

Prioritizing orders for COVID-19 related needs, health care professionals, and urgent orders. All other orders are being shipped as soon as possible.

- [Shop](#)
- [Learn](#)
- [Blog](#)
- [Forums](#)
- [Videos](#)
- [Adabox](#)
- [IO](#)

[toggle menu](#)



0

- [Sign In](#) | [Create Account](#)
- [New Guides](#)
- [Series](#)
  
- [Shop](#)
- [Learn](#)
- [Blog](#)
- [Forums](#)
- [Videos](#)
- [Adabox](#)
- [IO](#)

[Sign In](#)

0

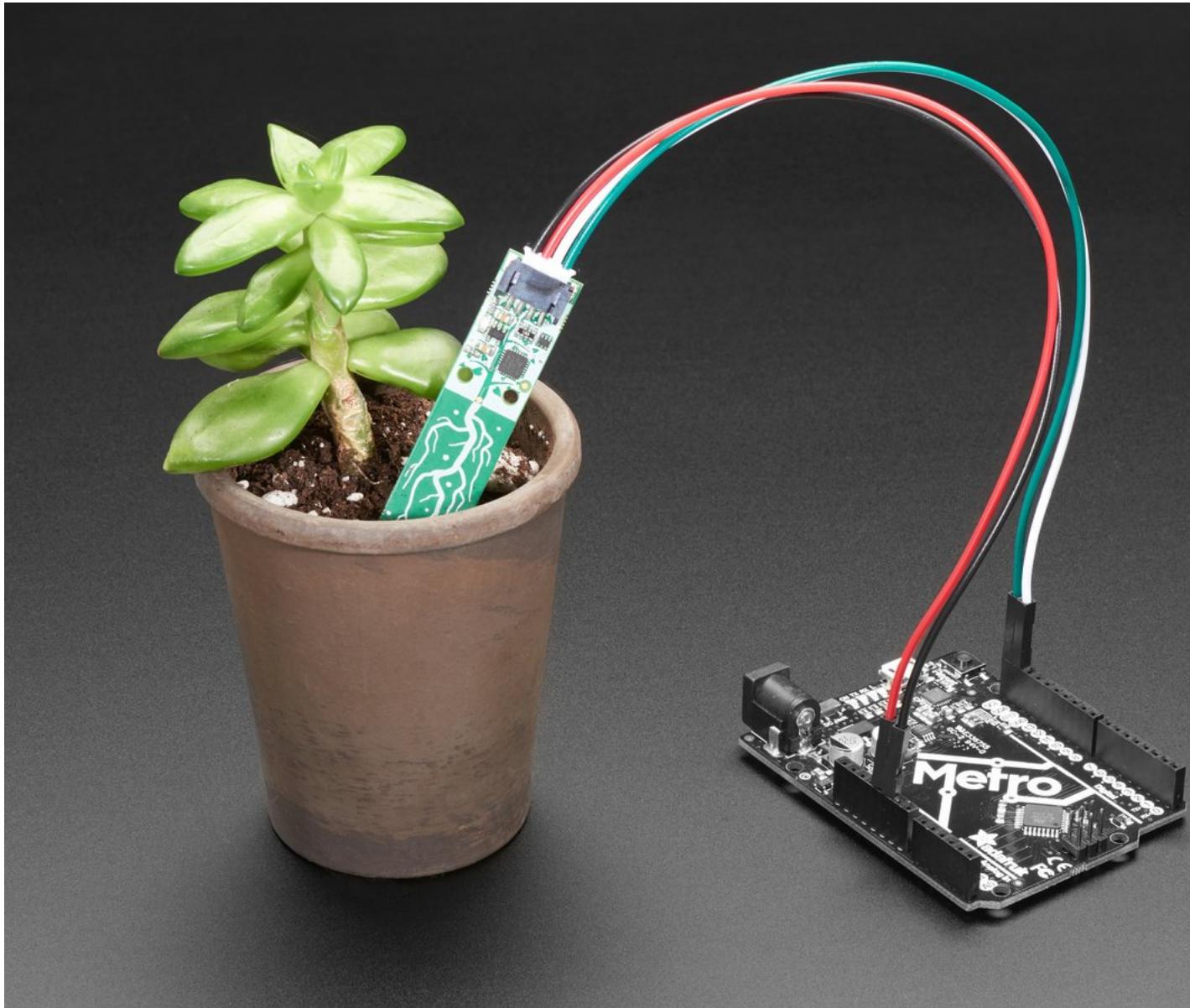
- [Explore & Learn](#)

Learn Categories [view all](#)

- [AdaBox](#)
- [Adafruit Products](#)
- [Arduino Compatibles](#)
- [Breakout Boards](#)
- [Circuit Playground](#)
- [CircuitPython](#)
- [Collin's Lab](#)
- [Components](#)
- [Community Support](#)
- [Customer & Partner Projects](#)
- [Development Boards](#)
- [Educators](#)
- [EL Wire/Tape/Panel](#)
- [Feather](#)
- [Hacks](#)
- [Internet of Things – IOT](#)
- [LCDs & Displays](#)
- [LEDs](#)
- [Maker Business](#)
- [micro:bit](#)
- [Microcontrollers](#)
- [Programming](#)
- [Projects](#)
- [Raspberry Pi](#)
- [Robotics & CNC](#)
- [Sensors](#)
- [Tools](#)
- [Wearables](#)
- [3D Printing](#)
- [Circuit](#)
- [MakeCode](#)
- [Trellis](#)
- [Machine Learning](#)

Series [view all](#)

- [Circuit Playground](#)
- [Adafruit IO Basics](#)



## STEMMA

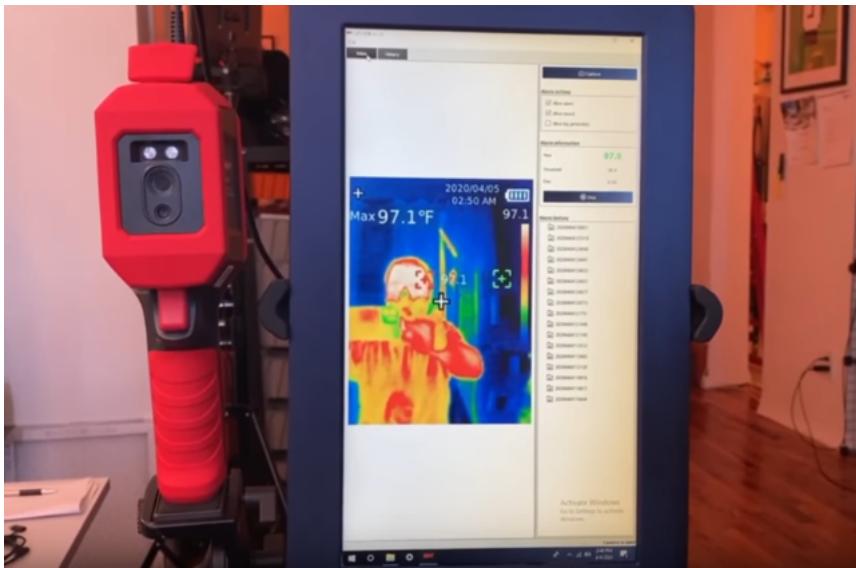
Plug-n-play components

[Get connected](#)

- [New Guides](#)

### Latest Guides

[View All](#)



[How to use your precision fever scanner camera](#)  
[lady ada](#)



#### [UTi165 Thermal Fever Scanner Camera](#)

Beginner

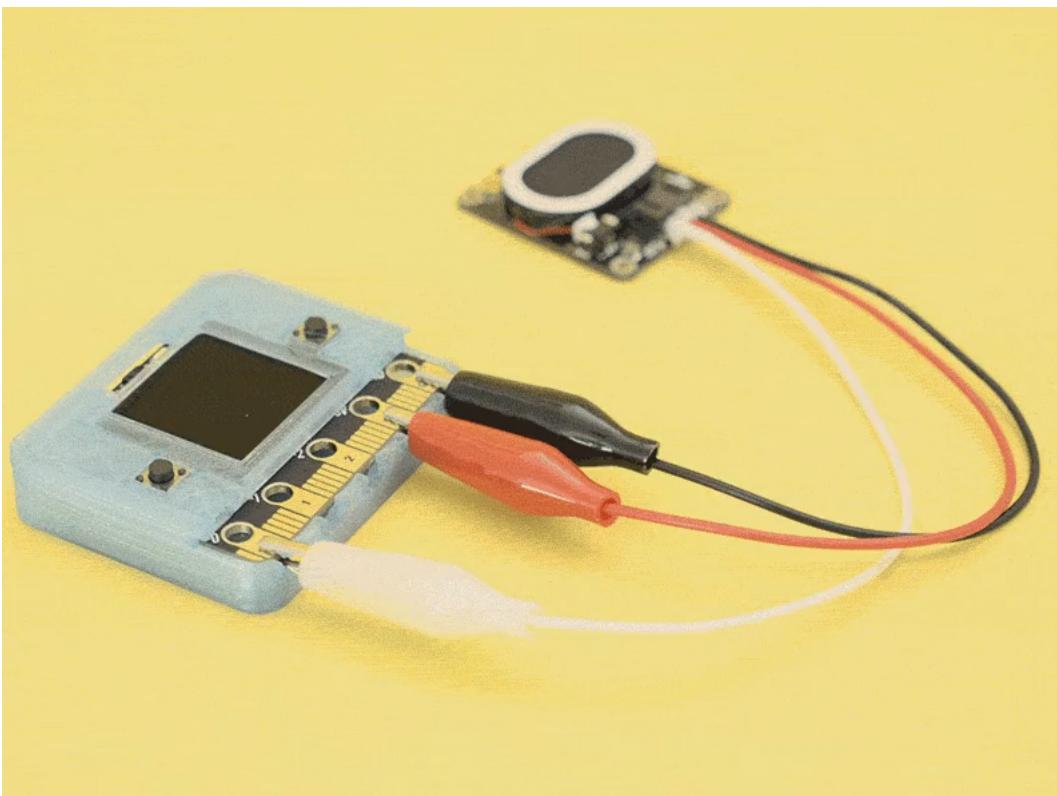


[Log pulse oximeter data to an SD card over Bluetooth LE with Feather and Adalogger](#)  
[John Park](#)



#### [Pulse Oximeter Wireless Data Logger](#)

Intermediate



[A Compact, Translucent Case for CLUE](#)  
[Ruiz Brothers](#)



#### [CLUE Slim Case](#)

Beginner

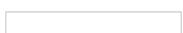


[Use a CLUE to create a hexagram you can use to consult the I Ching.](#)  
[Carter Nelson](#)



#### [CircuitPython CLUE I Ching Caster](#)

Beginner





## [Arcada Animated GIF Display](#)

[Treasure your memes by having them display on your desk](#)

- [Overview](#)
- [Convert Your GIF](#)
- [Quickstart](#)
- [Loading GIFs](#)
- [Arcada Libraries](#)
- [Build In Arduino](#)
- [Customization](#)
- [Example GIFs](#)
- 
- [Featured Products](#)
- [Multiple Pages](#)
- [Download PDF](#)

### Contributors

[lady\\_ada](#)

[Ruiz Brothers](#)

[Feedback? Corrections?](#)

[ARDUINO COMPATIBLES ADABOX LCDS & DISPLAYS / GRAPHIC TFTS](#)

## Overview

[lady\\_ada](#)

Animated GIFs are a staple of the Internet – and thanks to the speedy SAMD51 chip and our Arcada display library, we can show animated GIFs right from the onboard filesystem for keeping your favorite memes with you all the time even when the Internet is down!

Animated GIFs on the Adafruit PyPortal @adafruit #adaf...



In this project we'll show you how to use some nifty Arduino C++ code that can decode GIF files, and display them on a TFT display. Behind the scenes we use DMA for fast display updates so you don't see any flickering.

The SAMD51 microcontroller we use is pretty fast and does the best it can to decode GIFs, but at full-size, complex GIFs may not run at full speed!

We recommend sticking to GIFs that don't have high frame rates, lots of colors, or need to run at full-speed to look good. Plenty of GIFs will look just fine, just don't expect highest quality / high definition!

## Supported Boards

The GIF decoder builds on Adafruit Arcada which supports the SAMD51 chipset and color GFX displays. At this time we recommend the following boards which 'just work' and also have touch/joypad controls:



### [Adafruit PyPortal – CircuitPython Powered Internet Display](#)

PRODUCT ID: 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...

\$54.95

IN STOCK

[Add to Cart](#)



### [Adafruit PyGamer Starter Kit](#)

PRODUCT ID: 4277

<https://learn.adafruit.com/pyportal-animated-gif-display?view=all>

Please note: you may get a royal blue or purple case with your starter kit (they're both lovely colors)What fits in your pocket, is fully Open...

OUT OF STOCK

[Out of Stock](#)



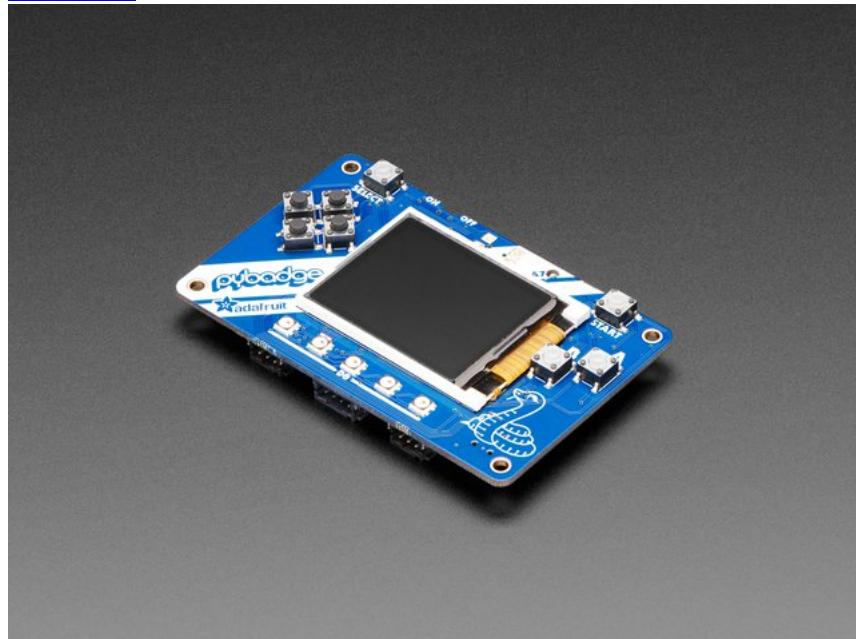
### [Adafruit PyGamer for MakeCode Arcade, CircuitPython or Arduino](#)

PRODUCT ID: 4242

What fits in your pocket, is fully Open Source, and can run CircuitPython, MakeCode Arcade or Arduino games you write yourself? That's right, it's the Adafruit...

OUT OF STOCK

[Out of Stock](#)



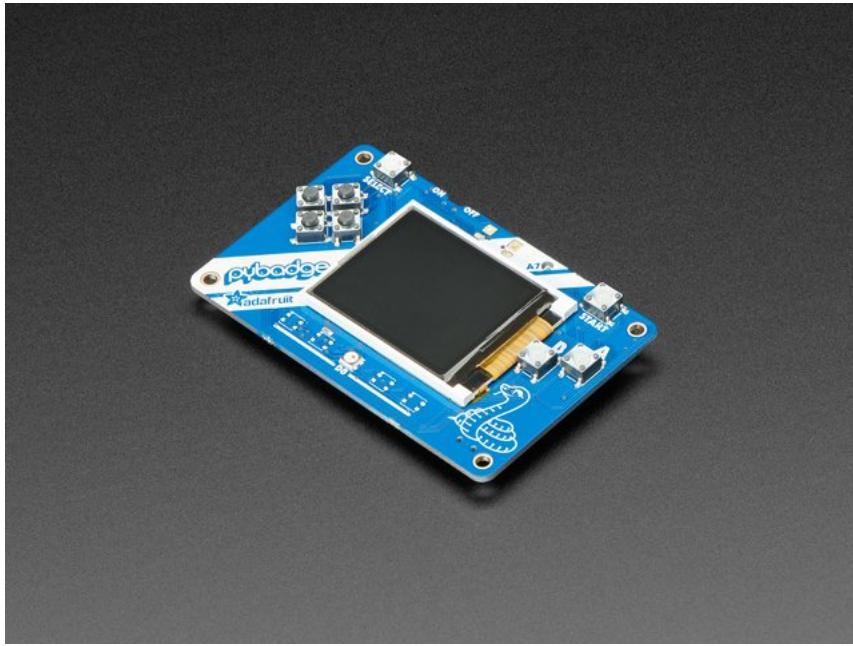
### [Adafruit PyBadge for MakeCode Arcade, CircuitPython or Arduino](#)

PRODUCT ID: 4200

What's the size of a credit card and can run CircuitPython, MakeCode Arcade or Arduino? That's right, its the Adafruit PyBadge! We wanted to see how much we...

OUT OF STOCK

[Out of Stock](#)



### [Adafruit PyBadge LC – MakeCode Arcade, CircuitPython or Arduino](#)

PRODUCT ID: 3939

What's the size of a credit card and can run CircuitPython, MakeCode Arcade or Arduino even when you're on a budget? That's right, it's the Adafruit...

OUT OF STOCK

[Out of Stock](#)

## Convert Your GIF

[ladyada](#)

The good news is that you do not have to do any particularly hairy transformations with your GIFs to get them to display – you don't have to make them into header files or anything. However, we do have to perform some steps to make them appear nicely on the TFT you're using

- Reduce size to no larger than 320 wide, 240 tall for PyPortal, and 160x128 for PyBadge or PyGamer (check your hardware to see the size you want to target)
- Remove transparent pixels (sometimes called *de-optimization*) because it slows down our decoding
- Reduce colors so we don't have to work as hard to decode
- Reduce frames if it makes sense to

This GIF viewer isn't good for detailed, high-frame-rate GIFs. Keep it simple and you'll be happy with the results!

## GIF by Example

Lets show an example GIF conversion so you can see how the process happens.

[We will convert this BadgerBadger GIF from Giphy](#)

You can download it via this zip link:

[badger.zip](#)

## Step one – Resize GIF

We'll use [ezgif.com](#) for these steps, its free and we find it works really well!

Click on **Resize**

A screenshot of the EZGIF.COM website. At the top, there is a navigation bar with links for Home, GIF Maker, Video to GIF, Resize (which is highlighted with a red arrow), Crop, and Gif. Below the navigation bar, there is a large input field labeled "Upload the original GIF and click Upload!" with the placeholder text "Or just drag &amp; drop your file here". At the bottom of the page, there is a footer with the text "EZGIF.COM ANIMATED GIFS MADE EASY".

Upload the original GIF and click **Upload!**

## Animated GIF resizer

Select image

Upload image from your computer:

badger.gif  
badger.gif

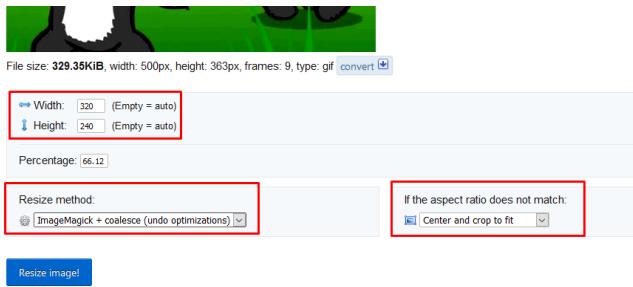
OR paste image url:

Supported image types: GIF, JPG, PNG, BMP, WebP, APNG, FLIF, MNG  
Max file size: 35MB

Now its uploaded,

- Set the **width** and **height** to be 320 and 240 (you can also set just one, as long as the other is no-greater than the max size)
- Use **ImageMagick + coalesce** as the resize method. **Don't use Gifsicle!**
- For the aspect ratio, you can stretch, center or force-aspect.

If you're not sure which one you want, keep going to resize and look at the preview, if you don't like it, change and resize again!



Click resize to get your preview! Check the width is no more than 320, and the height is no more than 240 pixels



You can use the GIF as is, but we found that you can do some optimizations on it to make smaller and faster-running

## Step 2 – Optimize

You can make a backup of the smaller GIF now. Then click **Optimize**



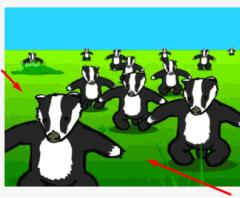
The first and easiest is color reduction and dithering. The fewer colors we have, the less decompression we have to do. Especially with a cartoony GIF like this, we can set the colors to be 32 and its not terribly noticeable



File size: 256.75KiB, width: 320px, height: 240px, frames: 9, type: gif [convert](#)

Optimization method (see description at bottom of the page):  
 Color Reduction + dither  
 Reduce colors to 32 (1 - 256)

The new GIF has banded colors on the grass but overall is pretty similar!



## Reduce Frames

Another very effective way to reduce the size and increase speed is to remove duplicate frames. Fewer frames means its easier to decode!

Try **Remove duplicate frames** and start with a **Fuzz** factor of around 25%, you can preview the GIF to see if you can fuzz higher (more savings but maybe it starts to look stuttery)

File size: 90.32KiB, width: 160px, height: 128px, frames: 9, type: gif [convert](#)

Optimization method (see description at bottom of the page):  
 Remove duplicate frames  
 Fuzz %:  (defines how similar colors can be considered equal)

[Optimize GIF](#)

Whatever you do, **don't add transparency** because it will make the file smaller but the decoding bigger!

OK here's our new final GIF:



[optimized\\_badger.zip](#)

## Quickstart

[ladyada](#)

You can get started instantly with these UF2's – doubleclick Reset to put your board into **BOOT** mode, and drag these **UF2** files over!

