

Update Firmware MMDVM_HS_Dual_Hat aus den Sourcen per USB (ab Revision 1.2 oder höher)

Vorwort

Für den Wechsel vom GPIO- zum USB-Betrieb des Modems ist es etwas schwieriger, fertige Firmwarepakete bereitzustellen. Die ganze Prozedur teilt sich eigentlich in zwei Teile, Flashen des Bootloaders im ersten Schritt und Flashen der eigentlichen Firmware des Modems im zweiten Schritt. Beim GPIO-Betrieb wird der Bootloader nicht benötigt, deswegen ist dort die Vorgehensweise etwas einfacher. Aus diesem Grund arbeiten wir hier mit dem Quellcode der Modemfirmware und nicht mit fertig compilierten Paketen, diese erzeugen wir hier uns in diesem Fall selbst. Der Quellcode ist Open Source und jedem auf Github frei zugänglich. Insofern möchte ich mich bei Andy CA6JAU, Florian DF2ET, Mathis DB9MAT und Andreas, DO7EN und alle Anderen wie Jonathan G4KLX, die an der MMDVM-Plattform mitwirken, für die Bereitstellung aller Teile als Open Source bedanken. Das ist ganz im Sinne des Amateurfunk-Gedankens.

Meine Anleitung bezieht sich auf die [MMDVM_HS_Dual_Hat](#)-Modem-Version ab Revision 1.2, die auch per USB an den Pi, PC oder Mac gesteckt werden kann. Somit kann man den MMDVM_HS_Dual_Hat auch an anderen Betriebssystemen einsetzen und ist nicht nur auf Linux bzw. den Raspberry Pi beschränkt. Die hier beschriebene Anleitung bezieht sich allerdings auf den Flashvorgang mittels eines Raspberry Pi3B+, da zum Transfer des Bootloaders - erforderlich für die korrekte Funktion des Modems an der USB-Schnittstelle - das Modem einmalig auf den GPIO-Connector des Pi gesteckt werden muss. Nach diesem Schritt können weitere Firmware-Updates dann per USB in das Modem eingespielt werden. Das alles ist wie immer nicht gerade für den Einsteiger gedacht, Linux-Erfahrungen und Basiskenntnisse im Umgang mit dem Linux-System sind auch hier zwingende Voraussetzung.

→Das Modem MMDVM_HS_Dual_Hat kann nur **ENTWEDER** als GPIO **ODER** per USB betrieben werden, beides gleichzeitig ist mit der Firmware **NICHT MÖGLICH** !

→Das Simplex-Modem [MMDVM_HS_HAT](#) besitzt keine USB-Schnittstelle und kann nicht per USB verwendet werden !

→Ältere, bei Ebay meist aus China angebotene MMDVM_HS_Dual_Hat-Modems in der Hardwarerevision 1.0, besitzen ebenfalls keine USB-Schnittstelle.

Aktuelle MMDVM_HS_Dual_Hat-Modems mit USB können in DL u.a. bei [Funk24.net](#) erworben werden (empfohlen).

Dann also mal los - **es bezieht sich alles auf die derzeit aktuelle [Pi-Star-Plattform](#) (3.4.17, 4.0.0RC3 geht auch) mit einem Raspberry Pi3B+ - kann aber auch bestätigen, dass das mit Raspian ebenfalls funktioniert.**

Schritt 1 (einmalig erforderlich) Flashen des Bootloaders und der Firmware mittels GPIO

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

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

2. **NUR Pi-Star:** Ganze SD-Card als Filesystem nutzen und Neustart (falls noch nicht erfolgt, wir brauchen etwas mehr Platz auf der SD-Card, >= 16GB empfohlen):

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

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

3. **NUR Pi-Star:** 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 (falls früher bereits erfolgt, kann das übersprungen werden)

```
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. Modem-Firmware (als Source-Code) MMDVM_HS aus Original-Quellen Github runterladen (derzeit 1.4.17)

```
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 && git submodule update
```

6. Laufende MMDVM-Prozesse anhalten (**hier Pi-Star**, andere Systeme wie Raspian entsprechend deren Konfig die MMDVMHost-Prozesse stoppen):

```
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_USB_HOST
// #define SERIAL_REPEATER
#define ENABLE_SCAN_MODE
// Enable for RPi 3B+, USB mode
#define LONG_USB_RESET
```

→ Bei einem Raspberry Pi2 bis Pi3B (nicht 3B+ !) bitte folgendes in der Config.h anpassen/ändern, da nur der Pi3B+ aufgrund geänderter USB-Funktionalitäten einen anderen (neuen) Bootloader benötigt, hier ist dann im Folgenden Schritt 10b anstatt 10a anzuwenden:

```
// Enable for RPi 3B+, USB mode
// #define LONG_USB_RESET
```

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

