

Malroboter mit dem Raspberry Pi

Filmbüro Wismar

Richard Grünert und Josefine Richey

12/2020

Inhaltsverzeichnis

1 Kurzbeschreibung	2
1.1 Ziel	2
2 Vorbereitung	2
2.1 Materialien	2
2.1.1 Werkzeuge	2
2.1.2 Material	3
2.2 Software	4
2.2.1 Script	4
2.3 Motoren	4
3 Durchführung	5
3.1 Bau des Roboterarmes	5
3.2 Verbindung mit RaspberryPi	6
3.3 Programmierung	8
3.4 Programm A (Beispiel)	8
3.5 Programm B	8
4 Projektvariation	9
5 Problembehandlung	9
6 Anhang	10
6.1 Vorbereitung der Python-Programme	10
6.2 Anbringen der Motorkabel	12

1 Kurzbeschreibung

Diese Anleitung dient dem Aufbau sowie dem Einsatz eines einfachen Roboters in einem KURSKURSKURS. Bei diesem Roboter werden zwei Motoren mithilfe eines RASPBERRY-PI-Einplatinencomputers angesteuert, um einen Stift über ein Papierblatt zu bewegen. Die Geschwindigkeit beider Motoren kann dabei durch einfache Programmierbefehle eingestellt werden.

Die ursprüngliche Idee des Projektes stammt [hierher¹](#). Da kein CALLIOPE im Filbüro vorhanden war, wurde dieses Projekt auf einem RASPBERRY PI umgesetzt, was einige Zwischenschritte erforderte.

1.1 Ziel

Das Projekt soll den Kursteilnehmern die Möglichkeiten der Programmierung einfacher Elektronik nahebringen.

2 Vorbereitung

Je nach Durchführung sowie verfügbarer Zeit sollten einige Schritte des Aufbaus im Vornherein erledigt werden, z.B. die Verdrahtung der Motoren (später beschrieben).



Stifte



Klammer

2.1 Materialien

2.1.1 Werkzeuge

- Heißklebepistole
- Schere / Teppichmesser
- Schraubenzieher

¹<https://tuduu.org/projekt/automatischer-malroboter>



Stifte

s



Mango 2

Optional:

- Abisolierzange
- Lötkolben

2.1.2 Material

- RaspberryPi
- Bildschirm + HDMI-Kabel
- Maus + Tastatur
- Zahnstocher
- Motoren
- L298N Motortreiber
- Pappe
- Kabel
- Räder
- Klammern
- Unterlage (Telefonbuch / Karton)
- X Female-Female Jumperkabel
- 1x Female-Male Jumperkabel
- 12 V Netzteil und Adapter

2.2 Software

Auf dem RaspberryPi sollte bereits ein funktionierendes **Betriebssystem** installiert sein. Es bietet sich **Raspberry Pi OS** (Raspbian) an.

Bei dem ursprünglichen Projekt wurde die Programmierumgebung **Scratch** verwendet. Da diese die Ansteuerung von Motoren mit dem Raspberry Pi jedoch unnötig kompliziert macht, wurde hier auf die Verwendung der Programmiersprache **Python** ausgewichen. Python bietet zwar keine grafische Programmieroberfläche (wie Scratch), ist jedoch, nach Meinung der Autoren, eine für alle Altersgruppen intuitive und mindestens genauso zugängliche Programmiermethode wie Scratch.

Zur Ausführung der Python-Skripte eignet sich eine Entwicklungsumgebung, wie z.B. **Thonny** (vorinstalliert auf dem Pi).

Vorkenntnisse über Python sind nicht zwingend notwendig!

2.2.1 Script

Zur Vorbereitung sollten einige Funktionalitäten innerhalb von Python eingebaut werden, welche die Verwendung bei der Kursdurchführung für die Kursteilnehmer erleichtern.

Auf dem Raspberry Pi im Filmbüro sollten diese Dateien bereits vorhanden sein. Sind sie es nicht, führen Sie die notwendigen Schritte im Anhang am Ende der Dokumentation aus.

2.3 Motoren

Sollten die Motoren noch nicht mit Kabeln versehen sein, schauen Sie in den Anhang am Ende der Dokumentation. Dort finden Sie eine Anleitung.

