Quickset PTZ protocol
PTCR-1000

Version 1.0.0

Abbreviations used in the document

NVM	Non volatile Memory
PTCR	Pan and Tilt unit
LLA	Latitude, Longitude and Altitude
GPS	Geo Positioning system
AZ	Azimuth
EL	Elevation
FIFO	First In First Out

Notice of Disclaimer

Quickset makes no claims, expressed or implied, regarding the usefulness of this protocol, its implementation, or its correctness. Any use of this protocol is the sole responsibility of the agency implementing the protocol. The contents of this document and the function of the protocol are subject to change without notice.

Questions:
Questions regarding this protocol, its implementation, use, and distribution should be addressed to:

Moog Quickset Inc.
3650 Woodhead Drive
Northbrook, IL 60062
www.moog.com, www.quickset.com

PTCR-1000 Embedded Controller Protocol Rev 1.0.0

09/19/11

Table of Contents

1. Pan & Tilt Communications Protocol Overview:

Quickset's PTZ Serial Line protocols prior to PTCR-1000 are master-Slave protocols. These protocols take place at levels 1, 2 and 7 of the OSI model.

A typical master-slave type system is,

- With one master node that issues explicit commands to one of the "slave" nodes (max. 255 in Quickset PTZ network) and process responses.
- The master always initiates communication.
- Slave nodes do not transmit data, without a request from the master node, and do not communicate with other slaves.
- The master node can have only one open Quickset PTZ transaction at any given time, where master addresses an individual slave. After receiving and processing the request, the slave returns a message (a 'reply') to the master. If addressed slave is not present, master can start another transaction after 100 ms.
- In this mode, a Quickset PTZ transaction consists of two messages: a request from the master, and a reply from the slave. Each slave must have a unique address (from 1 to 255) so that it can be addressed independently from other nodes.
- Quickset's PTZ Master Node has no specific address; only the slave nodes have an address.

The PTCR-1000 protocol is designed keeping in mind increasing demand of feature-set requirements. PTCR-1000 is capable to work in master-slave mode for backwards compatibility, but can be configured to behave as multi-master in special cases like target handover for tracking or broadcasting an alarm. PTCR-1000 is extended from PTCR96 protocol and holds similar command/response structure. Controller or software developed for PTCR96 protocol can be use with PTCR-1000 without any change for PTCR96 feature-set.

At the physical level, Quickset's PTZ over Serial Line systems may use different physical interfaces (RS422, RS485, and RS232). TIA/EIA-422 (RS422) Four-Wire, interface is the most common. As an option, TIA/EIA-485 (RS485) Four-Wire and TIA/EIA-485 (RS485) Two-Wire interface may also be implemented.

Bidirectional Communication:
The Master should not send packets faster than 30 times per second and, once communications is established, should wait for a valid response from the PTCR prior to transmitting the next packet. This is especially important during the saving of non-volatile parameters, preset, and tour information.

Unidirectional Communication:
The Master should not send packets faster than 30 times per second and, once communications is established, should wait for a 25 ms between subsequent commands. This is especially important during the saving of non-volatile parameters, preset, and tour information.

The following table gives a general representation of Quickset's PTZ serial communication stack compared to the 7 layers of the OSI model.

Layer ISO/OSI Model:

7	Application:	Quickset protocol command and data bytes.
6	Presentation:	Empty
5	Session:	Empty
4	Transport:	Empty
3	Network:	Empty

2	Data Link	Quickset protocol, bytes STX, ID, CHECKSUM and ETX
1	Physical	EIA/TIA-422 or EIA/TIA-485 or EIA-232

1.1 Byte Format

Transmitters will format a single character and receivers will be able to decipher a single character with: 1 start bit, 8 data bits, 1 stop bit, and no parity. All units that support PTCR-1000 and PELCO-D Protocol have an ability to operate from 2400 baud to 115200 baud.

• Coding System: 8-bit binary

• Bits per Byte: 1 start bit

• 8 data bits, least significant bit sent first

• 1 stop bit

• No parity

1.2 Auto baud:

An automatic baud detection procedure (auto baud) has been implemented. Supported baud rates are 2400, 4800, 9600bps, 14.4kbps, 19.2kbps, 28.8kbps, 38.4kbps, 57.6kbps and 115.2 kbps. Structure is 8 data bits, 1 stop bit, and no parity. Auto baud synchronization will only occur on power up and will not adapt "on the fly". The baud rate will be retained as long as power is applied to the PTCR. To change baud rates, the user may power down the PTCR, change host baud rate, and then power the PTCR back up to redetect.

