SwitchDoc Labs

SkyWeather
Your Weather On The Cloud

SkyWeather Assembly and Testing Guide

July 2019
Version 1.4

Table of Contents

Version 1.4 July 2019

Errata

April 28, 2019:   Version 1.2 - Fixed typos regarding nuts being identified as bolts.
June 30, 2019: Version 1.3 – Added update software instructions
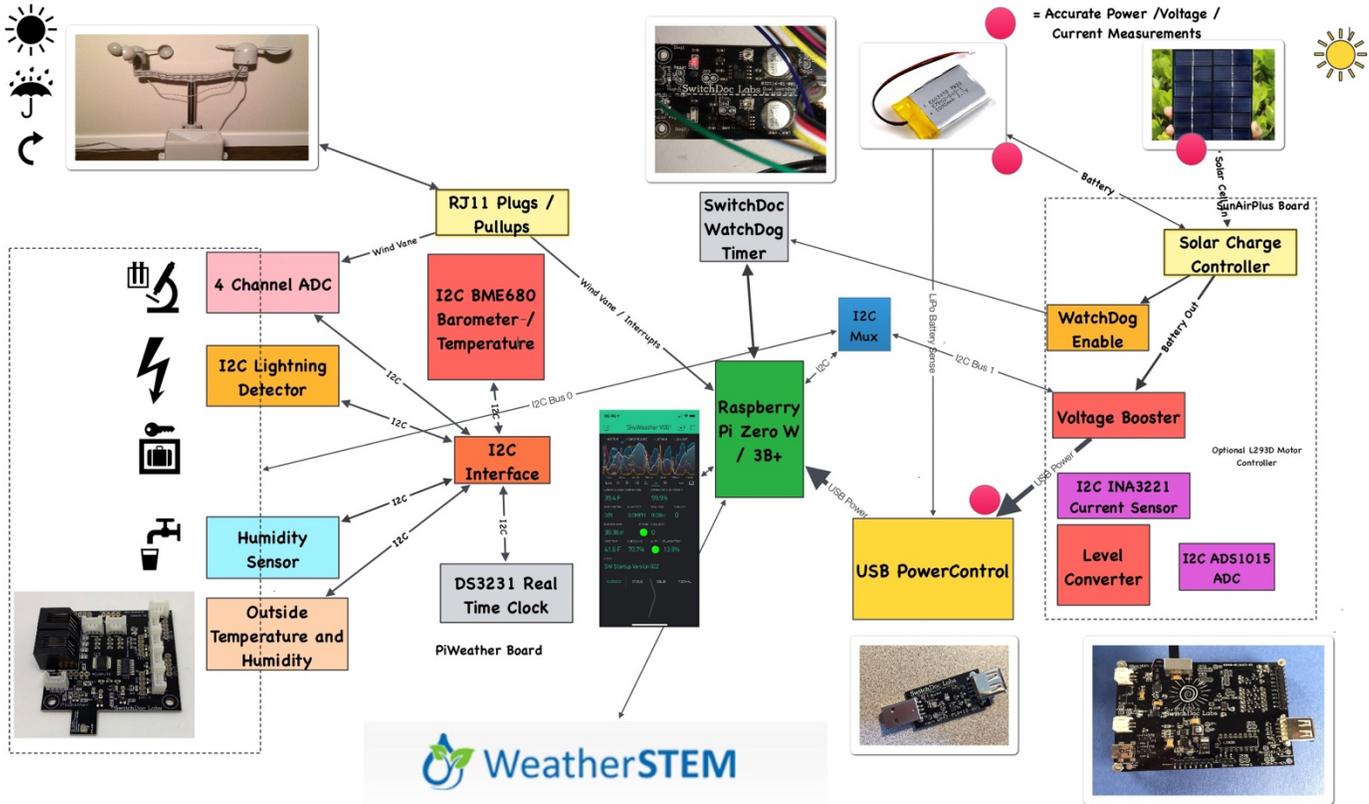July 2, 2019: Version 1.4 – Fixed typo on page 54

What is SkyWeather?

This is a perfect project kit for kids with some help from the adults and for adults trying to learn some new things.   We have done this before with our successful OurWeather KickStarter so we know what we are talking about.   People all over the world have built the OurWeather weather station with great success.   This project has **no soldering** involved and uses Grove connectors to wire everything up! You can't reverse them and blow things up.   Here is our tutorial on the Grove system.

SkyWeather Features

- Barometric Pressure
- LIGHTNING!
- Outside Temperature
- Outside Humidity
- Altitude
- Inside Temperature (in box)
- Inside Humidity (in box)
- Air Quality - AQI (your own local Air Quality Sensor)
- Sunlight
- Wind Speed
- Wind Direction
- Rain
- All your weather information on the Cloud including history

Easy to build. Easy to learn about the IOT (Internet Of Things) and the Raspberry Pi.

= Accurate Power /Voltage / Current Measurements

RJ11 Plugs / Pullups

SwitchDoc WatchDog Timer

Solar Charge Controller

SunAirPlus Board

4 Channel ADC

I2C BME680 Barometer-/ Temperature

WatchDog Enable

I2C Lightning Detector

I2C Mux

Raspberry Pi Zero W / 3B+

Voltage Booster

Humidity Sensor

I2C Interface

I2C INA3221 Current Sensor

Level Converter

I2C ADS1015 ADC

Outside Temperature and Humidity

DS3231 Real Time Clock

USB PowerControl

PiWeather Board

Optional L293D Motor Controller

Wind Vane

Wind Vane / Interrupts

I2C Bus 0

I2C Bus 1

LiPo Battery Sense

Battery

Battery Out

Solar Cells In

USB Power

I2C

I2C

WeatherSTEM

SkyWeather
Raspberry Pi On The Cloud

SwitchDoc Labs

Versions of SkyWeather

SkyWeather

The full SkyWeather kit including the Lightning and Wind Direction, Speed and Rain sensors. Includes: Rain, Wind Speed / Direction, Lighting Detection, Outside Temperature and Humidity, Barometric Pressure, Internal Temperature/Humidity, Sunlight Strength, and Outside Air Quality.

SkyWeather Lite

SkyWeather Lite does not contain the Lightning Detector and the WeatherRack wind and rain sensors. Because of that, do not drill the holes for the Lightning Detector Pylon and you do not need to have the RJ11 box connectors for the WeatherRack

SkyWeather Solar

SkyWeather Solar adds a set of solar panels on the top of the SkyWeather Box.   We have a special assembly manual for that add on to the SkyWeather kit.   Note that you have to think about where and how to orient your solar panels versus the orientation you want for your Sky Camera.   Solar Panels should generally point south (in the northern hemisphere) and north (in the southern hemisphere – right Topher?).

SkyWeather Plus Solar WXLink Remote

This SkyWeather package places the WeatherRack wind / rain sensors and the outside temperature and humidity sensor, along with a solar system in an external box connected by wireless LoRa.   See the weatherproofing manual for the WXLink Box. Basically, you place the WeatherRack and the outside temperature / humidity sensors outside and the rest of SkyWeather can either be inside or outside.   There are no wires between the WXLink remote box and the SkyWeather system.   You may still want to place the Sky Camera and SkyWeather system outside and in that case you do not need to have the RJ11 box connectors for the WeatherRack or the hole AM2315 Outside Temperature and Humidity Sensor.

This manual is for the base SkyWeather kit.

## Preparing and Learning your Raspberry Pi

The SkyWeather system requires a working Raspberry Pi. You can use virtually any not too old Raspberry Pi (2, 3, Zero, etc.) but you do need to set it up before starting the process of building SkyWeather.

Initial setting up your Raspberry Pi and connecting to it on your network is well beyond the scope of this manual.   There are just too many variables in how you might set up your Raspberry Pi.

SwitchDoc Labs provides an SD Card that has the Raspberry Pi operating system, version Stretch, and all the SkyWeather software installed.   https://shop.switchdoc.com/products/16gb-sd-card-with-stretch-smart-garden-system-groveweatherpi

Insert your SD Card (Part N from below if you have a SkyWeather Kit) into the Raspberry Pi SD Card. It goes colored face down on the Raspberry Pi 3B+ and face up on the Raspberry Pi ZeroW. The picture below shows the SD Card pluggend into a Raspberry Pi 3B+ (with the Pi2Grover board installed – Part A if you have the SkyWeather Kit).

(Default user: pi Default password: raspberry)
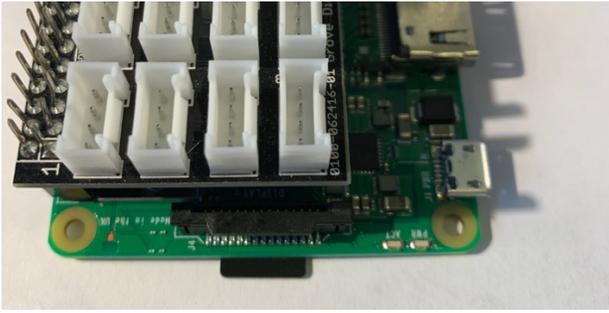
Once you have your Raspberry Pi setup up, running and can access a command line window (terminal) than you are ready to go with this manual.

This is the reason that we consider The SkyWeather Kit an advanced beginners kit, rather than a beginners kit (like the OurWeather Weather Station).

Here are some resources to get you set up and running as quickly as possible.

Helpful Getting Started Videos:
https://www.raspberrypi.org/help/videos/

Helpful Getting Started Written Tutorial:
https://www.raspberrypi.org/wp-content/uploads/2012/12/quick-start-guide-v1.1.pdf

If you want to set up a headless (no monitor, keyboard, mouse) Raspberry Pi, it is more complicated. Here are some links to tutorials for that process.

http://www.circuitbasics.com/raspberry-pi-basics-setup-without-monitor-keyboard-headless-mode/

http://blog.self.li/post/63281257339/raspberry-pi-part-1-basic-setup-without-cables

Many, many more tutorials are available on the web.

Once you have it set up, take a brief tutorial about using the terminal window and the very powerful Raspberry Pi Command Line.

https://www.raspberrypi.org/blog/learning-the-command-line/

https://www.raspberrypi.org/blog/learn-to-love-the-command-line-with-the-magpi/

You don't need to know a lot about the command line to enjoy building and running SkyWeather but you do need a bit of knowledge.
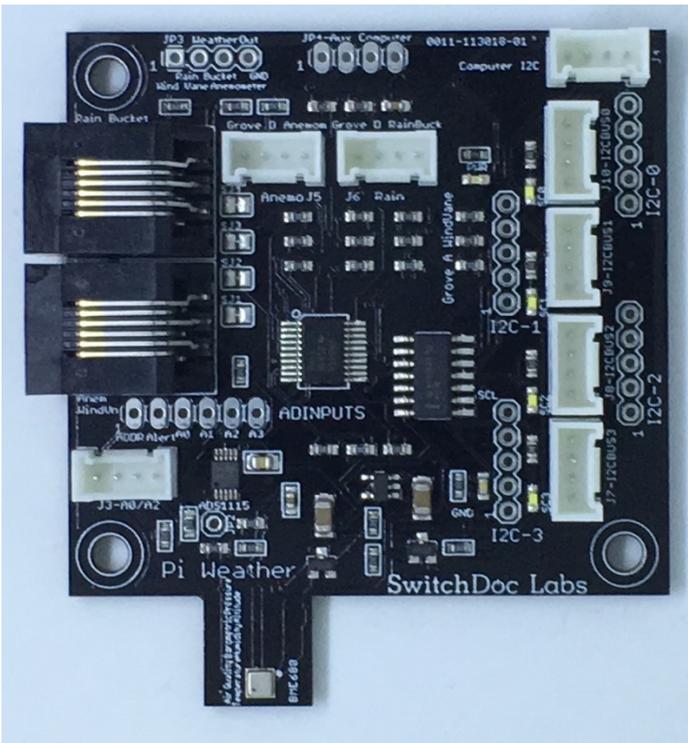
What is in the SkyWeather Box

- A Raspberry Pi is NOT included and must be purchased separately. Any Raspberry Pi with a 40 pin GPIO connector will work (such as the Raspberry Pi 2, 3 or Zero).
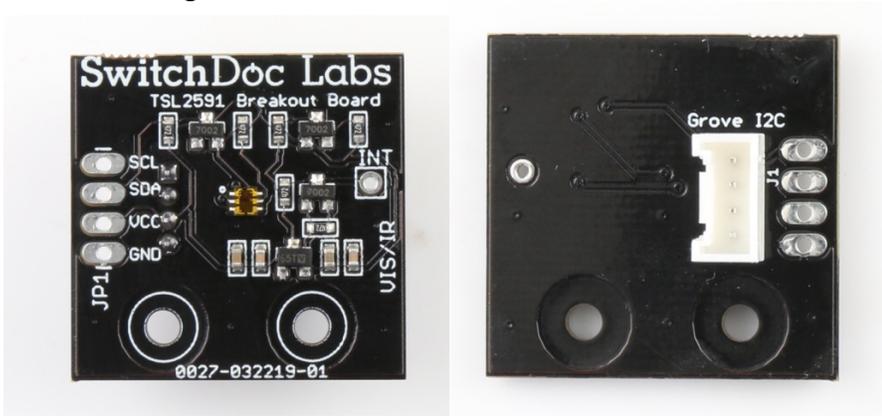
Part A – Pi2Grover

