LUNATICOASTRO

CloudWatcher User Manual

CloudWatcher & CloudWatcher SOLO — complete reference guide

Compiled from the official online manual (lunaticoastro.com/aagcw/enhelp) · July 2026
Lunaticoastro by Lunaticoastro Pacific Ltd. · info@lunaticoastro.com · shop.lunaticoastro.com

Contents

Overview

Getting Started

We have a video series, dedicated to the CloudWatcher and Solo, available on the Lunatico Astro YouTube channel.

If you encounter problems, then you may find the answer on our user forum.

Finally, if all else fails, please contact us at support@lunaticoastro.com where will be very pleased to assist you.

Deployment Options

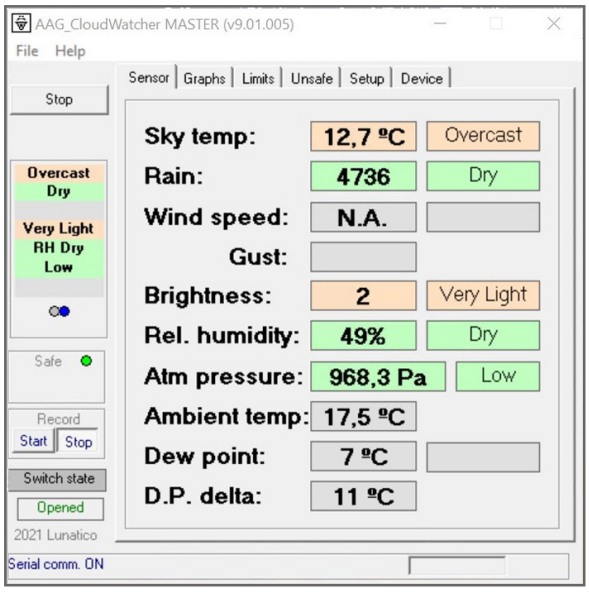
We have a video dedicated to getting started with the CloudWatcher and Solo available on our YouTube channel.

There are two ways to use your CloudWatcher:

As a Stand-Alone Product

You can connect the CloudWatcher directly to a PC using the supplied RS232 connector. If your computer does not have an RS232 port you will require an RS232 to USB adapter cable which can be purchased from most suppliers of computer equipment. Once the CloudWatcher is attached you can install the latest software package.

RS232 to USB adapter.

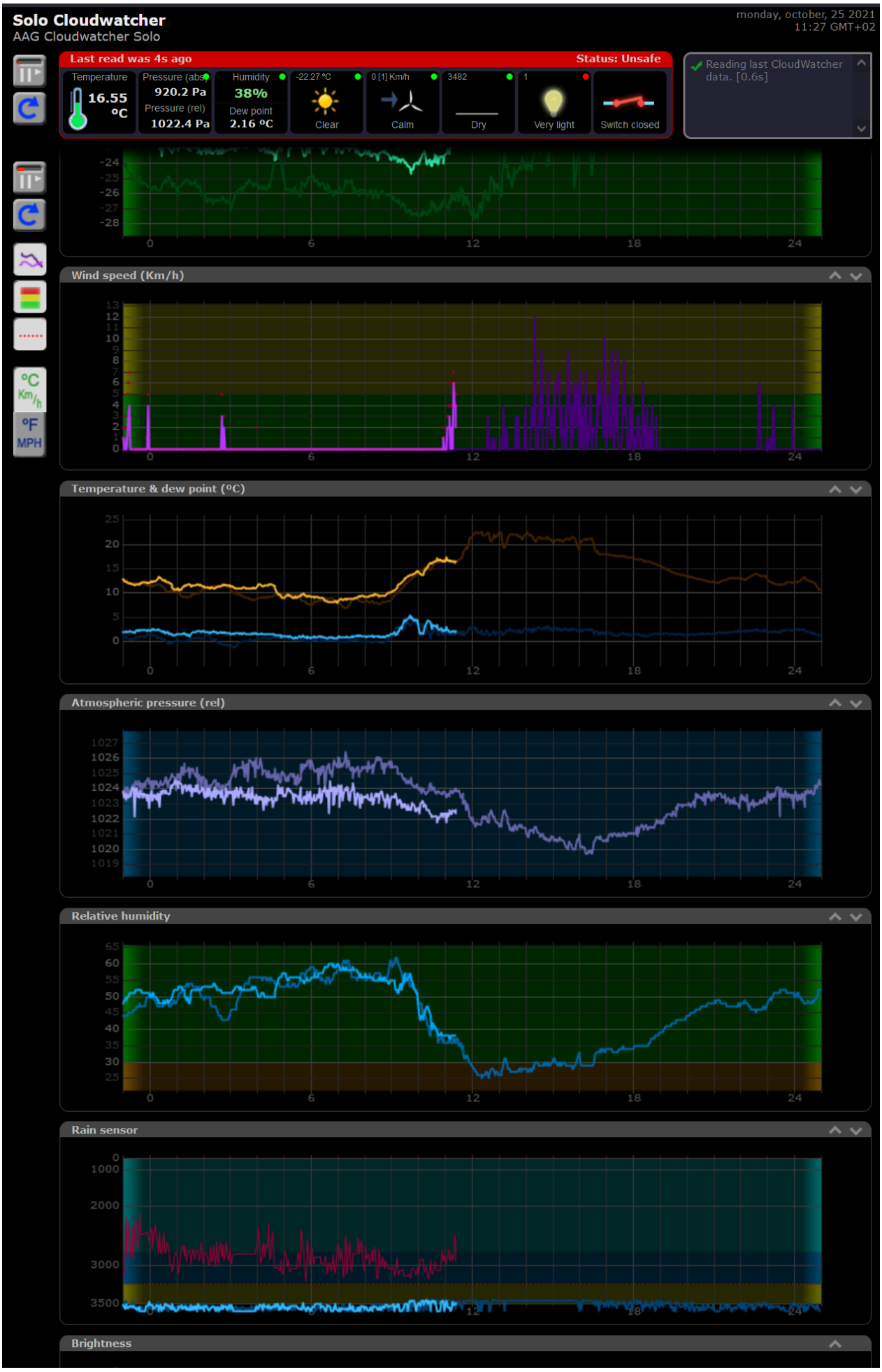


CloudWatcher data presented via PC software.

In Combination with the Solo

If you have purchased the optional Solo device, then you will need to install this first and then simply connect the CloudWatcher's RS232 cable to the Solo's RS232 port. If you are using the Solo, you do not need to install the CloudWatcher software on your PC, although you may do so if you wish. The Solo presents the information in the form of a web page and a datafile, that can be read by third party software if required. Using the Solo allows you to access the data from your CloudWatcher at any time via the Internet, even when your computer is switched off.

If you are using the Solo, you do not have to install the CloudWatcher software on your PC.



CloudWatcher data presented via Solo to a web browser.

You should spend some time determining the correct threshold values for clear, cloudy and overcast sky conditions for the region where the unit is installed, as well as the Temperature Correction Coefficients as described in section Tuning Sky Temperature Parameters.

The program default values were empirically determined for a mild climate (Algarve - south of Portugal).

Please also read the Limitations section of this manual prior to installing and using your CloudWatcher.

This software version allows different PCs in the same local area network to share information from the same CloudWatcher device. The program installed in the PC connected to the CloudWatcher must run in MASTER mode and any other PC in the network should run in REMOTE mode.

The MASTER installation reads the CloudWatcher device and stores this information in a data file which is read by the REMOTE installations. The MASTER role can also be performed by a Lunatico Solo device, which will then be physically connected to the CloudWatcher.

It is important to note that the REMOTE installations will process the information from the MASTER according to their own parameter values. For example:

- Threshold limits can be different yielding conditions that are different than those indicated by the MASTER installation.
- Unsafe Conditions can have a different composition.
- Alarm Sounds can be different.
- The sky infrared temperature is calculated according to the Temperature Correction Factors defined under Setup tab => Device section. If different from those of the MASTER installation, the sky temperature will be different from that of the MASTER installation.

Also note that the REMOTE installation:

- Has no control of the switch.
- Has no control of the rain sensor heater.

If the software is to be installed on a single PC, use it in MASTER mode (unless using a Solo, see above).

The software is an Active EXE module which can also provide information to other programs regarding cloud, rain, wind and light conditions (see Working With Other Programs)..

Software Licence

This software is freeware.

It is designed to operate with the respective hardware which is not free.

Hardware

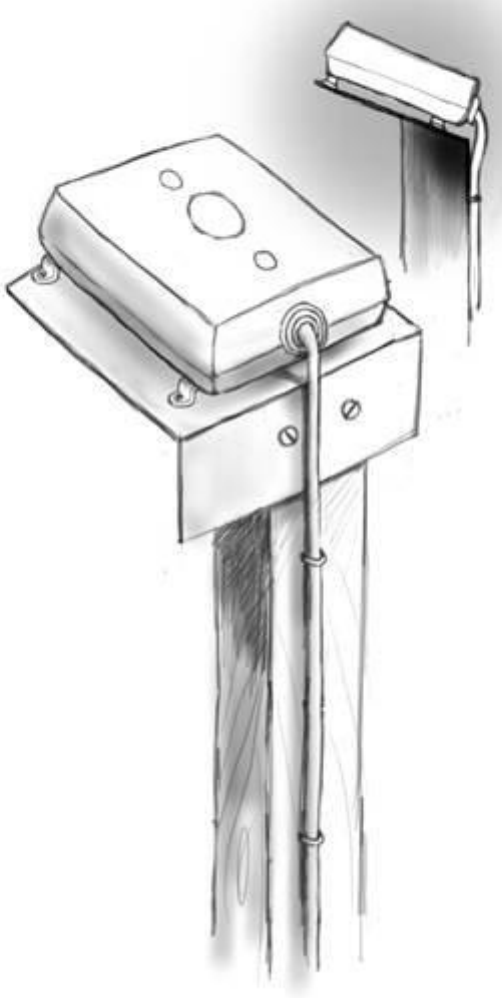
Package Contents

The CloudWatcher package contains the following:

- Main box with a: 3, 5, 7, or 10 meter cable.
- Wall mounting kit.
- Installation and maintenance notes.

Caution!
Do not remove the blue cap unless connecting an anemometer.
Do not remove sealant around the case or the coating on the rain sensor.

Physical Installation



To allow water to run off the device, it should be installed tilted at an angle of 10 to 15Âº. It is recommended to make the unit point to the zone from where clouds more usually come, thus allowing for an earlier detection. Allow some space between the CloudWatcher and the supporting surface, so water can flow out and air can freely circulate.

A location with a wide sky view and as far as possible from high-radiation areas (such as roofs or concrete flooring) is best to place the unit. If there's no option other than placing the unit over such high-radiation areas, it should be installed high above it to minimize the radiation effects.

Caution!
Avoid touching the sensors with your bare fingers.
Do not apply pressure over the sensors seeing that they are glued to the cover.
Be careful when touching the rain surface because it may be hot.
The rain sensor is an active component that is subject to heating. The maximum temperature is limited by software and it depends on the constants chosen by the user.
The rain sensor should be cleaned to work correctly.

Connecting the Unit

There are three plugs at the end of the CloudWatcher, as seen in the image:

- A DB9 connector, for the PC's serial port (or the USB/Serial converter)
- A jack plug for powering the unit. (12 or 15 V, less than 1A, any polarity)
- Two terminals for the relay output, which can be optionally used to operate the opening / closing mechanism.
- A weather resistant IP68 plug connects to our optional anemometer.

Caution!
Do not remove the blue cap unless connecting an anemometer.
Do not remove sealant around the case or the coating on the rain sensor.

Technical Specifications

- Power 12 or 15v, current < 1 A.
- RS232 Serial port.
- Size: 110x85x37 mm.
- 3m / 5m / 7m / 10 m. weather-proof cable.

Master Mode

The MASTER mode computer is connected directly to the CloudWatcher via the serial cable.

Configuring for MASTER Mode Operation

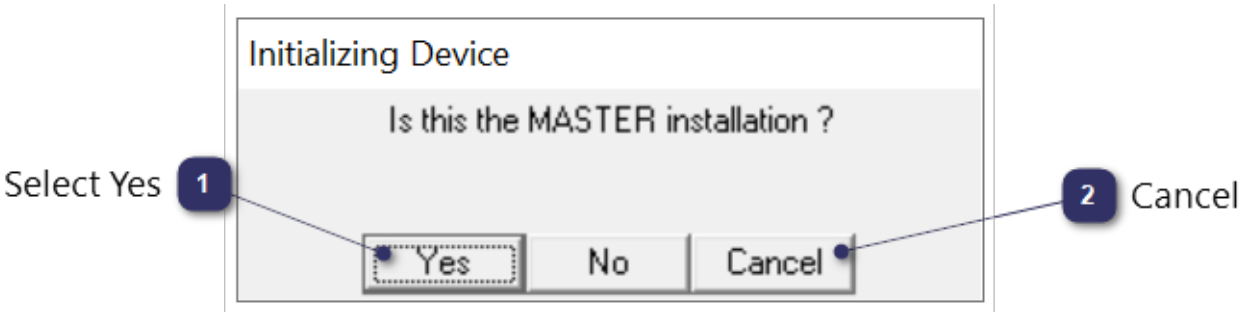
a. Run Setup.exe.

b. Create a directory in a local disk and make it shareable to other users in the local network.

c. Connect the CloudWatcher before starting the program.

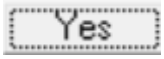
d. Start program.

e. The program will prompt you:



1 Select Yes

Selects Master mode.

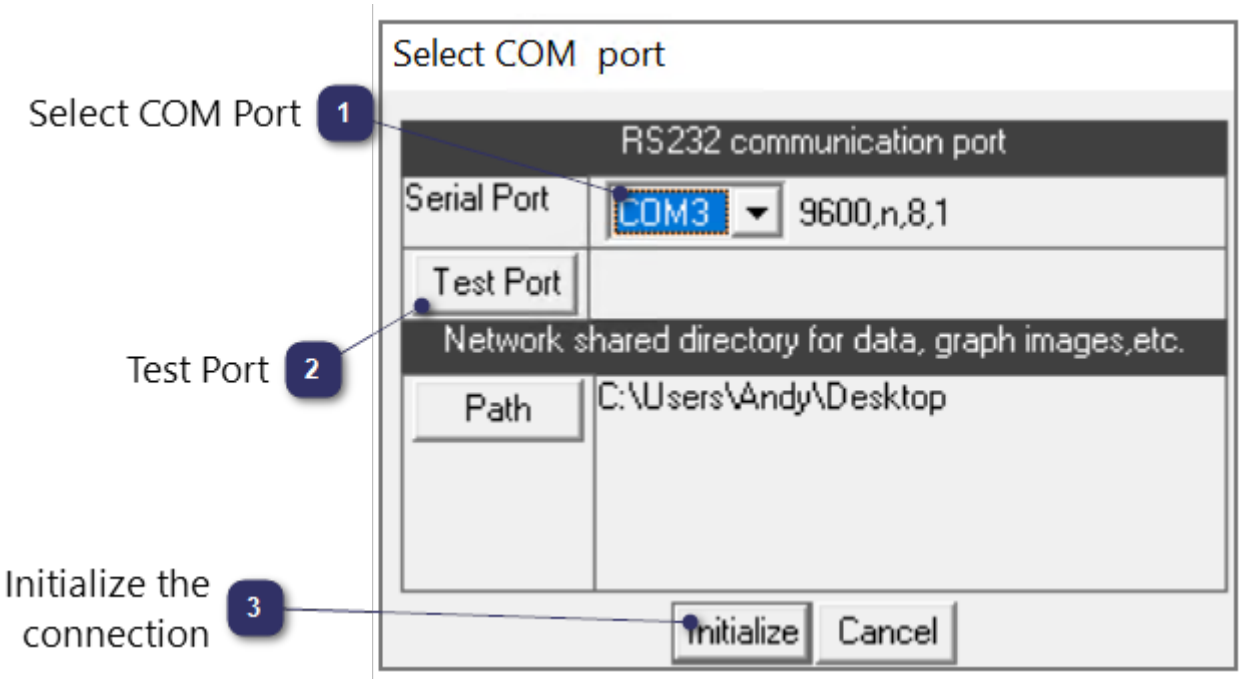


2 Cancel

Leave this dialogue box.

f. Select Yes for MASTER installation.

g. The program will display the following window:



1 Select COM Port

Select COM port that connects to the CloudWatcher.

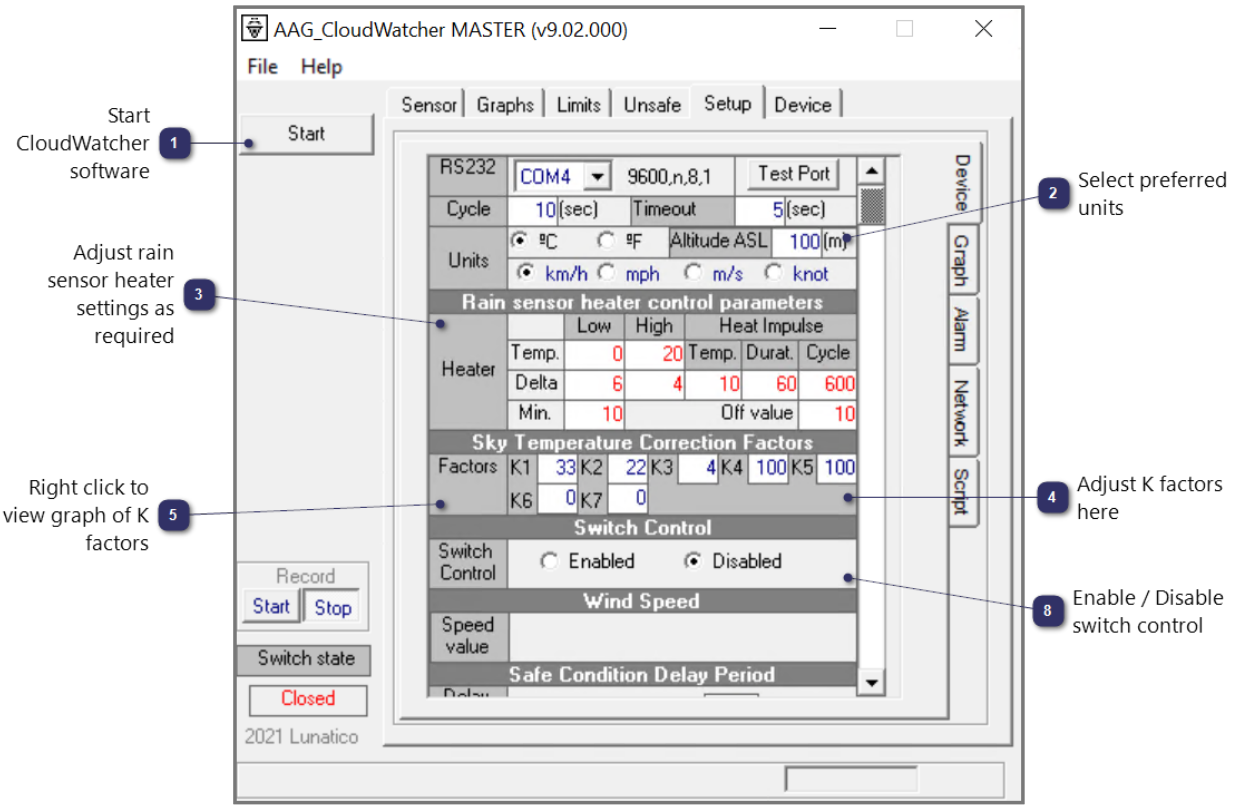
2 Test Port

Test the serial connection.

3 Initialize the connection

Connects to the CloudWatcher

h. Check the Setup tab => Device section.



1 Start CloudWatcher software

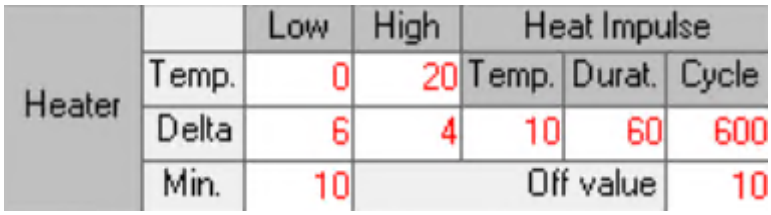
Connects to the CloudWatcher and presents weather data. If correctly configured, writes the Parameters Transfer File. (See Remote Mode.)

2 Select preferred units

Select required combination of units. Altitude allows calculation of sea level pressure.

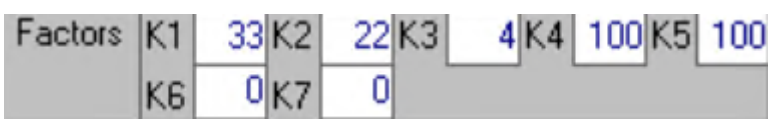
3 Adjust rain sensor heater settings as required

Items in red can be amended. You may wish to start with default values.



