PC to Sky Tracker Communication

Version 4.0

Jan 30, 2010

The Sky Tracker system allows the user to slew an alt-azimuth telescope to the selected object and to track on that object. The system consists of the Sky Commander digital setting circle computer with its optical encoders and the Sky Tracker microstepping driver box with stepper motors attached to each axis of the scope. The Sky Commander and Sky Tracker are connected together through a serial communications cable. This allows the tracker to receive information from the Sky Commander such as the distance and direction to the target, or the tracking speed, once the target is reached. The Sky Tracker also has a hand controller box for manual control of slewing and fine positioning. It is also possible to control the scope via a PC.

PC communications with the Sky Commander has been supported for a number of years. The Sky Commander provides positional information to the PC and allows the PC to upload coordinates of special objects. With the introduction of the Sky Tracker, communications protocol changes slightly. This is because the Sky Commander has only one serial port and the Sky Tracker now takes it up. The Sky Tracker however has other serial ports, one of which can be connected to the PC. This provides a satisfactory hardware solution. There is still one problem however. The Sky Commander must be shared between the PC and the Sky Tracker. Each requires different information. The Sky Commander cannot service both at the same time. Therefore the following simple protocol has been worked out which solves the problem:

When the PC needs to communicate with the Sky Commander, it will send 'Q' for bus reQuest. The PC then waits for up to100ms for a bus grAnt, 'A'. In most instances the grant will be given immediately, but if the Sky Tracker is in the middle of receiving information for its own use from the Sky Commander, then the PC will have to wait. Upon receiving the bus grant, the PC will send its request and receive the response from the Sky Commander in the normal way. During this exchange the Sky Tracker is essentially transparent and simply relaying characters as they are received. After the reply is received, the PC must release the bus with 'q'. 'q' is seen by the Sky Tracker and interpreted as the end of the exchange and is not echoed. In this way the Sky Tracker does not have to understand the context of the communication. It can simply keep relaying characters until 'q' is received from the PC. After receiving the bus release 'q', the Sky Tracker can initiate its own request for information from the Sky Commander if it needs to.

Bus requests should not be used when sending commands to the Sky Tracker itself. Any commands or data sent after the bus has been acquired are simply passed to the Sky

Commander. If those commands were not meant for the Sky Commander, they will be lost or possibly misinterpreted.

The PC should not hold the bus any longer than is necessary to complete one task, such as requesting and receiving position information. Keep in mind that the bus is a shared resource and holding it for a prolonged period may cause undesirable operation of the Sky Tracker.

Sky Tracker Commands

The following commands are defined for communication from the PC to the Sky Tracker.

'U' - UTILITY Mode. This command causes the Sky Tracker to enter UTILITY mode. The scope can be slewed in any desired direction (up, down, left or right) at up to 4 degrees/second with ramp up and ramp down speed. Use this command to abort a GOTO operation; motors will ramp down.

'T' - Track Mode.   This command causes TRACK mode to be entered. In TRACK mode the scope will track an object using rate information supplied by the Sky Commander. The '**u**', '**d**', '**l**', '**r**', and '**b**' commands discussed below can be used for positioning. In TRACK mode, the arrow buttons add or subtract (depending on direction) to the track rate – 8 arc minutes per second. This is usually adequate for search and coarse positioning.

'D' - **GUIDE Mode.** This command is used to obtain finer positioning and possibly photographic guiding. It is the same as TRACK mode but the correction rate is 2 arc minutes per second (4 times slower). This is the default rate, but can be changed using the 'K' command described below. Note: use the 'T' command to revert back to normal TRACK mode speeds.

'B' - Standby Mode. This command causes the Sky Tracker to enter STANDBY mode. DC drive to the stepper motors is cut, thus saving battery power.

'G' - Go To.   This command causes the Sky Tracker to move the scope to the selected object. After target is reached, TRACK mode is invoked.

'u' - **Press Up** – this command causes the UP directional button to be pressed, e.g. to make a guiding correction. If a 'd' command was sent previously, it is cancelled.
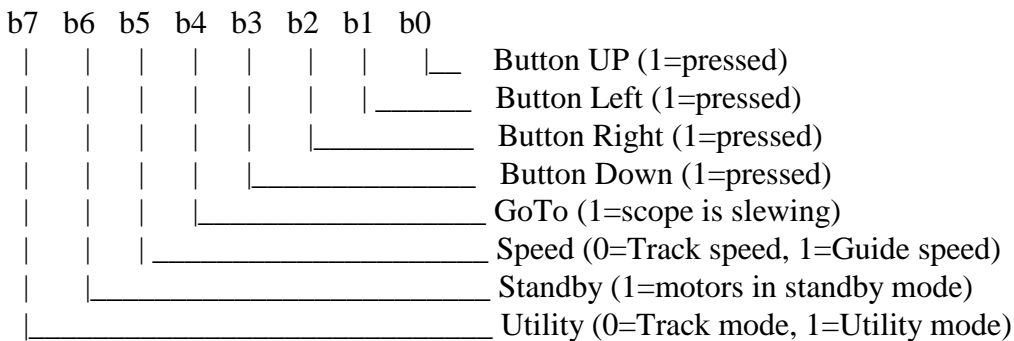
'd' - **Press Down** – this command causes the DOWN directional button to be pressed, e.g. to make a guiding correction. If a 'u' command was sent previously, it is cancelled.

'l' - **Press Left** – this command causes the LEFT directional button to be pressed, e.g. to make a guiding correction. If an 'r' command was sent previously, it is cancelled.

'r' - **Press Right** – this command causes the RIGHT directional button to be pressed, e.g. to make a guiding correction. If an 'l' command was sent previously, it is cancelled.

'b' - **Button Release** – this command causes all directional buttons to be released, e.g. end of guiding correction.

'R' – Read Buttons – This command is used to read the buttons and status of the Tracker. The Tracker will respond with one byte encoded as follows.

```
b7  b6  b5  b4  b3  b2  b1  b0
 |   |   |   |   |   |   |   |__  Button UP (1=pressed)
 |   |   |   |   |   |   |_____  Button Left (1=pressed)
 |   |   |   |   |   |_____  Button Right (1=pressed)
 |   |   |   |   |_____  Button Down (1=pressed)
 |   |   |   |_____  GoTo (1=scope is slewing)
 |   |   |_____  Speed (0=Track speed, 1=Guide speed)
 |   |_____  Standby (1=motors in standby mode)
 |_____  Utility (0=Track mode, 1=Utility mode)
```

'S' – Set Calibration Factor – This command is used to calibrate the motor speed according to the scope size. It is followed by eight bytes representing two floating point numbers. The numbers correspond to the diameters (in inches) of the azimuth ground board and the elevation bearing or trunnion.

For Azimuth (first four bytes) the number is 332500/ (ground board diameter in inches).
For Elevation (second four bytes) the number is 320300/ (el bearing diameter in inches).

The numbers are programmed into non-volatile EEPROM. When programming is complete, the Tracker will respond with 'A' for acknowledge. Note: This command is intended for fine tuning of tracking speed. If it is used, it must be the first command sent following turn on, and it will work only once following each turn on sequence. This is to minimize the possibility of accidently changing this critical parameter.

's' – Read Calibration Factor – This command is used to read the motor speed calibration factor. The Tracker will respond with eight bytes representing two floating numbers. The numbers correspond to the diameters (in inches) of the azimuth ground board and the elevation bearing or trunnion. This command can be used to verify calibration or to identify scope size.

For Azimuth (first four bytes) the number is 332500/ (ground board diameter in inches). For Elevation (second four bytes) the number is 320300/ (el bearing diameter in inches).

'V' – Read Version – This command is used to read the EPROM software version. The Tracker will respond with an eight character ASCII string. Example: 'v4.0____'. Note the underscore represents an ACSII space character (0x20). The string is always eight characters.

'K' – Set Guide Speed – This command is used to set the guide correction speed for azimuth and elevation independently. Two bytes follow the command byte, one for azimuth and one for elevation. The bytes are treated as unsigned values (0 to 255) with the first for azimuth and the second for elevation. Each count is equal to 1 arc second per second, e.g. a value of 10 would equal 10 arc second per second correction rate. Note – these parameters are stored in volatile memory, they are not retained after turn off. At power on, they are both initialized to 120, or 2 arc minutes per second.

Note: Hand controller buttons also use the guide rate set with the 'K' command.

'k' – Read Guide Speed – This command is used to read the guide correction speed for azimuth and elevation. The Tracker will respond with two unsigned bytes, the first for azimuth and the second for elevation. Each count is equal to 1 arc second per second.

The following table summarizes the effect of each remote command and the resulting change to 'mode flags' read using the 'R' command.

Command		B7	B6	B5	B4	Description
		(B_UTIL)	(B_CUT)	(B_SPEED)	(B_GOTO)	
U	Utility	1	0	0	0	**Utility Mode** - Scope slews fast with arrow buttons
T	Track	0	0	0	0	**Track Mode** - Scope tracks object and arrow buttons used for coarse position, 8 arc min/sec
D	Guide	0	0	1	0	**Track Mode (fine position)** - Scope tracks object, arrow buttons used for fine positioning - 1 to 255 arc sec/sec
B	Standby	0	1	0	0	**Standby Mode** - Motor drive is cut to save battery.
G	Go To	0	0	0	1	**GOTO Mode** - Scope slews to target object in Sky Commander.

Table 1 - Summary of Remote Commands and Resulting Bit Settings

Notes on Remote Commands

- While the scope is in go-to mode (slewing to target object), any of the other commands, U,T,D,B will cause a go-to abort. The motors will ramp down, stop and the tracker will revert to Utility mode. Go-to status is easily obtained using the 'R' command.

- Other than the restriction above, a single character command can be used to transition from any mode to any other mode.

- While the scope is GoTo mode, if the PC should request bus access to Sky Commander ('Q' command), a null character (all zero bits) is returned indicating that bus access is not allowed. The reason is that time sensitive messaging between the Tracker and Sky Commander is taking place and must not be interrupted.

- Arrow button commands can be combined, e.g. 'l'+'d' or 'r'+'u' to achieve movement on both axes, but of course 'u'+'d' or 'l'+'r' don't make any sense.

Floating Point Format

Numbers sent and received with the 'S' and 's' commands are expressed in IEEE floating point format as described below:

Each number consists of four bytes (32 bits) with the most significant byte sent first. The bits are assigned the following meaning:

    31:   sign of mantissa, 1=negative.
  30-23: exponent, biased by 128.
   22-0: mantissa, 24 bits, MSB hidden and always=1.

The mantissa is normalized, 24 bits with the MSB hidden. The MSB is always assumed to be 1, therefore only 23 bits are needed to store the mantissa. The exponent is biased by 128; i.e. a 128 value represents an exponent of zero, a 129 value an exponent of 1, etc. Example: if the exponent has a 131 value (+3), the number is 2^3 * (mantissa). The mantissa is always less than one, and greater than or equal to 1/2.

The number zero is stored as four zero bytes.

New Sky Commander Commands

The following commands have been added or revised to the existing set.

'N' - Realign to current object

'L99' -   Load Temporary Object Position. The 'L00' through 'L58' commands are used to store object coordinates in Sky Commander EEPROM memory. L99 has a special meaning. The coordinates that follow will be put into RAM memory and become current for the purpose of going to the object or realigning to it.

Operating Examples

To make the PC link useful, the user needs to send object coordinates to the Sky Commander and make them 'current' in memory so that scope can be positioned there. The following is the sequence of commands:

PC:	Q	bus request (to S.T.)
ST:	A	bus grant (from S.T.)
PC:	L99 12.345 +67.890<cr>*	upload of object coordinates (to S.C.)
ST:	A	acknowledge (same as grant, from S.C.)
PC:	q	bus release (to S.T.)
PC:	G	command to go to the object (to S.T.)

*<cr> indicated carriage return character (hexadecimal 0D).

The L value of 99 in the above sequence is specially defined and tells the Sky Commander to put the coordinates into RAM and make them current. L values of 00 to 58 (as previously defined) are used to store object coordinates into EEPROM memory and do not cause the object to become current.

Upon reaching the target coordinates, the Sky Tracker will revert to TRACK mode automatically. Note: The scope does not have to be on any object in order to track. It will track properly at any position it happens to be at. The Sky Commander provides rate information based on the scope's position only.

A GOTO can be aborted. For instance if the user issues a G command the scope starts to move to the current object. If the user changes his mind while the scope is moving, it may be stopped by sending the U (UTILITY) command. This will ramp down the speed, stop the scope and put it in the UTILITY mode. Tracking can be resumed by sending the T command for instance, or another G command could be sent to start the GOTO again.

Realigning to an object's coordinates

Often it is desirable to realign the Sky Commander on a known star or object. This can now be done from the PC using the command sequence below.

PC:	Q	bus request (to S.T.)
ST:	A	bus grant (form S.T.)
PC:	L99 12.345 +67.890<cr>	upload of desired object coordinates (to S.C.)
ST:	A	acknowledge (same as grant, sent from S.C.)
PC:	N*	Realign Command (to S.C.)
PC:	q	bus release (to S.T.)

*The realign command, 'N' is directed to the Sky Commander. Therefore it must be sent before releasing the bus.

Special Requirements

At the time of this writing it appears that at a very minimum, two stop bits will be required when sending more than one character at a time; as in sending coordinates for instance. This is due to the fact that the serial port to the PC is not buffered. Characters, which are not read by the Sky Tracker microprocessor before the start of the next character, will be lost or corrupted. Only 9600 b/s can be used.