If you are using SD card storage, make a directory called **gifs** on the SD card and put your GIFs in there

If you are using QSPI storage, and at some point you had CircuitPython loaded, use a USB cable, click reset to have it show up as a disk drive. Make a directory called **gifs** on the SD card and put your GIFs in there. If you've never loaded CircuitPython, check the Loading GIFs page up ahead.

For boards that have both SD and QSPI storage, the gif player will default to SD if its found, and if not, fall back to QSPI

## PyBadge or EdgeBadge using QSPI for storage

[PYBADGE\\_QSPI.UF2](#)

<https://learn.adafruit.com/pyportal-animated-gif-display?view=all>

## PyGamer using QSPI or SD for storage

[PYGAMER\\_GIFPLAYER.UF2](#)

## PyPortal using QSPI or SD for storage

[PYPORAL\\_SD.UF2](#)

[PYPORAL\\_QSPI.UF2](#)

## CLUE using QSPI for storage

[CLUE\\_gifplayer.UF2](#)

### Loading GIFs

[ladyada](#)

### Getting GIFs onto the Filesystem

Alright so now you want your own favorite animations to show up, right? The fun part is getting those files onto your Arcada board. There's two options

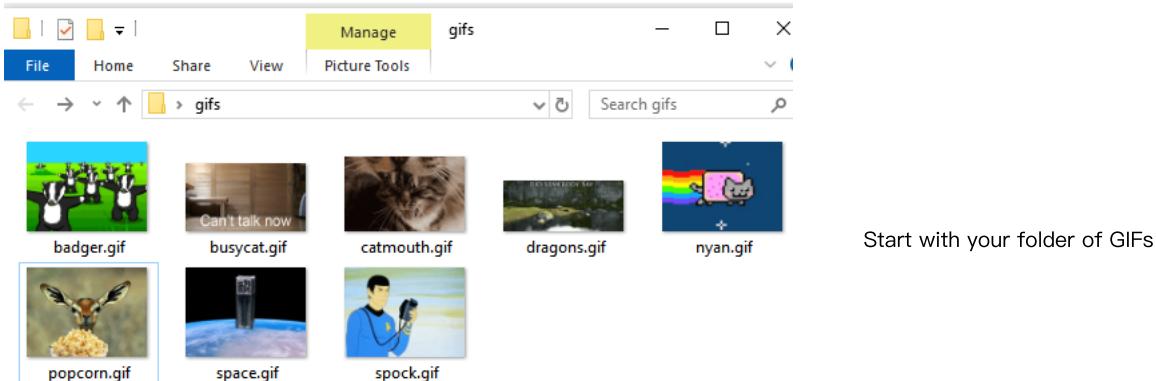
- In the previous step, select using SD card storage UF2 then create a folder called `/gifs` on an SD card that is FAT formatted, and put your GIFs in there. Pop it into your SD card slot and you're good to go!
- Use the default QSPI flash storage, which you can access via the computer USB port if you use our pre-compiled code or compile with TinyUSB support

### Format QSPI with CircuitPython

For QSPI, you need to have it 'formatted' as a CircuitPython filesystem. The easiest way to do that is to load CircuitPython. You only have to do this once, then you can load more gifs or remove them over USB

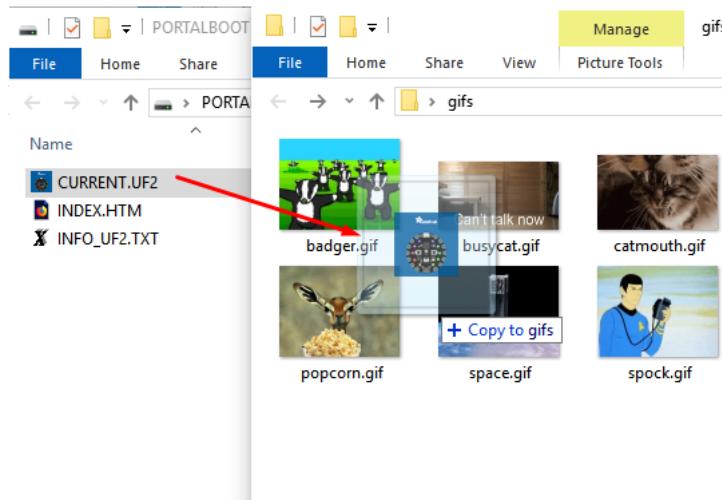
(If you have an SD Card just format it with an SD card writer/reader on your computer)

You only have to do this process once to format it, but its a fast way to get your board set up and create the `gifs` directory + maybe add a few gifs



With your board running the animated GIF Arduino code, plug it into USB and then double-click to launch the bootloader.

Grab **CURRENT.UF2** from the `xxxxBOOT` drive and drag onto your `gifs` folder on your computer, making a backup of your Arduino firmware



## PyPortal

by Adafruit



This board hasn't been fully documented yet. Please make a pull request adding more info to this file.

The description should be written to inform a CircuitPython user what makes the board unique and link to relevant info about it.

### CircuitPython 4.0.0-beta.5

This is the latest unstable release of CircuitPython that will work with the PyPortal.

Unstable builds have the latest features but are more likely to have critical bugs.

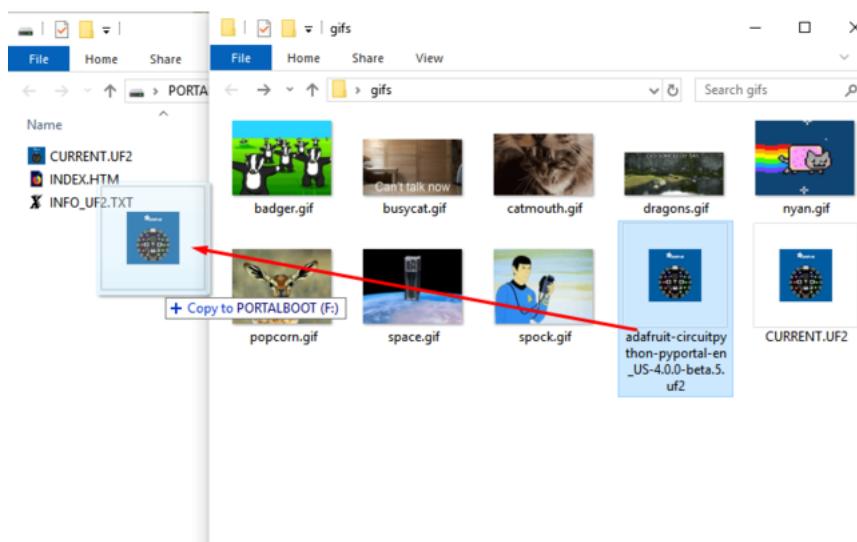
Visit <https://circuitpython.org/downloads> and download the latest firmware for your board as a UF2

