

## PAT – getting started

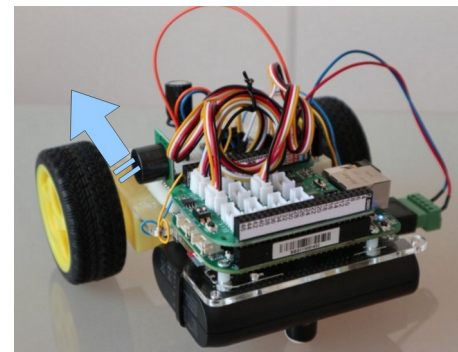
Dieses Dokument ist für einen schnellen Einstieg ins Praktikum gedacht. Im Skript finden sich viele Hilfestellungen und (fast) alle nötigen Informationen. Informationen über z. B. Sensoren müssen selbstständig beschafft werden. Auf eine genaue Aufgabenbeschreibung und die Vorgabe einer Reihenfolge wird bewusst verzichtet.

**Aufgabenstellung:** Lasse den RoboToGo autonom und in einem definierten Abstand mindestens eine Runde um die Sensorlabor-Innenwand fahren.

**Dokumentation:** Video-Dokumentation über Arbeitsweisen, einzelne Arbeitsschritte und Projekt-Ergebnisse.

### RoboToGo (Skript Kapitel 2)

- Differenzsteuerung mit zwei DC Getriebemotoren
- Ansteuerung Motoren über H-Brücke
- Fahrrichtung: Räder vorne (sonst blockiert BallCaster häufig)
- Stromversorgung über 5V Powerbank
- Träger für Sensoren und Beaglebone



### Beaglebone Green (Skript Kapitel 3 + 7)

- Linux Embedded System (LES) mit Ubuntu 20.04 LTS  
[wiki.seeedstudio.com/BeagleBone\\_Green/](http://wiki.seeedstudio.com/BeagleBone_Green/)
- Texas Instruments Sitara 1 GHz ARM-A8 Cortex, 512 MB RAM
- 65 GPIO (**3,3V**) / 8 PWM Ausgänge / 7 ADC (**1,8V**) / 2 I<sup>2</sup>C / 2 SPI / 2 UART / 2 CAN
- USB Host (USB-A), USB Client (µUSB Stromversorgung ohne Motoren), Ethernet (RJ45)
- WLAN über USB-Dongle (Edimax EW-7811Un)
- Grove Base Cape zum Schutz der GPIOs und Pegelwandlung 3,3 / 5 V Pegel  
[wiki.seeedstudio.com/Grove\\_Base\\_Cape\\_for\\_BeagleBone\\_v2/](http://wiki.seeedstudio.com/Grove_Base_Cape_for_BeagleBone_v2/)

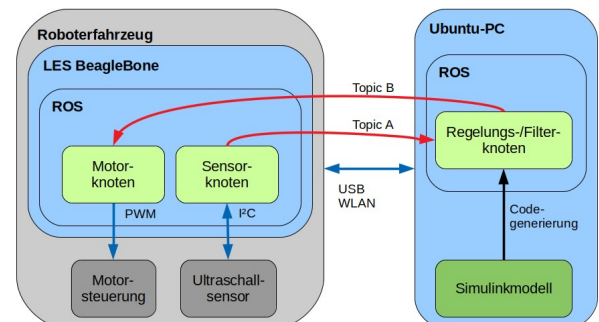
**ACHTUNG:** Ausschließlich Grove-Kabel und weiße Buchsen auf dem Cape verwenden!

### Sensoren (I<sup>2</sup>C)

- Ultraschall Abstandssensor SRF02 (nachgeschalteter Kalman-Filter in Simulinkknoten)
- TOF-Sensor VL53L1X (Messwertfusion innerhalb Sensorknoten)

### ROS (Skript Kapitel 4)

- ROS1 Noetic installiert auf SSDToGo
- Middleware zur Steuerung von Robotic-Systemen mit dedizierten Software-Knoten für jede Systemkomponente
- Knoten im PAT erstellbar mit Simulink, Python und C++



### MATLAB 2019b / Simulink 10

- Software zur Erstellung der Regelung und des Kalman-Filters (Simulink). Knoten dafür auf PC.

### SSDToGo

- Ubuntu 20.04 LTS inkl. aller nötigen Software zum booten am eigenen Laptop (**ACHTUNG:** Funktioniert nicht mit Apple M1!). Aktivierung eigenes Mathworks-Account nötig.

