



















Die aktuellste Version des Webtools „EVA-ESP32“ kann unter folgendem Link heruntergeladen werden:

<http://hit-karlsruhe.de/hit-info/info-ws19/EVA-ESP32/webapp/>

Im Verzeichnis \blocks sind Javascript-Dateien hinterlegt, in denen alle Blöcke programmiert sind. Google stellt für die Erstellung solcher Blöcke bzw. Dateien ein intuitives Tool (<https://blockly-demo.appspot.com/static/demos/blockfactory/index.html>) bereit. Mit diesem Tool wurden auch alle Blöcke für EVA-ESP32 erstellt.

 _vti_cnf	23.10.2019 16:44	Dateiordner	
 basic_blocks_Arduino.js	02.10.2019 19:11	JavaSkriptdatei	3 KB
 colour.js	27.09.2019 14:25	JavaSkriptdatei	4 KB
 lists.js	27.09.2019 14:25	JavaSkriptdatei	29 KB
 logic.js	27.09.2019 14:25	JavaSkriptdatei	20 KB
 loops.js	27.09.2019 14:25	JavaSkriptdatei	11 KB
 MalRob_FG_display.js	22.10.2019 10:22	JavaSkriptdatei	2 KB
 MalRob_FG_led.js	03.10.2019 17:00	JavaSkriptdatei	1 KB
 MalRob_FG_limit_switch.js	03.10.2019 15:58	JavaSkriptdatei	1 KB
 MalRob_FG_motor.js	02.10.2019 16:51	JavaSkriptdatei	2 KB
 MalRob_FG_pen.js	03.10.2019 16:49	JavaSkriptdatei	1 KB
 MalRob_FG_start.js	23.10.2019 17:23	JavaSkriptdatei	1 KB
 math.js	27.09.2019 14:25	JavaSkriptdatei	16 KB
 new_block.js	22.10.2019 10:41	JavaSkriptdatei	1 KB
 procedures.js	27.09.2019 14:25	JavaSkriptdatei	40 KB
 text.js	27.09.2019 14:25	JavaSkriptdatei	27 KB
 variables.js	27.09.2019 14:25	JavaSkriptdatei	6 KB
 variables_dynamic.js	27.09.2019 14:25	JavaSkriptdatei	7 KB

Weiterhin erstellt das Tool weitere Dateien, die im Verzeichnis \generators abgelegt werden. Hier müssen in die Variable „code“ jeweils die Befehlszeilen eingetragen werden, die ein Block ausführen soll.

	_vti_cnf	23.10.2019 16:44	Dateiordner	
	dart	23.10.2019 16:44	Dateiordner	
	javascript	23.10.2019 16:44	Dateiordner	
	lua	23.10.2019 16:44	Dateiordner	
	php	23.10.2019 16:44	Dateiordner	
	python	23.10.2019 16:44	Dateiordner	
	basic_blocks_Arduino_syntax.js	27.09.2019 14:25	JavaSkriptdatei	2 KB
	dart.js	27.09.2019 14:25	JavaSkriptdatei	11 KB
	javascript.js	27.09.2019 14:25	JavaSkriptdatei	20 KB
	lua.js	27.09.2019 14:25	JavaSkriptdatei	8 KB
	MalRob_FG_display_syntax.js	22.10.2019 12:01	JavaSkriptdatei	1 KB
	MalRob_FG_led_syntax.js	03.10.2019 16:58	JavaSkriptdatei	1 KB
	MalRob_FG_limit_switch_syntax.js	22.10.2019 12:40	JavaSkriptdatei	1 KB
	MalRob_FG_motor_syntax.js	03.10.2019 16:14	JavaSkriptdatei	2 KB
	MalRob_FG_pen_syntax.js	03.10.2019 16:48	JavaSkriptdatei	1 KB
	MalRob_FG_start_syntax.js	23.10.2019 17:24	JavaSkriptdatei	2 KB
	new_block_syntax.js	23.10.2019 17:01	JavaSkriptdatei	2 KB
	php.js	27.09.2019 14:25	JavaSkriptdatei	12 KB
	python.js	27.09.2019 14:25	JavaSkriptdatei	13 KB

```

7  Blockly.JavaScript['motor_rotate'] = function(block) {
8      var dropdown_motor_selection = block.getFieldValue('motor_selection');
9      var dropdown_motor_direction = block.getFieldValue('motor_direction');
10     var number_motor_velocity = block.getFieldValue('motor_velocity');
11     var code = 'MOTORS.move_' + dropdown_motor_direction + '(' + dropdown_motor_selection + ', ' + number_motor_velocity + ');' + '\n';
12
13     return code;
14
15 };

```

Zum Schluss ist es noch erforderlich die Javascript-Dateien sowie die Blockbezeichnungen in der index.html einzubinden, damit EVA-ESP32 die Blöcke auch richtig verwenden kann.

```

<script src="blocks/MalRob_FG_motor.js"></script>
<script src="generators/MalRob_FG_motor_syntax.js"></script>

<category name="Motor" colour="#a3d95b">
<block type="motor_rotate"></block>
<block type="motor_stop"></block>
<block type="motor_revolution"></block>
</category>

```

Für den MalRob-FG wurde eine eigene Bibliothek entwickelt, um die Ansteuerung der einzelnen Komponenten in wenigen Befehlszeilen zu ermöglichen. Dies erleichtert auch Neueinsteigern das Verständnis für den Programmcode und macht den Gesamtcode übersichtlicher. So wird beispielsweise ein Motor mit folgender C-Code-Zeile aktiviert:

```
MOTORS.move_forward(MOTOR_LEFT, 50);
```

Der linke Motor bewegt sich also mit 50% der Maximalgeschwindigkeit nach vorn.

http://hit-karlsruhe.de/hit-info/info-ws19/EVA-ESP32/MalRob_FG.zip