# NJAD VFO – Hardware Manual
# Version 1.0


## John Price – WA2FZW
## Glenn Percy – VK3PE


## Table of Contents


Introduction .... 3

General Architecture – WA2FZW Version .... 4

The Processor Board .... 5

- The ESP32 Processor .... 8
- The Display .... 9
- Frequency Encoder .... 10
- R6 & R7 .... 11
- The Clarifier Encoder or Potentiometer .... 11
- The Function Button .... 12
- The Frequency Increment Button .... 12
- The Radio Connections .... 13
- R4 & R5 .... 14

The RF Board .... 15

- Designing the Filters .... 18

The PCF8574 Board .... 19

- Address Pins .... 22
- Data Pins .... 22
- Other Options .... 23

Power Source .... 23

GPIO Pin Assignments .... 24

General Architecture – VK3PE Version .... 24

The VK3PE Circuit Board .... 25

- Power Source .... 28

The Display ........ 29
Frequency Encoder ........ 30
The Clarifier Encoder ........ 32
The Si5351 and Low Pass Filter ........ 32
Bill of Materials – WA2FZW Version ........ 33
Processor Board ........ 33
RF Board ........ 34
PCF8574 Board ........ 34
External Components ........ 35
Bill of Materials – VK3PE Version ........ 35
Coming Attractions ........ 39
Suggestion Box ........ 39

## Introduction

This document describes the hardware for the the ESP32 based VFO that I have been working on along with Glenn (VK3PE) and Jim (G3ZQC).

It's not just another digital VFO! This project is a heavily modified version of the VFO originally created by T.J. Uebo (JF3HZB).

The hardware modifications made so far include:

- Multiband capability
- Allow the use of a PCF8574 chip to read a band switch.
- Allow use of a PCF8574 chip to read a mode switch; note, different PCF8574s can be used for the band and mode switches or they can be read from a single chip.
- Allow use of either an ST7735 or an ILI9341 based TFT display and allow a display size of up to 240x320 pixels.
- Added the ability to use a high speed encoder for frequency control and adjust the effective PPR rate in the software.
- Addition of a clarifier which can be implemented via a second encoder or a potentiometer (still having issues with the later).
- Added the ability to read the TX/RX status of the radio. This is needed for the clarifier operation and to prevent changing the frequency while the radio is transmitting.
- Added CAT control.
- Added the ability for the VFO to key the transmitter when commanded to do so via the CAT interface.
- 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.
- Added an optional button to cycle the frequency increments through 10Hz, 100Hz and 1KHz.
- Added the ability to run the display backlight from either the on-board 3.3V or from 12V from the radio.

Something to be aware of; this is not your typical build it, program it and plug it in project. First of all, the hardware configuration needed is going to depend a lot on the radio you intend to use it with.

From both the hardware and software perspective We’ve tried to make it possible to interface this with pretty much any radio, but you might find that in places, you may have to use a HIGH indication for something where the current software expects a LOW or vice-versa. Such things are probably going to require changes in the hardware and/or software.

There are actually two different versions of the hardware implementation described here:

- The WA2FZW (John’s) version uses multiple circuit boards that can be stacked so there are several different configurations available. The PCBs use all through-hole components.
- The VK3PE (Glenn’s) version consists of a smaller single PCB which uses SMD components. His version was designed specifically for a Yaesu FT-7 radio in which space is very limited.
- The third version was designed specifically for my Swan-250C radio and will be described in a separate document when I finish it.

The electrical circuity for all three versions is almost identical and the software will run equally well on any version of the hardware.

## General Architecture – WA2FZW Version

The hardware I describe in this document was designed primarily to allow the VFO to be installed in my old Swan-250C 6 meter transceiver and Glenn’s Yaesu FT-7 transceiver; however, I’ve provided a lot of flexibility in how to interface any radio with the VFO.

I band a major boo-boo in the version I designed for the Swan; the circuit boards didn’t fit in the space allocated, so I designed another set of PCBs for that radio. The boards I describe here should work in any radio that they can be crammed into!

The major components for the WA2FZW implementation include:

- Hiletgo 240x320 pixel ILI9341 TFT display
- TTGO T8 V1.7 ESP32 Processor board
- ~400 pulse per revolution optical encoder
- ~25 pulse per revolution mechanical encoder

- Adafruit Si5351 Clock Generator Module

The links above are not the only places where the parts are available; shop around for the best prices!

The WA2FZW version uses three PCBs which are designed to be stacked on top of each other:

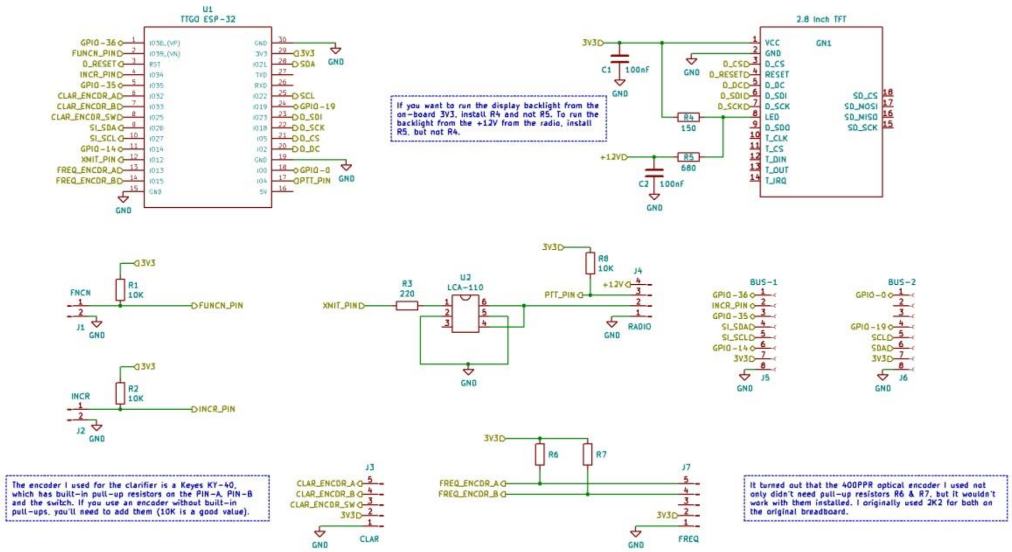
- The Processor Board, which contains the ESP32, the display and connectors for the encoders, radio PTT circuitry and a function button (more about that later) and a button to change the tuning increments.
- The RF Board which contains the Si5351 module and provision for three low pass filters if required.
- The PCF8574 GPIO expander board. This allows an additional eight GPIO-like pins to be added to the unit; as a matter of fact, multiple PCF8574 boards can be added to the stack for even more additional GPIO pins.

All three of the WA2FZW PCBs will be thoroughly explained as we proceed.

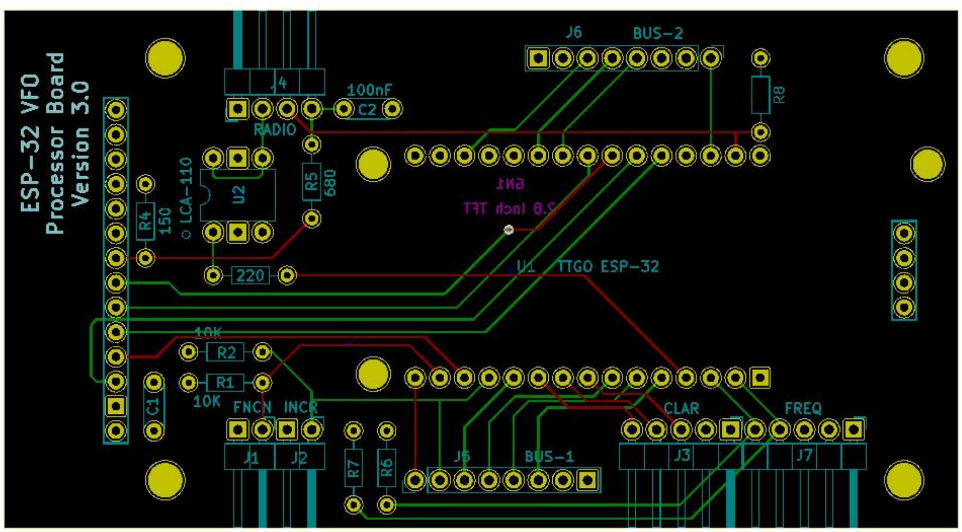
## The Processor Board

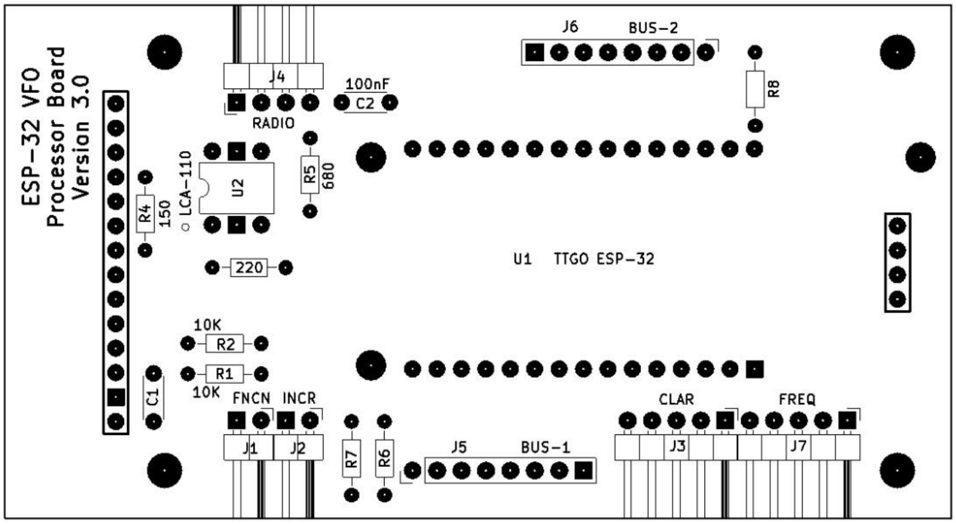
This board is the main component of the whole VFO. It is designed so that it can be mounted on the back of the display or connected to the display via a cable and contains almost everything needed to make the VFO work except the Si5351 clock generator and related circuitry and the PCF8574(s).

