

mBot2 **TUTORIAL**







Inhalt

1	Notv	vendi	ge Software3
	1.1	Die l	Programmoberfläche6
	1.2	Basi	sgerät und Erweiterungen7
2	Verb	indu	ng herstellen9
	2.1	Verk	indung über USB-Kabel:9
	2.2	Verk	indung über Bluetooth:
3	Inter	aktiv	e Verbindung ("Spielemodus"):13
4	Einfü	ihren	de Aufgaben:17
	4.1	Grur	ndlegende Aufgaben:
	4.1.	1	Geradlinige Bewegungen:
	4.1.	2	Drehungen:
	4.1.	3	Signale:
	4.2	Einfa	ache Bewegungsaufgaben:19
	4.3	Verv	vendung der Sensoren des mBot2:20
	4.3.	1	Lichtsensor:
	4.3.	2	Schallsensor:
	4.3.	3	Gyroskop/Beschleunigungssensor:22
	4.3.4	4	Ultraschallsensor:23
	4.3.	5	4-fach RGB Sensor:24
5	Kom	plexe	re Aufgaben:
	5.1	Eins	atz von Variablen:
	5.2	Qua	drat:
	5.3	Rege	elmäßige Vielecke:
	5.4	Line	-Follower:
	5.5	Chai	näleon:
	5.6	Der	Roboter tanzt:
6	Abbi	ldung	sverzeichnis:



1 NOTWENDIGE SOFTWARE

Der Roboter kann über die Software mBlock programmiert werden:



Abbildung 1: Programmversionen

Hier existiert eine Programmversion für Windows und Mac. Alternativ kann auch direkt aus dem Browser (Chrome wird empfohlen) die Oberfläche gestartet werden. Für mobile Devices (Handy, Tablets) gibt es eine APP-Version für Android und für IOS.



Der mBot2 kommt mit folgenden Basiskomponenten:



Abbildung 2: mBot2 - Grundstruktur

Das Herzstück ist der CyperPi Mikrocontroller über den auch die Programmierung läuft. Zusätzliche Komponenten der Grundausstattung sind:

- der mBot2-Rahmen (zur Befestigung weiterer Komponenten)
- der Ultraschallsensor 2 (mit 2 blauen LEDs)



Abbildung 3: Ultraschallsensor



• der 4fach RGB Sensor (zur Farb- und Linienerkennung, Weiterentwicklung des 2fach Sensors bei mBot1)



Color LED 4x

Abbildung 4: RGB Sensoren

• Encoder-Motoren (präzisere Steuerung als beim Vorgängermodell)



Abbildung 5: Encoder-Motoren



1.1 DIE PROGRAMMOBERFLÄCHE

Die Programmoberfläche mBlock zeigt zunächst eine Scratch-Oberfläche mit den entsprechenden Komponenten (nach der Installation der notwendigen Erweiterungen)



Abbildung 6: Programmoberfläche

Neben der blockorientierten (Scratch-)Oberfläche kann der Roboter auch direkt mit Python programmiert werden:



1.2 BASISGERÄT UND ERWEITERUNGEN

Um nun den mBot2 (und dessen Basiskomponenten) aus der Programmieroberfläche entsprechend ansprechen zu können wird nun zunächst der Controller (CyperPi) als Gerät hinzugefügt:



Abbildung 8: CyperPi hinzufügen



Danach müssen noch die vorhandenen Sensoren/Motoren als Erweiterungen hinzugefügt werden:

keblock r	mBlock 🕲 🚞 Datei	🔑 Bearbeiten	Unbenannt	Speichern	Veröffentlichen	📀 Kurse	10 Tutorials	🖶 Rückmeldu
			spiele (Hi 🔹 bis fertig					
		Audio	spiele Hi 🔹					
	ē.	LED						
	9		Aufnahme starten					
		Anzeige	Aufnahme beenden					
		Bewegung:	spiele Aufzeichnung bis fertig					
			Aufnahme wiedergeben					
eräte	Figuren Hintergrund	Fühlen						
		Lan	spiele Ton 60 für 0.25 T	akte				
berPi	628		spiele Snare • für 0.25	Schläge				
Ð		AI						
ufüg	Verbinden Sie Ihr Gerät		audiogeschwindigkeit um					
	Wie wird das Gerät verwendet?		Audio-Geschwindigkeit auf	100 %				
		Ereignisse	Audio-Geschwindigkeit					
		Stauerung	Lautstärke um (10) % erhöh	en a a a a a				
			setze Lautstärke auf 30 %					
		Operatorer	■ Lautstärke(%)					
\mathbf{i}		Variablen	spiele Ton 700 Hz für 1	s i i i i i				
			spiele Ton 700 Hz					
		Meine Blöcke						
	Modus wechseln ③		stoppe alle Klänge					
	Hochladen Live							
	& Verbinden							

Abbildung 9:Erweiterungen hinzufügen

Aus den angebotenen Erweiterungen werden nun folgende installiert:



By mBlock official 📿 🛱 🌡 Mehr

• mBot2-Chassis (Motorensteuerung):



