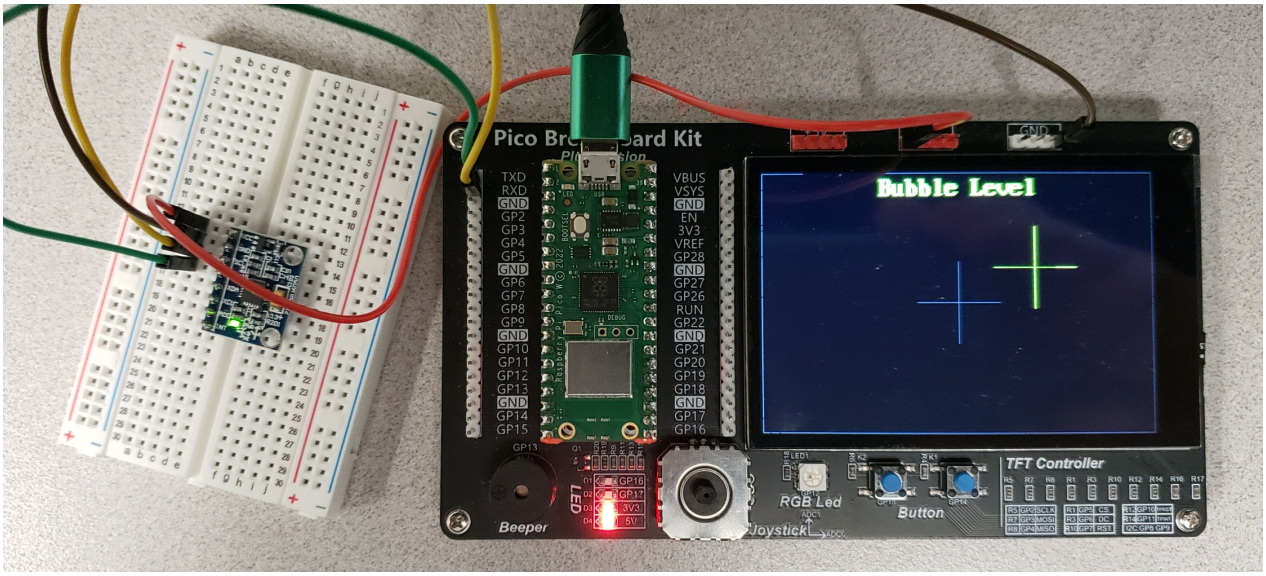
ECE 476/676 - Homework #9

Acceleration & NeoPixels

Bubble Level

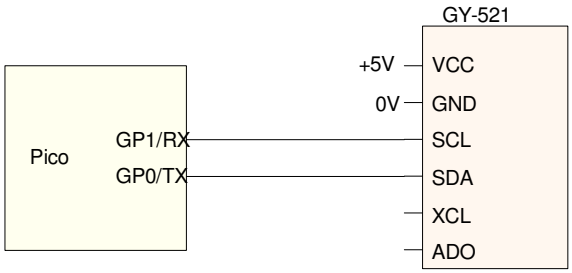
Build a bubble-level with the Pi-Pico and a 52Pi Breakout Board Plus

- Measure acceleration with a GY-521 accelerometer
- Display the acceleration on the LCD display as a cross-hair
  - 0g acceleration x and y results in the cross-hair being in the middle of the display
  - Xg acceleration in the x direction displaces the cross-hair by 100X pixels left/right
  - Yg acceleration in the y direction displaces the cross-hair by 100Y pixels up/down
- Turn on the NeoPixel when the net x-y acceleration is small
  - green: |(x,y)| acceleration < 0.05
  - red: |(x,y)| acceleration < 0.01



Hardware:

Connect to the I2C port 0 on pins 1 and 2



Software:

First check that the sensor is seen on the I2C port. Write a short test program

```
from machine import Pin, I2C, bitstream
from time import sleep_ms, sleep

i2c = machine.I2C(0, scl=machine.Pin(1), sda=machine.Pin(0))

# Print out any addresses found
devices = i2c.scan()
if devices:
    for d in devices:
        print('I2C Device Found:',hex(d))

addr = devices[0]
print('Communicating with ', hex(addr))
```

```
I2C Device found:  0x68
Communicting with 0x68
```

