SAGEBRUSH Technology, Inc.

Command Set Documentation

Pan-Tilt Gimbals

Servomotor Version

Protocol Revision: 1.9

Document Revision Status: A
Revision Date: June 1, 2004

5911 Jefferson Street NE Albuquerque, NM 87109 505-299-6623 Fax: 505-298-2072

www.sagebrushtech.com

CONTENTS

1. COMMAND INPUT DEFINITION 1

1.1. OVERVIEW 1
1.2. WHAT'S NEW IN SOFTWARE REV 1.9 1
1.3. CONFIGURATION BLOCK DIAGRAMS 3
1.4. STANDARD MESSAGE FORMATS 6
1.5. PROTOCOL DEFINITION 10
1.6. PAN/TILT MOUNT COMMANDS 16
1.7. CAMERA COMMANDS 20

FIGURES AND TABLES

FIGURE 2. MCU/GCU WITH FOUR PORTS AND ONE DEVICE PER PORT. 3
FIGURE 3. MCU/GCU WITH ONE PORT AND MANY DEVICES. 3
FIGURE 4. MCU/GCU WITH MANY PORTS AND MANY DEVICES PER PORT. 4
FIGURE 5. BLOCK DIAGRAM OF EXPANDED SYSTEM. 5
TABLE 1. COMMAND MESSAGE FORMAT 6
TABLE 2. SPECIFIC GROUP, PORT, AND DEVICE ADDRESSES 7
FIGURE 6. MESSAGE FLOW FOR SIMPLE AND RESULT RESPONSES TO MESSAGES. 8
TABLE 3. EXAMPLE OF HEADER ADDRESS BYTES FOR COMMAND AND RESPONSE MESSAGES 9
FIGURE 7. BLOCK DIAGRAM OF GCU TO GCU COMMUNICATION (SEE TABLE 3.). 9
TABLE 4. GENERAL DEVICE CONTROL COMMANDS 11
TABLE 4A. AUXILIARY RELAY STATUS RESULTS MESSAGE 12
TABLE 5. AWAKE STATUS RESPONSE FORMAT 12
TABLE 6. DEVICE IDENTIFICATION RESPONSE MESSAGE 13
TABLE 6A. ERROR CODE RESPONSE MESSAGE 13
TABLE 6B. STATUS BIT RESPONSE MESSAGE 13
TABLE 7. RATE GAIN SET MESSAGES 14
TABLE 8. RATE GAIN QUERY MESSAGE 14
TABLE 9. INITIALIZATION RESULT MESSAGE FORMAT 14
TABLE 10. SOFTWARE LIMITS SET COMMAND FORMAT, QUERY COMMAND & RESPONSE MESSAGE FORMAT 16
TABLE 11. PAN/TILT MOUNT COMMANDS 18
TABLE 12. ACCELERATION AND RATE QUERY AND SET COMMAND FORMATS 19
TABLE 13. ACCELERATION AND RATE QUERY RESPONSE FORMAT 19
TABLE 14. VERIFY HOME RESPONSE MESSAGE FORMAT 19
TABLE 15. PAN - TILT MOUNT GO TO COMMAND AND RESPONSE MESSAGE FORMAT 20
TABLE 16. CAMERA CONTROL COMMANDS 21
TABLE 17. LENS POSITION SET COMMAND AND RESULT MESSAGE FORMAT 21
TABLE 18. SERIAL PORT CONFIGURATION MESSAGE FORMAT 22

1. COMMAND INPUT DEFINITION

1.1. Overview

This document defines a standard for the control of the Sagebrush Technology Pan-Tilt Gimbals using servomotors. The purpose of this standard is to facilitate the integration and interoperability of these gimbals. To achieve that goal, this document defines the message packets and responses that are used for the command and control of the features of the pan-tilt gimbal.

The main features of this standard are the definition of the command messages and responses between gimbals in a group of gimbals that are interconnected by some media (network), and the message header that specifies the addressing of the command messages to assure appropriate delivery. Other features of this standard include methods to perform error checking and retransmission attempts.

Note: References to "device" or "devices" in the remainder of this section may be interpreted as "gimbal" and "gimbals" respectively.

1.2. What's new in Software Rev 1.9

New Commands:

HP	Sends a goto command for the last saved Gimbal position [saved by HA].
VP	Sends a goto command for the last saved Lens position [saved by HA].
VS	Returns Camera and Relay Status.
F?	Returns current speed number.
Z?	Returns current speed number.
C?	Returns lens software limits or preset position values.
P?	Returns Gimbal position or preset position values.
I?	Returns status bits regarding limits, initialization and error conditions.
E?	Returns error bits regarding specific error conditions.
*dA	Returns analog channel values.
*di	Returns digital input values.
*do	Returns digital output values.
*dl	Returns status messages.
*dm	Set Loop Mode.
*mr	Send Rate commands.
*me	Send/Read Position in encoder counts.
*mp	Send/Read Position in floating point.

*mu Set/Read user parameters [integer].

*sa Set/Read Inertial Gain in floating point.

*sgr Set/Read Rate Gain in floating point.

*sgp Set/Read Position Gain in floating point.

*sl Set/Read Rate/Acceleration Limits in floating point.

*x0c Configure Auxiliary Serial Port 0

*x0t Transmit to Auxiliary Serial Port 0

*x0r Receive data from Auxiliary Serial Port 0

*x0n Get Number of Characters in Aux. Serial Port 0 Buffer

*x0f Flush Buffer for Auxiliary Serial Port 0

*x1c Configure Auxiliary Serial Port 1

*x1t Transmit to Auxiliary Serial Port 1

*x1r Receive data from Auxiliary Serial Port 1

*x1n Get Number of Characters in Aux. Serial Port 1 Buffer

*x1f Flush Buffer for Auxiliary Serial Port 1

Please contact the factory regarding which commands are implemented on your system, and their correct use.

