



Bluetooth Turtle Bot with CircuitPython and Crickit

Created by John Park



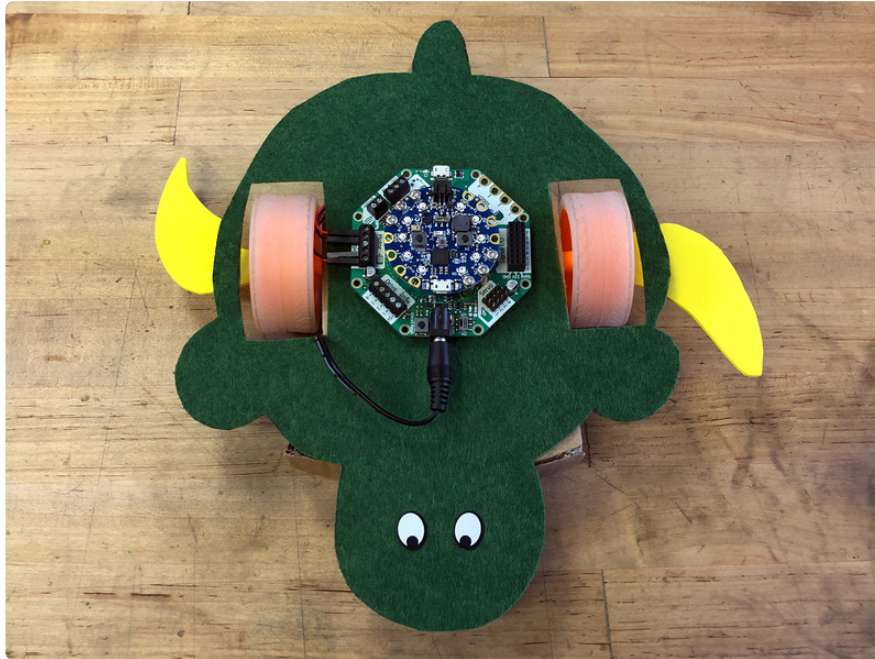
<https://learn.adafruit.com/bluetooth-turtle-bot-with-circuitpython-and-crickit>

Last updated on 2024-06-03 02:57:47 PM EDT

Table of Contents

Overview	3
<hr/>	
• Parts	
• Materials and Tools	
Build the Turtle Bot	6
<hr/>	
• Circuit Build	
• CPB to Crickit	
• Motors to Crickit	
• Moar Powaaaaarr!	
• Chassis	
• Form the Chassis	
• Attach Motors	
• Add Wheels	
• Optional Bumper	
• Turtle Decoration	
• Attach Crickit	
• Flippers	
CircuitPython on Circuit Playground Bluefruit	21
<hr/>	
• Install or Update CircuitPython	
Code with CircuitPython	23
<hr/>	
• Libraries	
• The Mu Editor	
• Turtle Rover Code	
• How the Code Works	
• Motor Movement	
• Control the Rover with the Bluefruit LE Connect App	
• Connect to CPB	
• Drive and Light	

Overview



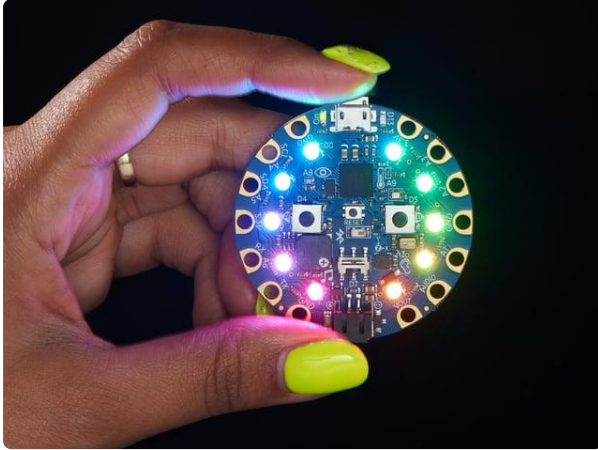
You can make your own adorable turtle bot and control it from your mobile device using Bluetooth LE!

This cute little bot uses the Circuit Playground Bluefruit as the brains and BLE communicator, and the Crickit platform as the brawn to drive two DC gearhead motors.

You can drive forward and backward, turn left and right, and even change the colors of the built-in NeoPixel ring on the Circuit Playground Bluefruit, all from the Adafruit Bluefruit app for iOS and Android.

Plus you can get crafty with the felt or foam and decorate this cardboard chassis bot like a cute turtle!

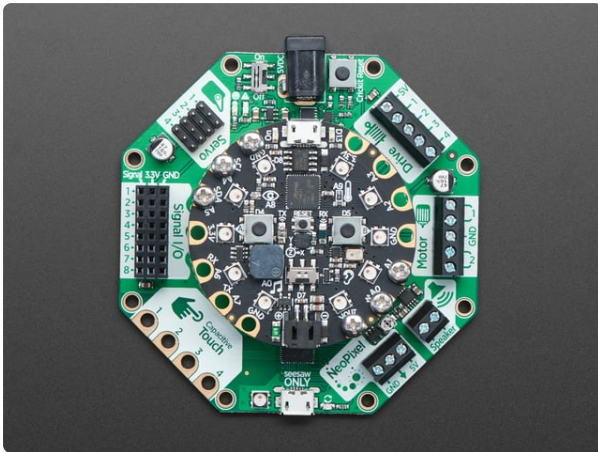
Parts



[Circuit Playground Bluefruit - Bluetooth Low Energy](https://www.adafruit.com/product/4333)

Circuit Playground Bluefruit is our third board in the Circuit Playground series, another step towards a perfect introduction to electronics and programming. We've...

<https://www.adafruit.com/product/4333>



[Adafruit CRICKIT for Circuit Playground Express](https://www.adafruit.com/product/3093)

Sometimes we wonder if robotics engineers ever watch movies. If they did, they'd know that making robots into servants always ends up in a robot rebellion. Why even go down that...

<https://www.adafruit.com/product/3093>



[USB Battery Pack - 2200 mAh Capacity - 5V 1A Output](https://www.adafruit.com/product/1959)

A smaller-sized rechargeable battery pack for your Raspberry Pi or Raspberry...

<https://www.adafruit.com/product/1959>



USB to 2.1mm Male Barrel Jack Cable

There's two standard ways to power electronics - USB or 5.5mm/2.1mm DC barrel jack. This or that! With this USB to 2.1mm Male Barrel Jack Cable, you can now power...

<https://www.adafruit.com/product/2697>



EVA Foam Pack in Rainbow Colors - 2mm thick - 10 sheets

EVA foam is one of the best and easiest go-to materials for constructing costumes and props. It's light, inexpensive, versatile, and can easily be cut and heat-shaped for a snug...

<https://www.adafruit.com/product/4349>

2 x DC Gearbox Motor

"TT Motor" - 200RPM - 3 to 6VDC

<https://www.adafruit.com/product/3777>

2 x Orange and Clear TT Motor Wheel

for TT DC Gearbox Motor

<https://www.adafruit.com/product/3766>

1 x Adafruit Pocket Screwdriver

Black

<https://www.adafruit.com/product/3284>

1 x Black and White Eye Stickers

eyes!

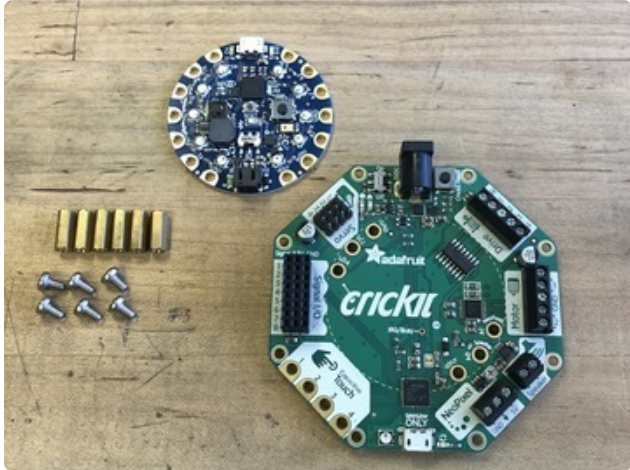
<https://www.adafruit.com/product/3840>

Materials and Tools

- Medium corrugated cardboard box
- Ruler
- Hobby knife or scissors
- Double stick foam tape
- Hot melt glue and gun

