# mySQMPRO

## A DIY Project
## SQM, NELM, IR Sensor (cloud), Rain Sensor, Lux, Barometric Sensor, GPS
## Version 018, 23 August 2019

Author Robert Brown
Date 27 February 2017
Email brown_rb@yahoo.com
© Robert Brown 2014-2019. All rights reserved

## CONTRIBUTIONS

If you wish to make a small contribution in thanks for this project, please use PayPal and send the amount to user **rbb1brown@gmail.com (**Robert Brown). All contributions are gratefully accepted.

# COPYRIGHT RESTRICTIONS

**// This project is protected by International Copyright Law.**
**// Permission is granted for personal and Academic/Educational use only.**
**// Software distributed under MIT License https://opensource.org/licenses/MIT**

**NOTE: If you intend to run this with every option, you should consider using a MEGA 2560 R3 board as the Arduino Nano may run out of memory (if all options are enabled). If you intend to add Bluetooth support, or support for APT and SQM readings (via the Rx command) then a MEGA 2560 is mandatory.**

# Table of Contents

# DISCLAIMER

This project is released into the public domain as is where is, with no obligation or responsibility accepted on the part of the author, for any mishaps or failures caused by this product or use of this product. Users intending to use this project or code do so at their own risk and usage of product and code is deemed to be acceptance of those risks.

In other words, the author accepts no responsibility to damage caused to any equipment or goods or self by using the ideas, schematics and code associated with this project, or loss of income or all other losses that may be incurred.

This project is DIY and makes no pretence to be as accurate as any commercially available unit which can be purchased. Compromises have been made in terms of cost and accuracy. However, the unit should prove to be consistent and reliable and if used within the recommended parameters give acceptable performance.

# RELEASE

The schematic, code and ideas are released into the public domain. Users are free to implement these but may NOT sell projects based on this project for commercial gain without express written permission granted from the author.

# OVERVIEW

This is a DIY Sky Quality Meter based on an Arduino Nano, bread boarded and enclosed in a project hobby box. It is similar to the Unihedron unit but is un-calibrated so may give values that differ slightly when compared with a Unihedron unit. Additional sensors include a Cloud sensor, Rain sensor and Barometric sensor.

# mySQMPRO SPECIFICATIONS

- Displays sky quality reading mag/arc-seconds (and conversion to NELM)
- Can be used manually
- Can be used remotely via RS232 serial command support (via USB, serial speed = 9600)
- Lux reading
- Rain Sensor
- Cloud Sensor (IR sensor MLX90614) indicating Clear, Partly Cloudy, Cloudy
- Barometric Sensor (Humidity, Ambient Temperature, Pressure, Dew Point)
- GPS – Neo-6M
- Choice of LCD displays, LCD1602, LCD1604, LCD2004
- Stripboard or PCB layout (Fritzing PCB can be ordered on-line)
- USB Interface for remote operation
- Windows application for viewing and logging data
- Powered from 12VDC, reverse voltage protection and 2A quick blow fuse

# mySQMPRO PURCHASE LIST

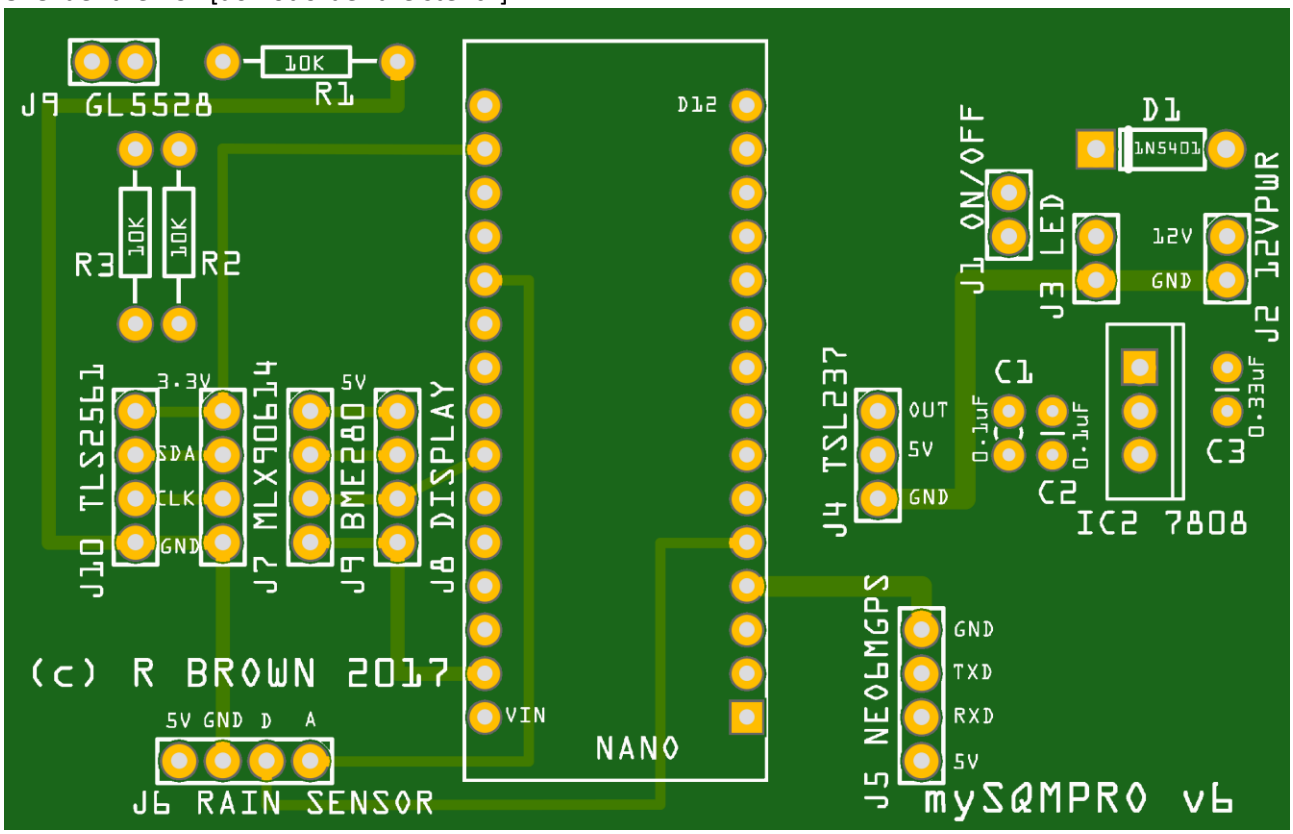
Please see the component list on the main site

# mySQMPRO PCB PURCHASE

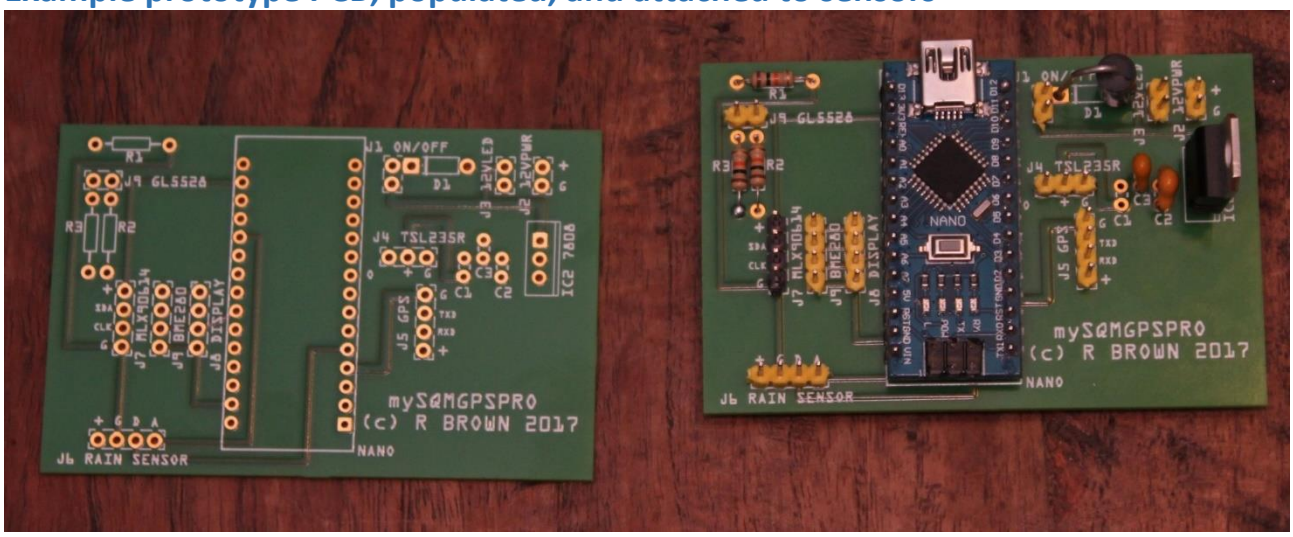
The PCB can then be ordered directly on-line.

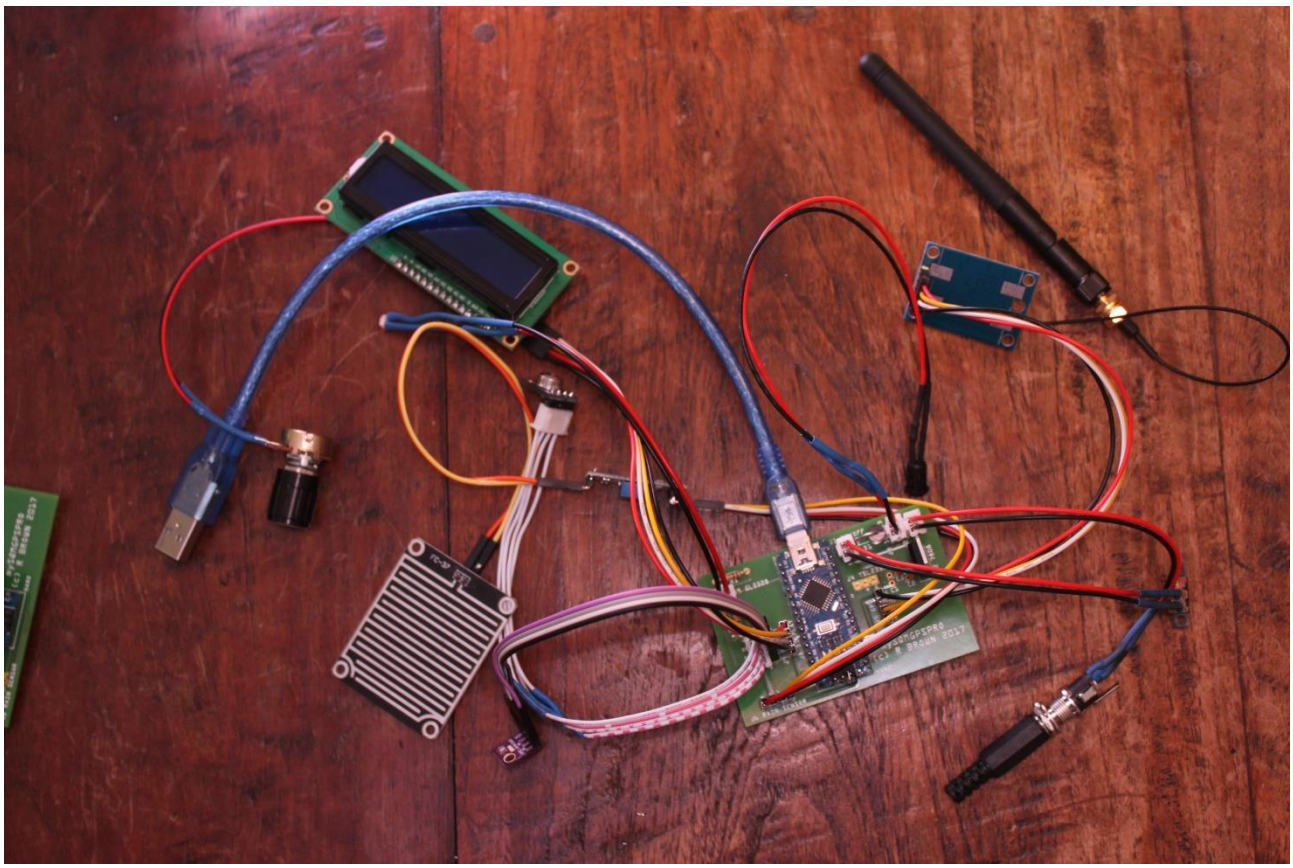
mySQMPRO PCB PURCHASE

1. Please download the zip file mySQMPRO-PCB.zip
2. Goto https://easyeda.com/order
3. If you do not have an account, please register
4. Login
5. Click the button "Add your Gerber file"
6. Navigate to where you saved the mySQMPRO-PCB.zip file and select it
7. Wait for the PCB to be generated
8. Order the PCB [do not order the stencil]




## Example prototype PCB, populated, and attached to sensors





# POWERING THE NANO

The NANO v3 CH340 ATMega328 can derive power from the mini-USB cable, as well as an external source connected to Pin 30, VIN. This pin accepts between 7-12VDC (maximum 20VDC). A LM7808 voltage regulator provides external power so the external power of 12VDC can be used to power to the NANO board.

