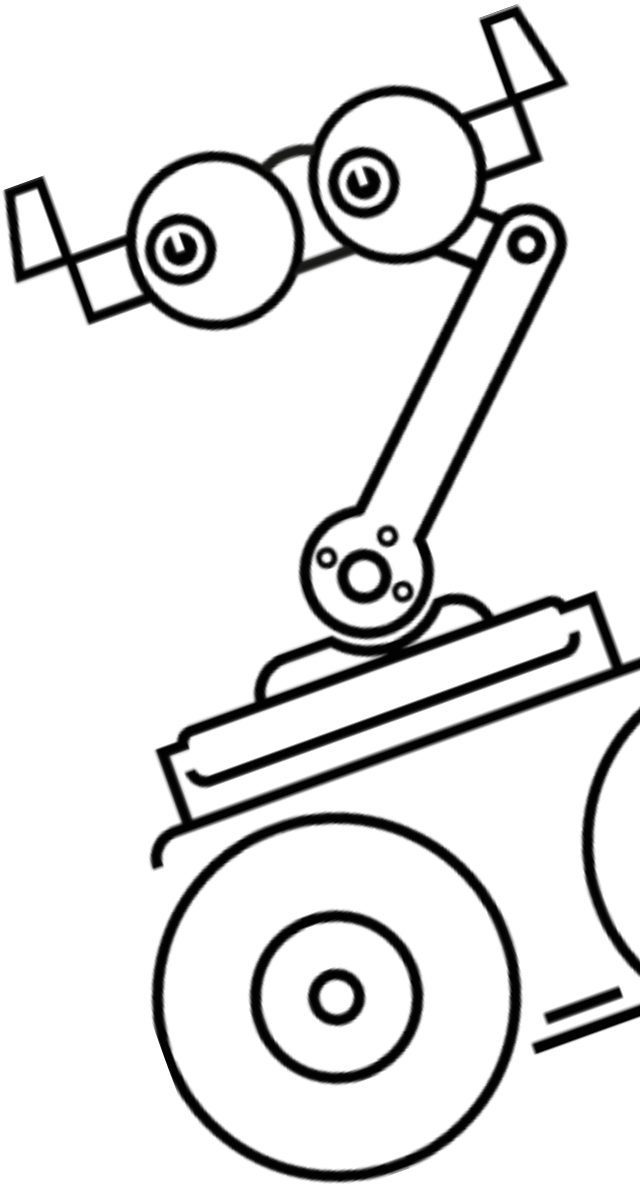
Sichere Architektur autonomer mobiler Systeme

Roberta-Experiment mit dem LEGO Mindstorms NXT



ROBERTA
INITIATIVE

Roberta ist ein eingetragenes Warenzeichen der Fraunhofer-Gesellschaft e.V.
Roberta ist seit 2010 Mitglied der Fraunhofer Academy

Inhalt

1. Einführung von Sicherheitsanforderungen, Richtlinien, Normen 3
2. Experiment »Safe Mobile Transport Robot (SMTR)« 20
3. Zusammenfassung 43
Literaturverzeichnis 44
Kontakt 46

1. Einführung von Sicherheitsanforderungen, Richtlinien, Normen

Einleitung

Lange Zeit wurden Roboter hauptsächlich in der Industrie verwendet, um Arbeitsschritte automatisiert ablaufen zu lassen. Um Unfälle zu vermeiden arbeiten Industrieroboter üblicherweise in fest definierten Arbeitsbereichen, welche durch passive (Schutzzäune, Abdeckungen etc.) und aktive (z. B. Überwachungssensoren) Sicherheitsvorkehrungen gesichert sind. Der Trend beim Einsatz von Robotern geht jedoch stark in Richtung Serviceroboter. Ein Indiz hierfür sind unter anderem die hohen Verkaufszahlen des iRobot Roomba-Staubsaugerroboters, welche bereits im Oktober 2004 die 1 Millionen Marke überschritten haben.[1] Derzeit werden mobile autonome Roboter vor allem für Flächenreinigungsaufgaben und Transportaufgaben sowie als Informationsroboter verwendet. Zwar hat jeder dieser Roboter unterschiedliche Aufgaben, die er zu bewältigen hat, doch im Grunde haben sie eine Gemeinsamkeit, sie interagieren direkt mit Menschen und arbeiten im selben Raum/Arbeitsumfeld. Aufgrund dieser Tatsache sind besondere Sicherheitsanforderungen für mobile autonome Roboter notwendig, die in dieser Ausarbeitung genauer erläutert werden sollen.

Definition

Gemäß Wikipedia werden autonome mobile Roboter definiert als:

»Als autonome mobile Roboter werden Roboter bezeichnet, die sich in ihrer Umgebung selbständig bewegen und agieren können. Dabei existieren aktuell verschiedene Abstufungen in Bezug auf die Autonomie, also die Unabhängigkeit des Roboters. Mobile Roboter werden oft schon als autonom bezeichnet, wenn die sie steuernde Software/Elektronik/Hardware sich auf dem Roboter befindet. Der Roboter ist dann solange autonom, wie seine Energieversorgung dies zulässt. Dem Roboter Anweisungen zu übermitteln, wie oder welche Aufgabe er erledigen soll, stört nicht seine Autonomie. Ein Roboter ist erst dann vollständig autonom, wenn der Roboter auch in Bezug auf seine Energieversorgung unabhängig ist, z. B. mittels einer Versorgung mit Energie zum Laden der Akkus über Solarzellen.[2]«

Asimovschen Robotergesetze

Bereits 1942 beschreibt der Schriftsteller Isaac Asimov die »asimovschen Robotergesetze«[3] in seiner Kurzgeschichte »Grundregeln des Roboterdienstes«. Die folgenden drei hierarchisch aufgebauten Gesetze finden beim heutigen Einsatz und Umgang mit Robotern keine Verwendung, allerdings können sie immer noch als Orientierungspunkt gesehen werden:

---

[1] IRobot, September 2010 http://www.irobot.com/de/about_irobot_at_glance.cfm
[2] Wikipedia, September 2010 http://de.wikipedia.org/wiki/Autonome_mobile_Roboter
[3] Wikipedia, August 2010 http://de.wikipedia.org/wiki/Robotergesetze

1. Ein Roboter darf kein menschliches Wesen (wissentlich) verletzen oder durch Untätigkeit gestatten, dass einem menschlichen Wesen (wissentlich) Schaden zugefügt wird.
2. Ein Roboter muss den ihm von einem Menschen gegebenen Befehlen gehorchen – es sei denn, ein solcher Befehl würde mit Regel eins kollidieren.
3. Ein Roboter muss seine Existenz beschützen, solange dieser Schutz nicht mit Regel eins oder zwei kollidiert.

Diese Gesetze sind jedoch nur als abstrakte Grundlage anzusehen. Konkrete Schritte zur Programmierung von Industrie- und Servicerobotern können daraus nicht abgeleitet werden. Für die konkrete Vorgehensweise gelten Normen, wie bspw. die der Maschinenrichtlinie.

Überblick

Im ersten Abschnitt »Sicherheitsanforderungen« (ab Seite 13) werden Gefahren und Risiken beim Einsatz von mobilen autonomen Robotern behandelt. Hierfür werden eine Reihe von Sicherheitsanforderungen für mobile autonome Roboter erläutert, die durch die richtige Anwendung von Sicherheitsvorkehrungen realisiert werden können.

Im zweiten Abschnitt »Anforderungen - Normen und Richtlinien« (ab Seite 18) werden Normen und Richtlinien eingeführt, die für die Planung und Konstruktion mobiler autonomer Roboter von Bedeutung sind. Im Bereich Robotik finden sich auch speziellere Informationen mit Bezug zu Sicherheitsrichtlinien. Des Weiteren werden die einzelnen Normentypen, Haftungsfragen sowie die Hierarchie der Europäischen Normen erläutert.

Im dritten Abschnitt »Gefahrenanalyse« (ab Seite 27) wird eine Methode zur Beurteilung der minimalen Sicherheitsanforderungen eines mobilen autonomen Roboters vorgestellt. Erläutert werden die Risikobeurteilung verschiedener Robotertyp sowie deren Einsatzbedingungen im Hinblick auf bestimmte Risikoparameter.

Im vierten Abschnitt »Steuerungskomponenten« (ab Seite 30) wird eine Methode zur Ermittlung der benötigten Steuerungskomponenten für einen autonomen mobilen Roboter skizziert. Zudem werden einige Begrifflichkeiten im Bezug auf Steuerungskomponenten beschrieben.

Im fünften Abschnitt (ab Seite 37) wird das Experiment „Safe Mobile Transport Robot (SMTR)“ « beschrieben. Die in Abschnitt 2 definierten Sicherheitsanforderungen eines autonomen mobilen Roboters werden auf einen Lego Mindstorm NXT Roboter übertragen und implementiert. Der im Experiment verwendeten Roboter imitiert einen Transportroboter. Die dafür relevante Norm DIN EN 1525 dient als Grundlage für die Implementierung des Roboters entlang der Sicherheitsanforderungen an fahrerlosen Flurförderzeugen.

Sicherheitsanforderungen

Neben den klassischen Anwendungsszenarien für Roboter im industriellen Umfeld, entsteht, wie eingangs erwähnt, zusätzlich ein Bedarf an mobilen Servicerobotern und Assistenzsystemen. Im Gegensatz zu den Industrierobotern, welche durch Schutzvorrichtungen wie Schutzzäunen, Lichtschranken etc. gesichert sind und zudem nicht direkt mit Menschen interagieren[4], müssen bei der Konstruktion, Programmierung und dem Einsatz von autonomen, mobilen Robotern eine Vielzahl an Sicherheitsvorkehrungen die dazu gehörigen Sicherheitsanforderungen einhalten.

Gefährdungen

Bei der direkten Interaktion zwischen Mensch und Roboter innerhalb der gleichen Arbeitsumgebung, entstehen gemäß der Norm ISO 12100-1 (Sicherheit von Maschinen) folgende Gefährdungspotentiale, welche für Serviceroboter relevant sind:

- Gefährdungen durch mechanisch-kinetische Energie, die bei Kollisionen zu Stößen, Quetschungen und Scherungen und somit zu Verletzungen führen können;
- Gefährdungen durch anliegende elektrische Spannungen und zu hohe Temperaturen (am Roboter);
- Gefährdung durch Erzeugung starker Vibrationen;
- Gefährdung durch Strahlenbelastung.

Eigenschaften

Für die Spezifizierung der Sicherheitsanforderungen, müssen die Gefährdungspotentiale einzeln betrachtet werden. Abhängig vom Robotertyp, dessen Aufgabe und dessen Arbeitsumgebung, sind geringere oder höhere Sicherheitsanforderungen zu beachten.

Abbildung 1 zeigt die Einstufung der benötigten Sicherheitsanforderungen im Bezug auf die folgenden Eigenschaften eines Roboters:

- Bauart/Typ des Roboters
- Aufgaben des Roboters
- Einsatzgebiet bzw. Umgebungsbedingungen

[4] Unter dem Schlagwort »Mensch-Roboter-Kooperation« entwickelt u.a. das Fraunhofer IPA Konzepte und Methoden zur »Kollisionsvermeidung« zwischen Mensch und Roboter.

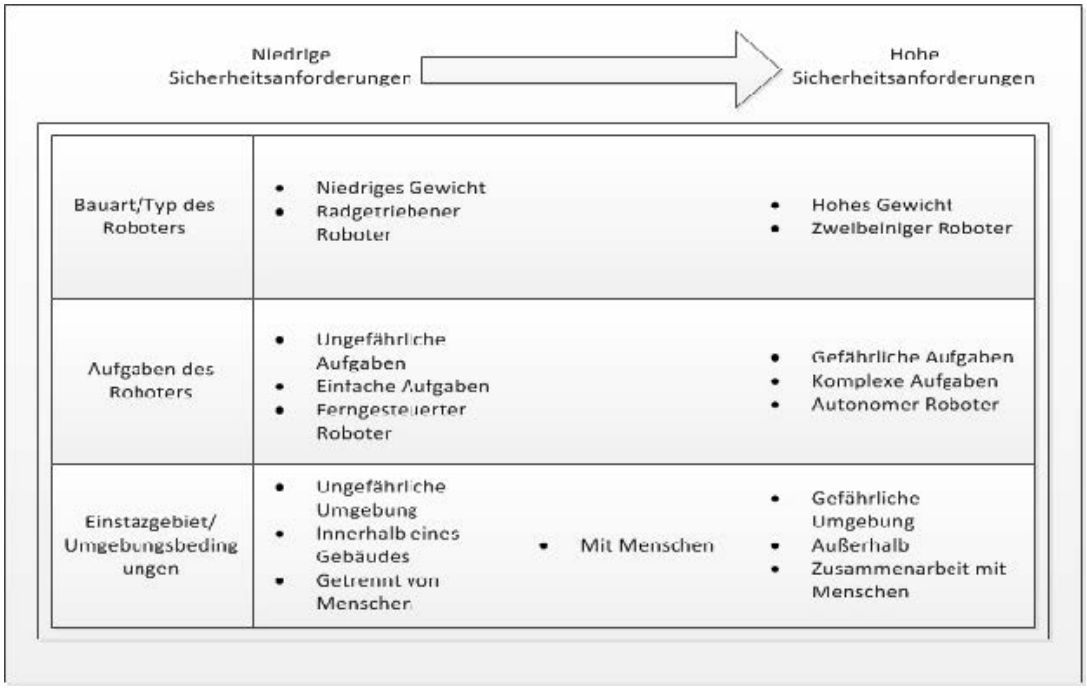


Abbildung 1: Sicherheitsanforderungen (nach: Adamy J. 2003, S. 438)

Aspekte die in Bezug auf Bauart und Typ des Roboters beachtet werden müssen, sind vor allem das Gewicht und die Fortbewegungstechnik des Roboters.[5] Ein Roboter der sich Radgetrieben fortbewegt und ein niedriges Gewicht aufweist, benötigt niedrigere Sicherheitsanforderungen als ein Roboter der sich auf zwei Beinen bewegt und ein hohes Gewicht besitzt.

Ausschlaggebend für die höheren Sicherheitsanforderungen sind zum einen die erhöhte Sturzgefahr bei zweibeinigen Robotern und zum anderen das erhöhte Verletzungsrisiko von Personen und die Beschädigung von Gegenständen bei der Kollision mit einem schweren Roboter. Zudem erhöhen sich die Sicherheitsanforderungen, sobald der Roboter gefährliche und komplexe Aufgaben bewältigen muss, wie beispielsweise das Arbeiten mit gefährlichen Werkzeugen oder das Transportieren von gefährlichen Gütern.

Je komplexer die Aufgabe ist und je höher der Grad der Autonomie ist, desto mehr Entscheidungen muss der Roboter treffen, um seine Aufgabe bewältigen zu können. Die Wahrscheinlichkeit, dass der Roboter einen Fehler macht und evtl. Schaden anrichtet, vergrößert sich demnach mit der Komplexität der Aufgabe. Arbeiten Roboter zusätzlich in gefährlichen Umgebungen, wie beispielsweise chemischen Anlagen oder in Umgebungen, in denen sie mit Menschen interagieren, steigen die Sicherheitsanforderungen ebenfalls.

Zudem kann die Vernachlässigung ergonomischer Aspekte, wie beispielsweise einer schlechten Mensch-Maschinen-Schnittstelle oder ein unverständliches Benutzerhandbuch, das Gefährdungspotential weiter erhöhen.

