



instructables

Temperature, Relative Humidity, Atmospheric Pressure Logger Using Raspberry Pi and TE Connectivity MS8607-02BA01

by Noredidine Kessa

Introduction:

In this project I will show you how to build setup by step a logging system for temperature humidity and atmospheric pressure. This project is based on the Raspberry Pi 3 Model B and TE Connectivity environmental sensor chip MS8607-02BA01, this chip is really tiny so i suggest you get it in an eval board, trying to hand solder it is not recommended, I got its eval board DPP901G000 on Amazon for \$17. The program that runs this project is on github and is written in python 3.

I will try as much as i can to provide all the boring details so anyone with basic computer skills can successfully build this system.

Sources and References:

<http://www.te.com/commerce/DocumentDelivery/DDECon...>

<http://www.te.com/commerce/DocumentDelivery/DDECon...>

https://en.wikipedia.org/wiki/Raspberry_Pi

<https://github.com/anirudh-ramesh/MS8607-02BA01/bl...>

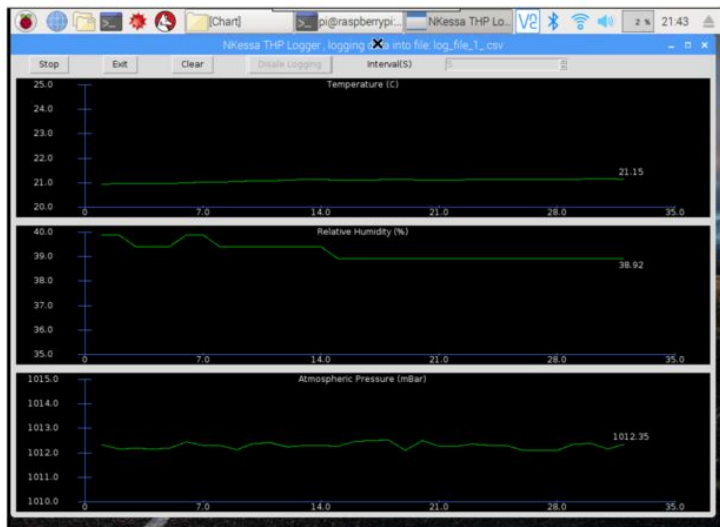
Parts and Tools Needed:

-Raspberry Pi 3 Model B and accessories: case, mouse, keyboard, monitor or TV, microSD card etc.

-MS8607-02BA01 eval Board, DPP901G000 or equivalent, will refer to it in the rest of this instructable as the Sensor Board.

- Four prototyping wires to connect the Raspberry Pi to the sensor board

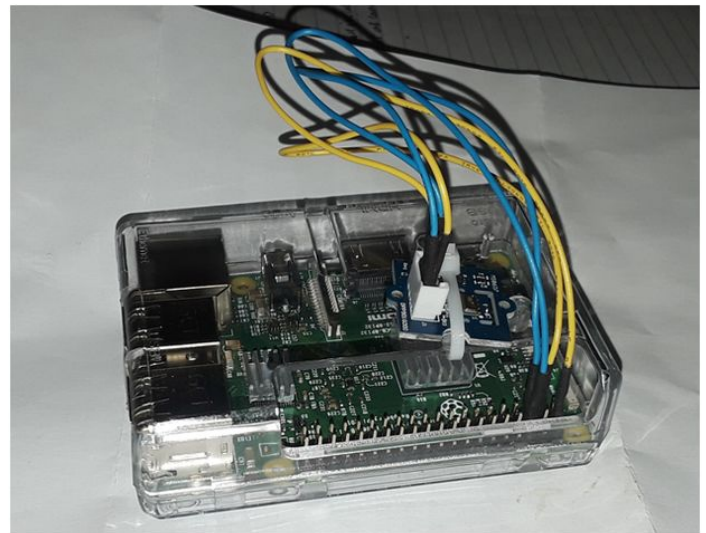
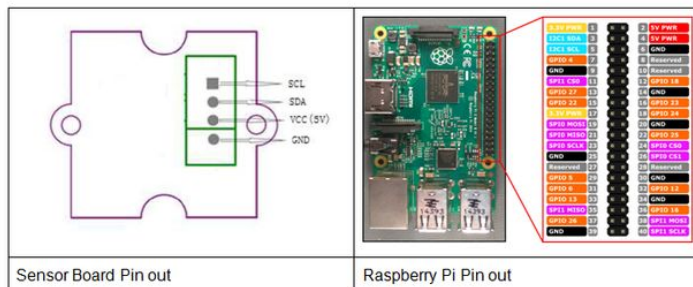
-Computer to setup the Raspberry Pi, I used a computer running Ubuntu, a Windows PC will work with some changes to the instructions.



