

shanes.net

clever Tagline goes here.

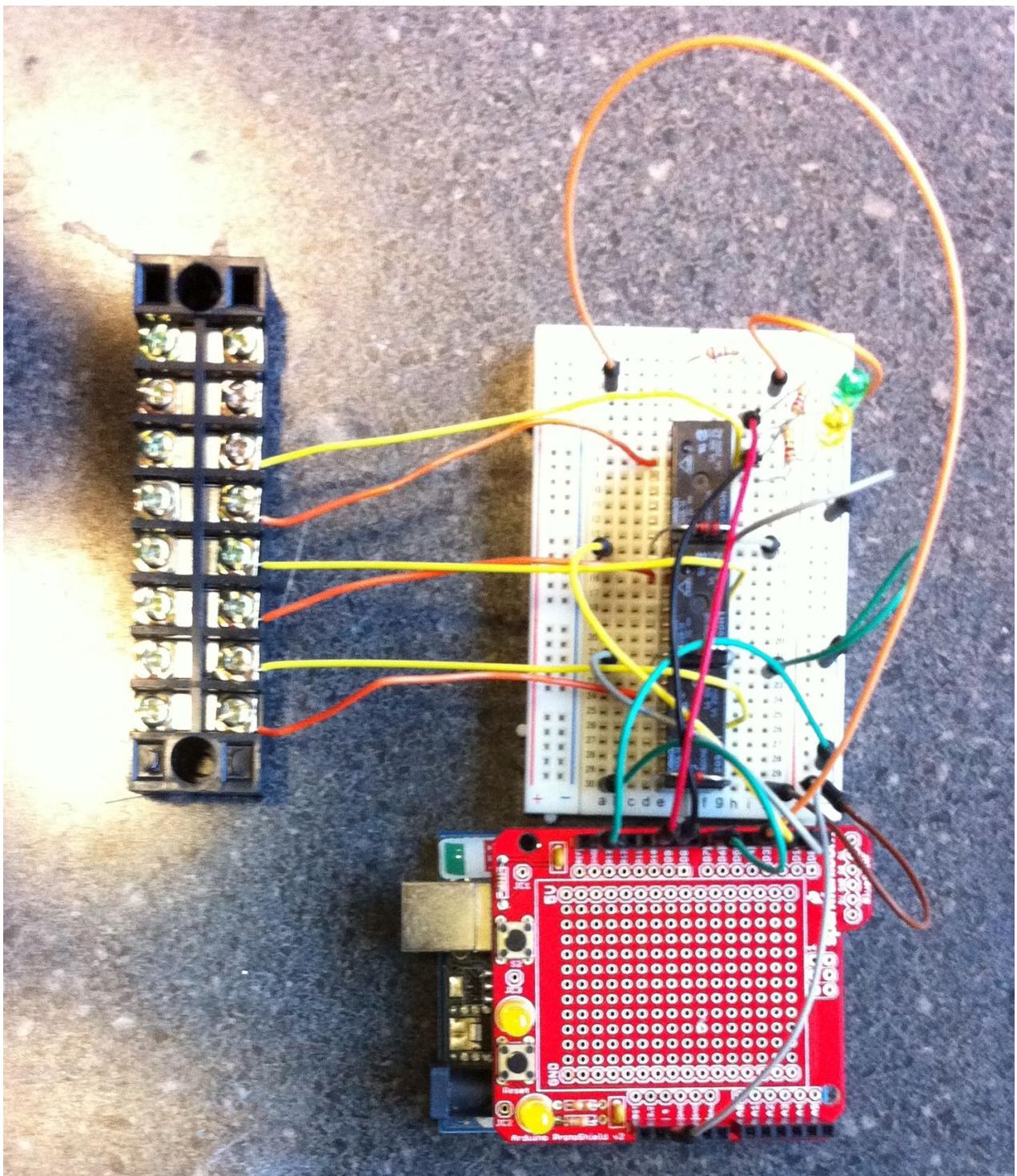
Arduino: from prototype to product

When I first got into working with Arduino, I had a hard time finding examples of projects that made it all the way to the ‘finished product’ stage. Hopefully this information will benefit someone looking to see what can be involved with bringing a simple prototype full circle. The following information is for background and does not have anything to do with the prototyping process itself:

The problem: We have an ‘instant’ water heater that also runs a hydronic air handler and radiant slab in the garage. In the winter, the system gets overloaded, and shower temps fluctuate.

