



A paradise for a computergeek

- [Home](#)
- [Dockstar Kernel](#)
- [Impressum](#)
- [Kontakt](#)
- [X4Daemon](#)

Digitaler Bilderrahmen von Pearl als Statusdisplay für Dockstar

April 11, 2011 / [Andreas](#) posted in [Dockstar](#), [Linux](#) / [75 Comments](#)

Jeder der eine Dockstar hat, kennt das Problem, es fehlt ein kleines Statusdisplay oder sowas zum erschwinglichen Preis. Nun gibt es im <10 € Bereich so genannte "Digitale Bilderrahmen" bzw. DPFs (Digital Photo Frame), die eigentlich optimal für so etwas sind. Ich nenne die Digitalen Bilderrahmen ab jetzt nur noch DPF, nicht wundern.

Es begann mit der Möglichkeit DPFs mit dem st2205 Chip zu "hacken" und sie als Display für [lcd4linux](#) zu nutzen. Problematisch: Es war schwer vorherzusagen, welche DPFs diesen Chip haben und welche nicht. Später entwickelte [Martin Strubl](#) von [Section5](#) eine Methode ([hier](#)) DPFs mit AppoTech's AX206 Chip als Display für lcd4linux zu nutzen, allerdings ist der "Hack" noch etwas im Anfangsstadium und nicht soweit fortgeschritten wie für st2205 Displays, dort gibt es bereits Firmwares usw. Das ist aber anscheinend auch für den AX206 in Zukunft geplant. Vorteil beim AX206 ist, dass wir dank einigen Usern ([hier](#) und [hier](#)) wissen, wo es genau solche DPFs mit 2.4" Display gibt. Aussehen tut das ganze dann so:



Klick für größere Ansicht

Es ist übrigens besser ablesbar als man hier erkennt, es ist doch nicht so einfach das richtig abzufotografieren. In der Realität ist das alles scharf und den schwarzen Text auf weißem Hintergrund kann man auch lesen.

Es gibt nun gefüllte 100 Puzzleteile im Netz, wie man das Teil vollständig zum Laufen bekommt. Ich werd das nun versuchen mal zusammengefasst zu schildern. Vorab schon einmal danke an alle die das möglich gemacht haben.

Voraussetzung: In meinem Fall eine Dockstar mit Debian Squeeze, prinzipiell sollte es aber auch ein PC mit beliebiger Distribution tun. (Die Paketnamen sind dann eben anders.)

Ich selbst nutze sudo, wer das nicht tut, muss sudo eben weglassen und die Befehle als root ausführen.

Warnung: Dabei kann man seinen digitalen Bilderrahmen durchaus unbrauchbar machen. Ich übernehme keine Verantwortung für gebrickte Digitale Bilderrahmen oder sonstige Schäden die durch diese Anleitung verursacht werden. Jeder muss selbst wissen was er tut.

1) Kompilieren von dpfhack und lcd4linux

Für den ersten Schritt können wir uns eigentlich an die Anleitung von [celemine1Gig](#) halten, die er [hier](#) gepostet hat. Allerdings hat er 1-2 kleine Fehler gemacht, die wir geschickt umschiffen:

Erstmal installieren wir alle Abhängigkeiten:

```
sudo apt-get install libtool automake autoconf zlibg-dev libssl-dev python-dev libc6 libusb-dev libibus-dev subversion libgd2-noxpm-dev li
```

Was fällt auf? Im Gegensatz zu celemine haben wir libgd2-noxpm dabei, damit hat man nachher auch das Image-Widget in lcd4linux, was ansonsten fehlen würde. (erwähnt wurde das [hier](#) von hackfin.)

Danach brauchen wir noch ein neueres sdcc für squeeze, was wir runterladen und ebenfalls installieren:

```
wget ftp://ftp.debian.org/debian/pool/main/s/sdcc/sdcc-libraries_2.9.0-5_all.deb
wget ftp://ftp.debian.org/debian/pool/main/s/sdcc/sdcc_2.9.0-5_armel.deb
dpkg -i sdcc-libraries_2.9.0-5_all.deb
dpkg -i sdcc_2.9.0-5_armel.deb
```

Danach laden wir dpfhack-0.1alpha.tgz, entpacken es und kompilieren es:

```
wget http://tech.section5.ch/files/dpfhack-0.1alpha.tgz
tar -xvzf dpfhack-0.1alpha.tgz
cd dpf/src
make
```

Dies dürfte nun erfolgreich kompiliert sein, achtet auf Fehlermeldungen. Nun kommt lcd4linux dran:

```
cd ..
wget http://tech.section5.ch/files/dpf-lcd4linux.tgz
tar -xvzf dpf-lcd4linux.tgz
```

Nicht wundern, er entpackt es ins gleiche Verzeichnis wie schon den dpfhack. An diesem Punkt müssen wir jetzt die Datei build-dpf-lcd4linux.sh etwas anpassen, und zwar anders als celemin das in seinem Beitrag tat. Wir ändern die Zeile

```
./configure --with-drivers=DPF
```

in

```
./configure --with-drivers='DPF' --with-plugins='all,!dbus,!mpris_dbus'
```

Wichtig sind die einfachen Anführungszeichen um DPF. Bei dem "plugin"-Teil lassen wir bewusst dbus Sachen weg, sonst motzt er ständig wegen dem fehlenden DBUS auf dem Debian der Dockstar. Wer DBUS installiert hat oder DBUS Plugins später nutzen will, sollte !dbus und !mpris_dbus aus der Zeile entfernen. Sollte es nicht gehen, versucht -with-drivers=ALL. Es geht aber definitiv so wie oben beschrieben, hab es selbst probiert.