Fixed in 1.9:

1. Slew commands no longer need to be sent twice to change from position mode to rate mode. This bug was introduced in Rev 1.7
2. Lens specific firmware no longer required for Integrated Lens Control.

1.3. Configuration Block Diagrams

Figure 2 shows an example of an MCU or GCU with four ports and a single device attached to each port. Figure 3 shows an example of an MCU or GCU configured with many devices attached to a single port. Figure 4 shows an example of an MCU or GCU with multiple ports and devices.

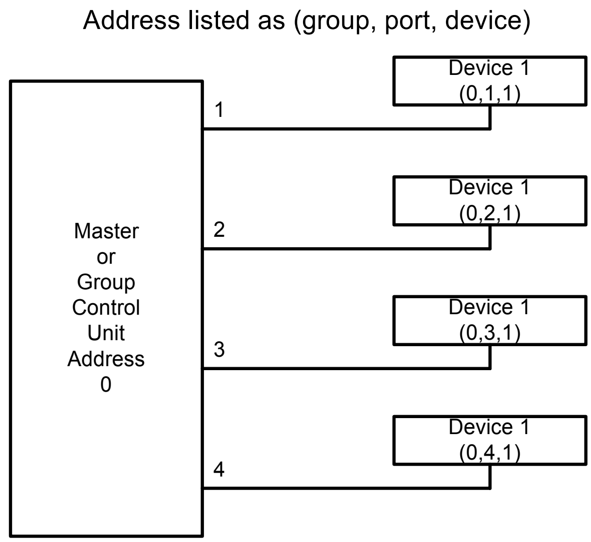

Figure 2. MCU/GCU with four ports and one device per port.

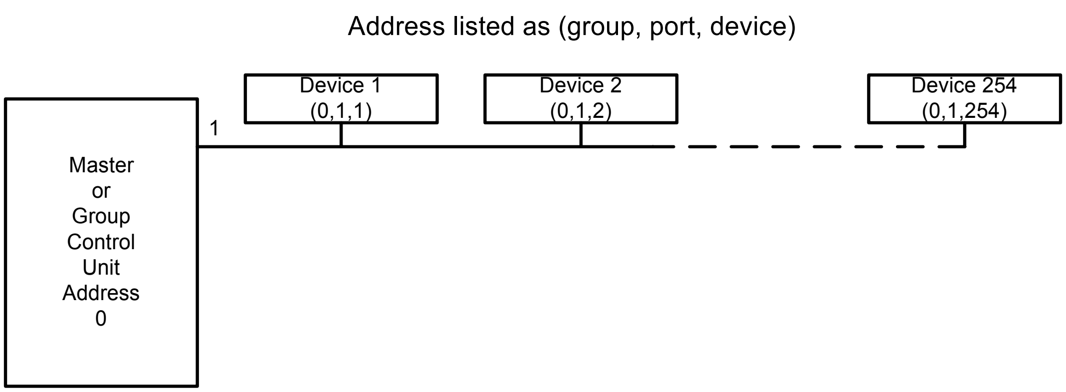

Figure 3. MCU/GCU with one port and many devices.

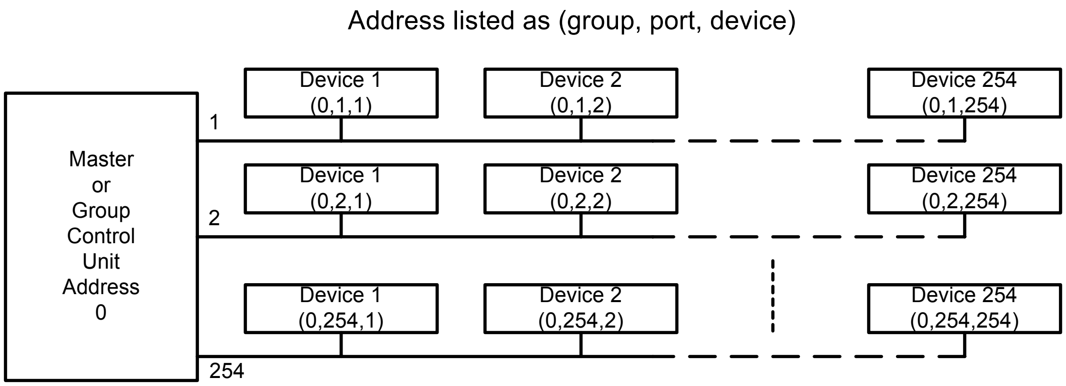

Figure 4. MCU/GCU with many ports and many devices per port.

Many GCUs and devices can be connected to form a network using the protocol defined in this document. Figure 5 shows a block diagram for a fully expanded system. In this configuration, up to 254 devices can be connected to a single communications port on a GCU. Each GCU can have up to 254 ports. Up to 254 group control units can be connected together in a network. One MCU is allowed in the system for sending messages to GCUs or directly to the devices on its own ports to be controlled.

