

Eigenes Image erstellen (VERALTET)

Diese Anleitung ist veraltet



Seit LoxBerry 3.0 nutzen wir die Images vom Projekt <https://dietpi.com/> und stellen kein eigenes Image mehr zur Verfügung. Der LoxBerry wird direkt auf einem DietPi Image aus den Quelltexten installiert. Die Erzeugung eines eigenen Images ist damit obsolete.

Veröffentliche Dein selbst erstelltes Image!

Wenn Du ein Image für eine andere Hardwareplattform erstellt hast teile Deine Arbeit bitte mit anderen! Je mehr Plattformen der Loxberry unterstützt, desto größere Verbreitung wird er finden! Gerne stellen wir Dir dazu Speicherplatz auf unserem Server bereit! Bitte nimm bei Bedarf Kontakt mit uns auf: info@loxberry.de

1. Auswahl der Distribution

Der Loxberry basiert auf der Linuxdistribution [Raspbian](#), ein speziell für den Raspberry Pi angepasstes [Debian Linux](#). Wir nutzen dazu das Lite-Image (Minimal Image) der [Raspberry Pi Foundation](#). Es ist also eine sehr gute Idee ein eigenes Image ebenfalls auf Debian Linux aufzubauen. Damit sind die Anpassungen am System erheblich einfacher zu gestalten und auch die Plugin-Schnittstelle, die auch Software für ein Plugin aus dem Distributionsrepository nachinstallieren kann, wird ohne Anpassungen weiter funktionieren.

Da Debian zu den stabilsten Linuxdistributionen gehört und eine sehr große Anzahl an Hardwareplattformen unterstützt (wenn nicht sogar die größte Anzahl von allen Distributionen überhaupt), besteht eigentlich nur in Spezialfällen die Notwendigkeit auf eine andere Distribution zu wechseln. Stell Dich in diesem Fall aber auf deutliche Mehrarbeit ein! Diese Anleitung geht von einem Debiansystem aus.

2. Softwaregrundausstattung

Plugins erwarten vom Loxberry eine auf allen Systemen identische Paket- und Softwareauswahl. Damit alle Plugins auch auf Deinem eigenen Image laufen, stelle sicher, dass Dein Image mindestens (besser: exakt) die Softwareausstattung des Original-Loxberry-Images enthält.

Zu empfehlen ist zunächst ein Minimalsystem zu installieren und später die benötigten Pakete automatisch nachinstallieren zu lassen (siehe Kapitel unten).

2.1 SSH

Um beim aktuellen Raspbian (Stretch) gleich den SSH-Zugang zu aktivieren, legt man im Rootverzeichnis der Bootpartition eine leere Datei mit dem Namen "ssh" (ohne Endung) an.

3. Systembenutzer

Auf dem Loxberry sollten nur 2 "reale" Benutzer eingerichtet sein:

Benutzername	Passwort	Bemerkung
root	loxberry	Systemadministrator
loxberry	loxberry	Standardbenutzer

Per SSH verbinden und als User "pi" auf dem Raspberry einloggen. Das Homeverzeichnis des Benutzers "loxberry" muss auf das Installationsverzeichnis der Loxberry-Scripte zeigen (siehe unter Punkt 5.).

```
KONSOLE sudo passwd //(Hier als Passwort "loxberry" setzen - es ist das  
Passwort für "root") KONSOLE su - KONSOLE adduser --uid 1001 --home  
/opt/loxberry loxberry //(Hier als Passwort "loxberry" setzen - es ist das  
Passwort für den Benutzer "loxberry")
```

```
KONSOLE usermod -a -G sudo,dialout,audio,gpio,ttty,www-data loxberry
```

Jetzt die SSH-Verbindung komplett beenden und neu Einloggen als User "loxberry":

```
KONSOLE su - (Bindestrich nicht vergessen! Eingabe des Root-Passworts)
```

```
KONSOLE deluser pi
```

```
KONSOLE delgroup pi
```

```
KONSOLE rm -r /home/pi
```

WICHTIG! Das Initialpasswort für beide Accounts muss auf "loxberry" gesetzt werden. Beide Passwörter werden bei der Ersteinrichtung des Loxberry neu gesetzt.

4. Locales

Standardlocale auf dem System sowie auch für beide oben genannten Benutzer ist `en_GB.UTF-8`.

```
KONSOLE dpkg-reconfigure locales
```

WICHTIG! Einige Skripte "grabben" den Output von Konsolenprogrammen. Daher ist es sehr wichtig, dass für beide benutzer das Standardlocale auf `en_GB.UTF-8` gesetzt wird, ansonsten könnten diese Befehle fehlschlagen! Das Standardlocale auf einem frisch installierten Raspbian Lite ist `en_GB.UTF-8`. Es ist also nicht nötig hier im ersten Schritt etwas zu ändern!

5. Loxberry-Sourcen installieren

Standardmäßig werden die Loxberry-Quelltexte unter `/opt/loxberry` installiert. Du solltest nach Möglichkeit nicht davon abweichen. Obwohl Plugin-Entwickler angehalten sind nur relative

Pfadangaben zu verwenden ist dieses nicht unbedingt sichergestellt. Am einfachsten kannst Du die Quellen direkt per *git* installieren:

```
KONSOLE apt-get update
```

```
KONSOLE apt-get install git sudo
```

```
KONSOLE cd /opt
```

```
KONSOLE rm -r /opt/loxberry
```

Dein Image sollte immer den letzten "Release" Stand haben (nicht PreRelease! Das sind Testversionen!). Um den letzten Stand herauszufinden, gehst Du auf die Webseite <https://github.com/mschlenstedt/Loxberry/releases> und schaust nach, wie der "Tag" des letzten Release heisst:



In diesem Fall hat das letzte Release den Tag "**1.0.2**". Diesen setzt man im folgenden git-Befehl ein:

```
<del>KONSOLE</del> git clone "%%https://github.com/mschlenstedt/Loxberry.git%%" -b  
**1.0.2** loxberry
```

5.1 Hardware-Architektur konfigurieren

Damit Plugins erkennen können auf welcher Hardware-Architektur sie laufen, muss diese entsprechend im Image konfiguriert werden.

Eines dieser drei Labels ist PFLICHT.

Für die Raspberry Architektur (Raspi 1, 2, 3 oder Zero):

```
KONSOLE echo "raspberry" > /opt/loxberry/config/system/is_raspberrry.cfg
```

Für andere ARM Architekturen (z.B. Odroid):

Hierfür sollte ebenfalls das Label für den Raspberry gesetzt werden, zusätzlich ein eigenes Label (siehe unten).

Für die X86 Architektur:

```
KONSOLE echo "x86" > /opt/loxberry/config/system/is_x86.cfg
```

Für die X64 Architektur:

```
KONSOLE echo "x64" > /opt/loxberry/config/system/is_x64.cfg
```

5.1.1 Eigenes Image-Label setzen

Du kannst *zusätzlich* zu dem PFLICHT-Label für dein Image ein eigenes Label setzen. *Maximale Länge dieses Tags ist 20 Zeichen.*

Zum Beispiel:

```
KONSOLE echo "VMware" > /opt/loxberry/config/system/is_vmware.cfg
```

oder

```
KONSOLE echo "Docker-Raspberry" > /opt/loxberry/config/system/is_raspdocker.cfg
```

Dein Label soll dein Image so eindeutig wie möglich identifizieren.

Das selbst definierte Label hat innerhalb LoxBerry keine Funktion. kann aber im Zuge eines LoxBerry Update für eine zukünftige, [statistische Verwendungsanzeige](#) deiner VM verwendet werden. Schreib an [Christian per PM](#), wie dein Label heißt.

5.2 Berechtigungen setzen

```
KONSOLE chown -R loxberry.loxberry /opt/loxberry
```

```
KONSOLE ln -s /opt/loxberry/system/sudoers /etc/sudoers.d
```

```
KONSOLE chmod 555 /opt/loxberry/system/sudoers
```

```
KONSOLE chown root:root /opt/loxberry/system/sudoers/lbdefaults
```

```
KONSOLE chmod 664 /opt/loxberry/system/sudoers/lbdefaults
```

6. Installation von benötigten Paketen / Deinstallation von nicht benötigten Dateien

Die Pakete des LoxBerry-Grundsystems findet man in der Datei /opt/loxberry/packages.txt Mit den LoxBerry-Quelltexten wird ein Skript mitgeliefert, welches die Pakete automatisch nachinstalliert.

Vor dem Update sollten die Paket-Informationen auf den aktuellen Stand gebracht werden (als "root"), damit während der folgenden Installation keine Fehler passieren (404 Not Found...).

```
KONSOLE apt-get update && apt-get upgrade
```

Des weiteren müssen noch ein paar Perl Pakete vor ab installiert werden (ebenfalls als "root"):

```
KONSOLE apt-get install perl libdevice-serialport-perl libio-socket-ssl-perl libwww-perl libconfig-simple-perl libfile-homedir-perl
```

Die nachfolgenden Befehle müssen als User "loxberry" ausgeführt werden! Zunächst kann man sich anschauen welche Pakete installiert werden würden:

```
KONSOLE /opt/loxberry/sbin/installpackages.pl --file /opt/loxberry/packages.txt --test
```

Wenn alles OK ist kann man die Installation automatisch starten mit:

```
KONSOLE /opt/loxberry/sbin/installpackages.pl --file /opt/loxberry/packages.txt
```

Jetzt installieren wir noch pishrink, um später das Image zu erstellen und automatisch auf die minimale Größe zu schrumpfen:

```
KONSOLE wget https://raw.githubusercontent.com/Drewsif/PiShrink/master/pishrink.sh -P /opt/loxberry/sbin
```

```
KONSOLE chmod +x /opt/loxberry/sbin/pishrink.sh
```

Besonderheit MySQL:

Entfällt. MySQL Server ist im Standardimage nicht mehr installiert.

Besonderheit ntp:

Entfällt.

Besonderheit Samba:

Der Loxberry-Benutzer muss zunächst als Root eingerichtet werden. Das Standardpasswort muss hier ebenfalls auf "loxberry" gesetzt werden:

```
KONSOLE smbpasswd -a loxberry
```

Besonderheit VIM:

Füge die folgende Option der Datei `/usr/share/vim/vim80/defaults.vim` hinzu:

```
/usr/share/vim/vim81/defaults.vim: Zeilen hinzufügen
\\ " In many terminal emulators the mouse works just fine.  By enabling it
you\\ " can position the cursor, Visually select and scroll with the
mouse.\\ if has('mouse')\\ set mouse=r\\ endif\\

\\ set viminfo='20,<1000,s1000\\
```

7. Anpassen von Systemdateien

Es müssen einige Systemdateien angepasst und neu installiert werden bzw. durch symbolische Links auf Dateien unterhalb von `/opt/loxberry` ersetzt werden.

7.1 Environment-Variablen setzen

```
KONSOLE /opt/loxberry/sbin/setenvironment.sh
```

```
KONSOLE exit
```

SSH-Session beenden und als User "loxberry" komplett neu einloggen, damit die Environment-Variablen eingelesen werden.

```
KONSOLE /opt/loxberry/bin/createconfig.pl
```

```
KONSOLE su -
```

```
KONSOLE /opt/loxberry/sbin/resetpermissions.sh
```

WICHTIG Solltest Du in Deinem Image die LoxBerry Quellen nicht nach /opt/loxberry kopiert haben (**nicht** empfohlen), musst Du den Pfad im Skript /opt/loxberry/sbin/setenvironment.sh anpassen!

Beschrieben werden dabei die Dateien /etc/environment, /system/apache2/envvars und /etc/php/7.0/apache2/conf.d/20-loxberry.ini, von wo bei Systemstart/Anmeldung persistente Environmentvariablen ausgelesen werden. Danach sollte man sich neu anmelden und prüfen, ob die Variablen funktionieren (folgender Aufruf muss /opt/loxberry ausgeben):

```
KONSOLE echo $LBHOMEDIR
```

7.2 Initscript

Entfällt. Wird automatisch per setenvironment.sh gesetzt.

7.3 Crontab

Entfällt.

7.4 Visudo

visudo: Zeile auskommentieren
Allow members of group sudo to execute any command
##%sudo ALL=(ALL:ALL) ALL

7.5 Apache / Lighttpd

Entfällt.

7.6 Network

Entfällt. Wird jetzt über setenvironment.sh gesetzt.

Info für Non-Raspberry Image-Bauer

Debian Stretch sowie auch RHEL, Fedora, CentOS usw. Haben kein ethX mehr die heißen jetzt ganz anders. Wenn man das nicht vor dem Reboot korrigiert wird das Netzwerk nicht gestartet und man steht vor der Tür.

7.7 PAM

```
/etc/pam.d/common-password: Zeilen ändern
```

```
\\ # here are the per-package modules (the "Primary" block)\\ password  
[success=1 default=ignore] pam_unix.so sha512 minlen=4\\ #password  
[success=1 default=ignore] pam_unix.so obscure sha512\\
```

7.8 Logrotate

```
/etc/logrotate.conf: Zeilen ändern
```

```
\\ # uncomment this if you want your log files compressed\\ compress\\
```

7.9 Samba

Entfällt. Wird jetzt über setenvironment.sh gesetzt.

7.10 VSFTPD

Entfällt. Wird jetzt über setenvironment.sh gesetzt.

7.11 RSYSLOG

Anpassen von Zeile 46 und 62 :

```
/etc/rsyslog.conf: Zeilen ändern
```

```
#  
# Where to place spool and state files  
#  
$WorkDirectory /tmp  
  
#  
# First some standard log files. Log by facility.  
#  
auth,authpriv.* /var/log/auth.log  
*.*;auth,authpriv.none;\n cron,daemon.none; -/var/log/syslog
```

7.12 SSMTP

Entfällt. Wird jetzt über setenvironment.sh gesetzt.

7.13 Unattended-Upgrades

KONSOLE `dpkg-reconfigure -plow unattended-upgrades` (//Automatische Upgrades aktivieren) `KONSOLE rm /etc/apt/apt.conf.d/50unattended-upgrades` `KONSOLE ln -s /opt/loxberry/system/unattended-upgrades/unattended-upgrades.conf /etc/apt/apt.conf.d/50unattended-upgrades` `KONSOLE rm /etc/apt/apt.conf.d/02periodic` `KONSOLE ln -s /opt/loxberry/system/unattended-upgrades/periodic.conf /etc/apt/apt.conf.d/02periodic` `KONSOLE chmod 660 /etc/apt/apt.conf.d/02periodic` `KONSOLE chmod 660 /etc/apt/apt.conf.d/50unattended-upgrades` `KONSOLE chown loxberry.loxberry /etc/apt/apt.conf.d/02periodic` `KONSOLE chown loxberry.loxberry /etc/apt/apt.conf.d/50unattended-upgrades` **==== 7.14 Hostname setzen ====**
`KONSOLE echo "loxberry" > /etc/hostname` | 127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters | Ein Eintrag mit der IP 127.0.**1**.* mit Hostname muss entfernt werden, wenn vorhanden! **==== 7.15 LoxBerry Update beim ersten Reboot ausführen ====**
`KONSOLE touch /boot/do_lbupdate` **==== 7.16 Image Resize ====** Info für Non-Raspberry Image-Bauer Auf der Raspberry Plattform wird das Image automatisch beim ersten Reboot auf die maximale SD-Kartengröße expandiert. Viele Funktionen (z. B. Swapping) hängen von der erfolgreichen Expandierung ab. Wenn Du diese Funktion nicht benötigst (z. B. in einer VM), dann führe bitte folgenden Befehl aus. Dieser zeigt dem System an, dass die SD-Karte "scheinbar" bereits expandiert wurde. `KONSOLE touch /boot/rootfsresized` **===== 8. Aufräumen und Dateisystem weitestgehend ReadOnly umbauen =====** Das hat den Vorteil, dass eine Beschädigung des Dateisystems minimiert wird, wenn der Raspberry Pi ohne einen richtigen Shutdown ausgeschaltet wird (sprich: wenn einfach der Stecker gezogen wird). zudem wird die SD-Karte entlastet, was ebenfalls die Lebensdauer verlängert. Die nachfolgenden Schritte in diesem Abschnitt müssen alle als User root ausgeführt werden. Zunächst löschen wir alle unnötigen Dateien auf dem System: `KONSOLE rm -rf /var/cache/apt/archives/*`
`KONSOLE sudo /opt/loxberry/sbin/createskelfolders.pl` `KONSOLE rm -rf /opt/loxberry/.git*` Nun verlagern wir einige Verzeichnisse in eine Ramdisk und konfigurieren SWAP so, dass es nur im absoluten Notfall verwendet wird: `KONSOLE echo "vm.swappiness = 1" > /etc/sysctl.d/97-swappiness.conf` `KONSOLE rm -rf /var/lib/dhcp/ /var/spool /var/lock` `KONSOLE rm -rf /var/swap` `KONSOLE rm -rf /var/log/*` `KONSOLE rm -rf /var/tmp/*` `KONSOLE rm -rf /tmp/*` `KONSOLE ln -s /tmp /var/lib/dhcp` `KONSOLE ln -s /tmp /var/spool` `KONSOLE ln -s /tmp /var/lock` | /lib/init/vars.sh: Option ändern | | `FSCKFIX=yes` | Als allerletzten Schritt werden sicherheitshalber noch 2 Dateien gelöscht, falls entweder der Wizard aus Versehen gestartet wurde oder während der Imageerstellung zwischendurch gebootet wurde: `KONSOLE rm -f /opt/loxberry/config/system/loxberryid.cfg`
`KONSOLE rm -f /opt/loxberry/data/system/wizard.dat` **===== 9. Image erstellen =====** Das System herunterfahren: `KONSOLE poweroff` Jetzt ein Linuxsystem booten (kann auch ein Rettungssystem wie Knoppix o. ä. sein), die SD-Karte mounten und von dort das **pishrink**-Script (siehe unter 6.) auf das aktuelle System kopieren und mit `chmod +x` ausführbar machen. Anschließend die SD-Karte auf das aktuelle Linuxsystem klonen (Achtung! Pfadangaben anpassen!

Pfad der SD-Karte herausfinden (die Größe gibt einen guten Hinweis welches Device die SD-Karte ist) und anschließend das Image der SD-Karte erstellen: KONSOLE sudo lsblk KONSOLE sudo dd if=/dev/mmcblk0 of=~/.loxberry-image-rasppi-1.0.0_gandalf.img //(Pfadangaben anpassen!)

Dann kann das Image auf das Minimum zusammengeschrumpft werden:

```
KONSOLE sudo pishrink.sh -s loxberry-image-rasppi-1.0.0_gandalf.img loxberry-image-rasppi-1.0.0_gandalf_shrunked.img
```

10. Image aktualisieren

Kleinere Änderungen

Nachträgliches mounten und verändern des Images (weitere Infos unter <https://raspberrypi.stackexchange.com/questions/13137/how-can-i-mount-a-raspberry-pi-linux-distro-image>), wenn keine größeren Änderungen notwendig sind - d. h. kein weiterer (größerer) Speicherbedarf benötigt wird:

```
KONSOLE fdisk -l loxberry-image-rasppi-1.0.0_gandalf.img
```

Die Ausgabe sieht wie folgt aus:

```
Disk loxberry-image-rasppi-1.0.0_gandalf.img: 3.2 GiB, 3389346304 bytes,
6619817 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x4bd27daf

Device                                Boot Start      End Sectors  Size Id
Type
loxberry-image-rasppi-1.0.0_gandalf.img1      8192    93236    85045 41.5M c
W95 FAT32 (LBA)
loxberry-image-rasppi-1.0.0_gandalf.img2     94208 4522664 4428457  2.1G 83
Linux
```

Die Bootpartition ist die erste angezeigte Partition, die Root-Partition ist die zweite angezeigte Partition. Man benötigt jeweils den Wert unter "Start" der Partition, die man mounten möchte. Dieser Wert muss in den nachfolgenden Befehl bei offset= eingesetzt werden. Der Partitionstyp muss ebenfalls angepasst werden. Die Rootpartition ist ext4, die Boot-Partition vfat.

```
KONSOLE mount -v -o offset=$((94208 * 512)) -t ext4 loxberry-image-rasppi-1.0.0_gandalf.img /mnt
```

Größere Änderungen

Wenn man mehr Speicherplatz im Image benötigt oder man direkt Befehle "im Image" ausführen möchte, z. B. ein Systemupgrade, muss das Image vor dem Mounten zunächst vergrößert werden. Nachdem man seine Arbeiten abgeschlossen hat, wird es wieder auf die minimale Größe verkleinert. Wir vergrößern das Image zunächst um 1 GB:

```
KONSOLE dd if=/dev/zero of=temp_image bs=1M count=1000
```

```
KONSOLE cat temp_image >> loxberry-image-rasppi-1.0.0_gandalf.img
```

```
KONSOLE rm temp_image
```

Anschließend wird es über ein Loop-Device gemountet und vergrößert. Zunächst findet man wieder den Start-Sektor der 2. Partition (Rootpartition) heraus:

```
KONSOLE fdisk -l loxberry-image-rasppi-1.0.0_gandalf.img
```

Die Ausgabe sieht wie folgt aus:

```
Disk loxberry-image-rasppi-1.0.0_gandalf.img: 3.2 GiB, 3389346304 bytes,
6619817 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x4bd27daf
```

Device Type	Boot	Start	End	Sectors	Size	Id
loxberry-image-rasppi-1.0.0_gandalf.img1 W95 FAT32 (LBA)		8192	93236	85045	41.5M	c
loxberry-image-rasppi-1.0.0_gandalf.img2 Linux		94208	4522664	4428457	2.1G	83

Die Rootpartition (2. Partition) wird nun über ein Loop-Device ins System eingebunden und vergrößert. Den ermittelten Startsektor muss man wieder in den unten stehenden Befehl einsetzen:

```
KONSOLE losetup --offset $((94208 * 512)) /dev/loop0 loxberry-image-
rasppi-1.0.0_gandalf.img
```

```
KONSOLE e2fsck -f /dev/loop0
```

```
KONSOLE resize2fs -f /dev/loop0
```

```
KONSOLE losetup -d /dev/loop0
```

Jetzt kann das Image bzw. die Rootpartition gemountet werden und ein "Chroot" durchgeführt werden:

```
KONSOLE mount -v -o offset=$((94208 * 512)) -t ext4 loxberry-image-
```

```
rasppi-1.0.0_gandalf.img /mnt
```

```
KONSOLE chroot /mnt
```

Nun kann das Image aktualisiert werden:

```
KONSOLE rm /opt/loxberry/sbin/loxberryupdate*.pl
```

```
KONSOLE wget https://raw.githubusercontent.com/mschlenstedt/Loxberry/master/sbin/loxberryupdate.pl -P /opt/loxberry/sbin
```

```
KONSOLE wget https://raw.githubusercontent.com/mschlenstedt/Loxberry/master/sbin/loxberryupdatecheck.pl -P /opt/loxberry/sbin
```

```
KONSOLE chmod +x /opt/loxberry/sbin/loxberryupdate*.pl
```

```
KONSOLE . /etc/environment && /opt/loxberry/sbin/loxberryupdatecheck.pl querytype=release update=1 nofork=1 nodiscspacecheck=1 nobackup=1
```

```
KONSOLE /usr/bin/apt-get -q -y update
```

```
KONSOLE /usr/bin/apt-get -q -y --fix-broken install
```

```
KONSOLE /usr/bin/apt-get -q -y upgrade
```

Aufräumen:

```
KONSOLE rm ~/.bash_history ~/.cache ~/.vim* ~/.wget* ~/dead.letter /var/swap
```

```
KONSOLE rm -r /opt/loxberry/log/system/loxberryupdate/* /var/cache/apt/archives/* /boot/* /opt/loxberry/system/daemons/system/99-updatereboot* /opt/loxberry/log/system_tmpfs/* /opt/loxberry/data/system/notifications_sqlite.dat
```

Am Ende noch einmal Kontrolle der `/opt/loxberry/config/general.cfg`, ob es ein fehlgeschlagenes updateskript gegeben hat. `FAILED_UPDATE` dann aus der `general.cfg` entfernen und gegebenenfalls manuell reparieren.

Sollen auch der Kernel und die Firmwares für den RaspBerry mit aktualisiert werden, muss nun noch `rpi-update` ausgeführt werden. Dabei werden alle Dateien, die für das `/boot`-Verzeichnis bestimmt sind, zunächst in ein temporäres Verzeichnis aktualisiert. Die Root-Partition wird direkt aktualisiert. Anschließend wird die Boot-Partition des Images gemountet und die im temporären Verzeichnis gespeicherten Dateien zurückgespielt. Dieser Umweg ist notwendig, da ein Image nur einmal gemountet werden kann. Somit ist zeitgleich nur eine Partition aus dem Image erreichbar.

```
KONSOLE ROOT_PATH=/mnt BOOT_PATH=/temp_boot SKIP_BACKUP=1 SKIP_WARNING=1 WANT_PI4=1 SKIP_CHECK_PARTITION=1 BRANCH=stable /usr/bin/rpi-update
```

```
KONSOLE umount /mnt
```

Nun wird die Boot-Partition des Images nach `/mnt` gemountet (siehe wieder Kapitel 9). Anschließend können alle Dateien aus dem temporären Boot-Verzeichnis auf die Original-Boot-Partition des Images kopiert werden:

```
KONSOLE cp -r /temp_boot/* /mnt
```

```
KONSOLE umount /mnt
```

Dann kann das Image auf das Minimum zusammengeschrumpft werden:

```
KONSOLE sudo pishrink.sh -s loxberry-image-rasppi-1.0.0_gandalf.img loxberry-image-rasppi-1.0.0_gandalf_shrunked.img
```

From:

<https://wiki.loxberry.de/> - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:

https://wiki.loxberry.de/entwickler/advanced_developers/eigenes_image_erstellen_loxberry_v1x/start

Last update: **2023/03/07 05:55**