

SwitchDoc Labs

WeatherSense Solar Powered Lightning Sensor Assembly and Test Manual

February 2021
Version 1.1



Table of Contents

Cautions when building and using WeatherSense Sensors.....	2
Errata.....	2
What is The WeatherSense Solar Powered Air Quality Sensor?	6
Before You Build Your WeatherSense Lightning	7
Step by Step Assembly and Parts List	8
What are we doing here?	8
Parts List	8
Parts you need to buy separately from the kit.....	11
How to select a LiPo Battery	11
Step by Step Assembly	12
Testing WeatherSense Lightning	19
What does the JSON from the WeatherSense LightningSensor mean?	21
Optional: Reducing the power needed for your WeatherSense Lightning	Error! Bookmark not defined.
Disclaimer	22

Cautions when building and using WeatherSense Sensors

- 1) Keep all water away from the electronics and power supply at all times!
- 2) This is not a toy! Keep it out of reach of young children and pets.
- 3) SwitchDoc Labs assumes no liabilities in the use of this kit, beyond the refund of the purchase price.

Errata

What is The WeatherSense Solar Powered Air Quality Sensor?

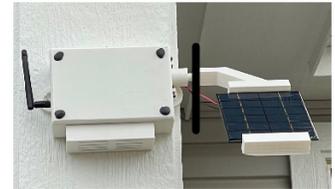
Easy to build. Easy to learn about the IoT (Internet of Things) and the Raspberry Pi.

The **heart** of the new WeatherSense Sensors is the our new 433MHz MiniProPlus CPU board in working in conjunction with the HM3301 Laser Air Quality Sensor.

The **WeatherSense Air Quality** kit is so simple that even middle school children can build it with just a little adult help for configuration and installation.



WeatherSense Wireless Kits and the Raspberry Pi



Air Quality

Software Defined Radio



433MHz



Lightning

SwitchDoc Labs

Full Open Source Arduino IDE compatible C Software that you can Modify and the open source Python3 software for the Raspberry Pi.

We provide the Python3 software (for the Raspberry Pi) and C for the WeatherSense Lightning. All open source with the kit. The Pure Python software can be modified to add new sensors, support new cloud software and connect up to your own projects and software.

Before You Build Your WeatherSense Lightning

You should build and test your WeatherSense Lightning system as below BEFORE you put it in the option 3D Printed case. Get it working first, then put it in the case. Believe us, it is always easier to debug the system before you close it up in the case! The manual for the case and weatherproofing is called the "WeatherSense Lightning Weatherproofing, Assembly and Test Manual". All manuals are available on the WeatherSense Lightning Product page on shop.switchdoc.com.

Step by Step Assembly and Parts List

Cautions: Keep your static charge to a minimum during your assembly and operation. Touch metal before handling parts. Avoid shuffling your feet. Before starting assembly, layout all the parts above and familiarize yourself with the various parts.

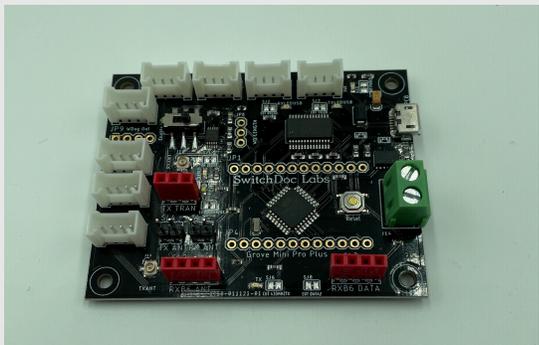
What are we doing here?

We are assembling the WeatherSense Lightning System.

In this manual, we are going to assemble the WeatherSense Lightning system and test all the functions. If you plan to put WeatherSense Lightning outside, and after you complete this manual, proceed to the “WeatherSense Lightning Weatherproofing, Assembly and Test Manual” on the WeatherSense Lightning Product Page.

