

HAUPTSEMINAR AUTOMATISIERUNGS-, MESS- UND REGELUNGSTECHNIK: ANLEITUNG UND AUFGABENSTELLUNG MODUL PERCEPTION

VERSION 2.2

1 Inhaltsverzeichnis

1	Inhaltsverzeichnis	1
2	Allgemein.....	1
3	Struktur der Perception.....	1
4	Vorbereitung	3
5	Aufgaben	8
6	Zielstellug 1. Verteidigung	10
7	Anhänge	10

2 ALLGEMEIN

Die Perception umschließt die Erfassung von Messdaten, ihre Vorverarbeitung sowie Bereitstellung an die angrenzenden Module. Dazu müssen die bereitstehenden Sensoren angesteuert und kalibriert werden. Um die Messdaten bewerten und geeignet fusionieren zu können, muss eine Aussage über ihre Zuverlässigkeit getroffen werden. Hierfür sind die Messunsicherheiten sowie die Einflussfaktoren darauf zu bestimmen. Anschließend soll untersucht werden, wie die Messunsicherheit der Positionsbestimmung des zur lokalen Navigation eingesetzten Maussensors verringert werden kann.

3 STRUKTUR DER PERCEPTION

Dem NXT-Roboter stehen 4 Arten von Sensoren zur Verfügung, diese könne in NXT-eigene und externe Sensoren unterschieden werden:

- 2 NXT Lichtsensoren: Sie messen die Helligkeit des Untergrundes. So können sie Fahrbahnmarkierungen detektieren und zur Spur- bzw. Linienverfolgung genutzt werden.
- 2 NXT Radencoder: Sie messen die Drehwinkeländerung der Räder und können so zur Geschwindigkeitsmessung genutzt werden.
- 1 Korrelationssensor (optische Maus) kann zur Orientierung des Fahrzeugs (Erkennung von Geschwindigkeit und Bewegungsrichtung) dienen (die lokale Navigation)
- 4 Triangulationssensoren: Sie messen Abstände und sollen so am Roboter installiert werden, dass sie u.a. zur Kollisionsvermeidung bzw. zur Erkennung von Begrenzungswänden genutzt werden können.

Die NXT-eigenen Sensoren können direkt an den NXT angeschlossen werden. Da der NXT nicht ausreichend passende Schnittstellen bietet und um Rechenleistung auszulagern, sollen die Sensordaten der externen Sensoren von einem Arduino UNO Mikrocontroller (μC) aufgenommen, weiterverarbeitet und dem NXT über eine Serielle RS485 Schnittstelle bereitgestellt werden (vgl. Abbildung 1). Der Arduino läuft unter der struktureigenen Software Arduino. Dafür ist bereits das Programm NXT_Laser.ino realisiert. Dieses liest die

analogen Triangulationssensordaten sowie digitalen Mausdaten zyklisch aus und stellt sie dem NXT über dessen serielle RS485 Schnittstelle zur Verfügung, sodass für das Funktionieren des Roboters kein weiterer Arduino-Code erzeugt werden muss. Über die USB – Schnittstelle des Arduino Uno, kann dieser mit dem PC verbunden werden. Zu Testzwecken lassen sich die Sensordaten in der Konsole der Arduino Programmierungsumgebung ausgeben. Zur Kalibrierung der Sensoren steht das Programm Laser_Kalib.ino bereit. Hiermit können alle Sensorwerte ausgelesen werden. Eine kurze Dokumentation der möglichen Befehle befindet sich im Kopf der des Programmcodes. Zum Ausführen eines neuen Befehls muss die entsprechende Zahl in der Eingabezeile der Arduino-Konsole eingegeben werden und mit der Enter-Taste bestätigt werden.