Step 1: Hardware Setup

-Connect the Raspberry Pi to the Sensor Board as described in the table and picture above

Connection Label	From Raspberry Pi 3 Model B	To Sensor Board DPP901G000
I2C-SCL	PIN-5	PIN-1
I2C-SDA	PIN-3	PIN-2
VCC (5V)	PIN-2	PIN-3
GND	PIN-6	PIN-4



Step 2: Download Raspbian

-Download Raspbian SD card image from <https://www.Raspberrypi.org/downloads/>

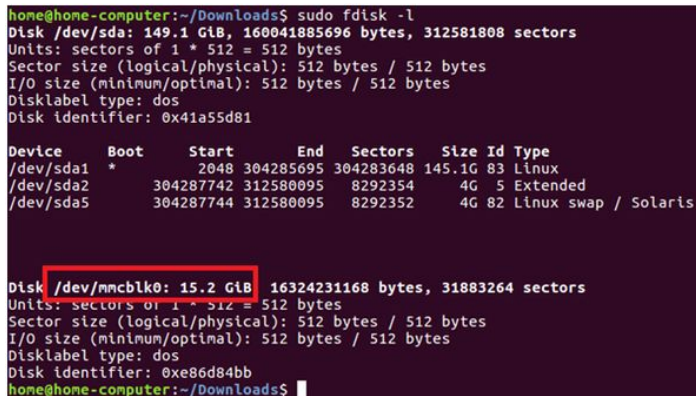
-Browse to your download folder and unzip the Raspbian SD card image using the command unzip.

```
home@home-computer: ~/Downloads
home@home-computer:~/Downloads$ unzip 2017-11-29-raspbian-stretch.zip
Archive: 2017-11-29-raspbian-stretch.zip
  inflating: 2017-11-29-raspbian-stretch.img
home@home-computer:~/Downloads$
```

Step 3: Identifying the Micro SD Device

-Place a MicroSD card in a Micro SD card reader/writer that is connected to the PC,

-Identify the micro SD card device name on your PC using “sudo fdisk -l” command as shown below, notice how the SD card device is identified by size and device name, in this particular case the SD card device name is “/dev/mmcblk0”, on your computer it can be different. If you have a windows computer use Win32 Disk Imager for this step.



```
home@home-computer:~/Downloads$ sudo fdisk -l
Disk /dev/sda: 149.1 GiB, 160041885696 bytes, 312581808 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x41a55d81

Device      Boot      Start         End      Sectors  Size Id Type
/dev/sda1   *          2048     304285695   304283648  145.1G 83 Linux
/dev/sda2             304287742   312580095     8292354    4G  5 Extended
/dev/sda5             304287742   312580095     8292352    4G 82 Linux swap / Solaris

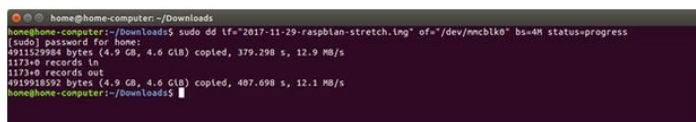
Disk /dev/mmcblk0: 15.2 GiB, 16324231168 bytes, 31883264 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xe86d84bb
home@home-computer:~/Downloads$
```

Step 4: Copying the Raspbian Image Into the MicroSd Card

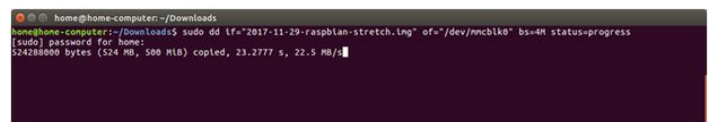
-Burn Raspbian into the MicroSD card using the command:

dd if=SDcard_image_file_name of=SD_Card_Device_Name status=progress.

