

Installation eines 3,5“-TFT am Raspberry Pi B2

Vorwort

Diese Dokumentation bezieht sich auf das TFT der Firma Waveshare, welches bei z.B. Ebay und Amazon angeboten wird. Meistens sieht man es nicht sofort, um welches Display es sich handelt. Allerdings waren bei meinen Käufen auf Ebay alle Display diese Sorte. Generell dürfte die Anleitung auch auf andere Displays übertragbar sein. Die Installation der Treiber und dessen Name weichen dann höchstwahrscheinlich ab. Das wäre in diesem Fall der Name **flexfb**. Dieser dürfte mit dem richtigen Treiber in der Meldung von **dmesg** (s.u.) zu finden sein.

73, de DL5KV (Klaus)

Allgemeine Informationen

Falls Sie auf der Linux-Konsole Erfahrungen haben, können Sie dieses Kapitel überspringen.

In dieser Dokumentation werden einige Konfigurationsdateien editiert. Ich nutze hierfür den Editor **nano**, da er einfach zu bedienen ist und bereits in den Distributionen enthalten ist. Wichtig sind hier 3 Tastaturkürzel:

Strg+**O** Datei speichern

Strg+**W** Suchen

Strg+**X** Editor beenden. Wenn ungespeicherte Änderungen vorhanden sind, fragt der Editor, ob diese gespeichert werden sollen

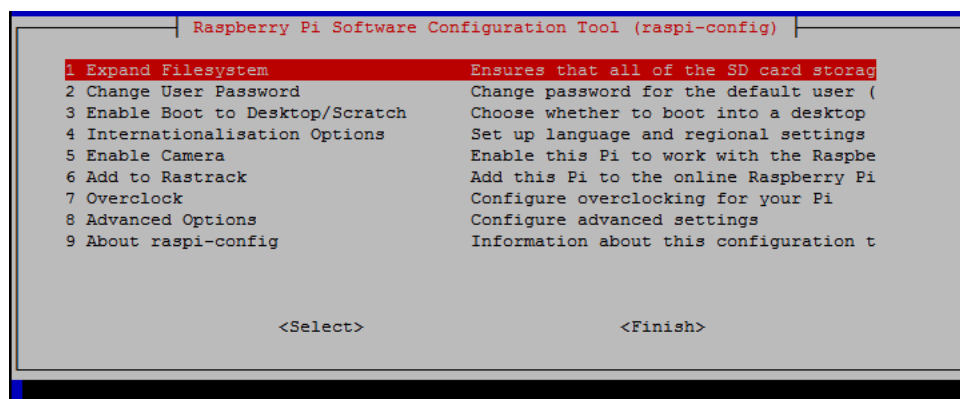
In der Konsole lässt sich ein laufender Befehl mit **Strg**+**C** abbrechen. Das ist z.B. nötig, wenn die grafische Oberfläche, oder **fbcp** ausprobiert wird, da die Konsole dann auf keinen weiteren Befehl reagiert.

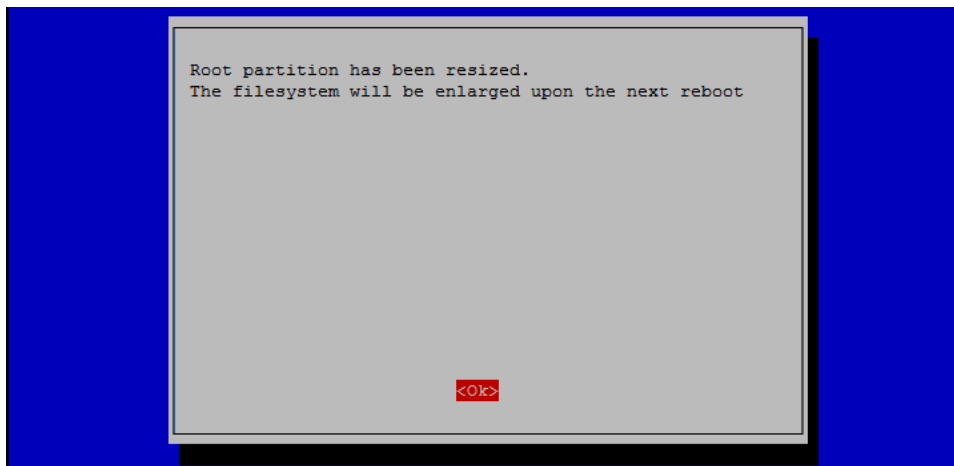
Image aufspielen

Zuerst muss ein Raspbian auf die SD-Karte kopiert werden. Wie das geht, soll nicht Teil dieser Dokumentation sein. Nachdem das Image auf die SD-Karte gespielt wurde, müssen wir mit

```
sudo raspi-config
```

und Partition vergrößern





Danach starten wir den Raspberry neu.

