

Sandwich Digital VFO Kit Assembly Manual

Rev. C, April 13, 2017

CRKITS.COM

The Sandwich digital VFO is a great addition to your KN-Q7A, [CS-series](#) or other [SSB radio](#). By adding Sandwich, you can have **full band coverage with LED frequency display**. The **frequency stability is the same as a crystal** with very little thermal drift even you transmit all the time. Unlike a DDS, the Sandwich digital VFO **will not bring lots of birdies**. It supports BFO calibration so **both USB and LSB modes will be supported**. It also supports crystal frequency calibration so you **will be exactly at the kHz or 100 Hz**.

Part List

For Ctrl and OSC boards:

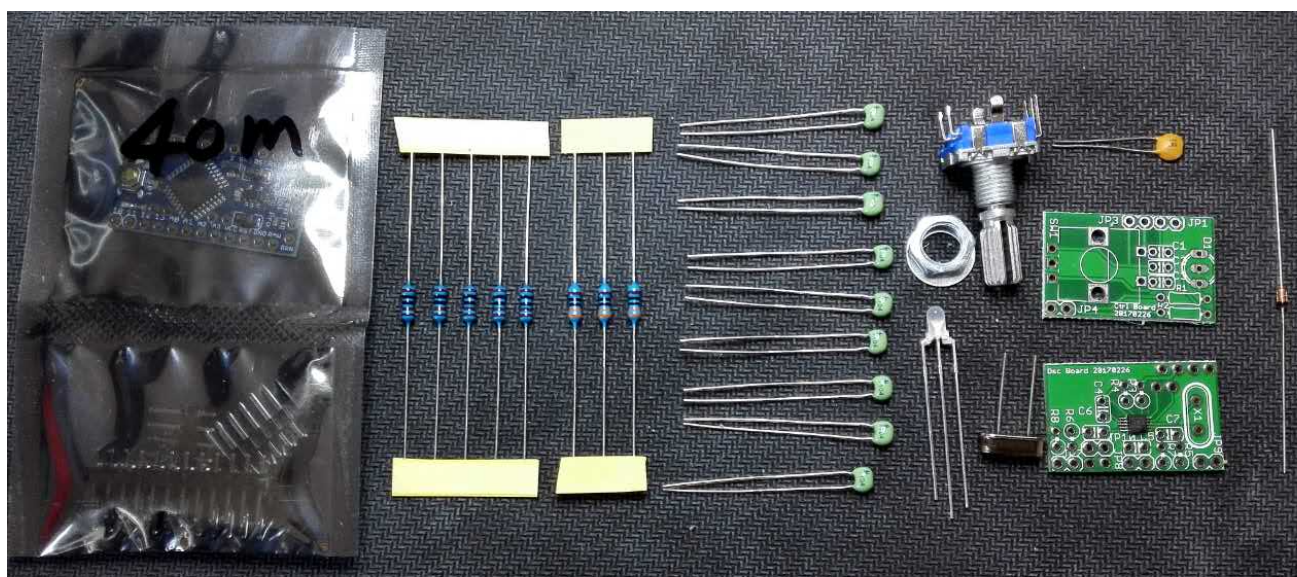
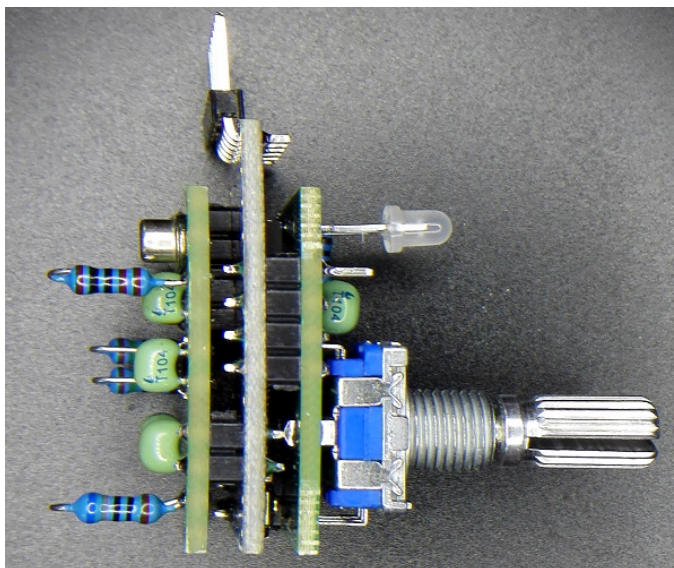
- R1, R3, R4, R5, R6 = 1k
- R2, R7, R8 = 330 ohm
- C1~C7 = 104
- X1 = 27 MHz crystal
- D1 = 3 mm dual color LED, see [the next page of this manual](#) to identify the polarity
- IC1 = si5351a, pre-mounted
- SW1 = rotary encoder with push-on button, 20 pulses per rev.
- JP10 = Jumper to select band high limit, BFO and default operation frequencies, [also known as ITU region jumper](#), see [the table in the next page of this manual to understand better](#).

