University of Illinois, Urbana Champaign

Physics 398DLP

Pressure Change Inside Residence Due to Wind-Induced Air Leakage

Charlie Xiao

Qier An

Xiye Hu

Abstract

This paper will discuss pressure changes inside a residence as an indication of wind-induced air leakage. To determine possible correlations, outdoor wind speed data and pressure changes inside the house are gathered and analyzed. Data of temperature and humidity are also considered to determine potential relationship with pressure change. Conclusion is that we didn't observe any pressure gradient with respect to wind speed. However, since we only have one sets of data with only 9 spots, we would still recommend further study into this probably with more condensed and time-accurate sampling in various weather conditions. This would probably require a lot more devices that have to be time-synchronized on data taking, and a very pleasant house owner who would allow breadboards and PCB boxes all over his place.

I. Introduction

Advisory discussion with Professor Scott Willenbrock at the early stage of this project involved the introduction of a structural tightness scale named Air Changes per Hour (ACH50). ACH50 is estimated by measuring the air change rate at the entrance of the house with a mechanical fan pumping air out of the house. The method of ACH50 yields conclusions that indicate the tightness of a residence; however, ACH50 does not include any adjustment to the effect of wind may have on the air change rate inside a house. (Meier, A. 1994.)

Inspired by the method of ACH50, we, in this experiment, focused on how wind speed outside the house can influence the inside pressure of the house. Specifically, we predicted that an increase in wind speed outside of the house correlates with a noticeable pressure difference inside the house due to air leakage. We also attempted to identify potential effects that temperature and humidity of the environment may have on the pressure inside the house.

II. Hardware

To test our hypothesis, we used Arduino Mega 2560 along with multiple sensors to build our measurement devices. Arduino Mega 2560 (Figure 1) is a single-board microcontroller, with which a portable device installed with multiple sensors for measurement can be built. The sensors used to build our devices are: BME680, Ultimate GPS, Real time clock, Anemometer.

Figure 1, Arduino Mega 2560, (image obtained from https://store.arduino. cc/usa/arduino-mega-2560-rev3)

Figure 2, BME680 (image obtained from https://www.adafruit.com/product/2652)

BME680 (Figure 2) is an environmental sensor that measures temperature, humidity, and pressure. In an ideal working environment (room temperature with 1 atmospheric pressure), the accuracy of BME680 is ±1 °C for temperature, ±0.6 hPa for pressure and ±3% r.H for humidity. In this experiment, BME680 is installed on each device and it is used for pressure, temperature, and humidity data gathering.

Figure 3,Ultimate GPS (image obtained from https://www.adafruit.com/product/746)

Ultimate GPS (Figure 3) is a GPS segment that detects and communicates with four orbital satellites to return location and time data offered by the system. In this experiment, the GPS is used for time synchronization among our devices.

Figure 4, Anemometer (image obtained from https://www.adafruit.com/product/1733)

The anemometer produces a voltage output that linearly scales with the wind speed it measures. In this experiment, the anemometer is used for measuring wind speed around the house. The accuracy of the wind speed measured is ±(0.3±0.03) meters per second.

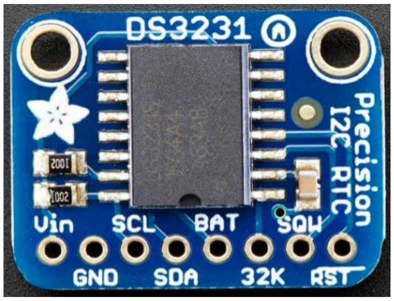


Figure 5, Real time clock (image obtained from https://www.adafruit.co m/product/3013)

DS3231 is an accurate real-time clock that maintains year, date, hour, minute, and second information. We use DS3231 to record timestamps in our data file.

We assembled Arduino Mega 2560 and the sensors discussed above onto multiple breadboards and circuit boards. We then composed the data acquisition program which controls and communicates with the sensors.

III. Data Acquisition Program

The data acquisition program written in C++ is executed through the Arduino Mega microcontroller board. The finalized version of the data acquisition program has three major parts: 1. launch and initialize sensor settings; 2. talk and control sensor measurements; 3. write and record data onto the micro-SD card.

Initialization part of the data acquisition program sets the operating parameters of all the sensors that will be used in the measurements. This part of the program is executed as soon as the Arduino board is powered. A piece of pseudo code that prepares a specific sensor is provided as follow:

```
// Specify the connected pin number on the Arduino board
const int sensor_pin = ##;

// Launch the sensor
sensor.begin(sensor_pin);

// Set operating parameters.
sensor set Setting 1;
sensor set Setting 2;
…
```

Sample codes that initialize default settings were provided by Adafruit, company from which we bought sensors used in this experiment. Some modifications to these default parameters were made to accommodate special requirements. For example, we disabled BME680 environmental sensor's gas heater and modified its sampling rate to increase overall measurement sampling rate per second.

Second part of the data acquisition program controls the overall measurements performed by various sensors. The execution and termination of this part of the program is controlled by user inputs from keypad. Some notable functionality provided are time synchronization between GPS and the real time clock, sensor measurement communication, and control of measuring rate. A piece of pseudo code that asks a specific sensor to perform measurement is provided as follow:

```
// Exception handling
IF sensor_not_working:
       restart after problem fixed
END IF

Asks the sensor to perform measurement

Save recorded data.
```

Lastly, after measurement takes place, the program saves all the data and writes them, in a line by line format, to the micro-SD card connected to the Arduino board. The data file is then analyzed offline which we will talk about later.

```
Data written below has such format:
hour(UTC), minute, second, millisecond
Temperature(°C), pressure(hPa), humidity(%)
Wind speed voltage (0.4V ~ 2V)

20,22,23,0
19.40,996.12,45.67
1.12
```

Challenges were encountered with sampling rate and time synchronization when during the debugging phase of this program. Since the measuring rate was decided to be every tenth of a second, every cycle of measurement needs to be completed within one-hundred milliseconds and the internal time count needs to be accurate within a millisecond. Two methods were proposed to maintain desired time accuracy, one relies on the GPS chip and the other relies on DS3231(real time clock). However, neither were adapted because GPS calls were highly time-consuming, and the real time clock was only accurate to a second. At the end, the time accuracy was achieved by utilizing the internal time count of Arduino board, which measures the time since the Arduino microcontroller has started in milliseconds. Time information was obtained by synchronizing the real time clock with GPS once at the start of the program. A piece of pseudo code to explain the process is provided as follow:

```
Synchronize RTC and GPS
SET timestamp1
Do the measurement
WRITE RTC hour, minute, second (Synchronized to GPS at the start)
WRITE Arduino time (millisecond)

WHILE current_time - timestamp1 < 100ms
      do nothing and wait
END WHILE
```

A minor systematic inaccuracy regarding measurement time accuracy is produced due to BME680, the environmental sensor used for temperature, pressure, and humidity measurement. BME680 has an integrated micro chip that processes information obtained from the sensors, when asked to perform a reading, BME680 will first read data from its sensors, then process and parse the data into readable numbers, and then send this data to the Arduino board. Because of this, rather than respond to the reading calls instantaneously, BME680 will respond with a slight delay of around forty milliseconds. Therefore, the time stamp written to the data file, which was taken as soon as BME680 responds, does not represent the exact time that the measurement took place. However, the exact time information of measurement is inaccessible because the BME680 cannot respond before it finishes processing its data. With further experimentation involving more intricate methodology, this measurement inaccuracy may be measured and adjusted. Due to resource and time limitations, we chose not to do so in this experiment. Nonetheless, because this error is present and consistent in all of our measurements, we believe its influence on our analysis is minimal.

A software problem with the data acquisition program was encountered during the data taking process where SD card file corruption was found in numerous devices due to insufficient exception handling. When a device loses power or crashes due to unforeseen circumstances, the file in the SD card was not closed properly which causes data to be buried inside SD card's "invisible" storage. Although the corrupted data can be recovered through data recovery software, improvements to file exception handling should be made to protect data file from being lost again.

IV. Data Taking

The data taking took place on December 2, 2018 at a residential house in Champaign, Illinois. Nine devices (labeled No. 1 to No. 9) were used in the measurement. A map of device locations is shown in figure 6. Two devices attached with anemometer were placed approximately 15 meters outside of the house for windspeed measurement. Device No.1 was placed on a ladder with an elevation of 1.8 meters above the ground; device No.2 was placed on a ladder with an elevation of 1.5 meters above the ground. The seven other seven devices were placed inside the house. Device No.3 was placed beside the front door on the floor; device No.4 was placed on the floor in a junction that connects the kitchen, living room, and the stairs; device No.5 was placed in the living room on a table 0.5 meters above the floor near the window; device No.6 was placed in the kitchen on a table 1 meter above the floor near the door to the backyard; device No.7 was placed on the table of dining room, 1 meter from the floor; device No.8 was placed beside the window of the dining room, lifted 0.6 meters above the floor; device No.9 was placed on a piano opposite to the dining room, with an elevation of 1.4 meters above the floor.

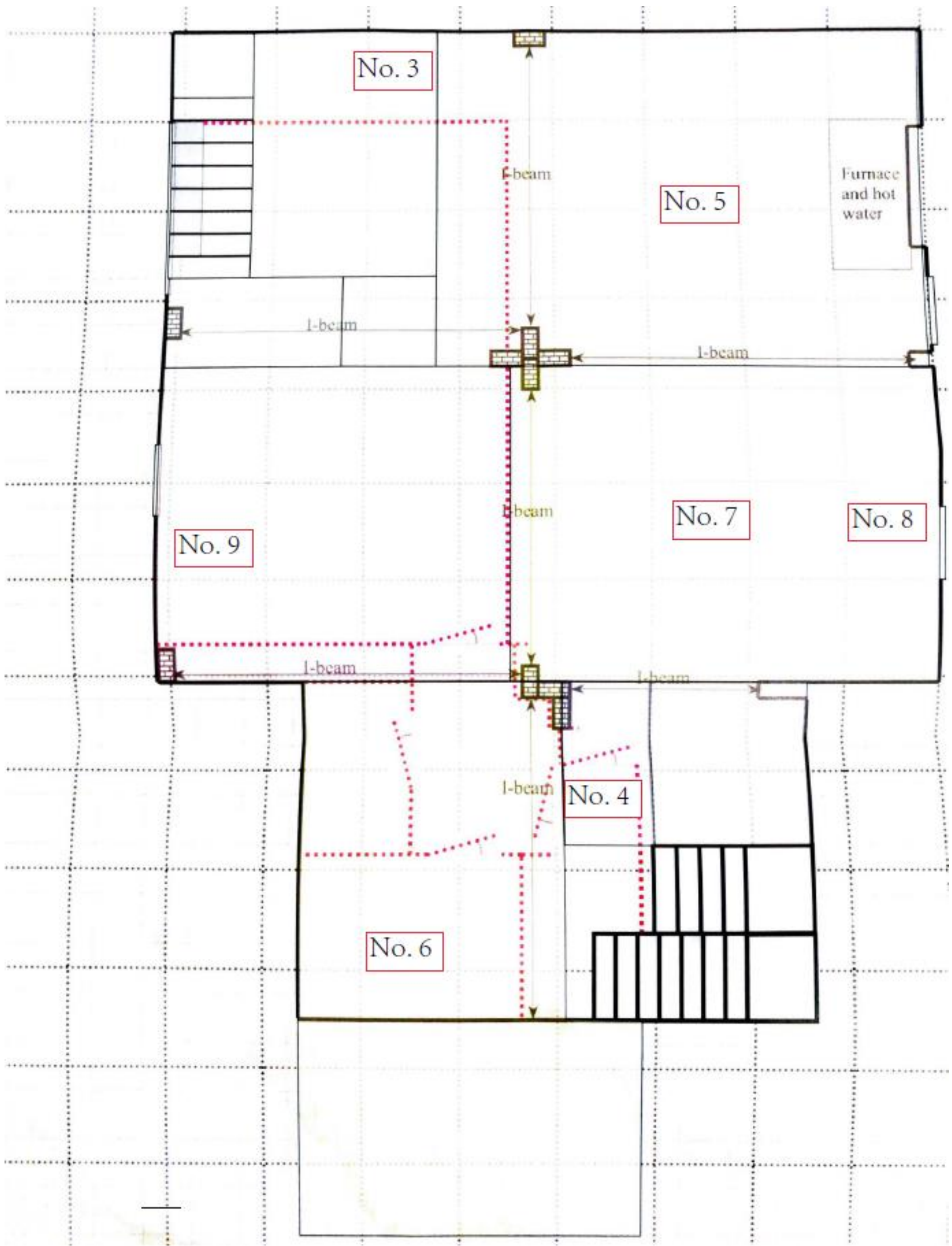


Figure 6, Device location with the layout of the house, each grid is 4 by 4 ft.

