Forty Niner PLL – Hardware Manual

Another 49’er project with a digital VFO. **Hardware : Luc ON7DQ & Gil ONL12523 – Software: Gil ONL12523**

Introduction

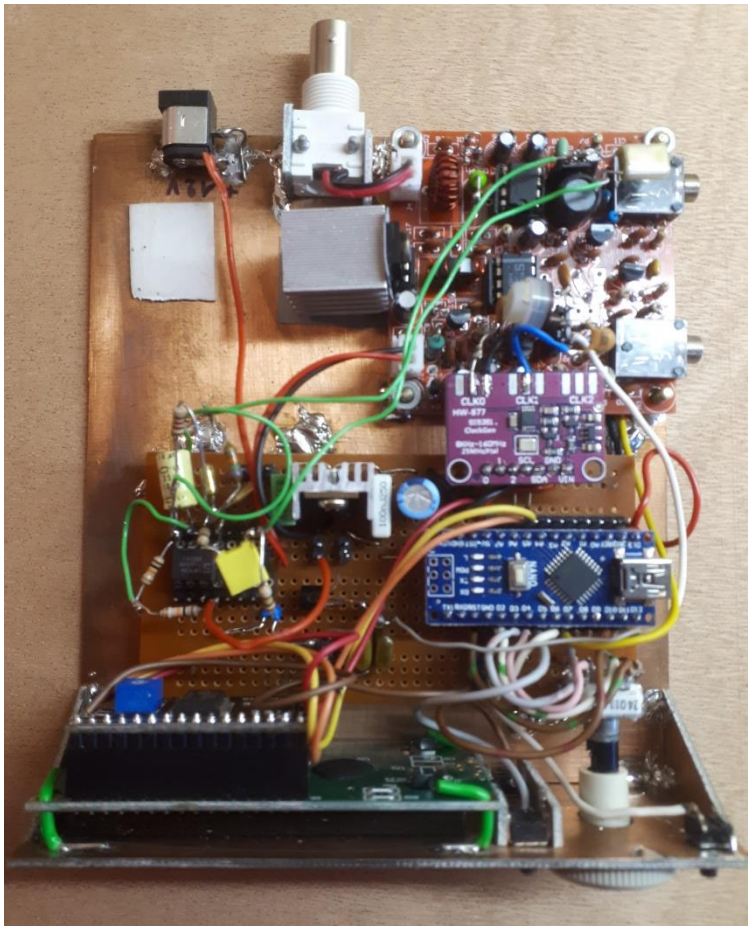
This project started as a basic 49’er project, based on the QST article by Jack Purdum W8TEE et al. "**A Modular 40 Meter CW Transceiver with VFO**" [1].
That article was published in March 2016, and I read it with great interest, thinking “YES! I must build one too” … but I had never found the time to actually do it.

In autumn 2022, I finally got to order a Chinese 49’er kit, and build it as instructed in the 25 page manual that can be downloaded from the ARRL website [2] or from [3].

Some more changes were needed though. The AD9850 DDS for the VFO has become difficult to source, and very expensive, so it was replaced by an Si5351 module.
We also wanted to do some other stuff like keying the TX from the Arduino, generate a PTT signal to switch a TX/RX relay, or generate a variable sidetone in the Arduino.

I wanted our project to be “educational”, like the one in the QST article, so it had to be in a clear plastic box. At the same time I wanted it to be environmentally friendly, so it had to be some recycled box, which I found in the form of a case from a video tape of the old Philips VCR standard from 1972.

And this what I made of it. Everything is mounted on an L-shaped PCB chassis, which then shoves inside the box, and is fixed with double-sided adhesive tape. The 49’er board is mounted on HEX standoffs, which are soldered to the ground plane, and the perfboard with the Arduino, the voltage regulator and the Si5351 module (and an experimental CW filter) is mounted by soldering some stiff wires to the ground plane.



This is a view of the simple front panel before mounting it inside the box.

The hardware

Building the 49'er board

Keep the circuit diagrams from the last two pages handy to follow this explanation.

First I built the 49'er kit, with the changes as described in the QST article and the W8TEE manual. Summary of those changes is:

- Do not install R3, C4, C5, C6, W1 (trimmer pot), Y1 and Y2 (xtal's)
- Replace R5 with a 10k resistor
- Replace C2 with 82 pF
- Replace D2 with a 5.1V Zener diode
- Put series LC filter (22µH + 56 pF) where Y1 was
- Put a 3 pin header where Y2 was, connecting the middle pin to GND.

Further changes we made because we used separate outputs from the SI5351 for the TX and RX oscillator, was disconnecting C10 (82pF) from pin 6 of the NE602, and feed it with the separate TX oscillator signal from CLK0. Also we changed C10 to 1nF.

The level from the Si5351 is a bit too much for the NE602, so we inserted two series capacitors (both the Si5351 and the NE602 have a DC level on their output or input), with a 220 Ω trimmer pot in between, to adjust the LO level.

D6 was disconnected from the keying line, and connected to a new NPN transistor which will serve as the PTT driven by the Arduino (I mounted it under the 49'er board with a 1k resistor at the base, and covered the resistor in shrink wrap).

In the same way, a keying transistor was put over the keying input on the 49'er, so that also the Arduino kan key the transmitter with a preprogrammed message.

A nice sinewave sidetone is made by the Arduino, it only needs a simple RC low pass filter and one transistor. So you can omit all components of the Twin-T oscillator in the 49'er, and feed the sidetone directly to pin 3 of the LM386. If you had already mounted the sidetone components on the 49'er, just remove resistor R1 to disable it.

I made a simple single stage opamp filter at 600 Hz, this is still wide enough for casual listening to SSB signals, and narrow enough to make CW operating possible. We're still working on a switched capacitor filter that follows the offset and sidetone frequency. This can be done by using the third output of the Si5351. More on that whenever this becomes available.

During tests I blew up the original PA transistor (Q5, D885), and replaced it with a BD139, but mounted it in a socket made from 3 pins of an IC socket. So now I can "plug in" my final transistor like in the tube days, hi. Always handy whenever it blows again …
Oh, oops, I did it again … and this time I replaced it with a SC1909 from the junkbox, which happens to be a CB transistor, and now I get about 2 Watts out, great !

This is the final circuit after making all the changes (marked in yellow), compare it to the original 49'er circuit which you can find online [4]:

Building the Arduino Controller

The Arduino circuit is nothing special, nor is it difficult, I just built it on a piece of perfboard.
An Arduino Nano, a 1602 LCD display with I2C interface (UC-146) to reduce the number of wires, and a Si5351 PLL module.
A voltage regulator, and a handful of discrete components complete the circuit.
See title picture for the final result.

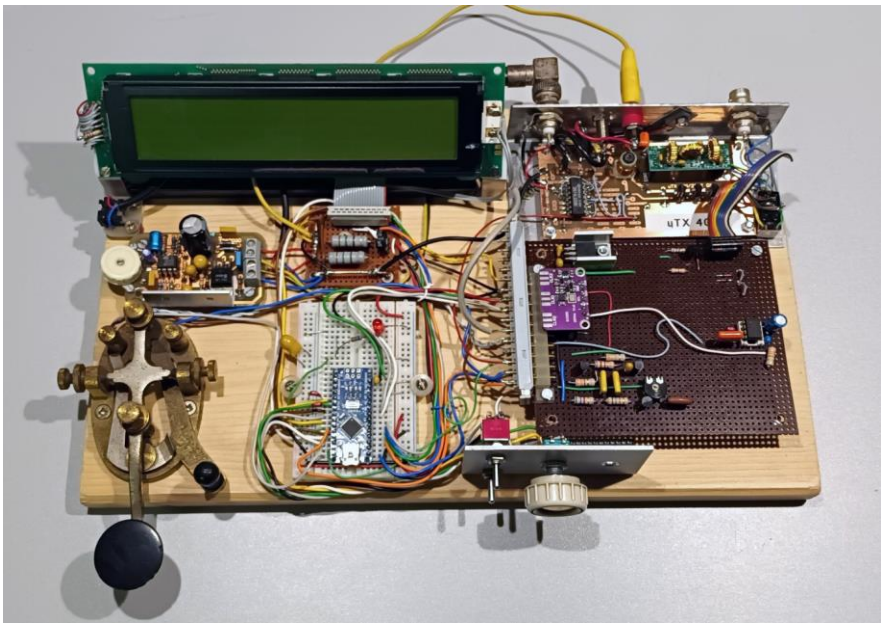
The software

So far so good, but then came the big disappointment … the Arduino code in the zip file from the ARRL website was far from complete: it didn't support the RIT function like was promised in the article, also the offset didn't work as described … the only thing it did was check the band limits for different license classes (not really needed for us in Europe), and store the last used frequency in EEPROM.
Even an updated version of the software I found in this newsgroup [5] was not what I was looking for. There is a solution by K4KRW, but I found it rather complicated [6].

So I called in the help from my good friend **Gil, ONL12523**, who has also written all excellent software for the **OST Morse Box** [7].

By using two of the three oscillators in the SI5351 PLL module, we were able to find an elegant solution to above problems.

As a sidenote, Gil did not have a 49'er kit, nor did he want to order one.
So he built a similar setup from scratch, on a wooden plank. Doesn't it look great ?



His circuit shows a few differences from mine: he used an existing audio amplifier and a class-E power stage like in the QCX transceivers from QRP Labs.
But essentially, the software is the same for both of our projects.

All needed files to program the Arduino are my github page. [8]
There you will also find this manual, a nice operating manual.

Any remarks or questions on this project can be sent to me at on7dq@hotmail.com

73 de Luc ON7DQ & Gil ONL12523

Credits

The software contains some libraries and works of others, credits are given in the code where needed.

References

[1] the original article can be downloaded from this website:
https://docplayer.net/32602367-A-modular-40-meter-cw-transceiver-with-vfo.html
or this one: https://manualzz.com/doc/42613183/a-modular-40-meter-cw-transceiver-with-vfo

[2] the extra files accompanying the article can be downloaded from the ARRL website (you must be a member): http://www.arrl.org/qst-in-depth

[3] as an alternative, the detailed 25 page assembly manual is also here:
https://github.com/LZ1DPN/Forty-9er/blob/master/PDP/Forty-9er/Purdum0316-QST-in-Depth-AssemblyManual.pdf or here:
http://www.radiomanual.info/schemi/Vari/S-Forty-9er_additional_assembly_instructions_2016.pdf

[4] manual for the Chinese 49'er kit with the original circuit diagram:
http://lz1lcd.spirka.net/downloads/Forty-9erV1.0.pdf

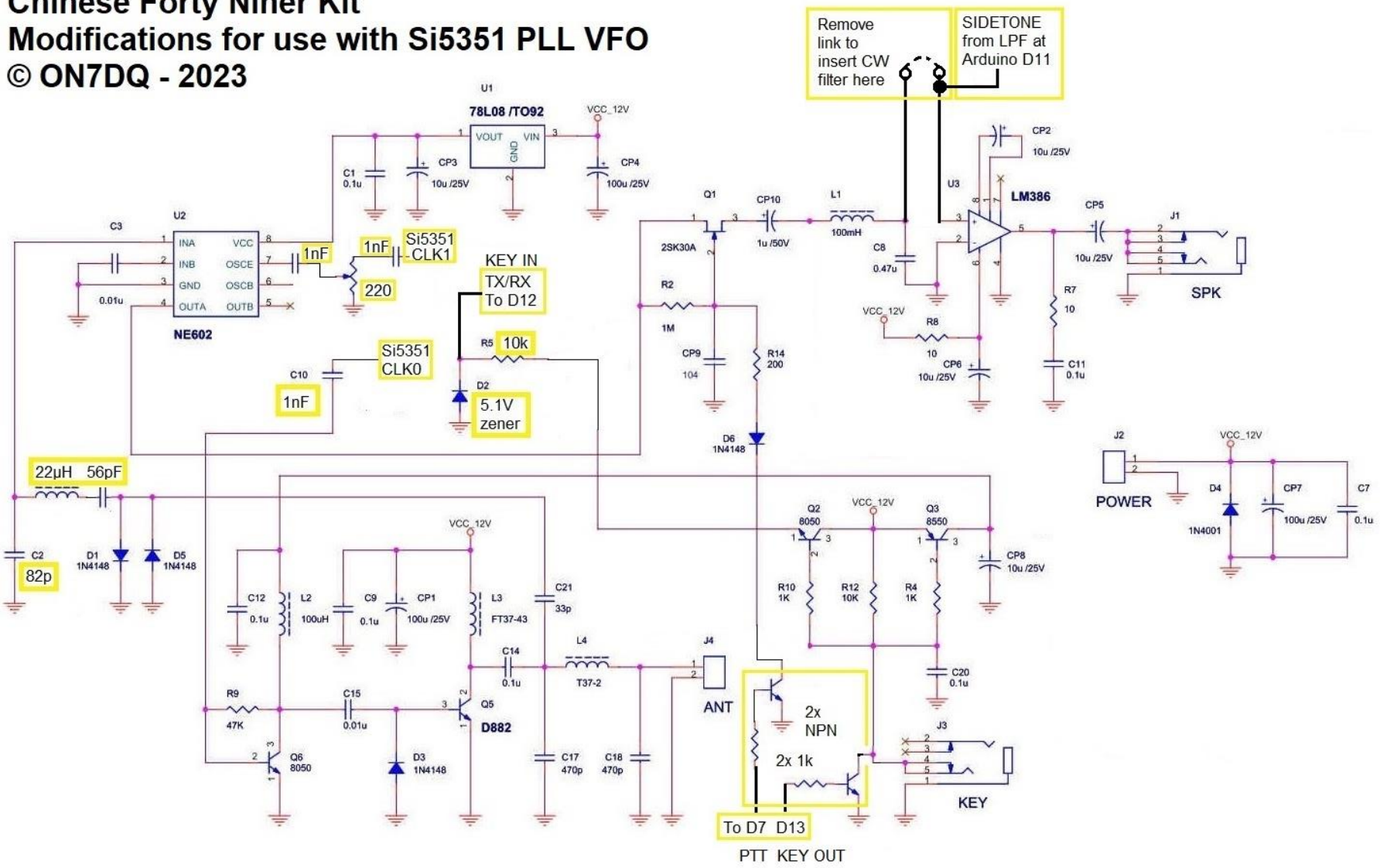
[5] more information and other software solutions can be found in this newsgroup:
https://groups.io/g/SoftwareControlledHamRadio

[6] another software solution by K4KRW:
https://github.com/rydodd/SCRadioSoftwareK4KRW

[7] OST Morse Box: basic version https://github.com/on7dq/OST-Morse-Box and extended version https://github.com/on7dq/OST-Morse-Box-DG , plus a new Windows Control program https://github.com/on7dq/OST-Morse-Box-V3

[8] Forty Niner PLL github page, with Arduino code and two manuals:
https://github.com/on7dq/Forty-Niner-PLL

Chinese Forty Niner Kit
Modifications for use with Si5351 PLL VFO
© ON7DQ - 2023



This is an overview of the Arduino control circuit.