

EFA LIVE AUF DEM RASPBERRY PI

Inhaltsverzeichnis

EFA LIVE AUF DEM RASPBERRY PI.....	1
Efalive auf dem Raspberry Pi.....	2
Vorwort.....	2
Image erzeugen.....	2
Image nutzen.....	4
Basis-Konfiguration.....	4
Konfiguration von Efa.....	7
Abschluss.....	11
Bei Bedarf: Änderung des Autologin.....	12
Bei Bedarf: Überwachung des HDMI-Schalters.....	13

Efalive auf dem Raspberry Pi

Vorwort

Efa benötigt nicht viel Rechenleistung und damit auch nicht unbedingt einen „richtigen“ Rechner. Schon um die Welt nicht mit mehr CO2 zu belasten, ist ein Umstieg auf den Raspberry Pi sinnvoll. So ein Gerät kostet etwa 60€ und rentiert sich schon durch die Stromrechnung.

Raspberry bietet als Betriebssystem Raspbian an. Das ist ein Derivat von Debian, dem Betriebssystem, auf welchem Efalive fußt. Daher ist es einfach, Efalive auch auf dem Raspberry laufen zu lassen. Efalive bringt das Setup-Tool mit, welches die Verwaltung des Rechners vereinfacht – auch das ein Grund efalive zu verwenden.

Bleiben nur noch die kleinen Tücken des Systems: Der Pi hat keine Uhr. Das kann man, wie von Ruderkameraden am Förggensee demonstriert, mit Hardware lösen. Der Ansatz hier ist: der Raspberry hat Internet. Das kann er dann gleich auch für das Backup per Mail nutzen, denn so ein Raspberry verwendet MikroSD Karten als „Festplatte“ und die sind u. U. nicht so zuverlässig. Tägliches Backup empfiehlt sich.

Das Image, was wir gebaut haben, setzt darauf, dass der Internet Zugang verfügbar ist. Eingesetzt wird es mit LAN (nicht WLAN). Prinzipiell müsste es aber auch offline verwendbar sein. Die einzige Besonderheit im Vergleich zu efaLive ist ein zusätzlicher Service, der das HDMI Interface abschaltet, wenn der Bildschirm dunkel wird. Das geht natürlich auch offline.

Das Image kann man sich selbst erzeugen, mit Hilfe von zwei Hilfsskripten. Wer es direkt herunterlädt, kann das nächste Kapitel überspringen.

Image erzeugen

Ein efaPi Image werden wir wohl zum Download bereithalten. Das ist dann relativ groß. Kompakter ist es, ein Raspbian image zu laden und den efaPi selbst aufzubauen. Die nötigen Schritte sind größtenteils automatisiert.

Die Micro-SD Karte in einen Kartenleser stecken, mit

```
df
```

das Device identifizieren (in der Regel sdb, sdc oder sdd). Das Raspbian image, hier Jessie-lite, mit

```
sudo dd if=/yourImagePath/2016-02-09-raspbian-jessie-lite.img of=/dev/sdd
```

auf die MicroSD Karte kopieren. (Dauer: gut 15 Minuten). Die Karte wird in den Raspberry gesteckt und dann wird der raspberry gebootet.

Der Login ist „pi“, das Kennwort „raspberry“. Achtung: US-amerikanisches Tastaturlayout hat im Gegensatz zum deutschen y<>z vertauscht. '-' liegt auf dem 'ß', der '/' auf dem '-'. ':' ist <Shift>-Ö, ^ ist <shift>-'6', '*' ist <shift>-'8', '>' ist <shift>-'.'

Für die Installation sind die Skripte hier vorbereitet: <http://tfyh.org/efaPi>. Diese lädt man mit:

```
sudo apt-get install curl"  
curl -O "http://tfyh.org/efaPi/install.sh"  
chmod +x install.sh
```

Danach führt man das Installationskript, Teil 1, aus mit

```
./install.sh
```

Das Skript lädt die restlichen Dateien, kopiert sie an die richtige Stelle und erweitert das Dateisystem mittels „fdisk“, weil die 1,5 GB des Original Images zu klein sind.

Wenn fdisk gestartet ist, bitte **p** eingeben, um eine Liste aller Partitionen zu erhalten. Bei einer frischen Raspbian Installation sieht die Ausgabe wie in Listing 2 aus.

Device	Boot	Start	End	Blocks	Id	System
/dev/mmcblk0p1		8192	131071	57344	c	W95 FAT32 (LBA)
/dev/mmcblk0p2		131072	5785599	2831360	83	Linux

In der ersten Zeile wird die Boot-Partition angezeigt. Das ist diejenige, die wir auch sehen, wenn wir die SD-Karte unter Windows öffnen. Diese Partition soll intakt bleiben. Aber Nummer zwei geht es an den Kragen. Genauer gesagt muss die Partition gelöscht werden. Dabei gehen keine Daten verloren. Vorher aber bitte den Startsektor dieser Partition notieren. In dem Beispiel aus Listing 2 ist das der Wert 131072. Hier aber auf jeden Fall den Wert aus eurer Ausgabe notieren! Anschließend können wir mit dem Kommando **d** für delete, gefolgt von der Nummer der Partition, in diesem Fall also **2**, die Partition löschen.