Normally, the user should make a practice of sending "keep alive" (Get Status/Jog) commands periodically if communications is lost in order to re-sync the PTCR to the host, even without the auto baud feature. With auto baud the host must be able to transmit up to 125-150 bytes of data (lowest baud rate worst case) without a response expected in order for the PTCR to properly determine the data rate. This equates to about 15 Get Status/Jog commands. With a refresh rate of 100ms a maximum of about 2 seconds is required to determine baud rate. Once baud rate is determined the PTCR will begin responding to the Get Status/Jog commands and the host software may then move on to actual operation. Note that units configured in a shared network can sync on packets destined for another unit that being said, all unit on one network should be configured for same baud rate (If fix baud rate configuration is selected).

1.3 Storage of Coordinate Information:

The PTCR tracks position using feedback from incremental encoders connected to the pan & tilt axes. Incremental encoder counts are continuously stored in non-volatile memory and are corrected as required at each crossing of the index position. This provides a “pseudo” absolute position. Each encoder is sampled and a running count is maintained, incremented, or decremented according to the current state of the encoder. This value is then stored in a non-volatile FRAM memory. The encoder also provides an index pulse once per revolution. When encoder crosses the index pulse, its reset the count to zero. The physical center is equated to a resolver value relative to index pulse on both axis and is stored in non-volatile EEPROM. The current running value stored in non-volatile memory is updated frequently and will provide high accuracy at power up. However, if the platform is physically moved or bumped off position when powered down, the encoder changes will be missed and current position accuracy will be compromised. Whenever the index point is crossed during movement the encoder count will be automatically reset to zero and from that point onwards unit should have accurate positions. Optionally, the user may also reset to maximum accuracy by performing a homing cycle. (See command 9EH). This command will automatically move the platform to the index point, realign the encoder count, and then return the platform to its original but corrected position.

PTCR1000 do not define encoder resolution. The resolution reported back depending on what is selected while sending the commands. In each command there is specific bit to sepcify resolution between 0.01 and 0.001. The unit may have fitted with 9000-line or higher encoder.

If PTCR is fitted with 9000 line encoders. The 9000-lines can be further decoded to detect 36000 valid transitions. Since the encoder is directly connected to the platform axis one encoder revolution of 36000 transitions equates to one platform revolution of 360.00°. Therefore, a single transition equals 0.01° of movement. To get 0.001° of movement, the PTCR should be fitted with 90000 line encoderes or higher. If encoders are other then 9000 and 90000, then there may be integer calculation inaccuracies will associate with it for converting encoder resolution back to 0.01° and 0.001°.

Absolute resolver units are used internally to track position. This does have an effect on the structure of preset table entries and user-defined angular corrections. Presets are stored internally in absolute resolver units. The angular position returned for any of those presets is a calculated value derived from the conversion of resolver units to angles and any angular offsets defined by the user. Let us say, for example, that the user has not defined any angular offset. Therefore, the 0°/0° point for the platform will truly read 0°/0°. The user then defines a preset at +20° in pan and -10° in tilt. If the user moves to this preset, the angular reading returned will be +20°/-10°. The user then decides to redefine the 0°/0° position to return -10° in pan and 0° in tilt by entering a -10° pan correction. Though the physical 0°/0° position remains the same, the angular position returned will now be -10°/0°. If the user then moves to the former +20°/-10° preset the platform will move to the same preset position but the angle returned will be +10°/-10°. This also applies to encoder with resolutions other then 9000 and 90000.

There is an important rational behind this coordinate system. Many will use the angular readings of the platform to observe the relative position of different targets. For example, the user may find that one object resides at -30° pan/-10° tilt relative to a true 0°/0° point. A second object resides at +25° pan/-20° tilt relative to the 0°/0° point. The user moves back to the first object and uses the "Align Angles to Center" command. This will introduce a pan angle correction of +30° and a tilt correction of +10°, adjusting the returned angular reading for the first object to 0°/0°. If the user now moves to the second object the angular reading will be +55° pan/-10° tilt, the actual angular distance from the first object.

Therefore, presets will always reflect known physical positions of the platform. No matter how the pan and tilt angles are shifted through offsets, a preset position will remain valid and correct and a command to move to that position will still return the platform to it. Introducing angular corrections will do nothing more than change the returned angular position but will not change the actual physical position of a preset. Additionally, software limits will also show this shift. Since a software limit is designed to prohibit movement beyond an absolute point of platform rotation, the actual position of the software limit will not change when an angular offset is set. Only the value returned will change.

