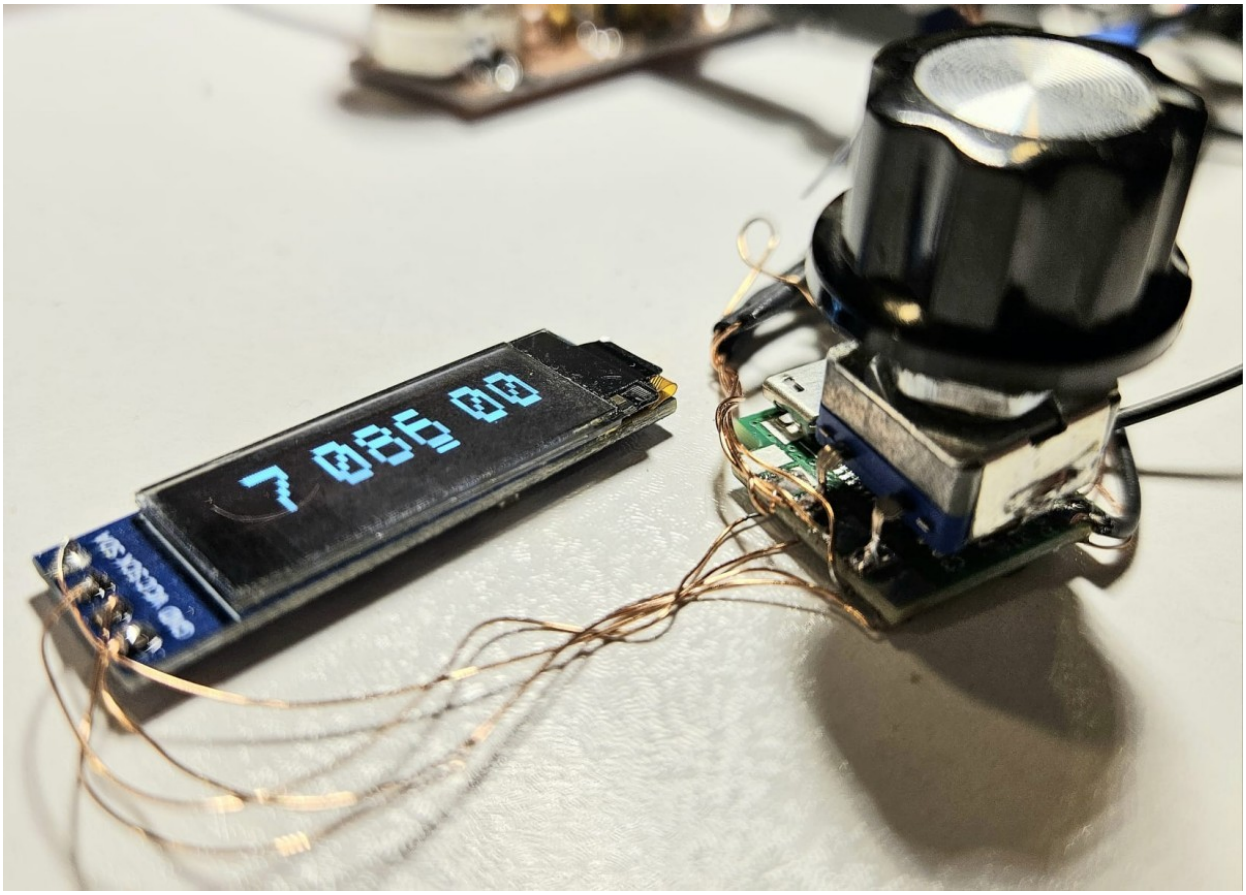
QRP Labs

DigiVFO

DigiVFO: Miniature digital VFO and BFO



Contents

1. Introduction

DigiVFO is a miniature but highly functional VFO using the MS5351M synthesized clock generator, and is based on the ProgRock2 module already produced by QRP Labs. Different firmware, and the addition of a standard rotary encoder and 0.91" I2C OLED module, facilitate the DigiVFO.

- Tiny size PCB, a little smaller than an HC6 crystal: 0.725 x 0.675 inches (18.4 x 17.1mm).
- Factory assembled PCB, ready-to-use: assembly is just connection of the rotary encoder and OLED module.
- 0.25ppm TCXO reference as standard; typical drift and accuracy will be less than 1Hz at 40m band, for example.
- Output frequency range is configured to cover 1.000000 MHz to 99.999999 MHz.
- Supports direct conversion receivers, including with 90-degree quadrature phase LO
- Supports superhet with low-side or high-side LO injection
- For superhet radios, also generates the BFO with configurable frequency
- Configurable CW receive offset (CW or CW-R), if you ground an input pin to indicate you are in CW mode
- RIT (Receive Incremental Tuning)
- VFO A / B / Split mode
- CAT control interface via built-in USB to Virtual COM Serial port (micro-USB connector)
- CAT via separate USART port for 3.3V logic level serial (e.g. other microcontroller)
- QRP Labs Firmware Update (QFU) bootloader for firmware updates

PLEASE READ THE ENTIRE MANUAL USE INSTRUCTIONS VERY CAREFULLY BEFORE APPLYING POWER TO THE BOARD!