Here's the schematic of the WA2FZW processor board:



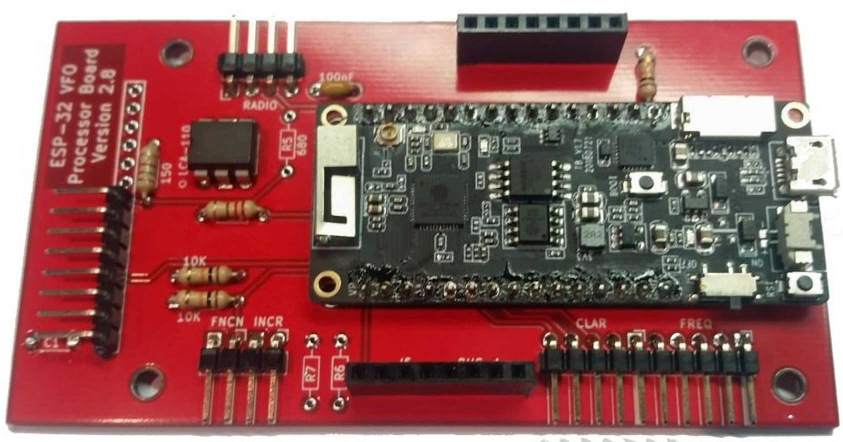

And here are a couple of pictures of the actual PCB layout:






The vertical headers labeled “BUS-1” and “BUS-2” provide the board stacking capability. You’ll see the same headers on all the boards.

Here’s a picture of the assembled (Version 2.8) processor board. The only difference from the Version 3.0 PCB is that the location of C1 moved:



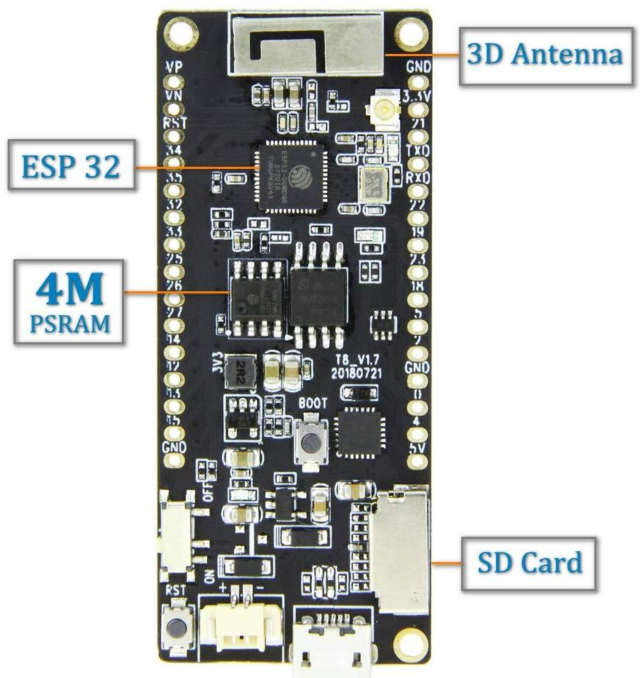

Because I needed to connect the processor board to the display via a cable as opposed to having the display mounted directly on the rear of the processor board, I had to add the horizontal pin header on the left side of the PCB. As you can see, this presented a problem with the position of C1, which I mounted on the back side of the board.

I moved C1 on the Version 3.0 PCB, which is reflected in the Gerber files included in the distribution package.

### The ESP32 Processor

The processor is an ESP32 WROVER development board from TTGO. This board is different from the ESP32 WROOM board used in the original implementation in that it includes an additional 4Mb of PSRAM. The additional PSRAM was needed to allow the use of the larger (240x320) displays. Note, however that although even larger displays could technically be used, as explained in the section on *The Display*, the VFO dial will lose the illusion of behaving like an analog dial.

Here’s what the proper board looks like:




All the versions that we have built so far use the TTGO V1.7 board. As of this writing, I believe there is now a V1.8 board available. We haven’t tested with the newer board, but I suspect it would work just fine.

The use of the ESP32 is key to this project for two primary reasons; first is speed. The ESP32 is capable of running at 240MHz. Secondly it is a dual core processor so there can essentially be two programs running simultaneously sharing common memory and GPIO pins.

Another great feature of the ESP32 is that interrupts can be generated from any GPIO pin, unlike the Arduino which has a limited interrupt capability. A lot of things in the software are interrupt driven.

If there is any downside, it’s that some of the GPIO pins really aren’t; some can only be used for inputs and some only for outputs. It also doesn’t have as many GPIO pins as, for example an Arduino Mega. Also, unlike the Arduinos, there are no internal pull-up resistors on the GPIO pins.

Also note that the ESP32 (and all the other components in the system) is a 3.3V device. Applying more than 3.3V to any of the pins can destroy at a minimum that particular pin or worst case the whole processor!

## The Display

TJ’s original design used a 128x160 pixel display; however that was way too small physically for my Swan-250C project which has a 2.5” x 1.25” hole where the analog VFO dial is. It took using a different processor and a lot of software modifications, but I was able to get it to work with the 240x320 pixel display listed above.

The display I am using uses an ILI9341 driver chip. The smaller displays can also be used; however one might have to use a cable to connect the display to the processor board instead of being to directly connect the two together depending on the pinout of the display to be used. I actually ended up having to do that for the Swan due to the limited space behind the front panel.

The display used also has a touch-screen capability however we’re not using it (yet).

The software has been tested with the ILI9341 based display and a number of smaller ST7735 based displays. The software should also be able to work with a number of other displays (see the *ESP32 Based VFO - Hacker’s Guide*) although no others have been tested.

In theory, displays larger than 240x320 could be used, however as previously mentioned, the illusion of the dial being an analog type dial will be lost; that is already slightly noticeable when using the 240x320 display. This is due to the time it takes to transfer the large amount of pixel data from the processor to the display.

The software updates the entire display using one large data transfer kind of like a TV screen. As the maximum SPI bus speed for the ILI9341 display is only 40MHz (27MHz for the ST7735 based displays), the time that it takes to transfer the data doesn’t allow the frame rate to be high enough for the analog illusion. For the 240x320 display, the software can only update the screen about 4 or 5 times per second whereas for the 128x160 display it can update the screen about 14 times per second.

One problem you might encounter is short noise bursts whenever the display is updated. I had the problem in the installation in my Swan-250C, however Glenn did not in the FT-7 installation. I fixed the problem in the Swan by putting one of these ferrite gimmicks

around the clock and data lines to the display along with the power lead to the backlight (which in my case is powered from the radio as opposed to from the on-board power).

## Frequency Encoder

The processor board contains pin headers to connect two encoders; one for frequency control, and the other for a clarifier.

I highly recommend the use of a high-speed optical encoder for the frequency control. The one I used is similar to this one however I found it on eBay for considerably less!

The 400 PPR (Pulse per Revolution) rating is a bit misleading. That is the number of pulses are produced by one of the two clock pins per revolution. The software, however, looks at both transitions of both pins so that the direction of rotation can be determined. This results in the encoder actually producing 1,600 interrupts per revolution, which is way too many to be useful. The software has a customizable factor (*ENCDR_FCTR*) which can be set to reduce the effective speed. I found that a factor of 40, which results in the software seeing and effective speed of 40 interrupts per revolution worked pretty well.

The particular encoder I obtained had an issue! It didn't work well on 3.3V. One way of dealing with that problem would be to use a higher voltage for the Vcc input to the encoder and adding voltage dividers to the output pins to limit the voltage going to the ESP32 to 3.3V.

Instead, I decided to try modifying the particular encoder I have. When I opened it up, I discovered that there was a diode (I assume for polarity protection), a bunch of resistors in parallel and an LM7805 regulator. I removed the LM7805 and hot-wired the Vcc line into the encoder to where the LM7805's output was connected. It now works fine on 3.3V.

You can use one of the mechanical encoders such as the KY-40 for the frequency control, but these only generate about 25 interrupts per revolution, which I find a bit slow. It would also be better to use one without the mechanical detents for a more natural feel to the tuning dial. The divisor factor in the software should be set to '1' if using one of these.

### R6 & R7

If you use a mechanical encoder with built-in pullup resistors (as the KY-40 has) you don't need to install R6 and R7 on the PCB.

You may or may not need them with other encoders. When I originally built the breadboard version of the VFO, I used 2K2 pullup resistors on the A & B outputs of the optical encoder. However when I assembled the first version of the PCB, the pullup resistors prevented the encoder from working correctly. Why is still a mystery!

## The Clarifier Encoder or Potentiometer

A mechanical encoder like the KY-40 is great for the clarifier, where you're not going to be changing the offset by a whole lot (the software works in 10Hz steps). The KY-40 has built-in pullup resistors on PIN-A, PIN-B and the switch. If you use one without the built-in pullups, you'll need to provide them (10KΩ is a good value).

You do want to get one with the built in pushbutton to turn the clarifier on and off, or a separate pushbutton switch will be needed to do that.

Instead of using an encoder for the clarifier, the software allows one to optionally use a potentiometer. Personally, I tried this with a cheap 100Ω trim pot and found that it had the jitters. When I did that, we had the potentiometer assigned to a different GPIO pin than it is now and since we changed the assignment, Jim (G3ZQC) who has also been working with me on this from the beginning seems to have made it work reasonably well, but I still haven't.

I'm still working on improving this option.

If you use the potentiometer option, you will also need a momentary pushbutton to toggle the clarifier on and off.

## The Function Button

The function button is a momentary on type pushbutton. It is used to manage the contents of the dual VFOs built into the software and to toggle the split frequency mode on or off.

The functions performed are:

- 1 short push — Swap VFO-A and VFO-B.
- 2 short pushes — Toggle "*splitMode*" on and off.
- 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.

## The Frequency Increment Button

The frequency increment button is a momentary on type pushbutton. It cycles through frequency change increments of 10 Hz, 100 Hz and 1 KHz. The current increment is indicated by a red line under the appropriate digit on the display.

## The Radio Connections

There is a 4 pin connector provided to interface with the radio's transmitter.

