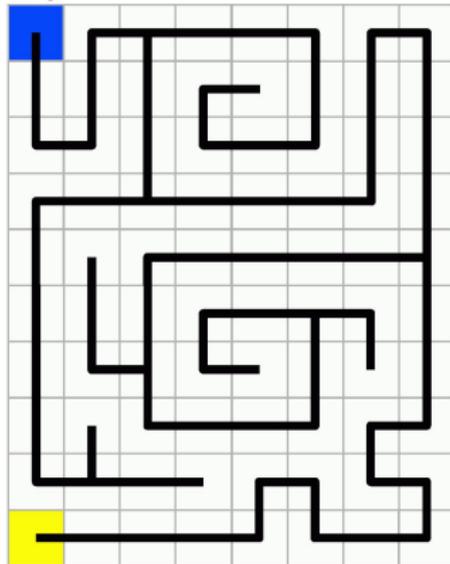
# Part I

## Der Labyrinth-Wettbewerb

– mit Lego Mindstorms Roboter –

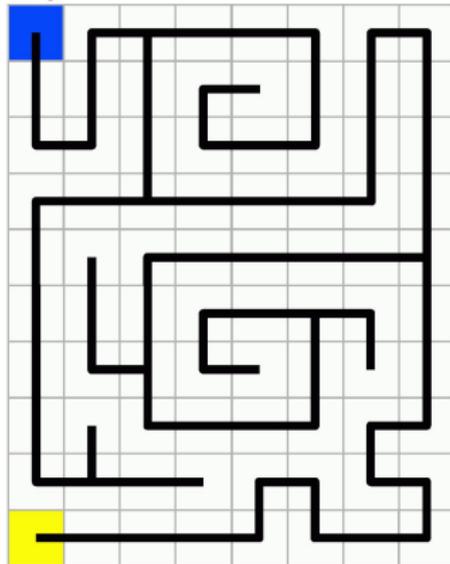


Ziel

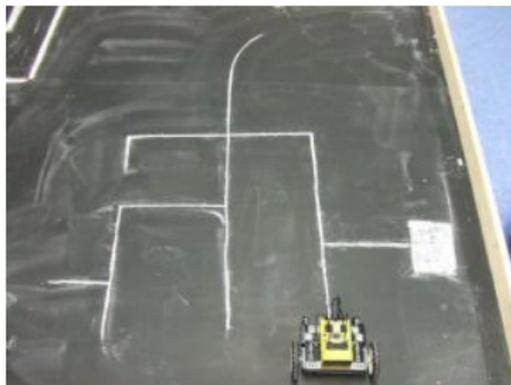


Start

Ziel



Start





1. Ziel ist es, in möglichst kurzer Zeit ein unbekanntes Labyrinth mit einem Roboter zu durchfahren.
2. Es ist nur erlaubt, sich auf den gezeichneten Wegen zu bewegen.
3. Das Labyrinth soll zuerst von einer Start- zu einer Zielposition durchfahren werden und anschließend von der Zielposition zurück zur Startposition.

~> **Das Team mit der kürzesten Zeit gewinnt!**



- ▶ Jedes Team darf aus maximal 2 Kindern bestehen.
- ▶ Jedes Team hat 2 Versuche, die kürzeste Zeit beider Versuche wird gewertet.
- ▶ Sollte ein Roboter den vorgegebenen Weg verlassen, so wird er vom Schiedsrichter an die Stelle zurückgesetzt, wo der Weg verlassen wurde. Das Team erhält eine Strafzeit von 15 Sekunden.
- ▶ Sollte ein Roboter einen Streckenabschnitt nach 4 Versuchen nicht bewältigen können, so wird er vom Schiedsrichter ohne weitere Zeitstrafe auf die nächste Position nach dem problematischen Streckenabschnitt gesetzt.



## Steuercomputer mit Sensoren und Aktoren

Roboter,  
Labyrinth  
und Lego

JENS STARKE

Der  
Labyrinth-  
Wettbewerb

Lego  
Mindstorms  
Roboter

Eure  
Aufgaben

Wie  
programmiert  
man einen  
Roboter?

Die  
Arbeitsab-  
schnitte



## Übersicht

Übersicht



Batterien einsetzen



Anschließen des NXT



Der NXT-Baustein



NXT-Hauptmenü



Berührungssensor



Geräuschsensor



Lichtsensoren



Ultraschallsensoren



Interaktiver Servomotor



Lampen



Bluetooth einsetzen



## Programmierung und Steuerung

Software installieren



Dein erstes Programm



Benutzeroberfläche



Robot Educator



Programmierpalette



Konfiguration



Der Controller



Notwendige Vorgehensweise:

Planen, Durchführen



und ständig Verbessern!!!