The measurement began at approximately 11:00 a.m. and ended at approximately 12:30 p.m. (90 minutes of measurement). Sampling rate is set to 10 Hz (10 measurements per second). Information of the weather at the time of measurement is provided below:

Time	Temperature	Dew Point	Humidity	Wind	Wind Speed	Wind Gust	Pressure
12:53 AM	48 ° F	45 ° F	89 %	SSE	9 mph	0 mph	28.6 in
1:53 AM	48 ° F	44 ° F	86 %	SSE	10 mph	0 mph	28.6 in
2:53 AM	48 ° F	43 ° F	83 %	SSE	17 mph	0 mph	28.6 in
3:53 AM	49 ° F	43 ° F	80 %	S	21 mph	0 mph	28.5 in
4:53 AM	48 ° F	43 ° F	83 %	S	15 mph	0 mph	28.5 in
5:53 AM	48 ° F	43 ° F	83 %	SW	13 mph	0 mph	28.6 in
6:53 AM	44 ° F	37 ° F	76 %	WSW	17 mph	0 mph	28.6 in
7:53 AM	43 ° F	38 ° F	82 %	WSW	12 mph	0 mph	28.6 in
8:53 AM	44 ° F	39 ° F	82 %	SW	15 mph	0 mph	28.6 in
9:07 AM	44 ° F	39 ° F	82 %	WSW	16 mph	0 mph	28.6 in
9:53 AM	43 ° F	38 ° F	82 %	SW	20 mph	0 mph	28.6 in
10:53 AM	43 ° F	36 ° F	76 %	SW	22 mph	0 mph	28.7 in
11:25 AM	41 ° F	38 ° F	89 %	SW	23 mph	30 mph	28.7 in
11:53 AM	42 ° F	38 ° F	85 %	WSW	17 mph	26 mph	28.7 in
12:53 PM	41 ° F	37 ° F	86 %	SW	17 mph	30 mph	28.7 in

Figure 7, weather data from morning to noon on Dec. 2, 2018.

V. Offline data analysis

The data analysis program is a script written in Python which reads the data file, shows plots of values and does linear fitting on graphs, since we don't know if there is any correlation between wind speed and environmental status.

These functions are achieved through numpy, pandas scipy.stats and matplotlib. The data is read from location using pandas.read_csv(). It is stored in the dataframe format, but is then re-saved as array using data.values.

```
import numpy as np
import pandas as pd
import scipy.stats
import matplotlib.pyplot as plt

file_name = 'test_11_7_inside_dooropen_1.txt'

data = pd.read_csv("C:\\Users\\14625\\OneDrive\\Documents\\GitHub\\DataAcquire\\byThePlants3.txt", delimiter = ',', s
data_2 = pd.read_csv("C:\\Users\\14625\\OneDrive\\Documents\\GitHub\\DataAcquire\\byThePlants3.txt", delimiter = ',',
data_3 = pd.read_csv("C:\\Users\\14625\\OneDrive\\Documents\\GitHub\\DataAcquire\\" + file_name, delimiter = ',', ski
data = data.values
data_2 = data_2.values
data_3 = data_3.values
```

Only one single function from scipy.stats library is used to do the fitting.

```
x = pressure_3
y = temperature_3


slope, intercept, r_value, p_value, std_err = scipy.stats.linregress(x, y)
```

This function is simple. It accepts my data in array form, does a linear regression and return everything needed to plot the regression and obtain correlation. (slope, intercept, standard error, R^2)

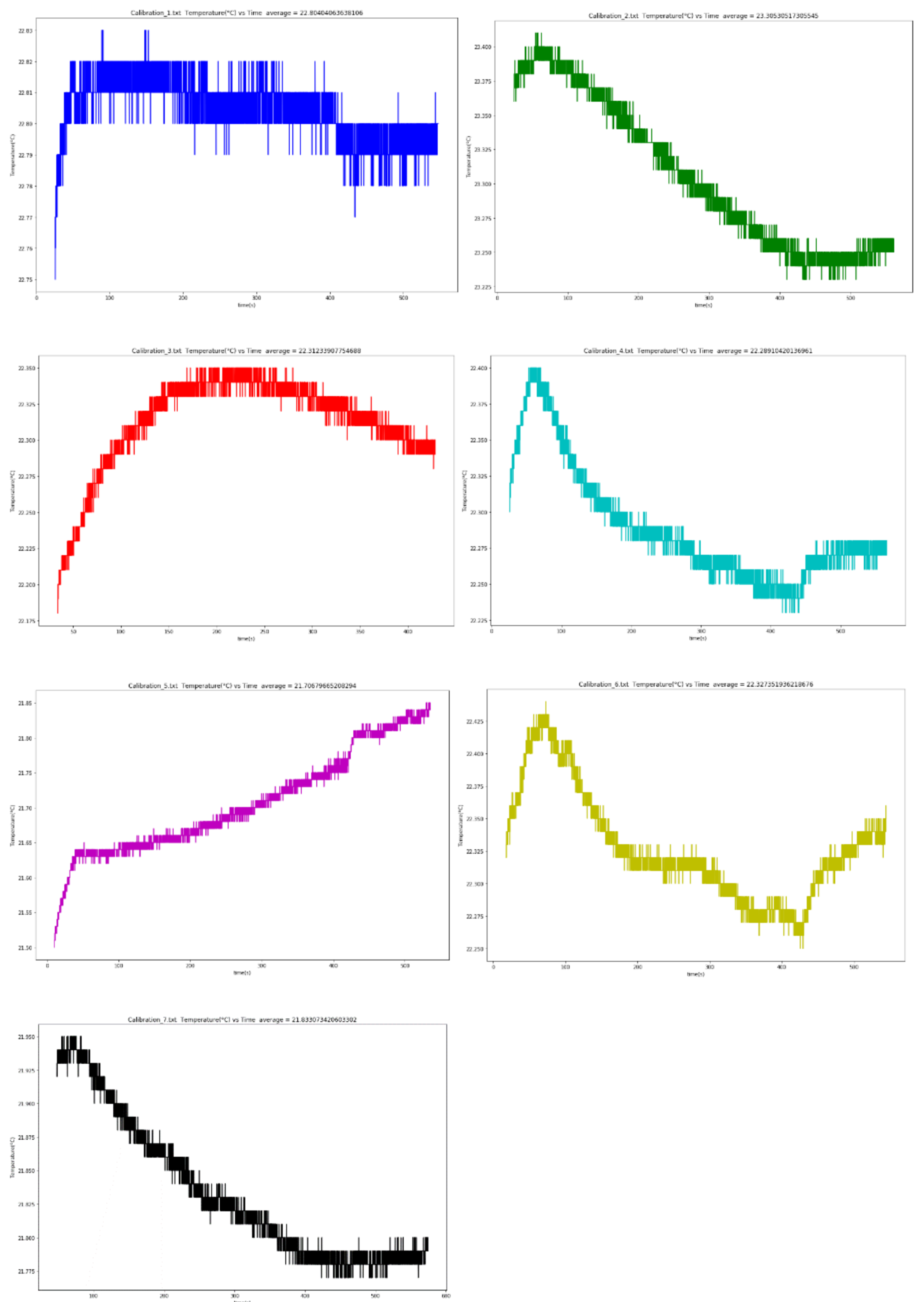
More Info from: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.linregress.html

VI. Calibration

Another successful set of data are taken at Professor Gollin's house and we placed 9 devices in different places. In this trial there are sufficient devices and data but first, calibration was done for 7 out of 9 devices.

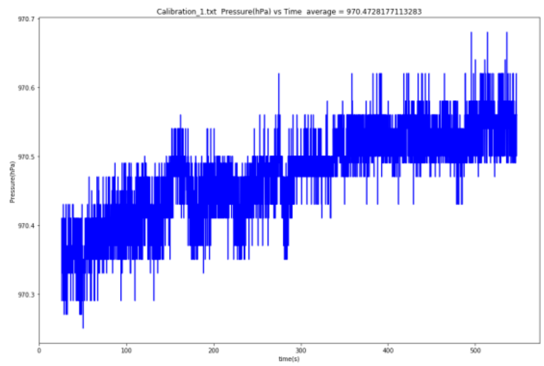
We let every device run for about 10 minutes and see how their data value varies and make adjustment to their value discrepancies.

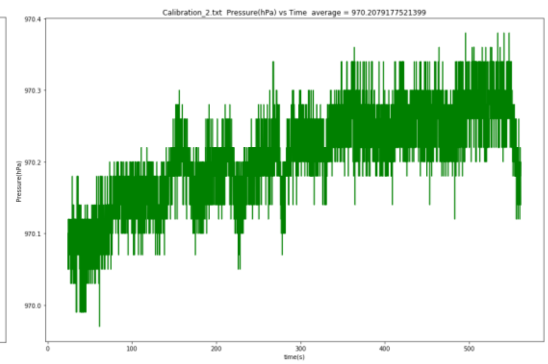
This is the calibration of temperature for 7 devices. Temperature varies in range of around 0.2 degree Celsius for all devices, indicating a pretty stable run, though the value differs between devices.

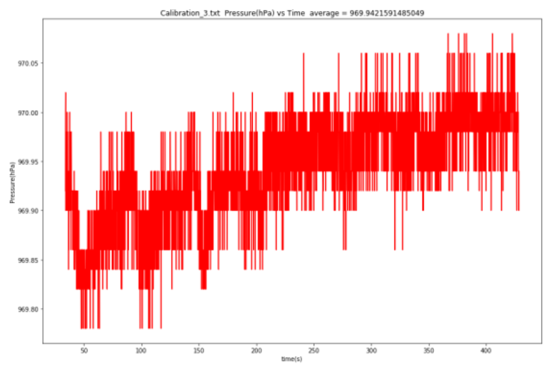


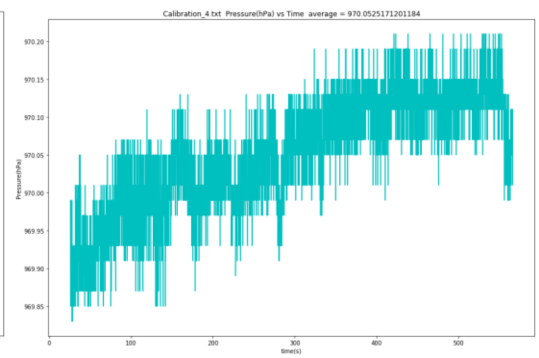
The pressure calibration for 7 devices. All devices show graphs of very similar shape.

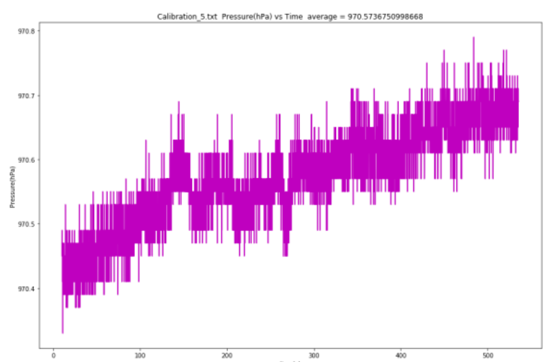
Also none of their value varies over 0.4, which is very small compared with typical 970hPa.

