NJAD VFO – Operator's Manual
Version 1.1
John Price – WA2FZW

Table of Contents

Introduction 2

Calibration 3

The VFO Program 3

- Startup Conditions 3
- The Display 4
- Mode Indicator 5
- Clarifier Status 5
- Split Mode Indicator 6
- The Dial 6
- Numerical Frequencies 6
- Battery Monitor 7

Frequency Management 8

- Digital VFOs versus Analog VFOs 8
- Changing the Frequency Change Increment 8
- The Clarifier 9
- VFO-A and VFO-B Frequencies 10
- Frequency Manipulation 11
- The Function Button 11
- Setting the Transmit Frequency in Split Mode 11

Manual Control versus CAT Control 12

Changing Modes and Increments 12

Suggestion Box 13

Introduction

Hopefully, you've read through all the technical manuals and successfully built, programmed and installed the VFO in your radio. Now it's time to walk you some things that will help you understand its operation better.

It's not just another digital VFO. This project is a heavily modified version of the VFO originally created by T.J. Uebo (JF3HZB). I've been working closely with Glenn (VK3PE) and Jim (G3ZQC) on this project for several months now and all have contributed ideas on how to make it work much better than the original version.

Modifications made so far from the operational perspective include:

- Multiband capability using a physical band switch or CAT control.
- Operating mode (USB, LSB, etc.) switching either via a physical switch or CAT control.
- Allow use of either ST7735 based TFT displays or ILI9341 based TFT displays and allow a display size of up to 240x320 pixels. Both displays now use a standard library (*TFT eSPI*), which can work with a number of other displays as well however we've only tested with the IL9431 and ST7735 displays.
- Enabled the use of a high-speed optical encoder for the tuning function.
- Addition of a clarifier.
- Added CAT control.
- Added 2nd VFO and all the stuff that goes with that capability like split frequency (but only in the same band, at least for now).
- Added an optional "function button" which allows the dual VFOs to be manipulated manually (as opposed to only via CAT control).
- Added an optional battery monitor function for those that chose to run this from a battery as might be the case in a QRP radio.
- Added an optional frequency increment selection button that allows the operator to cycle through tuning increments of 10Hz, 100Hz or 1KHz.

Calibration

Before using the VFO, the frequency must be calibrated. The crystal frequency of the Si5351 is nominally 25MHz (27MHz on some), but all of the Si5351 modules we've tested have been off by a little bit (some more than others).

To perform this task, we've provided a calibration program separate from the VFO program named *Calibrate_Si5351*. This program will run on the same ESP32 that you are using in the VFO. Its user interface is the Arduino IDE's Serial Monitor.

The *Calibrate Si5351* program and the associated documentation are available on GitHub.

If you don't perform the calibration procedure, your transmit and receive frequencies will not be accurate.

The VFO Program

Startup Conditions

At startup, if no physical band switch is installed, the program will start on the default frequency for the first band defined in the *bandData* array and the increment and operating mode (if no mode switch is installed) will be set based on the values in that entry.

If a physical band switch is installed, the startup frequency, mode (if no mode switch is installed) and increment will be as defined in the entry for the band selected by the switch.

If a physical mode switch is installed, the startup mode will be that selected by the switch.

If you performed the calibration procedure described above, the VFO program will read and use the Si5351 calibration factor you saved in the EEPROM. If you didn't perform the calibration, the calibration factor will be set to zero and you will probably be way off your desired operating frequency.

If you have the serial monitor turned on when you first load the software onto the ESP32, the program will display some messages (self-explanatory) regarding whether or not the EEPROM initialized correctly and whether or not a valid correction factor was found. You really should enable the Arduino IDE's serial monitor the first time you run the program.

The Display

As distributed, everything that can be displayed on the screen is turned on except the battery monitor regardless of which display size is in use. There are symbol definitions in the *config.h* (See the *NJDA VFO Hacker's Guide*) file that allow you to enable or disable the display of some items.

Here are a couple of examples of what the displays look like; the standard 128x160 display is on the left and the standard 240x320 display is on the right:

Here's an example of how the appearance of the display can be modified by simply changing parameters in the *config.h* file. This is one of the versions of the small screen that Glenn is considering using in his FT-7 for his project:

The following are brief descriptions of what everything is and how it all works:

Mode Indicator

This indicates the current operating mode. It can be changed either by the operation of a physical switch (if installed) or via CAT control.

Clarifier Status

When the clarifier (if installed) is turned off, the display will show "CLAR OFF" in green.

