

ECE 476/676 - Test #1: Name _____

Binary Outputs

Design a circuit to allow a Pi-Pico to turn on and off a red LED and a solenoid:

LED (GP0):

- When ON, the red LED draws 5mA and has a 1.9V drop across it

Solenoid (GP1):

- When ON, the solenoid draws 1A and has a 12V drop across it

For the LED, all you need is a resistor to limit the current to 5mA

$$R_r = \left(\frac{3.3V - 1.9V}{5mA} \right) = 220\Omega$$

For the solenoid, a transistor is needed. Assuming a 6144 NPN transistor with

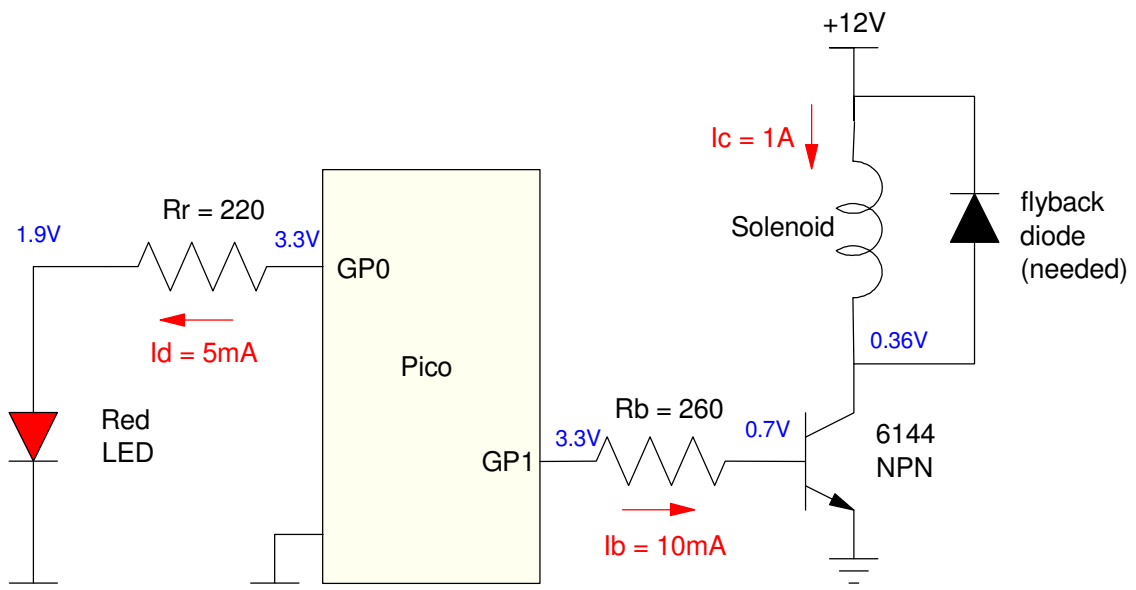
- $\beta = 200$
- $V_{ce(sat)} = 0.36V$

$$\beta I_b > I_c$$

$$I_b > \left(\frac{1A}{200} \right) = 5mA$$

Let $I_b = 10mA$

$$R_b = \left(\frac{3.3V - 0.7V}{200} \right) = 260\Omega$$



Timing & Combination Lock

Write a python program for a combination lock based upon timing. Assume is tied to ground with a push button:

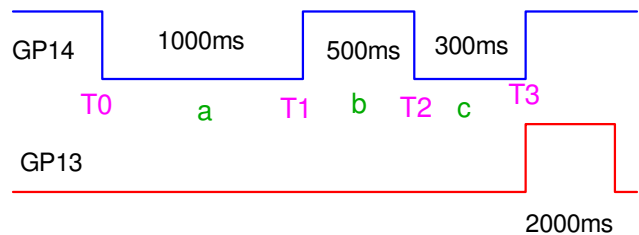
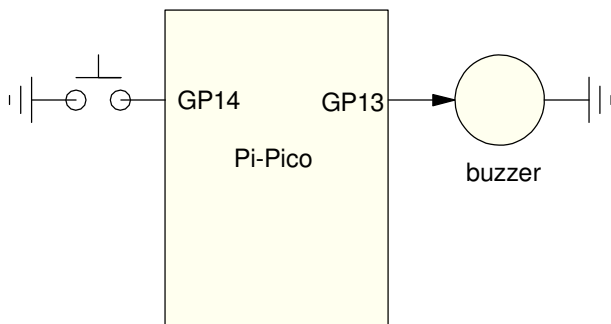
- When the button is pressed, GP14 = 0V
- When released, GP14 = 3.3V (internal pull up resistor)

To open the lock GP14 is

- Held down for 1000ms, +/- 100ms
- Released for 500ms, +/- 100ms, and
- Held down again for 300ms. +/- 100ms

If this sequence is detected, pin GP13 (buzzer) is turned on for 2000ms and then turned off.

