



NephiSys – Cloud Observing System

User Manual

ROB MUNN

19th May 2016

CONTENTS

INTRODUCTION.....	3
FEATURES.....	3
HARDWARE AND ARDUINO BOARD FEATURES.....	3
SOFTWARE FEATURES.....	5
<i>Cloud Amount/Type Controller.....</i>	<i>5</i>
<i>Cloud Height Controller.....</i>	<i>5</i>
<i>NephiSys Processor.....</i>	<i>5</i>
HARDWARE.....	8
HARDWARE SETUP.....	10
SOFTWARE.....	11
THE CLOUD AMOUNT/TYPE CONTROLLER SOFTWARE.....	12
<i>Log Page.....</i>	<i>12</i>
<i>Environment Graph Page.....</i>	<i>12</i>
<i>Maintenance Graph Page.....</i>	<i>13</i>
<i>Latest Image Page.....</i>	<i>14</i>
<i>Config Page.....</i>	<i>14</i>
THE CLOUD HEIGHT CONTROLLER SOFTWARE.....	23
<i>Log Page.....</i>	<i>23</i>
<i>Images Page.....</i>	<i>23</i>
<i>Manual Operations Page.....</i>	<i>24</i>
<i>WiFi Signals Graph Page.....</i>	<i>24</i>
<i>Configuration Page.....</i>	<i>25</i>
THE PROCESSOR SOFTWARE.....	28
<i>Main Page.....</i>	<i>28</i>
<i>Results Page.....</i>	<i>33</i>
<i>Cloud Height Page.....</i>	<i>34</i>
<i>Log Page.....</i>	<i>34</i>
<i>Previous Images.....</i>	<i>35</i>
<i>Sun/Moon Positions Page.....</i>	<i>35</i>
<i>Cloud Height Calibration.....</i>	<i>36</i>
<i>Live Camera Page.....</i>	<i>36</i>
<i>Sun Calculations Page.....</i>	<i>37</i>
<i>Configuration Page.....</i>	<i>37</i>
UTILITIES.....	53
NEPHISYS CALIBRATOR.....	53
<i>The Calibration Wizard.....</i>	<i>53</i>
<i>The Results Page.....</i>	<i>55</i>
<i>The Calculations Page.....</i>	<i>56</i>
<i>The Config Page.....</i>	<i>56</i>
NEPHISYS CLOUD VIEWER.....	61
NEPHISYS CLOUD SAMPLE EXTRACTOR.....	62

NephiSys - Cloud Observing System – User Manual

INTRODUCTION

The **NephiSys Cloud Observing System** detects the current cloud height, cloud amount and the main cloud type. The cloud height is computed using stereo-photogrammetry, using two cameras approximately 10 metres apart. The cloud amount is determined from the blue component of a whole sky image taken by a fish-eye camera. The same image is used to determine the cloud type using multiple methods: Blob Analysis (for Cumulus), Statistical Analysis, PCA Analysis, Exhaustive Matching Analysis and Cirrus Line Detection Analysis.

This data is available to any external software via a TCP/IP socket connection.

FEATURES

HARDWARE AND ARDUINO BOARD FEATURES

- A whole-sky image is acquired by a fish-eye lens camera, with a 180° field of view.
- An optically transparent acrylic plastic dome covers the whole unit to protect it from the elements.
- A black shade ball is positioned by three servo motors mounted on a robot arm, such that it casts a shadow from the sun over the camera lens, protecting the lens electronics from burn-out, and to minimise 'glare' on the image to be processed.
- An extractor fan ensures a steady flow of outside air through the unit.
- A shade cover over the plastic dome, controlled by a servo motor, keeps the unit cool during the summer, opening only to allow an image to be taken (every 10 minutes by default).
- A heater belt inside the unit is switched on during cold, moist periods (< 15°C and >85% by default), when the plastic dome would normally mist-up, obscuring the sky. The fan is switched off whilst the heater is on.
- The pan and tilt assemblies have rotation limiter switches to minimise the chance that the shade-ball will hit the plastic dome.
- The shade ball, fan, heater and shade cover servos are controlled by an Arduino Uno board.
- The power to the servos, fan and heater are all controlled via relays. The servos are only powered on when required. The fan is on by default (fan relay closed), and the heater is off by default (heater relay closed).
- The unit currently requires three external voltage leads: a) 240VAC for the heater, b) 12VDC for the fan, which also supplies 7.5VDC via a DC-DC converter for the Arduino controller board and servos, and c) 5VDC for the camera.
- Other external connections include a) an CAT6 network cable from the camera to a network switch, and b) a USB cable to the Whole-Sky Camera Controller computer.
- A temperature/humidity sensor constantly measures the temperature and humidity inside the dome.
- A watchdog board is pinged every second, and if the Arduino software freezes, or is commanded to stop pinging the board by the Cloud Amount/Type Controller, the Arduino board is reset automatically after ~8 minutes.
- A white reference panel is included (shown low on the southern horizon in the image in the southern hemisphere) which is used to determine the sky brightness.
- The onboard firmware (Arduino Sketch):
 - Monitors the rotational limit switches (connected in series).
 - Optionally resets itself at around 0100 (Default Off).

- Pings the watchdog.
- Monitors the dome temperature and humidity.
- Turns the heater on at 15°C and 85% between 0600 and 1900.
- Turns the heater off at 20°C and 75%.
- Waits for any of the following commands from the Whole-Sky Camera Controller:
 - Individual servo movement commands
 - Group servo movement commands
 - Request for temperature
 - Request for humidity
 - Handshake command, where any restart messages (if any) are returned
 - Command to set, or request, the current date/time
 - Command to turn the servo relays (two) on/off, or request servo relay state
 - Command to turn the fan relay on/off, or request fan relay state
 - Command to turn the heater relay on/off, or request heater relay state
 - Command to set or return the current 'Sun Tracking' mode. When set, this mode forces the software to check that shade-ball servo movements do not exceed 5° of movement. This mode is set whilst tracking the sun. The mode is always automatically set to false when the servo relays are turned off. For large shade-ball movements (i.e. during calibration), this mode is set to false (or not set to true!)
 - Command to soft reset the Arduino board, i.e. restart the Arduino sketch or program from the beginning
 - Command to hard reset the Arduino board, by telling it NOT to ping the watchdog, which should reset the board within 8 minutes
 - Commands to set the soft maximum and minimum limits on pan, tilt and arm angles. Movements beyond these limits are not allowed
 - Command to set the heater temperature threshold (default: 15°C)
 - Command to set the heater humidity threshold (default: 85%)
 - Command to set the heater temperature hysteresis value (default: 5°C) to be added to the heater temperature threshold
 - Command to set the heater humidity hysteresis value (default: 10%) to be subtracted from the heater humidity threshold
 - Command to return the current servo voltage supplied to the Arduino board
 - Command to return the current Arduino CPU voltage
 - Returns both informational and error log messages back to the Whole-Sky Camera Controller using single letter tokens (to minimise Arduino memory, which is limited). The Whole-Sky Camera Controller maps these tokens to messages
- As the cloud height is determined using stereo-photogrammetry, two SJ4000 (GoPro Compatible) WiFi cameras spaced 10 metres apart are installed, aligned east/west (width) and north/south (height), looking at exactly the same part of the sky. Both have a shade cover that is controlled by a separate Arduino board for each camera.

SOFTWARE FEATURES

CLOUD AMOUNT/TYPE CONTROLLER

The Cloud Amount/Type Controller:

- Computer is situated near the main camera unit (within USB cable range).
- Computes the shade ball positions and commands the Arduino board to position the shade ball, whilst synchronising the servo relays.
- Commands the fish-eye camera to take images every 10 minutes (default) and stores the latest image on the hard drive.
- Requests and records the dome temperature, humidity, CPU voltage, and the Arduino board and servo voltage, on two graphs. Also plots the air temperature originating from an external weather station, via the NephiSys Processor.
- Records the memory being consumed by the application, to check for memory leaks.
- Software parameters are highly customisable with 8 pages of user editable parameters.
- Logs all Cloud Amount/Type Controller and Arduino events and messages to screen and to the Processor.
- Automatically reconnects on lost of contact with the Arduino board

CLOUD HEIGHT CONTROLLER

- Commands the two cloud height cameras to take snapshots of the sky above every 15 minutes, at exactly the same time.
- Records the WiFi signal strength from both cloud height cameras.
- Records the memory being consumed by the application, to check for memory leaks.
- Requests the current weather from the Processor. The cloud height cameras remain covered during rain events.
- Allows remote camera parameter updates (via WiFi).
- Logs all significant events to screen and file.

NEPHISYS PROCESSOR

- Runs remotely from the Cloud Amount/Type Controller and Cloud Height Controller, but is connected via the network.
- Every 10 minutes (by default), the Processor downloads two images acquired by the Cloud Height Controller from cameras 10 metres apart, and computes the cloud heights using stereo-photogrammetry.
- Every 10 minutes (by default), the Processor gets a copy of the latest whole-sky image acquired by the Cloud Amount/Type Controller.
- A user defined black ellipse is optionally rendered onto the whole-sky image over the sun's position to exclude very bright pixels that surround the hardware shade-ball's image.
- The Processor acquires the latest METARs and SYNOPs from local stations and extracts the cloud type(s), but for quality control purposes only.
- If the Meteosoft Met Station software is on the network, it sends the air temperature, air humidity and present weather, which are recorded. The cloud amount is also passed to the Met Station software.

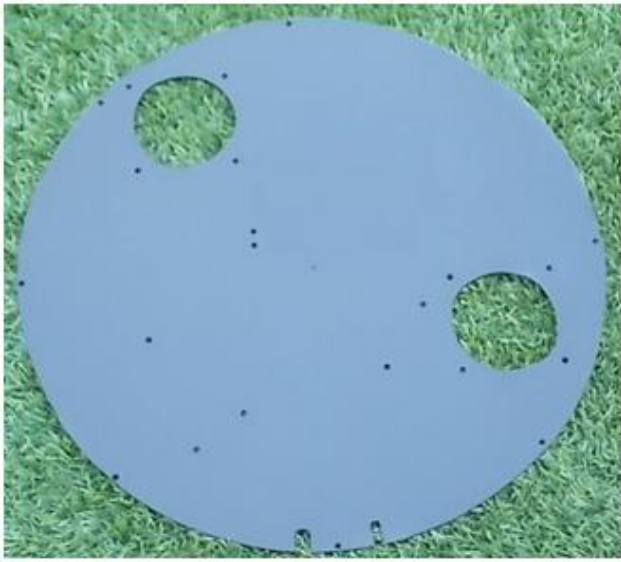
- Automatically reconnects on lost of contact with the Cloud Amount/Type Controller.
- Sends image via FTP to a selected FTP site (e.g. Weather Underground) on the reporting period.
- The Processor currently cannot determine cloud height, amount and type at night. Once the price of bolometer array cameras become affordable, night time detection should be possible.
- The Processor analyses the image to determine the cloud amount, by isolating the blue pixels from non-blue pixels (RBR Threshold Algorithm), accounting for the following:
 - Sky brightness (using the white reference panel on the image)
 - Ignoring trees and other structures on the image
 - Time just after sunrise, and time just before sunset when the horizon is bright compared to the rest of the image
 - Distance of the pixel from the sun, applying a correction to the RBR Threshold
- The Processor computes the cloud type using the following weighted techniques:
 - Blob Analysis, where fragmented Cumulus may be identified (~0.5 sec to process)
 - Maths Analysis, where the texture of the cloud is examined to determine all cloud types (~1.0 sec to process)
 - Principal Component Analysis, where smaller image panes of the whole sky image are transformed into their principal components, and compared to similar images with known cloud types, maintained in a database (~0.4 sec to process)
 - Exhaustive Matching, where smaller image panes of the whole sky image are compared to similar images with known cloud types, maintained in a database (~0.5 sec to process)
 - Line Detection for Cirrus Fibratus, where Cirrus cloud streaks are identified (~0.5 sec to process)
 - All methods are performed simultaneously on parallel threads and usually complete in 1.0 - 1.5 sec
- The Processor displays the following images on the Main tab:
 - The original image
 - A decorated image with the sun-block and today's sun and moon positions
 - A panoramic view of the original image
 - An image of the blue sky detected, masking the cloud in black, and non-cloud areas in green
 - An image of the cloud detected, masking the blue sky in black, and non-blue sky areas in green
 - An image of the detected cloud in grey scale only. This image is used in determining the cloud type, masking the blue sky and non-blue sky and other areas in black
 - The 'Blob Analysis' results image
 - The 'Maths Analysis' results image
 - The 'PCA Analysis' results image
 - The 'Exhaustive Matching Analysis' results image
 - The 'Cirrus Line Detection Analysis' results image
- The cloud amount error rate is ± 1 okta:
 - During the early morning or late afternoon due to horizon brightening,
 - During and just after precipitation (rain drops on the plastic dome may give artifacts)
 - When very thin cirrus cloud is not correctly detected (high blue content)
- Cloud amount is accurate outside the above adverse conditions.
 - METARs and SYNOPs cannot be used to verify cloud amount as:
 - METARs only report cloud amounts as FEW, SCT, BKN and OVC every 30 minutes
 - SYNOPs do report oktas of lowest cloud, but only every 3 hours
- The cloud type accuracy is harder to gauge, being more subjective:

- Computed cloud type is recorded in a file (CloudTypeVerification.txt), together with cloud types found in METARs and SYNOPs
- When the cloud type is compared to that given in local METARs and SYNOPs, there is usually general agreement, except in cases:
 - Where the site issuing the METAR or SYNOP observes and reports cloud on the horizon not visible to the NephiSys hardware
 - Where the site issuing the METAR or SYNOP gives more credence to low amounts of low cloud (that affect aviation) than the observed larger amounts of middle or upper levels cloud that NephiSys may detect
 - Where large Cumulus bases near the NephiSys location are detected as Stratocumulus as the vertical cloud extent is not known
- The Processor displays historical results of cloud amount and type in a table format, and cloud amount in a bar graph format.
- The Processor displays a message log of all events and errors, from the Processor, Cloud Amount/Type Controller and the Arduino control board, and records them to hard disk. These messages may be filtered by message type for display.
- The Processor displays the latest image with sun positions displayed for every hour for mid-Summer, mid-Spring and mid-Winter, between sun rise and sun set, as well as the current sun and moon position.
- A live camera feed is available.
- A calculation page is displayed showing the computed sun angles and paths.
- The Processor is highly customisable with nine pages of user editable parameters.



The central Cloud Amount/Type unit comprises a fish-eye camera, pointing upwards towards the sky which takes an image (1600 x 1200 pixels as a JPEG) every 10 minutes (by default). The lens of the camera is protected from direct sunlight by a black shade-ball on a stiff wire which is moved every minute by three servo motors on a robot arm, such that the shadow of the shade-ball (from the sun) falls on the camera lens. The sensor also contains an extraction fan which vents hot air out of the sensor, allowing cooler outside air in. A temperature and humidity sensor is also included, as is a heater belt (not shown above) that circumvents the unit, lying just inside the outer perimeter.

The whole unit is built on top of a circular 5mm thick Perspex plate, and covered with a high-quality plastic dome with good optical properties, hermetically sealing the unit from the elements.



The unit includes a shade cover which is pulled over the sensor's dome by a servo, protecting the sensor's hardware from direct sun light and heat, except for ~20 seconds prior to the reporting period, when it opens allowing the camera to take an image, before closing again.



Also included are pan and tilt rotation limit cut-off switches, that are activated when the pan or tilt servos are rotated beyond a physical point for any reason. When activated, the power to the servos is cut off (the servo power relay is deactivated) to protect them from overheating, for 15 seconds, then the pan, tilt and arm servos are each instructed to move to a neutral position, placing the ball directly over the camera lens. This condition is transmitted immediately to the Cloud Amount/Type Controller which in turn informs the Processor software, alerting the user to the fault condition.

Maintenance is minimal. Only a periodic clean of the dome exterior and shade cover is required.

The secondary cloud height units are two small SJ4000 WiFi Cameras, set 10 metres apart, both set on thick perspex plates and covered by a shade cover which is pulled over the camera's dome by a servo, protecting the camera from direct sun light and weather. The Cloud Height Controller computer communicates with these cameras using high gain WiFi adapters with aerials.

HARDWARE SETUP

The Cloud Amount/Type unit operates in both the northern and southern hemispheres. In the southern hemisphere, the sensor is positioned such that the robot pan/tilt/arm mechanism position faces north, and the camera position faces the south. This minimises the blockage from the wire arm attached to the shade-ball. In the northern hemisphere, the pan/tilt/arm mechanism position faces south, for the same reason. The unit is able to cover the sun's track from 0° to 120° elevation and 120° - 240° (through North for the southern hemisphere), or 0° to 120° elevation and 60° - 300° (through South for the northern hemisphere).

The two cloud height cameras must be installed such that they are looking at exactly the same part of the sky (at infinite focus), and both must be aligned on the same east/west and north/south axes. The software will fine-tune the calibration, as well as permit manual calibration.

SOFTWARE

The operational software comprises three modules:

- The **Cloud Amount/Type Arduino microcontroller** 'sketch', that moves the servos, reads temperature and humidity inside the dome, switches on the heater when required, pings the watchdog, and acts on instructions from the Cloud Amount/Type Controller. It resides in the dome unit.
- The **Cloud Height Arduino microcontroller** 'sketch' (x2), that moves the sun shade servo for each camera.
- The **Cloud Amount/Type Controller** software, which runs on a small computer near the hardware, and which computes the shade-ball positions during the day, instructing the microcontroller to move the shade-ball servos (via a USB cable), and the acquisition of the whole-sky images from the camera.
- The **Cloud Height Controller** software controls the image acquisition from the cloud height cameras, and allows remote setting of camera parameters.
- The **Processor** software, which processes the whole-sky image, runs on a server PC, computing the cloud amount and type from the fish-eye camera, and the cloud height from the two smaller cameras.

The **Cloud Amount/Type Controller** and **Processor** communicate via a TCP/IP socket connection, where the **Cloud Amount/Type Controller** is the socket server. The **Cloud Height Controller** and **Processor** communicate via a TCP/IP socket connection, where the **Cloud Height Controller** is the socket client.

Every 15 minutes, the **Cloud Height Controller** commands the two cloud height cameras to take snapshots (at exactly the same time) and saves the images for the **Processor** to download and process.

Every minute the **Cloud Amount/Type Controller** software commands the microcontroller to position the shade-ball so that the ball's shadow from the sun always falls on the camera lens, thereby protecting it from direct sunlight when the shade cover is open.

Forty seconds before the 10 minute (configurable) reporting period, the **Cloud Amount/Type Controller** commands the shade cover to open. The **Cloud Amount/Type Controller** then commands the camera to take between 1 and 5 (configurable) consecutive images of the sky with 2 seconds between each image. The reason that optional multiple images are allowed is that the D-Link DCS-6010L camera occasionally returns old images when the latest image is requested, which may include an image with the shade cover closed, confounding the **Processor** software. The latest image is saved on the **Cloud Amount/Type Controller's** hard disk.

Once the image has been acquired, the **Cloud Amount/Type Controller** commands the shade cover to close (approx. 25 seconds before the 10 min reporting period). If the command(s) to the camera fail(s) for any reason, the shade cover will close on the reporting period.

At the 10 minute reporting time, the **Processor** software copies the latest image from the **Cloud Amount/Type Controller** to the server on which it is running, and processes the image to determine the total cloud amount and cloud type(s).

The **Processor** software also provides a TCP/IP server that responds to client connections and requests for the latest cloud amount and type information. This connection is also used to send the **Processor** external

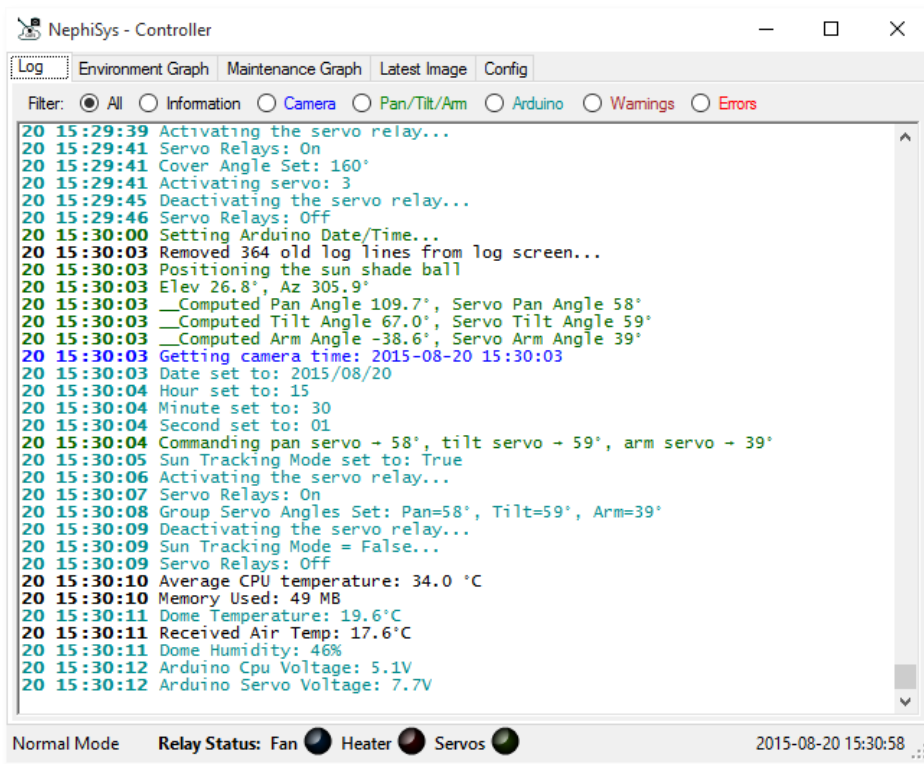
information such as air temperature, air humidity and current weather (rain, showers etc) - see [Appendix B in Supplement](#) for the communication protocol.

THE CLOUD AMOUNT/TYPE CONTROLLER SOFTWARE

The **Cloud Amount/Type Controller** software presents five tabbed views, the **Log View**, **Graph View**, **Maintenance View**, **Latest Image View** and the **Config View**.

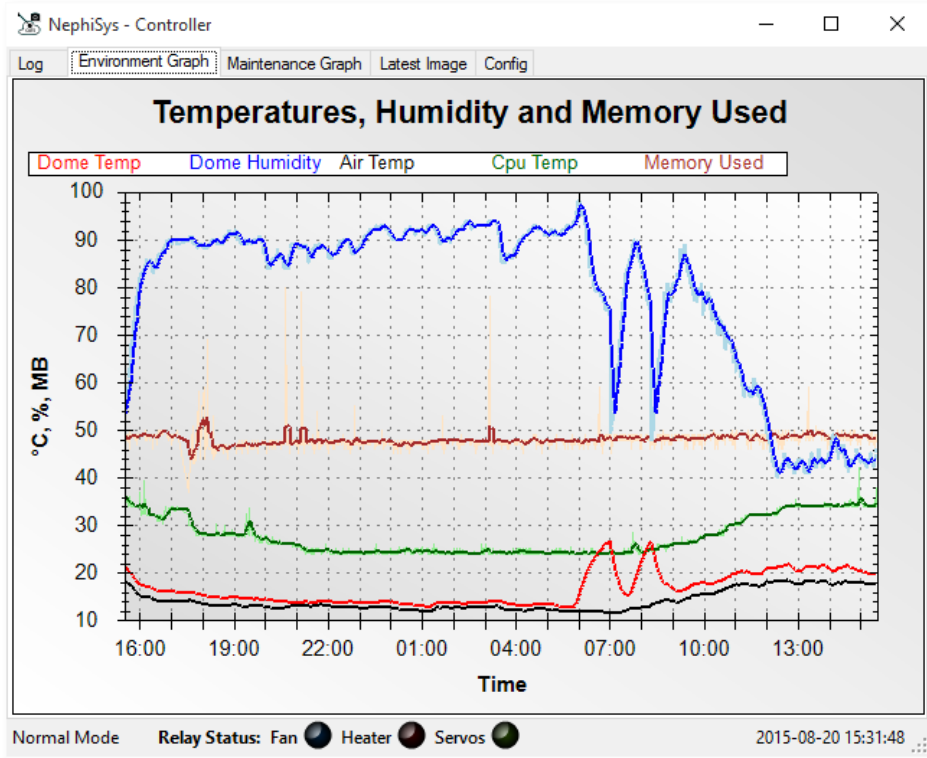
LOG PAGE

The **Log Page** displays the all significant events the controller performs, and all errors and exceptions, with date/time stamps, and sends these messages to the Processor. If the Processor is not connected, it buffers these log messages to be sent when the Processor eventually connects:



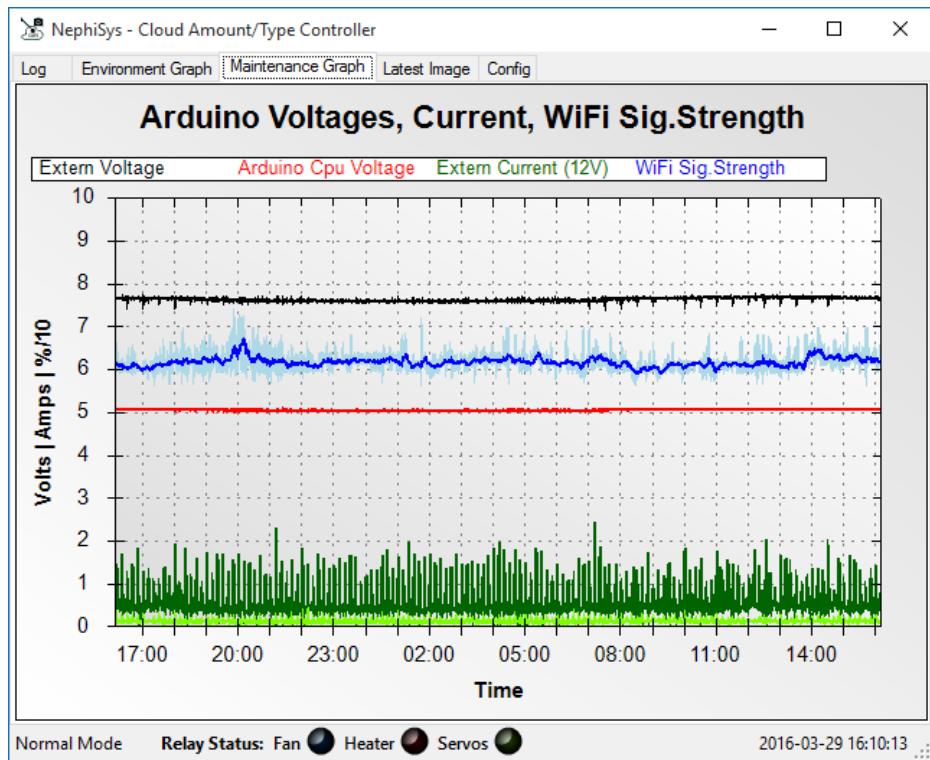
ENVIRONMENT GRAPH PAGE

The **Environment Graph Page** shows a plot of the dome temperature (in red), the air temperature (in black, provided by the Processor), the average controller CPU temperature (in green) and the average dome humidity (in blue). The average memory used by the controller software (in MB) is plotted in brown. The lighter colours are the respective instantaneous values (note that the heater was activated twice between 0600 and 0900LT):



MAINTENANCE GRAPH PAGE

The Maintenance Graph page displays a graph of the voltage being supplied to the Arduino board and servos (external voltage), the current being consumed by the Arduino board and fan (through the 12VDC supply to the DC-DC Converter), the voltage being supplied to the Arduino CPU, and the WiFi signal strength:



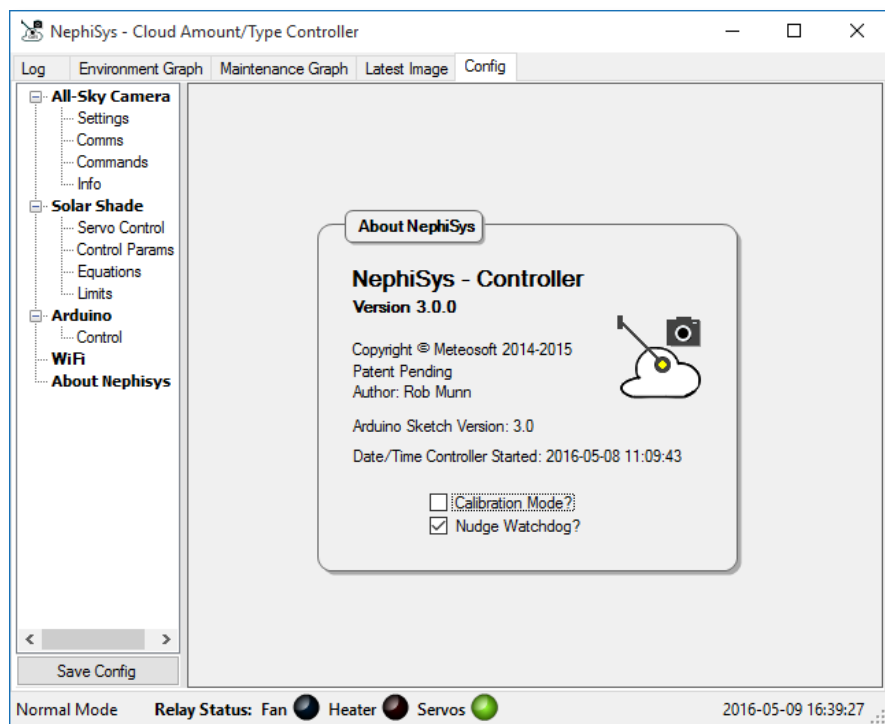
LATEST IMAGE PAGE

The **Latest Image Page** displays the latest image available to the Processor:



CONFIG PAGE

The **Configuration Page** allows the user to configure the controller:



The Controller software allows the user to configure the following:

All-Sky Camera | Settings

Brightness: Adjust the camera's brightness

Contrast: Adjust the camera's contrast

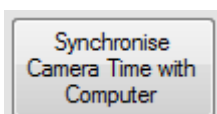
Saturation: Adjust the camera's saturation

Sharpness: Adjust the camera's sharpness

WhiteBalance: Adjust the camera's white balance

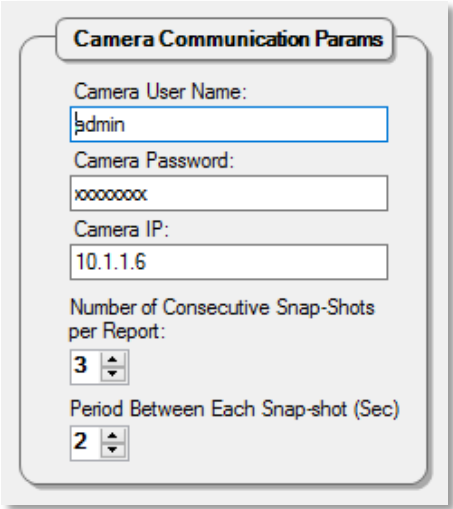
Flip Image?: Flip image vertically? Unchecked = North is Up, Checked = North is Down

Mirror Image?: Mirror the image horizontally? Checked = East is to Right, Unchecked = West is to Right

 Synchronises the Camera Time with Computer time

Camera LED On?: Toggle the camera's LED light

Camera | Comms



The screenshot shows a web-based configuration window titled "Camera Communication Params". It contains several input fields and two spinners. The "Camera User Name" field is set to "admin". The "Camera Password" field is masked with "xxxxxxxx". The "Camera IP" field is set to "10.1.1.6". The "Number of Consecutive Snap-Shots per Report" spinner is set to "3". The "Period Between Each Snap-shot (Sec)" spinner is set to "2".

Field Name	Value
Camera User Name:	admin
Camera Password:	xxxxxxxx
Camera IP:	10.1.1.6
Number of Consecutive Snap-Shots per Report:	3
Period Between Each Snap-shot (Sec)	2

Camera User Name: The user name to log onto the camera's web server

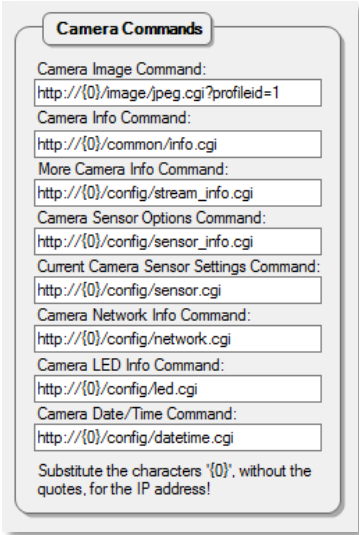
Camera Password: The password to log onto the camera's web server

Camera IP: The Camera IP address

Number of Consecutive Snap-Shots per Report: The number of consecutive snapshots to take to ensure we have the latest image for this report

Period Between Each Snap-shot (Sec): The period between consecutive snapshots for each report

Camera | Commands



The screenshot shows a web-based configuration window titled "Camera Commands". It contains several text input fields for different camera commands. Each field is pre-filled with a URL template where "{0}" represents the IP address. A note at the bottom states: "Substitute the characters '{0}', without the quotes, for the IP address!".

Command Name	Command Value
Camera Image Command:	http://{0}/image/jpeg.cgi?profileid=1
Camera Info Command:	http://{0}/common/info.cgi
More Camera Info Command:	http://{0}/config/stream_info.cgi
Camera Sensor Options Command:	http://{0}/config/sensor_info.cgi
Current Camera Sensor Settings Command:	http://{0}/config/sensor.cgi
Camera Network Info Command:	http://{0}/config/network.cgi
Camera LED Info Command:	http://{0}/config/led.cgi
Camera Date/Time Command:	http://{0}/config/datetime.cgi

Substitute the characters '{0}', without the quotes, for the IP address!


These are the POST commands to send to the camera to get and set images and data. They only need to be adjusted if the camera is changed. Note that the characters {0}, represent the IP address! If specific information is not required, leave the command field blank.

Camera | Info

Camera Information	Network Information
Name: NephiSys-Camera	MacAddr: 28:10:7B:06:76:BD
Model: DCS-6010L	IPAddr: 10.1.1.6
Product: NephiSys-Camera	Netmask: 255.255.255.0
Version: 1.00	Gateway: 10.1.1.1
Build: 0	Wireless: no
Nipca: 1.9	Dhcp: off
Location:	HttpPort: 80
Ptz: no	
Focus: no	
Inputs: 0	
Outputs: 0	
Speaker: yes	
VideoOut: yes	
Pir: no	
Icr: no	
IR: no	
Num Profiles: 3	
AspectRatio: 4:3	

This is a display of all the camera parameters.

Solar Shade | Direct Control



Do NOT use these controls unless the plastic dome has been removed from the unit, as damage may be caused to the servo motors, shade-ball and/or camera lens. Only adjust these when the unit is visible.

Cover: <input type="range" value="0"/> Close 0° Open	Pan: <input type="range" value="46"/> East 46° West	Tilt: <input type="range" value="68"/> Forward 68° Back	Arm: <input type="range" value="45"/> Up 45° Down
---	--	--	--

These controls are used to directly adjust the servo motors that control the shade cover and shade-ball's position. They also reflect the current servo angles. As the warning states: **Do NOT use the pan/tilt/arm controls unless the shade cover is open, and the plastic dome has been removed from the unit, as damage may be caused to the servo motors, shade-ball and/or camera lens. Only adjust these when the unit is visible.**

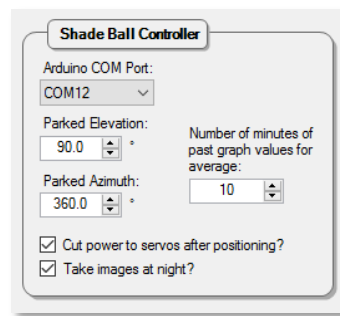
The **[Park Ball]** button will force the shade-ball into the parked neutral position, as set in the configuration.

The **[Day Test]** button will position the shade-ball in the correct hourly position for each hour (every 2 seconds) for the day of the test (today), between sun rise and sun set.

The **[Centre]** button will force the shade-ball into the centred position, just above the camera lens.

The **[Open/Close Cover]** button will force the shade cover to open if closed, or close if open.

Solar Shade | Control Params



Parked Elevation: The shade-ball's parked or centred elevation

Parked Azimuth: The shade-ball's parked or centred azimuth

Number of minutes of past graph values for average: The number of minutes of old graph values that should be used to compute the average value (default 10mins).

Cut servos to sleep after positioning?: If checked, the shade ball positioning servos are relaxed after activation

Take images at night?: If checked, images will be acquired every 10 minutes (default) during the night, otherwise no images will be available between sunset and sunrise. If this option is changed, please restart the Processor, so it too knows not to process night-time images.

Solar Shade-Ball Controller | Conversion Equations

The pan, tilt and arm astronomical angles (based on the elevation and azimuth of the sun at any particular time) are computed such that the shade-ball casts a shadow over the camera lens. However, the Arduino servo microcontroller requires the servo orientated angles in order to position the shade-ball. Each servo produces an (almost) linear relationship between the astronomical angle and servo angle, allowing for slight mis-alignment of servo fixtures:

	Astronomical Angle	Equivalent Servo Angle
Pan	East -> North -> West, 090° -> 360° -> 270°	180° -> 090° -> 0°
Tilt	Backward -> Upright -> Forward, 0° -> 090° -> 180°	0° -> 090° -> 180°
Arm	Pointing Up -> Pointing Towards Camera Lens, Pointing Down, 090° -> 0° -> -090°	0° -> 090° -> 180°

The linear regression equations that relate the astronomical angle to the servo angle are entered here:

Shade Ball Equations in the form $Y[\text{Servo}] = mX[\text{Astro}] + c[\text{Intercept}]$

Pan Servo Angle =	Pan Equation Gradient:	Pan Intercept:
	-1.314	196.3
Tilt Servo Angle =	Tilt Equation Gradient:	Tilt Intercept:
	1.467	-41.8
Arm Servo Angle =	Arm Equation Gradient:	Arm Intercept:
	1.618	108.0

[Edit User Methods](#)

☒ Do NOT Apply User Methods
☐ Fine Tune Linear Equations Applying User Methods
☐ Ignore Linear Equations and Apply User Methods Only

Pan Equation Gradient: The astronomical Pan angle to servo angle equation's gradient

Pan Intercept: The astronomical Pan angle to servo angle equation's Y intercept

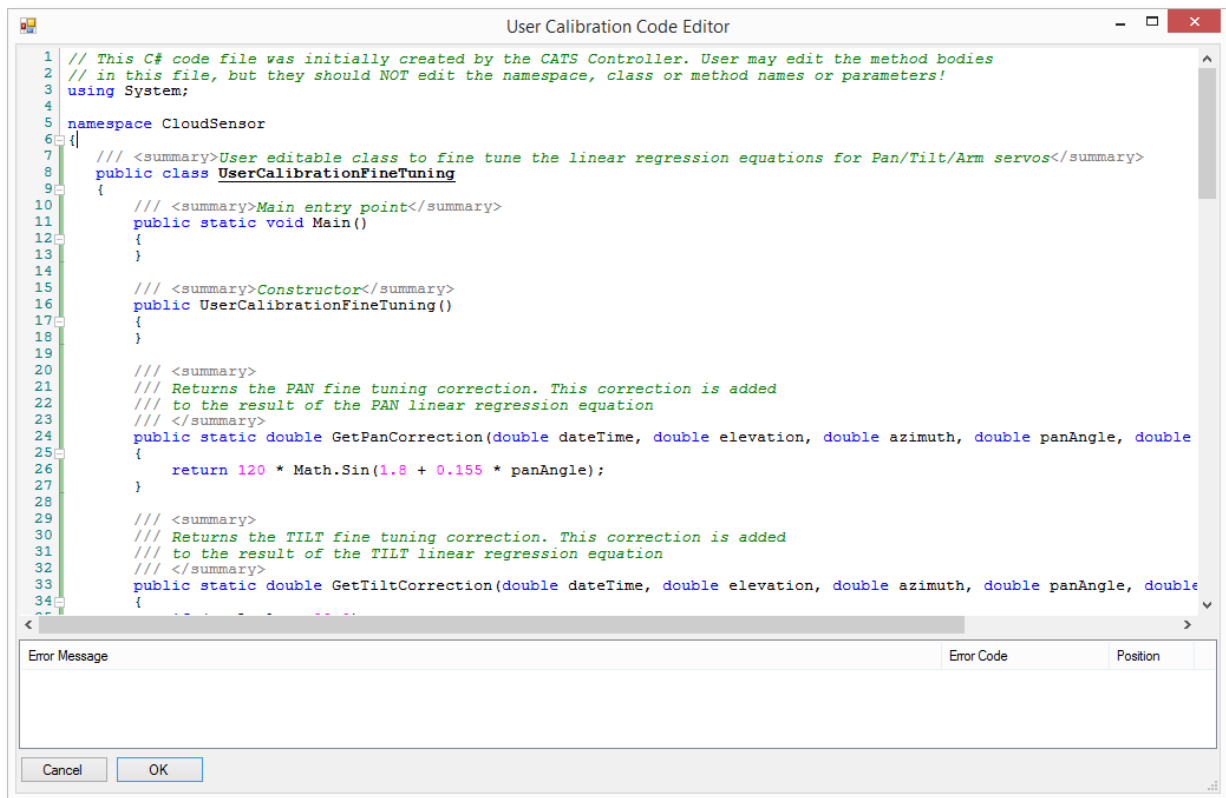
Tilt Equation Gradient: The astronomical Tilt angle to servo angle equation's gradient

Tilt Intercept: The astronomical Tilt angle to servo angle equation's Y intercept

Arm Equation Gradient: The astronomical Arm angle to servo angle equation's gradient

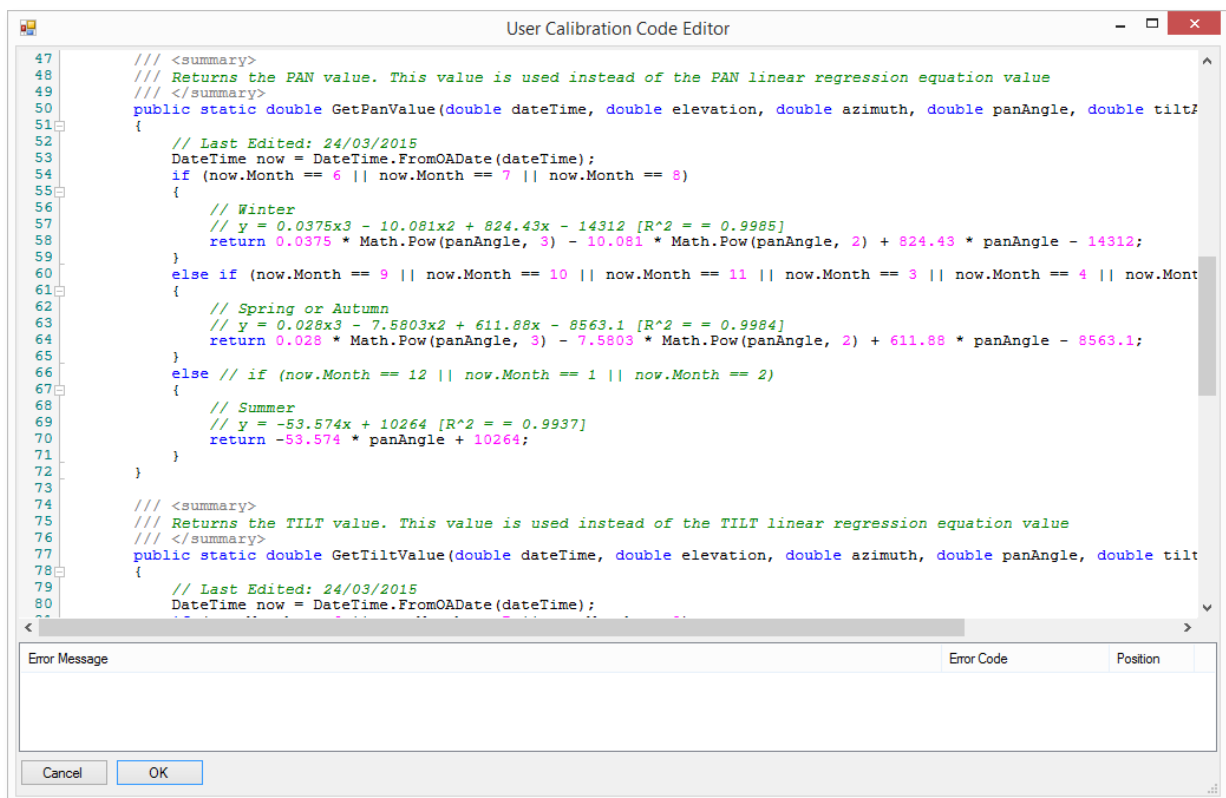
Arm Intercept: The astronomical Arm angle to servo angle equation's Y intercept

Clicking [Edit User Methods](#), presents a code editor in which the **advanced** user may enter C# scripting code to fine tune the servo angles (effectively adjusting the y-intercept of the linear equations), depending on the hemisphere (east/west), in the **GetXXXCorrection()** methods:



OR

the advanced user may ignore the linear equations altogether and write their own equations (e.g. polynomial) in the **GetXXXValue()** methods:



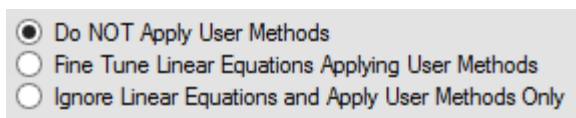
All methods are passed the **Date/Time** for the ball position, the corresponding **Elevation** and **Azimuth** of the sun, as well as the **Pan, Tilt and Arm astronomical angles**. The **GetXXXCorrection()** methods return a value to be added to the linear equation's Y-intercept. The **GetXXXValue()** methods return the computed servo angle for the associated servo.

If the **UserCalibrations.cs** file (the one edited above) does not exist, the button text reads [**Create User Methods**], and when clicked, will create a template for the user.

When the user clicks [**OK**] after editing the code, the whole module is verified and any syntax errors found will appear in the error message list below the code window. The user **must** correct the error(s) before saving, or click [**Cancel**]. Double clicking an error will position the editing cursor to the error position.

The code may be edited at any time, and the changes will be applied during the next calculation.

The servo angle computation method used by the **Controller** is selected using the the following radio buttons:

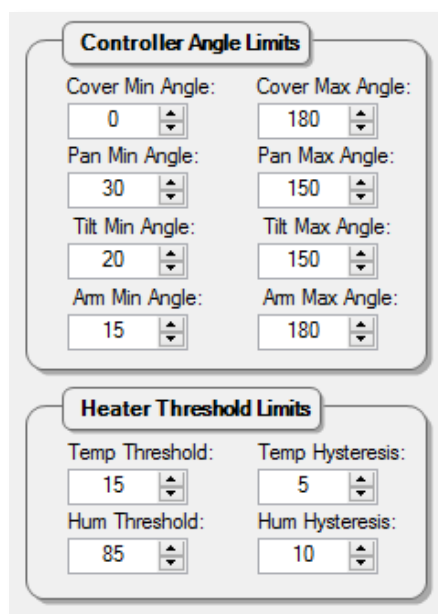


☒ Do NOT Apply User Methods
☐ Fine Tune Linear Equations Applying User Methods
☐ Ignore Linear Equations and Apply User Methods Only

- **“Do NOT Apply User Methods”** – The given linear gradient and intercept values are used to compute the servo angles to position the shade-ball
- **“Fine Tune Linear Equations Applying User Methods”** – The given linear gradient and intercept values are used to compute the servo angles, together with any corrections to the Y-intercept computed in the **GetXXXCorrection()** methods.
- **“Ignore Linear Equations and Apply User Methods Only”** – The **GetXXXValue()** methods are used to compute the servo angles to position the shade-ball

When either of the latter two options are selected and the **GetXXXCorrection()** or **GetXXXValue()** methods are called, the script is compiled into a Dll (in a separate AppDomain to allow unloading of the Dll on completion) and the results applied.

Solar Shade | Servo Limits



Controller Angle Limits

Cover Min Angle: 0	Cover Max Angle: 180
Pan Min Angle: 30	Pan Max Angle: 150
Tilt Min Angle: 20	Tilt Max Angle: 150
Arm Min Angle: 15	Arm Max Angle: 180

Heater Threshold Limits

Temp Threshold: 15	Temp Hysteresis: 5
Hum Threshold: 85	Hum Hysteresis: 10

Cover Min/Max Servo Angle: The minimum/maximum Shade Cover servo motor angles

Pan Min/Max Servo Angle: The minimum/maximum Pan servo motor angles

Tilt Min/Max Servo Angle: The minimum/maximum Tilt servo motor angles

Arm Min/Max Servo Angle: The minimum/maximum Arm servo motor angles

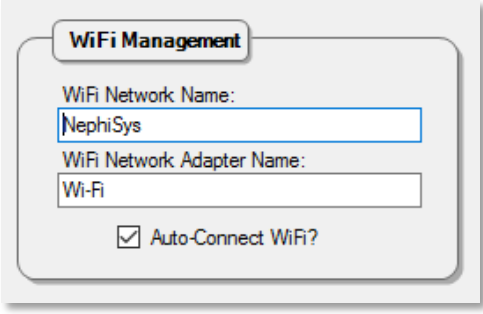
Temp Threshold: The temperature at which the heater must turn on, provided the Humidity Threshold is also reached

Hum Threshold: The humidity at which the heater must turn on, provided the Temperature Threshold is also reached

Temp Hysteresis: The difference in temperature, which when added to the threshold, produces a value that determines when the heater must turn off, provided the value (Humidity Threshold - Humidity Hysteresis) is also reached

Hum Hysteresis: The difference in humidity, which when added to the threshold, produces a value that determines when the heater must turn off, provided the value (Temperature Threshold + Temperature Hysteresis) is also reached

WiFi | WiFi Management

A screenshot of a software window titled "WiFi Management". Inside the window, there are two text input fields. The first is labeled "WiFi Network Name:" and contains the text "NephiSys". The second is labeled "WiFi Network Adapter Name:" and contains the text "Wi-Fi". Below these fields is a checkbox labeled "Auto-Connect WiFi?" which is checked.

WiFi Management

WiFi Network Name:
NephiSys

WiFi Network Adapter Name:
Wi-Fi

☒ Auto-Connect WiFi?

WiFi Network Name: The name of the WiFi network to connect to, to talk to the Processor

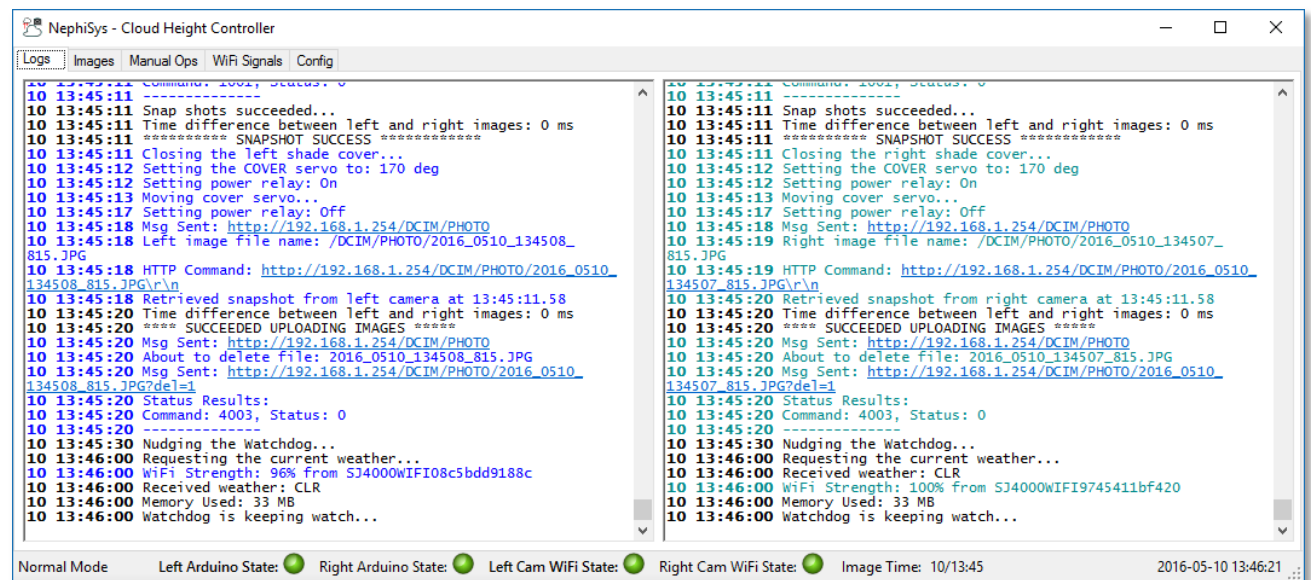
WiFi Network Adaptor Name: The name of the WiFi adapter on the Controller through which the connection will be made

THE CLOUD HEIGHT CONTROLLER SOFTWARE

The **Cloud Height Controller** software presents five tabbed views, the **Log View**, **Latest Images**, **Manual Operations**, **WiFi Strength Graph** and the **Config View**.

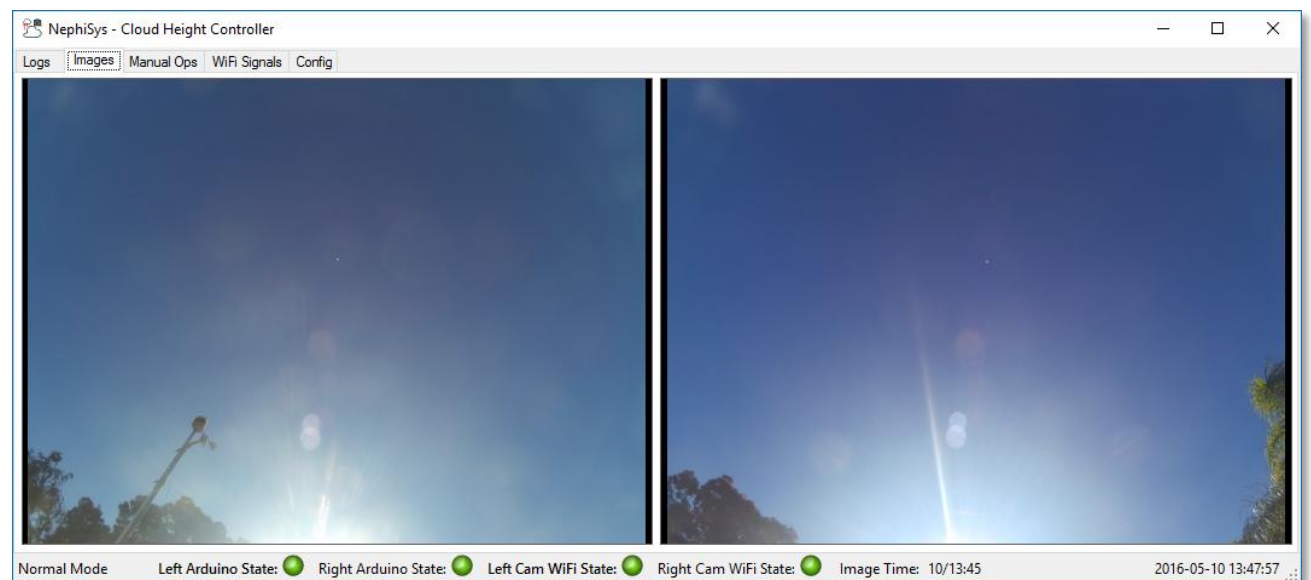
LOG PAGE

The **Log Page** displays the all significant events the controller performs for each camera, and all errors and exceptions, with date/time stamps, and saves these messages to the hard disk:



IMAGES PAGE

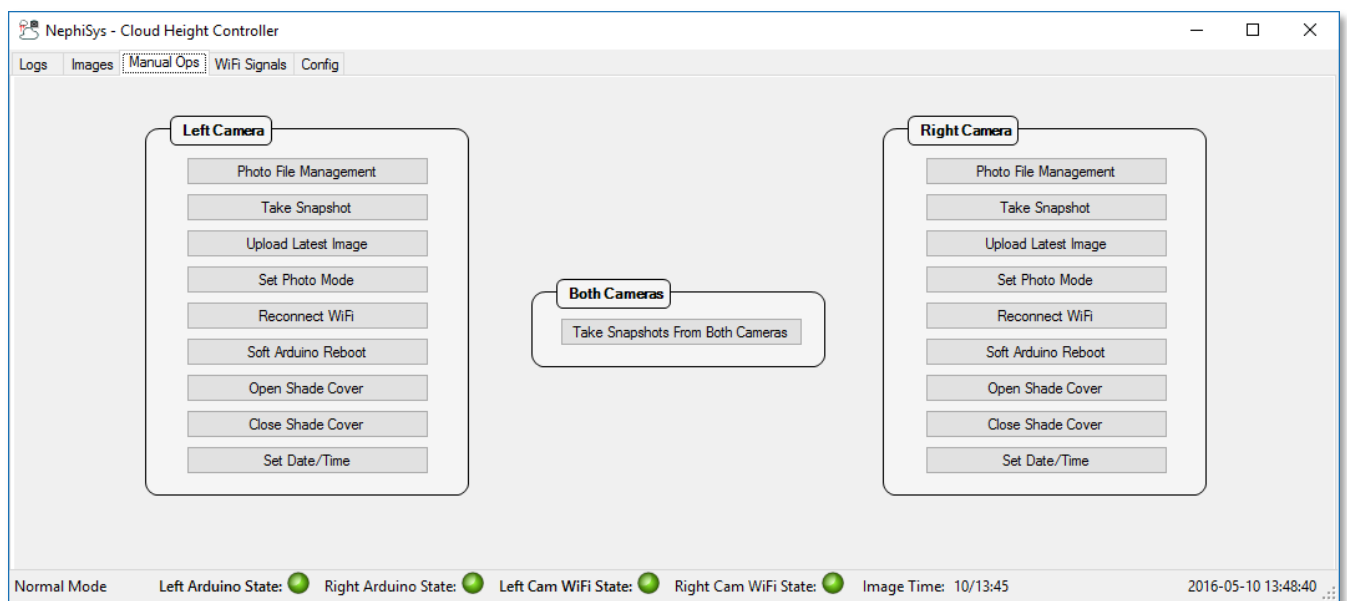
The **Images Page** displays the latest images acquired from both cameras:



MANUAL OPERATIONS PAGE

The **Manual Ops** displays a number of buttons allowing the user to:

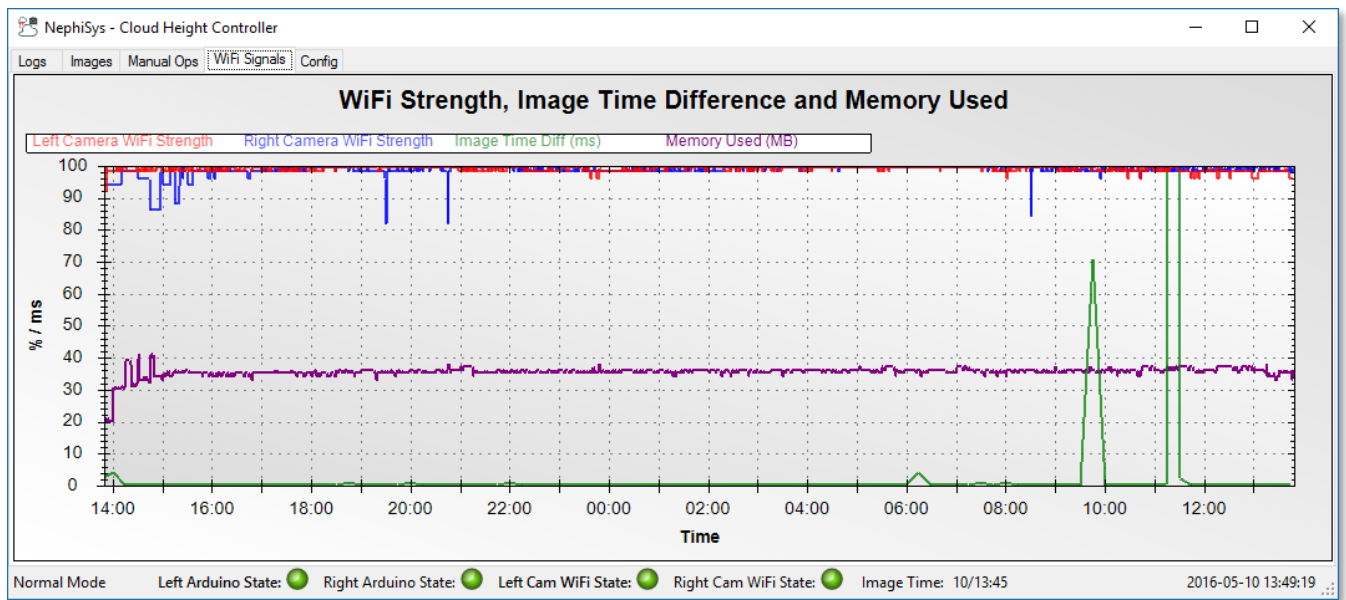
- Manage each camera's file system
- Take camera independent snapshots
- Upload a camera's latest image
- Set the camera's mode to 'Photo'
- Reconnect a camera's WiFi connection
- Reboot a camera's Arduino box (that controls the shade cover)
- Open a camera's shade cover
- Close a camera's shade cover
- Set a camera's date and time
- Take simultaneous snapshots from both cameras



WIFI SIGNALS GRAPH PAGE

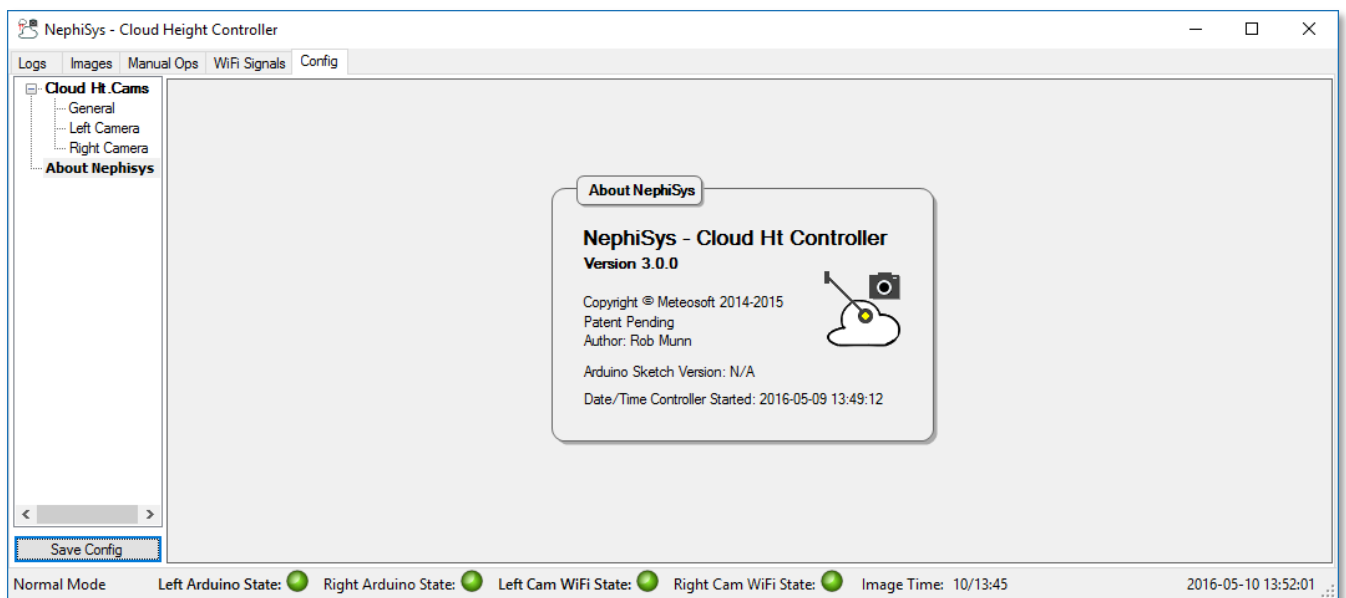
The **WiFi Signals** page displays a graph of:

- WiFi Signal strength from each camera
- The image time difference in milliseconds
- The memory used by the controller



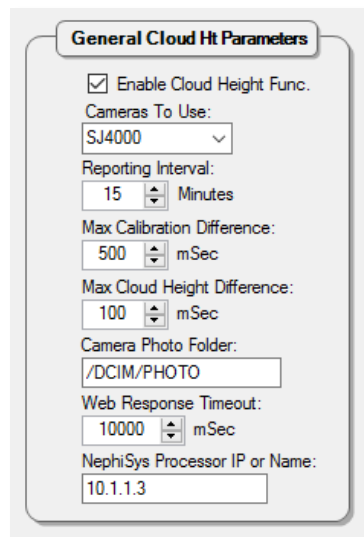
CONFIGURATION PAGE

The **Config** page allows the user to configure the controller and cameras:



The Controller software allows the user to configure the following:

Cloud Ht.Cams | General



General Cloud Ht Parameters

☒ Enable Cloud Height Func.

Cameras To Use:
SJ4000

Reporting Interval:
15 Minutes

Max Calibration Difference:
500 mSec

Max Cloud Height Difference:
100 mSec

Camera Photo Folder:
/DCIM/PHOTO

Web Response Timeout:
10000 mSec

NepiSys Processor IP or Name:
10.1.1.3

Enable Cloud Height Func: If set, images are acquired every reporting interval

Cameras To Use: The camera type in use

Reporting Interval: The interval between image acquisition from the cameras

Max Calibration Difference: The maximum number of milliseconds between the image snapshot times for calibration, which if exceeded, renders the images invalid

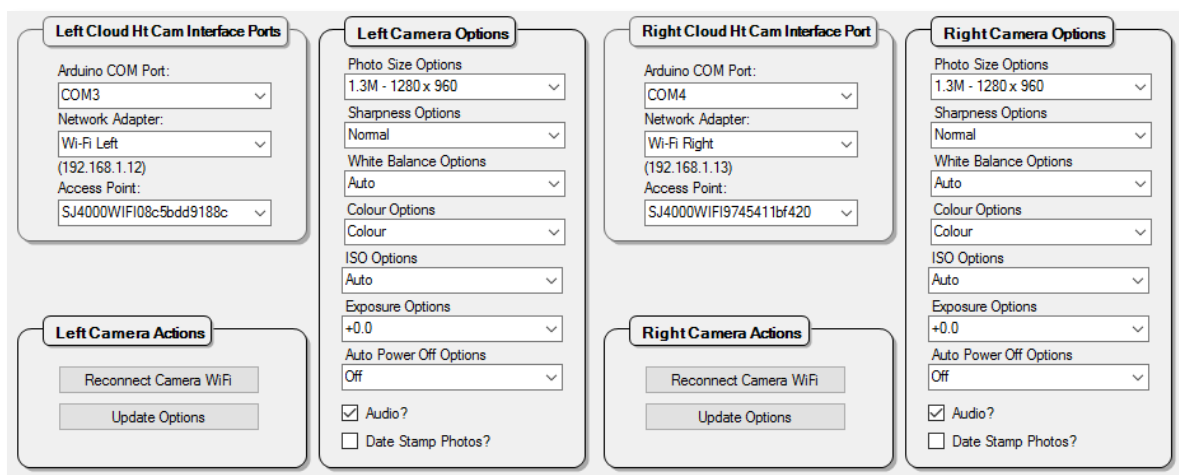
Max Cloud Height Difference: The maximum number of milliseconds between the image snapshot times for cloud height calculations, which if exceeded, renders the images invalid

Camera Photo Folder: The name of the folder on the cameras where the camera stores images

Web Response Timeout: The maximum number of milliseconds the controller will wait for responses from the camera after a request is sent

NepiSys Processor IP or Name: The IP or name of the computer on which the Processor is running

Cloud Ht.Cams | Left Camera and Right Camera



Left Cloud Ht Cam Interface Ports

Arduino COM Port:
COM3

Network Adapter:
Wi-Fi Left
(192.168.1.12)

Access Point:
SJ4000WIFI08c5bdd9188c

Left Camera Options

Photo Size Options
1.3M - 1280 x 960

Sharpness Options
Normal

White Balance Options
Auto

Colour Options
Colour

ISO Options
Auto

Exposure Options
+0.0

Auto Power Off Options
Off

☒ Audio?
☐ Date Stamp Photos?

Left Camera Actions

Reconnect Camera WiFi

Update Options

Right Cloud Ht Cam Interface Port

Arduino COM Port:
COM4

Network Adapter:
Wi-Fi Right
(192.168.1.13)

Access Point:
SJ4000WIFI9745411bf420

Right Camera Options

Photo Size Options
1.3M - 1280 x 960

Sharpness Options
Normal

White Balance Options
Auto

Colour Options
Colour

ISO Options
Auto

Exposure Options
+0.0

Auto Power Off Options
Off

☒ Audio?
☐ Date Stamp Photos?

Right Camera Actions

Reconnect Camera WiFi

Update Options

Arduino COM Port: The serial port to which the camera's Arduino box is connected

Network Adapter: The name of the network adapter to which the camera's WiFi Access Point connects

Access Point: The camera's WiFi Access Point name

[Reconnect Camera WiFi]: Reconnects the camera's WiFi connection

[Update Options]: Forces all the camera's options to update (i.e. a new query is sent to the camera)

Photo Size Options: Sets the camera's snapshot photo size

Sharpness Options: Sets the camera's photo sharpness options

White Balance Options: Sets the camera's photo white balance options

Colour Options: Sets the camera's photo colour options: Colour, Sepia, Black/White etc

ISO Options: Sets the camera's ISO options

Exposure Options: Sets the camera's exposure options

Auto Power Off Options: Sets the camera's auto power off options

Audio: Turns the audio on/off

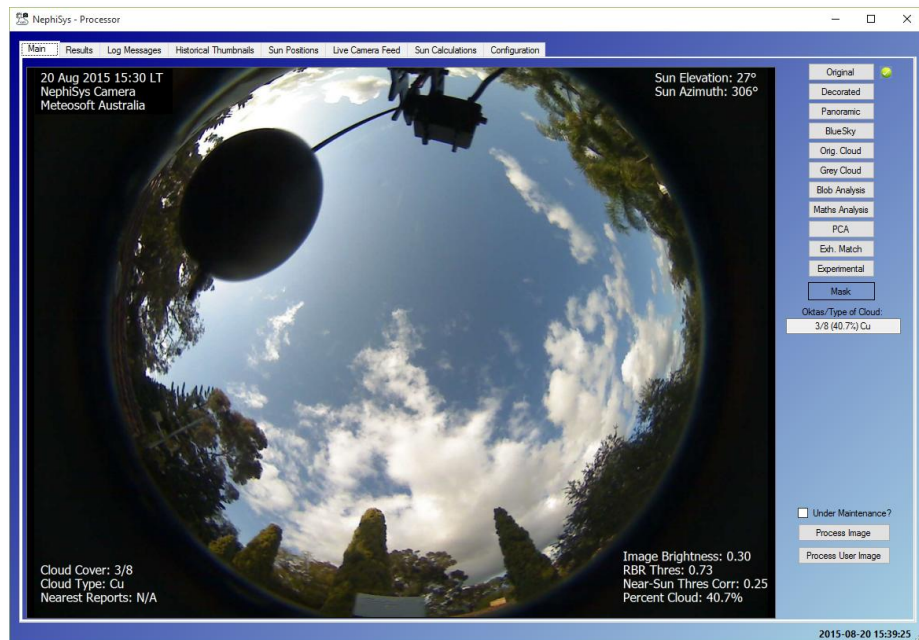
Date Stamp Photos: Turns the photo date stamping on/off

THE PROCESSOR SOFTWARE

The Processor software allows the user to view the current image in ten ways:

MAIN PAGE

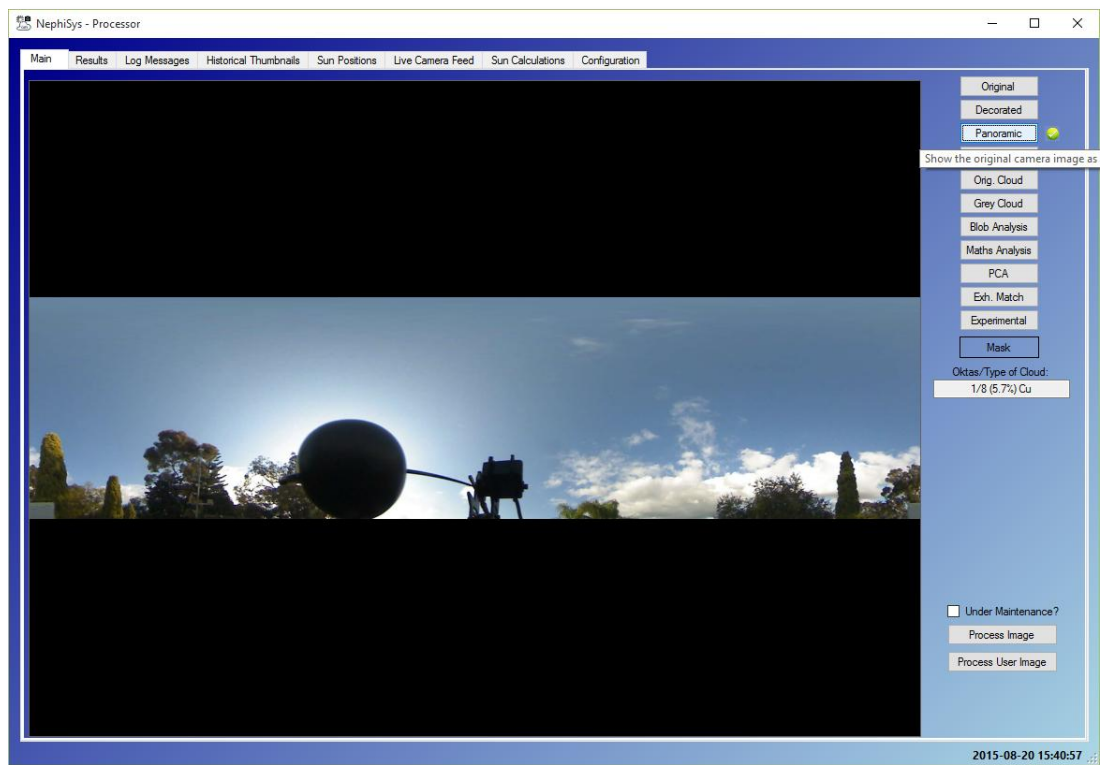
The **original image** from the camera (annotated with computed cloud amount, type and other useful information):



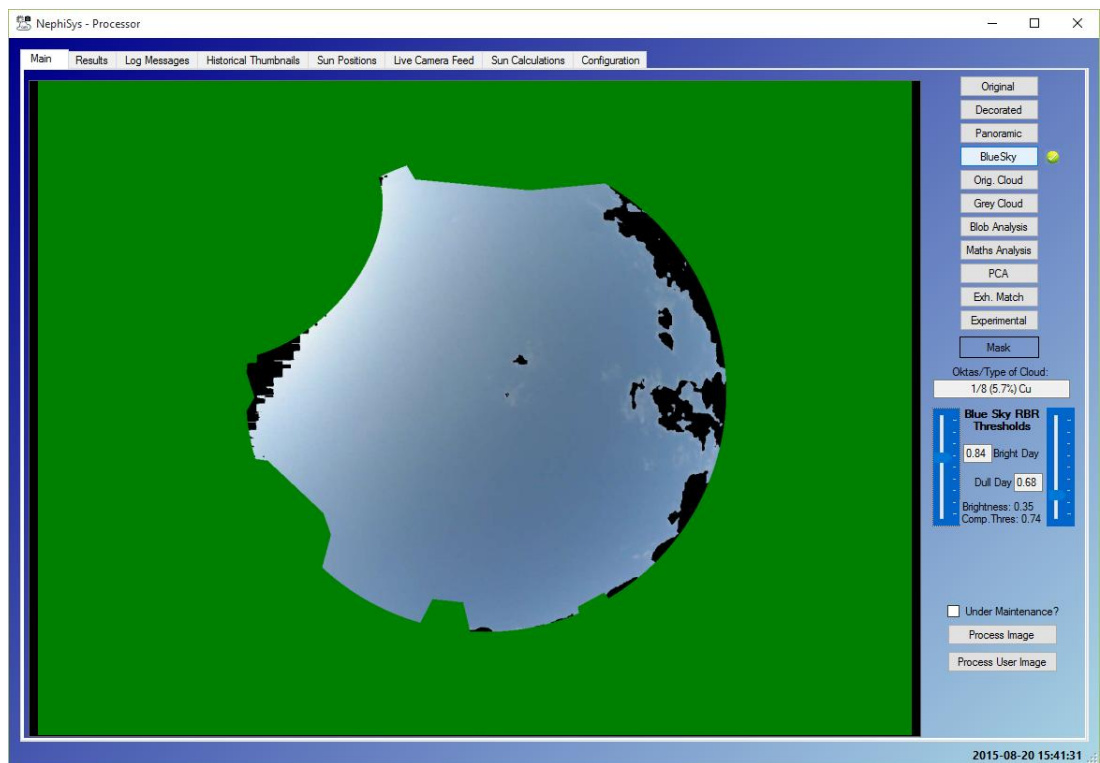
The **decorated image** with optional software heliosphere block (considered during processing), and any user requested information like hourly sun positions for the day (not processed):



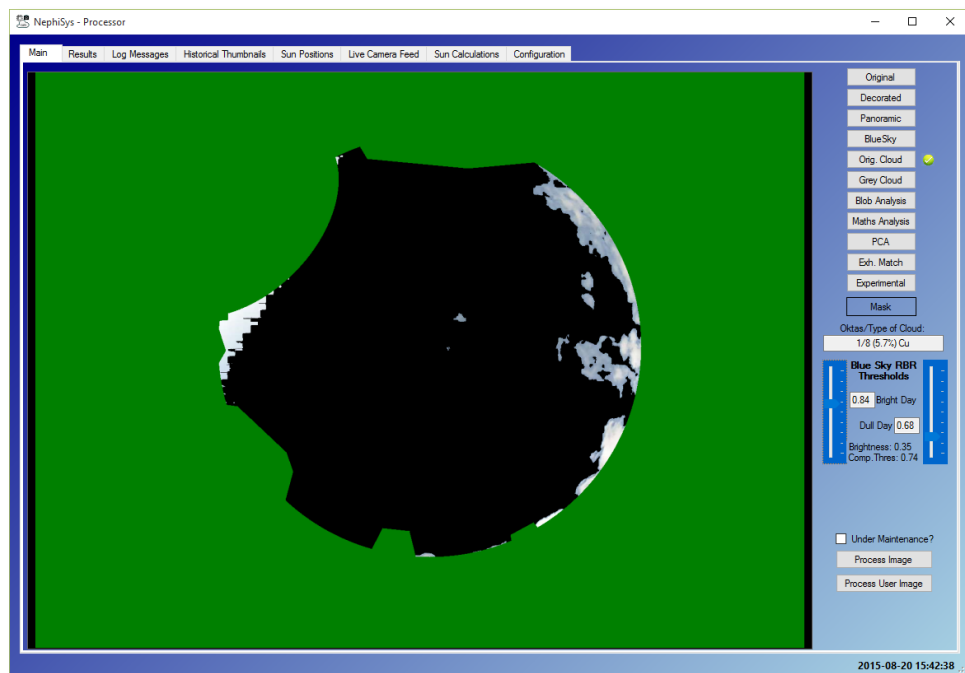
The **panoramic view** of the original fish-eye image:



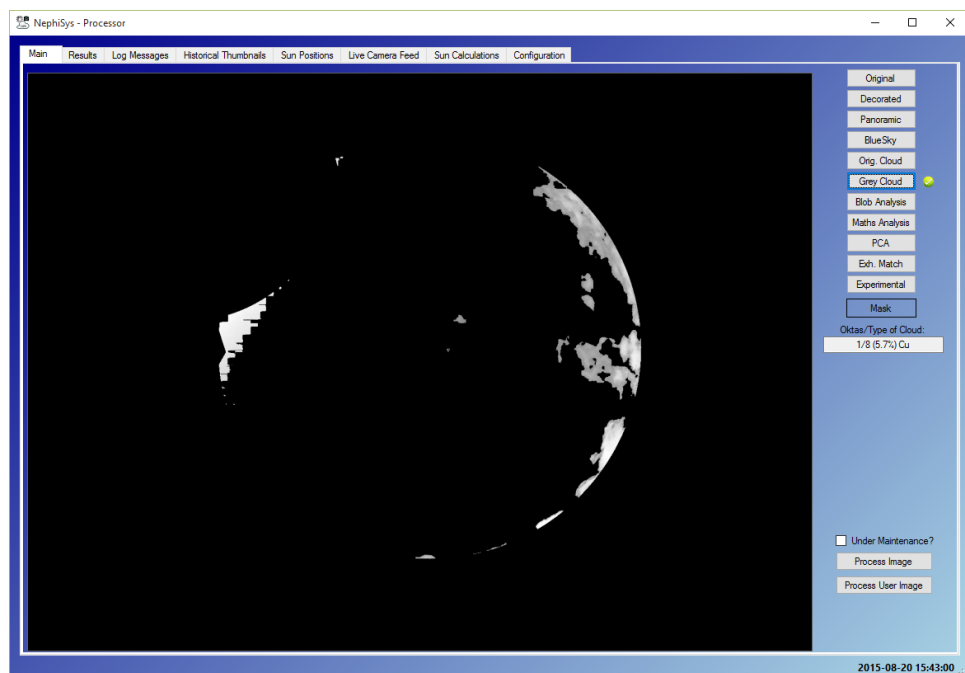
The processed **Blue-Sky image** showing blue sky only (cloud is black, everything else being green)



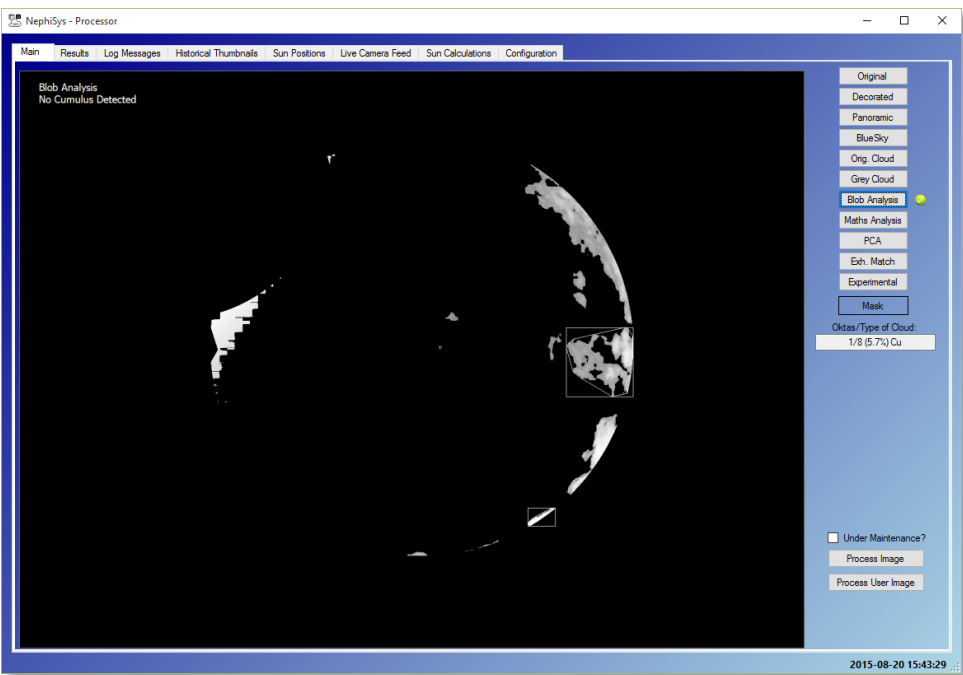
The processed **Cloud image** showing cloud only (blue-sky is black, everything else being green):



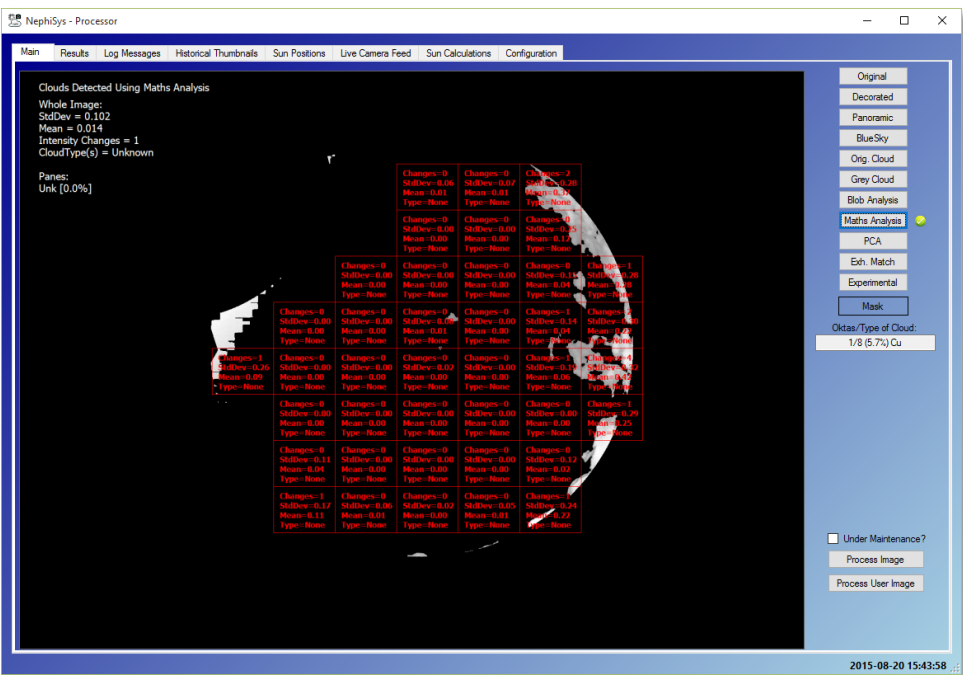
The **Grey Cloud image** used for processing (cloud is displayed, everything else is black):



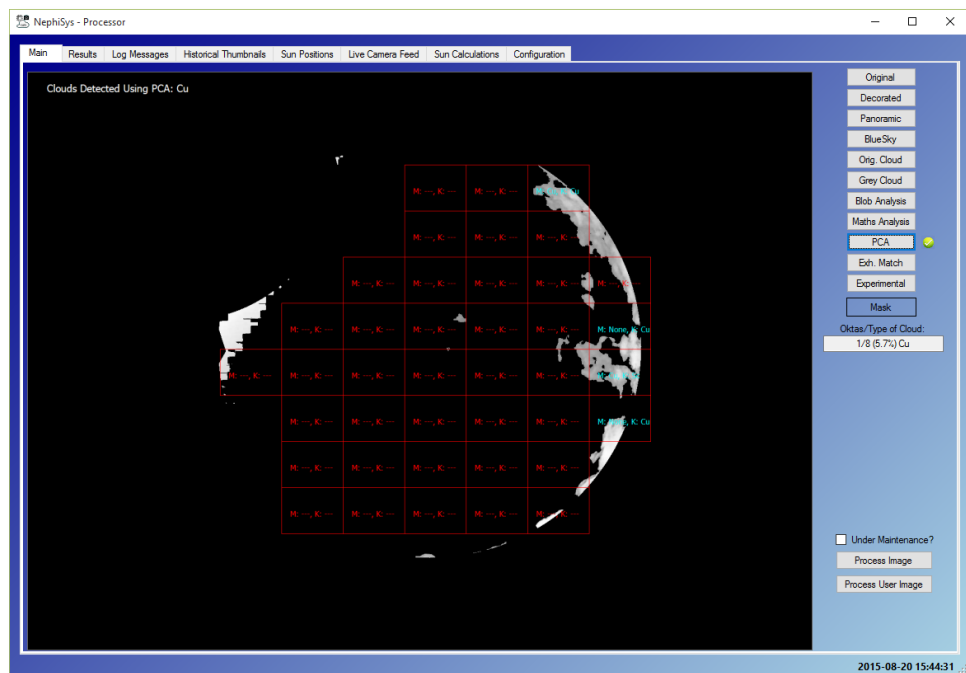
The processed **Blob Analysis** image, showing results of type processing:



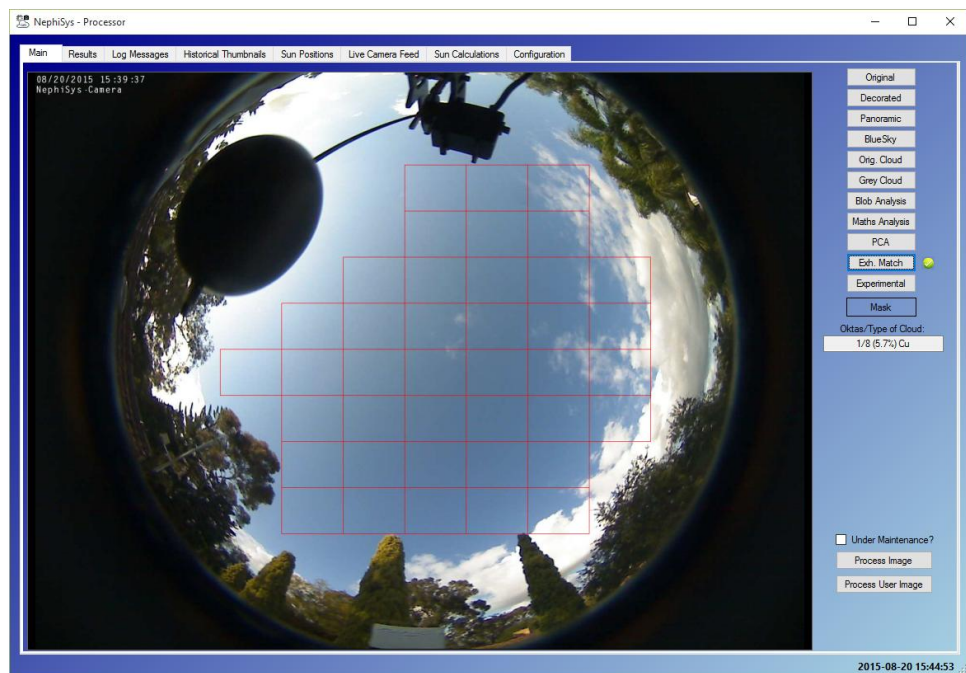
The processed **Statistical (Maths) Analysis** image, showing results of type processing:



The processed **Principle Component Analysis** image, showing results of type processing:



The **Exhaustive Match Analysis** image, showing the results of matching actual colour cloud panels with a set of database images (in this case, there were no matches):

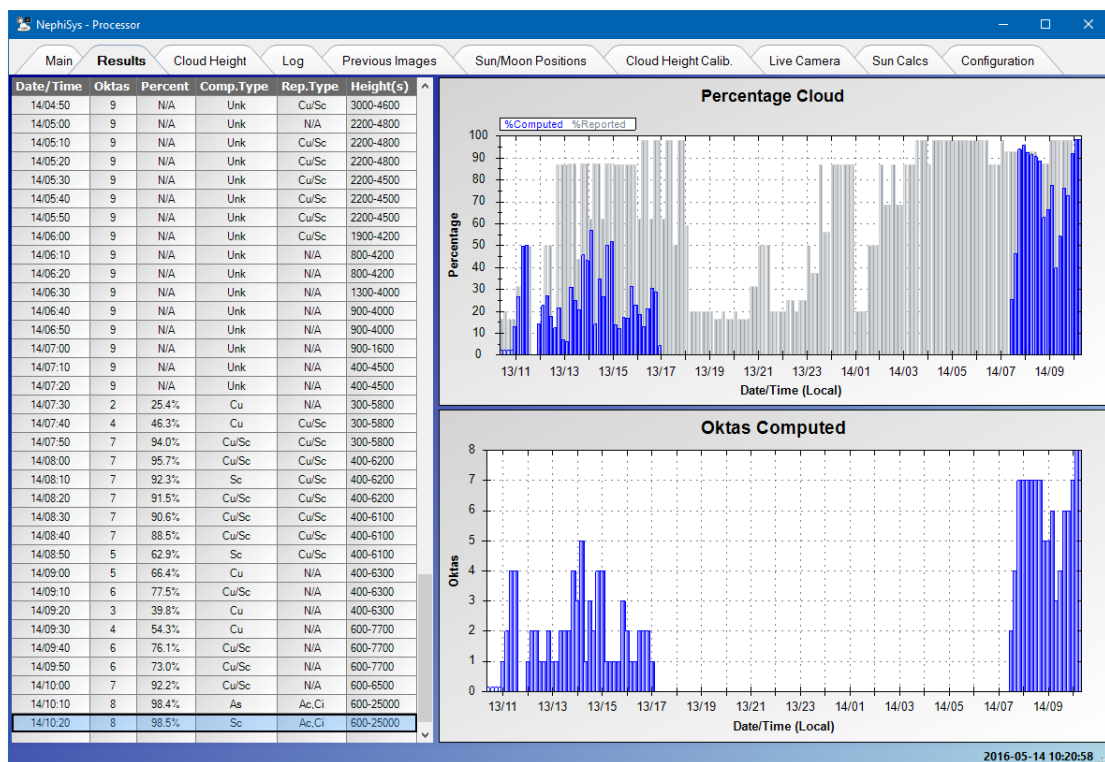


In the event of servo over-limit problem, the application background will turn red until the user acknowledges the error:

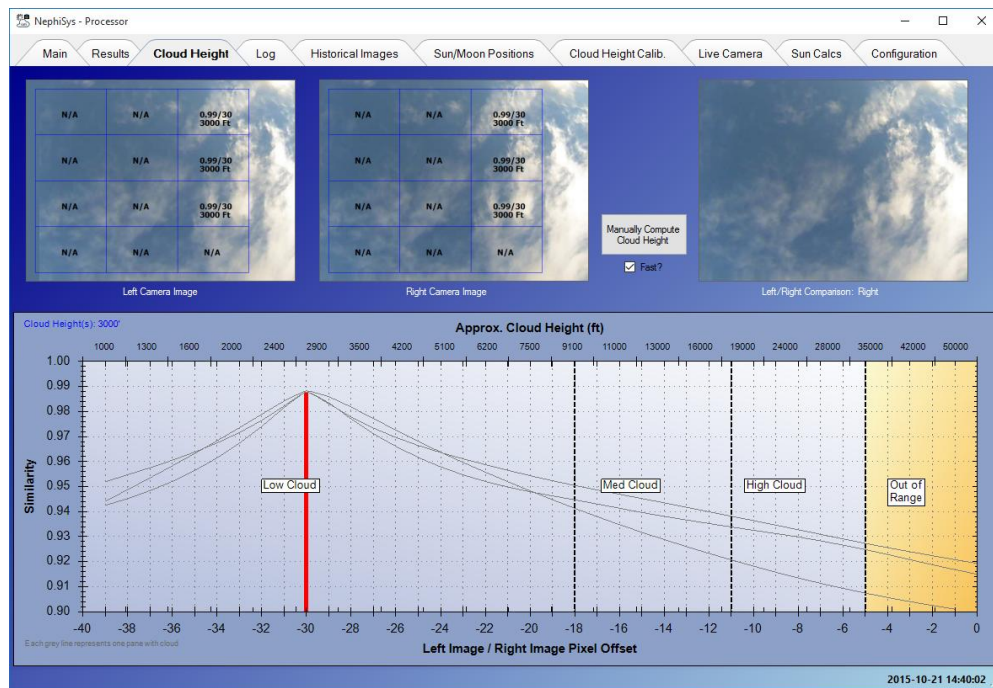


RESULTS PAGE

The detection results are displayed on the *Results* tab, together with reported cloud data from external sources. Blue denotes NephiSys results, and grey denotes results from external METAR/SYNOP sources:

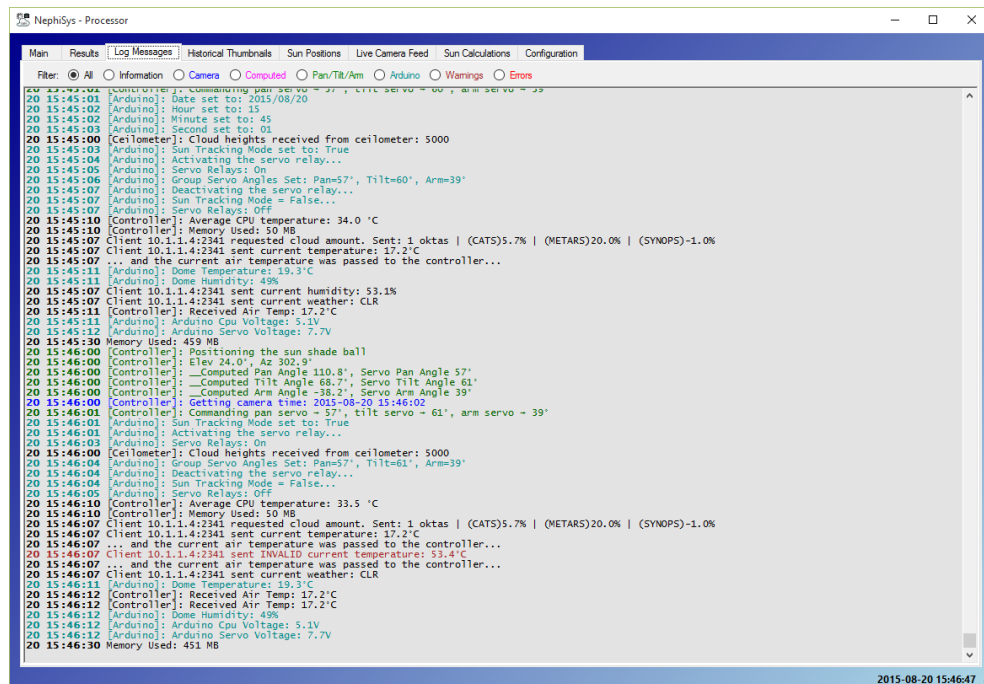


This view shows the results of the latest cloud height calculation:



LOG PAGE

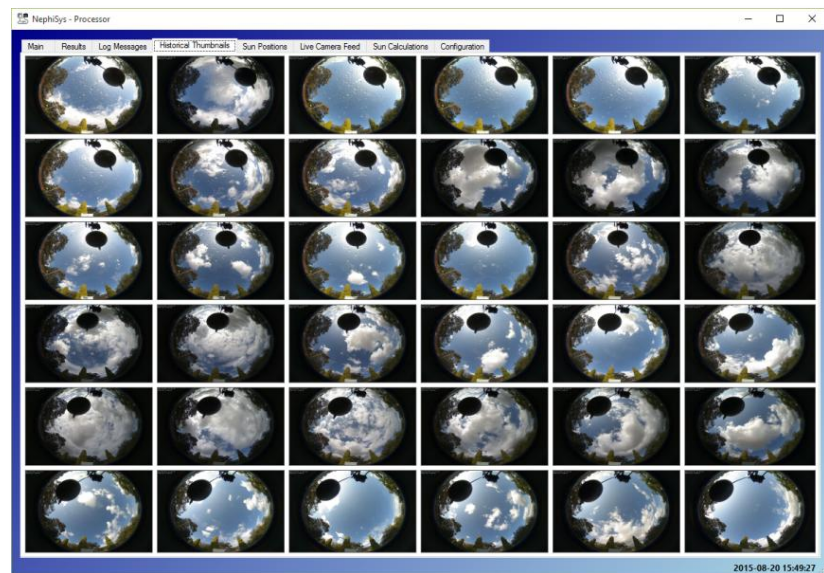
The software logs all significant events it performs, and all errors and exceptions, with date/time stamps, to a log page and to disk:



Black messages are general information, **magenta** messages are information regarding the computations, **blue** messages relate to camera operations, **green** messages relate to shade-ball servo motor operations, and **cyan** messages relate to Arduino operations. **Brown** messages are warnings, whereas **red** messages are errors or exceptions. Log messages from the **Controller** are identified by the inclusion of **[Controller]** or **[Controller]** or **[Controller]** in the message.

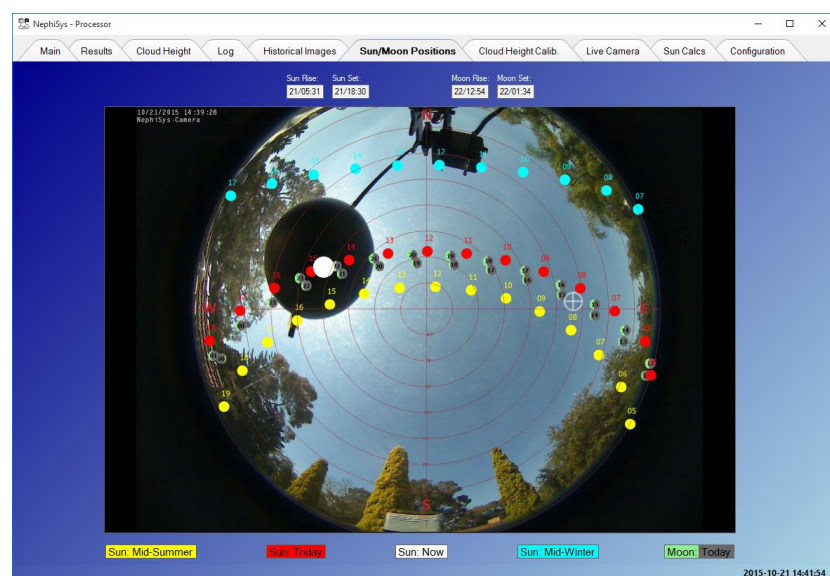
PREVIOUS IMAGES

The Previous Images Page displays all the previous images for the last 6 hours (36 images, 10 minutes apart):



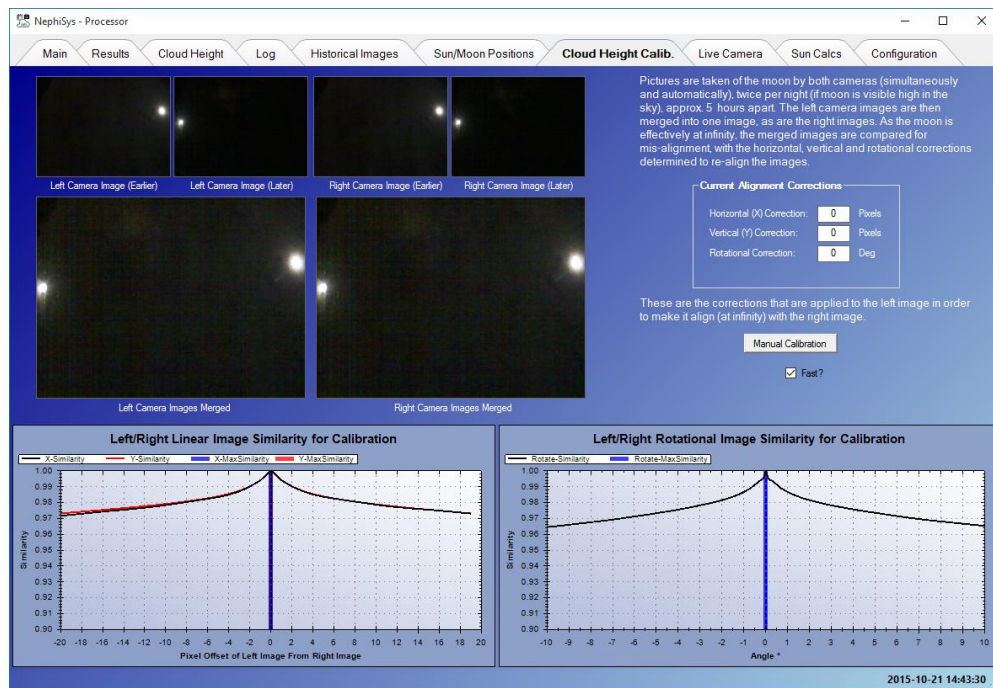
SUN/MOON POSITIONS PAGE

The Sun Positions Page shows the current position of the sun in relation to the shade-ball's position, the current sun track and the sun tracks for mid-summer and mid-winter:



CLOUD HEIGHT CALIBRATION

This view shows the results of the last cloud height calibration:



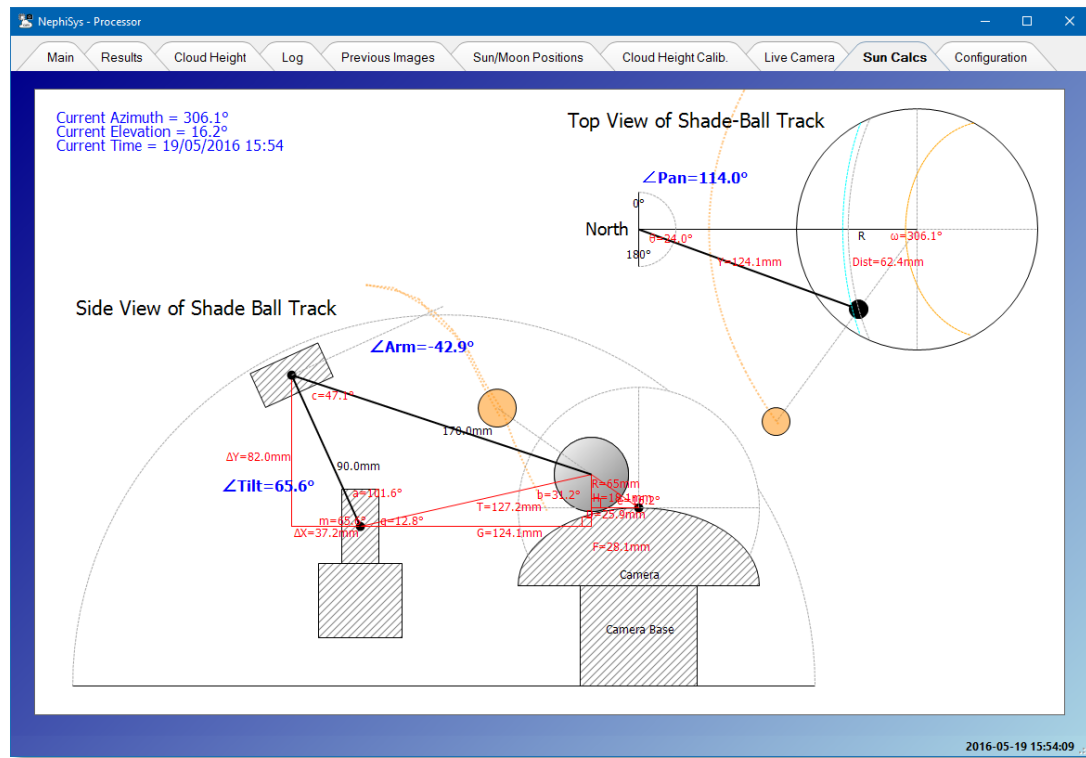
LIVE CAMERA PAGE

The Live Camera Feed page allows the user to view the current sky conditions in real time, when the shade cover is open:

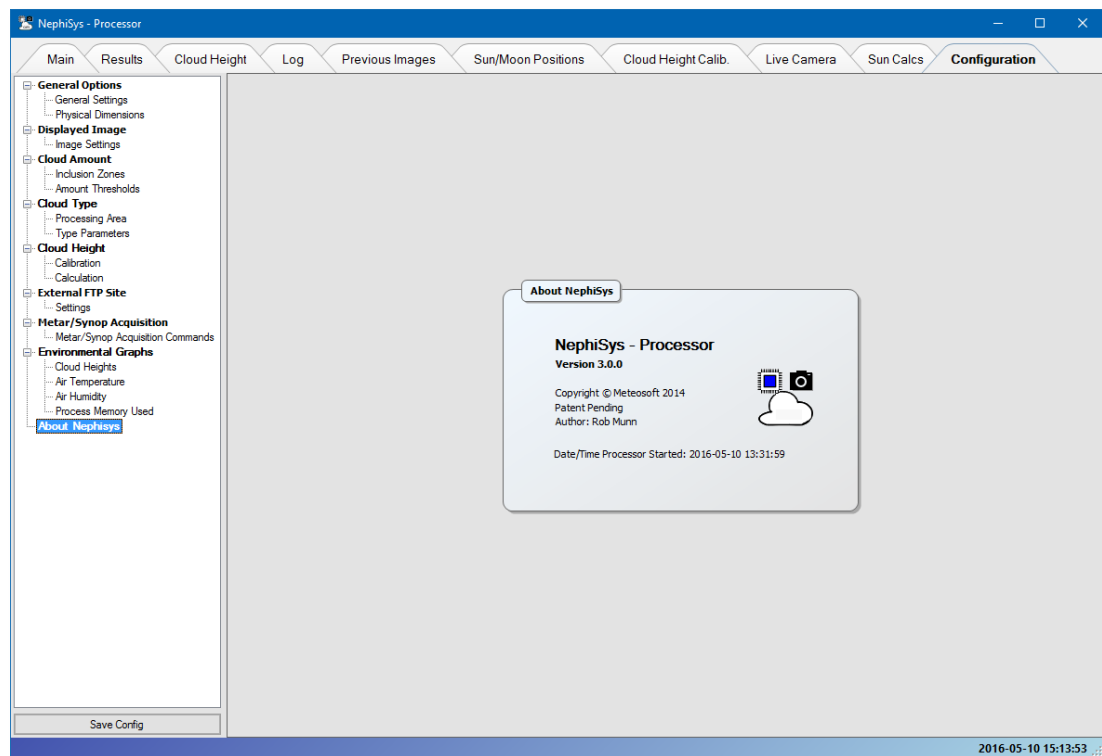


SUN CALCULATIONS PAGE

The Sun Calculations Page shows the calculations used to position the servo motors so that the shade-ball is moved to the correct position such that it casts a shadow from the sun over the camera lens:



CONFIGURATION PAGE



The Processor software allows the user to configure the following settings:

General Options | General Settings:

The screenshot shows a configuration window for the Processor software. It is divided into four main sections:

- General Settings:** Contains text boxes for Working Folder (D:\NephiSys\), Location Latitude (-32.064), Location Longitude (115.963), Nephisys Controller Computer IP or Name (CATDCONTROLLER), Cloud Height Controller Computer IP or Name (LAPTOP), and Remote Desktop Connection to Controller (C:\Users\Rob\Documents\CATS Controller.rdp). It also has a list of checkboxes for various display options: Show Hourly Sun Positions?, Show Image Disc Outline?, Show Heliosphere Block?, Process Even With Precipitation?, Temporarily Open Shade Cover On Controller?, Send To External FTP Site?, Send Cloud Data To External Client?, and Start/Stop Controller Remote Desktop. A Restart Nephisys Controller button is at the bottom.
- Live Video Parameters:** Contains text boxes for Camera User Name (admin), Camera Password (xxxxxxxx), Camera IP (10.1.1.6), and Camera Live Video Command (http://10.1.1.6/video/mjpg.cgi?profileid=1). A note below the command says: "Substitute the characters '{0}', without the quotes, for the command IP address!"
- SQL DB Connection String:** Contains a text box for SQL Database Connection String with the value Data Source=ROBSPC\SQLEXPRESS;Initial C=.
- White Reference Area in Main Image:** Contains four spinners for Top/Left X Pos (720), Top/Left Y Pos (1167), Ref Area Width (pixels) (120), and Ref Area Height (pixels) (30).

Working Folder: The root folder for the configuration and other working files. The images and log files for a particular day are stored in this folder under a new folder with the format: YYYY-MM-DD (e.g. 2014-07-11)

Location Latitude: The latitude of the NephiSys's location. Specify a negative number for the southern hemisphere.

Location Longitude: The longitude of the NephiSys's location. Specify a negative number for the western hemisphere (westwards from 0° to 180° through the USA)

Cloud Amount/Type Controller Computer IP or Name: The IP or name of the computer hosting the Cloud Amount/Type Controller

Cloud Height Controller Computer IP or Name: The IP or name of the computer hosting the Cloud Height Controller

Remote Desktop Connection to Cloud Amount/Type Controller: The path and remote desktop file to call to automatically open a remote desktop connection to the Cloud Amount/Type Controller

Show Hourly Sun Positions?: Displays the hourly sun positions on the final camera image, after processing

Show Image Disc Outline?: If checked, displays an outline of the image disc on the final camera image, after processing

Show Helioshere Block?: If checked, displays a software generated black ellipse over the helioshere (the sun and bright surrounds) before processing. The ellipse is considered an exclusion zone during processing

Process Even With Precipitation?: If checked, processing occurs even if it is raining. If unchecked, processing is suspended during rain, and sky is considered 'Obscured'

Temporarily Open Shade Cover On Controller? – If checked, the shade cover will open, and will stay open until unchecked

Send To External FTP Site?: Flags whether to send the image to an external entity set in *External FTP Site / Settings*

Send Cloud to External Client? – If checked, the cloud amount will be sent to all connected clients

Start/Stop Controller Remote Desktop? – If checked, starting/stopping the Processor will also start/stop the remote desktop connection to the Controller

[Restart Cloud Amount/Type Controller Computer]: Clicking this button will instruct the Cloud Amount/Type Controller's computer to perform a restart (shut-down and restart)

Camera User Name: The user name to log onto the camera's web server

Camera Password: The password to log onto the camera's web server

Camera IP: The camera's web server's IP address

Camera Live Video Command: The HTTP command sent to camera to request a video stream

SQL DB Connection String: The SQL connection string used to connect to the cloud database

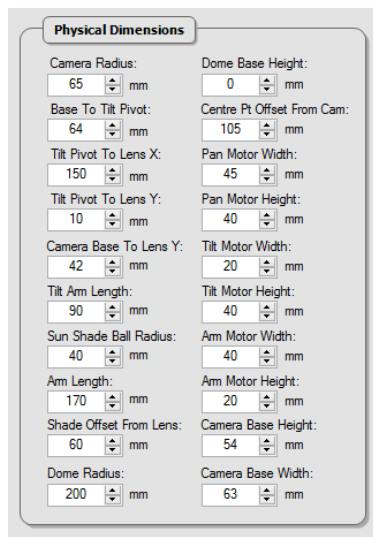
Top/Left X Pos: The X coordinate of the top/left white reference rectangle on the original image from which the white reference is calculated. The white reference allows the application to gauge the brightness of the image, and choose an appropriate RBR threshold value, between RBR for a bright day, or RBR for a dull day (see **Detection Thresholds|Cloud Amount** below)

Top/Left Y Pos: The Y coordinate of the top/left white reference rectangle on the original image from which the white reference is calculated

Ref Area Width (pixels): The width of the rectangle in the original image from which the white reference is calculated

Ref Area Height (pixels): The height of the rectangle in the original image from which the white reference is calculated

General Options | Physical Dimensions:

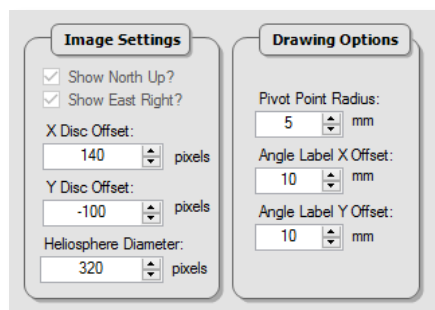


The 'Physical Dimensions' window contains 20 adjustable parameters in a two-column layout. Each parameter has a text label, a numeric input field, and a unit. The parameters are: Camera Radius (65 mm), Dome Base Height (0 mm), Base To Tilt Pivot (64 mm), Centre Pt Offset From Cam (105 mm), Tilt Pivot To Lens X (150 mm), Pan Motor Width (45 mm), Tilt Pivot To Lens Y (10 mm), Pan Motor Height (40 mm), Camera Base To Lens Y (42 mm), Tilt Motor Width (20 mm), Tilt Arm Length (90 mm), Tilt Motor Height (40 mm), Sun Shade Ball Radius (40 mm), Arm Motor Width (40 mm), Arm Length (170 mm), Arm Motor Height (20 mm), Shade Offset From Lens (60 mm), Camera Base Height (54 mm), Dome Radius (200 mm), and Camera Base Width (63 mm).

Parameter	Value	Unit
Camera Radius:	65	mm
Dome Base Height:	0	mm
Base To Tilt Pivot:	64	mm
Centre Pt Offset From Cam:	105	mm
Tilt Pivot To Lens X:	150	mm
Pan Motor Width:	45	mm
Tilt Pivot To Lens Y:	10	mm
Pan Motor Height:	40	mm
Camera Base To Lens Y:	42	mm
Tilt Motor Width:	20	mm
Tilt Arm Length:	90	mm
Tilt Motor Height:	40	mm
Sun Shade Ball Radius:	40	mm
Arm Motor Width:	40	mm
Arm Length:	170	mm
Arm Motor Height:	20	mm
Shade Offset From Lens:	60	mm
Camera Base Height:	54	mm
Dome Radius:	200	mm
Camera Base Width:	63	mm

These are the sizes of, and distances between, the various components in the sensor, used to position the shade-ball. These will only need to be adjusted if the hardware components are moved.

Displayed Image | Image Settings



The 'Image Settings' window includes checkboxes for 'Show North Up?' and 'Show East Right?', and numeric input fields for 'X Disc Offset' (140 pixels), 'Y Disc Offset' (-100 pixels), and 'Heliosphere Diameter' (320 pixels). The 'Drawing Options' window includes numeric input fields for 'Pivot Point Radius' (5 mm), 'Angle Label X Offset' (10 mm), and 'Angle Label Y Offset' (10 mm).

Parameter	Value	Unit
Show North Up?	<input checked="" type="checkbox"/>	
Show East Right?	<input checked="" type="checkbox"/>	
X Disc Offset:	140	pixels
Y Disc Offset:	-100	pixels
Heliosphere Diameter:	320	pixels
Pivot Point Radius:	5	mm
Angle Label X Offset:	10	mm
Angle Label Y Offset:	10	mm

Show North Up?: Indicates whether the top of the image is facing northwards. This option is set in Camera | User Settings - Flip Image?: *Flip image vertically?*

Show East Right?: - Indicates whether the image shows east to the right. This option is set in Camera | User Settings - Mirror Image?: *Mirror the image horizontally?*

X Disc Offset: - Gap in pixels between left edge of cloud image and leftmost edge of image disc (at vertical centre of disc)

Y Disc Offset: - Gap in pixels between top edge of picture and topmost edge of image disc (at horizontal centre of disc). Currently this value is negative as the camera cuts off a small section of the disk at the top of the image

Heliosphere Diameter: - The diameter of the optional (see General Options | General Settings - *Show Heliosphere Block?*) software heliosphere blocking ellipse in pixels. This ellipse is considered an exclusion zone during processing, that blocks the very bright surrounds of the sun.

Pivot Point Radius: - The radius of the circle on the Sun Calculations Page that signifies a servo pivot point in mm

Angle Label X Offset: - The horizontal offset used when labelling angles in mm on the Sun Calculations Page

Angle Label Y Offset: - The vertical offset used when labelling angles in mm on the Sun Calculations Page

Cloud Amount | Inclusion Zones



This page allows the user to add inclusion zones (as polygons) to the image, i.e. those parts of the image that are part of the sky and should be processed. These zones should exclude trees, structures, and/or image borders. If no inclusion zones are given, the whole image is considered an inclusion zone by default.

To draw an inclusion polygon, mouse click on an appropriate start point, then move the mouse to the next point and click again. A dashed line will display connecting the last point with the current position as the mouse is moved. Continue this process until the last point, when a double-click will complete the process.

Pressing the [ESC] key will abort the current drawing routine.

Click the  button to add this zone to the list:

Inclusion Zones:

Inclusion Zones
Start={X=626,Y=26}, End={X=626,Y=26}

Each inclusion zone (polygon) is listed with their start and end points.

To delete a specific inclusion zone, select that zone in the list and click **Delete Inclusion Zone**

To delete all inclusion zones, click: **Clear All Zones**

Cloud Amount | Amount Thresholds

Cloud Amount Thresholds

General

% Pseudo-Cloud Artifacts To Ignore: %
Humidity Limit: %

Percent Cloud For Overcast: %

Num Mins After Sun Rise and Before Sunset that Processing can occur:

Low Elevation Pixels to Ignore at Low Sun

☒ Ignore Low Elevation Pixels with Low Sun?

Elevation Below Which Pixels Are Ignored: °

Minutes After Sunrise/Before Sunset to Ignore Low Sun Pixels: Minutes

Thresholds

Bright Sky RBR Threshold:
Dull Sky RBR Threshold:

Near Sun RBR Thres. Corr:
Near Sun RBR Corr Radius:

Cloud RBR Threshold:
Min Cloud Luminance:

Brightness Adjust Factor:

Shade Cover Closed Detection

Test Point X:
Test Point Y:

Red:
Green:
Blue:

± Colour Component Margin:

% Pseudo-Cloud Artifacts To Ignore: - The percentage of artifacts that are misdetected as clouds on a clear day, that should be ignored, e.g. tree edges, internal reflection artifacts etc.

Humidity Limit: - The humidity (as given by an external AWS) above which the inside of the dome is deemed to be misty, obscuring the visible sky. With adequate internal heating or venting of dry air, this value can be set as 100%.

Percent Cloud For Overcast: - The percentage of cloud that demarcates a broken from overcast cloud cover, as the cloud cover is rarely computed as 100%.

Num Mins After Sun Rise and Before Sunset that Processing can occur: - Number of minutes after sun rise and before sunset that processing can occur. At sun rise and sun set, red hues can defeat the blue-sky and cloud detection algorithms, so processing can be delayed this number of minutes.

Ignore Low Elevation Pixels with Low Sun: When the sun has a low elevation, horizontal brightening can occur introducing errors into the cloud detection process. Set this checkbox to limit the amount of sky scanned to higher elevations only, during low sun conditions.

Elevation Below Which Pixels are Ignored: The elevation, during low sun conditions, below which image pixels are ignored, if checkbox “Ignore Low Elevation Pixels with Low Sun” is checked.

Minutes After Sunrise/Before Sunset to Ignore Low Sun Pixels: The period of low sun referred to above.

Bright Sky RBR Threshold: The Primary Red-Blue-Ratio $(1 + (R - B) / B)$ used to detect blue sky in bright conditions:

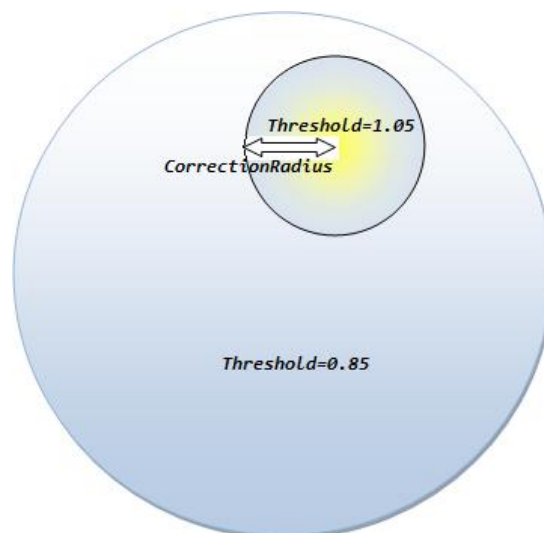
```
RedToBlueRatioOfPixel = (1.0 + (RedVal - BlueVal) / BlueVal)
If RedToBlueRatioOfPixel <= RBRThreshold (default 0.86)
    IsBlueSky = True
Else
    IsBlueSky = False
End If
```

Dull Sky RBR Threshold: The Primary Red-Blue-Ratio $(1 + (R - B) / B)$ used to detect blue sky in dull conditions:

Near Sun RBR Thres. Corr: - The Additive Red-Blue-Ratio $(1 + (R - B) / B)$ [**BlueSkyRBRThresholdCorrNearSun**] correction for pixels very close to sun

Near Sun RBR Corr Radius: - The distance [**CorrectionRadius**] in pixels from the sun position that the Blue Sky RBR ThresholdCorrection decreases to 0:

```
ClosenessToCentre = 1 - (Distance from the Sun Position / CorrectionRadius)
BlueSkyThreshold = BlueSkyRBRThreshold +
    BlueSkyRBRThresholdCorrNearSun * ClosenessToSun
```



Cloud RBR Threshold: - The Red-Blue-Ratio $(1 + (R - B) / B)$ [**CloudThreshold**] used to detect cloud

Min Cloud Luminence: - The minimum luminence [**MinCloudLuminence**] used to detect cloud

Secondary Blue Sky RBR Threshold: - The Secondary Red-Blue-Ratio $((B - R) / (B + R))$ [**CloudRBRThreshold2**] used to detect blue sky, between 0.6 and -0.4:

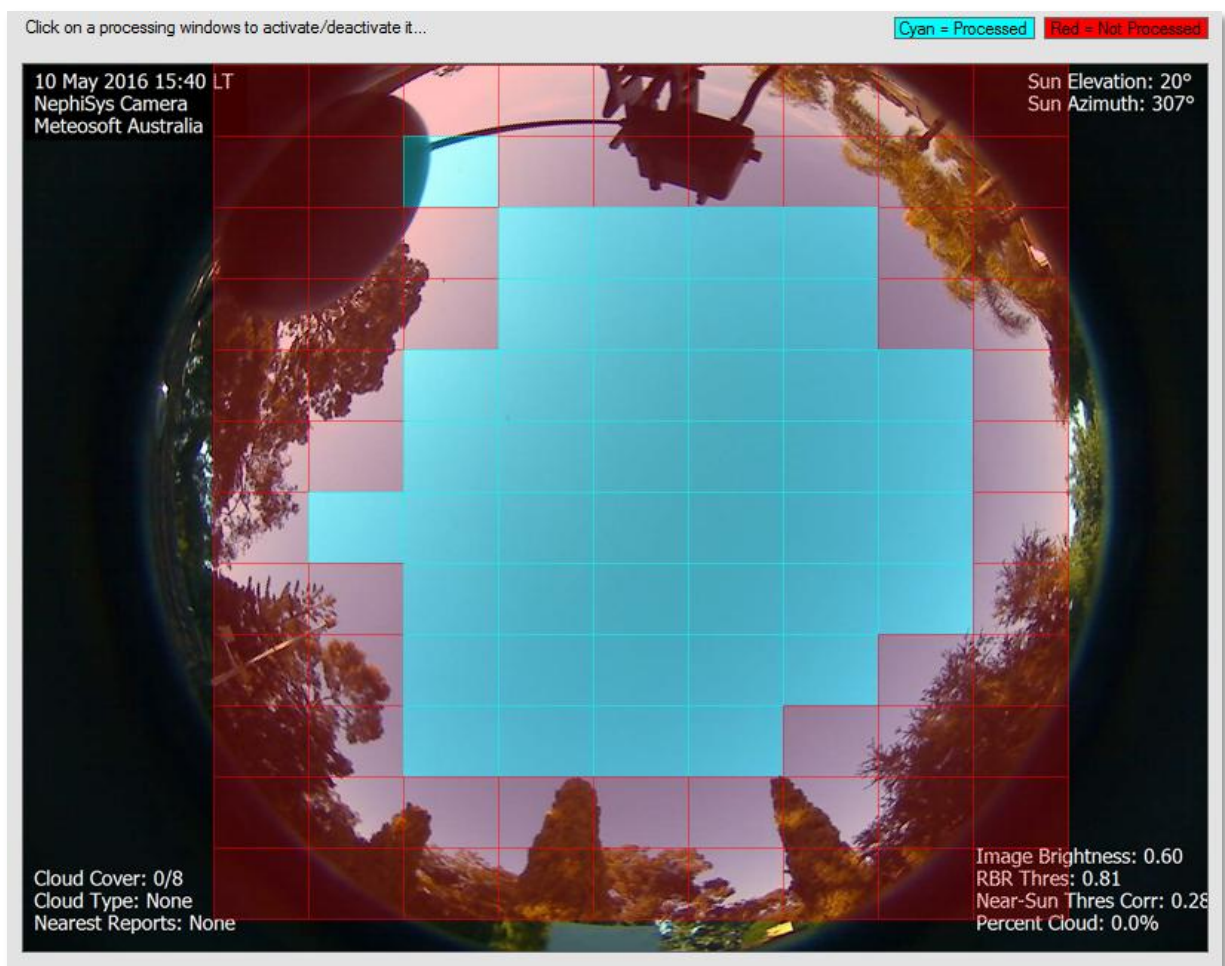
```
Luminence = (R + G + B) / 3
Type = Unknown
```

```

If RBR1 < RBRThreshold
    Type = BlueSky
Else If RBR1 <= CloudThreshold AND Luminance > MinCloudLuminence
    Type = Cloud
Else
    If RBR2 < 0.6 AND RBR2 > -0.4
        If RBR2 > CloudRBRThreshold2
            Type = BlueSky
        Else
            Type = Cloud
        Endif
    Else
        Type = Unknown
    Endif
Endif

```

Cloud Type | Processing Area



Two of the four algorithms used to determine cloud type analyse small panes from within the whole-sky image. To preclude any pane's cloud images containing artificial edges (e.g. horizon, trees, or other obstacles) from being processed, the user may select which panes should be processed. Clicking on any pane will turn the processing flag on (cyan) / off (red). Each NephiSys location will present a different sky-line, and will need to be configured for that location.

Cloud Type | Type Parameters

Cloud Type Detection Algorithm Weightings

	Weighting for 1st Cloud Detected	Weighting for 2nd Cloud Detected	Algorithm Descriptions:
Algorithm 1:	<input type="text" value="2"/>	<input type="text" value="2"/>	Detect Clouds on Whole Cloud Bitmap Using Statistical Analysis
Algorithm 2:	<input type="text" value="10"/>	<input type="text" value="5"/>	Detect Clouds in Small Windows Using Statistical Analysis
Algorithm 3:	<input type="text" value="4"/>	<input type="text" value="2"/>	Detect Clouds on Whole Cloud Bitmap Using Blob Analysis
Algorithm 4:	<input type="text" value="6"/>	<input type="text" value="3"/>	Detect Clouds in Small Windows Using Principle Component Analysis

Cloud Type Detection Statistical Parameters

Cloud Type	Min Inten.Chng	Max Inten.Chng	StdDev Min	StdDev Max	Cloud Level
Cu	1	15	0.09	0.60	Low
Sc	1	6	0.03	0.07	Low
Ac	2	26	0.05	0.30	Med
As	1	8	0.01	0.05	Med
Cs	1	8	0.01	0.05	High
Cc	20	40	0.01	0.04	High

Add Cloud Stats
Delete Cloud Stats

Principle Component Parameters

Nearest Neighbours:

Minimum Mean

Cloud Level Upper Thresholds

Low Level (ft)

Medium Level (ft)

Cloud Type Detection Algorithm Weightings allow the user to adjust the significance of each of the four cloud type detection algorithms. The results of each algorithm are multiplied by the weighting factor, and the final result is deduced using the maximum count.

Algorithm X: - The weighting factor to be applied to the primary cloud detected, and to the secondary cloud detected. The primary cloud usually has a higher weighting than the secondary.

Cloud Type Detection Statistical Parameters permit the Statistical Analysis algorithm to recognise cloud forms/shapes using mathematical techniques. Sample grey scale cloud images were analysed for various cloud types and both intensity changes (below to above the mean, and vice versa, per 128 pixels) and the standard deviation were found to demarcate non-stratiform cloud types significantly, whilst the stratiform cloud required an indication of cloud height to better determine the cloud type.

[Add Cloud Stats] – Adds a new row to the cloud stats table allowing the user to edit the fields.

[Delete Cloud Stats] – Deletes the currently selected row.

Principle Component Parameters – The Principle Component Analysis uses two methods to identify the cloud type in a grey scale image pane (128 x 96 pixels) against a database of known cloud types of the same size, namely the Minimum Mean Distance Classifier and the K-Nearest Neighbours Classifier. The latter method accepts a K parameter which indicates how many neighbouring pixels to match.

Nearest Neighbours – The K-Nearest Neighbours Classifier accepts a K parameter which indicates how many neighbouring pixels to look at when looking for a match. The smaller the K value, the better the match.

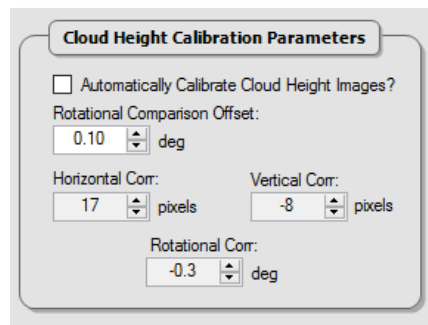
Minimum Mean – This value does **NOT** relate to the Minimum Mean Distance Classifier, but indicates to the algorithm that a pane with a grey scale image that has a mean value of less than this Minimum Mean, should NOT be processed, as it does not contain enough information.

Cloud Level Upper Thresholds – The cloud type detection routine accepts cloud heights from external sources (Synops, Metars, Ceilometers etc) and groups the heights into three levels, demarcated by the given upper thresholds.

Low Level (ft) – The upper threshold for low level clouds. The lower threshold is assumed to be 0 ft.

Medium Level (ft) – The upper threshold for medium level clouds. The lower threshold is assumed to be the low level upper threshold. **High Level** clouds are assumed to lie above the upper threshold for medium level clouds.

Cloud Height | Calibration



The screenshot shows a dialog box titled "Cloud Height Calibration Parameters". It contains the following controls:

- A checkbox labeled "Automatically Calibrate Cloud Height Images?" which is currently unchecked.
- A label "Rotational Comparison Offset:" followed by a text input field containing "0.10" and a unit dropdown menu set to "deg".
- A label "Horizontal Corr:" followed by a text input field containing "17" and a unit dropdown menu set to "pixels".
- A label "Vertical Corr:" followed by a text input field containing "-8" and a unit dropdown menu set to "pixels".
- A label "Rotational Corr:" followed by a text input field containing "-0.3" and a unit dropdown menu set to "deg".

The cloud height calibration may be performed manually (using the Cloud Height Calibrator) or automatically. The left and right cameras are commanded to take an image of the sky at night when a bright moon is visible high in the sky to the east, and again a little later (say 4 hours) when the moon is visible high in the sky to the west. The earlier and later left camera images are merged into one, as are the earlier and later right camera images. These two merged images are then compared by shifting the left merged image over the right merged image, horizontally, pixel by pixel, looking at the similarity of both images. The horizontal correction needed to align the images is the pixel offset of the maximum similarity. The same process is performed vertically. Finally the left merged image is rotated from minus 10 to plus 10 degrees and at each degree the similarity is computed. The angle with the maximum similarity is the rotational correction.

The Controller ensures that cloud height images are taken every 10 minutes (the reporting period), regardless of whether the moon is visible or not.

Automatically Calibrate Cloud Height Images: - If checked, the Processor will determine whether the moon will be visible above 50° elevation in the east, and a little later in the west. If true, then the Processor will copy over the 'early' moon images from the left and right cameras at the appropriate time. Later, the 'later' moon images are copied and the calibration process proceeds. The user is presented with the calibration result via a

message box, where he/she may confirm that the corrections are valid and may be used to calculate the cloud height. If confirmed, the **Horizontal Corr**, the **Vertical Corr** and the **Rotational Corr** boxes will be updated.

Horizontal Corr: - The horizontal correction in pixels

Vertical Corr: - The vertical correction in pixels

Rotational Corr: - The rotational correction in degrees

Cloud Height | Calculation

The screenshot shows a software interface for cloud height calculation. It is divided into two main sections: 'Cloud Height Calculation Parameters' and 'Cloud Height Pane Dimensions'.

Cloud Height Calculation Parameters:

- ☒ Automatically Calculate Cloud Height?
- The cloud height is directly proportional to the pixel offset of a cloud on two images taken at the same time, and spaced far apart, and can be represented by an equation.
- Set the equation parameters A and B, for the equation:

$$Y = A * \text{Exp}(B * X)$$
 where Y = Cloud Height, and X = Offset of (same) cloud shapes, the height of which is being determined.
- A: B:
- Cloud Height Upper Limit:

Cloud Height Calculation Sensitivity:

- High Sensitivity: Low Sensitivity:

Cloud Height Pane Dimensions:

The images of the sky directly above NephiSys are partitioned into multiple panes (default 3 across x 4 down = 12 panes), and each image pane is compared to the equivalent pane on the other image to determine the cloud height in that pane.

- Horiz. Pane Offset: Vert. Pane Offset:
- Num Panes Across: Num Panes Down:
- Pane Width: Pane Height:
- Num Pixel Steps:
- Cyan = Processed
Red = Not Processed

Below the parameters is a 3x4 grid of panes. The bottom row of panes is highlighted in red, indicating they are not processed. The top three rows are cyan, indicating they are processed. A small image of a cloud is visible in the bottom-left pane.

Click on a processing pane to activate/deactivate it...

The cloud height calculation is performed by applying any calibration offsets to the left cloud image. The left and right images are then divided into smaller panes (default 12). Each pane on the left image is then shifted horizontally over the equivalent pane in the right image, pixel by pixel from 0 to 10 (default) pixels. The offset with the maximum similarity is then used (variable X) in an equation of the form $Y = A * \text{Exp}(B * X)$, to compute the approximate cloud height, in feet (variable Y). The constants A and B were determined by observation. The higher the cloud, the less accurate the cloud height calculation becomes, so an upper limit is placed on the result (default: 50,000 ft).

Automatically Calculate Cloud Height: - If checked, the Processor will attempt to calculate the cloud height every 10 minutes (reporting period). It may fail if a) there is no cloud is visible within the image area, b) there is cloud, but there are no defined edges to the cloud, or c) the computed cloud height is greater than the "Cloud Height Upper Limit".

A: The constant A in the equation $Y = A * \text{Exp}(B * X)$.

B: The constant **B** in the equation $Y = A * \text{Exp}(B * X)$.

Cloud Height Upper Limit: The height in feet above which the cloud height calculation is deemed to be too inaccurate.

Horiz. Pane Offset: The offset in pixels of the top/left pane from the left edge.

Vert. Pane Offset: The offset in pixels of the top/left pane from the top edge.

Num Panes Across: The number of panes across the cloud height image (i.e. number of columns).

Num Panes Down: The number of panes down the cloud height image (i.e. number of rows).

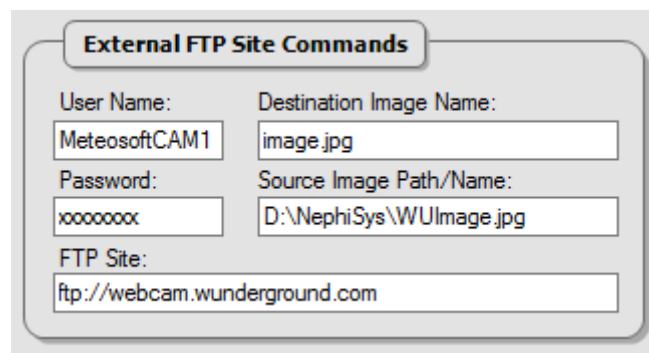
Pane Width: The width in pixels of each pane.

Pane Height: The height in pixels of each pane.

Num Pixel Steps: The number of pixel steps to be performed when shifting the left image over the right image looking for the maximum similarity.

Each pane may be activated or deactivated for processing by clicking on that pane. A pane with a cyan hue will be processed, whereas a pane with a red hue will be ignored, when computing the cloud heights.

External FTP Site | Settings



External FTP Site Commands

User Name:	Destination Image Name:
<input type="text" value="MeteosoftCAM1"/>	<input type="text" value="image.jpg"/>
Password:	Source Image Path/Name:
<input type="text" value="xxxxxxxx"/>	<input type="text" value="D:\NephiSys\WUImage.jpg"/>
FTP Site:	
<input type="text" value="ftp://webcam.wunderground.com"/>	

The original image (imprinted with the computed cloud amount and type) may be optionally sent to an external FTP site (see **General Options | General Settings**: Send To External FTP Site? above). In the example above, the image is being sent to Weather Underground.

User Name: - The user name to log onto the external FTP server

Password: - The password to log onto the external FTP server

Destination Image Name: - The external FTP destination image name

Source Image Path/Name: - The external FTP source image path/name

FTP Site: - The external FTP server's address

Metar/Synop Acquisition | Metar/Synop Acquisition Commands

Metar Acquisition Commands

Metar Acquisition Commands	ICAO Codes
http://en.allmetsat.com/metar-af/australia-oceania.php?icao=YPPH	YPPH
http://en.allmetsat.com/metar-af/australia-oceania.php?icao=YPJT	YPJT

[Add Cmd](#) [Delete Cmd](#) [Add Code](#) [Delete Code](#)

Synop Acquisition Commands

Synop Acquisition Commands	WMO Codes
http://weather.gladstonefamily.net/site/YPPH	94610
http://weather.gladstonefamily.net/site/YPJT	94609

[Add Cmd](#) [Delete Cmd](#) [Add Code](#) [Delete Code](#)

NephiSys is able to request Metar and Synop information from external sources using HTTP GET commands, in order to extract any cloud types. The cloud types are used ONLY to verify the accuracy of NephiSys's cloud type identification.

To add an HTTP GET command for either Metar or Synop, click on the [Add Cmd](#) button, then click on the new command entry that reads: [Add command here...](#) Edit the text with the appropriate command. To delete a command, click the button [Delete Cmd](#).

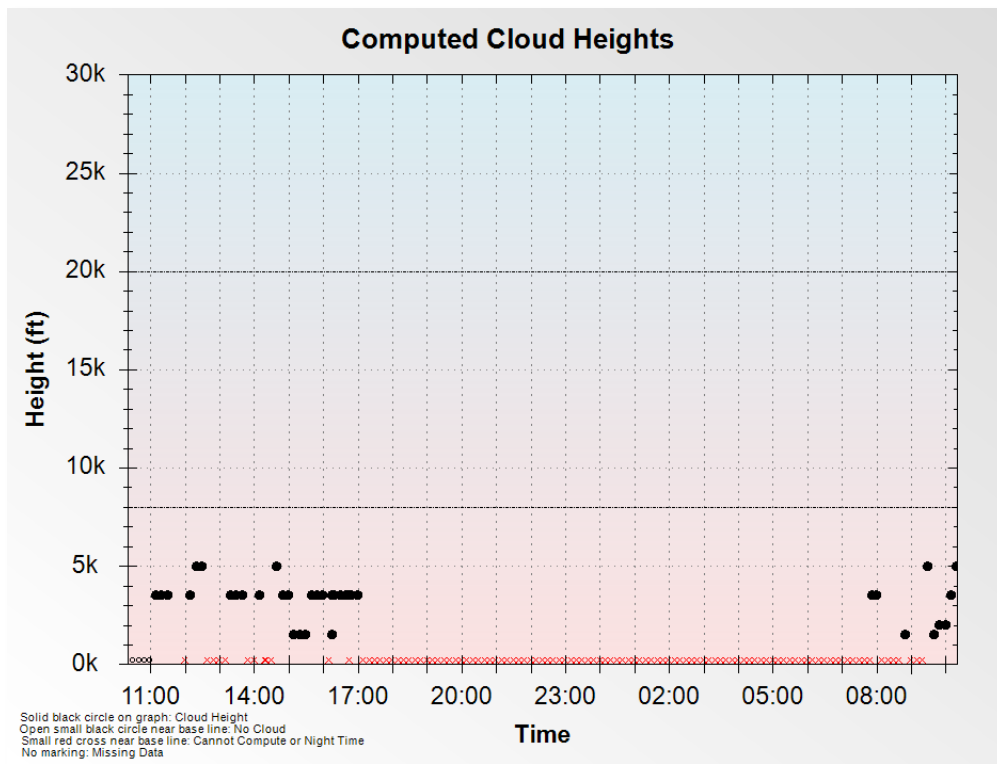
The ICAO or WMO codes are used to find the Metar or Synop message in the returned HTTP text. In the example above, a space is appended to the code so that the actual message can be identified, e.g.

[94610]41675 22416 10172 20027 30119 40143 51016 70200 82800, or

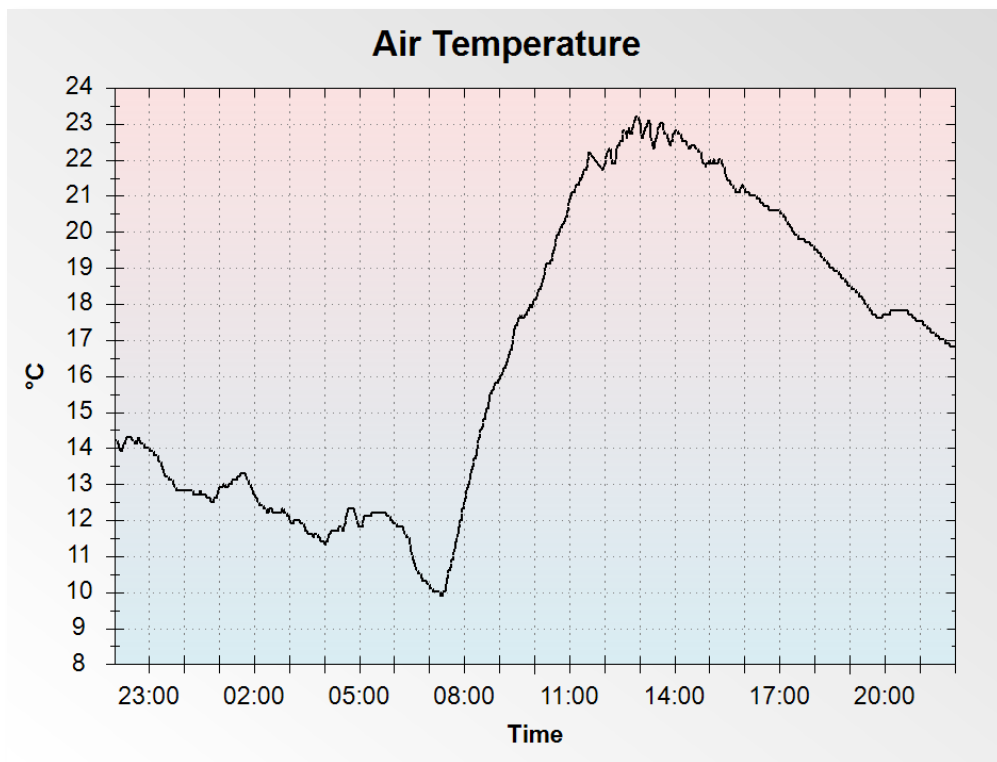
[YPPH]40600Z 25017KT 9999 VCSH FEW045 17/04 Q1013 NOSIG

To add an ICAO or WMO code for the associated command, click on [Add Code](#) then edit the new entry [Add code...](#) . To delete a code, click on [Delete Code](#).

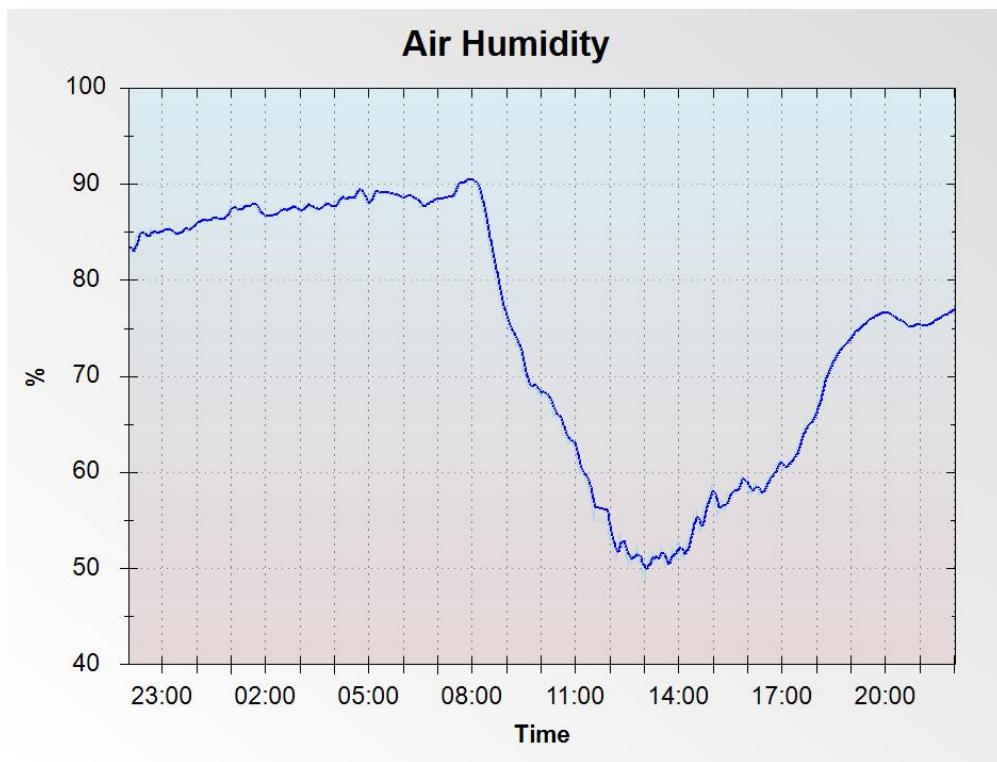
Environmental Graphs | Cloud Heights



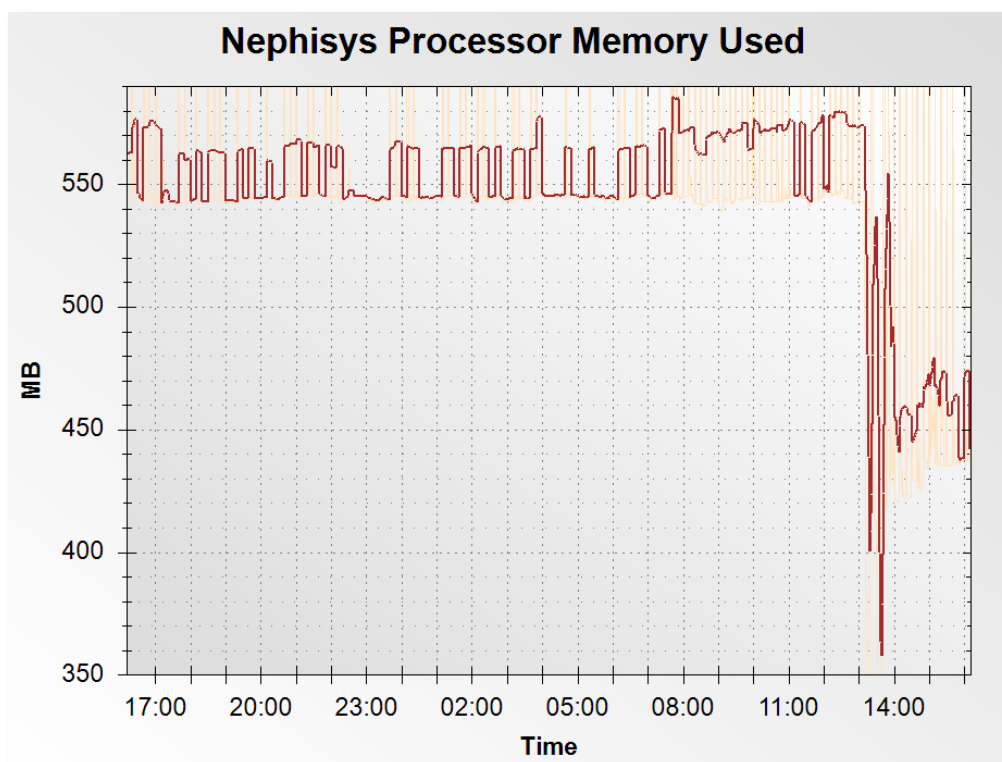
Environmental Graphs | Air Temperature



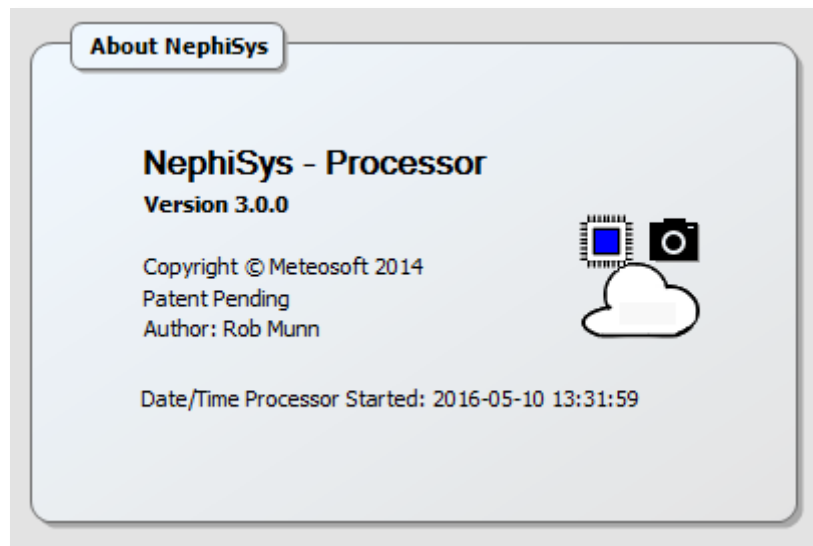
Environmental Graphs | Air Temperature



Environmental Graphs | Process Memory Used



About NephiSys Processor



UTILITIES

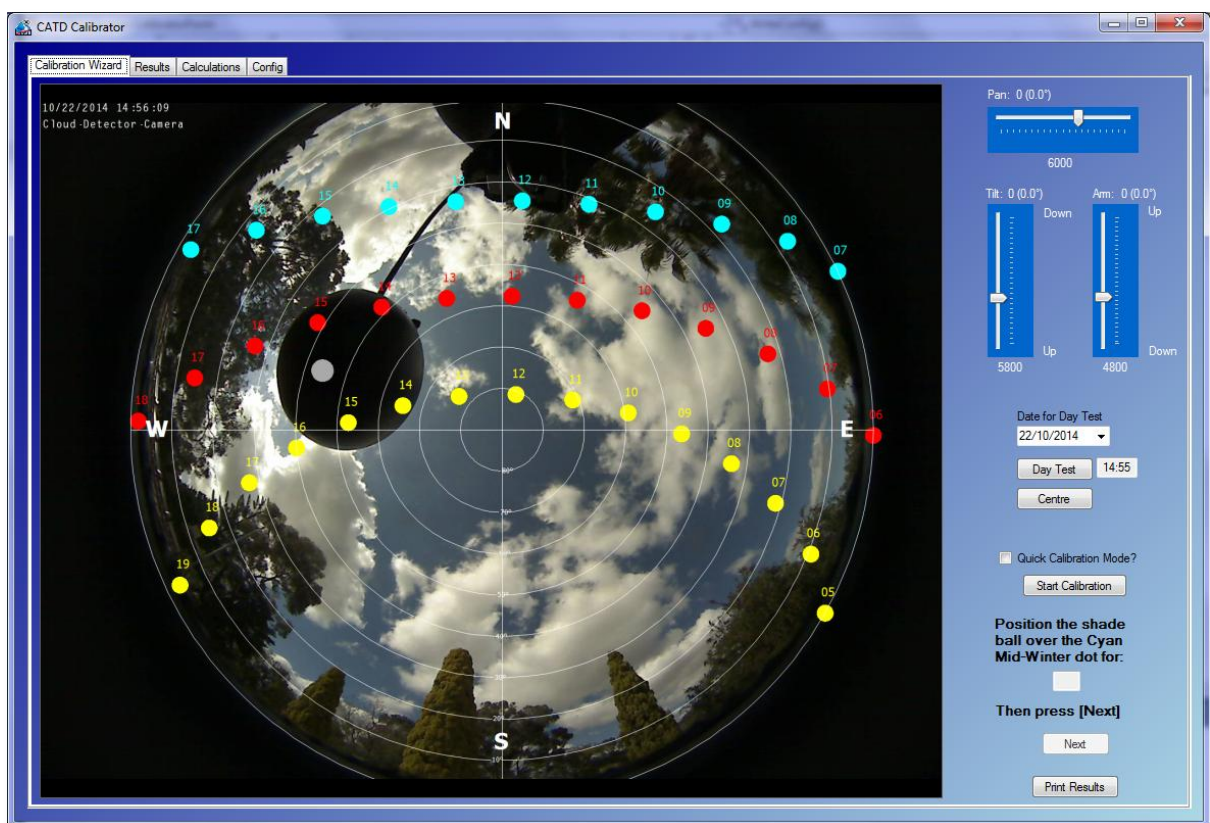
NEPHISYS CALIBRATOR

The NephiSys Calibrator is a utility that allow the user to calibrate the pan/tilt/arm servo motors' servo angles against the astronomical angles. The Calibrator instructs the user to position the shade-ball for each hour for three different sun tracks, mid-winter, mid-spring and mid-summer. A quick option is also available, which requires the user to position the shade-ball for every four hours, for the same three tracks. The resulting data (astronomical servo angle vs servo angle) is regressed linearly, and a solution for each servo is calculated in the form $\text{ServoAngle} = \text{SLOPE} * \text{AstroAngle} + \text{Y_INTERCEPT}$.

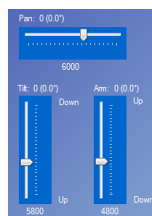
Four tabbed pages are available.

THE CALIBRATION WIZARD

The Wizard page presents a live video stream of the current sky, with the hourly sun positions marked for mid-winter (cyan), mid-spring (red) and mid-summer (yellow), so that the user can adjust the shade-ball position and immediately see the result.

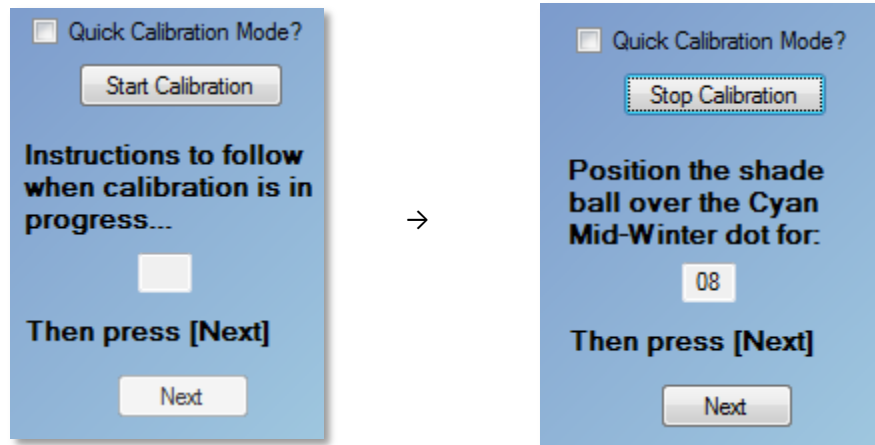


The user positions the shade-ball using the track-bar controls:



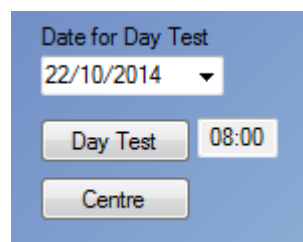
such that the ball covers the hourly dot evenly. The ball must also be 65mm (or as specified in the processor's configuration, **General Options | Physical Dimensions | Shade Offset from Lens**) from the camera lens.

The calibration starts when the user clicks **[Start Calibration]**.



Follow the wizard's instructions "Position the shade ball over the Cyan Mid-Winter dot for:" the hour specified. Once the shade-ball is positioned correctly, click **[Next]**. If the Quick Calibration Mode is checked, then the hour over which the shade-ball must be positioned increments by four, otherwise the user must position the ball every hour between sunrise and sunset. When the user clicks **[Next]**, the software reads the current servo angle for each servo. When the last hour of the mid-Summer track has completed, the calibrator immediately computes the results, that can be viewed on the **Results** tabbed page.

The user can check that the results are reasonable, by performing a day test for any day of the year. The user selects the date and clicks **[Day Test]** to start the test, The shade-ball will move to the sunrise position for the given date (wait for 5 seconds), and then slowly move along a track that the sun would take, in 15 minute steps (each step taking ½ second). The hour and minute of the position is shown next to the **[Day Test]** button.



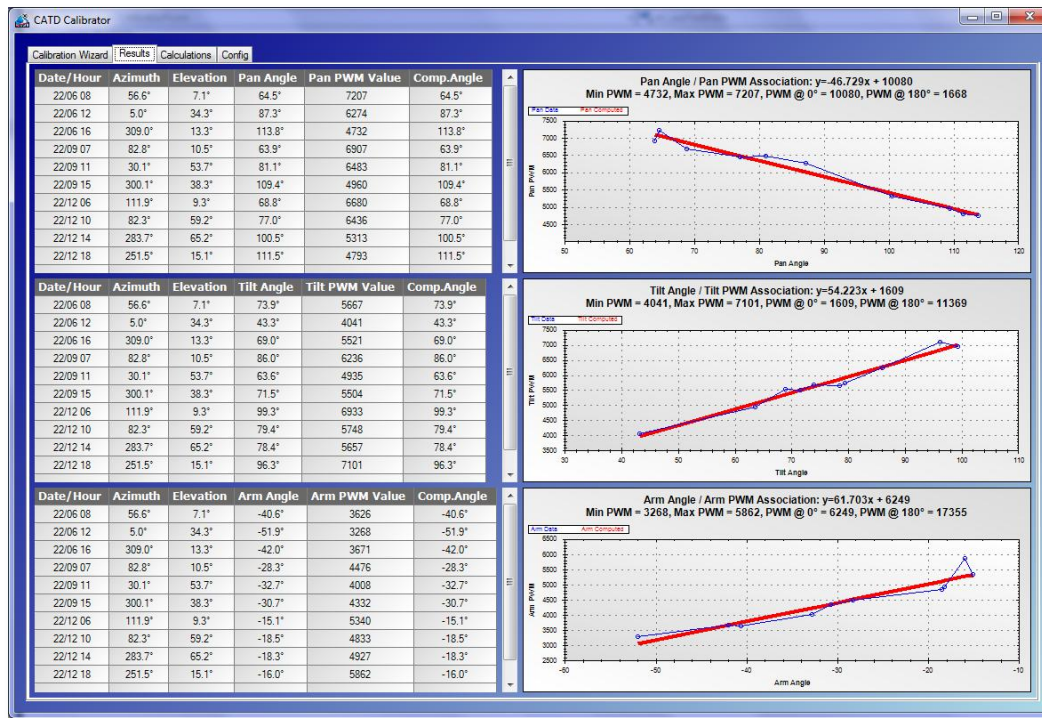
The user may centre the shade-ball directly over the camera lens at any time by pressing **[Centre]**.

The results may also be printed or print previewed by clicking the button **Print Results**.

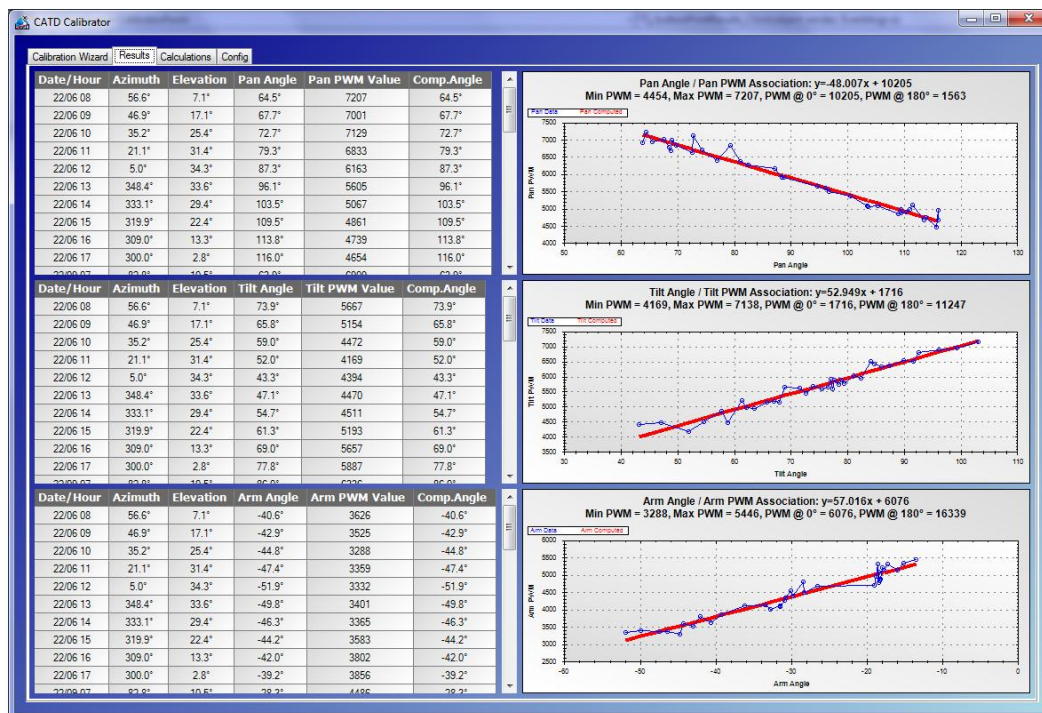
THE RESULTS PAGE

The results of the calibration can be seen on the Results page.

The Quick Calibration Mode would produce results similar to:

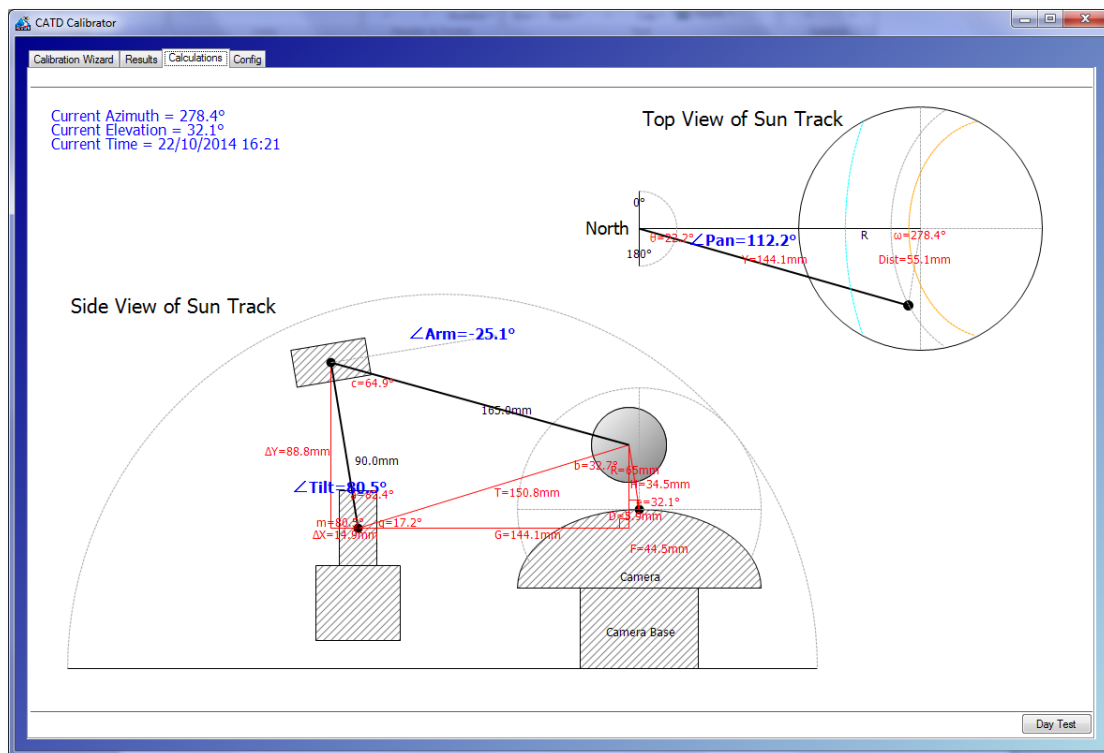


A full calibration would produce results similar to:



THE CALCULATIONS PAGE

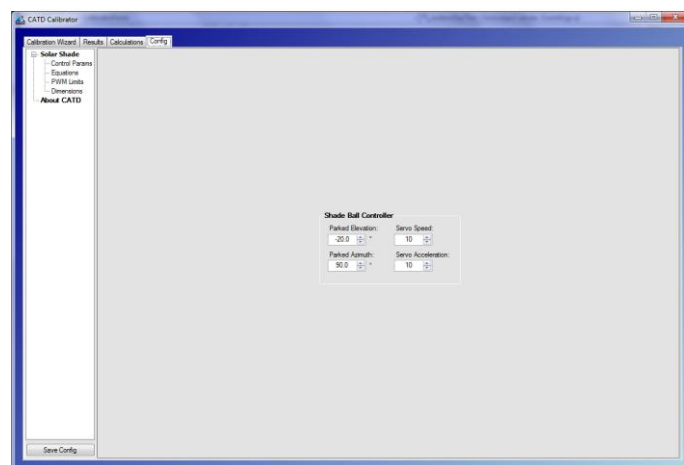
The Calculations Page shows the calculations used to position the servo motors so that the shade-ball in moved to the correct position:



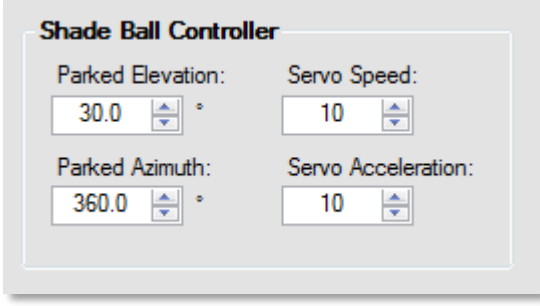
The user may physically move the shade-ball in the Top View of Sun Track, by clicking the mouse within the 'camera' circle and dragging the black dot to a preferred position. All shade-ball movements are reflected in this drawing.

THE CONFIG PAGE

The NephiSys Controller's configuration is mirrored here, and can be updated.



Solar Shade | Control Params



Shade Ball Controller

Parked Elevation: 30.0 °

Servo Speed: 10

Parked Azimuth: 360.0 °

Servo Acceleration: 10

Parked Elevation: The shade-ball's parked elevation

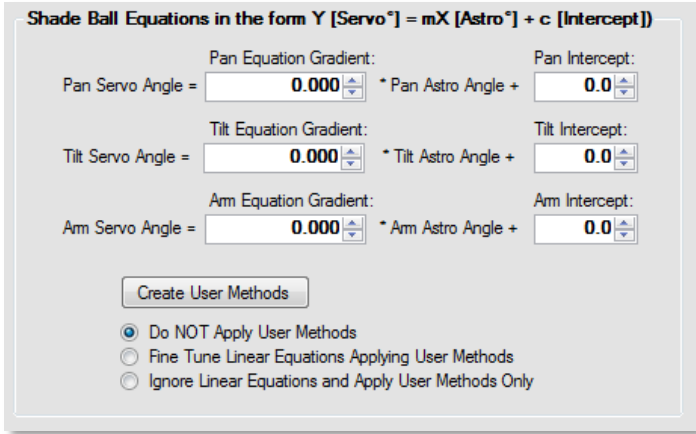
Parked Azimuth: The shade-ball's parked azimuth

Servo Speed: The pan/tilt controller servo speed - 0 = Fastest, 1 = Very slow, 127 = Very fast

Servo Acceleration: The pan/tilt controller servo acceleration - 0 = Fastest, 1 = Very slow, 127 = Very fast

Solar Shade Controller | Conversion Equations

The pan, tilt and arm astronomical angles are computed such that the shade-ball casts a shadow over the camera lens. However, the Arduino microcontroller requires the corresponding servo angles in order to position the shade-ball. Each servo produces an (almost) linear relationship between angle and servo angle. The linear regression equations that relate the astronomical angle to the servo angle are entered here:



Shade Ball Equations in the form $Y \text{ [Servo]} = mX \text{ [Astro]} + c \text{ [Intercept]}$

Pan Equation Gradient: 0.000 * Pan Astro Angle + Pan Intercept: 0.0

Tilt Equation Gradient: 0.000 * Tilt Astro Angle + Tilt Intercept: 0.0

Arm Equation Gradient: 0.000 * Arm Astro Angle + Arm Intercept: 0.0

Create User Methods

☒ Do NOT Apply User Methods

☐ Fine Tune Linear Equations Applying User Methods

☐ Ignore Linear Equations and Apply User Methods Only

Pan Equation Gradient: The astronomical Pan angle to servo angle equation's gradient

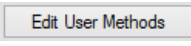
Pan Intercept: The astronomical Pan angle to servo angle equation's Y intercept

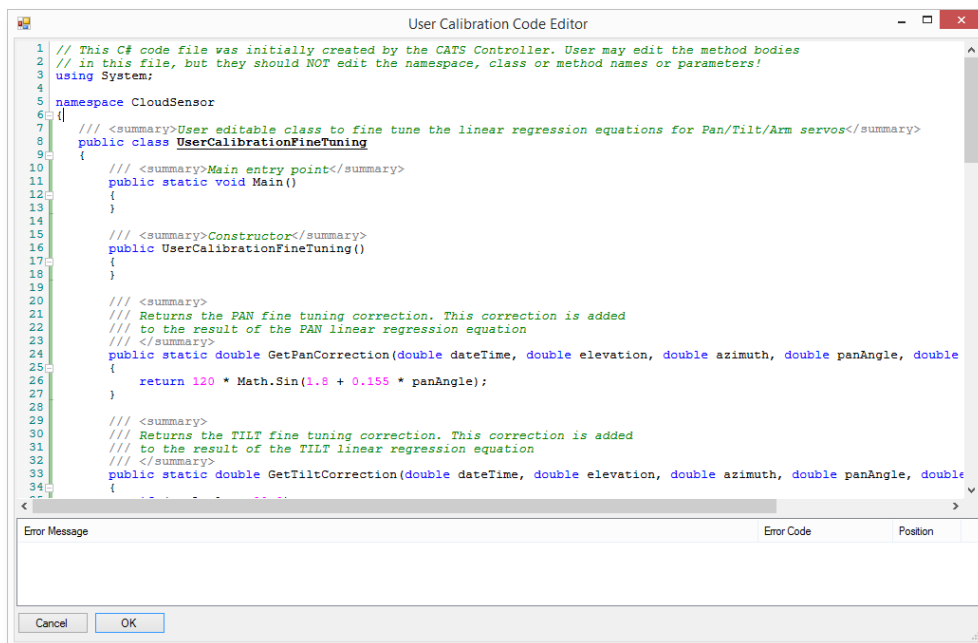
Tilt Equation Gradient: The astronomical Tilt angle to servo angle equation's gradient

Tilt Intercept: The astronomical Tilt angle to servo angle equation's Y intercept

Arm Equation Gradient: The astronomical Arm angle to servo angle equation's gradient

Arm Intercept: The astronomical Arm angle to servo angle equation's Y intercept

Clicking , presents a code editor in which the **advanced** user may enter C# scripting code to fine tune the servo angles (effectively adjusting the y-intercept of the linear equations), depending on the hemisphere (east/west), in the **GetXXXCorrection()** methods:



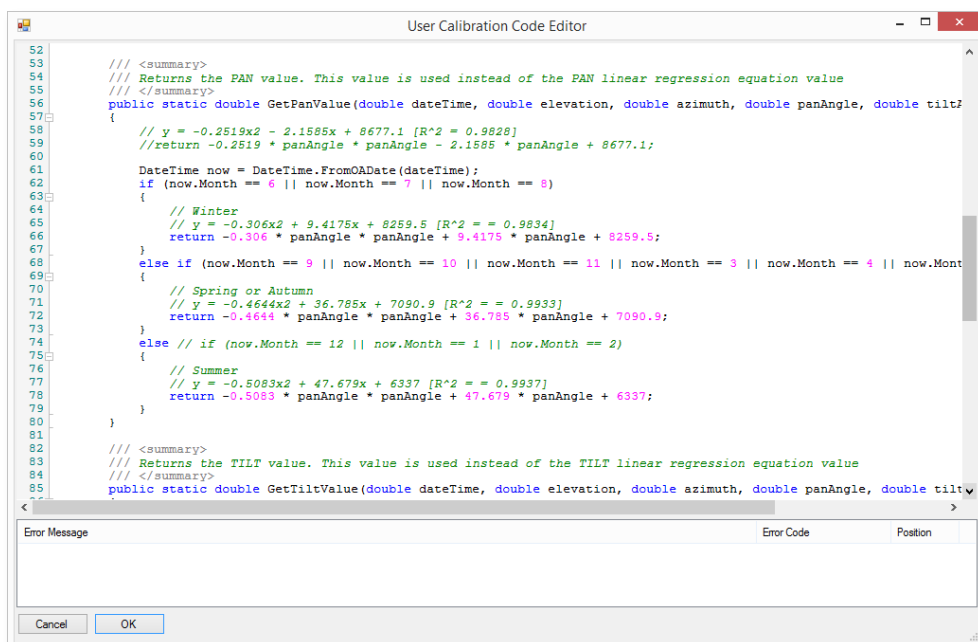
```

1 // This C# code file was initially created by the CATS Controller. User may edit the method bodies
2 // in this file, but they should NOT edit the namespace, class or method names or parameters!
3 using System;
4
5 namespace CloudSensor
6 {
7     /// <summary>User editable class to fine tune the linear regression equations for Pan/Tilt/Arm servos</summary>
8     public class UserCalibrationFineTuning
9     {
10         /// <summary>Main entry point</summary>
11         public static void Main()
12         {
13         }
14
15         /// <summary>Constructor</summary>
16         public UserCalibrationFineTuning()
17         {
18         }
19
20         /// <summary>
21         /// Returns the PAN fine tuning correction. This correction is added
22         /// to the result of the PAN linear regression equation
23         /// </summary>
24         public static double GetPanCorrection(double dateTime, double elevation, double azimuth, double panAngle, double tiltAngle)
25         {
26             return 120 * Math.Sin(1.8 + 0.155 * panAngle);
27         }
28
29         /// <summary>
30         /// Returns the TILT fine tuning correction. This correction is added
31         /// to the result of the TILT linear regression equation
32         /// </summary>
33         public static double GetTiltCorrection(double dateTime, double elevation, double azimuth, double panAngle, double tiltAngle)
34         {
35         }
36     }
37 }

```

OR

the advanced user may ignore the linear equations altogether and write their own equations (e.g. polynomial) in the **GetXXXValue()** methods:



```

52
53     /// <summary>
54     /// Returns the PAN value. This value is used instead of the PAN linear regression equation value
55     /// </summary>
56     public static double GetPanValue(double dateTime, double elevation, double azimuth, double panAngle, double tiltAngle)
57     {
58         // y = -0.2519x2 - 2.1585x + 8677.1 [R^2 = 0.9828]
59         //return -0.2519 * panAngle * panAngle - 2.1585 * panAngle + 8677.1;
60
61         DateTime now = DateTime.FromOADate(dateTime);
62         if (now.Month == 6 || now.Month == 7 || now.Month == 8)
63         {
64             // Winter
65             // y = -0.306x2 + 9.4175x + 8259.5 [R^2 = 0.9834]
66             return -0.306 * panAngle * panAngle + 9.4175 * panAngle + 8259.5;
67         }
68         else if (now.Month == 9 || now.Month == 10 || now.Month == 11 || now.Month == 3 || now.Month == 4 || now.Month == 5)
69         {
70             // Spring or Autumn
71             // y = -0.4644x2 + 36.785x + 7090.9 [R^2 = 0.9933]
72             return -0.4644 * panAngle * panAngle + 36.785 * panAngle + 7090.9;
73         }
74         else // if (now.Month == 12 || now.Month == 1 || now.Month == 2)
75         {
76             // Summer
77             // y = -0.5083x2 + 47.679x + 6337 [R^2 = 0.9937]
78             return -0.5083 * panAngle * panAngle + 47.679 * panAngle + 6337;
79         }
80     }
81
82     /// <summary>
83     /// Returns the TILT value. This value is used instead of the TILT linear regression equation value
84     /// </summary>
85     public static double GetTiltValue(double dateTime, double elevation, double azimuth, double panAngle, double tiltAngle)
86     {
87     }
88 }

```

All methods are passed the **Date/Time** for the ball position, the corresponding **Elevation** and **Azimuth** of the sun, as well as the **Pan**, **Tilt** and **Arm** angles. The **GetXXXCorrection()** methods return a value to be added to

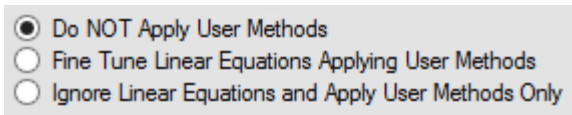
the linear equation's Y-intercept. The **GetXXXValue()** methods return the computed servo angle for the associated servo.

If the **UserCalibrations.cs** file (the one edited above) does not exist, the button text reads [**Create User Methods**], and when clicked, will create a template for the user.

When the user clicks [**OK**] after editing the code, the whole module is verified and any syntax errors found will appear in the error message list below the code window. The user **must** correct the error(s) before saving, or click [**Cancel**]. Double clicking an error will position the editing cursor to the error position.

The code may be edited at any time, and the changes will be applied during the next calculation.

The servo angle computation method used by the **Controller** is selected using the the following radio buttons:

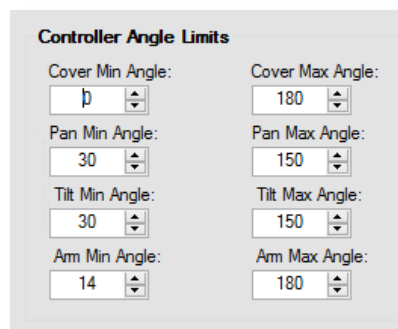


The image shows a set of three radio buttons for selecting the servo angle computation method. The first option, "Do NOT Apply User Methods", is selected with a filled circle. The other two options, "Fine Tune Linear Equations Applying User Methods" and "Ignore Linear Equations and Apply User Methods Only", are unselected with empty circles.

- ***“Do NOT Apply User Methods”*** – The given linear gradient and intercept values are used to compute the servo angle to position the shade-ball
- ***“Fine Tune Linear Equations Applying User Methods”*** – The given linear gradient and intercept values are used to compute the servo angles, together with any corrections to the Y-intercept computed in the **GetXXXCorrection()** methods.
- ***“Ignore Linear Equations and Apply User Methods Only”*** – The **GetXXXValue()** methods are used to compute the servo angles to position the shade-ball

When either of the latter two options are selected and the **GetXXXCorrection()** or **GetXXXValue()** methods are called, the script is compiled into a Dll (in a separate AppDomain to allow unloading of the Dll on completion) and the results applied.

Solar Shade | Servo Limits



The image shows a dialog box titled "Controller Angle Limits". It contains eight spin boxes arranged in a 4x2 grid. The left column contains minimum angles and the right column contains maximum angles for four servos: Cover, Pan, Tilt, and Arm. The values are: Cover Min 0, Cover Max 180; Pan Min 30, Pan Max 150; Tilt Min 30, Tilt Max 150; Arm Min 14, Arm Max 180.

Controller Angle Limits	
Cover Min Angle: 0	Cover Max Angle: 180
Pan Min Angle: 30	Pan Max Angle: 150
Tilt Min Angle: 30	Tilt Max Angle: 150
Arm Min Angle: 14	Arm Max Angle: 180

Cover Min/Max Servo Angle: The minimum/maximum Shade Cover servo motor angles

Pan Min/Max Servo Angle: The minimum/maximum Pan servo motor angles

Tilt Min/Max Servo Angle: The minimum/maximum Tilt servo motor angles

Arm Min/Max Servo Angle: The minimum/maximum Arm servo motor angles

Solar Shade | Dimensions

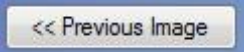

These are the sizes of, and distances between, the various components in the sensor, used to position the shade-ball. These will only need to be adjusted if the hardware components are moved.

Physical Dimensions	
Camera Radius: <input type="text" value="65"/> mm	Dome Base Height: <input type="text" value="0"/> mm
Base To Tilt Pivot: <input type="text" value="64"/> mm	Centre Pt Offset From Cam: <input type="text" value="105"/> mm
Tilt Pivot To Lens X: <input type="text" value="150"/> mm	Pan Motor Width: <input type="text" value="45"/> mm
Tilt Pivot To Lens Y: <input type="text" value="10"/> mm	Pan Motor Height: <input type="text" value="40"/> mm
Camera Base To Lens Y: <input type="text" value="42"/> mm	Tilt Motor Width: <input type="text" value="20"/> mm
Tilt Arm Length: <input type="text" value="90"/> mm	Tilt Motor Height: <input type="text" value="40"/> mm
Sun Shade Ball Radius: <input type="text" value="40"/> mm	Arm Motor Width: <input type="text" value="40"/> mm
Arm Length: <input type="text" value="165"/> mm	Arm Motor Height: <input type="text" value="20"/> mm
Shade Offset From Lens: <input type="text" value="65"/> mm	Camera Base Height: <input type="text" value="43"/> mm
Dome Radius: <input type="text" value="200"/> mm	Camera Base Width: <input type="text" value="63"/> mm

NEPHISYS CLOUD VIEWER

The Cloud Viewer allows the user to view all the images in a selected folder sequentially:

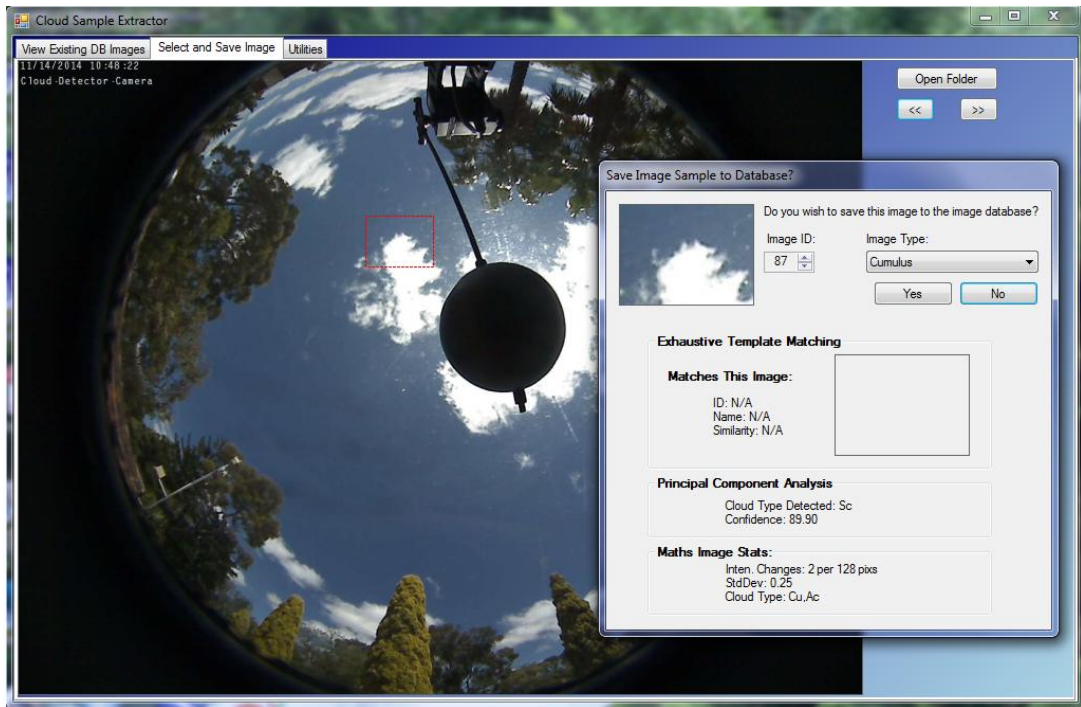


Load the folder and view all the images in that folder by clicking the  or  buttons. The user may also add notes describing the cloud types and timing for the day.

NEPHISYS CLOUD SAMPLE EXTRACTOR

The Sample Extractor allows:

- The user to select a whole-sky cloud image (one that contains well defined cloud definitions of interest) and to extract a 'pane' (128 x 96 pixels) for inclusion in the cloud database. This pane image will be used to match cloud types during the PC and Exhaustive Template Matching Analyses:



- The user to view the existing database images and their corresponding cloud types:

