ME106/Sp2016 Build Instructions for EduShields PortMaster v2.1-v2.3(.1)

Introduction to the Introduction

This semester's PortMaster build is going to be different from past semesters' build procedures. This is due to the fact that we're going to have you build leftovers from the previous (at least) two semesters' board stock, which are a little different from each other.

Although the build is 90% the same regardless of whether you're building a v2.1, v2.2 or v2.3 board, the v2.3 version of the board has a bug in it which requires that you hand-solder an extra piece of wire on the board to get it to work. This connection was on the board in v2.2 and v2.1, but was accidentally removed in v2.3.

While it might seem obvious to run and grab a v2.2 or v2.1 board to build, there's one reason why you might want to build a v2.3 board, other than simply getting the experience of soldering a patch-wire on a PCB (in fact, a very useful skill to have). In particular, the v2.3 board contains a header into which you can plug a very cheap ($4.50) 3-axis accelerometer board, which can make using your PortMaster even more interesting to play around with – and which you may also choose to use for some part of your final project. At some point in the near future there may be a video that demos this in action, so look for an announcement of this in the next few days/weeks.

If for some reason any v2.3.1 boards end up in the hands of students this semester, they should not have the bug that the v2.3 board does, sand o they can be built without any fix-wire.

Introduction

The following describes the recommended build procedures for the EduShields PortMaster v2.1-v2.3 for the Arduino. This document does not attempt to be an instructional treatise on soldering technique or the theory of operation of the board. It only provides assembly guidance directly applicable to this and peripherally applicable to other Arduino shields. This document also only describes the “through-hole” component build of this board[1].

Bug Summary on the v2.3 board: On the v2.3 board there's a missing trace from the resistor packs to ground (GND). Neither the v2.1 or v2.2 boards from previous semesters have this issue. The fix is documented at the end of the build instructions.

Bug Summary on the v2.1 board: Note that the leftmost button on the board is labeled SW12, suggesting that it is connected to Arduino Pin 12 *which is not the case*. This label should read SW11, as Pin 11 is the pin to which this button is connected.

Photographs “Bug” Note: Unfortunately I only found myself with one of the two needed resistor SIP packages when it came time to take the pictures for this document. Therefore what you're going to see in the photographs below is only one SIP. In place of the other resistor pack I used five individual SMD resistors in its place because I wanted a complete working board when I was done. ***In the pictures throughout, you will only ever see one resistor SIP soldered-in where you should otherwise expect to see two!***

1 For students with a decent amount of soldering experience that want to build a board using surface-mount LEDs and resistors can contact me directly about trading me for a set of SMD parts.

Soldering Technique Resources

The following URLs point to the best video through-hole soldering instructions that I found up until mid-2014. Those without any soldering experience *must* study one or more of them before attempting to solder anything. In combination they all provide fairly complete coverage of theory, basic instruction and safety issues related to electronics soldering.

Curious Inventor: through-hole, SMT
SparkFun: through-hole, SMT
Dave Jones: tools, through-hole, SMT

Soldering skill is acquired with practice, but not understanding the theory and proper technique *really will* result in sub-standard work. Doing the work right the first time will save you substantial time later debugging and repairing a non-working (or worse, an intermittently-working) board. Even trivial soldering errors take at least ten times as long to repair, and each attempted repair carries some risk of degrading or ruining the board and/or its components. Do your best to get it right the first time and you will save yourself time, money, and frustration.

Required Parts and Equipment

Blank PortMaster v2.3, v2.2 or v2.1 PCB
0.1” single-row male headers, perhaps already split into shorter segments
Two (2) through-hole tactile, momentary pushbuttons
One through-hole10-segment LED bar-graph display
Two 6-pin 470 ohm (or greater) SIP resistor arrays
Optional through-hole 17mm piezoelectric speaker and current-limiting resistor
Optional 5-pin through-hole female header for optional ADXL335 module
Small length of bare (formerly insulated) wire for “the v2.3 bug fix”

An Arduino (or suitable alternative) to use as a work-holding jig, or reverse-clamping tweezers
Soldering iron with suitably medium-sized tip
Solder for electronics soldering (only!) -- 60/40, 63/37 or not recommended for beginners) lead-free
Soldering iron tip cleaner – either a wet sponge or bronze wool
Eye protection (solder flux sputters when heated!)
Some type of fume dispersal/extraction mechanism and/or open, moving air
Lint-free wipe or cloth and acetone or isopropyl alcohol (semi-optional)
One or two gloves (latex/nitrile/cotton, suggested but optional)
A PCB vice, “third-hand” or some other type of work-holding arrangement, or a brave/naive/asbestos-handed assistant with steady hands

Pre-game Show

Many soldering construction problems can be avoided by eliminating distractions, having sufficient light and appropriate tools, and having the patience and discipline to orient/hold your work properly at all times. If all of these stars are not aligned, you probably shouldn't waste your time and materials soldering. Either get someone else to help you, or wait until you're better equipped to do so. For example, I personally do very little soldering in E125 because often there's not enough light for me to see what I'm doing. If I know that I'll be soldering there, I bring along my own work light. Many of you may not have this particular problem, but instead may lack the confidence or an ideal set of tools to use. Get some junk components and PCBs from the scrap box or your TA and practice until you're reasonably confident that you can accomplish this task successfully.

The build order specified in this document is definitely not the only way to assemble this board, but should be the easiest. For more complex boards, it sometimes takes building (at least) half-dozen boards before one comes up with the best strategy for getting the job done with the fewest annoyances or risks. Having better/proper tools at your disposal gives you more options and leeway in how a board is assembled. You generally will only learn this through experience, and by having (or not having) multiple options when doing an assembly based upon the tools you have on-hand.