Parts List

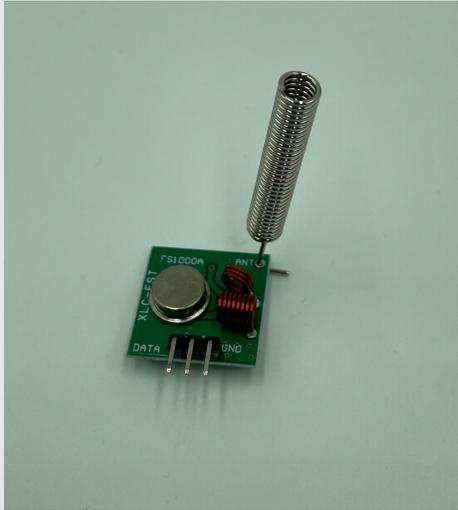
Part A
SwitchDoc Labs Grove Mini Pro Plus



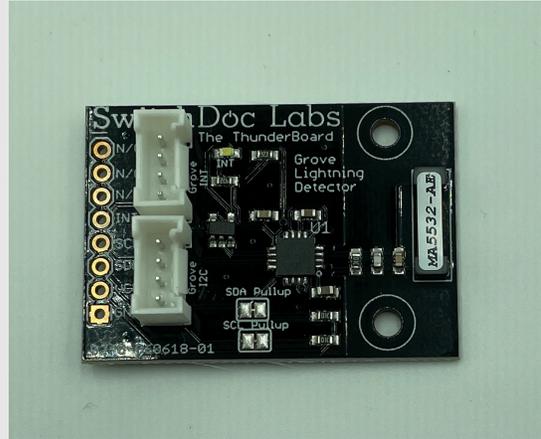
Part B
Short USB A to Micro USB



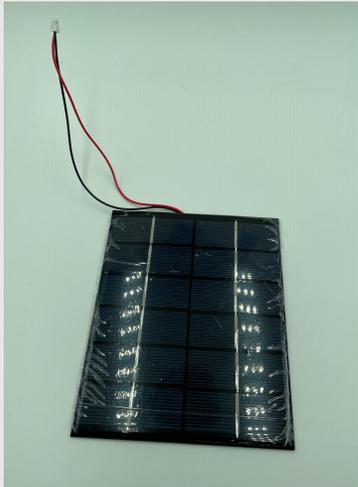
Part C
433MHz Transmitter with Antenna



Part D
ThunderBoard Lightning Detector



Part E
330mA 6V Solar Panel



Part F
SunAirPlus2 Solar Power Data Collector
and Controller



Part G
3.7V LiPo Battery (Not included)



Part H
One (1) 20cm Grove Cable



Part I

One (1) 50cm Grove Cable



Parts you need to buy separately from the kit

- 3.7V LiPo Battery
- Raspberry Pi
- Compatible SDR Software Defined Radio (For example: <https://hpjhlytllzrwf4qn-24552113.shopifypreview.com/products/software-defined-radio-sdr-and-antenna>)
- 16GB SD Card (unless you bought the SD Card from SwitchDoc Labs)
- Power Supply for the Raspberry Pi

How to select a LiPo Battery

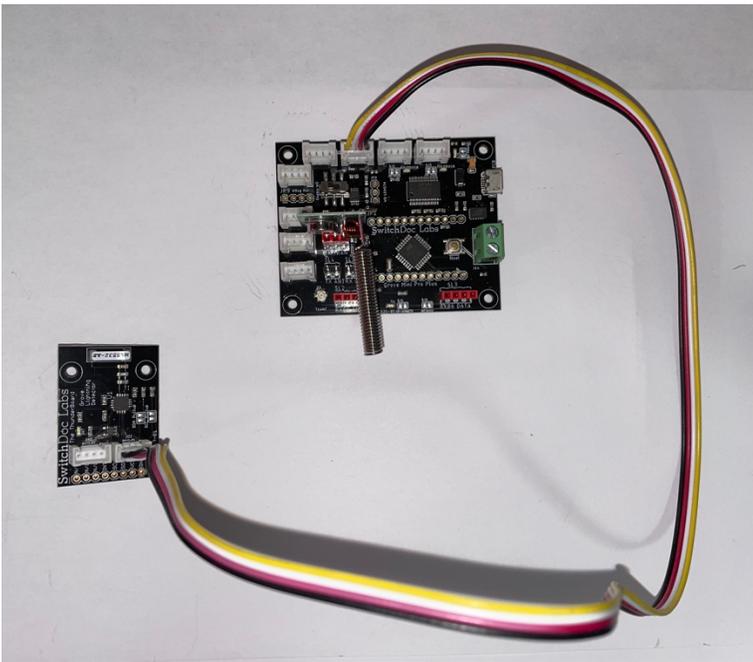
The WeatherSense Lightning requires 3.7V LiPo battery.