Wait for the copying to finish, this will take few minutes.



```
home@home-computer:~/Downloads$ sudo dd if="2017-11-29-raspbian-stretch.img" of="/dev/mmcblk0" bs=4M status=progress
[sudo] password for home:
4911529984 bytes (4.9 GB, 4.6 GiB) copied, 379.298 s, 12.9 MB/s
117340 records in
117340 records out
49159918592 bytes (4.9 GB, 4.6 GiB) copied, 407.698 s, 12.1 MB/s
home@home-computer:~/Downloads$
```

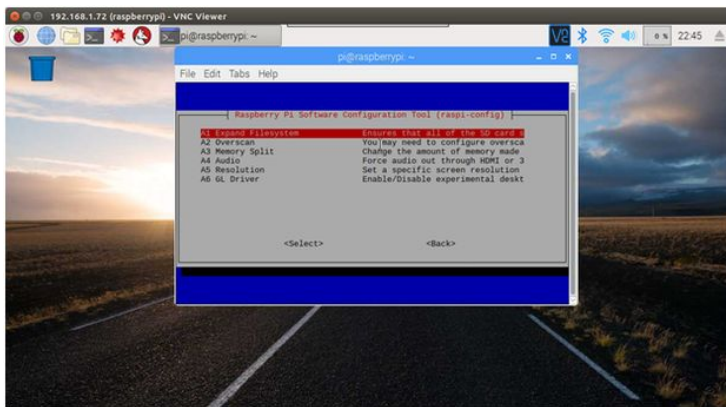


```
home@home-computer:~/Downloads$ sudo dd if="2017-11-29-raspbian-stretch.img" of="/dev/mmcblk0" bs=4M status=progress
[sudo] password for home:
524288000 bytes (524 MB, 500 MiB) copied, 23.2777 s, 22.5 MB/s
home@home-computer:~/Downloads$
```

Step 5: Coming Alive for the First Time

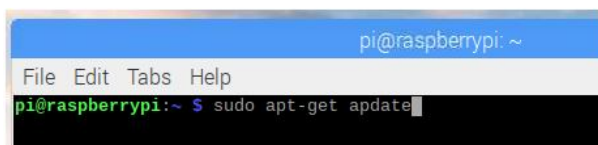
-Remove the micro SD from the PC and place in in the Raspberry, apply power, the Raspberry Pi should boot up.

- On the Raspberry Pi expand the SD, by opening the command line terminal, then type “sudo raspi-config”, select Expand Filesystem to take advantage of the whole space available on the SD card. Reboot when asked to reboot.



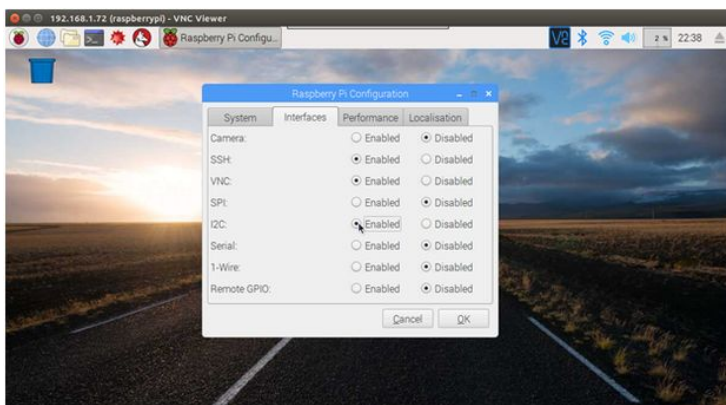
Step 6: Update Packages List

- Connect Raspberry Pi to Wifi or plug it in using an ethernet cable from your home router.
- On the command line in the Raspberry Pi run “sudo apt-get update” to update packages list.



