

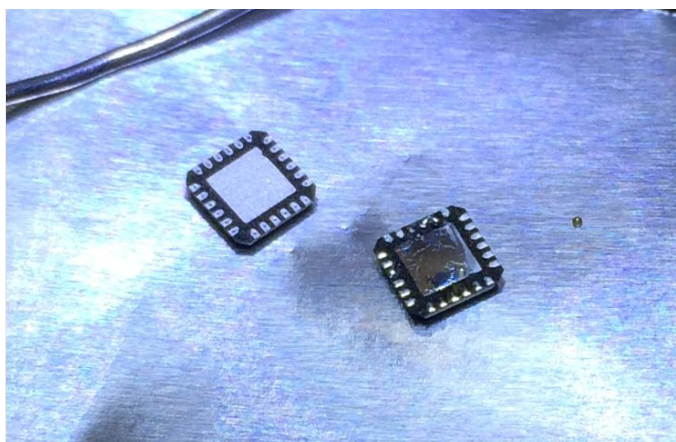
OpenVizsla assembly notes

OpenVizsla is not a good first SMT assembly project; you should have some experience with mounting SMT passives, drag-soldering QFP packages and hot-air soldering of stubborn components. With a fair bit of practice, it takes me 4-6 hours to assemble a board. I use:

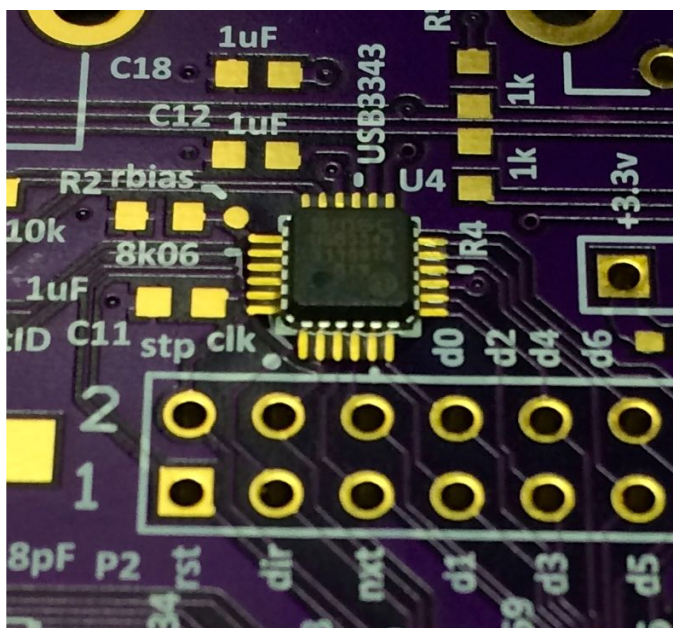
- “X-TRONIC” Hot Air Rework station ([amazon link](#))
- Amscope 7x-45x Stereo Zoom Microscope ([manufacturer's link](#))
- Rosin flux ([amazon link](#))
- Precision tweezers ([dx link](#); buy extra, because the tips get bent easily)
- Desoldering braid
- Fine-pitch solder ([amazon link](#))

Mounting the ULPI target PHY

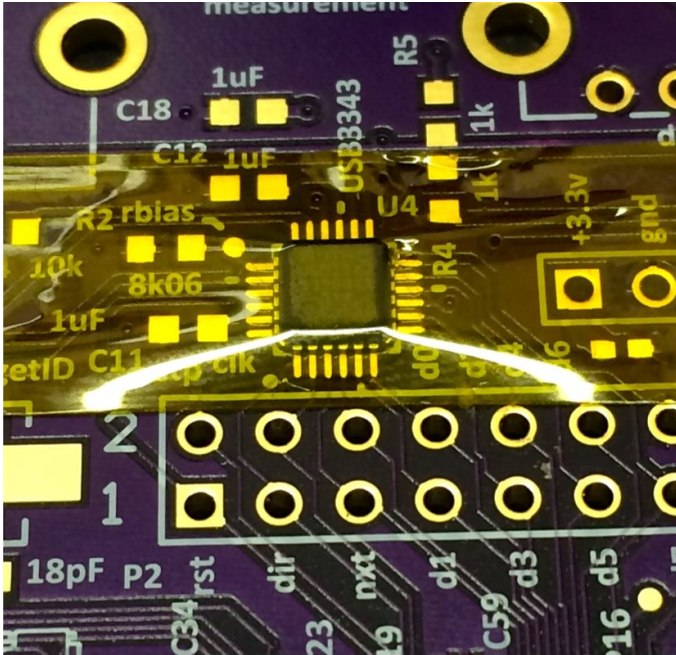
The most difficult part of the assembly process is mounting the QFN target PHY. Though it does not have many pins, it has a big ground pad underneath it that is impossible to reach with a soldering iron. If you've been looking for an excuse to buy a hot air workstation, this would be a good one; many are available for less than \$100 on Amazon. You will also need some rosin flux and some thin Kapton tape. There are many different ways to do it; here is the one I used.