Dieser Schritt lässt sich auch sehr gut mit den Kursteilnehmern durchführen!

3 Durchführung

3.1 Bau des Roboterarmes

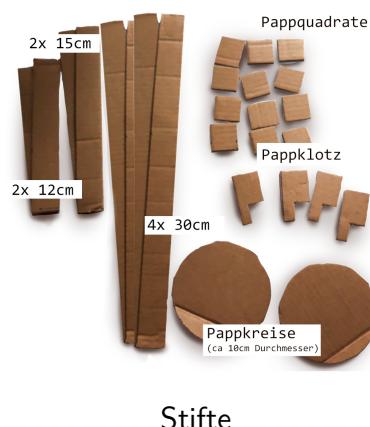
Schritt 1: Den Roboterarm vorbereiten

Die Pappe schneidet ihr in sechs 30cm lange und 2cm breite Streifen. Davon halbiert ihr zwei, sodass ihr vier 15cm lange Streifen habt. Zwei von diesen kürzt ihr nochmal auf 12cm.

Außerdem benötigt ihr vier Pappteile, die ihr wie auf dem Bild zuschneidet, zirka 12 kleine Pappquadrate und zwei Pappkreise, deren Durchmesser ungefähr mit der Größe der Reifen übereinstimmt.

Die Pappstreifen werden jetzt zusammengeklebt. Dafür klebt ihr jeweils zwei gleichlangen Streifen mit Heißkleber zusammen, sodass ihr zwei 30cm, ein 15cm und ein 12cm langes Pappstück habt.

Die vier Pappteile klebt ihr übereinander zu einem Klotz. An diesem können später die Stifte mit der Maulklemme befestigt werden.



Schritt 2: Den Roboterarm bauen

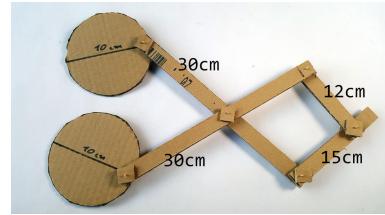
Die Pappteile müssen zu einem Arm zusammengebaut werden. Dabei bilden zwei kleine Pappquadrae und ein Zahnstocher ein Gelenk, das die Pappteile miteinander verbindet. Am besten orientiert ihr euch an der folgenden Grafik:

Falls ihr nicht weiterwisst, gibt es auch eine ausführlichere Beschreibung auf <https://tuduu.org/projekt/automatischer-malroboter>. Den Pappblock klebt ihr auf die Unterseite des Arms, also nicht die Fläche, auf welcher der zweite Pappstreifen aufliegt.

Wenn alles zusammengesteckt und verbunden ist, testet euren Arm vorsichtig.
Bewegt sich alles reibungslos?

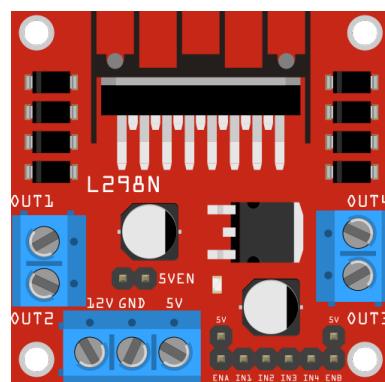
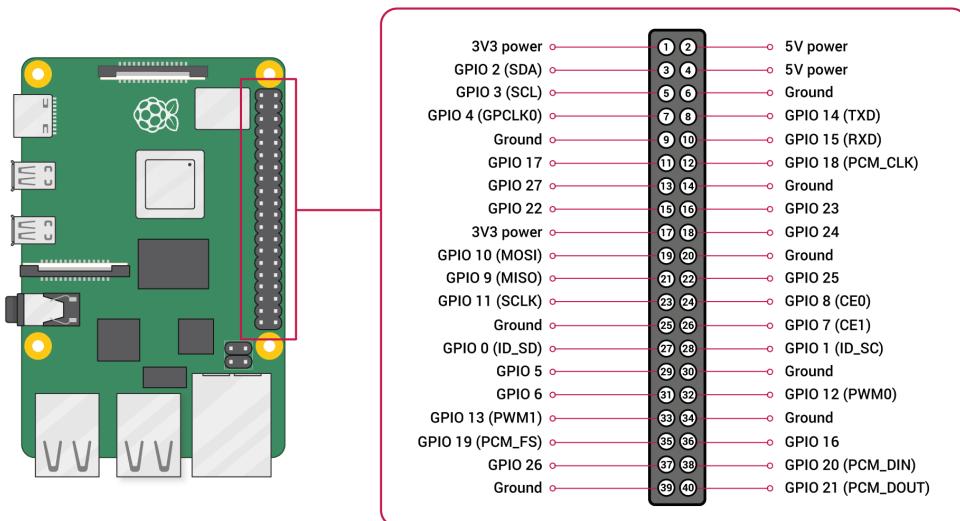
Dann verklebt die Zahnstocher zum Schluss oben und unten mit einem Tropfen Heißkleber, dann hältt alles ein bisschen besser.