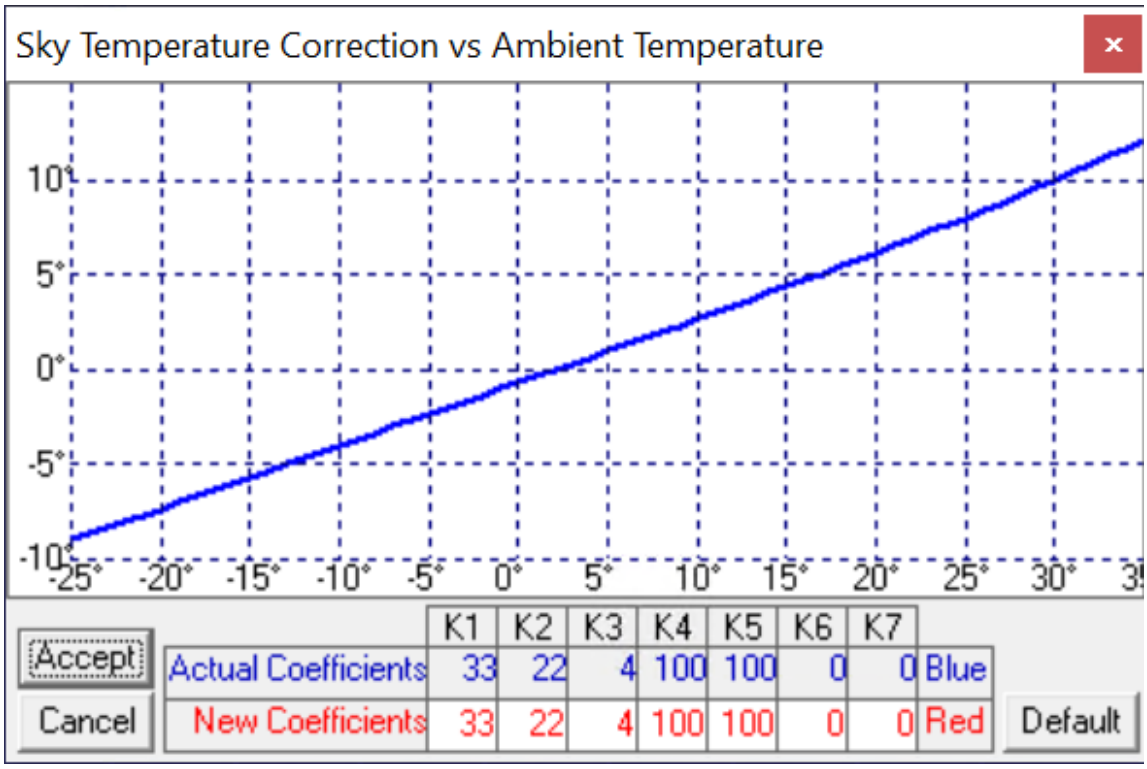
4 Adjust K factors here

The K factors can be to local conditions.



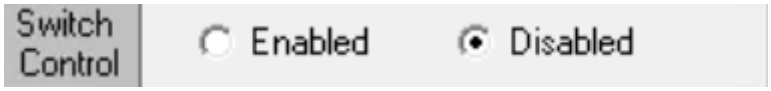
5 Right click to view graph of K factors

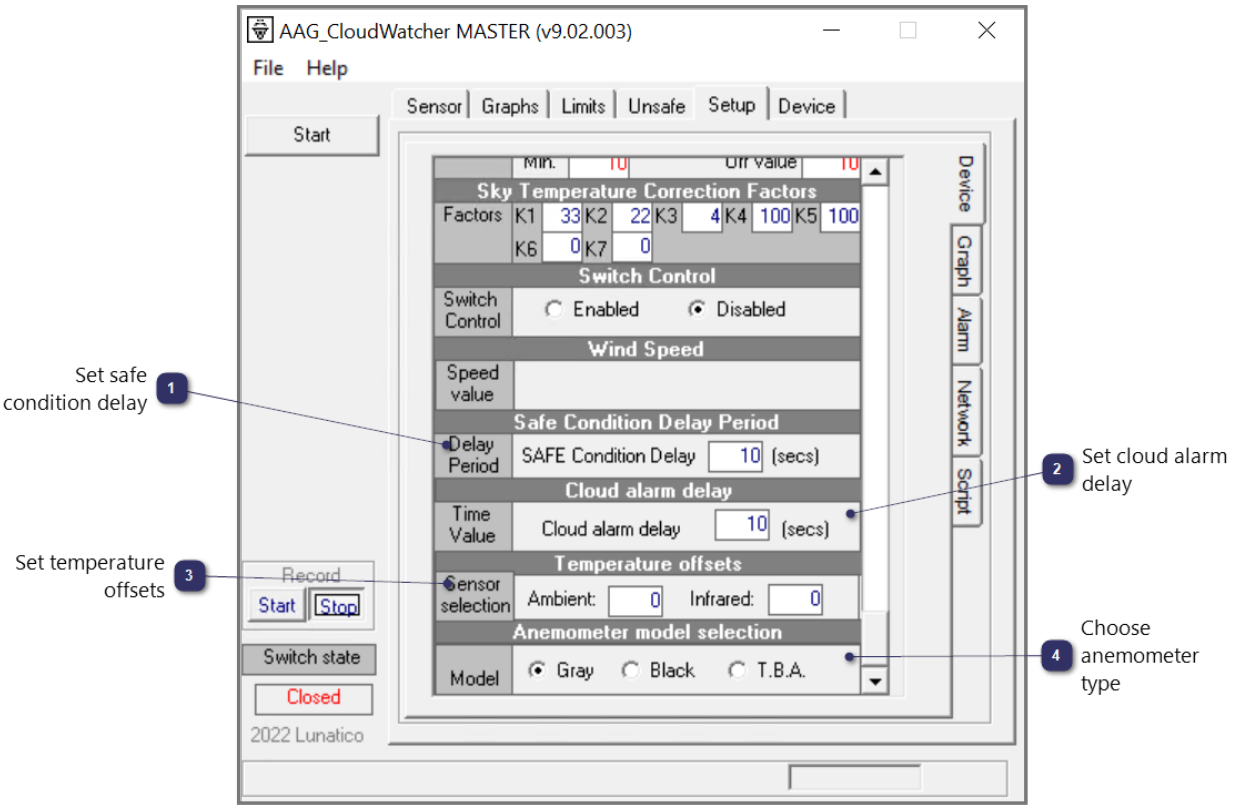
This can be helpful in visualising the effect of constants.



8 Enable / Disable switch control

If you intend to use the inbuilt switch, ensure it is enabled here.





1 Set safe condition delay

The CloudWatcher waits for this length of time after conditions become safe before reporting safe.

2

Set cloud alarm delay

The CloudWatcher waits for this length of time after detecting cloud before reporting unsafe.

3

Set temperature offsets

Allows tuning of temperature indictations.

4

Choose anemometer type

Older units have gray anemometers, more moder units are black.

Remote Mode

Any computer not directly connected to the CloudWatcher is said to be in REMOTE mode.

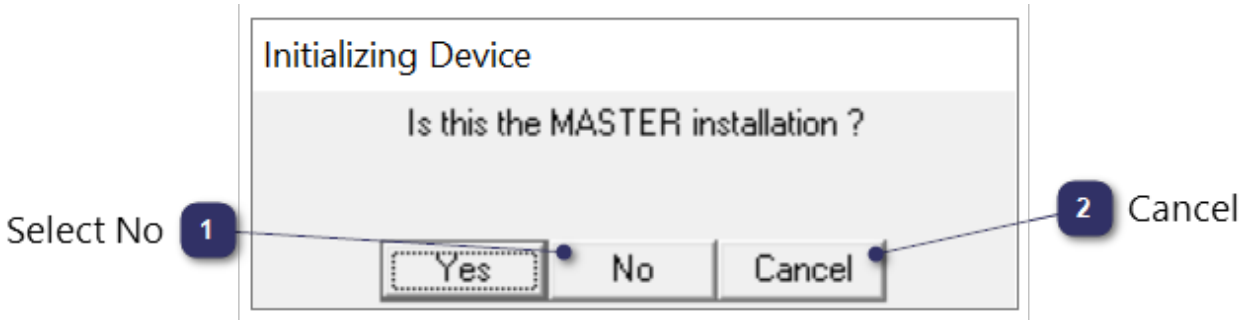
Setting up the CloudWatcher software in REMOTE mode is similar to the MASTER mode. The main difference is that the REMOTE computer needs access to the Parameter Transfer File and some settings are not available.

Configuring for REMOTE Mode Operation

a. Run Setup.exe.

b. Start program.

c. The program will prompt you:



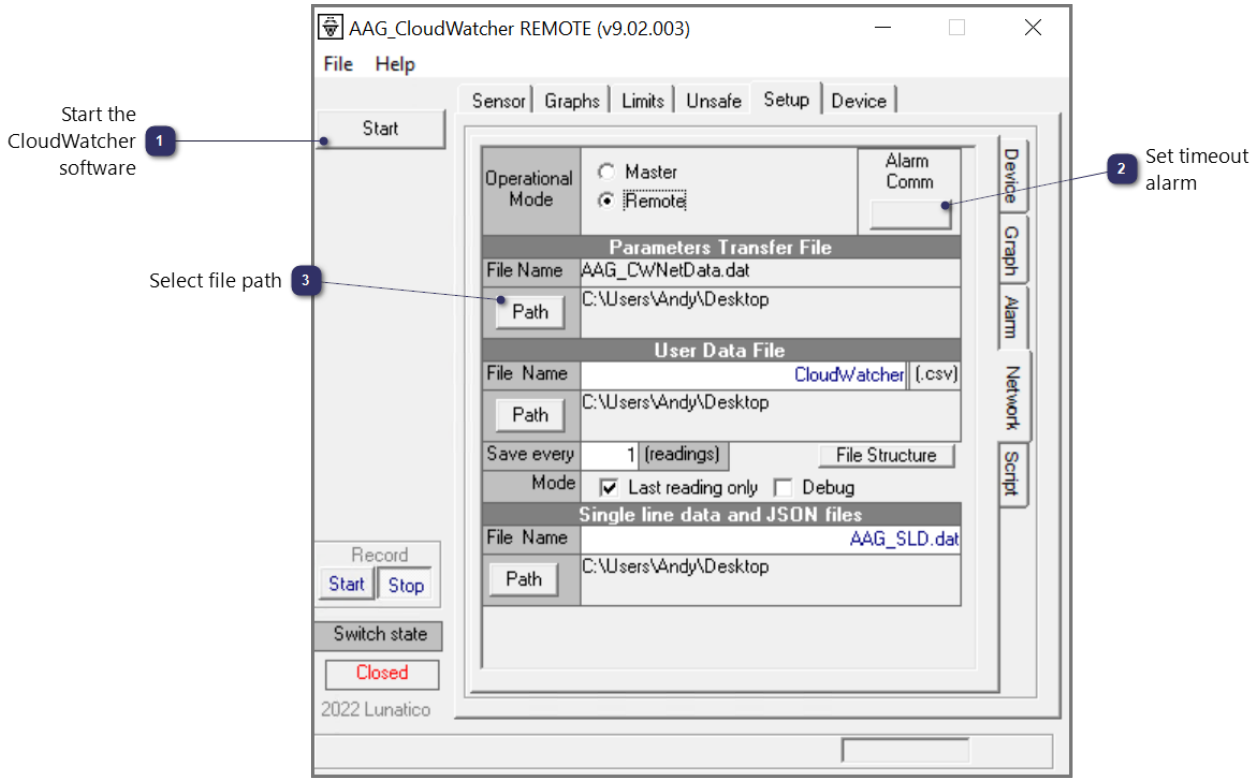
1 Select No

Selects Remote mode.

2 Cancel

Leave this dialogue box.

d. Select 'No' for REMOTE mode operation.



1 Start the CloudWatcher software

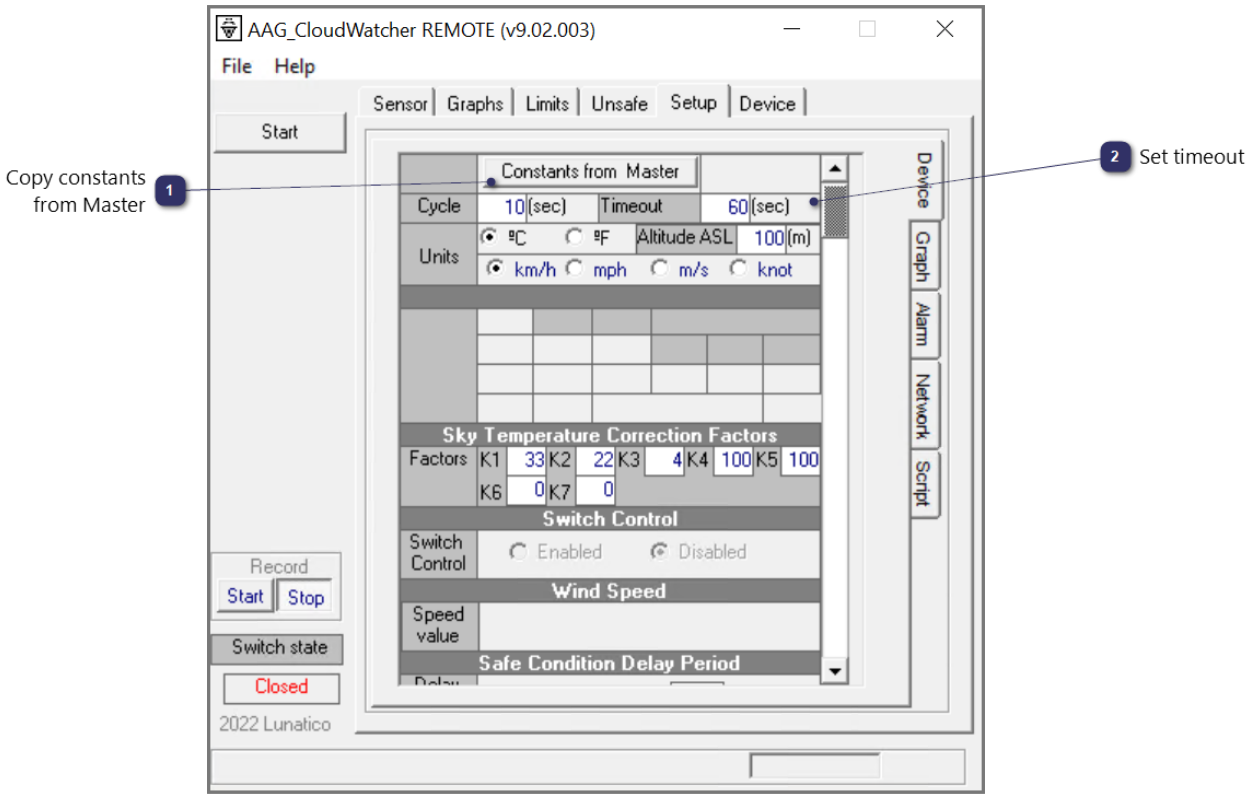
Starts reading the Parameters Data File and presents weather data.

2 Set timeout alarm

Right click to select an audio alarm. Left click to make the alarm active. The alarm sounds when the datafile is older than the Timeout defined in Setup=>Device tab.

3 Select file path

Select the location of the Parameters Tranfer File. If using a Solo, this will be the Solo's shared drive.



1 Copy constants from Master

Overwrites Sky Correction Factors with values from the Master.

2 Set timeout

Sets the time out used by the audio alarm on the Setup=>Network tab.

The Common Controls and Indications form an L around the Tabs section and can always be displayed, irrespective of the tab selected.

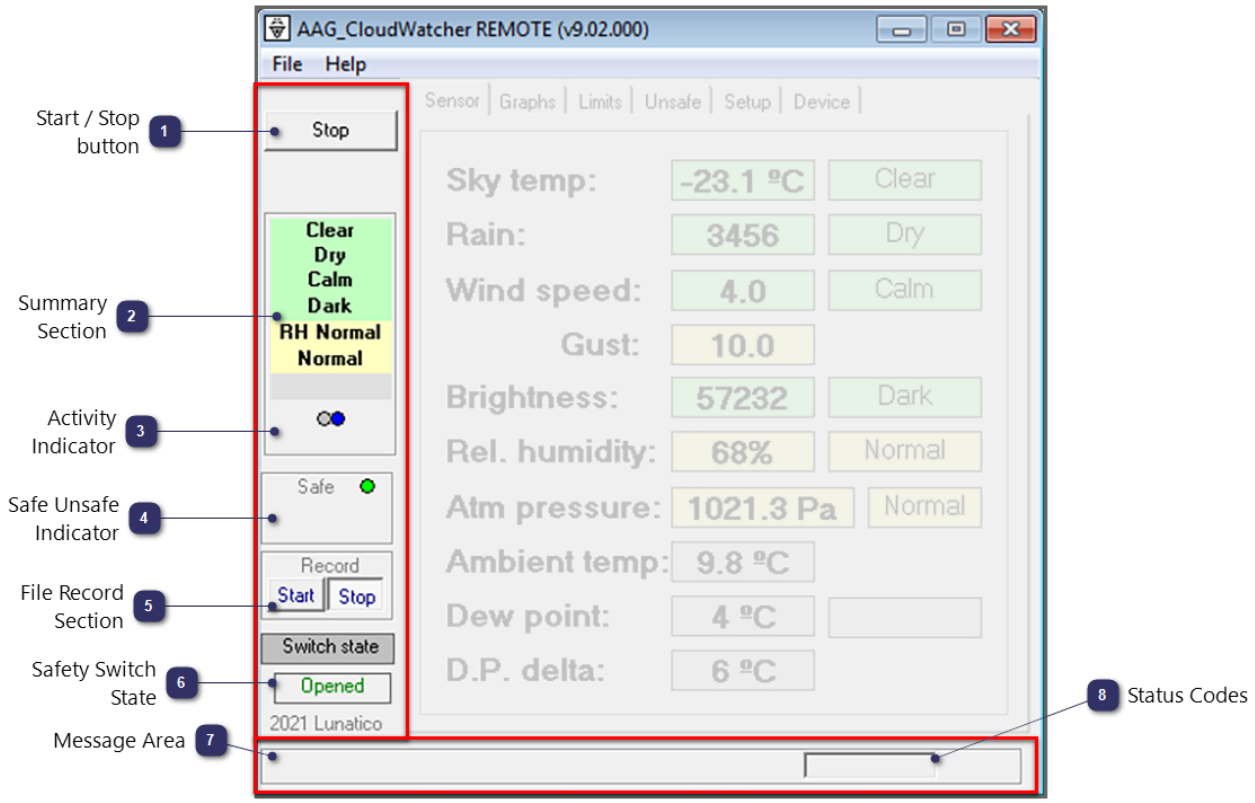
File and Help Menus

The File menu contains two entries:

- Start / Stop - Replicates the function of the Start / Stop button (see below). Start causes the software to continually update the information on the Sensor page from the CloudWatcher unit. Stop ceases the update process and removes the summary information from the left hand side of the screen. Note, when the Stop button is pressed the Sensor tab continues to display the last readings received from the CloudWatcher.
- Exit. Closes the CloudWatcher program.

The Help Menu contains two entries:

- Contents - Links to the inbuilt version of this manual.
- About - Gives information about the software version.



1 Start / Stop button

In Master mode, connects the software to or disconnects from the CloudWatcher.
In Remote mode, starts / stops reading the Parameters Transfer File.

Summary Section

2 Contains a summary of the weather conditions as defined on the Limits tab.

3 Activity Indicator

A blue dot moving left and right indicates that the CloudWatcher software is communicating correctly with the hardware unit.
A red dot indicates that the software is synchronising with the CloudWatcher.
An orange dot indicates a problem reading from the Parameters Transfer File.

4 Safe Unsafe Indicator

Indicates if the weather is safe or unsafe as defined on the Unsafe tab. The lower area displays a button to silence an audio unsafe alarm.

5 File Record Section

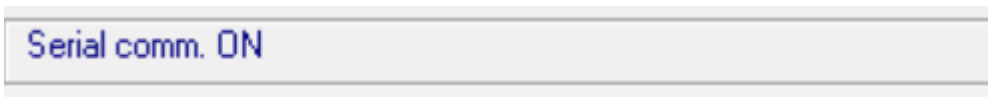
A .csv file can be written according to parameters set on the Setup => Network tab. Start and stop recoring using the respective buttons.
Care should be taken if using this feature to create log files as they may quickly become very large

6 Safety Switch State

The state of the CloudWatcher safety switch is displayed here.
Note, that if disabled (see Setup => Device, Switch Control), the device retains the last position. This can lead to a disagreement between the safe / unsafe state and the state of the switch.

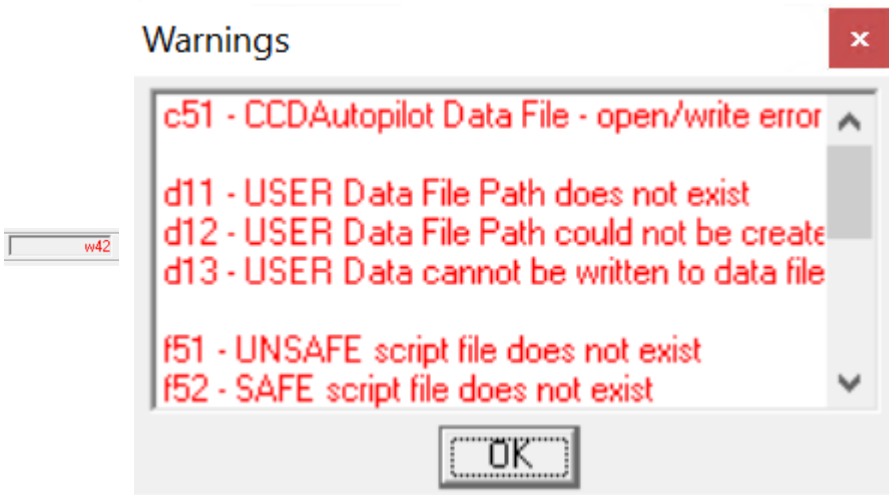
7 Message Area

System messages are displayed here.



