



# Smart Mirror with PyPortal

Created by Ruiz Brothers



<https://learn.adafruit.com/smart-mirror-with-pyportal>

Last updated on 2024-06-03 03:38:56 PM EDT

# Table of Contents

<b>Overview</b>	<b>5</b>
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Mirror Weather Display</a></li><li>• <a href="#">Interactive IoT</a></li><li>• <a href="#">Parts</a></li></ul>	
<b>CAD Files</b>	<b>6</b>
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">CAD Assembly</a></li><li>• <a href="#">CAD Parts List</a></li><li>• <a href="#">Build Volume</a></li><li>• <a href="#">Acrylic Template</a></li><li>• <a href="#">Design Source Files</a></li></ul>	
<b>Install CircuitPython</b>	<b>8</b>
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Set up CircuitPython Quick Start!</a></li><li>• <a href="#">PyPortal Default Files</a></li></ul>	
<b>Coding the Smart Mirror</b>	<b>11</b>
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Upload the Code and Libraries to the PyPortal</a></li><li>• <a href="#">secrets.py</a></li><li>• <a href="#">How the CircuitPython Code Works</a></li><li>• <a href="#">Setup the PyPortal</a></li><li>• <a href="#">Setup the Light Sensor</a></li><li>• <a href="#">The Loop</a></li><li>• <a href="#">Read the Light Sensor</a></li><li>• <a href="#">Turn on the PyPortal's Screen</a></li><li>• <a href="#">Reset Clocks</a></li></ul>	
<b>Acrylic Setup</b>	<b>18</b>
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Acrylic Panel</a></li><li>• <a href="#">Mirror Film</a></li><li>• <a href="#">Cut Roll of Film</a></li><li>• <a href="#">Trim Pieces of Film</a></li><li>• <a href="#">Pieces of Film</a></li><li>• <a href="#">Serving Tray and Gloves</a></li><li>• <a href="#">Peel Corners with Tape</a></li><li>• <a href="#">Peel Acrylic Backings</a></li><li>• <a href="#">Wet Acrylic</a></li><li>• <a href="#">Wet Film</a></li><li>• <a href="#">Place Film onto Acrylic</a></li><li>• <a href="#">Squeegee Wet Film</a></li><li>• <a href="#">Soak Up Water</a></li><li>• <a href="#">Squeegee Film Dry</a></li><li>• <a href="#">Dry Acrylic Film</a></li><li>• <a href="#">Trim Excess Film</a></li><li>• <a href="#">Rinse and Repeat!</a></li></ul>	
<b>Case Assembly</b>	<b>23</b>
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Case Hardware</a></li><li>• <a href="#">Install Stand</a></li><li>• <a href="#">Wipe Clean</a></li><li>• <a href="#">Install Acrylic</a></li><li>• <a href="#">Prep PyPortal for Mirror</a></li></ul>	

- [Install PyPortal](#)
- [Secure PyPortal](#)
- [Install Cover](#)
- [USB Power](#)



---

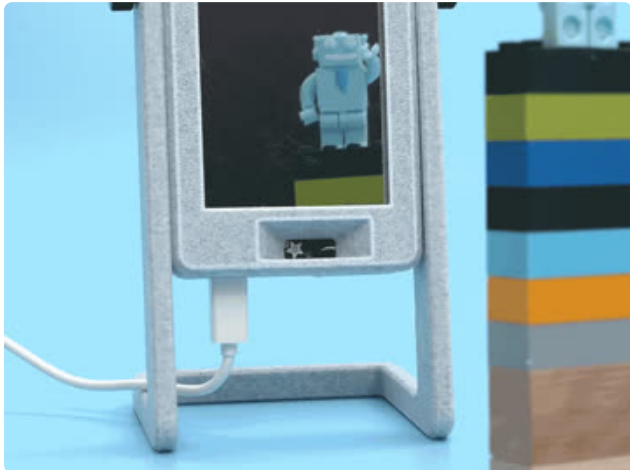
# Overview



## Mirror Weather Display

This project uses the ESP32 WiFi module to pull weather data from the open weather maps API to display temperature, weather conditions and the local time.

A piece of laser cut acrylic features a 2-way mirrored film so you can see the display through the reflections.



## Interactive IoT

Wave your hand (or paw) in front of the PyPortal light sensor to display weather conditions, temperature and local time magically inside a mirror!

The display automatically turns off after a few seconds, turning it back into a desktop mirror, the effect is whimsical!

## Parts



### [Adafruit PyPortal - CircuitPython Powered Internet Display](https://www.adafruit.com/product/4116)

PyPortal, our easy-to-use IoT device that allows you to create all the things for the “Internet of Things” in minutes. Make custom touch screen interface...

<https://www.adafruit.com/product/4116>



### Fully Reversible Pink/Purple USB A to micro B Cable - 1m long

This cable is not only super-fashionable, with a woven pink and purple Blinka-like pattern, it's also fully reversible! That's right, you will save seconds a day by...

<https://www.adafruit.com/product/4111>

---

### 4 x M3 x 6mm Screws

Button Head Hex Drive Screw

<https://www.mcmaster.com/92095A179/>

---

### 2 x M3 x 7mm Thumbscrews

Aluminum Alloy Knurled Thumb Screws Black 10pcs

[https://www.amazon.com/gp/product/B01MZYTRM3/ref=ppx\\_yo\\_dt\\_b\\_search\\_asin\\_title?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B01MZYTRM3/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1)

---

### 2 x M3 Locknut with Nylon Insert

Steel Nylon-Insert Locknut

<https://www.mcmaster.com/90576A102/>

---

### 1 x Acrylic Sheet

1/8in thick acrylic sheet for laser or CNC

<https://www.inventables.com/technologies/clear-acrylic-sheet>

---

### 1 x Roll of Mirror Film

BDF S05 One Way Mirror Film

<https://www.amazon.com/gp/product/B00FQPGH8I/>

---

## CAD Files



### CAD Assembly

The piece of acrylic snap fits into a recess inside the case. The PyPortal PCB is secured to the case using machine screws. The mirror stand attaches to the case using thumb screws and lock nuts. The back cover snap fits over the case.