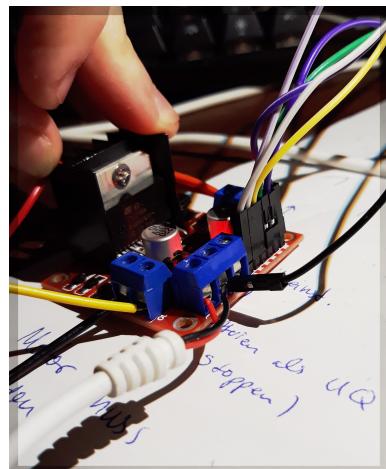
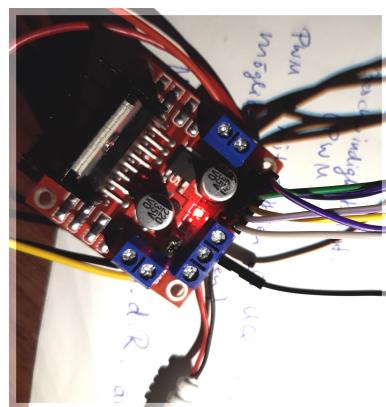
Die Pappkreise klebt ihr so auf die Reifen, dass die Pappstreifen oben auf liegen.



Der Roboterarm sollte sich reibungslos bewegen lassen

3.2 Verbindung mit RaspberryPi





Verbinden Sie den Raspberry Pi mit dem Motor sowie den Motor mit dem Treiber nach den Folgenden Tabellen. Für die Verbindungen an den Klemmen öffnen Sie diese mit einem Schraubenzieher, schieben das Kabel hinein und schrauben sie fest. Für die Verbindungen an den Pins verwenden Sie steckbare Jumperkabel (Female to Female).

Schritt 1: Verbindung der Motoren mit dem Treiber

Motortreiber L298 12V Spannungsquelle

OUT1	Motor1(+)
OUT2	Motor1(-)
OUT3	Motor2(+)
OUT4	Motor2(-)

Tabelle 1: Verbindungen zwischen dem Treiber und den Motoren. Die Polarität der Motoren spielt keine Rolle.

Schritt 2: Verbindung des Treibers mit dem Raspberry Pi

Die letzte Verbindung in der Tabelle sollte beachtet werden! Es bietet sich an,

Raspberry Pi GPIO	Motortreiber L298
GPIO 20	IN1
GPIO 16	IN2
GPIO 19	IN3
GPIO 4	IN4
GPIO 12	ENA
GPIO 13	ENB
Masse (GND) z.B. Pin 6	Masse (GND) - Klemme die selbe Klemme, an die auch (-) der Versorgungsspannung angeschlossen wird

Tabelle 2: Verbindungen zwischen dem Raspberry Pi und dem Motortreiber. Die Bezeichnungen des Treibers stehen auf der Board-Ober- oder Unterseite

sie mit einem Male-to-Female Jumperkabel zu verbinden, wie im Bild oben sichtbar.

Schritt 3: Verbindung der Versorgungsspannung an den Treiber

Für diese Verbindung eignet sich ein Adapter, der den Stecker des 12V Netzteiles auf zwei einfache Kabel führt, um diese an die Klemmen des Treibers anzuschließen.