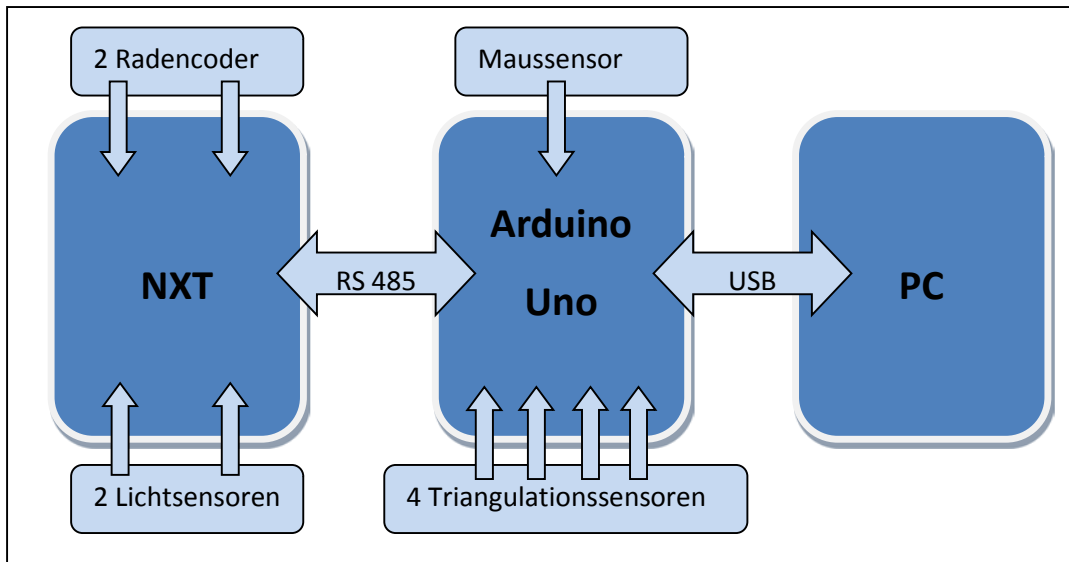


ABBILDUNG 1: HARDWARE STRUKTUR

Der NXT Controller wird mit dem Java-basierten Betriebssystem LeJOS betrieben. Jeder Aufgabenbereich (Perception, Guidance, Navigation, Regelung und Bedienschnittstelle) wird dabei in einem eigenen Modul realisiert. Die Kommunikation zwischen den Modulen erfolgt über die get-Methoden in den zugehörigen Interfaces, diese sind bereits vorgegeben, sodass die einzelnen Aufgabenbereiche weitestgehend unabhängig voneinander realisiert werden können. Das Interface IPerception.java ist bereits vollständig in der Klasse Perception.java implementiert, sodass im Aufgabenbereich Perception kein weiter Java-Code für das Funktionieren des Roboters erzeugt werden muss.

Zur Information folgt ein kurzer Überblick über die implementierten get-Methoden. Für ausführlichere Informationen machen Sie sich bitte mit dem Javadoc vertraut, da Sie der Ansprechpartner für Ihre Mitarbeiter zu Fragen zum Perception Modul sind.

- *line sensor (Lichtsensor):*
 - existiert in 2 Varianten:
 - 1. Stellt 2 ternäre Werte bereit (rechter Sensor, linker Sensor; ob sich Sensor auf der Linie, auf dem Rand oder neben der Linie befindet). Dient zur Implementierung einer einfachen Linienverfolgung.
 - 2. Stellt 2 Integerwerte zwischen 0 und 100 (relative gemessene Helligkeit in % beider Sensoren) für komplexere Linienverfolgungsalgorithmen zur Verfügung.
 - Die Initialisierung, Kalibrierung und Auswertung der Messungen wurde unter Zuhilfenahme der existierenden LeJOS LightSensor-Klasse implementiert.
- *Encoder (Radencoder):*

