Project Based Engineering Instrumentation With CircuitPython

A Brief Textbook Presented to the
Student Body of the University of South Alabama

Last Update: June 28, 2022

Manuscript Changes

1. Original tutorials in Google Docs created
2. Tutorials moved to LaTeX on this Github
3. December 21st, 2021 - Updated links for manuscript and hardware
4. Tutorials purchased by Tangibles that Teach and moved to url `https://tangibles-that-teach.gitbook.io/instrumentation-lab-manual/-MbMx70LQzRmEG_hS7Ld/`
5. May 30th, 2022 - Tangibles that teach went out of business and chapter began the move to Github
6. June 28th, 2022 - Work began on a Chromebook. Unfortunately the Figures folder is not back up on Git. As such a main_latest.pdf has been created that's the latest full version. The main.pdf is the version created by the Chromebook so it has new chapters but none of the older chapters. Figure files are now backup on Git but only figures from the 'Voltage Potentiometer' are currently there.

Changes Needed

1. All chapters from TTT need to be moved here
2. More theory is needed in this book or direction to further reading for the students
3. The new kit has a CPB - Might need to add a chapter explaining the difference and maybe even having the students getting the bluetooth to work.
4. The photo of the button lab could be better.
5. The circuit photo for the LSM6DS33 is not correct
6. Equations on thermistor need to be expanded
7. Equations relating voltage from protocol to Lux needs to be included
8. Example plots for light, sound, acceleration, etc needs to be expanded
9. Pendulum lab must be done in one of two ways. Either the pendulum is attached to a potentiometer or the CPX is mounted to the end of a string and data logged on board the CPX itself. The potentiometer is nice because you can record data with your laptop but the string idea is cool because you use the accelerometer. In either case you can make some really long pendulums
10. 3D printing a disc with holes on the outside to eventually mount to a shaft would be a really cool angular velocity sensor lab. Tangibles that teach could easily include a 3D printed disc that can mount to a pencil for ease of rotation. Could also include the CAD drawings so students can print more or even edit the design for better or worse performance.
11. Buying some load cells with the HMC converter and including them in the kit would add a whole lot different labs
12. Buying some magnetometers to measure magnetic field and do some sensor fusion would be neat. Could do roll, pitch and yaw calibration if we included a magnetometer.
13. Right now a lab just on roll and pitch estimation would be possible. Pendulum lab pretty much introduces them to this but could easily do an rc aircraft lab where they build an aircraft out of foam with an elevator and aileron so that the servos responds to roll and pitch change. There is a lab right now with just pitch but perhaps we could add roll to it.
14. Another cool project idea would be for the students to take temperature and light data on a cloudy day. Then have them infer if the amount of sunlight affected the temperature of the thermistor. They could plot the data with light on the x-axis and temperature on the y-axis and draw conclusions based on the plot they generate.
15. Could also have them take temperature and light data over the course of a whole day to plot sunrise and sunset and watch the ambient temperature rise and fall
16. For the aliasing lab, have the students sample as fast as possible and obtain the natural frequency of the system. Then have them sample at 1.0, 2.0 and 3.0 times the natural frequency they obtained. I originally picked 1,10 and 100 as arbitrary sampling frequencies and it would have been better to do 2,4 and 6.
17. On cool lab would be to take light data during sunset and watch light and temperature plummet.
18. Add this lab on frequency for notes

Acknowledgements

The author, Dr. Carlos Montalvo would like to acknowledge a few key members who made this textbook possible. First and foremost I would like to thank Adafruit for their entire ecosystem of electronics, tutorials, blogs and forums. Much of what I have learned here to teach Instrumentation was from Adafruit and the Adafruit Learn system and specifically people like Lady Ada and John Park who have helped shape CircuitPython and the Circuit-Playground Express to what it is today. I would also like to thank Dr. Saami Yazdani for creating the blueprint for Instrumentation at my university by creating a laboratory environment for an otherwise totally theoretical course. His course was the foundation for this textbook and for that I thank him for showing the way. I'd like to also thank and acknowledge Tangibles that Teach for giving me the opportunity to morph this loose set of projects into a textbook that can be used for multiple universities and classrooms and of course help students learn and acquire knowledge through creating.

