

Issues and Mitigation – Weather Station Build.

This document is to outline any trouble or issues I encountered and any fixes I implemented while following the tutorial at this link: <https://www.raspberrypi.org/learning/weather-station-guide/>

The tutorial is simple enough to follow but I found a good few issues while working through it. Hopefully, this document will help anyone else who is attempting this project with any issues they may have faced.

Solving some of the issues I found was a great way to understand some of the underlying functionality of the software and how everything ties together, so if you are up for the challenge, I would recommend attempting the tutorial yourself and trying out fixes you come across before consulting this guide.

Note: *This document contains fixes that worked for me as of early 2020, I cannot guarantee they will work for you or even be accurate in a few months/years time. Use this guide in conjunction with your own research to fix any issues you may stumble upon.*

All of these fixes were implemented in Raspbian Stretch Lite (the Terminal only version). I originally used the newer Raspbian Buster, but I switched to Stretch in the hopes of fixing some of my issues (As the guide was Originally written for Stretch). I have no doubt that most of the fixes implemented would work in newer versions, but I cannot guarantee.

I'll try link any sources I used for my fixes, but I might have omitted some.

Issue 1 – Setting up the RTC

This project should work fine without setting up the real time clock (RTC), but I wanted to get it working.

The Issue: No ***dev/rtc**** file(s) appearing in Raspbian. These seem to be necessary for the functioning of the RTC, and none of the ***hwclock*** commands seem to work without them and will throw errors.

Fix: Found on this forum post: <https://www.raspberrypi.org/forums/viewtopic.php?t=192564>

Someone outlines extra arguments they put in the ***boot/config.txt*** . I copied the ones that seemed necessary, rebooted, and RTC files showed up.

<pre># Additional overlays and parameters # Enable audio (loads snd_bcm2835) dtparam=audio=on dtoverlay=w1-gpio dtoverlay=i2c-rtc,pcf8523 dtparam=spi=on dtparam=i2c_arm=on_</pre>	<pre>Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. SSH is enabled and the default password is 'root'. This is a security risk - please log in with the usual user name rather than root! pi@raspberrypi:~\$ ls /dev/rtc* /dev/rtc /dev/rtc0 pi@raspberrypi:~\$</pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Issue 2 – Signing up to Oracle DB

Again, this project should work fine without using Oracle DB, as any data collected is stored on the device itself, but I thought I would note it in here.

Issue: Signing up for the Oracle DB for this project. I tried multiple times, but I never received an authentication code by email. The link is:

https://apex.oracle.com/pls/apex/f?p=81290:LOGIN_DESKTOP:0:::::&tz=1:00

Fix: I decided to use a different cloud based data displaying service. The choice is up to you on this one. It will however take some time to set up, as you must write new scripts to take data out of the database, connect to and upload the data to one of these services and troubleshoot them yourself.

Alternatively, you could get in contact with Oracle or the people behind this page regarding this issue. It would save you having to rewrite scripts as the ones that upload to the Oracle DB are provided. Since the weather kits used were limited and given out years ago, I decided not to bother as it seems they might have ended support for this program and more than likely have no interest in booting it back up for a handful of people. The prior option would be better for future proofing the project too and learning some programming, but the option is always there!

Issue 3 – Omitted or outdated necessary software Installs.

Issue: The tutorial leaves out some of the needed package installs, either because they are installed by default with whatever version of Raspbian they are using, or because they are outdated and do not work. You may already have some of these installed, but just incase: Use ***sudo (package manager) install (package)*** and the list below for each package manager and package. I noted down ones which I installed, but some may be missing from the list.

