

OSCAM op een BeagleBone Black.

Voor rond 48 Euro koop je tegenwoordig een Beaglebone Black (BBB). Een embedded computertje met vrijwel alles wat nodig is om een OSCAM server te draaien. Hij bezit een iets krachtiger cpu (TI AM3358 ARM Cortex-A8) dan de Raspberry Pi en dat laatste maakt het nou net mogelijk om zonder al te veel problemen UBUNTU te kunnen draaien, iets wat op de RPI waarschijnlijk niet zal lukken. Grafisch is ie wat minder dan de RPI en dat is dan ook de reden dat b.v. XBMC er (nog) niet goed op draait. Voor onze OSCAM-installatie vormt dit gemis aan grafische capaciteit natuurlijk geen enkele belemmering.

Met zijn stroomverbruik van ongeveer 2 watt zal niemand meer echt wakker liggen van een server die 24 uur per dag volcontinu staat te draaien.

Als je meerdere smargo's wilt aansluiten, heb je wel wat extra usbpoortjes nodig. Neem dan in ieder geval een powered hubje. Smargo's lusten nogal wat power en die kun je niet grenzeloos van je beagle-tje halen.

Het geheel is krachtig genoeg om er b.v. Ubuntu 13.10 op te draaien. Natuurlijk zijn er andere linux-smaken. Standaard wordt hij uitgeleverd met Angstrom-linux op het EMMC geheugen van 2GB (ingebouwde flashgeheugen van de BBB) . Leuk om mee te beginnen, maar heeft als nadeel dat het booten van een (niet bijgeleverde) micro sd-card vereist dat er telkens een knopje op het board dient te worden ingedrukt en dat wil je voor een OSCAM server absoluut niet. Bij een eventuele stroomuitval moet alles natuurlijk weer vanzelf in de benen komen. Door simpelweg ook een UBUNTU image op het ingebouwde EMMC geheugen te plaatsen is het hinderlijke ingedrukt houden van de Boot/USER knop definitief voorbij en wordt standaard geboot van de micro sd-card, als die is geladen.

We installeren achtereenvolgens via een Windows pc eerst UBUNTU 13.10 Saucy op een micro-sd card van minimaal 8 GB. Vervolgens installeren we diezelfde versie van UBUNTU op het EMMC geheugen . Dan halen we de laatste OSCAM source uit de TRUNK en compileren deze naar een executeerbaar geheel. Als alles goed is gegaan houd je een geheel operationele oscam server over voor slechts heel weinig euro's.

Hieronder de stappen om vanuit een maagdelijke BeagleBone Black een OSCAM-server met 1 of meerdere smargo's onder UBUNTU 13.10 (saucy) te creëren:

Noodzakelijke hardware:

Een BeagleBone Black, een micro-sd card (minimaal class4, minimaal 8 GB) inclusief een microsd-card naar sd –houder, een degelijke voeding van 5 Volt (minimaal 2 amp), minimaal 1 smargo en als je er meer hebt, heb je een powered usb hub nodig.

De Stappen:

1: Download en installeer op een Windows computer Windisk32imager , die vind je hier:

<http://sourceforge.net/projects/win32diskimager/files/latest/download>

2: Als je wat meer informatie wilt over de volgende omschreven stappen raad ik je aan om de volgende link te openen en hier gewoon wat te lezen....

Hier staan voldoende tips en trucs om UBUNTU te installeren. Lees dit stuk ter informatie eens door!

http://elinux.org/Beagleboard:Ubuntu_On_BeagleBone_Black

3: Haal nu, op diezelfde PC het image op voor je micro sd-card van 8 GB

<http://s3.armhf.com/debian/saucy/bone/ubuntu-saucy-13.10-armhf-3.8.13-bone30.img.xz>

4: Haal het programma 7zip op en installeer dit. Het dient om de bovenstaande tarball te kunnen uitpakken:

<http://downloads.sourceforge.net/sevenzzip/7z920.exe>

5: Pak nu de bovenstaande tarball uit in een aparte directory. Je houdt nu de volgende file over:

ubuntu-saucy-13.10-armhf-3.8.13-bone30 .img

6: De micro-sdcard moet fat32 geformatteerd zijn, is dit nog niet het geval, gebruik dan de HP-USB-DISK-STORAGE-FORMAT Tool.

Start nu Win32diskimager, en schrijf bovenstaande image file naar de card. Controleer op evt. foutmeldingen.

7: Insert de beschreven card in de BBB en houdt het Boot/User knopje even ingedrukt tijdens het aansluiten van de voeding. Als het goed is boot de BBB nu met UBUNTU 13.10 van de micro sd-card.

host: ubuntu-armhf, user: ubuntu, pass: ubuntu

8: Je sluit je BBB gewoon via een patchkabeltje aan je netwerk aan en doet alles via een ssh netwerkverbinding. (later kan dat evt ook via wifi) . Als je Ubuntu hebt geboot van je micro sd-card dan krijgt hij in principe een dhcp adres van je router. Om te zien welk adres dit is gebruik ik zelf op mijn android-smartphone de app "sharesfinder". Als je met je phone verbonden bent via wifi en je drukt op scan in genoemde app, geeft hij je alle nodes in je netwerk. Je BBB zal hier, als het goed is, ook bij staan als host "ubuntu-armhf" en dan heb je ook het dhcp-ipadres.

