

# OberBot

## Projekte die damit möglich sind

- Line following robot
- Kartographierung
- Hindernis Erkennung

## Software

Neben der Implementierung in C / C++ ist die Programmierung in MicroPython möglich. Am besten kapseln wir die ganze Hardware noch schön in ein Modul. Folgender Beispiel-Code lässt die Roboter gerade aus fahren:

```
import machine
import time

stepLeft = machine.Pin(0, machine.Pin.OUT)
stepRight = machine.Pin(12, machine.Pin.OUT)
dirLeft = machine.Pin(2, machine.Pin.OUT)
dirRight = machine.Pin(13, machine.Pin.OUT)
motorDisable = machine.Pin(14, machine.Pin.OUT)

motorDisable.off()
dirLeft.off()
dirRight.off()
while 1:
    stepLeft.on()
    stepRight.on()
    time.sleep_ms(1)
    stepLeft.off()
    stepRight.off()
    time.sleep_ms(1)
```

- [Oberbot Python Library](#)
- [MicroPython Doku](#)
- [WebREPL](#)

# IDE

<https://github.com/mu-editor/mu>

```
git clone https://github.com/mu-editor/mu.git
cd mu
pip3 pip install -r requirements.txt
python3 run.py
```

## Aktuelle Hardware Hypothese

Favorisierte Lösung: Wemos D1 mini mit steppern und A4988 Treibern. Alles Weitere wird, wenn möglich über I2C angeschlossen.

Kostenübersicht bei schneller Beschaffung:

Bauteil	Anzahl	Kosten	Gesamtkosten	Status
<a href="#">WEMOS D1 Mini</a>	1	4,10	4,10	lagernd
<a href="#">Boost Converter</a>	1	3,41	3,41	bestellt
<a href="#">A4988</a>	2	1,16	2,32	lagernd
<a href="#">stepper</a>	2		4,46	lagernd
<a href="#">port expander</a>	1	1,99	1,99	chips lagernd stangenware

Gesamtkosten knapp über 17 Euro. Und das noch ohne Spielerreien. Shopping in Europa macht keinen Spaß!!! ☹☹

Status: für Prototypen

Bei den Steppern ist der ULN2003 schon dabei! Eventuell Versuch mit port expander und uln2003

Pinout revision 0

d1 mini pro

reserved | RST TX | UART / programmer

| A0 RX | UART / programmer

reserved servo | D0 D1 | I2C SCL

motor enable | D5 D2 | I2C SDA

motor rechts step | D6 D3 | motor links step

motor rechts dir | D7 D4 | motor links dir

| D8 GND |

| 3V3 5V |

## I2C Bus

Adresse	Device
0x20	port expander 1
0x29	TOF
0x3C	Display
0x68	IMU

## Hardware Optionen

- Schrittmotoren: 28BYJ-48 mit
  - ULN2003 - Vorteil: geht mit 5 Volt Nachteil: 8 Pins Verbraucht
  - A4988 Driver Vorteil: Step/Dir Interface - dadurch nur 4 Pins. Einstellbarer Strom.  
Nachteil: operational >8V!. Der mit Stepup converter
- Controller: ESP8266 plain: Vorteil: Bauhöhe
  - WEMOS D1 Mini vorteilhaft, da programmer mit drauf.
  - Upgrade: ESP32 - eventuell notwendig wegen IO pins
- Line following hardware: IR Sensoren. Min 2, besser 4
- TOF sensor VL53L0X
- mpu6050
- OLED
- port expander

Kosten:

Bauteil	Anzahl	Kosten
mpu6050	1	0,66 \$
A4988	2	0,75 \$
28BYJ-48	2	1,25 \$
UL2003	2	0,40 \$
WEMOS D1 Mini	1	2,15 \$
VL53L0X	1	2.61 \$
Stepup Converter	1	0.34 \$
IR Sensoren	4	0,28 \$
OLED	1	1,95 \$
portexpander	1	0,80 \$

Gesamtkosten: 13,6 \$

Was kostet 3D Druck?!

# Supply Chain Management

Bauteil	Anzahl	Kosten pro Stück	Anmerkung	Status
<u>WEMOS d1 mini</u>	20	2.25	black friday + DHL eCommerce shipping	10 bestellt
<u>A4988</u>	40	0.80	black friday + DHL eCommerce shipping	26 bestellt
<u>stepup</u>	20	0.34		14 bestellt
<u>IR sensor</u>	+40	0.41	black friday + DHL eCommerce shipping	40 bestellt
<u>28BYJ-48 5v</u>	40	1.42	black friday + ePacket (20-25 tage)	17 bestellt
<u>oled</u>	20	1.94	black friday. uncooler versand. nur 10 stück ordern	10 bestellt
<u>dupont cable, f/f</u>	5x40	alle 3.3		200 bestellt
<u>motor connector</u>	1	1.13	für 50 stück!	50 bestellt
<u>pin header farbig</u>	1 mal je schwarz, rot, gelb, grün	2	preis für alle	4x400 bestellt

Bauteil	Anzahl	Kosten pro Stück	Anmerkung	Status
<u>dupont housing 4 pin</u>	1x100	1.69		100 bestellt
<u>dupont housing 2 pin</u>	1x100	1.21		100 bestellt
<u>batterie halter</u>	10	4.4	erst mal 10. muss erst getestet werden!	10 bestellt
<u>schalter</u>	10	1.75	erst mal 10. muss erst getestet werden!	10 bestellt
<u>taster</u>	2x10	1.24	erst mal 20. muss erst getestet werden!	//

PCB Fertiger

---

Version #1  
Erstellt: 20 Mai 2024 15:45:02 von Joel Hatsch  
Zuletzt aktualisiert: 20 Mai 2024 15:46:43 von Joel Hatsch