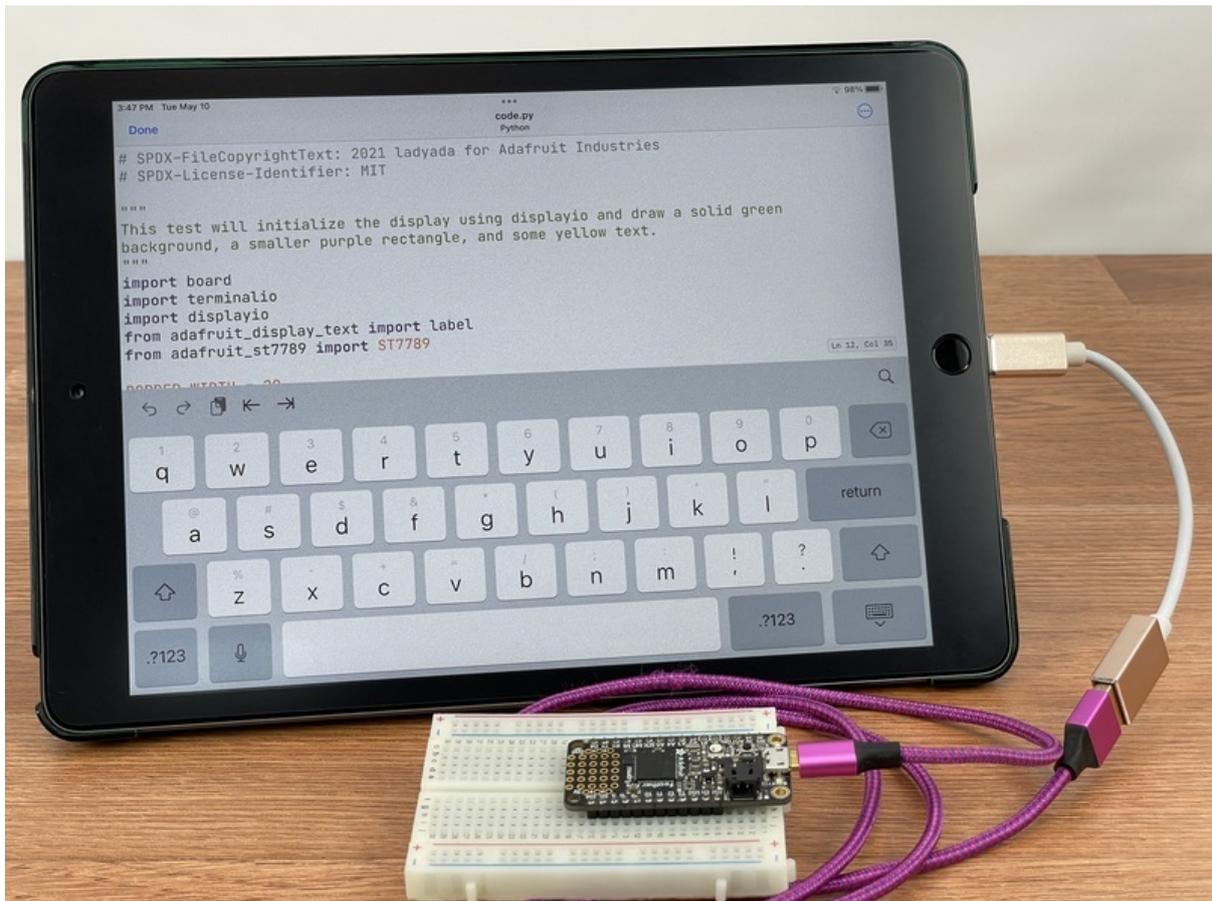




# Edit CircuitPython Code on iOS with Runestone

Created by Melissa LeBlanc-Williams



<https://learn.adafruit.com/editing-circuitpython-on-ios-with-runestone>

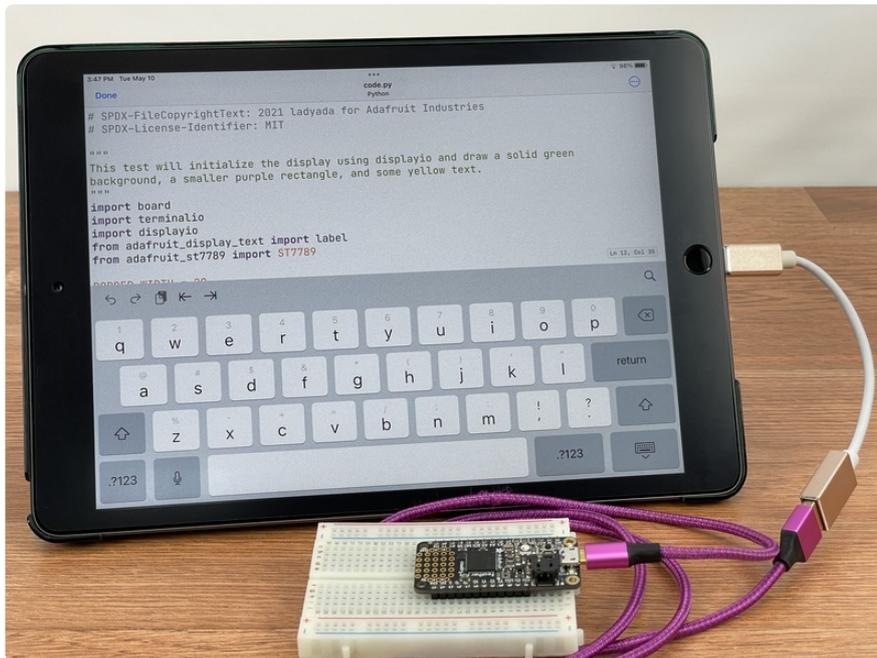
Last updated on 2024-06-03 03:37:09 PM EDT

# Table of Contents

<a href="#">Overview</a>	<a href="#">3</a>
<a href="#">Usage</a>	<a href="#">3</a>
<ul style="list-style-type: none"><li><a href="#">• Example Usage</a></li></ul>	

---

# Overview



Up to this point, editing code on CircuitPython devices has sometimes been tough to get set up and working. Some editors have trouble with not writing the code to the board immediately and cache it instead and some will just show everything as one color, making coding a little bit more difficult.

[Runestone](https://adafru.it/-a6) (<https://adafru.it/-a6>) is a streamlined and lightweight Text and Code Editor that runs on iOS devices such as the iPhone and iPad with color coding that supports [CircuitPython](https://adafru.it/BIM) (<https://adafru.it/BIM>). It is a free app, that has an option to unlock premium features such as line numbers, page guides, app icon customization, and displaying invisible characters.

Runestone does an excellent job of giving you exactly what you need in order to code on the go without too many bells and whistles which can make the learning curve much steeper. Although it doesn't have some features such as tabs, this works for CircuitPython because saving the file will cause the code to reload anyways. This guide should help get you up and running so you can start editing on iOS devices.

