

[phl0 / MMDVM_HS_Dual_Hat](#) Public

Duplex variant of MMDVM_HS_Hat

☆ 152 stars 42 forks Branches Tags ActivityStarNotificationsCode Issues 7 Pull requests Actions Projects Security Insights

1 Branch 4 Tags Go to file Go to file Code ...

 **mathisschmieder** Added revision 2.0 mini · e76e7d7 · 4 years ago

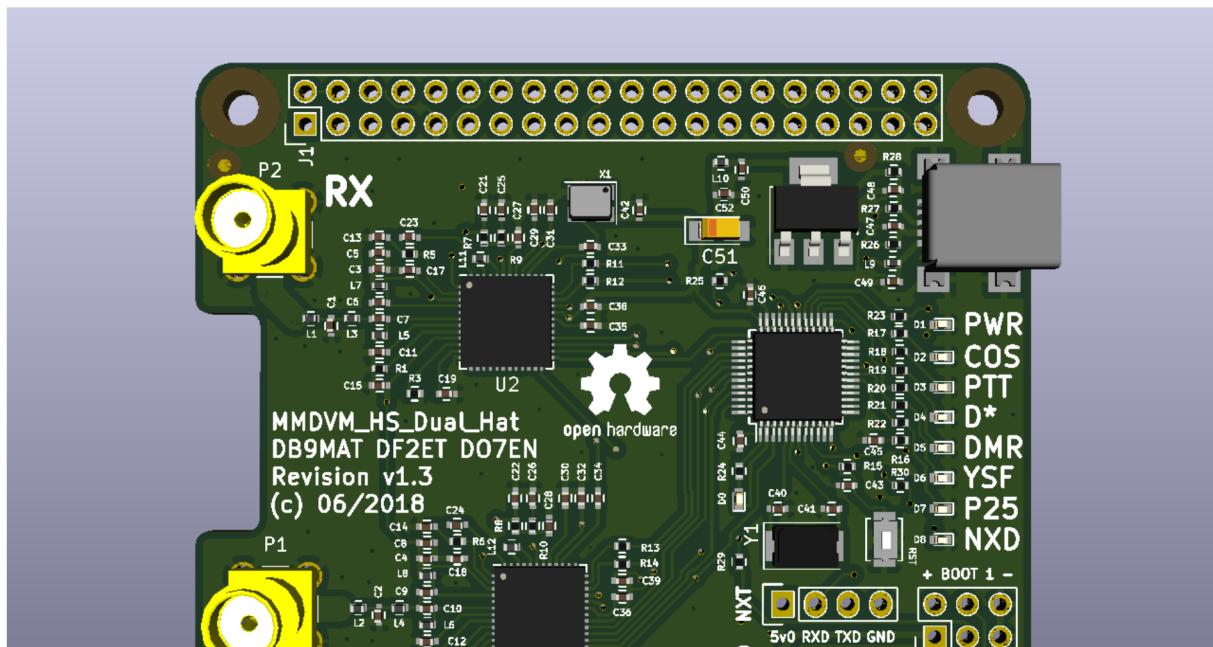
case	Add sources for 3D printed...	6 years ago
hardware	Added revision 2.0 mini	4 years ago
libraries	Added revision 2.0 mini	4 years ago
.gitignore	Added gitignore and pretty...	4 years ago
.gitmodules	Added gitignore and pretty...	4 years ago
BOM.csv	Update BOM	6 years ago
BOM.xlsx	Update BOM	6 years ago
README.md	Correct formatting	6 years ago
mmdvm_hs_dual_hat...	Add fiducials	6 years ago

README

MMDVM_HS_Dual_Hat

Duplex variant of MMDVM_HS_Hat DB9MAT DF2ET DO7EN

This PCB uses the [MMDVM_HS](#) by Andy CA6JAU. It has two ADF7021 onboard and allows for duplex operation with two time slots on DMR.





Revisions

Revision 1.0

Initial release.

Revision 1.1

LEDs D4 and D5 interchanged to align with scan order. Solder jumper JP1 was added. It can be closed if the board is used on Raspberry Pi only. It ties BOOT1 to GND. BOOT0 is controlled via GPIO. The 2x3 pin header can be unpopulated in this case to save space. We placed two coils on U1 and U2 for experiments with extended VCO. These are L11 and L12 and they are optional and do not need to be inserted for UHF use. Testpoints TP1 and TP2 on backside allows the demodulator output SNR to be measured with #define TEST_DAC compiled firmware.

This version is not recommended for (re-)production!

Revision 1.2

An accidentally deleted 3v3 wire has been added back to the PCB layout. BOM, Mouser cart and stencil stay the same as for rev1.1.

Revision 1.3

STlink header has been refactored to have the same pinout as the [MMDVM_HS_Hat](#) boards. OLED and STlink header have been exchanged. JP1 has been slightly moved. C44, C45 and C46 have been relocated nearer to the STM32F1 CPU. A table was added to the bottom silkscreen to show which TCXO is installed. This is important to the firmware. BOM and Mouser cart are still the same. The stencil has slightly changed. If you manually solder C44-C46 you can still use the old one.

BOM

Mouser cart for rev1.0 is [here](#).

[This](#) is the Mouser cart for rev1.1. L11 and L12 are optional.

Firmware installation

The device can be used on top on a RPi attached via the GPIO port or standalone and controlled via USB. Both variants require different builds of firmware and

connected via USB. Both variants require different naming or compiling and uploading the firmware. The USB connection requires firmware with bootloader support whereas the GPIO version does not.

For USB connection a bootloader has to be installed initially. This requires STLink connection. After that is done the MMDVM_HS firmware upgrade can be done via the USB connection. The STlink connection can be used as fallback if wrongly configured firmware was installed for example.

Install the firmware with bootloader support for USB connection

If you want to use the device via USB port you have to install a bootloader and build the firmware with bootloader support. As the raw device cannot be used with USB you have to use a USB-serial adapter or STlink device. Using STlink this can be done as follows:

```
user@host:/opt/MMDVM_HS$ make stlink-bl
./STM32F10X_Lib/utils/linux64/st-flash write ./STM32F10X_Lib/utils/
bootloader/generic_boot20_pc13.bin 0x8000000
2018-03-02T10:01:04 INFO src/usb.c: -- exit_dfu_mode
2018-03-02T10:01:04 INFO src/common.c: Loading device parameters...
2018-03-02T10:01:04 INFO src/common.c: Device connected is: F1 Medium-
density device, id 0x20036410
2018-03-02T10:01:04 INFO src/common.c: SRAM size: 0x5000 bytes (20
KiB), Flash: 0x10000 bytes (64 KiB) in pages of 1024 bytes
2018-03-02T10:01:04 INFO src/common.c: Attempting to write 7160
(0x1bf8) bytes to stm32 address: 134217728 (0x8000000)
Flash page at addr: 0x08001800 erased
2018-03-02T10:01:04 INFO src/common.c: Finished erasing 7 pages of
1024 (0x400) bytes
2018-03-02T10:01:04 INFO src/common.c: Starting Flash write for VL/F0/
F3 core id
2018-03-02T10:01:04 INFO src/common.c: Successfully loaded flash
loader in sram
6/6 pages written
2018-03-02T10:01:05 INFO src/common.c: Starting verification of write
complete
2018-03-02T10:01:05 INFO src/common.c: Flash written and verified!
jolly good!
./STM32F10X_Lib/utils/linux64/st-flash write bin/mmdvm_f1bl.bin
0x8002000
2018-03-02T10:01:05 INFO src/common.c: Loading device parameters...
2018-03-02T10:01:05 INFO src/common.c: Device connected is: F1 Medium-
density device, id 0x20036410
2018-03-02T10:01:05 INFO src/common.c: SRAM size: 0x5000 bytes (20
KiB), Flash: 0x10000 bytes (64 KiB) in pages of 1024 bytes
2018-03-02T10:01:05 INFO src/common.c: Attempting to write 55016
(0xd6e8) bytes to stm32 address: 134225920 (0x8002000)
Flash page at addr: 0x0800f400 erased
2018-03-02T10:01:07 INFO src/common.c: Finished erasing 54 pages of
1024 (0x400) bytes
2018-03-02T10:01:07 INFO src/common.c: Starting Flash write for VL/F0/
F3 core id
2018-03-02T10:01:07 INFO src/common.c: Successfully loaded flash
```

```
2010-03-02T07:51:07.017Z INFO src/common.c: successfully loaded flash
loader in sram
53/53 pages written
2018-03-02T10:01:12 INFO src/common.c: Starting verification of write
complete
2018-03-02T10:01:13 INFO src/common.c: Flash written and verified!
jolly good!
```

The device should now be usable as /dev/ttyACMx.