1.4 Camera and Lens Control:

In the PTCR unit several non-volatile parameters are set and stored for identifying the operation of the camera/ lens and for determining whether a camera/ lens has serial remote control capabilities (or analog lens control) and, if so, what the communications level, baud rate, number of data bits, and parity are. At start-up the main microcontroller sends these parameters to the slaves. This determines how the slaves interprets data and formats it for the camera and lens.

Analog Lens control is straightforward. PTCR unit is capable to control two or three proportional-speed bi-directional motor drivers per payload. One driver controls the zoom motor, the second controls the focus motor and if third is available can be used to control iris motor. Lenses are fitted with potentiometers for position feedback. The potentiometer output for each axis is connected to the microcontroller's ADC and is converted to a binary value indicating current position. This value is then returned to the host. The host may send a request to move the zoom and focus motors to an undetermined final position (jog) or a predetermined final position (move to.) These predetermined final positions are stored as part of the pan and tilt's preset table, allowing the system to move to specific pan and tilt coordinates while also adjusting zoom, focus and iris on both cameras.

Serial camera/ lens control is more difficult due to the numerous control signals available for each camera/ lens type. It can be acheived through two different ways depending on underlaying hardware configuration.

The first way is legacy PTCR96 pass through command. A special variable-length command 62H is used to wrap a complete camera command for transfer to the camera/ lens. The user should build a complete camera command

string. It should then be prefixed and appended with the appropriate PTCR control bytes for transfer to the PTCR. The PTCR will then strip the intact camera command out of the packet and transfer it to the camera/ lens. See command 62H, "Command Camera", below for more detail. The legacy 62H command is extended to communicate more than two devices as limited in PTCR96 protocol. User can get response to the passed command either using the "Get Status/Jog" command and response or can send 62H command to query response to previously send command. The PTCR will clear response from its buffer after sending it to user and will no longer be available. In Jog/Status response the PTCR indicates that a camera string is available by changing the "cam count" byte for a camera from 0 to the number of bytes returned. When the host sees this change the indicated number of bytes should be removed from the response and stored for camera response parsing. The PTCR will return the camera's data string verbatim, including all control characters, checksums, etc. See command 31H for more information.

The second way is to use PTCR1000 native control command/response for controlling camera/ lens features. This supported only on certain hardware configurations including Oracle board sets. These command provides control of 32 variables of 16 bit size and 64 toggle flags to control on/off actions.

1.5 General Communications Structure:

The general structure of the host-to-PTCR communications protocol is as follows:

Data	Format	Bytes	7	6	5	4	3	2	1	0
STX	02H	1								
Identity	xxH	1	1-99 (01H-63H) for RS-485 identity or 00 for dedicated/broadcast							
Command	xxH	1								
Data										
▼										
Data										
LRC	xxH	1								
ETX	03H	1								

A transfer from the host to the PTCR-95 will always start with a unique STX (Start-of-Text) 02H character followed by an identity address for the connected unit. The command number to execute and any command data will follow. The packet will then be completed by sending the LRC (Longitudinal Redundancy Checksum) and the ETX (End-of-Text) 03H character.

Data	Format	Bytes	7	6	5	4	3	2	1	0
ACK	06H	1								
Identity	xxH	1	1-99 (01H-63H) for RS-485 identity or 00 for dedicated/broadcast							
Command	xxH	1								
Data										
▼										
Data										
LRC	xxH	1								
ETX	03H	1								

The response from the PTCR-95 is very similar. It will always start with a unique ACK (Acknowledge) 06H character followed by an identity address for the connected unit. The unit will echo the command number for confirmation of the received command and include any data associated with the command number. The packet will then be completed by sending the LRC (Longitudinal Redundancy Checksum) and the ETX (End-Of-Text) 03H character.

When operating in RS-485 daisy-chain mode the identity address is used to determine which of the connected pan and tilt units is being commanded and should respond. All others will ignore the command and remain quiet. This address is set using Command 9FH and can range from 00-99. Address 00 is reserved for dedicated RS-232 and RS-422 mode and for broadcast mode. When the 00 address is used it will be internally ignored and any attached PTCR will respond. The current PTCR identity will be returned as part of the standard responses. If the user needs to change the identity of a unit the 9FH identity command should be sent using the current identity as the identity address and the new identity as the data byte. The PTCR will return the current identity as its identity

address and the new identity as its response. **From that point forward, the new identity will take effect.** If the user forgets the current identity it can always be set and retrieved using an identity address of 00. **However, all other platforms in the network must be disconnected before issuing ANY commands using address 00. Otherwise, all units will react to the command and seize the host receive line.**

In the command breakdowns that follow, the transmission from remote to PTCR will always be followed by the response from PTCR to remote. The command listings will show only the command number and pertinent data. **However, all transfers require and will return the control characters in the sequence listed above.**

1.6 Control Characters:

The following are definitions for control characters used in data transfer.