Write the corresponding python code.



```
from machine import Pin
from time import ticks_ms, sleep_ms

Button = Pin(14, Pin.IN, Pin.PULL_UP)
Buzzer = Pin(13, Pin.OUT)

T0 = T1 = T2 = T3 = 0
while(1):
    T0 = T2
    T1 = T3
    while(Button.value() == 1):
        pass
    T2 = ticks_ms()
    while(button.value() == 0):
        pass
    T3 = ticks_ms()
    a = abs(T1 - T0)
    b = abs(T2 - T1)
    c = abs(T3 - T2)
    if( abs( a-1000) < 100 ):
        if( abs(b-500) < 100 ):
            if( abs(c-300) < 100 ):
                Buzzer.value(1)
                sleep_ms(2000)
                Buzzer.value(0)
```

Analog Inputs: Thermometer

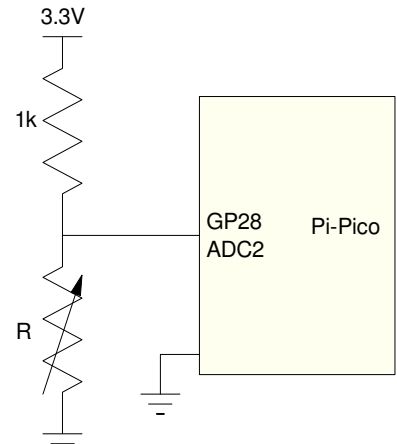
Assume your Pi-Pico is connected to a thermistor along with a voltage divider

$$R = 2000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right) \Omega$$

If the A/D is left justified:

- 0V is read as 0x0000
- 3.3V is read as 0xFFFF = 65,536

determine the following when the A/D reads 0x1234



A/D Reading	Voltage	Resistance	Temperature (C)
0x1234 (4660 decimal)	234.7 mV	76.55 Ohms	123.80 C

The voltage is

$$V = \left(\frac{4660}{65,535}\right) 3.3V = 0.2347V$$

The resistance is

$$V = 0.2347V = \left(\frac{R}{R+1000}\right) 3.3V$$

$$R = \left(\frac{0.2347V}{3.3V-0.2347V}\right) 1000\Omega$$

$$R = 76.5503\Omega$$

The temperature is

$$R = 76.5503\Omega = 2000 \cdot \exp\left(\frac{3905}{T+273} - \frac{3905}{298}\right) \Omega$$

$$T = 123.80^{\circ}C$$

Analog Outputs: LED Flashlight

Assume

- A push button is connected to GP14
 - 0V = button press
 - 3.3V = not pressed
- An LED is connected to pin GP16

The LED flashlight should operate as follows:

- The program loops every 100ms
- If the button is pressed (GP14=0), the LED turns on at 100% brightness (100% duty cycle)
- If the button is not pressed, the LED gets dimmer by 1%
- Once the duty cycle goes below 10%, the LED turns off

Write the corresponding python code.

```
from machine import Pin, PWM
from time import ticks_ms, sleep_ms

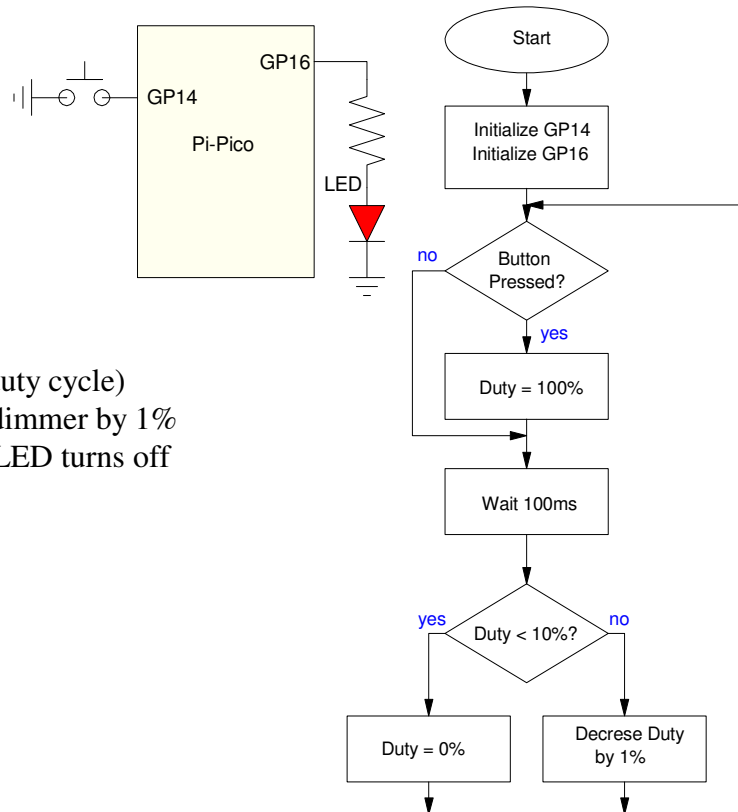
Button = Pin(14, Pin.IN, Pin.PULL_UP)
LED = PWM(Pin(16))
LED.freq(1000)
Duty = 0

while(1):
    if(Button.value() == 0):
        Duty = 65535

    sleep_ms(100)

    if(Duty < 6553):
        Duty = 0
    else():
        Duty -= 655

    LED.duty_u16(Duty)
```



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.read_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 random

x = random()
# x = 0.0000 to 0.9999
```

Measure time since reset

```
from time import ticks_us

x0 = ticks_us()
```

Logic

```
&    logical and
|    logical or
^    logical xor
>>  shift right
<<  shift left
```