ECE 476/676 - Test #1: Name ____________________

1) **Hardware: Binary Output** A 100 Watt LED requires the following

- Vf = 34V
- Id = 3000mA
- 9000 Lumens @ 3A

Design a circuit so that a Pi-Pico can turn on and off this LED with one of its binary outputs at 3 Amps. Note that the output of a Pi-Pico is

- Von = 3.3V, Iout < 12mA

If you need to make assumptions about the hardware you are using, state the assumptions you're making

Option #1: Use a BJT transistor capable of 3A with a gain of 300

Assume a 50V power supply (anything larger than 34.2 Volts works)

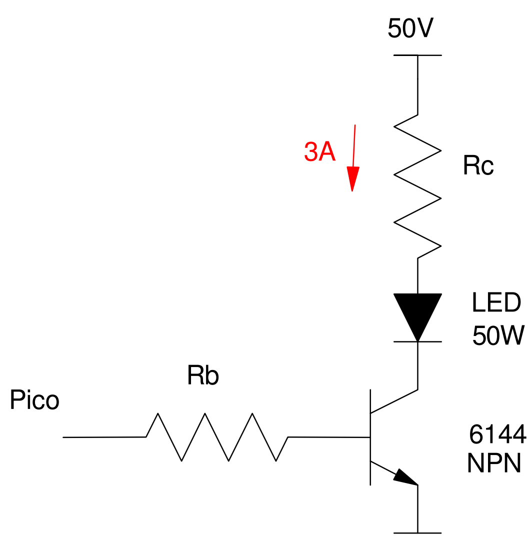
$$R_c = \left(\frac{50V-34V-0.2V}{3A}\right) = 5.27\Omega$$

To saturate the transistor

$$I_b > \frac{I_c}{\beta} = \frac{3A}{300} = 10mA$$

Let Ib = 12mA (max Pico can output)

$$R_b = \left(\frac{3.3V-0.7V}{12mA}\right) = 216\Omega$$



Option 2: Use a MOSFET capable of 3A and 50V.

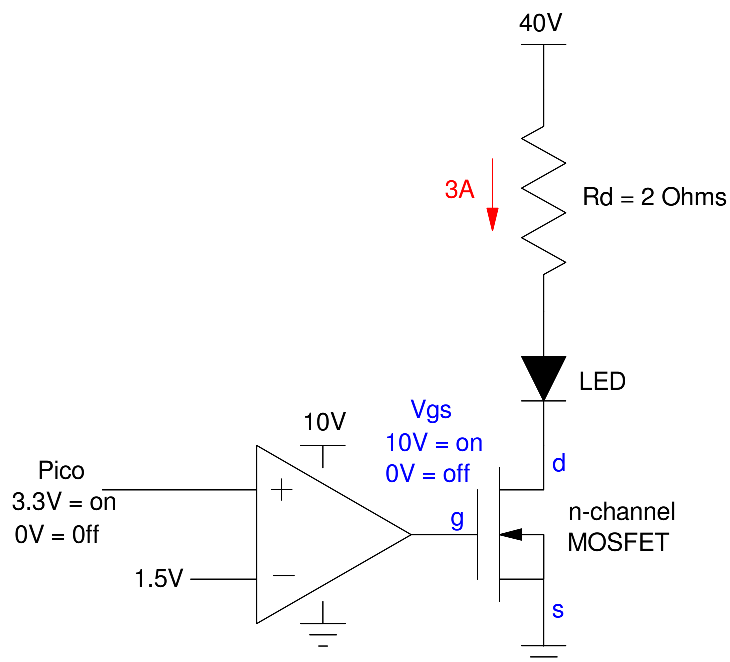
Example: FDP5800 (not necessary to find an actual MOSFET)

- Rds = 0.006 Ohm @ Vgs = 10V, Ids = 80A
- Vgs(th) = 2.5V (max)
- $2.90 each

$$R_d = \left(\frac{50V-24V}{2A}\right) = 2\Omega$$

(actually 1.994 Ohms due to Rds = 0.006 Ohms)

Use a comparitor to convert the Pico output from (0V, 3.3V) to (0V, 10V)



2) Hardware: Analog Inputs Design a circuit which converts x (a -5V to +10V analog signal) to y (a 0V to +3.3V analog signal)

- -5V in produces 0V out
- +10V in produces +3.3V out
- Proportional inbeteen

$$y = \left(\frac{3.3V}{15V}\right)x + \left(\frac{3.3 \cdot 5}{15}\right) = 0.22x + 1.1 \qquad \textit{correct equation}$$

$$y = 0.22x + 0.11 \qquad \textit{given equation}$$

Option 1: Use three resistors as a weighted average

Rewrite this as

$$y = ax + b(3.3V) + c(0V)$$

such that

$$a + b + c = 1$$

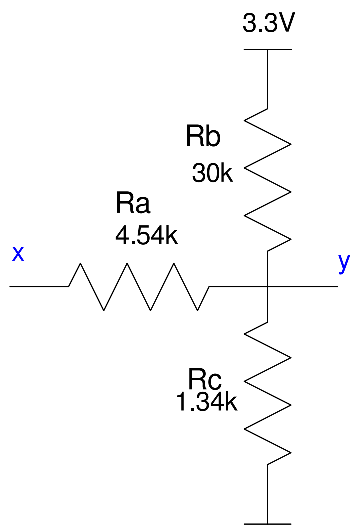
$$y = 0.22x + 0.0333(3.3V) + 0.7467(0V)$$