Next, check that I'm able to read the sensor. Write another short test program

```
from machine import Pin, I2C, bitstream
from time import sleep_ms, sleep

def reg_write(i2c, addr, reg, data):
    msg = bytearray()
    msg.append(data)
    i2c.writeto_mem(addr, reg, msg)

def reg_read(i2c, addr, reg, nbytes=1):
    if nbytes < 1:
        return bytearray()
    data = i2c.readfrom_mem(addr, reg, nbytes)
    return data

def accel_read(reg):
    x = reg_read(i2c, addr, reg, 2)
    y = (x[0] << 8) + x[1]
    if(y > 0x8000):
        y = y - 0x10000
    y = y / 16000
    return(y)

i2c = machine.I2C(0, scl=machine.Pin(1), sda=machine.Pin(0))

# Print out any addresses found
devices = i2c.scan()
if devices:
    for d in devices:
        print('I2C Device Found:',hex(d))

addr = devices[0]
print('Communicating with ', hex(addr))

# set bandwidth = 5Hz
reg_write(i2c, 0x68, 0x1a, 6)
# set range = 2g
reg_write(i2c, 0x68, 0x1c, 0x00)
RANGE = 2
# set clock freq
reg_write(i2c, 0x68, 0x6b, 0)

while(1):
    ax = accel_read(0x3b) * RANGE
    ay = accel_read(0x3d) * RANGE
    az = accel_read(0x3f) * RANGE
    print(ax, ay, az)
    sleep(0.1)
```

```
I2C Device Found: 0x68
Communicating with  0x68
 0.0025 -0.088  1.6025
-0.006  -0.0725 1.6155
 0.001  -0.0785 1.63
-0.009  -0.0659 1.6295
-0.0035 -0.0765 1.624
-0.0055 -0.0590 1.632
 0.0015 -0.13   1.638
```

Now that works, add LCD routines to display the position in cross-hairs.

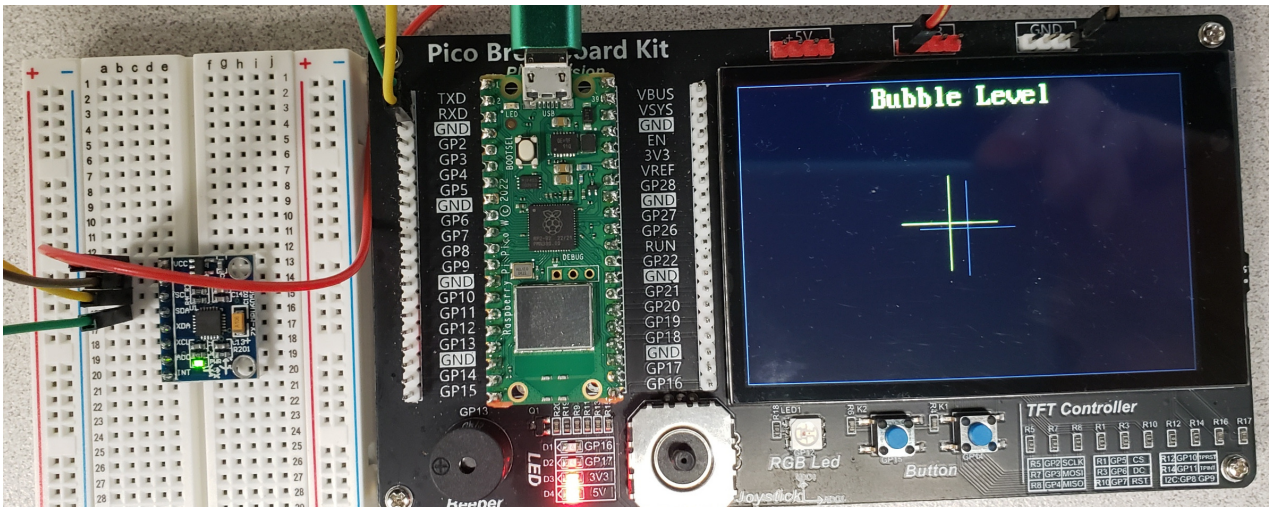
- Blue = GY521 functions
- Red = NeoPixel functions
- Green = LCD functions

```
import time
from machine import Pin, I2C, bitstream
import LCD
from time import sleep_ms, sleep

# I2C Communications with GY-521 Sensor

def reg_write(i2c, addr, reg, data):
    msg = bytearray()
    msg.append(data)
    i2c.writeto_mem(addr, reg, msg)

def reg_read(i2c, addr, reg, nbytes=1):
    if nbytes < 1:
        return bytearray()
    data = i2c.readfrom_mem(addr, reg, nbytes)
    return data

def accel_read(reg):
    x = reg_read(i2c, addr, reg, 2)
    y = (x[0] << 8) + x[1]
    if(y > 0x8000):
        y = y - 0x10000
    y = y / 16000
    return(y)

i2c = machine.I2C(0, scl=machine.Pin(1), sda=machine.Pin(0))

# Print out any addresses found
devices = i2c.scan()
if devices:
    for d in devices:
        print('I2C Device Found:',hex(d))

addr = devices[0]
print('Communicating with ', hex(addr))

x = y = z = 0

# set bandwidth = 5Hz
reg_write(i2c, 0x68, 0x1a, 6)
# set range = 2g
reg_write(i2c, 0x68, 0x1c, 0x00)
RANGE = 2
# set clock freq
reg_write(i2c, 0x68, 0x6b, 0)

# NeoPixel routines

timing = [300, 900, 700, 500]
np = Pin(12, Pin.OUT)
red = bytearray([0,20,0])
green = bytearray([20,0,0])
blue = bytearray([0,0,20])
black = bytearray([0,0,0])
```