How large of a LiPo battery you need depends on how much sun you get and how often you have cloudy weather. Generally, we would recommend a 6600mAh battery such as <https://www.adafruit.com/product/353>. Adafruit has a great selection and you can find good ones on SparkFun.com too.

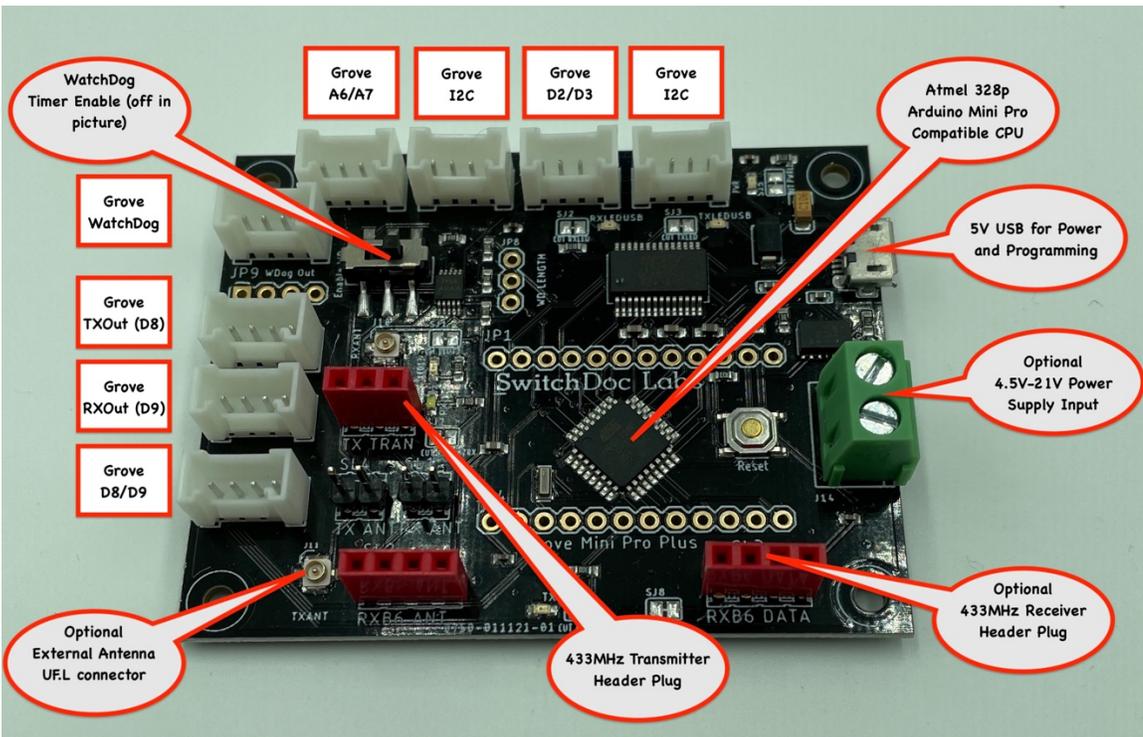
WARNING: if you get them off of Amazon, check the wiring. Most of them are wired backwards. Here's a great website showing the problem and how to rewire the batteries if you wish:

<https://docs.particle.io/tutorials/learn-more/batteries/>

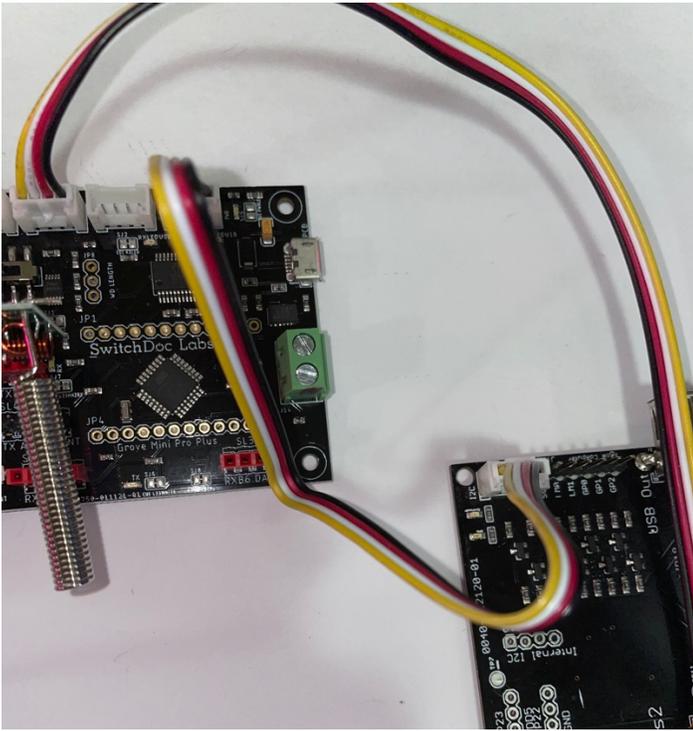
Step 3: Take a 50cm grove cable (Part I) and plug it into the grove connection on the Mini Pro Plus (Part A) marked J5-I2C and the other end into the I2C grove connector on the ThunderBoard Lightning (Part D). A red dot on the bottom means that the Mini Pro Plus has been programmed as a Lightning Sensor. A White means it has been programmed as an Air Quality Sensor.



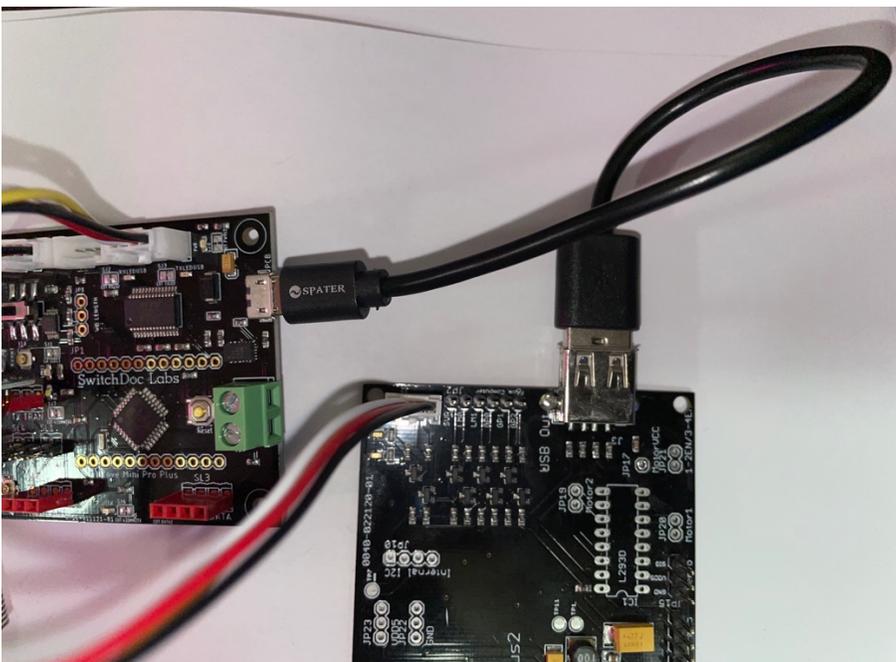
Step 4: Push the WatchDog Time Enable Switch to the Left (enable) on the Mini Pro Plus board (Part A).