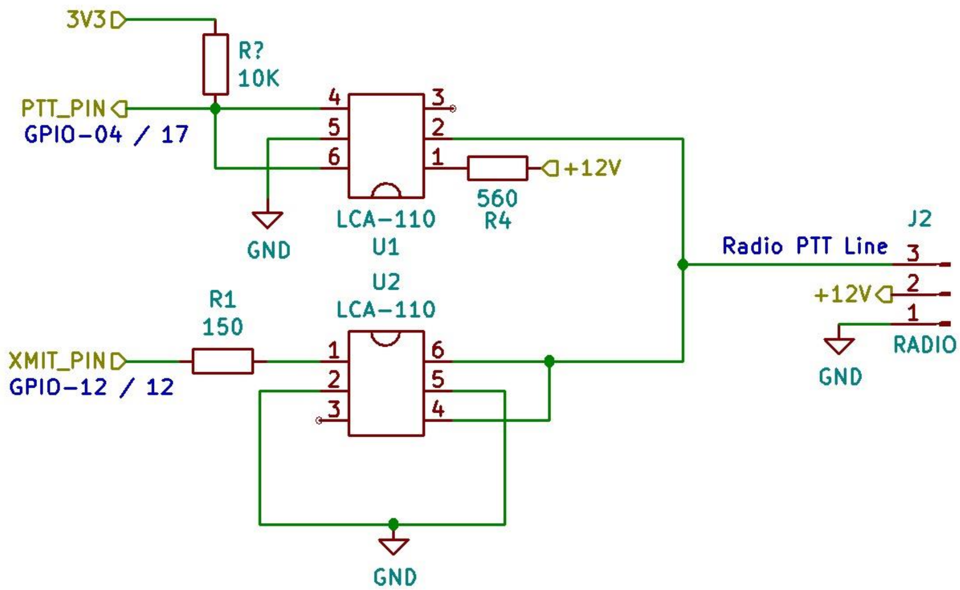
An LCA-110 solid state relay is controlled by GPIO pin 12 on the processor. The output of the SSR can be connected to the radio's PTT line to key the transmitter under command of the processor. This is primarily used when operating the radio under CAT control (see the *NJAD VFO - CAT Control Manual* for full details about how that works). The assumption is that grounding the radio's PTT line activates the transmitter.

Another pin on the connector is used for an indication that the radio is transmitting. The indication is designed to be provided by a relay contact, so no additional isolation is needed, however when I designed the 3rd version of the PCBs for the Swan, I found a better way of watching the PTT line and added a 2nd SSR on those boards.

Depending on how your particular radio indicates that it is transmitting, you might need to add some sort of isolation as I did for the Swan.

One might ask why we need both an indication that the radio is transmitting and a means of making it transmit. The answer is that the CAT control can order the radio to transmit, or the operator could do so by keying the microphone.

When I designed the new PCBs specifically for the Swan, I changed the PTT/Keying circuitry slightly. I will publish information about those boards when I finish modifying that radio, but here's the modified circuit should you want to implement it as an add-on:




## R4 & R5

The 4th pin on the RADIO connector is to provide an external DC voltage from the radio to operate the backlight in the display. In the case of my Swan-250C, this is a 12V source. The reason I added this is that I plan to run the VFO from USB power from my PC, which is essentially on all the time. By using the radio to power the backlight, the display will go dark when I turn the radio off even though the rest of the VFO is still powered up.

R4 (but not R5) should be installed if you want to power the backlight from the on-board 3V3 source.

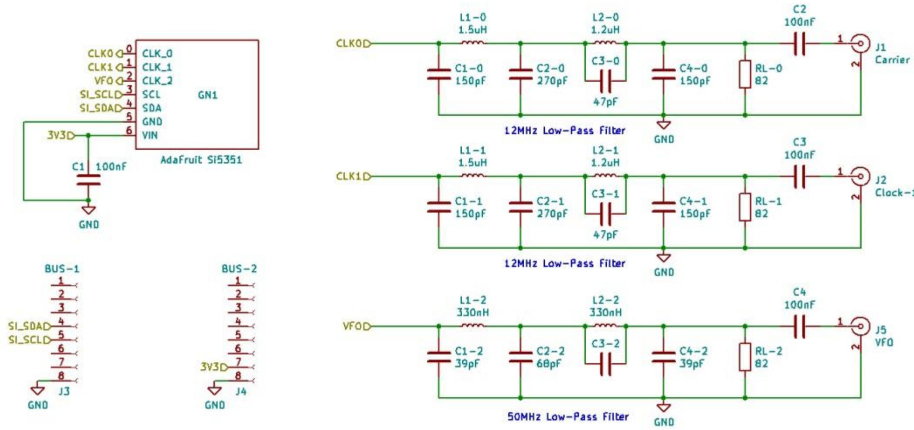
If you want to power the backlight from the radio, install R5 but not R4. If the voltage from your radio is not 12V, you may need to change the value of R5.

***DO NOT*** install both R4 and R5, or you are likely to see pretty white smoke coming from one place or another!

## The RF Board

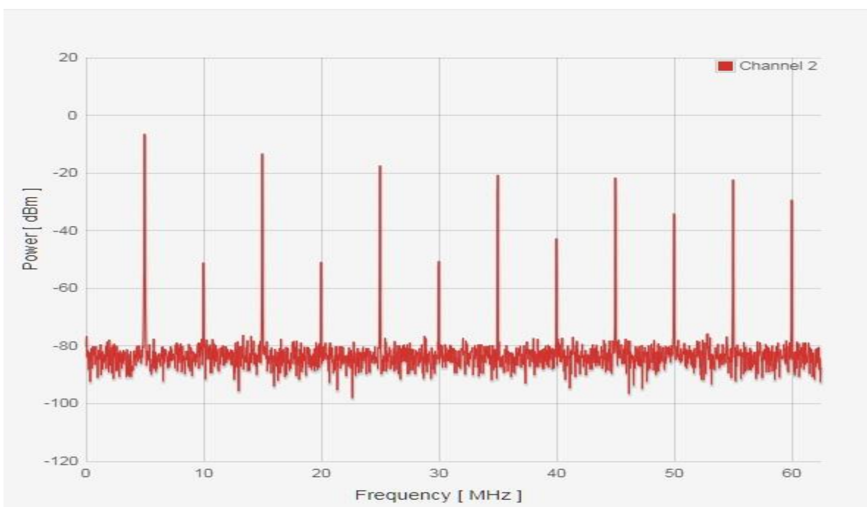
The second board in the package is the RF board. It contains the [Adafruit Si5351 Clock Generator Module] and provision for three low pass filters for each of the Si5351's clock outputs.

Here's the schematic:



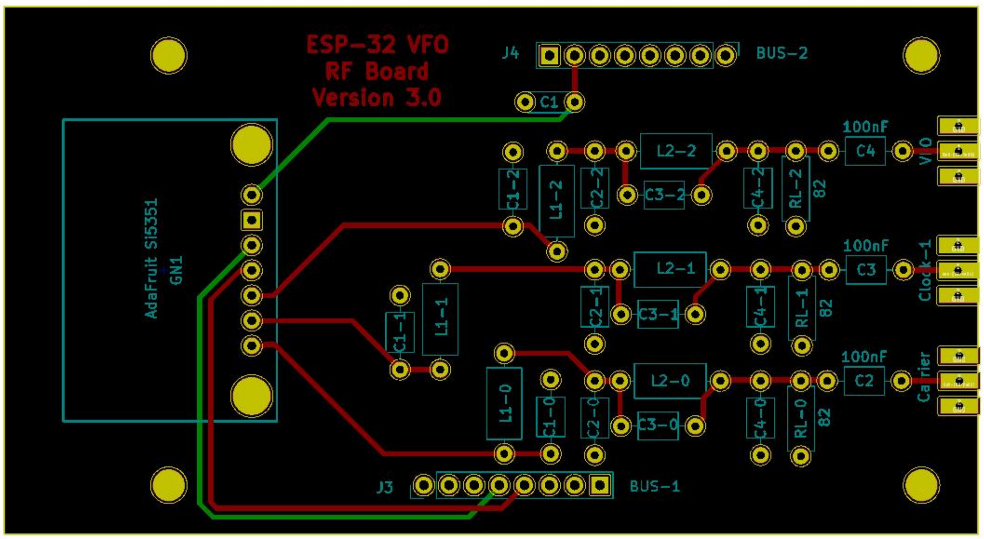

The values of the components shown in the filters are those used for my Swan-250C. The VFO injection frequency for that radio is in the 39 MHz to 43 MHz range and the carrier oscillator is about 10.9 MHz, so I designed the low pass filters for 50 MHz and 12 MHz.

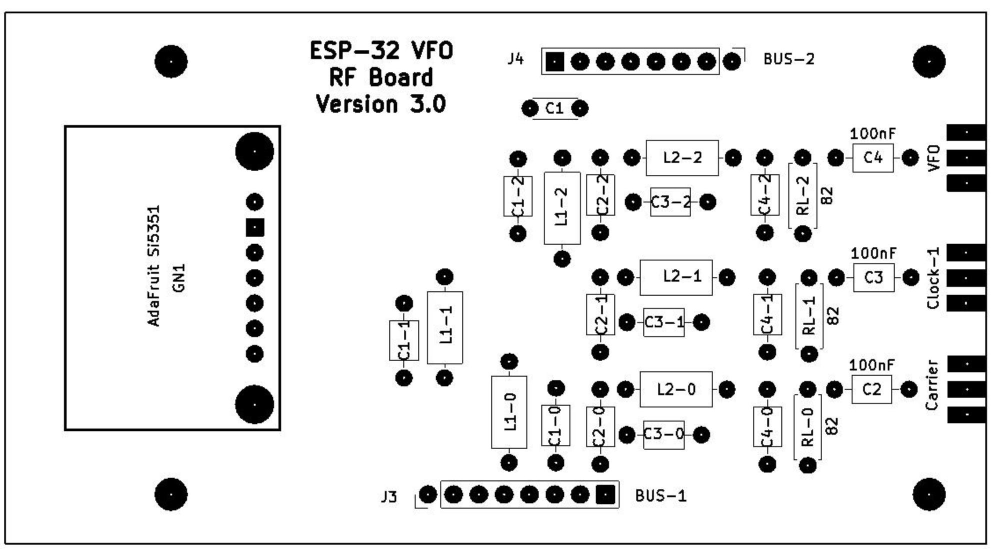
The output of the Si5351 is an imperfect square wave and thus there is lots of harmonic energy. Here's what its output looks like set to 5 MHz (VFO frequency for Glenn's FT-7) on the spectrum analyzer:



Note that the CLK2 output of the Si5351 is used for the VFO frequency and CLK0 and CLK1 are (optionally in the software) used for the carrier oscillator. The software is able to be configured to use only CLK0 or CLK1 for the carrier oscillator or to use them together in a quadrature mode (as was the case in the original implementation).

