SwitchDoc Labs

WeatherSense
Solar Powered SkyCam
Assembly and Test Manual

August 2021

Version 1.1

WeatherSense

SKYCAM

Table of Contents

Cautions when building and using WeatherSense Sensors

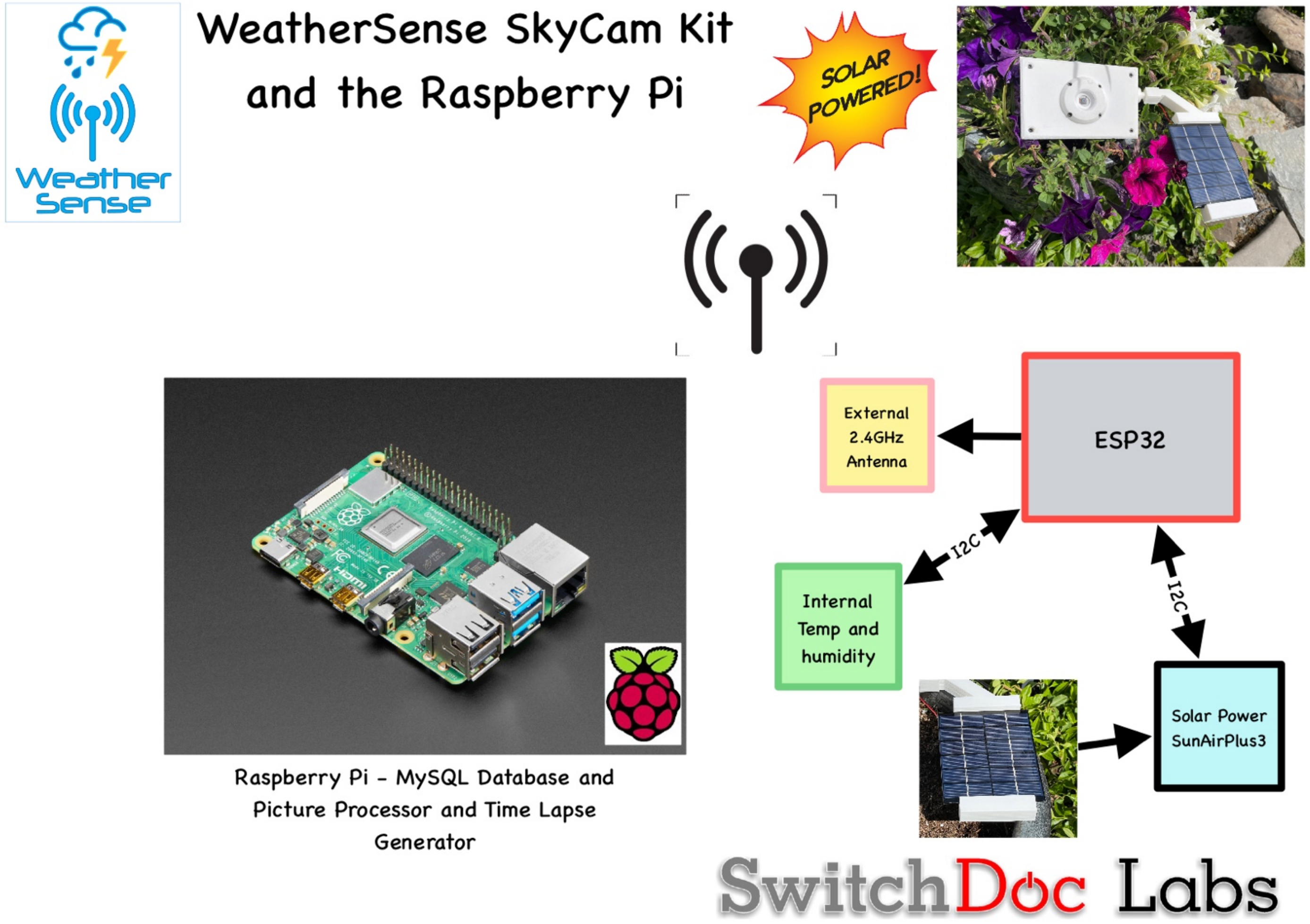
1) Keep all water away from the electronics and power supply at all times!

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 SkyCam Solar Powered SkyCam?

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



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

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 SkyCam ESP32 Board. All open source with the kit. The Raspberry Pi Pure Python software can be modified to add new sensors, support new cloud software and connect up to your own projects and software. We also provide the C++/C code f

Before You Build Your WeatherSense SkyCam System

You should build and test your WeatherSense SkyCam system as below BEFORE you put it in the optional 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! All manuals are available on the WeatherSense SkyCam 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?

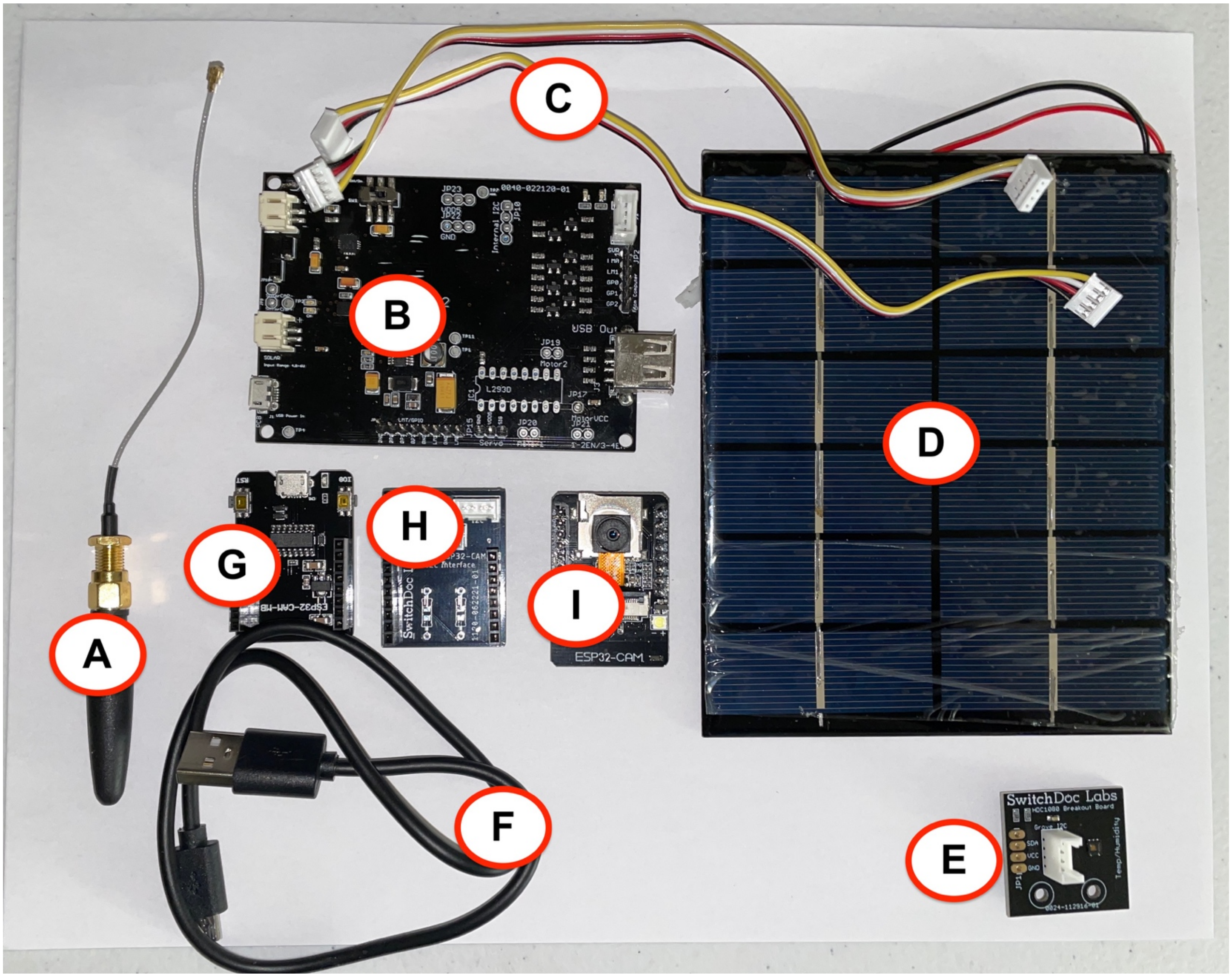
We are assembling the WeatherSense SkyCam System.

In this manual, we are going to assemble the WeatherSense SkyCam system and test all the functions. And then we will place it in the 3D Printed Box

Parts List

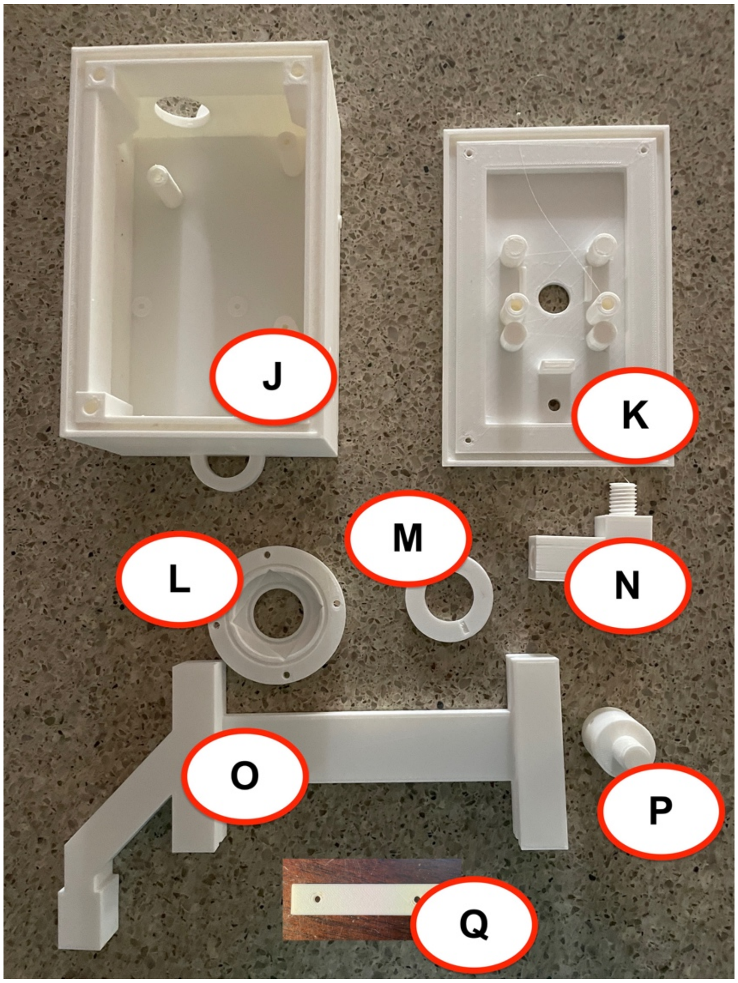
SkyCam Kit Parts List

A – 2.1GHz Antenna and uf.l cable
B – SunAirPlus2 or SunAirPlus3 Board (either will work)
C – Two 20cm Grove Cables
D – 330mA Solar Panel
E – HDC1080 Temperature and Humidity Sensor
F – USB Type A to USB Micro Cable
G – FTDI USB Interface for SkyCam Board
H – Grove and I2C Interface Board
I – ESP32-CAM Board



SkyCam 3D Print Parts List

J – Main SkyCam Box
K – SkyCam Box Top
L – SkyCam Window Top
M – SkyCam Window Bottom
N – 90 Degree Threaded Block
O – Solar Panel Bracket
P – Slotted Hole Plug
Q – SkyCam Mount Bracket



Parts you need to buy separately from the kit

- 3.7V LiPo Battery
- Raspberry Pi
- Compatible SDR Software Defined Radio (For example: https://shop.switchdoc.com/products/software-defined-radio-sdr-and-antenna)

- 32GB SD Card (unless you bought the SD Card from SwitchDoc Labs)
- Power Supply for the Raspberry Pi

The SDL SkyWeather2 kit comes with an SDR and antenna.

How to select a LiPo Battery

The WeatherSense SkyCam sensor requires a 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/

Not Enough Sun?

Want to look at more solar panels? Here is an article about adding them. https://www.switchdoc.com/2021/08/tutorial-adding-more-solar-panels-to-skycam-and-weathersense-sensors/

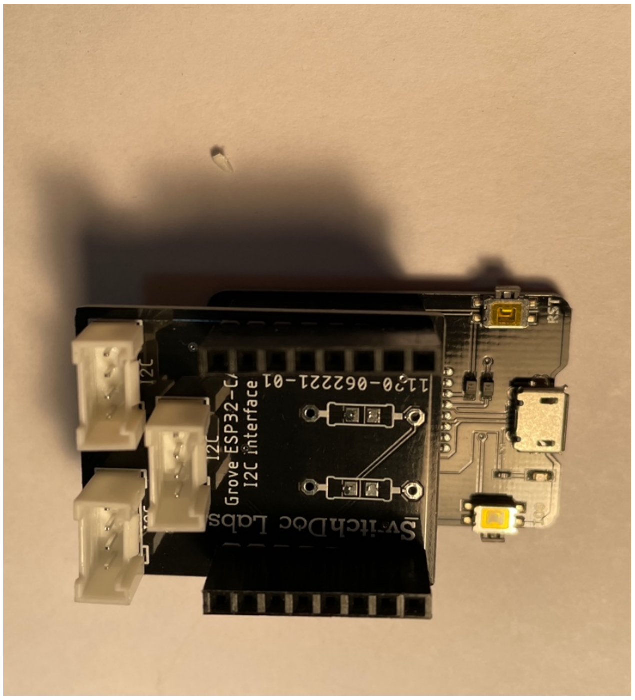
Step by Step Assembly

Remember you are putting together the WeatherSense SkyCam to do testing and debugging. You will then need to disassemble the unit and follow the re-assembly instructions for inside the 3D Print.

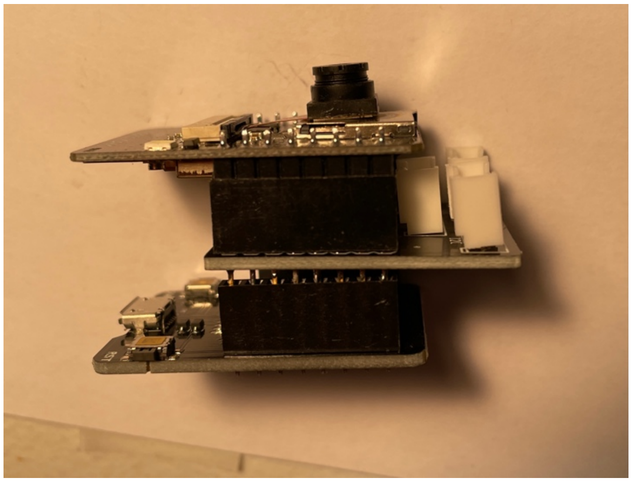
Step 1: Lay out all parts on a flat non-conductive (☺) surface.

Note on Steps 2 and 3: These three boards may come pre-assembled in your kit.

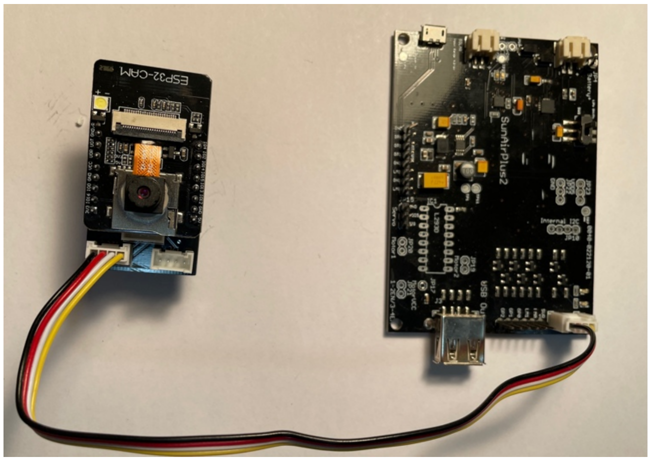
Step 2: Take the FTDI USB Interface Board (Part G) and plug it into the Grove and I2C Interface Board (Part H). Make sure that all the pins are aligned, and the boards are oriented properly as in the picture below.



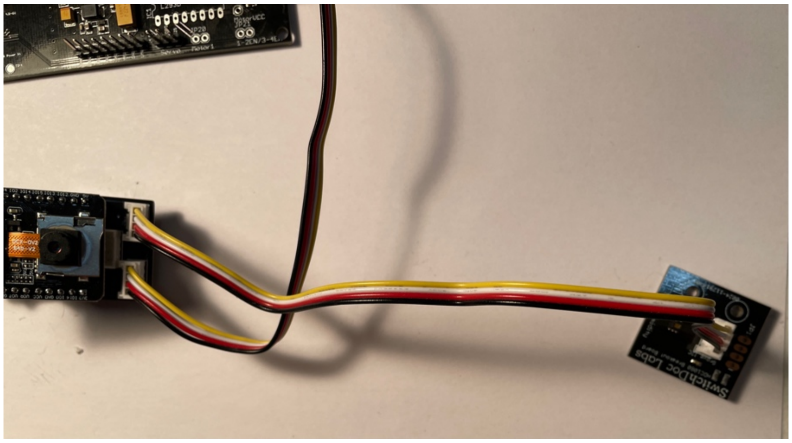
Step 3: Take the ESP32-CAM Board (Part I) and plug it int the top of the assembly from Step2. Make sure that all the pins are aligned and the boards are oriented properly as in the picture below.



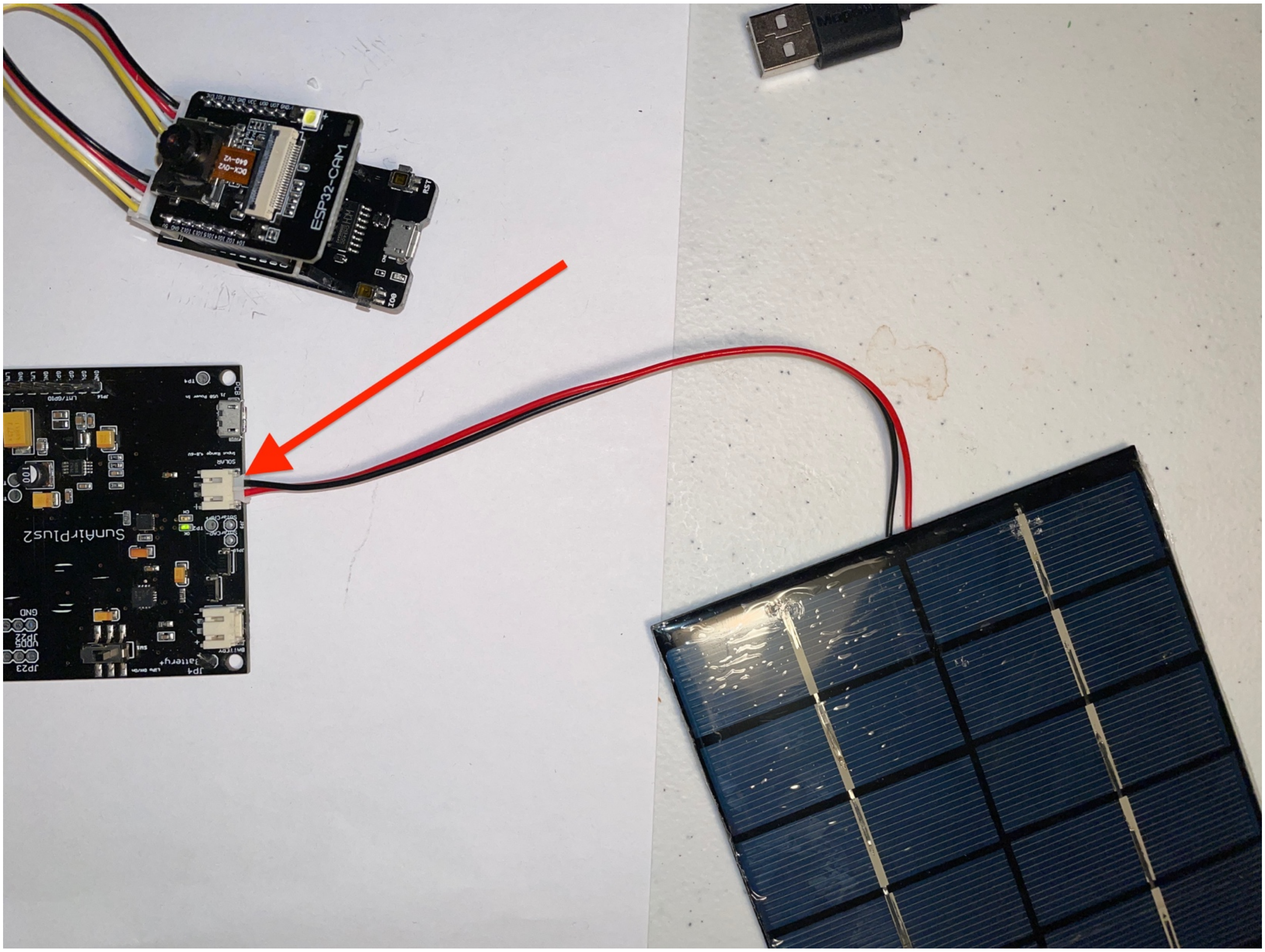
Step 4: Using one of the 20cm Grove Cables (Part C), plug one end into either of the I2C ports on the Camera assembly (Parts G,H,I) and the other end into the Grove I2C connector on the SunAirPlus2 (or SunAirPlus 3) board.



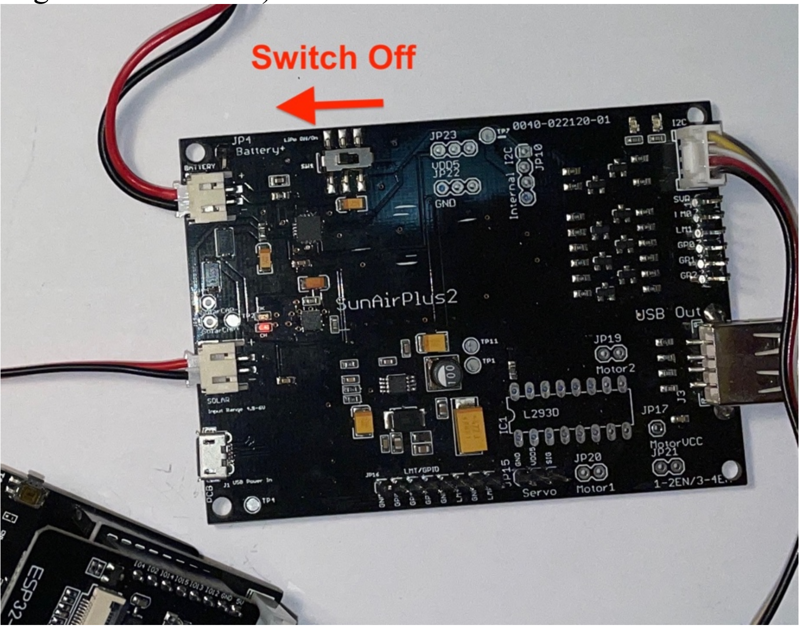
Step 5: Using the other 20cm Grove Cable (Part C), plug one end into either of the I2C ports on the Camera assembly (Parts G,H,I) and the other end into the HDC1080 Temperature and Humidity Sensor (Part E).



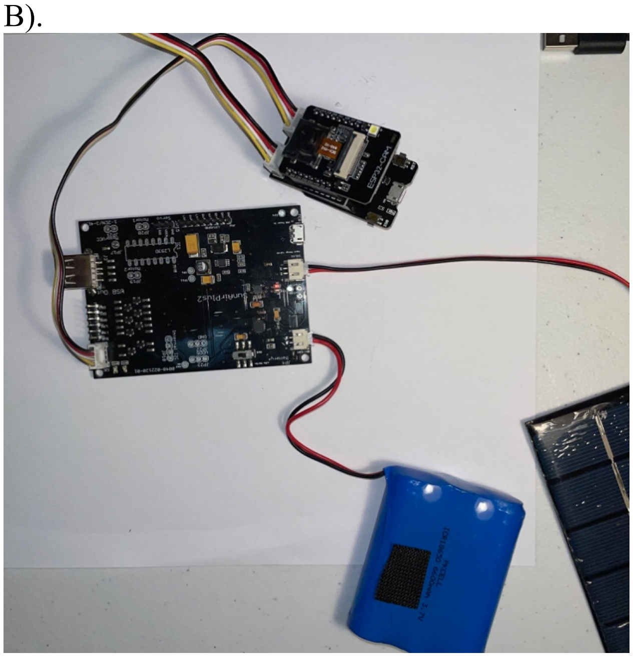
Step 6: Using the JST-2 plug on the 330mA Solar Panel (Part D) plug it into the Solar Input Plug on the SunAirPlus2 (or SunAirPlus3 board) (Part B). Not that the solar panel plug is next to the micro USB plug! Don't make a mistake here!



Step 8: Make sure the switch on the SunAirPlus2 (or SunAirPlus3) board (Part B) is off (towards the battery plug as shown below).

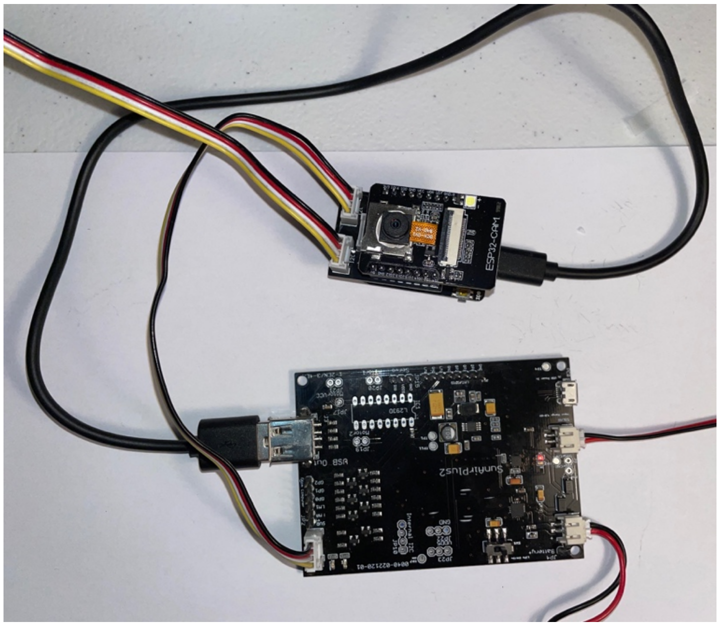


Step 8: Plug in your LiPo Battery (not included) to the Battery Port on the SunAirPlus2 (or SunAirPlus3) (Part B).

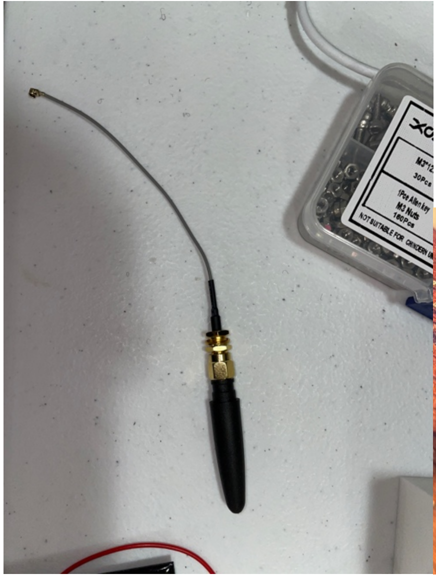


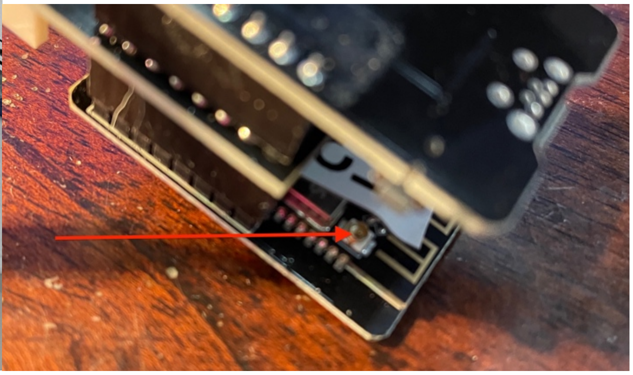
Step 9: Point a light at the Solar Panel. You should see a red LED turn on next to the Solar Input plug on the SunAirPlus (2 or 3) Board (Part B). This says the SunAirPlus (2 or 3) board recognizes your solar panel and is getting a small amount of charge from your interior light.

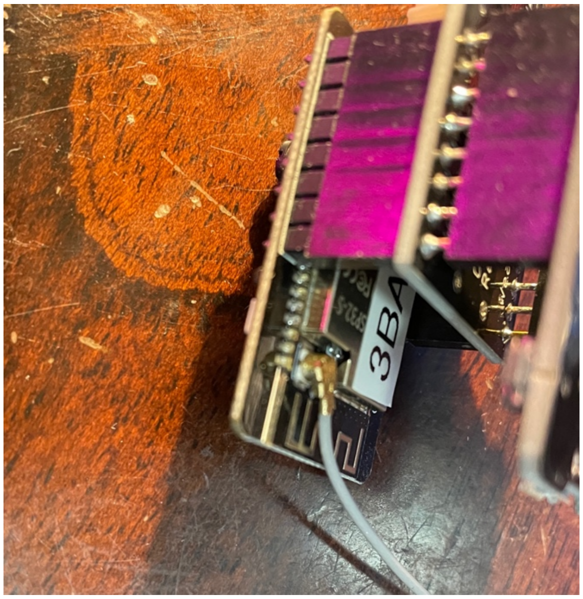
Step 10: Plug in the Type A to Micro USB Plug into the Camera Assembly (Parts G,H,I) and the Type A USB Port on the SunAirPlus (2 or 3) board (Part B).



Step 11:   Take the Antenna (Part A), remove the red cover, and screw the Antenna Stub onto the Antenna Cable.   Take the uf.l end of the cable and carefully snap it onto the uf.l socket on the bottom of the ESP32-Cam board (Part I) which is part of the Camera Assembly (Parts G,H,I).

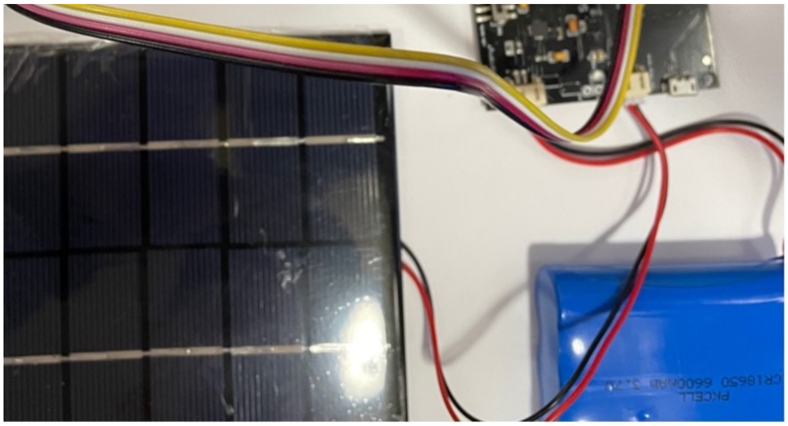






Now we are ready to provision SkyCam with your Wifi and Raspberry Pi information.

You have now completed initial assembly. Now on to testing.



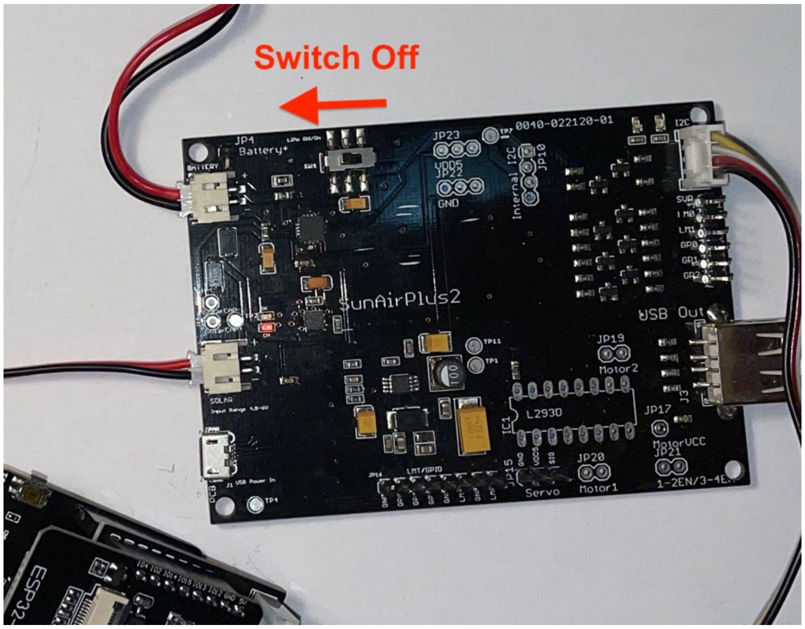
CR18650 6600mAh 3.7V
PKCELL

Testing WeatherSense SkyCam

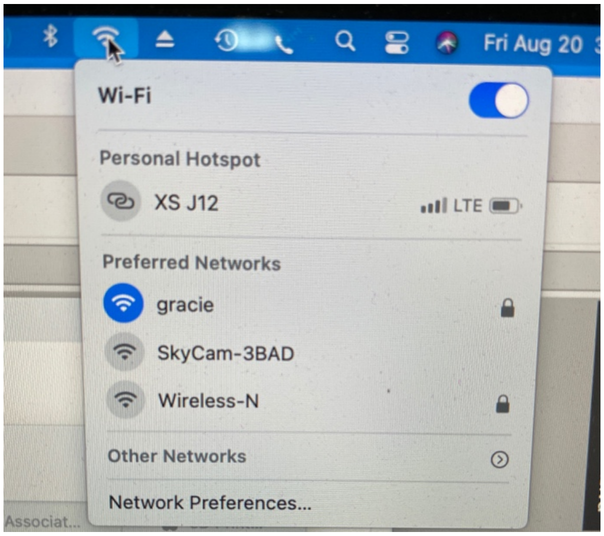
Provisioning your SkyCam

First you have to connect up to your local WiFi network.

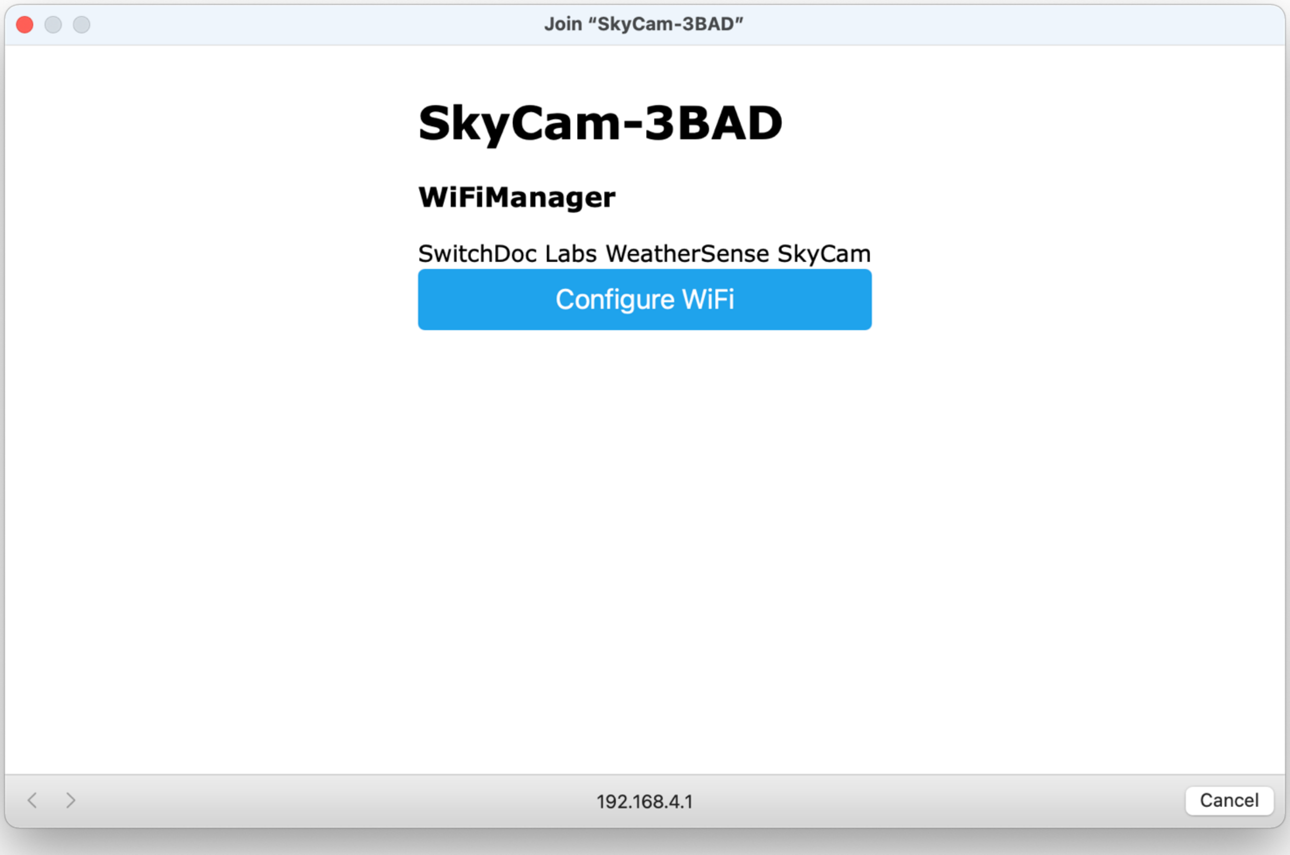
Step 1: Plug in the power and turn on your SkyCam by moving the switch on the SunAirPlus (2 or 3) board (Part B) to the right. You will see the USB Power LED illuminate on the SunAirPlus board and you will see a red LED on the bottom board on the Camera Assembly (Parts G,H,I).



Step 2: Look at your available networks. You should see a SkyCam-XXXX access point. This is your SkyCam. Below is the available networks screen from a Mac.



Step 3: Click on the SkyCam-XXXX (SkyCam-3BAD) access point. In a few seconds you will see this screen:



Join “SkyCam-3BAD”

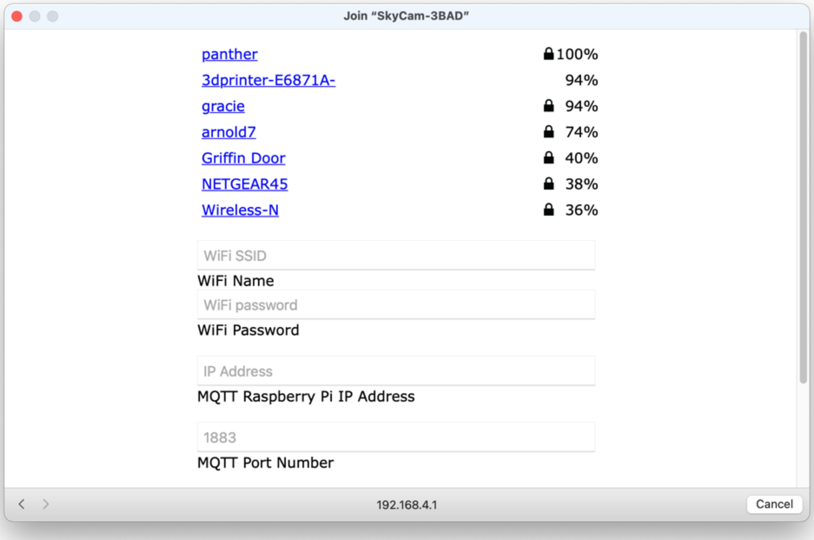
SkyCam-3BAD

WiFiManager

SwitchDoc Labs WeatherSense SkyCam

Note down the SkyCam-XXXX number. You will be using this later to identify your SkyCam unit (since you can have many of them!).

Step 4: Click on Configure WiFi. You will then see a screen like this. Select your access point SSID and then fill out the rest of the fields.



panther	🔒100%
3dprinter-E6871A-	94%
gracie	🔒 94%
arnold7	🔒 74%
Griffin Door	🔒 40%
NETGEAR45	🔒 38%
Wireless-N	🔒 36%

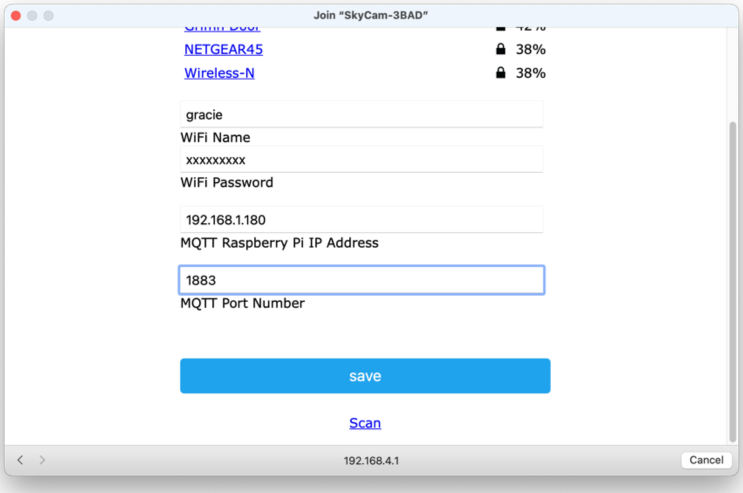
WiFi SSID
WiFi Name
WiFi password
WiFi Password

IP Address
MQTT Raspberry Pi IP Address

1883
MQTT Port Number

The fields are:

- WiFi Name – The SSID of your WiFi network.
- WiFi Password – The password for your WiFi network
- MQTT Raspberry Pi IP Address - The IP of your Raspberry Pi. To find the IP of your Raspberry Pi, type "hostname -I". Or follow one of the many tutorials on the web. **If you don't have this yet, stop** and figure it out before you continue. Click the red in the upper left corner to cancel and then connect to your WiFi. Start this process again later.
- MQTT Port Number – Unless you have changed this, use 1883 as the port number.



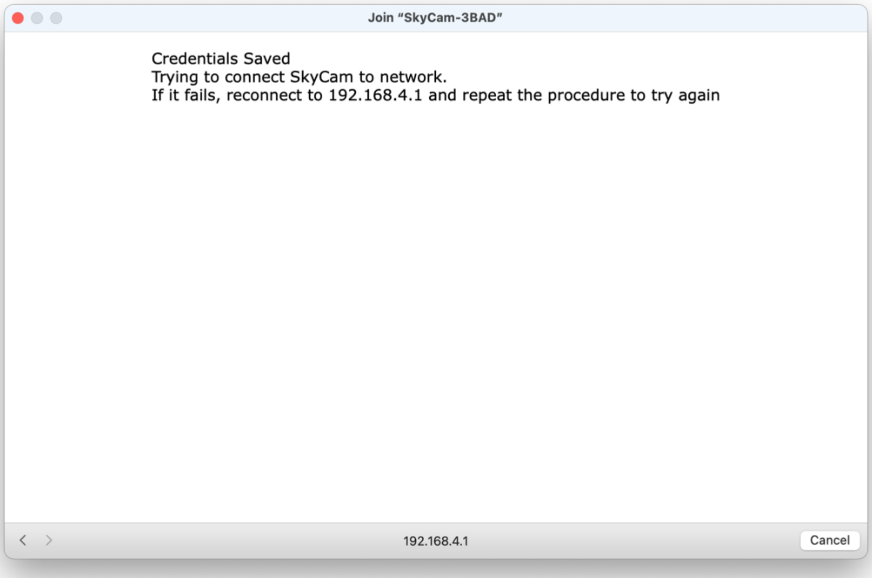
NETGEAR45	🔒 38%
Wireless-N	🔒 38%

gracie
WiFi Name
xxxxxxxxx
WiFi Password

192.168.1.180
MQTT Raspberry Pi IP Address

1883
MQTT Port Number

Step 5: Scroll down and click on the “Save” button. After a few seconds, you should see a screen like this:



Credentials Saved
Trying to connect SkyCam to network.
If it fails, reconnect to 192.168.4.1 and repeat the procedure to try again

Step 6: Reconnect to your WiFi. You SkyCam is provisioned.

Testing on the Raspberry Pi

If you have the SwitchDoc Labs SD Card, you can proceed to Step 4.

Step 1: Install the SDL modified version of rtl_433.

In a terminal window on your Pi at /home/pi type:

```
git clone https://github.com/switchdoclabs/rtl_433.git
```

Then compile it on the Raspberry Pi:

```
cd rtl_433/
mkdir build
cd build
cmake ..
make
make install
```

Step 2: Install the WeatherSense Software using these commands:

```
cd
git clone https://github.com/switchdoclabs/SDL_Pi_WeatherSense.git
```

Step 3: Add needed python modules to your system (a list will be provided on forum.switchdoc.com in the near future). If you want to jump into it, run

```
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$ 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
   f2193a0..c98a45c  master     -> origin/master
Updating f2193a0..c98a45c
Fast-forward
 README.md              |  2 ++
 SkyCamera.py           |  2 +-
 WeatherSenseMonitor.py         |  2 +-
 pclogging.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: Update the libraries and database:

```
sudo pip3 install python-aqi
sudo mysql -u root -p WeatherSenseWireless < updateWeatherSenseWireless.sql
```

Step 6: Test WeatherSense installation as follows. Note you must have completed the WeatherSense SkyCam assembly.

```
cd /home/pi/SDL_Pi_WeatherSense

sudo python3 WeatherSenseMonitor.py
```

Now make sure your SkyCam sensor is on by turning the SunAirPlus2 switch to on. You will then see something similar to this on your terminal window within about a minute:

```
pi@SwitchDocLabs:~/SDL_Pi_WeatherSense $ sudo python3 WeatherSenseMonitor.py
-----------------
WeatherSense Monitoring Software
Software Version  V009
-----------------
Pending jobs:
    readSensors (trigger: date[2021-08-20 15:56:39 PDT], pending)
    startMQTT (trigger: date[2021-08-20 15:56:39 PDT], pending)
    cleanPictures (trigger: cron[day='*', hour='3', minute='4'], pending)
    cleanTimeLapses (trigger: cron[day='*', hour='3', minute='10'], pending)
    buildTimeLapse (trigger: cron[day='*', hour='5', minute='30'], pending)
    readSensors (trigger: date[2021-08-20 15:56:39 PDT], pending)
-----------------
Scheduled Jobs
-----------------
creating new MQTT instance

######
Read Wireless Sensors
######
connecting to broker

######
Jobstore default:
Read Wireless Sensors
######
    cleanPictures (trigger: cron[day='*', hour='3', minute='4'], next run at: 2021-08-21 03:04:00
PDT)
starting 433MHz scanning
######
    cleanTimeLapses (trigger: cron[day='*', hour='3', minute='10'], next run at: 2021-08-21
03:10:00 PDT)
    buildTimeLapse (trigger: cron[day='*', hour='5', minute='30'], next run at: 2021-08-21 05:30:00
PDT)
-----------------
starting 433MHz scanning
Connected with result code 0
######

Picture ID 4182 Good Message 1 Resends: 0
CameraRotation= 90
mySkyCameraText= WeatherSense SkyCam-DE45 20-Aug-2021 15:57:36
query= INSERT INTO SkyCamPictures (cameraID, picturename, picturesize, messageID,
resends,resolution) VALUES('DE45', 'DE45_4182_2021-08-20-15-57-36.jpg', 19041, 4182, 0, 9 )
INFO msg Received: b'{"id": "DE45", "messageid": "4182", "messagetype": "4", "softwareversion":
"21", "sunairplusdevicepresent": "0", "hdc1080devicepresent": "1", "internaltemperature": "27.23",
"internalhumidity": "44.62", "currentrssi": "-57", "batteryvoltage": "0.00", "batterycurrent":
```

```
"0.00", "loadvoltage": "0.00", "loadcurrent": "0.00", "solarpanelvoltage": "0.00",
"solarpanelcurrent": "0.00", "gndrreboots": "0"}'
in INFO with SQL

Picture ID 47539 Good Message 2 Resends: 0
CameraRotation= 90
mySkyCameraText= WeatherSense SkyCam-26FD 20-Aug-2021 15:57:44
query= INSERT INTO SkyCamPictures (cameraID, picturename, picturesize, messageID,
resends,resolution) VALUES('26FD', '26FD_47539_2021-08-20-15-57-44.jpg', 117097, 47539, 0, 13 )
INFO msg Received: b'{"id": "26FD", "messageid": "47539", "messagetype": "4", "softwareversion":
"18", "sunairplusdevicepresent": "1", "hdc1080devicepresent": "1", "internaltemperature": "27.42",
"internalhumidity": "51.92", "currentrssi": "-70", "batteryvoltage": "3.75", "batterycurrent":
"138.40", "loadvoltage": "4.98", "loadcurrent": "100.20", "solarpanelvoltage": "4.70",
"solarpanelcurrent": "65.20", "gndrreboots": "1146"}'
in INFO with SQLWhat does the JSON from the WeatherSense SkyCam mean?
```

The WeatherSenseMonitor.py python3 program waits on the MQTT Packets from the SkyCam ls 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.

The above list shows two SkyCam cameras transmitting pictures.   SkyCam-DE45 and SkyCam-26FD.

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

What does this mean?

```
query= INSERT INTO SkyCamPictures (cameraID, picturename, picturesize, messageID,
resends,resolution) VALUES('26FD', '26FD_47539_2021-08-20-15-57-44.jpg', 117097, 47539, 0, 13 )
```

This query printout shows the SkyCam picture being entered in the database. The values are:

- cameraID: Unique ID for every SkyCam
- picturename: The name of the current picture
- picturesize: size in bytes of the current picture
- messageID: the Message ID (Starts at zero when you initialize a SkyCam) for the picture
- resends: How many chunks were resent for this message (SkyCam breaks up the picture into smaller chunks and resends them if they weren't received by the Pi correctly)
- resolution: What resolution of picture has been sent – see the "Programming SkyCam" later on in this manual.

INFO msg Received: b'{"id": "26FD", "messageid": "47539", "messagetype": "4", "softwareversion": "18", "sunairplusdevicepresent": "1", "hdc1080devicepresent": "1", "internaltemperature": "27.42", "internalhumidity": "51.92", "currentrssi": "-70", "batteryvoltage": "3.75", "batterycurrent": "138.40", "loadvoltage": "4.98", "loadcurrent": "100.20", "solarpanelvoltage": "4.70", "solarpanelcurrent": "65.20", "gndrreboots": "1146"}

The values and units of the are:

- id – SkyCam ID
- messageid: the Message ID (Starts at zero when you initialize a SkyCam) for the picture
- messagetype: 4 – The SkyCam Information Packet
- softwareversion: Which version of the C/C++ software is your SkyCam running
- sunairplusdevicepresent: 1 if the SunAirPlus is present and 0 if it is not detected
- hdc1080devicepresent: 1 if the HDC1080 is detected and 0 if it is not detected

- internaltemperature: internal box temperature. In degrees C
- internalhumidity: internal box humidity. In percent relative humidity
- currentrssi: Current Wifi signal SkyCam strength, more negative is weaker. This can help you debug range problems.
- 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)
- gndreboots: How many reboots have been called by the software (very rare in V19 and above).

You now have completed testing of your WeatherSense SkyCam sensor.

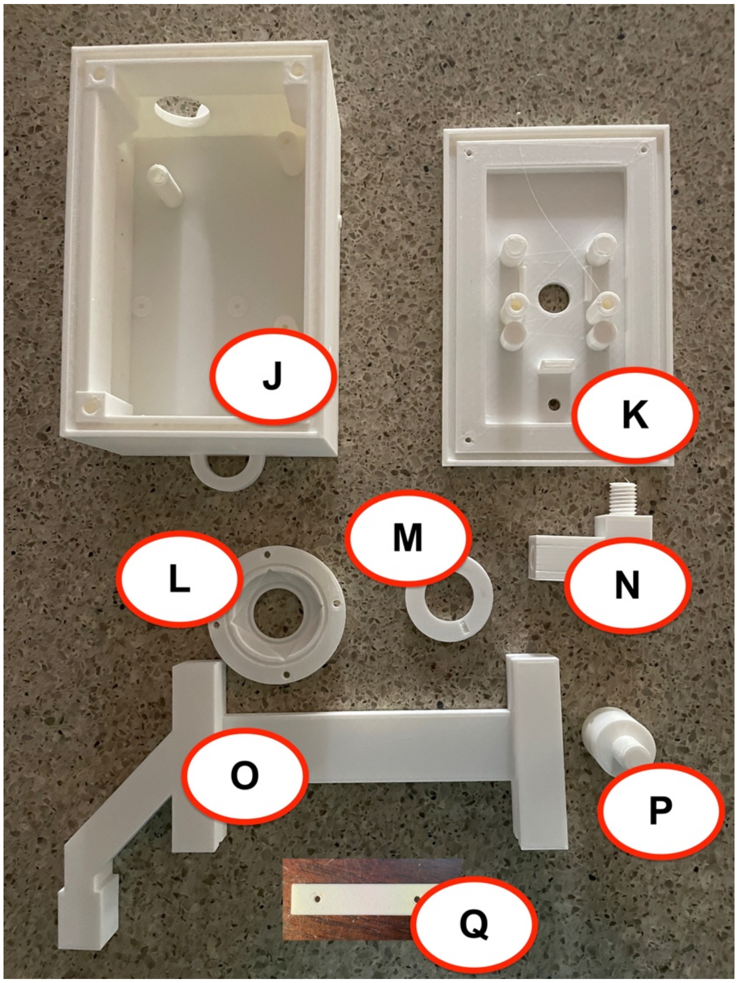
Now continue on to the final assembly using your 3D Printed parts.

Assembly into 3D Printed Box and Testing

If you haven't already, please print your 3D printed case (or order one from SwitchDoc Labs) for your SkyCam assembly.

Parts List

J – Main SkyCam Box
K – SkyCam Box Top
L – SkyCam Window Top
M – SkyCam Window Bottom
N – 90 Degree Threaded Block
O – Solar Panel Bracket
P – Slotted Hole Plug
Q – SkyCam Mount Bracket



Parts you need to buy separately from the kit for assembly

- 1 - 510mm x 3.6mm Screen Spline
- 14 - Brass Inserts
- 4 - 20mm M3 screws
- 8 - 12mm M3 screws
- 6 - 6mm M3 screws
- 1 - O-ring 1-1/16" ID
- 1 - O-ring - 1-3/4" ID

All of the above can be purchased at shop.switchdoc.com in one package for convenience.

https://shop.switchdoc.com/products/skyweather2-assembly-parts-kit

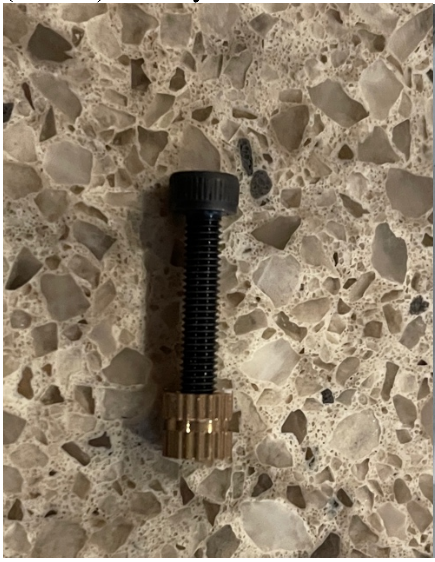
You will also need the following:

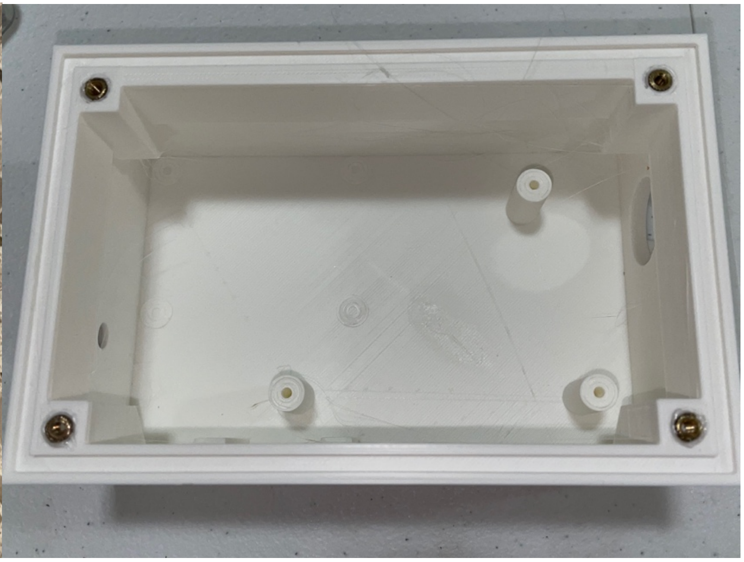
- Superglue
- 1 Package of Duct Seal (for covering holes and screws) (Amazon - https://amzn.to/33SMkyE Any Hardware Store)
- Velcro for securing LiPo Battery (any Hardware Store)
- 30mmx30mmx1mm Silica Quartz Glass Plate (https://shop.switchdoc.com/products/double-side-polished-far-uv-fused-silica-quartz-glass-sheets-plate-30-30-1mm or on eBay)

Step by Step Assembly

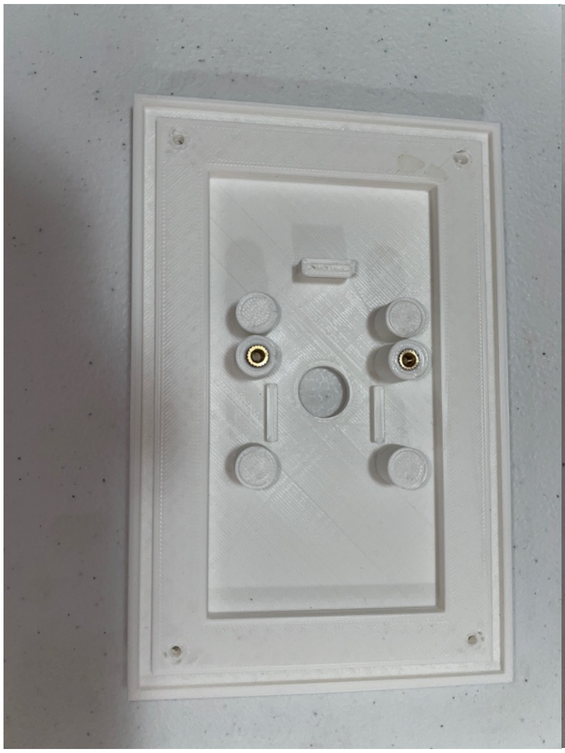
You will need to disassemble the SkyCam unit that you assembled in the above and follow the assembly instructions below. Don't unplug the Camera Assembly (Parts G,H,I) from each other.

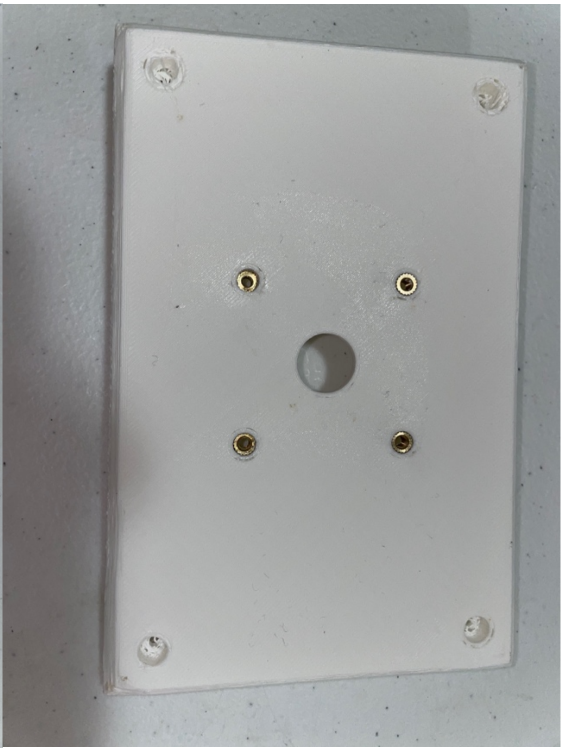
Step 1: Superglue 4 5mm brass inserts into the bottom box (Part J) as in the picture below. Use a screw to help you position and glue the inserts into the box.   Apply glue to the outside of the inserts then push them into the hole and gently unscrew the screw. If they are too tight to push in, gently pound the screw (with the insert attached as below) into the bottom box.   Super glue an additional 2 5mm brass inserts into the AQ Housing (Part K) Let dry at least 30 minutes.





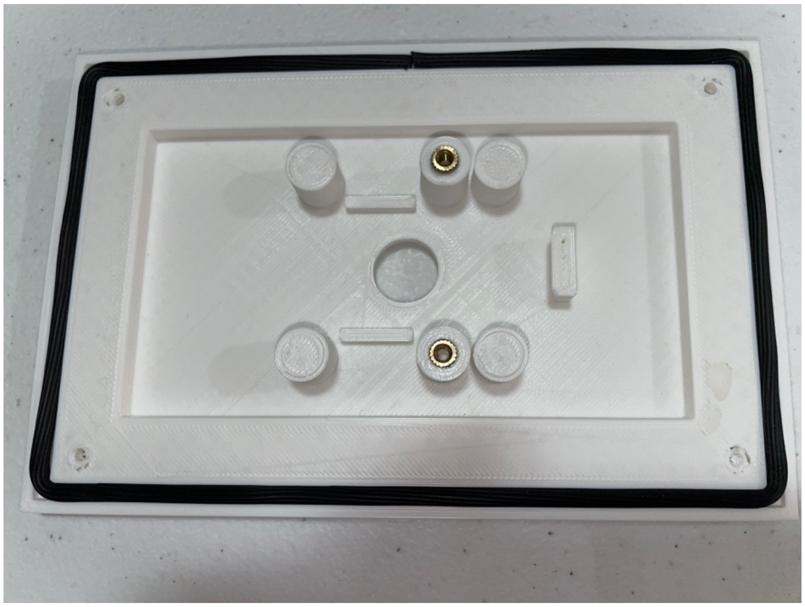
Step 2: Superglue 2 5mm brass inserts into the SkyCam Box Top (Part K) on the inside as shown and then glue 4 5mm brass inserts on the top of the SkyCam Box Top (Part K). Let dry 30 minutes.



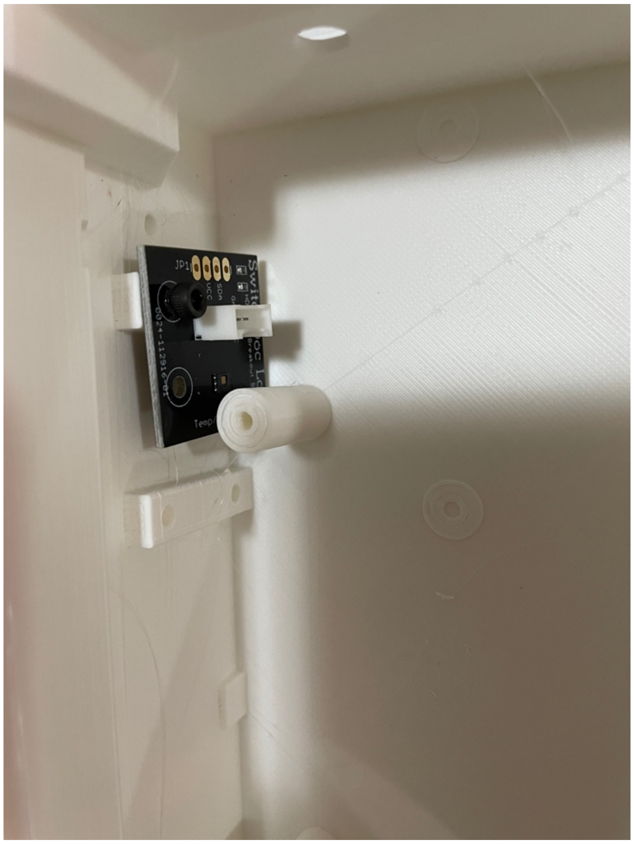


Step 3: Push a small screwdriver through the four corner holes of the SkyCam Box Top (Part K) to clear out any 3D printed extra material. This will help assembly later.

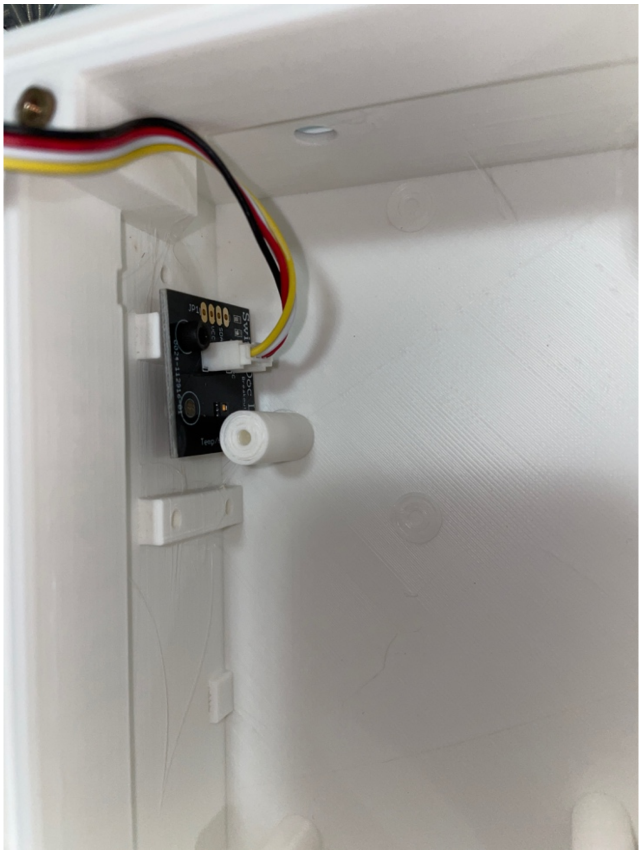
Step 4: Take the 510mm screen spline and press it into the groove on the inside of the SkyCam Box Top (Part K) trimming any excess.



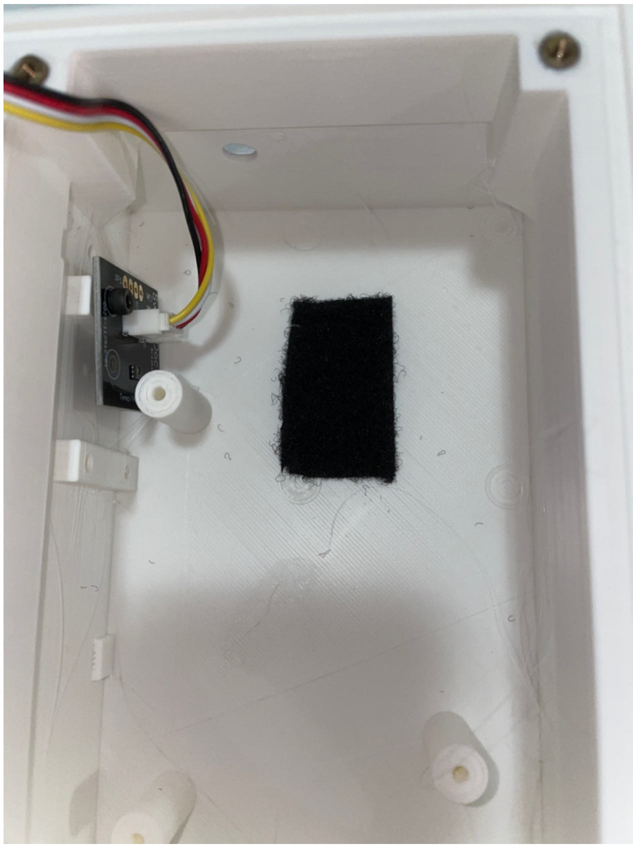
Step 5: Attach the HDC 1080 Temperature and Humidity Sensor (Part E) to the Main SkyCam Box (Part J) using one 5mm screws as shown below. As always, tighten, but don't over tighten the screws as you will strip the plastic threads. It can hang loose as long as it is secure.

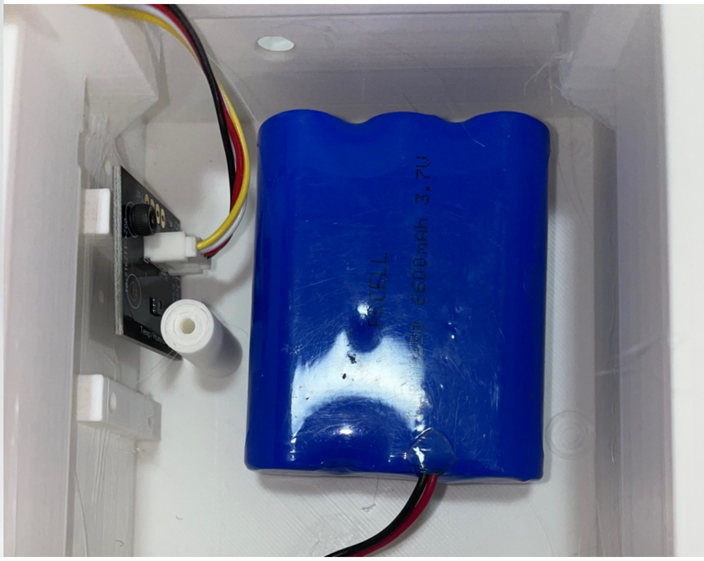


Step 6: Take a 20cm Grove Cable (Part C) and plug it into the Grove connector on the HDC1080 Temperature and Humidity Sensor (Part E).

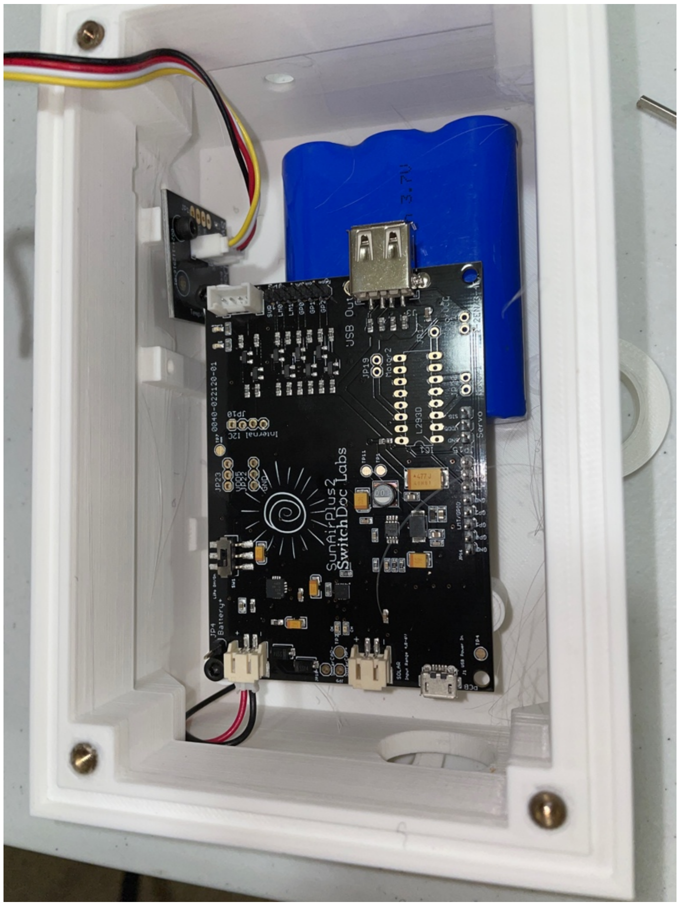


Step 8: Take a strip of Velcro and put it in the bottom of the Main SkyCam box (Part J) to secure the LiPo Battery (not included). Use the other side of the Velcro on the bottom of the battery. Make sure the battery cable will reach all the way to the other end of the box to plug into the SunAirPlus Board (Part B) later on.

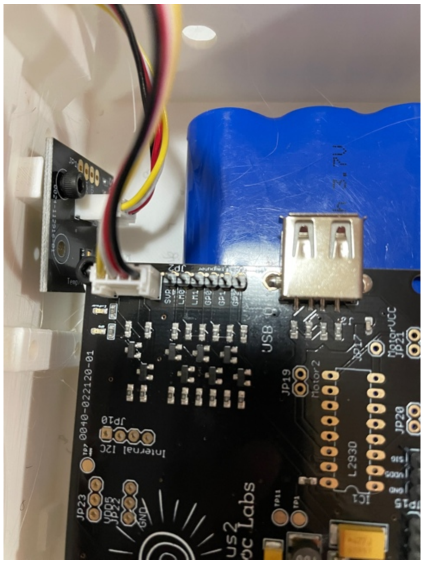




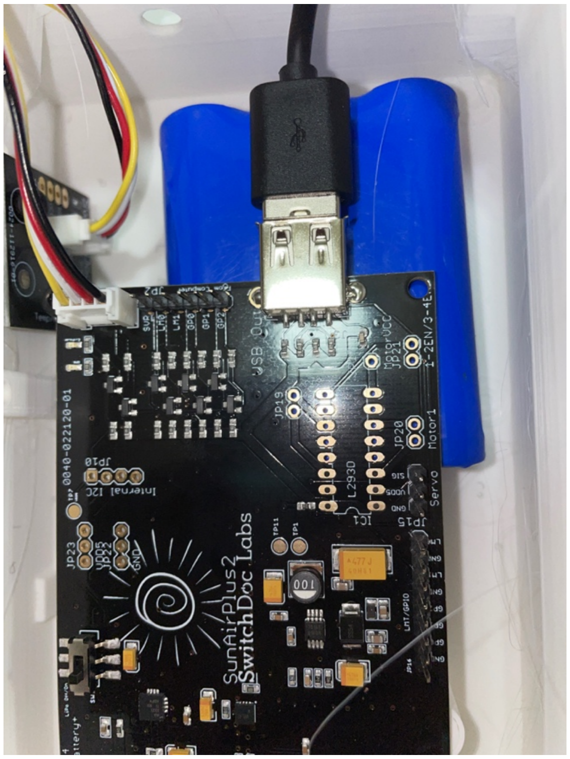
Step 9: Plug your LiPo battery into the battery JS2 plug (across from the switch in the lower right corner). Take your SunAirPlus Board (Part B), make sure the switch is off, and screw it down using two 5mm screws. Take care not to break the pylons or tight so much to strip the threads. You will only use the two mounting holes on the left side of the board.



Step 10: Plug the remaining 20cm Grove cable (Part C) into the I2C Grove connector on the SunAirPlus (Part B) board.

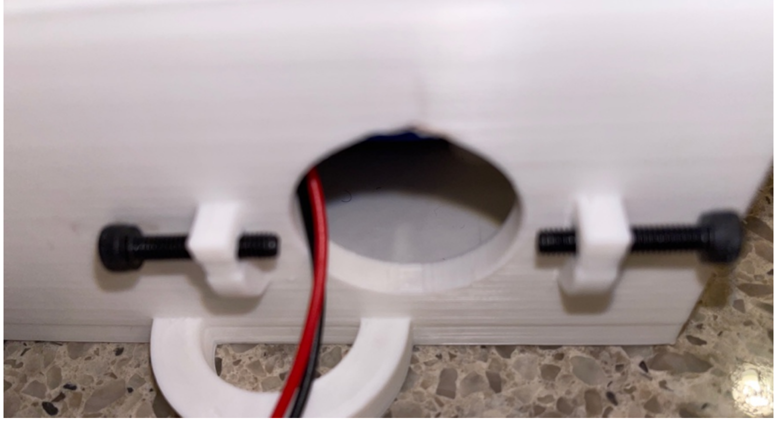


Step 11: Plug the Type A end of the USB micro to USB Type A cable (Part F) into the SunAirPlus Board (Part B). If the USB Plug won't fit, the remove the SunAirPlus board (Part B), attach the USB Cable (Part F) and replace the SunAirPlus Board (Part B).

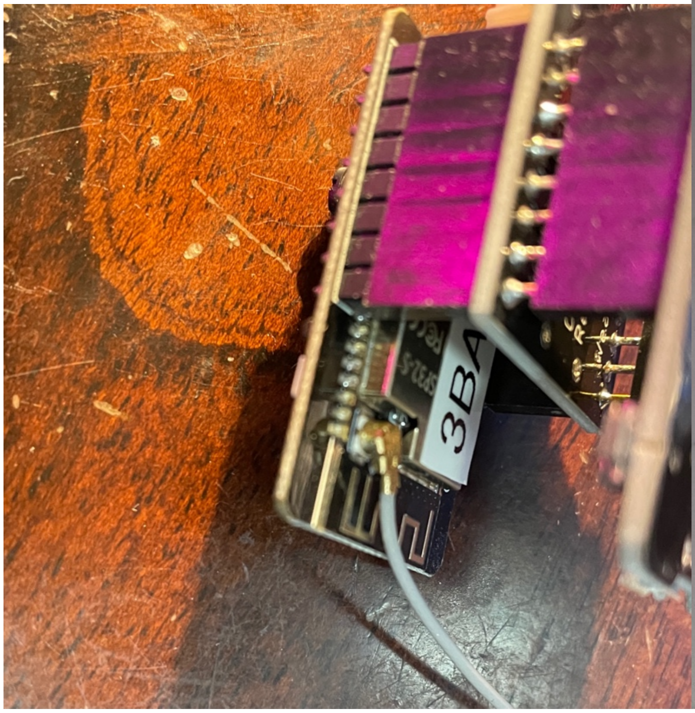


Step 12: Attach the Solar Panel (Part D) JST-2 plug to the JST-2 connector on the SunAirPlus (Part B) board as shown above. Remove the plastic covering of the Solar Panel at this time.

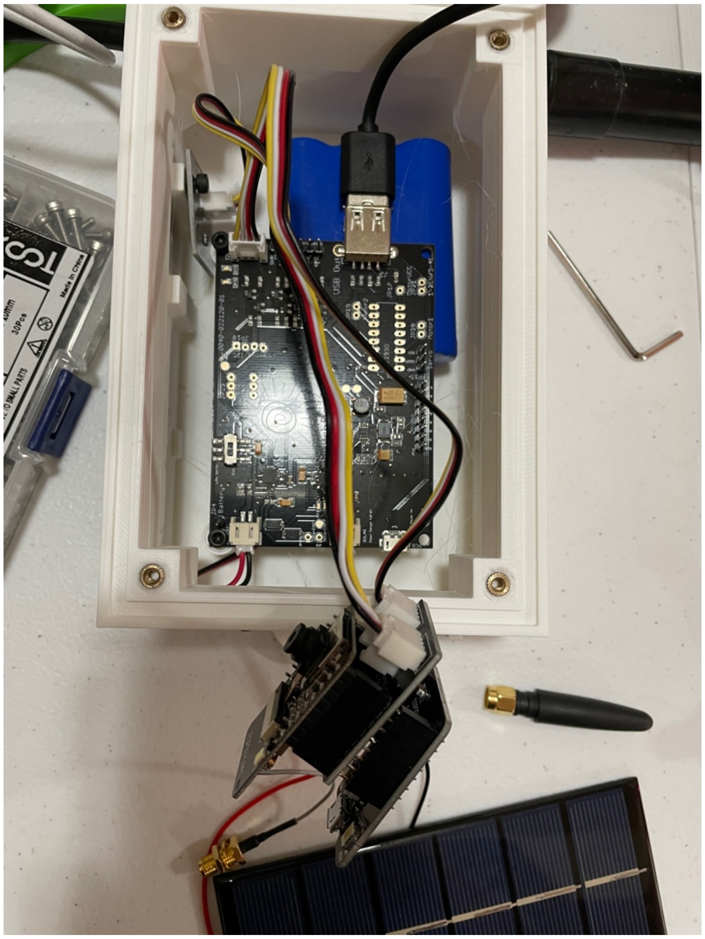
Step 13: Take the two 16mm screws and thread them through the lock blocks on the right side of the bottom box (Part I) as shown.



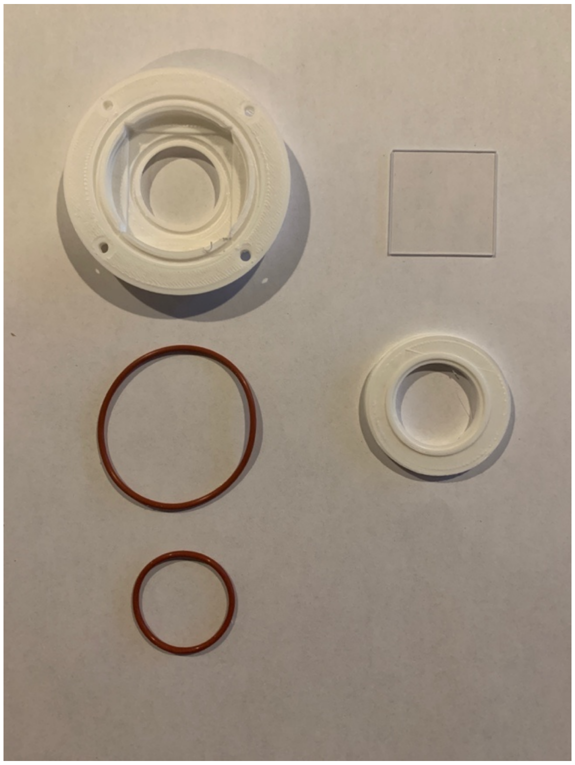
Step 14: Take the Antenna (Part A), making sure the top black Antenna stub is screwed off and connect the uf.l plug to the Camera Assembly using the uf.l socket on the ESP32-CAM board.



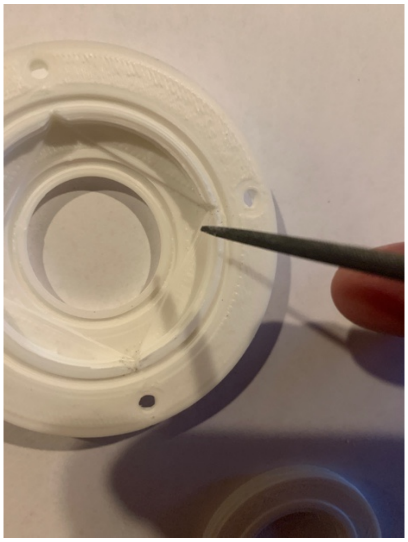
Step 15: Take the Camera Assembly (Parts G,H,I) and plug the two Grove 20cm Cables coming from the HDC1080 Temperature and Humidity Sensor (Part E) and the SunAirPlus board (Part B) into the Camera Assembly Grove connectors. All three Grove connectors on the Camera Assembly are I2C ports so you can plug them in any of the connectors (I2C is a bus, FYI).



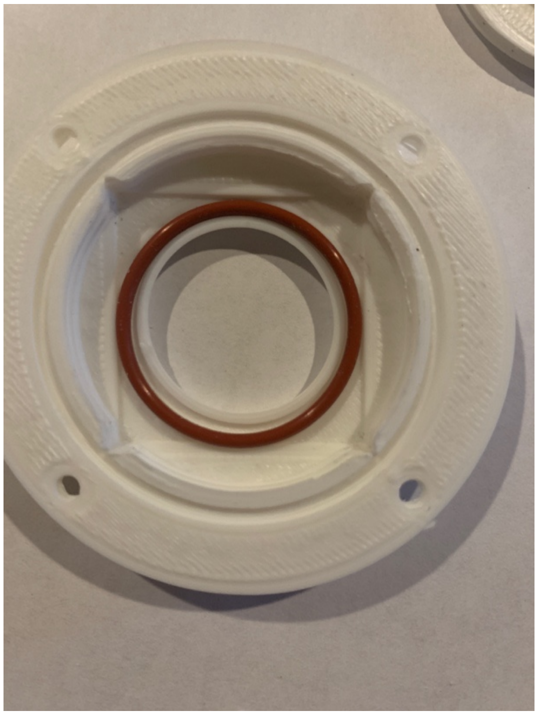
Step 16: Now layout the parts for the SkyCam Window as below (Parts L and M):



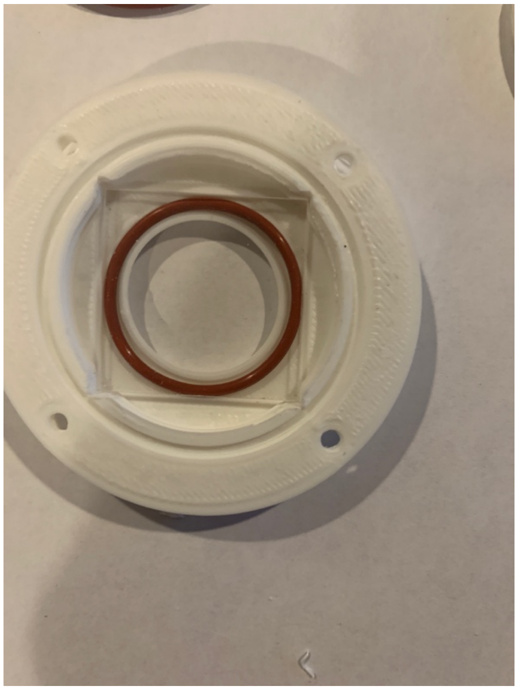
Step 17: Gently check the fit of the 30mm x 30mm x 1mm quartz window in the SkyCam Window Top (Part L). Don't force it. It should fit, but not too tight. Use a file to shave any corners on the 3D print if necessary. Don't break the window! After it fits, remove the quartz window from the SkyCam Window Top (Part L).



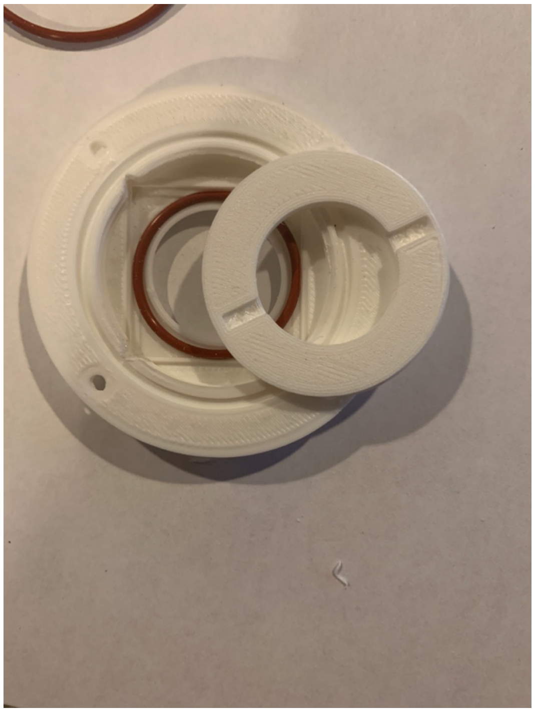
Step 18:   Place the smaller O-Ring in the SkyCam Window Top (Part L). It does not need to fit perfectly, but put as much of it the ring slot as possible.

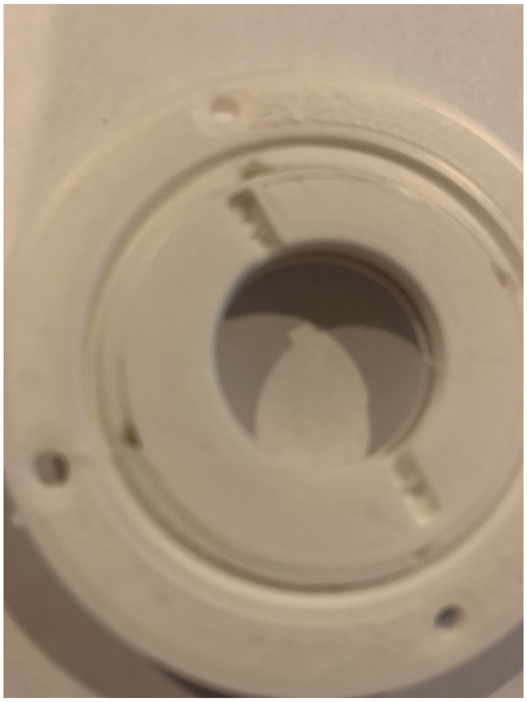


Step 19: Now gently push the quartz window into the SkyCam Window Top (Part L). Make sure it is resting on top of the O-ring.

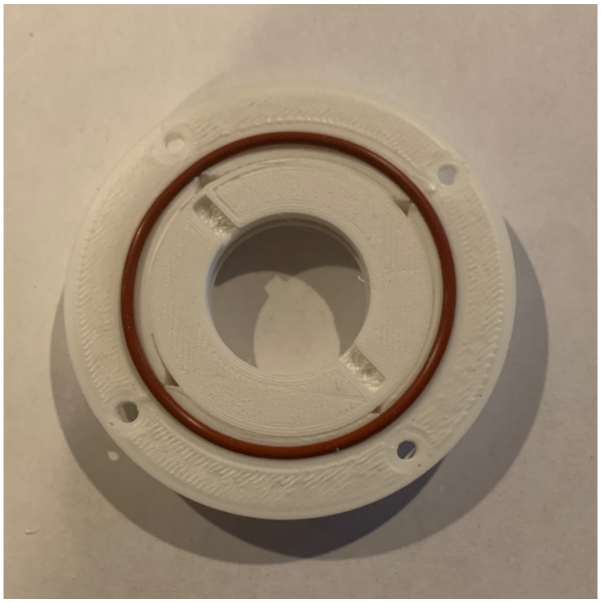


Step 20: Now take the SkyCam Window Bottom (Part M) and screw it into the SkyCam Window Top (Part L). Do not over screw this insert into the SkyCam Window Bottom (Part M). You could break the quartz window. Make it snug to flatten the O-ring inside. Make sure the insert is flush with the back of the Camera Housing.

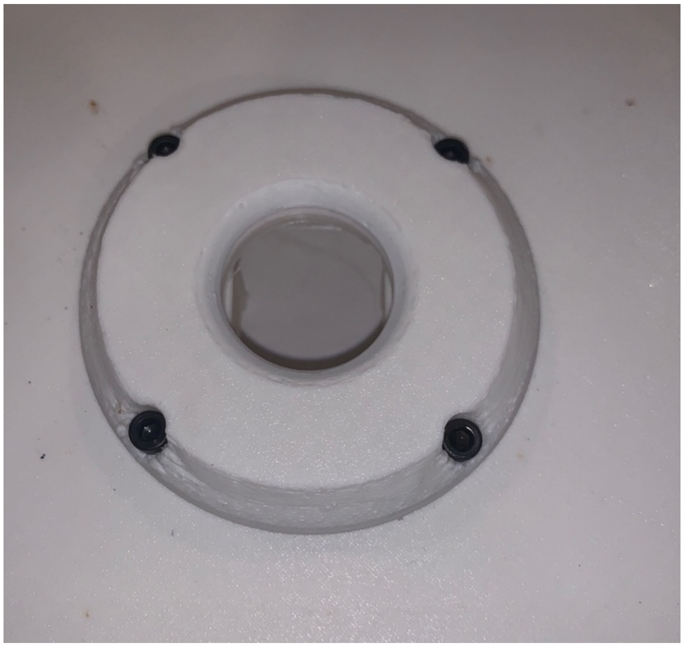




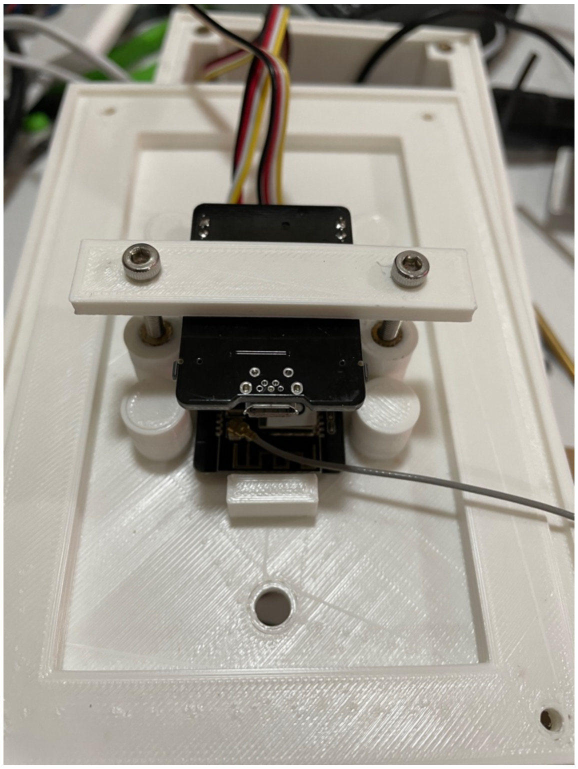
Step 21:   Now place the larger O-Ring on the ring in the SkyCam Window Top (Part L). Make as much fit of the O-Ring in the housing ring as possible.



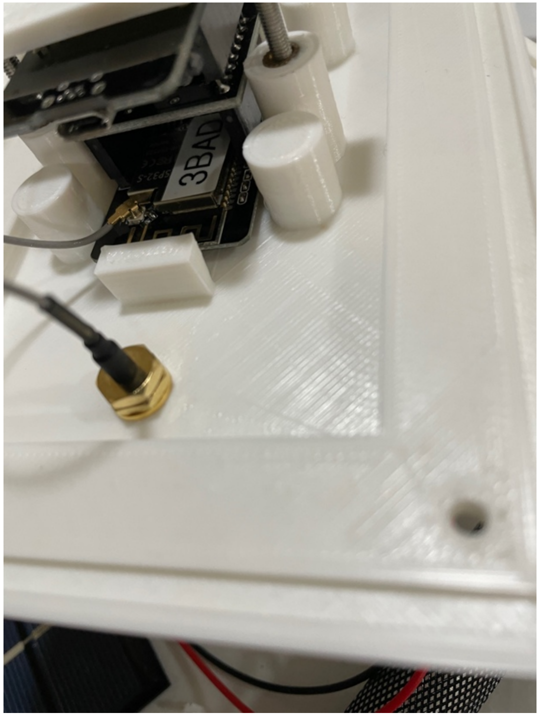
Step 19: Now secure the Camera Mount Housing Assembly (Parts D and E) with four 12mm M3 Screws. Put all four in the housing and make sure they are all started into the threads before tightening the screws. Make them tight, but don't over tighten them. You don't want to pull the Brass Inserts out of the housing.



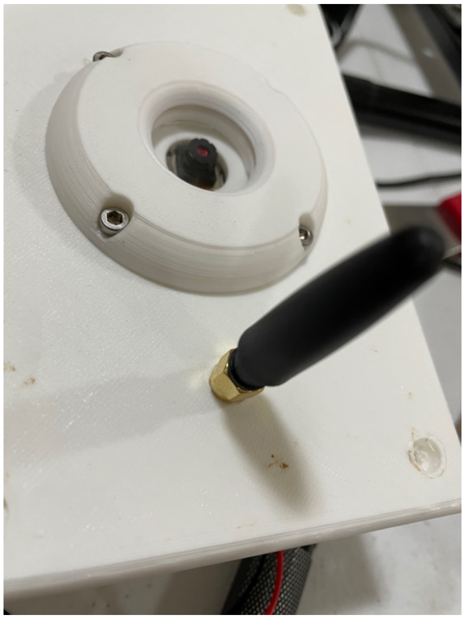
Step 19: Orient the Camera Assembly (Parts G,H,I) as shown and using the SkyCam Mount Bracket (Part Q), thread two M3 20mm screws through the two holes on the SkyCam Mount Bracket and then tight down the SkyCam Mount Bracket over the top of the Camera Assembly to hold it in place.



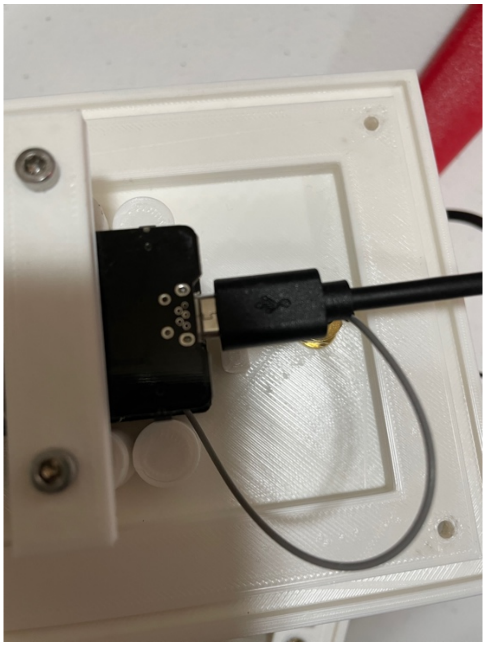
Step 20: Put the Nut and the split lock washer on the Antenna cable (Part A) and push it through the Antenna hole on the SkyCam Box Top (Part K). If it doesn’t easily slide through, either work it back and forth until it does or use a file to enlarge the hole slightly.   Welcome to 3D Printing!



Step 21: Now screw on the Antenna Stub on the top of the Antenna cable on the top of the SkyCam Box Top (Part K).



Step 22: Now plug the micro USB end of the USB cord (Part F) into the Micro USB port on the Camera Assembly (Part G,H,I).



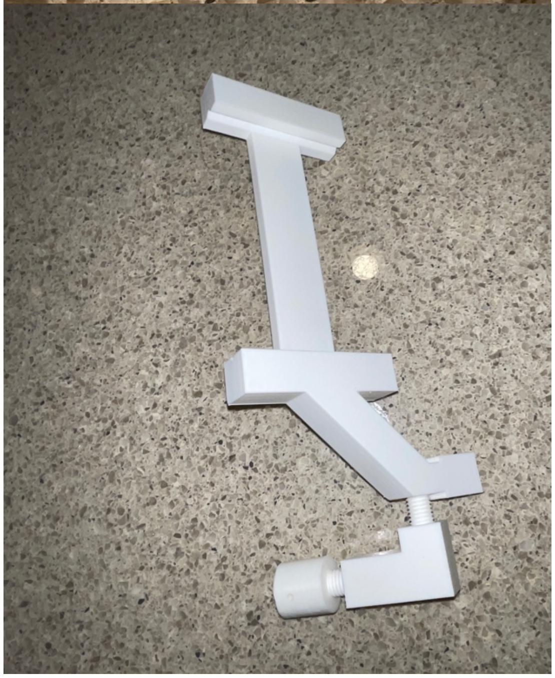
Step 22: Check to see that the Slotted Hole Plug (Part P) fits in the right side hole of the Main SkyCam Box (Part J). If it doesn't, file the Slotted Hole Plug (Part P) down until it does fit. And it should fit pretty snugly. Make sure the solar panel wires are in the slot!   Take it out after the fit test.

Step 21: Complete the Solar Bracket (Part O) and Slotted Hole Plug (Part P) assembly. Don't screw it all the way in, just until it's snug and oriented correctly as shown below.   The wire channel on the Box Plug (Part P) should be pointed down while the panel should up be at 45 degrees. The picture below shows the optional 90 Degree Threaded Block (Part N).

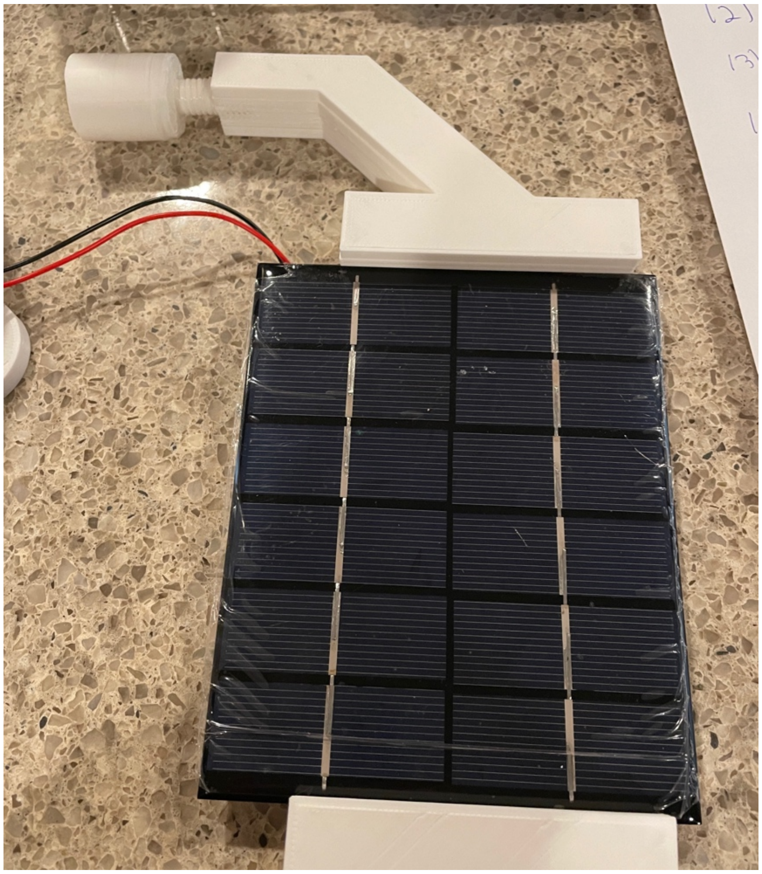
Note: Depending on how you are going to orient your solar panel (and this depends on where you want the camera pointing) you may wish to screw in the top hole on the Solar Bracket, or also use the 90 Degree Threaded Block (Part N) to get even different angles for pointing your panel at the sun. The rule of thumb for clear exposures is point your panel due south at an angle above the ground equal to your latitude at your location.   If you only get sun part of the day you may want to change.   Remember that SkyCam collects all your Solar data and sends it to the Raspberry Pi so you can use that data to optimize your placement.

If all else fails, add additional solar panels.   Here is a tutorial for just that project:

https://www.switchdoc.com/2021/08/tutorial-adding-more-solar-panels-to-skycam-and-weathersense-sensors/



Step 22: Slide the Solar Panel (Part D) into the slots on the Solar Bracket (Part O) with the wires facing down. If it is too loose, secure with tape or glue.   Your Solar Bracket Assembly is complete. If your wires are too short given your chosen position, you can solder a longer pair of wires to the panel, or you can buy a JST-2 Extender from SwitchDoc Labs (https://shop.switchdoc.com/products/jst-ph-battery-extension-cable-500mm).



Step 23:   Slide the complete Solar Bracket Assembly into the Main SkyCam Box (Part J) hole on the right side, making sure all wires are in the slot and not being pinched by the assembly.   Align the holes on both sides of the Slotted Box Plug (Part P) with the lock screws. Tighten the lock screws, again not too tightly or you will strip the 3D printed threads inside the Slotted Box Plug (Part P).

Test Your Assembled Solar SkyCam

Step 1: Point a light at the Solar Panel.   You should see a red LED turn on the SunAirPlus Board (Part B) next to the Solar Panel Input.   If you don't, check to see if you have the Solar Panel and Battery Plugs reversed.

The following steps are similar to the testing procedure in the first assembly section, but here we already have the SkyCam provisioned. If you haven't done that, go through the testing procedure in the first assembly section.

Now move over to your already set up Raspberry Pi (See the testing procedure in the first assembly section of this manual).

Step 2: **Now make sure your SkyCam sensor is on by turning the SunAirPlus2 switch to on.** You will then see something similar to this on your terminal window within about a minute:

```
pi@SwitchDocLabs:~/SDL_Pi_WeatherSense $ sudo python3 WeatherSenseMonitor.py
-----------------
WeatherSense Monitoring Software
Software Version  V009
-----------------
```

```
Pending jobs:
    readSensors (trigger: date[2021-08-20 15:56:39 PDT], pending)
    startMQTT (trigger: date[2021-08-20 15:56:39 PDT], pending)
    cleanPictures (trigger: cron[day='*', hour='3', minute='4'], pending)
    cleanTimeLapses (trigger: cron[day='*', hour='3', minute='10'], pending)
    buildTimeLapse (trigger: cron[day='*', hour='5', minute='30'], pending)
    readSensors (trigger: date[2021-08-20 15:56:39 PDT], pending)
-----------------
Scheduled Jobs
-----------------
creating new MQTT instance

######
Read Wireless Sensors
######
connecting to broker

######
Jobstore default:
Read Wireless Sensors
######
    cleanPictures (trigger: cron[day='*', hour='3', minute='4'], next run at: 2021-08-21 03:04:00
PDT)
starting 433MHz scanning
######
    cleanTimeLapses (trigger: cron[day='*', hour='3', minute='10'], next run at: 2021-08-21
03:10:00 PDT)
    buildTimeLapse (trigger: cron[day='*', hour='5', minute='30'], next run at: 2021-08-21 05:30:00
PDT)
-----------------
starting 433MHz scanning
Connected with result code 0
######

Picture ID 4182 Good Message 1 Resends: 0
CameraRotation= 90
mySkyCameraText= WeatherSense SkyCam-DE45 20-Aug-2021 15:57:36
query= INSERT INTO SkyCamPictures (cameraID, picturename, picturesize, messageID,
resends,resolution) VALUES('DE45', 'DE45_4182_2021-08-20-15-57-36.jpg', 19041, 4182, 0, 9 )
INFO msg Received: b'{"id": "DE45", "messageid": "4182", "messagetype": "4", "softwareversion":
"21", "sunairplusdevicepresent": "0", "hdc1080devicepresent": "1", "internaltemperature": "27.23",
"internalhumidity": "44.62", "currentrssi": "-57", "batteryvoltage": "0.00", "batterycurrent":
"0.00", "loadvoltage": "0.00", "loadcurrent": "0.00", "solarpanelvoltage": "0.00",
"solarpanelcurrent": "0.00", "gndrreboots": "0"}'
in INFO with SQL

Picture ID 47539 Good Message 2 Resends: 0
CameraRotation= 90
mySkyCameraText= WeatherSense SkyCam-26FD 20-Aug-2021 15:57:44
query= INSERT INTO SkyCamPictures (cameraID, picturename, picturesize, messageID,
resends,resolution) VALUES('26FD', '26FD_47539_2021-08-20-15-57-44.jpg', 117097, 47539, 0, 13 )
INFO msg Received: b'{"id": "26FD", "messageid": "47539", "messagetype": "4", "softwareversion":
"18", "sunairplusdevicepresent": "1", "hdc1080devicepresent": "1", "internaltemperature": "27.42",
"internalhumidity": "51.92", "currentrssi": "-70", "batteryvoltage": "3.75", "batterycurrent":
"138.40", "loadvoltage": "4.98", "loadcurrent": "100.20", "solarpanelvoltage": "4.70",
"solarpanelcurrent": "65.20", "gndrreboots": "1146"}'
in INFO with SQLWhat does the JSON from the WeatherSense SkyCam mean?
```

The WeatherSenseMonitor.py python3 program waits on the MQTT Packets from the SkyCam ls 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.

The above list shows two SkyCam cameras transmitting pictures.   SkyCam-DE45 and SkyCam-26FD.

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

What does this mean?

```
query= INSERT INTO SkyCamPictures (cameraID, picturename, picturesize, messageID,
resends,resolution) VALUES('26FD', '26FD_47539_2021-08-20-15-57-44.jpg', 117097, 47539, 0, 13 )
```

This query printout shows the SkyCam picture being entered in the database. The values are:

- cameraID: Unique ID for every SkyCam
- picturename: The name of the current picture
- picturesize: size in bytes of the current picture
- messageID: the Message ID (Starts at zero when you initialize a SkyCam) for the picture
- resends: How many chunks were resent for this message (SkyCam breaks up the picture into smaller chunks and resends them if they weren't received by the Pi correctly)
- resolution: What resolution of picture has been sent – see the "Programming SkyCam" later on in this manual.

```
INFO msg Received: b'{"id": "26FD", "messageid": "47539", "messagetype": "4", "softwareversion": "18",
"sunairplusdevicepresent": "1", "hdc1080devicepresent": "1", "internaltemperature": "27.42",
"internalhumidity": "51.92", "currentrssi": "-70", "batteryvoltage": "3.75", "batterycurrent": "138.40",
"loadvoltage": "4.98", "loadcurrent": "100.20", "solarpanelvoltage": "4.70", "solarpanelcurrent": "65.20",
"gndrreboots": "1146"}
```

The values and units of the are:

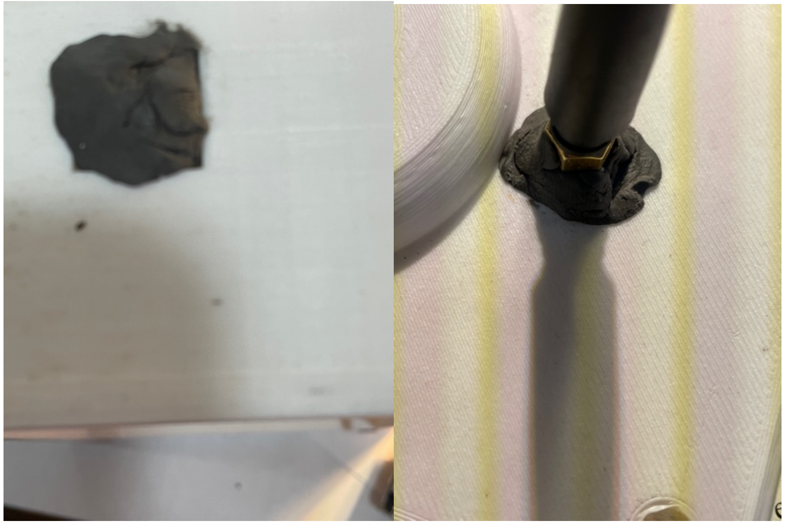
- id – SkyCam ID
- messageid: the Message ID (Starts at zero when you initialize a SkyCam) for the picture
- messagetype: 4 – The SkyCam Information Packet
- softwareversion: Which version of the C/C++ software is your SkyCam running
- sunairplusdevicepresent: 1 if the SunAirPlus is present and 0 if it is not detected
- hdc1080devicepresent: 1 if the HDC1080 is detected and 0 if it is not detected
- internaltemperature: internal box temperature. In degrees C
- internalhumidity: internal box humidity. In percent relative humidity
- currentrssi: Current Wifi signal SkyCam strength, more negative is weaker. This can help you debug range problems.
- 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)
- gndreboots: How many reboots have been called by the software (very rare in V19 and above).

You now have completed testing of your WeatherSense SkyCam sensor.

Final Assembly

Step 1: Take the SkyCam Box Top (Part K) and attach it to the SkyCam Box Bottom (Part J) using 4 M3 12mm screws. Start one screw, then move to the opposite corner and start the next screw, then to your right for the further screw and finally start a screw in the other corner. Kind of like putting on a tire. 3D Prints are often a bit skewed.

Step 2: Now we will use Duct Seal to waterproof all the screws and fill in any holes. One exception, leave the small hole on the bottom of SkyCam open for equalization. Prevents a lot of condensation in some weather conditions. Of course, you can check this with the on board temperature and humidity sensor in SkyCam. Duct Seal can be easily removed if you need to get inside of your SkyCam. Use just enough Duct Seal fully cover the hole. Don't forget to put a bead around your stub antenna on the top and push Duct Seal into the slot where the Solar Panel wires exit the SkyCam Box.



You are complete! Put your box outside and see those weather pictures!

The Dash_App and SkyCam

You have SkyCam running now. Go ahead and either start SkyWeather2 (using instructions in the manual – you must be running at least V027 to have access to SkyCam images) or you can use SDL_Pi_WeatherSense.

Using WeatherSense to View SkyCam

Install the software from the github repository. You need to have at least V009 of the software (V010 is the current version as of the writing of this manual).

```
cd
git clone https://github.com/switchdoclabs/SDL_Pi_WeatherSense
```

Next, start the main program running.