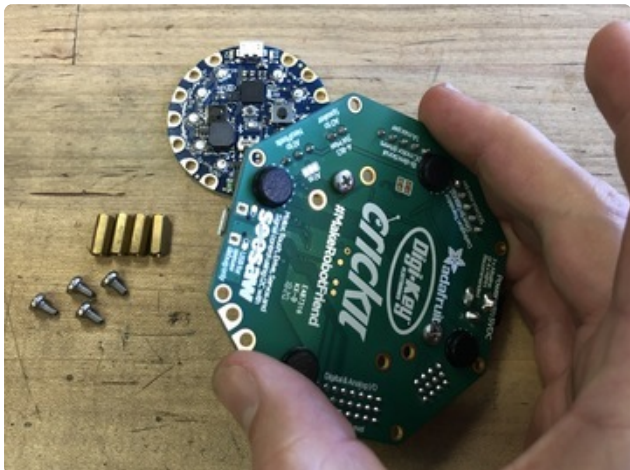
Build the Turtle Bot

Circuit Build

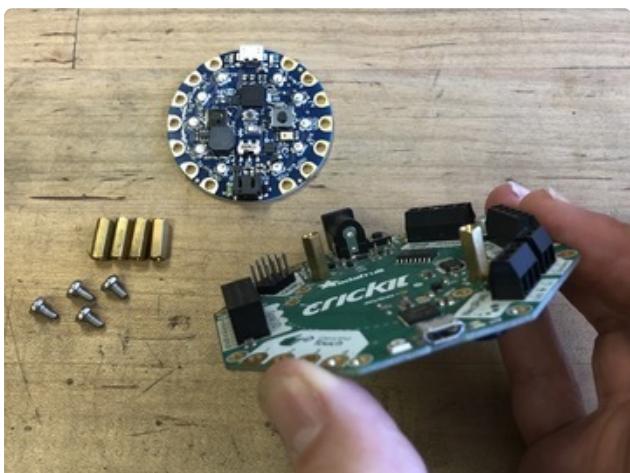


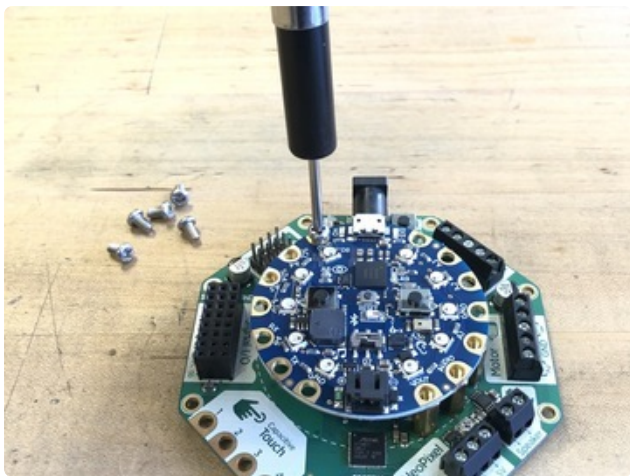
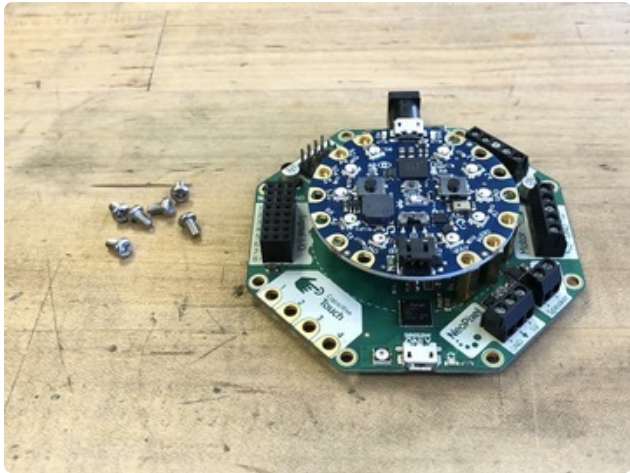
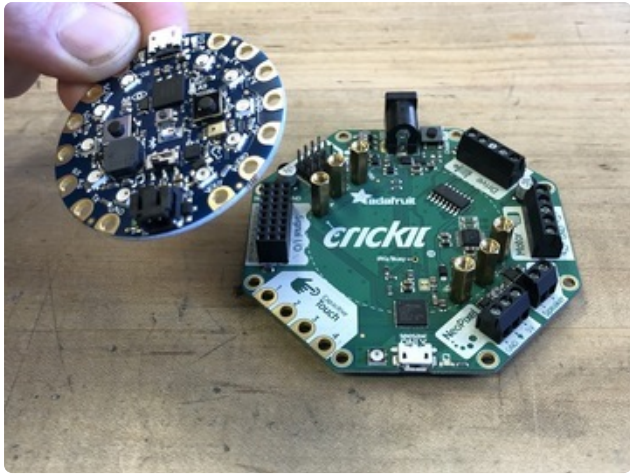
CPB to Cricket

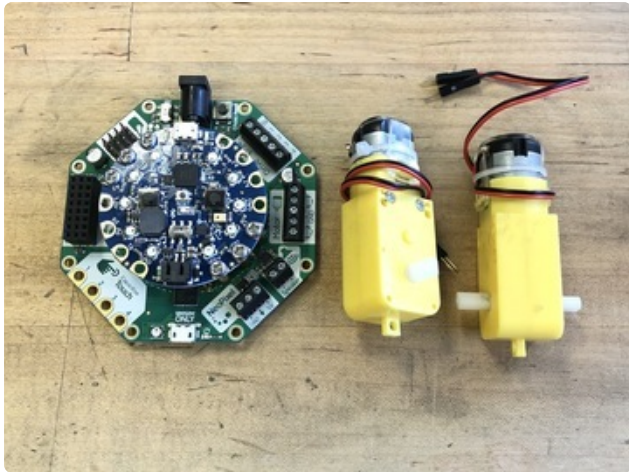
Connect the six hex standoffs to the Cricket as shown, using the screws fitted through the back of the board. Finger tight is fine, you don't need to crank them down too hard and risk damage to the board!



Then, set the Circuit Playground Bluefruit on top of the board with the shown orientation -- The Cricket DC power jack points north and the CPB USB port points north. (You may reverse both directions if you are in Australia.)







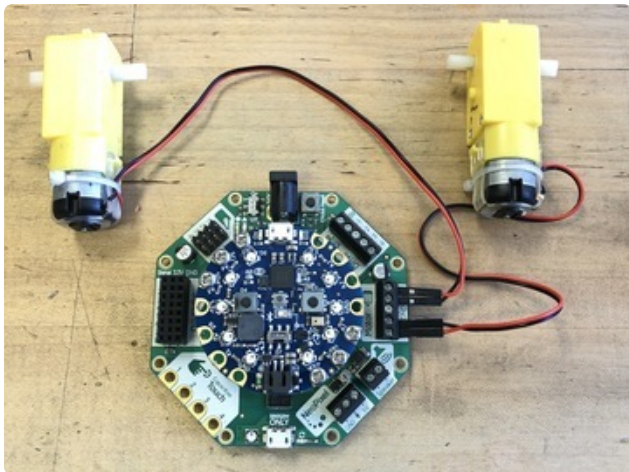
Motors to Crickit

Next, plug the motor wires into the Crickit's Motor 1 and Motor 2 ports

Left motor goes to Motor 1 port

Right motor goes to Motor 2 port

Follow the pictures for black and red wire assignments -- getting this backwards isn't a big deal, because it can always be reversed in software!

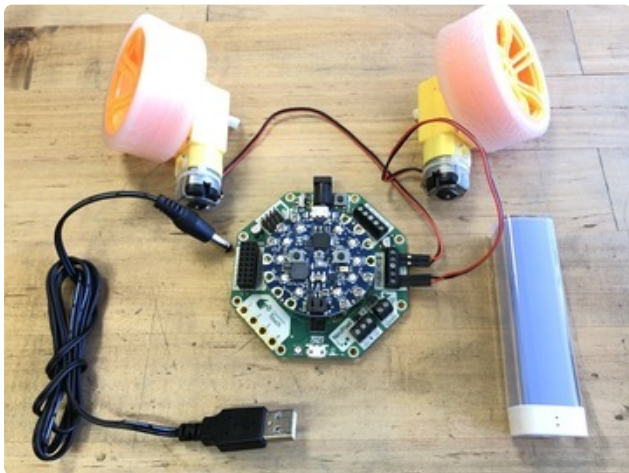


Moar Powaaaaarrrr!

