



C-3EPO SUCHT SEIN ZU HAUSE

Software-Entwicklungspraktikum (SEP)
Sommersemester 2015

Angebot

Auftraggeber
Technische Universität Braunschweig
Chair for Chip Design for Embedded Computing
Prof. Dr.-Ing. Mladen Berekovic
Mühlenpfordtstr. 23
38106 Braunschweig

Betreuer: Bastian Farkas

Auftragnehmer:

Name	E-Mail-Adresse
Christian Walther	christian.walther@tu-braunschweig.de
Jevgeny Nabotnikow	j.nabotnikow@tu-braunschweig.de
Sebastian Fiedler	sebastian.fiedler@tu-braunschweig.de
Tim Rohkohl	t.rohkohl@tu-braunschweig.de
Tim Wesemeyer	t.wesemeyer@tu-braunschweig.de
Van Tuan Tran	v.tran@tu-braunschweig.de

Braunschweig, 23. April 2015

Inhaltsverzeichnis

1	Einleitung	4
1.1	Ziel	4
1.2	Motivation	4
2	Formale Grundlagen	5
3	Projektablauf	6
3.1	Meilensteine	6
3.2	Geplanter Ablauf	6
4	Projektumfang	8
4.1	Lieferumfang	8
4.2	Kostenplan	8
4.3	Funktionaler Umfang	8
5	Entwicklungsrichtlinien	9
5.1	Konfigurationsmanagement	9
5.2	Design- und Programmierrichtlinien	9
5.3	Verwendete Software	9
6	Projektorganisation	10
6.1	Teameinteilung	10
6.2	Schnittstelle zum Auftraggeber	10
6.3	Schnittstelle zu anderen Projekten	10
6.4	Interne Kommunikation	10
7	Glossar	11

Abbildungsverzeichnis

3.1	Meilensteine des Projektes C-3EPO sucht sein Zuhause als Gantt-Diagramm . .	7
-----	---	---

1 Einleitung

1.1 Ziel

Das Ziel des Projektes besteht darin einen Saugroboter, bei niedrigem Akkustand oder auf expliziten Befehl hin, zur Ladestation fahren zu lassen. Dabei soll dieser mit Hilfe von Bluetooth Beacons navigieren. Zuletzt wird mit Hilfe des Laserscanners das Docking durchgeführt. Die Lösungen werden in die bereits bestehende ROS Infrastruktur integriert.

1.2 Motivation

Oftmals kommt es vor, dass der Roboter C-3EPO seine Orientierung verliert, wenn er nach dem Festfahren neu ausgerichtet oder anderweitig während des Betriebes in seiner Bewegung gestört wurde. Das ist vor allem fatal, wenn sein Akkustand kritisch wird und genau deshalb soll dafür eine Lösung gefunden werden.

2 Formale Grundlagen

Als Programmiersprache dient Python. Entwickelt wird unter einer embedded-Ubuntu Umgebung und mit Hilfe des ROS-Frameworks in der Version Indigo. Die Anwendung wird in Englisch ausgeliefert.

3 Projektablauf

Die Dokumente, im aktuellen Entwicklungszustand, sind jeweils Mittwochs vor dem Abgabetermin im SVN hochzuladen, damit der Betreuer diese sichten kann.

3.1 Meilensteine

Nummer	Meilenstein	Dokumente	Abgabetermin
1	Projektstart	-	13.04.15
2	Angebot	Angebot	24.04.15
3	Abnahmespezifikation Pflichtenheft	Abnahmespezifikation Pflichtenheft	13.05.2015
4	Zwischenpräsentation Modul Bluetooth-Signal Modul Status	-	15.05.2015
5	Fachentwurf	Fachentwurf	03.06.2015
6	Modul Bluetooth-Triangulation Modul Docking	-	10.06.2015
7	Technischer Entwurf	Technischer Entwurf	01.07.2015
8	Modul Signalverlust Modul GUI	-	08.07.2015
9	Testdokumente Systemintegration	Testdokumente	15.07.2015
10	Tag der jungen Softwareentwickler	-	23.07.2015

3.2 Geplanter Ablauf

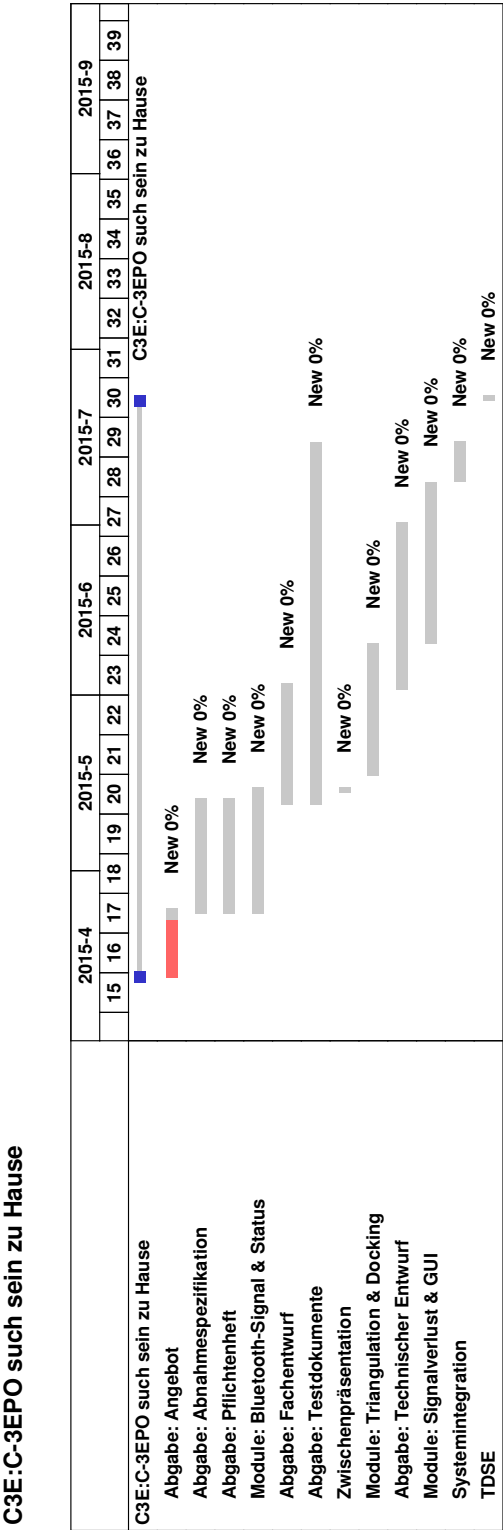


Abbildung 3.1: Meilensteine des Projektes C-3EPO sucht sein Zuhause als Gantt-Diagramm

4 Projektumfang

4.1 Lieferumfang

Am Ende des Projektes werden der Quellcode, die Anwendung und die vollständige, geforderte Dokumentation mitgeliefert.

4.2 Kostenplan

Der Lohn für einen Entwickler beträgt 100 Euro pro Stunde. Bei einem geschätzten Aufwand von insgesamt 130 Arbeitsstunden und anfallenden Materialkosten ergibt sich eine Gesamtkostenhöhe von 150.000 Euro. Diese Arbeitskosten verteilen sich auf die Analyse, die Implementierung, das Testen und die Dokumentation des Projektes.

4.3 Funktionaler Umfang

Der Roboter erkennt selbstständig, dass er zur Ladestation zurückkehren muss. Dann fährt er eigenständig oder auf Wunsch zurück zur Ladestation.

5 Entwicklungsrichtlinien

5.1 Konfigurationsmanagement

Benutzt wird das zentrale Redmine-SVN vom ISF. Alle Dokumente sollen jeweils am Mittwoch vor der bevorstehenden Abgabefrist im Repository hochgeladen werden. Dabei ist der Entwicklungsstand möglichst fortgeschritten. Beim Einstellen ins SVN sind immer Kommentare abzugeben, die sich auf die Änderungen im Dokument beziehen.

5.2 Design- und Programmierrichtlinien

Bei der Programmierung werden die allgemeinen und anerkannten Python-Sprachstandards beachtet.

5.3 Verwendete Software

Für die Dokumentation werden MikTeX und Texmaker verwendet. Die Entwicklungsumgebung ist Ubuntu und eine embedded-Version dieser. Als Framework kommt ROS Indigo zum Einsatz. yEd dient dem Projektteam als Diagrammerstellungssoftware. Inkscape dient der Grafikformatierung und als generelle grafische Unterstützung.

6 Projektorganisation

6.1 Teameinteilung

Das Softwareteam arbeitet in zwei unabhängigen Teams à drei Personen. Jedes Team implementiert insgesamt drei Module. Das erste Team besteht aus Christian Walther, Tim Rokohl und Van Tuan Tran und implementiert die Module Status, GUI und Docking. Team zwei bestehend aus Jevgeny Nabotnikow, Sebastian Fiedler und Tim Wesemeyer programmiert alle Module mit Bluetooth-Abhängigkeiten. Die letztliche Systemintegration wird wieder vom gesamtem Softwareteam durchgeführt.

6.2 Schnittstelle zum Auftraggeber

Die Kontaktaufnahme mit dem Auftraggeber erfolgt mittels Telefon, E-Mail oder im persönlichen Gespräch.

6.3 Schnittstelle zu anderen Projekten

Verwendet werden schon bestehende Schnittstellen, die der Navigation des Roboters dienen.

6.4 Interne Kommunikation

Das Softwareteam kommuniziert über E-Mail, Messengerdienste, Gruppentreffen und Telefonkonferenzen.

7 Glossar

Begriff	Bedeutung
ROS	ROS (engl. Robot Operating System) ist ein flexibles Software-Framework für Roboter. ROS beinhaltet Werkzeuge, Bibliotheken und Konventionen, die darauf abzielen das Erstellen komplexen Verhaltens für eine Vielzahl von Robotern zu erleichtern.