8 Status Codes

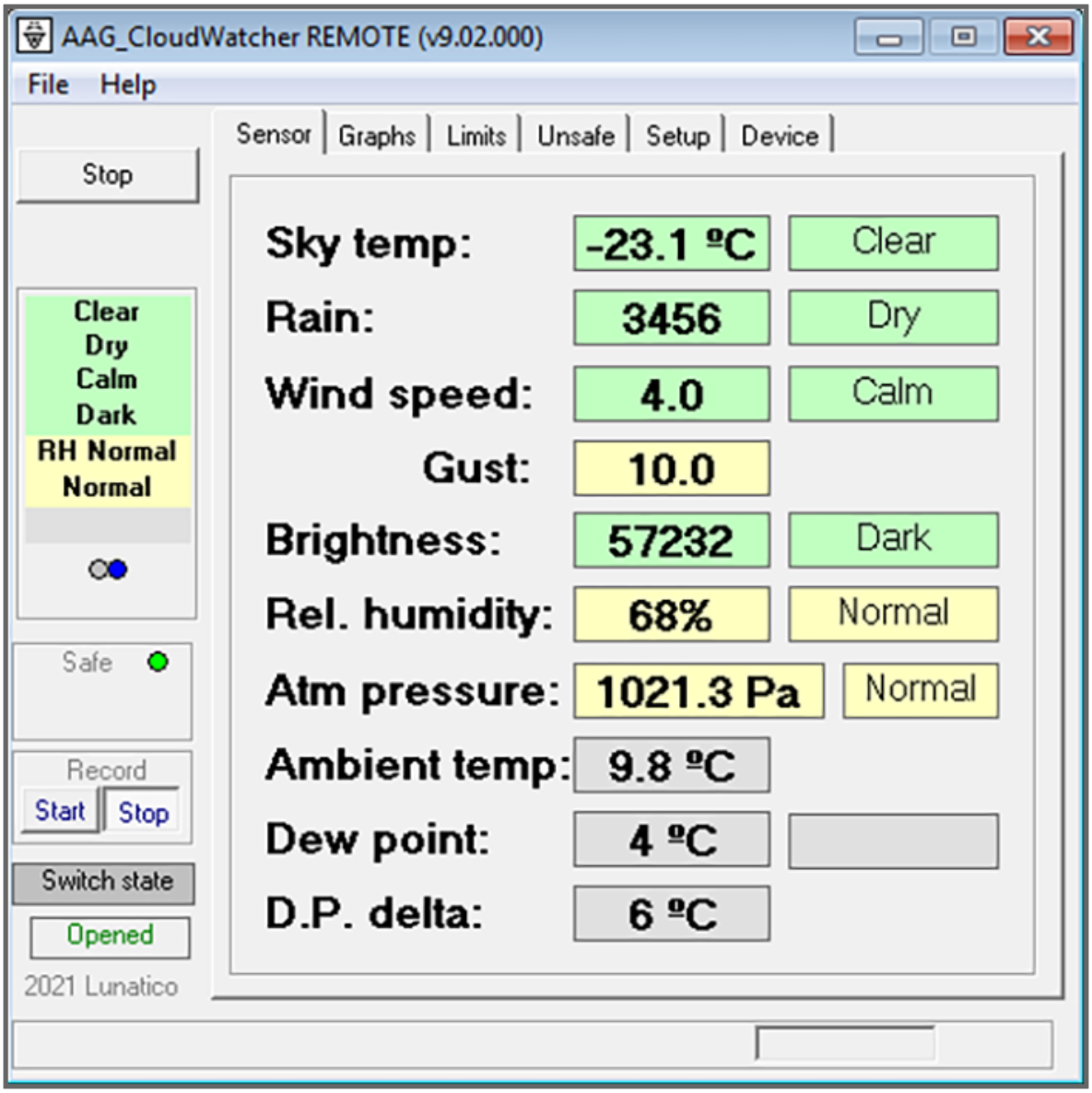
Three digit, alphanumeric codes are displayed in this window. Right click to view list of codes. Full details of codes can be found at Appendix 2.



Sensor Tab

The Sensor tab displays the primary parameters measured by the CloudWatcher. Some parameters such as the Wind Speed, Gust and Relative Humidity and Atmospheric Pressure are only available if the associated, optional sensors have been attached (anemometer) or fitted (atmospheric pressure and relative humidity).

Should you wish to upgrade your CloudWatcher at a later date, you can purchase additional components here. Please contact support@lunaticoastro.com for further advice on adding the Atmospheric Pressure and Relative Humidity sensor.



Notes:

Colours and subjective descriptions such as 'Clear', 'Dry' and 'Calm' are defined on the Limits tab.

'D.P. delta' is the rounded difference between the ambient temperature and dew point.

'Ambient temp' is temperature supplied by the Relative Humidity sensor. If this is not available, the temperature from the IR sensor is used. The source in use can be verified by reference to the Device tab.

Graphs Tab

This is an informational tab which displays the environmental conditions related to:

- Cloud
- Rain
- Wind
- Speed
- Brightness
- Ambient Temperature
- Humidity
- Atmospheric Pressure

The availability of: atmospheric pressure; humidity; and wind graphs depends on the specification of the CloudWatcher.

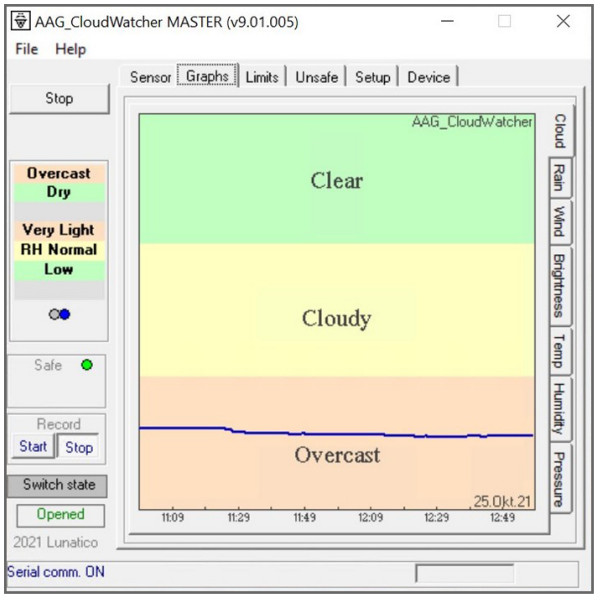
The horizontal axes are labeled in real time hours.

The backgrounds of Cloud, Rain, Wind, Brightness, Humidity and Pressure graphs are painted with the colours defined for the threshold limits undertab.

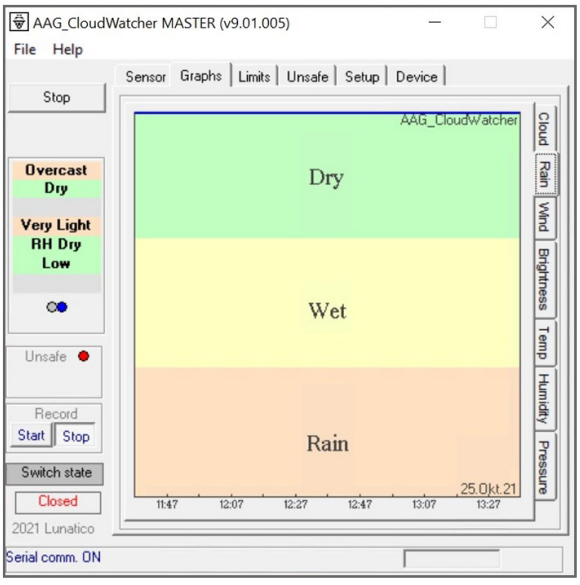
The Temperature graph's background, as well as other control parameters, such as number of points, recording frequency, label color, graph color, etc., are defined under the Setup => Graphs tab.

Graphs can be automatically exported on an individual basis in either .png or .jpeg formats. This behaviour is also determined by settings on the Setup => Graphs tab.

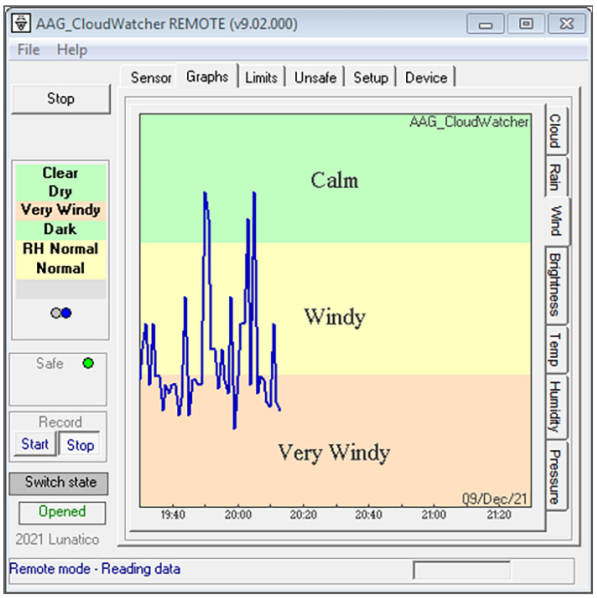
Right-click on the Temperature Graph in order to adjust the y-axis.



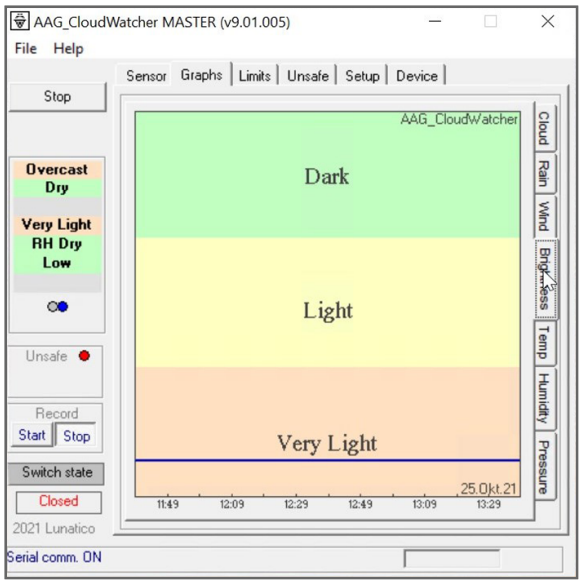
Cloud Section



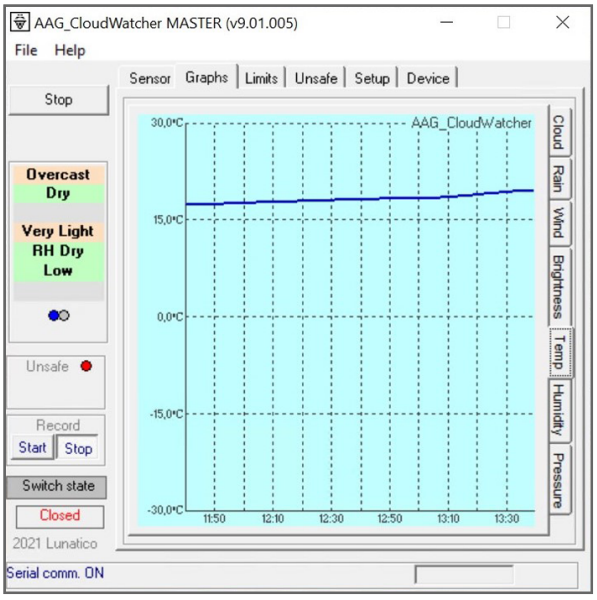
Rain Section



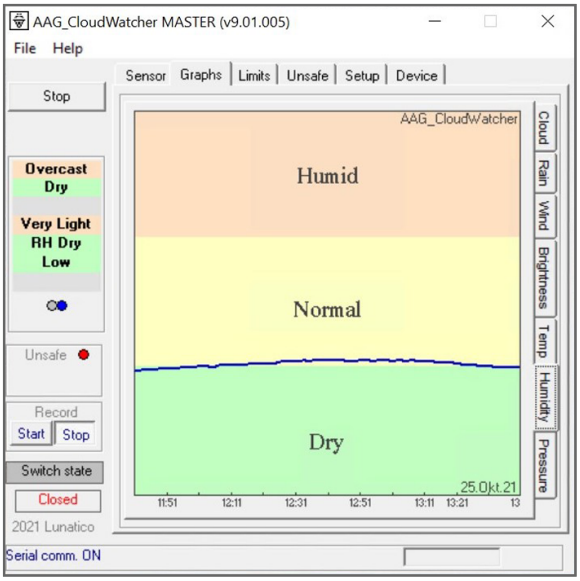
Wind Section



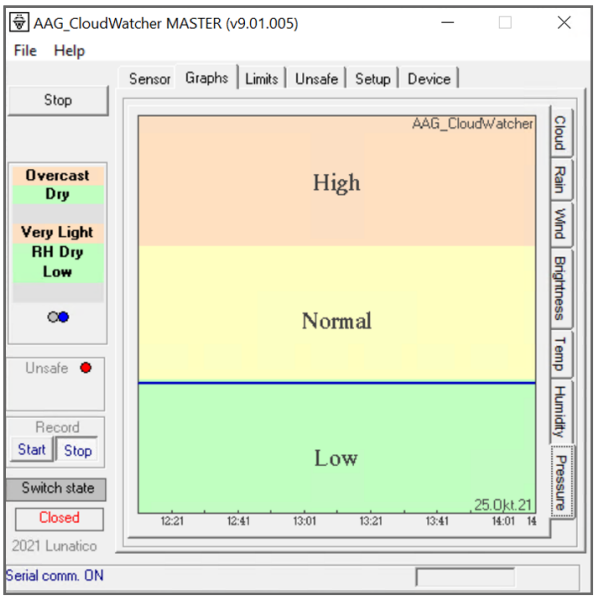
Brightness Section



Temperature Section



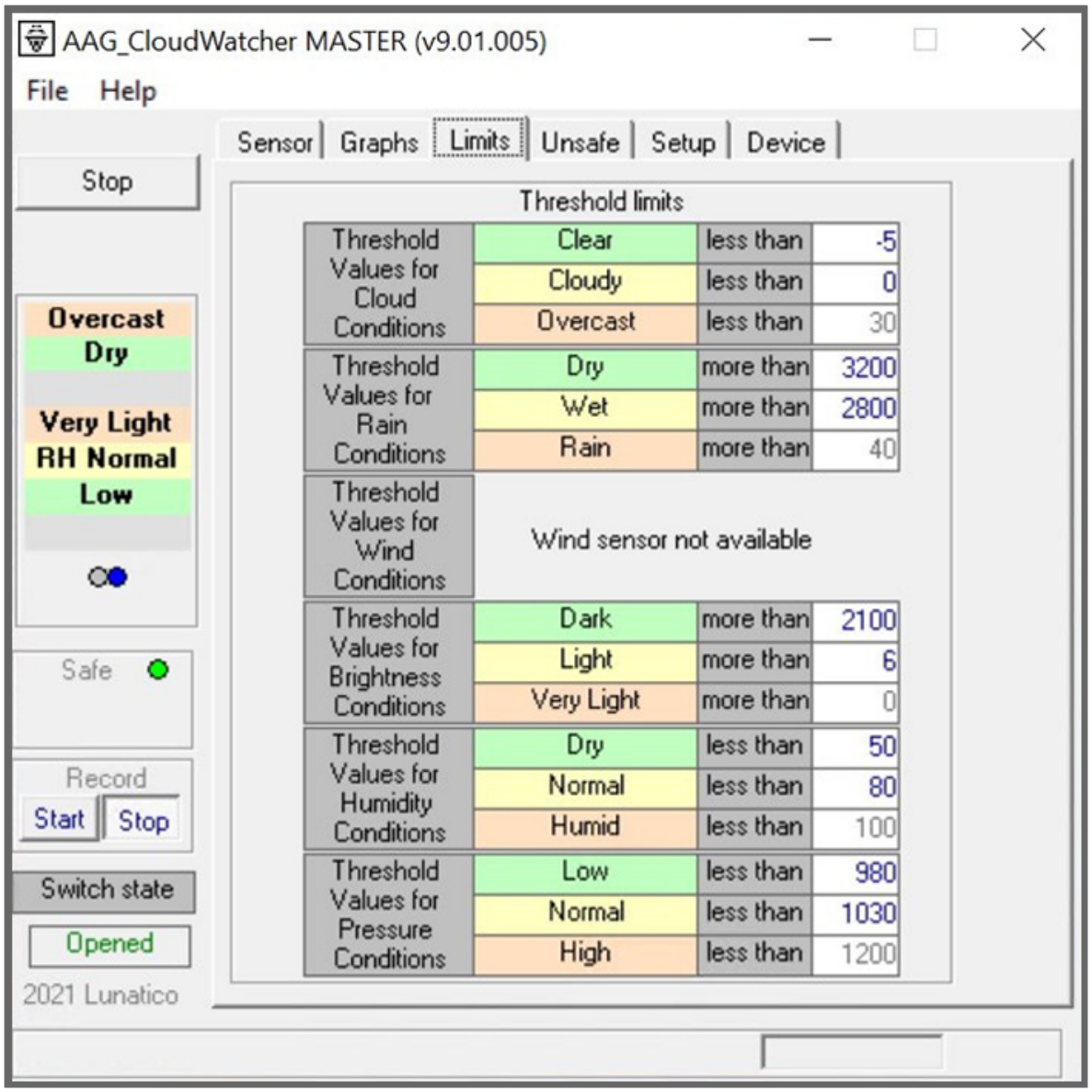
Humidity Section



Pressure Section

Limits Tab

This tab allows the definition of the threshold limits associated with each environmental condition. The colour and text for each condition can also be edited.



The default values have been empirically determined for a mild climate (Algarve - south of Portugal).
You will need to determine by observation, the best values for the region were the unit is to be installed.

In figure above, the Clear condition has been defined for sky infrared temperatures less than -5ÂºC, the Cloudy condition for sky infrared temperatures ranging between -5ÂºC and 0ÂºC and Overcast condition (very cloudy) for infrared temperatures between 0ÂºC and 30ÂºC. Note that the Overcast value of 30ÂºC corresponds to Cloud 'Unknown' condition defined in the Setup tab => Alarms section.

The Dry condition has been defined for values greater than 3200 cycles, the Wet condition for values ranging between 3200 and 2800 cycles and Rain condition for values between 2800 and 40 cycles. Note that the Rain value of 40 cycles corresponds to Rain 'Unknown' condition defined under Setup tab => Alarms section.

The Calm condition has been defined for values smaller than 5 km/h, the Windy condition for values ranging between 5 – 10 km/h, whereas the Very Windy condition for values ranging between 5 – 99 km/h.

The Dark condition has been defined for values greater than 2100 K ohms, the Light condition for values ranging between 2100 – 6 K ohms and the Very Light condition for values ranging between 6 – 0 K ohms.

The temperature and speed units correspond to those defined in the Setup tab => Device section.

The Cloud Condition threshold limits are subject to the following rules:

- Clear value is the smallest.
- Overcast value is the largest.
- Cloudy value must be smaller than Overcast value but larger than Clear value.

The Rain Condition threshold limits are subject to the following rules:

- Dry value is the largest.
- Rain value is the smallest.
- Wet value must be larger than Rain value but smaller than Dry value.

The Wind Condition threshold limits are subject to the following rules:

- Calm value is the smallest.
- Very Windy value is the largest.
- Windy value must be smaller than Very Windy value but larger than Calm value.

The Brightness Condition threshold limits are subject to the following rules:

- Dark value is the largest.
- Very Light value is the smallest.
- Light value must be larger than Very Light value but smaller than Dark value.

The Humidity Condition threshold limits are subject to the following rules:

- Dry value is the largest.
- Normal value is the smallest.
- Humid value must be larger than Very Light value but smaller than Dark value.

The Pressure Condition threshold limits are subject to the following rules:

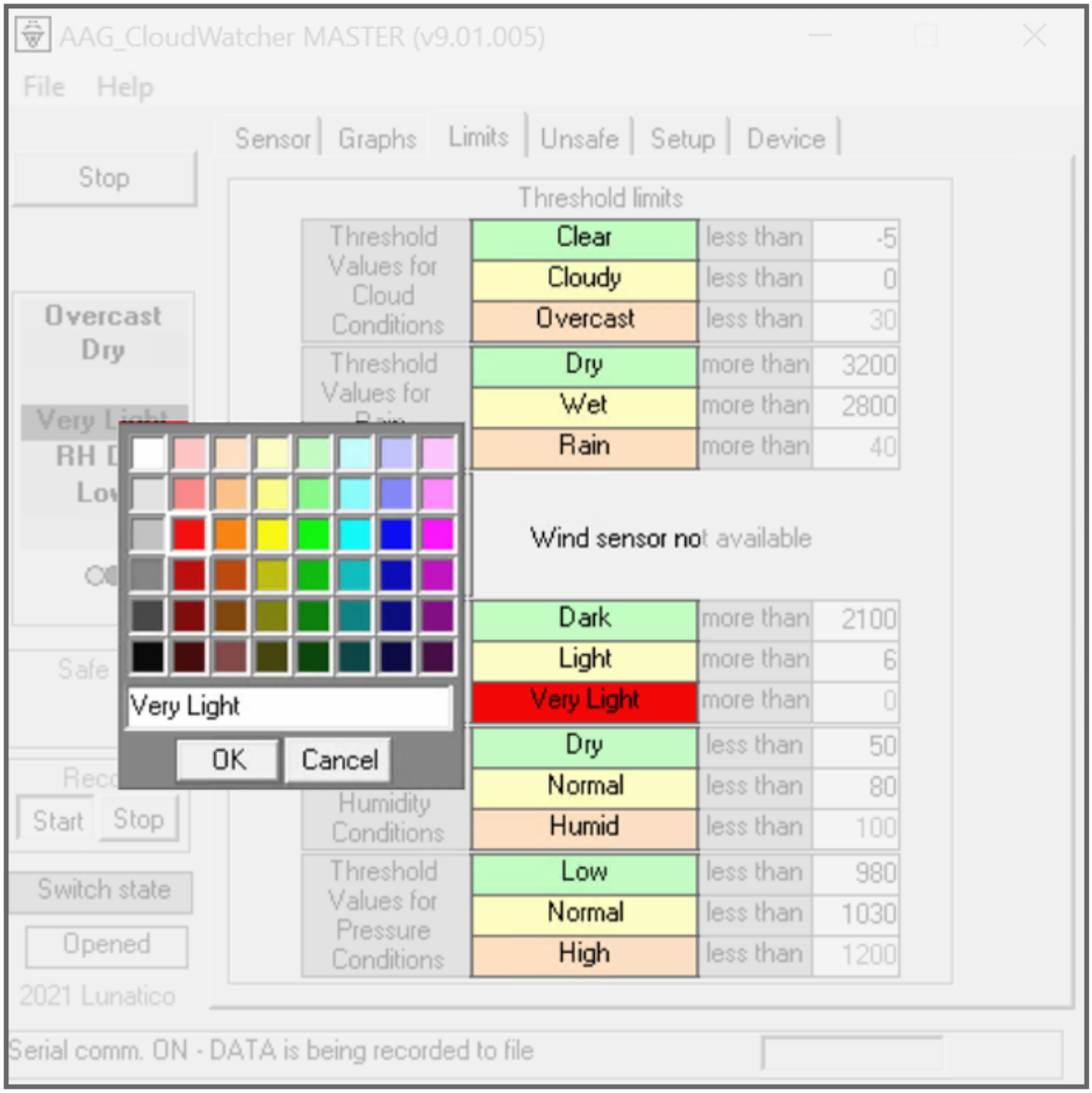
- Low value is the largest.
- Normal value is the smallest.
- High value must be larger than Very Light value but smaller than Dark value.

