Essential Tips and Tricks

What you will learn: In-house tips and tricks I learned along the way that I'm sure you'll find useful as well.

Introduction

Every engineer requires good tools and best practices to survive. They not only make the job much easier, but more importantly, they allow the engineer to be highly accurate and consistent. For example, imagine having to solder on something WITHOUT a soldering iron. How in the world would you do that?! I'm not an engineer myself but I'm going to share with you few essential tips and tricks I learned (googled) along my Arduino journey that helped me finish the project and save me weeks of trouble. I'm hoping you'll find these tips useful as well.

1. Multimeter

Multimeters are by far the most useful and most important debugging tool you'll have while finishing this project (or any electrical engineering project). If you don't have one at hand, stop reading this right now and order one on Amazon. Seriously. It's that important. (link)

Multimeters can measure a lot of things including continuity, resistance, voltage, current, capacitance etc. But the most useful functionality (in my opinion) is the continuity / diode testing mode. This functionality allows you to check if a diode is working or not (which has directionality), and whether two points in a circuit are electrically connected or not. The multimeter will beep (output of a piezo buzzer) if the two points are connected and will light up the LED if the cathode is connected to GND and anode is connected to PWR. But always remember to turn the knob to continuity / diode testing mode before testing anything! Otherwise the multimeter would be powered off…

Figure 1: Continuity test mode shown on the multimeter (white circle). The value on the display indicate the resistance value. Value "1" signifies that it's an open loop.

Figure 2: Close up view of the continuity / diode test option

To provide more perspective, continuity testing is extremely useful in verifying your solder joints either on the PCB or on any other wires. If you've soldered something on, probed the two solder points and got a beep, that means you have a short circuit. This means you'll have to redo the solder connection, but this would have saved you a ton of time in the future when you're running your software on the circuit board and the hardware component is not working as you expected (well, because it's a short circuit!).

Below is a summary I found on Adafruit.com on various things the continuity test can be useful for:

What is it good for?

Continuity is one of the most important tests. Here are some things it is good for

- Determine if your soldering is good. If your solder joint it is a *cold solder connection* it will appear connected but in actually it is not! This can be really frustrating if you are not experienced in visually detecting cold solder joints
- Determine if a wire is broken in the middle. Power cords and headphone cables are notorious for breaking inside the shielding, it appears as if the cable is fine but inside the wires have been bent so much they eventually broke.
- Making sure something **isn't** connected. Sometimes a solder joint will *short* two connections. Or maybe your PCB has mistakes on it and some traces were shorted by accident.
- Reverse-engineering or verifying a design back to a schematic

Figure 3: Source - https://learn.adafruit.com/multimeters/continuity

Using the multimeter in the lab

Using the multimeter on the PCB is fairly straightforward. If you want to verify the connections between two points, you can probe it using the clips. If It's connected, you should hear a beep. Using the multimeter on LEDs are simple as well. Once you connect them correctly, (cathode to

GND / anode to PWR) the LED will light up (you will not hear any beeps). For IR LEDs, you won't "see" it light up because human eyes can't detect infrared light but I've explained how you can "see" the IR in this **post.** For solenoids, it's a little different story. If you test out the two leads, although the two leads are not touching each other, the multimeter will give you a beep. How could that be? Is this a short circuit? Did the solenoid valve get broken before we even used it?

Figure 4: Red arrow signify PWR and the black arrow signify GND

Quite the contrary. Remember back in the solenoid valve section that the solenoids are continuous cylindrical coil of wire? Well the two leads you see in Figure 4 are actually two ends of the same wire. The cylindrical coil is hidden inside the component. The exact resistance of the coil will depend on a lot of things, but it will tend to be very low and seem like a short circuit. If they don't have a low resistance, then it's likely that the wire has broken internally, and the valve is dead. So to sum it up, the "beep" you hear when you test out the solenoid valves means that they are working!

2. Breaking off Pins

Breaking off header pins (especially female types) correctly is surprisingly difficult. If you try breaking them off with a wire cutter, you'll notice that often times you end up with one less header than you intended because the encasings break off. Thus, this method is very unreliable when you want to get a specific length of the header pins.

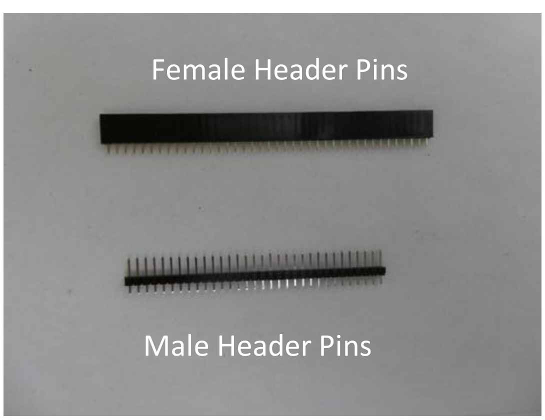

Figure 5: Two types of header pins – Female and Male pins.