- Stellt 2 Winkeldifferenzen in Grad (rechter und linker Sensor) zwischen aktueller und letzter Abfrage sowie die Zeitdifferenz zur letzten Abfrage in ms bereit.
- Da die Daten von verschiedenen Modulen abgefragt werden können, werden für jedes Modul eigene Objekte erzeugt in denen die entsprechenden Winkel und Zeitdifferenzen übergeben werden.
- Die Initialisierung und Auswertung der Messungen erfolgte unter Zuhilfenahme der existierenden LeJOS NXTMotor-Klasse.
- *distance sensor (Triangulationssensoren):*
 - Stellt die von den vier Triangulationssensoren (vorwärts vorn, seitlich vorn, seitlich hinten, rückwärts hinten) gemessenen Abstände in mm bereit.
 - Die Aufnahme und Auswertung der Messdaten sowie die Kalibrierung der Sensoren erfolgt über den externen μ C.
- *mouse odometry (Maussensor):*
 - Stellt die gemessene Verschiebung zwischen aktueller und letzter Abfrage in mm und die zugehörige Zeitdifferenz in ms bereit. Da es sich auch hier um eine Differenzgröße handelt, werden für jedes Modul eigene Objekte erzeugt in denen die entsprechenden Verschiebungen und Zeitdifferenzen übergeben werden.
 - Die Aufnahme und Auswertung der Messdaten sowie die Kalibrierung des Sensors erfolgt über den externen μ C.

Zur Bewertung der übergebenen Daten benötigt das Navigationsmodul die Messunsicherheiten (MU) für die einzelnen Sensoren. Diese sollen dem Navigationsmodul offline zur Verfügung gestellt und müssen daher bestimmt werden. Zur Bestimmung der MU sollen die Aufgaben unter 4. durchgeführt und die Ergebnisse der Gruppe mitgeteilt werden. Zur Durchführung dieser Aufgaben sind zunächst einige Vorbereitungen zu treffen.

4 VORBEREITUNG

I Installation der notwendigen Software

An der Professur für Mess- und Sensorsystemtechnik steht ein Arbeitsplatz mit PC, der notwendigen Software (Eclipse, LeJos, Arduino, Matlab) und ein Kalibrierstand bereit. Für die **Arbeit zu Hause** sind folgende Programme zu installieren:

- a Eclipse mit dem PlugIn LeJOS (entsprechend der Allgemeinen Anleitung)
- b Arduino [1]: Entpacken Sie den Ordner Hauptseminar in den entpackten Arduino-Ordner (ins Unterverzeichnis libraries!)

Die bereitgestellten Programme: NXT_Laser.ino und Laser_Kalib.ino lassen sich über Datei → Beispiele → Hauptseminar öffnen.
- c Matlab: Eine CD/DVD mit der Software Matlab erhalten Sie im Sekretariat des Institutes für Automatisierungstechnik bei Frau Möge (Raum E3).
- d Machen Sie sich mit der grundsätzlichen Softwarearchitektur vertraut.

II Aufbau der Hardware

Die externen Sensoren können nicht direkt an den NXT angeschlossen werden. Daher sollen sie über einen zweiten Mikrocontroller (μ C) mit dem NXT verbunden werden. Der dafür verwendete μ C (Arduino UNO[2]) wird zusätzlich genutzt, um eine Datenvorverarbeitung durchzuführen. Im Folgenden wird kurz beschrieben, wie beim Aufbau der Verbindung der externen Hardware vorzugehen ist.

Arduino Proto Shield

Der Arduino dient zum Auslesen der Sensordaten, ihrer Vorverarbeitung und Bereitstellung für den NXT. Dazu muss er mit allen externen Sensoren (optische Maus sowie 4 Triangulationssensoren) und dem NXT verbunden werden. Um entsprechend Platz für alle notwendigen Verbindungen zu schaffen, ist der Arduino mit einem Shield erweitert. Dieses ist dazu auf den Arduino aufgesteckt. Über Steckerleisten ist sind die Aktiven Pins des Shields und des Arduino zu verbinden.