Assume a base resistance of 1k Ohm

$$R_a = \frac{1k}{0.22} = 4.54k$$

$$R_b = \frac{1k}{0.0333} = 30k$$

$$R_c = \frac{1k}{0.7467} = 1.34k$$



Using the correct equations

$$y = 0.22x + 1.1$$

Rewrite as

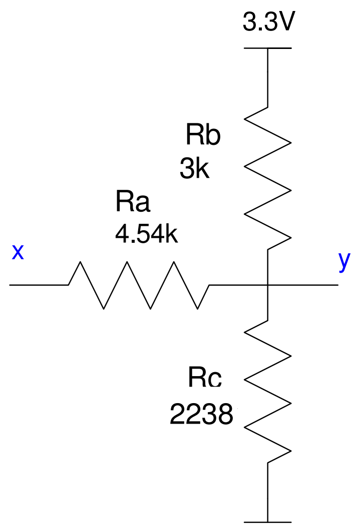
$$y = 0.22x + 0.333(3.3V) + 0.4467(0V)$$

Let R0 = 1k

$$R_a = \frac{1k}{0.22} = 4545$$

$$R_b = \frac{1k}{0.3333} = 3k$$

$$R_c = \frac{1k}{0.4467} = 2238$$



Option 2: Use an instrumentation amplifier

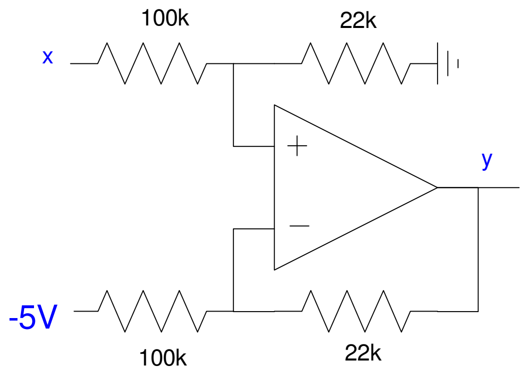
$$y = \left(\frac{R_1}{R_2}\right)(V_a - V_b)$$

Rewrite the output in this form

$$y = 0.22x + 1.1$$

$$y = 0.22(x + 5)$$

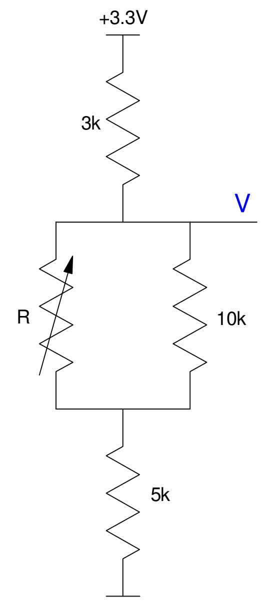
$$y = 0.22(x - (-5))$$



Other solutions exist

3) Python Subroutines: Write a Python subroutine which

- Is passed the temperature in degrees C, and
- Returns the voltage output for the following circuit.



Assume the thermistor has the temperature - resistance relationship of

$$R = 3000 \cdot \exp\left(\frac{4000}{T+273} - \frac{4000}{298}\right)\Omega$$

and a voltage divider with

$$R_a = \left(\frac{1}{R} + \frac{1}{10k}\right)^{-1}$$

$$R_b = R_a + 5000$$

$$V = \left(\frac{R_b}{R_b+3k}\right) \cdot 3.3V$$

```
# Start of subroutine

def Voltage(T):

    e = 2.7182818

    R = 3000 * e **( 4000 / (T+273) - 4000/298 )

    Ra = 1 / ( 1/R + 1/10000 )

    Rb = Ra + 5000

    V = Rb / (Rb + 3000) * 3.3

    return(V)
```

4) Python Programming Assume the hardware is set up so that a Pi-Pico can drive a 100W LED:

- GP16 = 1 (3.3V): LED is on (9000 Lumens)
- GP16 = 0 (0V): LED is off (0 Lumens)

Write a Python program adjusts the light's brightness based upon which button is pressed:

- GP0: Light Off
- GP1: Light On (1%)
- GP2: Light On (10%)
- GP3: Light On (100%)

Use whatever method you like to vary the light's brightness

```
from machine import Pin, PWM
from time import sleep

B0 = Pin(0, Pin.IN, Pin.PULL_UP)
B1 = Pin(1, Pin.IN, Pin.PULL_UP)
B2 = Pin(2, Pin.IN, Pin.PULL_UP)
B3 = Pin(3, Pin.IN, Pin.PULL_UP)

Aout = Pin(16, Pin.OUT)
Aout = PWM(Pin(16))
Aout.freq(1000)
Aout.duty_u16(0x0000)


while(1):
    if(B0.value() == 0):
        Aout.duty_u16(0)

    elif(B1.value() == 0):
        Aout.duty_u16(655)

    elif(B2.value() == 0):
        Aout.duty_u16(6555)

    elif(B3.value() == 0):
        Aout.duty_u16(65535)

    sleep(0.1)
```

Sidelight: The requirements didn't specify what happens if no button is pressed.