Jetzt brauchen wir eine neue, zweite Partition. Das gelingt über die Kommandos **n** für new, **p** für primär und dann die Nummer **2**. Damit wird eine neue, primäre Partition mit der Nummer zwei angelegt. Das Programm fdisk fragt noch den Start- und den Endsektor ab, um die neue Partition anlegen zu können. Bei Start ist zwingend der oben notierte Wert einzugeben. Auf keinen Fall den Standard übernehmen, da es ansonsten mit der korrekten Zuordnung der Partitionen vorbei ist. Als Endsektor einfach den Standard übernehmen.

Beenden mit **w**. Die Fehlermeldung ignorieren, weil eh neu gebootet wird.

Im Raspbian waren es 131072 Blöcke für die boot Partition und 5.000.000 Blöcke (2,3 GB root Partition) haben für die efaLive Installation gereicht.

Mit dem Aufruf von fdisk endet das erste Skript. Danach

```
sudo reboot
```

Anschließend sich wieder als User pi einloggen und ds Skript

```
./continue.sh
```

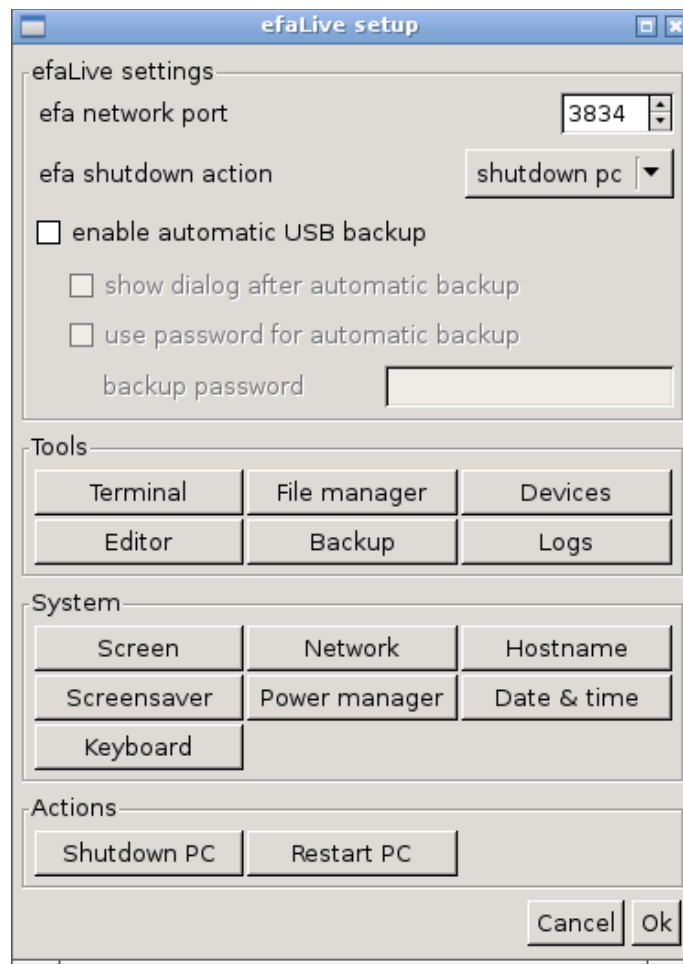
ausführen. Das Skript installiert den lightdm Display Manager, efa2 und efaLive und aktualisiert alle Pakete. Deswegen dauert es eine Weile.

Für efaLive muss das Debian Paket eingebunden werden. Dazu muss die sources.list ergänzt werden um.

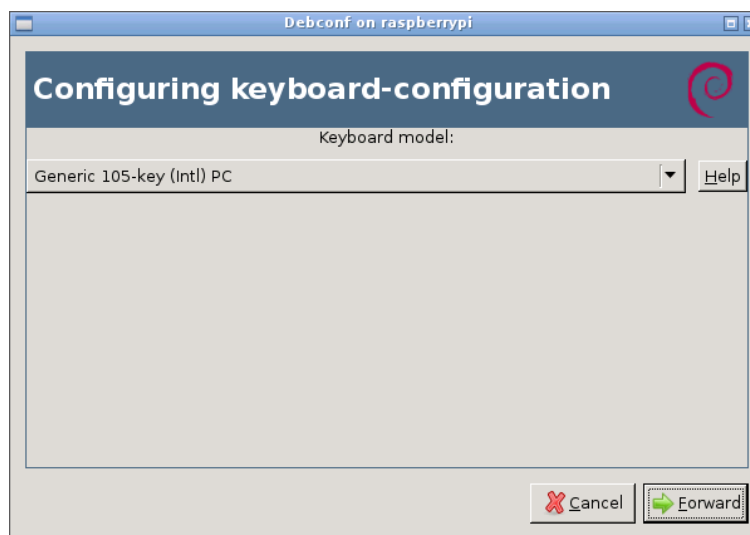
```
deb http://efalive.hannay.de/debian jessie main
```

Install.sh erledigt das durch überkopieren der /etc/apt/sources.list, legt für alle Fälle aber eine Sicherheitskopie an.

