Exercise Buddy: Motion aware BLE media controller

Created by Charlyn G

https://learn.adafruit.com/exercise-buddy

Last updated on 2024-06-03 03:33:59 PM EDT

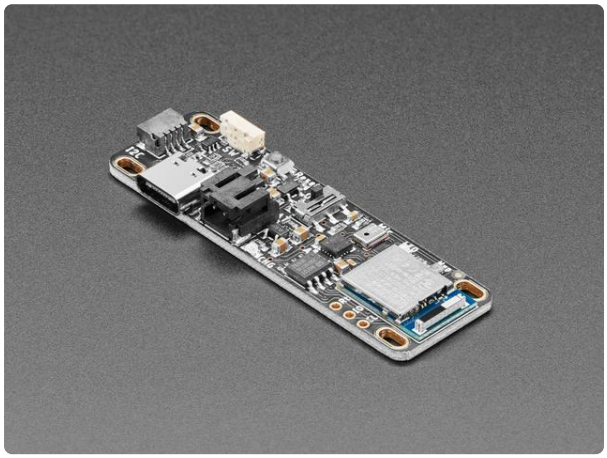
Table of Contents

Overview

Watching a riveting show while on a stationary bike is a good way to motivate yourself to exercise more regularly. There was even a scientific study done about it, which coined the term "temptation bundling" (https://adafru.it/Ywa). But sometimes, the show might get way too interesting and you might not even notice that you've stopped cycling!

This project creates a device that can detect simple movement, and when it detects that you've stopped moving, it will pause any videos or music playing on your Apple mobile device (iPad, iPhones, etc.). It's an exercise buddy that will remind you to keep moving so you can keep watching!

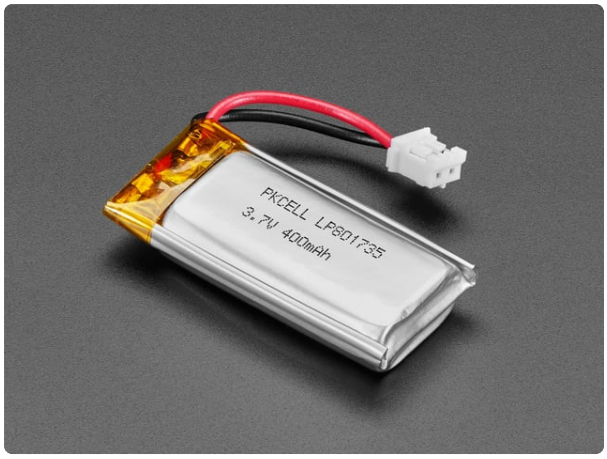
Parts



Adafruit LED Glasses Driver - nRF52840 Sensor Board

This board is designed to be a thin, bluetooth-enabled driver board for our Adafruit LED Glasses RGB LED matrix. That said, it's...

https://www.adafruit.com/product/5217



Lithium Ion Polymer Battery Ideal For Feathers - 3.7V 400mAh

Lithium-ion polymer (also known as 'lipo' or 'lipoly') batteries are thin, light, and powerful. The output ranges from 4.2V when completely charged to 3.7V. This...

https://www.adafruit.com/product/3898

1 x Velcro strap — https://amzn.to/3rIstP2

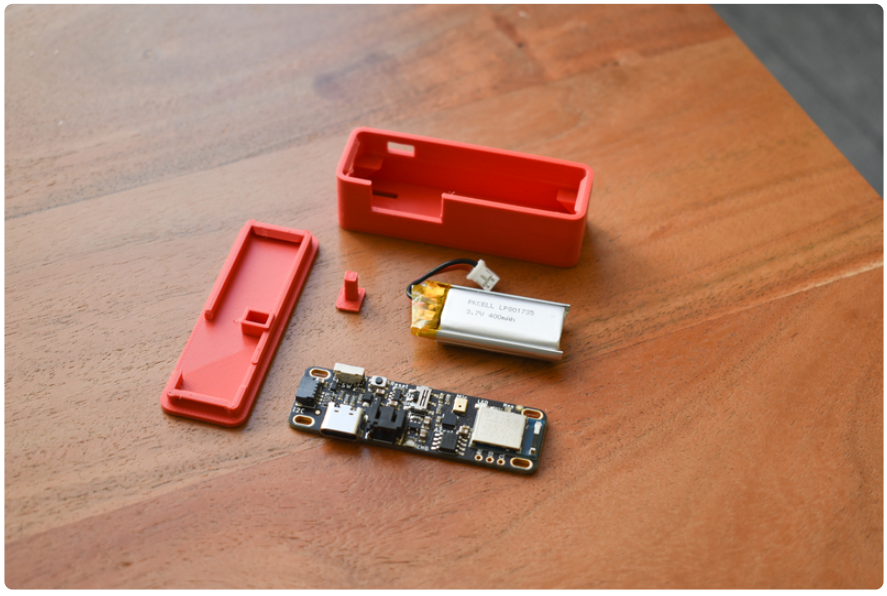
Any velcro strip with a hook side and a loop side will do.