Char	Description	Sent By	Value
STX	Start of Text	Host	02H
ETX	End of Text	Host/PTCR	03H
ACK	Acknowledge	PTCR	06H
NAK	Not Acknowledge	PTCR	15H

1.7 Calculating the LRC Checksum:

The checksum used for data transfer is a longitudinal redundancy check or LRC. It is calculated by XOR'ing bytes starting with the identity address and ending with the last data byte. The ACK/NAK/STX and ETX are **not** included in the LRC. The easiest method of calculating and comparing is to XOR all data bytes, then XOR the result with the LRC checksum. The result should be 0 (zero).

If a command string is received from the host, is parsed, and is found to have an incorrect checksum the PTCR will not respond. It is possible that the corruption occurred due to an improperly transmitted identity. Therefore, if the unit did respond it could collide with the return data from a properly addressed unit.

1.8 Passing Data that Matches Control Character Values (ESC/bit-7 Set):

When passing full 8-bit bitsets it is possible that a value may match a control character (ACK/NAK/STX/ETX.) Therefore, the protocol needs some method of distinguishing these values from control characters. The method used is the insertion of an ESC character prior to transmitting the conflicting data byte and the setting of Bit-7 of the conflicting byte. Since we must also be able to distinguish the ESC value of 1BH we will perform the same operation on ESC's.

Example: Data to send = 02H Data sent = 1BH 82H
Data to send = 1BH Data sent = 1BH 9BH

This insertion should be performed on **any byte that is not a control character, including the LRC.** Note that this procedure should be performed immediately prior to transmission and the companion decoding should be performed prior to checksum calculation after reception. These insertions are not included in the LRC calculations. The entire receive buffer should be scanned prior to LRC check and parsed for any occurrences of ESC. The ESC should be tossed, the following byte should have Bit-7 cleared, then the buffer should be shifted down. The buffer will then be ready for LRC calculation and data parsing.

The inclusion of ESC sequences provides two distinct advantages. First, the user is assured that any reception of an ACK or ETX is valid. Unless an error occurs, these control characters will not show up as data bytes in the packet. Therefore, it is perfectly valid to cue the start of reception on any ACK character. It is also valid to cue the end of reception and begin parsing data on the reception of an ETX. Since the ACK and ETX are not used in the calculation of the LRC, the user can simply use them as starting and stopping cues. Secondly, unlike the original implementation of IBM bisync on which this protocol was based, the user is allowed to pass full 8-bit binary values. Code snippets are provided at the end of this document demonstrating different approaches in C for implementing LRC calculation, LRC checking, and ESC insertion and removal.

1.9 Passing 16-bit, 24-bit, and 32-bit Integer Values:

As noted in the protocol command descriptions, some values sent between the host and PTCR units are integer values larger than a single byte. Any multi-byte integer value should be passed and received as a signed two's-complement little endian integer split between multiple individual bytes. The first byte should represent the LSB of the integer with the last byte containing the MSB of the integer. Negative values are represented as the two's-complement of the positive value. Sign is carried in the most significant bit of the MSB. For example:

16-bit Value	Integer in Hex	First Byte	Second Byte
32767	7FFFH	FFH	7FH
2	0002H	02H	00H
1	0001H	01H	00H
0	0000H	00H	00H
-1	FFFFH	FFH	FFH
-2	FFFEH	FEH	FFH
-32768	8000H	00H	80H

24-bit Value	Integer in Hex	First Byte	Second Byte	Third Byte
8388607	7FFFFFH	FFH	FFH	7FH
32767	007FFFH	FFH	7FH	00H
2	000002H	02H	00H	00H
1	000001H	01H	00H	00H
0	000000H	00H	00H	00H
-1	FFFFFFH	FFH	FFH	FFH
-2	FFFFFEH	FEH	FFH	FFH
-32768	FF8000H	00H	80H	FFH
-8388608	800000H	00H	00H	80H

Valid signed 16-bit values can be built by the host by simply assigning the first received byte to the LSB of the integer and the second byte to the MSB of the integer. Most PC-based systems will directly recognize these values as legitimate signed 16-bit integers. However, very few systems work directly with 24-bit integers. The three byte values are used to save bandwidth when transferring angular positions. They can be easily converted to the more standard 32-bit integer values used by most PC-based platforms by simply observing the sign bit, bit 7 of the third byte. If the sign bit is set also set the most significant byte of the 32-bit integer to FFH. If the sign bit is clear set the most significant byte of the 32-bit integer to 00H. For example, 8388607 convert to 007FFFFH as a 32-bit number. -8388608 converts to FF800000H.