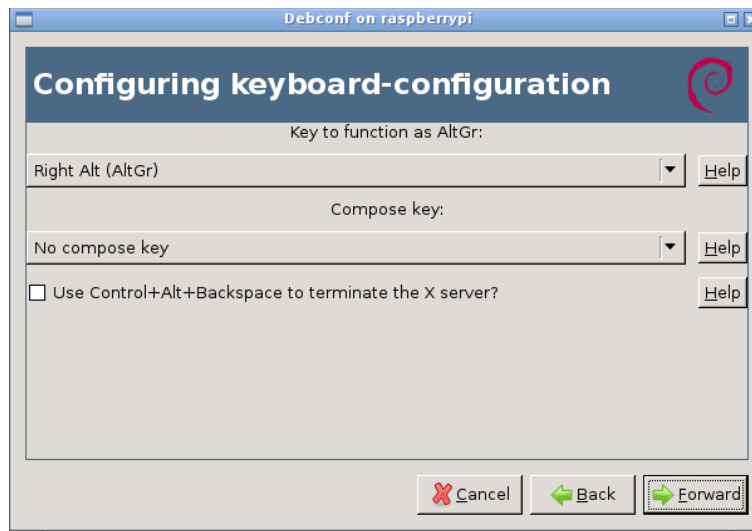
Das efaLive Paket ist nicht verifiziert. Den GPG Schlüssel Fehler bitte ignorieren. Die Installation lädt gut 500 Pakete und dauert deswegen 20-30 Minuten. Durch alles weitere führt das Installationskript.



Als erstes wird das Tasturlayout angepasst (Knopf: Keyboard in der mittleren Knopfgruppe unten links). Es erscheint der Debian Keyboard Konfigurator:



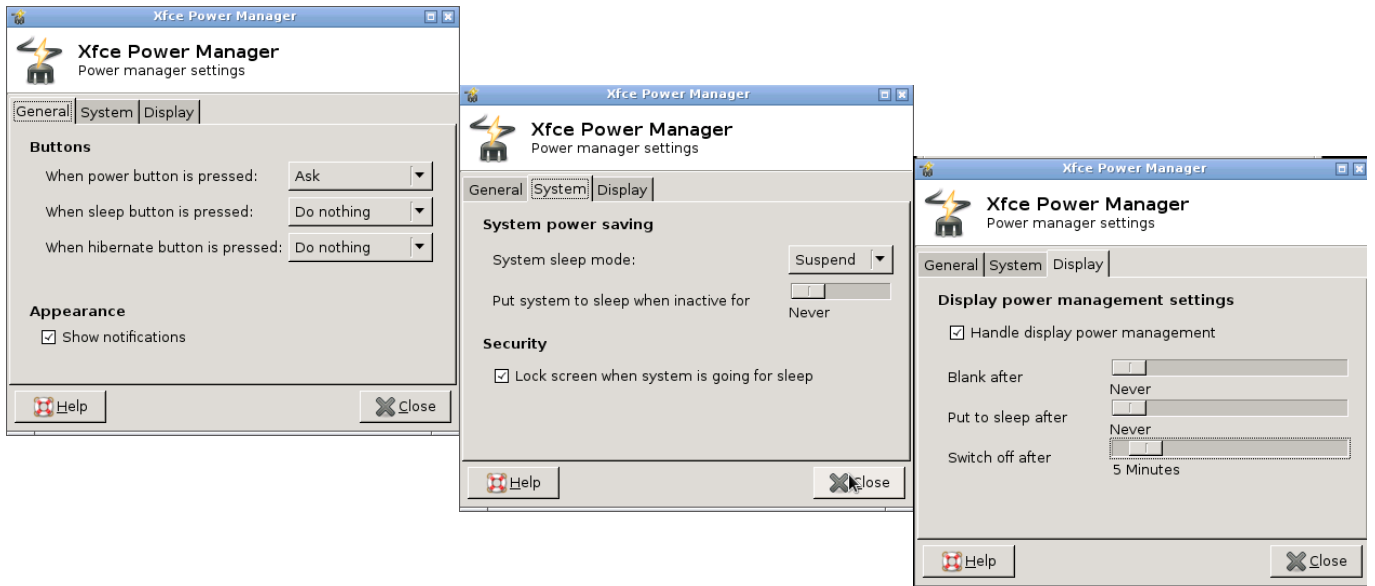
„Other“ wählen, dann „Forward“. Danach ist die Auswahlliste größer, man kann „German“ auswählen. Das muss man nochmal bestätigen, oder anpassen, wenn man in Österreich rudert.



Alle weiteren Einstellungen lässt man auf dem Standardwert.

Danach kommt der Power-Manager. Das ist tatsächlich elementar, denn der Raspberry schaltet von sich aus die HDMI Schnittstelle nicht ab. Deswegen gibt es ein kleines Skript im Image, welches immer dann, wenn der Bildschirm schwarz wird, auch die HDMI-Schnittstelle stromlos setzt. Nur so spart man Strom im Bildschirm.

Damit er auch wieder angeht, darf allerdings der Raspberry nicht in irgendeinen Ruhezustand versetzt werden. Braucht er auch nicht, denn er braucht eh nur 2-3 W und ist auf Durchlaufen angelegt.



