

# boneserver

## Installations- und Betriebsanleitung

Caspar Friedrich

7. Oktober 2014

### Inhaltsverzeichnis

<b>1</b>	<b>Hardware</b>	<b>2</b>
<b>2</b>	<b>Installation</b>	<b>2</b>
2.1	SD-Karte vorbereiten . . . . .	2
2.2	Installation im internen Speicher . . . . .	4
2.3	boneserver installieren . . . . .	4
<b>3</b>	<b>Betrieb</b>	<b>5</b>
3.1	Webinterface . . . . .	5
3.1.1	Netzwerkverbindung herstellen . . . . .	5
3.1.2	Webinterface aufrufen . . . . .	6
3.2	Bedienelemente . . . . .	6
3.2.1	GPIO . . . . .	6
3.2.2	PWM . . . . .	6
3.2.3	ADC . . . . .	6
<b>4</b>	<b>Wartung</b>	<b>6</b>
4.1	Backup . . . . .	7
4.2	System aktualisieren . . . . .	8
4.3	boneserver aktualisieren . . . . .	8
4.4	System bereinigen . . . . .	8

# 1 Hardware

Dieses Handbuch ist für den **BeagleBone Black Revision A5C** (im Folgenden kurz als BeagleBone bezeichnet) geschrieben und getestet. Sofern nachfolgende oder vorangegangene Revisionen zu dieser kompatibel ist, sollte die Installation aber dennoch problemlos möglich sein.

## 2 Installation

Als Betriebssystem wird [Arch Linux ARM](#) verwendet, eine Portierung von Arch Linux für ARM-Prozessoren. Arch Linux ARM stellt auch ein spezielles package repository zur Verfügung.

### 2.1 SD-Karte vorbereiten

Auf der Homepage von Arch Linux ARM gibt es eine Installationsanleitung, die laufend aktualisiert wird. Die Folgende Anleitung ist daher im wesentlichen eine Übersetzung. Ausgegangen von einem Linux als Host-System, dazu kann auch die mitgelieferte Ångström Linux auf den BeagleBone verwendet werden.

**Voraussetzungen** sind die Pakete *dosfstools* und *wget* sowie root-Rechte und eine Micro SD-Karte mit mindestens 2GB Speicherkapazität.

1. Finden sie zunächst heraus, welcher Laufwerkspfad der Vorgesehenen SD-Karte entspricht. Meist `/dev/sd[a, b, ...]` oder `/dev/mmcblk[0, 1, ...]`.

**Überprüfen Sie Laufwerkspfade genau bevor sie mit der Installation beginnen, da sonst irreparable Schäden am Host-System auftreten können!**

2. Starten sie *fdisk* um die SD-Karte zu formatieren:

```
fdisk /dev/sdX
```

3. Erstellen sie eine neue Partitionstabelle und die nötigen Partitionen  
Dazu geben sie nacheinander die folgenden Lommandos ein (jeweils mit *enter* bestätigen):

Kommando	Funktion
<code>o</code>	Erzeugt eine neue Partitionstabelle
<code>n, p, 1</code>	Erzeugt eine <i>neue, primäre, erste</i> Partition
<code>enter</code>	Bestätigt den Default-Wert für den ersten Sektor
<code>+64M</code>	+64M als letzten Sektor setzt die Partitionsgröße auf 64MByte
<code>t, e</code>	Ändert den Partitionstyp auf „W95 FAT16 (LBA)“
<code>a, 1</code>	Setzt das <i>boot flag</i> der ersten Partition (je nach <i>fdisk</i> -Version wird die erste Partition automatisch ausgewählt, da nur eine zur Verfügung steht)
<code>n, p, 2</code>	Erzeugt eine <i>neue, primäre, zweite</i> Partition
<code>2x enter</code>	Setzt die Default-Werte für den ersten und letzten Sektor der Partition
<code>w</code>	Schreibt Änderungen in die Partitonstabelle

