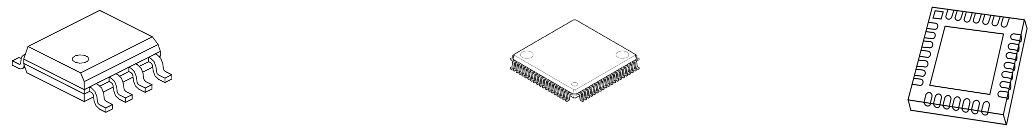
THINGOTRON
LABS
http://www.thingotron.com

Stencil Reflow Soldering

Or, how I learned to stop worrying and love Surface-Mount Technology

Electronics is a wonderful hobby, and one can do an enormous amount with regular old through-hole soldering. But more and more exciting parts these days come only in SMT packages like SOIC (Small-Outline Integrated Circuit), QFP (Quad Flat-Pack) and QFN (Quad Flat, No-leads).



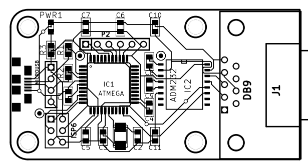
More basic components (sometimes affectionately called "jellybeans") come in standard sizes; a good fit for the hobbyist is 0805 (0.08 in by 0.05 in, or about 2mm by 1.25mm), though some like 0603. At these sizes, each component is about the size of a grain of rice, taking up very little space, costing relatively little, and often squirting from tweezers like a watermelon seed, never to be seen again; use caution, lots of light, and a clean workspace.

SMT components generally come in tape-and-reel packaging, and some distributors (Farnell and Digikey come to mind) have "re-reel" services so that you can buy a partial reel. Even for the hobbyist, reel packaging is useful for organization; search ebay for "SMT feeder" and you can find useful organization racks for DIY use.

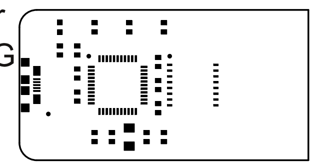
One additional benefit of SMT technology is that it eliminates the "drill" step necessary for PCB production, which can simplify things a great deal (vias and two-sided boards are still problematic)

Obtaining a Stencil

If you're producing more than 1-2 boards, using a stencil and solder paste is much more efficient than hand-soldering SMT components. Most circuit design programs such as KiCad allow export of



your design in multiple layers. The important layer for us is the "Paste" layer; on the left you can see an SVG of the complete circuit, while the right shows only the SMT pads which have been picked out for solder paste. Design rules can make the paste areas smaller

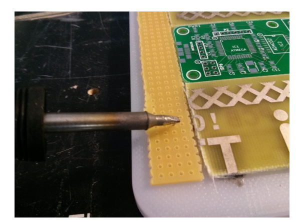


than the actual pads, which can be a very good thing if too much solder becomes a problem. Stencils can be obtained from several vendors; metal stencils can be fiendishly expensive, but 0.1mm PET stencils are inexpensive and darned effective; an A4 sheet from smtstencil.co.uk, for example, costs about GBP 15.

Making a Fixture

To hold your PCB and stencil, it's very helpful to make a fixture which surrounds your PCB to be stenciled with an area of "junk" PCB the same thickness. This keeps your target board and stencil stationary, but the real advantage is that it keeps the area around the PCB the same height, so the stencil doesn't flex near the edges and blur the paste.

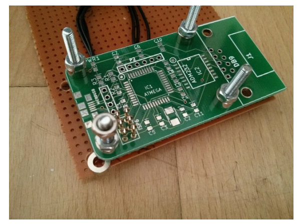
I tried a few ways to make stencil fixtures, including epoxying PCBs to a board; this works, but isn't great because the epoxy itself adds a height difference. Probably the easiest thing I've found to do is take a plastic cutting board and junk pieces of perfboard or old PCBs. Clip a small section of stiff wire (headers work) about twice as thick as a PCB; place them into a hole in your perfboard/PCB, heat with a soldering iron, and let the wire melt into the cutting board; your perfboarf/PCB will be held fairly securely and at the exact height you need.



13 Inspect. Using a microscope if you need, look closely at all fine-pitch areas, particularly the traces under the USB connector and all four edges of the AtMega32u4. If you see any visual bridges, use flux and a soldering iron or solder wick to clean them up.

