

Programmierung von
Steuerungsalgorithmen für mobile
Roboter (Lego-NXT)

SOFTWAREPRAKTIKUM SS 2007

(NICO GEBAUER, STEFAN KIRST, FLORIAN TANKE)

GLIEDERUNG

- × Motivation
- × 1. Teilaufgabe
- × 2. Teilaufgabe
- × 3. Teilaufgabe
- × Lösung des Hauptszenarios

MOTIVATION

- ✘ Lego ist ein interessantes „Spielzeug“
- ✘ Erste Programmierung von Autonomen Systemen
- ✘ Nicht nur rein theoretische Arbeit, sondern auch ein bisschen praktisch orientiert

Die ersten Schritte

1. TEILAUFGABE

1. TEILAUFGABE

- ✘ Annäherung an gegebene Hardware und Software unter Einbezug der im Internet angebotenen Quellen, sowie des NXT-Bausatzes
- ✘ Kennenlernen der einzelnen Entwicklungsumgebungen, sowie Gründe für die Anwahl bzw. Abwahl der einzelnen Umgebungen
- ✘ Auflistung der benutzten Quellen
- ✘ Programmierung einer ersten Steuerung zur Bewegung des Roboters in alle Richtungen, sowie Tests aller Motoren und Sensoren

TECHNISCHE DATEN DES BAUSATZES

✘ NXT-Baustein:

- CPU: 32-bit ARM7 Mikrocontroller 48 MHz Co-Processor: 8-bit AVR Mikrocontroller 4 MHz
- 4 Eingangsports / 3 Ausgangsports
- USB 2.0 und Bluetooth
- 100 x 64 Pixel LCD Display
- Lautsprecher

✘ Motoren:

- 3 Motoren mit integriertem Rotationssensor

✘ Sensoren:

- Lichtsensor
- Ultraschallsensor
- Tastsensor
- Tonsensor

„FRANK THE TANK I“

Konstruktion des ersten Roboters



„FRANK THE TANK I“

- ✘ Hierzu wurde die vom NXT-Baukasten gelieferte Bauanleitung verwendet, da so eine schnelle Inbetriebnahme zu ersten Tests der möglich war
- ✘ Die Montage lief daher schnell und reibungslos ab

DIE SENSOREN

- ✘ Die Tests der Motoren und Sensoren lief unproblematisch wie unspektakulär ab
- ✘ Interessant für spätere Arbeiten war, das beispielsweise die Drehbewegung der Motoren intern nach verfolgbar sind, und man so eine Stellung der Räder bzw. der zu bewegendenden Teile nachvollziehen kann

1. PROGRAMM ZUR BEWEGUNG IN ALLE RICHTUNGEN

```
#include "NXCDefs.h"
```

```
task main ()
```

```
{
```

```
    OnRev (OUT_A, 50);
```

```
    Wait(1500);
```

```
    OnFwd (OUT_AB,67);
```

```
        Wait(1500);
```

```
    OnRev (OUT_AB,67);
```

```
    ;
```

```
}
```

1. PROGRAMM ZUR BEWEGUNG IN ALLE RICHTUNGEN

Das in der vorherigen Folie vorhandene Programm zeigt auf einfache Weise die Bewegung in alle Richtungen.

Der Roboter vollführt hier eine Drehung nach rechts und fährt anschließend einmal vorwärts und einmal rückwärts.

EIN PAAR NÜTZLICHE INFORMATIONENQUELLEN

- × Einführung LEGO RCX vs. NXT
- × Essentials an NXT Tutorial
- × Lego öffnet den Zugang zur Firmware von Mindstorm-NXT
- × The Mindstorme NXT Review
- × Welcome to Next Byte Codes and Not eXactly C
- × WIKI Info
- × Lejos
- × ROBOLAB

WELCHE PROGRAMMIERUMGEBUNG?

Nach näherer Begutachtung der Quellen standen wir vor der Frage mit welcher Umgebung wir programmieren sollten.

Zur Auswahl stehen:

- NXC mit der BricxCC-Umgebung
- Lego Mindstorm-Software
- Lejos NXJ
- ROBOLAB
- und andere...

DIE LEGO-MINDSTORM-UMGEBUNG

- Hier war schnell klar, dass diese graphische Umgebung nicht für wissenschaftliche Verwendung geeignet ist

Vorteile	Nachteile
<ul style="list-style-type: none">–Einfache Bedienung durch Drag & Drop–Leichtes Erstellen von Programmen, da vorgefertigte Module vorhanden sind–Keine Programmierkenntnisse nötig	<ul style="list-style-type: none">–relativ unflexibel–Durch die Module nur bedingt zur Realisierung von eigenen Ideen geeignet

ROBOLAB

- Diese Umgebung wird zwar im Internet als recht umfangreich diskutiert
- Jedoch ist hier der Nachteil, dass dies eine kommerzielle Umgebung ist