The system performs validations using these above rules.

The threshold values displayed on the screen are used by the software to determine the different sky conditions. After entering new values, these are immediately used in the next calculation cycle.

To avoid confusion, it is advisable to stop the program before changing any threshold values.

The colour associated with each condition, as well as its description, may be changed by right-clicking the mouse over the cell containing the condition description as in the figure below.



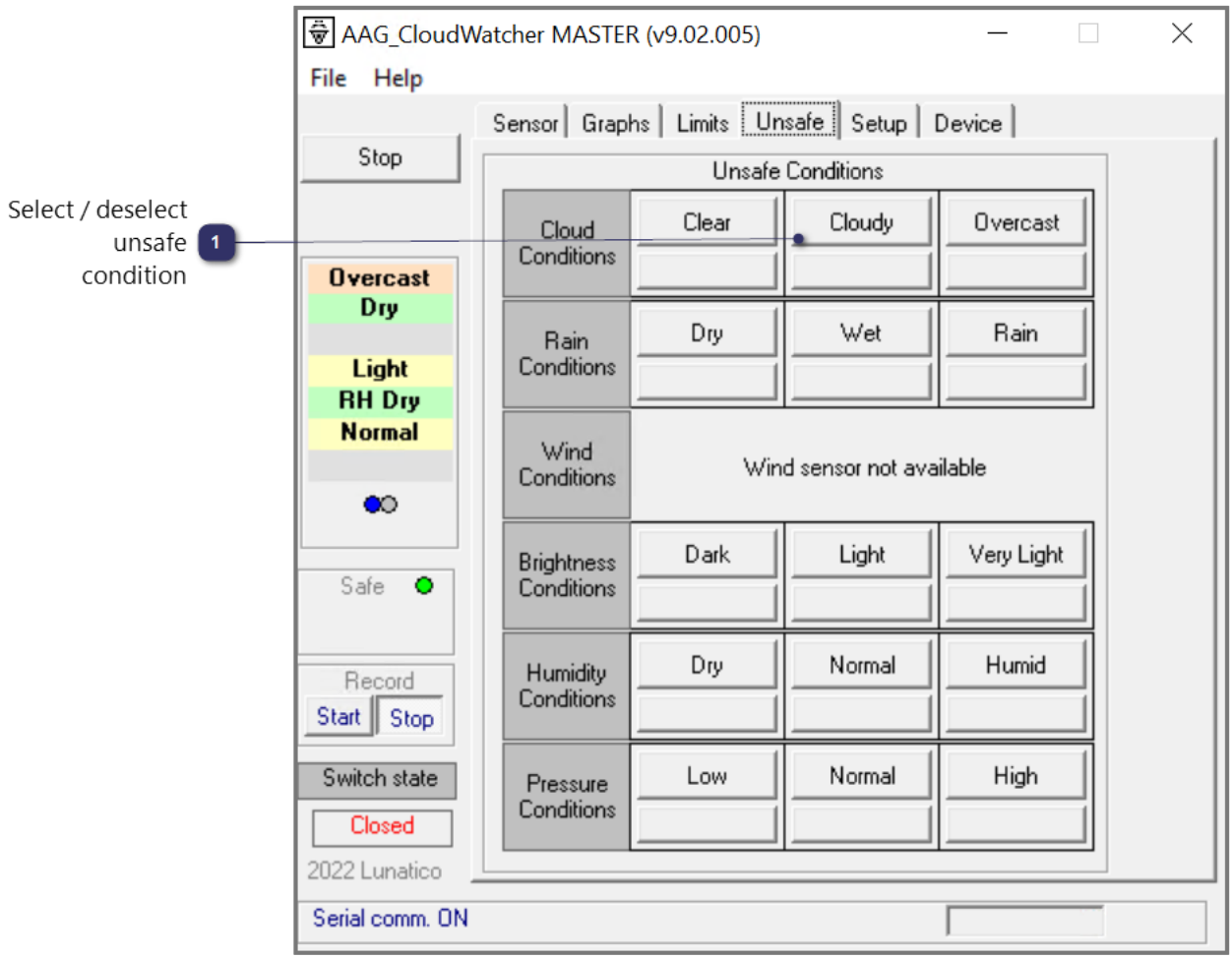
Colours and text changes affect:

- Sensor tab.
- Summary section of Common Controls and Indictations.
- Graphs.

Unsafe Tab

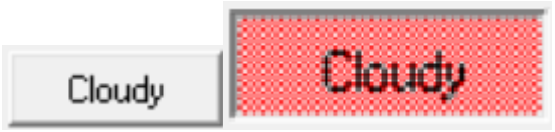
This tab allows the determination of Unsafe conditions, based on the criteria defined on the Limits tab.

For each condition there are two buttons. Selecting the top button instructs the software to treat that condition as unsafe.



1 Select / deselect unsafe condition

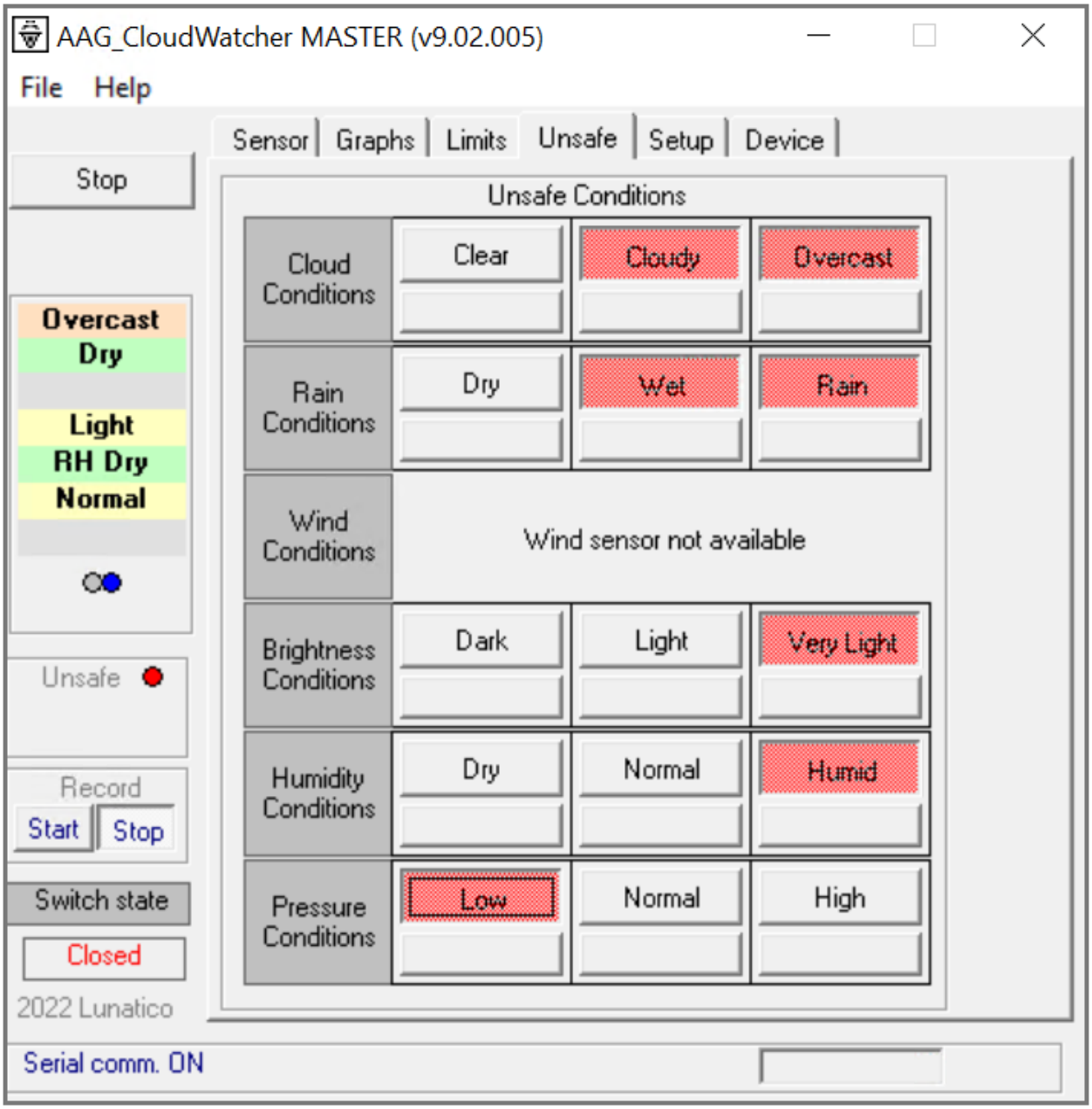
A selected unsafe condition appears in red.



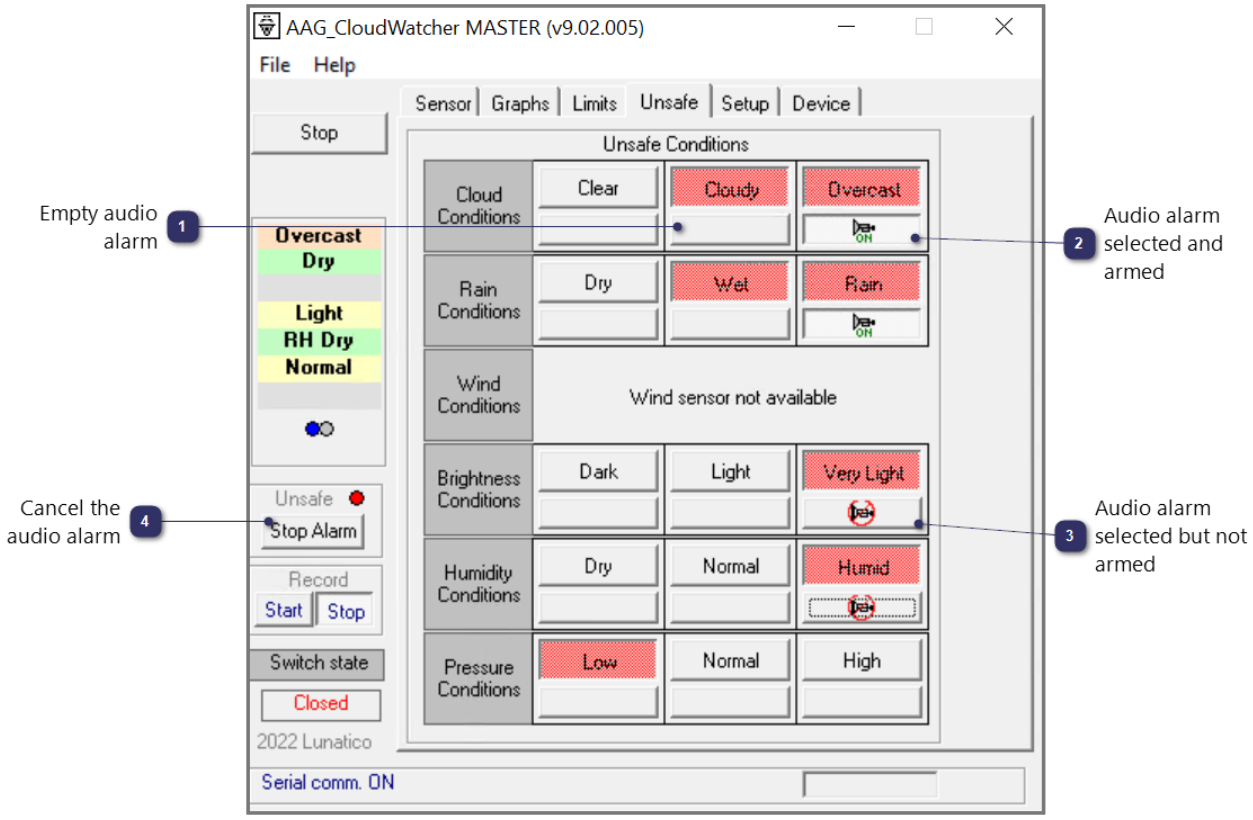
Note that when:

- Cloudy is selected Overcast is automatically selected.
- Wet is selected, Rain is automatically selected.
- Windy is selected, Very Windy is automatically selected.
- Light is selected, Very Light is automatically selected.
- Normal humidity is selected, Humid is automatically selected.
- Normal pressure is selected, High pressure is automatically selected.

A typical selection might appear as below:

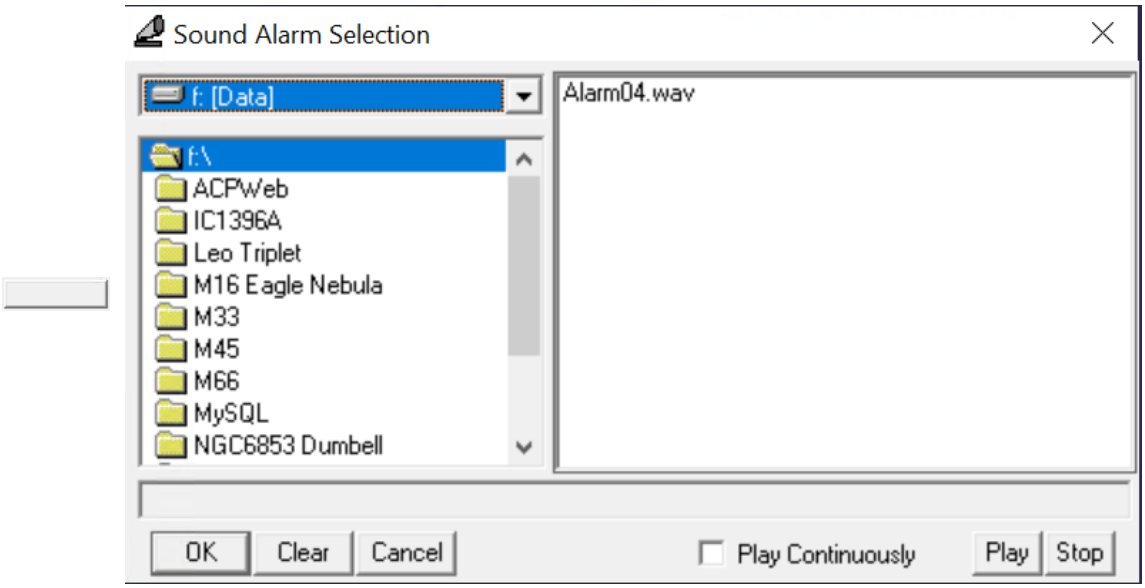


The lower buttons are used to select and audio alarms.



1 Empty audio alarm

Right-click to open audio alarm selection window.
Select 'Play Continuously', if required.
Select OK to confirm choice or clear to remove an existing alarm.
Each button can be programmed with a separate sound.



2 Audio alarm selected and armed

Left-click to arm the alarm.
To arm the sound, the associated Unsafe button must be armed (red).
Audio alarm will play when condition becomes unsafe.

3 Audio alarm selected but not armed

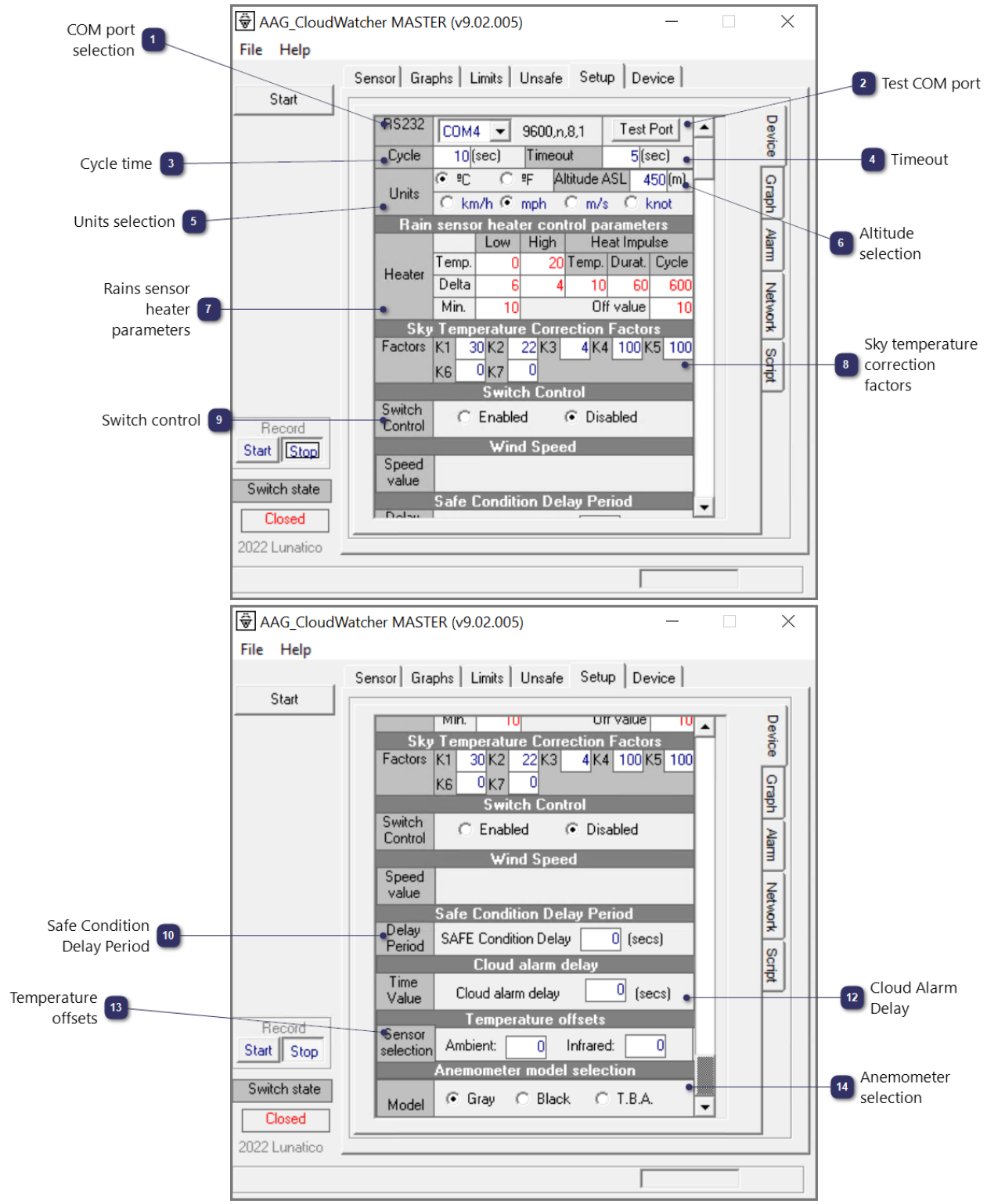
The symbol only appears on buttons when a sound has been selected (see 1. above).

4 Cancel the audio alarm

Left-click to cancel the alarm.

Settings Tab: Device Side Tab

Access to the Setup => Device tab parameters varies depending on the operational mode (Master or Remote) and whether the unit has been started or not. If a parameter is greyed out, check that the device is not running. The device is not running when the summary data in Common Controls and Indications section is not displayed and the Start / Stop button displays Start (see below).



1 COM port selection

The desired COM port can be selected from the drop down menu. Port settings are fixed, as displayed below.

2 Test COM port

Click this button to confirm that the correct COM port has been selected.

3 Cycle time

Master Mode: The time, in seconds, between readings of the CloudWatcher.
Remote Mode: The time, in seconds, between readings of the Parameters Transfer File.

4 Timeout

Master Mode: The time, in seconds, within which the CloudWatcher should update its readings.
Remote Mode: The time, in seconds, within which the Parameters Transfer File should be updated.

5 Units selection

Various combinations of temperature and wind units can be selected.

6 Altitude selection

Enter the altitude of the CloudWatcher in meters. The altitude is used to convert pressure readings to Mean Sea Level.

7 Rains sensor heater parameters

The *Rain Sensor Heater Parameters* are used by the program to adjust the power to the rain sensor heating element in order to avoid the formation of condensation on the rain sensor surface.

The field ***Temp*** of the ***Low*** column corresponds to the ambient temperature (ÂºC), below which the rain sensor operational temperature is kept at the value defined in the ***Delta*** field. For example, in the case illustrated in figure above, for ambient temperatures below 0ÂºC the rain sensor operational temperature will be kept at 6ÂºC.

The field ***Temp*** of the ***High*** column corresponds to the ambient temperature (ÂºC), above which the rain sensor operational temperature will be kept at the ambient temperature **+** the value defined in the ***Delta*** field. For example, in the case illustrated in figure above, for temperatures greater than 20ÂºC the rain sensor operational temperature will be adjusted to the ambient temperature **+** 4ÂºC.

For ambient temperatures comprised between ***Low Temp*** and ***High Temp*** values, the system will calculate an increment which is directly proportional to the ambient temperature and limited by the *Low Delta* and *High Delta* values. This increment is added to the ambient temperature to establish the rain sensor operational temperature.

