



WLAN-AP mit regelmäßigem PSK-Tausch und
QR-Code Anmeldung

Luca Asmus
Marius Würstle
Rolf Wiersch

November 26, 2020

1 Zusammenfassung

Contents

1 Zusammenfassung	1
2 Abbildungsverzeichnis	2
3 Allgemeines	3
3.1 Fachbegriffe	3
4 Hardware	3
4.1 Raspberry Pi	3
4.2 Raspberry Pi Shield - Display LCD-Touch, 3,2in	4
4.3 SD-Karte	5
5 Software	5
5.1 balenaEtcher	5
5.2 hostapd	5
5.3 dnsmasq	5
5.3.1 netfilter-persistent und iptables-persistent	5
5.4 cron	5
5.5 Python3.7	5
5.5.1 pyqrcode	6
5.5.2 gpiozero	6
6 Vorbereitung des Raspberry Pi	6
6.1 Auswahl und Installation des Betriebssystems	6
6.2 Aktualisierung und Paketinstallation	6
6.3 SSH Zugriff einrichten	6
7 Konfiguration des RaspberryPi als funktionalen Access-Point	6
7.1 WLAN Interface	6
7.2 Routing	7
7.3 DNS und DHCP	7
7.4 Access Point Einstellungen	7
8 Passwortgenerierung	7
9 Ausgabe des Passworts	8
10 Fazit mit Ausblick	8
11 Quellenverzeichnis	8

2 Abbildungsverzeichnis

List of Figures

1	Raspberry Pi 3b - Quelle: [1]	3
2	Touchscreen Display für den Raspberry Pi - Quelle: [3]	4

3 Allgemeines

3.1 Fachbegriffe

4 Hardware

4.1 Raspberry Pi

Der Raspberry Pi wurde für junge Menschen entwickelt, um ihnen eine preisgünstige Möglichkeit zu bieten, sich mit der Informatik zu beschäftigen. Der Einplatinencomputer ist etwa Kreditkartengroß und kam Anfang 2012 auf den Markt. Er ermöglicht einen schnellen und praktischen Weg um Wissen in den Bereichen Programmieren und Hardware. Zudem ist er vielseitig einsetzbar, in diesem Fall wird er zu einem Access-Point konfiguriert. Tech-

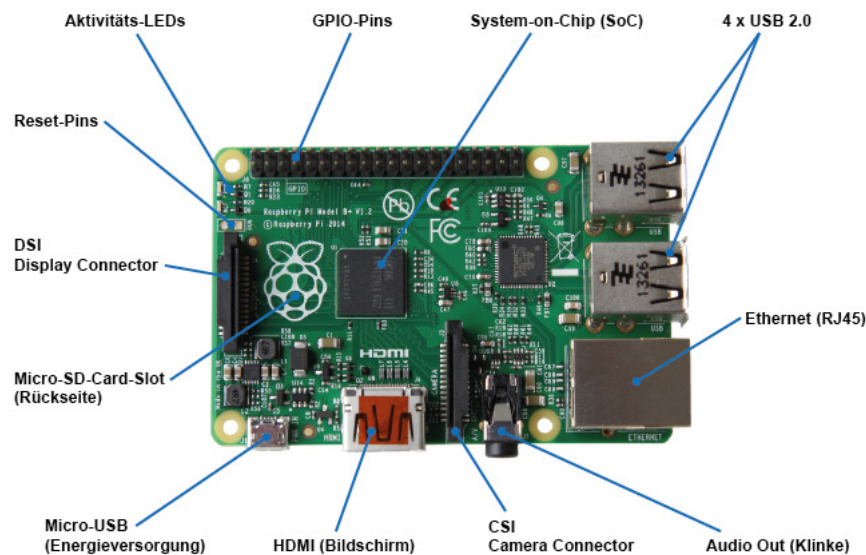


Abbildung 1: Raspberry Pi 3b - Quelle: [1]

nische Spezifikationen unseres Raspberry Pi 3b:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO

- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

4.2 Raspberry Pi Shield - Display LCD-Touch, 3,2in

Der Touchscreen verwandelt den Raspberry Pi zu einem vollwertigen Touch-PC auf. Für zusätzliche Funktionen besitzt der Display 3 Buttons an der Seite, welche einfach über die GPIO Pins eingelesen werden können. Technische



Abbildung 2: Touchscreen Display für den Raspberry Pi - Quelle: [3]

Spezifikationen unseres Raspberry Pi Shield - Display LCD-Touch, 3.2in:

- Display 8,13cm (3,2")
- Auflösung 320 x 240 Pixel
- LED-Hintergrundbeleuchtung
- 3 frei belegbare Taster (angebunden an GPIO12, 16, 18)
- SPI-Schnittstelle
- Touchscreen Technologie resistiv

4.3 SD-Karte

Die SD-Karte ist eine SanDisk extreme mit einer Speicherkapazität von 32GB. Sie dient als Speichermedium des Raspberry Pi's. Zu Beginn wird das Betriebssystem auf die Karte geflasht von dieser wird der Einplatinencomputer gebootet.

5 Software

5.1 balenaEtcher

Flash Sd Card

5.2 hostapd

Mit hostapd ist es möglich Geräte, die ein WLAN-Modul besitzen als Access Point zu betreiben. Jedoch können keine Einstellungen im Bereich IP und Routing vorgenommen werden. Die Software ist nur für das Erstellen eines "wireless Ethernet switches" zuständig. [2]

5.3 dnsmasq

Geräte in einem Netzwerk benötigen zur Kommunikation eine IP Adresse und einen DNS Server für die Namensauflösung. Deshalb muss in diesem Projekt ein DHCP und DNS erstellt werden. Von diesen bekommen die Endgeräte ihre IP Konfiguration im WLAN. Die Software dnsmasq wird in diesem Projekt verwendet um dies zu ermöglichen.

5.3.1 netfilter-persistent und iptables-persistent

Für die Durchführung des Projektes ist es nötig iptable-Regeln anzulegen. Diese sollten nach einem Neustart nicht neu angelegt werden müssen. Deshalb wurden die Pakete netfilter-persistent und iptables-persistent installiert. Damit können die Regeln in eine Datei abgespeichert und beim Neustart automatisch geladen werden.

5.4 cron

Planen der Ausführung des Skripts

5.5 Python3.7

Für die Skripte zur Passwortgenerierung, QR-Code Generierung und zum Einlesen der Buttons wird die Sprache Python verwendet. Python 3.7 kommt bei Raspbian vorinstalliert und erleichtert durch verschiedene Bibliotheken die Umsetzung des Projektes.

5.5.1 pyqrcode

5.5.2 gpiozero

6 Vorbereitung des Raspberry Pi

6.1 Auswahl und Installation des Betriebssystems

Um mit dem Projekt beginnen zu können musste zuerst ein Betriebssystem bestimmt werden. Es wurde sich für das Raspberry Pi OS Lite entschieden. Begründet wurde diese Entscheidung durch die weniger vorinstallierten Pakete und einer fehlender grafischen Bedienoberfläche. Hierdurch konnte Speicherplatz und Sicherheitsrisiken eingespart werden. Je weniger unbenutzte Software, desto weniger Angriffsfläche.

Nach der Auswahl des Betriebssystems konnte dieses auf eine SD-Karte geschrieben werden. Hierzu wurde die Software balenaEtcher verwendet.

6.2 Aktualisierung und Paketinstallation

Nach der Neuinstallation eines Betriebssystems fehlt diesem oft die aktuellsten Versionen von Softwarepaketen und Updates. Deshalb wurde diese zuerst aktualisiert und installiert. So werden Konflikte aufgrund veralteter Software vermieden und die Sicherheit verbessert. Darauf folgte das Nachinstallieren der für das Projekt noch benötigten Pakete. Diese wurden im Abschnitt Software genauer beschrieben.

