

Mindroid Workshop Dokumentation

Fachgebiet Echtzeitsysteme / MAKI
TU Darmstadt



TECHNISCHE
UNIVERSITÄT
DARMSTADT

1 Wichtige Funktionen

Hier eine kleine Übersicht über die wichtigsten Funktionen beim Programmieren der Roboter.

1.1 Fahren

Mögliche Eingabewerte für den *speed*-Parameter liegen zwischen 0 und 1000. Eine maximale Geschwindigkeit von 300 sollte ausreichen. Niedrigere Geschwindigkeiten schonen den Akku. Die Distanz wird im *distance*-Parameter immer als Kommazahl in Zentimetern (cm) angegeben (z.B.: 20cm werden als 20.0f angegeben)

Typ	Methode und Beschreibung
void	setMotorSpeed(int speed) Bestimmt die Geschwindigkeit für Fahrmethoden ohne <i>speed</i> -Parameter.
void void	forward() backward() Fahren mit der von <i>setMotorSpeed(...)</i> gesetzten Geschwindigkeit.
void void	driveDistanceForward(float distance) driveDistanceBackward(float distance) Fahren mit der von <i>setMotorSpeed(...)</i> gesetzten Geschwindigkeit Die Distanz muss in Zentimetern angegeben werden.
void void void void	forward(int speed) backward(int speed) driveDistanceForward(float distance, int speed) driveDistanceBackward(float distance, int speed) Wie oben, nur dass der <i>speed</i> -Parameter die von <i>setMotorSpeed()</i> gesetzte Geschwindigkeit überschreibt. Nach Beendigung des Aufrufs, wird wieder die vorher gesetzte Geschwindigkeit genutzt.
void void void void	turnLeft(int degrees) turnRight(int degrees) turnLeft(int degrees, int speed) turnRight(int degrees, int speed) Dreht den Roboter um den im <i>degrees</i> -Parameter bestimmten Wert. Der <i>Speed</i> -Parameter verhält sich wie bei den anderen Methoden.
void	stop() Stoppt sofort alle Motoren.

1.2 Sensoren

Typ	Methode und Beschreibung
float	<code>getAngle()</code> Liefert den Winkel des Gyrosensors in Grad
float	<code>getDistance()</code> Liefert die vom Ultraschallsensor gemessene Distanz in Zentimetern
Colors Colors	<code>getLeftColor()</code> <code>getRightColor()</code> Liefert den Wert des Linken/Rechten Farbsensors Farbwerte: Colors.BLACK, Colors.BLUE, Colors.BROWN, Colors.GREEN, Colors.RED, Colors.WHITE, Colors.YELLOW, Colors.NONE

1.3 Kommunikation

Typ	Methode und Beschreibung
boolean	<code>hasMessage()</code> Prüft ob Nachricht vorhanden ist
MindroidMessage	<code>getNextMessage()</code> Ruft nächste Nachricht ab
void	<code>broadcastMessage(String message)</code> Sendet eine Nachricht an alle Roboter
String	<code>getRobotID()</code> Gibt den Namen des Roboters zurück.
void	<code>sendLogMessage(String logmessage)</code> Sendet eine Nachricht an den Message Server
void	<code>sendMessage(String destination, String message)</code> Sendet eine Nachricht an den <i>destination</i> -Roboter

Um eine Nachricht zu empfangen, muss zuerst mit `hasMessage()` überprüft werden ob eine Nachricht vorhanden ist. Liefert `hasMessage()` `true` zurück, kann mit `getNextMessage()` eine Nachricht abgerufen werden. Das Beispiel in Listing 1 zeigt wie das geht.

```
if (hasMessage()){  
    String msg = getNextMessage().getContent();  
}
```

Listing 1: Beispiel zum Abrufen einer Nachricht

`broadcastMessage(...)` schickt eine Nachricht an alle mit dem selben Message-Server verbundenen Roboter.

1.4 Brick

1.4.1 Display

Typ	Methode und Beschreibung
void	<code>clearDisplay()</code> Löscht den Aktuellen Inhalt des Displays
void	<code>drawString(String text, Textsize textsize, int xPosition, int yPosition)</code> Schreibt den im <i>text</i> -Parameter gegebenen Text auf das Display an die durch <i>xPosition</i> und <i>yPosition</i> definierte Stelle (siehe Abb. 1.1) mit Textgröße <i>textsize</i>