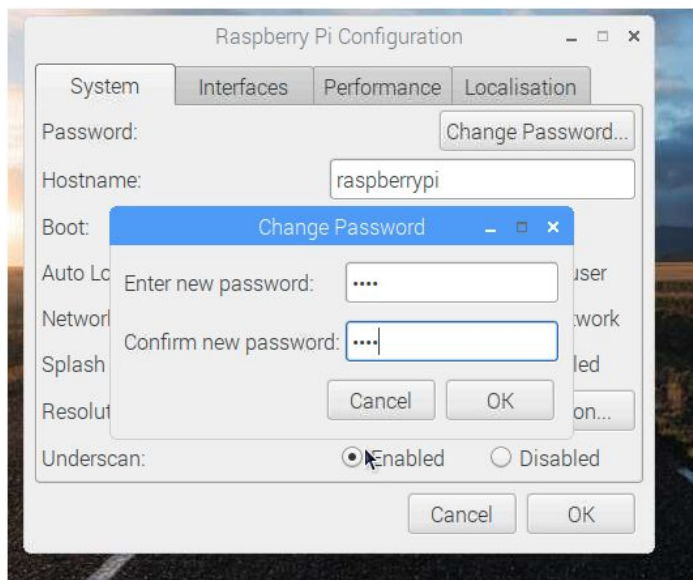
Step 7: Enable VNC, SSH and I2C

On the Raspberry Pi Desktop main menu, click on Preference then select Raspberry Pi Configuration utility. In the Interfaces tab, enable SSH, VNC, and I2C.



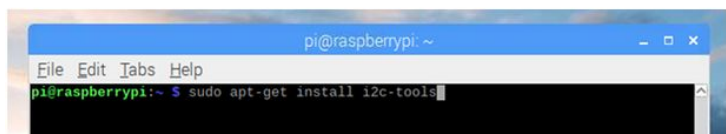
Step 8: Change Raspberry Pi Password

-Now is a good time to change the Raspberry Pi password.



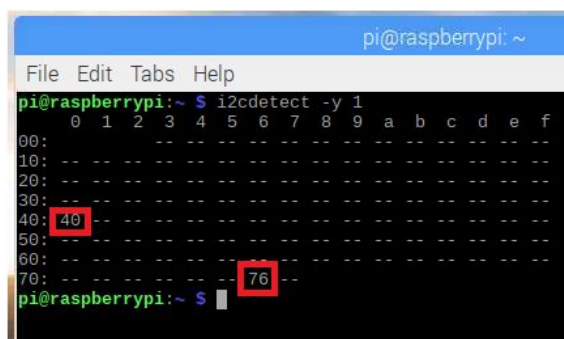
Step 9: Install I2c-tools

On the command line install I2C tools using the command “sudo apt-get install i2c-tools”



Step 10: Verifying I2C Communication

- Verify that Raspberry Pi can communicate with the Sensor Board over I2C using the command “i2cdetect -y 1”, the Sensor Board actually has two I2C devices, device address 0x76 is for measuring pressure and temperature, device address 0x40 is for measuring relative humidity. Verify that both of them are found.

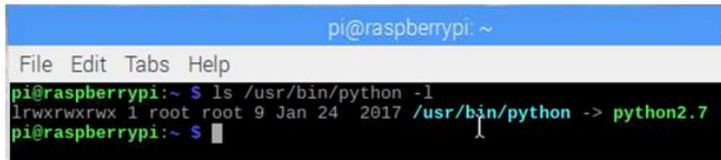


Step 11: Checking Python Version

The program that we will run to read the sensors data needs at least Python version 3.2 to run, older versions will not run the program properly.

Linux uses a symbolic link (look up symbolic links on linux OS online to understand what i am talking