1.10 The Programmer's Responsibility for Input Range and Format

The microcontroller used for the PTCR has a relatively small amount of code space. Extensive range and format checking of all users input would seriously task the processor and limit the amount of space left for executable procedures. The listings for each command in this document present the allowable range for each byte or integer of data. The user/programmer is responsible for providing inputs that are properly formatted and within the specified numeric range for each command. However, **absolute coordinates, preset numbers and static tour step locations** are checked and flagged if out of range or otherwise invalid.

1.11 Status Definitions:

Bits are active high, i.e., “set” or “1” indicates the condition exists. (S)oft faults are self-healing. (H)ard faults require a RESET (RES) command to clear.

Sym	Type	Name	Description
xHL	S Fault	Hard Limit	An axis hard limit has been reached.
xSL	S Fault	Soft Limit	An axis soft limit has been reached.
TO	H Fault	Timeout	A commanded axis is not moved within the prescribed timeframe.
DE	H Fault	Direction Error	A commanded axis has moved in the wrong direction.
xxxM	Stat	Moving	The commanded axis is currently moving.

OSLR	Stat	Override Return	The controller is in soft limit override.
DES	Stat	Destination	The coordinates returned are destination coordinates, not current.
EXEC	Stat	Executing	The PTCR is executing a remote initiated command.
CON	Stat	Cont Rotation	Platform is continuous rotation. Pan soft/hard limits are ignored.

2. PTCR Command Set:

This section includes information on the actual commands used to control and monitor the pan & tilt unit and attached camera.

2.1 Numeric Command List:

Cmd	Name	Cmd	Name
	Jog / Move Commands		*Configuration Commands - 1*
30H		90H	Get Center Position in RU's
31H	Get Status/Jog (This command has extensions)	91H	Set Center Position
32H	Move To Preset	92H	Get/Set Pan and Tilt Ramp Parameters
33H	Move To Entered Coordinates	93H	
34H	Move To Delta Coordinates	94H	Initialize Preset Table to 0/0
35H	Move To Absolute 0/0	95H	Get/Set Motor and Resolver Direction
36H	Move To Home	96H	Get/Set Communication Timeout
37H	Start Preset Tour (for Legacy) (0,1,2)	97H	Get/Set Heater Configuration. (All Heaters)
38H		98H	Get/Set Unit Report resolution(only in new units) and encoder counts
39H		99H	
3AH	Move To Entered Pan/Tilt/Zoom/Focus	9AH	Get Firmware Revision/s
3BH	Move To Entered Pan/Tilt/Zoom/Focus with max speed limit	9BH	Get/Set Pan and Tilt Ramp - II parameters
3CH		9CH	Get/Set/Store Maximum automated move speeds
3DH	Move Motor (Direct step control of motor)	9DH	Initial Encoder Align
3EH		9EH	Perform Homing Cycle (extended command)
3FH		9FH	Get/Set Identity Address
	Preset Commands		*Configuration Commands - 2*
40H	Retrieve Preset Table Entry	A0H	
41H	Save Coordinates As Preset Table Entry	A1H	Get/Set Pan Step Rate Parameters
42H	Save Current Position As Preset Table Entry	A2H	Get/Set Tilt Step Rate Parameters
43H	Add Current Z/F To Preset Table Entry	A3H	Set MAC address (Factory only. Requires password)
44H		A4H	Set Serial Number / Model Number (Factory only. Requires password)
45H		A5H	
46H		A6H	
47H		A7H	
48H		A8H	
49H		A9H	
4AH		AAH	Start Bootloader
4BH		ABH	ID setting for Oracle IBUS (Factory only)
4CH		ACH	
4DH		ADH	
4EH		AEH	
4FH		AFH	
	Tour Commands		*Loop Gains, Stabilization,Rates and special functions*
50H	Flush Preset Tour	B0H	
51H	Query Preset Tour	B1H	Get/Set Stabilization
52H	Append To Preset Tour	B2H	Get/Set PID
53H	Insert Into Preset Tour	B3H	Get/Set Drift Compensation

