

Mobile Rückfahrkamera

Projektübersicht

Gegenwärtig werden bereits sehr viele Fahrzeuge mit integrierten Rückfahrkameras produziert und erfolgreich verkauft. Hat man einst die Gelegenheit genutzt, per Rückfahrkamera das Fahrzeug in eine Parklücke zu manövrieren, möchte man nur ungern auf diese Technik verzichten.

Aus diesem Ansatz entstand diese Projektarbeit, welche sich mit der Konfiguration einer Rückfahrkamera, bestehend aus einem Raspberry Pi Zero W in Kombination mit einer 160° Weitwinkelkamera, beschäftigt.

Funktionsweise

Der Raspberry Pi stellt mit einer W-Lan Komponente ein lokales Netzwerk zur Verfügung, auf diesem die aufgenommen Bilder in Echtzeit gestreamt werden. Der Zugriff auf den Stream wird mittels Webbrowser über die IP-Adresse: 192.172.13.2:8080 erlangt. Dadurch können mobile wie stationäre Endgeräte auf die aufgenommenen Bilder zugreifen. Alternativ ist der Stream auch über eine Android-App (siehe Kapitel „

Verbindung mit dem Netzwerk

Kurz nach dem Startvorgang des Raspberry Pi ist das W-Lan Netzwerk bereits aktiv. Wurde der Netzwerkname nicht umbenannt (Möglichkeit zur Umbenennung siehe Kapitel „Mobile App“) ist das W-Lan unter dem Namen RFKNET via Computer oder Handy zu finden. Das zu Beginn festgelegte Passwort lautet „hallohallo“.

Mobile App“) abrufbar. Die aufgenommenen Bilder werden bei vorgegebenen Einstellungen mit einer Latenz von etwa 0.1 Sekunden übertragen.

Je nach Konfiguration des Raspberry Pi im Kapitel „Python“ und in Abhängigkeit davon, ob dieser an das Rücklicht angeschlossen ist oder nicht, wird das Netzwerk beim Starten des Motors bzw. Einlegen des Rückwärtsgangs aktiv.

Erster Ansatz, den Raspberry Pi bei Einlegen des Rückwärtsgangs zu starten, ist durch die längere Bootdauer des Pis nicht zweckdienlich. Dementsprechend wird der Pi derart implementiert, dass dieser mit dem Start der Zündung den Bootvorgang startet und optional durch Erkennung des Rückfahrlichts der Stream gestartet werden kann.

Verwendete Komponenten



Abbildung 1 Projektkomponenten

- 160° Weitwinkelkamera mit csi Schnittstelle
- Raspberry Pi Zero mit W-Lan Komponente (betrieben mit Raspbian)
- 16 GB SD-Karte
- Raspberry Pi Netzteil

Integration in das Fahrzeug

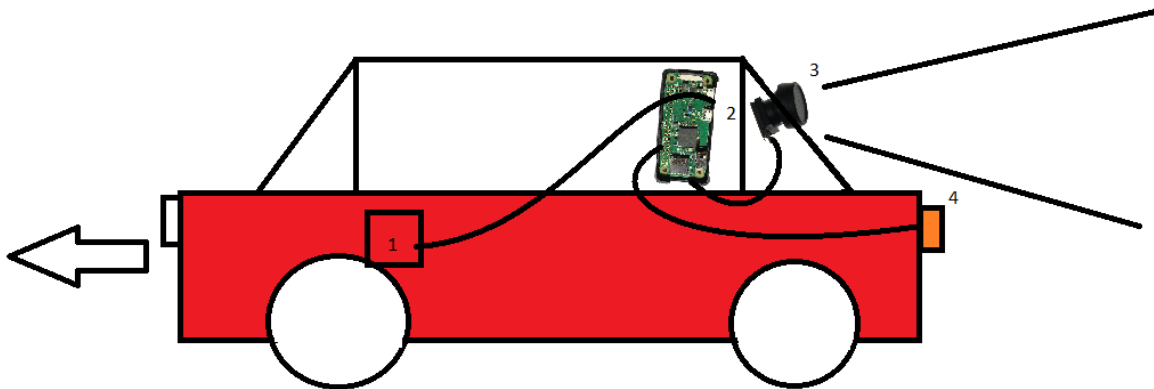


Abbildung 2 Integration der Komponenten in das Fahrzeug

- 1 Versorgung durch eine Sicherung oder ein geeignetes Netzteil
- 2 Raspberry Pi Zero W
- 3 Weitwinkelkamera
- 4 Abfrage des Rückfahrlichts (optional)



