TimeCapsule mit dem Raspberry Pi erstellen.

Beitrag von „Dr. Ukeman“ vom 30. November 2012, 12:40

Da wir ja alle Freunde von Computerhardware sind auf denen "Obst" abgebildet ist hier mal mein Beitrag dazu.
Diesmal soll es ausnahmsweise nur indirekt um angebissene Äpfel gehen, sondern um Himbeeren.
Und zwar in Form des Einplatinen Computers Raspberry Pi.
Damit erstellen wir heute eine TimeCapsule zum sichern unserer geliebten Hackis.

Das ganze ist selbstverständlich auch auf andere Linux Rechner/distributionen übertragbar.

Ihr braucht:

- Einen Raspberry Pi
- Eine Externe Platte für die TimeMachine Backups
- Eine SD Karte mit dem Raspbian Betriebsystem

1. Das Raspbian Image auf die SD Karte schreiben:

Dazu öffnet ihr das Diskutility und klickt bei der SD Card auf Info

In meine Fall hat die SD Card die Bezeichnung *disk4*

Jetzt wählt Ihr die aktive Partition der SD Karte aus und klickt oben auf Deaktivieren.

Nun wird etwas Linux typischer ... Wir öffnen ein Terminal und führen folgenden Befehl aus:

Code

1. sudo dd if=Ordner_in_dem_das_Image_liegt.img of=/dev/diskn bs=1m

Bei "diskn" müsst ihr jetzt die Nummer angeben die Ihr vorher im Diskutility ausgelesen habt.
Bei mir sieht das dann wie folgt aus:

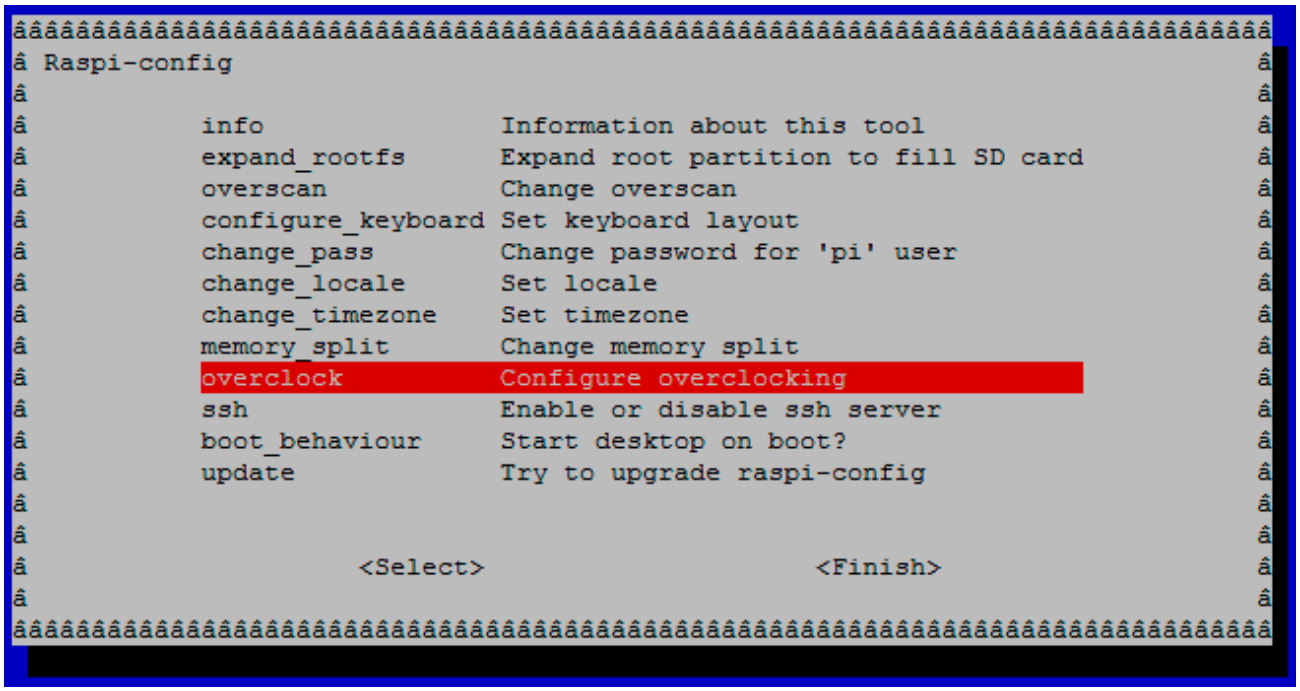
Passwort eingeben, Enter drücken und warten.

