

# Mobile Roboter SS12

## Gruppe x

Kevin Walter, Gerhard Klostermeier, Andreas Jansche

© 12. Juni 2012

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Inhalte und Ziele . . . . .	3
1.2	Der ct-Bot . . . . .	3
<b>2</b>	<b>Installation und Inbetriebnahme</b>	<b>4</b>
2.1	Installation . . . . .	4
2.1.1	System vorbereiten . . . . .	4
2.1.2	Quellcode holen . . . . .	4
2.1.3	Eclipse einrichten . . . . .	4
2.2	Inbetriebnahme . . . . .	5
2.2.1	AVR ISP mkII und avrdude . . . . .	5

# 1 Einleitung

## 1.1 Inhalte und Ziele

Im Rahmen der Vorlesung “Mobile Roboter“ der Hochschule Aalen wurden die Aufgaben aus dem Vorlesungsskript bearbeitet. Genutzt wurde Eclipse zum Entwickeln der Programme sowie avrdude zum Flashen des Microcontrollers des ct-Bot.

## 1.2 Der ct-Bot

...

## 2 Installation und Inbetriebnahme

### 2.1 Installation

Für die Entwicklung wurde ein aktuelles Ubuntu Linux verwendet. In diesem Abschnitt ist beschrieben was für vorbereitende Schritte auf dem System durchgeführt werden müssen um entwickeln und den ct-Bot flashen zu können.

#### 2.1.1 System vorbereiten

Zunächst müssen einige Softwarepakete nachinstalliert werden:

- `eclipse-cdt`  
Die Eclipse-Variante zur C/C++-Entwicklung. (Der ct-Bot wird in C programmiert.)
- `binutils-avr`, `gcc-avr`, `avr-libc`  
Werden zum Kompilieren für den Microcontrollers des ct-Bot benötigt. (Cross-Compiler)
- `avrdude`  
Wird zum Flashen des Microcontrollers des ct-Bot benötigt.
- `subversion`  
Wird zum Holen des aktuellen Quellcodes für den ct-Bot aus dem heise-Repository benötigt.

Der konkrete Befehl um die Pakete unter Ubuntu zu installieren sieht folgendermaßen aus:

```
sudo apt-get install eclipse-cdt binutils-avr gcc-avr \
avr-libc ^avrdude subversion
```

#### 2.1.2 Quellcode holen

Für den Quellcode erstellen wir zunächst ein Verzeichnis *ctbot* und wechseln hinein:

```
mkdir ctbot && cd ctbot
```

Nun holen wir uns den aktuellen stable-Code vom heise-Repository:

```
svn checkout https://www.heise.de:444/svn/ctbot/stable
```

Der aktuelle Quellcode des ct-Bot befindet sich nun also unter `~/ctbot/stable` und muss im nächsten Schritt nur noch in Eclipse eingebunden werden.

### 2.1.3 Eclipse einrichten

TODO:

#source im eclipse einbinden:

File → Import

General → Existing Projects into Workspace

Browse...

~/ctbot/stable/ct-Bot

#evtl nochmal mit ct-Sim wenn gewollt

ct-Bot Projekt wählen #evtl. selbes mit ct-Sim

Project → Properties

C/C++-Build

Manage Configurations...

Debug-MCU-m32

Set active

Ok

Apply

Ok

#SPEED\_CONTROL\_AVAILABLE aktivieren für Kompilieren ohne Warnungen  
in der ct-Bot.h den Kommentar vor

#define SPEED\_CONTROL\_AVAILABLE

entfernen...

## 2.2 Inbetriebnahme

### 2.2.1 AVR ISP mkII und avrdude

TODO:

#in betrieb nehmen

Programmer einstecken

lsusb zeigt "Atmel Corp. AVR ISP mkII"

#flashen mit avrdude

avrdude

-c avrispmkII → legt den Programmer fest

-P usb → usb als connection port

-p m32 → legt das AVR device fest: m32

-U flash:w:"<file>":i → flash gibt an dass der Flash-Speicher verwendet  
→ w gibt an dass geschrieben werden soll (file -  
→ "<file>" ist der Pfad zur Datei die ins Flash  
geschrieben werden soll (hier ct-Bot.hex)

→ i gibt das Format der Datei an, hier i f r I  
sudo !!!

konkret:

```
sudo avrdude -c avrispmkII -P usb -p m32 -U flash:w:"~/ctbot/stable/ct-Bo
```