

Dokumentation Modul „chassiESP32“ für Knotech Calli:bot

Hardware: Calli:bot mit Microcontrollerboard ESP32 und Spannungsversorgung 3,3 Volt über Akku (Batterie), Boardkopplung mit Calli:bot über I²C-Bus

Modul import: `from chassiESP32 import *`

Motoren

Methode	Aktion
<code>getMotor(m)</code>	m = Modus: 0 = links, 1 = rechts, 2 = beide Motoren, Rückgabe: Liste mit drei Werten
<code>setMotor(m, r, g)</code>	m = Modus (s.o.), r = Richtung: 0 = vorwärts, 1 = rückwärts, g = Geschwindigkeit von 0..255
<code>stopMotor(m)</code>	m = Modus (0 = links, 1 = rechts oder 2 = beide)

LEDRot

Methode	Aktion
<code>getLEDRot()</code>	... liefert eine Liste des Zustandes der beiden LED [links, rechts] binär 0 oder 1
<code>setLEDRot(ledrl, ledrr)</code>	... setzt die linke und rechte LED mit 0 = aus und 1 = an
<code>setLEDRotLinksOn()</code>	... schaltet linke LED ein
<code>setLEDRotLinksOff()</code>	... schaltet linke LED aus
<code>setLEDRotRechtsOn()</code>	... schaltet rechts LED ein
<code>setLEDRotRechtsOff()</code>	... schaltet rechts LED aus

RGB-LED

Methode	Aktion																				
	<code>led =</code> vornLinks = 1 hintenLinks = 2 vornRechts = 3 hintenRechts = 4 farbe = 0..7 hell = 0..16																				
<code>setRGBLed(led, farbe, hell)</code>	Daten für RGB-LEDs: Zusammensetzung eines RGB-Datenbytes: Bit (0..2) = Farbe (Wert 0..7) Bit 3 = nicht genutzt Bit (4..7) = Helligkeit (0 = aus, 15 Stufen)																				
	Das Farbspektrum der RGB-LEDs ist stark reduziert, um die Programmierung seitens der Schüler zu vereinfachen.																				
	<table><thead><tr><th>Wert</th><th>Farbe</th><th></th><th></th></tr></thead><tbody><tr><td>0</td><td>schwarz (aus)</td><td>1</td><td>grün</td></tr><tr><td>2</td><td>rot</td><td>3</td><td>gelb</td></tr><tr><td>4</td><td>blau</td><td>5</td><td>türkis</td></tr><tr><td>6</td><td>violett</td><td>7</td><td>weiß</td></tr></tbody></table>	Wert	Farbe			0	schwarz (aus)	1	grün	2	rot	3	gelb	4	blau	5	türkis	6	violett	7	weiß
Wert	Farbe																				
0	schwarz (aus)	1	grün																		
2	rot	3	gelb																		
4	blau	5	türkis																		
6	violett	7	weiß																		

Ultraschallsensor

Methode	Aktion
<code>getUltraschallSensor()</code>	... liefert den 16-Bit Bytewert der Entfernung in mm

LinienSensor

Methode	Aktion
<code>getLinienSensor()</code>	Methode liefert als Bytewert den Zustand der Liniensensoren. 0=beide dunkel, 1=links dunkel, 2=rechts dunkel, 3=beide hell Bit[0]: linker Sensor: 0=dunkel, 1=hell Bit[1]: rechter Sensor: 0=dunkel, 1=hell

Empfehlung:

```
myChassi = Chassis() # oder: mc = Chassis()
mc.setMotor(0, 0, 50)
print(mc.getMotor(2))
mc.setLedRotLinksOn()
print(int.from_bytes(mc.getUltraschallSensor(), 'big'), ' mm')
print(int.from_bytes(mc.getLinienSensor(), 'small'))
mc.i2cStop()
```