When it is turned on, the display will show "CLAR +/- nnn Hz" in red. The clarifier offset will not be used to adjust the reading on the dial or the numerical display of VFO-A (always the receive frequency), however the offset will be applied to the frequency sent to the Si5351 clock generator in receive mode only; it does not affect the transmit frequency.

The setting of the symbol *CLAR_SW_RESET* in the *config.h* file can be set to indicate whether the offset should be reset to zero or not when the encoder type clarifier only is turned off and the *CLAR_FA_RESET* symbol can be set to indicate whether or not the offset should be zeroed when the frequency is changed.

Split Mode Indicator

When not enabled, the word “SPLIT” is displayed in green; when it is on, it changes to red.

Turning split mode on or off also affects the numerical frequency displays as described in the section *VFO-A and VFO-B Frequencies* below.

The Dial

The dial always displays the receive frequency which is always in VFO-A.

Moving the main encoder will change the frequency and the frequency can also be changed via CAT control.

There are a lot of settable options in the *config.h* file that can be used to modify the appearance of the dial. They are pretty well described in the *config.h* file and in the *NJDA VFO - Hacker’s Guide*, so I won’t go through them all here.

Numerical Frequencies

The numerical frequencies for VFO-A and VFO-B are displayed inside a box (which you can turn off). VFO-A is always the receive frequency (plus or minus any clarifier offset) and the dial always shows the VFO-A frequency. The clarifier offset is not added to the dial reading or the numerical VFO-A reading.

With split mode enabled, VFO-B is the transmit frequency. You cannot change it directly with the encoder. It can only be changed via CAT control or by using the function button.

When not in split mode, the text “TR” is displayed to the left of the VFO-A numerical frequency in green. In split mode when not transmitting, the text “Rx” is displayed in green to the left of VFO-A’s numerical frequency and “Tx” to the left of the VFO-B numerical frequency in green.

When transmitting in split mode, the “Tx” associated with VFO-B turns from green to red. If transmitting while split mode is off, the “TR” associated with VFO-A changes to a red “Tx”.

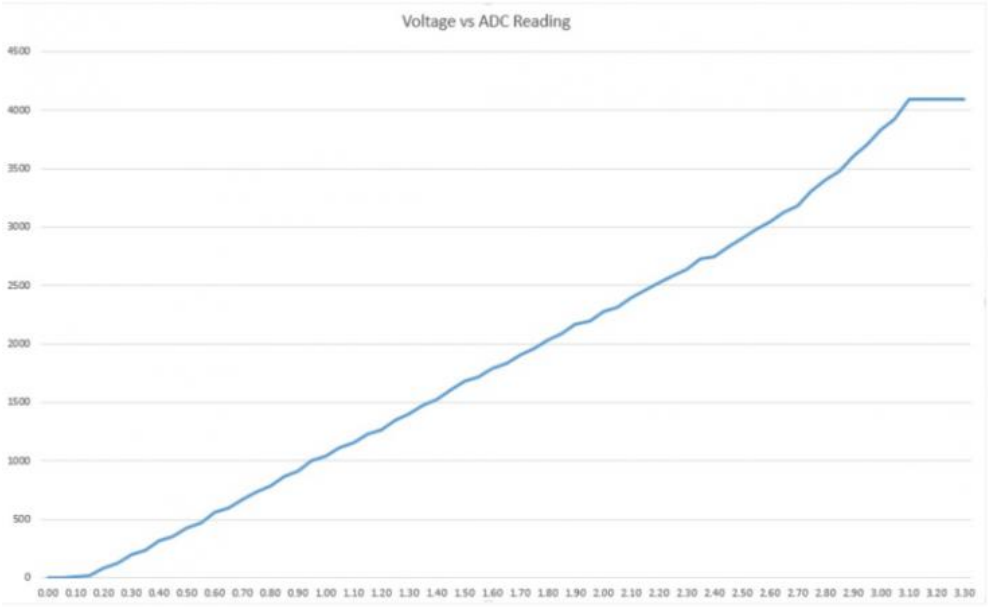
If enabled in *config.h*, the digit that will change when the encoder is moved can be underlined. The frequency increment can be changed if the optional increment button is installed.

Battery Monitor

If turned on, the battery voltage is displayed at the bottom of the screen. It shows the approximate battery voltage is a battery is connected (directly to the ESP32 board). If no battery is connected, the numbers displayed are meaningless!

