Make It Hot or Cold

Created by Anne Barela

https://learn.adafruit.com/make-it-hot-or-cold

Last updated on 2023-08-29 03:50:25 PM EDT
Table of Contents

Overview 3
  • Parts

Make It Hot 4
  • Hook Up

CircuitPython 6

MakeCode 8
  • Results

Use 10
  • Application
  • More on the Heating Pad
  • If you plan to use a different heater:

Make It Cold 12
  • Checking It Out

Connections 14
  • How to Vary the Temperature

CircuitPython 15

MakeCode 17

Use 19
  • Heat Sinks
Overview

How can I heat something up? How can I cool something? Controlling the temperature of a substance is fundamental to automation. Most buildings have thermostats and a heating & cooling unit. Products like Nest automate that hard job. But what about smaller requirements?

There are small-scale heating and cooling needs but not a great deal has been written to show Makers how to provide temperature control. This guide will show you some techniques for heating something up and cooling something down, along with methods to monitor the temperatures.

This work will be done by two Adafruit products: the Circuit Playground Express microcontroller and the Crickit add-on controller to provide the interface between digital signals and analog devices.

Heating things up and cooling things down can cause the potential for burns or numbing cold. Please be sure to protect surfaces and the potential for hands to not touch heated/cooled surfaces.

Parts

1 x Circuit Playground Express
Circuit Playground Express is the next step towards a perfect introduction to electronics and programming.

1 x Adafruit CRICKIT for Circuit Playground Express
https://www.adafruit.com/product/
Our Creative Robotics & Interactive Construction Kit. It's an add-on to our popular Circuit Playground Express.

1 x **Electric Heating Pad - 10cm x 5cm**  
This is hot! Literally! Apply 5-12VDC and the stainless steel fibers in this heating fabric will warm up, creating a little heating pad.  
[https://www.adafruit.com/product/1481](https://www.adafruit.com/product/1481)

1 x **Peltier Thermo-Electric Cooler Module - 5V 1A**  
Apply 5V to the red (positive) and black (negative) wires and one side will get cold while the other side gets hot.  
[https://www.adafruit.com/product/1331](https://www.adafruit.com/product/1331)

1 x **TMP36 - Analog Temperature sensor**  
Wide range, low power temperature sensor outputs an analog voltage that is proportional to the ambient temperature.  
[TMP36%20-Analog%20Temperature%20sensor%20-%20TMP36](https://www.adafruit.com/product/1331)

1 x **Panel Mount 10K potentiometer**  
Breadboard Friendly - 10K Linear  
[https://www.adafruit.com/product/562](https://www.adafruit.com/product/562)

1 x **Potentiometer Knob**  
Soft Touch T18 - Red  
[https://www.adafruit.com/product/2046](https://www.adafruit.com/product/2046)

1 x **Small Alligator Clip to Male Jumper Wire Bundle - 12 Pieces**  
These cables will be your best friends! No longer will you have long strands of alligator clips that are grabbing extension wires.  
[https://www.adafruit.com/product/3255](https://www.adafruit.com/product/3255)

1 x **Short Wire Alligator Clip Test Lead (set of 12)**  
These are shorter - if you need longer, order the longer "small alligator clip" bundles.  
[https://www.adafruit.com/product/1592](https://www.adafruit.com/product/1592)

1 x **Female DC Power adapter - 2.1mm jack to screw terminal block**  
Connect a DC power wall wart to a board that doesn't have a DC jack.  
[https://www.adafruit.com/product/368](https://www.adafruit.com/product/368)

---

**Make It Hot**

There are a great many heating elements on the market but most require a great deal of energy to provide heating. And most of those products are way too much for the needs of the Maker.

The Adafruit 10cm x 5cm heating pad is different. Apply 5-12VDC and the stainless steel fibers in the heating fabric will warm up, creating a little heating pad. On one
hand, it's just a gigantic resistor. On the other hand, it's flexible, light, and can be wrapped around a project. Originally designed for portable wearable heating pads, the fabric can be used for wearables, weather balloons, diy-bio projects, thermal cycling for materials testing, etc.