Nun konfigurieren & kompilieren wir lcd4linux mit der AX206 Library:

```
./build-dpf-lcd4linux.sh ./src/dpflib/
```

(Wichtig sind die 2 Punkte vor /src/dpflib/, warum erklärt celemin [hier](#).)

Beim Download des Quellcodes aus dem Repository kommt anfangs eine Zertifikatsfrage:

```
- Das Zertifikat ist nicht von einer vertrauenswürdigen Instanz ausgestellt
Überprüfen Sie den Fingerabdruck, um das Zertifikat zu validieren!
Zertifikats-Informationen:
- Hostname: ssl.bulix.org
  usw.
```

Einfach mit "t" temporär akzeptieren. (Es wird übrigen Revision 1142 ausgecheckt. Ich hab es kurz mit der neusten Revision 1143 getestet, der Patch dürfte dort auch klappen – bringt aber keine Vorteile.)

Danach können wir es einfach mit

```
cd lcd4linux
sudo make install
```

bzw. mit

```
cd lcd4linux
sudo checkinstall
```

installieren. Ich selber empfehle checkinstall, weil man später das Paket leicht wieder löschen kann und man die Paketverwaltung nicht umgeht. Allerdings mag [checkinstall](#) die Versionsnummer nicht, welche man auf 0.11.0-SVN ändern sollte. (oder irgendwas beliebiges).

Man kann sich den gesamten Schritt 1 sparen, wenn man mir vertraut und mein fertiges Debian Paket [hier](#) (md5: 23dc6b5c4cc44390e9364970be97cf03) herunterlädt. Das muss man dann nur noch mit dpkg -i <name des pakets> installieren. Von den Abhängigkeiten muss man nur libgd2-noxpm installieren, alle anderen sind nur fürs kompilieren nötig. (Kann sein dass noch irgendwas dazu kommt, was ich eh schon drauf hatte.). Man kann so auch das Paket auf einer zweiten Dockstar kompilieren und dann das Paket von checkinstall übertragen. (so hab ich es gemacht.). ABER: Man braucht für Schritt 2 Teile des dpfhack und muss diesen demnach sowieso kompilieren und brauch auch die anderen Abhängigkeiten u.U. Es spart einem also nur Zeit, wenn der DPF bereits gehackt ist.

Als Backup hab ich die beiden Quelltextarchive [dpfhack-0.1alpha.tgz](#) (md5: 415398689d2ea77e83a79c513d44da82) und [dpf-lcd4linux.tgz](#) (md5: 3cccca2797ebb85459dc272a8b9e9b67) auch nochmal als Download hochgeladen.

2) "Hacken" des DPF

Wir haben nun schon einiges in Schritt 1 vorbereitet, jetzt müssen wir den DPF "hacken". Dazu gehen wir in den dpf/src Ordner. Der Befehl dazu, sofern ihr noch im lcd4linux Ordner seid, lautet:

```
cd ./src
```

Nun schalten wir den DPF an (etwas länger Menü drücken) und stecken ihn an, gehen ins Menü des DPF und aktivieren "mit PC verbinden" und schauen uns an, was dmesg nun sagt: (es sollte bei euch in etwa gleich aussehen)

```
[ 6803.710565] usb 1-1.3: new full speed USB device using orion-ehci and address 4
[ 6803.875039] usb 1-1.3: New USB device found, idVendor=1908, idProduct=0102
[ 6803.881970] usb 1-1.3: New USB device strings: Mfr=2, Product=3, SerialNumber=0
[ 6803.889318] usb 1-1.3: Product: Digital Photo Frame
[ 6803.894253] usb 1-1.3: Manufacturer: BUILDWIN
[ 6803.901500] usb 1-1.3: configuration #1 chosen from 1 choice
[ 6803.916037] scsi1 : SCSI emulation for USB Mass Storage devices
[ 6803.925274] usb-storage: device found at 4
[ 6803.925285] usb-storage: waiting for device to settle before scanning
[ 6808.921809] usb-storage: device scan complete
[ 6808.927740] scsi 1:0:0:0: CD-ROM                buildwin Photo Frame    1.01 PQ: 0 ANSI: 2
[ 6808.982804] sr0: scsi3-mmc drive: 40x/40x writer cd/rw xa/form2 cdda tray
[ 6808.989629] Uniform CD-ROM driver Revision: 3.20
[ 6808.996006] sr 1:0:0:0: Attached scsi CD-ROM sr0
[ 6809.028232] sd 0:0:0:0: Attached scsi generic sg0 type 0
[ 6809.036215] sr 1:0:0:0: Attached scsi generic sg1 type 5
```

