Debian Server Installation - Hardware

Evil
m.stroh@softhome.net

11. Mai 2014

Inhaltsverzeichnis

1 Vorwort

Diese Dokumentation ist eine Erweiterung der Dokumentationen 'Debian Server Installation auf einem Thin Client'. Es wird deshalb von einem bestehenden Debian System ausgegangen. Das Dokument beschreibt die Installation von modifizierter oder gebastelter Hardware die mit einem Server verbunden werden kann.

1.1 Warnhinweis

Das modifizieren von Hardware druch Änderung der Firmware kann zum Garantieverlust oder zur Unbrauchbarkeit des Geräts (brick) führen. Ich kann nicht haftbar gemacht werden für Fehler die durch diese Anleitung am Gerät oder Server entstehen können!

2 Temperatursensoren (digitemp)

Anleitung: *http://lena.franken.de/hardware/temperaturmessung.html*
Datenblatt Sensor: *http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf*

2.1 Schaltplan

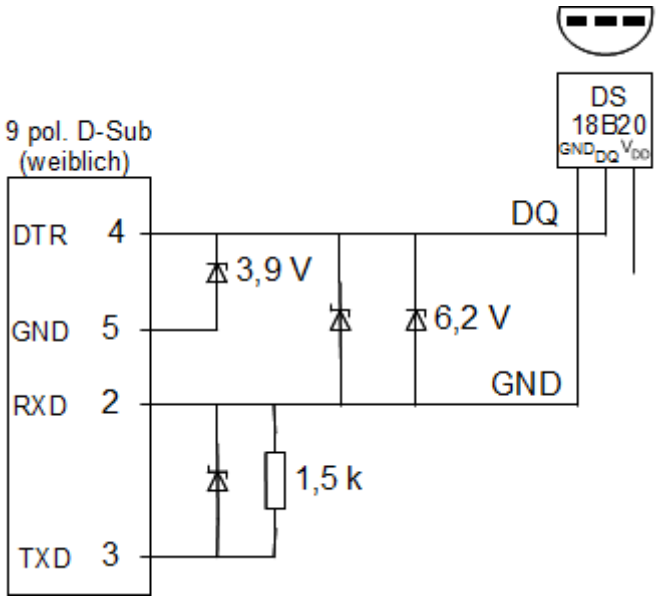

Abbildung 1: Schaltplan

- DS18B20 1-Wire Digital Thermometer IC
- Zenerdiaode 0,5 W, 3,9 V (BZX 79/BZX 55)

- Zenerdiaode 0,5 W, 6,2 V (BZX 79/BZX 55)
- Schottky-Diode 1N5819
- Widerstand 1,5 kOhm, 0.25 W

2.2 Einrichtung

```
apt-get install digitemp
digitemp_DS9097 -i -s /dev/ttyS0
digitemp_DS9097 -a
cp .digitemprc /etc/digitemp
```

/etc/digitemp [-rw-r--r-- root root]

```
TTY /dev/ttyS0
READ_TIME 850
LOG_TYPE 1
LOG_FORMAT "%.1C"
CNT_FORMAT "%b %d %H:%M:%S Sensor %s #%n %C"
HUM_FORMAT "%b %d %H:%M:%S Sensor %s C: %.2C H: %h%%"
SENSORS 1
ROM 0 0x10 0x69 0xC8 0x23 0xA1 0x08 0xD0 0xA0
```

Temperatur auslesen:

```
digitemp_DS9097 -a -q -c /etc/digitemp
```

Script für collectd:
/usr/local/bin/roomtemp [-rwxr-xr-x root staff]

```
#!/bin/bash
HOST=‘uname -n‘
INTERVAL=60
while true
do
        digitemp_DS9097 -a -q -c /etc/digitemp -o "PUTVAL $HOST/temp/temperature-%s interval=$INTERVAL N:%.1C"
       sleep $INTERVAL
done
```

3 Mini TFT-LCD-Display (LCD4Linux)

Anleitung:
http://geekparadise.de/2011/04/digitaler-bilderrahmen-von-pearl-als-statusdisplay-fur-dockstar/
Bezugsquellen:
Zur Zeit gibt leider keine mir bekannten Bezugsquellen, da es bei Pearl nicht mehr lieferbar ist.
http://www.pearl.de/a-PX1184-5618.shtml
http://www.pearl.de/a-HPM1184-5618.shtml

3.1 Hardware

Typ Digitaler Bilderrahmen 6,1 cm / 2,4”

Auflösung 320 x 240 Pixel

Anschluss USB

Das Display kann direkt an eine Raspberry Pi angeschlossen werden.