If you want to update the MMDVM_HS firmware itself later on via USB you can use the dfu upload method. The settings in Config.h are:

```
#define MMDVM_HS_DUAL_HAT_REV10
#define ENABLE_ADF7021
#define DUPLEX

#define ADF7021_14_7456
```

or #define ADF7021_12_2880

for boards with 12.288MHz TCXO.

```
#define STM32_USB_HOST
#define ENABLE_SCAN_MODE
```

Optionally:

```
#define SEND_RSSI_DATA
#define SERIAL_REPEATERS
```

Compile the firmware with "make bl" and then:

```
user@host:/opt/MMDVM_HS$ sudo make dfu devser=/dev/ttyACM3
./STM32F10X_Lib/utils/linux64/upload-reset /dev/ttyACM3 750
./STM32F10X_Lib/utils/linux64/dfu-util -D bin/mmdvm_f1bl.bin -d
1leaf:0003 -a 2 -R -R
dfu-util 0.7
```

```
Copyright 2005-2008 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2012 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to dfu-util@lists.gnumonks.org
```

```
Filter on vendor = 0x1leaf product = 0x0003
Opening DFU capable USB device... ID 1leaf:0003
Run-time device DFU version 0110
Found DFU: [1leaf:0003] devnum=0, cfg=1, intf=0, alt=2,
name="STM32duino bootloader v1.0 Upload to Flash 0x8002000"
Claiming USB DFU Interface...
Setting Alternate Setting #0
```

```
Setting alternate setting #2 ...
Determining device status: state = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 0110
Device returned transfer size 1024
No valid DFU suffix signature
Warning: File has no DFU suffix
bytes_per_hash=1100
Copying data from PC to DFU device
Starting download:
[########################################] finished!
state(8) = dfuMANIFEST-WAIT-RESET, status(0) = No error condition is
present
Done!
Resetting USB to switch back to runtime mode
```

Compile and upload manually for GPIO connection

For specific details about the firmware installation, check [these](#) instructions. The process is similar to the installation on the ZumSpot Pi. You can make use of the preconfigured scripts in MMDVM_HS. Editing Config.h by hand requires these settings:

Enable the following settings in Config.h:

```
#define MMDVM_HS_DUAL_HAT_REV10
#define ENABLE_ADF7021
#define DUPLEX
#define ADF7021_14_7456
#define STM32_USART1_HOST
#define ENABLE_SCAN_MODE
```



Optionally:

```
#define SEND_RSSI_DATA
#define SERIAL_REPEATERS
```



And for 12.288MHz TCXO equipped boards

```
#define ADF7021_12_2880
```



instead of the 14_7456 option.

Build the firmware:

```
make
```



Make sure the two BOOT jumpers are set as follows:



BOOT0: completely removed
BOOT1: set to BOOT 1 -



And finally upload the firmware to the MMDVM_HS_Dual_Hat:

```
sudo make mmdvm_hs_dual_hat
```



Update within Pi-Star

[PiStar](#) contains a binary that allow for flashing precompiled binary firmware to the MMDVM_HS_Dual_Hat. To update the firmware within PiStar execute the following commands:

```
$ rpi-rw
$ pistar-mmdvmhshatflash hs_dual_hat
```



If you have a board equipped with 12.288MHz TCXO you have to flash the according firmware type using:

```
$ pistar-mmdvmhshatflash hs_dual_hat-12mhz
```



License

This project is released under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 (CC-BY-NC-SA 3.0, <https://creativecommons.org/licenses/by-nc-sa/3.0/>) license. You may edit and share it as you like, as long as credit is given and the license is not changed. You can build as many boards for you and your friends as you like and you can even sell it to them to cover your costs, **however it is strictly forbidden to turn this into a commercial product! You are not allowed to build and sell these boards for profit!**

Releases 4

 [Revision 1.3](#) Latest
on Jun 22, 2018

[+ 3 releases](#)

Packages

No packages published

Contributors 3



phl0 Florian (DF2ET)



SkyAndy SkyAndy



mathisschmieder Mathis Schmieder