---

1 x 3D printing filament — https://amzn.to/3oMf8Dg

Pick your favorite PLA filament! This is the one I used for this project

---



3D printing and assembly

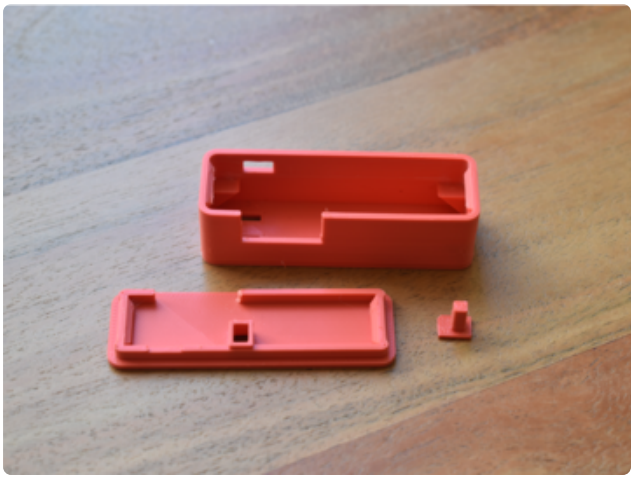
Download the files for this case using the buttons below. If downloading from the zip file, note that the models inside the `STL_files` folder and `3MF_files` folder are the same. There is also a `.f3z` file included in case you wanted to modify or adjust parameters on the model itself.

Download from PrusaPrinters

https://adafru.it/Ywb

3D_files.zip

https://adafru.it/Ywc



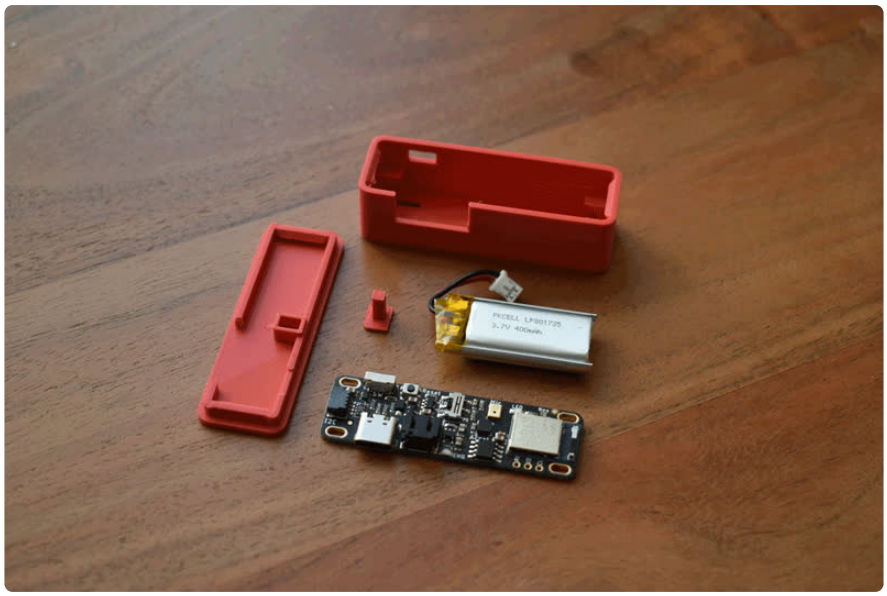
The case is made up of 3 pieces:

case
cover - snap fits onto the case.
switch extender - fits on top of the onboard switch so it can be reachable from outside the case.
Print out all three, using 0.2mm layers and no supports. Choose your favorite color PLA filament.

The case has a "shelf" that the board sits on so that it doesn't crush the battery. It also has a pair of slots that you can thread some velcro through to secure it to an armband or a bike pedal arm.

Cut about 17cm length of velcro, and reduce the strip width to around 11mm. Thread it through the slots before placing the board inside as shown below.

Depending on where you will mount it, you can also opt to add a rubber or silicone piece under the case, to add a bit more stability. The clear silicone piece shown here was from a local dollar store, but a couple of the more common small circular rubber feet might just do the trick as well.



