

ECE 476/676 - Homework #4

Timing, Analog I/O - Due Monday, September 22nd

Variable Brightness LED

1) Hardware: Connect your Pico to the 1W white LED in your lab kit so that your Pico can turn the LED on and off.

Write a test program to verify the Pico can turn the LED on and off

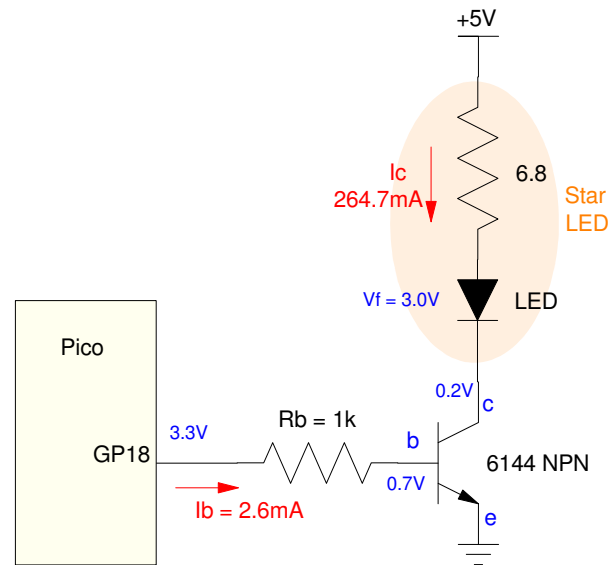
Same hardware as homework #3

Test Program

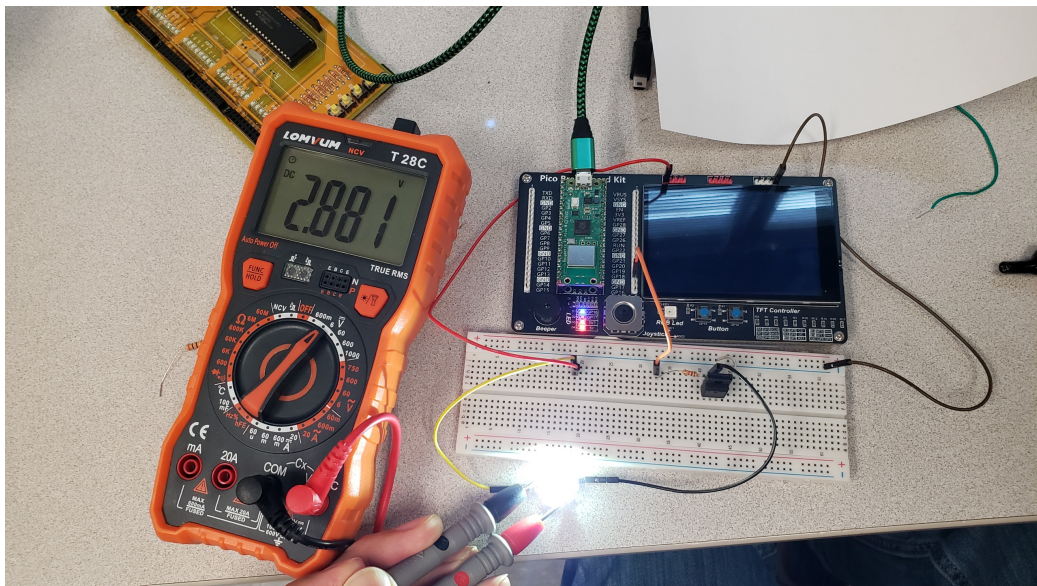
```
from machine import Pin
from time import sleep

LED = Pin(18, Pin.OUT)

while(1):
    LED.toggle()
    sleep(0.5)
```



The LED turns on and off (really bright)



PWM With Push-Button Control

2) Write a Python program which allows you to adjust the brightness of the LED from 0% to 100% using PWM and the push buttons.

