

The STM32-DVM-MTR2K is an integrated MMDVM modem and host platform on an easy-to-use plug-in-card for the MTR2000. This document describes how to set up your MTR2000 for use with this module. This document expects you to have some familiarity with Motorola RSS software.

The STM32-DVM-MTR2K should be plugged into the center slot in the front of the MTR2000 repeater. The bottom slot can also be used, but the center slot is preferred due to heat concerns. The top slot **MUST NOT** be used, as its connector is wired differently and is only for use with a wireline card.

As purchased, the board comes with a Micro SD Card pre-installed that has been already imaged with Pi-Star.

Connecting to the Pi-Star Dashboard:

Begin by connecting the NanoPi MMDVM to a router or other network facility capable of issuing DHCP IP addresses. Apply power and wait a few minutes for the NanoPi to fully boot. Look in the router's DHCP lease table (sometimes called "Device List" or similar) for the hostname "pi-star". Note the IP address associated with this hostname. Open a web browser on and type the IP address into the address bar.

If all goes well, you will be prompted for a login and password: pi-star / raspberry

This document will not cover configuration of Pi-Star's settings as most of the necessary information is available from online sources.

If you need to re-image the SD card with Pi-Star, here are the "necessary" items that need changed from default:

Change the following on the main configuration screen:

*Radio/Modem type: STM32-DVM / MMDVM\_HS – Raspberry Pi Hat (GPIO)*

Please change the following on the expert/MMDVMHost screen under "Modem":

*TXInvert: 0*

*RXInvert: 0*

*DMDelay: 162*

If you have an optional OLED display connected, here are the settings:

On the configuration screen:

*MMDVM Display Type: OLED Type 3*

*Port: /dev/ttyS2*

This board is heavily based on the Repeater-Builder STM32-DVM. In fact, most of the documentation is identical, including the calibration procedure and firmware updates.

One important difference is that the STM32-DVM-MTR2K contains no trim pots. All calibration is done with the softpots in the MTR2000. There are three things that normally need calibration: RX, TX and RSSI.

The RX and TX alignment are described in this document, as well as the wildcard programming necessary to run the DVM with analog revert utilizing the repeater's internal repeat audio path. The procedure for RSSI calibration can be found with online searches.

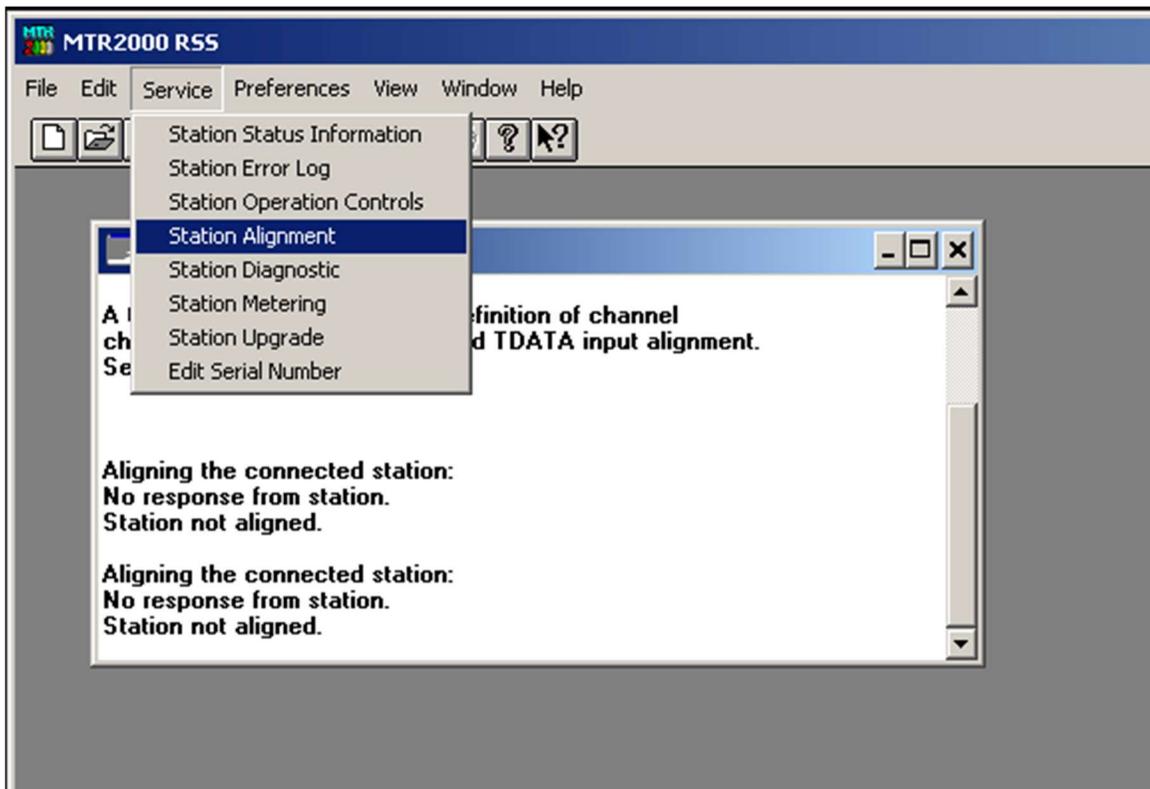
A note on RSSI:

Since the repeater's RSSI output is fixed, one only needs to run through creating a mapping file. As you do so, you may note that at higher RF levels, the MTR2000 will saturate the ADC input. Careful tuning will allow good numbers at higher RF levels, but also allow useable resolution down in the -120dB to -125dB range. This is simply trial and error.

Before diving into the station alignment procedure, be sure to **program the operating frequencies**. Changing the frequency of operation *can* change the alignment slightly.

