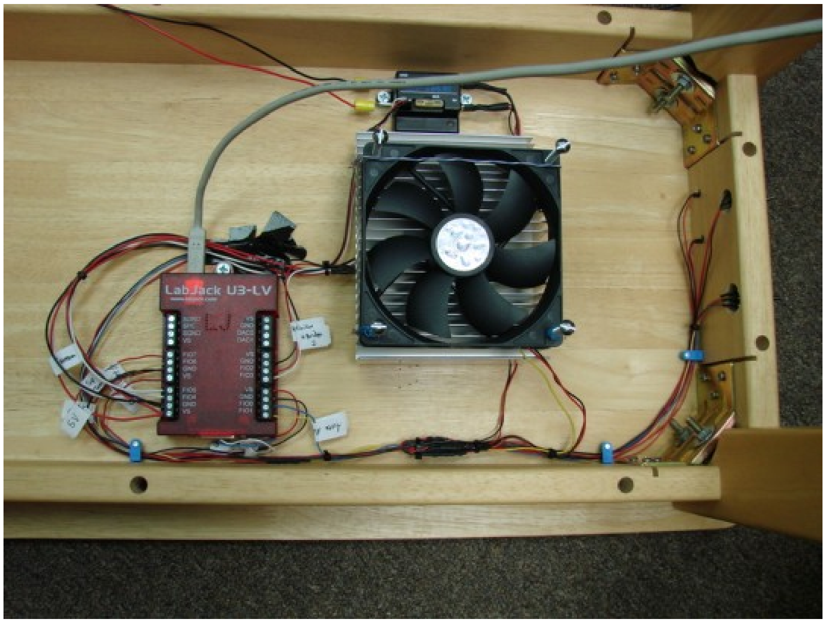
Published on *LabJack* (https://labjack.com)

Home > End Table Drink Cooler

End Table Drink Cooler

Submitted by LabJack Support on Thu, 06/10/2010 - 17:09



The goal of the project was to automatically detect whether a drink is hot or cold, and keep it that way, without the mess of an extra device typically associated with peltier drink cooling.

The main components include: 12V 60W PSU, Labjack U3, IR Thermopile Sensor, H-Bridge, and a 60W Peltier Cooler. Wiring from the U3 goes to LED indicators and switches on the side, along with 2 temp sensors, the IR sensor, and the H-Bridge for polarity reversal.

How To Make Your Own

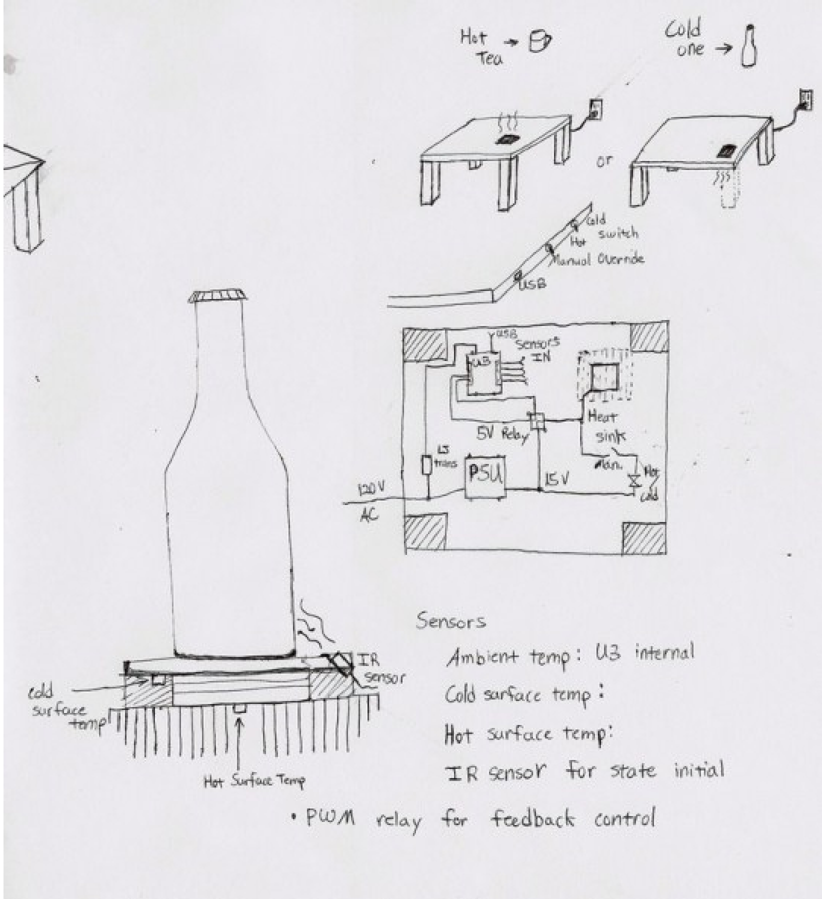
It takes a bit of programming, woodworking, and soldering experience, but the rest is straightforward enough. Most tinkerers/hobbiests could assemble something similar.

