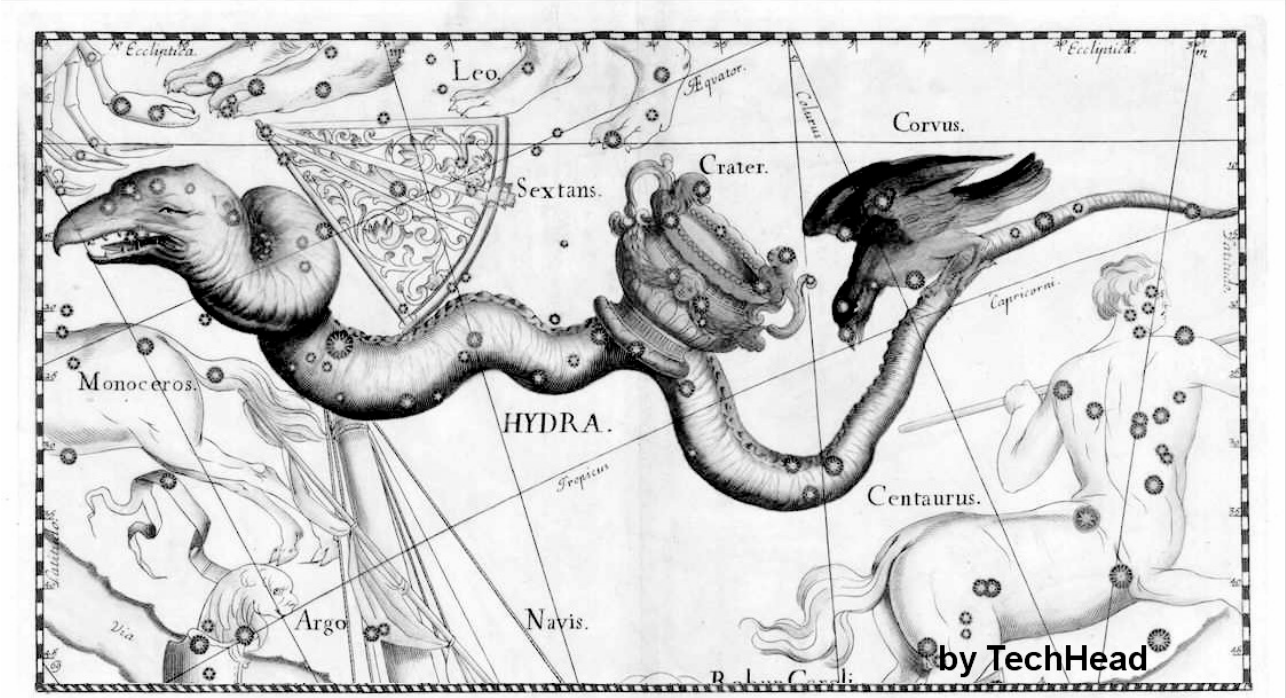
Hydra system
User Guide



TechHead

2014

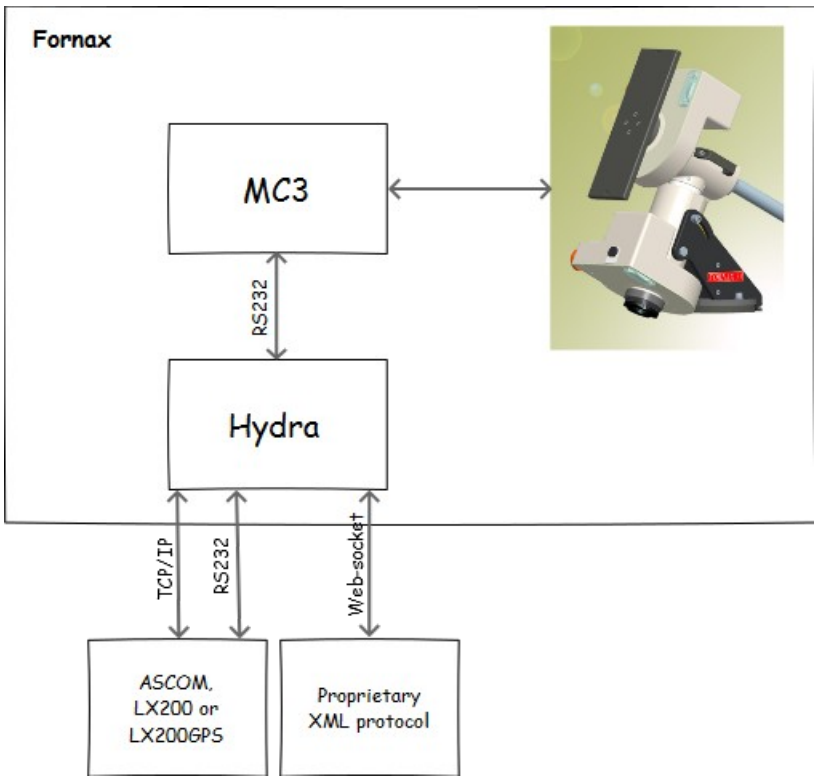
Table of Contents

Overview of the Hydra system....3
Basic terms....4
Setting up the system....6
Connections....6
Network setup....6
Parameters....11
Site parameters....11
Mount parameters....12
LX200 connection parameters....14
Global parameters....14
Web-server parameters....14
Obtaining the calibration constants of the mount....14
Operation....16
TheSkyX session....17
SkySafari session....19
Methods....23
Properties....23
Events....24
CompletedCommand....24
StateTransition....25
Diagnostics....25
Status codes....25
Information codes....25
Warning codes....25
Error codes....25
Appendix A – Assembly instructions....26
Appendix B – Supported Meade commands....27
C – Sync Control....27
G – Get Telescope Information....27
h – Home Position Commands....28
H – Time Format Command....29
M – Telescope Movement Commands....29
Q – Movement Commands....29
R – Slew Rate Commands....30
S – Telescope Set Commands....30
T – Tracking Commands....32
U – Precision Toggle....32
W – Site Select....32
Appendix C – XML based protocol definition....33

Overview of the Hydra system

Hydra is a Raspbery PI based software suite which controls a German equatorial telescope mount via an MC3 electronics unit. Hydra can operate in two independent modes. One is mainly for the amateur while the other is for the professional astronomer.

In its first form Hydra emulates the Meade LX200 (or LX200GPS) protocol either over an RS232 serial line or a multi-threaded TCP/IP connection. This enables PC, iPhone or Android mobile phone based planetarium programs to control Fornax (http://fornaxmounts.com/) mounts. The professional version uses a proprietary XML protocol over web-sockets. This later solution is for observatories where control with a much higher reliability (e.g. watchdogs, events) is necessary. The schematic drawing of the system is as follows:

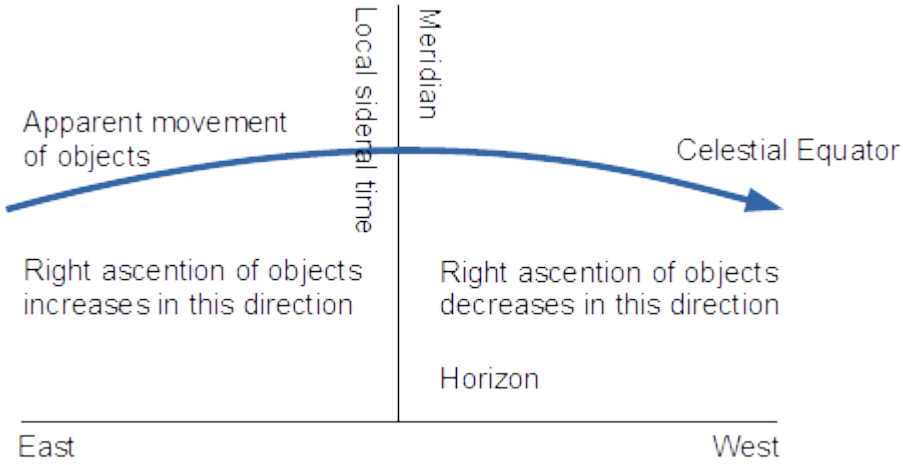


Mounts can be built with or without an absolute encoder. Mounts with no encoders can execute LX200 commands only. These mounts can not be used with the professional interface. For mounts built with encoders on both axes, Hydra enables the interpretation of those LX200GPS commands which can park and home the mechanics and also the usage of the professional interface. The following table shows which extra features come when the mount is equipped with an encoder.

Feature	Encoder present	No encoder
Park telescope	Yes	No
Home (calibrate) telescope	Yes	No
Definition of dead-zones	Yes	No
Physical end-sensors	Yes	No
Extended mount states (see later)	Yes	No

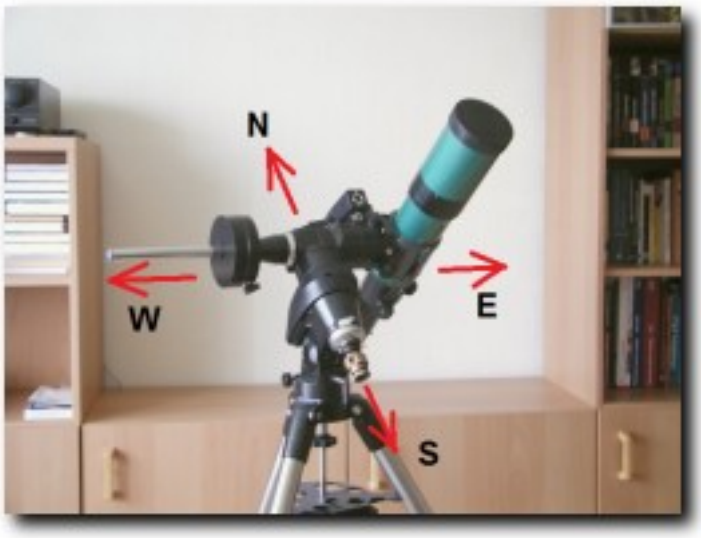
Basic terms

Hydra is a bridge between PC based planetarium programs (or control systems) and an MC3 telescope control unit. To better understand its operation and the meaning of each configuration parameter we will discuss shortly some basic terms we will use in the next paragraphs.



The figure above illustrates and explains our basic terms for the northern hemisphere. On the southern hemisphere the apparent movement of objects, East and West and the RA directions are mirrored. Right ascension (RA) of celestial objects on the meridian equal to the local sidereal time. Declination is measured from the Celestial equator and it is positive towards the north pole and negative towards the south pole.

When we track an object, its RA does not change but its hour angle increases as it moves to west. German equatorial mounts have two „standings”. East stand, when the telescope is at the eastern side of the RA axis, while the west standing is when the telescope is on the opposite side.



