Introduction – I'm getting too old to solder surface mount devices by hand so I decided to convert a toaster oven into a reflow oven. There are tons of examples on the internet but I believe the best one is from [Whizoo](). I used many of the Whizoo methods for sealing and insulating the oven but I'm on a tight budget and can't afford their kit. Besides, it's a lot more fun to figure out my own design and learn along the way. My requirements are not as challenging as Whizoo's because I only solder one small board at a time and I use standard tin-lead solder. With the lower melting temperature of leaded solder, a 3rd heating element is not needed. An inexpensive Arduino Nano clone has just enough I/O's to control two solid state relays (SSRs), read the temperature, and talk to a 16x2 liquid crystal display (LCD). There is no power switch, touch screen, or input buttons of any kind. If a different solder profile is needed, new Arduino code must be downloaded. The software tells me when to manually open the door during cool-down so there is no servo motor. The oven is powered from a plug strip with a switch. 5 volts DC is provided via a USB power supply plugged into the strip. The minimal features of my oven bring the cost down to about $86.

This is the Hamilton Beach model 311-34 toaster oven I bought for $10 at my local Goodwill store.

Parts List:

Hamilton Beach model 311-34 toaster oven (used) from Goodwill $10.

Arduino Nano clone V3.0 from Aliexpress $3.

MAX6675 Module from Aliexpress $3. Don't use their thermocouple!

K Type Thermocouple from Aliexpress $2.

Fotek SSR-25 DA Solid State Relay quantity 2 from Aliexpress $8.

1602 LCD from Aliexpress $2.

J-B Weld 31314 High Temperature RTV Red - 3 oz from Amazon $7.

Design Engineering 050503-16 Floor and Tunnel Shield 10" x 10" from SummitRacing $19.

Ceramic Fiber Blanket 2300F, 8# Density, 1/2" x 12" x 12" quantity 2 from Amazon $12.98.

1/32" thick 5052 aluminum tray with flange, cut to 8"x9-5/8" from EBay $14.45

ABS Plastic Project Box, pack of 2 from Amazon $9. $4.50 each.

Total = $86

Parts from the junk drawer:

Prototype board with or without plated holes. I used Vectorbord from Digikey.

Sheet metal for mounting SSR's. I used a Simpson TP35 tie plate from Home Depot.

Male Header pins

Female header wires

Mounting Screws, bolts, nuts

Crimping lugs

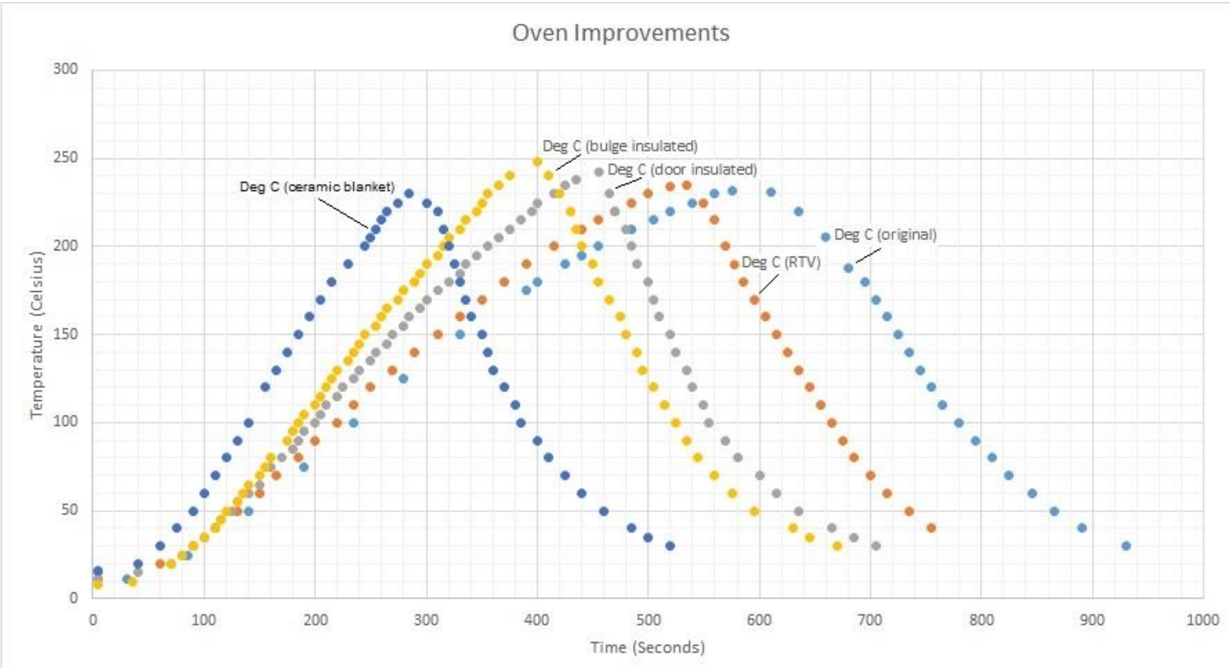
8.2KΩ and 1KΩ 1/8 watt resistors (or 10KΩ potentiometer)

AC Plug strip with power switch

USB Power supply

Mini B USB Cable

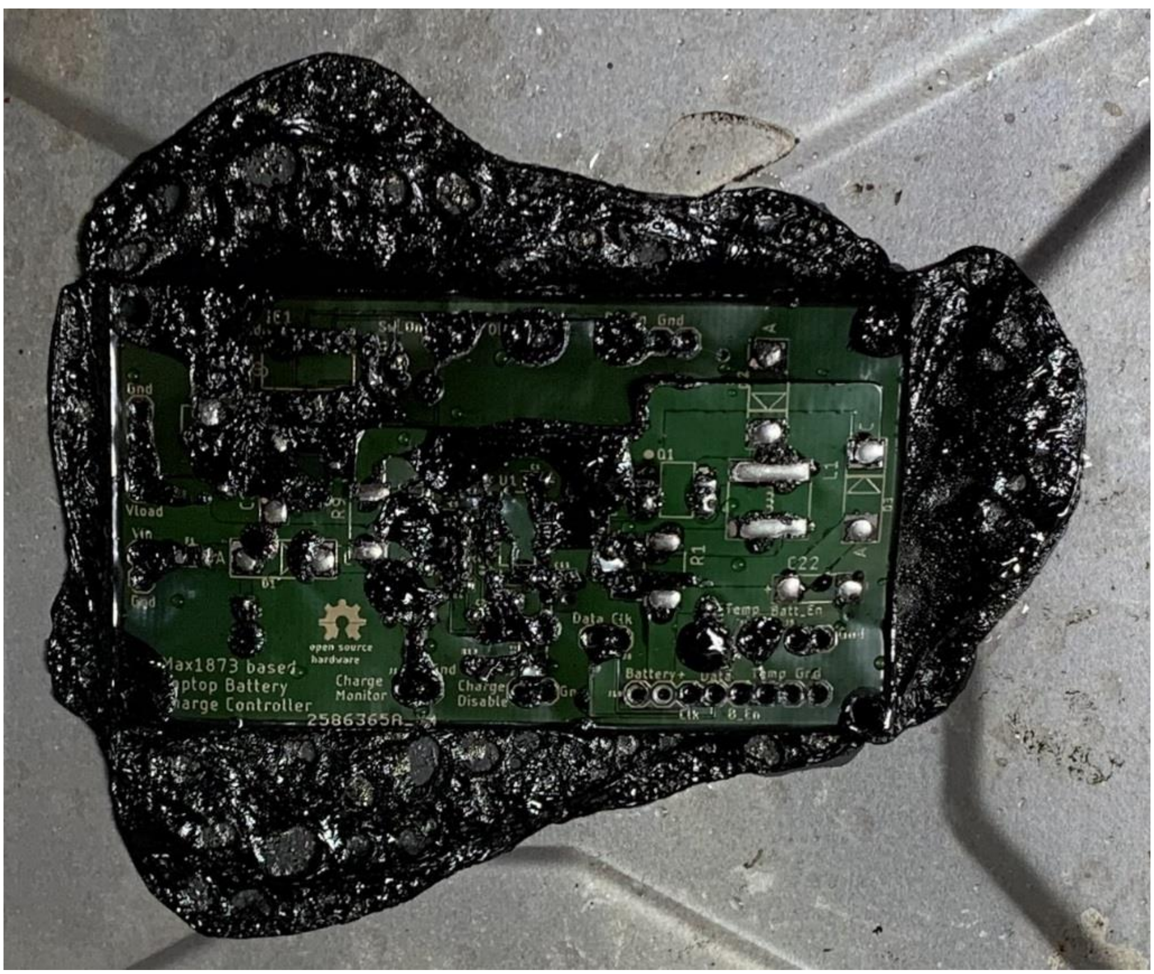
Oven Modifications – The Wizoo build guide details how to modify the oven to make it more efficient. The curves shown below were captured after each of the modifications were completed. The curve labeled “original” on the far right shows how slowly the temperature increased. Applying RTV sealant to the gaps and holes made the temperature rise quicker. Likewise, the door insulation, bulge insulation, and ceramic blanket also showed measureable improvements in temperature rise time.



