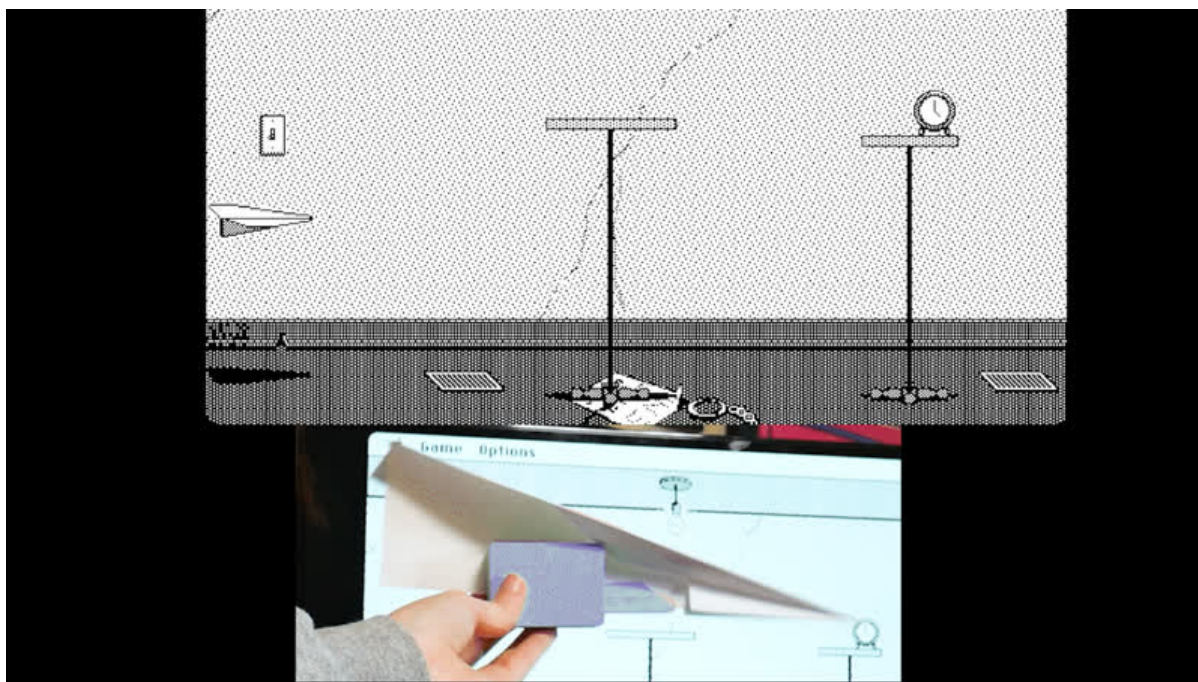




# Glider Paper Airplane Controller

Created by Liz Clark



<https://learn.adafruit.com/glider-paper-airplane-controller>

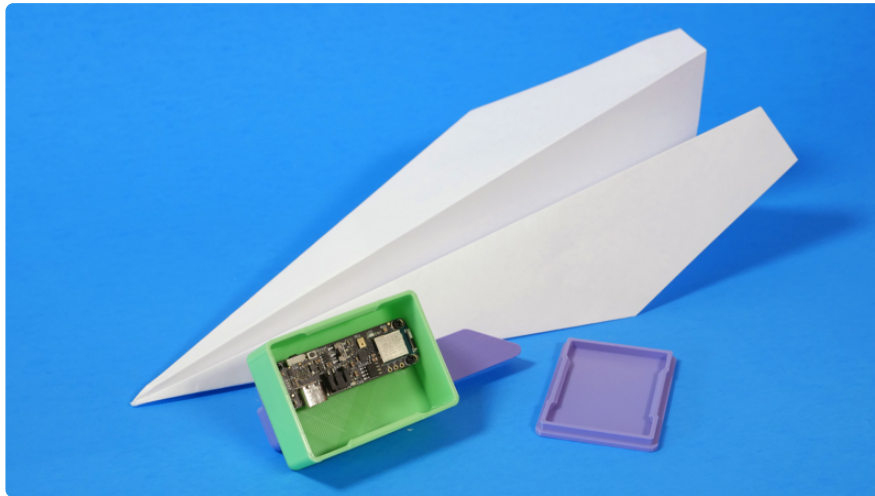
Last updated on 2024-06-03 03:35:59 PM EDT

# Table of Contents

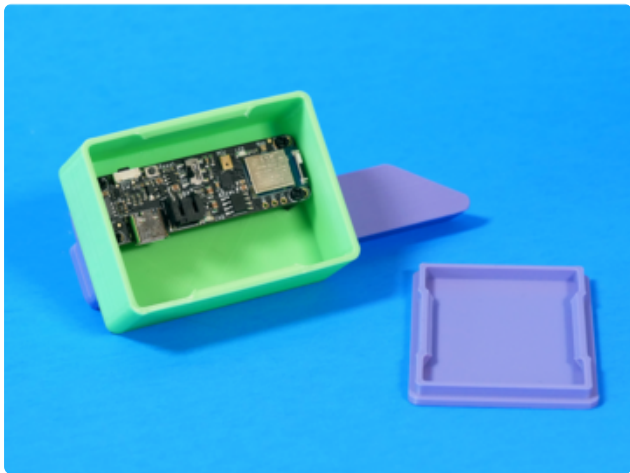
Overview	3
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Prerequisite Guides</a></li><li>• <a href="#">Parts</a></li></ul>	
3D Printing	5
<hr/>	
CircuitPython	7
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">CircuitPython Quickstart</a></li></ul>	
Coding the Glider Paper Airplane Controller	9
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Upload the Code and Libraries to the nRF52840 LED Glasses Driver</a></li><li>• <a href="#">Other Board Options</a></li><li>• <a href="#">How the CircuitPython Code Works</a></li></ul>	
Fold Your Airplane	13
<hr/>	
Assembly	16
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Solder the Lipo Charge Jumper</a></li><li>• <a href="#">Mount the Board</a></li></ul>	
Usage	20
<hr/>	
<ul style="list-style-type: none"><li>• <a href="#">Game Software</a></li><li>• <a href="#">Game Play</a></li></ul>	