LEJOS

- Lejos stellt eine Java- Umgebung zur Programmierung des NXT-Bausteines dar
- Nachteile sind hier, dass sich:
 - o die aktuelle Version in der alpha- Phase befindet
 - o die Java-Einbindung relativ schwierig gestaltet (wir haben es nach einer halben Stunde aufgegeben)

BRICXCC

- Das Brick Command Center ist eine C-ähnliche Möglichkeit zur Programmierung des NXT-Bausteins
- Wir haben uns hier schnell eingearbeitet und sofort wohlgefühlt.
- Die Möglichkeit der Code-unabhängigen Ansteuerung des Bausteins für Funktionalitätstests und das einfache aber mächtige Auftreten der Umgebung überzeugten
- Es bieten sich hier alle Möglichkeiten der Implementierung und Ausnutzung der Fähigkeiten des Roboters
- Somit fiel unsere Wahl auf diese Umgebung

Die Sensoren

2. TEILAUFGABE

2. TEILAUFGABE

Was sind Sensorschnittstellen?

In welcher Form liegen Messdaten vor?

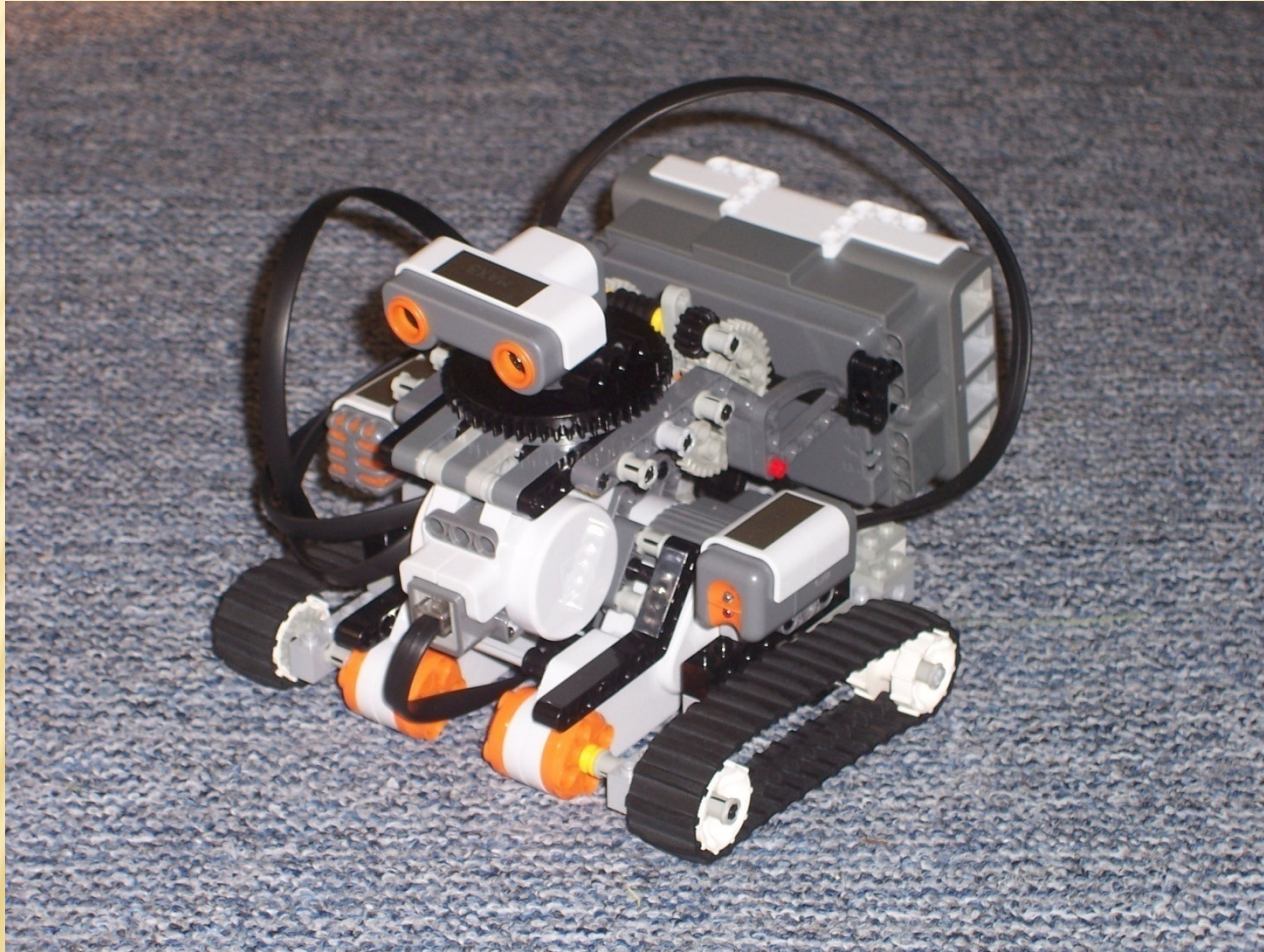
Welchen Typus sind sie?