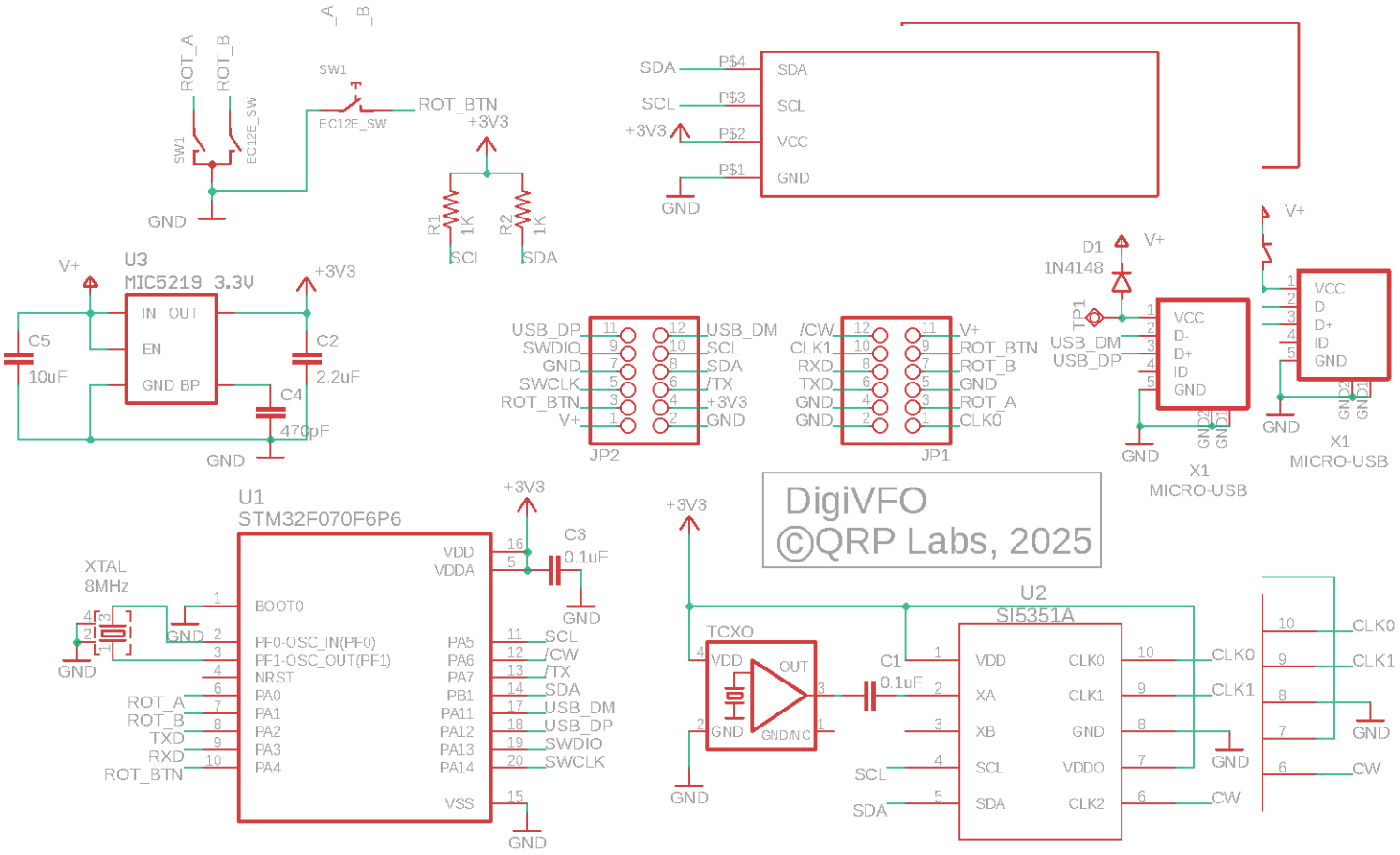
2. Design

This is a very simple circuit consisting of:

- STM32 Cortex M0 CPU controller
- MS5351M triple clock generator
- 25MHz 0.25ppm TCXO reference (Temperature Controlled Crystal Oscillator)
- MIC5219 3.3V voltage regulator
- Micro-USB connector
- Pads for other connections
- Rotary encoder
- 0.91" OLED module

All the SMD components are pre-assembled at the PCB factory onto the tiny PCB. The constructor only needs to solder on the rotary encoder and 4 wires to the OLED module.

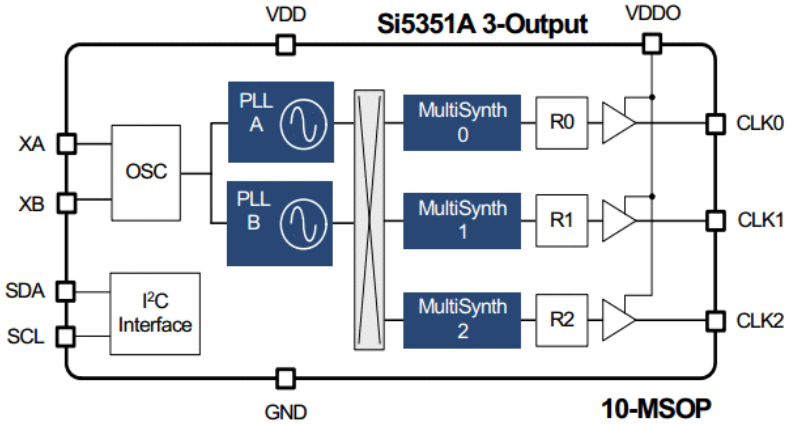
The DigiVFO schematic is shown below.



All the SMD components are pre-assembled at the PCB factory onto the tiny PCB. The constructor only needs to solder on the rotary encoder and 4 wires to the OLED module.

The Si5351A synthesizer chip is now used in many other QRP Labs products such as the QCX CW transceiver series and QDX digital transceiver. This is a Digital Phase Locked Loop (PLL or

DPLL) synthesizer which provides three separate frequency outputs, each having a frequency range spanning 3.5kHz to 200MHz. The frequency stability is governed by the a crystal reference.



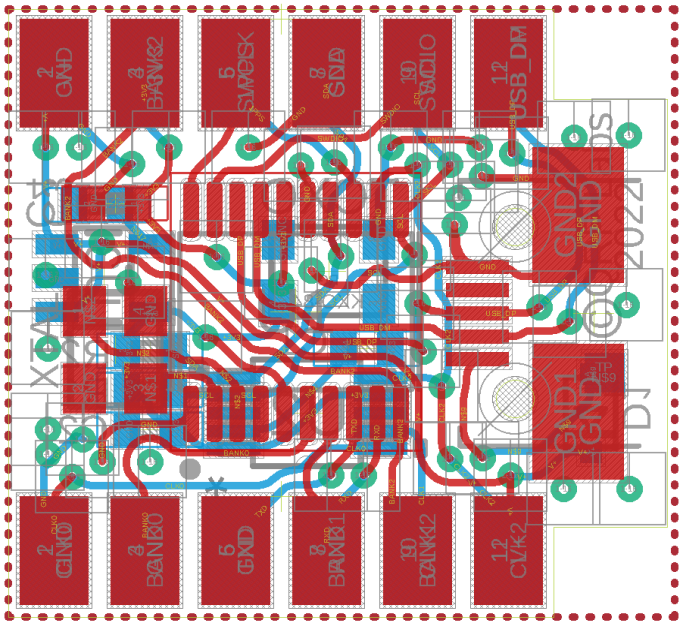
Due to unavailability of the Si5351A, the equivalent MS5351M may be used. For test details on the performance of Si5351A vs MS5351M demonstrating the suitability of the MS5351M (in fact, slight superiority in many regards), please see http://qrp-labs.com/synth/ms5351m.html

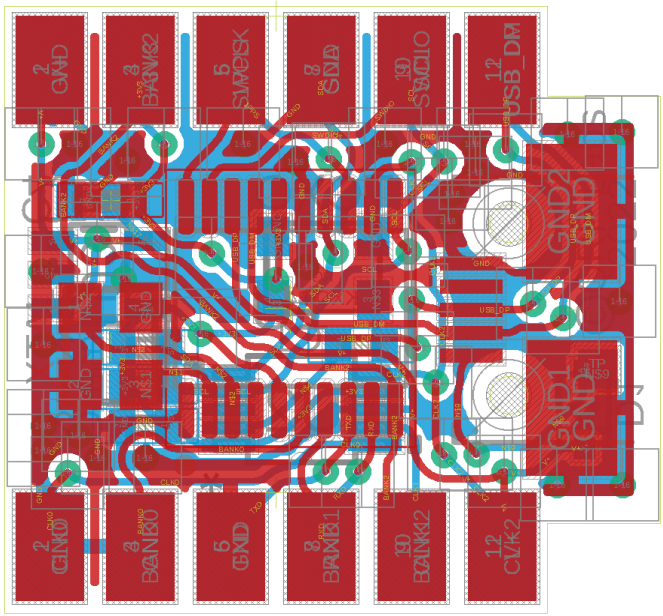
The block diagram (right) is taken from the SiLabs Si5351A datasheet. Briefly, the 27MHz reference oscillator is multiplied up to an internal Voltage Controlled Oscillator in the range 600-900MHz (the PLL), then divided down to produce the final output frequency. The multiplication up and the division down are both fractional and so the frequency resolution is extremely finely controlled. The chip has two PLLs and three output divider units. The chip must be configured using its serial I2C interface. R1 and R2 are 1K pullup resistors required for the I2C bus.

For high frequency stability, a 0.25ppm TCXO is used (the same TCXO as used in QCX-series CW transceivers and QDX digital transceivers).

