ECE 476/676 - Homework #9

Acceleration, I2C, and NeoPixels - Due Wednesday, November 13th

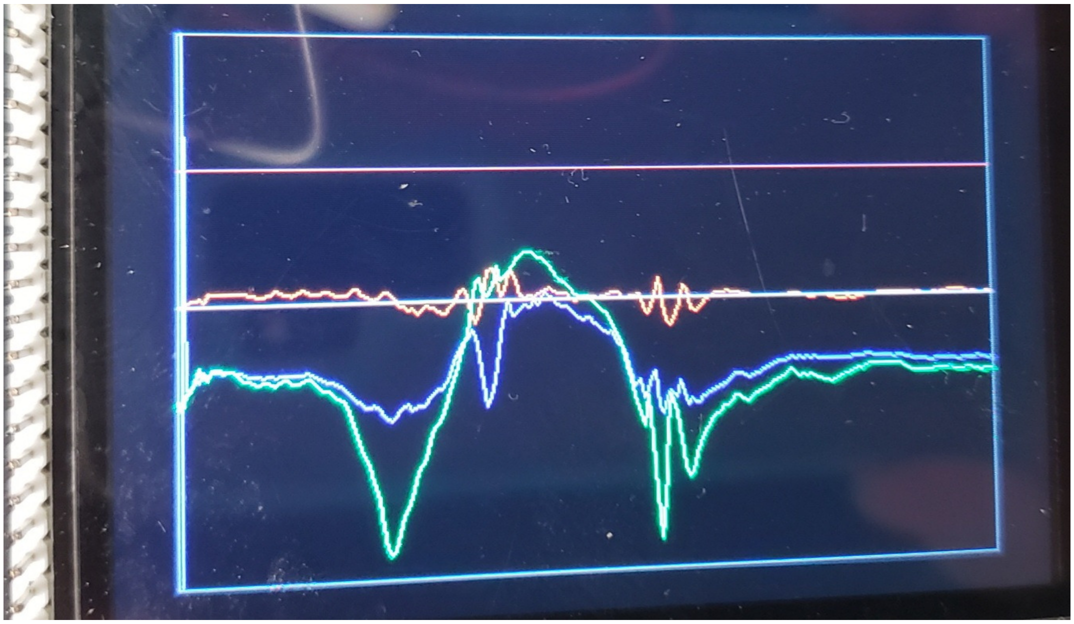
HW9: Acceleration: How High Can You Jump?

1) (30 points): Write a Python program which measures how high can you jump using an accelerometer

- Start a new test by pressing button GP15
- Measure acceleration using a GY-521 accelerometer (or similar sensor)
- Detect then duration that you're experiencing zero g's
- From that time, compute distance you jumped
- Display top three distances on the graphics display

Statring out, read the GY521 accelerometer's readings x", y", and z" for two seconds with a sampling rate of 10ms. Display this on the LCD display.

```
flag = 0
    while(Button14.value() == 1):
        pass
    Beep()
    for i in range(0,npt):
        x = accel_read(0x3b) * RANGE
        y = accel_read(0x3d) * RANGE
        z = accel_read(0x3f) * RANGE
        Data[0][i] = x
        Data[1][i] = y
        Data[2][i] = z
        t[i] = i
        time.sleep(0.01)
    Data[0][0] = -RANGE
    Data[0][1] = +RANGE
    LCD.Clear(Navy)
    LCD.Plot(t,[Data[0],Data[1],Data[2],T0,T1])
```



Acceleration during a jump: Acceleration x", y", z" along with 0g (white) and +1g (red)

Next, compute the net acceleration as

$$a = \sqrt{\ddot{x}^2 + \ddot{y}^2 + \ddot{z}^2}$$

Compute the time that the acceleration is less than 0.5g (green line in figure below)

- Record the time that the acceleration drops below 0.5g (Ton)
- Record the time that the acceleration goes back above 0.5g (Toff)
- Air time is the time difference (Toff - Ton)
- Keep track of the maximum time found in the data set

Once found, compute the distance in cm as

$$d = \tfrac{1}{8}at^2$$

```
Ton = 0
Toff = 0
dT = 0
flag = 0
for i in range(0,npt):
    Accel[i] = (Data[0][i]**2 + Data[1][i]**2 + Data[2][i]**2)**(0.5)
    if(Accel[i] < 0.5):
        if(Ton == 0):
            Ton = i
    if(Accel[i] > 0.5):
        if(Ton > 0):
            if(Toff == 0):
                Toff = i
                dT = max(dT, Toff - Ton)
                Ton = 0
                Toff = 0

cm = 980/8 * (dT*0.01) ** 2
Top4[3] = cm
```

