WS2014

**Projekt Papaschlump**

Anleitung erstellt von Raphael Kawasch

Rev. 0 vom 17.11.2014

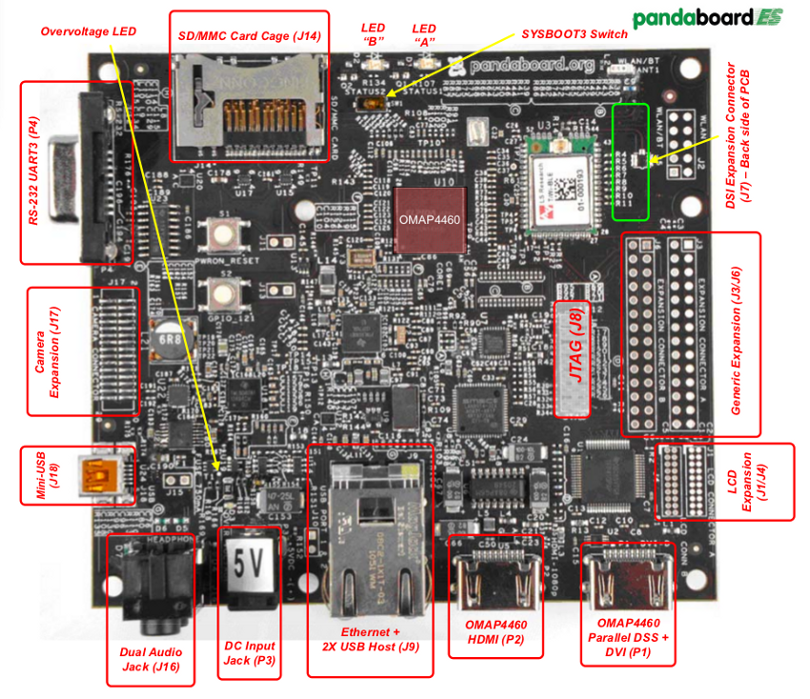


Abbildung : PandaBoard Anschlüsse http://pandaboard.org/node/223/#PBSetUp

Setup Pandaboard ES Rev. B3

Inhalt

[Abbildungsverzeichnis 1](#_Toc404100064)

[Ursprung und Absicht 2](#_Toc404100065)

[Ubuntu 12.04 bootfähig auf SD Karte installieren 3](#_Toc404100066)

[1. Download der nötigen Komponenten. 3](#_Toc404100067)

[2. Bootloader für PandaBoard Rev. B3 konfigurieren 4](#_Toc404100068)

[3. Cross Compiler installieren 4](#_Toc404100069)

[4. SD Karte vorbereiten 5](#_Toc404100070)

[5. Daten auf SD schreiben 5](#_Toc404100071)

[6. PandaBoard erstmalig booten 7](#_Toc404100072)

[7. WLAN einrichten 7](#_Toc404100073)

[8. SSH einrichten 8](#_Toc404100074)

[9. VNC Server einrichten 8](#_Toc404100075)

[Papaschlumpf Konfiguration 9](#_Toc404100076)

# Abbildungsverzeichnis

[Abbildung 1: PandaBoard Anschlüsse http://pandaboard.org/node/223/#PBSetUp 0](file:///E:\Dokumente\Unterlagen\Hochschule\DT\SetupPB_ESRev3B.docx#_Toc404100077)

[Abbildung 2: Erster Testaufbau 3](#_Toc404100078)

[Abbildung 3: Zweiter Testaufbau – Erster Boot Abbildung 4: PandaBoard im Testaufbau 3](#_Toc404100079)

[Abbildung 5: lsblk - Anzeigen der Speichermedian 5](#_Toc404100080)

[Abbildung 6: Kontrolle des Einhängepunktes 6](#_Toc404100081)

[Abbildung 7: WLAN Konfiguration für das eduroam Netz der OTH 8](#_Toc404100082)

# Ursprung und Absicht

 **Anleitung auch auf Pandaborad ES Rev. A3 4430 anwendbar.**

Im Rahmen unseres Projektes sollte ein PandaBoard eingesetzt werden. Beim Installieren des Betriebssystems (Ubuntu 12.04) wie unter <https://wiki.ubuntu.com/ARM/OmapDesktopInstall> beschrieben, haben wir folgende Fehlermeldung erhalten.