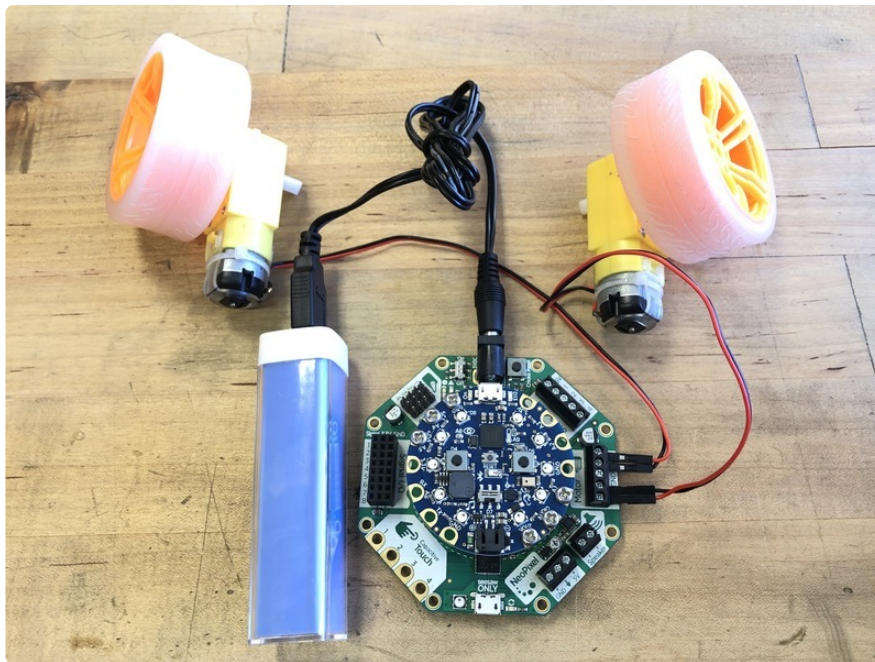
The final thing we'll need is a way to provide power. The Crickit would like to have 5VDC at 1A available for all of its duties, including powering the CPB and driving the motors.

The 5V USB phone-charger style battery is perfect, except for one thing -- how to connect it?

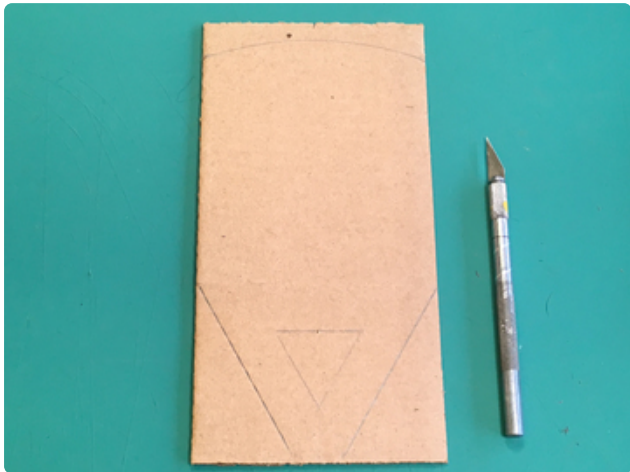
Thankfully, the USB Type-A to 2.1mm center positive cable is just the thing! Plug the USB end into the battery and the barrel plug into the Crickit's barrel jack.



Here's the final circuit ready to go!



Chassis



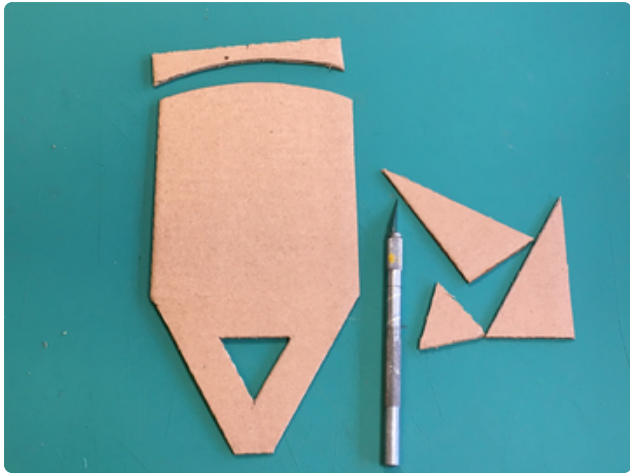
Form the Chassis

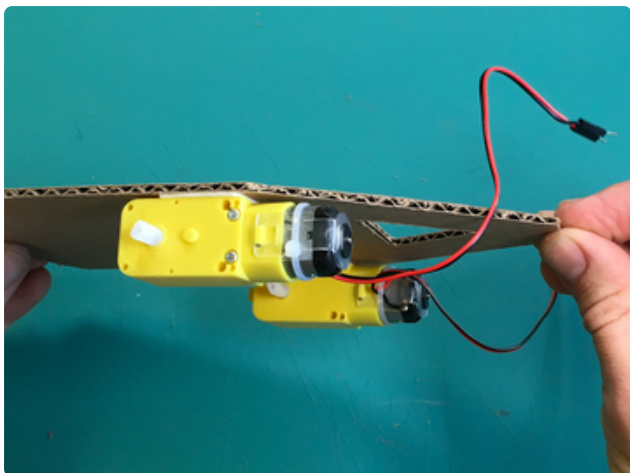
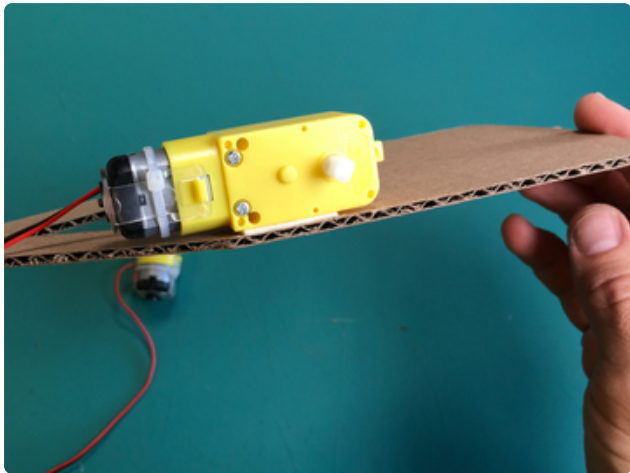
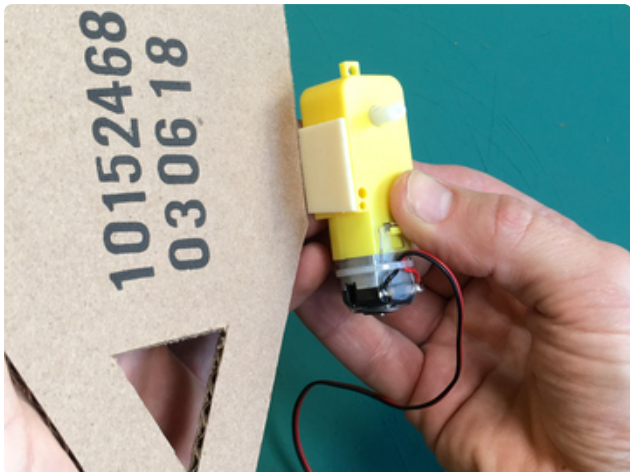
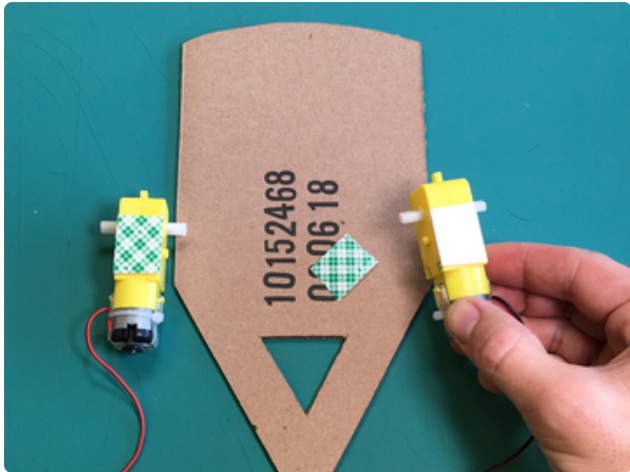
A piece of corrugated cardboard is perfect for building your turtle bot chassis!

Draw some guides for cutting your chassis -- it doesn't have to match this one exactly, just so long as there's room for the parts you can get creative with it!

Looking through the rest of this page will give you an idea of the size needed for proper parts placement.