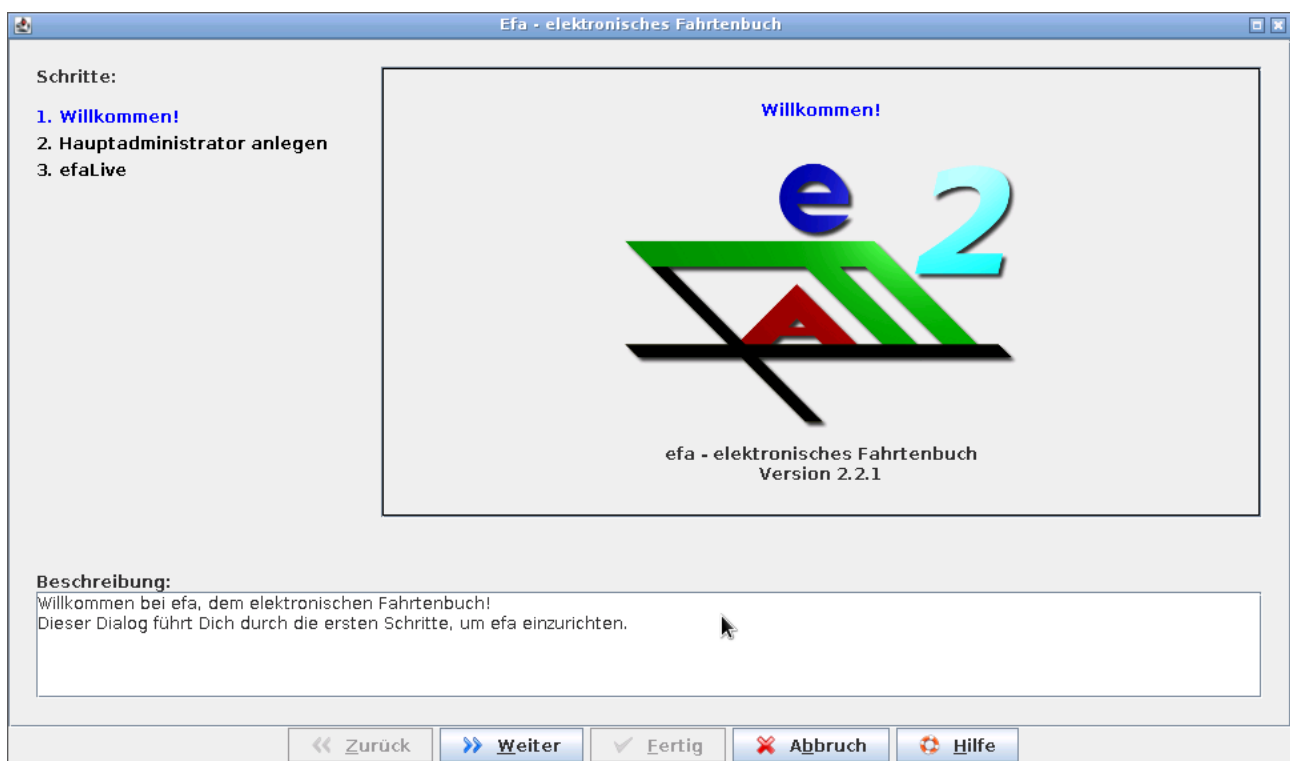
Man kann natürlich statt der 5 Minuten aus dem Beispiel andere Werte nehmen, aber alle anderen bitte auf „Never“ setzen. Im Test hat sich das Intervall manchmal auf den Standardwert von 15 Minuten zurückgesetzt. Gegebenenfalls muss man es halt wieder anpassen.

Konfiguration von Efa

Jetzt schließt man den Basis-Konfigurator mit OK ab und schon startet Efa das erste Mal.



Der erste Schritt ist die Spracheinstellung. Jetzt werden die Dialoge auch deutsch. Efa begrüßt und man richtet den Efa Admin ein.



Zur Erinnerung: Das Linux Betriebssystem hat einen Nutzer efa, das ist Betriebssystem-Ebene. Das Programm Efa hat einen Nutzer Admin, der gilt nur innerhalb des efa-Programmes. Kann man schon mal verwechseln.

Efa - elektronisches Fahrtenbuch

Schritte:

1. Willkommen!
2. **Hauptadministrator anlegen**
3. efaLive

Hauptadministrator anlegen

Neuer Hauptadministrator

Name:

Paßwort:

Paßwort (Wiederholung):

Admin 'efalive' erstellen

Beschreibung:

Alle Administrationsaufgaben in efa erfordern Administratorrechte.
Bitte lege ein Paßwort (mindestens 6 Zeichen) für den Hauptadministrator 'admin' fest.

Das war es dann auch schon. Einfach mit „Fertig“ abschließen.

Efa - elektronisches Fahrtenbuch

Schritte:

1. Willkommen!
2. Hauptadministrator anlegen
3. **efaLive**

efaLive

Beschreibung:

Und die Meldung quittieren:

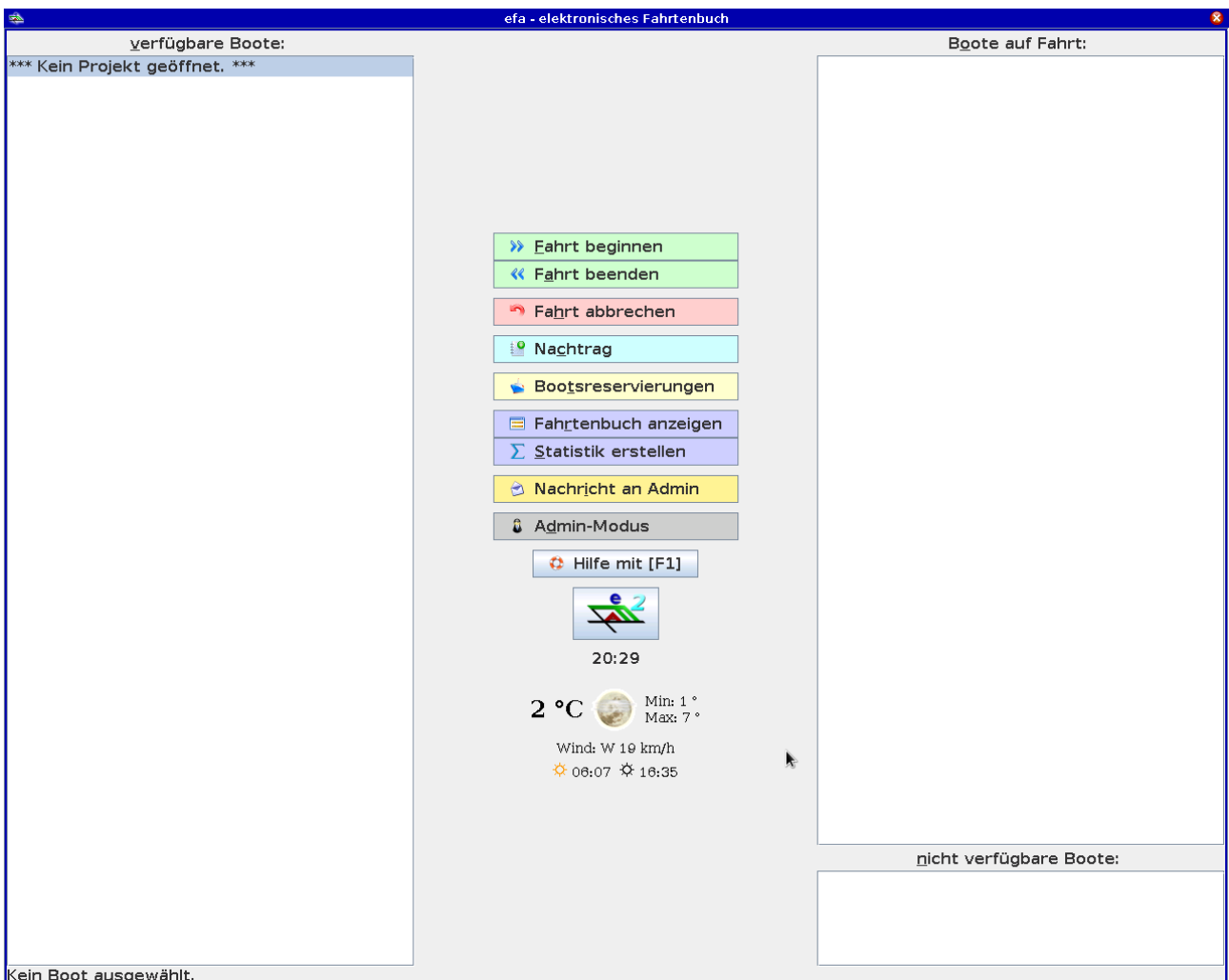


SD-Karte erweitert nach „<http://www.fabiandeteilhoff.de/2014/07/raspberry-pi-speicherplatz-der-sd-karte-ausnutzen/>“

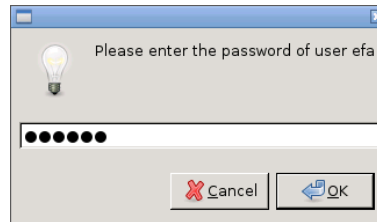
Dann ist es geschafft. Efa startet



Und zeigt den üblichen Fahrtenbuchbildschirm:



Den Efalive Konfigurator öffnet man mit <Ctrl>-F12:



Und nicht vergessen: Hier ist das Kennwort des Betriebssystemnutzers efa gefragt, also efa2pi.

Wer noch prüfen will, ob der HDMI Ein-Ausschalter läuft, kann das im Terminal:

```
efa@raspberrypi: ~  
efa@raspberrypi:~$ ps ax | grep hdmi  
389 ? S 0:01 /bin/bash /usr/sbin/hdmiOnOff  
5125 pts/0 S+ 0:00 grep --color=auto hdmi  
efa@raspberrypi:~$ ps ax | grep monitor  
581 ? S 0:01 /bin/bash /home/efa/monitorStatus.sh  
5141 pts/0 S+ 0:00 grep --color=auto monitor  
efa@raspberrypi:~$
```

Und jetzt spätestens ist es angemessen die Kennwörter des Nutzers efa und des Nutzers pi zu ändern. Efa geht direkt, man muss noch wissen, dass im Standard des PI images der Nutzer das Kennwort 'efa2pi' hat:

```
efa@raspberrypi: ~  
efa@raspberrypi:~$ passwd  
Changing password for efa.  
(current) UNIX password:  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully  
efa@raspberrypi:~$
```

Für den Nutzer pi muss man mit „su pi“ den Nutzer im Terminal wechseln und dann geht es genauso, Standard ist Kennwort 'Pi4Efa' für den Nutzer pi.

Zu diesem Zeitpunkt ist die Sprache u. U. noch Englisch und die angenommene Tastatur ebenfalls. (Tastenbelegung siehe Abschnitt „Image Erzeugen“, am besten ausprobieren).