The ***Min***. parameter defines the minimum heating level. This parameter corresponds to minimum level of heating at which the rain sensor heater is kept and it may create a situation where the rain sensor temperature is above the value calculated by the previous parameters.

The ***Heat Impulse*** mechanism has the objective to dry the rain sensor surface if the rain sensor stays continuously *Wet / Rain* for the period of time defined by the ***Cycle*** field (in seconds).

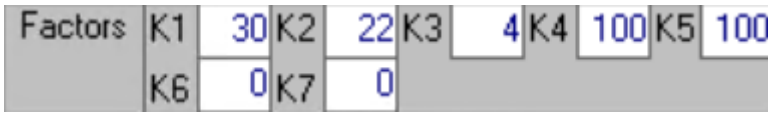
The duration of the pulse is given by the ***Durat*** field (in seconds) and the pulse intensity is calculated in terms of the rain sensor temperature as the value of the ambient temperature (ÂºC) + the ***Temp*** field (ÂºC).

Setting any of the fields *Durat*, *Temp*, *Cycle* to zero disables this mechanism. It is recommended to use values smaller than 15.

		Low	High	Heat Impulse		
Heater	Temp.	0	20	Temp.	Durat.	Cycle
	Delta	6	4	10	60	600
	Min.	10		Off value		10

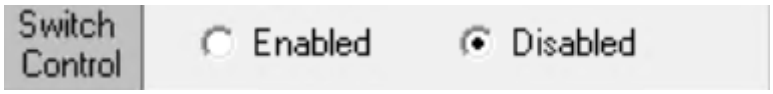
8 Sky temperature correction factors

The values of the constants K1 to K7 can be set / amended here. For the full meaning of these constants, please refer to Appendix 1.



9 Switch control

Enable or disable the hardware switch.
Note: The switch retains its state when disabled. This can lead to an undesirable mismatch between the reported safe / unsafe state and the state of the switch. Excercise caution when disabling the switch.

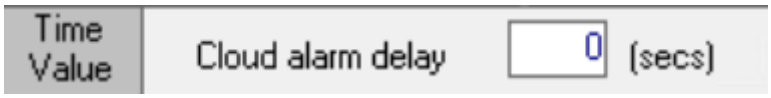


10 Safe Condition Delay Period

When a condition changes from unsafe to safe, the software will wait for this time, in seconds, before changing its state to safe.
The purpose of the delay is to ensure that the weather has definitely improved before allowing the roof to be opened and an obervation to be continued.

12 Cloud Alarm Delay

When cloud is sensed, the software will wait for this time, in seconds, before registering unsafe.
The purpose of the delay is to avoid a small amount of cloud causing a session to abort.
Note: The Cloud Alarm Delay for Unsafe signalling is activated only with a change from *Clear* to *Cloud* conditions. No delay occurs with a change from *Clear* to *Overcast* conditions and Unsafe is triggered immediately.

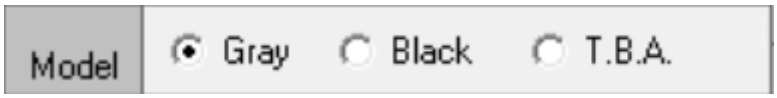


13 Temperature offsets

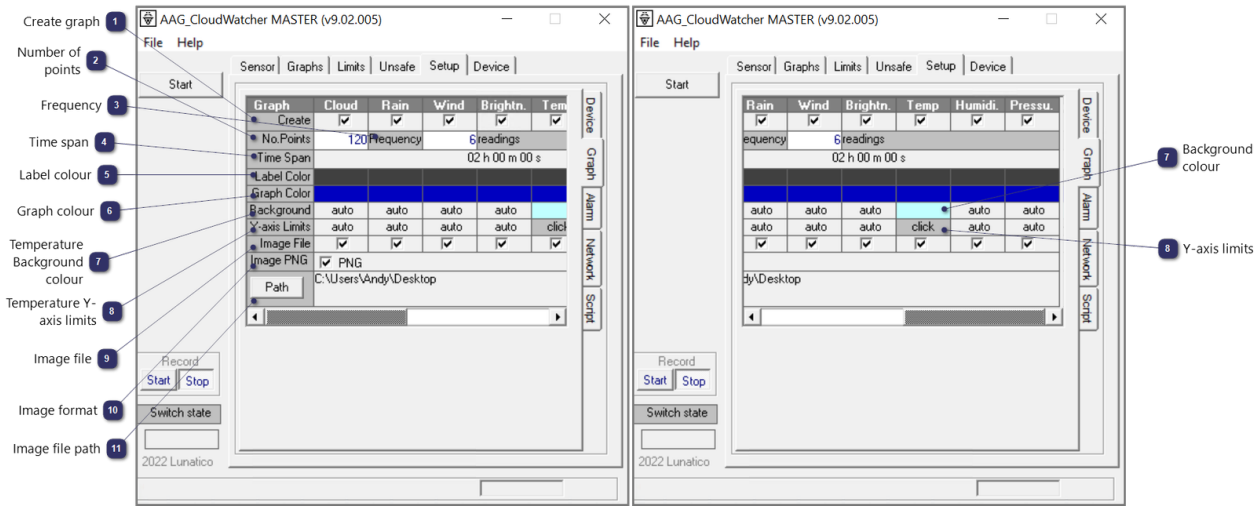
Depending on the location of the CloudWatcher, ambient temperature can differ from the IR temperature by several degrees. It is possible to offset either value to meet the user's specific requirements.

14 Anemometer selection

There are currently two anemometers in service, which differ in colour. The correct model can be selected here.



Settings Tab: Graphs Side Tab



1 Create graph

Checked boxes cause the associated graphs to appear under the *Graphs* tab.
Unchecking this box, automatically deselects the associated image file.

2 Number of points

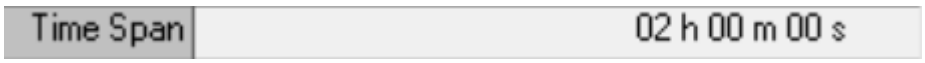
Enter the desired number of datapoints here. (See Time Span)

3 Frequency

The frequency determines when a point is created. In the example, one point is created every 6 *Cycles* (see Setup => Device tab and Time span).

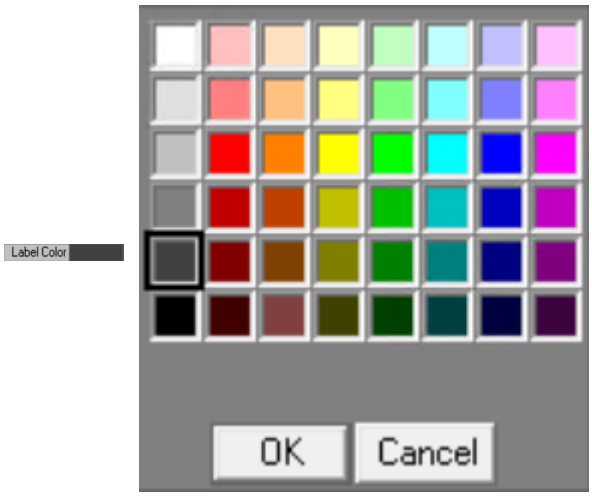
4 Time span

The time derived from the formula: *Number of points* x *Frequency* x *Cycle* (See Setup => Device tab).
The time span forms the x-axis of the graph.



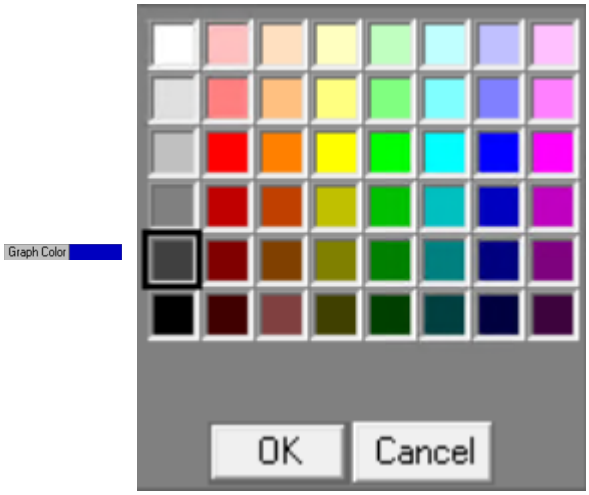
5 Label colour

Right-click to access the colour selector panel.



6 Graph colour

Right-click to access the colour selector panel.



7 Background colour

With the exception of the Temperature graph, the background colour is selected according to the selections made on the Limits tab.

7 Temperature Background colour

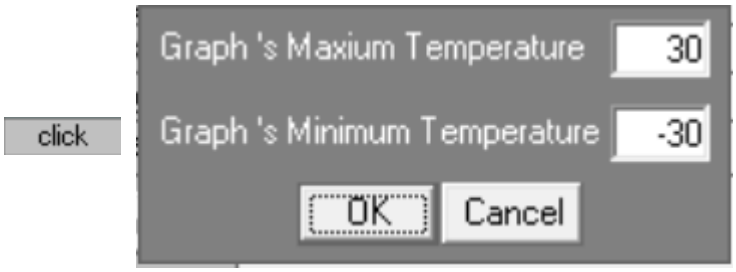
Write description here...

8 Y-axis limits

With the exception of the Temperature graph, the Y-axis limits are fixed.

8 Temperature Y-axis limits

Left click to adjust the Temperature graph Y-axis to the desired range.



9 Image file

Images files will be created for all graphs where this box is checked.
The checkbox is only available if the associated *Create graph* check box is selected.

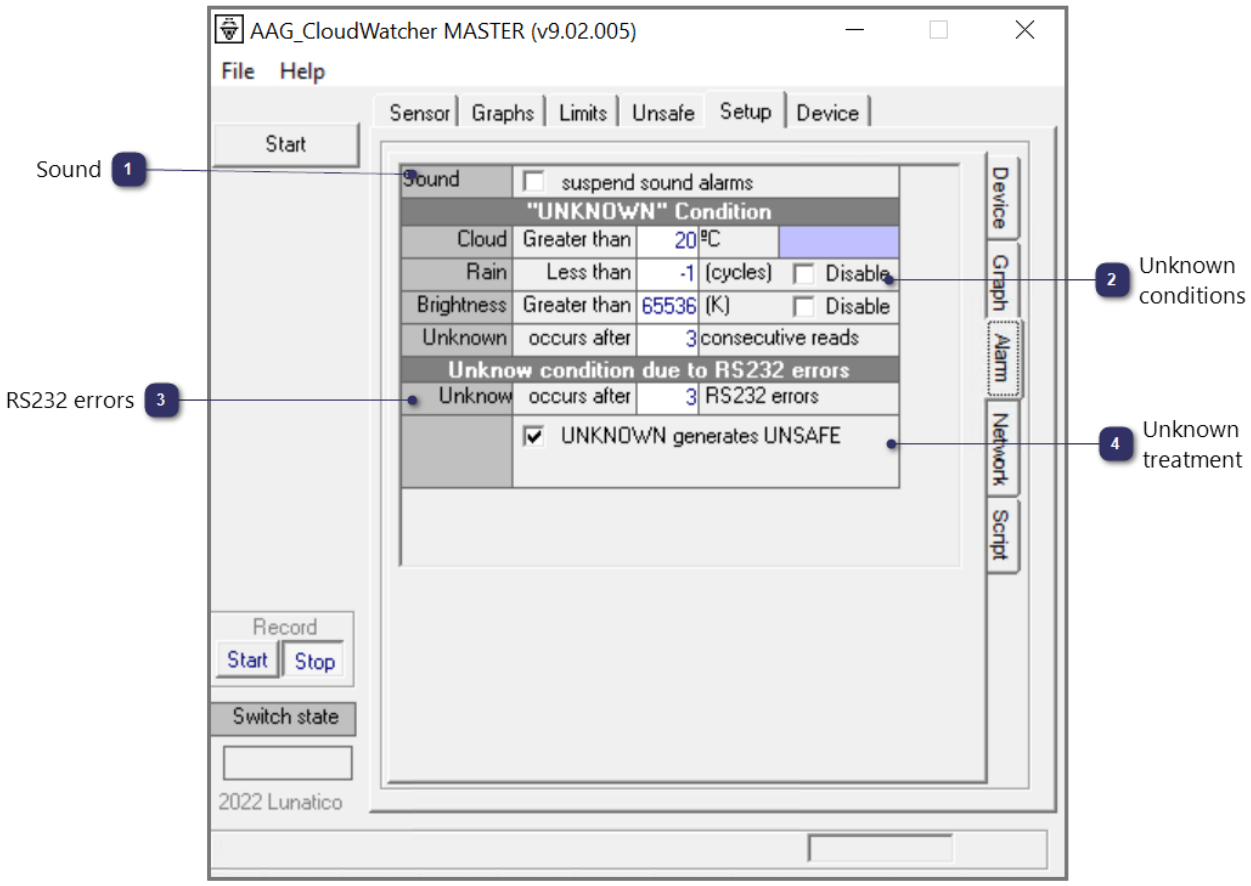
10 Image format

Graph images can be produced in .PNG or .JPEG formats. Deselect the PNG checkbox to select JPEG.
With JPEG selected, further options are available to select the desired quality.

11 Image file path

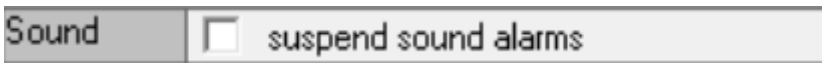
Select a writeable location for the graph image files.

Settings Tab: Alarm Side Tab



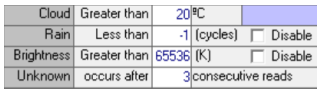
1 Sound

Unsafe sound alarms (see *Unsafe* tab) can be suspended here.
Note: There is no indication of this setting on the Unsafe tab, the alarm will simply fail to play.



2 Unknown conditions

Values to be considered as unknown, i.e. unreliable, can be specified here.
Unknown conditions appear in the colour indicated in the upper right-hand panel. Right-click to change the colour.
Rain and brightness unknown conditions can be disabled.
The number of cycles required for a condition to be considered unknown can be adjusted.

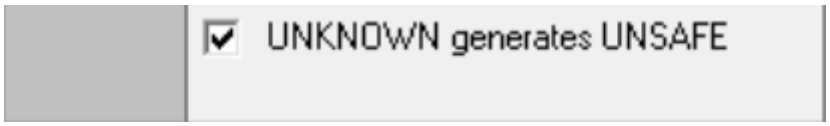


3 RS232 errors

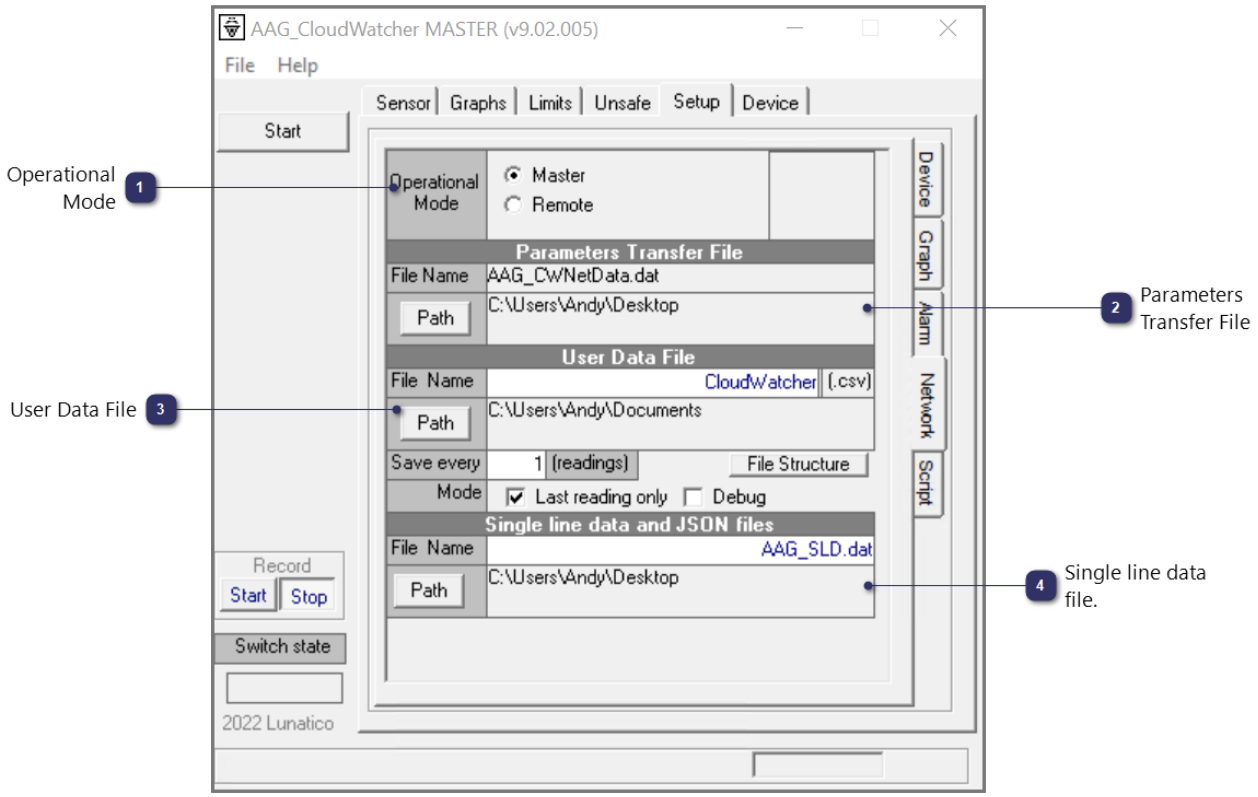
There is a separate section to determine when the serial connection is considered to be unknown.

4 Unknown treatment

An unknown state is considered as unsafe, when this option is selected.

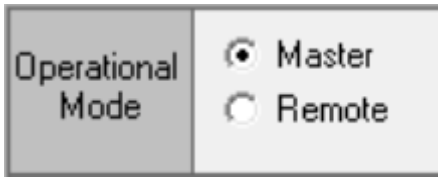


Settings Tab: Network Side Tab



1 Operational Mode

Toggle between Master and Remote Modes.

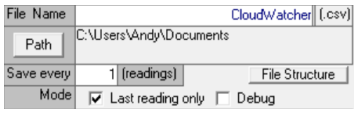


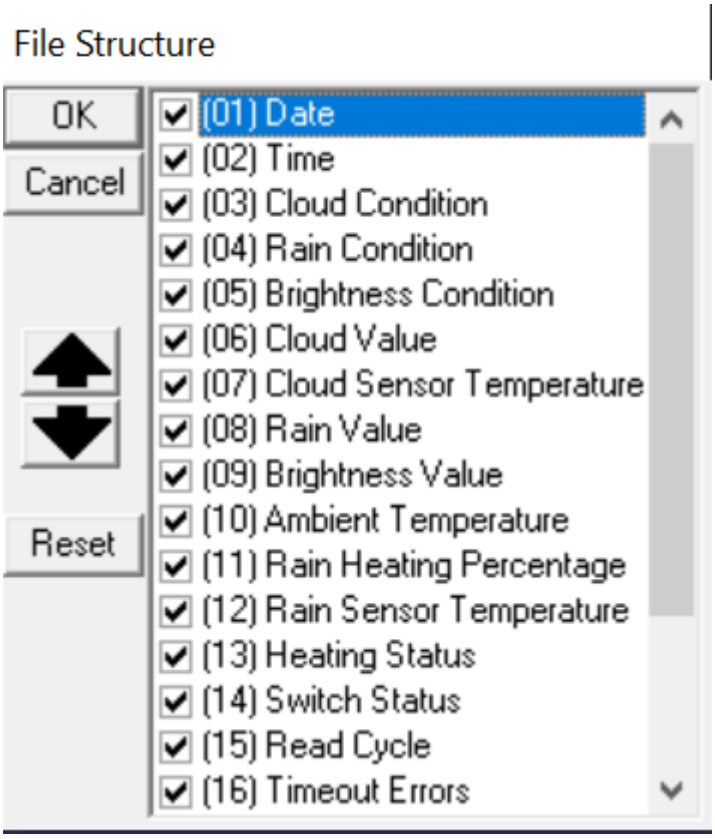
2 Parameters Transfer File

The Parameters Transfer Data file is required if any Remote Mode installations are to be installed on the network.
Only software running in Master mode can the Parameters Transfer File.
Left-click on 'Path' to set the output directory.
Note: The default file path is the installation directory. Recent Windows versions do not permit files to be written to this directory and therefore an alternative, writeable location should be used.

3 User Data File

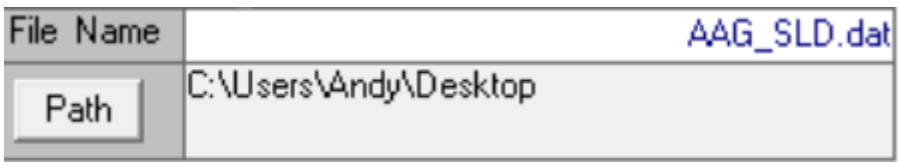
The file is optional and can be written by software running in either Master or Remote modes.
The file is is .csv format. Left-click on the current file name, in blue, to amend it.
Left-click on 'Path' to set the output directory.
Note: The default file path is the installation directory. Recent Windows versions do not permit files to be written to this directory and therefore an alternative, writeable location should be used.
By default, the file updates on every reading. This interval can be set to any integer value.
Left click on 'File Structure' to open a menu that allows the content and order of file to be determined.
Select 'Last reading only' and / or 'Debug' as required.
Caution: Choosing Debug can result in large file sizes.