3.2 Display anschließen

Display per USB verbinden.
Durch 3 Sekunden langes drücken der Menü-Taste schaltet sich das Display ein.
Nun nochmals die Menü-Taste kurz drücken.
Der Curser steht nun auf „Mit PC verbinden“ den Menü-Punkt mit der Menü-Taste auswählen.

```
dmesg
```

```
Bus 001 Device 004: ID 1908:0102 GEMBIRD
```

3.3 lcd4linux patchen und installieren

```
apt-get install lcd4linux
/etc/init.d/lcd4linux stop
```

```
apt-get install libtool automake autoconf zlib1g-dev libssl-dev python-dev \
 libc6 libusb-dev subversion
```

Debian Squeeze:

```
apt-get install libibus-dev
```

Debian Wheezy:

```
apt-get install libibus-1.0-dev
```

Es muss entweder die GD Library mit xpm support oder ohne installiert werden.
Bei noxpm werden weniger abhängige Librarys benötigt als bei xpm.
Wenn allerdings bereits ein Programm installiert ist das die xpm Library benötig so muss diese verwendet werden (wird beim Installieren ausgewiesen). Darum habe ich mich für libgd2-xpm entschieden.
Bei Debian Wheezy wurde lcd4linux von libgd2-xpm abhängig gemacht und so entfällt die Installation hier.

```
apt-get install libgd2-noxpm-dev libgd2-noxpm
```

oder

```
apt-get install libgd2-xpm-dev libgd2-xpm
```

Debian Squeeze:

```
wget http://evil.hn.vc/linux/lcd4linux/sdcc-libraries_2.9.0-5_all.deb
wget http://evil.hn.vc/linux/lcd4linux/sdcc_2.9.0-5_i386.deb
wget http://evil.hn.vc/linux/lcd4linux/sdcc_2.9.0-5_armel.deb
```

Für ARM System:

```
dpkg -i sdcc-libraries_2.9.0-5_all.deb sdcc_2.9.0-5_armel.deb
```

Für x86 System:

```
dpkg -i sdcc-libraries_2.9.0-5_all.deb sdcc_2.9.0-5_i386.deb
```

Debian Wheezy:

```
apt-get install sdcc sdcc-libraries
```

Debian Wheezy - Raspberry Pi:

```
apt-get install cc1111
```

```
cd /usr/src
mkdir lcd4linux
cd lcd4linux
wget http://tech.section5.ch/files/dpfhack-0.1alpha.tgz
wget http://tech.section5.ch/files/dpf-lcd4linux.tgz
tar xzvf dpfhack-0.1alpha.tgz
tar xzvf dpf-lcd4linux.tgz
cd dpf/src
make
cd ..
```

```
./build-dpf-lcd4linux.sh ../src/dpflib/
```

```
Error validating server certificate for 'https://ssl.bulix.org:443':
(R)eject, accept (t)emporarily or accept (p)ermanently?
```

t

```
mv /usr/sbin/lcd4linux /usr/sbin/lcd4linux.old
cp lcd4linux/lcd4linux /usr/sbin/lcd4linux
```

3.4 Firmware ersetzen

Dieser Schritt muss nur gemacht werden wenn noch die originale Firmware aktiv ist!

dmesg

```
[ 4351.928590] usb 1-1.3: new full speed USB device number 4 using orion-ehci
[ 4352.093204] usb 1-1.3: New USB device found, idVendor=1908, idProduct=0102
[ 4352.100132] usb 1-1.3: New USB device strings: Mfr=2, Product=3, SerialNumber=0
[ 4352.107480] usb 1-1.3: Product: Digital Photo Frame
[ 4352.112408] usb 1-1.3: Manufacturer: BUILDWIN
[ 4352.128096] scsi1 : usb-storage 1-1.3:1.0
[ 4353.133543] scsi 1:0:0:0: CD-ROM            buildwin  Photo Frame     1.01 PQ: 0 ANSI: 2
[ 4353.239105] sr0: scsi3-mmc drive: 40x/40x writer cd/rw xa/form2 cdda tray
[ 4353.245930] cdrom: Uniform CD-ROM driver Revision: 3.20
[ 4353.253251] sr 1:0:0:0: Attached scsi CD-ROM sr0
[ 4353.307152] sd 0:0:0:0: Attached scsi generic sg0 type 0
[ 4353.315679] sr 1:0:0:0: Attached scsi generic sg1 type 5
[ 4384.098609] usb 1-1.3: reset full speed USB device number 4 using orion-ehci
```