Materials:

- Labjack U3-LV
- 60W Thermoelectric Peltier & Insulation - purchased from a surplus center with heat sink already attached

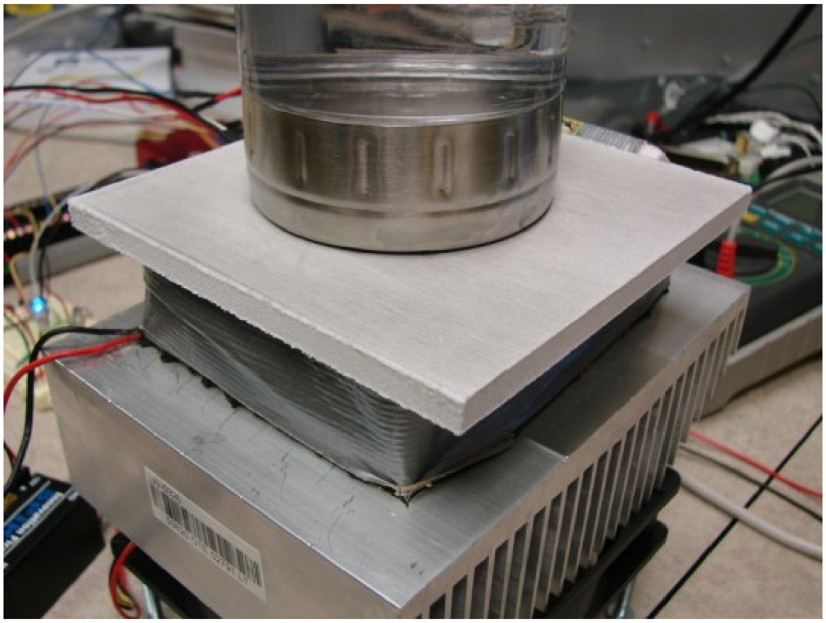
- Heat Sink
- Top Surface Plate
- H-Bridge Relay
- 12V 60W Capable PSU
- LED indicators & Resistors
- Thermal Paste
- 120mm Fan
- IR A2TPMI334 Sensor
- LM34 Temp Sensor
- Wire

Concept
When a drink is placed in front of the IR sensor, the reading will either be hot, cold, or ambient; Based on this information, a script running on the Chumby activates the H-Bridge via the U3-LV to turn on the Thermoelectric Cooler. The live data gathered while the script is running is exported to CloudDot. Once on CloudDot, any device with internet can access the information. At the end of the video, the Chumby is simply displaying a few of the relevant temperatures.



This is the initial sketch of the project, however some of the specifics with regard to wiring/switches changed during assembly.