ENGLISH

DOWNLOAD .UF2 NOW

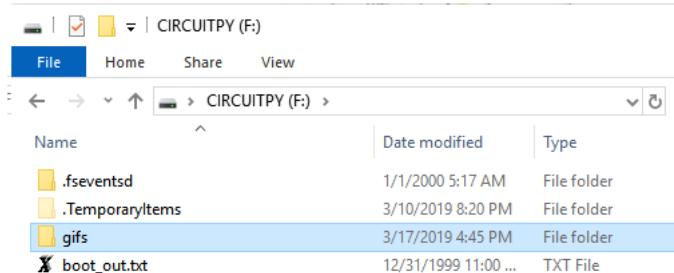
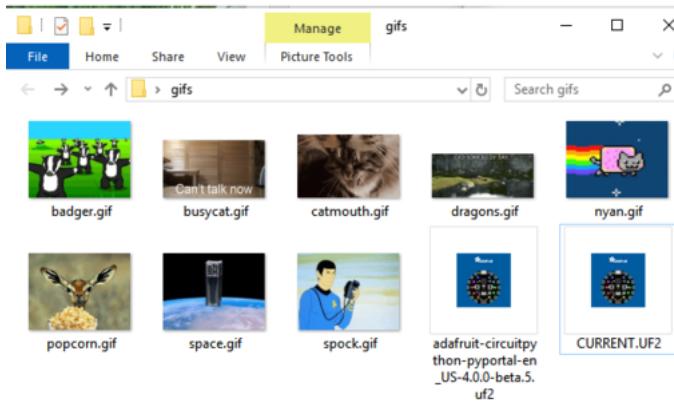
### Absolute Newest

Every time we commit new code to CircuitPython we automatically build binaries for it. They are stored on Amazon S3 by language



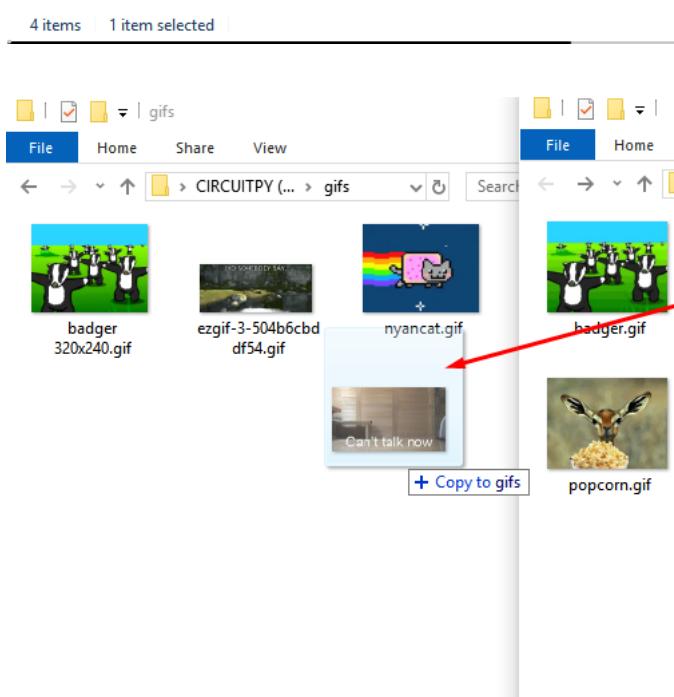
Now drag the `circuitpython.UF2` onto `xxxxBOOT`, which will replace the Arduino code with CircuitPython runtime.

Your `gifs` folder now has two UF2s – `CURRENT.UF2` and `circuitpython.UF2` (with the version info)



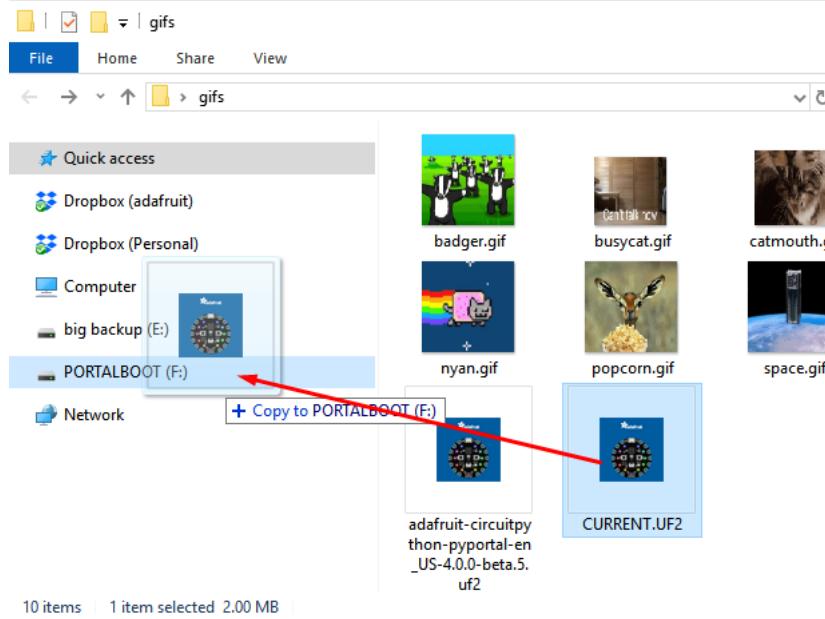
In a few moments you'll get a **CIRCUITPY** drive, this may contain circuitpython code, files, etc. This is the 8 MB Flash storage on the board, available as a disk drive

Now's a good time to create a folder called **gifs** on the **CIRCUITPY** drive if you have not done so



Drag all the GIFs you want to display into the **CIRCUITPY/gifs** folder

Once you're done, you can re-load the GIF code by double-clicking the reset button again, to launch the **PORTALBOOT** bootloader drive, then dragging the backup you made before, **CURRENT.UF2** back on.



## Loading GIFs onto QSPI / SD over USB

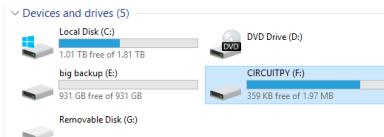
Arduino doesn't have the ability to access both the SD and QSPI at the same time (because they both have a File type and are incompatible) so we can't copy files from SD to QSPI easily.

However Arduino can be convinced to show the QSPI or SD as USB mass storage, so you can have it show up as a disk drive.

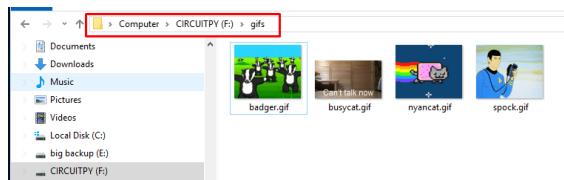
For QSPI, you need to have it 'formatted' as a CircuitPython filesystem. The easiest way to do that is to load CircuitPython. You only have to do this once, as shown above.

For SD, format using any USB SD card formatter dongle.

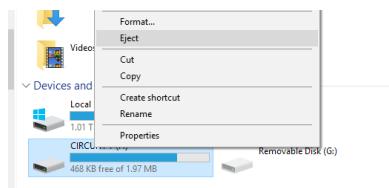
You must have the board plugged into USB **with a Data/Sync Cable** then press the reset button. You should see a disk drive appear. It may be called CIRCUITPY OR USB Disk (you can rename it if you like)



You can explore the disk, if it doesn't yet contain a **gifs** folder, please make one now – then put a bunch of gifs in it!



When you're done, be sure to **Eject** the disk – it won't disappear but it will flush/save all your gif data so it doesn't get corrupted



## Arcada Libraries

[ladyada](#)

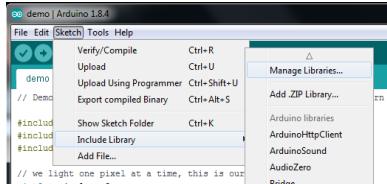
OK now that you have Arduino IDE set up, drivers installed if necessary and you've practiced uploading code, you can start installing all the Libraries we'll be using to program it.