Qualitätsbestimmung: Logging von Sensor-Daten und deren Speicherung für die Ermittlung von Kennlinien.

„FRANK THE TANK II“ - BETA



„FRANK THE TANK II“



BLUETOOTH

- ✘ Die Bluetooth-Einheit des NXT-Bausteins ist als kabellose Kommunikationsvariante durchaus geeignet.
- ✘ Die von uns festgestellte Reichweite einer aktiven Verbindung zum Rechner beträgt ca. 25 Meter.
- ✘ Bei größeren Entfernungen bricht die Verbindung zusammen bzw. ist nicht mehr herstellbar.
- ✘ Desweiteren ist es möglich mit anderen Robotern über Bluetooth zu kommunizieren

AKKULEISTUNG

- ✘ Stellt die Akkuleistung eine kritische Leistungsgröße für die Kommunikationsqualität oder Sensoren dar?

Eine schwächer werdende Akkuleistung hat zur Folge, dass die Motoren nicht mehr die Angestrebte Leistung erbringen, Sensoren falsche bzw. ungenügende Messwerte liefern und die Bluetooth- Verbindung zusammenbricht.

DIE SENSOREN UND IHRE MESSWERTE

- ✘ Welche Messwerte liefern welche Komponenten; in welchen Größenordnungen? Feststellung der Messwertgüte unter versch. Bedingungen.
- ✘ *Vorgehensweise: Für Licht-, Ultraschall- und Akustiksensoren wird ein Bandmaß auf den Boden gelegt und dann ein Objekt in diskreten Abständen vom zu testenden Sensor fortbewegt.*