The input is diode protected against accidental reversal. A quick blow 2A fuse provides additional protection.

The controller requires an external 12VDC supply for normal operation, even when connected to a computer via USB cable. The 12VDC external supply is necessary as the GPS and other devices exceed the current draw limit of 500mA USB cable (if you try to power the controller using only USB then some functions may not work; the controller may not update correctly or not respond to update requests).

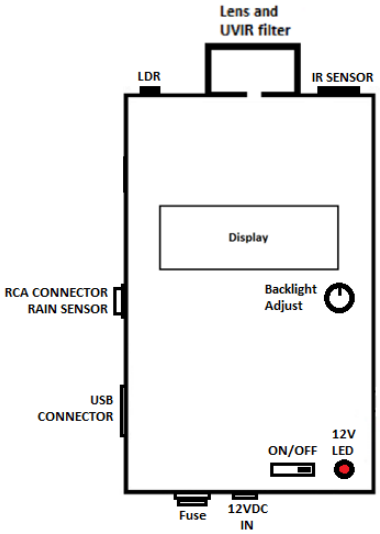
# mySQMPRO OPERATION

The mySQMPRO controller uses a TSL237 light to frequency sensor. The sensor is mounted behind a lens assembly (FOV approximately 20 degrees) and a 9.5mm UV/IR block filter. The filter is important to block out UV/IR light and avoid inaccurate readings of the TSL237 sensor, which has a peak response around 700nm which tapers off down to about 1100nm. The Unihedron uses a Hoya-CM-500 filter which has a response from about 340nm to 660nm. The UV/IR block filter used in mySQMPRO has the following specifications

- Tavg >95% @440-620nm
- T =50% @648+-15nm
- Tavg <= 1 @ 700-1000nm
- T : < 1% @ 1050nm

Note: Another source of UV/IR filters are old web cameras which are often fitted with a UV/IR lens. Different filters will cause differences in SQM readings.

A sample is taken every 5 seconds. The board and display are mounted in an ABS plastic enclosure.



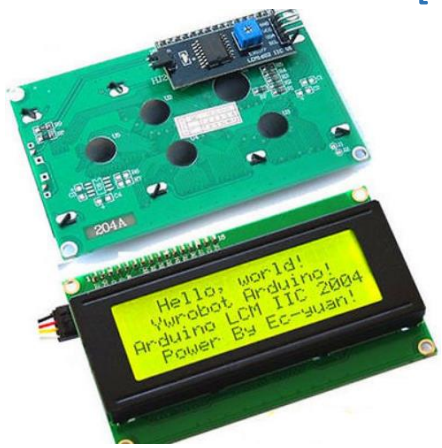
# UPLOADING THE mySQMPRO ARDUINO CODE TO THE NANO

Please use the Arduino Sketch IDE software (1.6.8) with this project. Select the correct board from the Tools > Board menu. Select the correct serial port from the Tools > Serial Port menu. Press the upload button in the Arduino environment. The board will automatically reset and the sketch will be uploaded.

# ON-OFF SWITCH

A mini sliding ON-OFF SPST switch is provided that turns ON/OFF power to the NANO. When the switch is in the ON position, the NANO will repeatedly take sky measurements and display the measurements on the display. The ON-OFF switch is connected to the 12V battery.

# LCD2004 I2C DISPLAY [RECOMMENDED]




The LCD2004 I2C display is the recommended choice as it is easy to interface, operates at 5V and provides 4 lines of information.

http://www.ebay.com/itm/20x4-LCD-2004-Character-Display-IIC-I2C-TWI-SPI-Serial-interface-Board-Module-/201748237624

alice1101983
$4.95USD

Alternatively, a LCD1602, LCD1604 or OLED display fitted with I2C can be used.

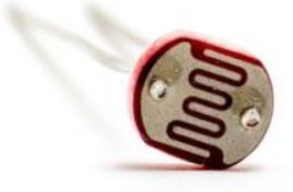
# A NOTE ABOUT THE PROGRAM CODE AND FREQUENCY MEASUREMENT

A number of alternative methods of calculating frequency were tried. Some of these are readily available on the Internet. These methods used a number of different methods to determine the frequency, such as pulseIn() or using Interrupts with the sensor connected to Pin 2 (as on the Arduino playground).

However, I tested each of these methods using a function generator set to 1Mhz square wave output and connected to the pin being measured. In each instance there was too much variation or inaccuracy in reading of the frequency.

The frequency counter library implemented by Martin Nawrath at http://interface.khm.de/index.php/lab/experiments/arduino-frequency-counter-library/ gave accurate reliable results so has been used in this project.

The mySQMPRO uses a Light Dependant Resistor (LDR) connected to Analog Pin 0 of the Nano. The purpose of the LDR is to change the timing related to frequency measurement. When the LDR is in bright light, this reduces the period over which pulses are captured and when the LDR is in dark areas, the period over which pulses are captures is lengthened. This prevents overruns and adjusts the measurements dynamically in response to light conditions.



If an LDR was not used, either the unit might lock up in bright sunlight, or give inaccurate results in dark conditions.

*Note that in dark conditions the sampling time gate period is set to two seconds, so the display page update is set to every 2 seconds (there are multiple pages to display).*

# mySQMPRO READINGS

A timer loop is used to sample various sensors 2.2s apart. Due to the use of a maximum 2s gate time for the frequency routine, this means samples must be greater than 2s apart, which is why the firmware uses a 2.2s delay between sample readings. At each sample a different sensor is sampled, meaning that every sensor is sampled over an 11s period.

To determine the SQM value, the irradiance value is used, which in turn is derived from the frequency measured by the sensor. The irradiance is measured in uW/(cm^2) and the frequency in Hertz.

The first displayed reading is nearly always invalid as it takes a few seconds from power up for components and readings to stabilize.

# FIRMWARE FILES AND BUILD OPTIONS

The main PCB and firmware supports LCD1602, LCD2004 and OLED displays.