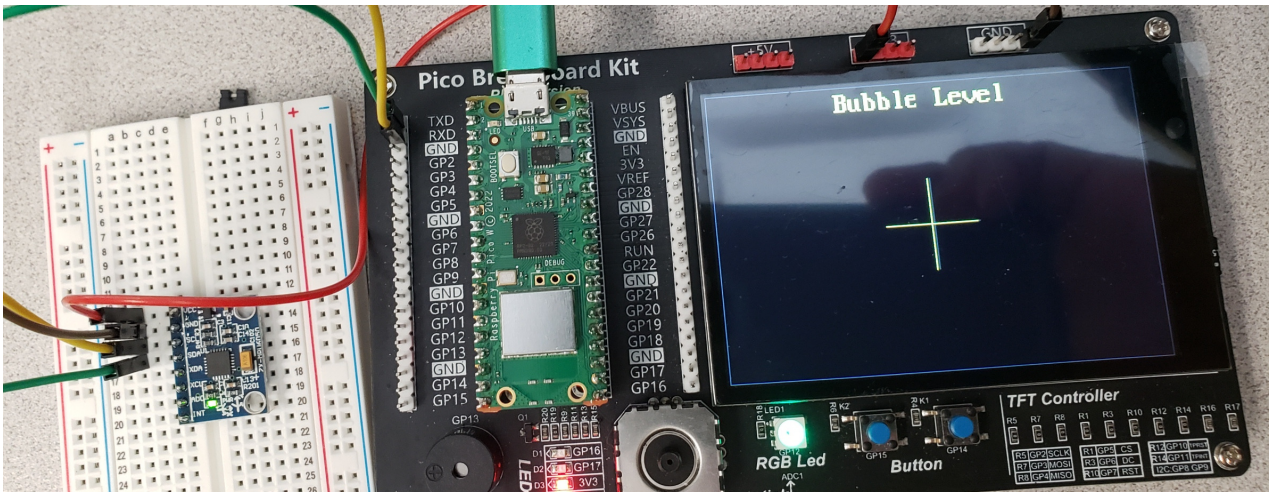
```
# LCD Display Routines

White = LCD.RGB(250,250,250)
Black = LCD.RGB(0,0,0);
Grey = LCD.RGB(100,100,100)
Red = LCD.RGB(250,0,0);
Yellow = LCD.RGB(250,250,0);

x0 = 240
y0 = 160

LCD.Init()
LCD.Clear(Black)
LCD.Box(1,1,479,319,Grey)
LCD.Line(x0-50,y0,x0+50,y0,Grey)
LCD.Title('Bubble Level', Yellow, Black)


while(1):
    LCD.Line(x0+x,y0+y-50,x0+x,y0+y+50,Black)
    LCD.Line(x0+x-50,y0+y,x0+x+50,y0+y,Black)

    ax = accel_read(0x3b) * RANGE
    ay = accel_read(0x3d) * RANGE
    az = accel_read(0x3f) * RANGE

    x = ax * 500
    y = ay * 500
    z = az * 500

    LCD.Line(x0,y0-50,x0,y0+50,Grey)
    LCD.Line(x0-50,y0,x0+50,y0,Grey)
    LCD.Line(x0+x,y0+y-50,x0+x,y0+y+50,Yellow)
    LCD.Line(x0+x-50,y0+y,x0+x+50,y0+y,Yellow)

    Error = (x*x + y*y) ** 0.5
    if(Error < 1):
        bitstream(np, 0, timing, red)
    elif(Error < 5):
        bitstream(np, 0, timing, green)
    else:
        bitstream(np, 0, timing, black)

    sleep_ms(20)
```

Test Results:

- Main Loop: 12.358ms (not counting the sleep_ms(20) routine)
- GY-521 Sensor Aknowledged
  - ID = 0x068
- Acceleration Readings
  - x, y, z acceleration change as sensor is tilted
  - Acceleration goes from -1 to + 1 as rotated (scaling is correct)
- Graphics Display
  - Shows a cross related to the tilt of the sensor
  - Up/down moves the cursor up/down
  - Left/right moves the cursor left/right
- NeoPixel lights up when cursor is at (0,0) position
  - Green when close (within 5 pixels)
  - Red when within 1 pixel



Cross shows the acceleration of the sensor



Green LED turns on when close (within 5 pixels)