## CAD Parts List

STL files for 3D printing are oriented to print "as-is" on FDM style machines. Parts are designed to 3D print without any support material. Original design source may be downloaded using the links below:

Mirror-Case.stl

Mirror-Cover.stl

Mirror-Stand.stl

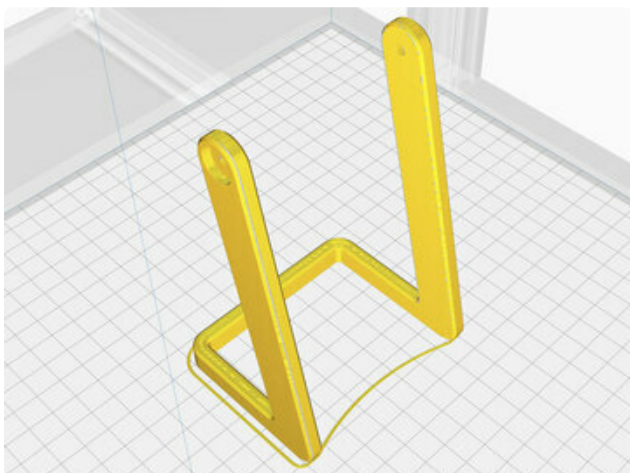
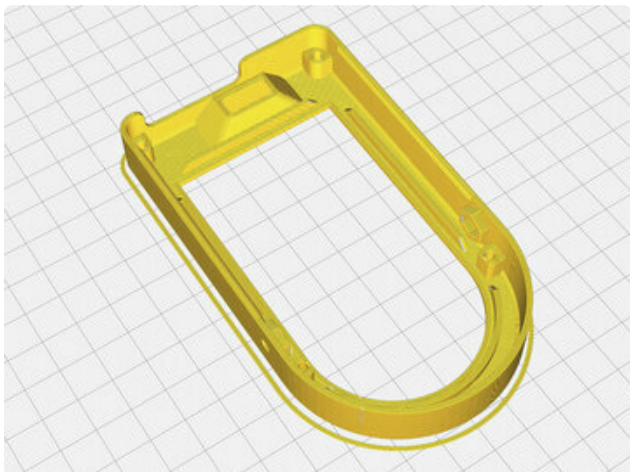
Mirror-Acrylic.svg

Download STL files

<https://adafru.it/-F8>

Download CAD source

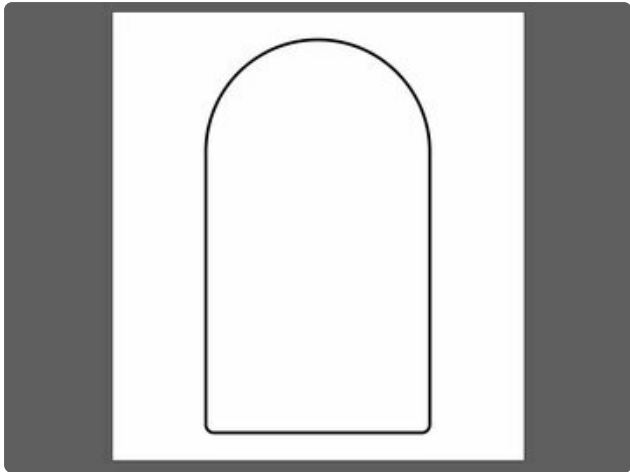
<https://adafru.it/-F9>



## Build Volume

The parts require a 3D printer with a minimum build volume.

122mm (X) x 122mm (Y) x 132mm (Z)



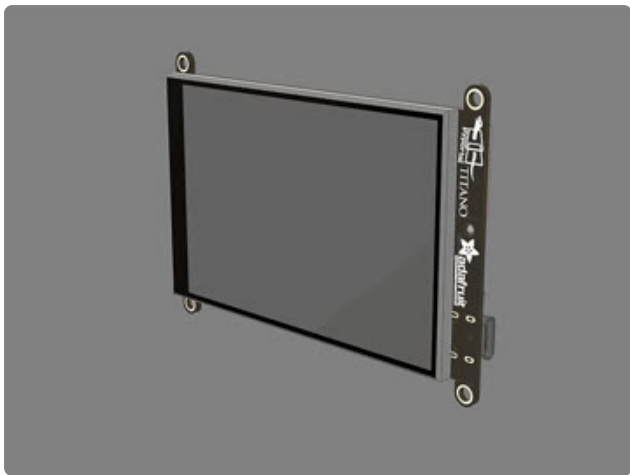
## Acrylic Template

Use the DXF or SVG file to laser cut or CNC mill the 1/8in(3.17mm) thick acrylic piece to make the mirror.

Acrylic Dimensions: 96.5mm x 55mm

[acrylic-template.zip](#)

<https://adafru.it/-Fa>



## Design Source Files

The project assembly was designed in Fusion 360. This can be downloaded in different formats like STEP, STL and more. Electronic components like Adafruit's board, displays, connectors and more can be downloaded from the [Adafruit CAD parts GitHub Repo](#) (<https://adafru.it/AW8>).