SPI aktivieren

Da das Display per SPI angesprochen wird, muss dieser auch aktiviert werden.

```
sudo nano /boot/config.txt
```

darin die folgende Zeile auskommentieren oder rein schreiben:

```
dtoverlay=spi=on
```

Fbtf installieren

Zuerst muss eine neue Firmware installiert werden:

```
sudo rpi-update  
sudo reboot
```

```
pi@raspberrypi ~ $ sudo REPO_URI=https://github.com/notro/rpi-firmware rpi-update  
*** Raspberry Pi firmware updater by Hexxeh, enhanced by Andrews and Dom  
*** Performing self-update  
*** Relaunching after update  
*** Raspberry Pi firmware updater by Hexxeh, enhanced by Andrews and Dom  
*** Downloading specific firmware revision (this will take a few minutes)  
% Total    % Received % Xferd Average Speed   Time    Time     Time  Current  
           Dload  Upload   Total   Spent    Left  Speed  
100 167    0 167    0    0   388      0 --:--:-- --:--:-- --:--:--   513  
100 47.9M 100 47.9M    0    0 1347k      0 0:00:36 0:00:36 --:--:-- 1023k  
*** Updating firmware  
*** Updating kernel modules  
*** depmod 4.0.7+  
*** depmod 4.0.7-v7+  
*** Updating VideoCore libraries  
*** Using HardFP libraries  
*** Updating SDK  
*** Running ldconfig  
*** Storing current firmware revision  
*** Deleting downloaded files  
*** Syncing changes to disk  
*** If no errors appeared, your firmware was successfully updated to b2f5782d2a61cdfa1967942d34eae48900266ae1  
*** A reboot is needed to activate the new firmware  
pi@raspberrypi ~ $
```

Ausprobieren, ob Treiber funktioniert:

```
sudo modprobe fbtf_device name=flexfb  
sudo dmesg
```

```
[ 71.161297] fbtft: module is from the staging directory, the quality is unknown, you have been warned.
[ 71.167211] fbtft_device: module is from the staging directory, the quality is unknown, you have been warned.
[ 71.168500] fbtft_device: SPI devices registered:
[ 71.168528] fbtft_device: spidev spi0.0 500kHz 8 bits mode=0x00
[ 71.168545] fbtft_device: spidev spi0.1 500kHz 8 bits mode=0x00
[ 71.168557] fbtft_device: 'fb' Platform devices registered:
[ 71.168589] fbtft_device: soc:fb id=-1 pdata? no
[ 71.168649] fbtft_device: Deleting spi0.0
[ 71.169229] fbtft_device: GPIOs used by 'flexfb':
[ 71.169250] fbtft_device: 'reset' = GPIO25
[ 71.169262] fbtft_device: 'dc' = GPIO24
[ 71.169273] fbtft_device: SPI devices registered:
[ 71.169288] fbtft_device: spidev spi0.1 500kHz 8 bits mode=0x00
[ 71.169303] fbtft_device: flexfb spi0.0 32000kHz 8 bits mode=0x00
pi@raspberrypi ~ $
```

Sollte es damit nicht gehen, müsste eine ältere Version mit

```
sudo REPO_URI=https://github.com/notro/rpi-firmware rpi-update
```

installiert werden.

LCD installieren

Nun müssen wir uns den Treiber holen und installieren. Leider ist das Zertifikat der Webseite abgelaufen, weshalb wir das Zertifikat ignorieren müssen.

```
LCD-show.tar.gz holen
wget --no-check-certificate https://ecksteining.de/Datasheet/LCD-show.tar.gz
tar xfvz LCD-show.tar.gz
cd LCD-show
sudo dpkg -i xinput-calibrator_0.7.5-1_armhf.deb
./LCD35-show
```

Nach dem Neustart:

```
sudo dmesg|grep graphics
```

Es sollte ein Eintrag erscheinen:

```
pi@raspberrypi ~ $ sudo dmesg|grep graphics
[ 8.493496] graphics fb1: flexfb frame buffer, 480x320, 300 KiB video memory, 4 KiB DMA
buffer memory, fps=20, spi0.0 at 16 MHz
pi@raspberrypi ~ $
```

X-Server testen:

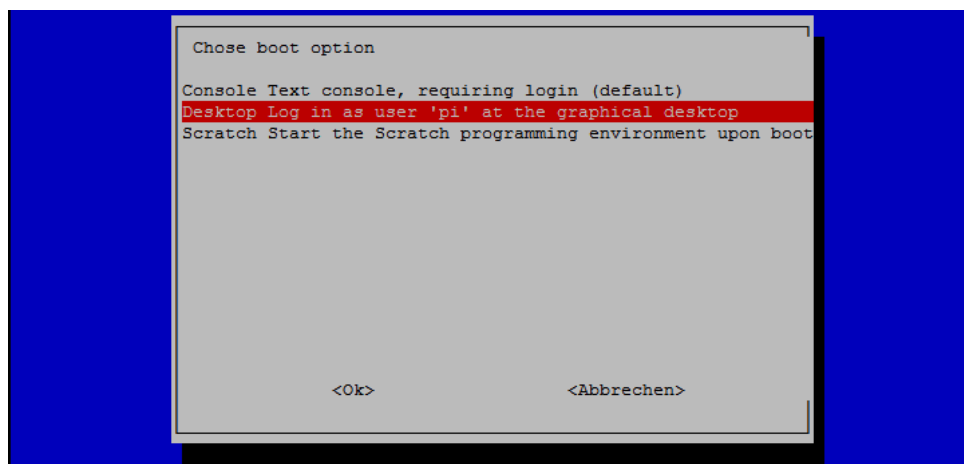
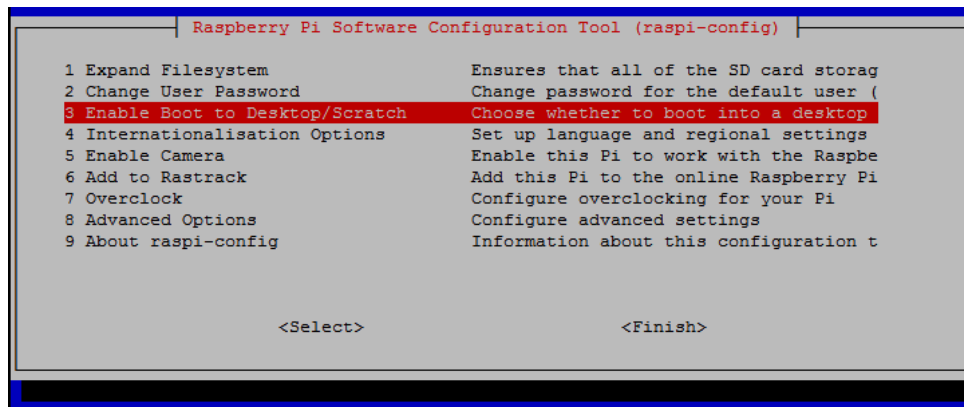
```
FRAMEBUFFER=/dev/fb1 startx
```

Wenn alles funktioniert, können wir den Autostart des X-Servers einrichten.

X-Server autostart

Start der grafischen Oberfläche aktivieren:

```
sudo raspi-config
```



Raspberry neu starten

Diagnose, Test und Kalibrierung

Falls der Touchscreen nicht funktioniert, kann man mit dem Tool **evtest** schauen, ob er überhaupt erkannt wird:

```
sudo apt-get -y install evtest tslib libts-bin
```

Nun kann man die Eingabegeräte alle ausprobieren:

```
evtest /dev/input/eventX
```

Dort kann man am Anfang der Ausgabe sehen, welches Gerät das ist und Events beobachten

```
pi@raspberrypi ~ $ evtest /dev/input/event3
Input driver version is 1.0.1
Input device ID: bus 0x0 vendor 0x0 product 0x0 version 0x0
Input device name: "ADS7846 Touchscreen"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 330 (BTN_TOUCH)
  Event type 3 (EV_ABS)
    Event code 0 (ABS_X)
      Value    209
      Min      200
      Max     3900
    Event code 1 (ABS_Y)
      Value    834
      Min      200
      Max     3900
    Event code 24 (ABS_PRESSURE)
      Value     0
      Min       0
      Max     255
Properties:
Testing ... (interrupt to exit)
█
```

Mit dieser Information können wir auch einen symbolischen Link anlegen.

```
sudo nano /etc/udev/rules.d/95-stmpe.rules
```

Dort legen wir folgenden Eintrag an:

```
SUBSYSTEM=="input", ATTRS{name}=="<Gerätename>",  
ENV{DEVNAME}=="*event*", SYMLINK+="input/touchscreen"
```

Als Gerätename kann auch ein Teil verwendet werden, in diesem Beispiel „ADS7846*“

Und starten den Raspberry neu.