Also note that the voltage displayed is that of the backup battery connected directly to the ESP32 board, not the voltage of, for example, a 12V battery that might be used to power an entire radio.

This graph shows the relationship between the voltage reading on a pin being read in analog mode and the numerical value that is read on the pin:



The value of the symbol *BATTERY_ADJ* in the *config.h* file can be adjusted to attempt to get more accurate readings for your particular hardware.

Frequency Management

Of course, the primary function of any VFO is to set the operating frequency of the radio; what more is there to understand? Well, a few things!

Digital VFOs versus Analog VFOs

There is one major difference between the behavior of a digital VFO versus an analog one particularly when you have the tuning increment set to a large value such as 1 KHz.

As you move the tuning control on a radio with an analog VFO, you're going to hear every signal on the band as you tune through them. This is not the case with a digital VFO. You will only hear signals within the bandpass of your receiver when the frequency is at precise multiples of the tuning increment. If the increment is set to a large value like 1 KHz and you have the receiver's bandwidth set to a narrow range as you might for CW operation, you may actually miss signals as the tuner jumps over them as opposed to passing through them.

Another thing you might notice when using a digital VFO is that you might hear short bursts of noise or clicks each time the frequency changes or anything else on the display is updated. The *NJDA VFO Hardware Manual* has a cure for this in the section on the display.

The default frequency change increments are part of the *bandData* array entry for each band.

Changing the Frequency Change Increment

If you installed the optional pushbutton that allows the frequency change increments to be modified, there are a couple of things about its behavior that needs a bit of explanation.

Each time you press the button, the software will cycle through a list of increments of 10 Hz, 100 Hz and 1 KHz. But be aware, that when you move the encoder, the next frequency tuned will be rounded off the increment currently set.

For example, let's assume you have the increment set to 10 Hz and are tuned to 7.123,450. If you move the encoder one click one way or another, the frequency will change to 7.123,460 or 7.123,440.

But, if you have the increment set to 10 Hz and are tuned to 7.123,450 and change the increment to 100 Hz then move the encoder, the next frequency will be either 7.123,400 or 7.123,500.

One other thing to note is that if you change bands, the increment will revert back to the setting for the new band regardless of what you had it set to on the vacated band.

Whenever the increment is changed, it is saved in the *bandData* entry for the active band. So if for example, you were operating on 40 meters and changed the increment to 1 KHz then move to 20 meters, the current setting of the increment (default or previously modified) increment will be in effect. If you move back to 40 meters, the increment will revert to 1 KHz which is the last setting you had for that band.

The Clarifier

If you have the optional clarifier installed (either using an encoder or a potentiometer), when it is turned on, the offset will be displayed on the screen, however neither the dial nor the numerical frequency displayed for VFO-A will change (the dial always tracks VFO-A).

Note that because of the difference in the behavior of digital tuning versus analog tuning described above, the addition of the clarifier is almost a necessity.

If the clarifier is showing an offset on the screen, the offset will be applied to the frequency actually sent to the Si5351 clock generator when the radio is in receive mode. The clarifier does not affect the transmit frequency.

If you are using the *ENCODER* implementation and have the *CLAR_SW_RESET* symbol in the *config.h* file set to *true*, the clarifier offset will revert to zero when it is turned off. If *CLAR_SW_RESET* is set to *false*, the offset will be retained the next time it is turned on. This does not apply if you are using the *POTENTIOMETER* implementation.

If you are using the *ENCODER* implementation and have the *CLAR_FA_RESET* symbol in *config.h* set to true, the clarifier offset will be reset to zero anytime the VFO-A (always the receive) frequency is changed. The VFO-A frequency can be changed by moving the main encoder or by using either the function button or CAT commands to swap VFOs (either temporarily or permanently) or when you SET VFO-A equal to VFO-B. Setting VFO-B equal to VFO-A will not reset the clarifier offset. Also note the offset will be set to zero regardless of whether the clarifier is turned on or not.

If you are using the *POTENTIOMETER* clarifier implementation, make sure you set both *CLAR_SW_RESET* and *CLAR_FA_RESET* to false or weird things happen.

VFO-A and VFO-B Frequencies

The VFO program really has two VFOs; identified as VFO-A and VFO-B. The frequencies in each can be swapped or copied from one to another either manually or via CAT control.

The behaviors of the two VFOs mimic those of the Yaesu FT-891 which is the CAT control language used as described in more detail in the *NJDA VFO - CAT Control Manual*.

VFO-A always contains the receive frequency regardless of the state of the *splitMode* variable. The actual receive frequency will be adjusted based on the clarifier offset if that is turned on; the transmit frequency is unaffected by the clarifier. The dial always shows the receive frequency without the clarifier offset and moving the encoder only changes the VFO-A frequency

When not in split mode, the VFO-A frequency is also the transmit frequency and you'll see the green "TR" indicator to the left of the frequency indicating that VFO-A is being used for both transmit and receive.

In split mode, VFO-B always contains the transmit frequency.

When split mode is turned on, the indicators to the left of the numerical frequencies change to a green "Rx" for VFO-A and a green "Tx" for VFO-B. When actually transmitting, the "Tx" indicator changes from green to red.

When turned off, the "SPLIT" indicator on the screen is displayed in green; when enabled, it is displayed in red.

Frequency Manipulation

The VFOs can be swapped or copied from one another either via CAT control or by operation of the “Function Button”.

There are five things that can be done:

- Set the VFO-A frequency. This can be done by moving the encoder or via the “FAnnnnnnnnn;” CAT command.
- Set the VFO-B frequency. This can be only be done directly via the “FBnnnnnnnnn;” CAT command.
- Copy VFO-A to VFO-B (CAT command “AB;”).
- Copy VFO-B to VFO-A (Cat command “BA;”).
- Swap VFOs (CAT command “SV;”).

The Function Button

A pushbutton has been added to the hardware which can also be used to manipulate the VFO frequencies. Here’s what it does:

- 1 short push — Swap VFO-A and VFO-B permanently.
- 2 short pushes — Toggle "*splitMode*".
- 3 short pushes — Set VFO-A equal to VFO-B.
- 4 short pushes — Set VFO-B equal to VFO-A.
- Held down — Swap VFO-A and VFO-B until released.

A short push is less than ½ second. Anything greater than 1 second is considered as the button being held down.

Setting the Transmit Frequency in Split Mode

In split mode, VFO-A is always the receive frequency and VFO-B is always the transmit frequency. There is no way using manual control to directly change the VFO-B (transmit) frequency.

The way to do this is to hold the function button down for more than one second which will temporarily swap the VFO-A and VFO-B frequencies. With the button held down, the main encoder can be used change the VFO-A frequency and when the button is released, the VFOs will swap back.

While the button is held, you will also be listening on what will become the transmit frequency which allows you to make sure it is clear before transmitting on it.

You can also use one short push of the button to swap them, then change the frequency then use a short push of the button to swap them back.

Manual Control versus CAT Control

CAT control is always enabled regardless of whether or not you've configured the VFO to use a manual band and/or mode switch.

But there's a catch! If you have either the physical mode or band switch enabled and the program receives a CAT command to change one or the other, it will get changed, but only for a few milliseconds!

Within 25 mS or less, the program will read the physical switch and the band and/or mode will revert back to what the switch is set to.

If the physical band switch is not enabled and a CAT command is received to change to a frequency to one in a different band than the one currently active, the program looks for the band that the new frequency falls into and automatically changes the band. If the new frequency does not fall into the range of any of the defined bands, neither the frequency nor band will change.

The VFO frequencies and split mode can be controlled via CAT messages or the function button at all times.

Note that although the Yaesu FT-891 CAT language does have three band change commands (Band Up, Band Down and Band Select), these are not recognized by the CAT handler. The only way to change bands via CAT control is by changing the VFO-A frequency.

Also note that if a CAT command to change VFO-B to a frequency outside of the active band, the request will be ignored.

Changing Modes and Increments

The default operating mode and frequency increment for each band is set in the *bandData* array.

At startup, the frequency, band, mode and increment will be set as described in the section *Startup Conditions* above.

When one and/or the other physical switches are not installed, frequencies and modes can be changed by CAT control. Changing the frequency under CAT control can also force a band change.

The increment can be changed using the (optional) increment button.

Any time the mode or increment is changed, the new values are stored in the *bandData* entries for the active band. If you change frequencies to a different band via CAT control, or change bands using a physical switch, the increment and mode will be set according to the entries in the new band.

If you go back to the previous band, the mode and increment settings will revert to what they were previously set for on that band.

In the Version 1.1 software, we added an optional pushbutton that can be used to cycle through the modes defined in the *modeData* array. This approach to changing modes has an advantage over the use of a rotary type switch read by the PCF8574 in that it can be used in conjunction with CAT control.

Suggestion Box

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