---

[5] Adamy J., Bechtel P.: Sicherheit mobiler Roboter – Safety of Mobile Robots, 2003, S. 438

Hierarchie der Roboterfunktionen

Damit ein Roboter seine Aufgaben erfüllen kann, ist es wichtig, dass alle zuvor definierten Sicherheitsanforderungen erfüllt sind. Erst wenn die Sicherheit von Menschen, Gegenständen und dem Roboter selbst durch Sicherheitsfunktionen gewährleistet sind, darf dieser die ihm gestellten Aufgaben ausführen. Deshalb ist es notwendig, die Sicherheitsfunktionen mit allen anderen Funktionen des Roboters in Beziehung zu setzen.[6]

Sicherheitsfunktionen besitzen die höchste Priorität. Dies bedeutet, dass der Roboter seine Aufgaben sofort abbrechen muss, sobald die Sicherheit von Personen, Gegenständen und seine eigene Sicherheit nicht mehr gewährleistet werden kann. Die Grafik in

Abbildung 2 veranschaulicht die Rangordnung der einzelnen Funktionen. Dabei nimmt die Wichtigkeit der einzelnen Funktionen nach unten zu.

Passive und aktive Sicherheits-methoden

Die vier untersten Schichten bilden die Sicherheitsfunktionen. Rechts in Abbildung 2, sind Beispiele und Möglichkeiten aufgelistet, um einzelne Funktionen zu realisieren. Als aktiv werden die Möglichkeiten bezeichnet, die während des Betriebes vom Roboter verwendet werden. Bei den passiven Methoden hingegen handelt es sich um die Möglichkeiten, die einen Zusammenhang mit der Konstruktion des Roboters aufweisen.

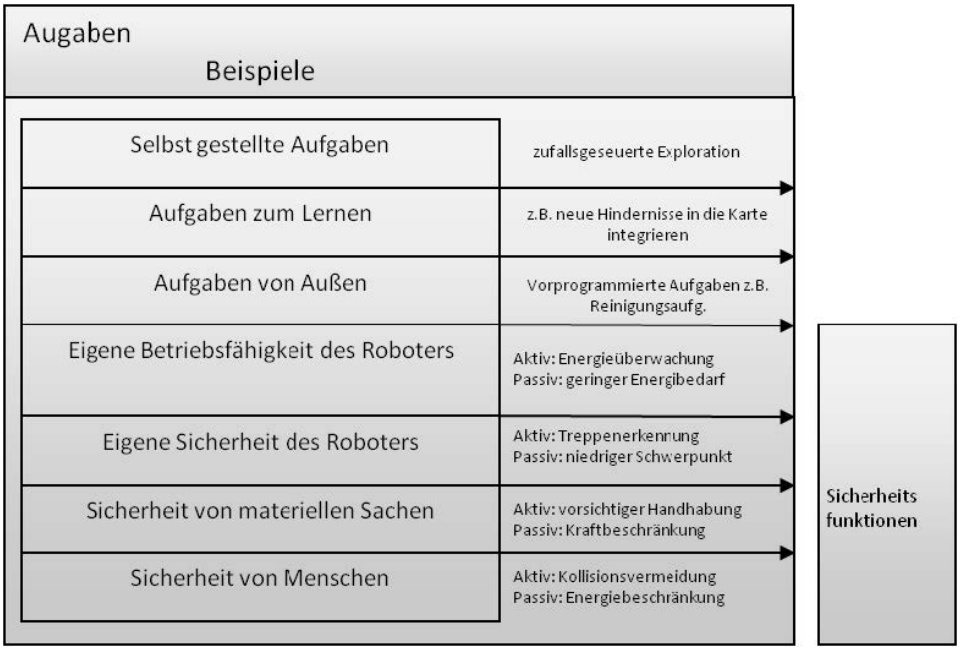


Abbildung 2: Sicherheitsfunktionen (nach: Adamy J. 2003, S.439)

[6] Adamy J., Bechtel P.: Sicherheit mobiler Roboter – Safety of Mobile Robots, Darmstadt 2003, S. 439

Höchste Priorität hat die Sicherheit von Personen. Für die Realisierung der Sicherheitsfunktionen ist es wichtig die passiven Methoden, wenn möglich, mit den aktiven Methoden zu kombinieren. Hierdurch wird gewährleistet, dass die passiven Methoden bei einer Fehlfunktion des Roboters ein gewisses Sicherheitsniveau aufrechterhalten.

Sicherer Zustand

Eine weitere wichtige Sicherheitsanforderung ist die Überführung des Roboters in einen sicheren Zustand sobald ein Fehler auftritt. Dabei muss der Roboter bei Auftreten und Erkennen des Fehlers seine laufende Aufgabe beenden um einen sicheren Zustand einzunehmen. Dieser Zustand kann beispielsweise das sofortige Ausschalten des Roboters bei einem Fehler sein. Diese Definition des sicheren Zustandes ist jedoch unter bestimmten Umständen gefährlich. Befindet sich der Roboter beispielsweise bei Auftreten eines Fehlers vor einem Notausgang, könnte er diesen blockieren. Eine Alternative wäre es, dass der Roboter erst zu einem speziell festgelegten Platz navigiert, um dort in einen sicheren, definierten Zustand überzugehen.

Ausblick

Um die Sicherheitsanforderungen zu erfüllen, muss ein Zusammenspiel zwischen Sicherheitskomponenten, wie Laserscanner, Ultraschallsensor etc., und Steuerungskomponenten, welche die Funktionalität der Sicherheitskomponenten überprüfen, erfolgen.

Diese Thematik wird im Kapitel Steuerungskomponenten erläutert. Zudem zeigt das Experiment – Safe Mobile Transport Roboter (SMTR) die Implementierung der hier vorgestellten Sicherheitsfunktionen.

Anforderungen – Normen und Richtlinien

Damit autonome Roboter als sicher eingestuft werden können, müssen bestimmte Sicherheitsanforderungen erfüllt sein. Diese Anforderungen sind durch Richtlinien festgelegt und werden durch die entsprechende Anwendung der dazugehörigen Normen erfüllt. Im Bereich der Industrieroboter bspw. existiert die Norm ISO 10218 – »Industrieroboter – Sicherheit«. Diese Norm gilt jedoch (bis zum jetzigen Zeitpunkt) nicht für autonome mobile Roboter. Allgemeine Informationen zur Sicherheit autonomer mobiler Roboter existieren nur ansatzweise. Deshalb müssen Sicherheitsanforderungen aus übergeordneten Normen wie der Sicherheitsgrundnorm EN ISO 12100 »Sicherheit von Maschinen« auf (mobile) autonome Roboter angewendet werden.

Richtlinien

Für Hersteller von Maschinen und somit auch für Hersteller von (mobilen autonomen) Robotern ist die wichtigste Richtlinie die so genannte Maschinenrichtlinie 2006/42 EG. Diese Richtlinie definiert die grundsätzlichen und einzuhaltenden Sicherheitsziele für Maschinen. Wie diese Sicherheitsziele erreicht werden sollen, ist in der Maschinenrichtlinie nicht definiert. Für eine Vielzahl von Maschinen wird die Umsetzung der Sicherheitsziele in den jeweiligen maschinenspezifischen Normen beschrieben.

Für Maschinen müssen in den meisten Fällen neben der Maschinen-richtlinie noch weitere Richtlinien, z. B. die EMV-Richtlinie 2004/108 EG und die Niederspannungsrichtlinie 2006/95 EG, berücksichtigt werden. Roboter (-Maschinen) unterliegen den oben genannten Richtlinien.

Normentypen

Die Maschinenrichtlinie enthält neben den Sicherheitszielen eine Vielzahl von Normen. Diese unterteilen sich in die folgenden drei Normentypen A, B und C unterteilt:[7]

- **Typ A: Sicherheitsgrundnormen**
  Die Sicherheitsgrundnormen enthalten Gestaltungsleitsätze, Grundbegriffe und allgemeine Aspekte für Maschinen.
- **Typ B: Sicherheitsgruppennormen**
  Die Sicherheitsgruppennormen werden in 2 Untergruppen aufgeteilt –Typ B1 und Typ B2- und enthalten sicherheitstechnische Einrichtungen und Sicherheitsaspekte, die auf eine ganze Reihe von Maschinen anwendbar sind.
- **Typ B1: Sicherheitsaspekte**
  Sicherheitsaspekte, die eine ganze Reihe von Maschinen gleichermaßen betreffen (z.B. Sicherheitsabstände)
- **Typ B2: Sicherheitseinrichtungen**
  Sicherheitseinrichtungen, die für eine Vielzahl von verschiedenen Maschinen eingesetzt werden können (z.B. Not-Aus-Schalter)
- **Typ C:**
  Sicherheitsproduktnormen Die Sicherheitsproduktnormen enthalten konkrete Anforderungen und Schutzmaßnahmen zu den wichtigsten Gefährdungen, die von einer speziellen Art von Maschine ausgehen.

---

[7] Dietz Sensortechnik: Sicherheit an Maschinen, 2007, S. 2f.

Hierarchie

Abbildung 3 veranschaulicht den hierarchischen Aufbau der Europäischen Normen. Auf Ebene 1 befinden sich die allgemeinen Sicherheitsgrundnormen. Auf Ebene 2 befinden sich die Sicherheitsgruppennormen, und auf der letzten Ebene sind die spezifischen Sicherheitsproduktnormen angesiedelt.[8]

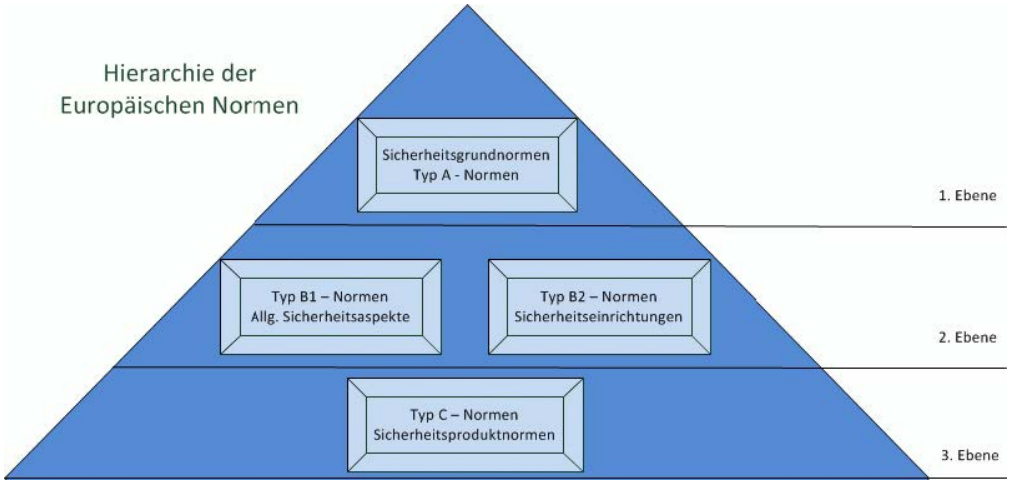


Abbildung 3: Normenhierarchie (nach: Leuze Electronic 2010)

Anwendung

Existiert für eine Maschine eine spezifische Sicherheitsproduktnorm (Typ C), welche soweit wie möglich Bezug auf die Sicherheitsgrundnormen (Typ A) und Gruppennormen (Typ B) nimmt, müssen die in der Sicherheitsproduktnorm definierten Anforderungen erfüllt werden. Sicherheitsproduktnormen können bspw. Anforderungen enthalten, die von den Gruppennormen abweichen, sodass ihnen eine größere Bedeutung zukommt.[9] Existiert jedoch für eine Maschine keine spezielle Sicherheitsproduktnorm, so müssen die in den übergeordneten Typ B Normen spezifizierten Anforderungen erfüllt werden. Ist dies ebenfalls nicht möglich, da die Maschine in keine der Kategorien der Sicherheitsgruppennormen fällt, müssen die Anforderungen der Sicherheitsgrundnorm (Typ A) realisiert werden.

Beispiel

Um den zuvor beschriebenen Zusammenhang der Normentypen besser verstehen zu können, wird an dieser Stelle ein Beispiel aufgegriffen.

Als Beispiel dient die Typ C Norm DIN EN ISO 3619-4: »Flurförderzeuge – Sicherheitstechnische Anforderungen und Verifizierung - Teil 4: Fahrerlose Flurförderzeuge und ihre Systeme«[10]

---

[8] Leuze Electronic: August 2010, http://www.arbeitssicherheit.leuze.de/a/a_07.html
[9] Dietz Sensortechnik: Sicherheit an Maschinen, 2007, S. 4
[10] Nachfolger der DIN EN 1525 – Fahrerlose Flurförderzeuge und ihre Systeme. Die DIN EN ISO 3619-4 befindet sich zurzeit im Harmonisierungsverfahren. http://www.din.de/cmd?level=tpl-unterrubrik&cmssubrubid=47507&languageid=de

In dieser Norm werden die benötigten Sicherheitsanforderungen für fahrerlose Flurförderzeuge spezifiziert. Um beispielsweise Verletzungen die durch eine Kollision zwischen Flurförderzeug und Personen entstehen, zu vermeiden, werden in der Norm entsprechende Anforderungen gestellt, an …

… das Bremssystem,

… die Geschwindigkeitsregelung,

… die Einrichtungen zur Personenerkennung im Fahrweg,

… Not-Aus Einrichtung,

…

Die genauen Anforderungen und die dazu gehörigen Lösungsansätze werden in der Norm genauestens beschrieben. Dabei wird in vielen Fällen innerhalb der Norm auf andere Normen (Typ A und Typ B) verwiesen. Zur Gestaltung der Not-Aus-Einrichtung beispielsweise, wird für ein fahrerloses Flurförderzeug eine Not-Aus-Einrichtung der Kategorie B der Norm ISO 13850 benötigt. Diese muss zudem nach der Norm DIN EN ISO 3619-4 von beiden Enden und beiden Seiten des Flurförderzeuges gut sichtbar, erkennbar und zugänglich sein. Bei den Normen handelt es sich um eine Art Rezept für eine Maschine, zur Gewährleistung der Sicherheit, welche wiederum auf andere Rezepte (Normen) verweist, um somit redundante Informationen zu vermeiden.

Normen für autonome mobile Roboter

Da für mobile autonome Roboter keine spezifischen Sicherheitsproduktnormen (Typ C) existieren, mit Ausnahme der Normen EN 1525 und EN 1526 für fahrerlose Flurförderfahrzeuge, müssen die gestellten Sicherheitsvorkehrungen auf Grundlage der Sicherheits-gruppennormen (Typ B) und Sicherheitsgrundnormen (Typ A) realisiert werden.

Die derzeit (Stand 2010) für die autonome mobile Robotik wichtigsten Normen der Maschinenrichtlinie sind in Tabelle 1 aufgelistet. Die rot gekennzeichneten Normen sind veraltete Normen, die durch entsprechend neue Normen abgelöst wurden.

