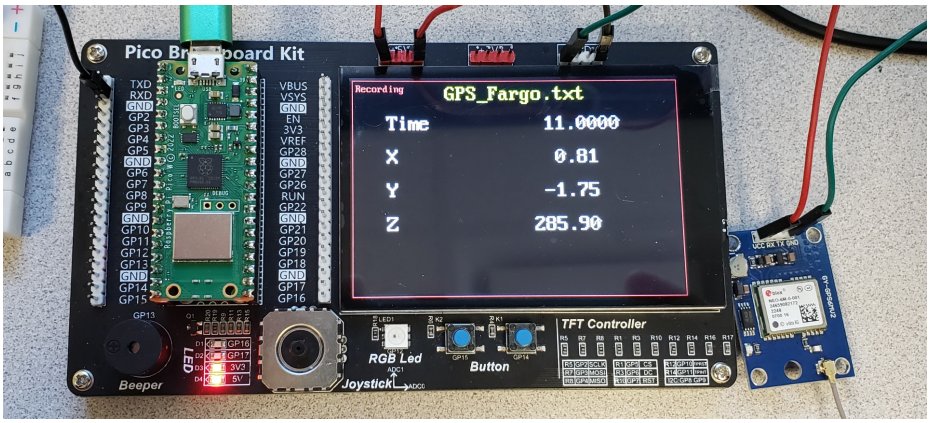
ECE 476/676 - Homework #10

GPS & SCI Communications - Due Monday, April 14th

Design an embedded sytem which uses the GPS sensor in your lab kit to measure your position, speed, and/or heading. Display this information on your LCD display.

How Flat is Fargo?



1) Requirements: Measure the elevation of Fargo going along

- 12th Ave North from NDSU to 45th St (east/west road)
- 45th Street from 12th Ave North to 72nd Ave South

Inputs: GPS Sensor

Outputs: Text file, LCD display,

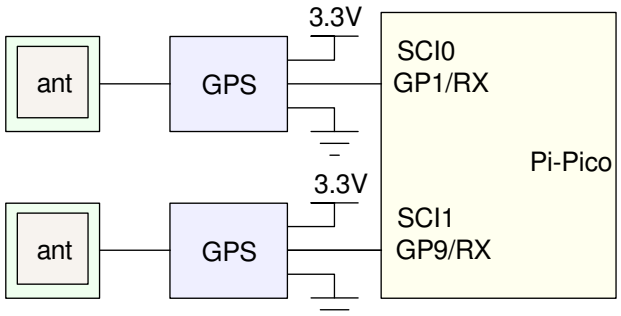
How they relate:

- Use GPS to find your position (latitude, longitude, height)
- Display it on the LCD display
- When GP14 is pressed, reset the home position and toggle recording on and off
- When recording, record to a text file

45th Ave South facing North

2) Code: Give a Python program which reads the GPS sensor and meets your requirements.

Hardware:



Software:

Step 1: Read GPS messages

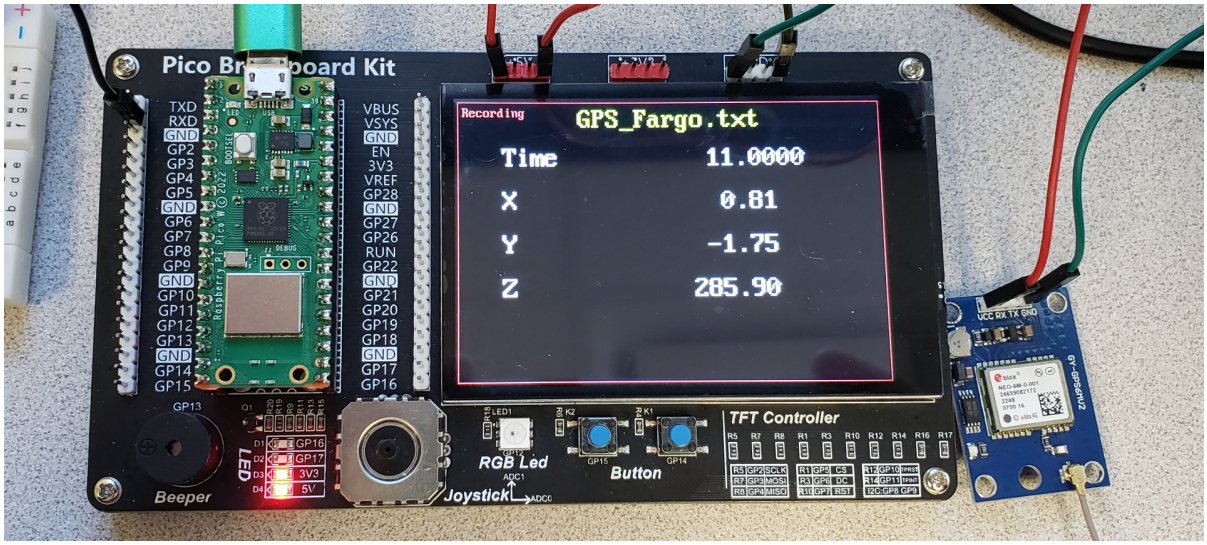
```
from machine import UART, Pin

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)
Button15 = Pin(15, Pin.IN, Pin.PULL_UP)
Beeper = Pin(13, Pin.OUT)

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)

while(1):
    msg = GPS_Read_Line(chan)
    print(msg)
```

```
$GPGSA,A,3,29,18,15,13,20,23,,,,,,,2.52,1.27,2.18*0F
$GPGSV,4,1,16,01,21,283,17,05,55,055,07,07,02,027,,11,08,099,*7C
$GPGSV,4,2,16,13,42,105,19,15,45,156,12,16,11,324,,18,50,296,19*7F
$GPGSV,4,3,16,20,26,062,14,23,20,238,20,25,00,206,,26,16,289,*75
$GPGSV,4,4,16,29,64,190,19,30,03,056,,46,28,221,,48,30,216,*77
$GPGLL,4649.55763,N,09652.11931,W,173924.00,A,A*7F
$GPRMC,173925.00,A,4649.55729,N,09652.11947,W,1.143,,180724,,,A*67
$GPVTG,,T,,M,1.143,N,2.116,K,A*20
```

Step 2: Pull out latitude, longitude, and height from the GPGGA message

```
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)])

while(1):
   [t, X, Y, Z] = GPS_GPGGS(0)
   print(t, X, Y, Z)
```

```
      dT        dX        dY        Z
      1.0      -0.04      0.07     283.2
      2.0      -0.07      0.13     283.2
      3.0      -0.14      0.24     283.1
      4.0      -0.24      0.45     282.8
      5.0      -0.04      0.46     282.8
```



Displaying GPS information for latitude, longitude, and elevation
time, X, and Y are relative to home position

Step 3: Save the data to a text file

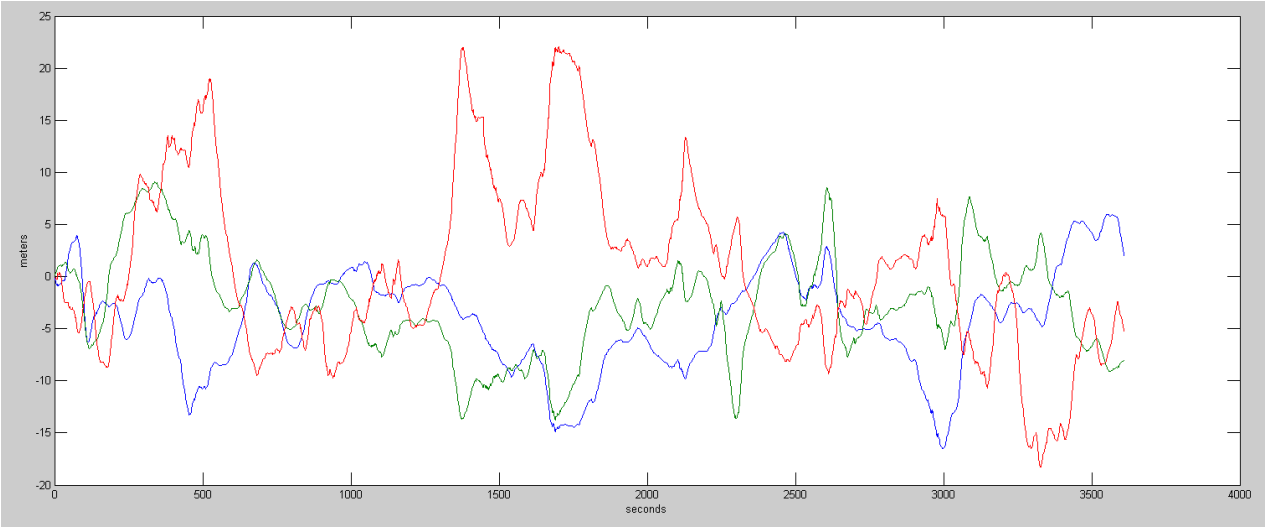
- GP14 sets the home position and time (T0, X0, Y0)
- GP14 toggles the 'save to file' option on and off
- GP15 closes the file and stops the program

```
while(Button15.value() == 1):
    [time, X, Y, Z] = GPS_GPGGA(0)
    dT = time - Tref
    dX = ( X - Xref ) * 1269.448
    dY = ( Y - Yref ) * 1852.222
    print(dT, dX, dY, Z)

    LCD.Number2(dT,   5, 2,  250,  50, White, Black)
    LCD.Number2(dX,   5, 2,  250, 100, White, Black)
    LCD.Number2(dY,   5, 2,  250, 150, White, Black)
    LCD.Number2(Z,    5, 2,  250, 200, White, Black)

    if(Button14.value() == 0):
        Record_Flag = not Record_Flag
        Xref = X
        Yref = Y
        Tref = time
        if(Record_Flag):
            Beep()
            f = open(FileName, "a")
            print('Recording')
            LCD.Text('Recording',5,5,Pink,Navy)
        else:
            Beep()
            sleep(0.1)
            Beep()
            f.close()
            print('File Closed')
            LCD.Text('         ',5,5,Pink,Navy)
        while(Button14.value() == 0):
            pass

    if(Record_Flag):
        msg0 = str('{:9.1f}'.format(dT) + ' ' )
        msg1 = str('{:9.2f}'.format(dX) + ' ' )
        msg2 = str('{:9.2f}'.format(dY) + ' ' )
        msg3 = str('{:9.1f}'.format(Z) + ' ' )
        f.write(msg0 + msg1 + msg2 + msg3 + '\n')

Beep()
Beep()
Beep()
print('stop')
```

