

EV3-Programmierung mit Python und PyCharm

Natascha Bomm und Christopher Kukkel

26. September 2017

Dieses Skript basiert auf dem alten Roboterskript „EV3 Programmierung mit Python 2.7 V1.5“ von Christopher Kukkel und Dustin Gärtner.

1 Einführung

Im Hardwarepraktikum werden neben den EWB-Versuchen auch Roboterversuche durchgeführt. Hierzu werden Lego Mindstorm EV3-Roboter genutzt. Diese können unter anderem mit der Programmiersprache Python gesteuert werden. Um die Grundlagen zu erlernen, verweisen wir auf das Buch “Python for Informatics”¹ und die Online-Quelle “Codeacademy”. Ersteres ist in der Universitätsbibliothek als Download verfügbar. Da Sie auf den Rechnern des Hardwarepraktikums keinen Zugang zum Internet haben, sollten Sie sich die nötigen Unterlagen anderweitig besorgen.

Für die Versuche erhalten Sie einen Roboter mit einer SD-Karte, auf der die Dateien gespeichert werden. Dieser Roboter (samt SD-Karte) kann unter Umständen auch von anderen Gruppen verwendet werden, daher werden die Dateien auf der SD-Karte bei jedem (Neu-)Start automatisch gelöscht. Stellen Sie deswegen sicher, dass die Dateien bei Ihnen auf dem Rechner (Partition H) gespeichert sind.

Lassen Sie die Roboter und Sensoren zusammengebaut. Falls Sie weitere Teile benötigen, wenden Sie sich an die Tutoren. Wenn Sie Sensoren oder den Roboter umbauen, bauen Sie diese am Ende jeder HaPra-Sitzung wieder zurück. Stellen Sie sicher, dass bei Beendigung Ihrer Hapra-Sitzung alle Materialien vollständig sind, da diese auch von anderen Gruppen verwendet werden (s. Bauanleitungen in Kapitel 6).

Fahren Sie am Ende jeder Sitzung die Rechner herunter und schalten Sie die Bildschirme aus.

¹http://www.pythongeeks.org/book_007.pdf

2 PyCharm

2.1 Projekt erstellen

Öffnen Sie zunächst die sich auf Ihrem Desktop befindliche Entwicklungsumgebung PyCharm. Wenn Sie aufgefordert werden Lizenzinformationen einzugeben, wählen Sie den Punkt „License Server“ aus, tragen als Adresse „<http://129.217.51.254:4455>“ ein und klicken auf „Aktivieren“. Als nächstes erstellen Sie ein neues Projekt. Klicken Sie dazu auf „Create New Project“ (Abbildung 1, rechts). Es empfiehlt sich, ein Projekt für alle Versuche zu erstellen, dann haben Sie alle Dateien auf einen Blick. Links an der Seite sollte bereits „Pure Python“ eingestellt sein, wenn nicht, müssen Sie das noch tun. Unter „Location“ können Sie den Pfad angeben, in dem Ihr Projekt gespeichert werden soll. Den voreingestellten Pfad sollten Sie ändern, damit sich Ihr Projekt auf Ihrer H-Partition und somit auf dem Server befindet (s. Abbildung 1, unten). Dann können Sie mit „Create“ bestätigen.



Abbildung 1: Links - PyCharm Entwicklungsumgebung; Rechts - Neues Projekt erstellen; Unten - Neues Projekt benennen

2.2 EV3 als SFTP-Server

Im daraufhin öffnenden Fenster sehen Sie links ihr Projekt, bislang noch ohne Dateien. Zunächst müssen Sie noch einige Einstellungen vornehmen. Gehen Sie dazu oben links auf „File > Settings“ und dann auf „Build, Execution, Deployment“. Klicken Sie am oberen Fensterrand auf das Plus, um einen SFTP-Server für den Austausch der Dateien mit dem Roboter hinzuzufügen. Benennen Sie ihn mit „EV3“, wie in Abbildung 2 gezeigt. Bestätigen Sie mit „OK“. Die daraufhin erscheinenden Einstellungsmöglichkeiten übernehmen Sie wie in Abbildung 3 gezeigt. Das Passwort des Roboters ist „maker“. Möglicherweise müssen Sie bei „SFTP host“ nur „ev3dev“ eingeben (s. Abschnitt 2.3).

Danach wählen Sie die Reiterkarte „Mappings“ aus (Abbildung 4). Der lokale Pfad sollte bereits eingetragen sein, für den Pfad auf dem Roboter geben Sie „/home/robot“ an. Der Webpfad wird nicht benötigt, aber wählen Sie die Option „Use this server as default“ aus. Danach bestätigen Sie mit „OK“ und öffnen die Einstellungen erneut. Wählen Sie unter „Build, Execution, Deployment > Deployment“ die Optionen aus. Ändern Sie „Upload changed files automatically to the default server“ von „Never“ zu „On explicit save action (Strg+S)“ (Abbildung 5). Wenn Sie den Roboter mit dem Rechner verbunden haben, wird Ihre geänderte Datei so bei jedem Speichern automatisch auf den Roboter heruntergeladen. Dann können Sie die Einstellungen wieder verlassen. Sollte das automatische Herunterladen nicht funktionieren, können Sie die Dateien auch wie in Abbildung 6 gezeigt über „Tools > Deployment > Upload to...“ manuell herunterladen.