Flip the chip over and use a soldering iron to apply a very little amount of solder to the center ground pad. Apply as little as possible; you might want to just apply some and then remove most of it with solder wick. If there is too much solder, then you won't be able to get the chip to sit flat on the PCB and then you will have trouble soldering the pins.

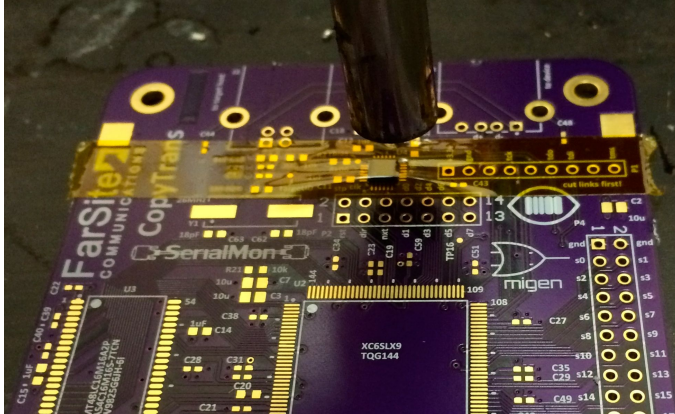


Take the PCB and apply a small dab of rosin flux on the center of the chip footprint, where the pad will go. Carefully place the chip on the PCB, aligning it so that all the pins line up with the pads on the PCB. It doesn't have to be perfect, because you will end up fine-tuning it in the next step. Make sure that you orient the chip correctly — with the host side of the PCB facing you, the dot on the package should be on the bottom-left, and the “SMSC” text on the chip should be readable.



Tape the chip in place with the Kapton tape. The chip will inevitably move a little bit when you do this, so go back in with some tweezers and push it back in to place.

Using Kapton tape is important here, because it will not melt when we solder down the chip. We can actually apply heat through the tape. (This tape is available cheaply from DealExtreme — <http://www.dx.com/p/high-temperature-tape-5mm-200-c-5396> should work.)



Use a hot air pencil to heat the chip through the tape. I used a setting of 450 degrees C for 30 seconds.

Afterwards, let the board cool for a minute, taking care not to disturb it until the solder hardens. Remove the tape. Look at the chip from the side; it should be flat against the board, and all pins should line up with the pads. If not, use the hot air pencil to remove the chip and start over. It often required 2 or 3 tries for me, even with practice.

Once the ground pad is soldered down, you can apply flux around the chip and use an iron (with a broad, flat tip) to drag-solder the pins. You may also want to use hot air to mount L1, the switching regulator inductor; it is also difficult to solder with an iron.