Use a hobby knife or scissors to cut the chassis to shape

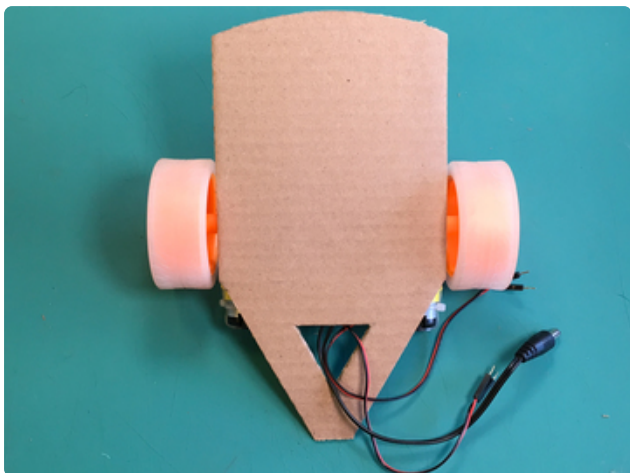
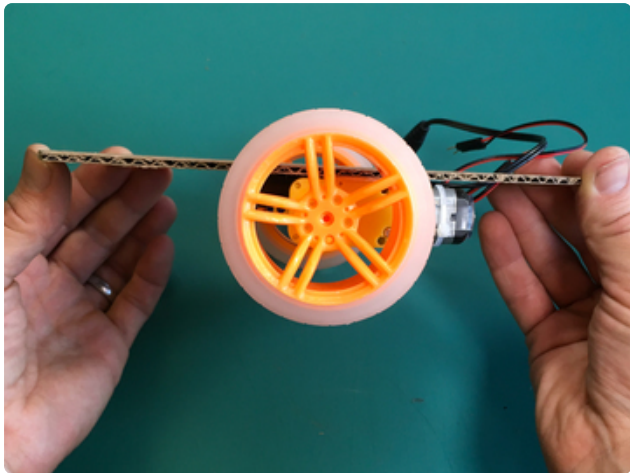
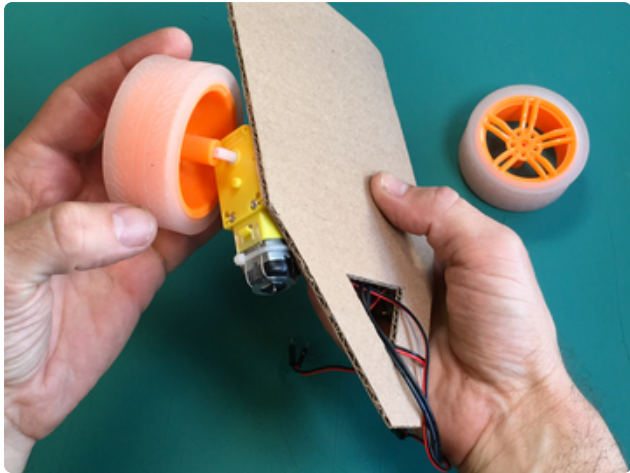
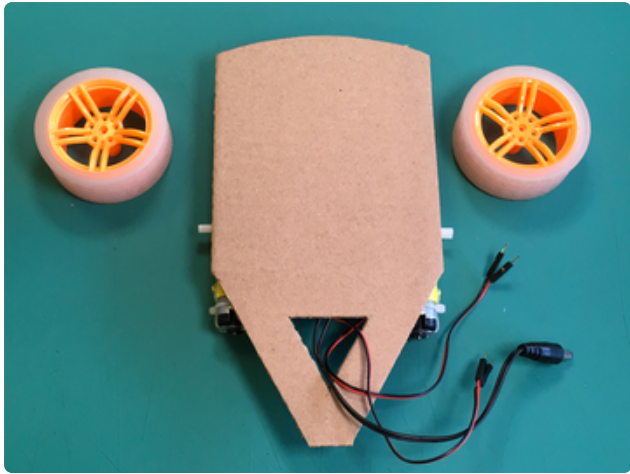




Attach Motors

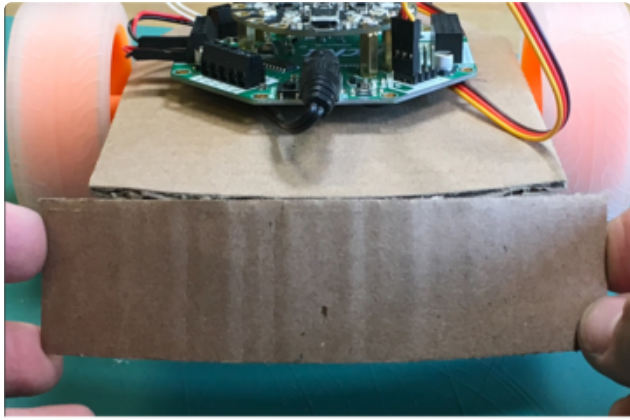
Use double stick foam tape to connect the motors to the underside of the chassis

Keep them straight and even with each other -- the cart will drive straighter than if they're offset.



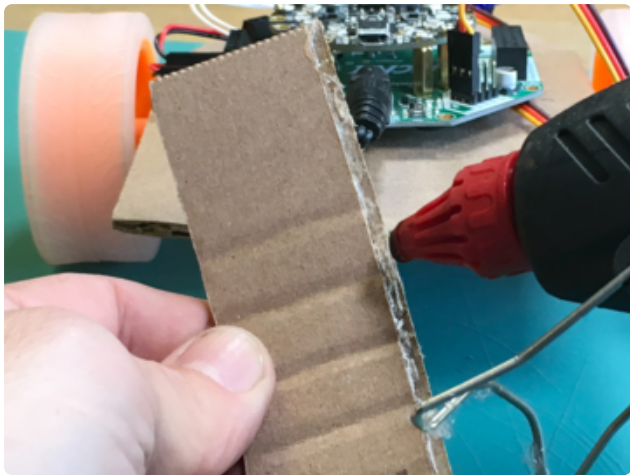
Add Wheels

Place the big honkin orange wheels onto the motor shafts as shown!



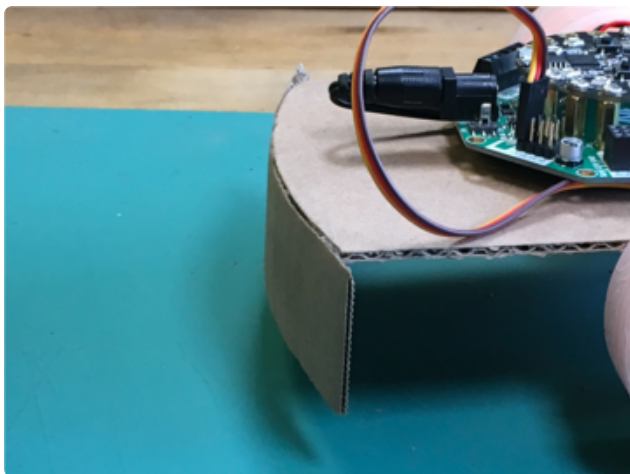
Optional Bumper

This step is optional, but you may find it helpful. You can add a cardboard bumper to the front of the bumper bot so that it won't accidentally drive over lower obstacles that are cleared by the chassis.

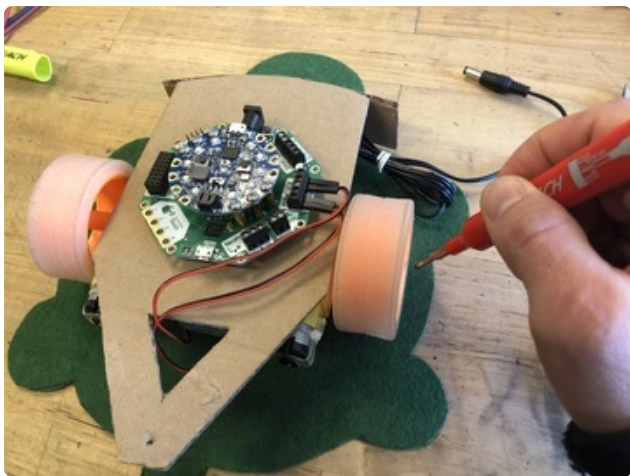


Cut a length of cardboard that is a bit wider than the cart chassis and that reaches from the front down about 3/4 of the way to the ground

Bend the cardboard to fit the curve of your cart's front



Glue the joint and then connect the two pieces, holding them in place while the glue cures