The project: Reset switch for domestic/hydronic water heater. The device senses flow from a flow switch and intelligently shuts off power to pumps connected to the air handler after a preset amount of time. The device also resets itself if it’s locked out for too long.

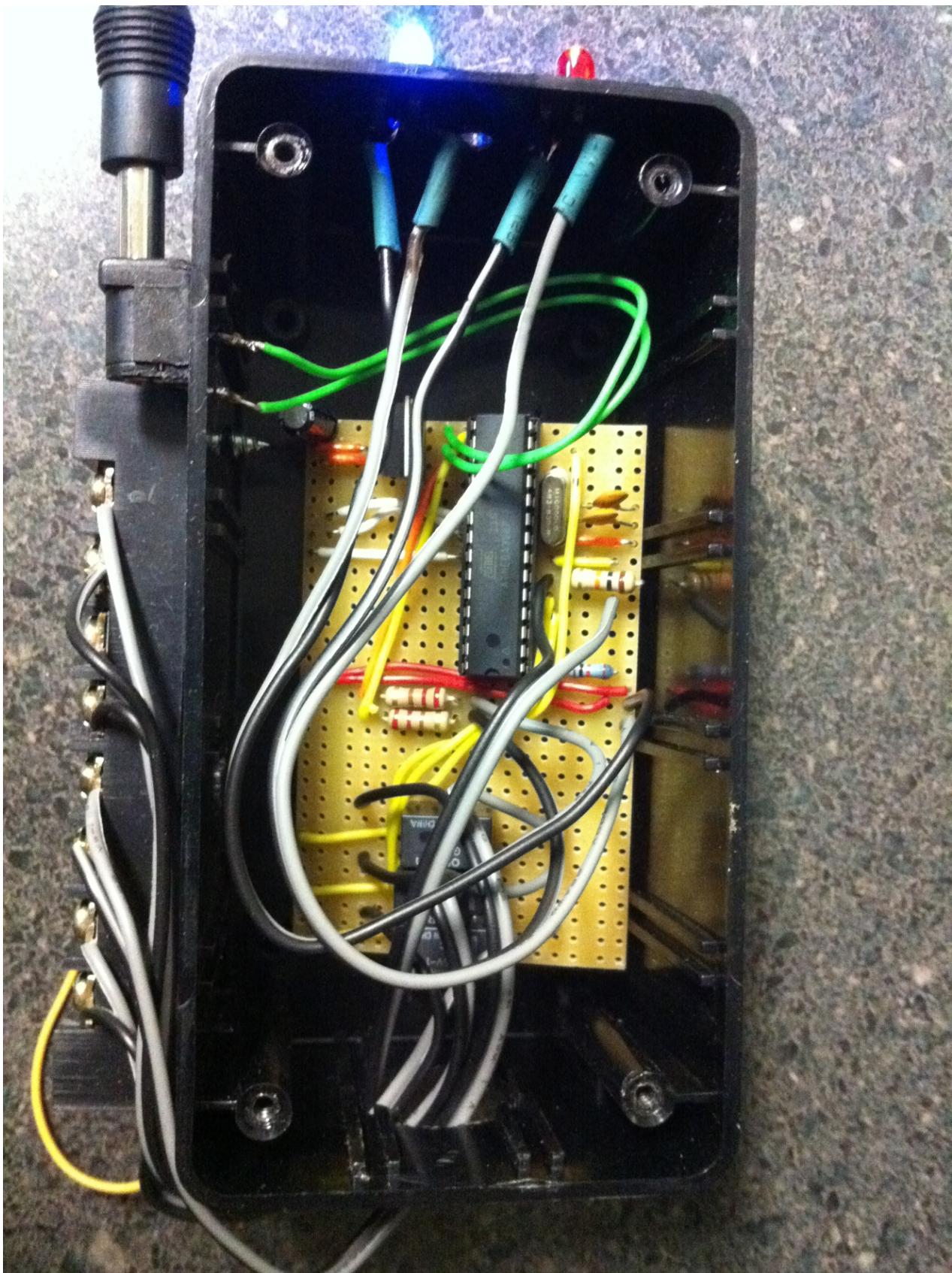
So, here’s what I started out with:



The first version of the device is big, bulky, and (relatively) expensive. It consists of an arduino, a protoshield (which is unnecessary in this case), a breadboard, a large terminal block, and a collection of components. As you can see, this is not portable, at least not easily. As well, it's got lots and lots of places where something could fall out, be pulled out, or otherwise experience a failure.