Danach starten wir den Hack(komischerweise meldet sich mein DPF als sg0 und sg1 .. achtet bitte auf eure dmesg Ausgabe. Das Skript warnt euch aber, falls ihr eure Festplatte schrotten wollt, kann eigentlich nicht viel passieren, zumal solche Geräte sda,sdb usw. heißen):

```
sudo python hackit.py /dev/sg1
```

Das Skript erklärt am Schluss auch gleich, wie wir den DPF ab jetzt in den Developer Mode bekommen:

Zum Developer Mode: Press and hold MENU while USB is plugged in.If successful, you will get the 'USB connect' message and the device will appear as non-USB storage device

To put the device back into (almost) original state working as USB storage, press the RESET button.

Also machen wir das: Kurz ab und anstecken und danach Menü gedrückt halten (2-3 Sek.) bis USB verbunden dran steht. Danach sagt dmesg:

```
[ 7027.710565] usb 1-1.3: new full speed USB device using orion-ehci and address 5
[ 7027.875035] usb 1-1.3: New USB device found, idVendor=1908, idProduct=0102
[ 7027.881966] usb 1-1.3: New USB device strings: Mfr=2, Product=3, SerialNumber=1
[ 7027.889314] usb 1-1.3: Product: USB-Display
[ 7027.893551] usb 1-1.3: Manufacturer: hackfin
[ 7027.897843] usb 1-1.3: SerialNumber: 0000
[ 7027.904154] usb 1-1.3: configuration #1 chosen from 1 choice
```

Das sollte nun bei euch auch so aussehen.

Nun brauchen wir noch eine /etc/lcd4linux.conf. Wie man die genau erstellt und was man damit für Möglichkeiten hat, kann jeder selber ergoogeln. Zum Testen könnt ihr einfach folgendes nehmen.

```
Display dpf {
```

```

    Driver      'DPF'
    Port        'usb0'
    Font        '6x8'
    Foreground  'ffffff'
    Background  '000000'
    Basecolor   '000066'
}
Widget FIRST {
    class 'Text'
    expression 'www.Geekparadise.de'
    width 54
    align 'L'
    update 5000
    Background 'ffffff'
    Foreground '000000'
}
Display 'DPF'

Layout Dockstar{
    Row01.Col01 'FIRST'
}

Layout 'Dockstar'

```

Beispielsweise könnt ihr das mit Nano in die Datei `lcd4linux.conf` reineditieren.

```
sudo nano /etc/lcd4linux.conf
```

Falls ihr ein Bild anzeigen wollt, sieht das Widget dafür so aus (es gehen nur .png):

```

Widget IMAGE {
    class 'Image'
    file  '/pfad/zur/bla.png'
    update 10000
    reload 1
    visible 1
    inverted 0
}

```

Neugestartet wird `lcd4linux` mit

```
sudo killall lcd4linux && sleep 1 && sudo lcd4linux
```

Ebenso muss die Datei `lcd4linux` dem ausführendem Besitzer (hier `root`, anders ging es nicht) gehören und die Rechte 600 besitzen. (`sudo chown root:root /etc/lcd4linux.conf` und `sudo chmod 700 /etc/lcd4linux.conf`).

Will man eine andere Datei nutzen, so muss man `-f <dateiname>` anhängen.

Die Ausgabe von `lcd4linux` sieht man, wenn man es nicht als Daemon startet, also mit `-F`.

Einen Tipp hab ich noch: Sollte irgendwo ein Fehler in der Konfig sein oder irgendwas nicht evaluierbar sein, kommt nicht immer ein Fehler, sondern `lcd4linux` verursacht dann manchmal plötzlich eine sehr hohe Systemauslastung. Normal sollte diese < 2% sein.

Weitere Informationen gibt es [hier](#) und am Ende des Artikel, dort verlinke ich auch ein paar Beispielkonfigurationen einiger Dockstaruser, welche mir so über den Weg gelaufen sind während dem Suchen.

3) Udev Regel / rc.local

Schritt 3 ist eher der Bequemlichkeit geschuldet, weil es nervt, `lcd4linux` immer von Hand zu starten. Ausserdem bleibt es hängen, sobald man dem DPF unbedacht absteckt. Also schreiben wir eine Udev Regel, wie das genau geht findet sich [hier](#) und [hier](#).