About this textbook

This textbook has been designed with the student and faculty member in mind. First, this textbook goes hand in hand with Engineering Instrumentation taught at the undergraduate level at many universities. The course begins with simple plotting and moves into data analysis, calibration and more complex instrumentation techniques such as active filtering and aliasing. This course is designed to get students away from their pen and paper and build something that blinks and moves as well as learn to process real data that they themselves acquire. There is no theory in these projects. It is all applied using the project based learning method. Students will be tasked with downloading code, building circuitry, taking data all from the ground up. By the end of this course students will be well versed in the desktop version of Python while also the variant CircuitPython designed specifically for microelectronics from Adafruit. After this course students will be able to understand Instrumentation at the fundamental level as well as generate code that can be used in future projects and research to take and analyze data. Python is such a broad and useful language that it will be very beneficial for any undergraduate student to learn this language. To the professors using this textbook, 1 credit hour labs are often hard to work into a curriculum and "live" demonstrations in the classroom cost time and money that take away from other faculty duties. I've created this kit and textbook to be completely stand-alone. Students simply need to purchase the required materials and follow along with the lessons. These lessons can be picked apart and taught sequentially or individually on a schedule suited to the learning speed of the course. I hope whomever reads and learns from this textbook will walk away with an excitement to tinker, code and build future projects using microelectronics and programming.

Contents

1 Measuring Voltage Across a Potentiometer

1.1 Parts List

1. Laptop
2. CPX/CPB
3. USB Cable
4. Potentiometer
5. Resistor (the Ohms depends on how large your potentiometer is)
6. Breadboard

1.2 Learning Objectives

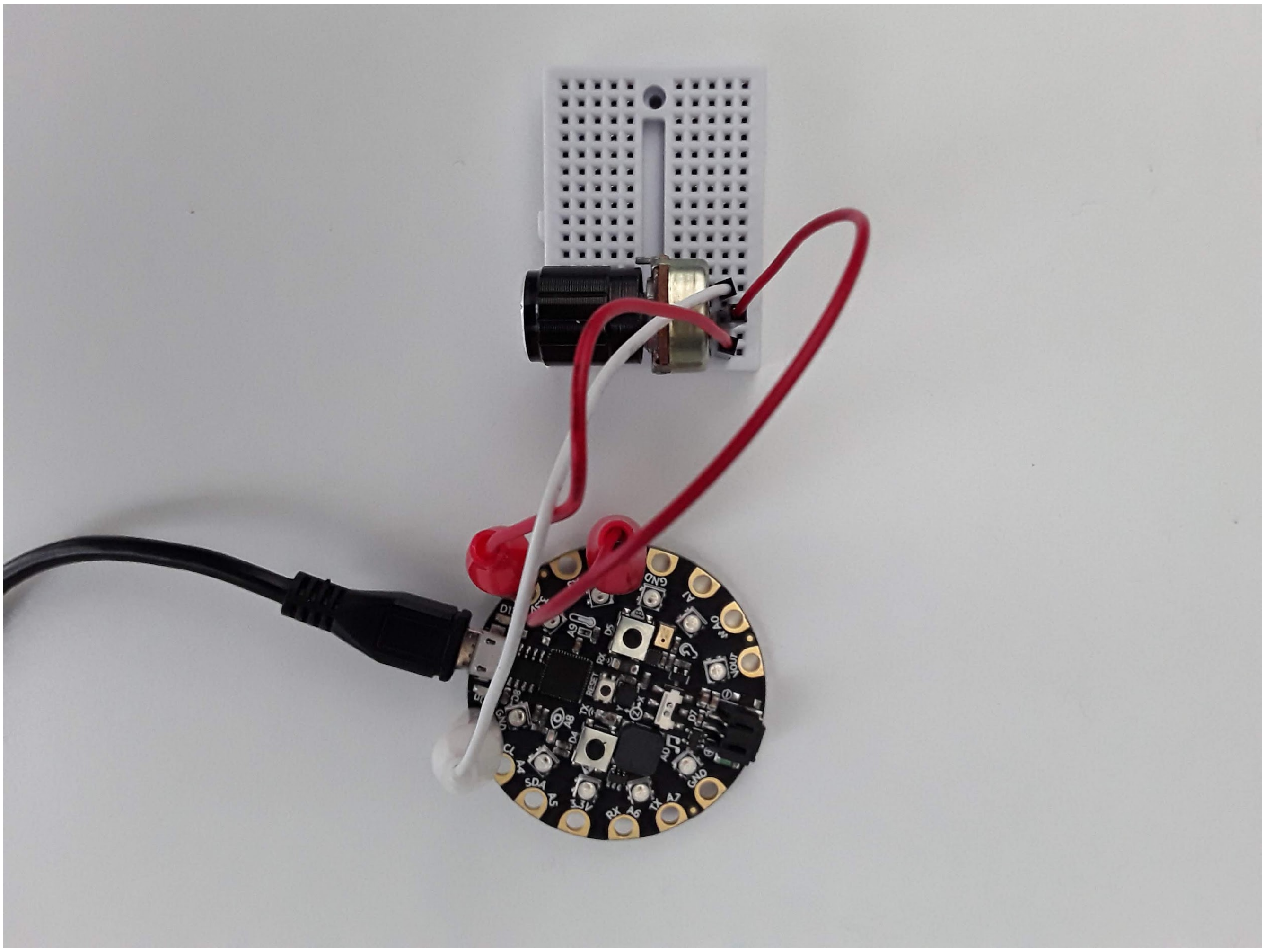
1. Understand voltage division of resistors in series
2. Measure an analog signal on the CircuitPlayground
3. Understand the binary measurement done by the analog to digital conversion (ADC)

