Set up Web Workflow on the Adafruit MEMENTO

Created by Melissa LeBlanc-Williams

https://learn.adafruit.com/set-up-web-workflow-on-the-adafruit-memento

Last updated on 2024-06-03 03:58:48 PM EDT

Table of Contents

Overview 3

- Parts

Device Setup 5

- Creating a settings.toml File
- Disabling USB Mass Storage

Usage 7

- Install Fancy Camera Example
- Install an SD Card
- Put in Timelapse Mode
- Accessing the Photos
- Downloading a Single Photo
- Downloading All Photos

Overview

Web Workflow is an excellent way to update your MEMENTO camera board remotely. If your project involves keeping the MEMENTO in a specific position or perhaps it is mounted in a difficult to reach place, then editing your code with Web Workflow will be essential.

This guide focuses on getting you up and running with Web Workflow specifically for the MEMENTO. If you need a guide with a more general focus on the editor itself, check out the CircuitPython Web Workflow Code Editor Quick Start (https://adafru.it/19Fy) guide. Web Workflow is baked into CircuitPython itself, so there's nothing additional to install and it's only a matter of setting up a few things.

This guide will show you an example of using the Web Workflow with time lapse photography. If you would like to mount the camera to a tripod other by another means that requires a 1/4"-20 thread and you have a 3D printer available, you may want to check out the https://learn.adafruit.com/memento-3d-case/3d-printing (https://adafru.it/19Fz) guide, which can make timelapse photography much easier.

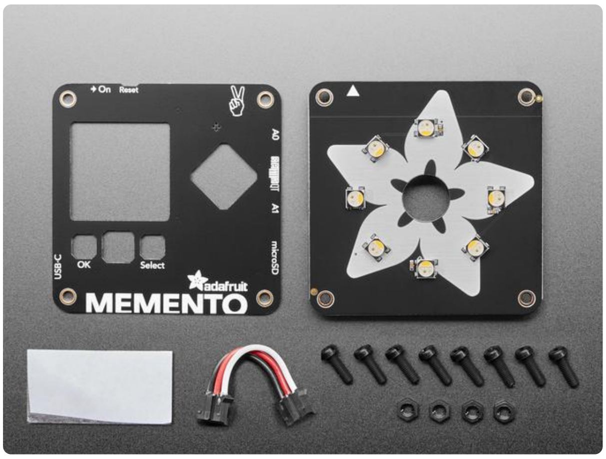
The guide will show you one use case of Web Workflow that is more specific to the Adafruit MEMENTO. If you do any timelapse phtography, you may wish to test that images are being saved. This is particularly helpful if you have a long timelapse going or are in low power mode with the preview window off.

Parts

MEMENTO - Python Programmable DIY Camera - Bare Board

Make memories, or just a cool camera-based project, with Adafruit's MEMENTO Camera Board. It's a development board with everything you need to create...

https://www.adafruit.com/product/5420



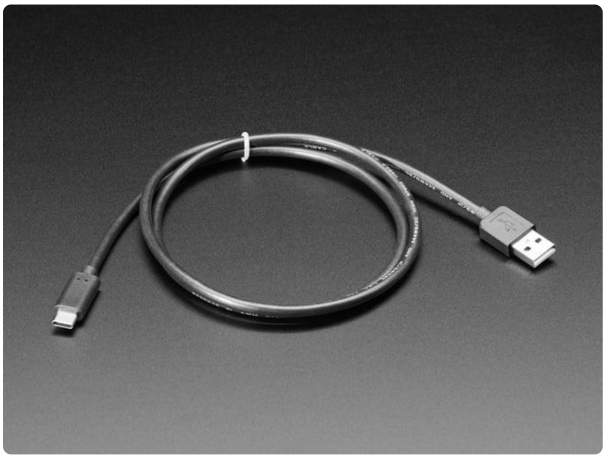
Adafruit MEMENTO Camera Enclosure & Hardware Kit

Once you've picked up your MEMENTO Camera and you're ready to take it out into the world, here is a chic and minimalist enclosure that will look great on the...

https://www.adafruit.com/product/5843

ADABOX 021 - MEMENTO Python Programmable DIY Camera

https://www.adafruit.com/product/5419



USB Type A to Type C Cable - approx 1 meter / 3 ft long

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

Device Setup

Make sure you are running **version 9.0.0-rc.2 or later** of CircuitPython. This release includes a fix for Web Workflow. You can download it from the downloads page on circuitpython.org (https://adafru.it/18e6). Once you have that downloaded, you will want to check out the Adafruit MEMENTO Camera Board (https://adafru.it/19ja) guide for installation instructions.