Einrichten der richtigen Sprache mit z. B.

```
update-locale de_DE.UTF-8
```

Und Einstellen der Zeitzone mit:

```
dpkg-reconfigure tzdata
```

Abschluss

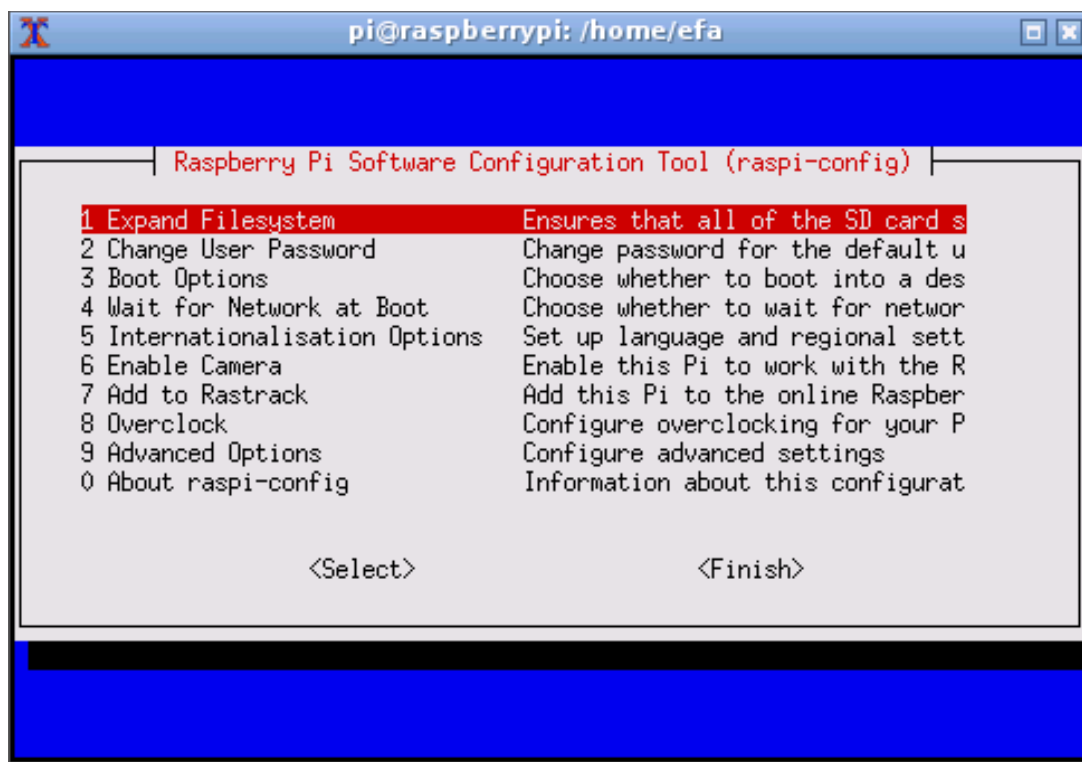
Eines haben wir noch unterschlagen für Nutzer des Basis-Images: Das Image ist nur 2,5 GB groß. Damit geht es auf 4G und 8G und größere Karten. Im Standard wird zurzeit der Raspberry mit 8GB Karten ausgeliefert.

Damit man den Platz auf der Karte auch ausnutzen kann, muss man im Terminal, als User pi die raspberry Konfiguration öffnen:

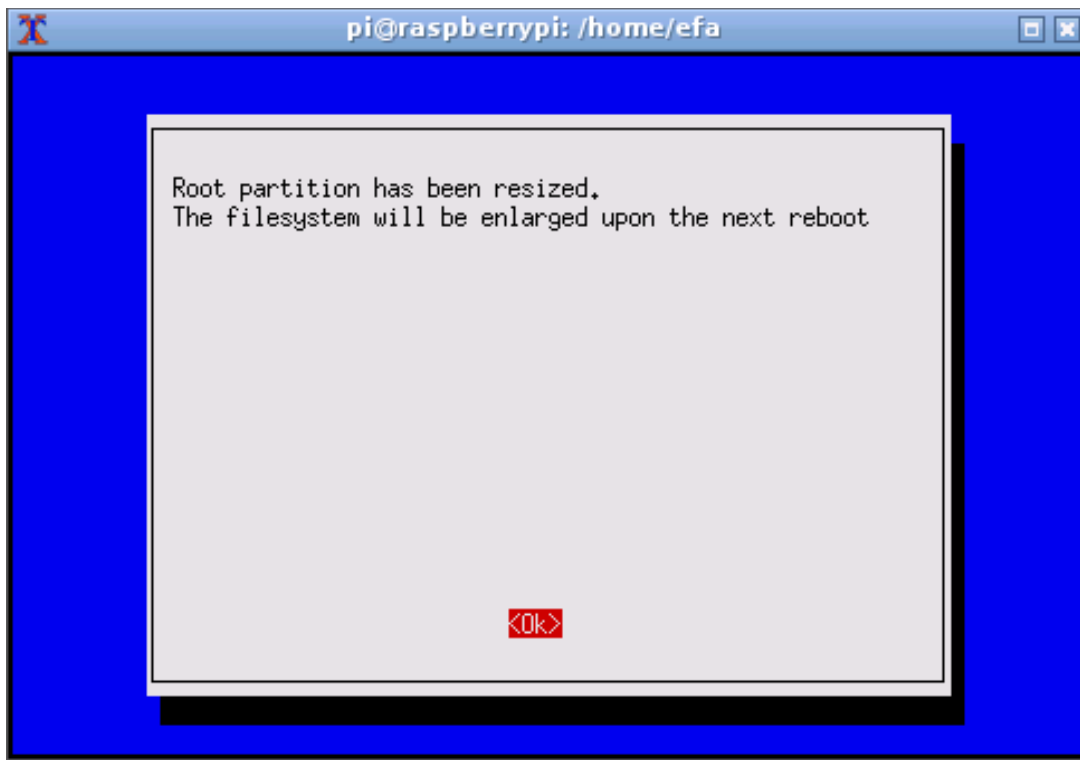
```
sudo pi
```

```
sudo raspi-config
```

Und dann öffnet sich dieses Auswahlmenü



Mit der Tabulatortaste kommt man zu „Finish“, mit Enter geht es weiter:



Und ein sudo reboot schließt das Ganze ab.

Bei Bedarf: Änderung des Autologin

Es gibt zwei Benutzer im Betriebssystem: #1 ist pi, Kennwort Pi4Efa, mit „Root-Rechten“ und #2 ist efa, Kennwort „efa2pi“ ohne eine solche Berechtigung. Anpassungen am Betriebssystem erfordern Linux Kenntnisse. Wer das nicht aus der Efa Oberfläche heraus machen will muss den autologin umkonfigurieren im „Terminal“.

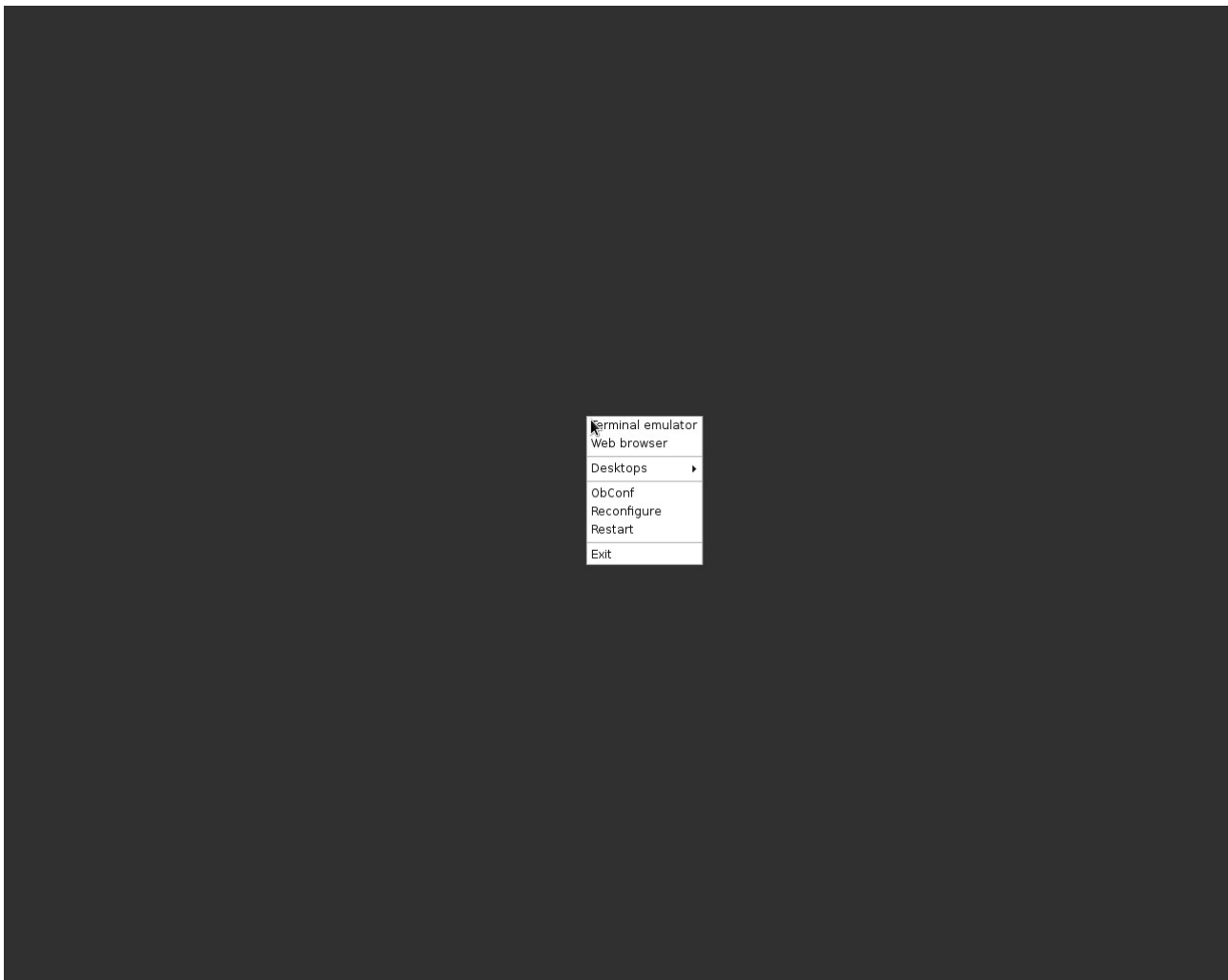
Im Terminal öffnet man den Editor (nano) und editiert die Datei /etc/lightdm/lightdm.conf, wechselt dort etwa in Zeile 120 den Eintrag autologin=efa gegen autologin=pi und speichert mit ^O, dann beendet man mit ^X. Im Terminal tippt man „sudo reboot“ und schon geht es mit pi los.