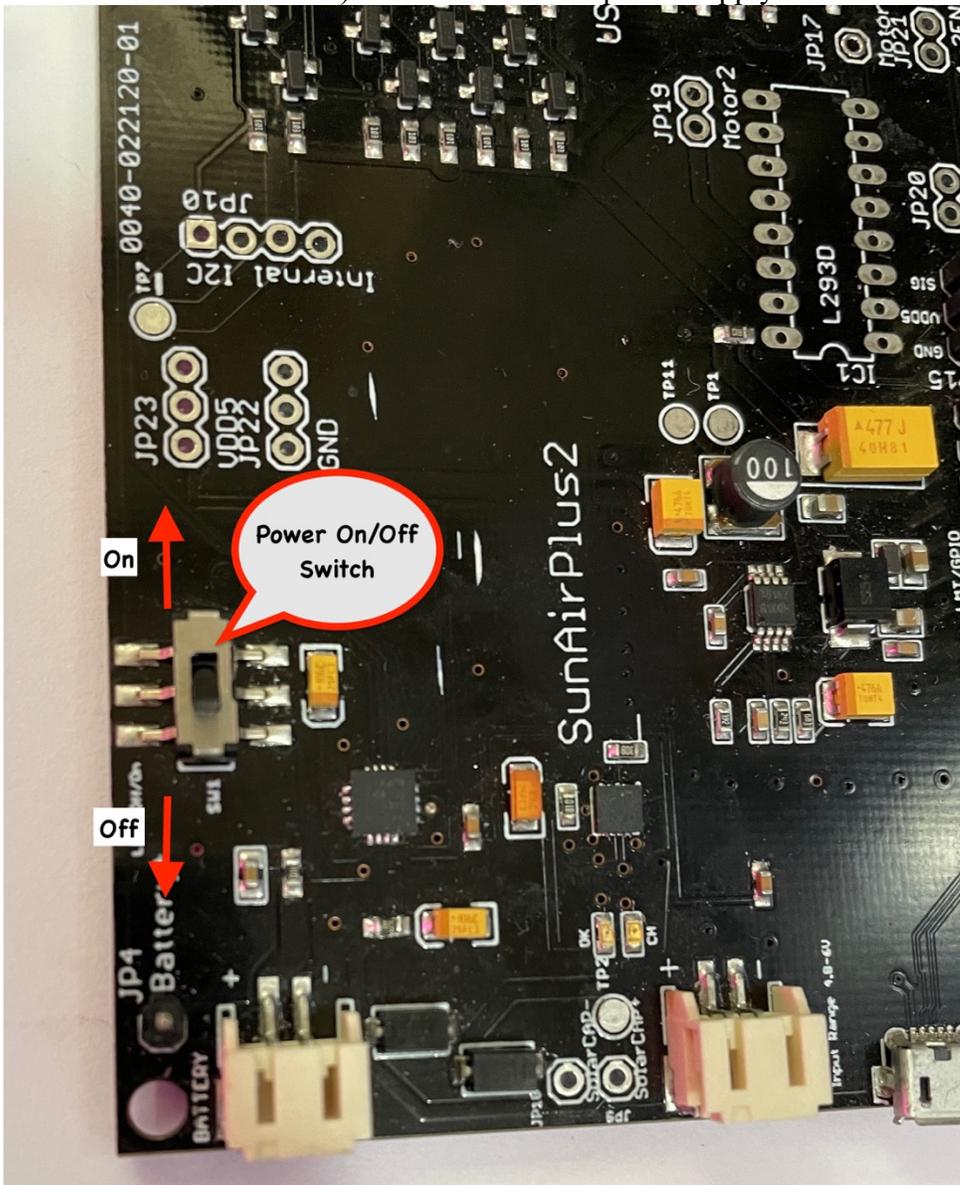
Step 5: Take the other grove cable (Part H) and plug it in into the Mini Pro Plus (Part A) grove connector marked J7-I2C and the other end into the grove connector on the SunAirPlus2 board (Part F).



