



## The Connected Mezuzah



by rabbitcreek

Well its that time of year again--when tons of pseudo-religious talismans come flying out of the attic. Recently while contemplating my escape to warmer climates as the snow and darkness descended I remembered last year when the heating system in the remarkably cold time up here went dead. Luckily, I had aimed a camera at one of those 2 foot farm thermometers on the wall and I noticed its inexorable descent. The scary destructive cascade had started: freezing and bursting pipes followed by visits from the insurance company. I was just informed by Alexa that she has a thermometer on board--no way to ask the temp but good to know. So this year I built a unit that would send temperature information to my

smartphone. And what better way to disguise it then a wall decoration from my distant past. A Mezuzah used to be a common apparition in Jewish homes of my childhood. It was a tilted object mounted near the doorpost. I was told it contained something special much like a Catholic holy relic that informs most old churches. A handwritten scroll with a small prayer. The scroll I kept intact in the design--it is a 18650 battery--what better talisman for the new age. The design also suffers from a couple extra symbols besides the requisite shin. Mount it up near the doorposts of your house and keep it safe from more modern worries.



## Step 1: Gather Your Materials

Well you really need a 3D printer. Got one this summer and its really fun. Generally like everyone else I hate printers but to do design work you need a cheap \$300 one. The top and bottom STL files are included.

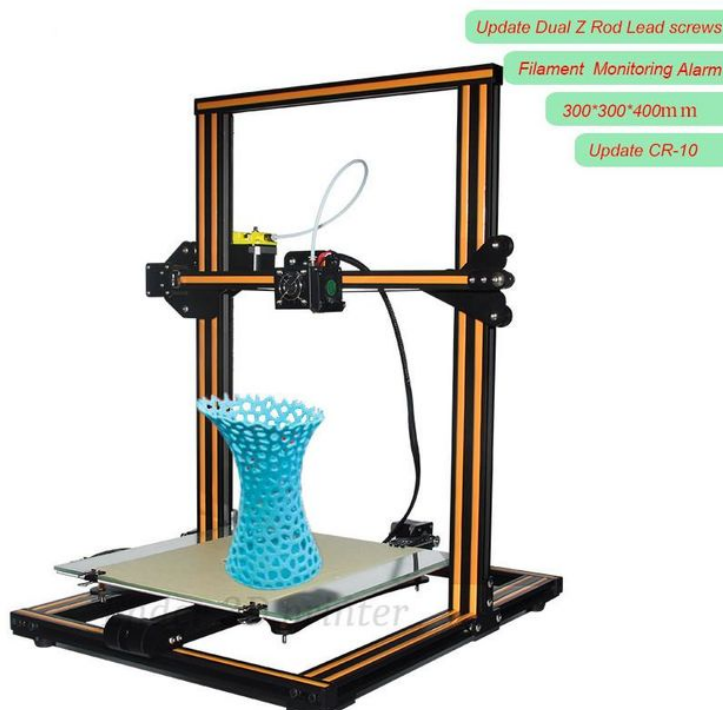
1. Feather ESP32 from Adafruit--Great board!

2. DPDT switch-- I got a very small push-on/off that fits the insert in the top well

3. Lithium Ion Cylindrical Battery - 3.7v 2200mAh  
\$9.00

4. Adafruit TPL5111 Low Power Timer Breakout --\$6  
Great little timer board

5. DS18B20 Digital temperature sensor -- you can get them anywhere for cheap (with resistor 470 k)



## Step 2: Print It

The magical object was designed on Fusion360. The design is slightly mid-century clunk modern. The small square at the top fits the switch perfectly. The hole is of course to reprogram and charge the battery. No supports are needed. Printed in PLA. The two parts are designed to be super glued together--no user serviceable parts inside. There are no mounts specifically built into the unit for wall mounting--I

really don't like these -- just use some double sided tape and wack it up there. (If its good enough for Ring alarm units its good enough for a Mezuzah). Spray paint the PLA with the Krylon "Make it Stone" paint. Its great for hiding the general chatchka nature of all 3D printed objects and gives it the sanctity of its nature.



<https://www.instructabl...>

[View in 3D](#)

[Download](#)