Verbindung zum NXT

Zur Kommunikation zwischen NXT und Arduino wird auf Arduinoseite eine serielle RS485 Schnittstelle genutzt. Der NXT bietet dafür keinen nativen Support. Um den Arduino über RS485 mit dem NXT zu verbinden, ist dieser um einen entsprechenden Transceiver zu erweitern. Dafür ist der MAX485 von Maxim [5] zu verwenden.

Auf Verbindung auf Arduinoseite ist hierfür:

- Digital 11 – Ro
- Digital 12 – Re und De
- Digital 13 – Di

(vgl. **Abbildung 2**). NXT seitig wird zur Verbindung ein Standard – NXT – Sensorkabel verwendet. Pinbelegung des Kabels lautet [4]:

- Pin1 Weiß ANA Analog
- Pin2 Schwarz GND (-)
- Pin3 Rot GND (-)
- Pin4 Grün +4,3V
- Pin5 Gelb SCL (digital0)
- Pin6 Blau SDA (digital1).

Dabei ist der Digitalpin0 (gelbes Kabel) mit dem A-PIN des MAX485 und der Digitalpin1 (blaues Kabel) mit dem B-PIN verbunden. Das Sensorkabel muss am Sensorport 4 des NXTs angeschlossen werden, da nur dieser eine RS485-Schnittstelle bereitstellt [4]. In **Abbildung 2** und **Abbildung 3** ist die Verdrahtung dargestellt.

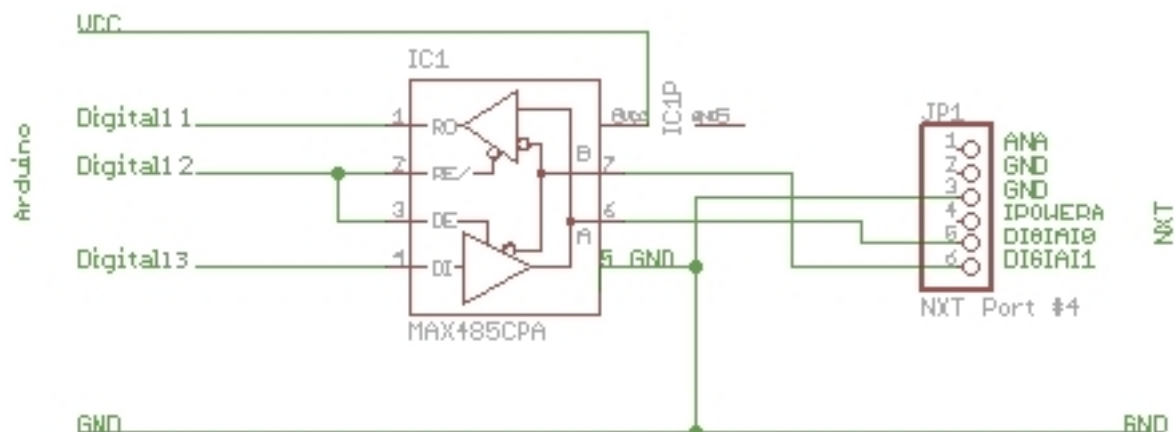


ABBILDUNG 2: VERBINDUNG ARDUINO – NXT ÜBER MAX 485

Stromversorgung

Der Arduino und die angeschlossenen Sensoren können mit 2 Arten mit Strom versorgt werden. Entweder über USB durch den PC oder von Batterie (9V). Für den Batteriebetrieb ist kann ein kleiner Schalter aufgelötet werden. Beachten Sie, die Versorgung von der Batterie abzuschalten, falls der Arduino über USB versorgt wird oder nicht benötigt wird.

Triangulationssensoren

Die Triangulationssensoren (GP2Y0A21YK0F von Sharp[3]) liefern entsprechend der gemessenen Entfernung eine analoge Spannung. Die Kabel der Triangulationssensoren sind farbkodiert. Rot entspricht der Spannungsversorgung, Schwarz entspricht Ground und Gelb dem analogen Ausgang. Zur Nutzung des bereitgestellten Arduino Codes sind die analogen Ausgänge der Sensoren mit den Analogpins des Shields 1 bis 4 zu verbinden (vgl. **Abbildung 3**).