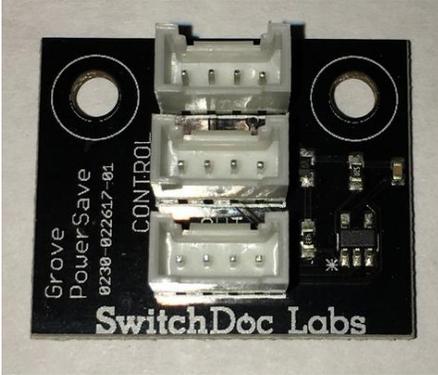Part B – PiWeather Board

Part C – Sunlight Sensor TSL2591

Part D – AM2315 Outside Temperature and Humidity Sensor

Part E – (2) Grove PowerSave for AM2315 and for Air Quality Sensor. Both parts are identical

Part F – No Part

G - Wind Direction Wind Vane

H - Bracket for Anemometer and Wind Vane
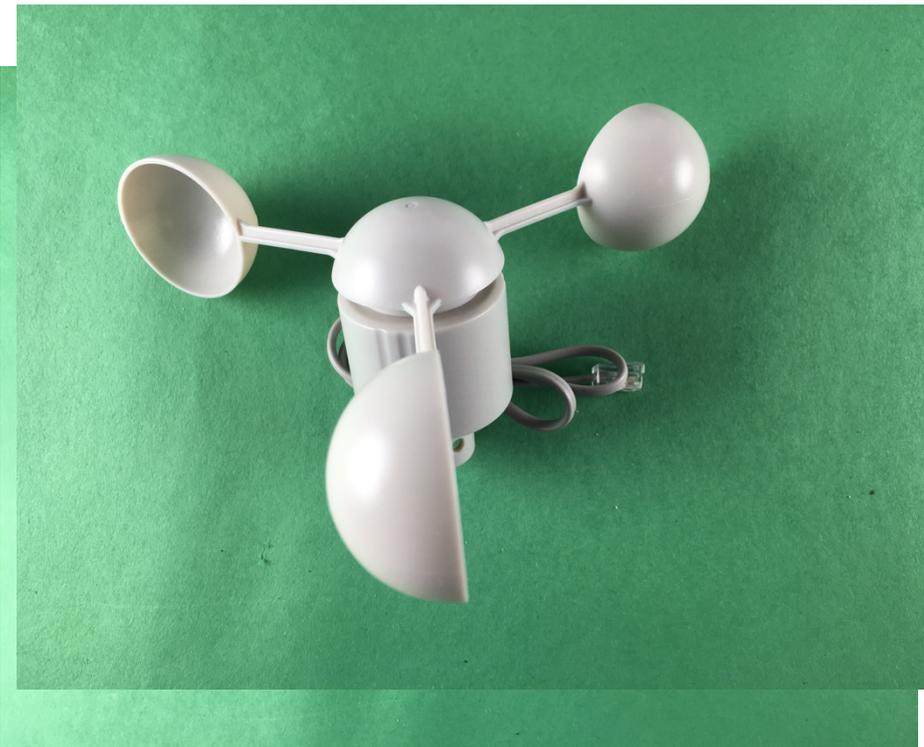
I - Rain Bucket

J - Wind Speed Anemometer
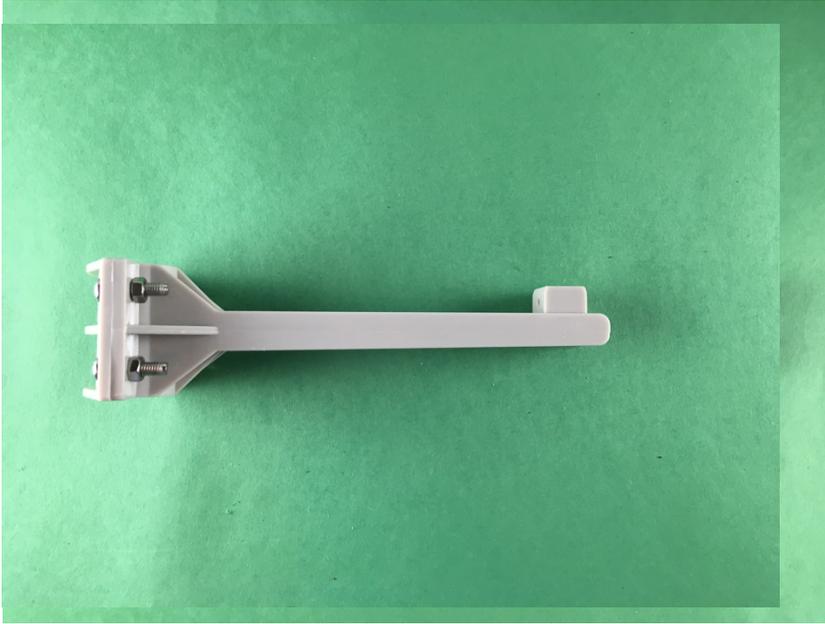
K - Rain Gauge Mounting Assembly

L - Wire Ties

M - Metal Mounting Mast for Weather Instruments

N – 16GB SD Card for Raspberry Pi
(Default user: pi Default password: raspberry)

O - Mounting Brackets for Metal Mast

P - Screws for Mounting Weather Instruments to Brackets. Contains two small self tapping screws and three screws with nuts

Part Q – SkyCamera
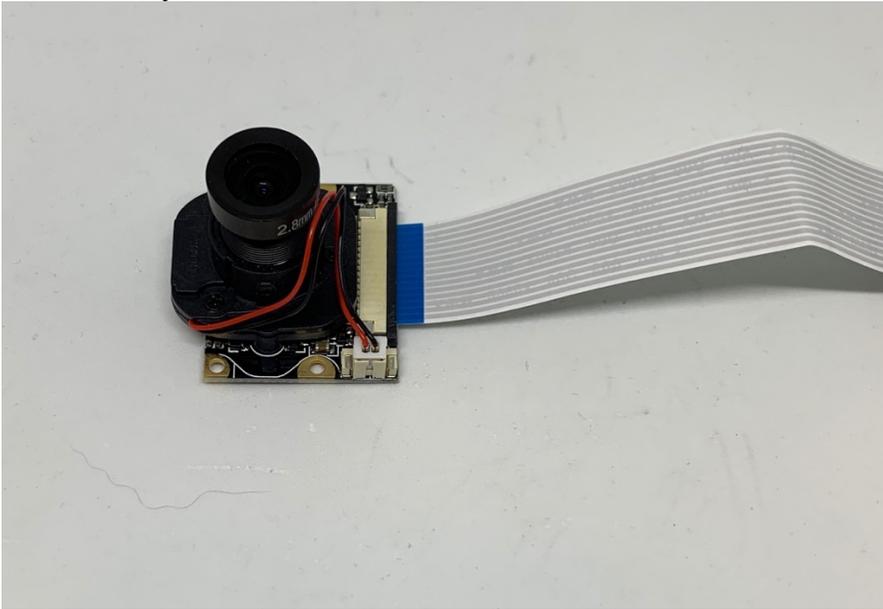
Part R – 50cm Pi3 Camera Cable

Part S – Pi Zero Camera Cable

Part T – 6 Port I2C Hub

Part U – (9) 20cm Grove Cables

Part V – 2 50cm Grove Cables
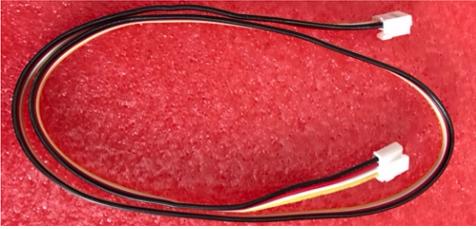
Part W – AS3935 ThunderBoard
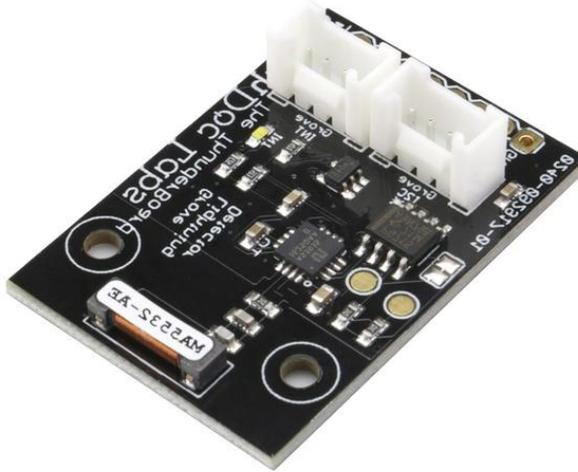
Part X – Grove Dust Sensor – Air Quality Sensor

Part Y – USB Cable – Type A USB to Micro USB

Step by Step Assembly

Setting up the Raspberry Pi

In this section, we show you how to install the Pi2Grover Raspberry Pi to Grove connector interface on your Raspberry Pi.   While these pictures show a Raspberry Pi3, the process is the same on any Raspberry Pi such as a Raspberry Pi Zero W.

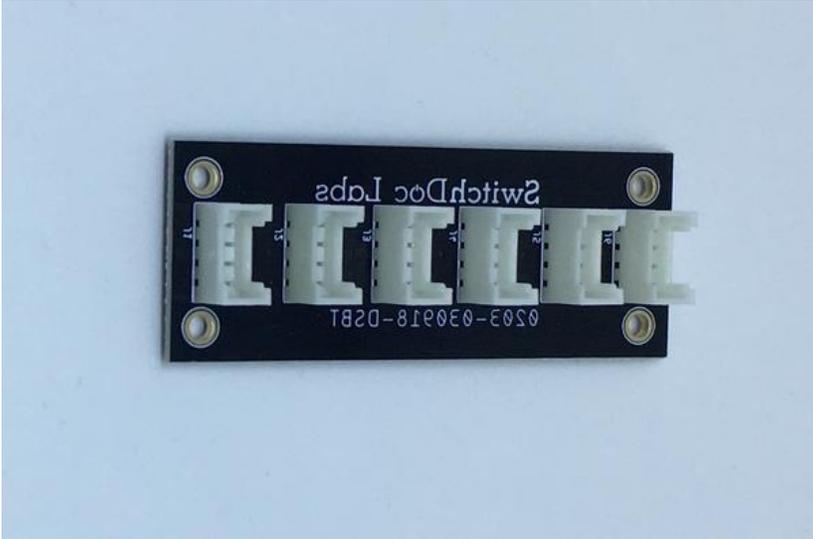**Step 1)** Connect the 50cm Raspberry Pi Camera Cable (Part R) to the Raspberry Pi.   Gently pull up on the black camera connector wings next to the HDMI connector.
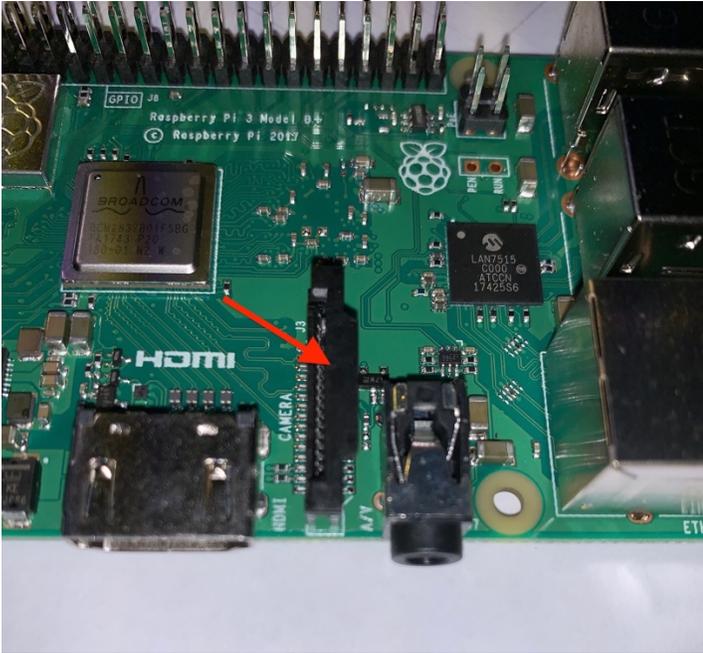
Step 2) Insert the cable (Part R) into the Raspberry Pi. The cable slots into the connector situated between the Ethernet and HDMI ports, with the silver connectors facing the HDMI port.   Push down the wings on the camera connector.
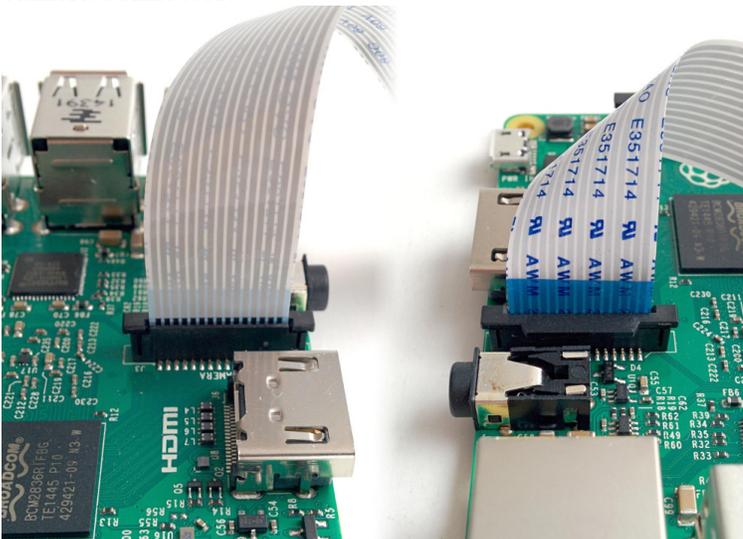
Step 3) Identify the Pi2Grover board (Part A)   (Note: Camera cable is removed for clarity. It will still be installed when you do this. Becareful with the cable and don't pinch it)

Step 4) Making sure you align the pins correctly gently press the Pi2Grover Board (Part A) onto the 40 pin GPIO connector on the Raspberry Pi

Step 5) Gently finish pushing the Pi2Grover (Part A) onto the Raspberry Pi GPIO pins, making sure the pins are aligned. There will be no pins showing on either end.

