Arduino Weather Station And Ascom Observing Conditions

This is a weather station using Arduino with an Ascom Observing Conditions driver on top so we can plug into various Astronomy Software packages.

Basic idea is to have Arduino be the microcontroller which is reading various sensors and reporting weather conditions. Everything is enclosed in a weather sealed box and connected to the computer via a USB cable. The same USB cable provides power to Arduino and all sensors. Sensors contemplated are:

1. Ambient Temperature
2. Sky Temperature (This along with Ambient temperature can tell us when clouds are present. Some calibration will be needed)
3. Barometric Pressure
4. Humidity (This along with Ambient temperature can tell us dew point as well)
5. Light level (Can tell us when the sky is dark)
6. Rain levels
7. Wind Speed and Direction

Acquiring the hardware

There are various bits and bobs we need to acquire to build this system out:

1. Arduino Uno: https://store-usa.arduino.cc/products/a000066?utm_source=redirects&utm_medium=store.arduino.cc&utm_campaign=303_Redirects
2. SparkFun Weather Shield: https://www.sparkfun.com/products/12081
3. Sparkfun Weather Meters: https://www.sparkfun.com/products/8942
4. Two RJ11 connectors to connect the Weather Meters to the Weather Shield: https://www.sparkfun.com/products/132

5. Melexis MLX90614 IR contactless temperature sensor: https://www.sparkfun.com/products/9570
6. A small breadboard to wire up the MLX90614 to the weather shield and Arduino: https://www.sparkfun.com/products/12043
7. 4.7K Ω pull up resistors for MLX90614 SDA and SCL pins
8. Heat Shrink tubes for Wiring up MLX90614
9. Patch cables: https://www.sparkfun.com/products/13870
10. Big Red Box to enclose everything: https://www.sparkfun.com/products/11366
11. Cable glands: https://www.amazon.com/gp/product/B01GJ03AUQ/ref=oh_aui_detailpage_o01_s00?ie=UTF8&psc=1
12. Stackable header kit to wire up weather shield to Arduino: https://www.amazon.com/gp/product/B00PCCWEJG/ref=oh_aui_detailpage_o02_s00?ie=UTF8&psc=1
13. USB cable Type A to Type B of desired length
14. Solder gun and Solder wire

Preparing the Hardware

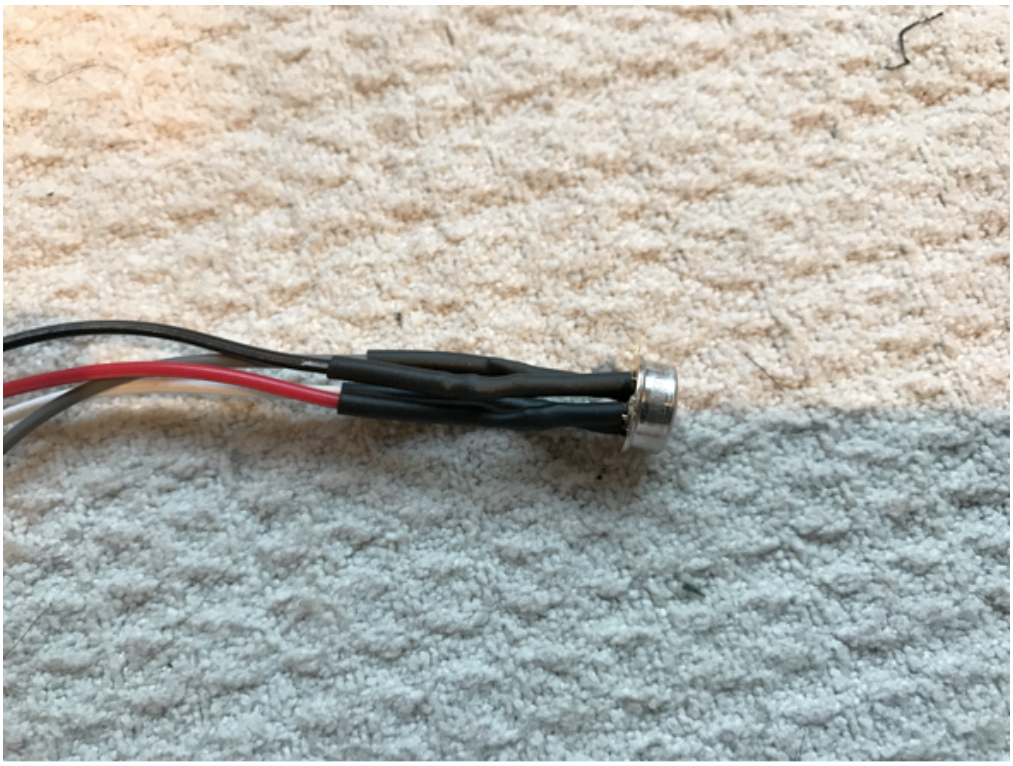
Once we have everything in our hands, lets wire them up. The weather shield has a light sensor. But the unit is going to be sitting inside the Big Red Box (BRB) and won't see the light. So we need some sort of a window so light can come in a give the sensor an idea of the light levels outside. We need an optical window.