Referenz	Titel der Norm	Normentyp
EN ISO 12100-1	Sicherheit von Maschinen – Grundbegriffe, allg. Gestaltungsleitsätze – Teil 1: Grundsätzliche Terminologie, Methodologie	A
EN ISO 12100-2	Sicherheit von Maschinen – Grundbegriffe, allg. Gestaltungsleitsätze – Teil 2: Technische Leitsätze	A
EN 1050 ISO 14121-1	Sicherheit von Maschinen – Leitsätze zur Risikobeurteilung	A
EN 954-1 EN ISO 13849-1	Sicherheit von Maschinen – Sicherheitsbezogene Teile von Steuerungen – Teil 1: Allg. Gestaltungsleitsätze	B
EN 954-2 EN ISO 13849-1	Sicherheit von Maschinen – Sicherheitsbezogene Teile von Steuerungen – Teil 2: Validierung	B
EN 999 ISO 13855	Sicherheit von Maschinen – Anordnung von Schutzeinrichtungen im Hinblick auf Annäherungsgeschwindigkeiten von Körperteilen	B

Referenz	Titel der Norm	Normentyp
EN 60204-1 IEC 60204-1	Sicherheit von Maschinen – Elektrische Ausrüstung von Maschinen – Teil 1: Allg. Anforderungen	B
EN 62061 IEC 62061	Funktionale Sicherheit sicherheitsbezogener elektrischer, elektronischer und programmierbar elektronischer Steuerungssysteme	B
prEN/TS 62046 IEC/TS 62046	Sicherheit von Maschinen – Anwendungen von Schutz-einrichtungen zum Erkennen von Personen	B
EN 418 EN ISO 13850	Sicherheit von Maschinen – NOT-AUS-Einrichtungen, funktionale Aspekte – Gestaltungsleitsätze	B
EN 61496-1 IEC 61496-1	Sicherheit von Maschinen – Berührungslos wirkende Schutz-einrichtungen – Teil 1: Allg. Anforderungen und Prüfungen	B

Tabelle 1: Maschinensicherheitsnormen

EN = Europäische Normen

ISO/IEC = Internationale Normen

EN ISO = Europäische Norm die auf einer internationalen Norm der ISO beruht

CE-Kennzeichnung

Produkte die innerhalb der EU verkauft werden, benötigen eine CE Kennzeichnung (früher EG für Europäische Gemeinschaften). Dieses CE-Zeichen bestätigt, dass die Maschine (z.B. ein Roboter) den geltenden Anforderungen der EN-Richtlinien und Normen entspricht. Um eine CE Kennzeichnung verwenden zu dürfen, muss der Hersteller eine so genannte Konformitätserklärung ausfüllen.[11] In dieser Konformitätserklärung bestätigt er alle, erforderlichen Richtlinien eingehalten und Normen angewendet zu haben. Der Hersteller belegt dies mit einer ausführlichen Dokumentation. Inhalt dieser Dokumentation sind dabei folgende Punkte:

- Gefahrenanalyse (nach EN 12100, EN 14121)
- Risikobewertung (nach EN 14121)
- Technische Dokumentation
- Betriebsanleitung

Abbildung 4 stellt den oben beschriebenen Zusammenhang zwischen CE Kennzeichnung, der Konformitätserklärung, den Normen und den Richtlinien grafisch dar. Zuerst muss sich der Hersteller mit den Richtlinien befassen, die für die Herstellung seiner Maschinen relevant sind. Darunter fallen, wie bereits erwähnt, die Maschinenrichtlinie 2006/42 EG, sowie die EMV-Richtlinie 2004/108 EG und die Niederspannungsrichtlinie 2006/95 EG. Um diese Richtlinien zu erfüllen, muss der Hersteller die in den Normen spezifizierten Anforderungen, die für seine Maschinen relevant sind, erfüllen. Wie oben bereits beschrieben muss der Hersteller nun die Konformitätserklärung ausstellen, um das CE-Kennzeichen an seine Maschinen anbringen zu dürfen.

[11] Dietz Sensortechnik: Sicherheit an Maschinen, 2007, S. 4f.

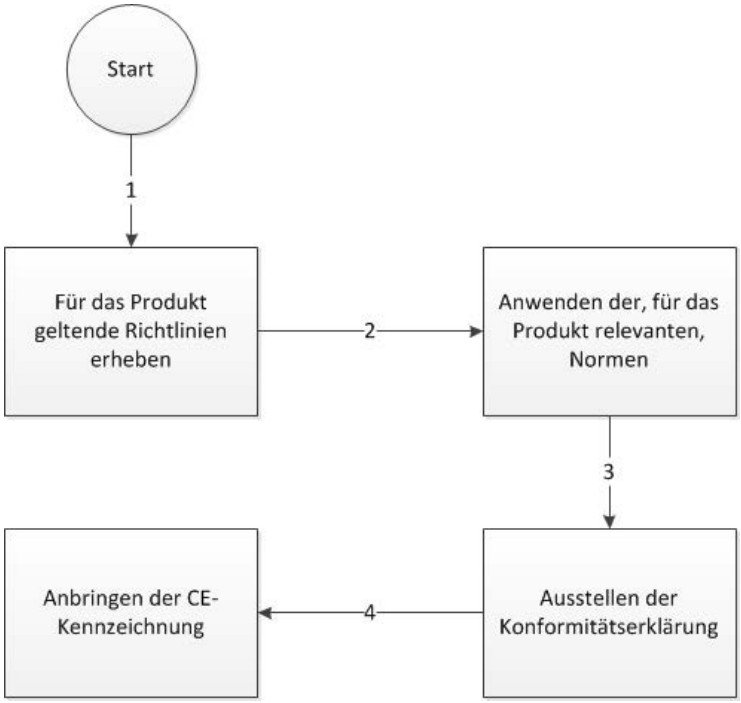


Abbildung 4: Der Weg zur CE-Kennzeichnung

Haftung

Kommt es trotzt einer CE-Kennzeichnung zu einem Unfall, welcher durch ein sicherheitstechnisches Problem zustande kam, wird der Hersteller zur Verantwortung gezogen, der die Maschine hergestellt hat und die Konformitätserklärung abgegeben hat. Bei Maschinen, die aus Nicht-EU-Ländern importiert werden, trägt der Importeur die Verantwortung.[12] Dabei zu beachten ist, dass ein Zusammenbau mehrerer CE zertifizierter Teile zu einer neuen Maschine nicht automatisch besagt, dass diese neue Maschine ebenfalls CE zertifiziert ist. Erst wenn diese Maschine ebenfalls die geltenden Richtlinien und Normen erfüllt und eine Konformitätserklärung durch den Hersteller erfolgt, ist diese Maschine wiederum CE zertifiziert.

Softwarefehler, Viren

Im Hinblick auf autonome Roboter ergibt sich die Frage, wer haftet, wenn der Roboter auf Grund fehlerhafter bzw. schadhafter Software Schaden an Menschen und Gegenständen anrichtet?
Wer haftet, wenn der Roboter durch einen Virus infiziert wurde und somit einen sicherheitskritischen Zustand erreicht?

Mit dieser Thematik beschäftigt sich zurzeit der Würzburger Robotik-Experte Prof. Klaus Schilling. Ziel seiner Arbeit[13] ist es, ethische und juristische Grundlagen zu schaffen um die Verantwortung für Maschinen besser einstufen zu können. Diese Rechtsicherheit soll in naher Zukunft Entwicklern einen Orientierungsrahmen bereitstellen.

---

[12] Bayerisches Staatsministerium für Wirtschaft, Infrastruktur, Verkehr und Technologie: Sicherheit von Maschinen, März 2005, S.3
[13] Spiegel Online Juli 2010, http://www.spiegel.de/wissenschaft/technik/0,1518,705961,00.html

Gefahrenanalyse

Über die Sicherheit eines Roboters gibt die in der Norm EN 12100-1 beschriebe Gefahrenanalyse verschiedene Anhaltspunkte. Die Gefahrenanalyse sollte bereits während der Entwicklungsphase des Roboters durchgeführt werden. Sie formuliert Risiken und Leitsätze zur Gefahrenminimierung. Folgende Punkte sind hierbei zu berücksichtigen:

Bestimmung der Grenzen einer Maschine Es müssen Eigenschaften wie Verwendungszweck, zeitliche Grenzen etc. der Maschine festgelegt werden

Identifizierung der Gefährdungen Analyse von Gefährdungen, wie das Eingreifen von Personen, die Betriebszustände der Maschine, Missbrauch durch Dritte etc.

Verfahren zur Risikoeinschätzung Ermittlung von Parametern wie Verletzungsgrad, Schadensumfang, Häufigkeit und Dauer der Gefahr, Eintrittswahrscheinlichkeiten, Möglichkeiten zur Vermeidung oder zur Begrenzung des Schadens

Hierarchie der Risikominderung

Die ermittelten Risiken müssen nach einer strengen Hierarchie neutralisiert oder auf ein minimales Restrisiko verringert werden. Nach Möglichkeit sollte das Risiko als erstes durch Veränderung der Konstruktion der Maschine[14] minimiert werden. Ist dies nicht möglich, müssen zusätzlich technische Schutzmaßnahmen das entsprechende Risiko reduzieren. Sollte auch dies nicht möglich sein, kann in letzter Instanz das Risiko durch Unterweisen der Mitarbeiter und der Anbringung von Warnschildern minimiert werden. Diese Reihenfolge dient dazu, dass sich Hersteller nicht der Verantwortung entziehen, technische Sicherheitsmaßnahmen umzusetzen.

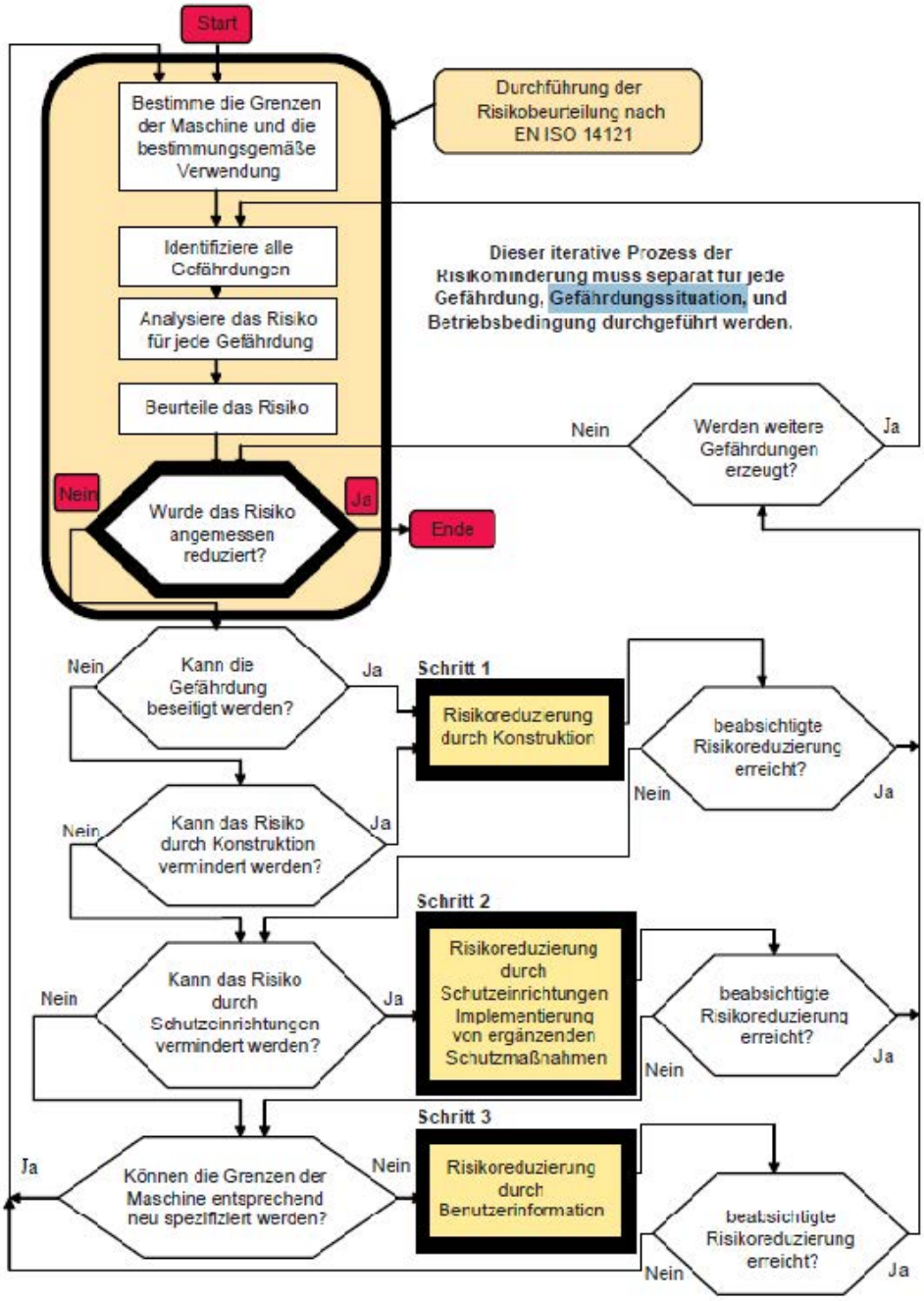
Beispielsweise könnte der Hersteller eines Roboters gemäß seiner erstellten Benutzerinformation darauf aufmerksam machen, dass sich kein Mitarbeiter dem Roboter bis auf 1 Meter Entfernung nähern darf, da der Roboter keine technischen Maßnahmen zur Vermeidung einer Kollision integriert hat. Der Hersteller würde damit das Sicherheitsniveau abhängig von Handlungen der betroffenen Personen machen. Damit dieses Szenario nicht eintritt und eine Verschiebung der zu treffenden Sicherheitsmaßnahmen vom Hersteller auf den Kunden übergeht, wurde die zuvor erläuterte Hierarchie festgelegt.

Für die Bewertung und Minimierung der Risiken kann die Risikobewertung der Norm EN 14121 entnommen werden. Um eine ausreichende Sicherheit des Roboters zu gewährleisten, müssen schrittweise wiederholte Analysen durchgeführt werden. Erst wenn das Restrisiko auf ein Minimum reduziert ist, kann der Prozess der Risikobewertung beendet werden.[15] Abbildung 5 Diese Vorgehensweise der Risiko-bewertung und -minimierung wird in veranschaulicht.

---

[14] DIN EN ISO 12100-1: Sicherheit von Maschinen – Grundbegriffe, allg. Gestaltungsleitsätze – Teil 1: Grundsätzliche Terminologie, Methodologie, April 2004

[15] DIN EN ISO 14121-1: Sicherheit von Maschinen – Risikobeurteilung – Teil 1: Leitsätze, Dezember 2007



[tt1]

Abbildung 5: Iterativer Prozess zur Risikobewertung und –minimierung (nach: DIN EN ISO 14121-1 2007)

Steuerungskomponenten

Nachdem die Risikobewertung durchgeführt wurde, muss eine Festlegung der benötigten Steuerungskomponenten erfolgen. Die durch die Risikobewertung identifizierten Gefahren können durch sicherheitstechnische Maßnahmen, wie z.B. Ultraschallsensoren, neutralisiert bzw. reduziert werden, dennoch kann es zu Ausfällen genau dieser Komponenten kommen. Die Sicherheit wäre somit nicht mehr gewährleistet. Um dies zu verhindern werden sogenannte sicherheitsgerichtete Steuerungskomponenten verwendet, welche die Funktionalität der sicherheitstechnischen Maßnahmen überwachen.

Risikoanalyse nach DIN EN ISO 13849-1

Zur Beurteilung und Auswahl einer geeigneten Steuerungskomponente für eine Sicherheitskomponente muss eine Risikoanalyse nach der Norm DIN EN ISO 13849-1 durchgeführt werden. Dabei müssen die Gefahren, die beim Ausfall einer Sicherheitskomponente auftreten könnten, identifiziert werden. Bspw. könnte der Ausfall eines Laserscanners dazu führen, dass eine im Weg stehende Person nicht erkannt wird und der Roboter diese Person aufgrund einer Kollision verletzt. Dieser Ausfall muss durch die Steuerungs-komponente erfasst werden, damit der Roboter in einen sicheren Zustand überführt werden kann. Die wichtigsten zu ermittelnden Informationen sind nach DIN EN ISO 13849-1:

- Schwere der Verletzung (S)
  - S1 = leichte (reversible) Verletzung
  - S2 = schwere (irreversible) Verletzung einschließlich Tod

- Häufigkeit und/oder Dauer der Gefährdung (F)
  - F1 = selten bis öfters und/oder kurze Dauer
  - F2 = häufig bis dauernd und/oder lange Dauer

- Möglichkeiten zur Vermeidung der Gefährdung (P)
  - P1 = möglich unter bestimmten Bedingungen
  - P2 = nicht oder kaum möglich

Performance Level (PL)

Je nach Zusammenspiel dieser Informationen, erfolgt eine Einteilung in sogenannte Performance Level (PL). Die Performance Level sind in fünf Stufen unterteilt, von »A« bis »E«. Beim PL »A« ist die Gefährdungssituation, die von der Maschine ausgeht, gering; somit ist der Beitrag der Steuerungskomponente zur Risikominderung gering. Beim PL »E« hingegen ist die Gefährdungssituation extrem hoch, wodurch eine zuverlässige und genaue Steuerungs-komponente benötigt wird. Zur Beurteilung des mindestens benötigten PLs der Maschine kann die Grafik in Abbildung 6 angewendet werden.

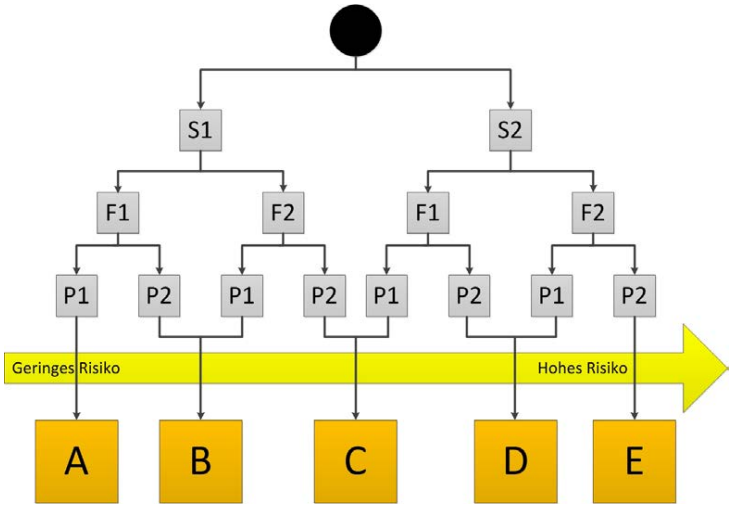


Abbildung 6: Ermittlung des Performance Levels (nach: DIN EN ISO 13849-1)

Performance Level required (PLr)

Das aus Abbildung 6 resultierende PL wird als PLr bezeichnet, wobei das r für »required« (engl. für »erforderlich«) steht. Somit wird festgelegt, welches PL die Steuerungskomponente mindestens für die Maschine besitzen muss.

Beispiel-Szenario

Das bereits beschriebene Szenario »Ausfall des Lasersensors« würde demnach folgender PLr hervorgehen:

Fällt der Laserscanner aus, kann es durch Kollisionen mit Personen zu schweren Verletzungen kommen (S1). Die Dauer und Häufigkeit einer Gefährdung ist sehr hoch (F2), da der Roboter keine Hindernisse mehr erkennen kann. Eine Möglichkeit zur Vermeidung der Gefährdung ist kaum möglich (P2), da zwar Personen eventuell dem Roboter ausweichen könnten, Gegenstände und Sachen jedoch nicht. Die Ermittlung des PLr's für den Transportroboter mit integriertem Laserscanner wird in Abbildung 7 dargestellt.

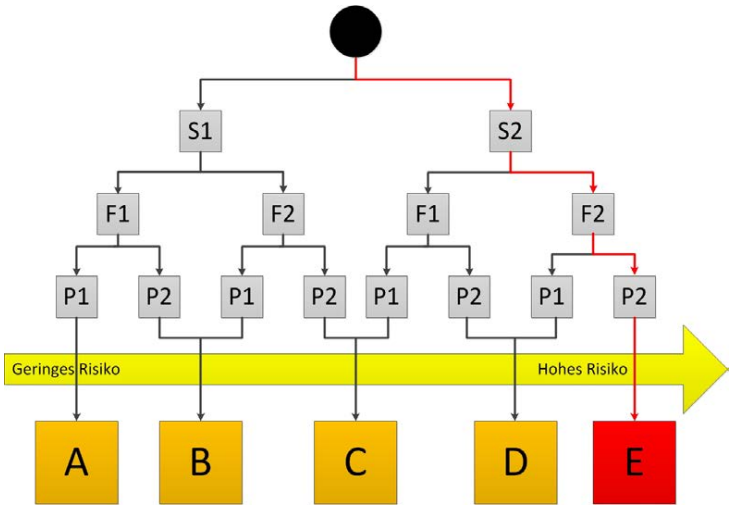


Abbildung 7: PL-Ermittlung eines Laserscanners (nach: DIN EN ISO 13849-1)

Redundante Steuerungskomponenten

Ein Roboter, der einen Laserscanner verwendet, entspricht somit dem PLr »E«, wodurch eine zuverlässige und genaue Steuerungskomponente unverzichtbar ist. Deshalb ist eine Steuerungskomponente die dem PL »E« entspricht erforderlich. Je nach Performance Level der Maschine müssen die Steuerungskomponenten die in der Norm DIN EN ISO 13849 spezifizierten Anforderungen erfüllen. In niedrigeren PLs (A und B) wird bspw. die Verwendung von sicherheitstechnisch bewährten Bauteilen für die Steuerung gefordert. Jedoch kann es auch bei diesen Komponenten zu Einzelfehlern kommen, die zum Verlust der Sicherheitsfunktion führen. Um dies zu verhindern werden bei Sicherheitskomponenten mit höheren PLs (»D« und »E«) redundante Steuerungskomponenten verwendet. Bei der Redundanz unterscheidet man zwischen der homogenen Redundanz und der heterogenen Redundanz. Bei der homogenen Redundanz werden parallel gleichartige Bauteile für die Steuerung von Sicherheitskomponenten verwendet. Bei der heterogenen Redundanz hingegen werden parallel unterschiedliche Bauteile von verschiedenen Herstellern verwendet, die jedoch ein und dieselbe Aufgabe ausführen. Dies hat den Vorteil, dass sogenannte »common-case«-Fehler (z. B. Serienfehler einer Baureihe) dadurch seltener sind. Daher sind sicherheitstechnische Steuerungen mit heterogener Redundanz sicherer, aber auch in der Regel teurer zu realisieren.[16]

Existiert für die zu entwerfende Maschine bereits eine Sicherheits-produktnorm (Typ C), so kann der PLr Wert dieser Norm entnommen werden. Hierdurch ist eine Risikobeurteilung nach DIN EN ISO 13849 (s. Abbildung 6) nicht mehr notwendig.

Kenngrößen

Um den PL einer Steuerungskomponente ermitteln zu können, müssen die verwendete Architektur und die einzelnen Sicherheitsbauteile der Steuerung genau betrachtet werden. In die Berechnung des PLs fließen folgende qualitative und quantifizierbare Kenngrößen, welche im Anhang dieser Ausarbeitung genauer beschrieben werden, ein:

- Architektur des Systems (qualitativ)
- Mean Time To Failure (MTTF, oder auch MTTFd) (quantitativ)
- B10 (quantitativ)
- Diagnosedeckungsgrad – Diagnostic Coverage (quantitativ)
- Ausfallrate (quantitativ)

SISTEMA

Die durchschnittlichen Werte der oben genannten Kenngrößen können der Norm DIN EN ISO 13849 entnommen werden. Des Weiteren werden diese Kenngrößen in der Regel vom Hersteller des jeweiligen Bauteils angegeben. Andernfalls können die Werte durch das kostenfreie Tool SISTEMA ermittelt werden. SISTEMA steht für Sicherheit von Steuerungen an Maschinen und ermöglicht die Bewertung von sicherheitsbezogenen Teilen einer Steuerung. Auf Basis der vorgegebenen Architektur berechnet SISTEMA die Zuverlässigkeitswerte einschließlich des erreichten Performance Levels. Die erforderlichen Kenngrößen wie MTTFd oder B10d, werden dabei über Bauteilbibliotheken ermittelt. Entwickelt wurde SISTEMA vom Institut für Arbeitsschutz der Deutschen Gesetzlichen Unfallversicherung.[17]

---

[16] Adamy J., Bechtel P.: Sicherheit mobiler Roboter – Safety of Mobile Robots, Darmstadt 2003, S. 440
[17] Rexroth Bosch Group: »EN ISO 13849: Und sie gilt immer häufiger doch«, Juli 2007

Alternativ kann man den PL-Wert mit Hilfe eines Diagramms, welches in der Norm EN ISO 13849 spezifiziert ist, ermitteln.

Entspricht der erreichte PL-Wert mindestens dem zuvor ermittelten PLr-Wert, kann die Steuerungskomponente als sicher eingestuft und verwendet werden.

2. Experiment »Safe Mobile Transport Robot (SMTR)«

Anforderungen

Szenario

Beim Experiment SMTR geht es darum, einen Transportroboter zu implementieren, der gewisse Sicherheitsanforderungen erfüllt und einer vordefinierten Fahrstrecke folgt. Der Schwerpunkt dieses Experiments liegt nicht beim Transport von Gütern durch den Roboter, sondern beim Aspekt der Sicherheit vor und während des Betriebes. Die Strecke (siehe Abbildung 8) stellt den Transportweg des Roboters dar und verfügt über zwei Be- und Entladepunkte, sowie eine Wartungszone. Der Transportroboter wird im Abschnitt »Komponenten des Roboters« spezifiziert.

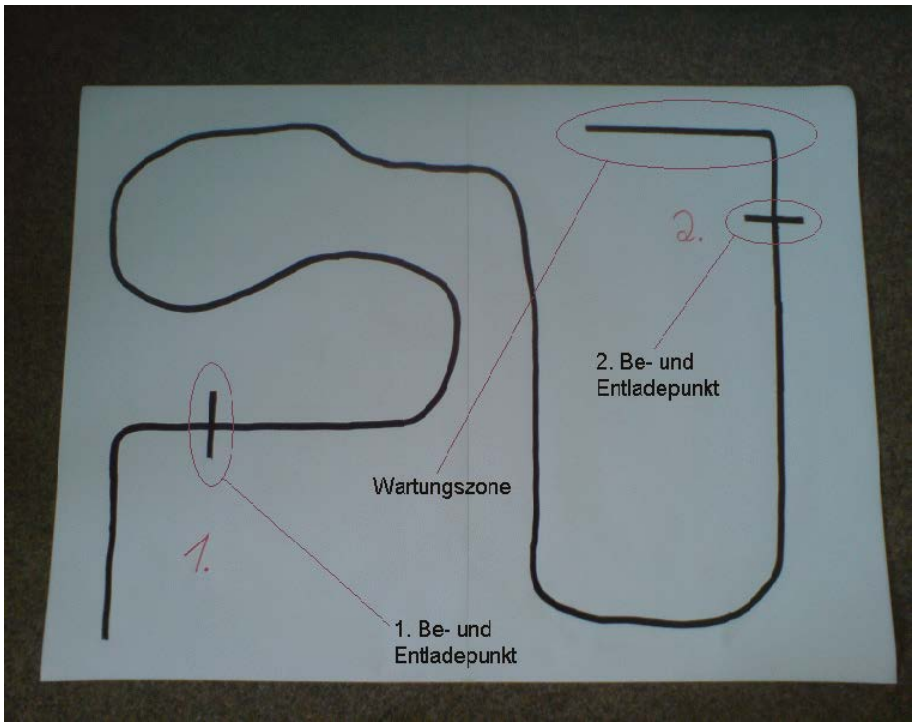


Abbildung 8: Fahrbahn des Transportroboters

Be- und Entladepunkte

Die beiden Be- und Entladepunkte sind die beiden Begrenzungen der Fahrstrecke. Der Roboter wird bspw. von einem Roboterarm oder einer Person be- und entladen. Anschließend muss der Roboter um 180 Grad wenden und seinen Weg in entgegengesetzter Richtung fortsetzen.

Wartungszonen

Sofern eine Fehlfunktion oder ein bestimmtes Ereignis, wie bspw. eine zu geringe Stromversorgung oder ein defekter Sensor, eintritt und der Roboter dies selbstständig registriert, muss der Roboter am

nächsten Be- und Entladepunkt die Strecke verlassen. Dies geschieht, indem er die jeweilige Wartungszone betritt, um sich daraufhin selbstständig auszuschalten. In der Wartungszone, die außerhalb des Transportbetriebs liegt, kann der Roboter durch einen Techniker gewartet bzw. repariert werden.

Schwarze Linie

Die schwarze Linie visualisiert den genauen Transportweg. Mittels Lichtsensor(en) fährt der Roboter entlang der schwarzen Linie vom Be- zum Entladepunkt (und ggf. wieder zurück).

Musskriterien

Auf Grund der in Kapitel Sicherheitsanforderungen definierten Sicherheitsfunktionen (siehe Abbildung 2) und der in der Typ C Norm DIN EN 1525 – Fahrerlose Flurförderzeuge und ihre Systeme spezifizierten Anforderungen, gemäß denen sich der Transportroboter aufgebaut ist, sind im weiteren Verlauf einige Kriterien aufgelistet, deren Erfüllung höchste Priorität besitzt. Zudem sind aufgrund des gegebenen Szenarios zusätzliche selbstdefinierte Kriterien hinzugekommen.

Der Transportroboter muss folgende Kriterien erfüllen:

- Der Roboter muss in der Lage sein, die gekennzeichnete Fahrbahn abfahren zu können, ohne diese auch nur kurzzeitig verlassen zu dürfen.
- Vor Inbetriebnahme des Roboters und während des Betriebes muss gewährleistet werden, dass alle Sensoren funktionstüchtig sind.
- Wenn einer der beiden Lichtsensoren ausfällt, muss der Roboter in der Lage sein, nur mit Hilfe eines Lichtsensors die Strecke weiter zu fahren. Erreicht der Roboter daraufhin einen Be- und Entladepunkt, muss dieser die Fahrbahn verlassen und sich selbstständig abschalten um eine Reparatur des beschädigten Sensors durch einen Techniker einleiten zu können.
- Um eine Kollision zu vermeiden muss der Roboter in der Lage sein, auf der Fahrbahn befindliche Hindernisse zu erkennen.
- Sobald die Energiequelle des Roboters einen vordefinierten Tiefstand erreicht hat, muss dieser am nächsten Be- und Entladepunkt die Fahrbahn verlassen und sich selbstständig abschalten.
- Es muss eine redundante Not-Aus-Einrichtung implementiert werden, die nach DIN EN 1525 von beiden Enden und beiden Seiten des Flurförderzeuges gut sichtbar, erkennbar und zugänglich sein.
- Die Implementierung muss in Java erfolgen oder alternativ in Ada[18,19]

[18] Wikipedia, September 2010, http://de.wikipedia.org/wiki/Ada_%28Programmiersprache%29
[19] AdaCore, September 2010, http://libre.adacore.com/libre/tools/mindstorms/

Komponenten des Roboters

Transportroboter

Der für das Projekt entwickelte Lego Mindstorm NXT-Roboter (s. Abbildung 9-10) besteht aus:

- 3 Motoren,
- 1 Ultraschallsensor,
- 2 Lichtsensoren,
- 1 Tastsensor
- 1 NXT-Steuereinheit.

Die NXT-Steuereinheit wurde so tief wie möglich platziert, um einen niedrigen Schwerpunkt und damit eine höhere, eigene Betriebssicherheit zu erreichen.

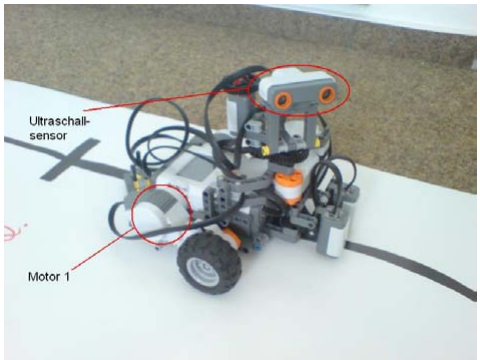


Abbildung 9: Transportroboter (Seitenansicht)

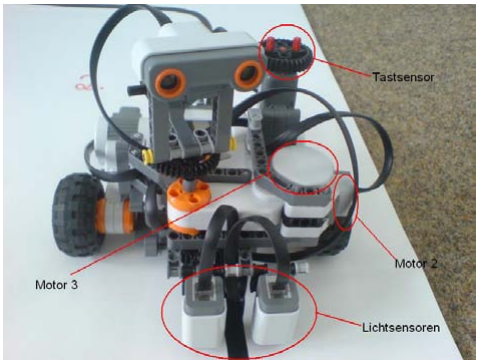


Abbildung 10: Transportroboter (Frontansicht)

NXT-Steuereinheit

Bei der NXT-Steuereinheit (siehe Abbildung 11) handelt es sich um eine programmierbare Steuereinheit, auf dem entwickelte Programme ausgeführt werden können. Die NXT-Steuereinheit besitzt einen 32-Bit Mikroprozessor mit 256 KB Flash- und 64 KB Arbeitsspeicher (RAM). Über die 7 Ports können bis zu 4 Sensoren und 3 Motoren gleichzeitig angeschlossen und gesteuert werden. Des Weiteren ist eine Bluetooth Schnittstelle implementiert, die es ermöglicht via PC oder mit anderen NXT Steuereinheiten zu kommunizieren. Als Stromquelle für die NXT Steuereinheit dient ein Akku, der 7,4 V liefert. Um Textnachrichten ausgeben zu können verfügt die NXT Steuereinheit zudem über ein LC-Display mit den Maßen 100 x 64 Pixel. Als weitere Ausgabequelle kann der interne Lautsprecher genutzt werden. Für die Eingabe und zur Menüsteuerung (Ein- und Ausschalten) dienen 4 Knöpfe, welche unterhalb des LC-Displays platziert sind.

Abbildung 11: NXT-Steuereinheit

NXT-Motor

Beim NXT-Motor (s. Abbildung 12) handelt es sich um einen Servomotor mit einem integrierten Rotationssensor, welcher in der Lage ist, die Rotationen des Motors in Grad zu messen. Der Motor des NXT kann dabei in 1 Grad (1 Umdrehung = 360 Grad)-Schritten angesprochen werden. Die Leistung bzw. Motorgeschwindigkeit ist dabei abhängig von der angeschlossenen Energiequelle und kann entweder durch eine prozentuale Angabe (1-100) der Motorleistung oder durch Angabe der Gradzahl (1-900), um die der Motor pro Sekunde rotieren soll, festgelegt werden.

Abbildung 12: NXT-Motor

Antriebe des Roboters

Für das Experiment werden 2 Motoren als Antrieb zur Fortbewegung (s. Abbildung 13-14) des Roboters verwendet. Der dritte Motor dient dazu, den Ultraschallsensor um 360 Grad rotieren zu können (s. Abbildung 15).

Abbildung 13: Linker Antrieb

Abbildung 14: Rechter Antrieb

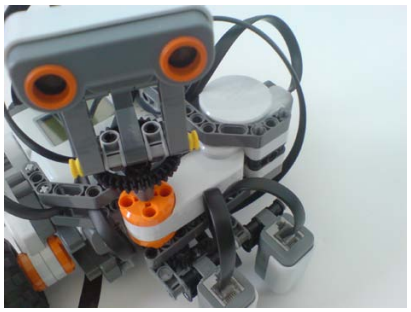


Abbildung 15: Antrieb für den Ultraschallsensor

Ultraschallsensor

Mit Hilfe des Ultraschallsensors (siehe Abbildung 16) können Abstände zu Objekten in cm gemessen werden. In einem Winkel von 30 Grad und einer maximalen Entfernung von 250 cm werden vornehmlich eckige Gegenstände mit einer glatten Oberfläche erkannt. Sofern die Ultraschallwellen nicht frontal auf den jeweiligen Gegenstand treffen, kann es schnell zu großen Messfehlern kommen. Dies hat gewisse Auswirkungen auf die Konstruktion und die Programmierung des Roboters.

Bei der Konstruktion des Roboters sollte darauf geachtet werden, dass sich der Ultraschallsensor um 360 Grad rotieren lässt. Zudem sollte die Sicht des Ultraschallsensors nicht durch eigene Bauteile (z. B. Kabel, andere Sensoren) beeinträchtigt werden.

Mehr Informationen zum Ultraschallsensor finden sich auf dem Roberta-Portal unter: http://www.roberta-home.de/de/was-bietet-roberta/roberta-reihe/roberta-thema-hardware-des-lego-mindstorms-nxt-systems

Abbildung 16: Ultraschallsensor

Im das Projekt wird der Ultraschallsensor dazu verwendet, um Hindernisse auf der Fahrbahn detektieren zu können.

Lichtsensor

Der Lichtsensor (siehe Abbildung 17) ist in der Lage, den Helligkeitswert zu bestimmen. Als Rückgabewert wird ein prozentualer Wert der Differenz zwischen schwächstem und stärkstem Licht zurückgegeben. Der Lichtsensor besteht aus einer Diode und dem Lichtmesser und kann in zwei verschiedenen Modi betrieben werden. Im ersten Modus misst der Sensor nur die Helligkeit. Im zweiten Modus hingegen sendet die Lampe Lichtstrahlen aus, deren Reflektion aufgefangen wird. Dadurch wird die Helligkeit der reflektierenden Fläche gemessen. Zu beachten ist, dass aufgrund unterschiedlicher Lichtbedingungen an verschiedenen Orten die gemessenen Werte stark voneinander abweichen können.

Um dies zu verhindern, sollte der Roboter jedes Mal mit Hilfe eines geeigneten Algorithmus die Lichtsensoren vor Inbetriebnahme der eigentlichen Aufgabe des Roboters kalibrieren.

Abbildung 17: Lichtsensor

Lichtsensoren am Roboter

Die beiden Lichtsensoren (siehe Abbildung 18-19) wurden dazu verwendet, um der schwarzen Linie des Transportwegs zu folgen.

Abbildung 18: Lichtsensoren (Frontansicht)

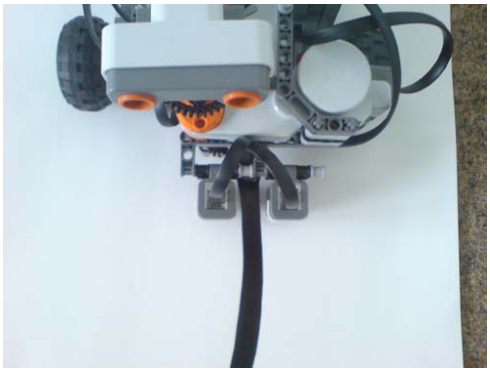


Abbildung 19: Lichtsensoren (Vogelperspektive)

Tastsensor

Beim Tastsensor (siehe Abbildung 20), handelt es sich um einen sehr einfachen Sensor. Dieser ist ausschließlich in der Lage zu erkennen, ob er betätigt wurde oder nicht.

Abbildung 20: Tastsensor

Tastsensor am Roboter

Der Tastsensor wurde für das Experiment als Not-Aus Schalter (siehe Abbildung 21-22) implementiert, welcher den Roboter veranlasst, in einen sicheren Zustand überzugehen. Dieser ist von allen Seiten gut sichtbar und gut erreichbar.

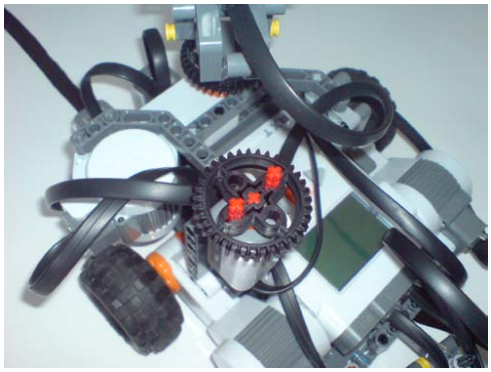


Abbildung 21: Not-Aus-Schalter (Vogelpespektive)

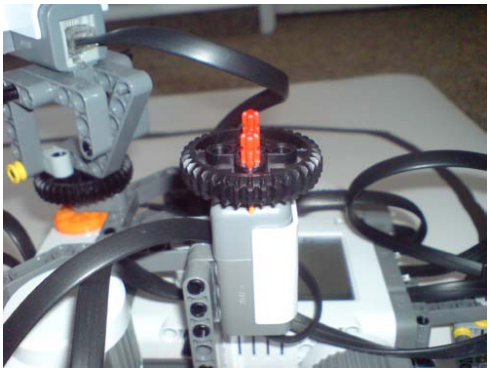


Abbildung 22: Not-Aus-Schalter (Seitenansicht)

Redundanter Not-Aus-Schalter

Als redundanter Not-Aus-Schalter wurde der orangefarbene EN-TER-Button (siehe Abbildung 23) der NXT-Steuereinheit implementiert.

Abbildung 23: Sekundärer Not-Aus-Schalter

Implementierung

Um die in der Anforderung gestellten Muss-Kriterien zu erfüllen, wurden 9 Klassen (siehe Tabelle 2) in der Programmiersprache Java/leJOS implementiert. Jede dieser Klassen wurde als Thread implementiert und beinhaltet eine spezifische Aufgabe, die im Folgenden erläutert wird.

Übersicht

Klassenname	Beschreibung
Execute	Ausführbare Hauptklasse
StartupCheck	Klasse zur Überprüfung der Funktionalität der Sensoren und Motoren vor Inbetriebnahme des Roboters
RCheck	Klasse zur Überprüfung der Funktionalität der Sensoren während des Betriebs
ODetection	Klasse zur Detektierung von Hindernissen auf der Fahrbahn
VMonitoring	Klasse zur Überprüfung der Versorgungsspannung
EmergenyStop	Klasse zur Implementierung des Not-Aus-Schalters
LineFo	Klasse zur Navigation mit 2 Lichtsensoren
LineFoS3	Klasse zur Navigation mit dem Lichtsensor am Sensorport 3 (falls ein Lichtsensor ausfällt)
LineFoS4	Klasse zur Navigation mit dem Lichtsensor am Sensorport 4 (falls ein Lichtsensor ausfällt)

Tabelle 2: Klassenübersicht

Sicherheitsfunktionen

Jede in Tabelle 2 dargestellte Klasse implementiert eine oder mehrer Roboterfunktionen. Der Großteil dieser Funktionen, entspricht den in Abschnitt »Sicherheitsanforderungen« vorgestellten Sicherheitsfunktionen (siehe Abbildung 2). Um die an den Transportroboter gestellten Sicherheitsanforderungen gewährleisten zu können, müssen die Sicherheitsfunktionen implementiert werden. Abbildung 24 zeigt, welche Klasse in diesem Experiment, welche Sicherheits-funktion implementiert.

Sicherheit von Menschen

Die Klassen ODetection, EmergencyStop und RCheck sorgen für die Kollisionsvermeidung. Die Klasse ODetection greift dabei auf einen Ultraschallsensor zu. Die Klasse EmergencyStop implementiert mit Hilfe eines Tastsensors eine Not-Aus-Einrichtung, welche im Notfall von Personen ausgelöst werden kann. Die Klasse RCheck sorgt indirekt für die Sicherheit, da sie die Sensoren auf ihre Funktionsfähigkeit überprüft. Fällt bspw. der Ultraschallsensor aus, wären die Sicherheit von Menschen, materiellen Sachen und die des Roberts nicht mehr gegeben. Die Klasse RCheck registriert diesen Ausfall und teilt dies dem Programm mit, welches entsprechend darauf reagiert (z. B. durch das Beenden des Programms).

Sicherheit von materiellen Sachen

Die Sicherheit von materiellen Sachen sowie die eigene Sicherheit des Roboters, wird durch die beiden Klassen ODetection und RCheck gewährleistet.

Betriebsfähigkeit

Die eigene Betriebsfähigkeit ist durch die Klassen VMonitoring, StartupCheck und RCheck gegeben. VMonitoring überwacht die Energieversorgung des Roboters. Bei einem definierten Tiefstand verlässt der Roboter beim nächsten Be- und Entladepunkt die Fahrbahn und betritt die Wartungszone. StartupCheck überprüft vor Inbetriebnahme des Roboters alle Motoren und Sensoren.

Aufgaben

Die vorprogrammierten Aufgaben, die der Roboter zu erfüllen hat, wie beispielsweise dem Folgen der Fahrbahn, werden durch die Klassen LineFo, LineFoS3 und LineFoS4 implementiert. Dabei handelt es sich um keine Sicherheitsfunktionen.

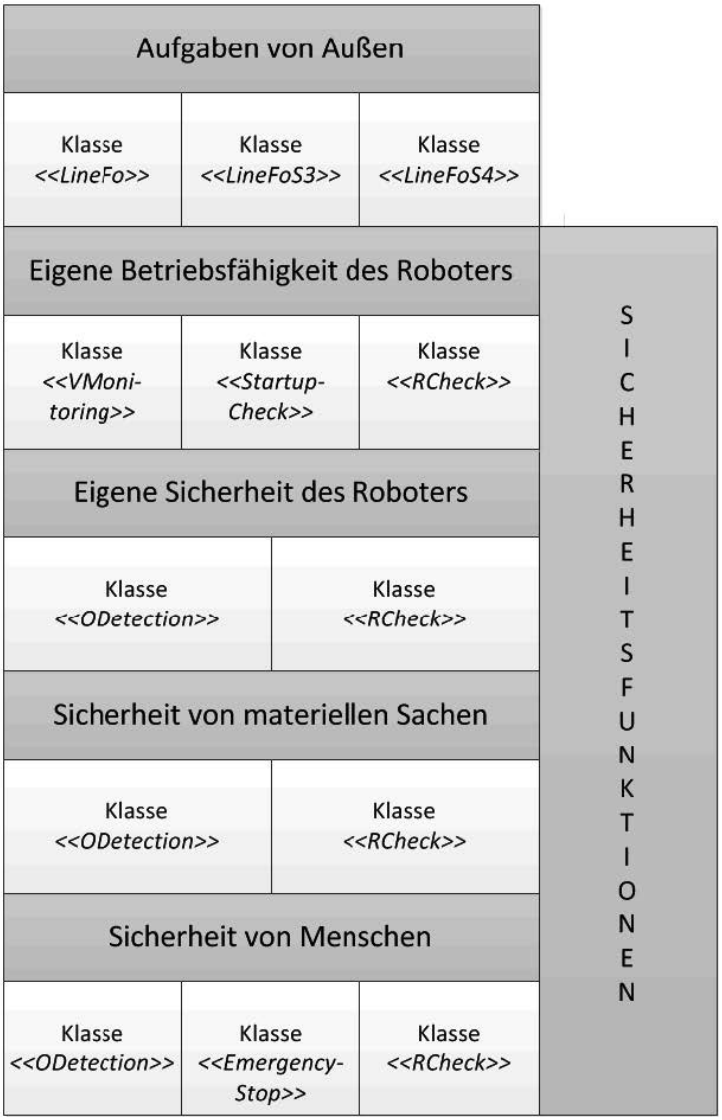


Abbildung 24: Implementierung der Sicherheitsfunktionen

Ermittlung des PLr

Mit Hilfe der im Abschnitt vier »Steuerungskomponenten« dargestellten Vorgehensweise (s. Abbildung 6), wird das benötigte Performance Level (PLr) der Steuerungskomponente für den bereits zuvor dargestellten Transportroboter ermittelt. Auf Grundlage des Ausfalls der wichtigsten sicherheitstechnischen Maßnahme wird das PLr berechnet. Für den Transportroboter ist dies der Ultraschallsensor, welcher die zwei wichtigsten Sicherheitsfunktionen (siehe Abbildung 2) Sicherheit von Maschinen und materiellen Sachen gewährleistet.

Szenario US-Sensor fällt aus

Auf Grund der geringen Masse und des geringen Ladevolumens des Transportroboters, ist bei einer Kollision mit Personen von einer leichten (reversiblen) Verletzung auszugehen (S1). Die Fahrwege des Transportroboters sind für Personen frei zugänglich. Deshalb muss mit einem relativ häufigen Aufenthalt von Personen im Gefahrenbereich gerechnet werden. Somit ist die Häufigkeit der Gefährdung sehr hoch (F2). Da der Roboter mit niedriger Geschwindigkeit (in der Regel 2-4 km/h) fährt, hat eine Person bei Herannahen des Roboters meist die Möglichkeit auszuweichen (P1). Somit ergibt sich ein erforderlicher PLr-Wert = B (siehe Abbildung 25).

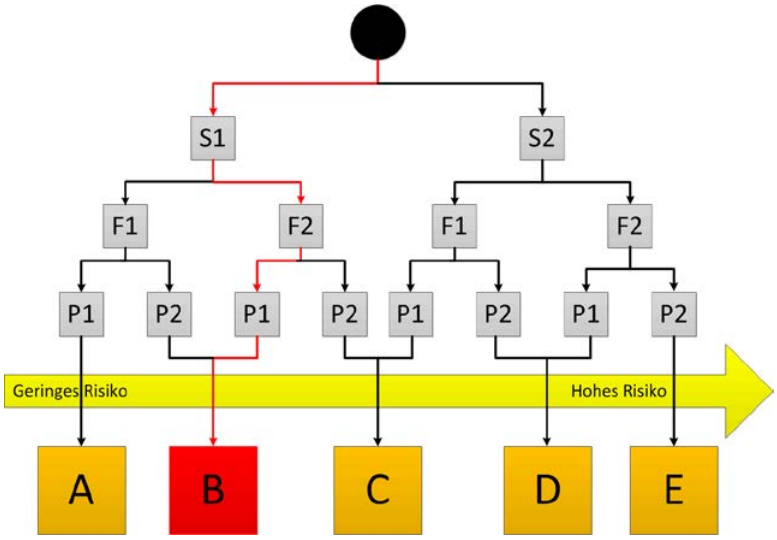


Abbildung 25: PLr Wert bei Ausfall des Ultraschall-Sensors

NXT-Steuereinheit

Nach der Risikoanalyse gemäß DIN EN ISO 13849 wird für den Transportroboter eine Steuerungskomponente mit dem PL = B benötigt. Bei einer Steuerungskomponente mit dem PL = B, handelt es sich um eine einkanalige Steuerungskomponente ohne Redundanz. Diese ist durch die NXT-Steuereinheit gegeben.

Vor Inbetriebnahme

Vor Inbetriebnahme des Roboters wird die Funktionalität der Sensoren und der Motoren überprüft. Im ersten Schritt wird die Funktionalität der beiden Not-Aus-Schalter getestet. Dabei wird der Anwender über das LC-Display aufgefordert, die beiden Not-Aus Schalter (Tastsensor und orangener ENTER-Button) der Reihe nach zu betätigen (siehe Abbildung 26-27). Sofern einer der beiden Not-Aus-Schalter nicht funktioniert, wird die Inbetriebnahme des Roboters abgebrochen und das Programm fährt herunter. Diese Kontrollfunktion wird vor allen anderen Funktionen in der Hauptklasse Execute ausgeführt. Erst nach dem Ende dieser Kontrollfunktion können die anderen Funktionen (Threads) starten.

Abbildung 26: Überprüfung des primären Not-Aus-Schalters

Abbildung 27: Überprüfung des sekundären Not-Aus-Schalters

StartupCheck

Im nächsten Schritt werden die 3 Motoren überprüft, indem diese für kurze Zeit im und entgegen dem Uhrzeigersinn, rotiert werden. Anhand der ermittelten Rotationen der Motoren wird festgestellt, ob alle Motoren funktionstüchtig sind. Zudem werden nach Überprüfung der Motoren, der Ultraschallsensor und die beiden Lichtsensoren auf deren Funktionalität überprüft. Dabei wird getestet, ob die Sensoren, nach einer Testmessung einen Wert größer 0 zurück-liefern. Sollte dies nicht der Fall sein, erfolgt eine akustische sowie eine textuelle Fehlermeldung durch das NXT-Display und das Programm wird heruntergefahren. Aus der Fehlermeldung auf dem Display, erfährt der Anwender, ob ein Fehler bei den Motoren, dem Ultraschallsensor oder den Lichtsensoren vorliegt (siehe Abbildung 28-29). Folgt zudem kein Rückgabewert, da die Sensoren oder Motoren evtl. mechanisch defekt sind oder nicht angeschlossen sind, folgt ebenfalls eine Fehlermeldung. Die Überprüfung der Motoren und Sensoren ist in der Klasse StartupCheck implementiert.

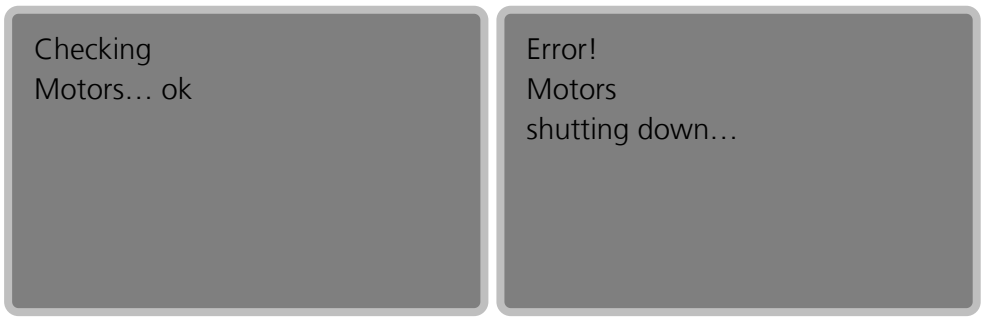


Abbildung 28: Motoren in Ordnung (links) und Motor(en) defekt (rechts)

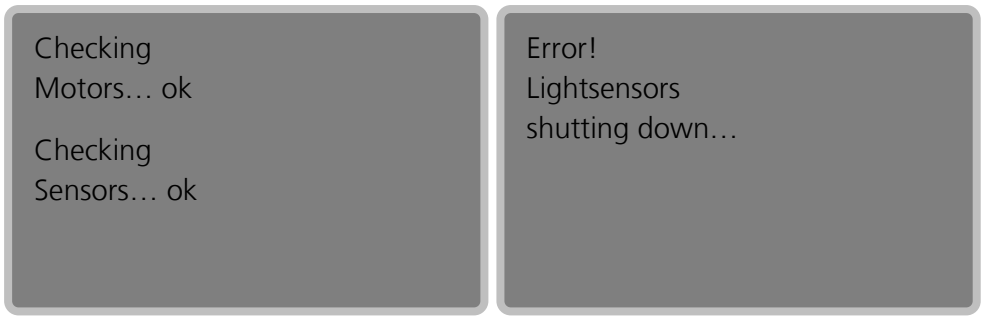


Abbildung 29: Sensoren in Ordnung (links) und Sensor(en) defekt

Fahrbahn mit 2 Lichtsensoren folgen

Wenn alle Motoren und Sensoren funktionstüchtig sind, nimmt der Roboter, nach einer akustischen Signalisierung die Fahrt auf. Mit Hilfe der beiden Lichtsensoren ist der Roboter in der Lage, der schwarzen Linie zu folgen. Dabei stellt die schwarze Linie eine Art Schiene dar, die zwischen den beiden Lichtsensoren platziert sein muss und den Roboter führt. Solange die beiden Sensoren auf weißem Grund sind, fährt der Roboter geradeaus. Sobald einer der Sensoren die schwarze Linie registriert, lenkt der Roboter in die Richtung des registrierenden Lichtsensors ein. Das Prinzip wird in den

Abbildung 30-32 verdeutlicht. Somit ist der Roboter in der Lage, die vorgegebene Fahrbahn abzufahren. Dieses Verhalten wurde in der Klasse LineFo implementiert.

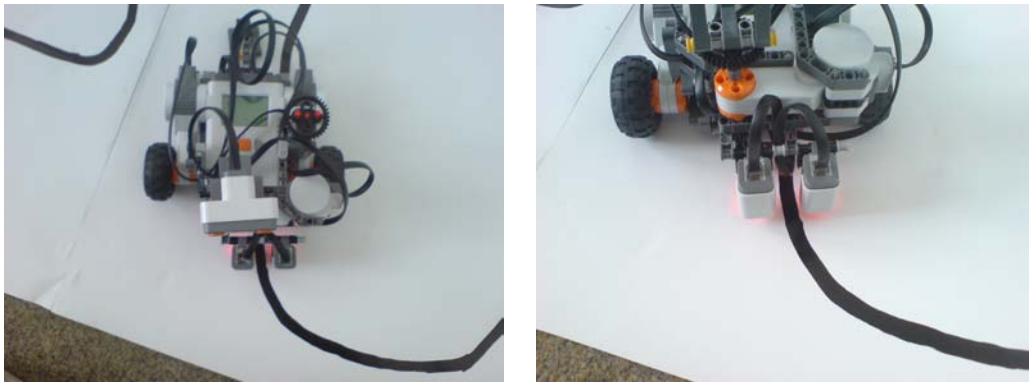


Abbildung 30: Keine Detektion, der Roboter fährt weiter geradeaus

Abbildung 31: Rechter Lichtsensor registriert die schwarze Linie

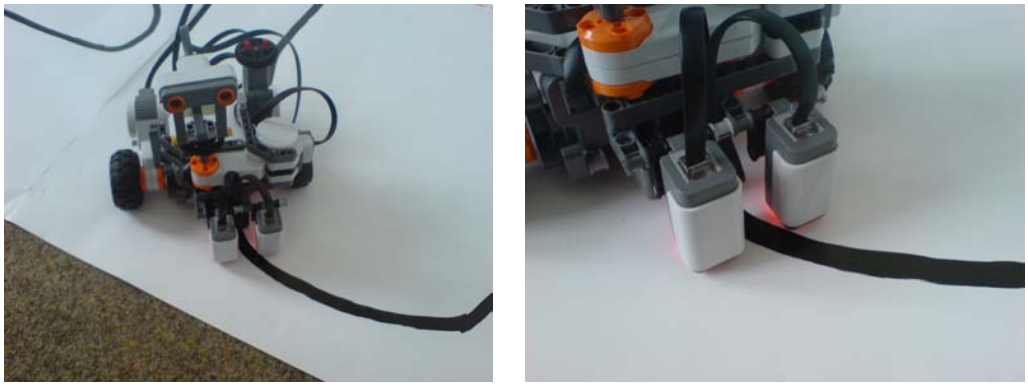


Abbildung 32: Der Roboter dreht, bis der Lichtsensor die schwarze Linie nicht mehr registriert

Ausfall eines Lichtsensors

Sollte einer der beiden Lichtsensoren ausfallen, wird dies durch die Klasse RCheck registriert. Daraufhin ist der Roboter in der Lage, mit Hilfe eines einzelnen Lichtsensors die Fahrbahn bis zum nächsten Be- und Entladepunkt weiter zu verfolgen (siehe Abbildung 33). Dort angekommen verlässt der Roboter die Fahrbahn und fährt in die Wartungszone um sich daraufhin selbstständig abzuschalten. In der Wartungszone kann der Roboter dann gegebenenfalls gewartet bzw. repariert werden ohne dass dieser die Transportstrecke blockiert.

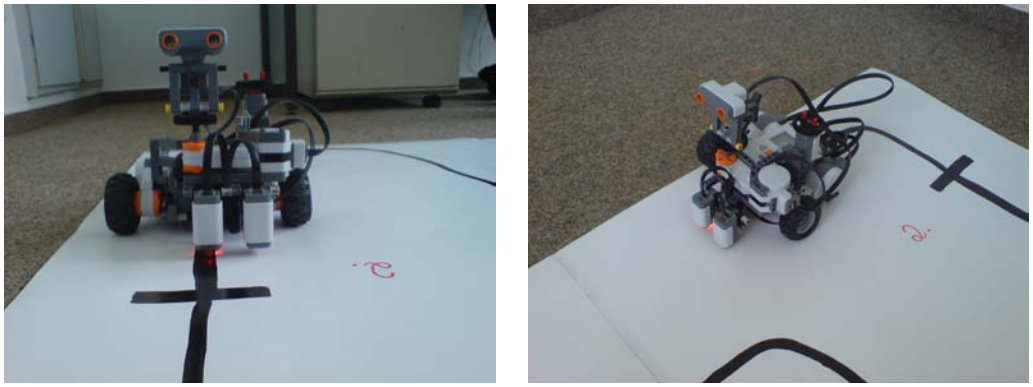


Abbildung 33: Der Roboter navigiert mit einem Lichtsensor bis zum nächsten Be- und Entladepunkt und verlässt daraufhin die Strecke

Fahrbahn mit einem Lichtsensor folgen

Im Gegensatz zur Implementierung mit 2 Lichtsensoren funktioniert das Prinzip mit einem Lichtsensor anders. Solange der Licht-sensor die schwarze Linie detektiert, fährt der Roboter geradeaus (siehe Abbildung 34). Trifft er nun auf eine Kurve, befindet sich der Lichtsensor nicht mehr länger über der schwarzen Linie und der Roboter hält an (siehe Abbildung 35). Anschließend dreht er sich solange um die eigene Achse, bis er die Linie wiedergefunden hat oder eine vorgegebene Zeitspanne t abgelaufen ist (siehe Abbildung 36). Diese Zeitspanne soll verhindern, dass der Roboter sich um 180 Grad dreht und damit die Fahrt in die falsche Richtung fortsetzt. Hat er die Linie wiedergefunden, fährt er weiter gerade aus. Ist dies nicht der Fall, rotiert er in entgegengesetzter Richtung, bis er die Linie wiederentdeckt (siehe Abbildung 37) oder die Zeitspanne t abgelaufen ist.

Damit der Roboter möglichst wenige Kurskorrekturen vornehmen muss, merkt er sich die Drehrichtung bei der die Linie wiederentdeckt wurde und beginnt bei nächsten Kurskorrekturen eine Rotation mit derselben Richtung. Dieses Verhalten wurde in den Klassen LineFoS3 und LineFoS4 implementiert. Abhängig davon, ob der Lichtsensor an Port 3 oder Port 4 ausgefallen ist, wird eine der beiden Klassen als Thread gestartet.

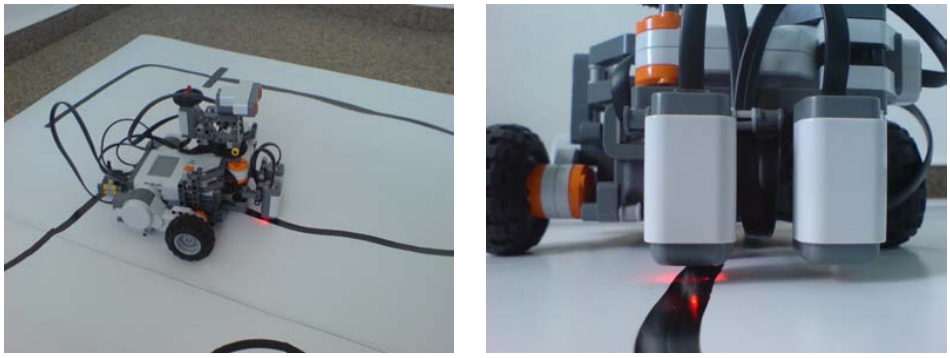


Abbildung 34: Der Lichtsensor registriert die Linie, somit fährt der Roboter geradeaus

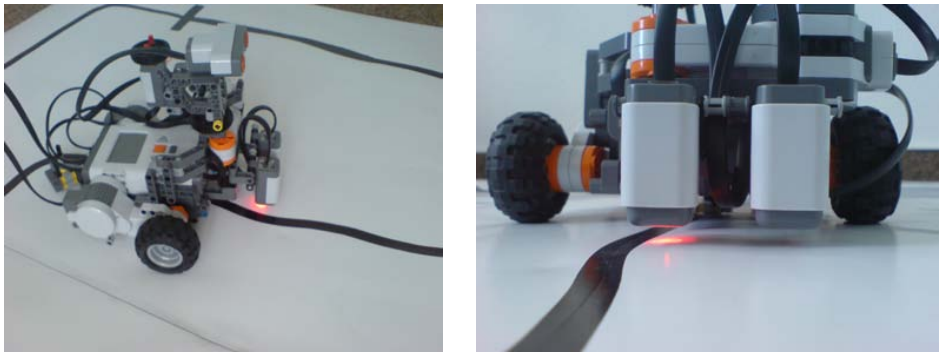


Abbildung 35: Der Lichtsensor registriert nicht mehr die schwarze Linie, somit kommt der Roboter zum Stillstand

Abbildung 36: Roboter rotiert, bis die Zeitspanne t abgelaufen ist

Abbildung 37: Roboter rotiert in entgegengesetzter Richtung, bis die schwarze Linie wieder registriert wird

Ausfall mehrer Sensoren

Sofern beide Lichtsensoren und/oder der Ultraschallsensor ausfallen, wird dies durch die Klasse RCheck registriert, wird der Roboter auf der Stelle zum Stillstand gebracht und anschließend heruntergefahren. Durch einen Ausfall der Lichtsensoren und des Ultraschallsensors wäre der Roboter nicht mehr manövrierfähig. Die wichtigste Sicherheitsfunktion »Sicherheit von Menschen« (siehe Abbildung 2) ist somit nicht mehr gewährleistet, was einen sicheren Betrieb des Roboters unmöglich macht.

Hindernisse auf der Fahrbahn

Um Hindernisse auf der Fahrbahn erkennen zu können, scannt der Ultraschallsensor alle 120-200 ms[20] in Fahrtrichtung die Gegend nach Objekten. Sobald sich ein Objekt auf eine Entfernung von 30 cm dem Roboter nähert, kommt dieser zum Stillstand (siehe Abbildung 38). Solange sich das Hindernis weiterhin im Weg befindet, verharrt der Roboter an Ort und Stelle und versucht, sich mit einem akustischen Signal sowie einer Textnachricht auf dem LC-Display bemerkbar zu machen. Sobald das Hindernis nicht mehr registriert wird, nimmt der Roboter seine Fahrt wieder auf. Dieses Verhalten wurde in der Klasse ODetection implementiert.

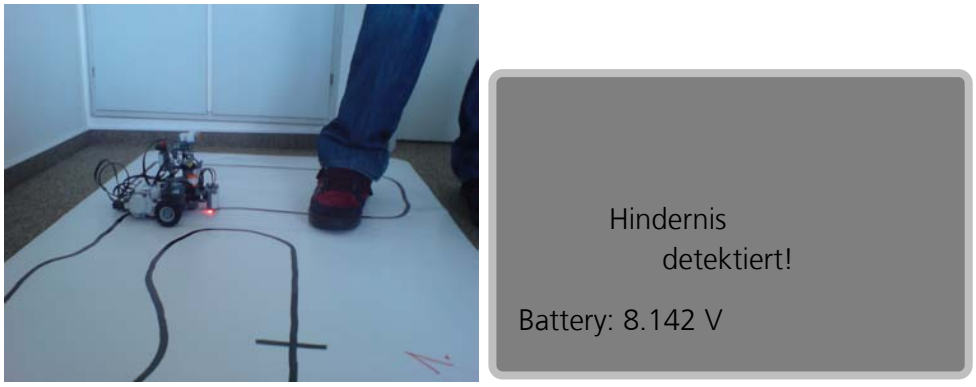


Abbildung 38: Der Roboter detektiert ein Hindernis auf der Fahrbahn

[20] Michael Gerhäuser, September 2010, http://nxt.michael-gerhaeuser.de/4_sensoren.html

Überwachung der Energieversorgung

Als weiteres Sicherheitsfeature wurde in der Klasse VMonitoring, eine Überwachung der Energieversorgung implementiert. Sobald am Roboter nur noch eine Versorgungsspannung von unter 6,5 V anliegt, verlässt der Roboter am nächsten Be- und Entladepunkt die Fahrbahn, um sich daraufhin abzuschalten (siehe Abbildung 39). Des Weiteren macht sich der Roboter bei niedrigem Batteriestand, durch ein akustisches Signal bemerkbar. Zusätzlich wird der Ladestand der Batterie, während des Betriebes auf dem LC-Display angezeigt.

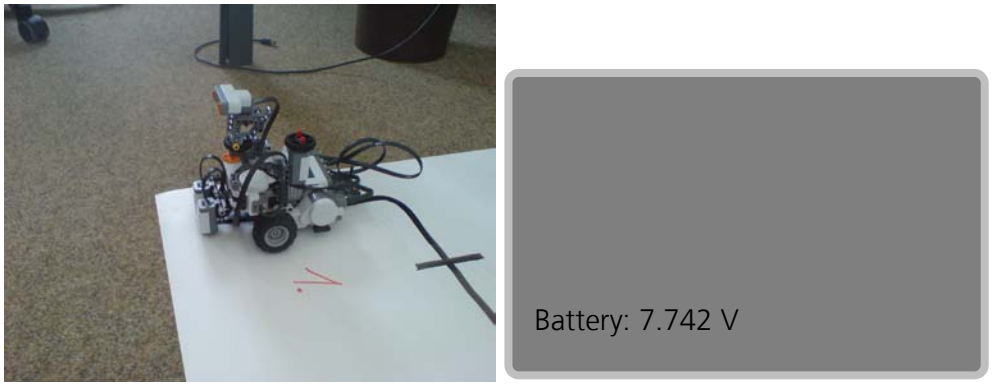


Abbildung 39: Der Roboter verlässt die Fahrbahn und schaltet sich ab

Not-Aus Schalter

Sollte der Roboter trotz der integrierten Sicherheitsfunktionen ein unkontrolliertes Verhalten zeigen, kann dieser im Notfall durch betätigen einer der beiden Not-Aus-Schalter in einen sicheren Zustand versetzt werden (siehe Abbildung 40). In diesem Zustand werden alle Motoren abgeschaltet, die Sicherheitsfunktionen jedoch bleiben aufrechterhalten. Der Roboter kann durch wiederholtes Drücken eines der beiden Not-Aus-Schalter vollständig ausgeschaltet werden oder durch Drücken des ESCAPE-Buttons, welcher sich unterhalb des ENTER-Buttons befindet, wieder aktiviert werden. Nach einer akustischen Signalisierung nimmt der Roboter daraufhin seine Fahrt wieder auf. Die Implementierung der Not-Aus-Schalter wurde in der Klasse EmergencyStop realisiert.

Abbildung 40: Nach Betätigen des Not-Aus-Schalters wechselt der Roboter in einen sicheren Zustand

Programmablauf

Nachdem zuvor die Aufgaben der einzelnen Klassen (Threads) erläutert wurden, betrachten wir nun den internen Ablauf des Pro-grammes. Dabei soll deutlich werden, welche Klassen (Threads) miteinander in Beziehung stehen. Nach einem Ereignis, wie bspw. Ausfall eines Sensors, benachrichtigt eine Klasse eine oder mehrere andere Klassen, um entsprechend auf dieses Ereignis zu reagieren.

Programmablauf beim Systemstart

Abbildung 41 skizziert den internen Programmablauf beim Start des Transportroboters.

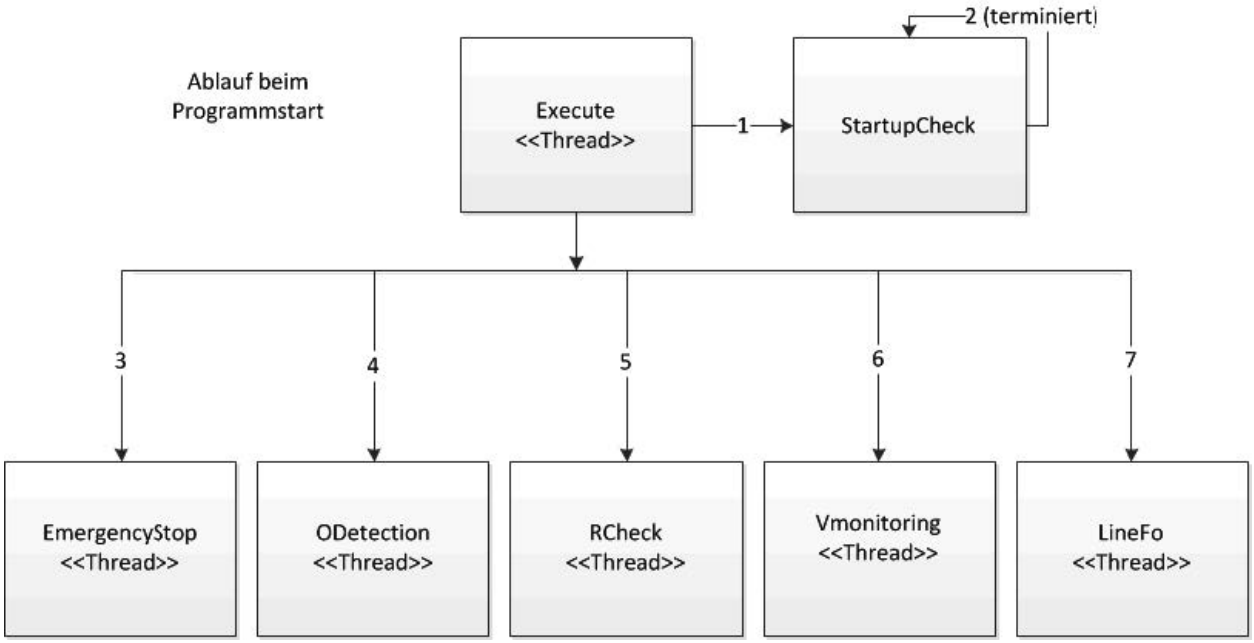


Abbildung 41: Programmstart

Als erstes startet die Hauptklasse Execute, die die Klasse StartupCheck aufruft. Die Klasse StartupCheck überprüft die Funktionalität der Motoren und Sensoren. Sobald die Überprüfung erfolgreich abgeschlossen ist und somit alle Motoren und Sensoren funktionsfähig sind, starten die Klassen EmergencyStop, ODetection, RCheck und VMonitoring, welche im Zusammenspiel mit den Sicherheitskomponenten (z.B. Ultraschallsensor) die Sicherheitsfunktionen (siehe Abbildung 2) implementieren. Die Klassen EmergencyStop und ODetection gewährleisten die Sicherheit von Menschen.
Die Klasse ODetection gewährleistet zudem die Sicherheit von materiellen Sachen, sowie die des Roboters. Die Klassen RCheck und VMonitoring sorgen für die Betriebsfähigkeit des Roboters. Genau betrachtet sorgt zudem die Klasse RCheck indirekt für die Sicherheit von Menschen, materiellen Sachen und des Roboters selbst (s. Abbildung 24). Aufbauend darauf startet als letztes die Klasse LineFo, welche die eigentlichen Arbeitsaufgaben des Roboters implementiert.

Defekter Sensor oder Motor beim Systemstart

Sollte die Klasse StartupCheck einen defekten Sensor oder Motor während des Systemstarts registrieren, veranlasst die Klasse StartupCheck die Hauptklasse Execute zur Terminierung. Das Programm wird beendet (siehe Abbildung 42).

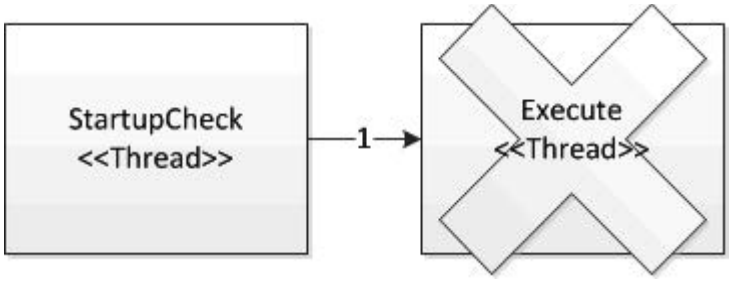


Abbildung 42: StartupCheck veranlasst Execute zum Beenden

Programmablauf bei Ereignissen

Während des Betriebes können unterschiedliche Ereignisse eintreffen unter anderem:

- Detektion eines Hindernisses
- Auslösen des Not-Aus-Schalters
- Niedriger Akkustand
- Ausfall eines Sensors

Je nach, dem welches Ereignis eintrifft, unterscheidet sich der Programmablauf.

Detektion eines Hindernisses

Solange der Ultraschallsensor ein Hindernis registriert, stoppt die Klasse ODetection die Klassen LineFo, LineFoS1 und LineFoS4 (s. Abbildung 43).

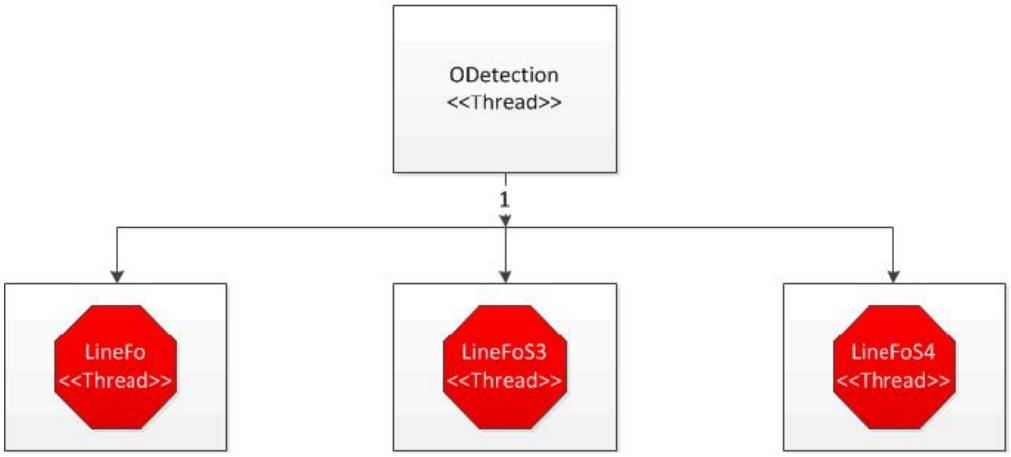


Abbildung 43: Ultraschall-Sensor registriert ein Objekt

Sofern der Ultraschallsensor kein Hindernis mehr detektiert, werden die drei oben erwähnten Klassen wieder gestartet.

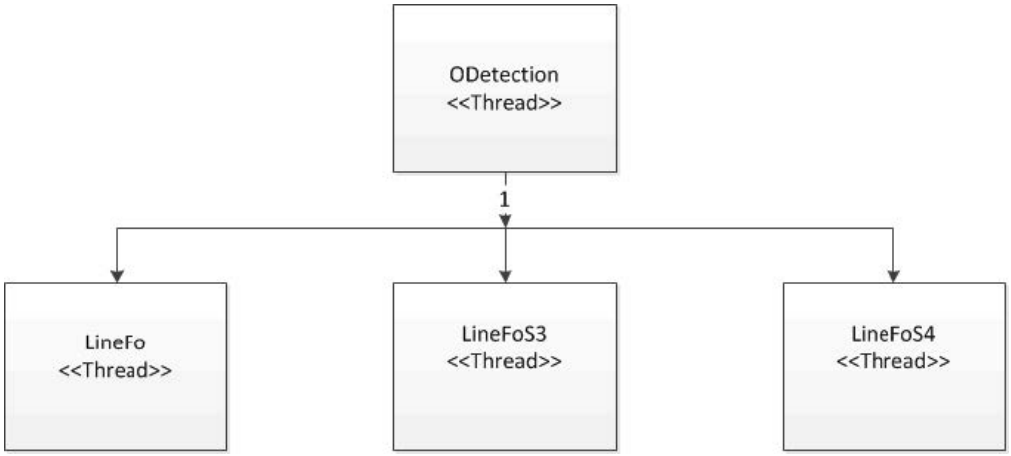


Abbildung 44: Ultraschall-Sensor registriert kein Objekt

Auslösen des Not-Aus-Schalters

Nach einmaligem Betätigen einer der beiden Not-Aus-Schalter wird der Roboter in einen sicheren Zustand versetzt. Dabei stoppt die Klasse EmergencyStop die Klassen LineFo, LineFoS3 und LineFoS4 (siehe Abbildung 45). Damit mit wird die Weiterfahrt des Roboters verhindert. Die Sicherheitsfunktionen bleiben weiter aktiv.

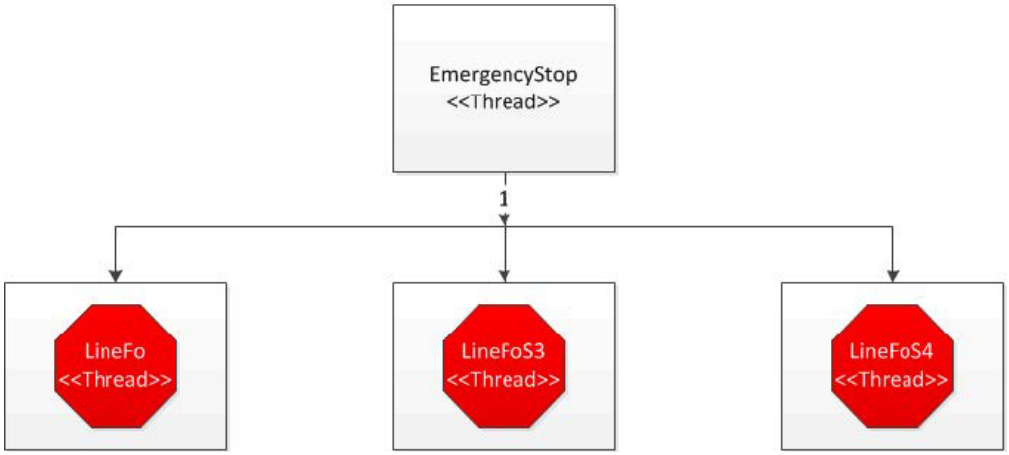


Abbildung 45: Übergang in den sicheren Zustand

Betätigen des ESCAPE-Buttons

Wird im sicheren Zustand der ESCAPE-Button gedrückt, nimmt der Roboter wieder seine Fahrt auf. Dabei startet die Klasse EmergencyStop wieder die Klassen LineFoS3 und LineFoS4 (siehe Abbildung 46).

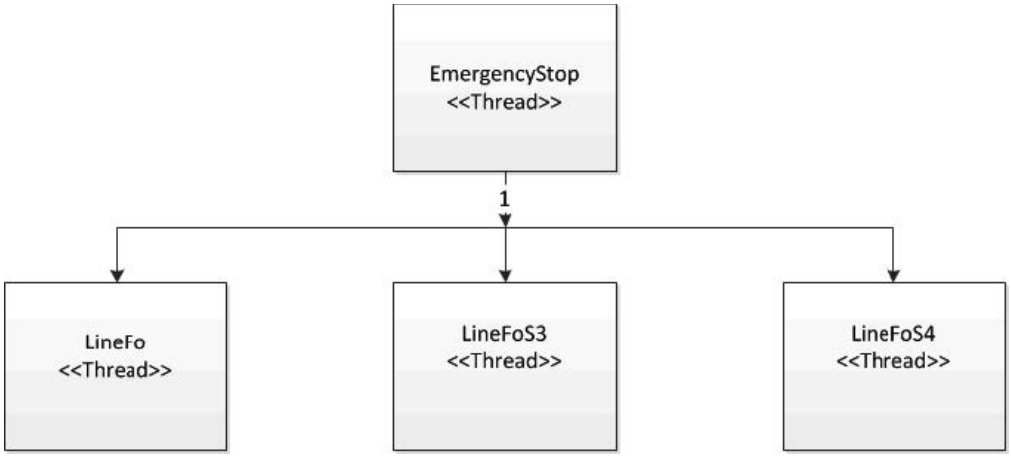


Abbildung 46: Roboterfunktionen werden wieder gestartet

Auslösen des Not-Aus-Schalters im sicheren Zustand

Sofern sich der Roboter durch Auslösen einer der beiden Not-Aus-Schalter im sicheren Zustand befindet und der Not-Aus-Schalter erneut betätigt wird, endet das Programm. Die Klasse EmergencyStop veranlasst die Hauptklasse Execute zum sofortigen Beenden des Programms (s. Abbildung 47).

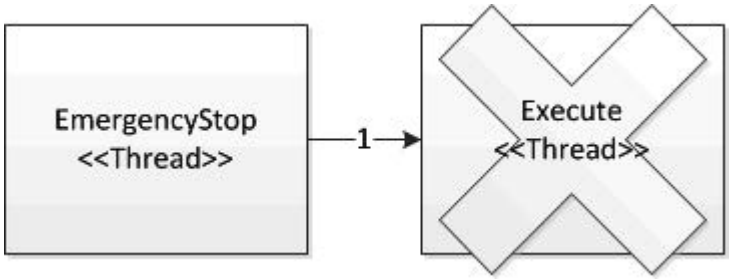


Abbildung 47: Die Klasse EmergencyStop beendet das Programm

Niedriger Akkustand

Sollte der Akkustand des Roboters auf einen definierten Tiefstand fallen, so wird der Roboter veranlasst, die Fahrbahn beim nächsten Be- und Entladepunkt zu verlassen, um die Wartungszone zu betreten. Die Klasse VMonitoring signalisiert der Klasse LineFo, dass der Akkustand zu gering ist (siehe Abbildung 48).

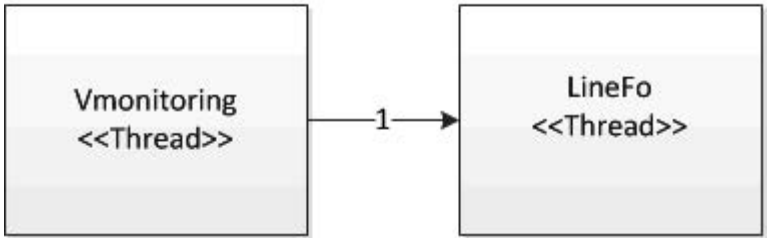


Abbildung 48: Die Klasse VMonitoring signalisiert den schwachen Akku-stand

Ausfall eines Lichtsensors

Registriert die Klasse RCheck, dass einer der beiden Lichtsensoren ausgefallen ist, signalisiert RCheck der Hauptklasse Execute, welcher der Sensoren ausgefallen ist. Anschließend stoppt Execute die Klasse LineFo, welche für die Fahrt mit Hilfe von zwei Lichtsensoren verantwortlich ist. Abhängig davon, welcher der beiden Lichtsensoren ausgefallen ist, startet Execute entweder die Klasse LineFoS3, bei Ausfall des rechten Sensors (Frontansicht) oder die Klasse LineFoS4 bei Ausfall des linken Sensors (siehe Abbildung 49).

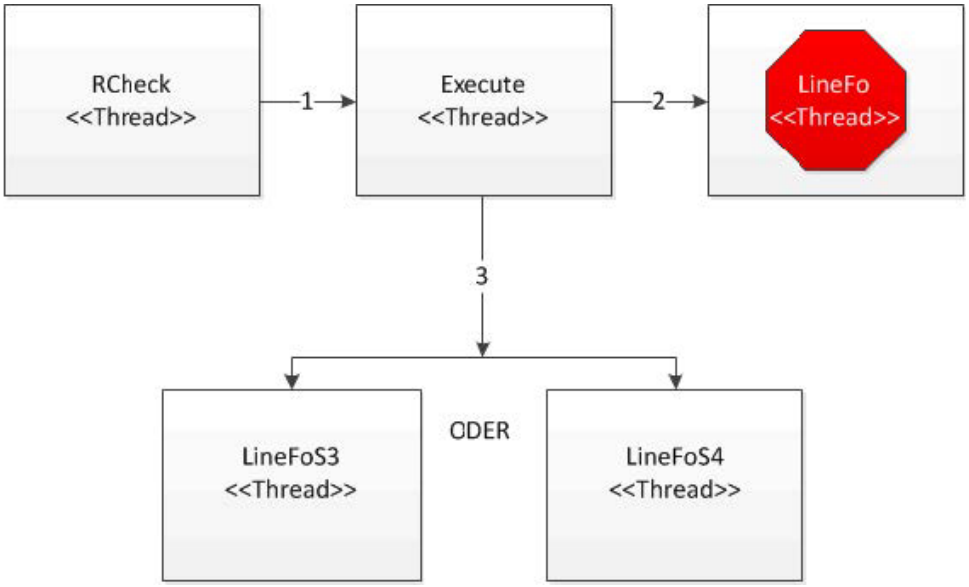


Abbildung 49: Ein Lichtsensor fällt aus

Ausfall mehrerer Sensoren

Registriert die Klasse RCheck den Ausfall beider Lichtsensoren und/oder des Ultraschallsensors, signalisiert RCheck der Hauptklasse Execute den Ausfall. Daraufhin leitet Execute den sofortigen Programmabbruch ein (siehe Abbildung 50).

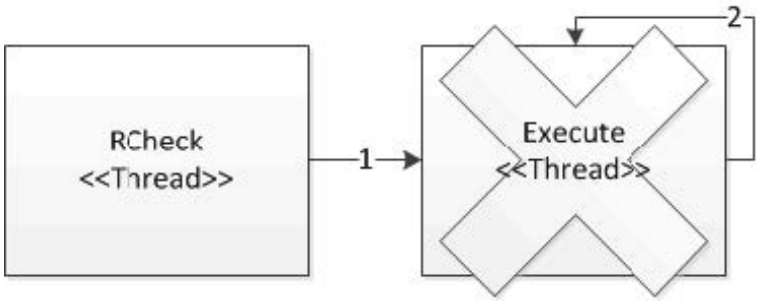


Abbildung 50: Ausfall mehrerer Sensoren

3. Zusammenfassung

Sicherheitsrichtlinien für Serviceroboter

Diese Ausarbeitung wurde mit der Feststellung begonnen, dass sich der Trend stark in Richtung autonomer mobiler Roboter im Sektor Serviceroboter entwickelt. Diese Roboter teilen sich in der Regel den Arbeitsbereich und interagieren mit Menschen. Um Unfälle zu vermeiden, werden im Sektor Serviceroboter allgemein gültige Sicherheitsrichtlinien gefordert. Allgemeine Informationen zu Sicherheitsrichtlinien mobiler autonomer Roboter existieren jedoch nur ansatzweise. Im Gegensatz zu den Industrierobotern, wo bereits seit 1993 spezielle Normen existieren, fehlen diese für autonome mobile Roboter. Durch das Fehlen dieser speziellen Sicherheitsnormen für autonome mobile Roboter wird die Entwicklung von Servicerobotern gehemmt, da ein entsprechender Orientierungsrahmen für Entwickler fehlt. Zudem bleibt die Haftungsfrage bei einem Unfall durch Serviceroboter ungeklärt. Zusammenfassend wurde mit Hilfe dieser Ausarbeitung ein Orientierungsrahmen erstellt.

Inhalt

In Abschnitt »Sicherheitsanforderungen« wurden einige Sicherheitsfunktionen definiert, die durch einen Roboter gewährleistet werden müssen. Im darauffolgenden Abschnitt »Anforderungen – Normen und Richtlinien« wurde erläutert, welche Richtlinien und Normen von Bedeutung für autonome mobile Roboter sind. Zudem wurde die Haftung bei Missachtung der Richtlinien und Normen beschrieben. Daraufhin wurde in Abschnitt »Gefahrenanalyse« eine Methode zur Analyse der Gefährdungspotentiale vorgestellt, die von einem autonomen mobilen Roboter ausgehen. In Abschnitt »Steuerungskomponenten« wurde auf Grundlage von definierten Risikoparametern eine Methode zur Ermittlung der benötigten Steuerungskomponenten für einen autonomen mobilen Roboter vorgeführt. Im letzten Abschnitt »Experiment »Safe Mobile Transport Robot (SMTR)«« wurden die in den Abschnitten zuvor behandelten Grundlagen auf einen Lego NXT Mindstorm Roboter implementiert, welcher im Experiment als Transportroboter fungiert.

Literaturverzeichnis

Jürgen Adamy, Peter Bechtel
Sicherheit mobiler Roboter – Safety of Mobile Robots, Darmstadt, 2003

Dietz Sensortechnik
Sicherheit an Maschinen, Heppenheim, 2007

Bayerisches Staatsministerium für Wirtschaft, Infrastruktur, Verkehr und Technologie
Sicherheit von Maschinen, März 2005

DIN EN ISO 12100-1
Sicherheit von Maschinen –Grundbegriffe, allg. Gestaltungs-leitsätze – Teil 1: Grundsätzliche Terminologie, Methodologie, April 2004

DIN EN ISO 14121-1
Sicherheit von Maschinen – Risikobeurteilung – Teil 1: Leitsätze, Dezember 2007

DIN EN ISO 13849-1
Sicherheit von Maschinen – Sicherheitsbezogene Teile von Steuerungen – Teil1: Allg. Gestaltungsleitsätze, Dezember 2008

DIN EN 1525
Fahrerlose Flurförderzeuge und ihre Systeme, 1997

Rexroth Bosch Group
EN ISO 13849: Und sie gilt immer häufiger doch, Juli 2007

Wikipedia
Robotergesetze, August 2010
http://de.wikipedia.org/wiki/Robotergesetze

Spiegel Online
Roboter werden Fehlentscheidungen treffen, Juli 2010
http://www.spiegel.de/wissenschaft/technik/0,1518,705961,00.html

Deutsches Institut für Normung e.V.
Europäischer Binnenmarkt, September 2010
http://www.din.de/cmd?level=tpl-unterrubrik&cmssubrubid=47507&languageid=de

Wikipedia
Ada (Programmiersprache), September 2010
http://de.wikipedia.org/wiki/Ada_%28Programmiersprache%29

AdaCore
GNAT GPL for LEGO MINDSTORMS NXT, September 2010
http://libre.adacore.com/libre/tools/mindstorms/

Michael Gerhäuser

Sensoren, September 2010
http://nxt.michael-gerhaeuser.de/4_sensoren.html

IRobot

Meilensteine des Unternehmens, September 2010
http://www.irobot.com/de/about_irobot_at_glance.cfm

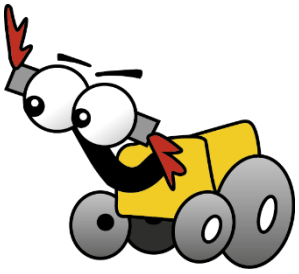
Wikipedia

Autonome mobile Roboter, September 2010
http://de.wikipedia.org/wiki/Autonome_mobile_Roboter

Leuze Electronic

Hierarchie der Europäischen Normen zur Maschinensicherheit, August 2010
http://www.arbeitssicherheit.leuze.de/a/a_07.html

Kontakt



Die Roberta-Initiative im Web

roberta-home.de

lab.open-roberta.org

FAQ rund um die Roberta-Initiative

roberta-home.de/faq

Informationen zum Datenschutz

roberta-home.de/datenschutz

Info

Verfasser: Andreas Brück, Thorsten Leimbach I Roberta-Initiative

© 2004-2018 Fraunhofer-Institut für Intelligente Analyse und Informationssysteme IAIS

Version: 1.0

Stand: Juli 2011

Warenzeichen

LEGO® und Mindstorms™ sind eingetragene Warenzeichen der Firma The LEGO Group.

Microsoft®, Windows® und Internet Explorer® sind eingetragene Warenzeichen der Firma Microsoft Corporation.

fischertechnik® ist ein eingetragenes Warenzeichen der Firma fischertechnik GmbH.

Java™ ist eingetragenes Warenzeichen der Firma SUN Microsystems Inc.

LabVIEW ist ein Produkt der Firma National Instruments.

Roberta, Open Roberta und NEPO sind eingetragen Warenzeichen der Fraunhofer-Gesellschaft e.V.

ROBERTA INITIATIVE