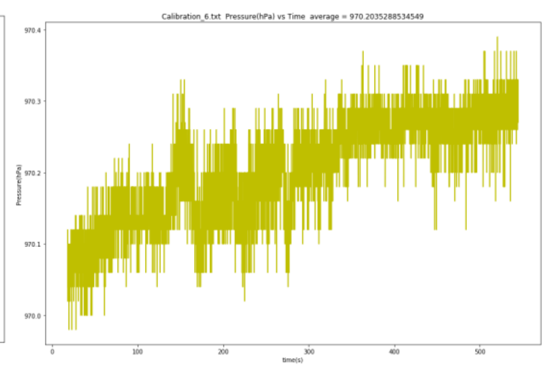


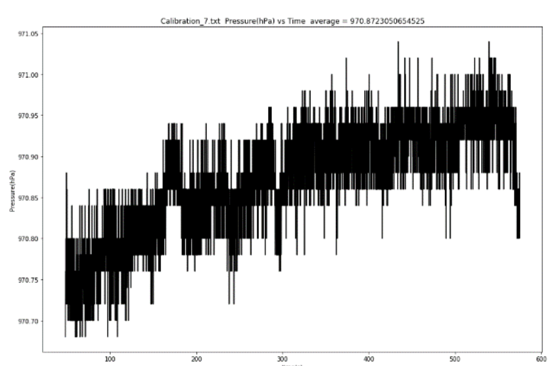




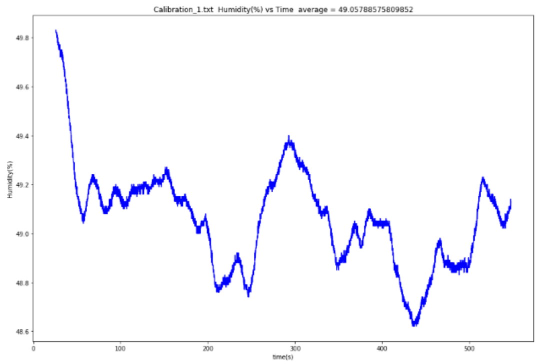


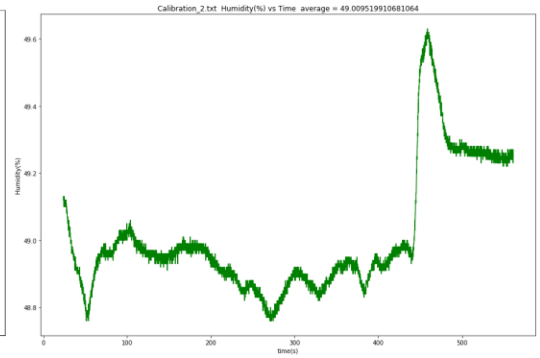


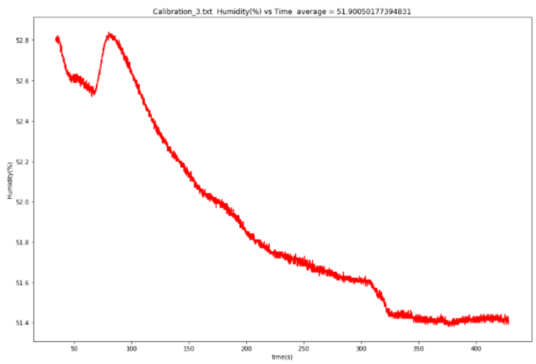


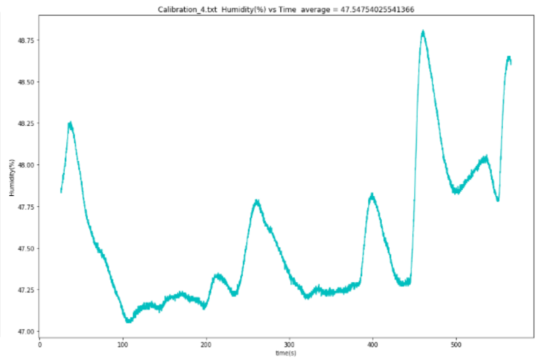


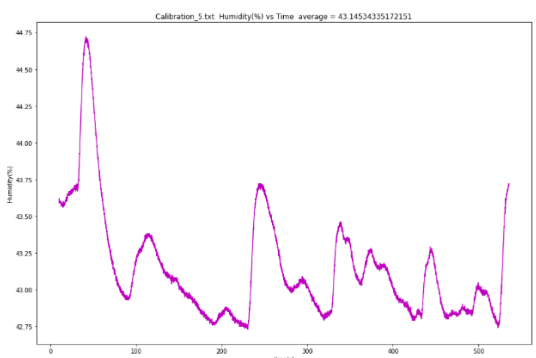
Below is the calibration of humidity. It can vary up to 2% over the test period. However, none of the devices shows an obvious sign of settling except for No.3(red line).

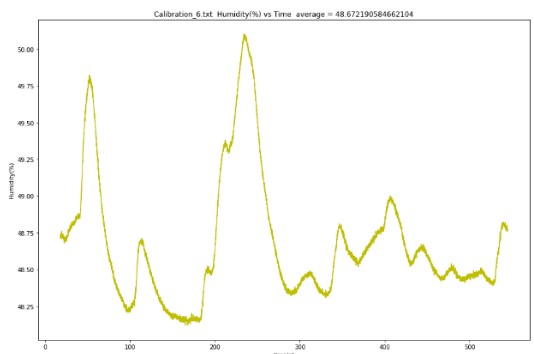

Calibration_1.txt Humidity(%) vs Time average = 49.05788575809852
Humidity(%)
time(s)

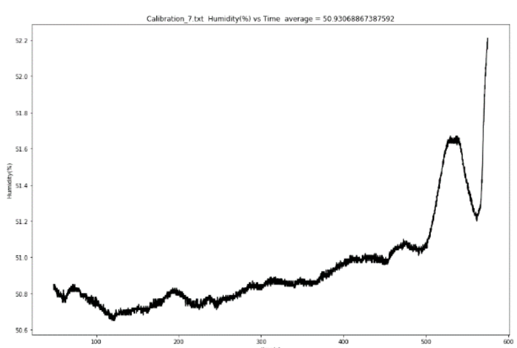

Calibration_2.txt Humidity(%) vs Time average = 49.009519910681064
Humidity(%)
time(s)


Calibration_3.txt Humidity(%) vs Time average = 51.90050177394831
Humidity(%)
time(s)

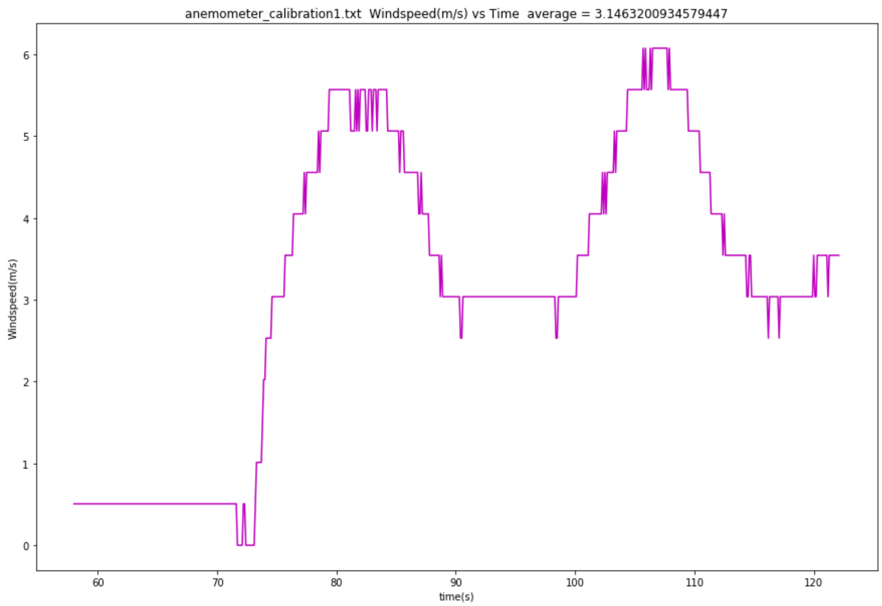

Calibration_4.txt Humidity(%) vs Time average = 47.54754025541366
Humidity(%)
time(s)

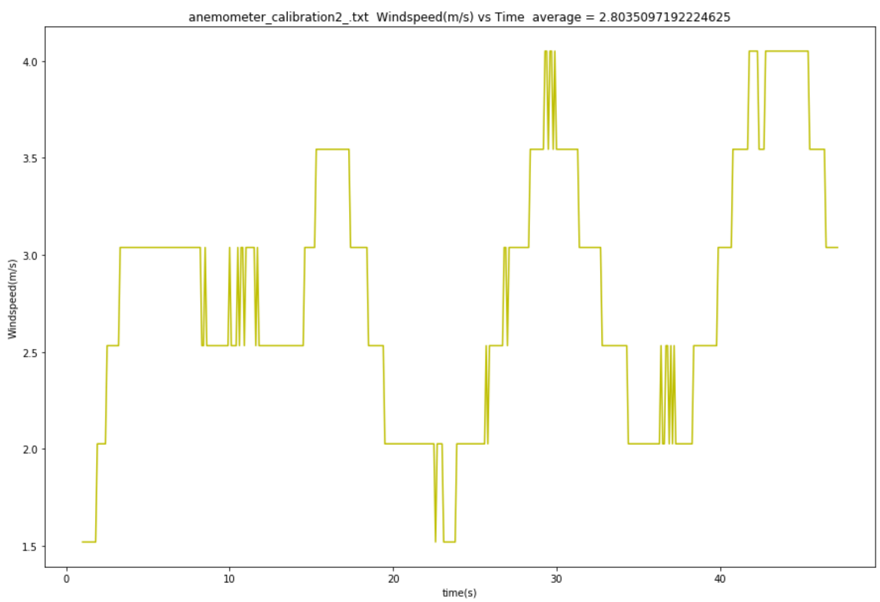

Calibration_5.txt Humidity(%) vs Time average = 43.14534335172151
Humidity(%)
time(s)


Calibration_6.txt Humidity(%) vs Time average = 48.672190584662104
Humidity(%)
time(s)


Calibration_7.txt Humidity(%) vs Time average = 50.93068867387592
Humidity(%)
time(s)

The calibration of windspeed is done by holding 2 anemometers out of the window of a car driving at 10mph. No.2(yellow line) starts taking data a bit earlier so the first 25 seconds are ignored while taking time average.





Therefore, the calibration is done by:

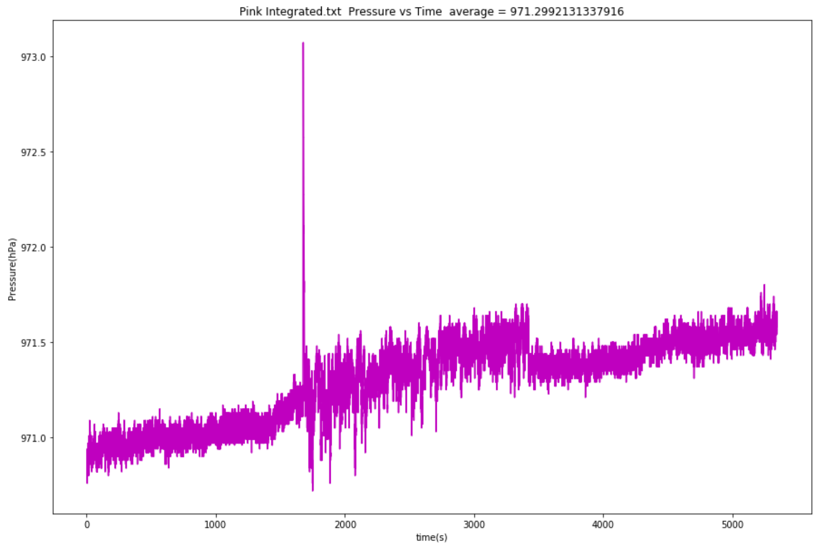
1. Calculating the average of the test run data from the 7 available devices put next to each other.
2. Getting the deviation from the average for each device data.

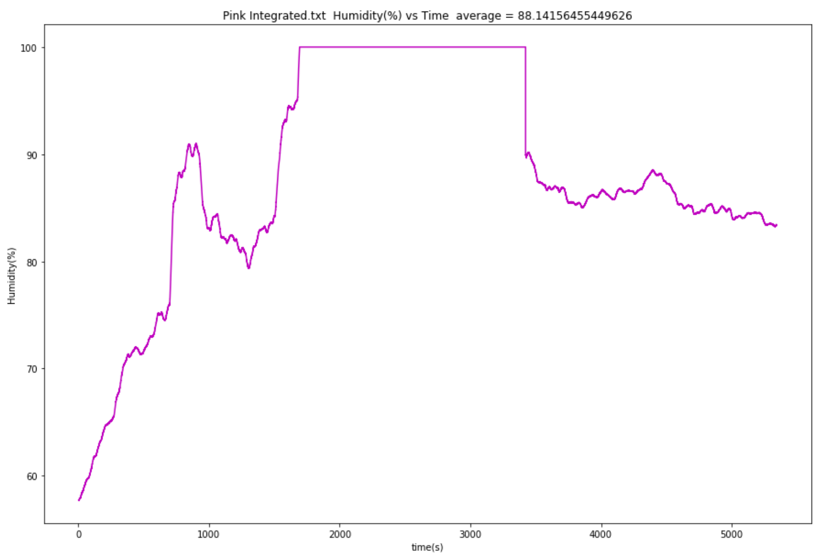
3. Using the deviation value to adjust the averaged data from the 2hr test.