Electric Heating Pad - 10cm x 5cm
This is hot! Literally! Apply 5-12VDC and the stainless steel fibers in this heating fabric will warm up, creating a little heating pad. On one hand, it's just a gigantic resistor....
https://www.adafruit.com/product/1481

The benefits of using this pad:

- It can be controlled by lower voltages
- It can be pulse-width modulated to control temperature
- It is flexible for many projects

Ok, time to think electrical. The data sheets for this pad say that at 5 volts, the sheet may require 740 milliamps (0.74 amps). The Motor block can source up to 1 amp (1000 milliamps) which will work fine. We can control the power getting to the pad via pulse-width modulation as the motor throttle. Other things like direction will not have any bearing on how the pad works.

The Drive port of Crickit is good for up to 500mA, so the Motor port is a better pick here, to get the most power output.
Hook Up

A potentiometer (variable resistor) center pin is connected to Circuit Playground Express pad A7, the outside potentiometer pins connected to 3.3 V and Ground. This provides a variable voltage which can be read via the Circuit Playground Express Analog In. The programs will read this to determine how much it should drive the heating pad.

A TMP36 temperature sensor is connected to Circuit Playground Express pad A3 with power and ground to sense the temperature. If you'd like to learn more about the TMP36, see this guide for details.

The code is on the following pages in MakeCode and CircuitPython.

CircuitPython

We'll be using CircuitPython for this project. Are you new to using CircuitPython? No worries, there is a full getting started guide here.
Adafruit suggests using the Mu editor to edit your code and have an interactive REPL in CircuitPython. You can learn about Mu and its installation in this tutorial.

Get the code from GitHub, place it in Mu and save it onto the Circuit Playground Express as code.py (not hot.py!)

```python
# SPDX-FileCopyrightText: 2018 Anne Barela for Adafruit Industries
# SPDX-License-Identifier: MIT
import time
import board
from analogio import AnalogIn
from adafruit_crickit import crickit
import neopixel

print("Heating Pad Demo")

pixels = neopixel.NeoPixel(board.NEOPIXEL, 10, auto_write=False)

def show_value(heat_val):  # Show throttle on NeoPixels on CPX
    num_pixels = int(10 * (heat_val + 0.002))
    for i in range(num_pixels):
        pixels[i] = (10*(i+1), 0, 0)
    for i in range(num_pixels, 10):
        pixels[i] = (0, 0, 0)
    pixels.show()

TMP36 = AnalogIn(board.A3)  # TMP36 connected to A3, power & ground
POT = AnalogIn(board.A7)    # potentiometer connected to A7, power & ground

heating_pad = crickit.dc_motor_2  # Set the motor object to heating pad

while True:  # Loop Forever
    voltage = TMP36.value * 3.3 / 65536.0  # Read temp sensor, get voltage
    tempC = (voltage - 0.5) * 100.0        # Calculate Celsius
    tempF = (tempC * 9.0 / 5.0) + 32.0     # Calculate Fahrenheit
    heat_value = POT.value / 65536.0      # Value (0.0 to 1.0) to drive pad

    print((tempF, heat_value))  # Display temperature and drive
    show_value(heat_value)  # set Motor throttle value to heat_value
    heating_pad.throttle = heat_value  # set Motor throttle value to heat_value
    time.sleep(0.25)  # Wait a bit before checking all again
```

If you open the Mu serial terminal (press the menu bar Serial button), you’ll see the temperature measured by the temperature sensor and the value of the potentiometer from 0 to 1. At 0, the heating pad is not powered, at 1.0 it is fully powered, and values between the heating pad is pulse width modulated to be at at a temperature lower than maximum.

©Adafruit Industries
The NeoPixels show the value of the potentiometer in 1/10th increments. The more NeoPixels, the hotter the heating pad.

MakeCode

If you are new to using Microsoft MakeCode, check out our guide to getting started.

We'll need to have the Crickit extension loaded to have the Circuit Playground Express and Crickit to work together. See this guide for the instructions.