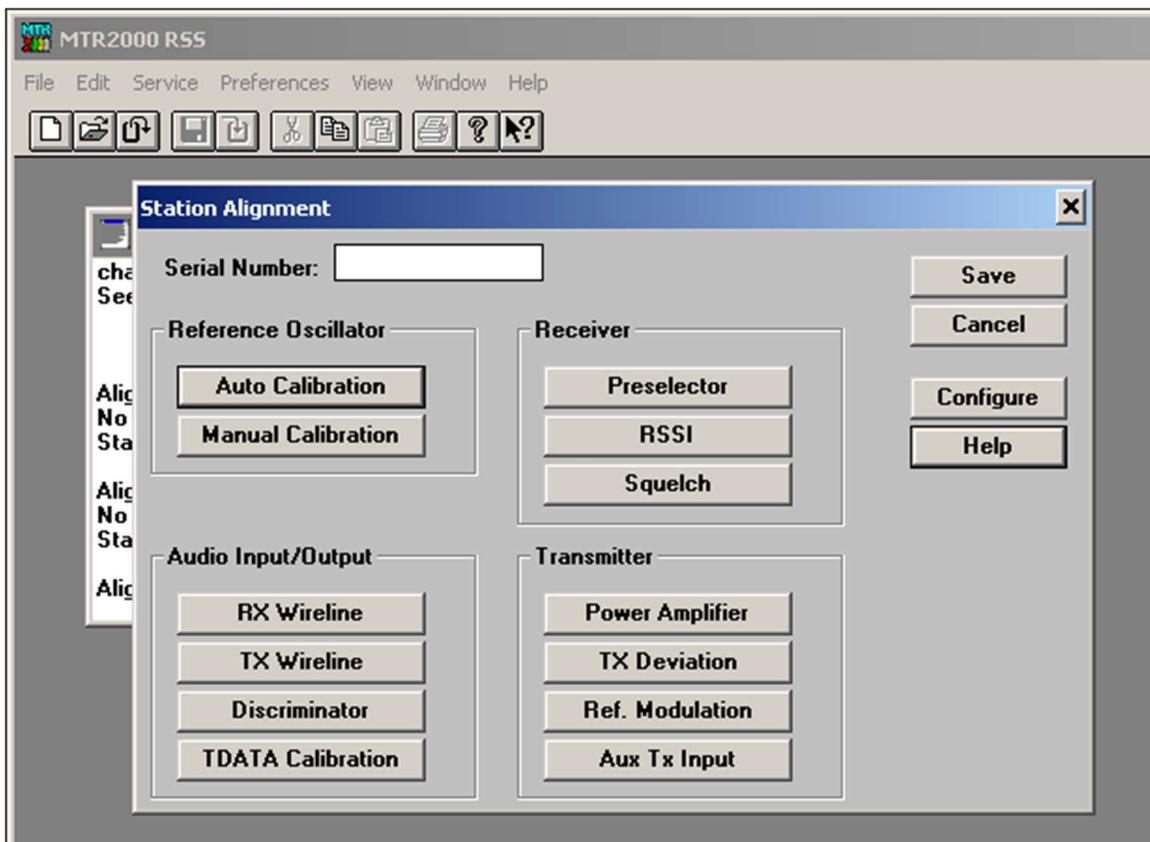
Let's get started. 😊

To set the TX and RX levels, navigate to the "Service" menu in the RSS and select "Station Alignment".



*Station Alignment from the Service Menu*

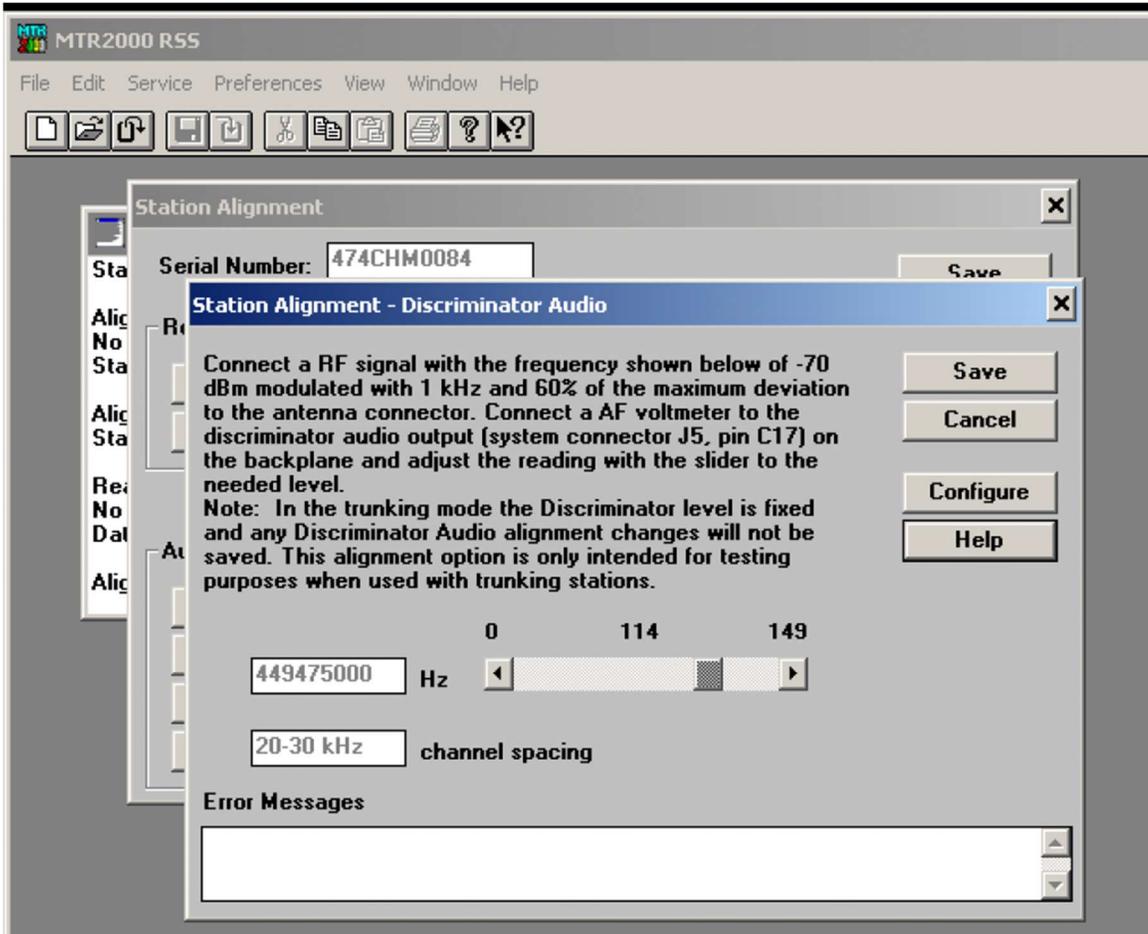
You'll have to wait for a few seconds while the RSS reads the alignment information from your station. The sections we'll be working with are the "Discriminator" button under "Audio Input/Output" and "Aux TX Input" under "Transmitter". The calibration procedure matches the procedure for the STM32-DVM.



