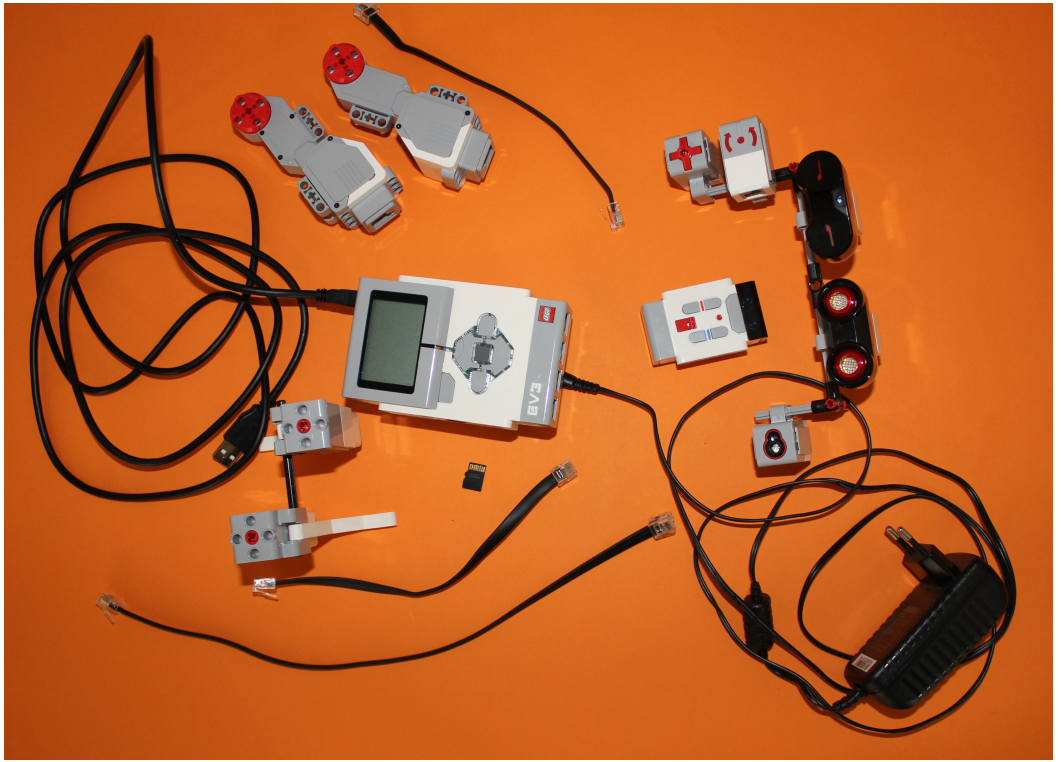
Irrungen und Wirrungen

Gforth auf dem EV3 von Lego®
Aktoren und Sensoren

Tagung Forth eV. 2015 Hannover

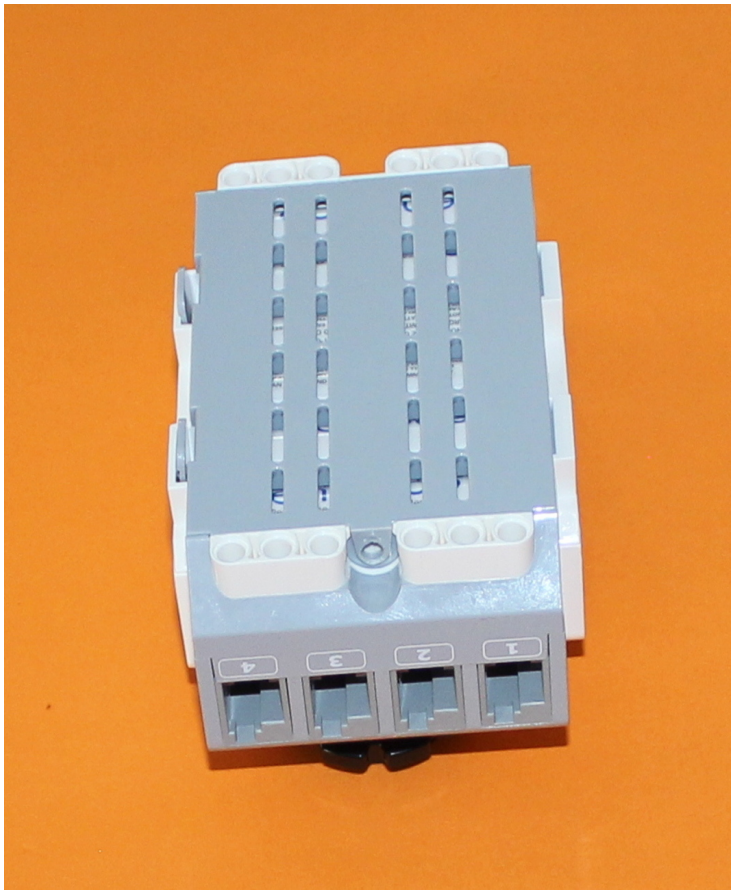
Der EV3

Das bringt ein handelsüblicher EV3 an Aktoren und Sensoren mit.