4 Single line data file.

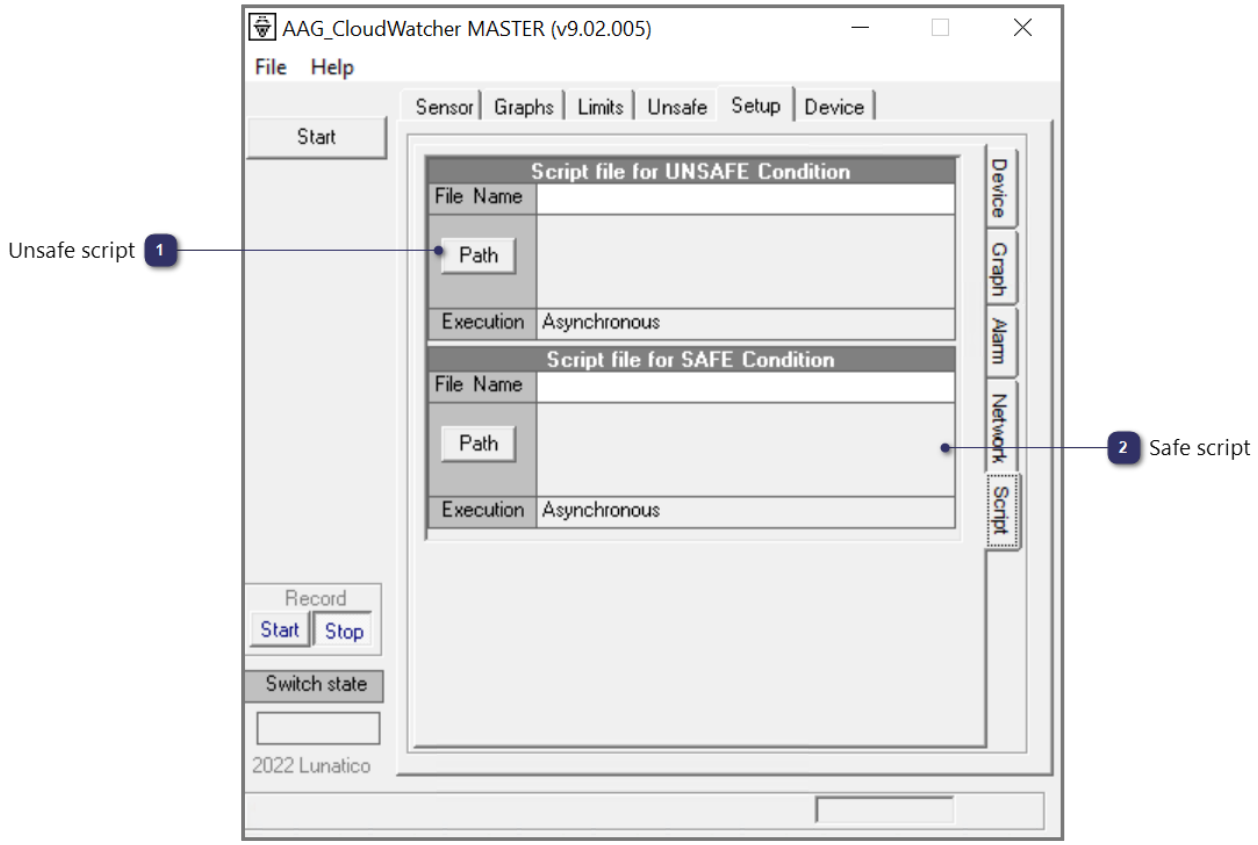
The Single Line Data (SLD) file and new JSON format files are written to the path specified in this section.
The name and extension of the SLD file may be freely chosen by left-clicking on the current file name, in blue.
The name of the JSON format file, *aag_json.dat*, is fixed. The file is required by the Lunatico ASCOM Safety Switch.
Left-click on 'Path' to select the file path.
Note: The default file path is the installation directory. Recent Windows versions do not permit files to be written to this directory and therefore an alternative, writeable location should be used.



Settings Tab: Scripts Side Tab

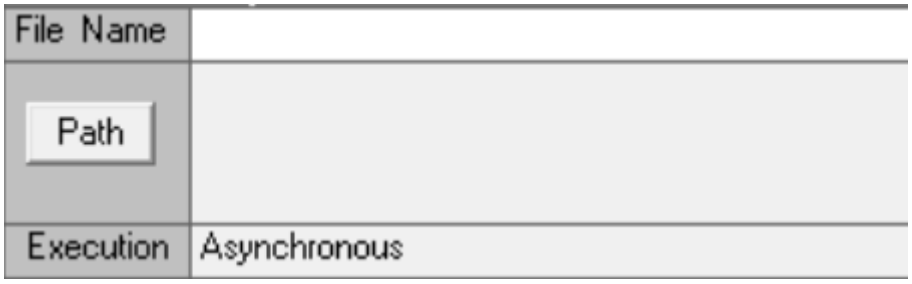
The CloudWatcher software has the ability to execute scripts whenever the state changes from safe to unsafe or from unsafe to safe.

Please contact us or visit our forum if you require and assistance in writing scripts for your CloudWatcher.



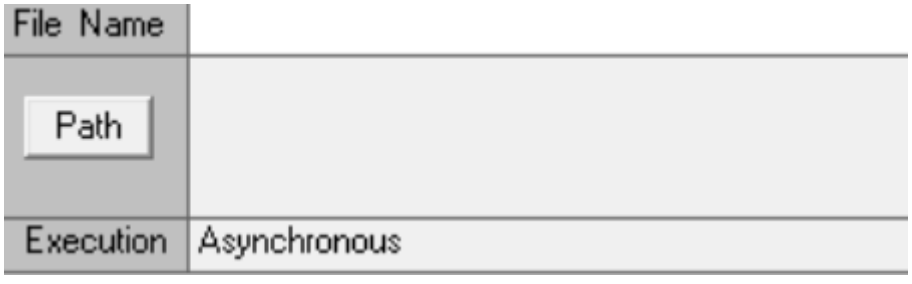
1 Unsafe script

Left-click on 'Path' to choose the location of a script that should run when the CloudWatcher reports unsafe.



2 Safe script

Left-click on 'Path' to choose the location of a script that should run when the CloudWatcher reports safe.



Working With Other Software

The Lunatico ASCOM Driver

The question of safety monitoring can be confusing as imaging software can deal with the issue in two ways:

If weather information is supplied to the software then there is often an option to define the parameters for safe operation within the software itself. This may conflict with limits defined within the CloudWatcher.

Increasingly developers are adding support for a simple 'switch' that allows an external product such as the CloudWatcher to decide on the parameters and send a single safe / unsafe command to the imaging software using the ASCOM Safety Switch standard.

The CloudWatcher can be used in either of the above scenarios and how you choose to configure the system is a matter of personal choice and the capabilities of your software package.

If you use the ASCOM Safety Switch approach, then it is important to note that the responsibility for dealing with the unsafe condition lies with your imaging software. Normally, the minimum actions would be to park the mount and close the roof. Some packages go beyond this and will attempt to restart a session if the weather improves and the switch reports safe.

You may now be asking why the CloudWatcher offers the opportunity to launch a script when the weather becomes unsafe / safe? The reason is in order to support users whose imaging software does not provide protection features. A script can stop the imaging software, park the mount and close the observatory roof in order to protect your equipment. In theory, you could also restart the session if the weather improves, although this is somewhat harder to achieve in practice.

If your software supports observatory safety protection features then we recommend that you use these rather than relying on the CloudWatcher scripts. If you would like further peace of mind then you may care to install the 'Lunatico Last Line Of Defence' software package which, in addition to monitoring the weather, will secure the observatory in the case your imaging software crashes or the internet connection is lost.

Consult your software's doumentation for details of how to configure the safety driver.

CCDAutoPilot 3

The software includes two files: *AAG_ClarityIEmulator.dll* and *AAG_ClarityIIEmulator.dll* which emulate the inter-program protocols of the Boltwood Cloud Sensor version I and version II.

To use AAG_CloudWatcher with CCDAutoPilot3, one should select either *CloudSensorI* or *CloudSensorII* in the *Settings* section under *Cloud Sensor* field of CCDAutoPilot3 window.

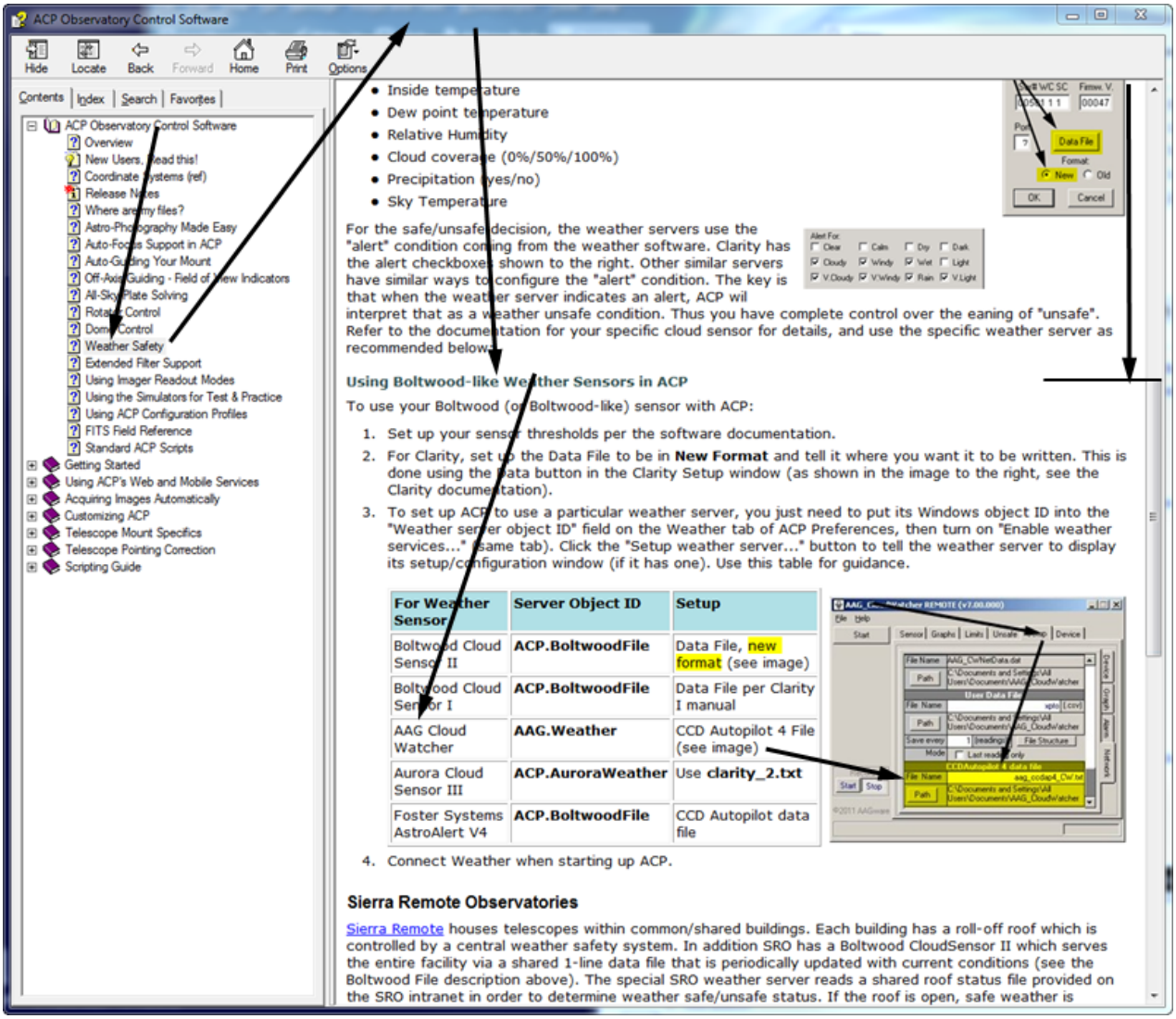
CCDAutoPilot3 only requests *Cloud* conditions and *Rain* conditions, therefore any of the CCDAutoPilot3 options available for Cloud Sensor will work with the CloudWatcher.

The software creates a data file that can be used by CCDAutopilot 4 and 5 (see Setup tab => Network section).

Since version 8.0 of the software, this file has been renamed as aag_sld.dat (from AAG Single Line Data). The contents are still compatible with CCDAP.

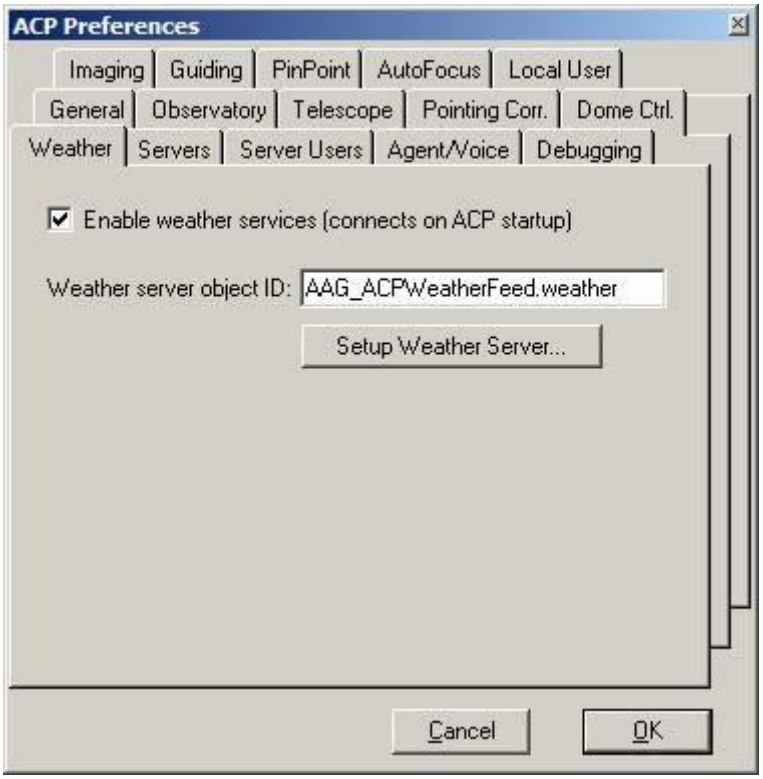
ACP

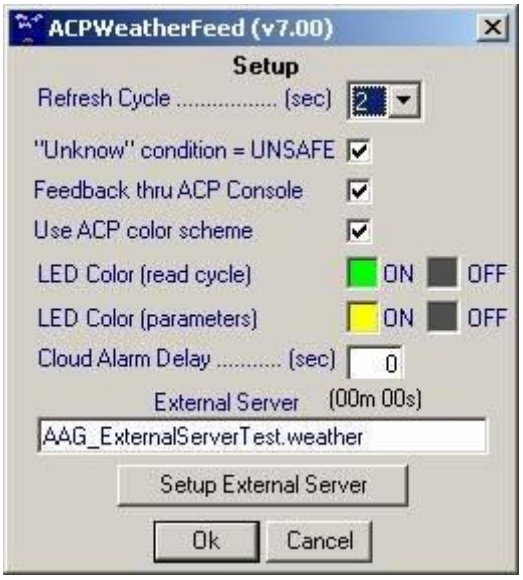
ACP's extensive help file includes specific information on how to get weather data from the CloudWatcher.

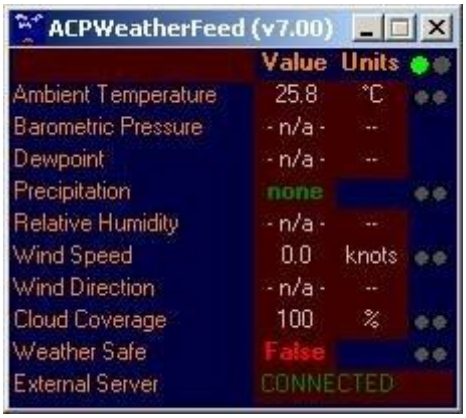


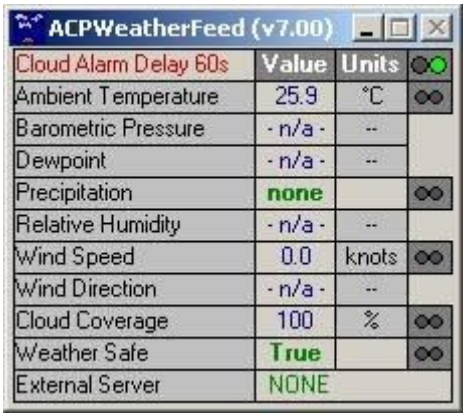
The CCDAutopilot 4 data file is now (versions 8.0+) called the Boltwood compatible single line data file and is by default named aag_sld.dat.

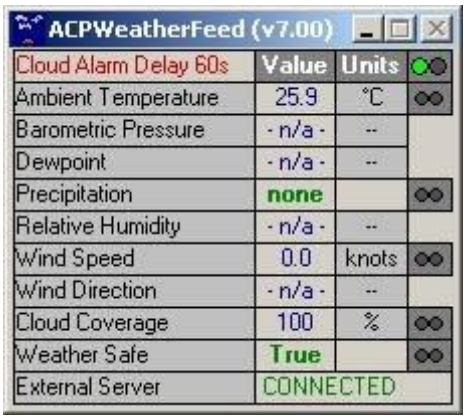
Ensure that the correct folder path is selected, in order that the program can update the file.

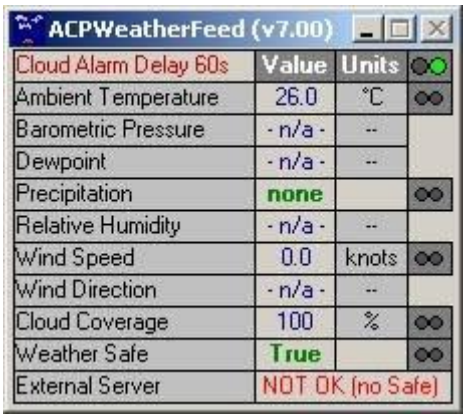

ACP Preferences
Imaging
Guiding
PinPoint
AutoFocus
Local User
General
Observatory
Telescope
Pointing Corr.
Dome Ctrl.
Weather
Servers
Server Users
Agent/Voice
Debugging
Enable weather services (connects on ACP startup)
Weather server object ID:
AAG_ACPWeatherFeed.weather
Setup Weather Server...
Cancel
OK

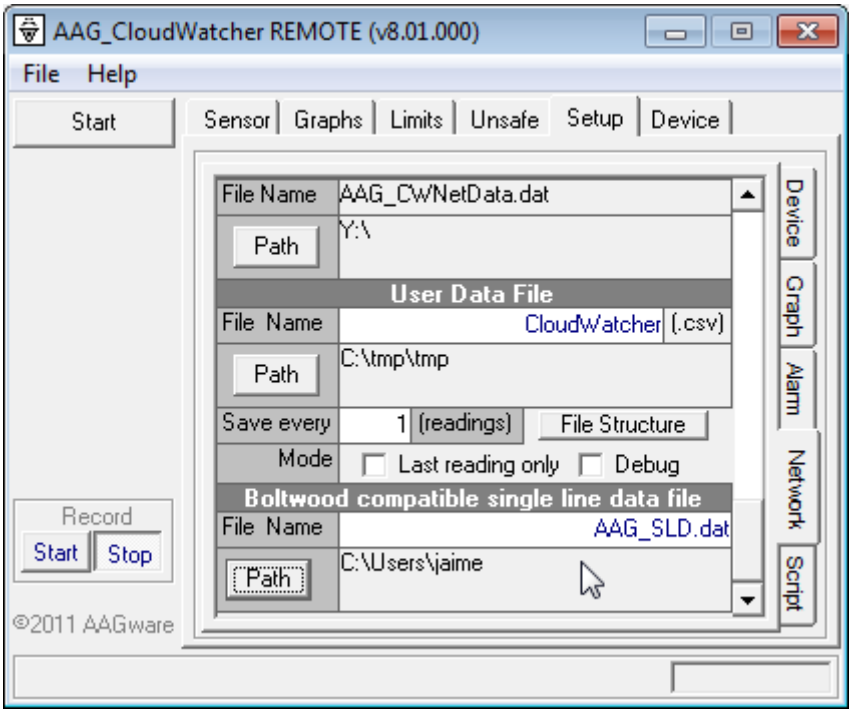

ACPWeatherFeed (v7.00)
Setup
Refresh Cycle ..... (sec)
2
"Unknow" condition = UNSAFE
Feedback thru ACP Console
Use ACP color scheme
LED Color (read cycle)
ON
OFF
LED Color (parameters)
ON
OFF
Cloud Alarm Delay ..... (sec)
0
External Server
(00m 00s)
AAG_ExternalServerTest.weather
Setup External Server
Ok
Cancel


ACPWeatherFeed (v7.00)
Value
Units
Ambient Temperature
25.8
°C
Barometric Pressure
- n/a -
--
Dewpoint
- n/a -
--
Precipitation
none
Relative Humidity
- n/a -
--
Wind Speed
0.0
knots
Wind Direction
- n/a -
--
Cloud Coverage
100
%
Weather Safe
False
External Server
CONNECTED


ACPWeatherFeed (v7.00)
Cloud Alarm Delay 60s
Value
Units
Ambient Temperature
25.9
°C
Barometric Pressure
- n/a -
--
Dewpoint
- n/a -
--
Precipitation
none
Relative Humidity
- n/a -
--
Wind Speed
0.0
knots
Wind Direction
- n/a -
--
Cloud Coverage
100
%
Weather Safe
True
External Server
NONE


ACPWeatherFeed (v7.00)
Cloud Alarm Delay 60s
Value
Units
Ambient Temperature
25.9
°C
Barometric Pressure
- n/a -
--
Dewpoint
- n/a -
--
Precipitation
none
Relative Humidity
- n/a -
--
Wind Speed
0.0
knots
Wind Direction
- n/a -
--
Cloud Coverage
100
%
Weather Safe
True
External Server
CONNECTED


ACPWeatherFeed (v7.00)
Cloud Alarm Delay 60s
Value
Units
Ambient Temperature
26.0
°C
Barometric Pressure
- n/a -
--
Dewpoint
- n/a -
--
Precipitation
none
Relative Humidity
- n/a -
--
Wind Speed
0.0
knots
Wind Direction
- n/a -
--
Cloud Coverage
100
%
Weather Safe
True
External Server
NOT OK (no Safe)



CCD Commander

CCDCommander includes specific setup configuration for connecting to the CloudWatcher software.