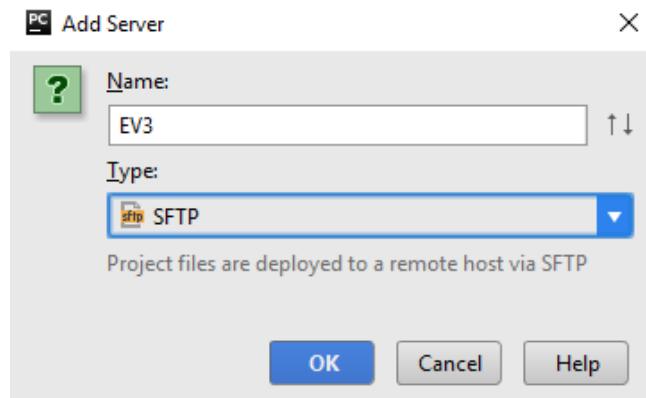


Abbildung 2: Neuen SFTP-Server hinzufügen

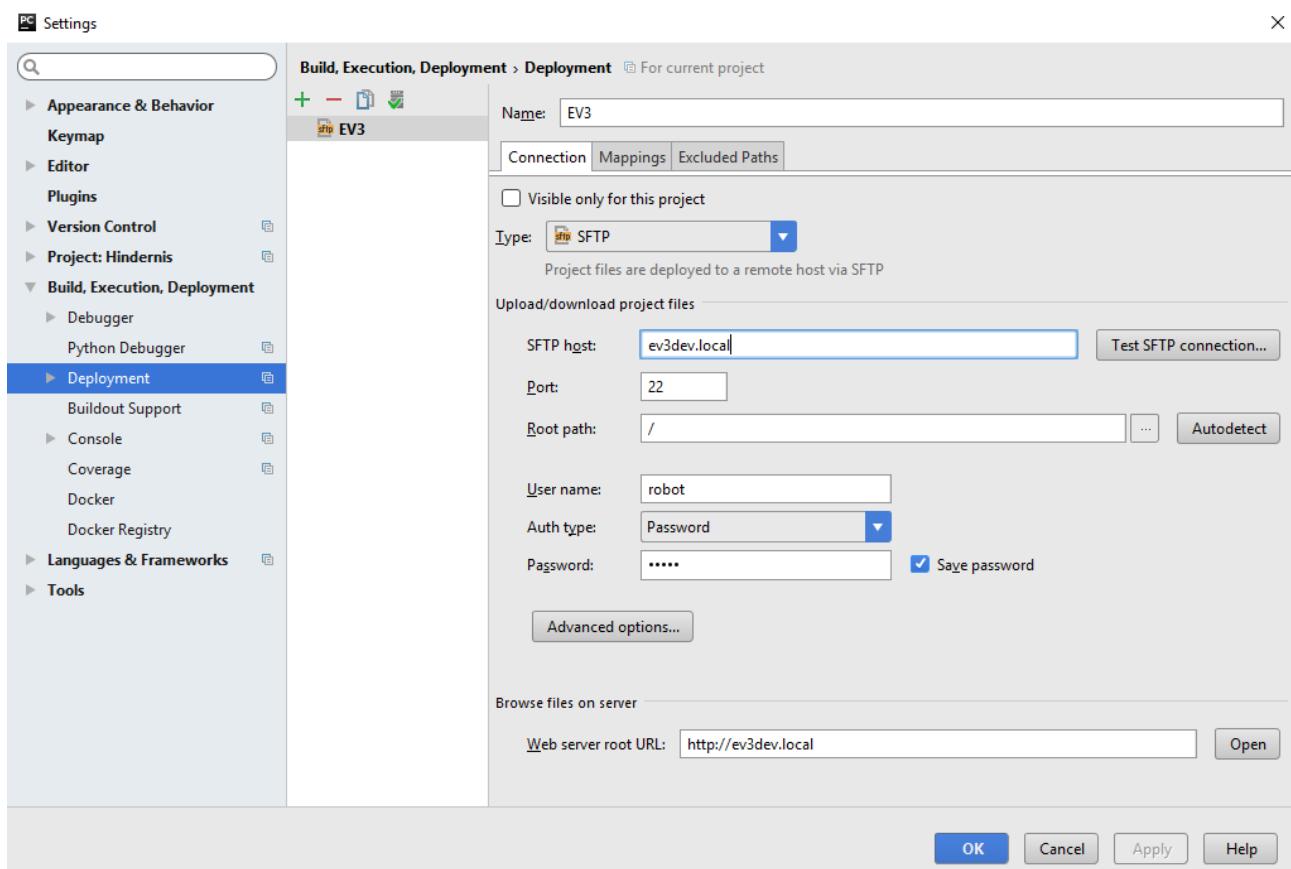


Abbildung 3: Verbindungseinstellungen

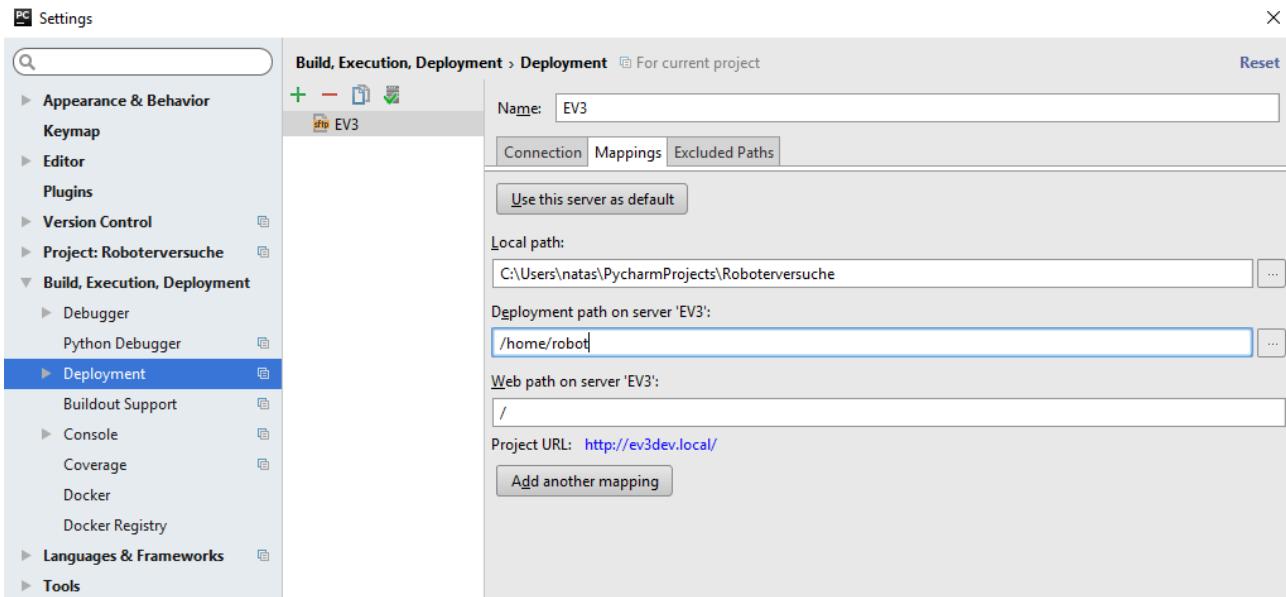


Abbildung 4: Mappingeinstellungen

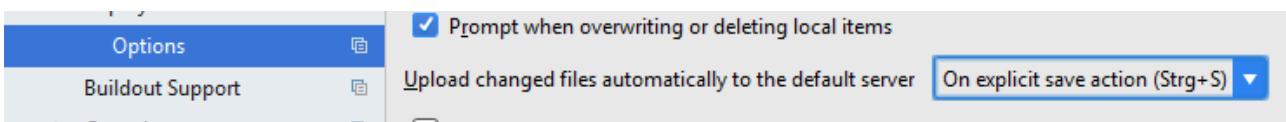


Abbildung 5: Option „Speichern und Herunterladen“

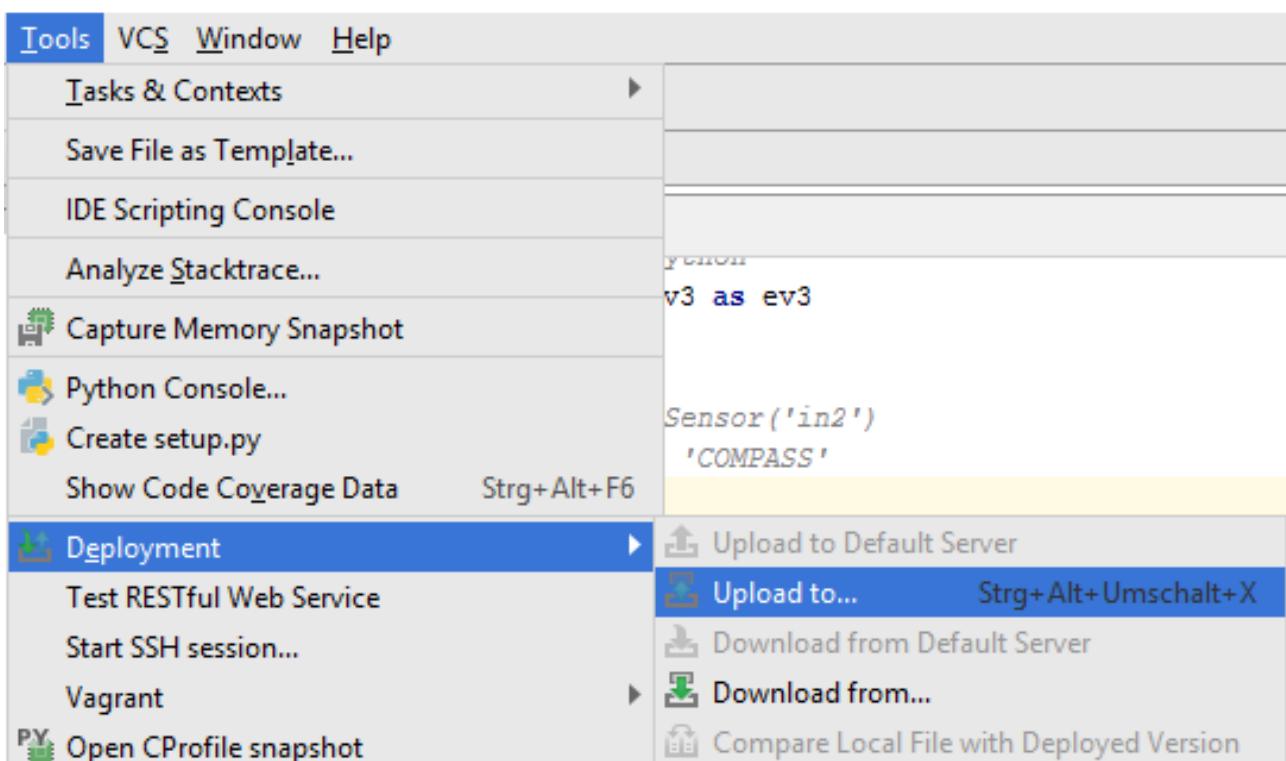


Abbildung 6: Manuelles Herunterladen

2.3 SSH-Verbindung mit dem Roboter

Als nächstes müssen Sie eine SSH-Verbindung mit dem Roboter einrichten. Gehen Sie dazu am oberen Bildschirmrand auf „Tools > Start SSH session...“. Sie werden dann gefragt, mit welchem Host Sie sich verbinden möchten. Als Optionen sollten „Edit credentials...“ und „EV3“ erscheinen. Wählen Sie „EV3“ aus. Die Nachfrage nach der Authenzität (Abbildung 7) bestätigen Sie mit „ja“. Das geöffnete Terminal sollte dann wie in Abbildung 9 aussehen.