## Video-Dokumentation

- Hochformat, Laufzeit 2 Minuten
- Dokumentation Projektkinhalt, Arbeitsschritte, Arbeitsweisen, Projektergebnis und Einordnung ins Mechatronik-Studium.

## Einrichten und Verwenden der SSDToGo (Skript Kapitel 12.18)

### Über UEFI-Bootmenü

- Booten über UEFI-Bootmenü (Anzeigename: "StoreJetTranscend") durch Drücken der Funktionstaste beim Starten  
Dell/Lenovo → F12 / HP → F9 / Mac → ALT  
Surface → meist nur über WIN10 (siehe unten)
- Secure-Boot muss häufig deaktiviert werden
- Fast-Boot muss häufig deaktiviert werden

### Aus WIN10 (z. B. bei Surface-Geräten)

- Optionen für den erweiterten Start
- „Erweiterter Start“ → „Jetzt neu starten“ -> „Ein Gerät verwenden.“
- „EFI USB Device“ auswählen

**Mac mit M1 Prozessor** → Keine Kompatibilität

## Verbindungen aufbauen (Skript Kapitel 5)

### BeagleBone mit PC via USB verbinden (Skript Kapitel 5.1)

- BB per USB (µUSB-Buchse !!fragil!!) einstecken
- Terminal öffnen und per SSH auf 192.168.6.2 verbinden (SSH → CheatSheet)  
User: beagle | PW: tempwd

### BeagleBone mit PC via WLAN verbinden (Skript Kapitel 5.3)

- WLAN Netzwerk auswählen (Sensor1...Sensor4 | PW: 0123456789 )
- PC im gewählten WLAN anmelden
- BB per USB verbinden und per ssh anmelden
- WLAN des BB nach Anleitung konfigurieren  
(Skript: Mit dem BB in ein vorhandenes WLAN einwählen)

### Verwendung des Hostnames für Verbindungen

- Hostname in Dateien hostname und hosts ändern (alle BB haben ident. Hostname)  
→ sudo nano /etc/hostname und sudo nano /etc/hosts
- BB neu starten

### Bekannte Probleme bei WLAN: (Skript Kapitel 5.4)

- Wird die Lease-Time des Router überschritten, kann es sein, dass die Geräte neue IP-Adressen zugewiesen bekommen. Die neue Adresse muss dann herausgefunden werden (beim BB via USB-Verbindung) und ggf. in der ROS-Config (~/.bashrc) geändert werden.
- Wird der Hostname geändert kann es bei der Auflösung durch den Router (Name/IP) zu Problemen kommen. Ein (mit den anderen abgestimmter) Neustart hilft manchmal
- Bei Spannungsabfall fällt der WLAN-Dongle aus.  
Ein Neustart des Netzwerkmanagers kann helfen

### Dateiübertragung zwischen PC und BB (Skript Kapitel 6.1.4)

- Im Dateimanager „Andere Orte“ auswählen
- die SFTP-Verbindung eintagen und „Mit Server verbinden“
- Passwort (tempwd) eingeben

## Testen der Sensoren und Antriebe (Skript Kapitel 8 + 9)

Auf BB und [github.com/StefanMack/PAT-ROS](https://github.com/StefanMack/PAT-ROS) Beispielprogramme für Sensoren und Antriebe.

## Einarbeiten in ROS (Skript Kapitel 10)

ROS1 Noetic inkl. Beispielknoten ist auf der SSDToGo bereits installiert.

### ROS einrichten

1. Herausfinden der Adressen/Domains (Aktueller Rechner & Rechner mit ROS-Master)
2. Ändern der Adressen in `~/.bashrc` (siehe unten)
3. Neu laden der Konfiguration über `source`-Befehl oder Neustart des Terminals

Umgebungsvariablen:

- ROS-MASTER-URI → IP oder Domain auf dem der Master läuft
- ROS-HOSTNAME → IP oder Domain des Rechners

Die Einträge können per **export**-Befehl **oder direkt** in der Datei `~/.bashrc` eingetragen werden:

`ROS_MASTER_URI=192.168.0.XXX:11311` **oder** `ROS_MASTER_URI=bb.local:11311`

`ROS_HOSTNAME=192.168.0.XXX` **oder** `ROS_HOSTNAME=bb.local`

**ACHTUNG:** Werden die Adressen in der `~/.bashrc` geändert, muss diese mit `source ~/.bashrc` in **jedem offenen Terminal** neu **geladen** werden, oder alle Terminals neu gestartet werden.