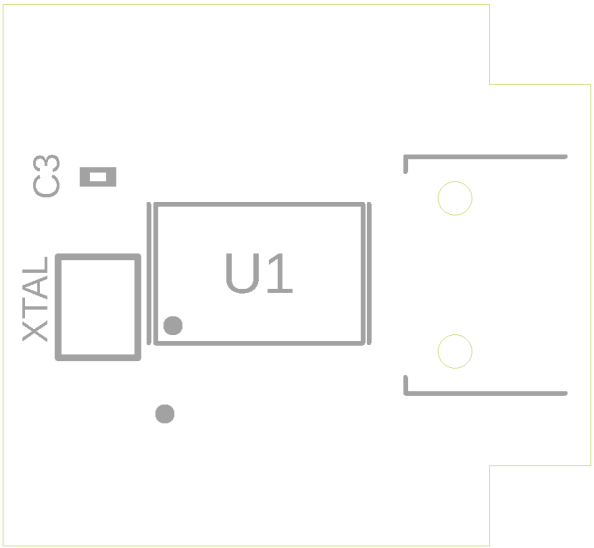
The Si5351A does not preserve its configuration registers through a power cycle. To provide persistent storage of parameters, the microcontroller needs to retain the configuration parameters in non-volatile storage. STM32-series microcontrollers do not have onboard EEPROM and to save parts count and board area, no additional I2C EEPROM chip has been used in DigiVFO. Instead, the top 1KByte sector of the Flash memory of the STM32 is used as a non-volatile storage area for the configuration parameters.

PCB Trace diagrams (Left: without groundplanes; Right: with groundplanes)

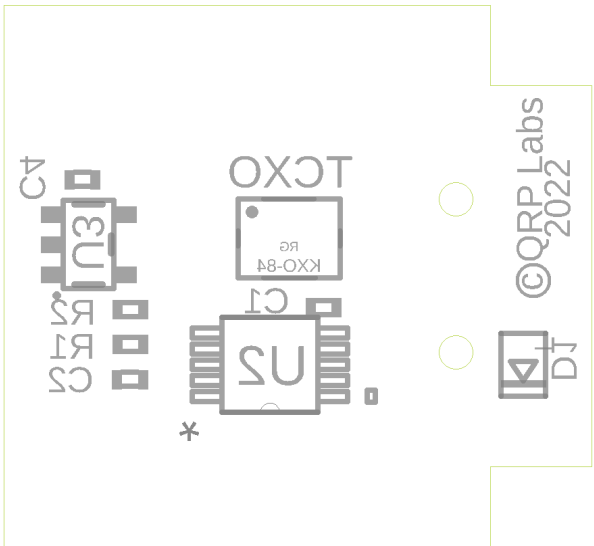




PCB component layout, top side:

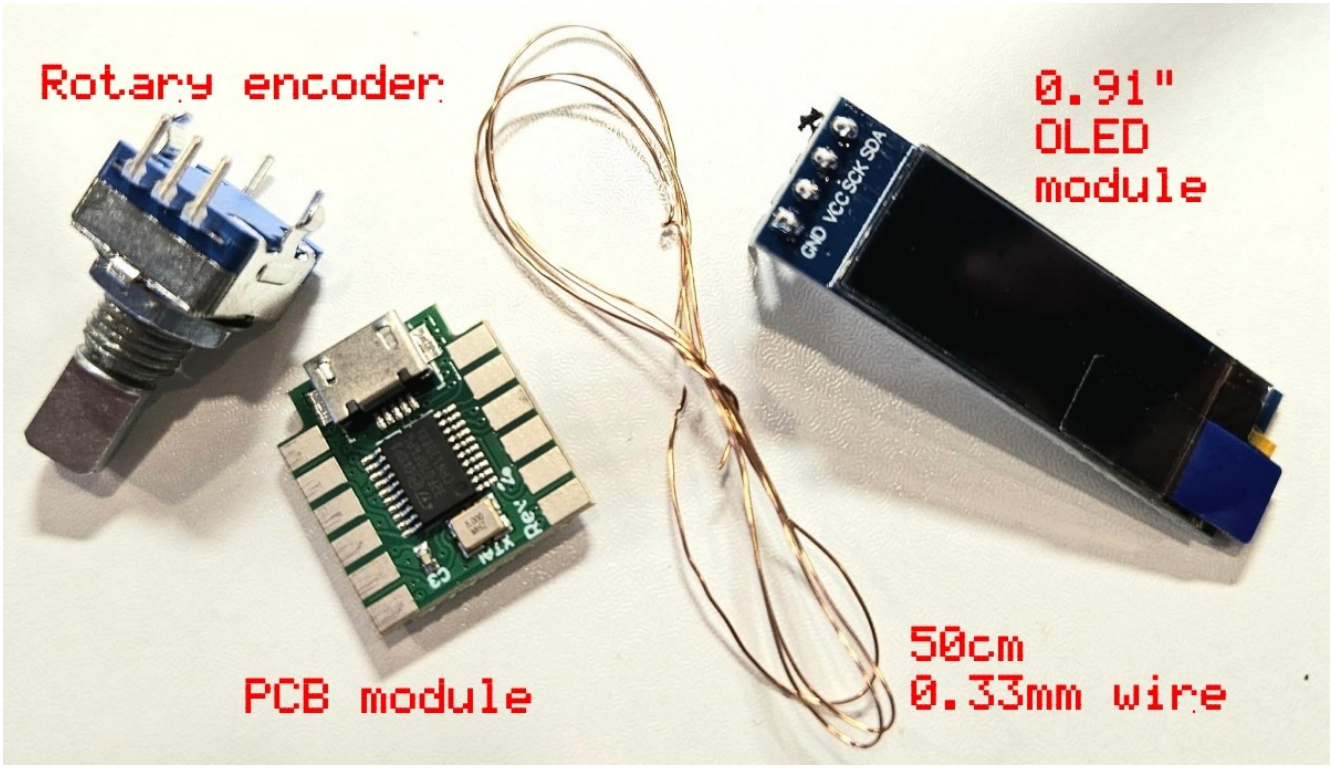


PCB component layout, bottom side:



Kit Contents:

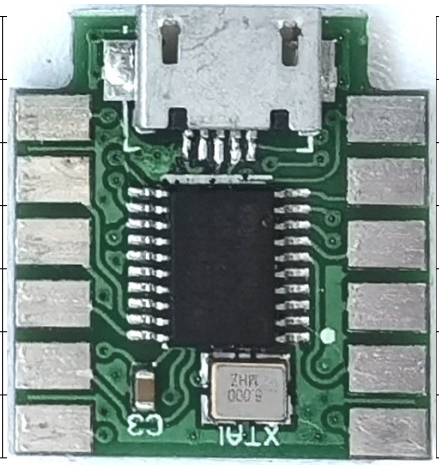
- 18.4 x 17.1mm PCB
- Rotary encoder
- 0.91” OLED display
- 50cm 0.3mm wire for connecting PCB to display



3 Connections

This table shows the pinout of the DigiVFO module:

Bottom	Top		Top	Bottom
13. USB -	1. USB +		12. V+	24. / CW
14. SCL	2. SWDIO		11.ROT_BTN	23. CLK 1
15. SDA	3. GND		10.ROT_B	22. RXD
16. / TX	4. SWCLK		9. GND	21. TXD
17. +3V3	5. ROT_BTN		8. ROT_A	20. GND
18. GND	6. V+		7. CLK 0	19. GND



You will note that there is a row of 6 pads along each long edge of the PCB, on top and bottom sides, making a total of 24 possible connections. Some, such as GND, are on multiple pads. For convenience these are labeled 1 to 24. In the diagram above, the pads on the top side are shown closest to the PCB pads, and the bottom side pads are the outer table column.