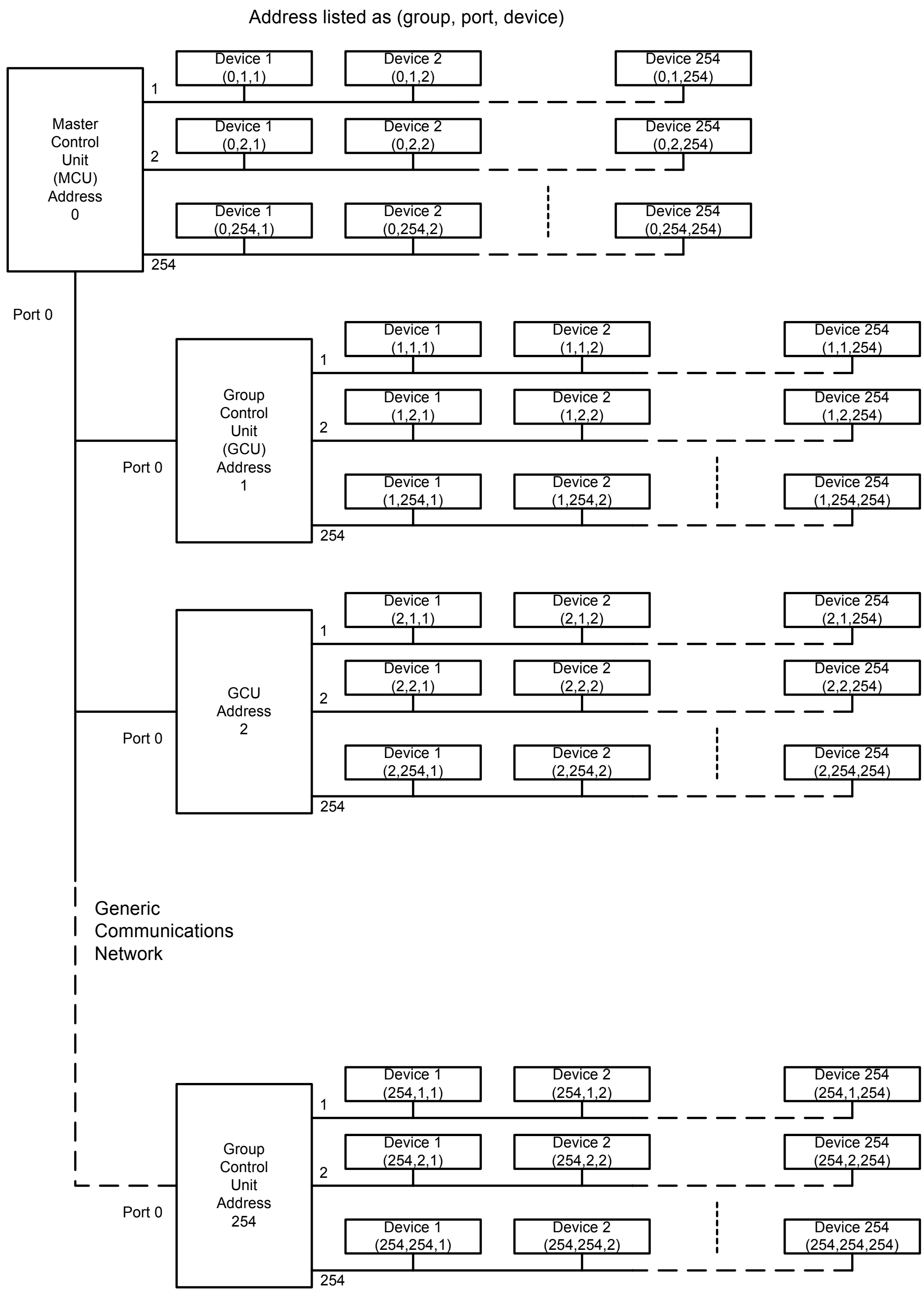

Figure 5. Block diagram of expanded system.

1.4. Standard Message Formats

1.4.1. Command Message From Control Unit To Remote Device

The standard, fixed command message format is six bytes of header plus one to 256 bytes of payload or command message plus a one byte checksum. See Table 1.

Table 1. Command Message Format

BYTE	DATA	DESCRIPTION
1	0xF8	Start character
2	Destination Group Address	Address of control unit destination address
3	Destination Port Address	Address of port within a control unit
4	Destination Device Address	Address of device on a port
5	Source GCU Address	Group Control Unit address of source of message
6	Length	Length in bytes of command data. Does not include bytes 1 through 6 or the checksum byte.
7 to 7 + length-1	Command data	See tables in following sections.
7 + length	Checksum	Computed on byte 2 through byte 7 + length -1

1.4.1.1. Header. The header consists of the first six bytes of the command message. All bytes in the header have hexadecimal values.

The first byte of a message always has the value 248 (hexadecimal 0xF8). This is the start character.

The second byte of a message indicates the destination Group Control Unit (GCU) address of the specific device for which the message is intended. The reserved addresses are shown in Table 2.

There is a limit of 254 GCUs per system with addresses from 1 to 254 (0x01 to 0xFE). GCU address zero (0) is reserved for the Master Control Unit (MCU). GCU address 255 is a wild-card address for messages that can be broadcast to all GCUs in a complete system.

The third byte of the address field is the destination communications port identifier within the GCU.

There is a limit of 254 ports per GCU with addresses from one to 254. Port address zero (0) is used to identify communications between GCUs and an MCU on a separate communications link. Port address 255 is a wild-card address for messages that can be broadcast to all ports in a GCU.

The fourth byte of the address field is the destination address of the device connected through the communications port to the GCU.

There is a limit of 254 devices per port, and device numbers can range from one to 254. Device address 0 is used for identifying the MCU/GCU as the destination in a result response message from a remote device. Device address 255 is a wild-card address for messages that can be broadcast to all ports in a GCU.

The fifth byte is the GCU address of the source of the message. This allows messaging from the MCU to a GCU, and between GCUs.