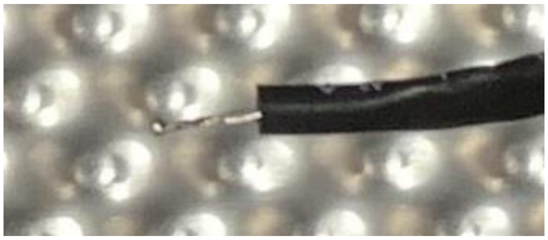
I chose not to spend any more money for a 3rd heating element, Reflect-A-Gold tape, door gaskets, or servo motor since my oven is good enough for tin-lead solder. Perhaps I’ll add some of these items in the future if I want to use lead-free solder.

I initialy used the threaded M6 thermocouple (shown below) that came with the Max6675 module but I found the temperature reading would lag behind the actual temperature due to the thermal mass of the bolt. This caused the actual temperture to be considerably higher than the measured value.

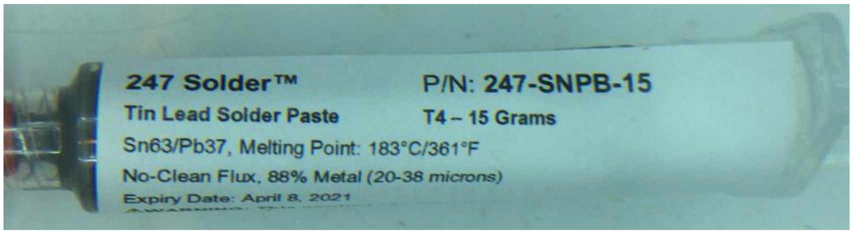
I melted a test board due to this thermocouple and because I used the upper heating element at full strength. Luckily I didn’t populate the board with components.



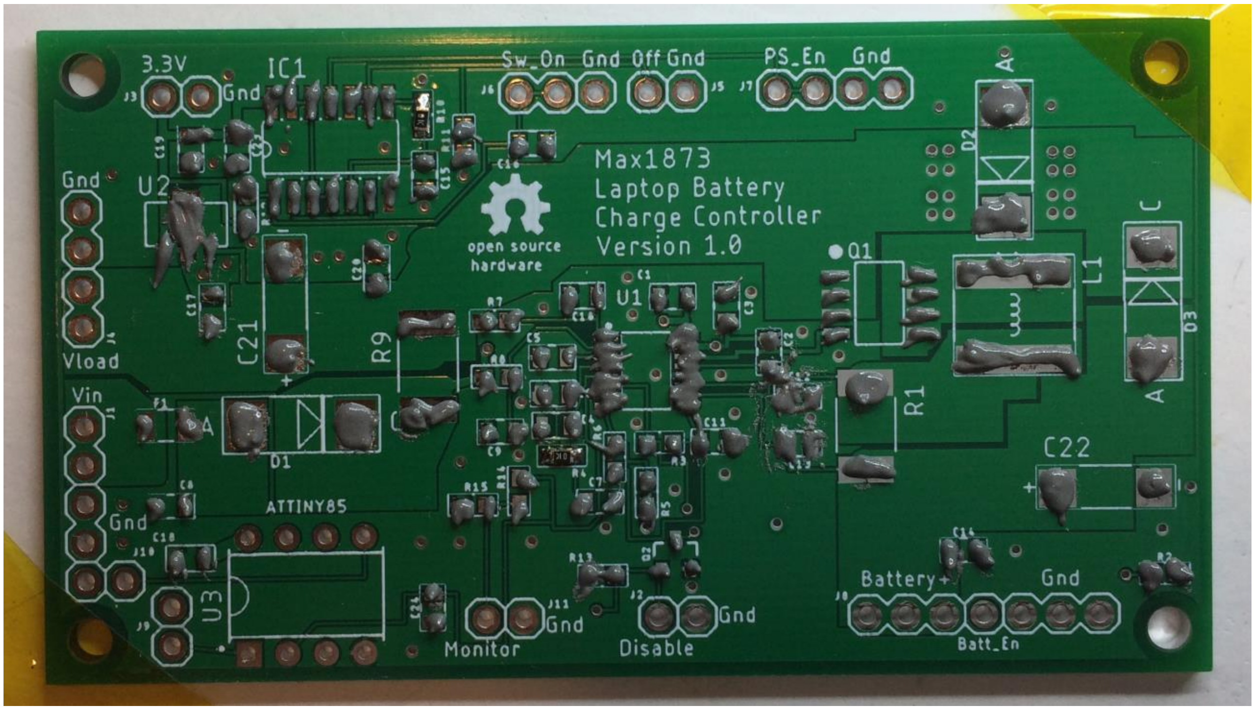
I switched to a low mass thermocouple as recommended by Wizoo and it worked much better. The tiny ball that welds the two wires together is able to quickly change temperature. The same test board flowed just fine so I've got the temperatures set correctly.



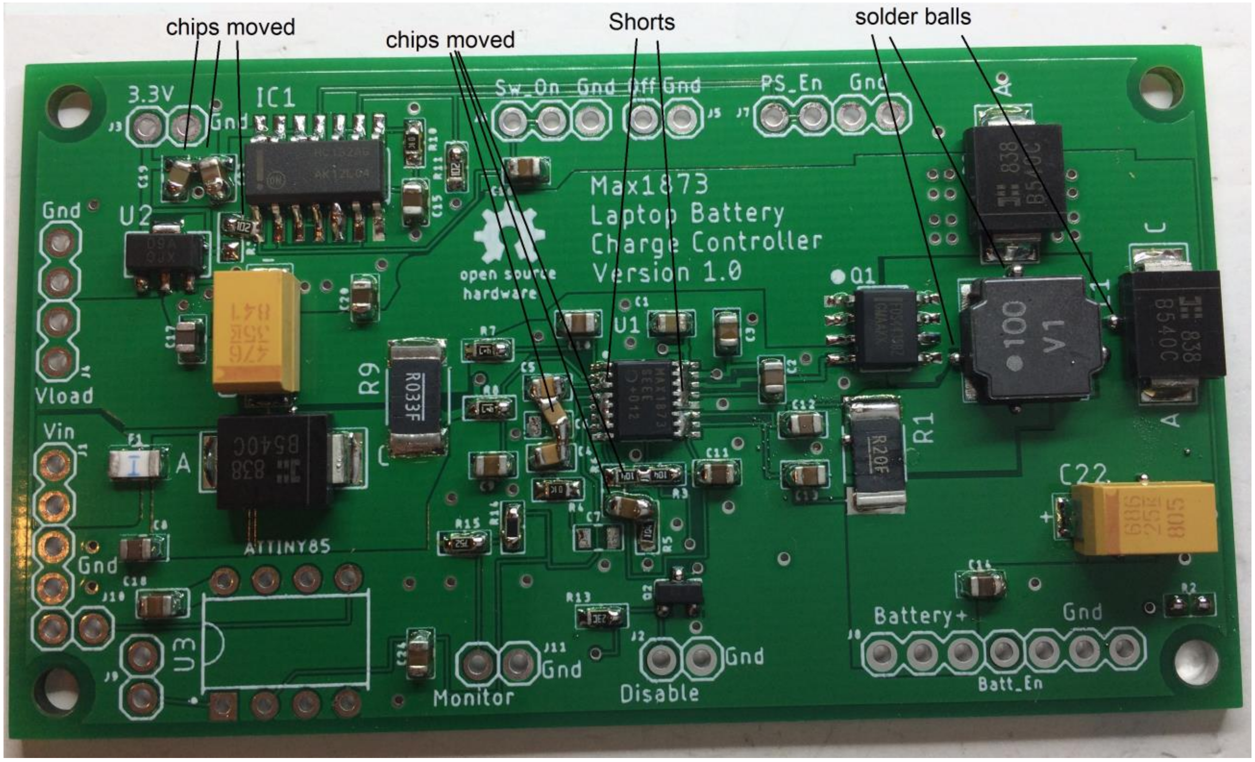
I don't have a stencil for this board so I applied the solder paste to each pad using a syringe tip.