---

## Usage

To use Runestone on Lightning port-equipped iOS devices (currently all iPhones and many iPad models), you will need an iOS Lightning to USB adapter such as Apple's official "camera connection" adapter or a similar cable available in the Adafruit shop...



### [iOS Lightning to USB OTG Cable](#)

Your iOS phone or tablet may not have a USB port on the bottom but that doesn't mean you can't use it to connect USB devices. Secretly known as a 'Camera Connector' or...

<https://www.adafruit.com/product/3940>

Recent iPads use USB C instead of the Lightning connector, and a common USB C-to-C or C-to-micro-B cable can be used.

Unlike the USB on a computer, which typically has a maximum current of **500ma** on USB 2.0 ports, Apple iOS devices will only deliver a maximum of **100ma** before the device disables power output to the port.

Adafruit boards that are running the latest CircuitPython should advertise that they are not over the 100ma threshold. However, when they go into bootloader mode, they may advertise that they use 500ma, which will cause the iOS device to shutdown the port.

If you have this issue or you have peripherals attached to your board that draw a large amount of current, such as a bunch of Neopixels, you may also need a powered USB hub to get around this limitation. This allows you to power higher current devices without exceeding the 100ma limit.



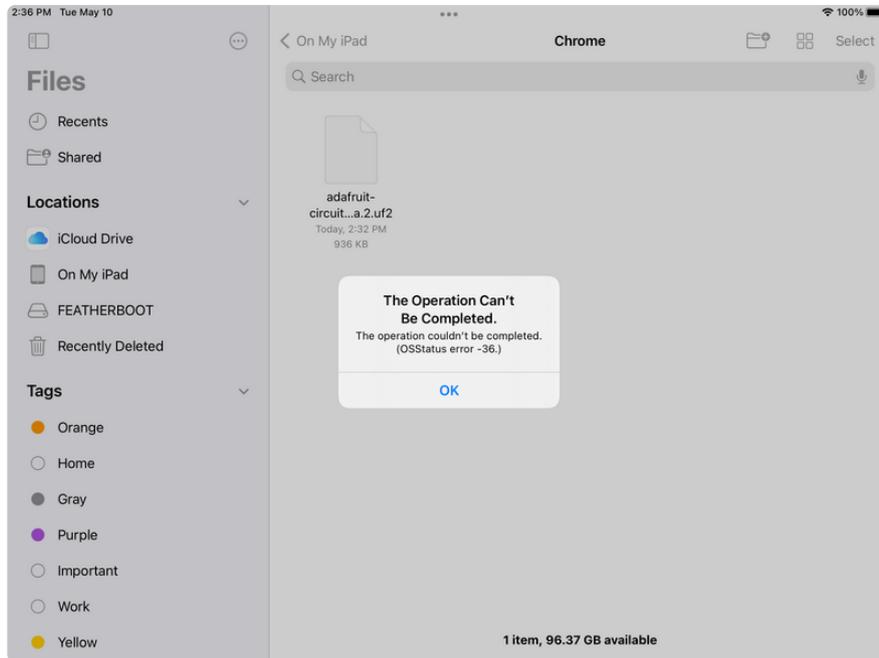
### [USB 2.0 Powered Hub - 7 Ports with 5V 2A Power Supply](#)