---

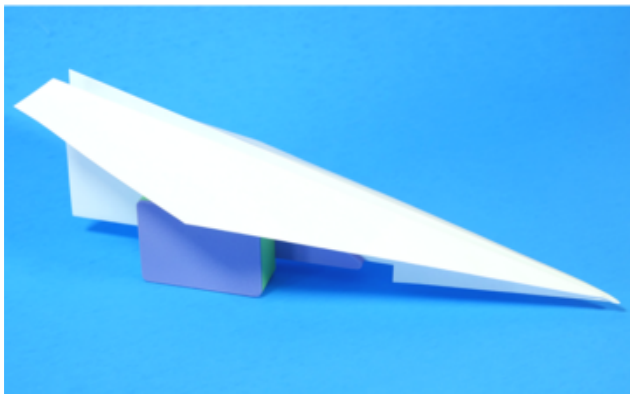
# Overview



[Glider is a classic Macintosh game \(https://adafru.it/ZDL\)](https://adafru.it/ZDL) where you guide a paper airplane through a series of rooms. What could make this game even better? Controlling it with an actual paper airplane! You'll combine your paper airplane with an nRF52840 LED Glasses Driver board running CircuitPython to guide your glider to glory.



The nRF52840 LED Glasses Driver has Bluetooth and an onboard accelerometer. The LIS3DH accelerometer detects the angle of the paper airplane. The CircuitPython code converts the angle to a keypress to control the game over BLE.



There is a 3D printed part for your paper airplane to slot into. The board has a case that attaches to the paper airplane holder.

## Prerequisite Guides

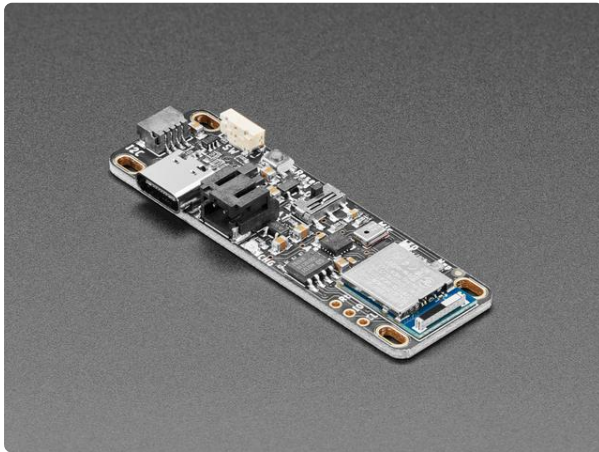
### Adafruit EyeLights LED Glasses and Driver

<https://adafru.it/VnF>

### Adafruit LIS3DH Triple-Axis Accelerometer Breakout

<https://adafru.it/AJo>

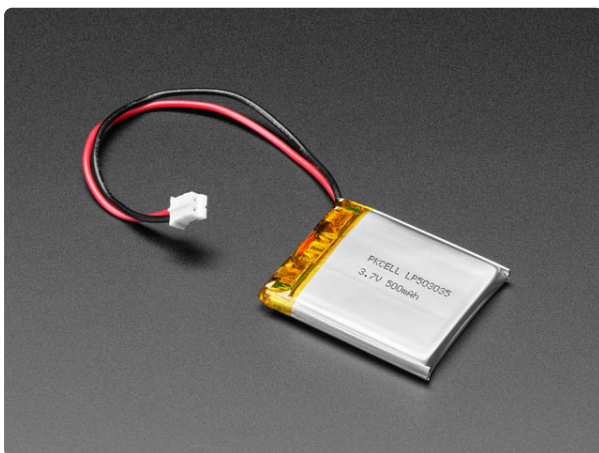
## Parts



### 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 - 3.7v 500mAh

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/1578>



### Pink and Purple Woven USB A to USB C Cable - 1 meter long

This cable is not only super-fashionable, with a woven pink and purple Blinka-like pattern, it's also made for USB C for our modernized breakout boards, Feathers, and...

<https://www.adafruit.com/product/5153>



### USB C to USB C Cable - USB 3.1 Gen 4 with E-Mark - 1 meter long

As technology changes and adapts, so does Adafruit! Rather than the regular USB A, this cable has USB C to USB C plugs! USB C is the latest...

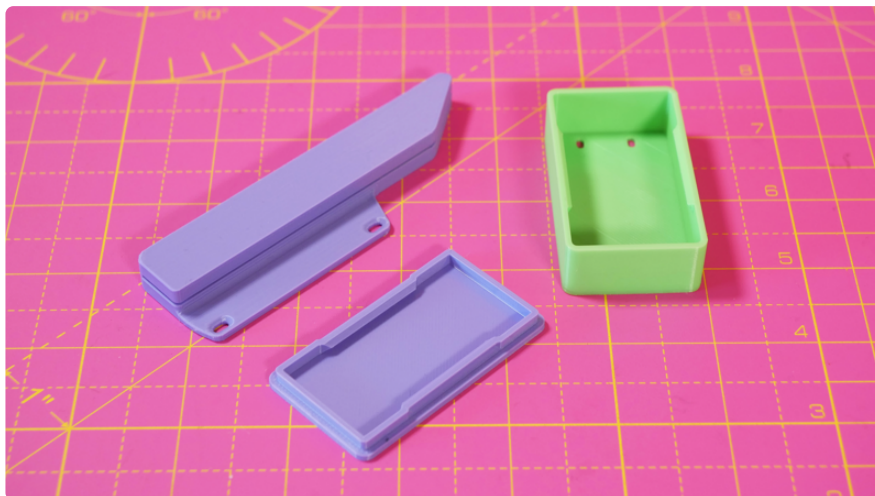
<https://www.adafruit.com/product/4199>

1 x 320PCS M2 Male Female Nylon Hex Spacer  
Standoff Screw Nut Kit  
M2 hardware

[https://www.amazon.com/dp/B07D78PFQL?psc=1&ref=ppx\\_yo2ov\\_dt\\_b\\_product\\_details](https://www.amazon.com/dp/B07D78PFQL?psc=1&ref=ppx_yo2ov_dt_b_product_details)

---

## 3D Printing





The controller may be housed in a 3D printed case, described below. The case consists of three parts: a top lid, a board case and a paper airplane holder. All parts print with no supports.