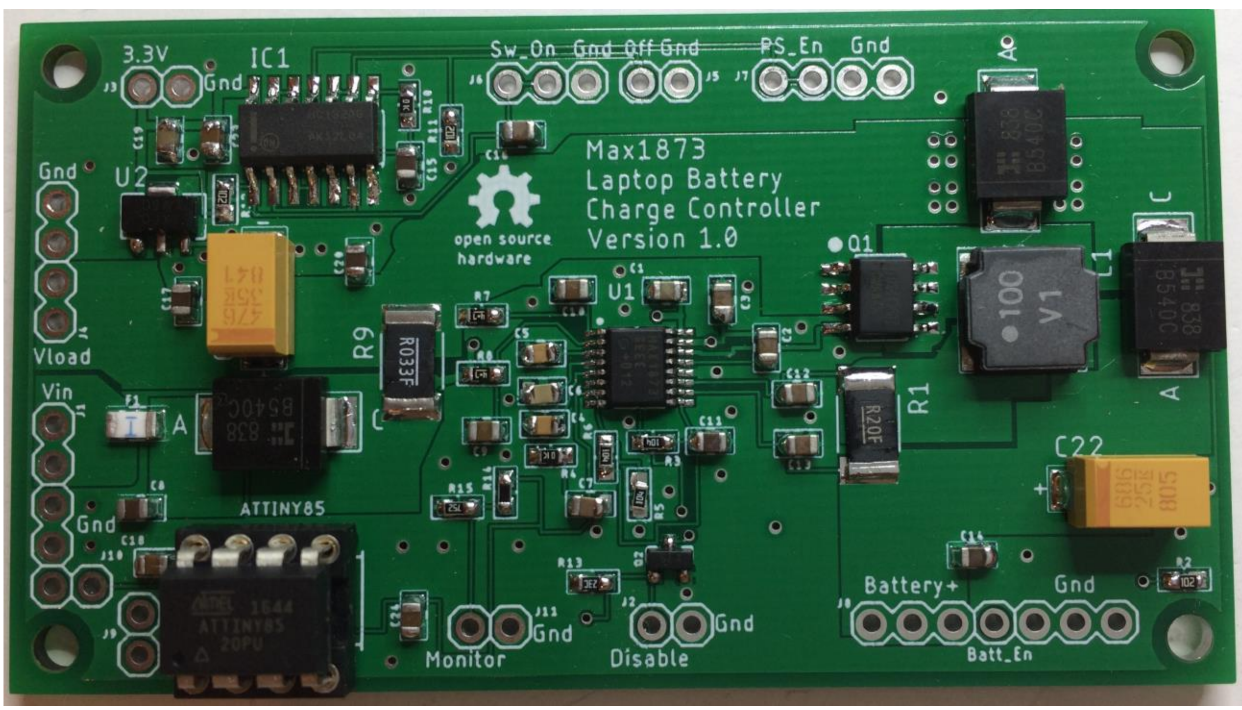
I used too much solder paste as you can see below. A stencil would fix this problem.



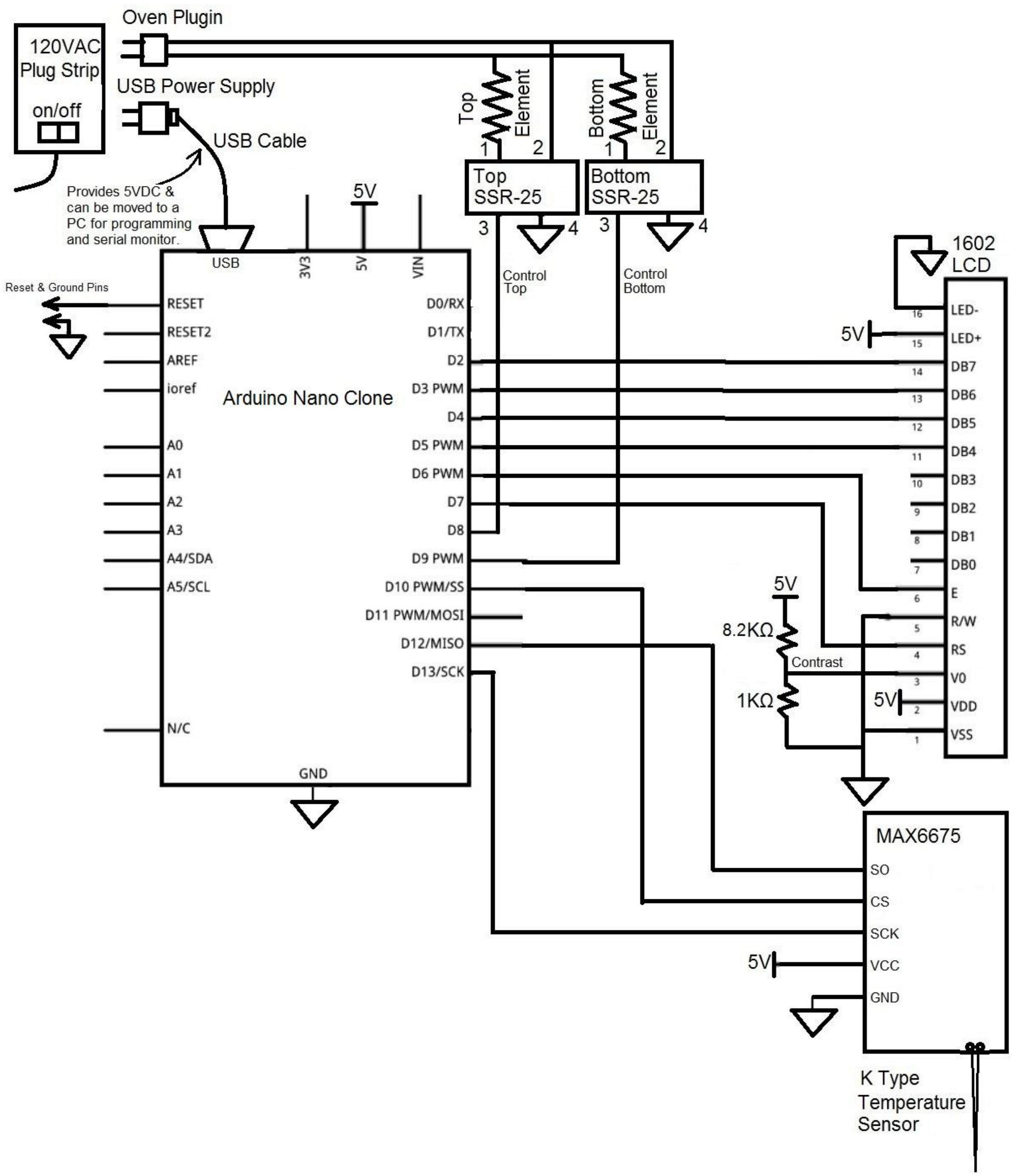
After reflow, you can see the parts that moved and have shorts.



After cleanup with solder wick and isopropyl alcohol.



Schematic – The low and high voltage wiring is shown below. The oven plugin goes to a plug strip with an on/off switch. A USB power supply is also plugged into the strip to provide 5 volts DC for the Nano, LCD, and Max6675. The USB cable can be moved to a PC to reprogram the Nano and to read the temperature data with the Arduino serial monitor. The top and bottom oven heating elements are turned on & off with separate Fotek SSR-25 relays. The 1602 LCD is configured in 4 bit mode to reduce the Nano's I/O count. Fixed resistors are used to set the LCD contrast voltage instead of using a potentiometer. The MAX6675 digitizes the signal from a type-K thermocouple and outputs the data over a 3 wire SPI bus to the Nano.

