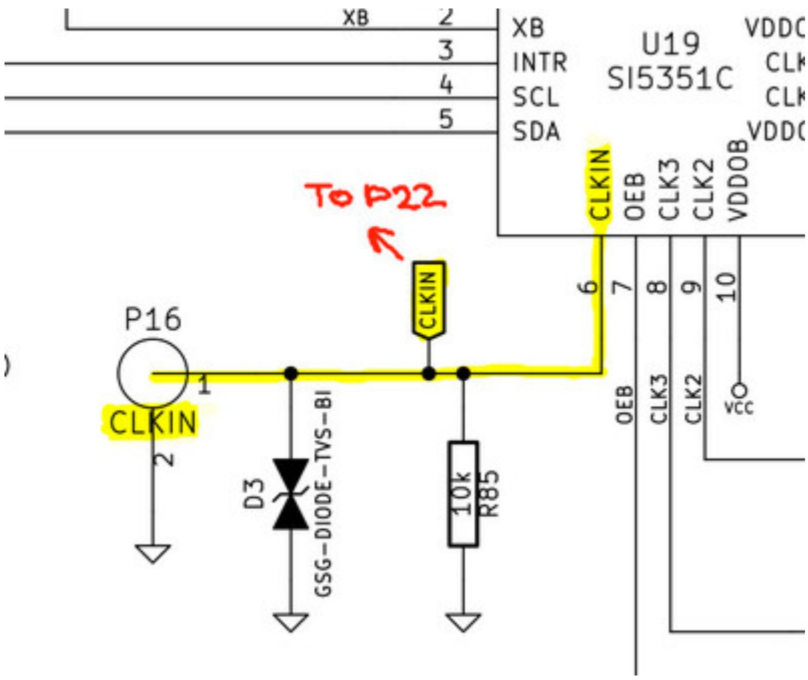
tablix.org

Avian’s Blog: HackRF clock converter

Tomaž

6–8 minutes

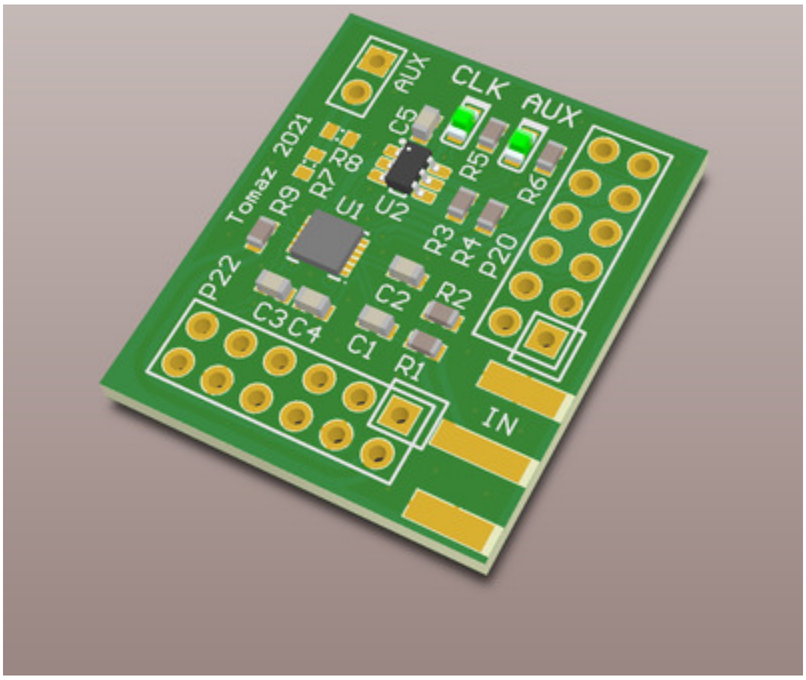
HackRF can use an external 10 MHz reference clock instead of the built-in crystal oscillator. The CLKIN input accepts a DC coupled, CMOS-level, 3.3V square wave signal since it's connected directly to the digital input pin on the SI5351C PLL chip. I want to run my HackRF from the 10 MHz reference signal generated by my ERASynth Micro. Unfortunately, the TCXO output from the ERASynth Micro is an AC coupled, sinewave-ish signal and hence not directly compatible with the HackRF's CLKIN. While I've seen reports that sine wave signals on CLKIN also tend to work, I wanted to make a proper interface that didn't drive the SI5351 input outside of its rated signal levels.



In the future I might also want to synchronize the HackRF to other clock sources and I think a DC coupled, CMOS-level output is quite rare on instruments. Hence modifying the HackRF to accept a wider range of signals on the CLKIN connector seems useful to me.

I very much copied the idea for the circuit design from the Osmocom project's osmo-clock-conv. osmo-clock-conv is a stand-alone board that uses an Analog Devices LTC6957 clock buffer to convert a wide range of clock signals into a CMOS-level square wave. The LTC6957 is a specialized chip for this purpose that introduces very little additional phase noise and jitter into the signal during conversion. It should perform much better than, for example, a diode and a Schmitt trigger "self-biasing clock squarer" circuit with a similar function in the osmo-clock-gen.

I could have just used osmo-clock-conv board directly, or in fact I could just order the LTC6957 evaluation board and connect it via a coax to CLKIN. However I felt like making a more elegant solution that would be more tightly integrated with the HackRF. HackRF offers quite a lot of possibilities through various extension headers on its circuit board. The header P22 is connected to CLKIN and can be used to add add a custom circuit that supplies the reference clock signal. Adding a small TCXO board to P22 is quite popular and there are HackRF enclosures readily available that leave enough space for the TCXO mod. Hence adding a small clock converter circuit in place of the TCXO should be relatively straightforward and I could get a nice enclosure off-the-shelf that would nicely fit my modified HackRF.

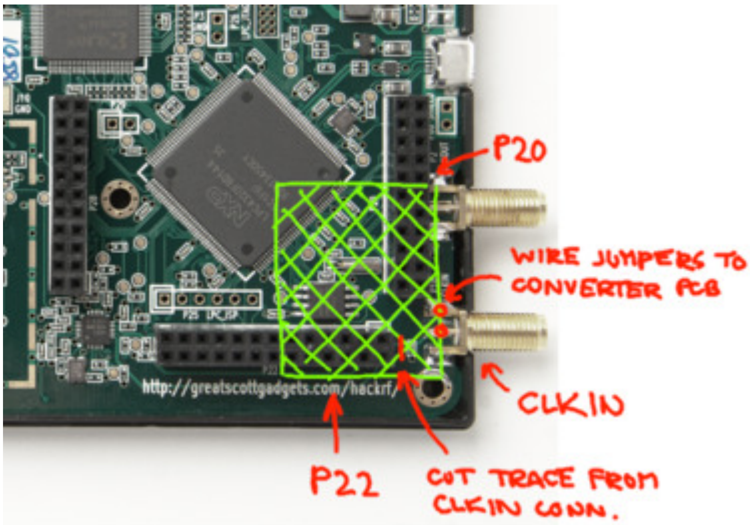


The circuit required to support the LTC6957 is quite minimal, so it wasn't hard to cram it all into a small two-layer board that will sit in the corner between P22 and P20 headers. Compared to the typical TCXO mod that only mounts onto the P22 I decided to also use the P20 header. This both makes it a bit more mechanically stable as well as gives me access to some unused GPIO lines on the HackRF's LPC4320 CPU.

I designed the input circuit to be 50 Ω terminated and hence work best with 50 Ω sources. The input is AC coupled and should work with AC or DC coupled sources. The converter should work with square wave signals with amplitudes between 0.8 V and 8 V peak-to-peak (when measured without a 50 Ω load) and sine wave signals with levels between -4 dBm and +16 dBm.

The LTC6957 has some digital inputs that affect its operation. This includes setting filter bandwidth (useful for adjusting for a sine wave or a square wave input) and turning the clock conversion on and off. osmo-clock-conv uses jumpers to configure those, but since I had GPIO lines available I simply used those. This makes the LTC6957 configurable from software. I also wanted to make sure I can power down the LTC6957 on request - LTC6957 with a floating input will likely produce a random clock signal and I don't want the SI5351 to lock onto that if I leave CLKIN unconnected. With the LTC6957 output disabled, the SI5351 should automatically switch back to its own crystal oscillator.

The LTC6957 has two identical outputs. The second one isn't used on the board, but I wired it to an AUX header in case it later turns out to be useful.



The only hairy part of this design is the fact that the HackRF offers no clean way for an extension board to sit between CLKIN and the SI5351 clock input. The P22 header only allows a board to be connected in parallel to the clock line (see the schematic at the top of the post). There is also no series element on the clock line that can be desoldered to isolate the CLKIN from the SI5351.

What I plan to do is cut the trace on the HackRF PCB going from the CLKIN connector to the SI5351 right before it connects to the P22 header. I then plan to use a short piece of coax, or simply a pair of thin wires, to connect the original CLKIN SMA connector to the input of my clock converter board. This way the external clock signal will enter through the original CLKIN connector, go through a wire jumper to the clock converter board. After conversion the signal will then go back onto the HackRF board through P22.

I also left a footprint for an edge-mount SMA connector on the clock converter board. This makes it possible to use it without modifying the HackRF PCB by having a separate SMA connector for the clock converter input. I probably won't be using that since the additional connector will not fit in existing HackRF enclosures.

I'm currently waiting for the PCBs, which should arrive any day now. I was lucky to get what appears to be the last two LTC6957-3 chips on the market, so I should be able to assemble the board and test its design shortly. I also still have to write the software. Unfortunately, the HackRF firmware doesn't provide a general way of controlling the spare GPIOs so I will have to modify and recompile it. I did some quick tests and I don't think that will be much of a problem. The latest firmware release also introduces a new *hackrf_clock* utility and I'm hoping I can integrate with that.

I'll be publishing the designs and the firmware patch after I verify that it works as intended. If you're also interested in modifying your HackRF like this, please drop me a mail. I might do a small production run of the clock converter board after the current component shortage passes if I see enough interest.