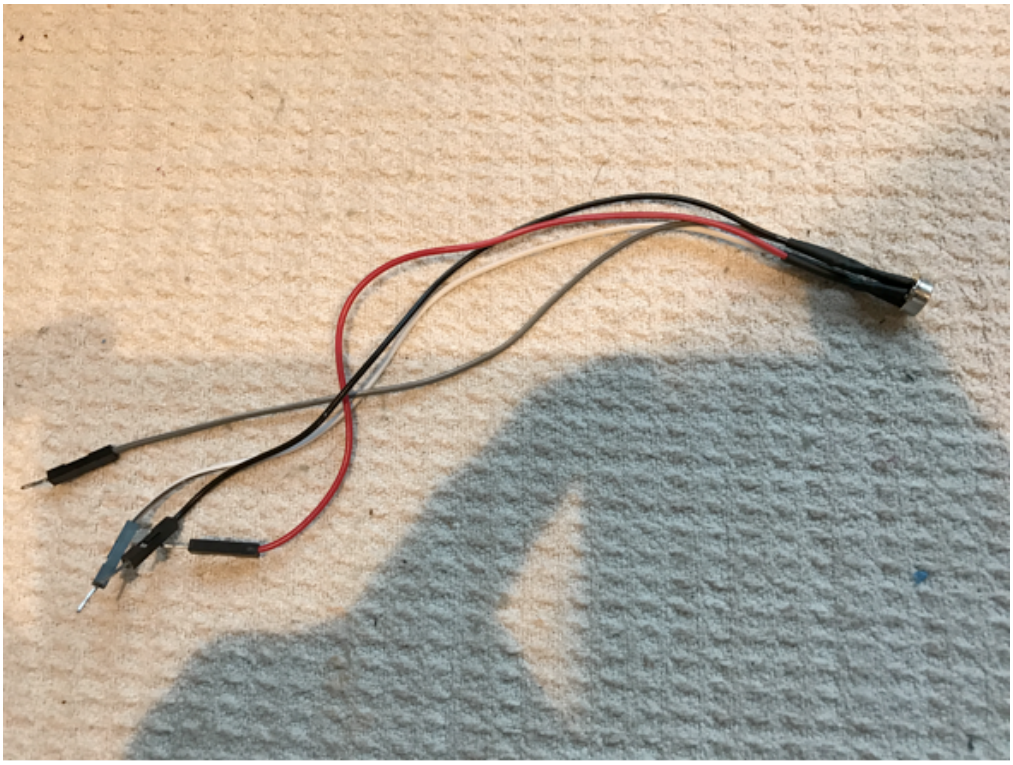
A 3" hole on the lid with a 2" round acrylic window suffices. The acrylic window was glued using superglued and then caulked around.

This should now let enough light which will let the light sensor, after some calibration of the voltage levels to give us a good idea of when it's day, twilight, night, full moon etc.

Next we want to prepare the MLX90614. We want the MLX90614 to be facing the sky and on the outside. So we have to figure out how to still wire it up while protecting the rest from the elements. Idea is to extend the pins by soldering jumper wires to it (one side of the jumper wire stripped for soldering to the pins) and then wrapping everything up in heat shrink tubes so we can pass through a cable gland.

Next we use 4 of the jumper cables and solder it to the four pins of the MLX90614. I used red for VDD, black for VSS, White for SDA and Brown for SCL. Heat shrink tubes are used on to to pack everything and protect the solder joints.





One more heat tube on top to bundle everything into one cable interface

Using a PG9 cable gland for and the right sized hole on the top we create an entry for the MLX90614 cabled while keeping the sensor itself outside so it can accurately measure Sky and Ambient temperatures.

The window you see on the right was my first attempt at letting enough light in for the light sensor, but this proved inadequate.

Next we prepare the entry for the USB cable. This time I am using a PG16 cable gland on the bottom.I still had to sand the rubber housing around the USB Type B pin so it would pass through

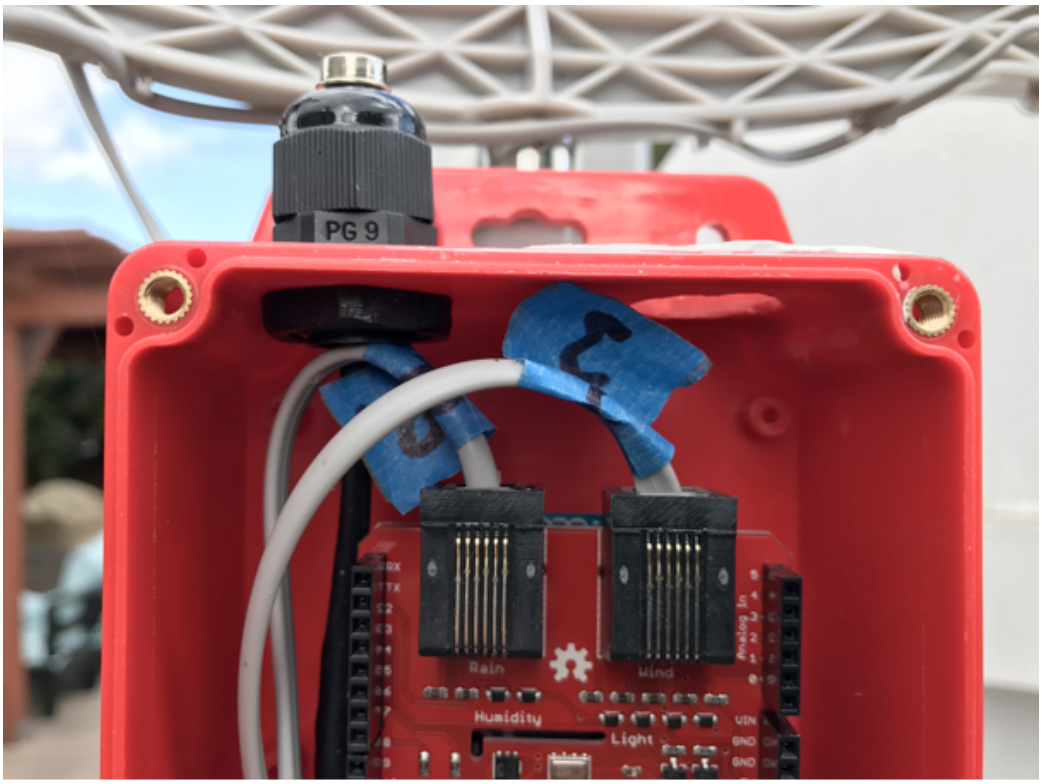
The cable itself is too thin to form a proper. So I used some Sugru to give it more girth. I am using PVC pipes to pass the USB cable from the Weather Station Pole, under the deck, to the USB hub to connect to the computer. The mouth of the PVC pipe is sealed up using Sugru as well.