Device	Type	Location	Calibration Temp(Celcius)	Deviation from Avg	Calibration Pressure(hPa)	Deviation from Avg	Calibration Humidity(%)	Deviation from Avg	Calibration Windspeed(m/s)	Deviation from Avg
No.1	Pink Box	Outside West	22.80404064	0.435753337	970.4728177	0.140686175	49.05788576	0.520290253	3.146320093	0.171405187
No.2	Yellow Box	Outside East	23.30530517	0.937017873	970.2079178	-0.124213784	49.00951991	0.471924406	2.803509719	-0.171405187
No.3	White Box	Front Door	22.31233908	-0.055948222	969.9421591	-0.389972387	51.4	2.862404495		
No.4	Bred	Triple Doors	22.2891042	-0.079183098	970.0525171	-0.279614416	47.54754026	-0.99005525		
No.5	Bred	Living Room	21.70679665	-0.661490648	970.5736751	0.241543564	43.14534335	-5.39225215		
No.6	Bred	Kitchen	22.32735194	-0.040935363	970.2035289	-0.128602682	48.67219058	0.13459508		
No.7	Bred	Dining Room	21.83307342	-0.535213879	970.8723051	0.54017353	50.93068867	2.393093169		
No.8	Red Box	Piano	N/A		N/A		N/A			
No.9	Transluscent Box	Plants	N/A		N/A		N/A			
Avg			22.3682873		970.3321315		48.5375955		2.974914906	

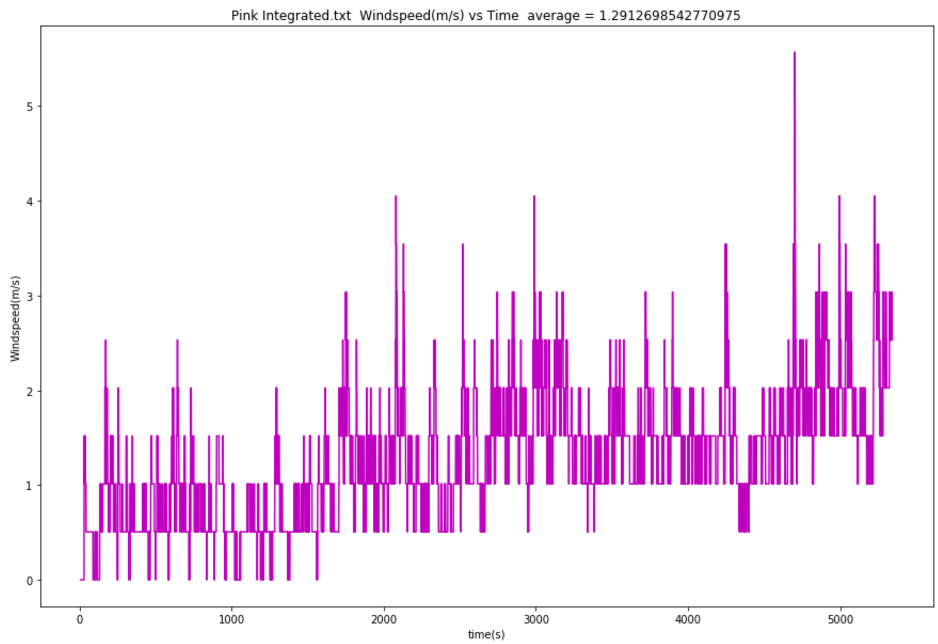
VII. Analysis

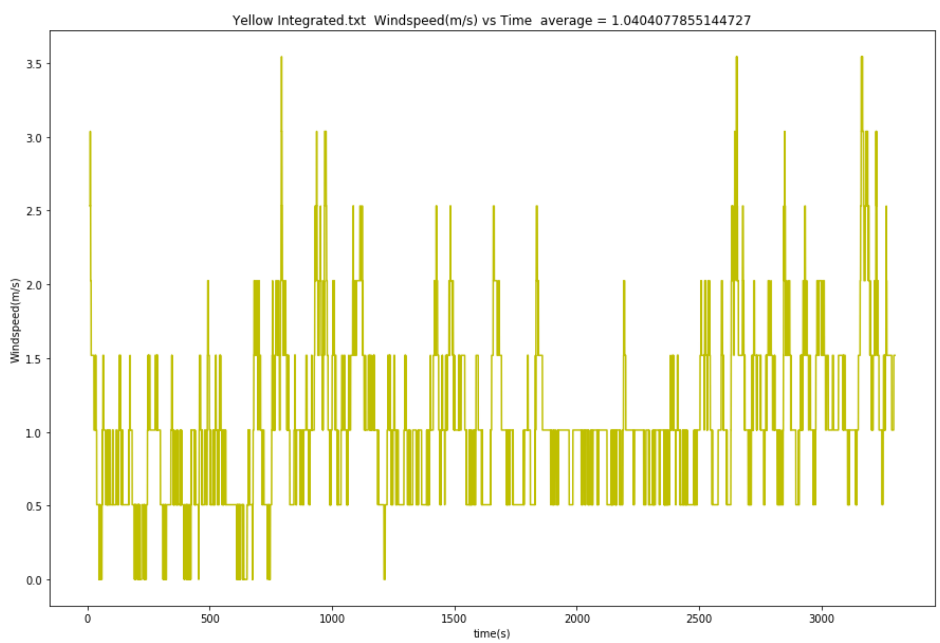
There are only two devices, No.1 and No.2, that measure wind speed. Due to the concern of avoiding losing data, the data taking of both devices are separated into two segments. Device No.2 had an issue at the first data taking part so it doesn’t include the data for the first half hour. Although no data were lost, device No. 1 has a peak on pressure and an increase on humidity as shown below. It might because there was a raindrop onto the BME during data taking.





We have the wind speed value over time. In this case wind speed is calculated by the formula 'Wind speed = (Voltage - 0.4) / 1.6 * 32.4' and the functioning voltage of the anemometer is supposed to be over 0.4. The pink graph shows the value of device No.1 and the yellow shows that of No.2.

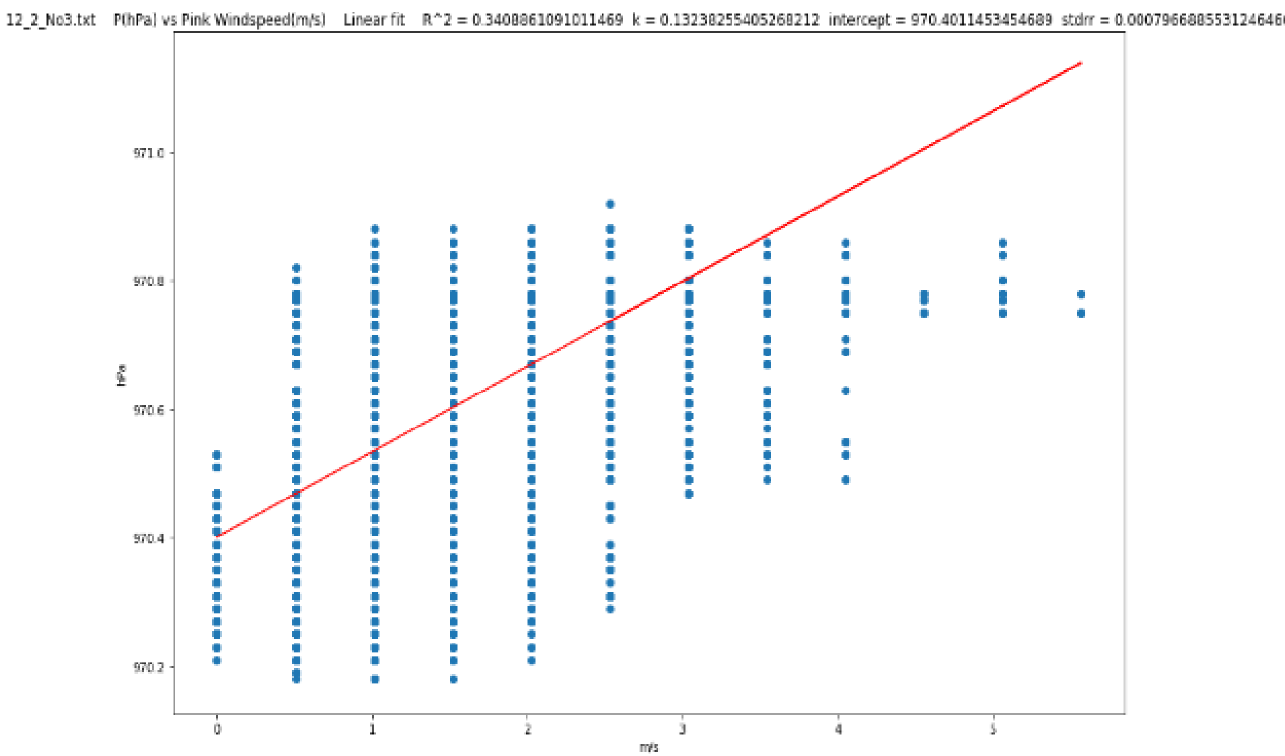


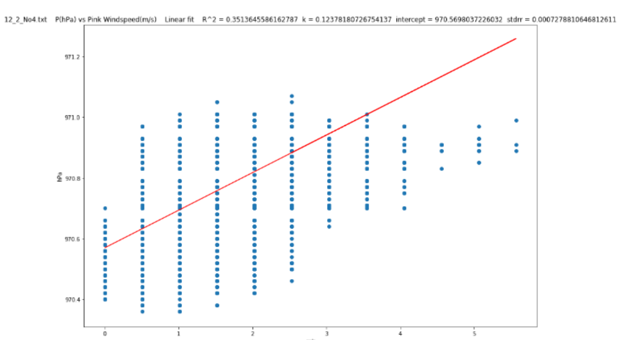


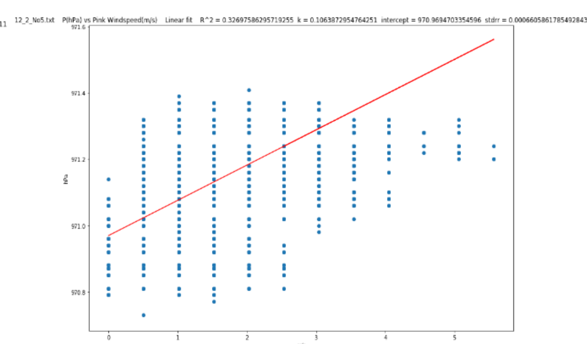
From my personal experience it seems to be colder and have more wind blowing at the position of device No.3, No.4 and No.6. According to Professor Gollin it is due to the poor heating setup but they might act as examples of a not well-sealed place.

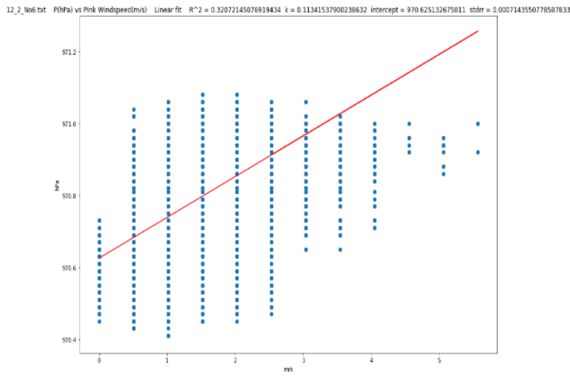
Since the major part of our measurement is about pressure vs windspeed, we plotted the graphs of pressure vs wind speed for both devices and tried a linear fit on them

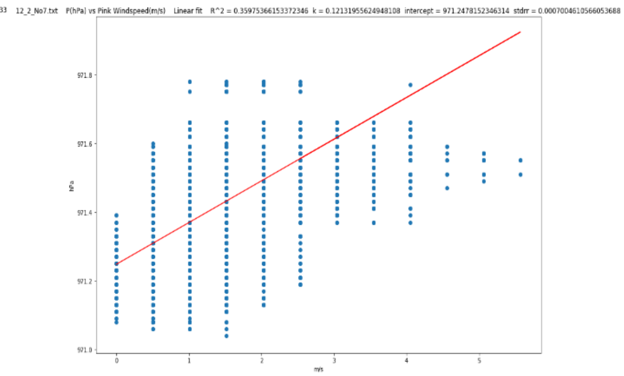
We mainly focus on indoor pressure so here are how they distribute with wind speed recorded on device No.1.