Wir benötigen 2 Regeln, einmal fürs "Abstecken" und einmal fürs "Anstecken". Beginnen wir mit der "Ansteckregel", dafür 2 Ausgaben:

`dmesg`:

```

[6493023.221634] usb 1-1.3: USB disconnect, address 34
[6493100.511631] usb 1-1.3: new full speed USB device using orion-ehci and address 35
[6493100.676135] usb 1-1.3: New USB device found, idVendor=1908, idProduct=0102
[6493100.683461] usb 1-1.3: New USB device strings: Mfr=2, Product=3, SerialNumber=1
[6493100.691430] usb 1-1.3: Product: USB-Display
[6493100.696177] usb 1-1.3: Manufacturer: hackfin
[6493100.701038] usb 1-1.3: SerialNumber: 0000
[6493100.707923] usb 1-1.3: configuration #1 chosen from 1 choice

```

`lsusb -v`:

```

Bus 001 Device 035: ID 1908:0102
Device Descriptor:
  bLength                18
  bDescriptorType        1
  bcdUSB                  1.10
  bDeviceClass            255 Vendor Specific Class
  bDeviceSubClass         255 Vendor Specific Subclass
  bDeviceProtocol         255 Vendor Specific Protocol
  bMaxPacketSize0         8
  idVendor                 0x1908
  idProduct                0x0102
  bcdDevice               2.00
  iManufacturer           2 hackfin
  iProduct                 3 USB-Display
  iSerial                  1 0000
  bNumConfigurations      1

```

Nun erzeugen wir nun die Datei `/etc/udev/rules.d/70-dpf.rules` und fügen folgende Zeile hinein:

```
BUS=="usb", ACTION=="add",SYSFS{idProduct}=="0102", SYSFS{idVendor}=="1908", RUN+="/usr/local/bin/lcd4linux"
```

Damit wird automatisch `lcd4linux` gestartet wenn man den DPF im Developer Mode dran hängt. Die beiden ids sollten bei euch gleich sein, wenn ihr den Pearl DPF habt. Schaut aber zur Sicherheit nach!

Achja, damit die Regel geht, müsst ihr `udev` neustarten mit `sudo /etc/init.d/udev restart` oder `sudo /etc/init.d/udev reload`.

Für die "Absteckregel" müssen wir andere Tools zu Rate ziehen, da das obige beim "remove" nicht geht, bzw. die SYSFS Informationen schlicht nicht mehr verfügbar sind. Daher schauen wir uns an was `udev` so treibt mit folgendem Befehl:

```
sudo udevadm monitor --property
```

Wenn man den DPF absteckt, sieht das so aus (ich hab das gekürzt):

```

UDEV [1302474093.969410] remove /devices/platform/orion-ehci.0/usb1/1-1/1-1.3/1-1.3.1 (usb)
UDEV_LOG=3
ACTION=remove
[...]
ID_VENDOR=hackfin
ID_VENDOR_ENC=hackfin
ID_VENDOR_ID=1908
ID_MODEL=USB-Display
ID_MODEL_ENC=USB-Display
ID_MODEL_ID=0102
ID_REVISION=0200
ID_SERIAL=hackfin_USB-Display_0000
[...]

```

Daraus ergibt sich die zweite Regel welche wir ebenfalls für `/etc/udev/rules.d/70-dpf.rules` hinzufügen. Diese Datei sollte dann so aussehen (killall sollte installiert sein, keine Ahnung ob Debian das standardmäßig hat):

```

BUS=="usb", ACTION=="add",SYSFS{idProduct}=="0102", SYSFS{idVendor}=="1908", RUN+="/usr/local/bin/lcd4linux"
SUBSYSTEM=="usb", ACTION=="remove", ENV{ID_MODEL_ID}=="0102", ENV{ID_VENDOR_ID}=="1908", RUN+="/usr/bin/killall lcd4linux"

```

Wem udev zu kompliziert ist, hat wenigstens die Möglichkeit beim Starten der Dockstar das Display aktivieren zu lassen, in dem er die Zeile `"/usr/local/bin/lcd4linux"` zur `/etc/rc.local` hinzufügt. (Eine Zeile vor "exit 0".) Dann muss aber beim Booten der DPF bereits im Developer Modus an der Dockstar hängen. (oder später angesteckt und von hand lcd4linux gestartet werden.)

4) Wo gibts das Teil? Costa Quanta?

Es gibt mehrere Quellen und auch mehrere AX206 DPF, aktuell bevorzuge ich eigentlich [Pearl.de](#). Dort gibt es den DPF, den ihr oben im Bild seht, für [2,90 €](#) (HPM-1184-904) und für [9,90 €](#) (PX-1184-904). Die sind beide gleich, von ersterem darf allerdings pro Bestellung nur einer dazu bestellt werden. Gegen mehrere Bestellungen spricht natürlich nichts... ☹

Dazu kommen 4,90 € (LSV) bzw. 6,90 € (NN) Porto und 2,50 € Mindermengenzuschlag, falls ihr unter 17,50 € seid. Zusätzlich könnt ihr euch noch einen Aktionsartikel mit in den Warenkorb legen, ich empfehle den [8 GB Verbatim USB Stick für 2,90 €](#) (GRA-8294-935) zu nehmen oder die [Osram LED Taschenlampe für 0 €](#) (GRA-16629-904), die den Vorteil hat, dass der Mindermengenzuschlag entfällt. Warum weiß ich auch nicht, Bug im Shop denke ich, wird vermutlich bald gefixt. Für andere Gratisartikel einfach mal [Google](#) anschmeißen.

Edit: Ich muss sagen, die Taschenlampe lohnt sich echt. Unbedingt nehmen! Was ich schade finde: Man bezahlt 4,90 € Porto + Mindermengenzuschlag und Pearl schickt einem das Zeug mit einer [Warensendung](#) für 1,65 € (Sowohl der Preis, als auch "Warensendung" stehen auf dem Paketaufkleber).