Sam technique used for the rain and wind RJ11 cables that come in. The cable gland used here is a PG11 so the RJ11 jack can pass through

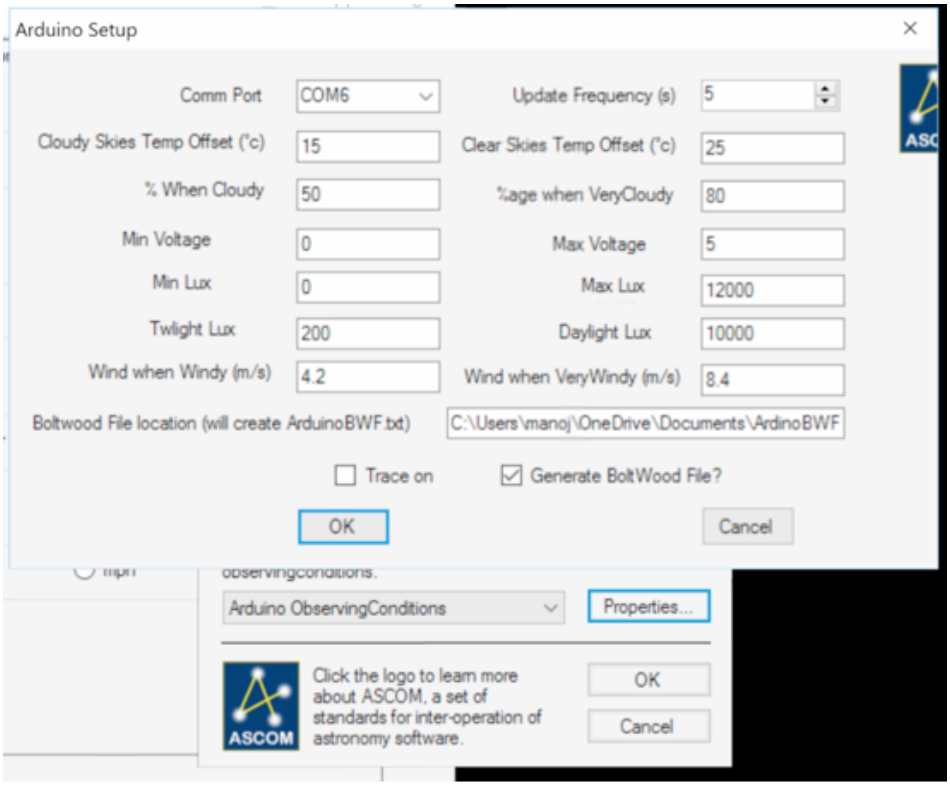
The Arduino and Weather Shield (which has stacking headers and RJ11 receptors soldered on), are velcored inside the box. There is a small breadboard to wire up the 10k pull up resistors for the MLX90614. See wiring instructions on how to do this: https://learn.adafruit.com/using-melexis-mlx90614-non-contact-sensors/wiring-and-test


Status
Pressure
Soft
Hard
Serial
sparkfun
Weather Shield

Next, plug in the RJ11 wind and rain sensor cables and close the box up. You are all done!



The software is fairly simple and self-explanatory. There is an ASCOM Arudino Observing Conditions driver that you install and choose in your imaging software. There are various calibration settings in there which need to be set.



Note the Cloudy Skies and Clear Skies Temp Offset. This is asking for what the difference in ambient temperature and sky temperature will be when the sky is completely cloudy and completely clear. It assumes a linear slope from those to points. The driver will automatically clip the %age between 0 and 100, so it's best to measure the cloudy sky temp offset when the sky is clouded over with high clouds on a cold day. This gives the highest temp difference possible when the sky is cloudy. Similary for clear skies best to measure in the night when you have clear skies and a cold day. This gives the lowest temp difference possible when the sky is clear.

The next two ask when the driver should report the sky is “Cloudy” or “Very Cloudy”.

Min. Voltage and Max. Voltage refer to what the light sensor reports. To find these you will, unfortunately have to connect to the Arduino via a serial monitor and read the raw sensor voltage reported. The command for this is “SB” with a new line ending. Measure when you have a clear dark moonless night and when you a clear fully sunny day. While the sensor response to luminance is not linear the driver assumes linearity based on these two values. It should suffice, for astronomy purposes.