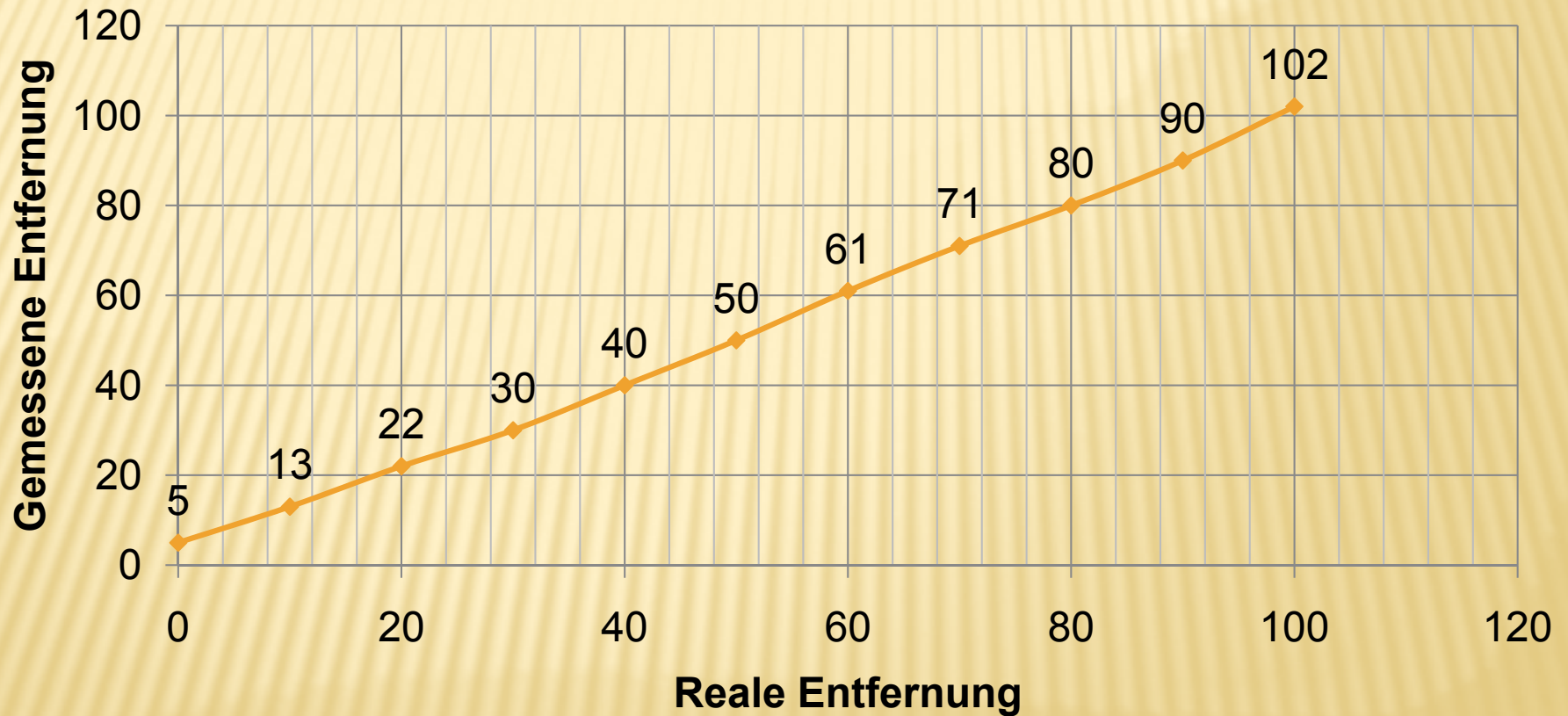
DIE SENSOREN

× Die verschiedenen Sensoren und ihre Eigenschaften

Sensor	Einheit	Verwendung
Ultraschallsensor	Entfernung in cm und inch	Entfernungsbestimmung
Lichtsensor	Helligkeit in Prozent	Suche nach Lichtquellen
Soundsensor	Lautstärke in dB und dBA	Steuerung per Geräusche
Tastsensor	Wert 1 oder 0	Erkennung physischer Kontakte
Motoren	Rotationen im Gradmaß	Erfassung der Umdrehung eines Motors

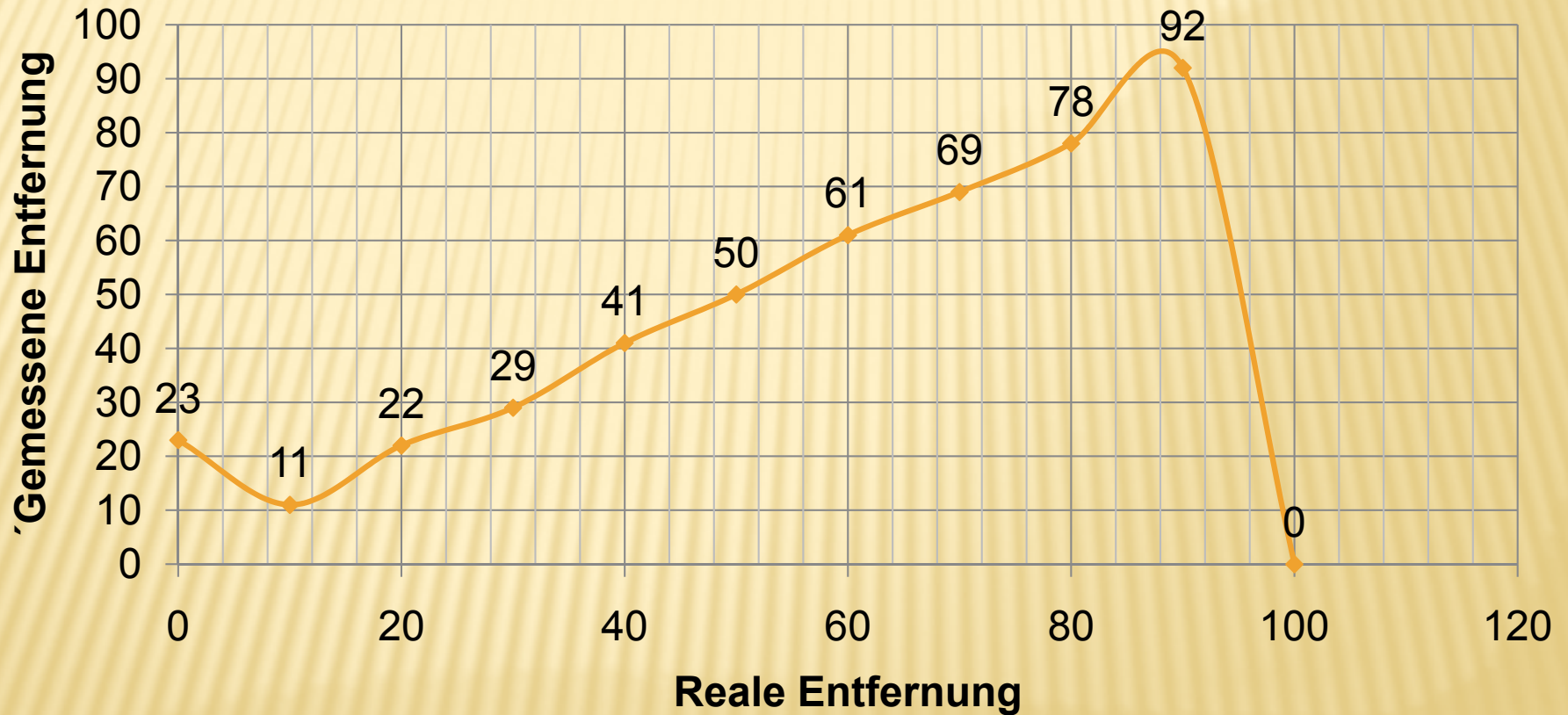
DER ULTRASCHALLSENSOR

Obj.: Teller



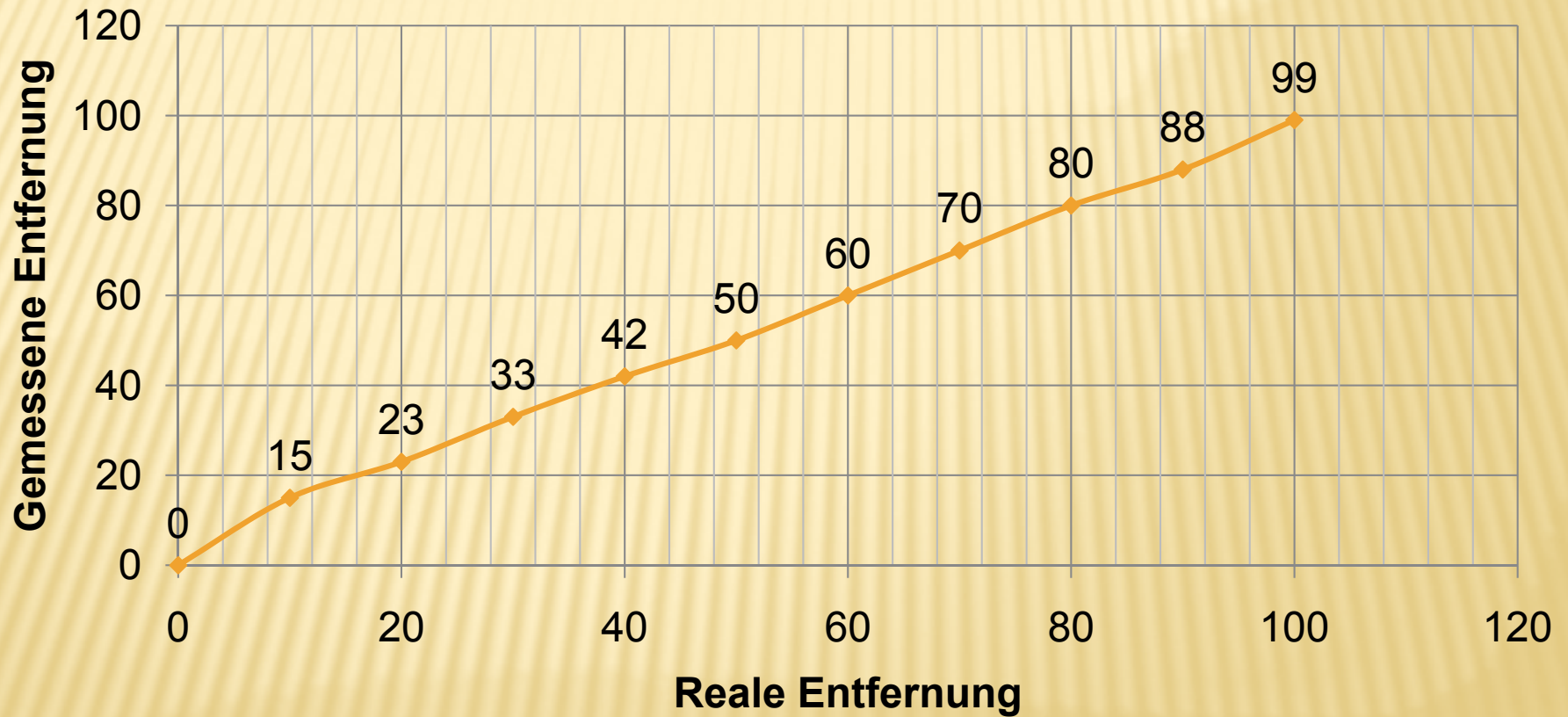
DER ULTRASCHALLSENSOR

Obj.: Pappe



DER ULTRASCHALLSENSOR

Obj.: Kissen

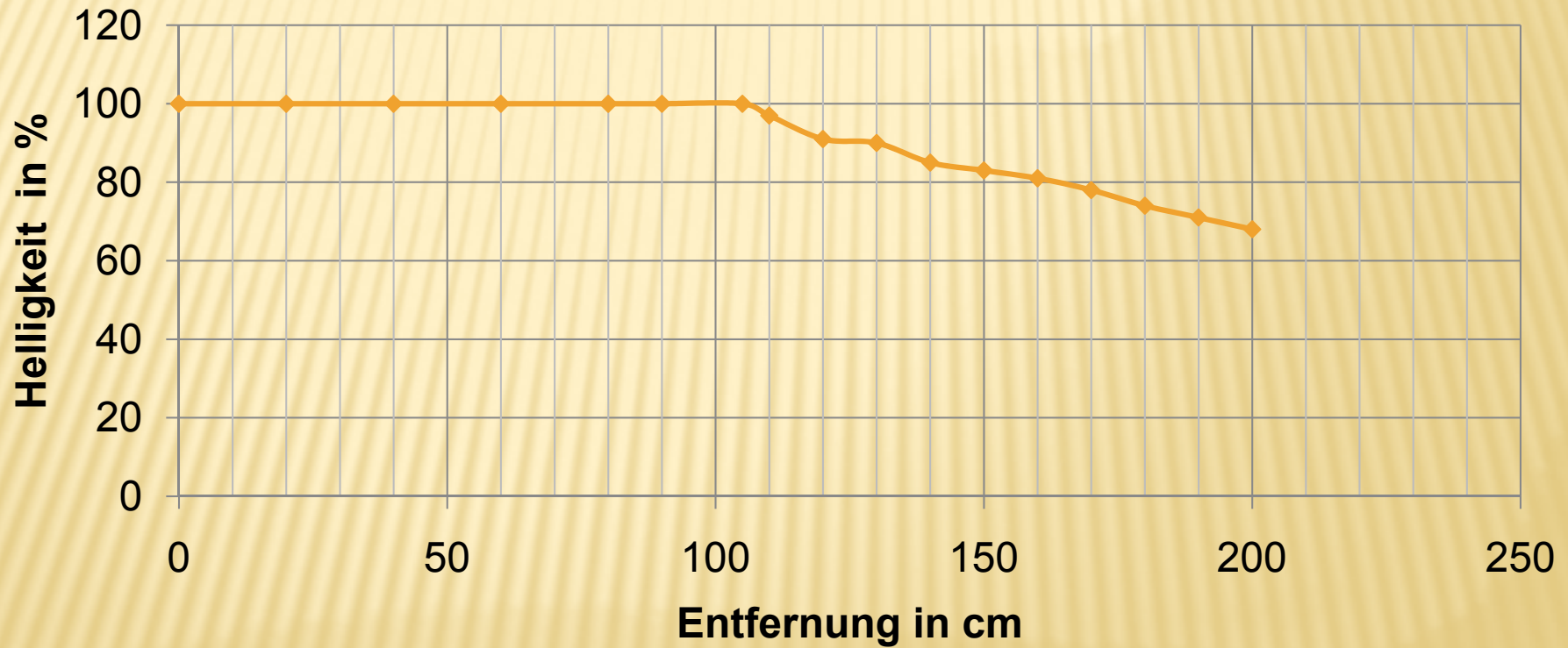


DER ULTRASCHALLSENSOR - AUSWERTUNG

- ✘ Die Reichweite des Sensors ist bedingt vom Material abhängig (ca. 150 cm)
- ✘ Man kann sagen, dass der Sensor recht genaue Messwerte liefert (ca. $\pm 3\text{cm}$)

DER LICHTSENSOR

[Umgebunghelligkeit : 46%]



DER LICHTSENSOR - AUSWERTUNG

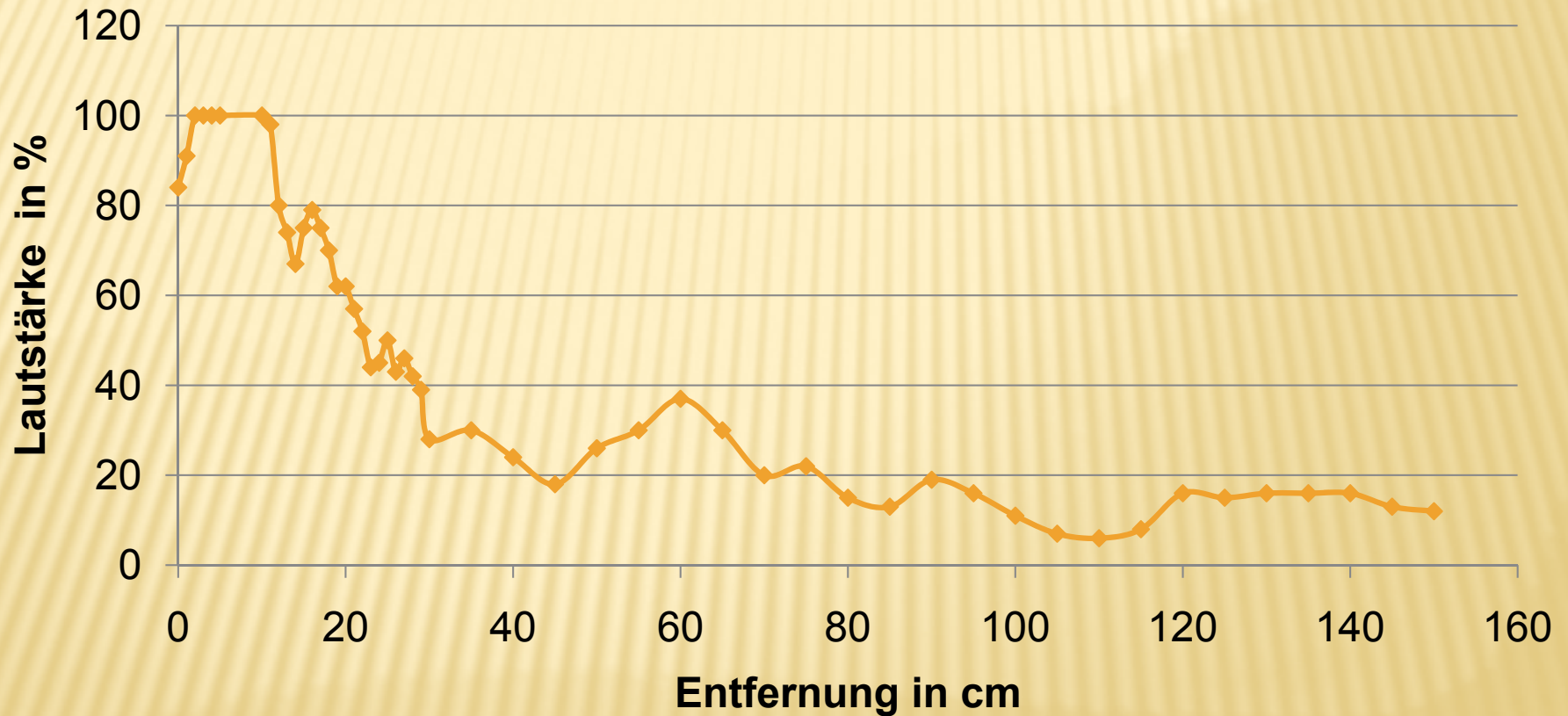
- ✘ Der Lichtsensor hat zwei Operationsmodi:
Ambientlight und Reflectedlight

- ✘ Bei heller Raumhelligkeit liefert der Sensor stetig 100%.
 - Somit ist der Gebrauch in einer solchen Umgebung von keinem Nutzen.

- ✘ Bei dunkler und mittlerer Raumhelligkeit sind die gelieferten Ergebnisse gut
 - Eine differenzierte Lichtquellenortung ist bei entsprechender Umgebungsbeleuchtung also möglich.

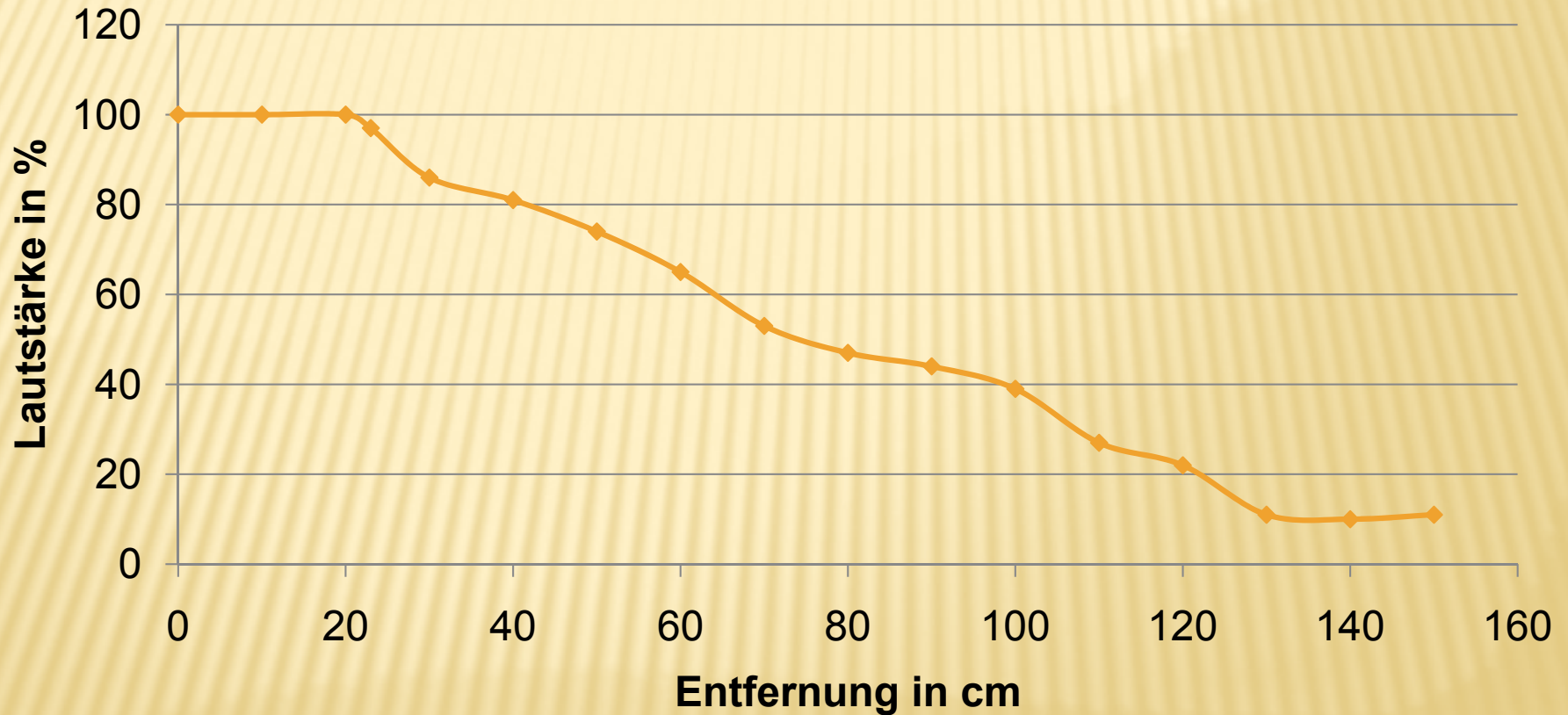
DER SOUNDSENSOR

[mittlere Lautstärke]



DER SOUNDESENSOR

[geringe Lautstärke]



DER SOUNDESENSOR – AUSWERTUNG

- ✘ Beim Soundsensor waren über den gesamten Messbereich starke Schwankungen zu beobachten
- ✘ Lautstärkemessungen einer lauten Quelle liefern stets volle 100%
- ✘ Problematisch sind die Schwankungen bei einer mittelstarken Quelle bzw. der dauerhaft volle Ausschlag bei starken Quellen

DER TASTSENSOR

- × Ist ein rein analoger Sensor
- × Dient zur Kollisionserkennung
- × Damit er ausgelöst wird benötigt es 34 Gramm Gewicht, also 0,34 Newton

DIE MOTOREN

- × Beim Test der Motoren ist uns aufgefallen, dass (komischerweise) kein Leistungsabfall bei höherem Gewicht festzustellen ist
- × Unsere Messungen auf einer Strecke von 1,5 Metern:

Gewicht in g	Zeit in s
Eigengewicht	8,3
250	8,2
500	8,0

DIE MOTOREN

- ✘ Wir haben festgestellt das mit schwächer werdendem Akku die Motorleistung abnimmt
- ✘ Uns ist auch aufgefallen, dass die Motoren unter Vollast mehr Strom verbrauchen

3. TEILAUFGABE

- 1. Messdaten vom NXT Roboter an Logging- Station (Laptop) senden, speichern und visualisieren**
- 2. Ball Detektion**
- 3. Fernsteuerung via Bluetooth**

LOGGING VON MESSDATEN

- ✘ Dies ist möglich mit dem in BricxCC integriertem Tool „Watch the Brick“
- ✘ Hier kann man alle Werte von Sensoren und Motoren ablesen

Watching the brick [?] [X]

	<input checked="" type="checkbox"/> A	<input checked="" type="checkbox"/> B	<input checked="" type="checkbox"/> C
<input checked="" type="checkbox"/> Power	0	0	0
<input checked="" type="checkbox"/> Mode	0	3	3
<input checked="" type="checkbox"/> Reg Mode	1	0	0
<input checked="" type="checkbox"/> Run State	0	32	32
<input checked="" type="checkbox"/> Turn Ratio	-100	-100	-100
<input checked="" type="checkbox"/> Tacho Lim	180	0	0
<input checked="" type="checkbox"/> Tacho Cnt	-147	21	23
<input checked="" type="checkbox"/> Block TCnt	-147	21	23
<input checked="" type="checkbox"/> Rotation	-147	-1464	1699

Port	US Buffer	Length	Response
<input type="checkbox"/> I2C 1	<input type="checkbox"/>	0	
<input type="checkbox"/> I2C 2	<input type="checkbox"/>	0	
<input type="checkbox"/> I2C 3	<input type="checkbox"/>	0	
<input type="checkbox"/> I2C 4	<input type="checkbox"/>	0	

<input checked="" type="checkbox"/> Sensor 1	1023	All
<input type="checkbox"/> Sensor 2	0	None
<input checked="" type="checkbox"/> Sensor 3	525	Clear
<input checked="" type="checkbox"/> Sensor 4	147	

<input type="checkbox"/> Timer 0		Poll Now
<input type="checkbox"/> Timer 1		Poll Regular
<input type="checkbox"/> Timer 2		100 ms
<input type="checkbox"/> Timer 3		<input type="checkbox"/> Only if active

Message

Sync series

Graph

Help

BALL DETEKTION

- ✘ Zur Ball Detektion haben wir den Modus Ambientlight gewählt
- ✘ Der Reflectedlight Modus wurde von uns nicht weiter getestet

FERNSTEUERUNG VIA BLUETOOTH

- ✘ Dies haben wir mit dem Notebook ausprobiert
- ✘ Hierzu mussten wir nur einen Treiber installieren, womit der PC den NXT erkennen konnte
- ✘ Nun konnten wir den NXT auch schon über das Steuerkreuz aus er BricxCC- Umgebung steuern

Technische Umsetzung und Programmiertechnik

LÖSUNG DES HAUPTSZENARIOS

„FRANK THE TANK III“



UNSERE DEMO

- ✘ Nun gibt es die Live- Demo
- ✘ Nebenbei ein paar Erklärungen zu den technischen Besonderheiten
- ✘ Danach noch ein kurzer Blick auf den Quelltext

DAS WAR'S

Vielen Dank
für eure Aufmerksamkeit!

Noch Fragen?