The MakeCode program below is very similar to the CircuitPython program on the previous page.
You will need to use the VARIABLE block group to make four variables names `raw_temp`, `temp_C`, `temp_F` and `heat_percent`. `raw_temp` holds the raw analog reading of the TMP36 sensor with three MATH block operations: a multiply, a divide, and a subtraction. `temp_C` is the final calculated value of the temperature sensor in degrees Celsius. `temp_F` is the sensor temperature in Fahrenheit. `heat_percent` is the value read by the potentiometer expressed as a percentage from 0 to 100% and is used to drive the motor terminal 2 containing the heating pad.

Variable `show` is used to signal the user has pushed the Circuit Playground Express A button toggling between the NeoPixels reading the potentiometer from 0 to 100% and reading the temperature sensor from 80 degrees to 120 degrees Fahrenheit.

If you run the program with the [MakeCode app in Windows 10](https://makecode.microbit.org), the console output will print off and graph the temperature and heating percent.
Results

Note that heating elements take time to either heat up or cool down. When you adjust the potentiometer, you may not detect any immediate result until it registers a temperature change.

The NeoPixels will show the percentage of the Crickit applying voltage to the heating pad. No lights = 0%, 10 lights = 100%. To try it out, turn the potentiometer all the way counter-clockwise. Let the pad cool and note the temperature - it should be room temperature. Now turn the knob clockwise, wait and record the highest temperature. Do the same all the way to 100% so you have 11 values.

You can plot this yourself or record the data in a quick spreadsheet:

![Plot of temperature data](image)

The data should be more of a line but measurements may be affected by where on the heating pad you measure and other conditions. I found it was best to measure in the middle of the pad if you can.

Use

Be sure to not have flammable material or skin on the heating pad during testing and use.
In use, the pad will be at room temperature (with the potentiometer reading 0 on the serial monitor, meaning the pad is off) to a measured 129 degrees F / 56 degrees C with the potentiometer fully on.

The center of the pad seems to gets hotter than the edges but make measurements around the pad to check for even heating.

Do not expect the pad to go from cold to hot quickly. The resistive material in the pad does not act quickly, it may take 30 seconds or so to adjust to temperature change requests.

If you want something more instantly hot or cold, the Peltier module in the next section may be what you need.

As you turn the potentiometer, the pad should warm up. If you press the Plotter button in Mu for the CircuitPython version, you should see the temperature rise on the temperature sensor as you turn the potentiometer but it will not rise fast.

These are the readings you will find with this setup:

- Pad Fully On, Crickit outputs 4.49 volts, 670 milliamps, just over 3 watts
- Bypass Crickit, pad connected directly to 5 volts: 5.12 volts, 740 milliamps, 3.79 watts

These are readings with a multimeter and non-contact thermometer. The TMP36, when used to measure the temperature, should be affixed in direct contact near the hottest place on the pad if possible.

Application

This type of setup would be very good for remote control of heating in an outside electronics enclosure where you don't want the enclosure contents to get too cold. You could monitor the temperature and vary the heating accordingly.
This would not be the best way to heat ones drink. You should plan your temperature range to fit the capabilities of the heater element.

More on the Heating Pad

There are two data sheets available:

- First one [link]
- Second one [link]

If you are planning custom projects, look at the capabilities of the pad.

If you plan to use a different heater:

The heating element should be 5 volts and not consume more than one amp of current for the Motor outputs or 500 milliamps from the drive terminals.

Make It Cold

Making things cold is usually much harder, both mechanically and physically, than heating something up due to thermodynamics. But Makers and Engineers always find a way.
In the same set of physics is a phenomenon called the Peltier Effect or the **Thermoelectric effect**. Driving electrical current through a semiconductor junction transfers heat from one side to another (thus cooling the side that has the heat taken away from it). Often times the junctions are sandwiched together to amplify the effect.

**Peltier junctions** are used for cooling small spaces, electronics, and small refrigerators. When something needs a large amount of cooling, it makes more sense power-wise to use some other cooling method often with more mechanical components like refrigeration systems.

Unlike the heating pad, Peltier modules get hot on the opposite very quickly, possibly to the point of causing burns on skin and heat marks on surfaces. Please be cautious on placement and use.

### Checking It Out

There are a number of Peltier coolers on the market which require different methods of power and handling. For this tutorial we will use the [Adafruit #1331 Peltier module](https://www.adafruit.com/product/1331). This module is powered by 5 volts. This unit has a maximum current draw of 1.6 amps. Use with Crickit is ok, as the Motor driver has current limiting to ensure the electronics are not damaged.

Using the [Peltier Thermo-Electric Cooler Module - 5V 1A](https://www.adafruit.com/product/1331) with two alligator to pin connection cables and a [female power supply connector](https://www.adafruit.com/product/1331), you can hook the Peltier directly to a 5V 2 amp power supply. The supply must be able to provide > 1.6 amps (a supply that provides more than 2 amps at 5 volts is fine). The connections are shown below.

Assemble such that the Peltier module will not damage anything while it heats up!
You'll see one side gets rather hot and one side gets cold. Cool, literally! Note which side gets hot and which side gets cold for later.

Feel free to use the Peltier module in this manner to heat or cool any project you need.

If you'd like some variability in the heating and cooling, see the next page.

**Connections**

The circuit is nearly identical to the Make It Hot project earlier in the tutorial. Note the Peltier module connects to the Motor 2 connection where the heating pad had been previously. Be sure you power the Crickit from a 5 volt 2 amp (or greater amperage) power supply.

The TMP36 temperature sensor is connected to Circuit Playground Express pad A3.

The potentiometer center terminal is connected to Circuit Playground Express pad A7.
The Peltier Module is on Motor terminal 2 with the black wire towards the speaker icon and the red wire next to it.

How to Vary the Temperature

The Crickit Motor terminals use pulse-width modulation (PWM) to quickly turn the connections off and on to vary what the device receives. This is perfect for devices like motors and resistive heating elements. But not so good here!

Peltier modules have solid state junctions that do not work well with fast on/off times like in PWM.

The modules can be turned off and on less quickly, a second or more. The CircuitPython and MakeCode programs on the following pages do exactly that, always having the "motor" throttle run either 0 (off) or 100% (full on) but switched by the program rather slowly on the order of seconds.

The potentiometer selects on/off cycle time between 0 and 10 seconds. If the time selected via the potentiometer is > 0 then the code ensures that the time is at least one second. The entire cycle time was chosen as ten seconds. At the ends, zero results in full on, fully turned, the module is off.

The NeoPixels on the Circuit Playground Express show how strong the cooling is from no pixels (off) to 5 (half), to 10 (full on) and in between.

CircuitPython

Get the code from GitHub, place it in Mu and save it onto the Circuit Playground Express as code.py (not cold.py).

```python
# SPDX-FileCopyrightText: 2018 Anne Barela for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import board
from analogio import AnalogIn
from adafruit_crickit import crickit
import neopixel

print("Peltier Module Demo")

pixels = neopixel.NeoPixel(board.NEOPIXEL, 10, auto_write=False)

def show_value(time_val):         # Show time on NeoPixels on CPX
    num_pixels = int(10-time_val)
    for i in range(num_pixels):
        pixels[i] = (10*(i+1), 0, 0)
```
for i in range(num_pixels, 10):
    pixels[i] = (0, 0, 0)
pixels.show()

TMP36 = AnalogIn(board.A3)  # TMP36 connected to A3, power & ground
POT = AnalogIn(board.A7)    # potentiometer connected to A7, power & ground
peltier = crickit.dc_motor_2  # Drive the Peltier from Motor 2 Output

while True:                   # Loop forever
    voltage = TMP36.value * 3.3 / 65536.0
    tempC = (voltage - 0.5) * 100.0
    tempF = (tempC * 9.0 / 5.0) + 32.0
    cool_value = POT.value / 6553.6  # convert 0.0 to 10.0

    # timing can be zero or can be 1 second to 10 seconds
    # between 0 and 1 is too short a time for a Peltier module
    if cool_value < 0.2:
        cool_value = 0.0
    if 1.0 > cool_value >= 0.2:
        cool_value = 1.0

    print((tempF, cool_value))  # Show in REPL
    show_value(cool_value)      # Show on NeoPixels

    # Peltier cannot be PWM - either off or on
    # Use potentiometer read to set seconds b
    if cool_value > 0:
        peltier.throttle = 0.0  # turn off
        time.sleep(cool_value)  # wait

        peltier.throttle = 1.0  # turn on
        time.sleep(10.0 - cool_value)  # wait

If the potentiometer is not fully "off " (to left) you should see some NeoPixels on telling you roughly how many seconds the Peltier module is cycled on. Zero is fully off, ten lights fully on.