Attach the battery to the BLE sensor board, and align the USB-C port to the biggest hole on the case. Push the board down so that it is sitting on the shelf. The battery wire will stick out a little. Then, place the switch extender on top of the board switch, you should feel the switch slot into the hole.

Finally, align the switch extender to the switch port on the cover, and snap fit into place. Voila! You should still have access to the right-angle button and the STEMMA QT connector.

CircuitPython

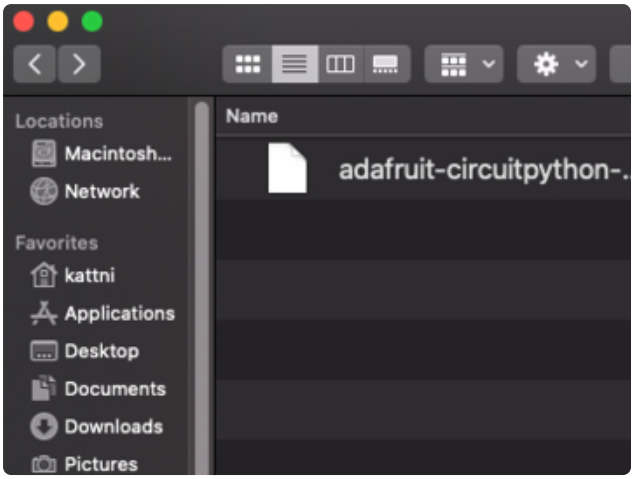
CircuitPython (https://adafru.it/tB7) is a derivative of MicroPython (https://adafru.it/BeZ) designed to simplify experimentation and education on low-cost microcontrollers. It makes it easier than ever to get prototyping by requiring no upfront desktop software downloads. Simply copy and edit files on the **CIRCUITPY** drive to iterate.

CircuitPython Quickstart

Follow this step-by-step to quickly get CircuitPython running on your board.

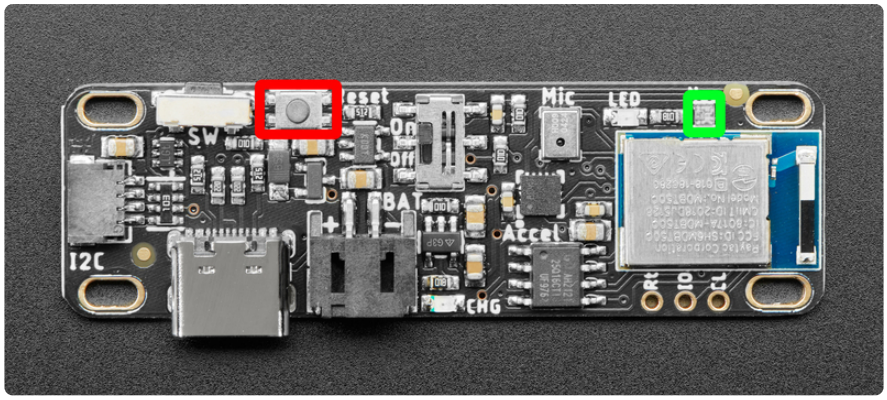
Download the latest version of CircuitPython for this board via circuitpython.org

https://adafru.it/VcN



Click the link above to download the latest CircuitPython UF2 file.

Save it wherever is convenient for you.

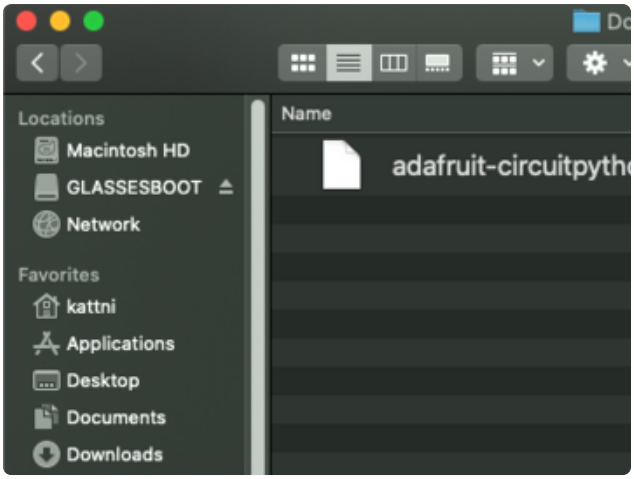


Plug your board into your computer, using a known-good data-sync cable, directly, or via an adapter if needed.