• Ultraschallsensor 2:





Quad RGB Sensor(beta) By mBlock official

• QuadRGB-Sensor:

2 VERBINDUNG HERSTELLEN

Die Verbindung zwischen der Programmieroberfläche (PC, Notebook, Tablett, ...) kann auf 3 Arten hergestellt werden: (Dabei wird noch grundsätzlich unterschieden, ob das Gerät im Livemodus oder

2.1 VERBINDUNG ÜBER USB-KABEL:



Abbildung 10: CyperPi über USB verbinden



USB	×	
		G
Alle anschlussfähigen Geräte an	zeigen	Gerät verbunden
СОМ5	×)	<u>Wie wird das Gerät</u> <u>verwendet?</u>
Verbindung		Modus wechseln ③ Hochladen Live
Abbildung 12: Schnittstelle auswählen	n	الله کې

Das Gerät ist nun verbunden. Über die Einstellungen kann nun der Hostname (wichtig für Bluetooth und WLAN Verbindungen) und die WLAN Einstellung vorgenommen werden:

Abbildung 11: USB Verbindung funktioniert

2.2 VERBINDUNG ÜBER BLUETOOTH:

Funktioniert für Mobilgeräte (oder am PC über speziellen Bluetooth-Adapter). Auf den Mobilgeräten (Tablets, iPads) sollte die App "makeblock / mBlock" installiert sein. Verbindung

Bluetooth einschalten:

*

iPad 🗢	09:42 @ * 1									
Settings	Bluetooth	n								
0 844	Bluetooth									
Apple ID, iCloud, iTunes & App Store	Now discoverable as "									
	MY DEVICES									
	AirPods	Not Connected 🚺								
	HOTGO H1	Not Connected (i)								
E Airplane Mode	OTHER DEVICES									
🛜 WLAN										
Rivetooth Oo	1 1.1 M									
	CONTRACTOR NO.									
Notifications	100									
Control Center	The end of the second sec									
C Do Not Disturb	- (10 M H H H H									
	8-04-									
General 1	0.000									
AA Display & Brightness										

Abbildung 13: Bluetooth am iPad aktivieren



In der App mBlock den Roboter (cyperPi) wählen:



Abbildung 14: CyperPi als Gerät auswöhlen

Erweiterungen (mBot2, Ultraschallsensor, QuadRGB Sensor) installieren:



Abbildung 15: Erweiterungen hinzufügen



CyperPi via Bluetooth verbinden:

<	Geräteserstellung	Das Gerät löschen
	CvberPi	
(& Verbinden	
		Hilfe zum Firmware-Update

Abbildung 16: mit Bluetooth verbinden

Verbindung herstellen:



Abbildung 17: Pairing



Verbindung hergestellt:

<	Geräteserstellung	Das Gerät löschen
	Bluetooth verbunden	
	Trennen Zurück zur Programmierung	
		Hilfe zum Firmware-Update

Abbildung 18: Bluetooth erfolgreich

3 INTERAKTIVE VERBINDUNG ("SPIELEMODUS"):

Alternativ zum Programmiermodus (mBlock-APP) kann über das Tablet der Roboter auch im "Spielemodus" (über die App makeblock) verbunden werden. Dabei wird das Tablet als Fernsteuerung für den Roboter verwendet:

Nach dem Starten der App kann über das Symbol das zu ve

das zu verbindende Gerät ausgewählt werden.





Abbildung 19: Spielemodus (Livemodus)

In unserem Fall wird hier der Roboter mBot2 ausgewählt.



Abbildung 20: mBot2 als Gerät



Wenn dieser nun (statt xLight) aufscheint, kann durch einen Klick auf das rote Bluetoothsymbol (¹) die Verbindungsaufnahme gestartet werden:



Abbildung 21: via Bluetooth verbinden

Das nun weiße Bluetoothsymbol zeigt den verbundenen Roboter.



Abbildung 22: aktive Verbindung

Durch Auswahl von "Spielen" => "Fahren" wird das Tablet nun zur Fernsteuerung:





Abbildung 23: Spielemodus - Fahren



Abbildung 24: iPad als Fernbedienung



4 EINFÜHRENDE AUFGABEN:

4.1 GRUNDLEGENDE AUFGABEN:

Start: Das Programm soll jeweils durch drücken auf den Button "A" (siehe Abbildung 25) gestartet werden.