Weitere, unbestätigte Quellen sind: (unbestätigt weil ich sie nicht selbst probiert hab...)

[Dealextrème 2.4" DPF](#) - [Dealextrème 2.4" DPF #2 als Herz](#) - [Dealextrème 1.5"](#) - [Dealextrème 1.5" #2](#) - [Ebay #1](#) - [Ebay #2](#)

Im Endeffekt ist Pearl am günstigsten für 10,30 € mit der LED Taschenlampe oder 13 € mit dem 8 GB USB Stick.

5) Links zum Thema:

- [Wikiartikel auf picframe.spriteserver.nl](#) (da er immer mal wieder Probleme hat, [hier](#) der Google Cache Link)
- [Blog von section5](#)
- [hkramski's buildzeile](#)
- [hkramski's Erklärung + Links](#)
- [hackfin's Erklärung wie man Image Widget Support bekommt](#)
- [celemine1Gig Tutorial](#)
- [weiteres Post von celemine1Gig](#)
- [weiteres Post von celemine1Gig #2](#)
- [LCD4LINUX WikiHauptseite Manual](#)
- [Bild](#) des Pearl DPF
- [Bild](#) des Pearl DPF nackt [#1](#) [#2](#) [#3](#) [#4](#)
- [Bild](#) des Pearl DPF ohne den durchsichtigen Rahmen
- [Bild #1](#) und die Beispielkonfig daraus [hier](#). Quelle [hier](#).
- [Bild #2,#3](#) und die Beispielkonfig daraus gibts [hier](#). Quelle [hier](#).
- [Beispielkonfig](#)

6) Ausblick

Man liest öfters dass es bald ebenfalls eine "custom"-FW für die AX206 DPFs geben soll und auch die Hintergrundbeleuchtung soll bald steuerbar sein via lcd4linux. Ich denke und hoffe dass sich da noch einiges tun wird, obwohl es, bis auf die fehlende Steuerung der Hintergrundbeleuchtung, echt gut klappt. Ausserdem hoffe ich, dass man auch bald direkt in den Developermodus kommen kann sobald man den DPF ansteckt – "Menütaste gedrückt halten" nervt doch etwas. Alternativ wäre es gut, wenn man die Hintergrundbeleuchtung einfach deaktivieren könnte.

Um die Stecker und Nerven zu schonen nutzt ich aktuell einen [USB Hub mit Ausschalter](#) von DealExtreme. Somit muss man nur noch Schalter umlegen & Menü drücken ☺

7) Nachtrag

- Der Standfuß geht mega schnell kaputt, beim 3. oder 4. mal ab-klipsen bricht er. Vorsicht also!
- Wer vor hat, das Display mit Akku zu betreiben .. lasst es. Alle meine Displays kamen mit praktisch schon total plattem Akku an. Da geht nicht mehr viel, selbst wenn es 2 Tage am USB hing, hält der Akku nur kurz.
- Wer den Akku (hauptsächlich aus mangelndem Vertrauen zu China-Akus) ausbauen will, findet [hier](#) meine Anleitung dazu.
- Wer lcd4linux mit -q startet, erspart sich die Werbung beim Starten/Beenden von lcd4linux auf dem Display. Geht natürlich auch mit Udev.
- Wer -q benutzt, wird (sofern er kein Image Widget als Hintergrund nutzt) einen Bug bemerken: lcd4linux ignoriert

```

Background !...!
Basecolor !...!

```

Und man hat auch nach dem Anstecken noch das "USB verbunden" mit dem weißen Hintergrund dastehen. Ansonsten wenn man lcd4linux neustartet bleibt es einfach blau als Hintergrund.

Lösung: -q weglassen oder Hintergrundbild (schwarz z.B.)

- falls jemand folenden Fehler erhielt:

```

calling SCSI ioctl(): Bad file descriptor
SCSI inquiry failed
Traceback (most recent call last):
  File "hackit.py", line 134, in
    d = dpf.open(sys.argv[1])
SystemError: Failed to open port:
File open error

```

Das war meine Schuld (mehr oder weniger) .. ich hab oben vergessen zu schreiben, das man natürlich im Menü des DPF die USB Verbindung aktivieren muss. I am sorry for that! -> korrigiert.

- Es gibt eine neuere Version des Dpffacks [hier](#). Vorsicht: Noch im experimentellen Stadium!

- [Bastel aus hwluxx-forum](#) hat etwas interessantes entdeckt:

Wenn man den DPF ansteckt und 5-10 Minuten wartet, geht er automatisch in den "Hackfin" Modus. Voraussetzung ist das aktivieren der Diashow, es klappt aber definitiv in den Standardeinstellungen (ohne AKKU also). Krasse Sache, Danke Bastel!

