FT-990 Troubleshooting Hints - Ed K1MMI 12/5/2014

NOTE: This is a first draft and may have some errors.

Over the years I have had a number of problems with my 990 and used a number of techniques to have success at repairing my 990.
Hopefully, some of this information may be of help should you have a failure with your 990.

Data Collection

Whenever I have had a failure my first thought was what was I doing when the failure occurred? Did I do something to cause the failure or was it just a random failure? If I had been transmitting on a second transceiver in my shack there is a possibility RF got into my FT-990 and damaged the front end. Then days later when I turned on my 990 I discovered my 990 had a receive problem. If the 990 was my only radio and I turned it on after not using it for a few days and it had a problem then it was just a random error.

The second step is to collect as much information as possible about what things work and what things don't work on the 990.

Some basic things I do:
Does the S-Meter Light?
Is a frequency displayed and does the frequency change when I move the tuning dial?
I check to see if it receives on all bands or just some or none. Does the S-Meter work? This would check all the band buttons in the band matrix of keys. When you hit a band button the antenna tuner will move motors and you can hear the motors if the tuner is working.
I check to see if it transmits on all bands or just some or none. I also check the antenna tuner. Does the Meter work on PO, SWR, VCC, and IC?
I check the LSB, USB, CW, AM, FM, RTTY, PKT buttons. It they work an LED will go on. Success - indicates the ROM program detected a button was pressed and responded with doing whatever was necessary to activate that function. (e.g, pressing LSB would setup the 990 for LSB mode and light the LSB LED.)
I check all of the other buttons and controls. (e.g, when the RF gain control is moved the S-Meter should deflect between S0 and +60dB. The AF Gain control should increase and decrease the receive audio volume.)
Verify all voltages are normal.

Once I have collected all of the information about the current state of the 990 and whatever I was doing when the failure occurred, it helps me to identify the different problems and to try to pick the problem that seems like the simplist one for me to troubleshoot first. The assumption is there is a good chance there is only one failure and fixing that will get the 990 repaired. If I have no success at fixing the simplist problem, I then try to troubleshoot what seems like another fairly simple problem to troubleshoot. Eventually, by troubleshooting different problems - more information is learned and I find something wrong and that gets my 990 repaired.

Brute Force

When your first discover a problem and have no idea where to start, you basically do the brute force testing. Pretty much visual examination for burned components and checking all the voltages.

Unplug and plug back in the 5 plug in boards. Some group members say it has fixed their problems.

Check all power supply voltages:
+13.8, +12, -12, +5, R12 and T12

In Receive Mode R12 should be 11V and 0V in Transmit Mode. (If R12 doesn't go to 0V in Transmit Mode, the PO, SWR, IC, and VCC meter positions will not work.)
In Receive Mode T12 should be 0V and 11V in Transmit Mode.

Note: If the ROM Control program fails R12 and T12 will be 0V because they didn't get initialized by the ROM Control Program. So receive and transmit would be dead. If R12 and/or T12 have incorrect voltages that could indicate a bad Q5003.

If something is working on the 990 I would expect:
To see the frequency displayed and changing when you move the main tuning dial. The ROM Control Program initializes the frequency display early on in the ROM Control Program startup sequence.

To see the VCC meter position show 13.8 V in transmit mode.

Notes:
All input and output signals to the 5 plug-in boards can be checked with a voltmeter or a scope via the bottom of the Motherboard.
Without extender cards you can tack solder 30 guage wirewrap wire to circuit points on a plug-in board and extend the wire(s) vertically above the board when it is plugged in to measure the voltage or look at the signals with a scope. It's a bit slow tack soldering wires to troubleshoot a 990 plug-in board and can be error prone but it opens the door to fix a 990 without extender cards. It's also possible to unplug a board and use an ohmmeter to test diodes and some components. It is slow to unsolder some components - so they can be tested out of circuit - but it opens the door to find an inductor or component that is bad or has changed value. Some components can't be measured in circuit. There is also visual examination to locate burned components or possible cold solder joints. I have also used the eraser on a pencil to apply pressure to pc boards to find intermittent problems caused by a bad solder joint or cracked etch.

About half of the 990 is on separate boards and the other half is the 5 plug-in boards, so extender boards can be quite helpful but for some problems extender boards are useless.
You definitely need extender boards if you want to re-align a 990 but if you only want to vary a pot on a pc board, you can take the board out and move the pot position similar to a clock (eg; 7,8,9,10,11,12,1,2,3,4,5,6). Then check the effect to arrive at the best pot setting - It's slow but works.

For serious debugging of a 990 problem you have to walk your way thru the 990 via the Block Diagram, Interconnection Diagram, and the Motherboard diagram to figure out how the signals move thru the receive and transmit paths. This helps to figure out what circuits to check with wirewrap wire or via the Motherboard connectors.

ROM Control Program

Note: Yaesu doesn't provide any information on how the Rom Control Program works. It's a matter of studying the datasheets for the chips on the Control Unit board and paying attention to what happens when you press a button on the Front Panel to get a pretty good idea of exactly what the Rom Control Program does.

Here's a basic idea of the things the ROM Control Program software/firmware does. There would be a lot more steps than what is listed. It's guess work as to the exact order of steps but here's my basic picture:

1. It configures and initializes the Q5004 Microcontroller. This sets up a lot of signals as input and output pins. It configures the RS-232 port that is inside the Microcontroller and many other things.

2. It configures and initializes Q5014, Q5015, Q5016 and sets up a lot of additional signals as input and output pins.

3. It configures and initializes PLL, DDS(car), DDS(PLL), Tuner, and Display units.

4. It sets up all of the default values for bands and what frequency and modes it would be on when power is applied for the first time or after a reset. When the 990 has been in use all of the current bands, modes, and state of the 990 at power down would be saved in the Microcontroller RAM memory. This way when the 990 is turned on again the ROM Control Program will setup up the 990 to be in the same state as it was when the power had been shut down. It gets this information from the RAM memory area that is inside the Microcontroller.

5. Q5003 is setup so R12 is turned on. This sets R12 to 11V. R12 provides the DC power for all circuits that are only active in receive mode.

6. Q5003 is setup so T12 is turned off. This sets T12 to 0V. T12 provides the DC power for all circuits that are only active in transmit mode.

7. The ROM Control Program checks the band buttons to see if the band buttons are pressed with a "Power Up Selection". If GEN and ENT are pressed it will reset to the default memory values. If 1 and 3 yellow band buttons are pressed it will perform the "Las Vegas" diagnostic test and display the ROM version number on the frequency display.

8. If the ROM Control Program detects an error during the Initialization and Configuration process it may quit and it will not detect a button pressed by the operator. In some cases you might see the frequency display flashing or you may get no indication of what went wrong. If the ROM Control Program fails to startup it's a bit of a brute force approach to try and discover what has failed.

9. If the ROM Control Program is successful and thinks the 990 is in an operational state then it will have three main tasks to do:

(1) Detect the operator has pressed a button; figure out what button has been pressed; then do whatever is necessary to do the function the operator wants.
Just about all of the buttons on the front panel are in a matrix. This matrix is shown in the Keypad Unit diagram and the Display Unit diagram. The matrix consists of rows and columns. In each row up to 4 buttons are defined. For example, on the Display Unit Diagram **ROW S6** detects the CLAR TX, CLAR CLR, VFO/WR, and A<->B buttons; and on the Key Unit Diagram **ROW S0** detects the 0,1,2,and 3 buttons.

(2) Detect there is activity on the CAT port; read 5 bytes to determine the CAT command; then do whatever is necessary do the function requested via the command.

(3) When the operator moves the main tuning dial to detect the frequency change and do whatever is necessary to get the frequency changed and the display updated with the new current frequency.

One way to think of the ROM Control Program is it is the traffic cop of the 990. It's the boss and communicates with the other modules/boards in the 990 to get some job/task/function done. It does it by sending information/commands to the other modules/boards via Output Pins. It also monitors/reads Input Pins to get a response or to check the status from the modules/boards that were told to do something. All of the Input and Output Pins are located on connectors J5001 and J5002 on the Control Unit board.

