



# **ENGR 21:**

# **Computer Engineering Fundamentals**

Lecture 5  
Tuesday, September 16, 2025

# Test #1 of 6 on Thu 9/18

Based on Weeks 1-2, incl. HW 1-2

In-class, 8:30 to 8:55

Paper exam; no computers / electronic devices

Bring a calculator & pen/pencil

---

---

# Accuracy & Precision for the accelerometer on the Circuit Playground Bluefruit

# The on-board accelerometer (first, try interactively!)

Make sure you first run the line

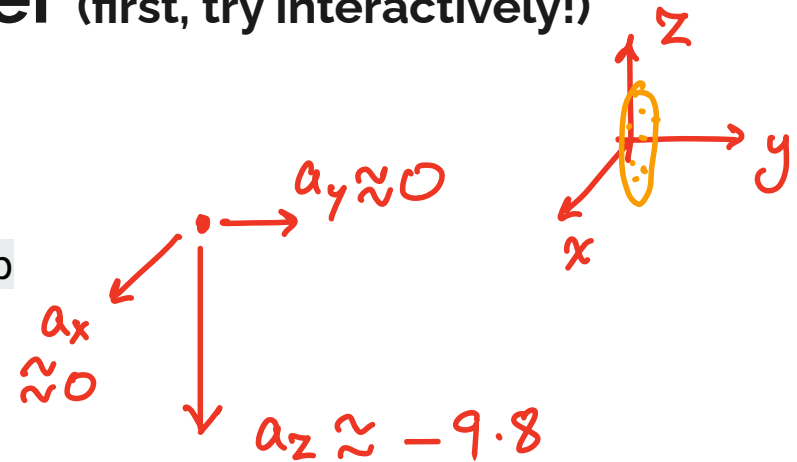
```
from adafruit_circuitplayground import cp
```

Returns three values:

- X, Y and Z

What are the units?

What happens if you tilt the board?



$$|\vec{a}| = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

x	0.0
y	0.0
z	9.78

```
>>> cp.acceleration
acceleration(x=0.0383047, y=-0.114914, z=9.69109)
```

```
>>> cp.acceleration.z    is acceleration.x
9.69109                  equal to zero?
```

# Accelerometer

copy code  
from here  
into code.py !

ENGR 21 Fall 2025

## Resources

- Resources
  - External Guides and Tutorials
  - Instructor's Circuit Playground Guide for E21
  - Links and Code Snippets
    - Lec 1.1, Tue Sep 2
    - Lec 2.1, Tue Sep 9
    - Lec 2.2, Thu Sep 11
    - Lec 3.1, Tue Sep 16

...

# Number of readings

N = 10

# Create a list that will store the readings

readings\_z = [0] \* N

readings = [0] \* N

# Time delay between measurements

delay = 0.1 : 10 times per second

for j in range(N): # Number of seconds to run

    accel\_z = cp.acceleration.z

    accel = magnitude(cp.acceleration.x, cp.acceleration.y, cp.acceleration.z)

    print((accel\_z, accel))

    readings[j] = accel

    readings\_z[j] = accel\_z

    time.sleep(delay) # delay of 1 second

avg\_reading = sum(readings)/len(readings)

avg\_reading\_z = sum(readings\_z)/len(readings\_z)

print("After ", N, " readings, the acceleration is ", avg\_reading, " m/s^2")

print("With standard deviation ", std(readings))

print("After ", N, " readings, the z-acceleration is ", avg\_reading\_z, " m/s^2")

print("With standard deviation ", std(readings\_z))

Collect more !  
Quantify  
Precision : "spread" of data  
+ Accuracy : How close to true value ? ~ 5% error  
How low is standard deviation ?



# Quantifying accuracy & precision for accelerometer

Accuracy

Precision

---

# **Saving Data in Files**

## **On the Circuit Playground Bluefruit**

## Resources

- Resources
  - External Guides and Tutorials
  - Instructor's Circuit Playground Guide for E21
  - Links and Code Snippets
    - Lec 1.1, Tue Sep 2
    - Lec 2.1, Tue Sep 9
    - Lec 2.2, Thu Sep 11
    - Lec 3.1, Tue Sep 16

# Storing Data on the Circuit Playground Bluefruit

It is possible to store data on the board, even when disconnected from your PC and powered with battery

Task: Download boot.py and save it to CIRCUITPY. *don't change*

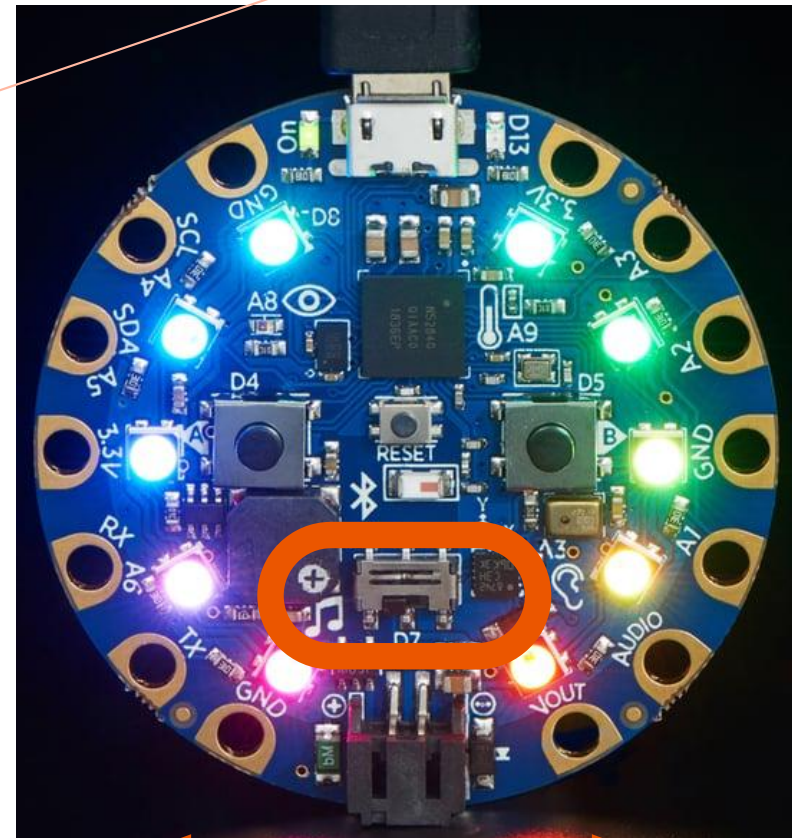
After boot.py is on your board, you will be able to switch between

1. Read-only mode *Computer can save files*
2. Writable mode *Circuit Python can save files.*

**Either your computer can save files to your board, or CircuitPython can save files to your board. not both!!**

To switch between modes:

- Slide switch
- Eject CIRCUITPY from OS
- Press reset button



**Read-only**  
(by CircuitPython)

**Writable**  
(by CircuitPython)  
*saving from PC gives error.*



- Resources
  - External Guides and Tutorials
  - Instructor's Circuit Playground Guide for E21
  - Links and Code Snippets
    - Lec 1.1, Tue Sep 2
    - Lec 2.1, Tue Sep 9
    - Lec 2.2, Thu Sep 11
    - Lec 3.1, Tue Sep 16

# Storing Data on the Circuit Playground Bluefruit

Download Reaction times game from Resources page

after saving to code.py, slide switch → to write mode

## Reaction times game

```
from adafruit_circuitplayground import cp
import time
import random

# Choose the number of data points to collect
N = 5

# Create a list to collect data points
data = [0] * N

# Print some information
print("Welcome to the reaction time game.")
print(f"We will collect {N} samples.")
print("Press button A when an LED lights up.")

# Open file for writing
f = open("/reaction_times.txt", "a")
for j in range(N):
    # Turn off all LEDs
    cp.pixels.fill((0, 0, 0))

    # Wait for a random time between 1 and 5 seconds
    random_delay = random.uniform(1, 5)
    time.sleep(random_delay)
```

Has extra lines compared to HW that save data on board.

a for 'append'

```
# Open the file
f = open("/reaction_times.txt", "a")

# Write to file
f.write(f"{reaction_time:.4f}\n")
f.flush()

# Close the f
f.close()
```

file name precision  
Variable name  
newline character



## Activity: Modify Accelerometer to collect data

Goal: Save 30 seconds of accelerometer data when switched on with battery power, into a text file on board the CIRCUITPY.

Start from the code from [Reaction Time Game](#) and [Accelerometer](#)

(both from Resources Page)