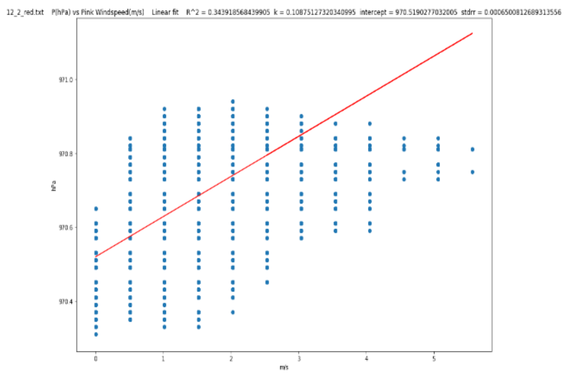


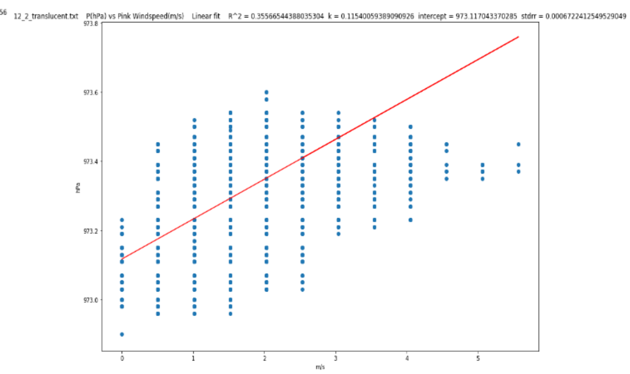






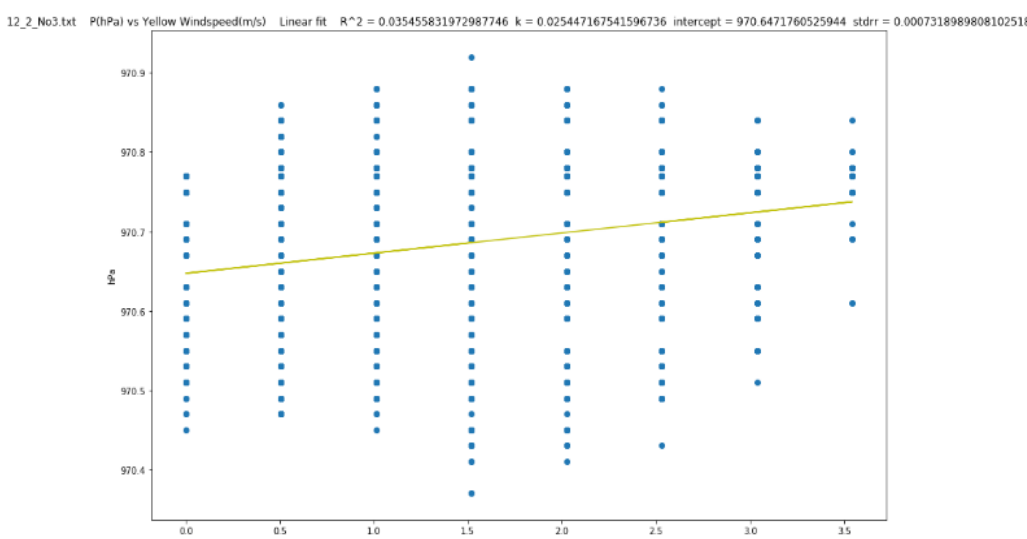


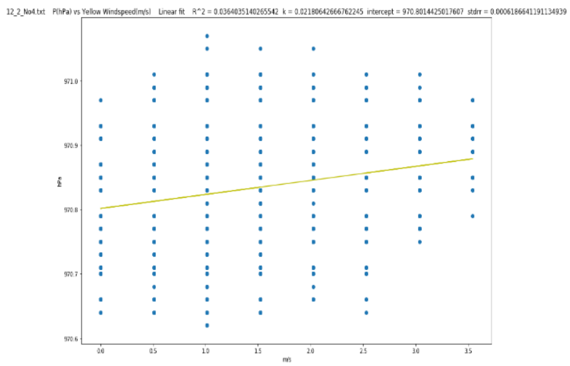


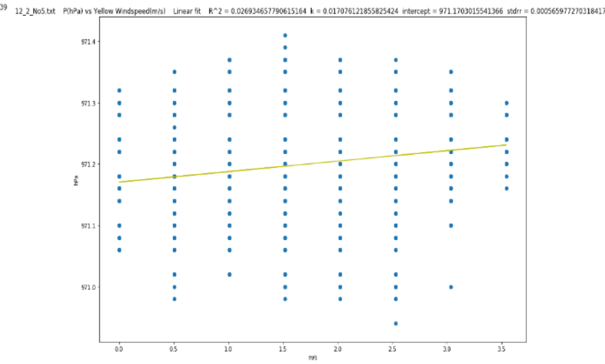


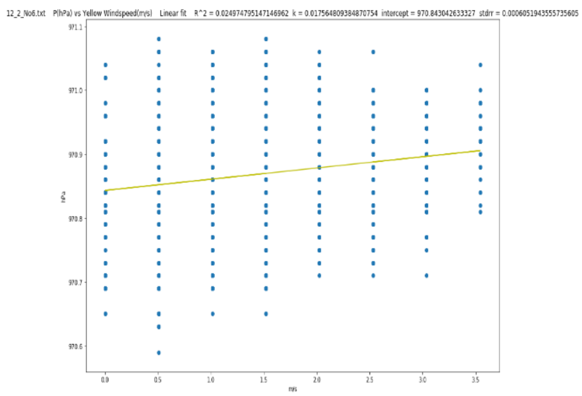
No.3 is a good example of how all sets of data looks like. Python does give a very similar results for all of them, but the linear fit doesn't seem to fit distribution quite well. R^2 is 0.34 for device No.3, which is hardly a well-correlated relationship.

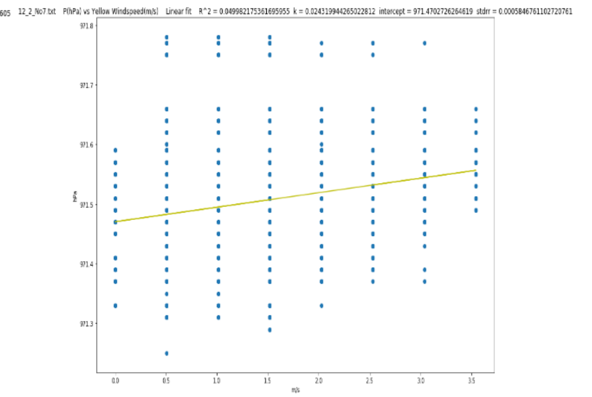
We move on to look at indoor pressure with wind speed recorded on device No.2.

