

ECE 476/676 - Final Exam: Name _____

1) Python Programming - The following flow chart counts the number of times button on GP15 is pressed:

- N = the number of button presses
- When N matches your birth month (1..12), the LED on GP16 turn on
- When N matches your birth day (1..31), the LED on GP17 turn on

Write the corresponding Python code

```
from machine import Pin

Button = Pin(15, Pin.IN, Pin.PULL_UP)
LED1 = Pin(16, Pin.OUT)
LED2 = Pin(17, Pin.OUT)

N = 0

while(1):

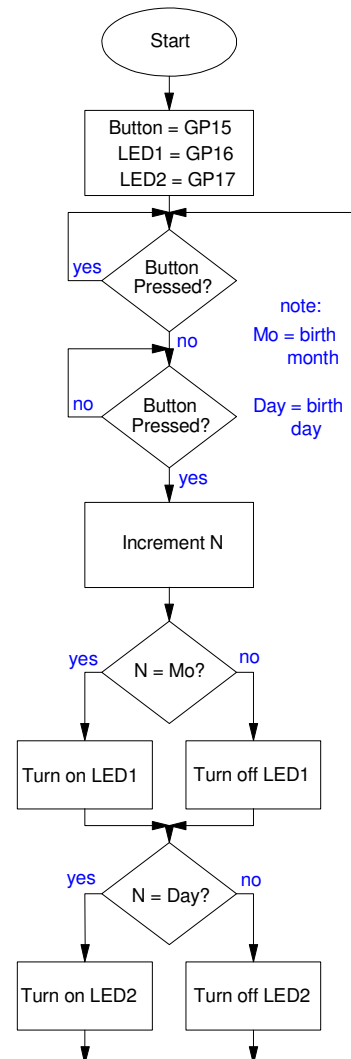
    while(Button.value() == 0):
        pass

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

    N += 1;

    if(N == 5): # birth month
        LED1.value(1)
    else:
        LED1.value(0)

    if(N == 14): # birth date
        LED2.value(1)
    else:
        LED2.vaue(0)
```

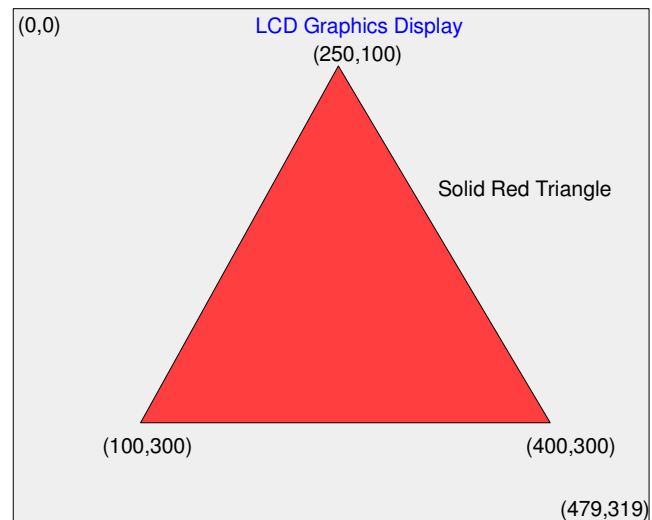


2) Python Programming - Subroutines.

Write a Python subroutine which draws a solid red triangle on the graphics display with vertices at

- $(x_0, y_0) = (250, 100)$
- $(x_1, y_1) = (100, 300)$
- $(x_2, y_2) = (400, 300)$

```
def Draw_Red_Triangle():  
    red = LCD.RGB(200, 0, 0)  
    for i in range(0, 201):  
        y = i + 100  
        xmin = 250 - (150/200)*i  
        xmax = 250 + (150/200)*i  
        LCD.Line(xmin, y, xmax, y, red)
```



note:

- There are many other solutions
- This draws horizontal lines since the display displays vertical or horizontal lines a lot faster than diagonal lines

3) Sensors: The following Python program reads the temperature from a DS18B20 sensor.