The sixth byte is be the length of the payload or command string, in bytes, not including the addressing bytes and checksum byte. This allows the receiver to accurately check message length. For example, if the payload or command message is two bytes, then the length byte will have a value of 0x02. If the length of the payload is 256 bytes, the length byte will have a value of zero.

All tables below that describe command formats only define the payload. The header and checksum, as defined in Table 1., must always be added.

Table 2. Specific Group, Port, and Device Addresses

ADDRESS	GCU ID BYTE	PORT ID BYTE	DEVICE ID BYTE
0	Reserved for MCU address	Port ID for MCU/GCU inter-communications	Identifies MCU/GCU as destination device
1	GCU 1	Port 1	Device number 1
2	GCU2	Port 2	Device number 2
…	…	…	…
254	Last GCU on System	Last port on GCU	Last device on port
255	Wild Card address for all GCUs	Wild Card address for all ports on a GCU	Wild Card address for all devices on a port

1.4.1.2. Payload. The payload of the message consists of at least one byte, up to a maximum of 256 bytes of data. All information is to be represented by hexadecimal ASCII characters. Numbers above 9 are to be represented by capital letters (A, B,…F). A hexadecimal number like 0x3AF is represented as 0x33, 0x41, 0x46 (not as 0x33, 0x61, 0x66). The command data is sent as the ASCII codes for the characters shown in the tables. Commands generally consist of 2-byte pairs (2 ASCII characters), but may be more.

1.4.1.3. Checksum. A checksum is transmitted with every command message. The checksum byte is the last byte of the command message. The checksum is calculated as the straight sum of bytes, ignoring overflow. The checksum is computed on byte 2 through byte 7 + length-1.

1.4.2. Simple Response Messages From Remote Devices

Every command message is responded to with a single ACK character (0x06, or 6_{10}) or a single NAK character (0x15 or 21_{10}) at the transmitted bit rate to acknowledge receiving the message and to indicate if the message was received properly. An example of this simple handshake is shown in the first example in Figure 6. Note that these simple responses **do not** have a header and checksum.

If the calculated checksum is the same as the transmitted checksum, a single ACK character is sent back from the device to the GCU.

If the calculated checksum is not the same as the transmitted checksum, a NAK character is sent back from the device to the GCU.

After an ACK response is sent, if the message contains an invalid, unrecognized, or non-applicable command, the receiving device will also respond with the "Non-Implemented Command (NIC) character, which is a single question-mark character ("?", 0x3F, or 63_{10}), to alert the control unit that the message sent has no meaning to the receiver.

If no ACK or NAK is received within a "time-out" period, the message should be re-transmitted.

1.4.3. Result Response Messages From Remote Devices

If the command requires a result response, one response will be sent. This is shown in the example in Figure 6.

The result response is sent following the ACK. There may be a time delay between the ACK and the result response message, depending on the command.

The response message generally has ASCII characters to specify status and parameter settings back to the MCU/GCU.

The format of the result response message uses the same header, payload, and checksum as the command format in Table 1.

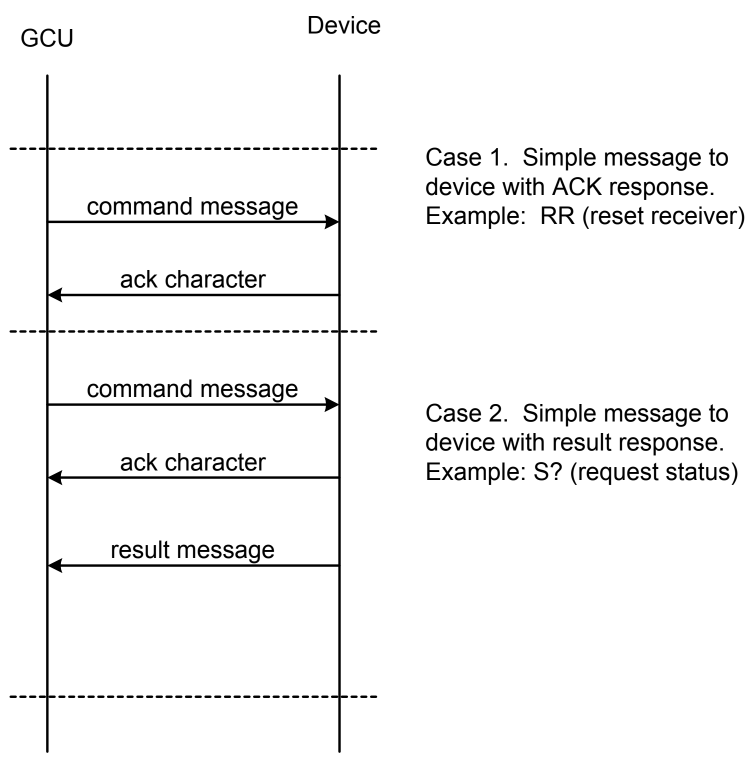

Figure 6. Message flow for simple and result responses to messages.

In the response message, the Destination Group Address and the Source Group Address are exchanged (assuming they are different, that is, that the source of the command was a GCU different from the one that the device is directly connected to). Because of the local communications between the GCU and the device, the GCU implicitly recognizes this message as one being transmitted back to the GCU, therefore the Destination Port Address and the Destination Device Address in the response message are unchanged from the original command message. The GCU can thus use the Destination Group information to determine if this message needs to be retransmitted to another GCU or keep the results local. Table 3. is provided as an example of this exchange.

Table 3. Example of Header Address Bytes for Command and Response Messages

HEADER BYTE	TRANSMIT COMMAND	RESULT RESPONSE
Destination Group	5	2
Destination Port	1	1
Destination Device	12	12
Source Group	2	5

In the diagram shown in Figure 7, a command issued from GCU-2 is passed through GCU-5 to Device 12 on Port 1. The header bytes are reversed according to Table 3. for the result response from the device back to GCU-2.

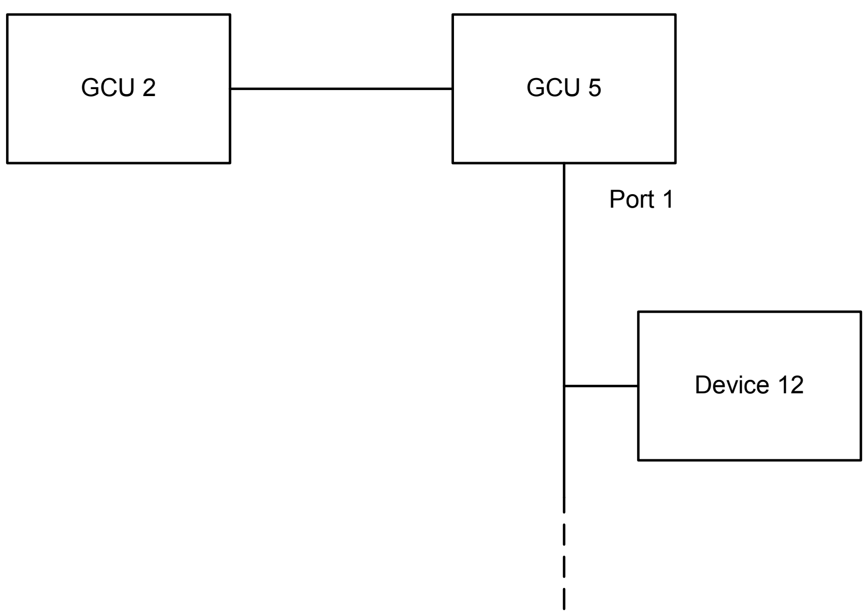

Figure 7. Block diagram of GCU to GCU communication (see Table 3.).

1.5. Protocol definition

1.5.1. Electrical Standards, Signal Formats, and Data Rates

An RS-232 serial data interface is used. The receiving device (Gimbal) supports data rates in bits per second (bps) of one or more of the following: 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200 bps.

The default data rate is 9600 bits per second (bps) with one start bit, eight data bits, one stop bit, and no parity. There are no provisions defined to change start, stop, data, or parity bits.

The control unit may query the receivers to determine maximum data rate for all devices on the link.

The control unit may send commands to receivers to change data communication rates.

1.5.2. General Functions of the Protocol

The general functions included in the protocol consist of control functions for a pan-tilt mount (PTM).

1.5.3. Control Receiver and General Device Control Commands

General device commands are defined in Table 4.

A control unit can program a device's ID number. This can be accomplished by connecting only one device to the port and issuing the three byte command "#nn" where nn is the device ID represented by two ASCII characters with a range of 01 to FE.

All devices must respond to a message requesting the status of the device. This is known as a request awake status command.

All devices must respond to a maximum data rate query. The response message specifies the highest data rate at which the device can communicate.

The control unit can specify to the devices the communications rate within the range of possible data rates the devices support.

All devices must respond to an ID request message. The response message identifies the unit by including a block of information. The format for the payload of the result response is described in Table 6.

The control unit can set or query the rate gain settings in the embedded software.

The control unit can direct the device to perform an initialization that will reorient the software with the center and limit switches.

The control unit can set, query and clear/reset the software (stop) limits used by the embedded gimbal control software.

Table 4. General Device Control Commands

COMMAND	DESCRIPTION	RESPONSE	COMMENTS
#*NN*	Set device Address/ID	ACK	NN = new device address/ID (two ASCII characters, range 01 to FE)
AW	Request Awake Status	ACK + Response	Response is one byte – see Table 5.
B?	Request maximum data rate	ACK + Result	Response is C0…CF. See below
C0 C1 C2 C3 C4 C5 C6 C7 C8…CF	Set communication data rate	ACK	0 = 1200 bits per second 1 = 2400 bits per second 2 = 4800 bits per second 3 = 9600 bits per second 4 = 19,200 bits per second 5 = 38,400 bits per second 6 = 57,600 bits per second 7 = 115,200 bits per second Reserved for future data rates
D?	Request ID information	ACK + Result	See Table 6. for response format
E?	Request error codes	ACK + Result	See Table 6. for response format
I?	Request status bits	ACK + Result	See Table 6. for response format
GRP*nn*	Set Pan Rate Gain	ACK	See Table 7.
GRT*nn*	Set Tilt Rate Gain	ACK	See Table 7.
G?	Rate Gains Request	ACK + Response	See Table 8. for response format
IX	Initialization	ACK + Result	See Table 9. for result format
L0	Open Relay 0	ACK	Reserved
L1	Open Relay 1	ACK	Reserved
L2	Open Relay 2	ACK	
L3	Open Relay 3	ACK	
L8	Close Relay 0	ACK	Reserved
L9	Close Relay 1	ACK	Reserved
LA	Close Relay 2	ACK	
LB	Close Relay 3	ACK	
LR	Auxiliary Relay Status Request	ACK + Result	See Table 4A for result format
LC	Clear Software Limits	ACK	
LS	Set Software Limits	ACK	See Table 10. $\mathbf{LS}p_{cc2}p_{cc1}p_{cc0}\ p_{c2}p_{c1}p_{c0}t_{u2}t_{u1}t_{u0}\ t_{d2}t_{d1}t_{d0}$
L?	Software Limits Request	ACK + Result	See Table 10. for result format

Table 4A. Auxiliary Relay Status Results Message

BYTE	DATA	DESCRIPTION
7-8	L0…LF	Results indicate which relays are on (1) or off (0). For example: 0 = all relays off 1 = relay 0 on 2 = relay 1 on 3 = relays 0 and 1 on F = all relays on

1.5.4. Changing Device Addresses

The device ID may be programmable over the data link. The following process may be used: A single device can be connected to a control unit and the "#NN" message transmitted using wild-card addressing to set the device ID to a new address, where "#" is character 0x23 and NN is the new device ID number.

The group ID is not changeable over the data link.

1.5.5. Awake Status Response Message

Table 5. defines the message returned from a Status request.

Table 5. Awake Status Response Format

BYTE	DATA	DESCRIPTION
7	Status Code	• If Status Code=0, hardware is functioning properly. • If bit 0 is set, the azimuth axis is not initialized. • If bit 1 is set, the elevation axis is not initialized.. • If bit 2 is set, motors are turned off. • If bit 3 is set, an error condition exists. • Bits 4..7 reserved.

1.5.6. Maximum Data Rate Response Message

In Table 4., the C0…CF messages define the set of returned values from a Maximum Data Rate request.

1.5.7. Identification Response Message

Table 6. defines the message returned from a Request ID request.

Table 6. Device Identification Response Message

BYTE	DATA	DESCRIPTION
7-8	ID	ID Response
9-10	Protocol Revision Level	Two ASCII characters (leading space character)
11..30	ASCII character string	Device name, padded with 0x20 (space) if necessary
31-50	ASCII character string	Serial number of device, padded with 0x20 (space) if necessary

Table 6A. Error Code Response Message

BYTE	DATA	DESCRIPTION
7-8	Azimuth Error Bits	Two ASCII-hex characters.
9-10	Elevation Error Bits	Two ASCII-hex characters.
		ASCII-hex Error Codes: bit 0 - center reference. bit 1..3 - reserved. bit 4 - initialization. bit 5 - motor time-out. bit 6 - temperature. bit 7 - reserved.

Table 6B. Status Bit Response Message

BYTE	DATA	DESCRIPTION
7-8	Azimuth Status Bits	Four ASCII-hex characters.
9-10	Elevation Status Bits	Four ASCII-hex characters.
		ASCII-hex Status Codes: LSB bit 0 - not initialized. bit 1 - positive limit found. bit 2 - negative limit found. bit 3 - positive center found. bit 4 - negative center found. bit 5 - positive hard stop. bit 6 - negative hard stop. bit 7 - error code present. MSB bit 0..7 - reserved.

1.5.8. Rate Gain Set and Query Messages

Table 7. defines the commands to set the pan and tilt rate gains. The format of the information returned by the gimbal in response to a request rate gains command is defined in Table 8.

Table 7. Rate Gain Set Messages

BYTE	DATA	DESCRIPTION
7-9	GRP or GRT	GRP=Set Pan rate gain. GRT=Set Tilt rate gain.
10-11	rate gain value	Pan/Tilt Rate Gain Value: 2 bytes representing a unitless value between 0.3 and 3.0 multiplied by 10, i.e., 1.5 would be 15 decimal and would be sent as 0E hex.

Table 8. Rate Gain Query Message

BYTE	DATA	DESCRIPTION
7-8	G?	G?= Request/Query Rate Gains
9-10	prg	Pan rate gain value (multiplied by 10, i.e., returned value of 13(hex) -> 19/10=1.9 decimal.
11-12	trg	Tile rate gain value (multiplied by 10, i.e., returned value of 13(hex) -> 19/10=1.9 decimal.
13-16	nnnn	Unused

1.5.9. Initialization Response Message

Table 9. defines the message returned after a device has been directed to and has completed an initialization sequence. If the initialization failed, the embedded gimbal control software will shut down the motors.

Table 9. Initialization Result Message Format

BYTE	DATA	DESCRIPTION
7-8	Accw	Azimuth (pan) counterclockwise (+ direction) result (2 bytes)
9-10	Acw	Azimuth (pan) clockwise (- direction) result word (2 bytes)
11-12	Eccw	Elevation (tilt) counterclockwise (+ direction) result (2 bytes)
13-14	Ecw	Elevation (tilt) clockwise (- direction) result (2 bytes)
		2 Byte Result Explanation: Least Significant Byte: bit 0 - Center switch: 1 = detected (good) bit 1 - reserved bit 2 - Hard limit: 0 = not detected (good), bit 3 - Physical limit switch: 1 = hit (good) bits 4…7 - reserved Most Significant Byte: bits 4…7 - reserved

1.5.10. Software Limits Response Message

Table 10. defines the set the software limits command and associated data and also defines the request current software limits command and the format of the information returned by the gimbal. Refer to Section 1.5 for the definition of clockwise and counterclockwise terms.

Table 10. Software Limits Set Command Format, Query Command & Response Message Format

BYTE	DATA	DESCRIPTION
7	L	L= Software Limit command
8	S or ?	S= Set Limits, ?= Query Limits
		The follow information is sent with the Set command (LS) and is returned by the device after receiving a Query command (L?). Byte numbers in parenthesis correspond to the response message bytes sent by the device.
9 (7)	$accw_2$ (MS nibble)	Azimuth (pan) counterclockwise (+ direction) limit, bits 8-11
10 (8)	$accw_1$	bits 4-7
11 (9)	$accw_0$ (LS nibble)	bits 0-3
12 (10)	acw_2 (MS nibble)	Azimuth (pan) clockwise (- direction) limit, bits 8-11
13 (11)	acw_1	bits 4-7
14 (12)	acw_0 (LS nibble)	bits 0-3
15 (13)	$eccw_2$ (MS nibble)	Elevation (tilt) counterclockwise (+ direction) limit, bits 8-11
16 (14)	$eccw_1$	bits 4-7
17 (15)	$eccw_0$ (LS nibble)	bits 0-3
18 (16)	ecw_2 (MS nibble)	Elevation (tilt) clockwise (- direction) limit, bits 8-11
19 (17)	ecw_1	bits 4-7
20 (18)	ecw_0 (LS nibble)	bits 0-3

1.6. Pan/Tilt Mount Commands

1.6.1. General Commands

The control unit can specify the mount to pan left, right, or stop panning. The speed is adjustable.

The control unit can specify the pan/tilt mount to tilt down, up, or stop tilting. The speed is adjustable.

The control unit can specify the pan/tilt mount to program a preset position.

The control unit can specify the pan/tilt mount to move to a preset position.

The number of preset positions defined in this protocol is 256 with a range from 00..FF. If an access is made to an unimplemented position, the reply shall be a NIC. The implemented positions should have “home” as a default value.

The control unit can specify the pan-tilt mount to move to an absolute azimuth and elevation position determined by two 16-bit values, each represented by four hexadecimal ASCII characters. The 16-bit position values correspond to the number of degrees from the center switch for each axes. Positive values indicate a position counterclockwise from the center switch. Negative values indicate a clockwise position from the center switch. See Section 1.5 for the definition of clockwise and counterclockwise.

The control unit can request the current position from the pan-tilt mount. The position data is formatted as a 16-bit value represented by four hexadecimal ASCII characters for both azimuth and elevation.

Table 15. shows the format for the command to direct the pan-tilt mount to go to a position and the result of a position request (P?) command.

The control unit can specify a discrete pan and tilt speed in hundredths of degrees per second ((deg/sec) / 100), see Table 12.

The control unit can specify a discrete pan and tilt acceleration in hundredths of degrees per second squared ((deg/sec^2) / 100), see Table 12.

The proportional speed range for pan and tilt moves is 0...15, where 0 may be the slowest speed or a time-dependent proportional rate, and 15 is the maximum speed for a move.

The control unit can request the current pan and tilt rates/speeds from the device. The rates returned are in hundredths of degrees per second times ((deg/sec) / 100), see Table 12. and Table 13.

The control unit can request the current pan and tilt accelerations form the device. The accelerations are returned in hundredths of degrees per second squared ((deg/sec^2) / 100), see Table 12. and Table 13.

The control unit can command the pan/tilt mount to move to a position such that both axes arrive at the designated position at the same time. This is referred to as a vector move.

The pan/tilt command format is shown in Table 11.

Table 11. Pan/Tilt Mount Commands

COMMAND	DESCRIPTION	RESPONSE	COMMENTS
AP*nnnn*	Set acceleration for Pan Moves (deg/sec^2 * 100)	ACK	See Table 12.
AT*nnnn*	Set acceleration for Tilt Moves (deg/sec^2 * 100)	ACK	See Table 12.
A?	Request Current Acceleration Settings	ACK + Result	See Table 12. and Table 13.
HA	Halt	ACK	Stores current position. Turns off motors.
HV	Verify Home	ACK + Response	See Table 14. for response format.
HO	Go to the "Home" Position	ACK	Home position is azimuth = 0 and elevation = 0 degrees (level)
HS	Stop	ACK	Turns off motors. Leaves brake released(optional).
P?	Current Position Query	ACK + Result	See Table 15. for format
P $a_3a_2a_1a_0$ $e_3e_2e_1e_0$	Go to Pan-Tilt Position	ACK	Go to specific pan and tilt position. See Table 15. for format
PL	Pan Left (CW)	ACK	At constant speed. Send PS to stop.
PR	Pan Right (CCW)	ACK	At constant speed. Send PS to stop
PS	Pan Stop	ACK	
PT	Stop Motion	ACK	
P*nn*	Go To Preset nn	ACK	nn = 00…FF (256 presets) Note: Three byte command
P?*nn*	Get Preset nn	ACK + response	nn = 00…FF (256 presets) see Table 15.
RP*nnnn*	Set Discrete Rate/Speed for Pan Moves	ACK	Hundredths of degrees/second, see Table 12.
RT*nnnn*	Set Discrete Rate/Speed for Tilt Moves	ACK	Hundredths of degrees/second, see Table 12.
R?	Request Current Rate Settings	ACK + response	See Table 12. and Table 13.
S*nn*	Store Current Position as Preset nn	ACK	nn = 00…FF (256 presets) Note: Three byte command
SP0…SPF	Set Proportional Speed for Pan Moves	ACK	SP0 = slow/proportional, SPF = maximum
ST0…STF	Set Proportional Speed for Tilt Moves	ACK	ST0 = slow/proportional, STF = maximum
S?*nn*	Request Position Stored at Preset nn	ACK + Result	See Table 15. for result message format.
TD	Tilt Down (CW)	ACK	At constant speed. Send TS to stop.
TS	Tilt Stop	ACK	
TU	Tilt Up (CCW)	ACK	At constant speed. Send TS to stop.
VM*n*	Set Vector Motion	ACK	n: '1' = vector on, '0' = vector off

Table 12. Acceleration and Rate Query and Set Command Formats

BYTE	DATA	DESCRIPTION
7	A or R	A = acceleration, R = rate
8	P or T	Pan or Tilt axis
9	?	Query
Or		
9-12	XXXX	Set Command: ASCII hexadecimal representation in degrees/sec or degrees/sec^2 multiplied by 100 (i.e. 10.25 would be sent as 1025 which equals 0401 hexadecimal), i.e., values are sent as hundredths of degrees/sec or degrees/sec^2 .

Table 13. Acceleration and Rate Query Response Format

BYTE	DATA	DESCRIPTION
7-10	XXXX	ASCII hexadecimal representation in degrees/sec or degrees/sec^2 multiplied by 100 (i.e. 10.25 would be sent as 1025 which equals 0401 hexadecimal), i.e., values are sent as hundredths of degrees/sec or degrees/sec^2 .

Table 14. Verify Home Response Message Format

BYTE	DATA	DESCRIPTION
7	ls	Positive (CCW) direction limit switch found: bit 0: Azimuth limit switch found bit 1: Elevation limit switch found
8	cs	Center switch found: bit 0: Azimuth center switch found bit 1: Elevation center switch found
9	c	Encoder count is correct and gimbal is on center switch: bit 0: Azimuth count correct (initialized) bit 1: Elevation count correct (initialized)

1.6.2. Pan-Tilt Mount Go-To Position Command and Response Message

Table 15. shows the format for the command to direct the pan-tilt mount to go to a position and the result of a position (P?) request message. The response does not include the command character.

Table 15. Pan - Tilt Mount Go To Command and Response Message Format

BYTE	DATA	DESCRIPTION
7	P	P = position command or status request
8	a3 (MS nibble)	Azimuth (pan) position, bits 12-15. Values are sent as hundredths of a degree.
9	a2	bits 8-11
10	a1	bits 4-7
11	a0 (LS nibble)	bits 0-3
12	e3 (MS nibble)	Elevation (tilt) position, bits 12-15. Values are sent as hundredths of a degree.
13	e2	bits 8-11
14	e1	bits 4-7
15	e0 (LS nibble)	bits 0-3

1.7. Camera Commands

1.7.1. General Command and Response Message

The control unit can command the camera to focus near and far, and adjust the focus speed.

The control unit can command the camera to zoom in, out and adjust the zoom speed.

The control unit can command the lens to a specific zoom and focus position, with up to 12-bit resolution. Each 4-bit nibble is represented by a hexadecimal ASCII character (for example, 0x4BF). See Table 17 for the position parameter format.

The control unit can read zoom and focus information from the camera..

The camera control commands are shown in the following table:

Table 16. Camera Control Commands

COMMAND	DESCRIPTION	RESPONSE	COMMENTS
CN	Power On	ACK	Enables Lens and Camera Relay
CX	Power Off	ACK	Disables Lens and Camera Relay
F0…F9	Focus Speed	ACK	F0 = slowest, F9 = fastest
FA	Auto-focus On	ACK	Not Implemented
FF	Focus Far	ACK	Send FS to stop focus
FN	Focus Near	ACK	Send FS to stop focus
FS	Focus Stop	ACK	
FX	Auto-focus Off	ACK	Not Implemented
V*z2z1z0f2f1f0*	Lens Go-To Command	ACK	Sets lens zoom and focus position See Table 17 for format
V?	Lens Position Request	ACK + Result	See Table 17 for result format
Z0…Z9	Zoom Speed	ACK	Z0 = slowest, Z9 = fastest
ZA…ZF	Digital Zoom	ACK	Not Implemented
ZI	Zoom In	ACK	Send ZS to stop zooming in
ZO	Zoom Out	ACK	Send ZS to stop zooming out
ZS	Zoom Stop	ACK	
*x*p*c	Configure Auxiliary Serial Port	ACK	"*p*" is Aux. Port (0 or 1) See Table 18 for format
*x*p*t	Transmit to Auxiliary Serial Port	ACK	"*p*" is Aux. Port (0 or 1) Format is: *x*p*t + up to 200 data bytes
*x*p*r*nnn*	Receive from Auxiliary Serial Port	ACK + Result	"*p*" is Aux. Port (0 or 1), *nnn* is maximum number of receive bytes. Result is string of up to *nnn* bytes
*x*p*n	Get No. of Characters in Aux. Port Buffer	ACK + Result	"*p*" is Aux. Port (0 or 1) Result is decimal ASCII character count
*x*p*f	Flush Aux. Port Buffer	ACK	"*p*" is Aux. Port (0 or 1)

Table 17. Lens Position Set Command and Result Message Format

BYTE	DATA	DESCRIPTION
7	V	V = go-to command
8	z2 (MS nibble)	Zoom position, bits 8-11 + 0x30 for 0-9 and + 0x41 for A-F
9	z1	bits 4-7 + 0x30 for 0-9 and + 0x41 for A-F
10	z0 (LS nibble)	bits 0-3 + 0x30 for 0-9 and + 0x41 for A-F
11	f2 (MS nibble)	Focus position, bits 8-11 + 0x30 for 0-9 and + 0x41 for A-F
12	f1	bits 4-7 + 0x30 for 0-9 and + 0x41 for A-F
13	f0 (LS nibble)	bits 0-3 + 0x30 for 0-9 and + 0x41 for A-F

Table 18. Serial Port Configuration Message Format

Message String; *x*p*c,baud,dataBits,stopBits,parity

Example: *x0c,9600,8,1,0 Aux. Port 0
9600 Baud
8 Data Bits
1 Stop Bit
No Parity

PARAMETER	DATA	DESCRIPTION
p	0 or 1	The Auxiliary Port Address
Baud	9600, 19200, etc	Baud rate for the auxiliary serial port. The maximum tested rate is 115,200
DataBits	5,6,7, or 8	The number of data bits per character
StopBits	1 or 2	The number of stop bits
Parity	0, 1, or 2	0=None, 1=Odd, 2=Even