<https://www.instructabl...>

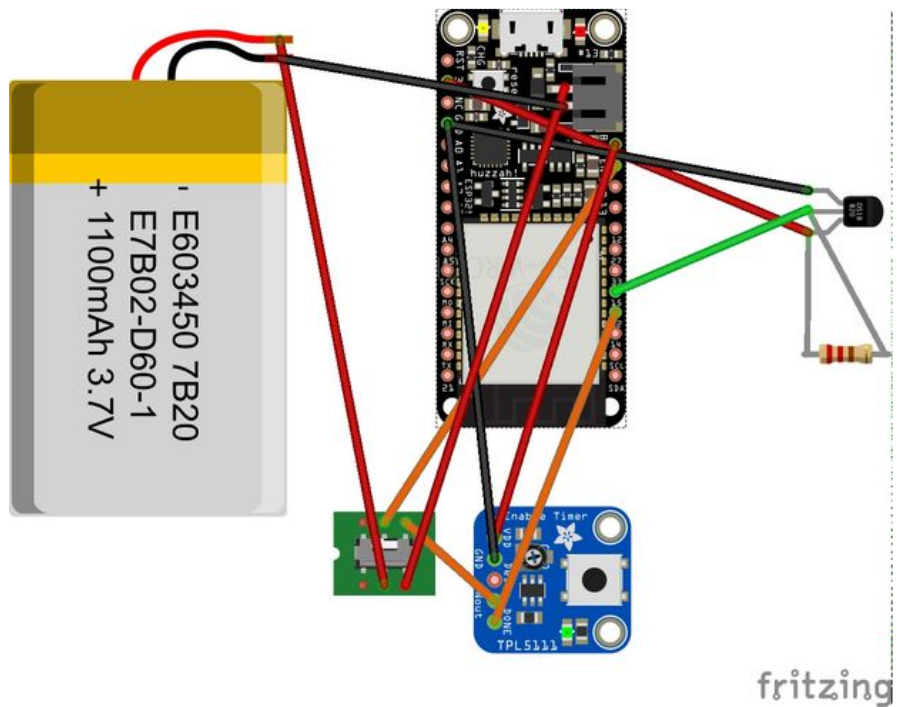
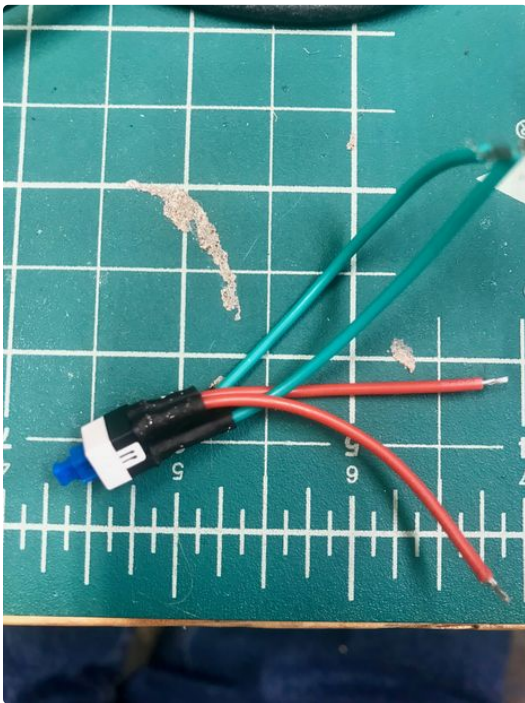
[View in 3D](#)

[Download](#)

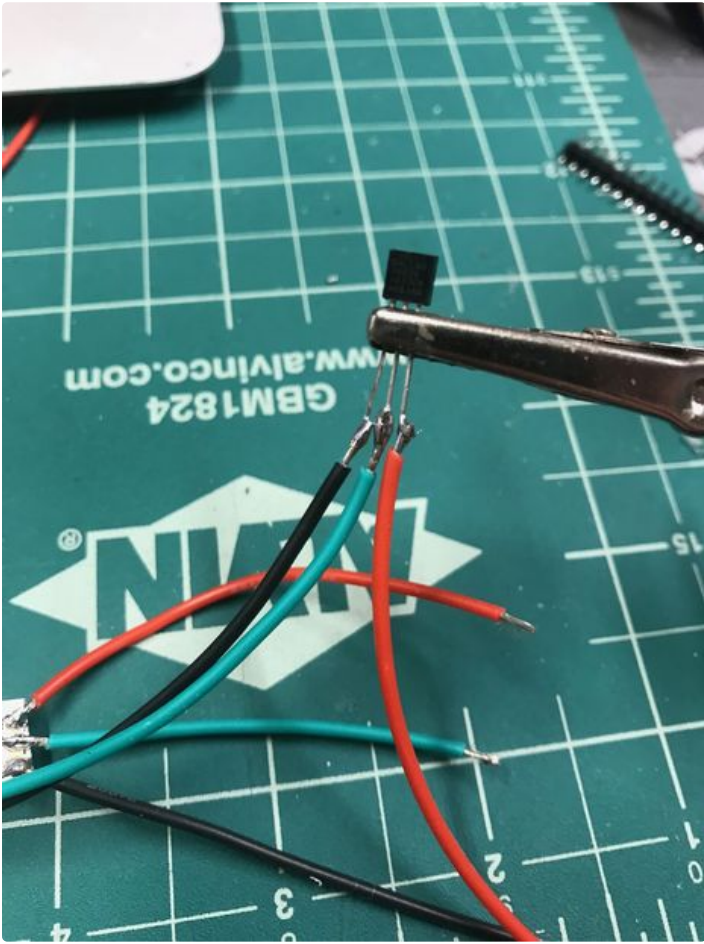
### Step 3: Wire It

As always the fritzing diagram is available for the connections. The concept is straight forward like a million other temp sensor projects. The Feather ESP32 board is used because it works. I wasted a lot of time with a couple other ones from overseas including one with a built in battery holder and there were always issues. This one works flawlessly with the Dallas One-Wire hook up and has good output/input pins. The TPL 5111 unit I used quite a few times before and its cool with a couple warnings. It is designed to toggle the enable pin on the ESP board to ground in between wake-up calls to read and send the temp. In wiring this unit you have to be able to cut out this enable line to program it -- thats the reason for the double switch--one part turns off the battery supply at the same time turning off the ground