Abbildung 3 provisorische Anbringung der Rückfahrkamera an der Heckscheibe

Standalone Network:

Um ein eigenständiges Netzwerk, für das Streamen des Videomaterials zu erstellen und konfigurieren, müssen die Add on's DNSMasq und HostAPD installiert werden. Dies erfolgt durch folgenden Befehl, welcher wie bei zukünftigen Befehlen im Terminal eingegeben wird:

```
sudo apt install dnsmasq hostapd
```

Dadurch wird automatisch eine Datei dhcpd.conf erstellt. Diese Datei muss nun durch folgende Zeilen ergänzt werden:

```
interface wlan0
static ip_address=192.172.13.1/24
nohook wpa_supplicant
```

Dies konfiguriert eine statische IP Adresse, über welche man später einen Zugriff auf das Netzwerk erlangt. Der DHCP benötigt eine Konfigurationsdatei, welche man mit folgenden Befehlen erstellt und aufruft:

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
sudo nano /etc/dnsmasq.conf
```

Diese ist durch zwei Zeilen zu erweitern:

```
interface=wlan0
dhcp-range=192.172.13.2,192.172.13.20,255.255.255.0,24h
```

Dies reserviert und stellt IP Adressen zwischen 192.172.13.2 und 192.168.4.20 für mindestens 24 Stunden zur Verfügung.

Access point host software (hostapd)

Durch den Befehl

```
sudo nano /etc/hostapd/hostapd.conf
```

öffnen sich die Konfigurationen der Access point software, in denen nach gegebener Vorgabe, Zeilen ergänzt bzw. zu ändern sind:

```
interface=wlan0
ssid=RFKNET
hw_mode=g
channel=6
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=hallohallo
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Damit das System diese Konfigurationsdatei findet, wird durch den Befehl

```
sudo nano /etc/default/hostap.conf
```

eine Datei geöffnet in welcher die auskommentierte Zeile

```
#DAEMON_CONF
```

durch

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

zu ersetzen ist.

Bildübertragung

Durch folgende Befehle wird die Bildübertragung installiert:

```
$ sudo apt-get update  
$ sudo apt-get install uv4l uv4l-raspicam
```

Nach der Installation wird die Datei „UV4l-raspicam.conf“ generiert in welchem die Einstellung des Streams eingesehen und modifiziert werden kann. Durch Erfahrungswerte erlangt man mit folgenden Einstellungen die bestmögliche Übertragungsqualität bei akzeptablerer Performance

Quality: 5

Width = 640

Hight = 480

Framerate= 50

Encoding= mjpeg

Durch Erhöhung der Qualität oder der Bildgröße wird die Performance negativ beeinflusst.

Python

Der Ablauf des Systems wird mit einem Python Programm gesteuert.

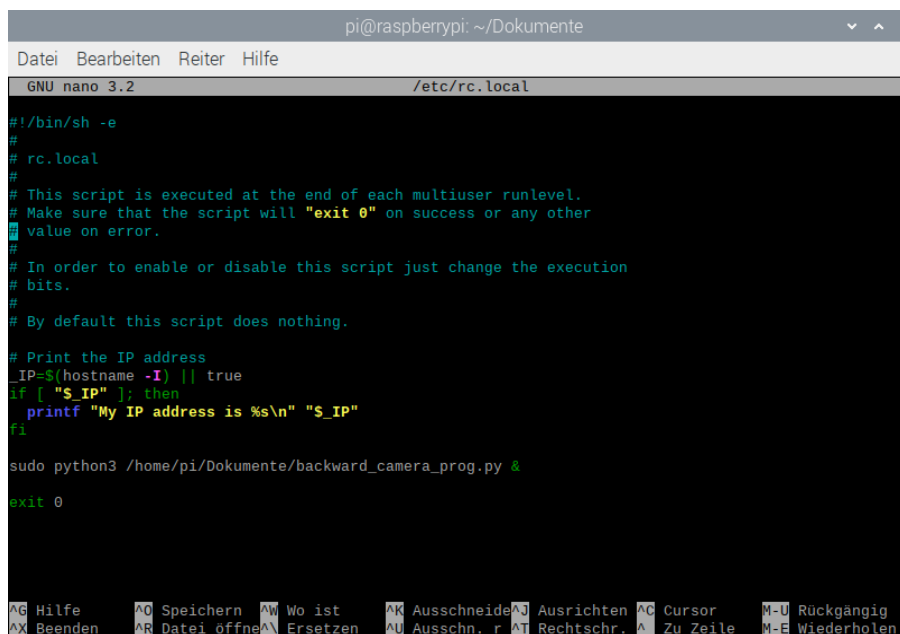
Autostart des Python Programm:

Um ein Python Programm im Autostart zu starten muss folgende Zeile vor

```
exit 0
```

in der Datei „/etc/rc.local“ integriert werden.

```
sudo python3 /home/pi/Dokumente/backward_camera_prog.py &
```



```
pi@raspberrypi: ~/Dokumente
Datei Bearbeiten Reiter Hilfe
GNU nano 3.2 /etc/rc.local

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
# By default this script does nothing.
#
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi

sudo python3 /home/pi/Dokumente/backward_camera_prog.py &

exit 0

^G Hilfe      ^O Speichern ^W Wo ist     ^K Ausschneide ^J Ausrichten ^C Cursor     ^M-U Rückgängig
^X Beenden    ^R Datei öffne ^E Ersetzen  ^U Ausschn. r ^T Rechtschr. ^_ Zu Zeile   ^M-E Wiederholen
```

Der Server startet nun den Stream nach dem Bootvorgang des Raspberry Pi's

Abfrage eines Rückfahrlichts

Eine Erkennung des Rückfahrlichtes kann als Trigger des Streams genutzt werden. Dafür wurde eine Abfrage eines Inputsignals in die Main Funktion des Python Programmes integriert. Die Variable `en_uv4l_start` steht dabei für den Status der Übertragung und `backlight_connected` ist eine Variable, die manuell von 0 auf 1 gesetzt werden muss, sollte man das Rückfahrlicht eines Fahrzeuges abfragen. Die Zeile **`if(GPIO.input(input_pin) == backlight_connected):`** überprüft, ob am `input_pin` ein Signal anliegt und dadurch das Rückfahrlicht erkannt wurde. Sollte dies der Fall sein und der Stream noch nicht laufen, (**`en_uv4l_start=0`**) wird er nun gestartet. Sobald das Signal am `input_pin` abfällt (**`if(GPIO.input(input_pin) == ~backlight_connected):`**) wird der Stream gestoppt. Ist die Abfrage des Rückfahrlichts inaktiv und **`backlight_connected = 0`** kann, ohne dass das ein Signal erkannt wird, der Stream gestartet werden.

```
# main Program

input_pin = 18
pin_detection_init(input_pin)
os.system("sudo systemctl start hostapd")      # aktiviert accesspoint
os.system("sudo pkill uv4l")                  # beenden des Streams
en_uv4l_start = 0                             # Status der Übertragung: aus
backlight_connected = 0                      # bei Ueberwachen des Ruecklichts,
                                              # manuell backlight_connected = 1
                                              # setzen- ansonsten backlight_connected
                                              # = 0

while True:

    if(GPIO.input(input_pin) == backlight_connected):
                                                # solange rueckwaertsgang
                                                # eingelegt / Wenn
                                                # Ueberwachung
                                                # backlight_connected =1 muss
                                                # input_pin =1
        os.system("sudo systemctl start hostapd")    # activate accesspoint

    if en_uv4l_start == 0:

        os.system("sudo service uv4l_raspicam restart") # starten des Streams
        en_uv4l_start = 1                             # Status der
                                                         # Übertragung: an

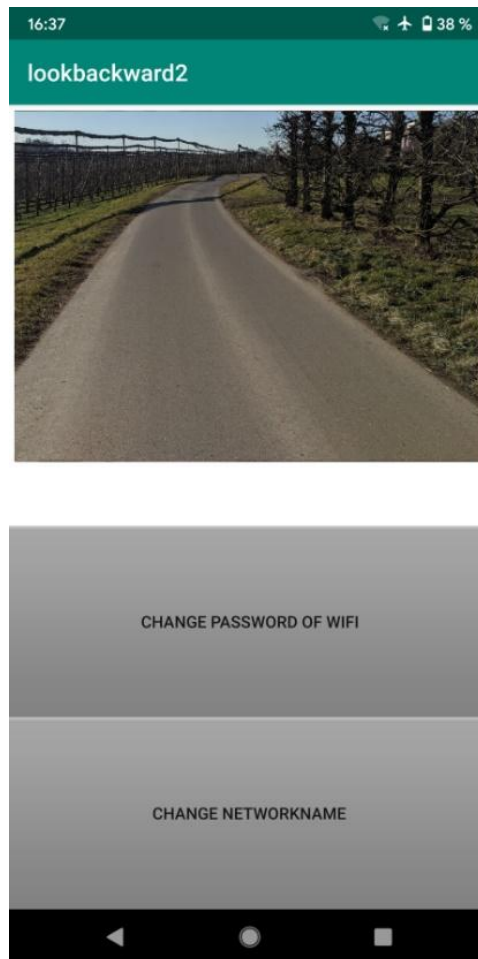
    server_communication()

    if(GPIO.input(input_pin) == ~backlight_connected):
        os.system("sudo pkill uv4l")              # beenden des Streams
        en_uv4l_start == 0                        # Status der Übertragung: aus
```