Fix: **apt-get :**

- git
- python-pip
- python3-smbus

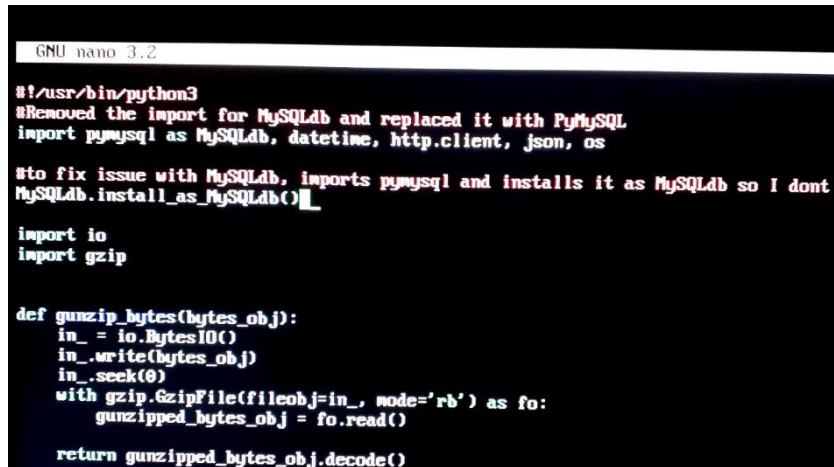
pip (NOTE: *I found this worked instead of pip3, which didn't work for me*) :

- mysqlclient

NOTE: *There is something about the installs for the **php** stuff where you must change the name of the packages to include **php7.0** or **php** instead of **php5** or else the packages will not show up.*

Issue 4 – Scripts.

Issue: In *log_all_sensors.py* , The Module **MySQLdb** could not be imported. This should have been an easy fix right? Nope. **MySQLdb** is not a valid package on any package manager I tried. My first attempt was to use an alternative: **PyMySQL** and import it as **MySQLdb** using the built-in compatibility function as suggested.



```
GNU nano 3.2

#!/usr/bin/python3
#Removed the import for MySQLdb and replaced it with PyMySQL
import pymysql as MySQLdb, datetime, http.client, json, os

#to fix issue with MySQLdb, imports pymysql and installs it as MySQLdb so I dont
MySQLdb.install_as_MySQLdb()

import io
import gzip

def gunzip_bytes(bytes_obj):
    in_ = io.BytesIO()
    in_.write(bytes_obj)
    in_.seek(0)
    with gzip.GzipFile(fileobj=in_, mode='rb') as fo:
        gunzipped_bytes_obj = fo.read()

    return gunzipped_bytes_obj.decode()
```

Fix: I discovered this: <https://raspberrypi.stackexchange.com/questions/78215/how-to-connect-mysqldb-in-python-3> that using **pip** to install **mysqlclient** worked, and **MySQLdb** was not giving any import errors anymore.

Issue 5 – Script Syntax.

Issue: Some of the scripts used for this project were giving me awful trouble, the main contenders were: **database.py** and **log_all_sensors.py**. Funnily enough after spending many frustrating hours trying to figure out what went wrong with these, and after reading many forum posts I finally discovered it was a simple syntax error, brought up by a change in

```
Traceback (most recent call last):
  File "/home/pi/weather-station/log_all_sensors.py", line 12, in <module>
    db = database.weather_database() #Local MySQL db
  File "/home/pi/weather-station/database.py", line 114, in __init__
    self.db = mysql_database()
  File "/home/pi/weather-station/database.py", line 25, in __init__
    self.connection = MySQLdb.connect(user=credentials["USERNAME"], pa
  File "/usr/lib/python3/dist-packages/MySQLdb/__init__.py", line 81,
    return Connection(*args, **kwargs)
  File "/usr/lib/python3/dist-packages/MySQLdb/connections.py", line 26
    super(Connection, self).__init__(*args, **kwargs2)
TypeError: 'database' is an invalid keyword argument for this function
Exception ignored in: <bound method mysql_database.__del__ of <database
Traceback (most recent call last):
  File "/home/pi/weather-station/database.py", line 42, in __del__
    self.connection.close()
AttributeError: 'mysql_database' object has no attribute 'connection'
Connection closed
pi@raspberrypi:~/weather-station $ _
```

(In hindsight I should have been able to figure out the issue pretty easily. Breaks are important!)

keywords in one of the packages (I was basically kicking myself after figuring it out). If you check the documentation for **MySQLdb** it notes these keywords and their uses.

Fix: In **database.py** replace the word: **password** with: **passwd** and also replace: **database** with: **db**

In **log_all_sensors.py** I had to replace: **weather_database** with: **weather_db**. This was probably brought up by me using the find and replace tool in the nano editor with the other script. This script was not problematic otherwise like I thought it was.

Easy as *pi* the scripts now work!