to the enable pin allowing the board to be reprogrammed with a new wifi code or some other tinkering through the usb-micro port. The timer duration is set between wakes by adjusting a variable resistor on the board--monitor this reading as described in the Adafruit site so you get the right resistance. This cannot be adjusted once power is applied so do it early. This unit must be tied to the battery power and not the board voltage regulator for obvious reasons. The DS18B20 must be connected to power and ground and the sensor wire must be held high with the included resistor. No other magic--one input for the One-Wire readings the other for turning off the TPL 5111 when your done uploading your data.







---

#### Step 4: Assemble It

Just stuff it in.



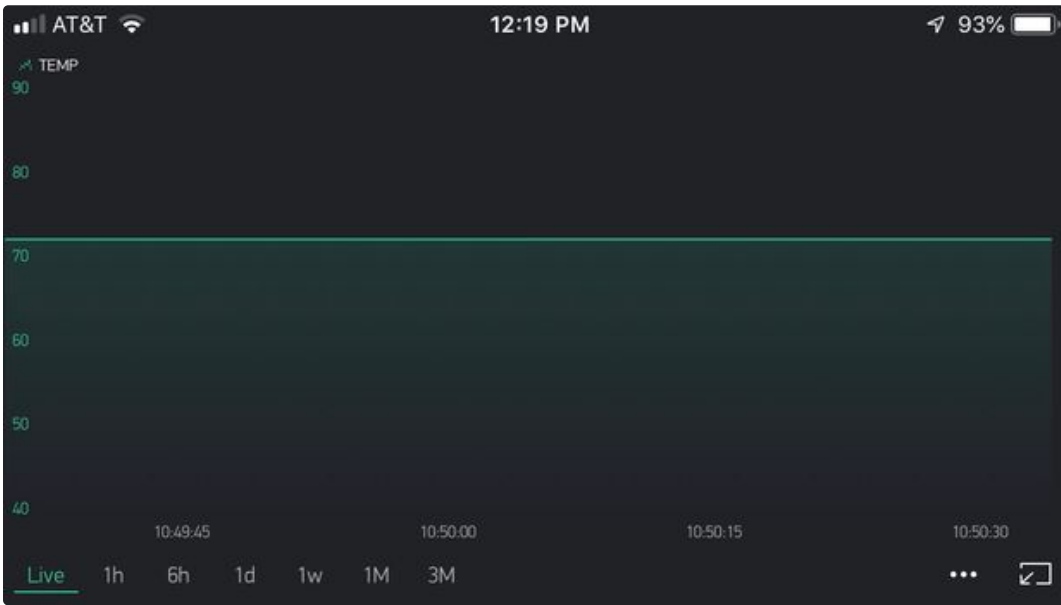
---

## Step 5: Program It

The program is really straight forward. It is based on the standard Blynk sketch for the ESP32. The whole piece of software is basically designed for objects like this. Blynk is really nice with personalized set-ups that are very easy to build for bespoke outputs. You can set up a chart with 24 hour output graph or little boxes with the numbers displayed or other graphs or bars.

The blynk sketch is designed to run a function when a preset timer hits a limit which can be adjusted for

time. They want you to do it this way so you don't dump a bunch of data in your loop function into their server which would trigger a bad error. In this case it doesn't really matter how often you call the function because as soon as it sends the data to the Blynk server it sends a high signal to the TPL 5111 which connects the enable pin to ground and the timer starts. I have it set on 2 hour wakes to save battery life.



AT&T 12:20 PM 93%

SuperChart OK

Graph

FONT SIZE: T T T ALIGNMENT: [Icons]

DATASTREAMS: Temp

+ Add DataStream

AT&T 12:20 PM 93%

← Temp

DESIGN

STYLE: LINE COLOR

INPUT: V6

Y-AXIS SCALING

AUTO MIN/MAX HEIGHT DELTA

Set fixed Min and Max Y-axis values

MIN: 40 MAX: 90

SUFFIX



## Step 6: Using It

The battery in this unit should last a couple months-- and when its built in protection shuts it down just douse it with juice overnight through the built in battery management system on the Adafruit ESP32. Set your Wifi password and get your key from Blynk for a new project on a ESP32. Then set up a chart

and request data through virtual pin 6 and you should be off and running. The **Ring Mezuzah** and the one you built are completely compatible unlike most home automation systems.

