

GPS: How Fast Can You Run?

1) (30 points): Write a Python program which measures how fast you can run using a GPS sensor.

- Record your speed with a GPS sensor
- Start new recording by pressing button GP15
- While recording, detect the maximum speed seen from the GPS sensor, and
- Display the top three speeds on graphics display

Include

- Your Python program
- Data showing it is working
- A photo of your graphics display showing your three fastest speeds

hw_10_1.py was modified from *25 GPS Speedometer* on Bison Academy. Additions include *get_max_speed_mph*, along with support for a top 3 recorded speeds.

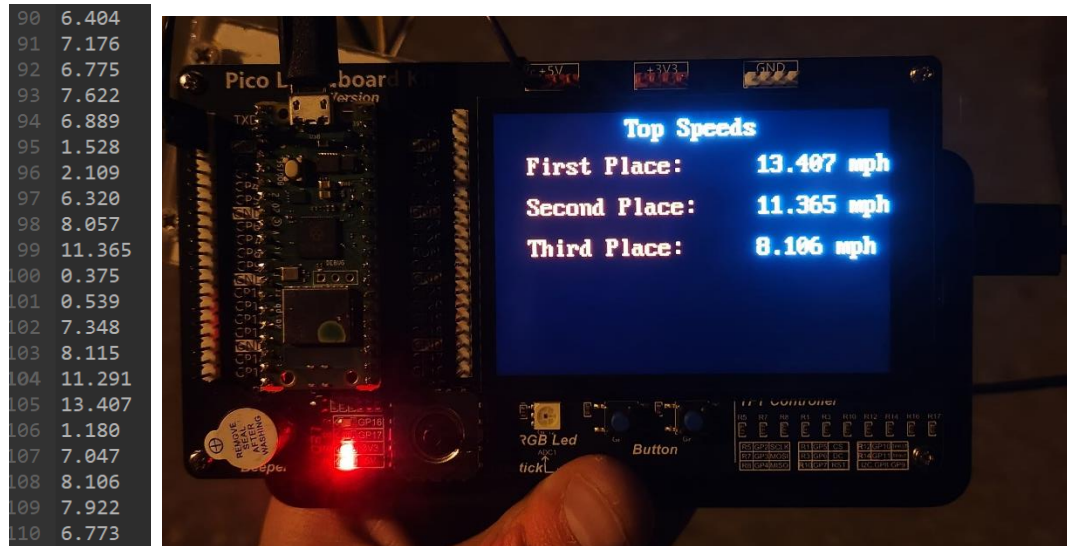
```
def get_max_speed_mph(file, current_index):
    f = open(file, "r")
    data = f.readlines()
    max_speed_mph = -1
    for i in range(current_index, len(data)):
        speed_mph = float(data[i])
        if speed_mph > max_speed_mph:
            max_speed_mph = speed_mph
    return max_speed_mph

LCD.Title('Top Speeds', White, Black)
LCD.Text2('First Place: ', 40, 50, Orange, Black)
LCD.Text2('Second Place: ', 40, 100, Orange, Black)
LCD.Text2('Third Place: ', 40, 150, Orange, Black)
```

```
while(Button14.value() == 1):
    Error_Flag = 1
    while(Error_Flag == 1):
        Error_Flag = 0
        [t, x, y, v] = GPS_Read()
    if(Button15.value() == 0):
        Record_Flag = not Record_Flag
        if(Record_Flag):
            Beep()
            f = open("GPS_Speed.txt", "a")
            current_index = len(f.readlines())
            print('Recording')
            LCD.Text('Recording',5,5,Red,Black)
        else:
            Beep()
            sleep_ms(100)
            Beep()
            f.close()
            print('Recording Stopped')
            LCD.Text(' ',5,5,Red,Black)

            # parse the file to get the max speed for the recording and display it on the LCD Screen
            max_speed_mph = get_max_speed_mph("GPS_Speed.txt", current_index)
            speed.append(max_speed_mph) # add the velocity in mph to top speeds
            print(f"Max Speed = {max_speed_mph:.3f}")
            speed.sort(reverse=True) # sort the list to get the top 3 speeds
            print(f"List={speed[0]:.3f}, {speed[1]:.3f}, {speed[2]:.3f}")
            sleep_ms(50)
            LCD.Text2(f'{speed[0]:.3f} mph ', 320, 50, White, Black)
            LCD.Text2(f'{speed[1]:.3f} mph ', 320, 100, White, Black)
            LCD.Text2(f'{speed[2]:.3f} mph ', 320, 150, White, Black)
        while(Button15.value() == 0):
            pass
    if(Record_Flag):
        f.write(f'({v * knots_to_mph):.3f} \n')
```

Snapshot of the data from the GPS sensor, along with the top 3 speeds:



2) (10 points): Demo your program

- Short video preferred
- Photos also work

See [HW10-1.mp4](#) for video demonstration

BlueTooth & Motor Speed:

3) (30 Points): Write a Python program which allows you to control the speed of a DC motor using your cell phone and a bluetooth interface. Some options are:

- Use PWM to vary the speed and direction of the DC motor
- Use commands to set the direction (CW or CCW) and speed (000 to 100)
- Use a long string to input both (+100, -085, etc)

When completed, you should be able to set the voltage to the motor from -100% to +100%

Include:

- Your Python program
- Data showing it is working (CW, CCW, speed or voltage changes with data input)

```
def set_speed(num):
    return int((num / 100) * 65535)

def update_motor_speed(duty):
    speed = set_speed(abs(duty))
    if duty > 0:
        a_out_pwm.duty_u16(speed)
        b_out_pwm.duty_u16(0)
    else:
        a_out_pwm.duty_u16(0)
        b_out_pwm.duty_u16(speed)
    sleep_ms(10)

def LCD_reinit():
    LCD.Clear(Black)
    LCD.Text2('Duty Cycle: ', 50, 50, Yellow, Black)
    LCD.Title('Bluetooth Motor Control', White, Black)

def between(num, bound):
    if num > bound:
        num = bound
    elif num < -bound:
        num = -bound
    return num

def on_rx(data):
    print("Data received: ", data)
    try:
        global motor_duty
        motor_duty = int(data[0:4]) # look for 0:4 to account for negative symbol (range of +/- 100)
        motor_duty = between(motor_duty, 100)
        update_motor_speed(motor_duty)
        global flag
        flag = 1
    except:
        print('invalid data entry')
```

```
LCD.Init()
White = LCD.RGB(250,250,250)
Black = LCD.RGB(0,0,0)
Yellow = LCD.RGB(250,250,0)
LCD_reinit()
flag = 1

while(1):
    if sp.is_connected():
        sp.on_write(on_rx)
    else:
        update_motor_speed(0)

    if(flag):
        print(f'Duty Cycle: {motor_duty}%')
        LCD.Text2(str(motor_duty) + ' % ', 320, 50, Yellow, Black)
        flag = 0
```

hw_10_2.py

4) (10 points); Demo your program

- Short video preferred
- Photos also work

See HW10-2.mp4