Step 6) Place your Raspberry Pi in a plastic case bottom (not included) or use pylons (not included). The picture below uses metal pylons but you may have nylon pylons.
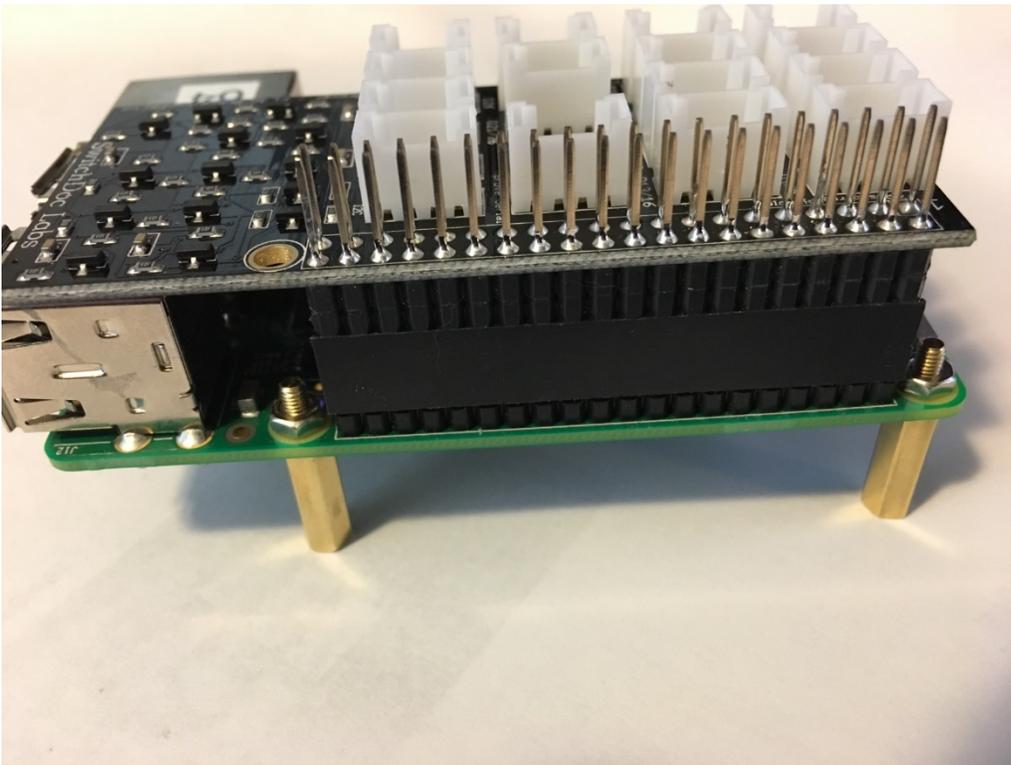
You have now completed your Raspberry Pi assembly. On to the sensors.

Step 7) Take the SkyCamera (Part Q) and install the other end of the Pi Camera Cable (Part R) into the SkyCamera connector by gently lifting the black wings, pushing in the cable and then pushing the black wings

back down into the connector.   Make sure the cable is in straight.   You can remove the camera cap at this time.

I2C Sensor Connections

What is I2C?

An I2C bus is often used to communicate with chips or sensors that are on the same board or located physically close to the CPU. It stands for standard Inter-IC device bus. I2C was first developed by Phillips (now NXP Semiconductors). To get around hardware licensing issues, sometimes the bus will be called TWI (two wire interface). SMBus, developed by Intel, is a subset of I2C that defines the protocols more strictly.

I2C provides good support for slow, close peripheral devices that only need be addressed occasionally. For example, a temperate measuring device will generally only change very slowly and so is a good candidate for the use of I2C, whereas a camera will generate lots of data quickly and potentially changes often.

I2C uses only two bidirectional open-drain lines, SCL (serial clock) and SDA (serial data). Kind of like two serial data lines next to each other. *Open-drain* means the I2C device can pull a level down to ground ("0"), but cannot pull the line up to VDD ("1"). Hence the name open-drain. You put a resistor on the line to pull it up to a "1" between "0" serial pulses, very much like a dance between SDA and SCL.

I2C devices are addressed by using a 7-bit address (0-127 in decimal) so you can have many devices on the same I2C bus, which is a very cool feature.

The Grove I2C connector has the standard layout. Pin 1 is the SCL signal and Pin 2 is the SDA signal. Power and ground are the same as the other connectors. This is another special version of the Grove digital connector. In fact, often the I2C bus on a controller (such as the ESP8266, Raspberry Pi, and the Arduino) just uses digital I/O pins to implement the I2C bus. The pins on the Raspberry Pi and Arduino are special with hardware support for the I2C bus. The ESP8266 has a purely software I2C interface, which is called "bit banging" for those children of the 90s.

Step By Step

Step 8) Take the PiWeather Board (Part B) and plug a 20cm Grove cable (Part U) into the Computer I2C Grove Connector in the top right corner of the picture below. Plug the other end of the Grove Cable into any of the I2C Grove connectors on the Pi2Grover (Part A) board.
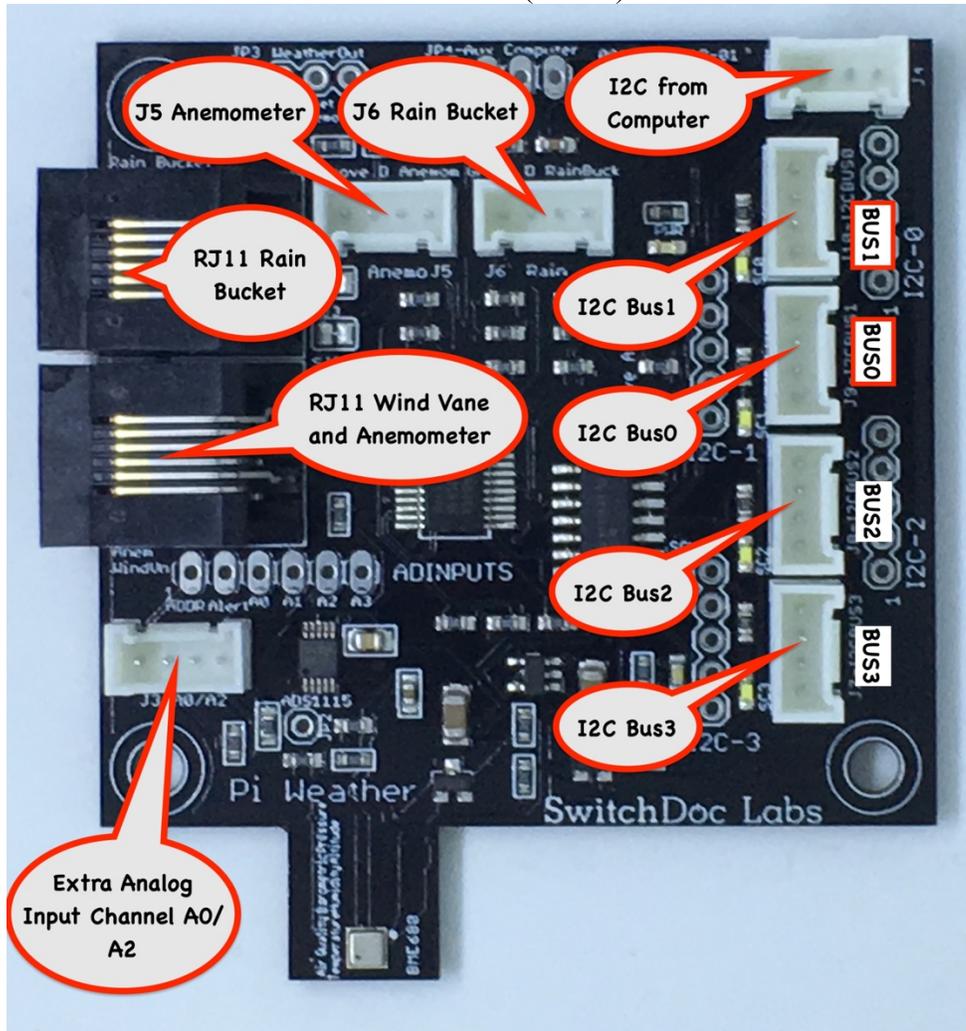
PiWeather Board (Part B)
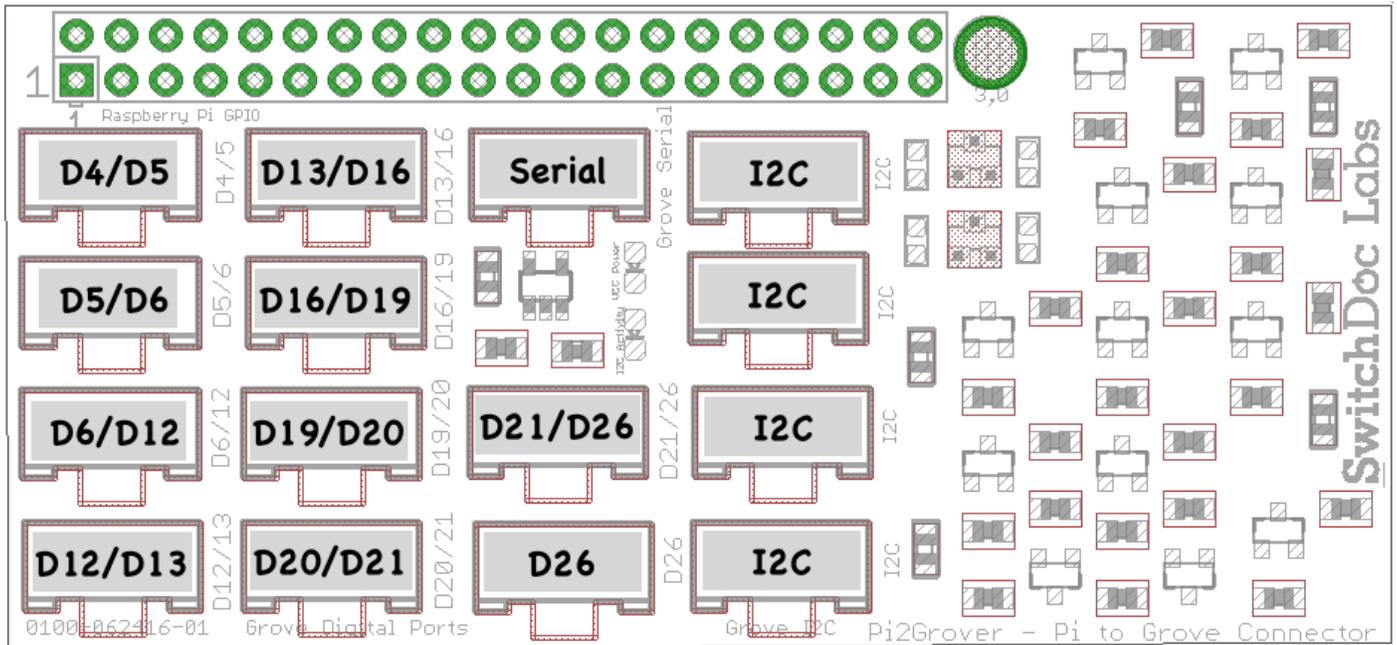
Pi2Grover Board (Part A)