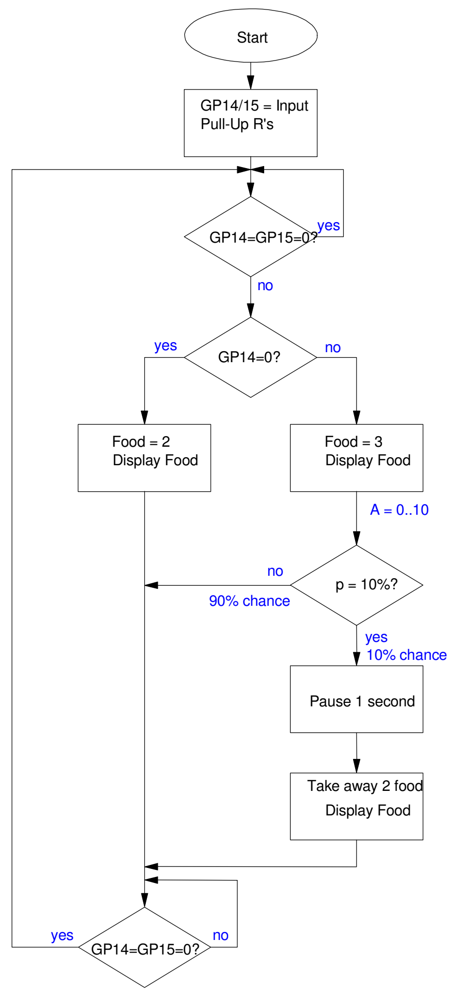
- Some solutions turned off the light
- Some set the brightness to 50%
- Some set the brightness to 100%
- Some left it alone (last button pressed)

5) Python Programming: A Pico is to control a mechanism which gives out food.

- When GP14 is pressed, two pieces of food are given out (Food = 2)
- When GP15 is pressed, three pieces of food are given out (Food = 3).
- However, 10% of the time the mechanism will then pause for one second, then take away two pieces (resulting in Food = 1)
- The program then waits for the buttons to be release and it starts over

Write the corresponding Python program

```
from machine import Pin
from time import sleep
from random import randrange

B14 = Pin(14, Pin.IN, Pin.PULL_UP)
B15 = Pin(15, Pin.IN, Pin.PULL_UP)

while(1):

    while( (B14.value() == 0) and (B15.value() == 0) ):
        pass

    if(B14.value() == 0):
        Food = 2
        print('Food = ',Food)
    else():
        Food = 3
        print('Food = ',Food)
        a = randrange(10)
        if(a == 0):
            sleep(1)
            Food = 1
            print('Food = ',Food)

    while( not( (B14.value() == 0) and (B15.value() == 0) ):
        pass
```



note: You can use DeMorgan's law and also write the last statement as

```
    while( (B14.value() == 1) or (B15.value() == 1) ):
        pass
```

Generally Useful Python Routines

Binary Input (Button Pressed)

```
from machine import Pin

Button = Pin(15, Pin.IN, Pin.PULL_UP)
x = Button.value()
```

Binary Output (Blinking Light)

```
from machine import Pin

LED = Pin(16, Pin.OUT)
LED.toggle()
LED.value(1)
LED.value(0)
```

Analog Input (A2D Read)

```
from machine import ADC

a2d0 = ADC(0)
x = a2d0.real_u16()
```

Analog Output (PWM Output)

```
from machine import Pin, PWM

Aout = Pin(16, Pin.OUT)
Aout = PWM(Pin(16))
Aout.freq(1000)

# 0% duty cycle
Aout.duty_u16(0x0000)

# 100% duty cycle
Aout.duty_u16(0xFFFF)

# 50us pulse
Aout.duty_ns(50_000)
```

Measure a pulse width in micro-seconds

```
from machine import Pin, time_pulse_us

X = Pin(19, Pin.IN, Pin.PULL_UP)
low = time_pulse_us(19, 0, 500_000)
high = time_pulse_us(19, 1, 500_000)
```

Pause 1.23 seconds

```
from time import sleep

sleep(1.23)
```

For Loops

```
for i in range(0,6):
    d1 = i
    for j in range(0,4):
        d2 = j
        y = d1 + d2
```

While Loops

```
t = 0
while(t < 5):
    t = t + 0.01
    print(t)
```

If - else if - else statements

```
if(x < 10):
    a = 1
elif(x < 20):
    a = 2
else:
    a = 3
```

Random Numbers

```
from random import randrange

x = randrange(10)
# x = 0 to 9
```

Measure time since reset

```
from time import ticks_us

x0 = ticks_us()
```