1.3 Getting Started

At this point you've learned about analog to digital converters (ADC). It turns out that the CPX has 8 analog ports hooked up to a 3.3V logic 16 bit ADC. The input range on the ADC is 0 to 3.3V and the output range is 0 to 65536 which is 2^{16} hence 16 bits. In order to get accustomed to the ADC on the CPX, we're going to do a simple example where we measure the voltage drop across a potentiometer. You can read about potentiometers online if you wish. Basically though, a potentiometer is a variable resistance resistor that changes resistance by turning a knob. The knob changes the connection point of a wire and thus the length of the wire. This in turn changes the resistance. Potentiometers come in all shapes and sizes. Here are some examples.

Here's my circuit all hooked up. Two legs are connected to 3.3V and GND while the middle leg of the potentiometer is connected to pin A2.

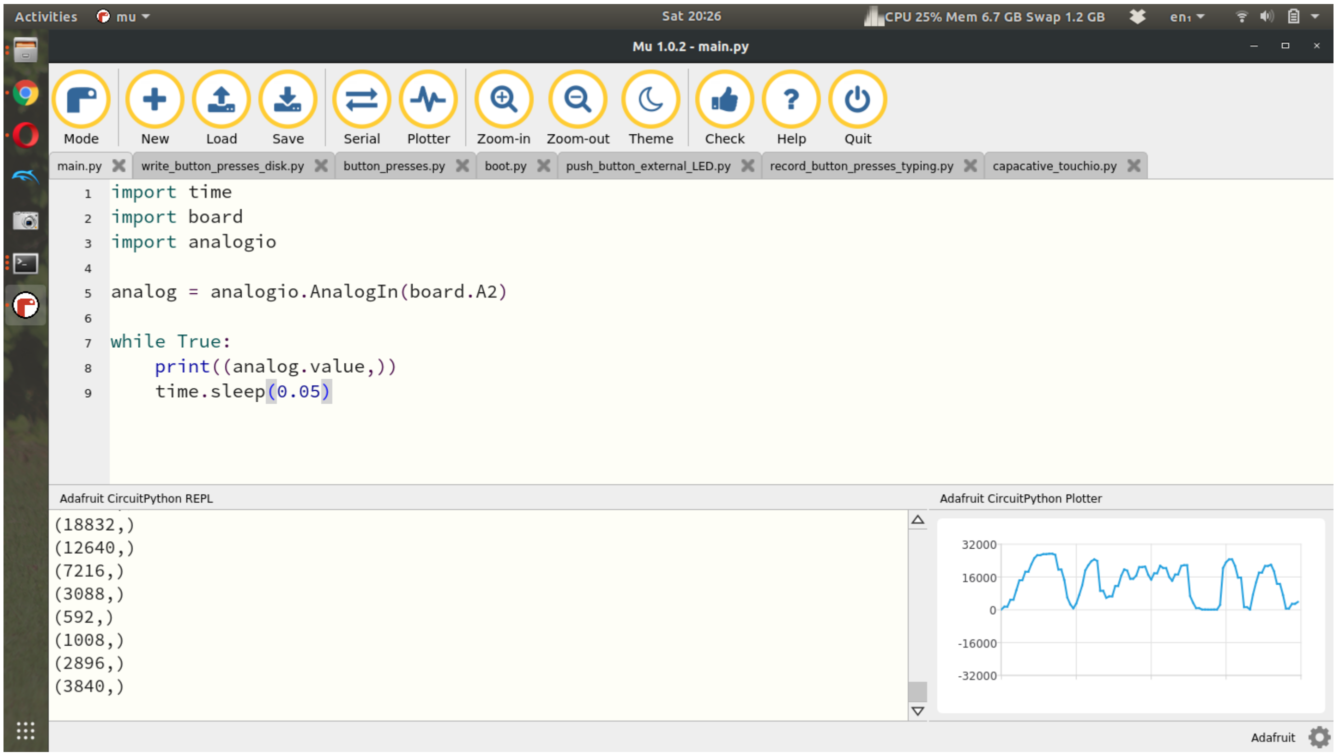
CAUTION!!!: Some potentiometers do not have enough resistance when turned all the way down. I suggest that you put a resistor in between the third leg and ground. Some experimenters have melted plastic or gotten really hot. One student even blew up a potentiometer.