---

# Install CircuitPython

[CircuitPython](https://adafru.it/tB7) (<https://adafru.it/tB7>) is a derivative of [MicroPython](https://adafru.it/BeZ) (<https://adafru.it/BeZ>) designed to simplify experimentation and education on low-cost microcontrollers. It makes it easier than ever to get prototyping by requiring no upfront desktop software downloads. Simply copy and edit files on the **CIRCUITPY** "flash" drive to iterate.

The following instructions will show you how to install CircuitPython. If you've already installed CircuitPython but are looking to update it or reinstall it, the same steps work for that as well!

## Set up CircuitPython Quick Start!

Follow this quick step-by-step for super-fast Python power :)



Download the latest version of  
CircuitPython for the PyPortal via  
CircuitPython.org

<https://adafru.it/Egk>

Download the latest version of  
CircuitPython for the PyPortal Pynt  
via CircuitPython.org

<https://adafru.it/HFd>

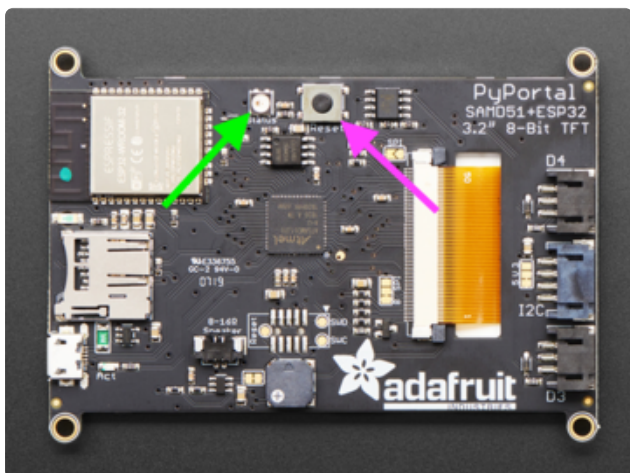


Click the link above to download the  
latest version of CircuitPython for the  
PyPortal.

Download and save it to your desktop (or  
wherever is handy).

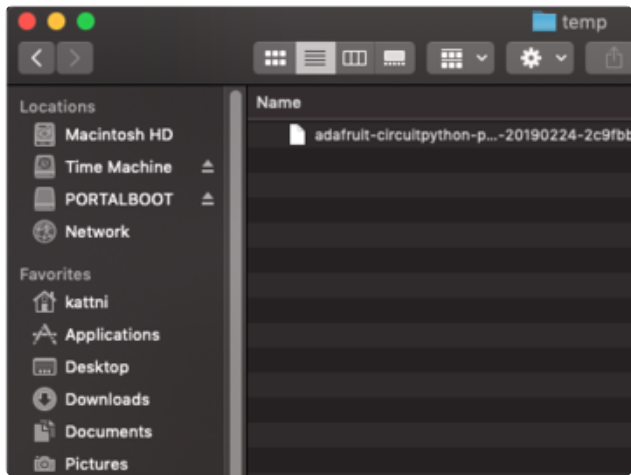
Plug your PyPortal into your computer  
using a known-good USB cable.

A lot of people end up using charge-only  
USB cables and it is very frustrating! So  
make sure you have a USB cable you  
know is good for data sync.

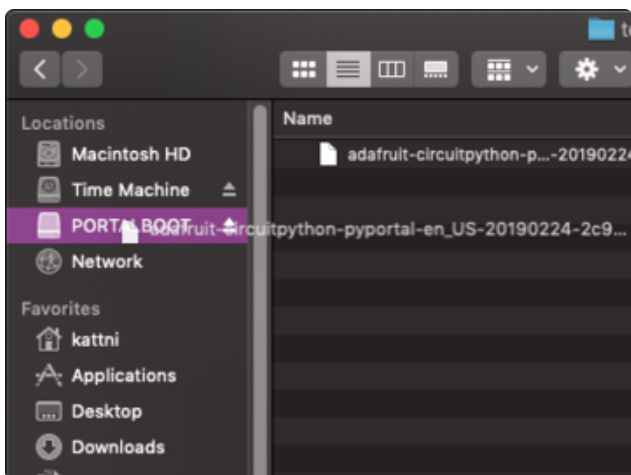


Double-click the **Reset** button on the top in  
the middle (magenta arrow) on your board,  
and you will see the NeoPixel RGB LED  
(green arrow) turn green. If it turns red,  
check the USB cable, try another USB port,  
etc. **Note:** The little red LED next to the  
USB connector will pulse red. That's ok!

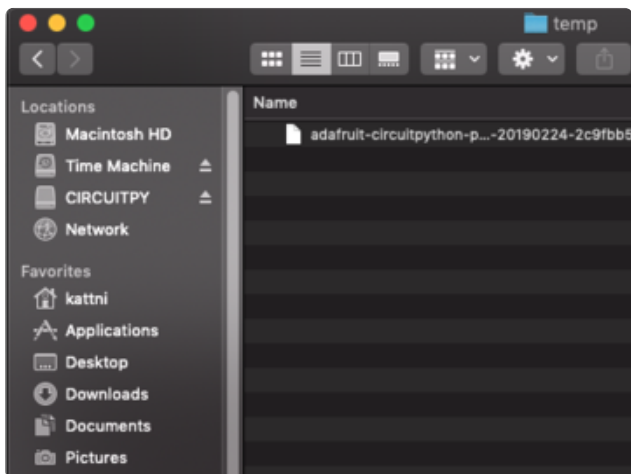
If double-clicking doesn't work the first  
time, try again. Sometimes it can take a  
few tries to get the rhythm right!



You will see a new disk drive appear called **PORTALBOOT**.