On this board, with only the exception of the pin headers, all of the components are loaded onto the top side of the board, and soldered on the bottom. This is fairly typical of many boards, and especially those using primarily through-hole components. Note that "top side" of the board is the side containing the labels and component placement outlines.

The general strategy for assembling a board is to work on the board from the center of the board to the outside edge, from lower-to-higher profile parts, and smaller-to-larger parts. By starting near the center of the board you allow yourself multiple attack angles with the soldering iron and the reverse-clamping tweezers. By dealing with smaller parts first, you're also less considerably less likely to be blocked from the sides or above by taller components that have already been added.

At some point during the build process, a board may no longer flat on the table surface. At this point you should find a way to anchor the board so that it remains steady as you're soldering it. One of the better ways to get frustrated is trying to solder the board while it's rocking back-and-forth or skidding around the work surface. Find a way to keep the board steady while you're working on it.

Optional Features and Add-Ons

The optional parts are a piezo speaker and its series resistor, and on the v2.3 board the five-pin accelerometer header. If so inclined, you can purchase these parts (either from me, or getting them yourself) and add them to your board. These optional parts are just that, *optional.* They are by no means required to complete the build or to do the homework for the course. If you find the board useful/interesting enough that you'd like to beef it up, you can do this any later time. More detail on these optional parts may be found immediately below.

1. The piezo speaker is the same type that are used on the Experimenter boards from ME30, older versions of which you'll be using for Lab2 (Intro to Arduino) of this course. It's a tinny-sounding, lo-fi speaker that's common in retro, handheld games for decades in many appliances. It's the round part that's in the upper left quadrant of the board, and has an accompanying 1.5Kohm voltage-limiting resistor immediately below it. The speaker is connected to Arduino Pin 12.

2. v2.3 boards have a spot for an optional 5-pin female header to be installed into which an inexpensive

ADXL335 analog accelerometer may be attached. If you're interested in getting one of these (sub-$5) modules, contact me. Soldering-in the header is (only!) a matter of keeping it flush to the board as it's flipped upside-down. This can be done by using either reverse-clamping tweezers or by applying steady, perpendicular force to the header to keep it flush to the board.

Build Procedures (The Big Game)

Please note that all photographs below are of the v2.1 board, not the v2.2/v2.3 board which most of you will have this semester. The build procedures and board layout are basically unchanged from v2.1 to v2.3 (with the exception of the final bug fix section for the v2.3 boards) however. An update of this document may have these photographs updated to show the v2.3 board.

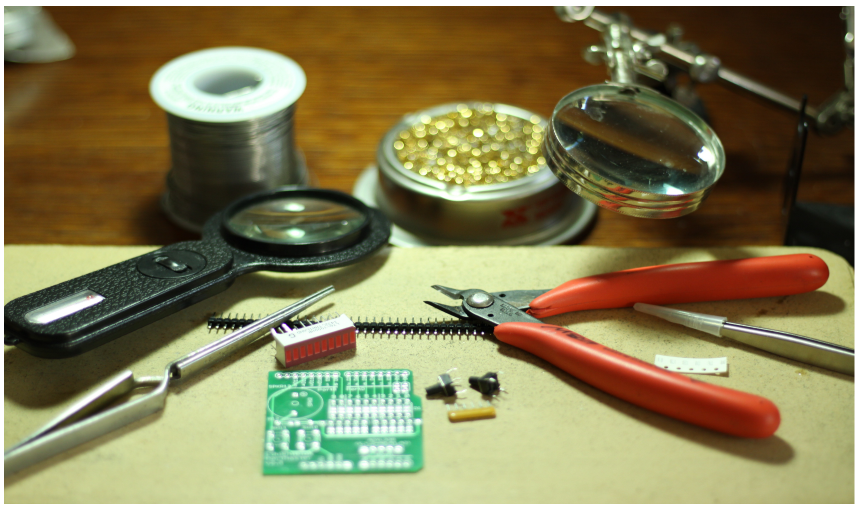
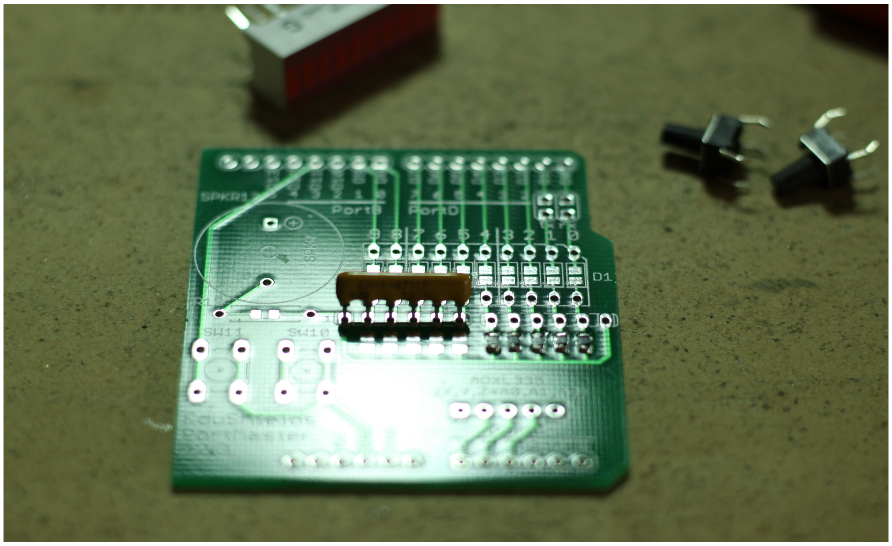