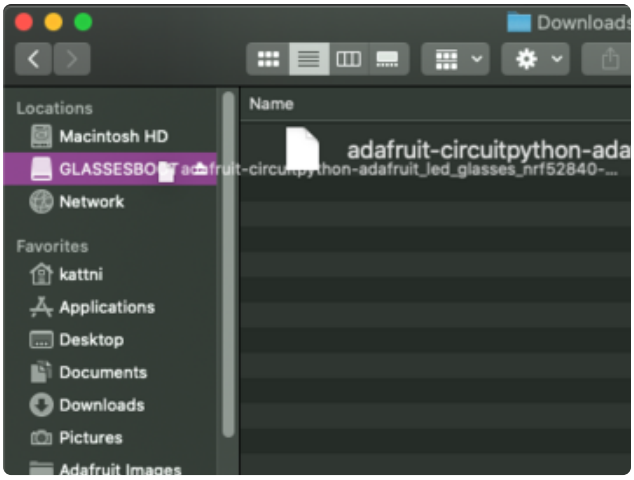
Double-click the **reset** button (highlighted in red above), and you will see the **RGB status LED(s)** turn green (highlighted in green above). If you see red, try another port, or if you're using an adapter or hub, try without the hub, or different adapter or hub.

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

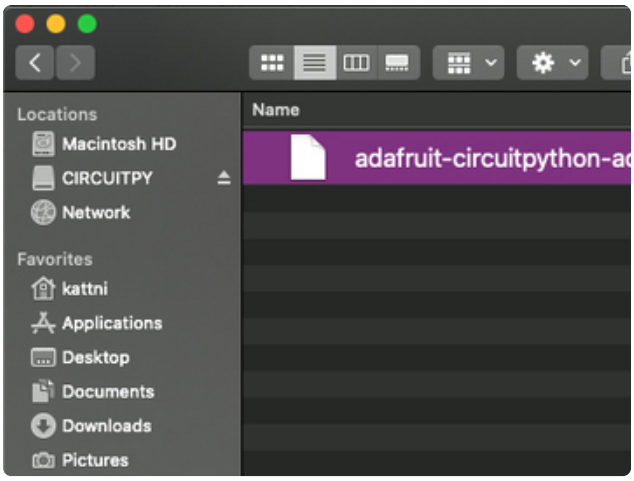
A lot of people end up using charge-only USB cables and it is very frustrating! **Make sure you have a USB cable you know is good for data sync.**



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



Drag the **adafruit_circuitpython_etc.uf2** file to **GLASSESBOOT**.



The **BOOT** drive will disappear and a new disk drive called **CIRCUITPY** will appear.

That's it!

CircuitPython libraries and code

Plug your Adafruit LED Glasses Driver - nRF52840 Sensor Board (http://adafru.it/5217) into the computer's USB port with a known good power+data USB cable. You should see a new flash drive appear in the computer's File Explorer or Finder (depending on your operating system) called **CIRCUITPY**.

Click the "Download Project Bundle" button under the Code section below.

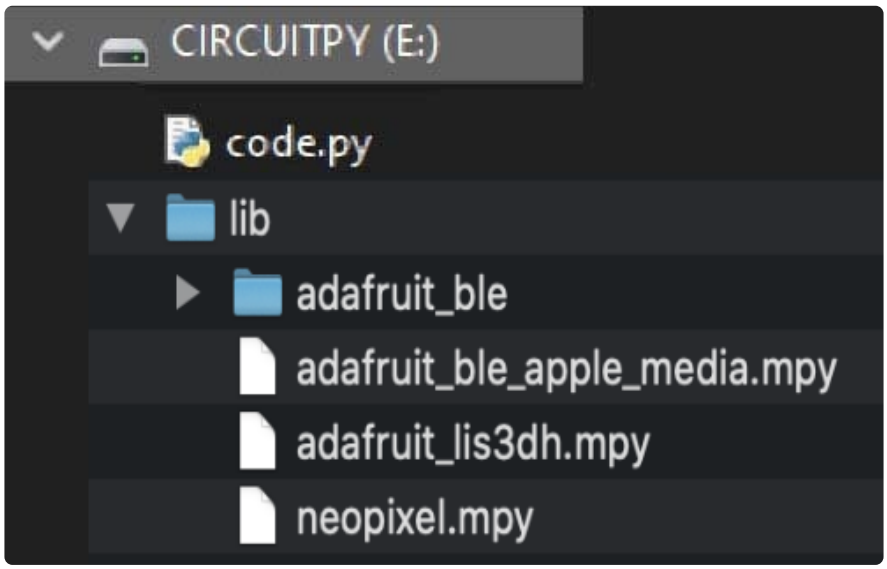
Unzip the folder and copy the following items to the **CIRCUITPY** drive:

- **lib** folder
- **code.py**

The project uses the **lib** folder libraries to access the accelerometer and to be able to send play and pause commands to an Apple mobile device.

