Midland Titan 70-0671C - 6 Meter Conversion

Disclaimer !

The author of this paper is a hobbyist. He is not a professional radio service technician, nor is he affiliated with any company that sponsors such activities. It is in no way meant to represent or replace any formal training or technical service documents. Please use this information at your own risk for your own trials. Read through all of the hints and other sections before starting the process. This will help you avoid some pitfalls later.

Summary

This document will attempt to explain one way of re-purposing the 70-0671C low band VHF radio onto the adjacent 6 meter amateur radio band. It should be considered a starting point, and it will require further refinement to become more usable across a population of radios and users. Be advised that if you have seen other processes where a Hex editor can be used to modify channel frequencies, that will not apply to the Titan radios. They have much different EEPROM map requirements.

Tools Needed

- 70-1470 Titan320.exe programming software. This is available in the YG Midland LMR file section where you found this document.
- 70-1470 Programming Guide
- 70-0671 Titan Service Manual
- TITN6PRG.exe file. This was created and given to me by Terry, WA7GY. You can ask him for the latest version.
- 70-1309 programming cable or equivalent. I recommend the Greene County Communications MP-1, and it is readily available unlike the original.
- DOS capable PC. I recommend a native OS like Windows 98 on Pentium or older hardware.
- 30 Amp DC power supply for the radio
- DC Voltmeter
- 50 Ohm Dummy load rated for at least 150 Watts
- Phillips Screwdriver
- Flat blade tuning screwdriver (solid plastic preferred)

Process:

1. Acquire a 70-0671C Midland Titan radio. This is the 42-50MHz version of the radio. Other frequency variants of this model will not readily tune above 50MHz
2. Connect the power wiring, ignition sense power wire, external speaker (optional), and dummy load to the radio.

3. Connect the programmer to the radio mic jack, and apply power. Make sure the radio is not in scan mode. The radio will supply the power to the programming cable level shift circuitry.
4. Connect the programmer to the PC serial port (COM1) and start the Titan320.exe file from the DOS prompt.
5. Go to Utility, Radio COMM and click the *Up Load* button to get a copy of the radio loaded into memory.
6. Modify the radio settings for the channel plan you wish to use (Rx, Tx, Signaling tones, Wide band, Alpha Character, Lowpower, etc.). For channel frequencies between 52 and 54, I recommend storing them as 42 through 44 as a 10MHz offset.
7. Save your configuration as a descriptive file (i.e. - 6Meter.dmp).
8. Start TITN6PRG.exe and open your file (i.e. - 6Meter.dmp).
9. Edit your frequencies to your desired plan. If used the offset method, 42.575 would become 52.575.
10. Save your file with the frequency modifications.
11. Go back to the Titan320.exe program, and load the modified file into memory (i.e. - 6Meter.dmp). DO NOT attempt to view the channel data or the program will give continuous errors for incorrect frequencies.
12. Make sure the radio and programmer are powered and connected. Go to Utility, Radio COMM, and click the Down Load button to put the channel configuration into the radio. You now have a radio programmed with 6 meter frequencies.
13. Remove the programming cable, and attach the mic to the radio.
14. Power cycle the radio. If it starts on a channel above 50MHz, it will likely beep and show the E2 error on the display. This is because the VCO is out of lock.
15. Shut off the power, and flip the radio upside down. Remove the large bottom panel by removing the 6 screws.
16. Attach the DC voltmeter, negative to chassis ground and positive to VC. This test point is on the front side of the VCO cover, near the int/ext speaker jumper.
17. With the unit upside down, apply power and monitor the DC voltage at VC during Rx. Midland recommends 1.5VDC at this point on 42MHz. You will not get there at 52 MHz, but adjust L713 to the lowest voltage value. This is the tunable inductor under the VCO can closest to the front of the radio and VC.
18. With the dummy load attached, key the mic while monitoring VC during Tx. Adjust L702 for the lowest voltage at VC. This is the tunable inductor under the VCO can closest to the back of the radio.
19. That's it. Power down the radio, and re-attach the cover. You are now ready to try your radio on 6 meter amateur frequencies. Enjoy!

Issues

- The programmer may refuse to communicate with the radio. This is very frustrating. Even the programming guide acknowledges this by stating you should repeat the power cycle until comms are established. See hints for some suggestions to avoid this.

- The conversion application used was only under a beta release. It takes lots of time and effort to make an application function without any issues.
- There were some issues with the conversion process not retaining certain parameters. You may notice this as well.
- There may or may not be other nuances with the conversion process like not properly saving 5 kHz channel steps (like for simplex channels.)
- The VCO adjustment may not work properly if you immediately go to a 53 MHz channel. It might just stay out of lock where you don't see any change in VC during adjustment. (see Hints for an easy way around this.)
- Just because the VCOs lock, does not mean you have a radio that "hears" well or "talks" well. You may be pushing the Rx/Tx into the slope of the lowpass/bandpass filters. Expect some drop in performance.
- This is not a replacement for a Complete Realignment, which could enhance the performance to get more from the radio. This requires more equipment and knowledge than was available to me for documenting the process.
- The entire process is not robust since we are asking the radio to work outside of its designed parameters. It was not my goal to redesign the radio, so I accept the limitations.
- Your experience may vary.

Hints

- I found the best way around communication issues (as has my colleague) is to use the oldest PC hardware you can find. Run a DOS based OS and your reliability will go up tremendously. This is what the early radio software was designed to use.
- My radios were aligned for 47 MHz. ADD several frequencies to your channel plan from 42 to 50 MHz as test frequencies. Using 47MHz, I could see the VCO voltages that were set by the installer. This gave me a point of reference for setting higher frequencies for optimal voltages at VC. By checking frequencies above and below 47MHz, I could characterize the range on the VCOs. Armed with this knowledge, I could set the VCOs for the best possible VC before moving to an out of band channel.
- Improve the software conversion process. This can be achieved in two ways. The best way would be to modify the original Titan320 program to accept frequencies above 50, or rewrite the entire software programming package. The next best way would be to modify the conversion application to accept all input variables for modification. I believe Terry has something for doing this using an Excel spread sheet. Either way would be more work for a programmer, but a big help to everyone reprogramming these radios.

Background (two views of reverse engineering the EEPROM mapping)

The key to all of the reprogramming is to understand the mapping of the EEPROM data. There was no human readable channel frequency numbers that could be derived by examining the dmp file with a hex editor. Instead a study of the EEPROM and the correlation between parame ter changes and bit representations had to be learned. This is a tricky and time consuming process that involves some trial and error, as well as some investigative work. Several people helped to do this, for which I am extremely grateful. Two main reverse engineering processes emerged. The key to this was to find the relationship of frequency to bits and also to realize the checksum used to validate the changes to the radio software. They are similar but different. This is my view of how they came to be.

The Bit Manipulation Method

In this method the correlation of frequencies to the IF was found and a series of bit masks was applied. The key to this was mapping out a channel location byte sequence into 4 bit nibbles since certain data spills over from byte to byte including a 4 bit checksum which was needed in the channel data. This turns out to be 20 nibbles followed by 13 bytes per channel.

1. Convert decimal frequency to Midland binary RX frequency number (ex. 52.575 Mhz)
1a. Add IF frequency: 52.575 Mhz + 10.7 Mhz=63.275 Mhz
1b. Divide by channel step size: 63.275Mhz / 5000= 12655d or 316fh
1c. Mask with 3F: (316F & 3F)= 2F
*1d. Mask with FFFC0 and multiply by 2: (316f & FFFC0)*2=6280h*
1e. Add result from 1c: 6280+2f= 62AFh
62AF is the binary number used to change the RX frequency to 52.575 Mhz.

The TX frequency is the same...only start at 1b...do not add the IF frequency.
5213h is the binary number used to change the TX frequency to 52.575 Mhz.

The PLL Method

In this method it was discovered that the bit data in the channel bytes was actually derived from the MB1504 PLL IC used in the 70-0671C radios. The PLL uses a formula to set the frequencies, and this is what is read from the EEPROM. The software varies the **N** (11 bit programmable counter divide ratio) and **A** (7 bit swallow counter divide ratio)values to adjust the frequency where **P** (prescaler) = 64. N can vary from 64-511 and A can vary from 0-63.

$F = [(P X N) + A] X F_{ref} / R$

P = 64, N = Value, , A = Value, R = selected Freq. Resolution. Output frequency is set 10.7 MHz above desired to account for 1st LO. When A = 64 it is rolled over to 0 and N is incremented.

Example: Rx Frequency of 42.250 MHz is desired. PLL is set to 42.250 + 10.7.

F = [(64 x 166) + 1] x 0.005 = 53.125

53.125 – 10.7 = 42.425

Programmed Value = NA (0101001100000001) = 0x5301

> For the example a 5kHz step size is used. In this instance the Rx channel is set with the addition of the 10.7MHz IF added in. The Tx PLL settings would not use the 10.7 off set from the IF. This represents 2 bytes for Rx data and 2 bytes for Tx data. The rest of the channel nibble data would be the same in both methods.

I do have a descriptive copy of the entire EEPROM map that I believe is fairly accurate. We are refining it with every discovery, and there have been a lot of them. Please contact me if you would like a copy to review and further refine.

The End

Thank you for taking the time to read this. I hope it proves helpful to others. Good luck on your quest to keep the Midland radios alive during their post municipal service. We hope to make some new HAMs very happy by introducing them to new bands and the world of 6 meter. Cheers!