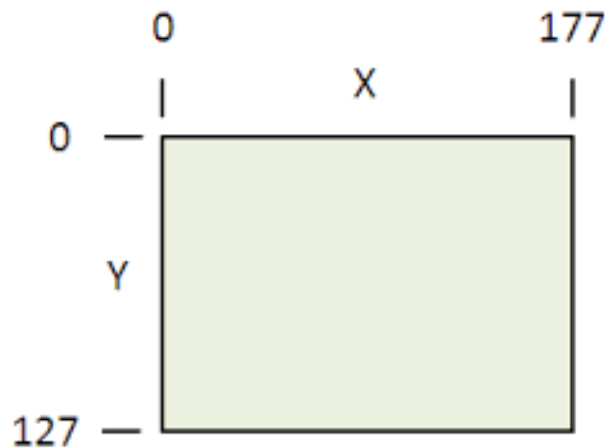


Abbildung 1.1: Koordinaten der Pixel des Displays des EV3¹

1.4.2 Buttons

Typ	Methode und Beschreibung
boolean	<code>isDownButtonClicked()</code>
boolean	<code>isEnterButtonClicked()</code>
boolean	<code>isLeftButtonClicked()</code>
boolean	<code>isRightButtonClicked()</code>
boolean	<code>isUpButtonClicked()</code>

Die Funktionen liefern *true* wenn der entsprechende Button gedrückt wurde. Die Benennung der Buttons kannst du Abbildung 2.1 auf Seite 5 entnehmen

¹ https://services.informatik.hs-mannheim.de/~ihme/lectures/LEGO_Files/01_Anfaenger_Graphisch_EV3_BadenBaden.pdf

1.4.3 Sound

Typ	Methode und Beschreibung
void	setSoundVolume(int volume)
void	playBeepSequenceDown()
void	playBeepSequenceUp()
void	playBuzzSound()
void	playDoubleBeep()
void	playSingleBeep()

Der Parameter *volume* nimmt Werte von 0 bis 10 entgegen.

1.4.4 LED

Typ	Methode und Beschreibung
void	setLED(int mode) Lässt die LED des EV3 im angegebenen Modus leuchten Der Parameter <i>mode</i> kann entweder als Ganzzahl von 0 bis 9 oder als Konstante angegeben werden. Siehe Tabelle 1.1

Tabelle 1.1: Funktion der einzelnen Modi der LED

Wert	Modus (Parameter <i>mode</i>)	Farbe	Intervall
	Konstante		
0	LED_OFF	Aus	Aus
1	LED_GREEN_ON	Grün	Dauer
2	LED_GREEN_BLINKING	Grün	Blinken
3	LED_GREEN_FAST_BLINKING	Grün	Schnell Blinken
4	LED_YELLOW_ON	Gelb	Dauer
5	LED_YELLOW_BLINKING	Gelb	Blinken
6	LED_YELLOW_FAST_BLINKING	Gelb	Schnell Blinken
7	LED_RED_ON	Rot	Dauer
8	LED_RED_BLINKING	Rot	Blinken
9	LED_RED_FAST_BLINKING	Rot	Schnell Blinken

2 EV3 Tasten

Abbildung 2.1 zeigt dir wie die Tasten am EV3-Brick genannt werden. Die Enter-Taste wird zum Bestätigen genutzt, mit der Escape-Taste, geht es ein Menü zurück.

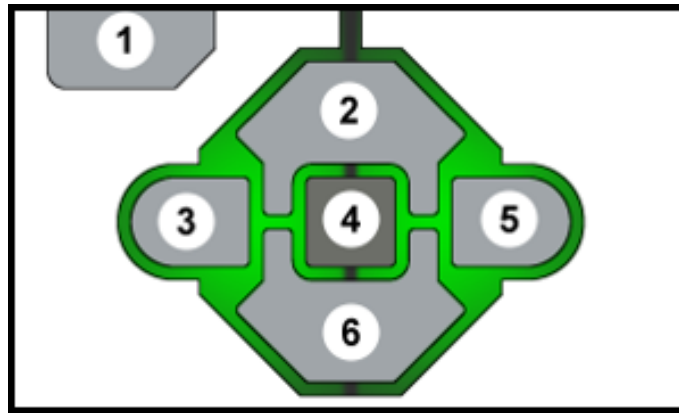


Abbildung 2.1: EV3-Tastenbelegung²

Die Bedeutung der Tasten kannst du der folgenden Aufzählung entnehmen.

1. **Escape / Zurück**
2. **Up / Hoch**
3. **Left / Links**
4. **Enter / Bestätigen**
5. **Right / Rechts**
6. **Down / Unten**

3 PAN Einrichtung

Wird im Hauptmenü noch nicht die richtige IP-Adresse angezeigt, müssen zuerst die PAN³-Einstellungen korrigiert werden.

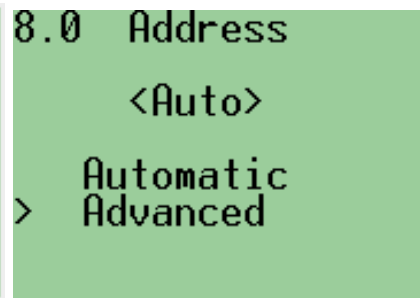
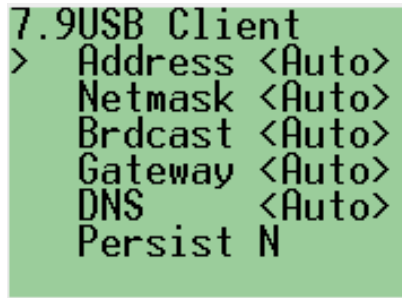
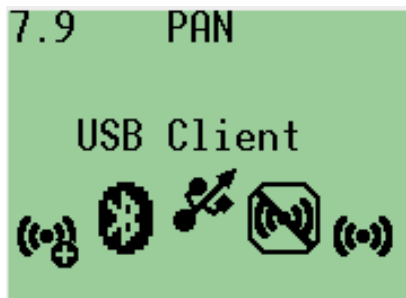
1. Dazu musst du zuerst in das **PAN-Menü** des Roboters navigieren. Wechsle mit den Links-/Rechts-Tasten bis du den Menüpunkt **PAN** siehst und betätige die **Auswahltaste**.



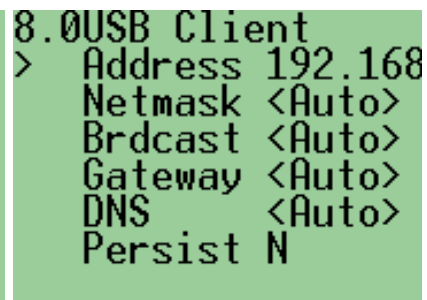
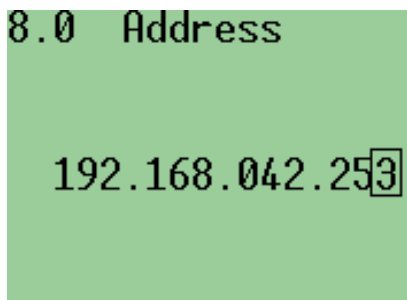
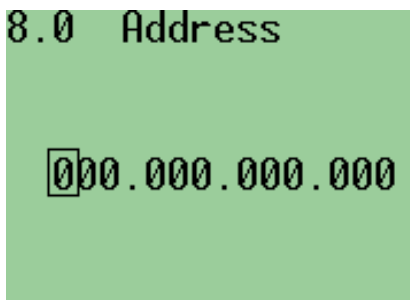
2. Nun navigierst du durch das Menü des Roboters wie auf den Bildern zu sehen und bestätigst jeweils mit der Auswahltaste: **USB-Client - Address - Advanced**

² Quelle <http://www.ev3dev.org/images/ev3/labeled-buttons.png>

³ PAN = Personal Area Network



Nun musst du die IP Adresse 192.168.42.253 einstellen. Dazu navigierst du mit den rechts-/links-Tasten zu den einzelnen Ziffern und änderst deren Wert mit den oben-/unten-Tasten. Orientiere dich an den Bildern! Am Ende bestätigst du wieder mit der Enter-Taste.



3. Mit der **Zurück**-Taste kommst du wieder in das Hauptmenü und die Einstellungen werden übernommen.



4 Troubleshooting

4.1 Installation über WLAN funktioniert nicht

Im Message-Server über **File->Connected Devices** schauen ob alle Smartphones in der Liste auftauchen und der ADB-state auf **connected** steht. Ist dies nicht der Fall, tippe in der App auf **TRENNEN** und stelle die Verbindung danach erneut her. Falls das nicht klappt, kontaktiere einen Betreuer. Falls das Installieren per WLAN gar nicht mehr funktioniert, kann jederzeit eine USB-Verbindung zwischen Smartphone und PC hergestellt werden und darüber die App installiert werden.

5 Sensorbelegung

Tabelle 5 zeigt die standardmäßige Sensorbelegung, wie sie in der App unter "Mein Roboter" definiert sein muss.

Tabelle ?? zeigt den standardmäßigen Motoranschluss, wie er in der App unter "Mein Roboter" definiert sein muss.

Anschluss	Sensortyp	Modus
1	Farbe	ColorID
2	Ultraschall	Distance
3	Gyroskop	Angle
4	Farbe	ColorID