The main difference between the two modes is that North up (the direction where declination is increasing) is mirrored between them. When the telescope is turned from one side to the other is called meridian flip. **Meridian flip is handled by Hydra only if you start up your equipment in East stand!** Later, Hydra itself will change the side on demand, but the user can not change it by hand. Any operation which is outside the knowledge of Hydra can lead to unwanted operation, and the electronics might turn the scope into the pier. You can also avoid physical contact between your telescope and pier by using an angled pier.

Using a professional mount with absolute encoders and physical end-contacts will need more parameters than simpler backyard mounts. In the next section we will have a look at the web-interface which is used for setting up the different parameters of your mount. These parameters are important because the movement ranges of the mount are limited by the entered values. If you use wrong numbers your mount will either have a smaller range of motion or will try to move outside of the physical limits and brake or cause other damage. **Please note that setting the right values is your responsibility!** Hydra will give you all the help to set these values correctly (see next chapter).

Setting up the system

Connections

The following figure shows all connections Hydra has.



From left to right:

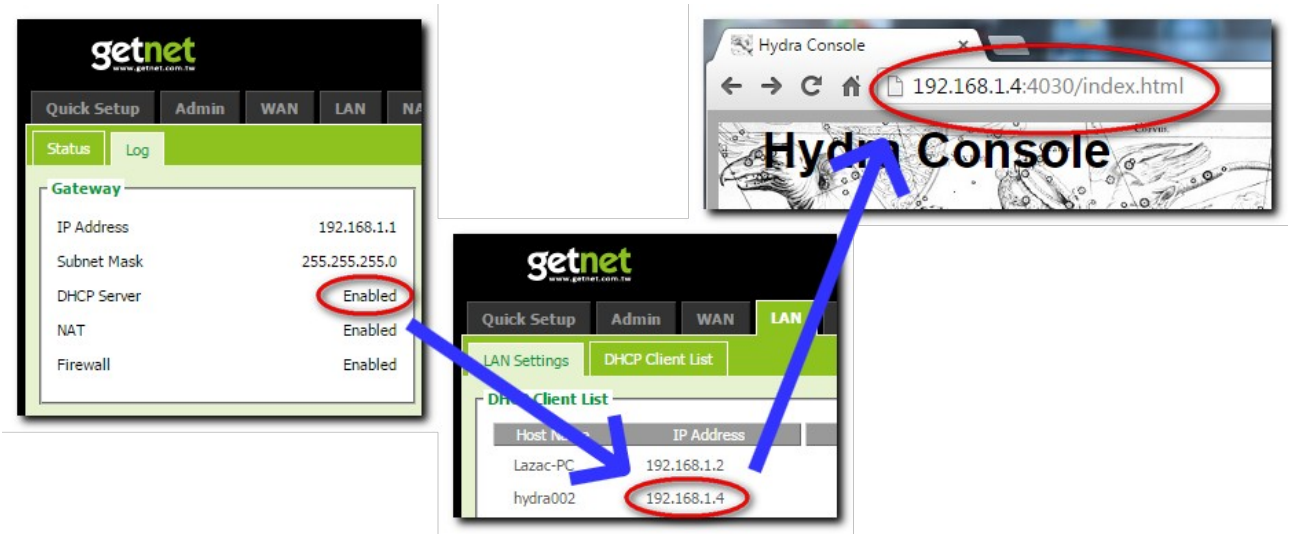
- RCA socket accepting the 12V DC input (plus power indicator LED).
- RJ45 network connector to link Hydra with your router (please see the next paragraph about network setup).
- USB socket for the PC connection. Please use the supplied USB to serial cable to connect to your PC (running your favorite planetarium program) or any other Prolific PL2303 chip based adapter.
- USB socket to connect MC3 to Hydra.
- USB socket for the WiFi dongle.
- Free USB socket (can be used for an optional web camera).
- RJ45 socket for native MC3 connection. When you use this connector, MC3 supplies the necessary 12V power for Hydra and the RCA socket should not be connected.
- Status LEDs (please read-on for a detailed explanation).

Network setup

Hydra comes with a pre-installed operating system and application software. All default parameters are already set, but of course you have to customize them to your needs. The easiest way to do it is via its built-in web-server.

Hydra is configured to use DHCP to require an IP address when connected via the RJ45 socket. For this to work, your home network router should also be configured for DHCP (Dynamic Host Configuration Protocol). If your router is not configured for DHCP, then connect to its management console (with a browser like Chrome or Firefox) using a different computer that is already connected and enable it.

Then connect Hydra directly into your network router with an RJ45 cable. On your router's management console search for the IP address Hydra has just received after being connected to the router. Now from the browser connect to Hydra with the following URL: http://ip_from_the_router:4030/index.html.



Hydra also configures a so called “ad hoc” network (with DHCP) over its WiFi interface. This interface can be used by mobile devices under the heavens, where no router is available. Parameters are as follows:

IP:	192.168.42.1
DHCP range:	192.168.42.10 – 20
SSID:	Hydraxxx
Password:	12345678

Once you have setup Hydra, you have to reboot it. Hydra will automatically use either the wired or the WiFi interface for TCP/IP connections, depending on which have a valid network IP address. If both cable and WiFi channels have valid (but different) addresses, Hydra will open all communication channels on the wired connection and no service will be available on the WiFi, or the other way round. This means that you can not connect to Hydra via RJ45 and WiFi at the same time.

When Hydra is rebooted it will initialize all connections (host, MC3, etc) and indicate the overall status of the device with the two status LEDs. The following table shows their possible states:

Red	Green	Meaning
Off	On	Normal operation
Off	Blinks 4/sec	SD card problem
Blinks 4/sec	On	No communication with the mount
Blinks 1/sec	On	Mount is in FAULT state
On	Blinks 4/sec	No communication with the host
On	Blinks 1/sec	Error starting the web-server

Please make sure that all electronic components are powered up before Hydra. When the LEDs indicate that no error happened during start-up you can connect with the URL mentioned above. After successful connection the following page will appear.

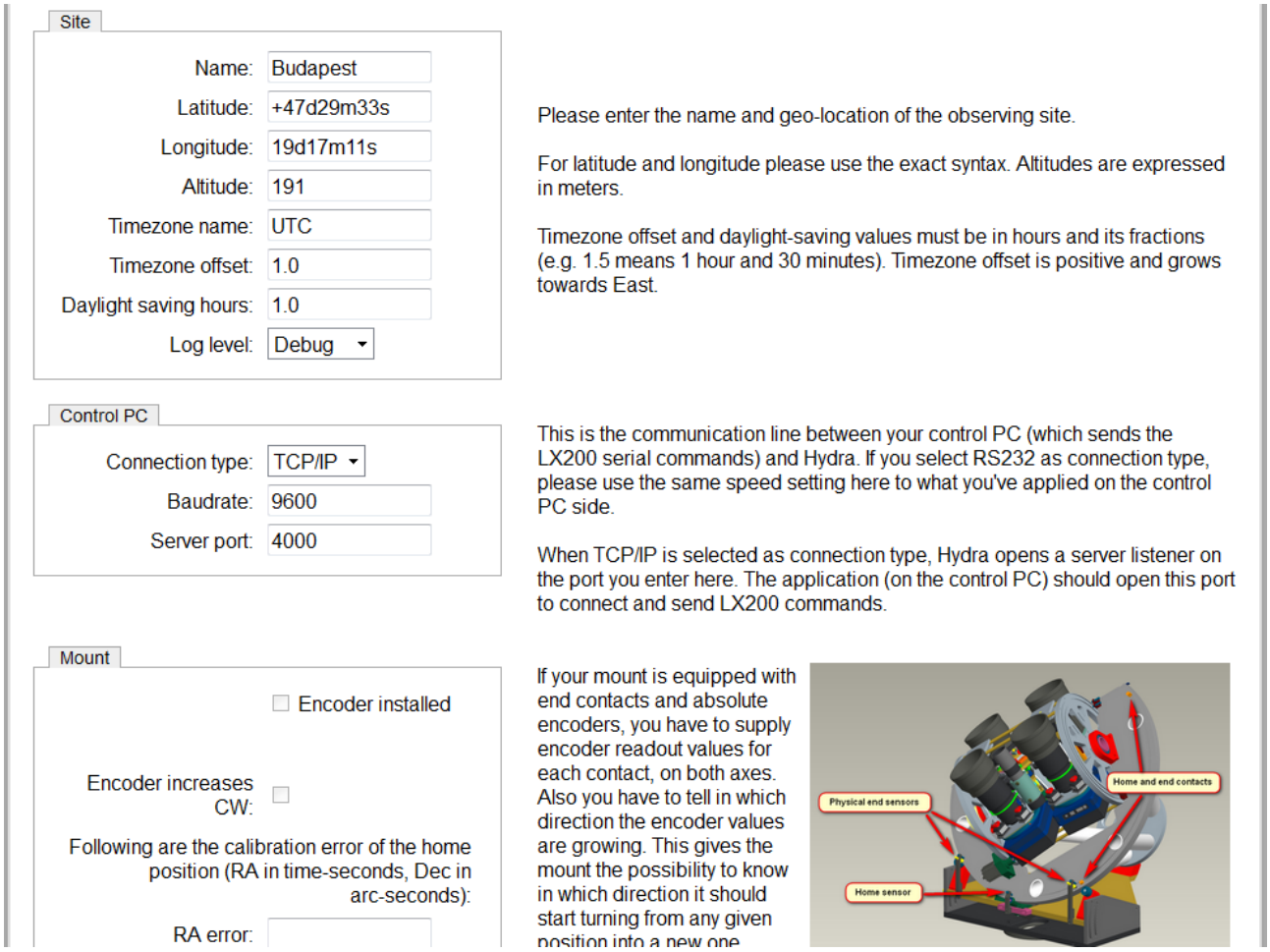


The upper right corner contains the site parameters. The rest of the page contains four tabs. The first one shows the actual overall status of the mount, which refreshes every 5 seconds. The second page is used for viewing or setting the values of the operation parameters. The third one is for viewing the logs of the Hydra system, while the last one is for taking snap-shots with an optionally attachable web-camera.