Gefällt mir 5 4 Twittern 14

< [AppoTech](#), [AX206](#), [Bilderrahmen](#), [DealExtreme](#), [Digital](#), [Digitaler](#), [DPF](#), [dpffack](#), [Frame](#), [lcd4linux](#), [Pearl](#), [Photo](#), [picframe](#), [ST2205](#) >
 « [Linux: Großer Geschwindigkeitsvergleich EncFS/eCryptfs/TrueCrypt/LuksNetio auf der Dockstar](#) »

Comments (75)



1.

18:49, October 27, 2011 [Andreas](#) / [Reply](#)

Du versuchst ein arm Paket auf i386 Hardware zu installieren. Du brauchst sdcc passend für Ubuntu i386, weiß gerade aber nicht aus dem Stehgreif ob es das in den Quellen dort gibt, denke aber.



2.

18:48, November 19, 2011 [Thomas](#) / [Reply](#)

Hi! Danke für die Anleitung. Mit deinem fertig gebauten Paket läuft das ganze hervorragend an einer NSLU2.

Ich verwurschtel das Ganze bei mir auch mal in einem Blogbeitrag, natürlich mit Referenz auf deinen Beitrag hier 😊



3.

08:34, December 7, 2011 [Thorsten](#) / [Reply](#)

Geht 😊

Hier beschreibe ich das für Ubuntu, natürlich mit Referenz auf diesen Beitrag hier: <http://www.staerk.de/thorsten/Lcd>



4.

13:45, December 13, 2011 [ThomasK](#) / [Reply](#)

Unter Gentoo muss das Makefile in src/python ein wenig anders aussehen.

Hier die relevante Stelle:

```
ifeq (Linux, $(PLATFORM))
PYTHON=$(shell python-config --libs)
include $(DEVLIB)/unixdll.mk
DLDEPS = $(CLIENTLIB) $(PYTHON)
PYTHONINCLUDES = $(shell python-config --includes)
else
```



5.

17:54, December 27, 2011 [alex](#) / [Reply](#)

hi leute, danke für die tolle anleitung.

ich habe versucht, das für ein 64bit system zum laufen zu bringen (mit eurer und der anleitung von http://linuxnetwork.li.funpic.de/v2_addon_ax206_src.html).

leider bekomme ich beim starten von lcd4linux folgenden fehler:

```
lcd4linux.c: initializing driver GLCD2USB
drv_init(GLCD2USB) failed: no such driver
Error initializing driver GLCD2USB: Exit!
```

kann mir jemand weiterhelfen?

grüsse, alex



6.

22:21, January 12, 2012 [SK](#) / [Reply](#)

I've bring the driver into lcd4linux svn [1168].

But only the driver, the lib needs to be in an include and lib search path.

The scaletext patch is fixed for svn and send to ML.



7.

00:35, January 14, 2012 [Jens](#) / [Reply](#)

Hallo Hackfin,

der Bootloader ist gehackt, siehe <http://tech.section5.ch/news?p=77>



8.

00:44, January 14, 2012 [Andreas](#) / [Reply](#)

Soweit ich mich erinnere, war tech.section5 doch hackfin's Blog 😊



9.

11:55, January 17, 2012 [Jens](#) / [Reply](#)

[@Andreas](#)

Jep. Das heißt also, es kommt irgendwann eine "Richtige" Firmware???



10.

17:33, January 17, 2012 [rmb](#) / [Reply](#)

hi, koenntest du vielleicht in deine lcd4linux.deb folgenden patch einfüegen:

<http://www.vdr-portal.de/board18-vdr-hardware/board11-lcds/109344-pearl-display-textskalierung-unter-lcd4linux/>

aus dem ersten post. ich krieg das irgendwie nicht hin

11. 

20:19, January 25, 2012rebdel / [Reply](#)

Hat zwar nix mit dem Pearl zu tun, aber ich poste es trotzdem.

Habe heute ein DION SD708 mal geöffnet, da waren JTAG/VESA/LAN und wie es aussieht USB für die Bestückung vorgesehen. Vielleicht lässt sich mit dem 7" was anstellen ...

12. 

18:36, February 4, 2012SK / [Reply](#)

Der scaletext Patch ist inzwischen auch im lcd4linux SVN.

13. 

16:42, February 10, 2012Andre / [Reply](#)

Hallo,

Tolle Idee das hier! Habe mir auch so einen Peral-Biderrahmen geholt und sogar schon gehackt. Ich bin also in diesem Hack-Screen (blau mit weißer Schrift). Das Flashen war extrem easy, habe es mit einem Tool unter Windows gemacht. So weit so gut.

Nun habe ich aber ein Problem, es unter Linux anzusteuern. Ständig fehlen mir Pakete und überhaupt scheine ich nicht das richtige OS zu haben: openSUSE und Debian wollen nicht. Der hier erwähnte "Dockstar" ist meinen Recherchen zufolge ein Gerät mit ARM-Architektur, komme ich deswegen nicht weiter? Ich nutze eine 'normale' Architektur.