4. Formatieren der ersten Partition:

```
mkfs.vfat -F 16 /dev/sdX1
```

5. Formatieren der zweiten Partition:

```
mkfs.ext4 /dev/sdX2
```

6. Laden sie den *bootloader tarball* herunter und entpacken sie ihn auf die erste Partition der SD-Karte:

```
wget http://archlinuxarm.org/os/omap/BeagleBone-bootloader.tar.gz
mkdir boot
mount /dev/sdX1 boot
tar -xvf BeagleBone-bootloader.tar.gz -C boot
sync && umount boot
```

7. Laden sie den *rootfs tarball* herunter und enpacken sie ihr auf die zweite Partition der SD-Karte (hierzu müssen sie als *root* eingeloggt sein, *sudo* reicht in diesem Fall nicht):

```
wget http://archlinuxarm.org/os/ArchLinuxARM-am33x-latest.tar.gz
mkdir root
mount /dev/sdX2 root
tar -xf ArchLinuxARM-am33x-latest.tar.gz -C root
sync && umount root
```

8. Stecken sie die SD-Karte in den BeagleBone und haleten sie die Taste um von der SD-Karte zu booten gedrückt, während sie die Power-Taste betätigen.  
Wenn das System gestartet ist, können sie sich auf der Kommandozeile oder via *ssh* einloggen.

Benutzernahme/Passwort lautet **root/root**.

Aus Sicherheitsgründen sollten sie nach dem Systemstart als erstes das root-Passwort ändern:

```
passwd root
```

Da sich, außer zu Wartungszwecken nicht am System angemeldet werden muss, kann auf die Erstellung eines regulären Benutzers verzichtet werden.

## 2.2 Installation im internen Speicher

**Hinweis:** Der BeagleBone hat zwar eine eingebaute Uhr allerdings keine Batterie. Nach einem Neustart kann es daher passieren, dass die interne Uhr auf den default-Wert zurück gesetzt wird. Überprüfen sie mittel `date` die aktuelle Systemzeit und aktualisieren diese gegebenenfalls via `ntpdate -u pool.ntp.org`

1. Um Arch Linux direkt auf der eMMC zu installieren, aktualisieren sie zunächst das eben gestarteten System und installieren die Pakete *wget*, *dosfstools* und *ntp*.

```
pacman -Syu wget dosfstools ntp
```

Das Paket *ntp* stellt hierbei vor allem das Programm *ntpdate* zur Verfügung (s. O.).

2. Der interne Speicher ist bereits korrekt partitioniert, folgen sie daher nur den Schritten 4 bis 7. Die Partitionen sind *mmcblk1p1* bzw. *mmcblk1p2* (s. O.).
3. Fahren sie das System herunter und warten sie bis alle LEDs erloschen sind.
4. Entfernen sie die SD-Karte und starten sie das System erneut.

## 2.3 boneserver installieren

**Repository klonen** *boneserver* ist via GitHub verfügbar. Führen sie dazu zunächst ein Systemupdate durch um alle Pakete auf den neusten Stand zu bringen und installieren sie das Paket *git*. Anschließend klonen sie das Repository nach */opt*.

```
pacman -Syu git
git -C /opt clone https://github.com/XMrVertigoX/boneserver.git
```

**Installationsskript ausführen** Im root-Verzeichnis des Repositories befindet sich ein Skript, welches die weitere Installation übernimmt. Wechseln sie dazu in das Verzeichnis und führen sie das Installationsskript aus.

```
cd /opt/boneserver
./install.sh
```

Dabei werden alle erforderlichen Pakete und Module installiert, die Konfigurationsdateien verlinkt sowie die Daemons installiert und gestartet.

Wenn das Skript fehlerfrei durchgelaufen ist, wird der BeagleBone automatisch neu gestartet und die Installation ist angeschlossen.

## 3 Betrieb

### 3.1 Webinterface

Für die Verwendung des webinterface wird eine Netzwerkverbindung zum BeagleBone und ein webbrowser<sup>1</sup> mit aktiviertem JavaScript vorausgesetzt.

#### 3.1.1 Netzwerkverbindung herstellen

**Hinweis:** Die default-Konfiguration des BeagleBone sieht den Betrieb mit einem DHCP-Server vor, sollte das nicht gewünscht oder möglich sein, kann über die üblichen Wege eine statische IP eingestellt werden. Anleitungen hierzu sind im Internet zur Genüge verfügbar.

Steht ein DNS-Server zur Verfügung, kann der BeagleBone über seinen hostname erreicht werden, standardmäßig *boneserver*. Ansonsten finden sie zunächst heraus, welche IP dem BeagleBone zugewiesen wurde. Hierfür kann entweder, falls verfügbar, die routing-Tabelle den DHCP-Servers konsultiert werden oder in der Kommandozeile via *ip* die aktuelle Adresse der einzelnen Netzwerkadapter abgerufen werden (Abbildung 1).

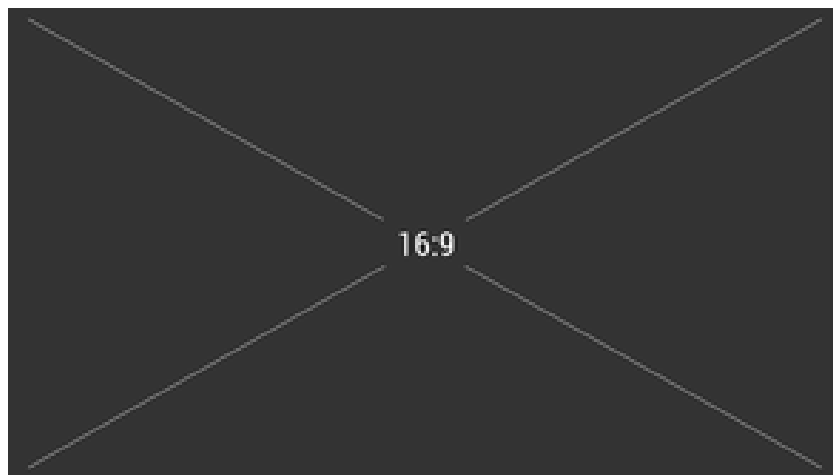


Abbildung 1: IP des BeagleBone abrufen

---

<sup>1</sup>Das webinterface verwendet *jQuery* in der Version 2.1.1, aktuelle webbrowser sollten hier keine Probleme bereiten. Ansonsten kann die Homepage von JQuery konsultiert werden: <http://jquery.com/browser-support/>

### 3.1.2 Webinterface aufrufen

Das webinterface kann einfach über DNS-Namen oder die IP dann im webbrowser aufgerufen werden. Ist die Verbindung hergestellt, wird dies durch einen grünen Haken rechts in der Titelleiste angezeigt (Abbildung 2) und die Steuerelemente werden generiert. Sollte die Verbindung einmal unterbrochen werden, wechselt dieser Haken in ein rotes Kreuz. In diesem Fall kann die Seite einfach neu geladen werden, eventuelle Konfigurationen bleiben dabei erhalten.



Abbildung 2: Webinterface des boneserver

**Hinweis:** Das webinterface kann immer nur von einem Fenster aus aufgerufen werden, es kann daher passieren, dass bei einem schnellen Fensterwechsel oder page reload die Verbindung nicht sofort hergestellt wird. In dem Fall einfach ein paar (Milli-)Sekunden warten, bis die Verbindung wieder frei ist.

## 3.2 Bedienelemente

### 3.2.1 GPIO

### 3.2.2 PWM

### 3.2.3 ADC

## 4 Wartung

Als Wartungssystem wird die oben erstellte SD-Karte verwendet. Dazu starten sie den BeagleBone von der SD-Karte und führen sie die oben beschriebenen Installationsschritte aus.

Der package manager unter Arch Linux heißt *pacman*, über ihn können neue Pakete aus den repositories installiert bzw. aktualisiert werden.  
Ein kurzer Auszug aus der man-page zu den hier verwendeten Parametern:

Synopsis: `pacman <operation> [options] [targets]`

Parameter	Beschreibung
<i>Operations</i>	
<code>-S, -sync</code>	Synchronize packages. Packages are installed directly from the ftp servers, including all dependencies required to run the packages.
<i>Sync Options</i>	
<code>-c, -clean</code>	Remove packages that are no longer installed from the cache as well as currently unused sync databases to free up disk space.
<code>-u, -sysupgrade</code>	Upgrades all packages that are out of date.
<code>-y, -refresh</code>	Download a fresh copy of the master package list from the server(s) [...]. This should typically be used each time you use <i>-sysupgrade</i> or <i>-u</i> .

## 4.1 Backup

Da der interne Speicher des BeagleBone „nur“ 2GB beträgt, kann ohne größerem Zeitaufwand ein komplettes Speicherabbild erstellt werden. Dies hat den Vorteil, dass es beim Einspielen von Backups keine Kompatibilitätsprobleme auftreten können.

Im Ordner „scripts“ sind zwei shell-Skripte, die diesen Vorgang vereinfachen: *backup.sh* und *restore.sh*. Dabei wird das Image automatisch mit *gzip* komprimiert um Speicherplatz zu sparen. Das restore-Skript verwendet dann diese Dateien um das Speicherabbild wieder auf den BeagleBone zu kopieren.

Sollte die SD-Karte nicht genügend Speicherplatz zur Verfügung haben kann ein USB-Stick verwendet werden. Dazu einfach das USB-Laufwerk einhängen<sup>2</sup> und das entsprechende Verzeichnis als Zielverzeichnis angeben.

`backup.sh [Zielverzeichnis]`

Das Zielverzeichnis ist dabei optional. Wenn kein Parameter übergeben wird, erstellt das Skript automatisch eine Datei in der Form *backup-[timestamp].img.gz* im aktuellen Verzeichnis.

`restore.sh <Quelldatei>`

Die Quelldatei ist hier Voraussetzung.

**Hinweis:** Die Skripte verwenden intern *dd* um eine bitweise Kopie der eMMC des BeagleBone anzufertigen, zu dem ist die Quelle bzw. das Ziel immer *mmcblk1*. Daher sollten diese Skripte nur von der SD-Karte aus verwendet werden.

---

<sup>2</sup>Anleitungen hierzu gibt es in ausreichender Zahl im Internet

## 4.2 System aktualisieren

Arch Linux verwendet die rolling-release-Technik, ein System bei dem es keine großen Upgrades des gesamten Betriebssystems gibt sondern die Softwarepakete einzeln laufend aktualisiert werden.

Trotz umfangreicher Tests der Pakete kann es dennoch zu Inkompatibilitäten kommen, die ist wahrscheinlicher je mehr Pakete gleichzeitig aktualisiert werden. Daher sollte, gerade wenn das System nur selten aktualisiert wird, vorher ein vollständiges Backup gespeichert werden (s. O.).

Das System kann jederzeit via *pacman* aktualisiert werden:

```
pacman -Syu
```

## 4.3 boneserver aktualisieren

Um die boneserver-Software zu aktualisieren, aktualisieren sie zunächst ihr Kopie des git repositories und führen sie das Installationsskript erneut aus. Pakete, die bereits installiert sind, werden dabei nicht erneut installiert.

```
cd /opt/boneserver
git pull
./install.sh
```

## 4.4 System bereinigen

*pacman* speichert bei jeder Aktualisierung die alten Pakete um jederzeit auf frühere Versionen zurückgeifen zu können. Je nach Häufigkeit der Aktualisierung und gemessen an der Kapazität der eMMC, kann der der Speicher schnell knapp werden. Hierfür können alte Pakete via *pacman* in zwei Stufen gelöscht werden:

```
pacman -Sc
```

Löscht alle Paketversionen nicht mehr installierter Pakete und

```
pacman -Scc
```

löscht sämtliche nicht verwendete Pakete.