Add lots more USB capability to your Raspberry Pi or computer using this powered USB 2.0 hub. It adds a full seven powered ports, all at USB 2.0 speeds so you can use...

<https://www.adafruit.com/product/961>

Make sure you are running the latest version of iOS. Earlier versions had trouble with external devices.

You will need to [install CircuitPython \(https://adafru.it/Amd\)](https://adafru.it/Amd) onto your board ahead of time using a computer because iOS currently throws an error if you attempt to install CircuitPython.



## Example Usage

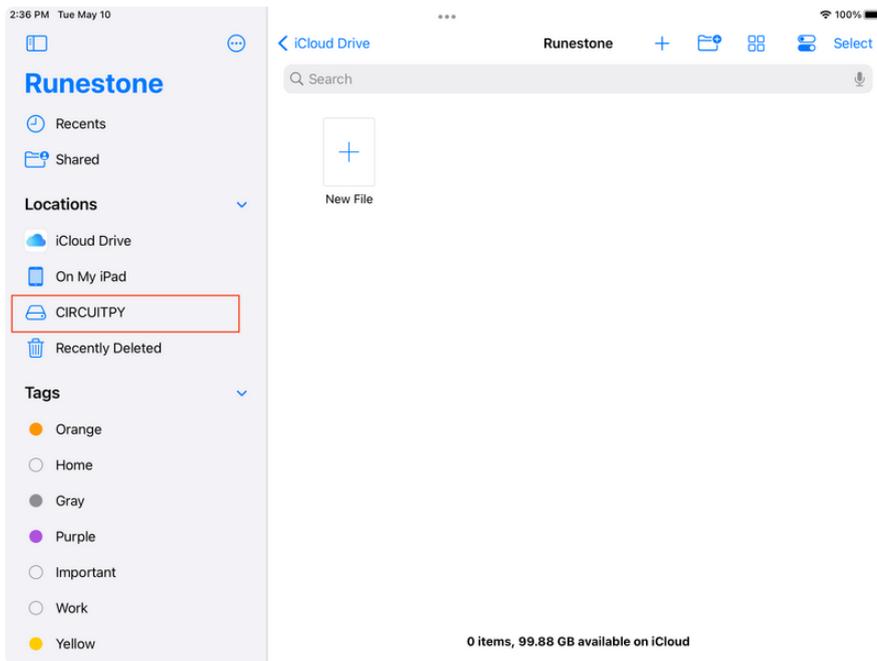
If your `code.py` file contained the following code to blink the on-board LED and you wanted to slow the LED down, here's how you would go about doing that.

```
import time
import board
import digitalio

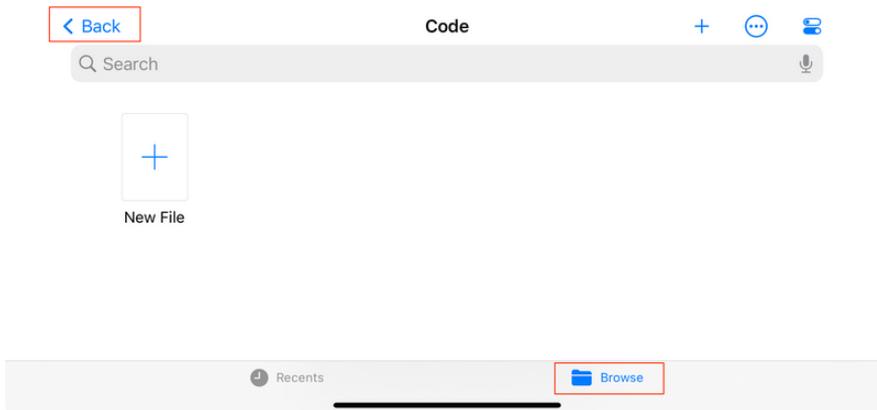
led = digitalio.DigitalInOut(board.LED)
led.switch_to_output(value=False)

while True:
    led.value = not led.value
    time.sleep(0.5)
```

