Michael Putz

HTL Pinkafeld  Meierhofplatz 1

Ferngesteuertes Fahrzeug

Inhalt

[Vorwort 2](#_Toc160474100)

[Projektbeschreibung 2](#_Toc160474101)

[Vorbereitung Raspberry 3](#_Toc160474102)

[Zeitdokumentation 3](#_Toc160474103)

[Kostenaufstellung 3](#_Toc160474104)

[Schaltplan 4](#_Toc160474105)

[Quellcode Arduino 4](#_Toc160474106)

[Quellcode Python 6](#_Toc160474107)

[Quellcode NodeRed 8](#_Toc160474108)

[Literaturverzeichnis 54](#_Toc160474109)

[Abbildungsverzeichnis 54](#_Toc160474110)

[Tabellenverzeichnis 54](#_Toc160474111)

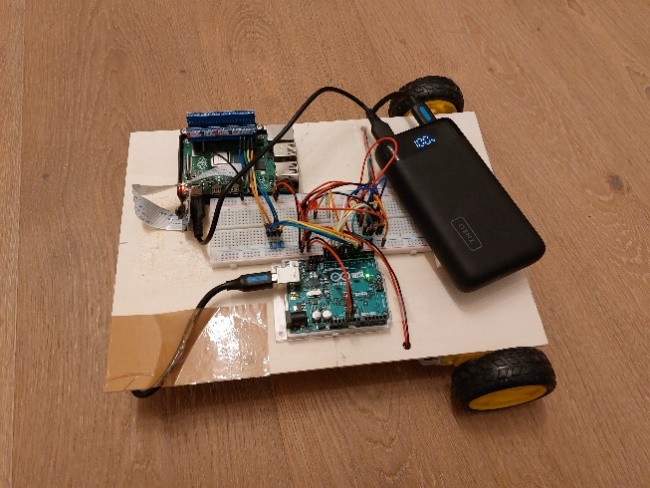


Abbildung 1 – Fahrzeug

(Putz, Eigenkreation, 2024)

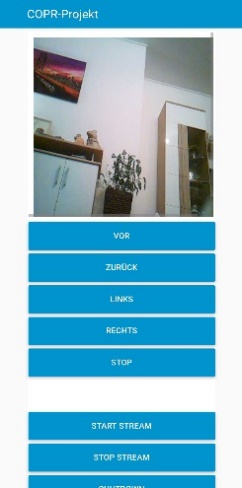


Abbildung 2 - User Interface

(Putz, Eigenkreation, 2024)

# Vorwort

Für mich war dieses Projekt der erste tiefgreifende Kontakt mit Raspberry und Arduino. Im Vordergrund stand die Machbarkeit und die Umsetzung des Projekts in einem Zeitumfang von rund 17 Arbeitsstunden. Ich wollte keine fertige Lösung verwenden, sondern das Fahrzeug komplett selbstständig umsetzen. Dafür habe ich diverse Internetseiten und Foren durchsucht.

Ich habe mich für die Umsetzung mit einem Raspberry in Verbindung mit einem Arduino entschieden, da ich beide Dinge verbinden wollte. Anfangs wollte ich die Verbindung über ein VPN ermöglichen, aus Zeitgründen habe ich mich allerdings für einen lokalen Hotspot entschieden.

Nach der Projektabgabe werde ich mich noch um die Optik bemühen, damit es nicht mehr aussieht wie ein Prototyp. Ich denke da an ein 3D gedrucktes Gehäuse.

Ziel dieser Dokumentation ist es, das andere Personen das Projekt vollständig nachbauen können, deswegen ist auch der gesamte Quellcode in diesem Dokument eingefügt.

# Projektbeschreibung

Ich kam während des Projekts an verschiedene schwierige Punkte, durch die ich allerdings auch eine Menge gelernt habe.

Zum Beispiel stellte sich die Verwendung von vier Antrieben für die Steuerung beziehungsweise für die Lenkung als nicht zielführend heraus, da die Motoren zu wenig Kraft hatten, um das Fahrzeug zu drehen. Deswegen habe ich mich entschieden, auf ein „Dreirad“ umzubauen.

Ebenso war die Platzierung der Powerbank anfangs nicht ideal und die Traktion des Fahrzeuges war dadurch negativ beeinflusst. Nach dem Umplatzieren über die Antriebe war die Traktion des Fahrzeuges deutlich besser.

Nachdem die Hardware funktioniert hatte, machte ich mich an die Softwareprogrammierung am Arduino sowie die Programmierung am Raspberry. Die Programmierung am Arduino ging problemlos, jedoch die Programmierung am Raspberry kostete etwas mehr Zeit, da ich mir hier erst Wissen aufbauen musste. Dadurch dauerte die Entwicklung der Raspberry-Software am längsten. Die Software wurde schließlich mit NodeRed als Framework erstellt.