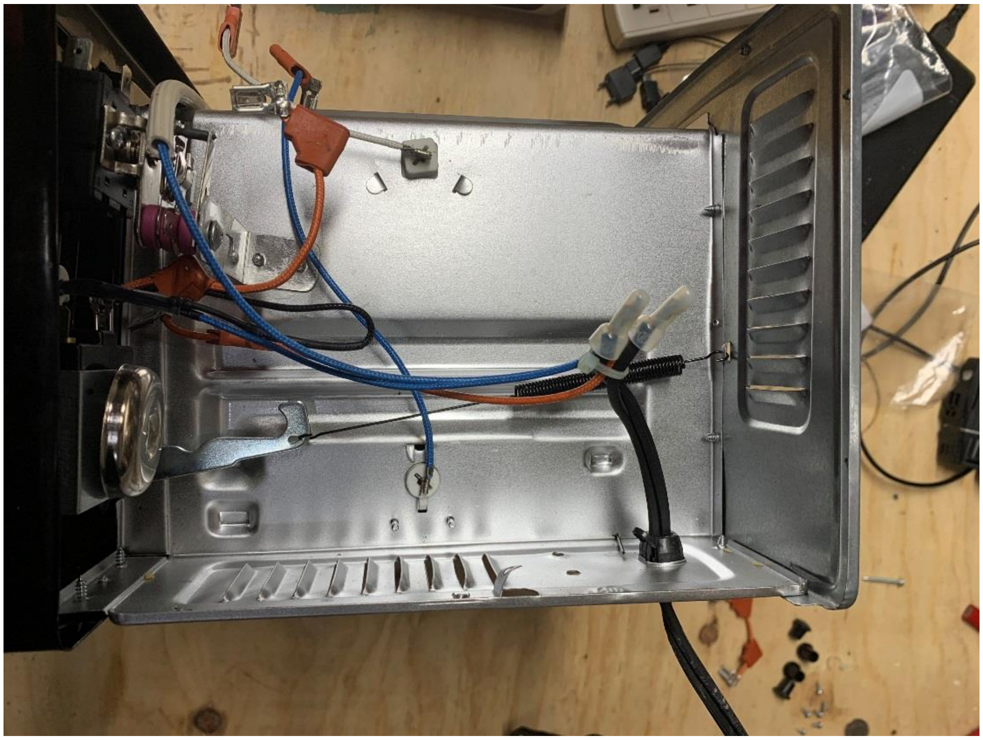


Build description – The following pictures show how the solder reflow oven was built.

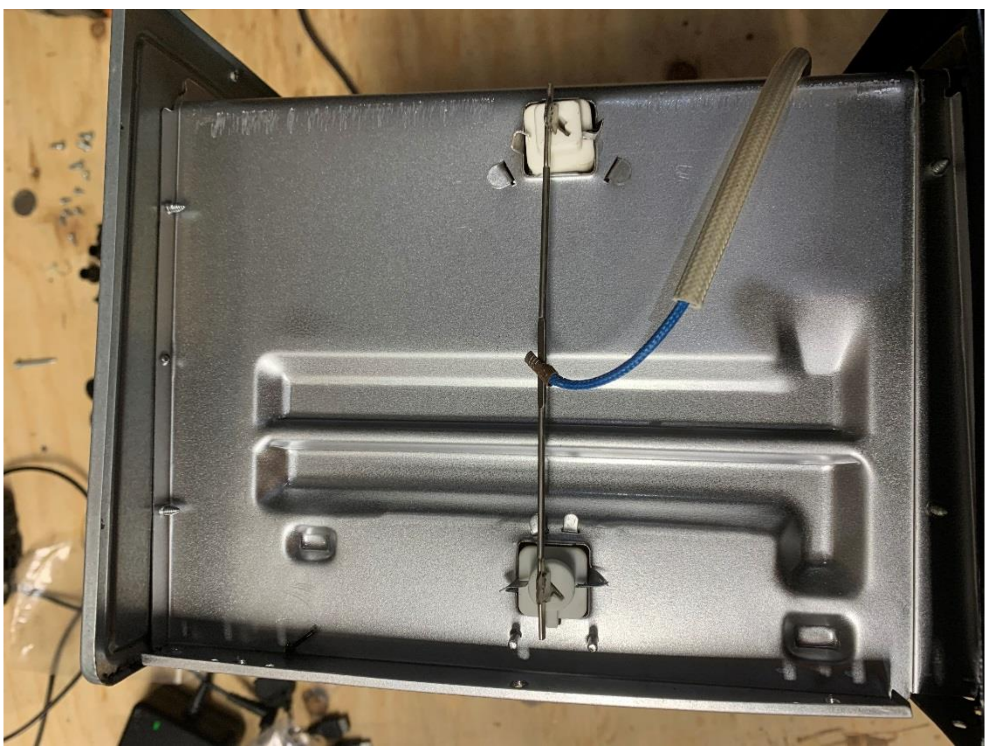
The toaster oven came with a wire rack that I don’t use.

The back side has a “pizza bulge” and the bottom has a trap door to clean out the crumbs.

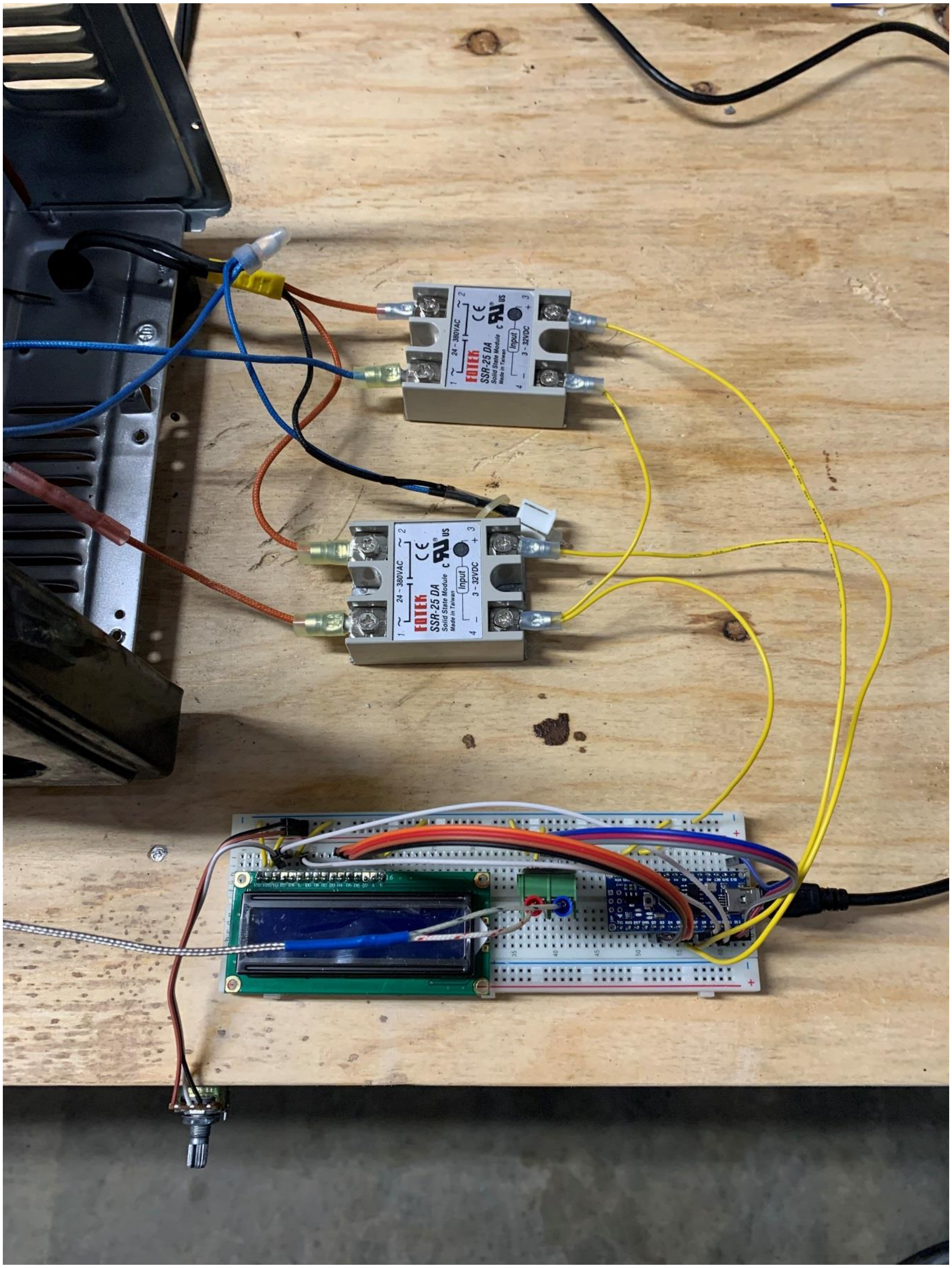
This shows the AC connections to the top and bottom elements.