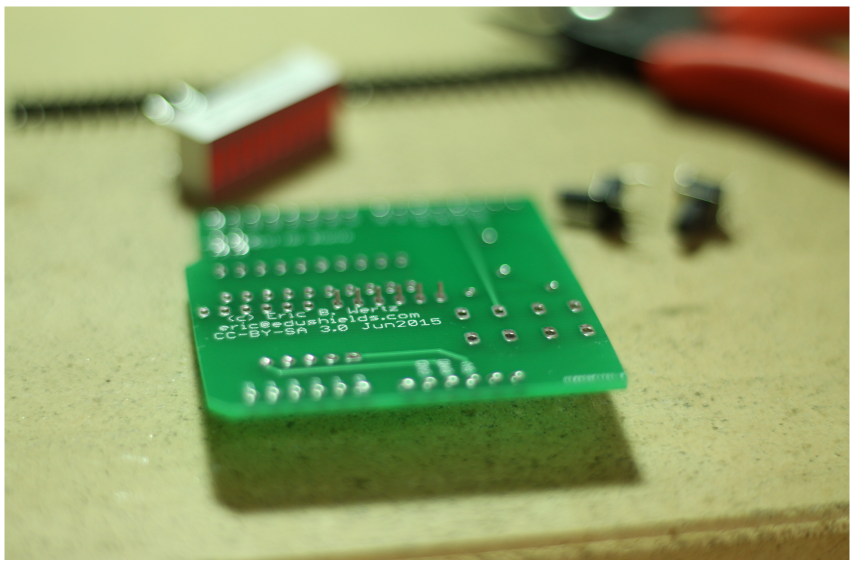
Illustration 1: Parts and tools ready for hot soldering action

Make sure that you have all of the components and tools you'll need before you begin work. All of the components are listed at the beginning of this document and shown above.

If you have one or more gloves, now is a good time to put them on. Prepare the board by wiping it down with a lint-free cloth and some isopropyl alcohol or acetone, and dry the board thoroughly. Until you have assembled and successfully tested the board, from here on you should avoid touching the solder pad with oily/dirty fingers, and having gloves here is helpful. Also, if/when taping components down to hold them in place, a board re-wipe may be necessary to remove any stray adhesive.

Following the general rule of inside-out and lowest-to-highest-profile, the first components to install are the SIP resistor packs. Unlike single, discrete resistors, (bussed) SIP resistor packs have an orientation that must be heeded. The common connection is marked on the shrink-coating with a mark, in this case, a black dot on the marked side above the left-most pin. The dots on both resistor SIPs must be oriented so that they're closest to their respective board edge. The photo above also demonstrates which way each component must face (well, not really, because I don't have both SIPs... :-( ).



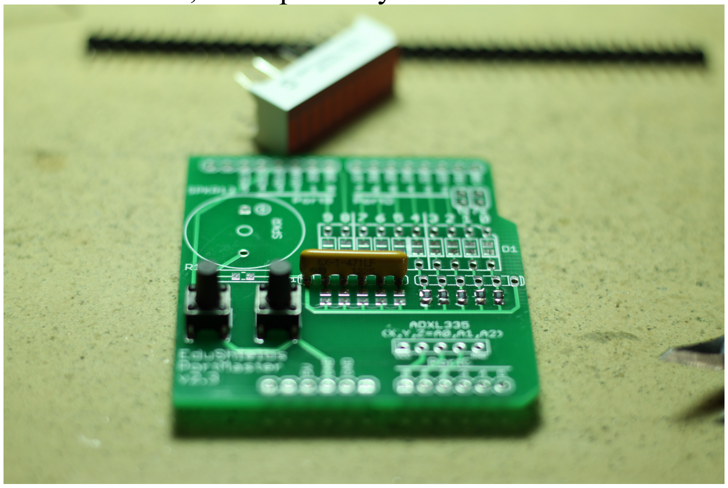


When inserting a leaded[2] component like this, it very often wants to either slide or fully fall out of the board when you flip it over to solder it. There are three ways to solve this problem – tape the component flush to the board, press something against the top side to keep it flush, or anchor the part by bending the leads to keep it in place. There are situations where some of these are better solutions than others, but we're going to do the latter. While pressing the component flush against the top face of the board, flip it over and bend the opposite pins on the package flush against the board, ideally in different directions (to ensure that it still doesn't unhook and slip out). To bend the leads you may use either any stiff tool (being careful not to slip and scratch the board) or even your fingernails. Flip the board over topside-up and check to see that the component isn't wonky with respect to the surface of the board.

In many situations it's best to insert just a single part or just a few and solder them as you go, rather

2 “lead”, as in component with a lead (pronounced 'leed”, *not* “led”)

than inserting many and soldering them all in one flip-over. This is especially true with components with long leads on them because many of them will interfere with you, or each other, as you're soldering. In this case, none of our components have long leads, so it's feasible to insert all of the topside components and solder them in one pass, and that's what we'll have you do. However, if you'd like to take it one component at a time, that's perfectly fine.

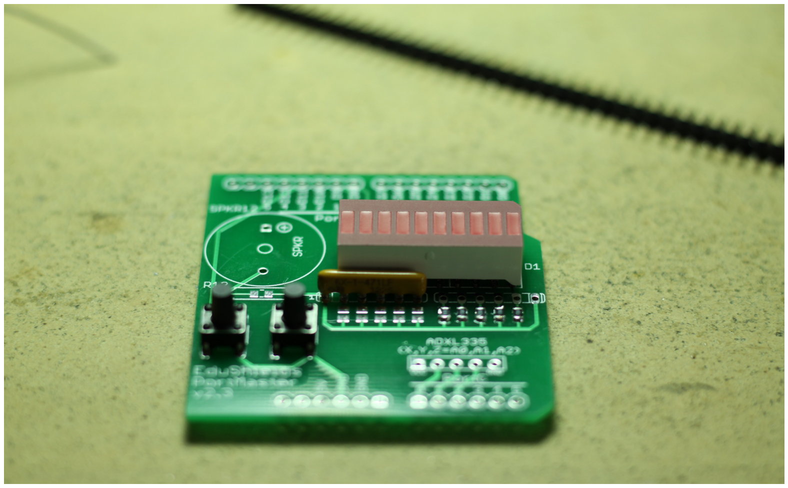


Insert the two buttons into the top side of the board. This is done by aligning the leads so that all four are pointing down into the holes in the board and then applying just enough pressure to pop them flush into place. If they do not pop into place with gentle pressure, it's almost certainly because one or more of the leads are not headed straight down into the hole, but have instead folded-under the button body. Not guiding each lead into the hole can badly bend the errant ones. If you're especially inelegant about it, you can break one of the leads when trying to straighten it back out.

Note first that although the body of the buttons is square, with the leads they are not. There is a short side of the button and a long side, so orientation does matter. I typically partially insert two of the leads on one of the short sides of the button into their holes first, then use a tool or fingernail to guide the remaining two leads on the opposite side into/over their holes. Only then do I try to snap it flush into the board.

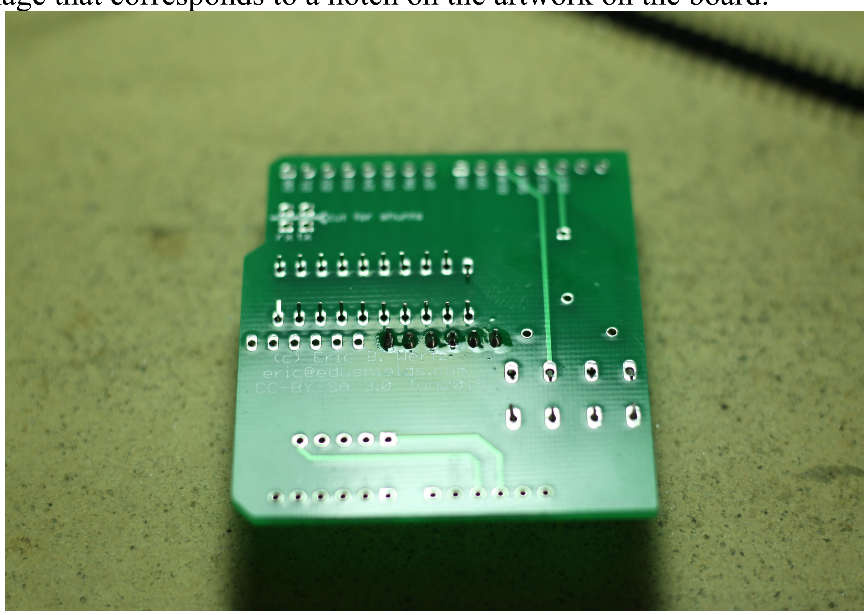
Warning: don't press your fingers against the back of the board as you're inserting buttons because when the leads pop through the back when you press them in, they're likely to lance your fingers if you're “lucky” enough to have them in the right place.

Before soldering in each of the buttons, ensure that all four leads made it through OK before starting, and that the body of the button is flush to the board.

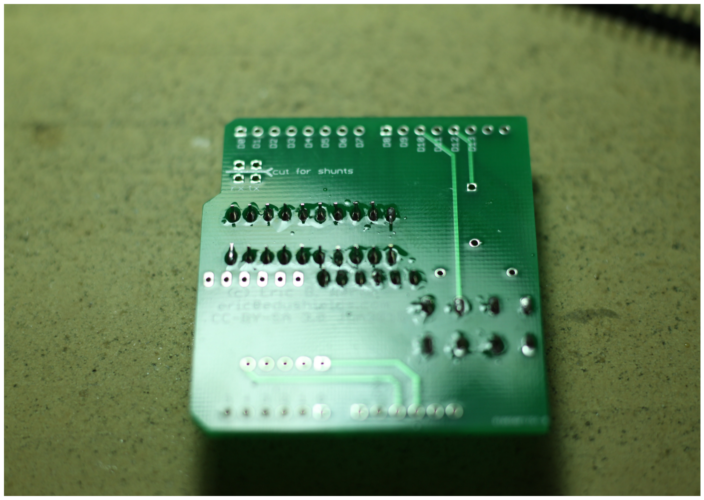


Finally we'll insert the LED display, using the same "trick" as with the resistor SIPs – namely, bending the opposite pins on the package to hold the component flush to the board. Note that in this case it's the opposite corner pins that one typically bends over.

Please note that the LED display (like all LEDs) also has a required orientation – there is a notch in one corner of the package that corresponds to a notch on the artwork on the board.



Once you've inserted and anchored the LED display in place, you should have something that looks like the picture above.



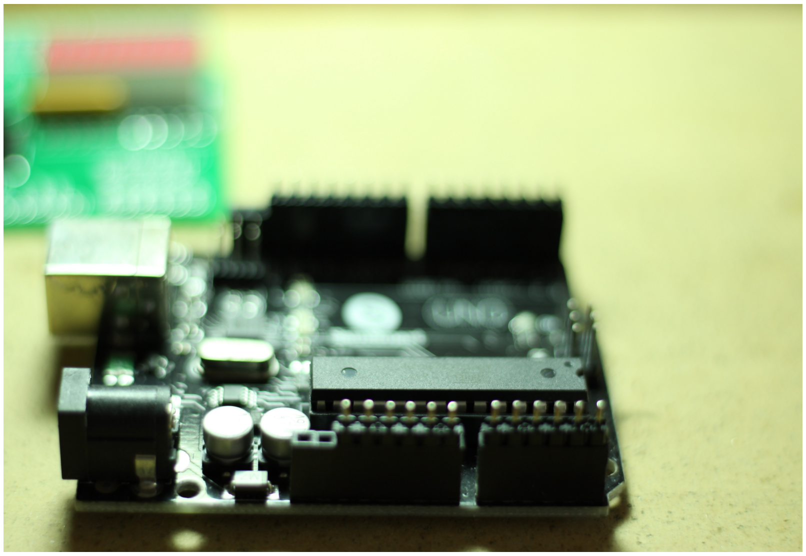
At this point we're all done inserting components into the topside of the board, and we're ready to solder. Using all of the great knowledge you gained from the instructional videos and web pages above (at least some of which you *did* review, right?), solder all of the connections. When you're done you should have something that looks like the picture above. Take a good look at all of the solder joints and make sure that they all look OK and that you haven't missed any of them. It's not a bad idea to have someone else look at your work before you continue.

The last parts to attach are the headers that connect down into the Arduino. .The only locations that need headers soldered into them are those that are adjacent and parallel to the immediate edge of the board. Not all of these locations are strictly necessary (electrically) but are included both because they make the board more mechanically stable during use, as well as making it easier to figure out how to align the board with the Arduino when it comes time to join them together. I'm recommending that all headers be soldered in, which means that you'll need header pin segments of length 8, 8, 6 and 6.

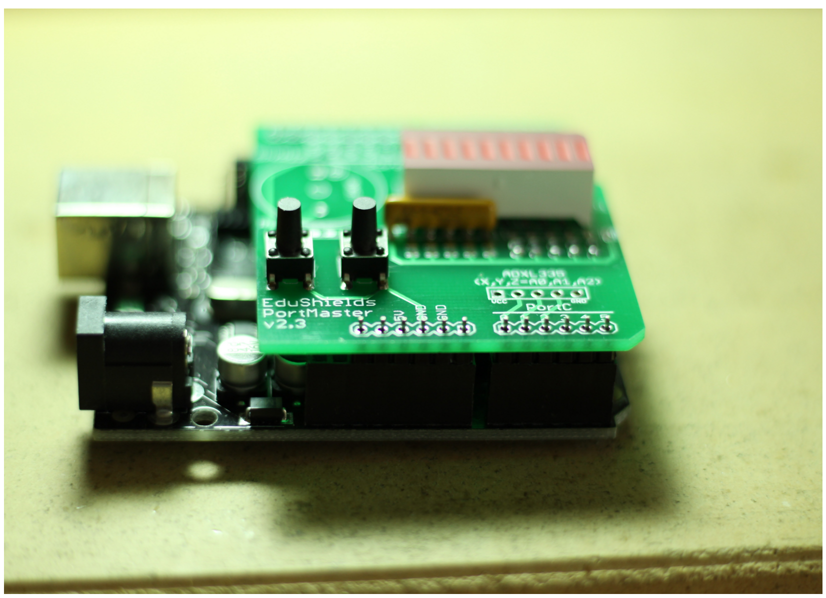
If you have an unbroken header strip, you'll have to break it into the correct segment lenghts. It's best to use a pair of snips to cut the header at the desired break-points, but with some care and both of your thumbnails you can simply snap it into two pieces.

We are going to use an Arduino to hold everything in place, essentially using it as a soldering jig. The only issue with using a working Arduino is that you do have to be somewhat careful about not overheating the headers because they're a direct thermal connection to the chip. In this case, it's not so bad because the chip is fairly far away from the solder joint you'll be making, so it's not as much of a concern in this particular case.

The primary downside of using a working Arduino is if you're soldering many boards (say 50-100), then all of the insertions and removals is a lot of wear and tear on the Arduino's connectors. In this case, making a jig out of a custom Arduino prototype board rather than wearing out a good Arduino is a smart idea.



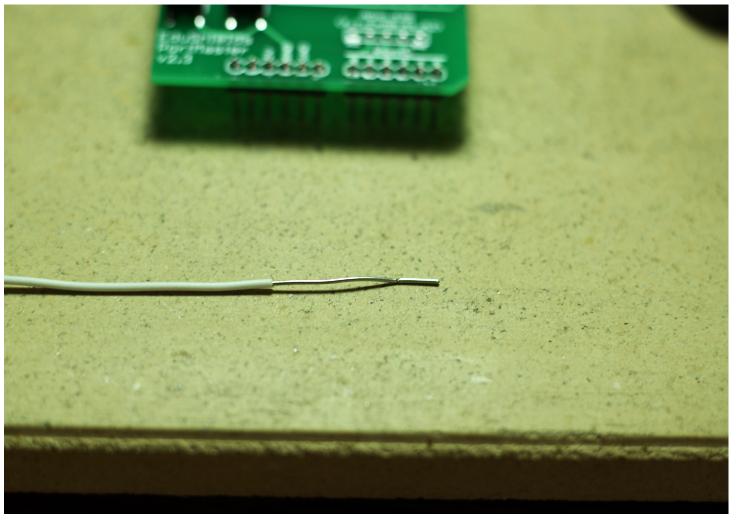
Grab an Arduino and insert all of the headers into the Arduino as shown above. Note that the long sides of the pins go down into the Arduino and the short sides are facing up.



Next, set the PortMaster PCB on top, making sure that all three sets of headers are poking through. Solder all of the header pins down after ensuring that the board is level, by noting that all of the pin heads are all extending equally above the board surface.

Applying the board bug-fix for the missing GND trace *on the v2.3 boards only.*

On the v2.3 board there's a missing trace that's supposed to join the low-side of the resistors to GND. Although there are numerous ways to fix this, the easiest two are shown below. The difference between the two methods is whether or not you're planning on later installing the optional speaker component on the board (which only one student did last semester). *Solution1* is easier than *Solution2* because it takes advantage of an existing hole in the board which would otherwise be used for the speaker. **The overwhelming majority of you (those choosing not to purchase and install the optional speaker) will be choosing *Solution1.***



For both *Solution1* and *Solution2*, only a short length of wire needs to be end up being added -- something on the order of 0.25-0.375". Because this wire so short and because there's no safety issues with this wire being bare, an un-insulated segment of wire is suggested. To make handling easier, strip at least 0.75" of insulation off of the end of a piece of solid core wire (which you can get from the scrap box in the lab). Because it's challenging to handle only a 3/8" segment of wire, you'll be soldering-on a substantially longer segment of wire, and later trimming off the excess.

The approach that is pictured below is that you will first solder the free end of this wire to a spot on the board, and then use the free end to guide the wire into place for the second connection. The excess wire ("handle") will then be snipped-off with wire snips, preferably as close to the solder joint as possible to keep the sharp, pointy end of the fix wire from poking you while using the board. To eliminate this "hazard" you may want to apply the fix wire to the underside of the board rather than the top side if you're using *Solution1*.

I'll note that having a someone else hold the patch wire in place while the second solder joint is made will make this procedure a lot easier. This leaves both of your hands free for the solder and the soldering iron, respectively. If you have some type of clip or tweezers available that will hold the wire in place while soldering, that can also help.

"Solution1" fix wire location

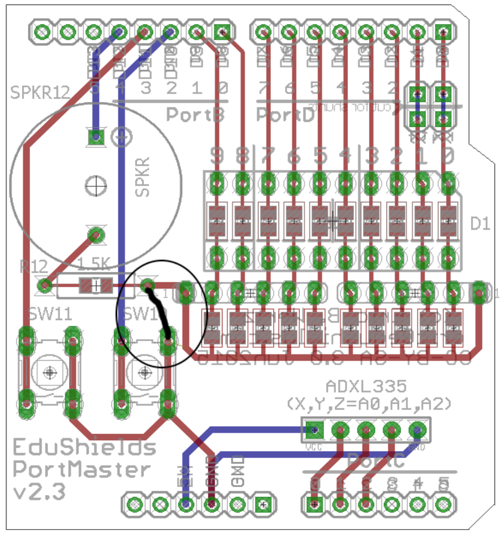


Illustration 2: Solution1 fix wire location, circled

The black hand-drawn line circled above shows the location of the patch wire. This fix uses an existing hole on the PCB, which makes applying the fix significantly easier than that for *Solution2*. Stick the end of the bare, stripped wire into this hole and solder it, leaving you enough bare wire so that you can run it over to the upper-right terminal of the SW10 button, as shown below.

After making the first, upper solder joint, guide the remaining segment of stripped wire over to the designated terminal on the button and join them together with another solder joint. Clip the excess wire off of both ends and you're done.

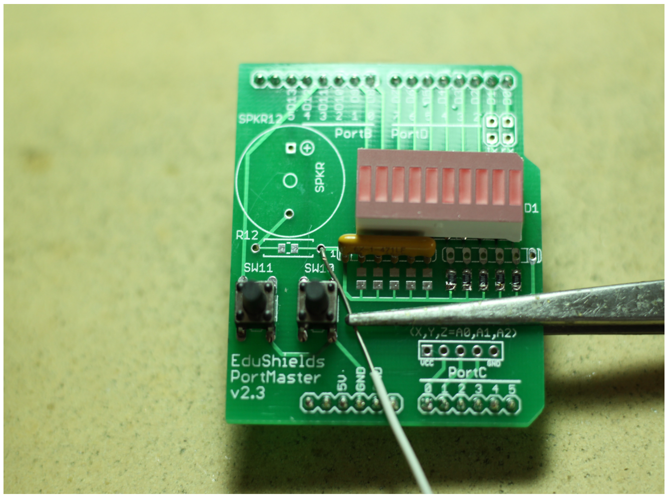


Illustration 4: "Solution1" topside fix method

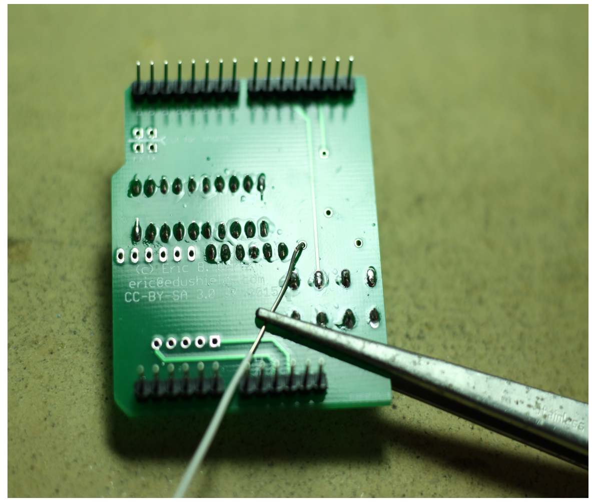


Illustration 3: "Solution1" bottomside fix method

As mentioned above, you may want to apply this fix-wire on the underside of the board rather than the topside. This will ensure that the sharp ends where the wires were snipped won't poke soft, juicy fingers using the button. Pictures of both the topside and bottomside methods are demonstrated above.

Solution2 fix wire location

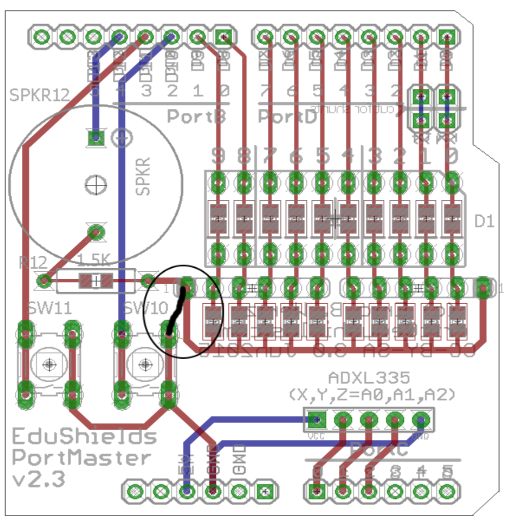


Illustration 5: Solution2 fix wire location, circled

The difference between this apprach and *Solution1*'s is that you'll be using one of the terminals from the resistor SIP for one end of the connection, leaving the hole for the speaker's resistor free for later use. I only recommend applying this fix to the back side of the board because of the presence of the resistor SIP's insulation on the top side (which will likely be damaged by the heat from the soldering iron tip if approached from the topside).

