

ECE 476/676 - Homework #10

GPS & NeoPixels - Due Monday, November 17th

Simulated Gas Leak Game

Design an embedded system which uses the GPS sensor in your lab kit along with a NeoPixel to simulate locating a gas leak

- On startup, the GPS sensor is used to determine your current position
- Once your position is known, press GP15 to start the game.
- When GP15 is pressed,
 - You record your current GPS position (home position)
 - You also generate two random numbers (x, y) in the range of (-100, +100) are generated.
 - This is the location of a simulated gas leak relative to your home position
- The distance to the gas leak in meters is displayed on the LCD (just distance, not direction)
- The color of the NeoPixel changes based upon distance
- Beep once you're within 1m of the gas leak

Distance	> 100m	75m	50m	25m	5m	<1m
Color	blue	cyan	green	yellow	orange	red

Grading:

- (10 pt) Able to read the GPS sensor to get your position
 - If you can't get the GPS sensor to work, use the analog input
 - Pushing the joystick moves you left/right, up/down
- (10 pt) Generate a random spot for the gas leak on pressing GP15
- (10pt) Interrupts used to detect GP15 press
- (10 pt) Display your position in real time on the LCD display
 - Cross-hairs indicate zero position
- (10pt) Display the distance to the target on the LCD in real time as you move
- (10pt) Set the color of the NeoPixel based upon distance
- (10pt) Beep once you're within 1m of the gas leak
- (20pt) Demo

Code:

NeoPixel Drivers:

Set the color of the NeoPixel based upon range. Start with a subroutine that returns the (r, g, b) colors based upon range

```
def Set_Color(Range):
    if(Range < 50):
        b = 0
        g = int(250*Range/50)
        r = 250 - g
    elif(Range < 100):
        r = 0
        b = int(250*(Range-50)/50)
        g = 250 - b
    else:
        g = 0
        b = 250
        r = 0
    #print(Range, r, g, b)
    return(r, g, b)
```

Check:

- 1 returns red (245, 5, 0)
- 49 returns green (5, 245, 0)
- 51 returns green (0, 245, 5)
- 99 returns blue (0, 5, 245)

The NeoPixel is then driven using *bitstream()*

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

[r, g, b] = Set_Color(Range)
np_color = bytearray([g,r,b])
bitstream(np, 0, timing, np_color)
```

Reset the position on GP15 using interrupts:

```
# --- Edge Interrupt ---

Button15 = Pin(15, Pin.IN, Pin.PULL_UP)
def IntServe(Button15):
    global X0, Y0, X1, Y1
    X0 = 0
    Y0 = 0
    X1 = X0 + 200*random() - 100
    Y1 = X1 + 200*random() - 100
    LCD.Clear(Black)
    LCD.Box(0,0,479,319,Pink)
    LCD.Title('Gas Leak', Yellow, Black)
    LCD.Line(200,160,280,160,Grey)
    LCD.Line(240,120,240,200,Grey)
    Print(X0, Y0, X1, Y1)

Button15.irq(trigger=Pin.IRQ_FALLING, handler=IntServe)
```

Check: Print out the positions each reset. Results are

X0	Y0	X1	Y1
0	0	-74.6026	82.6752
0	0	26.4718	-80.4919
0	0	-44.3004	9.3763

Read the GPS Sensor

- No working GPS sensor, so I'm using the analog input

```
def Joystick():
    a0 = a2d0.read_u16()
    a1 = a2d1.read_u16()

    dx = (a0 - 0x8000) / 0x400
    dy = (a1 - 0x8000) / 0x400
    return(dx, dy)

