



# Purple Air AQI Display

Created by John Park



<https://learn.adafruit.com/purple-air-aqi-display>

Last updated on 2024-06-03 03:13:09 PM EDT

# Table of Contents

Overview	3
• Parts	
Using MatrixPortal	6
Prep the MatrixPortal	8
• Power Prep	
• Power Terminals	
• Panel Power	
• Dual Matrix Setup	
• Board Connection	
Install CircuitPython	12
• Set up CircuitPython Quick Start!	
• Further Information	
Using M4 Airlift	14
Install CircuitPython	15
• Set up CircuitPython Quick Start!	
• Further Information	
Build Metro M4 Airlift Matrix Display	18
• Assembly	
• Shields Up	
• Power Connections	
• Data Cable	
• Wall Adapter	
Code the Purple Air Display	21
• Libraries	
• Connect to the Internet	
• Text Editor	
• Code	
• Purple Air	
• API Key	
• How it Works	
• Matrix Setup	
• Main Loop	
LED Matrix Diffuser	27
• LED Diffusion Acrylic	
• Measure and Cut the Plastic	
• Uglu Dashes	
• Stand	

---

# Overview



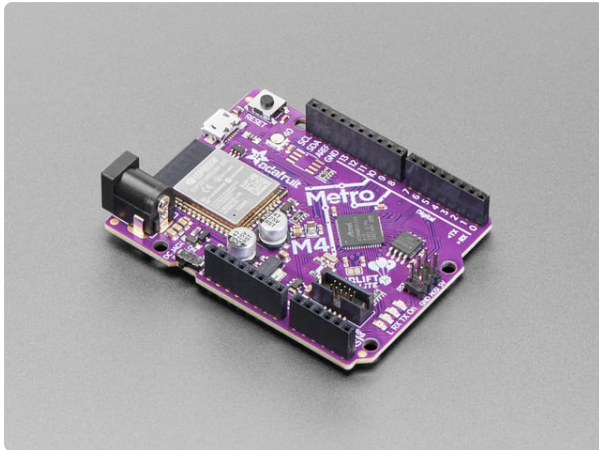
The quality of the air we breathe can be very important, particularly right now with recent wildfires on the west coast of the US.

This project shows how to display the Air Quality Index (AQI) of a based on the Purple Air real-time monitoring community's outdoor sensors sensor right on your own LED Matrix display.

Your readings will be updated automatically over WiFi, and you can plug in the sensor ID of any Purple Air sensor in the world!

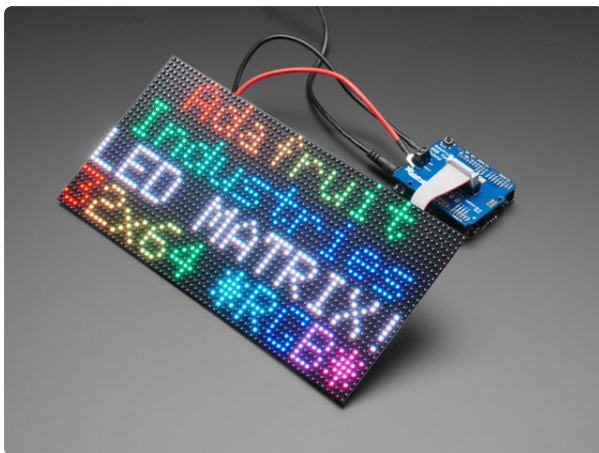
We'll show you how to build this project with either the Matrix Portal or a Metro M4 Express AirLift with RGB Matrix shield.

## Parts



### [Adafruit Metro M4 Express AirLift \(WiFi\) - Lite](https://www.adafruit.com/product/4000)

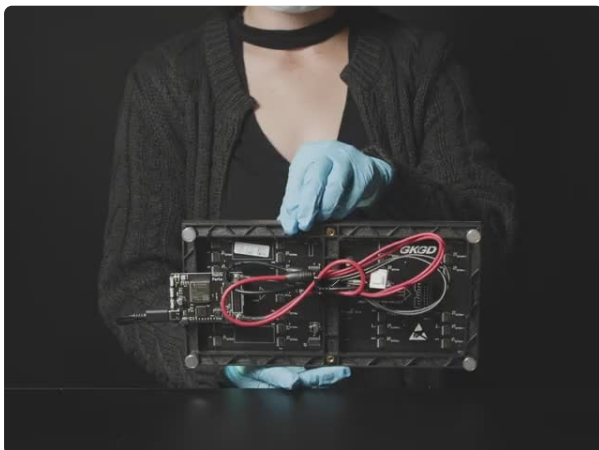
Give your next project a lift with AirLift - our witty name for the ESP32 co-processor that graces this Metro M4. You already know about the Adafruit Metro...  
<https://www.adafruit.com/product/4000>



### [Adafruit RGB Matrix Shield for Arduino](https://www.adafruit.com/product/2601)

Our RGB matrices are dazzling, with their hundreds or even thousands of individual RGB LEDs. Compared to NeoPixels, they've got great density, power usage and the...  
<https://www.adafruit.com/product/2601>

or



### [Adafruit Matrix Portal - CircuitPython Powered Internet Display](https://www.adafruit.com/product/4745)

Folks love our wide selection of RGB matrices and accessories, for making custom colorful LED displays... and our RGB Matrix Shields...  
<https://www.adafruit.com/product/4745>



#### Micro B USB to USB C Adapter

As technology changes and adapts, so does Adafruit, and speaking of adapting, this adapter has a Micro B USB jack and a USB C...

<https://www.adafruit.com/product/4299>

and



#### 64x32 RGB LED Matrix - 4mm pitch

Bring a little bit of Times Square into your home with this sweet 64 x 32 square RGB LED matrix panel. These panels are normally used to make video walls, here in New York we see them...

<https://www.adafruit.com/product/2278>



#### Black LED Diffusion Acrylic Panel 12" x 12" - 0.1" / 2.6mm thick

A nice whoppin' slab of some lovely black acrylic to add some extra diffusion to your LED Matrix project. This material is 2.6mm (0.1") thick and is made of special cast...

<https://www.adafruit.com/product/4594>





### 5V 2.5A Switching Power Supply with 20AWG MicroUSB Cable

Our all-in-one 5V 2.5 Amp + MicroUSB cable power adapter is the perfect choice for powering single-board computers like Raspberry Pi, BeagleBone, or anything else that's...

<https://www.adafruit.com/product/1995>



### USB cable - USB A to Micro-B

This here is your standard A to micro-B USB cable, for USB 1.1 or 2.0. Perfect for connecting a PC to your Metro, Feather, Raspberry Pi or other dev-board or...

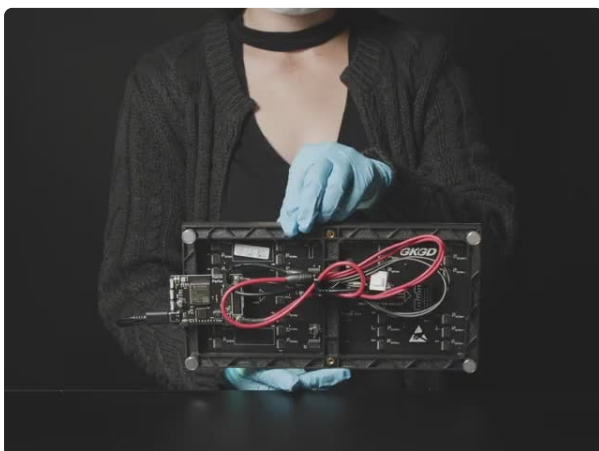
<https://www.adafruit.com/product/592>

---

## Using MatrixPortal

You can build this project with an all-in-one Matrix Portal board, its definitely the easiest and least-expensive way to go about it.

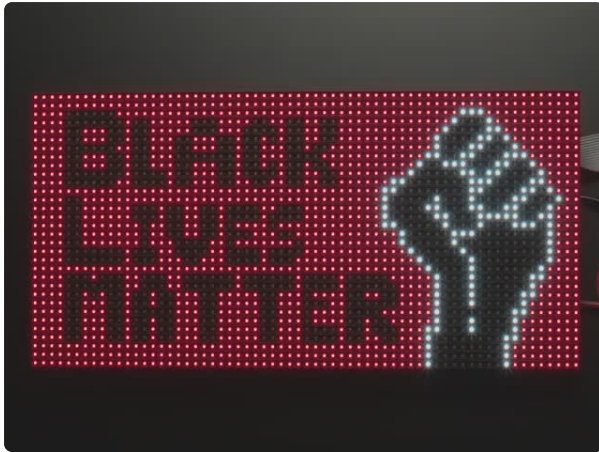
You will need a matrix portal, matrix, and USB C power/data cable



### Adafruit Matrix Portal - CircuitPython Powered Internet Display

Folks love our wide selection of RGB matrices and accessories, for making custom colorful LED displays... and our RGB Matrix Shields...