Extra parts for radio assembly:

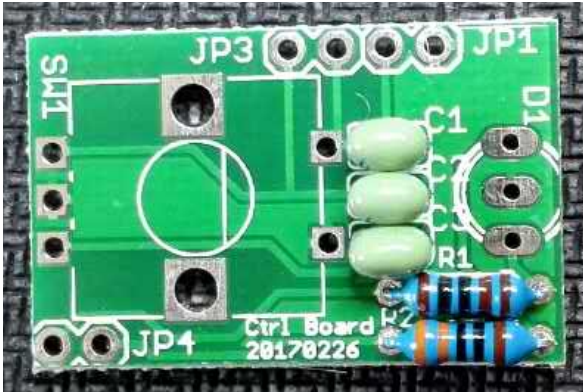
- 104 x 2 for VFO and BFO wiring
- 15 pF x 1 for optional VFO filtering (might remove from kit later)
- 1N4148 x 1 for optional power line current leaking prevention (might remove from kit later)

Arduino Pro Mini ATmega328 3.3V/8MHz board with firmware pre-programmed (optional)

FTDI Serial Programmer based on FT232RL chip, default VCC=3.3V (optional)



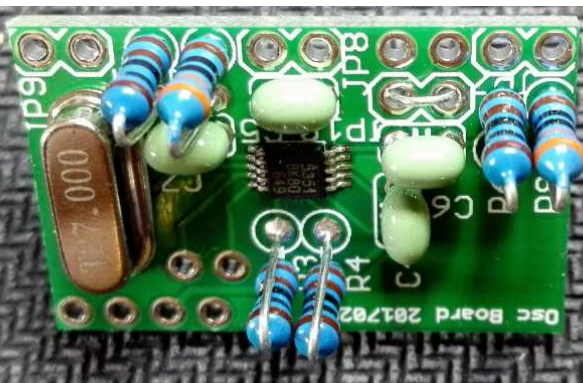
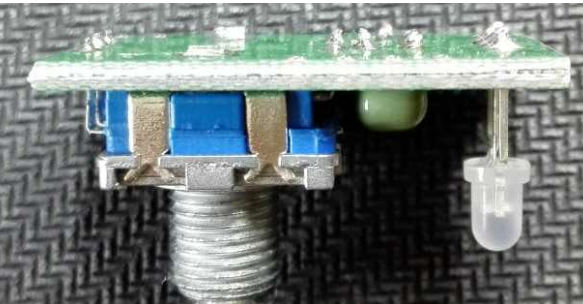
Kit Building



Start with Ctrl board.
Install R1 = 1k, and R2 = 330R.
Install C1 = C2 = C3 = 104.
Cut the extra pins flush.



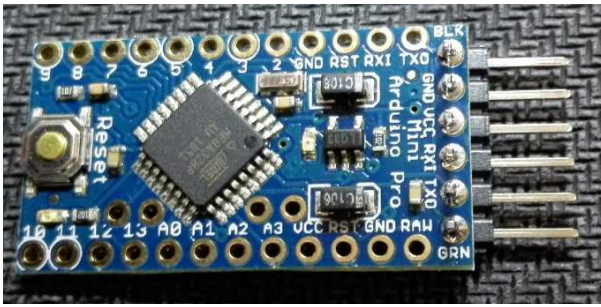
Install SW1 = rotary encoder with push-on button.
Install D1 = 3 mm dual color LED. The shortest pin is installed near D1 marking. Leave LED pins about 6 mm long so it protrudes from the face plate.
Cut the extra pins flush.



Proceed to Osc board.
The IC is pre-mounted. Double check for solder bridges or cold soldering now.
Install C4 = C5 = C6 = C7 = 104.
Install R3 = R4 = R5 = R6 = 1k.
Install R7 = R8 = 330R.
Install X1 = 27 MHz crystal.
Install JP10 using a discarded component pin if necessary. See table below for KN-Q7A settings.
Cut the extra pins flush.
It is recommended that you install resistors as shown for more convenient test probe contact.