(dx, dy) = Joystick()
X0 += dx * dt
Y0 += dy * dt
```

Calculate the range

```
Range = ( (X0-X1)**2 + (Y0-Y1)**2 ) ** 0.5
#print(X0, Y0, Range)
```

Beep if the Range < 1m

- Actually using 2 meters. One meter was too hard to get

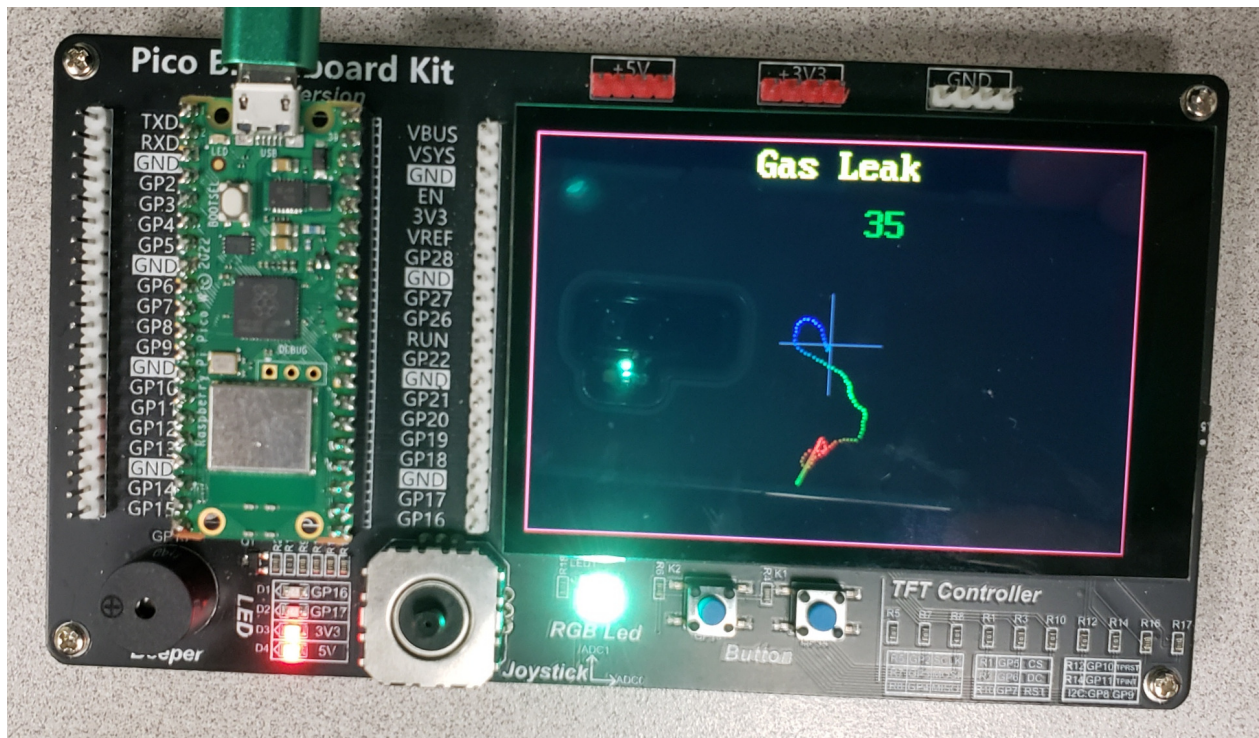
```
Beeper = Pin(13, Pin.OUT)

def Beep():
    Beeper.value(1)
    sleep(0.1)
    Beeper.value(0)

if(Range < 2):
    Beep()
    sleep(1)
```

Demo

- YouTube video posted on Bison Academy



Overall Code

```
from machine import UART, Pin, ADC, bitstream
from time import sleep
import LCD
from math import sin, cos, pi
from random import random

uart0 = UART(0, 9600)
uart0.init(9600, bits=8, parity=None, stop=1, tx=0, rx=1)

Button14 = Pin(14, Pin.IN, Pin.PULL_UP)
Beeper = Pin(13, Pin.OUT)

a2d0 = ADC(0)
a2d1 = ADC(1)

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

np_color = bytearray([0,0,0])