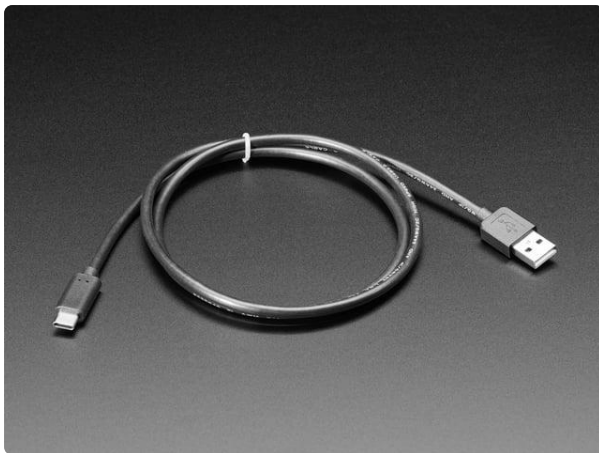
<https://www.adafruit.com/product/4745>



#### [64x32 RGB LED Matrix - 4mm pitch](https://www.adafruit.com/product/2278)

Bring a little bit of Times Square into your home with this sweet 64 x 32 square RGB LED matrix panel. These panels are normally used to make video walls, here in New York we see them...

<https://www.adafruit.com/product/2278>



#### [USB Type A to Type C Cable - approx 1 meter / 3 ft long](https://www.adafruit.com/product/4474)

As technology changes and adapts, so does Adafruit. This USB Type A to Type C cable will help you with the transition to USB C, even if you're still...

<https://www.adafruit.com/product/4474>

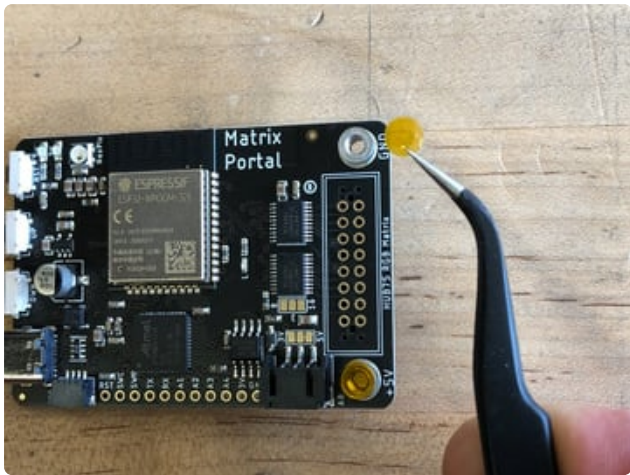
---

# Prep the MatrixPortal



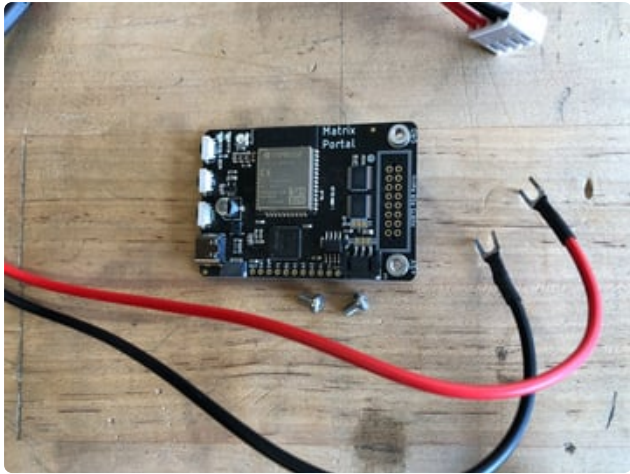
## Power Prep

The MatrixPortal supplies power to the matrix display panel via two standoffs. These come with protective tape applied (part of our manufacturing process) which **MUST BE REMOVED!**



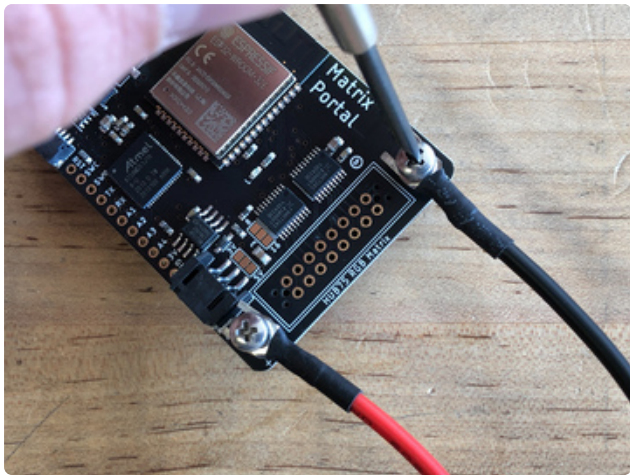
Use some tweezers or a fingernail to remove the two amber circles.





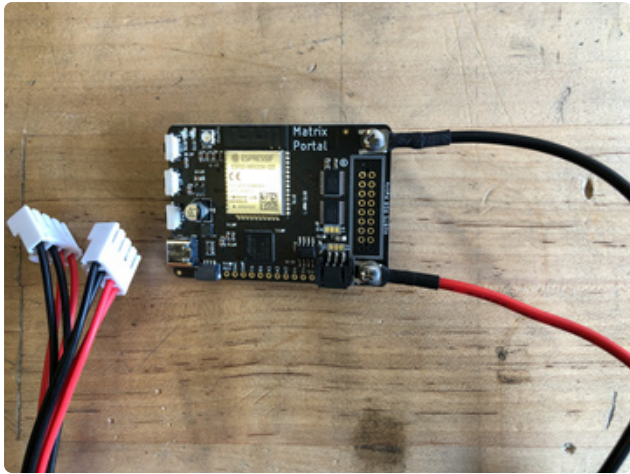
## Power Terminals

Next, screw in the spade connectors to the corresponding standoff.



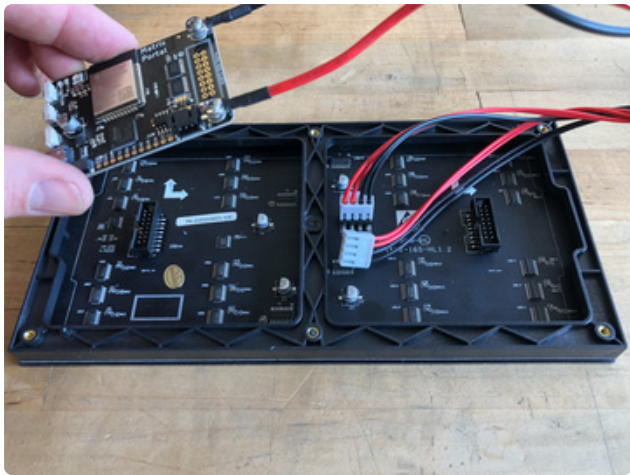
**red** wire goes to **+5V**

**black** wire goes to **GND**



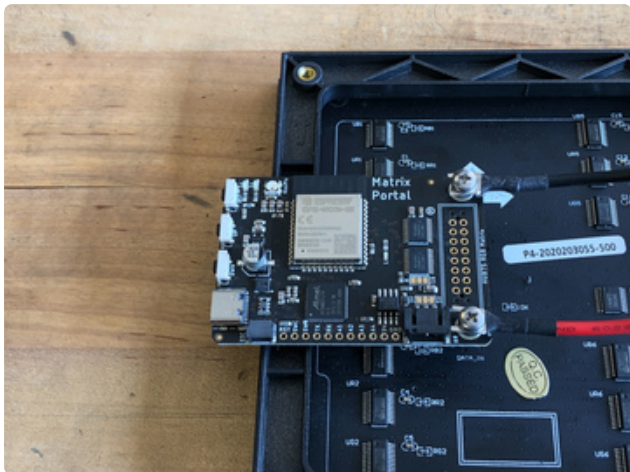
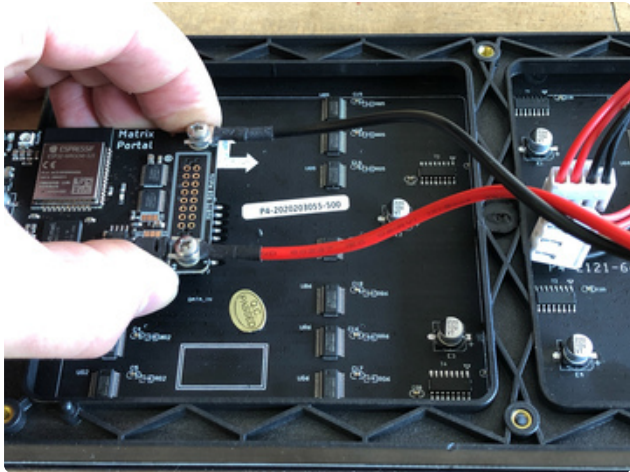
## Panel Power

Plug either one of the four-conductor power plugs into the power connector pins on the panel. The plug can only go in one way, and that way is marked on the board's silkscreen.