The layout was carefully designed to allow as much future flexibility as possible.

The signals are as follows:

Signal	Pins	Description
GND	3, 9, 18, 19, 20	Ground.
V+	6, 12	Positive supply voltage.
+3V3	17	3.3V output from onboard voltage regulator.
USB-, USB+	13, 1	USB port (also connected to USB-C/micro-USB connector); note, also sometimes called USB_DM, USB_DP respectively.
SWDIO, SWCLK	2, 4	Chip programming pins: Factory use only.
SCL, SDA	14, 15	I2C serial bus – MS5351M and OLED communication
RXD, TXD	22, 21	USART port
ROT_A	8	Rotary encoder direction pin A
ROT_B	10	Rotary encoder direction pin B
ROT_BTN	5, 11	Rotary encoder button pin
CLK 0	7	MS5351M Clock 0 output.
CLK 1	23	MS5351M Clock 1 output.
/ CW	24	Input pin: low on this pin signals CW mode, and the CW receive offset is added/subtracted (CW/CW-R dependent). If unconnected, an internal pull-up signals non-CW mode.
/ TX	16	Input pin: low on this pin indicates Transmit; this is used in the LO calculation based on RIT or VFO A/B/Split modes. If unconnected, an internal pull-up signals Receive mode.

Supply voltage:

DigiVFO current consumption is approximately 35-40 mA. It does vary a little depending on MS5351M output loading, the configured outputs and their frequency.

The MIC5219 datasheet specifications for voltage and power dissipation need to be observed. According to the MIC5219 datasheet the input voltage range, for a 3.3 V 50 mA output is +3.5 to +12 V.

At a +12 V supply and 30 mA current consumption the power dissipation in the voltage regulator will be 261 mW. According to the MIC5219 datasheet this level of power dissipation is acceptable. Therefore even a +12V supply may be used. However, 261 mW is quite a lot of power dissipation (and hence heating) so you may wish to bear that in mind and use a lower supply voltage if possible.

The minimum acceptable supply voltage is determined by the dropout voltage graph in the MIC5219 datasheet which for 30mA load is approx 100mV. I'd suggest allowing a little safety margin and supply DigiVFO with at least 3.5V.

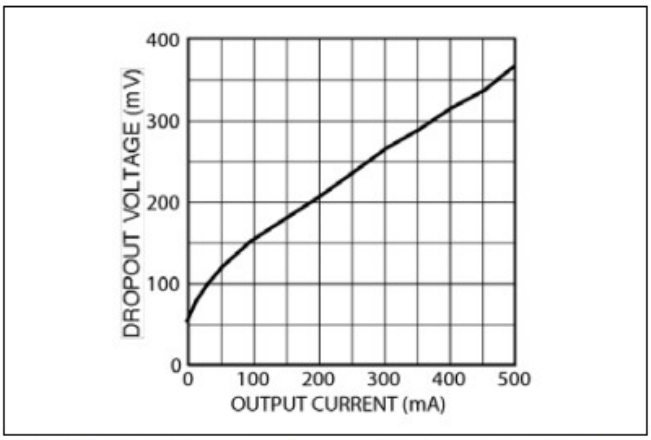


FIGURE 2-11: *Dropout Voltage vs. Output Current.*

Care should be taken when powering DigiVFO directly from the USB cable (see below), and using a power supply connected to +V at the same time. The USB cable +5V will power DigiVFO via an onboard diode, resulting in about 4.4V supply to DigiVFO. If you have connected an additional external supply voltage to DigiVFO, and that is less than 4.4V, then your external supply will fight with the USB voltage, potentially drawing excessive current through the onboard diode.

3.3V output

The regulated 3.3V output from the onboard voltage regulator is provided on one of the pads. If you use this, please be sure to observe all MIC5219 datasheet specifications regarding loading etc.

WARNING

Processor I/O pins should not be connected to a voltage higher than 3.3V – including USART pins.

Clock outputs

CLK 0, CLK 1 are two of the three clock outputs from the MS5351M synthesizer. They are unbuffered, direct connections to the MS5351M chip. As such, they should be connected with care, so as not to damage the MS5351M chip.

The outputs are 3.3 V peak-to-peak squarewaves, with a declared output impedance in the MS5351M datasheet of 50-ohms. What this appears to mean in practice is that if you connect a 50-ohm load, the output will be reduced by 50%, to 1.65 Vpp.

For best phase noise performance, as well as least crosstalk between MS5351M outputs, it is recommended to use loads of at least 1 K-ohm.

USB connection

DigiVFO has a USB-C or micro-USB connector for connecting to a host PC terminal emulator, for CAT control of DigiVFO (see section below for CAT command details).

+5V power may also be supplied by the USB-C / micro-USB connector. There is a diode feeding the +5V connection of the USB cable to the +V supply voltage of the DigiVFO. So if you connect only a USB cable, DigiVFO will be supplied from that. If you are supplying an external voltage below 4.4V damage can occur, please see section above regarding power supply.

The USB connections are also available as pads on the edge connectors.

Other connections

SWDIO, SWCLK are In-Circuit-Programming connections used only by the factory during initial download of the bootloader into the DigiVFO module. They have no further use.

SCL, SDA are the I2C bus, used for processor communication with the Si5351A and the OLED module.

ROT_A, ROT_B and ROT_BTN are the connection points for the rotary encoder, which may be soldered directly to the DigiVFO board or on (short) wires.

TXD, RXD are a USART transmit/receive signal pair that may be used as a CAT interface. See section below for more information on the USART.

Rotary encoder installation

The rotary encoder has two pins on one side, and three on the other. The two-pin side are the connection to the buttons. The three-pin side is the connection to the rotary encoder quadrature output. The center pin is ground, the outer pins are the quadrature signal which allows the microcontroller to work out whether you are tuning clockwise or anticlockwise and by how many steps.

Bottom	Top
13. USB -	1. USB +
14. SCL	2. SWDIO
15. SDA	3. GND
16. / TX	4. SWCLK
17. +3V3	5. ROT_BTN
18. GND	6. V+

Top	Bottom
12. V+	24. / CW
11.ROT_BTN	23. CLK 1
10.ROT_B	22. RXD
9. GND	21. TXD
8. ROT_A	20. GND
7. CLK 0	19. GND

The rotary encoder can be connected by wires; or simply the PCB can be soldered directly to the PCB. The pads (see yellow-shaded pads in the diagram below) are perfectly placed for the rotary encoder pins to be soldered directly to them (yes the original ProgRock2 was designed with this in mind from the start).

In the diagram "Top" refers to the side of the board having the USB connector and the 20-pin microcontroller.

Connecting the OLED

Use four thin wires – with a length appropriate to your installation intentions – to connect the OLED to the PCB. The four wires are connected to pads on the LEFT side of the board, on the UNDERside of the board. Which is to say, the side opposite the large 20-pin microcontroller.