U-Boot SPL 2011.12 (Apr 02 2012 – 18:13:04)  
Texas Instruments OMAP4460 ES1.1  
Bypassing DPLL failed 4a008180  
SDRAM: identified size not same as expected size identified: 80 expected: 400000  
OMAP SD/MMC: 0

Eine online Recherche hatte unter anderem ergeben, dass die Boards mit Elpida Chip andre Memory- Settings benötigen, auf was diese Fehlermeldung hindeutet. Es war uns daher nicht möglich ohne weiteres die „Prebuilt Binaries“ zu verwenden. Daher versuchten wir es mit andren Kernel und Bootloader Paketen von diversen Anbietern. Auch mit einer selbst Cross- Compiliertem Version war der Bootvorgang nicht möglich. Letztendlich konnten wir dann doch zu einer, rückblickend recht simplen Lösung kommen.

Nähere Informationen zur Fehlermeldung wurden folgenden Quellen entnommen.

<https://groups.google.com/forum/#!msg/pandaboard/OWvyla72e00/KdhQNi-pagsJ>

<http://lists.denx.de/pipermail/u-boot/2013-November/167199.html>

<http://eewiki.net/display/linuxonarm/PandaBoard#PandaBoard-BasicRequirements> (FAQ)

<https://www.svtronics.com/index.php?route=product/product&product_id=22>

**Error triangle Nachfolgend wird das von uns verwendete Installationsprozedere beschrieben.**

**Im Rahmen der Recherche wurden diverse Anleitungen befolgt, wobei der Autor immer von gewissen Gegebenheiten ausging welche nicht ersichtlich waren. Aus diesem Grund seien nun hier unsere Annahmen gelistet.**

* Es wird von einem Ubuntu 14.04 Host ausgegangen (Neuinstallation).
* Serial auf USB Kabel / bzw. Adapter vorhanden
* 5V Netzteil bzw. Mini USB auf USB A Kabel zur Stromversorgung
* Optional HDMI Kabel und Monitor evtl. Adapter HDMI auf DVI
* Optional Tastatur und Maus
* Optional Netzwerkkabel



Abbildung 2: Erster Testaufbau

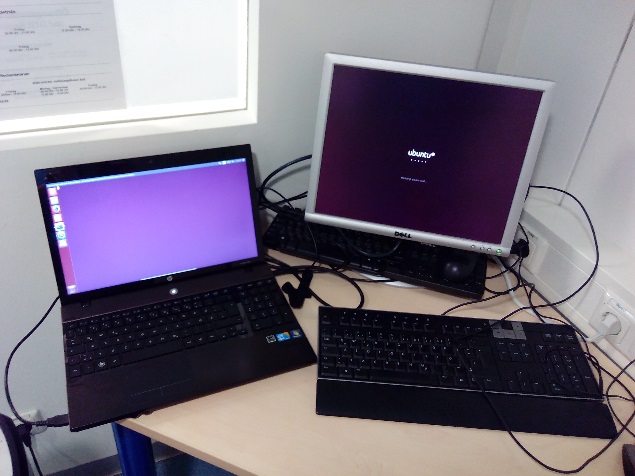


Abbildung 3: Zweiter Testaufbau – Erster Boot Abbildung 4: PandaBoard im Testaufbau

# Ubuntu 12.04 bootfähig auf SD Karte installieren

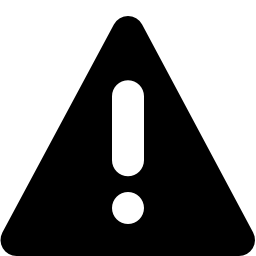
### Download der nötigen Komponenten.

