NJAD VFO - CAT Control Manual
Version 1.0
John Price – WA2FZW

Table of Contents

Introduction

This document describes the addition of CAT (Computer Aided Transceiver) control to the ESP32 based VFO that I have been working on along with Glenn (VK3PE) and Jim (G3ZQC).

The project is a heavily modified version of the VFO originally created by T.J. Uebo (JF3HZB).

The CAT control language used will be that used by the Yaesu FT-891. Why that language instead of the more commonly used Kenwood language variations? There are a couple of reasons:

- I have an FT-891 and am thoroughly familiar with the CAT control capabilities.
- The primary CAT control program that I use is *DxCommander* and when I first got the FT-891, I discovered that there were a number of issues when using the two together. Some of these issues were misunderstandings by the *DxCommander* author, and others were due to bugs in the Yaesu software. Working with Dave (AA6YQ, the author of *DxCommander*), we either fixed all of the issues or invented work-arounds. I found similar issues in other CAT controllers as well.

- The message handler software was 90% written, as I’ve used it in two other projects, so no need to reinvent the wheel.

CAT Implementation

The actual message handling is done using my *FT-891 CAT Control Emulator Library* which I developed along with the VFO project. The documentation for the library is also available on GitHub.

Although the CAT implementation uses the Yaesu FT-891 language, only a small sub-set of the language is implemented. Some functions are not implemented simply because the radios that we plan on using this with can’t possibly support them; for example, software adjustable filters.

Others are omitted simply because they are; we don’t provide the capability to save and transmit voice or CW messages, nor is a built-in keyer implemented (although I already have code for that, and will add it in the future).

There are also some limitations related to band switching and mode switching capabilities when using CAT control if the VFO is configured to read a physical band and/or mode selector switch.

In the code, there are three options for how the physical band and mode switching options can be implemented which are controlled by symbolic definitions:

• NOT_AVAIL	No physical switch is implemented
• GPIO_EXPNDR	The switch is read using a PCF8574(A)
• CAT_CONTROL	The option is only available via CAT control

The *NOT_AVAIL* and *CAT_CONTROL* options are essentially identical.

If the *CAT_CONTROL* (or *NOT_AVAIL*) option is selected, the band and mode can *ONLY* be changed via CAT commands. Why? Some radios (e.g. Glenn’s FT-7) rely on the physical band switch to change things in the radio. Obviously, a CAT command can’t move the switch (without adding a whole lot of hardware)! The same applies to the mode switch.

When using manual switches for either the band or mode switch, the status of those switches will be available to the CAT control library on the PC along with the transmit/receive status.

Messages Recognized

Not all of the FT-891 commands and status requests are implemented. The focus of this project has been to use the VFO to upgrade legacy radios, so many of the commands just can’t be implemented as they could be in a modern software controlled radio.

The commands/status requests that are recognized (in alphabetical order) by the library are:

	AB	Copy VFO-A to VFO-B
*	AI	(Data 0 or 1) Turn auto-information on or off
	BA	Copy VFO-B to VFO-A
*	BS	Band select (not implemented)
*	EX	Menu command (does nothing but needs to be here)
	FA	Set or request VFO-A frequency
	FB	Set or request VFO-B frequency
*	ID	Request radio's ID (0650 for the FT-891)
	IF	Information request/answer
*	IS	Set or request IF shift
	MD	Set or request mode (USB, LSB, CW, etc.)
*	NA	Set or request narrow IF shift
	OI	Opposite Band Information request/answer
	RIC	Alternate request for split status
*	RM	Read meter
*	SH	Set or request IF bandwidth
*	SM	Read S-meter
	ST	(Data 0 - 1) Split mode off or on
	SV	Swap VFOs
	TX	Set or request transmit/receive status

The commands marked with an asterisk are recognized by the library, but not used in the VFO program. The ‘*IF*’ and ‘*OI*’ Are not directly used by the VFO program. These 2 messages combine the status of a number of parameters into a single message. The two components of those messages that are of interest to the VFO are the frequency and operating mode.

Those messages cannot actually be implemented on the legacy radios that Glenn, Jim and I have been playing with, such as the ones related to filter settings, however, *DxCommander* can try to set these parameters and it requests their status in its routine polling sequence, thus they have to be handled although they do nothing.

The descriptions of some of the messages are a bit misleading, but the *Yaesu FT891 CAT Manual* (click on the “Files” tab) will clarify what they actually do.

Application to Library Interface

The VFO program can change many of the parameters through function calls to the CAT control library. For example, when one moves the encoder to change the VFO-A frequency, the VFO program calls *CAT.SetFA()* to change the VFO-A value stored in the CAT module.

The VFO program routinely checks the status of things that might be changed as a result of a received command. For example, it calls the *CAT.GetFA()* function to see if the VFO-A frequency has been changed via a CAT command.

There are similar get and set methods available for mode, VFO-B frequency, split status, etc.

Besides the Get and Set functions available to the application, there are a few other functions in the library that need to be used:

FT891_CAT — The constructor function which creates the CAT control module object.

begin — This function initializes a number of variables used for processing CAT messages.

There is an optional argument that can be used for debugging. Normally FT-891 messages are terminated only by a semicolon with no newlines.

If one wants to debug using the serial monitor of the Arduino IDE, adding a newline to messages makes things more readable, thus if the call to the *begin* function is written as:

```
CAT.begin ( true );
```

newlines will be added to the output. With this enabled, however, the program will *NOT* work correctly with *DxCommander* or other CAT controllers (the newline confuses them).

CheckCAT

This function is part of the VFO program itself. It is called every time through the loop function, but only actually checks for incoming commands every *CAT_READ_TIME* milliseconds (set in *config.h*).

If a new command has been processed by the library, *CheckCAT* looks to see in anything the VFO program needs to know about changed and processes the data accordingly.

Suggestion Box

I welcome any suggestions for further improvements. Please feel free to email me at WA2FZW@ARRL.net.