Abbildung 25: Tasten am CyperPi

4.1.1 GERADLINIGE BEWEGUNGEN:

-990	moves for	ward 🔻	um	50	U/m	in fü	ir 🚺) Se	ekur	nden	Ľ	
		· · ·		50		ain						
-8530	moves for	ward •	um	50	, 0/n	nin						
wart	te 1 Seki	unde(n)										
	Stop-Motor	alle /	alles	•								
-	vorwärts 🔻	beweg	en 🤆	100	der	Zen	time	ter 🔻	b	is fe	rtig	

Abbildung 26: Befehle für geradlinige Bewegungen

Aufgabe 1: Versuche mit mehreren Befehlen den Roboter genau 25cm fahren zu lassen. Notiere dir die dazu passenden Daten (wie viele U/min – welche Zeit) – Teste die Genauigkeit mit längeren Strecken (1m, 2m)



	■B wenn Taste A ▼ gedrückt
-	
	₩ vorwärts bewegen ▼ mit 7 U/min
	warte 10 Sekunde(n)
	Encoder-Motor anhalten alle / alles ▼
	stopp alle / alles 🔻 👘 👘 👘 👘 👘 👘

Abbildung 27: Beispiellösung Aufgabe 1

Forschungsaufgaben: Liefern die Kombinationen 7U/min – 10s, 10U/min – 7s und 70U/min – 1s gleiche Ergebnisse? – Finde weitere Lösungen – Wie sieht es mit den Tests auf längere Distanzen aus? – Wie genau ist die eingebaute Funktion?

4.1.2 DREHUNGEN:

Drehungen können durchgeführt werden, indem z.B. nur ein Motor angesteuert wird. Versuche auf unterschiedliche Arten eine Drehung um 90° (rechts oder links) zu erzeugen. Was ist der Unterschied zur Bewegung beim Befehl "Nach links drehen 90° bis fertig"?

-1	🕮 Encode	er-Motor (li	nkes Rad	(EM1) 🔻	ື dreht f	ür 1	Seku	nden	um	50	rot	atior	nal s	peed	i (RF	PM)	-	
		ancod 🕮	er motor	linkes Ra	id (EM1) ▼	っ C	tates	at (5	0	rotat	ional	spee	ed (R	RPM)	•			
	🛞 Setze M	lotor M1-Lei	stung au	f <mark>50</mark> %,	Motor M2-	Leistur	ig auf	50) %									
		跪 Nach	links 🔻	drehen	90° bis fe	rtig												

Abbildung 28: Basisbefehle zur Erzeugung von Drehungen

Aufgabe 2:

- Erzeuge eine Linksdrehung um 90° nur durch Ansteuerung des rechten Motors. Welche Daten zur Motoransteuerung funktionieren bei dir? (Findest du ev. mehrere Möglichkeiten?)
- Erzeuge eine Rechtsdrehung um 90° nur durch Ansteuerung des linken Motors. Welche Daten zur Motoransteuerung funktionieren bei dir? (Findest du ev. mehrere Möglichkeiten?)
- Vergleiche die Bewegungen aus den ersten beiden Aufgaben mit der Bewegung bei "Nach links (bzw. rechts) drehen 90° bis fertig" – Wie könntest du diese Bewegung nachbauen?



						• •			• •							
									• •							• •
														-		
💶 wenn Taste 🛛 🔻	gedrück	t			1						:					
📸 Encoder-Motor	rechtes F	Rad (EM2)	•	5) mit	-	10	Dr	ehza	ahl	(U/n	nin)	•	dre	hen
warte 5 Sekunde(n)				•		•			-	•			-		
🐯 Encoder-Motor a	nhalten	alle	/ alle	es 🔻												
stopp alle / alles ▼						• •										

Abbildung 29: Lösung Teilaufgabe von 2

4.1.3 SIGNALE:

Der Roboter kann Töne aussenden und auch Informationen am Display ausgeben. Die Basisbefehle siehst du in nachfolgender Abbildung. Probiere sie einmal aus:





4.2 EINFACHE BEWEGUNGSAUFGABEN:

Programmiere den Roboter so, dass er einem einfachen Bewegungsmuster (vgl. Abbildung 31) folgen kann. Notiere dir deine jeweilige Lösung der Aufgabenstellung.

Zusatzaufgabe: Schaffst du es eventuelle, dass der Roboter beim Start und beim Ankommen im Ziel jeweils ein passendes Signal (Audio) aussendet und/oder auch etwas Passendes am CyperPi anzeigt?





Abbildung 31: Beispiel für eine Bewegungsaufgabe

Weitere Bewegungsmuster:



Abbildung 32: Vorschläge für weitere Aufgaben

4.3 VERWENDUNG DER SENSOREN DES MBOT2:

Der mBot2 verfügt über einige Sensoren, mit denen er die Umgebung wahrnehmen kann. Neben dem Ultraschall-Entfernungssensor und dem 4fach Lichtsensor gibt es im zentralen CyperPi auch schon integrierte Sensoren:





Abbildung 33: CyperPI – Funktionen

Hier findet man einen integrierten Lichtsensor, einen Schallsensor und einen Beschleunigungssensor, die entsprechend in Programmen auch angesprochen werden können. Die nachfolgenden Einführungsaufgaben sollen dir einige Grundfunktionen demonstrieren:

4.3.1 LICHTSENSOR:

Im Beispiel wird die Umgebungshelligkeit im Display angezeigt:



 wenn Taste A gedrückt wiederhole fortlaufend zeige Beschriftung 1 Intensität des Umgebungslichts an Center im Bildschirm 																							
u wenn Taste A ▼ gedrückt wiederhole fortlaufend	🖽 zeige Beschriftun	g 1 🔻	•••	Inte	nsitä	t des	s Um	geb	ungs	licht	an	C	ente	er im	Bilds	schi	rm s	d d	urch	mi	ittig	•	pixe
■ wenn Taste A gedrückt	wiederhole fortlaufend																						
	🖬 wenn Taste 🛛 🔻	gedrückt																					