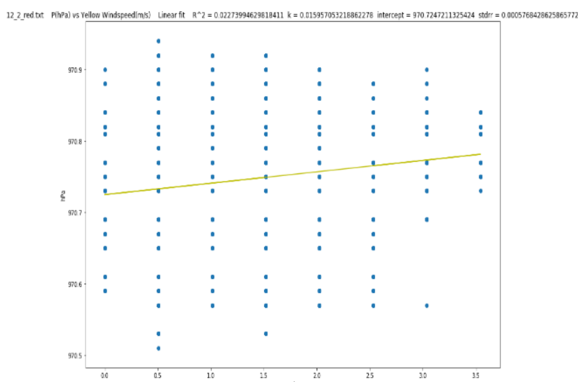


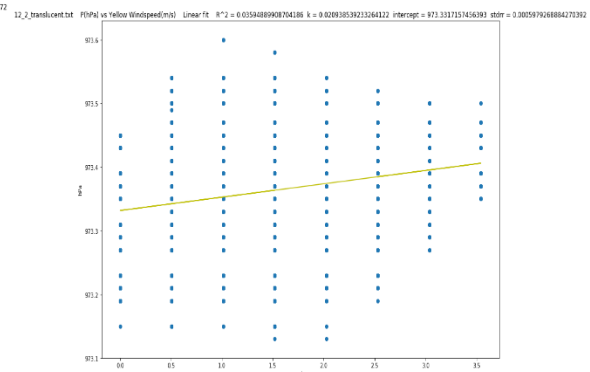






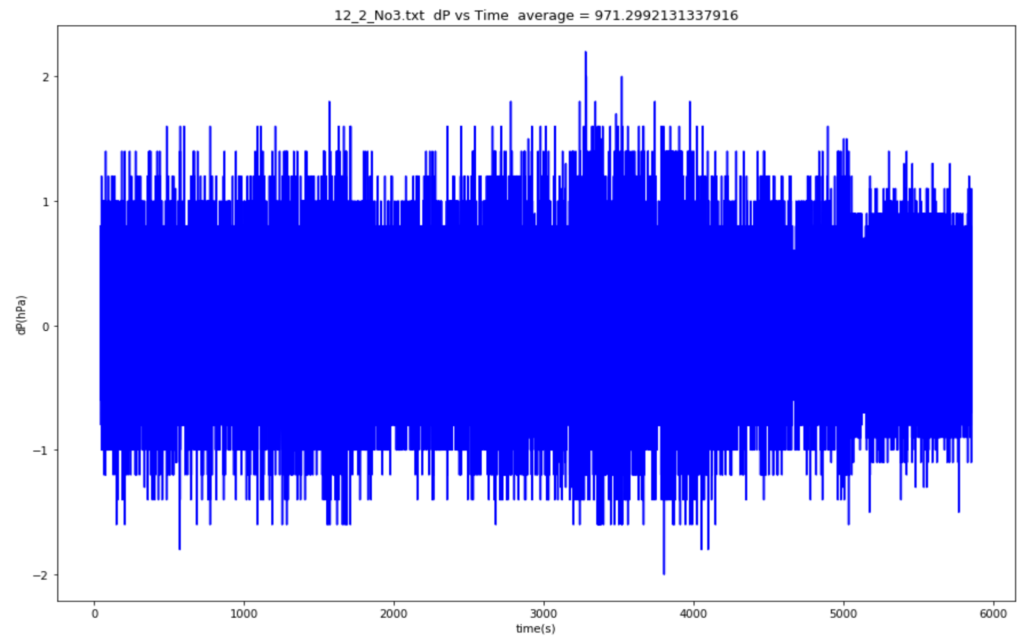


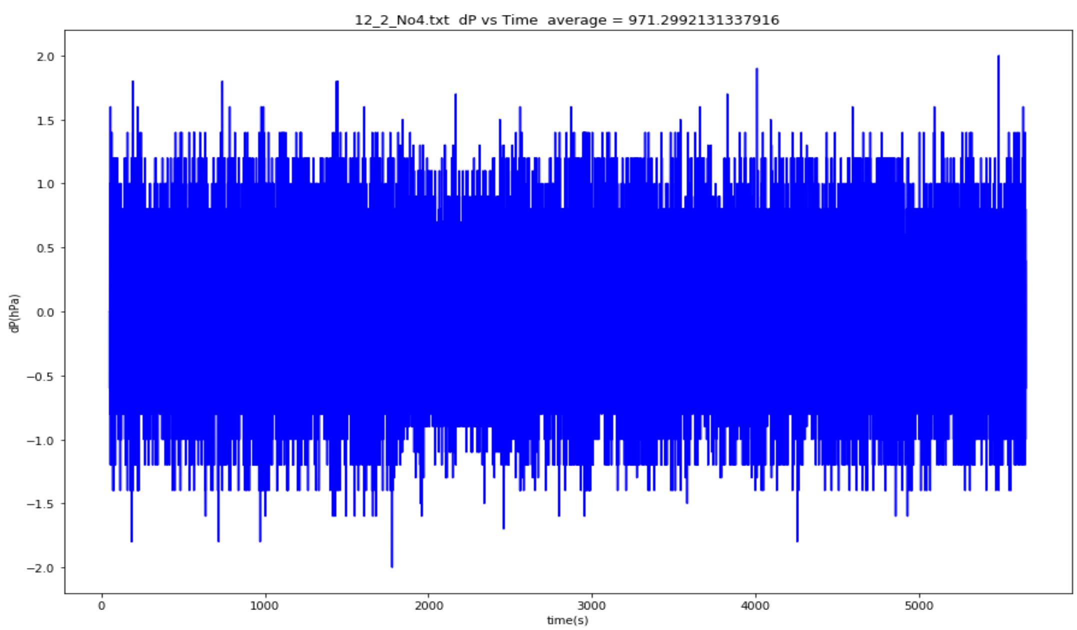


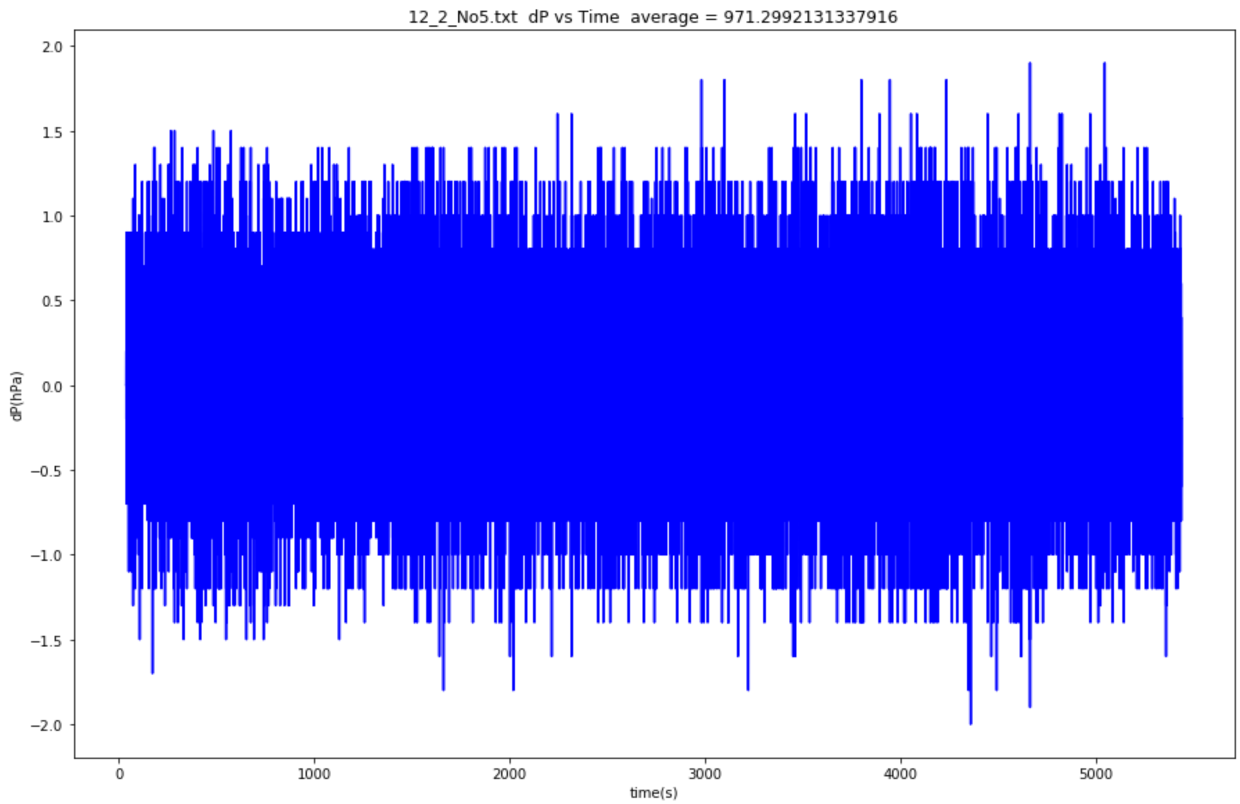


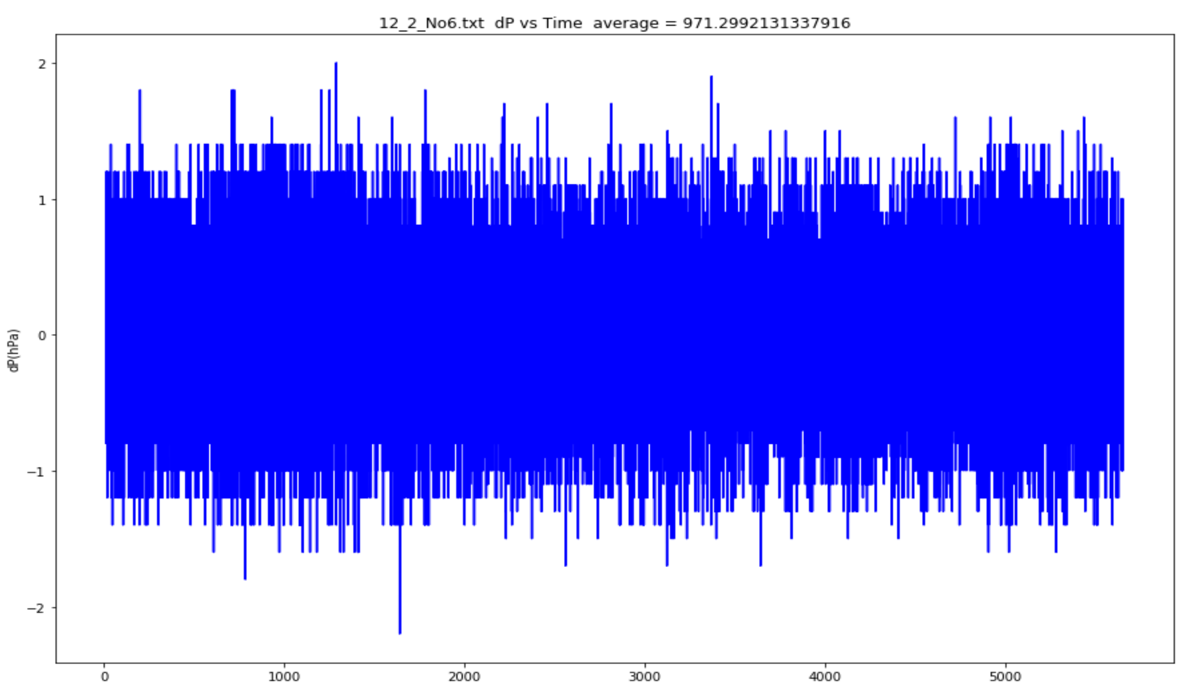
Again, all devices share a same pattern. However, the correlation seems to get even more vague with this set of wind speed. The pressure are distributed quite evenly and the variations are just about 0.5hPa which is really small. Due to the raindrop happening on device No.1, we suspect this is the more general distribution of pressure vs wind speed. With R^2 = 0.035. We don't think there's an observable pressure change caused by windspeed.

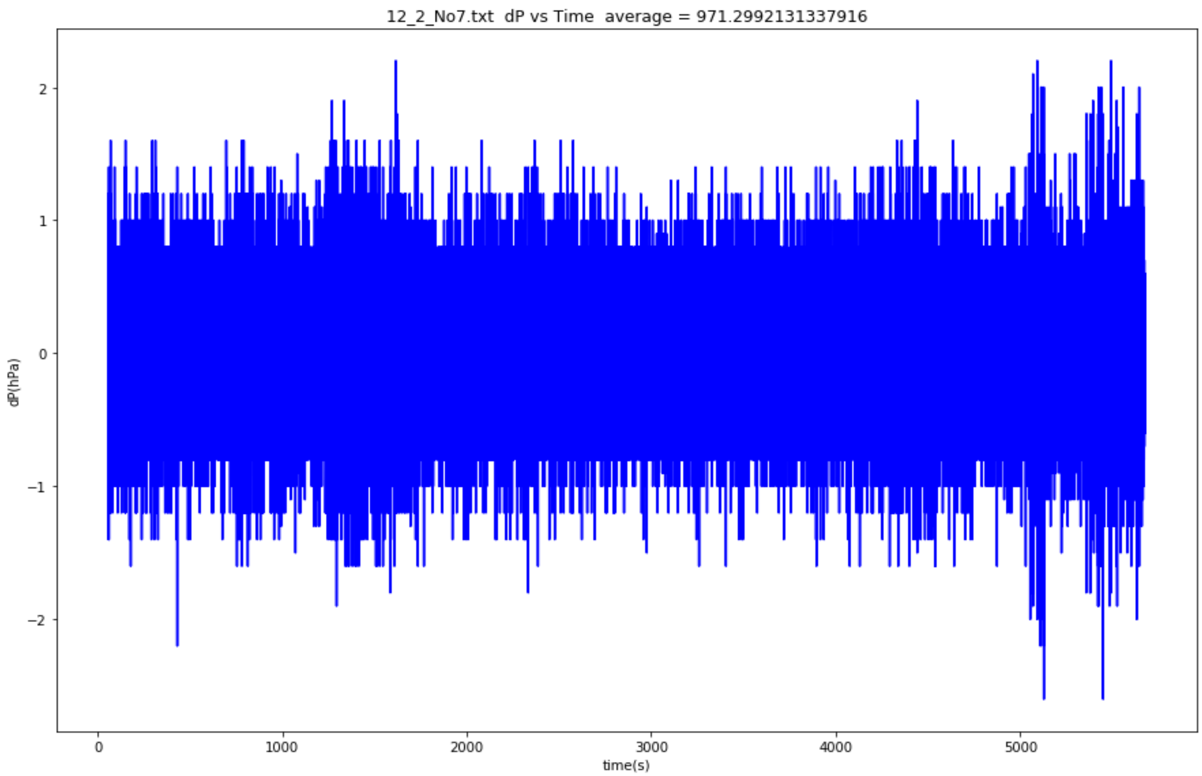
We also looked at the dP/dt, how pressure changes with time. I think this is very important and probably needs a comparison of all devices.


12_2_No3.txt dP vs Time average = 971.2992131337916
dP(hPa)
time(s)
2
1
0
−1
−2
0
1000
2000
3000
4000
5000
6000

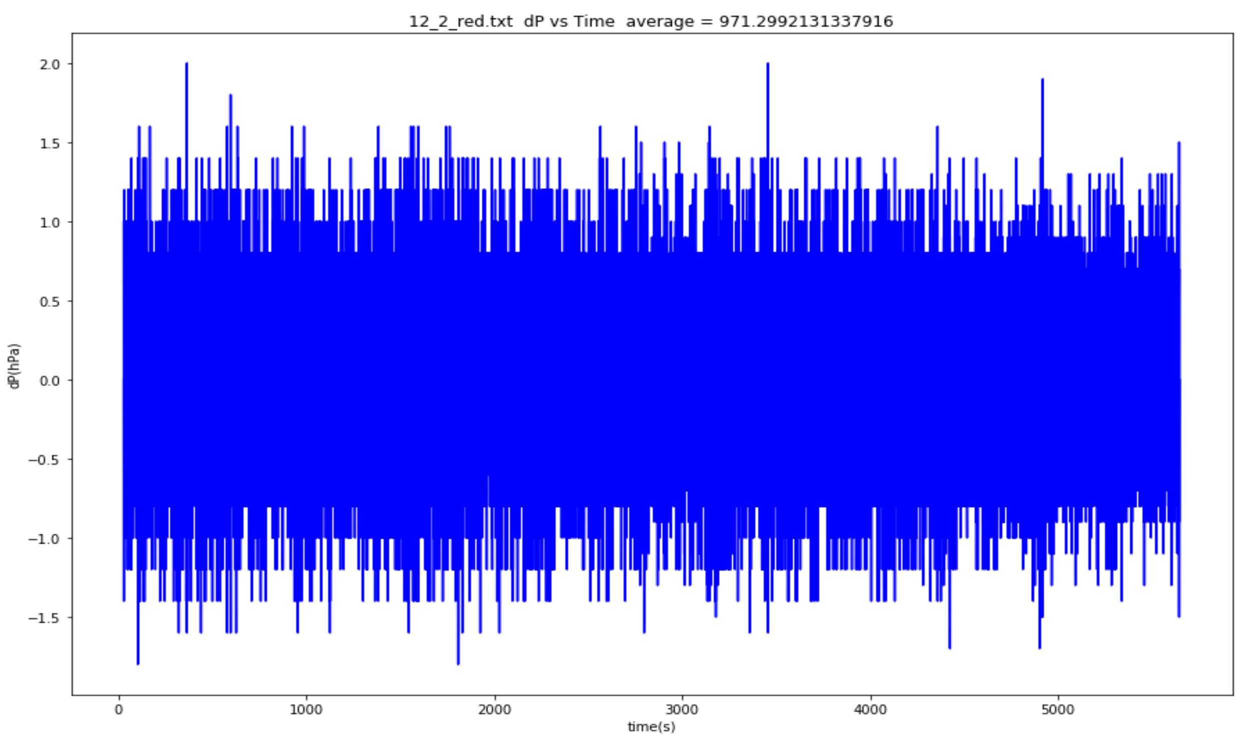

12_2_No4.txt dP vs Time average = 971.2992131337916
dP(hPa)
time(s)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0
1000
2000
3000
4000
5000


12_2_No5.txt dP vs Time average = 971.2992131337916
dP(hPa)
time(s)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0
1000
2000
3000
4000
5000

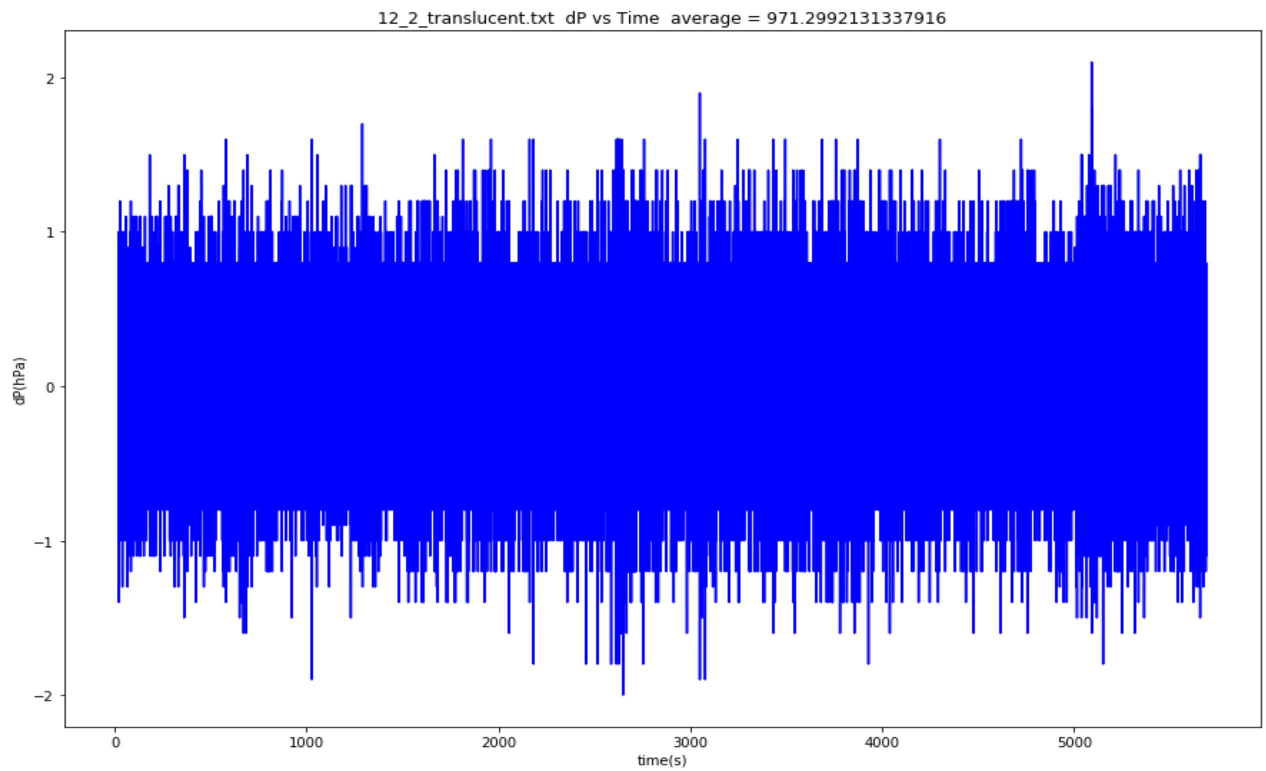

12_2_No6.txt dP vs Time average = 971.2992131337916
dP(hPa)
time(s)
2
1
0
−1
−2
0
1000
2000
3000
4000
5000



12_2_No7.txt dP vs Time average = 971.2992131337916
dP(hPa)
time(s)



12_2_red.txt dP vs Time average = 971.2992131337916
dP(hPa)
time(s)



There aren't any obvious difference on how much the dP/dt changes depending on the location. Only device No.7 seems to have a slightly bigger noise compared with others, but it is on the table of the dining room, which isn't as windy as the kitchen or triple doors. I assume the relatively bigger variations come from people walking pass it, since it is a crucial path to the back of the house, and I did remember our group members and professor Gollin walking through it a few times during data taking.