Dann sollte das Device existieren:

```
ls -la /dev/input
```

```
pi@raspberrypi ~ $ ls -la /dev/input/  
insgesamt 0  
drwxr-xr-x  4 root root    260 Jan  1  1970 .  
drwxr-xr-x 13 root root   3360 Sep 25 12:05 ..  
drwxr-xr-x  2 root root    220 Jan  1  1970 by-id  
drwxr-xr-x  2 root root    160 Jan  1  1970 by-path  
crw-rw----  1 root input 13, 64 Jan  1  1970 event0  
crw-rw----  1 root input 13, 65 Jan  1  1970 event1  
crw-rw----  1 root input 13, 66 Jan  1  1970 event2  
crw-rw----  1 root input 13, 67 Jan  1  1970 event3  
crw-rw----  1 root input 13, 63 Jan  1  1970 mice  
crw-rw----  1 root input 13, 32 Jan  1  1970 mouse0  
crw-rw----  1 root input 13, 33 Jan  1  1970 mouse1  
crw-rw----  1 root input 13, 34 Jan  1  1970 mouse2  
lrwxrwxrwx  1 root root     6 Jan  1  1970 touchscreen -> event3  
pi@raspberrypi ~ $
```

Nun können wir testen, ob der Touchscreen vollkommen funktioniert:

```
sudo TSLIB_FBDEVICE=/dev/fb1 TSLIB_TSDEVICE=/dev/input/touchscreen  
ts_calibrate
```

```
sudo TSLIB_FBDEVICE=/dev/fb1 TSLIB_TSDEVICE=/dev/input/touchscreen  
ts_test
```

Nun kann das Touchscreen endgültig kalibriert werden:

```
DISPLAY=:0.0 xinput_calibrator
```

Die angezeigten Werte müssen nun von Hand in die Datei **/etc/X11/xorg.conf.d/99-calibration.conf** eingetragen werden.

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

```
GNU nano 2.2.6      Datei: /etc/X11/xorg.conf.d/99-calibration.conf      Verändert  
  
Section "InputClass"  
    Identifier      "calibration"  
    MatchProduct    "ADS7846 Touchscreen"  
    Option "Calibration"    "3953 103 202 3955"  
    Option "SwapAxes"      "1"  
EndSection
```

Klon des HDMI-Monitors einrichten

Bisher haben wir auf dem TFT ein eigenes Bild. Soll stattdessen ein Klon des HDMI-Monitors erscheinen, müssen wir noch etwas basteln. Das Tool **fbcp** erlaubt, den Inhalt von **/dev/fb0** (HDMI) auf **/dev/fb1** (TFT) zu kopieren

Zuerst müssen wir uns **fbcp** holen

```
sudo apt-get -y install cmake
git clone https://github.com/tasanakorn/rpi-fbcp
cd rpi-fbcp
mkdir build
cd build
cmake ..
make
sudo install fbcp /usr/local/bin/fbcp
```

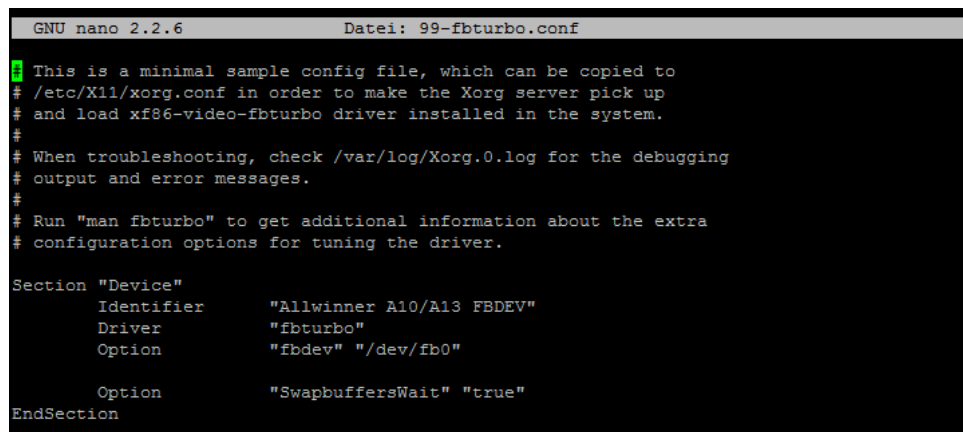
Jetzt können wir mit

```
fbcp
```

ausprobieren, ob es bereits läuft. Es sollte bereits eine Kopie der Konsole erscheinen.