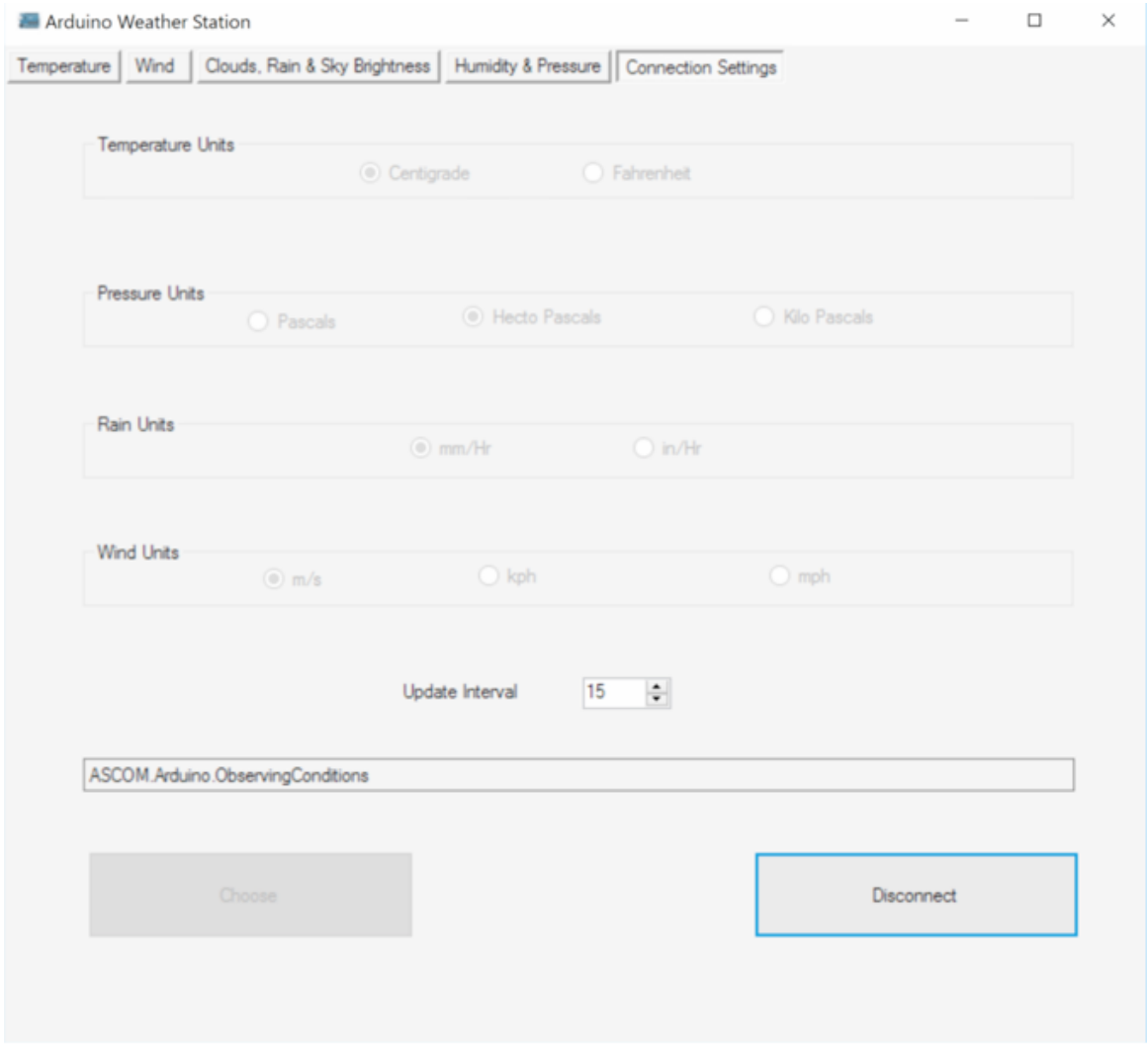
The scaling from voltage to LUX values which is what the ASCOM driver needs to report, is based on the next two values. This sets the floor and ceiling of the LUX range. Again, as mentioned before, there is a linear conversion from the voltage scale to the LUX scale.