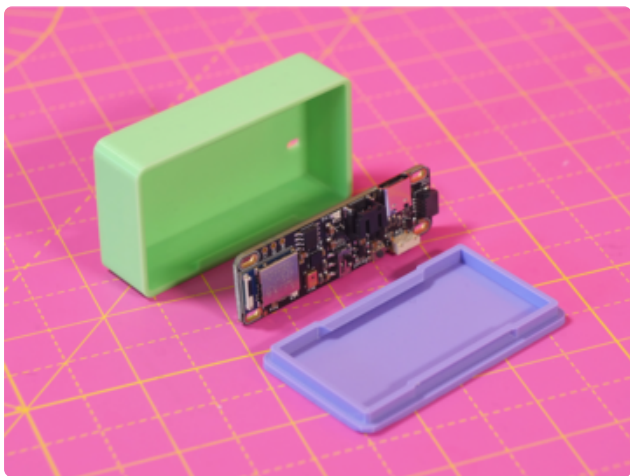
The STL files can be downloaded directly here or from Thingiverse.

**paperAirplaneControllerSTL.zip**

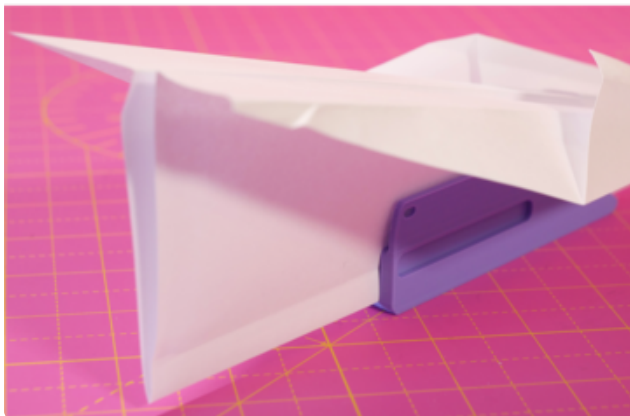
<https://adafru.it/ZD7>

**Thingiverse download**

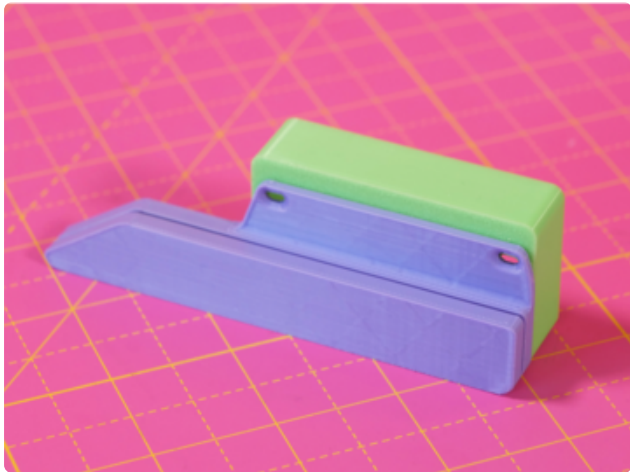
<https://adafru.it/ZD8>



The nRF52840 LED Glasses Driver mounts to the board case using M2 screws and stand-offs.



The paper airplane holder has a slot for your paper airplane to sit in.



The paper airplane holder has mounting holes to attach to the board case.

---

## CircuitPython

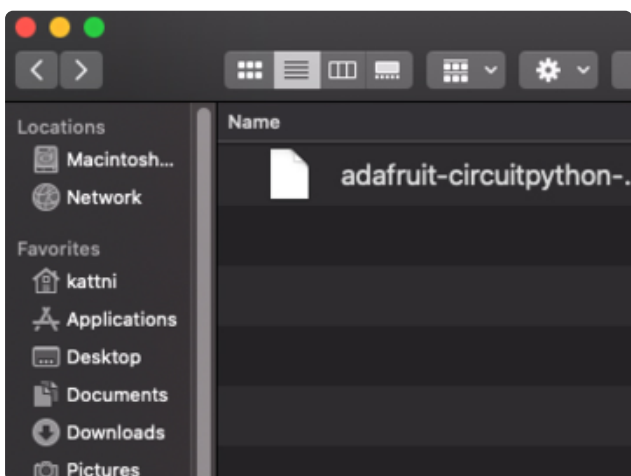
[CircuitPython \(https://adafru.it/tB7\)](https://adafru.it/tB7) is a derivative of [MicroPython \(https://adafru.it/BeZ\)](https://adafru.it/BeZ) designed to simplify experimentation and education on low-cost microcontrollers. It makes it easier than ever to get prototyping by requiring no upfront desktop 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.

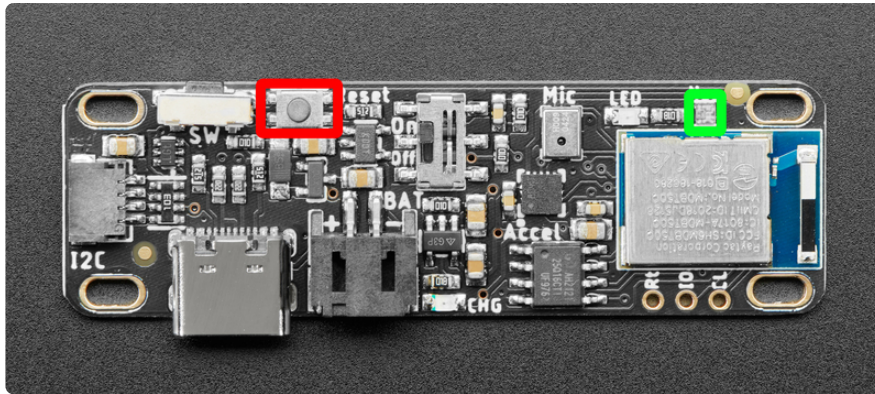
Download the latest version of  
CircuitPython for this board via  
[circuitpython.org](https://adafru.it/VcN)

<https://adafru.it/VcN>



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

Save it wherever is convenient for you.