To display the top three distances, store the distance in an array (Top4[i]). Do a bubble sort to show the top three distances

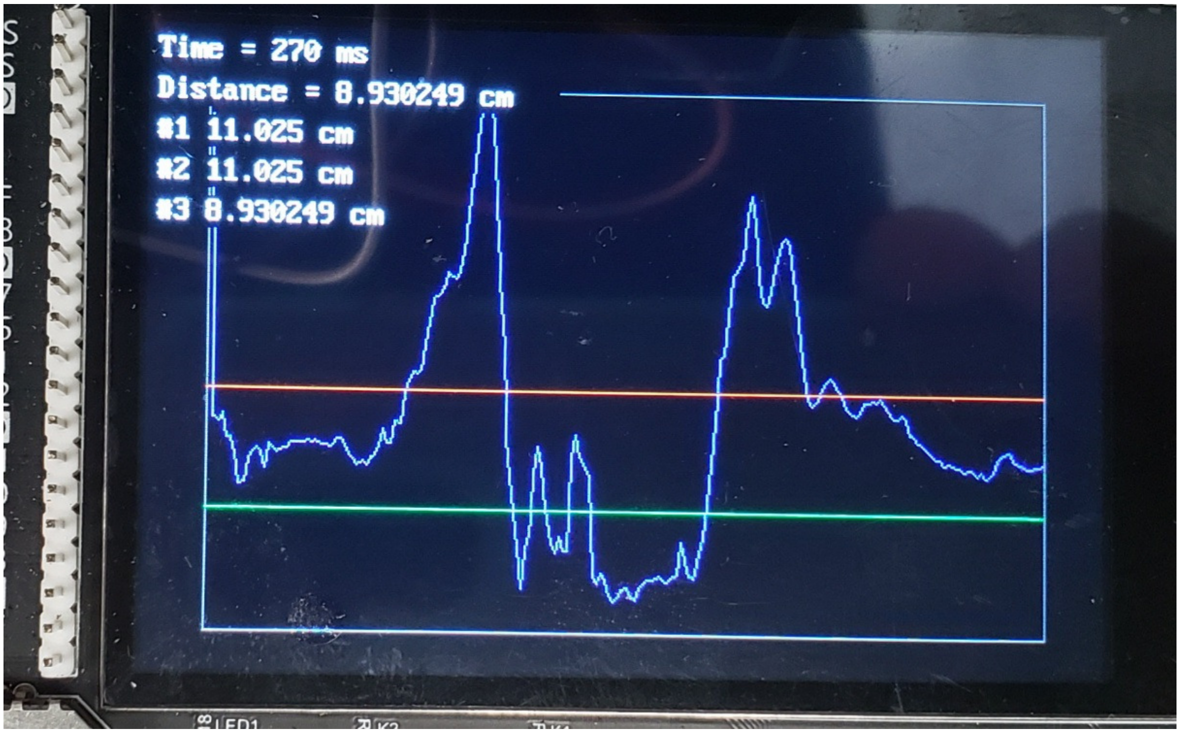
```
for i in range(0,4):
       for j in range(i+1,4):
           if(Top4[j]>Top4[i]):
               Temp = Top4[i]
               Top4[i]=Top4[j]
               Top4[j] = Temp

   while(Button15.value() == 1):
       pass


   LCD.Clear(0);
   LCD.Plot(t,[Accel, T1, T05, T0])
   LCD.Text('Time = ' + str(dT*10) + ' ms   ',5,5,White, Black)
   LCD.Text('Distance = ' + str(cm) + ' cm   ',5, 25, White, Black)
   LCD.Text('#1 ' + str(Top4[0]) + ' cm ', 5, 45, White, Black)
   LCD.Text('#2 ' + str(Top4[1]) + ' cm ', 5, 65, White, Black)
   LCD.Text('#3 ' + str(Top4[2]) + ' cm ', 5, 85, White, Black)
```

The net result is every time you jump,

- Your acceleration over a 2 second interval is displayed
- The time and height of this jump is displayed, and
- The top three jumps is displayed



```
Time = 270 ms
Distance = 8.930249 cm
#1 11.025 cm
#2 11.025 cm
#3 8.930249 cm
```

Net acceleration (blue), 0g (red) and 0.5g (green)