The following image shows a partial dump of the configuration page. To obtain the encoder values for the various positions proceed as follows:

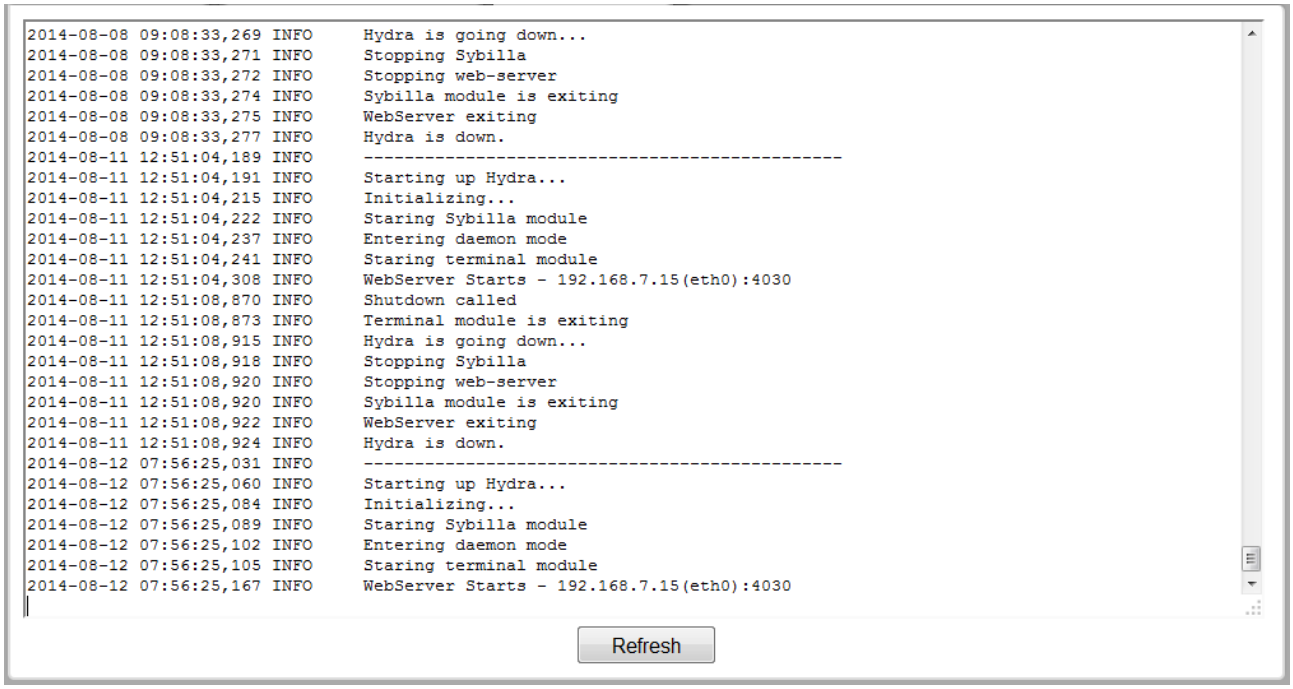
- turn the mount into the wanted position
- on the web-page click on the **Status** tab and read out the encoder values in parentheses right after the RA and Dec values
- click back to the **Configuration** tab and enter the previously read value(s) into the appropriate input cell(s)

If you scroll down the Configuration page you can find the two buttons to save or revert your changes.



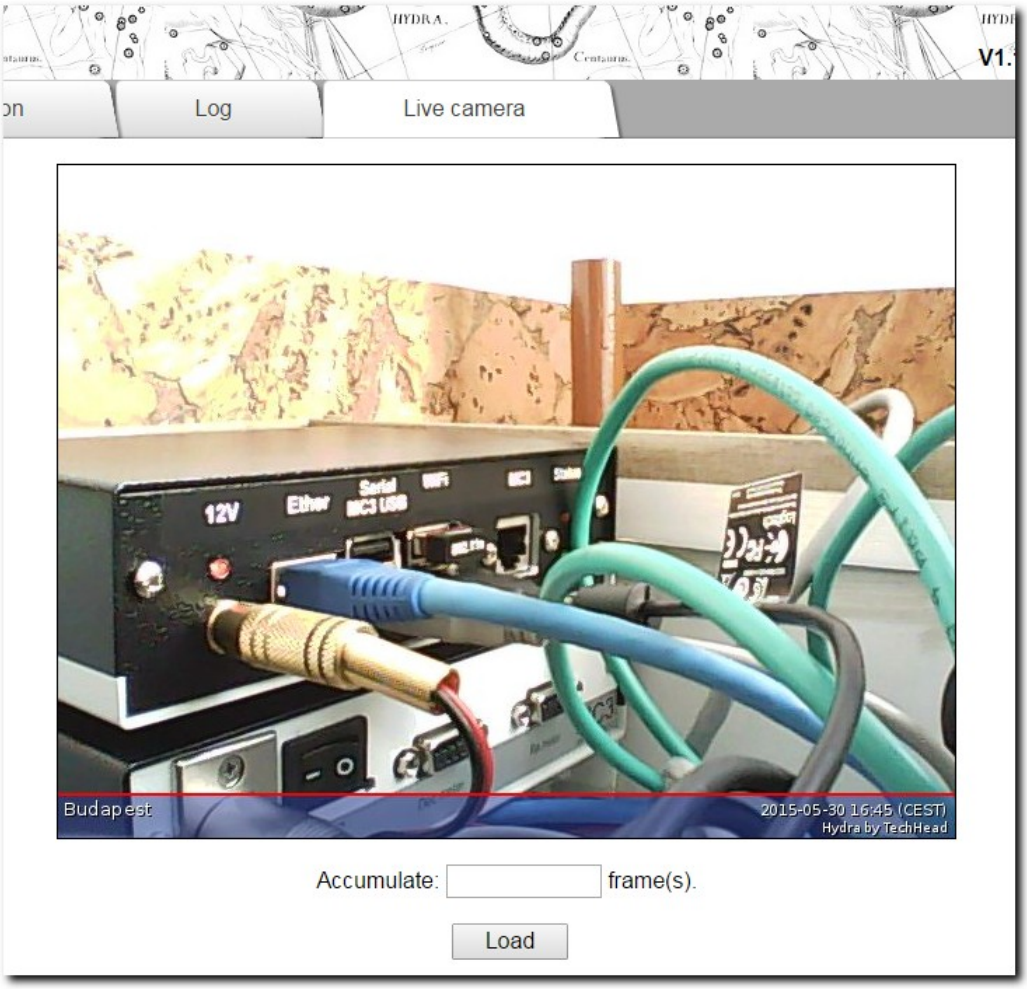
You can also upgrade the firmware in Hydra via the configuration web-page. First download the latest firmware and place it on your local drive. Then upload it to Hydra. In case the upload and installation were successfully executed you can reboot Hydra to start the upgraded version.

The log tab-page looks as follows:



For each push of the „Refresh” button the last 500 lines of the log-file is displayed in the page.

The next tab delivers still images from the optional web-camera. Hydra automatically detects USB web-cameras attached to the fourth USB socket. Please note that not all cameras are supported by the device. For the list of supported web cameras please visit http://elinux.org/RPi_USB_Webcams. If you want to attach the device directly into Hydra please choose one which can run without an external power HUB. Images are delivered only when the **Refresh** button is clicked. No continuous or streaming images are supported currently. During the night or under low-light conditions the web-camera can not deliver clear images. For this reason you have to ask the camera to integrate multiple images into one single image. The quality of the resulting image depends on many factors, therefore no exact values can be given for the number of frames. Please enter different numbers into the entry field and experiment until the resulting image is satisfactory.



Parameters

The following tables explain the parameters which are necessary to operate your mount.

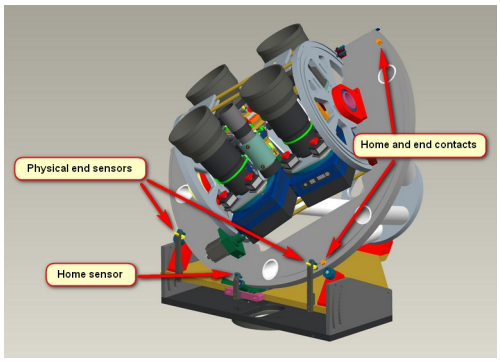
Site parameters

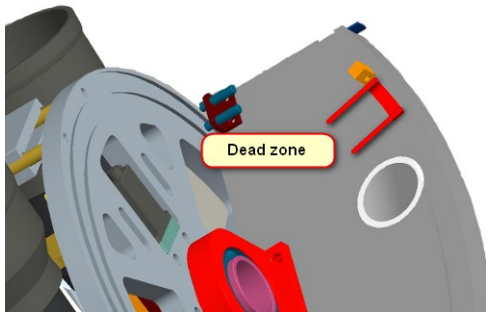
These parameters define the geographical location of the mount for proper computation of the local sidereal time.

Name	Description	Type	Possible values
name	defines the name of the site	string	
latitude	latitude of the site	+47d29m33s	
longitude	longitude of the site	19d17m11s	
altitude	altitude of the site, in meters	floating point	
tz_offset	time zone offset, in hours	floating point	
dst_offset	daylight saving offset, in hours	floating point	

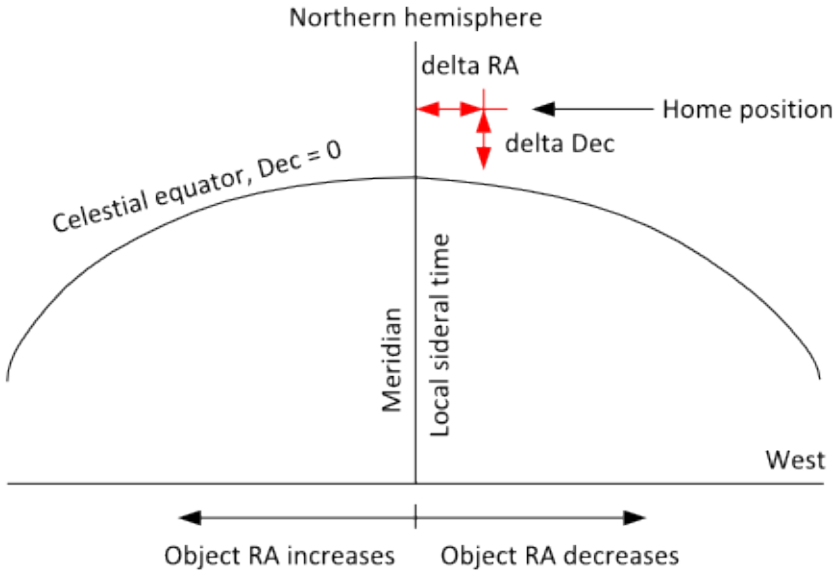
Mount parameters

When the mount is equipped with absolute encoders and end-sensors a few parameters must be given to define the range of motion and dead-zones (forbidden zones).





For observatory mounts it is recommended that the mount is calibrated with precise astrometric measurements. Normally we start our nightly observations by pointing the mount's RA axis to South and the Dec to the celestial equator. Later we will refer to this point as Home position. The encoder readout for this position can be obtained from the web-page of Hydra. From astrometry the coordinates of this point can be computed. With these values the mount can be calibrated any time. A possible calibration method is described later in this document.



Thus, the „transformation” of the mount into the celestial coordinate system is reduced to

the simple process of homing the mount into a known position.