So, I decided to put it all in a box:

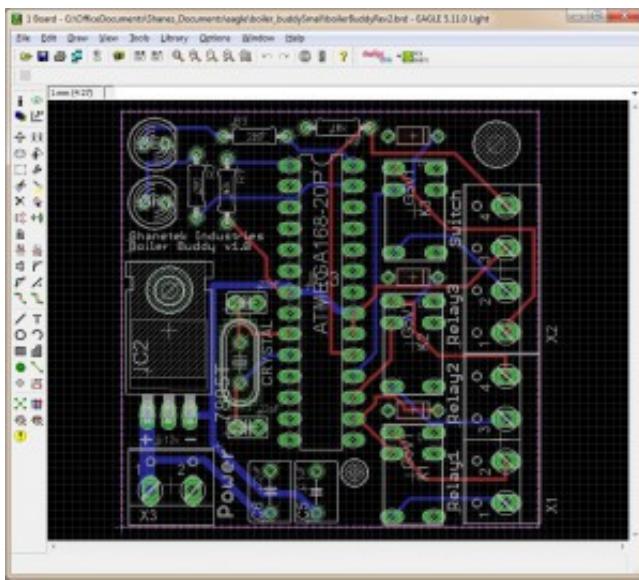




By moving the necessary components off of the arduino/breadboard and onto a prototyping board, I was able to condense the whole device into a more usable form factor. I used this [instructables](#) tutorial to get all of the necessary components off the Arduino. This intermediate step would have been fine if the goal was

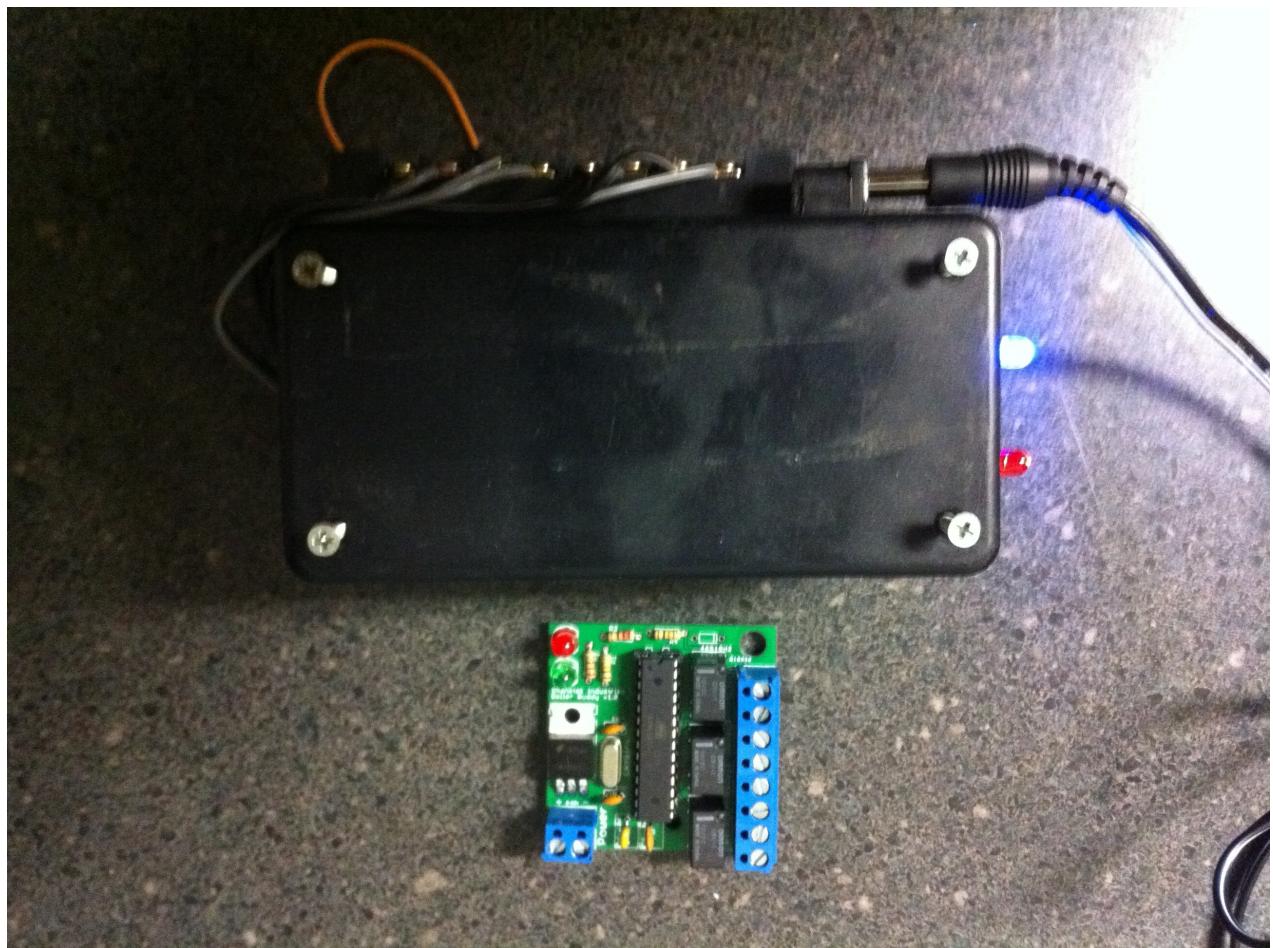
a one-off, but I was looking to design an easily repeatable ‘product-like’ final version.