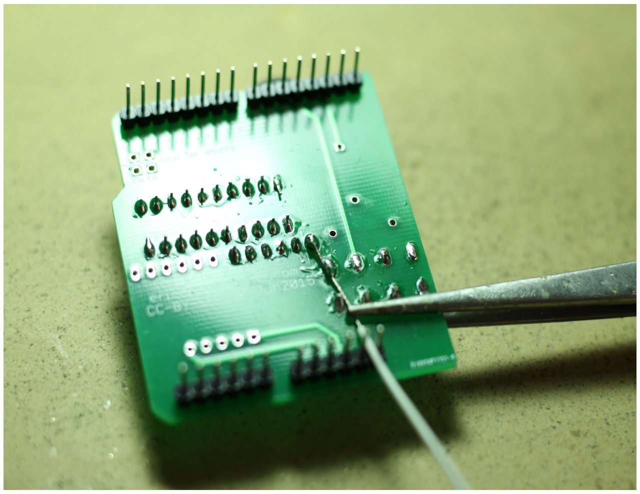


Illustration 7: "Solution2" fix wire in place

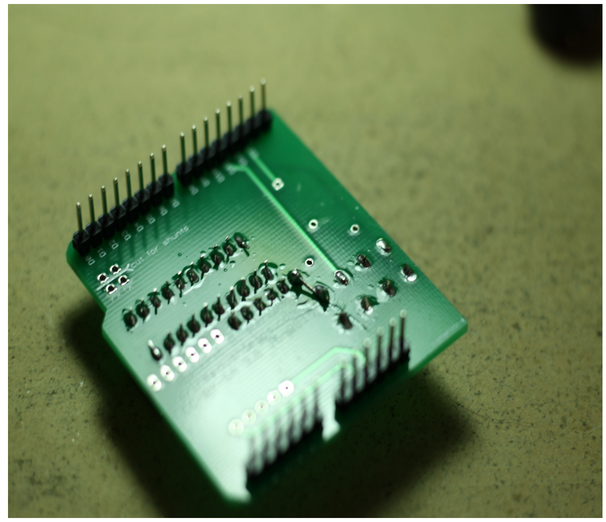


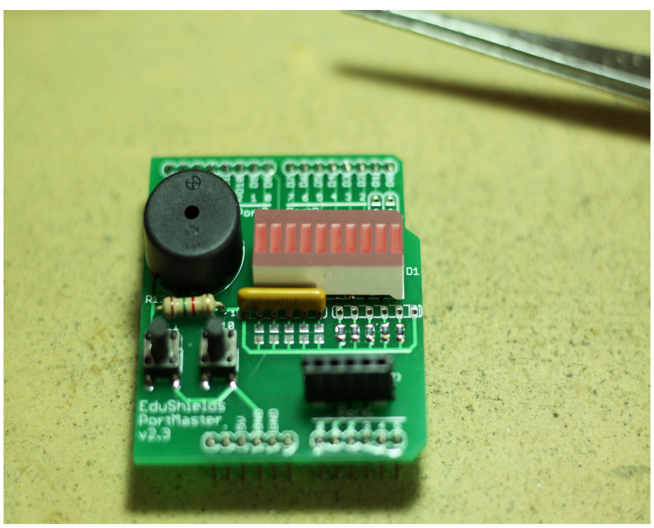
Illustration 6: "Solution2" fix wire soldered and clipped

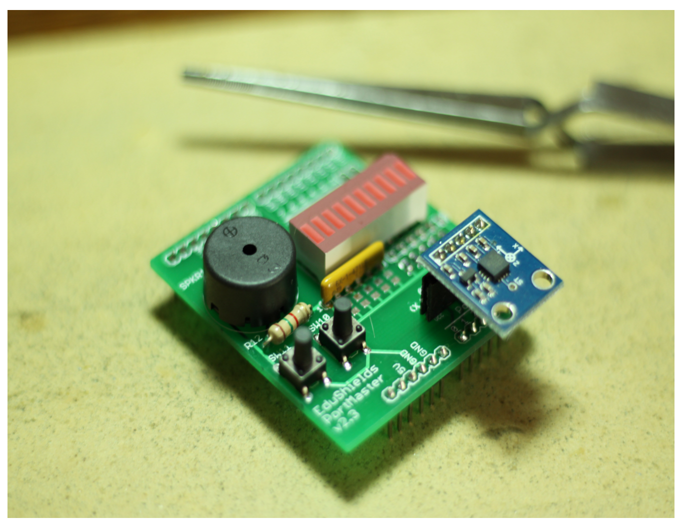
Start by soldering the end of the stripped wire to this joint on the bottom side of the board. Using the long free end of the fix wire, guide the stripped wire segment over to the correct terminal of the pushbutton. Snip any excess wire off of the fix wire and you're done.

Well, you're done if your board tests out OK, that is...

Boards with optional components

These are examples of the board with the optional speaker and accelerometer header+module on them.





Board test with software

To test your board, you can use the simple program *PortMasterTest_AS* (Arduino-style) or *PortMasterTest_PS* (PORT-style) for testing. Pressing either of the two user buttons lights some/all of the LEDs to demonstrate that all of the buttons and LEDs work.

Note: You may notice that after you're done assembling your PortMaster and you've plugged it in the first time, one, two or more of the LEDs are full/half-on while the others are all off. This is *not* necessarily a sign that you've assembled anything incorrectly. It's often just an indication that whatever program was last put on the Arduino isn't initializing any/all of the LED pins in software. Once you load the test program (or write your own), you're likely to see all of the LEDs start off in the same state.

