

Installationsanleitung und Konfigurationsbeschreibung - Benutzerdefiniertes Raspbian Image für den Raspberry Pi - Pi4FWEI

Lukas Fritzenwanger/Feuerwehr Uttendorf/Pzg.

09.03.2016

1 Versionsübersicht

1.1 05.02.2016

- Erstversion
- Image 2016-02-04 fwei4pi_V0.zip

1.2 09.03.2016

- Dokumentation korrekturgelesen und korrigiert.
- Im Image den Dienst nodm beim Start von einer vorhandenen Netzwerkverbindung abhängig machen
- Image 2016-03-09 fwei4pi_V1.zip

2 Zielsetzung

Erstellung eines vorgefertigten SD-Karten-Images für die Verwendung auf dem pi. Das image soll online zur Verfügung stehen und für den Enduser so einfach wie möglich zu installieren und konfigurieren sein. Der Inhalt des Images soll sich nur auf die notwendigsten Programme beschränken, um die Image-Größe zu reduzieren. FWEI soll beim Start des PI automatisch im Vollbildmodus ohne URI-Leiste angezeigt werden. Zur Bedienung des PI sollte eine Maus oder ein anderes Zeigergerät (Funktionen Zoom, Pan) ausreichen. Die Anleitung soll auch für Benutzer verständlich sein, die sich nicht unbedingt auf unixoiden Systemen zu Hause fühlen.

3 Anleitungen

3.1 Verwendung des vorgefertigten Image

Hier wird beschrieben, wie das zur Verfügung gestellte vordefinierte Image zur eigenen Verwendung am PI installiert und konfiguriert wird. Möchte man sich

selbst ein solches Image zusammenschneiden, einfach unter 3.2 weiterlesen. Hier aber nun die Beschreibung für den einfacheren Weg. Benötigt werden:

- Raspberry Pi (gegenständlich Pi 2)
- Micro-SD Karte (empfohlen ab 4 GB) und Adapter auf SD-Standard-Größe
- PC/Laptop mit Kartenleser und Betriebssystem (Linux, Windows, MacOS)
- Download des Images von s122.at
- Entpacken des images in einen wiederfindbaren Ordner (Betriebssysteme haben Unzip-Programme an Bord)
- Schreiben der img-Datei auf die SD-Karte. Hier gibts je nach verwendetem Betriebssystem Unterschiede.

3.1.1 Linux

- SD-Karte mit Adapter einlegen
- Auf der Shell mit dem Befehl *dmesg* nachsehen, welches Device erkannt wurde, z.B. *sdb*
- Gegebenfalls die automatisch gemountete Partition wieder unmounten. *sudo umount /dev/sdb1 /dev/sdb2*
- Mit *dd* das Image auf die SD-karte schreiben. Dazu in einer Shell in das Verzeichnis wechseln, wo das Image liegt. Dann folgenden Befehl absetzen: *sudo dd if=raspi-fwei.img of=/dev/sdb bs=4M*. Achtung - beim Ziel-Device (of=) keinen Tipp-Fehler machen - es gibt keine Sicherheitsabfrage und *dd* schreibt auf jedes Device - auch auf die eigene Festplatte!
- warten bis fertig geschrieben wurde (dauert je nach sd-karte bis zu 10 Minuten!)
- Zwei mal (hab angst ;o)) den Befehl *sudo sync* absetzen - Schreibt den file-system-puffer fertig
- SD-Karte ggf. unmounten, entnehmen und in pi einbauen

3.1.2 Windows

- Den Schreibvorgang mit dem Programm Win32 Disk Imager durchführen

3.1.3 MacOS

- *Imagewriter* oder *dd* mit dem richtigen device verwenden (*diskutil* weiß Bescheid)
- Pi fertig anschliessen - Bildschirm, Maus, Tastatur, Netzwerk (USB-Wlan?). Durch das Anschließen der Stromversorgung startet der Pi automatisch

- Wenn alles geklappt hat (Internetverbindung), wird bereits ein Browserfenster mit dem Demologin angezeigt. Jetzt muss noch der Login-Code der eigenen Feuerwehr eingegeben werden. Dazu mit STRG+ALT+F1 auf das erste tty wechseln und einloggen.
- Standard-logindaten (stand 02/2016) User: pi Passwort: raspberry
- mit *echo logincode» /home/pi/.config/fwei/fwei-login* den eigenen Zugangscode in die Config-Datei schreiben (Bsp: <https://feuerwehreinsatz.info/1/123456> => 123456 als logincode verwenden.
- Mit *sudo reboot* den Pi neu starten. Wenn alles glatt gegangen ist, wird FWEI der eigenen Feuerwehr jetzt im Browser angezeigt.