There is a relevant Adafruit Learn Tutorial to help with the *analogio* module but I'll explain the minimum required here to get some analog values plotted in *Plotter* and Python on your computer. First let's take a look at some simple example code to read an analog signal and plot it using the *Plotter*.

```
1  import time
2  import board
3  import analogio
4
5  analog = analogio.AnalogIn(board.A2)
6
7  while True:
8      print((analog.value,))
9      time.sleep(0.05)
```

In the example code above, lines 1-3 again import the necessary modules with *analogio* being the new module here. Line 5 creates the analog object by attaching pin A2 to the analog function. Lines 7-9 then simple read the analog value and print it to *Serial* and the *Plotter*. Running this code on my laptop and turning the knob on the potentiometer produces this output. My potentiometer has a very large knob on the front and is easy to turn. Some potentiometers have a small screw on top that you need to turn with a screwdriver. Turning the screw or the knob results in chaning the resistance and therefore changing the voltage read by the CPX.



For this lab I want you to spin the potentiometer all the way to one side and then the other while recording time and the analog value. I then want you to plot the data with time on the x-axis and voltage on the y-axis. Remember to convert a digital output to voltage you just need to use the equation below where D is the raw value from the analog port. 3.3V is the range of the ADC and 2^{16} is the maximum value the ADC can represent.