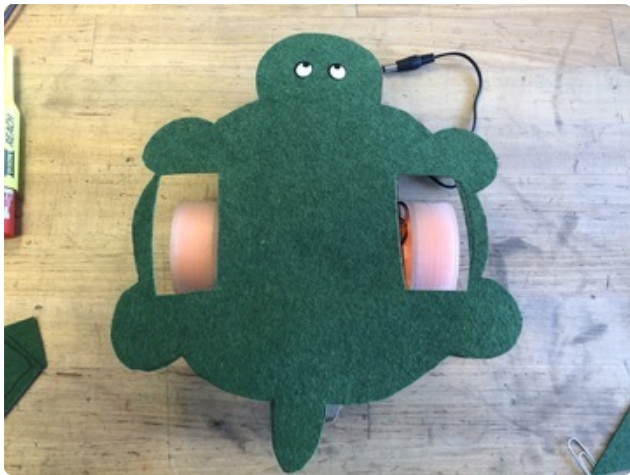
Turtle Decoration

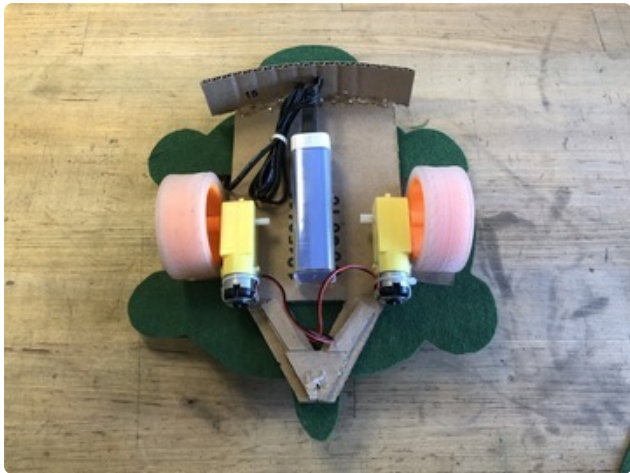
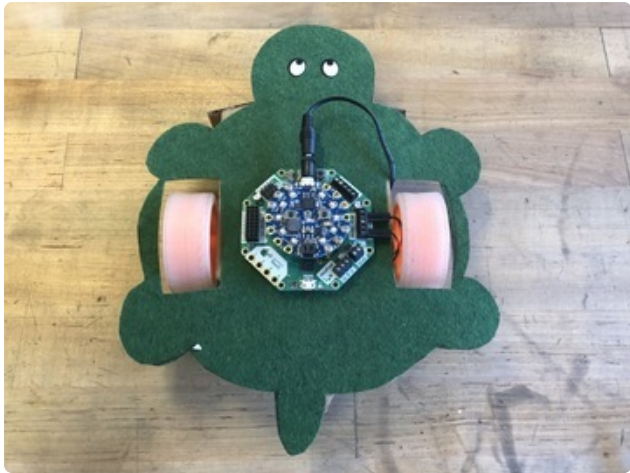
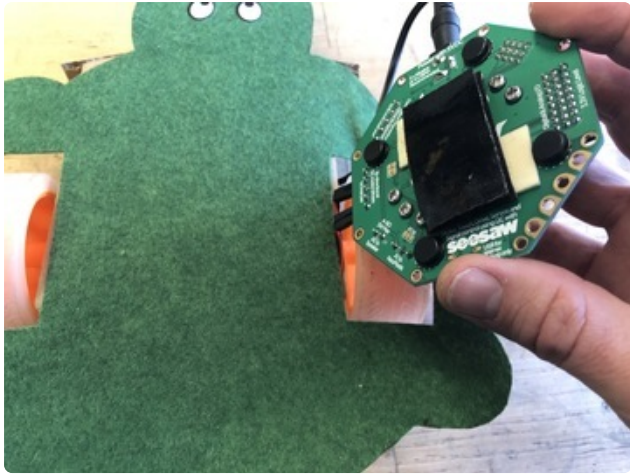
Cut out the body of the turtle from some EVA foam or green felt as shown.

Bonus points for eye stickers, googly eyes, or draw or paint on your own creation!

Use the wheels as a guide, tracing around them to create a pair of cutouts as shown here.

You can use double stick tape or another adhesive to affix the turtle pattern to the cardboard chassis.

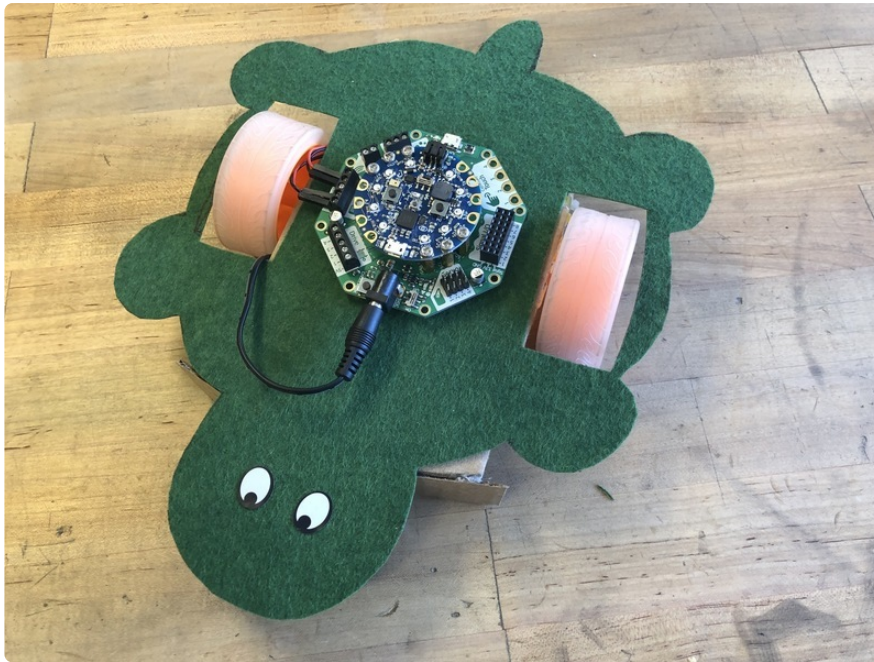


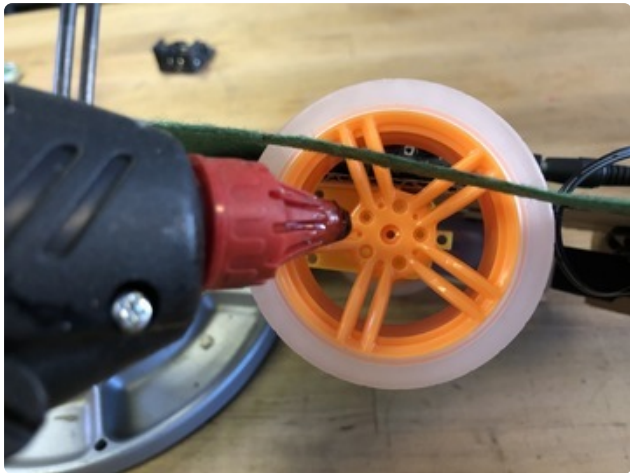
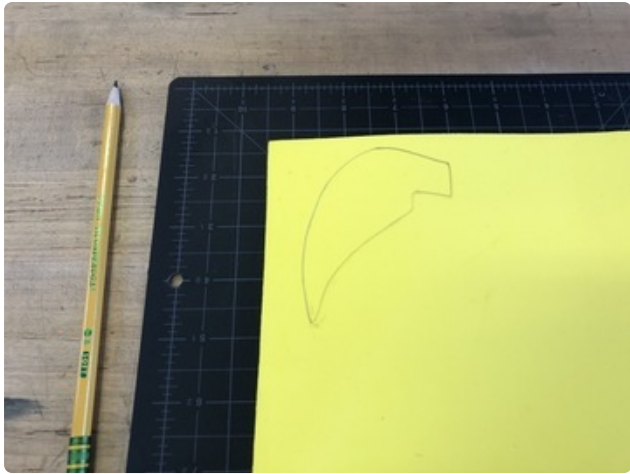


Attach Crickit

Use adhesive-backed Velcro tape to attach the CPB/Crickit board sandwich on top of the turtle layer.

Route the power cable up through a wheel well and attach the battery underneath, between the motors, using some double-stick foam tape or Velcro.





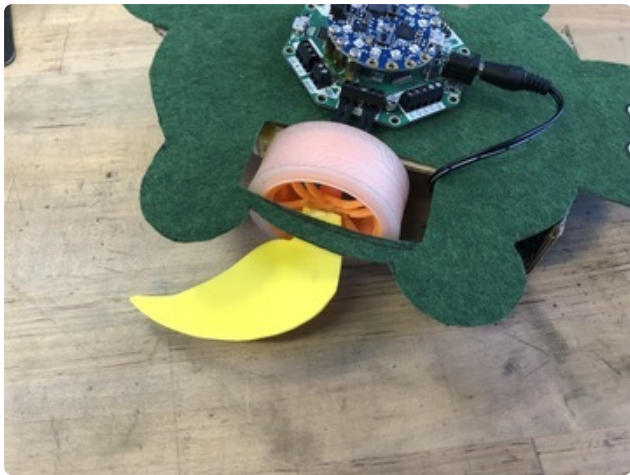
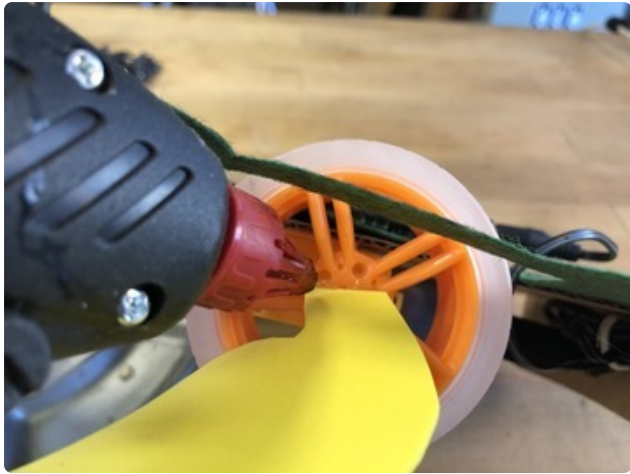
Flippers

Add to the charm of your little robot friend with some flipper feet! If you elect to add these flippers, simply trim off the back feet from the green foam/felt.