- GP15: Increase the brightness
- GP14: Decrease the brightness
- PWM frequency = 1kHz (somewhat arbitrary)

Code:

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

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

def Analog_Out(Pct):
    if(Pct < 0):
        Pct = 0
    if(Pct > 100):
        Pct = 100
    PW = int( (65535/100)*Pct )
    LED.duty_u16(PW)

LED = Pin(17, Pin.OUT)
LED = PWM(Pin(17))
LED.freq(1000)

Pct = 0
dt = 0.1

while(1):
    if(B15.value() == 0):
        Pct += 1
    if(B14.value() == 0):
        Pct -= 1

    Analog_Out(Pct)

    print(Pct)
    sleep(dt)
```

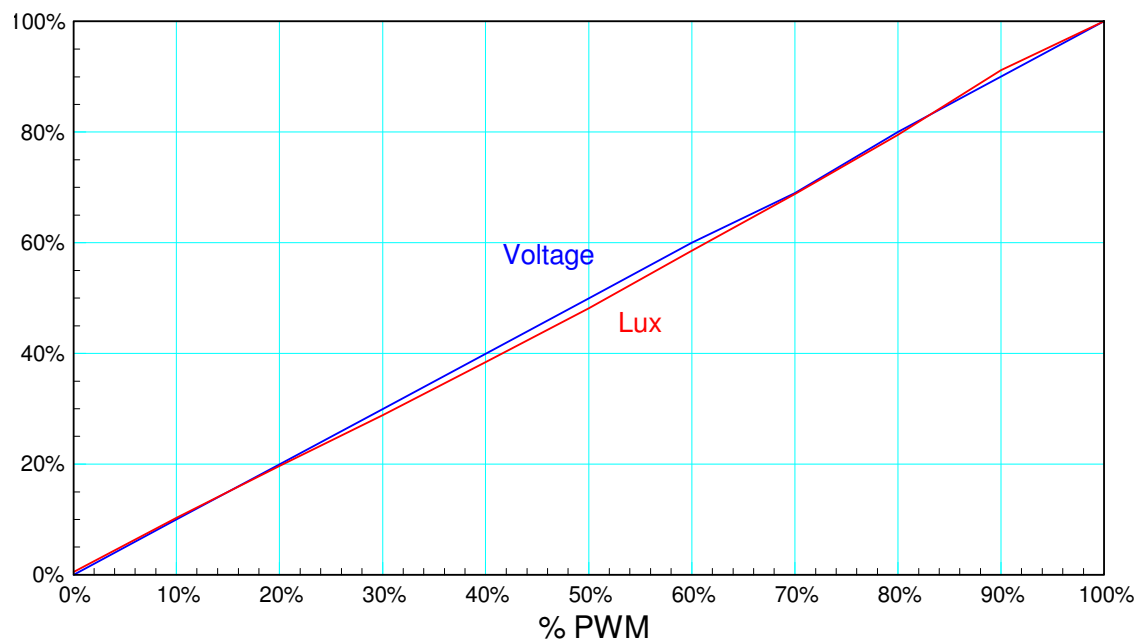
3) Run your program and verify that the brightness varies from 0% to 100%

- Measure the voltage on GP18
- The DC voltage on GP18 should vary from 0V (0%) to 3.3V (100%) as the duty cycle varies.

Note:

- % duty cycle read from the shell window
- Voltage read with a multi-meter
- lux read with a cell-phone app
- Both voltage and lux appear to be proportional to duty cycle (as expected)

0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
0V	0.313V	0.627V	0.941V	1.255V	1.570V	1.885V	2.168V	2.514V	2.828V	3.142V
25 lux	538 lux	1024 lux	1506 lux	2008 lux	2513 lux	3055 lux	3590 lux	4150 lux	4760 lux	5220 lux



PWM with Analog Inputs

4) Write a Python program which allows you to turn on and off the LED using the push buttons

- GP15 = on
- GP14 = off

The brightness is adjustable with the analog input

- AN1: Sets the brightness from 0% to 100%

Code:

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

def Analog_Out(Pct):
    if(Pct < 0):
        Pct = 0
    if(Pct > 100):
        Pct = 100
    PW = int( (65535/100)*Pct )
    LED.duty_u16(PW)

a2d1 = ADC(1)

LED = Pin(17, Pin.OUT)
LED = PWM(Pin(17))
LED.freq(1000)

center = a2d1.read_u16()
Pct = 0
dt = 0.1

while(1):
    a1 = a2d1.read_u16()
    V1 = k * (a1 - center)

    Pct += V1 * dt * 10

    Analog_Out(Pct)

    print(V1, Pct)
    sleep(dt)
```

5) Run your program and verify that the brightness varies from 0% to 100%

Collect data to show

- Light turns on and off with the push buttons
- Brightness is adjustable with the analog input

Pushing up the joystick up increases the brightness

- The further you push, the faster it changes

Pushing up the joystick down decreases the brightness

- The further you push, the faster it changes

Same PWM code and same results as problem #4

Demostration

6) Demonstrate either working program

- In-person
- With a video