Falls es bei der Verbindung Probleme gibt, können Sie folgendes versuchen: Ändern Sie in den Einstellungen den „SFTP host“ zu „ev3dev“ oder benennen Sie die LAN-Verbindung des Roboters um. Ob letzteres nötig ist, können Sie wie folgt überprüfen: Gehen Sie auf „Geräte und Drucker“. Wählen Sie den Roboter aus und öffnen Sie mit einem Rechtsklick auf den Roboter die Netzwerkeinstellungen. Gehen Sie dann in den geöffneten Einstellungen links auf „Adapter-Einstellungen“. Wenn es eine Verbindung gibt, die bereits „EV3“ oder ähnlich heißt, brauchen Sie nichts machen. Andernfalls bitten Sie einen der anwesenden Tutoren um Hilfe.

Tipp: Sie können EV3 als Standard-Server auswählen. Gehen Sie dazu in die Einstellungen zu „Tools > SSH Terminal“. Ändern Sie die Einstellung „Deployment server“ von „Select server on every run“ zu „EV3“ (Abbildung 8).

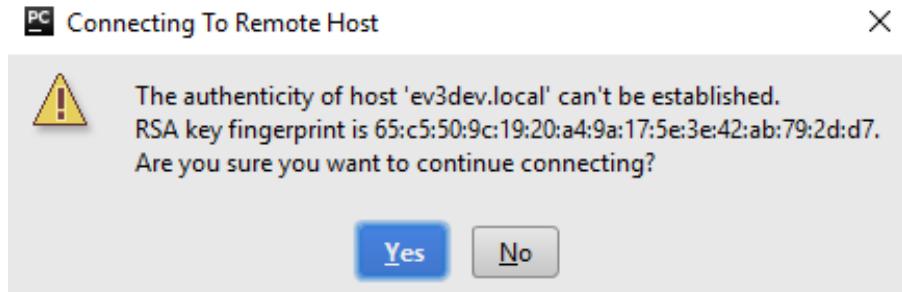


Abbildung 7: Authentizitäts-Nachfrage

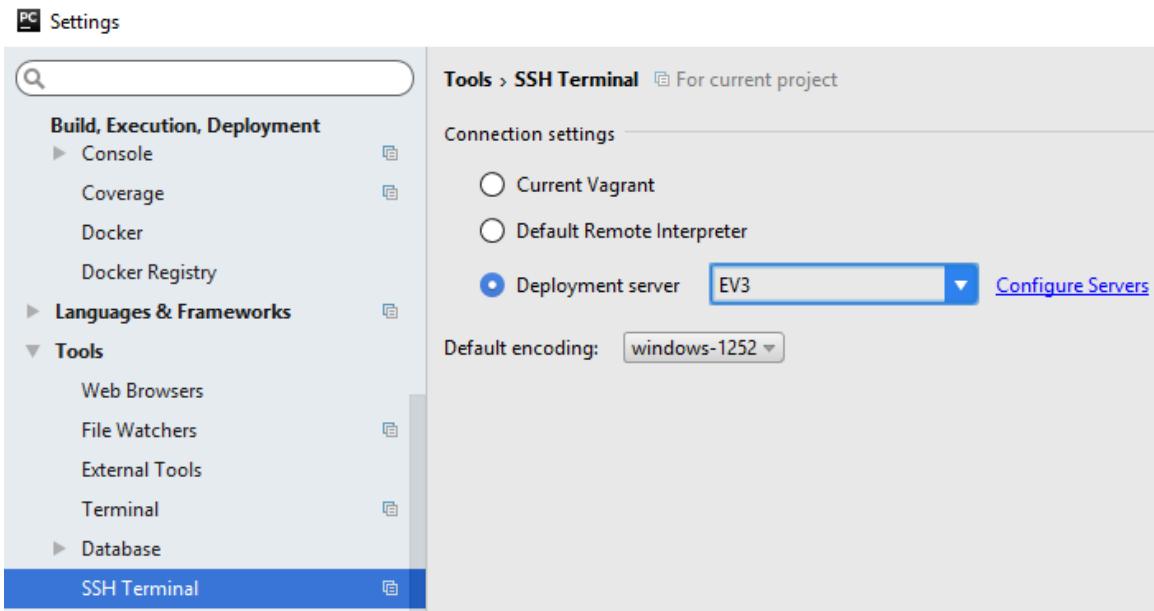


Abbildung 8: Standard-Server Einstellungen

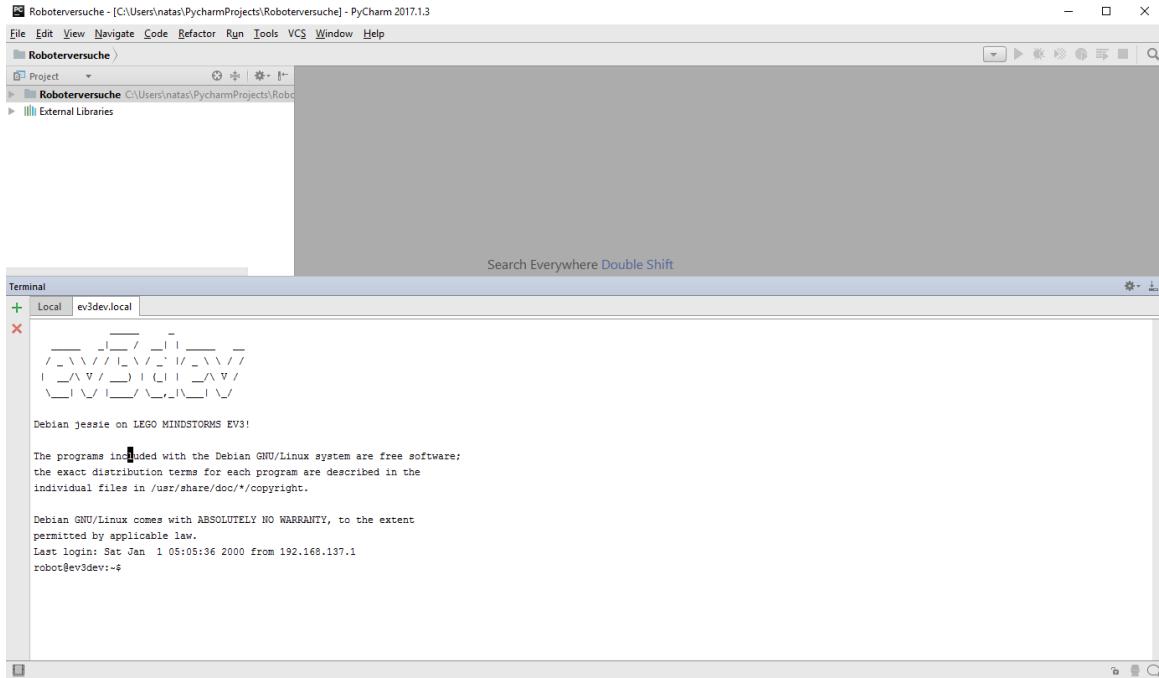


Abbildung 9: geöffnetes Terminal

2.4 Implementierung

Jetzt können Sie mit der Implementierung Ihrer Versuche beginnen. Klicken Sie dazu mit der rechten Maustaste auf ihr Projekt links im Projektexplorer und wählen Sie „New > Python File“ aus (Abbildung 10). Im geöffneten Dialogfenster (Abbildung 11) können Sie der Datei einen Namen geben. In Abbildung 12 sehen Sie ein Beispielprogramm. In der ersten Zeile Ihres Skripts muss immer folgender Code stehen:

```
#!/usr/bin/env python
```