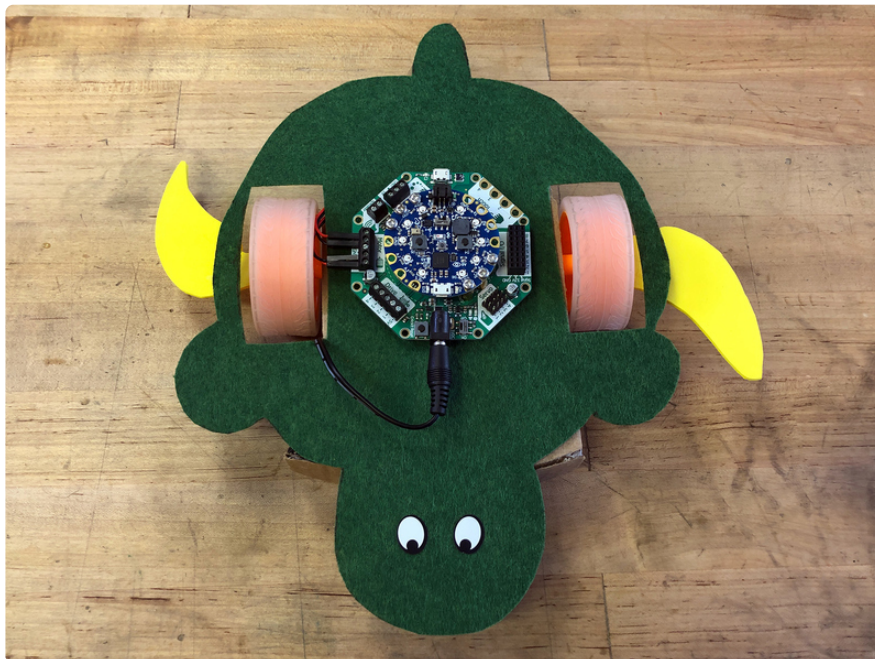
Draw out a flipper shape on some EVA foam and cut it out with scissors or a hobby knife.

Use some hot glue to affix the flipper to one of the wheel hubs as shown.

Repeat this step for the other side.



Our turtle is ready for coding!



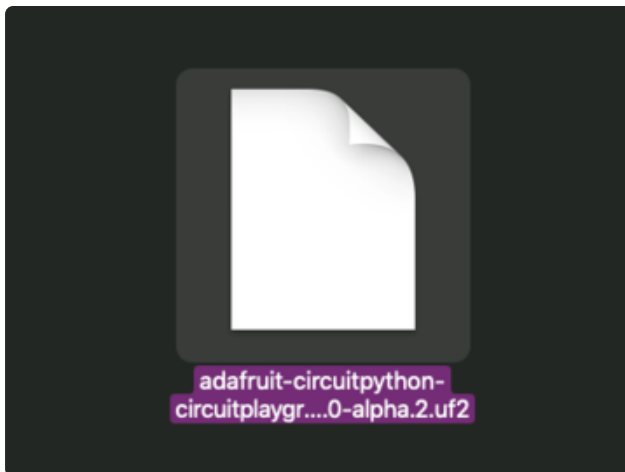
CircuitPython on Circuit Playground Bluefruit

Install or Update CircuitPython

Follow this quick step-by-step to install or update CircuitPython on your Circuit Playground Bluefruit.

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

<https://adafru.it/FNK>

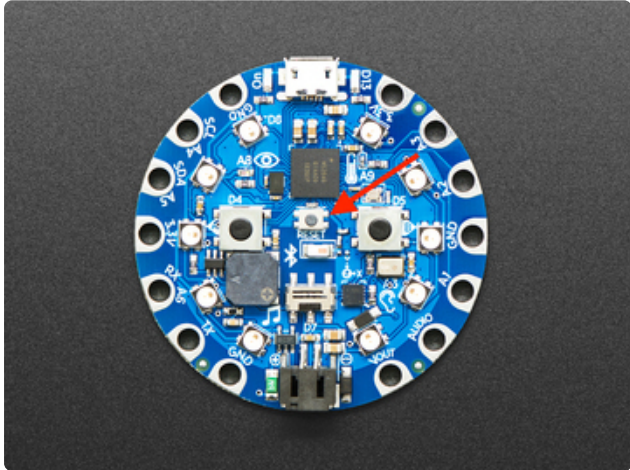


Click the link above and download the
latest UF2 file

Download and save it to your Desktop (or
wherever is handy)

Plug your Circuit Playground Bluefruit into your computer using a known-good data-capable USB cable.

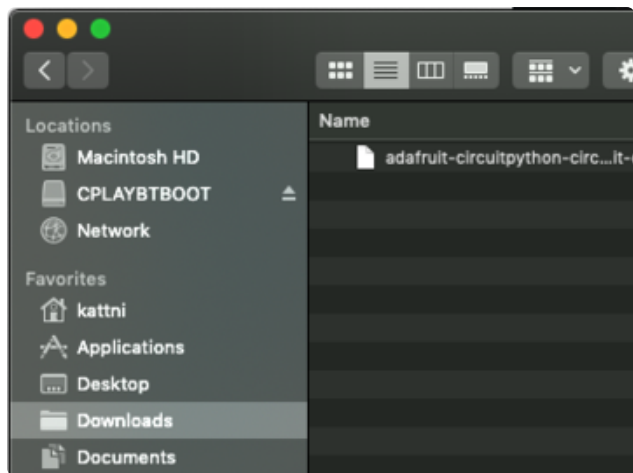
A lot of people end up using charge-only USB cables and it is very frustrating! So make sure you have a USB cable you know is good for data sync.



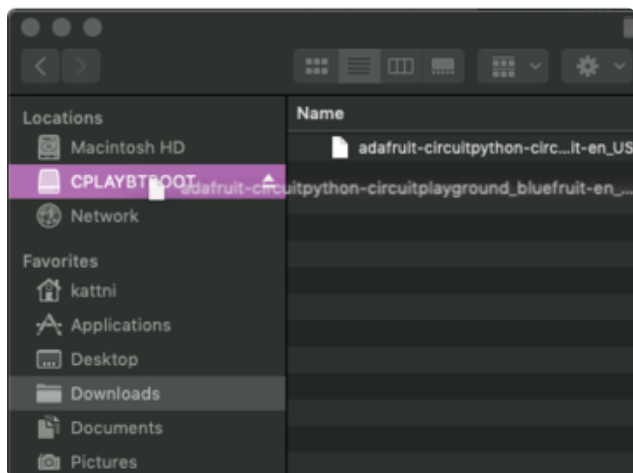
Double-click the small **Reset** button in the middle of the CPB (indicated by the red arrow in the image). The ten NeoPixel LEDs will all turn red, and then will all turn green. If they turn all red and stay red, check the USB cable, try another USB port, etc. The little red LED next to the USB connector will pulse red - this is ok!

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

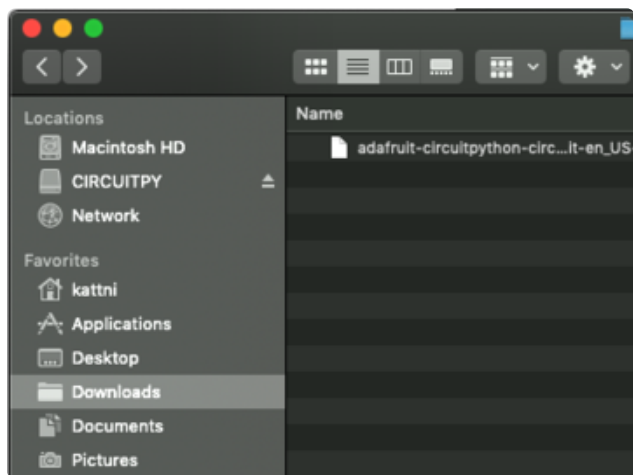
(If double-clicking doesn't do it, try a single-click!)



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



Drag the `adafruit_circuitpython_etc.uf2` file to **CPLAYBTBOOT**.



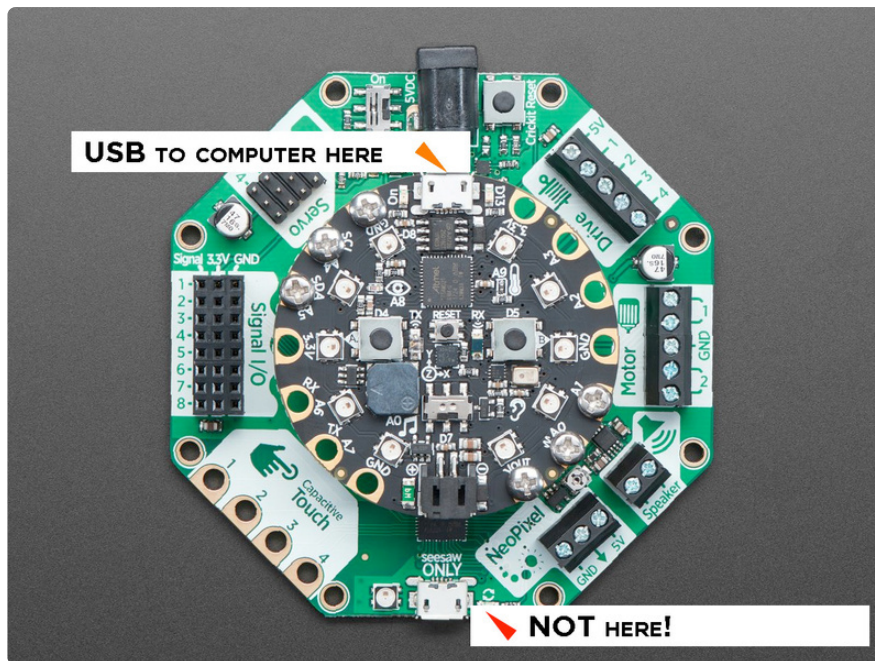
The LEDs will turn red. Then, the **CPLAYBTBOOT** drive will disappear and a new disk drive called **CIRCUITPY** will appear.