Step 1: Find/build a surface contact plate

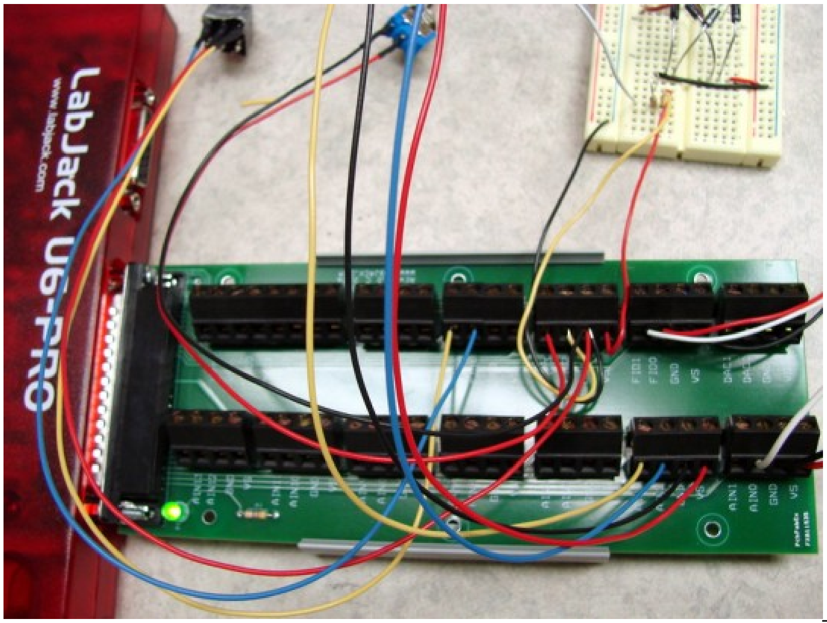


Assemble the heat sink with the peltier and surface plate. The top surface shown is an aluminum plate machined from some scrap. Dimensions approx 1/4" x 4" x 4"

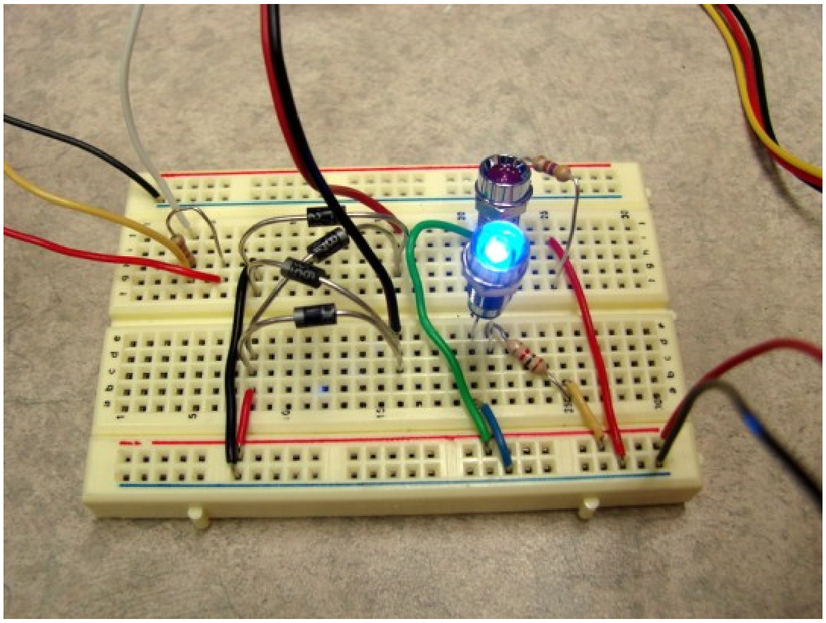
Step 2: Connect H-Bridge for polarity reversal (cold *or* hot top)

Shown is an old Spike motor controller that was lying around, but any digitally controlled H-Bridge capable of the current draw through the peltier would have worked. Wire the PSU to the input lines, and the peltier on the M- M+ lines. Connect the digital control wires to any of the Labjack U3 <u>FIO</u> (Flexible Input Output) terminals.