A terminal window titled 'pi@raspberrypi: ~' showing the nano text editor. The editor is editing the file '/etc/lightdm/lightdm.conf'. The visible content includes configuration options for lightdm, such as session wrapper, greeter wrapper, and autologin settings. The current line is 'autologin-user=efa'. The bottom of the screen shows nano editor shortcuts like '^G Get Help', '^O WriteOut', '^R Read File', '^Y Prev Page', '^K Cut Text', '^C Cur Pos', '^X Exit', '^J Justify', '^W Where Is', '^V Next Page', '^U UnCut Text', and '^T To Spell'.

```
pi@raspberrypi: ~
GNU nano 2.2.6 File: /etc/lightdm/lightdm.conf Modified
#guest-session=
#session-wrapper=lightdm-session
#greeter-wrapper=
#guest-wrapper=
#display-setup-script=
#display-stopped-script=
#greeter-setup-script=
#session-setup-script=
#session-cleanup-script=
#autologin-guest=false
autologin-user=efa
#autologin-user-timeout=0
#autologin-in-background=false
#autologin-session=UNIMPLEMENTED
#exit-on-failure=false

#
# Seat configuration
#
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Dabei ist zu beachten, dass der Nutzer pi keine Oberfläche voreingestellt hat. Nach dem Reboot sieht man also einen leeren Bildschirm. Mit der rechten Maustaste kommt das verfügbare Menü:



Bei Bedarf: Überwachung des HDMI-Schalters

Wer den HDMI Schalter überwachen möchte hat folgende Möglichkeiten:

Prüfen der Datei `/var/run/hdmiOnOff/service.log`. Hier wird protokolliert, wann der Service gestartet wurde, und wann er gestoppt wurde. Die Datei wird mit jedem Reboot gelöscht.

Prüfen der Datei `/var/run/hdmiOnOff/hdmiOnOff.log`. Hier wird protokolliert, wann der Monitor an und ausgestellt wurde. Die Datei wird mit jedem Reboot gelöscht.

Prüfen der Datei `/var/run/hdmiOnOff/monitorStatus.txt`. Hier wird der aktuelle Monitorstatus vom User für den Service hinterlegt. Der Service hat keinen Zugriff zum `xset` des Users, sondern liest diese Datei aus. Sie wird mit dem Kommando „`xset q | grep Monitor`“ aus dem User Kontext heraus geschrieben. Die Datei wird mit jedem Reboot gelöscht.

Ein Post ins Internet. Hierzu muss per nano in der Datei /usr/sbin/hdmiOnOff die „PostUrl“ eingetragen werden und die Zeile „postUrl=""“ gelöscht oder auskommentiert werden.

```

pi@raspberrypi: /home/efa
GNU nano 2.2.6 File: /usr/sbin/hdmiOnOff Modified
#!/bin/bash
# Author: Martin Glade <tfyh.org>
# Reads the monitor status via "xset q"
# Switches the HDMI interface according to the changes in the monitor status on$

rpiHdmiSwitch=/usr/sbin/rpiHdmiSwitch.sh
monitorStatusFile=/var/run/hdmiOnOff/monitorStatus.txt
LogFile=/var/run/hdmiOnOff/hdmiOnOff.log

# the script can post messages via curl when switching the monitor off or on.
# the post will provide monitor, hdmi, sender, and message variables to the pos$
# Edit some php page at the url target to read the post.
# set postUrl="" to ignore.
sender=efaPi
postUrl="http://www.tfyh.org/Efa-Monitorstatuslog/PostStatus.php"
postUrl=""

echo "Monitor is On" > $monitorStatusFile
chmod 666 $monitorStatusFile

[ Read 86 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Serverseitig schreibt dann zum Beispiel eine PostStatus.php Datei den Post weg. Ein Beispielcode:

```

<html>
<head>
<title>Efa-Monitorstatuslog</title>
</head>
<body>
<?php
$file = 'status.log';
$logMessage = date('l jS \of F Y h:i:s A') . "\nMonitor: " .
$_POST['monitor'] . ", HDMI: " . $_POST['hdmi'] . ", Sender: " .
$_POST['sender'] . ", Message: " . $_POST['message'] . "\n";
$log = file_get_contents($file);
$log .= $logMessage;
file_put_contents($file, $log); ?>
</body>
</html>

```

erzeugt einen Status.log der Form:

```

Wednesday 24th of February 2016 07:24:30 AM
Monitor: X, HDMI: X, Sender: efaPi, Message: turningHdmiOff
Wednesday 24th of February 2016 07:25:02 AM
Monitor: X, HDMI: X, Sender: efaPi, Message: turningHdmiOn
Wednesday 24th of February 2016 07:26:04 AM
Monitor: X, HDMI: X, Sender: efaPi, Message: turningHdmiOff
Wednesday 24th of February 2016 07:27:22 AM
Monitor: X, HDMI: X, Sender: efaPi, Message: turningHdmiOn
Wednesday 24th of February 2016 07:28:24 AM
Monitor: X, HDMI: X, Sender: efaPi, Message: turningHdmiOff
Wednesday 24th of February 2016 06:22:43 PM
Monitor: X, HDMI: X, Sender: efaPi, Message: turningHdmiOn

```

Wednesday 24th of February 2016 06:37:45 PM
Monitor: X, HDMI: X, Sender: efaPi, Message: turningHdmiOff
Wednesday 24th of February 2016 07:37:45 PM
Monitor: X, HDMI: X, Sender: efaPi, Message: turningHdmiOn
Wednesday 24th of February 2016 07:52:47 PM
Monitor: X, HDMI: X, Sender: efaPi, Message: turningHdmiOff