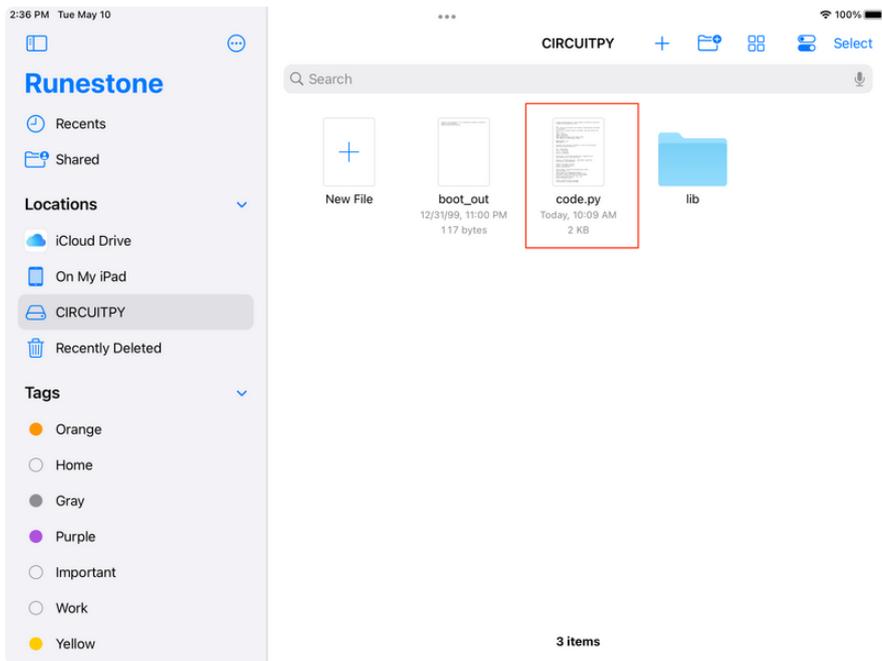
Once you have the board running CircuitPython connected, a drive called **CIRCUITPY** should appear in the list of locations.



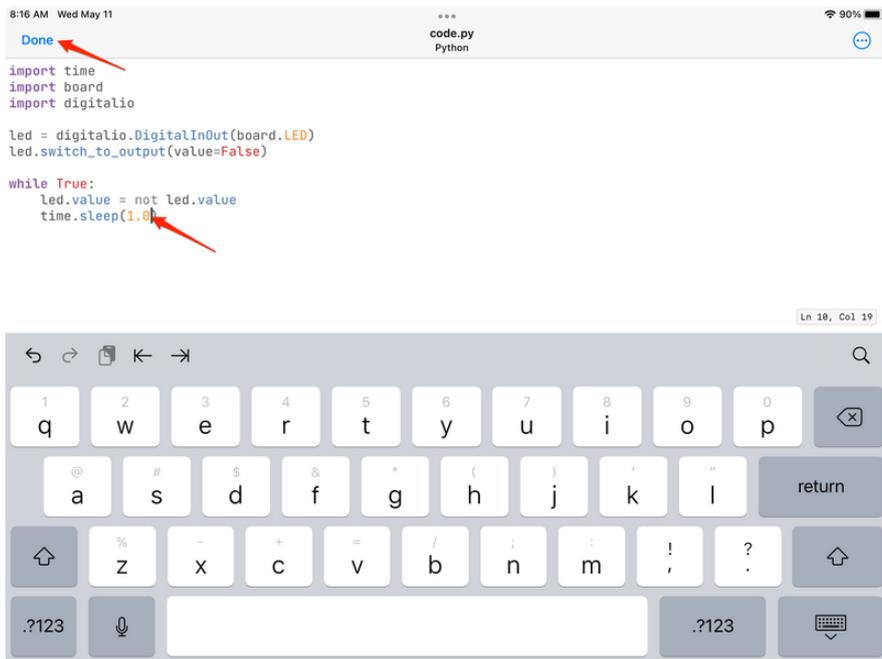
If you are editing on an iPhone device, the interface is a bit more compact. When you first start up the app, you may need to make sure you are on **Browse** and hit the **Back** link a few times to get to locations where you can select the **CIRCUIPTY** drive.



Select the drive and a list of files on the drive should appear. Then select file that you would like to edit. In this example it would be **code.py**.



Tap next to the **0.5** inside of the `time.sleep()` function and change the value to **1.0**. To save and reload the running program, press the **Done** link in the upper lefthand corner.



Now the LED should be waiting for one after turning on or off for a total of two seconds between blinks.

