"Mots6" Amateur 900mhz band Modification for the Motorola MTX900 HT

Contact info: recurry@curry.org

The Motorola MTX900 handheld radio is designed to operate in the commercial 900 MHz band. It normally receives 930 MHz to 942 MHz and transmits 890 MHz to 902 MHz. This is the range of the commercial band. The transmitter and receiver are capable of operating in the Amateur Radio 900 MHz band (901 MHz to 928 MHz) but the firmware in the radio is incapable of accepting programming in that range. This is because all frequencies are stored in the radio EEPROM as offsets from 930 MHz. This was determined with great effort by disassembling and single stepping through the Motorola programming software.

There were several possible solutions explored to solve this problem:

- Change the synthesizer reference frequency
- Reprogram the on-board MTX900 processor ROM somehow
- Find a way to reprogram the synthesizer with amateur frequencies on the fly
- Several other less likely possibilities

After much experimentation, the only feasible solution was found to be finding some way of translating the synthesizer programming information from the on-board processor to amateur band programming information before it got to the synthesizer. This would allow the MTX900 to be programmed for standard commercial frequencies with standard Motorola programming software which would be decoded and translated “on-the-fly” to amateur frequencies and then programmed into the synthesizer.

The "mots6" modification accomplishes this by inserting another microprocessor in the serial link between the MTX900 on board control microprocessor and the on board synthesizer chip. This second microprocessor intercepts the serial data stream from the MTX900 processor which normally directly programs the MTX900 synthesizer. It decodes the commercial frequency that was sent, translates this frequency to amateur band frequencies, re-encodes them to the format the synthesizer will accept, and programs the MTX900 synthesizer with the new frequency. The microprocessor must re-program the MTX900 synthesizer in real time and in only a few milliseconds under the following conditions:

- When the radio is turned on
- When the channel switch or system knob is changed
- When the PTT is pushed (to program TX frequency)
- When the PTT is released (to re-program RX frequency)

Determining how to do this was fairly difficult since the synthesizer chip that Motorola uses in the MTX900 is not documented in any available document. Since no data on it's programming was available I had to decipher it. Two approaches where used. The first was to use a logic analyzer to get a high level view of what was being sent to the synthesizer - number of bits, timing, relationship to clock pins, etc. Once this was determined, a microcontroller was programmed to monitor these signals in real time, translate them to ASCII representations of one's and zero's, and send them via a serial link to a PC. The MTX900 was then programmed with many different frequencies and the data monitored as it was sent to the synthesizer from the MTX900 microprocessor. The results were noted in a big Excel spreadsheet. This process was followed until finally a pattern was determined. I worked on this off and on over several months. Code was then written and debugged for a small microcontroller to implement the required serial protocol and the frequency translations.

The microcontroller I used was the MicroChip 16c84. Due to the small form factor requirements of fitting inside the MTX900 case, I choose a prefabricated module called the "PicStic I" from Micromint Inc. This is a small "Basic Stamp" like PCB with the 16c84, a voltage regulator and a ceramic resonator on board. The board measure approximately 3/4" x 1.5" in size. The cost $30-$40.

With a pre-programmed "PicStic" in hand, modifying the MTX900 is fairly straightforward. It requires disassembly of the MTX900, the removal of 5 pins from a connector in the radio, the installation of the PicSticI by soldering 12 wires to the MTX900 PCB's, and the programming of the MTX900 with the right frequencies. The most tedious part is cutting, striping and tinning the 12 wires.

Installing the Modification

To install the modification you need a PicSticI programmed with the Mots6 code. Then you must add a small jumper from pin1 of the 16c84 to pin 5 of the module as shown in the photo below (black line). We need an extra I/O line for the modification to work and this is how we get it.

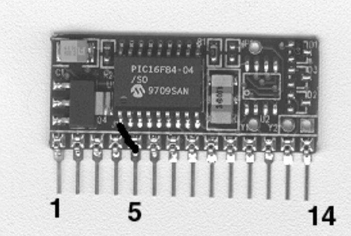


Photo of the component side of the PicSticI showing small jumper to be added

In addition to adding the jumper you must cut the trace leading to pin 5 of the module as show below.

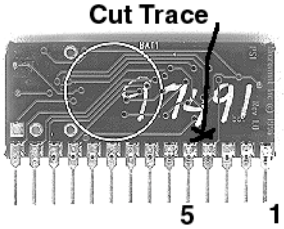


Photo of the backside of the PicSticI showing where to cut the trace leading to Pin 5.

Now, set the PicSticI aside. You must now make modifications to the MTX900. Now is the time to stop if you are unsure of yourself since the modifications we will be making to the MTX900 will be difficult to reverse.

Take a look at the photo labeled “NTN5234B and NTN5432B Controller PC board, Layer1”. After disassembling your MTX900, remove the screws and cables attached to this board and CAREFULLY remove it from the HT’s chassis. If you don’t understand how to do this STOP NOW, reassemble the unit and ask someone else to do the modification for you.

Once you have this removed you will be looking at a set of pins that connect J2 on the board you just removed to P2 on the board still attached to the chassis (reference the photos). You will need to remove several of the pins from P2 (the board still attached to the chassis. I’ve found that the best way to do this is to use a small piece of brass tubing just slight larger than the pin. Slip it over the pin and use it as a handle to wiggle the pin back and forth until it breaks off. If you want to stop now this is your last chance. Once you remove these pins the only way to go back is to solder wires from one board to the other. Remove pins 3,4,5,7,8

Once you have completed this task re-install the controller board (the one you removed). Connect all cables and put the screws back in it.

Now you are ready for wiring. The PicSticI will be taped up and positioned under the speaker and down towards the battery connector of the MTX900 on re-assembly. Take this into account for wire length. The best way to wire the PicSticI into the radio is to build a small harness. You will need a single in-line socket with 14 pins. These are available from Radio Shack in longer lengths which you can cut. Alternately, you could solder right to the PicSticI but I don’t recommend it.

You must use very small wire. Wire wrap wire or small ribbon cable cut to length, stripped and tinned will work. I prefer ribbon cable because wire wrap wire breaks easily. Use your judgement and a very small soldering pencil tip, the connections are small. Wire all the wires going to J2 and J1-4 first then turn the MTX900 over and wire P2. Use the photos for reference and wire as follows:

PicSticI Pin	MTX900 Pin
1 VCC	J2-10 +8v
2 GND	J2-1 GND
3 RA3	J2-4 SSYI
4 RA4	P2-4 SSYO
5 RA2	J1-4
6 /RST	No Connection
7 RB0	J2-3 SROI
8 RB1	J2-8 SDGI
9 RB2	J2-7 SDAI
10 RB3	J2-5 SCLI
11 RB4	P2-3 SROO
12 RB5	P2-8 SDGO
13 RB6	P2-7 SDAO
14 RB7	P2-5 SCLO

Double check all the wiring for proper connection and no shorts or solder bridges. You should be able to re-assemble the unit back in the case. Very carefully slide the case over the PicSticI. You may have to gently wiggle things around a bit to get it to fit. Re-attach the batter connector. Probably a good idea to have taped up the PicSticI prior to doing this. One layer of tape is sufficient.

Assuming you wired it right and have a PicSticI with the Mots6 firmware, attach a battery and turn the radio on. It should beep. If not press the PTT, it should beep. If not change to a different system and try again. If the radio is dead, you did something wrong. Go back and check everything again.

Once the Mots6 modification is properly installed, the MTX900 can be programmed using standard Motorola MTX900 RSS programming software. To enable amateur band frequencies commercial frequencies are programmed using the following tables:

; **RX translations**

;

	Commercial Band Freq. programmed		**Resulting Amateur Band Freq.**
;	940.0000 - 940.9875	==>	927.0000 - 927.9875 for CA band plan RX or simplex RX
;	937.0000 - 939.9875	==>	918.0000 - 920.9875 for ARRL band plan RX or simplex RX

Any frequency not listed in the "Commercial Band Freq." column will not be translated and results in commercial RX/TX frequencies which is illegal for amateurs to transmit on. Remember to select talk-around if you want simplex.

The Motorola MTX900 software automatically calculates the TX frequency. For reference here is what the commercial frequencies translate to:

; **TX freq translations**

;

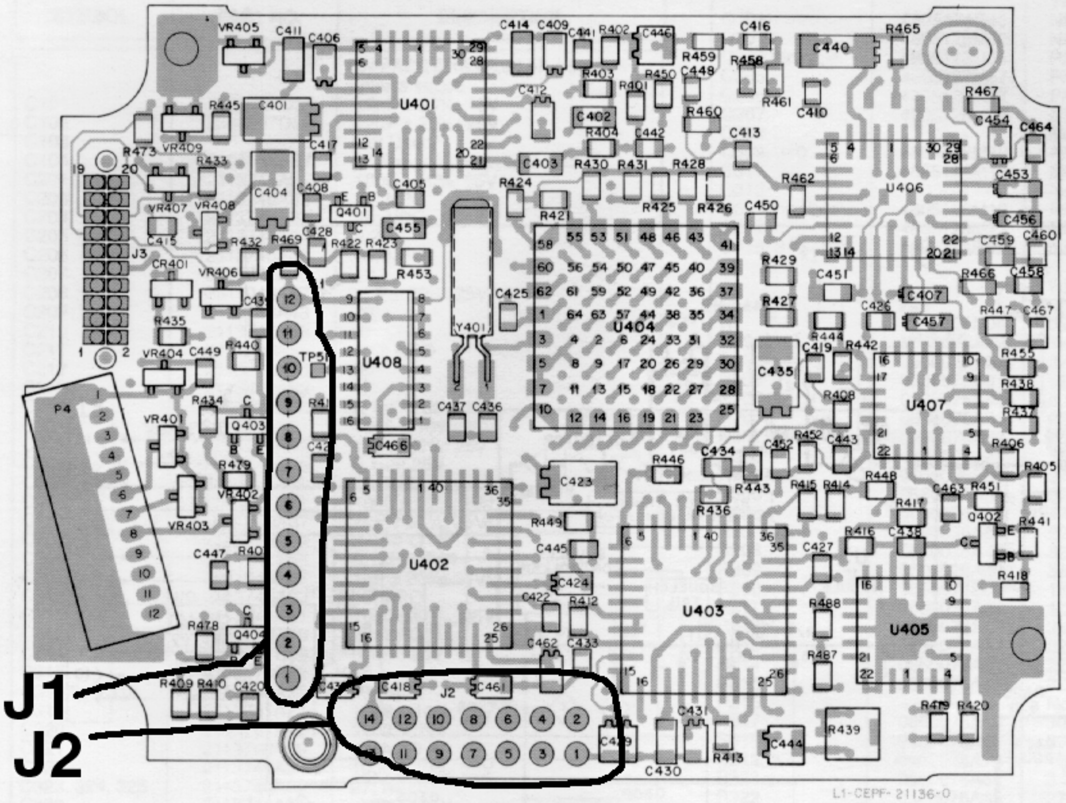
	Commercial Band Freq. programmed		**Resulting Amateur Band Freq.**
;	901.0000 - 901.9875	==>	902.0000 - 902.9875 for TX -25mhz offset for CA band plan
;	898.0000 - 900.9875	==>	906.0000 - 908.9875 for TX -12mhz offset for ARRL band plan
;	940.0000 - 940.9875	==>	927.0000 - 927.9875 for simplex TX
;	937.0000 - 939.9875	==>	918.0000 - 920.9875 for simplex TX

;

Program in your desired frequencies and you should be on your way. You might want to go into the MRSS calibration mode to adjust power output, frequency, etc.

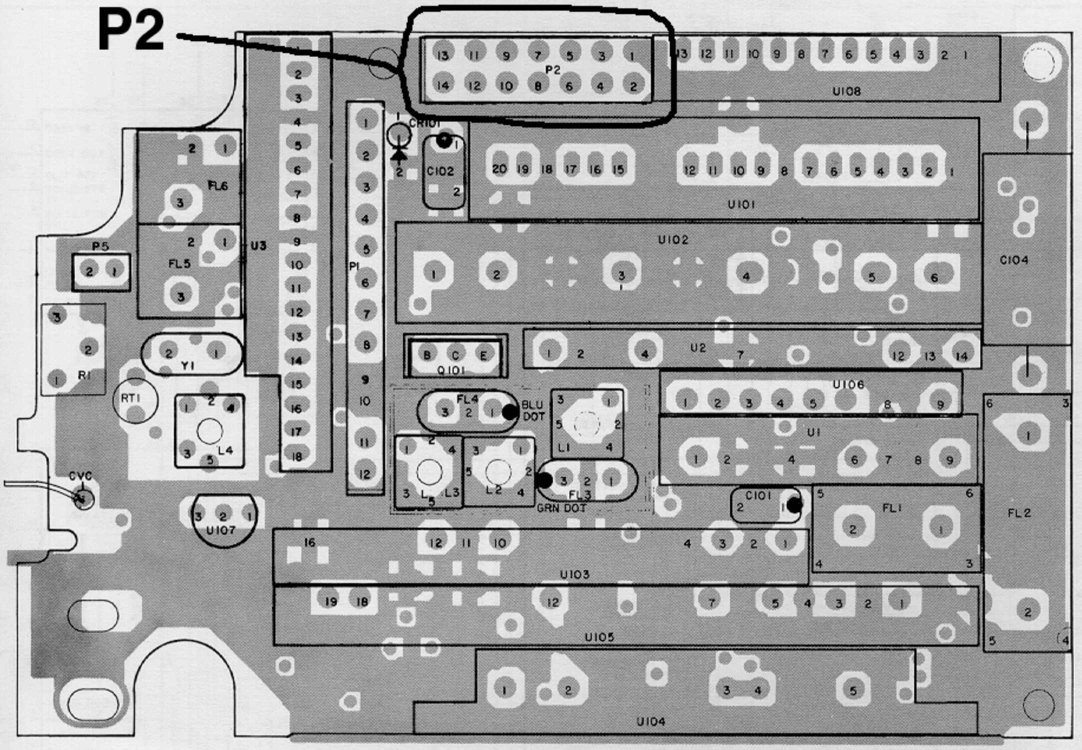
Have fun and good luck!

NTN5234B AND NTN5432B CONTROLLER PC BOARD, LAYER 1 WITH COMPONENT OVERLAY VIEWED FROM TOP (LEADLESS COMPONENT) SIDE



L1-CEPF-21136-0
OL-CEPF-21060-0

NUF6150C RF PC BOARD, LAYER 1 WITH LEADED COMPONENT OVERLAY VIEWED FROM BOTTOM (LEADLESS COMPONENT) SIDE



L1-CEPF-20749-0
OL-CEPF-20750-A