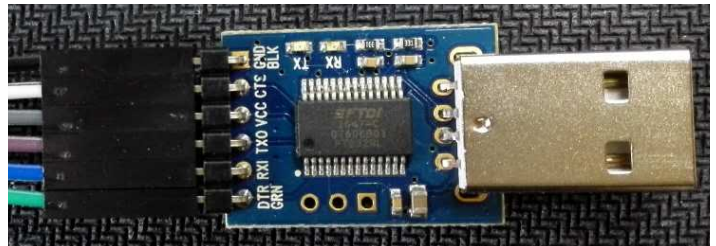
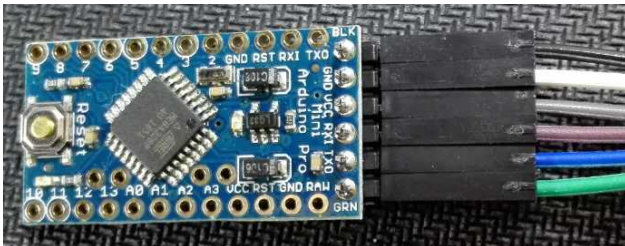
JP10	80-meter	40-meter	20-meter
OPEN	band high limit = 3800kHz default frequency = 3700kHz BFO = around 8467.2kHz	band high limit = 7200kHz default frequency = 7100kHz BFO = around 8467.2kHz	band high limit = 14350kHz default frequency = 14300kHz BFO = around 4096kHz
SHORT	band high limit = 3900kHz default frequency = 3700kHz BFO = around 8467.2kHz	band high limit = 7300 kHz default frequency = 7200 kHz BFO = around 8192kHz	band high limit = 14350kHz default frequency = 14200kHz BFO = around 4194.3kHz

Note: The above setting is only applicable for KN-Q7A. For CS-series, as it uses common IF of 8467.2 kHz, the setting is different. See CS-series detailed manual or CS-series Arduino sketches for more information.



The complete kit comes with a pre-programmed Arduino Pro Mini ATmega328 3.3V/8MHz. If you wish to update firmware by yourself, it is good time to install the L-shaped pin headers on the bottom side. If you don't want to update firmware, you can skip this page.

Connect wires as shown to the FTDI 3.3V Serial Programmer (optional).



You need to use Arduino IDE (Integrated Development Environment) to update firmware. If you haven't installed yet, please choose correct version for your operating system and download and install:

<https://www.arduino.cc/en/Main/Software>

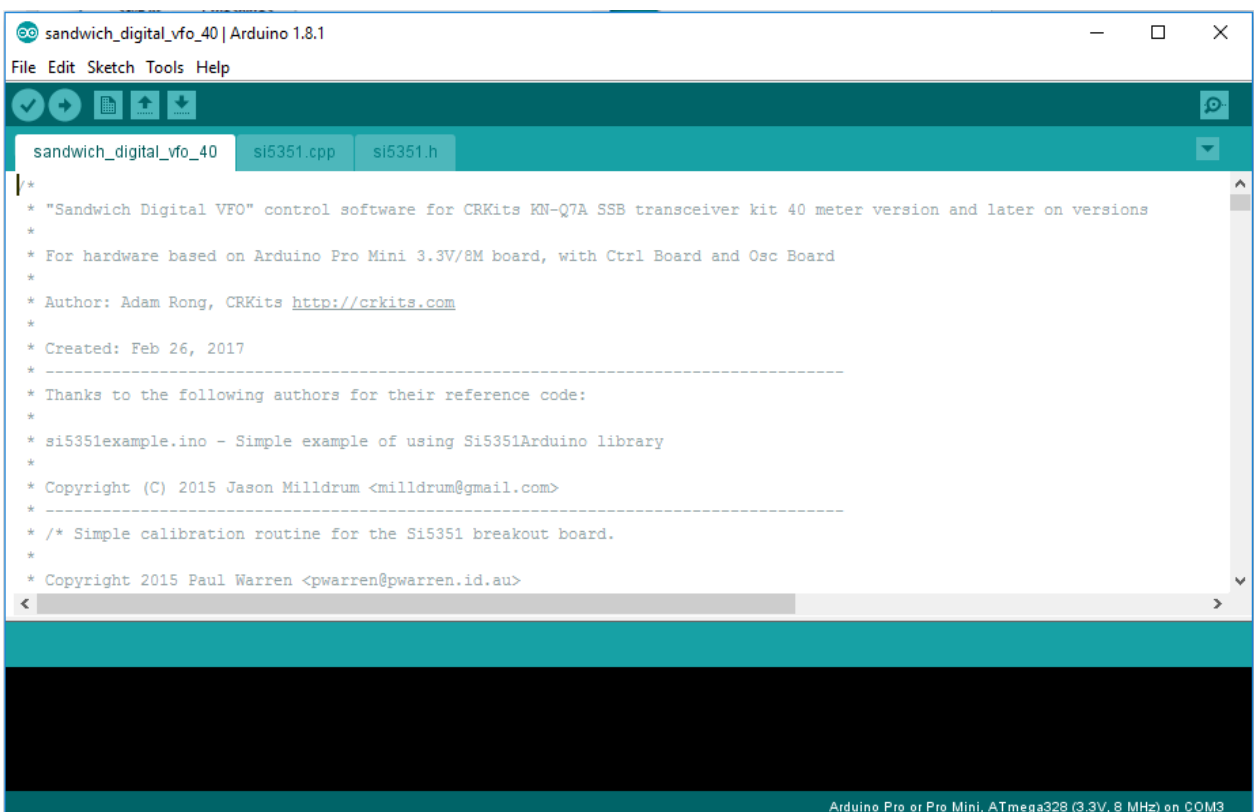
Plug the FTDI programmer to one of your USB ports on your computer. If it needs to install driver, download and install here:

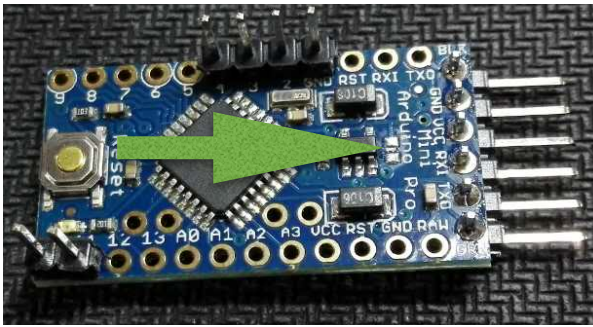
<http://www.ftdichip.com/Drivers/VCP.htm>

Download Arduino sketch (source code, choose your band from 40, 20 or others) from:

<http://crkits.com>

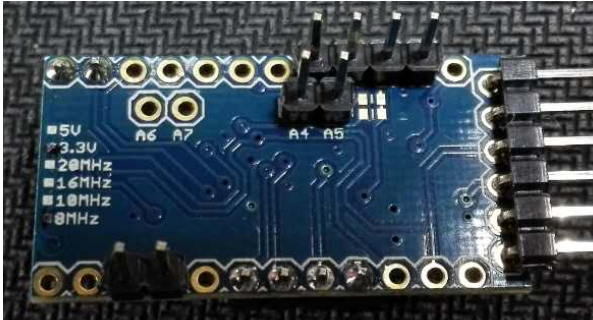
Unzip the file, and double click the file with extension name .ino and it will open up Arduino IDE and load the files automatically. In Tools menu, set Board as "Arduino Pro or Pro Mini" and Processor as "ATmega328 (3.3V, 8MHz)", and choose Port as the serial port newly created for the FTDI programmer. If your Arduino Pro Mini board is connected through the FTDI programmer, clicking the 2nd icon on top left of the window will compile and upload the firmware to your board.





Install 4-pin header from GND to 4 on the top side.
Install 2-pin header from 10 to 11 on the top side.
Make sure they are straight up and not floating then solder.

[Optional] Remove this arrow pointed chip resistor to save current consumption of about 1.5 mA as power on indicator is not needed on this board.



Install 4-pin header from GND to A3 on the bottom side.

Install 2-pin header from A4 to A5 on the bottom side.

Install 2-pin header from 6 to 7 on the bottom side.
Make sure they are straight up and not floating then solder.



Stack up the Ctrl board from the top side of the Arduino board.

Observe no inter-board short circuit.

Make sure no floating then solder.

Cut the extra pins flush.