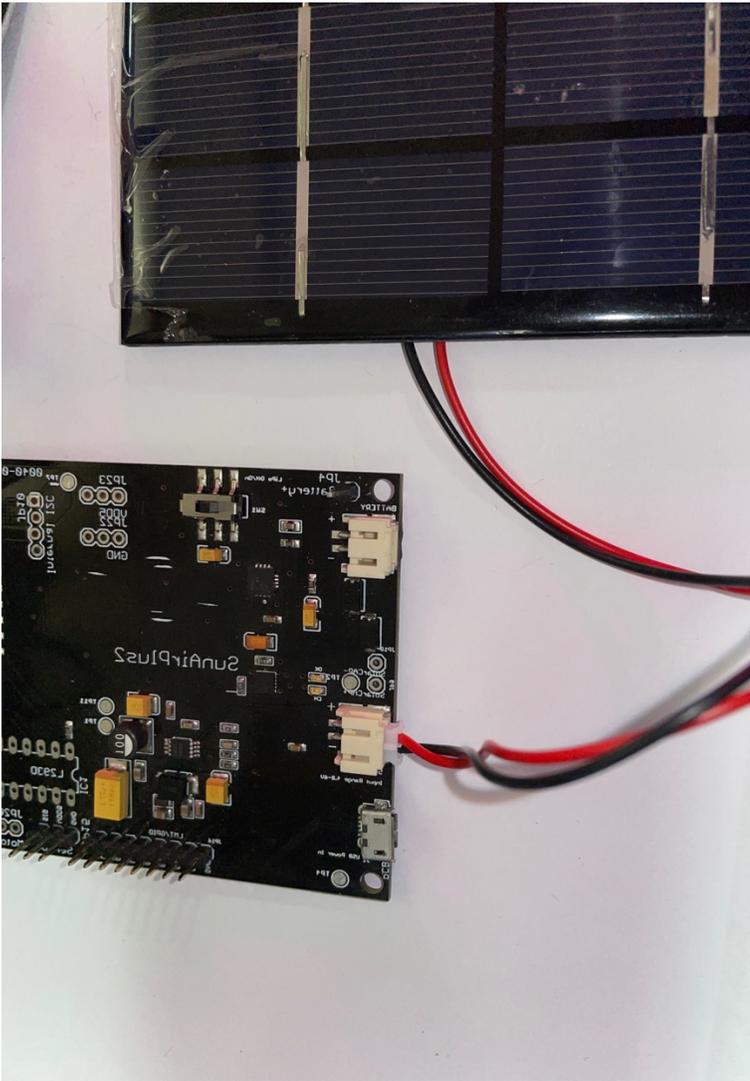
Step 6: take the USB micro to Type A cable (Part B) and plug it into the USB micro connector on the MiniProPlus board (Part A) and the other end into the TypeA connector on the SunAirPlus2 Board (Part F).



Step 7: Make sure the switch on the SunAirPlus2 board (Part F) is pushed toward the Battery Connector (away from the Grove Connector). This turns off the power supply to the MiniProPlus (Part A)



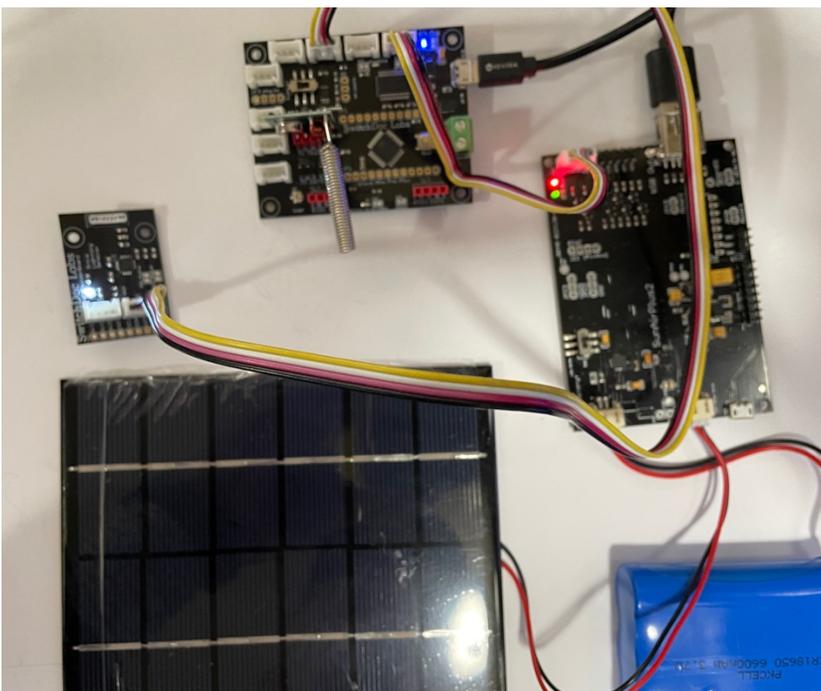
Step 8: Plug the Solar Panel (Part E) JST-2 connector into the SunAirPlus2 (Part F) board JST-2 plug marked SOLAR (next to the USB Micro connector on SunAirPlus2 (Part F)). You can take the protective plastic off of the solar panel (Part E) at this time.