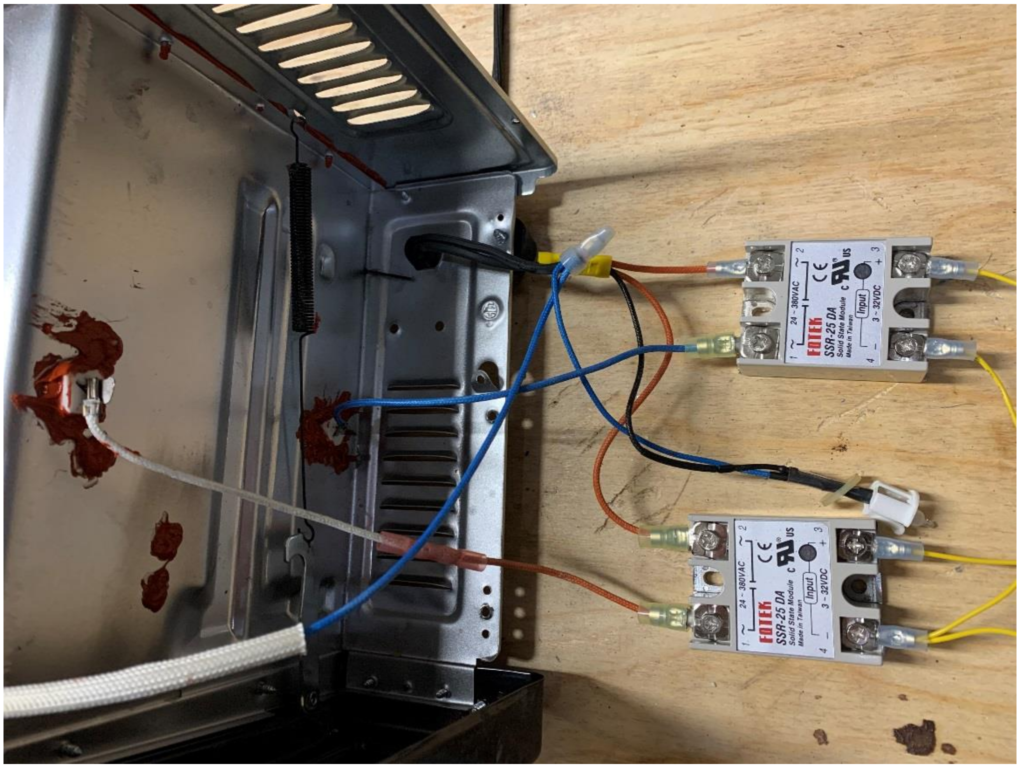
The high temperature insulated blue wire goes to the other side and is tied to both elements.



Lugs were attached to the high voltage wires so they could be connected to the SSR-25 relays. The Arduino Nano, 1602 LCD, and Max6675 module were connected using a breadboard for initial testing.



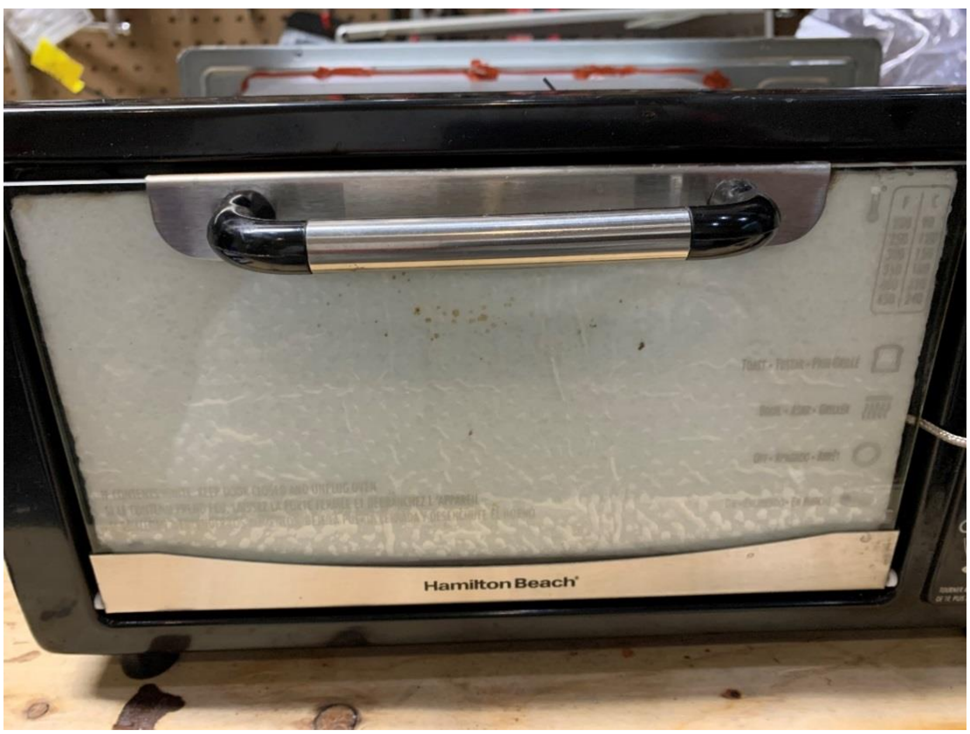
High Temperature RTV red sealant was applied to all holes and gaps in the oven wall.



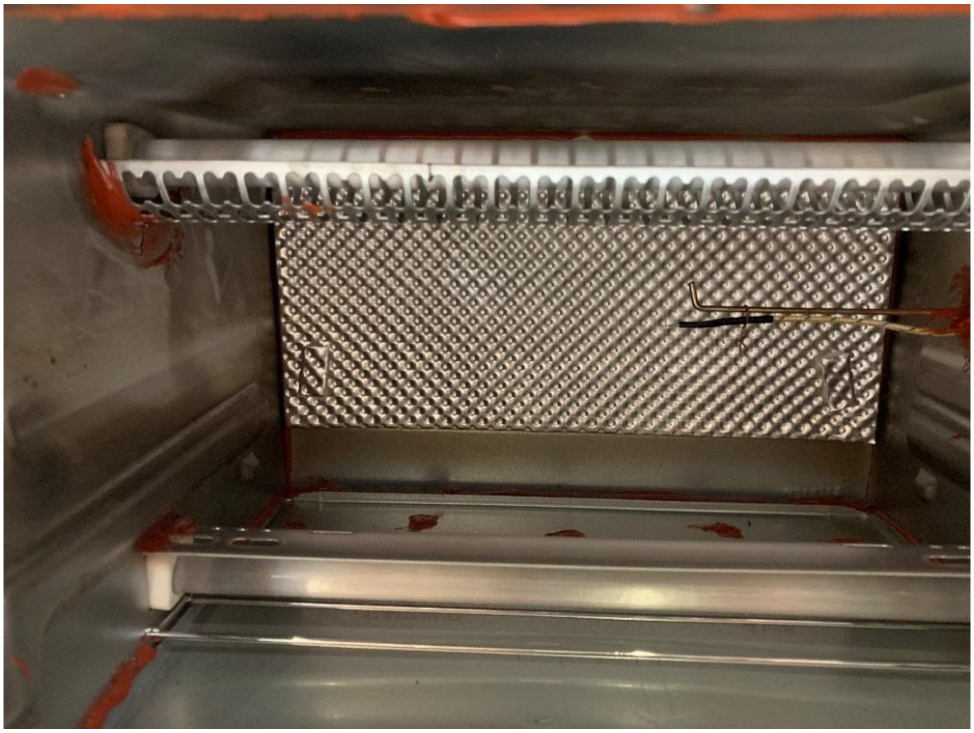
The worst air gaps were in the hinged floor.