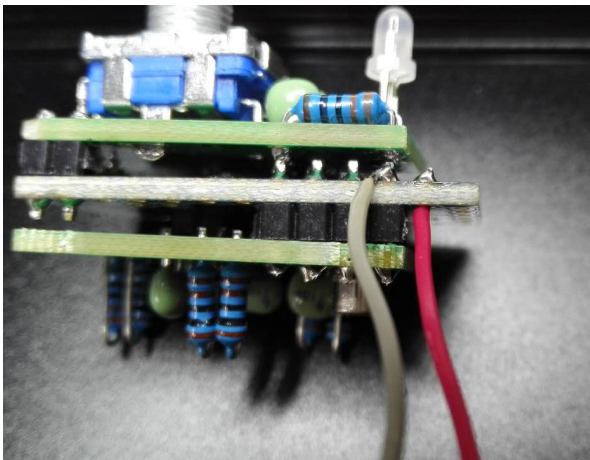
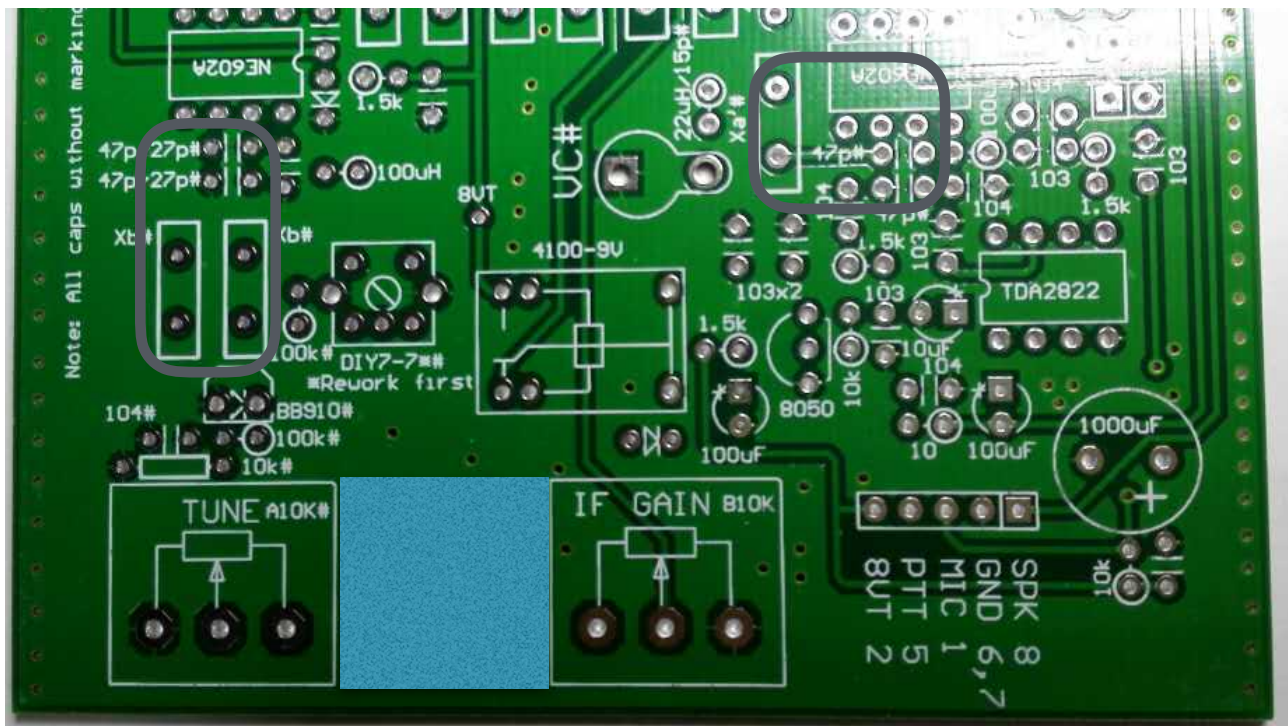
Stack up the Osc board from the bottom side of the Arduino board.

Observe no inter-board short circuit.

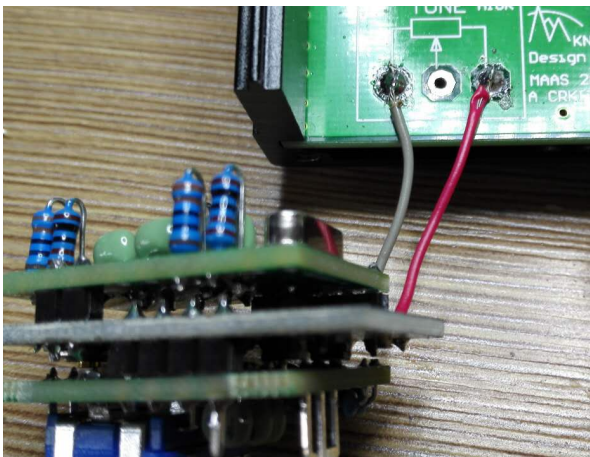
Make sure no floating then solder.

Cut the extra pins flush.