Name	Description	Type	Possible values
hasEncoder	tells that the mount is equipped with an encoder	boolean	True, False
encoderIncrWest	tells that the positive encoder readout along the RA axis increases to West	boolean	True, False
encoderIncrNorth	tells that the positive encoder readout along the DEC axis increases to North	boolean	True, False
encoderBitsRA	precision of the RA encoder in readout bits	integer	
encoderBitsDEC	precision of the DEC encoder in readout bits	integer	
homePositionRAError	RA calibration error of the home position in time seconds. Positive on the EAST side of the meridian	floating point	
homePositionDECError	DEC calibration error of the home position in arc seconds. Positive if it points above the celestial equator	floating point	
homePositionRA	RA axis encoder readout for the home position	integer	
homePositionDEC	DEC axis encoder readout for the home position	integer	
parkPositionRA	RA axis encoder readout for the park position	integer	
parkPositionDEC	DEC axis encoder readout for the park position	integer	
eastEndPosition	RA axis encoder readout for the east end position	integer	
westEndPosition	RA axis encoder readout for the west end position	integer	
northEndPosition	DEC axis encoder readout for the north end position	integer	
southEndPosition	DEC axis encoder readout for the south end position	integer	
safetyZoneLength	length of the safety (dead) zone in encoder unit	integer	

Encoder readout values for the above mentioned positions can be obtained from the web-page of Hydra. Simply turn the mount into the wanted position and read the corresponding encoder value from the web-page (value after the RA and DEC).

LX200 connection parameters

Parameters below are used for telling Hydra which connection type is used for LX200 command reception. E.g. if you want to control the mount via SkySafari (WiFi connection) you have to choose the TCP/IP type. When your planetarium program runs on a laptop and connection is made via RS232, please select RS232 as type. The serverport value must be equal to the value you use in your mobile device as server port.

Name	Description	Type	Possible values
conntype	defines the connection type for receiving LX200 commands.	string	RS232, TCPIP
serverport	when TCPIP is selected as connection type, this parameter defines the port on which Hydra is listening	integer	> 1000
baudrate	speed of the serial line	integer	1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200

Global parameters

Currently only one global parameter can be set, the granularity of the logging.

Name	Description	Type	Possible values
loglevel	granularity for logging	string	DEBUG, INFO, WARN

Web-server parameters

Name	Description	Type	Possible values
port	port number where the web-server is listening for incoming websocket connections	integer	> 1000, but different from the serverport value, default value is 4030

Obtaining the calibration constants of the mount

As we mentioned earlier, the real sense of having encoders on the mount is to turn it into a robotic telescope. If we can send the mount to a known position and we know the mount's geographic location and local time at the same time we can compute the local sidereal time and thus “transform” the mount into the celestial coordinate system. When it is done, the mount can be commanded to any coordinate. With the following few steps you can go through the calibration phase. Our goal is to learn the ΔRA and ΔDec constants of the Home position.

1. Command the mount to go to the Home position and start tracking immediately. The pointing of the telescope is now the RA and Dec of the Home position.
2. Start taking an image of this field of view through your telescope.

3. Go into the web-pages of Hydra and click on the log tab and then on the **Refresh** button. Page down to the end and from the bottom up search for the **LOCAL SIDEREAL TIME FOR CALIBRATION** string. The indicated time is the LST for the exact instance when Hydra started to track at the Home position. This is/was the RA of the meridian when we started to track. Call this value LST(cal).
4. From the image taken, work out the exact RA and Dec coordinates of the center of the field (using astrometry). Those values will be the equatorial coordinates of our pointing (also at the start of tracking). Call these values as RA(center), Dec(center).
5. Now subtract RA(center) from LST(cal) to get the calibration constant for the RA of the Home position. Convert the difference to time seconds and its fractions. The Dec calibration constant is simply the Dec(center) value, which should be converted to arc seconds and its fractions.

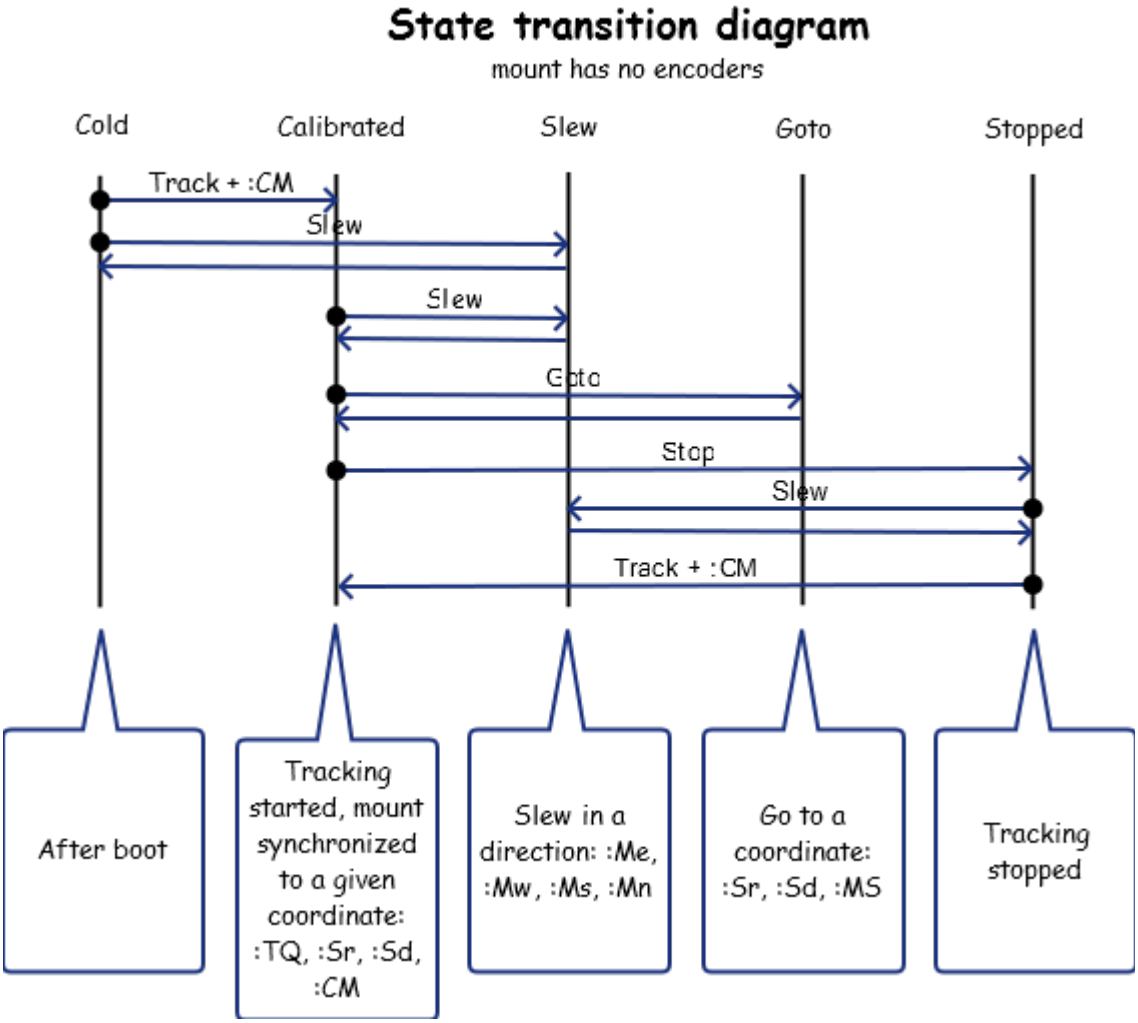
   Let's look at an example:

LST(cal):	02:15:12.44
RA(center):	02:18:15.31
Difference:	00:03:02.87, in time seconds: 182.87
Dec(center):	-00:10:10.23, in arc seconds: -610.23

Enter the values into the web-page. Next time when Hydra homes, it will use these constants to compute the celestial coordinates of the Home position.

Operation

The mount operates in definite states. While it is in one state another state can only be initiated via commands. When the mount has no encoders the state diagram is as follows:

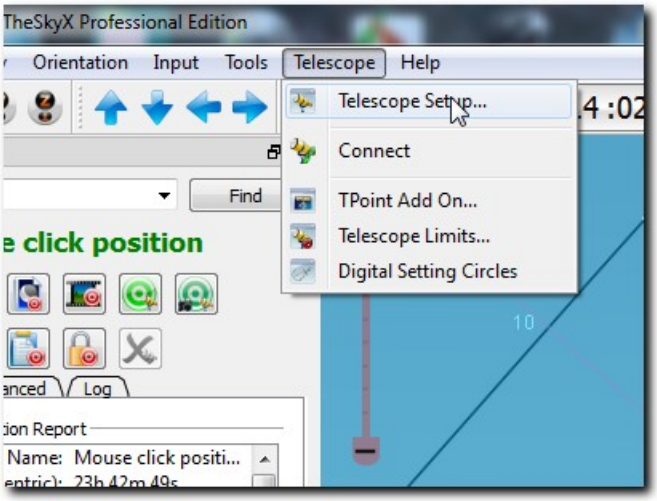


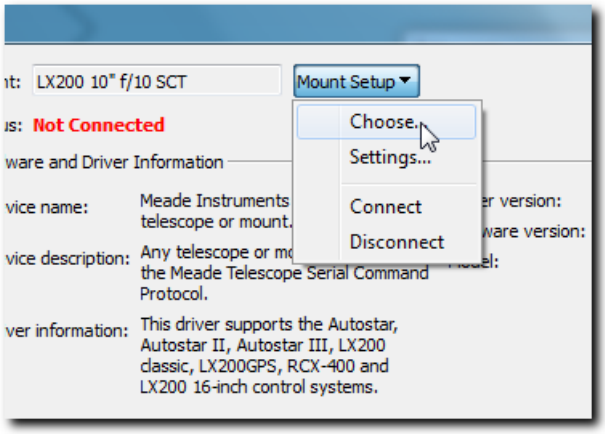
After booting the electronics the mount (or Hydra) is in Cold state. To reach the Calibrated state – to start observing – the mount should be transformed into the celestial coordinate system. Taken that the mount has been Polar aligned we only need to start tracking and to tell the coordinates or the current pointing. After this point, the mount can be sent to any coordinate point on the sky (Goto) or the user can slew in any direction (Slew). After the movement stops the mount remains in Calibrated state and its current coordinates tell its real pointing. If we stop the tracking at any point, the pointing will no longer tell the real coordinates (Stopped) and the mount must be re-calibrated before initiating any other goto operation. To make the calibration easier we allow that the mount is slewed into a position (to point to a star).