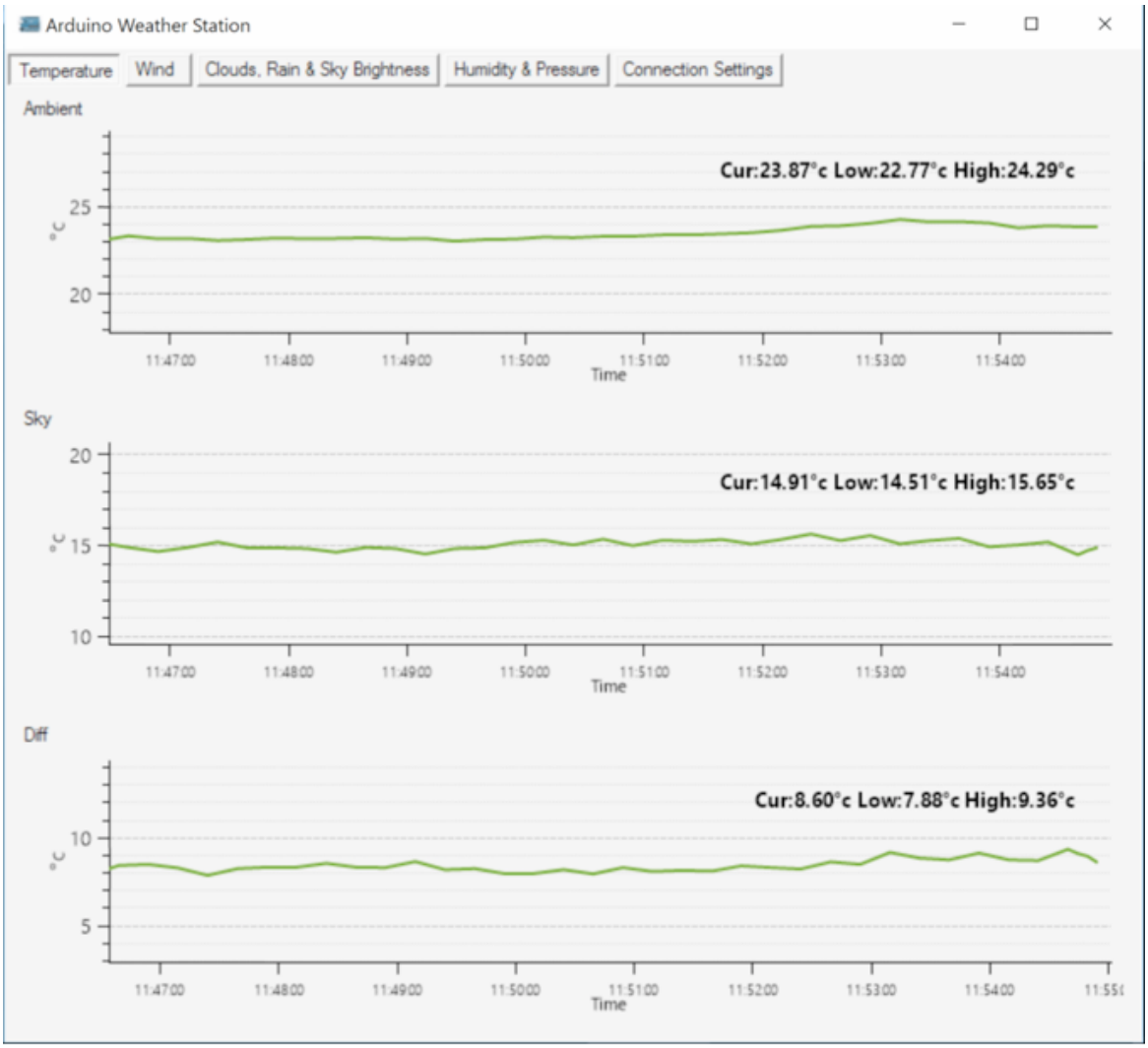
Next you set what defines twilight and what defines daylight. This is “Light” and “Very Light” conditions.

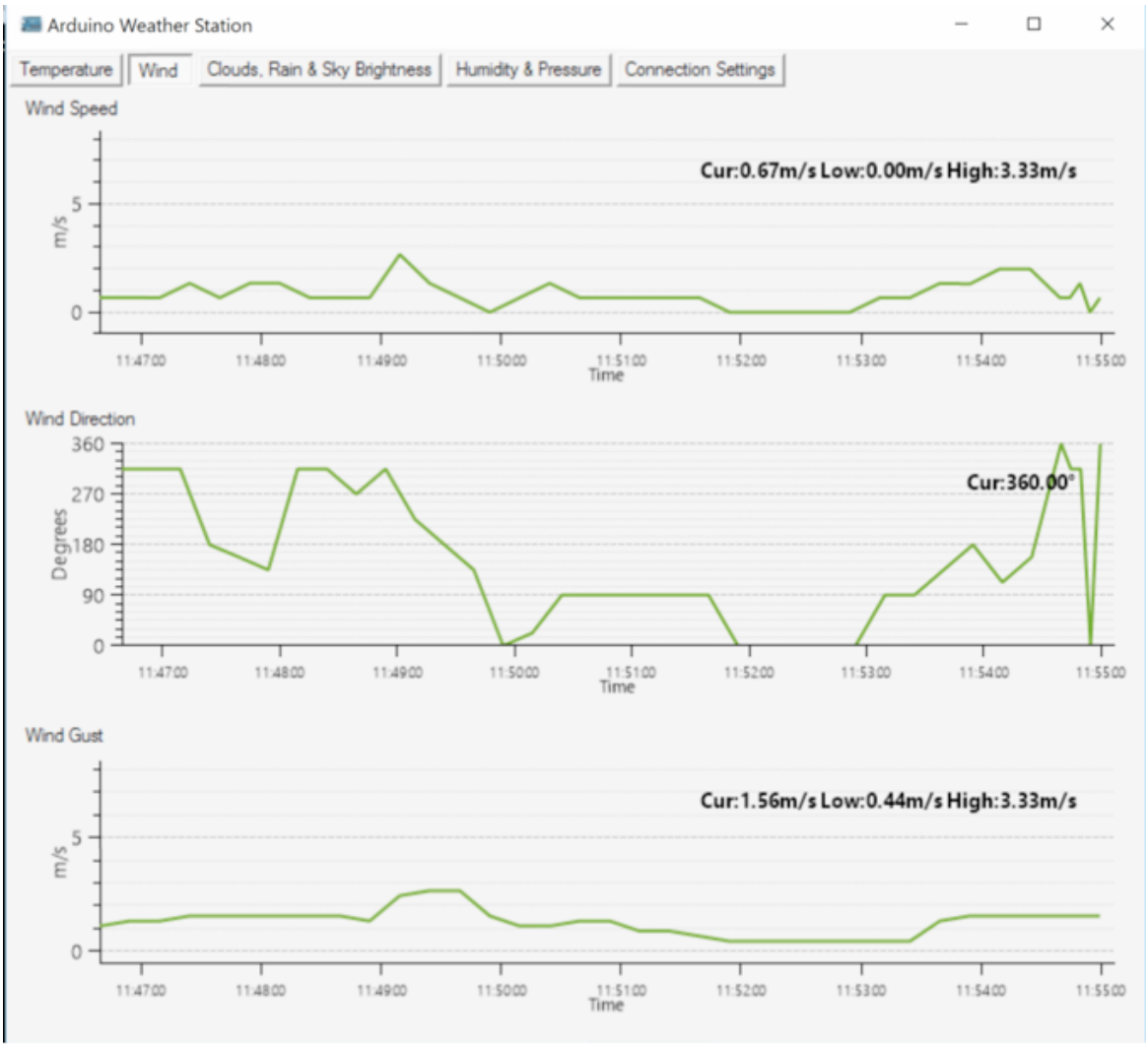
“Windy” and “Very Windy” conditions are defined next. Default values should suffice, but change up as you desire.