Step 3: Wire in the sensors, switches and LEDs



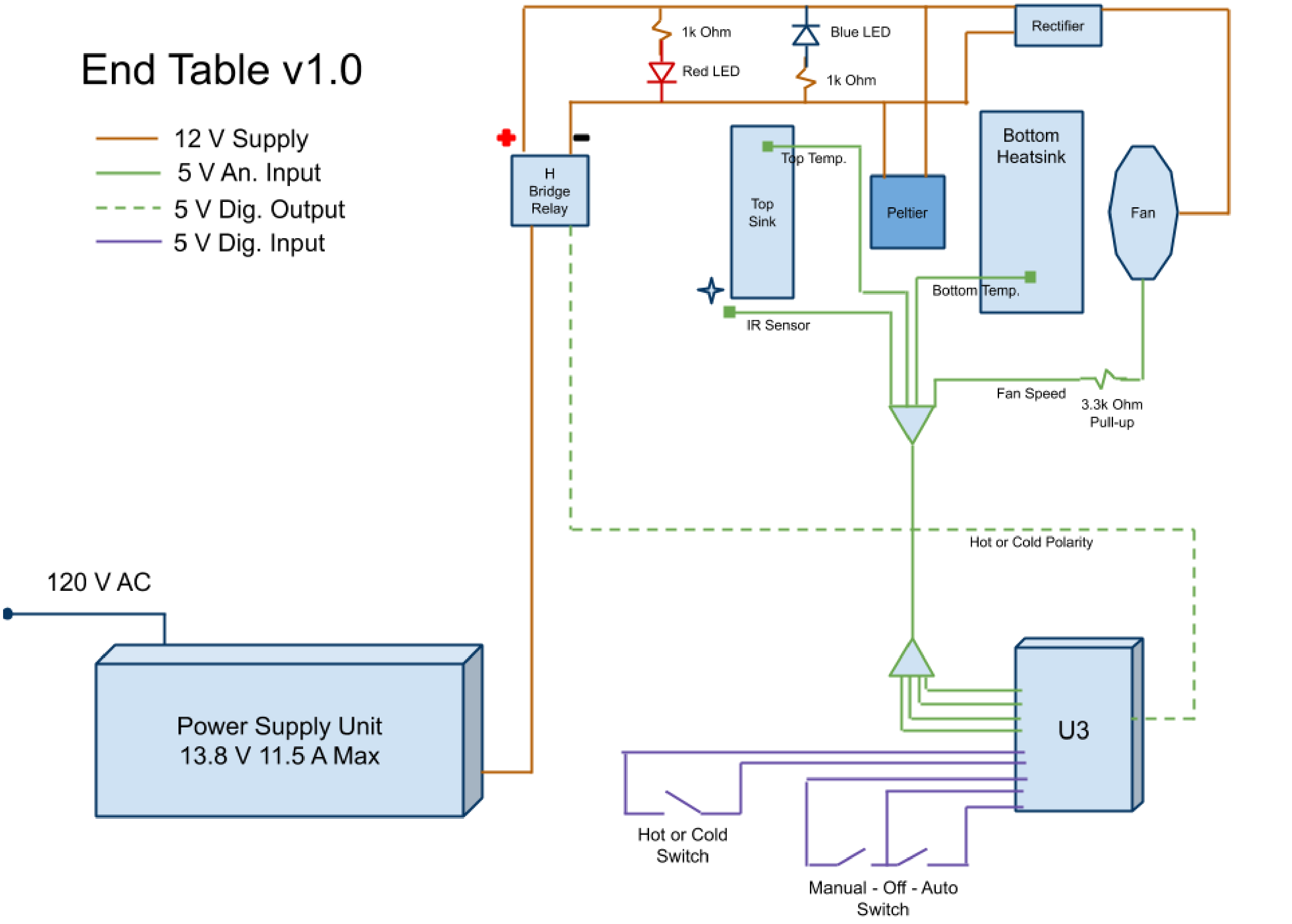
Shown is an On/Off/Manual Override switch, and a Hot/Cold switch (if on manual) wired to three of the Labjack FIO ports. Note: The Labjack has an internal pull-up resistor for all of the FIO ports, so when the switch is floating, it will register digital high. I used a U6-Pro for testing, but later adapted the script to a U3-LV since it is more suited for this application.



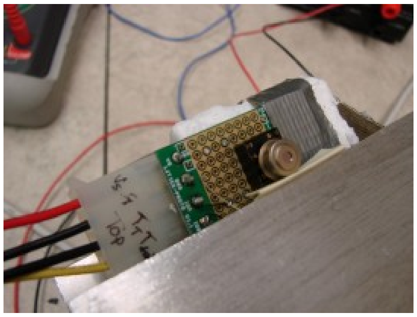
The LED indicators, blue for cold, red for hot. Also a simple rectifier using 4 diodes so that the fan will work on either polarity. The block diagram below explains the circuitry.

