

Fakultät für Elektrotechnik und Informatik

# ROS auf dem Raspberry Pi

14 November, 2013

Systemadministration Projekt in Angewandter Informatik  
von Denis Herdt und Almin Causevic

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Motivation . . . . .	3
1.2	Zielsetzung . . . . .	3
1.3	Eigene Leistung . . . . .	4
1.4	Aufbau der Arbeit . . . . .	4
<b>2</b>	<b>Grundlagen</b>	<b>4</b>
2.1	Raspberry Pi Hardwarekomponenten . . . . .	4
2.2	Smartphone . . . . .	6
2.3	Volksbot . . . . .	6
2.4	ROS . . . . .	6
<b>3</b>	<b>Problem</b>	<b>6</b>
3.1	Verwandte Arbeiten . . . . .	6
<b>4</b>	<b>Anforderungen</b>	<b>6</b>
<b>5</b>	<b>Lösungsvorschläge</b>	<b>7</b>
<b>6</b>	<b>Bewertung der Lösungen</b>	<b>7</b>
<b>7</b>	<b>Implementation</b>	<b>8</b>
<b>8</b>	<b>Fazit</b>	<b>8</b>
8.1	Zusammenfassung . . . . .	8
<b>9</b>	<b>Ausblick</b>	<b>8</b>
<b>10</b>	<b>Eigene Leistung</b>	<b>8</b>
<b>11</b>	<b>Quellen</b>	<b>8</b>

# 1 Einleitung

1-2 Seiten

## 1.1 Motivation

Wir haben uns für dieses Thema entschieden, weil wir mit Hard- und Software arbeiten wollten und mitbekommen haben, wie umständlich z.B. der Roboter "Volksbot" im Robotiklabor der Hochschule mit einem zusätzlichen Laptop gesteuert werden muss. Deshalb haben wir uns entschlossen, dieses Problem mithilfe des Raspberry Pi zu lösen. Die Idee stieß auch auf großes Interesse bei den Labormitarbeitern und bringt viel praktischen Nutzen. Unser Ziel möchten wir mithilfe des Betriebssystems ROS (Robot Operating System) realisieren. Für ROS haben wir uns entschieden, weil es zur Zeit die aktuellste und inovativste Methode ist, ein flexibles Netzwerk von Komponenten für z.B. die Robotersteuerung zu realisieren. Es wird weltweit eingesetzt und könnte auch in Zukunft sehr nützlich für die Forschung werden. Durch die Kombination mit der Mächtigkeit und Flexibilität von ROS und der geringen Größe und dem niedrigen Energieverbrauch des Pi eröffnen sich außerdem neue und effizientere Einsatzmöglichkeiten.

Sehr gut finden wir auch die Tatsache, dass wir mit Linux arbeiten können und dieses Projekt viel Konfiguration des Betriebssystems voraussetzt, da wir auch gerne mehr Praxisanwendung mit Linux haben möchten.

## 1.2 Zielsetzung

Recherche über Linux Systeme auf Raspberry Pi  
Passendes Linux System auf Raspberry Pi aufsetzen  
Recherche über ROS geeigneten Roboter wählen (Recherche Verfügbarkeit, Komponenten etc.)  
eventuell Komponenten beschaffen (Wlan Stick, etc.)  
ROS auf Raspberry Pi aufsetzen  
Wlan-Netzwerk zwischen 2 ROS-kompatiblen Systemen herstellen  
Netzwerk mithilfe von Testausgaben überprüfen

**optional** Roboter durch einen Raspberry Pi mithilfe von ROS steuern  
Bild einer Webcam durch ROS auf einen Bildschirm streamen

### 1.3 Eigene Leistung

anstatt Linux PC wird Pi benutzt

### 1.4 Aufbau der Arbeit

Recherche über Linux Systeme auf Raspberry Pi  
Passendes Linux System auf Raspberry Pi aufsetzen  
Recherche über ROS geeigneten Roboter wählen (Recherche Verfügbarkeit, Komponenten etc.)  
eventuell Komponenten beschaffen (Wlan Stick, etc.)  
ROS auf Raspberry Pi aufsetzen  
Wlan-Netzwerk zwischen 2 ROS-kompatiblen Systemen herstellen  
Netzwerk mithilfe von Testaufgaben überprüfen