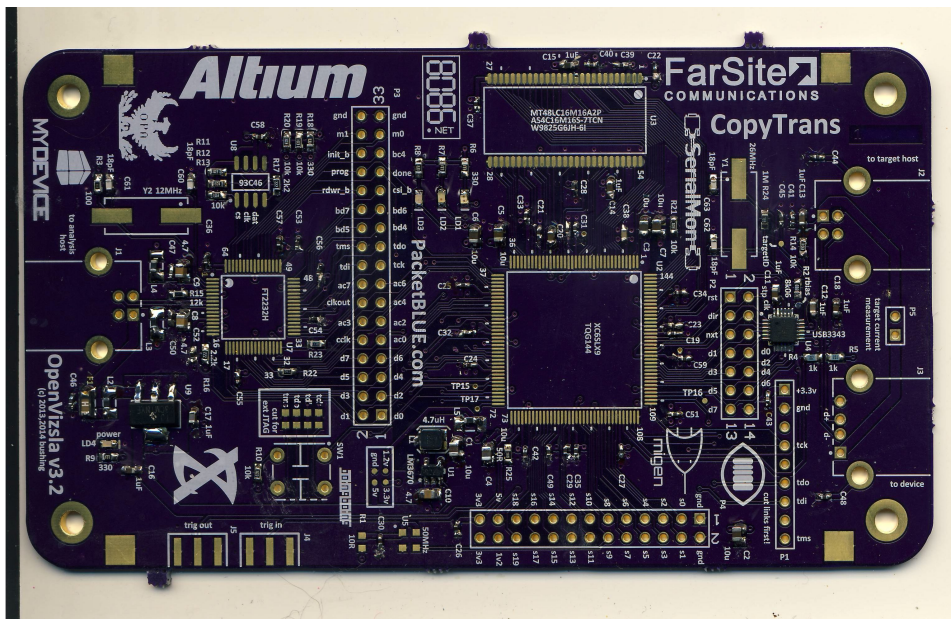
Small components and power supply

Next, using a narrow-tipped iron, solder down all of the small parts (resistors, capacitors, ferrite beads, inductors, LEDs, fuse). Every part on the board is marked with its value, except for the 0.1uF 0402 capacitors. These are difficult to hand-solder; in many cases, you may be able to get away with using 0603 parts (it will be a tight fit on the pads).

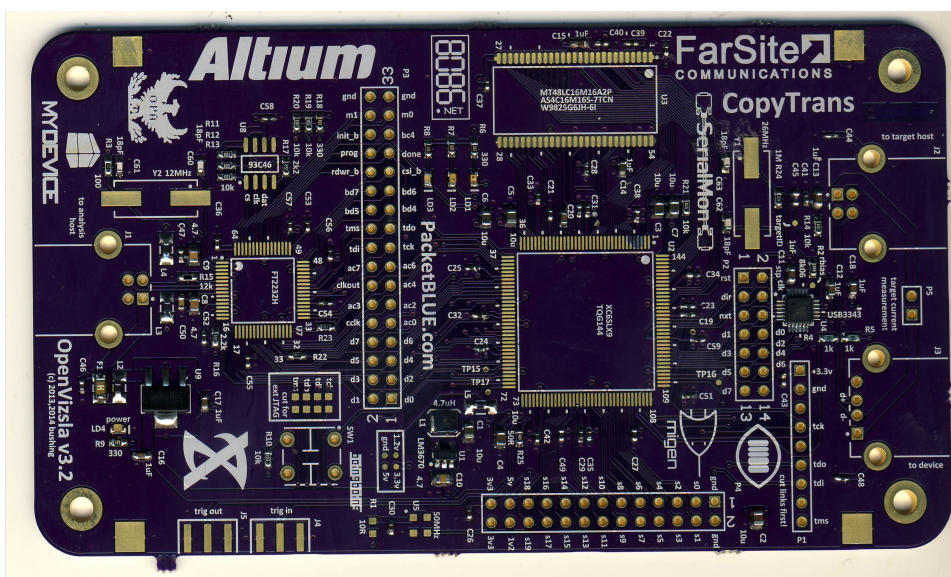
I like to solder these by tinning one pad, placing the part using tweezers and a microscope, and then melting the solder on the tinned pad. I then go on to the next part. At the end, I go back and double-check that I haven't missed any parts, and then solder down the other half of each part. You should also solder down the power supply chips U1 and U9 (the +1.2v regulator and the +3.3v regulator). Watch the orientation of the LEDs — most LEDs have some sort of marking on the negative end. If not, use a multimeter with a “diode” setting, and apply it to the two LED sides — when you have the negative and positive leads connected correctly, the LED should light up (dimly). Once it does, the side you probed with the negative DMM probe should go towards the silkscreen line on the PCB.

After you finish, I recommend applying flux to all of the parts you soldered down and then going over each with the hot air pencil until the solder melts; this will cause the parts to center themselves on the pad and will make for a neater end result.

Before:



After:



Once you have double-checked your work, make sure that all resistors and capacitors have been installed, except for possibly the following optional parts: R1, C30, R4, R5, R25, R26 (on other side of board). Solder on J1, the host USB jack; resist the temptation to solder on the other USB jacks, because they will make rework of the ULPI PHY more difficult.

Plug the USB cable into your computer; if you get a USB overcurrent error, check your soldering. Otherwise, use a DMM to measure the +5v, +3.3v and +1.2v rails; all should be within 10%. The green Power LED (LD4) should light up. (If the Power LED ever fails to light up, disconnect the board immediately and carefully touch the PCB around U9; if it's burning hot, you have +3.3v shorted to ground somewhere.)

If these work, you can proceed to install the rest of the parts with less risk of damaging them due to incorrect supply voltage.

FTDI

Next, solder down U8, the EEPROM, and Y2, the 12MHz crystal. Switch to a wide soldering iron tip, and then tack down opposite corners of U7, the FTDI chip. Apply a generous amount of flux around the chip, and then drag-solder across all of the pins. (Search on YouTube for “drag soldering” if you are unfamiliar with the technique.)

Once you are done with this, test it by plugging the USB cable back into your computer. The device should show up with the FT2232H default descriptor — check USB Prober on OS X, lsusb on Linux, and the Device Manager on Windows. If it does not enumerate, check your soldering on the FT2232H and the 12MHz crystal; if it still does not enumerate, check to make sure you have a 12MHz signal on the pins of Y2, and proceed from there.

Once you have gotten the FT2232H to enumerate, it's time to program the OpenVizsla USB descriptor into the EEPROM. Use `“ovctl.py -p ov3.fwpkg eep-program 10000”`, or whatever your serial number is (check sticker on bottom side of board). If you are running OS X, you may need to unload the FTDI kext first using `“sudo kextunload -b com.FTDI.driver.FTDIUSBSerialDriver”` first; you will only need to do this to reprogram the EEPROM. Once it has been programmed, the FTDI driver will no longer recognize it.

FPGA and RAM

Now it's time to solder down the FPGA, using the same corner-tacking / drag-soldering technique mentioned above. Triple-check the orientation of the FPGA before soldering it down — there are 3 dots on the chip in the corners. Pin 1 is the smaller of the 3 dots. If you rotate the board counter-clockwise (“Altium” logo will be upside-down), then the SPARTAN-6 logo should be right-side-up.

Solder down the RAM chip — text should face the same way as the FPGA. If you haven't already, solder down Y1, the 26MHz crystal, and SW1, the switch.

Once you have soldered those down, connect it to your computer and run ‘python3.3 ovctl.py -p ov3.fwpkg report’. You should see output similar to:

```
$ python3.3 ovctl.py -p ov3.fwpkg report
FPGA not loaded, forcing reload
FPGA: Bitstream timestamp 2014/03/09 22:34:23
FPGA: sending configuration bitstream
FPGA: CRC OK
FPGA: configured
USB PHY Tests
  ULPI PHY ID: 04240009 (SMSC 334x)
  ULPI Scratch register IO test: OK
  PHY Function Control Reg: 41
  PHY Interface Control Reg: 00
SDRAM tests
  ... 0: OK
  ... 1: OK
  ... 2: OK
  ... 3: OK
  ... 4: OK
  ... 5: OK
  ... all passed
```

If you don't get the “FPGA: configured” message, check your supply voltages again, and then check your soldering on the FT2232H and the FPGA (especially the pins closest to the FT2232H).

If you do not get the “ULPI PHY ID: 04240009” message, you probably have a problem with your soldering on the ULPI PHY; remove it with hot air and resolder.

If you do not get the “SDRAM tests ... all passed” message, check your soldering on the FPGA side closest to the RAM, as well as the RAM itself.

Wrapping up

If everything checks out, go ahead and solder down the two target USB connectors. At this point, you should have soldered down everything except for the following:

- P1, P2, P3, P4 — headers only used for debugging and development
- P5 — only used for target current measurement, and you must cut a trace and solder a resistor down (R26) for this to be useful
- J4 and J5 — reserved for future expansion (external trigger input)
- R1, C30 and U5 provide a 50MHz clock, which is unnecessary in the final design
- R4 and R5 provide a direct USB signal tap for the FPGA, currently unimplemented

To test USB sniffing functionality, plug a second USB cable into J2 and your computer, and plug a USB flash drive into J3. The flash drive should enumerate. Run 'ovctl.py -p ov3.fwpkg sniff hs', and you should see output like

```
$ python3.3 ovctl.py -p ov3.fwpkg sniff hs
[ F      ] 0.077749 d= 0.077749 [ .0 +77748.717] [ 0]
[      ] 0.308898 d= 0.231149 [ 20.7 + 11.700] [ 3] PING : 28.2
[      ] 0.308899 d= 0.000000 [ 20.7 + 12.183] [ 1] ACK
[      ] 0.308901 d= 0.000002 [ 20.7 + 14.667] [ 3] OUT : 28.2
[      ] 0.308901 d= 0.000000 [ 20.7 + 15.000] [ 34] DATA0: 55 53
42 43 15 02 00 00 00 00 00 00 00 00 00 06 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 58 2b
[      ] 0.308902 d= 0.000001 [ 20.7 + 15.917] [ 1] NYET
[      ] 0.309028 d= 0.000126 [ 21.0 + 16.900] [ 3] IN : 28.1
[      ] 0.309029 d= 0.000000 [ 21.0 + 17.367] [ 16] DATA0: 55 53
42 53 15 02 00 00 00 00 00 00 00 00 2a fb
```

55 53 42 43 is “USBC”, a command response from the USB device. 55 53 42 53 is “USBS”, an outgoing command.

For reference, here is what my finished unit looks like. You may not have R4, R5 or C30 soldered, which is fine. You will also need to solder SW1.