If you have previously installed an earlier version, you will need to backup your data and erase the filesystem, as it was previously formatted with the incorrect size. This can be achieved by connecting to the terminal and running the following commands:

```
import storage
storage.erase_filesystem()
```

Creating a **settings.toml** File

Once you have the correct version of CircuitPython installed and running, the next step is to create an **settings.toml** file in the root of the **CIRCUITPY** drive. Then configure it by adding the following contents and changing the values as appropriate:

```
# To auto-connect to Wi-Fi
CIRCUITPY_WIFI_SSID="yourwifissid"
CIRCUITPY_WIFI_PASSWORD="yourpassword"

# To enable modifying files from the web. Change this too!
# Leave the User field blank when you type the password into the browser.
CIRCUITPY_WEB_API_PASSWORD="passw0rd"

CIRCUITPY_WEB_API_PORT=80
```

A summary of the values is as follows:

- **CIRCUITPY_WIFI_SSID**: This is the name of the WiFi network that you usually see advertised in a list of available WiFi networks on your computer. It should match exactly.
- **CIRCUITPY_WIFI_PASSWORD**: This is the password for your WiFi network. It should also match exactly.
- **CIRCUITPY_WEB_API_PASSWORD**: This is the password that is used to protect access to the files on your device when connecting with Web Workflow. This is what you will type in when prompted for a password. You will want to change it to something that you'll remember.
- **CIRCUITPY_WEB_API_PORT**: This is the TCP port that the built-in Web Workflow web server on the device will use. At this point in time, the Code Editor hasn't been thoroughly tested with ports other than the default, so leaving it at **port 80 is recommended**.

If you would like a more technical detail of the underlying aspects of Web Workflow, be sure to check out the official CircuitPython documentation (https://adafru.it/10Bp).

After entering in your settings.toml a "hard" reset is required: power cycle or press the reset button, a soft reset is not enough!

Disabling USB Mass Storage

If you plan on running the camera off battery only, you shouldn't need to disable USB Mass Storage.

One of the more challenging aspects of using web workflow is that in order to avoid corrupting files, the USB mass storage device may need to be disabled **on native USB chipsets like ESP32-S3** if you plan on using the camera while connected to a computer over USB, otherwise the files will only be available in read-only mode over the WiFi link.

The easiest way to do this is to simply eject the **CIRCUITPY** drive in your OS. However, this will only be temporary until the next time you reset the device or plug it back in. If you'd like it to stay disabled, you can create a **boot.py** file in the root folder of the **CIRCUITPY** drive with the following contents:

```
import storage

storage.disable_usb_drive()
```

Once you have saved the file, go ahead and perform a hard reset on the board by pressing the reset button or by temporarily disconnecting it from power. After it starts back up, the **CIRCUITPY** drive should no longer appear. If you'd like to learn more about editing **boot.py** to customize the device, check out the Customizing USB Devices in CircuitPython (https://adafru.it/SC9) guide.

Re-enabling USB Mass Storage

If you find you want to access the mass storage device after you have disabled it, there are a couple of ways you can do it. The first way to temporarily enable it is to boot it into safe mode. To do this, just reset the device and while it is booting, pay attention to the status NeoPixel. When it starts flashing yellow, press the boot button.