The front of the oven felt like it had a lot of heat loss so I decided to use metal faced insulation called Floor and Tunnel Shield on the glass door. I chose not to cut an opening to look thru during operation but I may add one in the future.

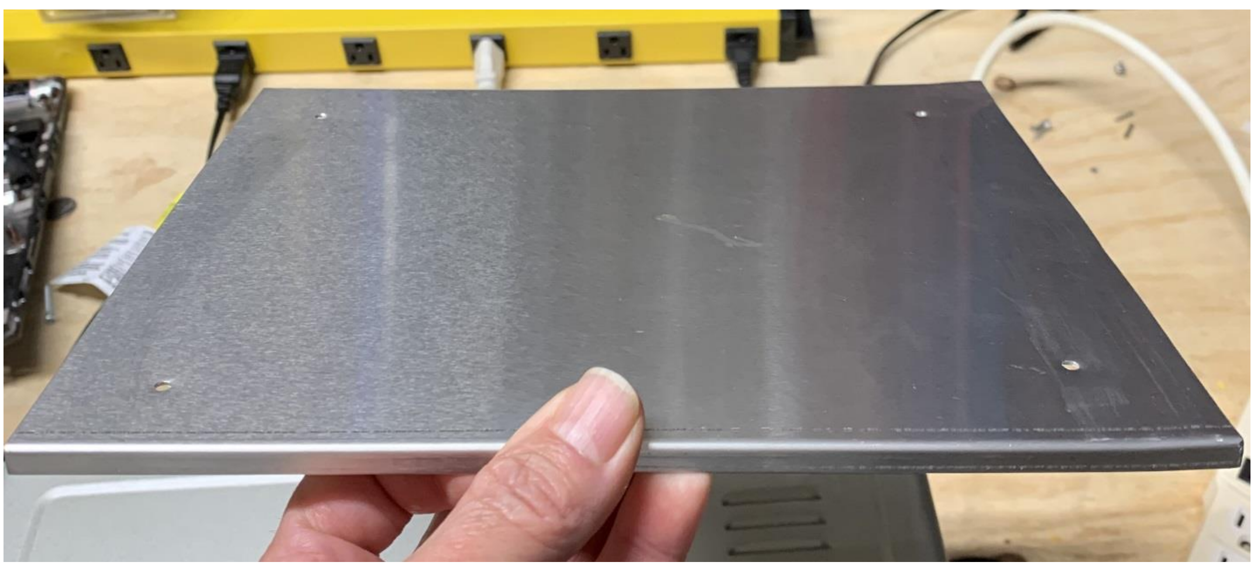
The insulation makes a pretty good seal around the edges of the door so I didn't add a high temperature gasket.



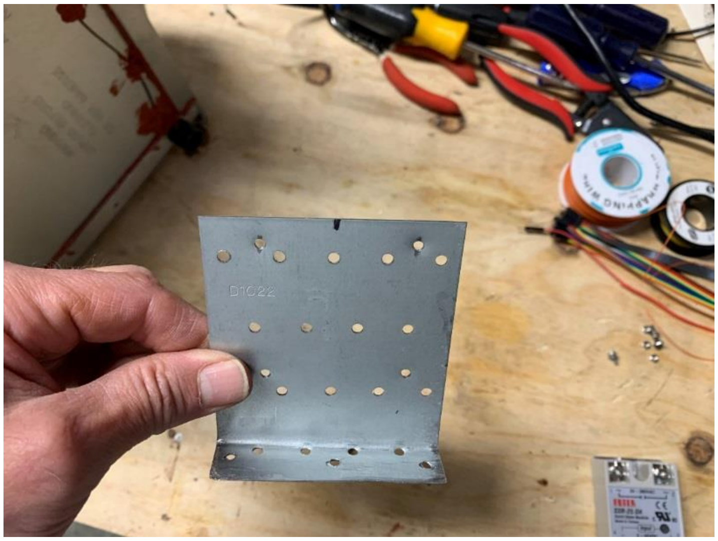
There was just enough insulation left over to cover the pizza bulge in order to reduce the area of the oven. This picture also shows the metal rod (cut from the oven rack) that was used to support the original heavy thermocouple. It's not needed now that I use a low mass thermocouple. There are two extra insulation squares on the back wall to make sure there is a gap for airflow when the aluminum tray is installed.



This is the 8" x 10-5/8" aluminum PCB tray from EBay. My oven tray needed to be cut to 9-5/8" so I drilled 4 holes to hold the plate on my workbench while I used a jigsaw to trim the sides. At 1/32" thick, the plate would bow if not for the 90° flange on the front that adds strength.



A metal tie plate from Home Depot was bent with a shop vise and drilled for mounting the SSR's to the floor in the control chamber.

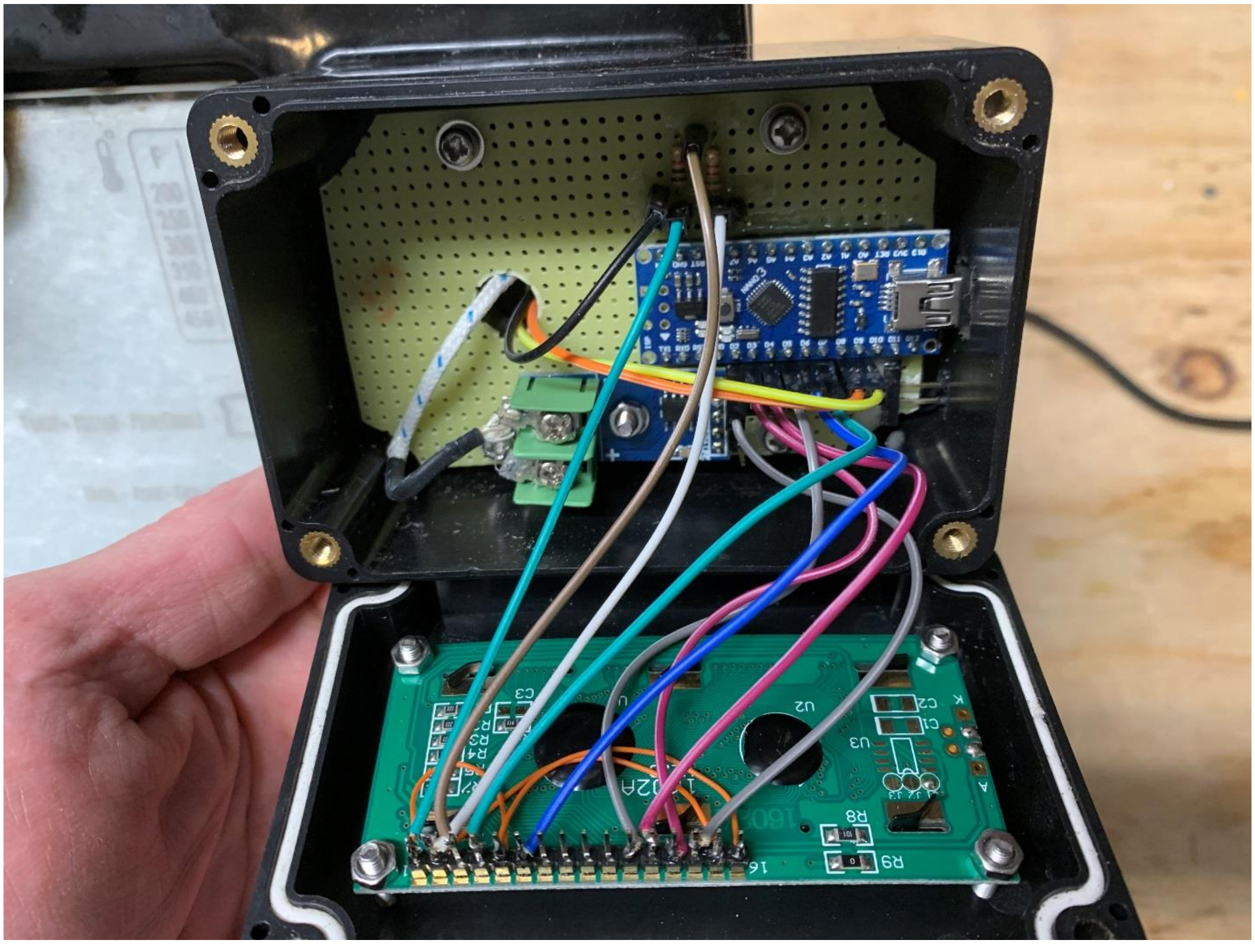


Ceramic fiber insulation was cut to fit the sides and top of the oven. This insulation keeps the control chamber from getting hot so the SSR's don't need a heat sink or fan. I wore a dust mask, safety glasses and gloves during the installation.

