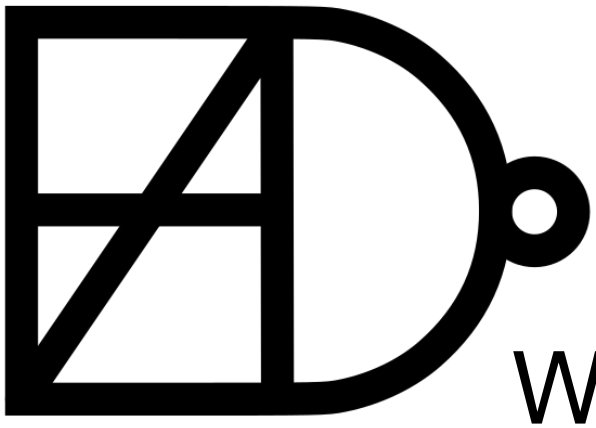


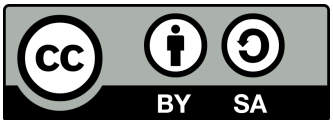
WA0EDA

# STM32-DVM-MASTR3 v1.0a

Revision: 20200602

# Table Of Contents








The WA0EDA Skunkworks may be contacted at:
5505 Plymouth Drive
Lawrence, KS 66049
n0mjs@me.com
(telephone support not provided)

# Disclaimers and Conditions of Use

### 3rd Party Software Disclaimer

The STM32-DVM-MASTR3 contains various 3rd party softwares redistributed under various versions of the GPL. WA0EDA and the Regents of the K0USY Group make no claims or warranties regarding such software and we are not responsible for its use. Using the STM32-DVM-MASTR3 requires acceptance of and compliance with the license terms of included 3rd party software.

### Use at Your Own Risk

The STM32-DVM-MASTR3, as a whole unit, is not UL (or similarly) listed, and has not been tested for FCC Part 15 (or other) compliance. By using the STM32-DVM-MASTR3, the user agrees to indemnify the WA0EDA Club, the Regents of the K0USY Group, Cortney T. Buffington, N0MJS and other affiliated entities against liability resulting in its use. The STM32-DVM-MASTR3 is intended for use in experimental and educational environments consistent with the amateur radio service.

### Configuration Disclaimer

WA0EDA provides configuration advice, but is not responsible or liable for providing comprehensive or authoritative information about MMDVM or the MASTR III. Configurations presented in this manual represent how we set up MMDVM and the MASTR III with the STM32-DVM-MASTR3. This manual does not attempt to cover alternative configurations, and is not an authoritative source of information for MMDVM or the MASTR III.

# Overview

The STM32-DVM-MASTR3 is an integrated MMDVM modem and Host designed to plug into any of the three “spare” slots in the station T/R shelf. The STM32-DVM-MASTR3 is an all-in-one solution for adding MMDVM capabilities to the MASTR III, no additional hardware or software is necessary.

The modem section is typical of most MMDVM hardware interfaces in that it includes 3rd order 4kHz low-pass filters for the TX & RX audio paths, and signal conditioning for logic signals. The STM32-DVM-MASTR3 shares a common design heritage with, among others, the Repeater Builder STM32-DVM from Scott Zimmerman, N3XCC.

Unique to the STM32-DVM-MASTR3 is the inclusion of a host SBC (single-board computer) permanently affixed to the card. Because the circuits are optimized for the MASTR III station, only TX and RX audio level adjustments are required. The SBC provided on the STM32-DVM-MASTR3 is a NanoPi NEO from FriendlyElec, and comes pre-loaded with Armbian Linux and a fully functioning, start-on-boot MMDVMHost installation. The only end-user tasks required are to modify the MMDVM.ini configuration file to meet their needs, and transmit audio/key routing in the MASTR III programming software (MSEDIT).

Programming access to the STM32F4 series microcontroller is provided for those who wish to use alternative or updated firmware. The NanoPi NEO comes with 512MB of RAM and a micro SDHC card which may also be re-used for customers wishing to use their own builds or other 3rd party images, such as PiStar.

Because the STM32-DVM-MASTR3 is purpose built for the MASTR III, not only is configuration and calibration much easier, but it also allows for deeper integration with the MASTR III system itself by adding the option to connect the third UART of the NanoPi NEO to the programming/diagnostics interface of the station.

# Conventions used in this manual

The STM32-DVM-MASTR3 V1.0a may be equipped with either an STM32F446RET6 or an STM32F405RTG6 microcontroller for the MMDVM modem. Both are very similar, but used alternately based on availability at the time of manufacture. STM32F4xx is used to mean either. When “4xx” is seen as part of a file name in code examples, it should be replaced with the appropriate number, either “446” or “405”, for the microcontroller used on your board.

## Physical and Electrical Specifications

| | |
|---|---|
| Size: | 156mm x 108mm (connectors, actuators and face plate excluded) |
| Radio Connector: | 96 pin male “eurocard” (3 x 32) |
| Voltage: | 10-18VDC***** |
| Current: | 2A maximum, 150mA typical |
| Network: | RJ45 10/100Mbps Ethernet, auto-sensing |
| PTT Outputs: | Open Collector 50VDC max, 100mA max sink |
| Lockout Input: | Transistor Buffered, 50VDC max |
| RSSI Input: | 3.3VDC maximum useable |
| Audio Input/Output: | AC Coupled, preconfigured for MASTR III |
| Host SBC: | NanoPi NEO 512MB RAM, 16GB or 32GB micro SDHC |
| Modem processor: | STM32F446RET6 or STM32F405RGT6 |
| MMDVM Indicators: | Power, Activity, Heartbeat, PTT, RX Clip, COR, DMR, P25, NXDN, YSF, D* |

***WARNING: NEVER APPLY POWER DIRECTLY TO THE +5VDC or +3.3VDC BUS ON THE STM32-DVM-MASTR3, INCLUDING THE MICRO USB POWER PORT ON THE NANOPI NEO. SEVERE DAMAGE MAY RESULT.**

# MASTR III Station Requirements

Several variations of the MASTR III station exist, including backplanes. The STM32-DVM-MASTR3 specifically uses the following connections on any of the “Spare” slots in the T/R shelf. Please ensure the necessary signals are available prior to attempting to install the STM32-DVM-MASTR3 into any MASTR III station.

Discriminator Audio is recovered from one of two jumper selectable (JP11) sources:

**6A RCVR_VOL\SQ_HI**

*– or –*

**2B SYS_VOL\SQ_HI**

Transmit Audio is injected to:

**26B EXT_HSD**

Modem PTT connected to:

**31C LOCAL_PTT**

For installations that share the station between MMDVM and analog applications, modem lockout can be connected via jumpers installed at JP8 to any combination of the following:

**28B REMOTE_PTT_OUT**
**29B RX_1_MUTE**
**31B REMOTE_PTT IN**

# NanoPi NEO Access

The STM32-DVM-MASTR3 is shipped with a micro SDHC memory card installed in the NanoPi NEO running a customized version of Armbian Linux that includes pre-built MMDVMHost, MMDVMCal (renamed CalMMDVM) and utilities necessary for operation and maintenance. There is no host firewall configured on the pre-built version, but may be added by the end-user. The NanoPi NEO may be accessed over the network by secure shell (SSH). Writes to the SDHC memory card are buffered for up to 10 minutes. This saves wear and tear by flushing changes to the card much less frequently. This behavior is normally transparent to the end user, but be sure that you properly shutdown or reboot the NanoPi NEO after making changes to ensure all changes are written to the card. Making changes and immediately pulling the power will cause the changes to be lost.

As shipped, the SSH server is listening on port 32 (not the standard SSH port, 22). Direct root logins via SSH are also disabled. The as-shipped configuration is:

- SSH Port: 32
- Username: mmdvm
- Password: mmdvm
- Root Password: mmdvm

The default hostname that will be reported to your DHCP server is “STM32-DVM-MASTR3”. Use this to find the IP address assigned by your DHCP server.

