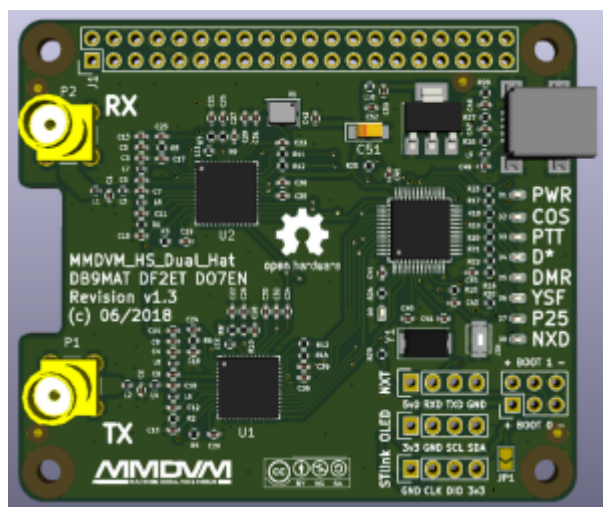


Update Firmware MMDVM_HS_Dual_Hat aus den Sourcen



Für alle, die einen Hotspot **MMDVM_HS_Dual_Hat** haben und selbst die Firmware auf den aktuellen Stand bringen wollen - und zwar aus den Quellen - hier mal eine kleine Quick-and-dirty Anleitung. Es setzt allerdings voraus, in der Lage zu sein

- a) sich per SSH auf den Pi einloggen zu können
- b) einigermaßen mit der Shell unter Linux umgehen zu können

Wer das sich nicht zutraut - lieber sich von jemanden helfen lassen, der das kann. Das ist also nix für "Einsteiger" und auch kein Grundkurs in Linux. Fundierte Kenntnisse im Umgang mit Linux muss ich hier zur Voraussetzung machen, um diese Anleitung korrekt umsetzen zu können.

Meine per Ebay gekaufte MMDVM_HS_Dual_Hat hatten oft einen Versionstand von nur 1.3.x, was mit dem aktuellen Pi-Star und dem dortigen MMDVMHost Probleme bereitet, also musste das alles auf aktuellen Stand gebracht werden.

Es wird für das derzeit aktuelle Pi-Star mindestens **Firmwareversion >= 1.4.8** des MMDVM-Modems empfohlen, da ab dieser Version verschiedene Probleme im Zusammenspiel Modem ↔ Gerät korrigiert bzw. minimiert werden.

Auf dem Web- bzw. Browserinterface des Pi-Star kann man das unter dem Punkt *Radio-Info* -> *FW* ablesen.

Radio Info	
Trx	TX DMR Slot 1
Tx	439.500000 MHz
Rx	431.900000 MHz
FW	HS_Hat:v1.4.11

Meine Anleitung bezieht sich auf die Version, die per GPIO auf den Pi gesteckt wird - nicht die Variante per USB, da ist das Vorgehen beim Update der Firmware etwas anders. Desweiteren kommt hier in dieser Anleitung **Pi-Star zum Einsatz - es bezieht sich also alles**

darauf.

Natürlich kann man das auch auf andere Linux-Distributionen des Raspberry Pi entsprechend adaptieren, könnte aber zusätzliche Installationen fehlender Pakete erforderlich machen.

→ WICHTIG für Experten und Linux-Kundige: Bitte derzeit unter Pi-Star kein `sudo apt-get upgrade` oder `sudo apt-get dist-upgrade` ausführen, danach läuft der Webserver vom Pi-Star nicht mehr !!!

Dann also mal los - **es bezieht sich alles auf die derzeit aktuelle [Pi-Star-Plattform](#).**

1. SSH auf den Pi-Star - also dort einloggen

```
pi-star@dmr-hs-dllbz(ro):~$
```

2. Ganze SD-Card als Filesystem nutzen und Neustart:

```
pi-star@dmr-hs-dllbz(ro):~$ sudo pistar-expand
```

```
pi-star@dmr-hs-dllbz(ro):~$ sudo reboot
```

3. wieder SSH auf den Pi-Star nach Reboot, SD-Card zum Schreiben öffnen

```
pi-star@dmr-hs-dllbz(ro):~$ rpi-rw
```

```
pi-star@dmr-hs-dllbz(rw):~$
```

4. Voraussetzungen installieren

```
pi-star@dmr-hs-dllbz(rw):~$ sudo apt-get update
```

```
pi-star@dmr-hs-dllbz(rw):~$ sudo apt-get install gcc-arm-none-eabi gdb-arm-  
none-eabi libstdc++-arm-none-eabi-newlib libnewlib-arm-none-eabi
```

5. Firmware-Quellen runterladen

```
pi-star@dmr-hs-dllbz(rw):~$ cd ~
```

```
pi-star@dmr-hs-dllbz(rw):~$ git clone
```

```
https://github.com/juribeparada/MMDVM\_HS
```

```
pi-star@dmr-hs-dllbz(rw):~$ cd MMDVM_HS/
```

```
pi-star@dmr-hs-dllbz(rw):~$ git submodule init
```

```
pi-star@dmr-hs-dllbz(rw):~$ git submodule update
```

6. Laufende MMDVM-Prozesse anhalten

```
pi-star@dmr-hs-dllbz(rw):~$ sudo pistar-watchdog.service stop && sudo  
systemctl stop mmdvmhost.timer && sudo systemctl stop mmdvmhost.service
```

7. Anpassung der Sourcen an den **MMDVM_HS_Dual_Hat**:

```
pi-star@dmr-hs-dllbz(rw):~$ nano Config.h
```

8. Folgende Optionen für den **MMDVM_HS_Dual_Hat** in der **Config.h** anpassen - diese *müssen* gesetzt sein.

Man muss aber **wissen**, welcher TXCO (12 oder 14 MHz) auf dem Modem zum Einsatz kommt - also den korrekten TXCO entsprechend auswählen, sonst arbeitet das Modem dann nicht. Wer sich nicht sicher ist - einfach ausprobieren und eben nochmals flashen mit dem anderen TXCO. Bei meinen Modems war das bisher immer die 14MHz-Version wie auch weiter unten in der **Config.h** gesetzt (also die Option `#define ADF7021_14_7456` definiert die 14MHz-Version des TXCO, `#define ADF7021_12_2880` definiert die 12MHz-Version. Bitte nur **eine** der beiden setzen).

```
#define MMDVM_HS_DUAL_HAT_REV10
```

```
#define ENABLE_ADF7021
#define DUPLEX
// TCX0 of the ADF7021
// For 14.7456 MHz:
#define ADF7021_14_7456
// For 12.2880 MHz:
// #define ADF7021_12_2880
#define STM32_USART1_HOST
#define ENABLE_SCAN_MODE
```

weiterhin (optional) in der **Config.h** - diese *können* gesetzt werden:

```
// Send RSSI value:
#define SEND_RSSI_DATA
// Enable Nextion LCD serial port repeater on USART2 (ZUMspot Libre Kit and
ZUMspot RPi):
#define SERIAL_REPEATER
// Enable modem debug messages
#define ENABLE_DEBUG
```

Wichtig ab Firmware-Version 1.4.12:

Ab der Version 1.4.12 wird der Frequenzbereich 435-438 MHz per default gesperrt, das gilt auch für die Simplex-Version. Diese Sperre lässt sich in der **Config.h** aber wieder aufheben (war so generell bis 1.4.11):

```
// Disable frequency bands check
#define DISABLE_FREQ_CHECK

// Disable frequency restrictions (satellite, ISS, etc)
#define DISABLE_FREQ_BAN
```

und die geänderte **Config.h** speichern.

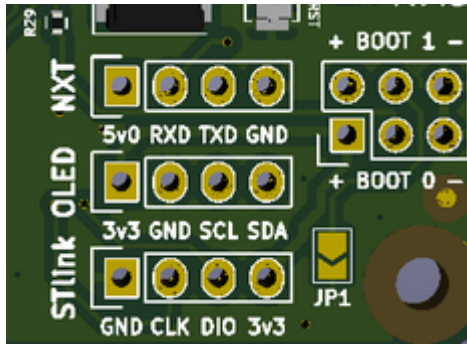
9. Compilieren:

```
pi-star@dmr-hs-dl1bz(rw):~$ make
```

10. Upload der Firmware ins Modem:

```
pi-star@dmr-hs-dl1bz(rw):~$ sudo make mmdvm_hs_dual_hat
```

→ Zum Flashen der **MMDVM_HS_Dual_Hat (GPIO)** muss bei Hardwarerevision 1.2 bzw. 1.3 des MMDVM_HS_Dual_Hat-Modems der **Jumper JP1 (rechts unten auf der Platine, meistens leider nicht bestückt)** gebrückt werden. Ansonsten geht das Flashen der Firmware NICHT wie im Schritt 10 beschrieben.