Bottom	Top
13. USB -	1. USB +
14. SCL	2. SWDIO
15. SDA	3. GND
16. / TX	4. SWCLK
17. +3V3	5. ROT_BTN
18. GND	6. V+

Top	Bottom
12. V+	24. / CW
11.ROT_BTN	23. CLK 1
10.ROT_B	22. RXD
9. GND	21. TXD
8. ROT_A	20. GND
7. CLK 0	19. GND

Connect the four wires to the corresponding pads on the OLED module. Note that the OLED silkscreen pins are on the left side and are labeled, as per the photograph below, from top to bottom: SDA, SCK, VCC and GND. The OLED pin labeled SCK is to be connected to the “SCL” pad of the PCB. The OLED pin labelled VCC is to be connected to +3V3 on the PCB.

Connections to the radio transceiver

Connecting the VFO output to the transceiver will be different depending on the transceiver design. Remember that the output of this digital VFO is a 3.3V squarewave. It is recommended to drive it into loads of at least 1K resistance.

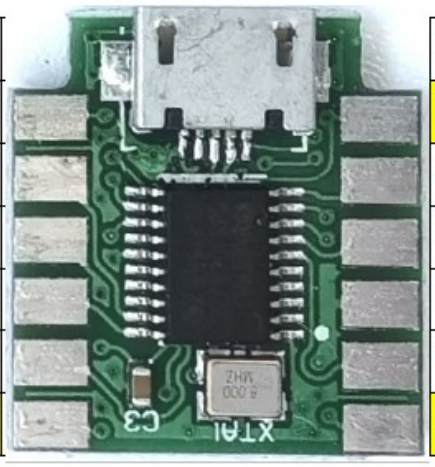
For a superhet:

- CLK0 = VFO output
- CLK1 = BFO output (if required)
- Use mode “LOW” or “HIGH” (low injection VFO or high injection VFO)
- Set BFO frequency in the configuration to the actual IF frequency

For a direct conversion receiver, or phasing QSD receiver:

- CLK0 = VFO output at operating frequency
- CLK1 = VFO output with 90-degree phase shift (for use in Quadrature Sampling Detectors)
- Use mode “QSD”

Bottom	Top
13. USB -	1. USB +
14. SCL	2. SWDIO
15. SDA	3. GND
16. / TX	4. SWCLK
17. +3V3	5. ROT_BTN
18. GND	6. V+



Top	Bottom
12. V+	24. / CW
11.ROT_BTN	23. CLK 1
10.ROT_B	22. RXD
9. GND	21. TXD
8. ROT_A	20. GND
7. CLK 0	19. GND

Connect the positive supply voltage to one of the pins labeled V+ (pin 6 or pin 12), and ground to one of the spare GND pins (pin 19 or pin 20). NOTE the comments in the previous section about supply voltage ranges.

/TX pin: If you want to be able to use some of the features, specifically the RIT (Receive Incremental Tuning) and Split VFO feature (RX on VFO A, TX on VFO B) then the DigiVFO needs your transceiver to tell it when it is in the transmit state (keydown, PTT active). A low-signal on the /TX pin tells DigiVFO you're in TX mode and it will alter the VFO frequency accordingly if necessary (if RIT or Split is active). There is a weak pull-up resistor so if unconnected, DigiVFO assumes "Receive". Note that the /TX pin should NOT be fed more than 3.3V.

/CW pin: If you want the DigiVFO display to correctly indicate the actual CW transmission frequency, as is conventional for amateur radio transceivers, then it has to apply a 700 Hz (or whatever you configured) offset to the displayed frequency during Receive (or always, if you have a tone-injection CW generation method – see configuration settings). The /CW pin is how you inform DigiVFO that you are in CW mode. It is grounded to indicate CW mode. If unconnected, a weak pull-up resistor causes it to assume non-CW mode and the CW Offset will never be applied.

4 Operating instructions

User interface

DigiVFO is a minimalist VFO module. The user interface consists of the single rotary encoder, which also contains a push-button in its shaft. This combination of Clockwise, Anticlockwise and Press actions is used to construct the entire user interface. This necessarily means accessing some functions requires different types of press-and-rotate operations. Remember also that this microcontroller is a very inexpensive low-capability member of the STM32 family, and only 21 KBytes are available to the application firmware (10K for bootloader, 1K for configuration storage).

The user interface has been designed to be as simple as possible, under the constraints of this single control.

The push-button has three types of press:

- Short click: a momentary press lasting less than 220 milliseconds
- Long press: a long press lasting more than 220 milliseconds but less than 900 milliseconds
- Very long press: a very long press lasting more than 900 milliseconds

For copying VFO A to B and other multiple VFO copy operations, a combination of press-and-rotate is used.

At power-up

On power-up, the DigiVFO screen displays the current operating frequency. The default frequency at power-up is a configuration menu item (more on this, later).

The frequency is displayed to 10 Hz resolution in a large, easy-to-use font.

The underlined digit indicates the current tuning step. Therefore in the above example, the tuning step is 100 Hz. Each click of the rotary encoder, will alter the frequency by 100 Hz.

Press the rotary encoder button (short click) to cycle the tuning step between 10 Hz → 100 Hz → 500 Hz → 1 kHz. When the tuning rate is 500 Hz, the underline appears under the gap between the 1kHz digit and the 100 Hz digit.

After changing the tuning step, if the tuning step is in the upward direction (e.g. 10 Hz → 100 Hz) at the next click of the rotary encoder, the frequency will "snap" to the nearest 100 Hz (in this case) increment.

Configuration menu

A very long press (longer than 900ms) enters the configuration menu.

The configuration menu contains the settings that are saved in the device at power-off, such as the CW offset, BFO frequency, and type of VFO.

You may use the rotary encoder now to scroll back and forth between the configuration parameters.

To edit a parameter, press the rotary encoder button (short press). An underline appears under the digit being edited (or the leftmost character of a text selection). Then use the rotary encoder to adjust the value. Press the button again to move to the next right digit. When the cursor disappears off the right side of the number, editing is complete and you are back to the menu in parameter VIEW mode.

When editing a text selection, such as the example image above, use the rotary encoder to select the desired value then a single button press to exit back to view mode. The list does not "wrap" around at either end; so for example when there are three values, continued clockwise rotation stops at the highest value.

When you wish to exit the Configuration menu, apply a long-press to the button. The configuration settings are saved to non-volatile (Flash) memory when you exit the menu in this way. If you happened to power down DigiVFO without exiting the menu, your changes would NOT be saved.

The list of configuration parameters now follows.

TYPE

This is the type of VFO being implemented (see example image above). There are three options:

- LOW: Low-side injection for a superhet transceiver. The LO is calculated as the difference between the BFO frequency and the displayed operating frequency. The LO is output on Clk0 and the BFO is output on Clk1.
- HIGH: High-side injection for a superhet transceiver. The LO is calculated as the sum of the BFO frequency and the displayed operating frequency. The LO is output on Clk0 and the BFO is output on Clk1.
- QSD: Direct LO output (at the displayed frequency exactly) for a direct conversion transceiver. The LO is output on Clk0. A 90-degree phase offset signal is output on Clk1, facilitating easy use with a Quadrature Sampling Detector in a phasing method unwanted sideband cancellation receiver or SDR. Note that the minimum frequency for a quadrature output is 3.5 MHz.

BFO

This is the BFO frequency in an superhet transceiver. In LOW and HIGH superhet VFO modes, this is subtracted or added to the displayed frequency to get the LO frequency which is output on Clk0. The BFO frequency is output on Clk1. Remember to press the button once to start editing the value, then press to move the underline cursor right along the number.

START

This is the frequency used to set the operating frequency at power-up.

CW OFFSET

When the CW input signal is asserted (taken low, a.k.a. grounded) to indicate CW mode, this is the offset which is applied to the operating frequency during Receive. Or always (see CW TONE).

CW R

The CW R (CW Reverse) parameter should be set to YES when your transceiver is operating CW-R mode (CW Reverse). Conventionally CW is operated as Upper Sideband (USB). If your transceiver is using LSB for CW, you should set CW R to Yes.

CW TONE

CW can be generated in a transceiver by direct synthesis of the CW carrier frequency, for example in a superhet, the mixer may be unbalanced to generate the carrier, so the VFO + BFO = CW carrier frequency. In this case, the CW OFFSET should be applied to the VFO during Receive, but during Transmit the VFO to generate the actual required carrier should be generated. The /CW pin must be connected to inform DigiVFO that CW mode is active, and the /TX pin should be activated on Transmit to tell DigiVFO the offset should be applied.

Another way to generate CW is to generate a sinusoidal 700 Hz tone, and apply it as the audio modulating input to an SSB Exciter. Transceiver convention is that in CW mode, the displayed frequency is the actual transmit frequency. Therefore a CW Offset should be applied ALL the time (in Receive AND Transmit).

For a direct synthesis CW transceiver set CW TONE to NO. For a tone injection transceiver set CW TONE to YES. The /CW pin should be activated (grounded) to indicate CW mode, so that the CW OFFSET is applied to the display frequency.

FIRMWARE UPDATE

This configuration setting is the method by which a firmware update is triggered. Normally this setting is always set to NO. If you set it to YES then exit the menu system (to save the parameters), you will trigger the firmware update procedure. See Firmware update section below for more details.

RIT Mode

RIT is Receiver Incremental Tuning. When in the normal operating mode, a long-press of the button causes the DigiVFO to enter RIT adjustment mode. The display looks something like this example (right).

The RIT adjustment range is -5,000 to +5,000 Hz with a 1 Hz resolution. Just like the main tuning display, the underline indicates which digit will be adjusted. Pressing the button (short press) moves the underline from 1 Hz → 10 Hz → 100 Hz → 1 Hz etc.

RIT adjustments are effective immediately. When you wish to return the main tuning mode, use a long-press of the button. This returns to the main tuning display mode.

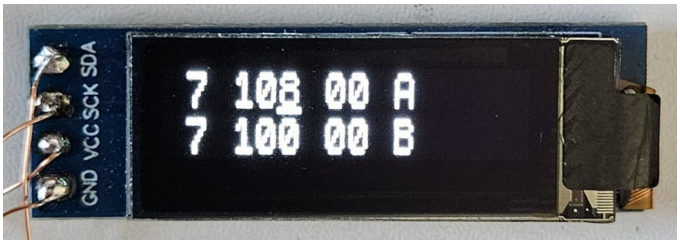
If RIT is non-zero, this is indicated by a little R between the 1 kHz and 100 Hz digits as illustrated in this photograph (right).

VFO A / B / Split

An alternative to RIT is to use VFO A / B / Split modes. Now to enter the A / B / Split mode, first enter RIT mode (long press) then do a very long press to enter A / B / Split mode.

The mode initially looks like this photograph (right). Again there is a cursor under the tuning step digit and the rotary encoder can be used for tuning.

Between the 1 MHz digit and the 100 kHz digit there is an ‘A’ or ‘B’ which indicates A or B mode (in these two modes, the indicated VFO is used for both Receive and Transmit operations).



A long-press of the button changes the mode, A → B → Split → A → B → Split → A etc in that cycle.

In the “Split” mode, both A and B VFO frequencies are displayed. Only the A frequency may be adjusted, using the rotary encoder in

tuning steps indicated by the underline cursor. VFO A is used for Receive, VFO B is used for Transmit.

There are also functions to copy the VFO A to B frequencies etc. This is achieved by rotating the encoder at the same time as pressing it. So press the button, then start rotation (within 900 milliseconds). This is actually surprisingly intuitive and easy. There are three functions:

- Press-then-rotate-clockwise: Copies VFO A to B
- Press-then-rotate-clockwise: Copies VFO B to A
- Press-then-jiggle-clockwise/anticlockwise: Swaps VFO A and B

To EXIT the VFO A / B / Split mode and get back to ordinary plain VFO mode, do a very long button press.

5 USB Virtual COM port

DigiVFO has a USB-C or micro-USB connector onboard, and provides a Virtual COM serial port for CAT control. No USB to Serial converter dongle is required.

Port names

The port naming differs on Linux and Windows. You need to find out the correct port to connect to, to be able to properly configure whatever PC software you are using for CAT control.

On windows systems, serial ports are called COM1, COM2, COM3 etc but it is not always easy to know which COM port belongs to your specific device (DigiVFO in this case).

On Linux systems, the serial port is called dev/ttyACM0 or dev/ttyACM1 etc with the rightmost digit starting at 0 then increasing by 1 for every “ACM” type serial device attached to the computer.

Drivers

No additional drivers are required for operation with most Linux distributions, Apple Mac or MS Windows 10 or Windows 11.

For older versions of MS Windows, it may be necessary to install a driver for the serial port because this driver is not on your computer already by default. This driver is available from the ST Semiconductor website at https://www.st.com/en/development-tools/stsw-stm32102.html and is applicable to 98SE, 2000, XP, Vista®, 7, and 8.x Operating Systems. There is a description for installation on Windows 7/8 on the QRP Labs QLG2 page http://qrp-labs.com/qlg2 so if in doubt, please check this.

Linux special note

On Linux systems, a particular problem can occur. When the DigiVFO (Serial) connection is detected, the PC thinks that a modem has been connected and starts trying to send it Hayes AT-commands dating back to 1981, implemented on Hayes’ 300-baud modem. Yes! 40 years ago…

The Operating System attempting to send AT commands to your DigiVFO will certainly mess everything up. Not least because when DigiVFO receives a carriage return character, it will enter Terminal Applications mode; this will send all sorts of characters back to the PC (as DigiVFO

thinks it is now talking to a terminal emulator) and it will disable CAT command processing, so your PC digi modes software will not be able to talk to DigiVFO. Disaster.

To fix this you need to issue the following commands to disable ModemManager:

```
sudo systemctl stop ModemManager
sudo systemctl disable ModemManager
sudo systemctl mask ModemManager
```

This will permanently stop ModemManager. If for some reason, you actually DO need ModemManager operational, for some other reason… well there IS a way to stop it just for DigiVFO… but Google will be your elmer on this!

Additional information from Greg Majewski:

There is another Linux service, BRITTY, that does the same. BRITTY is a Braille service for access by sight impaired people. I have encountered the problem with the G90 and Ubuntu on a laptop (Ubuntu full version), Raspberry Pi 3 with Raspberry OS and the Orange PI 800. Here are commands that remove BRITTY:

sudo systemctl stop brltty-udev.service

sudo systemctl mask brltty-udev.service

note output: Created symlink /etc/systemd/system/brltty-udev.service

→ /dev/null.

sudo systemctl stop brltty.service

sudo systemctl disable brltty.service

These commands are similar as used for Modem Manager service.

5 USART port

CAT commands may also be sent/received using the onboard USART port. This could be useful if integrating DigiVFO into a larger system with a host microcontroller. Remember that the indicated TX and RX lines should be swapped for connection to your microcontroller! Your microcontroller's TX line goes to the DigiVFO RX line, and the microcontroller's RX line should be connected to the DigiVFO's TX signal. The baud rate is 19,200 and 3.3V signal logic should be used.

6. CAT command protocol

CAT (Computer Aided Transceiver) is a standard serial interface which allows computer control of your radio transceiver. Many software applications such as WSJT-X, JS8Call and logging applications etc use CAT to retrieve or control the VFO frequency etc.

The DigiVFO CAT command protocol is similar to that used in the QRP Labs QCX-, QDX- and QMX-series transceivers. Originally the CAT commands were based loosely on the Kenwood TS-480 transceiver, which is well supported in all software, and well documented (see https://www.kenwood.com/i/products/info/amateur/ts_480/pdf/ts_480_pc.pdf) – also similar to TS-

440. Subsequently QRP Labs transceivers now have their own entry in the hamlib library transceiver dropdown. Additional QRP Labs specific commands were added.

CAT commands never contain a carriage return or linefeed character. They are alwasys terminated by a semicolon.

DigiVFO CAT commands can be applied either via the USART port or the USB Virtual COM Serial port. Command responses are sent to the same port that issued the command.

DigiVFO implements a subset of the Kenwood TS-480/TS-440 CAT command set which is an old standard, but contains all the necessary commands for WSJT-X and other software to be able to control the VFO, and is old enough that it is widely supported by most software packages.

DigiVFO provides a standard set of two-character CAT commands, which are similar to the Kenwood ones, and which act on the transceiver, such as changing the VFO frequency or RIT. These commands have a temporary effect only on the transceiver, the values are not saved stored in non-volatile storage.

The following lists the commands and responses in alphabetical order:

FA: Get/Set VFO A

Set: Sets VFO A value. Example: FA7030000; sets VFO A to 7.030MHz

Get: Returns the VFO A contents as an 11-digit number. Example: “FA;” returns “FA00007030000;”

FB: Get/Set VFO B

Set: Sets VFO B value. Example: FB7016000; sets VFO B to 7.016MHz

Get: Returns the VFO B contents as an 11-digit number. Example: “FB;” returns “FA00007016000;”

FR: Get/Set Receive VFO Mode

Set: Set VFO Mode: 0, 1, 2 correspond to VFO A, VFO B or Split respectively. This is the case for both the FR and FT commands (which are nominally Receive and Transmit VFOs) because in the QMX the VFO mode use does not correspond exactly to TS-480.

Get: Get Receive VFO Mode: 0 means VFO A is used for receive (could be due to VFO mode being VFO A, or VFO Mode being Split); 1 means VFO B is being used for receive (must be VFO Mode B).

FT: Get/Set Transmit VFO Mode

Set: Set VFO Mode: 0, 1, 2 correspond to VFO A, VFO B or Split respectively. This is the case for both the FR and FT commands (which are nominally Receive and Transmit VFOs) because in the QMX the VFO mode use does not correspond exactly to TS-480.

Get: Get Transmit VFO Mode: 0 means VFO A is used for transmit (must be VFO Mode A); 1 means VFO B is being used for transmit (could be due to VFO mode being VFO B, or VFO Mode being Split)

ID: Get radio ID

Get: Always returns 020 (Kenwood TS-480)

IF: Get transceiver information (TS-480 format).

Get: Returns a composite information string containing the state of the transceiver, as follows (excluding command ID and ; terminator character):

- 11-digit operating frequency (VFO A or B, according to the VFO mode setting and transmit/ receive state
- 5 spaces
- 5-digit RIT frequency, as +/-9999Hz e.g. RIT up 200Hz returns “+0200” in this field
- RIT status: 0 = RIT OFF, 1 = RIT ON
- XIT status: always 0 because QMX has no XIT
- Memory channel bank number: always 0
- Memory channel number: always 00
- Transceiver status: 0 = RX, 1 = TX
- Operating mode: returns mode character (see MD command for details)
- Receive VFO: 0 = VFO A, 1 = VFO B
- Scan status: always 0
- Split: 0 = Simplex operation (VFO mode A or VFO mode B), 1 = Split
- Tone: always 0
- Tone number: always 0
- Space character

MD: Get/Set operating mode

Set: Set to 3 (CW), 6 (FSK), 7 (CWR) or 9 (FSR/FSK Reverse)

Get: Returns 3 (CW), 6 (FSK), 7 (CWR) or 9 (FSR/FSK Reverse)

These commands actually have NO effect on the VFO at all, they only set and retrieve a Mode character, which is also sent in the IF; command response.

OM: Get the radio’s model number

Get: Returns the radio’s model number. For QMX this is QC so the result is simply OMQC;

RC: Clear RIT mode

Set: RC; clears RIT mode, setting RIT to zero

RD: Set negative RIT offset amount (absolute setting of RIT, not a change to current value)

Set: Sets negative (down) RIT; for example “RD200;” sets the RIT to -200 Hz.

RU: Set positive RIT offset amount (absolute setting of RIT, not a change to current value)

Set: Sets positive (up) RIT; for example “RU150;” sets the RIT to +150 Hz.

SP: Get/Set Split mode

Set: Sets Split mode: 0 = OFF, 1 = ON. For example “SP1;” switches QMX to split mode

Get: Returns the Split state: 0 = OFF, 1 = ON.

TQ: Get transmit state

Get: Returns the transmit state: 0 = RX, 1 = TX.

VN: Returns firmware version

Get: Returns the firmware version. For example, VN; command returns DGV_00_021QMX; (the same as the firmware file name, without the dot)

7. Firmware update procedure

One of the absolute best features of DigiVFO is that firmware updates can be done easily, by anyone, with no special hardware, software, tools or experience. This feature is called “QFU” (**Q**RP Labs **F**irmware **U**pdate) and provides the following features:

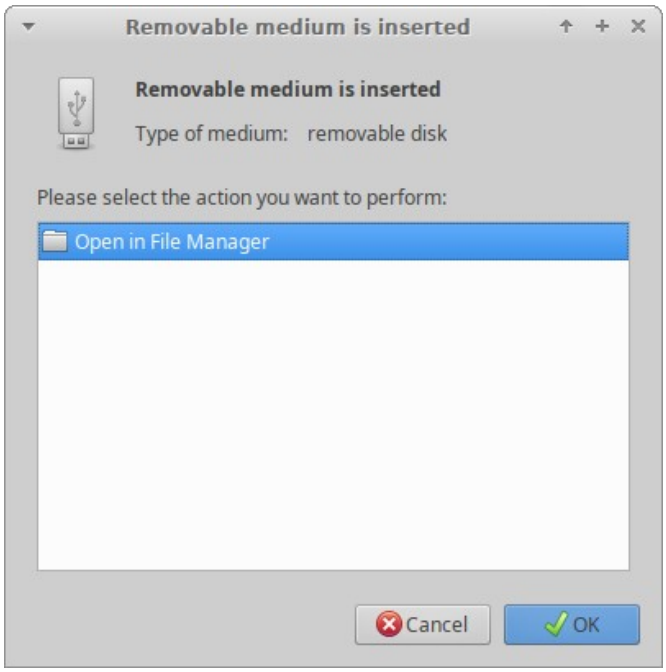
- **Easy** – anyone can do the firmware update
- **No additional hardware required:** only a standard USB-C cable (or micro-USB cable if you have a micro-USB connector installed)
- **No additional software required:** just the standard file manager application that is already available on any PC
- **No drivers:** no special drivers need to be installed, the existing drivers on any modern operating system are used
- **Works on any PC Operating System:** and in the same way: Windows, Linux, Mac
- **Secure:** firmware files are published on the QRP Labs website and are encrypted using 256-bit AES encryption technology

USB Flash memory stick emulation:

In the firmware update mode, the DigiVFO module pretends to be a USB Flash memory stick, having a 4MByte capacity and implementing a FAT16 file system. This virtual “Flash stick” contains a single file, the firmware program file in the DigiVFO microcontroller. You may read the file from the DigiVFO, or write a new one, just by dragging files in your file manager application.

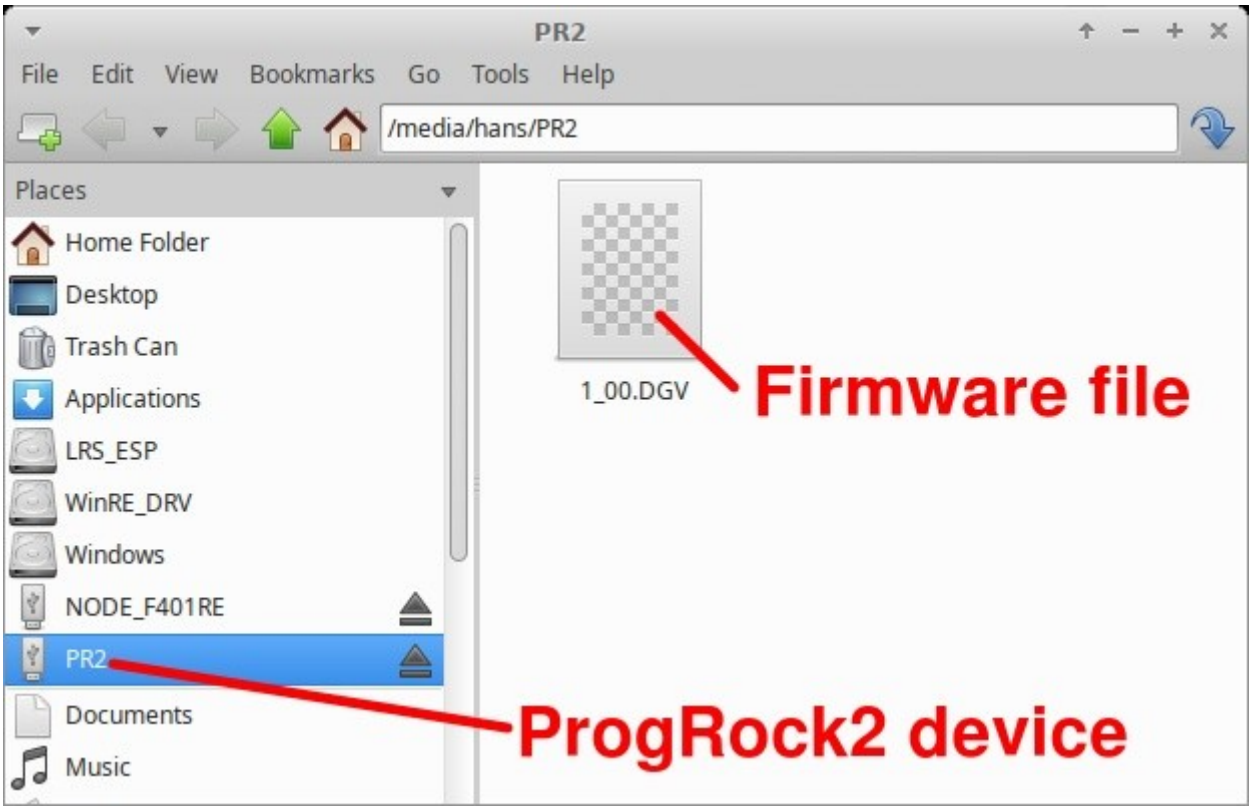
Entering bootloader (firmware update) mode:

To enter bootloader mode, set the “FIRMWARE UPDATE” configuration parameter, in the Configuration menu, to YES. Then use a long-press to exit the configuration menu, which saves the configuration parameters. Now DigiVFO will boot automatically in firmware update mode. A pop-up window should appear on your PC. On my system (Linux XUbuntu 18.04) it looks like this (right).



Click the OK button.

The File Manager window will then open, and on my system looks like this:



The DigiVFO appears as a removable USB Flash device named “PR2”, and the folder shows a single file which is the firmware version file, 1_00.DGV in this example. Note that the reason the drive shows up as PR2 is because the same bootloader is used for ProgRock2 and DigiVFO applications, because both of these applications use the same PCB.

The file name must not be longer than 8 characters, and cannot contain punctuation or spaces; the file extension must be no more than 3 characters (hence “DGV”). This is because the file system emulation is FAT16 and these are the specifications of the FAT16 format.

You may check the properties of the file and will note that it is a 23.5K file. DigiVFO firmware images are always a 23.5K file. The creation date and modification date etc. have not been set, because it was important to minimize the size and complexity of the DigiVFO QFU bootloader, in order to maximize the space available to the application firmware.

You may copy the existing firmware file to another directory of your computer. Crucially, to do the firmware update, all you need to do is copy the new firmware file to this “Flash disk”. Download the new firmware file from the QRP Labs website, unzip it, and simply drag it into the folder where the existing firmware file version is shown. Or copy and paste it, however you wish.

As soon as you copy the new file to the DigiVFO “flash drive”, the DigiVFO QFU bootloader erases the current program from its memory and installs the new one.

The DigiVFO firmware is 256-bit AES encrypted and this means:

- The encrypted DigiVFO firmware file will only work on a QRP Labs DigiVFO board, it cannot be installed on any other board, even one containing the same processor.
- No other firmware file will work on the QRP Labs DigiVFO board except an official QRP Labs encrypted DigiVFO firmware file.

The procedure will vary slightly for different Operating systems but in all cases is just a simple matter of copying the new firmware file to the emulated DigiVFO USB Flash drive.

The above firmware update procedure works on ANY modern OS because the QFU bootloader emulates a USB Flash memory stick with the USB Mass Storage Device (MSD) class, for which drivers are already present.

The QFU bootloader implements a USB device stack (Mass Storage Device class), emulated FAT16 file system, Flash erase/write, and 256-AES encryption. It occupies the first 10K of Flash memory leaving 21K for the application itself and 1K for the non-volatile storage of the configuration parameters.

Important notes about the DigiVFO firmware implementation

1. The only way to enter firmware update mode, is to set the “FIRMWARE UPDATE” parameter to YES.
2. The only way to get out of firmware update mode, is to update the firmware by copying in a new firmware file. Even power cycling doesn’t get you out of firmware update mode. If you enter firmware update mode by mistake, you can just to a “Copy and paste” of the current firmware in the directory. This will overwrite the firmware with itself, which is pointless except that it does get you out of the firmware update mode.
3. When you execute a firmware update, the stored configuration parameters are set back to their defaults; so effectively a firmware update is also a factory reset.

7. Resources

- For updates and tips relating to this kit please visit the QRP Labs QDX kit page http://qrp-labs.com/DigiVFO
- For any questions regarding the assembly and operation of this kit please join the QRP Labs group, see http://groups.io/g/qrplabs for details

8. Document Revision History

1.00	19-Dec-2025	First draft version version 1.00
1.00a	27-Feb-2026	Updates because Rev 2 PCB has USB-C connector; and to clarify that the USB Virtual COM Serial port is only for CAT (not a terminal interface).