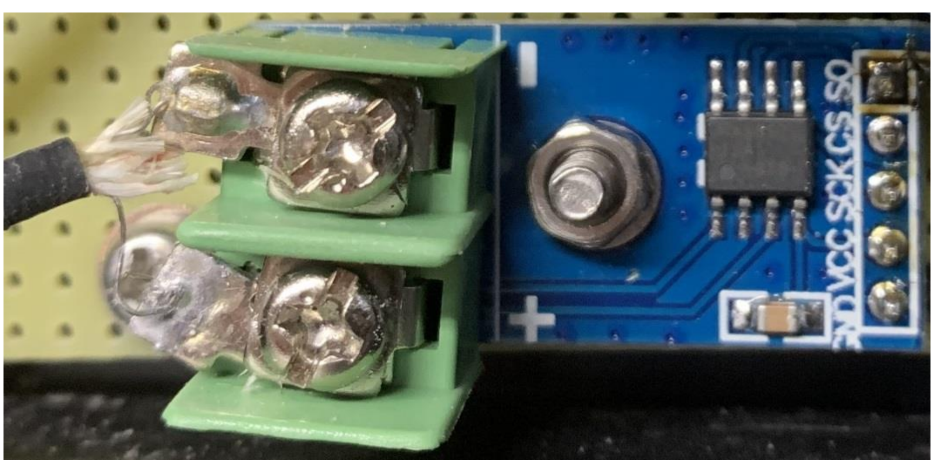
The top half of a plastic project box was cut with a jig saw for the LCD.

The bottom half of the box was attached where the oven control knob was located using the original 4 screws threaded directly into the plastic. The large hole from the original control knob provides access for the wires.

A perf board was used to mount the Nano and Max6675 in the box. Header pins were used to provide easy connection to the board for the LCD and SSR controls. A hole in the side of the box provides access to the Nano's USB port.



The low mass thermocouple had two flat pins for installation in a meter so I took the plug apart and cut off most of the metal from both pins, leaving just enough to slide under the screw terminals of the Max6675 module.



The LCD is programmed to show the elapsed seconds, profile stage, temperature, and the top and bottom relay PWM control levels (3 and 6 in the picture below).

When it’s time to open the door and cool down, “OpenDoor” is shown. The downward slope is displayed so I can open the door more or less in order to get the desired -2° C per second.

When complete, the program displays the maximum temperature, reflow time, soak time, and time to peak.

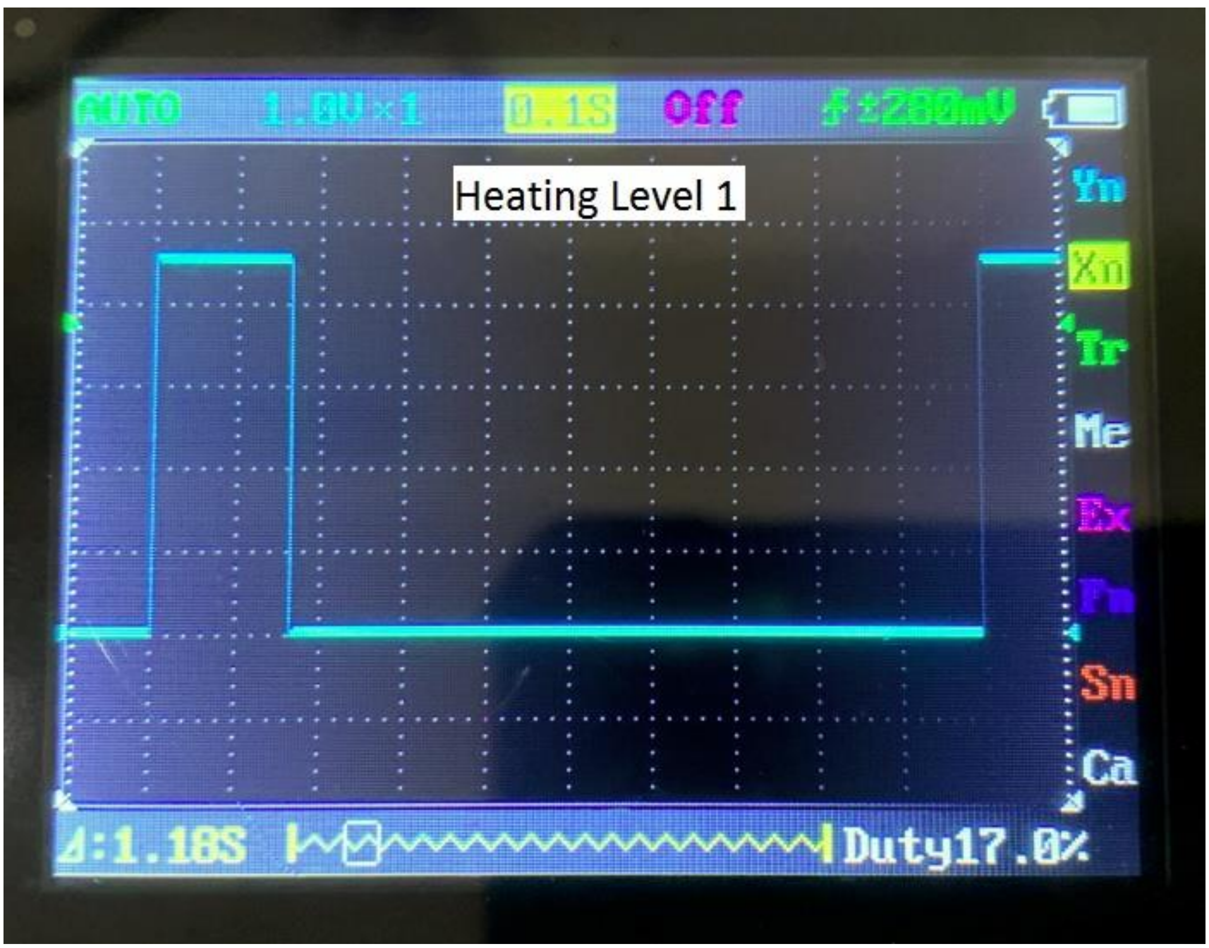
This shows the oven in use, connected to a PC so that the serial monitor in the Arduino IDE can record the time and temperature. The data is comma delineated so it can be loaded into Excel for plotting the temperature versus time. New Arduino code can also be loaded into the Nano with this configuration. A plug strip with an on/off switch is used instead of adding a switch to the oven.

If the PC interface is not needed, 5 volts DC is provided from a USB power supply plugged into the plug strip. I used a TM902C temperature meter to check the accuracy of my circuit so I didn't melt any more boards.