## Dual Matrix Setup

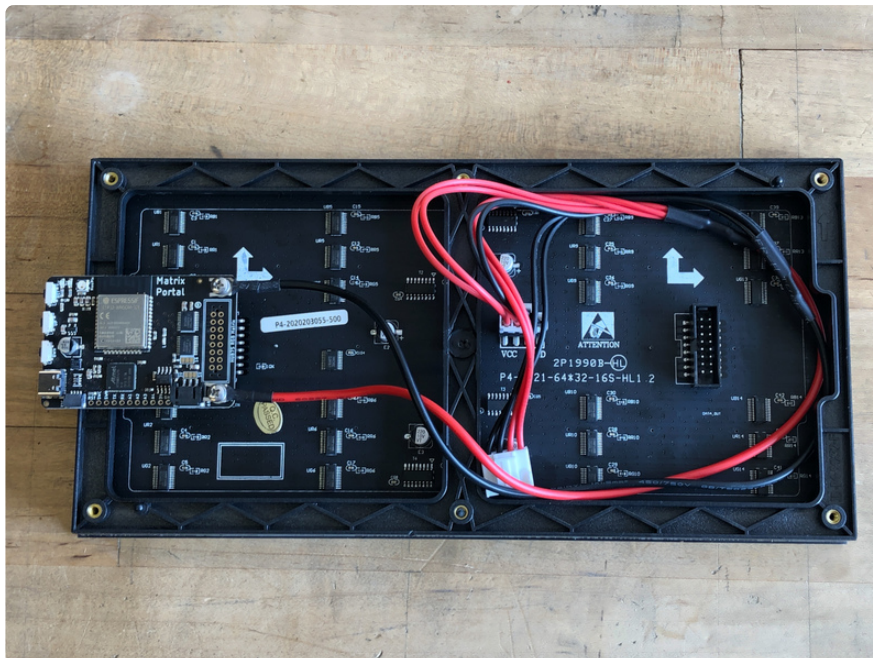
If you're planning to use a 64x64 matrix, [follow these instructions on soldering the Address E Line jumper](https://adafru.it/OdJ) (<https://adafru.it/OdJ>).



## Board Connection

Now, plug the board into the left side shrouded 8x2 connector as shown. The orientation matters, so take a moment to confirm that the **white indicator arrow on the matrix panel is oriented pointing up and right** as seen here and the MatrixPortal overhangs the edge of the panel when connected. This allows you to use the edge buttons from the front side.

Check nothing is impeding the board from plugging in firmly. If there's a plastic nub on the matrix that's keeping the Portal from sitting flat, cut it off with diagonal cutters







For info on adding LED diffusion acrylic, see the page [LED Matrix Diffuser](#).

## 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** drive to iterate.

## Set up CircuitPython Quick Start!

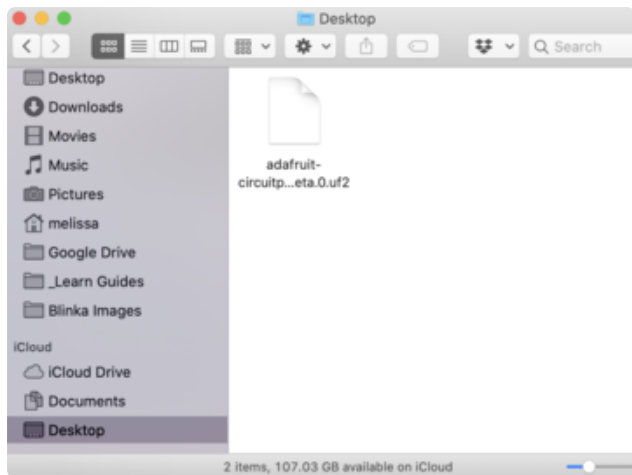
Follow this quick step-by-step for super-fast Python power :)

Download the latest version of  
CircuitPython for this board via  
[circuitpython.org](https://circuitpython.org)

<https://adafru.it/Nte>

## Further Information

For more detailed info on installing CircuitPython, check out [Installing CircuitPython \(https://adafru.it/Amd\)](https://adafru.it/Amd).

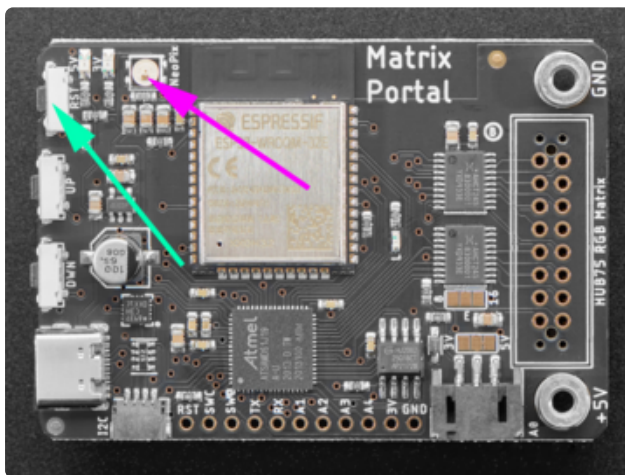


Click the link above and download the latest UF2 file.

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

Plug your MatrixPortal M4 into your computer using a known-good USB cable.

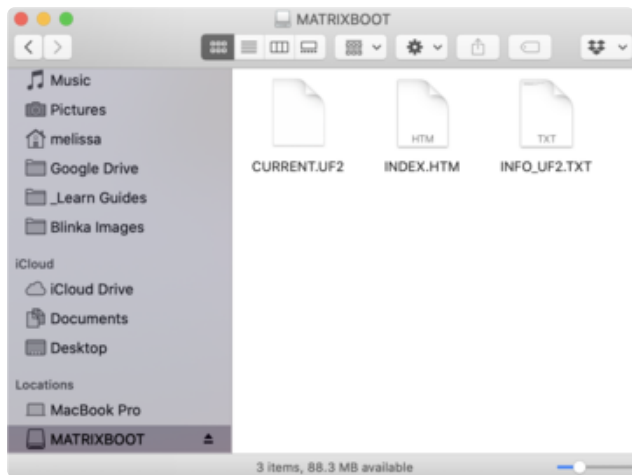
A lot of people end up using charge-only USB cables and it is very frustrating! So make sure you have a USB cable you know is good for data sync.



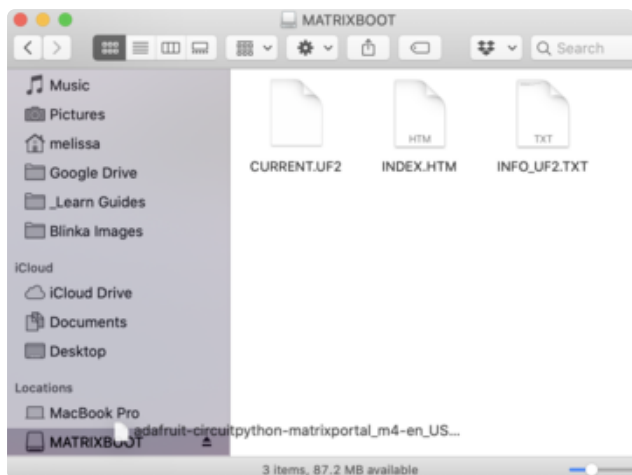
Double-click the **Reset** button (indicated by the green arrow) on your board, and you will see the NeoPixel RGB LED (indicated by the magenta arrow) turn green. If it turns red, check the USB cable, try another USB port, etc.

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

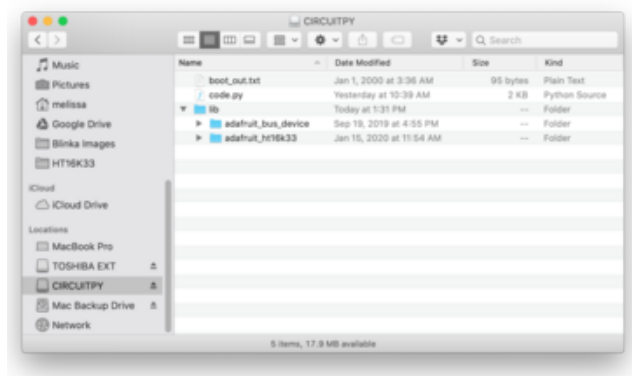




You will see a new disk drive appear called **MATRIXBOOT**.