The resulting connection is seen below.
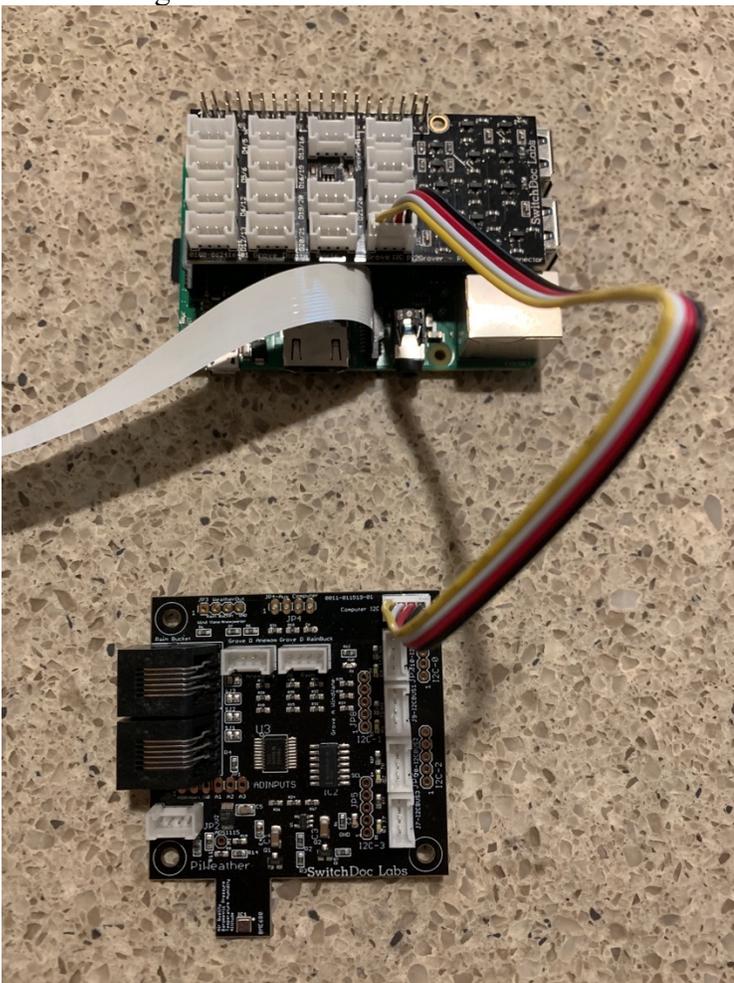
Step 9) Connect a 20cm Grove Cable (Part U) from the PiWeather Board (Part B) Bus3 to the I2C Grove Plug on the TSL2591 Sunlight Sensor (Part C)

Step 10) Connect a 20cm Grove Cable (Part U) from the PiWeather Board (Part B) Bus 0 (Check the diagram in above picture of PiWeather. The silkscreen on PiWeather boards from versioin 0011-011519-01 and earlier are reversed. The picture above is correct.) to any of the connectors on the 6 Port I2C Hub (Part T).

Step 11) Connect a 20cm Grove Cable (Part U) from any of the open connectors on the 6 Port I2C Hub (Part T) to the IN Grove Connector on the Grove PowerSave (Part E).
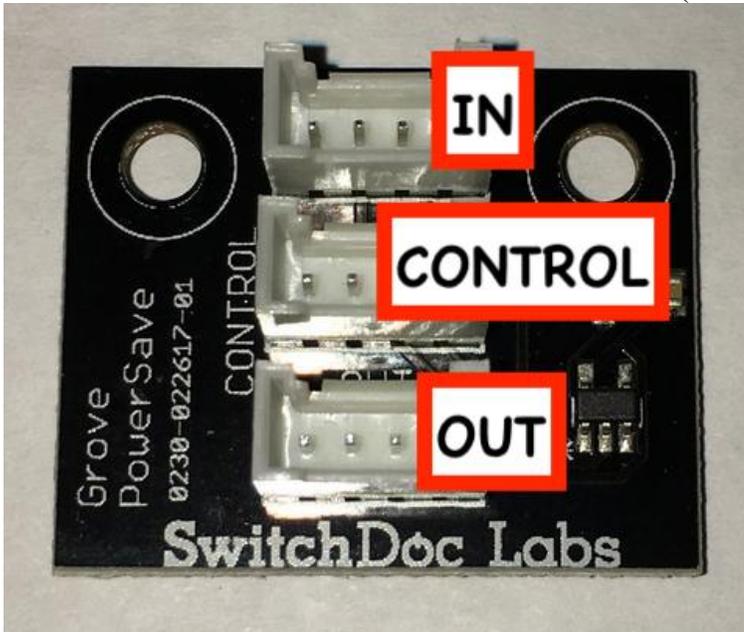
Step 12) Plug the Grove Cable (I2C) on the AM2315 Temperature/Humidity Sensor (Part D) into the OUT connector on the Grove PowerSave (Part E).

Step 13) Plug a 50cm Grove Cable (Part V) into the IGrove I2C Connector on the AS3935 ThunderBoard (Part W) and Plug the other into an unused I2C connector on the Pi2Grover (Part A).

This completes all the I2C connections in SkyWeather. Now on to the GPIO connections

GPIO Connections

What is GPIO?

A huge and powerful feature of the Raspberry Pi is the row of GPIO (general purpose input-output) pins along the top of the Raspberry Pi. It is a 40-pin header into which we can plug a large number of sensors and controllers to do amazing things to expand your Raspberry Pi.

GPIO pins can be designated (using Python software) as input or output pins and used for many different purposes. There are two 5V power pins, two 3.3V power pins and a number of ground pins that have fixed uses.
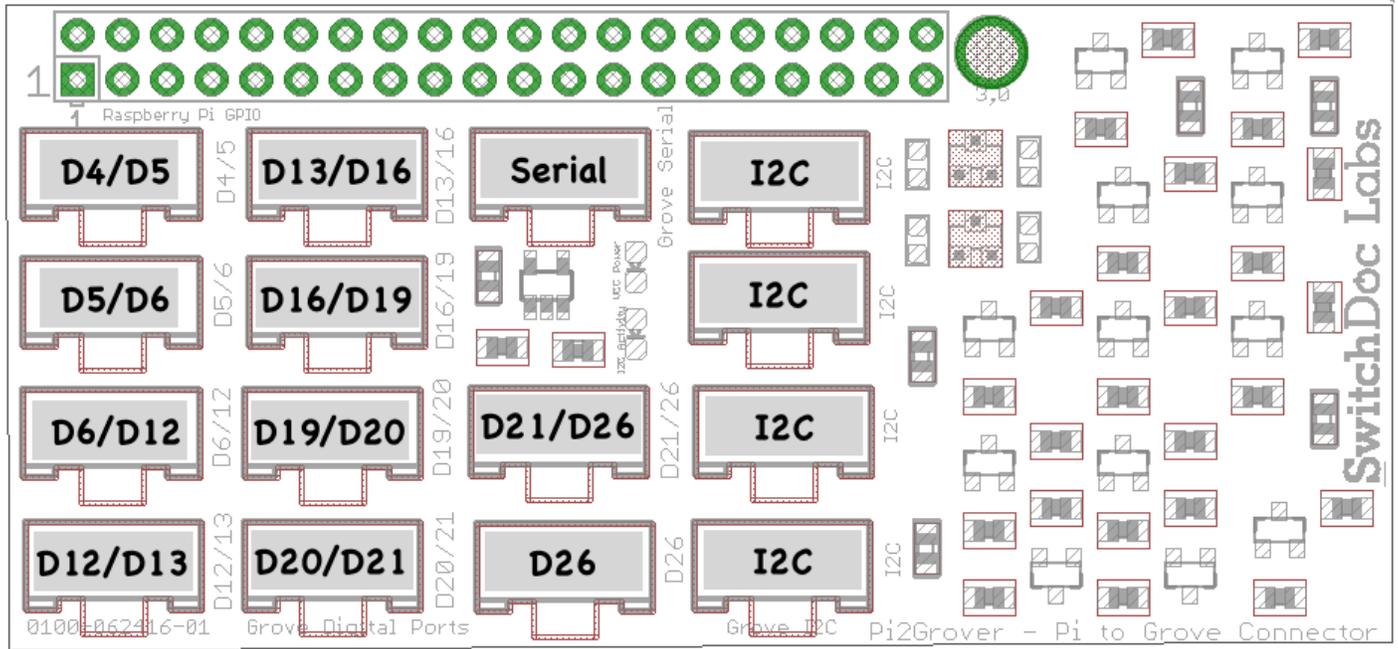
Step 14) Using a 20cm Grove Cable (Part U) Plug one end into the D6/D12 plug on the Pi2Grover Board (Part A). Plug the other end of the cable into the Control Plug on the Grove PowerSave (PartE).
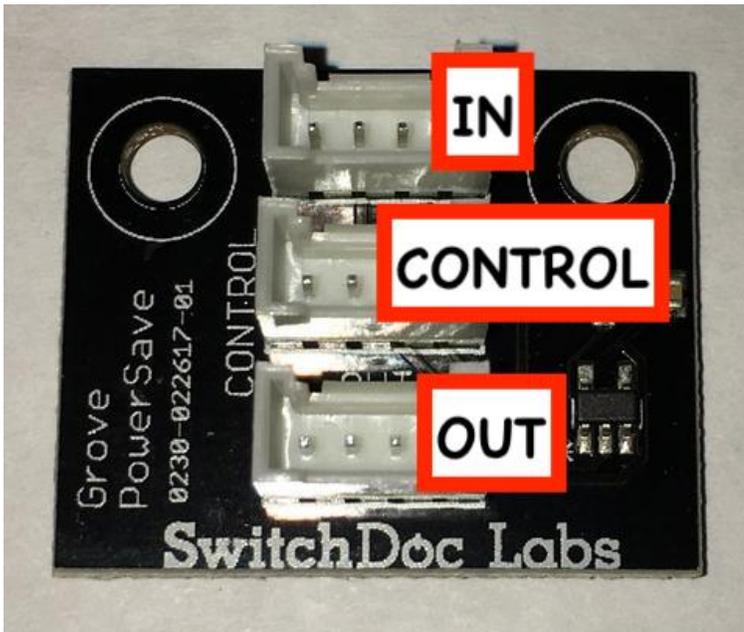
Step 15) Plug the non-Grove end of the the Air Quality Dust Sensor (Part X) into the Sensor if it is not already connected. It is keyed and only can be plugged in one direction.
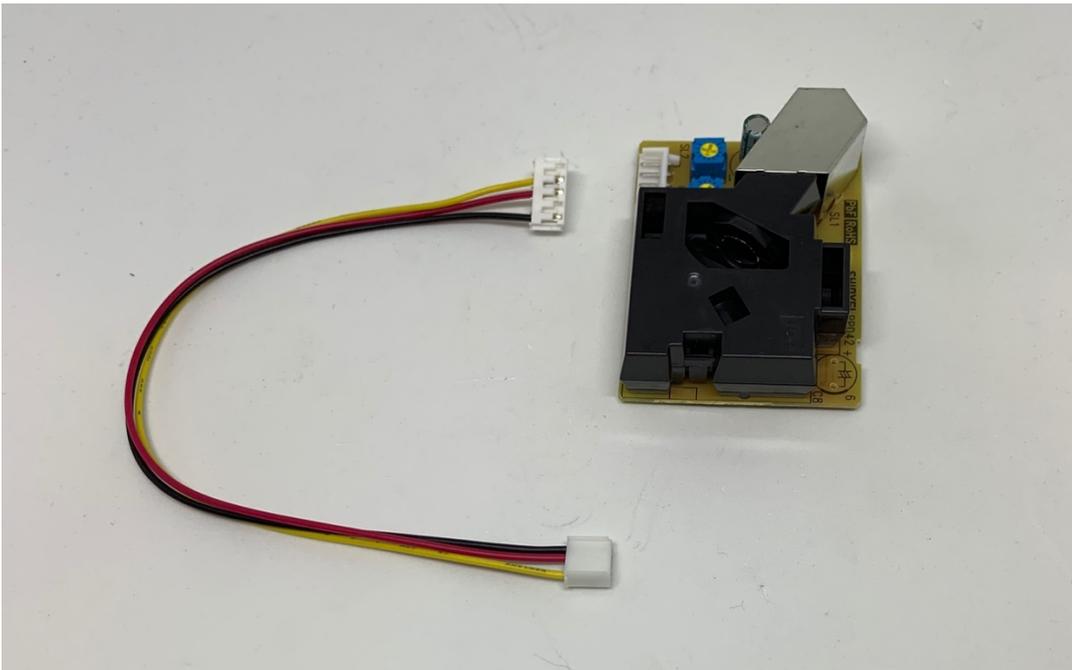
Step 16) Plug the Grove end of the Air Quality Dust Sensor Cable (Part X) (see Step 15) into the OUT Plug in the second AQI Grove PowerSave (Part E).
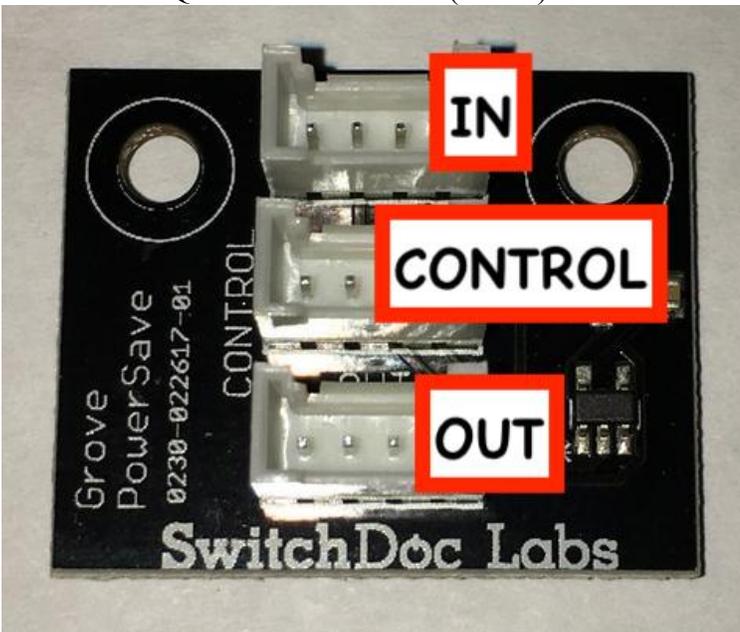
Step 17) Plug the End of a 20cm Grove Cable (Part U) in the CONTROL Input on the AQI Grove PowerSave (Part E) and plug the other end of the Grove Cable into the D26 plug on the Pi2Grover Board (Part A).
Step 18) Plug the End of a 20cm Grove Cable (Part U) in the IN Input on the AQI Grove PowerSave (Part E) and the plug the other end of the Grove Cable into the D19/D20 plug on the Pi2Grover Board (Part A).