Inter-Program Communication Protocol

These are the public properties / methods of class CloudWatcher available for inter-program communications.

```
'---------- Enumerators ----------------------------------------------------------
Enum CloudCond
  CloudUnknown = 0
  CloudClear = 1
  CloudCloudy = 2
  CloudVeryCloudy = 3
End Enu
Enum BrightnessCond
  BrightnessUnknown = 0
  BrightnessDark = 1
  BrightnessLight = 2
```

```
  BrightnessVerylight = 3
End Enum
Enum RainCond
  RainUnknown = 0
  RainDry = 1
  RainWet = 2
  RainRain = 3
End Enum
Enum WindCond
  WindUnknown = 0
  WindCalm = 1
  WindWindy = 2
  WindVeryWindy = 3
End Enum
Enum SkyCond
  skyUnknown = 0
  skyClear = 1
  skyCloudy = 2
  skyVeryCloudy = 3
  skyWet = 4
End Enum
'------------ Graph Data Array methods
--------------------------------------------------------------------------------

The graph arrays are 2 dimension arrays where 1st dimension contains time and
2nd dimension contains the data values. The arrays contain data from element 1 to upper
bound limit given by UBound() function. NB: Element 0 is not used.

The graph data starts at the element given by
AAG_GraphArrayFirstElem(graphNo) method and it ends at the element  given by
AAG_GraphArrayLastElem( graphNo) method.

The data arrays wraps around the UBound() element to element 1.  NB: Element 0 is not
used.

Public Function AAG_GraphArrayFirstElem(ByVal GraphNo As Long) As Long

        GraphNo = 1 for Cloud temperature data, 2 for Rain sensor data, 3 for Light sensor
        data & 4 for ambient temperature data;

        Returns the first element of the graph data array;

End Function

Public Function AAG_GraphArrayLastElem(ByVal GraphNo As Long) As Long

        GraphNo = 1 for Cloud temperature data, 2 for Rain sensor data, 3 for Light sensor
        data & 4 for ambient temperature data;

        Returns the last element of the graph data array;

End Function

Public Function AAG_GraphArrayCloud() As Variant

        Returns a variant containing a 2 dimensional array where the 1st dimension
        corresponds to the time and 2nd dimension contains the sky temperature value;

End Function
```

Public Function AAG_GraphArrayRain() As Variant

Returns a variant containing a 2 dimensional array where the 1st dimension corresponds to the time and 2nd dimension contains the rain sensor value;

End Function

Public Function AAG_GraphArrayDay() As Variant

Returns a variant containing a 2 dimensional array where the 1st dimension corresponds to the time and 2nd dimension contains the light sensor value;

End Function

Public Function AAG_GraphArrayTemp() As Variant

Returns a variant containing a 2 dimensional array where the 1st dimension corresponds to the time and 2nd dimension contains the ambient temperature value.

End Function

'---------- Main methods to get device data

Public Function AmbientTemperature() As Double

Public Function Condition_Cloud() As CloudCond

Public Function Condition_Brightness() As BrightnessCond

Public Function Condition_Rain() As RainCond

Public Function Condition_Wind() As WindCond

Return WindCalm

Public Function Condition_Sky() As SkyCond

Public Function Condition_WetF() As Boolean

Return True if Wet or Rain

Public Function DataReady() As Boolean

True – if cycle reading data from the device has finished;

False – if it is still reading data from the device;

Public Function BrightnessValue() As Double

Public Function DewPointTemp() As Double

Returns always zero

Public Function HumidityRelative() As Double

Returns always 10

Public Function RainHeaterPercent() As Double

Returns decimal values e.g. 0.1 for 10% relative humidity

Public Function RainHeaterStatus() As String

Public Function RainHeaterTemperature() As Double

Public Function RainValue() As Double

Public Function Safe() As Boolean

True - Conditions are Safe

False - Conditions are Unsafe

Public Function SecondsSinceGoodData() As Long

Public Function SkySensorTemperature() As Double

Public Function SkyTemperature() As Double

```
Public Function SwitchStatus() As Long
        Returns 1= switch open
                2= switch closed
Public Function WindValue() As Double
         Returns always zero
Public Function WindDir() As Double
         Returns always zero
'--------- Limits TAB ----------------------------------------------------------------------
‘Properties to get / set values in Limits TAB
Public Property Get ThreshBrightnessDark() As Double
Public Property Let ThreshBrightnessDark(ByVal vData As Double)
Public Property Get ThreshBrightnessLigth() As Double
Public Property Let ThreshBrightnessLigth(ByVal vData As Double)
Public Property Get ThreshBrightnessVLigth() As Double
Public Property Let ThreshBrightnessVLigth(ByVal vData As Double)
Public Property Get BrightnessUnknowConditionValue() As Long
Public Property Let BrightnessUnknowConditionValue(ByVal vData As Long)
Public Property Get ThreshRainDry() As Double
Public Property Let ThreshRainDry(ByVal vData As Double)
Public Property Get ThreshRainWet() As Double
Public Property Let ThreshRainWet(ByVal vData As Double)
Public Property Get ThreshRainRain() As Double
Public Property Let ThreshRainRain(ByVal vData As Double)
Public Property Get RainUnknowConditionValue() As Long
Public Property Let RainUnknowConditionValue(ByVal vData As Long)
Public Property Get ThreshTempClear() As Double
Public Property Let ThreshTempClear(ByVal vData As Double)
Public Property Get ThreshTempCloudy() As Double
Public Property Let ThreshTempCloudy(ByVal vData As Double)
Public Property Get ThreshTempVCloudy() As Double
Public Property Let ThreshTempVCloudy(ByVal vData As Double)
Public Property Get ThreshWindCalm() As Double
Public Property Let ThreshWindCalm(ByVal vData As Double)
Public Property Get ThreshWindWindy() As Double
Public Property Let ThreshWindWindy(ByVal vData As Double)
Public Property Get ThreshWindVWindy() As Double
Public Property Let ThreshWindVWindy(ByVal vData As Double)
Public Property Get CloudUnknowConditionValue() As Long
Public Property Let CloudUnknowConditionValue(ByVal vData As Long)
Public Property Get ColorClear() As Long
Public Property Let ColorClear(ByVal vData As Long)
Public Property Get ColorCloudy() As Long
Public Property Let ColorCloudy(ByVal vData As Long)
Public Property Get ColorVCloudy() As Long
```

```
Public Property Let ColorVCloudy(ByVal vData As Long)
Public Property Get DescriptionClear() As String
Public Property Let DescriptionClear(ByVal vData As String)
Public Property Get DescriptionCloudy() As String
Public Property Let DescriptionCloudy(ByVal vData As String)
Public Property Get DescriptionVCloudy() As String
Public Property Let DescriptionVCloudy(ByVal vData As String)
Public Property Get ColorDry() As Long
Public Property Let ColorDry(ByVal vData As Long)
Public Property Get ColorWet() As Long
Public Property Let ColorWet(ByVal vData As Long)
Public Property Get ColorRain() As Long
Public Property Let ColorRain(ByVal vData As Long)
Public Property Get DescriptionDry() As String
Public Property Let DescriptionDry(ByVal vData As String)
Public Property Get DescriptionWet() As String
Public Property Let DescriptionWet(ByVal vData As String)
Public Property Get DescriptionRain() As String
Public Property Let DescriptionRain(ByVal vData As String)
Public Property Get ColorDark() As Long
Public Property Let ColorDark(ByVal vData As Long)
Public Property Get ColorLight() As Long
Public Property Let ColorLight(ByVal vData As Long)
Public Property Get ColorVLight() As Long
Public Property Let ColorVLight(ByVal vData As Long)
Public Property Get DescriptionDark() As String
Public Property Let DescriptionDark(ByVal vData As String)
Public Property Get DescriptionLight() As String
Public Property Let DescriptionLight(ByVal vData As String)
Public Property Get DescriptionVLight() As String
Public Property Let DescriptionVLight(ByVal vData As String)
'--------- Unsafe TAB ------------------------------------------------------------------------
‘Properties to get / set conditions in Unsafe TAB
Public Property Get SoundClear() As String
Public Property Let SoundClear(vData As String)
Public Property Get SoundCloudy() As String
Public Property Let SoundCloudy(vData As String)
Public Property Get SoundOvercast() As String
Public Property Let SoundOvercast(vData As String)
Public Property Get SafeClear() As Long
Public Property Let SafeClear(vData As Long)
Public Property Get SafeCloudy() As Long
Public Property Let SafeCloudy(vData As Long)
Public Property Get SafeOvercast() As Long
```

```
Public Property Let SafeOvercast(vData As Long)
Public Property Get AlarmClear() As Long
Public Property Let AlarmClear(vData As Long)
Public Property Get AlarmCloudy() As Long
Public Property Let AlarmCloudy(vData As Long)
Public Property Get AlarmOvercast() As Long
Public Property Let AlarmOvercast(vData As Long)
'----
Public Property Get SoundDry() As String
Public Property Let SoundDry(vData As String)
Public Property Get SoundWet() As String
Public Property Let SoundWet(vData As String)
Public Property Get SoundRain() As String
Public Property Let SoundRain(vData As String)
Public Property Get SafeDry() As Long
Public Property Let SafeDry(vData As Long)
Public Property Get SafeWet() As Long
Public Property Let SafeWet(vData As Long)
Public Property Get SafeRain() As Long
Public Property Let SafeRain(vData As Long)
Public Property Get AlarmDry() As Long
Public Property Let AlarmDry(vData As Long)
Public Property Get AlarmWet() As Long
Public Property Let AlarmWet(vData As Long)
Public Property Get AlarmRain() As Long
Public Property Let AlarmRain(vData As Long)
'----
Public Property Get SoundDark() As String
Public Property Let SoundDark(vData As String)
Public Property Get SoundLight() As String
Public Property Let SoundLight(vData As String)
Public Property Get SoundVLight() As String
Public Property Let SoundVLight(vData As String)
Public Property Get SafeDark() As Long
Public Property Let SafeDark(vData As Long)
Public Property Get SafeLight() As Long
Public Property Let SafeLight(vData As Long)
Public Property Get SafeVLight() As Long
Public Property Let SafeVLight(vData As Long)
Public Property Get AlarmDark() As Long
Public Property Let AlarmDark(vData As Long)
Public Property Get AlarmLight() As Long
Public Property Let AlarmLight(vData As Long)
Public Property Get AlarmVLight() As Long
```

```
Public Property Let AlarmVLight(vData As Long)
'---------- Setup TAB --------------------------------------------------------------------
‘Properties to get / set values in Device section of Setup TAB
Public Property Get ReadCycle() As Long
Public Property Let ReadCycle(ByVal vData As Long)
Public Property Get HeaterTempLow() As Long
Public Property Let HeaterTempLow(ByVal vData As Long)
Public Property Get HeaterTempHigh() As Long
Public Property Let HeaterTempHigh(ByVal vData As Long)
Public Property Get HeaterDeltaLow() As Long
Public Property Let HeaterDeltaLow(ByVal vData As Long)
Public Property Get HeaterDeltaHigh() As Long
Public Property Let HeaterDeltaHigh(ByVal vData As Long)
Public Property Get HeaterImpTemp() As Long
Public Property Let HeaterImpTemp(ByVal vData As Long)
Public Property Get HeaterImpDuration() As Long
Public Property Let HeaterImpDuration(ByVal vData As Long)
Public Property Get HeaterImpCycle() As Long
Public Property Let HeaterImpCycle(ByVal vData As Long)
'Temperature correction factors
Public Property Get TempFactorON() As Boolean
Public Property Let TempFactorON(ByVal vData As Boolean)
Public Property Get TempFactorK1() As Long
Public Property Let TempFactorK1(ByVal vData As Long)
Public Property Get TempFactorK2() As Long
Public Property Let TempFactorK2(ByVal vData As Long)
Public Property Get TempFactorK3() As Long
Public Property Let TempFactorK3(ByVal vData As Long)
Public Property Get TempFactorK4() As Long
Public Property Let TempFactorK4(ByVal vData As Long)
Public Property Get TempFactorK5() As Long
Public Property Let TempFactorK5(ByVal vData As Long)
 ‘Properties to get / set values in Graph section of Setup TAB
Public Property Get GraphFrequency() As Long
Public Property Let GraphFrequency(ByVal vData As Long)
Public Property Get GraphImagePath() As String
Public Property Let GraphImagePath(ByVal vData As String)
Public Property Get GraphGenerateCloud() As Long
Public Property Let GraphGenerateCloud(ByVal vData As Long)
Public Property Get GraphPoints() As Long
Public Property Let GraphPoints(ByVal vData As Long)
Public Property Get GraphLabelColorCloud() As Long
Public Property Let GraphLabelColorCloud(ByVal vData As Long)
Public Property Get GraphGraphColorCloud() As Long
```

```
Public Property Let GraphGraphColorCloud(ByVal vData As Long)
Public Property Get GraphCreateImageCloud() As Long
Public Property Let GraphCreateImageCloud(ByVal vData As Long)
Public Property Get GraphGenerateRain() As Long
Public Property Let GraphGenerateRain(ByVal vData As Long)
Public Property Get GraphLabelColorRain() As Long
Public Property Let GraphLabelColorRain(ByVal vData As Long)
Public Property Get GraphGraphColorRain() As Long
Public Property Let GraphGraphColorRain(ByVal vData As Long)
Public Property Get GraphCreateImageRain() As Long
Public Property Let GraphCreateImageRain(ByVal vData As Long)
Public Property Get GraphGenerateBrightness() As Long
Public Property Let GraphGenerateBrightness(ByVal vData As Long)
Public Property Get GraphLabelColorBrightness() As Long
Public Property Let GraphLabelColorBrightness(ByVal vData As Long)
Public Property Get GraphGraphColorBrightness() As Long
Public Property Let GraphGraphColorBrightness(ByVal vData As Long)
Public Property Get GraphCreateImageBrightness() As Long
Public Property Let GraphCreateImageBrightness(ByVal vData As Long)
Public Property Get GraphGenerateTemp() As Long
Public Property Let GraphGenerateTemp(ByVal vData As Long)
Public Property Get GraphLabelColorTemp() As Long
Public Property Let GraphLabelColorTemp(ByVal vData As Long)
Public Property Get GraphGraphColorTemp() As Long
Public Property Let GraphGraphColorTemp(ByVal vData As Long)
Public Property Get GraphBackColorTemp() As Long
Public Property Let GraphBackColorTemp(ByVal vData As Long)
Public Property Get GraphMaxTempTemp() As Long
Public Property Let GraphMaxTempTemp(ByVal vData As Long)
Public Property Get GraphMinTempTemp() As Long
Public Property Let GraphMinTempTemp(ByVal vData As Long)
Public Property Get GraphCreateImageTemp() As Long
Public Property Let GraphCreateImageTemp(ByVal vData As Long)
'Properties to get / set values in Alarm section of Setup TAB
Public Property Get AlarmSuspendSound() As Long
Public Property Let AlarmSuspendSound(ByVal vData As Long)
Public Property Get NoConsecutiveUnknownReads() As Long
Public Property Let NoConsecutiveUnknownReads(ByVal vData As Long)
Public Property Get NoConsecutiveRS232Errors() As Long
Public Property Let NoConsecutiveRS232Errors(ByVal vData As Long)
'Properties to get / set values in Network section of Setup TAB
Public Property Get FileName() As String
Public Property Let FileName(ByVal vData As String)
Public Property Get filePath() As String
```

Public Property Let filePath(ByVal vData As String)

Public Property Get FileSaveFrequency() As Long

Public Property Let FileSaveFrequency(ByVal vData As Long)

'---------- Miscellaneous Methods / Properties

Public Sub AAG_AlarmStart(vData As String)

Plays sound file defined by vData (=path + name)

Public Sub AAG_AlarmStop()

Stops playing sound alarm

Public Property Get AAG_DeviceRole() As String

Returns "MASTER" or "REMOTE"

Public Property Let AAG_DeviceSwitchCtrl(ByVal vData As Boolean)

vData = true - enables the switch control by AAG_CloudWatcher

vData = False – disables switch control by AAG_CloudWatcher

NB: It is only valid if operating in MASTER mode.

NB: When it is disabled use AAG_SetSwitch to control switch.

Public Property Get AAG_DeviceSwitchCtrl() As Boolean

True = switch control is enabled

False = switch control is disabled

Public Sub AAG_SetSwitch(ByVal vData As Long)

vData = 2 closes switch

vData = 1 opens switch

Public Sub AAG_LocalControl(ByVal vData As Boolean, ByVal sLabel as String)

vData = True disables the AAG_CloudWatcher internal algorithm that determines the Safe / Unsafe condition and it hides the Unsafe TAB displaying sLabel string in its place.

NB: If True then the Safe / Unsafe condition must be calculated by an external program and it should be passed to AAG_CloudWatcher using SetUnSafe method

Public Sub SetUnSafe(ByVal iP As Boolean)

Unsafe when vData = True

Safe when vData = False

Public Property Get AAG_SpeedUnits() As String

Returns always “x”

Public Property Let AAG_SpeedUnits(ByVal vData As String)

Has no effect

Public Property Get AAG_TemperatureUnits() As String

“C” for degree Celsius

“F” for degrees Fahrenheit

Public Property Let AAG_TemperatureUnits(ByVal vData As String)

vData = “C” for degree Celsius

vData = “F” for degrees Fahrenheit

Public Function AAG_XFERFileErr() As Boolean

True - Error with data transfer file used between MASTER & REMOTE. It usually corresponds to an open / create file.

False – No error with data transfer file

Public Sub RecordStart(ByVal flgAppendToFile As Boolean)

flgAppendToFile = True starts recording in user data file in append mode;
False starts recording in user data file but clears file first.

Public Sub RecordStop()

Stops recording in user data file

Public Function DeviceIsRunning() As Boolean

True - if device is running

False - if device is not running

Public Sub WindowNormal()

Restores window to its normal size

Public Sub WindowMinimize()

Minimizes window

Public Property Get WindowLeft() As Long

Public Property Let WindowLeft(ByVal leftPos As Long)

Public Property Get WindowTop() As Long

Public Property Let WindowTop(ByVal topPos As Long)

To set / get window position of TOP LHS corner of window.

Public Sub BlinkLEDCOM()

Blinks yellow LED in panel showing number of attached users.

Public Sub Device_Start()

It starts device.

Public Sub Device_Stop()

It stops device.

NB: This method does not check the presence of other programs communicating with AAG_CloudWatcher.

Use AAG_TotalNoUsers to check the total number of users - if it is greater than 1, the device must **NOT** be stopped.

Public Sub AAG_DeviceStop()

Similar to Device_Stop but it only stops the device if number of users = 1.

Public Function AAG_TotalNoUsers() As Long

NB: This method was added in version **5.00.110**

Return the total number of users.

NB: The minimum value is 1.

Public Function LastReading() As Variant

Variant containing Date + time of the last device reading

Public Function Device_Mode() As String

NB: This method was **eliminated** from version **5.00.110.**

Use AAG_DeviceRole.

Maintenance

Routine Maintenance

After a period of time, the rain sensor surface as well as the infrared sensor window will accumulate dirt.

Use a piece of soft cotton wool and (maybe soapy) water to clean these surfaces. Let them dry. Do not rub strongly, especially not on the rain sensor surface.

Take care not to scratch or leave fingerprints on the surface of the sensor.

Appendix 1 - Limitations

Please read this section carefully.

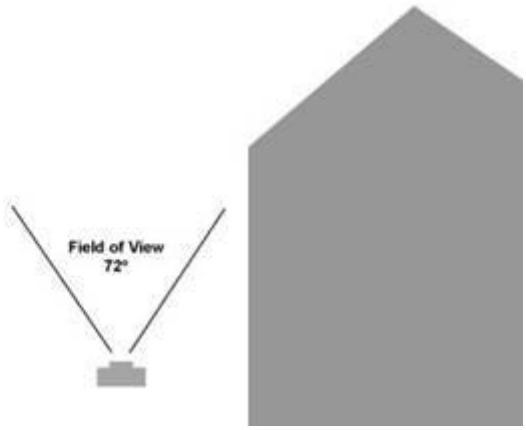
It is important to realize that the good operation of the device depends on a number of aspects, namely,

- the windows of the infrared sensor and the light sensor as well as the surface of the rain sensor are kept clean (see Maintenance).
- there are no local obstacles in the path of the sensors.

Some aspects / limitations that one should be aware of:

a. The device has not the capability to warm up the infrared sensor and so it cannot remove automatically snow and ice from the IR sensor window - under these circumstances the sensor will yield unreliable information.

b. The infrared sensor unit covers a sky angle around 72Âº. This field of view should be oriented towards the sky and completely unobstructed of any surrounding bodies such as walls, roof edges, etc. to avoid the capture of infrared radiation from these sources.



Device too close to the wall.

It should be placed further away from the wall and the roof edge

c. Mount the unit at a slight angle to allow the water to drain from rain sensor surface.

d. Do not place transparent glass, transparent acrylic materials (Perspex), transparent plastic sheet, etc. in front of the device because they act as infrared blocking filters preventing the correct measurement of the sky infrared radiation.

e. If foreign matters get stuck to the rain sensor surface (e.g. bird excrements, etc.) this will cause the device to permanently indicate either a WET or RAIN condition.

f. The external cover of the connecting cable may deteriorate with sun radiation, so it is advisable to protect it from direct sun radiation.

g. The software has been fully tested in Windows XP Home & Professional Editions, and all other Windows versions Vista, 7, 8, up to Windows 10, both in 32 and 64 bits.