2 Dec 2020

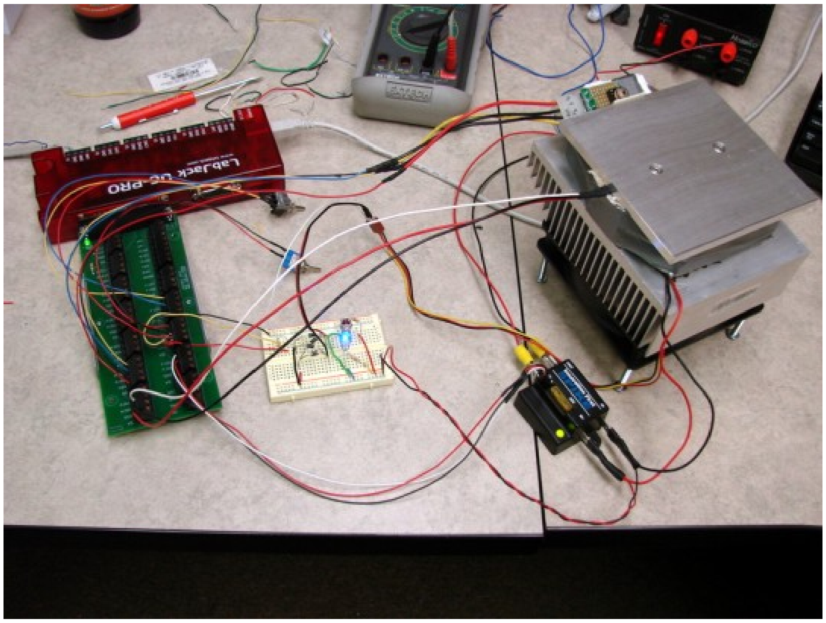
End Table v1.0



The diagram shows key components to the setup, including where monitoring sensors should be placed.



The IR thermopile sensor pointing upwards at about a 45 degree angle towards space above the surface plate. This sensor is used to indicate if the drink being set on the table is initially hot or cold.



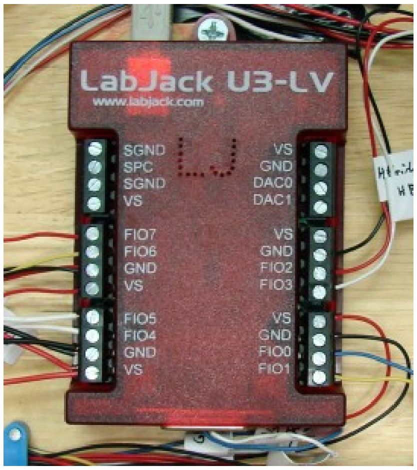
The completed test setup.

Step 5: Writing a control script

There are many ways to go about writing the script, but I chose to use Python, because it's flexible and easy to understand. The essential parts of the script are shown below, feel free to download the full version from the links on the bottom.

```
def hot(d):
    d.setFIOState(3,0) # FIO3 Low
    d.setFIOState(2,1) # FIO2 High
def cold(d):
    d.setFIOState(3,1) # FIO3 High
    d.setFIOState(2,0) # FIO2 Low
def off(d):
    d.setFIOState(3,0) # FIO3 Low
    d.setFIOState(2,0) # FIO2 Low
```

Functions which control the H-bridge polairty. Note the digital lines of the bridge are connected to FIO2, FIO3.



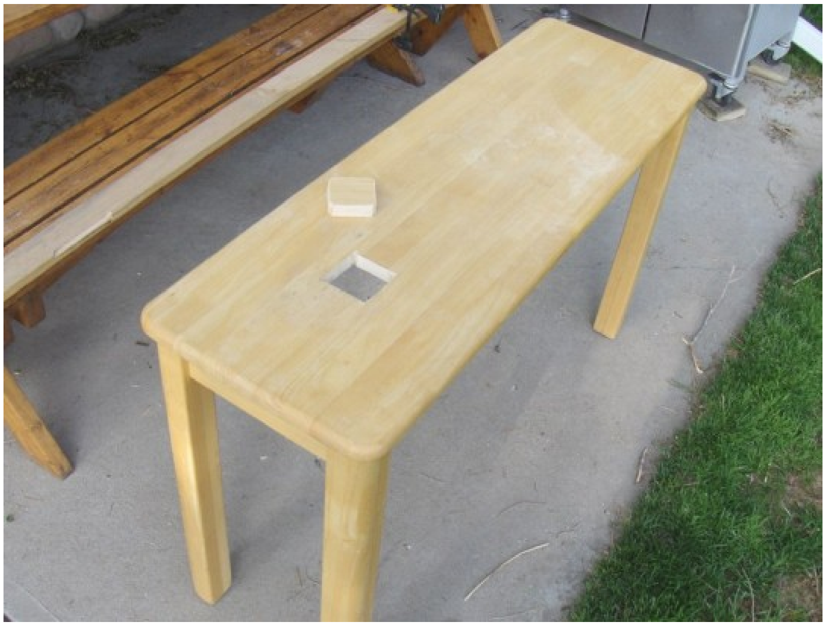
A close up of the U3-LV showing the FIO screw terminals, along with +5V VS and GND

```
    if not Auto: # Switch is in Automatic position
        print 'Automatic'
        if Tdrink > (AMB_TEMP + 8): # 8 degrees more than ambient
            print 'Hot!\n'
            hot(d)
        elif Tdrink < (AMB_TEMP - 8): # 8 degrees less than ambient
```