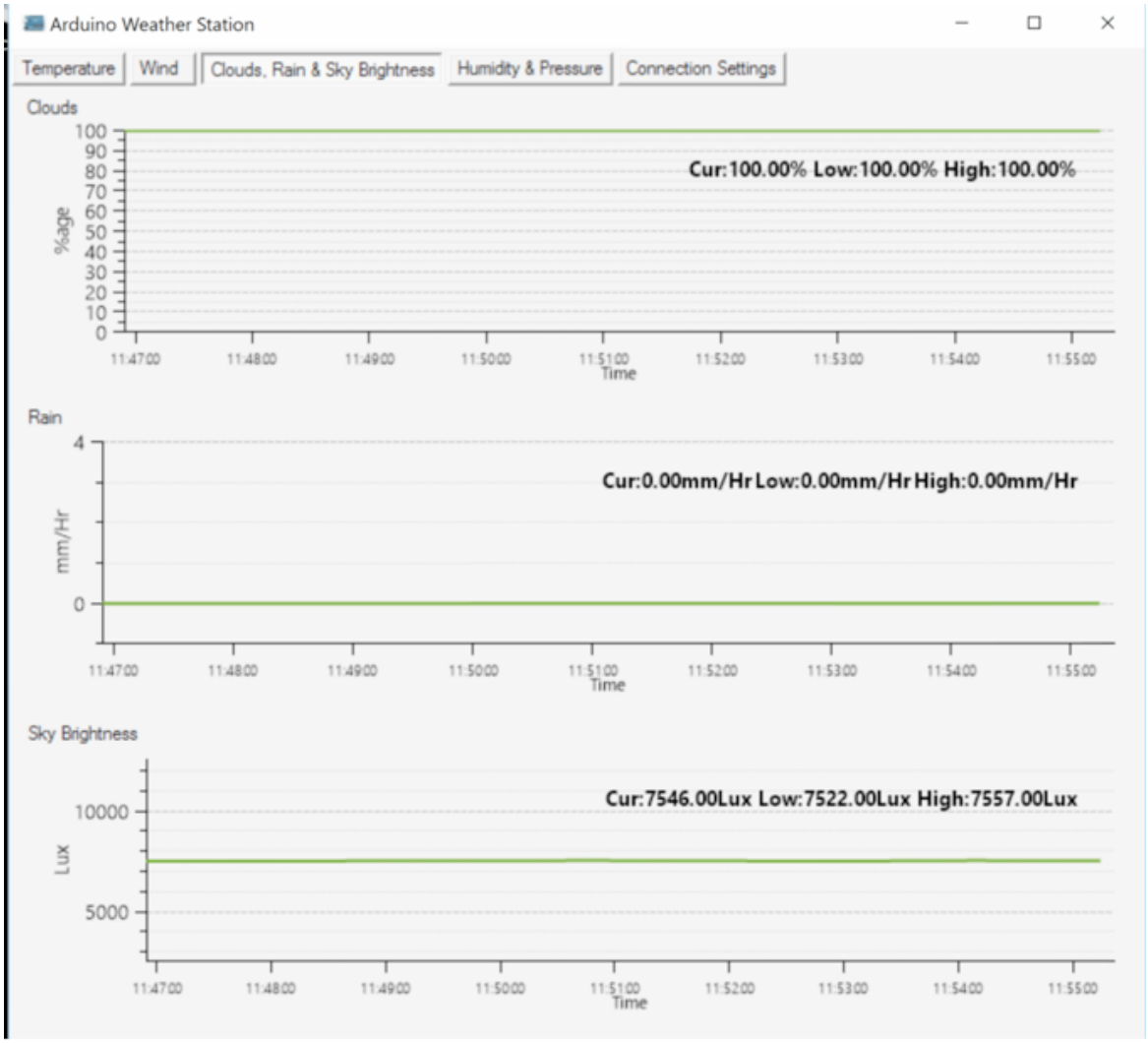
The driver also generates a one-line clarity/boltwood file, which can then be used in conjunction with Chris Rowland’s ASCOM safety driver (http://www.ascom-standards.org/Downloads/SafetyMonitorDrivers.htm) to determine whether it’s “OK to image” or “OK to open” observatory. Remember that the “Cloudy”, “Very Cloudy”, “Light”, “Very Light”, “Windy” and “Very Windy” settings determines what is written to the boltwood file which then subsequently determines when the safety driver will report as safe or unsafe. The location which is set in the Observing Conditions driver is the same one you would pick in the Safety Monitor Driver. Make sure that the location you pick is something that has write privileges for the driver.

