SwitchDoc Labs

SkyWeather2 Installation Guide

This is a Maker Community Provided Document
(No affiliation with SwitchDoc Labs)

Written by: *Sopwith*
Revision 1.5
June 4, 2022
sopwith@ismellsmoke.net

Important Information

This document is a summation of the steps I took to get my *SkyWeather2* system up and running. I did not purchase the pre-installed SDCard from *SwitchDoc Labs,* so I was on my own to figure out how to make it work. As of today, there is no SDCard image posted on *Github*, and, as far as I know, there is no document that describes how to do it.

IMPORTANT: I am not an employee of *SwitchDoc Labs*. They did not request that I produce this document. All the content in this document is the result of my discovery efforts on how to get all of the software working. Any errors, omissions, oversights, and any other incorrect information described here belong to me. *DO NOT* contact *SwitchDoc Labs* for technical support if you have problems getting your *SkyWeather2* working based on these instructions. Instead, send me an Email (sopwith@ismellsmoke.net) and I will try to help solve any problems you discover.

REALLY IMPORTANT: The script is designed to be used for an initial installation of the *SkyWeather2* system. If you already have a *SkyWeather* system up and running, and you want to *update* your installation, **DO NOT** use my script. It will not work. Your existing database will be deleted. The script is to be used on a clean Raspberry Pi installation only.

There is a complete set of SkyWeather2 documentation on the *SwitchDoc Labs* website. You should consider their documentation as the definitive guides.

If you are a new Maker or have little experience with a Raspberry Pi and/or Linux, then you should purchase the pre-loaded SDCard. It will save you hours of frustration.

If you have some experience with a Raspberry PI and are fearless when it comes to trying something new or unfamiliar, you should be able to follow this guide to get your *SkyWeather2* working.

If you are an experienced Maker with Pi experience, following this guide, you should be able to get your *SkyWeather2* running in less than an hour.

If you want the TL:DR version of the install process, refer to the "*skyweather2_install_cheater.txt" document.*

This document is based on my experiences in getting the *SkyWeather2* working. Your results may vary. There is no guarantee that these instructions will work for you. Software and hardware components are constantly changing, and these may affect your results.

SkyWeather2 is an open-source project created by *SwitchDoc Labs*. Any copyrighted material, proprietary information, and intellectual property created and owned by *SwitchDoc Labs* is respected and appreciated.

Introduction

Congratulations on being the owner of a *SkyWeather2* weather station. This innovative product is very well designed and really fun to hack. It is based on years of experience collecting weather data using a Raspberry Pi in some very remote parts of the world.

This is an install guide on how to get your *SkyWeather2* working without having to purchase the pre-loaded SDCard. I did not want to spend the money for the card, so I was on my own to figure out how to get it running. After many hours of discovering all of the software dependencies, my *SkyWeather2* is fully operational. I spent the time to create this document, so you do not have to go through the pain that I did.

The installation process you are about to go through involves the following steps:

- ✔ Gather together all of the required components.
- ✔ Install batteries in the WeatherRack2 sensor station.
- ✔ Install batteries in the indoor temperature sensor.
- ✔ Install the SkyWeather hat on the Raspberry Pi.
- ✔ Connect the SDR receiver to the Raspberry Pi.
- ✔ Connect a camera to the Raspberry Pi.
- ✔ Connect a Air Quality Sensor (AQI) **Note**: This is an optional component.
- ✔ Create a Pi OS SDCard.
- ✔ Boot and configure the Raspberry Pi.
- ✔ Run the SkyWeather2 installation script.
- ✔ Secure MySQL and create the SkyWeather2 databases.
- ✔ Test all the software components.
- ✔ Run the SkyWeather2 configuration script.
- ✔ Run the SkyWeather2 script to collect sensor data.
- ✔ Run the SkyWeather2 dash app to visualize your data.
- ✔ Finalize configuration.

NOTE: You do not need to assemble the *SkyWeather2* 3D printed base station in order to get your system configured and working. I simply laid out my Raspberry Pi-3 on my workbench, and made sure the *WeatherRack2* sensor station and indoor temperature sensor were within range to receive their radio signals. In fact, I recommend that you do not install the components into the base station until you complete all the steps in this document.

If you are an experienced maker, this process should take 1-2 hours to complete. Follow the instructions in the order given and there is a high probability you will succeed.

If you have problems along the way, do not ignore them. Figure out what is wrong right then and there. Do not proceed on until the step is complete and working.

The installation script is quite ‘chatty’ and is designed to fail if anything goes wrong. It something fails, you will see an error message to help you figure out how solve the issue.

Step-1: Gather All the Required Components

The first thing to do is gather all of the required components.
This includes:

- *WeatherRack2* sensor array.
- The indoor temperature sensor (Thermo-hygrometer transmitter).
- 3 – AA batteries.
- 2 – AAA batteries.
- Raspberry Pi.
- Micro SDCard (16GB min.)
- *SkyWeather2* hat.
- Air Quality Sensor (AQI) Note: This is an optional component.
- SkyCamera (For testing, any Pi camera will work).

- SDR receiver USB dongle with antenna.
- Small phillips screwdriver
- Installation script zip file.

Step-2: Install batteries in the WeatherRack2 sensor array

The *WeatherRack2* sensor array is the device with the anemometer, wind vane, and rain gauge you install outdoors. Remove it from the box. Turn it over and remove the battery door cover using a small phillips head screwdriver. Install 3 AAA batteries in the right orientation. Replace the door and tighten the screw.
If the device is working correctly, you should see a faint red LED flashing on the bottom of the unit about every 16 seconds

Step-3: Install batteries in the indoor temperature sensor

The indoor temperature sensor requires 2 AAA batteries. Slide the battery cover off on the back of the unit. *Before* you install the batteries, make sure the DIP switches are in the correct position. There is a guide on the battery cover. I left my device on channel 1. I switched the #4 DIP switch down (off) so the display shows temperature in F not C. Install the batteries in the right orientation and replace the battery cover.

If the device is working correctly you should see the temperature, humidity, and channel number on the display.

Step-4: Install the *SkyWeather* hat on the Raspberry Pi

Very carefully slide the *SkyWeather* hat onto the GPIO pins of your Raspberry Pi. Make sure the hat is aligned with all the pins. The hat board should be be facing the Pi board.

Step-5: Install the SDR 433mhz radio receiver

Insert the SDR radio receiver into one of the Pi USB slots. Connect the antenna to the radio receiver.

Step-6: Connect a camera to the Raspberry Pi

Carefully connect a Raspberry Pi camera to the Pi. Make sure the orientation of the ribbon cable is correct for your version of the Pi. I used a Pi-3, so the blue mark on the camera cable faces the Ethernet port. If you do not get the ribbon orientation correct, the camera will not work.
Note: For the purposes of testing the system, you do not need to install the *SkyWeather2* camera. Any standard Pi camera can be used.

Step-7: Connect an Air Quality Sensor (AQI) to the Raspberry Pi

If you purchased an AQI, install it now. Connect the Grove cable to the right slot on the *SkyWeather2* hat.
Note: I did not purchase the AQI device, so I cannot offer any advise.