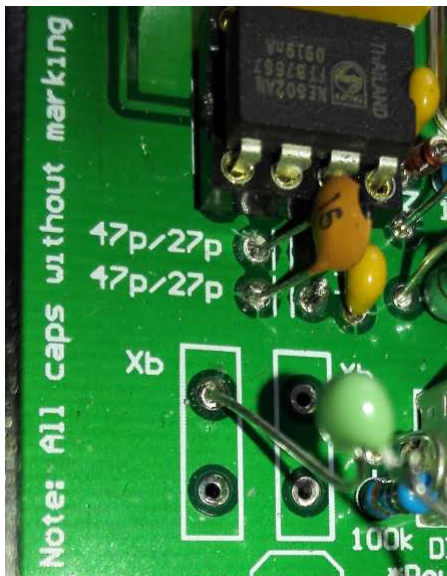
Rework KN-Q7A main board to remove a few components marked with "#". In VXO portion, remove 2x 47p/27p, 2x Xb, 2x 100k, DIY7-7*, BB910, 104 and 10k. In BFO portion, remove 2x 47p, Xa', 22uH/15p, VC. At least you need to remove the underlined parts (outlined in photo).



Solder two wires (about 5 cm long) to Arduino board. RAW for power positive and GND for ground.

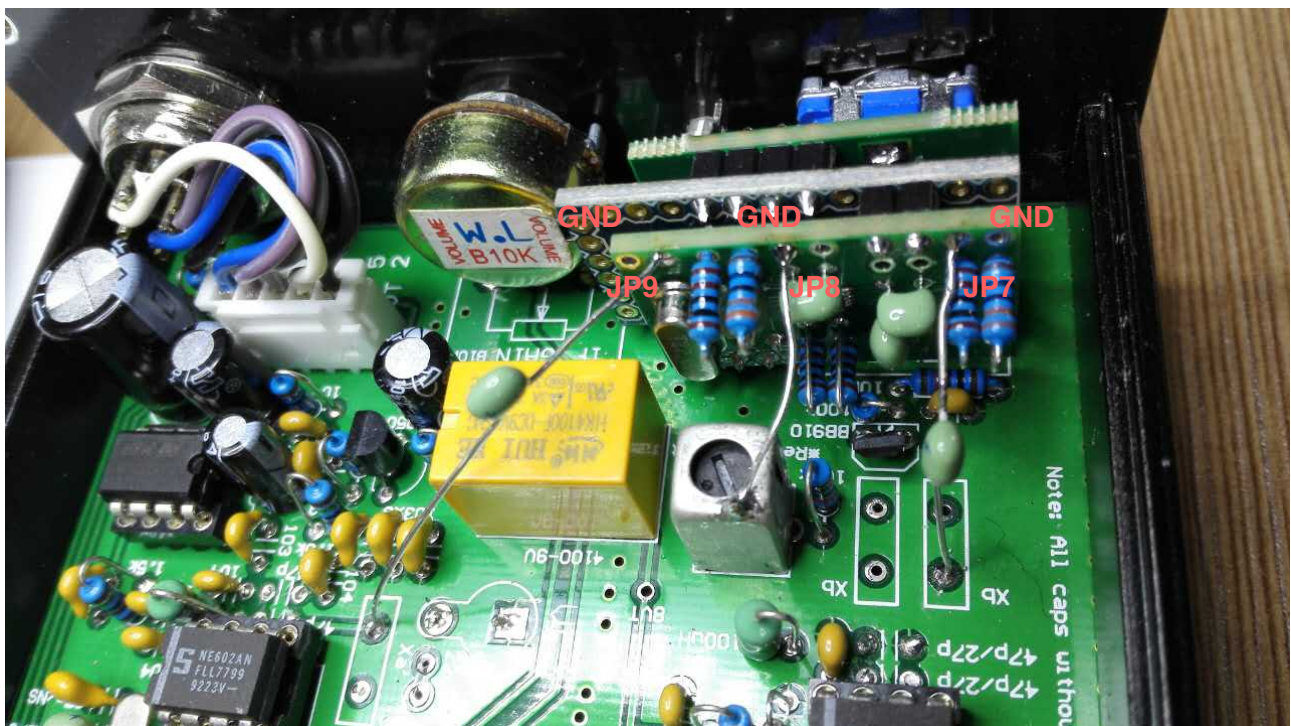


Solder another side to A10K pads, ground to left side pad and power positive to right side pad to get regulated 8V power supply.



Use a 104 capacitor to connect JP9 (the pin on the right, GND on the left) to BFO (one pin of Xa' as shown), and another 104 to connect JP7 (the pin on the left, GND on the right) to VFO (one pin of Xb as shown). You can use a short wire to connect GND pin (JP8 left pin) to the ground of the main board for shorter and better grounding. As for 15pF capacitor, you can install near NE602 as shown to help improve spurs but it is optional here.

You might find a piece of 1N4148 diode in early version of kit. It can be used in the power line to block the reversed current leakage while in circuit programming, but in most cases it is not needed as the 3.3V LDO on the Arduino board has the same function.



You will need to drill a 3mm diameter hole for LED in the middle of two holes for potentiometers.

You can reuse the TUNE knob but a smaller one same as the IF GAIN knob will do the job alright as well.

Function Description

The current firmware dated February 26, 2017 supports two modes: Normal mode and Calibration mode. The latter is activated by pressing and holding the encoder button at power on. In calibration mode, pressing the encoder button can toggle between BFO calibration and crystal calibration. At power on, you can easily tell the mode from the color of the LED: red for Normal mode and orange for BFO calibration mode.

In normal mode, pressing the encoder button will toggle between step 1 kHz and 100 Hz, and it is good enough for SSB application. In 1 kHz step mode, LED green means 1, 3, 5, 7, 9 kHz, LED out means 2, 4, 6, 8 kHz, and LED orange means 10, 20, 30, 40, 50, 60, 70, 80, 90 kHz, and LED red means 100 kHz, 200 kHz, and LED red also means out of band. In 100 Hz step mode, every 2, 4, 6, 8 kHz, you can also see green. See the following table to understand it better.

70	70	70	70	70	70	70	70	70	70	71	71	71	71	71	71	71	71	71	71
90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09
Orange	Green	White	Green	White	Green	White	Green	White	Green	Red	Green	White	Green	White	Green	White	Green	White	Green
White	White	Green	White	Green	White	Green	White	Green	White	White	White	Green	White	Green	White	Green	White	Green	White

In calibration mode, pressing the encoder button will toggle between BFO calibration and crystal calibration, and meanwhile memorize the last setting to EEPROM. In BFO calibration mode, LED is orange unless it is tuned out of range (red), and the step is 100 Hz. In crystal calibration mode, LED is green, and the step is 20 Hz.

Calibration

Press and hold the button on the rotary encoder, then power on the radio. You will see LED is orange meaning it is in BFO calibration mode. Press button again to enter crystal calibration mode, carefully hook a frequency counter to JP8 (right pin), turn the encoder counter-clockwise for about 70 steps and monitor the frequency counter to as close as 10.000000 MHz, then press the button to memorize the crystal calibration setting.

Remove power supply from radio. Re-enter BFO calibration mode by pressing and holding the button and power on again. Same as KN-Q7A, you will need to download an audio spectrum analyzer software from:

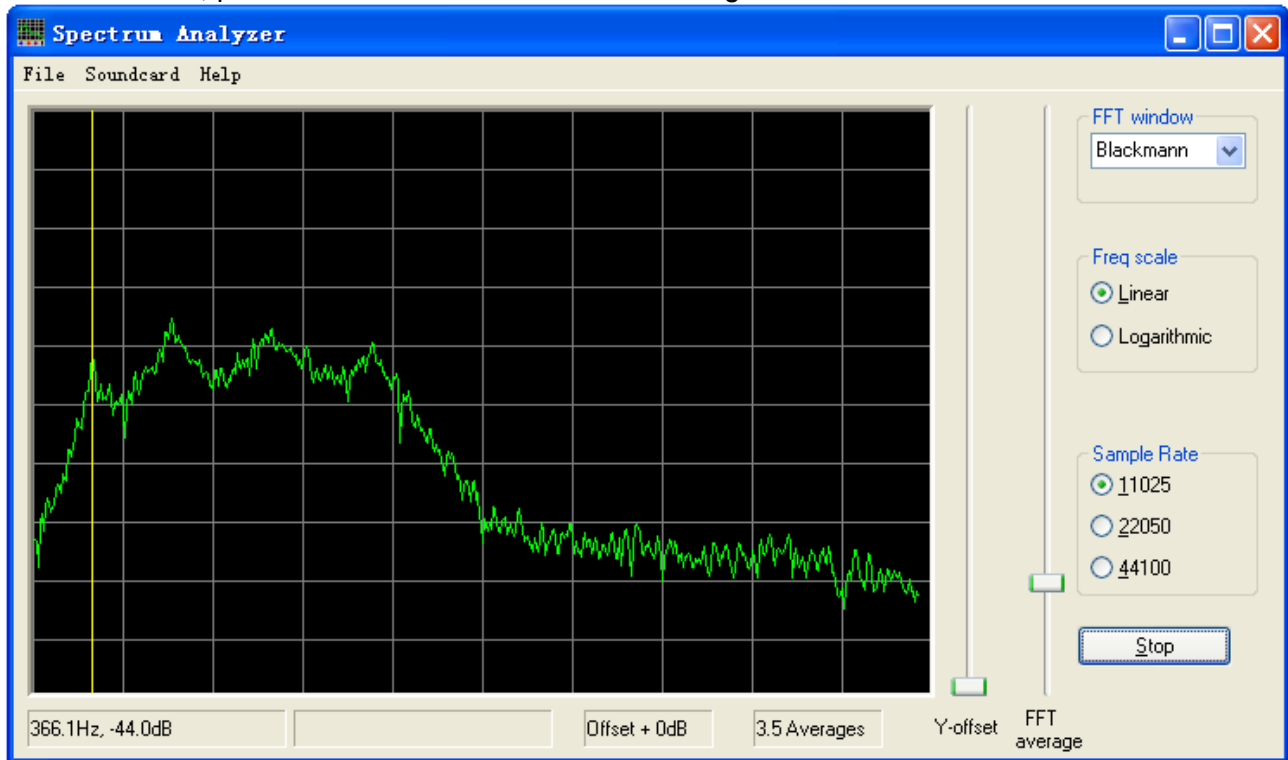
<http://www.qsl.net/zl1an/Software/Spectrum3.zip>