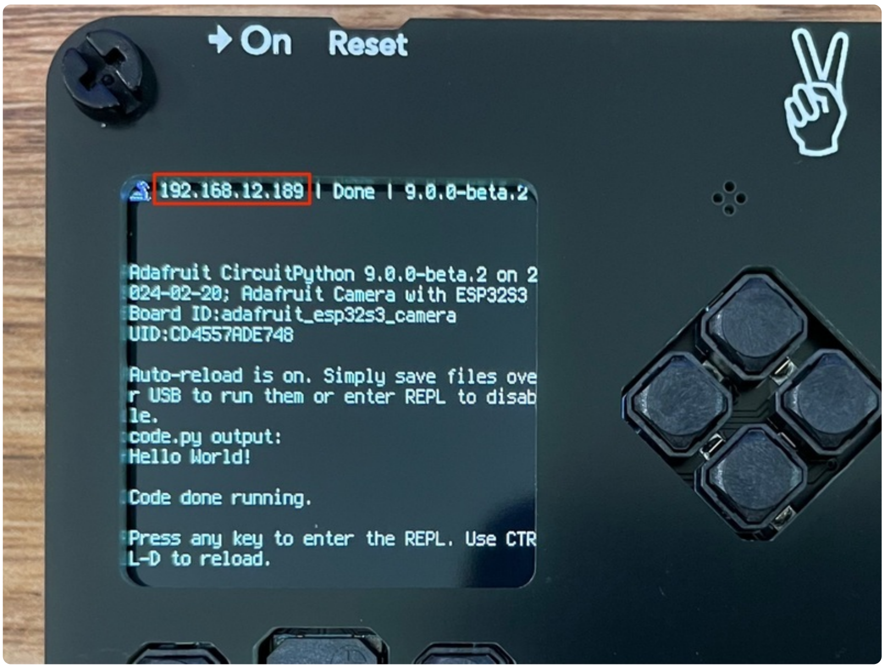
Alternatively, if you'd like to permanently re-enable it, you can run the following code from a serial terminal to remove the **boot.py** file:

```
import storage
import os
storage.remount("/", False)
os.remove("/boot.py")
```

After hard resetting your device, the **CIRCUITPY** drive should reappear, though re-enabling it will cause web workflow to be in read-only mode again.

Usage

At this point, your MEMENTO should be connected to your network. You can confirm this be looking at the top line of your display and seeing that it shows an IP address instead of "Wi-Fi: No IP". If it says "Wi-Fi: off", then you likely haven't added your Wi-Fi info to settings.toml yet.



```
192.168.12.189 | Done | 9.0.0-beta.2

Adafruit CircuitPython 9.0.0-beta.2 on 2
024-02-20; Adafruit Camera with ESP32S3
Board ID:adafruit_esp32s3_camera
UID:CD4557ADE748

Auto-reload is on. Simply save files ove
r USB to run them or enter REPL to disab
le.
code.py output:
Hello World!

Code done running.

Press any key to enter the REPL. Use CTR
L-D to reload.
```

If it launches code instead of displaying the REPL, you can try connecting to the serial console with a terminal application. For more details on how to do this, you can take a look at the Connecting to the Serial Console (https://adafru.it/19FA) page.

Install Fancy Camera Example

For the purposes of the guide, you will need to install the fancy camera example. This example has more features than the basic camera example and includes changing the mode to timelapse. To install, click **Download Project Bundle** below, unzip the file, and copy the full contents of the folder matching your version of CircuitPython to your **CIRCUITPY** drive.

If you already have an existing **boot.py** file, you will need to edit it and add the contents of both files. To learn more about this example, check out the MEMENTO Fancy Camera (https://adafru.it/19d4) guide page.

```
# SPDX-FileCopyrightText: Copyright (c) 2023 john park for Adafruit Industries
#
# SPDX-License-Identifier: MIT
""" simple point-and-shoot camera example. No bells! Zero whistles! """

import time
import adafruit_pycamera  # pylint: disable=import-error

pycam = adafruit_pycamera.PyCamera()
pycam.mode = 0  # only mode 0 (JPEG) will work in this example

# User settings - try changing these:
pycam.resolution = 8  # 0-12 preset resolutions:
#                       0: 240x240, 1: 320x240, 2: 640x480, 3: 800x600, 4: 1024x768,
#                       5: 1280x720, 6: 1280x1024, 7: 1600x1200, 8: 1920x1080, 9:
2048x1536,
#                       10: 2560x1440, 11: 2560x1600, 12: 2560x1920
```

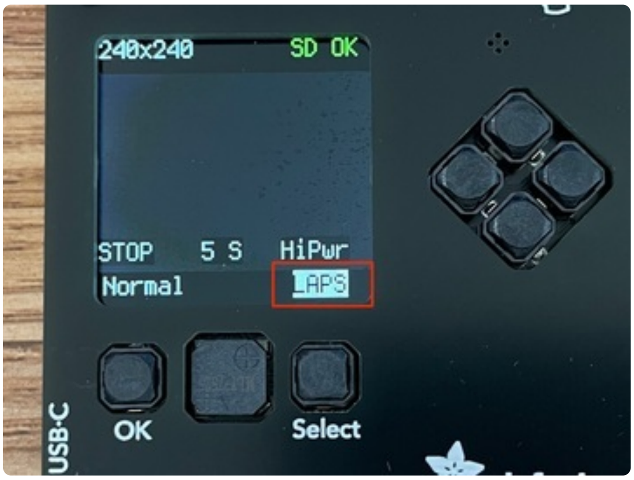
```
pycam.led_level = 1  # 0-4 preset brightness levels
pycam.led_color = 0  # 0-7  preset colors: 0: white, 1: green, 2: yellow, 3: red,
#                                          4: pink, 5: blue, 6: teal, 7: rainbow
pycam.effect = 0  # 0-7 preset FX: 0: normal, 1: invert, 2: b&w, 3: red,
#                                  4: green, 5: blue, 6: sepia, 7: solarize

print("Simple camera ready.")
pycam.tone(800, 0.1)
pycam.tone(1200, 0.05)

while True:
    pycam.blit(pycam.continuous_capture())
    pycam.keys_debounce()

    if pycam.shutter.short_count:
        print("Shutter released")
        pycam.tone(1200, 0.05)
        pycam.tone(1600, 0.05)
        try:
            pycam.display_message("snap", color=0x00DD00)
            pycam.capture_jpeg()
            pycam.live_preview_mode()
        except TypeError as exception:
            pycam.display_message("Failed", color=0xFF0000)
            time.sleep(0.5)
            pycam.live_preview_mode()
        except RuntimeError as exception:
            pycam.display_message("Error\nNo SD Card", color=0xFF0000)
            time.sleep(0.5)

    if pycam.card_detect.fell:
        print("SD card removed")
        pycam.unmount_sd_card()
        pycam.display.refresh()

    if pycam.card_detect.rose:
        print("SD card inserted")
        pycam.display_message("Mounting\nSD Card", color=0xFFFFFF)
        for _ in range(3):
            try:
                print("Mounting card")
                pycam.mount_sd_card()
                print("Success!")
                break
            except OSError as exception:
                print("Retrying!", exception)
                time.sleep(0.5)
        else:
            pycam.display_message("SD Card\nFailed!", color=0xFF0000)
            time.sleep(0.5)
        pycam.display.refresh()
```

Install an SD Card

Make sure you have a micro SD card installed in the camera. This is where the photos will be written to. One of the newer features of Web Workflow is the ability to access the files from your SD card remotely and download photos. The SD Card doesn't need to be very big. You can use a small one like the 256MB Micro SD Card (http://adafru.it/5251) in the Adafruit shop.

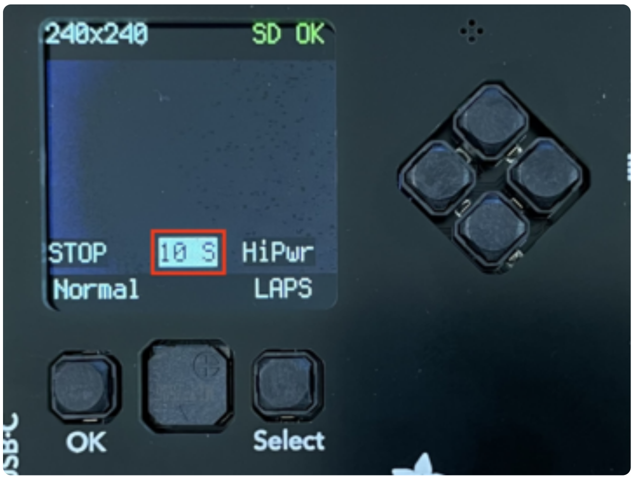
Put in Timelapse Mode

For this example, this guide will show you how you can use timelapse photography and access the photos while it is recording. To find out more information about the timelapse mode, visit the Timelapse Project (https://adafru.it/19FB) page of the main MEMENTO guide.

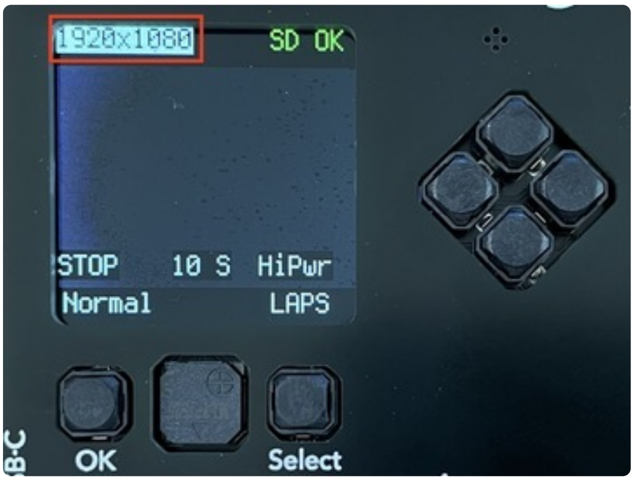


Press the **left/right** buttons, move the selector over to the mode option in the lower righthand side of the screen. It should say **JPEG** if you haven't already adjusted it. Once it is highlighted, use the **up/down** buttons to change it to **LAPS**.

To conserve battery, you can change it from Hi Power to Medium or Low power mode at this point by pressing the **Select** button.



Now using the left/right buttons, select the time interval between photos and adjust to whatever you like. For the purposes of this example, it will be set to 10 seconds.



If desired, change the resolution to one for which you would like to capture images. The default is 240x240 pixels, but you may desire a little bit higher resolution than that.

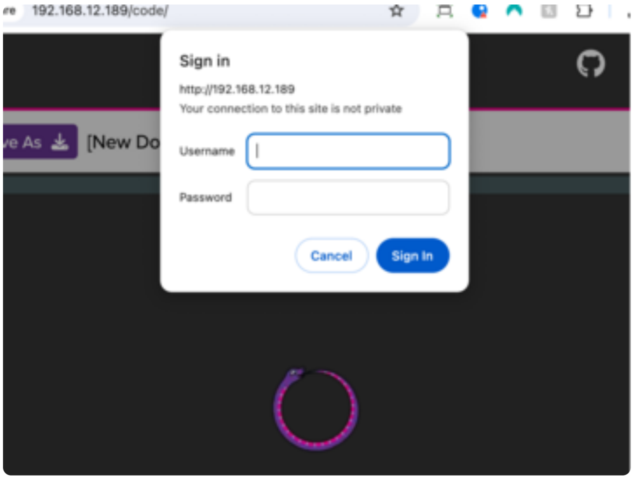
Set up your MEMENTO camera to point at the subject that you would like to take a timelapse of and press **OK** to start. You may wish to set the camera on a steady surface so that it doesn't inadvertently tip over while recording. The timer countdown will start and after it finishes, it will save a photo on the SD Card.

Accessing the Photos

Uploading or deleting any photos during a timelapse can cause the MEMENTO to reset and the current timelapse will need to be started again!

Now that your camera is all set up and taking a timelapse, you may want to check the progress or download the photos.

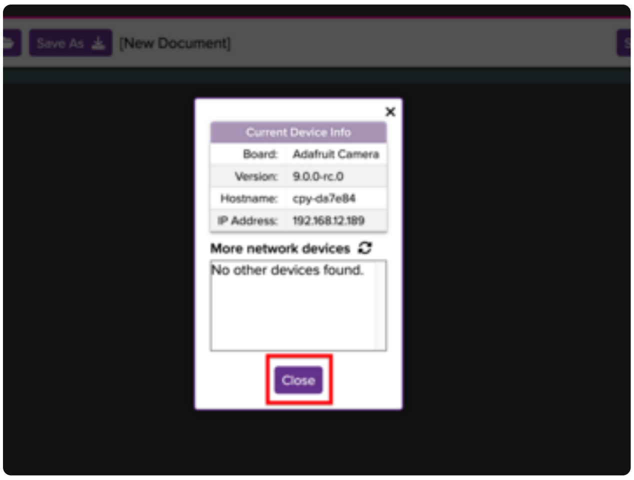
You can do this through the enhanced code editor, which can be opened by either going to **http://circuitpython.local/code/** or typing **http://[device IP address]/code/** (replacing "[device IP address]" with the actual IP address). Be sure to include the trailing slash or it won't load the page.



http://192.168.12.189

Your connection to this site is not private

Once it loads, it should prompt for a password. Leave the username field blank and use the password that you set in your settings.toml file.



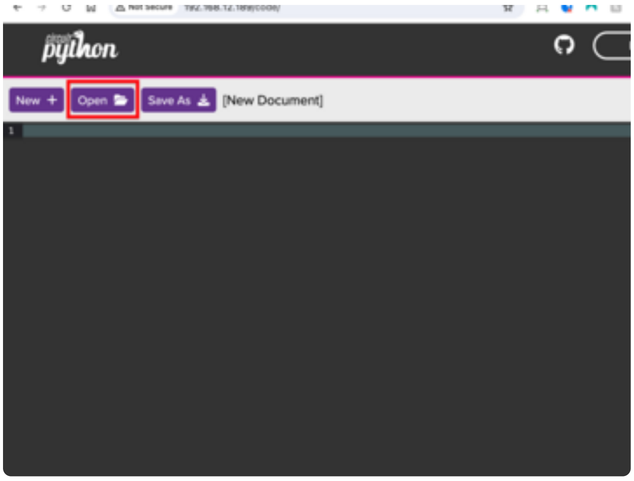
Current Device Info

Board:	Adafruit Camera
Version:	9.0.0-rc.0
Hostname:	cpy-da7e84
IP Address:	192.168.12.189

More network devices

No other devices found.

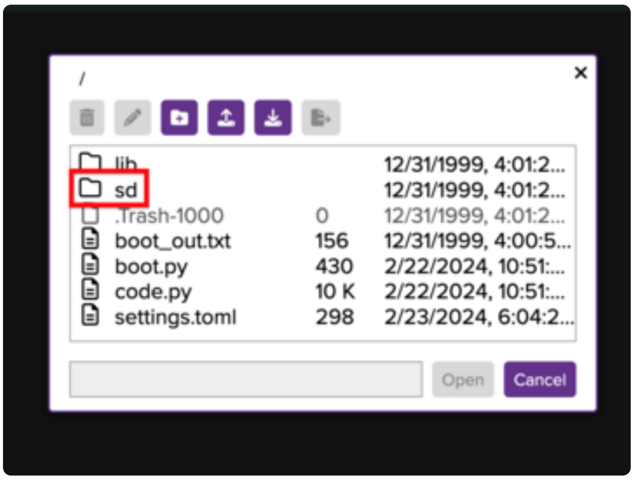
Once you login, an information box about the board you connected to will appear. Click the **Close** button.



Go ahead and click the **Open** button at the top. This will allow you to do some additional file operations such as downloading files.

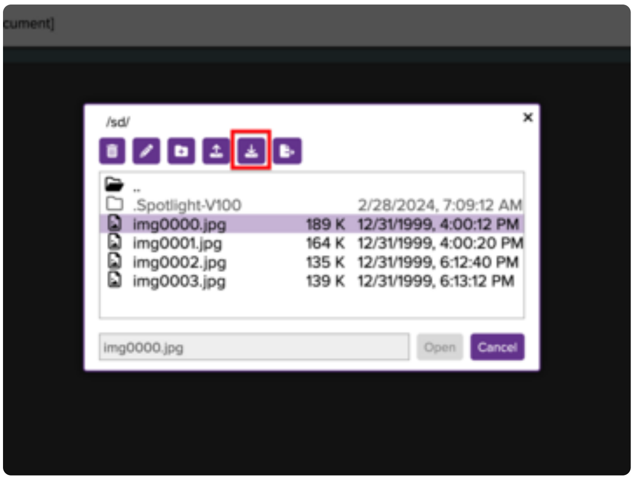
When Downloading, you may get a browser notice that the file you are downloading is over an insecure connection. This is because the device is communicating over HTTP rather than HTTPS because of the current limitations of Web Workflow and CircuitPython.