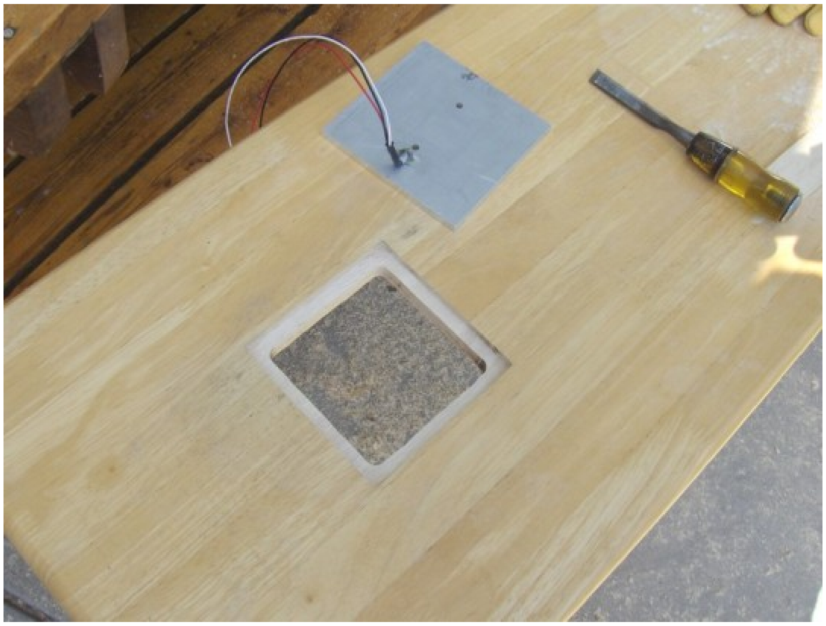
```
            print 'Cold!\n'
            cold(d)
        else: # IR temp reading ~ equal to ambient
            print 'No drink!'
            off(d)
    elif Auto and Man: # Switch is in Off position
        print 'OFF\n'
        off(d)
```

Statements which control the peltier when a drink is placed on the surface.

