



# GS Server

a simpler way to clear skies

ASCOM Telescope simulator and SkyWatcher driver. GS Server includes ASCOM telescope driver support and the SkyWatcher Protocol for Synta mounts: EQ8, EQ8R, HDX110, AZ-EQ5GT, Sirius Pro AZ/EQ-G, AZ-EQ6GT, Orion Atlas Pro AZ/EQ-G, EQ6-R PRO, NEQ6, HEQ5, EQ5, EQ4, AzGTi.

---

## Contents

Overview.....	6
1. Quick Start.....	6
2. Requirements.....	7
3. Installation.....	7
4. Running GSS.....	7
5. Main Window.....	9
Connected Applications .....	9
Mount Type – .....	9
6. SkyWatcher Controls.....	10
Top Coordinates Bar.....	10
RA (Right Ascension) .....	10
Dec (Declination).....	10
Azimuth.....	10
Altitude .....	10
Local Hour Angle (LHA) .....	10
GoTo Section.....	10
Copy Button.....	10
GoTo Button.....	10
Sync Button .....	10
PPEC Section .....	11
PPEC .....	11
PPEC Train.....	11
Buttons Bar.....	11
Park Positions .....	11
Park Button.....	11
Home Button .....	11
Stop Button .....	11
Tracking .....	11
Set Home .....	12

---

Flip SOP Button.....	12
AutoHome Button .....	12
Scheduler Button.....	12
Hand Controller.....	13
Speed.....	13
Mode .....	13
Guiding .....	13
Axes - .....	13
Anti-Lash .....	13
E/W .....	14
N/S.....	14
Graphic Section .....	14
3D.....	14
Diagnostics .....	14
Bottom Status Bar .....	14
Sky Watcher Settings .....	16
Baud Rate.....	17
Rates .....	17
Alignment Mode .....	17
Equatorial System .....	17
Mount.....	17
Max Slew Rate.....	17
Minimum Dec Pulse .....	17
Guide Rates .....	17
Over Meridian Limit .....	18
Encoders .....	18
Alternating PPEC .....	18
Full Current.....	18
GoTo Dec Pulse.....	18
Park Positions .....	18

---

Observatory Location .....	19
GPS .....	19
7. Option Settings.....	19
Available Tabs.....	20
No Sleep mode .....	20
Start Minimized .....	20
Start Window On Top.....	20
Home Warning On .....	20
Voice/Speech on .....	21
Theme Colors.....	21
Performance.....	21
Reset Settings.....	21
Monitor .....	22
Session Log .....	22
Log to File .....	22
8. Log Files .....	22
Session log.....	23
Error log.....	23
Monitor log .....	23
9. PPEC.....	23
10. Notes .....	24
Shortcuts Bar .....	25
11. Gamepad.....	25
12. 3D.....	26
13. Pole Locator .....	27
Polaris Polar Scope Alignment in 6 Easy Steps .....	27
Prerequisites .....	27
Step 1:.....	27
Step 2:.....	27
Step 3:.....	28

---

Step 4:.....	28
Step 5:.....	29
Step 6:.....	29
14. Pulses.....	30
GS ChartViewer .....	31
Start.....	31
Clear .....	31
Pause.....	31
Fit .....	31
Zoom .....	31
Capture .....	31
Type.....	32
Chart .....	32
Point Size.....	32
Invert.....	32
Scale .....	32
Quality .....	32
Animation .....	32
XScale .....	32
Max Point.....	32
PHD2 Host IP .....	32
15. Plot .....	33
Start.....	33
Clear .....	33
Pause.....	33
Fit .....	33
Zoom .....	33
Zero Base Axes.....	33
Y Scale.....	34
Zoom Quality.....	34

---

16. SkyWatcher Scripting .....	34
17. SkyWatcher API.....	35
18. Learn More .....	43

---

# Overview

GS Server (GSS) is an ASCOM compliant middleware known as a local server or hub. It's designed to allow other applications to access your telescope's mount in a practical and systematic way.

GS Server includes ASCOM telescope driver support using the SkyWatcher Protocol for Synta mounts: EQ8, EQ8R, HDX110, AZ-EQ5GT, Sirius Pro AZ/EQ-G, AZ-EQ6GT, Orion Atlas Pro AZ/EQ-G, EQ6-R PRO, NEQ6, HEQ5, EQ5, EQ4, AzGTi. It's also comes with a built in simulator so you can practice sessions without being connected to the mount.



Your mount should already be connected to the computer using the preferred method which is the EQ Direct approach. Specific cable required for this approach can be found on the internet. One popular site is <http://www.store.shoestringastronomy.com>.

## 1. Quick Start

1. Download and Install GS Server.
2. Open your planetarium program and select an ASCOM connection.
3. In the ASCOM chooser select ASCOM.GS.Sky.Telescope
4. In the ASCOM chooser click Configure and adjust any needed settings. Configure Com Port, Baud Rate, Mount, and Observatory Location. Click Save and Close
5. Click Connect from your planetarium program.
6. GS Server will start and connect to the mount. If you do not connect check your settings and click the 'Connect' button.
7. Connect any other applications needed.

---

## 2. Requirements

- Computer with Windows 7 SP1 or newer
- Microsoft .NET Framework 4.6.1 installed
- ASCOM Platform 6.4 or later
- Telescope mount that supports the syntax protocol
- EQ Direct cable to attach mount to the computer

## 3. Installation

After downloading Setup .exe file run and follow the prompts. The default location of files and settings can be found in the following locations on Windows systems.

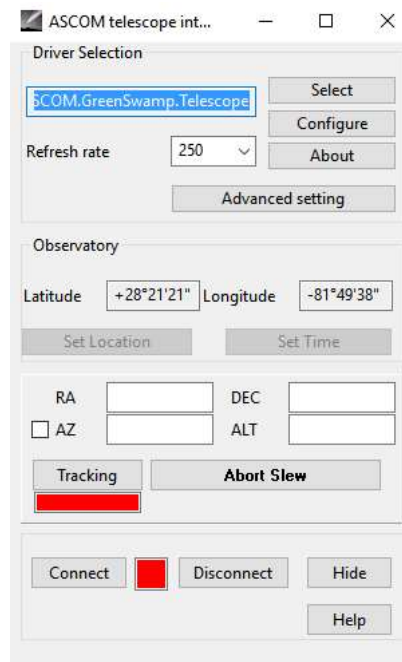
- GSS Program Files- C:\Program Files (x86)\Common Files\ASCOM\GSServer
- Log files – My Documents\GSServer
- Monitor files – My Documents\GSServer
- User Config – C:\Users\{User Name}\AppData\Local\GS\_Server
- Notes Templates- C:\Program Files (x86)\Common Files\ASCOM\GSServer\Templates

## 4. Running GSS

There are two ways to start GS Server; the preferred method is to allow other applications to start GSServer as they connect using the ASCOM interface. Each application that supports ASCOM will have their way to configure and connect.

Here is the ASCOM interface from SkyChart (CDC)



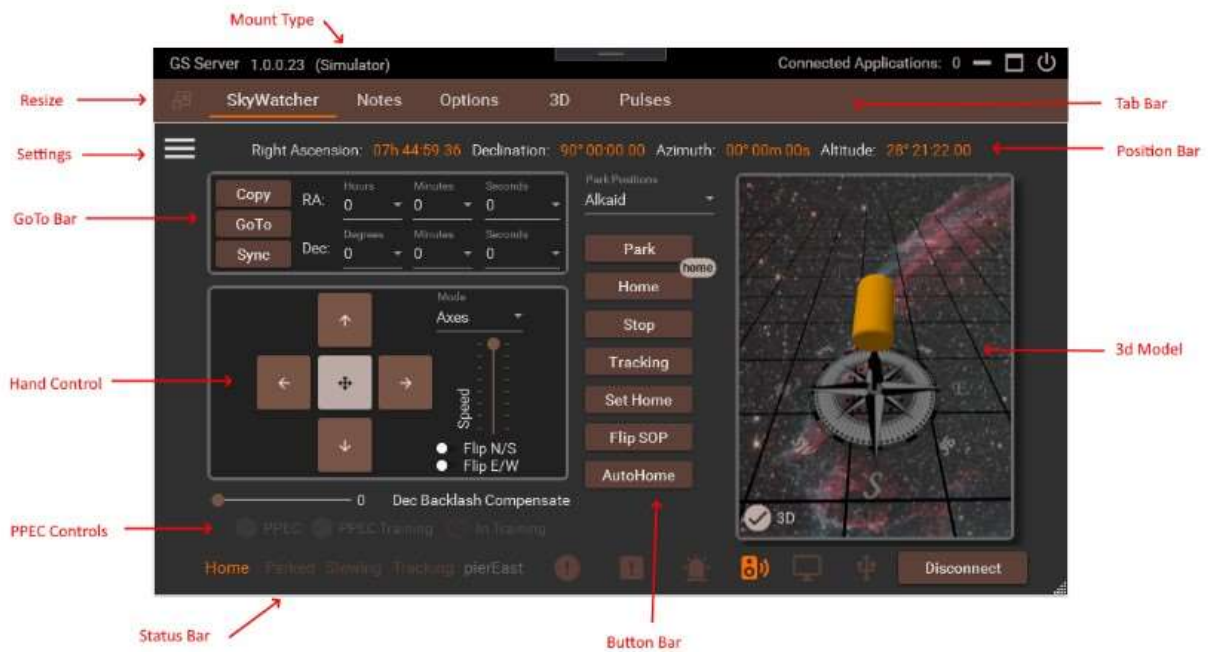


Use the Select button to select the ASCOM.GS.Sky.Telescope driver.

If this is the first time connecting to GS Server you will need to configure it before the applications will allow connections. See the configuration section. Once configured, click the connect button. If the configuration is correct GS Server will start within a few seconds. It's possible the connection may fail. If so GS Server will start but will not be connected to the mount. Reconfigure the driver and try connecting again.

The second method is to start GSServer.exe file located in the program directory. If starting for the first time configure the driver before hitting the connect button. Refer to the configuration section.

## 5. Main Window



The layout of the main window contains two primary areas; Tabs and Control. Each selected tab will show it's associated control.

The top of the window shows the version of GS Server, how many connected applications exist, minimize, close buttons and Mount Type.

**Connected Applications** – shows how many application are connected To GS Server. If you try to close GS Server when there are existing connections it will prompt you to confirm before closing. GS Server will exit on its own if the last calling application closes its connection.

**Mount Type** – shows either Simulator or SkyWatcher and is adjusted in the settings.

---

## 6. SkyWatcher Controls

List of items in the SkyWatcher Controls

### Top Coordinates Bar

**RA (Right Ascension)** – the distance of a point east of the First Point of Aries, measured along the celestial equator and expressed in hours, minutes, and seconds.

**Dec (Declination)** – the angular distance of a point north or south of the celestial equator.

**Azimuth** – the direction of a celestial object from the observer, expressed as the angular distance from the north or south point of the horizon to the point at which a vertical circle passing through the object intersects the horizon.

**Altitude** – the distance measurement, usually in the vertical or "up" direction, between a reference datum and a point or object.

**Local Hour Angle (LHA)** – it is the angle between the meridian of the observer and the meridian of the object you are pointing at. This can be used to tell how far the object is before or after the meridian.

### GoTo Section

**Copy Button** – Copies the current mount coordinates into the dropdown boxes.

**GoTo Button** – Tells the mount to move to the position defined in the dropdown boxes.

**Sync Button** – Takes the position entered and reset the mount to that location. Any syncs large than 45 degrees in either direction from the current position will not be accepted.

---

## PPEC Section

This control is enabled or disabled depending on the capabilities of each individual mount. Clicking the Mount info button in the status bar will show if PPEC is supported for the mount.

**PPEC (Permanent Periodical Error Correction)** - Checkbox to turn on or off PPEC. The grayed out timer will turn yellow when training is in progress. For more information see the PPEC section of this document.

**PPEC Train** – Checkbox to start the training process for PPEC

## Buttons Bar

**Park Positions** – List of available positions for parking. The default is two listings 'Default' and 'Home'. Home is same position as the Home button. Default is a copy of the park position from previous versions of GS. If you never set the previous version park position then Default will be set to the home position. Any applications requesting GS to park the mount uses the selected position at the time of the request. Adding and removing positions is done in the settings area of the SkyWatcher tab.

**Park Button** – move the mount to the defined park position defined within the setup settings. If mount is current slewing the button will stop the mount. Click again and the mount will park to the position in the dropdown.

**Home Button**– moves the mount to the initial home position which is typically counter weight down and the dec axis pointing to the pole. If mount is current slewing the button will stop the mount. Click again and the mount will move home.

**Stop Button**– Stops mount movement

**Tracking Button** – turn on or off the selected tracking rate to use. Will be automatically applied after a slew or goto.

---

**Set Home**– Resets the mount axes positions to the home position, Dec is pointing to the pole and Ra with the counter weights down.

**Flip SOP Button** – moves the mount to the opposite side of the pier. Works in both directions.

**AutoHome Button** – If your mount is equipped with auto home sensors then this button will assist in recovering the mount starting up in a position that is not recognized as a park or home position. This would be helpful if the mount loses power and is then powered on and the axis positions are unknown. Autohome will attempt to find each axis home sensor. It does this by moving each axis 5 degrees at a time towards the sensor for a maximum of 100 degrees. If it did not find the home sensor it will tell you at the end of the process. Once finished you will have the option to run it again for another 100 degrees search. If you're using a side saddle you can enter an offset that will move the mount +90 or -90 degrees after home is found auto home is complete. **WARNING - If you slew greater than +180deg from home and loose power, when doing the auto home, instead of slewing in a negative direction to get back it will slew in a Positive direction and possibly twist or bind existing cables. It's recommended to monitor the mount at all times while doing an autohome.**

**Scheduler Button** – Allows a park to be scheduled at a future date and time. When checked it uses the currently selected Park Position and the button will display "On" until the event is completed. It's the user responsibility to ensure no other software is interfering or working the mount at the time the event is scheduled to run. This option is reset when GS server restarts.

---

## Hand Controller

**Speed** – 8 settings to control the speed of the hand controller buttons. All speeds are relative to the Max Slew Rate in settings.

**Mode** – The direction of the hand controls are base on which mode is selected; Axes, Guiding, and Compass.

**Guiding** - Follow guiding directions in the same way guiding programs calibrate NESW with pulses for guiding. The following is defined by the ASCOM standard and would always apply. User could not flip directions.

- North (+ declination/altitude)
- South (- declination/altitude)
- East (+ right ascension/azimuth)
- West (- right ascension/azimuth)

**Axes** - Follow clockwise and counterclockwise directions of the axis. User would be able to set a flip switch for N vs S Hemi directions. Axis1 would be Ra/Az and Axis2 would be Dec/Alt

- North (CW Axis2)
- South (CCW Axis2)
- East (CW Axis1)
- West (CCW Axis1)

**Anti-Lash** – attempts to compensate for mechanical backlash errors in the mount when using the hand controls. Use the checkboxes to turn off/on the compensation for each axis. Best used at slower speeds when trying to use the hand controls to keep an object aligned.

---

**E/W** – Works when tracking is on and is based on the amount in the Backlash RA setting. If there is backlash in the RA axis each move east will move the lash to the wrong side. This will move the mount proportionally back to the west trying to remove the backlash so tracking will re-engage without having to travel the backlash distance.

**N/S** – Works when reversing the N/S direction and is based on the amount in the Backlash Dec setting. When the reverse direction is applied the compensation is applied before the actual move. This takes up the backlash in Dec before the reverse move is applied. Each reverse direction the backlash will be proportionally applied.

## Graphic Section

**3D** – Checkbox that displays a moving 3D model of a German equatorial mount. There is also a Tab item in Options to turn on/off a separate 3D model control. Right mouse click on the model to use the following shortcuts...

Pan - Shift W, A, S, D or Shift Arrow keys or Mouse wheel click and hold.

Pitch, Yaw - Right mouse click and hold and move

Zoom – Shift Page Up, Page Down or Mouse wheel

**Diagnostics** – If the diagnostic checkbox is selected in the settings the diagnostic information will be displayed in the graphics section

**Bottom Status Bar** – shows when the mount in the Home, Park, and Slewing positions or state.



**Connection** – is lit green when the mount is connected with a serial connection. Click for additional mount information.



**Monitor** – Status of the internal Monitor. Will be list green when active. See the Settings Control for more information about the Monitor.



**Errors Alert** – Will turn red if an error is logged. This can be reset by clicking on it.



**Warnings Alert** – Will turn yellow if a warning is logged. This can be reset by clicking on it. Warnings can be caused by a variety of reason from serial connections to the driver being ask for non-valid information. Check the session log for specific warning messages.



**Tracking Rate** – one of the 4 icons will be displayed representing the selected tracking rate. Sidereal, Lunar, Solar, or King.



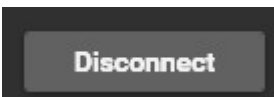
**Side of Pier** - Normal (pierEast) where the mechanical Dec is in the range -90 deg to +90 deg. Beyond the pole (pierWest) Where the mechanical Dec is in the range -180 deg to -90 deg or +90 deg to +180 deg.



**Axis Limits** – is yellow when one or more of the axes reaches a limit such as passing too far past the meridian. **Warning** – Axis limits will not stop the mount from moving and simply displays a warning light. It is up to the user to see that the mount is not damaged.



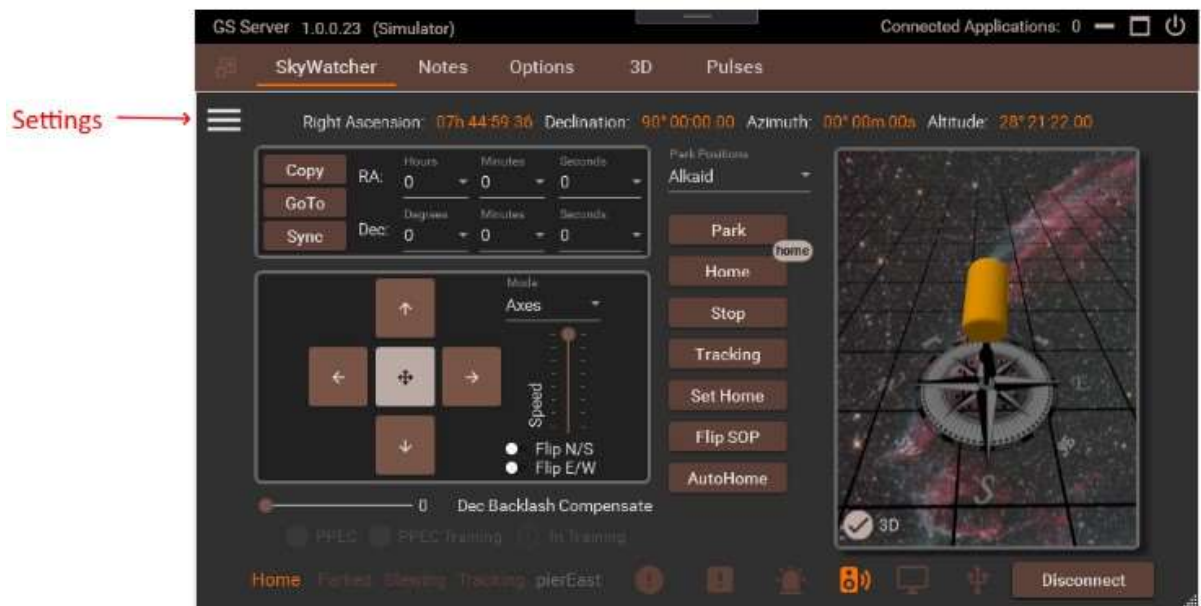
**Status Lights** – Parked, Home, Slewing, and Tracking are Items that are highlighted when active