```
// Send RSSI value:
#define SEND_RSSI_DATA
// Enable Nextion LCD serial port repeater on USART1 (Do not use with
STM32_USART1_HOST enabled - only with USB host communication)
// Hinweis: Bei Einsatz eines weiteren Modems per GPIO lieber weglassen -
könnte Konflikte mit den seriellen Ports verursachen, hängt aber vom
Anwendungsfall ab
// da ich selbst keine Displays verwende lasse ich das grundsätzlich immer
weg
#define SERIAL_REPEATER_USART1
// 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. 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_BAND
```

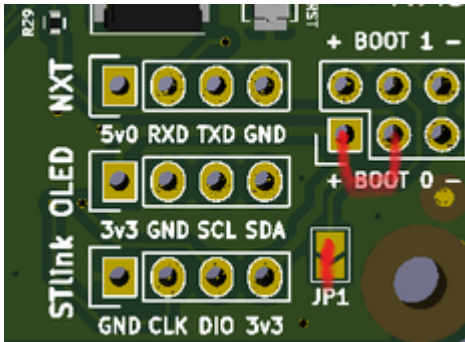
und die geänderte **Config.h** speichern.

9. Compilieren:

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

→ Zum Flashen der **MMDVM_HS_Dual_Hat (GPIO)** muss bei Hardwarerevision 1.2 oder höher des MMDVM_HS_Dual_Hat-Modems der **Jumper JP1 (rechts unten auf der Platine)** gebrückt werden, meistens ist das schon im Original so. **Desweiteren muss der Jumper BOOT 0+ gesetzt werden und vor dem Ausführen des Flashens im folgenden Schritt 10a bzw. 10b der kleine RESET-Taster auf dem Modem gedrückt werden!** Ansonsten geht das Flashen der Firmware NICHT wie im nächsten

Schritt beschrieben.



10a. Upload der Firmware ins Modem bei einem Raspberry Pi3B+ (neuer Bootloader mit #LONG_USB_RESET in der **Config.h**):

```
pi-star@dmr-hs-dl1bz(rw):~$ sudo make serial-bl devser=/dev/ttyAMA0
```

10b. ALTERNATIV: Upload der Firmware ins Modem bei **anderen Raspberry als den Pi3B+** (alter, bisheriger Bootloader beim Pi2 bis Pi3B *OHNE* #LONG_USB_RESET in der **Config.h**):

```
pi-star@dmr-hs-dl1bz(rw):~$ sudo make serial-bl-old devser=/dev/ttyAMA0
```

11. Jumper BOOT0 + wieder entfernen, JP1 kann aber weiterhin gebrückt bleiben !

12. Pi ausschalten, Modem vom GPIO abnehmen und mittels USB-Kabel an den Pi anstecken. Beim Pi-Star in der Modemauswahl dann MMDVM_HS_Dual_Hat for Pi (USB) auswählen bzw. in der mmdvmhost.ini die Modemschnittstelle von /dev/ttyAMA0 (bei GPIO) auf /dev/ttyACM0 (USB) ändern.

Schritt 2 Flashen/Update der Firmware MMDVM_HS_Dual_Hat mit USB

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

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

2. SD-Card zum Schreiben öffnen (nur bei Pi-Star)

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

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

3. Firmware compilieren

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

```
pi-star@dmr-hs-dl1bz(rw):~$ cd MMDVM_HS/
```

OPTIONAL: Sourcen der Firmware auf aktuellen Stand aus Github

https://github.com/juribeparada/MMDVM_HS bringen:

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

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

4. Flashen per USB (bei USB muss kein BOOT0+ gesetzt werden)

```
pi-star@dmr-hs-dl1bz(rw):~$ sudo make dfu devser=/dev/ttyACM0
```

→ Sollte das Flashen fehlschlagen ("Failed to init device") bitte kurz den RESET-Taster auf dem Modem drücken und nach ca. 2 sec oberen Befehl wiederholen. Alternativ kann auch das Modem kurz

vom USB abgezogen werden und wieder angesteckt werden, nach dem Blinken der LED auch wieder oberen Befehl wiederholen. Hilft das alles nicht bleibt nur die Option nach Schritt 1 vorzugehen und das Modem muss dann erneut per GPIO geupdatet werden, was natürlich alternativ auch immer geht.

→ Hat man z.B. zwei Modems im Einsatz, bitte die entsprechenden Schnittstellen verwenden. Ein zweites Modem nutzt dann anstatt /dev/ttyACM0 entsprechend /dev/ttyACM1 (bei zwei Modems per USB). Bei Pi-Star ist diese Konstellation allerdings nicht vorgesehen, mit Raspian als Alternative kann man aber mehr als eine Instanz MMDVMHost starten, natürlich mit verschiedenen mmdvmhost.ini-Dateien jeweils angepasst auf das anzusprechende Modem.

→Hinweis: Es gibt USB-Schnittstellen oder USB-Controller (nicht der Pi !) wo es bei der USB-Flash-Prozedur Probleme geben kann. Bei mir war das an einem PC so, wo ich mittels Linux (Debian) die Modem-Firmware updaten wollte. Hier muss man dann ggf. auf einen Pi ausweichen und den zum Updaten verwenden. Bitte achtet auch beim Einsatz per USB darauf, ein vollwertiges USB-Kabel zu verwenden, manchmal existieren USB-Kabel, die nur zum Laden gedacht sind und demzufolge nur GND und VCC (+5V) verschalten haben, aber keine Datenleitung DATA+ bzw. DATA- besitzen. Damit kann das Modem per USB natürlich dann nicht funktionieren.

Ob das Modem richtig erkannt wird, kann man so überprüfen:

```
pi-star@dmr-hs-dl1bz(ro):~$ sudo dmesg
```

und sollte in etwa folgende Ausgabe liefern (ggf. etwas hoch blättern):

```
[33949.436034] usb 1-1.2: new full-speed USB device number 13 using dwc_otg
[33949.569124] usb 1-1.2: New USB device found, idVendor=leaf,
idProduct=0004
[33949.569137] usb 1-1.2: New USB device strings: Mfr=1, Product=2,
SerialNumber=0
[33949.569146] usb 1-1.2: Product: Maple
[33949.569154] usb 1-1.2: Manufacturer: LeafLabs
[33949.572320] cdc_acm 1-1.2:1.0: ttyACM0: USB ACM device
```

vy 73 de Heiko, DL1BZ

Sysop DB0PIB/DB0OLL/DB0NLS

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-usb&rev=1571650824>

Last update: **21.10.2019 11:40**