Der EV3

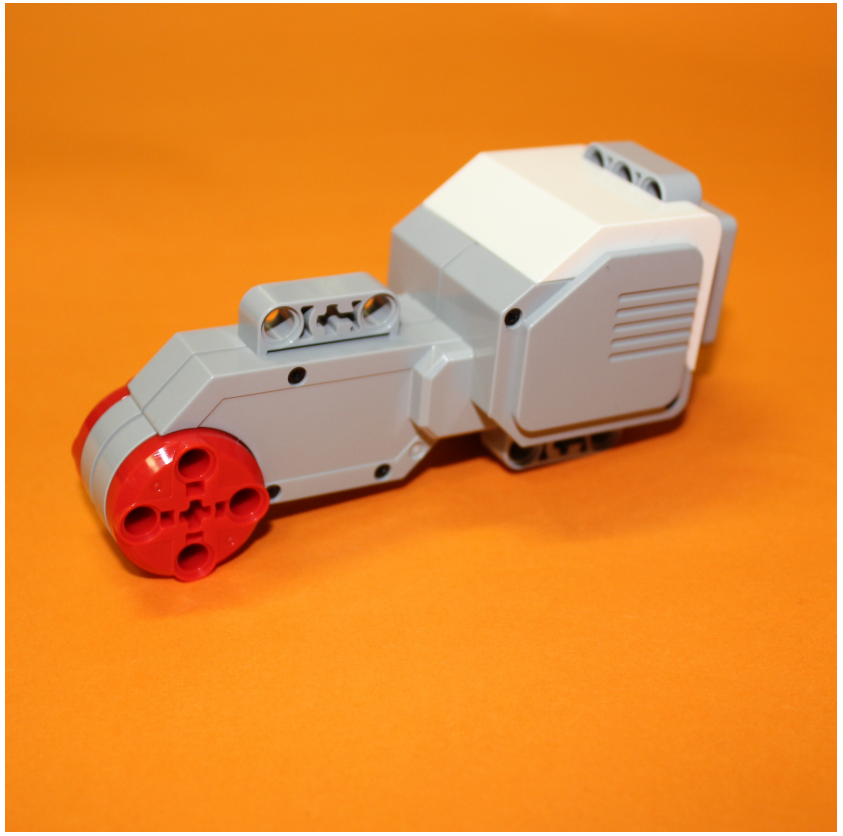
Handelsversion Spielwarenhandel



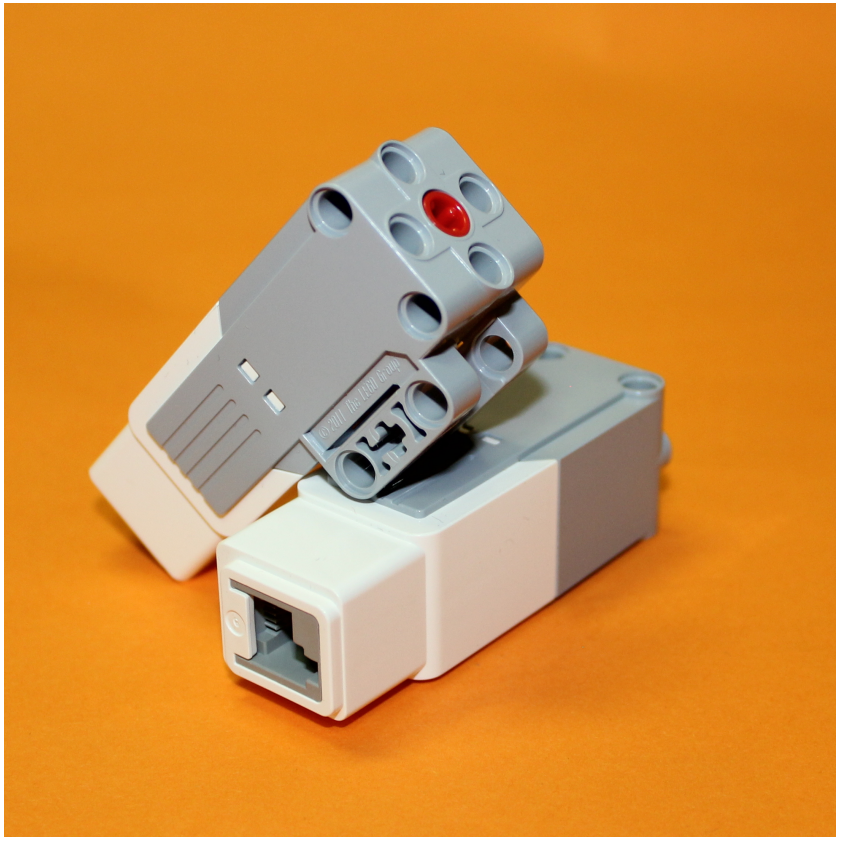
