

boneserver

Installations- und Betriebsanleitung

Caspar Friedrich

22. November 2014

Inhaltsverzeichnis

1	Hardware	2
2	Installation	2
2.1	SD-Karte vorbereiten	2
2.2	Installation im internen Speicher	4
2.3	boneserver installieren	4
3	Betrieb	5
3.1	Netzwerkverbindung herstellen	5
3.2	Webinterface aufrufen	5
3.3	Bedienelemente	6
3.3.1	Digitale In- und Outputs	7
3.3.2	Pulsbreitenmodulation (PWM)	7
3.3.3	Analoge Inputs	9
3.4	Erweiterte Einstellungen	10
4	Wartung	11
4.1	Backup	11
4.2	System aktualisieren	12
4.3	boneserver aktualisieren	12
4.4	System bereinigen	12

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 sind, 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 ausgehend von einem Linux als Host-System. Stattdessen kann auch die mitgelieferte Ängström Distribution verwendet werden, die mit dem BeagleBone ausgeliefert wird.

Voraussetzungen sind die Softwarepakete *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, ...]`.

ACHTUNG 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 Kommandos ein (jeweils mit *enter* bestätigen):

Kommando	Funktion
<code>o</code>	Erzeugt eine neue Partitionstabelle
<code>n, p, 1</code>	Erzeugt eine neue (n), primäre (p), erste (1) 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 <i>W95 FAT16 (LBA)</i>

a, 1	Setzt das boot flag der ersten Partition (je nach <i>fdisk</i> -Version wird die erste Partition automatisch ausgewählt, da nur eine zur Verfügung steht)
n, p, 2	Erzeugt eine neue (n), primäre (p), zweite (2) Partition
2x <i>enter</i>	Setzt die Default-Werte für den ersten und letzten Sektor der Partition..
w	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 entpacken Sie ihn 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 halten Sie die Taste S2 gedrückt, um von der SD-Karte zu booten, während Sie die Power-Taste (S3) 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 man sich, außer zu Wartungszwecken, nicht am System anmelden 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 mittels *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 gestartete System und installieren die Pakete *wget*, *dosfstools* und *ntp*.

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

Das Paket *ntp* stellt hierbei 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 das System erneut.

2.3 boneserver installieren

Repository klonen *boneserver* ist via [GitHub](https://github.com/XMrVertigoX/boneserver)¹ 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 das Installationsskript aus.

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

Hierbei 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 abgeschlossen.

¹<https://github.com/XMrVertigoX>

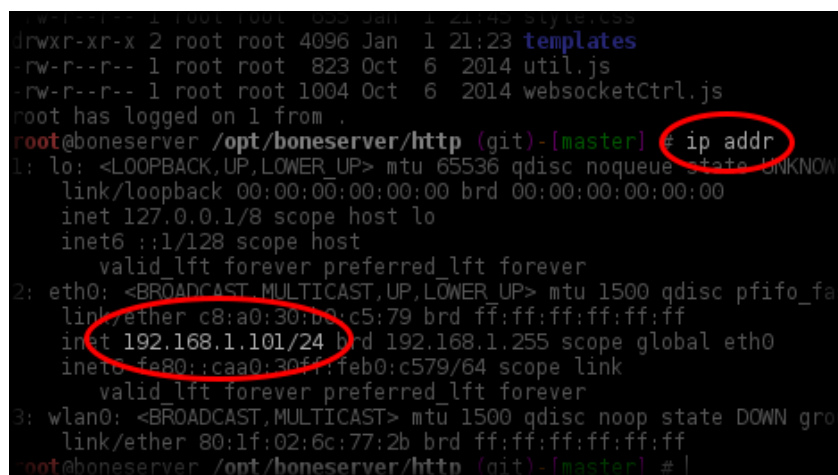
3 Betrieb

Für die Verwendung des Webinterface wird eine Netzwerkverbindung zum BeagleBone und ein Webbrowser² mit aktiviertem JavaScript vorausgesetzt.

3.1 Netzwerkverbindung herstellen

Hinweis: Die Standardkonfiguration des BeagleBone sieht den Betrieb mit einem DHCP-Server vor. Sollte dies nicht gewünscht oder möglich sein, kann über die üblichen Wege eine statische IP eingestellt werden. Anleitungen hierzu findet man im Internet.

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 die Routing-Tabelle des DHCP-Servers konsultiert werden oder in der Kommandozeile via *ip* die aktuelle Adresse der einzelnen Netzwerkadapter abgerufen werden (Abb. 1).



```
root@boneserver /opt/boneserver/http (git)-[master] # ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    link/ether c8:a0:30:b0:c5:79 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.101/24 brd 192.168.1.255 scope global eth0
        inet6 fe80::c8a0:30b0:c579:64 scope link
            valid_lft forever preferred_lft forever
3: wlan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group
    link/ether 80:1f:02:6c:77:2b brd ff:ff:ff:ff:ff:ff
root@boneserver /opt/boneserver/http (git)-[master] #
```

Abbildung 1: IP des BeagleBone abrufen

3.2 Webinterface aufrufen

Das Webinterface kann einfach über DNS-Namen oder die IP im Webbrowser aufgerufen werden. Ist die Verbindung hergestellt, wird dies durch einen grünen Haken rechts in der Titelleiste angezeigt (Abb. 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 neu geladen werden, eventuelle Konfigurationen bleiben erhalten.

²Das Webinterface verwendet *jQuery* in der Version 2.1.1, aktuelle webbrowsers sollten hier keine Probleme bereiten. Ansonsten kann die Homepage von jQuery konsultiert werden: <http://jquery.com/browser-support/>

Passwortschutz Um unbefugten Zugriff zu verhindern ist das Webinterface passwortgeschützt. Wenn Sie dieses Passwort ändern möchten, generieren Sie zunächst einen neuen Datensatz z. B. mit dem *htdigest Generator Tool*³ und tragen die Zugangsdaten in die Datei *config/lighttpd/lighttpd.user* ein. Die Default Login-Daten sind **admin/AgG7KgW4**



Abbildung 2: Webinterface verbunden

Hinweis Das Webinterface kann immer nur von einem Fenster aus aufgerufen werden. Es kann daher passieren, dass bei einem schnellen Fensterwechsel oder Neuladen der Seite die Verbindung nicht sofort hergestellt wird. In dem Fall einfach ein paar (Milli-)Sekunden warten, bis die Verbindung wieder frei ist.

3.3 Bedienelemente

Wie in Abbildung 3 gezeigt hat die Weboberfläche des boneserver drei Anzeigegruppen:

1 Verbindungsanzeige

zeigt grün, wenn die Steuereinheit verbunden ist und rot, wenn die Verbindung unterbrochen ist (vgl. Abb. 2).

2 Bedienfelder für digitale I/O, PWM und AIN

Hier findet die tatsächliche Bedienung der GPIO statt. Es gibt drei Sektionen jeweils für digitale I/O, PWM und AIN. Die Bedienung der verschiedenen Kacheln wird weiter unten beschrieben.

³ <http://jesin.tk/tools/htdigest-generator-tool/>

3 Anzeigenschalter für die einzelnen Pins

Hier können einzelne Kacheln ein- bzw. ausgeblendet werden, um die Oberfläche übersichtlicher zu gestalten und an die Arbeitsumgebung anzupassen. Diese Funktion dient ausschließlich der Übersicht, eine ausgeblendete Kachel bleibt weiterhin aktiv und kann jederzeit wieder eingeblendet werden. Diese Einstellungen bleiben auch nach einem Neustart erhalten.

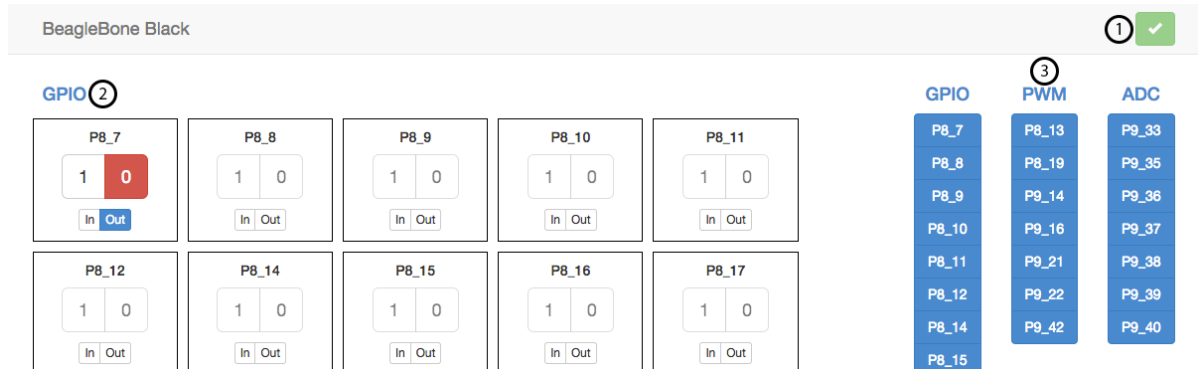


Abbildung 3: Weboberfläche

3.3.1 Digitale In- und Outputs

Die Konfigurationskachel (Abb. 4) für digitale I/O besteht aus zwei Schaltern: Betriebsrichtung und logic level.

Betriebsrichtung Jeder digitale I/O kann entweder als Input oder als Output konfiguriert werden. Dazu kann über den Wahlschalter **In/Out** jederzeit das Gewünschte ausgewählt werden.

logic level Wenn der GPIO als Output konfiguriert ist, kann hier mittels der beiden Schaltflächen, **1** und **0**, ein logisches high und ein logisches low eingestellt werden. Ist der GPIO als Input konfiguriert, ist diese Schaltfläche deaktiviert und zeigt stattdessen den Status der Leitung an. Die GPIO sind mit einem internen Pulldown-Widerstand beschaltet.

3.3.2 Pulsbreitenmodulation (PWM)

Mit Hilfe der PWM-Kacheln (Abb. 5) werden die GPIO konfiguriert, über die eine Pulsbreitenmodulation möglich ist.

Der BeagleBone stellt insgesamt sieben PWM-Ausgänge mit insgesamt vier Timern zur Verfügung. Dabei teilen sich jeweils die Pinne P8_13/19, P9_14/16 und P9_21/22 einen Timer. P9_42 hat einen exklusiven Timer. Die Ausgänge mit einem gemeinsamen

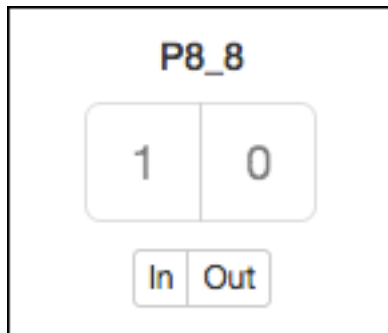


Abbildung 4: Deaktivierte GPIO-Kachel

Timer haben dementsprechend immer dieselbe Frequenz und laufen absolut synchron.

Über die Buttons **Enable** und **Disable** kann der jeweilige Ausgang aktiviert bzw. deaktiviert werden. Wenn ein PWM-Ausgang deaktiviert wird, werden alle Einstellungen bezüglich Frequenz und Pulsbreite gelöscht!

Periodendauer Über das Eingabefeld **Period** wird die Periodendauer in Nanosekunden (ns) eingestellt. Kleinster Wert ist hier 1 ns ($\hat{=}$ 1GHz) und der größte 10^9 ns (= 1s $\hat{=}$ 1Hz).

Pulsbreite Über das Eingabefeld **Duty** wird die Pulsbreite zwischen 0 und 1 eingestellt. Hier wird intern ebenfalls mit Nanosekunden gearbeitet, daher kann der tatsächliche Wert zusätzliche Nachkommastellen bekommen.

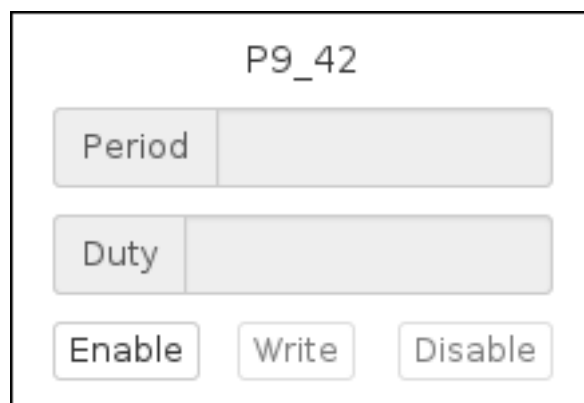


Abbildung 5: Deaktivierte PWM-Kachel

Mit **Write** werden die Parameter abgesendet.

Hinweis Der Linux Kernel arbeitet intern mit ganzen Nanosekunden, daher ist die Genauigkeit der Pulsbreite von der Höhe der Periodendauer anhängig.

Bug: Wenn beide Ausgänge eines PWM-Generators aktiviert sind, lässt sich die Frequenz bei keinem der beiden ändern. Wenn bei einem der beiden PWMs die Frequenz zunächst geändert wurde, kann der zweite Ausgang nicht verwendet werden. Dies ist ein Problem der im Hintergrund verwendeten device tree overlays und wird in nachfolgenden Versionen der Bibliothek behoben.

3.3.3 Analoge Inputs

Mit diesen Kacheln (Abb. 6) werden die Analog/Digital-Converter gesteuert und die Eingangswerte in einem Echtzeitdiagramm angezeigt.

Mit den Tasten **Start** und **Stop** wird die Aufzeichnung gestartet bzw. gestoppt. Parallel zur Anzeige werden die Messdaten aufgezeichnet. Über **Download** können sie als CSV-Datei heruntergeladen werden.

Die Taste **Delete** löscht die zu diesem Eingang gespeicherte Messreihe um Speicherplatz frei zu machen.

ACHTUNG Laden sie Messreihen immer herunter, bevor Sie sie löschen, die Messdaten sind sonst unwiederbringlich verloren!

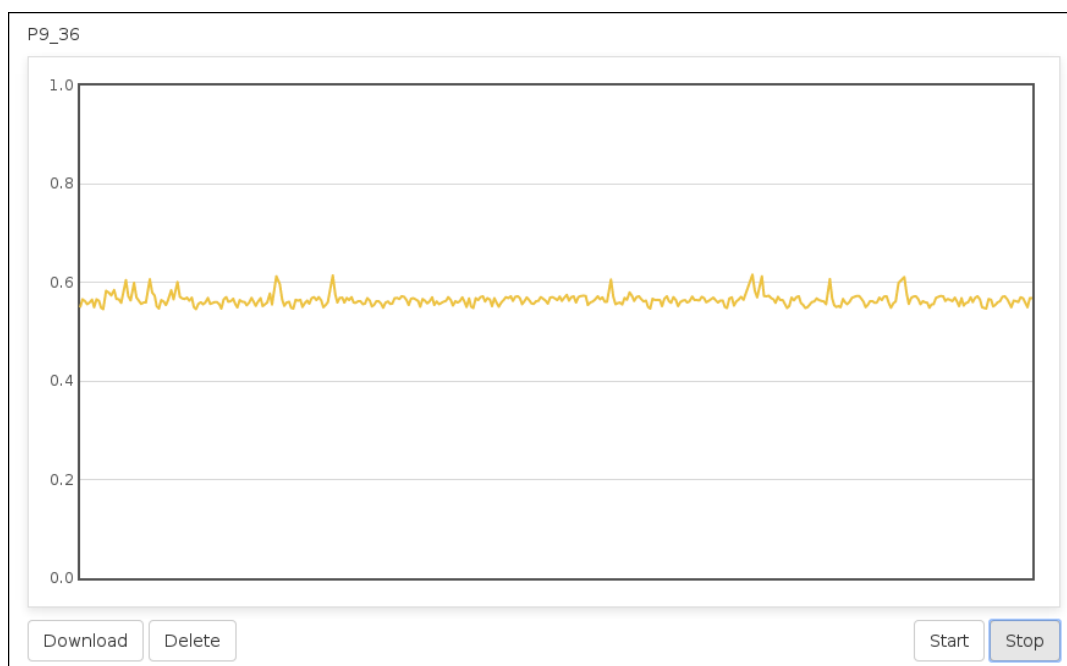


Abbildung 6: ADC-Kachel

3.4 Erweiterte Einstellungen

Zusätzlich gibt es die Möglichkeit, über die Datei *settings.json* im root-Verzeichnis des boneserver weitere Einstellungen vorzunehmen. Ist keine solche Datei vorhanden, wird die mitgelieferte *settings-default.json* verwendet. Die Parameter überschreiben die Default-Werte, es müssen daher nur abweichende Werte eingetragen werden.

Die Datei ist eine JSON⁴-Datei in der folgende Parameter eingestellt werden können:

host	IP des WebSocket Servers Dieser Wert sollte nicht verändert werden, da sonst der WebSocket Server möglicherweise nicht mehr über das Webinterface erreichbar ist. <i>default: localhost</i>
port	Netzwerk-Port des WebSocket Servers Dieser Wert sollte nicht verändert werden, da der WebSocket Server sonst nicht mehr über das Webinterface erreichbar ist. <i>default: 8081</i>
gpioSampleRate	Abtastrate der digitalen Inputs in Millisekunden Angabe der Zeit zwischen den Abfragen. Dieser Wert kann erhöht werden um die System- und Netzwerklast zu verringern. <i>default: 100</i>
adcSampleRate	Abtastrate der analogen Inputs in Millisekunden Angabe der Zeit zwischen den Abfragen. Dieser Wert kann erhöht werden, um die System- und Netzwerklast sowie um die Menge der erhobenen Messwerte zu verringern. Dies ermöglicht bei gleichem Speicherplatz längere Messreihen. <i>default: 10</i>
dataLocation	Speicherpfad für die Messdaten der ADC Hier kann ein alternativer Pfad zur Speicherung der Messdaten eingetragen werden. Es können auch externe Speicherorte wie z. B. USB-Sticks, USB-Festplatten oder Netzlaufwerke verwendet werden. <i>default: ./data</i>

Hinweis Die Datei *settings-default.json* sollte nicht verändert werden, da sonst nicht ohne weiteres ein Update durchgeführt werden kann.

⁴JavaScript Object Notation (JSON) ist in der RFC 7159 beschrieben

4 Wartung

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

Die Paketverwaltung unter Arch Linux heißt *pacman*, über sie 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 <code>-u</code> or <code>--sysupgrade</code> .

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 diese Dateien, um das Speicherabbild wieder auf den BeagleBone zu kopieren.

Sollte die SD-Karte nicht genügend Speicherplatz zur Verfügung stellen, kann ein USB-Stick verwendet werden. Dazu einfach das USB-Laufwerk einhängen⁵ 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.

⁵Anleitungen hierzu gibt es in ausreichender Zahl im Internet

```
restore.sh <Quelldatei>
```

Die Quelldatei ist hier Voraussetzung.

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

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 zu Inkompatibilitäten kommen. Dies ist wahrscheinlicher, je mehr Pakete gleichzeitig aktualisiert werden. Daher sollte, gerade wenn das System nur selten aktualisiert wird, vorher ein vollständiges Backup angelegt 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 Ihre Kopie des git Repositories und führen 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 zurückgreifen zu können. Je nach Häufigkeit der Aktualisierung und gemessen an der Kapazität der eMMC, kann der Speicher schnell knapp werden. Daher 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 verwendeten Pakete.