There's a lot of libraries!

## Install Libraries

<https://learn.adafruit.com/pyportal-animated-gif-display?view=all>

Open up the library manager...



And install the following libraries:

## Adafruit Arcada

This library generalizes the hardware for you so you can read the joystick, draw to the display, read files, etc. without having to worry about the underlying methods

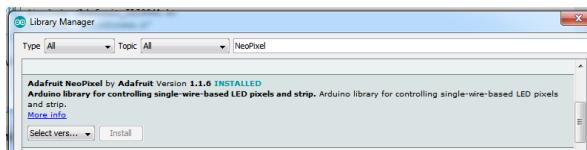


If you use Arduino 1.8.10 or later, the IDE will automagically install all the libraries you need to run all the Arcada demos when you install Arcada. We strongly recommend using the latest IDE so you don't miss one of the libraries!

**If you aren't running Arduino IDE 1.8.10 or later, you'll need to install *all* of the following!**

## Adafruit NeoPixel

This will let you light up the status LEDs on the front/back



## Adafruit FreeTouch

This is the open source version of QTouch for SAMD21 boards



## Adafruit Touchscreen

Used by Adafruit Arcada for touchscreen input (required even if your Arcada board does not have a touchscreen)



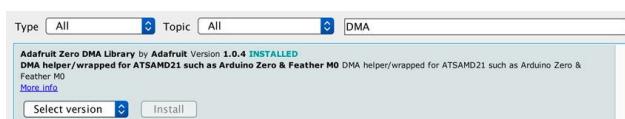
## Adafruit SPIFlash

This will let you read/write to the onboard FLASH memory with super-fast QSPI support



## Adafruit Zero DMA

This is used by the Graphics Library if you choose to use DMA



## Adafruit GFX

This is the graphics library used to draw to the screen



## Adafruit ST7735

The display on the PyBadge/PyGamer & other Arcada boards



## Adafruit ILI9341

The display on the PyPortal & other Arcada boards



## Adafruit LIS3DH

For reading the accelerometer data, required even if one is not on the board



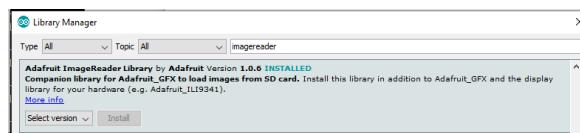
## Adafruit Sensor

Needed by the LIS3DH Library, required even if one is not on the board



## Adafruit ImageReader

For reading bitmaps from SPI Flash or SD and displaying



## ArduinoJson

We use this library to read and write configuration files



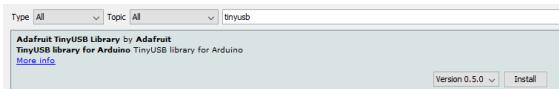
## Adafruit ZeroTimer

We use this library to easily set timers and callbacks on the SAMD processors



## Adafruit TinyUSB

This lets us do cool stuff with USB like show up as a Keyboard or Disk Drive



## Adafruit WavePlayer

Helps us play .WAV sound files.



## SdFat (Adafruit Fork)

The Adafruit fork of the really excellent SD card library that gives a lot more capability than the default SD library



## Audio – Adafruit Fork

Our fork of the Audio library provides a toolkit for building streaming audio projects.



## Build In Arduino

[ladyada](#)

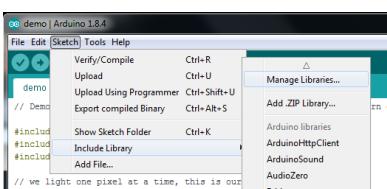
## Compiling the Code

If you don't want to use the pre-compiled code, you can build it yourself to load onto your Arcada board. This project requires using the Arduino IDE because it needs every bit of computational energy!

[Visit the PyPortal Arduino Setup page to install the Arduino IDE \(or update it\), install PyPortal support](#) or ditto for whatever hardware you're using. [Also make sure you installed all the Arduino Libraries](#) (and you need quite a lot!)

## Download the Animated GIF Demo code

Open up the Arduino library manager



Search for the Adafruit Arcada GifDecoder library and install it



Verify you have the [latest version of all the libraries on this page](#) via the Arduino Library Manager. The code will not compile with earlier versions.

We also have a great tutorial on Arduino library installation at:

<http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use>

## Configuring SdFat

If you plan to use the SD card to play GIFs you'll need to update `sdfatconfig.h` in the SdFat library you installed, and change the following #define's:

Download: [file](#)

[Copy Code](#)

```

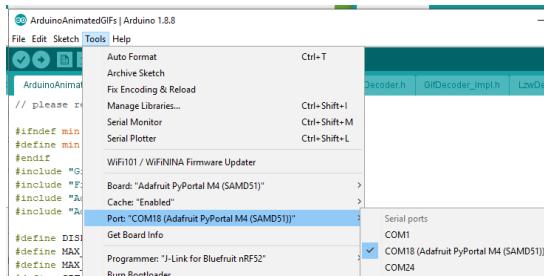
1. //-----
2. /**
3.  * If the symbol ENABLE_EXTENDED_TRANSFER_CLASS is nonzero, the class SdFatEX
4.  * will be defined. If the symbol ENABLE_SOFTWARE_SPI_CLASS is also nonzero,
5.  * the class SdFatSoftSpiEX will be defined.
6. *
7.  * These classes used extended multi-block SD I/O for better performance.
8.  * the SPI bus may not be shared with other devices in this mode.
9. */
10. #define ENABLE_EXTENDED_TRANSFER_CLASS 1
11. //-----
12. /**
13.  * If the symbol USE_STANDARD_SPI_LIBRARY is zero, an optimized custom SPI
14.  * driver is used if it exists. If the symbol USE_STANDARD_SPI_LIBRARY is
15.  * one, the standard Arduino SPI.h library is used with SPI. If the symbol
16.  * USE_STANDARD_SPI_LIBRARY is two, the SPI port can be selected with the
17.  * constructors SdFat(SPIClass* spiPort) and SdFatEX(SPIClass* spiPort).
18. */
19. #define USE_STANDARD_SPI_LIBRARY 2

```

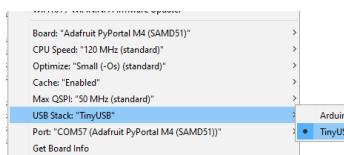
## Initial Test

Let's make sure you can get the basics going before we customize it

Make sure you have the **Adafruit PyPortal** type of board selected or whatever board you're looking to use, and pick the right COM port. It may or may not have Adafruit PyPortal next to the serial port name. If you can't find it, double-click the reset button to put it into bootloader mode, then select the serial port when it appears.