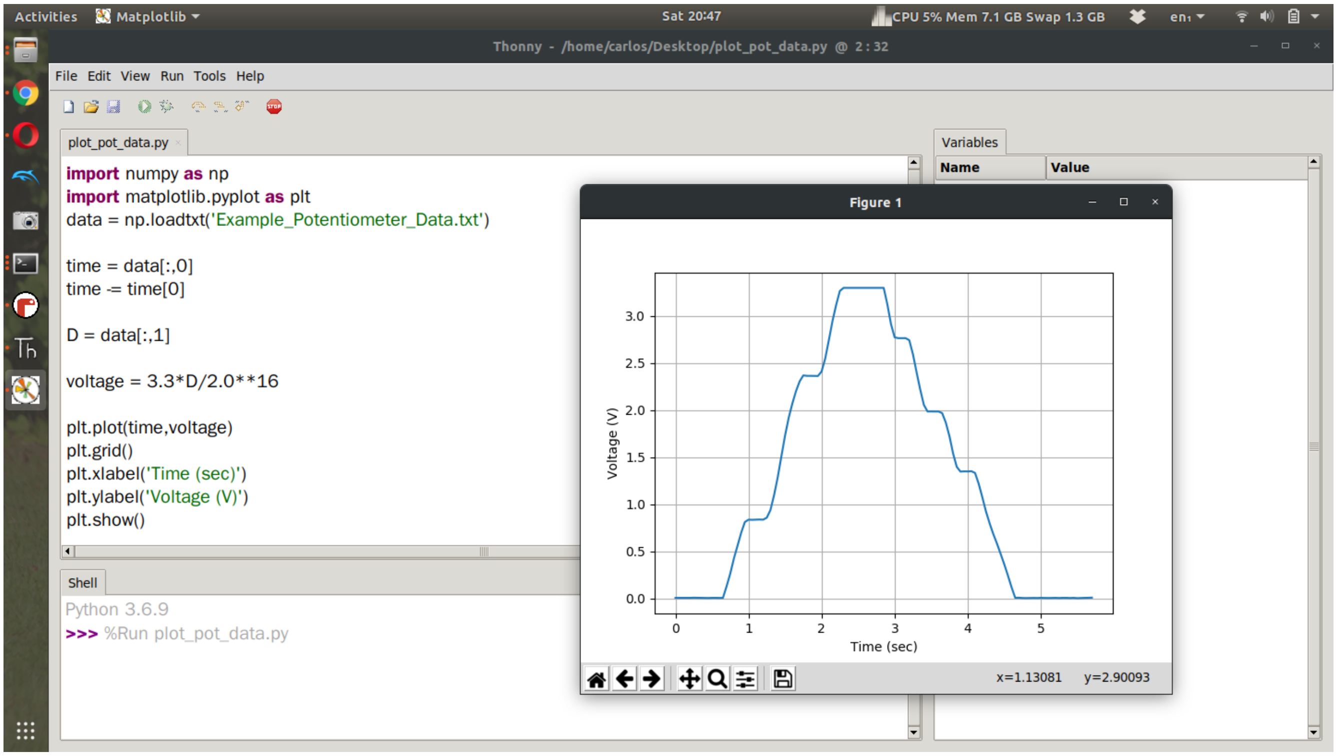
$$V = \frac{3.3D}{2^{16}} \tag{1}$$

After doing this experiment myself, this is the plot I obtain. The code is not provided as reading data and plotting has been discussed in a previous lab (See chapter **??**). From the screenshot though you can see how I convert the digital output to an analog signal.

NOTE THAT ON LINE 6 IT READS

```
time -= time[0]
```

Notice the minus sign in front of the equal sign. That effects a lot.



Your assignment for this lab is to do the same as I've done above. Wire up the potentiometer, read the analog signal and plot it in Python on your desktop computer. I've made some youtube videos on first just creating the circuit and plotting the data and then another video where I write data to the CPX using method 3.

1.4 Assignment

Once you've done that upload a PDF with all of the photos and text below included. My recommendation is for you to create a Word document and insert all the photos and text into the document. Then export the Word document to a PDF. For videos I suggest uploading the videos to Google Drive, turn on link sharing and include a link in your PDF.

1. Include a video of you wiggling the potentiometer and watching the digital signal in the Plotter in Mu go up and down (make sure your face is in the video at some point and you state your name) - 50%
2. Embed your plot of voltage vs time in a document and also include your code In Python both from the CPX and in Thonny or Spyder to plot - 50%

2 Wind Speed from Pitot Probe

2.1 Parts List

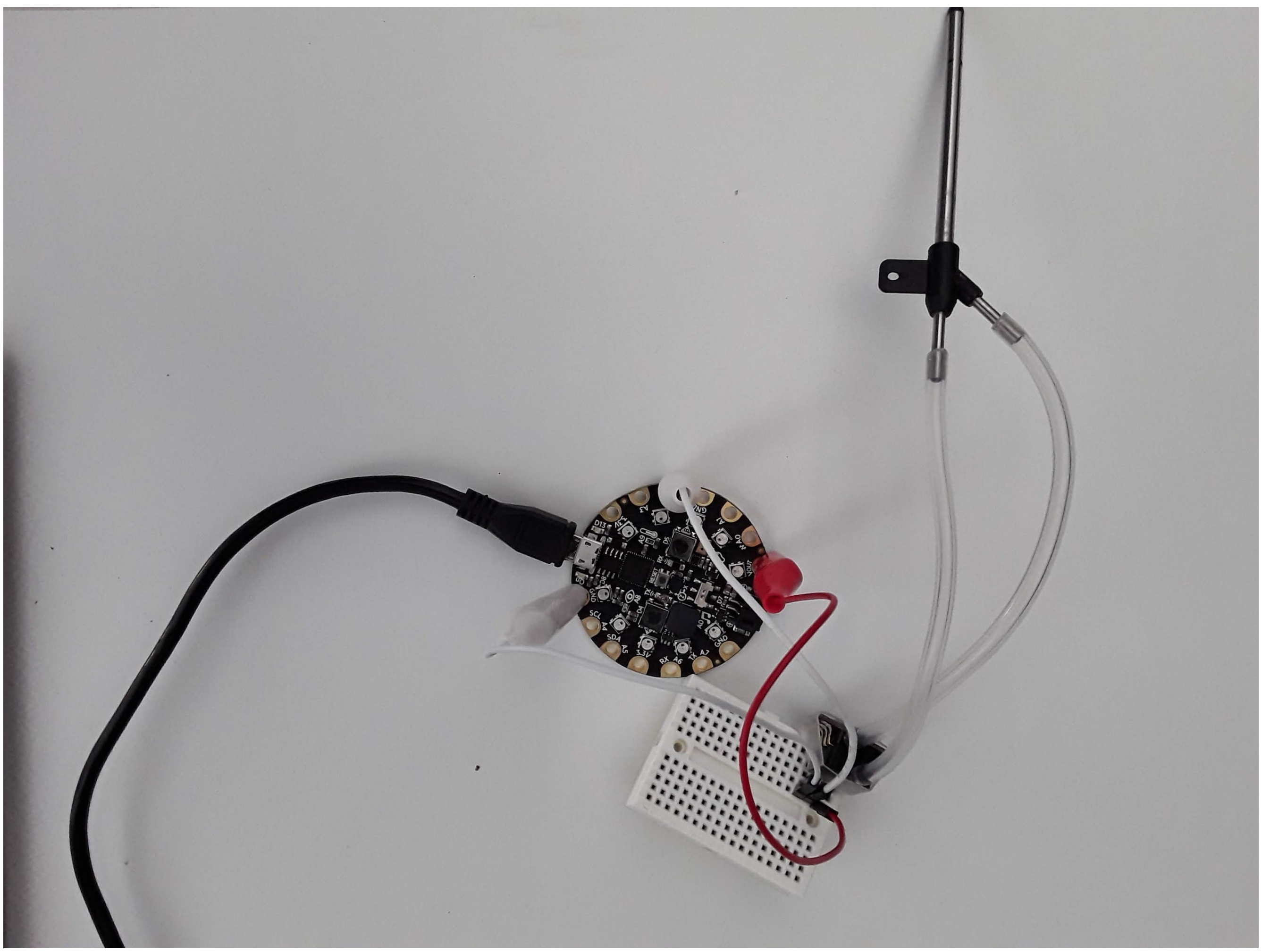
1. Laptop
2. CPX/CPB
3. USB Cable
4. Alligator Clips (x3)
5. Pitot Probe (Not included in kit at the moment so will need to buy this separately or borrow one)
6. Breadboard