Step 9: Plug your LiPo Battery (Part G – not included) into the JST-2 connector marked Battery (near the power switch) on the SunAirPlus2 board (Part F).



Step 10: Push the switch on the SunAirPlus2 (Part F) board away from the battery connector. You should see LED lights come on on both boards and see the white Interrupt LED light up (and then go off shortly) on the ThunderBoard. If the white LED on the ThunderBoard (Part D) does not go out, you have plugged your 50cm Grove cable into the wrong plug on the ThunderBoard (Part D) or on the MiniProPlus (Part A). Go back to Step 3.



Step 11: On powerup, the WeatherSense Lightning will send a message. You will see the LED market TX

flash on the MiniProPlus (Part A). A message has been sent and should have been received by your Raspberry Pi software.

You can use the SwitchDoc Labs Lightning Simulator (<https://shop.switchdoc.com/products/the-thunder-board-i2c-lightning-simulator>) to cause lightning events.

You can also connect a micro USB cable to the SunAirPlus2 board to your computer to watch the serial port – lots of data printing there. Check out the WeatherSense Advanced Programming Manual to see how to do this with the Arduino IDE.

If there is a lightning event, the Lightning board will send a message almost immediately. You can hold a drill near the Thunderboard to cause a non-lightning event (which will be noted by the WeatherSense Lightning detector as a disturber) which will generate a message. Check out forum.switchdoc.com for ideas on creating lightning.

Step 12: The Lightning board will send a message every ~11 minutes giving the latest Lightning readings and all the solar power data. You can see the graphs on the dash_app once you have started the WeatherSense Software (see below).

Step 13: Turn off the SunAirPlus2 (Part F) board by pushing the switch towards the battery connector


```
sudo python3 WeatherSenseMonitor.py
```

And start adding in the missing libraries as they come up as missing imports, otherwise do at least the following:

Installing mariadb:

<https://pimylifeup.com/raspberry-pi-mysql/>

```
sudo apt-get install python3-dev libmysqlclient-dev
sudo pip3 install mysqlclient
```

Next:

```
sudo -u root -p < WeatherSenseWireless.sql
```

Step 4: Note the IP Address of your Raspberry Pi. Type in the following commands to get your IP address:

```
hostname -I
```

You will get something like this:

```
pi@SwitchDocLabs:~/SDL_Pi_WeatherSense Lightning $ hostname -I
192.168.1.44
```

Step 5: Update WeatherSense to the latest version. Type the following commands into a terminal window.

```
cd
cd SDL_Pi_WeatherSense
git pull
```

You will see something like this:

```
pi@SwitchDocLabs:~/SDL_Pi_WeatherSense$ git pull
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 12 (delta 10), reused 12 (delta 10), pack-reused 0
Unpacking objects: 100% (12/12), done.
From https://github.com/switchdoclabs/SDL_Pi_WeatherSense Lightning
 f2193a0..c98a45c master    -> origin/master
Updating f2193a0..c98a45c
Fast-forward
 README.md                | 2 ++
 SkyCamera.py             | 2 +-
 WeatherSense Lightning.py | 2 +-
 pcclogging.py            | 12 ++++++-----
 state.py                 | 3 +++
 testWirelessSensors.py  | 18 ++++++-----
 6 files changed, 22 insertions(+), 17 deletions(-)
```

Or, if your software is up to date:

```
pi@SwitchDocLabs:~/SDL_Pi_WeatherSense$ git pull
Already up to date.
```

Step 6: First we will test the reception of the wireless weather sensors. Test your SDR and WeatherSense installation as follows. Note you must have completed the WeatherSense Lightning assembly.

```
cd /home/pi/SDL_Pi_WeatherSense
sudo python3 WeatherSenseMonitor.py
```