Könnt ihr mir helfen, dass ich das Display ansteuern kann? Mir würde es schon genügen, wenn ich die Bilder selber zeichne und sie immer drauf kopiere. Ich habe dieses, sowie andere Tutorials befolgt, aber immer hakt es irgendwo:

Lcd4linux habe ich. Aber ich versteh nicht, was es mit dem Patchen auf sich hat; wird dadurch dem Lcd4linux der DPF-Driver hinzugefügt, bevor ich Lcd4linux mit "make" installiere? Wenn ja, wo kriege ich den her? Die Homepage von LCD4Linux sagt: "To build this driver you need libdpf. But it isn't online at this moment."

Hoffe, ihr könnt mir helfen, dieses Tutorial zu Ende durch zu bekommen oder mir einen anderen Weg aufzuzeigen 😊

14. 

17:15, February 15, 2012Kurt / [Reply](#)

Tolle Anleitung! Jetzt habe ich für meine 2 NSLU's (laufen mit Lenny) die DPF's hier , scheitere aber schon im ersten Schritt, weil das Paket "libibus-dev" nicht gefunden wird. Kann mir da bitte mal jemand einen Tipp geben?

E: Konnte Paket libibus-dev nicht finden 😞

Gruß
Kurt

15. 

15:46, February 19, 2012rebdel / [Reply](#)

@Kurt:

dann ändere doch die /etc/apt/sources.list auf squeeze ...

deb <http://ftp.de.debian.org/debian/> squeeze main

#deb <http://ftp.de.debian.org/debian/> lenny main

deb-src <http://ftp.de.debian.org/debian/> squeeze main

deb <http://security.debian.org/> squeeze/updates main

deb-src <http://security.debian.org/> squeeze/updates main

#deb <http://volatile.debian.org/squeeze-updates> squeeze/volatile main

#deb-src <http://volatile.debian.org/squeeze-updates> squeeze/volatile main

#deb <http://volatile.debian.org/debian-volatile> squeeze/volatile main

deb <http://ftp.de.debian.org/debian> squeeze-updates main

16. 

15:47, March 4, 2012Murry / [Reply](#)

fastoff patch (after 5sec switch to Hack mode):

<http://www.vdr-portal.de/board18-vdr-hardware/board11-lcds/111585-pearl-dpf-fastoff-firmware?s=e38c94ee0b647b058e4523bd784aa10b8712ba3>

17. 

18:43, March 5, 2012Kurt / [Reply](#)

@rebdel

Danke! Da hätte ich auch selbst drauf kommen können. Zumal es für Lenny ja keine Updates mehr gibt... Installation läuft. Mal sehen, wie es dann aussieht.

18. 

02:28, March 14, 2012hackfin / [Reply](#)

Moin Gemeinde,

neue 0.2devel Firmware ist endlich draussen, nix mehr mit Bricken und Wartezeiten/Menu drücken für den Bluescreen.

Einfach das Binary mit ProgSPI.exe flashen:

http://tech.section5.ch/files/fw_pearl_landscape.bin

Achtung: Funktioniert nur mit original Pearl-Frames. Bei manchen ist ev. das Display gespiegelt.

Viel Spass!

19. 

11:19, March 16, 2012 [Andreas](#) / [Reply](#)

Cool! Muss ich demnächst mal probieren ... ein Linuxflash tool hast du nicht zufällig auch noch ? 😊 (sonst muss eben die Windows VM wieder herhalten...)

Sofern ich die Zeit finde, kommt dann auch endlich mal ein aktualisierter Post mit einer aktuellen Anleitung.... 😊

20. 

21:30, May 3, 2012 [RonSon](#) / [Reply](#)

Moinsen!

Wer hat es schon hinbekommen das Ding als VNC "Monitor" bzw. als Terminal Monitor zu benutzen. Anscheinend soll das ja jetzt möglich sein nur ich blick da volle kanne NICHT durch!

21. 

22:38, June 6, 2012 [Androv](#) / [Reply](#)

Display habe ich geflasht...wenn ich es an usb anschließe geht das display an...glaube die hintergrundbeleuchtung..drücke ich die menütaste 3 sekunden, dann erkennt er es als dpf display.

Hier meine Ausgaben:

<http://andreasbos.no-ip.org/meinproblem.txt>

Wo liegt das Problem? ich verwende debian squeeze u. i386 architektur.

Wäre für jede Hilfe dankbar

22. 

11:36, June 10, 2012 [unbekannt](#) / [Reply](#)

Hallo zusammen,

ich hatte alles nach dieser Anleitung gemacht, jedoch bekomme ich bei dem Display keine png Grafik angezeigt, nur der Textmodus funktioniert...an was könnte es liegen?

23. 

16:11, October 11, 2012 [bimmi](#) / [Reply](#)

Hoffentlich laeuft das so auch auf meinem raspberry unter wheezy... da schon jemand erfahrung gemacht?

24. 

19:36, October 14, 2012 [bimmi](#) / [Reply](#)

Hi, ich hab folgendes Problem auf meinem Raspberry... Also LCD4Linux funktioniert soweit allerdings bekomme ich meinen Display nicht zum laufen... Es kommt immer die Fehlermeldung "bulk write: Resource temporarily unavailable".

Das Display ist auf jeden Fall schon gepatcht, da es an meiner Fritzbox schon guten Dienst erwiesen hat. Desweiteren hängt es an einem aktiven USB Hub...