```
cd SDL_Pi_WeatherSense
sudo python3 WeatherSenseMonitor.py
```

Next, open up another terminal window and do these commands:

```
cd
cd SDL_Pi_WeatherSense
cd dash_app
cd sudo python3 index.py
```

Open another terminal window and type:

```
cd
cd SDL_Pi_SkyWeather2
cd dash_app
sudo python3 index.py
```

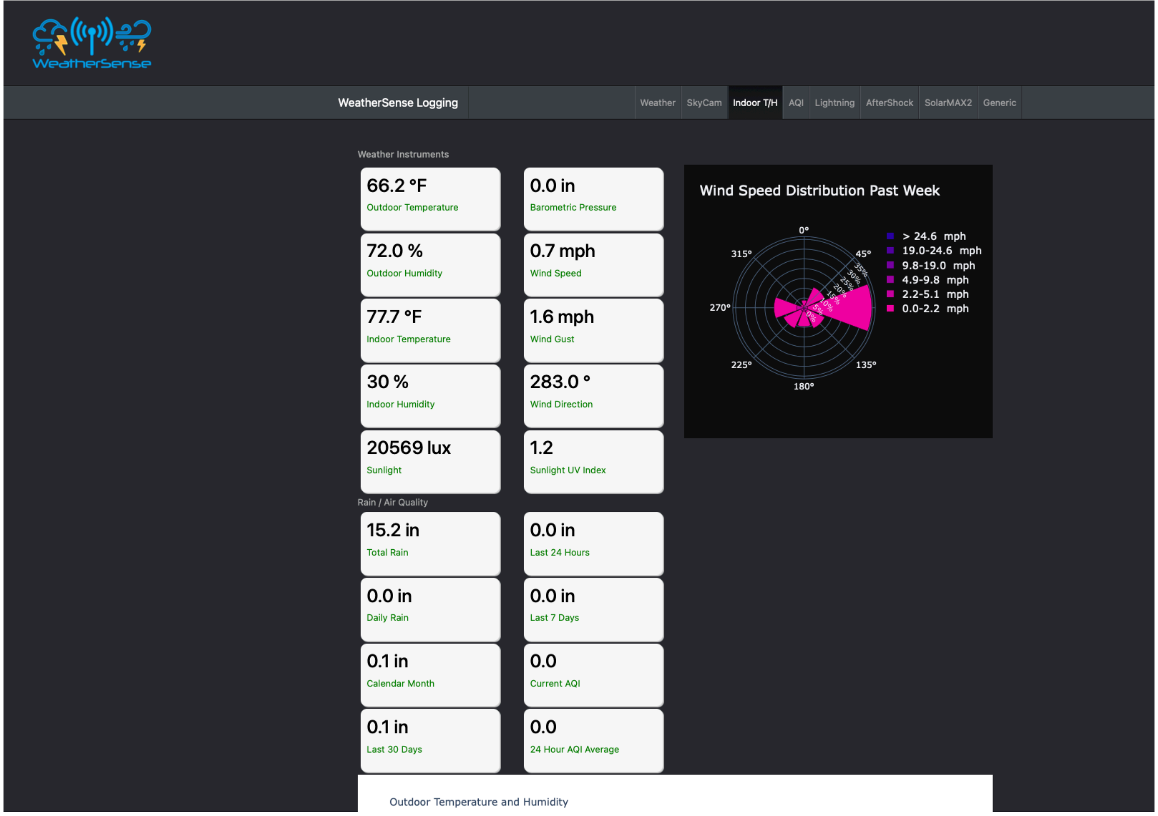
Now open up a browser window (either on the Raspberry Pi or on another computer on your local WiFi network) and enter this URL:

On your Raspberry Pi: http://127.0.0.1:8050/

On another computer type: http://xxx.xxx.xxx.xxx:8050/

Where "xxx.xxx.xxx.xxx" is the IP address of your Raspberry Pi that you wrote down above.

You will see this screen on your browser (or one of the other pages in the app):



Weather Instruments

66.2 °F
Outdoor Temperature

0.0 in
Barometric Pressure

72.0 %
Outdoor Humidity

0.7 mph
Wind Speed

77.7 °F
Indoor Temperature

1.6 mph
Wind Gust

30 %
Indoor Humidity

283.0 °
Wind Direction

20569 lux
Sunlight

1.2
Sunlight UV Index

Rain / Air Quality

15.2 in
Total Rain

0.0 in
Last 24 Hours

0.0 in
Daily Rain

0.0 in
Last 7 Days

0.1 in
Calendar Month

0.0
Current AQI

0.1 in
Last 30 Days

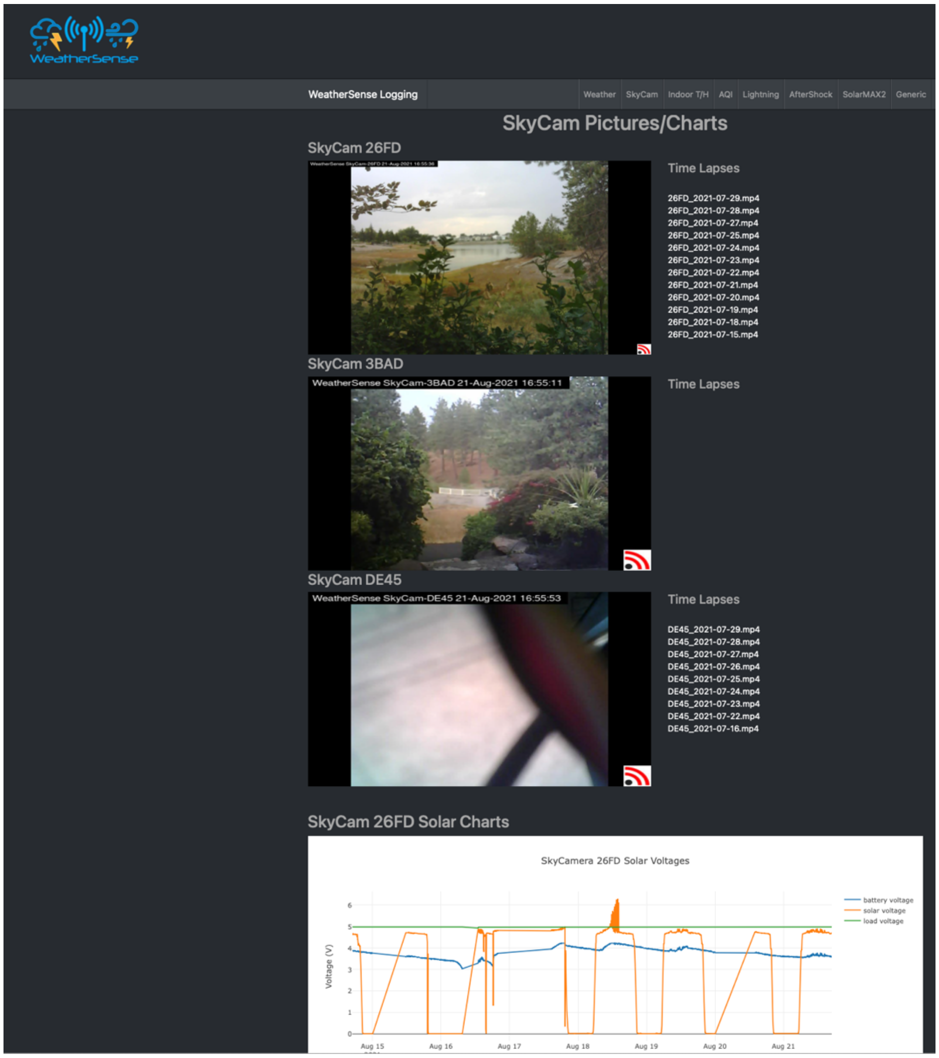
0.0
24 Hour AQI Average

Outdoor Temperature and Humidity

If you don't have a WeatherRack2, then this screen will be filled with zeros.

Click on the SkyCam tab to see your cameras.   Note that we have a bunch of SkyCams running. You should only see yours.   WeatherSenseMonitor builds time lapses at 5:00am local time.   Come back tomorrow to see your Time Lapse!

See the SkyCam 26FD solar graphs? All sorts of good information there. Your charts will fill in below the 26FD (alphabetical by SkyCam ID).



SkyCam Pictures/Charts

SkyCam 26FD

WeatherSense SkyCam-26FD 21-Aug-2021 16:55:36

Time Lapses

26FD_2021-07-29.mp4
26FD_2021-07-28.mp4
26FD_2021-07-27.mp4
26FD_2021-07-25.mp4
26FD_2021-07-24.mp4
26FD_2021-07-23.mp4
26FD_2021-07-22.mp4
26FD_2021-07-21.mp4
26FD_2021-07-20.mp4
26FD_2021-07-19.mp4
26FD_2021-07-18.mp4
26FD_2021-07-15.mp4

SkyCam 3BAD

WeatherSense SkyCam-3BAD 21-Aug-2021 16:55:11

Time Lapses

SkyCam DE45

WeatherSense SkyCam-DE45 21-Aug-2021 16:55:53

Time Lapses

DE45_2021-07-29.mp4
DE45_2021-07-28.mp4
DE45_2021-07-27.mp4
DE45_2021-07-26.mp4
DE45_2021-07-25.mp4
DE45_2021-07-24.mp4
DE45_2021-07-23.mp4
DE45_2021-07-22.mp4
DE45_2021-07-16.mp4

SkyCam 26FD Solar Charts

Changing Orientation for your SkyCam Camera

Depending on how you oriented your SkyCam top, you may have your picture upside down (or in the case of some cameras, off by 90 degrees, no kidding), so we have provided you with a method for correcting this.

In both SkyWeather2 and the WeatherSense software there is a file called config.py

First find that 4 digit hex number from your SkyCam (you saved it earlier in this manual – our example unit was “3BAD”). Open config.py up in your favorite editor and add a new line to the the following code in config.py:

```
############
# SkyCam Remote configuration
```

```
############
#configure SkyCam Remote camera
#default
DefaultCameraRotation = 90
SkyCamRotationArray = {}
SkyCamRotationArray["3BAD"] = 180
#SkyCamRotationArray["F329"] = 0
#SkyCamRotationArray["26FD"] = 270
```

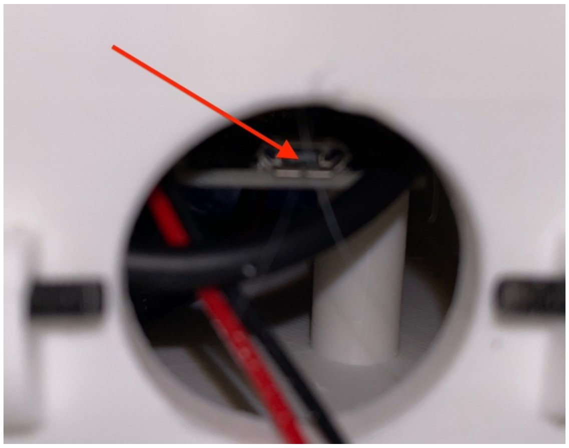
After editing this file, restart your WeatherSenseMonitor.py program.   This will fix the orientation.

The rotation takes effect after the next picture comes in from SkyCam! Be patient.

Charging SkyCam

At the beginner or sometimes after a long cloudy and rainy spell, you may want to charge your SkyCam from the wall.   We have provided a method for doing that.

Step 1: Take the Solar Panel Assembly off of SkyCam by loosening the set screws and gently it pulling out. **Put your hand on the Slotted Hole Plug (Part P) and NOT on the rest of the assembly**. You could snap off the threaded screws. Look inside.



Step 2: Plug in a micro USB cable and connect the other end to your power supply.

It can take as many as 15 hours to charge large batteries. You will see the red charging light inside the case (on SunAirPlus next to the Solar Panel plug). The LED next to the red LED will turn green when fully charged.

Programming SkyCam

This is not a section on how to modify your SkyCam software. We do provide the full source code (https://github.com/switchdoclabs/SDL_ESP32_WeatherSenseSkyCam) but this requires you to install the Arduino IDE, install the ESP32 libraries and is beyond the scope of this manual. We encourage you to learn how to do this! Start with the ESP32 Adafruit Arduino IDE tutorial.

This section talks about being able to change behavior and many parameters inside your SkyCam without having to reprogram the SkyCam.

See the testSkyCamRemote.py software in either SkyWeather2 or WeatherSense software.

This software will watch for your SkyCam and then send it a command when it is found. This works because after sending a picture, your SkyCam stays on the channel for a couple of seconds before turning off the WiFi and going to sleep to save power.

To send a message to a SkyCam is a three step process.

Here are the currently supported command (from the python):

```
#commands

MQTTUPDATEPARAM = 10
MQTTCYCLECHANGE = 11
MQTTSTARTDELAY = 12
MQTTTURNOFFBLINK = 13
MQTTBLINKXTIMES = 14
MQTTSETTODEFAULTS = 15
MQTTREBOOT = 16
MQTTRESOLUTION = 17
MQTTERASEMEMORY = 18
```

First the appropriate part of the testSkyCamRemote.py code:

```
# main program

# what ID to test
cameraID = "3BAD"
#cameraID = "F329"
#cameraID = "DE45"
#cameraID = "26FD"
#cameraID = "+"   #sends to all cameras
# this command will be sent after an INFO messagetype 4  from cameraID
#sendWhatCommand = MQTTERASEMEMORY
#sendWhatCommand = MQTTCYCLECHANGE
#sendWhatCommand = MQTTRESOLUTION
sendWhatCommand = MQTTBLINKXTIMES
#sendWhatCommand = MQTTUPDATEPARAM
#sendWhatCommand = MQTTSTARTDELAY
#sendWhatCommand =  MQTTREBOOT
```

1) You change the cameraID variable to match your SkyCam ID (that pesky number again)
2) Uncomment or change what command you want sent.
3) Change the parameters in the command that you are sending. For example, the function MQTTBLINKXTIMES consists of:

```
   if (messageType == MQTTBLINKXTIMES):
```

```
 # send blink message
     myMessage = {
         "messagetype": MQTTBLINKXTIMES,
         "myip": myIP,
         "length": 300,
         "count": 3
         }
```

"length" means the duration of the blink in milliseconds (and the gap between two blinks). "count" means the number of blinks you want.

The program will keep sending the message to the SkyCam, waiting each time for a new picture to be sent. Break the program to stop it running.

What You Can Change on the SkyCam Remotely

We have built in a protocol using MQTT that allows you to change a variety of parameters on the SkyCam without reprogramming the SkyCam (you have all the open source software to do that if you wish, however!). We also provide the example Python3 software to accomplish these tasks. These settable parameters are:

- MQTTCYCLECHANGE - Change the delay between pictures in seconds - defaults to 50 seconds
- MQTTSTARTDELAY - Change delay in seconds from initialization of camera to the picture. This helps with setting the exposure of the camera - long delay, better high light exposure
- MQTTTURNOFFBLINK - Turns off the default bright light blink before the camera is invoked. Saves power too!
- MQTTBLINKXTIMES - Blink the bright light X number of times. Great for identifying the SkyCam units.
- MQTTSETTODEFAULTS - Resets SkyCam to factory default
- MQTTREBOOT - Forces the SkyCam unit to reboot
- MQTTERASEMEMORY - Clears all non-volatile memory and reboots.
- MQTTRESOLUTION - Changes the resolution of the picture. Defaults to SVGA (800x600). Can set any of these values: QVGA|CIF|VGA|SVGA|XGA|SXGA|UXGA
    - FRAMESIZE_QVGA for 320x240 - value 5
    - FRAMESIZE_CIF for 400x296 - value 6
    - FRAMESIZE_VGA for 640x480 - value 8
    - FRAMESIZE_SVGA for 800x600 - value 9 (Default)
    - FRAMESIZE_XGA for 1024x768 - value 10
    - FRAMESIZE_SXGA for 1280x1024 - value 12
    - FRAMESIZE_UXGA for 1600x1200 - value 13
- MQTTUPDATEPARAM - Sets all the initialization parameters for the camera. The default list is:
  DEFAULTSENSORSTRING "0;0;0;0;1;1;0;1;0;0;300;1;0;0;0;1;1;1;0;0;1;0;"
  Each location for the sensor string corresponds to the table below:

SkyCam Camera Parameters

Function	Meaning	Values
set_brightness()	Set brightness	-2 to 2
set_contrast()	Set contrast	-2 to 2
set_saturation()	Set saturation	-2 to 2
set_special_effect()	Set a special effect	0 – No Effect 1 – Negative 2 – Grayscale 3 – Red Tint 4 – Green Tint 5 – Blue Tint 6 – Sepia
set_whitebal()	Set white balance	0 – disable 1 – enable
set_awb_gain()	Set white balance gain	0 – disable 1 – enable
set_wb_mode()	Set white balance mode	0 – Auto 1 – Sunny 2 – Cloudy 3 – Office 4 – Home
set_exposure_ctrl()	Set exposure control	0 – disable 1 – enable
set_aec2()		0 – disable 1 – enable
set_ae_level()		-2 to 2
set_aec_value()		0 to 1200
set_gain_ctrl()		0 – disable 1 – enable
set_agc_gain()		0 to 30
set_gainceiling()		0 to 6
set_bpc()		0 – disable 1 – enable
set_wpc()		0 – disable 1 – enable
set_raw_gma()		0 – disable 1 – enable
set_lenc()	Set lens correction	0 – disable 1 – enable
set_hmirror()	Horizontal mirror	0 – disable 1 – enable
set_vflip()	Vertical flip	0 – disable 1 – enable
set_dcw()		0 – disable 1 – enable
set_colorbar()	Set a colorbar	0 – disable 1 – enable

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.