Arduino Nano Weather Observing Conditions

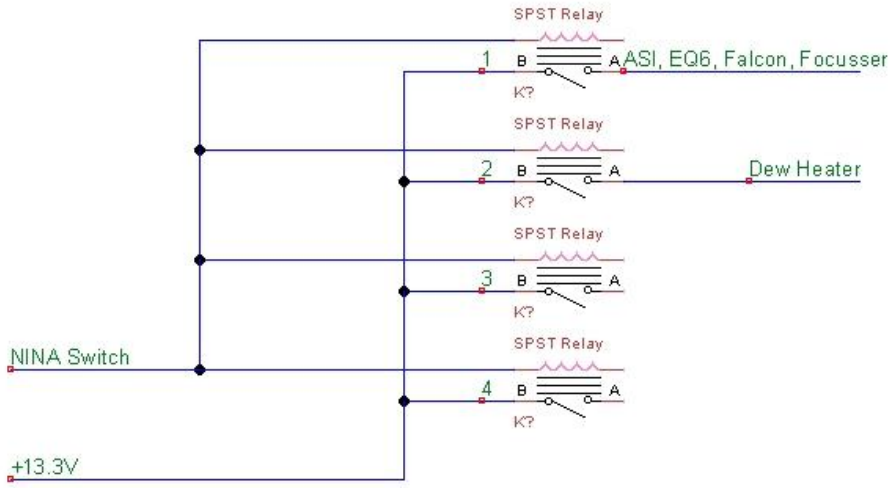
Pre-requisites for software (Arduino/Ascom):

- Siril 1.2.1 or newer
- Arduino IDE 1.8.19
- NINA 3
- NINA-Plugin : Sequencer Powerup
- Polemaster
- Ascom Platform 6.5 or newer

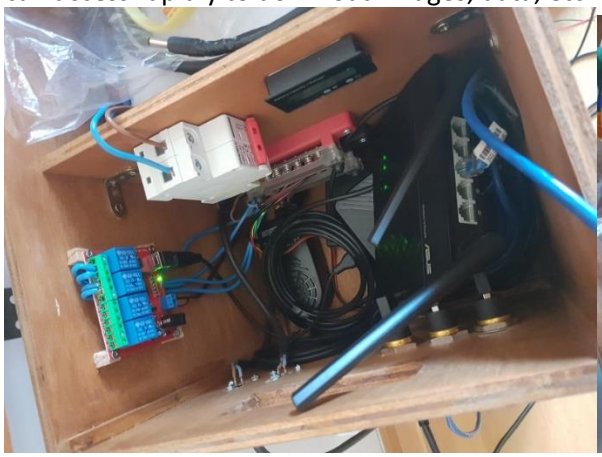
Depending if you are using an Arduino Nano (clone) you might have to download a specific driver for the USB-serial chip on this board (CH341SER.zip
You can download for example: https://github.com/electronicsf/driver-ch341)

After building a new powerbox for my mobile telescope setup, this time I opted for a LifePo4 battery of 12V 100Ah.
When properly treated this battery should almost last a lifetime.
Output voltage is approx. 13.3V constant even after a long and deep discharge over a complete imaging session with dew heaters etc.
If doing an imaging session it usually is from 8 o'clock in the afternoon to approx. 7 o'clock the next morning, sleeping in my car.
A lot of time it was a negative surprise the imaging session was interrupted halfway during the night because of dew on the guidescope lens.
Since the former AGM battery couldn't supply current for everything during the whole night, I started thinking to save battery power.
For that I needed to be able to measure the dewpoint in order to switch the dew heaters on instead of switching them on before I go to sleep In my car.

So, I built the new power box and included a minipc and wifi-router and a USB-switch in order to control power from out of NINA.
Fortunately a USB-switch is easy to find on Aliexpress and the Ascom driver for it can be downloaded for free.



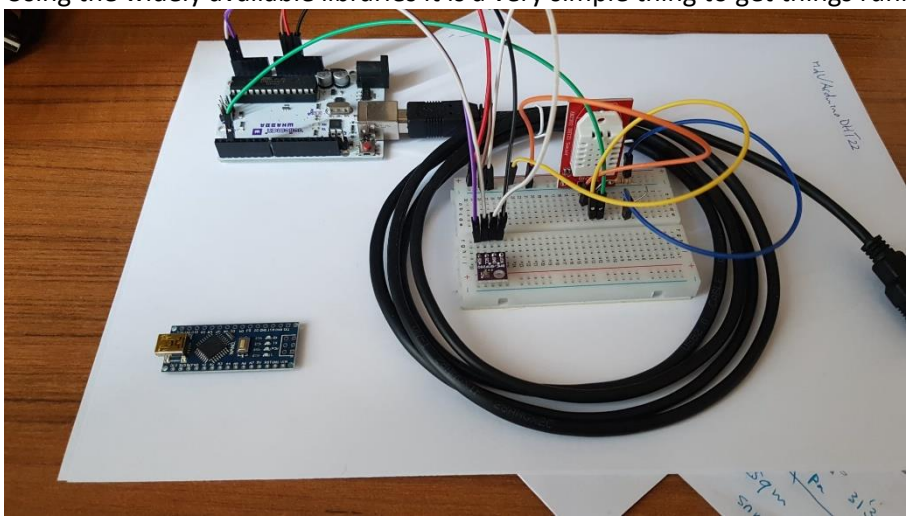
After wiring the new powerbox, with a differential switch limiting the current draw to 10Amp I needed to get to the parameters of ambient temperature and relative humidity in order to calculate the dewpoint.
So, finally I have 2 power-outlets each independently switchable. The first outlet is NC so I don't have to apply power to switch the relay to have power for the mount, scope, cameras etc.
The second outlet is for the dew heaters.
When at home I connect the Wifi-router directly to the grid and can access rapidly to download images, data, etc.



So once finished (can be much better finished but will do the way it is) I started with the Arduino part of the story.

First step was to buy the components, Arduino One, Arduino NANO, breadboard and

Jumper cables.
Then I had to investigate what sensors I'd like to connect.
First trial was done with a simple DHT22 temperature and humidity sensor just to get some feeling on the programming side of things. (just Arduino code).
Using the widely available libraries it is a very simple thing to get things running.



On the lower-left you can see the Arduino NANO, which itself is very small, so ideal to integrate into some small device or project box.
On the breadboard is a I2C BMP280 pressure sensor (very small) and also the DHT22 sensor.

Finally I settled down for the BMP280, the HTU21D (both I2C comm's) and a TL237 Light to frequency sensor.
Since the last sensor is also used in the well known Unihedron SQM meters I opted for the same sensor.

Libraries to program the different sensors are here below listed, each with their respective copyright for free distribution.

For the **BMP280** sensor:
Written by Kevin (KTOWN) Townsend for Adafruit Industries.
https://www.adafruit.com/product/2651
For the **HTU21D** sensor:
Written by Limor Fried/Ladyada for Adafruit Industries.
https://www.adafruit.com/products/1899
For the **TSL237** sensor I used a library originally written for the TSL235R sensor but work the same.
Written by Rob Tillaart, https://github.com/RobTillaart/TSL235R

Finally the whole setup has to connect via an ASCOM driver to NINA which is the imaging program I use.
Using the advanced sequencer and some plugins I will describe lateron you can automatically actuate the switches according to the values obtained by this little device.
Since I use a mobile setup and don't have plans (at my age) to build an observatory I left out a couple of values that are implemented in the

Weather observing conditions driver class.
Specifically I don't measure the Cloudcover, SkyBrightness, Wind speed, Wind-gusts and wind-direction nor the rain.

My interest is to measure SQM, seeing (FWHM), Temperature, Humidity, Dewpoint, and Atmosferic pressure.

In order for the BMP280 sensor to work on the I2C "bus" I had to edit the Library so that it's **base addres 0X76** was used instead of the standard address of 0X77 and then it all worked.
To find out what devices are on the I2C bus there is a little sketch (**I2Cscan**) that will print out the different base addresses that respond.

Finally the sketch on the Arduino has to respond to requests coming from the Ascom driver connected to NINA.

For the SQM part of the device I bought a led-lens of 15º angle to limit the Skyangle as in the SQM-L meter.
Also I glued an UVIR filter in front of the lens to get rid of unwanted UV or IR light.

For the FWHM part of the story I use my Polemaster at the moment and use a conversion factor adjust for the angular resolution of the scope and camera used.
In case of different cameras and scopes with focal length X just calculate the conversion factor.
Example: seeing of 2.1" according to meteoblue
Polemaster uses 3,8um pixels and has focallength of 25mm
According to Siril the FWHM was 2.5 pixels.

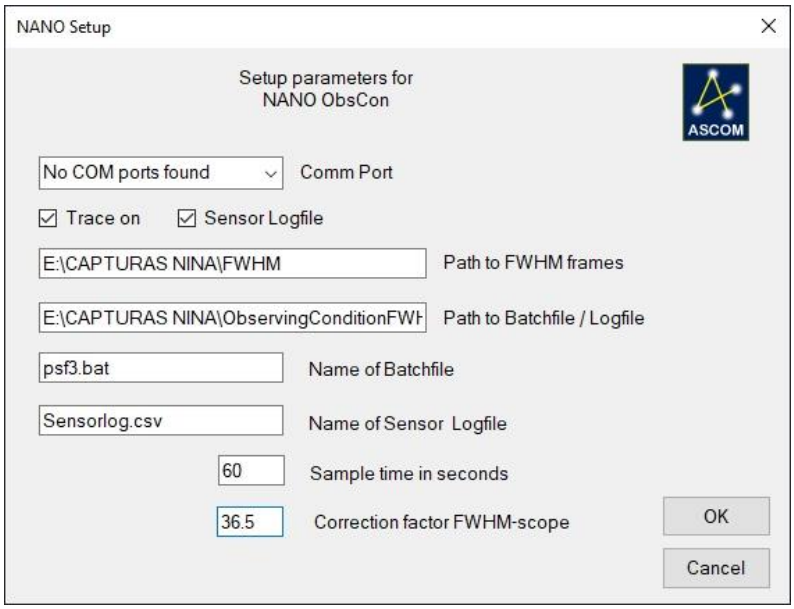
FWHM =(FWHM(pixel) * pixelsize * 206.3)/ Focallength

FWHM teorical	2.1	2.1	2.1
Focal length mm	25	400	280
FWHM in pixels	2.5	2.5	2.5
Pixelsize um	3.75	3.75	3.75
FWHM in arcsec	77.3625	4.83515625	6.90736607
Correction	**36.83928571**	**2.30245536**	**3.28922194**

Now using a second instance of NINA to connect to the Polemaster and setting up a sequence that takes images for example every 60 seconds 1 frame of 0,3 second exposure.

I use SIRIL to calculate the FWHM obtained in the frame of the Polemaster using a little batch command file in my case called Psf3.bat. (you can modify/reconfigure that)

On the setup screen of the Ascom driver you can fill in the correction factor for the scope/camera combination you want to use.
In my case the Polemaster.



********** psf3.bat for SIRIL **********

del "E:\CAPTURAS NINA\FWHM\FWHM.old"
echo OFF
FOR /F "tokens=2 " %%g IN ('siril --version') do (SET version=%%g)
set ext=fits
set Folder="**E:\CAPTURAS NINA\FWHM**"
(
echo requires %version%
echo setext %ext%
echo cd %Folder%
echo load FWHM
echo findstar
echo close
) | "C:\Program Files\SiriL\bin\siril-cli.exe" -s - 2>&1 | findstr (FWHM >%Folder%\fwhm.txt

********** end of psf3.bat **********

In my case I have setup a folder where the captured frames of the Polemaster go called **E:\CAPTURAS NINA\FWHM.**

List of components:

Lenses for SQM meter:
https://es.aliexpress.com/item/32960072994.html?spm=a2g0o.order_list.order_list_main.15.21ef194dUWqrqP&gatewayAdapt=glo2esp

TSL237:
https://es.aliexpress.com/item/4000908691026.html?spm=a2g0o.order_list.order_list_main.20.21ef194dUWqrqP&gatewayAdapt=glo2esp

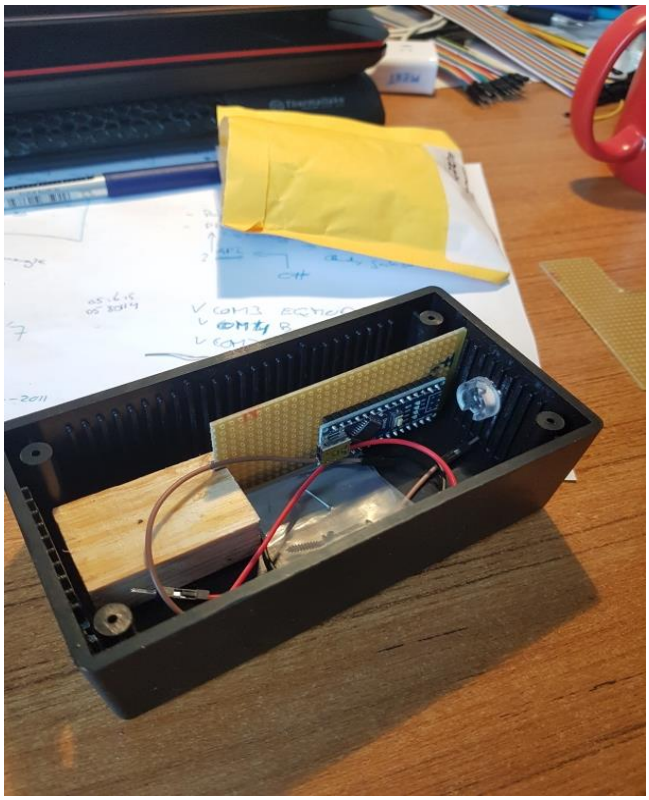
BMP280 sensor:
https://es.aliexpress.com/item/32817286611.html?spm=a2g0o.order_list.order_list_main.25.21ef194dUWqrqP&gatewayAdapt=glo2esp

HTU21D sensor:
https://es.aliexpress.com/item/32656285360.html?spm=a2g0o.order_list.order_list_main.10.21ef194dUWqrqP&gatewayAdapt=glo2esp

USB-switch x4:
https://es.aliexpress.com/item/1005006027324932.html?spm=a2g0o.order_list.order_list_main.40.21ef194dUWqrqP&gatewayAdapt=glo2esp

Ascom driver for USB-switch written by Carl Eric Svensson download link:
https://bitbucket.org/cesvensson/ascom.noyito.switch/src/main/

You also need some plastic project box to put everything into.
Like this for example:

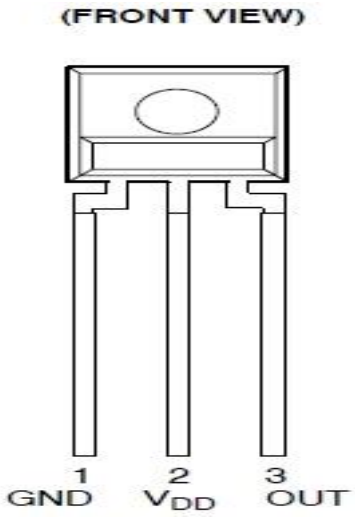


Wiring of the sensors to the Arduino:

BMP280 and HTU21D with I2C

Sensor<------>Arduino

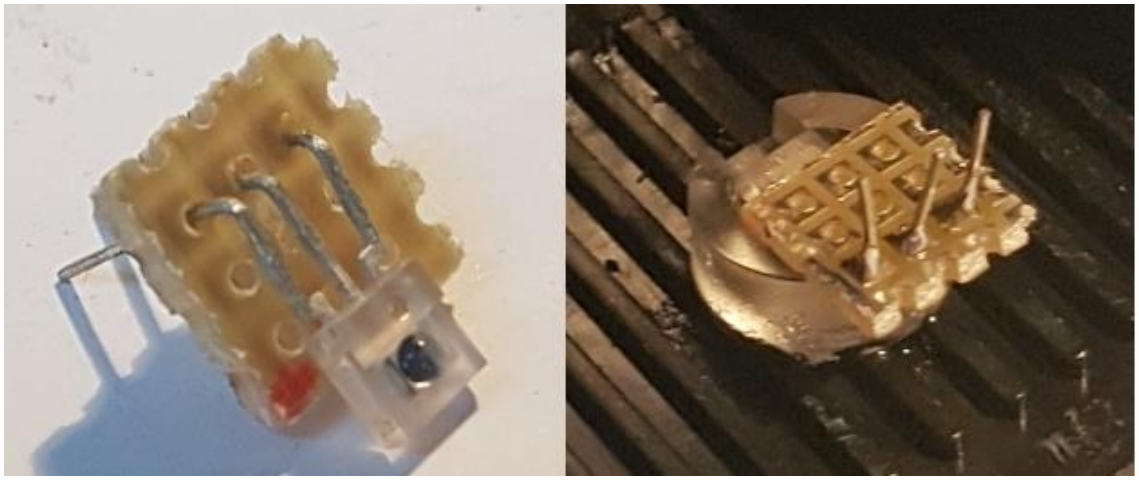
SDA ---------- A4 (data)
SCL ---------- A5 (clock)
VCC ---------- 3.3V
GND ---------- GND



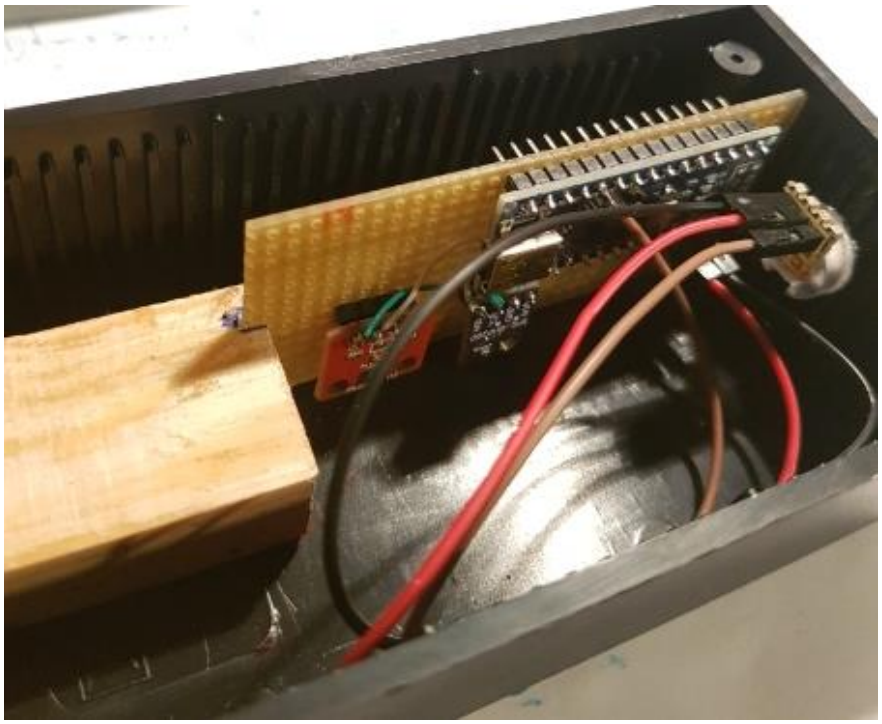
TSL237 ------ Arduino

1 ---------- GND
2 ---------- 5V
3 ---------- D8

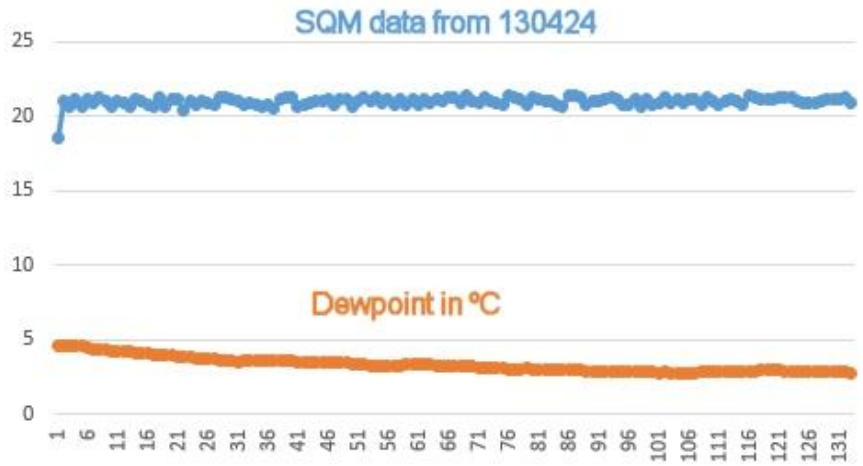
Some idea to handle the very small TSL237 sensor would be to solder it onto some small pertinax board and glue it on the backside of the led-lens.



Finally the sensors soldered/wired onto the pertinax board with the Arduino Nano.

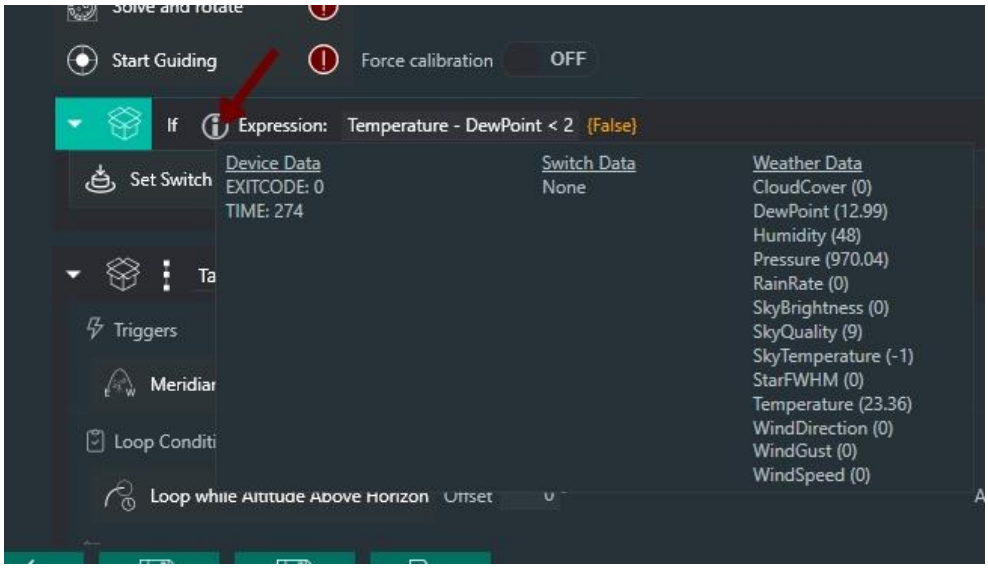


A first run with the meter on 13 april 2024:



The sky was clear and very steady, with the moon at 25% low in the sky causing the SQM value to be lower then normal.

So once the Ascom driver connects to the Arduino we have the values available for use in the advanced sequencer, using the plugin **Sequencer Powerups**.
This gives you the option to have a switch (relay) actuate according to the values programmed in the sequencer.
To know how the parameters are called or what values they have in each moment just hover the mouse over the information icon (see image)



Installation steps:

1) Connect sensors to the Arduino board of your choice (Arduino Uno or Nano)
2) Install the Ascom driver
3) Start the Arduino IDE and load the code for the board (.ino file included)
4) Install the Sequencer Powerups plugin in NINA
5) On the setup screen of the Ascom driver fill in the path's to the directory where you will direct the captured frames of the Polemaster to. (or your Scope of choice of course)
6) If you are using the same filename for the batchfile just leave it as is.
7) Adjust the value of SQM_LIMIT to fine adjust the value of the SQM measured and "reflash" the Arduino board.
   (I glued a UVIR filter in front of the sensor and had to adjust this value a bit)

```
// #include <HTU21D.h>   // responds to address 0X40
#include <Adafruit_HTU21DF.h>
#include <FreqMeasure.h>  //  * Copyright (c) 2011 PJRC.COM,
// #include "Config.h"       //  * Copyrigh, (c) 2011 PJRC.CO
#include "TSL235R.h"      //  * Copyright Rob Tillaart, https

TSL235R  mySensor;
#define SQM_SAMPLES 5
#define SQM_AVG_DELAY 5
#define SQM_LIMIT 13.55   // Calibration factor ( 14.55 with
#define Pedestal 0


// Address of BMP280 is 0X76 checked with I2C-check sketch

Adafruit_BMP280 bmp; // I2C Interface
Adafruit_HTU21DF htu = Adafruit_HTU21DF(); // I2C interface
```