* Aktuelle Prebuilt Binaries (ubuntu-12.04-preinstalled-desktop-armhf+omap4.img)
* Alternativer Bootloader
* Evtl. fehlende Programme, Compiler, Pakete (wird nicht umfänglich beschrieben)

Dazu öffnen wir einen Terminal (Strg+Alt+T) und wechseln in das Verzeichnis indem die Daten abgelegt werden sollen z. B.

$ cd Downloads

**An dieser Stelle sei erwähnt, dass alle folgenden Eingaben in der Konsole mit einem Rahmen versehen sind. In die Konsole ist alles nach dem „$“ Zeichen einzugeben. Zur automatischen Vervollständigung der Eingabe kann die Tabulator Taste verwendet werden.**



Nun laden wir die „Prebuilt Binaries„ herunter.

$ wget http://cdimage.ubuntu.com/releases/12.04/release/ubuntu-12.04-preinstalled-desktop-armhf+omap4.img.gz

Alternativ kann dies auch direkt auf <http://omappedia.org/> herunterladen.

Jetzt benötigen wir noch einen alternativen Bootloader. Diesen erhalten wir z.B. über

$ git clone git://git.linaro.org/boot/u-boot-linaro-stable.git

Es werden nun ca. 53 MB in das Verzeichnis. ./u-boot-linaro-stable heruntergeladen.

Sollte git nicht verfügbar sein muss es nachinstalliert werden

$ sudo apt-get update

$ sudo apt-get install git (kann im OTH Netzf Probleme fehlschlagen)

Wenn das Paket nicht gefunden wird muss noch das Repository eingebunden werden

$ sudo add-apt-repository ppa:git-core/ppa

$ sudo apt-get update

### Bootloader für PandaBoard Rev. B3 konfigurieren

Nun müssen Änderungen in drei Dateien vorgenommen werden.

$sudo gedit ./u-boot-linaro-stable/arch/arm/cpu/armv7/omap4/sdram\_elpida.c

Nun die Zeile const u32 ddr3\_ext\_phy-ctrl\_const\_base[…]; suchen und vor das #ifdef CONFIG\_SYS\_EMIF-PRECALCULATED\_TIMING\_REGS ziehen. Speichern und schließen.

sudo nano ./u-boot-linaro-stable/arch/arm/cpu/arm7/omap5/sdram.c

Hier gehen wir analog vor, allerdings ziehen wir die komplette Arraydefinition nach vorne. D.h. const u32 ddr3\_ext\_phy-ctrl\_const\_base[…]= {……..}; vor das #ifdef …

sudo gedit ./u-boot-linaro-stable/include/configs/omap4\_common.h

Hier die Zeile #def CONFIG\_SYS\_EMIF\_PRECALCULATED\_TIMING\_REGS auskommentieren. Speichern und schließen.

Im Anschluss kann der Bootloader kompiliert werden.

### Cross Compiler installieren

$ sudo apt-get install gcc-arm-linux-gnueabihf cpp-arm-linux-gnueabihf

installiert den Compiler

$ export CROSS\_COMPILE=arm-linux-gnueabihf-

legt eine Variable für den Compilernamen an

$ make CROSS\_COMPILE=arm-linux-gnueabi- omap4\_panda

kompiliert den Bootloader (Leerzeichen vor omap4\_panda beachten)

Wobei CROSS\_COMPILE für die Bezeichnung des Compilers steht und durch den Export automatisch durch arm-linux-gnueabihf- ersetzt wird.

### SD Karte vorbereiten

Vorab müssen wir lokalisieren wo die SD-Karte eingehängt wurde.

Der Befehl lsblk ist dazu sehr nützlich.