Step-8: Create a Pi OS SDCard

NOTE: SkyWeather2 software only works on 32-bit Pi OS operating systems.
In this step, you will create a Pi OS SDCard. You will download a Pi OS image from the Raspberry Pi web site and 'burn' it to an SDCard. For this document, I downloaded the Raspberry PI OS Lite – release date of October 30, 2021 (Kernel 5.10 Buster). This is the 'headless' version and does not provide a GUI.
I have tested my script on the latest 32-bit Raspberry Pi OS systems (Buster and Bullseye). You are free to use either, although it is highly recommended you install Buster which is officially supported by SDL.
There is plenty of documentation on the web on how to create a Raspberry Pi OS SDCard so I will not describe the steps here. It is highly recommended you use a quality SDCard that is at least 16GB in size.

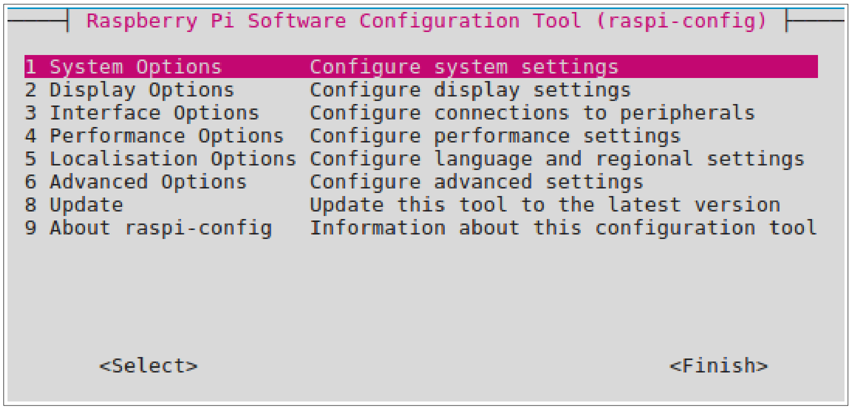
Step-9: Boot and configure the Raspberry Pi

Once you have the Pi OS SDCard ready, install it in your Pi and boot it up.
Note: I highly recommended that you connect a monitor and keyboard to your Pi. This will make it a lot easier to configure. There are ways to hack the Pi OS SDCard with wireless network configuration information, but that is not covered here. It is assumed you have direct control of your Pi.

Note: When you boot your Pi, you should see a constant bright blue LED on the *SkyWeather* hat.

Note: When you boot a Pi running Bullseye, for security reasons, it no longer creates a user named *pi*. It will prompt you to create a user. You ***must*** create a user named *pi* for the system to work correctly. It is recommended that you set the *pi* user password to *raspberry*.

Once the Pi is booted, log in with the user: *pi* and the password: *raspberry*.
From the command line run:
$ sudo raspi-config



```
Raspberry Pi Software Configuration Tool (raspi-config)

1 System Options       Configure system settings
2 Display Options      Configure display settings
3 Interface Options    Configure connections to peripherals
4 Performance Options  Configure performance settings
5 Localisation Options Configure language and regional settings
6 Advanced Options     Configure advanced settings
8 Update               Update this tool to the latest version
9 About raspi-config   Information about this configuration tool

     <Select>                                     <Finish>
```

Figure-1: raspi-config

Use *raspi-config* to configure the following items:

- System Options – S1 Wireless LAN (SSID and password)
- Interface Options – P1 Pi Camera (Enable) **Note:** *May say legacy camera support (Bullseye).*
- Interface Options – P2 SSH (Enable)
- Interface Options – P5 I2C (Enable)
- Localisation Options – L1 Locale (Select correct locale)
- Localisation Options – L2 Timezone (Select correct timezone)
- Localisation Options – L3 Keyboard (Select keyboard)

Once you have your Pi configured, reboot it:
$ sudo reboot

When the Pi has rebooted, log in and enter the below two commands:
$ sudo apt update
$ sudo apt upgrade -y

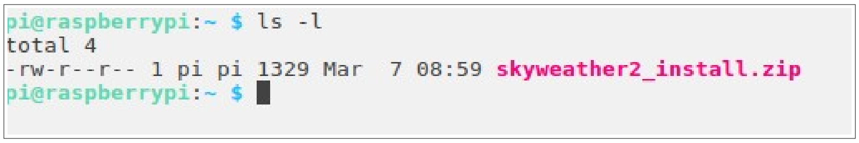
This will take a few minutes to complete. Once completed, reboot it:
$ sudo reboot

Your Pi is now configured, updated, and ready for the *SkyWeather2* install.

Step-10: Run the SkyWeather2 installation script

I created a bash shell script to install all of the required software components for the *SkyWeather2* system. It is in a zip file (*skyweather2_install.zip*) with two other text files that contain the names of the software components. Download the zip file and copy it to the */home/pi* folder on your Pi.

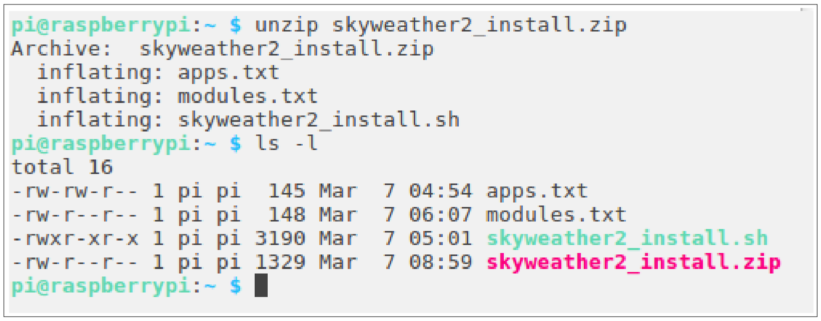
Note: You can copy the zip file to a USB thumb drive or *scp* it from another computer.
From my Linux laptop, the *scp* command I use is:
$ *scp skyweather2_install.zip pi@192.168.1.xx:~/*
(Be sure to use the IP address of your SkyWeather2 Pi.)



```
pi@raspberrypi:~ $ ls -l
total 4
-rw-r--r-- 1 pi pi 1329 Mar  7 08:59 skyweather2_install.zip
pi@raspberrypi:~ $ 
```

Figure-2: Install script in Pi home folder

Unzip install file using the below commands:
$ cd
$ unzip skyweather2_install.zip



```
pi@raspberrypi:~ $ unzip skyweather2_install.zip
Archive:  skyweather2_install.zip
  inflating: apps.txt
  inflating: modules.txt
  inflating: skyweather2_install.sh
pi@raspberrypi:~ $ ls -l
total 16
-rw-rw-r-- 1 pi pi  145 Mar  7 04:54 apps.txt
-rw-r--r-- 1 pi pi  148 Mar  7 06:07 modules.txt
-rwxr-xr-x 1 pi pi 3190 Mar  7 05:01 skyweather2_install.sh
-rw-r--r-- 1 pi pi 1329 Mar  7 08:59 skyweather2_install.zip
pi@raspberrypi:~ $ 
```

Figure-3: Install script unzipped

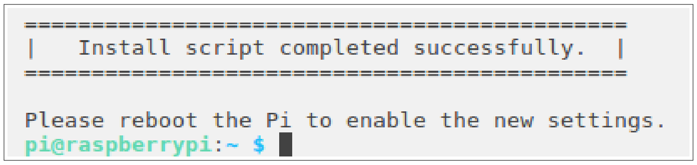
The *apps.txt* file contains the names of all of the applications that will be installed by the installation script.
The *modules.txt* file contains all of the required Python modules.
The files.txt file contains all of the required files used by the script.

Note: Before you run the script, make sure your Pi is connected to the Internet. The script must be able to download software or it will fail.

To run the script, enter the below commands:
$ sudo chmod +x skyweather2_install.sh
$ sudo ./skyweather2_install.sh

The install script will start to run and will tell you exactly what it is doing. There is a lot of software, so the script takes a while to run. On my system, the script takes just under 15 minutes to complete.