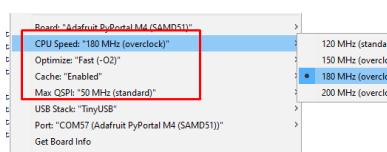


Under USB Stack select **TinyUSB** – this will let the board show the filesystem up as a disk drive



In order to really be able to display GIFs at full speed, we need to overclock a bit... Make sure that before you continue, you have the following selected:

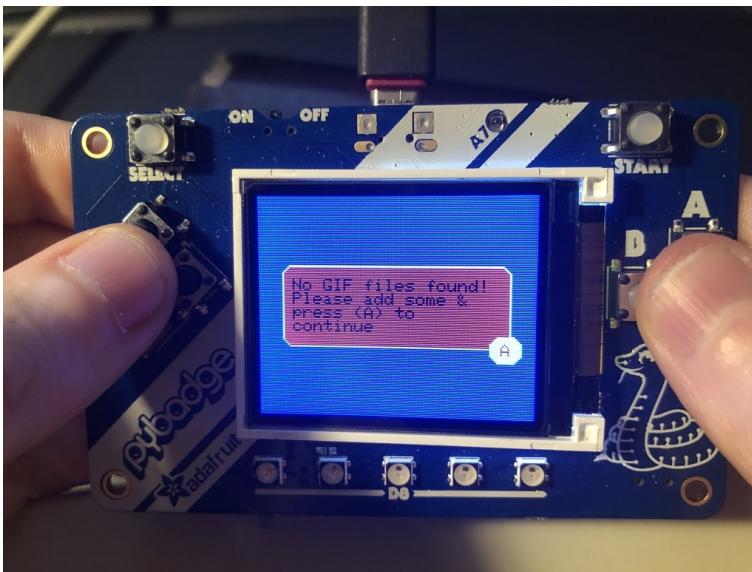
- CPU Speed: 180 MHz (you can also try 200)
- Optimize: fast
- Cache: Enabled



Click **Upload** to upload the sketch to the PyPortal

```
[=====] 99% (886/887 pages) checksumBuffer[start_addr=0x40c00, size=0xf8] = e0
read(addr=0x40000, size=0x200)
[=====] 100% (887/887 pages)
Verify successful
Done in 1.663 seconds
writeWord(addr=0xe000ed0c, value=0x5fa0004)
```

You may get a notice that there's not configuration file found, and also likely



## Customization

[lady ada](#)

There are a few settings you can change

### Loop Time & Brightness

First is how long to display a GIF, second is how bright the backlight is. We can display multiple images, by default waiting 10 seconds until continuing to the next one. You can change this by creating or replacing a text file called `arcada_config.json` in the `root` directory of the QSPI flash or SD card. If you have the Arcada board plugged in and press reset, it will appear in the filesystem for you to access

Here's an example of the contents of the file you can use

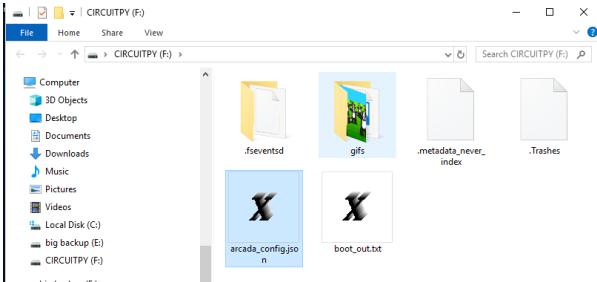
Download: [file](#)

[Copy Code](#)

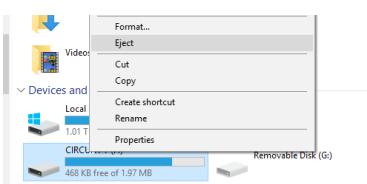
```
1. {"volume":255,"brightness":255,"seconds_per_gif":5}
2.
```

`volume` is not used in this project. `brightness` ranges from 0 (display off) to 255 (all the way max brightness). `seconds_per_gif` is what you expect

And here's where it resides, remember its *not in the `gifs` folder!* Place it in the 'root' directory



When you're done, be sure to **Eject** the disk – it won't disappear but it will flush/save your config data so it doesn't get corrupted



## Storage Locations

Of course you want to change the built in GIFs! You can store GIFs in *one of two places* – either an SD card you insert **OR** on the internal QSPI Flash

You cannot pick or choose in one compiled version due to some unavoidable constraints on how Arduino handles Files – so you have to pick one!

By default we use QSPI, because almost all our boards have it built in for free.

However, if you'd like to use SD cards for storage, go to the `Adafruit_Arcada` library and look in `Adafruit_Arcada.h` for a section starting with `#elif defined(ADAFRUIT_PYPORTAL)`

```
#elif defined(ADAFRUIT_PYPORTAL)
// 8 bit 320x40 TFT
#define ARCPADA_DC 34 // Data bit 0 pin (MUST be on PORT byte boundary)
#define ARCPADA_TFT_CS 26 // Write-Enable pin (CCL-inverted timer output)
#define ARCPADA_TFT_DC 10 // Data/command pin
#define ARCPADA_TFT_CS 11 // Chip-select pin
#define ARCPADA_TFT_RST 24 // Reset pin
#define ARCPADA_TFT_RD 9 // Read-strobe pin
#define ARCPADA_TFT_LITE 25
#define ARCPADA_TFT_DEFAULTFILL 0xFFFF
#define ARCPADA_TFT_ROTATION 9
#define ARCPADA_TFT_WIDTH 320
#define ARCPADA_TFT_HEIGHT 240

#define ARCPADA_TFT_TYPE Adafruit_ILI9341
#define ARCPADA_TFT_INIT begin()

#define ARCPADA_SPEAKER_ENABLE 50

#define ARCPADA_NEOPIXEL_PIN 2
#define ARCPADA_NEOPIXEL_NUM 1

#define ARCPADA_SD_CS 32
#define ARCPADA_USE_QSPI_FS
//##define ARCPADA_USE_SD_FS

#define ARCPADA_ACCEL_TYPE ARCPADA_ACCEL_NONE
```



Find the lines within that section that look like:

Download: [file](#)

[Copy Code](#)

1. `#define ARCPADA_USE_QSPI_FS`
2. `//##define ARCPADA_USE_SD_FS`

And change it to

Download: [file](#)

[Copy Code](#)

1. `//##define ARCPADA_USE_QSPI_FS`
2. `#define ARCPADA_USE_SD_FS`

Then save and re-upload

Displaying GIFs from SD cards is easier to move big files on and off the device because you can use an SD card reader, but they will display slower than when stored on the QSPI internal memory! That's because QSPI is a much faster storage system, we can read the data a few seconds faster!

## Example GIFs

[Ruiz Brothers](#)

## Animation Demos

Looking for some cool GIFs to get the creative juices flowing? We put together some neat ones we think you'll like. Download the zip file and drop the files into your GIFs folder.

- `blinka-wipe.gif` — Blinka transition for project videos
- `cpython-logo.gif` — The circuit python logo
- `mu-editor.gif` — Our favorite python editor
- `pyportal-blinka.gif` — PyPortal logo featuring Blinka

[GIFs.zip](#)

And here's a pack with them resized to 160x128 for 1.8" Displays

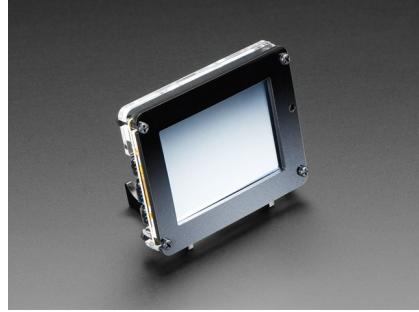
[160x128\\_GIFs.zip](#)

This guide was first published on Mar 20, 2019. It was last updated on Mar 20, 2019.