Drag the `adafruit_circuitpython_etc.uf2` file to **MATRIXBOOT**.



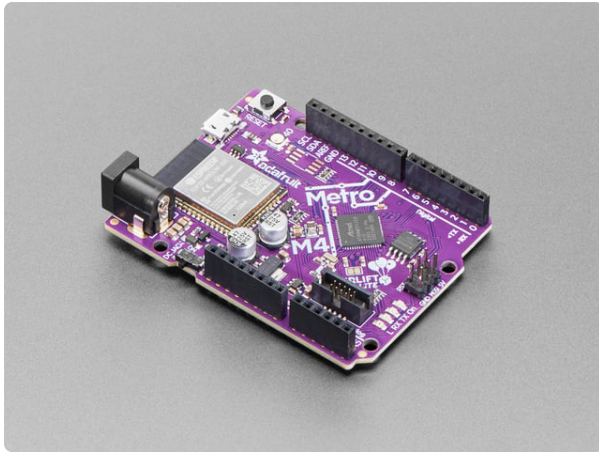
The LED will flash. Then, the **MATRIXBOOT** drive will disappear and a new disk drive called **CIRCUITPY** will appear.

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

## Using M4 Airlift

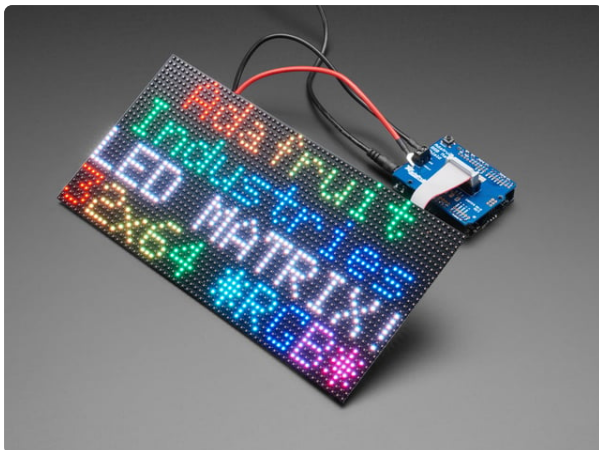
If you have a Metro M4 AirLift, you can build this project easily - you just need an RGB Matrix shield to help connect!

You will need a Metro M4 Airlift, matrix shield and matrix



### [Adafruit Metro M4 Express AirLift \(WiFi\) - Lite](#)

Give your next project a lift with AirLift - our witty name for the ESP32 co-processor that graces this Metro M4. You already know about the Adafruit Metro...  
<https://www.adafruit.com/product/4000>



### [Adafruit RGB Matrix Shield for Arduino](#)

Our RGB matrices are dazzling, with their hundreds or even thousands of individual RGB LEDs. Compared to NeoPixels, they've got great density, power usage and the...  
<https://www.adafruit.com/product/2601>



### [64x32 RGB LED Matrix - 4mm pitch](#)

Bring a little bit of Times Square into your home with this sweet 64 x 32 square RGB LED matrix panel. These panels are normally used to make video walls, here in New York we see them...  
<https://www.adafruit.com/product/2278>

---

## 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** drive to iterate.

# Set up CircuitPython Quick Start!

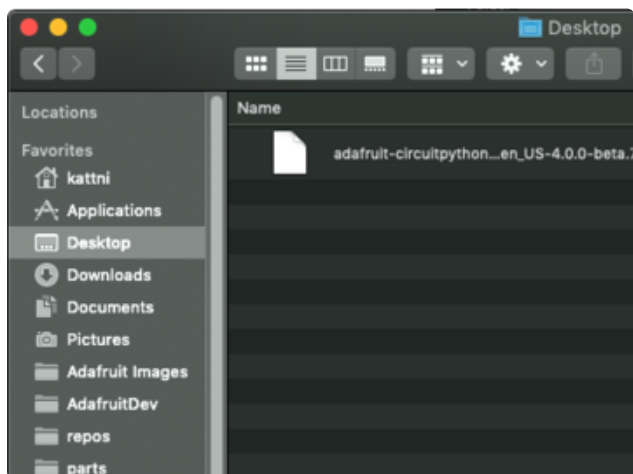
Follow this quick step-by-step for super-fast Python power :)

Download the latest version of  
CircuitPython for this board via  
circuitpython.org

<https://adafru.it/EDA>

## Further Information

For more detailed info on installing CircuitPython, check out [Installing CircuitPython \(https://adafru.it/Amd\)](https://adafru.it/Amd).



Click the link above and download the latest UF2 file.

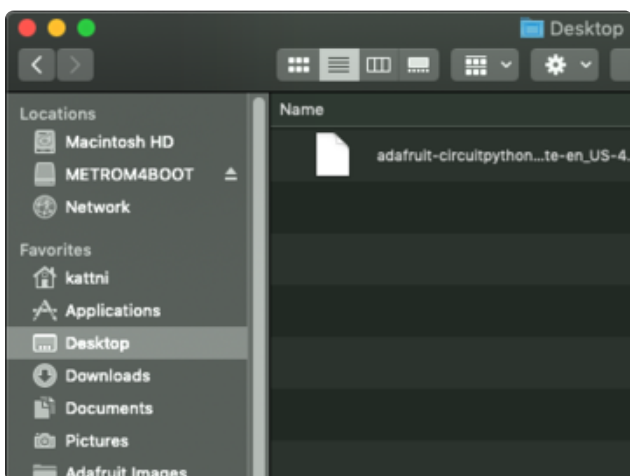
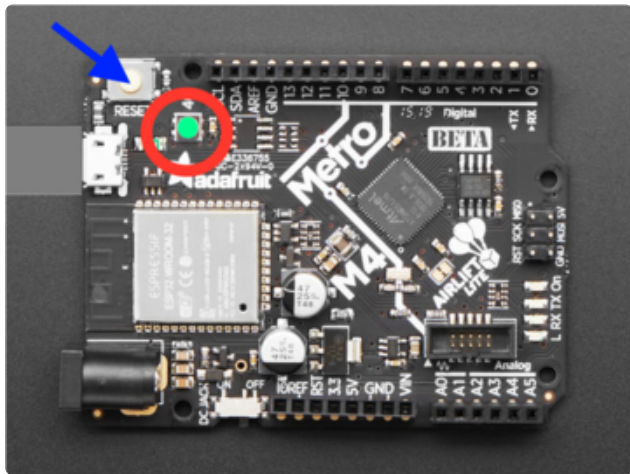
Download and save it to your desktop (or wherever is handy).

Plug your Metro M4 into your computer using a known-good USB cable.

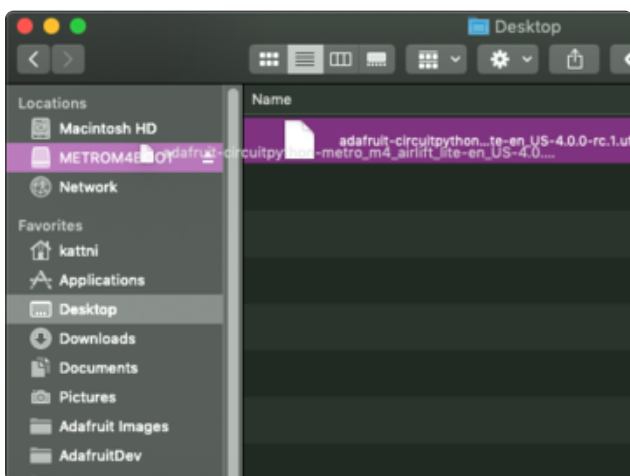
A lot of people end up using charge-only USB cables and it is very frustrating! So make sure you have a USB cable you know is good for data sync.

Double-click the **Reset** button next to the USB connector on your board (blue arrow), and you will see the NeoPixel RGB (LED circled in red) turn green. If it turns red, check the USB cable, try another USB port, etc.

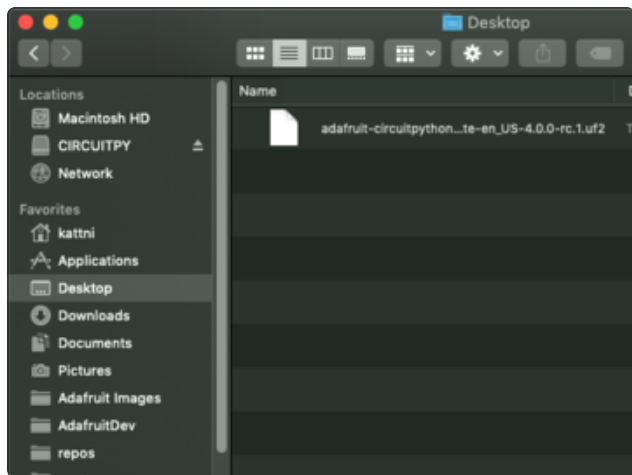
If double-clicking the Reset button doesn't work the first time, try again. Sometimes it can take a few tries to get the rhythm right!



You will see a new disk drive appear called **METROM4BOOT**.



Drag the `adafruit_circuitpython_etc.uf2` file to **METROM4BOOT**.



The LED will flash. Then, the **METROM4BOOT** drive will disappear and a new disk drive called **CIRCUITPY** will appear.

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

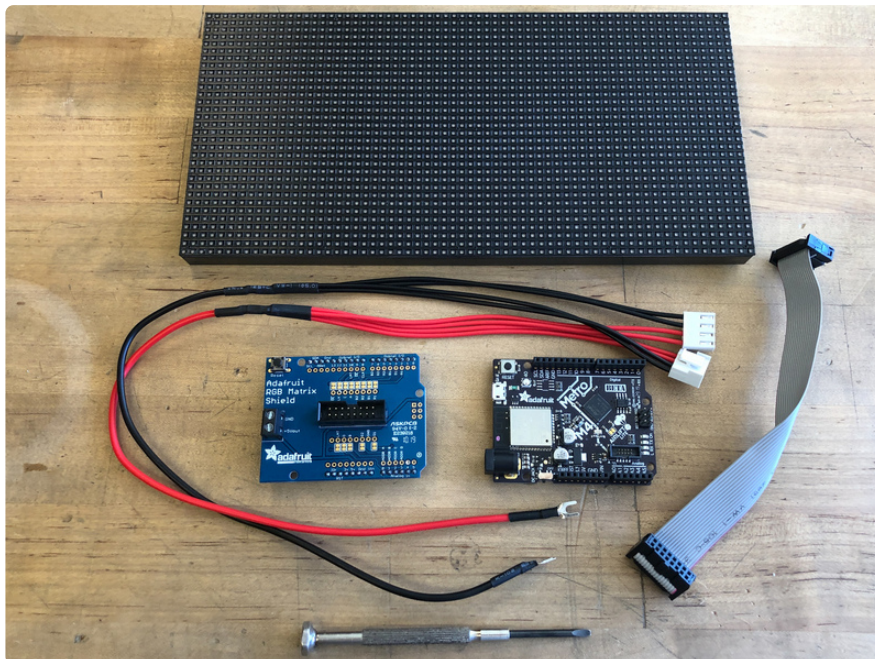
---

## Build Metro M4 Airlift Matrix Display

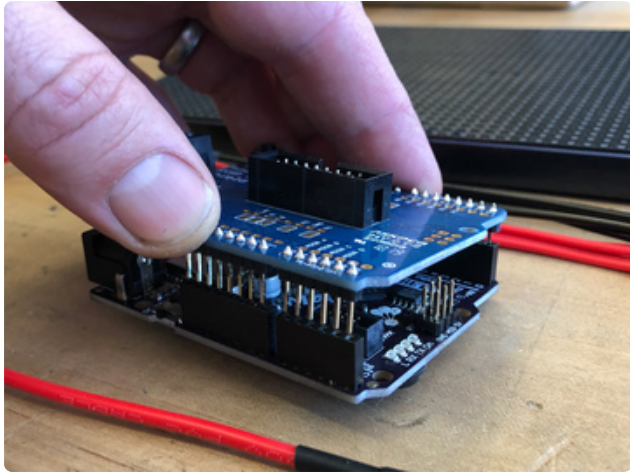
### Assembly

Talking to an LED matrix display can be tricky! The 64 x 32 LED used here has a whopping 2,048 pixels, and each can display RGB colors, which makes for a whole lot of data to sling around. Thankfully, our RGB Matrix shield paired with the Metro M4 Airlift does most of the heavy lifting.

Let's assemble the boards and the display so we can get things running!





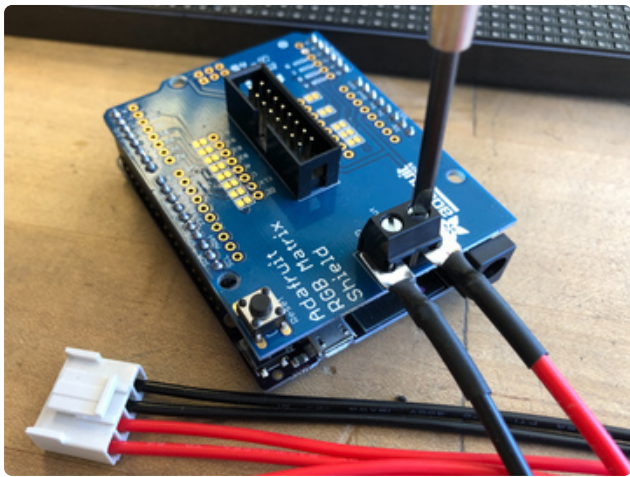


## Shields Up

First, add the male headers, screw terminal block, and the 8x2-pin socket to the Matrix shield, by [following this guide \(https://adafruit.it/FBU\)](https://adafruit.it/FBU). Be careful to match the socket polarity to the silkscreen image on the board.

Be sure to also perform the clock pin mod [as shown here \(https://adafruit.it/FBU\)](https://adafruit.it/FBU).

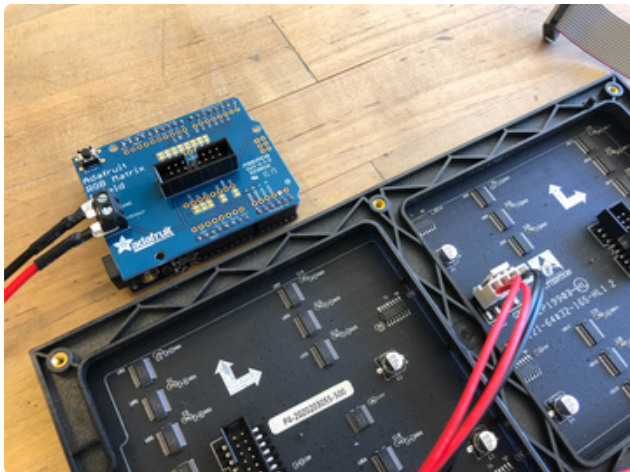
Then plug the shield into the Metro M4 Airlift.

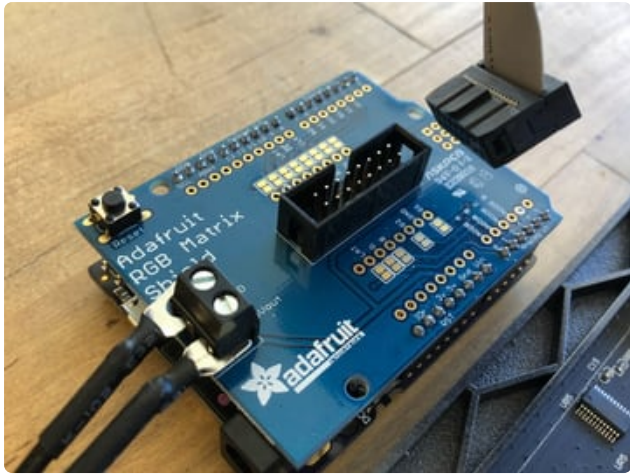


## Power Connections

To provide power, we'll screw the wiring harness connectors to the screw terminal blocks of the shield. Be sure to match the **black wire to GND** and the **red wire to +5Vout**.