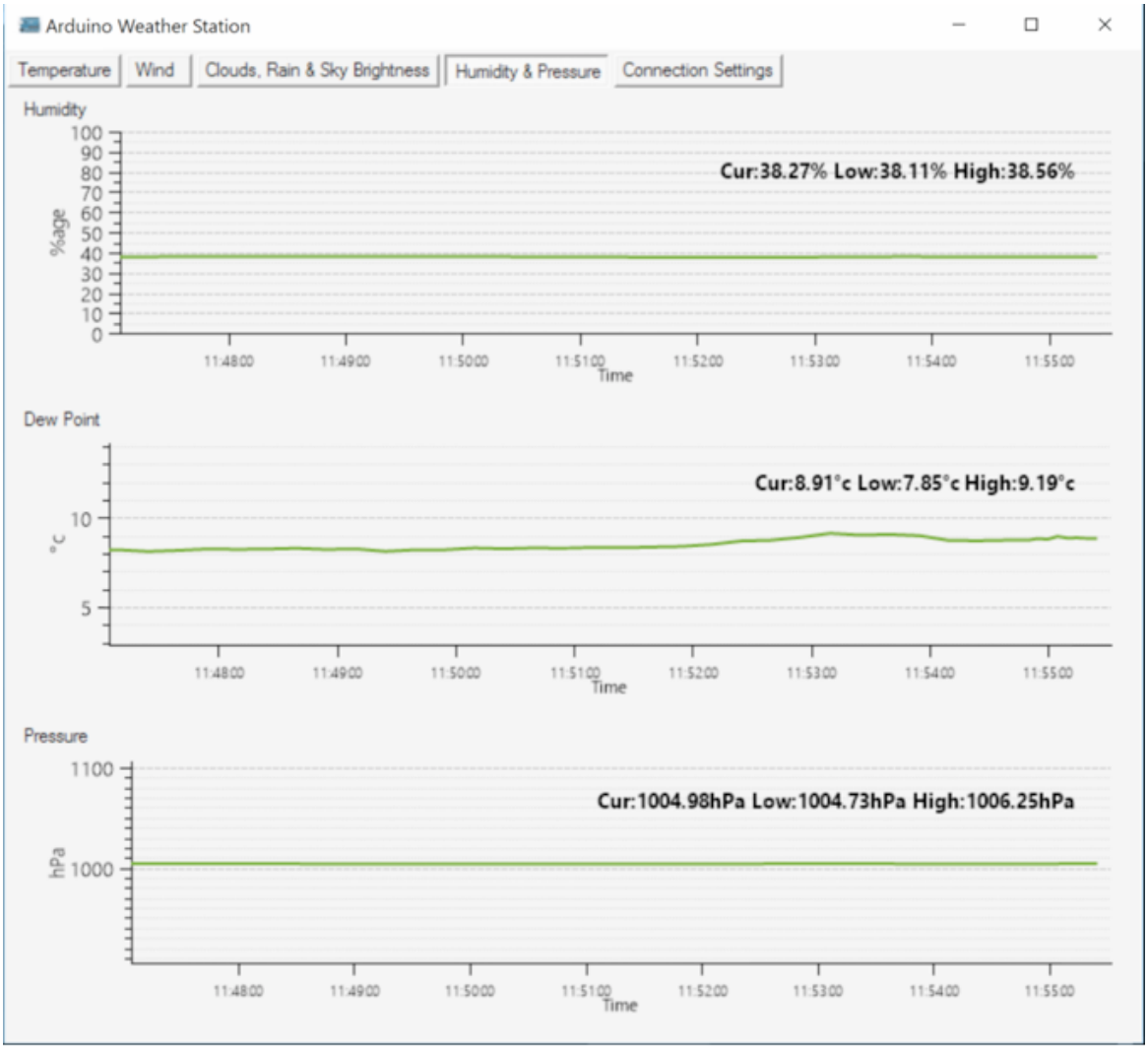
To help with some of these calibrations, or for other use, there is also an executable called Arduino_Weatherstation.exe. It has various different tabs to display running graphs of weather parameters (100 points, rolling). And a settings tab which lets you choose the observing conditions driver, the graph update frequency and the units used to display (These can only be changed when not connected to the driver). The axis scale is automatically chosen based on the data being displayed.