Die meisten Probleme beim Umgang mit ROS werden durch falsch gesetzte IP-Adressen bzw.

Probleme mit der Auflösung der Hostnames verursacht.

Die Adressen müssen die des aktuell verwendeten Netzwerks (USB/WLAN) sein.

### ROS auf dem PC testen (Skript Kapitel 10.1)

Für Einarbeitung in ROS folgende Abschnitte des ros.org Tutorials ([wiki.ros.org/ROS/Tutorials/](http://wiki.ros.org/ROS/Tutorials/)):

- Understanding ROS Nodes
- Understanding ROS Topics
- Understanding ROS Services and Parameters
- Writing a Simple Publisher and Subscriber (Python) (catkin)
- Examining the Simple Publisher and Subscriber

→ Beide ROS- Adressen zeigen auf den PC

### ROS auf dem BB testen (Skript Kapitel 10.2)

Vier Terminals öffnen und folgende Programme in je einem starten

- `roscore` # ROS-Master starten
- `roslaunch pat_intro srf02_talker.py` #Knoten starten
- `rostopic list` #Vorhandene Topics auflisten
- `rostopic echo /range_val` #Topic /range\_val ausgeben (CTRL+C zum beenden)
- `roslaunch pat_intro gpio_listener.py` #Knoten starten

→ Beide ROS-Adressen zeigen auf den BB

### ROS mit Simulink (Skript Kapitel 10.4)

- Beispielprogramm `count_pub.slx` (10.4.1)
  - Beispielprogramm `adcconv_subpub.slx` (10.4.2)
- ROS Master läuft auf dem BB → Adressen anpassen

**ACHTUNG:** Laufen die Simulink-Programme zu schnell (vergl. Zeit unten rechts) muss der RealTimePacer-Block eingebunden werden. Damit läuft das als Simulation gestartete Modell in realer Zeit.

[github.com/StefanMack/PAT-ROS/raw/master/Simulink/RealTime\\_Pacer.zip](https://github.com/StefanMack/PAT-ROS/raw/master/Simulink/RealTime_Pacer.zip)

**Unter Linux in Konsole Tab-Vervollständigung und zuletzt eingegebenen Befehle mit den Pfeiltasten durchblättern um Tipparbeit und Tippfehler zu reduzieren.**

# Linux CheatSheet (Skript Kapitel 12.2 + 12.3)

**ls** - directory listing  
**ls -al** - zeigt auch versteckte files)  
**cd dir** - change directory to dir  
**cd /dir** - change directory to dir absolute from root  
**cd** - change to home (~ → shortcut home dir)  
**cp file1 file2** - copy file1 to file2  
**cp -r dir1 dir2** - copy dir1 to dir2  
**!!** - repeats last command → sudo !!  
**↑** - shows last used command  
**chmod +x file** - make file executable  
**python3 myCode.py** - execute python program  
**gcc myCode.c -o myCode** - compile myCode.c to myCode  
**./myCode** - execute compiled program myCode  
**Ctrl+C** - halts the current command  
**Ctrl+D** - log out current terminal session  
**lsusb** - list all USB devices  
**dmesg | grep tty** - list Serial devices  
**ifconfig** - list active network connections  
**ifconfig -a** - list all network connections  
**sudo systemctl restart connman** - Netzwerkmanager neu starten  
**ping beaglebone.local** - ping to local domain name  
**ping 192.168.6.2** - ping to (local) IP address  
**export SETTINGS** - writes SETTINGS to ~/.bash.rc  
**source ~/.bashrc** - reloads terminal configuration  
**ssh user@host** - connect ssh to host (IP/hostname) as user  
**sftp://user@host** - connect sftp to host as user (in filemanager)

## NANO Editor

**nano filename** - use simple Nano Editor  
**Ctrl+O** - save file  
**Ctrl+X** - close file  
**Ctrl+W** - search in file

## Hardware

**i2cdetect -y -r 2** - listet alle I2C TN am Bus 2 auf  
**minicom -s** - startet UART Schnittstelle (Konfig über Menü)  
**minicom -b 115200 -o -D /dev/tty04** - startet UART mit 115200  
Baud an Device tty04

## ROS

**roscore** - ROS-Master starten  
**roslaunch pat\_intro name.py** - Knoten starten  
**rostopic list** - Vorhandene Topics auflisten  
**rostopic echo /Tname** - Daten zu Topic /Tname ausgeben  
**Ctrl+C** - Ausgabe stoppen