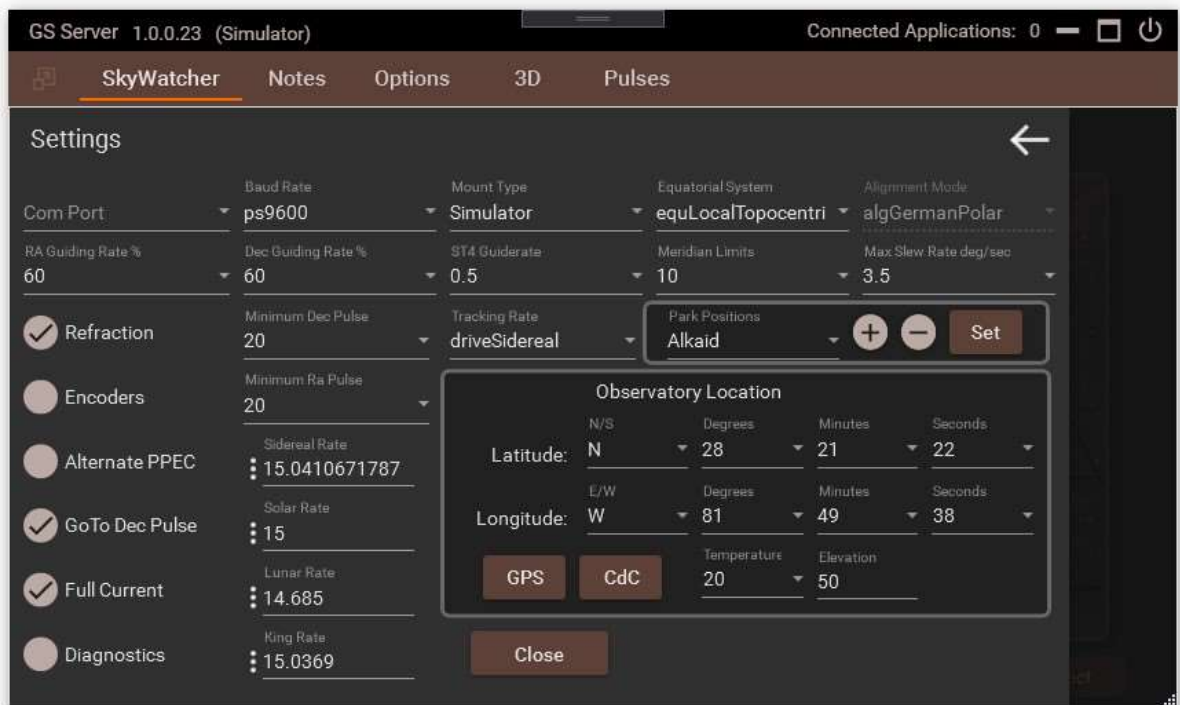
**Mount connection button.** When connected Disconnect will be displayed. When disconnected Connect will be displayed. When connecting any errors will be logged and shown in a popup window.



## Sky Watcher Settings



Click the 3 bars to open the settings for the SkyWatcher Control.



---

**Com Port** – Lists available or active ports. It is up to the user to select the correct port for the mount. If you don't know the mount port open or search for Device Manager and look in the 'Ports (COM & LPT)' section of the Device Manager.

**Baud Rate** – Rates from 300 to 230400 are available. Most mounts work at the default 9600 rate.

**Rates** – Sidereal, Solar, Lunar, and King are present from the ascom standards. Clicking the 3 dots will confirm a reset to the default setting.

**Alignment Mode** – GernalPolar is the default. Others will work with the Mount set to simulator.

**Equatorial System** – Local or TopoCentric is the default but. TopoCentric is also known as JNOW. Other is set to apparent. The server will accept input and output to the display the coordinates selected.

**Mount** – The simulator will work in all alignment modes. Which mount is selected will be used when an application attempts to connect to the server.

**Max Slew Rate** – speeds for the hand controller are a percent of the max rate. If you want slower movements change this to a smaller number.

**Minimum Dec Pulse & Minimum Ra Pulse** – **Warning, its recommended you leave this set to 20 and only change after discussing issues and options on the support forum.** This option is to refine guiding to a very small step increment. A correct setting is based on baud rate, guide rate, type of mount and settings in your guiding software. 20 milliseconds is the default and will work in almost all cases. A lower setting could cause poor performance depending on the capabilities of individual mounts.

**Guide Rates** – A percentage of the selected rates to use for guiding. The default 50% is a good starting point for guiding with applications like PHD2.

---

**Over Meridian Limit** – degrees before or passed the meridian the mount can travel before running into a limit alert. **Warning** – limits will not stop the mount from moving and simply displays a warning light. It is up to the user to see that the mount is not damaged or is flipped when necessary.

**Encoders** – Turns on and off the internal mount encoders.

**Alternating PPEC** – This allows pulses from guiding software to not be truncated or cut when the mount implements PPEC. When the internal PPEC runs it's been shown that some pulses maybe randomly cut short because of the timing in PPEC changes. This problem doesn't seem to exist for mounts using the ST4 or handbox interfaces only using an EQDir cable and the internal PPEC. The ST4 and Handbox interfaces implement a PPEC on/off strategy when dealing with pulses. This means that PPEC is turn off then the pulse is executed, then PPEC is turned back on. This strategy keep PPEC from interfering with the pulses. Using this option emulates that on/off strategy. Any mount using the internal PPEC and an EQDir cable should consider turning this option on and evaluate its long term affects. If not using the internal PPEC then there is there is no need to use this option.

**Full Current** – When you turn on the mount, by default the mount will track at half the watts/ampereage. The full current option tells the mount not to track at half current and to use the full current. Full current is recommended unless you're running off a battery and power consumption needs to be at a minimum.

**GoTo Dec Pulse** – A new alternative method when tracking and guiding. This turns pulses from guiding commands into a Dec goto command. It will slew like a normal goto which is different than the default pulse, which is based on the guide rate. GoTo slews are quicker and accurate. You can turn this on at any time even when guiding or tracking. If this option is turned off then guiding in Dec follow the normal guide rate and percentage. It's recommended to use this option and evaluate it over the long term.

**Park Positions** – Click the + and – buttons to Add or remove park positions. Use the Set button to assign the current mount axes positions to the selected park position. You cannot

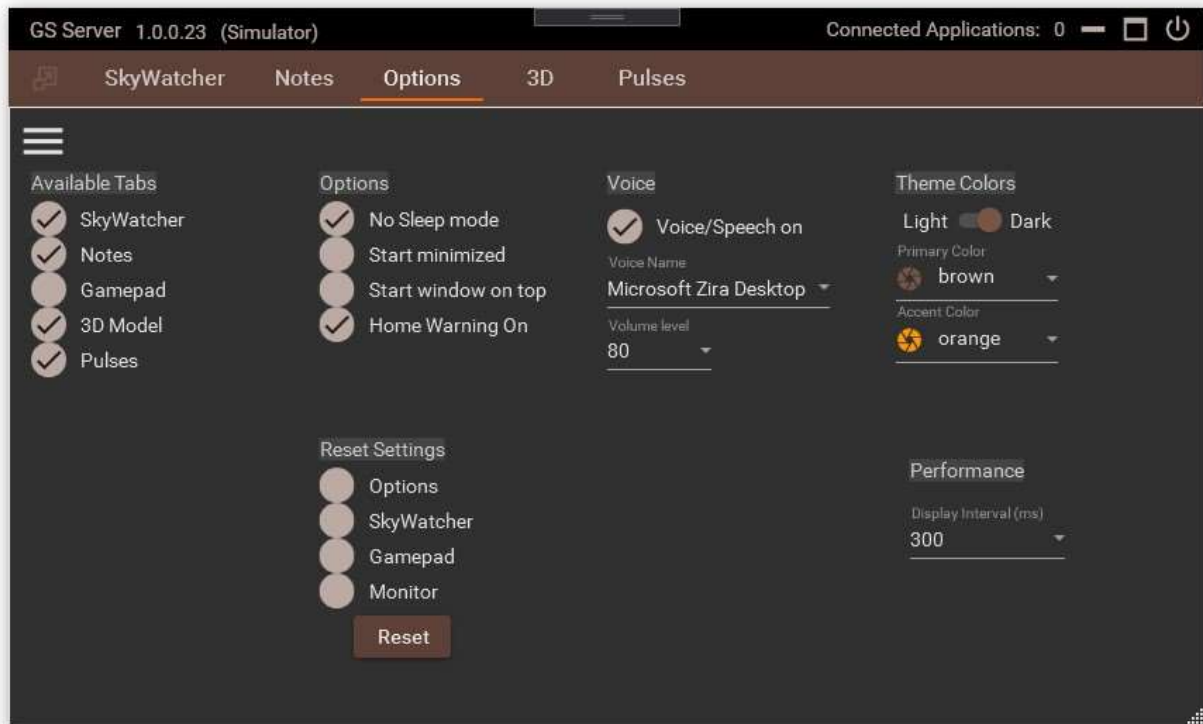
---

edit or modify the name of an existing position but you can reset its positions using the Set button. If you need to change a name it's best to create a new park position and then remove the old one.

**Observatory Location** - Latitude and Longitude of your observatory is used to calculate local positions.

**GPS** – Reads a COM port for available NEMA sentences. Select GGA or RMC. GGA is the default and RMC does not contain elevation data. GS will retrieve the first found tag GNGGA, GPGGA, GNRMC, GPRMC. Hit retrieve again for another tag search. The time is also pulled for and compared against the local pc clock. Any discrepancies are displayed. If you ran GS as Administrator you will see a checkbox to update the system time. Check your Windows system options for items that may change the time again like Internet Time Services. For viewing, guiding, and imaging a few seconds off isn't going to matter.

## 7. Option Settings



**Available Tabs** – Turns off or on selected tabs at top of the window. You cannot turn off the Settings tab it will always be available. It's recommended to set what tabs you want before connected to any hardware. If you have a gamepad the tab must be turned on to activate.

**No Sleep mode** – When GS server is running it will move the mouse a few pixels every 50 seconds to keep the windows screen saver from starting. This may also help to keep USB ports from going into sleep mode.

**Start Minimized** – When GS server is started it will show only in the windows taskbar in a minimized state. Click the GS Server icon in the taskbar to open the window to normal state.

**Start Window On Top** - When GS Server is started the window will be forced to show and stay on top of all other windows.

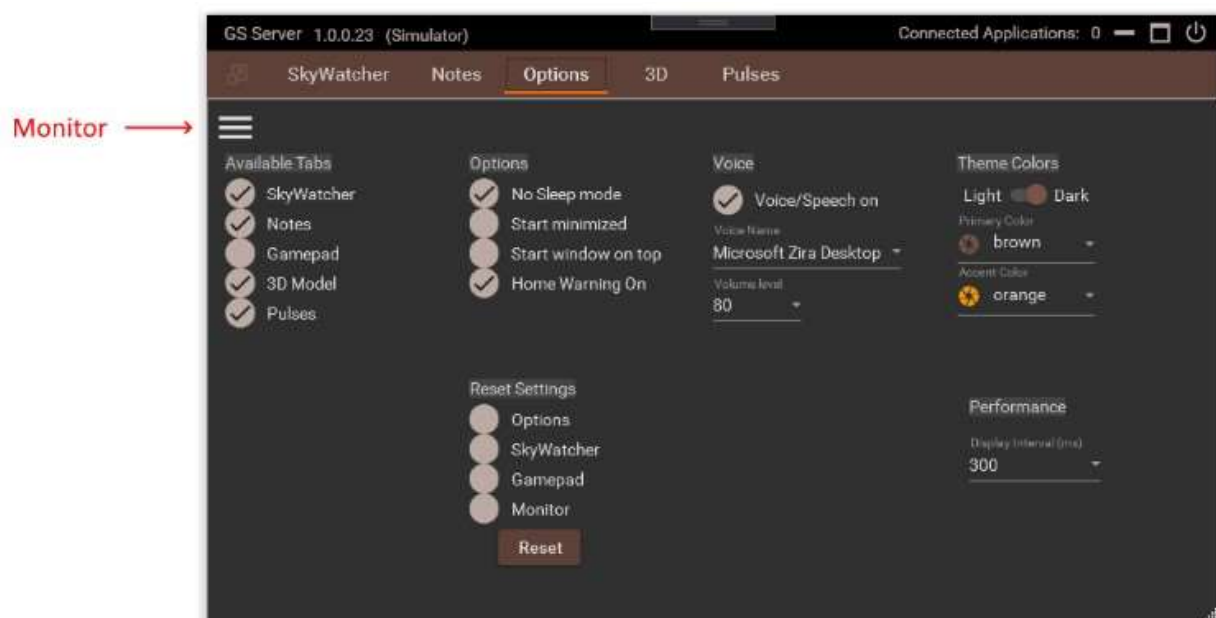
**Home Warning On** – When checked a popup warning will show that asks the user to put the mount in the home position. Uncheck this to stop showing the warning.

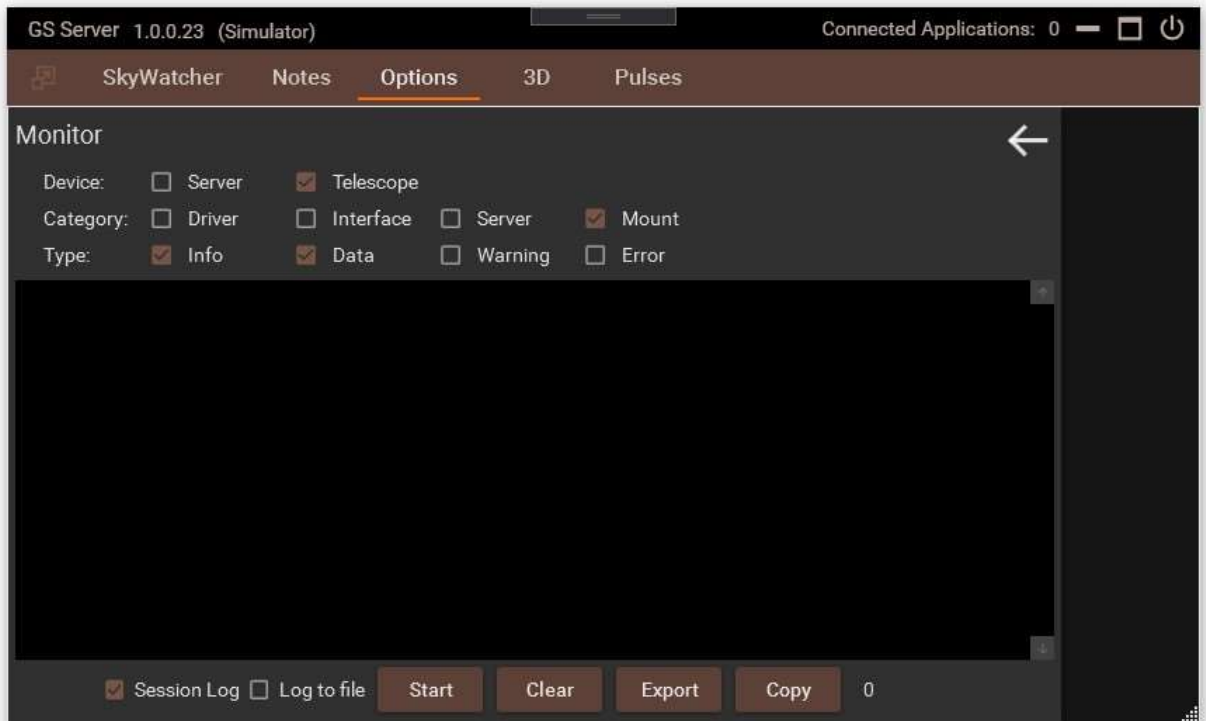
**Voice/Speech on** – Turns on/off the selected Microsoft synthesized voice. Windows loaded voices will be shown in the selection box. Refer to Microsoft for loading or unload voices into Windows operating systems.

**Theme Colors** – Choose light or dark theme along with the Primary and Accent colors.

**Performance** – This controls how often the driver polls the mount for position information. The faster the polling the more CPU utilization is needed. Unless there is a specific issue it's recommended to leave at the default 300.

**Reset Settings** – Rests the selected settings back to their default settings. Each checkbox goes to a set of option and matches the menu items. i.e. if you select SkyWatcher and click reset it will set all the settings under the SkyWatcher menu back to their default settings. The default setting will take effect the next time GS server is started. It's recommend after you reset any setting to restart GS server.





**Monitor** – The Monitor shows real-time logging by Device, Category, and Type. The Telescope selection refers to any control that is a type of Telescope like SkyWatcher. Category of Driver is for any incoming operations directly to the ascom driver. Interface would be the User Interface commands such as the hand controller items. Category Server would be the internal workings of GS Server. When the monitor is started it will not quit until turned off, even is the GS server is exited and started again. All files logged are kept in My Documents/GSServer.

**Session Log** – When checked turn on the log and keeps the last 5 rolling sessions logs.

**Log to File** – When checked anything shown in the monitor will output to a log file.

## 8. Log Files

Log files are in "My Documents\GSServer". There are 4 types of logs; Session, Error, Pulses, and Monitor. Any logs older than 7 days are automatically removed when GS starts. Session, Error,

---

and Pulses are started by default while Monitor is created manually using the Monitor screen. All log entries are created by the Monitor so they are also available using the Monitor screen.

**Session log** – Shows user actions taken along with any warnings issued by GS. Format:

GSSessionLogyyyy-dd-MM-HH.txt {record Number}, {yyyy:dd:MM:HH:mm:ss.fff}, {Device}, {Category}, {Type}, {Thread}, {Method}, {Message}

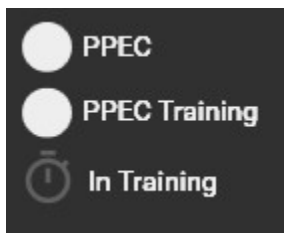
**Error log** – All errors issued by GS. Format: GSErrorLogyyyy-dd-MM-HH.txt {record Number}, {yyyy:dd:MM:HH:mm:ss.fff}, {Device}, {Category}, {Type}, {Thread}, {Method}, {Message}

**Monitor log** – Items created from the Monitor screen. Format: GSMonitorLogyyyy-dd-MM-HH.txt {record Number}, {yyyy:dd:MM:HH:mm:ss.fff}, {Device}, {Category}, {Type}, {Thread}, {Method}, {Message}

**Pulses log** – Items created from the Pulses screen. Format: GSPulsesLogyyyy-dd-MM}.txt

## 9. PPEC

Permanent Periodic Error Correction (PPEC) allows the mount to correct the right ascension (RA) for manufacturing errors in the worm wheel. PPEC Training your mount will take small movement corrections and store them within the mount. These corrections are then played back as the RA moves when PPEC is turned on.



**How to PPEC train your mount** - Before starting a training session your mount should be able to track and guide at a stable rate. When you are happy with the way in which the mount is guiding and you think it's stable and consistent, start a guiding session near the meridian and let it stabilize. Start the PPEC training session by clicking the PPEC Train checkbox. When the mount starts collecting data the timer icon will turn yellow. The mount will continue to collect data for several minutes. When finished the timer will turn gray and the PPEC Train checkbox is unchecked. You can now start and stop the PPEC replay using the PPEC checkbox. It's recommended that you use PPEC for any future



tracking or guiding sessions. If you're guiding session worsens turn off PPEC and validate your guiding is back to normal. You can retrain the mount at any time as long as the tracking is turned on.

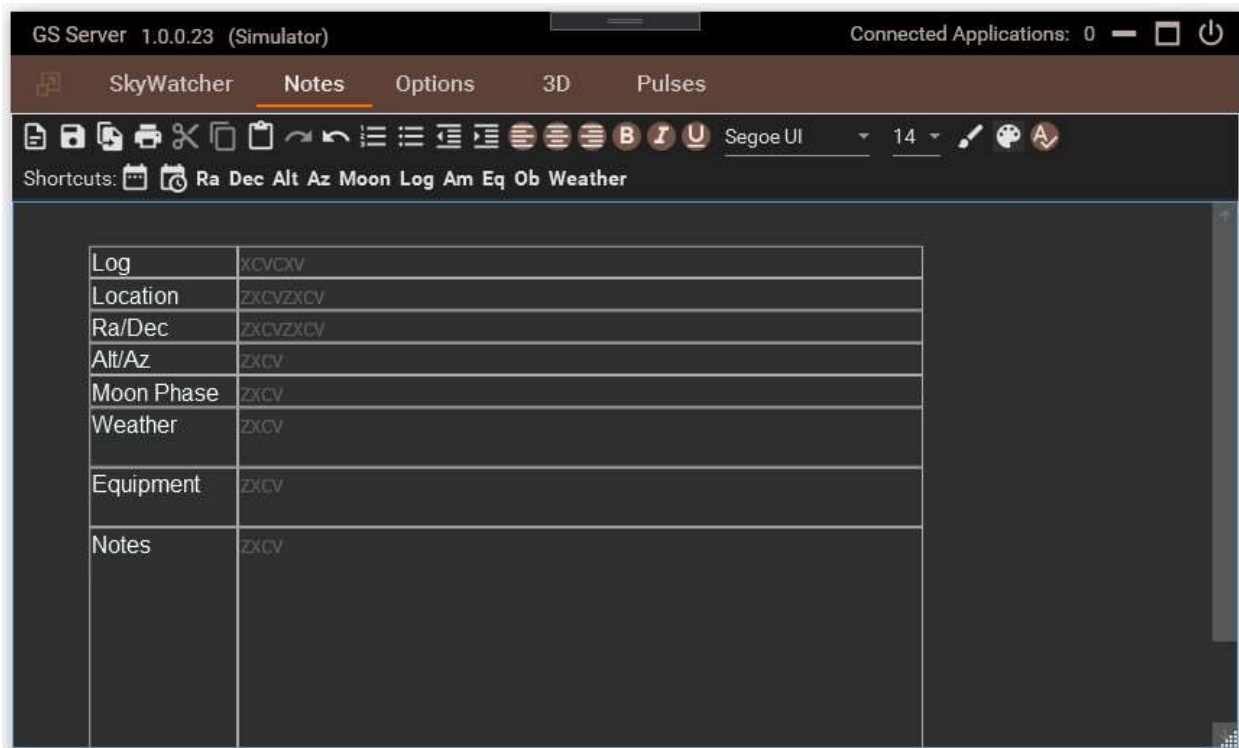
PPEC on/off checkbox – Turn on or off playback of corrections

PPEC Training Checkbox – Used to start a new collection session

PPEC Data collection icon – Indicates mount is collecting error correction data

When using PHD2 for guiding it's recommended to turn off PPEC during the PHD2 calibration process and turn it back when finished.

## 10. Notes

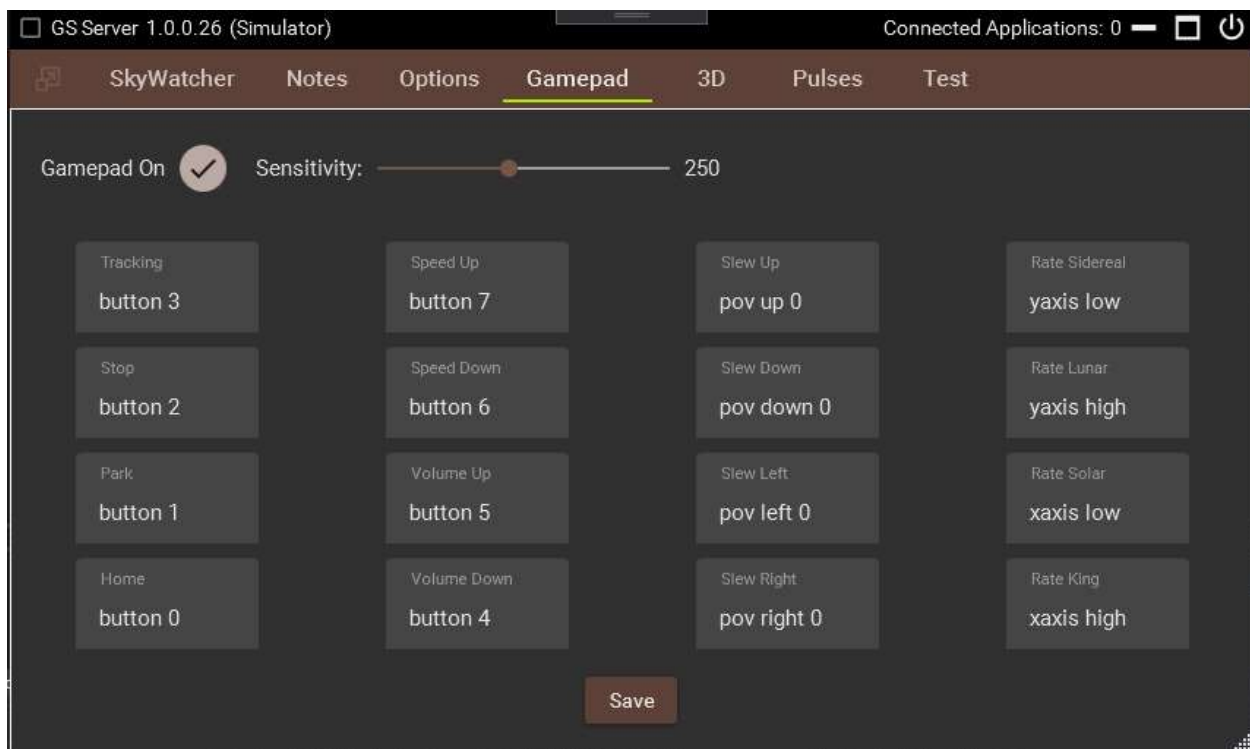


Notes works with files that use the Rich Text Format (.rtf). Other programs that also support rtf are Microsoft Word and WordPad along with Dropbox, OneDrive, and Google Drive.

Notes can be used to store information or logs during your viewing sessions. Various templates can be created using MS Word or WordPad then loaded into Notes. A sample template was install into the application directory. See the Installing section for locations.

**Shortcuts Bar** – contains number of buttons that will insert information where your cursor is located on a form or in the window.

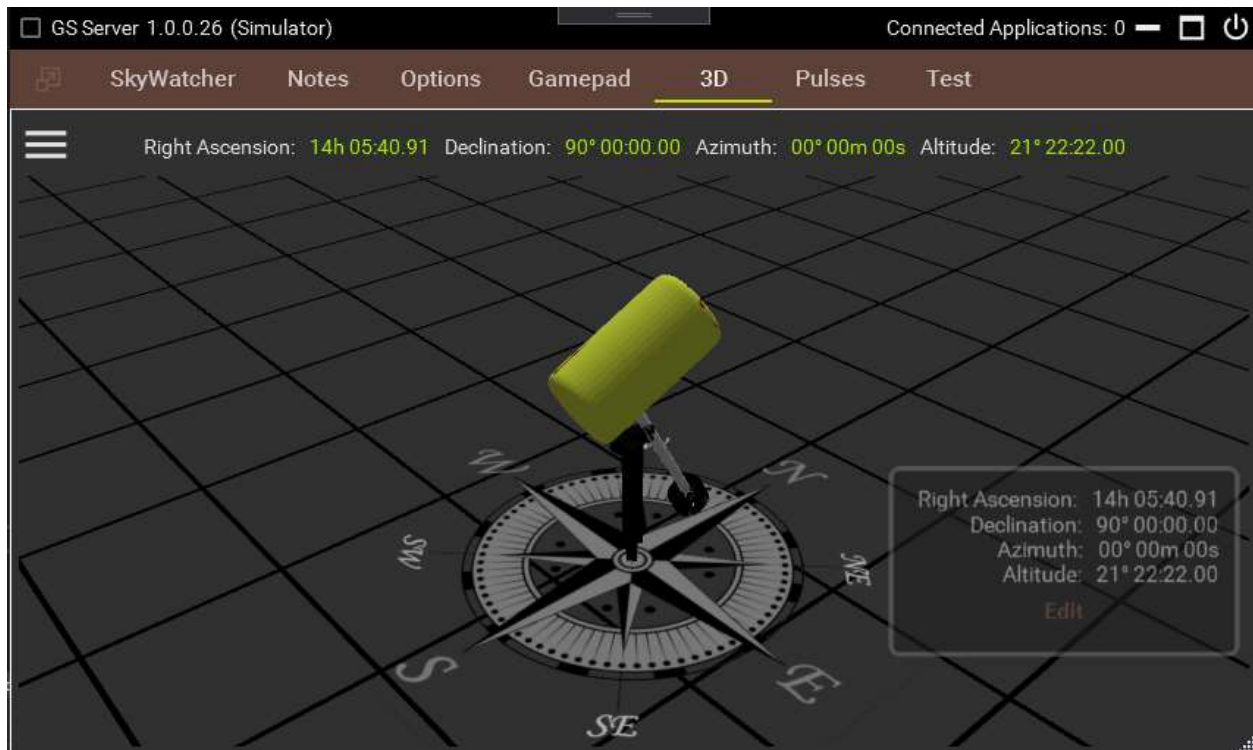
## 11. Gamepad



When configuring a gamepad or joystick it's recommend you not be connected to a mount. Turn on the Gamepad and click the mouse in the box where you want to select the gamepad

control. Click save to try out the new settings. After you connect to the mount you can judge how the sensitivity should be set. A faster setting will repeat buttons that are held down too long.

## 12. 3D



With the 3d tab you can watch a model of the mount as it moves in real time. The position of the mount is dependent on the proper starting of the mount and GS being in the home position or a parked position. If the mount starts up in an unknown position this view may not be correct.

Use the mouse right click & hold to pan or the mouse wheel to zoom the model for different viewing perspectives.

---

## 13. Pole Locator

If you don't have access to hardware like Polemaster or camera for plate solving then using the polar scope maybe your only option. Use can use the Pole Locator tab to polar align the mount using a polar scope. If you're in the north hemisphere use the directions for Polaris. In the southern hemisphere use the location of Sigma Octans as an indication of location.

---

### Polaris Polar Scope Alignment in 6 Easy Steps

**Prerequisites** - Verify Observatory information. If not correct adjust in the SkyWatcher/Settings tab.

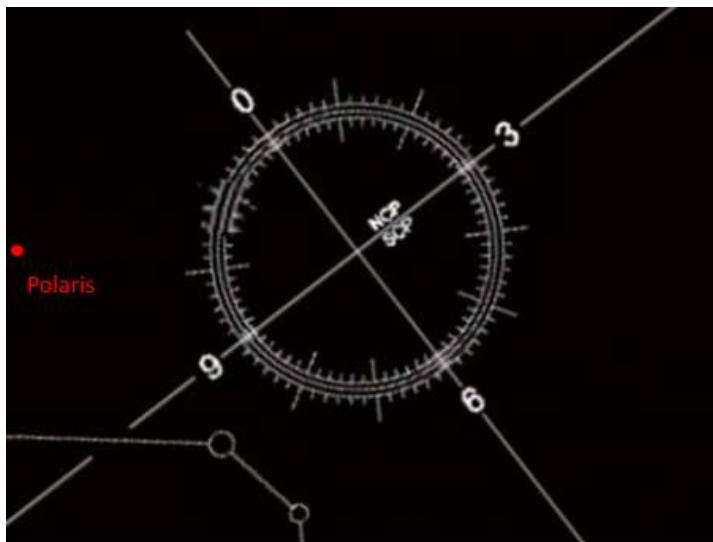
**Step 1:** Align and level the mount - Align the mount roughly towards Polaris, don't want to move it afterwards. Level it and get as close as you see fit. It does not need to be perfect but as close as possible is good enough. Bring Polaris anywhere inside the viewport using the Alt and AZ knobs on your mount.



**Step 2:** Center Polaris in the Crosshairs - Tip: carefully rotate the RA axis and make sure Polaris doesn't stray as the circle moves around. Polar scope should be aligned properly.



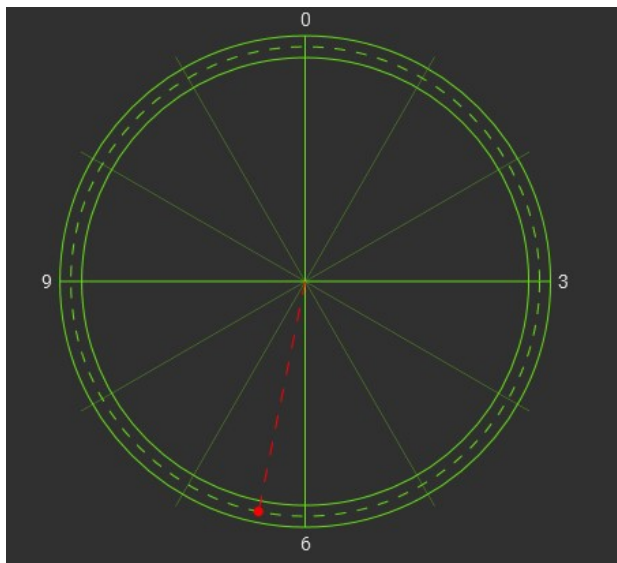
**Step 3:** Move Polaris off to the side and outside the circle – manually use ONLY either Altitude or Azimuth knobs to move Polaris outside the circle. Do this to align the polar circle perfectly horizontal in the next step. Move it approximate double the distance from the center of the circle. It's important that you only touch one set of polar alignment knobs, not both!



**Step 4:** Align the RA axis and the polar scope - rotate the right ascension axis until one of the lines is on Polaris.



Step 5: Use the Locator in GS to find correct position for Polaris (red dot). The dash circle line represents position for year 2020 and the inside circle is 2028.

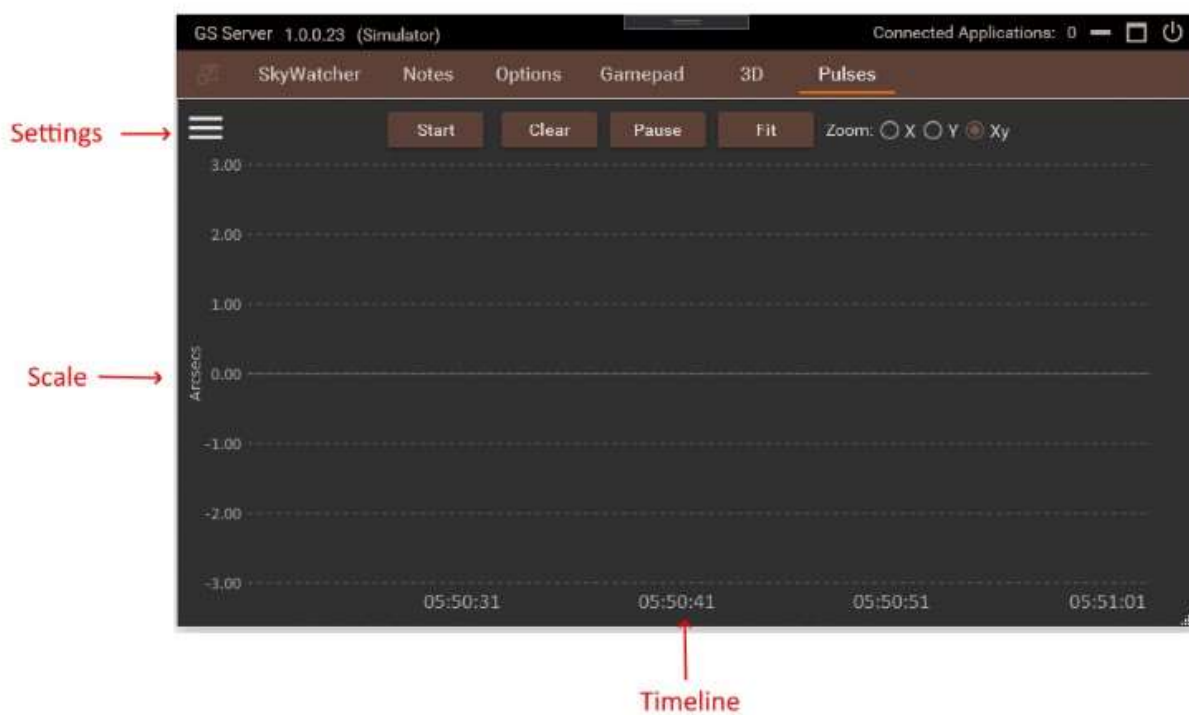


Step 6: Adjust the Alt and AZ knobs on your mount to move Polaris to the exact position



Complete, to refine the polar alignment consider doing a drift alignment.

## 14. Pulses



Pulses are captured from guiding applications like PHD2 which will send the guiding corrections to the driver in the form of individual pulses. These pulses can be captured and displayed here to see the performance of guiding for the mount.

**GS ChartViewer** – ChartViewer is another program packaged with GS that allow external viewing of the Pulses Logs. This allows logs to be viewed by others that have the ChartViewer program. Locate GS.ChartViewer.exe in the install directory.

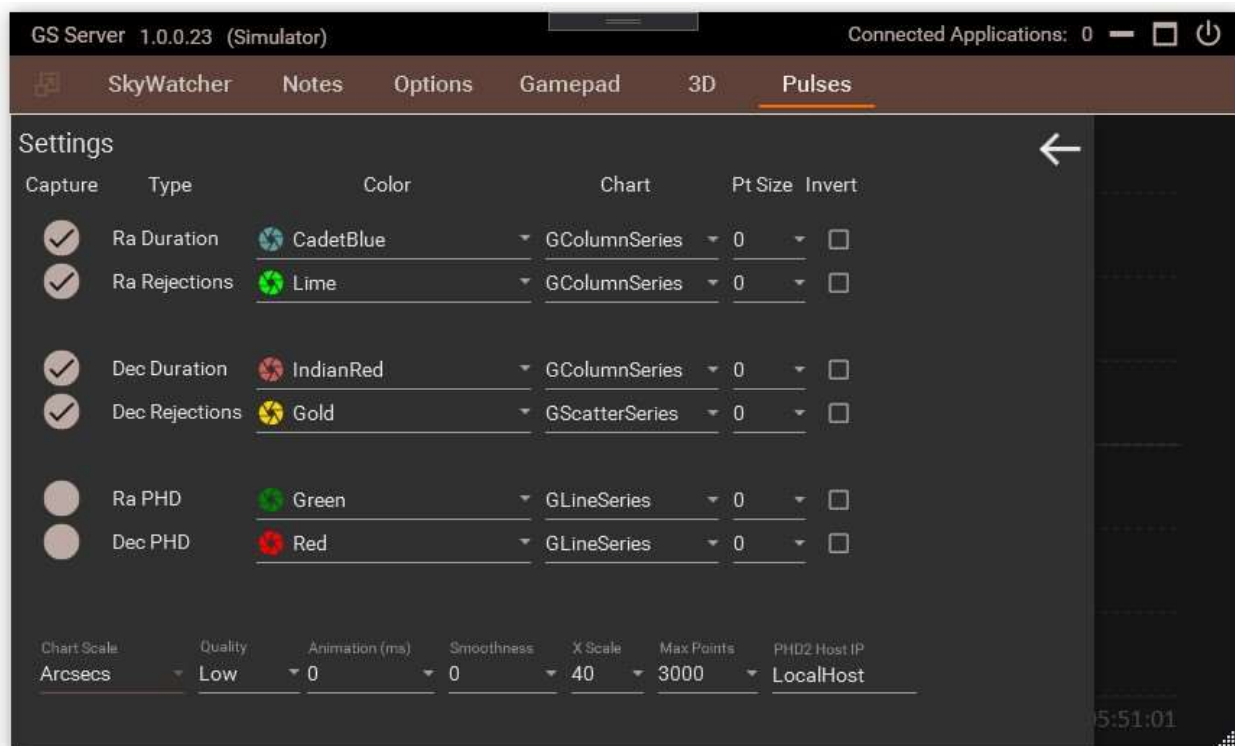
**Start** – Stars and stops the chart. Pulse logs are automatically saved to the documents area.

**Clear** – Removes any drawings on the screen only

**Pause** – Stops the timeline from auto scrolling and allows the chart to be panned and zoomed with the mouse.

**Fit** – Readjusts the chart to fit the screen using the selected scale.

**Zoom** – Changes the mouse behavior.



**Capture** – Check to show and log a series.



---

## Type

1. Ra Duration – Normal Ra corrections being process by the driver
2. Ra Rejections – Rejected Ra pulses
3. Dec Duration - Normal Dec corrections being process by the driver
4. Dec Rejections - Rejected Dec pulses
5. RA PHD and Dec PHD – Displays the guiding performance from PHD2. If one is checked a connection will be made to the PHD2 server so make sure it's turned on in the PHD2 settings. The PHD Host IP is also required for the connection to work.

Rejected pulses are pulses that are not accepted to process by the driver and are typically too small. The rejection size is determined by the minimum pulse setting in the SkyWather tab. Specific minimum settings are mount specific but typically 20 milliseconds is a good generic settings. If lots of pulses are being rejected you might have to adjust the guiding application or adjust the minimum pulse setting.

**Chart** – Selects a specific series to show such as Line, Column, Step Line, or Scatter.

**Point Size** – Some series can show specific points on the chart and this would determine the size.

**Invert** – Will invert the data so basically it will flip the data from above zero to below zero or visa-versa.

**Scale** – Scale will determine the Y scaling to be use for all captured items.

**Quality** – Determines how much detail to show when zoomed out or in. If set to low the farther you zoom out the less detail is shown. The more you zoom in more detail will be shown. Low is best setting for performance and a higher quality will slow down zooming affects.

**Animation** – Each series can animate how points are drawn on the screen. Use this in conjunction with smoothness for different effects. Any setting above 0 will affect performance of the chart.

**XScale** – How many seconds to show on the timeline

**Max Point** – How many points to keep in the series while the chart is being shown on the screen. Larger settings can affect performance of the chart. This can be useful if you pause the chart and scroll back to view the history.

**PHD2 Host IP** – IP or host name of the PHD server. This is required if capture is turned on for PHD items. LocalHost is the default computer.

## 15. Plot

The Plot tab will chart the axes real time.



**Start** – clear the plots and starts a new chart based on the settings.

**Clear** – removes the current plots from the screen.

**Pause** – stops the plots from moving forward in time. This allows you to use the mouse to move the chart around.

**Fit** – Attempts to readjust the plot to fit in the current chart's Y axis. Using this in conjunction with the zoom options.

**Zoom** – Allows mouse wheel movement for one or both chart axes.

**Zero Base Axes** – uses zero as the axes starting point rather than the home positions which are typically set at 90/90 degrees.

---

Y Scale – Display plots in microsteps, arcseconds, or degrees

Zoom Quality – the farther zoomed out the less detail is shown. When adjusted higher more detailed is shown at zoomed out distances. Default is set to Low so you need to zoom in to improve the details. Low is best used for performance reasons.

## 16. SkyWatcher Scripting

WARNING! – use the API at your own risk. It's recommended that you do not leave the amount alone while running scripts. Green Swamp is not responsible for any damage resulting from using the API interface.

Running scripts requires that GSS be running and connected to the mount. You can start GSS on its own or by using other external programs that will load the GSS ASCOM driver such as CDC or SGP. Unless you're experienced with how GSS separates ASCOM commands from the API It's recommended that GSS be running on its own so no other external sources can interfere or interact with running scripts.

GSS exposes a number of class members available to external programs and scripting languages. There are Powershell example scripts located in the GSS Program files scripting directory. Find the installation section of this manual for directory locations.

The examples scripts we've created using the windows 10 Powershell ISE. To run Powershell, type in the Windows search box "Powershell ISE" and run either the 64 bit or X86 versions. You may run into a problems executing scripts because of Windows security settings. There are a number of ways to correct this and you should search google for Microsoft's recommended approach. One way is to allow the current logged in user the ability to run script. The following command can be run within PowerShell by placing it on the first line and clicking execute or the run button.

Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Bypass -Force;

---

## 17. SkyWatcher API

This list contains the definition for the available API commands.

```
public interface ISky
{
    /// <summary>
    /// Tells GSS not to process any ASCOM moment commands for external programs using the
    ASCOM driver.
    /// </summary>
    /// <returns>bool</returns>
    bool AscomOn { get; set; }
    /// <summary>
    /// Starts the AutoHome slew to home sensors
    /// </summary>
    /// <param name="degreelimit"></param>
    /// <param name="offsetdec"></param>
    void AutoHomeStart(int degreelimit = 100, int offsetdec = 0);
    /// <summary>
    /// Stops Autohome from completing
    /// </summary>
    void AutoHomeStop();
    /// <summary>
    /// Move axis number of microsteps, not marked as slewing
    /// </summary>
    /// <param name="axis">>axis number 1 or 2</param>
    /// <param name="steps">number of microsteps</param>
    /// <returns>nothing</returns>
    void AxisMoveSteps(int axis, long steps);
    /// <summary>
    /// Send a pulse command
    /// </summary>
    /// <param name="axis">Axis 1 or 2</param>
    /// <param name="guiderate">Guiderate degrees, 15.041/3600*.5, negative value denotes
    direction</param>
    /// <param name="duration">length of pulse in milliseconds, always positive numbers</param>
    /// <param name="backlashsteps">Positive microsteps added for backlash</param>
    /// <returns>nothing</returns>
    void AxisPulse(int axis, double guiderate, int duration, int backlashsteps = 0);
    /// <summary>
    /// Goto position in degrees
    /// </summary>
    /// <param name="axis">axis number 1 or 2</param>
    /// <param name="targetPosition">position in degrees</param>
    /// <returns>nothing</returns>
```

---

```

void AxisGoToTarget(int axis, double targetPosition);
/// <summary>
/// Slew axis based on a rate in degrees. Use this for small movements
/// like pulseguiding, rate changes, guiding changes, not gotos
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="rate">rate/sec in degrees</param>
/// <returns>nothing</returns>
void AxisSlew(int axis, double rate);
/// <summary>
/// K Slows to a stop movement of an Axis
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
void AxisStop(int axis);
/// <summary>
/// L Abruptly stops movement of an Axis
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns>nothing</returns>
void AxisStopInstant(int axis);
/// <summary>
/// q Axes slews must start independently
/// </summary>
bool CanAxisSlewsIndependent { get; }
/// <summary>
/// q Does mount support AZ/EQ mode
/// </summary>
bool CanAzEq { get; }
/// <summary>
/// q Does mount support dual encoders
/// </summary>
bool CanDualEncoders { get; }
/// <summary>
/// q Does mount support half current tracking
/// </summary>
bool CanHalfTrack { get; }
/// <summary>
/// q Does mount support home sensors
/// </summary>
bool CanHomeSensors { get; }
/// <summary>
/// Test result if the mount can move Dec a single step in GoTo mode
/// </summary>
bool CanOneStepDec { get; }
/// <summary>
/// Test result if the mount can move Ra a single step in GoTo mode
/// </summary>
bool CanOneStepRa { get; }

```

---

```

/// <summary>
/// q Does mount support a polar LED
/// </summary>
bool CanPolarLed { get; }
/// <summary>
/// q Does mount support PPEC
/// </summary>
bool CanPpec { get; }
/// <summary>
/// q Does mount support WiFi
/// </summary>
bool CanWifi { get; }
/// <summary>
/// Sets the amount of steps added to Dec for reverse backlash pulse
/// </summary>
int DecBacklash { get; set; }
/// <summary>
/// Gets the number of steps from the angle in rad
/// </summary>
/// <param name="axis"></param>
/// <param name="angleinrad"></param>
/// <returns>Steps in rad</returns>
long GetAngleToStep(int axis, double angleinrad);
/// <summary>
/// e Gets versions of axis in long format
/// </summary>
/// <param name="axis"></param>
/// <returns>long axis version</returns>
long GetAxisVersion(int axis);
/// <summary>
/// e Gets version of axis in string readable format
/// </summary>
/// <param name="axis"></param>
/// <returns>string axis version as string</returns>
string GetAxisStringVersion(int axis);
/// <summary>
/// j Gets current axis position in degrees
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns>Get Current Axis position as double</returns>
double GetAxisPosition(int axis);
/// <summary>
/// j Gets axis position counter
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns>Cardinal encoder count as long</returns>
long GetAxisPositionCounter(int axis);
/// <summary>

```

---

```

    /// d Gets Axis Current Encoder count
    /// </summary>
    /// <param name="axis">axis number 1 or 2</param>
    /// <returns>count as double</returns>
    double GetEncoderCount(int axis);
    /// <summary>
    /// Multiply the value of radians/second by this factor to get a 32-bit integer for the set speed used
    by the motor board.
    /// </summary>
    /// <param name="axis">axis number 1 or 2</param>
    /// <returns>factor used to get the speed</returns>
    double GetFactorRadRateToInt(int axis);
    /// <summary>
    /// Inquire motor high speed ratio
    /// </summary>
    /// <param name="axis">axis number 1 or 2</param>
    /// <returns>Ratio used to determine high speed</returns>
    long GetHighSpeedRatio(int axis);
    /// <summary>
    /// q Get Home position
    /// </summary>
    /// <param name="axis">axis number 1 or 2</param>
    long GetHomePosition(int axis);
    /// <summary>
    /// h Get Current "goto" target
    /// </summary>
    /// <param name="axis">axis number 1 or 2</param>
    double GetLastGoToTarget(int axis);
    /// <summary>
    /// i Get Current "slew" speed
    /// </summary>
    /// <param name="axis">axis number 1 or 2</param>
    long GetLastSlewSpeed(int axis);
    /// <summary>
    /// Margin used to move from high speed to low speed
    /// </summary>
    /// <param name="axis">axis number 1 or 2</param>
    /// <returns></returns>
    long GetLowSpeedGotoMargin(int axis);
    /// <summary>
    /// e Gets the complete version string
    /// </summary>
    /// <returns></returns>
    string GetMotorCardVersion(int axis);
    /// <summary>
    /// Runs a motor test to see if axis can move one step in GoTo mode
    /// </summary>
    /// <param name="axis">axis number 1 or 2</param>

```

---

```

/// <returns>Result of each axis test</returns>
bool GetOneStepIndicator(int axis);
/// <summary>
/// s Inquire PEC Period ":s(*1)", where *1: '1'= CH1, '2'= CH2, '3'= Both.
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
double GetPecPeriod(int axis);
/// <summary>
/// c Microsteps from target where the rampdown process begins
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
double GetRampDownRange(int axis);
/// <summary>
/// D Sidereal rate in axis speed
/// </summary>
/// <returns></returns>
/// <param name="axis">axis number 1 or 2</param>
long GetSiderealRate(int axis);
/// <summary>
/// Gets the angle in rad from amount of steps
/// </summary>
/// <param name="axis"></param>
/// <param name="steps"></param>
/// <returns></returns>
double GetStepToAngle(int axis, long steps);
/// <summary>
/// a Steps per revolution
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns>Step Count</returns>
long GetStepsPerRevolution(int axis);
/// <summary>
/// b Frequency of stepping timer
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns>Frequency of stepping timer</returns>
long GetStepTimeFreq(int axis);
/// <summary>
/// The current Declination guide rate
/// </summary>
double GuideRateDeclination { get; set; }
/// <summary>
/// The current Right Ascension guide rate
/// </summary>
double GuideRateRightAscension { get; set; }
/// <summary>
/// F Initialize both Axes
/// </summary>

```



---

```
void InitializeAxes();
/// <summary>
/// Is the autohome process running
/// </summary>
bool IsAutoHomeRunning { get; }
/// <summary>
/// Is mount in a connected serial state
/// </summary>
bool IsConnected { get; }
/// <summary>
/// q Is the mount collecting PPEC data
/// </summary>
bool IsPpecInTrainingOn { get; }
/// <summary>
/// q Does the mount have PPEC turned on
/// </summary>
bool IsPpecOn { get; }
/// <summary>
/// j Is axis at full stop
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns></returns>
bool IsFullStop(int axis);
/// <summary>
/// j Is axis in highspeed mode
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns></returns>
bool IsHighSpeed(int axis);
/// <summary>
/// Is mount type set to SkyWatcher
/// </summary>
bool IsServerSkyWatcher { get; }
/// <summary>
/// f Is axis slewing normal mode
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns></returns>
bool IsSlewing(int axis);
/// <summary>
/// f Is axis slewing in a positive direction
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns></returns>
bool IsSlewingFoward(int axis);
/// <summary>
/// f Is axis slewing in goto mode
/// </summary>
```

---

```

/// <param name="axis">axis number 1 or 2</param>
/// <returns></returns>
bool IsSlewingTo(int axis);
/// <summary>
/// Last known error from the AutoHome Process
/// </summary>
string LastAutoHomeError { get; }
/// <summary>
/// e Identify type of mount
/// </summary>
bool MountType { get; }
/// <summary>
/// e Identify board version
/// </summary>
bool MountVersion { get; }
/// <summary>
/// Turns PPEC off during movements and then back on for error correction moves
/// </summary>
/// <param name="on"></param>
void SkySetAlternatingPpec(bool on);
/// <summary>
/// E Reset the position of an axis
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="position">degrees</param>
void SetAxisPosition(int axis, double position);
/// <summary>
/// M Set the break point increment
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="stepsCount">The steps count.</param>
void SetBreakPointIncrement(int axis, long stepsCount);
/// <summary>
/// Turns on or off converting a Dec pulse guide into a Dec GoTo
/// </summary>
/// <param name="on"></param>
void SetDecPulseToGoTo(bool on);
/// <summary>
/// W 4-5 Turn on off encoders
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="on"></param>
void SetEncoder(int axis, bool on);
/// <summary>
/// W 6 Enable or Disable Full Current Low speed
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="on"></param>

```

---

```

void SetFullCurrentLowSpeed(int axis, bool on);
/// <summary>
/// H Set the goto target increment in steps
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="stepsCount"></param>
void SetGotoTargetIncrement(int axis, long stepsCount);
/// <summary>
/// W 8 Reset the home position index
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
void SetHomePositionIndex(int axis);
/// <summary>
/// J Start motion based on previous settings
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
void StartMotion(int axis);
/// <summary>
/// G Set a different motion mode
/// </summary>
/// <param name="axis">Axis number 1 or 2</param>
/// <param name="func">'0' high speed GOTO slewing,'1' low speed slewing mode,'2' low speed
GOTO mode,'3' High slewing mode</param>
/// <param name="direction">0=forward (CW) right, 1=backaward (CCW) left, also based on
obstortary settings</param>
void SetMotionMode(int axis, int func, int direction);
/// <summary>
/// W 2-3 Turn on off PPEC
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="on"></param>
void SetPpec(int axis, bool on);
/// <summary>
/// W 0-1 Turn on off PPEC training
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="on"></param>
void SetPpecTrain(int axis, bool on);
/// <summary>
/// I Set slewing rate, seems to relate to amount of skipped step counts.
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="stepSpeed">StepSpeed = 1 motor step movement, higher counts means slower
movements</param>
void SetStepSpeed(int axis, long stepSpeed);
/// <summary>
/// S Set absolute goto target
/// </summary>

```

---

```
/// <param name="axis">axis number 1 or 2</param>
/// <param name="position"></param>
void SetTargetPosition(int axis, double position);
/// <summary>
/// shutdown and close the server
/// </summary>
void ShutdownServer();
} }
```

---

## 18. Learn More

Join us on Groups.io at <https://groups.io/g/GSS/topics>

Videos available at <https://vimeo.com/user103775098>

Facebook at <https://www.facebook.com/groups/517303835383709/>