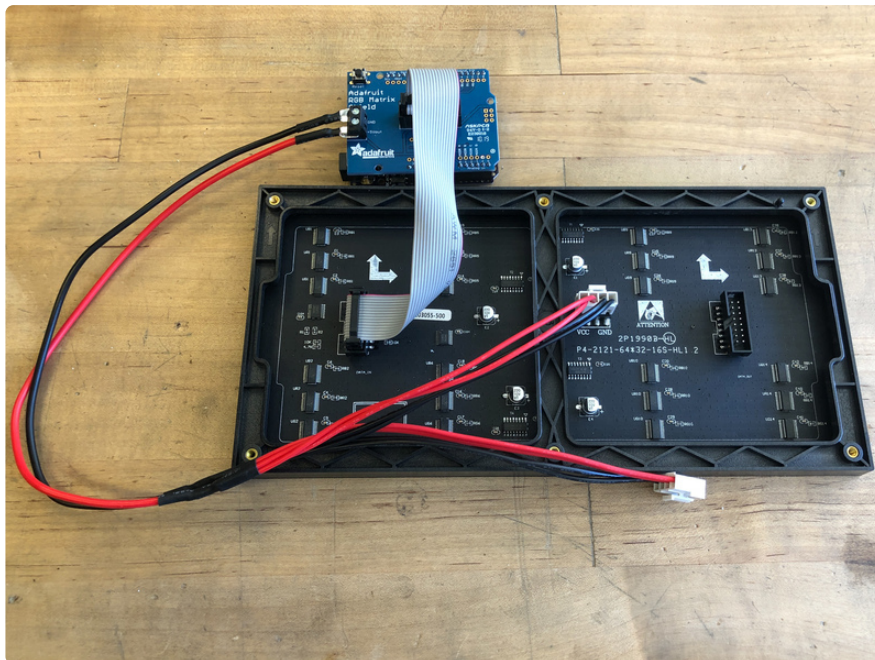
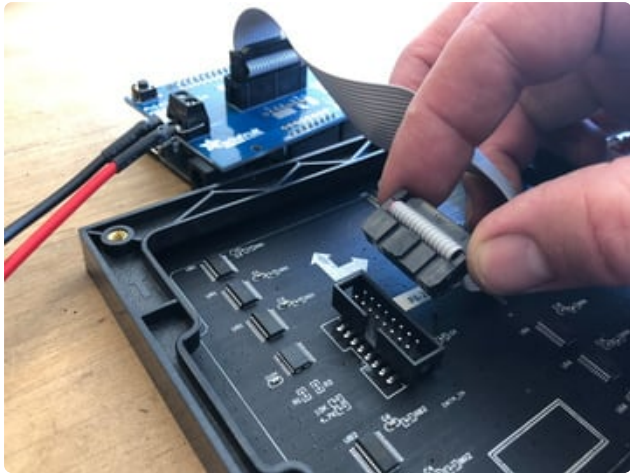
Now, simply plug the other end into the panel's power header. It can only go in one way.



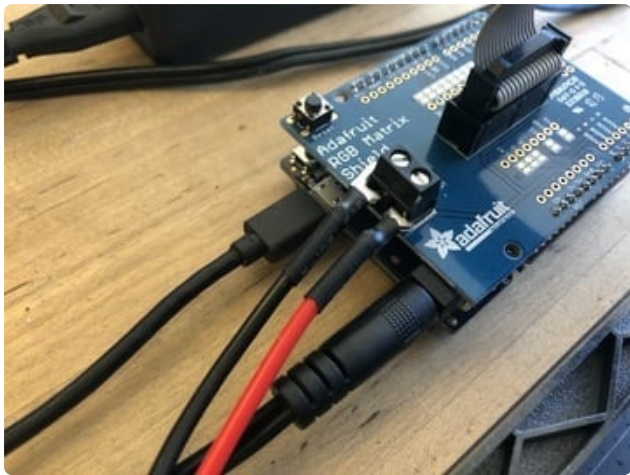
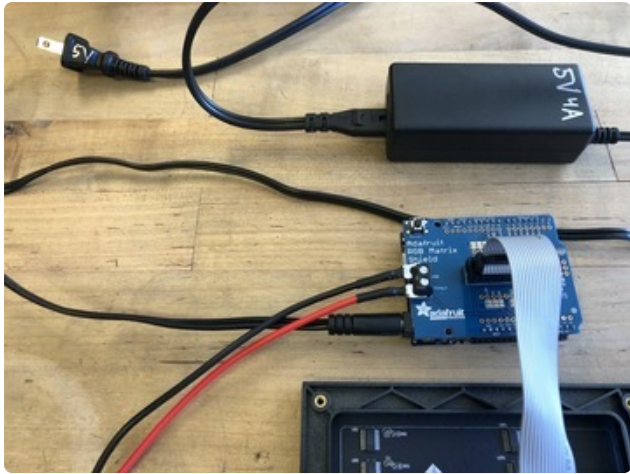


## Data Cable

Plug in the two ends of the ribbon cable, note that the connectors are keyed to only fit in the correct orientation.





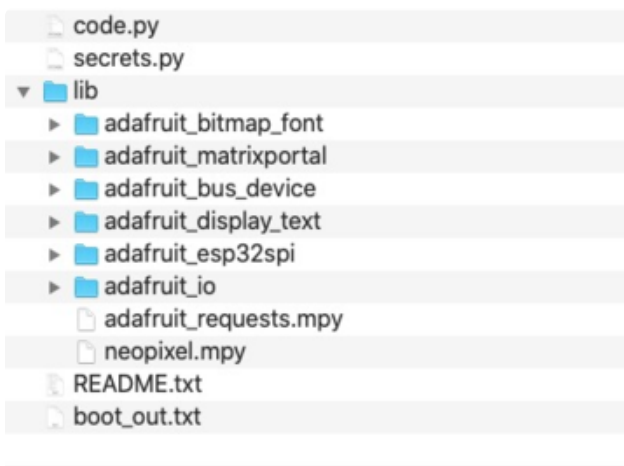


## Wall Adapter

We'll power the Metro M4 from the 5V 2.5 (or a 4A) DC wall adapter plugged into the barrel jack. Even though USB can provide power to the board, the current isn't adequate for lighting up hundreds and thousands of LEDs!

For info on adding LED diffusion acrylic, see the page [LED Matrix Diffuser](#).

## Code the Purple Air Display



## Libraries

We'll need to make sure we have these libraries installed. (Check out this [link \(https://adafru.it/ABU\)](https://adafru.it/ABU) on installing libraries if needed.)

adafruit\_bitmap\_font  
adafruit\_bus\_device  
adafruit\_display\_text  
adafruit\_esp32spi  
adafruit\_io  
adafruit\_matrixportal  
adafruit\_requests.mpy  
neopixel.mpy

## Connect to the Internet

Once you have CircuitPython setup and libraries installed we can get your board connected to the Internet. The process for connecting can be found [here \(https://adafru.it/NFK\)](https://adafru.it/NFK).

## Text Editor

Adafruit recommends using the Mu editor for editing your CircuitPython code. You can get more info in [this guide \(https://adafru.it/ANO\)](https://adafru.it/ANO).

Alternatively, you can use any text editor that saves simple text files.

## Code

Click the Download: Zip File link below in the code window to get a zip file with all the files needed for the project. Copy **code.py** from the zip file and place it on the **CIRCUITPY** drive.

```
# SPDX-FileCopyrightText: 2020 John Park for Adafruit Industries
#
# SPDX-License-Identifier: MIT

# Purple Air AQI Display
# for Metro M4 Airlift with RGB Matrix Shield
# or Matrix Portal
# and 64 x 32 RGB LED Matrix

import time
import board
import terminalio
from adafruit_matrixportal.matrixportal import MatrixPortal

# 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

def aqi_transform(val):
    aqi = pm_to_aqi(val) # derive Air Quality Index from Particulate Matter 2.5
    value
    return "AQI: %d" % aqi

def message_transform(val): # picks message based on thresholds
    index = aqi_to_list_index(pm_to_aqi(val))
    messages = (
        "Hazardous",
        "Very Unhealthy",
        "Unhealthy",
        "Unhealthy for Sensitive Groups",
        "Moderate",
        "Good",
```

```

    )

    if index is not None:
        return messages[index]
    return "Unknown"

SENSOR_ID = 3085 # Poughkeepsie # 30183 LA outdoor / 37823 oregon / 21441 NYC
SENSOR_REFRESH_PERIOD = 300 # seconds
DATA_SOURCE = f"https://api.purpleair.com/v1/sensors/{SENSOR_ID}?fields=pm2.5_10minute"
SCROLL_DELAY = 0.02
DATA_LOCATION = ["sensor", "stats", "pm2.5_10minute"] # navigate the JSON response

# --- Display setup ---
matrixportal = MatrixPortal(
    status_neopixel=board.NEOPIXEL,
    debug=True,
    url=DATA_SOURCE,
    headers={"X-API-Key": secrets["purple_air_api_key"], # purpleair.com
            "Accept": "application/json"},
    json_path=(DATA_LOCATION, DATA_LOCATION),
)

# Create a static label to show AQI
matrixportal.add_text(
    text_font=terminalio.FONT,
    text_position=(8, 7),
    text_transform=aqi_transform,
)

# Create a scrolling label to show level message
matrixportal.add_text(
    text_font=terminalio.FONT,
    text_position=(0, 23),
    scrolling=True,
    text_transform=message_transform,
)

# pylint: disable=too-many-return-statements
def aqi_to_list_index(aqi):
    aqi_groups = (301, 201, 151, 101, 51, 0)
    for index, group in enumerate(aqi_groups):
        if aqi >= group:
            return index
    return None

def calculate_aqi(Cp, Ih, Il, BPh, BPl): # wikipedia.org/wiki/Air_quality_index#Computing the AQI
    return round(((Ih - Il)/(BPh - BPl)) * (Cp - BPl) + Il)

def pm_to_aqi(pm):
    pm = float(pm)
    if pm < 0:
        return pm
    if pm > 1000:
        return 1000
    if pm > 350.5:
        return calculate_aqi(pm, 500, 401, 500, 350.5)
    elif pm > 250.5:
        return calculate_aqi(pm, 400, 301, 350.4, 250.5)
    elif pm > 150.5:
        return calculate_aqi(pm, 300, 201, 250.4, 150.5)
    elif pm > 55.5:
        return calculate_aqi(pm, 200, 151, 150.4, 55.5)
    elif pm > 35.5:
        return calculate_aqi(pm, 150, 101, 55.4, 35.5)
    elif pm > 12.1:
        return calculate_aqi(pm, 100, 51, 35.4, 12.1)
    elif pm >= 0:

```



```

        return calculate_aqi(pm, 50, 0, 12, 0)
    else:
        return None

def get_color(aqi):
    index = aqi_to_list_index(aqi)
    colors = (
        (115, 20, 37),
        (140, 26, 75),
        (234, 51, 36),
        (239, 133, 51),
        (255, 255, 85),
        (104, 225, 67),
    )
    if index is not None:
        return colors[index]
    return (150, 150, 150)

sensor_refresh = None
while True:
    # only query the weather every 10 minutes (and on first run)
    if (not sensor_refresh) or (time.monotonic() - sensor_refresh) >
SENSOR_REFRESH_PERIOD:
        try:
            value = matrixportal.fetch()
            print("Response is", value)
            matrixportal.set_text_color(get_color(pm_to_aqi(value[0])))
            sensor_refresh = time.monotonic()
        except RuntimeError as e:
            print("Some error occurred, retrying! -", e)
            continue

# Scroll it
matrixportal.scroll_text(SCROLL_DELAY)

```

## Purple Air

Purple Air is an air quality monitoring network based on IoT sensors. You can learn more about the sensors [here \(https://adafru.it/NC6\)](https://adafru.it/NC6). When people buy and install their own sensors and enable them to appear on the Purple Air map, we all benefit!

Head to the [Purple Air map here \(https://adafru.it/NC7\)](https://adafru.it/NC7) and then try to find a sensor near your location. Even if you don't have your own sensor installed outside, you may be able to find a sensor nearby that you can choose to monitor on your Matrix display.

Here, I've zoomed in on a sensor in downtown Los Angeles. If I look at the URL for this sensor, I can see the sensor ID number:

<https://www.purpleair.com/map?opt=1/mAQI/a10/cC0&select=30183>

The number after `select=` is `30183` so I'll copy that number to use in my `code.py` file so that I can display that air quality sensor value.



In your `code.py` file, go to the line that has the variable `SENSOR_ID` and paste in the number of the sensor you'd like to display, then re-save the file to your **CIRCUITPY** drive.

For example:

```
SENSOR_ID = 30183
```

## API Key

Purple Air made a change in 2022 that now requires an API Key. To get a key, you'll need to head over to <https://develop.purpleair.com/keys> (<https://adafru.it/19cE>) and

sign in with a Google Account. When you first sign up, your new account should have 1,000,000 free points. A new project will automatically be created for you. When generating your API Key, that project is selected by default. Read Access will also be selected for the API key, which is all you'll need.

Copy and paste this key into your **secrets.py** file that is on the root level of your **CIRCUITPY** drive, so it looks something like this:

```
secrets = {
    'ssid' : 'your_wifi_ssid',
    'password' : 'your_wifi_password',
    'purple_air_api_key' : 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx'
}
```

You will also need to assign points to the project before you can use the API key, otherwise you will get a **Payment Required error** message. Each API call uses up points that can vary depending on the amount of data returned and how frequently it is called. The code specifies which fields to include in order to minimize the cost. It is also set to check every 10 minutes by default. Increasing the frequency by lowering **SENSOR\_REFRESH\_PERIOD** will use up points faster. Once you have used up all of your points, it will require purchasing more points. You can limit the amount of points per a project by only assigning a portion of them.

## How it Works

### Libraries

Most of the heavy lifting is done by the **adafruit\_matrixportal** library we import. This will take care of getting online through the access point (the info is in your **secrets.py** file), as well as fetching and parsing the PM2.5 particle data from the Purple Air JSON file.

### Functions

The **aqi\_transform()** function converts PM (particulate matter) values to AQI (air quality index) values and returns the text string we'll display. It does this by using both the **pm\_to\_aqi()** and the **calculate\_aqi()** functions.

The **message\_transform()** function selects the text description associated with a given AQI value.

## Variables

These are the variables used to pick a Purple Air sensor as well as a refresh rate, data source, scroll speed, and the data location withing the JSON file:

```
SENSOR_ID = 3085 # Poughkeepsie
SENSOR_REFRESH_PERIOD = 300 # seconds
DATA_SOURCE = f"https://api.purpleair.com/v1/sensors/{SENSOR_ID}?fields=pm2.5_10minute"
SCROLL_DELAY = 0.02
DATA_LOCATION = ["sensor", "stats", "pm2.5_10minute"] # navigate the JSON response
```

## Matrix Setup

Next, the Matrix is set up for displaying text data using the **terminalio** font.

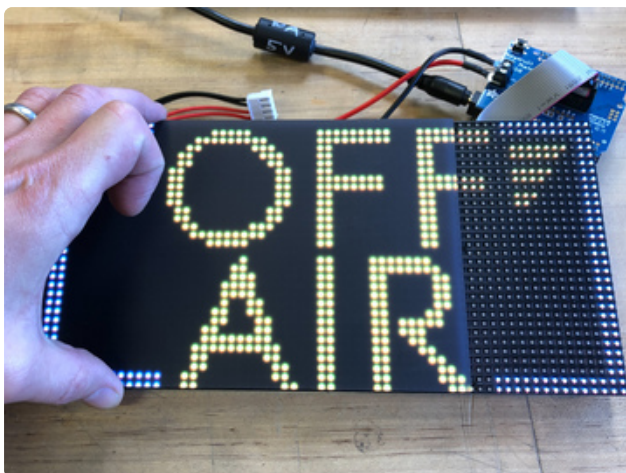
## Main Loop

In the main loop of the code the latest value from the selected Purple Air sensor is fetched using `value = matrixportal.fetch()` and then the result is formatted and updated to the display.

The text description is then scrolled across the display and the process is repeated every thirty seconds.

---

## LED Matrix Diffuser



### LED Diffusion Acrylic

You can add an [LED diffusion acrylic faceplate](http://adafru.it/4594) (<http://adafru.it/4594>) to the your LED matrix display. (Pictured here with the [ON AIR project](https://adafru.it/MPE) (<https://adafru.it/MPE>))

This can help protect the LEDs as well as enhance the look of the sign both indoors and out by reducing glare and specular highlights of the plastic matrix grid.

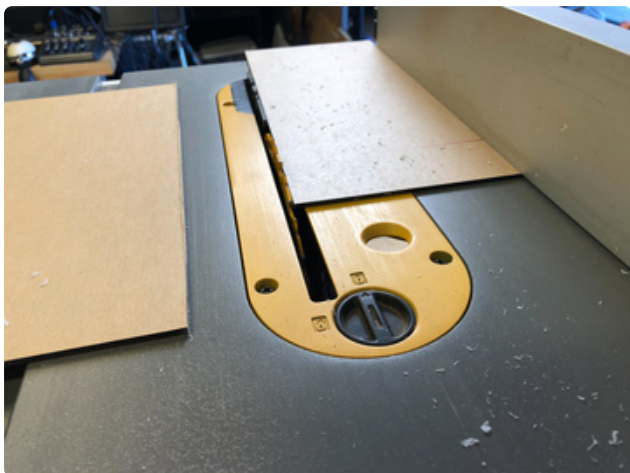
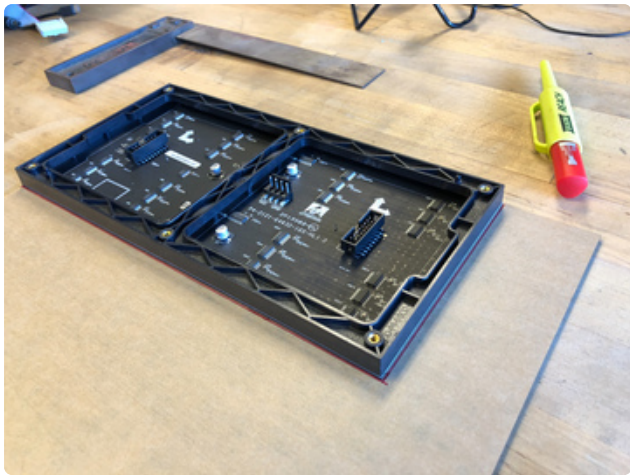


## Measure and Cut the Plastic

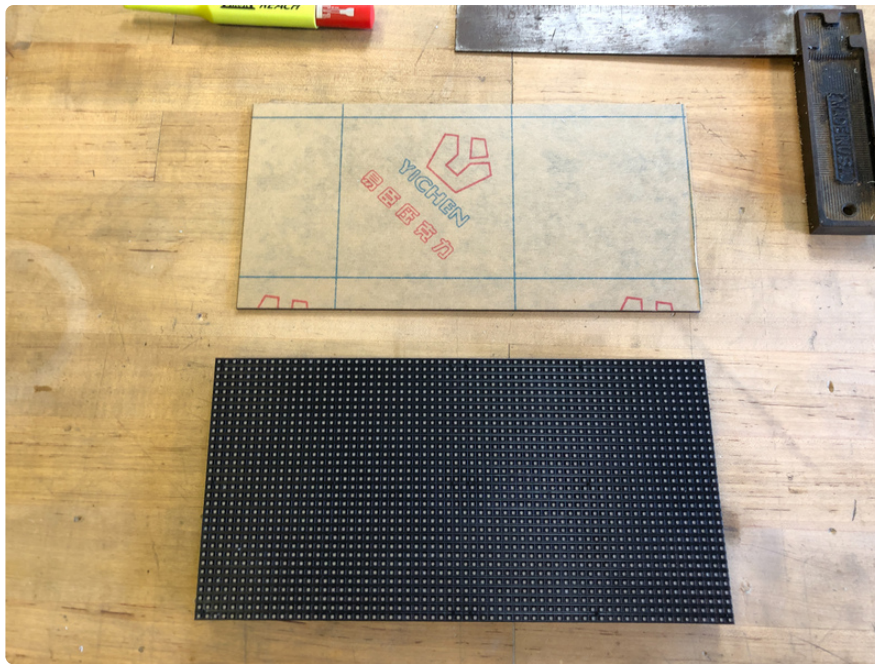
You can use the sign to measure and mark cut lines on the paper backing of the acrylic sheet.

Then, use a tablesaw or bandsaw with a fine toothed blade and a guide or sled to make the cuts.

Note: it is possible to score and snap acrylic, but it can be very tricky to get an even snap without proper clamping.







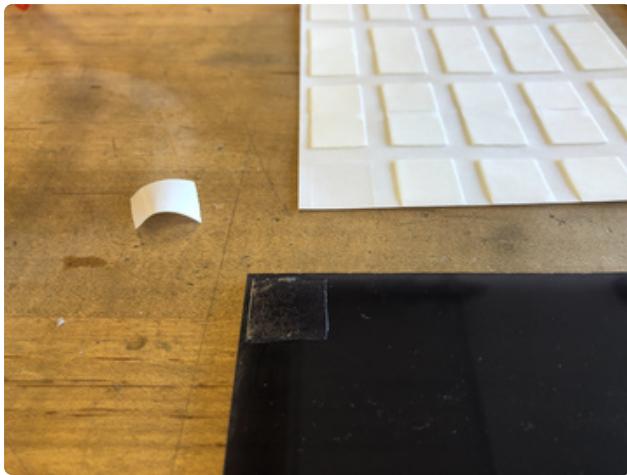
Peel away the paper backing from both sides and set the acrylic onto your matrix display with the matte finished side facing out.



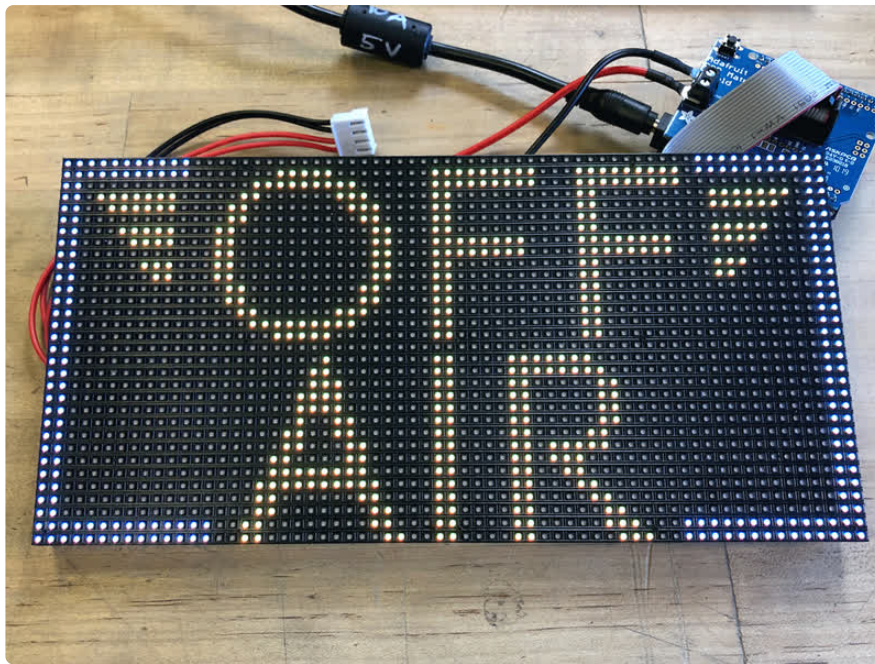
## Uglu Dashes

The best method we've found for adhering acrylic to the matrix display is to use [Uglu Dashes clear adhesive rectangles from Pro Tapes](https://adafru.it/NcP) (<https://adafru.it/NcP>). They are incredibly strong (although can be removed if necessary), easy to apply, and are invisible once attached.

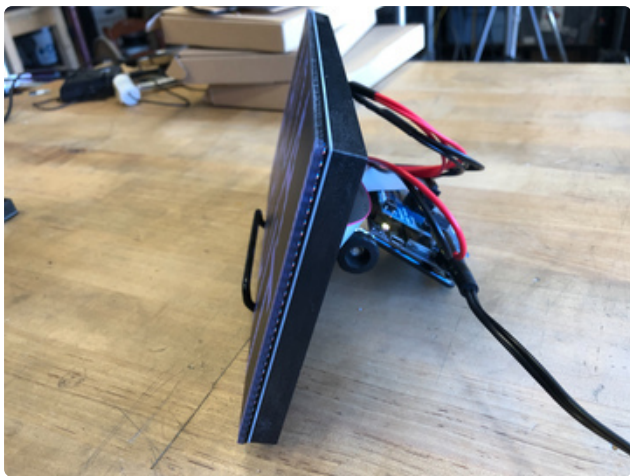
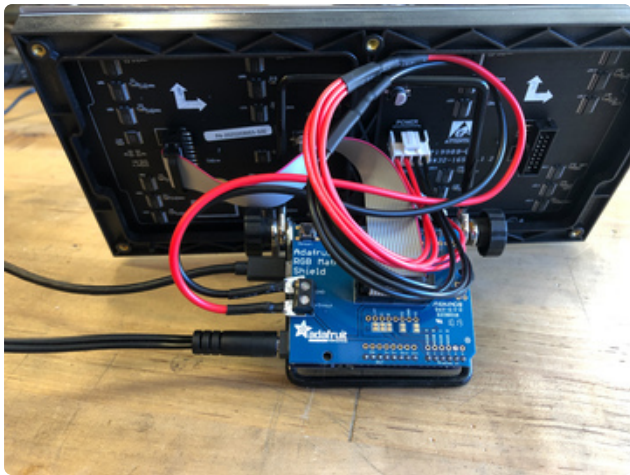
Use one at each corner and one each at the halfway point of the long edges, then press the acrylic and matrix panel together for about 20 seconds.



Here you can see the impact of using the diffusion acrylic. (Pictured here with the ON AIR sign project)







## Stand

A very simple and attractive way to display your matrix is with the adjustable [bent-wire stand](http://adafru.it/1679) (<http://adafru.it/1679>).





Alternately, you can use a frame, [3D printed brackets \(https://adafru.it/MZf\)](https://adafru.it/MZf), tape, glue, or even large binder clips to secure the acrylic to the sign and then mount it on a wall, shelf, or display cabinet.

[These mini-magnet feet \(http://adafru.it/4631\)](http://adafru.it/4631) can be used to stick the sign to a ferrous surface.