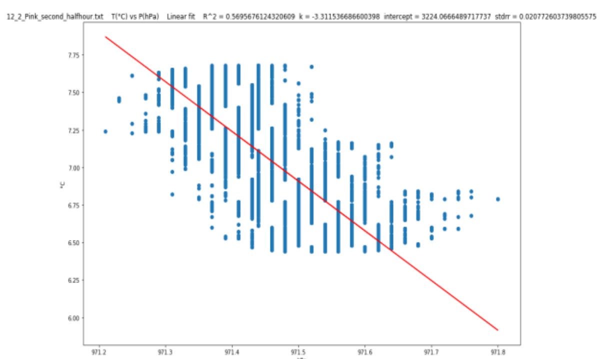
We don't find any observable correlation between pressure and wind speed. The correlation might exist, but our data lacks both accuracy and universality. Further study is recommended, provided an ability to take sufficient data.

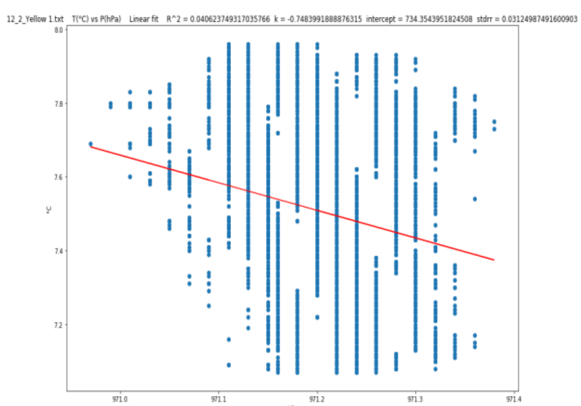
Still, we have time-averaged data for every device, shown below:

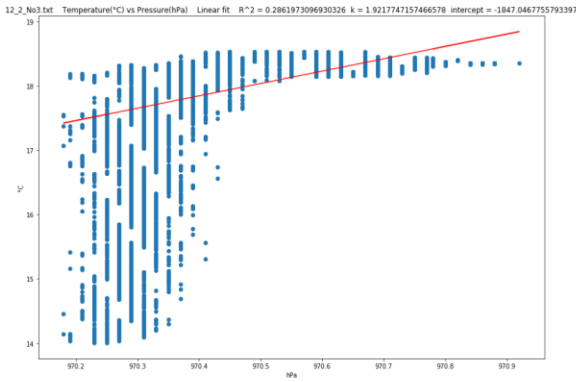
Device	Type	Location	Average Temperature	after correction	Average Pressure	after correction	Average Humidity	after correction	Average Wind	after correction
No.1	Pink Box	Outside West	7.4675	7.031746663	970.838	970.6973138	85.912	85.39170975	1.291269854	1.119864667
No.2	Yellow Box	Outside East	7.5715	6.634482127	971.238	971.3622138	85.94	85.46807559	1.040407786	1.211812973
No.3	White Box	Front Door	18.1307	18.18664822	970.55	970.9399724	57.602	54.7395955	N/A	
No.4	Bred	Triple Doors	18.974	19.0531831	970.719	970.9986144	51.385	52.37505525	N/A	
No.5	Bred	Living Room	22.1667	22.82819065	971.105	970.8634564	40.4942	45.88645215	N/A	
No.6	Bred	Kitchen	19.3713	19.41223536	970.76	970.8886027	52.2572	52.12260492	N/A	
No.7	Bred	Dining Room	21.3297	21.86491388	971.395	970.8548265	50.1	47.70690683	N/A	
No.8	Transluscent Box	Plants	21.006	21.006	973.254	973.254	50.6359	50.6359	N/A	
No.9	Red Box	Piano	22.1832	22.1832	970.649	970.649	37.7941	37.7941	N/A	

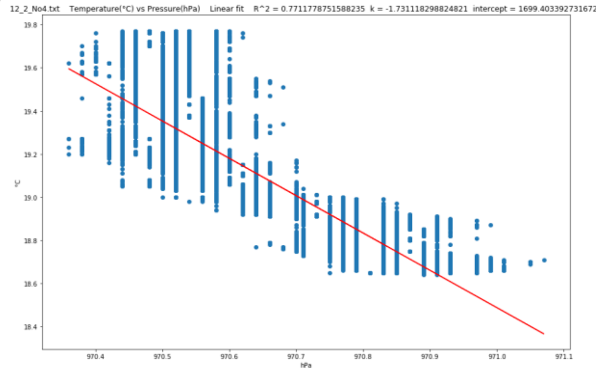
Then again, we tried to find some correlation between pressure and temperature for each individual device to see if it provides any correlations. Temperature indoor and outdoor differs more than 10 degrees so we might be able to use temperature to relate pressure gradient.

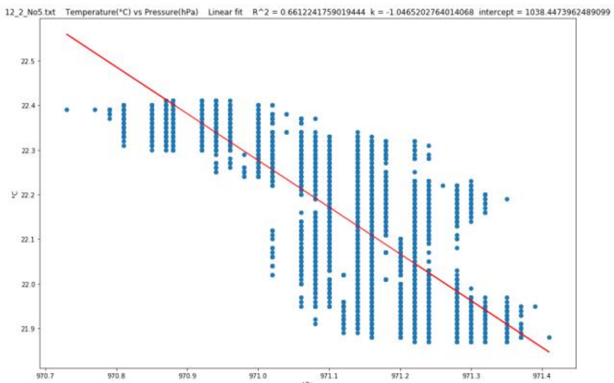
There are scattering plots for each device and below are some examples:

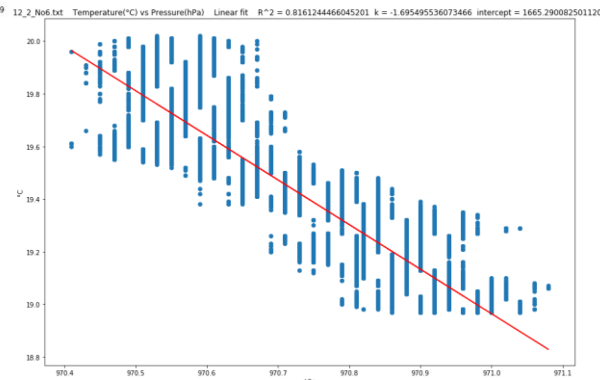












Most of them indicate that pressure would increase when temperature drops, except for device No.3 which shows a quite different relation.

We choose to do a linear fit for all these graphs since both P and t were both varying in a small range. They don't have big R^2 value, but again we assume neither pressure nor temperature is varying much, and a similar correlation certainly exists.

Device	Type	Location	k' for PT graph	Intercept for PT graph
No.1	Pink Box	Outside West	-3.3115	3224.066
No.2	Yellow Box	Outside East	-3.618	3521.627
No.3	White Box	Front Door	1.9218	-1847.046
No.4	Bred	Triple Doors	-1.7311	1699.403
No.5	Bred	Living Room	-1.0465	1038.4474
No.6	Bred	Kitchen	-1.6955	1665.2901
No.7	Bred	Dining Room	-1.2212	1207.5872
No.8	Red Box	Piano	-0.6996	701.2511
No.9	Transluscent Box	Plants	-0.7994	799.0075

We collect both invariants for ‘kx+b’ for every device. Only device No.3 differs a lot so we choose to ignore it.Combined with time-averaged data we see that:

1. k for both devices outside is significantly lower than others just as temperature and the opposite of humidity.

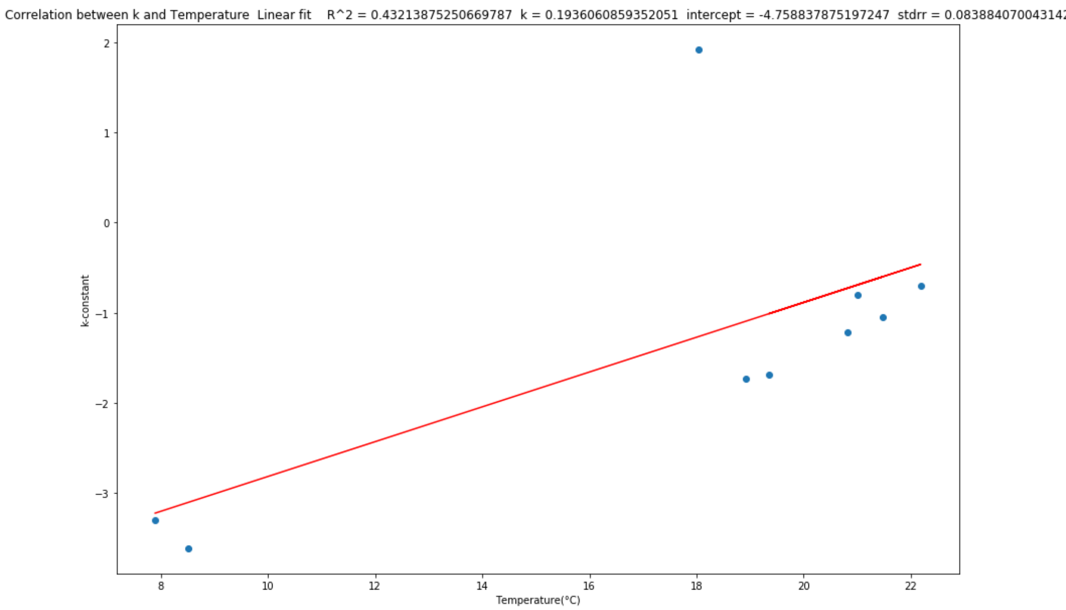
2. b seems to increase as k decreases and humidity increases. So k is in line with T and b is in line with H

3. We can express k & b in terms of T and find a PT correlation

4. We also need to fit PH graph to get a PH correlation

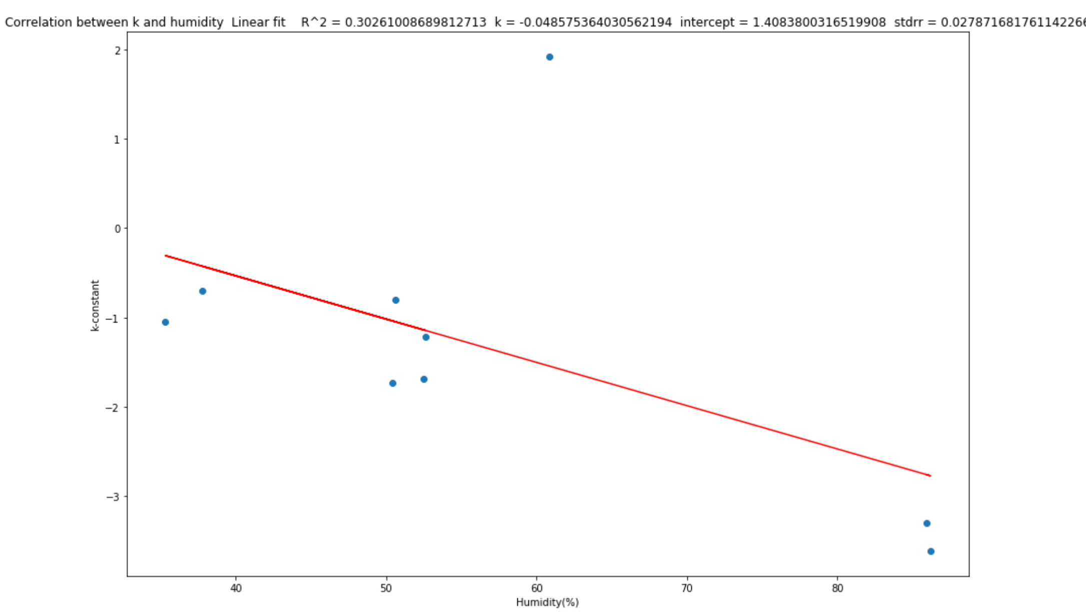
5. After we get both correlation we can use pressure to measure how much to air is leaked into the house, provided we know both outdoor and sealed indoor air properties.