Abbildung 34: Verwendung des Lichtsensors

4.3.2 SCHALLSENSOR:

Die Umgebungslautstärke soll am Display angezeigt werden:

wenn Taste	A 🔻	ged	rückt																		
zeige Besc	aufenc hriftur	ng 1	•	•	Lau	tstä	irke	an	Се	nter	im l	Bilds	chin	m 🔻	d	urcł	n (r	nitt	ig 🔻	pix	el
	4																				

Abbildung 35: Verwendung des Schallsensors

4.3.3 GYROSKOP/BESCHLEUNIGUNGSSENSOR:

Dient zum Anzeigen von Beschleunigungen bzw. der Lage (Neigung) des Roboters. Wenn der Roboter nach hinten kippt (oder bergauf fährt) wird am Display "UP!" angezeigt.

🛚 wenn Taste 🛛 🔻 geg	drückt												
ederhole fortlaufend													
The Mark Mark													
wenn 💵 Nach ninte	n geki	ppt 🗸		, dan	n								
T zoigo Roschriftung	1 -			Con	tor im	Bilde	obirm		dur	-b (e -	niv
🖬 zeige Beschriftung	1 🔻	UP	an	Cent	ter im	Bilds	chirm	•	dure	ch (gro	ß ▼	pix
💶 zeige Beschriftung	1 🔻	UP	an	Cent	ter im	Bilds	chirm	•	dure	ch (gro	B ▼	pix
■ zeige Beschriftung	1 •	UP	an	Cent	ter im	Bilds	chirm	•	dure	ch (gro	B ▼	pix
zeige Beschriftung warte Sekunde(n)	1 ▼ 2 - 2	UP	an	Cent	ter im	Bilds	chirm	•	dure	ch (gro	B ▼	pix
 zeige Beschriftung warte 1 Sekunde(n) 	1.	UP	an	Cent	ter im	Bilds	chirm	•) 	dure	ch (gro	β ▼	pix
 zeige Beschriftung warte 1 Sekunde(n) Bildschirminhalt löscl 	1 •	UP!	an	Cent	ter im	Bilds	chirm	•	dure	ch (gro	β ▼) pix
 zeige Beschriftung warte 1 Sekunde(n) Bildschirminhalt löscl 	1 ▼		an	Cent	ter im	Bilds	chirm 	•	dure - -	ch (gro	ß ▼) pix
 zeige Beschriftung warte 1 Sekunde(n) Bildschirminhalt löscl 	1 ▼ hen	UP!	an an a a a a	Cent	ter im	Bilds	chirm 	• • • • •	dure 	ch (gro	B▼) pix

Abbildung 36: Gyroskop

Kannst du das Programm so anpassen, dass auch ein Kippen nach unten, rechts oder links angezeigt wird?



R wann Tasta	nedrückt													
	jeurucki													
ederhole fortlaufend														
venn 💶 Nach hir	nten aekir	ppt 🔻 ?) da	ann										
					D.1					6	•			
Le zeige Beschriftu	ng 1 ▼	UP! a	n Ce	enter II	m Bil	dsch	irm v	<u> </u>	lurci	n g	roß	/ P	lixel	
venn 🤇 🖽 🛛 Nach vo	rne gekip	pt 🔹 🤶	🕨 , da	nn 👘										
zeige Beschriftung	na 1 🔻	DOWN	l) an	Cen	ter ir	n Bil	dschi	irm		lurcł	n ar	σβ		ixel
									-					_
				- A										
wenn 💵 Nach lin	кѕ декірр		, dan	n .										
💶 zeige Beschriftu	ng 1 🔻	LEFT!	an (Center	im E	Bildso	chirm	•	dur	ch (groß	•	pixe	el
wenn 🖽 Nach reg	chts aekir	ont 🔻 ?	da	ann										
	anto genip	pe · J ·	,											
💶 zeige Beschriftu	ng 1 🔻	RIGHT	an	Cent	er im	n Bild	lschir	m	d	urch	gro	oß ▼) pi	xel
zeige Beschriftung	ng 1 🔻	RIGHT	an	Cent	er in	n Bild	lschir	m s	d d	urch	gro	оВ▼) pi	xel
zeige Beschriftung warte 1 Sekunde(r	ng 1 🔹	RIGHT	an	Cent	er im	n Bild	lschir	m •	d d	urch	gro	oß ▼) pi	xel
varte 1 Sekunde(r	ng 1 💌	RIGHT	an A A	Cent	er in	n Bild	lschir	m v	• d	urch	gro	oß ▼	pi	xel
 zeige Beschriftun warte 1 Sekunde(r Bildschirminhalt lö 	ng 1 • n) oschen	RIGHT	an	Cent	er im	n Bild	lschir	m •	• d	urch	gro	oB 🔻	pi	xel

Abbildung 37: Lösung Zusatzaufgabe Gyroskop

4.3.4 ULTRASCHALLSENSOR:

Das nachfolgende Beispiel zeigt die Verwendung des Ultraschallsensors. Damit kann der Roboter (nach vorne) Hindernisse erkennen. Das Programm wird wieder mit der Taste "A" gestartet und läuft so lange, bis jemand auf die Taste "B" drückt. Falls der Sensor einen Gegenstand ortet, der näher als 10cm ist, dreht sich der Roboter um 90° nach links und fährt weiter.