Once the script completes successfully, you should see the below text.



```
=============================================
|   Install script completed successfully.  |
=============================================

Please reboot the Pi to enable the new settings.
pi@raspberrypi:~ $ 
```

Figure-4: Script success text

If the script errored out – do not panic. It should provide you hints to where the problem lies. Depending on the error message, you must take the corrective action and re-run the script.

Reboot your Pi before continuing on to Step-11.

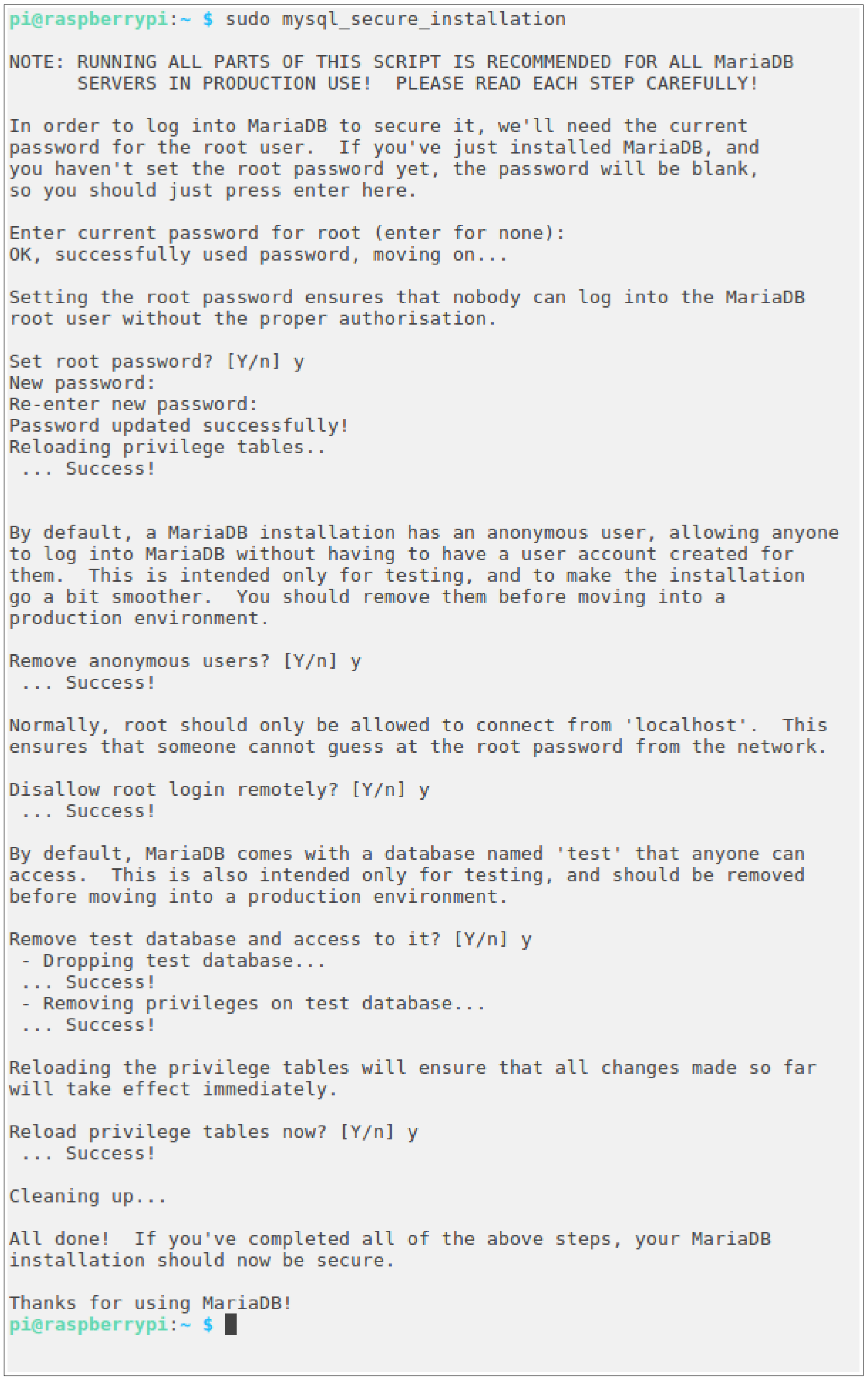
Step-11: Secure MySQL and create the SkyWeather2 databases

In this step, you will secure the base MySQL installation and create the required *SkyWeather2* databases. From a command window, run the below command:
$ *sudo mysql_secure_installation*

When prompted to: ‘Enter current password for root (enter for none):
Press <Enter>

When prompted to ‘Set root password? [Y/n]:
Enter: Y
Type: “password” twice without the quotes.

For the next three prompts:
Enter Y



```
pi@raspberrypi:~ $ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!


By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
 ... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
pi@raspberrypi:~ $ 
```

Figure-5: MySQL secure sequence

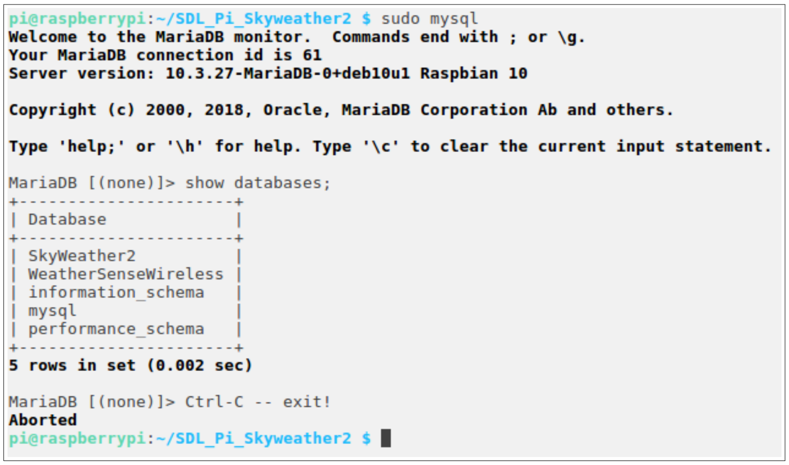
The SkyWeather2 databases were created by the install script. Let’s confirm the databases have been created.

From a command window enter:
$ *sudo mysql*
At the MariaDB prompt enter:
> show databases;

In the database list you should see five (5) databases:

- SkyWeather2
- WeatherSenseWireless
- information_schema
- mysql
- Performance_schema

Enter <Ctrl> C to exit mysql.
This sequence is shown below in Figure-9.



```
pi@raspberrypi:~/SDL_Pi_Skyweather2 $ sudo mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 61
Server version: 10.3.27-MariaDB-0+deb10u1 Raspbian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+----------------------+
| Database             |
+----------------------+
| SkyWeather2          |
| WeatherSenseWireless |
| information_schema   |
| mysql                |
| performance_schema   |
+----------------------+
5 rows in set (0.002 sec)

MariaDB [(none)]> Ctrl-C -- exit!
Aborted
pi@raspberrypi:~/SDL_Pi_Skyweather2 $ █
```

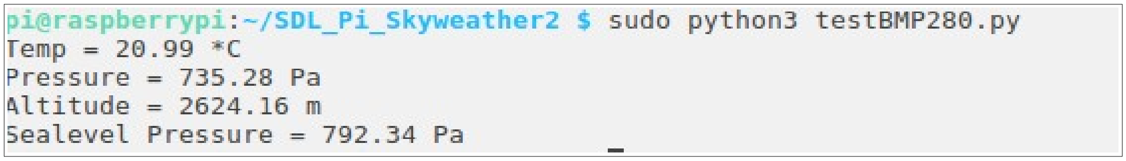
Figure-9: Database creation confirmation

You have now confirmed that the *SkyWeather2* databases have been created and is ready for use.

