

## Dokumentation: Softwareinstallation und Konfiguration für Roboter-Autos mit Raspberry Pi

### 1. Einleitung

Diese Dokumentation beschreibt die Schritte zur Installation der Software und Konfiguration für das Roboter-Auto mit einem Raspberry Pi.

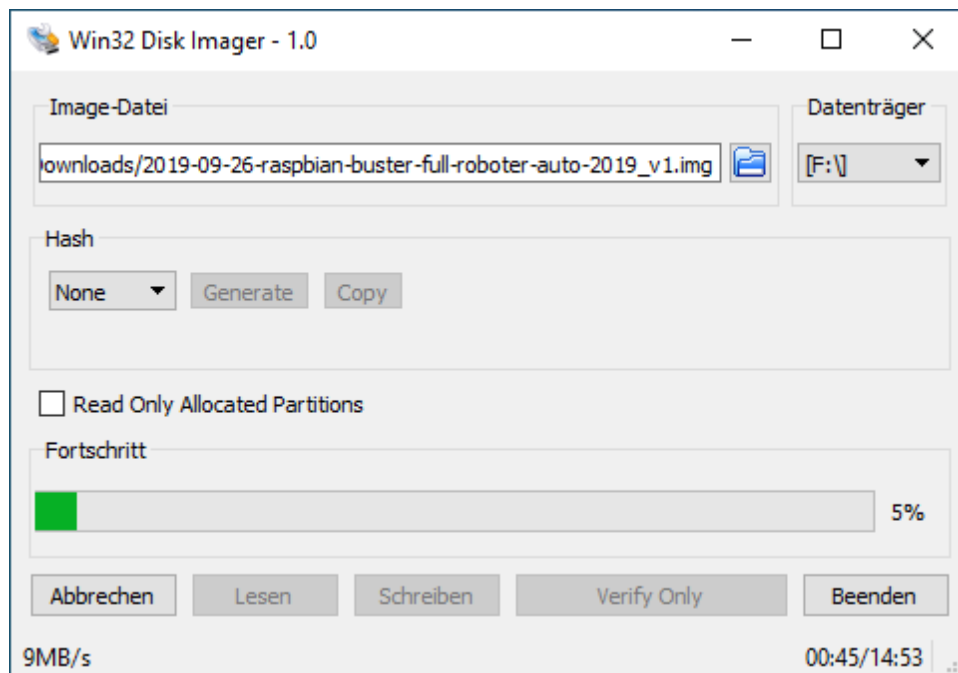
### 2. Voraussetzungen

- Raspberry Pi (Modell 3)
- SD-Karte mit ausreichender Kapazität (mindestens 32 GB)
- Kartenlesegerät
- HDMI-Monitor, Tastatur und Maus für die anfängliche Konfiguration
- Stabile Internetverbindung für den Software-Download und -Aktualisierung

### 3. Betriebssysteminstallation

#### 3.1 Herunterladen der Raspbian Image Datei

Die neueste Raspbian Image Datei wird von der offiziellen Raspberry Pi-Website heruntergeladen.



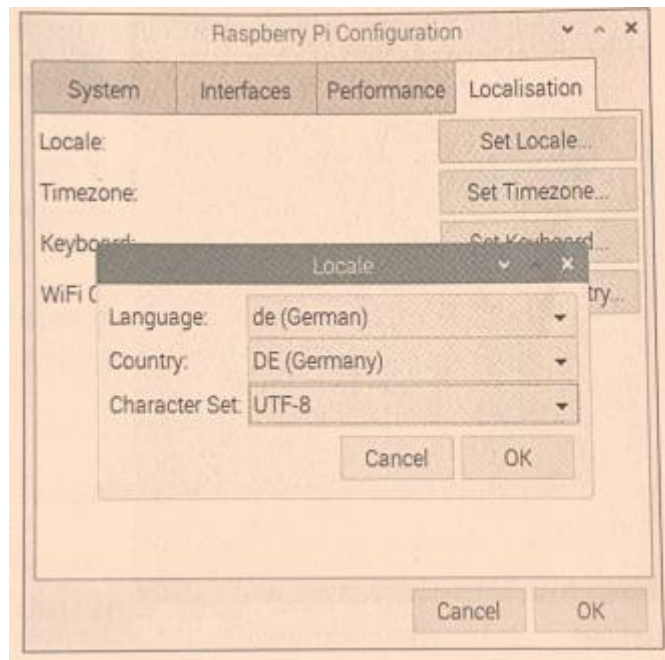
#### 3.2 Schreiben des Images auf die SD-Karte

Die Raspbian Image Datei wird mithilfe des Programms "win32diskimager" auf die SD-Karte geschrieben. Die SD-Karte wird in ein Kartenlesegerät eingelegt und den Anweisungen des Programms gefolgt.