## Recommended Build

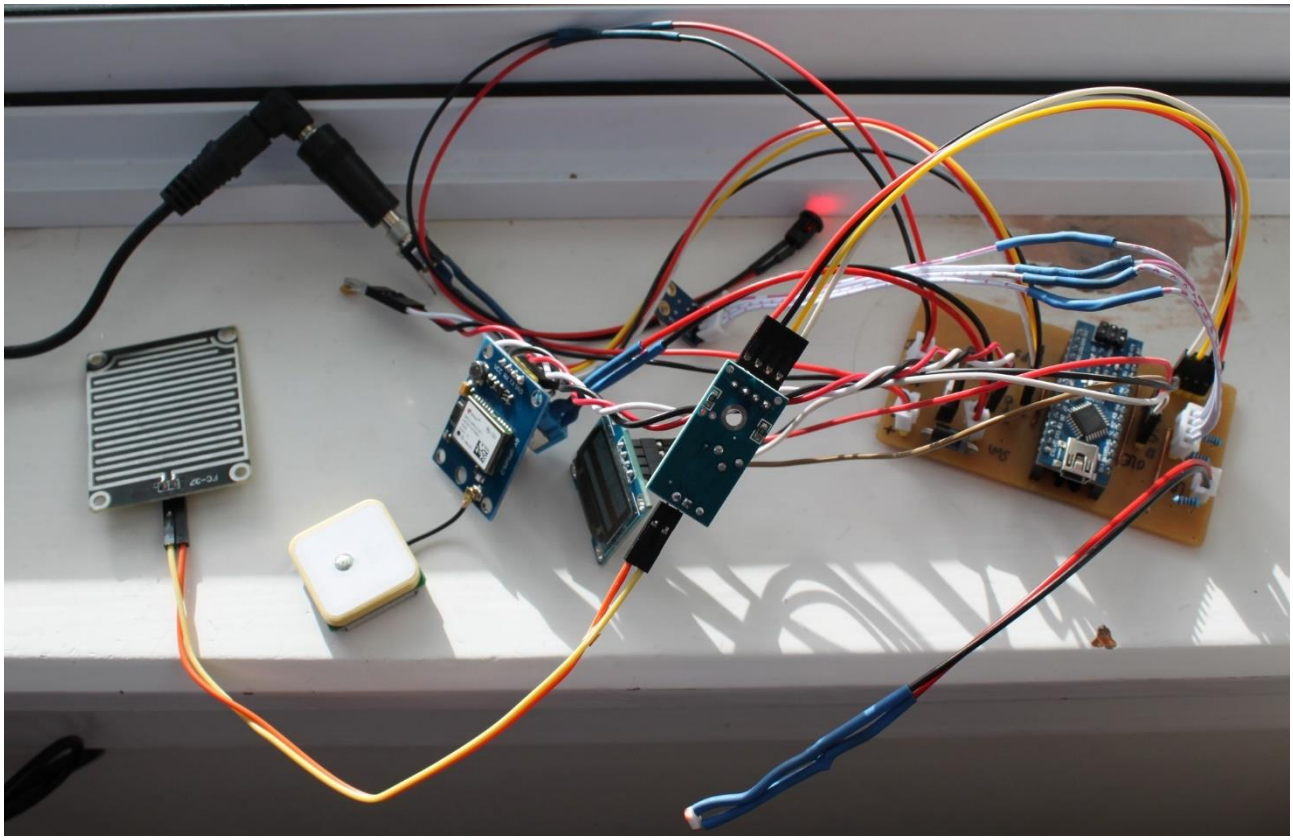
LCD2004

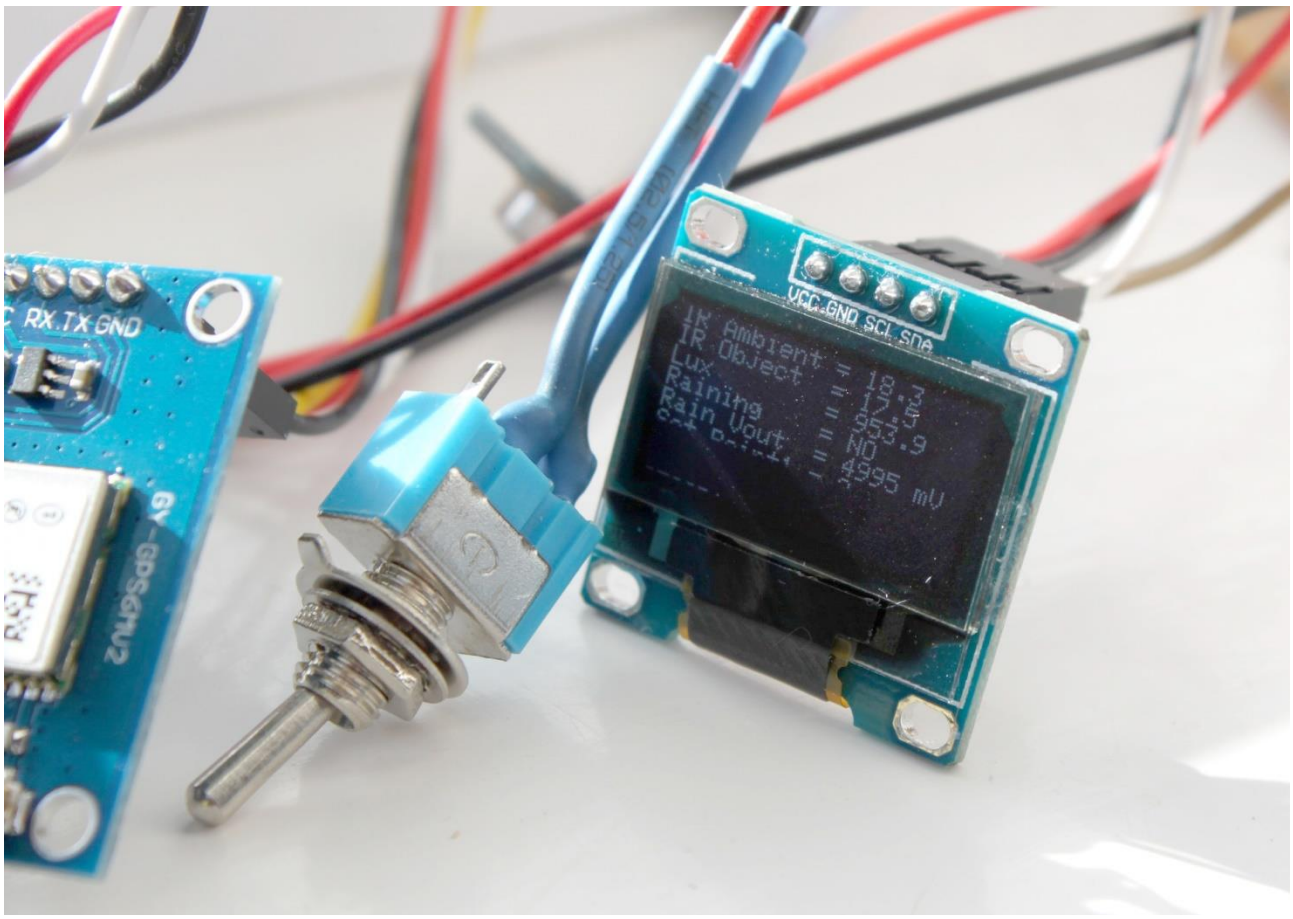

The v5 PCB (double sided, plated through holes, silk screened, professionally made) can be ordered (with or without a part pack – not all parts are in the pack and some may need to be purchased from an alternative supplier such as eBay) at

https://aisler.net/p/JHYFZBXW

# mySQMPRO PROTOTYPE BUILD PICTURES

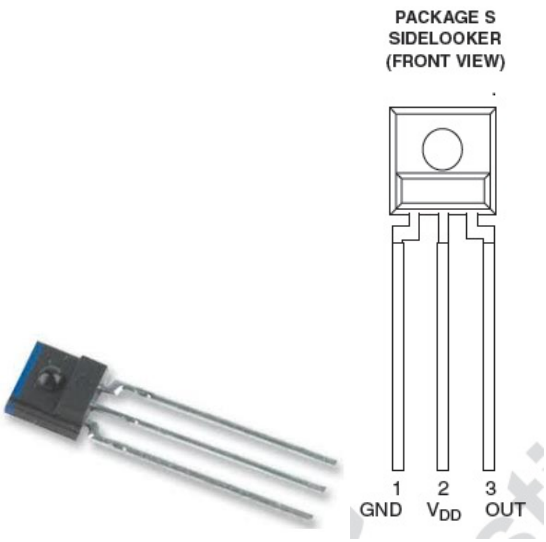
You must mount the 0.1uf capacitor at the connector nearest to the TSL237 sensor (across VCC and GND) else you will got get any real readings from the sensor. In the production version, this capacitor is mounted directly across the sensor leads (not shown in the photo on the left) instead of being placed on the PCB.




RX TX GND
VCC GND SCL SDA
IR Ambient = 18.3
IR Object = 17.6
Lux = 953.9
Raining = NO
Rain Vout = 4995 mV

# mySQMPRO SENSORS

## TSL237




The TSL237 sensor was purchased from Ali-express. The sensor converts the received light to frequency, and then this value is used to determine the Sky Quality Measurement (SQM) value.

Ensure the sensor is connected the correct way and that the 0.1uF capacitor is soldered across the VCC and GND leads of the sensor.

The sensor needs to be mounted behind the lens assembly and protected from any stray light – which would affect the value returned from the sensor and give incorrect values.

## MLS90614 IR SENSOR

The MLX90614 sensor has an I2C interface (5V) and provides values for Infrared Ambient and Object temperatures. The sensor must point to the night sky.

These values can be used together with set points to determine the sky state (cloudy, clear).

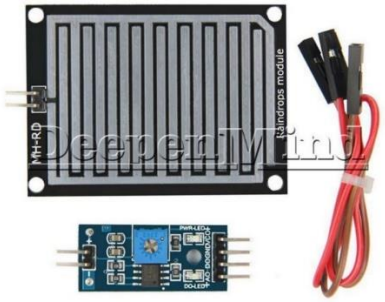
## BME280 SENSOR


The BME280 barometric sensor has an I2C interface (you must purchase the 5V option) and provides values for atmospheric pressure, ambient temperature and humidity.

The controller uses the ambient temperature and humidity values to calculate the current dew point.

The code to handle the BME280 sensor is contained within the controller firmware file.

## RAIN SENSOR




The rain sensor comprises two parts, a logic board and a sensor plate.

The sensor plate must be mounted externally to the controller unit, and is connected to the controller unit via 2 wires. The recommended connector is a RCA type connector.

The logic board for the rain sensor is mounted internally inside the case.

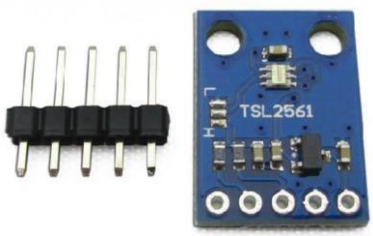
## NEO-6M GPS SENSOR

The NEO-6M GPS sensor interfaces to the controller via data lines which are used to handle TXD and RXD data exchanges.

The sensor requires a lot of power and the controller must be powered from a 12VDC source, even when connected via USB.


## TSL2561 LUX SENSOR

The lux sensor interfaces to the controller via I2C and requires 3.3V. The sensor has a range of 0.1 to 40,000 lux. If an error occurs reading the lux value, the resultant lux value is set to -1.0

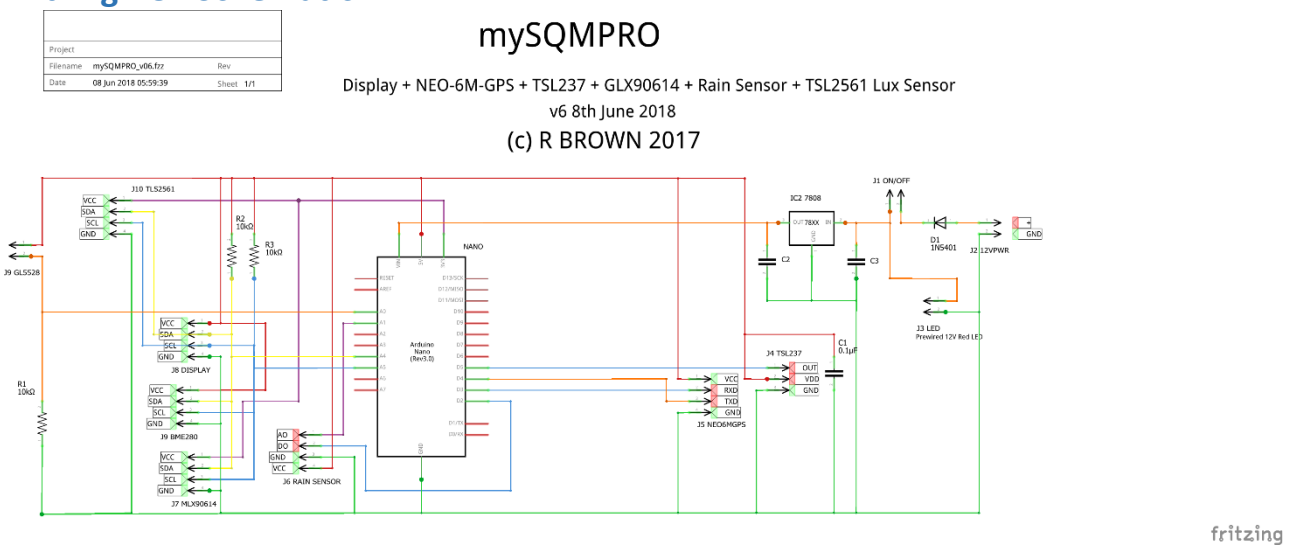


# BUILD OPTION FRITZING PCB - RECOMMENDED

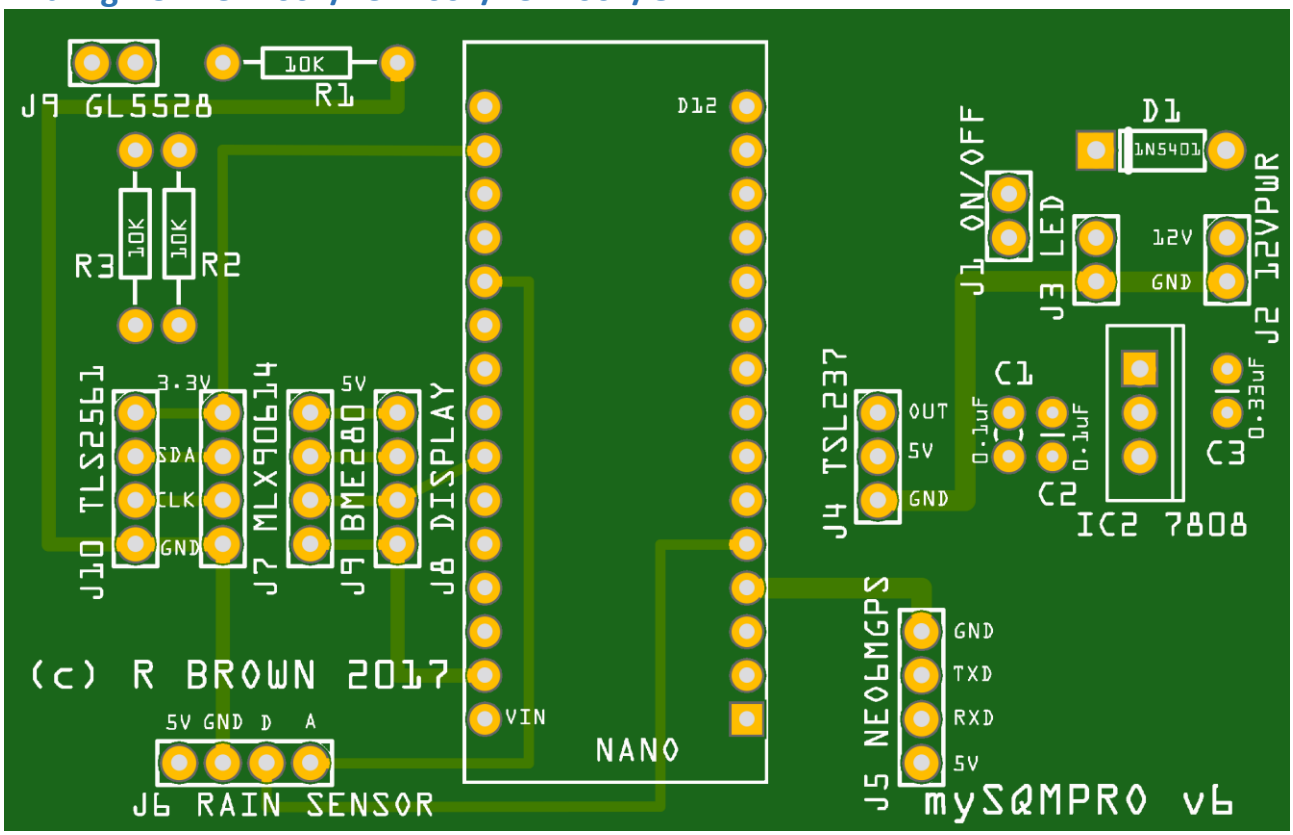
The current PCB (double sided, plated through holes, silk screened, professionally made) can be ordered (with or without a part pack – not all parts are in the pack and some may need to be purchased from an alternative supplier such as eBay) at

https://aisler.net/p/JHYFZBXW

## Fritzing PCB Schematic




## Fritzing PCB LCD1602/LCD1604/LCD2004/OLED




## FRITZING PCB FIRMWARE TO USE

mySQMPRO-0xx.ino

# BUILD OPTION LCD2004/LCD1604/LCD1602

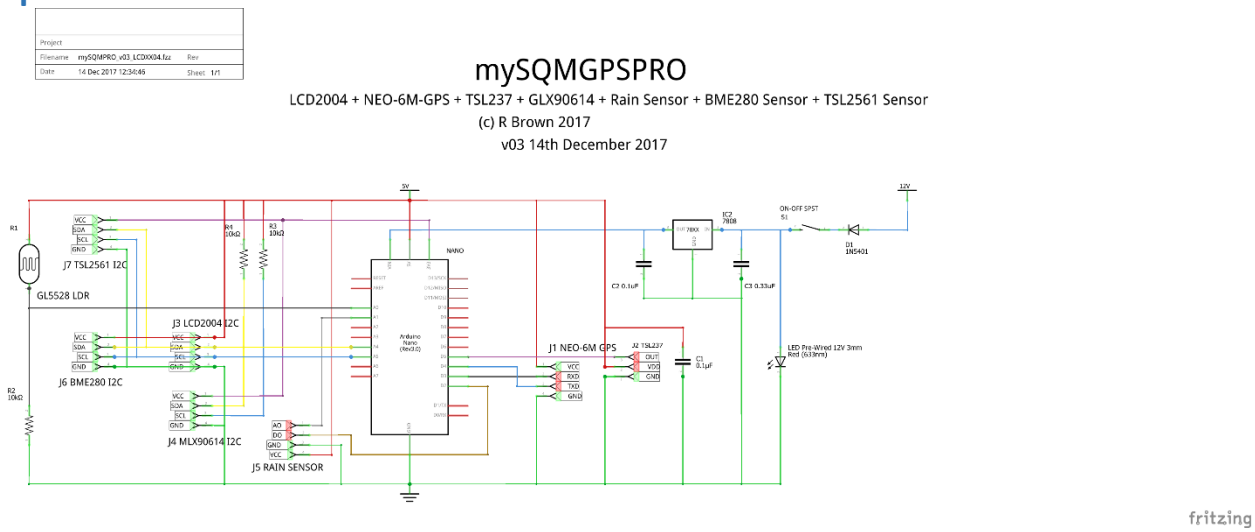
If you do not wish to purchase a PCB from Aisler, you can build a stripboard version.

Stripboards are based on the original prototypes. This means that they have not been updated with new features that have been added to the PCB versions. They may also contain errors.

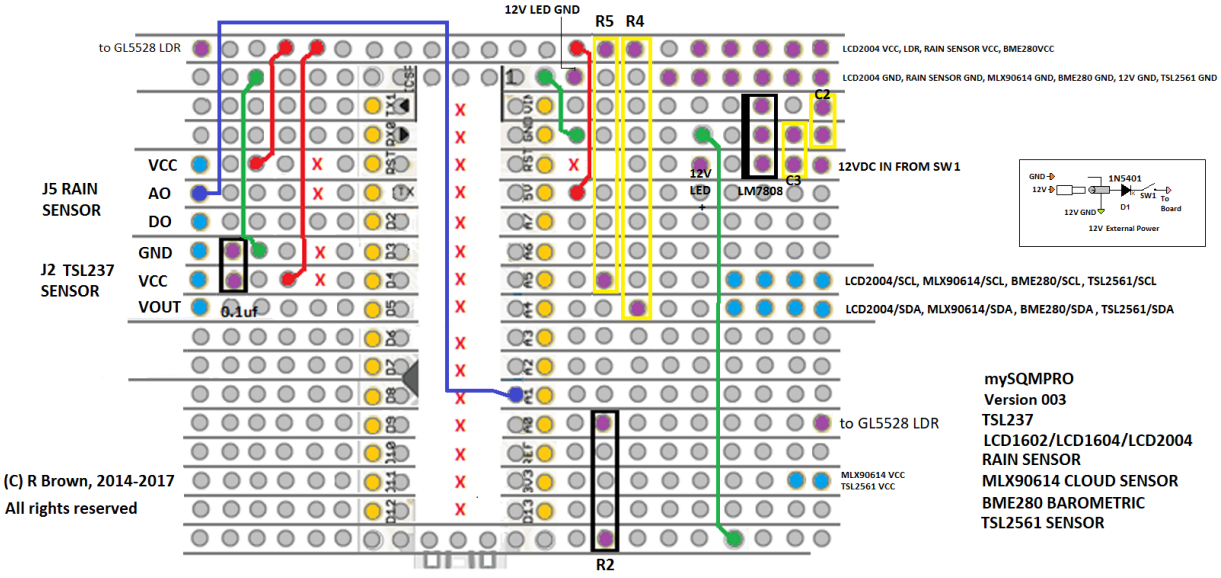
However, stripboards are

- relatively easy to make
- require skills in soldering
- take longer to assemble than PCB versions
- easier to make mistakes

## Stripboard Schematic




## Stripboard Layout




## STRIPBOARD FIRMWARE FILE

mySQMPRO-0xx.ino

# A NOTE ABOUT THE SERIAL COMMAND INTERFACE

The mySQMPRO controller supports a remote serial command protocol via USB, so that an application program can request the current values from the controller. The default serial speed is 9600.

# ARDUINO IDE

The version of the Arduino IDE used in this project to compile the firmware and program the Nano controller is version 1.6.8 and can be downloaded here. Run the installer and remember to install the drivers for the Arduino at installation of the IDE.

# REQUIRED LIBRARY FILES

The libraries are files that are necessary to compile the firmware for the mySQMPRO. The libraries are contained within the Arduino Firmware mysqmpro vxxx.zip file. Unzip this file and you will have folders for the firmware and another folder called **Required Libraries**.

The folder Required Libraries should contain NINE folders. Copy these NINE folders to the Arduino\libraries folder in your Documents folder (for windows), replacing any existing folders or files.

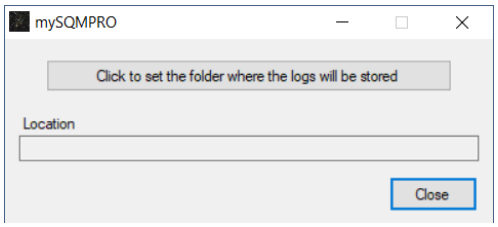
Do not edit or replace any of these files. They have been edited and modified to use with the mySQMPRO firmware.

# UPLOADING THE mySQMPRO FIRMWARE TO THE NANO

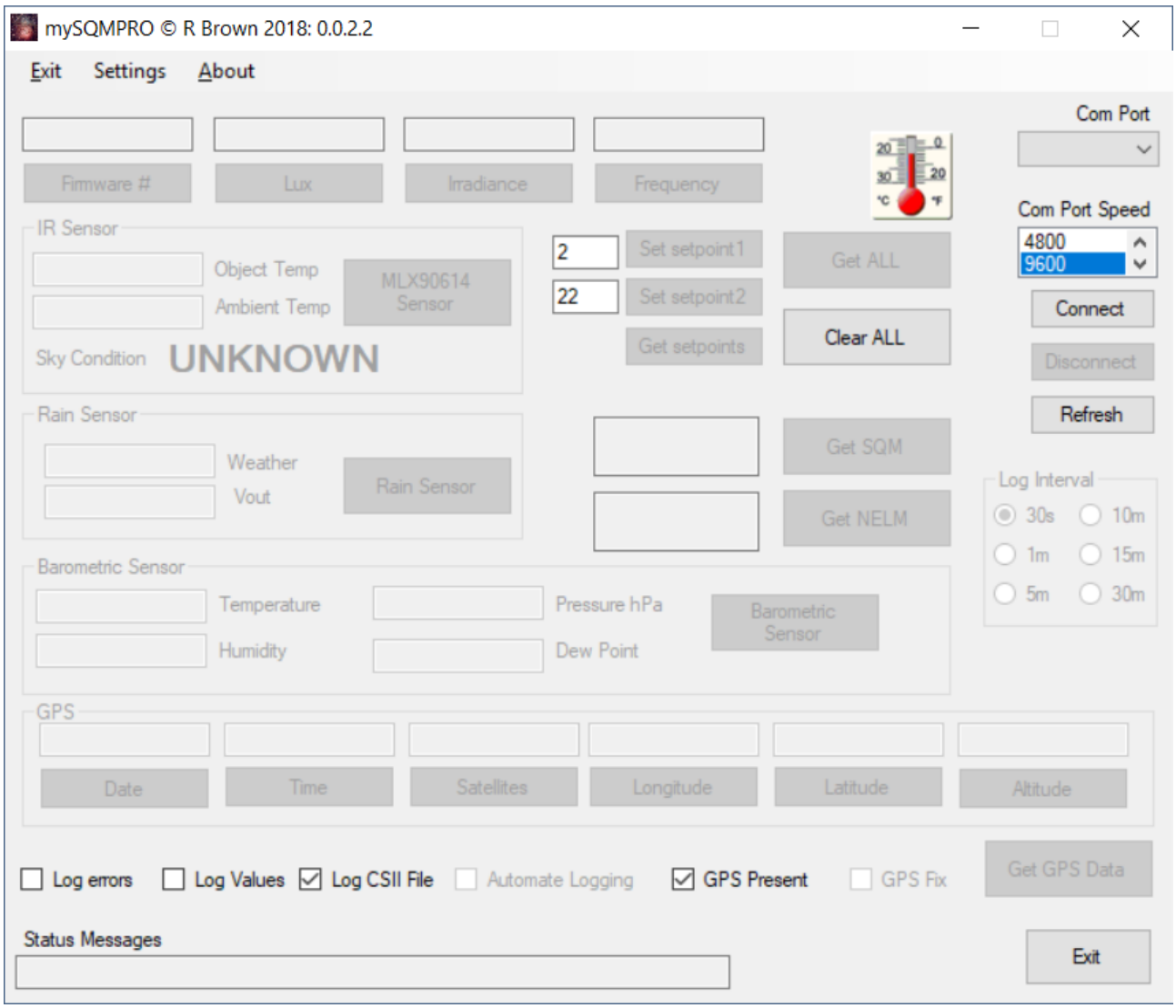
Please use the latest Arduino Sketch IDE software (1.6.8 or higher) with this project.
Select the correct board from the Tools > Board menu. Select the correct serial port from the Tools > Serial Port menu. Press the upload button in the Arduino environment. The board will automatically reset and the sketch will be uploaded.

# mySQMPRO WINDOWS APPLICATION

When the Windows application is run for the first time, it will ask for the location where logfiles can be stored. Clicking on the “Click to set the folder…” button will display a file/path dialog box where the user can select a drive or path where the log files will be stored. The name of the files is automatically created by the application. Once this path is set the user can close the dialog box by clicking on the Close button.