Als je dat hebt ga je naar een Windows-pc en hierop installeer je de terminal emulator "putty". Volgens mij kun je dit vinden op www.putty.org. Pak dit even uit in een directory en start. Vul bij Hostname or ipadres het zojuist gevonden ipadres in en geef deze verbinding in Saved sessions een naam. een ssh verbinding loopt in principe over poort 22, dus daar verander je verder niets aan. Sla op met save voor je gaat connecten. Denk erom:

er wordt dhcp geboot, dus de volgende keer kan het ip-adres anders zijn.

Inloggen op dit terminalscherf doe je met Username: ubuntu en passwd: ubuntu

Als je eenmaal binnen bent, verander je het root password als volgt:

```
sudo su -  
passwd
```

dan 2 keer (geloof ik kan ook 3x zijn!) het nieuwe rootpasswd opgeven . Je kunt nu gewoon in het vervolg een ssh sessie opstarten onder de user root. Als je onder root (superuser) bent ingelogd, ben je al dat gedoe met sudo voor de systeemcommando's kwijt.

Afsluiten van je putty-sessie met ctrl-d of exit in het terminal scherm.

8: De linuxpartitie moet nog ge-resized worden. De micro sd-card heeft namelijk nogal wat ongebruikte ,ongepartitioneerde ruimte, die we aan de linuxpartitie willen toevoegen. Hoe het re-sizen in bij de BBB in zijn werk gaat staat hier heel mooi beschreven:

<http://rayhightower.com/blog/2014/01/06/beaglebone-black-ubuntu-ruby-rails-part-2/>

Na het vergroten van de partitie kun je gemakkelijk het UBUNTU 13.10 image dat bedoeld is voor het 2GB ingebouwde EMMC geheugen van je BBB downloaden en vanuit je micro sd-card installeren:

9: Hiertoe ga je in een browserscherm even naar de volgende site van Robert C Nelson

<http://rcn-ee.net/deb/flasher/saucy/> en zoekt even de laatste beschikbare image.

Copy/paste de koppelingslocatie van deze laatste image in je putty ssh terminalscherm en zet er wget voor:

wget <http://rcn-ee.net/deb/flasher/saucy/BBB-eMMC-flash-ubuntu-13.10-2014-01-17-2gb.img.xz>

installatie:

xz -cd BBB-eMMC-flash-ubuntu-13.10-2014-01-17-2gb.img.xz > /dev/mmcbk1

Let op: wacht zeker een minuut of 10 voordat je het systeem reboot. Het schrijven van 2GB zal echt enige tijd in beslag nemen.

Wanneer het beschrijven van het EMMC geheugen gereed is, kun je de BBB rebooten.

Je boot nu automatisch van de micro sd-card als je deze in de BBB hebt gestoken. Je hebt dus geen gedoe meer met knopjes ingedrukt houden.

Indien je micro sd-card niet is geplaatst, moet je systeem van je EMMC geheugen booten. Controleer dit!

Plaats nu de micro sd-card. Daar gaan we nu OSCAM op installeren.

10: login onder root !!! Dit is belangrijk, anders moet je voor (bijna) ieder van de onderstaande commando's sudo zetten!

11:

cd /etc/network

12: Wijzig het variabele ip-adres in een vast adres

Als voorbeeld werd hier als netwerk 192.168.0.* genomen.

Indien dit bij jou anders is moet je de onderstaande adressen natuurlijk wijzigen!

#start de vi editor en wijzig/voeg toe de volgende regels in plaats van: iface eth0 inet dhcp

vi interfaces

auto eth0

iface eth0 inet **static**

address 192.168.0.202 ##vast ipadres van de node;laatste getal tussen 1 en 254 en moet uniek zijn.

gateway 192.168.0.254 ##adres router

netmask 255.255.255.0 ##in normale lokale huisnetwerken altijd 255.255.255.0

network 192.168.0.0 ##in normale lokale huisnetwerken altijd de laatste 2 getallen een 0

broadcast 192.168.0.255##in normale lokale huisnetwerken altijd het laatste getal een 255

dns-nameservers 192.168.0.254 ##let op altijd dns-serverS ook al is het er maar 1! adres router

en sluit af met <esc : wq!>

13:

reboot

en log opnieuw in via putty (ssh), maar nu op <vast ipadres>

14:

apt-get update

15:

apt-get upgrade

16: Installeer ftp server

apt-get install vsftpd

17: Configureer dit (/etc/vsftpd.conf)

haal met de vi editor de # weg voor de volgende opties:

vi /etc/vsftpd.conf

local_enable=YES

write_enable=YES

en schrijf weg: <esc :wq!>

18:

service vsftpd restart

19:

apt-get install build-essential automake subversion

20:

apt-get install libusb*

21: Haal nu de actuele oscam sourcefiles; hier een beschrijving:

<http://www.streamboard.tv/oscam/wiki/GettingOscam>

Hieronder de benodigde commando's op de BBB:

svn checkout http://www.streamboard.tv/svn/oscam/trunk oscam-svn

cd oscam-svn

make USE_LIBUSB=1 EXTRA_FLAGS="-I/usr/local/include -L/usr/local/lib"

22: Let goed op dat er geen fouten worden gemaakt in de compilatie/make-slag!

Kopieer de gemaakte executable naar /usr/local/bin met als naam oscam:

cd Distribution

cp oscam-1.20-unstable_svn%%%%-arm-linux-gnueabi-hf-libusb /usr/local/bin/oscam

(hier staat %%%% voor de actuele svn)

23: Maak nu het start/stop script: (dmv copy/paste van onderstaande text)

creeer oscam script dmv vi editor:

vi /etc/init.d/oscam

insert de volgende text: <i> en paste begin hier:

```
#!/bin/sh

# Start/stop the OScam daemon.

#

### BEGIN INIT INFO
# Provides:      OScam
# Required-Start:  $syslog $network $pcscd
# Required-Stop:  $syslog $network $pcscd
# Default-Start:  2 3 4 5
# Default-Stop:   0 1 6
# Short-Description: Start and Stop OScam
# Description: OScam init script. This script start and stops OScam.
### END INIT INFO

NAME=oscam

DAEMON=/usr/local/bin/$NAME
CONFIG_DIR=/usr/local/etc
PIDFILE=/var/run/oscam.pid
LOG=/var/log/oscam/oscam.log
LOGUSR=/var/log/oscam/oscamuser.log
LOGCW=/var/log/oscam/cw.log
STARTAS="$DAEMON -- -b -c $CONFIG_DIR"

#test -f $DAEMON || exit 0
[ -x $DAEMON ] || exit 0
[ -d $CONFIG_DIR ] || exit 0

# Get lsb functions
. /lib/lsb/init-functions

echo $NAME

clear_file() {
    # Clear log and pid file if exists
    if [ -e $PIDFILE ]
```

```
then
    rm -f $PIDFILE
fi
if [ -e $LOG ]
then
    rm -f $LOG
fi
if [ -e $LOGUSR ]
then
    rm -f $LOGUSR
fi
if [ -e $LOGCW ]
then
    rm -f $LOGCW
fi
}
case "$1" in
start)
    clear_file
    log_daemon_msg "Starting OScam daemon" "OScam"
    start-stop-daemon --start --quiet --exe $DAEMON --startas $STARTAS
    log_end_msg $?
;;
stop)
    log_daemon_msg "Stopping OScam daemon" "OScam"
    start-stop-daemon --stop --quiet --exe $DAEMON
    log_end_msg $?
;;
restart)
    log_daemon_msg "Restarting OScam daemon" "OScam"
    start-stop-daemon --stop --retry 5 --quiet --exe $DAEMON
    clear_file
    start-stop-daemon --start --quiet --exe $DAEMON --startas $STARTAS
    log_end_msg $?
;;
status)
status_of_proc -p $PIDFILE $DAEMON $NAME && exit 0 || exit $?
;;
```

```
*)
log_action_msg "Usage: /etc/init.d/Oscam {start|stop|restart|status}"
exit 2
;;
esac
exit 0
#Tot aan deze regel kopiëren#####
```

en wegschrijven met <esc : wq!>

24: Zet de rechten van het script goed

chmod 755 /etc/init.d/oscam

25: Voeg het script toe aan de opstartfiles:

update-rc.d oscam defaults

26:

reboot

28 Het starten/stoppen gaat nu geheel automatisch met een reboot. Mocht je willen (her)starten dan doe je dit door het commando

service oscam (re)start

of stoppen:

service oscam stop

27: Als alle goed is gegaan heb je nu een hele cleane installatie van de allerlaatste oscam versie.

test dit door met een webbrowser naar het adres 192.168.0.<x>:8888 te gaan. Het ipadres is natuurlijk hetzelfde als die van je server; het basispoortnr is 8888.

Dit kun je natuurlijk aanpassen binnen oscam.

Je kunt nu de config handmatig vullen via de webif/of bestaande oscam-configfiles naar /usr/local/etc kopiëren.

Er is nogal wat gewijzigd de laatste tijd dus het kan zijn dat bestaande configfiles hier en daar wat problemen geven.

De oscamconfiguratie zelf valt buiten de context van dit verhaal!

Succes!!!!!! Polarbear.