Schulversion Lehrmittelversand

Der EV3
Tachomotoren

Servomotor (intelligent)

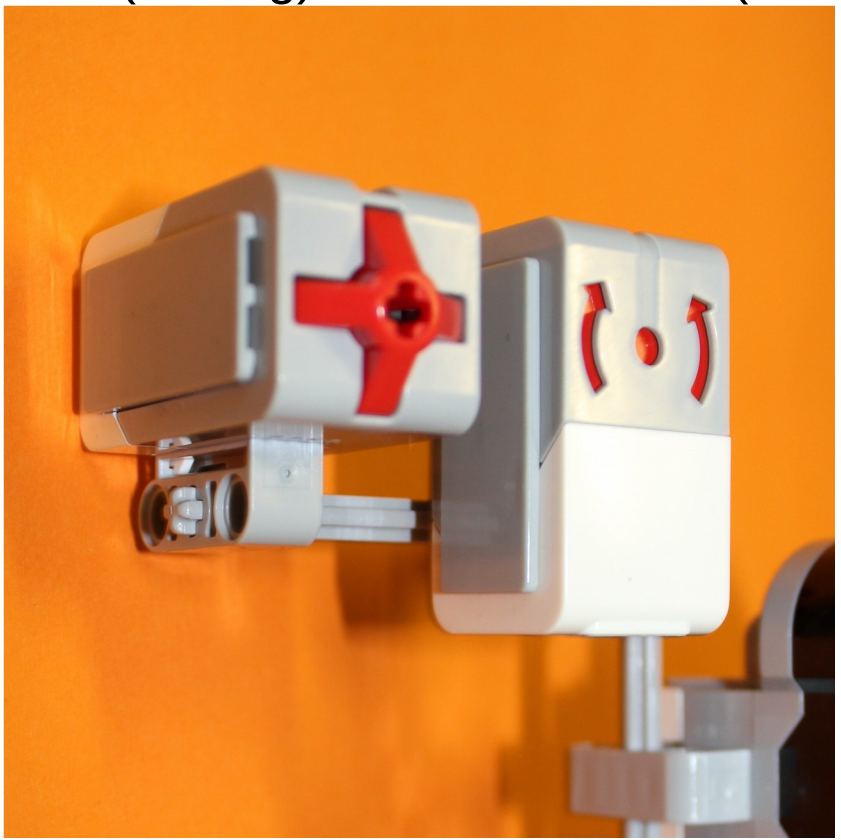


Servomotor medium (intelligent)