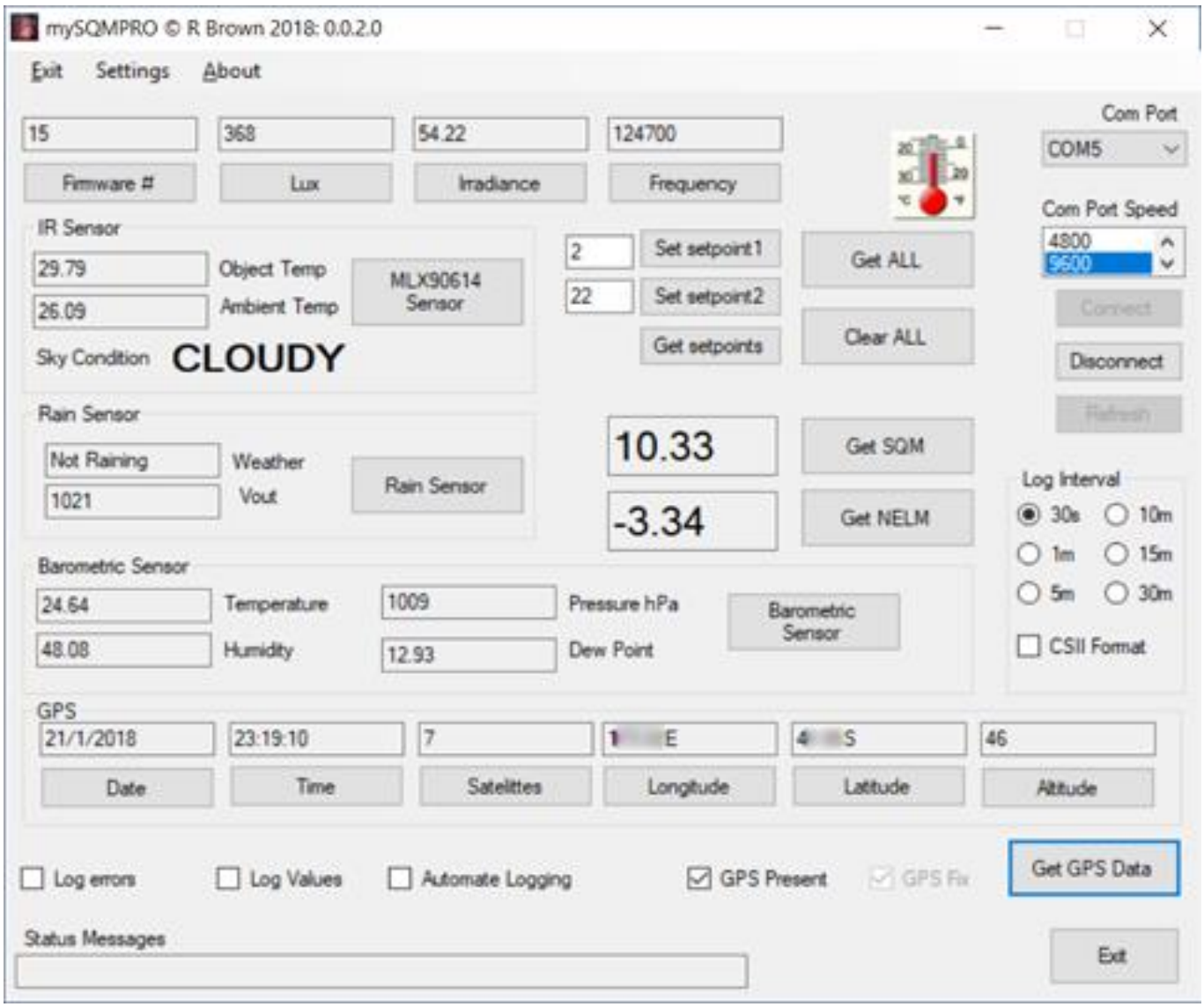



Once the path is specified, the form can be closed and then the main application window is displayed.




To connect to the mySQMPRO controller, first select the correct COM port, set the COM port speed to 9600, then click the Connect button.

When connected, a “Connected” message will appear in the Status Messages text box. This may take up to about 30 seconds or longer depending upon how long it takes the GPS controller to establish a GPS fix. If the connection request times out, dismiss the warning dialog box and attempt to connect again.




The above screenshot shows the Windows application connected to a mySQMPRO controller (daytime, inside house).

**Note: Wait at least 10 seconds after connecting to the controller before trying to retrieve values from the controller.**

## Com Port Speed

The speed is 9600.

## Automate

Once connected, the automate function will automatically retrieve the data from the controller at a user specific rate of 30s, 1m, 5m, 10m, 15m or 30m intervals if enabled. If **Log Values** is checked then the values will be written to a comma delineated text file.

## Log Errors

When enabled, errors are logged to an error file.

## Log Values

When enabled, values are logged to a data file.

## GPS Present
Check this control if your controller has a NEO-6M GPS device attached.

## GPS Fix
This control indicates if the GPS unit has established a fix (check box is checked). This is only updated when the Get GPS Data button is clicked.

## Get GPS Data
This button requests the latest GPS data from the controller. You should only click this button if a GPS is present. If the GPS Present checkbox is unchecked, the request for GPS data will be ignored.

## Rain Sensor
Gets the current rain sensor values from the controller.

## Barometric Sensor
Gets the current barometric values (ambient temperature, humidity, dew point and pressure in hPa) from the controller.

## MLX9014 Sensor
Gets the current Infra-Red object and ambient temperature readings from the controller.

## Get SQM
Get the current SQM value from the controller.

## Get NELM
Get the current NELM value (calculated from SQM value) from the controller.

## Get ALL
Gets all values from the controller.

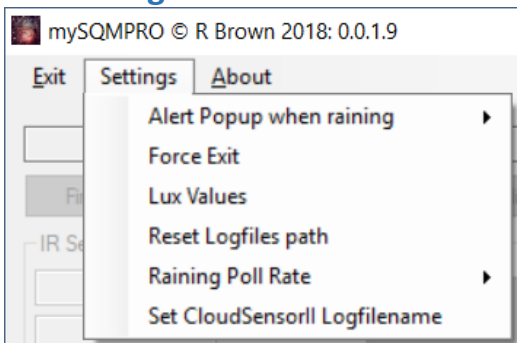
## Clear ALL
Clears all displayed values.

## Set Setpoint1/2
The set point values are used to determine the cut-off points for clear/cloudy/overcast sky conditions. These values are sent to the controller when connecting and are saved by the application. If you want to change these values, you can enter new values either before or after connection to a mySQMPRO controller. More information on setpoint values is found here.

## Get Setpoint1/2
Get the current setpoint values from the controller.

## Lux
Gets the lux value from the controller. This is a value either based on the reading of the LDR or the TLS2561 sensor. The lux reading from the LDR is only an approximation. See appendix C for further information

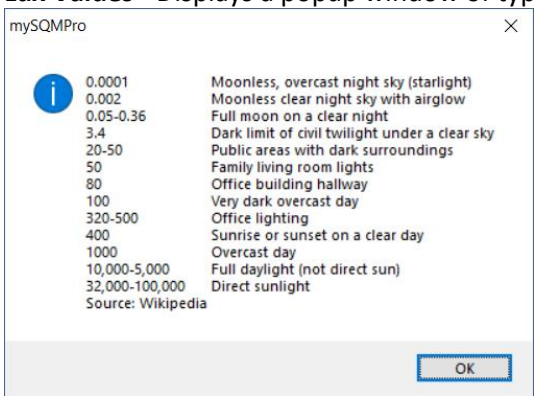
## The Settings Menu




**Alert Popup when raining** - Enables or disables the alert popup box when rain is detected (only works when an update of values occurs or when Automate is checked)

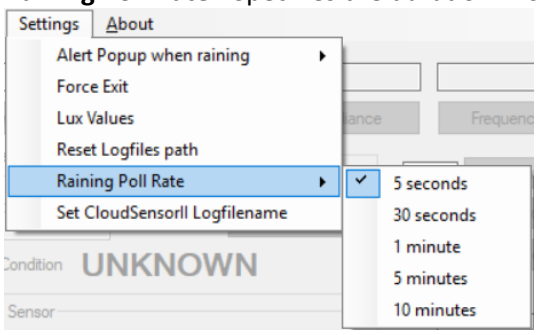
**Force Exit** - Exits the application even if the controller is connected

**Lux Values** - Displays a popup window of typical Lux values for reference purposes



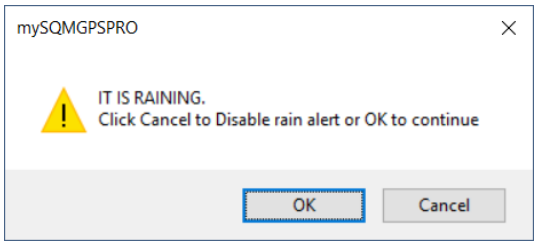

**Reset Logfiles Path** - The path where logfiles are kept can be changed using this option

**Raining Poll Rate** - Specifies the duration in seconds between polling the controller for a rain status update



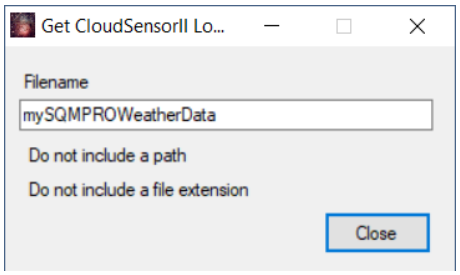

## Alert Popup when raining

This option enables/disables an alert popup message-box if rain is detected by the controller. The message box will only appear if connected to a controller and this option is enabled.



## Set CloudSensorII Logfilename

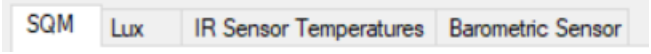
This option is available on the Settings menu. The default filename is mySQMPROWeatherData. When entering a new filename, do not include a path (such as D:\) and do not include a filename extension (such as .txt). The filename will be appended with an extension of .txt and the file will be stored in the path specified for the error/log files.




## GRAPH OPTIONS

To display the graph window, click on the Graph icon.

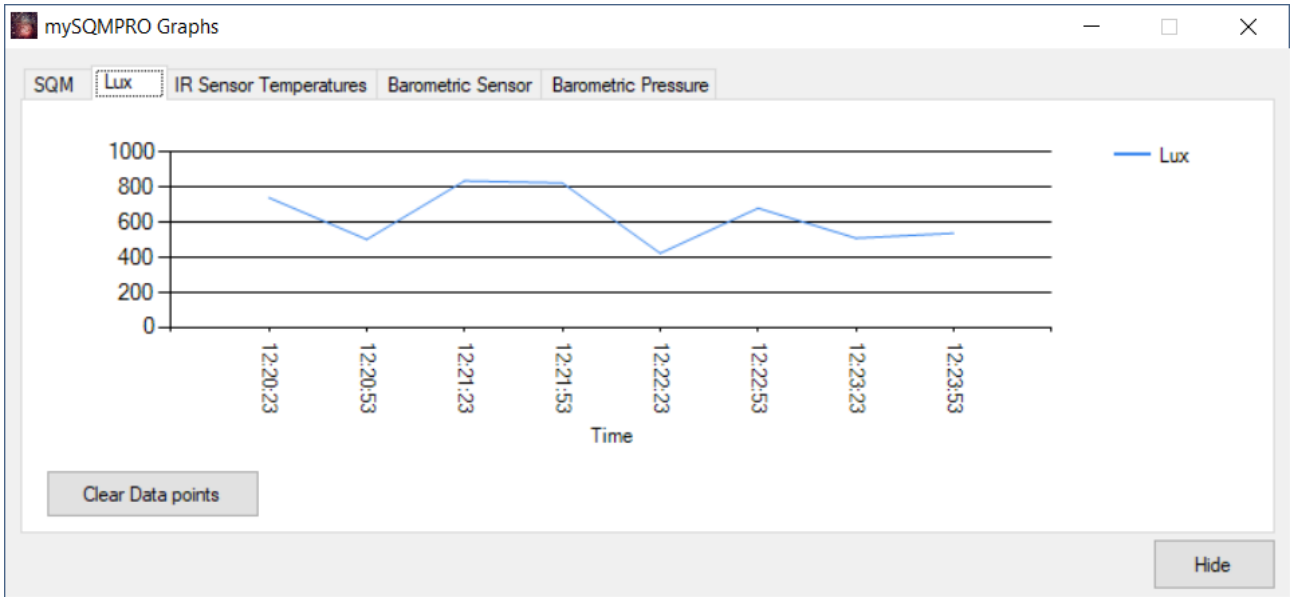
This displays the graph window, which has 5 tabs to display 4 different graphs, shown below.




Values are written to the graph when connected to a controller and the associated value is retrieved via a specific call or Get ALL button is clicked. For example, if the user clicks the Get SQM button, then the SQM returned value is also added to the SQM graph. Each graph shows the last 9 retrieved values. The y scale is recalculated whenever a new data point is added to the graph.