3.1.4 WLAN

Wenn ein USB WLAN Adapter zur Internetverbindung verwendet wird, muss eine weitere Konfigurationsdatei angepasst werden.

- Ebenfalls mit STRG-ALT-F1 auf die Shell wechseln und einloggen.
- Mit *sudo nano /etc/wpa_supplicant/wpa_supplicant.conf* die wlan-config datei öffnen.

```
network={
    ssid='SSID_eintragen'
    scan_ssid=1
    key_mgmt=WPA-PSK
    psk='passwort_eintragen'
}
```

- Ssid und Psk-Passwort richtigstellen, dann die Datei mit STRG-X beenden. Abfrage mit y und Enter beantworten.
- Pi neu starten - wlan sollte jetzt funktionieren
- Fertig :o)

3.2 Anleitung zur Selbst-Erstellung eines vorgefertigten Images

Hier wird beschrieben, wie das unter 3.1 vorgefertigte Image selbst erstellt werden kann. Benötigt werden:

- Raspberry Pi (gegenständlich Pi 2)
- Micro-SD Karte (min. 2 GB, empfohlen ab 4 GB) und Adapter auf SD-Standard-Größe
- PC/Laptop mit Kartenleser und Betriebssystem
- Keine Scheu vor der Commandline/Shell
- Grundlegende Kenntnisse von Texteditoren wie nano oder vi
- Viel Kaffee oder Getreidegärstoffgetränk ;o)

Zur besseren Lesbarkeit wird hier nur der Weg in auf Debian Linux beschrieben. Mac- und Windows-Profis können gerne ergänzen.

- Download des raspbian-lite Images von raspberry.org
- Entpacken des Images in einen wiederfindbaren Ordner
- SD-Karte mit Adapter einlegen
- Auf der Shell mit dem Befehl `dmesg` nachsehen, welches Device erkannt wurde, z.B. `sdb`
- Gegebenfalls die automatisch gemountete Partition wieder unmounten.
`sudo umount /dev/sdb1 /dev/sdb2`
- Mit `dd` das Image auf die SD-Karte schreiben. Dazu in einer Shell in das Verzeichnis wechseln, wo das Image liegt. Dann folgenden Befehl absetzen:
`sudo dd if=2015-11-21-raspbian-jessie-lite.img of=/dev/sdb bs=4M` Achtung - beim Ziel-Device (`of=`) keinen Tipp-Fehler machen - es gibt keine Sicherheitsabfrage und `dd` schreibt auf jedes Device - auch auf die eigene Festplatte!
- Warten bis fertig geschrieben wurde (dauert je nach sd-karte bis zu 5 Minuten!)
- Zwei mal (hab angst ;o)) den Befehl `sudo sync` absetzen - Schreibt den file-system-puffer fertig
- Partitionsgröße der SD-Karte adaptieren. Am Einfachsten geht das aus Sicht des Authors mit `gparted` (installieren z.B unter debian mit `apt-get` (oder `aptitude`) `install gparted`)
- `gparted` starten (GUI, X erforderlich).
- Gegebenfalls unmounten der frisch geschriebenen SD-Karte `sudo umount /dev/sdb1 /dev/sdb2`

- Im Fenster von gparted im Dropdown-Menü rechts oben das richtige Device SD-Karte auswählen (in gegenständlichem Fall /dev/sdb)
- Die zweite partition /dev/sdb2 rechts-klicken und resizen, z.B. auf 4 GB (minimum 2 GB)
- Erfolgsmeldung abwarten und gparted beenden
- SD-Karte entnehmen und in Pi einbauen, alle notwendigen IO-Geräte anschließen. Es reicht die Stromversorgung und ein LAN Anschluss in einem Netz mit DHCP Server. Wenn verfügbar, gehts natürlich auch mit Maus, Bildschirm und Tastatur. (Empfohlen, wenn WLAN benötigt wird - andernfalls muss mit der seriellen Console des PI gearbeitet werden, um das WLAN zu konfigurieren. Geht auch, sprengt aber ein wenig diesen Rahmen - Anleitung unter linux.org)
- Booten des Pi
- Ermitteln der IP-Adresse des Pi. Entweder am DHCP-Server nach neuen Geräten suchen, mit *nmap* `nmap man page` das netzwerk scannen oder wenn verfügbar physisch am Pi per `/sbin/ifconfig` ermitteln
- Physisch bzw. per ssh einloggen. standard-logindaten (stand 02/2016)
user: pi pw: raspberry
- Achtung: britisches Tastaturlayout, d.H. beim Passwort statt y z eingeben!
- Tastaturlayout ändern:
- `sudo raspi-config`
- Im Menü unter Punkt 5 können Spracheinstellungen, Zeitzone und Tastaturlayout angepasst werden (nicht zwingend erforderlich). Einfach der Menüführung folgen, sollte selbsterklärend sein. Wenn fertig, raspi-config mit Finish beenden und neu starten mit `sudo reboot`
- WLAN-Verbindung: Die Datei `/etc/wpa_supplicant/wpa_supplicant.conf` mit einem Texteditor (nano, vi) anpassen (eigene SSID und Passwort)

```
network={
    ssid="SSID_eintragen"
    scan_ssid=1
    key_mgmt=WPA-PSK
    psk="passwort_eintragen"
}
```
- SSID und PSK-Passwort richtigstellen, dann die Datei speichern, Editor beenden und Pi neu starten
- Prüfen der Internetverbindung: `sudo ping -c 4 8.8.8.8` sollte erfolgreich sein (kein Timeout/Packet loss)
- Updaten der Paketverwaltung und des Images: `sudo aptitude update` -(es kann auch mit apt-get gearbeitet werden - Geschmackssache. In dieser Anleitung wird aptitude verwendet)
- Installieren der notwendigen Pakete