$ lsblk

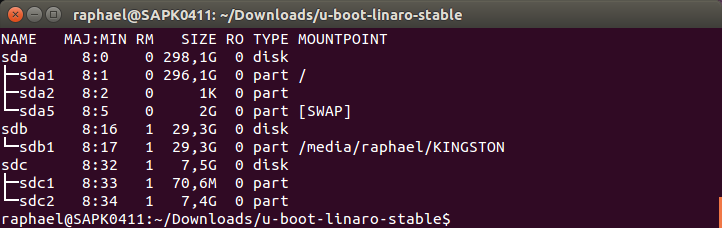


Abbildung 5: lsblk - Anzeigen der Speichermedian

In der nun angezeigten Auflistung suchen wir das Ziellaufwerk. Hier /dev/sdc und merken uns diesen Pfad.

Alternativ kann man mittels export SDCARD=/dev/sdc eine Variable für den Pfad erstellen.

$ export SDCARD=/dev/sdc

Und nun anstelle von /dev/sdc mit ${SDCARD} arbeiten. Wichtig ist jetzt, dass in obiger Auflistung noch kein Einhängepunkt angegeben ist.

Wenn doch kann das Medium über $ sudo umount ${SDCARD} ausgehängt werden.

Wenn nötig:

$ sudo umount ${SDCARD}

Nun formatieren wir die SD Karte im FAT32 Format.

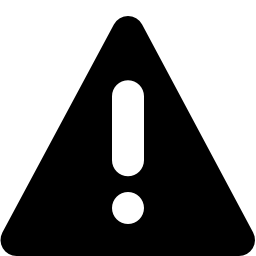
$ sudo mkdosfs -I -F 32 ${SDCARD}

Näheres unter <http://www.ehow.com/how_4810735_format-sd-card-linux.html> bzw. <http://wiki.ubuntuusers.de/Formatieren>

### Daten auf SD schreiben

Als erstes schreiben wir die „Prebuilt Binaries“ auf die SD-Karte.

**An dieser Stelle befinden wir uns direkt im Verzeichnis Downloads**



$ sudo gunzip -c ./ubuntu-12.04-preinstalled-desktop-armhf+omap4.img.gz | sudo dd bs=4M of=${SDCARD}

$ sudo sync

Nun ersetzen wir **u-boot.img**, **u-boot.bin** und **MLO** im boot – Verzeichnis der SD Karte durch die aus dem selbst erstellten Bootloader. Diese finden wir unter ./Downloads/u-boot-linaro-stable/

Wer das ganze gerne per Konsole macht, muss nun folgende Befehle absetzen.

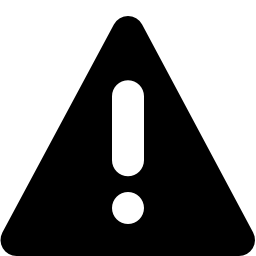
$ lsblk

Nun den mountpoint der Boot Partition suchen.

Hier /media/raphael/165A-3C60

Die Boot Partition ist ca. 72 MB groß!

**Wurde die SD Karte nicht automatisch eingehängt muss sie an dieser Stelle manuell eingebunden werden.**



Siehe dazu: <http://askubuntu.com/questions/95391/how-do-i-mount-an-sd-card>

Wenn man die Karte abzieht und wieder einsteckt sollte diese automatisch eingehängt werden!

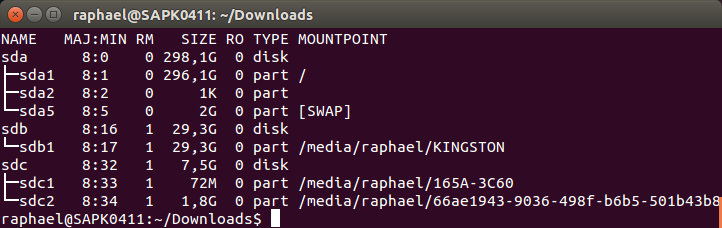


Abbildung 6: Kontrolle des Einhängepunktes

$ sudo cp ./u-boot-linaro-stable/u-boot.bin /media/raphael/165A-3C60/

$ sudo cp ./u-boot-linaro-stable/u-boot.img /media/raphael/165A-3C60/

$ sudo cp ./u-boot-linaro-stable/MLO /media/raphael/165A-3C60/

Weitere Infos zum Kopieren unter <http://wiki.ubuntuusers.de/cp>

Nun die SD – Karte aushängen.