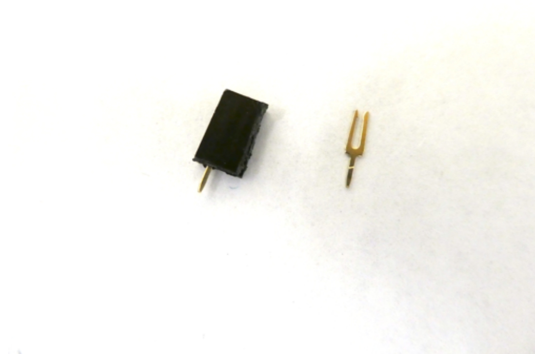

Figure 6: Improper way of breaking off pins. Notice that the header pin has fallen off from the encasing.

In this section, I'll teach you how to reliably get any number of pins you want. To do this however, we need to sacrifice one pin every time. In this example, our goal is to get a 1 x 4 header pin.

As a first step, you need to remove the (n + 1)th pin. You can see from the yellow arrow that the 5th pin has been removed since our goal is to get a 1 x 4 pin.

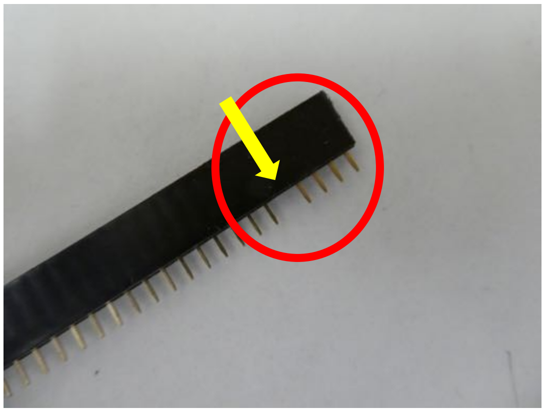

Figure 7: Female header pins after removing the (n+1)th pin. This location will later serve as the spot to make etched guidelines.

Next, you want to make "cutting guidelines" into the 5th encasing. The encasings are made of plastic so it's pretty easy to do. You can see the etched guidelines in the red circle. Make these guidelines on both sides of the encasing. After making the guidelines, you can just snap them off. The header will easily snap off along the guidelines you have formed.

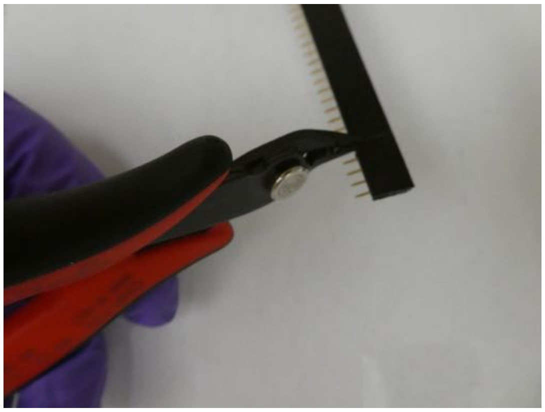

Figure 8: Making the guidelines with a wire cutter

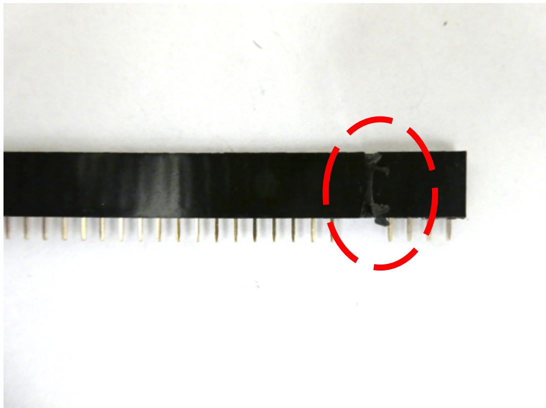

Figure 9: Notice the etched guidelines at the empty pin location (red circle)

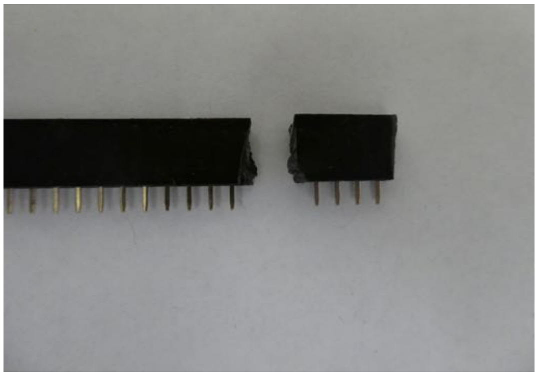

Figure 10: We have successfully made a 1x4 female header pin! Compare and contrast this to Figure 6 where we failed to properly break off the pin size of our interest.