Step 19) Plug the end of a 50cm Grove Cable (Part V) into the Grove Digital Plug (Interrupt) on the ThunderBoard Lightning Detector (Part W). Plug the other end of the cable into the D16/D19 Plug on the Pi2Grover Board (Part A).
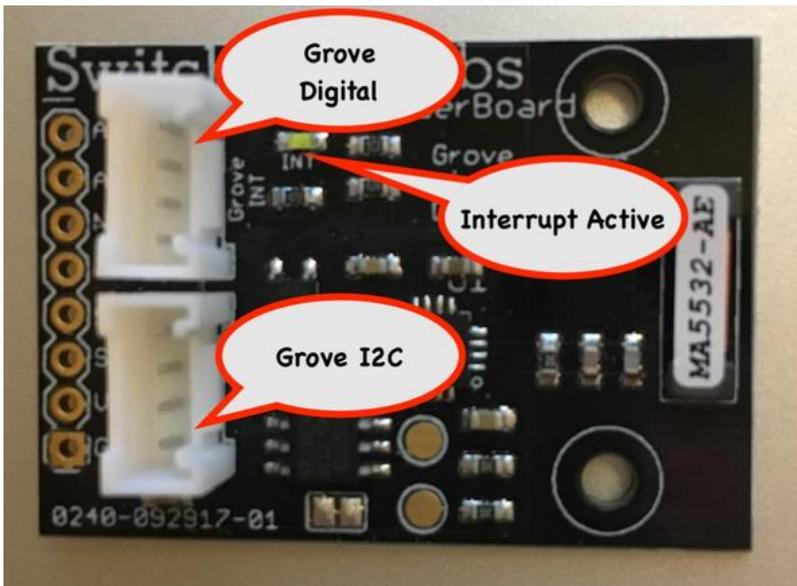
Step 20) Using a 20cm Grove Cable (Part U) plug one end into the Anemometer/J5 Input on the PiWeather Board (Part B) and the other end into the D20/D21 plug on the Pi2Grover Board (Part A).
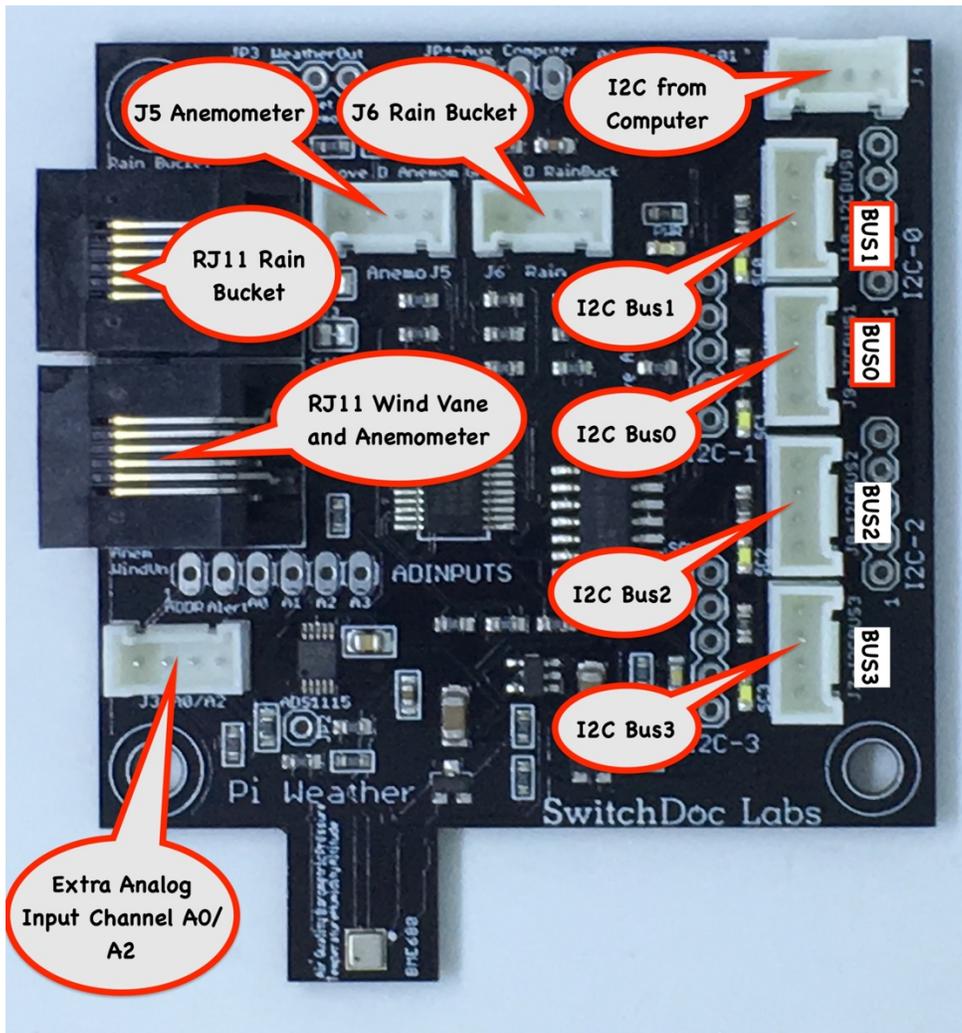
Step 21) Take a 20cm Grove Cable (Part U) and plug one end into the J6 Rain Bucket Plug on the PiWeather Board (Part B) and the other end into the D13/D16 Plug on the Pi2Grover Board (Part A).

This completes the GPIO Wiring. Next is the WeatherRack assembly.

WeatherRack Weather Instruments Assembly

Step 22) Pick up the Metal Mounting Mast (Part M) with the mounting holes drilled in the top of the mast.

Step 23) Place the Bracket for the Wind Vane and Anemometer (Part H) into the Mounting Mast (Part M).

Note the Mounting tab in the bracket and Key in the Mounting Mast and make sure they are aligned.

Step 24) Take a screw and nut (Part P), remove the nut, slide the screw through the hole in the Mounting Mast
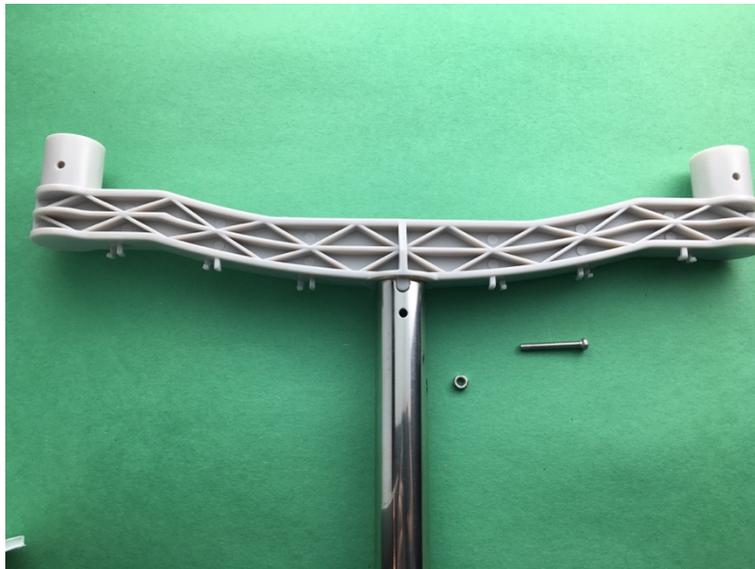
(Part M) from step and tighten the nut to the Mast.

Step 25) Take the Rain Gauge Mounting Assembly (Part K) and slide it on the Mounting Mast (Part M) to about 5 inches from the top of the Mast as shown.   Tighten the nuts to secure the Rain Gauge Mounting
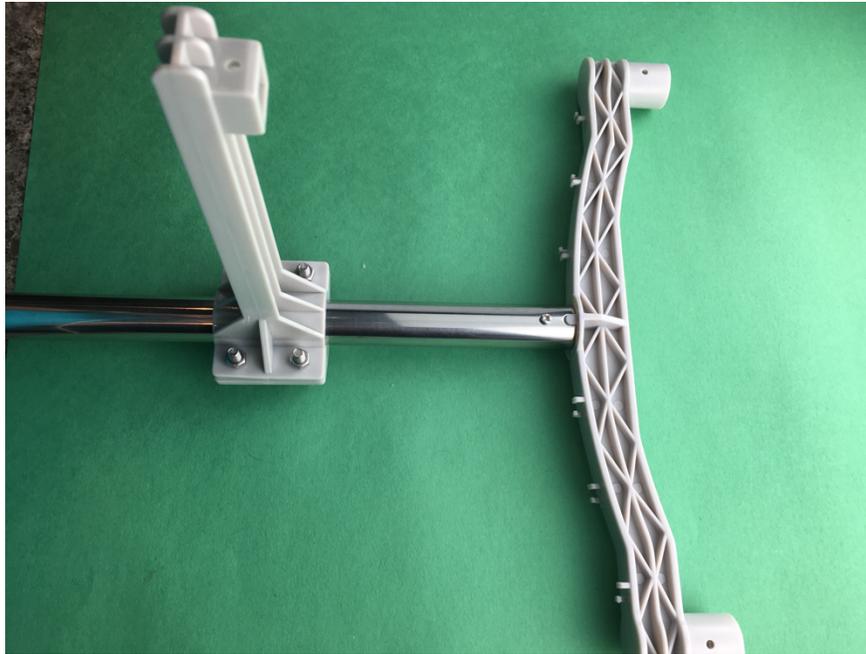
Assembly to the the Mounting Mast. Note that the Square Bracket points towards the top of the Mast.

Step 26) Take the Anemometer (Part J) and place it at the left end Wind Vane and Anemometer Bracket (Part
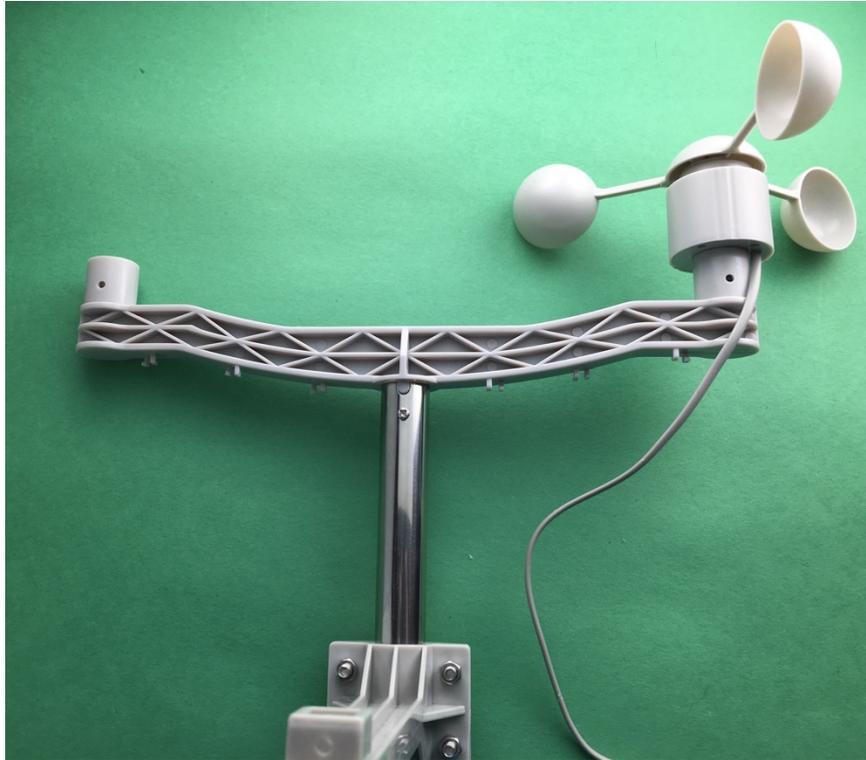
H). Note the alignment tab on the Anemometer.