Ihre Programme können Sie in der Konsole mit dem Befehl `python Dateiname.py` direkt vom Rechner ausführen. Wenn Sie die Programme auf dem Roboter ohne Verbindung zum Rechner ausführen möchten, müssen Sie dem Roboter dafür die notwendigen Rechte geben. Dies können Sie ebenfalls in der Konsole mit dem Befehl `chmod u+x Dateiname.py` machen. Wenn Sie die Dateien danach trotzdem noch nicht auf dem Roboter ausführen können, ändern Sie die Line Separators (s. Abbildung 13, im Hauptfenster unten rechts) zu „LF - Unix and OS X (\n“ und versuchen Sie es erneut. Wie die Programmierung der Roboter und Sensoren im Einzelnen funktioniert, wird Ihnen in den folgenden Abschnitten erklärt.

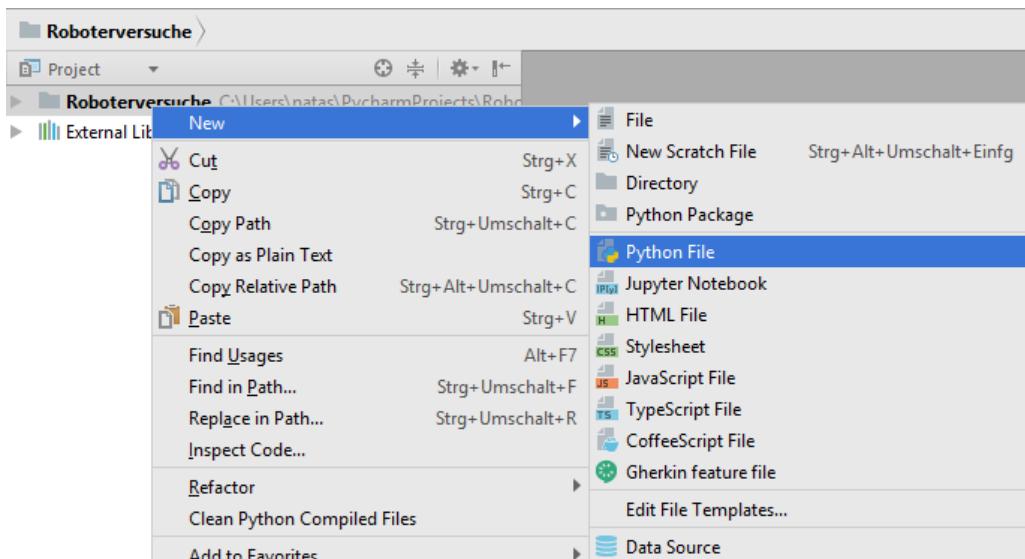


Abbildung 10: Auswahl - neue Python-Datei

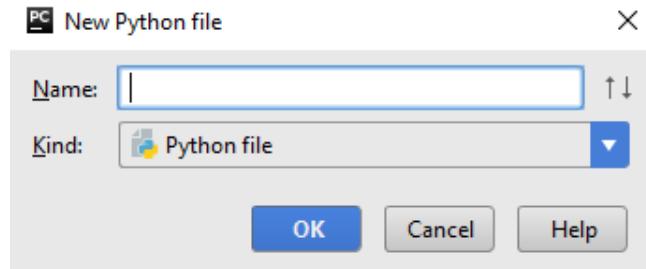


Abbildung 11: Benennung der neuen Python-Datei

```

#!/usr/bin/env python
import ev3dev.ev3 as ev3

motor_links = ev3.LargeMotor('outA')
motor_rechts = ev3.LargeMotor('outB')

def vorwaerts():
    motor_links.run_direct(duty_cycle_sp=75)
    motor_rechts.run_direct(duty_cycle_sp=75)

while True:
    vorwaerts()

```

Terminal output:

```

Debian jessie on LEGO MINDSTORMS EV3!
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Jan  1 05:05:36 2000 from 192.168.137.1
robot@ev3dev:~$
```

Abbildung 12: Beispielprogramm

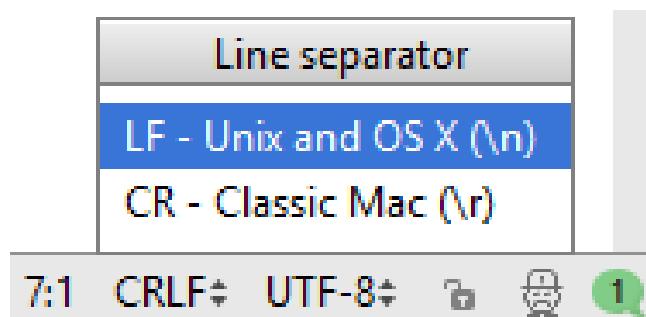


Abbildung 13: Ändern der Line Separators

3 Programmierung mit Python

Achten Sie darauf, dass alle angeschlossenen Motoren (Ports A-D) und Sensoren (Ports 1-4) definiert und auch benutzt werden, sonst können unvorhergesehene Fehler entstehen!

3.1 Motoren

Die normalen Motoren können wie folgt gesteuert werden:

```
#!/usr/bin/env python
import ev3dev.ev3 as ev3

motor_links = ev3.LargeMotor('outB')
motor_rechts = ev3.LargeMotor('outC')

#Vorwaerts
motor_links.run_direct(duty_cycle_sp=75)
motor_rechts.run_direct(duty_cycle_sp=75)

#Rueckwaerts
motor_links.run_direct(duty_cycle_sp=-75)
motor_rechts.run_direct(duty_cycle_sp=-75)
```

Über den Parameter `duty_cycle_sp` können Geschwindigkeit und Richtung eingestellt werden.

3.2 Funktionen definieren

Oft ist es sinnvoll, wiederkehrende Befehle in Funktionen auszulagern. Gerade bei der Motor-Steuerung spart dies sehr viel Tipparbeit!

```
#!/usr/bin/env python
i = 1
def schreibe():
    global i
    print "Ich schreibe mal was"
    i +=2

for i in xrange(0, 5):
    schreibe()
```

Wenn Variablen von außerhalb in Funktionen genutzt werden sollen, müssen diese zuvor als `global` definiert werden. So wird keine Schattenvariable erstellt, sondern die globale Version verwendet.

3.3 Textausgabe in Display und Konsole

In Python können Sie mittels `print "Beispieltext"` den Beispieltext ausgeben lassen. Wird das Programm in der Konsole aufgerufen, wird dort auch der Text ausgegeben. Wird das Programm auf dem Roboter geöffnet, wird der Text im Display angezeigt.

Mit dem Befehl `ev3.Sound.speak('Hello World').wait()` spricht der Roboter mit Ihnen.

Eine Tonausgabe erzeugen Sie mit `ev3.Sound.tone(1500, 500).wait()`. Der erste Parameter beschreibt die Frequenz in Hertz (Hz) und der zweite die Dauer in Millisekunden (ms).

3.4 Warten und Zeit

Um passiv zu warten, müssen Sie zunächst das `time`-Paket importieren.

```
import time
```

Anschließend können Sie folgenden Befehl nutzen: `time.sleep(sec)`. Mit `time.clock()` erhalten Sie die derzeitige CPU-Zeit in Sekunden.

3.5 Zufallszahlen

Um Zufallszahlen zu erzeugen, muss ein weiteres Paket importiert werden.

```
from random import randint
```

Mit diesem kann die Funktion `randint(von, bis einschließlich)` genutzt werden. So erzeugt `randint(1, 100)` zufällige Zahlen zwischen 1 und 100.

3.6 Sensoren

Die Sensoren müssen vor Programmstart mit dem Roboter verbunden sein und dürfen während der gesamten Ausführung nicht von dem Roboter getrennt werden. Des Weiteren existieren bei manchen Sensoren verschiedene Modi. Diese können mit „`mode = 'mode name'`“ verändert werden. Die verschiedenen Modi werden im Folgenden zunächst in einer Tabelle aufgelistet, gefolgt von einem Minimalbeispiel in Python. Wie Sie die Sensoren am besten an Ihrem Roboter anbringen, können Sie in den Aufbauanleitungen in Abschnitt 6 nachgucken.

3.6.1 TouchSensor

mode	Description	units	Values
TOUCH	Button state	none	State (0 or 1)

```
#!/usr/bin/env python
import ev3dev.ev3 as ev3
ts = ev3.TouchSensor()
ts.mode = 'TOUCH'
while True :
    # Beide Varianten funktionieren
    if ts.value() :
        print ts.value()
    if ts.value() == True :
        print ts.value()
```

3.6.2 ColorSensor

mode	Description	units	Values
COL-REFLECT	Reflected light	pct (percent)	Reflected light intensity (0 to 100)
COL-AMBIENT	Ambient light	pct (percent)	Ambient light intensity (0 to 100)
COL-COLOR	Color	col (color)	Detected color (0 to 7)

Die LED nimmt je nach Modus unterschiedliche Farben an.

- Reflected light: red
- Ambient light: blue
- Color: Alle LEDs sind an

```
#!/usr/bin/env python
import ev3dev.ev3 as ev3
cs = ev3.ColorSensor()
cs.mode = 'COL-COLOR'
while True :
    print cs.value()
```

Die zurückgegebenen Farben sollten folgende Nummerierung haben:

Farbe	Keine	Schwarz	Blau	Grün	Gelb	Rot	Weiß	Braun
Wert	0	1	2	3	4	5	6	7

3.6.3 Ultrasonic-Sensor

mode	Description	units	Values
US-DIST-CM	Continuous measurement	mm (Millimeter)	Distance (0-2550)

```
#!/usr/bin/env python
import ev3dev.ev3 as ev3
us = ev3.UltrasonicSensor()
us.mode = 'US-DIST-CM'
while True :
    print us.value()
```

3.6.4 Gyro-Sensor

mode	Description	units	Values
GYRO-ANG	Angle	deg (degrees)	value: Angle (-32768 to 32767)

Die Position des Sensors zum Start des Programms gibt den 0-Punkt an. Dreht sich der Sensor im Uhrzeigersinn steigt der Wert. Bei einer vollen Umdrehung ist der Wert des Sensors 360. Bei einer Drehung gegen den Uhrzeigersinn, wird der Wert negativ.

```
#!/usr/bin/env python
import ev3dev.ev3 as ev3
gs = ev3.GyroSensor()
gs.mode = 'GYRO-ANG'
while True :
    print gs.value()
```

3.6.5 Sound-Sensor

Der Sound-Sensor stammt von dem veralteten NXT Roboter und muss deshalb anders angesteuert werden.

mode	Description	units	Values
ANALOG-1	?	Millivolt (mV)	0 to 5000

```
#!/usr/bin/env python
import ev3dev.ev3 as ev3
ss = ev3.Sensor()
ss.mode = 'ANALOG-1'
while True :
    print ss.value()
```

Der zurückgegebene Wert ist die Spannung in Millivolt, also zwischen 0 und 5.000. Je lauter es ist, desto geringer ist die Spannung.