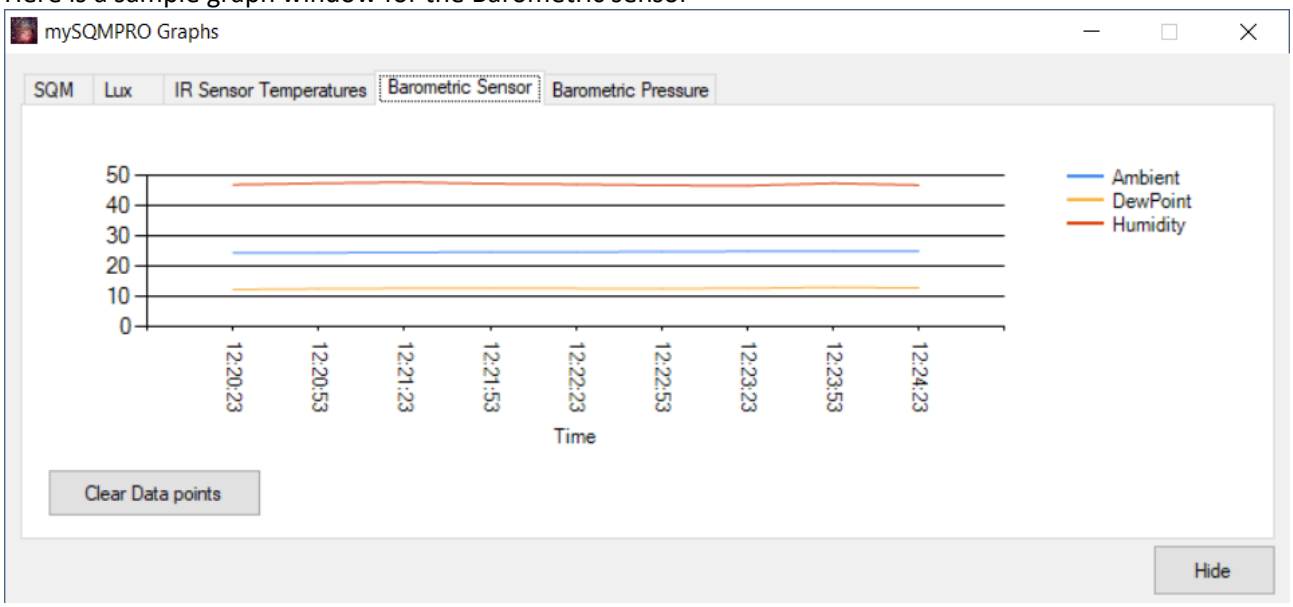
To show the IR sensor values, click the IR Sensor Temperatures tab.

To automate the graph so that values are retrieved and displayed at regular intervals, check the Automate Logging checkbox and select the update interval on the main form.

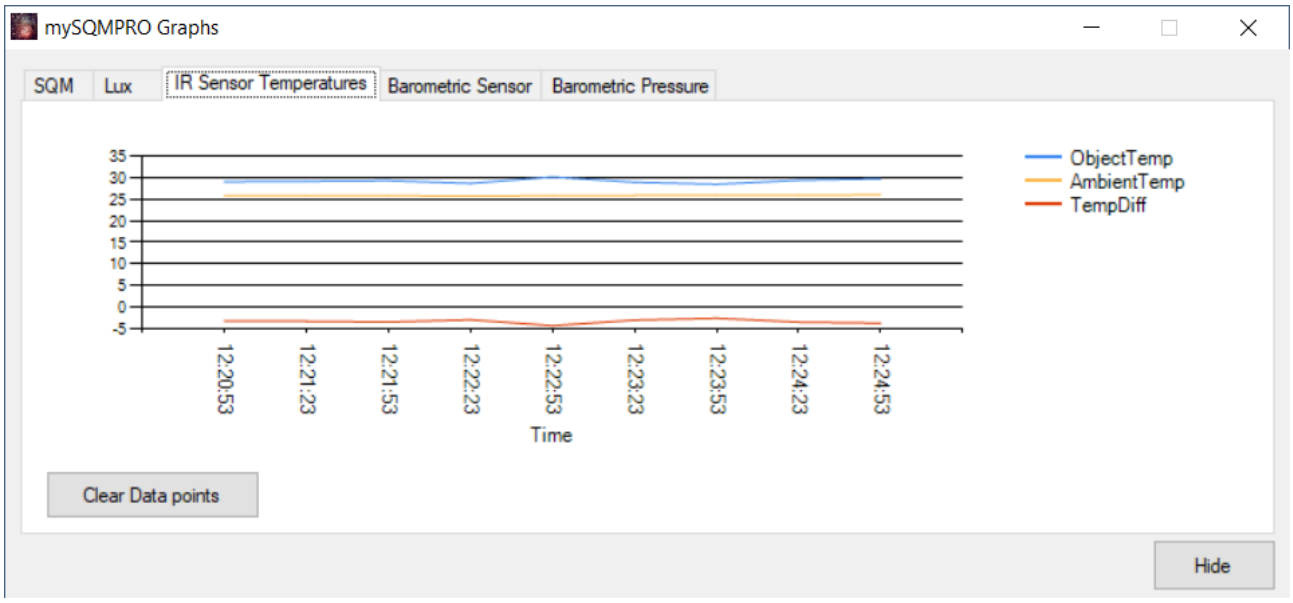



The datapoints on the graph can be cleared by clicking the Clear Data points button.

Here is a sample graph window for the Barometric sensor




and the IR Sensors



# DATA LOGGING

The mySQMPRO application can log all the data to a log file when data logging is enabled. For data logging to work, the “Automate” checkbox must be checked and the refresh interval selected.

Data logging only occurs when the application is connected to a mySQMPRO controller and the Automate checkbox is checked.

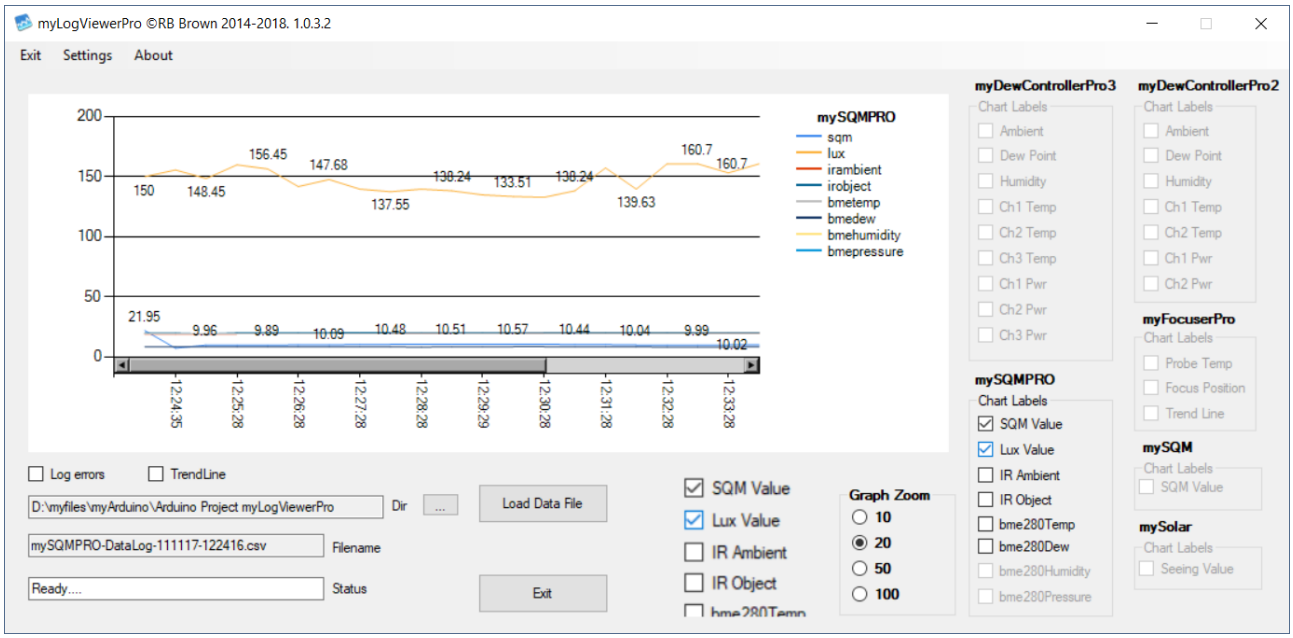
The datafile has the following format with all values separated by a semi-colon.

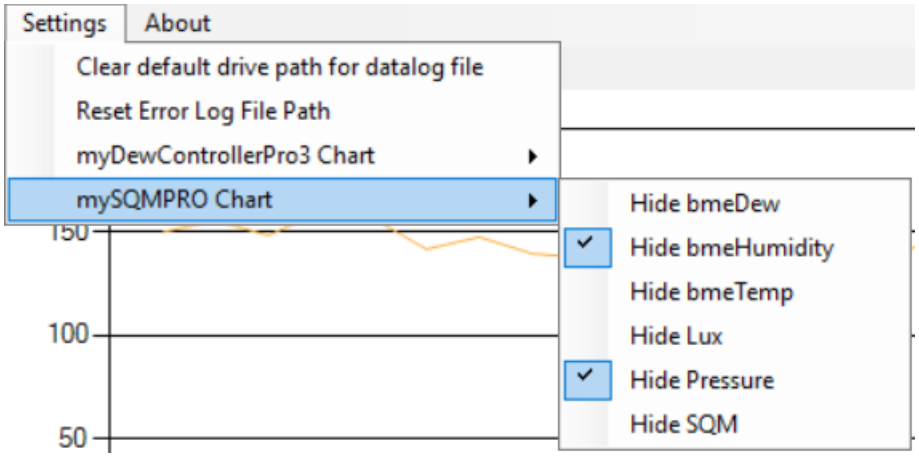
Line 1 Header Line 1 `// mySQMPRO`

Line 3 Header Line 2

Line 3 `dd/MM/yy, HH:mm:ss, SQM, lux, irsensor-ambienttemp, irsensor-objecttemp, irradiance, frequency, firmware, setpoint1, setpoint2, raining, rainsensor-voltage, NELM, BME280Temperature, BME280Humidity, BME280DewPoint, BME280Pressure, gpsdate, gpstime, gpslongitude, gpslatitude, gpssatelittes, gpsaltitude, sky-state`

Data log files can be viewed using myLogViewerPro. Not all values are shown on the graph. Please use myLogViewerPro v1.0.3.2 or higher.



# WHERE TO BUY

## ARDUINO NANO CH340G (1pc)

http://www.ebay.com/itm/Mini-USB-Nano-V3-0-ATmega328-16M-5V-Micro-controller-CH340G-board-Arduino-Cable-/201539955347

http://www.ebay.com/itm/Mini-USB-Nano-V3-0-ATmega328-16M-5V-Micro-controller-CH340G-board-Cable-Arduino-/262638369334

## PC STRIPBOARD

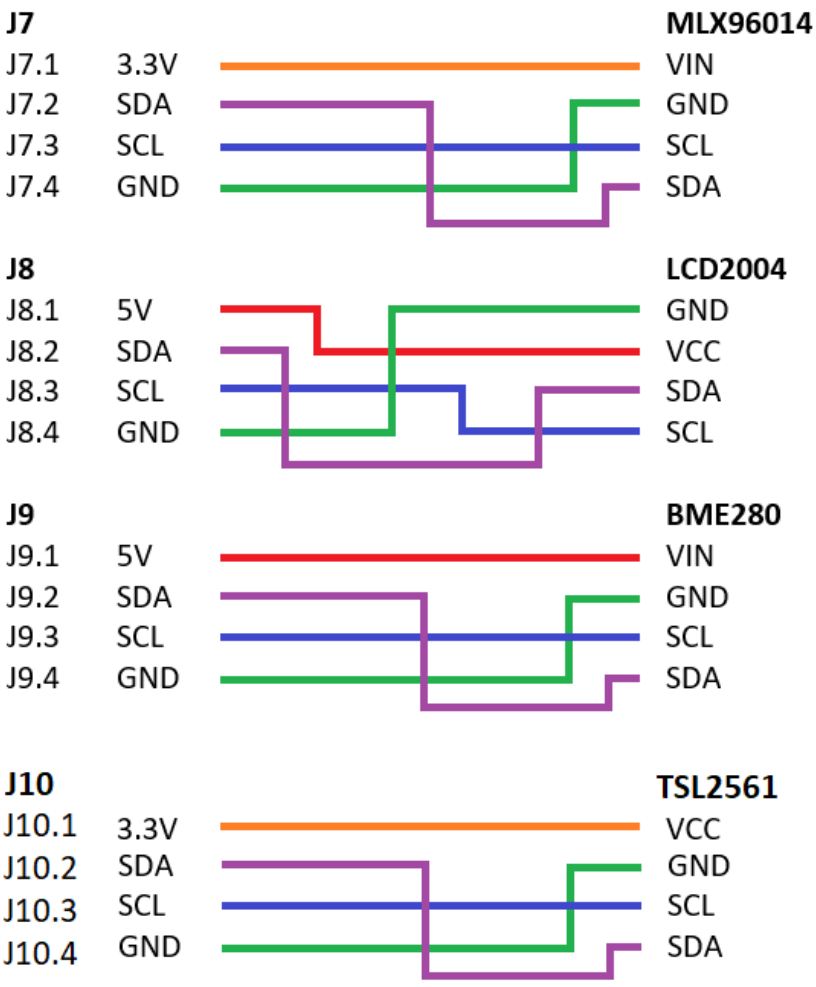
http://www.ebay.com/itm/Prototyping-PCB-Circuit-Board-Stripboard-94x53mm-/260829423263

# HARDWARE CABLES TO I2C SENSORS

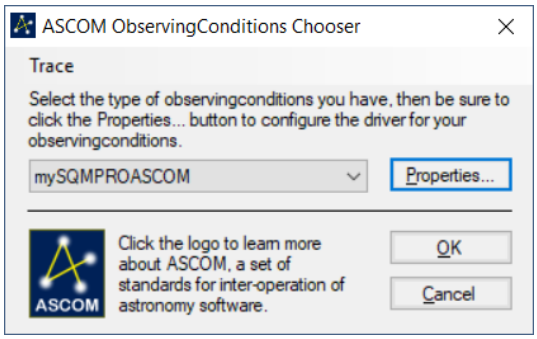
This applies to the Fritzing Rev04 PCB.