Step-12: Test all software components

At this point, all of the required software has been installed and the MySQL database has been created. It is important to test all the components to make sure they are working.

First, test the *BMP280* sensor that is on the *SkyWeather2* hat.
At a command prompt, enter the following three commands:
$ cd
$ cd SDL_Pi_Skyweather2
$ sudo python3 testBMP280.py
If the sensor if working correctly, you should see similar data as shown in Figure-10 below:



```
pi@raspberrypi:~/SDL_Pi_Skyweather2 $ sudo python3 testBMP280.py
Temp = 20.99 *C
Pressure = 735.28 Pa
Altitude = 2624.16 m
Sealevel Pressure = 792.34 Pa
```

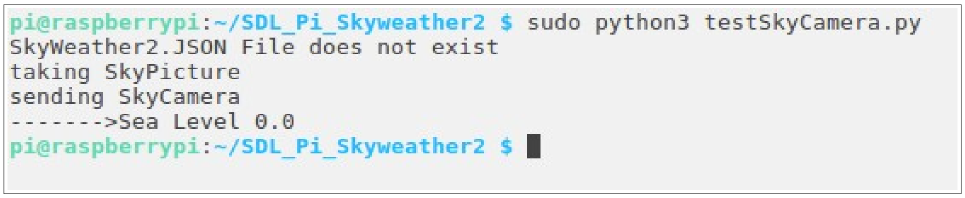
Figure-10: testBMP280.py

If you do not get the above results, then there is an issue with the *SkyWeather2* hat. Be sure you have it connected correctly on the Pi GPIO pins, and that the blue LED is illuminated.

Next, test the camera setup.
At a command prompt, enter the following command:

$ python3 testSkyCamera.py

You should see the output shown in Figure-11.



```
pi@raspberrypi:~/SDL_Pi_Skyweather2 $ sudo python3 testSkyCamera.py
SkyWeather2.JSON File does not exist
taking SkyPicture
sending SkyCamera
------->Sea Level 0.0
pi@raspberrypi:~/SDL_Pi_Skyweather2 $ 
```

Figure-11: testCamera.py

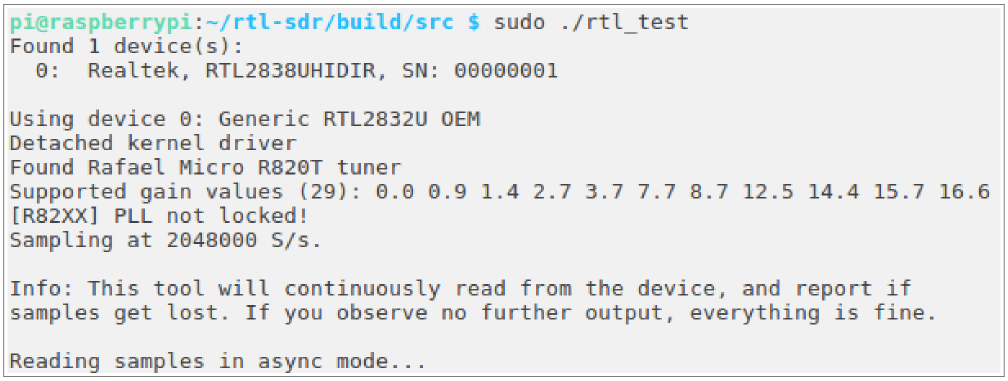
If you get an error running the test script make sure the camera is installed correctly.

You can also test the camera using the following command:
Note: This will only work if you are directly connected to the Pi via a keyboard and monitor.
$ *sudo raspistill -f -t 0*

If the camera is working, you should see a full-screen live video feed from the camera.

The next text makes sure the *rtl-sdr* software installed correctly.
From a command window type in the following three commands:
$ *cd*
$ *cd rtl-sdr/build/src*
$ *sudo ./rtl_test*

If the SDR radio receiver can be found, you should see output similar to that shown in Figure-12.



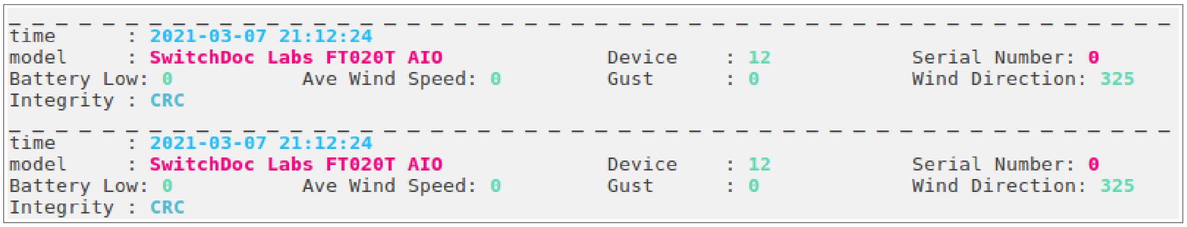
```
pi@raspberrypi:~/rtl-sdr/build/src $ sudo ./rtl_test
Found 1 device(s):
  0:  Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
Detached kernel driver
Found Rafael Micro R820T tuner
Supported gain values (29): 0.0 0.9 1.4 2.7 3.7 7.7 8.7 12.5 14.4 15.7 16.6
[R82XX] PLL not locked!
Sampling at 2048000 S/s.

Info: This tool will continuously read from the device, and report if
samples get lost. If you observe no further output, everything is fine.

Reading samples in async mode...
```

Figure-12: ./rtl_test

Press <Ctrl>c to exit the test program.
Next, we need to make sure the SDL rtl_433 software installed correctly.
From a command window type in the following three commands:
$ *cd*
$ *cd rtl_433/build/src*
$ *sudo ./rtl_433*

This command will generate a lot of output. If the software is working correctly, you should periodically see output similar that that shown below in Figure-13.

If you see similar output, then the software is receiving data from the *WeatherRack2* sensor array.

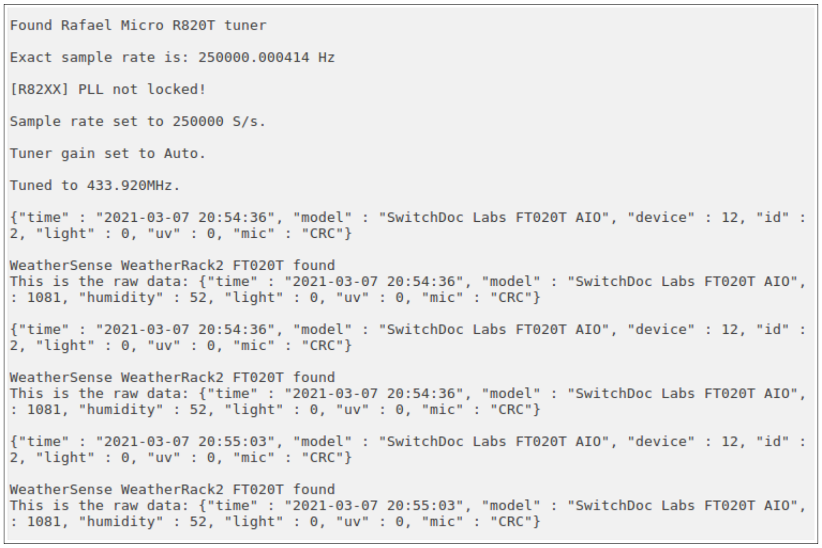


```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
time      : 2021-03-07 21:12:24
model     : SwitchDoc Labs FT020T AIO                Device     : 12              Serial Number: 0
Battery Low: 0            Ave Wind Speed: 0          Gust       : 0               Wind Direction: 325
Integrity : CRC
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
time      : 2021-03-07 21:12:24
model     : SwitchDoc Labs FT020T AIO                Device     : 12              Serial Number: 0
Battery Low: 0            Ave Wind Speed: 0          Gust       : 0               Wind Direction: 325
Integrity : CRC
```

Figure-13: ./rtl_433

Now comes the big test. This one will confirm the entire *SkyWeather2* software stack is working.
From a command window enter the following three commands:
$ *cd*
$ *cd SDL_Pi_Skyweather2*
$ *sudo python3 testWirelessSensors.py*

After a few seconds, you should see a text dump. If the radio receiver is working you should see output text similar to that shown in Figure-14 below:



```
Found Rafael Micro R820T tuner

Exact sample rate is: 250000.000414 Hz

[R82XX] PLL not locked!

Sample rate set to 250000 S/s.

Tuner gain set to Auto.

Tuned to 433.920MHz.

{"time" : "2021-03-07 20:54:36", "model" : "SwitchDoc Labs FT020T AIO", "device" : 12, "id" :
2, "light" : 0, "uv" : 0, "mic" : "CRC"}

WeatherSense WeatherRack2 FT020T found
This is the raw data: {"time" : "2021-03-07 20:54:36", "model" : "SwitchDoc Labs FT020T AIO",
: 1081, "humidity" : 52, "light" : 0, "uv" : 0, "mic" : "CRC"}

{"time" : "2021-03-07 20:54:36", "model" : "SwitchDoc Labs FT020T AIO", "device" : 12, "id" :
2, "light" : 0, "uv" : 0, "mic" : "CRC"}

WeatherSense WeatherRack2 FT020T found
This is the raw data: {"time" : "2021-03-07 20:54:36", "model" : "SwitchDoc Labs FT020T AIO",
: 1081, "humidity" : 52, "light" : 0, "uv" : 0, "mic" : "CRC"}

{"time" : "2021-03-07 20:55:03", "model" : "SwitchDoc Labs FT020T AIO", "device" : 12, "id" :
2, "light" : 0, "uv" : 0, "mic" : "CRC"}

WeatherSense WeatherRack2 FT020T found
This is the raw data: {"time" : "2021-03-07 20:55:03", "model" : "SwitchDoc Labs FT020T AIO",
: 1081, "humidity" : 52, "light" : 0, "uv" : 0, "mic" : "CRC"}
```

Figure-14: testWirelessSensors.py

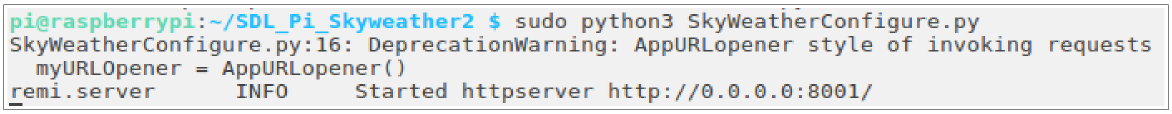
You may have to wait a minute or so to see the output results.
If you do not see the above results, then make sure the *WeatherRack2* sensor array is powered on and within receive distance, and that you have the DSR USB receiver and antenna connected to the Pi.
If you do see results similar to Figure-14, then you should have high confidence your system is working.
Press <Ctrl> c to exit the test program.

Step-13: Run the SkyWeather2 configuration script

How to configure your *SkyWeather2* system is described in detail in the SDL documentation set. Here I will only describe what you need to do to ensure you have MySQL logging turned on so you can see your data in the Dash App.

From a command window on your Pi, enter the following commands:
$ *cd*
$ *cd SDL_Pi_Skyweather2*
$ *sudo python3 SkyWeatherConfigure.py*

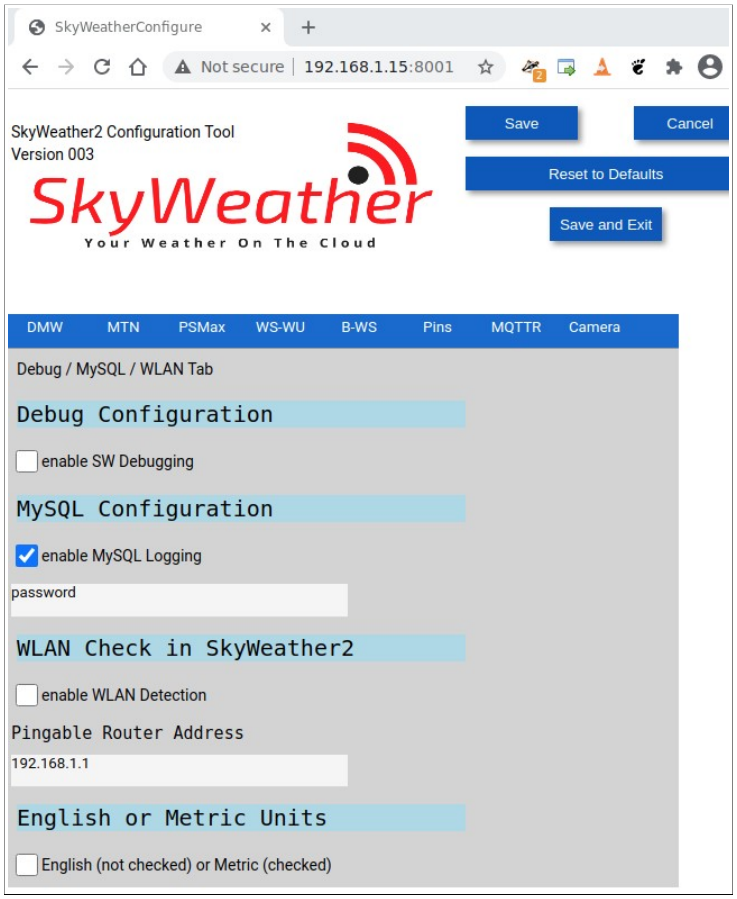
You should see the *remi* web server start on port 8001 (Figure-15).



```
pi@raspberrypi:~/SDL_Pi_Skyweather2 $ sudo python3 SkyWeatherConfigure.py
SkyWeatherConfigure.py:16: DeprecationWarning: AppURLopener style of invoking requests
  myURLOpener = AppURLopener()
remi.server       INFO      Started httpserver http://0.0.0.0:8001/
```

Figure-15: Configuration web service

From another computer or tablet, open a web browser and enter the IP address of your Pi followed by a colon and port number 8001.



SkyWeather2 Configuration Tool
Version 003

Debug / MySQL / WLAN Tab

Debug Configuration

☐ enable SW Debugging

MySQL Configuration

☑ enable MySQL Logging

password

WLAN Check in SkyWeather2

☐ enable WLAN Detection

Pingable Router Address

192.168.1.1

English or Metric Units

☐ English (not checked) or Metric (checked)

Figure-16: SkyWeather configuration tool

As shown above in Figure-16, my Pi is at address 192.168.1.15. The landing page is the Debug/MySQL/WLAN tab.

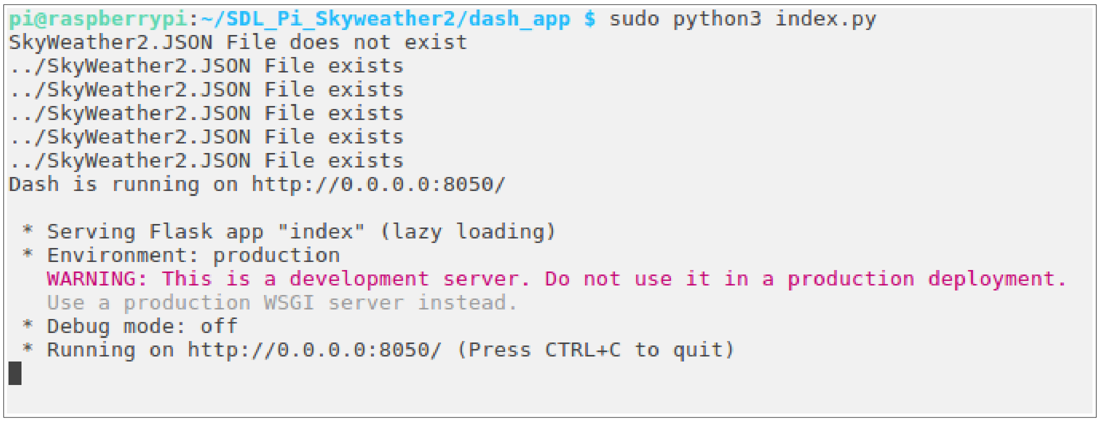
To save weather sensor data to the MySQL database, you need to check the ‘enable MySQL Logging’ checkbox and add the MySQL root password (SkyWeather2) in the edit box if it is not already there. Then click the ‘Save and Exit’ button. This will create and save the configuration JSON file and stop the *remi* web service.

NOTE: There are other configuration items you need to configure for your system to work properly. Refer to the SDL documentation for these settings.

Step-14: Run the SkyWeather2 script to collect sensor data

The SkyWeather2 system includes a handy web service (Dash App) that visualizes the collected weather data stored in the MySQL database.

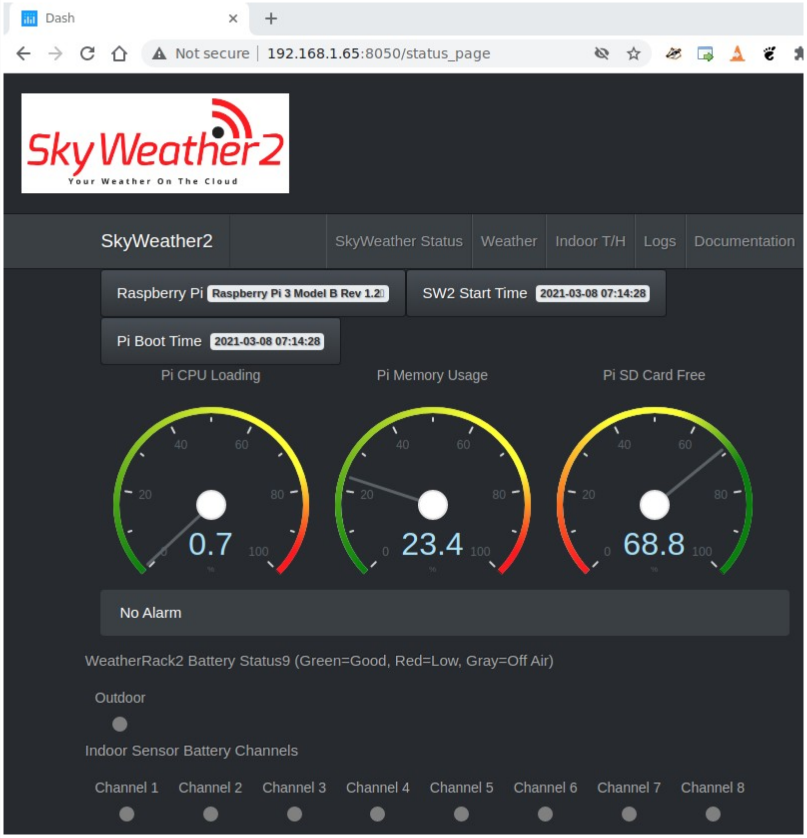
To start the service, from a command window type in the following commands:
$ *cd*
$ *cd SDL_Pi_Skyweather2/dash_app*
$ *sudo python3 index.py*



```
pi@raspberrypi:~/SDL_Pi_Skyweather2/dash_app $ sudo python3 index.py
SkyWeather2.JSON File does not exist
../SkyWeather2.JSON File exists
../SkyWeather2.JSON File exists
../SkyWeather2.JSON File exists
../SkyWeather2.JSON File exists
../SkyWeather2.JSON File exists
Dash is running on http://0.0.0.0:8050/

 * Serving Flask app "index" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:8050/ (Press CTRL+C to quit)
```

Figure-17: SkyWeather2 dash app

The web service will start on port 8050. From another computer or tablet, open a web browser and enter the IP address of your Pi followed by a colon and port number 8050 (Figure-17).



Raspberry Pi Raspberry Pi 3 Model B Rev 1.2
SW2 Start Time 2021-03-08 07:14:28
Pi Boot Time 2021-03-08 07:14:28

No Alarm

WeatherRack2 Battery Status9 (Green=Good, Red=Low, Gray=Off Air)

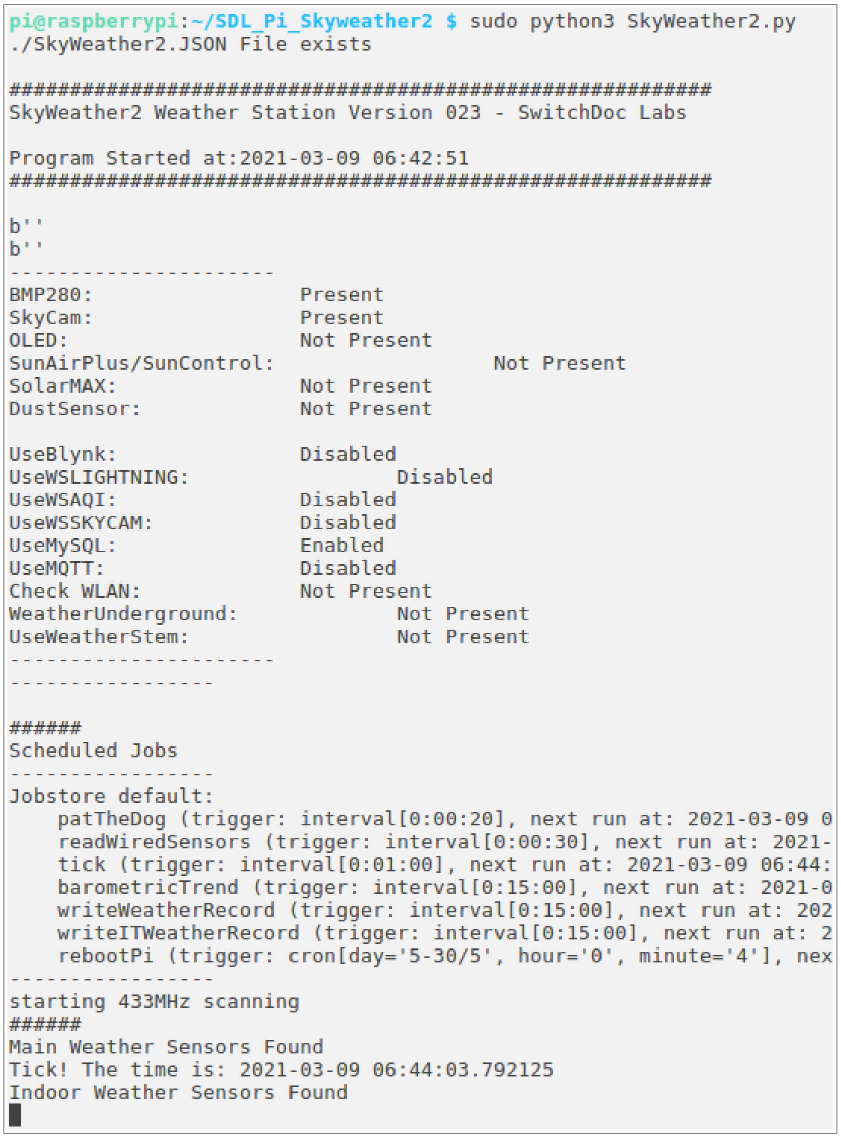
Figure-18: SkyWeather2 Dash App status page

The landing page shows the status of your *SkyWeather2* system. Notice the nice gauges and other status information about your Pi. At the bottom of the page you will notice the buttons are grayed out. This is because there is no sensor data in the MySQL database.

To fix this – we need to collect some sensor data.

From your Pi command window press <Ctrl> c to stop the *Dash App* web service.
Start the *SkyWeather2* script to start collecting data.

From a command window on your Pi type in the below commands:
$ *cd*
$ *cd SDL_Pi_Skyweather2*
$ *sudo python3 SkyWeather2.py*



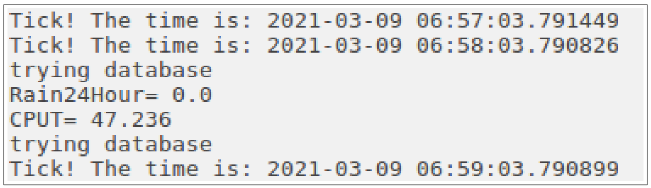
```
pi@raspberrypi:~/SDL_Pi_Skyweather2 $ sudo python3 SkyWeather2.py
./SkyWeather2.JSON File exists

##########################################################
SkyWeather2 Weather Station Version 023 - SwitchDoc Labs

Program Started at:2021-03-09 06:42:51
##########################################################

b''
b''
----------------------
BMP280:                 Present
SkyCam:                 Present
OLED:                   Not Present
SunAirPlus/SunControl:                  Not Present
SolarMAX:               Not Present
DustSensor:             Not Present

UseBlynk:               Disabled
UseWSLIGHTNING:                 Disabled
UseWSAQI:               Disabled
UseWSSKYCAM:            Disabled
UseMySQL:               Enabled
UseMQTT:                Disabled
Check WLAN:             Not Present
WeatherUnderground:             Not Present
UseWeatherStem:                 Not Present
----------------------
-----------------

######
Scheduled Jobs
-----------------
Jobstore default:
    patTheDog (trigger: interval[0:00:20], next run at: 2021-03-09 0
    readWiredSensors (trigger: interval[0:00:30], next run at: 2021-
    tick (trigger: interval[0:01:00], next run at: 2021-03-09 06:44:
    barometricTrend (trigger: interval[0:15:00], next run at: 2021-0
    writeWeatherRecord (trigger: interval[0:15:00], next run at: 202
    writeITWeatherRecord (trigger: interval[0:15:00], next run at: 2
    rebootPi (trigger: cron[day='5-30/5', hour='0', minute='4'], nex
-----------------
starting 433MHz scanning
######
Main Weather Sensors Found
Tick! The time is: 2021-03-09 06:44:03.792125
Indoor Weather Sensors Found
```

Figure-19: SkyWeather2 script

The script will start running and display the current configuration settings as shown above in Figure-19. Notice both the *WeatherRack2* and the indoor temperature devices were found.

The script will periodically collect sensor data and store it in the MySQL database. You should run this script for several hours to collect enough data for the Dash App to show trends.

As you monitor the script, you will see checkpoints when it stored the data as shown below in Figure-20.



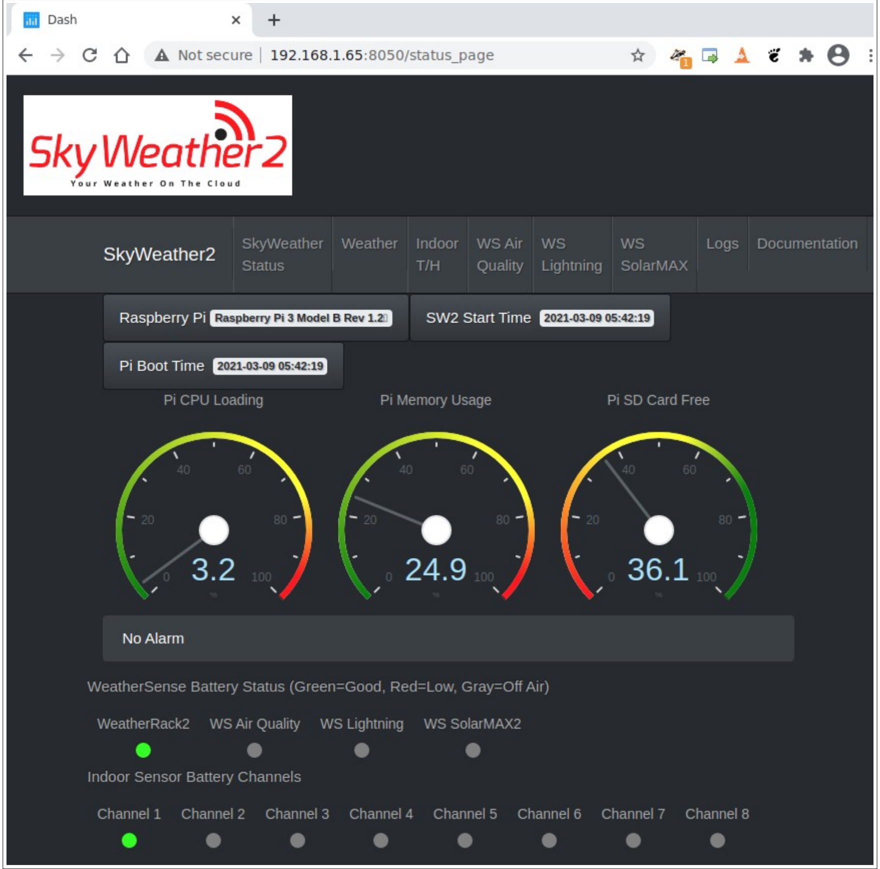
```
Tick! The time is: 2021-03-09 06:57:03.791449
Tick! The time is: 2021-03-09 06:58:03.790826
trying database
Rain24Hour= 0.0
CPUT= 47.236
trying database
Tick! The time is: 2021-03-09 06:59:03.790899
```

Figure-20: SkyWeather2 database checkpoint

Step-15: Run the SkyWeather2 Dash App to visualize your data

Once you have collected several hours of sensor data, restart the Dash App.
$ *cd*
$ *cd SDL_Pi_Skyweather2/dash_app*
$ *sudo python3 index.py*

This time when you browse to the Dash App web server you will see your collected data (Figure-21).



Raspberry Pi Raspberry Pi 3 Model B Rev 1.2
SW2 Start Time 2021-03-09 05:42:19
Pi Boot Time 2021-03-09 05:42:19

No Alarm

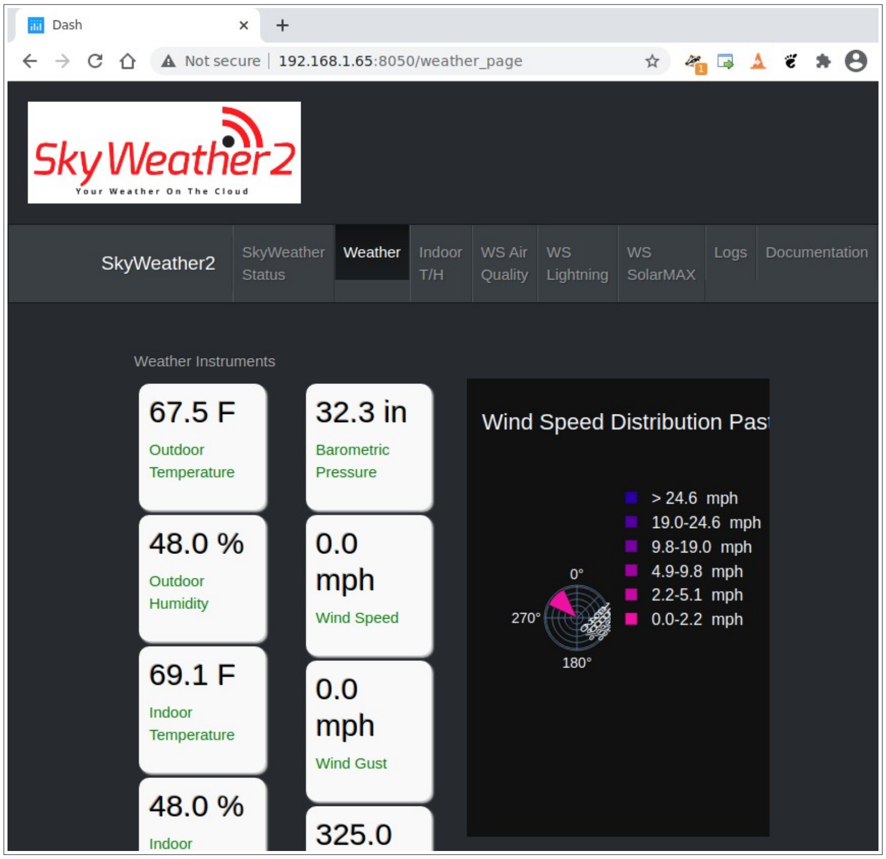
WeatherSense Battery Status (Green=Good, Red=Low, Gray=Off Air)

WeatherRack2	WS Air Quality	WS Lightning	WS SolarMAX2
🟢	●	●	●

Indoor Sensor Battery Channels

Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8
🟢	●	●	●	●	●	●	●

Figure-21: Dash App with data (1)



Weather Instruments

67.5 F Outdoor Temperature

32.3 in Barometric Pressure

48.0 % Outdoor Humidity

0.0 mph Wind Speed

69.1 F Indoor Temperature

0.0 mph Wind Gust

48.0 % Indoor

325.0

Figure-22: Dash App with data (2)

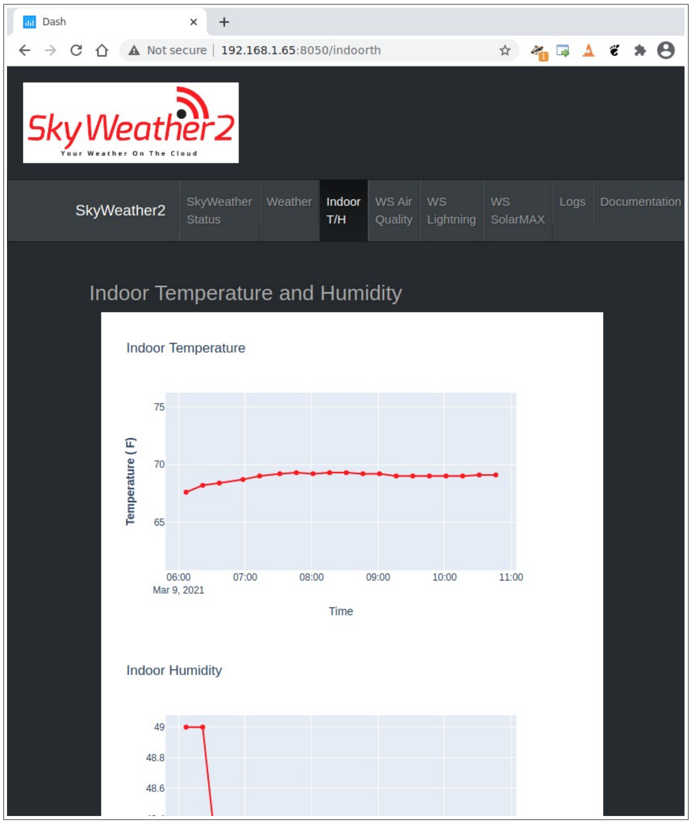


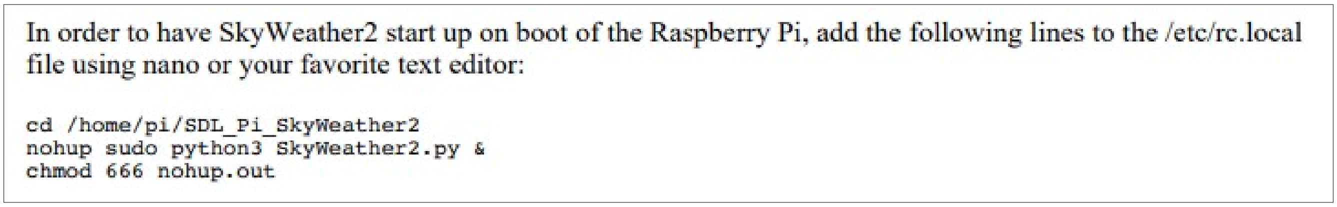
Figure-23: Dash App with data (3)

Step-16: Finalize SkyWeather2 configuration

At this point, you have confirmed that your *SkyWeather2* system is working and ready for deployment. There are two more things you need to do. Finalize the configuration and and ensure the *SkyWeather2* script runs when the Pi is booted.

All the *SkyWeather2* configuration items are described in detail in the SDL *SkyWweather2 Operations and Configuration Manual*. Refer to this document and configure your system as required.

On page 9 of the document are the instructions to make sure the *SkyWeather2* script starts at boot up (Figure-24).



In order to have SkyWeather2 start up on boot of the Raspberry Pi, add the following lines to the /etc/rc.local file using nano or your favorite text editor:

```
cd /home/pi/SDL_Pi_SkyWeather2
nohup sudo python3 SkyWeather2.py &
chmod 666 nohup.out
```

Figure-24: SkyWeather2 boot activation

Be sure to put the above three lines in your /*etc/rc.local* file.

Wrap Up

I hope you find this guide useful and that it saves you time. The *SkyWeather2* system is a very well designed, fun project to hack.

Thanks to all the fine folks at SwitchDoc Labs!

If you have issues getting your system up and running, send me an Email or post an entry on my blog. https://ismellsmoke.net

Don’t forget about the SDL support portal: https://forum.switchdoc.com/

Send corrections, comments, complaints, ideas, or any other feedback to: *sopwith@ismellsmoke.net*.

Sopwith
June 4, 2022