Motortreiber L298	12V Spannungsquelle
+12V	+12V (i.d.R. rotes Kabel)
GND	GND (i.d.R. schwarzes Kabel)

Tabelle 3: Verbindungen zwischen Treiber und Netzteil

3.3 Programmierung

3.4 Programm A (Beispiel)

3.5 Programm B

4 Projektvariation

je nach Altersklasse

5 Problembehandlung

6 Anhang

6.1 Vorbereitung der Python-Programme

Führen Sie folgende Schritte durch, sofern die genannten Ordner und Dateien nicht bereits vorhanden sind:

Schritt 1: Legen Sie einen Projektordner an

Legen Sie auf dem Desktop des Raspberry Pi einen neuen Ordner an und nennen Sie ihn **Malroboter**.

Schritt 2: Erstellen Sie zwei neue Dateien

Erstellen Sie im Ordner Malroboter die folgenden Dateien:

- **motorsteuerung.py**: Diese Datei dient der Übersetzung der bestehenden Funktionen und der Definitionen innerhalb des Projektes und wird i.d.R. nicht von den Kursteilnehmern verwendet.
- **kurs.py**: Diese Datei soll dann von den Kursteilnehmern programmiert werden, hier wird z.B. die Geschwindigkeit der Motoren eingestellt.

Schritt 3: Fügen Sie das Steuerungsprogramm ein

Fügen Sie den folgenden Programmtext in die Datei **motorsteuerung.py** ein:
! Hinweis: Sie finden die Programmdateien auch [hier](#)².

motorsteuerung.py

```
import gpiodzero

motorA_IN1 = gpiodzero.DigitalOutputDevice(20)
motorA_IN2 = gpiodzero.DigitalOutputDevice(16)

motorB_IN1 = gpiodzero.DigitalOutputDevice(19)
motorB_IN2 = gpiodzero.DigitalOutputDevice(4)

motorA_EN = gpiodzero.PWMOutputDevice(12)
motorB_EN = gpiodzero.PWMOutputDevice(13)

motorA_EN.frequency = 10000
motorB_EN.frequency = 10000
```

²<https://www.github.com/latenighticecream/malroboter/Malroboter>

```

# Funktionen, um die Programmierung für die
# Kursteilnehmer zu erleichtern
# Die Geschwindigkeitsangaben sind in %
def motorA_an():
    motorA_EN.on()

def motorA_off():
    motorA_EN.off()

def motorA_rechts (geschwindigkeit = 10):
    motorA_EN.value = geschwindigkeit / 100
    motorA_IN1.on()
    motorA_IN2.off()

def motorA_links (geschwindigkeit = 10):
    motorA_EN.value = geschwindigkeit / 100
    motorA_IN1.off()
    motorA_IN2.on()

def motorB_an():
    motorB_EN.on()

def motorB_off():
    motorB_EN.off()

def motorB_rechts (geschwindigkeit = 10):
    motorB_EN.value = geschwindigkeit / 100
    motorB_IN1.on()
    motorB_IN2.off()

def motorB_links (geschwindigkeit = 10):
    motorB_EN.value = geschwindigkeit / 100
    motorB_IN1.off()
    motorB_IN2.on()

```

Schritt 4: Binden Sie das Steuerungsprogramm ein

Öffnen Sie die Datei ***kurs.py*** und fügen sie den folgenden Programmtext ein:

```
from motorsteuerung import *
```

Dies bindet die Definition innerhalb der Datei `motorsteuerung.py` in die Kursdatei ein, sodass sie dort verwendet werden können.

6.2 Anbringen der Motorkabel

Sollten die Motoren noch keine Kabel angebracht haben, muss das natürlich noch passieren. Dazu benötigen Sie pro Motor zwei einfache Kabel und eine Möglichkeit, deren Isolierung zu entfernen.

Schritt 1: Kabel abisolieren

Isolieren Sie die Kabel auf etwa 0.5 bis 1 cm ab.

Schritt 2: Kabel am Motor anbringen

Stecken Sie die Kabel jeweils durch eine der Ösen an den Motorkontakten und knicken diese um.

Schritt 3: (Optional) Kabel verlöten

Löten Sie die Kabel an den Motorkontakten an. Dieser Schritt ist optional, bietet jedoch einen stabileren und zuverlässigeren Kontakt.