1. Roboter zusammenbauen
2. Testen einfacher Befehle (mit Sensoren und Aktoren)
3. Verfolgen einer Linie (mit Kreide auf einer Tafel gemalt)
4. Erkennen von Kreuzungen
5. Suchalgorithmen in einem Labyrinth programmieren
6. Wege merken
7. Finde den kürzesten Weg zurück



Planen und Testen!!!

# Wie programmiert man einen Roboter?

Zuerstmal ausgiebig nachdenken ...



Roboter,  
Labyrinth  
und Lego

JENS STARKE

Der  
Labyrinth-  
Wettbewerb

Lego  
Mindstorms  
Roboter

Eure  
Aufgaben

Wie  
programmiert  
man einen  
Roboter?

Die  
Arbeitsab-  
schnitte

# Wie programmiert man einen Roboter?

Roboter,  
Labyrinth  
und Lego

JENS STARKE

Der  
Labyrinth-  
Wettbewerb

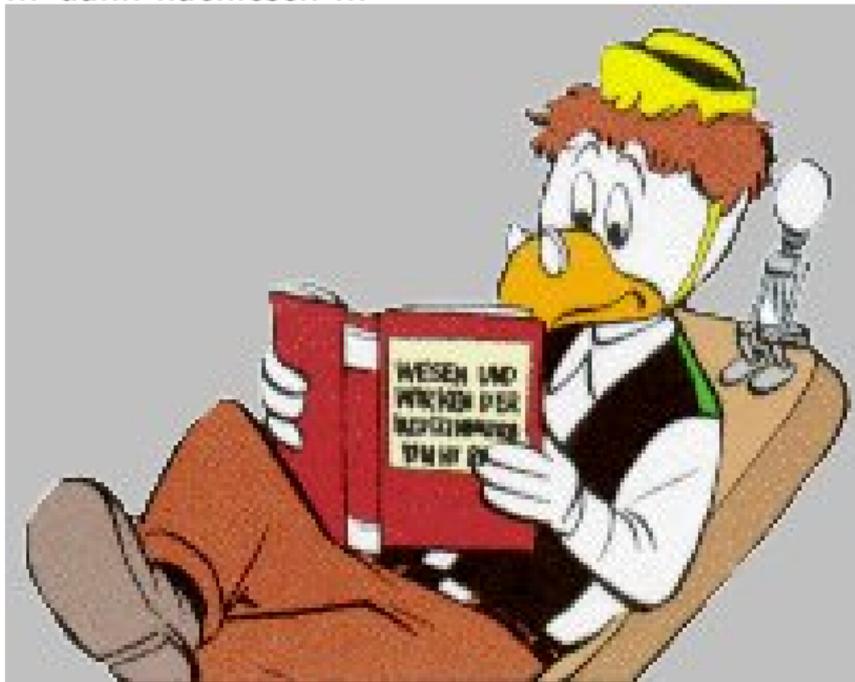
Lego  
Mindstorms  
Roboter

Eure  
Aufgaben

Wie  
programmiert  
man einen  
Roboter?

Die  
Arbeitsab-  
schnitte

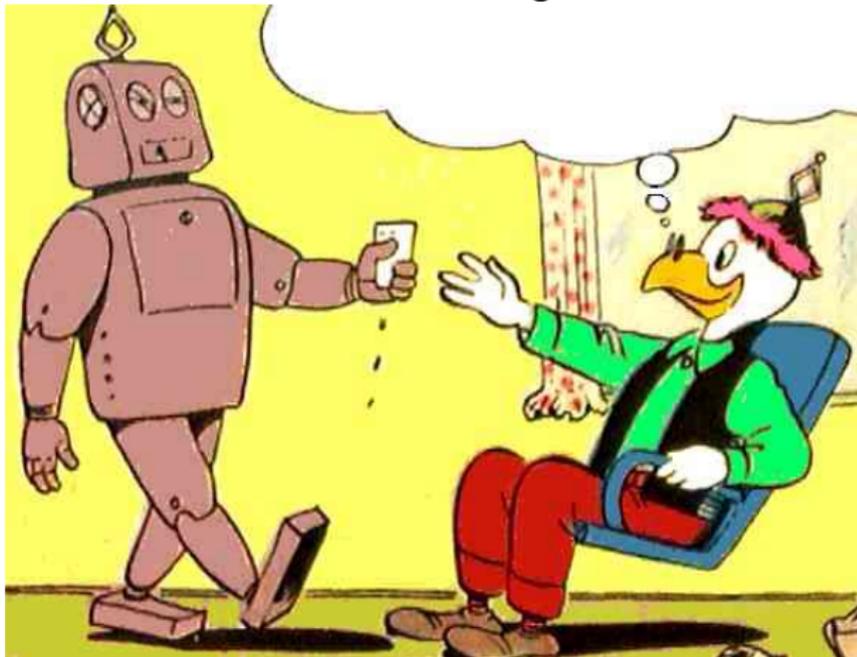
... dann nachlesen ...





... bauen ...

... und schließlich die Vorteile genießen!



DTU



**Roboter,  
Labyrinth  
und Lego**

JENS STARKE

Der  
Labyrinth-  
Wettbewerb

Lego  
Mindstorms  
Roboter

Eure  
Aufgaben

Wie  
programmiert  
man einen  
Roboter?

Die  
Arbeitsab-  
schnitte

1. Bauen der Roboter
2. Einfacher Befehle & Programme
3. Nachfahren einer Linie
4. Erkennen von Kreuzungen
5. Suchalgorithmen
6. Wo befinde ich mich?
7. Wege merken
8. Kürzeste Wege

## Part II

### Die Arbeitsabschnitte

– zur Vorbereitung auf den Labyrinth-Wettbewerb –

# 1. Arbeitsabschnitt: Zusammenbauen der Roboter

Roboter,  
Labyrinth  
und Lego

JENS STARKE

1. Bauen der Roboter

2. Einfacher Befehle & Programme

3. Nachfahren einer Linie

4. Erkennen von Kreuzungen

5. Suchalgorithmen

6. Wo befinde ich mich?

7. Wege merken

8. Kürzeste Wege



ist erledigt



1. Bauen der  
Roboter

2. Einfacher  
Befehle &  
Programme

3. Nachfahren  
einer Linie

4. Erkennen  
von  
Kreuzungen

5. Suchalgo-  
rithmen

6. Wo befinde  
ich mich?

7. Wege  
merken

8. Kürzeste  
Wege

## 2. Arbeitsabschnitt: Testen einfacher Befehle und Schreiben erster Programme



ist zum Teil erledigt



wird aber wiederholt



## 2. Arbeitsabschnitt: Aufgaben

**Aufgabe 2.1:** Programmiere den Roboter, so daß er 100 cm geradeaus fährt.

**Aufgabe 2.2:** Programmiere den Roboter, so daß er ein Quadrat mit Kantenlänge 100 cm abfährt, dann umdreht und das Quadrat rückwärts abfährt.

**Aufgabe 2.3:** Ändere das Programm aus Aufgabe 1, so daß der Roboter stoppt und piept wenn ein Hindernis im Weg ist.



**Aufgabe 2.4:** Programmiere den Roboter, so daß er ständig geradeaus fährt. Falls ein Hindernis im Weg ist, soll er anhalten und piepen. Wenn die Farbe des Bodens heller als ein vorgegebener Wert wird, soll der Roboter ebenfalls anhalten und applaudieren.

**Aufgabe 2.5:** Lese den Helligkeitswert der aktuellen Bodenfarbe ein und ändere das Programm aus Aufgabe 4, so daß es auch funktioniert, wenn die Farbe des Bodens bedeutend heller oder dunkler als der am Anfang eingelesene Wert wird.

### 3. Arbeitsabschnitt: Erkennen und Nachfahren einer Linie

Roboter,  
Labyrinth  
und Lego

JENS STARKE

1. Bauen der Roboter
2. Einfacher Befehle & Programme
3. Nachfahren einer Linie
4. Erkennen von Kreuzungen
5. Suchalgorithmen
6. Wo befinde ich mich?
7. Wege merken
8. Kürzeste Wege



Beispielprogramm ist vorhanden



muss aber verstanden und selbst programmiert werden



**Aufgabe 3.1:** Untersuche und verstehe das Beispielprogramm zum Erkennen und Nachfahren einer Linie.

**Aufgabe 3.2:** Erstelle selbst ein Programm, so daß der Roboter auch einem Kreidestrich auf einer Tafel nachfahren kann.

**Aufgabe 3.3:** Überlege Dir eine Strategie die dem Roboter ermöglicht bei einmal verlorener Linie trotzdem weiterzukommen. Experimentiere mit einer Linie mit kleinen Lücken und lasse den Roboter ein Warnsignal ausgeben, sobald er auf eine Lücke trifft.

## 4. Arbeitsabschnitt: Erkennen von Kreuzungen

- ▶ Notwendig: Untersuchung der Umgebung einer Linie
- ▶ mögliche Kreuzungen:



- ▶ Bewegung des Lichtsensors über dem Boden durch
  - a) Fahr-Manöver
  - b) Lichtsensor wird durch einen der Aktoren bewegt

1. Bauen der Roboter
2. Einfacher Befehle & Programme
3. Nachfahren einer Linie
4. Erkennen von Kreuzungen
5. Suchalgorithmen
6. Wo befinde ich mich?
7. Wege merken
8. Kürzeste Wege





## 4. Arbeitsabschnitt: Aufgaben

**Aufgabe 4.1:** Entscheide Dich, wie Du den Lichtsensor über dem Boden bewegst. Überlege Dir eine mögliche Strategie um den Sensor zu bewegen (durch Fahr-Manöver oder geeignete Bewegung eines Aktors). Baue Deinen Lego-Mindstorms entsprechend um.

**Aufgabe 4.2:** Verfolge erneut eine Linie, falls Du Deinen Roboter umgebaut hast.

**Aufgabe 4.3:** Experimentiere mit T-förmigen Straßen. Untersuche die Umgebung einer (Straßen-)Linie und fahre beginnend von einem Endpunkt erst eine mögliche Route nach, dann wieder zurück und anschließend die andere Route nach.

**Aufgabe 4.4:** Experimentiere mit kreuzförmigen Straßen und wiederhole damit Aufgabe 3.

## 5. Arbeitsabschnitt: Suchalgorithmen in einem Labyrinth

▶ **einfache Strategie:**

Bei einer Kreuzung immer rechts abbiegen, bei Sackgassen wieder zurückfahren.

▶ **aber:**

Problem bei geschlossenen Wegen im Labyrinth (im Kreis Laufen)

▶ **zukünftige Strategie:**

Der Roboter soll sich merken wo er schon war, damit er nicht im Kreis läuft.



1. Bauen der  
Roboter

2. Einfacher  
Befehle &  
Programme

3. Nachfahren  
einer Linie

4. Erkennen  
von  
Kreuzungen

5. Suchalgo-  
rithmen

6. Wo befinde  
ich mich?

7. Wege  
merken

8. Kürzeste  
Wege

**Aufgabe 5.1:** Programmiere eine einfache Strategie zur Durchsuchung eines Labyrinthes ohne geschlossene Wege. Kombiniere dies mit den bisherigen Programmen zur Linienverfolgung und Kreuzungserkennung.

## 6. Arbeitsabschnitt: Wo befinde ich mich?

### Weltkoordinaten, Odometrie und Koppelnavigation

#### **Weltkoordinaten (für die Ebene):**

Geben die Position des Roboters mit Hilfe eines Gitters und eines Bezugspunktes an. Die Nummer der waagrechten und senkrechten Gitterlinie gibt die Position an.

#### **Koppelnavigation (Dead Reckoning):**

Laufende Ortsbestimmung eines bewegten Objektes durch Messen der Bewegungsrichtung, Geschwindigkeit und der Zeit.

#### **Odometrie:**

Bestimmung der Geschwindigkeit und Richtung aus der Bewegungs- und Orientierungsmessung der Räder eines Fahrzeugs.



1. Bauen der Roboter
2. Einfacher Befehle & Programme
3. Nachfahren einer Linie
4. Erkennen von Kreuzungen
5. Suchalgorithmen
- 6. Wo befinde ich mich?**
7. Wege merken
8. Kürzeste Wege

**Aufgabe 6.1:** Bestimme während der Fahrt entlang einer Linie die aktuelle Position Deines Roboters und gebe die Weltkoordinaten auf dem Display aus.

# 7. Arbeitsabschnitt: Merken des zurückgelegten Weges

- ▶ Merken der Positionen (Weltkoordinaten) der Abzweigungen und der Richtungen die auf die Abzweigungen führten.
- ▶ Auf dem Rückweg werden an den Abzweigungen die entgegengesetzten Richtungen gefahren.



1. Bauen der Roboter
2. Einfacher Befehle & Programme
3. Nachfahren einer Linie
4. Erkennen von Kreuzungen
5. Suchalgorithmen
6. Wo befinde ich mich?
7. Wege merken
8. Kürzeste Wege

**Aufgabe 7.1:** Fahre durch ein Labyrinth und wähle dabei bei Abzweigungen zufällig die Route, speichere diese und fahre sie wieder zurück.

## 8. Arbeitsabschnitt: Finde den kürzesten Weg zurück

▶ **einfache Strategie:**

Merke Dir den zurückgelegten Weg und die Sackgassen und fahre auf dem Rückweg nicht in die Sackgassen.

~> Dies spart Zeit beim Labyrinth-Wettbewerb!!!

▶ **fortgeschrittene Strategie:**

Bei einem komplizierten Straßennetz kann z.B. mit dem Floyd-Algorithmus der kürzeste Weg von einem Ort zu einem anderen bestimmt werden.

(Dies ist hier aber nicht notwendig)

1. Bauen der  
Roboter

2. Einfacher  
Befehle &  
Programme

3. Nachfahren  
einer Linie

4. Erkennen  
von  
Kreuzungen

5. Suchalgo-  
rithmen

6. Wo befinde  
ich mich?

7. Wege  
merken

8. Kürzeste  
Wege

# Viel Spaß und Erfolg beim Labyrinth-Wettbewerb!!!

Das Datum wird noch bekanntgegeben.