# --- Edge Interrupt ---

Button15 = Pin(15, Pin.IN, Pin.PULL_UP)
def IntServe(Button15):
    global X0, Y0, X1, Y1
    X0 = 0
    Y0 = 0
    X1 = X0 + 200*random() - 100
    Y1 = Y1 + 200*random() - 100
    LCD.Clear(Black)
    LCD.Box(0,0,479,319,Pink)
    LCD.Title('Gas Leak', Yellow, Black)
    LCD.Line(200,160,280,160,Grey)
    LCD.Line(240,120,240,200,Grey)

Button15.irq(trigger=Pin.IRQ_FALLING, handler=IntServe)

Error_Flag = 0

def Beep():
    Beeper.value(1)
    sleep(0.1)
    Beeper.value(0)

def GPS_Read_Line(chan):
    flag = 0
    n = 0
    msg = ''
    while(flag == 0):
        if(chan == 0):
            x = uart0.read(1)
        else:
            x = uart1.read(1)
        if(x != None):
            x = ord(x)
            if(chr(x) == '$'):
                msg = ''
            if(x == 13):
                flag = 1
            else:
                msg = msg + chr(x)
    return(msg)
```

```

def GPS_Read(chan):
    flag = 0

    while(flag == 0):
        msg = GPS_Read_Line(chan)
        b = msg.split(',')
        if(b[0] == '$GPRMC'):
            if(len(b[3]) > 3):
                flag = 1
                time = b[1]
                Xdeg = b[3][0:2]
                Xmin = b[3][2:]
                Ydeg = b[5][0:3]
                Ymin = b[5][3:]
                Speed = b[7]
    return([float(time), float(Xmin), float(Ymin), float(Speed)])

def GPS_GPGGA(chan):
    flag = 0

    while(flag == 0):
        msg = GPS_Read_Line(chan)
        b = msg.split(',')
        if(len(b) > 8):
            #print(b[0], b[1], b[2], b[4], b[9])
            if(b[0] == '$GPGGA'):
                if(len(b[2]) > 3):
                    flag = 1
                    time = b[1]
                    Y = b[2][2:]
                    X = b[4][3:]
                    Z = b[9]
    return([float(time), float(X), float(Y), float(Z)])

def Joystick():
    a0 = a2d0.read_ul6()
    a1 = a2d1.read_ul6()

    dx = (a0 - 0x8000) / 0x400
    dy = (a1 - 0x8000) / 0x400
    return(dx, dy)

def Set_Color(Range):
    if(Range < 50):
        b = 0
        g = int(250*Range/50)
        r = 250 - g
    elif(Range < 100):
        r = 0
        b = int(250*(Range-50)/50)
        g = 250 - b
    else:
        g = 0
        b = 250
        r = 0
    #print(Range, r, g, b)
    return(r, g, b)

Navy = LCD.RGB(0,0,5)
White = LCD.RGB(250,250,250)
Pink = LCD.RGB(250,150,150)
Black = LCD.RGB(0,0,0)
Yellow = LCD.RGB(250,250,0)
Grey = LCD.RGB(100,100,100)

```

```

LCD.Init()
LCD.Clear(Black)

X0 = 0
Y0 = 0
X1 = X0 + 200*random() - 100
Y1 = X1 + 200*random() - 100

LCD.Box(0,0,479,319,Pink)
LCD.Title('Gas Leak', Yellow, Black)
LCD.Line(200,160,280,160,Grey)
LCD.Line(240,120,240,200,Grey)
dt = 0.1

Range = 100

while(1):
    (dx, dy) = Joystick()
    X0 += dx * dt
    Y0 += dy * dt

    Range = ( (X0-X1)**2 + (Y0-Y1)**2 ) ** 0.5
    #print(X0, Y0, Range)

    [r, g, b] = Set_Color(Range)
    Color = LCD.RGB(r, g, b)

    LCD.Pixel2(240+X0, 160-Y0, Color)
    np_color = bytearray([g,r,b])
    bitstream(np, 0, timing, np_color)

    LCD.Number2(Range, 3, 0, 230, 50, Color, Black)
    sleep(dt)

    if(Range < 2):
        Beep()
        sleep(1)

```