python hackit.py /dev/sg1

```
Reading flash...
done
Found matching version info
Identifier: pearl
Now patching. There is no 100% guarantee that your device will
        work after doing this. You must never unplug the device from USB while
        it is being updated.
        Are you sure you take all risks and that you want to continue?
Type 'yes' to continue >
```

yes

```
Now disconnect the DPF from USB.
To activate the 'developer mode':

Press and hold MENU while USB is plugged in.
If successful, you will get the 'USB connect' message and the device
will appear as non-USB storage device

To put the device back into (almost) original state working
as USB storage, press the RESET button.
```

Nun den USB Anschluss des Displays ausstecken und wieder anstecken.
Um den Bilderrahmen in den „Display"-Modus zu versetzen muss man die die Menü-Taste für 3 Sekunden halten, dann wird der Text „Mit PC verbunden" angezeigt. Wenn nach dem Anstecken keine Taste gedrückt wird, so wird nach ca. 1-2 Minuten das Display automatisch in den „Display"-Modus („Mit PC verbunden") versetzt.

```
[ 6967.478606] usb 1-1.3: new full speed USB device number 5 using orion-ehci
[ 6967.643085] usb 1-1.3: New USB device found, idVendor=1908, idProduct=0102
[ 6967.650019] usb 1-1.3: New USB device strings: Mfr=2, Product=3, SerialNumb
[ 6967.657362] usb 1-1.3: Product: USB-Display
[ 6967.661593] usb 1-1.3: Manufacturer: hackfin
[ 6967.665881] usb 1-1.3: SerialNumber: 0000
```

3.5 Konfiguration erstellen

mkdir /var/lib/lcd4linux

/etc/lcd4linux.conf [-rw------- root root]

```
Display dpf {
    Driver     'DPF'
    Port       'usb0'
    Font       '6x8'
    Foreground 'ffffff'
    Background '000000'
    Basecolor  '000066'
}

Widget CPUBar1 {
        class 'Bar'
        expression  proc_stat::cpu('busy', 500)
#        expression2  proc_stat::cpu('system', 500)
        min 0
        max 100
        length 19
        direction 'N'
        update 500
    Foreground  '006C34'
}

Widget CPUTEXT1 {
    class 'Text'
    expression 'Load'
    align 'L'
    update 5000
    Foreground  'ffffff'
}

Widget CPUBar2 {
    class 'Bar'
    expression  proc_stat::cpu('iowait', 500)
    min 0
    max 100
    length 19
    direction 'N'
    update 500
}

Widget CPUTEXT2 {
    class 'Text'
    expression 'Io'
    width 5
    align 'L'
    update 5000
    Foreground  'ffffff'
}

Widget ETH0Bar1 {
    class 'Bar'
# Bar 0-100%, 0 - 8 MBit/s
    expression netdev::fast('eth0', 'Rx_bytes', 500)/800000*100
    min 0
    max 100
    length 19
```

```
    direction 'N'
    update 500
    Foreground  '006C34'
}

Widget ETH0Text1 {
    class 'Text'
    expression 'Recv'
    direction 'L'
    update 5000
}

Widget ETH0Bar2 {
    class 'Bar'
# Bar 0-100%, 0 - 768 kBit/s
    expression netdev::fast('eth0', 'Tx_bytes', 500)/76800*100
    min 0
    max 100
    length 19
    direction 'N'
    update 500
}

Widget ETH0Text2 {
    class 'Text'
    expression 'Send'
    direction 'L'
    update 5000
}

Widget IMAGE {
    class    'Image'
    file     '/var/lib/lcd4linux/LCD.png'
    update   10000
    reload   1
    visible  1
    inverted 0
}

Display 'DPF'

Layout Display320x240{
    Row8.Col4 'CPUBar1'
    Row8.Col5 'CPUBar1'
    Row29.Col03 'CPUTEXT1'
    Row8.Col09 'CPUBar2'
    Row8.Col10 'CPUBar2'
    Row29.Col09 'CPUTEXT2'
    Row8.Col44 'ETH0Bar1'
    Row8.Col45 'ETH0Bar1'
    Row29.Col43 'ETH0Text1'
    Row8.Col49 'ETH0Bar2'
    Row8.Col50 'ETH0Bar2'
    Row29.Col48 'ETH0Text2'

    Layer 2 {
        X1.Y1 'Image'
    }
}

Layout 'Display320x240'
```

```
chmod 600 /etc/lcd4linux.conf
```