Now start your Lightning sensor by turning the SunAirPlus2 switch to on. You will then see something similar to this on your terminal window:

```
pi@SwitchDocLabs:~/SDL_Pi_WeatherSense $ sudo python3 WeatherSenseMonitor.py
Pending jobs:
  readSensors (trigger: date[2021-02-27 12:13:55 PST], pending)
-----

Scheduled Jobs
#####
-----
Read Wireless Sensors
#####
Jobstore default:
  No scheduled jobs
-----
starting 433MHz scanning
#####
processing Lightning TB Data
This is the raw data: {"time" : "2021-02-27 16:16:03", "model" : "SwitchDoc Labs TB", "len" : 44,
"messageid" : 0, "deviceid" : 1, "protocolversion" : 1, "softwareversion" : 1,
"weathersenseprotocol" : 16, "irgsource" : 0, "previousinterruptresult" : 0,
"lightninglastdistance" : 0, "sparebyte" : 0, "lightningcount" : 0, "interruptcount" : 0,
"loadvoltage" : 4.976, "batteryvoltage" : 4.208, "batterycurrent" : 52.800, "loadcurrent" : 27.000,
"solarpanelvoltage" : 2.960, "solarpanelcurrent" : -0.000, "auxa" : 3, "mic" : "CRC"}
```

What does the JSON from the WeatherSense LightningSensor mean?

The WeatherSenseMonitor.py python3 program reads in the 433MHz signals into the Raspberry Pi, decodes them and formats the results as a JSON packet that will be provided to the WeatherSenseMonitoring software for processing and storage in the database. More about the SDL_Pi_WeatherSense software in the WeatherSense Software Manual.

If you have other WeatherSense Sensors (such as SkyWeather2, WeatherRack2, F016TH Indoor sensors, etc), you will see them show up in the list too.

```
processing Lightning TB Data
This is the raw data: {"time" : "2021-02-27 16:16:03", "model" : "SwitchDoc Labs TB", "len" : 44,
"messageid" : 0, "deviceid" : 1, "protocolversion" : 1, "softwareversion" : 1,
"weathersenseprotocol" : 16, "irgsource" : 0, "previousinterruptresult" : 0,
"lightninglastdistance" : 0, "sparebyte" : 0, "lightningcount" : 0, "interruptcount" : 0,
"loadvoltage" : 4.976, "batteryvoltage" : 4.208, "batterycurrent" : 52.800, "loadcurrent" : 27.000,
"solarpanelvoltage" : 2.960, "solarpanelcurrent" : -0.000, "auxa" : 3, "mic" : "CRC"}
```

The values and units of the are:

- timestamp: When the packet has been received
- model: SwitchDoc Labs TB
- len: Number of bytes in the packet
- messageid: Reset to 0 on bootup

- `deviceid`: 1 – If you have more than one Lightning sensor, you can change this ID by recompiling the software
- `protocolversion`: Current protocol of the Lightning sensor
- `softwareversion`: This shows the current version of the software on the Mini Pro Plus
- `weathersenseprotocol`: 16 for WeatherSense Lightning
- `irqsource`: 0 for no interrupt, 1 for a noise interrupt, 4 for a disturber and 8 for lightning stroke.
- `previousinterruptresult`: same as above, but for an immediately preceding interrupt.
- `lightninglastdistance`: Distance of the current lightning bolt in km
- `Sparebyte`: not used
- `lightningcount`: Number of lightning interrupts since bootup
- `interruptcount`: Number of interrupts of all kinds since bootup

The solar data collected by the device is:

- `loadvoltage`: Voltage supplied to the computer (V)
- `batteryvoltage`: Voltage of the LiPo Battery (V)
- `batterycurrent`: Current being supplied by the battery (mA)
- `loadcurrent`: Current being supplied to the computer (mA)
- `solarpanelvoltage`: Voltage of the Solar Panel (V)
- `solarpanelcurrent`: Current being supplied by the Solar Panel (mA)

The `AuxA` variable contains state information about what sensors the WeatherSense Lightning has connected to and if the sensor is in low voltage mode.

Each bit of the lower four bits of the `AuxA` variable (`OCAB`) are coded as:

`C` = 1, Voltage below 2.8, Lightning interrupts are ignored, 0 means normal operation

`A` = 1, SunAirPlus2 (Solar) Present, 0 not present

`B` = 1, AS3935 ThunderBoard Present, 0 not present

You now have completed testing of your WeatherSense Lightning sensor.

Print your 3D Prints and move on over to the WeatherSense Lightning Weatherproofing manual. Then take a look at the WeatherSense software manual for the Raspberry Pi.

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.