### 4. Erster Start und Konfiguration

#### 4.1 Anschließen des Raspberry Pi

Der Raspberry Pi wird an einen HDMI-Monitor, eine Tastatur und eine Maus angeschlossen. Die vorbereitete SD-Karte wird eingelegt.



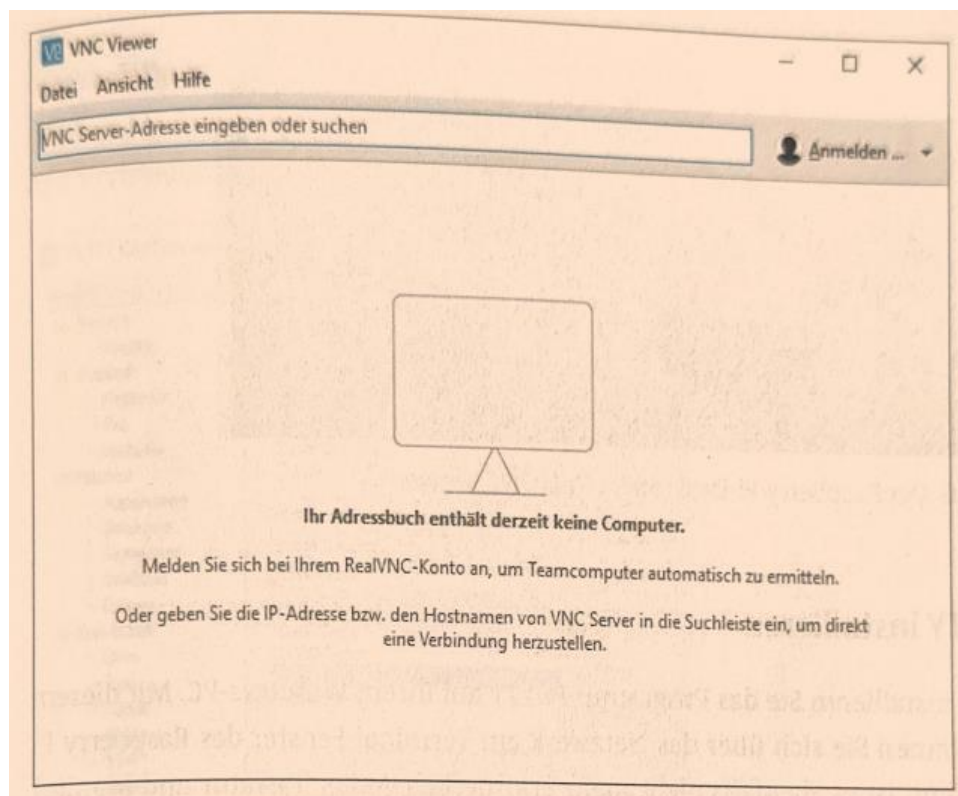
#### 4.2 Sprache und Land auswählen

Beim ersten Start werden die gewünschte Sprache und das Land ausgewählt. Die entsprechenden Einstellungen werden gemäß den Präferenzen vorgenommen.

#### 4.3 WLAN-Einrichtung

Die WLAN-Verbindung wird konfiguriert, um eine Internetverbindung herzustellen. Die Zugangsdaten für das WLAN-Netzwerk werden eingegeben.

### 5. Installation und Konfiguration von Software



## 5.1 Installation von Zusatzprogrammen

Die erforderlichen Programme werden installiert:

- Real VNC Server für den Remote-Zugriff
- Samba Server für die Netzwerkfreigabe
- Python als Programmbibliothek für das Roboter-Auto
- Flask für den Webserver
- MJPG-Streamer für das Live-Video-Streaming

## 5.2 Konfiguration der Programme

Die Konfiguration jedes installierten Dienstes erfolgt gemäß den Anforderungen des Roboter-Auto-Projekts. Netzwerkfreigaben, Benutzerzugriffsrechte und IP-Einstellungen werden durchgeführt.