Code Description – My code is very rudimentary with lots of room for improvements. Someday I may learn how to use the Arduino PID functions, interrupts, timers, and all the other fancy stuff but for now, I'll just enjoy having a reflow oven. The Nano reads the oven temperature from the Max6675 over a SPI bus using the code from this stackexchange post. The Nano then displays the time and temperature on the 1602 LCD using the liquidcrystal.h Arduino library. The main loop repeats every second and uses a switch case statement to adjust the heating based on the temperature. The two heater relays are independently controlled and can have 7 different drive strengths. The "On Time" and "Off Time" pulse width are defined in the table below:

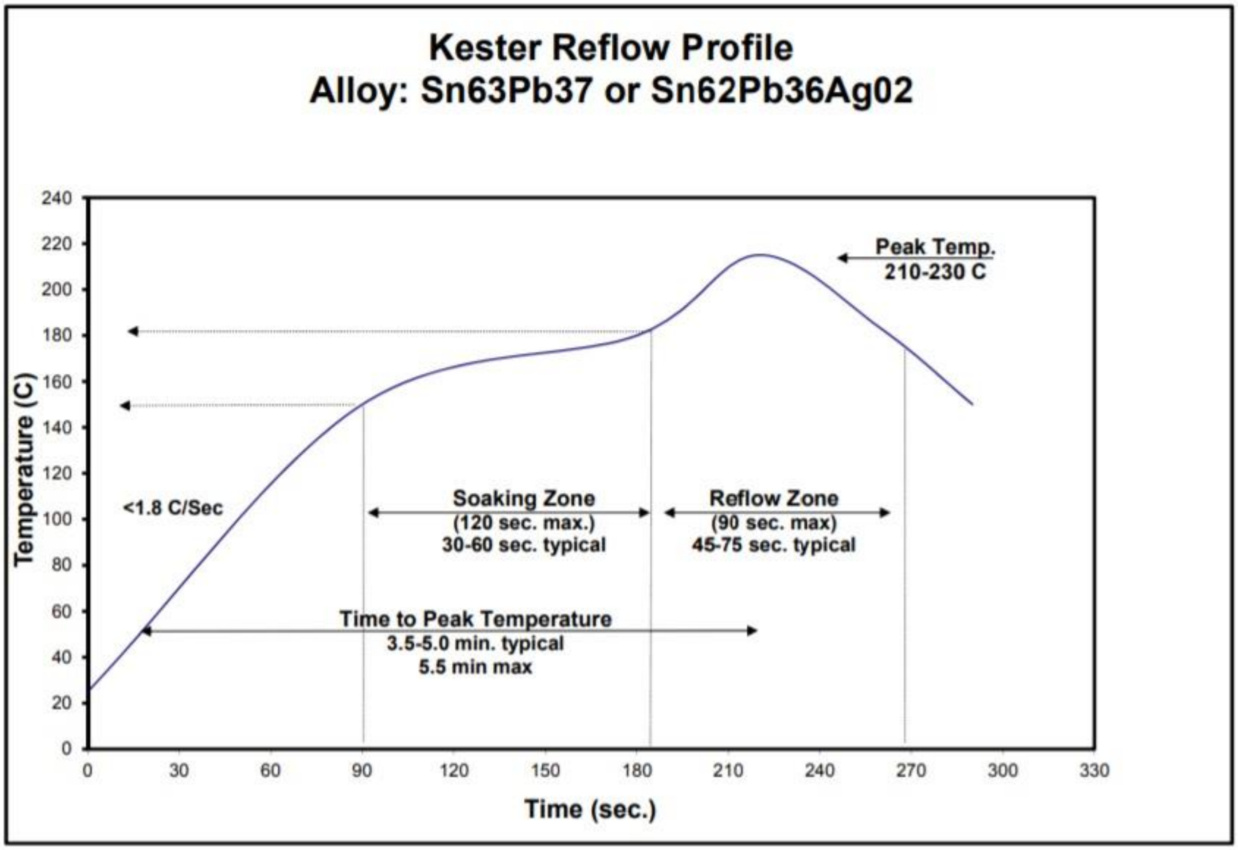
Heating Level	On Time	Off Time	Strength
0	0msec	1000msec	0%
1	167msec	833msec	16.6%
2	333msec	667msec	33.3%
3	500msec	500msec	50%
4	666msec	334msec	66.6%
5	833msec	167msec	83.3%
6	1000msec	0msec	100%

This is a PWM control signal set to heating level 1.



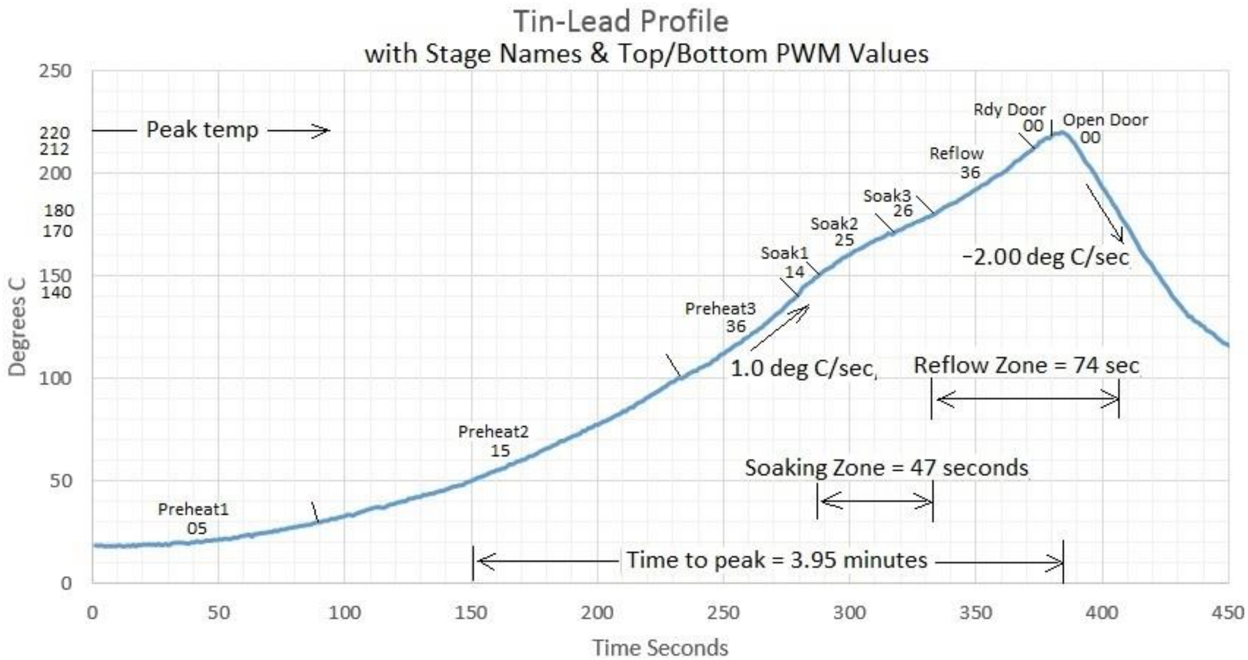
Once the oven modifications were completed, a dozen test runs were performed to determine the PWM settings and temperature trip points.

Desired Temperature Profile – I use Kester, no clean SnPb solder paste purchased as part number 247-SnPb-15 from Amazon for $10.99 in a 15 gram syringe. Kester provides the following profile on their website.



No ramp down rate is given but other sites say the down rate should be less than 5 deg/Sec and 2 deg/Sec is optimal.

The profile shown below was produced after the modifications and test runs were completed. All parameters from the Kester profile are satisfied. The stage names and PWM values shown on the LCD are given on the curve. 8 seconds after reaching 212°, the code displays "OpenDoor". At this time, the temperature has risen to 220° but as soon as the door is cracked open, it starts to fall. The door opening is adjusted manually based on the down-slope value on the LCD. The door is completely opened by the time the temperature reaches 130°.



Future Changes – The following changes could be made to the oven in the future:

Use two of the analog inputs to sense when two push button switches are pushed. Each switch would be pulled up to 5 volts with a resistor. Pushing a switch would tell the program to select a temperature profile or adjust the temperature manually.

Use an analog input to read the voltage from the center pin of a potentiometer that is tied from 5 volts to ground. The voltage could be broken down into several ranges so the operator can select different temperature profiles.

Add high temperature foam gasket around the door edge.

Cut out a small square of the insulation on the door if I find it necessary to see inside the oven.

Add Reflect-A-Gold tape to the walls of the oven.

Add floor and tunnel shield insulation to the oven floor and ceiling.

Change the 2 wire power cord to a 3 wire cord so that the oven metal case can be tied to safety ground.

Wire the timer/bell with the 120 VAC input so it can act as a turn-on switch for the oven or add a separate on/off switch.

Add a 5 volt USB power supply inside the oven that feeds the Arduino.

Use a PID algorithm in the Arduino code.

Add solder profiles for other types of solders.