Creating Slideshows in CircuitPython

Created by Anne Barela

https://learn.adafruit.com/creating-slideshows-in-circuitpython

Last updated on 2023-08-29 04:13:42 PM EDT

Table of Contents

Overview 3

- Parts

Setup CircuitPython 5

- Set up CircuitPython Quick Start!
- Issues?

Load the Software 7

- Download the Library
- Get the Code!

Load Your Pictures 9

- Copying the Pictures to Your Device
- Sample Pictures

Use It 11

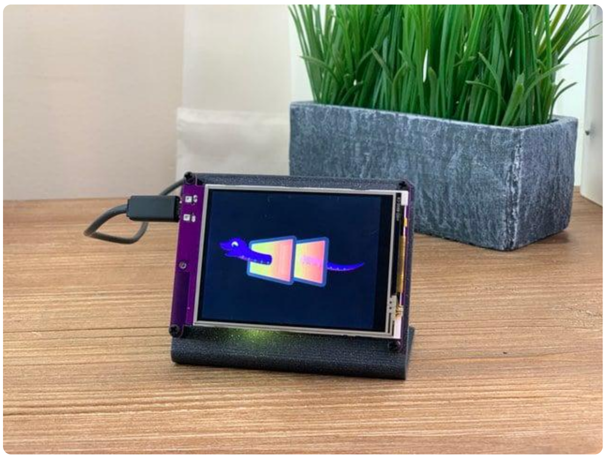
- Customization of the Slideshow

Overview

Pictures capture our memories, our fandom, and the world around us. But they do little good sitting on your hard drive. Grab a CircuitPython board and connected screen and display your images in a slideshow with only a few lines of CircuitPython code.

Get up and running in a few minutes. No hardware assembly and all free software.

Parts



Adafruit PyPortal - CircuitPython Powered Internet Display

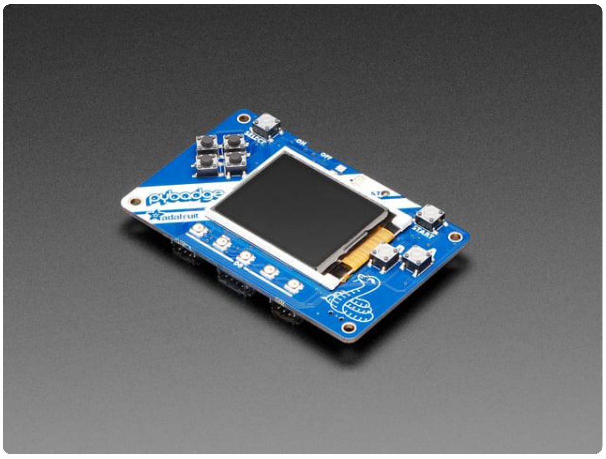
PyPortal, our easy-to-use IoT device that allows you to create all the things for the “Internet of Things” in minutes. Make custom touch screen interface...

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

Adafruit PyGamer for MakeCode Arcade, CircuitPython or Arduino

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...

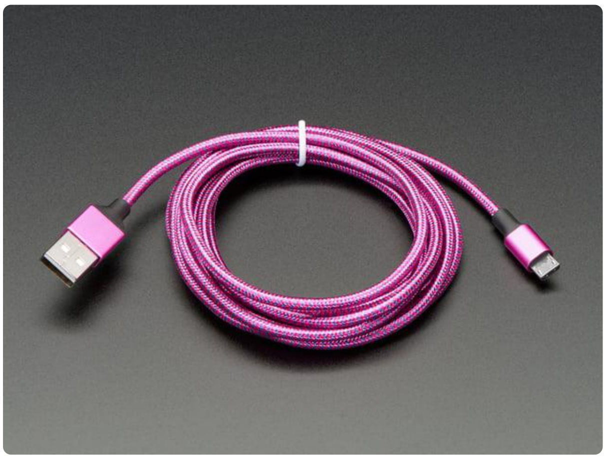
https://www.adafruit.com/product/4242



Adafruit PyBadge for MakeCode Arcade, CircuitPython, or Arduino

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...

https://www.adafruit.com/product/4200



Pink and Purple Braided USB A to Micro B Cable - 2 meter long

This cable is super-fashionable with a woven pink and purple Blinka-like pattern! First let's talk about the cover and over-molding. We got these in custom colors,...

https://www.adafruit.com/product/4148

Setup CircuitPython

CircuitPython () is a derivative of MicroPython () 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!

It is suggested that if your board already has CircuitPython on it that you download and update to the latest stable version to ensure bugs are quashed and any new features are available.

Set up CircuitPython Quick Start!

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

Go to CircuitPython.Org/Downloads () to find your CircuitPython compatible board.

Download the latest stable version of CircuitPython for your board. Save the .UF2 file to some known place on your computer.

Note the major version of CircuitPython you are downloading (the first number). As of this guide that is CircuitPython 4 although CircuitPython 5 is in alpha testing and may be the stable version when you get to this guide. Or perhaps CircuitPython 11? Just note that number for the next page of this guide.

Plug your board 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 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 may 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, ending in BOOT, like PYPORTALBOOT.

Drag the adafruit-circuitpython--.uf2 file to the <boardname>BOOT drive.

The LED will flash. Then, the BOOT 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! That file is a text file listing the version of CircuitPython which can be handy.

It's time for you to add your code.py and get started!

Issues?

See this page () on FAQ with troubleshooting pages following.

Load the Software

Your board will need two code files to work:

1. A library containing the slideshow code
2. Your program, named code.py, which calls the slideshow library

All of the nitty-gritty is coded in the library, making working with the code super simple.

Even if your board has libraries in the /lib folder, considering getting the latest version from circuitpython.org as there may be bug fixes and additional features. And you'll ensure the version of the library matches the version of CircuitPython you are running.

Download the Library

On the previous page you did note the version of CircuitPython you downloaded, yes?

Good. Go to CircuitPython.Org/libraries () and you'll look to download the library bundle matching the major version of CircuitPython you installed.

Save the file to your computer (not on the CIRCUITPY drive, please).

For this project, only one library file is needed!

Ensure your board is connected to your computer via a known good USB cable. Use your operating system file explorer/finder to go to the CIRCUITPY drive. Create a new folder named lib. We'll put the library we need in this folder.

Now use your file explorer/finder to go to where you saved the library bundle zip file. Open the file and find the file named adafruit_slideshow.mpy. Copy that file into the lib folder on the CIRCUITPY drive you just made. Now you're set.

Get the Code!

The code for the project is below. Click any of the save options and save to your computer. The file you want is named code.py, the default run name for CircuitPython code on a CircuitPython device.

```
# SPDX-FileCopyrightText: 2019 Anne Barela for Adafruit Industries
#
# SPDX-License-Identifier: MIT

# CircuitPython Slideshow - uses the adafruit_slideshow.mpy library
import board
from adafruit_slideshow import PlayBackOrder, SlideShow

# Create the slideshow object that plays through once alphabetically.
slideshow = SlideShow(board.DISPLAY,
                      folder="/images",
                      loop=True,
                      order=PlayBackOrder.ALPHABETICAL,
                      dwell=60)

while slideshow.update():
    pass
```

That's it - rather short, yes? Adafruit has made the library do all the heavy lifting.

There are customizations that may be done, they will be explored after getting pictures displaying on the screen.

Load Your Pictures

As is, the code has no pictures on the CIRCUITPY drive to use in your slideshow. Let's get some.

While the slideshow code is short and simple, know that microcontrollers, unlike computers, don't have the greatest amount of speed and memory. So any scaling of images and conversion of graphics formats should be done on your computer before they are copied to your device.

All images must be in .bmp format.

The size of the image depends on the size of your display. Here are some dimensions, if there is a board without an entry, go to the Adafruit product page to see the specifications for the display and use those.

- PyPortal - 320x240 pixels (landscape)
- PyBadge and PyBadge LC - 160x128 pixels (landscape)
- PyGamer - 160x128 pixels (landscape)
- Hallowing M0 Express - 128x128 pixels (portrait)
- Monster M4sk and HalloWing M4 Express - 240x240 pixels (square)

The image editing software available is dependent on the operating system you are using. The GNU program GIMP runs on all operating systems and is free.

In your editing software, open the image you want in the slideshow. Resize the image to the demensions for your device. Then save the image in BMP format in a folder on your computer.

The slideshow code is set to show the pictures in alphabetical order. It is suggested you save the pictures sequentially, re. the first one is saved as 001.bmp, the second as 002.bmp, etc., this will allow for 999 images you can display in order.

You can name your files any names you want (keep the filenames from being absurdly long though). Just know they'll be run in alphabetical order. If you would like random pictures, you can change that later in this guide.

Copying the Pictures to Your Device

In your computer's file explorer/finder, create a new directory on your CIRCUITPY drive names images.

Go to the folder you copied your resized BMP images into and copy the files to your CIRCUITPY\images directory. Once done, the device should reset and run your slideshow!

Feel free to reorder your images by changing filenames, if necessary.

Sample Pictures

Below are sets of 6 sample pictures, formatted for the devices listed above.

What happens when you use larger images than the display resolution? You'll see as much of the picture as the display can render, say 160x128 pixels out of a larger 320x240 picture. At this time, CircuitPython cannot change the resolutions of files for display. Perhaps one day.

BMP Image Pack in 128x128 pixels

BMP Image Pack in 160x128 pixels

BMP Image Pack in 240x240 pixels

BMP Image Pack in 320x240 pixels

These images are from Unsplash () and come with an unlimited, free license.

Original JPG Files from Unsplash

Use It

The basic code will present a new picture each 60 seconds.

The image below has been set for 3 seconds between pictures to better show a selection of images.

If you get an error "no module named adafruit_slideshow", go back to the "Load the Software" Page and load the required library noted in the "Install the Library" section.

Customization of the Slideshow

The adafruit_slideshow library function has several parameters that can be changed. The latest documentation for parameters is on ReadTheDocs ().

Let's look at the code again:

```
# SPDX-FileCopyrightText: 2019 Anne Barela for Adafruit Industries
#
# SPDX-License-Identifier: MIT

# CircuitPython Slideshow - uses the adafruit_slideshow.mpy library
import board
from adafruit_slideshow import PlayBackOrder, SlideShow
```

```
# Create the slideshow object that plays through once alphabetically.
slideshow = SlideShow(board.DISPLAY,
                      folder="/images",
                      loop=True,
                      order=PlayBackOrder.ALPHABETICAL,
                      dwell=60)

while slideshow.update():
    pass
```

board.DISPLAY is common to all boards and is required. If there is an error, it probably is that your board does not have a predefined display. If you are an advanced user and hook your own display up, this will need to be defined.

folder = "/images" sets the image folder to the CIRCUITPY /images directory that we used. You can specify other directories.

loop = True has the images looping back to the starting image when the last image completes displaying.

order = PlayBackOrder.ALPHABETICAL plays the images in alphabetical order - the only other option is order=PlayBackOrder.RANDOM

dwell = 60 is a parameter you may want to change. This is the amount of time a picture is on the screen in seconds. I think a minute (60 second) is a bit long, perhaps you want 30 seconds for a fast display or 3600 seconds for an hour.

For all the available parameters:

class `adafruit_slideshow.` `SlideShow`(display, backlight_pwm=None, *, folder='/', order=0, loop=True,
dwell=3, fade_effect=True, auto_advance=True, direction=1)

Parameters:

- folder (str ()) – Specify the folder containing the image files, in quotes. Default is the root directory, `"/"`.
- order (PlayBackOrder ()) – The order in which the images display. You can choose random (`RANDOM`) or alphabetical (`ALPHABETICAL`). Default is `ALPHABETICAL`.
- loop (bool ()) – Specify whether to loop the images or play through the list once. `True` () if slideshow will continue to loop, `False` if it will play only once. Default is `True`.
- dwell (int ()) – The number of seconds each image displays, in seconds. Default is 3.

- fade_effect (bool ()) – Specify whether to include the fade effect between images. `True` tells the code to fade the backlight up and down between image display transitions. `False` maintains max brightness on the backlight between image transitions. Default is `True`.
- auto_advance (bool ()) – Specify whether to automatically advance after dwell seconds. `True` if slideshow should auto play, `False` if you want to control advancement manually. Default is `True`.
- direction (PlayBackDirection ()) – The playback direction. 1 = forward, -1 backward.