## 5.3 Hinzufügen der Programmcodes

- Ein Steuerungsprogramm für den Motortreiber in Python wird erstellt.
- Ein Steuerungsprogramm für das Roboter-Auto in Python wird entwickelt.
- Ein Webinterface zur Steuerung des Roboter-Autos über den Webbrowser wird implementiert.

## 6. Aktivieren der Peripheriegeräte

### 6.1 Kamera anschließen

Die Kamera wird über die CSI-Schnittstelle an den Raspberry Pi angeschlossen.

### 6.2 Kameraaktivierung in Raspbian

Das Raspberry Pi Konfigurationsmenü im Raspbian-Betriebssystem wird geöffnet, um die Kamera in den Interface-Einstellungen zu aktivieren.

## 7. Konfiguration des Video-Streaming-Servers

Die Konfiguration des MJPG-Streamers für das Live-Video-Streaming von der angeschlossenen Kamera wird abgeschlossen.

## 8. Autostart der Programme konfigurieren

Der Cron-Daemon wird genutzt, um Programme zeitgesteuert auszuführen. Der MJPG-Streamer und das Steuerungsprogramm "robot-control-web.py" werden im Cron-Daemon eingetragen, damit der Video-Stream und die Webinterface-Steuerung automatisch gestartet werden.

## 9. Betriebssystem- und Firmware-Aktualisierung

Das Betriebssystem und die Raspberry Pi Firmware werden regelmäßig auf den neuesten Stand gebracht, um von den neuesten Funktionen und Sicherheitsupdates zu profitieren. Dazu werden folgende Schritte ausgeführt:

- Das Betriebssystem wird über das Terminal mit dem Befehl **sudo apt update** gefolgt von **sudo apt upgrade** aktualisiert.
- Die Raspberry Pi Firmware wird mit dem Befehl **sudo rpi-update** aktualisiert.

```
pi@raspberrypi: ~  
buster/main armhf libsmbclient armhf 2:4.9.5+dfsg-5 [152 kB]  
Holen:9 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
buster/main armhf samba-libs armhf 2:4.9.5+dfsg-5 [4.695 kB]  
Holen:10 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
n buster/main armhf libwbclient0 armhf 2:4.9.5+dfsg-5 [127 kB]  
Holen:11 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
n buster/main armhf bzip2 armhf 1.0.6-9.1 [46,1 kB]  
Holen:12 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
n buster/main armhf libbz2-1.0 armhf 1.0.6-9.1 [42,9 kB]  
Holen:13 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
n buster/main armhf nano armhf 3.2-3 [523 kB]  
Holen:14 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
n buster/main armhf bind9-host armhf 1:9.11.5.P4+dfsg-5.1 [268 kB]  
Holen:15 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
n buster/main armhf libbind9-161 armhf 1:9.11.5.P4+dfsg-5.1 [243 kB]  
Holen:16 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
n buster/main armhf libiscfg163 armhf 1:9.11.5.P4+dfsg-5.1 [258 kB]  
Holen:17 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
n buster/main armhf libisccl161 armhf 1:9.11.5.P4+dfsg-5.1 [233 kB]  
Holen:18 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
n buster/main armhf libdns1104 armhf 1:9.11.5.P4+dfsg-5.1 [1.076 kB]  
Holen:19 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
n buster/main armhf libisc1100 armhf 1:9.11.5.P4+dfsg-5.1 [421 kB]  
14% [19 libisc1100 45,1 kB/421 kB 11%] [3 libraspberrypi-doc 14,5 MB/31,4 MB 46%]
```

## 10. Fazit

In dieser Dokumentation wurden die Schritte zur Softwareinstallation und Konfiguration für Roboter-Autos mit einem Raspberry Pi beschrieben. Mit diesen Schritten sollte das System einsatzbereit sein und die Grundlage für die Entwicklung und Programmierung des Roboter-Auto-Projekts bieten.