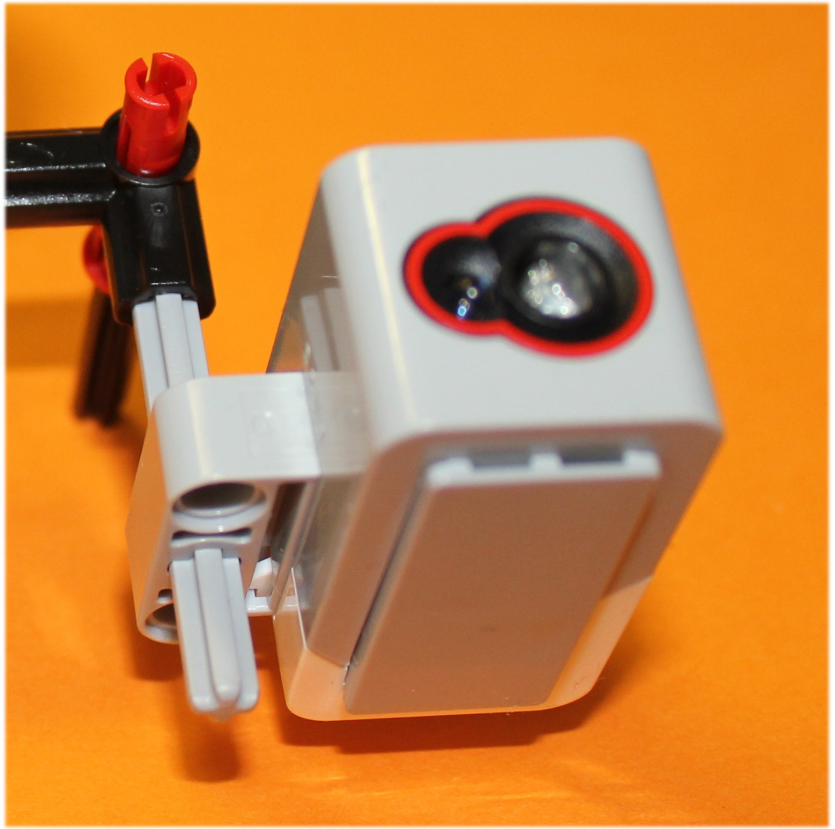


Der EV3
Sensoren

Taster (analog) Rotationssensor (UART)

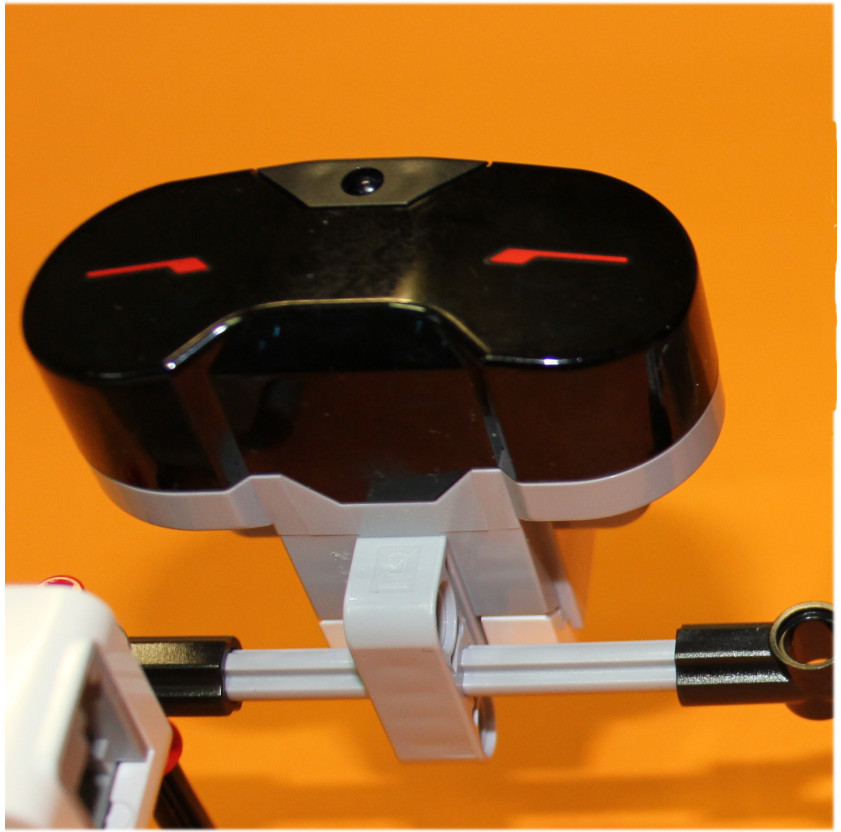


LED-Sensor (UART)

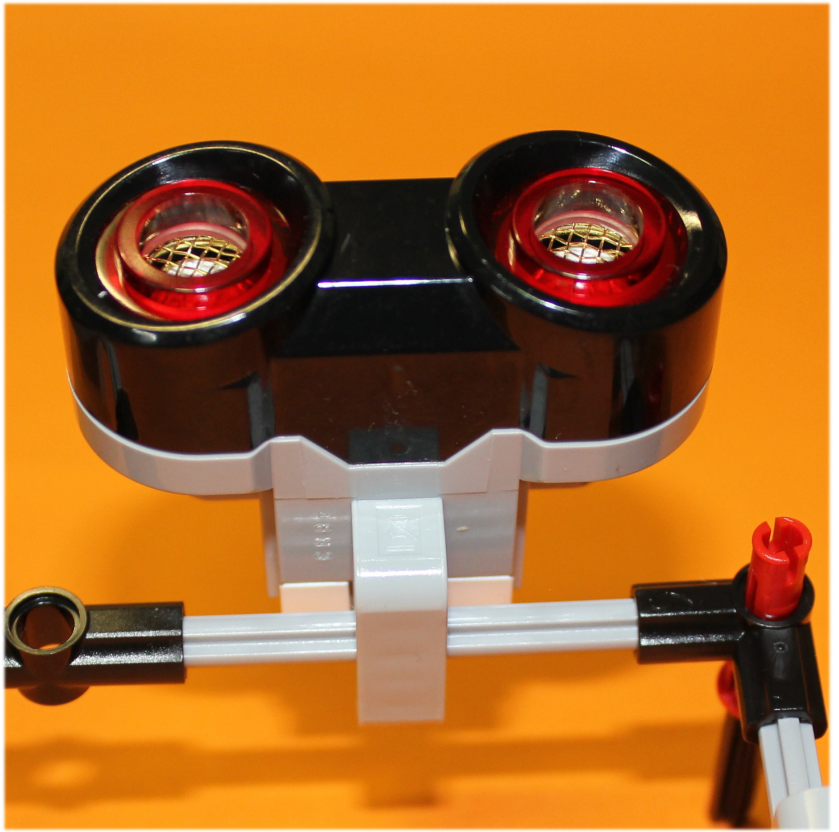


Der EV3
Sensoren

Infrarotsensor (UART)



Ultraschallsensor (UART)



Der EV3
Technische Daten 1

- ARM 9-Prozessor mit Linux-basiertem Betriebssystem
- Vier Eingänge zur Messwerterfassung mit einer Abtastrate von maximal 1000 Messungen pro Sekunde
- Vier Ausgänge zur Ausführung von Befehlen
- Integrierter Programmspeicher mit 16 MB Flash-Speicher und 64 MB RAM
- Mini-SDHC-Kartenleser zur Erweiterung des Speichers um 32 GB
- Beleuchtetes, dreifarbiges Bedienfeld mit sechs Tasten; die Farbe zeigt den jeweiligen Status des Steins an.
- Hochauflösendes Display (178 x 128 Pixel) zur detaillierten Anzeige von Graphen und zur Beobachtung der Sensordaten

Der EV3
Technische Daten 2

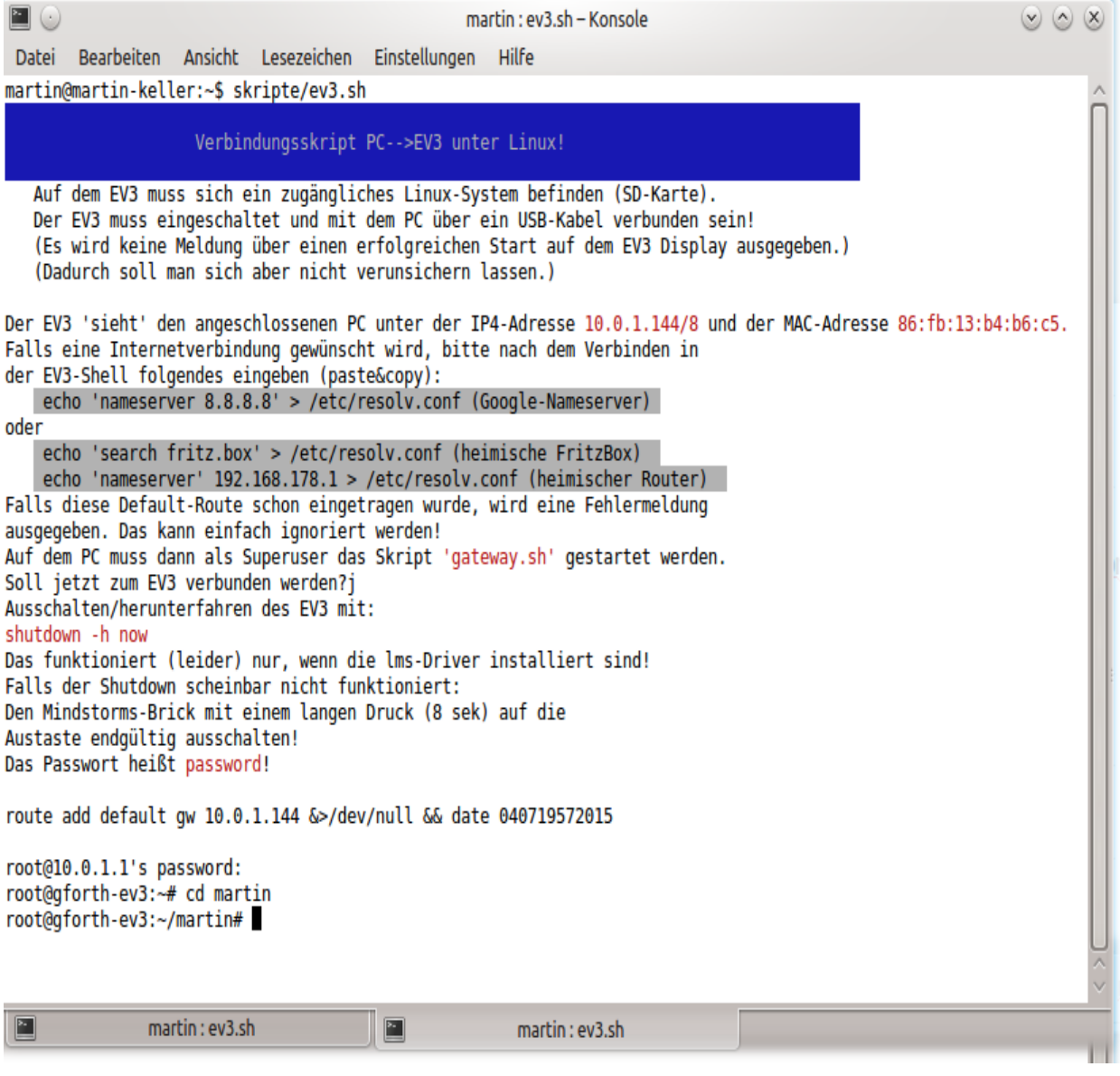
- Hochwertiger Lautsprecher
- Die auf dem Stein vorgenommene Programmierung und Messwerterfassung kann in die EV3-Software hochgeladen werden.
- Die Kommunikation zwischen Computer und Stein kann über den integrierten USB-Port oder drahtlos über das externe WLAN bzw. mit Bluetooth erfolgen.
- Der USB 2.0-Host ermöglicht die Kommunikation zwischen mehreren EV3-Steinen und gestattet zudem die drahtlose Kommunikation über WLAN und den Anschluss von USB-Speichersticks.
- Stromversorgung durch sechs AA-Batterien oder den EV3-Lithium-Ionen-Akku (Kapazität: 2050 mAh)

Linux 1

- http://sourceforge.net/projects/python-ev3/
- python-ev3.img.tar.bz2
- Version vom 2013-11-24 immer noch aktuell
- Image auf micro-SD-Karte kopieren
- In den Kartenschacht des EV3 schieben
- EV3 einschalten

Linux 2

- via microUSB mit dem (Linux-) PC verbinden
- Eine Konsole starten ssh Verbindung aufbauen (minicom, picocom, cutecom)
- Wenn gewünscht eine Internetverbindung über den PC aufbauen und …
- Ein Systemupdate durchführen
- Gforth 0.7.0 via aptget oder Freunden installieren (Achtung Version! Leider nur bedingt einsetzbar)
- So sieht der Startvorgang bei mir aus: (Danke Carsten!)

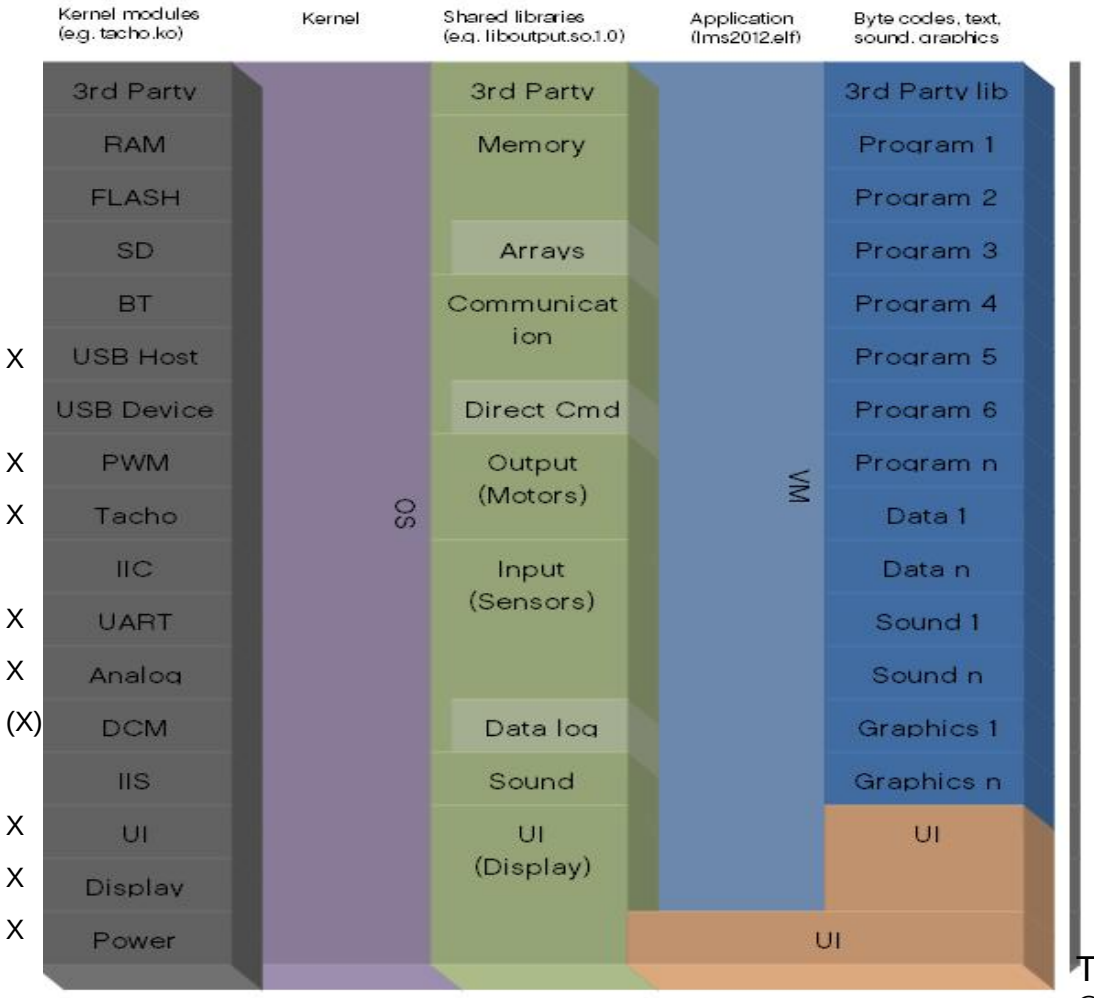

martin : ev3.sh – Konsole
Datei Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
martin@martin-keller:~$ skripte/ev3.sh
Verbindungsskript PC-->EV3 unter Linux!
Auf dem EV3 muss sich ein zugängliches Linux-System befinden (SD-Karte).
Der EV3 muss eingeschaltet und mit dem PC über ein USB-Kabel verbunden sein!
(Es wird keine Meldung über einen erfolgreichen Start auf dem EV3 Display ausgegeben.)
(Dadurch soll man sich aber nicht verunsichern lassen.)
Der EV3 'sieht' den angeschlossenen PC unter der IP4-Adresse 10.0.1.144/8 und der MAC-Adresse 86:fb:13:b4:b6:c5.
Falls eine Internetverbindung gewünscht wird, bitte nach dem Verbinden in
der EV3-Shell folgendes eingeben (paste&copy):
echo 'nameserver 8.8.8.8' > /etc/resolv.conf (Google-Nameserver)
oder
echo 'search fritz.box' > /etc/resolv.conf (heimische FritzBox)
echo 'nameserver' 192.168.178.1 > /etc/resolv.conf (heimischer Router)
Falls diese Default-Route schon eingetragen wurde, wird eine Fehlermeldung
ausgegeben. Das kann einfach ignoriert werden!
Auf dem PC muss dann als Superuser das Skript 'gateway.sh' gestartet werden.
Soll jetzt zum EV3 verbunden werden?j
Ausschalten/herunterfahren des EV3 mit:
shutdown -h now
Das funktioniert (leider) nur, wenn die lms-Driver installiert sind!
Falls der Shutdown scheinbar nicht funktioniert:
Den Mindstorms-Brick mit einem langen Druck (8 sek) auf die
Austaste endgültig ausschalten!
Das Passwort heißt password!
route add default gw 10.0.1.144 &>/dev/null && date 040719572015
root@10.0.1.1's password:
root@gforth-ev3:~# cd martin
root@gforth-ev3:~/martin#
martin : ev3.sh
martin : ev3.sh

Na endlich!

Forth

- gforth-0.7.9_20140402.bin.arm-none-linux-gnueabi
- 32-bit forth
- Eine extra 32-bit Toolchain benötigt (nur auf 64-bit Maschinen)
- Auf dem PC kompiliert und gepackt (tar.gz)
- Auf eine SD-Karte kopiert
- Auf dem EV3 entpackt und installiert
- Gforth-07.9 kann C-Funktionen einbinden. Wichtig!

Die EV3 Soft- Firmware



Die Kernelmodule

Der EV3 startet im Runlevel 2, dort wird ein Startscript 'lms2012.sh' aufegrufen, das die folgenden Kernelmodule einbindet und Devices erzeugt:

d_sound	/dev/lms_sound
d_analog	/dev/lms_analog
d_ui	/dev/lms_ui
d_pwm	/dev/lms_pwm
d_power	/dev/lms_power
d_uart	/dev/lms_uart
d_iic	/dev/lms_iic
g_ether	
?	/dev/lms_motor
?	/dev/lms_dcm
?	/dev/lms_usbhost
(pythonscript)	/dev/lms_bt

/lib/modules/2.6.33-rc4/lms2012/d_ppboard.ko
/lib/modules/2.6.33-rc4/lms2012/rt5572sta.ko
/lib/modules/2.6.33-rc4/lms2012/d_display.ko
/lib/modules/2.6.33-rc4/lms2012/d_sound.ko
/lib/modules/2.6.33-rc4/lms2012/d_uart.ko
/lib/modules/2.6.33-rc4/lms2012/d_usbdev.ko
/lib/modules/2.6.33-rc4/lms2012/d_analog.ko
/lib/modules/2.6.33-rc4/lms2012/d_ui.ko
/lib/modules/2.6.33-rc4/lms2012/d_bt.ko
/lib/modules/2.6.33-rc4/lms2012/d_usbhost.ko
/lib/modules/2.6.33-rc4/lms2012/d_pwm.ko
/lib/modules/2.6.33-rc4/lms2012/d_power.ko
/lib/modules/2.6.33-rc4/lms2012/d_iic.ko

Grundlegende Kommuniktion

Ich kenne bis jetzt vier Möglichkeiten

Output:

- Einfaches Schreiben in ein Device
- Schreiben in eine Memorymap
- Schreiben mit ioctl in ein Device

Input:

- Lesen aus einer Memorymap, bei „hoher“ Intelligenz gerne nach einem ictl Schreiben.

Die EV3 Sourcen

- Die Firmware ist Open Source

- https://github.com/topikachu/python-ev3
- https://github.com/mindboards/ev3sources
- https://github.com/mindboards/ev3dev
- https://github.com/hmml/ev3

Die EV3 Sourcen
Beispiele Dokumentation

- Known Devices

- Output Library

Die EV3 Sourcen

Beispiele Dokumentation

- lms2012.h
- UART Device html (Python)
- UART Device C

Der andere Weg!

Ralph Hempel und David Lechner sind auf ev3dev.org einen anderen (einfachereren?, besseren?) Weg gegangen.
Sie benutzen die virtuelle Maschine um Bytecodes an diese abzusetzen. So können sie alles, was Mindsorm-Lego in der virtuellen Maschine kann und müssen sich nicht mit Devices, ioctl u.Ä. herumschlagen.
Dies haben sie bisher für:

bash/dash	awk/gawk
perl	Lua
guile	ruby
python	Google Go (golang)
Node.js	

zugänglich gemacht.

Nun geht’s los!

Für die, die nur die Folien sehen können, ist jetzt Schluss!
Das Live-Publikum kann jetzt Einblicke in die gforth-lego Quellen nehmen und eine Demo, die viel von dem erreichten Stand zeigt, verfolgen.

Danke für Eure Aufmerksamkeit!