Nun kann die Konfiguration und das Display getestet werden.

```
lcd4linux -vF
```

3.6 Automatische Erkennung

```
lsusb
```

```
Bus 002 Device 007: ID 1908:0102 GEMBIRD
```

```
lsusb -v -s 002:007 | egrep "idVendor|idProduct|iManufacturer|iProduct"
```

```
  idVendor            0x1908 GEMBIRD
  idProduct           0x0102
  iManufacturer            2 hackfin
  iProduct                 3 USB-Display
```

/lib/udev/rules.d/usb-display.rules [-rw------- root root]

```
SUBSYSTEM=="usb", ACTION=="add", ENV{ID_MODEL_ID}=="0102", ENV{ID_VENDOR_ID}=="1908", \
 RUN+="/etc/init.d/lcd4linux start"
SUBSYSTEM=="usb", ACTION=="remove", ENV{ID_MODEL_ID}=="0102", ENV{ID_VENDOR_ID}=="1908", \
 RUN+="/etc/init.d/lcd4linux stop"
```

```
/etc/init.d/udev reload
```

4 Webcam

Datenblatt
http://www.microsoft.com/hardware/de-at/p/lifecam-vx-800/JSD-00003

Anforderung
USB 1.1 oder 2.0

4.1 Hardware

Sensor CMOS VGA sensor

Auflösung 640 x 480 Pixel (0.31 Megapixel)

Blickwinkel 59°

Sonstiges Fokus fix, Automatische Helligkeitsanpassung

Audio Integriertes rundum Mikrofon

Anschluss USB

Stromaufnahme 150 mA

Da die Kamera eine Stromaufnahme von über 100 mA hat muss sie bei der Raspberry Pi über einen aktiven USB-Hub betrieben werden. Nur die Audioaufnahme hat direkt am USB-Port funktioniert.

4.2 Installation

```
lsusb

Bus 001 Device 013: ID 045e:0766 Microsoft Corp.

dmesg | grep "Microsoft LifeCam VX-800"

usb 1-2.2: Product: Microsoft LifeCam VX-800
uvcvideo: Found UVC 1.00 device Microsoft LifeCam VX-800 (045e:0766)
input: Microsoft LifeCam VX-800 as /devices/pci0000:00/0000:00:0f.5/usb1/1-2/1-2.2/1-2.2:1.0/input/input7
```

Da im System bereits eine DVB-Empfänger vorhanden ist wurde die Kamera als zweites Video Device (/dev/video1) erkannt. Es ist aber möglich der Web-Cam einen Namen zu geben um so später ungeachtet der anderen Geräte im System darauf zugreifen zu können.

```
udevadm info -a -p  $(udevadm info -q path -n /dev/video0) | grep name

  ATTR{name}=="em28xx #0 video"

udevadm info -a -p  $(udevadm info -q path -n /dev/video1) | grep name

  ATTR{name}=="Microsoft LifeCam VX-800"
```

/lib/udev/rules.d/videodevices.rules [-rw-r--r-- root root]

```
SUBSYSTEM=="video4linux", \
  ATTR{name}=="Microsoft LifeCam VX-800", \
  SYMLINK+="webcam"
```

```
/etc/init.d/udev reload
udevadm --debug test /sys/class/video4linux/video1

v4l-info /dev/webcam | grep card

  card  : "Microsoft LifeCam VX-800"

arecord -l

 **** Liste der Hardware-Geräte (CAPTURE) ****
Karte 0: Audio [CS5535 Audio], Gerät 0: CS5535 Audio [CS5535 Audio]
  Sub-Geräte: 1/1
  Sub-Gerät #0: subdevice #0
Karte 1: VX800 [Microsoft LifeCam VX-800], Gerät 0: USB Audio [USB Audio]
  Sub-Geräte: 1/1
  Sub-Gerät #0: subdevice #0
```

Die Web-Cam wird als Karte 1 ausgegeben darum muss card1 für den Zugriff benutzt werden.

```
cat /proc/asound/card1/stream0
```

```
Microsoft Microsoft LifeCam VX-800 at usb-0000:00:0f.5-2, high speed : USB Audio

Capture:
  Status: Stop
  Interface 3
    Altset 1
    Format: S16_LE
    Channels: 1
    Endpoint: 3 IN (NONE)
    Rates: 8000, 11025, 16000, 22050, 32000, 44100, 48000
    Data packet interval: 1000 us
```