[Adafruit PyPortal – CircuitPython Powered Internet Display](#)

\$54.95

[Add to Cart](#)[Adafruit PyPortal Desktop Stand Enclosure Kit](#)

\$9.95

[Add to Cart](#)[Adafruit PyGamer Starter Kit](#)[Out of Stock](#)[Adafruit PyGamer for MakeCode Arcade, CircuitPython or Arduino](#)[Out of Stock](#)[Adafruit PyBadge for MakeCode Arcade, CircuitPython or Arduino](#)[Out of Stock](#)[Adafruit PyBadge LC – MakeCode Arcade, CircuitPython or Arduino](#)[Out of Stock](#)



[Adafruit EdgeBadge – TensorFlow Lite for Microcontrollers](#)

Out of Stock

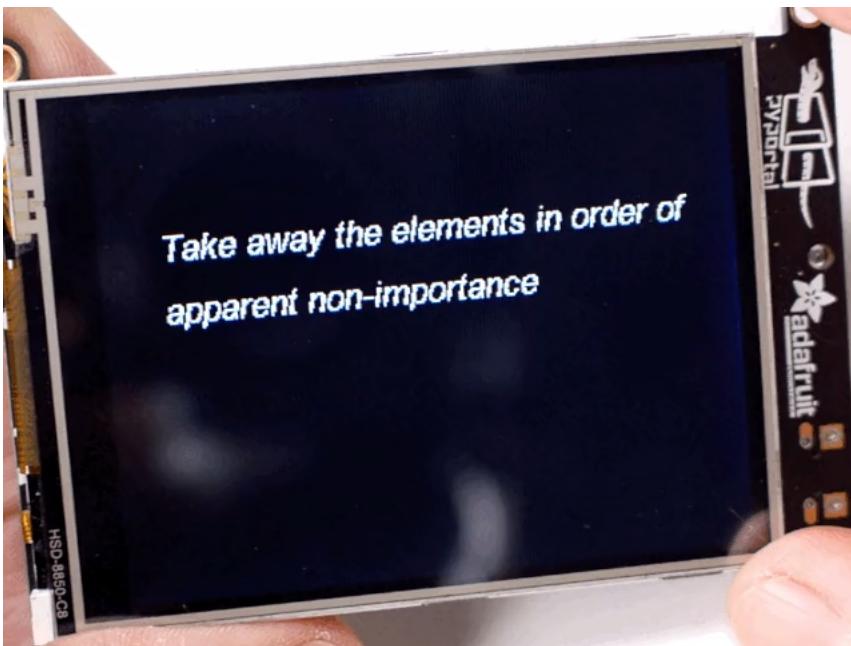


[USB cable – USB A to Micro-B](#)

\$2.95

[Add to Cart](#)

## Related Guides



[A virtual deck of creative suggestions for artists & composers](#)

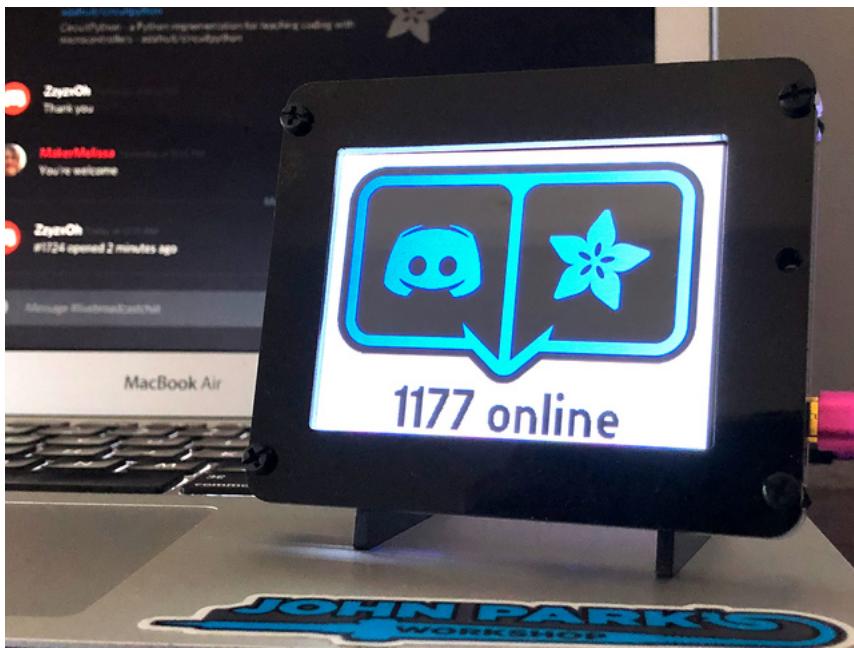
Collin Cunningham



[PyPortal Oblique Strategies](#)

4

Beginner



[Use the PyPortal to see how many Discord users are on a favorite server!](#)

[John Park](#)



#### [PyPortal Discord Online Counter](#)

2

Beginner



[Keep track of your Twitter follower count with this lovely trophy!](#)

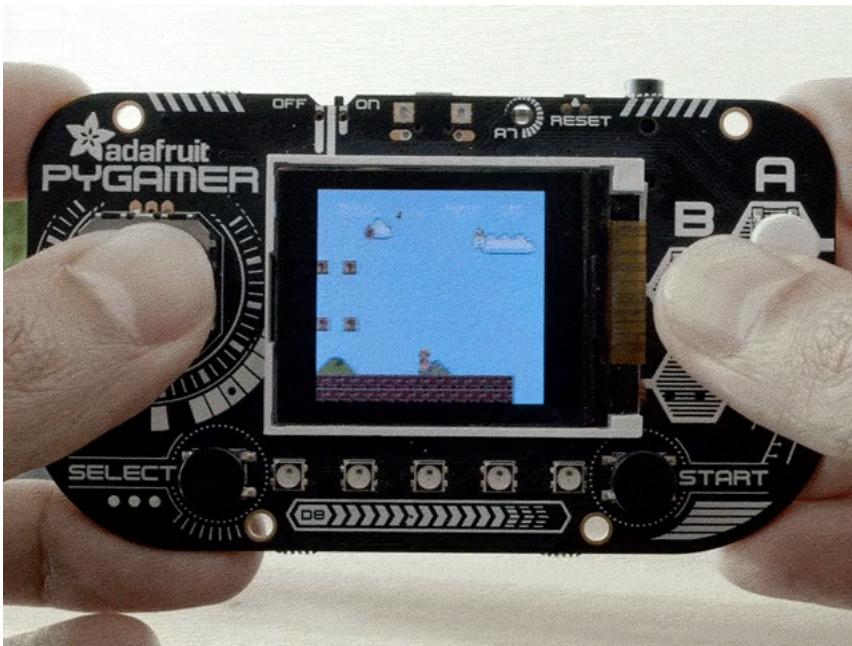
[John Park](#)



#### [PyPortal Twitter Follows Trophy](#)

2

Beginner



[Play your favorite retro games on Adafruit Arcada boards](#)

[lady\\_ada](#)



### [NES Emulator for Arcada](#)

12

Beginner



### OUT OF STOCK NOTIFICATION

YOUR NAME

YOUR EMAIL

[NOTIFY ME](#)

## Help

- [Contact Us](#)
- [Tech Support Forums](#)
- [FAQs](#)
- [Shipping & Returns](#)
- [Terms of Service](#)
- [Privacy & Legal](#)

## Explore

- [About Us](#)
- [Press](#)
- [Educators](#)
- [Distributors](#)
- [Jobs](#)
- [Gift Cards](#)

"To be persuasive we must be believable; to be believable we must be credible; credible we must be truthful."

[Edward R. Murrow](#)