# MMDVM Configuration

MMDVM configuration requirements for the STM32-DVM-MASTR3 and MASTR III are minimal. Specifically, the following “Modem” stanza from the MMDVM.ini (contained in /etc/ on our Armbian build) shows recommended values. ***NOTE: Depending on the injection side (high/low) used in a given receiver, RXInvert may need to be set to 1. This setting is station/module/frequency dependent.***

```
[Modem]
Port=/dev/ttyS1
Protocol=uart
TXInvert=1
RXInvert=0
PTTInvert=0
TXDelay=100
RXOffset=0
TXOffset=0
DMRDelay=0
RXLevel=50
TXLevel=50
RXDCOffset=0
TXDCOffset=0
RFLevel=0
RSSIMappingFile=/usr/local/etc/rssi.dat
Trace=0
Debug=0
```

The receiver (discriminator audio may be pulled from one of two sources within the station and is selectable with Jumper JP11 to use either SYS_VOL/SQ_HI (upper two pins jumpered while looking at the component side of the circuit board) or RCVR_VOL/SQ_HI (lower two pins jumpered while looking at the component side).

It is also necessary to invert the TX audio signal from MMDVM, which is set by default in the supplied MMDVM.ini file. If the TX audio is not inverted, a subscriber radio will appear to wake up the repeater, but will never complete the handshake, and result in a log entry from MMDVMHost similar to the one shown below.

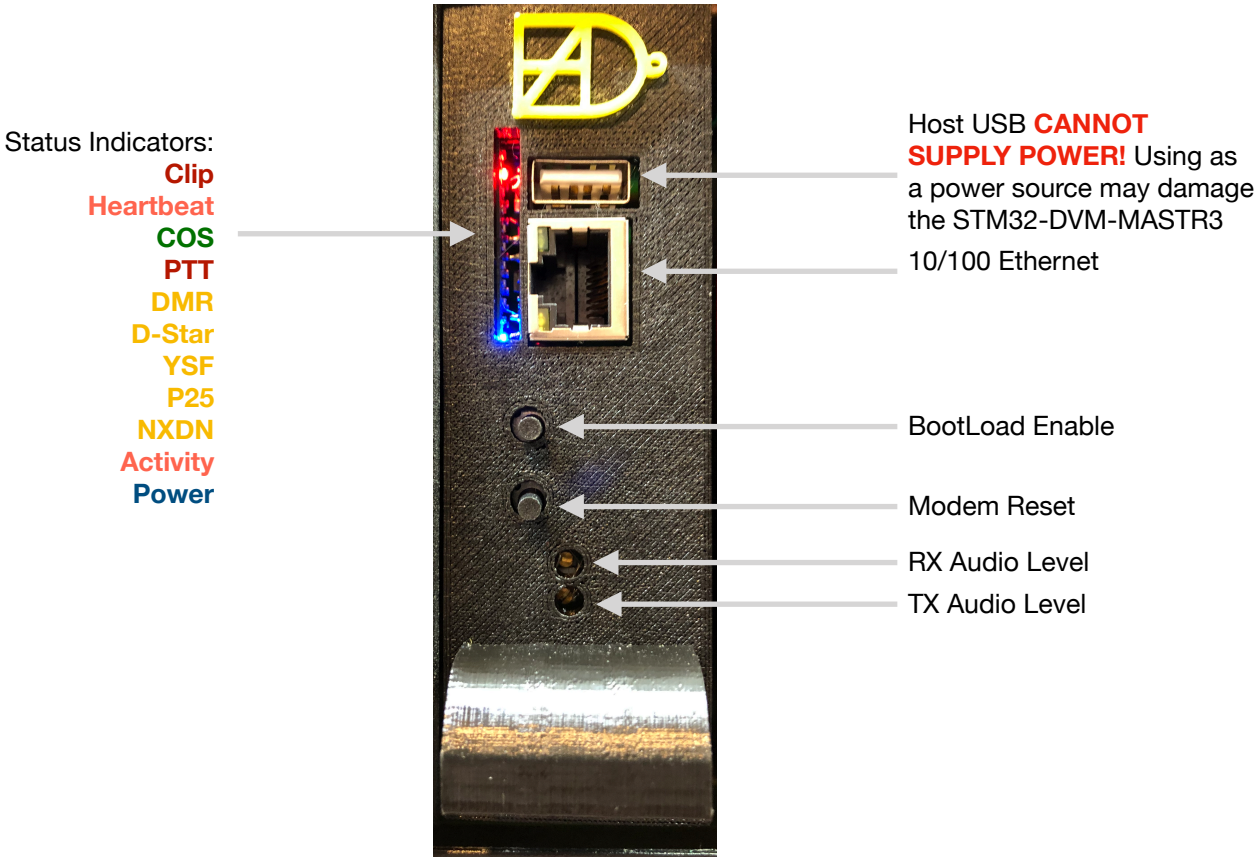
```
D: 2019-06-14 14:44:34.358 Downlink Activate CSBK
D: 2019-06-14 14:44:34.358 0000:  B8 00 00 00 FF FF FF 2F 9B E5 DF 60
M: 2019-06-14 14:44:34.358 Downlink Activate received from 3120101
```

The other, optional task, is RSSI. This is by no means mandatory, but remains a popular feature within MMDVM. For RSSI mapping, the MMDVM.ini requires a user supplied mapping file with measured values of the MMDVM RSSI ADC input, and the corresponding signal level in dBm. Not everyone has the means to measure this. WA0EDA has included a map file for a typical MASTR III,

but there is some variation from station to station. The included file (/usr/local/etc/rssi.dat) should work well for most users. The MASTR III RSSI mapping table is included later in this manual.

## Front Panel Features

All connectors, controls and indicators for calibration and operation are located on the front panel of the STM32-DVM-MASTR3. Please see the figure below for a description of each item:



# Calibration and Configuration

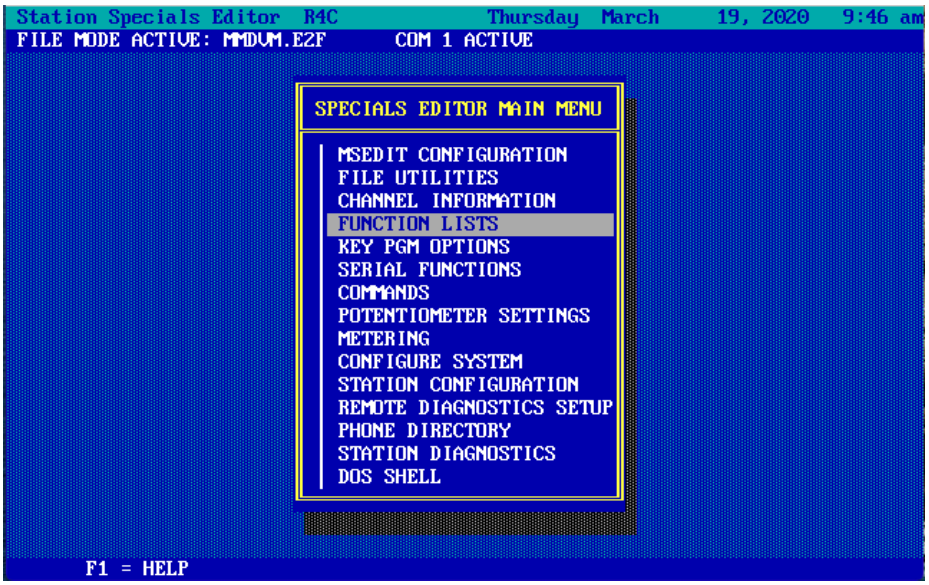
### MASTR III Station Configuration

It is important to start with a station that has already been properly aligned and calibrated for analog (20/25kHz spacing) operation as originally intended. Specifically, transmit deviation/limiting levels and receiver tuning (if necessary) should be adjusted per the MASTR III station manual first, and then calibrate the STM32-DVM-MASTR3 to the station.

WA0EDA recommends you program the channel information into your station first, and then perform transmit audio calibration. MASTR III programming is beyond the scope of this manual. Please refer to the MASTR III station documentation.

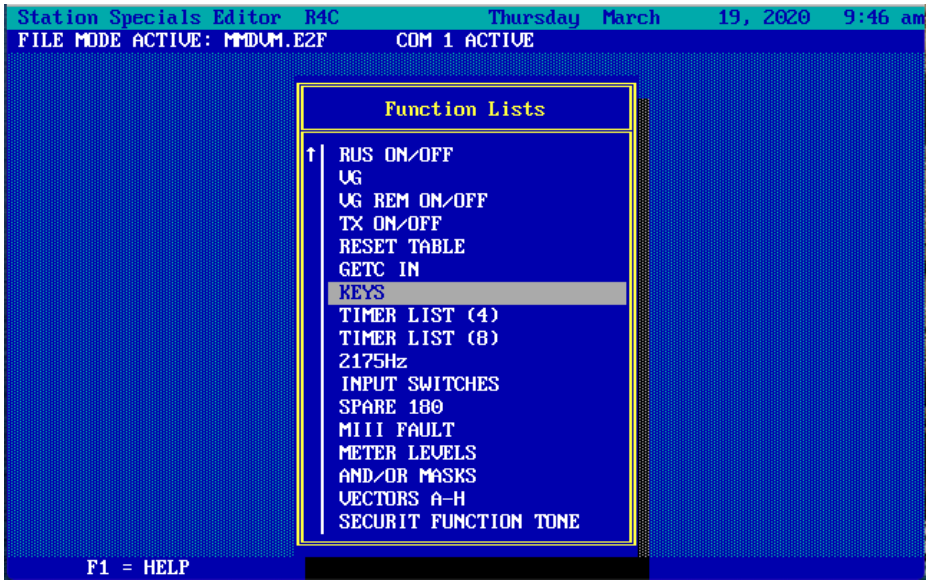
The STM32-DVM-MASTR3 uses the “EXT_HSD” input for the modem’s TX audio signal. This input was intended specifically for high-speed data. Using this input requires mapping a PTT signal as the key line. We have chosen the LOCAL_PTT input, leaving both the repeat and remote PTT inputs for the internal or an external analog controller.

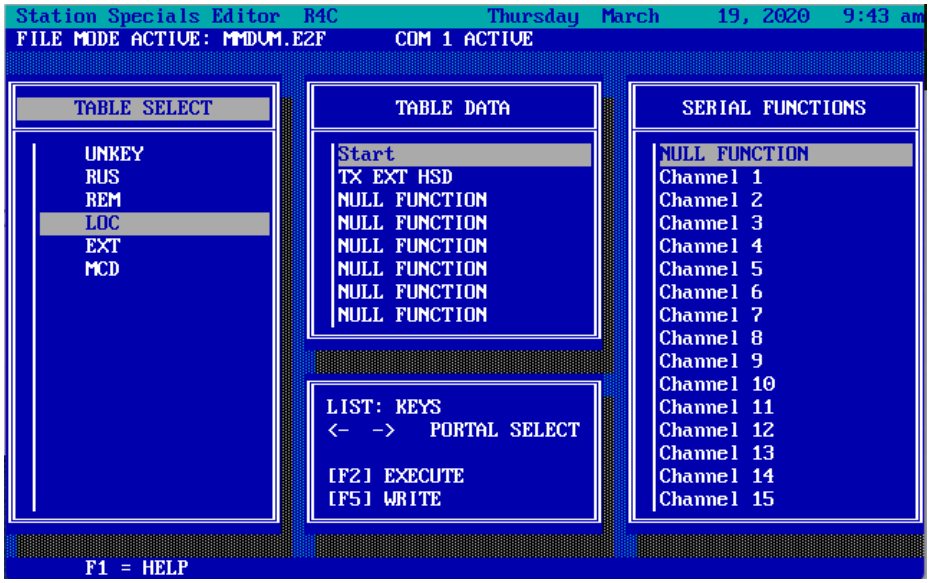
Mapping of the PTT signal is achieved though the use of the MSEDIT software, and requires editing the “KEYS” Function List. In MSEDIT, navigate to the “FUNCTION LISTS” menu item as show below:

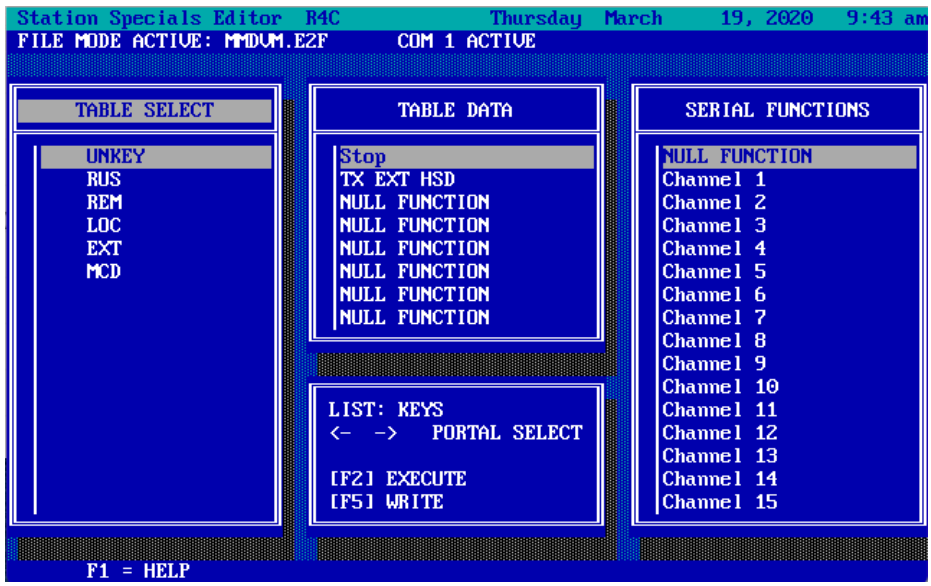



Upon entering the “FUNCTION LISTS” item in the Specials Editor Main Window, navigate to the “KEYS” Function list as shown on the next page.

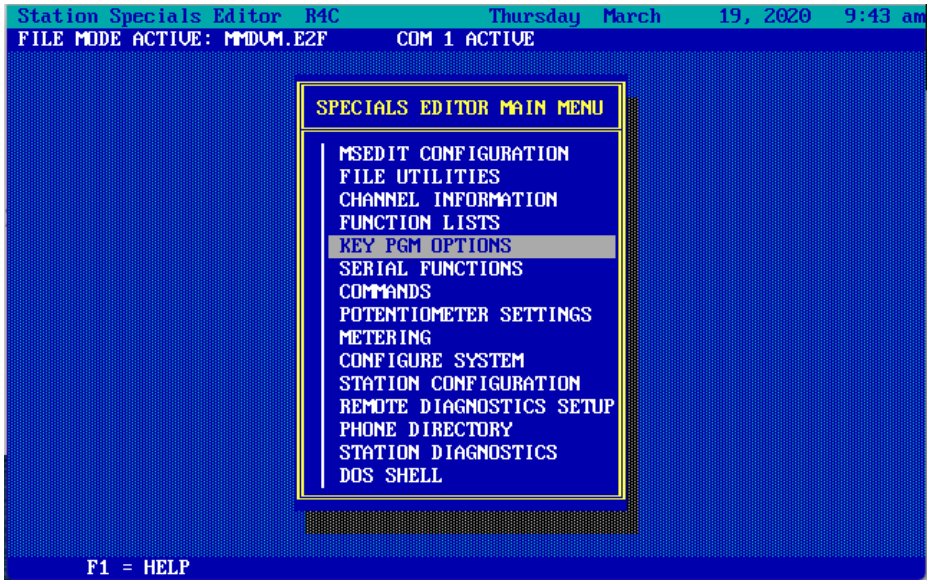
Once in the “KEYS” menu item, two tables must be modified. First we map the “LOC” local PTT to KEY from (start) the TX EXT HSD input. Similarly the “UNKEY” table must map to un-key from (stop) the TX EXT HSD input. Images for each of these steps are shown on the next page.


Station Specials Editor R4C Thursday March 19, 2020 9:46 am
FILE MODE ACTIVE: MMDVM.E2F COM 1 ACTIVE
Function Lists
RUS ON/OFF
VG
VG REM ON/OFF
TX ON/OFF
RESET TABLE
GETC IN
KEYS
TIMER LIST (4)
TIMER LIST (8)
2175Hz
INPUT SWITCHES
SPARE 180
MIII FAULT
METER LEVELS
AND/OR MASKS
VECTORS A-H
SECURIT FUNCTION TONE
F1 = HELP

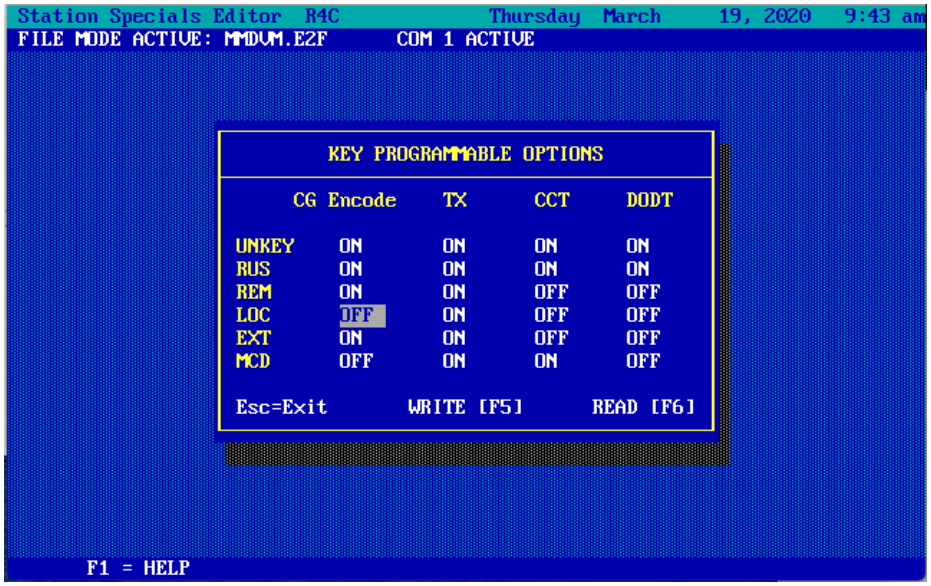

Station Specials Editor R4C Thursday March 19, 2020 9:43 am
FILE MODE ACTIVE: MMDVM.E2F COM 1 ACTIVE
TABLE SELECT
UNKEY
RUS
REM
LOC
EXT
MCD
TABLE DATA
Start
TX EXT HSD
NULL FUNCTION
NULL FUNCTION
NULL FUNCTION
NULL FUNCTION
NULL FUNCTION
NULL FUNCTION
LIST: KEYS
<- -> PORTAL SELECT
[F2] EXECUTE
[F5] WRITE
SERIAL FUNCTIONS
NULL FUNCTION
Channel 1
Channel 2
Channel 3
Channel 4
Channel 5
Channel 6
Channel 7
Channel 8
Channel 9
Channel 10
Channel 11
Channel 12
Channel 13
Channel 14
Channel 15
F1 = HELP


Station Specials Editor R4C Thursday March 19, 2020 9:43 am
FILE MODE ACTIVE: MMDVM.E2F COM 1 ACTIVE
TABLE SELECT
UNKEY
RUS
REM
LOC
EXT
MCD
TABLE DATA
Stop
TX EXT HSD
NULL FUNCTION
NULL FUNCTION
NULL FUNCTION
NULL FUNCTION
NULL FUNCTION
NULL FUNCTION
LIST: KEYS
<- -> PORTAL SELECT
[F2] EXECUTE
[F5] WRITE
SERIAL FUNCTIONS
NULL FUNCTION
Channel 1
Channel 2
Channel 3
Channel 4
Channel 5
Channel 6
Channel 7
Channel 8
Channel 9
Channel 10
Channel 11
Channel 12
Channel 13
Channel 14
Channel 15
F1 = HELP

There is one remaining configuration item necessary if you intend to use the same channel for analog repeat – either as a “fall back” emergency mode, or full-time. So that items like the time-out timer (useful for in-cabinet analog repeat), drop-out delay (DODT) and CTCSS (Channel Guard, or CG) are not applied to the local (LOC) PTT input, which we have mapped to the EXT_HSD data signal input used by MMDVM, select the “KEY PGM OPTIONS” from the Specials Editor Main Menu:




Once into the Key Programmable Options, make sure that options for CG Encode, TX, CCT (time-out timer) and DDOT (TX drop-out timer) are set as shown below:




Other configurations certainly exist, and certainly, depending on your application may be necessary. The myriad of possibilities for programming the station are beyond the scope of this manual, however, these settings will get you up and running with MMDVM quickly, with the ability to keep channel settings configured for analog repeat.

### Modem Calibration

The STM32-DVM-MASTR3 contains two 25-turn trim pots, one for RX audio and one for TX audio. To set the transmits audio level, use MMDVMCal (renamed CalMMDVM in our Armbian image to make tab-completion more effective) and either set TX deviation with the DMR test tone as described with the MMDVM documentation, or use a spectrum analyzer (25kHz sweep works well) to set modulation as close to Bessel zero (nulling the signal at the carrier center frequency) as possible. Turning the TX Audio Level pot on the STM32-DVM-MASTR3 clockwise will increase the audio output.

The receiver level is set by observing the red “clip” LED on the front panel. An appropriate signal level is achieved when the clip light just extinguishes while there is no signal (only discriminator noise) on the channel. We’ve had the best results by turning the pot up high enough (clockwise) for clip to remain on, if it is not already, then back off (counter clockwise) just until the light stays off. It is normal for the clip to light when the station is first powered up, and should settle down with about 5-10 seconds as the modules stabilize.

RSSI is fixed and requires no calibration. WA0EDA has supplied a mapping file with our software load, but for absolute accuracy, we recommend creating one specifically for your station. Testing has shown that the MASTR III RSSI signal is useable from -128dB through about -60dB, at which point, stronger signals do not produce a larger RSSI signal voltage – in fact, the RSSI voltage has been observed to go back down on **very strong signals** of -40dB or more.

## Modem Firmware

STM32-DVM-MASTR3 boards are shipped with a mature version of MMDVM. Generally, it should not require firmware updates, but at some point, features may be added and users may wish to update firmware. The STM32-DVM-MASTR3 is equipped with the means to update firmware in the field (i.e. remotely), or with manual intervention by the operator. In either case, the on-board NanoPI NEO is used to upload firmware into the STM32F4xx series microcontroller. For those with any reservations about performing a firmware update, we strongly recommend using the manual method, at least the first time, as this method is somewhat simpler.

All utilities needed to perform these procedures are included on the micro SDHC card supplied with your STM32-DVM-MASTR3, and all procedures can and should be performed from the mmdvm user’s home directory as the mmdvm user (do not su or sudo). If you have changed the software, a more detailed article, including information on the necessary utilities, for uploading firmware is available at the KS-DMR website:

http://ks-dmr.net/2019/05/28/STM32-DVM-MTR2K-deep-dive-updating-firmware

The article was originally written for the STM32-DVM-MTR2K (Motorola MTR2000 version), however this portion is electrically identical to the STM32-DVM-MASTR3, and is equally applicable to both modems. WA0EDA would like to acknowledge the assistance of Steve, N4IRS, and Mike, N4IRR, at DVSwitch for their contributions to the firmware build process necessary for our STM32F4xx-based modems. Firmware is available at:

http://ks-dmr.net/STM32-DVM-MTR2K-information

Again, the firmware for the MASTR III and MTR2000 versions are identical, the important difference being which microcontroller is used. Modems are equipped with either and STM32F446RET6 or STM32F405RGT6. Please be sure to download the correct firmware file. The wrong file will load and verify, but will not function. Where you see “4xx” in these examples, it should be replaced with either “446” or “405” depending on which version your board is equipped with.

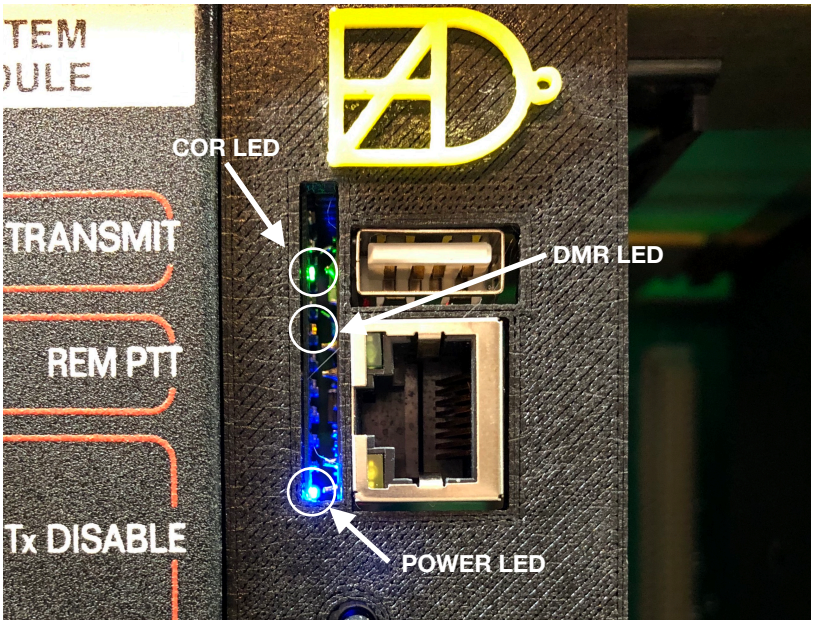
It is fastest to download this directly into the NanoPi NEO. Log in as the mmdvm user and in the home directory (where you are after logging in) use wget to download the firmware:

```
wget http://ks-dmr.net/wp-content/uploads/2019/12/mmdvm_f4xx.hex
```

Before firmware can be uploaded, the STM32F4 series microcontroller must be placed in bootloader mode. There are two ways to do this: Physically, with the front panel pushbuttons, and via software with GPIO pins of the NanoPi NEO. Before proceeding with either method, please ensure the MMDVMHost process is terminated. If MMDVMHost is running during the firmware update process, the update will fail.

**Manual Method**

Place the STM32F4xx into bootstrap mode by holding in the **BootLoad Enable** button on the front panel while momentarily pressing the **Modem Reset** button. Note that while the board is being reset only the Power LED should be illuminated. Once the board has successfully been reset and is in bootstrap mode, the **power and activity LEDs will remain lit on the STM32F446**, while only **the power LED will be illuminated on the STM32F405**. The **DMR and COR LEDs are dimly lit on both boards**; it may not be possible to see the DMR LED in a well lighted room.




After the STM32F4xx enters bootloader mode, send the following command at the NaoPi NEO’s shell prompt (substituting the proper firmware image name for your board):

```
stm32flash -v -w ./mmdvm_f4xx.hex -R /dev/ttyS1
```

The process takes some time to complete. Specifically while the stm32flash utility is erasing the flash memory on the STM32F4xx, there is no feedback to the user. That's ok; if you've made it this far, the processing is working correctly. After erase, stm32flash will program and verify the onboard flash memory then reset the STM32F4xx. Output should be similar to the following:

```
stm32flash 0.5

http://stm32flash.sourceforge.net/

Using Parser : Intel HEX
Interface serial_posix: 57600 8E1
Version      : 0x31
Option 1     : 0x00
Option 2     : 0x00
Device ID    : 0x0421 (STM32F446xx)
- RAM        : 128KiB  (12288b reserved by bootloader)
- Flash      : 512KiB (size first sector: 1x16384)
- Option RAM : 16b
- System RAM : 30KiB
Write to memory
Erasing memory
Wrote and verified address 0x0800fea0 (100.00%) Done.

Resetting device... done.
```

After upload is completed you should see the normal power-up sequence on the STM32-DVM-MASTR3 modem status LEDs. Firmware update is now complete and you may restart MMDVMHost.

The most common problem is a failure of the stm32flash utility to properly initialize the STM32F4xx series microcontroller. By far, the most common causes are that the microcontroller is either not in bootloader mode, or MMDVMHost was either running or started while the microcontroller was in bootloader mode. An initialization failure will produce an error such as this:

```
stm32flash 0.5

http://stm32flash.sourceforge.net/

Using Parser : Intel HEX
Interface serial_posix: 57600 8E1
Failed to init device.
```

**In-Field (Remote) Method**

The STM32-DVM-MASTR3 includes necessary connections between GPIO (General Purpose Input/Output) lines on the NanoPi NEO and the STM32F4 series microcontroller for placing the micro controller into bootstrap mode, without the use of jumpers. A script, "update_modem.sh" has been provided that will automate the procedure. The script expects to be run from the same

directory as the firmware file, but does need you to tell it the firmware file's name. Use the following command to run the script, replacing "4xx" with either "405" or "446" depending the chip used:

```
./update_modem.sh mmdvm_f4xx.hex
```

The script works using the "gpio" command line utility from the WiringPi utility package to manipulate GPIO (placing the microcontroller into bootloader mode) before running the stm32flash utility. The sequence is as follows:

- Set GPIO pins to output:
  - gpio mode 7 out
  - gpio mode 1 out
- Set the BOOTLOADER line high (pin 7), toggle the RESET line (1), Set BOOTLOADER low
  - gpio write 7 1
  - gpio write 1 0
  - gpio write 1 1
  - gpio write 7 0
- Set GPIO pins to input:
  - gpio mode 7 in
  - gpio mode 1 in

If completed successfully, the script will provide output similar to the following, with errors resembling those from the manual method section.

```
Setting Up GPIO Pins
Sending STM32 Device Into Bootloader Mode
Resetting GPIO Pins
Attempting to program STM32 device
The following output is from stm32flash:

stm32flash 0.5

http://stm32flash.sourceforge.net/

Using Parser : Intel HEX
Interface serial_posix: 57600 8E1
Version      : 0x31
Option 1     : 0x00
Option 2     : 0x00
Device ID    : 0x0421 (STM32F446xx)
- RAM        : 128KiB  (12288b reserved by bootloader)
- Flash      : 512KiB (size first sector: 1x16384)
- Option RAM : 16b
- System RAM : 30KiB
Write to memory
Erasing memory
Wrote and verified address 0x0800fea0 (100.00%) Done.

Resetting device... done.

stm32flash completed
```

## Bus Expansion

To create a more extensible system, the i2C bus on the NanoPi NEO (not used by MMDVM) is completely available for end-user additions/customizations. SDA, SCL and GND have been conveniently connected to a .1” pin header (JP14) near the NanoPi NEO, with pull-up resistors installed on the board.

See the information in section V1.0a Jumpers and Connections for pin assignments.

## For More Information

Additional information, updates, detailed articles, etc. regarding the STM32-DVM-MASTR3 can be found in the “Shop Talk” section of the KS-DMR website (http://www.ks-dmr.net).

WA0EDA Skunkworks is the R&D subsidiary of the K0USY Group (http://www.k0usy.org), which are both amateur radio clubs and founding members of KS-DMR.

The K0USY Group is the author of popular DMR networking software packages HBlink, DMRlink and dmr_utils (http://github.com/n0mjs710), operates many networked DMR repeaters in northeast and north-central Kansas, and provides hosted DMR networking services to the amateur radio community.

## MASTR III RSSI Map Table

```
2698 -56
2697 -58
2696 -60
2693 -62
2685 -64
2676 -66
2661 -68
2640 -70
2596 -72
2533 -74
2473 -76
2407 -78
2346 -80
2251 -82
2191 -84
2146 -86
2109 -88
2073 -90
2015 -92
1966 -94
1922 -96
1865 -98
1809 -100
1688 -102
1625 -104
1578 -106
1534 -108
1492 -110
1419 -112
1368 -114
1320 -116
1262 -118
1202 -120
1132 -122
1093 -124
1061 -126
1034 -128
```

# V1.0a Jumpers and Connections

STM32-DVM-MASTR3 Connections:

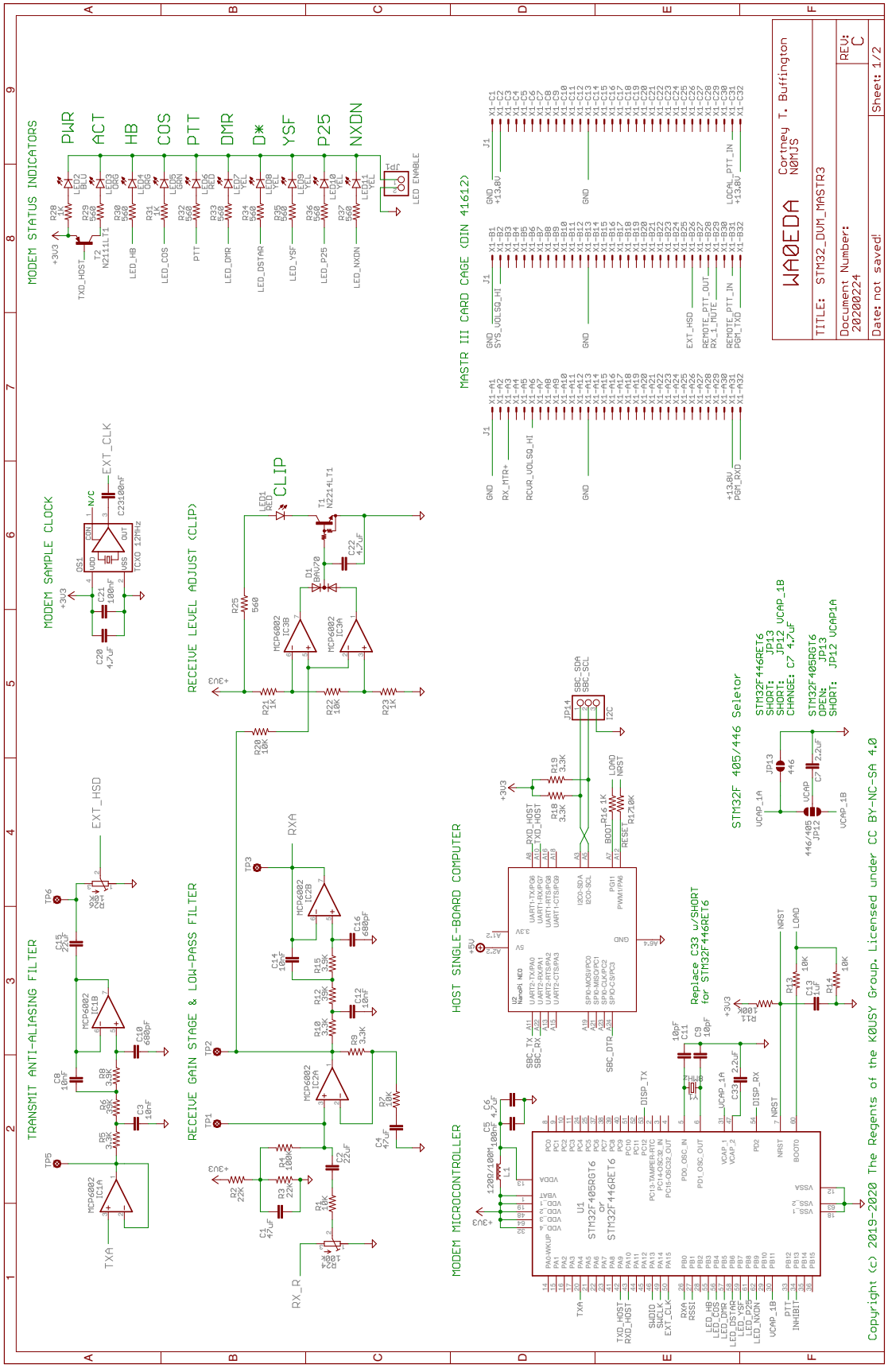
| ID | Name | Purpose | Type | Description |
|---|---|---|---|---|
| J1 | J1 | MASTR III | 96 Pin (3x32) Eurocard | Connects STM32-DVM-MASTR3 to the MASTR III |
| JP1 | LED ENABLE | Enable MODEM LEDs | .1” Pin Header 1x2 | Enables MODEM status LEDs |
| JP2 | EXT PWR +7-15V | Testing/ Programming | .1” Pin Header 1x2 | External power for the STM32-DVM-MASTR3 while not inserted into a MASTR III T/R Shelf |
| JP3 | STM32 ICSP | Programming | .1” Pin Header 2x3 | In-circuit SPI programmer for the STM32F4xx |
| JP4 | MODEM DISP | Peripheral | .1” Pin Header 1x4 | Serial “repeater” connection via the MODEM |
| JP5 | MODEM SERIAL | Testing/ Programming | .1” Pin Header 1x5 | Exposes serial (/dev/ttyS1) connection between the MODEM (STM32F4xx) and HOST (NanoPi NEO) |
| JP6 | SBC DISP | Peripheral | .1” Pin Header 1x4 | Serial (/dev/ttyS2) connection from the Host (NanoPi NEO) for serial display, peripheral or communication with the ATmega328P I/O processor |
| JP7 | PGM Async Enable | Access Station Serial | .1” Pin Header 2x2 | Connects NanoPi NEO (/dev/ttyS2) to the MASTR III T/R shelf serial programing/diagnostic interface |
| JP8 | Inhibit Source | Modem Lockout | .1” Pin Header 2x3 | Selects modem lockout source(s) |
| JP11 | RX Audio Source | Modem Input | .1” Pin Header 1x3 | Selects RX audio source from MASTR III backplane |
| JP12 | 446/405 | Processor Select | Solder Pad 1x3 | Bridge to select STM32F446 or STM32F405 (see schematic) |
| JP13 | 446 | Processor Select | Solder Pad 1x2 | Bridged for STM32F445 microcontroller |
| JP14 | I2C | Peripheral | .1" Pin Header 1x3 | Access to SBS I2C bus for peripheral expansion |
| +3.3V | 3.3V DC | Testpoint | Solder Pad | 3.3V modem supply rail |
| +5V | 5.0V DC | Testpoint | Solder Pad | 5.0v modem supply rail |
| GND | GND | Testpoint | Solder Pad | Ground bus |
| TP1 | RX Buffer Feedback | Testpoint | Solder Pad | RX Buffer feedback loop test point |
| TP2 | RX Buffer Output | Testpoint | Solder Pad | Output of RX audio buffer used for calibration/ testing |
| TP3 | RX to MODEM | Testpoint | Solder Pad | RX audio as presented to the MODEM ADC |
| TP5 | TX Buffer Output | Testpoint | Solder Pad | DAC output after buffer amplifier |
| TP6 | TX to Repeater | Testpoint | Solder Pad | TX audio output from filter |
| TP7 | RSSI | Testpoint | Solder Pad | RSSI from receiver, as supplied to the MODEM ADC |

# MASTR III Spare T/R Shelf Connector:

| ID | Name | Purpose | Description |
|---|---|---|---|
| A1,A13, B1, B13, C1, C13 | GND | Signal Ground | Combines station DGND and AGND |
| A3 | RX_MTR+ | RSSI | Station RSSI Output |
| A6 | RCVR_VOL\SQ_HI | RX Audio | Preferred discriminator audio source |
| A31, C2, C32 | +13.8VDC | Modem Power | Station +13.8VDC supply |
| A32 | PGM_RXD | Station Programming | Station async serial programming/diagnostic interface |
| B2 | SYS_VOL\SQ_HI | RX Audio | Secondary discriminator audio source |
| B26 | EXT_HSD | TX Audio | Station High Speed Data input (flat response) |
| B28 | REMOTE_PTT_OUT | Modem Lockout | Station PTT towards wireline interface |
| B29 | RX_1_MUTE | Modem Lockout | CAS/RUS output from main receiver |
| B31 | REMOTE_PTT_IN | Modem Lockout | Station PTT input from wireline interface |
| B32 | PGM_TXD | Station Programming | Station async serial programming/diagnostic interface |
| C31 | LOCAL_PTT_IN | Modem PTT Output | Local PTT used by modem to key the station |

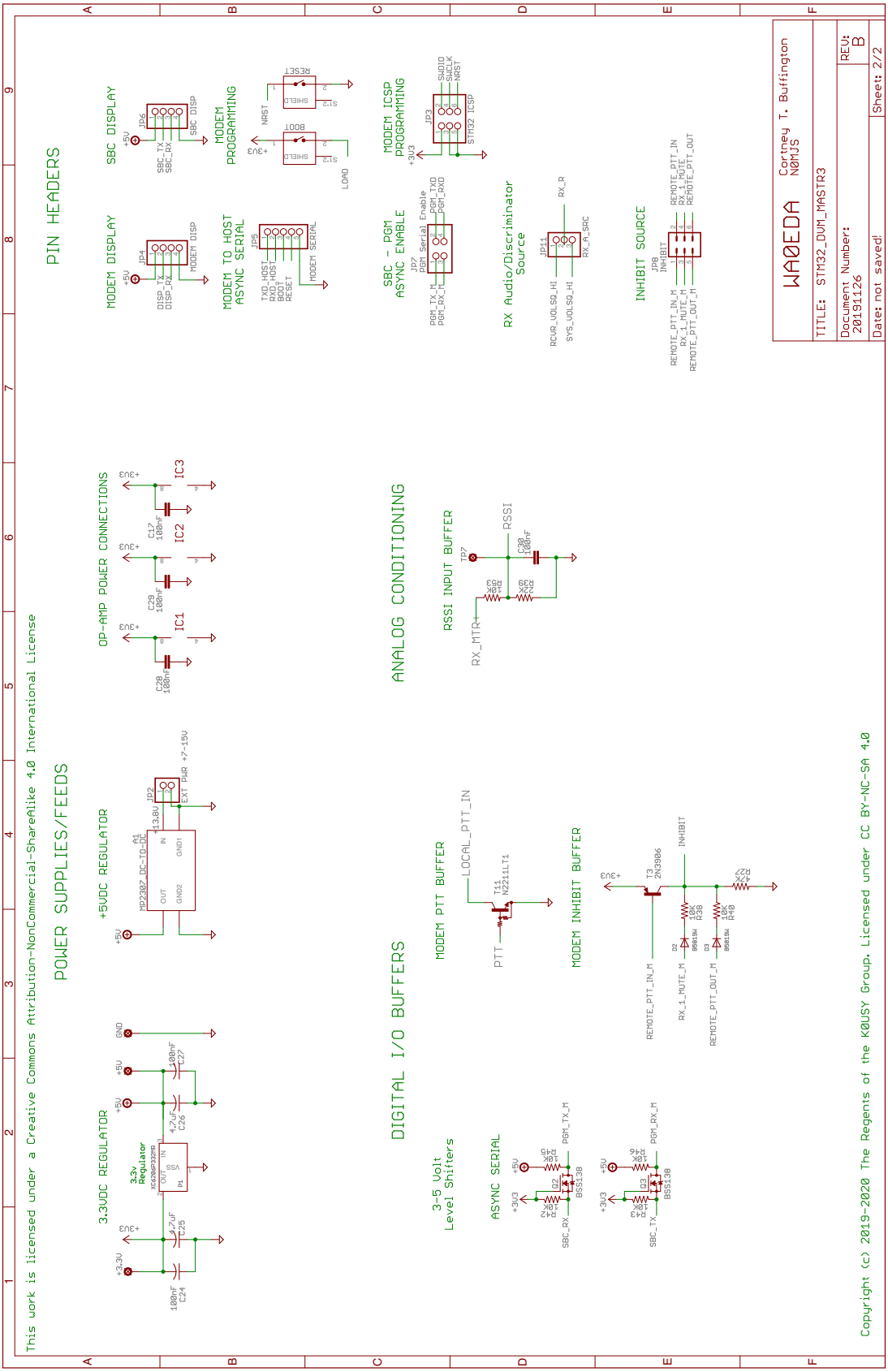
# V1.0a Schematic Diagram

(page 1 of 2)

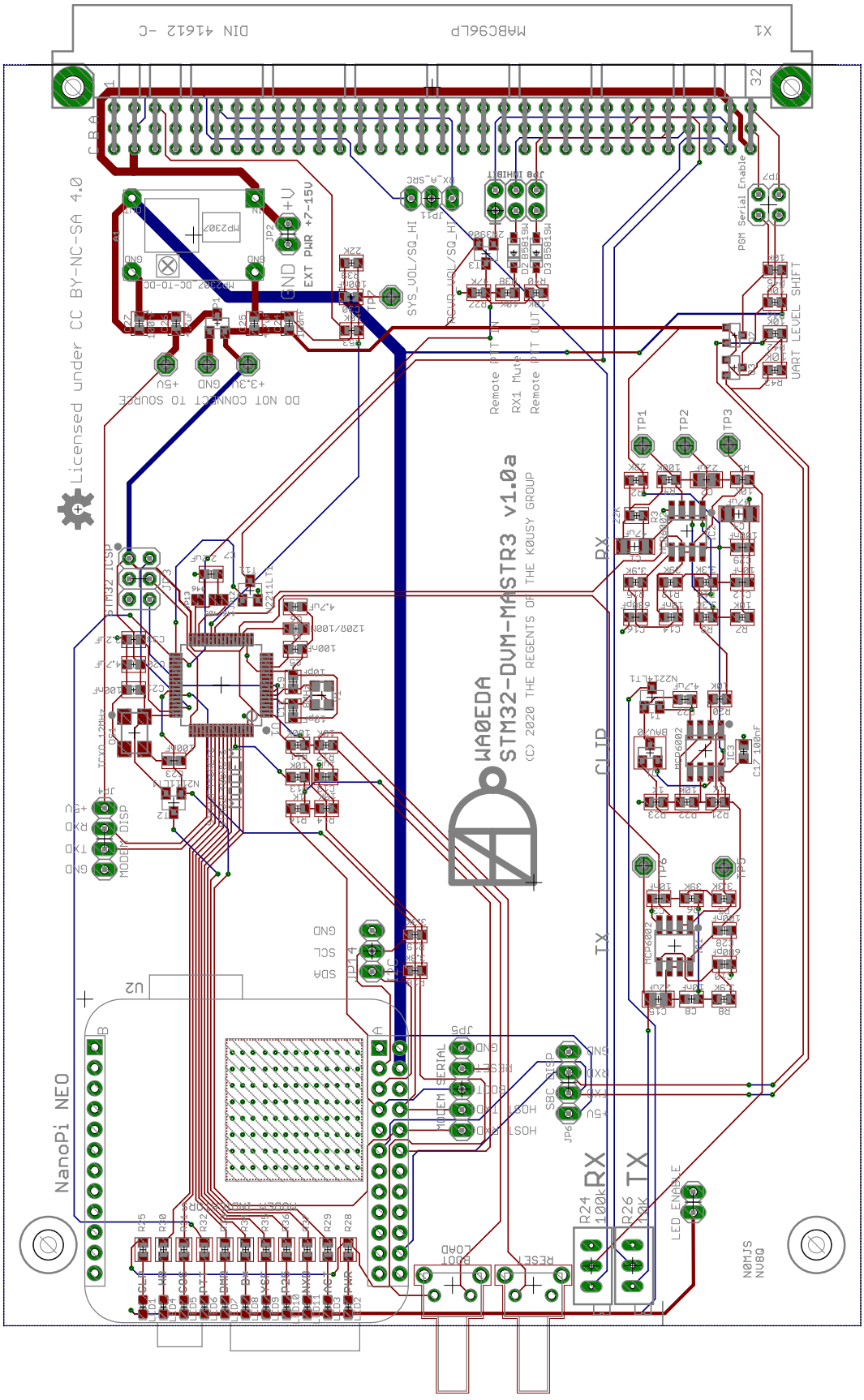


# V1.0a Schematic Diagram

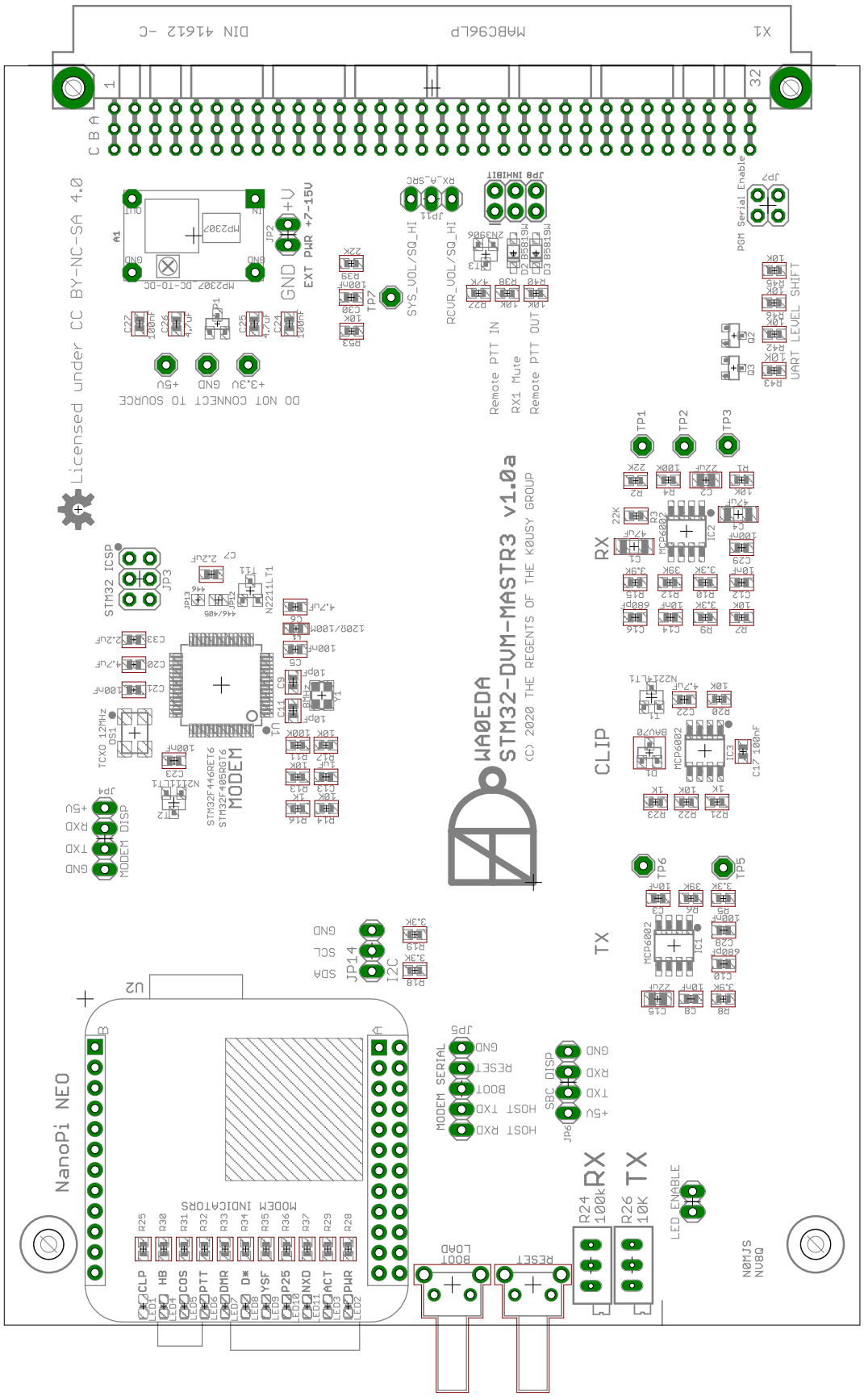
(page 2 of 2)



# V1.0a PCB Layout Diagram



# V1.0a Parts Placement Diagram



# V1.0a Parts List

STM32-DVM-MASTR3 V1.0a

| Value | Device | Package | Parts | Sheet |
|---|---|---|---|---|
| **446 (uP)** | JUMPER-SMT | SMT-JUMPER_2 | JP13 | 1 |
| **560** | R-US_R0603 | R0603 | R25, R29, R30, R32, R33, R34, R35, R36, R37 | 1 |
| **SPST** | Pushbutton | TE_2-1825027-0 | BOOT, RESET | 2 |
| **100K** | Chip Resistor | R0603 | R4, R11 | 1 |
| **100K** | 25 Turn Trimpot | RTRIMT93XA | R24 | 1 |
| **100nF** | Ceramic Capacitor | C0603 | C5, C17, C21, C23, C24, C27, C28, C29, C30 | 1 & 2 |
| **10K** | Chip Resistor | R0603 | R1, R7, R13, R14, R17, R20, R20, R22, R38, R40, R42, R43, R45, R46, R53 | 1 & 2 |
| **10K** | 25 Turn Trimpot | RTRIMT93XA | R26 | 1 |
| **10nF** | Ceramic Capacitor | C0603 | C3, C8, C12, C14 | 1 |
| **10pF** | Ceramic Capacitor | C0603 | C9 | 1 |
| **10pF** | Ceramic Capacitor | C0603 | C11 | 1 |
| **L-USL2012C** | Ferrite Bead | 603 | L1 | 1 |
| **1K** | Chip Resistor | R0603 | R16, R21, R23, R28, R31 | 1 |
| **1uF** | Ceramic Capacitor | C0603 | C13 | 1 |
| **2.2uF** | Ceramic Capacitor | C0603 | C7, C33 | 1 |
| **22K** | Chip Resistor | R0603 | R2, R3, R39 | 1 |
| **22uF** | Ceramic Capacitor | C0805 | C2, C15 | 1 |
| **MMUN2111LT1** | PNP Prebias Transistor | SC59-BEC | T2, T3 | 2 |
| **3.3K** | Chip Resistor | R0603 | R5, R9, R10, R18, R19, R8, R15 | 1 |
| **39K** | Chip Resistor | R0603 | R6, R12 | 1 |
| **4.7uF** | Ceramic Capacitor | C0603 | C6, C20, C22, C25, C26 | 1 & 2 |
| **446/405 (uP)** | JUMPER-SMT | SMT-JUMPER_3 | JP12 | 1 |
| **47K** | Chip Resistor | R0603 | R27 | 2 |
| **47uF** | Ceramic Capacitor | C1206 | C1, C4 | 1 |
| **680pF** | Ceramic Capacitor | C0603 | C10, C16 | 1 |
| **8MHz** | SMD Crystal | CRYSTAL-SMD-3.2X2.5MM | Y1 | 1 |
| **B5819W** | Schottky Diode | SOD-123 | D2, D3 | 2 |

# STM32-DVM-MASTR3 V1.0a-1

| Value | Device | Package | Parts | Sheet |
|---|---|---|---|---|
| **BLU** | SMD LED | LED-0603 | LED2 | 1 |
| **BSS138** | SMD MOSFET | SOT23 | Q2, Q3 | 2 |
| **EXT POWER** | +7-15V Input | 1X02 | JP2 | 2 |
| **GRN** | SMD LED | LED-0603 | LED5 | 1 |
| **I2C** | .1” Pin Header 1x3 | 1X03 | JP14 | 1 |
| **INHIBIT SOURCE** | .1” Pin Header 2x3 | 2X3 | JP8 | 2 |
| **X1** | 96 Pin Eurocard Connector | MABC96LP | J1 | 1 |
| **LED ENABLE** | .1” Pin Header 1x2 | 1X02 | JP1 | 1 |
| **MCP6002** | Operational Amplifier | SO08 | IC1, IC2, IC3 | 1 |
| **Solder Pad** | Solder Pad/Pin | MCS10B | +3.3V, +5V, GND, TP1, TP2, TP3, TP5, TP6, TP7 | 2 |
| **MODEM DISP** | .1” Pin Header 1x4 | 1X04 | JP4 | 2 |
| **MODEM SERIAL** | .1” Pin Header 1x5 | 1X05 | JP5 | 2 |
| **MP2307** | DC-DC Converter | MPS2307_DC-TO-DC | A1 | 2 |
| **MMUN2211LT1** | NPN Prebias Transistor | SC59-BEC | T11 | 2 |
| **MMUN2214LT1** | NPN Prebias Transistor | SC59-BEC | T1 | 1 |
| **NanoPi NEO** | ARM Core SBC | NANOPI-NEO | U2 | 1 |
| **ORG** | SMD LED | LED-0603 | LED3, LED4 | 1 |
| **PGM ASYNC EN** | .1" Pin Header 2x2 | 2X02 | JP7 | 2 |
| **RED** | SMD LED | LED-0603 | LED1, LED6 | 1 |
| **RX Audio Source** | .1” Pin Header 1x3 | 1X03 | JP11 | 2 |
| **SBC Display** | .1” Pin Header 1x4 | 1X04 | JP6 | 2 |
| **STM32F4xx ISCP** | .1” Pin Header 2x3 | 2X03 | JP3 | 2 |
| **STM32F4xx** | STM32F446/STM32F405 | QFP50P1200X1200X160-64N | U1 | 1 |
| **TCXO** | 12MHz TCXO | 5.2X3.4-4-PAD | OS1 | 1 |
| **XC6206P332MR** | 3.3V Linear Regulator | XC6206_SOT-23 | P1 | 2 |
| **YEL** | SMD LED | LED-0603 | LED7, LED8, LED9, LED10, LED11 | 1 |

* Parts C7 and C33 vary depending on the STM32F4xx microcontroller variant used. See the schematic diagram for installation details.