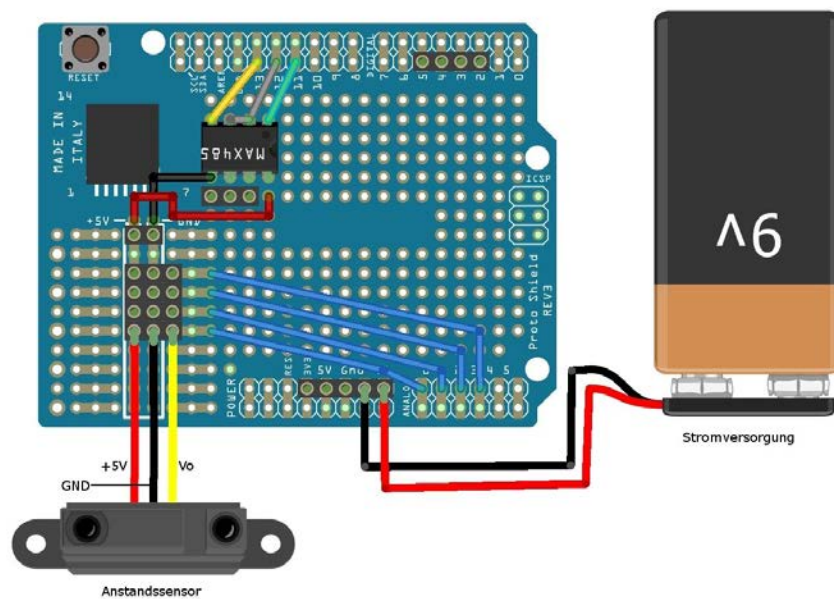


ABBILDUNG 3: VERDRAHTUNG SHIELD

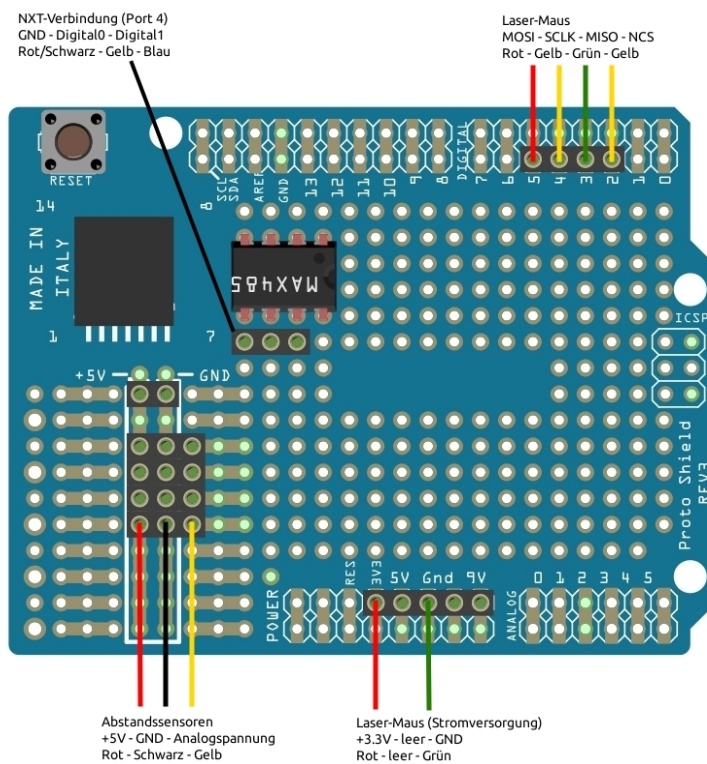


ABBILDUNG 4: ANSCHLUSSPLAN BEI VERWENDUNG DER LASER-MAUS

Der Maussensor

In der bereitgestellten Lasermouse findet der Sensor ADNS-7550 von AVAGO Verwendung. *Es sind (sofern noch nicht geschehen) folgende Pins mit dem Arduino zu verbinden:*