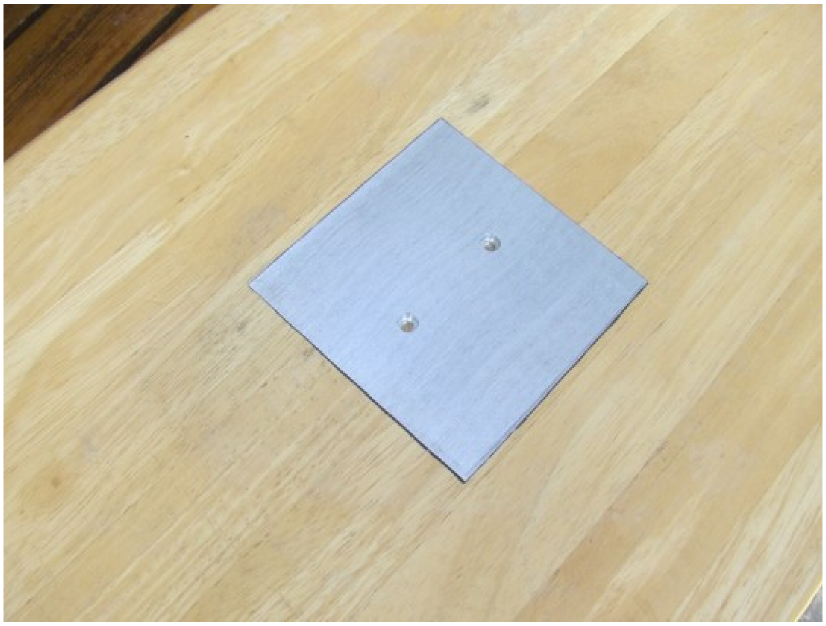
Step 6: Build it into a table



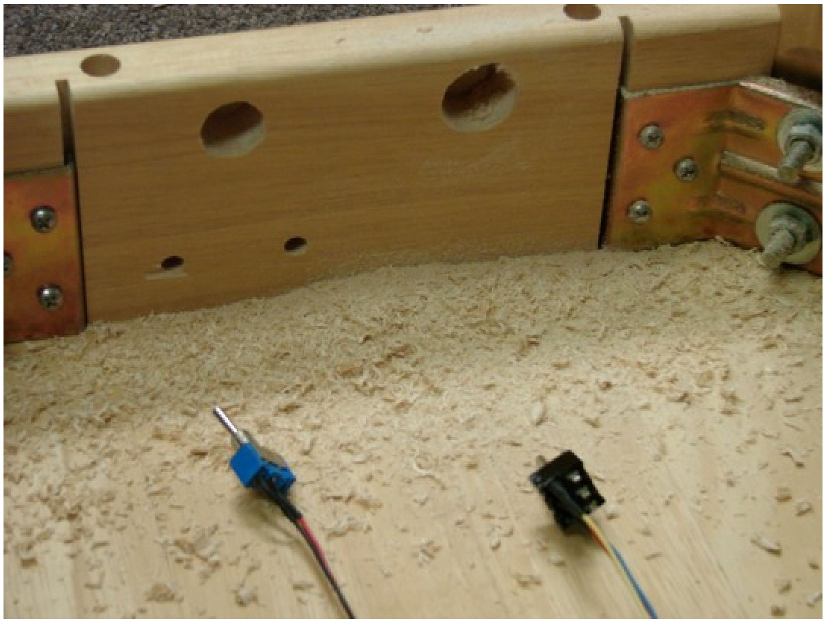
Cut an opening using a drill and a reciprocating saw. Make sure the peltier and insulation can fit through.



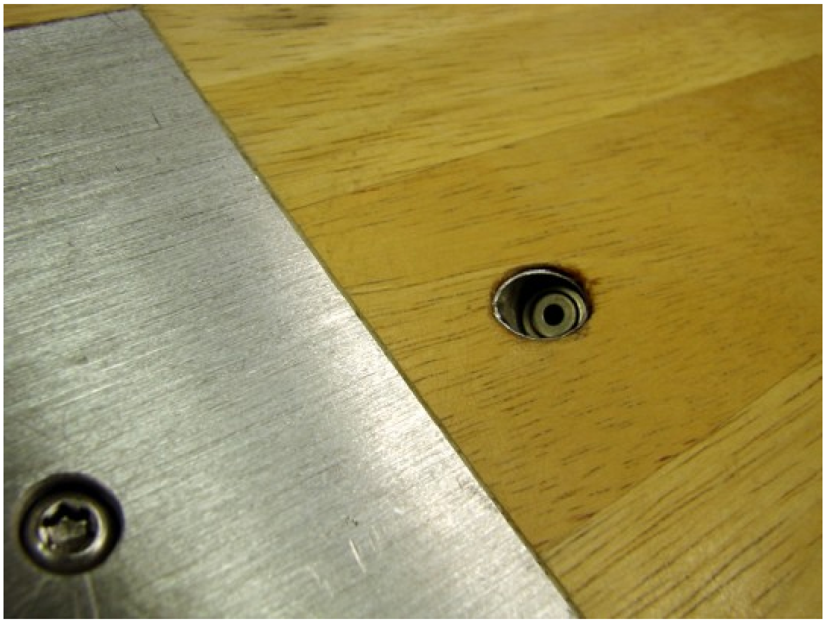
Using a router make an inset portion for the thickness of the surface plate. Then chisel out the corners.



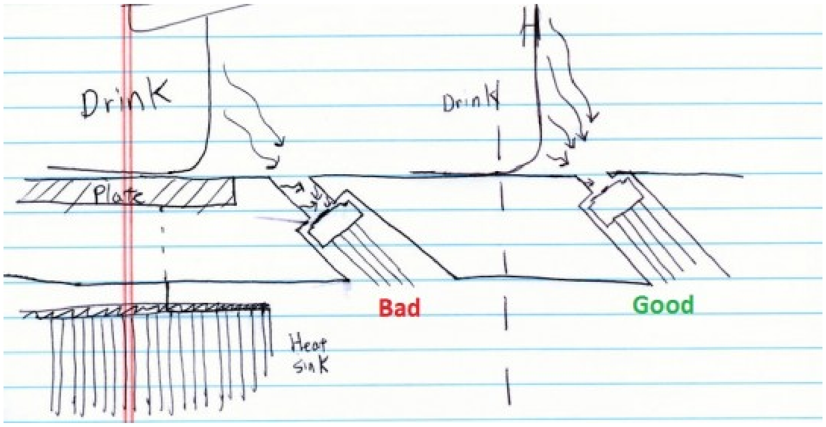
Mount the Surface plate with some adhesive caulk on the edges. The gaps should be filled to prevent condensation next to the wood.