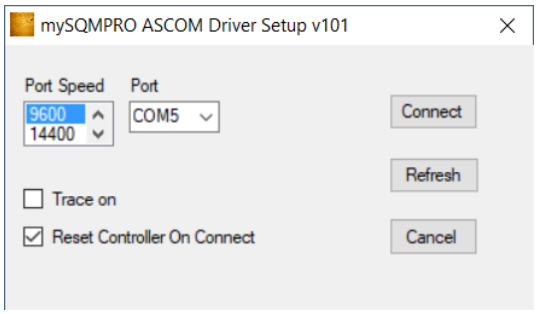
The 4pin connector on each of the I2C sensors is different, meaning a straight 4p to 4p connector cable cannot be used unless the cable wires are transposed.

**You will need to make up your own cables. Take extreme care using the correct cable with the correct connector on the PCB to the correct sensor and that the cable is orientated the correct way.**

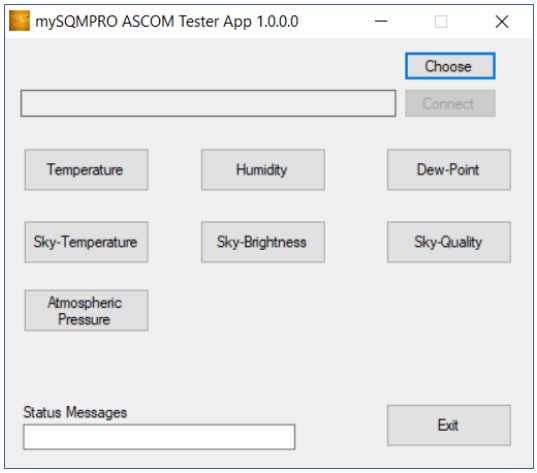


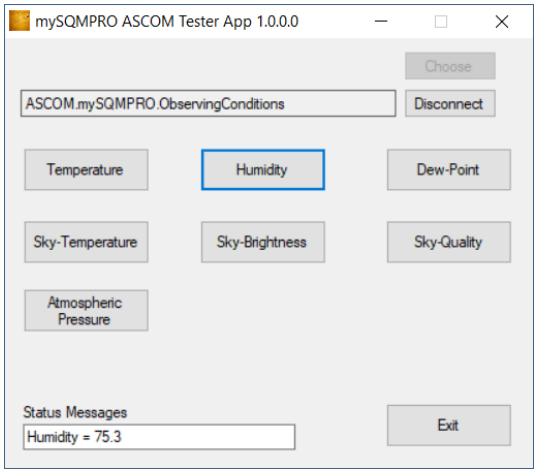
# ASCOM OBSERVING CONDITIONS DRIVER




# ASCOM OBSERVING CONDITIONS TEST APP




Gintautas posted this on 21 August 2019

In SGPro I have to use ASCOM Observing Conditions Hub instead of direct use of mySQMASCOM driver. I setup then every parameter in the Hub to look at the mySQMASCOM, set an "Average Period" and the data become updated.

# CLOUD SENSOR LOG FILE FORMAT

Date Time T V SkyT AmbT SenT Wind Hum DewPt Hea R W Since Now() Day's c w r d C A
2005-06-03 02:07:23.34 C K -28.5 18.7 22.5 45.3 75 10.3 3 0 0 00004 038506.08846 1 2 1 0 0 0

The header line is here just for illustration. It does not actually appear anywhere.

The fields mean:
Date 1-10 local date yyyy-mm-dd
Time 12-22 local time hh:mm:ss.ss (24 hour clock)
T 24 temperature units displayed and in this data, 'C' for Celsius or 'F' for Fahrenheit
V 26 wind velocity units displayed and in this data, 'K' for km/hr or 'M' for mph or 'm' for m/s
SkyT 28-33 sky-ambient temperature, 999. for saturated hot, -999. for saturated cold, or –998. for wet
AmbT 35-40 ambient temperature
SenT 41-47 sensor case temperature, 999. for saturated hot, -999. for saturated cold. Neither saturated condition should ever occur.
Wind 49-54 wind speed or:
-1. if still heating up,
-2. if wet,
-3. if the A/D from the wind probe is bad (firmware <V56 only) ,
-4. if the probe is not heating (a failure condition),
-5. if the A/D from the wind probe is low (shorted, a failure condition) (firmware >=V56 only),
-6. if the A/D from the wind probe is high (no probe plugged in or a failure) (firmware >=V56 only).
Hum 56-58 relative humidity in %
DewPt 60-65 dew point temperature
Hea 67-69 heater setting in %
R 71 rain flag, =0 for dry, =1 for rain in the last minute, =2 for rain right now
W 73 wet flag, =0 for dry, =1 for wet in the last minute, =2 for wet right now
Since 75-79 seconds since the last valid data
Now() Day's 81-92 date/time given as the VB6 Now() function result (in days) when Clarity II last wrote this file
c 94 cloud condition (see the Cloudcond enum in section 20)
w 96 wind condition (see the Windcond enum in section 20)
r 98 rain condition (see the Raincond enum in section 20)
d 100 daylight condition (see the Daycond enum in section 20)
C 102 roof close, =0 not requested, =1 if roof close was requested on this cycle
A 104 alert, =0 when not alerting, =1 when alerting

## mySQMPRO Output for CSII

- Connect to controller
- Enable checkbox CSII Format
- Set logging interval
- Enable Automate logging

| | |
|---|---|
| Date/Time | yyyy-MM-dd HH:mm:ss.ff |
| Temperature | C |
| Wind Velocity | m |
| Sky temperature | IRObject Temperature |
| Ambient temperature | IRAmbientTemp |
| Sensor case temperature | bme280temperature |
| Wind Speed | -1 |
| Humidity | bme280humidity |
| Dew Point | bme280dewpoint |
| Heater % | 0 |
| Rain Flag | Raining |
| Seconds since the last valid data | timerinterval tick / 1000 |
| Now | 038506.08846 |
| c sky state | skystate |
| w wind condition | 0 |
| r rain condition | Raining |
| d daylight condition | lux |
| C roof close | 0 |
| A Alert | 0 |

# APPENDIX A TESTING THE mySQMPRO CONTROLLER

There are test programs provided which can test each component separately. These are located in the Tests folder of the unzipped Firmware zip file.

## Using a different LDR

To test the LDR see if the value changes when you cover the LDR with your finger – if the value does not change then there is something wrong and you will need to check your circuit wiring.

IF YOU DO NOT USE THE CORRECT LDR THEN THE VALUES WILL BE DIFFERENT AND YOU WILL NEED TO MAKE SOME CHANGES TO THE SQM CODE

Running the test program will let you determine the LDR values for daylight, dusk and dark. These are values which are used to determine the period of which the frequency is measured - 100, 1000, 2000 meaning 100milliseconds, 1 second and 2 seconds

When it's really dark, the frequency is lowest, hence we need to measure over 2s, when bright sunlight the frequency is highest so we need to measure over 100 milliseconds. If in bright sunlight we measured over 2seconds, then the counters within the frequency measurement routine would overflow and we would get a false reading,

The two LDR values that set this are [you will need to alter]

```
int LDRCutoff1 = 275;          // boundary between black and dark, used to determine Gate time
int LDRCutoff2 = 500;          // boundary between dark and light
```

The values for changing the time gate period are [please do not change]

```
short period;                  // gate time - specifies duration time for frequency measurement
short blackperiod = 2000;      // 2s Gate time, at very dark sites
short darkperiod = 1000;       // 1s Gate time, at dark sites
short lightperiod = 100;       // 100ms Gate time for day light
```

The first part of the code in loop() determines what time gate period to use for measuring frequency - it does this by checking the LDRvalue and then setting the time gate period appropriately

```
 // first read LDR to determine background light level
 // and use this to set the Gate Time (duration) for frequency measurement
 LDRval = analogRead( LDRpin );
 if ( LDRval < LDRCutoff1 ) {
   period = blackperiod;    // its very dark
 }
 else if ( LDRval < LDRCutoff2 ) {
   period = darkperiod;    // its dark
 }
 else
   period = lightperiod;  // its light
```

So, first we need to determine your LDRvalues.

So load Test-LDR.ino and then compile and upload to your sqm controller. Run this program and you should see a value on the Arduino IDE serial monitor which represents the LDR value. It should be fairly constant.

If you place your finger over the LDR to block the light, the value should be lower. Take the controller into a bedroom, draw all the curtains etc, try to make the room dark as you can, but its daylight so there will still be some light, write down the LDR value you get. (we are trying to simulate dusk). We will call this value LDRCutoff2.

For the time being, make LDRCutoff1 half of your LDRCutoff2 (later you need to take it out on a dark night about midnight, hold the controller up and check what the LDRvalue is then use this as your LDRCutoff1 value

So set those new values into your mySQMPRO firmware. Now upload the updated firmware and see what happens.

## What happens if the SQM reading is inf or never changes

The SQM reading should change if you place your hand over the lens assembly. A reading that does not change indicates a wiring issue. The most common cause is failure to solder the 0.1uf capacitor between VCC and GND directly across the pins of the sensor (not on the PCB or stripboard).

## What happens if the SQM reading never falls lower than say 10 or 11

Either you live in a very light polluted area or there is some light leakage happening.

It is important that the TSL237 sensor be fitted right next to the IR filter (centered in the middle as shown).

It is important that the lens assembly be fitted and mounted correctly so that then lens can focus the received light downwards onto the TSL237 sensor.

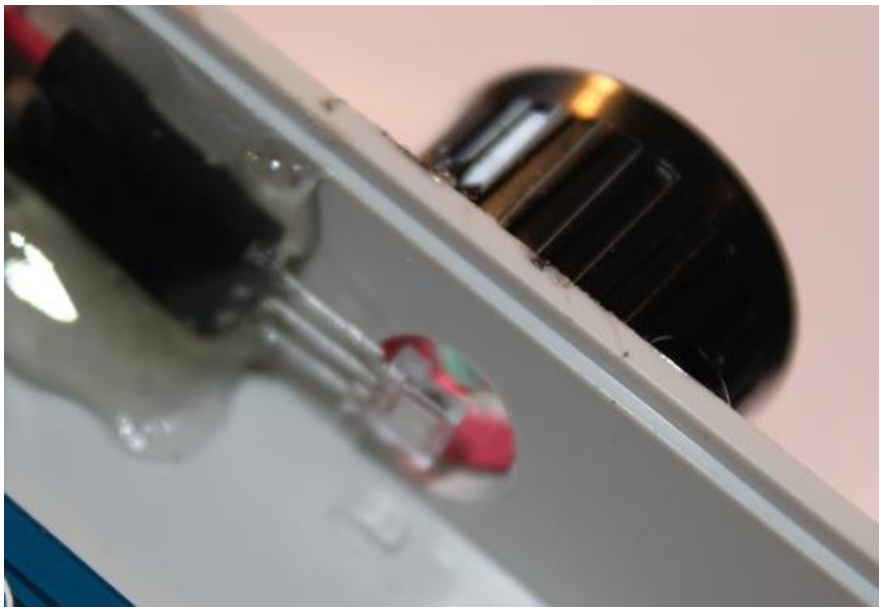
It is important that once you have fitted the TSL237 sensor, that you hot glue it in place and create a light seal so that light cannot enter from the bottom of the lens assembly where the sensor is mounted. After hot gluing the sensor in place, use dark black paint to totally cover this area so no light can enter from the bottom into the sensor or lens assembly. The 0.1uf capacitor MUST be fitted right at the sensor leads across VCC and GND.

The LDR and lens assembly needs to be fitted to the top edge of the case, away from the LCD display so that the light from the display cannot affect the readings or the LDR.

## Fitting the LDR and TSL237 and Lens Assembly and UV/IR filter

Showing LDR mounted on top edge of case, wires fed through

Showing the TSL237 sensor in place (inside of case) behind the UV/IR filter and the Lens Assembly on the outside of the case. The received light is focused by the lens assembly down through the small hole (around 8mm), through the UV/IR filter, and then onto the sensor surface. I measured the diameter of the UV/IR filter and then drilled a hole just a fraction smaller, so the filter can be fitted inside the hole as a snug interference fit and not fall out.



**Once fitted, and positioned, then use black cover tape placed over the sensor and opening to ensure no light can leak onto the sensor or filter from inside the case. Hot glue the tape in place and then spray with dark black paint to ensure there is no way light can leak onto the sensor or filter from inside the case (if using a LCD display then the backlight from the display can affect the readings)**

Looking at outside of case with lens assembly positioned over the hole where the UV/IR filter is located, then the lens assembly is hot glued in place.

## Testing the rain sensor

To test the rain sensor, wet one finger and place on the rain sensor plate. This should change to Yes – remove the finger and moisture and will revert back to No.

## Testing the IR Sensor

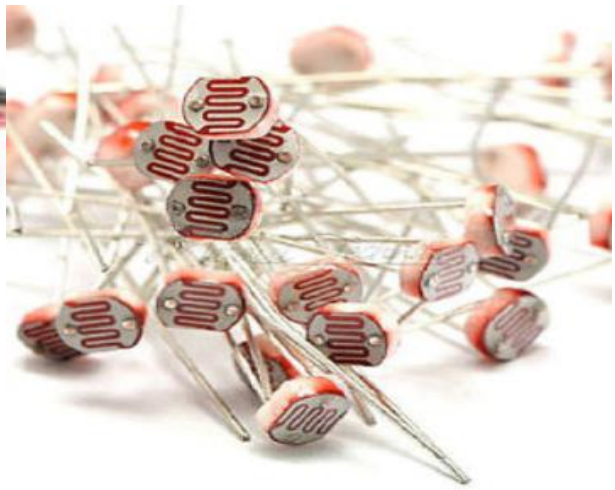
To test the IR Object temp - place your hand about 1" in front of sensor and the temperature should go up to about 33 degrees (body temp) - take your hand away and temperature should fall back down.

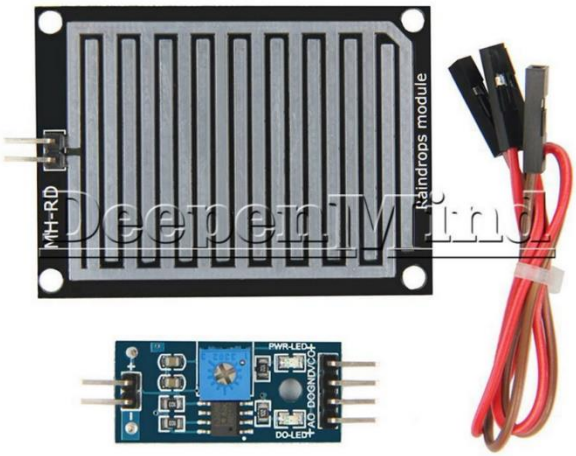
## Sky-State

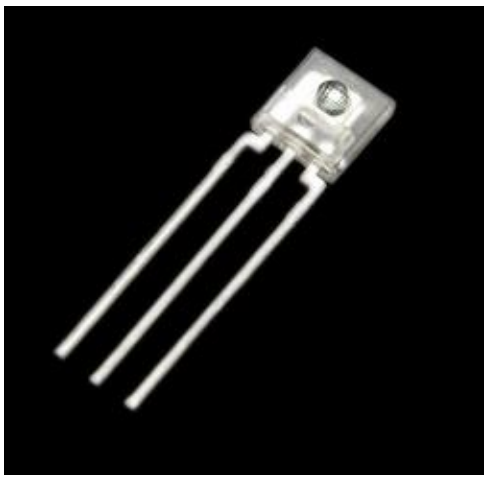
Sky-state is NOT accurate during the day so don't bother - it only works at night as it relies on the irobject temp sensor shining up at the sky - which if clear will give a very low temp reading (hence setpoint2 being 2 degrees). A good starting point for testing at night is setpoint1 of 22 and setpoint2 of 2. (setpoint1 should be close to irambient and setpoint2 should be close to irobject - set on a clear cloudless night to give indication of clear sky and then the values should be left as is).

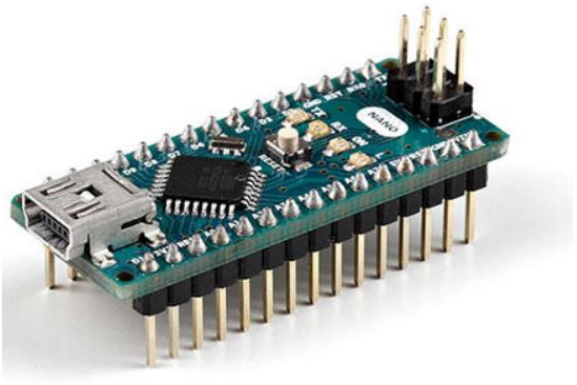
Note that a change in seasons will necessitate a change in setpoint1. The user must determine these values by trial and error. Setpoint2 should not change but will be different from location to location.

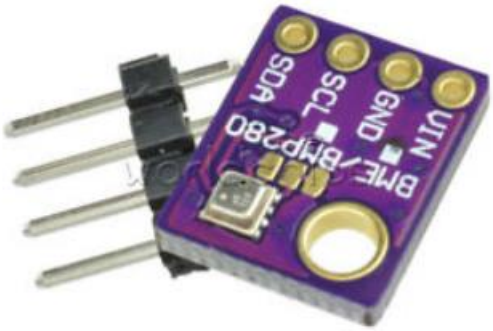
# APPENDIX B ORDERED PARTS

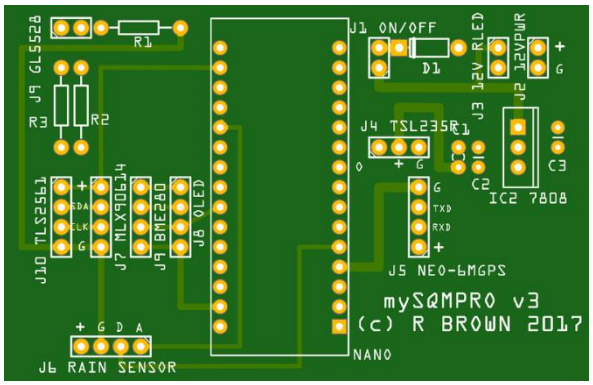





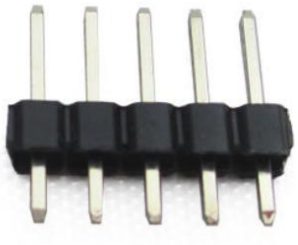



SDA
SCL
GND
VIN
BME/BMP280


J1 ON/OFF
R1
D1
J9 GL5528
R3
R2
J4 TSL2561
J10 TLS2561
J7 MLX90614
J9 BME280
J8 OLED
J5 NEO-6MGPS
J6 RAIN SENSOR
NANO
IC2 7808
mySQMPRO v3
(c) R BROWN 2017


ModuMe


TSL2561



# APPENDIX C LUX

Values read from the TLS2561 sensor are accurate.
Values read from the LDR are only an approximation.

| **Illuminance (lux)** | **Surfaces illuminated by (from Wiki)** |
|---|---|
| 0.0001 | Moonless, overcast night sky (starlight) |
| 0.002 | Moonless clear night sky with airglow |
| 0.05-0.36 | Full moon on a clear night |
| 3.4 | Dark limit of civil twilight under a clear sky |
| 20-50 | Public areas with dark surroundings |
| 50 | Family living room lights |
| 80 | Office building hallway/toilet lighting |
| 100 | Very dark overcast day |
| 320-500 | Office lighting |
| 400 | Sunrise or sunset on a clear day |
| 1000 | Overcast day |
| 10,000-5,000 | Full daylight (not direct sun) |
| 32,000-100,000 | Direct sunlight |

**Note: The TLS2561 sensor cannot return values under 0.1 lux**

# APPENDIX D SETPOINT1 AND SETPOINT2

The algorithm for determining the sky state depends on the following values

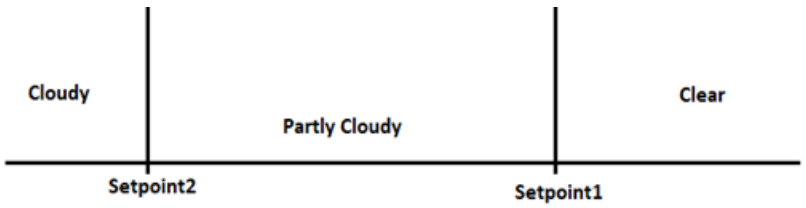
Setpoint1
Setpoint2
Temperature Difference between ambient and IR object values

```
TempDiff = mlx90614ambient - mlx90614object;
if ( TempDiff > setpoint1 )
        skystate = 0;      // clear
  else if ( (TempDiff > setpoint2) && (TempDiff < setpoint1) )
        skystate = 1;      // partly cloudy
else if ( TempDiff < setpoint2 )
        skystate = 2;      // cloudy
else
        skystate = 3;      // unknown
```

Remember that the values are in the infrared (responds to heat/cold). Clear skies at night should equate to a low object temperature reading (the clear sky acts as a huge heat sink), whilst cloudy skies should equate to a high object temperature reading.

```
// object temp is IR temp of sky which at night time will be a lot less than ambient temp
// so TempDiff is basically ambient + abs(object)
// setpoint 1 is set for clear skies
// setpoint 2 is set for cloudy skies
// setpoint2 should be lower than setpoint1
// For clear, Object will be very low, so TempDiff is largest
// For cloudy, Object is closer to ambient, so TempDiff will be lowest

// Readings are only valid at night when dark and sensor is pointed to sky
// During the day readings are meaningless
```




Under clear skies at night, determine ambient and object temperatures then calculate temperature difference and use that to determine setpoint1

Ambient = 10
Object = -7
TempDiff = 10 – (-7) = 17
Set setpoint1 initial value to 15 // set to slightly lower

Under cloudy skies at night, determine ambient and object temperatures then calculate temperature difference and use that to determine setpoint2

Ambient = 8
Object = -1
TempDiff = 8 – (-1) = 9
Set setpoint2 initial value to 10 // set to slightly higher

Now let us look at how the algorithm uses these examples to calculate the sky condition

Case 1: Ambient=12, Object=-5, TempDiff = 17, Setpoint1=15, Setpoint2=10

**if ( 17 > 15 ) skystate = 0;  // clear**

Case 2: Ambient=12, Object=-1, TempDiff = 13, Setpoint1=15, Setpoint2=10

if ( 13 > 15 ) skystate = 0; // clear

**else if( ( 13 > 10 ) && ( 13 < 15 ) skystate = 1; // partly cloudy**

Case 3: Ambient=7, Object=-1, TempDiff = 8, Setpoint1=15, Setpoint2=10

if ( 8 > 15 ) skystate = 0; // clear

else if( ( 8 > 10 ) && ( 8 < 15 ) skystate = 1; // partly cloudy

**else if( 8 < 10 ) skystate = 2; // cloudy**

**NOTE:** The values for setpoint1 and setpoint2 may need to be adjusted slightly during each season, especially summer versus winter.

# OTHER SQM PRODUCTS

These products are listed here as commercially available or DIY products. No comment is made on their features, capabilities or suitability.

**Unihedron**
http://unihedron.com/projects/darksky/

**Airy**
http://airysc.com/product_Airy_Eye.html

**Black Sky**
https://sites.google.com/view/astro-gadget/black-sky-meter

https://www.cloudynights.com/topic/652630-diy-sky-quality-meters/