4.3 Aufzeichnung Audio/Video

Audio Aufzeichung:

```
arecord -c 1 -d 10 -t wav -f S16_LE -r 44100 -D hw:1,0 audio.wav
arecord -c 1 -d 30 -t wav -f S16_LE -r 44100 -D hw:1,0 | lame -V2 - audio.mp3
```

-c . . . Anzahl der Kanäle
-d . . . Dauer die aufgezeichnet werden soll
-t . . . Datei Typ
-f . . . Format
-r . . . Aufzeichungsrate in Hz
-V . . . MP3 Qualität (0=high quality,bigger files. 9=smaller files)

```
alsamixer -c 1 -V capture
```

/usr/local/bin/webcam_micro.sh [-rwxr-xr-x root root]

```
#!/bin/bash
sleep 1
/usr/bin/amixer -c 1 set Mic 100%
```

/lib/udev/rules.d/videodevices.rules [-rw-r--r-- root root]

```
 SUBSYSTEM=="video4linux", \
   ATTR{name}=="Microsoft LifeCam VX-800", \
   SYMLINK+="webcam", \
   RUN+="/usr/bin/logger Set up webcam micro", \
   RUN+="/usr/local/bin/webcam_micro.sh"
```

Bild Aufzeichung:

```
apt-get install ffmpeg mencoder
```

```
ffmpeg -f video4linux2 -ss 10 -r 1 -i /dev/webcam -vframes 1 -f image2 image2.jpg
```

Audio Aufzeichung:

```
ffmpeg -f alsa -ac 1 -ar 44100 -i hw:1,0 -acodec mp2 -ab 128k -t 20 audio.mp3
```

Video Aufzeichung:

```
ffmpeg -f video4linux2 -i /dev/webcam -r 4 -t 10 -f avi -vcodec mpeg4 video.avi
```

Audio/Video Aufzeichung:

```
ffmpeg -f alsa -ac 1 -i hw:1,0 -f alsa -f video4linux2 -i /dev/webcam \
 -r 4 -t 10 -f avi -vcodec mpeg4 movie.avi
```

-vframes . . . Anzahl der Frames die Aufgezeichnet werden soll
-t . . . Dauer die Aufgezeichnet werden soll
-ss . . . Dauer um die die Aufzeichung verzöger ist
-r . . . Frames pro Sekunde
-i . . . Web-Cam Video Device

```
mencoder tv:// -tv driver=v4l2:width=640:height=480:device=/dev/webcam:forceaudio:alsa:adevice=hw.1,0 \
-ovc lavc -oac mp3lame -lameopts cbr:br=128:mode=3 -o mencoder.avi
```

LifeCam VX-800:
Da die Helligkeitsanpassung mehrere Sekunden benötigt muss die Bild-Aufzeichung verzögert werden (Parameter -ss). Die automatische Helligkeitsanpassung funktioniert bei einer Video Aufzeichung unter 4 fps nicht.
Da /dev/video0 von einer DVB-Empfänger belegt ist verwendet die Web-Cam /dev/video1 und als Alias /dev/webcam.

4.4 Klassische Webcam Funktion

```
apt-get install webcam v4l-conf
```

```
v4l-info /dev/webcam | grep name | head -n 1
```

```
        name                         : "Camera 1"
```

/etc/webcam.conf [-rw-r--r-- root root]

```
[ftp]
host = localhost
user = webcam
pass = xxxxxx
dir  = /var/www
file = webcam.jpeg
tmp  = uploading.jpeg
passive = 1
debug = 0
auto = 0
local = 1
ssh = 0

[grab]
```

```
device = /dev/webcam
text = "Webcam %Y-%m-%d %H:%M:%S"
#infofile = filename
fg_red = 255
fg_green = 255
fg_blue = 255
width = 640
height = 480
delay = 5
wait = 0
input = Camera 1
rotate = 0
top = 0
left = 0
bottom = -1
right = -1
quality = 75
trigger = 0
once = 0
```

webcam /etc/webcam.conf &

Leider funktioniert das Programm nicht korrekt weil die automatische Helligkeitsanpassung nicht schnell genug reagieren kann. Die dafür vorgesehene Verzögerung (wait = 3) funktioniert nicht.
Es wird zwar alle 5 Sekunden eine Bild erzeugt, allerdings nicht mit der optimalen Helligkeit.