Step 27) Take a screw and nut (Part P), slide it into the Wind Vane and Anemometer Bracket (Part H) under the Anemometer (Part J) and tighten in place to secure the Anemometer (Part J).

Step 28) Take the Wind Vane (Part G) and place it on the other end of the Wind Vane and Anemometer

Bracket (Part H). Note the alignment tab on the Wind Vane.   Also notice the small "N" on the plastic just under the wind vane. That is what you will align to "North" when you place the wind vane in its final place.

Step 29) Take a screw and nut (Part P), slide it into the Wind Vane and Anemometer Bracket (Part H) under

the Wind Vane (G) and tighten in place to secure the Wind Vane.

Step 30) Take the RJ45 plug on the cable from the Anemometer (Part J) and snap it into the bottom of the Wind Vane (Part G).   The plug will only fit one way and it will snap into place.

Step 31) Take the Rain Gauge (Part I) and place it on the Rain Gauge Mounting Assembly (Part K)
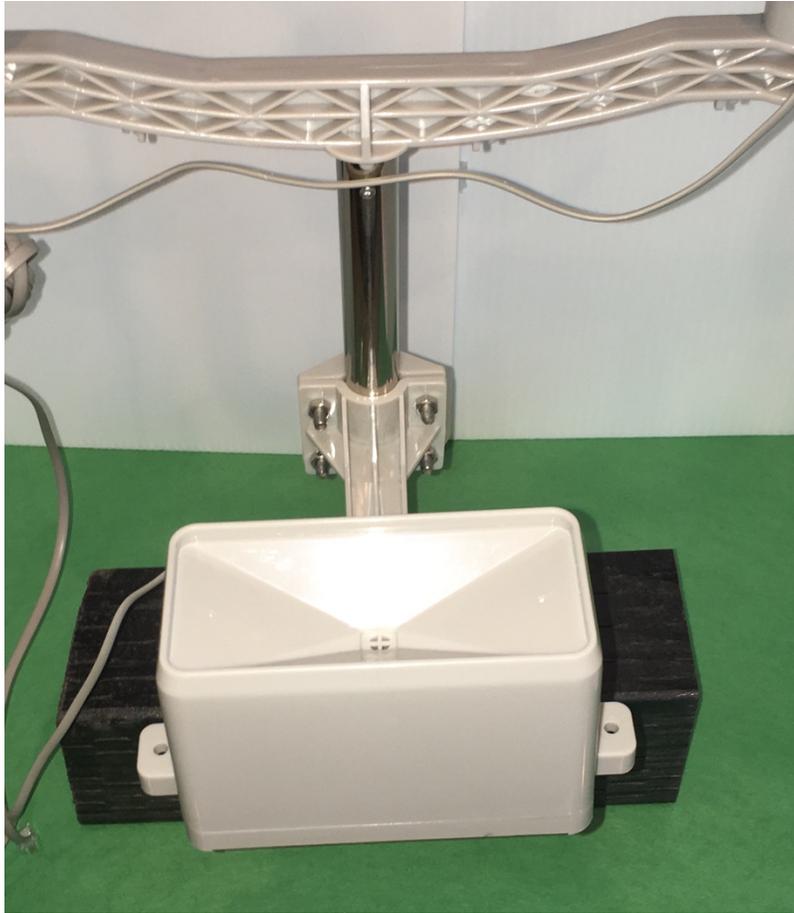
perpendicular to the Rain Gauge Mounting Assembly (Part K) as shown.

Step 32) Take a self tapping screw and screw it in the bottom of the Rain Bucket Mounting Assembly to secure the Rain Bucket.

Step 33) Push the second Mounting Mast (M) into the first Mounting Mast (M).

Step 34) Stand the Mounting Mast up and secure it with books, wood or bricks to keep it secure for testing.

Step 35) Plug the RJ54 Cable (also called an RJ11 cable) coming from the Rain Gauge (Part I) into the Plug receptacle labeled Rain Bucket on the PiWeather Board (Part B)   Note that there are two Plug receptacles on the PiWeather Board (Part B) One is labeled Rain Bucket and the other is labeled Anemometer/Wind Vane. If you plug them in the wrong position, nothing will be damaged, but the Weather sensors will not work until you plug them in correctly.
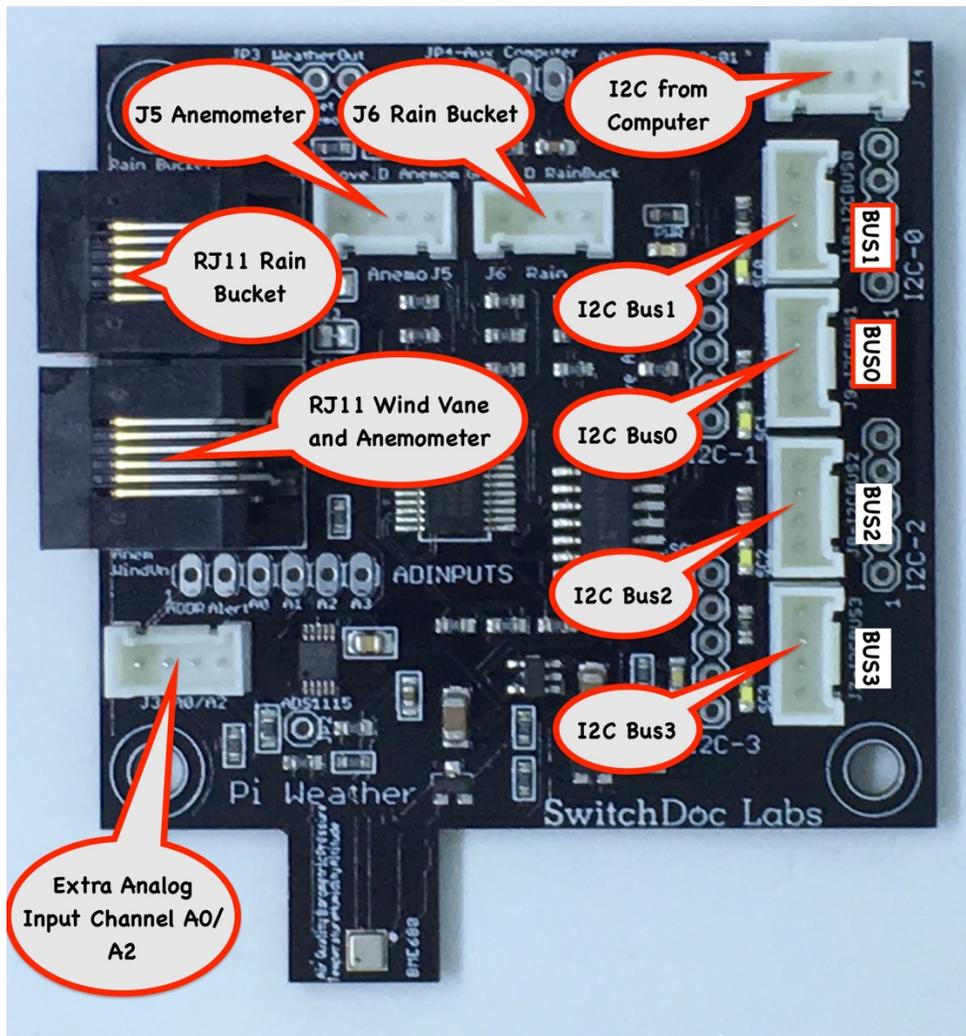
Step 36) Plug the RJ54 Cable (also called an RJ11) coming from the Wind Vane (Paert G) into the the Plug receptacle on the WeatherPlus board (Part R). One is labeled Rain Bucket and the other is labeled Anemometer/Wind Vane. If you plug them in the wrong postition, nothing will be damaged, but the Weather sensors will not work until you plug them in correctly.
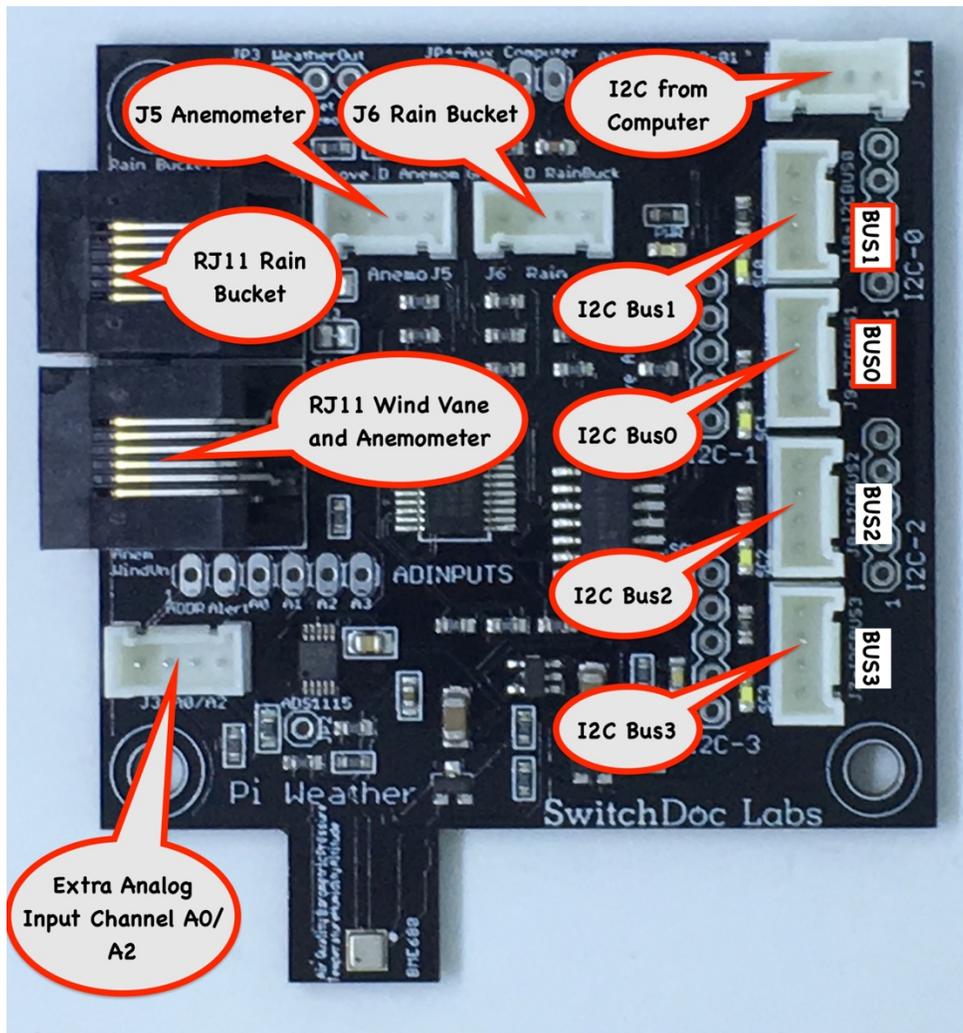
This completes the WeatherRack assembly.

Photo Courtesy of Argent Data Systems

Note that you have clamps and wire ties left over in the assembly process for the WeatherRack. You can use these to install the WeatherRack assembly outdoors and use the wire ties to secure the cables from the WeatherRack.