6.3 SSH Zugriff einrichten

Da das Projektteam aus drei Personen besteht, wurde ein SSH Zugriff in den Einstellungen des Raspberry Pi eingerichtet. Die Einstellungen können mit folgendem Befehl geöffnet werden:

```
1 sudo raspi-config
```

In diesem Zuge wurde das der SSH Zugriff aktiviert und das Standardpasswort geändert. Durch den Zugriff konnte das parallele Arbeiten am Projekt ermöglicht werden.

7 Konfiguration des RaspberryPi als funktionalen Access-Point

7.1 WLAN Interface

Der Raspberry Pi benötigt eine statische IP Konfiguration für sein WLAN Interface. Diese wird in der Datei `/etc/dhcpd.conf` vorgenommen. Die Datei

wir geöffnet und um folgendes ergänzt:

```
1 interface wlan0
2     static ip_address=192.168.4.1/24
3     nohook wpa_supplicant
```

Mit der 192.168.4.1 wird eine statische IP Adresse vergeben unter die der Raspberry Pi im WLAN erreichbar ist. Weiterhin wird der wpa_supplicant deaktiviert um keine Konflikte mit hostapd zu verursachen.

7.2 Routing

Der Access Point muss den Datenverkehr der Endgeräte im WLAN zum Router weiterleiten können. Hierzu wird in /etc/sysctl.d/routed-ap.conf ein Eintrag hinzugefügt bzw. das Kommentarzeichen entfernt:

```
1 # Enable IPv4 routing
2 net.ipv4.ip_forward=1
```

Endgeräte können nun den Hauptrouter erreichen. Um jedoch eine Kommunikation zu ermöglichen muss NAT eingestellt werden. Die wird durch einen Eintrag in die iptables Firewall erreicht:

```
1 sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Bei Datenverkehr zum Hauptrouter wird nun die Absender IP Adresse der Endgeräte mit der IP der LAN-Schnittstelle ersetzt. Bei Rückantworten an den Raspberry Pi werden diese an den jeweiligen Absender richtig weitergeleitet.

Um Firewall Regel bei einem Neustart zu behalten, wurde diese abgespeichert:

```
1 sudo netfilter-persistent save
```

7.3 DNS und DHCP

7.4 Access Point Einstellungen

8 Passwortgenerierung

Die Passwortgenerierung wird mithilfe eines Python Skripts gelöst. Dieses ist in unserem GitHub repository hinterlegt und für jeden zugänglich (ref zum Link). Das Skript verwendet die zwei Imports string und secrets. Mithilfe der Bibliothek string können die für Bash problematischen Zeichen aus dem

Alphabet entfernt werden. Das secrets Modul wird für das Generieren von stark kryptographischen Passwörter verwendet. Die verwendete Funktion secrets.choice wählt aus der mitgelieferten Sequenz ein zufälliges Zeichen aus. Welches anschließend an den schon vorhandenen String angehängt wird. Dies wird 10 mal wiederholt.

```
1 import secrets
2 import string
3
4 def get_random_password():
5     temp = string.ascii_letters + string.digits
6     + string.punctuation
7
8     alphabet = temp.replace('\\', '')
9     .replace('\\', '')
10    .replace('"', '').replace("'", '')
11    .replace(';', '')
12
13    password = ''.join(secrets
14    .choice(alphabet) for i in range(10))
15    return password
16
17 if __name__ == "__main__":
18     print(get_random_password())
```

9 Ausgabe des Passworts

10 Fazit mit Ausblick

11 Quellenverzeichnis

References

- [1] Elektronik-kompodium, 2020. <http://www.elektronik-kompodium.de/sites/raspberry-pi/bilder/19052512.jpg> [aufgerufen am 25.11.2020].
- [2] Gentoo, 2020. https://wiki.gentoo.org/wiki/Hostapd#Capabilities_of_Hostapd [aufgerufen am 26.11.2020].
- [3] Reichelt, 2020. https://cdn-reichelt.de/bilder/web/artikel_ws/A300/TFTV2.jpg [aufgerufen am 26.11.2020].