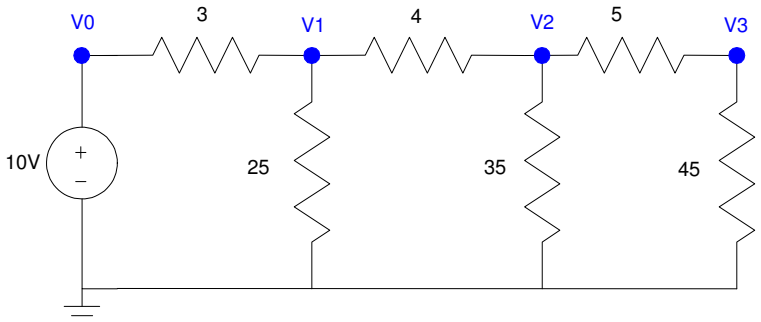
ECE 476/676 - Homework #6

Math, Random, and Matrix Routines - Due Monday, October 13th

Matrix Functions

1) Determine the voltages {V0, V1, V2, V3} using matrix funcitons with Python



a) Write the voltage node equations for this circuit

$$V_0 = 10$$

$$\left(\frac{V_1 - V_0}{3}\right) + \left(\frac{V_1}{25}\right) + \left(\frac{V_1 - V_2}{4}\right) = 0$$

$$\left(\frac{V_2 - V_1}{4}\right) + \left(\frac{V_2}{35}\right) + \left(\frac{V_2 - V_3}{5}\right) = 0$$

$$\left(\frac{V_3 - V_2}{5}\right) + \left(\frac{V_3}{45}\right) = 0$$

b) Express these equations in matrix form

Rewrite & group terms

$$V_0 = 10$$

$$-\left(\frac{1}{3}\right)V_0 + \left(\frac{1}{3} + \frac{1}{25} + \frac{1}{4}\right)V_1 - \left(\frac{1}{4}\right)V_2 = 0$$

$$-\left(\frac{1}{4}\right)V_1 + \left(\frac{1}{4} + \frac{1}{35} + \frac{1}{5}\right)V_2 - \left(\frac{1}{5}\right)V_3 = 0$$

$$-\left(\frac{1}{5}\right)V_2 + \left(\frac{1}{5} + \frac{1}{45}\right)V_3 = 0$$

Place in matrix form

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \left(\frac{-1}{3}\right) & \left(\frac{1}{3}+\frac{1}{25}+\frac{1}{4}\right) & \left(\frac{-1}{4}\right) & 0 \\ 0 & \left(\frac{-1}{4}\right) & \left(\frac{1}{4}+\frac{1}{35}+\frac{1}{5}\right) & \left(\frac{-1}{5}\right) \\ 0 & 0 & \left(\frac{-1}{5}\right) & \left(\frac{1}{5}+\frac{1}{45}\right) \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

c) Solve using matrix operations in Python

Program Window

```
import matrix

B = [[1,0,0,0],
     [-1/3,1/3+1/25+1/4,-1/4,0],
     [0,-1/4,1/4+1/35+1/5,-1/5],
     [0,0,-1/5,1/5+1/45]]
A = [[10],[0],[0],[0]]
print('--- B ---')
matrix.display(B)
print('--- A ---')
matrix.display(A)
Bi = matrix.inv(B)
V = matrix.mult(Bi,A)
print('--- Votlages ---')
matrix.display(V)
```

Shell Windoe

```
--- B ---
     1.000     0.000     0.000     0.000
    -0.333     0.623    -0.250     0.000
     0.000    -0.250     0.479    -0.200
     0.000     0.000    -0.200     0.222

--- A ---
    10.000
     0.000
     0.000
     0.000

--- Votlages ---
    10.000
     8.051
     6.742
     6.067
```

2) Check your answers using CircuitLab (or Matlab)

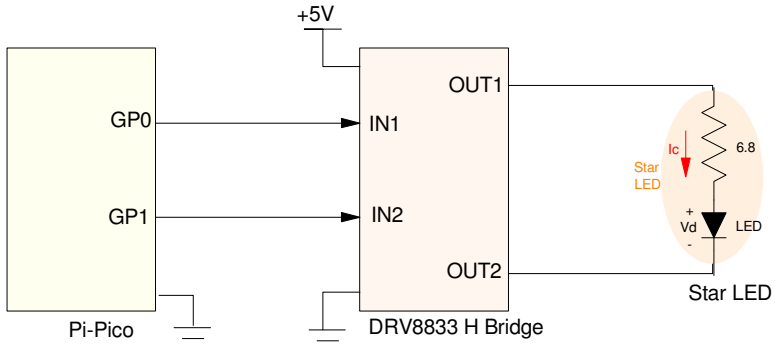
```
>> b1 = [1,0,0,0];
>> b2 = [-1/3, 1/3+1/25+1/4, -1/4, 0];
>> b3 = [0, -1/4, 1/4+1/35+1/5, -1/5];
>> b4 = [0, 0, -1/5, 1/5+1/45];
>> B = [b1;b2;b3;b4]

    1.0000         0         0         0
   -0.3333    0.6233   -0.2500         0
         0   -0.2500    0.4786   -0.2000
         0         0   -0.2000    0.2222

>> A = [10;0;0;0]

    10
     0
     0
     0

>> V = inv(B)*A

   10.0000
    8.0515
    6.7417
    6.0675
```

Candle Flicker (Hardware)

3) Set up a circuit so your Pico can turn a white LED on and off

- H-bridge works (overkill, but works)
- 6144 Transistor works

Verify you can turn the LED on and off with the Pico



Test Code

```
from machine import Pin, PWM, I2C
from time import sleep_ms
import LCD_16x24 as LCD
from gt911 import GT911
from random import random
from math import log

# intialize PWM outputs
Candle = PWM(Pin(0))
Candle.freq(1000)
Candle.duty_u16(0)

# initialize the LCD display
LCD.Init()
Black = LCD.RGB(0,0,0)
White = LCD.RGB(250,250,250)
Grey  = LCD.RGB(100,100,100)

LCD.Clear(Black)

LCD.Box(1,1,478,318,White)
LCD.Text3('HW6: Candle Flicker', 5, 5, White, Black)
LCD.Text3('Problem 3', 5, 55, White, Black)

X = 0

while(1):
    X += 1000
    Y = 100
    Candle.duty_u16(X)
    sleep_ms(Y)
```

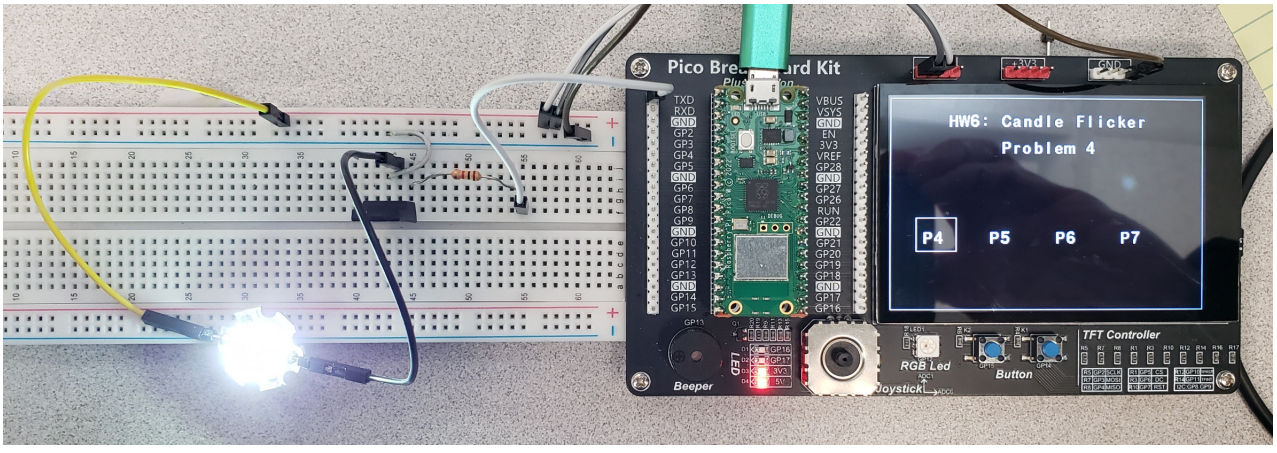
4) Flicker (take 1): Make the LED flicker using PWM