Nun ändern wir in der Datei `/usr/share/X11/xorg.conf.d/99-fbturbo.conf` `fb1` in `fb0`

```
sudo nano /usr/share/X11/xorg.conf.d/99-fbturbo.conf
```



```
GNU nano 2.2.6      Datei: 99-fbturbo.conf

This is a minimal sample config file, which can be copied to
# /etc/X11/xorg.conf in order to make the Xorg server pick up
# and load xf86-video-fbturbo driver installed in the system.
#
# When troubleshooting, check /var/log/Xorg.0.log for the debugging
# output and error messages.
#
# Run "man fbturbo" to get additional information about the extra
# configuration options for tuning the driver.

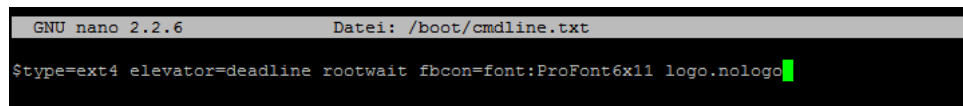
Section "Device"
    Identifier      "Allwinner A10/A13 FBDEV"
    Driver          "fbturbo"
    Option          "fbdev"  "/dev/fb0"
    Option          "SwapbuffersWait" "true"
EndSection
```

Damit wird der Start des X-Servers wieder auf den HDMI-Monitor verschoben.

Starte man nun in einer Konsole **fbcp**, und wechselt mit `[Alt]+[F1]` zur Konsole (man sieht keine Änderung), sieht man aber, dass die Konsole beim Start auf dem TFT liegt und bei jedem Tastendruck in der Konsole auf dem Monitor die Anzeige auf dem TFT flackert. Wir wechseln wieder mit `[Alt]+[F7]` zurück.

Wir entfernen wir in der Datei `/boot/cmdline.txt` den Eintrag **fbcon=map:10**

```
sudo nano /boot/cmdline.txt
```



```
GNU nano 2.2.6      Datei: /boot/cmdline.txt

$type=ext4 elevator=deadline rootwait fbcon=font:ProFont6x11 logo.nologo
```

Damit **fbcp** nun immer gestartet wird, trägt man in der Datei `/etc.rc.local` folgende Zeilen ein:

```
su -c "fbcp &"
```

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
su -c "fbcp &"
exit 0
/etc/rc.local (END)
```

Nach einem Neustart sollte bereits eine Kopie auf dem TFT erscheinen. Ist das nicht der Fall, entfernen Sie am Anfang der Datei das `-e`, so dass nur noch `#!/bin/sh` da steht.

Allerdings wird die Auflösung derzeit noch vom angeschlossenen Monitor bestimmt. Das kann nachteilig sein, wenn andere Monitore angeschlossen werden, oder gar keiner dran sein soll.

Dafür stellen wir die Auflösung fest ein, indem wir die Datei `/boot/config.txt` editieren:

```
disable_overscan=1
...
framebuffer_width=1152
framebuffer_height=768
```

Natürlich lässt sich jede andere Auflösung einstellen. Anschließend muss nochmal eine Kalibrierung mit `xinput_calibrator` durchgeführt werden.

Vorsicht! Wenn Sie den Raspberry ohne angeschlossenen Monitor betreiben wollen, probieren Sie die Auflösung auch ohne aus. Mit angeschlossenem Monitor wird das angezeigte Bild auf dem TFT teilweise verzerrt dargestellt und führt schnell zu falschen Schlüssen.

Rechte Maustaste simulieren

Um mit dem Touchscreen eine rechte Maustaste zu simulieren, muss man eine Wartezeit definieren, wie lange der Stift auf dem TFT gehalten werden muss und dann als rechte Maustaste erkannt wird. Dazu editieren wir die Datei **99-calibration.conf**:

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

und fügen die folgenden Zeilen in die Sektion für den ADS7846:

```
Option "EmulateThirdButton" "1"
Option "EmulateThirdButtonTimeout" "1500"
Option "EmulateThirdButtonMoveThreshold" "30"
```

Anhang

Einige Links zu den Waveshare-Displays:

<http://www.ebay.de/itm/271834567052>

<http://www.ebay.de/itm/181866281775>

<http://www.ebay.de/itm/381154140502>

<http://www.ebay.de/itm/271741755678>

<http://www.amazon.de/dp/B00SGJIV12>

<http://www.amazon.de/dp/B00UV7EOT0>

<http://www.amazon.de/dp/B00SKOPWC4>