Modify this code so that

- It reads the temperature every 1.00 second
- It keeps track of
 - The number of readings (n),
 - The sum of temperatures (m1), and
 - The sum of temperature squared (m2)

$$m_1 = \sum(T)$$

$$m_2 = \sum(T^2)$$

- It then computes the mean, variance, and 90 percent confidence interval

$$x = \text{mean} = \frac{1}{n} \cdot m_1$$

$$v = \text{variance} = \left(\frac{1}{n-1}\right) \left(m_2 - \frac{1}{n} \cdot m_1^2\right)$$

$$x_{\max}, x_{\min} = 90\% \text{ confidence interval} = \text{mean} \pm 1.64 \sqrt{\frac{\text{variance}}{n}}$$

```
import machine, onewire, ds18x20, time

ds_pin = machine.Pin(4)
ds_sensor = ds18x20.DS18X20(onewire.OneWire(ds_pin))

roms = ds_sensor.scan()

n = m1 = m2 = 0

while(1):
    ds_sensor.convert_temp()
    time.sleep_ms(750)
    T = ds_sensor.read_temp(rom[0])

    n += 1

    m1 += T

    m2 += T**2

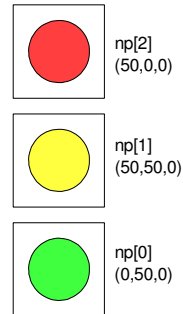
    x = m1/n

    if(n>1):
        v = (m2 - (m1**2)/n) / (n-1)
    else:
        v = 999

    Xmax = x + 1.64 * ( (v / n) ** 0.5 )
    Xmin = x - 1.64 * ( (v / n) ** 0.5 )
```

4) Interrupts & Stoplight: Write a Python program which uses interrupts to drive a stoplight.

- Edge Interrupt: Each time you press the button GP15 (falling edge interrupt), it counts mod 3.
 - Mode = 0 >> 1 >> 2 >> 3 >> repeat
- Timer Interrupt: Interrupt every 1.00 second



The variable Mode sets the operation:

- Mode = 0: set the color as green for 5 seconds, yellow for 1 second, red for 6 seconds, repeat
- Mode = 1: set the color to red
- Mode = 2: flashing red (red for 1 second, off for one second, repeat)
- Mode = 3: flashing yellow (yellow for 1 second, off for one second, repeat)

The main routine to set the stoplight to red is as follows (from test #3). The main loop does nothing (all work is done inside the interrupts)

Comments

- Code is on the next page
- Many other solutions exist

Edge Interrupts: (show in red)

- Edge interrupts detect the falling nedge of button 15
- This counts mod 4

Timer Interrupts (show in blue)

- Called every 1000ms (1 second)
- If Mode == 0, it goes through the green - yellow - red sequence
- If Mode == 1, it turns on red
- If Mode == 2, it alternates between red and off
- If Mode == 3, it alternates between yellow and off

Main Routine

- Does nothing
- Interrupts do all of the work

```

from machine import Pin, bitstream

timing = [300, 900, 700, 500]
np = Pin(12, Pin.OUT)

red = bytearray([0,0,0,0,0,0,0,50,0])
yellow = bytearray([0,0,0,50,50,0,0,0,0])
green = bytearray([50,0,0,0,0,0,0,0,0])
off = bytearray([0,0,0,0,0,0,0,0,0])

bitstream(np, 0, timing, red)

pin = Pin(15, Pin.IN, Pin.PULL_UP)
Mode = N = 0

def Button_Press(pin):
    global Mode
    Mode = (Mode + 1) % 4

pin.irq(trigger=Pin.IRQ_FALLING, handler=Button_Press)

tim = Timer()

def tic(timer)
    global N, Mode

    if(Mode == 0):
        N = (N + 1) % 15
        if(N < 5):
            bitstream(np, 0, timing, green)
        elif(N < 6):
            bitstream(np, 0, timing, yellow)
        else:
            bitstream(np, 0, timing, red)

    if(Mode == 1):
        bitsream(np, 0, timing, red)

    if(Mode == 2):
        N = (N + 1) % 2;
        if(N==0):
            bitstream(np, 0, timing, off)
        else:
            bitstream(np, 0, timing,, red)

    if(Mode == 3):
        N = (N + 1) % 2;
        if(N==0):
            bitstream(np, 0, timing, off)
        else:
            bitstream(np, 0, timing,, yellow)

tim.init(period = 1000, mode=Timer.PERIODIC, callback=tic)

while(1):
    pass

```