Verbindung mit dem Netzwerk

Kurz nach dem Startvorgang des Raspberry Pi ist das W-Lan Netzwerk bereits aktiv. Wurde der Netzwerkname nicht umbenannt (Möglichkeit zur Umbenennung siehe Kapitel „Mobile App“) ist das W-Lan unter dem Namen RFKNET via Computer oder Handy zu finden. Das zu Beginn festgelegte Passwort lautet „hallohallo“.

Mobile App



Verwendung der Android-App:

Es besteht die Möglichkeit, den Stream der Raspberry PI Kamera im Browser des Handys bzw. PC's über die IP Adresse des Pi's abzurufen. Um die Bedienung für den Anwender zu erleichtern, wurde die Android App mit dem Namen „lookbackward2“ geschrieben. Die Android Applikation wurde für Geräte der Version Android 4.0.3 (IceCreamSandwich) geschrieben und auf den Android Versionen 6.0.1 (Marshmallow) und 10 getestet.

Funktionen der App

Die Anwendung ermöglicht es, das Kamerabild bei Verbindung mit dem Netzwerk in die App zu übertragen und gleichzeitig Befehle an den Raspberry Pi zu übersenden. Der Nutzer kann optional den vorkonfigurierten Namen „RFK“ des Raspberry Pi Networks sowie das Passwort beliebig abändern. Zu beachten ist allerdings, dass zur Änderung einer der genannten Optionen der Raspberry Pi im nicht verkabelten Zustand sein sollte. Andernfalls ist die Konfiguration nur bei eingelegtem Rückwärtsgang, bzw. einem High Pegel an Pin 18 möglich.

Konfiguration des Wifi-Passworts:

Die Konfiguration des Passworts geschieht über den Button mit der Aufschrift „Change PASSWORD OF WIFI“.

Wird der Button betätigt, erscheint ein Dialogfeld, in welches das aktuelle sowie das neue Passwort eingegeben werden muss. Der Vorgang wird über den „CONFIRM“ Button abgeschlossen.

War die Änderung erfolgreich, wird vom Raspberry Pi ein Feedback in der Form gegeben, dass das Netzwerk für eine kurze Zeit offline und anschließend wieder online geht.

Konfiguration des Wifi-Namen:

Die Änderung des Wifi Namens erfolgt ähnlich dem des Passworts. Nach Betätigung des Buttons mit der Aufschrift „CHANGE NETWORKNAME“ erscheint ein Dialogfeld in welchem der neue Wifi Name sowie das aktuelle Passwort eingegeben werden muss. Auch hier wird der Vorgang wieder über den „CONFIRM“ Button bestätigt und als Feedback zur erfolgreichen Änderung das Netzwerk kurzzeitig ausgeschaltet.