Damit sollte der Raspi dann booten können.

2. Den Raspi booten und einrichten

2.1 Die ersten Schritte

Beim ersten boot startet der pi automatisch in das Tool raspi-config in dem die Grundeinstellungen vorgenommen werden können.



Ich habe die Punkte der Reihe nach abgearbetet:

- expand rootfs: erweitert das Filesystem, so dass die komplette SD Karte verwendet werden kann.
- configure keybord: Legt das Layout der Tastatur fest.
- change pass: hier kann man das Passwort des Standardnutzers "pi" ändern
- change locale: setzt das Betriebsystem auf Deutsch
- change Timezone: hier sollte Berlin passen
- memory split: legt fest wieviel Speicher an die CPU und wieviel an die GPU gehen. Hier sollte 16 oder 32 passen als Server.
- overclock: übertaktet den Raspberry auf Maximal 1Ghz und macht ihn spürbar flotter
- SSH: sollte man aktivieren so dass man auch vom Mac den raspi managen kann

Dann das ganze mit finish abschließen und rebooten.

2.2 Aktualisieren des pi

Beim nächsten start landen wir auf dem Konsolenlogin an dem wir uns mit pi und dem Passwort dass oben gesetzt wurde anmelden.
(Das Standardpasswort falls man es nicht geändert hat ist: "raspberry" ist die Tastatur noch english dann "raspberrz")

Alternativ kann man sich jetzt auch vom Mac aus per Terminal anmelden mittels

Code

1. ssh pi@IPADRESSE_DES_PI

Um die benötigten Rechte für alle Aktionen zu haben wechseln wir zum benutzer Root:

ACHTUNG: der root user kann alles und darf alles, und sollte entsprechend mit Respekt behandelt werden, da man auch die Rechte hat die Kiste in den Sand zu setzen und von vorne anzufangen!!

Code

```
1. sudo su -
```

Jetzt bringen wir ersteinmal die Kiste auf den aktuellen Stand.
Dazu nutzen wir den Befehl

Code

```
1. apt-get update
```

und danach

Code

```
1. apt-get upgrade
```

Das dauert ein paar Minuten also holt euch nen Kaffee oder geht eine rauchen 👍
Wenn der Pi damit durch ist habt ihr die richtige Basis geschaffen und es kann mit der eigentlichen Konfiguration losgehen.

2.3 HFS+ einrichten
Ein echtes TimeMachine Backup fühlt sich nur auf dem Apple eigenen Format richtig wohl.
Um dem Raspberry das beizubringen werden ein paar Pakete benötigt, wir mit dem Befehl

Code

1. apt-get install hfsplus hfsutils hfsprogs

herunterladen und installieren.

Jetzt erstellen wir einen Mountpoint mit

Code

1. mkdir /mnt/TimeMachine

und lassen uns von Raspbian sagen wie unsere Platte heisst mit dem Befehl

Code

1. blkid

In meinem Fall heisst die Platte "sda1". Diese weisen wir jetzt dem erstellten Mountpoint zu

Code

1. mount -o force /dev/sda1 /mnt/TimeMachine

Sollte das nicht klappen, dann führt (Achtung Plattenbezeichnung anpassen)

Code

1. fsck.hfsplus /dev/sda1

aus und versucht es danach mit

Code

1. mount -t hfsplus -o force,rw /dev/sda1 /mnt/TimeMachine

Da wir ja auch auf das Laufwerk schreiben wollen sollten wir noch die Rechte setzen.