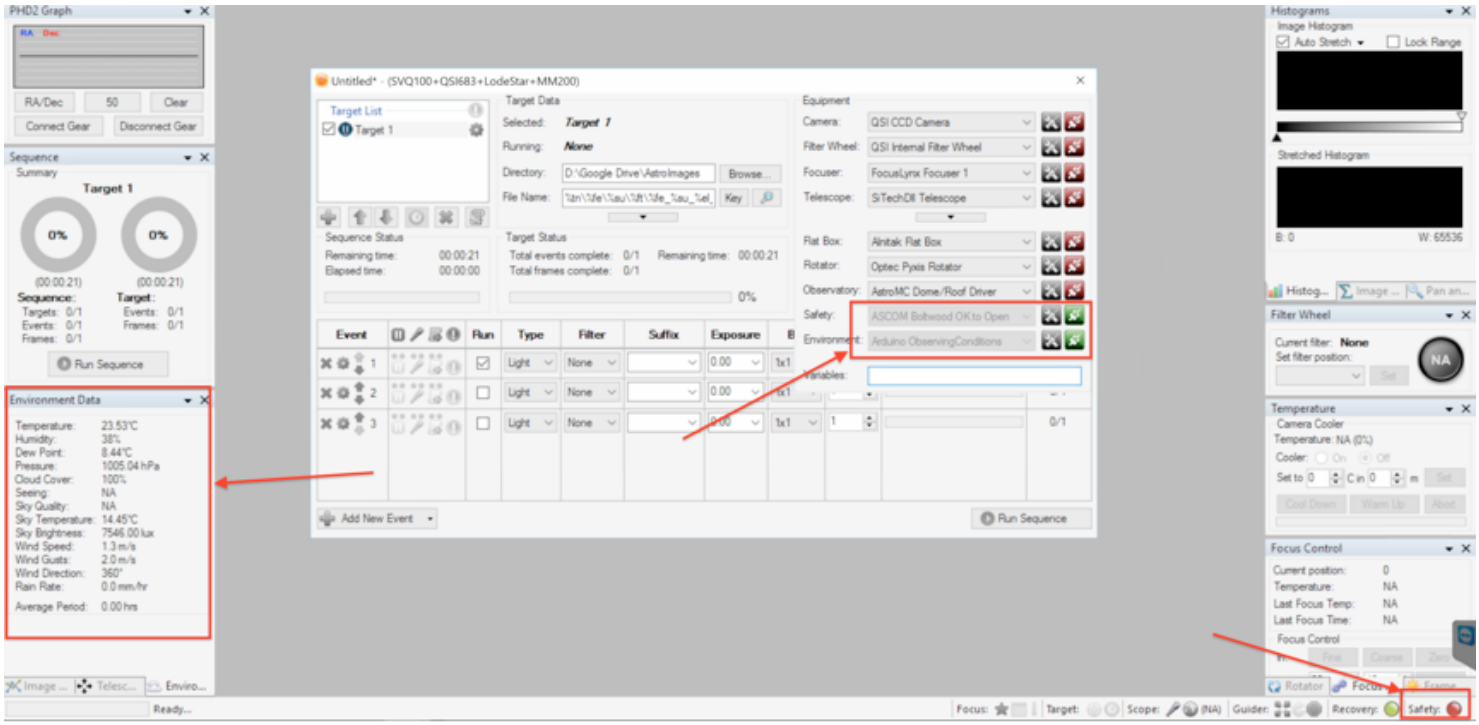

Arduino Weather Station
Temperature
Wind
Clouds, Rain & Sky Brightness
Humidity & Pressure
Connection Settings
Temperature Units
Centigrade
Fahrenheit
Pressure Units
Pascals
Hecto Pascals
Kilo Pascals
Rain Units
mm/Hr
in/Hr
Wind Units
m/s
kph
mph
Update Interval
15
ASCOM.Arduino.ObservingConditions
Choose
Disconnect


Arduino Weather Station
Temperature
Wind
Clouds, Rain & Sky Brightness
Humidity & Pressure
Connection Settings
Ambient
Cur:23.87°c Low:22.77°c High:24.29°c
°c
25
20
11:47:00
11:48:00
11:49:00
11:50:00
11:51:00
11:52:00
11:53:00
11:54:00
Time
Sky
Cur:14.91°c Low:14.51°c High:15.65°c
°c
20
15
10
11:47:00
11:48:00
11:49:00
11:50:00
11:51:00
11:52:00
11:53:00
11:54:00
Time
Diff
Cur:8.60°c Low:7.88°c High:9.36°c
°c
10
5
11:47:00
11:48:00
11:49:00
11:50:00
11:51:00
11:52:00
11:53:00
11:54:00
Time


Arduino Weather Station
Temperature
Wind
Clouds, Rain & Sky Brightness
Humidity & Pressure
Connection Settings
Wind Speed
Cur:0.67m/s Low:0.00m/s High:3.33m/s
m/s
5
0
11:47:00
11:48:00
11:49:00
11:50:00
11:51:00
11:52:00
11:53:00
11:54:00
11:55:00
Time
Wind Direction
Cur:360.00°
Degrees
360
270
180
90
0
Time
Wind Gust
Cur:1.56m/s Low:0.44m/s High:3.33m/s
m/s
5
0
Time


Arduino Weather Station
Temperature
Wind
Clouds, Rain & Sky Brightness
Humidity & Pressure
Connection Settings
Clouds
%age
Cur:100.00% Low:100.00% High:100.00%
Time
Rain
mm/Hr
Cur:0.00mm/Hr Low:0.00mm/Hr High:0.00mm/Hr
Time
Sky Brightness
Lux
Cur:7546.00Lux Low:7522.00Lux High:7557.00Lux
Time


Arduino Weather Station
Temperature
Wind
Clouds, Rain & Sky Brightness
Humidity & Pressure
Connection Settings
Humidity
%age
Cur:38.27% Low:38.11% High:38.56%
Time
Dew Point
°c
Cur:8.91°c Low:7.85°c High:9.19°c
Time
Pressure
hPa
Cur:1004.98hPa Low:1004.73hPa High:1006.25hPa
Time



ENJOY!