TheSkyX session

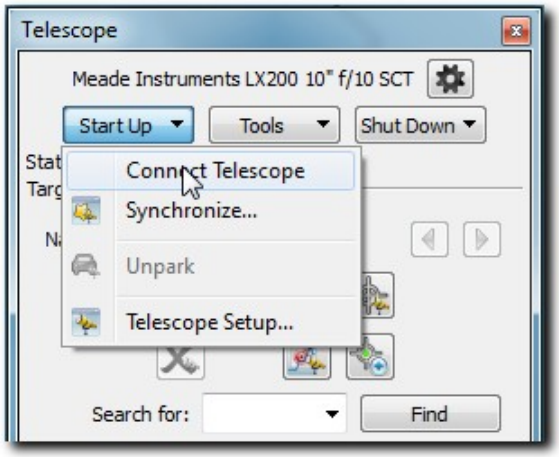
Let's have a look how the TheSkyX planetarium program from Software Bisque can be used with Hydra. Align your mount and telescope so that the telescope it at the eastern side of the mount and points to the South, towards the celestial equator. We assume that you have previously polar-aligned it accurately enough for astronomical work.

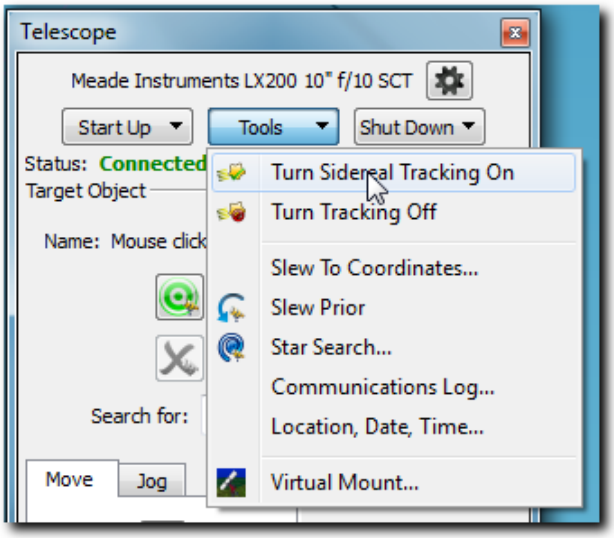
1. Connect your PC with Hydra and start up TheSkyX and all electronics. Check in Windows which COM port has been assigned to the USB-to-Serial connection.
2. Open the **Telescope > Telescope Setup** dialog. Click **Choose...** on the **Mount Setup** drop-down, and select the LX200 protocol. Then click on **Settings...** and select the COM port of the USB connection. We are now ready to connect TheSkyX with Hydra.



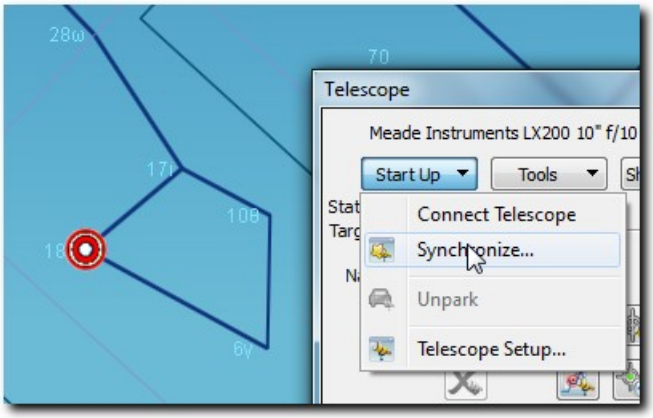


3. In the **Telescope** dialog click on **Connect Telescope** from the **Start Up** drop-down. After a successful connection the Status turns to: **Connected**. From the **Tools** drop-down select **Turn Sidereal Tracking On**. You should hear now the stepper of your mount tracking the stars.



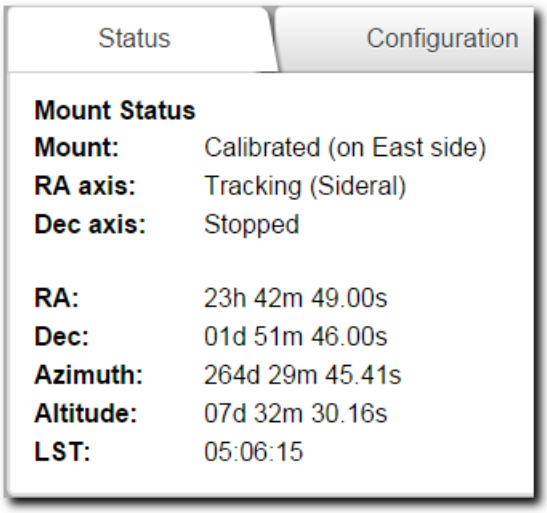


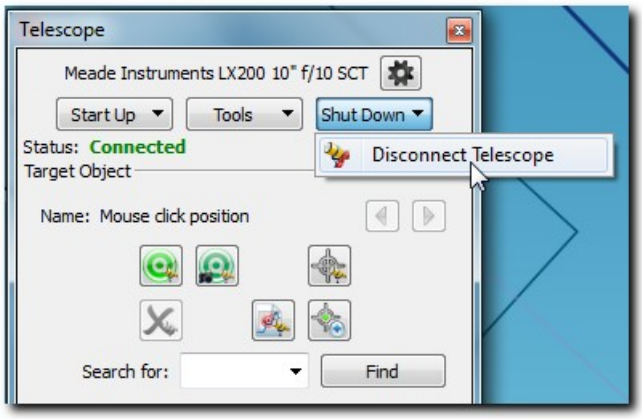
4. With the next step we tell Hydra to which equatorial coordinate the telescope is pointing. This allows us later to send the telescope to any other object simply by clicking on it in TheSkyX. For this, center a nearby star in the scope and find the same star in TheSkyX. Click on this star and then select **Synchronize...** from the **Start Up** drop-down. This brings up the synchronization confirmation dialog. Click on **Yes** to synchronize the telescope's current pointing with the planetarium programs pointing.





5. From this point on Hydra and TheSkyX are in the same coordinate system. Any movement in one will result with the same movement in the other. You can check it by opening the web-page of Hydra. This synchronization can be done any time or any number of times later e.g. when you move the mount by hand.
6. When you are finished with your work you can disconnect Hydra from the program by selecting **Disconect Telescope** in the **Shut Down** drop-down.



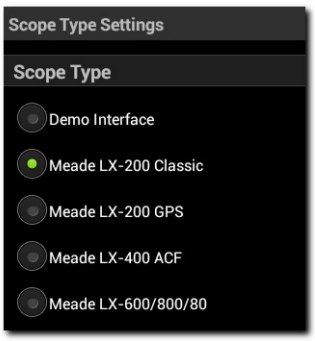


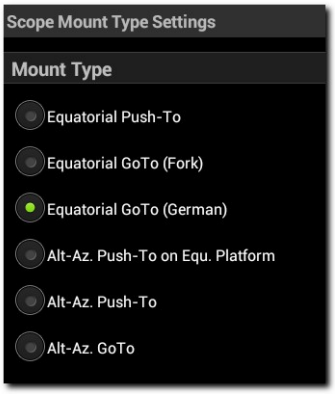
SkySafari session

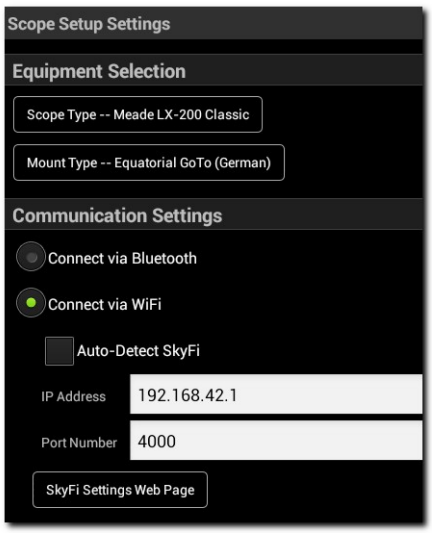
When Hydra is booted with the WiFi dongle (and no RJ45 connection), it automatically starts an ad-hoc wireless network with the fix 192.168.42.1 address. All clients connecting to it will receive a DHCP IP address in the 192.168.42.10 – 192.168.42.20 range. Please select the Hydraxxx Wifi network on your mobile device. For password enter 12345678. After establishing connection to it, configure the host connection to TCP/IP (if it is not already been done). The server port is 4000 by default but you can enter any other number. Be careful to assign a number above 1000 and a different one from the web-server (which is 4030 by default). When this has been done Hydra will receive all incoming commands via the TCP/IP channel.

Start SkySafari and go into the **Setup** page.

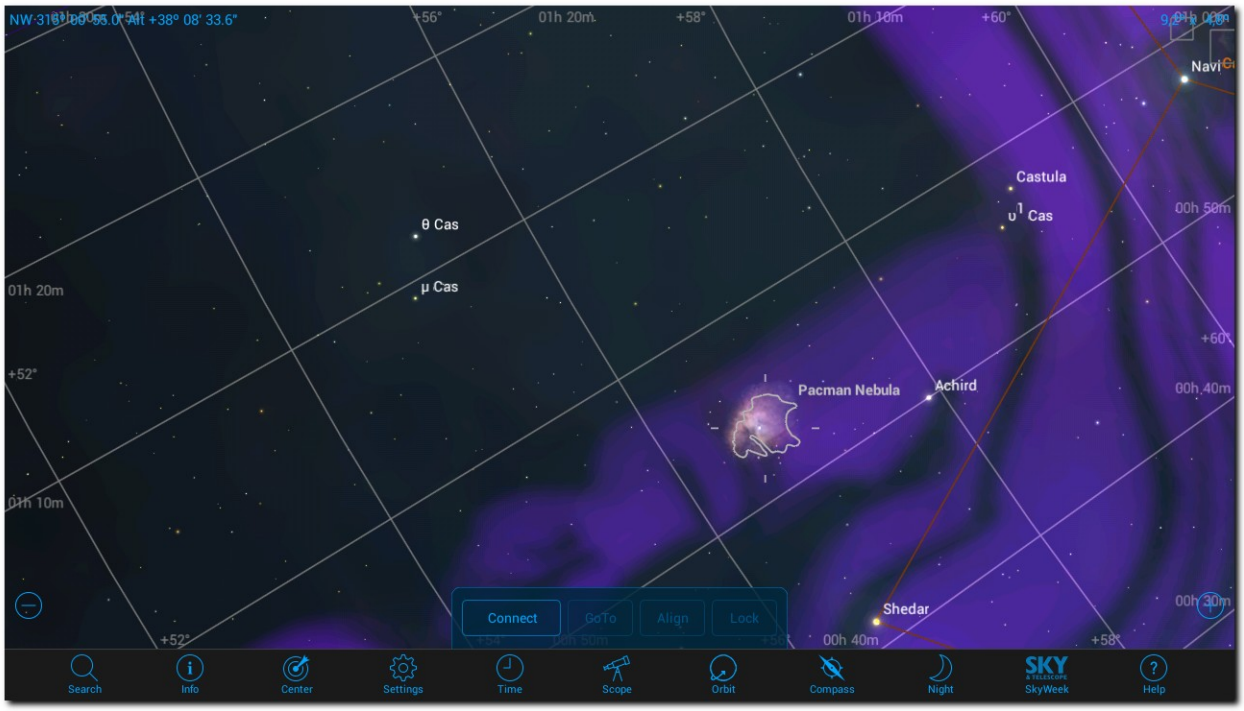
1. Choose **Meade LX-200 Classic** and scope type
2. Choose **Equatorial Goto (German)** as mount type







3. In the **Communication Settings** part choose **Connect via Wifi** and enter 192.168.42.1 for IP address and the port you have entered for the host connection or 4000 if you haven't changed the default value.
4. Exit setup, tap on the telescope icon, then on **Connect**.