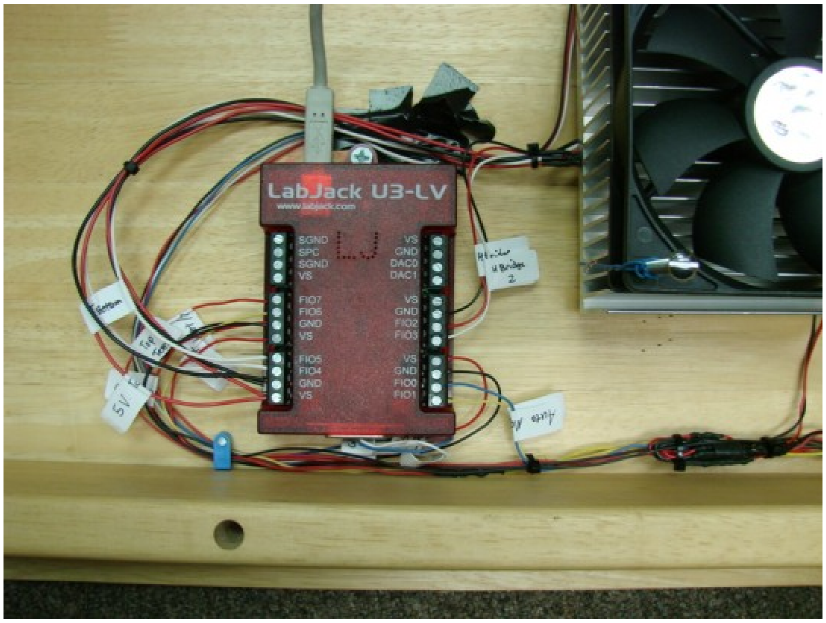
Drill out some holes for any switches or LEDs.



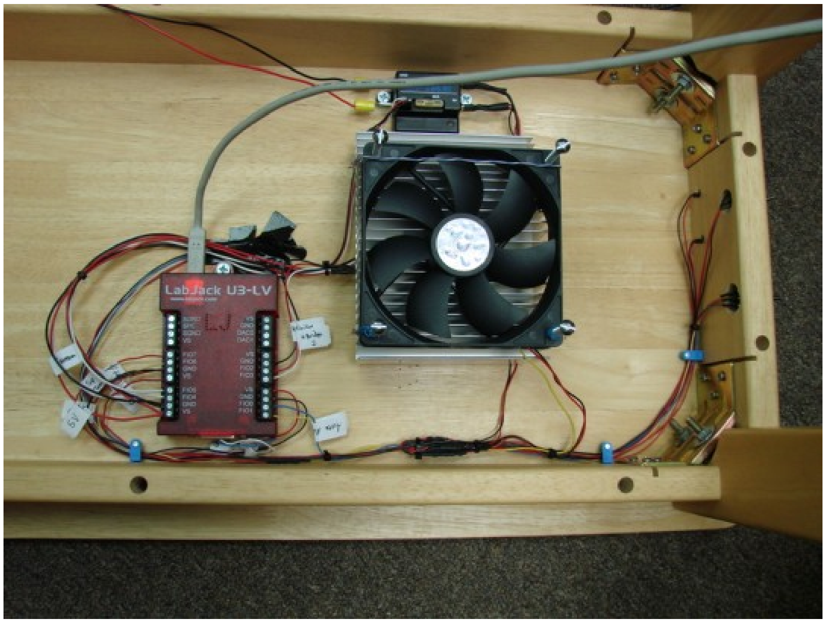
Drill a hole for the IR sensor. This is quite important to consider: If the hole is too narrow, or not close enough to the surface, the IR sensor reading will be heavily influenced by the wood temperature, instead of the drink placed in front of it. In addition, I chose to add an aluminum sleeve inside the hole, to further shield the sensor from unwanted infrared interference.



Above is a simple sketch demonstrating two different IR sensor depths.



Wire up the U3 to the sensors and switches. Make careful note of which ports are connected to what, as it will be important in the script. Note: If there are discrepancies after the connections are made, use LJControlPanel to monitor voltage readings in real time.



All finished. Depending on the PSU, it can also be stowed under the table, but mine is not shown here. As for the script, it can be run on almost any platform. A nice stand-alone solution would be to connect a wireless USB hub to the U3, then just run the script on a local PC. If there is no PC available, it is also possible to connect the U3 through a Chumby and run the script on that. In our office we opted to use a Chumby because it's so compact, and the data is accessible anywhere through CloudDot.

File Attachment:

Python Control Script for U3