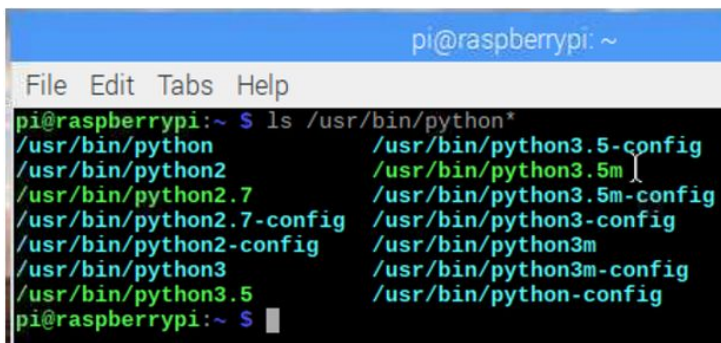
about) to point to which python interpreter version to be used to run python scripts. Use the command “ls /usr/bin/python -l” to see the version pointed to, in this particular case it is pointing to python2.7 which will not work for us.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ ls /usr/bin/python -l  
lrwxrwxrwx 1 root root 9 Jan 24 2017 /usr/bin/python -> python2.7  
pi@raspberrypi:~$
```

Step 12: Checking Available Python Versions

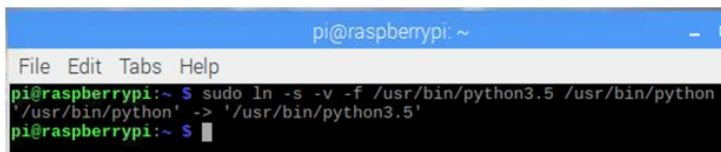
Use the command “ls /usr/bin/python*” to see all available python versions on your Raspberry Pi.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ ls /usr/bin/python*  
/usr/bin/python          /usr/bin/python3.5-config  
/usr/bin/python2         /usr/bin/python3.5m  
/usr/bin/python2.7       /usr/bin/python3.5m-config  
/usr/bin/python2.7-config /usr/bin/python3-config  
/usr/bin/python2-config  /usr/bin/python3m  
/usr/bin/python3         /usr/bin/python3m-config  
/usr/bin/python3.5       /usr/bin/python3m-config  
pi@raspberrypi:~$
```

Step 13: Update Python Symbolic Link

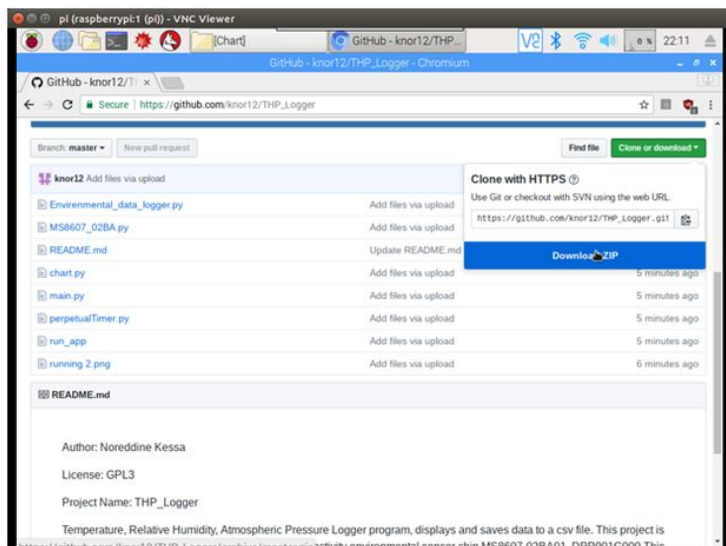
It appears that we have version python3.5, let's link it symbolically to /usr/bin/python



