

1 Komponenten

2 Aufsetzen des Raspberry Pis

Das Betriebssystem des Raspberry Pis benötigt etwas Zusatzsoftware, um über das I2C-Interface mit dem Adafruit 16 Channel Servo Driver kommunizieren zu können. Im folgenden wird von Raspbian als Betriebssystem ausgegangen.

I2C Software Zuerst wird die Software zum Ansteuern des PWM-Moduls installiert:

```
sudo apt-get update
sudo apt-get install python-smbus
sudo apt-get install i2c-tools
```

Dann muss `/etc/modprobe.d/raspi-blacklist.conf` überprüft werden. Falls die Datei die Zeile `blacklist i2c-bcm2708` enthält, mit `#` auskommentieren. Nun noch in `/etc/modules` die Zeilen

```
i2c-dev
i2c-bcm2708
```

hinzufügen und den Raspberry Pi neustarten.

Jetzt wird der Adafruit 16 Channel Servo Driver an die GPIO-Pins des Raspberry Pis angeschlossen. Mithilfe von `sudo i2cdetect -y 0` bzw. `sudo i2cdetect -y 1` (je nach Baureihe des Raspberry Pis) kann die Verbindung zwischen dem Servo Driver und dem Raspberry Pi überprüft werden.

Sturo Nun richten wir das Programm zur Steuerung des RC-Autos ein.

Die aktuelle Version des Programmes befindet sich momentan auf Github:

```
wget https://github.com/linucc/sturo
cd sturo
chmod +x sturo_start.sh
```

Das Programm Sturo benötigt Python 2.7 und muss für Funktionen wie Sensoren privilegiert ausgeführt werden. Nun soll der Raspberry Pi Sturo direkt nach dem Bootvorgang ausführen.

Dazu editiert man die Datei `/etc/inittab`, sodass der Login beim Booten automatisch ausgeführt wird:

Die Zeile `1:2345:respawn:/sbin/getty 115200 tty1` wird mit `#` auskommentiert. Dann kann man direkt darunter in einer neuen Zeile diesen Code einfügen:

```
1:2345:respawn:/bin/login -f pi tty1 </dev/tty1 >/dev/tty1 2>&1
```

Beim Login ist man nun als Benutzer Pi eingeloggt.

Jetzt wird die Skriptdatei aufgerufen, indem am Ende der Datei `/etc/profile` die Zeile

```
. /home/pi/sturo/sturo_start.sh
```

eingefügt wird.

Damit Sturo erweiterte Funktionen benutzen kann, muss es privilegiert mithilfe von `sudo` ausgeführt werden. Hier wird momentan noch nach einem Passwort gefragt, so dass sich das Programm nicht automatisch startet.

Um dies zu umgehen, können wir das `setuid` bit auf den Nutzer `root` setzen oder die `/etc/sudoers` Datei mithilfe von `visudo` editieren und folgende Zeile an das Ende fügen:

```
pi ALL=NOPASSWD: /usr/bin/python
```

Nun wird beim Ausführen von Python mit `sudo` nicht mehr nach dem Passwort gefragt.

Das führt in Kombination mit dem Auto-Login zu einem sehr unsicheren System, was für dieses Projekt aber erstmal unwichtig ist.