That's it, you're done! :)

Code with CircuitPython

Plug your Circuit Playground Bluefruit board into your computer with a known good USB A to USB micro B cable (good = had both data & power wires in it).

Do not plug your USB cable into the Crickit - that port is rarely used, use the one on the Circuit Playground Bluetooth board only).

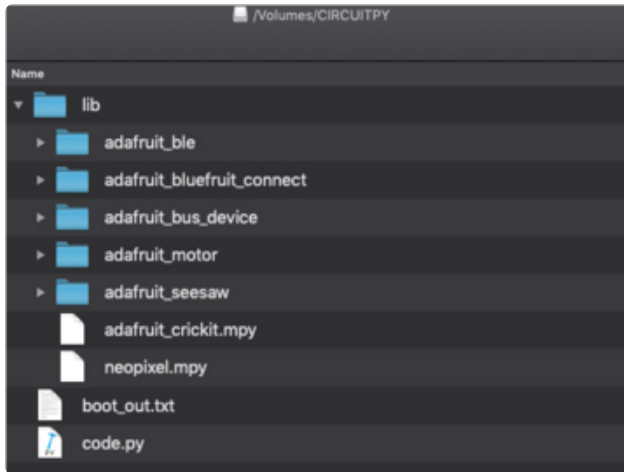


Your computer should recognize the Circuit Playground Bluefruit as a new flash drive named **CIRCUITPY**. If you see a new drive named **CPLAYBTBOOT** then press reset. If still no **CIRCUITPY** drive, go back to the CircuitPython on the Circuit Playground Bluefruit page to install CircuitPython again.

Using your computer's operating system, make a directory named **lib** in the main (root) directory of the **CIRCUITPY** drive if it is not already present. This will hold some code libraries for the project.

Libraries

Now we'll install the libraries that we need to run the rover code on the Circuit Playground Express with Crickit.



Click this link (<https://adafru.it/ENC>) to go to the **circuitpython.org** Libraries page.

Download the latest version of the Bundle library **.zip** file that matches the version of CircuitPython you're using on the board.

Uncompress the **.zip** file and then copy the following directories and **.mpy** files to the **lib** directory of the **CIRCUITPY** drive:

adafruit_ble
adafruit_bluefruit_connect
adafruit_bus_device
adafruit_motor
adafruit_seesaw
adafruit_crickit.mpy
neopixel.mpy

Your **lib** directory on the **CIRCUITPY** drive should be similar to the picture above.

The Mu Editor

Adafruit recommends using the free program Mu to edit your CircuitPython programs and save them on your Circuit Playground Bluefruit. You can use any text editor, but Mu has some handy features.

[See this page on the Circuit Playground Bluefruit guide \(https://adafru.it/GDt\)](https://adafru.it/GDt) on the steps used to install Mu.

Turtle Rover Code

The event we've all been waiting for! The actual CircuitPython code for our rover!

Copy this code and then paste it into a new document in Mu, then save it to your **CIRCUITPY** drive as **code.py**

```
# SPDX-FileCopyrightText: 2019 John Park for Adafruit Industries
#
```

```

# SPDX-License-Identifier: MIT

# Circuit Playground Bluefruit Rover
# Use with the Adafruit BlueFruit LE Connect app
# Works with CircuitPython 5.0.0-beta.0 and later
# running on an nRF52840 CPB board and Crickit

import time
import board
import digitalio
import neopixel
from adafruit_crickit import crickit

from adafruit_ble import BLERadio
from adafruit_ble.advertising.standard import ProvideServicesAdvertisement
from adafruit_ble.services.nordic import UARTService

from adafruit_bluefruit_connect.packet import Packet
# Only the packet classes that are imported will be known to Packet.
from adafruit_bluefruit_connect.button_packet import ButtonPacket
from adafruit_bluefruit_connect.color_packet import ColorPacket

# Prep the status LED on the CPB
red_led = digitalio.DigitalInOut(board.D13)
red_led.direction = digitalio.Direction.OUTPUT

ble = BLERadio()
uart_service = UARTService()
advertisement = ProvideServicesAdvertisement(uart_service)

# motor setup
motor_1 = crickit.dc_motor_1
motor_2 = crickit.dc_motor_2

FWD = 0.25
REV = -0.25

neopixels = neopixel.NeoPixel(board.NEOPIXEL, 10, brightness=0.1)
RED = (200, 0, 0)
GREEN = (0, 200, 0)
BLUE = (0, 0, 200)
PURPLE = (120, 0, 160)
YELLOW = (100, 100, 0)
AQUA = (0, 100, 100)
BLACK = (0, 0, 0)
color = PURPLE # current NeoPixel color
neopixels.fill(color)

print("BLE Turtle Rover")
print("Use Adafruit Bluefruit app to connect")
while True:
    neopixels[0] = BLACK
    neopixels.show()
    ble.start_advertising(advertisement)
    while not ble.connected:
        # Wait for a connection.
        pass
    # set a pixel blue when connected
    neopixels[0] = BLUE
    neopixels.show()
    while ble.connected:
        if uart_service.in_waiting:
            # Packet is arriving.
            red_led.value = False # turn off red LED
            packet = Packet.from_stream(uart_service)
            if isinstance(packet, ColorPacket):
                # Change the color.
                color = packet.color
                neopixels.fill(color)

```



```

# do this when buttons are pressed
if isinstance(packet, ButtonPacket) and packet.pressed:
    red_led.value = True # blink to show packet has been received
    if packet.button == ButtonPacket.UP:
        neopixels.fill(color)
        motor_1.throttle = FWD
        motor_2.throttle = FWD
    elif packet.button == ButtonPacket.DOWN:
        neopixels.fill(color)
        motor_1.throttle = REV
        motor_2.throttle = REV
    elif packet.button == ButtonPacket.RIGHT:
        color = YELLOW
        neopixels.fill(color)
        motor_2.throttle = 0
        motor_1.throttle = FWD
    elif packet.button == ButtonPacket.LEFT:
        color = YELLOW
        neopixels.fill(color)
        motor_2.throttle = FWD
        motor_1.throttle = 0
    elif packet.button == ButtonPacket.BUTTON_1:
        neopixels.fill(RED)
        motor_1.throttle = 0.0
        motor_2.throttle = 0.0
        time.sleep(0.5)
        neopixels.fill(color)
    elif packet.button == ButtonPacket.BUTTON_2:
        color = GREEN
        neopixels.fill(color)
    elif packet.button == ButtonPacket.BUTTON_3:
        color = BLUE
        neopixels.fill(color)
    elif packet.button == ButtonPacket.BUTTON_4:
        color = PURPLE
        neopixels.fill(color)
# do this when some buttons are released
elif isinstance(packet, ButtonPacket) and not packet.pressed:
    if packet.button == ButtonPacket.UP:
        neopixels.fill(RED)
        motor_1.throttle = 0
        motor_2.throttle = 0
    if packet.button == ButtonPacket.DOWN:
        neopixels.fill(RED)
        motor_1.throttle = 0
        motor_2.throttle = 0
    if packet.button == ButtonPacket.RIGHT:
        neopixels.fill(RED)
        motor_1.throttle = 0
        motor_2.throttle = 0
    if packet.button == ButtonPacket.LEFT:
        neopixels.fill(RED)
        motor_1.throttle = 0
        motor_2.throttle = 0

```

How the Code Works

The first part of the program sets up a `UARTServer`, which is a BLE service that wirelessly sends streams of characters between a BLE device and a client computer (which could be a phone or tablet). The program then starts advertising its UART service to anyone listening. The client computer receives the advertisement, and tells the user it's available. The user can then choose to connect as a client of the service.