Pin des Sensors	Port des Ardino	Kabelfarbe
Supply Voltage (3.3V)		rot
Ground		grün
NCS (Chip Select, Low-aktiv)	2	gelb
MISO (Serial Port Data, Output)	3	grün
SCLK (Serial Port Clock)	4	gelb
MOSI (Serial Port Data, Input)	5	Rot

Der Anschluss erfolgt nach **Abbildung 3** und **Abbildung 4**. Die Spannungsversorgung ist über einen gesonderten Steckverbinder sicher zustellen.

Weitere Hinweise zum Aufbau der Externen Hardware:

Um den Arduino bei Fehlern neu zu starten, kann neben dem Resettaster auf der Arduino-Platine auch einer auf dem Shield aufgelötet werden (vgl. Abbildung 4). Der Taster kann der Maus entnommen werden.

An der Professur f. Mess- und Sensorsystemtechnik befindet sich eine Elektronik Werkstatt mit der notwendigen Ausrüstung, um den Arduino aufzubauen. Mechanische Komponenten können in der Diplomantenwerkstatt gefertigt werden.

Achtung!: Der Zugang zur Elektronikwerkstatt und zum Kalibrierstand ist zeitlich auf 9:00 bis 18:00 Uhr beschränkt und erfolgt ausschließlich auf Rücksprache mit dem zuständigen Betreuer.

5 AUFGABEN

Die Durchführung aller Aufgaben ist schriftlich in angemessener Form zu dokumentieren. Die Aufgaben sind nicht zwangsläufig in der vorgegeben Reihenfolge zu bewerkstelligen. Ihre Kommilitonen benötigen zunächst vor allem die Messwerte und Kalibrierfunktionen zur Bewerkstellung Ihrer Aufgaben, so dass der Aufbau des Roboters und die Sensorkalibrierung zuerst zu erledigen sind.

Beschreiben Sie kurz das Funktionsprinzip der 4 eingesetzten Sensoren (in Summe maximal 3 Seiten inklusive Abbildungen). Welche Vor- und Nachteile ergeben sich aus den Funktionsprinzipien für die Zuverlässigkeit der ausgegeben Werte. Welche alternativen Messprinzipien existieren, zum Erfassen der jeweiligen Messgrößen? Welche Parameter beeinflussen die systematischen und zufälligen Messunsicherheiten (MU) der auszugebenden Werte.

I Untersuchung der Sensoren

a Lichtsensor:

- i Bestimmen Sie die zufällige Messunsicherheit der Lichtsensoren. Positionieren Sie die Lichtsensoren dazu wie auch beim Roboter üblich wenige mm über dem Untergrund. Nehmen Sie jeweils mindestens 15 Helligkeitswerte für verschiedene, **sinnvolle** Untergründe auf.
- ii Überprüfen Sie den Einfluss der Zimmerbeleuchtung auf die gemessenen Werte. Was folgt daraus für den Aufbau des Roboters?
- iii Legen Sie geeignete Schwellwerte für die Linienverfolgung fest (neben der Linie, auf der Linie, auf der Kante der Linie) und tragen Sie diese im Modul Perception ein.

b Encodersensor:

- i Beschreiben Sie den Zusammenhang zwischen Inkrementanzahl des Encoders und der MU für die Winkelmessung [7].
- ii Welche Faktoren neben der MU für den Drehwinkel beeinflussen die MU bei Bestimmung der Position?
- iii Was folgt daraus für den Aufbau des Roboters. Welche Anforderungen ergeben sich an die Steuerung des Roboters?