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ sudo ln -s -v -f /usr/bin/python3.5 /usr/bin/python  
'/usr/bin/python' -> '/usr/bin/python3.5'  
pi@raspberrypi:~$
```

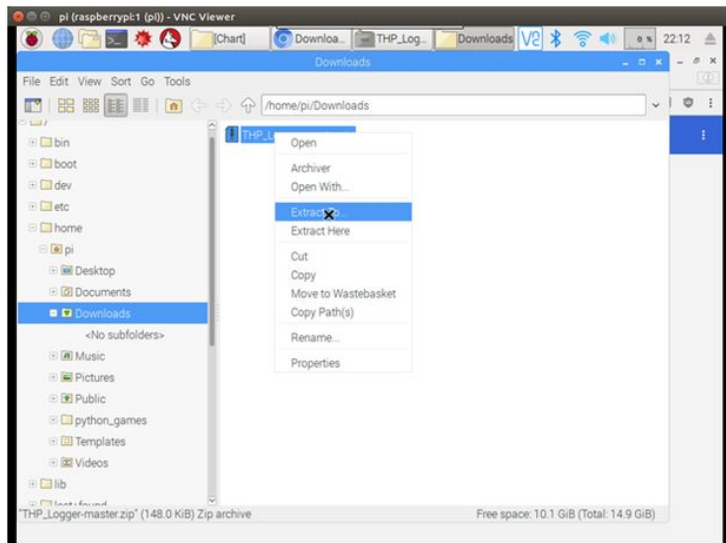
Step 14: Download the THP Logger Source Code

-Download the THP Logger source code from Github https://github.com/knor12/THP_Logger



Step 15: Unzip the Source Code Zip File.

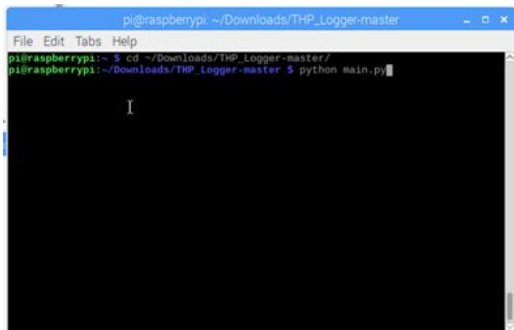
-Unzip the source code zip file.



Step 16: Run the THP Logger

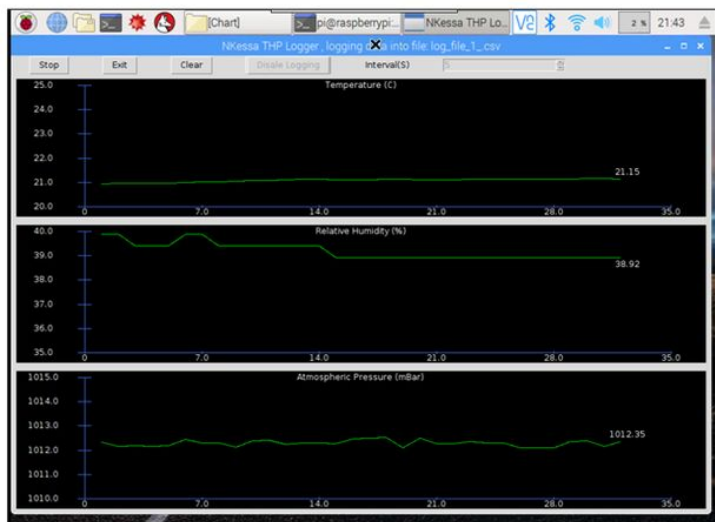
-Using the command line terminal change the current working directory using “cd ~/Download/THP_Logger-master”

-Run the THP Logger app using the command “python main.py”



Step 17: Start Measuring THP

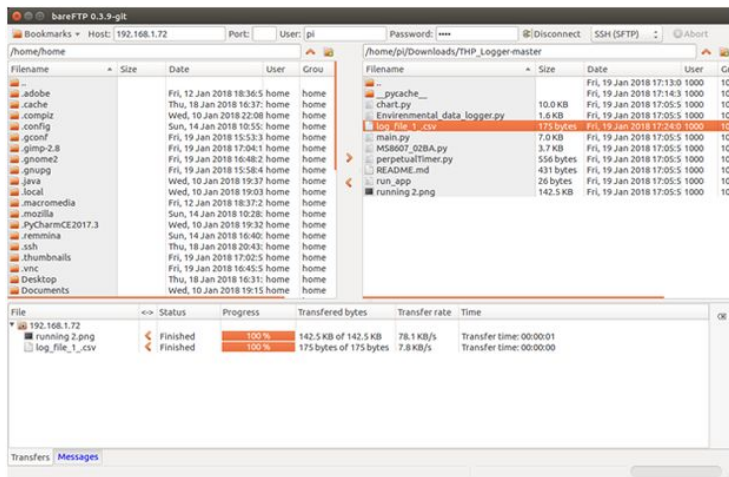
- Enable logging, select the appropriate log interval for your needs, let it run.



Step 18: Getting Data Over SFTP

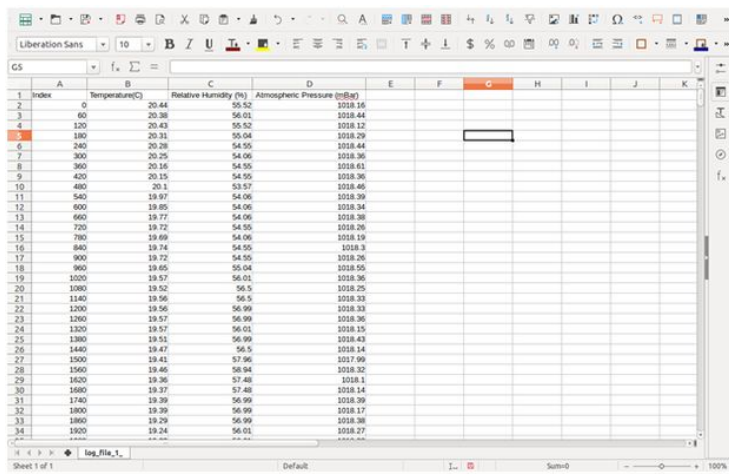
-I haven't tested it against calibrated test equipment but the measurements reported are consistent with my heating thermostat. I also noticed a drop in humidity when i open the door because it is freezing outside and the humidity outside is significantly less than inside.

-Get the data over in csv format from the Raspberry Pi to your PC over SSH, using your favorite SFTP client program, for Windows you can use WinSCP, I use bareFTP for my linux machine.