2) (10 points): Add a NeoPixel to your design as a starter tree:

- Press button GP15 to start a jump session
- When pressed, the lights on the NeoPixel light up, one at a time
- When all lights are lit, it's time to jump (data collection starts)

Start with adding a 16-element NeoPixel

```
import time, LCD, neopixel
from machine import Pin, I2C

N = 16
p = machine.Pin(12, Pin.OUT)
np = neopixel.NeoPixel(p, N, bpp=3, timing=1)
```

Change the main loop so that when you press GP15, it starts a count-down timer on the NeoPixel

- 0.5 seconds between lights
- Turn on one red light every 500ms
- When all 16 lights are turned on, switch to green, beep, and start data collection

```
while(1):
    flag = 0
    while(Button14.value() == 1):
        pass
    np.fill([0,0,0])
    np.write()
    for i in range(0,N-1):
        np[i] = (50,0,0)
        np.write()
        time.sleep(0.5)
    np.fill([0,50,0])
    np.write()
    Beep()
```

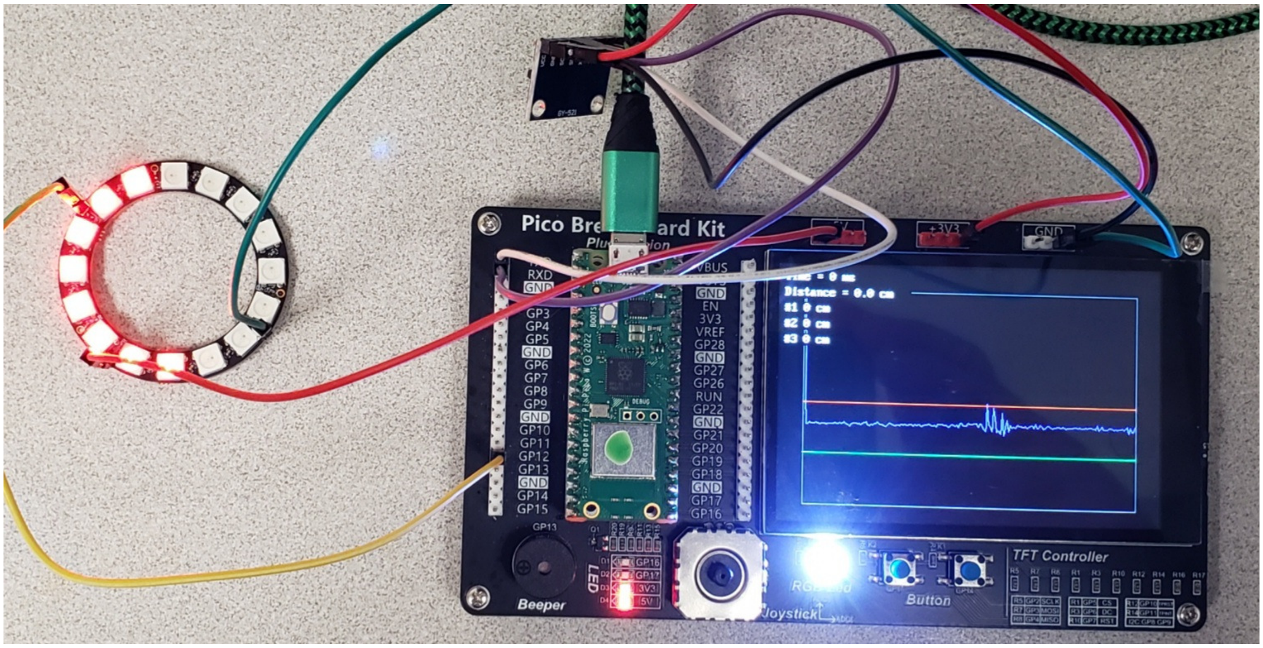
The rest of the code is pretty much the same

3) (10 points): Demo your program

- shows better in the video

Notes:

- It would be better to use timer interrupts to set the sampling rate
- The sampling time (10ms) could be reduced.
- By measuring the time it takes to record the acceleration x", y", z", you could determine the minimum sampling rate this program allows
- Recording and plotting the acceleration isn't necessary - but it's kind of fun to see what the sensor is seeing as well as whether the threshold of 0.5g is reasonable.



```
Time = 0 ms
Distance = 0.0 cm
#1 0 cm
#2 0 cm
#3 0 cm
```

Resulting system:GY521 accelerometer, NeoPixel, and 52pi Pico Breakout Board Kit