Unzip and double click the spectrum2.exe file to run. Connect the speaker out to the microphone input of the sound card on PC. You will also need to connect a noise generator or simply an antenna to the antenna connector, and apply power supply to KN-Q7A radio.

Follow the setup on the screen capture below, and click start, you will see a similar audio spectrum. If not, you may need to turn clockwise or counter clockwise the IF GAIN control. In BFO calibration mode, tune encoder to move the pass-band to the left or to the right. You can use the cursor to read the frequency and level of the signal (on the left bottom corner) to identify the low and high end of the pass-band, so you can get about 350 Hz~2200 Hz range. If you have wider or narrower range, please make sure you still set the lower end of 350 Hz. Please note that, if you are using an antenna to do this, you should tune away from any signal to use the band noise as a random noise source, or the spectrum will jump up and down making the measurement very difficult.

Please also note, both LSB and USB will be possible since the BFO calibration range is wider now. If you can see correct audio spectrum range, but cannot receive clear voice signal of the intended side band, you might need to tune encoder again to use the correct side band. Normally, the

default value should be very close to the intended mode. Just a few steps tuning should be okay. Once it is done, press the button to memorize the setting to EEPROM.



Modification Possibilities

Sandwich is a small kit providing the basic feature as a digital VFO. If you know Arduino and electronics quite well, you might want to add a few features by yourself.

On the Osc board, you can find a few unused pins of Arduino board, including analog pin A3 and digital pin D6 (PWM output possible). You probably need to detect TX or RX status, so you can show different LED colors, or provide TX/RX offset for CW mode. You probably also need to detect SWR, RF power and other analog parameters. Another great idea might be adding an audio Morse Code reporter of the current frequency instead of counting all the time.

The fun of a kit is not only building but modifying and sharing. Please share your great ideas to the team at:

http://groups.yahoo.com/group/CHINA_QRP/

Adapt to other SSB radios

Take BITX-40 as an example, it has a BFO frequency of around 12 MHz, while VFO frequency range is around 4.7 ~ 5.0 MHz for ITU region 2 or around 4.8 ~ 5.0 MHz for ITU region 1&3. Unlike KN-Q7A, the operation frequency is a result of BFO frequency - VFO frequency, while KN-Q7A's operation frequency is a result of VFO frequency - BFO frequency. In the Arduino sketch, here are a few changes:

```
if (digitalRead(JP)) //For BITX-40, JP is used to select ITU region
{
    op_freq = 7100000; //set op_freq default
    band_high_limit = 7200000; //set band high limit to 7.2 MHz
    bfo_freq = 11998000; //for IF = 12000 kHz, default to LSB mode
    bfo_low_limit = 11992000;
    bfo_high_limit = 12002000;
```



```

}
else
{
    op_freq = 7100000; //set op_freq to default
    band_high_limit = 7300000; //set band high limit to 7.3 MHz
    bfo_freq = 11998000; //for IF = 12000 kHz, default to LSB mode
    bfo_low_limit = 11992000;
    bfo_high_limit = 12002000;
}

```

Also, CLK0 frequency setting code should be changed in multiple places from:

```
si5351.set_freq(uint64_t((bfo_freq + op_freq)*100), SI5351_PLL_FIXED, SI5351_CLK0);
```

to:

```
si5351.set_freq(uint64_t((bfo_freq - op_freq)*100), SI5351_PLL_FIXED, SI5351_CLK0);
```

Besides firmware change, output amplitude is also very important. As BITX-40 uses diode mixer which requires higher level of both VFO and BFO signals, it is recommended to remove attenuation resistors of 1k and 330R (R5~R8) and use a wire to replace 1k. Note that two 1k resistors (R3 and R4) for I2C pull up should not be changed.