Using Mu, you can plot the data and see the values in the Serial monitor. The logging is slower than in the Hot example due to the slow cycling of the Peltier module compared to the heating pad.

Be sure the TMP36 sensor is in good thermal contact with the cold side of the module.
MakeCode

If you are new to using Microsoft MakeCode, check out our guide to getting started.

We'll need to have the Crickit extension loaded to have the Circuit Playground Express and Crickit to work together. See this guide for the instructions.

The code below is the MakeCode for driving the Peltier Module. It's got a lot going on, let's break it down. The Temperature sensor and the potentiometer are connected to analog inputs on the Circuit Playground Express. The first 4 `set` blocks convert the analog voltage read (from 0 to 1023) to the temperature and a number of seconds from 0 to 10 for timing the Peltier cycle.

Note: In CircuitPython the number of possible values from analog reads are 0 to 65532, in MakeCode only 0 to 1023, in case you are comparing the code.
If the Peltier `cool_time` is less than a second, we ensure it's either zero or one as we don't want to cycle the module less than a second at a time as it does not like fast cycle times.

The values are printed out in the Windows 10 app if running via `console log` blocks.

The next if block will stop the Peltier for `cool_time` seconds and run it for `(10 - cool_time)` seconds. As the pause block is in milliseconds, we multiply the number of seconds by 1000.

The function `Display` shows the number of seconds the Peltier is running off. If Button A is pressed, the display is switched to temperature from 32 to 82 degrees with each light 5 degree steps. Pressing Button A again switches back to the seconds dialed in by the Potentiometer.
The module will get colder at the far left (zero). Once the potentiometer is turned, the Crickit will cycle the power off then on in a ten second cycle. This is why the NeoPixels do not react immediately when you turn the knob, the previous off/on cycle must complete.

The Peltier will not get as cold as it did when it was directly connected to the power. The current is limited to one amp by Crickit so the unit will not draw all the current it wants. So if you want very cold, directly connect it to power.

If you want an environment that you want to measure the temperature (with the TMP36 sensor) and adjust the temperature, this setup will work for you.

If using a Peltier device to cool something off, it is best to use a heat sink or fan on the hot side to draw the heat off.

Heat Sinks

A heat sink is usually a piece of metal, often with metal fins, used to dissipate heat into the air. They are used on a large number of heat-generating electronics. Most likely your computer has several, the main one for the CPU and others for the power supply system. Adafruit actually sells a 12 volt Peltier module with a heat sink
attached although for this "Make It" guide we are limited by the Crickit for control which is 5 volts.

Placing the Peltier module's hot side on metal will help dissipate heat. If you plan to use a metal project enclosure, using the enclosure as a heat sink may be possible.

If you have access to scrounged electronics, you can often remove heat sinks from circuit boards. Heat sinks are available at many electronics stores also, both online and brick & mortar stores.

I found an old large chip heat sink in the junk drawer, but feel free to take any metal (usually heavier is better) to place on the hot side of the module to draw the heat away as we're looking to cool things.

A small fan also works, even better both a heat sink and a fan.

To get a good electronic contact between a heat sink and the cooler, consider some thermal tape, paste or thermal grease to place between the cooler and the heat sink. The material will allow for a good thermal contact between the two surfaces.

Heat Sink Thermal Tape - 3M 8810 - 80mm x 80mm
Get that heat sink stuck on good and strong with 3M's thermal tape. Specifically designed for attaching heat sinks to ICs. Works better and is less messy than thermal glues or...

https://www.adafruit.com/product/1468