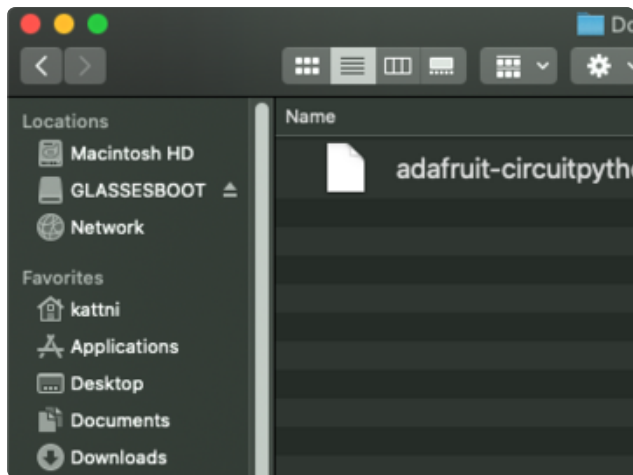
Plug your board into your computer, using a known-good data-sync cable, directly, or via an adapter if needed.

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.

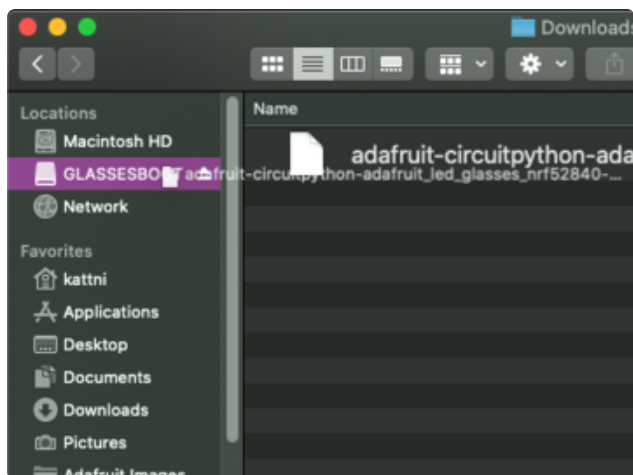
If double-clicking doesn't work the first time, try again. Sometimes it can take a few tries to get the rhythm right!

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.**

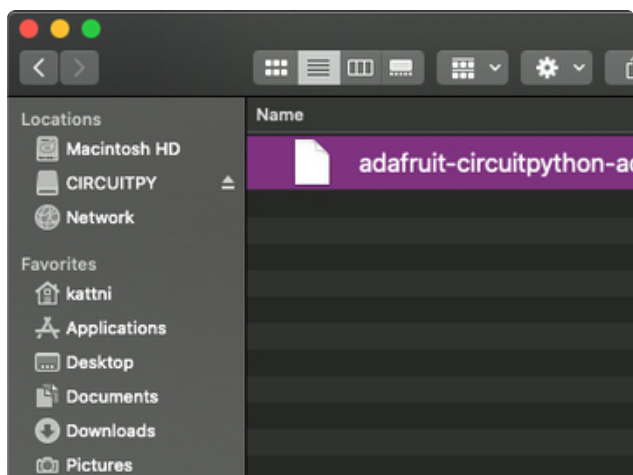




You will see a new disk drive appear called **GLASSESBOOT**.



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!

## Coding the Glider Paper Airplane Controller

Once you've finished setting up your nRF52840 LED Glasses Driver with CircuitPython, you can access the code and necessary libraries by downloading the Project Bundle.

To do this, click on the **Download Project Bundle** button in the window below. It will download as a zipped folder.

```

# SPDX-FileCopyrightText: 2022 Liz Clark for Adafruit Industries
# SPDX-License-Identifier: MIT

import time
import board
import adafruit_lis3dh
import simpleio
import adafruit_ble
from adafruit_ble.advertising import Advertisement
from adafruit_ble.advertising.standard import ProvideServicesAdvertisement
from adafruit_ble.services.standard.hid import HIDService
from adafruit_hid.keyboard import Keyboard
from adafruit_hid.keycode import Keycode

# I2C setup
i2c = board.I2C() # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C() # For using the built-in STEMMMA QT connector on a
microcontroller
lis3dh = adafruit_lis3dh.LIS3DH_I2C(i2c)

# range of LIS3DH
lis3dh.range = adafruit_lis3dh.RANGE_2_G

# BLE HID setup
hid = HIDService()

advertisement = ProvideServicesAdvertisement(hid)
advertisement.appearance = 961
scan_response = Advertisement()
scan_response.complete_name = "CircuitPython HID"

# BLE instance
ble = adafruit_ble.BLERadio()

# keyboard HID setup
keyboard = Keyboard(hid.devices)

# BLE advertisement
if not ble.connected:
    print("advertising")
    ble.start_advertising(advertisement, scan_response)
else:
    print("connected")
    print(ble.connections)

while True:
    while not ble.connected:
        pass
    # while BLE connected
    while ble.connected:
        # read LIS3DH
        x, y, z = [
            value / adafruit_lis3dh.STANDARD_GRAVITY for value in
lis3dh.acceleration
        ]
        # map Y coordinate of LIS3DH
        mapped_y = simpleio.map_range(y, -1.1, 1.1, 0, 3)
        # convert mapped value to an integer
        plane = int(mapped_y)

        # if you're tilting down...
        if plane == 0:
            # send R, glider moves right
            keyboard.press(Keycode.R)
            # debug
            # print("right")
        # if there's no tilt...
        if plane == 1:

```

```

        # release all keys, send nothing to glider
        keyboard.release_all()
        # debug
        # print("none")
# if you're tilting up...
if plane == 2:
    # send L, glider moves left
    keyboard.press(Keycode.L)
    # debug
    # print("left")
    time.sleep(0.01)
# if BLE disconnects, begin advertising again
ble.start_advertising(advertisement)

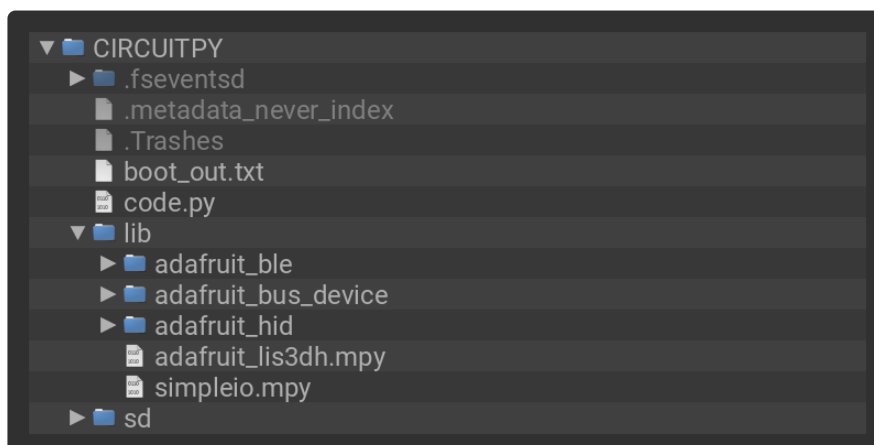
```

