Es wurde das ganze via ATOM-Console gemacht und dabei einen USB2Serial-Adapter verwendet welcher gleich 1.8V kann (Modell: DSD TECH SH-U09C5 via amazon). Löten war nicht notwendig - es hat bei gereicht die 3 Pins (RX, TX und GND) ein wenig schief zu stecken.

Hier noch eine Info wie man den kompletten Datenbestand vorab sichert:

Benötigt wird dazu ein USB-Stick der mehr als 8GB Nutzdaten zulässt. Es wurde mit ext4 formatiert und an die FB6591 gesteckt.

Code:

```
### USBSTICK-Namen raussuchen (oder beim Einstecken aus dem Consolen-Output merken):
# mount | grep /dev/sda1
# mkdir -p /var/media/ftp/<USBSTICK-NAME>/dev
# find /dev -name mmcblk0* | while read line; do dd if="$line" of=/var/media/ftp/<USBSTICK-NAME>"$line"; done
# du -h /var/media/ftp/<USBSTICK-NAME>/dev/*
# sync
# sync
# umount /dev/sda1
```

Danach kann der Stick entfernt werden. Man hat jetzt einmal alle einzelnen Partitionen (ausser /dev/mmcblk0rpmb - was auch immer das ist) und mit "mmcblk0" nochmal den kompletten Inhalt des eMMC-Speichers.

Hier eine Beschreibung wie man per serieller Schnittstelle Zugriff auf die FritzBox-6591 bekommt.

- **Das ganze setzt voraus daß man etwas mit Hardware/Lötkolben/Linux umgehen kann, wer sich das nicht zutraut, Finger weg!**
- Es versteht sich dass man das nicht mit Provider-/Mietboxen machen sollte!
- **Disclaimer**: Keine Garantie für dass ich mich nicht irgendwo vertippt habe! Man sollte prinzipiell verstehen was man hier tut.
- Wer das hier liest, bitte auf das **Datum** des letzten updates achten. Ich werde das nicht bis in alle Ewigkeit maintainen.
- Die Prozedur funktioniert **unter Umständen NICHT für gebrandete Providerboxen** (firmware_version != avm).

Die 6591 hat zwei RS232 Konsolen (4-pin through-holes), eine für den Atom, eine für den ARM core. Im Prinzip können beide verwendet werden um sich einzuloggen bzw. eine firmware-Modifikation vorzunehmen. Aber:

- Über die Atom-Konsole kann man sowohl ein Update als auch Recovery durchführen, sollte man sich die Box "gebrickt" haben. Es ist also die zu bevorzugende Variante.
- Die Atom-Konsole wird mit 1.8V betrieben, d.h. man benötigt einen RS232 Adapter der das kann (siehe thread weiter unten), oder man muss für die üblichen USB Adapter noch einen Pegelwandler dazwischenschalten.
  Hier tut es ein ganz einfaches Platinchen mit Transistorschaltung, gibts bei Ebay, maker shops etc (ja, auch ein "3.3V auf 5V" Modell sollte gehen, zumindest tut es das bei mir).
- Die ARM Konsole läuft mit 3.3V. Hiermit kann man eigentlich "nur" eine modifizierte Firmware einspielen (bei mindestens einer FW-Version hat auch das nicht funktioniert, s.u.).

Prozedur

1. Gehäuse möglichst zerstörungsfrei öffnen. Die Position der Laschen erkennt man in etwa am Bild oben (je 3 links. rechts, unten).
2. ARM und/oder Atom Konsole anbringen (Pin header anlöten oder, wenn möglich, Verbindung temporär irgendwie anklemmen). Auf dem Bild oben:

    1. Atom Konsole: oben (am silbernen shield), Belegung von links nach rechts: 1.8V (eckig), Rx, Tx, GND.
        1. Bei einem passenden 1.8V Adapter nur Rx, Tx, GND anschliessen.
        2. Für einen Pegelwandler muss V/GND der "Low" Seite an 1.8V/GND des Steckers angebunden werden, entsprechend die "High" Seite an 3.3V/GND des RS232 Adapters.
           Am besten nochmal nachmessen.
    2. *(Optional) ARM Konsole in der Mitte, Belegung andersherum: GND, Tx, Rx, 3.3V (Eckig). Hier nur GND/Rx/Tx anschließen.*

3. Terminal-Programm öffnen, mit **115200/8/n/1 ohne flow control** verbinden.
4. Box einschalten und sich wie üblich nach ein paar Sekunden an der EVA Konsole per ftp anmelden:

    ftp 192.168.178.1

5. Folgende Kommandos:

    quote SETENV kernel_args mute=0
    quote REBOOT

    Die mute=0 Einstellung bleibt prinzipiell persistent.
6. Je nachdem welche Konsole man verbunden hat sollten nach einiger Zeit Ausgaben kommen (u.A. vom kernel), am Ende return drücken und man hat eine shell.
7. Das wird natürlich erst einmal nicht klappen, weil man (RS232-Gesetz!) Rx und Tx vertauscht

    hat, also umdrehen und noch einmal versuchen.