Here are a couple of pictures of the circuit board layout:



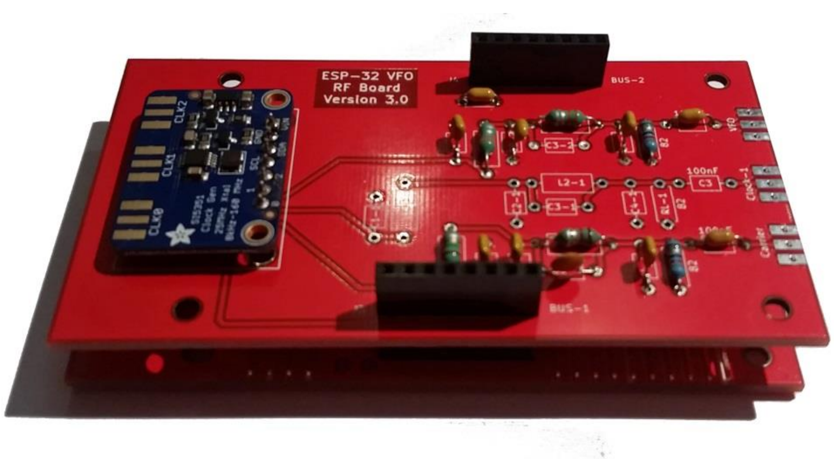
Note, the filter components are only labeled with their designations from the schematic and no actual values are shown as they will be different for every application.

The filters are optional. The Si5351 puts out imperfect square waves, which are rich in odd harmonics (the even ones are there also as the output isn't a perfect square wave).

If you don't need them (I didn't for the VFO input to the Swan-250C (not sure about the carrier oscillator yet) you can either run a short piece of RG-174 from C1 to C4 on the specific filter, or mount the SMA RF connectors directly on the Si5351 module (I recommend the 2nd solution).

It might also be a good idea to install 82Ω resistors in lieu of C1-0 and/or C1-1 if those clocks are not used (the stated output impedance of the Si5351 is ~85Ω).

Here's the (almost) completed RF board sitting atop the processor board; I haven't installed the SMA RF connectors yet. Also only the low pass filters for CLK0 (carrier oscillator) and CLK2 (VFO) are installed as I won't be using CLK1.




If you use CLK1 on the RF board, there is very little clearance between the SMA RF connector and the USB connector on the Processor board. Even though the SMA connector and USB connector are both at ground potential, I would recommend a piece of electrical tape between them.

## Designing the Filters

Glenn (VK3PE), who has been working on this project with me from the start, introduced me to a simple freebie tool for designing the filters called [Elsie from Tonne Software](Elsie from Tonne Software).

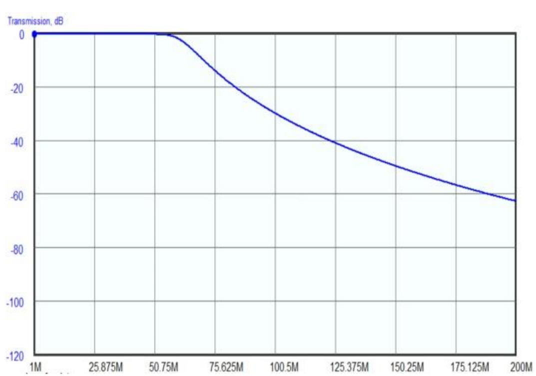
The basic design used here is a “Capacitor-input **lowpass**” 5$^{th}$ order Chebyshev design.

Other things you’ll have to specify are the input and output impedances for the filter (the datasheet for the Si5351 indicates its output impedance is ~85Ω). The filters all have an 82Ω load resistor on the output and a 100nF capacitor in series with the output.

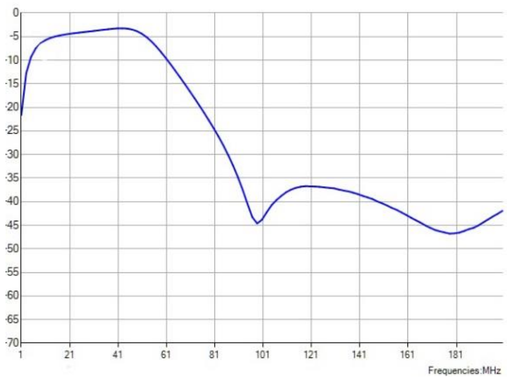
The resulting design using the model specified above will not include capacitor C3. Reducing the value of L2-n by about 25% and adding that component to the design adds a notch filter for a particular frequency; you have to do it by trial and error though.

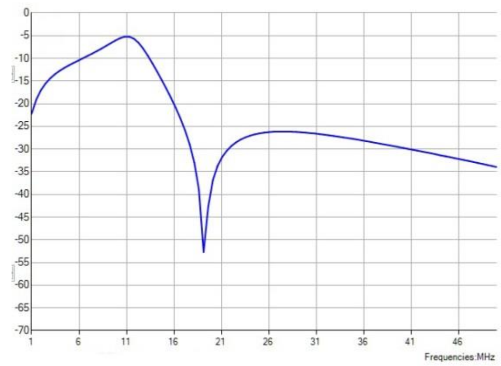
One thing I learned the hard way about Elsie is that it is case sensitive; in particular, I learned that “m” and “M” don’t mean the same thing, so pay attention to how you designate values for things! “M” means “Mega” and “m” means milli!

Here are the predicted throughput plots for the filters for the Swan-250C (50MHz and 12MHz):




And here are the actual plots done with the NanoVNA vector network analyzer (note, the vertical scales are different from the predicted plots above):



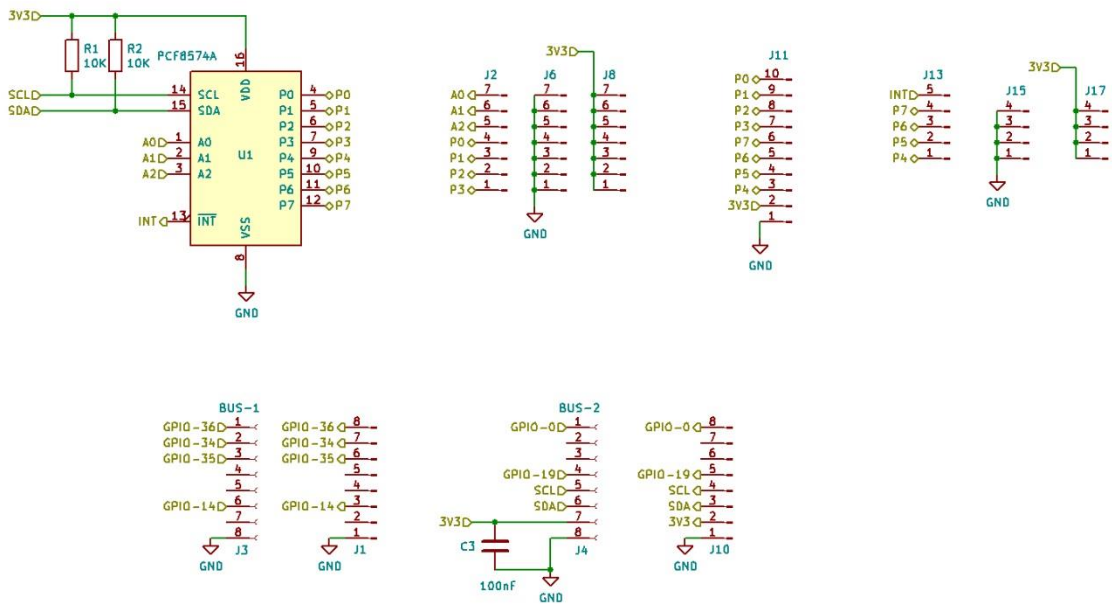

## The PCF8574 Board

The third board in the set is the PCF8574 board. You’ll only need one (or maybe two) of these if you need to read a physical band and/or mode switch in your radio.

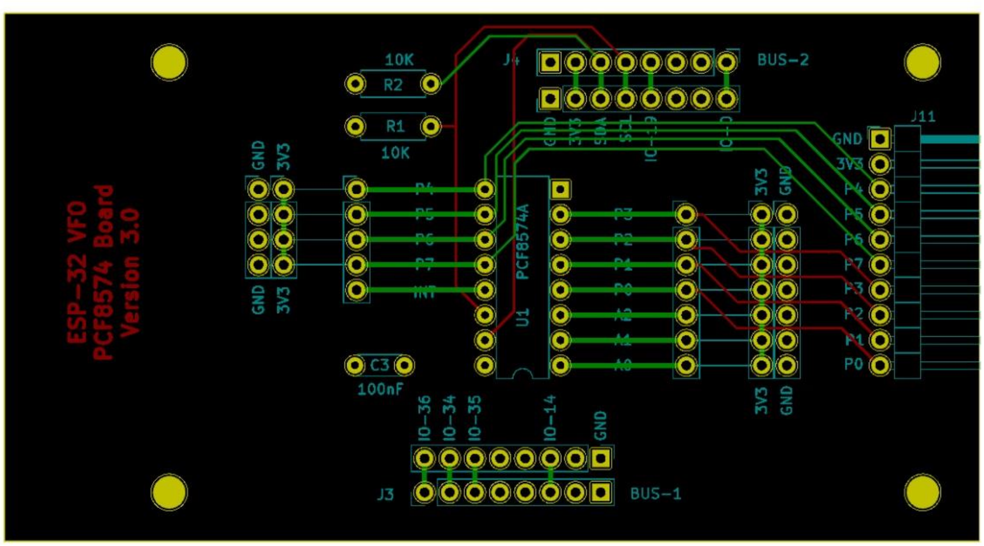
The PCF8574 is a very useful chip that can be used with the ESP32 or any of the other small processors. It allows you to effectively add 8 GPIO pins to the processor. The chip also provides the capability to generate an interrupt (which has to be read by a real GPIO pin).