In order to do this, it was necessary to design a printed circuit board that incorporated all of the components and circuit paths. After some research, I found that many hobbyists use an application called Eagle PCB Design to do this. Below, you can see what a PCB looks like in Eagle:

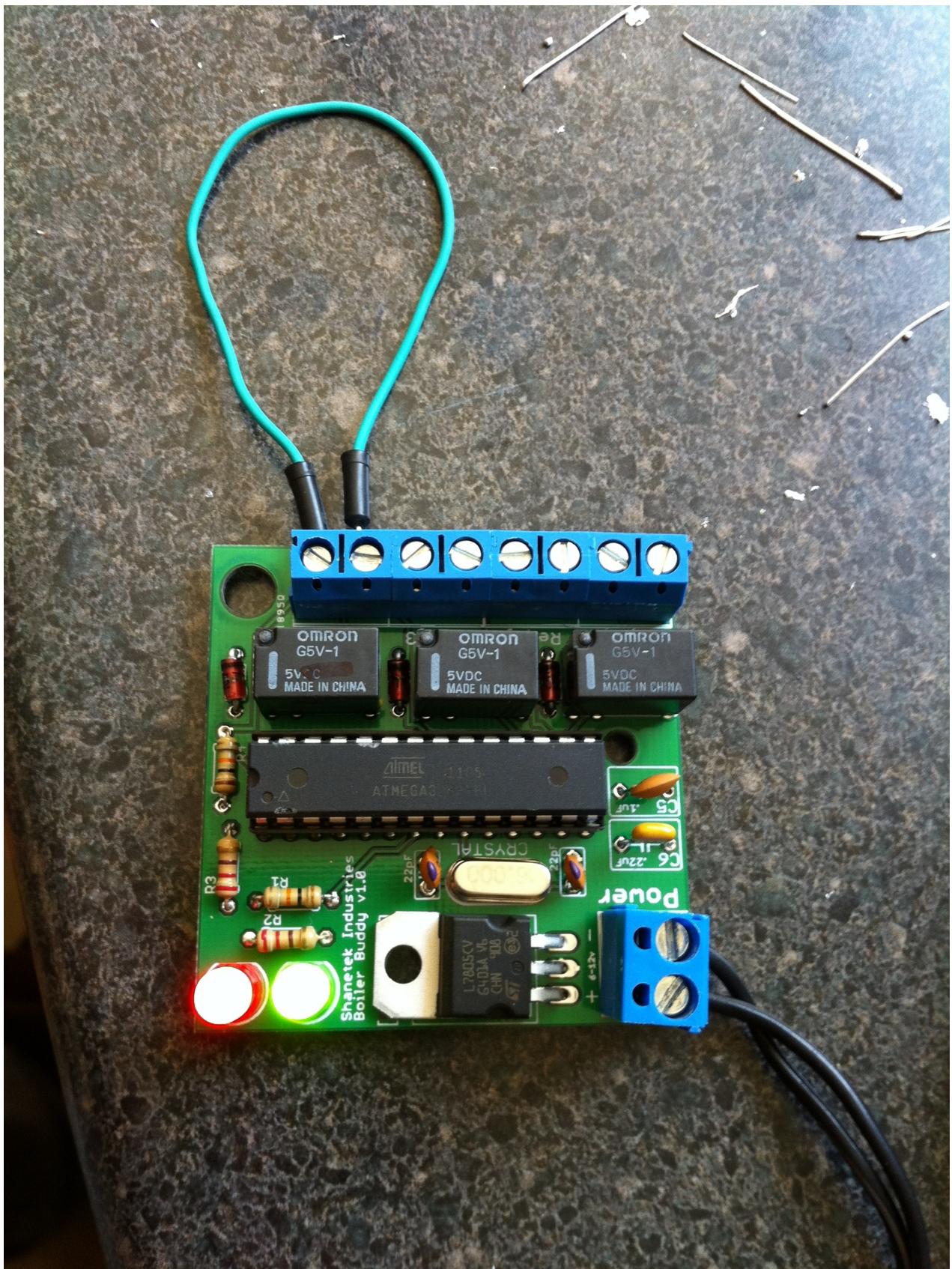


Eagle has a bit of a learning curve, but there are many great tutorials out there. Just do a youtube search and you’ll get quite a few examples. It took me a few nights of working with my design to get it the way I wanted. The next step is to find a PCB fabrication company to fabricate your boards. I chose to go with seeed-studio, there are many others. Unfortunately (but somewhat expectedly), some errors on my first design caused my first run of boards to be unusable (Tip: make sure to be extra careful about the orientation of all parts you place into your design, top/bottom orientation can be very confusing for a newbie). So, I made the necessary corrections and sent the board designs off again.

A few weeks later, my new boards arrived. Below, you can see the results:



Above: Intermediate prototype next to finished product. Quite a difference in size and complexity.



So as you can see, even someone with very little experience can go from prototype to (sort-of) final product. I have found the design of PCBs to be one of the most enjoyable parts of this hobby.

 March 16, 2012  Shane  Electronics/Arduino

7 thoughts on “Arduino: from prototype to product”

 **Eric**

December 4, 2013 at 10:37 am

Hey this is really great. I plan on using arduino to prototype a design of my own. Do you plan on doing a production run/selling your final product? Have you done any kind of cost analysis?

 **admin** 

December 4, 2013 at 10:44 am

I really just did this as an exercise in going from a design to a finished product. However, i do think this would work well for short production runs. For larger runs, I would look into smd components.

 **Ivo**

June 18, 2014 at 6:06 am

Nice post, thank you. One question though for a newbie: How do you manage to load your program on the new board which has no connection? It is send to the producer as well who will set up the chip with your program ?

 **Shane** 

June 18, 2014 at 7:53 am

Ivo, what I do is load the sketch on an Arduino and then just transfer the chip over to the protoboard. Now, I do have an ICSP header on that board, but I am using a few of the pins for the wireless. If they weren't being used, you could

transfer the sketch using a programmer.

**Ivo**

June 23, 2014 at 2:44 am

Thanks for the reply, clarifies how you do it for one product. Now the question would be if you wanna do “mass” production. You get in contact with a company that can produce for you your final product (at least the electronical part) and you wanna order let’s say 500 pcs. Can you ask them already to load your software on the chip or do you then have to do it one by one at home with your 500 products received, removing and reinstalling each chip ?

That’s really the part I am still wondering about. Now that you have your final product how do you manage “mass” production ? (an article about this would be brilliant :))

**Shane** ♀

June 23, 2014 at 10:45 am

Hey Ivo,

My understanding is that when you go from a prototype to quantities in the hundreds, you want to completely redesign your project down to individual components (and probably away from microcontrollers for simple systems). For example, the device shown in this post would definitely not require a microcontroller like the ATmega328, and instead would probably consist of several simpler/cheaper components. There are hobbyist companies that will manufacture and assemble your pcbs for you on a smaller scale. One that I am aware of is seeedstudio.com. The example I’ve provided here really only scales up to runs of about 25 or so before the assembly becomes overly tedious.

**brian**

January 18, 2015 at 5:44 pm

where do you get the arduino chip? how did you determine what was not needed on the original arduino board?

Proudly powered by WordPress