3) Validation: Collect data to verify your embedded system works and meets your requirements.

What is the noise level on the GPS readings when stationary?

Collect 10,011 data points as the sensor is stationary, sitting on my desk

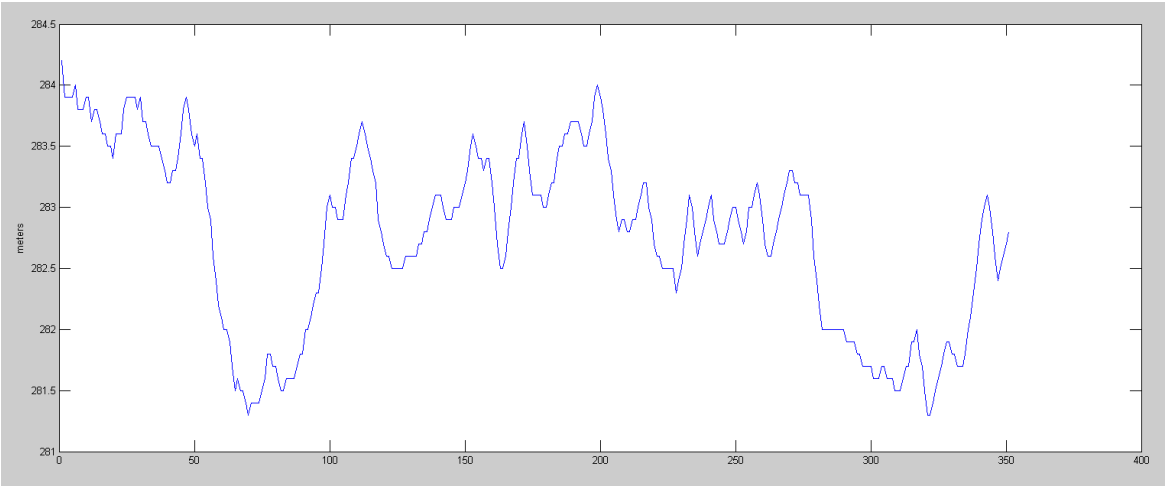


GPS location: X = blue, Y = green, Z = red

	X	Y	Z
mean	-4.0551 m	2.6667 m	0.5577 m
st dev	4.7873 m	4.8271 m	8.4071 m

From stationary readings, it looks like the altitude readings are good to within +/- 14 meters (90% confidence interval)

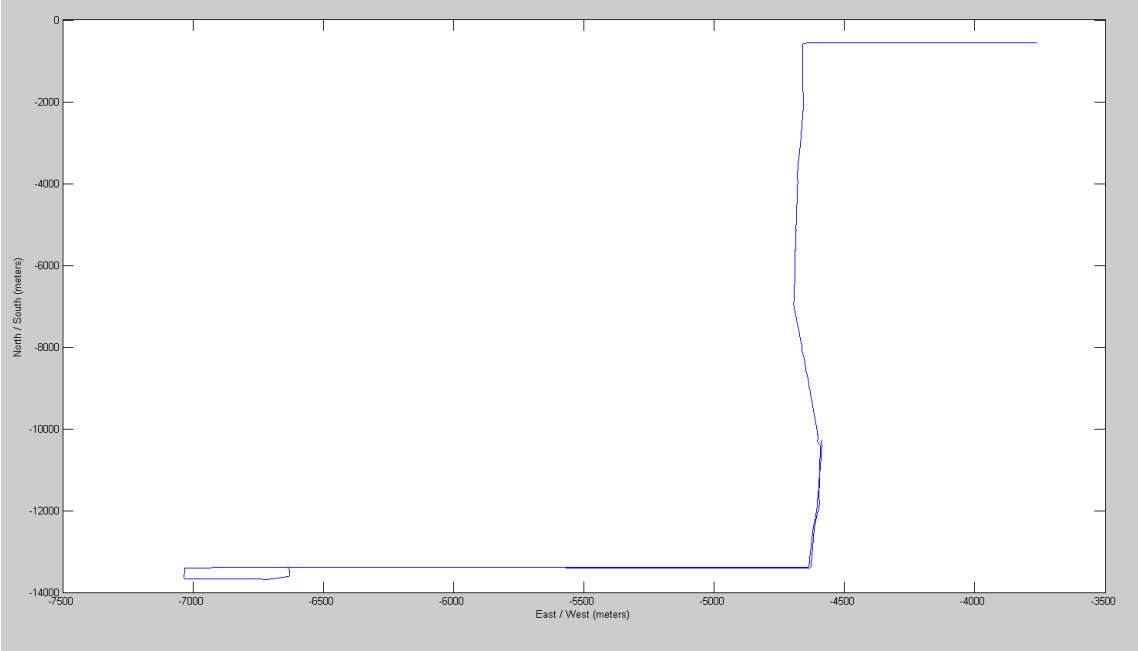
Data while Moving: Drive down 45th Ave south from I29 to 72nd Ave south:



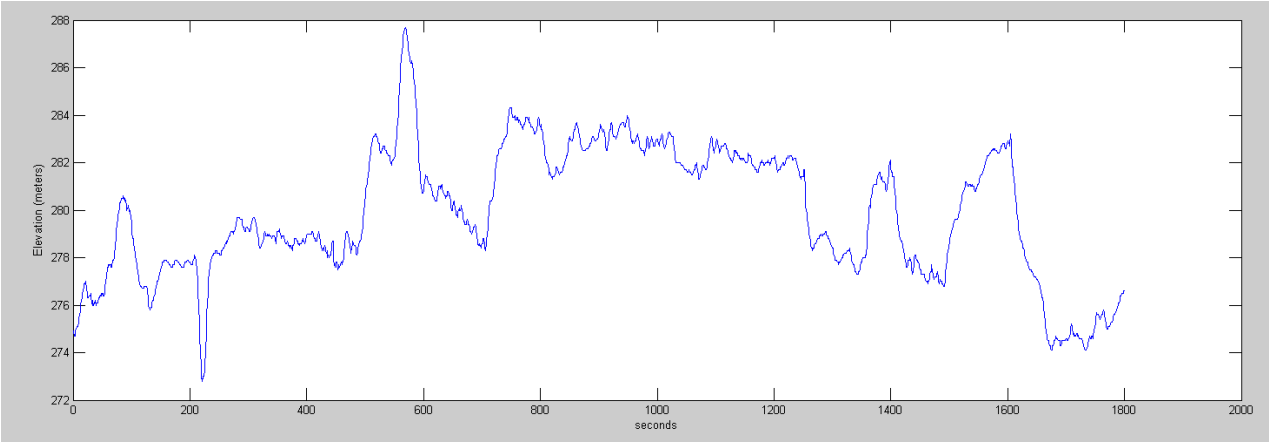
Elevation of 45th Ave South

	Elevation
mean	282.7644m
st dev	0.7316m

Statistics while moving



Drive along 12th Ave North (upper right) going west
Go south along 45th Ave (vertical line)
Go west along 72nd Ave South (botton line)
Turn around in Horace (lower left)



Elevation along this rout

	Elevation
mean	279.8675 m
st dev	2.7475 m

Statistics for Elevation from NDSU Campus to Horace ND

Conclusion:

- GPS readings are a lot more accurate when moving
  - s = 8.4071m when stationary
  - s = 0.7316m when moving
- Fargo is flat to within 4.5 meters (90% confidence interval)

4) Demo. Video or in person.

Driving North on 45th Ave South ckise ti 72nd Ave South