Final Assembly Pictures

OK, OK. This looks like a mess. It is! Don't worry, when you mount the boards in a weatherproofing box or on a piece of plastic or wood, it will look much better. One of the things that customers have requested in the past is a complete picture of the wiring. You can see that in the picture below.
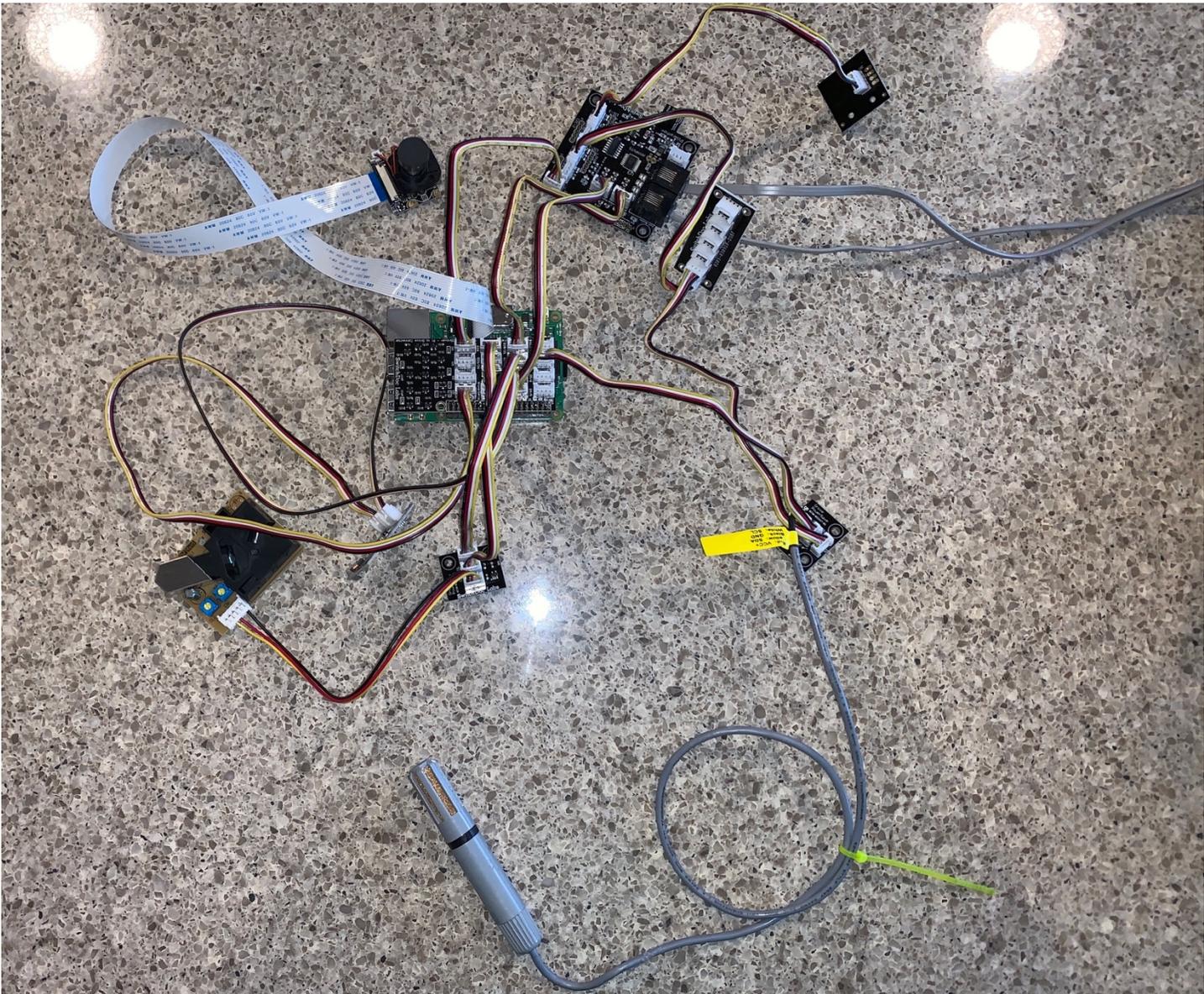
This is what your box will look like after mounting in a weatherproof enclousure:

Much better!   Proceed to testing.

Wiring Lists for SkyWeather

For those who prefer wiring lists to following step-by-step, here are the two charts. One is for I2C connections and one is for the GPIO connections. Note that you still have to hook up the camera cables!

SkyWeather I2C Chart					
Device	**Address**	**PiWeather Bus**	**I2C Plug Location**	**Description**	**Board**
TCA9545	0x73	All Buses	Built Into PiWeather	I2C 4 Channel Mux	PiWeather
BME680	0x77	Bus 0	Built into PiWeather	Barometric Pressure, AQI, Temp/Humidity	PiWeather
ADS1115	0x48	Bus 0	Built into PiWeather	ADC for Wind Vane	PiWeather
ThunderBoard	0x02	Base	ThunderBoard I2C to Pi2Grover I2C	Lightning Detection	ThunderBoard - AS3935
I2C Hub		Bus 0	PiWeather I2CBus 0 to I2C hub		
AM2315 Grove Power Save	0x5C	Bus 0	I2C Hub to AM2315 Grove Power Save In	Outside Temp/Humidity	AM2315
AM2315	0x5C	Bus 0	AM2315 to AM2315 Grove Power Save Out		
TSL2591 Sunlight	0x29	Bus 3		Visible / IR	Grove TSL2591
PiWeather Board			PiWeather Computer I2C to Pi2Grover I2C		
Optional Devices					
SunAirPlus/SunControl	0x48	Bus 2		Solar Controller	SunAirPlus/SunControl
OLED	0x3C	Bus 0		OLED Display	Grove OLED
Real Time Clock	0x68	Bus 0		Real Time Clock	DS3231 RTC

Table 1 – I2C Connections

SkyWeather GPIO Connections Chart							
Device	**From Port**	**To Port**	**Description**	**Board**	**Notes**		
Anemometer/Vane	WeatherRack	Anem Vane / RJ11	Anemometer/Wind Vane	PiWeather Board			
Rain Bucket	WeatherRack	Rain Bucket / RJ11	Rain Bucket	PiWeather Board			
Anemometer	PiWeather/J5	Pi2Grover / D20/21	Anemometer to Computer	PiWeather Board			
Rain Bucket	PiWeather/J6	Pi2Grover/ D13/D16	Rain Bucket to Computer	PiWeather Board			
AM2315 Grove PowerSave	Pi2Grover / D6/12	Grove PowerSave/ Control	Control for AM2315 Power	Grove Power Save			

AQI Grove Power Save	Pi2Grover / D19/20	AQI Grove PowerSave / IN	Input for AQI Grove Power	Grove Power Save		
AQI Grove Power Save	Pi2Grover / D26	AQI Grove PowerSave/ CONTROL	Control for AQI Grove Power	Grove Power Save		
AQI Dust Detector	AQI Grove PowerSave OUT	Dust Detector	Out for AQI Grove Power Save	Grove Dust Detector		
ThunderBoard	Pi2Grover / D16/19	ThunderBoard IRQ	Interrupt for the ThunderBoard	ThunderBoard		
WXLink	Pi2Grover / Serial	On Grove LoRa Board	connection to WXLink		Only used in SkyWeather WXLink Kit	
Optional Devices						
Fan Power	Pi2Grover / D5/6	5V Fan	Fan Control - **NOTE: Cut the White wire of this Grove Cable**	Input Grove PowerDrive	Only if you are using an optional 5V Fan for thermal control	
PixelStrips	Pi2Grover / D21/26	Pixel Sticks In			If you are using the optional Pixel Sticks with SkyWeather	
WatchDog Wiring						
USB Control	SunControl USB Control - J2/1 (Yellow)	SunControl - JP7 - WatchDog Out - JP7/1 RESETN			Only Used in SkyWeather Solar Kits	
USB Enable	SunControl USB Enable - J2/2 (White)	SunControl - JP7 - WatchDog Out - JP7/2 PULSEHIGH			Only Used in SkyWeather Solar Kits	
WatchDog Pat	Pi2Grover / D4/5	SunControl - Grove J4 - WatchDog Done / Wake Plug	**NOTE: Cut the White wire of this Grove Cable**		Only Used in SkyWeather Solar Kits	

Table 2 – GPIO Connections

SkyWeather Initial Testing

We are going to do the initial testing for the SkyWeather project now (with all of this assembly done but before you build your WeatherProofing box).   You will remove all the wires before the weather proofing assembly and then rewire the box according to the instructions above.   Makes debugging easier!

1) Before you start your testing you should have your Raspberry Pi setup as earlier. If you haven't done that yet, unplug the Pi2Grover board (Part A), setup your Raspberry Pi and then plug the Pi2Grover board very carefully back into the Raspberry Pi.

Before you power up your Raspberry Pi with all of SkyWeather devices plugged in, make sure you separate the vaious parts so none of them are touching.

Power Up your Raspberry Pi

Using the supplied USB Type A to USB Micro cable (or one of your own), plug in your Raspberry Pi and open up a terminal window (if you are using a video monitor) or ssh into your computer. Login to your computer (Default user: pi Default password: raspberry) and open a terminal window.

Change directory to SDL_Pi_SkyWeather

```
cd SDL_Pi_SkyWeather
```

Test 0 - Update your SkyWeather Software

The first thing to do is make sure you have your SkyWeather software updated to the latest version.

Type:

```
git pull
```

and it will update the software. FYI, the software is located on:

https://github.com/switchdoclabs/SDL_Pi_SkyWeather

Test 1 – Verify I2C Devices

The first set of tests is to make sure all our I2C devices are connected and are running.

Test 1A – Verify Detectable I2C devices

Type:

```
sudo python testSDL_Pi_TCA9545.py
```

Here is what you should see:

```
pi@switchdoclabs:~/SDL_Pi_SkyWeather $ sudo python testSDL_Pi_TCA9545.py

Test SDL_Pi_TCA9545 Version 1.0 - SwitchDoc Labs

Sample uses 0x73
Program Started at:2019-04-27 11:34:57

-----------BUS 0-------------------
tca9545 control register B3-B0 = 0x1
ignore Interrupts if INT3' - INT0' not connected
tca9545 control register Interrupts = 0x0
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

```
40: -- -- -- -- -- -- -- -- 48 -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- 73 -- -- -- 77

----------------------------------

-----------BUS 1-------------------
tca9545 control register B3-B0 = 0x2
ignore Interrupts if INT3' - INT0' not connected
tca9545 control register Interrupts = 0x0
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- 73 -- -- -- --

----------------------------------

-----------BUS 2-------------------
tca9545 control register B3-B0 = 0x4
ignore Interrupts if INT3' - INT0' not connected
tca9545 control register Interrupts = 0x0
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- 73 -- -- -- --

----------------------------------

-----------BUS 3-------------------
tca9545 control register B3-B0 = 0x8
ignore Interrupts if INT3' - INT0' not connected
tca9545 control register Interrupts = 0x0
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- 29 -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- 73 -- -- -- --

----------------------------------
```

Referring to the I2C table (Table 2) above shows you what these I2C devices are.   If you don't see one of these devices, check your wiring very very carefully.

There are two I2C devices that do not show up on this table and have to be tested separately: the AM2315 Temperature / Humidity sensor and the ThunderBoard Lightning Detector.

Test 1-B - Testing the AM2315

The AM2315 does not respond to the test above because of the unique way the AM2315 works. The AM2315 powers down between measurements to avoid self heating of the sensor. You first hit it with an I2C command to wake it up, pause a bit and then you can read the temperatures. You can simulate this by hitting it with two "i2cdetect -y 1" commands quickly on the command line. Then you will see it show up on 0x5C. We decided to write a test program to find and verify the functionality.

Type:

```
sudo python testAM2315.py
```

You will see something like the following. Hit "control-c" to terminate.