As with other components, you should sand down these headers using the belt grinder to give them a cleaner, smoother finish. The smooth finish will not only give the finished product a more professional look but also allow other components to fit better on the board.

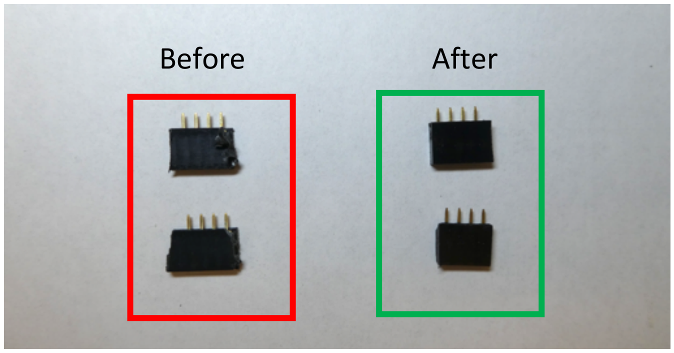

Figure 11: Before and after sanding down the headers. Notice the smoother finish in the 'after' products

3. Soldering Components to PCB

When soldering anything onto a PCB, you want the components to be flush with the board. Simply put, you don't want your components to look like the ***X***-marked headers in Figure 1. This is surprisingly pretty tricky to do because headers tend to move around a lot when soldering. As you go through the pins in a serial fashion, you'll quickly realize one small misalignment will lead to the misalignment of the whole component (like in Figure 1) because solder joints cool down and harden very quickly. In this section, I'll show you my tips on how to make the components flush with the board.

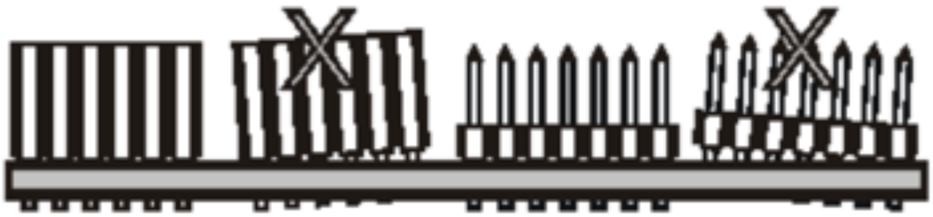

Figure 1

Examples of good soldered (first and third) and bad soldered headers (second and fourth)

The basic idea is to affix the components / headers flush with the board temporarily with some tape and solder key anchor points to the board. Once you have the anchor points, you can take the tape off and solder the remaining leads to the board. I'll use the stacking pins as an example to demonstrate this method.

<u>Step 1:</u> Use tape to fixate the components / headers to the shield so that it doesn't fall off even when the board is upside down

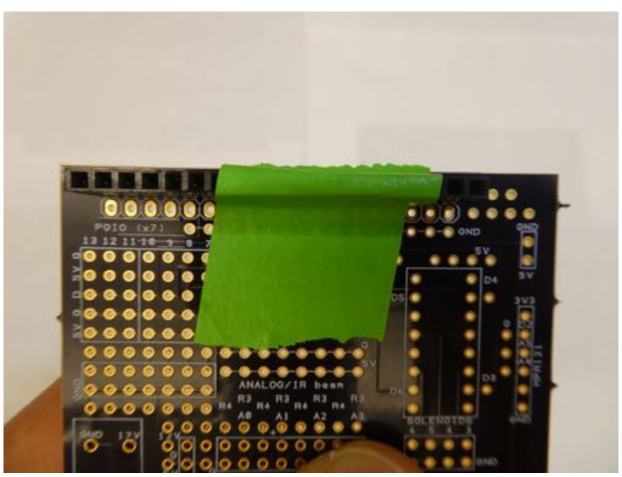

Figure 12: Using tape to affix the stacking pins to the shield. Top-down view

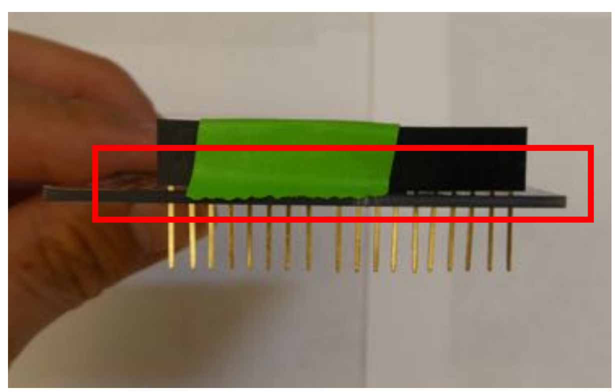

Figure 13: Side view of the shield. Note that the stacking pins are not yet flush with the shield (red box).

<u>Step 2:</u> With the tape attached, solder just the two ends of the stacking pins so that you can take the tape off. These will be your anchor points.

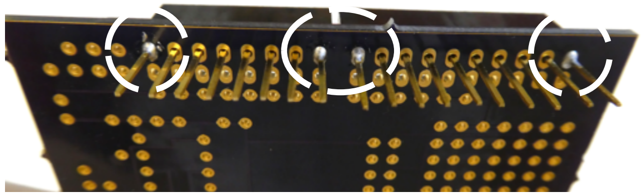

Figure 14: Stacking pins after the anchor points (white circles) have been soldered on.

<u>Step 3:</u> Note that step 2 will not anchor the stacking pins perfectly flush with the shield (red box). To anchor the stacking pins flush with the shield, heat up your anchor solder joints and push in / wiggle the stacking pins against the shield until you feel that the stacking pins are flush.

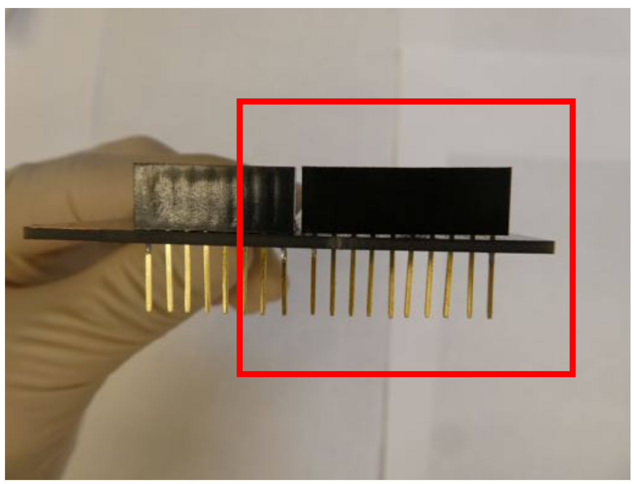

Figure 15: Note that the stacking pins are not yet flush with the shield (red box).

<u>Step 4:</u> After adjusting the stacking pins and your anchor points, it's time to solder in the remaining holes. If you feel that the stacking pins are getting misaligned as you solder, repeat step 3 so that the stacking pins are actually flush with board.

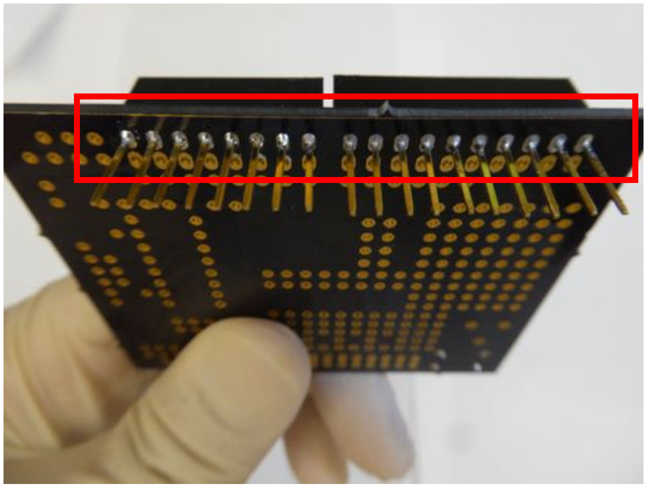

Figure 16: Bottom view of a finished solder work. Note all the solder connections (red box).

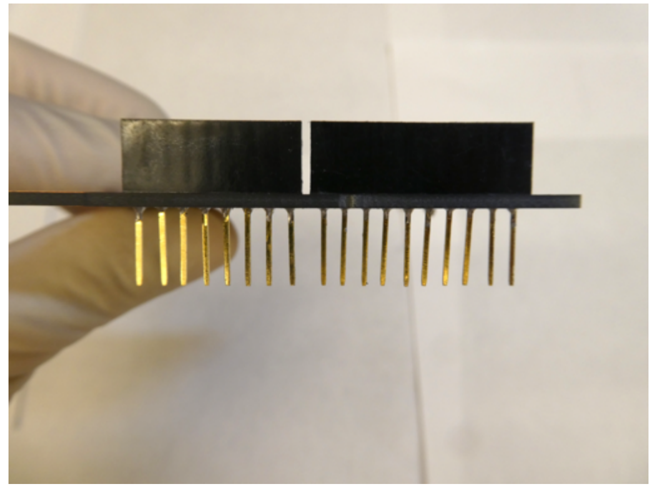

Figure 17: Side view of a finished solder work. Notice that the stacking pins are flush with the shield.