**optional** Roboter durch einen Raspberry Pi mithilfe von ROS steuern  
Bild einer Webcam durch ROS auf einen Bildschirm streamen

## 2 Grundlagen

### 2.1 Raspberry Pi Hardwarekomponenten

Der Raspberry Pi ist ein kreditkartengroßer Einplatinencomputer, der von der Raspberry Pi Foundation entwickelt wurde. Wir benutzen für dieses Projekt das leistungsstärkere Modell B.

Technische Details:

- Preis: ca. 35 Euro
- Prozessor: ARM1176JZF-S (700 MHz)
- Broadcom VideoCore IV
- SDRAM: 512 MB
- Bis zu 16 GPIO-Pins
- USB-Anschlüsse: 2

- FBAS, HDMI
- 3,5-mm-Klinkenstecker (analog), HDMI (digital)
- Kartenleser für SD (SDHC und SDXC)/MMC/SDIO
- 10/100-MBit-Ethernet-Controller
- 5 V, 700 mA (3,5 Watt)
- 5-V-Micro-USB-Anschluss (Micro-B), alternativ 4 x AA-Batterien

Für den Anschluss an einen Bildschirm benutzen den HDMI Ausgang. Diesen benutzen wir für Kontrollausgaben und Tests.

Als Stromquelle steht ein Akkupack .....(Firma,Modell) zum Einsatz. Hierbei ist wichtig, dass der Raspberry mind. 700 mA, besser 1 A zur Stromversorgung bekommt. Bei niedrigerer Amperzahl arbeitet der PC oft nicht zuverlässig.

Ein schneller und großer RAM Speicher ist für unsere Zwecke wichtig, da viele Signale, teils sogar synchron, verarbeitet werden. Ein 17-Zoll Monitor wird für die Bildschirmausgabe des Raspberry über HDMI zu DVI benutzt werden müssen.

Für die Datenübertragung über das Netz benutzen wir einen Standard 300Mbit/s WLAN-Stick.

Der Raspberry Pi arbeitet mit einer ARM-Prozessorarchitektur. Diese Architektur wird gerne für embedded Systems, wie PDAs oder Router, eingesetzt.

Sie ist auch auf jedem Smartphone zu finden, da sie den Vorteil einer sehr geringen Leistungsaufnahme bietet.

Erwähnenswert ist die Architektur deshalb, weil wir im Laufe des Projektes Schwierigkeiten hatten, auf die wir später eingehen werden. Wir benutzen das auf Linux basierende Betriebssystem Raspbian. Dabei handelt es sich um ein für Raspberry Pi optimiertes open-source Debian-System. Es enthält viele für die ARM-Architektur vorkompilierte Pakete (über 35.000), dazu auch Features wie etwa eine GUI.

Das System braucht 3GB Speicherbedarf unserer 16GB großen SD-Karte. Der zusätzliche Speicherplatz wäre nötig, falls man vorhat, Log-Dateien

und Ähnliches direkt auf dem Raspberry Pi zu speichern. In unserem Fall übernehmen das jedoch die leistungsstärkeren PC's über das ROS Netzwerk.

GPIO (General Purpose Input/Output) ist ein weiteres interessantes Feature des Raspberry Pi. Es gibt uns die Möglichkeit, jegliche Hardware-Funktionalität anzusteuern. Beispielsweise können LED-Leuchten oder der Start-Knopf der Kaffeemaschine damit über ROS gesteuert werden. Wir konnten uns leider jedoch zeitlich bedingt nicht mehr mit dieser Thematik beschäftigen, es bietet aber Anreiz für noch mehr Ansätze und Umsetzungen mithilfe des Raspberry Pi und ROS.