## Upload the Code and Libraries to the nRF52840 LED Glasses Driver

After downloading the Project Bundle, plug your nRF52840 LED Glasses Driver into the computer's USB port with a known good USB data+power cable. You should see a new flash drive appear in the computer's File Explorer or Finder (depending on your operating system) called **CIRCUITPY**. Unzip the folder and copy the following items to the nRF52840 LED Glasses Driver's **CIRCUITPY** drive.

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

Your nRF52840 LED Glasses Driver **CIRCUITPY** drive should look like this after copying the **lib** folder and the **code.py** file.



## Other Board Options

You could also use the [Feather nRF52840 Sense \(https://adafru.it/L6d\)](https://adafru.it/L6d) board, since it also has BLE and an onboard accelerometer. The Feather nRF52840 Sense's accelerometer is a LSM6DS33, different from the nRF52840 LED Glasses Driver's

LIS3DH. As a result, you would need to change the code to use the LSM6DS33 library.

Additionally, the [LIS3DH accelerometer \(https://adafru.it/AJo\)](https://adafru.it/AJo) is available as a STEMMA board. You could use that breakout with any CircuitPython compatible board with the code included on this page.

## How the CircuitPython Code Works

Before the loop, the LIS3DH accelerometer, HID keyboard and BLE are setup.

```
# I2C setup
i2c = board.I2C()
lis3dh = adafruit_lis3dh.LIS3DH_I2C(i2c)

# range of LIS3DH
lis3dh.range = adafruit_lis3dh.RANGE_2_G

# BLE HID setup
hid = HIDService()

advertisement = ProvideServicesAdvertisement(hid)
advertisement.appearance = 961
scan_response = Advertisement()
scan_response.complete_name = "CircuitPython HID"

# BLE instance
ble = adafruit_ble.BLERadio()

# keyboard HID setup
keyboard = Keyboard(hid.devices)

# BLE advertisement
if not ble.connected:
    print("advertising")
    ble.start_advertising(advertisement, scan_response)
else:
    print("connected")
    print(ble.connections)
```

## Reading and Mapping Y

The accelerometer's value is read in the loop. For the purposes of controlling Glider, you only need to look at the **y** axis. The **y** axis will tell you whether you are tilting the board up or down.

The range of **y** is mapped to a range of **0** to **3** and then converted to an integer. As a result, when you're tilting the board down **plane** will be **0**. When you're not tilting the board at all, **plane** will be **1**. When you're tilting the board up, **plane** will be **2**.

```
x, y, z = [
    value / adafruit_lis3dh.STANDARD_GRAVITY for value in
    lis3dh.acceleration
```

```

]
# map Y coordinate of LIS3DH
mapped_y = simpleio.map_range(y, -1.1, 1.1, 0, 3)
# convert mapped value to an integer
plane = int(mapped_y)

```

## Sending Key Presses

With the `y` axis mapped, you have three states for the tilt of your board: up, none and down. Each of these states has an assigned keyboard command. When you are tilting the board down, `Keycode.R` is sent. This moves the glider right in the game. When you are tilting the board up, `Keycode.L` is sent. This moves the glider left in the game. When you aren't tilting the board at all, all keys are released with the `keyboard.release_all()` function.

If you wanted to change the keycodes, you would edit this portion of the code.

```

# if you're tilting down...
if plane == 0:
    # send R, glider moves right
    keyboard.press(Keycode.R)
    # debug
    # print("right")
# if there's no tilt...
if plane == 1:
    # release all keys, send nothing to glider
    keyboard.release_all()
    # debug
    # print("none")
# if you're tilting up...
if plane == 2:
    # send L, glider moves left
    keyboard.press(Keycode.L)
    # debug
    # print("left")

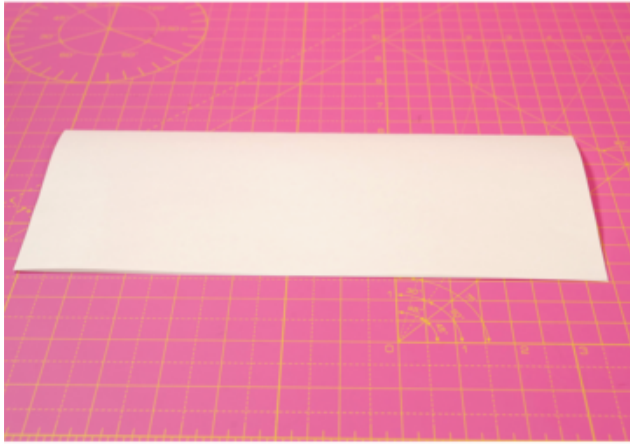
```

---

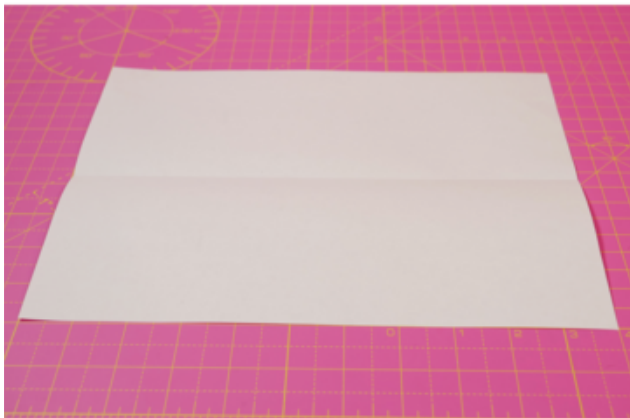
## Fold Your Airplane

There are many ways to fold a [paper airplane \(https://adafru.it/ZBY\)](https://adafru.it/ZBY). In the Wikipedia article on paper airplanes, there is a [classic design \(https://adafru.it/ZBZ\)](https://adafru.it/ZBZ) that you can fold that is detailed in the following section.