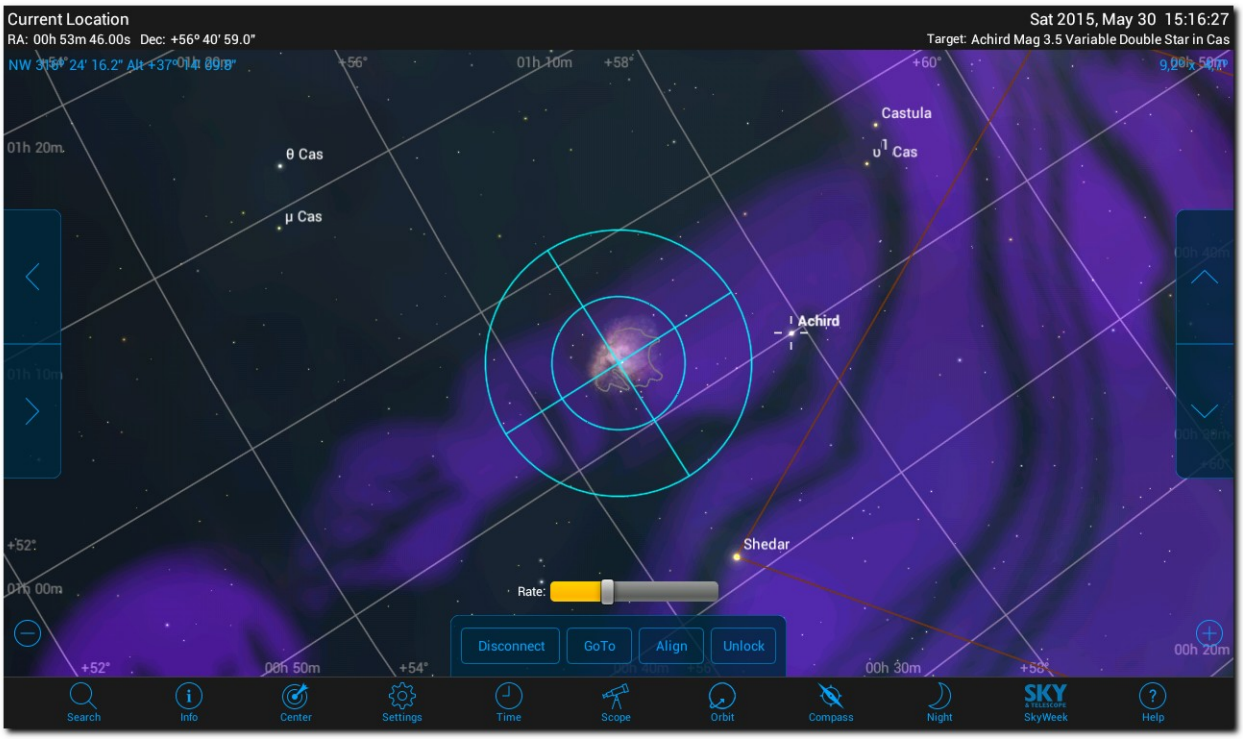
5. When the connection is made, SkySafari will start reading our the current pointing of Hydra. After boot, the pointing will be zero, which brings SkySafari to select the RA zero and Dec zero position. To calibrate your mount, and SkySafari point your telescope (with the polar aligned mount) to a known object and bring the same object in SkySafari into the middle. Select it by long tapping on it and then tap on the **Align** button. Choose **Align** in the dialog and the mount will start tracking and after this point Hydra and SkySafari are calibrated to the same object.



During the night (or when you accidentally moved the telescope by hand) you can repeat this process to re-align SkySafari and Hydra.

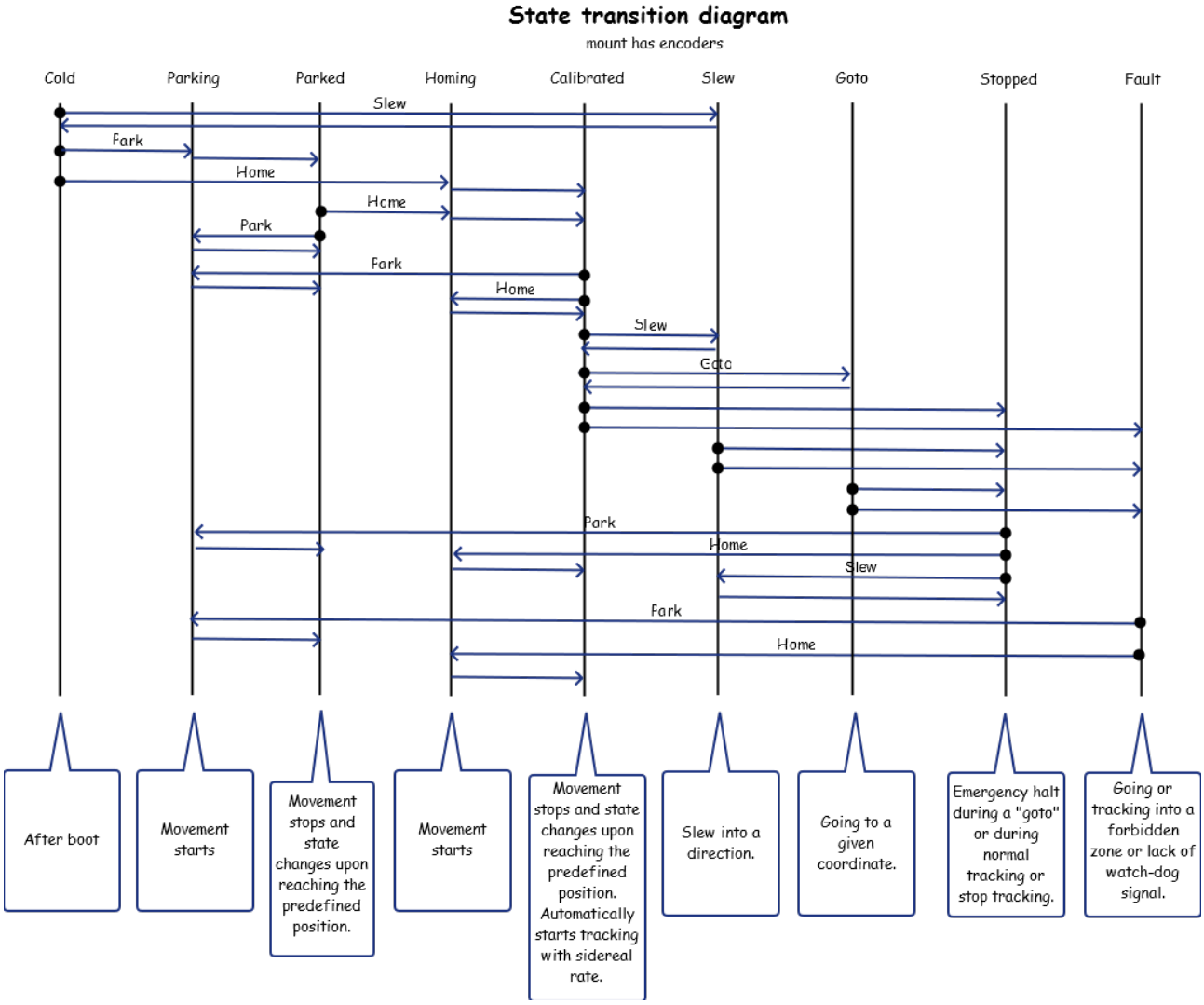
6. To slew the telescope use the four arrow buttons on the side of the screen. To move

to objects, search for them or long-tap on them when already on screen and use the **Goto** button to command Hydra to move the mount to the selected object.



7. At the end **Disconnect** from Hydra.

In case the mount is equipped with absolute encoders it has more states, because it can park, home and because of the end-sensors it can reach dead-zones. The state transition diagram for such a mount is as follows:



The explanation for the above diagram is analogous to the one without encoders.

When the mount is used in a more demanding scientific environment, a special communication can be used for control. The need for such a protocol comes from the fact that ASCOM and Meade LX200 does not allow fault tolerant, event-driven autonomous operation of the mount. The main features of this design are:

- XML based data format.
- WebSocket based communication.
- Asynchronous and event-driven architecture.
- Fault tolerance.
- Client-server architecture.

Fault tolerance will be achieved by implementing a watchdog pattern on the Hydra side, waiting for the watchdog reset method executed remotely from the active PC. All commands are called “methods” and all answers as “events” in the upcoming paragraphs. Their exact format is defined in Appendix C.

Methods

Methods are asynchronous when their execution requires more than 10 ms. Timeout of the method is the maximal execution time of the method plus the round-trip for the message going across the network (all communication happen through the Local Area Network (LAN), so 1 s is assumed for the message’s round trip). The following methods/commands can be sent:

- **GetMountProperties**: Return current state of the mount
- **GoToAltAz**: Initiate movement of the mount from current position to given altitude and azimuth coordinates
- **GotoCoordinates**: Initiate movement of the mount from current position to given right ascension and declination coordinates for given epoch
- **Home**: Move the mount to the home position, calibrate its position and start tracking
- **Park**: Move the mount to the park position and stop all movement
- **Stop**: Stop the mount’s movement (i.e. stop tracking or slewing)
- **ResetWatchdog**: Reset the watchdog in Hydra

Apart from the Stop and GetMountProperties methods, all other methods are asynchronous.