Station Alignment Window

# RX Audio Adjustment

Since the STM32-DVM-MTR2K is optimized for the MTR2000, there is no need to have two RX audio adjustments. Follow the procedure for the generic Repeater-Builder board using the “clip” LED, only using the MTR2000 Discriminator output level adjustment instead of on-board pots. The sweet spot tends to be right around 110, so you might start there.



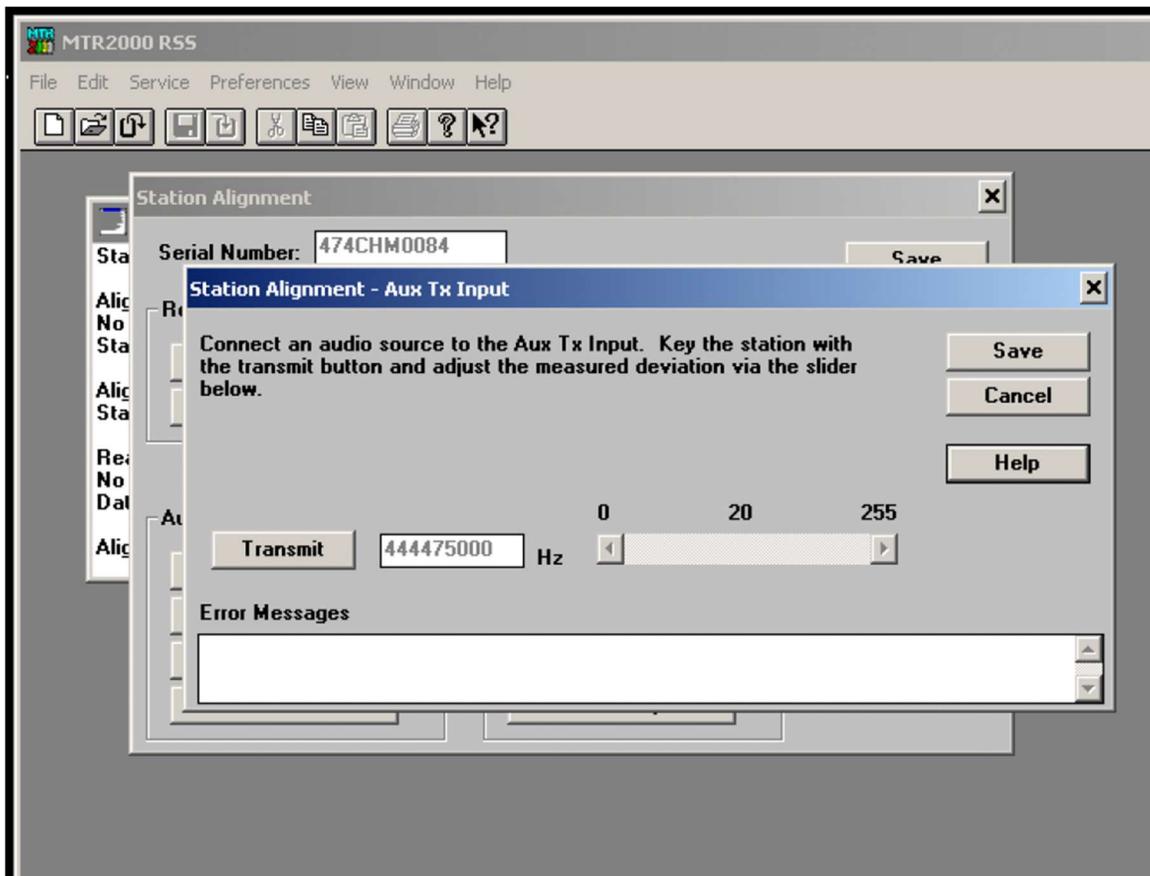
*Discriminator Audio Setting: Typically around 110*

# TX Audio Adjustment

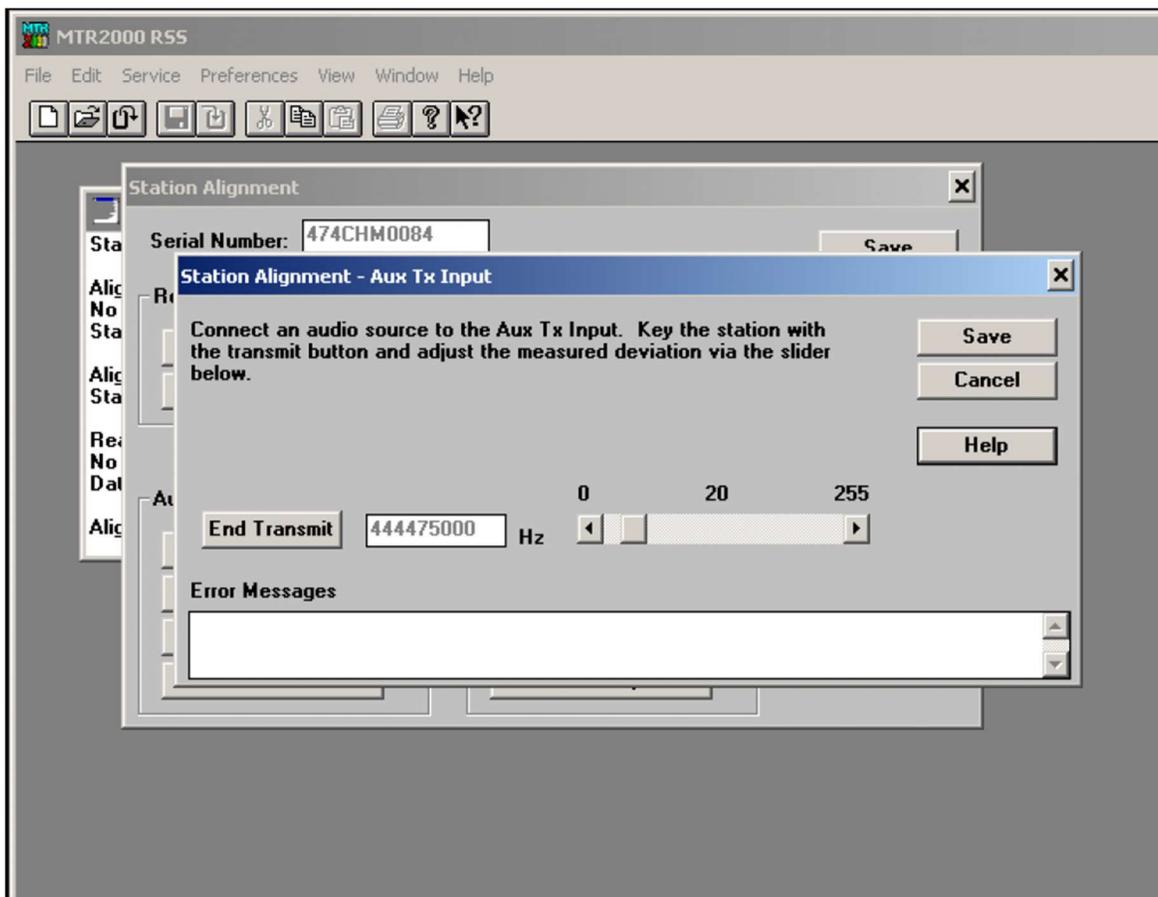
Transmit audio adjustment is pretty much the same as the receive and is done with the “Aux TX Input” adjustment. Use MMDVMCal and either set the deviation of the DMR test tone as described with the MMDVM documentation, or use a spectrum analyzer (25kHz sweep works pretty well) to set modulation as close to Bessel zero as possible. You will not achieve a perfect “null” using the Bessel zero technique as the soft pot’s steps are too coarse to do this. But if you note on a deviation meter how much change there is, between a perfect null and close to it, you’ll realize that the perfect null isn’t really necessary.

***Pi-Star users: MMDVMCal is accessible at the bash prompt as root: `pistar-mmdvmcal`***

Note: When adjusting Aux TX Input, the slider will not appear until you select the “Transmit” button. The slider appears when the station is keyed. The final result will be around 20-26, making that a good place to start.



*Aux Tx Input adjustment window*



*Aux Tx Input adjustment window with transmitter keyed (slider visible)*

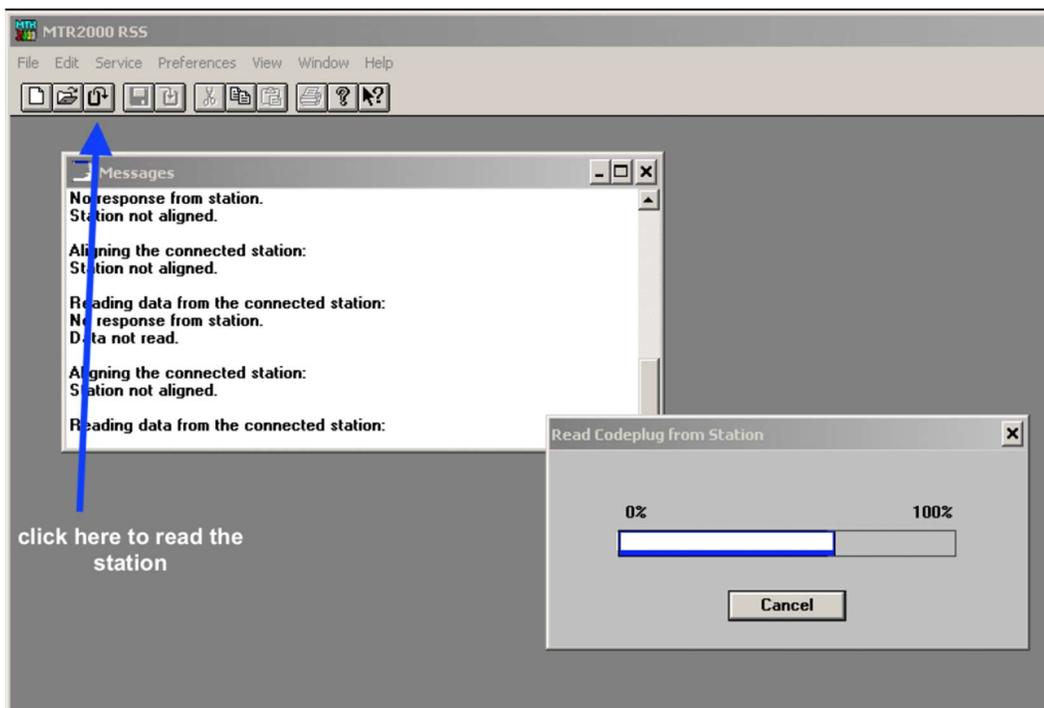
Make sure and use the “Save” buttons both on the individual alignment windows and then on the main station alignment window as well. Calibration values are not permanently written into the station until the main Station Alignment “Save” button is clicked. You know that you’ve actually written the station when you see it reboot:



*This means you’ve written the station’s memory*

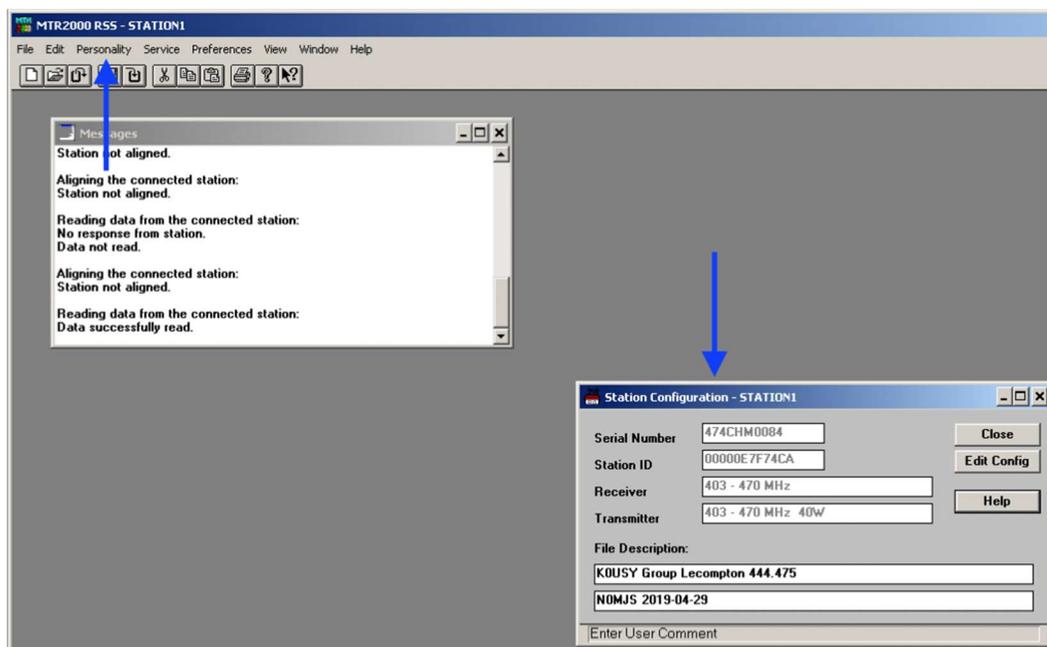
This is all that is required for calibrating an MTR2000 with the STM32-DVM-MTR2K.

There are a few more things that you'll need to do in the radio's channel settings. For this, you'll need to first read your radio's configuration (codeplug):



*Reading an MTR2000 codeplug into the RSS*

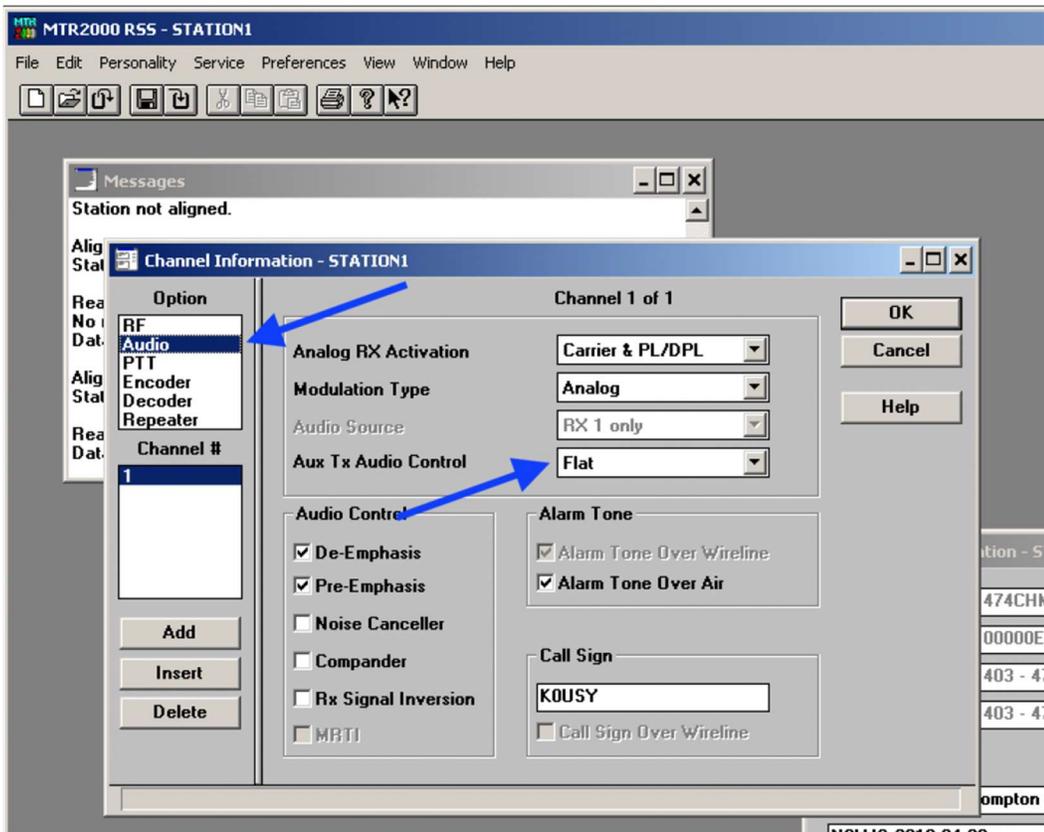
We need to ensure that, for the channel you will be using, two things are set: “Aux Tx Audio Control” and “External PTT Mapping”. Once the station is read, a new “Station Configuration” window will appear, and a new menu item called “Personality” will show up between “Edit” and “Service”. The personality menu is only present when a Station Configuration window is active.



*Station Configuration window after the station codeplug is read or loaded*

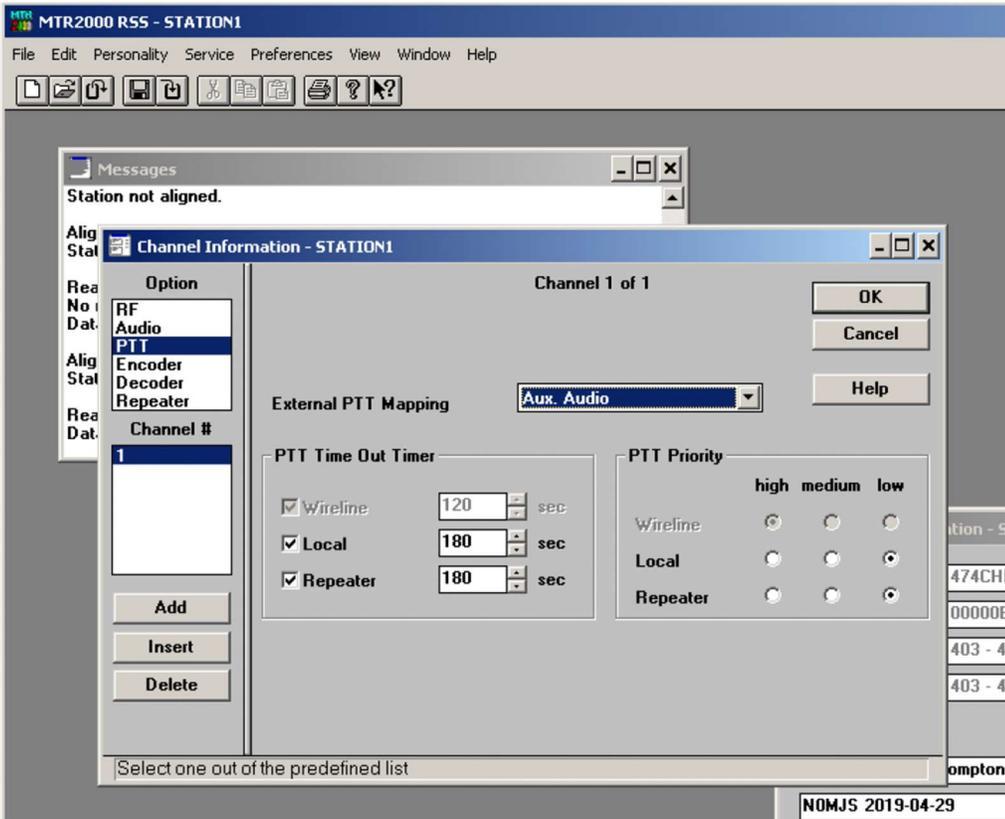
Under the “Personality” menu, select “Channel Information”. A new window will appear, in the upper left area of the window is a section box marked “Option”. From this list we'll be working

with the “Audio” and “PTT” options. In the Audio settings you will need to set “Aux Tx Audio Control” to the “Flat” selection from the drop-down list. This directs the station to NOT pre-emphasize or limit the Aux Tx Audio input.



MTR2000 RSS Channel Information screen

MTR2000 RSS Channel Information screen: Set the Aux Tx Audio Control to Flat  
Finally, you will need to set the External PTT Mapping to “Aux. Audio” under the PTT section of the Channel Information window. This directs the station to use the Aux Audio input (that we previously calibrated) as the transmit audio input when the “External PTT” is triggered.

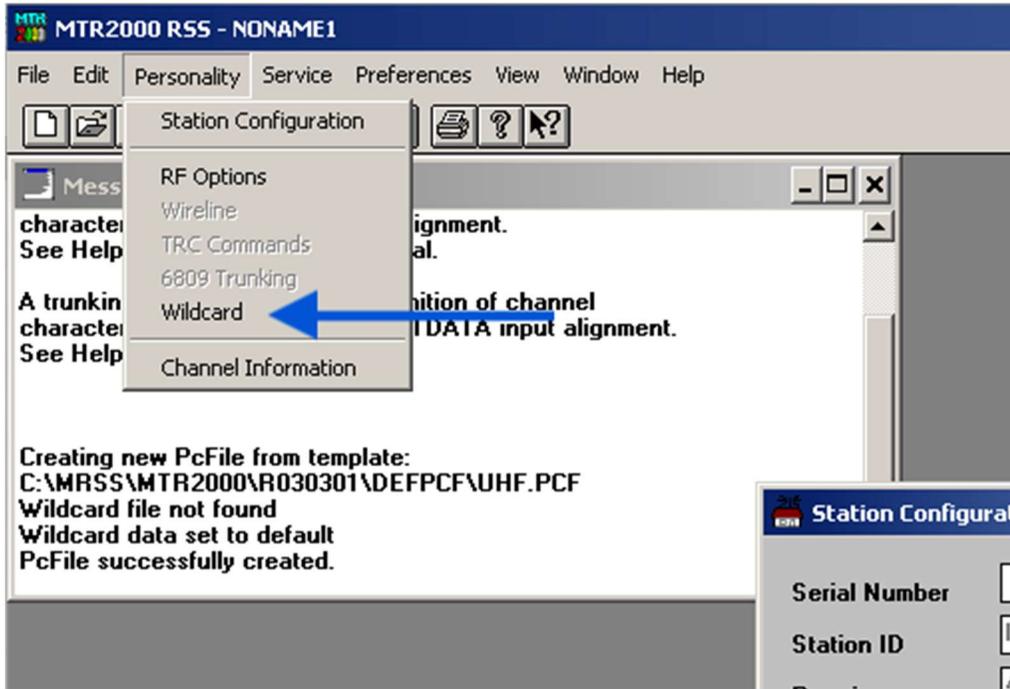


*MTR2000 RSS Channel Information screen*

MTR2000 RSS Channel Information screen: Set the External PTT Mapping to Aux. Audio. Make sure and click “ok” then write the station. None of the settings are stored until they are written back in, at which time the station will once again perform a reset. Now you are finished.

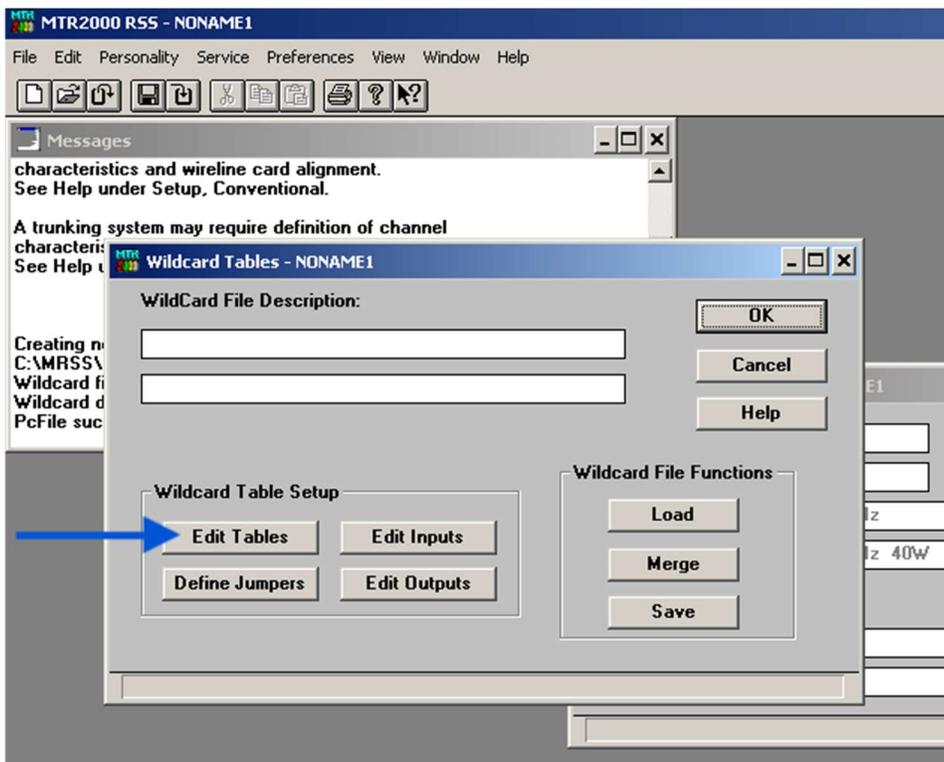
# Wildcard Programming

As long as the Station Configuration window is active, there will be a “Personality” menu between “Edit” and “Service”. Click the “Personality” drop-down menu and select “Wildcard” as shown below.



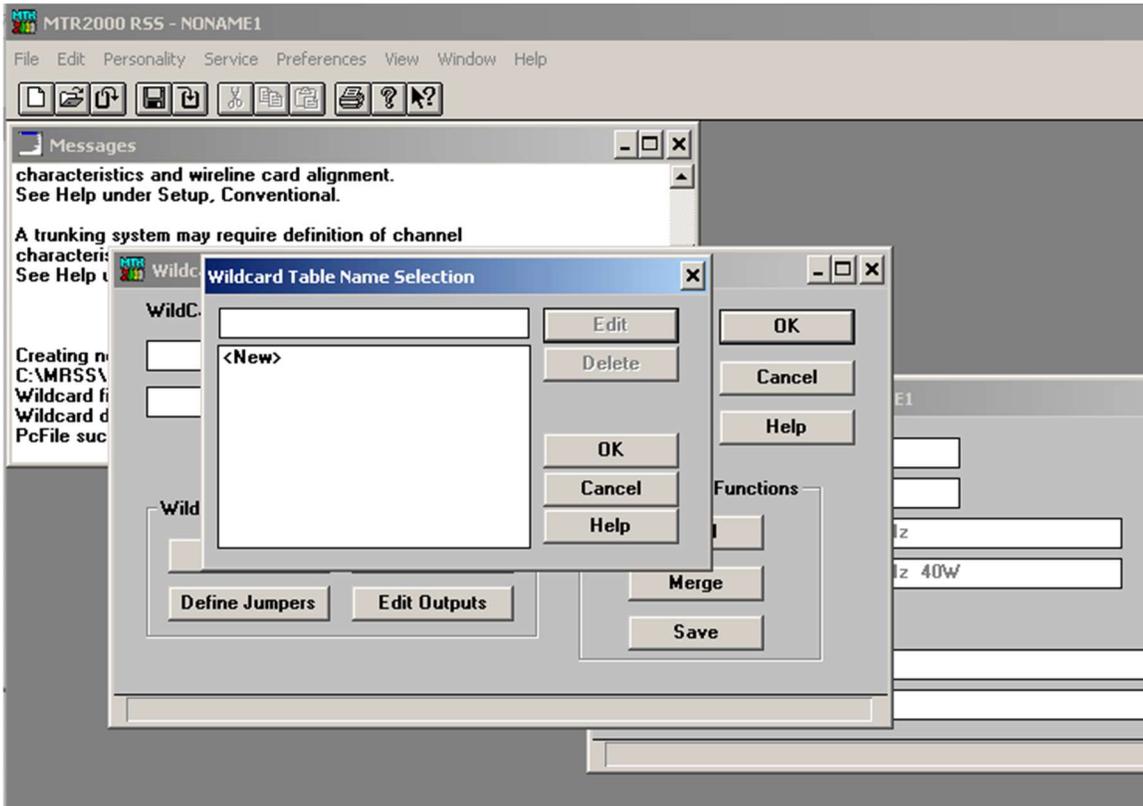
MTR2000 RSS: Wildcard programming selection

Once selected, a new window for Wildcard table programming will appear. Select edit tables.



MTR2000 RSS: Edit Wildcard tables

If the MTR already has existing Wildcard programming, there will be previously defined tables which need to be deleted. At this point the following window will appear:

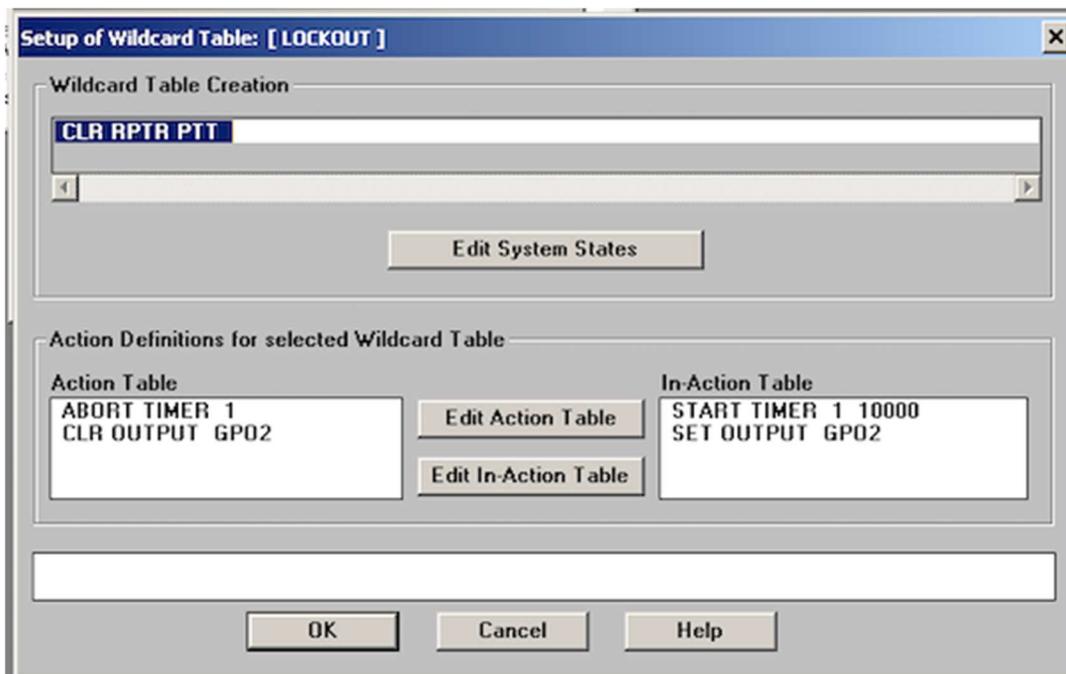


*MTR2000 RSS: Wildcard Table Name Selection*

This window is a bit confusing. Note that “Edit” is grayed out. I will first create two new Wildcard tables, typing the name in the single-line text box at the top of the window (above the window with <New> in it).

Table #1 named “LOCKOUT” uses GPO2 to lockout the MMDVM during analog operation, and table #2 named “KNOCKDOWN” that uses the PTT signal from MMDVM to knockdown/setup the analog repeat function of the MTR2000 station.

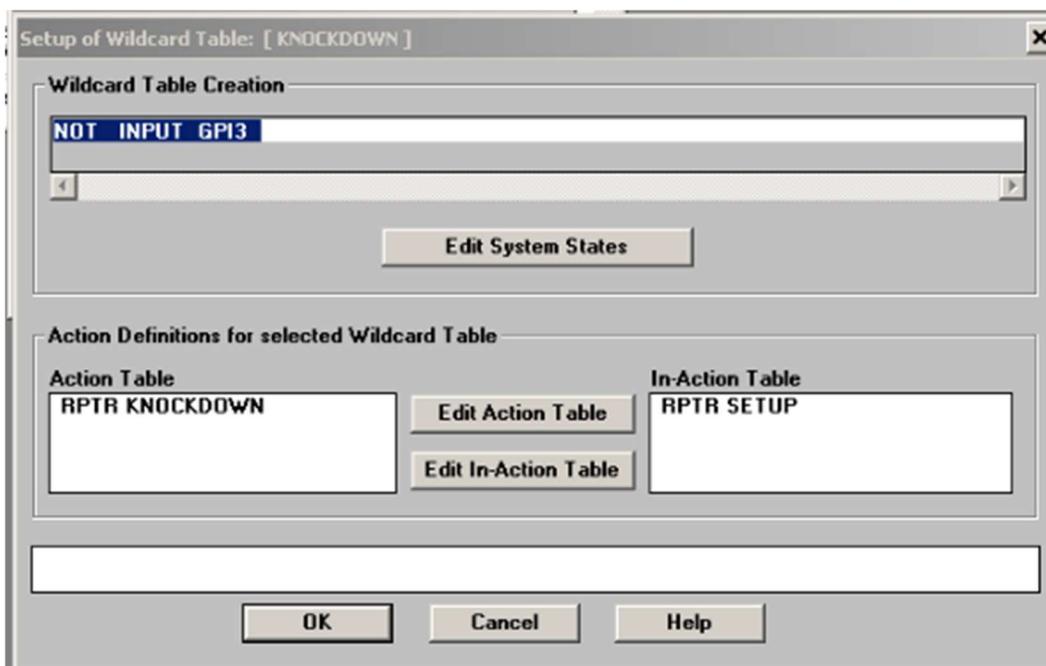
Wildcard programming is hard to describe in text, so I will show the screens for each and explain what is happening. In each Wildcard table, I will select a single “System State” and then describe a group of actions, in order, that will take place when the System State becomes true (Action Table) and a separate list for when the state becomes false (In-Action Table).



*MTR2000 RSS: Wildcard Table Programming – LOCKOUT*

For the LOCKOUT table, the system state I am setting “clearing” GPO2 (MMDVM lockout) when the in-chassis repeat PTT is active. Clearing it takes the output low, and since the lockout/COR input of the MMDVM is active low it locks out the MODEM. You will also notice a timer in the mix as well. If there were no timer, then as soon as an analog user unkeys, the repeater would again be available for MMDVM which would allow MMDVM to hijack the repeater in mid-QSO. Thus, I use a 10 second timer (10,000ms). This is an important part of action table programming: Items in any action table are executed sequentially and are fully blocking. That means that the in-action table item “SET OUTPUT GPIO2” does not happen until timer 1 is finished. This is also why the first action of the action table is “ABORT TIMER 1”. This ensures that timer 1 is aborted if there is a re-key in under 10s, otherwise, when timer 1 expired, GPO2 would be set high (unlocking the modem).

Ok, that’s a lot to take in. If you don’t want to reason it out, that’s cool, just duplicate what I put in above and you should be fine.



### *MTR2000 RSS: Wildcard Table Programming – KNOCKDOWN*

My KNOCKDOWN table is considerably simpler. I use a very long hangup time for MMDVM – on the order of 15s, so I don't really need a timer in addition to the MMDVM PTT. In this case, when "INPUT GPI3" is "NOT" active, the in-chassis repeat function is put in knockdown (Motorola terminology for 'disabled'). When the PTT goes back high (remember, it's active low) the repeat function is enabled, or in Motorola speak, "setup".

That's it! It's done. Just make sure and "OK" and/or "Save" out of each window when you close it, and make sure to remember to write the codeplug back into your radio so this changes are programmed in.

You will note that I did not get into any other general settings for the MTR2000. That's because there are a number of possible configurations that work – and there are already write-ups aplenty about which configuration is better than someone else's. Many are claiming that theirs is the only one that works. All I can say is – (as you might notice from what you see in my screen shots) that despite what you may read elsewhere, you can leave the station configured as a repeater and even engage the built-in analog control system. I do this to maintain the ability to troubleshoot the station, and as an emergency fall-back option. I also prefer using the CW ID built into the MTR2000 instead of MMDVM – but that's my personal preference.