## **2.2 Smartphone**

## **2.3 Volksbot**

## **2.4 ROS**

# **3 Problem**

Hauptproblem??

## **3.1 Verwandte Arbeiten**

Steffen, Marc ROS-Tut

# **4 Anforderungen**

Datenpakete übers ROS Netzwerk verschicken kann Beliebige Datentypen Beliebige Linux-Systeme Überprüfung davon Log-Dateien, Kontrollausgaben etc.. robuste und zuverlässige Kommunikation möglich modulares und flexibles Netzwerk Pakete(repos) sollen leicht in Projekte eingebunden werden

optional: Roboter soll bei Steuerung in richtige Richtungen fahren Roboter soll sofort auf Richtungsanweisungen reagieren Robotersteuerung Sensibilität sollte einstellbar sein Steuerung über Wlan Raspberry Pi auf Roboter ohne

Kabelzgewirr Node des Raspberry starte automatisch mit Raspbian Leichte Bedienbarkeit des Roboters

optional: Streaming-Bild ruckelfrei bearbeitet werden soll unkonvertiert in Echtzeit übertragen werden soll an externem Bildschirm ausgegeben werden

## 5 Lösungsvorschläge

Netzwerk wird mit ROS-Master aufgesetzt, über topics und nodes verschicken und empfangen von Paketen

beliebige Datentypen durch msg und srv Funktion realisierbar

ROS unterstützt beliebige Linux Systeme..

Horchen an Topics mit echo + Ausgabe im Code mithilfe ROS-Stream Fkt.

Begrenzung der Anzahl der Datenübertragung Verschickte und erwartete Datentypen müssen stimmen Richtige Konfiguration des Masters und Clients durch ROS und catkin gegeben

durch catkin gegeben

opt: abhängig vom richtigen Code der Motorsteuerung

abhängig von Raspberry Pi Hardware und effizientem Code und guter Hardware

abhängig vom Code

Hardware nötig Wlan-Stick, Akku-Pack

sollte in Linux beim booten konfiguriert werden

gegeben durch gute Smartphone App

opt: abhängig von Raspberry Pi Hardware und Übertragungs-Codierung

abhängig vom Codex Hardware Komponenten

nötige Hardware und Netzwerk legen

## 6 Bewertung der Lösungen

alles gut

## 7 Implementation

ROS Netzwerk (topics, nodes, catkin...) Linux SD Karte Pi Datenübertragung an Volksbot Konfiguration Motorsteuerung-Paket und Compilieren Kontrollausgaben Netzwerkkomponenten zusammen arbeiten lassen (Hard/-Software) Wlan -j Teilproblem Kamera -j Teilproblem Zeit

## 8 Fazit

1 Seite

### 8.1 Zusammenfassung

sehr knapper Zeitrahmen hohe Anforderungen für diesen Zeitraum interessantes Projekt anspruchsvolles Projekt viel über Linux und Ros und Hardware und Netzwerken gelernt Spaß

## 9 Ausblick

Benutzung des Pi's als Hardware Nodes Lab Benutzung des Pi's für kleine Roboterprojekte Lab Immer mehr binary packages für ROS Pi verfügbar -j mehr Möglichkeiten z.B.: GPIO Interface -j Knöpfe Kaffemaschine, Relays, alle möglichen Signale auf Hardwareebene

## 10 Eigene Leistung

anstatt Linux PC wird Pi benutzt

## 11 Quellen

[http://de.wikipedia.org/wiki/Raspberry\\_Pi](http://de.wikipedia.org/wiki/Raspberry_Pi)



<http://www.amazon.de/Raspberry-Pi-RBCA000-Mainboard-1176JZF-S/dp/B008PT4GGC>  
<http://www.raspbian.org/>  
<http://www.softwareok.de/?seite=faq-System-Allgemein&faq=13>  
<http://de.wikipedia.org/wiki/ARM-Architektur>  
[http://www.rn-wissen.de/index.php/Raspberry\\_PI:\\_GPIO](http://www.rn-wissen.de/index.php/Raspberry_PI:_GPIO)