2.2 Learning Objectives

1. Understand how pitot probes works
2. Understand the relationship between a voltage signal from a pitot probe to a pressure value
3. Understand the relationship between pressure and windspeed

2.3 Getting Started

Although a CPX has numerous sensors built in, you can easily augment the capabilities of the CPX using either I2C or just the ADC on board the CPX. In this lab, if you purchased a pitot probe you will be able to do this assignment. Since you don't need the pitot probe for very long you can always borrow one from some other team. Let's talk about the hardware and the wiring to get this to work.



The pitot probe has two pressure taps that measure ambient pressure and stagnation pressure. These taps move through two silicon tubes to a pressure transducer that has a strain gauge in separating both pressures. When the pressure on one side of the transducer is larger than the other, it will flex the membrane and create strain. This strain runs through a wheatstone bridge with a voltage offset to the pin labeled analog. The transducer has 3 pins, +5V, GND and Analog. It is pretty straightforward how to wire this up but remember that +5V needs to go to VOUT, GND to GND and Analog to any analog pin. I chose pin A2. At that point it's very simple to just print the analog signal in bits to Serial. I've done this below. The code is the same analog code that we've used in the past.

```
import time
import board
import analogio

analog = analogio.AnalogIn(board.A2)

while True:
    print((analog.value,))
    time.sleep(0.05)
```

The goal of the experiment is to take pitot probe data for 15 seconds with no wind, then 15 seconds of data with a fan on and then 15 seconds of no wind data. You'll need to use one of the datalogging methods (See chapter **??**) to log both time and pitot probe analog value. Once you have that data, import the data into Python on your desktop computer and convert the signal to windspeed. In order to convert the analog value to windspeed you need to first convert the analog value to voltage. Remember that the ADC on the CPX is going to convert the analog signal from the pitot probe to a digital output. So use this equation first to convert the analog signal (which I call D) to voltage. The 2 to the 16th power represents the 16 bit ADC.

$$V = \frac{3.3D}{2^{16}} \tag{2}$$

Before converting Voltage to windspeed we need to first subtract off the bias from the pitot probe. I explain this process in this accelerometer video. I've done this project before and have posted a video on Youtube about Converting Pitot Probe Data to Windspeed. *There is a typo in the video. V1 is supposed to have a sqrt()).* Once you have computed the bias you can compute the change in pressure using the equation below which converts the change in voltage to Pascals (V_b is the voltage bias you obtain when the wind is off). The data sheet states that the voltage is linearly proportional to pressure in Pascals which is nice.

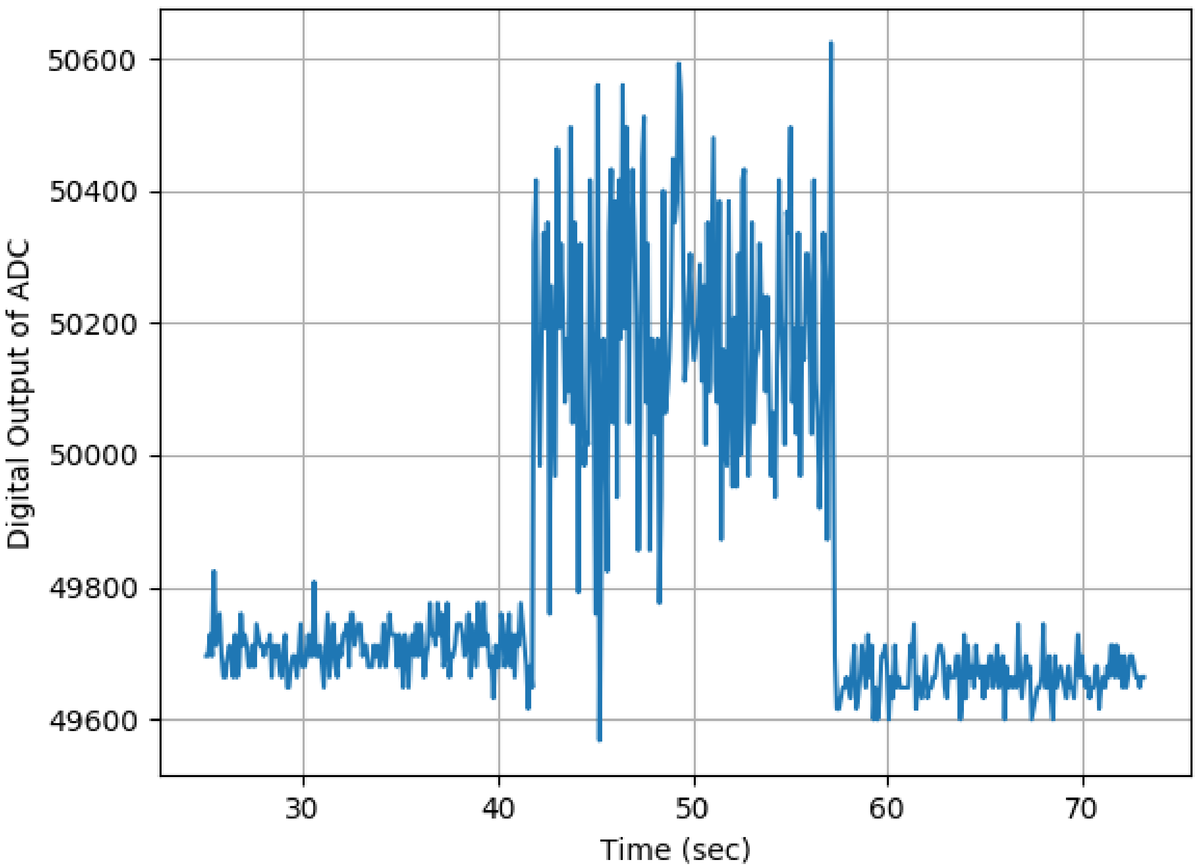
$$P = 1000(V - V_b) \tag{3}$$

Using pressure, the equation below can be used to compute windspeed, where U is the windspeed and the density at sea-level (ρ) is 1.225 kg/m^3.

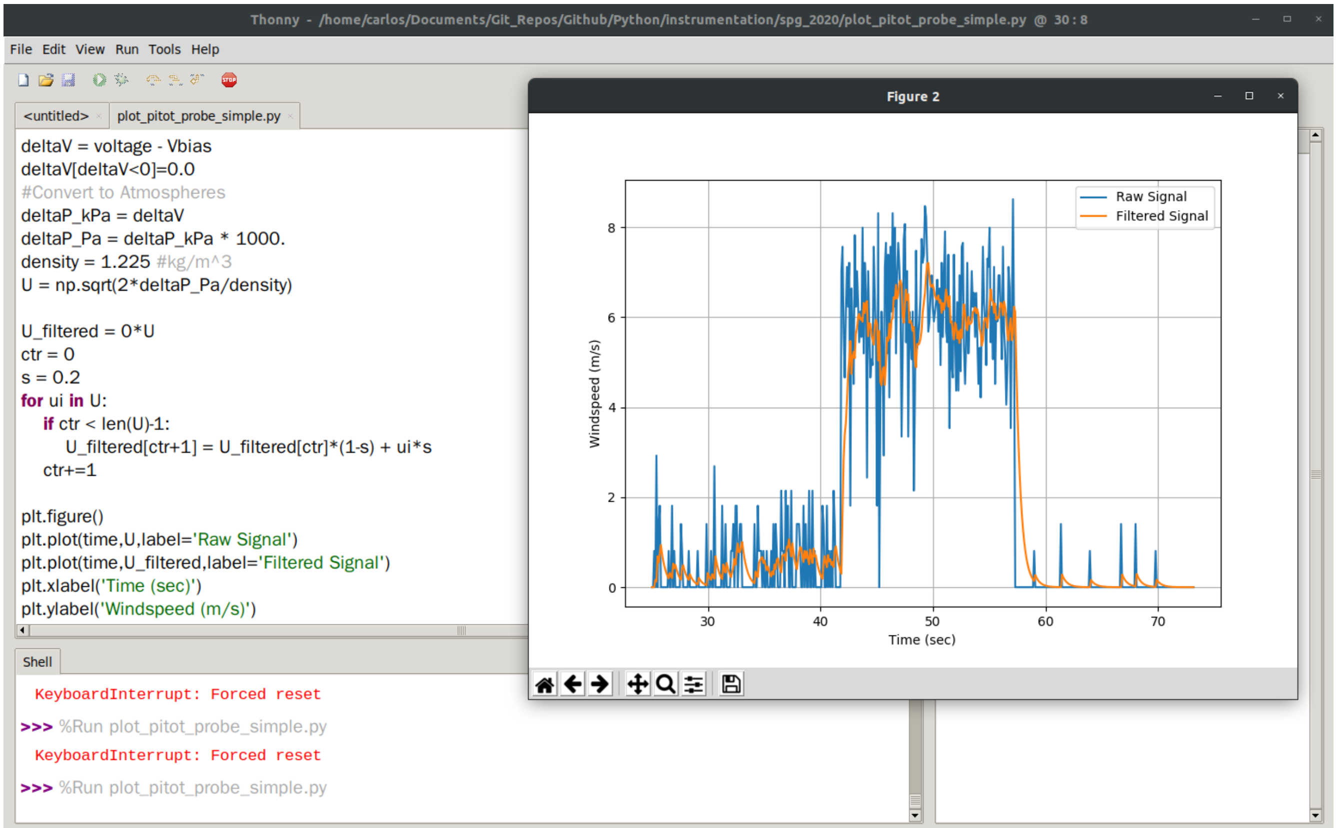
$$U = \sqrt{\frac{2P}{\rho}} \tag{4}$$

Using your data, create a plot of windspeed with time on the x-axis and windspeed on the y-axis. Some steps that might help you as you complete this project. First, have Mu plot the voltage coming from the pitot probe. If you've done everything right it will not be zero. The data sheet says there's an offset voltage of 2.5V so you will hopefully get something around 50,000 when you don't blow into the pitot probe. 50,000 multiplied by $3.3/2^16$ is around 2.5V. Make a note of that average value you get so you can subtract it off later. Once you've verified you're reading the pitot probe correctly, blow into the pitot probe and using the Plotter or Serial, verify that the analog signal increases. If the signal decreases, it means the pressure taps on the pressure transducer are backwards and you need to flip them. Either that or just flip the sign in your plotting routine on your computer but flipping the tubes might be easier for you. Hopefully when you do this lab you will get some data that looks like this. In this

Figure you'll see that when the fan wasn't running the signal was something around 49,800 which is fine. It means your bias is around 2.5 volts. Every pitot probe and circuit will be different. You can then convert this signal to voltage then and then pressure and then finally wind speed.



The code to accomplish this is relatively simple and a portion of the code is shown below. You'll see that when I subtracted the bias from the voltage I also zeroed out any negative values. That is, any delta voltage less than zero was set to zero. A couple of things about this chart. The data from the pitot probe is super noisy which means attaching a complementary filter is probably a good idea provided you don't over filter the signal and run into aliasing issues. You can see that I implemented an offline complementary filter and plotted it in the orange line which helps the noise issue quite a bit. You'll also notice that the noise is about 2 m/s. It turns out that pitot probes are actually not very accurate lower than about 2 m/s. They would be great for an airplane or you driving down the highway but they wouldn't be very good to take wind data outside on a calm day.



2.4 Assignment

Once you've done that upload a PDF with all of the photos and text below included. My recommendation is for you to create a Word document and insert all the photos and text into the document. Then export the Word document to a PDF. For videos I suggest uploading the videos to Google Drive, turn on link sharing and include a link in your PDF.

1. If you borrowed a pitot probe return the pitot probe - Pass/Fail - If you don't return the pitot probe you receive a zero
2. Include a video of you taking data and explaining the circuit (make sure you are in the video) - 50%
3. Include a plot of the raw analog signal vs time just like I did above - 20%
4. Include a plot of windspeed vs time as I did above - 20%
5. Filter your signal using an offline complementary filter and include it in your plot like I did above - 10%