Den Code für die Python-Kamera beziehungsweise den Livestream habe ich aus Zeitgründen aus dem Internet geladen und nach meinen Bedürfnissen angepasst.

# Vorbereitung Raspberry

Installiert wurde das Raspberry Betriebssystem mithilfe des „Raspberry Pi Imager“ in der 64bit Version und die Dienste SSH sowie VNC wurden gleich aktiviert. Anschließend wurde ein Paket-Update durchgeführt und NodeRed mit dem Befehl >bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)< installiert. Nach der Recherche für die Kamera wurde das Python-Script auf den Raspberry kopiert. Ein Test mit >python Cam.py< verlief erfolgreich, die Anpassung des Scripts auf meine Bedürfnisse wurde schnell erledigt. Danach machte ich mich an das Design in NodeRed und kümmerte mich um die Pin-Steuerung der Weboberfläche. Es wurde am Raspberry der WLAN-Accesspoint eingerichtet mit der SSID „MPP“ und auf Autostart gesetzt. Das Python-Script kann mit einem Klick auf den Button „Start Stream“ gestartet und bei Klick auf „Stop Stream“ gestoppt werden. Mit einem Klick auf „Shutdown“ kann der Raspberry heruntergefahren werden.

Zeitdokumentation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Datum | Beginn | Ende | Gesamt | Tätigkeitsbeschreibung |
| 03.03.2024 | 08:00 | 11:00 | 03:00 | Planung des Projekts in Thinkercad |
| 03.03.2024 | 11:00 | 12:00 | 01:00 | Verkabelung auf Breadboard |
| 03.03.2024 | 12:00 | 14:00 | 02:00 | Programmierung und Debugging des Arduino |
| 03.03.2024 | 15:00 | 16:00 | 01:00 | Umbau des Antriebes wegen nicht Steuerbarkeit |
| 03.03.2024 | 16:00 | 17:00 | 01:00 | Test des Fahrzeuges im Akkubetrieb |
| 03.03.2024 | 17:00 | 18:00 | 01:00 | Umbau des Fahrzeuges von Ultraschall auf Kamera |
| 03.03.2024 | 18:00 | 20:00 | 02:00 | Recherche für die Kamerasoftware in Python (Kopieren des Codes und Anpassung an die eigenen Bedürfnisse) |
| 04.03.2024 | 04:30 | 06:30 | 02:00 | Schreiben der Dokumentation |
| 04.03.2024 | 19:00 | 21:00 | 02:00 | Schreiben der Dokumentation |
|  |  |  | **15:00 / 17:00** | **Gesamt** |

Tabelle 1 - Arbeitszeit

# Kostenaufstellung

|  |  |
| --- | --- |
| Raspberry PI 4 4GB | € 71,59 |
| Arduino UNO | € 28,22 |
| Bausatzset | € 65,50 |
| Reifenset | € 12,10 |
| Powerbank | € 30,24 |
| Rolle (Frontreifen) | € 2,99 |
| Gehäuse für Raspberry | € 13,10 |
| **Gesamtkosten** | **€ 223,74** |