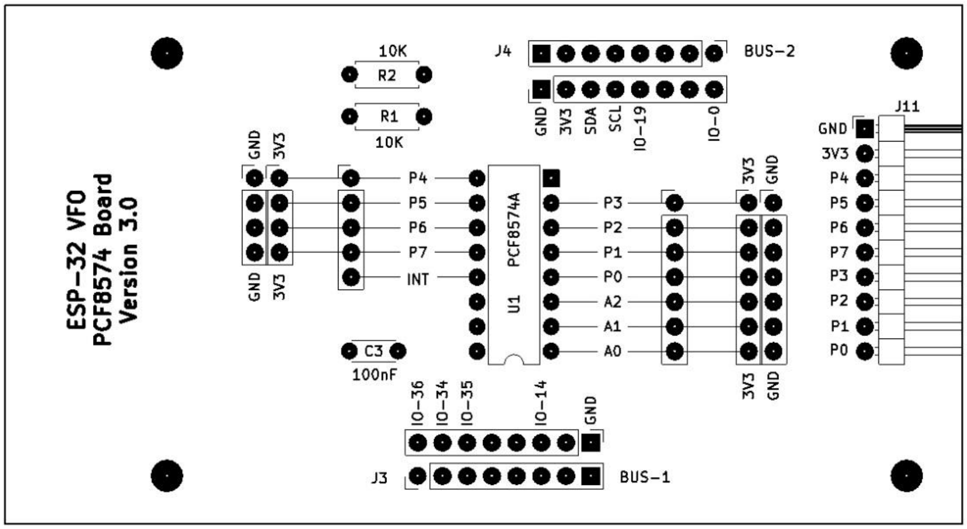
Either a PCF8574 or a PCF8574A can be used on the PCB. The only difference between the two versions is the I2C address. The PCF8574 address can be selected in the range of 0x28 to 0x2F and the PCF8574A address range is 0x38 to 0x3F. By using a mix of the two types, one could actually add 128 extra GPIO pins to the processor!

Here’s the schematic for the board:




Looks kind of strange, no? Just a chip and a bunch of headers! Those are not really headers, but rather places to install jumpers that you will need to add to set the I2C address for the chip and for pull-up or pull-down resistors as needed for your project. It becomes a little clearer looking at the PCB layout:

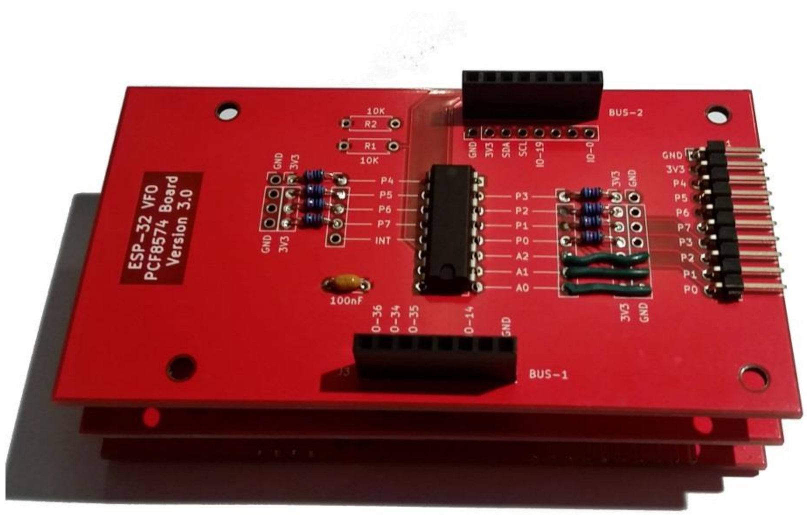





On each side of the PCF8574 chip are 3 sets of pads. The ones closest to the chip itself are connected to the address and data pins of the chip. The next row of pads going away from the chip are all connected to 3V3 and the final set are all connected to ground.

The horizontal pin header on the right side of the board is used to connect the board to the band and/or mode switch in the radio. Note the data pins are not in numerical order (it made the board easier to lay out that way).

Here’s the board on the top of the stack:



The green jumpers are connecting the address pins (A0 to A2) to ground. I used insulated wire as the wires need to pass over the 3V3 pads.

The resistors are pullups on data pins P0 through P7.

There are two additional resistors (R1 & R2) as shown on the schematic which are not installed. These are optional pullup resistors for the I2C bus. [The PCF8574 datasheet] would seem to indicate they are necessary; however, my build works fine without them; that may be due to how the ESP32 library software handles the I2C bus. If you need them, add them.

Also, if you plan to use 2 of the PCF8574 boards for different things, R1 and R2 need only be installed on one of the boards.

## Address Pins

As mentioned above, the I2C address of the chip can be programmed to a range of 8 addresses. To set it to 0x28 or 0x38 (depending on whether you are using the PCF8574 or PCF8574A), connect all three address pins to ground. To program it to address 0x29 or 0x39, connect A0 to 3V3 and the other two address lines to ground, and so on and so forth.

## Data Pins

The software can be configured to look for either an active HIGH or LOW signal from either the band switch or mode switch as the indication of the selected switch position. It expects all the rest of the active data pins to be in the opposite state.

Let’s assume that your radio has a 5 position band switch and that the active switch position is indicated by a LOW condition as is the case with Glenn’s FT-7. If that is the case, you should add pullup resistors (10K is probably a good value) to data pins P0 through P4, and connect data pins P4 through P7 directly to 3V3 assuming they aren’t used for anything else (no need for resistors).

If on the other hand, the radio has a 5 position switch and the active pin is indicated by a HIGH condition, then you would want the resistors on P0 through P4 to be installed as pull downs and the unused pins should be connected to ground.

As distributed, the software assumes that the active pin will be LOW.

### Other Options

Notice that there are also pads next to the "BUS-1" and "BUS-2" stacking headers. Should you want to modify the software to use the PCF8574's interrupt generating capability a jumper wire can be run from the chip's INT pin to any one of the few GPIO pins still available. According to page 10 of the PCF8574 datasheet, the interrupt pins from several PCF8574s can be connected to a common GPIO pin on the processor, although we didn't test that capability as the software isn't using the interrupt capability. The interrupt line does require an external pullup resistor however.

Note also that although there are 6 GPIO pins available, there are restrictions on which ones can be used for what. This web page describes those restrictions. GPIO-19 should definitely be avoided as although technically not used, it seems to be intertwined with the library code that handles the display. I've tried using it for a couple of other things unsuccessfully.

Since the board was designed, GPIO pin 34 was assigned to the frequency increment button, so if you have that option installed, that pin won't be available for other uses.

One final note; as mentioned above, you can use two of these boards if needed; one for the band switch and one for the mode switch simply by stacking them and programming the addresses differently. The software does currently support this option.

## Power Source

In my Swan-250C implementation, I plan to power the entire VFO except the backlight for the display from USB power. You can, however choose to power the VFO from the radio's power (voltage appropriately reduced).

The original versions of the ESP32 development kits had a problem where the board could not simultaneously be powered from a power supply and the USB connector. This is not a problem on the board used here.

## GPIO Pin Assignments

The following is a summary of the GPIO pin assignments. Note that these represent how I have them assigned on my PCBs and on Glenn's PCB. If you are using either of those PCB designs, you should NOT change any of the assignments other than flipping the numerical assignments for the encoder pins if the encoders operate backwards.

If you are using a PCB of your own design, you can change the assignments subject to the limitations described on this web page.

| | |
|---|---|
| 2 | Display DC |
| 4 | Transceiver PTT line (LOW indicates transmitting) |
| 5 | Display - CS (VSPI Bus) |
| 12 | For CAT control - Keys the transmitter |
| 13 | Frequency Encoder – A |
| 15 | Frequency Encoder - B |
| 18 | Display - SCLK (VSPI Bus) |
| 19 | Reserved - Display MISO (VSPI bus - for reads) |
| 21 | PCF8572 - SDA (Standard I2C bus) |
| 22 | PCF8572 - SCL (Standard I2C bus) |
| 23 | Display - MOSI/SDO (VSPI Bus) |
| 25 | Clarifier Switch |
| 26 | Si5351 - SDA (Alternate I2C bus) |
| 27 | Si5351 - SCL (Alternate I2C bus) |
| 32 | Clarifier Encoder - B (or optional Potentiometer Clarifier) |
| 33 | Clarifier Encoder - A |
| 34 | Increment adjustment button (optional) |
| 35 | Battery monitor (optional) |
| 39 | Function button (optional) |

## General Architecture – VK3PE Version

Glenn's implementation was designed specifically for his Yaesu FT-7 transceiver which has a very limited amount of space in which to install the new VFO.

His version uses a single PCB and SMD components. The ESP32 WROVER® board, the Adafruit Si5351 clock generator and a PCF8574 GPIO expander chip are all on the single board. Glenn's radio does use the PCF8574 to read the physical band switch in the radio.