$ sudo umount ${SDCARD}1

$ sudo umount ${SDCARD}2

Und per lsblk nochmals kontrollieren ob die Karte ausgehängt ist.

$ lsblk

Nun ist die Karte bereit zum ersten booten.

### PandaBoard erstmalig booten

Nun verbinden wir das PandaBoard mit Monitor (HDMI 1080p), Tastatur, Maus und Netzwerkkabel. Optional kann das Board auch nur mit einem Seriell zu USB Adapter am eigenen Host angeschlossen werden. (Nur sinnvoll wenn die Serverversion von Ubuntut verwendet wird, da sonst keine Interaktion mit dem Betriebssystem möglich ist. – Es wird nur der Bootvorgang ausgegeben!)

Wer das Board per Seriellen Adapter angeschlossen hat, wechselt nun in den Terminal und öffnet die Serielle Konsole.

Minicom

Zuerst öffnen wir den Terminal und suchen uns den Anschluss des Seriellen Adapters.

$ dmesg | grep -e tty

Hier müssen wir nun das richtige Gerät wählen.

...USB Serial Device converter now attached to ttyXXX < --- select the appropriate tty connection

Jetzt folgt das Einrichten von minicom.

$sudo minicom

Nun per Strg+A => Z => O in die Konfiguration wechseln.

Dort auf Einstellungen zum Seriellen Anschluss wechseln und den Port anpassen.

Zudem mit F die Hardware Flow Control auf nein setzen.

Einstellungen als dfl Speichern und minicom neu starten.

Eine detaillierte Anleitung wird unter <http://omappedia.org/wiki/Ubuntu_Minicom> bereitgestellt.

Screen

Screen kann ganz einfach wie folgt aufgerufen warden.

$sudo screen /dev/ttyUSB0 115200

Bei fehlendem Paket kann dies über

$ sudo apt-get install screen

nachgeladen werden.

Nun installieren wird die SD Karte im PandaBoard und schließen das Netzteil an.

### WLAN einrichten

An dieser Stelle sollte nun für die weitere Verwendung eine WLAN Verbindung hergestellt werden.

Dazu verwenden wir den unter Ubuntu mitgelieferten Netzwerkmanager.

Die Konfiguration für das WLAN der OTH wird nachfolgend durch einige Screenshots verdeutlicht, auf eine nähere Beschreibung wird verzichtet.

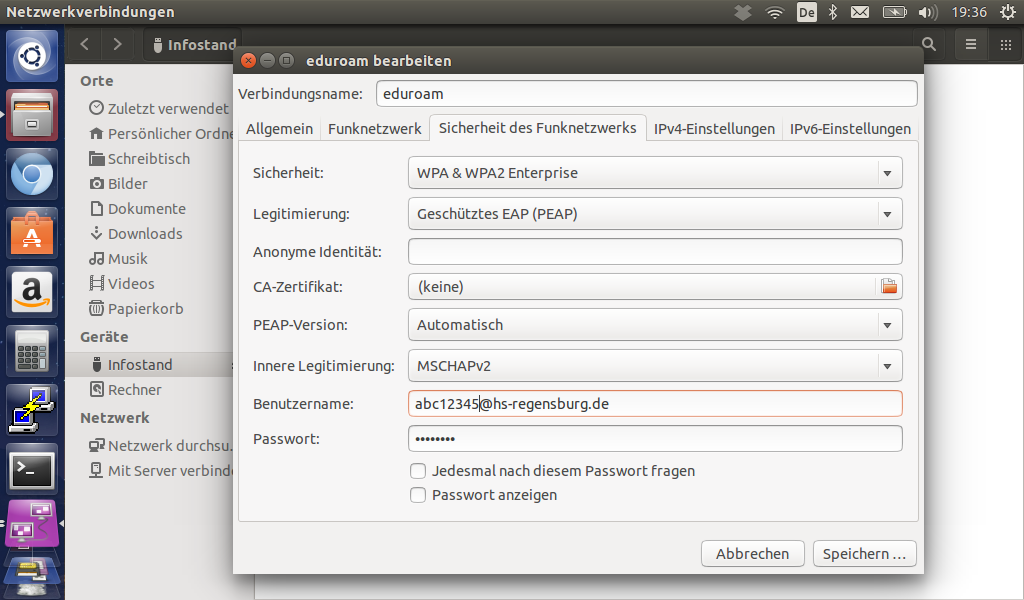


Abbildung 7: WLAN Konfiguration für das eduroam Netz der OTH

### SSH einrichten

Um das PandaBoard per SSH erreichen zu können, muss zuerst ein SSH Server installiert werden.

Dies erfolgt mittels

$ sudo apt-get install openssh-server

Nun kann man sich per

$ iwconfig

die IP-Adresse des PandaBoards ausgeben lassen.

 **Eine feste IP Adresse kann zum Projektzwecke vom Laborleiter vergeben werden!**

Über diese IP Adresse und den Port 22 (je nach Konfiguraiton) kann nun das Board erreicht werden.

Weitere Infos unter: https://help.ubuntu.com/10.04/serverguide/openssh-server.html

### VNC Server einrichten

Wenn nötig kann man nun noch einen VNC Server einrichten.

Bei unserer Konfiguration haben wir uns für TightVNC entschieden.

$ sudo apt-get install tightvncserver xtightvncviewer tightvnc-java

Nun kann der Server z.B. über SSH gestartet werden

$ tightvncserver

Weitere Informationen: <http://wiki.ubuntuusers.de/VNC>

### Schaltung der GPIO

Um die GPIO Ports ansteuern zu können müssen sie vorab angelegt werden. Danach kann der gewünschte Wert ( 1 bzw 0 ) gesetzt werden.

Im Manual des PandaBoards sucht man sich vorab die benötigten PINs aus.

Siehe:

Exemplarisch soll im Folgenden der Pin 18 (J6) angelegt werden.

Dazu öffnen wir den Terminal und legt den entsprechenden gpio32 für den PIN 18 an.

$ sudo su

$ cd /sys/class/gpio

$ echo 32 > export

$ cd gpio32

$ echo "low" > direction

Um nun den Port an bzw. ab zu stellen kann man die Werte entsprechend setzen.

$ echo "0" > value

$ echo "1" > value

Soll es möglich sein ohne sudo Rechte auf die GPIO zuzugreifen, kann dies mit untenstehenden Zeilen erfolgen.

sudo sh -c "echo 32 > /sys/class/gpio/export"

sudo sh -c "echo \"low\" > /sys/class/gpio/gpio32/direction"

sudo chgrp yourgroup /sys/class/gpio/gpio32/value

sudo chmod 664 /sys/class/gpio/gpio32/value

exit 0

Näheres dazu kann auf <http://www.nerdenmeister.org/2011/11/13/using-the-gpio-pins-on-a-pandaboard/> nachgelesen werden.

## Papaschlumpf Konfiguration

|  |  |
| --- | --- |
| PandaBoard | ES Rev. A6 – OMAP 4430 |
| OS | Ubunut 12.04 Trusty Tahr |
| Username | test |
| Hostname | test-desktop |
| Passwort | Test1234 |
| IP-Adresse WLAN | In der OTH variabel siehe /home/test/current\_ip |
| SSH Port | 22 |
| VNC Port | 5901 => (172.16.39.37:1) |