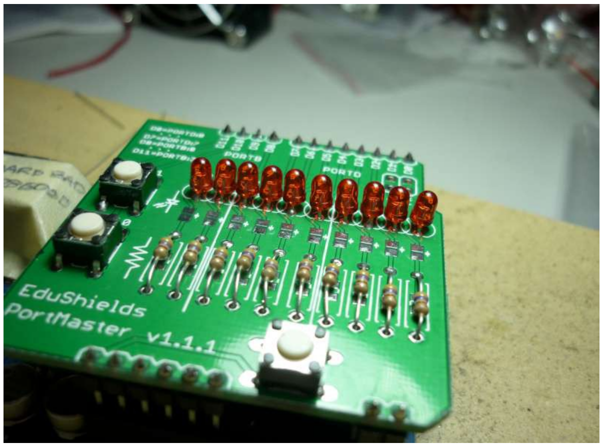
If the board does not test out properly, gently remove it and take a look again in the areas where it didn't function properly. The problem could be a bad (or unpowered) Arduino, bad software, a bad component, a bad board, or bad (or missing) soldering. By far the most common mistake is to simply miss a connection that should have been soldered. Between common sense and a few simple tools you should be able to diagnose your board's problem(s). Note that you can test each pin individually by connecting ground from the power supply (or Arduino) to the downward-facing GND header pin, and 5V on each of the LED header pins (D0-D9) to test the LEDs. The ohmmeter can be connected between the button pins (D10-D11) and the GND pin to check for continuity when the button is pressed.

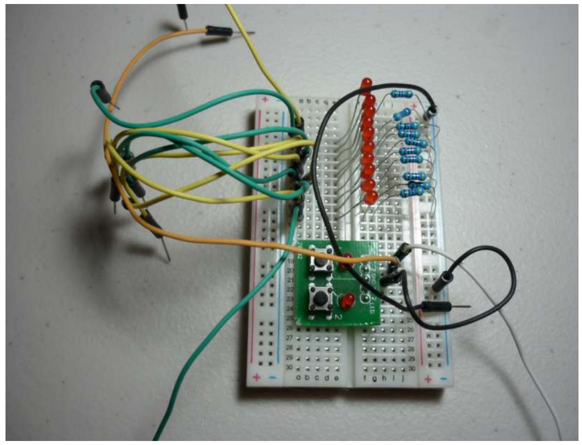
Should it become clear that one of your components is bad, or that you installed it incorrectly, your TA may be able to help you replace the part if you're not up to speed on solder rework skills. If your particular TA also isn't, get in touch with me and we'll see what we can do.

Important Note: Since you've been handling leaded solder, wash your hands.. now!

Post-game Wrap-up

For reference/entertainment purposes the following is one of the previous (v1.x) versions that students built which used only discrete LEDs and resistors, as well as a photo of the solderless breadboard prototype.





Test Code

```
/*
 * Test program for the PortMaster board.
 * Press each button to turn on half of the LEDs.
 *
 * 9876543210 <- LED pin number
 * BBBBBTTTTT <- control button (Bottom/Top)
 *
 * Eric B. Wertz v1.00 2012/08/27
 *               v1.1  2013/10/23 renamed some identifiers, removed pre-Arduino1.0 code
 *               v1.2  2014/09/01 improved function interfaces,
 *                                switched old names from TOP/BOTTOM to LEFT/RIGHT
 */
#define PIN_LED_MIN        0
#define PIN_LED_RIGHT_MIN    0
#define PIN_LED_RIGHT_MAX    4
#define PIN_LED_LEFT_MIN     5
#define PIN_LED_LEFT_MAX     9
#define PIN_LED_MAX        9

#define PIN_BUTTON_RIGHT  10
#define PIN_BUTTON_LEFT   11

void setLEDs(int ledPinMin, int ledPinMax, int highOrLow)
{
  for (int pin=ledPinMin; pin <= ledPinMax; pin++) {
    digitalWrite(pin, highOrLow);
  }
}

void setup()
{
  for (int pin=PIN_LED_MIN; pin <= PIN_LED_MAX; pin++) {
    pinMode     (pin, OUTPUT);
    digitalWrite(pin, LOW);
  }

  pinMode(PIN_BUTTON_LEFT,  INPUT_PULLUP);
  pinMode(PIN_BUTTON_RIGHT, INPUT_PULLUP);
}

void loop()
{
  if (digitalRead(PIN_BUTTON_LEFT) == LOW) {
    setLEDs(PIN_LED_LEFT_MIN, PIN_LED_LEFT_MAX, HIGH);
  }
  else {
```

```
    setLEDs(PIN_LED_LEFT_MIN, PIN_LED_LEFT_MAX, LOW);
  }

  if (digitalRead(PIN_BUTTON_RIGHT) == LOW) {
    setLEDs(PIN_LED_RIGHT_MIN, PIN_LED_RIGHT_MAX, HIGH);
  }
  else {
    setLEDs(PIN_LED_RIGHT_MIN, PIN_LED_RIGHT_MAX, LOW);
  }
}
```

Revision History:

v1.2 Initial version

v1.2.2 Added/replaced two pictures, replaced all pictures with shrunken versions, added orientation of LED display and breaking headers note. Other misc cleanups.

v2.1 Updated for v2.1 hardware version of the PCB, including explanation of the optional portion of the build.

v2.2 Updated text for v2.2 hardware version.

v2.3-pre1 Added section for new-to-v2.3 board bug along with general updates for v2.2 to v2.3 differences.

v2.3.0.1 Changes having to do with the possibility of building either a v2.3 versus a v2.1/v2.2 board. Highlighted some text that a few less-attentive students missed having to do with choosing between Solution1 and Solution2.