Lab 3 – Exercise 1a: Accelerometer

- Measure ANY acceleration
- Pitch & roll angle – due to gravity g
- Movement involve force acceleration, also measured
- Accelerometer measurement of tilt angle is NOISY

Lab 3 – Exercise 1b: Gyroscope

- Gyro gives angular velocity, not angle
- Needs to integrate to get angle
- Integration = accumulation
- Also accumulate errors – causing drift (or dc offset)

Lab 3 – Exercise 1: Measuring Angel of tilt – the IMU

- The IMU – insertia measurement unit – has built in 3-axis accelerometer and 3-axis gyroscope
- Easy to access from Matlab using PyBench:
  \[
  \begin{align*}
  &\{p, r\} = \text{pb.get_accel}(); \quad \%\ p, r = \text{pitch & roll angle in radians} \\
  &\{x, y, z\} = \text{pb.get_gyro}(); \quad \%\ x, y, z = \text{rate of rotation in 3-axes in rad/sec}
  \end{align*}
  \]
- Pitch angle – plane pointing up or down
- Roll angle – plane pointing left or right
- Angle can be in unit radian or degree: degrees = radians * 180 / \pi
- Generally use radian for calculations; use degree of display

- Learn usefulness and limitations of accelerometer and gyroscope
Lab 3 – Exercise 1c: Gyroscope

```
subplot(2,1,1);
axis([0 end_time -90 90]);
```

```
subplot(2,1,2);
axis([0 end_time -90 90]);
```

Lab 3 – Exercise 2: 3D visualization

```
model = IMU 3D();
model.draw(fig1, p, r, 'Accelerometer');
```

Lab 3 – Exercise 3: Complementary Filter - Concept

```
\[ \theta(t) = \alpha \times (\theta + \dot{\theta} \Delta t) + (1 - \alpha) \times \rho \]
```

where 
- \( \alpha \) = scaling factor chosen by users and is typically between 0.7 and 0.98
- \( \rho \) = accelerometer angle
- \( \theta \) = output angle computed
- \( \dot{\theta} \) = gyroscope reading of the rate of change in angle
- \( \Delta t \) = time interval between gyro readings

- What happens if \( \dot{\theta} \) is zero? Effectively average out the value of \( \rho \)
- What happens if \( \dot{\theta} \) has a small error? Effectively reduce this error over time

Lab 3 – Exercise 3: Complementary Filter - Implementation

```
(1 - \alpha)
```

```
\int \dot{\theta} \, dt
```

```
x \times \theta
```

Better angle measurement \( \theta \)

Better angle measurement \( \theta \)
Lab 3 – Exercise 4: Untethered – OLED Display

```python
# Create peripheral objects
b_LED = LED(4)           # blue LED
pot = ADC(Pin('X11'))    # 5k ohm potentiometer to ADC input on pin X11

# I2C connected to V0, V10 (I2C bus 2) and V11 is reset low active
oled = OLED_936(pinout={'sda': 'Y10', 'scl': 'Y9', 'res': 'Y8'}, height=64,
                 external_vcc=False, i2c_address=61)

oled.poweron()
oled.init_display()

# Simple Hello world message
oled.draw_text(0, 8, 'Hello World!!')  # each character is 6x8 pixels

tic = pyb.millis()           # store start time
while True:
    b_LED.toggle()           # read elapsed time
    toc = pyb.millis()
    oled.draw_text(0, 20, 'Delay time: ({:.3f}) sec'.format((toc-tic)/1000))
    oled.draw_text(0, 40, 'POT reading: ({:.5d})'.format(pot.read()))
    tic = pyb.millis()        # start time
    oled.display()
    delay = pyb.rng()%1000    # Generate random number btw 0 and 999
    pyb.delay(delay)         # delay in milliseconds
```