**ROTSE III Command List**

November 28, 2000
J. Alan Schier
(818)790−7481, schier@compuserve.com

Revised April 12, 2001

Generalities

The telescope control is a polled system; that is, it does not generate any unsolicited messages. To monitor the functioning of the system, the host must request the status.

The control system issues a reply each time it receives a message with a valid cyclic−redundancy checksum ("CRC", see Command Format below). If the CRC is incorrect, usually indicating an error in transmission, the message is ignored. It is possible to turn off the CRC checking in the control system. This is useful when issuing commands through a keyboard. Any four placeholder characters may then take the place of the CRC.

CRC checking is turned on and off in the file "servo.ini". To do so, set the parameter "m_bIgnoreCRC" to 1 in this file.

When valid operating parameters are sent to the telescope control, they are immediately implemented as well as saved to disk. This allows them to be used at the next startup.

At startup, the system is in the "halt" condition. That is, the brakes are on and the servos are disabled.

Command Format

$[Command][optional data][CRC16][CR]

$: Indicates the start of a message
[Command]: Command mnemonic. Not case sensitive.
[optional data]: Data required by the command. ASCII hex. Not case sensitive.
[CRC16]: Four character hex (HHHH) value of the CRC computed with the characters preceding the CRC. See "C Programmers Guide to Serial Communication", Joe Campbell.
[CR]: Carriage return (0x0D). Indicates the end of a message.

−−Commas are used to separate multiple parts of the optional data (NNNNN, NNNNN, NNN)
−−Spaces may be inserted in between the parts of the command string if desired. If inserted, they must be included in the CRC calculation.

Response Format

@[Command][optional data][CRC16][CR]

@: Indicates the start of a response.
[Command]: Repeats the received command.
[optional data]: Either:
−−Repeated data from the command (e.g., the value of the programmed acceleration)
or
−−Returned data (e.g. the status words)
[CRC16]: Four digit hex (HH) value of the CRC computed with the characters preceding the CRC.
[CR]: Carriage return (0x0D). Indicates the end of the response.

If the received CRC value is incorrect, the system will not respond.

If the received CRC value is correct, but the message is otherwise unrecognized or data is incorrect, the response will be:
@[Command]?[optional message][CRC16][CR].

[optional message]: ASCII text indicating the source of the problem, if available.

Command List

In the following:

NNNNNN indicates ASCII decimal digits (1,2,3, etc.). Where it makes sense, a digit may be positive or negative and may contain a decimal point.

HHHH indicates ASCII hex digits (0–9, A–F). These may not be negative and may not contain a decimal point.

$StopRA
$StopDec — Stop the axis with a controlled deceleration as quickly as possible and hold position. Servos are still enabled.

$HaltRA
$HaltDec — Halt the axis (Brake on, amplifiers disabled, position command set to actual position.) The first command after a Halt must be a Stop command. Other commands will generate an unrecognized message response.

$AccelRa NNNNNN
$AccelDec NNNNNN — Set the axis acceleration to NNNNNN counts/sec^2

$VelRa NNNNNN
$VelDec NNNNNN — Set the axis velocity to NNNNNN counts/sec

$PosRA NNNNNN
$PosDec NNNNNN — Set the axis target position to NNNNNN (will run at max velocity to NNNNNN)

$MaxVelRA NNNNNN
$MaxVelDec NNNNNN — Set the maximum axis velocity to NNNNNN counts/sec. If an axis exceeds this maximum velocity, it is detected as a fault and the control for that axis is set to the "Halt" state.

$HomeRA
$HomeDec — Move the axis to the index position and hold there.

$Status1RA
$Status1Dec — Report the axis status words.
Response: @Status1Dec [Command Pos], [Actual Pos][CRC][CR]
The values are reported as decimal encoder counts.

$Status2RA
$Status2Dec — Report the axis status word.
Response: @Status2Dec [Status Word]CRC][CR]
The status is reported as a 16 bit hex word (HHHH) with each bit indicating a status value.
The word is arranged as $b_{15}\ldots b_0$, and the meanings are:
$b_0$: Brake is engaged.
$b_1$: Drive amplifier is disabled.
$b_2$: In emergency–stop limit.
$b_3$: In negative travel limit.
$b_4$: In positive travel limit.
Other bits are not used.

$Status3RA
$Status3Dec — Report the axis status words.
Response: @Status3Dec [amplifier drive signal], [integrator value][CRC][CR]
The are reported as decimal values.

$Params1RA [P gain], [I gain], [D gain]
$Params1Dec [P gain], [I gain], [D gain]

Set the axis operating parameters:
amp drive signal = [P gain]*((Integral of position error)*[I gain] + (Derivative of pos. error) * [D gain])

$Params2RA [I limit], [Drive limit]

$Params2Dec [I limit], [Drive limit]

Set the limits for the axis servo operation:
[I limit]: Limiting magnitude of the (I gain)*(Integral of error).
[Drive limit]: Limiting magnitude of amplifier drive signal.

$RecentFaults

Returns a string with the most recent faults as [fault, time, and date]; [fault, time, and date]; …[CRC][CR]. Up to nine faults are returned. The faults may be:

Axis 1 Positive Limit
Axis 1 Negative Limit
E−Stop or Overtravel(1)
Axis 1 Encoder Failure
Axis 1 High Output
Axis 1 Amplifier Fault
Axis 1 Overspeed
Axis 2 Positive Limit
Axis 2 Negative Limit
E−Stop or Overtravel(2)
Axis 2 Encoder Failure
Axis 2 High Output
Axis 2 Amplifier Fault
Axis 2 Overspeed

Axis 1 is RA and Axis 2 is dec. With the exception of the positive and negative limits, all the these faults will cause an axis to enter the Halt condition. If there is an e−stop fault, the axis must be manually moved off of the e−stop limit switch.