- `sudo aptitude -y full-upgrade`
- `sudo aptitude install -y xserver-xorg nodm matchbox-window-manager unclutter uzbl`
- Konfigurationsdateien anpassen (wieder mit Texteditor nach Wahl: nano, vi)
- uzbl browser einmal starten mit `uzbl`, um Verzeichnisse zu schreiben
- `cd /home/pi/.config`
- `mkdir fwei`
- `echo mein-login-code» ./fwei/fwei-login`
- `cd uzbl`

- Die Datei `config-custom` mit einem Texteditor erstellen (z.B. `vi config-custom`). Folgender Inhalt:

```
set show_status      = 0
set status_top       = 0
set status_format    = <span font_family="monospace">@mode_section @keycmd_section \
@progress_section @status_section @scroll_section @selected_section @download_section</span>
set default_mode     = command
set geometry=maximized
```

- Die Datei `.xsession` mit einem Texteditor erstellen
- `cd /home/pi`
- `vi` oder `nano .xsession`
- Folgender Inhalt:

```
#!/bin/bash

xset s off          # don't activate screensaver
xset -dpms          # disable DPMS (energy star) features
xset s noblank      # don't blank the video device

FWEICODE='head -n 1 /home/pi/.config/fwei/fwei-login '

#wait for network to come up - essential on wlan
#sleep 5s;

while true; do
  uzbl -u https://feuerwehreinsatz.info/1/SFWEICODE -c /home/pi/.config/uzbl/config-custom &
  exec matchbox-window-manager -use_titlebar no
  sleep 2s;
done
```

- Den Login Manager konfigurieren
- `sudo vi` oder `sudo nano /etc/default/nodm`
- Folgender Inhalt:

```
# nodm configuration

# Set NODM_ENABLED to something different than 'false' to enable nodm
NODM_ENABLED=true

# User to autologin for
NODM_USER=pi

# First vt to try when looking for free VTs
NODM_FIRST_VT='7'

# X session
NODM_XSESSION=/etc/X11/Xsession

# Options for the X server
```

```
NODM_X_OPTIONS='-nolisten tcp'
# If an X session will run for less than this time in seconds, nodm will wait an
# increasing bit of time before restarting the session.
NODM_MIN_SESSION_TIME=60
```

- nodm beim Start des Pi von einer vorhandenen Netzwerkverbindung abhängig machen
- *sudo vi* oder *sudo nano /etc/init.d/nodm* (nodm verwendet noch init anstatt von systemd)

- Folgender Inhalt:

```
die Zeile
# Required-Start: $remotefs
durch
# Required-Start: $network
ersetzen.
```

- Automatischen Reboot konfigurieren
- *sudo su*
- *cd /root/*
- *mkdir scripts*
- Im Ordner scripts mit einem Texteditor (vi, nano) die Datei reboot-rand.sh erstellen und folgenden Inhalt einfügen:
- *vi /root/scripts/reboot-rand.sh*

```
#!/bin/bash
#reboot the machine now + random minutes
#3 options for creating a random number between 1 and 10.
# using option 2 in this case, gnu shuf is installed on pi image
#option 1: use bash RANDOM
#OFFSET1=$((RANDOM%10)+1)
#option 2: use gnu shuf
OFFSET2=$(shuf -i1-10 -n1)
#option 3: use /dev/urandom
#OFFSET3=$(grep -ml -ao '[0-9]' /dev/urandom | sed s/0/10/ | head -n1)
#echo $OFFSET1
echo $OFFSET2
#echo $OFFSET3
#
shutdown --reboot +$OFFSET2
```

- Datei speichern und ausführbar machen
- *chmod 0700 /root/scripts/reboot-rand.sh*
- root-session verlassen mit *exit*
- crontab anpassen
- *sudo crontab -e*
- Folgende Zeilen ganz unten einfügen:

```
#restart weekly on monday, 01:10+rand min
10 1 * * 1 /bin/bash /root/scripts/reboot-rand.sh &
```

- Boot-Splash erstellen

- `sudo aptitude install -y fbi`
- Init-Script erstellen
- `sudo vi /etc/aasplash`
- Folgender Inhalt:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          aasplash
# Required-Start:
# Required-Stop:
# Should-Start:
# Default-Start:    S
# Default-Stop:
# Short-Description: Show custom splashscreen
# Description:       Show custom splashscreen
### END INIT INFO

do_start () {
    /usr/bin/fbi -T 1 --noverbose -a /etc/splash.jpg
    exit 0
}

case "$1" in
    start|*)
        do_start
        ;;
    restart|reload|force-reload)
        echo "Error: argument '$1' not supported" >&2
        exit 3
        ;;
    stop)
        # No-op
        ;;
    status)
        exit 0
        ;;
    *)
        echo "Usage: aasplash [start|stop]" >&2
        exit 3
        ;;
esac
:
```

- Script ausführbar machen und zum entsprechenden Runlevel verlinken
- `sudo chmod 0755 /etc/init.d/aasplash`
- `sudo inserv /etc/init.d/aasplash`
- Übertragen des gewünschten boot-splash bildes z.B. per `scp`
- Auf PC/Laptop: `scp bild.jpg pi@ip-adresse-des-pi:`
- Am Pi: `sudo mv /home/pi/bild.jpg /etc/splash.jpg`
- Boot-Nachrichten auf anderes Terminal umleiten
- `sudo vi /boot/cmdline.txt`
- `console=tty1` auf `console=tty4` ändern
- Am Ende der Zeile `loglevel=3` ergänzen
- Falls die Himbeeren beim Starten auch noch stören, am Ende der Zeile noch `logo.nologo` anhängen
- Pi neu starten `sudo reboot` - Voila!
- Das Image ist somit fertig konfiguriert.

- Jetzt den Pi hinunterfahren mit `sudo poweroff`
- Die SD-Karte mit Adapter auf einem Linux-PC oder Laptop mounten
- Den Inhalt der kompletten SD-Karte in eine Datei schreiben
- Dazu die Karte mounten und schauen, wie groß alle Partitionen der SD-Karte zusammen sind
- `df -B1` gibt die Größen in 1-Byte-Blocken an
- Die Größen der Partitionen addieren und um MB umrechnen.
- mit `dd` ein Image der SD-karte erstellen
- Beispiel: SD-Karte wurde als `/dev/sdb` erkannt
- `sudo dd if=/dev/sdb of=fwei4pi.img bs=1M count=2100`
- 2100*1 MB-Block ergibt 2100 MB
- Hier lieber etwas mehr lesen als wirklich benötigt - wir schneiden das Image noch zu. Dauert je nach SD-Karte ca 5 Minuten.
- Die Partitionen des Image anschauen
- `sudo fdisk -l fwei4pi.img`

- Die Ausgabe sollte in etwa so aussehen:

```
Disk fwei4pi.img: 2.2 GiB, 2306867200 bytes, 4505600 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xa17b9219

Device      Boot  Start      End  Sectors  Size Id Type
fwei4pi.img1    8192  131071  122880    60M c W95 FAT32 (LBA)
fwei4pi.img2  131072 4227071 4096000    2G 83 Linux
```

- Den End-Sektor der zweiten Partition und die Block-Grösse notieren (in diesem Fall: Sectors of 1*512, und 4227071) Anhand dieser Daten kann man schliessen, dass die zweite Partition am 4227071*512 ten Byte der Datei endet. Das heisst genau da koennen wir die Datei abschneiden. Da die Block-Zaehlung bei 0 beginnt, muessen wir noch eine 1 dazurechnen. somit ergibt sich zum abschneiden folgendes Kommando:
- `sudo truncate -size=$((4227071+1)*512) fwei4pi.img`
- Nun kann das Image noch gezippt werden
- `zip fwei4pi.zip fwei4pi.img`
- Habe fertig! :o)

4 Quellen, Ideen, Anregungen (Auszug)

1 2 3

this L^AT_EX2 file was written by

$\delta\alpha\chi\upsilon\phi$