Drag the **adafruit-circuitpython-pyportal-<whatever>.uf2** file to **PORTALBOOT**.



The LED will flash. Then, the **PORTALBOOT** drive will disappear and a new disk drive called **CIRCUITPY** will appear.

If you haven't added any code to your board, the only file that will be present is **boot\_out.txt**. This is absolutely normal! It's time for you to add your **code.py** and get started!

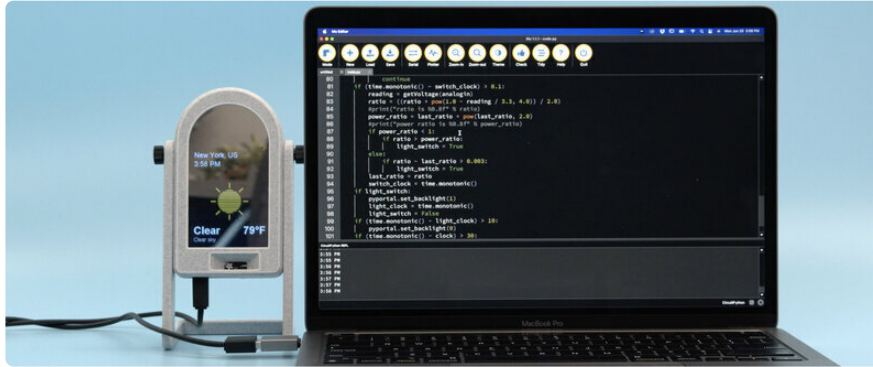
That's it, you're done! :)

## PyPortal Default Files

Click below to download a zip of the files that shipped on the PyPortal or PyPortal PynT.

[PyPortal Default Files](#)

## Coding the Smart Mirror



Once you've finished setting up your PyPortal with CircuitPython, you can access the code and necessary libraries by downloading the Project Bundle.

To do this, click on the **Download Project Bundle** button in the window below. It will download as a zipped folder.

```
# SPDX-FileCopyrightText: 2022 Liz Clark for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import sys
import time
import board
from analogio import AnalogIn
from adafruit_pyportal import PyPortal
cwd = ("/" + __file__).rsplit('/', 1)[0] # the current working directory (where this
file is)
sys.path.append(cwd)
import openweather_graphics # pylint: disable=wrong-import-position

# Get wifi details and more from a secrets.py file
try:
    from secrets import secrets
except ImportError:
    print("WiFi secrets are kept in secrets.py, please add them there!")
    raise

# Use cityname, country code where countrycode is ISO3166 format.
# E.g. "New York, US" or "London, GB"
LOCATION = "New York, US"

# Set up where we'll be fetching data from
DATA_SOURCE = "http://api.openweathermap.org/data/2.5/weather?q="+LOCATION
DATA_SOURCE += "&appid="+secrets['openweather_token']
# You'll need to get a token from openweather.org, looks like
# 'b6907d289e10d714a6e88b30761fae22'
DATA_LOCATION = []
```

```

# Initialize the pyportal object and let us know what data to fetch and where
# to display it
pyportal = PyPortal(url=DATA_SOURCE,
                    json_path=DATA_LOCATION,
                    status_neopixel=board.NEOPIXEL,
                    default_bg=0x000000)

display = board.DISPLAY
# rotate display for portrait orientation
display.rotation = 270

# instantiate the openweather_graphics class
gfx = openweather_graphics.OpenWeather_Graphics(pyportal.splash, am_pm=True,
celsius=False)

# time keeping for refreshing screen icons and weather information
localtile_refresh = None
weather_refresh = None

# setup light sensor as an analog input
analogin = AnalogIn(board.LIGHT)

# analog scaling helper
def getVoltage(pin):
    return (pin.value * 3.3) / 65536

# timer for updating onscreen clock
clock = time.monotonic()
# timer for keeping backlight on
light_clock = time.monotonic()
# timer for checking the light sensor
switch_clock = time.monotonic()
# variable to scale light sensor reading
ratio = 0
# storing last light sensor ratio reading
last_ratio = 0

while True:
    # only query the online time once per hour (and on first run)
    if (not localtile_refresh) or (time.monotonic() - localtile_refresh) > 3600:
        try:
            print("Getting time from internet!")
            pyportal.get_local_time()
            localtile_refresh = time.monotonic()
        except RuntimeError as e:
            print("Some error occurred, retrying! -", e)
            continue

    # only query the weather every 10 minutes (and on first run)
    if (not weather_refresh) or (time.monotonic() - weather_refresh) > 600:
        try:
            value = pyportal.fetch()
            print("Response is", value)
            gfx.display_weather(value)
            weather_refresh = time.monotonic()
        except RuntimeError as e:
            print("Some error occurred, retrying! -", e)
            continue

    # every 0.1 seconds check the light sensor value
    if (time.monotonic() - switch_clock) > 0.1:
        # read the light sensor and scale it 0 to 3.3
        reading = getVoltage(analogin)
        # calculate the % of light out of the maximum light
        ratio = ((ratio + pow(1.0 - reading / 3.3, 4.0)) / 2.0)
        # create a comparison ratio with the last reading
        power_ratio = last_ratio + pow(last_ratio, 2.0)
        # if the comparison ratio is less than 1
        if power_ratio < 1:

```

```

        # and the current ratio is larger
        if ratio > power_ratio:
            # turn on the backlight
            pyportal.set_backlight(1)
            light_clock = time.monotonic()
        # otherwise (if in a darker room)
        else:
            # if there's a difference greater than 0.003
            # between the current ratio and the last ratio
            if ratio - last_ratio > 0.003:
                # turn on the backlight
                pyportal.set_backlight(1)
                light_clock = time.monotonic()
            # update last_ratio
            last_ratio = ratio
            switch_clock = time.monotonic()
        # after 10 seconds, turn off the backlight
        if (time.monotonic() - light_clock) > 10:
            pyportal.set_backlight(0)
        # every 30 seconds update time on screen
        if (time.monotonic() - clock) > 30:
            gfx.update_time()
            clock = time.monotonic()

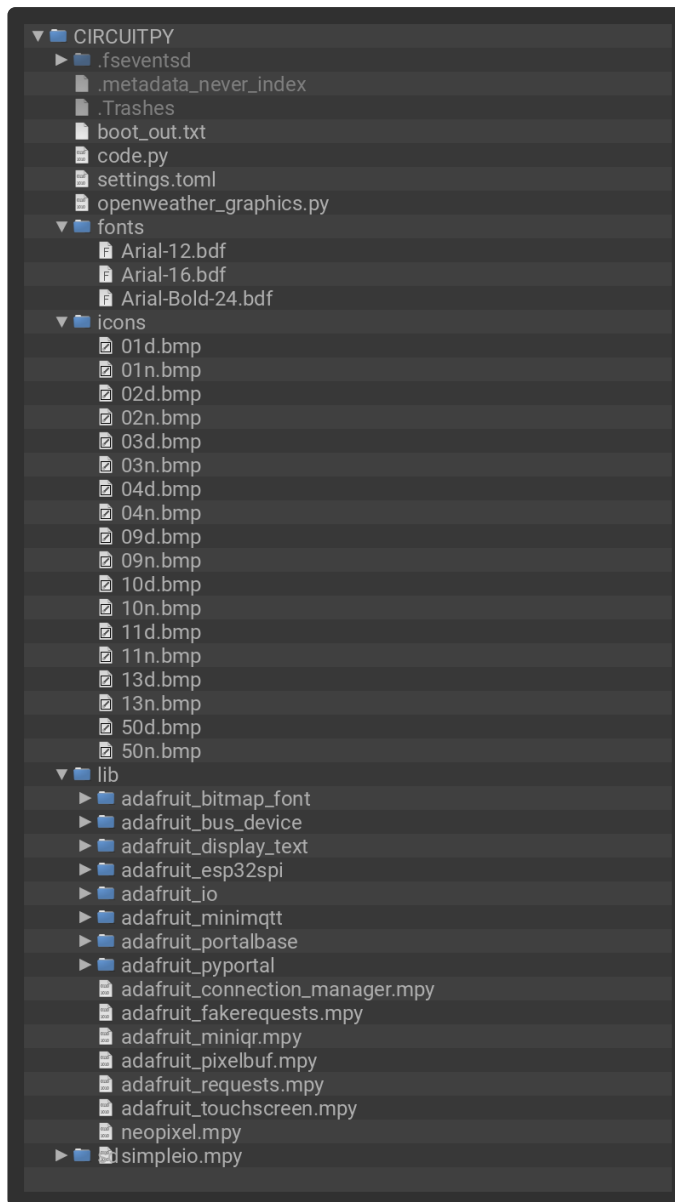
```

## Upload the Code and Libraries to the PyPortal

After downloading the Project Bundle, plug your PyPortal into the computer's USB port with a known good USB data+power cable. You should see a new flash drive appear in the computer's File Explorer or Finder (depending on your operating system) called **CIRCUITPY**. Unzip the folder and copy the following items to the PyPortal's **CIRCUITPY** drive.

- **lib** folder
- **fonts** folder
- **icons** folder
- **openweather\_graphics.py**
- **code.py**

Your PyPortal **CIRCUITPY** drive should look like this after copying the **lib** folder, **fonts** folder, **icons** folder, **openweather\_graphics.py** file and the **code.py** file.



## secrets.py

You will need to create and add a **secrets.py** file to your **CIRCUITPY** drive. Your **secrets.py** file will need to include the following information:

```
secrets = {
    'ssid' : 'YOUR-SSID-HERE',
    'password' : 'YOUR-SSID-PASSWORD-HERE',
    'aio_username' : 'YOUR-AIO-USERNAME-HERE',
    'aio_key' : 'YOUR-AIO-KEY-HERE',
    'openweather_token' : 'YOUR-OPENWEATHER-TOKEN-HERE',
    'timezone' : "America/New_York", # http://worldtimeapi.org/timezones
}
```

We'll be using OpenWeatherMaps.org to retrieve the weather info through its API. In order to do so, you'll need to register for an account and get your API key.

Go to this [link \(https://adafru.it/EeH\)](https://adafru.it/EeH) and register for a free account. Once registered, you'll get an email containing your API key, also known as the "openweather token".

You will also need your Adafruit IO username and key. Check out [this guide for getting started with Adafruit IO \(https://adafru.it/Ef8\)](https://adafru.it/Ef8) if you haven't used Adafruit IO before.

## How the CircuitPython Code Works

This code was adapted from the [PyPortal Weather Station project \(https://adafru.it/ICZ\)](https://adafru.it/ICZ). The code has been modified to rotate the display so that it is vertical and to turn on the display's backlight when it senses movement in front of the light sensor.

You'll want to update `LOCATION` at the top of the code to the location that you want to receive data for. Weather data is retrieved via the [OpenWeatherMap API \(https://adafru.it/KeU\)](https://adafru.it/KeU).

```
# Use cityname, country code where countrycode is ISO3166 format.
# E.g. "New York, US" or "London, GB"
LOCATION = "New York, US"

# Set up where we'll be fetching data from
DATA_SOURCE = "http://api.openweathermap.org/data/2.5/weather?q="+LOCATION
DATA_SOURCE += "&appid="+secrets['openweather_token']
# You'll need to get a token from openweathermap.org, looks like
# 'b6907d289e10d714a6e88b30761fae22'
DATA_LOCATION = []
```

## Setup the PyPortal

The display is rotated using `display.rotation`. `OpenWeather_Graphics()` is a helper class from the `openweather_graphics.py` file. That file takes care of the graphics on screen and parsing the OpenWeatherMaps JSON feed to display the weather data.

```
# Initialize the pyportal object and let us know what data to fetch and where
# to display it
pyportal = PyPortal(url=DATA_SOURCE,
                    json_path=DATA_LOCATION,
                    status_neopixel=board.NEOPIXEL,
                    default_bg=0x000000)

display = board.DISPLAY
# rotate display for portrait orientation
display.rotation = 270

# instantiate the openweather_graphics class
gfx = openweather_graphics.OpenWeather_Graphics(pyportal.splash, am_pm=True,
celsius=False)
```

## Setup the Light Sensor

The light sensor is accessed with `board.LIGHT`. The `getVoltage()` helper is used to scale the analog input from `0` to `3.3`.

```
# setup light sensor as an analog input
analogin = AnalogIn(board.LIGHT)

# analog scaling helper
def getVoltage(pin):
    return (pin.value * 3.3) / 65536
```

## The Loop

In the loop, `pyportal.get_local_time()` is called every hour to get the internet time from Adafruit IO. The weather data feed is updated every 10 minutes with `gfx.display_weather(value)`.

```
while True:
    # only query the online time once per hour (and on first run)
    if (not localtime_refresh) or (time.monotonic() - localtime_refresh) > 3600:
        try:
            print("Getting time from internet!")
            pyportal.get_local_time()
            localtime_refresh = time.monotonic()
        except RuntimeError as e:
            print("Some error occurred, retrying! -", e)
            continue

    # only query the weather every 10 minutes (and on first run)
    if (not weather_refresh) or (time.monotonic() - weather_refresh) > 600:
        try:
            value = pyportal.fetch()
            print("Response is", value)
            gfx.display_weather(value)
            weather_refresh = time.monotonic()
        except RuntimeError as e:
            print("Some error occurred, retrying! -", e)
            continue
```

## Read the Light Sensor

Every 0.1 seconds, the light sensor's value is read. The reading is scaled with `getVoltage()`. The code calculates what percentage out of the maximum amount of light is being read. This value is held in `ratio`. `power_ratio` holds the value of `last_ratio` plus `last_ratio` to the 2nd power.

`power_ratio` is used to compare the current `ratio` against to see if a substantial change in light value is being read by the sensor. This allows for the project to work in rooms with various levels of light.



```
# every 0.1 seconds check the light sensor value
if (time.monotonic() - switch_clock) > 0.1:
    # read the light sensor and scale it 0 to 3.3
    reading = getVoltage(analogin)
    # calculate the % of light out of the maximum light
    ratio = ((ratio + pow(1.0 - reading / 3.3, 4.0)) / 2.0)
    # create a comparison ratio with the last reading
    power_ratio = last_ratio + pow(last_ratio, 2.0)
```

## Turn on the PyPortal's Screen

If the `ratio` is larger than `power_ratio` when `power_ratio` is less than `1`, then the PyPortal's backlight is turned on.

```
# if the comparison ratio is less than 1
if power_ratio < 1:
    # and the current ratio is larger
    if ratio > power_ratio:
        # turn on the backlight
        pyportal.set_backlight(1)
        light_clock = time.monotonic()
```

If `power_ratio` is less than one, then the PyPortal's backlight is turned on when the difference between `ratio` and `last_ratio` is greater than `0.003`.

```
# otherwise (if in a darker room)
else:
    # if there's a difference greater than 0.003
    # between the current ratio and the last ratio
    if ratio - last_ratio > 0.003:
        # turn on the backlight
        pyportal.set_backlight(1)
        light_clock = time.monotonic()
```

## Reset Clocks

After `10` seconds, the PyPortal's backlight is turned off. Every `30` seconds, the clock time on screen is updated with `gfx.update_time()`.

```
# after 10 seconds, turn off the backlight
if (time.monotonic() - light_clock) > 10:
    pyportal.set_backlight(0)
# every 30 seconds update time on screen
if (time.monotonic() - clock) > 30:
    gfx.update_time()
    clock = time.monotonic()
```

---

## Acrylic Setup



### Acrylic Panel

The acrylic can be laser cut or cnc milled using the svg or dxf file. The enclosure was designed for a specific size of acrylic.

96.5mm(3.8in) x 55mm (2.17in)  
3.17mm (1/8in) Thick



### Mirror Film

This one-way mirror film comes in a roll in different sizes. It comes with a box cutter and a squeegee tool for applying the film. I used the following roll size:

12in x 24ft



### Cut Roll of Film

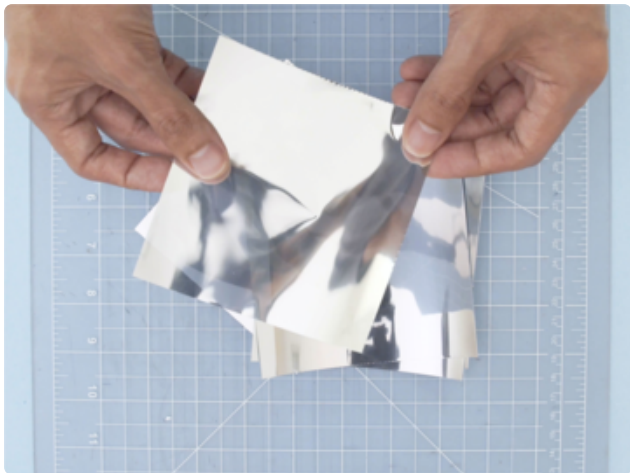
Start by laying out a sheet from the roll onto a cutting mat. I used a T-square ruler to guide the blade while cutting a sheet from the roll.

Take extreme caution when using a sharp tool!



## Trim Pieces of Film

I placed the acrylic the sheet to gauge how large the piece needs to be. You'll want to add a bit of extra space around the edges so the acrylic has total coverage.



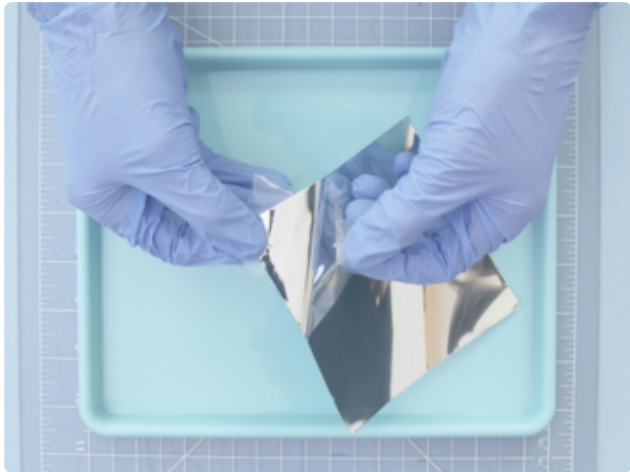
## Pieces of Film

I suggest cutting out several pieces. You'll want to have extra pieces in case one of them gets messed up while applying.



## Serving Tray and Gloves

I suggest using a serving tray to catch the soapy water so your work surface doesn't get wet. Gloves will help to prevent getting smudges and fingerprints on the acrylic or film.



## Peel Corners with Tape

The film has a protective backing that needs to be removed. Adding pieces of tape to the corners can help you peel apart the two layers.



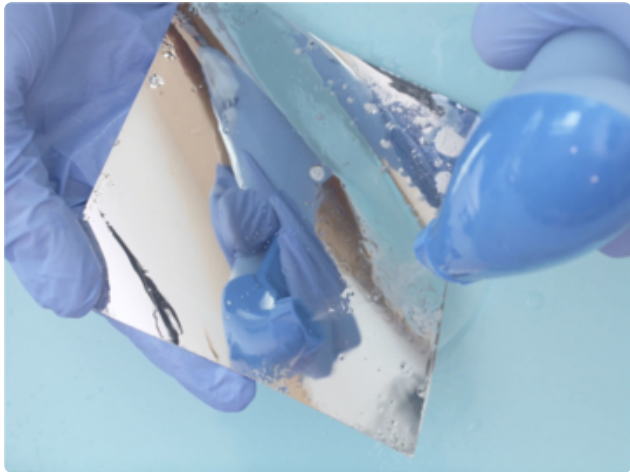
## Peel Acrylic Backings

Laser cut acrylic can often arrive with protective backings that will need to be removed by carefully peeling them off.



## Wet Acrylic

The roll of film has instructions for applying the film to windows (acrylic in our case). The directions say to use soapy water. I filled an empty spray bottle with **16oz** of water and **6 drops** of liquid hand soap. Spray the acrylic so the surface is has been fully wet.



## Wet Film

The sticky side of the film also needs to be wet, so spray it down and make sure you get total coverage. The trick is to use lots of soapy water so the two surfaces can adhere smoothly.



## Place Film onto Acrylic

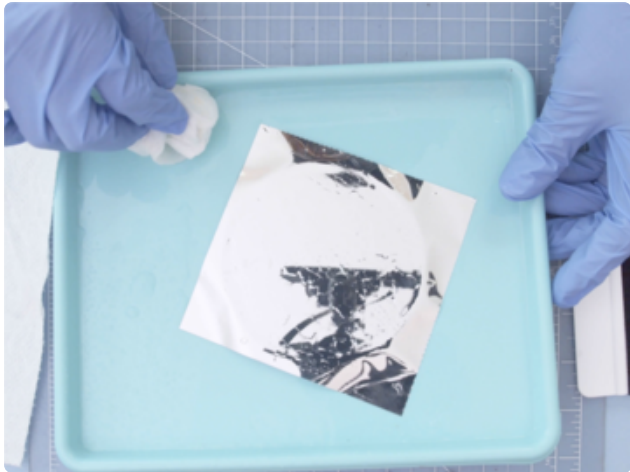
Place the film over the acrylic and try to get it in the center. Lay the film on top and press down. You can shift it into place if you need to.



## Squeegee Wet Film

I used the squeegee that came with the film and rinsed out all of the water and air bubbles.





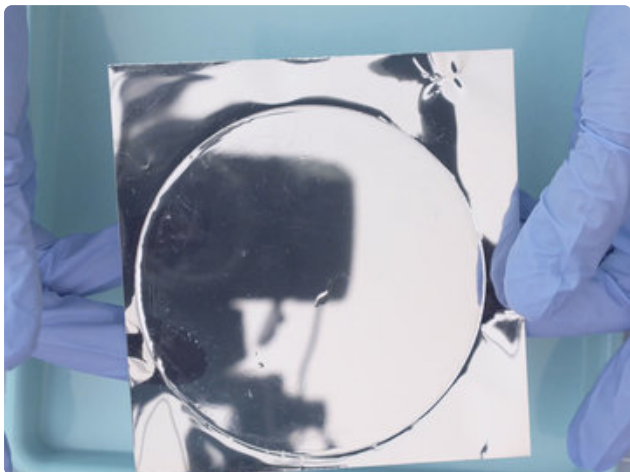
## Soak Up Water

The serving tray should catch most of the water. I made sure to use plenty of paper towels to soak it all up.



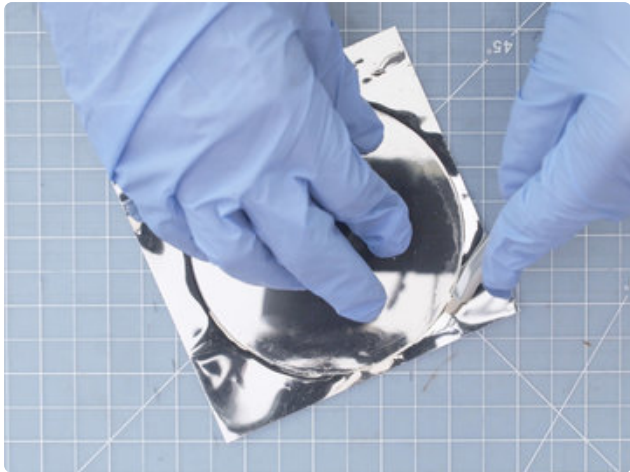
## Squeegee Film Dry

Squeegee the film until all of the soapy water has either been wiped away or dried out. You'll want to start from the center and wipe out towards to the edges.



## Dry Acrylic Film

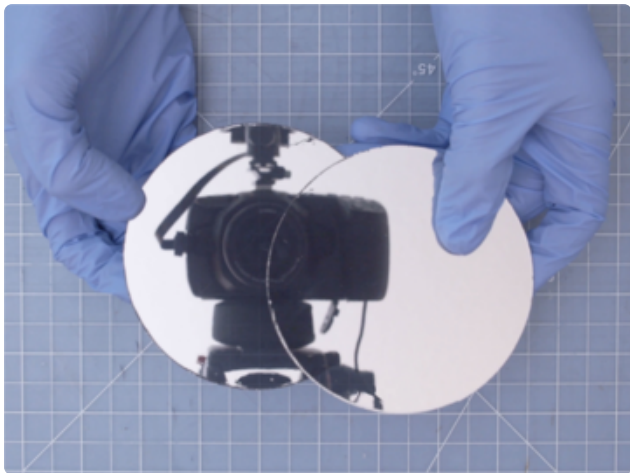
Once the application looks good, make sure to let the film dry several minutes before trimming the excess. The edges take a bit of time to fully dry.



## Trim Excess Film

I flipped it over and used a box cutter to trim along the edge on the outside. You'll want to be super careful not to scratch up the surface while doing this.

The acrylic piece only needs one side of the mirrored film.



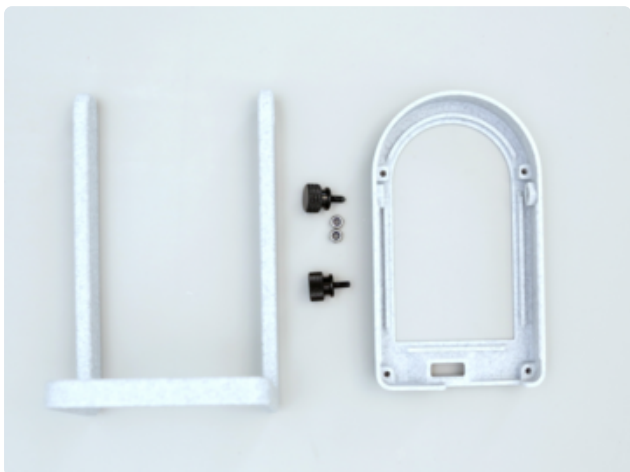
## Rinse and Repeat!

This took me several times to get right, so it takes a bit of effort to get a nice finish. With some practice and patience, we were able to get these two pieces.

If things don't look as good as you'd like, you can always peel off the film and try again!

---

## Case Assembly



## Case Hardware

Use the following hardware to secure the mirror stand to the case.

2x M3x7mm Thumbscrews

2x M3 Locknuts with nylon insert



## Install Stand

Place the stand over the case and line up the mounting holes on the sides. Insert an M3 locknut into the hexagonal recess inside the case. Insert and fasten an M3 thumbscrew through the hole on the side of the arm to secure the parts together. Repeat this process for the other side.



## Wipe Clean

Before installing the PyPortal, wipe away any smudges from the acrylic. Use alcohol and a paper towel or a lens cleaning wipe.





## Install Acrylic

Get the piece of acrylic ready to install into the case.

Handle acrylic only by the edges to avoid smudging the surface.

Orient the acrylic with the case.

Place the piece of acrylic into the recess inside the case. Press the piece of acrylic into the recess.



## Prep PyPortal for Mirror

A piece of black tape is used to block light from the display from leaking into the mirror.

Place a strip of gaffers tape over the top of the PyPortal so it wraps over the edge.



## Install PyPortal

Orient the PyPortal in the case so that the microUSB port is pointing down and the edge with the tape is pointing up.

Place the PyPortal into the case and line up the mounting holes.



## Secure PyPortal

Insert and fasten 4x M3x6mm screws into the mounting tabs on the PyPortal to secure to the case.

Do not over tighten the screws!





## Install Cover

Get cover ready to install onto the case.

Orient the back cover with the case.

Press fit the case into the case so the edges snap fit together.



## USB Power

Plug in a micro USB cable into the pyportal and power it using a 5V power supply or USB battery bank.

Congratulations on building your mini small mirror with PyPortal!