Ich weiß nicht mehr weiter...

Hier ausführlicher:

```

sudo ./lcd4linux -Fvv -f lcd.conf
LCD4Linux 0.11.0-SVN-1168 starting
Dump of lcd.conf:
Display:DPF
Display:DPF.Background '000000'
Display:DPF.Basecolor '000099'
Display:DPF.Driver 'DPF'
Display:DPF.Font '6x8'
Display:DPF.Foreground 'ffffff'
Display:DPF.Port 'usb0'
Display:DPF.Size '128x128'
Layout 'Simple'
Layout:Simple.Row01.Col01 'HELLO'
Widget:HELLO.align 'L'
Widget:HELLO.Background 'ffffff'
Widget:HELLO.class 'Text'
Widget:HELLO.expression 'Hello World.'
Widget:HELLO.Foreground '000000'
Widget:HELLO.update 5000000
Widget:HELLO.width 12

lcd4linux.c: initializing driver DPF
Found AX206 DPF
Got LCD dimensions: 320x240
bulk write: Resource temporarily unavailable
initializing layout 'Simple'
layout.c: Layout:Simple: migrating 'row01.col01' to 'Layer:1.row01.col01'
Creating new timer group (0 ms)
widget 'HELLO': Class 'text', Parent "", Layer 1, Row 0, Col 0 (to 0,0)
lcd4linux.c: starting main loop
Removing timer group (0 ms)
Timer #0 skipped 2147483647 interval(s) or 0 ms.
Huh? Not even a single timer left? Dazed and confused...
lcd4linux.c: leaving main loop
DPF: shutting down.
bulk write: Resource temporarily unavailable

```

drv_dpfc: closing connection

Ich hoffe, jemand kann helfen...

25. 

15:31, October 18, 2012Roland / [Reply](#)

Hallo, leider komme ich nur bis hier:

Danach brauchen wir noch ein neueres sdcc für squeeze, was wir runterladen und ebenfalls installieren:

wget ftp://ftp.debian.org/debian/pool/main/s/sdcc/sdcc-libraries_2.9.0-5_all.deb wget ftp://ftp.debian.org/debian/pool/main/s/sdcc/sdcc_2.9.0-5_armel.deb dpkg -i sdcc-libraries_2.9.0-5_all.deb dpkg -i sdcc_2.9.0-5_armel.deb

Rückmeldung: No uch file 'sdcc_2.9.0-5_armel.deb'

gibt es bereits etwas neueres?

Roland

[« Previous 1 2](#)

Leave a Reply

Name (*)

Email (*)

Website

Allowed Tags - You may use these [HTML](#) tags and attributes in your comment.

 <abbr title=""> <acronym title=""> <blockquote cite=""> <cite> <code> <del datetime=""> <i> <q cite=""> <strike>

Post Comment (Ctrl+Enter)

Pingbacks (6)

- 02:34, April 14, 2011  [GeekParadise | Pearl Display als Dockstar-gesteuerter Bilderrahmen](#)
- 19:00, May 23, 2011  [LCD Display an DockStar | pkern.at](#)
- 21:57, January 1, 2012  [GeekParadise | Fakten & Frohes Neues Jahr & Danke](#)
- 15:44, January 3, 2012  [LCD an Fritz!Box mit lcd4linux und digitalen Bilderrahmen von Pearl | TomTom](#)
- 17:25, January 18, 2012  [Dockstar LCD-Display | Geek by Nature – Freak by Choice](#)
- 14:46, July 23, 2012  [Raspberry Pi am Pearl-Photoframe \(DPF\) unter Raspbian Wheezy « Wordpress-Blog von Guido Lehwalder](#)

Search:

Search...

Recent Posts

- [X4Daemon 0.4.4 veröffentlicht](#)
- [X4Daemon 0.4.3 veröffentlicht](#)
- [Neuer Dockstar Kernel für Debian \(3.0.42\)](#)
- [Neuer Dockstar Kernel für Debian \(3.0.40\)](#)
- [Neuer Dockstar Kernel für Debian \(3.0.38\)](#)

Archives

- [September 2012](#)
- [August 2012](#)
- [July 2012](#)
- [June 2012](#)
- [May 2012](#)
- [April 2012](#)
- [March 2012](#)
- [February 2012](#)
- [January 2012](#)
- [December 2011](#)
- [October 2011](#)
- [September 2011](#)
- [August 2011](#)
- [July 2011](#)
- [June 2011](#)
- [April 2011](#)
- [March 2011](#)
- [February 2011](#)
- [January 2011](#)
- [December 2010](#)
- [November 2010](#)
- [October 2010](#)
- [July 2010](#)
- [June 2010](#)
- [February 2010](#)
- [January 2010](#)
- [October 2009](#)
- [September 2009](#)
- [August 2009](#)
- [June 2009](#)

Links

- [Log in](#)
- [Entries RSS](#)
- [Comments RSS](#)
- [WordPress.org](#)

[Follow me on Twitter](#)

- No public Twitter messages.

Powered by [WordPress](#) / Theme [SimpleDark](#) by Justice