- adafruit_ble
- adafruit_ble_apple_media
- adafruit_lis3dh
- neopixel

Your **CIRCUITPY** drive should contain the files listed below:




```
# SPDX-FileCopyrightText: 2022 Charlyn G for Adafruit Industries
#
# SPDX-License-Identifier: MIT
#
# Code for the Adafruit Learning System tutorial
#   Exercise Buddy: Motion aware BLE media controller
#   https://learn.adafruit.com/exercise-buddy/overview
#
import time
import board
import supervisor

import neopixel
import adafruit_ble
import adafruit_lis3dh
from adafruit_ble.advertising.standard import SolicitServicesAdvertisement
from adafruit_ble_apple_media import AppleMediaService, UnsupportedCommand

# Initialize the accelerometer
i2c = board.I2C()  # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C()  # For using the built-in STEMMA QT connector on a
```

```
microcontroller
lis3dh = adafruit_lis3dh.LIS3DH_I2C(i2c)

#  Initialize BLE radio
radio = adafruit_ble.BLERadio()
a = SolicitServicesAdvertisement()
a.solicited_services.append(AppleMediaService)
radio.start_advertising(a)

# Neopixel indicator
pixel_pin = board.NEOPIXEL
pixel = neopixel.NeoPixel(pixel_pin, 1, brightness=0.5)
YELLOW = (200, 150, 0)
CYAN = (0, 100, 100)
PINK = (231, 84, 128)
pixel.fill(PINK)

while not radio.connected:
    pass

print("connected")
pixel.fill(YELLOW)

# Initialize variables
last_x = 0
last_y = 0
last_z = 0
paused = True

WAIT = 0.2
WIGGLE_ROOM = 5  # Increase this for more jitter compensation.


def is_same_pos(last_position, current_position):
    # Returns true if current_position is similar enough
    # to last_position, within the specified wiggle room.
    diff = abs(current_position - last_position)
    print((diff,))
    return diff <= WIGGLE_ROOM


def not_enough_movement(x, y, z):
    same_x = is_same_pos(last_x, x)
    same_y = is_same_pos(last_y, y)
    same_z = is_same_pos(last_z, z)
    return same_x and same_y and same_z


while radio.connected:

    for connection in radio.connections:
        if not connection.paired:
            connection.pair()
            print("paired")
            pixel.fill(PINK)
            time.sleep(1)

        if connection.paired:
            pixel.fill(CYAN)
            ams = connection[AppleMediaService]
            print("app:", ams.player_name)

            try:
                xf, yf, zf = lis3dh.acceleration

                if not_enough_movement(xf, yf, zf):
                    # Keep pausing.
                    print("pause!")
                    paused = True
```

```
                    ams.pause()

                else:
                    last_x = xf
                    last_y = yf
                    last_z = zf

                    if paused:
                        print("play!")
                        paused = False
                        ams.play()

            except OSError:
                supervisor.reload()
            except UnsupportedCommand:
                # This means that we tried to pause but there's
                # probably nothing playing yet, so just wait a bit
                # and try again.
                pixel.fill(PINK)
                time.sleep(10)
                supervisor.reload()

            time.sleep(WAIT)

print("disconnected")
pixel.fill(PINK)
```

Code

This board has a built-in accelerometer to detect and calculate movement. To get a rundown of the basics of how accelerometers work, check out the excellent Adafruit guide Make It Shake, Rattle, and Roll: Accelerometer Use (https://adafru.it/Ywd).

The idea behind the code is simple: remember the last x, y and z position, and check to see if there has been enough movement in each axis. If there has not been enough movement since the last saved position, send a pause command to the Apple mobile device (and keep sending the pause command so that the video stays paused even if you press "play" on the device itself). Devious, but necessary.

Also the onboard NeoPixel is used as a status indicator, so that you can get a clue about what the device is up to, even when it's not connected to a computer.

Pair your Apple device to your project and take it for a spin! If you are not moving, the audio should be paused.

Literally, attach it to your stationary bike pedal and get ready to be motivated to keep pace. You might also consider attaching it to your body if you're doing some other form of physical exercise, but you may have to fiddle around with the `WIGGLE_ROOM` value depending on what kind of movement is involved.

Good luck and have a good workout!


ASK AN ENGINEER
LIMOR "LADYADA" FRIED
FOUNDER AND ENGINEER, ADAFRUIT
CBC