Downloading a Single Photo



lib		12/31/1999, 4:01:2...
sd		12/31/1999, 4:01:2...
.Trash-1000	0	12/31/1999, 4:01:2...
boot_out.txt	156	12/31/1999, 4:00:5...
boot.py	430	2/22/2024, 10:51:...
code.py	10 K	2/22/2024, 10:51:...
settings.toml	298	2/23/2024, 6:04:2...

With the File Dialog box open, doubleclick on the **sd folder** or select the file and click **Open**.



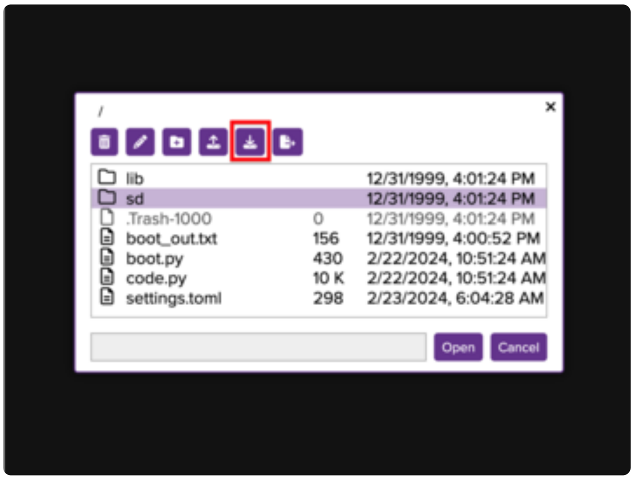
/sd/

..		
.Spotlight-V100		2/28/2024, 7:09:12 AM
img0000.jpg	189 K	12/31/1999, 4:00:12 PM
img0001.jpg	164 K	12/31/1999, 4:00:20 PM
img0002.jpg	135 K	12/31/1999, 6:12:40 PM
img0003.jpg	139 K	12/31/1999, 6:13:12 PM

img0000.jpg

Select the file you wish to download and then click on the **Download button** (the button with the arrow pointing down).

Downloading All Photos

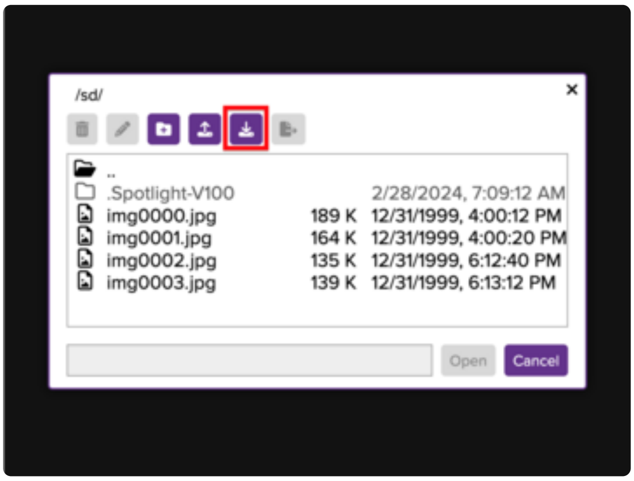


/

lib		12/31/1999, 4:01:24 PM
sd		12/31/1999, 4:01:24 PM
.Trash-1000	0	12/31/1999, 4:01:24 PM
boot_out.txt	156	12/31/1999, 4:00:52 PM
boot.py	430	2/22/2024, 10:51:24 AM
code.py	10 K	2/22/2024, 10:51:24 AM
settings.toml	298	2/23/2024, 6:04:28 AM

There are a couple of ways that you can do this.

The first is by selecting the SD folder and then clicking on the **Download button** (the button with the arrow pointing down).



/sd/

..		
.Spotlight-V100		2/28/2024, 7:09:12 AM
img0000.jpg	189 K	12/31/1999, 4:00:12 PM
img0001.jpg	164 K	12/31/1999, 4:00:20 PM
img0002.jpg	135 K	12/31/1999, 6:12:40 PM
img0003.jpg	139 K	12/31/1999, 6:13:12 PM

The second way is by entering into the SD Folder and then **without selecting anything**, just click the **Download button**.

This is only one example of what you can do with Web Workflow. You'll probably be able to think of many more things you can use this for. Future improvements in CircuitPython and the Code Editor should make using Web Workflow even more useful.