Hat man Zugriff auf die Shell kann man ein modifiziertes Atom rootfilesystem einspielen, z.B. mit Netzwerklogin. Ich haben meine Toolchain (https://bitbucket.org/fesc2000/ffritz/src/6591/) bereits auf die 6591 angepasst, so dass man sich ein solches modifiziertes Image generieren kann (wer nur die Konsole zum einloggen verwenden will kann natürlich auch das original Image nehmen).
Zum Bauen (momentan 7.12):

Code:
```
git clone --branch 6591 https://bitbucket.org/fesc2000/ffritz
cd ffritz
Make
```

Installation

Die Methode von der Atom-Konsole ist im Prinzip im README.md beschrieben. Für die ARM Konsole muss man die unten beschriebenen Kommandos mittels

rpc sh -c "kommandos"

ausführen. Ich empfehle das nicht, denn es funktioniert wohl nicht zuverlässig und wenn etwas schief geht benötigt man eh die Atom Konsole.

1. Das update-image (hier z.B. **release23/fb6591_7.12-23.tar**) in das NAS Verzeichnis auf der box kopieren.
2. Entpacken:

   cd /var/media/ftp; tar xf fb6591_7.12-23.tar

3. Update installieren (dauert ca. 20sek):

   /sbin/burnuimg /var/media/ftp/var/firmware-update.uimg || echo FAILED

4. Wenn erfolgreich, bootbank switch und reboot:

   /bin/aicmd pumaglued uimg switchandreboot

5. Nach dem reboot sollte fuer einige Zeit ein telnet daemon laufen. Einloggen, ssh login credentials vergeben (passwd oder pubkey nach /.ssh/authorized_keys), sichern (nvsync), fertig.
   Der telnet service wird nach ein paar Minuten terminiert.

Den bootbank-switch kann man auch wie üblich in der EVA ftp console mittels "**quote SETENV linux_fs_start *0|1***" durchführen (oder per EFI shell, siehe unten). Wenn eine bank gar nicht bootet wird auch ein automatischer switch durchgeführt.

Recovery

Sollte man sich beide Bootbänke so zerstört haben so dass kein Linux mehr bootet hilft nur noch die Atom-Konsole.

1. Die Datei **var/firmware-update.uimg** im release-tarball auf einen USB stick kopieren.
2. USB Stick an die FritzBox anschliessen.
3. Box einschalten. Um in die EFI-Shell zu gelangen muss man

   1. auf der Konsole "***exit***" eingeben während der EVA ftp server läuft (nach der Ausgabe von "**EvaHack ready**"),
   2. unmittelbar gefolgt von ein paar mal Escape-Taste.

4. "***map***" eingeben. Hier werden die device mappings aufgelistet. Der USB stick sollte etwa so erscheinen:

   FS2: Alias(s):HD36c0b:;BLK22:
   PciRoot(0x0)/Pci(0x14,0x0)/USB(0x2,0x0)/HD(1,MBR,0x0047BD56,0x40,0x7807C0)

5. Das image in den Speicher laden (FS2: ggf. ändern wie in Schritt 4 gelistet):

   load2mem -f FS2:\firmware-update.uimg

   .. und sich die angegebene Adresse kopieren.
6. Üblicherweise möchte man das aktuelle boot Image beibehalten und auf das backup Image schreiben. Dazu die bootbank umschalten:

   aid toggle
   aid update

7. Jetzt das Image programmieren (Adresse aus schritt 5):

   update -a A -s 0x513A010

8. Bei Erfolg ("Congrats! Looks like everything went as planned! Your flash has been updated! Have a good day!") rebooten:

   reset

Die Box sollte jetzt mit dem neuen Image starten.

Edit: Vergessen, **VIELEN Dank** an @Flole für die wertvolle Unterstützung!
Edit2: Pegelwander, RS232 Vcc sollte 3.3V sein.
Edit3: EFI shell: Atom statt ARM Konsole
Edit4: Recovery Sektion
Edit5: Hinweis zu Providerboxen
Edit6: Ein paar Schönheitskorrekturen.
Edit7: Installationskommandos für ARM Konsole entfernt.

Das Ganze ist nicht von mir. Ist nur zusammengestellt.