- Let X be the brightness of the LED
  - X has a uniform distribution from 0% to 100%
- Let Y be the samping rate (how often X is updated)
  - Y is 100ms (fixed)

Give you opinion for how much this looks like a candle flickering

- kind of subjective

```
from machine import Pin, PWM, I2C
from time import sleep_ms
import LCD_16x24 as LCD
from gt911 import GT911
from random import random
from math import log

# intialize PWM outputs
Candle = PWM(Pin(0))
Candle.freq(1000)
Candle.duty_u16(0)

# initialize the LCD display
LCD.Init()
Black = LCD.RGB(0,0,0)
White = LCD.RGB(250,250,250)
Grey  = LCD.RGB(100,100,100)

LCD.Clear(Black)

LCD.Box(1,1,478,318,White)
LCD.Text3('HW6: Candle Flicker', 5, 5, White, Black)
LCD.Text3('Problem 4', 5, 55, White, Black)

while(1):
    X = int(random() * 65535)
    Y = 100
    Candle.duty_u16(X)
    sleep_ms(Y)
```



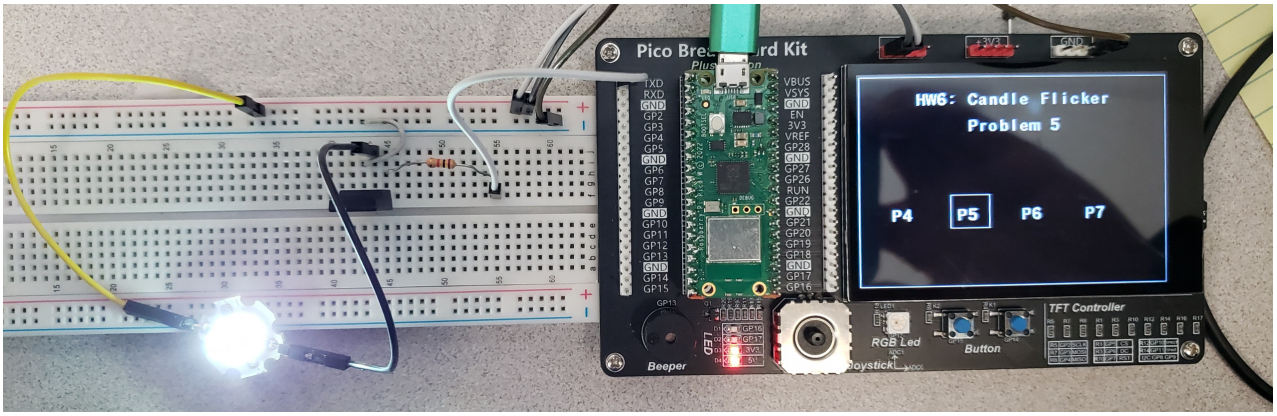
5) Flicker (take 2): Repeat, only

- Let X be the brightness of the LED
  - X has a uniform distribution from 0% to 100%
- Let Y be the samping rate (how often X is updated)
  - Y has an exponential disribution with a mean of 100ms

Give you opinion for how much this looks like a candle flickering

- kind of subjective

```
LCD.Box(1,1,478,318,White)
LCD.Text3('HW6: Candle Flicker', 5, 5, White, Black)
LCD.Text3('Problem 5', 5, 55, White, Black)

while(1):
    X = int(random() * 65535)
    Y = int(-log(random() * 100))
    Candle.duty_u16(X)
    sleep_ms(Y)
```



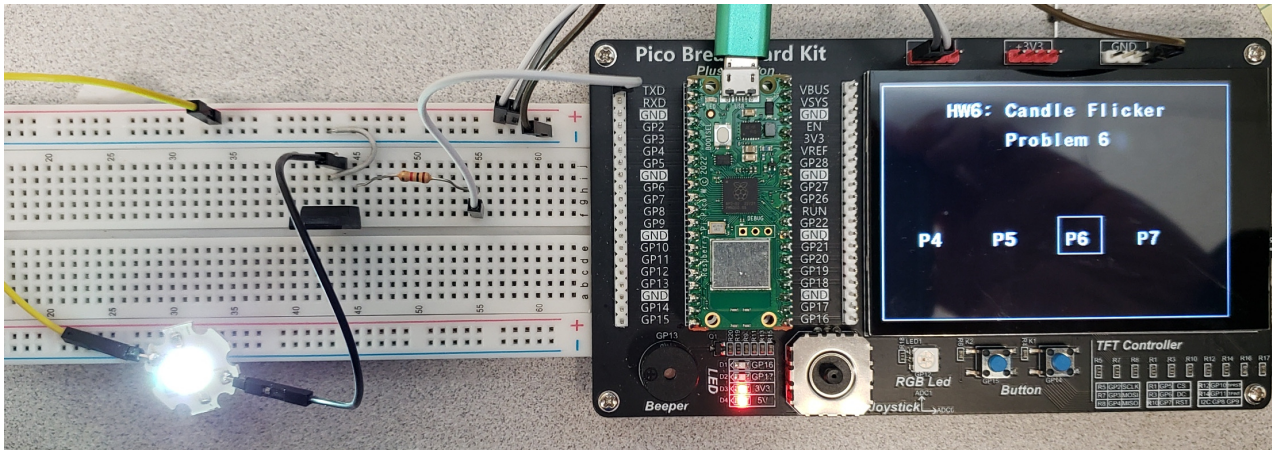
6) Flicker (take 3): Repeat, only

- Let X be the brightness of the LED
  - X has a normal distribution with a mean of 50% and standard deviation of 20%
- Let Y be the samping rate (how often X is updated)
  - Y is 100ms (fixed)

Give you opinion for how much this looks like a candle flickering

- kind of subjective

```
LCD.Box(1,1,478,318,White)
LCD.Text3('HW6: Candle Flicker', 5, 5, White, Black)
LCD.Text3('Problem 6', 5, 55, White, Black)

while(1):
     X = 32000 + int(13000*randn())
     Y = 100
    Candle.duty_u16(X)
    sleep_ms(Y)
```



Demonstration

7) Demonstrate what you think is the best flickering candle

Bonus! Use the touch screen to select the distribution of the flicker:

- Uniform (problem 4)
- Exponential (problem 5)
- Normal (problem 6)

By using the touch-screen, you can change the flicker distribution on the fly.

```
from machine import Pin, PWM, I2C
from time import sleep_ms
import LCD_16x24 as LCD
from gt911 import GT911
from random import random
from math import log


def randn():
    x = -6;
    for i in range(0,12):
        x += random();
    return(x)

# intialize PWM outputs
Candle = PWM(Pin(0))
Candle.freq(1000)
Candle.duty_u16(0)

# Touch Screen Initialization

rst_pin = Pin(10, Pin.OUT)
irq_pin = Pin(11)
sda_pin = Pin(8)
scl_pin = Pin(9)

touch = GT911(
    I2C(0, scl=scl_pin, sda=sda_pin, freq=100_000),
    rst_pin,
    irq_pin
)

touch.init(touch_points=1, refresh_rate=50)

# initialize the LCD display
LCD.Init()
Black = LCD.RGB(0,0,0)
White = LCD.RGB(250,250,250)
Grey  = LCD.RGB(100,100,100)

LCD.Clear(Black)

LCD.Box(1,1,478,318,White)
LCD.Text3('HW6: Candle Flicker', 5, 5, White, Black)
LCD.Text3('P4', 50, 200, White, Black)
LCD.Text3('P5', 150, 200, White, Black)
LCD.Text3('P6', 250, 200, White, Black)
LCD.Text3('P7', 350, 200, White, Black)


mode = 0
```

```
while(1):
    num_points, points_data = touch.read_points()
    if(num_points > 0):
        mode = (points_data[0][0]) // 100
        for i in range(0,4):
            LCD.Box(40+100*i,180,100+100*i,230,(mode==i)*White)

    if(mode == 0):
        X = int(random() * 65535)
        Y = 100
    elif(mode == 1):
        X = int(random() * 65535)
        Y = int(-log(random() * 100))
    elif(mode == 2):
        X = 32000 + int(13000*randn())
        Y = 100
    else:
        X = 32000 + int(13000*randn())
        Y = int(-log(random() * 100))
    Candle.duty_u16(X)
    sleep_ms(Y)
```