

Hochschule für angewandte Wissenschaften Coburg  
Fakultät Elektrotechnik und Informatik

Studiengang: Automatisierungstechnik und Robotik, Elektro- und Informationstechnik (Bc, M.Eng.), Erneuerbare Energien, Energietechnik und Erneuerbare Energien,   
Informatik (Bc, M.Sc.), Visual Computing

Softwarearchitektur für einen autonom fahrenden Roboter

Christof Kern

Frederik Schöpplein

Andreas Scharf

# Inhaltsverzeichnis

[Inhaltsverzeichnis 2](#_Toc70702336)

[Abbildungsverzeichnis 3](#_Toc70702337)

[1 Einleitung 4](#_Toc70702338)

[1.1 Kontext 4](#_Toc70702339)

[1.2 Motivation 4](#_Toc70702340)

[1.3 Aufgabenbeschreibung 4](#_Toc70702341)

[1.4 Aufbau 4](#_Toc70702342)

[2 Hauptteil 5](#_Toc70702343)

[2.1 Eigenschaften und Anforderungen 5](#_Toc70702344)

[2.2 Interakteure 5](#_Toc70702345)

[2.3 Softwaremodule 6](#_Toc70702346)

[2.4 Entwicklungsschritte 7](#_Toc70702347)

[3 Zusammenfassung 9](#_Toc70702348)

[Ehrenwörtliche Erklärung 10](#_Toc70702349)

# Abbildungsverzeichnis

[Abbildung 1: Anwendungsfalldiagramm 6](#_Toc70702412)

[Abbildung 2: Softwaremodule 6](#_Toc70702413)

# Einleitung

## Kontext

Im Rahmen des Praktikums des Moduls Autonome Eingebettete Systeme für die Fakultät Elektrotechnik Informatik der Hochschule Coburg soll ein autonom Fahrender Roboter programmiert werden.

## Motivation

Damit die Studierenden der Hochschule Coburg den Umgang mit Microchipprozessoren erlernen, sollen die Teilnehmer, im Praktikum des Fachs Autonome Eingebettete Systeme einen autonom-fahrenden Roboter programmieren. Um komplexe Softwaresysteme zu realisieren ist es üblich, dass von den Entwicklern des Systems eine Softwarearchitektur entworfen wird. Diese Softwarearchitektur dient dazu währenden der Entwicklung einen Überblick über Anforderungen, Interakteuren, Softwaremodule und Entwicklungsschritte zu erhalten.

## Aufgabenbeschreibung

Folgende Teilaufgaben müssen erfüllt werden, um das oben beschriebene Ziel zu erreichen.

Die Evaluierung der Anforderungen und Eigenschaften des Roboters. Das Aufnehmen aller Interakteure des Systems. Die Aufteilung der Robotersoftware in einzelne Softwaremodule. Und im letzten Schritt zum Ermitteln einer Softwarearchitektur für den Roboter, sollen die einzelnen Entwicklungsschritte für das System zusammengefasst werden.

## Aufbau

Im Abschnitt 2.1 werden die Eigenschaften und Anforderungen des Systems ermittelt. Dazu wurden die Vorlesungsfolien und teilweise die Datenblätter verwendet. Im darauffolgenden Abschnitt werden die Interakteure mit dem Roboter festgelegt und die Interaktionen von diesen mit dem System. Um eine übersichtliche Robotersteuerung zu realisieren, wird die Software im Abschnitt 2.3 in verschiedene Module aufgeteilt. Der Abschnitt 2.4 die einzelnen Entwicklungsschritte und erarbeitet einen groben Zeitplan für die Entwicklung.

# Hauptteil

## Eigenschaften und Anforderungen

Um sich im Raum orientieren zu können besitzt der von der Hochschule bereitgestellten Roboter drei Ultraschallsensoren. Diese können Entfernungen im Raum feststellen. Zur Ermittlung einer Distanz mit einem Ultraschallsensor, wird ein 10µs langes Signal ausgesandt und die an Gegenständen reflektierte Antwort aufgenommen. Damit lässt sich dann der Abstand mit der Formel d [cm] = T [µs]/58 berechnen.

Für die generelle Ausrichtung im Raum besitzt der Roboter einen digitalen Kompass, welcher dazu dient, eine Zielrichtung zu definieren und diese im autonomen Fahrprozess auch einzuhalten. Dieser digitale Kompass wird über die UART Schnittstelle ausgelesen.