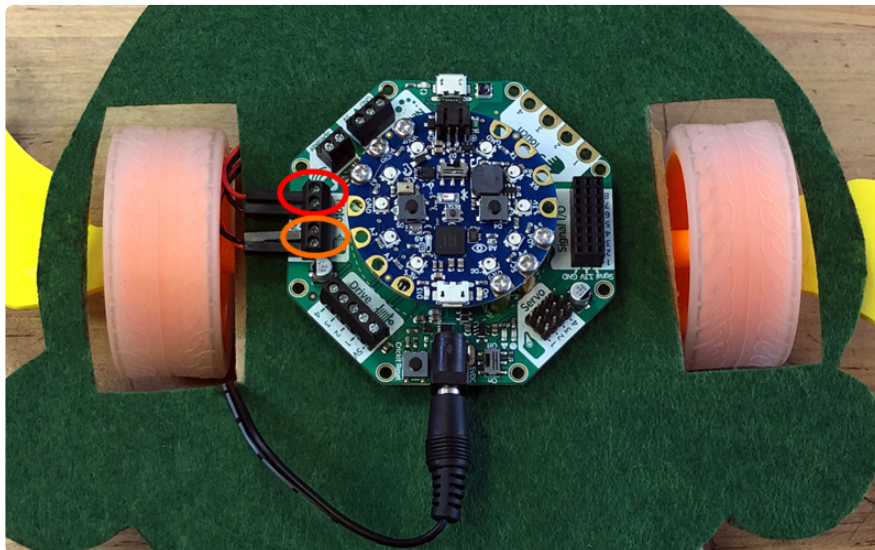
Once a connection is established, the program code waits for incoming command packets from the client, either a `ColorPacket` or a `ButtonPacket`. When one arrives, the program waits to see which button has been pressed or which color value is sent and then either runs the motors or changes the colors on the NeoPixels.

Motor Movement

We set up a couple of variables -- `FWD` and `REV` -- that tell the program which base values to send for forward and reverse. Depending on how you wired the motors, and which way you want to consider "forward" on the cart, you can adjust these values anywhere from `-1.0` to `1.0`.

When you test the rover, if it goes backwards when you press the up button, you can switch the `FWD` variable from `0.25` to `-0.25` and flip the `REV` value from `-0.25` to `0.25`. The alternative is to swap which wire -- red or black -- is going into each port of the Motor 1 and Motor 2 terminals on the Crickit.

If you find the rover spinning in circle when you press up or down on the remote app, then one of the motors is wired "backwards". Simply swap the two wires of that motor into each others' ports on the Crickit!

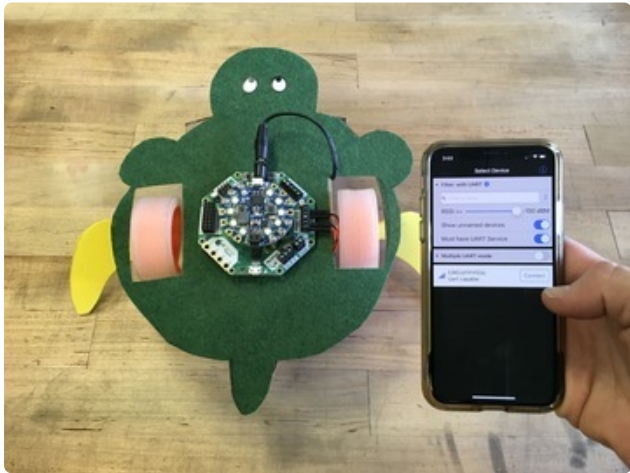


Control the Rover with the Bluefruit LE Connect App

Download the App

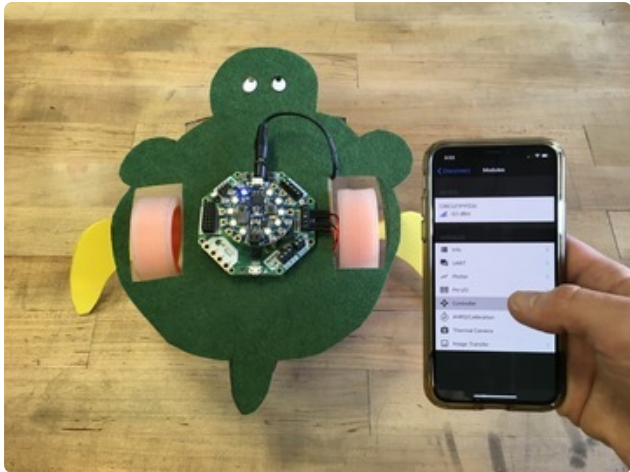
To control the motors and NeoPixels from a client device (phone or tablet), you use the free Adafruit [Bluefruit LE Connect App](https://adafru.it/iCi) (<https://adafru.it/iCi>). Install it from the [Apple App Store](https://adafru.it/BYj) (<https://adafru.it/BYj>) or [Google Play App Store](https://adafru.it/f4G) (<https://adafru.it/f4G>).

Connect to CPB



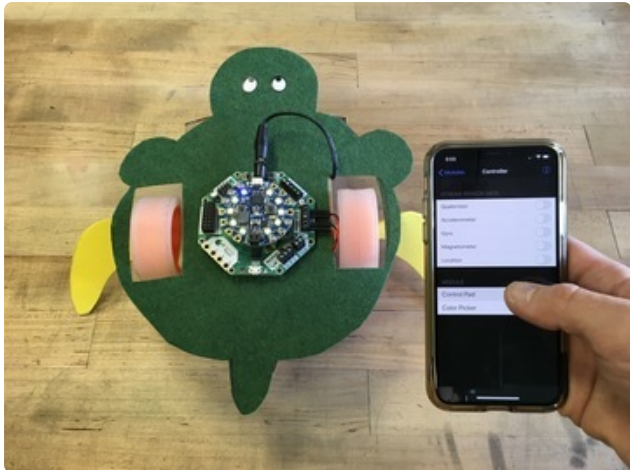
With the Crickit plugged into DC power and turned on with its on-board power switch, the CPB will also get power and automatically start running your code. Then start up the Bluefruit LE Connect app, and make sure it's in Central Mode (left button on the bottom). When you start the app, you should see a device named **CIRCU** or **CIRCUITPY**. If there's a long list of devices, you can shorten it by turning on the "**Must have UART Service**" switch.

To connect to the bot, touch the **Connect** button.

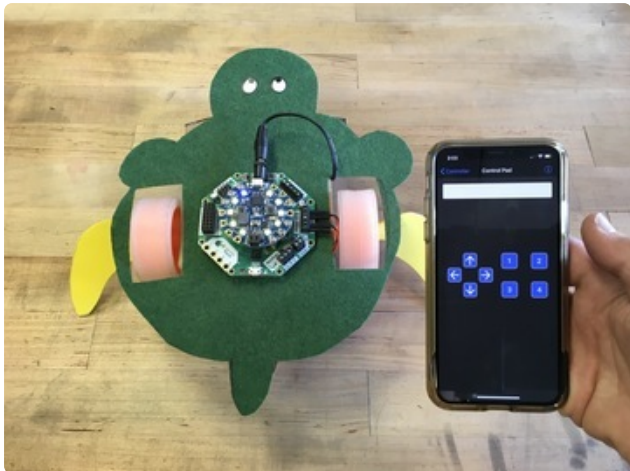


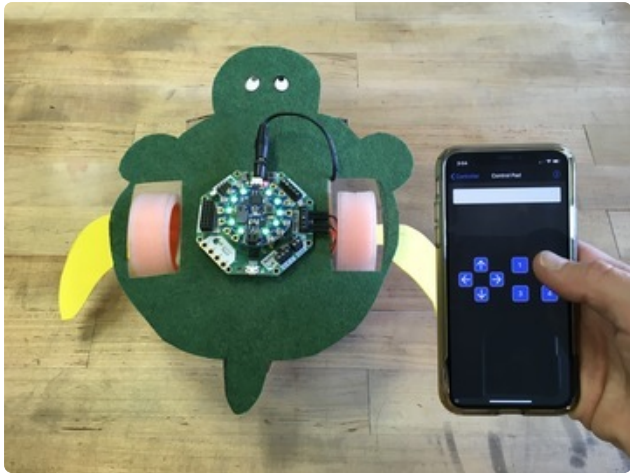
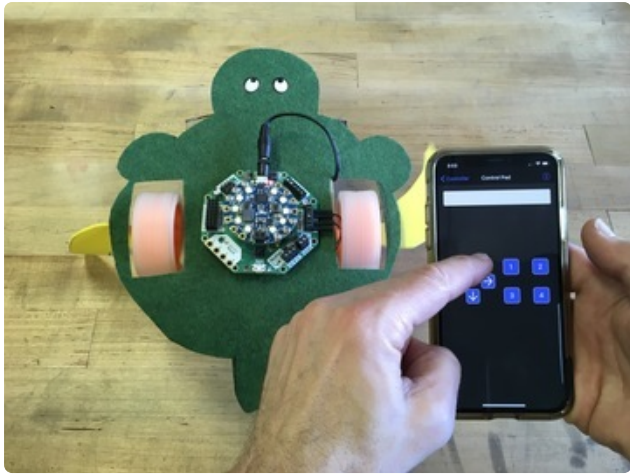
Drive and Light

Once connected, you can use the Controllers -> Control Pad to drive with the arrow buttons, stop with the 1 button, and change to some preset colors on 2, 3, & 4



The incoming packet for any arrow button press event will tell the motors to throttle to the preset speed variable, such as `motor_1.throttle = FWD`. When you release the button, the button packet will tell the motor to stop. So, you can press and hold to go, and then release to stop, it's easy and fun!





Here's our turtle up on blocks to demonstrate the remote control:

And here it is with wheels (flippers!) down: