Reverse Engineering The Internal Memory Map of the Yaesu FT-991A

...or, taking apart the FT-991A without using a screwdriver

Gil Kloepfer KI5BPK

What is this about?

- Quick intro to the FT-991A (for those not familiar with it)
- Memory vs. Memory - We'll be talking about the memory CHIP inside the radio - NOT so much about the memory CHANNELS
- Why would we want to read (or write) the radio's internal memory?
- How can we do it?
- What were/are the challenges?
- How does this apply to other radios?
- Cautions … many of them

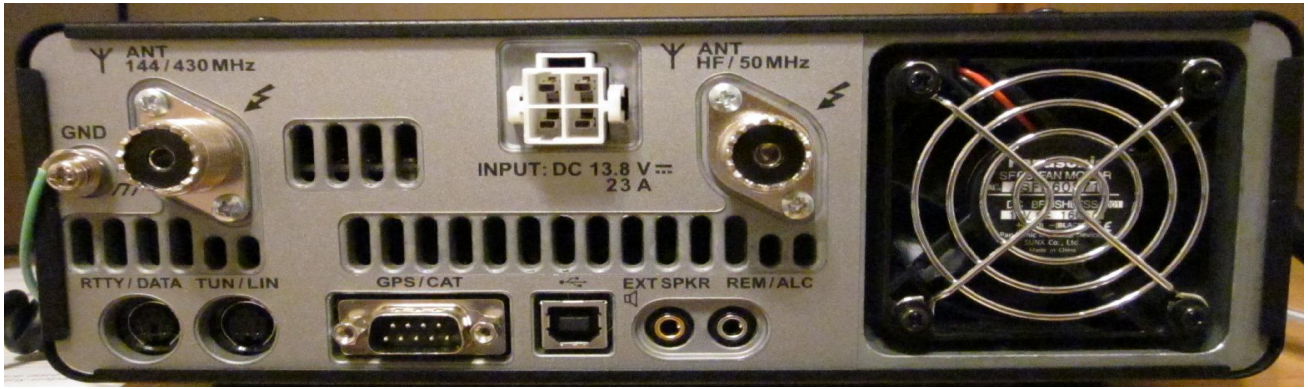
Yaesu's FT-991A - A Very Quick Summary

- It's a Shack-In-A-Box radio - Can operate 160m through 70cm excluding 1.25m
- CW, AM, SSB, FM, C4FM (WIRES-X access)
- 100W max transmit power (HF/6m); 50W max transmit power (2m/70cm)
- Uses a combination of a LCD front-panel screen and a variety of knobs and buttons for control, similar to the Icom IC-7300

Yaesu’s FT-991A - A Very Quick Summary

- Has a waterfall “scope” display / configurable from 50KHz to 1 MHz wide
- Good CW support - But unfortunately it can’t decode CW (even Bill can’t teach it!)
- Unlike IC-7300, it’s a hybrid software-controlled radio, rather than a SDR. DSPs used for some filters, audio processing/shaping, C4FM digital, scope/pan adapter. Still lots of analog stuff in the 991A!

Yaesu's FT-991A - A Very Quick Summary



- Built-in USB sound card and USB/Serial interface (no TNC needed for Winlink, RTTY, FT8, other digital modes)
- Two USB "virtual" serial interfaces defined - one for CAT control, one for simple PTT control. Both available at the same time!
- Separate antenna inputs: VHF/UHF and HF
- Separate traditional RS-232 serial CAT (subset) or GPS input for WIRES

How did I start this project??

- Wanted to automate (under Linux) some functions of radio; wanted to write my own Linux-based configuration back-up
- Specifically wanted to control and back-up the per-channel skip setting, but it was not in the CAT specification for the radio (many aren't, unfortunately)
- Asked people on the FT-991[A] groups.io lists… (Thanks, Bennett!) Nobody there knew how either...
- Internet: Chirp project had similar issues - noted that RT Systems' ADMS Windows software could access/modify information not in CAT specification

BEFORE SOMEONE HERE ASKS...

YES, DARNIT - I DID CONTACT YAESU!

To Yaesu Tech Support:
When using the CAT interface, what command can be used to erase a memory channel or set/clear the memory skip? This command does not appear to be listed in the CAT command manual.

Yaesu's Response: (a little over a week later)
For CAT control of your radio we do have a manual on our website.
I have attached it below.
I do not know if this will meet your need, but it is the only CAT information we have for this radio.

How did I start this project?? (continued)

- Someone on the Chirp project with ADMS captured the serial transfer between the computer and radio and found they were using undocumented CAT commands (SPR/SPW) but he could not figure out the syntax...so….
- I spent an afternoon with the posted serial capture and figured out how the undocumented SPR and SPW CAT commands (read and write non-volatile memory) worked and documented them.
- It was not something anyone could have figured out “by accident.”
- Q: So how did RT Systems know about it to use in ADMS????
- Q: How can this new found knowledge be used? Can I set/reset/read the channel skip setting? (Spoiler alert: Yes, with caveats)

Read Memory CAT Command Syntax

Send:
S P R <addr-high> <addr-low> <req-check-byte> ;
<addr-high> and <addr-low> are the high/low bytes of the 16-bit memory address (0 through 32767)
<req-check-byte> = (<addr-high> + <addr-low> + Hex F5 (magic)) AND Hex FF

Successful response:
S P R <addr-high> <addr-low> <content> <nxt-content> <checksum> ;
<content> is the contents of memory at the address, and <nxt-content> is the contents of the next address.
<checksum> = (<req-check-byte> + <content> + <nxt-content>) AND Hex FF

What was in there?

- Dumped the entire contents of non-volatile memory (takes about 1 min 15 sec at 38,400 baud CAT rate using efficient read strategy - about 437 bytes/sec)
- Used a bunch of small programs written in perl to format the memory dump and looked for patterns.
- With some help from the FT-991A Technical Supplement (“Service Manual”) to access the factory calibration menu (DANGER WILL ROBINSON!) found a list of all (and more of) the calibration parameters in the non-volatile memory.
- Also found a list of all the memory channels, temporary memory channels, VFO settings, band stacking memories, scattered in various places

More of what was in there...

- Through trial-and-error (change parameter, see what changed in memory) was able to find nearly all of the user setup parameters (these are all available through documented CAT commands, though).
- The unique WIRES ID of the radio…
- FINALLY -- Many of the parameters stored for each of the stored memory channels, **including the channel skip bit!!**
- Everything, especially the user setup parameters, is arranged in a completely chaotic manner.

Snippet of what I started with...

```
             0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  |   ASCII TEXT   |
12848 3230: 00 00 80 03 0E 00 00 00 00 FF 7F 00 00 00 00 00  |................|
12864 3240: 00 00 80 03 0E 00 00 00 00 05 00 80 90 00 00 01  |................|
12880 3250: 00 03 00 04 40 00 1E 00 96 06 0D 00 00 05 40 00  |....@.........@.|
12896 3260: 1E 00 96 09 0F 00 00 04 40 00 1E 00 96 05 09 00  |........@.......|
12912 3270: 00 04 40 00 1E 00 96 05 09 00 00 04 40 00 1E 00  |..@.........@...|
12928 3280: 96 00 00 00 00 01 40 00 1E 00 96 00 00 00 00 1B  |......@.........|
12944 3290: 00 00 00 00 00 08 BD 8C 80 00 00 00 00 4E 35 54  |.............N5T|
12960 32A0: 54 2F 47 54 20 20 20 20 20 05 80 00 88 00 02 01  |T/GT     .......|
12976 32B0: 00 03 00 04 40 00 1E 00 96 06 0D 00 00 05 40 00  |....@.........@.|
12992 32C0: 1E 00 96 09 0F 00 00 04 40 00 1E 00 96 05 09 00  |........@.......|
13008 32D0: 00 04 40 00 1E 00 96 05 09 00 00 04 40 00 1E 00  |..@.........@...|
13024 32E0: 96 00 00 00 00 01 40 00 1E 00 96 00 00 00 00 1B  |......@.........|
13040 32F0: 00 00 00 00 00 08 AB 64 10 00 00 00 00 4E 35 54  |.......d.....N5T|
13056 3300: 54 2F 54 41 59 20 20 20 20 05 00 00 90 00 02 01  |T/TAY    .......|
```

After a lot of work...

```
Byte            Description
0   00    Mode (00=LSB 01=USB 02=CW-USB 04=AM 05=FM 0B=C4FM)
1   01    [b7] Skip (1=yes/0=no)
          [b0] Locked(0=editable/1=locked)
2   02    [b7] (unknown) Originally 0, 1 after edit RPT
          [b3-b2]=RPT (repeater offset): -=0 SIMP=1 +=2
          [b1]=NB-On
3   03    [b7] 5/10Hz[0=5Hz;*1=10Hz],
          [b5-b3] Tone/DCS (0=OFF, 1=ENC, 2=CTCSS 3=D.ENC 4=DCS)
          [b0] ATT-On
4   04    [b6] NAR/WIDE HF[*W 2400=1;N 1500=0]
          [b4] NAR/WIDE UHF/VHF [*W 16k=0; N 9k=1]
5   05    [b1-b0] IPO[IPO=0,AMP1=2,AMP2=3]
6   06    [b4] DNR-On
          [b3-b0] DNR Level (1-15) default=1
(and so on…)
```

More examples of notes...

```
Address   LenB10  Description
0000-0009    10   Unknown Hex: 00 00 A5 5A A5 5A 00 70 A5 00  (DO NOT CHANGE!)
000A-00D1   200   Calibration/Service Parameters (WARNING: DO NOT CHANGE!)
00D2-0124    83   (Unknown constants - LIKELY CALIBRATION - DO NOT CHANGE!)
0125          1   [b7-b5] Mem=2,MT=3,VFO-A=4
0126          1   ?? [b7-b6] Clarifier: Off=0, RX=3
                  [b1-b0] Current: 0=HF, 1=50MHz, 2=VHF, 3=UHF
0127          1   [b7] FAST: On=1 Off=0; [b5] LOCK: On=1 Off=0
0129          1   [b6] Keyer:*Off=0, On=1
                  [b1] Menu 084 (ARS 144MHz) 0=OFF, *1=ON  (saved @ menu exit)
                  [b0] Menu 085 (ARS 430MHz) 0=OFF, *1=ON  (saved @ menu exit)
012A          1   [b7-b5] M-LIST=4; BAND=2; MODE=1; Waterfall=0
012B          1   Menu number (-1) selected with MULTI knob (recall with MENU)
012C          1   Menu display line selected w/MULTI (0=top, 1=middle, 2=bottom)
```

Even more examples of notes...

```
Address   LenB10  Description
012F           1   ?? [b3-b0] Band/Bandstack band ID (see note 5)
013C           1   MONI level (default=Dec:50)
013D           1   (key) PITCH level (Hz)  300+(val)*10, default=Dec:30, 0->75
013E-0147     10   Callsign (10 characters, padded with ASCII blank)
0148           1   [b7] (unknown) Occasionally toggles, not sure why
014A           1   Multi control  Hex:RF_PWR=01, MIC GAIN=02,
                                  MONI=17, CH.DIAL=1F,
                                  SPEED=20, SQL=27, APF=21, PITCH=22,
                                  MCH=35, TONE=39, DCS=3A
                      (seen also on 014E instead, why?)
                       idea: 01A7: lower nybble may be offset to this…
                              may also explain power being in different places
0158           1   Menu 017 (CONTEST NUMBER) - Updated only when unit powered
                      off, will change if operating CW in contest mode
```

Output from one of my test applications:

```
**********************************************************************************
MEM 01          FM        146.64 MHz  >N5TT/GT        <   (Addr: 0x3249)
+--------------+--------------+--------------+--------------+
|      Skip    |     Locked   |      RPT     |       NB     |
|       No     |    Editable  |       -      |      Off     |
+==============+==============+==============+==============+
|     5/10Hz   |    Tone/DCS  |      TONE    |      DCS     |
|      10Hz    |     CTCSS    |    162.2Hz   |      023     |
+==============+==============+==============+==============+
|      ATT     |  NAR/WIDE[HF] | NAR/WIDE[U/V] |      IPO     |
|      Off     |     N 1500   |     W 16k    |      IPO     |
+==============+==============+==============+==============+
|      DNR     |     NOTCH    |              |              |
|      Off     |      Off     |              |              |
+--------------+--------------+--------------+--------------+
```

Some lessons learned...

- Why should you care?? Radios are becoming more and more software, and if we are to make them last as long as the old “boatanchors,” we need to understand as much as possible about how our radios **REALLY WORK**.
- I’m glad I found my radio’s calibration parameters - they can be saved in case the non-volatile memory chip ever goes bad
- **DO NOT WRITE TO ANYTHING YOU DON’T UNDERSTAND**! You could easily mess-up (“brick”) your radio - possibly requiring service or unknowingly transmitting on unauthorized frequencies!
- From the service manual, found that the non-volatile storage is CMOS FRAM (Ferroelectric Random Access Memory) - can be written at least 10^10 times!

Some more lessons learned...

- Sadly, companies like Yaesu and Icom are “hiding” and limiting capabilities of the radios, and only allowing companies like RT Systems access to the information. What happens if ADMS doesn’t work on the next version of Windows, and/or RT Systems stops supporting the FT-991A?
- While the FT-991A is a very cool radio, there are some hardware and software demons lurking… I have been able to make the radio behave weird with some perfectly legitimate and documented CAT commands.
- If you can use one of the documented CAT commands to accomplish what you need, use that rather than accessing the configuration memory (it is closer to real-time control, less likely to change, easier to parse)

Conclusions and Questions

As Callum (“DX Commander”) M0MCX says: **“Enjoy your radio!”**

These slides and current notes will be available at:
http://www.kloepfer.org/ft991a/

Please be patient for updates as I want to be careful to release information that is correct to the best of my knowledge. Contact **ki5bpk@arrl.net** if you are interested in exploring this further with me.

Any questions???