Der Roboter soll sich im Raum bewegen, dafür besitzt er zwei Räder, welche mit elektrischen Motoren verbunden sind. Diese werden per Pulsweitenmodulation angesteuert.

Für Debugging oder die Möglichkeit während der Laufzeit relevante Informationen anzuzeigen, besitzt der MC Mobile ein LCD-Display, welches mit dem External Memory Controller angesteuert wird.

Ziel des Roboters ist es ein Labyrinth autonom zu durchfahren und dabei keine Hindernisse zu berühren.

## Interakteure

Die Akteure des Systems sind die Umwelt bzw. der Parkour den der Roboter durchfahren muss, Sowie der Bediener welcher den Roboter starten und ausrichtet.

Die Interaktionen mit den Akteuren werden in folgendem UML Anwendungsfalldiagramm dargestellt.

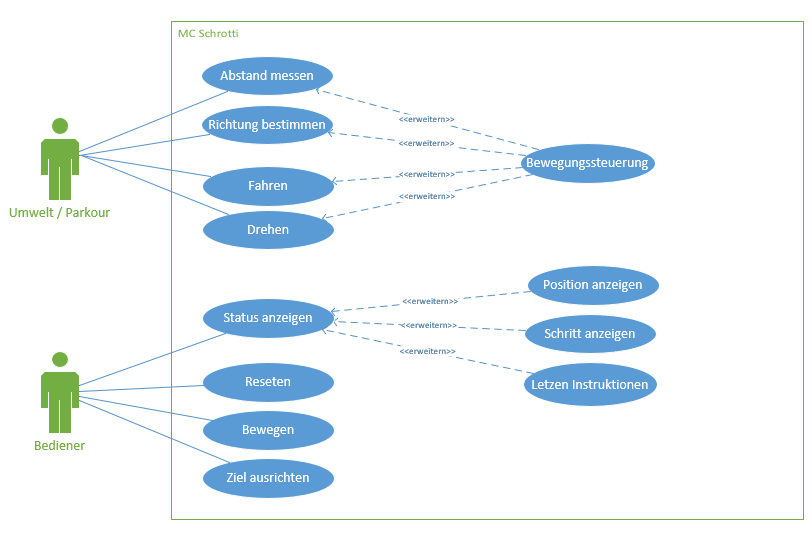


Abbildung : Anwendungsfalldiagramm

## Softwaremodule

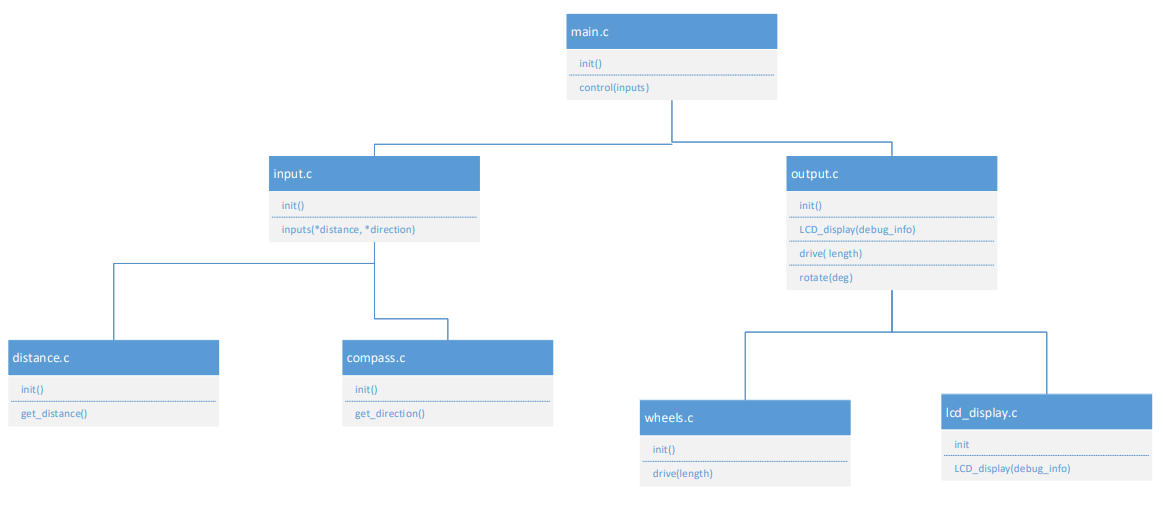


Abbildung : Softwaremodule

Im obigen Diagramm werden die Module aufgezeigt, die für die Robotersteuerung verwendet werden. Als Hauptmodule wird die *main.c* verwendet. Jedes Modul besitzt eine init Methode, welche beim Start des Systems von *main.c*-init herab kaskadierend aufgerufen werden. Die Hardwareanbindung findet an den untersten Modulen statt, die *input.c* und *output.c* sind eine Zusammenfassung aller Input, sowie Output Module. Diese dienen dazu, die Hauptsteuerung in der *main.c* übersichtlich zu halten.

## Entwicklungsschritte

Da die Entwicklung einer Steuerungssoftware für einen autonom fahrenden Roboter eine komplexe Aufgabenstellung mit zeitlicher Begrenzung ist, bietet es sich an die zu programmierende Steuerung in Entwicklungs- und Zeitabschnitte zu unterteilen.

Um im Fahrprozess einen Semantischen Fehler erkennen zu können, wird im ersten Entwicklungsschritt eine Status LED programmiert. Da die LED mit einem Interrupt an- bzw. ausgeschalten wird, lässt sich schnell erkennen, ob die Software auf eine unbehandelte Exception trifft und das Programm sich aufhängt.

Der zweite Entwicklungsschritt ist ebenfalls für den Debugging Prozess vorgesehen. Das auf dem Roboter installierte LCD-Display wird über den External Memory Controller angesteuert. Das Display soll aktuelle Messdaten, wie Entfernungen, Lage und Ausrichtung anzeigen um für die Programmierer während der Entwicklung der Fahrstrategie ausreichend Debugging- Informationen bereitzustellen. Zusätzlich sollen die aktuelle Schrittkette bzw. der Programmstatus angezeigt werden, um im Falle eines Fehlers die Entscheidungen des Roboters nachvollziehen zu können.

Im dritten Entwicklungsschritt werden die Ultraschallsensoren an das System angebunden. Die Ultraschallsensoren sollen verwendet werden, um die Abstände und Strecken ermitteln zu können. Durch die radial versetzten Sensoren an der Roboterfront lässt sich ebenfalls ein Kollisionsschutz implementieren, die Grundlagen für eine solche Funktion soll in diesem Entwicklungsschritt programmiert werden.

Als letzten Sensor wird noch der digitale Kompass integriert. Dieser wird über die UART Schnittstelle des Microcontrollers angebunden. Der Kompass soll in unserem Robot als Richtungsvorgabe dienen, er wird beim Start ausgerichtet und orientiert sich im Fahrprozess an der Himmelsrichtung.

Damit sich der Roboter im Raum bewegen kann, werden im fünften Entwicklungsschritt die PWM gesteuerten Motoren in Betrieb genommen. Es sollen außerdem zwei Fahrfunktionen für den Roboter programmiert werden. Die erste Funktion lässt den Roboter geradeausfahren, als Übergabewerte wird hierbei eine Strecke vorgegeben. Die zweite Bewegungsfunktion soll eine Rotation realisieren. Hier wird als Übergabeparameter eine Gradanzahl an den Roboter übergeben.

Um die Bewegungsfunktionen und deren Ausführung zu validieren, sollen ein Hodometer für die Bewegungsstrecke und ein Radencoder für die Drehungen integriert werden.

Im letzten Schritt der Softwareimplementierung soll die Fahrstrategie und die Zusammenfassung von Inputs und Outputs nach dem EVA Prinzip ausgearbeitet werden.

# Zusammenfassung

Die Systemarchitektur des zu entwickelnden Roboters orientiert sich stark nach dem EVA Prinzip. Ähnlich ist auch die Softwarearchitektur aufgebaut. Sobald alle Implementierungsschritte durchlaufen, beginnt die Testphase für die einzelnen Komponenten. Als Voraussetzung für den Schritt in die nächste Phase der Entwicklung, die Implementierung der Fahrstrategie, ist das alle Hardware und Softwarekomponenten unabhängig voneinander einwandfrei funktionieren. Die Fahrstrategie wird im Module *main.c* realisiert.