→ **Wenn nicht Pi-Star:** Es ist zu prüfen, ob das Flashtool **stm32flash** entweder unter **/usr/local/bin** oder unter **/usr/bin** vorhanden ist. Fehlt das Flashtool **stm32flash** an den entsprechenden Stellen - ist also nicht installiert - kann das Modem nicht geflasht werden, also Pkt.10:

```
pi-star@dmr-hs-d11bz(rw):~$ sudo make mmdvm_hs_dual_hat
```

wäre nicht durchführbar oder schlägt fehl.

→ **Bei Pi-Star ist das *stm32flash* bereits vorhanden, da gibt es kein Problem.**

11. am besten reboot:

```
pi-star@dmr-hs-d11bz(rw):~$ sudo reboot
```

Wenn alles geklappt hat, hat das MMDVM_HS_Dual_Hat jetzt aktuelle Firmware (V1.4.12, Stand 5.11.2018).

Das eben beschriebene gilt auch in etwas geänderter Form für die **MMDVM_HS_Hat** - also der Simplex-Variante.

Hier nur die geänderten Schritte für den MMDVM_HS_Hat - der Rest identisch wie eben beschrieben:



8. Folgende Optionen für den **MMDVM_HS_Hat** in der **Config.h** anpassen - diese *müssen* gesetzt sein::

```
#define MMDVM_HS_HAT_REV12
#define ENABLE_ADF7021
```

```
#define ADF7021_14_7456
#define STM32_USART1_HOST
#define ENABLE_SCAN_MODE
```

9. Compilieren:

```
pi-star@dmr-hs-dl1bz(rw):~$ make
```

10. Upload der Firmware ins Modem:

```
pi-star@dmr-hs-dl1bz(rw):~$ sudo make mmdvm_hs_hat
```

Hinweis: Hier gibts wohl keinen Jumper JP1 wie beim MMDVM_HS_Dual_Hat, sollte also einfach so funktionieren mit dem Upload der Firmware.

11. am besten reboot:

```
pi-star@dmr-hs-dl1bz(rw):~$ sudo reboot
```

Zum Abgleich der MMDVM_HS_Hat bzw. MMDVM_HS_Dual_Hat

Dazu wieder auf dem Pi-Star per SSH einloggen, dann das Tool MMDVMCal benutzen:

```
pi-star@dmr-hs-dl1bz(rw):~$ sudo pistar-mmdvmcal
```

1. In dem Tool mit **E** die QRG definieren, z.B. 439950000

2. mit **D** den DMR Deviation Mode aktivieren

3. **SPACE/LEER-Taste** für Senden drücken, Hotspot sendet jetzt, nochmaliges Drücken **SPACE/LEER** beendet die Aussendung

optional: 4. Taste **M** sendet ein Testpattern CC1 TS1 TG9 mit 1031 Hz Ton - sollte man also auf dem DMR-Gerät hören können

Mit einem z.B. SDR (bei mir SDRPlay RSP2Pro) die Frequenzlage prüfen, auch kann man mit **t** bzw. **T** das TXLevel erhöhen bzw. absenken. Wenn der Träger in der Mitte verschwindet, ist das Bessel-0. Den Wert merken und davon 5% abziehen - bei mir war das immer so 53% - die Vorgabe von 50 wäre dann also in Ordnung.

Fast alle Hotspot-MMDVMs hatten bei mir eine um ca. 500Hz zu hohe Frequenz. Kann man dann mit RXOffset=-500 bzw. TXOffset=-500 in der MMDVM.ini entsprechend korrigieren.

Prüfen kann man das im MMDVMCal durch Auswahl von **C**, was ein Carrier-Signal auf der per **E/e** gesetzten Frequenz erzeugt. Am besten mit Analyser oder SDR nachprüfen und mit TXOffset=+/- <Korrekturwert in Hz> entsprechend anpassen.

Danach sollte der Hotspot eigentlich funktionieren. Im Dashboard des Pi-Star die Aussendung zum Hotspot nochmals prüfen, die BER sollte schon unter 1% sein, was alle meine Hotspots bisher mühelos erreicht haben.

Die Taste **q/Q** verlässt bzw. beendet das Kalibrationstool wieder.

Update 25.2.2019:

Inzwischen ist das aktuelle [MMDVMCal](#) in der Lage, auch RX-seitig die BER zu messen und anzuzeigen. Ich habe also nun etwas die Strategie des Abgleichs geändert und auf minimales BER des Modem-RX im Mode DMR abgeglichen. Den dadurch bestimmten RXOffset habe ich gleich dem

TXOffset gesetzt, was mir speziell bei DMR ein nochmals verbessertes Sync-Verhalten des Modems bescherte. Leider - das ist eine nicht so gute Erkenntnis - ist dieser Offset bei C4FM nicht gleich und lieferte dort eine hohe BER. So einen richtigen Kompromiss für den Einsatz als Multimode-Hotspot habe ich bei DMR und C4FM leider nicht finden können, als Ergebnis setze ich jetzt 2 Modems ein, wobei eins DMR macht und das andere C4FM, beide auf den jeweiligen Mode hinsichtlich BER optimiert.

Noch einen abschließenden Hinweis zur (möglichen) Verbesserung der Funktion des MMDVM-Hotspot-Modems:

In der **Config.h** gibt es noch folgende Sektion:

```
// Configure receiver gain for ADF7021
// AGC automatic, default settings:
#define AD7021_GAIN_AUTO
// AGC automatic with high LNA linearity:
// #define AD7021_GAIN_AUTO_LIN
// AGC OFF, lowest gain:
// #define AD7021_GAIN_LOW
// AGC OFF, highest gain:
// #define AD7021_GAIN_HIGH
```

Ich habe auf allen MMDVM-Modems allerdings die letzte Option

```
// AGC OFF, highest gain:
#define AD7021_GAIN_HIGH
```

verwendet, da ich bisher nur DMR und C4FM mit den Hotspots gemacht habe. Beide nutzen ja 4-FSK als Modulation und ich hatte mal gelesen, das bei 4-FSK die AGC keine Wirkung zeigt. Also hab die kurzerhand ausgeschalten - bisher ohne erkennbare Probleme.

Update 29.10.2018:

In Ergänzung zu meiner Anleitung zum Selbstcompilieren der Firmware der MMDVM_HS_Dual_Hat bzw. MMDVM_HS_Hat kann man noch folgende Änderungen bzgl. des ADF7021 (also dem TRX-IC des Hotspots) in der **ADF7021.h** vornehmen:

Der Entwickler schreibt dazu:

```
#define ADF7021_ENABLE_4FSK_AFC
```

enable AFC support for DMR, YSF and P25. This is experimental, depending on your frequency offset this option will improve or not your BER reception.

```
#define ADF7021_AFC_POS
```

enable this option if you can not receive any signal after enable the ADF7021_ENABLE_4FSK_AFC

option.

Das schaltet die AFC auch bei 4FSK (DMR,C4FM) ein - ich schreib ja bereits das die eigentliche AFC-Funktion in der **Config.h** bei 4FSK keinen Effekt hat. Erst damit wird die dann auch bei 4FSK aktiviert . Das ist aber derzeit ein Experimental-Feature(!), es *könnte* Verbesserungen beim RX des Hotspots bringen oder auch *nicht*.

```
#define ADF7021_DISABLE_RC_4FSK
```

disable TX Raised Cosine filter for 4FSK modulation in ADF7021. Default TX pulse shaping filter for 4FSK is not optimum for DMR, YSF and P25. Activating this option might improve audio in 4FSK digital modes.

Bedeutet: "Der Standard-TX-Impulsformungsfilter für 4FSK ist nicht optimal für DMR, YSF und P25. Wenn Sie diese Option aktivieren, kann das Audiosignal in digitalen 4FSK-Modi verbessert werden." Sprich, es könnte besser in Eurem DMR-Gerät "klingen", also der Audio könnte sich verbessern durch ein besseres korrekteres Modulationssignal bei 4FSK.

Update 25.2.2019:

Aktuell rate ich davon ab, die AFC-Funktion zu benutzen und abgeschalten zu lassen:

```
// #define ADF7021_ENABLE_4FSK_AFC
// #define ADF7021_AFC_POS
```

Das brachte eher mehr Probleme mit sich als Verbesserungen, der Sync des Modem verschlechterte sich damit erheblich. Den "TX Raised Cosine filter for 4FSK modulation" schalte ich aber ab:

```
#define ADF7021_DISABLE_RC_4FSK
```

Damit verbesserte sich das Sync-Verhalten des Modems speziell bei DMR etwas ins Positive.

01.11. 17:48	DMR Slot 2	DL1BZ	262997	RF	6.1	0%	0.2%
----------------	------------	-------	--------	----	-----	----	------

So sollte das in etwa aussehen - wir haben hier eine BER (= **Bit Error Rate**, dt. Fehlerrate) am RX des Hotspots von 0,2% erreicht, das ist schon sehr gut.

Wie gesagt, das ist eher was für schon etwas erfahrene Linux-User.

73 Heiko, DL1BZ

From:

<http://kb.amft-it.de/> - **Amateurfunk - Knowledge Base und Wiki by DL1BZ**

Permanent link:

<http://kb.amft-it.de/doku.php?id=kb-afu:mmdvm-hospot&rev=1551099734>

Last update: **25.02.2019 14:02**