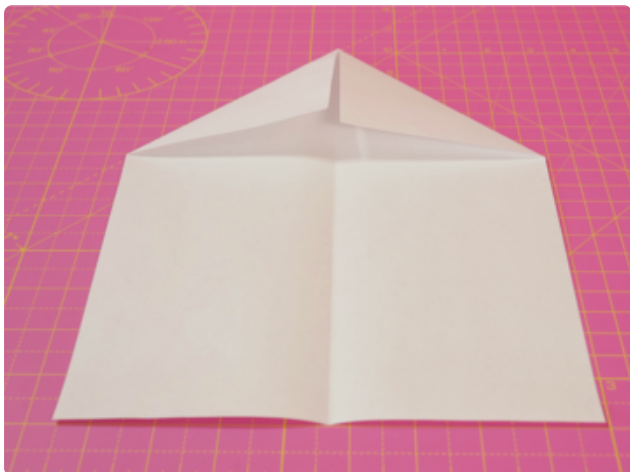




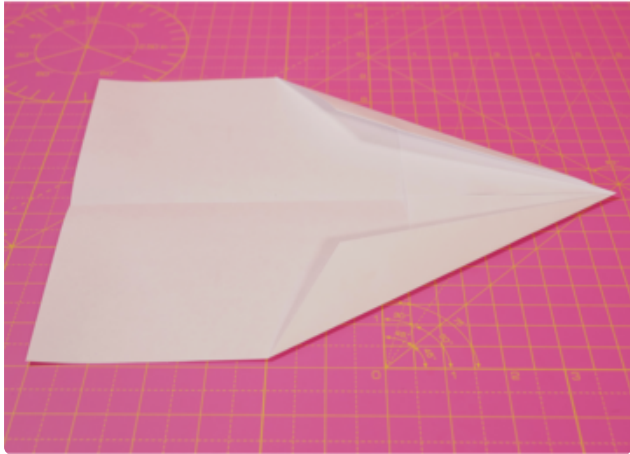
Fold your piece of paper in half length-wise.



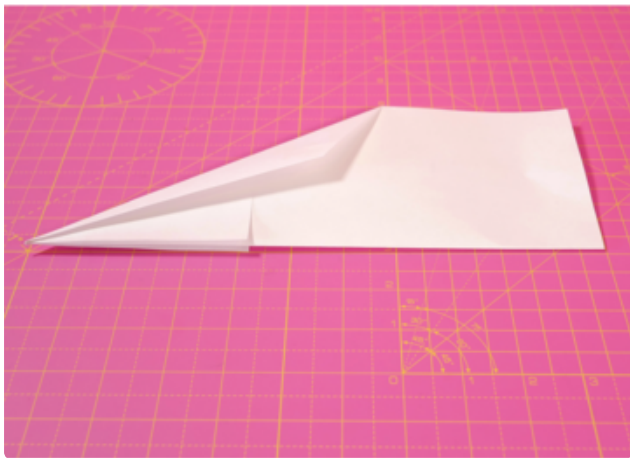
Unfold your piece of paper.



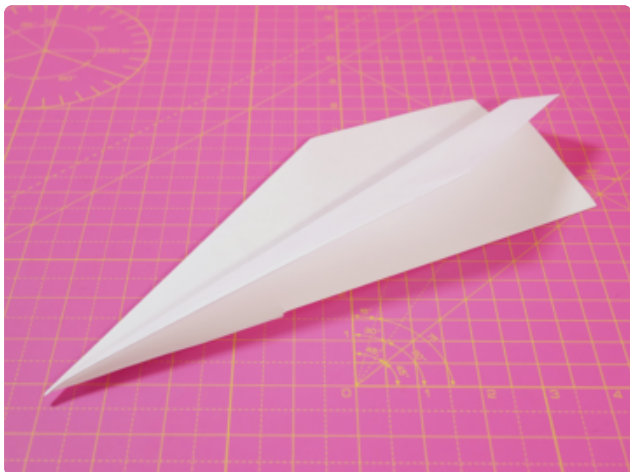
Fold the top two corners towards the center fold.



To make the wings, fold the corners in half towards the center. Your plane should have a sharp point.

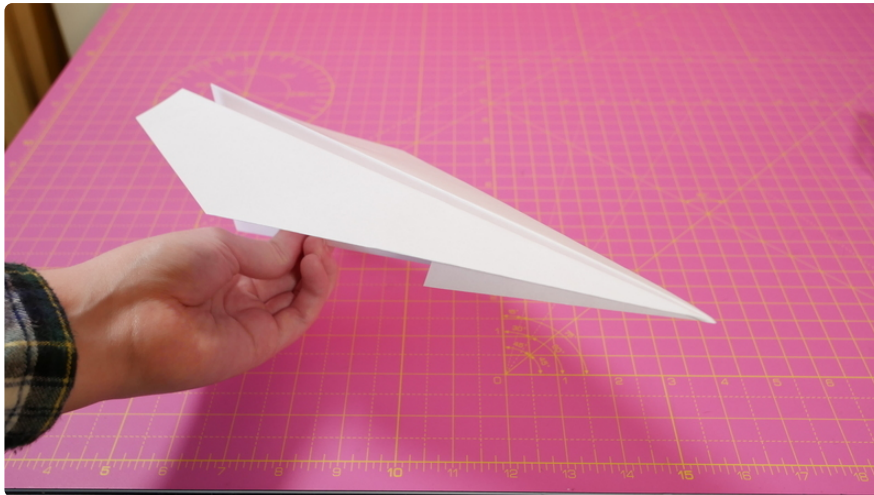


Flip the paper over. Fold the plane in half again using the original center fold.



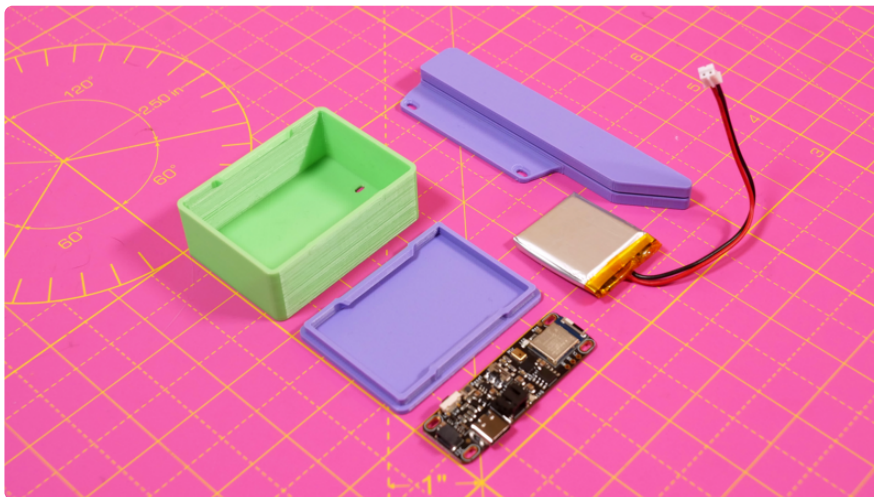
Fold each wing in half. Bring the edge of the wing down to the center fold of the plane.

Now you have a paper airplane!



---

## Assembly



## Solder the Lipo Charge Jumper



The nRF52840 LED Glasses Driver does not support LiPo batteries by default. To use it with a LiPo battery in this project, you will need to solder the jumper on the back of the board to short it.



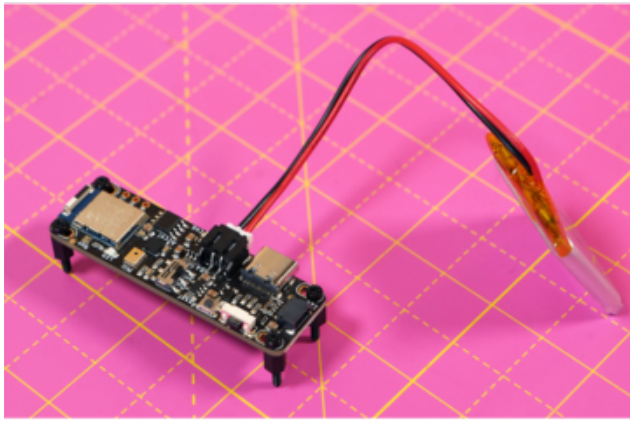
After shorting the jumper, only use 4.2V/3.7V rechargeable batteries in the battery port!

## Mount the Board

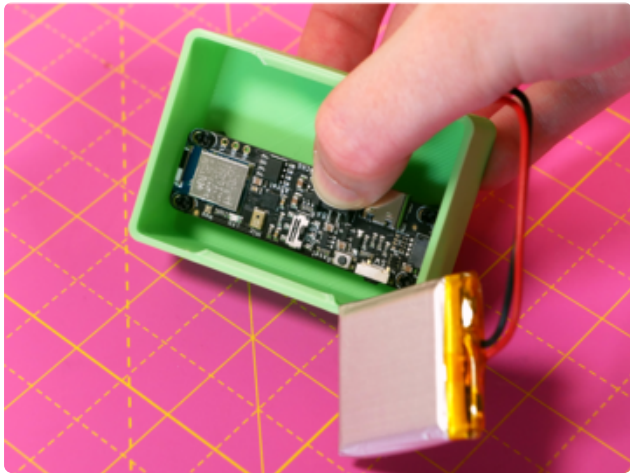


In the board's four mounting holes, attach M2 screws and stand-offs.

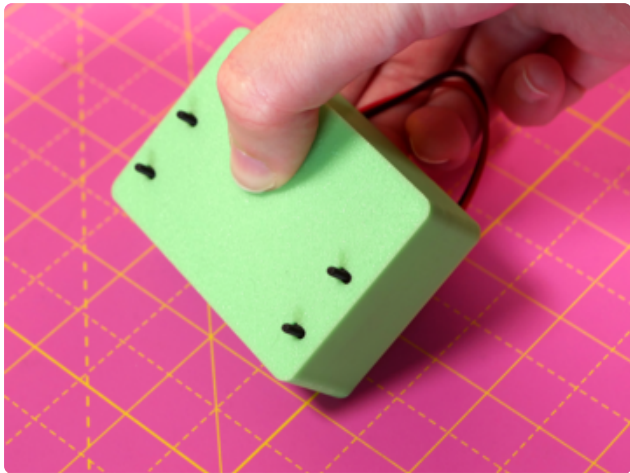




Plug in a LiPo battery to the board.



Slot the board's stand-offs into the case's mounting holes.



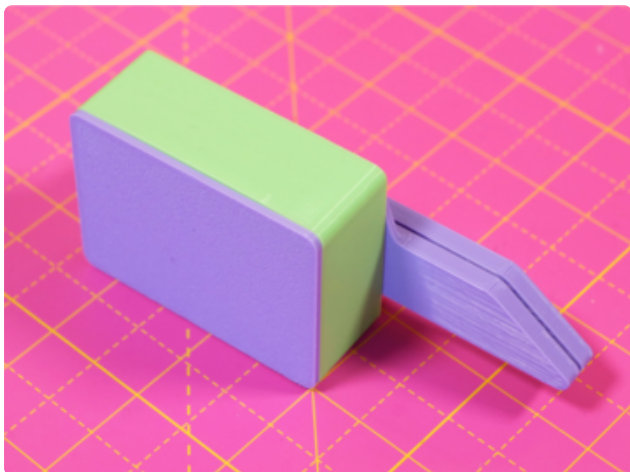




Secure the board's bottom two stand-offs with M2 nuts.