Infrared Sensor

The infrared sensor is a MLX90614 ESF-AAA from Melexis Microelectronic Integrated Systems (www.melexis.com).

Rain Sensor

The rain sensor is currently manufactured by Lunatico AstronomÃa.

It was originally from Telecontrolli Srl (http://www.telecontrolli.com/eng/default.html)

Switch

The switch makes use of a reed relay IC (PRMA1A12 from SRC@ DEVICES) and its contacts are rated for:

Switching current	0.5 A DC Max.
Carry current	1.0 A DC Max.
Switching power	10 VA Max.
Maximum voltage	100 V DC Max.
Contact resistance	150 milliohms Max.

Input Power

The recommended input voltage is 15 volts DC (max.400 mA), but it may vary from 12–18 volts DC.

The polarity of the contacts is irrelevant seeing there is an internal rectifier bridge, thus the unit may also be supplied with 12-18 volts AC (not recommended).

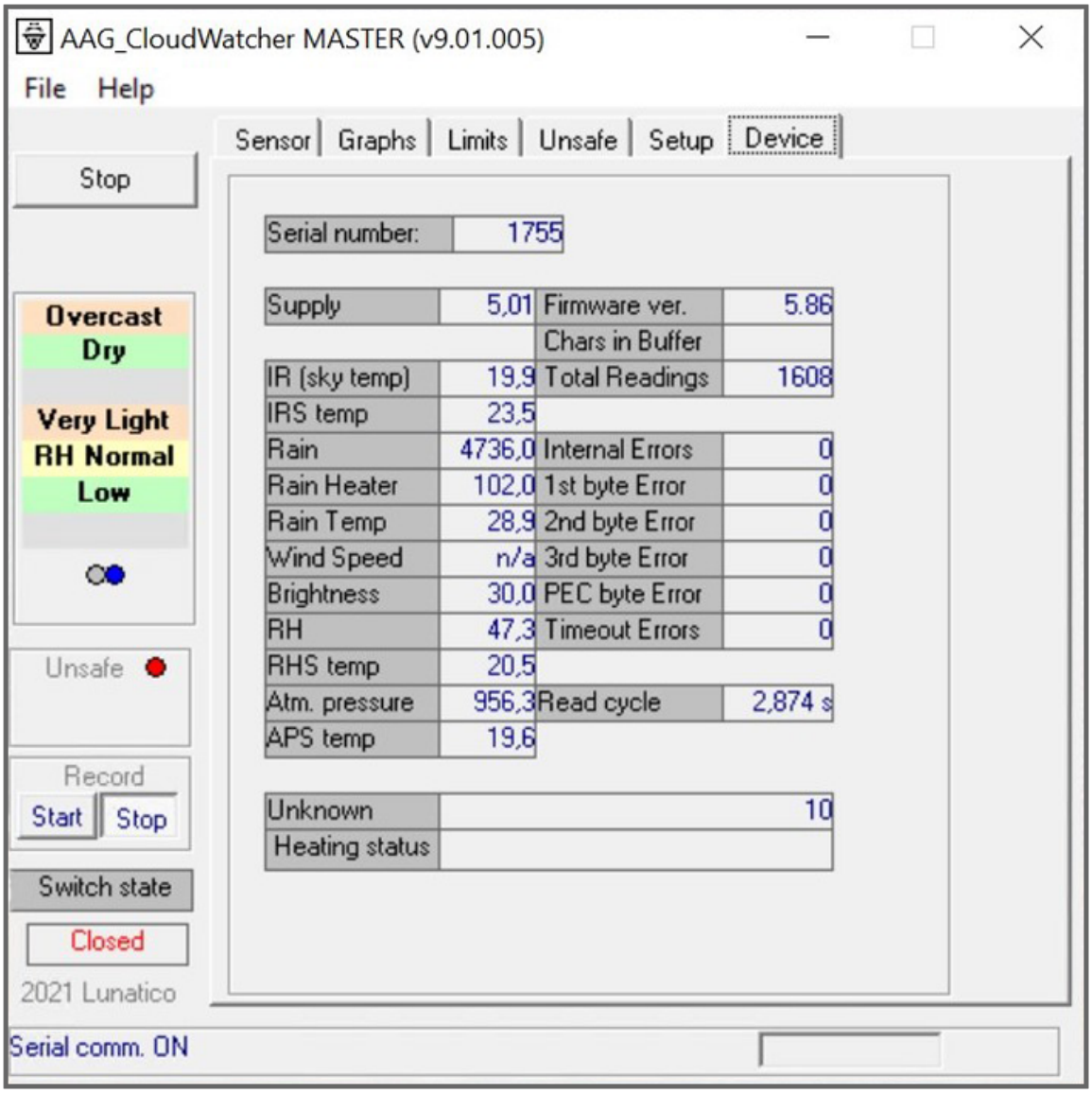
The more recent versions of the CloudWatcher device have a firmware update capability.

The process is performed through the RS232 communication port and using the utility AAG_HASFileLoader.exe located in the same directory as AAG_CloudWatcher.exe (by default C:\Program Files\AAG_CloudWatcher).

The firmware image should be stored in a file with name AAG_CloudWatcherxxx.has and be present in the same directory as AAG_HASFileLoader.exe, where xxx corresponds to the firmware version (e.g. AAG_CloudWatcher550.has corresponds to firmware version 5.50).

Firmware update procedure

a. Check the firmware version installed in the device before performing any update. The firmware version is shown on the Device tab.

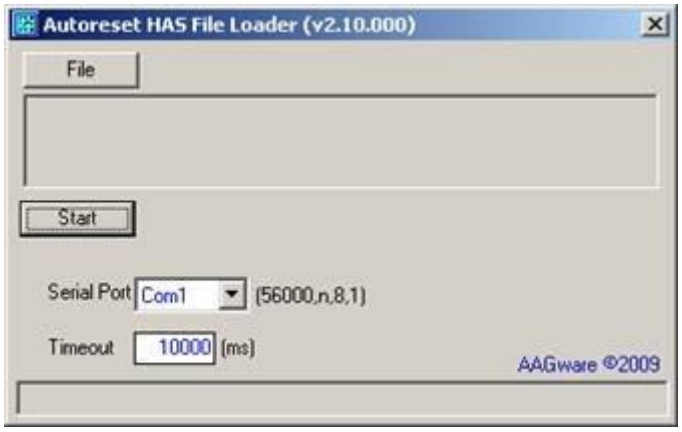


IMPORTANT WARNING
If firmware version is 3.xx - DO NOT update it with a firmware version 5.xx. Firmware version 5.xx is designed for a different microprocessor than versions 3.xx and they are not compatible.

b. Connect the device serial port to the computer serial port.

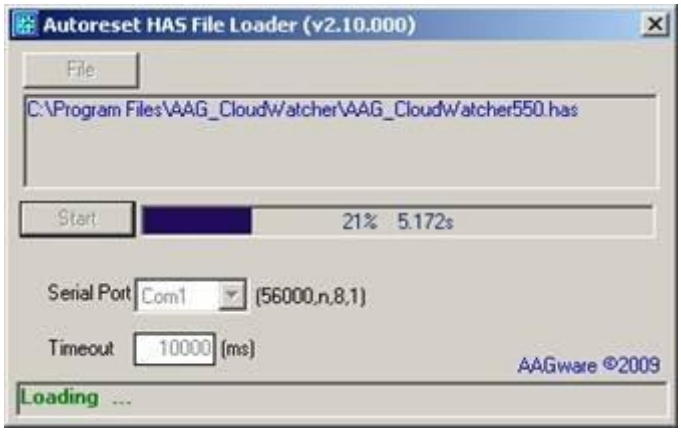
c. Disconnect the power from the device;

d. Start program AAG_HASFileLoader.exe and select the port number that corresponds to the computer port to which the device is connected.



e. Press 'File' button to select file containing firmware image (e.g. file AAG_CloudWatcher550.has for firmware version 5.50).

f. After selecting file, press 'Start' button and then immediately connect the power to the device. The program will start updating the firmware as illustrated below.



g. After finishing the firmware update the program displays the following message.

Appendix 4 - Status Codes

Message	Description
C51 – CCDAutopilot Data File – open/write error	This warning indicates that CCDAutopilot4 data file cannot be created / written to.

d11 - Data File Path does not exist	The directory corresponding to the path defined for the data file under Setup TAB / Device section has been deleted or its access has been restricted. In a network, it may mean that a network map has not been connected.
d12 - Data File Path could not be created	The program attempted to re-create this path without success. This may be caused by a change of write permission for the drive defined in the path. In a network, it may mean that a network map has not been connected.
d13 - Data cannot be written to data file	This message is shown when Start record button is pressed and there is no-write permission for the directory pointed by the path.
i21 - Image File Path does not exist	The directory corresponding to the path defined for the image files under Setup TAB / Graphs section has been deleted or its access has been restricted. In a network, it may mean that a network map has not been connected.
i22 - Image File Path could not be created	The program attempted to re-create this path without success. This may be caused by a change of write permission for the drive defined in the path. In a network, it may mean that a network map has not been connected.
i23 - Image Files cannot be created	This warning indicates that the image files are not being created. This may be caused by a lack of write permission for the directory pointed by the path.
i24 - Image Files FreeImage problem	This warning indicates that the image files are not being created because of a problem with FreeImage library. The FreeImage.dll file should be installed in the same directory as AAG_CloudWatcher.exe (default) or in the windows system directory (usually \Windows\System32).
r31 – COMM Data File– open / read error	This warning indicates the COMM data file does not exist or is corrupted.
w41 – COMM Data File– open / write error	This warning indicates that COMM data file cannot be created.
w42 – COMM Data too old.	

	This warning indicates that the COMM Data is older than the timeout value defined in Device section of the Setup TAB.

Appendix 5 - Boltwood Format

The following can be used to decode the data in the aag_sld.dat file:

- File write date.
- File write time.
- Temperature scale (Celsius or Fahrenheit).
- Wind speed scale (Mph or Knots).
- Sky temperature.
- Ambient temperature.
- Sensor temperature.
- Wind speed.
- Humidity.
- Dew point.
- Dew heater percentage.
- Rain flag.
- Wet flag.
- Elapsed time since last file write.
- Elapsed days since last write.
- Cloud/Clear flag (1=Clear,2=Light Clouds,3=Very Cloudy).
- Wind Limit flag (1=Calm,2=Windy,3=Very Windy).
- Rain flag (1=Dry,2=Damp,3=Rain).
- Darkness flag (1=Dark,2=Dim,3=Daylight).
- Roof close flag.
- Safe / unsafe flag (0=Safe, 1=Unsafe).

Appendix 6 - Sky Temperature Correction Model

Sky Temperature Correction Model

The basic approach to determine the sky temperature is simply subtracting the ambient temperature from the infrared measured temperature, as in:

$$T_{sky} = T_s - T_a$$

where: T_{sky} = Corrected Sky Temperature (ÂºC); T_s = Infrared Sky Measured Temperature (ÂºC); T_a = Ambient temperature (ÂºC).

This simple approach, however, requires frequent changes to the limits - that is, the resulting cloud detection temperature does not remain the same as the weather changes through the year.

To improve on this, the CloudWatcher has a polynomial model to adjust the correction value depending on the ambient temperature, with different weights, given by:

$$T_d = (K1 / 100) * (T_a - K2 / 10) + (K3 / 100) * (Exp (K4 / 1000* T_a)) ^ (K5 / 100) + T_{67}$$

where: T_d = Correction value (ÂºC); T_a = Ambient temperature (ÂºC); K1, K2, K3, K4, K5, K6 and K7 are the coefficients defined in the Setup => Device section; T_{67} (cold weather factor) calculation is shown below.

Exp(n)= *e* (the base of natural logarithms) raised to the power of n. A^b = a raised to the power of b.

The corrected sky temperature (Tsky) is then given by:

$$T_{sky} = T_s - T_d$$

where T_{sky} = Corrected Sky Temperature (ÂºC); T_s = Infrared Sky Measured Temperature (ÂºC); T_d = Correction value (ÂºC)

Computing the sky temperature this way, and after proper calibration, makes it possible to have accurate detection along a wide temperature range.

Calculation of T67(cold weather) term

If Abs((K2 / 10 - T_a)) < 1 Then

$$T_{67} = Sgn(K6) * Sgn(T_a - K2 / 10) * Abs((K2 / 10 - T_a))$$

Else

$$T_{67} = K6 / 10 * Sgn(T_a - K2 / 10) * (Log(Abs((K2 / 10 - T_a))) / Log(10) + K7 / 100)$$

End If

where. *Sgn*(x) = function that returns the sign of x (or 0 if K6 is 0), Log(x) = function that returns the natural logarithm of x, Abs(x) = function that returns the absolute value of x:

Important: In all calculations the values of the temperatures are in degrees Celsius.

Please note: Leaving K1=100, K2, K3â ¦ K7 = 0 results in the simplified $T_{sky} = T_s - T_a$ Leaving all K factors at 0, the result is the raw measured IR, that is $T_{sky} = T$

Appendix 7 - Tuning The Sky Temperature Model

Tuning the sky temperature correction model

The amount of clouds in the sky is directly related to the infrared radiation emitted by the clouds in the wavelength range of 8-15 Âµm.

The infrared sensor measures the total downward radiation in this range and it contains several components:

- The radiation emitted by the clouds.
- The radiation emitted by the atmosphere.
- The radiation reflected by the ground.

To setup the CloudWatcher one should define temperature limit below which the sky is considered clear and the temperature limit which distinguishes a cloudy sky from an overcast (very cloudy) sky.

From empirical observation, the limit between clear and cloudy conditions corresponds to a value between -6Â°C and -3Â°C (the program default value = -5Â°C) whereas the limit between cloudy and overcast conditions is a value between 0Â°C and 2Â°C (the program default value = 0Â°C).

However, during warm days / evenings, the measured sky temperature values contain a large component of other atmospheric radiation. In order to cope with this effect a temperature correction model has been introduced (please refer to Sky Temperature Correction Coefficients).

The model is a combination of a linear and an exponential relationships where the linear relationship predominates for ambient temperatures below 20Â°C whereas the exponential becomes more pronounced for ambient temperatures above this value.

To tune this model, one should observe the Cloud Conditions graph from sunup to sunset for a clear day.

As the ambient temperature changes during the day the Cloud Conditions graph should remain horizontal, provided the sky conditions remain stable and cloudless.

A downward trend in the graph line as the ambient temperature increases means that the calculated sky temperature correction factor must be increased.

If this trend occurs for temperatures below 25Â°C, the coefficient K1 should be increased by a small amount. Try a value from 33 to 38.

If the trend is more noticeable for values above 30Â°C, then the coefficients K3, K4 and K5 should be adjusted. Try the following combination - K3 between 4 and 10, K4=100 and K5=100.

If the graph is horizontal but too low, adjust the coefficient K2. This shifts the calculated correction value upwards as the coefficient gets smaller and vice-versa. (Note that negative values are allowed for this coefficient.)

The default values for coefficients K1, K2, K3, K4 and K5 are 33, 0, 0, 0 and 0 and this corresponds to a simple linear relationship.

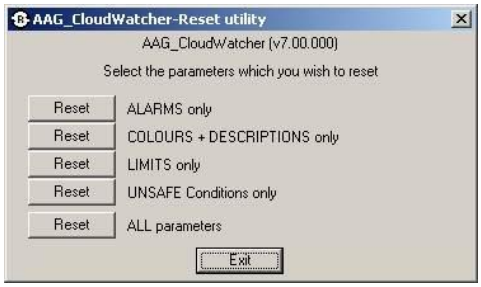
The following coefficients have also proved to yield good results: K1=33, K2=0, K3=8, K4=100 and K5=100. This combination is more nonlinear for ambient temperatures above 30Â°C.

Tip: When adjusting these coefficients, use the graphical display provided (see – Sky Temperature Correction Coefficients).

Appendix 8 - Reset Utility

To clear the default values stored in the Windows registry, run program AAG_ResetParameters.exe which is located in the same directory as AAG_CloudWatcher.exe (by default is C:\Program Files\AAG:CloudWatcher).

Do not run this utility simultaneously with CloudWatcher.



AAG_ResetParameters program

Appendix 9 - Device Internal Parameters

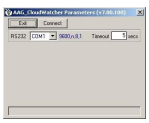
There are three internal parameters stored in the device's hardware which may be reprogrammed by the user.

After a period of time if the device does not receive any requests via the serial port, it runs an internal shut down routine by turning the rain sensor heater to a defined value and setting the internal switch to a particular state.

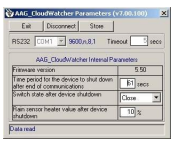
This time period, the switch state as well as the rain sensor heater power may be reprogrammed using the utility *AAG_AdjustInternalPar.exe,* which is installed in the same directory as the main program (by default, C:\Program Files\AAG_CloudWatcher).

To run this utility, simply double-click the exe file (or select menu option "*AAG_CloudWatcher Internal Parameters*" in AAG Software group) and the following screen is displayed:

The device must be connect to the serial port of the computer and powered up but AAG_CloudWatcher program must not be running.



Press 'Connect' button



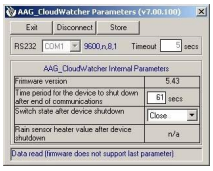
Enter the 'time period', the 'state of the switch' as well as the 'rain sensor heater power percentage' and

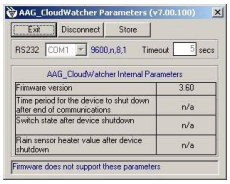
then press 'Store' button.

Note: The following differences with previous firmware versions:

If firmware version is 5.xx but before 5.50 the Rain Sensor Heater parameter is not available.

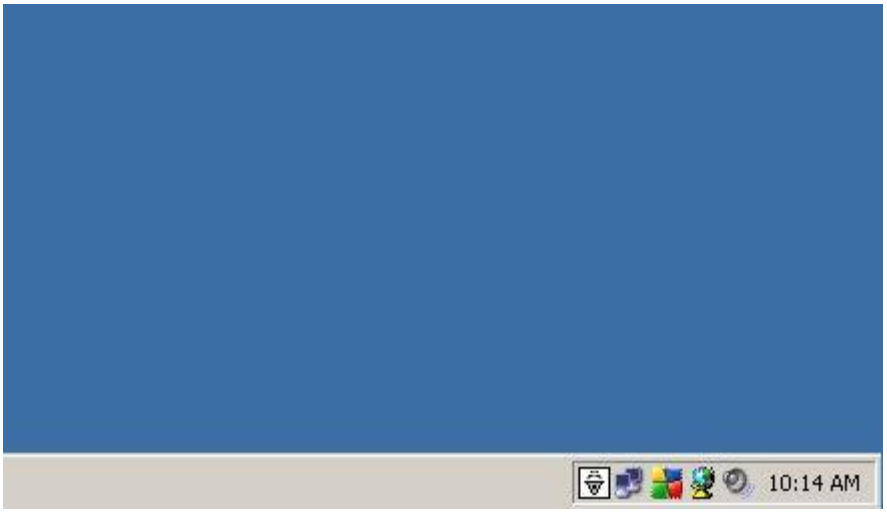
If firmware version is 3.xx these parameters are not available.





Appendix 10 - Program Icon

The program does not create an icon in the task bar but it keeps a small icon in the system tray area with an associated menu (mouse right-click over the icon), as illustrated in figure below. When the main window is minimized, the color of the system tray icon corresponds to the Cloud Condition colour.



Appendix 11 - Version History

Version 9.2.5	Improved handling of external sensors (humidity, pressure) Improved handling of temperature Removed pressure info if sensor disappears More retires on the sensors Some cosmetic fixes Partial support for the future TSL237 sensor
Version 9.2.0	Rain is unknown if > 8000. Fiixed the K-factors setting graph. Pocket PC working fine again. In remote mode amb temp origin is chosen by master (cannot be edited by remote).
Version 9.1.6	Fixed missing music icons. Corrected problems with some german computers.
Version 9.1.5	Wind gust. JSON data file. Bug fixes regarding decimal point and internationalisation.
Version 9.0.0	New interface. Pressure and full RH support. Current rain sensor model limits as defaults.
Version 8.3.1	Fixed a problem with the "black" anemometer. Humidity and dew point in user data file.
Version 8.2.0	Serial port retrying to recover from USB/Serial adapter problems. Cosmetic fixes.
Version 8.0.0	Removed AAG News (was causing an ugly window to appear over the program). Support for pocketCW - PC version. Support for increased resolution ambient temperature and relative humidity. Simplified the version numbering.
Version 7.80.001	Allowed the lower limit of the rain sensor valid value to be 0. Removed the (non working) software update option. Fixed the About dialog.
Version 7.80.000	Improved the unknown detection and indication.
Version 7.60.000	Added another CCDAP4 file with "," as decimal separator, for users with non-localized client software. Supported the Cloud Alarm Delay also in normal operation (previously only useful for external COM access). RH sensor ambient temperature also offset corrected. Support for the new, black, anemometer from IED. Changed the default names for CCAP4 files to "aag_sld.dat" (single line data) and "aag_sldc.dat" (same, decimal comma separated).

Version 7.50.100	Fixed some problems with time syncromization between master & remote mode.
Version 7.30.100	Ambient and IR temperature offsets added to the Master mode.
Version 7.20.000	Added 2 extra fields to the CCDAP4 files.
Version 7.10.000	CCDAP4 files always generated in the new, extended format.
Version 7.00.200	Minor adjustment to cloud graph temperature upper limit. This limit was not correctly converted when changing temperature units from Celsius to Fahrenheit.
Version 7.00.100	First release. This version is designed to run with the new device model which supports an anemometer (optional). The new device (firmware version 5.xx) has a built-in shutdown routine which sets the rain sensor heater power as well as the switch state when the device does not receive any requests for awhile (see Device Internal Parameters).