

LEGO Mindstorms



Softwarepraktikum Teamrobotik

SS 2007

Thema:

- Programmierung von Steuerungsalgorithmen für mobile Roboter

Gruppe 1:

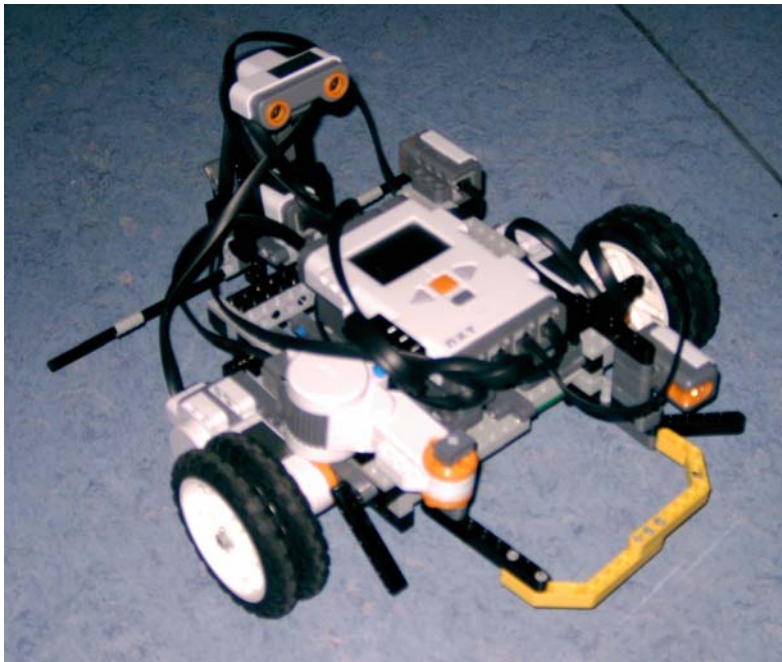
- Stefanie Müller und Ina Bosse

Hardware: LEGO MINDSTORM

Programmiersprache: NXC

Programmierungsumgebung: Bricx

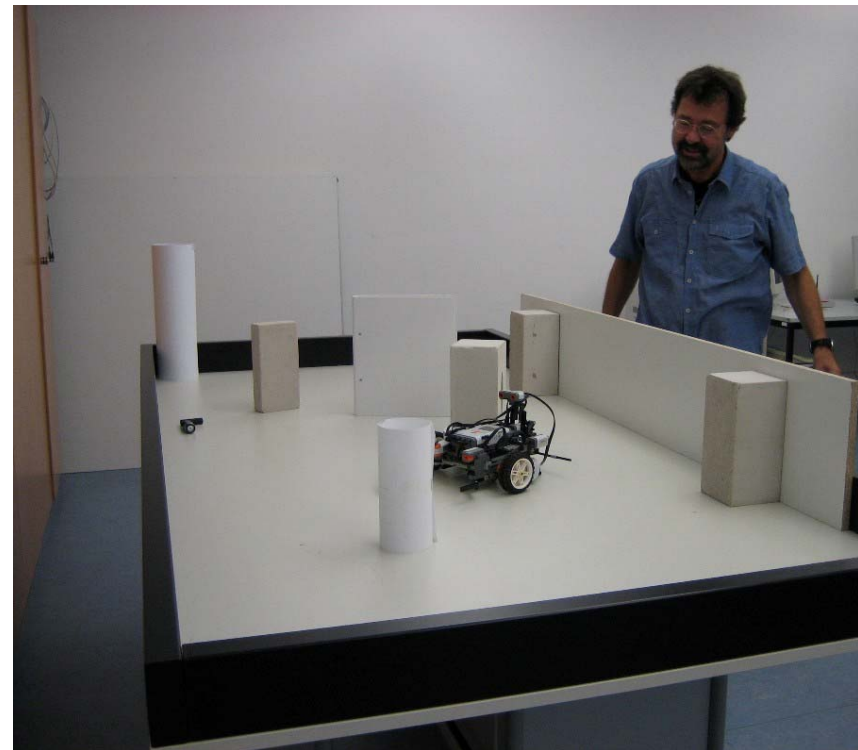
Gliederung



- Projektbeschreibung
- Allgemeines
- Konstruktion
- Sensoren
- Ausblick
- Quellen

Projektbeschreibung I

- Versuchstisch mit rechteckigen und runden Hindernissen
- Umfahren von Hindernissen



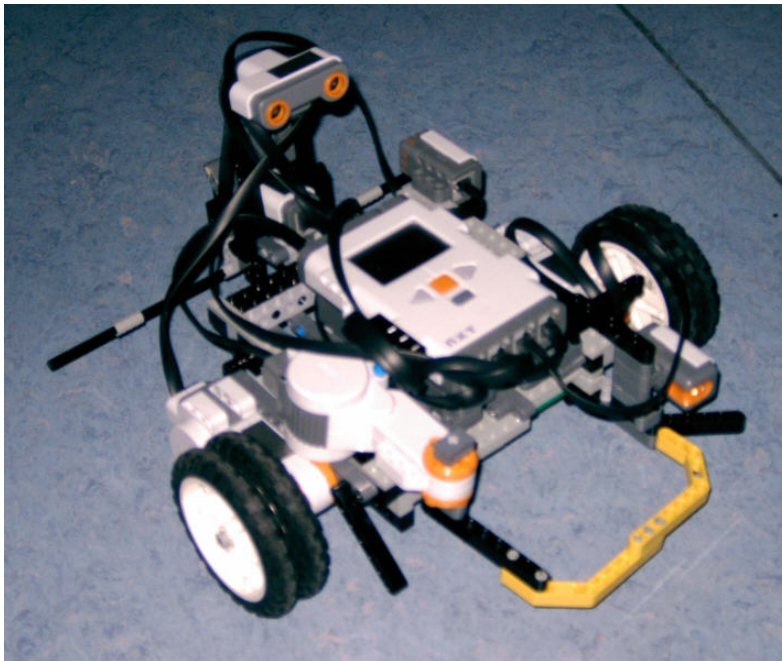
Projektbeschreibung II

- Suchen der LED-Kugel
- Aufnahme der Kugel
- Transport zu einer zweiten Lichtquelle



Gliederung

Allgemeines



- Allgemeines
 - NXT-Baustein
 - Sensoren
 - Motoren
 - LED - Kugel



meines *ustein*



- Weiterentwicklung des RXC
- 32-bit ARM7 Mikroprozessor
- 256kbyte FLASH- und 64kbyte RAM-Speicher
- Bluetooth Technologie für kabellose Kommunikation
- USB 2.0 Anschluss
- 4 Eingänge für Sensoren
- 3 Ausgänge für Motoren und Lämpchen
- Lautsprecher
- programmierbares, graphisches Display (60 x 100 Pixel)

Allgemeines

Sensoren

- *Geräuschsensor misst Schall in db und dbA, kann Klangmuster und unterschiedliche Tonhöhen erkennen*
- *Skala des Lichtsensors reicht von 0% (kein Licht) bis 100% (sehr hell), wenig geeignet Farben zu erkennen*
- *Ultraschallsensor misst die Zeit zwischen Aussendung und Rückkehr des Schalls und berechnet daraus die Entfernung zum Objekt*
- *Tastsensor kann nur zwischen zwei Zuständen (Tastsensor gedrückt/nicht gedrückt) unterscheiden, meldet ab einem bestimmten Druck eine Berührung*



neines



- Motoren verfügen über einen integrierten Rotationssensor, dieser kann die genaue Geschwindigkeit und zurückgelegte Entfernung messen
- präzise Fahrmanöver möglich, da die Motoren in 1° Schritten angesteuert werden können



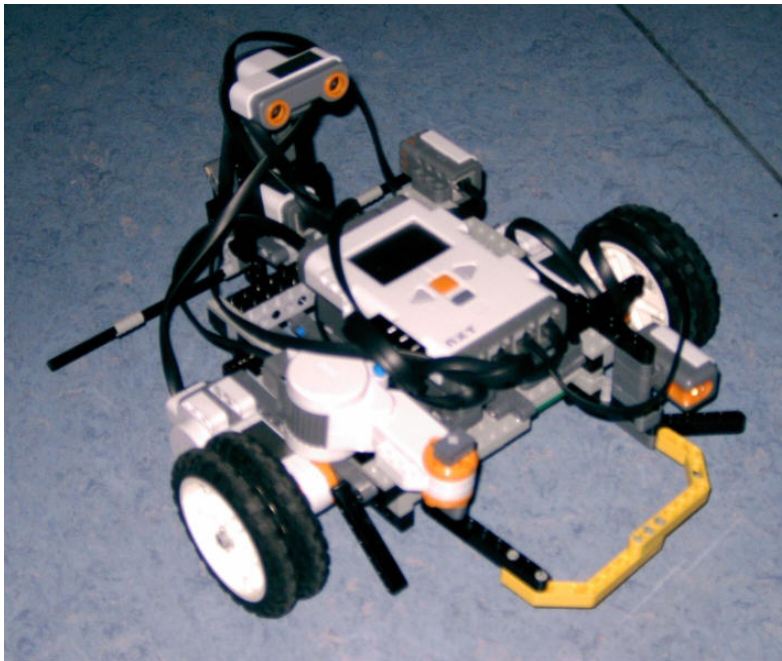
emeines *Kugel*



- Sendet über LEDs Infrarotlicht aus
- Ist ab 60 cm als Lichtquelle erkennbar

Gliederung

Konstruktion



- Konstruktion
 - Mit Ketten
 - Mit Steuerung hochkant
 - Mit Stützbalken
 - Endgültige Version

Konstruktion I

Ketten



Versuche mit Ketten brachten nicht den gewünschten Erfolg, schwierig war es die Ketten mit den schweren Motoren zu verbinden.

Konstruktion II

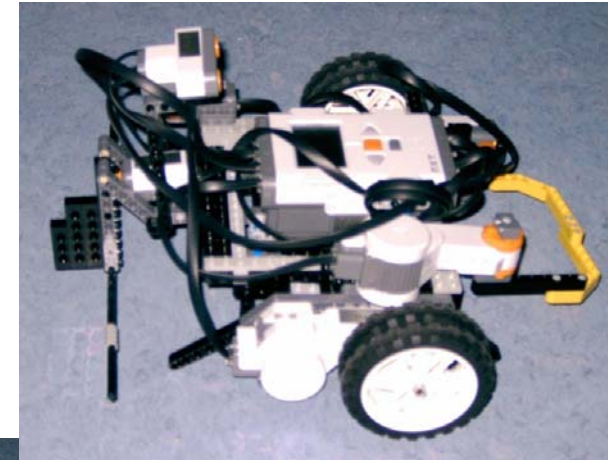
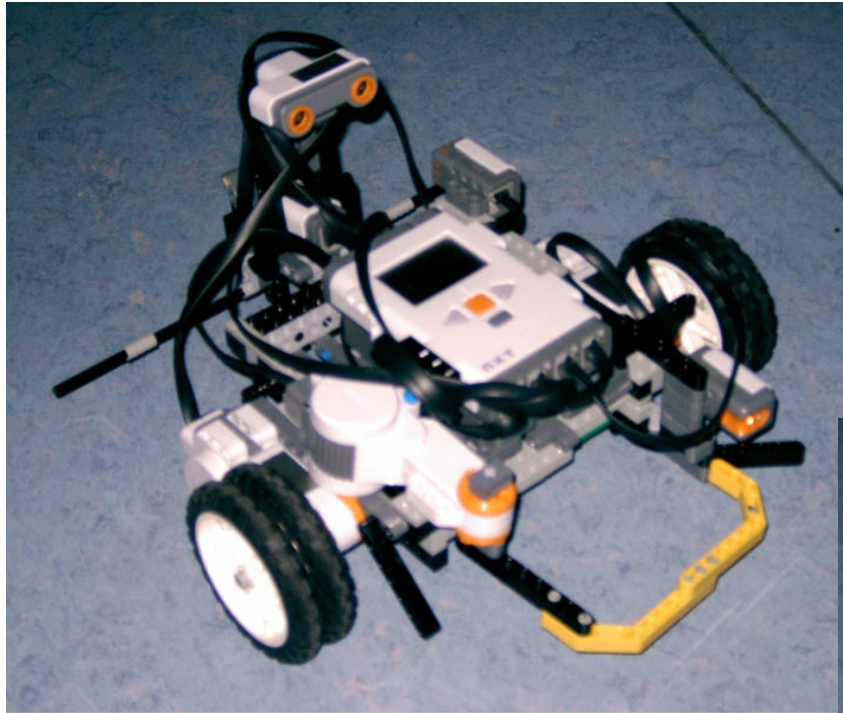
Steuerung hochkant



Die Bodenplatte war zu instabil, das Gewicht war falsch verteilt. Die großen hinteren Räder führten zu ungünstigen Fahreigenschaften (zu großer Wendekreis).

Konstruktion III

Stützbalken



Der Abbau der hinteren Räder und eine Umverteilung des Gewichtes führten zu besseren Fahreigenschaften auf glattem Untergrund.

Stefanie Müller & Ina Bosse

Konstruktion IV

endgültige Variante

- Umbau des Armes
- Zweiter Tastsensor, der bei Berührung (Kugel) den Arm schließt
- variabler Lichtsensor
- Problem:
zweiter Tastsensor reagiert nur auf starken Druck



Gliederung

Sensoren



- Sensoren
 - Geräusch
 - Geräusch mit laufendem Motor
 - Licht
 - Ultraschall





Ultraschallsensor



Versuchsaufbau



Lautsprecher mit angeschlossenem MP3-Player



Maßband

Lego-Roboter



Mikroschensensor



Lautstärkepegel:

- laut, mittel, leise
- größere Messwertfehler bei geringerer Entfernung zum Sensor
 - Überlagerung der Schallwellen
- Messwerte durch Hintergrundgeräusche leicht verfälscht



uschsensor



Abstand in cm	10	20	30	40	50	60	70	80	90	100
Messung 1	67	59	50	34	37	26	24	16	23	14
Messung 2	82	58	52	32	35	34	28	31	21	17
Messung 3	85	70	52	40	39	32	19	29	25	21
Messung 4	56	42	32	39	32	33	28	29	27	17
Messung 5	63	59	42	39	31	21	23	27	14	12
Mittelwert	70,6	57,6	45,6	36,8	34,8	29,2	24,4	26,4	22	16,2
Std.	11,2	8,96	7,74	3,19	2,99	4,96	3,38	5,35	4,47	3,06

laut

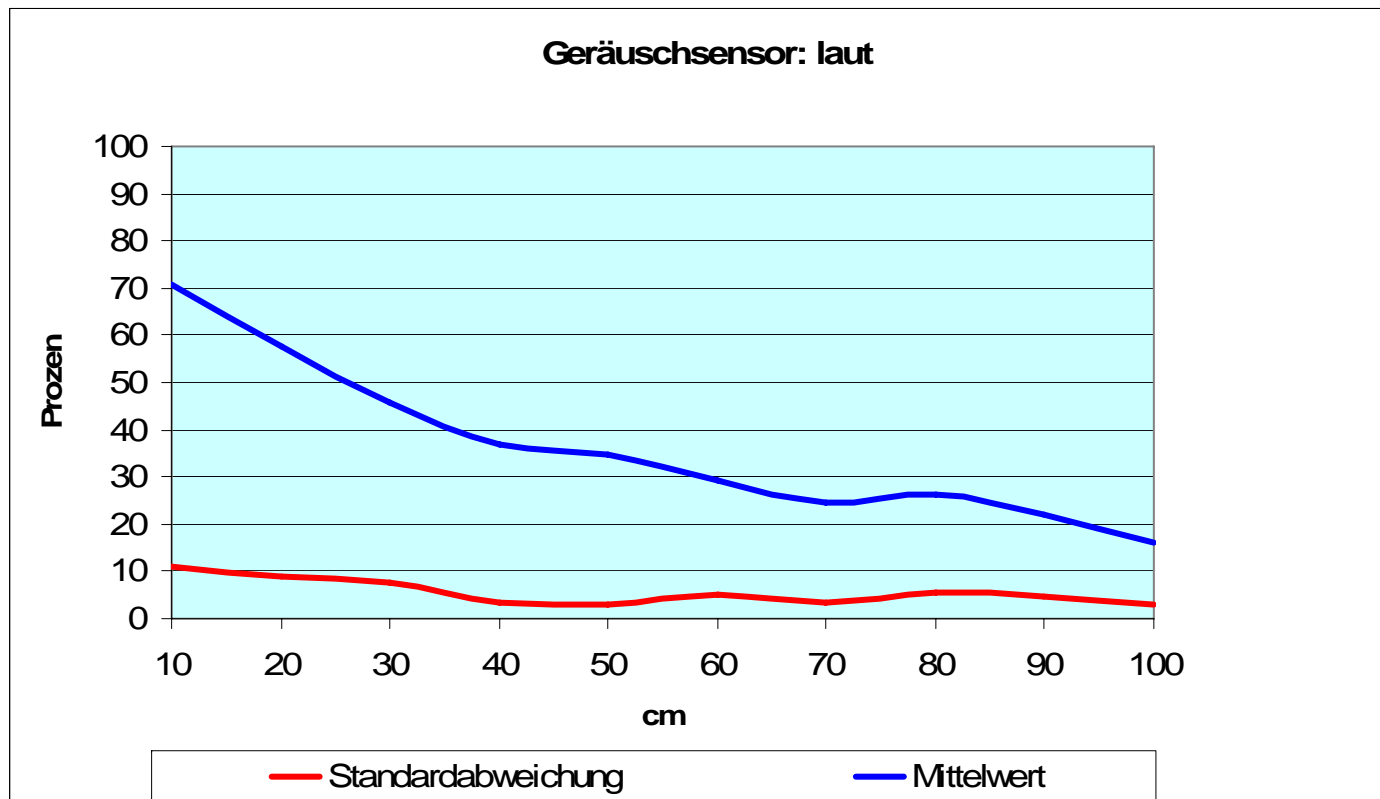
Std. - Standardabweichung

Messwerte in dB

Stefanie Müller & Ina Bosse



Geräuschsensor





Buschsensor

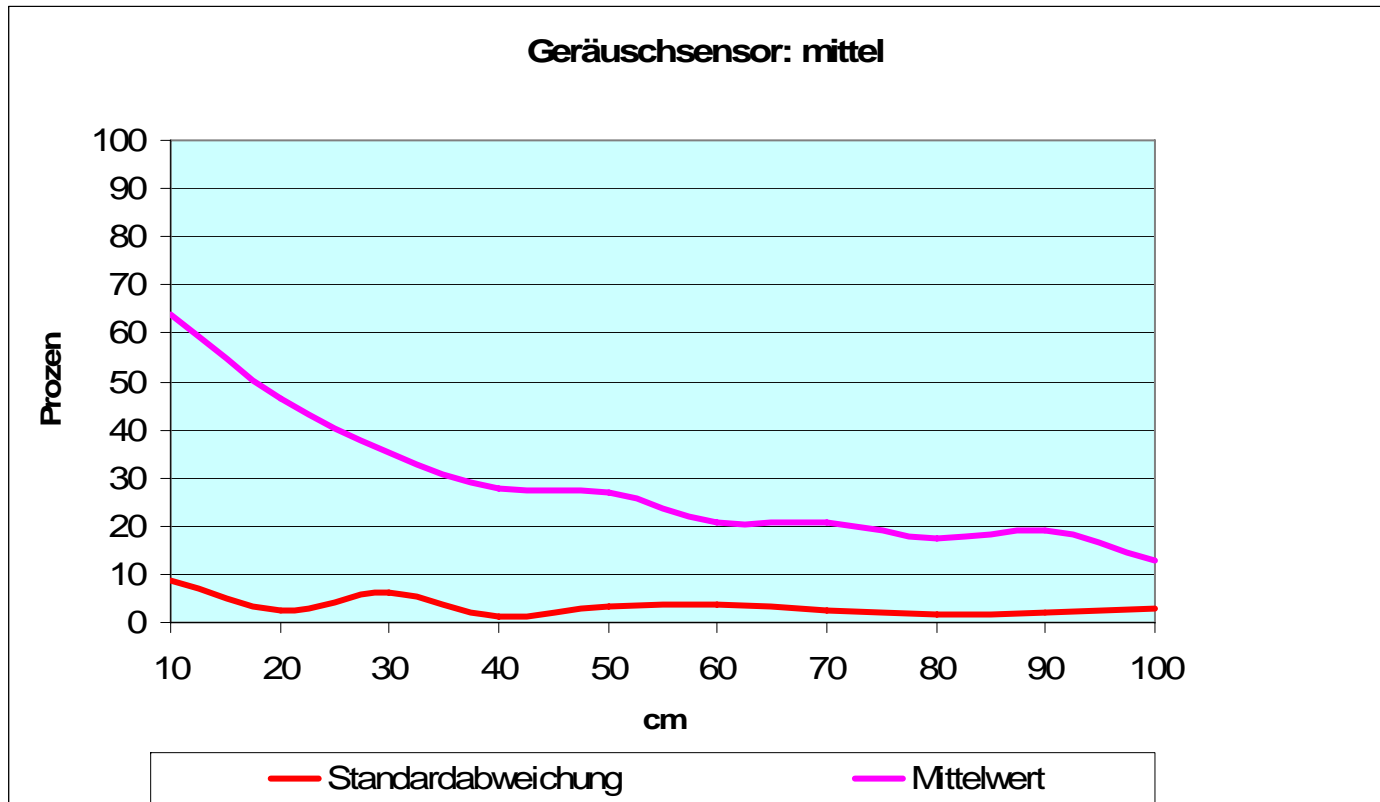


Abstand in cm	10	20	30	40	50	60	70	80	90	100
Messung 1	55	48	27	30	26	16	19	21	20	9
Messung 2	69	49	39	27	31	17	25	17	21	12
Messung 3	52	49	29	26	28	21	21	18	18	17
Messung 4	74	43	42	29	21	25	21	16	16	16
Messung 5	70	44	40	28	28	24	18	16	21	11
Mittelwert	64	46,6	35,4	28	26,8	20,6	20,8	17,6	19,2	13
Std.	8,79	2,58	6,15	1,41	3,31	3,61	2,4	1,85	1,94	3,03

mittel



Geräuschsensor





Ultraschallsensor

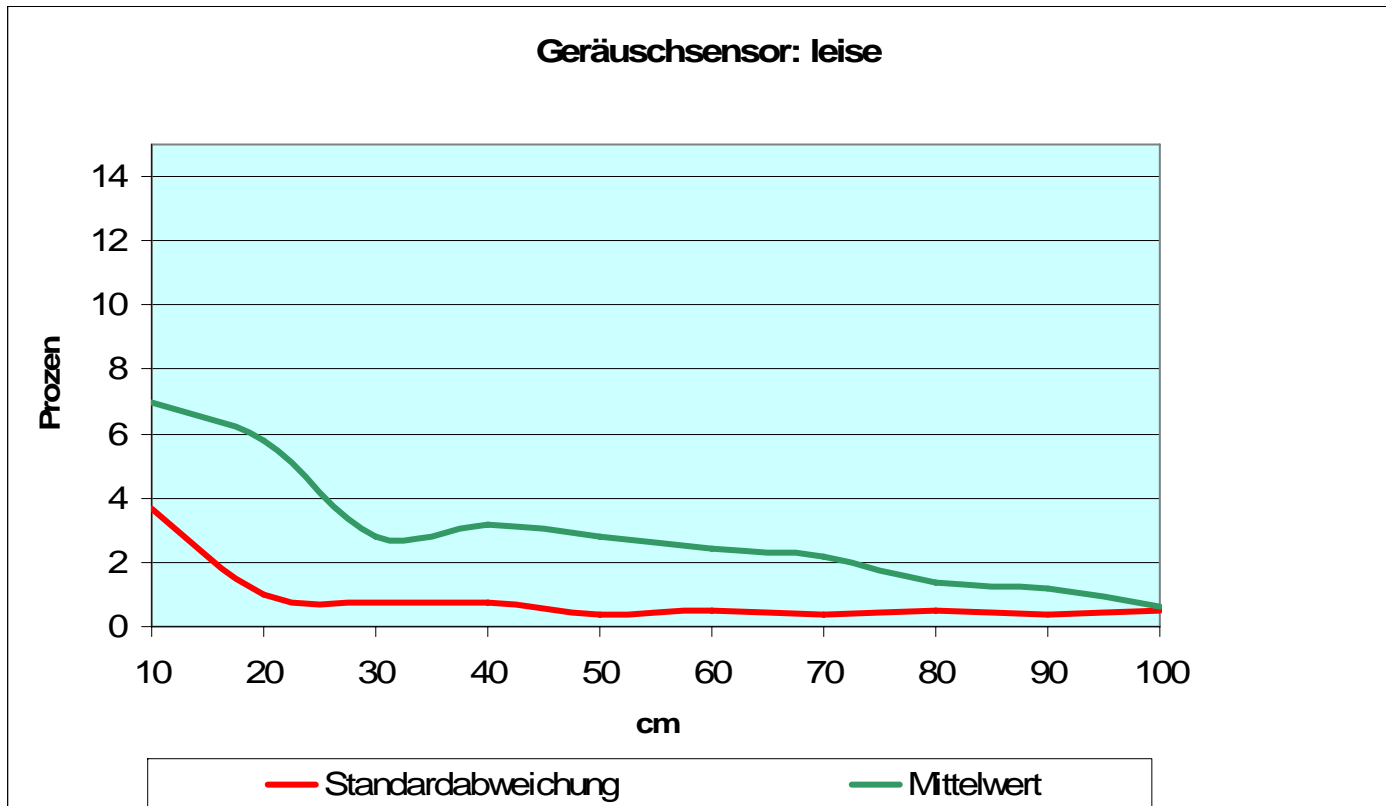


Abstand in cm	10	20	30	40	50	60	70	80	90	100
Messung 1	10	7	3	4	2	2	3	2	1	1
Messung 2	7	6	2	4	3	3	2	1	1	0
Messung 3	10	6	2	3	3	3	2	1	2	1
Messung 4	0	6	4	2	3	2	2	1	1	1
Messung 5	8	4	3	3	3	2	2	2	1	0
Mittelwert	7	5,8	2,8	3,2	2,8	2,4	2,2	1,4	1,2	0,6
Std.	3,69	0,98	0,75	0,75	0,4	0,49	0,4	0,49	0,4	0,49

leise

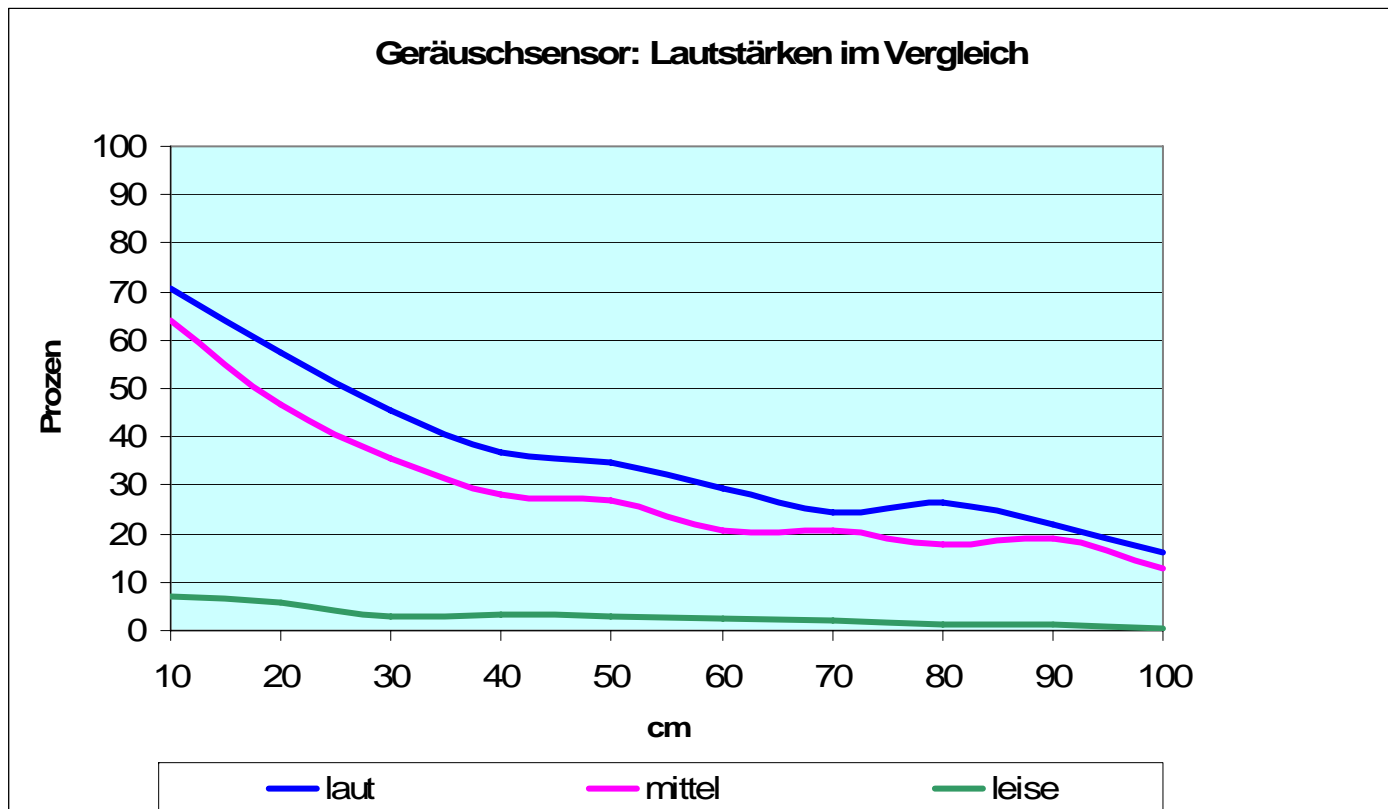


Geräuschsensor



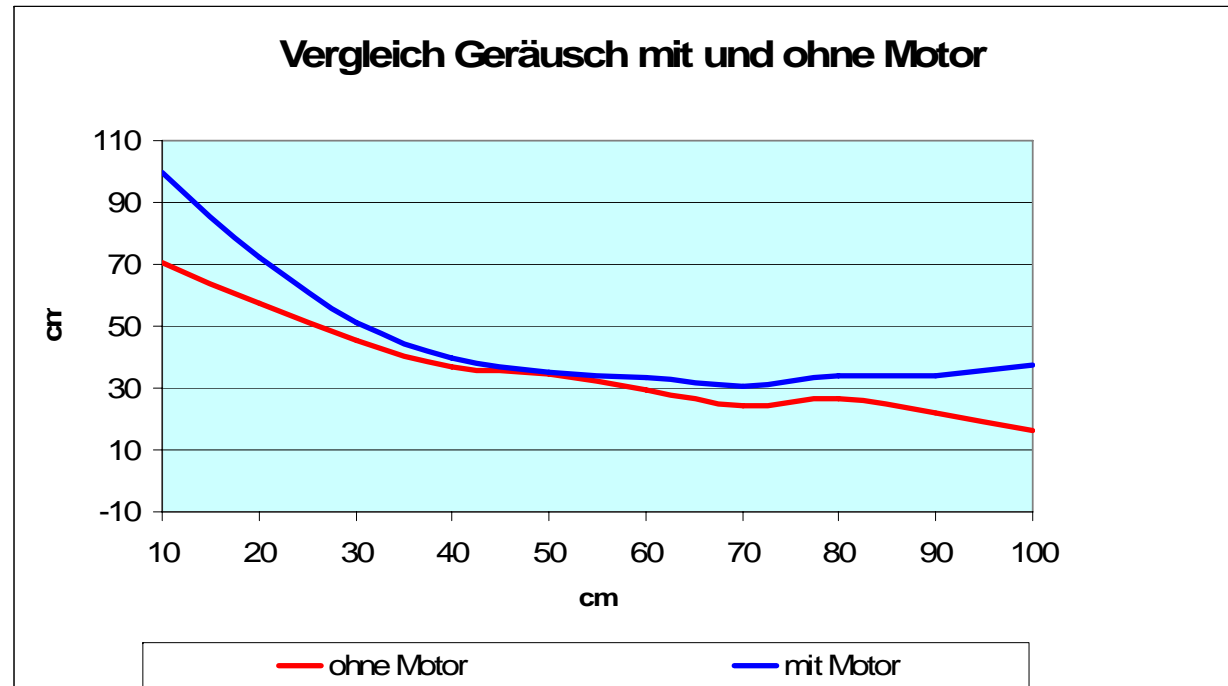


Geräuschsensor





uschsensor mit ndem Motor



- Mit laufendem Motor können im Gegensatz zu den Tests ohne Störgeräusche weiter entfernte Geräuschquellen (circa ab 50 cm) nicht mehr wahrgenommen werden

Lichtsensor



Lichtquellen:

- stark
 - kleiner Abstand → Lichtintensität 100%
 - dann → langsam abnehmend
 - mehr Messwertfehler ab 40 cm
- blau
 - Lichtintensität konstant abnehmend
- weiß
 - Lichtintensität konstant abnehmend

Lic



isor



blaue und weiße
Taschenlampe



helle, weiße Taschenlampe

Lichtsensor

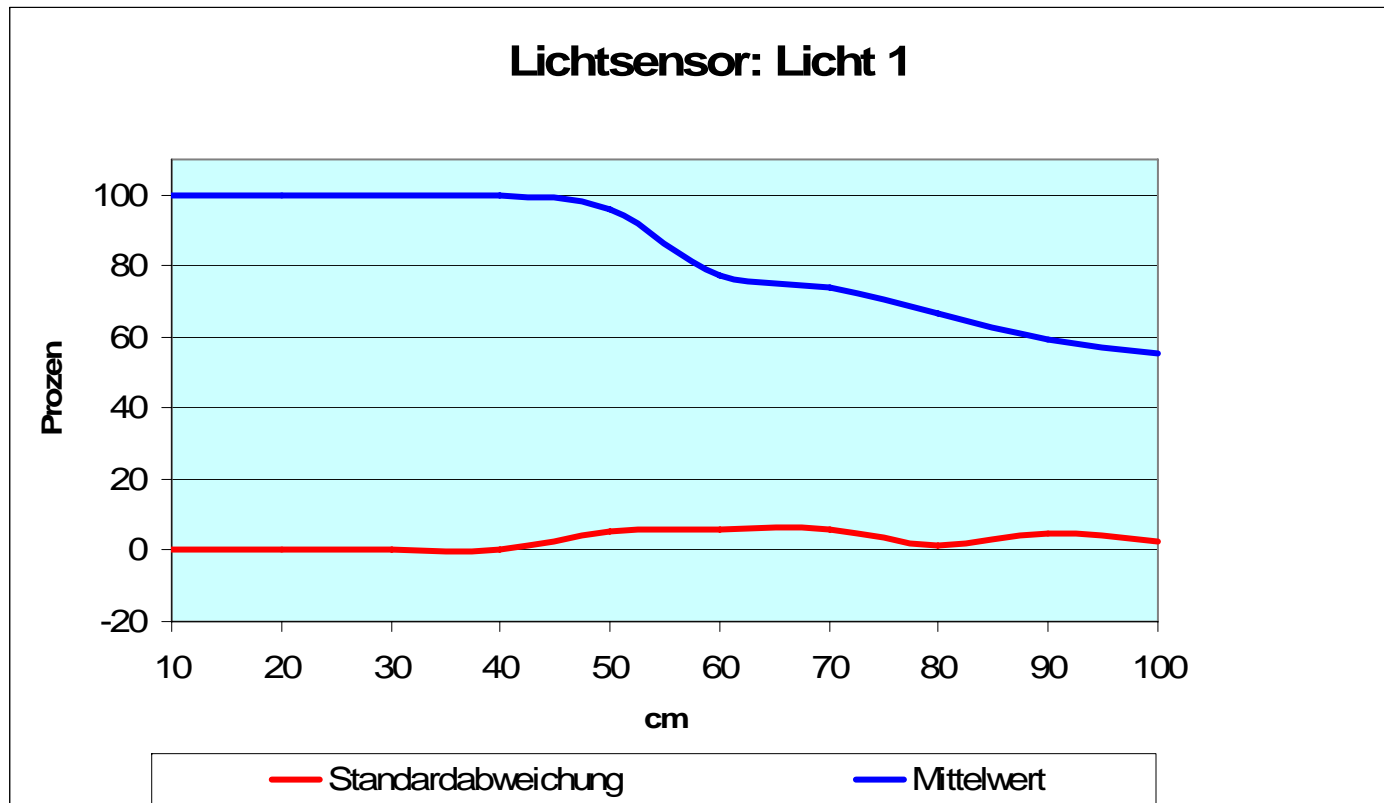


Abstand in cm	10	20	30	40	50	60	70	80	90	100
Messung 1	100	100	100	100	90	83	82	67	64	55
Messung 2	100	100	100	100	100	71	79	67	64	54
Messung 3	100	100	100	100	100	78	75	68	60	54
Messung 4	100	100	100	100	89	71	68	64	51	54
Messung 5	100	100	100	100	100	85	67	67	59	60
Mittelwert	100	100	100	100	95,8	77,6	74,2	66,6	59,6	55,4
Std.	0	0	0	0	5,15	5,85	5,91	1,36	4,76	2,33

Licht 1 (helles, weißes Licht)

Problem: bei sehr starken Lichtquellen Abstandsschätzung schwierig, da bereits ab 40 cm die Lichtintensität 100% beträgt, kein genaues Anfahren mehr möglich

Lichtsensor



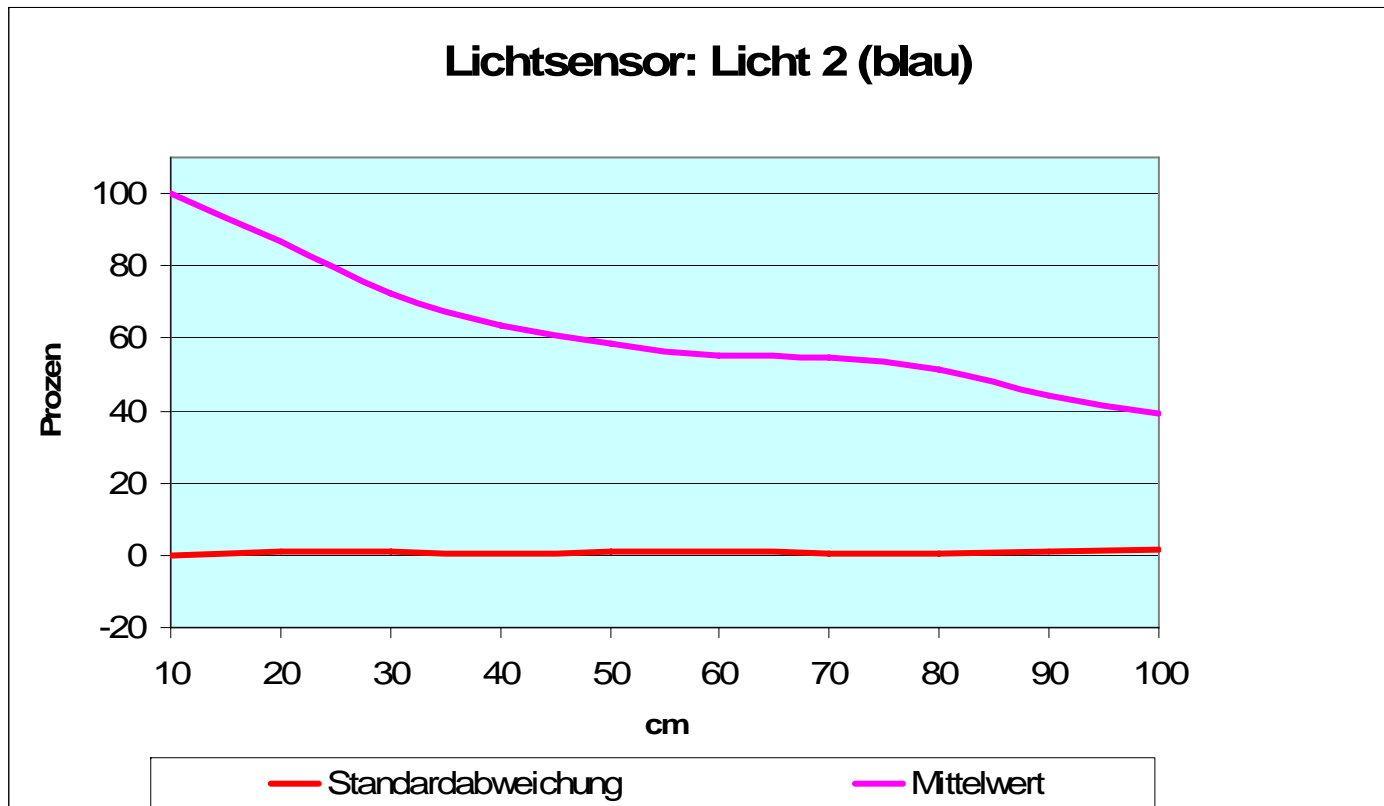
Lichtsensor



Abstand in cm	10	20	30	40	50	60	70	80	90	100
Messung 1	100	89	74	64	60	56	55	52	45	41
Messung 2	100	87	72	64	59	56	55	52	45	40
Messung 3	100	87	72	63	59	53	54	52	43	37
Messung 4	100	86	72	63	59	56	55	51	44	41
Messung 5	100	86	71	63	57	55	54	51	45	37
Mittelwert	100	87	72,2	63,4	58,8	55,2	54,6	51,6	44,4	39,2
Std.	0	1,1	0,98	0,49	0,98	1,17	0,49	0,49	0,8	1,83

Licht 2 (blaues Licht)

Lichtsensoren



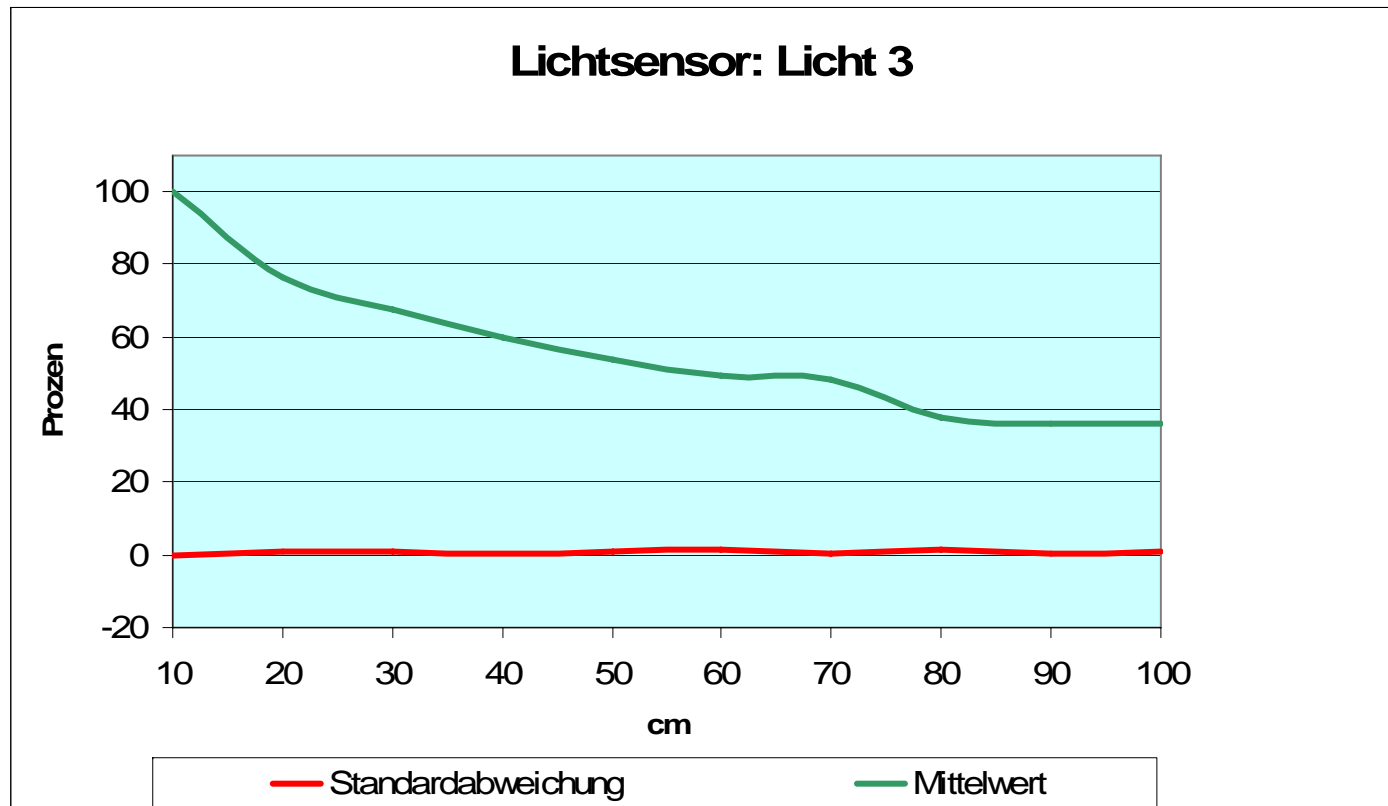
Lichtsensor



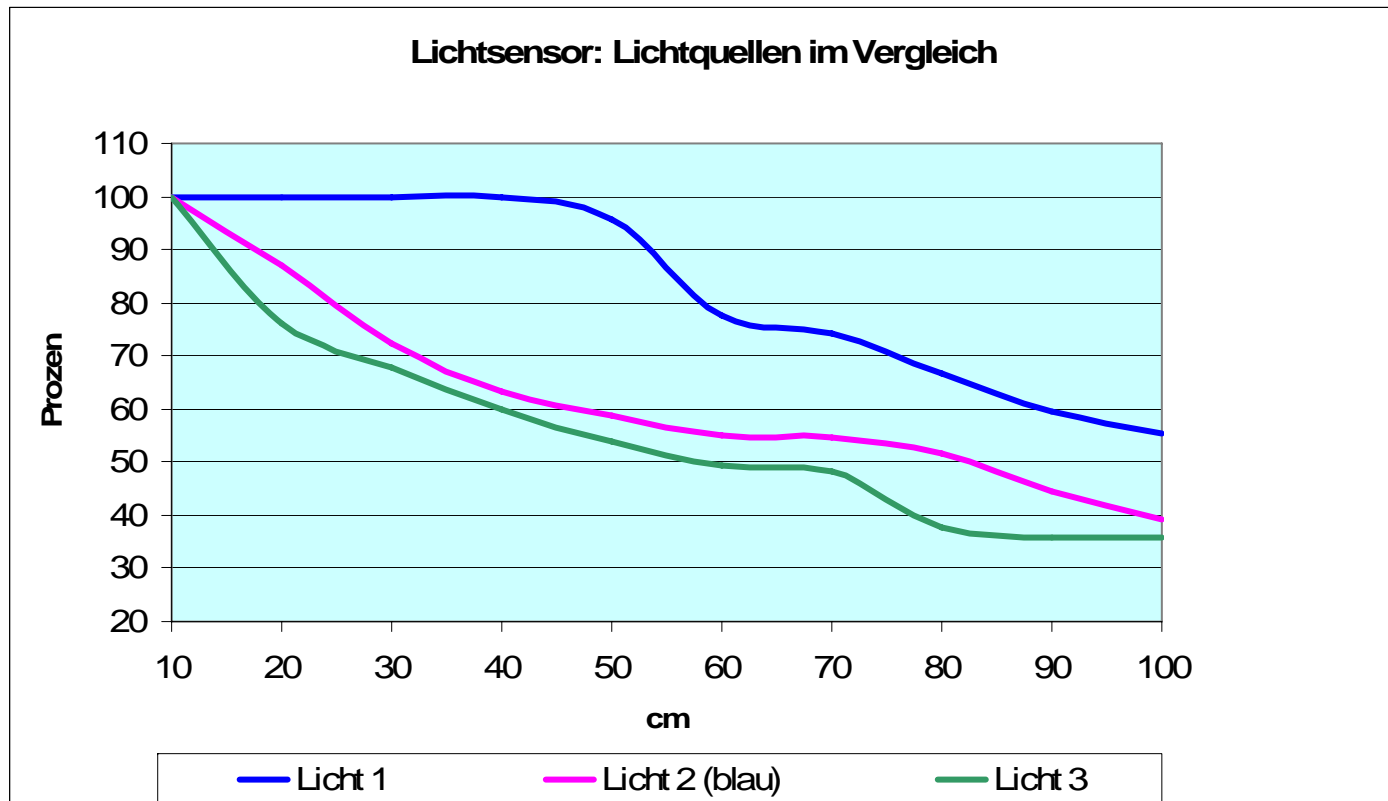
Abstand in cm	10	20	30	40	50	60	70	80	90	100
Messung 1	100	78	68	60	54	47	48	40	37	34
Messung 2	100	77	68	60	55	49	48	38	36	37
Messung 3	100	76	69	61	55	51	48	37	36	36
Messung 4	100	75	67	60	53	50	49	36	36	37
Messung 5	100	75	67	59	53	50	49	37	35	36
Mittelwert	100	76,2	67,8	60	54	49,4	48,4	37,6	36	36
Std.	0	1,17	0,75	0,63	0,89	1,36	0,49	1,36	0,63	1,1

Licht 3 (weißes Licht)

Lichtsensor



Lichtsensoren

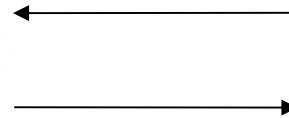




Schallsensor Echoweise

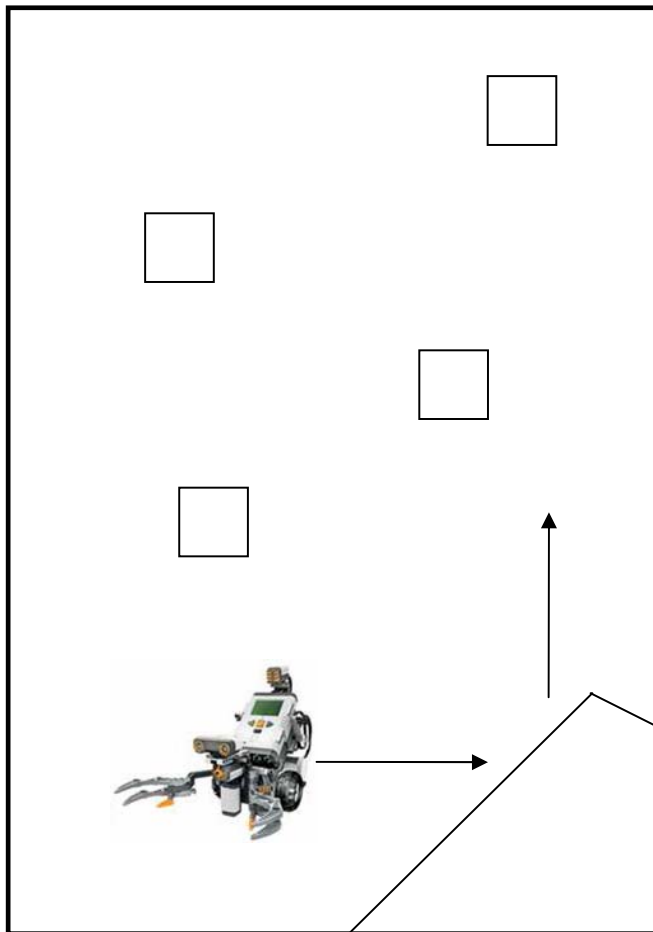


- Misst die Zeit zwischen dem Aussenden einer Schallwelle und dem Empfang des reflektierten Echos
 - Rechtes „Auge“ → Sender
 - Linkes „Auge“ → Empfänger
- Objekte rechts vom Sensor werden besser erkannt





Schallsensor ge Hindernisse



- Schallwellen werden nicht zum Empfänger zurückgesendet
- Nur bis 15° Abweichung brauchbare Ergebnisse
- Beste Messergebnisse:
 - Harte, glatte, gerade Objekte



challsensor



Materialien:



- CD-Hülle

- Porzellanteller



- Buch



challsensor



Abstand in cm	10	20	30	40	50	60	70	80	90	100
Messung 1	16	24	33	43	52	57	74	61	93	106
Messung 2	15	25	32	42	51	58	75	60	97	105
Messung 3	16	25	33	43	54	63	74	59	94	104
Messung 4	15	24	27	42	51	63	75	85	94	61
Messung 5	16	25	32	42	52	62	57	83	93	103
Mittelwert	15,6	24,6	31,4	42,4	52	60,6	71	69,6	94,2	95,8
Std.	0,49	0,49	2,24	0,49	1,1	2,58	7,01	11,8	1,47	17,4

CD-Hülle (rau):

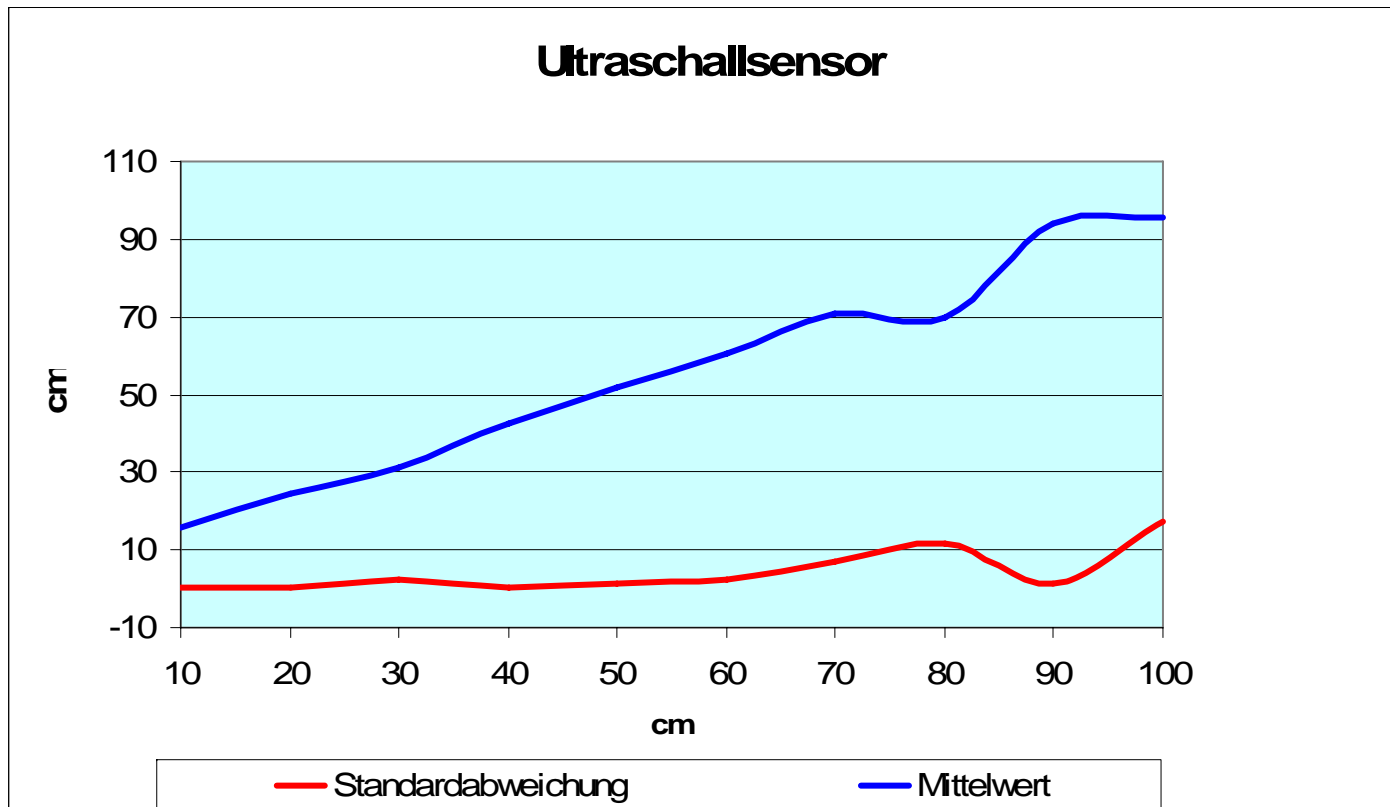
- Abstand steigt bis 70 cm konstant an
- dann Probleme bei der Wertermittlung und Zunahme von Fehlmessungen