Properties

Properties are the parameters used by methods to pass values between Hydra and the controlling application. The are as follows:

Name	Description	Type	Range	Default
altitude	Current altitude the mount points at	degrees, double	-90.0, + 90.0	0.0
azimuth	Current azimuth the mount points at	degrees, double	0.0, + 360.0	0.0
declination	Current declination the mount points at	degrees, double	-90.0, + 90.0	0.0
rightAscension	Current right ascension the mount points at	hours, double	0.0, +24.0	0.0
targetAltitude	Target altitude the mount should point at	degrees, double	-90.0, + 90.0	0.0
targetAzimuth	Target azimuth the mount should point at	degrees, double	0.0, + 360.0	0.0

targetDeclination	Target declination the mount should point at	degrees, double	-90.0, + 90.0	0.0
targetRightAscension	Target right ascension the mount should point at	hours, double	0.0, +24.0	0.0
isConnected	Is the mount controller connected to the mount	boolean	true, false	false
isHomed	Is the mount homed and calibrated	boolean	true, false	false
isParked	Is the mount at park position and in park state	boolean	true, false	false
isSlewing	Is the mount slewing (executes GoToAltAz, GoToCoordinates or any other movement command)	boolean	true, false	false
isTracking	Is the mount tracking with sidereal speed	boolean	true, false	False
name	Name or the id of the mount	string	3-10 characters	Fornax-000
watchdogTimeout	Amount of time in which watchdog should be reset. If not the mount will go into safe state	duration (example: PT00M50.0S)		PT0.0S
remainingTimeUntilLimit	Amount of time until the RA tracks into the safety zone	duration		

Events

Each execution of a method, internal state changes or other warnings produce events. They describe the outcome of a command, either success or failure. The returned id is a new Globally Unique Identifier (GUID) if the event was created by Hydra or is a received GUID if the event is a result of a received (and asynchronously executed) method. We distinguish three types of events: **CompletedCommand**, **StateTransition** and **Diagnostics**. Events are published to each connected client.

CompletedCommand

Produced by Hydra as a result of an incoming command, the id of received command is the **referenceId** of the **CompletedCommand**.

StateTransition

Produced by Hydra as a result of a successful or failed state transition. It is generated at the end of transition and provides the information about the initial state and new state. If the new state is different than the target state the event signals a fault.

Diagnostics

Event produced by Hydra to inform the clients about possibilities of errors, faults or about the device state. For example it can warn clients that the mount is close to its tracking limits and that in 10 minutes it will reach it.

Status codes

Status codes are returned in three categories: info, warning, error.

Information codes

Codes that are used for information purposes only, for example that asynchronous method execution completed successfully:

0 Unknown (The method returned but its return code is undefined.)
1 Success (The method completed with success.)

Warning codes

Codes that are used for raising a warning to the client, for example that asynchronous method could not be completed because of invalid input parameters that could for example result in mount pointing below horizon:

0 Unknown (The method returned but its return code is undefined.)
1 Object is below horizon
2 Object is above higher limit
3 Object is below the lower limit
4 Mount is not in proper state
5 Illegal/unknown command
6 Illegal command parameter, for example 32h55m13s for RA
7 XML parsing error
8 Mount would land in dead-zone

Error codes

Codes that are used for informing the client about device error (only the device failures are reported here):

0 Unknown (The method failed but its return code is undefined.)
1 Internal device error
2 Timeout

Appendix A – Assembly instructions

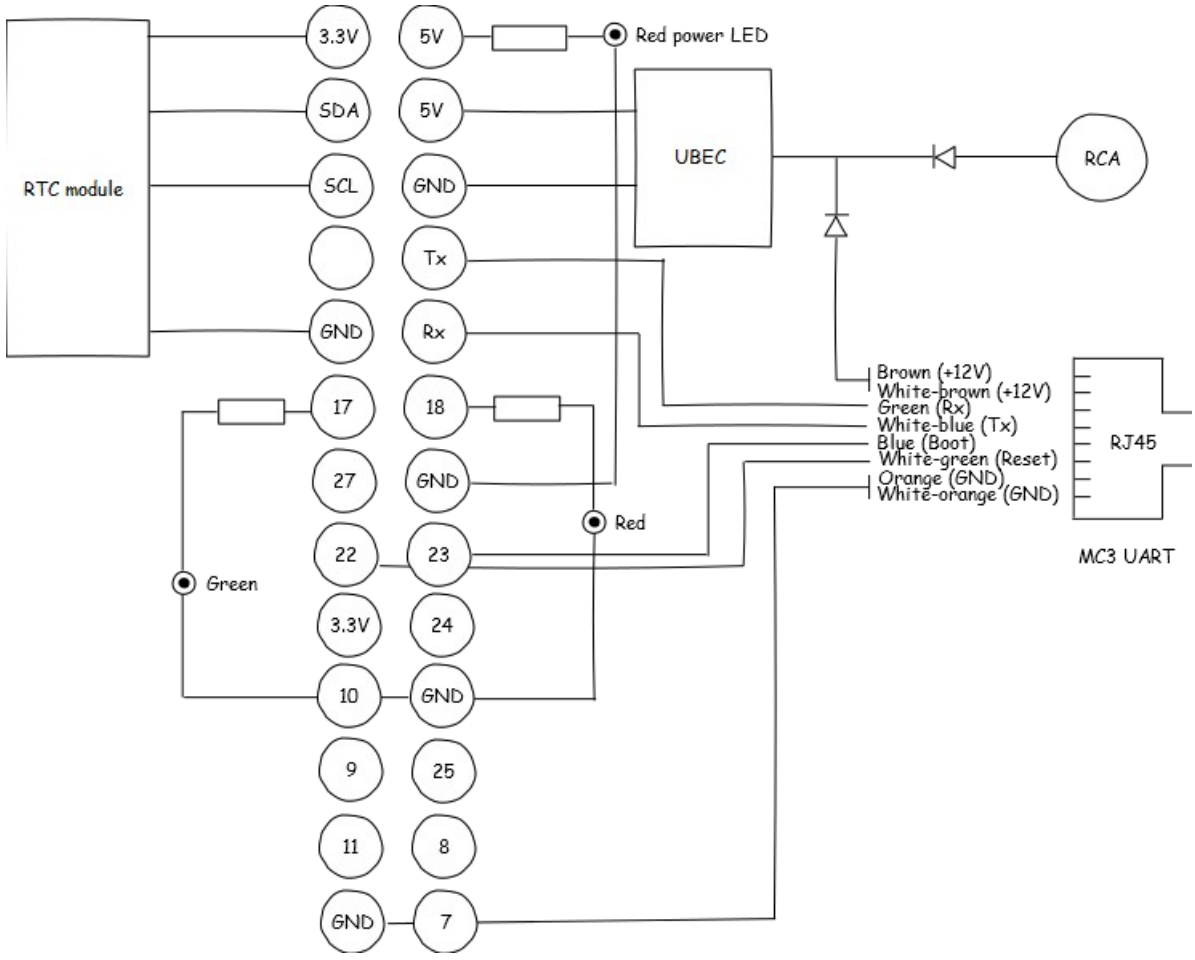
Hydra is built from commercially available stock components only. The whole system is built around a Raspberry Pi 2 B+ panel. The components are as follows:

- Raspberry Pi 2 Model B+ board computer 1 GB
- 8GB microSD memory card with installed Raspbian OS
- 150Mbps USB WiFi nano adapter
- I2C Real Time Clock module
- G.T Power UBEC 3A

To assemble the whole system you will also need:

- Jumper cables to attach the parts onto the GPIO head
- One green and two red LED + resistors
- One RCA female for the input 12V
- RJ45 feed-through cable coupler

Attach the parts to the GPIO head as follows:



NOTE: To install the Hydra system on the SD card please pick up contact with the author!

NOTE: The RTC unit is equipped with separate battery which should be replaced after 5 years after purchase!

Appendix B – Supported Meade commands

Hydra supports either an LX200 or an LX200GPS German mount emulating the below listed commands. Any other, not listed commands are not supported.

C – Sync Control

:CM# Synchronizes the telescope's position with the currently selected database object's coordinates.
Returns: Static string: "M31 EX GAL MAG 3.5 SZ178.0'#"

G – Get Telescope Information

:GA# Get telescope altitude
Returns: sDD*MM# or sDD*MM'SS#

:Ga# Get local telescope time in 12 hour format
Returns: HH:MM:SS#

:GC# Get current local calendar date for the telescope.
Returns: MM/DD/YY#

:Gc# Get calendar format
Returns: 12# or 24#

:GD# Get telescope Declination.
Returns: sDD*MM# or sDD*MM'SS#

:Gd# Get currently selected sbject/target Declination
Returns: sDD*MM# or sDD*MM'SS#

:GG# Get UTC offset time (The number of decimal hours to add to local time to convert it to UTC. If the number is a whole number the sHH# form is returned, otherwise the longer form is returned. On LX200GPS, the daylight savings setting in effect is factored into returned value.)
Returns: sHH# or sHH.H#

:Gg# Get current site longitude (East Longitudes are expressed as negative)
Returns: sDDD*MM#

:Gh# Get high limit (The minimum elevation of an object above the horizon to which the telescope will slew with reporting a "Below Horizon" error.)
Returns: sDD*

:GL# Get local time in 24 hour format
Returns: HH:MM:SS#

:GM# Get site 1 name (A '#' terminated string with the name of the requested site.)
Returns: <string>#

:GN# Get site 2 name
Returns: <string>#

:GO# Get site 3 name
Returns: <string>#

:GP# Get site 4 name
Returns: <string>#

:Go# Get lower limit (The highest elevation above the horizon that the telescope will be allowed to slew to without a warning message.)
Returns: DD*#

:GR# Get telescope RA
Returns: HH:MM.T# or HH:MM:SS#

:Gr# Get current/target object RA
Returns: HH:MM.T# or HH:MM:SS

:GS# Get the sidereal time (The sidereal time as an ASCII sexidecimal value in 24 hour format)
Returns: HH:MM:SS#

:Gt# Get current site latitude (Positive implies North latitude.)
Returns: sDD*MM#

:GT# Get tracking rate

Returns: TT.T#
Current Track Frequency expressed in hertz assuming a synchronous motor design where a 60.0 Hz motor clock would produce 1 revolution of the telescope in 24 hours.

:GVN# Get telescope firmware number
Returns: dd.d#

:GVP# Get telescope product name
Returns: <string>#

:GW# Get Scope Alignment Status
Returns: <mount><tracking><alignment>#
where:
mount: A-AzEl mounted, P-Equatorially mounted, G-german mounted equatorial
tracking: T-tracking, N-not tracking
alignment: 0-needs alignment, 1-one star aligned, 2-two star aligned, 3-three star aligned.