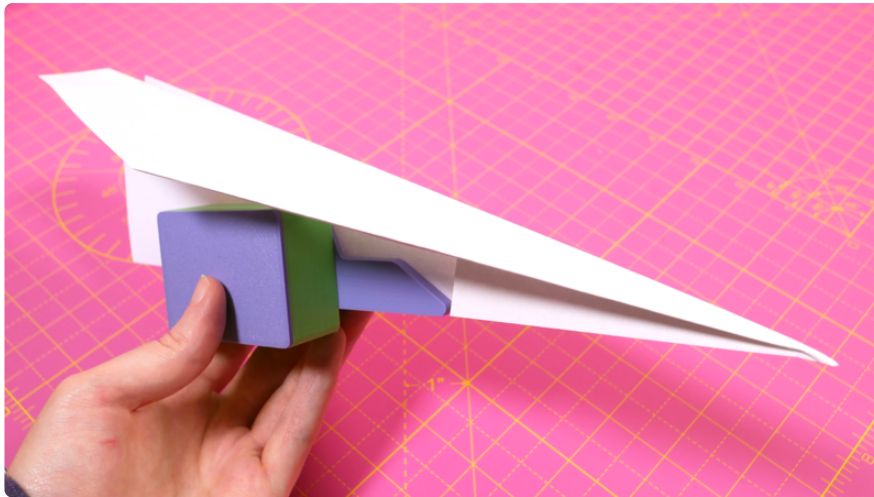


Attach the case to the airplane holder with two M2 nuts.



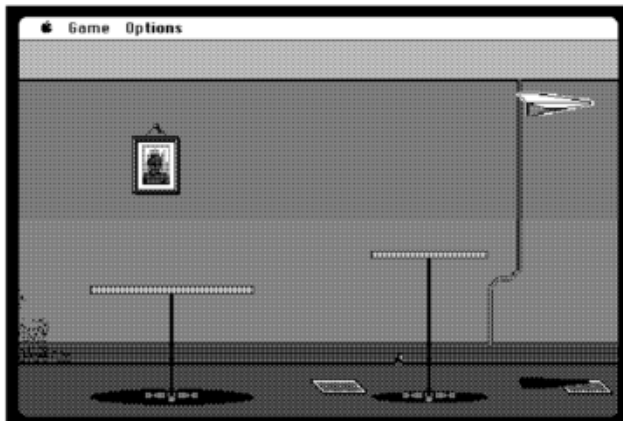
Close the case with the case lid.

Insert your paper airplane into the airplane holder. Now you're ready to fly!

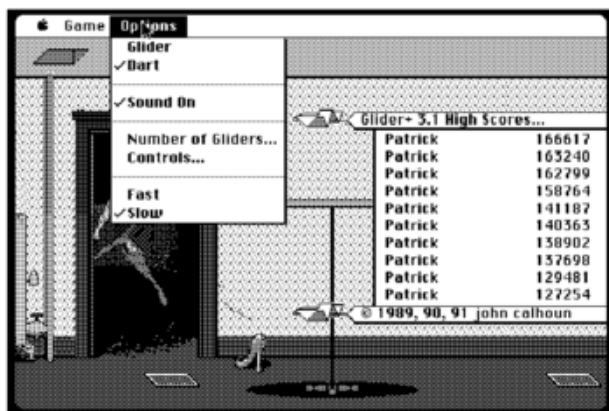


## Usage

### Game Software



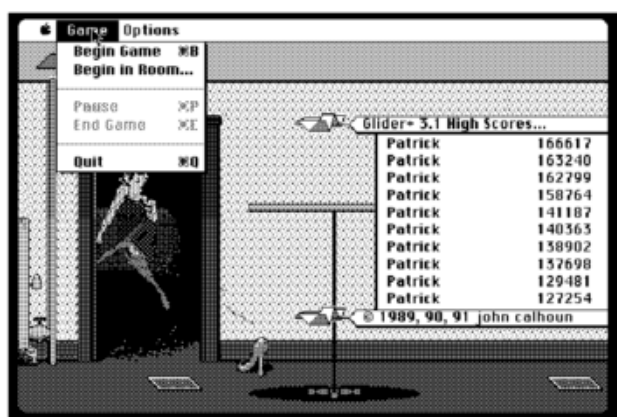
The original Glider game is available on the [Internet Archive \(https://adafru.it/ZDM\)](https://adafru.it/ZDM).



Before beginning a game, check the Game Controls by clicking on Options and Controls...

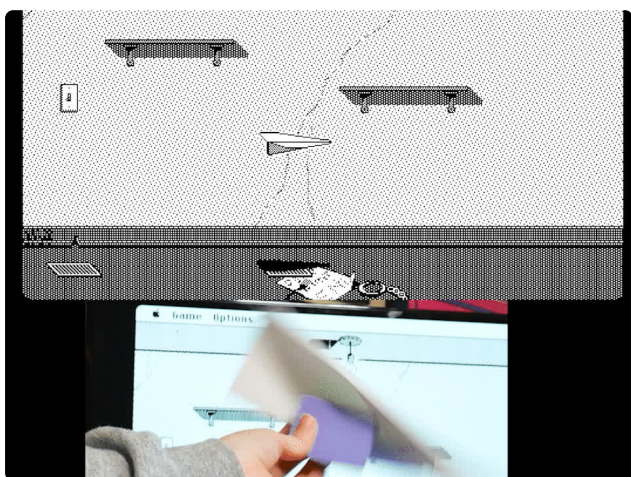


Make sure that l is set for left and r is set for right. This matches the CircuitPython code.

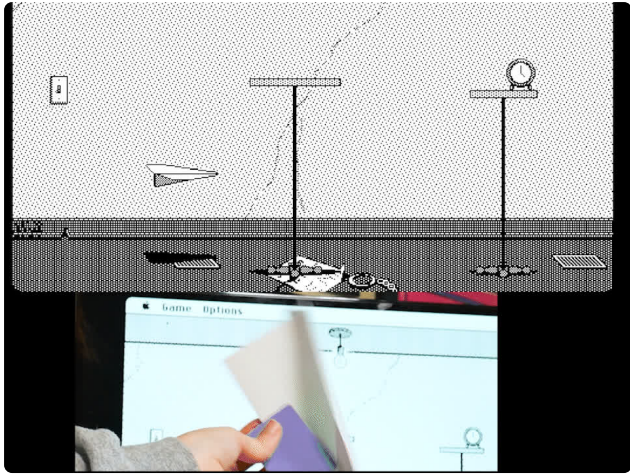


Begin a new game by clicking on Game and Begin Game.

## Game Play



Tilt your paper airplane down to move right.



Tilt your paper airplane up to move left.