

1. Software-Installation auf die SD-Karte

1. Betriebssystem-Sammlung Noobs aktuell von der [Raspberry-Seite](#) herunterladen (NOOBS_v1_5_0.zip ca. 1.02GB)
2. Sd-Karte mit dem SD-Formater formatieren: [SDFormatterv4](#)
3. Entpackte Daten auf SD-Karte kopieren und in Raspberry PI einlegen, dann Raspberry PI starten
4. Raspbian auswählen und somit Betriebssystem auf der SD-Karte installieren (dauert ca. 10 Minuten)
5. nach dem Start in der Menüleiste oben das Terminal starten
6. im Terminal durch "**sudo raspi-config**" das Konfigurationsprogramm starten (Bem.: - (minus-Zeichen) liegt auf der ß-Taste)
7. Punkt 5 Internationalisation Options anwählen=> I1 Change Locals
=> **de_DE.UTF-8 UTF-8** auswählen (zuerst space-Taste, dann "OK")
=> system environment: **de_DE.UTF-8** auswählen
8. Punkt 5 Internationalisation Options anwählen=> I2 Change Timezone
=> **Europe** => **Berlin**
9. Punkt 5 Internationalisation Options anwählen=> I3 Keyboard Layout
=> **Generic 105-key (Intl)** => **Other** => **German** => **German** => **Right Alt** (AltGr) => **No compose key** => **No**

2. Anpassungen für das 7 Zoll Touchdisplay

siehe auch:

[Quick Start Einrichtung: Display-Set LS-7T bei Raspbian \(Pollin Artnr.: 120964\)](#)

1. Anpassung für das TFT-Display: `sudo nano /boot/config.txt`

```
framebuffer_width=1024
framebuffer_height=600
framebuffer_depth=16
hdmi_group=2
hdmi_mode=87
hdmi_cvt=1024 600 60 30 3 0 0 0
```

mit `sudo reboot` neu starten

2. Anpassungen für den Touchscreen:

```
sudo apt-get update
```

```
sudo apt-get upgrade (eventuell mit J bestätigen)
```

```
sudo rpi-update
```

mit `sudo reboot` neu starten

Herunterladen der Kalibrator-Software:

<https://packages.debian.org/jessie/xinput-calibrator>

[Quellcode: [xinput-calibrator](#)]

Paket: `xinput-calibrator (0.7.5+git20140201-1)`

Generic touchscreen calibration program for X.Org

xinput-calibrator is a tool to calibrate touchscreens under X.Org. Its features include:

- work for any Xorg driver (use Xinput to get axis valuator);
- output the calibration as Xorg.conf, HAL policy and udev rule;
- support advanced driver options, such as Evdev's dynamic calibration;
- have a very intuitive GUI (normal X client);

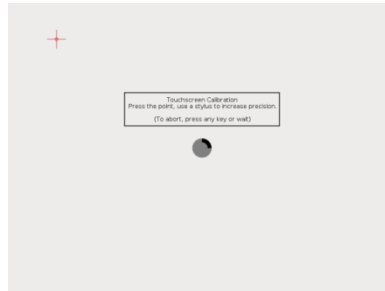
xinput-calibrator_0.7.5+git20140201-1_armhf.deb über link **armhf** herunterladen

xinput-calibrator herunterladen

Architektur	Paketgröße
amd64	42.3 kB
arm64	43.0 kB
armel	41.5 kB
armhf	40.3 kB
i386	41.7 kB
mips	42.2 kB
mipsel	42.5 kB
powerpc	42.2 kB
ppc64el	43.5 kB
s390x	42.8 kB

```
sudo dpkg -i -B xinput-calibrator_0.7.5-1_armhf.deb
```

Touch-Screen-Programm im Desktop über "**Menu => Einstellungen => Touchscreen Kalibrieren**" starten
und Touchscreen durch drücken auf die 4 "Eckpunkte" kalibrieren.



Der Inhalt des Ausgabebildschirms wird unten noch gebraucht.

```
pi@octopi ~ $ clear
pi@octopi ~ $ ./xinput-calibrator-0.7.5/src/xinput_calibrator
Warning: multiple calibratable devices found, calibrating last one (eGalax Inc. Touch)
use --device to select another one.
Calibrating EVDEV driver for "eGalax Inc. Touch" id=10
current calibration values (from Xinput): min_x=34, max_x=1989 and min_y=98, max_y=1954

Doing dynamic recalibration:
Setting new calibration data: 33, 1990, 90, 1974

--> Making the calibration permanent <--
copy the snippet below into '/etc/X11/xorg.conf.d/99-calibration.conf'
Section "InputClass"
    Identifier      "calibration"
    MatchProduct    "eGalax Inc. Touch"
    Option          "Calibration" "33 1990 90 1974"
EndSection
pi@octopi ~ $
```

```
sudo mkdir /etc/X11/xorg.conf.d
```

Nun die Kalibrierung speichern, in dem man die Konfigurations-Datei anpasst durch:

```
sudo nano /etc/X11/xorg.conf.d/99-calibration.conf
```

Jetzt den Inhalt des Ausgabebildschirms (siehe oben) von "Section" bis "Endsection" in die Datei **99-calibration.conf** kopieren:

```
Section "InputClass"
    Identifier "calibration"
    MatchProduct "eGalax Inc. Touch"
    Option "Calibration" "EIGENE WERTE"
EndSection
```

Mit Ctrl-O und Ctrl-X den Editor schließen, der Touch-Screen ist nun konfiguriert und einsatzbereit.

3. Root-Filesystem auf USB-Stick umstellen

1. `sudo mkfs.ext4 -L rootfs /dev/sda1`
(eventuell mit j bestätigen bzw. zuerst mit `sudo umount /dev/sda1` unmounten)
2. `sudo mount /dev/sda1`
oder durch Stick ziehen und einstecken anzeigen lassen wo Stick gemountet ist
3. `sudo rsync -axv / /dev/sda1`
(oder statt `/dev/sda1` den "Mounting-Point" als Ziel)
4. Ändern der Datei `cmdline.txt` das vom USB-Stick gebootet wird:
`sudo nano /boot/cmdline.txt`
`dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200`
`console=tty1 root=/dev/sda1 rootfstype=ext4 elevator=deadline`
`rootwait rootdelay=5`

(Wenn Sie schon die serielle Konsole deaktiviert haben, ist die Zeile ggf. kürzer.)

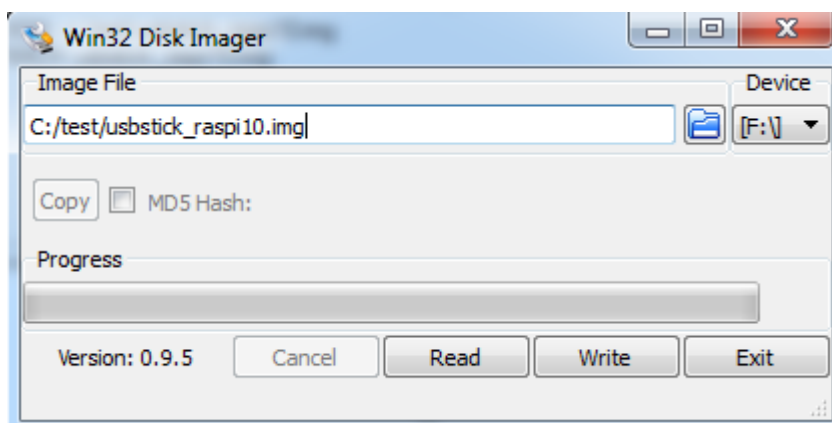
5. Ändern der Datei `fstab`, damit auch beim Neustart vom USB-Stick gebootet wird:

```
sudo nano /etc/fstab
```

<code>proc</code>	<code>/proc</code>	<code>proc</code>	<code>defaults</code>	<code>0</code>	<code>0</code>
<code>/dev/sda1</code>	<code>/</code>	<code>ext4</code>	<code>defaults,noatime</code>	<code>0</code>	<code>1</code>
<code>/dev/mmcblk0p1</code>	<code>/boot</code>	<code>vfat</code>	<code>defaults</code>	<code>0</code>	<code>2</code>
<code>#/dev/mmcblk0p2</code>	<code>/</code>	<code>ext4</code>	<code>Defaults,noatime</code>	<code>0</code>	<code>1</code>

4. Sicherheitskopien von SD-Karte und USB-Sticks erzeugen

Mit dem Programm [Win32Imager](#) können Images von SD-Karten und USB-Sticks erzeugt werden:



Erzeugen einer Sicherheitskopie der SD-Karte: z.B. `RBSTouch11.img`

Erzeugen einer Sicherheitskopie des USB-Sticks: z.B. `usbstick_raspi10.img`