Code

```
1. chmod 777 /mnt/TimeMachine
```

Damit haben alle Nutzer alle Rechte auf diesem Ordner.

und machen den Nutzer Pi noch zum Eigentümer des Ordners

Code

```
1. chown -R pi.users /mnt/TimeMachine
```

Das wars auch schon schon mit HFS+!

Wir haben nun ein gemountetes HFS+ Laufwerk auf einem aktualisierten Raspbian.
Das müssen wir allerdings den Macs im Netzwerk auch noch mitteilen 

2.4 AFP mit Hilfe von Avahi und netatalk

Dazu benötigen wir erst einmal wieder ein paar Pakete, die wir uns mit

Code

```
1. apt-get install avahi-daemon libavahi-client-dev libdb5.3-dev db-util db5.3-util libgcrypt11 libgcrypt11-dev
```

besorgen und installieren.

Der nächste Schritt ist am Terminal etwas zu fummelig. Darum öffnen wir eine X Session oder begeben uns direkt an den Raspi und führen dort im Terminal den Befehl

Code

```
1. startx
```

aus.

Terminal Puristen klicken hier

Spoiler anzeigen

Damit landen wir auf dem Desktop von Raspbian.
Mit dem Browser Midori laden wir jetzt die neusten Sourcen von "netatalk" runter.

Den Download finden wir dann im /home/pi Verzeichnis, wo wir die Sourcen entpacken.

Bevor uns das alles "zu bunt" wird melden wir die X Session wieder ab und es geht weiter auf dem Terminal.

Wir wechseln in den Ordner /home/pi/netatalk und führen

Code

```
1. ./configure --enable-debian --enable-zeroconf --with-init-style=debian-sysv
```

aus.

Das sollte ohne Fehler durchlaufen und dann starten wir das Kompilen mit dem Aufruf

Code

```
1. make
```

Dafür genehmigt sich der Pi schon ein paar Minuten (ca.20). Ihr habt also Zeit nochmal eure Kaffeemaschine zu besuchen.

Nach erfolgreichem kompilieren installieren wir das ganze mit

Code

```
1. make install
```

Ab Version 3.0 von Netatalk spielt sich die gesamte Konfiguration in einer einzigen Datei ab.
Sie heisst "afp.conf" und befindet sich im Ordner /usr/local/etc/afp.conf
Wer mehr über die Datei erfahren möchte wird hier fündig (allerdings nur in englisch)

Um sie zu bearbeiten öffnen wir sie mit dem Editor "nano"

Code

```
1. nano /usr/local/etc/afp.conf
```

und passen sie für unsere Bedürfnisse an

Code

```
1. ;
2. ; Netatalk 3.x configuration file
3. ;
4.
5.
6. [Global]
7. ; Global server settings
8. uam list = uams_guest.so
9. zeroconf = yes
10. save password = no
```

11.
12.
13. [Homes]
14. basedir regex = /home
15.
16.
17. [TM]
18. path = /mnt/TimeMachine
19. time machine = yes

Alles anzeigen

Speichern mit strg + o und beenden mit strg + x.

Der service sollte einmal neu gestartet werden mit

Code

1. service netatalk restart

Um den Service nach einem Reboot automatisch zu starten bearbeiten wir noch die Datei "/etc/rc.local"

Code

1. nano /etc/rc.local

Und fügen über der Zeile *exit 0* folgendes ein:

Code

1. service netatalk restart

und dann sollte am Mac im Finder der Raspi als Server auftauchen.

Um dem ganzen noch den Feinschliff zu verpassen können wir jetzt den Raspi noch als

TimeCapsule im Finder erscheinen lassen.
Dazu muss in der afp.conf noch der Entrag

Code

1. mimic model = TimeCapsule6,106

eingetragen werden

und wir erhalten letztendlich

Solltet ihr den Raspi ausschalten dann fährt ihn am besten per Terminal herunter, da sonst das nächste mal dasHFS Laufwerk Read Only gemountet wird.
Das geht mit dem Befehl:

Code

1. sudo shutdown -h now

Ein Dank geht noch an Gnarz und iphone_4s fürs Testen und Fehler suchen 

Edit 10.11.2013: Links zu Netatalk auf die aktuelle Version 3.1.0 aktualisiert