c Triangulationssensor

- i Kalibrieren Sie die Triangulationssensoren. Nehmen Sie dazu in 1cm Abständen die Spannungswerte mit dem Programm Laser_Kalib.ino auf und führen Sie eine Regression mit einem Polynom n-ten Grades durch (z.B. mit MatLab oder Excel). Hinterlegen Sie die Polynomkoeffizienten in NXT_Laser.ino.
- ii Wiederholen Sie die vorherige Messung 15-mal und bestimmen Sie die Standardabweichung. Messen Sie 15 mal in Folge an jeder Position und bestimmen Sie die Standardabweichung. Vergleichen Sie die ermittelten Standardabweichungen!
- iii Welche Schlussfolgerungen sind aus den Kalibrierfunktionen auf die Messbereiche der Sensoren zu schließen. Was folgt daraus für den Aufbau des Roboters?

d Maussensor

- i Kalibrieren Sie den Maussensor mit dem Programm Laser_Kalib.ino. Verfahren Sie den Sensor dazu von Hand am Kalibrierstand 15-mal jeweils 25cm vorwärts und 15-mal jeweils 25cm seitwärts. Tragen Sie die Kalibrierkoeffizienten in NXT_Laser.ino ein.

- ii Bestimmen Sie die systematische und zufällige MU durch Wiederholungsmessungen bei Variation folgender Parameter:
 - 1 Untergrund: mindestens drei **sinnvolle** Untergründe.
 - 2 Sensorhöhe: auf einem der Untergründe mit den gegebenen Halterungen
 - 3 Geschwindigkeit: auf einem der Untergründe mit Geschwindigkeiten von etwa 25cm/**1s** , 25cm/**2s** , 25cm/**5s** , 25cm/**10s**
- iii Bewerten Sie die erzielten Ergebnisse und vergleichen Sie sie mit denen der Encodersensoren. Was folgt aus den Ergebnissen für den Einsatz der Lasermaus zur Odometrie?
- iv Wie lässt sich die Messunsicherheit des Lasermaussensors reduzieren?

Hinweis: Je nach Untergrund kann es notwendig sein den Laserdiodenstrom der Maus anzupassen. Die benötigten Befehle sind im Kopf des Programms Laser_Kalib.ino beschrieben.

- II Adaptieren Sie die Sensoren in geeigneter Weise an das Fahrzeug und testen Sie die Sensoren im Betrieb (Ansprechen, Datenauslesen, Unsicherheit der Messwerte, hard- und softwareseitiger Anbindung in die Fahrzeugsteuerung)
- III Welcher Sensor stellt den größten Unsicherheitsbeitrag für die Navigation dar? Erarbeiten Sie ein Konzept zur Verbesserung dieses Sensors und setzen Sie diese um!

6 ZIELSTELLUNG 1. VERTEIDIGUNG

Zur 1. Verteidigung soll der Roboter auf dem Parcours nach Parklücken suchen und somit einen gewissermaßen reduzierten Scout Modus unterstützen. Sowohl die Lokalisierung als auch die Parklückendetektion sollen dabei mit einem Basisfunktionsumfang und Grundleistungen präsentiert werden. Dafür ist es notwendig, dass die Perception mindestens die Sensorwerte der Triangulationssensoren, Lichtsensoren und Radencoder bereitstellt. Im Rahmen der Präsentation sollen daher die Vorgehensweise zur Kalibrierung genannter Sensoren und die Ergebnisse der Kalibrierung vorgestellt werden.

7 ANHÄNGE

- [1] <http://arduino.cc/en/Guide/Windows>
- [2] <http://arduino.cc/en/Main/ArduinoBoardUno>
- [3] Datenblatt GP2Y0A21YK0F
- [4] LEGO MINDSTORMS NXT Hardware Developer Kit
- [5] Datenblatt RS 485RS 422 Transceivers
- [6] Vorlesungsunterlagen Messtechnik I +II
- [7] <http://www.philohome.com/nxtmotor/nxtmotor.htm>
- [8] Datenblatt ADNS 7550