Ultrasonic Sensor



CD-Hülle





challsensor



Abstand in cm	10	20	30	40	50	60	70	80	90	100
Messung 1	17	25	34	44	54	65	74	85	97	106
Messung 2	16	26	33	45	57	66	76	84	96	107
Messung 3	17	26	34	44	55	64	73	85	98	106
Messung 4	15	26	34	43	54	64	75	84	97	104
Messung 5	16	25	35	44	54	65	74	86	96	107
Mittelwert	16,2	25,6	34	44	54,8	64,8	74,4	84,8	96,8	106
Std.	0,75	0,49	0,63	0,63	1,17	0,75	1,02	0,75	0,75	1,1

Porzellanteller:

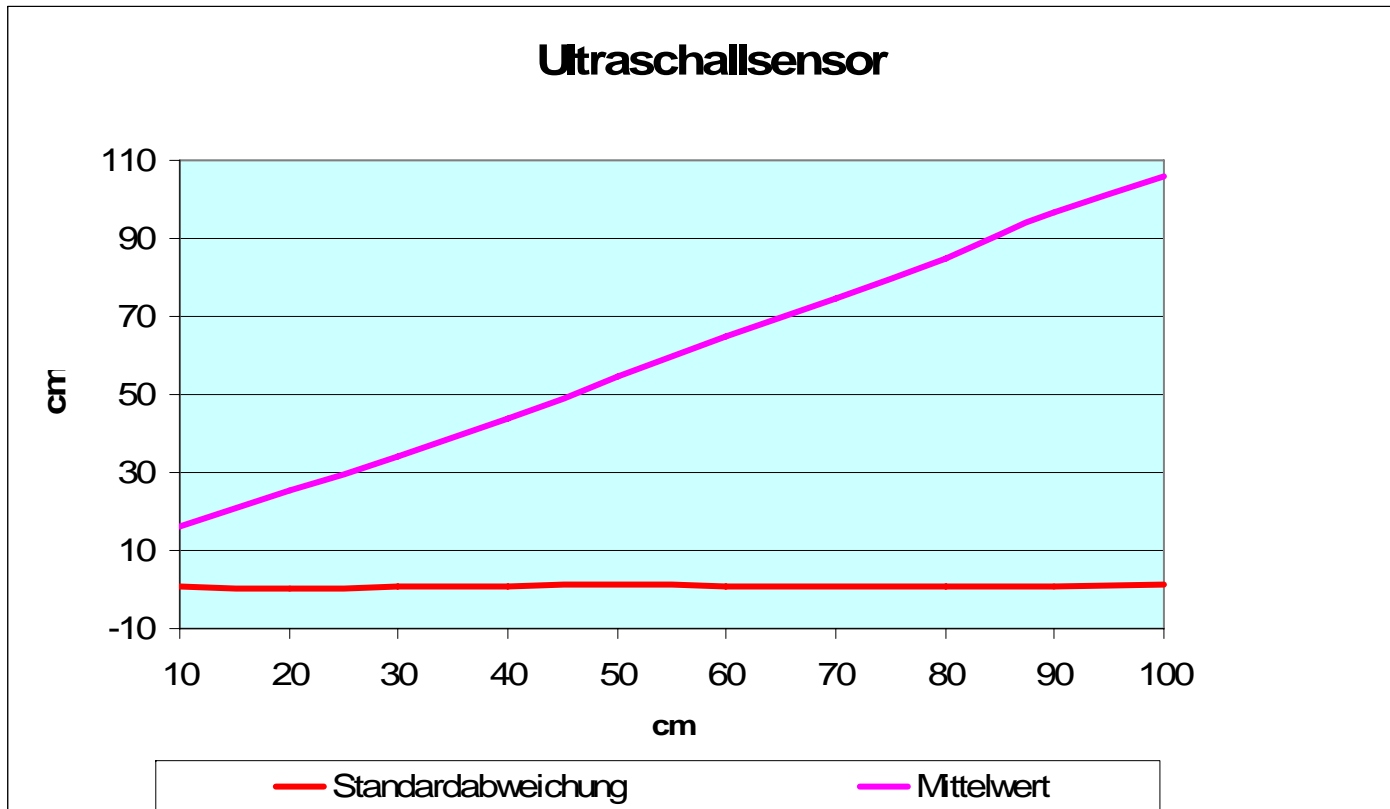
– gemessener Abstand steigt konstant an



challsensor



Porzellanteller





challsensor



Abstand in cm	10	20	30	40	50	60	70	80	90	100
Messung 1	15	24	32	42	52	63	73	85	95	104
Messung 2	16	25	31	41	52	63	74	84	96	106
Messung 3	14	25	33	43	53	63	73	84	95	104
Messung 4	15	24	33	42	53	62	79	86	96	103
Messung 5	16	24	32	42	52	63	84	85	94	104
Mittelwert	15,2	24,4	32,2	42	52,4	62,8	76,6	84,8	95,2	104
Std.	0,75	0,49	0,75	0,63	0,49	0,4	4,32	0,75	0,75	0,98

Buch:

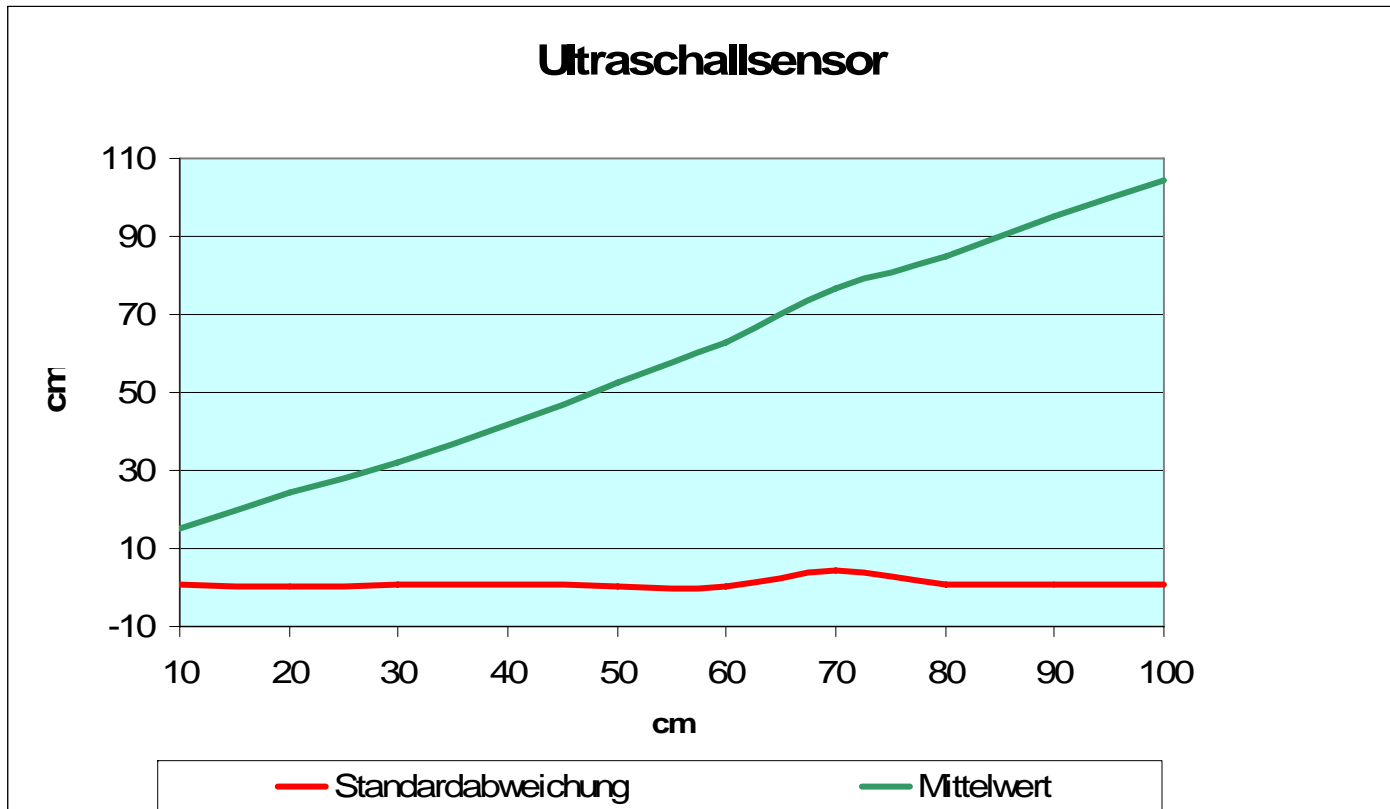
– gemessener Abstand steigt konstant an



challsensor

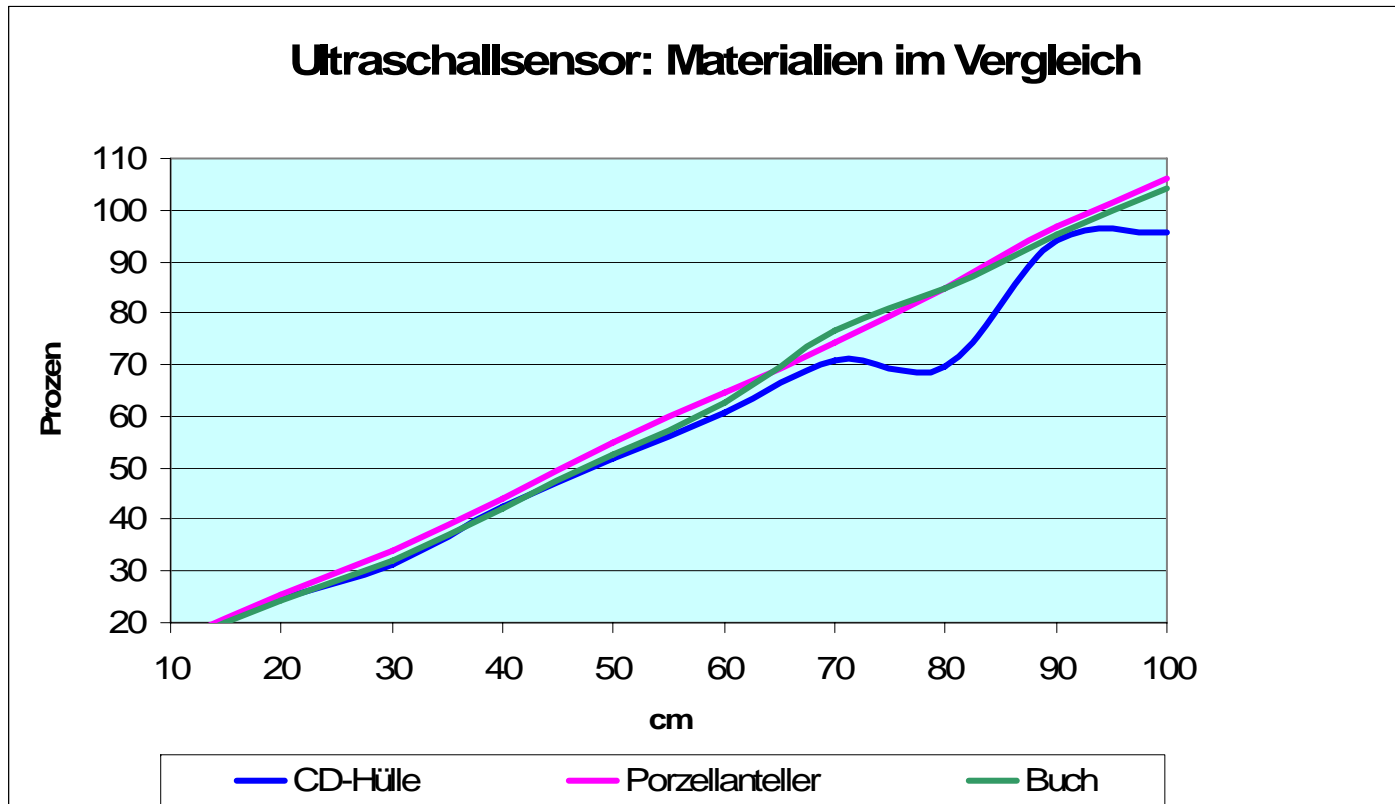


Buch





challsensor



Ausblick

- Konstruktion:
 - Bewegliches Rad anstatt Stützbalken
 - Räder direkt an Zentraleinheit befestigen
 - Mobile Sensoren
- Geräuschsensor:
 - Funktionalität des Geräuschsensors
 - Anweisungen durch Geräusche
- Tastsensor:
 - Leichter auslösbar

Quellen

- **Wikipedia, die freien Enzyklopädie**
<http://de.wikipedia.org/wiki/NXT> (Stand: 11:34 Uhr, 18.04.07)
- **Die NXTe Ebene – Januar**
*http://dienxteebene.blogspot.com/2007_01_01_archive.html
(Stand: 19:25 Uhr, 12.02.07)*
- **NXT Programming Software**
*<http://www.teamhassenplug.org/NXT/NXTSoftware.html>
(Stand: 16:52 Uhr, 12.02.07)*
- **Lego mindstorms education**
<http://www.nxt-in-der-schule.de/> (Stand: 18:05 Uhr, 07.05.07)
- **Deutsche NXT-FAQ**
<http://lego.brandls.info/nxtfaq/> (Stand: 15:37 Uhr, 28.05.07)

Fragen???

Vielen Dank für Ihre Aufmerksamkeit.


```

#include "NXCDefs.h"

// Variablen für Sensorenwerte
int tast = 0;           // Tastsensor
int licht;              // Lichtsensor
int ultra = 0;         // Ultraschallsensor
int test;              // Einzelablauf
int THRESHOLD;         // Schwellenwert für Licht
int ABSTAND = 30;      // Abstandswert für Ultraschall
int durchlauf;         // Durchlauf für Suche
mutex reg;             // Taskregulierung

int sem;               // Tasks aufeinander warten lassen
int oeffnen=0;         // ist Arm offen(1) oder zu(0) oder Kugel gefunden(2)
int starten=0;         // in der main einmal ausführen
int Stoppen=0;         // auf Druck warten
long timeout;          // CurrentTick() --> auf Druck warten

void init_sensoren(){ // Sensoren initialisieren
  // Sensortyp und Anschlußport festlegen
  SetSensor(IN_1,SENSOR_TOUCH);           // Tastsensor
  SetSensorType(IN_2,IN_TYPE_LOWSPEED_9V); // Ultraschallsensor
  ResetSensor(IN_2);
  SetSensorType(IN_3,IN_TYPE_LIGHT_INACTIVE); // Lichtsensor
  SetSensorMode(IN_3,IN_MODE_PCTFULLSCALE);
  ResetSensor(IN_3);
  SetSensor(IN_4,SENSOR_TOUCH);           // Tastsensor Ball
}

```

```
task fahre_vorw(){
    while (true){
        until (Stoppen==0);
        until (sem == 0); sem = 1;
        OnFwd(OUT_BC, -25); sem=0; Wait(50);
    }
}
```

```
void fahre_zurueck(){
    OnFwd(OUT_BC, -20); Wait(2000);
}
```

```
void hindernis(){
    OnFwd(OUT_BC, 50); sem=2; Wait(500);
    OnFwd(OUT_C, -50); sem=2; Wait(500);
}
```

```
void drehen(){
    OnFwd(OUT_C, 50); sem=2; Wait(600);
    OnFwd(OUT_BC, 40); sem=2; Wait(600);
}
```

```

task check_sensors(){
  while (true){
    until ((sem == 0) || (sem == 2)); sem = 2;
    ultra = SensorUS(IN_2);
    if (SENSOR_1 == 1){
      fahre_zurueck(); sem=0;
    }
    if (ultra <= ABSTAND){
      hindernis(); sem=0;
    }else {
      // wenn Ball tastsensor2 berührt und Arm offen, Arm schließen
      if ((SENSOR_4==1) && (oeffnen==1)){
        //Arm schließen
        // Stoppen
        Stoppen=1;
        timeout = CurrentTick();
        RotateMotor (OUT_A, 10, 90); Wait(100);
        OnFwd(OUT_A,0);
        Off(OUT_BC);
        until ((CurrentTick()-timeout) > 20000);
        Stoppen=0;
        oeffnen=2;
        licht = SENSOR_3;
        THRESHOLD = licht+20; sem=0;
      }
    }
  }
}

```

```

// Licht suchen
if(SENSOR_3 <= THRESHOLD){
  if (durchlauf<5 && SENSOR_3 <= THRESHOLD && ultra > ABSTAND){
    drehen();
    durchlauf=durchlauf+1; sem=0;
  }else{
    if (ultra <= ABSTAND){
      hindernis(); sem=0;
    }else{
      OnRev(OUT_BC, 40); sem=2; Wait(1000);
      durchlauf=Random(6)+1; sem=0;
    }
  }
}
}else{
  if (ultra <= ABSTAND){
    hindernis(); sem=0;
  }else{
    // vorwärts fahren --> heller
    OnFwd(OUT_BC, -30); sem=2; Wait(20);
    //Kugel fast erreicht -->Arm öffnen, wenn noch nicht offen
    if (oeffnen==0){
      RotateMotor (OUT_A, 10, -90);
      OnFwd(OUT_A,0);
      oeffnen=1;sem=0;
    } sem=0;
  } sem=0;
}
} sem=0;
}
}

```

```
task main(){
  if (starten==0){
    sem=0;
    starten=1;
    Stoppen=0;
    init_sensoren();
    licht = SENSOR_3;
    durchlauf = Random(6)+1;
    THRESHOLD = licht+20;
  }
  Precedes(check_sensors, fahre_vorw);
}
```