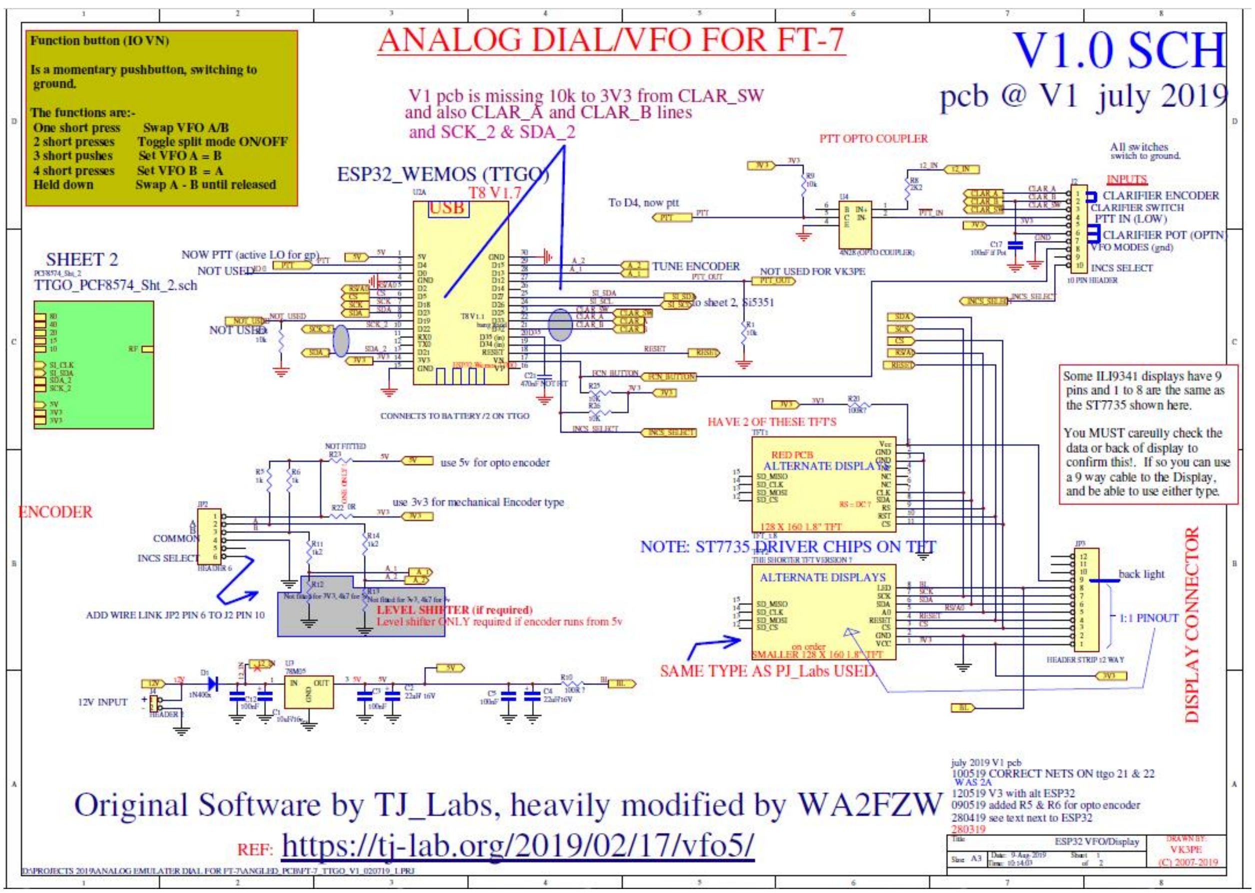
His design also uses the smaller 1.8” 128x160 pixel ST7735 based display.

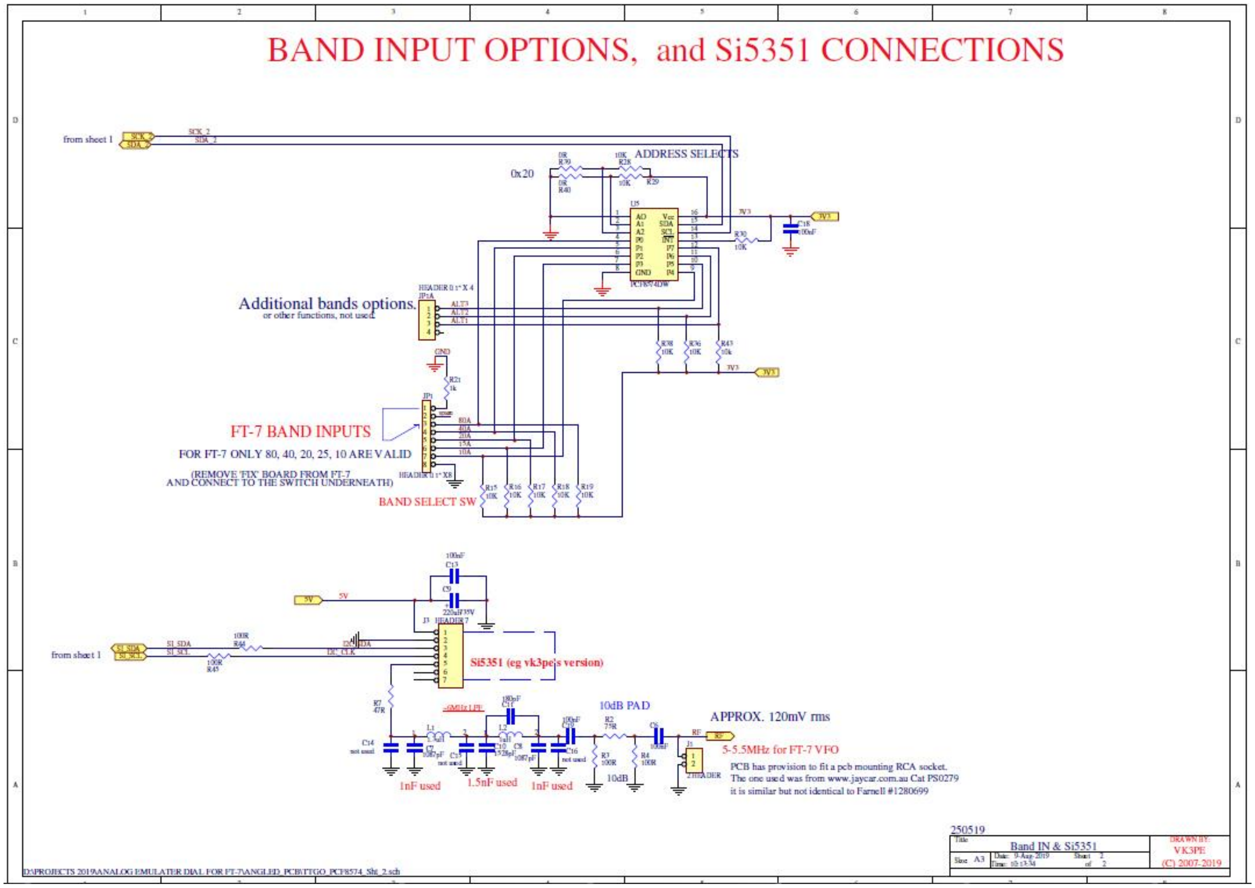
Most of what was previously described about the other external components above (buttons, encoders, etc.) applies to Glenn’s implementation as well except for component reference numbers which will be different.

The software runs equally well on both hardware implementations; both implementations use the same GPIO pin assignments.

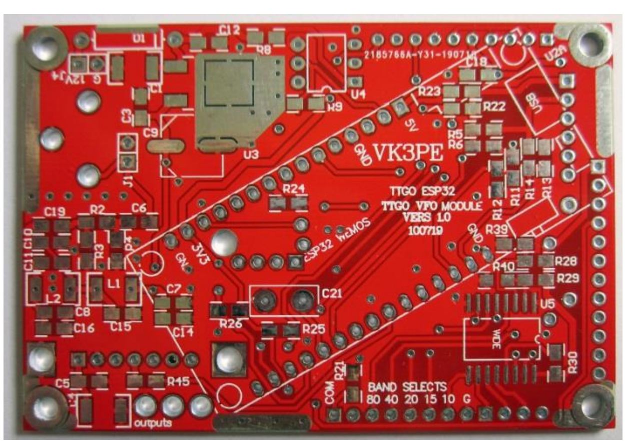
## The VK3PE Circuit Board

For the FT-7, a single ‘vk3pe’ PCB is used to take the ESP32 board. The circuitry for both hardware implementations is almost identical; there are some minor variations. Here are Glenn’s schematics:

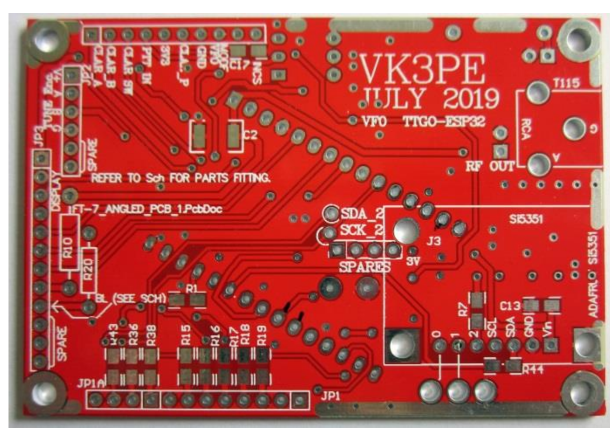





This is the blank VK3PE PCB. The PCB contains all of the required interfacing components. The ESP32 module is mounted on an angle due to the restricted height in the FT-7. It also allows easy access to the USB port for programming:

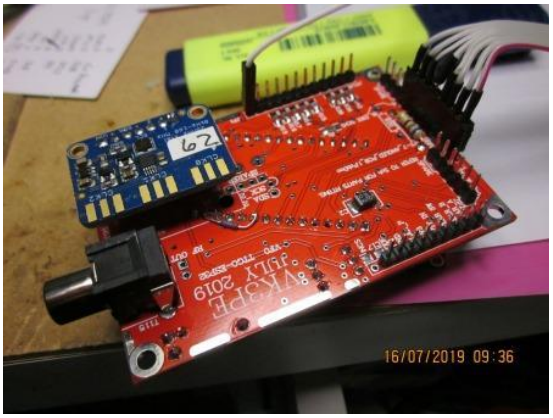


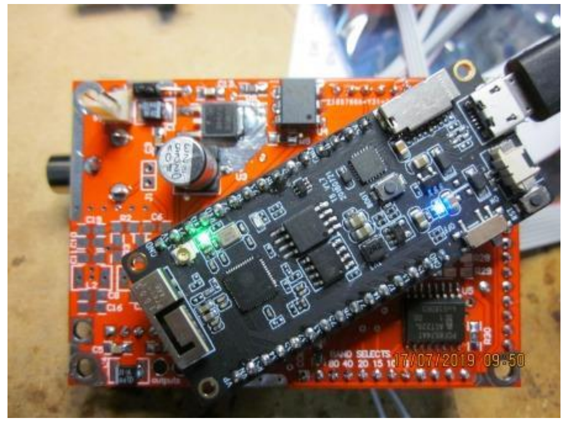

Top View




Bottom View

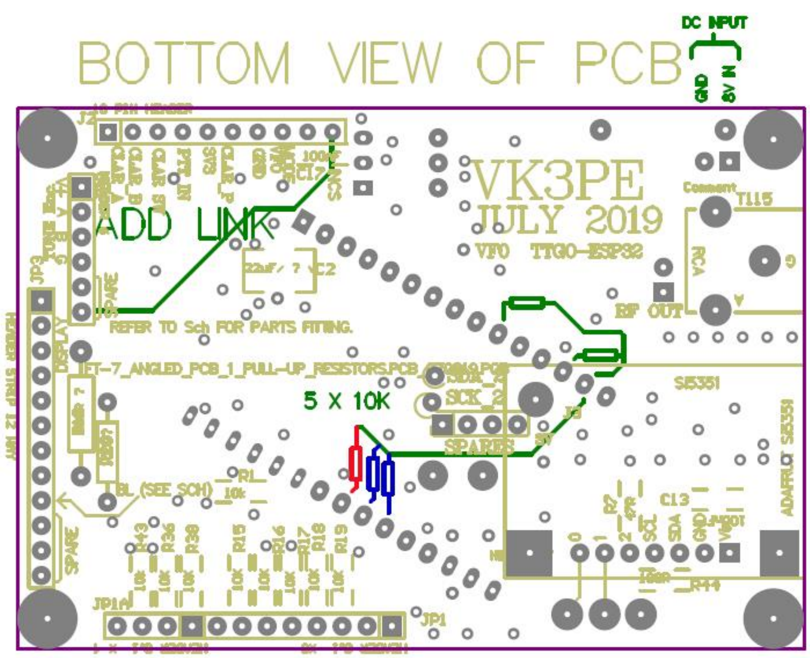
And here are a couple of pictures of the assembled PCB:




All external connections to the frequency encoder, clarifier encoder and the TFT were made using IDC cable with appropriate headers and sockets at each end. The RF output of the Si5351 board uses an RCA PCB mounted socket, to mate with the original FT-7 RCA plug.

Please note, that on the schematic, there are several resistors that might need to be hand fitted onto the PCB, as they were missing on the original prototype. Leaded parts are used and soldered onto the PCB.

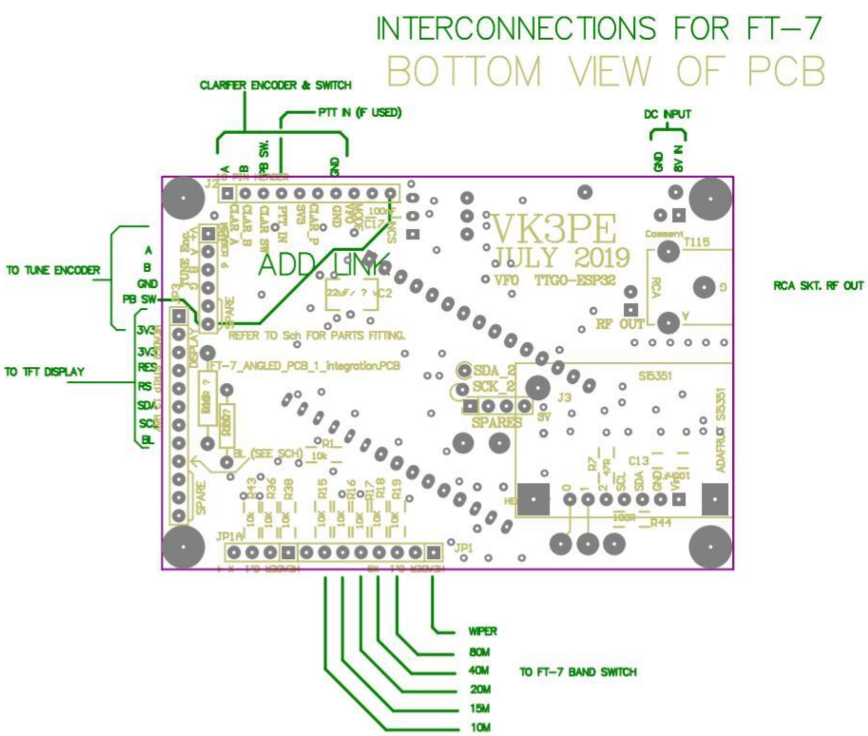


**The two resistors shown in blue are pullups for the clarifier encoder 'A' and 'B' pins; the red resistor is for the clarifier switch. If you don't install a clarifier at all, you won't need any of these three. They also won't be needed if you use an encoder like** the KY-40 **which has pullup resistors built in.**

**If you install the potentiometer type clarifier, you don't need the blue resistors; however you will need the red one.**

**The green resistors are pullups for the I2C bus for the PCF8574. According to** The PCF8574 datasheet they are required however note that the WA2FZW version of the VFO works fine without them.

Also, to make external wiring easier, a single wire must be **placed to connect together J2, pin 10 to JP2, pin 6. This is to make wiring of the Encoder and switch simpler.**



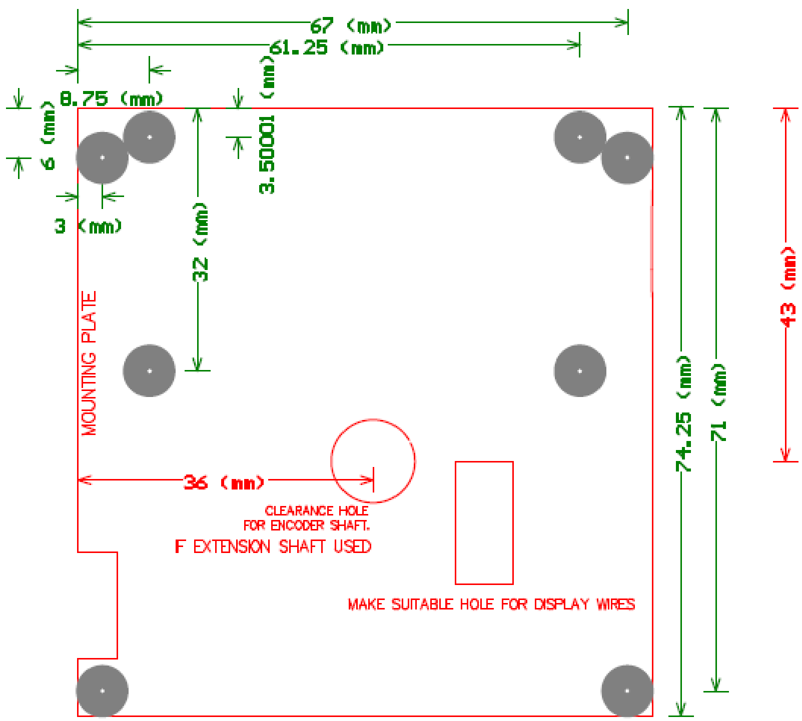

## Power Source

In Glenn's FT-7, the VFO is powered from the radio's power supply. Glenn's PCB has a built in 5 volt regulator to supply the ESP32 5V pin.

## The Display

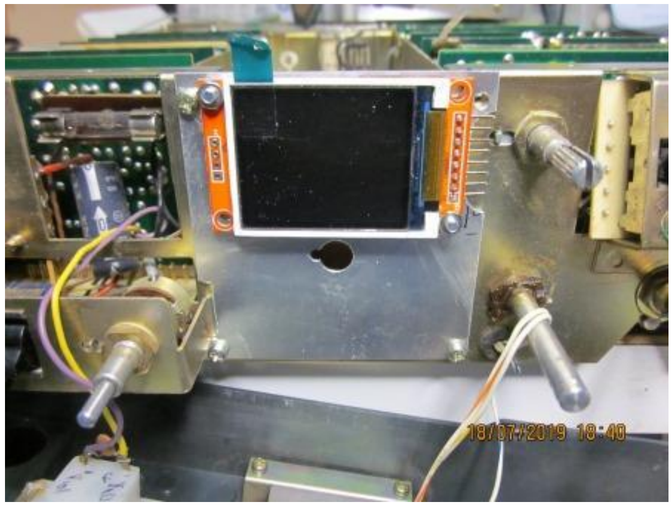
For the FT-7, space is at a premium and only a 1.8” ST7735 based display will fit. The window available in the FT-7 is smaller than the actual display area so a section at the bottom of the TFT will not be visible so is not used (the screen layout can be configured in the software).

A simple bracket of 1.2mm aluminum sheet was cut and is fastened into the FT-7 using existing holes and screws.




MOUNTING PLATE FOR TFT, FT-7

The TFT should be mounted as high as possible. The mounting screws for the TFT are countersunk on the rear of the bracket. Countersunk screws and a 3-4mm thick spacer are also required. Not shown in this picture is a hole to be made to pass through the cabling to the TFT display.




Here’s the display installed in Glenn’s FT-7:


**Frequency Encoder**

In the FT-7, there wasn’t enough room to install one of the high-speed optical encoders (which are rather large) so standard mechanical encoder with push switch was used for tuning.

Glenn originally mounted his encoder onto the same bracket that holds the ESP32 board but it proved difficult to correctly align the encoder plus extension shaft. A better place to mount the encoder is directly into the existing hole in the front of the rig. The large hole will need to be bushed down to suit the actual encoder mounting shaft. The original knob is used as both have a 6mm shaft.

Josep (EA3BDD) has already installed a numerical display into his FT-7 and came up with a better way of installing the frequency encoder:



If access to a lathe is available, a suitable bushing can be turned up.

For the FT-7 the tuning encoder selected has a built in push button switch which is used for the increment button function. The encoder type is similar to that shown for the Clarifier, below, but a non-splined shaft type is used, so the original tuning knob can be used.

### The Clarifier Encoder

The mechanical encoder used for the clarifier has a built-in push button switch which used for the clarifier ON/OFF function.

There is a problem here though in that the mechanical encoders generally available have a much shorter shaft length than the original clarifier potentiometer that is removed. So, a means to extend the shaft is required. Glenn used two encoders, one with a splined shaft to suit the FT-7 knob and the other with a standard shaft. An extension coupling was made from the two encoders. It was made from a piece of 10mm aluminum, but brass could also be used. Access to a lathe will be required or very accurate drilling of a piece of aluminum or brass rod to make the adapter.



### The Si5351 and Low Pass Filter

In the FT-7, the VFO only tunes from 5 to 5.00MHz so a simple LPF is fitted onto the PCB.  The values can be seen in the VK3PE schematic.

Following the filter, a 10dB pad brings the level down to about 120-150mV which is what the mixer expects to see in this rig.

If you are installing this hardware in a different radio with a different VFO frequency range, the values of the components in the low pass filter will need to be changed. Refer to the *Designing the Filters* section above for how to determine the correct component values. You might also have to eliminate or change the 10 dB pad.

## Bill of Materials – WA2FZW Version

The following lists do not include optional parts such as IC sockets, for example.

### Processor Board

| Designation | Description |
|---|---|
| U1 | LCA-110 solid state relay |
| U2 | TTGO T8 V1.7 ESP32 Processor board |
| C1 – C2 | 100nF disc capacitors |
| R1, R2 & R8 | 10KΩ 1/4W resistors |
| R3 | 220Ω 1/4W resistor |
| R4 | 150Ω 1/4W resistor; only required if using the on-board 3.3V to power the display backlight. |
| R5 | 680Ω 1/4W resistor; only required if using the radio’s 12V power to run the display backlight. |
| R6 & R7 | Only needed if your frequency encoder needs pull-ups. Mine didn’t. Determine the values by testing. |
| J1 and J2 | 2 pin male horizontal header; note a single 4 pin header can be used for both. |
| J3 & J7 | 5 pin male horizontal headers. The spacing on the board allows a single 10 pin header to be used for both. |
| J4 | 4 pin male horizontal header. |
| J5 & J6 | 8 pin female stacking headers. |
| GN1 | Hiletgo 240x320 pixel ILI9341 TFT display |

## RF Board

| Designation | Description |
|---|---|
| GN1 | Adafruit Si5351 Clock Generator Module |
| C1 | 100nF disc capacitor |
| C1-0 – C4-0<br>C1-1 – C4-1<br>C1-2 – C4-2 | Values depend on specific design of the low pass filters for your application |
| L1-0 – L2-0<br>L1-1 – L2-1<br>L1-2 – L2-2 | Values depend on specific design of the low pass filters for your application |
| J3 & J4 | 8 pin female vertical stacking headers (tall ones) |
| J1, J2 & J5 | Edge mount SMA female RF connectors |

## PCF8574 Board

| Designation | Description |
|---|---|
| U1 | PCF8574 or PCF8574A |
| C3 | 100nF disc capacitor (C1 & C2 were removed) |
| R1, R2 | 10K 1/8W Pullup resistors for the I2C bus (install only if needed). |
| R? | 10K 1/8W Pull-up or pull-down resistors as needed |
| J3 & J4 | 8 pin female vertical stacking headers (tall ones) |
| J1, J2, J6, J8, J10, J13, J15 & J17 | These aren’t really headers, but that was a convenient way to place the pads on the PCB for any required pull-up or pull-down resistors and the address line connections. |
| J11 | 10 pin male horizontal header. |

### External Components

| Designation | Description |
|---|---|
| Frequency Encoder | Preferably a high speed optical encoder similar to this one however a mechanical encoder could also be used. |
| Clarifier Encoder | Mechanical encoder like the KY-40. |
| Clarifier Potentiometer | Yet to be determined |
| Clarifier Switch | If not built into the clarifier encoder, or if using the potentiometer implementation, a momentary on pushbutton. |
| Function Button | A momentary on pushbutton |
| Frequency Increment **Button** | A momentary on pushbutton |

## Bill of Materials – VK3PE Version

"1206-0805" means any SMD part size; either 1206 or 0805; 0805 is preferred.

Not listed: Header socket material required to make up cables to connect controls, display, switches etc.

| Part Type | Designator | Footprint | Description | Comments |
|---|---|---|---|---|
| 10uF/16v | C1 | 1812 | Polarized cap | |
| 1528pF | C10 | 1206-0805 | | Used 1n5 COG |
| 180pF | C11 | 1206-0805 | | COG (NPO) |
| 100nF | C12 | 1206-0805 | | |

| Part Type | Designator | Footprint | Description | Comments |
|---|---|---|---|---|
| 100nF | C13 | 1206-0805 | | |
| Not fitted | C14 | 1206-0805 | | |
| Not fitted | C15 | 1206-0805 | | |
| Not fitted | C16 | 1206-0805 | | |
| 100nF if Pot | C17 | 1206-0805 | | Only install if the potentiometer type clarifier is used. |
| 100nF | C18 | 1206-0805 | | |
| 100nF | C19 | 1206-0805 | | |
| 22uF/ 16v | C2 | 1812 | Polarized cap | SMD |
| 470nF | C21 | POLY | Polyester cap | Not installed |
| 100nF | C3 | 1206-0805 | | |
| 22uF/16V | C4 | 1812 | Polarized cap | SMD |
| 100nF | C5 | 1206-0805 | | |
| 100nF | C6 | 1206-0805 | | |
| 1087pF | C7 | 1206-0805 | | Used 1nF COG |
| 1087pF | C8 | 1206-0805 | | Used 1nF COG |
| 220uF/35V | C9 | SMD | Polarized cap | Approx. 8.5mm square |
| 1N400x | D1 | Through-hole 0.5” pads | | Any 1A diode |
| 2 HEADER | J1 | 2-pin header; 0.1” spacing | | |
| 10 PIN HEADER | J2 | 10-pin header; 0.1” spacing | | All headers are male 0.1" pitch (2.54mm) |
| HEADER 7 (socket) | J3 | ADAFRUIT_SI5351pcb | | Or the VK3PE version; See the VK3PE web page |
| HEADER 2 | J4 | 2-pin header; 0.1” spacing | | Use locking type |
| "HEADER 0.1"" X 8" | JP1 | 8-pin header; 0.1” spacing | | |
| "HEADER 0.1"" X 4" | JP1A | 4-pin header; 0.1” spacing | | |
| HEADER 6 | JP2 | 6-pin header; 0.1” spacing | | |

| Part Type | Designator | Footprint | Description | Comments |
| --- | --- | --- | --- | --- |
| HEADER STRIP 12 WAY | JP3 | 12-pin header; 0.1” spacing | | |
| 1.3uH | L1 | 1812 | Inductor | Or hand wound on toroid |
| 1uH | L2 | 1812 | Inductor | Or hand wound on toroid |
| 10k | R1 | 1206-0805 | | |
| 100R ? | R10 | Axial resistor - 0.5” pads | See Schematic | |
| 1k2 | R11 | 1206-0805 | | |
| R12 | 1206-0805 | | | Not installed for 3V3; use 4k7 for 5v |
| R13 | 1206-0805 | | | Not installed for 3V3; use 4k7 for 5v |
| 1k2 | R14 | 1206-0805 | | |
| 10K | R15 | 1206-0805 | | |
| 10K | R16 | 1206-0805 | | |
| 10K | R17 | 1206-0805 | | |
| 10K | R18 | 1206-0805 | | |
| 10K | R19 | 1206-0805 | | |
| 75R | R2 | 1206-0805 | | |
| 100R ? | R20 | Axial resistor – 0.4” pads | | |
| 1k | R21 | 1206-0805 | | |
| 0R | R22 | 1206-0805 | | |
| R23 | 1206-0805 | | | Not installed |
| 10k | R24 | 1206-0805 | | |
| 10K | R25 | 1206-0805 | | |
| 10K | R26 | 1206-0805 | | |
| 10K | R28 | 1206-0805 | | |
| 10K | R29 | 1206-0805 | | |

| Part Type | Designator | Footprint | Description | Comments |
|---|---|---|---|---|
| 100R | R3 | 1206-0805 | | |
| 10K | R30 | 1206-0805 | | |
| 10K | R36 | 1206-0805 | | |
| 10K | R38 | 1206-0805 | | |
| 0R | R39 | 1206-0805 | | |
| 100R | R4 | 1206-0805 | | |
| 0R | R40 | 1206-0805 | | |
| 10k | R43 | 1206-0805 | | |
| 100R | R44 | 1206-0805 | | |
| 100R | R45 | 1206-0805 | | |
| 1k | R5 | 1206-0805 | | |
| 1k | R6 | 1206-0805 | | |
| 47R | R7 | 1206-0805 | | |
| 2K2 | R8 | 1206-0805 | | |
| 10k | R9 | 1206-0805 | | |
| TFT_1.8 | TFT1 | TFT_1.8 | | eBay; See the VK3PE web page |
| ESP32_WROVER (TTGO T8 V1.7) | U2A | esp32_wemos | | eBay, Newegg, Amazon, AliExpress |
| 78M05 | U3 | DPAK-3 (SMD) | Regulator IC | |
| 4N28 (OPTO COUPLER) | U4 | 6-pin DIP | | |
| PCF8574DW | U5 | SOIC-16 | | |
| PCB V1.0 | | | | See the section titled *The VK3PE Circuit Board* |

| Part Type | Designator | Footprint | Description | Comments |
|---|---|---|---|---|
| 10K AXIAL RESISTORS | need 6 maybe | | | Missing pullup resistors, see the section titled *The VK3PE Circuit Board* |

## Coming Attractions

There are 2 hardware modifications coming soon.

Glenn has made some minor changes to the FT-7 PCB to incorporate some of the modifications that were made during testing of the original software and hardware. When finished, we will publish another document describing the changes.

As I'm old and stupid, I designed my original boards so they wouldn't fit in the radio, thus I had to design smaller PCBs to fit in the Swan-250C. I also made a couple of changes to the hardware on the processor board to improve the PTT/Keying circuitry and added the capability to power the unit from the radio instead of the USB port. I will publish information about those PCBs shortly.

## Suggestion Box

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