14 Quick test: use a multimeter to check for shorts. The test ports behind the USB connector are placed to facilitate testing shorts between VCC and Ground (outer two holes), between the USB pins, and between the USB pins and the USB connector housing. If these look good, try programming!

15 Program the board using the ISP header. If you're programming more than a few boards, it's useful to make a "PogoPin" test fixture using perfboard and a sacrificial PCB; this way, you can program the boards without soldering a six-pin ISP header--much more efficient for multiple boards. I use AVRDude and a USBTiny to flash everything from an image.



16 Pre-test: At this point, I usually plug the board into a computer using the MicroUSB connector to make sure it powers up and appears to the operating system as a USB HID device. If it does, it's time to add the last piece, the through-hole RS232 connector.

17 Solder in the RS232 connector. I couldn't find an inexpensive male DB9 connector, so this is the only through-hole part in the device. Honestly, I like connectors like this to have through-hole pins, even non-circuit pins just for stabilization and strain relief.

18 Plug a SpaceOrb 360 into the serial port, and the USB port to the computer, and do a final check. If the axes wiggle and buttons press, your Orbotron 9000 is complete. Congratulations!

Well done! We hope you enjoyed the workshop. The Orbotron 9000 is a bit of a niche product but it's lovely seeing old devices work again (and someday it may support even more devices). If you want to keep the one you built and buy a serial SpaceOrb 360, SpaceBall 4000FLX, SpaceBall 5000, or Magellan Spacemouse off eBay, you can certainly purchase your Orbotron at a substantial discount (and optionally a perspex "sandwich case"). If not, thanks for helping build one!

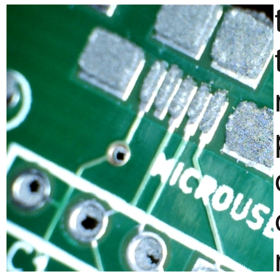
For more information on the Orbotron 9000, visit our wiki at http://wiki.thingotron.com or the forums at http://forums.thingotron.com.

For this class, we'll be assembling the Orbotron 9000, an interface board that acts like a "hardware driver" for old RS232 serial devices. Programmable with the Arduino environment, this allows a sketch to interpret an old serial device (in this case the Spacetec SpaceOrb 360) as a USB HID device (in this case a six-axis, sixteen-button joystick). It uses all SMD parts: a QFP Atmel AtMega32u4 (like the Arduino Leonardo), a SOIC ADM232AARWZ to interface with the RS232 serial port, and a host of 0805 "jellybeans"--capacitors, resistors, and the like.

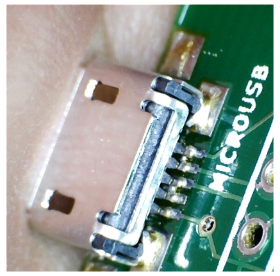
1 First, place the board on the stencil fixture and make sure everything is as flat as possible. Position the PET stencil as accurately as possible; check it with a magnifier if necessary!

2 Place a small bead of solder paste near the top of the stencil (it won't take much at all!) Holding the stencil firmly in place, use the squeegee or razor blade to gently smear the paste across the stencil and scrape clean. It's often helpful to wipe at a 45-degree angle to rows of pins, but careful how much force you use or you could force paste outside the solder pads. Don't let the stencil flex!

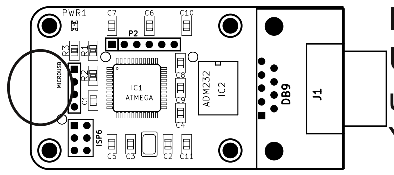
3 Inspect your work. This is a critical step. It's tempting to think that solder mask will prevent bridges, but it's far better to wipe the PCB and stencil clean with alcohol and try again than



to bridge lots of connections and force difficult rework after the fact. Keep in mind that being off by just a little bit can result in solder bridges that are very annoying to fix, particularly if they occur under a difficult-to-access component like the USB connector! Careful attention here can prevent a lot of headaches later. Is it good? Carefully remove the PCB and start placing parts!

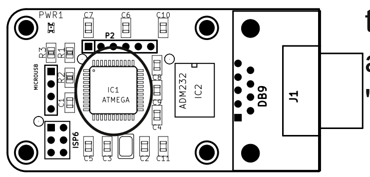


4 I generally try to place the hardest parts first while I'm fresh, so that's the order I'll go into here. If you'd rather start with the easier pieces, feel free to go the other direction!