:GZ# Get telescope azimuth
Returns: DDD*MM#T or DDD*MM'SS#

h – Home Position Commands

:hF# LX200GPS: Seeks the home position of the scope and sets/aligns the scope based on the encoder values stored in non-volatile memory. NOTE: all commands except „:Q#" and „:h:#" are disabled.
Returns: Nothing

:hN# LX200GPS: Sleep telescope. Power off motors, encoders, displays and lights. Scope remains in minimum power mode until a keystroke is received or a wake command is sent.
Returns: Nothing

:hP# LX200GPS: Slew to park position
Returns: Nothing

:hS# LX200GPS:– Sets the current scope position as the park position of the telescope. Subsequent to this command, a :hP# command will drive the scope to this mount relative position.
Returns: Nothing

:hW# LX200GPS: Wake up sleeping telescope.
Returns: Nothing

:h?# LX200GPS: Query homing status
Returns:
0 Home search failed
1 Home found
2 Home search in progress
LX200 – Not Supported

H – Time Format Command

:H# Toggle between 24 and 12 hour time format
Returns: Nothing

M – Telescope Movement Commands

:MA# Slew to target Alt and Az
Returns:
0 – No fault
1 – Fault
LX200 – Not supported

:MgnDDDD#
:MgsDDDD#
:MgeDDDD#
:MgwDDDD#
Guide telescope in the commanded direction (nsew) for the number of milliseconds indicated by the unsigned number passed in the command. These commands support serial port driven guiding.
Returns – Nothing
LX200 – Not Supported

:Me# Move telescope East at current slew rate
Returns: Nothing

:Mn# Move telescope North at current slew rate
Returns: Nothing

:Ms# Move telescope South at current slew rate
Returns: Nothing

:Mw# Move telescope West at current slew rate
Returns: Nothing

:MS# Slew to target object
Returns:
0 Slew is possible
1<string># Object is below horizon w/string message
2<string># Object is below higher limit w/string message
4<string># Object is above the lower limit w/string message

Q – Movement Commands

:Q# Halt all current slewing
Returns: Nothing

:Qe# Halt eastward slews
Returns: Nothing

:Qn# Halt northward slews
Returns: Nothing

:Qs# Halt southward slews
Returns: Nothing

:Qw# Halt westward slews
Returns: Nothing

R – Slew Rate Commands

:RC# Set slew rate to Centering rate (2nd slowest, 16x sidereal)
Returns: Nothing

:RG# Set slew rate to Guiding rate (slowest, 2x sidereal)
Returns: Nothing

:RM# Set slew rate to Find rate (2nd fastest, 1 degree per second)
Returns: Nothing

:RS# Set slew rate to max (fastest, 4 degree per second)
Returns: Nothing

:RADD.D# Set RA/Azimuth slew rate to DD.D degrees per second [LX200GPS Only]
Returns: Nothing

:REDD.D# Set Dec/Elevation slew rate to DD.D degrees per second [LX200GPS only]
Returns: Nothing

S – Telescope Set Commands

:SasDD*MM#
Set target object altitude to sDD*MM# or sDD*MM'SS# [LX200GPS]
Returns:
0 Object within slew range
1 Object out of slew range

:Sbn#
Set Baud rate n, where n is an ASCII digit (1..9) with the following interpretation:
1 56.7K
2 38.4K
3 28.8K
4 19.2K
5 14.4K
6 9600
7 4800
8 2400
9 1200
Returns:
1 At the current baud rate and then changes to the new rate for further communication

:SCMM/DD/YY#
Sets the date to MM/DD/YY
Returns: <D><string>
D = '0' if the date is invalid. The string is the null string.
D = '1' for valid dates. The string is "Updating Planetary Data#30 spaces#"
Note: For LX200GPS this is the UTC data!

:SdsDD*MM#
Set target object declination to sDD*MM or sDD*MM:SS depending on the current precision setting
Returns:
1 – Dec Accepted
0 – Dec invalid

:SgDDD*MM#
Set current site's longitude to DDD*MM an ASCII position string
Returns:
0 – Invalid
1 – Valid

:SGsHH.H#
Set the number of hours added to local time to yield UTC
Returns:
0 – Invalid
1 – Valid

:ShDD#
Set the minimum object elevation („higher”) limit to DD#
Returns:
0 – Invalid
1 – Valid

:SLHH:MM:SS#
Set the local time (The parameter should always be in 24 hour format.)
Returns:
0 – Invalid
1 – Valid

:SM<string>#
Set site 1’s name to be <string>. LX200s only accept 3 character strings. Other scopes accept up to 15 characters.
Returns:
0 – Invalid
1 – Valid

:SN<string>#
Set site 2’s name to be <string>. LX200s only accept 3 character strings. Other scopes accept up to 15 characters.
Returns:
0 – Invalid
1 – Valid

:SO<string>#
Set site 3’s name to be <string>. LX200s only accept 3 character strings. Other scopes accept up to 15 characters.
Returns:
0 – Invalid
1 – Valid

:SP<string>#
Set site 4’s name to be <string>. LX200s only accept 3 character strings. Other scopes accept up to 15 characters.
Returns:
0 – Invalid
1 – Valid

:SoDD*#
Set highest elevation („lower”) limit to which the telescope will slew
Returns:
0 – Invalid
1 – Valid

:SrHH:MM.T#
:SrHH:MM:SS#
Set target object RA to HH:MM.T or HH:MM:SS depending on the current precision setting.
Returns:
0 – Invalid
1 – Valid

:StsDD*MM#
Sets the current site latitude to sDD*MM#
Returns:
0 – Invalid
1 – Valid

:SwN#
Set maximum slew rate to N degrees per second. N is the range (2..8)
Returns:
0 – Invalid
1 – Valid

:SzDDD*MM#
Sets the target object Azimuth [LX200GPS only]
Returns:
0 – Invalid
1 – Valid

T – Tracking Commands

:TL# Set Lunar tracking rage
Returns: Nothing

:TQ# Select default tracking rate
Returns: Nothing

:TS# Select Solar tracking rate
Returns: Nothing

U – Precision Toggle

:U# Toggle between low/hi precision mode
Low - RA displays HH:MM.T sDD*MM
High – Dec/Alt/Az displays HH:MM:SS sDD*MM:SS
Returns: Nothing

W – Site Select

:W<n># Set current site to <n>, an ASCII digit in the range 0..3
Returns: Nothing

Appendix C – XML based protocol definition

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="AutonomousObservatory"
    targetNamespace="http://sybillatechnologies.com/AutonomousObservatory.xsd"
    elementFormDefault="qualified"
    xmlns="http://sybillatechnologies.com/AutonomousObservatory.xsd"
    xmlns:tao="http://sybillatechnologies.com/AutonomousObservatory.xsd"
    xmlns:xs="http://www.w3.org/2001/XMLSchema" >

  <xs:simpleType name="Guid">
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="RaDecCoordinates">
    <xs:sequence>
      <xs:element name="targetDeclination" type="xs:double"/>
      <xs:element name="targetRightAscension" type="xs:double"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="AltAzCoordinates">
    <xs:sequence>
      <xs:element name="targetAltitude" type="xs:double"/>
      <xs:element name="targetAzimuth" type="xs:double"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="CompletedCommand">
    <xs:sequence>
      <xs:element name="details" type="xs:string" minOccurs="0" />
      <xs:element name="name" type="xs:string" />
      <xs:element name="referenceId" type="tao:Guid" />
      <xs:element name="returnCode" type="xs:integer" />
      <xs:element name="severity" type="tao:SeverityCode" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="StateTransition">
    <xs:sequence>
      <xs:element name="oldState" type="xs:string" />
      <xs:element name="newState" type="xs:string" />
      <xs:element name="targetState" type="xs:string" />
    </xs:sequence>
  </xs:complexType>

 <xs:simpleType name="SeverityCode">
    <xs:restriction base="xs:string">
      <xs:enumeration value="info" />
      <xs:enumeration value="warning" />
      <xs:enumeration value="error" />
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
```

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="AutonomousObservatoryMount"
targetNamespace="http://sybillatechnologies.com/AutonomousObservatoryMount.xsd"
    elementFormDefault="qualified"
    xmlns="http://sybillatechnologies.com/AutonomousObservatoryMount.xsd"
    xmlns:taom="http://sybillatechnologies.com/AutonomousObservatoryMount.xsd"
    xmlns:tao="http://sybillatechnologies.com/AutonomousObservatory.xsd"
    xmlns:xs="http://www.w3.org/2001/XMLSchema" >

  <xs:import namespace="http://sybillatechnologies.com/AutonomousObservatory.xsd" />
   <xs:element name="command" type="CommandType"/>
    <xs:complexType name="CommandType">
    <xs:sequence>
      <xs:element name="name"  type="xs:string"/>
      <xs:choice id="commandParameters" maxOccurs="unbounded" minOccurs="0">
        <xs:element name="targetAltAz" type="tao:AltAzCoordinates" />
        <xs:element name="targetRADec" type="tao:RaDecCoordinates" />
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="id" type="tao:Guid" use="required"/>
    <xs:attribute name="timestamp" type="xs:dateTime" use="required"/>
  </xs:complexType>

  <xs:element name="event" type="EventType"/>
  <xs:complexType name="EventType">
    <xs:sequence>
      <xs:choice id="parameters" maxOccurs="1" minOccurs="1">
        <xs:element name="completedCommand" type="tao:CompletedCommand" />
        <xs:element name="diagnostics" type="xs:string" />
        <xs:element name="stateChanged" type="tao:StateTransition" />
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="id" type="tao:Guid" use="required"/>
    <xs:attribute name="timestamp" type="xs:dateTime" use="required"/>
  </xs:complexType>

  <xs:element name="mountProperties" type="MountPropertiesType"/>
    <xs:complexType name="MountPropertiesType">
    <xs:all>
      <xs:element name="altitude" type="xs:double"/>
      <xs:element name="azimuth" type="xs:double"/>
      <xs:element name="isConnected" type="xs:boolean"/>
      <xs:element name="declination" type="xs:double"/>
      <xs:element name="epoch" type="xs:string"/>
      <xs:element name="isHomed" type="xs:boolean"/>
      <xs:element name="isParked" type="xs:boolean"/>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="rightAscension" type="xs:double"/>
      <xs:element name="isSlewing" type="xs:boolean"/>
      <xs:element name="targetAltitude" type="xs:double"/>
      <xs:element name="targetAzimuth" type="xs:double"/>
      <xs:element name="targetDeclination" type="xs:double"/>
      <xs:element name="targetRightAscension" type="xs:double"/>
      <xs:element name="isTracking" type="xs:boolean"/>
      <xs:element name="watchdogTimeout" type="xs:duration"/>
      <xs:element name="remainingTimeUntilLimit" type="xs:duration"/>
    </xs:all>
    <xs:attribute name="id" type="tao:Guid" use="required"/>
    <xs:attribute name="timestamp" type="xs:dateTime" use="required"/>
  </xs:complexType>

</xs:schema>
```