Sobald die Taste "B" gedrückt wird, wird das Programm beendet.

- Was passiert, wenn du den Block "Encoder-Motor anhalten" entfernst?
- Kannst du in der Anzeige immer die aktuelle Entfernung zum nächsten Hindernis anzeigen lassen?



Abbildung 38: Ultraschall in der Tierwelt



■ wenn Taste A gedrückt								
wiederhole bis Taste B • gedrückt								
wenn 🔛 ultrasonic 2 1 🔹 distance	to an	obje	ect (cn	n) ·	< 10), c	lann	
🤐 Nach links 🔹 drehen 🤒 ° bis fe	rtig	,					,	
Image: Second secon								
	٦.							
🖀 Encoder-Motor anhalten 🛛 alle / alles 💌								
stopp alle / alles 🔹								

Abbildung 39: Ultraschallsensor (Entfernungsmessung)

4.3.5 4-FACH RGB SENSOR:

Er besteht aus 4 Sensoren die mit R2, R1, L1, L2 (Rechts oder Links) beschriftet sind und auch unter diesen Bezeichnungen angesprochen werden können:

		•											
Abl	bildung 40: RGB-Sensoren												
	💶 wenn Taste 🛛 🔻 gedrück	t											
	wiederhole bis	gedrückt?											
	wenn 🔛 quad rgb sensor	1 • probe (2)	R1 • detects so	hwarz 🔹 ?	, dann								
	🕮 moves forward ▼ un	n 50 U/min											
	sonst												
	🚓 Encoder-Motor anhalter	n alle / alles 🔻											
	💶 zeige Beschriftung 1 🔹	🔛 quad rgb se	nsor (1 🔹) probe	(2) R1 🔻 (detects (Farbe 🔻	an C	enter in	n Bildschi	rm 🔻	durch	groß 🔻	pixel
		و ا											
	🔛 quad rgb sensor 🚺 🔻 tur	ns off fill light											
	🚓 Encoder-Motor anhalten 🛛	alle / alles 🔹 📄											
	stopp alle / alles ▼	· · · · · · ·											

Abbildung 41: Beispiel mit RGB Sensoren

Das Beispiel (siehe Abbildung 41) zeigt eine Anwendung des RGB Sensors. Der Roboter fährt, solange der R1-Sensor die Farbe Schwarz sieht. Zusätzlich wird die aktuell festgestellte Farbe am Display angezeigt.



WIR GEBEN BILDUNG EIN ZUHAUSE



Abbildung 42: Testmuster für RGB-Aufgabe



5 KOMPLEXERE AUFGABEN:

5.1 EINSATZ VON VARIABLEN:

Durch einen Klick auf "Neue Variable" (im Block Variablen) kann ein Platzhalter erstellt werden. Im folgenden Beispiel wird eine Variable "Seitenlaenge" erzeugt und im Programm entsprechend verwendet. Die Bewegung des Roboters wird vom aktuellen Wert dieser Variablen abhängig gemacht. Was macht das nachfolgende Programm eigentlich?

Neue Variable		
Seitenlaenge	■ wenn Taste A ▼ gedrückt	
setze Seitenlaenge 🔹 auf 🕕	setze Seitenlaenge ▼ auf 1	
ändere Seitenlaenge 🔹 um 1	viederhole 36 vorwärts • bewegen Seitenlaenge der Zentimeter • bis ferti	
zeige Variable Seitenlaenge 🔹	and a second se	
Variable Seitenlaenge ▼ ausblenden	ändere Seitenlaenge ▼ um (1)	

Abbildung 43: Verwendung von Variablen

Kannst du das Programm so abändern, dass die aktuelle Seitenlänge am Display angezeigt wird?

5.2 QUADRAT:

Hinweis: In der Programmierung können auch Variable verwendet werden.

Programmiere den Roboter, dass er ein Quadrat mit 1m Seitenlänge abfährt. Achte auf entsprechende Präzision!

- Kannst du das Programm so abändern, dass die Seitenlänge des Quadrats 2m bzw. 50cm beträgt?
- Kannst du das Programm auch so ändern, dass eine Startseitenlänge von 100cm vorgegeben wird, diese aber durch den Joystick noch verändert werden kann (Anzeige zur Kontrolle) bevor der Roboter seine Bewegung startet?



🕞 wenn Taste 🛛 🔻 gedrückt																	
etze Seitenlaenge 🔹 auf 100																	
viederhole bis 💶 Taste 🛛 🔻 gedrückt?																	
wenn 💶 joystick ziehen † 🔹 ?), dann	1																
ändere Seitenlaenge ▼ um 1																	
wenn 💶 joystick ziehen‡ 🔹 ?), dann																	
	·																
ändere Seitenlaenge ▼ um -1																	
ändere Seitenlaenge ▼ um -1	-																
 ändere Seitenlaenge ▼ um -1 ∎ zeige Beschriftung 1 ▼ Seitenlaenge 	an	C	ente	r im B	lilds	chir	m •		lurc	h (gro	ов	•	pix	el		
ändere Seitenlaenge ▼ um -1	an	C	ente	r im B	lilds	chir	m •	• • c	lurc	h (gro	oß	•	pix	el		
ändere Seitenlaenge ▼ um -1 ■ zeige Beschriftung 1 ▼ Seitenlaenge J ederhole 4	an	C	ente	r im B	lilds	chir	n ▼ 	- - -	lurc	h ,	gro	оВ	•	pix	el		
ändere Seitenlaenge v um -1 ■ zeige Beschriftung 1 v Seitenlaenge Jederhole 4 worwärts v bewegen Seitenlaenge	an	Co Co Ze	ente	r im E eter v	ilds	chir Dis f	m • ertig		lurc	h (gre	οβ	•	pix	el		
ändere Seitenlaenge ▼ um -1 I zeige Beschriftung 1 ▼ Seitenlaenge J ederhole 4 I vorwärts ▼ bewegen Seitenlaenge I vorwärts ▼ bewegen Seitenlaenge	der	C C C Z Z Z Z Z Z Z Z Z Z Z Z Z Z	ente	r im B	ilds	chir ois f	m • ertiç		lurc	h (gro	oß		pix	el		
ändere Seitenlaenge ▼ um -1 ■ zeige Beschriftung 1 ▼ Seitenlaenge ∎ ederhole 4 1 1 ■ vorwärts ▼ bewegen Seitenlaenge ■ Nach links ▼ drehen 90 ° bis fertig) an der	C	ente	r im E eter s	ilds	chir Dis f	m • ertiç	, c	lurc	h (gro	oß		pix	el		
andere Seitenlaenge v um -1 ■ zeige Beschriftung 1 v Seitenlaenge iederhole 4 worwärts v bewegen Seitenlaenge Nach links v drehen 90 ° bis fertig	der	C	ente	r im E	iilds	chir bis f	m •		lurc 	h (gro	and		pix	el		
 ändere Seitenlaenge ▼ um -1 I zeige Beschriftung 1 ▼ Seitenlaenge viederhole 4 vorwärts ▼ bewegen Seitenlaenge Nach links ▼ drehen 90 ° bis fertig Encoder-Motor anhalten alle / alles ▼ 	der	C	ente	r im E	iilds	chir bis f	m • ertiç	, c	lurc	• h (• •	gro	oß		pix	el		
andere Seitenlaenge um -1 Presentation of the seitenlaenge Presentation of the seitenl	der	C Ze	ntim	eter v	ilds	chir bis f	m •		lurc	• • • • •	gro	a b b b b b b b b b b b b b b b b b b b		pix	el		

Abbildung 44: Lösung der Quadrataufgabe mit variabler Seitenlänge

5.3 REGELMÄßIGE VIELECKE:

Kannst du das Programm so abändern, dass statt des Quadrats ein gleichseitiges Dreieck, ein regelmäßiges 5-Eck bzw. ein regelmäßiges 6-Eck abgefahren wird? Wie bestimmst du jeweils den Winkel um den hier gedreht werden muss? Kannst du das für beliebige regelmäßige Vielecke verallgemeinern?

5.4 LINE-FOLLOWER:

Eine wichtige Aufgabe bei Transportrobotern in Betrieben ist es vorgegebenen Wegen zu folgen. Versuche anhand einfacher vorgegebener Wegstrecken, den Roboter so zu programmieren, dass er der schwarzen Linie folgt und, wenn er diese verliert, durch passende Drehungen wieder auf den Weg zurückfindet. Da unser Roboter sogar über 4 Sensoren verfügt, kann man daraus erkennen, in welche Richtung sich der Roboter drehen muss.

- Bei einer Abzweigung nach rechts wird der Sensor R2 die Richtungsänderung erkennen → Drehung nach rechts!
- Bei einer Abzweigung nach links wird der Sensor L2 die Richtungsänderung erkennen → Drehung nach links!







∎ wenn Taste B ▼ g	edrückt	· · ·	· · ·				Blöcke	
stopp andere Skripte de wiederhole fortlaufend	er Figur 🔹							
wenn 🔛 quad rgb s	sensor 1 🔹	L1, R1's	Linie 🔻	status is	(3) 11 🔻	? , da	ann a a a	
setze Links ▼ auf setze Rechts ▼ auf	15 a a						· · · ·	
wenn 🔛 quad rgb s	sensor 1 🔻	L1, R1's	Linie 🔻	status is	(1) 01 🔻	? , da		
setze Links ♥ auf							· · · ·	
wenn 🔛 quad rgb s	sensor 1 🔻	L1, R1's	Linie 🔻	status is	(2) 10 🔻	? , da	ann a s	
setze Links ▼ auf								
wenn 🔁 quad rgb	sensor 1 •	L1, R1's	Linie 🔻	status is	(0) 00 🔻	? , da		
setze Links ▼ auf setze Rechts ▼ au	-4 A A A	· · ·	· · ·	· · ·	· · ·	· · ·	· · · ·	
ancoder motor EM	1 🎝 rotates	at Links	RPM, e	encoder m	otor EM2	🤊 rotat	es at Rechts	RPI
bildung 45: LineFollower	-Hauptprogra	ımm						
stopp andere Skripte	der Figur 🔹			· · ·	· · ·	· · ·	· · · ·	
ancoder motor EM	1 🎝 rotates	at 0	RPM, enc	oder moto	or EM2 🎝	rotates	at 0 RPM	

💶 wenn Taste 🛛 🔹 gedrückt														
stopp andere Skripte der Figur	•													
🕮 encoder motor EM1 뜃 rota	tes at	0	RPN	И, en	code	er mot	tor E	M2	າ	rota	tes a	t 🕕	RP	М
Encoder-Motor anhalten all	e / all	les 🔻												
stopp alle / alles •														

Hinführung zur LineFollower-Lösung:

Erstelle 2 Variablen: Links und Rechts – sie beschreiben die "Antriebskraft" für die beiden Motoren. ٠

🕮 encoder motor EM1 为 rotates at 🕻 Links) RPM, encoder motor EM2 为 rotates at 🕻 Rechts) RPM



• Der Roboter soll fahren, solange er "auf der Linie" ist: (Achte auf die Antriebsrichtung der Motoren)



• Bei einer "Rechtskurve" sieht der rechte Sensor noch die Linie (=1), während die linke Seite nichts mehr sieht (=0). Daher muss der linke Motor sich drehen, während der rechte steht:



• Versuche nun dein Programm in Richtung des Line-Followers weiterzuentwickeln!

Beispiellösung:

💶 Bwenn Taste 🛛 🔻 gedrückt	
stopp andere Skripte der Figur 🕈	📕
wiederhole fortlaufend	
wenn QuadRGB-Sensor	1 ▼ L1, R1's linie ▼ im Status (3) 11 ▼ ? , dann
setze Links 🔻 auf 15	
setze Rechts auf -15	
wenn 🔛 QuadRGB-Sensor	1 ▼ L1, R1's linie ▼ im Status (1) 01 ▼ ? , dann
setze Links auf 4	
setze Rechts 🔻 auf 🕕	
wenn	1 ▼ L1, R1's linie ▼ im Status (2) 10 ▼ ? , dann
setze Links 🔻 auf 🕕	· · · · · · · · · · · · · · · · · · ·
setze Rechts ▼ auf -4	
wenn 😐 QuadRGB-Sensor	1 • 11 R1% linia • im Status (0)00 • 2 dann
setze Links 🔻 auf -4	
cotzo Pochte - ouf	
aul T	
🕮 encoder motor EM1 쓋 r	rotates at 🛛 Links 🛛 RPM, encoder motor EM2 🤚 rotates at 🛛 Rechts 🛛 RPM
۲. ۲. ۲. ۲. ۲. ۲. ۲. ۲. ۲. ۲. ۲. ۲. ۲. ۲	
The second	
ull∎ wenn laste A ▼ gedruckt	
stopp andere Skripte der Figur	
encoder motor EM1 🌖 ro	otates at 0 RPM, encoder motor EM2 *) rotates at 0 RPM
Encoder-Motor anhalten	alle / alles 🔻
stopp alle / alles 🔻	





5.5 CHAMÄLEON:

Hintergrundinformation: Jede Farbe kann durch Mischung der Grundfarben (Rot, Grün, Blau) erzeugt werden. Der RGB-Sensor des mBOT2 kann die einzelnen Farbanteile messen, die LEDs können durch Angabe des Farbanteils gefärbt werden.

Als Vorübung kann die LED mittels RGB-Anteiles in unterschiedlichen Farben gefärbt werden.