```
pi@switchdoclabs:~/SDL_Pi_SkyWeather $ sudo python testAM2315.py
./SDL_Pi_AM2315/AM2315.py:35: RuntimeWarning: This channel is already in use, continuing anyway.
Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(self.powerpin, GPIO.OUT)
AM2315temperature= 20.8
AM2315humdity= 33.4
AM2315crc= 40849
AM2315c= 40849
temperature: 20.8
humidity: 33.4
crc: 40849

AM2315temperature= 20.9
AM2315humdity= 33.4
AM2315crc= 24400
AM2315c= 24400
temperature: 20.9
humidity: 33.4
crc: 24400

^CTraceback (most recent call last):
  File "testAM2315.py", line 51, in <module>
    outsideHumidity, outsideTemperature, crc_check = am2315.read_humidity_temperature_crc()
  File "./SDL_Pi_AM2315/AM2315.py", line 216, in read_humidity_temperature_crc
    self._read_data()
  File "./SDL_Pi_AM2315/AM2315.py", line 118, in _read_data
    time.sleep(0.09)
KeyboardInterrupt
```

You may or may not see the first warning depending on what you have been doing with the Raspberry Pi to this point. You can ignore the warning in either case.

If you don't see the AM2315, go back and check your wiring very, very carefully. Especially the wiring for the Grove PowerSave.

Test 1C - Test AS3935 ThunderBoard Lightning Detector

The AS3935 ThunderBoard does not show up on the above tests because it is connected to address 0x02 which is not scanned for devices by the standard software. Because the AS3935 is very sensitive to I2C bus loading, the device sits on the Base I2C bus (plugged directly into the Pi2Grover) and when used, we switch the I2C Mux to a lightly loaded bus to be able to reliably read the AS3935 Lightning Detector. To test this device type:

```
sudo python testAS3935.py
```

```
pi@switchdoclabs:~/SDL_Pi_SkyWeather $ sudo python testAS3935.py
```

```
AS3935 detected at I2C port 0x02
Waiting for lightning - or at least something that looks like it
Noise level too high — adjusting
```

The above test will be different for you due to your electromagnetic environment. Move the device away from monitors and cell phones to give the chip a chance to really detect lightning. The above results shows that the ThunderBoard is present and working.   You can try moving the Thunderboard closer to your monitor or cell phone to see if it picks up noise. If you want to do more testing with simulated lightning, you can check out the ThunderBoard Lightning Simulator here:

https://shop.switchdoc.com/products/the-thunder-board-i2c-lightning-simulator

This completes testing of the I2C devices. Now we proceed to initial SkyWeather testing that will show us the rest of the devices.

Test 2 – Testing Specific GPIO Connected Devices (and the WeatherRack)

Test 2A – Testing the Dust Sensor

Because of the nature of the Grove AQI Dust sensor, we need to run a specific test to make sure it is hooked up correctly and functioning. Type:

```
sudo pigpiod
sudo python testDust.py
```

You will see something like this (after about 30 seconds for warmup time and then wait for 30 seconds between readings):

```
pi@switchdoclabs:~/SDL_Pi_SkyWeather $ sudo python testDust.py
Concentration Error

Air Quality Measurements for PM2.5:
  5288 particles/0.01ft^3
  8 ugm^3
  Current AQI (not 24 hour avg): 34

Air Quality Measurements for PM2.5:
  19673 particles/0.01ft^3
  30 ugm^3
  Current AQI (not 24 hour avg): 90

^CTraceback (most recent call last):
  File "testDust.py", line 27, in <module>
    time.sleep(30)
KeyboardInterrupt
```

Your output will vary depending on your initial environment, but if you see something similar, your device is working.

Test 2B – The TSL2591 Sunlight Sensor Test

This will test the TSL2591 sensor. Type:

```
sudo python testTSL2591.py
```

You should see something like this:

```
pi@switchdoclabs:~/SDL_Pi_SkyWeather $ sudo python testTSL2591.py
(41.4528, 20, 6)
()
Lux = 29.212800   full = 17   ir = 6
integration time = 0
gain = 0

Lux = 31.824000   full = 32   ir = 10
integration time = 1
gain = 0

Lux = 27.744000   full = 45   ir = 15
integration time = 2
gain = 0

Lux = 25.336800   full = 56   ir = 19
integration time = 3
gain = 0

Lux = 24.414720   full = 66   ir = 22
integration time = 4
gain = 0

Lux = 23.609600   full = 79   ir = 27
integration time = 5
gain = 0

Lux = 26.275200   full = 13   ir = 4
integration time = 0
```
gain = 0
```
Lux = 22.293120   full = 317   ir = 110
integration time = 0
gain = 16

Lux = 22.707297   full = 5521   ir = 1914
integration time = 0
gain = 32

Lux = -4.226146   full = 37888   ir = 37888
integration time = 0
gain = 48
```

Test 2B – Testing the WeatherRack Sensors

This allows us to test the Wind Vane, the Anemometer and Rain Bucket.

Type:

sudo python testWeatherRack.py

You will see something like the following.   While this is running, turn the WeatherVane (you will see the direction change), spin the Anemometer (you will see the wind speed change) and also tilt the Rain Bucket back and forth (you will hear clicking and then see rain fall amounts increase).

You can terminate the test with a "control-c" or "control-|".

```
pi@switchdoclabs:~/SDL_Pi_SkyWeather $ vi testWeatherRack.py
pi@switchdoclabs:~/SDL_Pi_SkyWeather $ sudo python testWeatherRack.py
```

```
  File "testWeatherRack.py", line 61
    except:
          ^
IndentationError: unindent does not match any outer indentation level
pi@switchdoclabs:~/SDL_Pi_SkyWeather $ vi testWeatherRack.py
pi@switchdoclabs:~/SDL_Pi_SkyWeather $ sudo python testWeatherRack.py
---------------------------------------
-----------------
 SDL_Pi_WeatherRack Library
 WeatherRack Weather Sensors
-----------------
Rain Total=   0.00 in
Wind Speed=   0.00 MPH
max Ever Wind Speed=         0.00 MPH
MPH wind_gust=        0.00 MPH
max Ever Gust wind_gust=   0.00 MPH
(' value = ', 1299.75)
(' voltageValue = ', 1.29975)
Wind Direction=                     0.00 Degrees
Wind Direction Voltage=             1.300 V
-----------------
-----------------
---------------------------------------
-----------------
 SDL_Pi_WeatherRack Library
 WeatherRack Weather Sensors
-----------------
Rain Total=   0.00 in
Wind Speed=   0.00 MPH
max Ever Wind Speed=         0.00 MPH
MPH wind_gust=        0.00 MPH
max Ever Gust wind_gust=   0.00 MPH
(' value = ', 425.4375)
(' voltageValue = ', 0.4254375)
Wind Direction=                     67.50 Degrees
Wind Direction Voltage=             0.425 V
-----------------
-----------------
---------------------------------------
-----------------
 SDL_Pi_WeatherRack Library
 WeatherRack Weather Sensors
-----------------
Rain Total=   0.00 in
Wind Speed=   0.59 MPH
max Ever Wind Speed=         0.59 MPH
MPH wind_gust=       20.39 MPH
max Ever Gust wind_gust=   20.39 MPH
(' value = ', 840.9375)
(' voltageValue = ', 0.8409375)
Wind Direction=                     0.00 Degrees
Wind Direction Voltage=             0.844 V
-----------------
-----------------
---------------------------------------
```

This completes the special GPIO testing. Now our final test is using the internal SkyWeather device detection system to complete the testing.

Test 3 – Testing SkyWeather Device Detection

To run the detection test, we start and then terminate (with a control-C or a control-| key) the SkyWeather.py main software.   We haven't configured the software yet. That will be done in the SkyWeather Operations manual, but the SkyWeather software is pretty good at detecting what is wired correctly and what is not.

You also may want to take the black cap off of your skycamera module.

Note: If you are not using the SwitchDoc Labs preloaded SD Card, then you should download the SkyWeather Software from https://github.com/switchdoclabs/SDL_Pi_SkyWeather
and following the installation instructions in the README.md file.

Start the SkyWeather Software by typing:

```
sudo pigpiod
sudo python SkyWeather.py
```

You will get a display similar to what follows.   Interrupt the program with "control-c" or "control-|" after the device availabilities are listed.

```
pi@switchdoclabs:~/SDL_Pi_SkyWeather $ sudo python SkyWeather.py
(33.2928, 18, 6)
()
('Pi Camera Revision', u'ov5647')
('after bme680', True)
as3935 start
as3935 present at 0x02
as3935 Interrupt
Fast Read AM2315temperature= 20.1
Fast Read AM2315humdity= 99.9
Fast Read AM2315crc= 52609
Fast Read AM2315c= 52609
outsideTemperature: 20.1 C
outsideHumidity: 99.9 %
crc: 0xcd81


SkyWeather Solar Powered Weather Station Version 029 - SwitchDoc Labs


Program Started at:2019-04-27 14:25:23


---------------------
I2C Mux - TCA9545:              Present
BME680:         Present
BMP280:         Not Present
SkyCam:         Present
DS3231:         Not Present
HDC1080:        Not Present
AM2315:         Present
ADS1015:        Not Present
ADS1115:        Present
AS3935:         Present
OLED:           Not Present
SunAirPlus:           Not Present
SI1145 Sun Sensor:           Not Present
TSL2591 Sun Sensor:          Present
DustSensor:           Present
WXLink:         Not Present

UseBlynk:       Not Present
UseMySQL:       Not Present
Check WLAN:           Present
WeatherUnderground:          Not Present
UseWeatherStem:          Not Present
```

```
--------------------
sendmail exception raised
----------------
  Sample and Display
----------------
----------------
  Weather Sampling
----------------
(' value = ', 2948.4375)
(' voltageValue = ', 2.9484375)
----------------
AM2315temperature= 20.1
AM2315humdity= 99.9
AM2315crc= 52609
AM2315c= 52609
----------------
  SunAirPlus Not Present
----------------
----------------
----------------
----------------
  AS3935 Lightning Detector
----------------
Last result from AS3935:
----No Lightning detected---
Lightning Count =  0
----------------
----------------
  Sample and Display Done
----------------
----------------
Scheduled Jobs
----------------
Jobstore default:
    patTheDog (trigger: interval[0:00:10], next run at: 2019-04-27 14:25:36 PDT)
    checkForButtons (trigger: interval[0:00:10], next run at: 2019-04-27 14:25:36 PDT)
    sampleAndDisplay (trigger: interval[0:00:30], next run at: 2019-04-27 14:25:56 PDT)
    tick (trigger: interval[0:01:00], next run at: 2019-04-27 14:26:26 PDT)
    takeSkyPicture (trigger: interval[0:01:00], next run at: 2019-04-27 14:26:26 PDT)
    updateRain (trigger: interval[0:05:00], next run at: 2019-04-27 14:30:26 PDT)
    checkForShutdown (trigger: interval[0:05:00], next run at: 2019-04-27 14:30:26 PDT)
    doAllGraphs (trigger: interval[0:15:00], next run at: 2019-04-27 14:40:26 PDT)
    barometricTrend (trigger: interval[0:15:00], next run at: 2019-04-27 14:40:26 PDT)
    read_AQI (trigger: interval[0:15:00], next run at: 2019-04-27 14:40:26 PDT)
    WLAN_check (trigger: interval[0:30:00], next run at: 2019-04-27 14:55:26 PDT)
    statusRain (trigger: interval[1:00:00], next run at: 2019-04-27 15:25:26 PDT)
    rebootPi (trigger: cron[day='5-30/5', hour='0', minute='4'], next run at: 2019-04-30 00:04:00
PDT)
----------------
AS3935 Interrupt
processing Interrupt from as3935
Last Interrupt = 0x1:  Noise Floor too low. Adjusting
^CTraceback (most recent call last):
  File "SkyWeather.py", line 1744, in <module>
    time.sleep(1.0)
KeyboardInterrupt
```

Compare your output with this section:

```
---------------------
I2C Mux - TCA9545:              Present
BME680:             Present
BMP280:             Not Present
SkyCam:             Present
DS3231:             Not Present
HDC1080:            Not Present
AM2315:             Present
```

Version 1.2 April 2019

```
ADS1015:            Not Present
ADS1115:            Present
AS3935:             Present
OLED:               Not Present
SunAirPlus:             Not Present
SI1145 Sun Sensor:          Not Present
TSL2591 Sun Sensor:         Present
DustSensor:             Present
WXLink:             Not Present

UseBlynk:           Not Present
UseMySQL:           Not Present
Check WLAN:             Present
WeatherUnderground:         Not Present
UseWeatherStem:         Not Present
--------------------
```

You should match this status.   If you don't match this, check your wiring carefully. It is almost never a bad part, just something mis-wired.

You have now completed testing. Time to move to the SkyWeather Configuration and Operations Manual.

The Science and Education Goals Behind SkyWeather

Everything we build for the Maker market is designed for education and learning.   Making is education.   Making is learning.   Building your own projects allows you to innovate around a framework and do wonderful things that of which we have never thought.

The educational goals for SkyWeather are:
•   Learn about the Raspberry Pi and installing software on the Pi
•   Connecting up sensors to the Raspberry Pi
•   Learning about Feedback loops
•   Understand your indoor environment and what affects it
•   Learn about the new technology called the Internet of Things
SkyWeather designed to be the hub to which you connect everything to turn your Raspberry Pi into a complete Weather Station that talks to the Cloud.   Just ready to be customized to your project and usage.   It is designed to be a great way of learning to hook up hardware to the Raspberry Pi. And you have all the source code to modify to work the way you want it to do.
Our partnership with WeatherSTEM brings this kickstarter into the realm of cloud based data mining, great graphics displays and even time lapse photography.   SkyWeather and WeatherSTEM together rock.
This is a great kit in which to learn about weather sensing, data sharing in the cloud and the Raspberry Pi.

Support

As with all SwitchDoc Labs products, technical support is given through the forums on Forum.switchdoc.com If you have issues that can be solved by our fabulous customer service department, please go to www.switchdoc.com and send your issues through our Contact page on the top menu.

Disclaimer

SwitchDoc Labs, LLC takes no responsibility for any physical injuries and possession loss caused by those reasons which are not related to product quality, such as operating without following the operating manual and cautions, natural disasters or force majeure.

SwitchDoc Labs, LLC has compiled and published this manual which covers the latest product description and specification.   The contents of this manual are subject to change without notice.