When the operator wants to do something he/she presses a button and the ROM Control Program detects a button has been pressed and does its thing to get that function done for the operator.

The general sequence:
Operator presses a button.
ROM Control Program cycles thru the different rows until it finds a button pressed in a column in a row.
It does whatever is necessary to do the function associated with the button that is pressed. It sets an ouput signal via Q5004, Q5014, Q5015, Q5016, Q5018, Q5019, Q5020 to setup the 990 for that function. If an LED has to be turned on it sends a request to the display unit to light the LED. The output signal is sent to the appropriate plug-in module via connectors J5001 and J5002 on the Control Unit board.

Some examples:
Operator presses LSB button.
ROM Control Program figures out what key is pressed in the matrix.

a) J5001 OUT (Key0,Key1,Key2,Key3) are output keys that select a row in the matrix.
b) J5001 IN (Key0,Key1,Key2,Key3) are input keys that read or examine the buttons in the matrix for the current row selected to see if a button in that row is pressed.
c) it goes to a) and the next row in the matrix is selected. This process is repeated until the ROM Control Program finds the LSB button has been pressed.

It sends a request to the display unit to light the LSB LED indicator and also communicates with the appropriate module to get the 990 setup for LSB mode.

Operator presses 14 Mhz button; ROM Control Program detects 14 Mhz button is pressed and sets J5002 - BPF0,BPF1,BPF2, BPF3 output pins to select 14 Mhz on the RF Unit and sets J5002 - A,B,C,D output pins to select 14 Mhz on the LPF Unit.

General Comments

If you have a friend who has a 990 it sure would be nice to do some board swapping to determine if the problem is on a plug-in board but there is a risk you could end up with two 990's that don't work - so I view that approach as a last resort. In my experience with my 990 I have found every failure involved a bad cable, a failure with a standard component (e.g, diode, transistor, inductor, resistor, capacitor or integrated circuit), or a cold solder joint. I have had very good success at getting standard components from Yaesu in California. The 990 has a lot of unique assemblies, such as the Band button unit, the frequency display and the 6 Microprocessors.

A few people totally disassemble a 990 into 20 to 30 pieces and sell them off on places like eBay or other websites. If they could sell every item they would get well over $2000 for a disassembled 990. It's very doubtful Yaesu would have a unique assembly after all these years but it's worth contacting Yaesu in California and the UK to see if they have it. If a Microprocessor fails you probably need to buy a used board on eBay.

The ROM Controller Program, be it version 1.2 or 1.3 is contained in the Microcontroller Chip. Yaesu refers to it as the Rom Control Program but it's really a Microcontroller chip that has an internal ROM area. This is very different than an FT-1000. The FT-1000 has a separate EPROM which can easily be re-programmed. If the Microcontroller fails the 990 becomes a brick. My understanding is Yaesu in California has not had any Microcontrollers with version 1.3 for about 10 years. They may have a few Microcontrollers with version 1.2 in stock. Yaesu in the UK had 3 Microcontrollers with version 1.3 but Group members purchased all of them in recent years. It would appear if there are any version 1.3 Microcontrollers in existence they are in some bench at an old Ham Radio repair shop. Once in a great while I'll see a Control Unit plug-in board on eBay with version 1.3 but they are not cheap and are usually gone in a hurry.

When the front end gets overloaded it can destroy a bunch of diodes wiping out the receiver on several bands. It can cook some surface mount resistors. D1056 usually gets destroyed resulting in 0 watts output. D1063 can get destroyed killing the receive. Very early production units may have D1057 that can get destroyed killing the receiver.
(D1057 was replaced with a capacitor on later units and L1084 was moved.)

Q9003 or Q9004 sometimes have a cold solder joint. This results in no +5V or +12V voltages.

Several people have had problems with the R12 and/or T12 voltages. Q5003 is a driver chip that has failed for at least 5 group members.