

Space Workshop Dokumentation



TECHNISCHE
UNIVERSITÄT
DARMSTADT

NeXT Generation on Campus
TU Darmstadt



Aufgabe 1 Die EV3-Steuereinheit



Abbildung 1: EV3-Brick

Motorausgänge	MotorPort.A MotorPort.B MotorPort.C MotorPort.D
Links	LEFT
Rechts	RIGHT
Oben	UP
Unten	DOWN
Mitte	ENTER
Oben-Links	ESCAPE
Sensoreingänge	SensorPort.S1 SensorPort.S2 SensorPort.S3 SensorPort.S4

Tabelle 1: Tastenbenennung

Durch Drücken der mittleren und unteren Taste wird das laufende Programm beendet.

Aufgabe 2 Motorsteuerung

Dem Roboter stehen 2 starke große und ein schwächerer mittelgroßer Motor zur Verfügung.

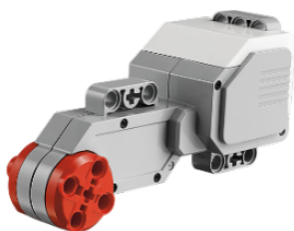


Abbildung 2: großer Motor



Abbildung 3: mittlerer Motor

Motorausgang	lejos.hardware.port.MotorPort
großer Motor	lejos.hardware.motor.EV3LargeRegulatedMotor
mittlerer Motor	lejos.hardware.motor.EV3LargeMediumRegulatedMotor

Tabelle 2: benötigte Imports

forward()	Motor dreht sich vorwärts
backward()	Motor dreht sich rückwärts
stop()	Motor stoppt
rotate(int a)	Motor dreht sich um a Grad
setSpeed(int x)	setzt die Geschwindigkeit des Motors Das Maximum ist hierbei 800

Tabelle 3: wichtige Methoden

Um die Motoren verwenden zu können, muss zuerst der Motorport und der entsprechende Motor oben in der Datei importiert werden.

Bsp.: import lejos.hardware.motor.EV3LargeRegulatedMotor;

Als nächstes muss Im folgenden Beispiel steht „name“ für einen frei wählbaren Namen und „X“ für den Port, also A, B, C oder D.

EV3LargeRegulatedMotor name = new EV3LargeRegulatedMotor(MotorPort.X);

Bsp.: EV3LargeRegulatedMotor motor = new EV3LargeRegulatedMotor(MotorPort.A);

Damit sich der Motor bewegt, müssen dem Motor nach folgendem Muster eine der gegebenen Methoden gegeben werden.

name.Methode;

Bsp.: motor.forward();

Aufgabe 3 Warten

Beim Programmieren ist es immer wieder notwendig, Pausen einzubauen.

Hierfür ist der Import **lejos.utility.Delay** notwendig.

Mit der Methode **Delay.msDelay(int time)** wird eine Pause mit „time “ Millisekunden ausgeführt. (1000 Milisekunden sind eine Sekunde)

Bsp.: Delay.msDelay(2000);

Aufgabe 4 Tasten

Tasten	lejos.hardware.Button
--------	-----------------------

Tabelle 4: benötigter Import

isDown()	wahr, wenn Taste gedrückt
isUp()	wahr, wenn Taste nicht gedrückt
waitForPress()	wartet, bis Taste gedrückt

Tabelle 5: wichtige Methoden

Die Methoden werden nach folgendem Muster aufgerufen: **Button.name.Methode** wobei der „name“ für die Bezeichnung der Taste steht.

Bsp.: Button.LEFT.waitForPress();

Mit der Methode „**Button.LEDPattern(int i)**“ wird die LED unter den Tasten gesteuert. Hierbei leuchtet die LED mit unterschiedlichem **i** von null bis acht in verschiedenen Rhythmen und Farben.

Aufgabe 5 Lautsprecher

Lautsprecher	lejos.hardware.Sound
--------------	----------------------

Tabelle 6: benötigter Import

beep()	spielt einen Ton ab
twoBeeps()	spielt den gleichen Ton zweimal ab
beepSequence	spielt eine absteigende Tonfolge ab
beepSequenceUp()	spielt eine aufsteigende Tonfolge ab
buzz()	summt
setVolume(int vol)	setzt die Lautstärke auf den Wert vol (0-100)

Tabelle 7: wichtige Methoden

Die Methoden werden nach folgendem Muster aufgerufen: **Sound.Methode**
Bsp.: Sound.Beep();

Aufgabe 6 Display



Abbildung 4: Koordinatensystems des Displays

Display	lejos.hardware.lcd.LCD
---------	------------------------

Tabelle 8: benötigter Import

drawString(String str, int x, int y)	Zeigt einen Text an, beginnend in Spalte x und Zeile y
drawInt(int i, int x, int y)	Zeigt eine Ganzzahl an, beginnend in Spalte x und Zeile y
clear()	löscht den Inhalt des Displays
clear(int y)	löscht den Inhalt der y-ten Zeile
scroll()	verschiebt den Inhalt um eine Zeile nach oben

Tabelle 9: wichtige Methoden

Das Koordinatensystem auf dem Display hat seinen Ursprung (0,0) oben links und geht in x-Richtung nach rechts 16 Spalten und in y-Richtung nach unten 8 Zeilen.

Das Display wird angesprochen mit: **LCD.Methode**

Bsp.: **LCD.drawString(„Hallo Welt!“, 0, 0);**

Aufgabe 7 Sensoren



(a) Ultraschallsensor



(b) Farbsensor



(c) Winkelsensor



(d) Berührungssensor

Sensorausgang	lejos.hardware.port.SensorPort
Ultraschallsensor	lejos.hardware.sensor.EV3UltrasonicSensor
Farbsensor	lejos.hardware.sensor.EV3ColorSensor
Winkelsensor	lejos.hardware.sensor.EV3GyroSensor
Berührungssensor	lejos.hardware.sensor.EV3TouchSensor

Tabelle 10: benötigte Imports

setCurrentMode (String mode)	setzt den Modus des Sensors
sampleSize()	gibt die Anzahl der zurückgegeben Werte zurück
fetchsample(float[] signal, int offset)	Sensor misst und speichert es im Array signal ab Stelle offset
getColorID()	Farbsensor gibt erkannte Farbe zurück (-1=keine Farbe, 0=Rot, 1=Grün, 2=Blau, 3=Gelb, 6=Weiß, 7=Schwarz, 13=Braun)

Tabelle 11: wichtige Methoden

Ultraschallsensor	Distance (Sensor gibt Distanz in Metern zurück)
Farbsensor	Ambient (Sensor gibt Umgebungshelligkeit in Werten zwischen 0 und 1 zurück)
Winkelsensor	(Sensor gibt Winkel zurück)
Berührungssensor	(Sensor gibt zurück, ob Taste gedrückt (1) oder nicht gedrückt (0) ist)

Tabelle 12: Modi

Die Benutzung der Sensoren ist auf den ersten Blick etwas kompliziert, jedoch folgt die Benutzung einem festen Aufbau.

Zuerst muss wie bei den Motoren der Sensor benannt werden.
Bsp.: EV3UltrasonicSensor ultra = new EV3UltrasonicSensor(SensorPort.S1)

Als nächstes muss der Sensor in den richtigen Modus gesetzt werden mit
sensorname.setCurrentMode(String mode)
Bsp.: ultra.setCurrentMode(„Distance“);

Der nächste Schritt ist das Anlegen eines Arrays, in dem die Daten gespeichert werden. Hierbei wird direkt mit **sampleSize()** die Größe gesetzt. **Bsp.: float[] signal = new float[sensorname.sampleSize()];**

Um neue Daten zu erfassen wird mit dem Sensor die Methode **fetchSample** aufgerufen und im vorher angelegten Array gespeichert.
Bsp.: ultra.fetchSample(signal, 0);

Aufgabe 8 Kurze Übersicht über Java

Schleife mit Bedingung	<pre>while(Bedingung) {Programmcode}</pre> <p>Beispiel: <code>while(i<100){...}</code></p>
Zählschleife	<pre>for(Start; Bedingung; Zählschritte) {Programmcode}</pre> <p>Beispiel: <code>for(int i=0;i<10;i++){...}</code></p>
Bedingung	<pre>if(Bedingung)</pre> <p>{wenn die Bedingung wahr ist, wird dieser Code ausgeführt}</p> <pre>else</pre> <p>{wenn die Bedingung falsch ist, wird dieser Code ausgeführt}</p>

Aufgabe 9 Quellen

Die Dokumentation mit weiteren Methoden befindet sich unter:
lejos.org/ev3/docs/

Die Lejos Software kann auf der folgenden Seite heruntergeladen werden. Weiterhin gibt es dort ausführliche Anleitungen zum Einrichten auf dem EV3-Roboter:
lejos.sourceforge.io

Der genutzte Java-Editor ist kostenfrei im Netz verfügbar:
javaeditor.org

Bei Fragen stehen wir immer gerne zur Verfügung:
next-generation@etit.tu-darmstadt.de

Die Bilder sind der Internetseite des offiziellen Lego-Onlineshops lego.com entnommen. Die Urheberrechte befinden sich im Besitz der LEGO Gruppe, diese Anleitung ist unabhängig und wurde von der LEGO Gruppe weder autorisiert noch gesponsert.