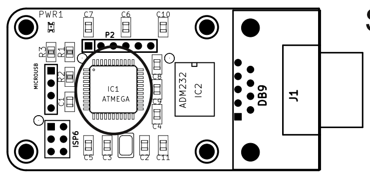
Because it's by far the most annoying part on the board, I start with the USB connector. There are no great markings, so you just have to line it up by eye (a lesson: choose connectors based on ease of assembly!) You may need a magnifier or microscope for this step.

5 Next hardest is the AtMega 32u4. Again, take your time and line it up as precisely as possible, making sure to orient it so that the dot marking pin 1 is in the right place (top left if



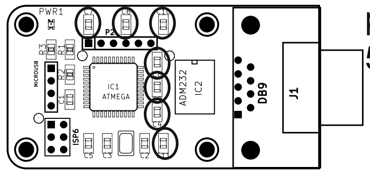
the PCB is oriented this way). If you need, you can move the chip around once dropped, but try to avoid smearing the paste. Watch for "skewed" chips; line everything up as precisely as you can!

6 After that, the SOIC is easy. Again, align pin 1 with the dot on the board (top left as depicted here). You'll find the SOIC a lot easier than the USB connector or QFP chip;



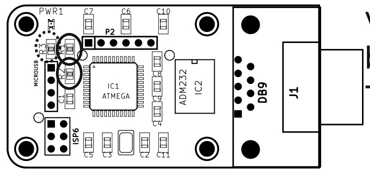
SOICs are actually pretty easy to hand-solder.

7 Time to start placing jellybeans, in no particular order. I start with the seven 0.1uf capacitors, placed at C4 and C6-C11. These are used as decoupling capacitors to smooth



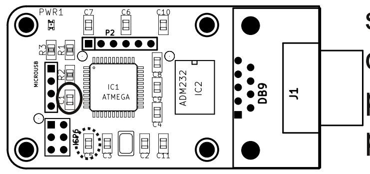
power to the chips and as charge pumps for the ADM232 to convert the 5V VSS power to the 9-12V necessary for old serial devices.

8 Now the resistors. You have two 22 ohm resistors (small, black top, labeled "220" in very tiny letters and "22R" on the back), which go in R1 and R2 (solid lines); these change the



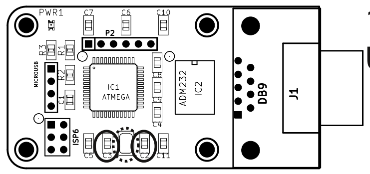
voltage on the USB data lines. There's also a 1k resistor (unlabeled but there's only one, black top, tiny letters) which goes R3 (dotted line). This regulates current through the power LED.

9 More capacitors! There's a 1uf (clear tape, marked "1u") smoothing power to the USB section of the chip) in C1 (solid line). A chunkier 10u (marked 10u) is intended to smooth



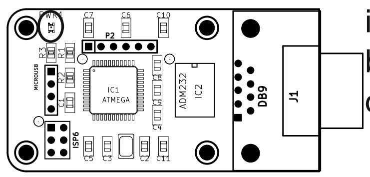
supply power, but probably unnecessary since USB power is pretty clean. This goes in C5 (dotted line). On some schematics this is a polarized electrolytic--if you don't have to use those, don't! It's easy to put them in backward, resulting in--literally--a smoking hole.

10 More capacitors and the crystal. Two 30pf capacitors go in C2 and C3 (solid line); these provide loading for the crystal (X3, dotted line), which keeps the ATMega running at



16MHz. A crystal provides very precise frequencies, necessary for USB communications to work.

11 Finally, the power LED (PWR1, solid line). This is the only "jellybean" component that must be placed in the correct orientation. On the BOTTOM of the LED is a triangle



indicating direction of current flow and pointing to the cathode. For this board, that triangle must be pointing UP -- but of course on the bottom of the component where you can't see it.

12 Bake! Using either the reflow oven or a controlled hot plate (I use a cheap hot plate and the "Kettletron 9000", a homemade PID controller), raise the temperature of the board to about 160C to preheat, then up higher for a peak (for my homemade controller, I shoot for about 240C, but watch closely to cut the power once things reflow).

The surface tension will help align parts as the solder paste melts into shiny liquid. Once the board has cooled, it's time to test!