Tabelle 2 - Projektkosten

# Schaltplan

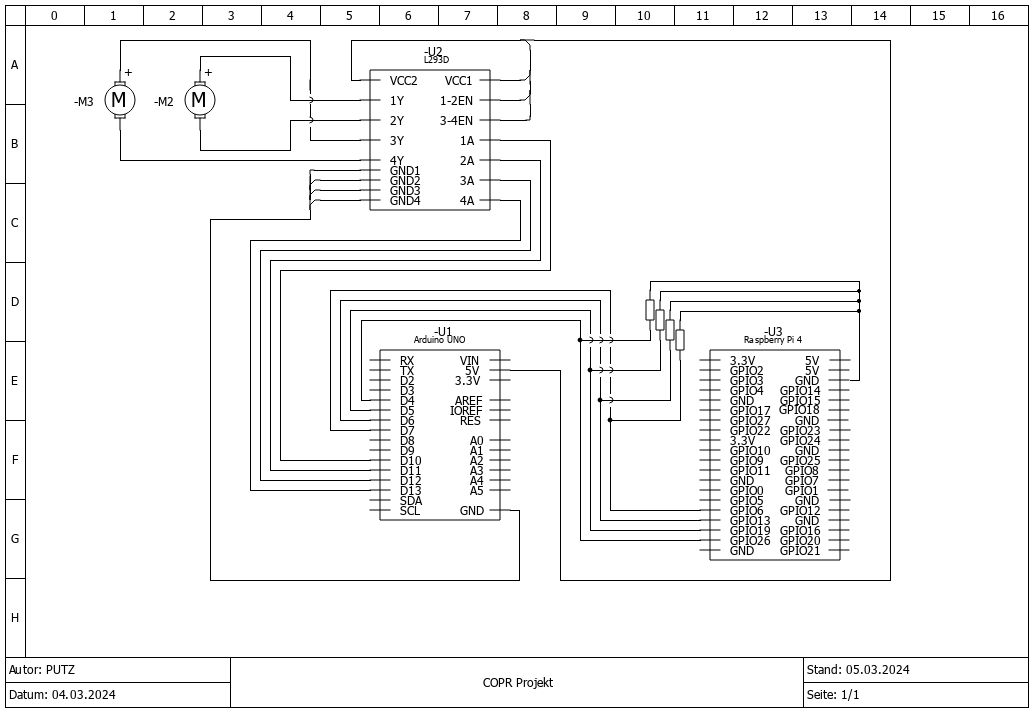


Abbildung 3 – Schaltplan

(Putz, Eigenkreation, 2024)

# Quellcode Arduino

(Putz, Github, 2024)

// C++ code

//

// Pins für die Antriebsrichtung

int linkerAntriebVor = 11;

int linkerAntriebZurueck = 12;

int rechterAntriebVor = 13;

int rechterAntriebZurueck = 10;

int zurueckInput = 4;

int vorInput = 5;

int rechtsInput = 6;

int linksInput = 7;

// Pins für die LED Anzeige

int VorPin = 8;

int ZurueckPin = 9;

void setup() {

  Serial.begin(9600);

  pinMode(VorPin, OUTPUT);

  pinMode(ZurueckPin, OUTPUT);

  pinMode(linkerAntriebVor, OUTPUT);

  pinMode(linkerAntriebZurueck, OUTPUT);

  pinMode(rechterAntriebVor, OUTPUT);

  pinMode(rechterAntriebZurueck, OUTPUT);

  pinMode(vorInput, INPUT);  //Orange - Vor

  pinMode(zurueckInput, INPUT);  //Gelb - Zurück

  pinMode(linksInput, INPUT);  //Grün - Links

  pinMode(rechtsInput, INPUT);  //Blau - Rechts

}

void loop() {

  if (digitalRead(vorInput)) {

    geradeVor();

  } else if (digitalRead(zurueckInput)) {

    geradeZurueck();

  } else if (digitalRead(linksInput)) {

    links();

  } else if (digitalRead(rechtsInput)) {

    rechts();

  } else {

    stop();

  }

}

void geradeVor() {

  digitalWrite(rechterAntriebVor, LOW);  // LINKS Zurück

  digitalWrite(linkerAntriebZurueck, HIGH);     // Rechts Vor

  digitalWrite(rechterAntriebZurueck, HIGH);       // LINKS Vor

  digitalWrite(linkerAntriebVor, LOW);  //Rechts Zurück

  digitalWrite(VorPin, HIGH);

  digitalWrite(ZurueckPin, LOW);

}

void geradeZurueck() {

  digitalWrite(rechterAntriebVor, HIGH);     // Rechts Vor

  digitalWrite(linkerAntriebVor, HIGH);       // LINKS Vor

  digitalWrite(linkerAntriebZurueck, LOW);  // LINKS Zurück

  digitalWrite(rechterAntriebZurueck, LOW);  //Rechts Zurück

  digitalWrite(VorPin, LOW);

  digitalWrite(ZurueckPin, HIGH);

}

void rechts() {

  digitalWrite(linkerAntriebZurueck, LOW);  // LINKS Zurück

  digitalWrite(rechterAntriebVor, LOW);     // Rechts Vor

  digitalWrite(linkerAntriebVor, HIGH);       // LINKS Vor

  digitalWrite(rechterAntriebZurueck, HIGH);  //Rechts Zurück

}

void links() {

  digitalWrite(linkerAntriebZurueck, HIGH);    // LINKS Zurück

  digitalWrite(rechterAntriebVor, HIGH);       // Rechts Vor

  digitalWrite(linkerAntriebVor, LOW);       // LINKS Vor

  digitalWrite(rechterAntriebZurueck, LOW);  //Rechts Zurück

}

void stop() {

  digitalWrite(linkerAntriebZurueck, LOW);  // LINKS Zurück

  digitalWrite(rechterAntriebVor, LOW);     // Rechts Vor

  digitalWrite(linkerAntriebVor, LOW);       // LINKS Vor

  digitalWrite(rechterAntriebZurueck, LOW);  //Rechts Zurück

  digitalWrite(VorPin, LOW);

  digitalWrite(ZurueckPin, LOW);