Step 19: Looking at the Data

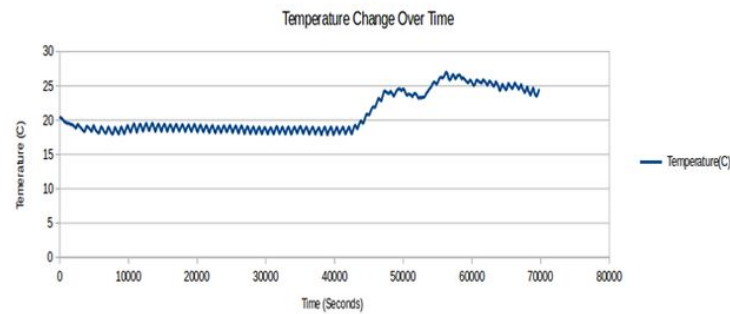
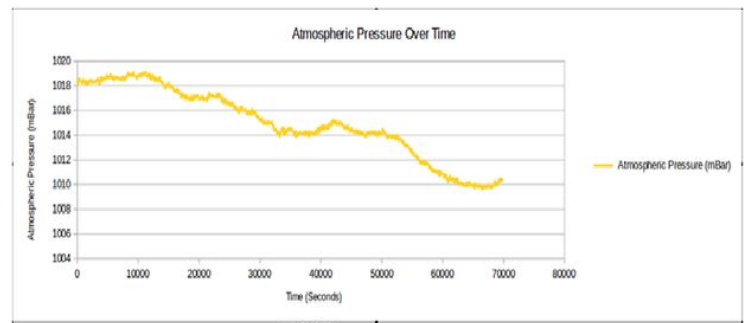
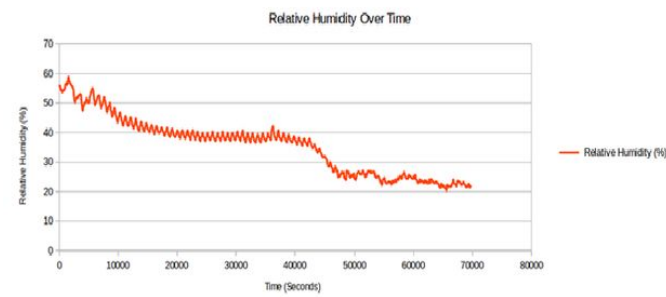
-Open the csv file imported using Microsoft Excel or OpenOffice Calc, use the data to generate chart to view the environmental changes over the day or days.



Step 20: Processing the Data

The app doesn't generate too much data for instance if you run the app over 24 hours with acquisition intervals of 60 seconds, the size of the data file is about 50 KiB

Above are charts I generated with LibreOffice Calc program using the data generated over 70000 seconds (19 hours), one measurement is taken every 60 seconds.



Step 21: Room for Improvement.

Feel free to improve on this project, few suggestions:

1-Publish the data to an internet server such as <https://thingspeak.com/>

2-Have the data processed and displayed by your own web server hosted on the Raspberry Pi

3-Have the program run headless at startup and acquire data indefinitely and warn you if certain

conditions are met etc.

4-Extend the functionality of the system by adding more sensors and actuators on the I2C bus, or SPI bus.

5-Save the data to a USB flash drive instead of the SD card, have the program name data files based on date/time.



**THERE'S
ALWAYS ROOM
FOR
IMPROVEMENT.**

QUOTEHD.COM

Shannon Fritcher

Step 22: