

Softwarepraktikum Teamrobotik SS 2007

Abschlusspräsentation

**von Norman Siemer, Thomas Poltrock
und Martin Schemmer**

Aufgabenstellung:

Kollisionsfreie Navigation durch einen Parcours von variabel positionierbaren Hindernissen. Dabei soll ein Signal gebendes Objekt identifiziert, angesteuert und aus dem Parcours zu einem Zielpunkt befördert werden.

Motivation:

- Verknüpfung von Technik und Kreativität
- Interesse an autonomen Systemen
- sichtbarer Fortschritt

1. Auswahl der Programmierumgebung

2. Qualitätsprüfung der Sensoren und Motoren

- Lichtsensor
- Ultraschall
- Akustiksensoren
- Motor
- Akku
- Messdaten loggen und visualisieren

3. Lösung der Aufgabe

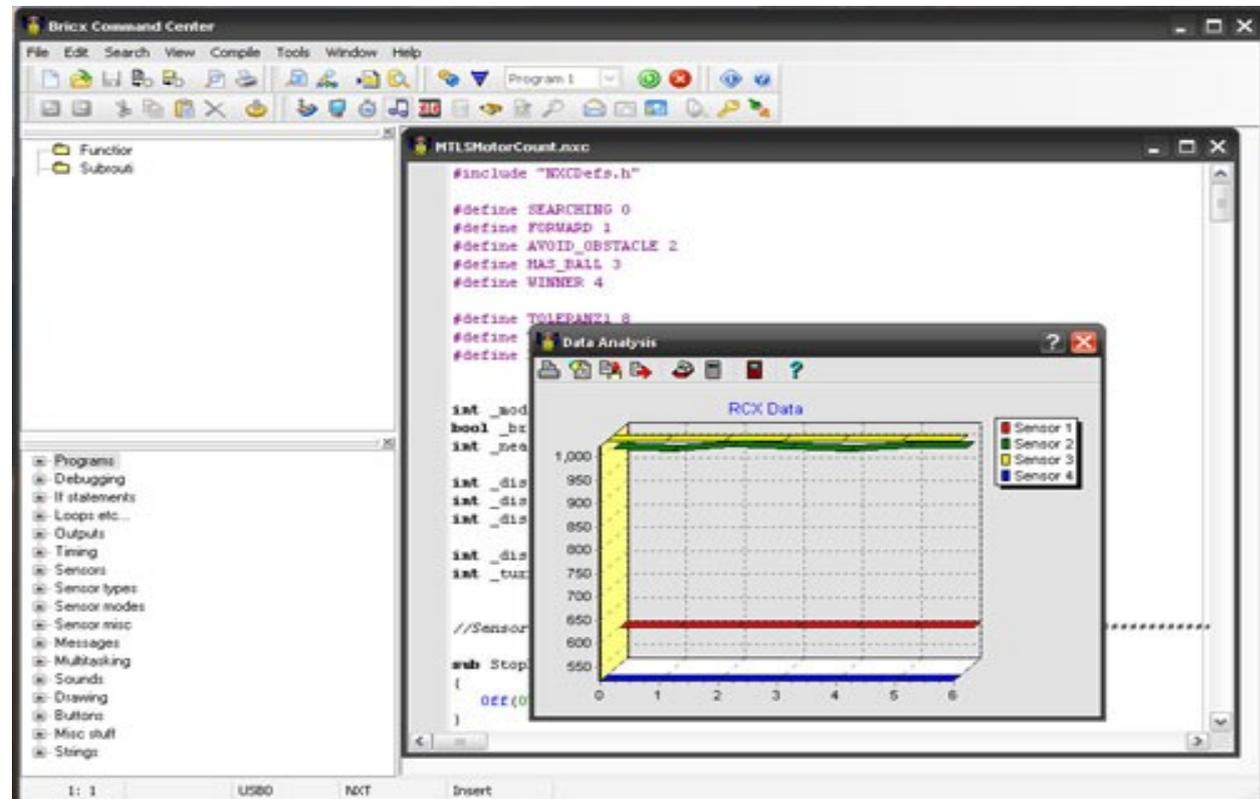
1. Roboter-Konstruktion
2. Ball-Detektion und -Suche
3. Ball einfangen
4. Ball zum Zielpunkt befördern
5. Hindernisse

4. Zusammenfassung

Auswahl der Programmierumgebung

Bricx Command Center

- NXC als Programmiersprache
→ an C angelehnt, leicht verständlich
- Freeware
- gute Dokumentation
- viele Beispielprogramme
- viele eingebaute Tools
- leicht zu bedienen
- ausgereift

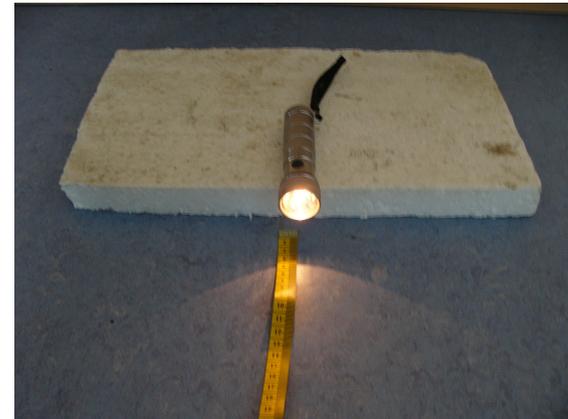
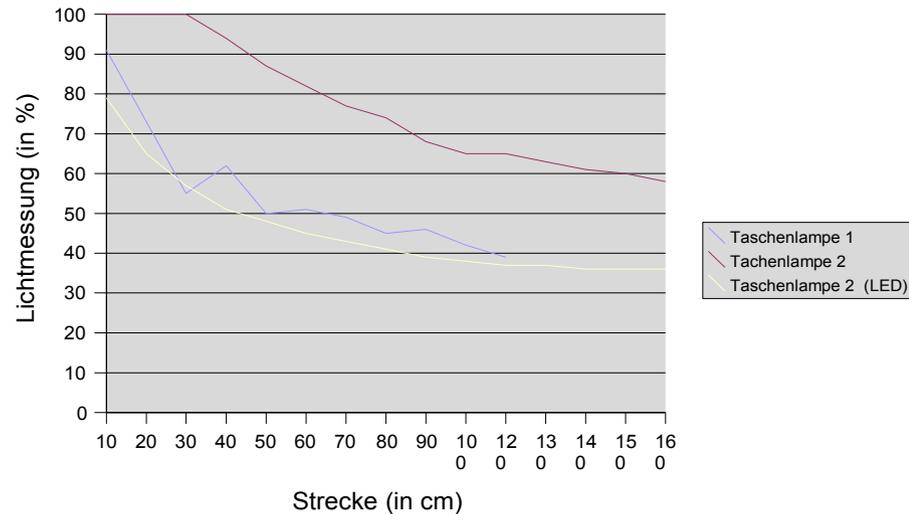


- Lichtsensor
- Ultraschall
- Akustiksensoren
- Motor
- Akku
- Messdaten loggen und visualisieren

Lichttypen

- Ambient
misst Umgebungslicht in %
- Reflected
beleuchtet einen Gegenstand mit rotem Licht und misst das reflektierte Licht in %

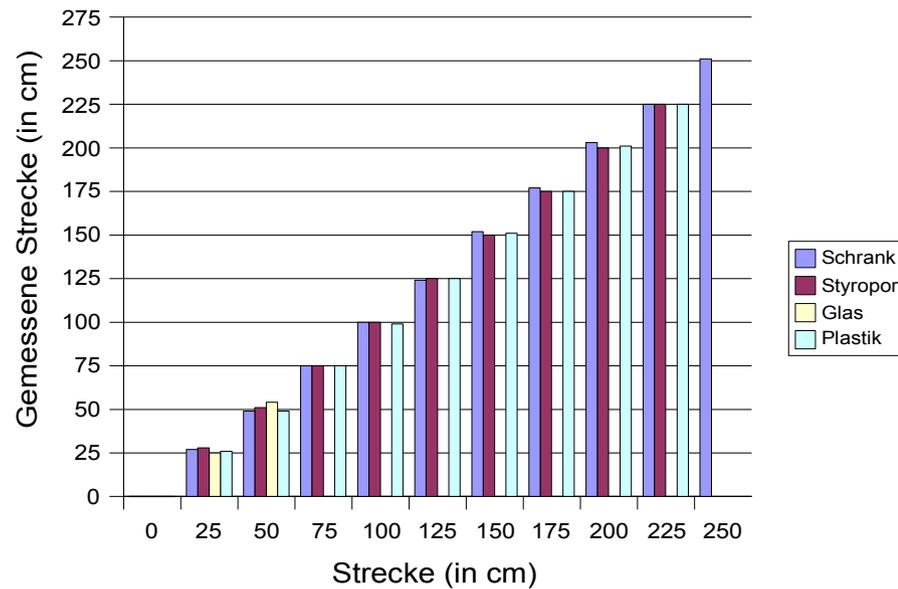
Messung – 3 verschiedene Taschenlampen



Probleme:

- Umgebungslicht ändert sich ständig → keine konstanten Messbedingungen

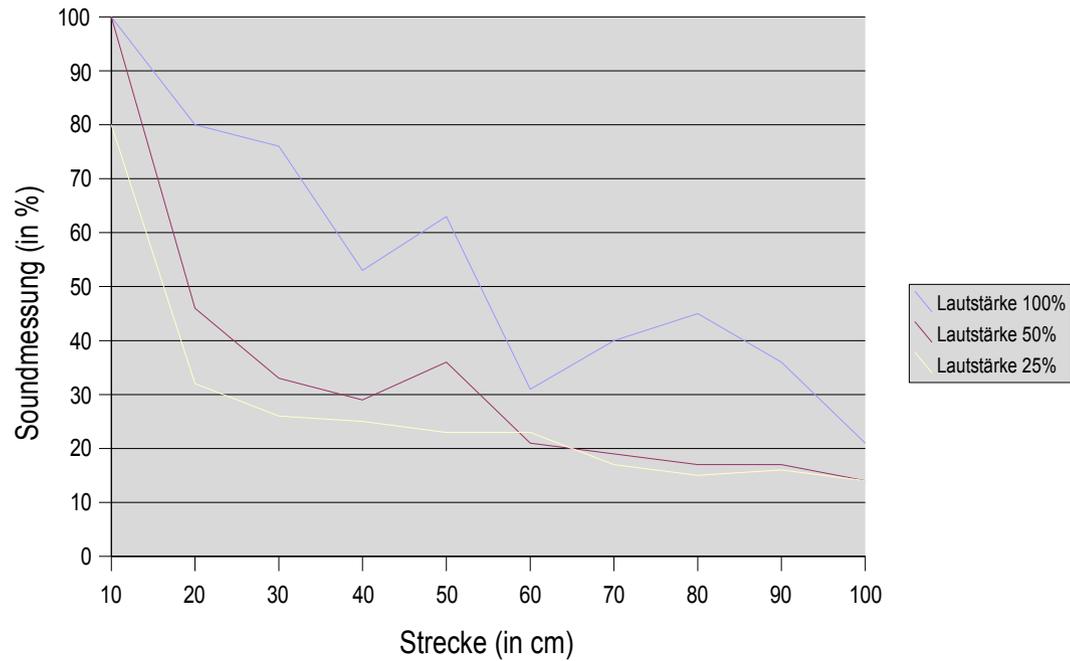
Messung – 4 verschiedene Materialien



Probleme:

- Messung erfolgt nur, wenn Ultraschall in einem bestimmten Winkel auf Gegenstand trifft
- unter 5 cm und über 250 cm keine Messung möglich

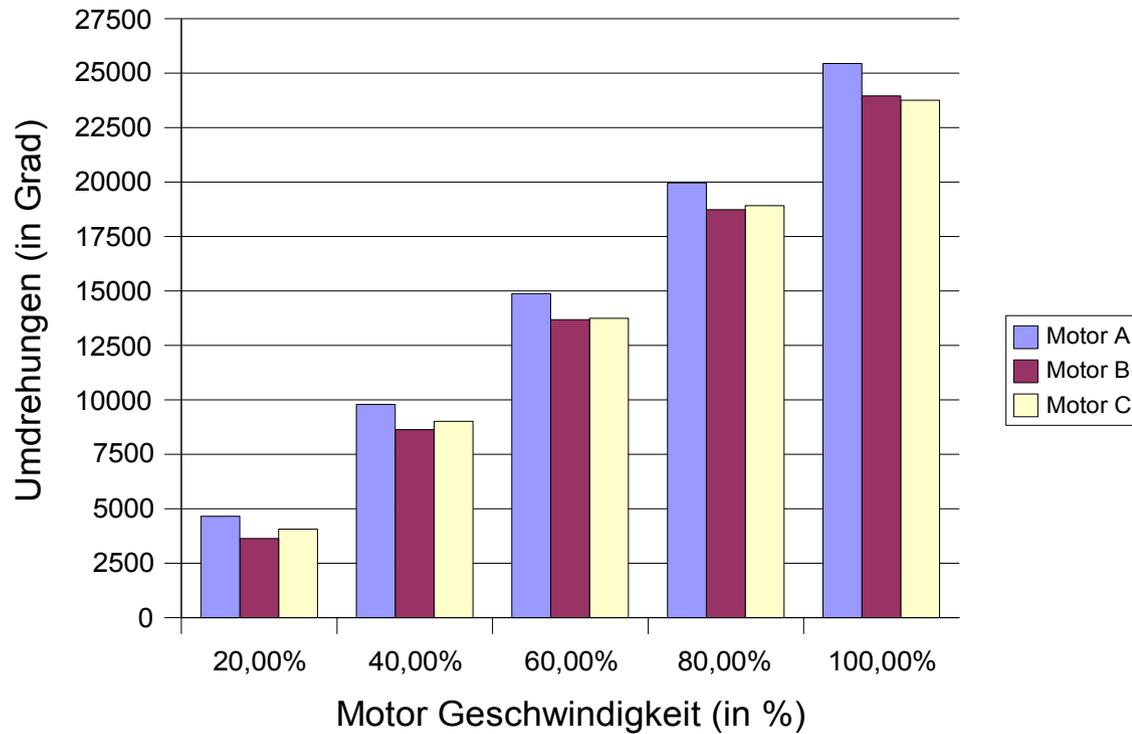
Messung – 3 verschiedene Lautstärken



Probleme:

- keine konstanten Bedingungen möglich
- schwankende Messwerte → genaues Messen schwierig

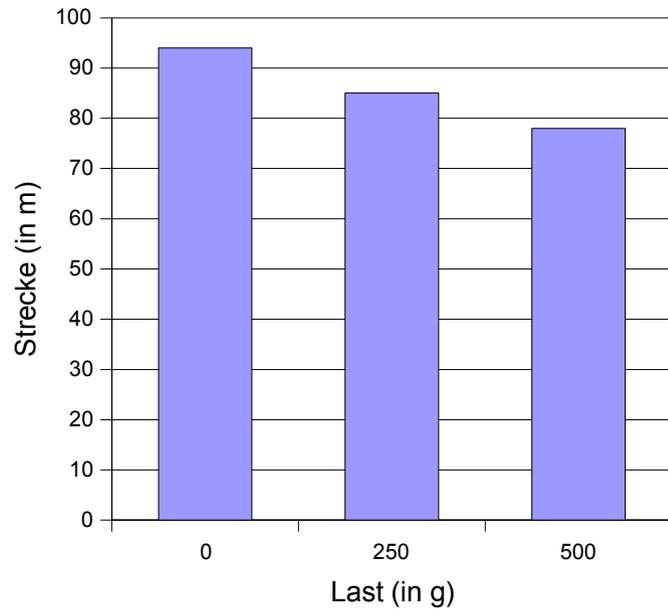
Messung – Umdrehungen in 30 Sekunden



Probleme:

- Motoren laufen nicht synchron

Messung – zurückgelegte Strecke in 10 Minuten

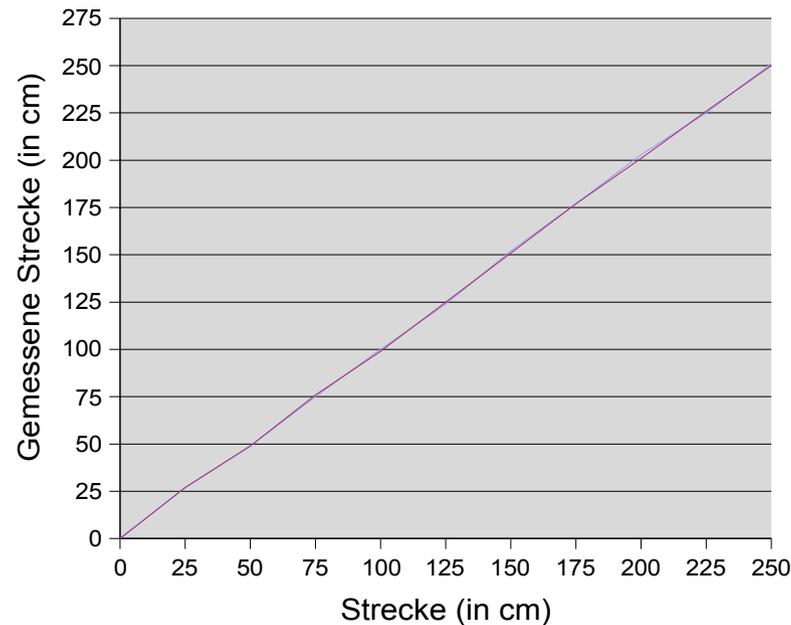


→ zurückgelegte Strecke nimmt bei größerer Last ab

Akku entladen:

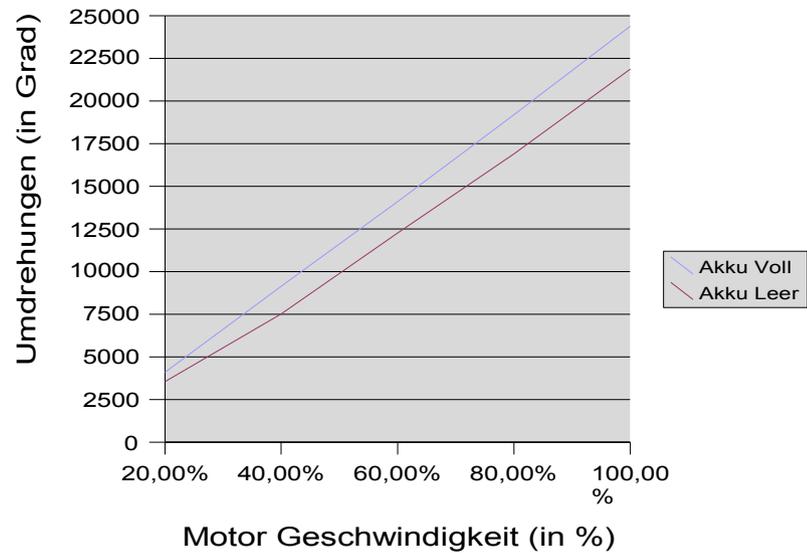
- Akku voll: 8156 mV
- Alle 3 motoren auf 100%
- Letzte Messung: 5012 mV
- Zeit: 3:14h

Messung – Ultraschall bei vollem und halbvollem Akku



Messergebnisse sind mehr oder weniger identisch
(Standardabweichung)

Messung – Motor bei vollem und leerem Akku



→ Motorleistung nimmt mit sinkender Akkuleistung ab.

Messdaten vom NXT Roboter an Logging-Station (Laptop) senden, speichern und visualisieren:

1. Ansatz:

1. NXT via Bluetooth oder USB mit dem Laptop verbinden
2. In BricxCC mithilfe der internen Watch Tools Messdaten in Echtzeit ausgeben und visualisieren

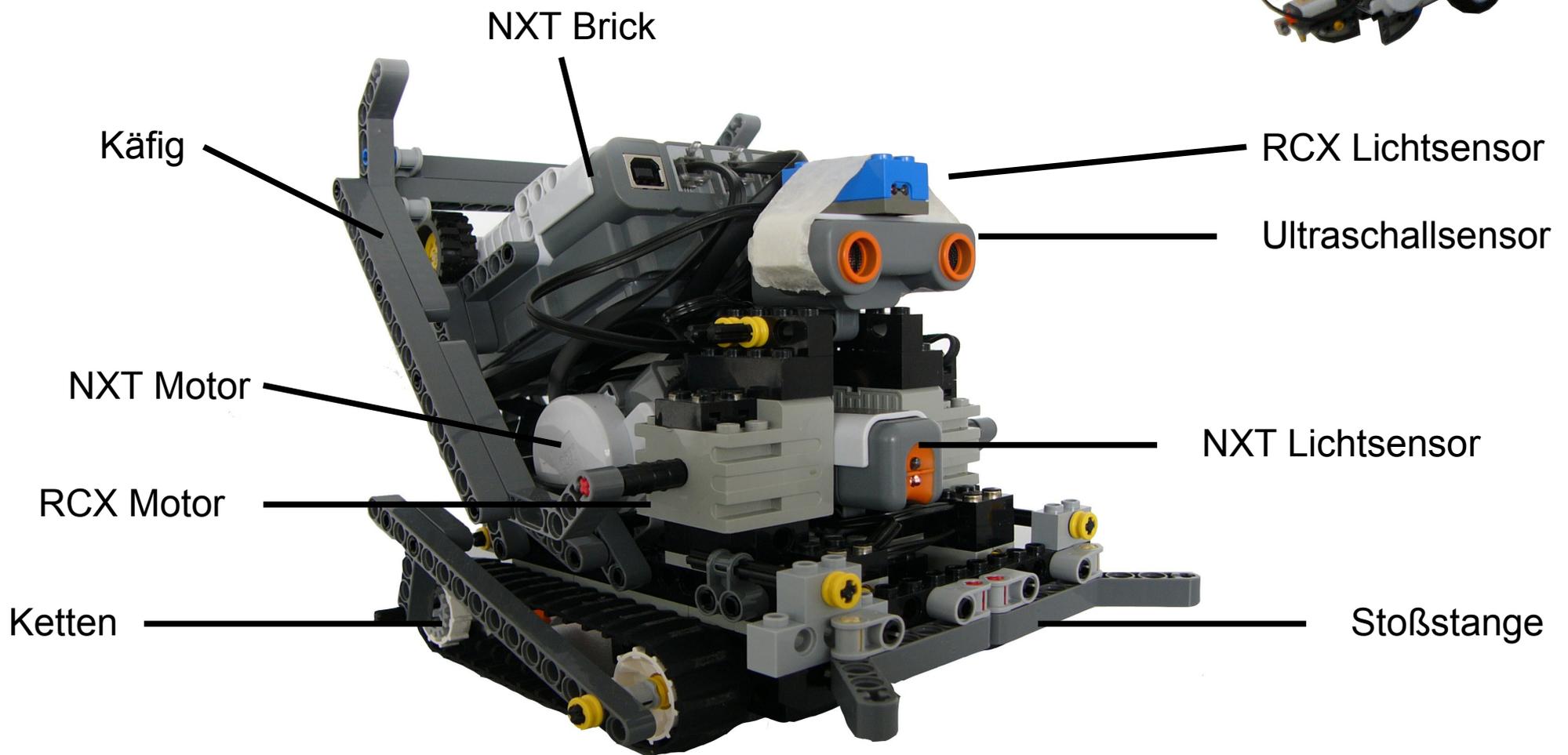
2. Ansatz:

1. Abspeichern von Messdaten in einer Textdatei auf dem NXT
2. Auslesen dieser via BricxCC

1. Roboter-Konstruktion
2. Ball-Detektion und -Suche
3. Ball einfangen
4. Ball zum Zielpunkt befördern
5. Hindernisse

1. Roboter Konstruktion

alte Konstruktion



1. Roboter Konstruktion

Ziele:

- robust
- wendig
- kompakt
- flexibel

Probleme:

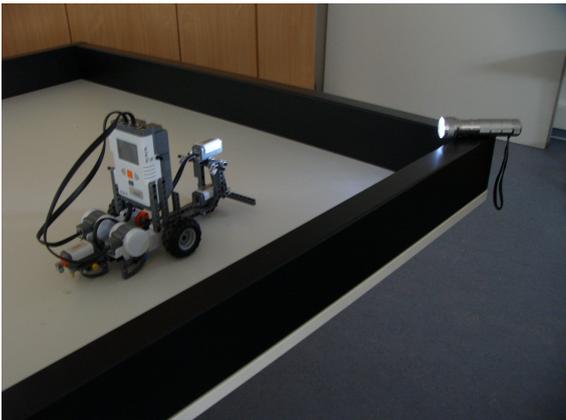
- durch Ketten ungenaue Drehbewegung
 - Toleranzwert bei unterschiedlichem Untergrund: Im Suchalgorithmus berücksichtigt
- Motoren laufen nicht synchron
 - ähnlichste Motoren ausgewählt und Bricx Methode OnFwdReg
- neue NXT Motoren sowie Sensoren schlecht zu verbauen
 - verschiedene Bauarten getestet
- zu wenig Teile im Kasten
 - alten RCX Bausatz hinzugenommen
- alte RCX Sensoren nicht vollständig kompatibel
 - verschiedene Modi ausprobiert und für den geeignetesten entschieden

2. Ball Detektion und Suche

Algorithmus zum Lichtsuchen:

1. Ansatz:

- Lichtsensor starr montiert
 1. Messung an vordefinierten Punkten → NXT dreht sich um 360° (4 Messungen)
 2. dreht sich zurück zum höchsten Messwert
 3. fährt geradeaus
 4. neue Messung → NXT dreht sich um 180° (4 Messungen) usw.
bis NXT gegen Lichtquelle fährt



→ Video

2. Ball Detektion und Suche

Algorithmus zum Lichtsuchen:

2. Ansatz:

- Lichtsensor starr montiert
1. NXT dreht sich um 360 Grad, misst dabei ständig die Motorumdrehungen und merkt sich die Position des größten Messwertes
 2. dreht sich zurück zur Position des größten Messwertes entsprechend der Drehzahl
 3. fährt vorwärts bis die gemessene Lichtintensität abnimmt mit Kollisionsvermeidung
 4. Fängt wieder mit Schritt 1 an

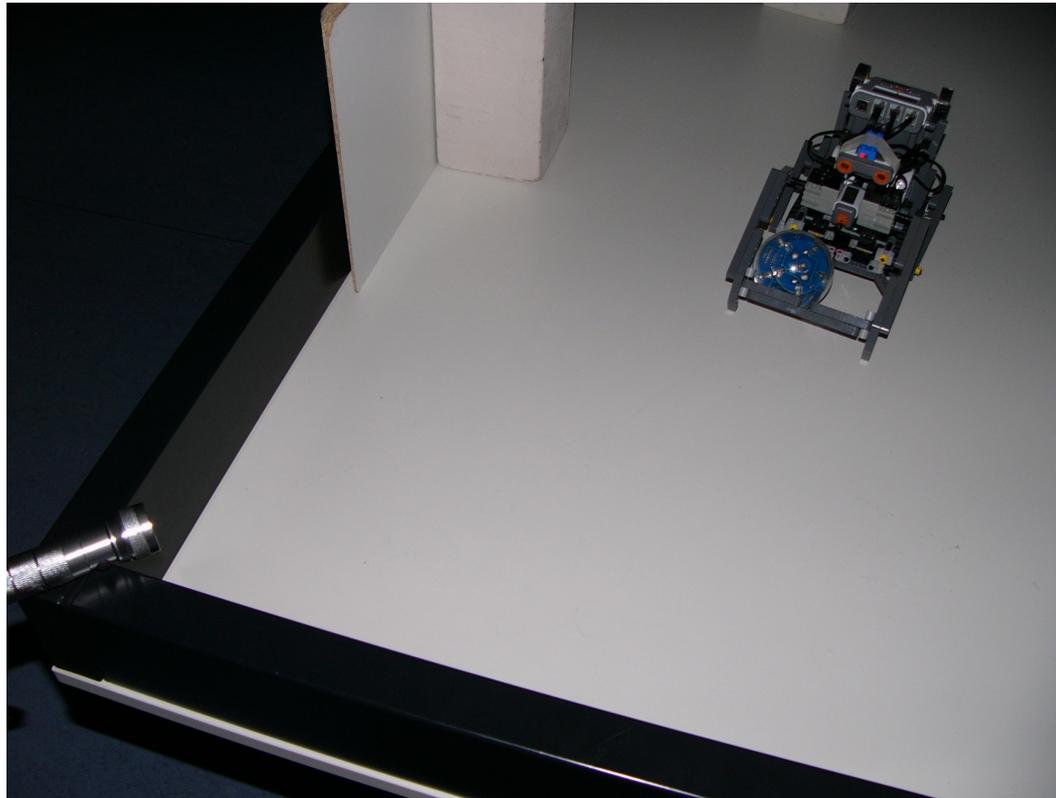
3. Ball Einfangen

1. Tastsensoren an Stoßstange werden im Vorwärtsmodus ständig abgefragt
2. bei Auslösen der Tastsensoren und entsprechender Lichtintensität wird Käfig geschlossen
3. Roboter geht in nächsten Modus



4. Ball zum Zielpunkt befördern

1. Ausführen des Lichtsuchalgorithmus mit zweitem Sensor
2. wenn Lichtintensität Grenzwert erreicht, wird Käfig geöffnet



5. Hindernisse

Beim Vorwärtsfahren werden Ultraschall- und Tastsensoren ständig abgefragt

- bei Kollision mit Stoßstange oder wenn Ultraschall ein Hindernis erkennt
→ Roboter geht in den Hindernis-Modus

Hindernis-Modus:

- bei Kollision mit Stoßstange fährt rückwärts und leichte Drehung
- wenn Ultraschall Hinderniss erkennt, dreht Roboter nach links und rechts und misst dabei die Entfernungen → wählt Weg mit weitester Entfernung
- Roboter fährt vorwärts bis eine neue Kollision auftritt oder er ein bestimmten Weg zurückgelegt hat
- geht zurück in den Suchmodus

- Praktikumsbedingungen klar formuliert
- regelmäßiges Treffen sorgte dafür, dass man den Anschluss nicht verliert
- vielseitige Aufgabenstellung erforderte immer wieder neue Konstruktionen, viel Testarbeit und Kreativität
- selbständiges Recherchieren und Problemlösen erforderlich

Fragen?