3.7 Weitere Sensoren von Drittanbietern

Falls Sie noch weitere Sensoren von Drittanbietern (s. Abschnitt 5) für Ihre Versuchen verwenden möchten, können Sie diese fast genauso wie den Sound-Sensor ansteuern. Dafür müssen Sie einen anderen Mode eingeben. Welcher das ist, können Sie auf dem Roboter im Device-Browser unter Sensoren nachsehen. Dann können Sie Ihren (bereits angeschlossenen) Sensor einfach auswählen und den Mode ablesen.

3.8 Mehrere Sensoren des gleichen Typs

Um mehrere Sensoren des gleichen Typs anzuschließen, muss zusätzlich der Port angegeben werden, an dem diese angeschlossen sind:

```
#!/usr/bin/env python
import ev3dev.ev3 as ev3
us1 = ev3.UltrasonicSensor('in1')
us2 = ev3.UltrasonicSensor('in2')
```

4 Tipps und Tricks

- Wenn Sie bei einem der Versuche nicht weiterkommen, versuchen Sie an verschiedenen Stellen des Programms Textausgaben zu erzeugen, um Ihr Problem zu lösen.
- Wenn der Roboter sich aufgehängt hat und Sie keine Eingaben mehr machen können, halten Sie die Mittel- und Zurück-Taste so lange gedrückt, bis der Roboter neu startet.
- Wenn Sie sich die Funktionsweise der Sensoren bewusst machen wollen oder der Meinung sind, einer Ihrer Sensoren sei fehlerhaft, dann können Sie sich auf dem Roboter unter „Device Browser > Sensors > *Sensor* > Watch Values“ die ausgegebenen Werte anzeigen lassen.
- Wenn Sie unvorhergesehene Fehlermeldungen bekommen, überprüfen Sie, ob die Motoren an den Ports A-D und die Sensoren an 1-4 angeschlossen sind. Bei mehreren Sensoren des gleichen Typs überprüfen Sie, ob Sie die Input-Ports richtig angegeben haben (s. Abschnitt 3.7). Überprüfen Sie außerdem, ob Sie möglicherweise im Code nicht verwendete Sensoren oder Motoren initialisiert oder verwendete nicht initialisiert haben.
- Eine Fehlermeldung kann auch entstehen, wenn die erste Zeile Ihres Programms nicht wie in Abschnitt 2.4 beschrieben aussieht.
- Wenn sich eines Ihrer Programme nicht ohne Verbindung zum Rechner auf dem Roboter starten lässt, überprüfen Sie, ob Sie dem Roboter die nötigen Rechte geben haben (s. Abschnitt 2.4).
- Bei Python sind Einrückungen Teil der Syntax. Überprüfen Sie bei Fehlern zunächst, ob alle Zeilen richtig eingerückt sind. Beachten Sie, dass es einen Unterschied zwischen „Tabs“ und „Leerzeichen“ gibt. Verwenden Sie Tabs für Ihre Einrückungen.
- Achten Sie bei den Gyro-Sensoren darauf, dass Sie den Roboter nicht zu schnell drehen. Je schneller die Drehung ist, desto ungenauer ist der Sensor. Außerdem sollte der Sensor beim Einsticken vollständig still stehen, sonst liefert der Sensor ständig steigende Werte.

5 Liste der Sensoren

EV3 Sensoren

- Color Sensor - Farbsensor
- Ultrasonic Sensor - Ultraschallsensor (Entfernung)
- Gyro Sensor - Gyroskop
- Touch Sensor - Berührungssensor

NXT Sensoren

- Sound Sensor - Schalldruckmessgerät (Mikrofon)

Drittanbietersensoren - HiTechnic

- Magnetic Compass Sensor - Kompass
- Accelerometer Sensor - Beschleunigungssensor
- Color Sensor - Farbsensor
- Gyroscopic Sensor - Gyroskop

6 Bauanleitungen

6.1 Materialien

Folgende Materialien befinden sich in der Roboterkiste:

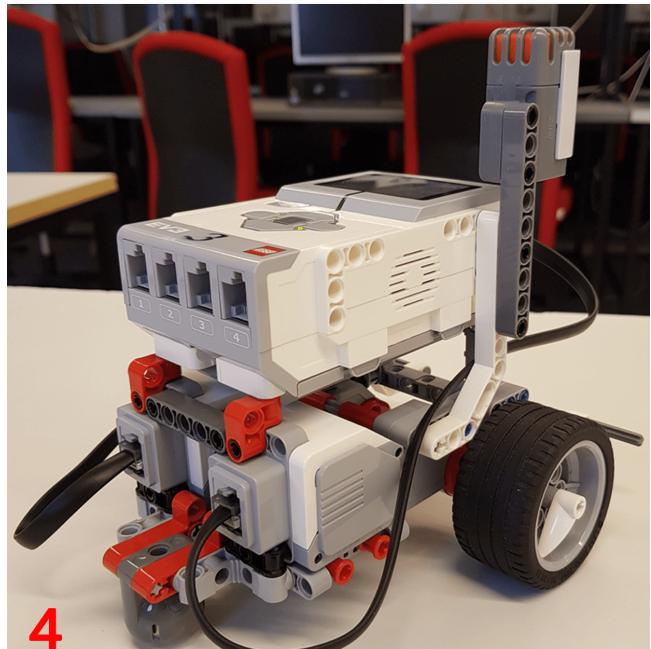
- EV3-Roboter
- USB-Kabel
- 2 Sensorkabel
- Ultraschallsensor
- Gyrosensor
- Farbsensor
- Soundsensor
- Touchsensor
- Lego-Anleitung
- Stromkabel

Stellen Sie sicher, dass die Materialien in der Roboterkiste stets vollständig sind.

6.2 Roboter

Falls Sie etwas an dem Aufbau des Roboters geändert haben, befolgen Sie bitte die Aufbauanleitung (S.7 bis S. 38) von Lego, die sich in ihrer Roboterkiste befindet, um den Roboter wieder in seinen Ausgangszustand zu bringen.

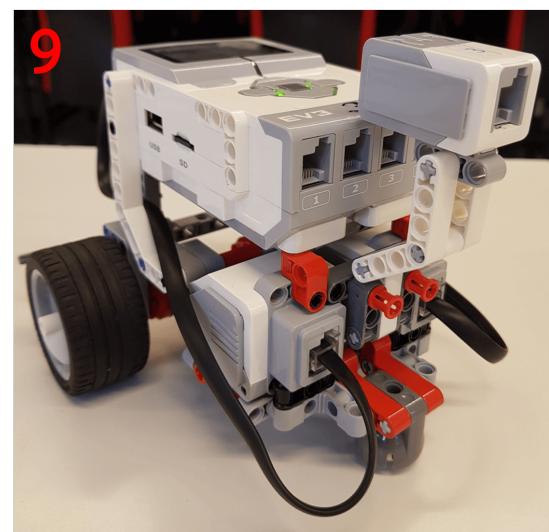
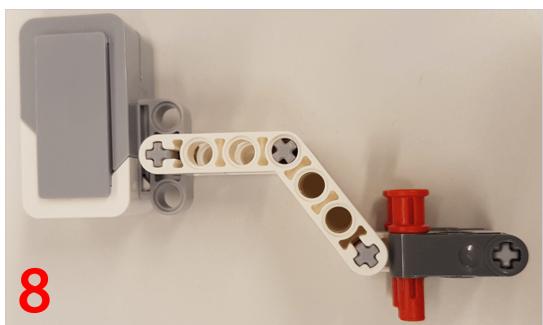
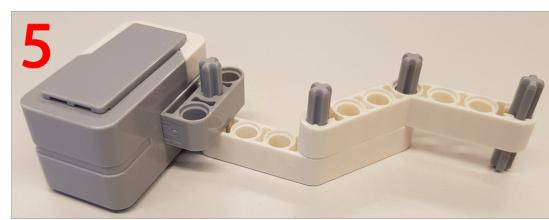
6.3 Soundsensor



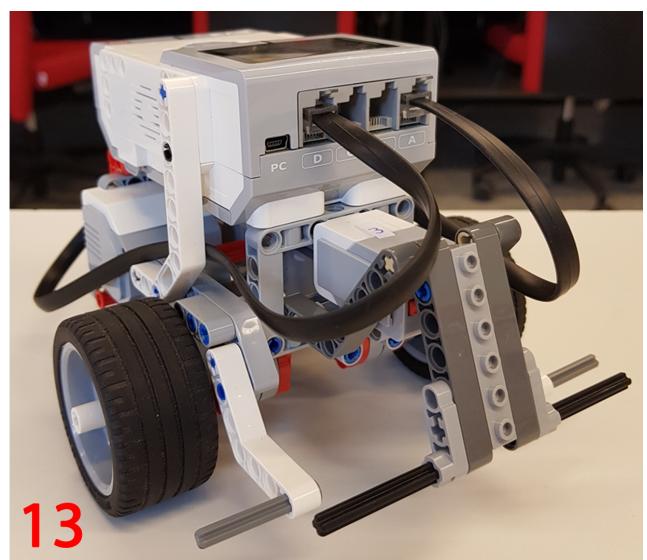
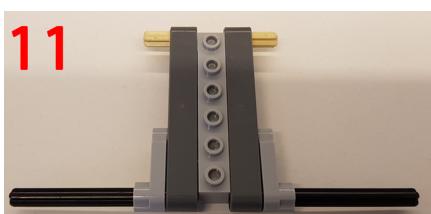
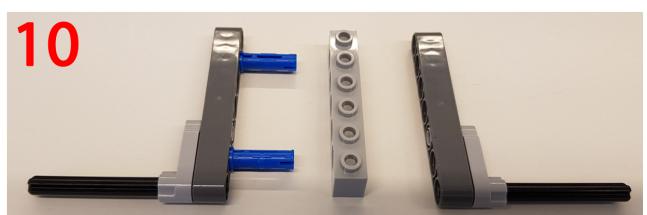
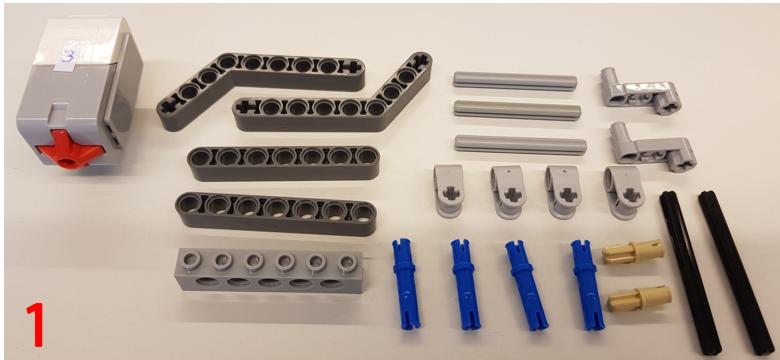
6.4 Ultraschallsensor



6.5 Gyrosensor



6.6 Touchsensor



6.7 Farbsensor