}

# Quellcode Python

(raspberrytips.com, 2024)

import io

import logging

import socketserver

from http import server

from threading import Condition

from picamera2 import Picamera2

from picamera2.encoders import JpegEncoder

from picamera2.outputs import FileOutput

# HTML page for the MJPEG streaming demo

PAGE = """\

<html>

<head>

<title></title>

</head>

<body>

<img src="stream.mjpg" width="300" height="300" />

</body>

</html>

"""

# Class to handle streaming output

class StreamingOutput(io.BufferedIOBase):

    def \_\_init\_\_(self):

        self.frame = None

        self.condition = Condition()

    def write(self, buf):

        with self.condition:

            self.frame = buf

            self.condition.notify\_all()

# Class to handle HTTP requests

class StreamingHandler(server.BaseHTTPRequestHandler):

    def do\_GET(self):

        if self.path == '/':

            # Redirect root path to index.html

            self.send\_response(301)

            self.send\_header('Location', '/index.html')

            self.end\_headers()

        elif self.path == '/index.html':

            # Serve the HTML page

            content = PAGE.encode('utf-8')

            self.send\_response(200)

            self.send\_header('Content-Type', 'text/html')

            self.send\_header('Content-Length', len(content))

            self.end\_headers()

            self.wfile.write(content)

        elif self.path == '/stream.mjpg':

            # Set up MJPEG streaming

            self.send\_response(200)

            self.send\_header('Age', 0)

            self.send\_header('Cache-Control', 'no-cache, private')

            self.send\_header('Pragma', 'no-cache')

            self.send\_header('Content-Type', 'multipart/x-mixed-replace; boundary=FRAME')

            self.end\_headers()

            try:

                while True:

                    with output.condition:

                        output.condition.wait()

                        frame = output.frame

                    self.wfile.write(b'--FRAME\r\n')

                    self.send\_header('Content-Type', 'image/jpeg')

                    self.send\_header('Content-Length', len(frame))

                    self.end\_headers()

                    self.wfile.write(frame)

                    self.wfile.write(b'\r\n')

            except Exception as e:

                logging.warning(

                    'Removed streaming client %s: %s',

                    self.client\_address, str(e))

        else:

            # Handle 404 Not Found

            self.send\_error(404)

            self.end\_headers()

# Class to handle streaming server

class StreamingServer(socketserver.ThreadingMixIn, server.HTTPServer):

    allow\_reuse\_address = True

    daemon\_threads = True

# Create Picamera2 instance and configure it

picam2 = Picamera2()

picam2.configure(picam2.create\_video\_configuration(main={"size": (640, 480)}))

output = StreamingOutput()

picam2.start\_recording(JpegEncoder(), FileOutput(output))

try:

    # Set up and start the streaming server

    address = ('', 8000)

    server = StreamingServer(address, StreamingHandler)

    server.serve\_forever()

finally:

    # Stop recording when the script is interrupted

    picam2.stop\_recording()

# Quellcode NodeRed

(Putz, Eigenkreation, 2024) mithilfe von (nodered.org, 2024)

[

{

"id": "58e93a0c492763f5",

"type": "tab",

"label": "Flow 1",

"disabled": false,

"info": "",

"env": []

},

{

"id": "94fa9e51ad8e8e8c",

"type": "ui\_tab",

"name": "COPR-Projekt",

"icon": "dashboard",

"disabled": false,

"hidden": true

},

{

"id": "1ff986957cdf9c73",

"type": "ui\_base",

"theme": {

"name": "theme-light",

"lightTheme": {

"default": "#0094CE",

"baseColor": "#0094CE",

"baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",

"edited": true,

"reset": false

},

"darkTheme": {

"default": "#097479",

"baseColor": "#097479",

"baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",

"edited": false

},

"customTheme": {

"name": "Untitled Theme 1",

"default": "#4B7930",

"baseColor": "#4B7930",

"baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"

},

"themeState": {

"base-color": {

"default": "#0094CE",

"value": "#0094CE",

"edited": false

},

"page-titlebar-backgroundColor": {

"value": "#0094CE",

"edited": false

},

"page-backgroundColor": {

"value": "#fafafa",

"edited": false

},

"page-sidebar-backgroundColor": {

"value": "#ffffff",

"edited": false

},

"group-textColor": {

"value": "#1bbfff",

"edited": false

},

"group-borderColor": {

"value": "#ffffff",

"edited": false

},

"group-backgroundColor": {

"value": "#ffffff",

"edited": false

},

"widget-textColor": {

"value": "#111111",

"edited": false

},

"widget-backgroundColor": {

"value": "#0094ce",

"edited": false

},

"widget-borderColor": {

"value": "#ffffff",

"edited": false

},

"base-font": {

"value": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"

}

},

"angularTheme": {

"primary": "indigo",

"accents": "blue",

"warn": "red",

"background": "grey",

"palette": "light"

}

},

"site": {

"name": "Node-RED Dashboard",

"hideToolbar": "false",

"allowSwipe": "false",

"lockMenu": "false",

"allowTempTheme": "true",

"dateFormat": "DD.MM.YYYY",

"sizes": {

"sx": 48,

"sy": 48,

"gx": 6,

"gy": 6,

"cx": 6,

"cy": 6,

"px": 0,

"py": 0

}

}

},

{

"id": "278458c5cf7fa1d3",

"type": "ui\_group",

"name": "Frame",

"tab": "94fa9e51ad8e8e8c",

"order": 1,

"disp": false,

"width": "6",

"collapse": false,

"className": ""

},

{

"id": "cca711a2924f7cfc",

"type": "ui\_group",

"name": "Kontroll",

"tab": "94fa9e51ad8e8e8c",

"order": 2,

"disp": false,

"width": "6",

"collapse": false,

"className": ""

},

{

"id": "b138b94747483c46",

"type": "ui\_iframe",

"z": "58e93a0c492763f5",

"group": "278458c5cf7fa1d3",

"name": "",

"order": 1,

"width": "6",

"height": "6",

"url": "http://10.42.0.1:8000",

"origin": "\*",

"scale": "110",

"x": 470,

"y": 260,

"wires": [

[

"885832ec151f4929",

"f042dfce0905cb7b",

"9b267c97b8449241",

"04b4e96b3b1ff9bc"

]

]

},

{

"id": "885832ec151f4929",

"type": "ui\_button",

"z": "58e93a0c492763f5",

"name": "",

"group": "278458c5cf7fa1d3",

"order": 2,

"width": 0,

"height": 0,

"passthru": false,

"label": "Vor",

"tooltip": "",

"color": "",

"bgcolor": "",

"className": "",

"icon": "",

"payload": "VOR",

"payloadType": "str",

"topic": "topic",

"topicType": "msg",

"x": 710,

"y": 200,

"wires": [

[

"ede01a8315b538f8"

]

]

},

{

"id": "f042dfce0905cb7b",

"type": "ui\_button",

"z": "58e93a0c492763f5",

"name": "",

"group": "278458c5cf7fa1d3",

"order": 3,

"width": 0,

"height": 0,

"passthru": false,

"label": "Zurück",

"tooltip": "",

"color": "",

"bgcolor": "",

"className": "",

"icon": "",

"payload": "ZURÜCK",

"payloadType": "str",

"topic": "topic",

"topicType": "msg",

"x": 710,

"y": 240,

"wires": [

[

"ede01a8315b538f8"

]

]

},

{

"id": "9b267c97b8449241",

"type": "ui\_button",

"z": "58e93a0c492763f5",

"name": "",

"group": "278458c5cf7fa1d3",

"order": 4,

"width": 0,

"height": 0,

"passthru": false,

"label": "Links",

"tooltip": "",

"color": "",

"bgcolor": "",

"className": "",

"icon": "",

"payload": "LINKS",

"payloadType": "str",

"topic": "topic",

"topicType": "msg",

"x": 710,

"y": 280,

"wires": [

[

"ede01a8315b538f8"

]

]

},

{

"id": "04b4e96b3b1ff9bc",

"type": "ui\_button",

"z": "58e93a0c492763f5",

"name": "",

"group": "278458c5cf7fa1d3",

"order": 5,

"width": 0,

"height": 0,

"passthru": false,

"label": "Rechts",

"tooltip": "",

"color": "",

"bgcolor": "",

"className": "",

"icon": "",

"payload": "RECHTS",

"payloadType": "str",

"topic": "topic",

"topicType": "msg",

"x": 720,

"y": 320,

"wires": [

[

"ede01a8315b538f8"

]

]

},

{

"id": "38e21db1f1deae3e",

"type": "rpi-gpio out",

"z": "58e93a0c492763f5",

"name": "",

"pin": "6",

"set": true,

"level": "0",

"freq": "",

"out": "out",

"bcm": true,

"x": 2260,

"y": 140,

"wires": []

},

{

"id": "ede01a8315b538f8",

"type": "switch",

"z": "58e93a0c492763f5",

"name": "",

"property": "payload",

"propertyType": "msg",

"rules": [

{

"t": "eq",

"v": "LINKS",

"vt": "str"

},

{

"t": "eq",

"v": "RECHTS",

"vt": "str"

},

{

"t": "eq",

"v": "VOR",

"vt": "str"

},

{

"t": "eq",

"v": "ZURÜCK",

"vt": "str"

}

],

"checkall": "false",

"repair": false,

"outputs": 4,

"x": 990,

"y": 220,

"wires": [

[

"41794da1dda23483",

"55faa8f05ace8e3a"

],

[

"41794da1dda23483",

"86a4ebcd78e1f4e0"

],

[

"41794da1dda23483",

"72f78b2e8742816d"

],

[

"41794da1dda23483",

"f7b12adfaee02527"

]

]

},

{

"id": "3cc4cc2c1f6e110c",

"type": "rpi-gpio out",

"z": "58e93a0c492763f5",

"name": "",

"pin": "13",

"set": true,

"level": "0",

"freq": "",

"out": "out",

"bcm": true,

"x": 2260,

"y": 320,

"wires": []

},

{

"id": "e8420e242f82d5e0",

"type": "rpi-gpio out",

"z": "58e93a0c492763f5",

"name": "",

"pin": "19",

"set": true,

"level": "0",

"freq": "",

"out": "out",

"bcm": true,

"x": 2260,

"y": 500,

"wires": []

},

{

"id": "257073b84f0aaf69",

"type": "rpi-gpio out",

"z": "58e93a0c492763f5",

"name": "",

"pin": "26",

"set": true,

"level": "0",

"freq": "",

"out": "out",

"bcm": true,

"x": 2260,

"y": 660,

"wires": []

},

{

"id": "55faa8f05ace8e3a",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "true",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1430,

"y": 180,

"wires": [

[

"109932eca160ff7c"

]

]

},

{

"id": "86a4ebcd78e1f4e0",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "true",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1430,

"y": 220,

"wires": [

[

"eb0a358e63d7c3ca"

]

]

},

{

"id": "72f78b2e8742816d",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "true",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1430,

"y": 260,

"wires": [

[

"1f5e66089745a5c6"

]

]

},

{

"id": "f7b12adfaee02527",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "true",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1430,

"y": 300,

"wires": [

[

"ce67ec3aac929346"

]

]

},

{

"id": "109932eca160ff7c",

"type": "switch",

"z": "58e93a0c492763f5",

"name": "",

"property": "payload",

"propertyType": "msg",

"rules": [

{

"t": "true"

},

{

"t": "true"

},

{

"t": "true"

},

{

"t": "true"

}

],

"checkall": "true",

"repair": false,

"outputs": 4,

"x": 1670,

"y": 160,

"wires": [

[

"38e21db1f1deae3e"

],

[

"3a7b57268775cbf5"

],

[

"b6d7dcae16271aea"

],

[

"b82c812fa01a8e09"

]

]

},

{

"id": "3a7b57268775cbf5",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "false",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1970,

"y": 200,

"wires": [

[

"3cc4cc2c1f6e110c"

]

]

},

{

"id": "b6d7dcae16271aea",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "false",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1970,

"y": 240,

"wires": [

[

"e8420e242f82d5e0"

]

]

},

{

"id": "b82c812fa01a8e09",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "false",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1970,

"y": 280,

"wires": [

[

"257073b84f0aaf69"

]

]

},

{

"id": "eb0a358e63d7c3ca",

"type": "switch",

"z": "58e93a0c492763f5",

"name": "",

"property": "payload",

"propertyType": "msg",

"rules": [

{

"t": "true"

},

{

"t": "true"

},

{

"t": "true"

},

{

"t": "true"

}

],

"checkall": "true",

"repair": false,

"outputs": 4,

"x": 1670,

"y": 340,

"wires": [

[

"3cc4cc2c1f6e110c"

],

[

"b9209f708ee65e7f"

],

[

"311a18b8af1e68fa"

],

[

"9294450d283a9189"

]

]

},

{

"id": "1f5e66089745a5c6",

"type": "switch",

"z": "58e93a0c492763f5",

"name": "",

"property": "payload",

"propertyType": "msg",

"rules": [

{

"t": "true"

},

{

"t": "true"

},

{

"t": "true"

},

{

"t": "true"

}

],

"checkall": "true",

"repair": false,

"outputs": 4,

"x": 1670,

"y": 520,

"wires": [

[

"e8420e242f82d5e0"

],

[

"a5c06cd162da7582"

],

[

"4b0dff2ce6458ccb"

],

[

"ac66737e707e23f8"

]

]

},

{

"id": "ce67ec3aac929346",

"type": "switch",

"z": "58e93a0c492763f5",

"name": "",

"property": "payload",

"propertyType": "msg",

"rules": [

{

"t": "true"

},

{

"t": "true"

},

{

"t": "true"

},

{

"t": "true"

}

],

"checkall": "true",

"repair": false,

"outputs": 4,

"x": 1670,

"y": 680,

"wires": [

[

"257073b84f0aaf69"

],

[

"8b07c903687e5267"

],

[

"82e8f9ce375e70bc"

],

[

"8b318c5ab2e37eb9"

]

]

},

{

"id": "b9209f708ee65e7f",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "false",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1970,

"y": 360,

"wires": [

[

"38e21db1f1deae3e"

]

]

},

{

"id": "311a18b8af1e68fa",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "false",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1970,

"y": 400,

"wires": [

[

"e8420e242f82d5e0"

]

]

},

{

"id": "9294450d283a9189",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "false",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1970,

"y": 440,

"wires": [

[

"257073b84f0aaf69"

]

]

},

{

"id": "a5c06cd162da7582",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "false",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1970,

"y": 540,

"wires": [

[

"38e21db1f1deae3e"

]

]

},

{

"id": "4b0dff2ce6458ccb",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "false",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1970,

"y": 580,

"wires": [

[

"3cc4cc2c1f6e110c"

]

]

},

{

"id": "ac66737e707e23f8",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "false",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1970,

"y": 620,

"wires": [

[

"257073b84f0aaf69"

]

]

},

{

"id": "8b07c903687e5267",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "false",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1970,

"y": 720,

"wires": [

[

"38e21db1f1deae3e"

]

]

},

{

"id": "82e8f9ce375e70bc",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "false",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1970,

"y": 760,

"wires": [

[

"3cc4cc2c1f6e110c"

]

]

},

{

"id": "8b318c5ab2e37eb9",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "false",

"tot": "bool"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1970,

"y": 800,

"wires": [

[

"e8420e242f82d5e0"

]

]

},

{

"id": "c343789afa0c08cf",

"type": "ui\_button",

"z": "58e93a0c492763f5",

"name": "",

"group": "278458c5cf7fa1d3",

"order": 6,

"width": 0,

"height": 0,

"passthru": false,

"label": "Stop",

"tooltip": "",

"color": "",

"bgcolor": "",

"className": "",

"icon": "",

"payload": "false",

"payloadType": "bool",

"topic": "topic",

"topicType": "msg",

"x": 2010,

"y": 80,

"wires": [

[

"38e21db1f1deae3e",

"3cc4cc2c1f6e110c",

"e8420e242f82d5e0",

"257073b84f0aaf69",

"a1cac76d9f2942a8"

]

]

},

{

"id": "41794da1dda23483",

"type": "file",

"z": "58e93a0c492763f5",

"name": "",

"filename": "/tmp/move",

"filenameType": "str",

"appendNewline": true,

"createDir": true,

"overwriteFile": "true",

"encoding": "none",

"x": 1010,

"y": 420,

"wires": [

[

"349995dd04c83cf2"

]

]

},

{

"id": "349995dd04c83cf2",

"type": "file in",

"z": "58e93a0c492763f5",

"name": "",

"filename": "/tmp/move",

"filenameType": "str",

"format": "utf8",

"chunk": false,

"sendError": false,

"encoding": "none",

"allProps": false,

"x": 1010,

"y": 520,

"wires": [

[

"931afe48d1a5766b"

]

]

},

{

"id": "931afe48d1a5766b",

"type": "ui\_text",

"z": "58e93a0c492763f5",

"group": "278458c5cf7fa1d3",

"order": 7,

"width": 0,

"height": 0,

"name": "",

"label": "",

"format": "{{msg.payload}}",

"layout": "col-center",

"className": "",

"style": false,

"font": "",

"fontSize": 16,

"color": "#000000",

"x": 1210,

"y": 520,

"wires": []

},

{

"id": "de4e265ca1e4efaf",

"type": "file",

"z": "58e93a0c492763f5",

"name": "",

"filename": "/tmp/move",

"filenameType": "str",

"appendNewline": true,

"createDir": true,

"overwriteFile": "true",

"encoding": "none",

"x": 1270,

"y": 420,

"wires": [

[

"349995dd04c83cf2"

]

]

},

{

"id": "a1cac76d9f2942a8",

"type": "change",

"z": "58e93a0c492763f5",

"name": "",

"rules": [

{

"t": "set",

"p": "payload",

"pt": "msg",

"to": "STOP",

"tot": "str"

}

],

"action": "",

"property": "",

"from": "",

"to": "",

"reg": false,

"x": 1290,

"y": 360,

"wires": [

[

"de4e265ca1e4efaf"

]

]

},

{

"id": "d207a76012c3becf",

"type": "ui\_button",

"z": "58e93a0c492763f5",

"name": "",

"group": "278458c5cf7fa1d3",

"order": 8,

"width": 0,

"height": 0,

"passthru": false,

"label": "Start Stream",

"tooltip": "",

"color": "",

"bgcolor": "",

"className": "",

"icon": "",

"payload": "true",

"payloadType": "bool",

"topic": "topic",

"topicType": "msg",

"x": 170,

"y": 260,

"wires": [

[

"fd62ed470cbb648c"

]

]

},

{

"id": "fd62ed470cbb648c",

"type": "exec",

"z": "58e93a0c492763f5",

"command": "python /home/michael/Desktop/Cam.py &",

"addpay": "",

"append": "",

"useSpawn": "false",

"timer": "",

"winHide": false,

"oldrc": false,

"name": "",

"x": 320,

"y": 320,

"wires": [

[

"b138b94747483c46"

],

[],

[]

]

},

{

"id": "18935474feee185f",

"type": "ui\_button",

"z": "58e93a0c492763f5",

"name": "",

"group": "278458c5cf7fa1d3",

"order": 8,

"width": 0,

"height": 0,

"passthru": false,

"label": "Stop Stream",

"tooltip": "",

"color": "",

"bgcolor": "",

"className": "",

"icon": "",

"payload": "",

"payloadType": "str",

"topic": "topic",

"topicType": "msg",

"x": 170,

"y": 400,

"wires": [

[

"7bad1e4d1526414d"

]

]

},

{

"id": "7bad1e4d1526414d",

"type": "exec",

"z": "58e93a0c492763f5",

"command": "pkill python",

"addpay": "",

"append": "",

"useSpawn": "false",

"timer": "",

"winHide": false,

"oldrc": false,

"name": "",

"x": 390,

"y": 400,

"wires": [

[],

[],

[]

]

},

{

"id": "bec790a77d3340a4",

"type": "ui\_button",

"z": "58e93a0c492763f5",

"name": "",

"group": "278458c5cf7fa1d3",

"order": 9,

"width": 0,

"height": 0,

"passthru": false,

"label": "Shutdown",

"tooltip": "",

"color": "",

"bgcolor": "",

"className": "",

"icon": "",

"payload": "",

"payloadType": "str",

"topic": "topic",

"topicType": "msg",

"x": 160,

"y": 480,

"wires": [

[

"89755d714226d252"

]

]

},

{

"id": "89755d714226d252",

"type": "exec",

"z": "58e93a0c492763f5",

"command": "sudo shutdown now",

"addpay": "",

"append": "",

"useSpawn": "false",

"timer": "",

"winHide": false,

"oldrc": false,

"name": "",

"x": 420,

"y": 480,

"wires": [

[

"0cbe754dc1850e22"

],

[

"0cbe754dc1850e22"

],

[

"0cbe754dc1850e22"

]

]

},

{

"id": "0cbe754dc1850e22",

"type": "debug",

"z": "58e93a0c492763f5",

"name": "debug 1",

"active": true,

"tosidebar": true,

"console": false,

"tostatus": false,

"complete": "false",

"statusVal": "",

"statusType": "auto",

"x": 620,

"y": 480,

"wires": []

},

{

"id": "0ac64139a6ab29f4",

"type": "comment",

"z": "58e93a0c492763f5",

"name": "Blau - Links > 7",

"info": "",

"x": 2280,

"y": 100,

"wires": []

},

{

"id": "2aad57dd525d7e18",

"type": "comment",

"z": "58e93a0c492763f5",

"name": "Grün - Rechts > 6",

"info": "",

"x": 2290,

"y": 280,

"wires": []

},

{

"id": "c71c1af32e80ed63",

"type": "comment",

"z": "58e93a0c492763f5",

"name": "Gelb - Vor > 5",

"info": "",

"x": 2270,

"y": 460,

"wires": []

},

{

"id": "92de297ffde5a09a",

"type": "comment",

"z": "58e93a0c492763f5",

"name": "Orange Zurück > 4",

"info": "",

"x": 2290,

"y": 620,

"wires": []

}

]

# Literaturverzeichnis

*nodered.org*. (2024). Abgerufen am 03. 03 2024 von nodered.org: https://nodered.org/docs/

Putz, M. (03. 03 2024). Eigenkreation.

Putz, M. (08. 06 2024). *Github*. Von Github: https://github.com/Dronei/COPR.git abgerufen

*raspberrytips.com*. (2024). Abgerufen am 03. 03 2024 von https://raspberrytips.com/how-to-live-stream-pi-camera/

# Abbildungsverzeichnis

[Abbildung 1 – Fahrzeug 1](#_Toc160473791)

[Abbildung 2 - User Interface 1](#_Toc160473792)

[Abbildung 3 – Schaltplan 4](#_Toc160473793)

# Tabellenverzeichnis

[Tabelle 1 - Arbeitszeit 3](#_Toc160473794)

[Tabelle 2 - Projektkosten 3](#_Toc160473795)