We can now try plotting and fitting some quantities.



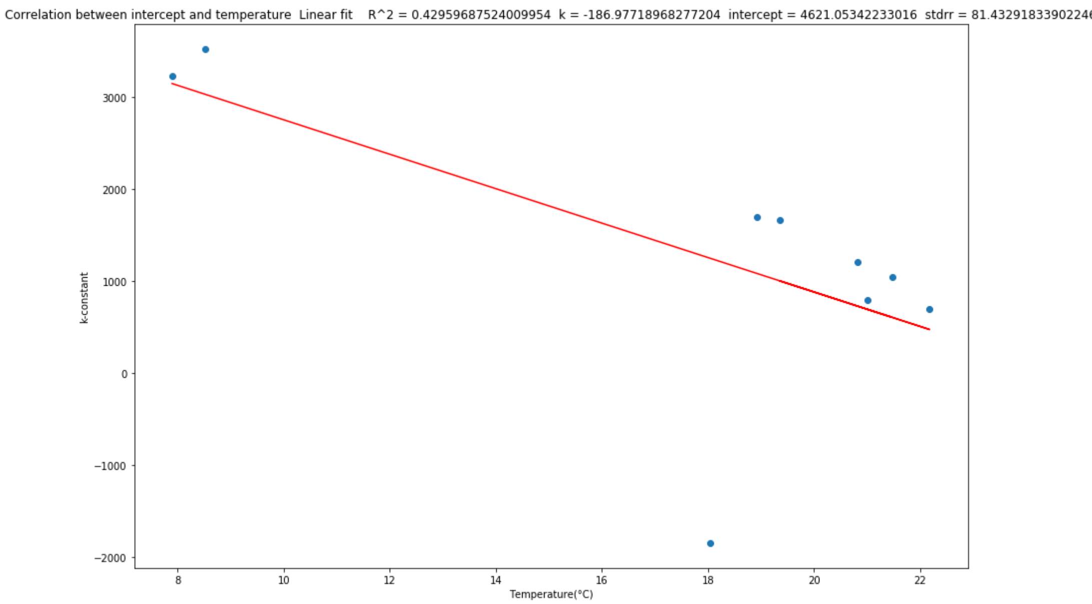
This is ‘k-constant vs Temperature’ graph. It seems k is growing proportionally with temperature.

$$k = 0.1936T - 4.75883$$



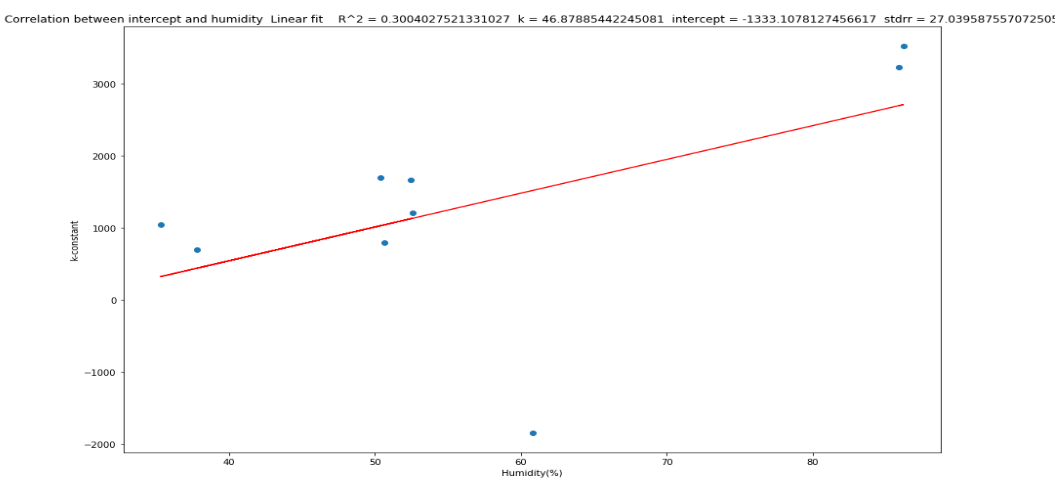
This is ‘k-constant vs Humidity’ graph. It looks like k is decreasing with humidity.

$$k = -0.04857H + 1.4084$$



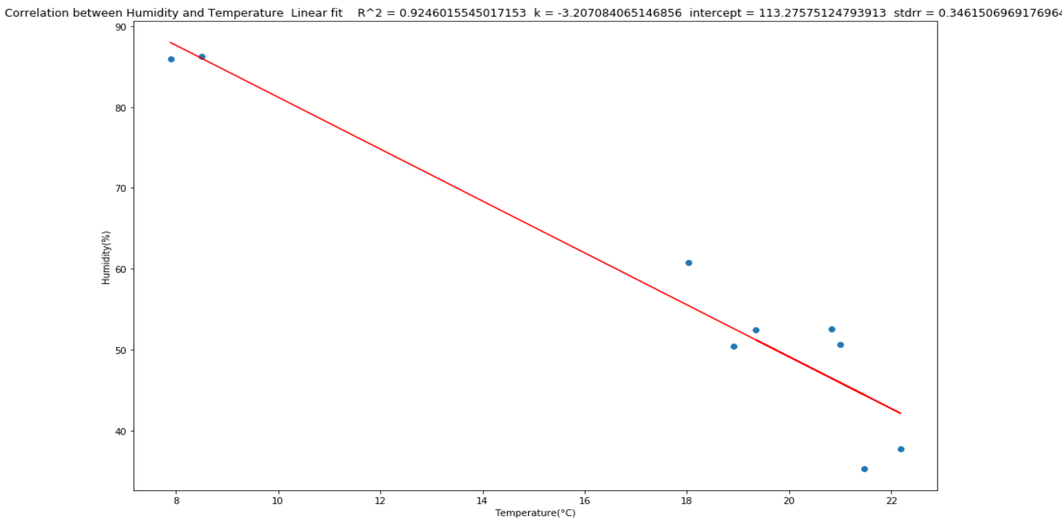
Intercept vs Temperature

$$b = -186.977T + 4621.053$$



Intercept vs Humidity

$$b = 46.8789H - 1333.108$$

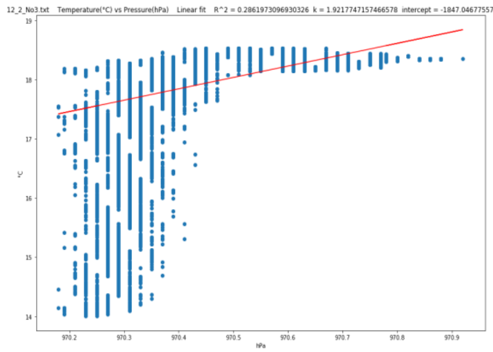


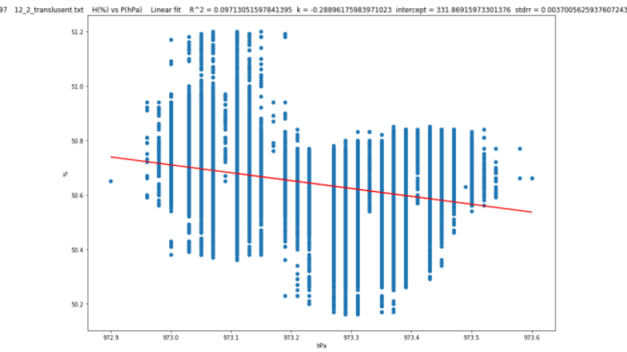
Last there is a Humidity vs Temperature graph. In this case it means the outdoor air is cooler and more humid than the indoor air.

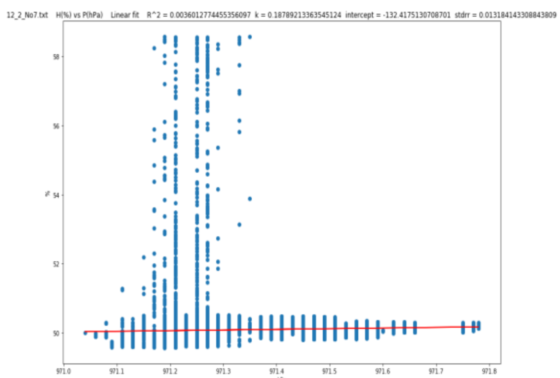
$$H = -3.2071T + 113.2758$$

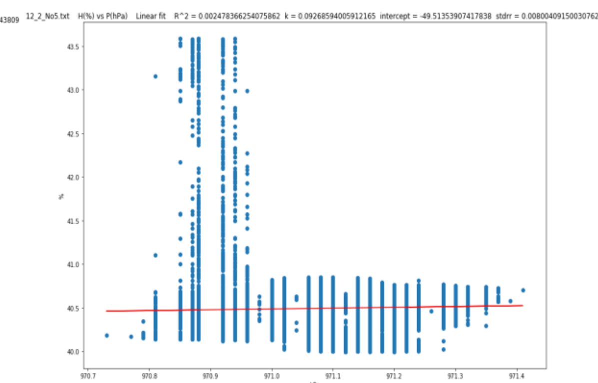
This approximated linear fit will vary if the indoor(bottom-right) and outdoor(top-left) air property changes. It provides us with a HT equation which we can plug back in the PT relation and get a PH relation. But before that, we should check with our data: the H vs P graph.

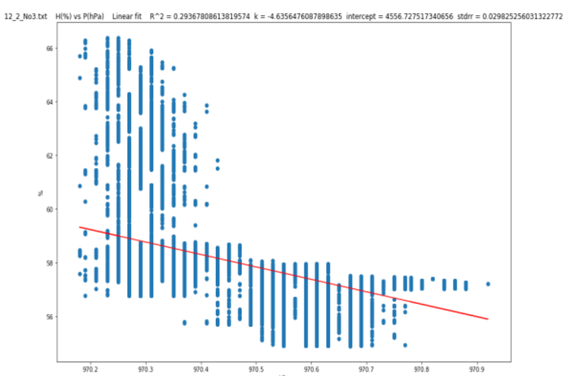
So we need some correlations of Humidity vs Pressure. But unfortunately as shown below they differs a lot between devices.

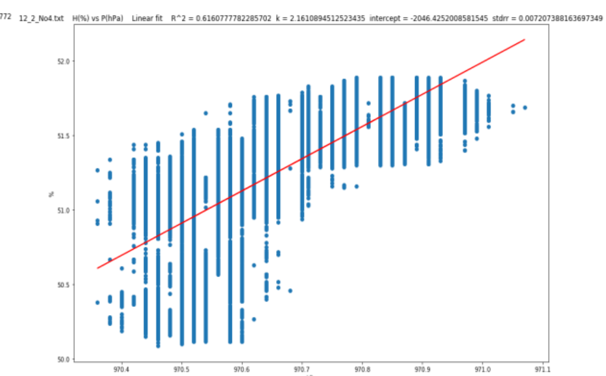












VIII. Conclusion

So we know pressure depends on temperature by approximately T = kP + b, which is P = (T - b) / k :

P ~ (T + 186.977T - 4621.053) / (0.1936T - 4.75883) around 1 atm & room temperature of course

If the approximation we did isn't taking it too far, we can assume that we already get our result, since T can be substituted with H.

However, this expression probably lacks both accuracy and universality, provided that R^2 on PT graph isn't big enough, and that there's only 9 sets of data taken at mere 1 place in 1 day due to our limited hardware setup and sample size. Fathermore, our data is not sufficient or accurate enough to show any valid wind speed or support our humidity-pressure correlation, so further study and experimenting is recommended to draw convincing conclusion to the matter.

Reference

Meier, A. (1994). Home Energy Magazine. Infiltration: Just ACH50 Divided by 20? Retrieved from http://www.homeenergy.org/show/article/nav/blowerdoor/id/1015

Figure 1, https://store.arduino. cc/usa/arduino-mega-2560-rev3

Figure 2, https://www.adafruit.com/product/2652

Figure 3, https://www.adafruit.com/product/746

Figure 4, https://www.adafruit.com/product/1733

Figure 5, https://www.adafruit.co m/product/3013

Figure 6, https://www.wunderground.com/history/daily/us/il/champaign/KCMI/date/2018-12-2