Aufgabenstellung: Mit dem RGB-Sensor soll die Untergrundfarbe bestimmt werden und die LEDs dann in dieser Farbe gefärbt werden (vgl: Chamäleon)



Abbildung 48: Chamäleon in der Natur



Abbildung 49: Prinzip der additiven Farbmischung



			ge	drück	t												
<u> </u>	uad-RG	B-Sens	sor (1	1 🔻)	schal	tet die E	Beleuc	htung au	5								
iederh	ole bis	< 🚥	Tast	te B	•	gedrück	t? 🔪										
setze	R 🔻	auf	Θ	Quad	-RGB	-Sensor	(1 🔻	Sonde	(2) R1	• •	erkennt	R-V	Vert	des	Obje	kts 🔻	
																	K
setze	G 🔻	auf	\square	Quad	-RGB	-Sensor	1 🔻	Sonde	(2) R1	•	erkenn	t G-\	Nert	des	Obje	kts 🔻	
																	Y
setze	В 🔻	auf	\square	Quad	-RGB	-Sensor	1 •	Sonde	(2) R1	•	erkennt	B-V	Vert	des	Obje	kts 🔻	
	\square		110														2
	zeige B	eschrif	ftung	1 🔻	R	an	ober	n links 🔻	durch	mit	tel 🔻	pixel					
												·					
	zeiae B	eschrif	ftuna	2 •		an	ober	mittia 🔹	durch	m	ittel 🔻	pixe	•				
	20.90 0											/ P					
	zeige B	oschrif	ftuna	3 -		an	oher	rechts	durch	m	ittel 💌	nive					
	zeige b	esenni	ung	<u> </u>			Obei			<u> </u>	itter v	Pine					
			Toio	+ D													
	LED a		zeigi				P .										
							٠										

Abbildung 50: Chamäleon - Beispiellösung

5.6 DER ROBOTER TANZT:

Über den Akustikteil kann der Roboter Musik erzeugen, über die LEDs Lichteffekte gestalten und über seine Motoren Bewegungen durchführen. Verwende diese Eigenschaften um einen kreativen Tanz zu gestalten. Dazu können Blöcke erzeugt werden, die einzelne Teile dieses Tanzes erzeugen.

Um übergreifend an einem Projekt arbeiten zu können, kann auf den iPADs folgendes Login (in App makeblock) verwendet werden:

Email: <u>dir@brg.at</u>

PW: bfbm4BRG!

Wenn zwei (oder mehr) Aktivitäten gleichzeitig durchgeführt werden sollen, müssen diese Blöcke durch das gleiche Ereignis gestartet werden (siehe Musterlösung):





Stadionstraße 1D A-2700 Wiener Neustadt



6 ABBILDUNGSVERZEICHNIS:

Abbildung 1: Programmversionen	3
Abbildung 2: mBot2 - Grundstruktur	4
Abbildung 3: Ultraschallsensor	4
Abbildung 4: RGB Sensoren	5
Abbildung 5: Encoder-Motoren	5
Abbildung 6: Programmoberfläche	6
Abbildung 7: Block- oder Pythonoberfläche?	7
Abbildung 8: CyperPi hinzufügen	7
Abbildung 9:Erweiterungen hinzufügen	8
Abbildung 10: CyperPi über USB verbinden	9
Abbildung 11: USB Verbindung funktioniert	10
Abbildung 12: Schnittstelle auswählen	10
Abbildung 13: Bluetooth am iPad aktivieren	10
Abbildung 14: CyperPi als Gerät auswöhlen	11
Abbildung 15: Erweiterungen hinzufügen	11
Abbildung 16: mit Bluetooth verbinden	12
Abbildung 17: Pairing	12
Abbildung 18: Bluetooth erfolgreich	13
Abbildung 19: Spielemodus (Livemodus)	14
Abbildung 20: mBot2 als Gerät	14
Abbildung 21: via Bluetooth verbinden	15
Abbildung 22: aktive Verbindung	15
Abbildung 23: Spielemodus - Fahren	16
Abbildung 24: iPad als Fernbedienung	16
Abbildung 25: Tasten am CyperPi	17
Abbildung 26: Befehle für geradlinige Bewegungen	17
Abbildung 27: Beispiellösung Aufgabe 1	18
Abbildung 28: Basisbefehle zur Erzeugung von Drehungen	18
Abbildung 29: Lösung Teilaufgabe von 2	19
Abbildung 30: Audio- und Bildschirmausgabe	19
Abbildung 31: Beispiel für eine Bewegungsaufgabe	20
Abbildung 32: Vorschläge für weitere Aufgaben	20
Abbildung 33: CyperPI – Funktionen	21
Abbildung 34: Verwendung des Lichtsensors	22
Abbildung 35: Verwendung des Schallsensors	22
Abbildung 36: Gyroskop	22
Abbildung 37: Lösung Zusatzaufgabe Gyroskop	23



Abbildung 38: Ultraschall in der Tierwelt	23
Abbildung 39: Ultraschallsensor (Entfernungsmessung)	24
Abbildung 40: RGB-Sensoren	24
Abbildung 41: Beispiel mit RGB Sensoren	24
Abbildung 42: Testmuster für RGB-Aufgabe	25
Abbildung 43: Verwendung von Variablen	26
Abbildung 44: Lösung der Quadrataufgabe mit variabler Seitenlänge	27
Abbildung 45: LineFollower-Hauptprogramm	28
Abbildung 46: LineFollower: Nebenprogramm für Ende	28
Abbildung 47: LineFollower: Beispiellösung	29
Abbildung 48: Chamäleon in der Natur	
Abbildung 49: Prinzip der additiven Farbmischung	
Abbildung 50: Chamäleon - Beispiellösung	31
Abbildung 51: Mustertanz	32