54H	Delete From Preset Tour	B4H	
55H	Replace In Preset Tour	B5H	
56H	Get Tour Size	B6H	
57H		B7H	
58H		B8H	
59H		B9H	
5AH	Get / Set PTZF Tolerance for Tour (Factory)	BAH	Get / Set PTCR's Geo Position
5BH		BBH	Get/Set PTCR's North position
5CH		BCH	Move To Entered GPS Target location
5DH		BDH	Get / Set / Clear GPS Preset
5EH		BEH	Query / Clear GPS target History points
5FH		BFH	
	Camera / Lens Commands		*Debugging and Health related*
60H	Get/Set Camera Communication Parameters	C0H	
61H	Get/Set Lens Parameters	C1H	Get Health Data (Voltage, Current, Temperature etc.)
62H	Command Selected Camera	C2H	
63H	Reserved for Future for Video Switching	C3H	
64H	Get/Set Camera/Power/Video/OSD (96 style)	C4H	
65H	Get/Set Camera Timeouts	C5H	
66H	Get/Set Aux Control Outputs	C6H	
67H		C7H	
68H		C8H	Reset Debug Parameters
69H		C9H	Get Debug Parameters
6AH		CAH	
6BH		CBH	
6CH		CCH	
6DH	Set Special Features	CDH	
6EH		CEH	
6FH		CFH	Custom Debug Command (Get Position Coordinate Transform Data)
	Video Switch and OSD		*Misc.*
70H	Get/ Set OSD angle format (Standard 96 function)	D0H	Get Serial Number / Model Number
71H	Get/ Set OSD site name / number format	D1H	Get Serial Type (RS232, 422, 485 4W, 2W)
72H		D2H	
73H		D3H	
74H		D4H	
75H	Get OSD name string (This command is extended)	D5H	
76H	Set/Get Status of different OSD lines	D6H	
77H		D7H	
78H	Payload Configuration Commands (Factory)	D8H	
79H	Payload Configuration Commands (Factory)	D9H	
7AH		DAH	
7BH		DBH	
7CH		DCH	
7DH		DDH	
7EH		DEH	
7FH		DFH	

	Offsets and Limits		*Intelligent Control*
80H	Set Pan & Tilt Angle Correction	E0H	
81H	Get/Set Pan and Tilt Soft Limit	E1H	
82H	Align Angles To Center	E2H	
83H	Align Angles To Coordinates	E3H	
84H	Clear Angle Correction	E4H	
85H	Get Pan & Tilt Angle Correction	E5H	
86H	*Get All Soft limits*	E6H	
87H		E7H	
88H		E8H	
89H		E9H	
8AH	Get/Set Pan and Tilt Encoder corrections	EAH	
8BH		EBH	
8CH		ECH	
8DH		EDH	
8EH		EEH	
8FH	Custom command - 1 (Get/Set NF Registration Angles)	EFH	

2.2 "Get Status/Jog" & "Keep Alive" Command

This command can be used as a standard "Keep Alive" from the host unit to continuously gather coordinate data. This command should be the one sent when no other command is required. This will keep the host advised of current position and any faults that may exist. This will also confirm to the PTCR that the host is connected and properly communicating.

The PDIR and TDIR bits set the motor direction for jog commands. By default, PDIR set is CW and TDIR set is UP. The PSLO and TSLO bits reduce the overall jog speed range for an axis by 64. This is only functional in jog mode. Automated moves will still execute at normal speed whether the bits are set or clear. The STOP bit can be toggled on and off to stop motors and terminate an automated move or a tour. The RES bit is used to clear latching (hard) faults. These include motor directional errors (DE) and timeouts (TO). A timeout fault will be set if an axis fails to move within 1 second. This may be the result of a stalled motor or an overloaded platform. A directional error fault will be set if an axis is detected as moving in the wrong direction. This may be the result of improper motor wiring. **Note that the DE and TO faults will only occur during automated moves.** The user should observe the angular readings during jog to confirm motors are moving the proper direction and are not stalled. The OSL bit allows overriding soft limits during jog. **This bit should only be set when initially setting up the soft limits.** Its current setting will be returned in the OSLR bit.

The jog bytes provide full speed range from 0-255 for each axis. This allows proportional, simultaneous jog control of both axes. Holding the speed value for an axis at 0 will prohibit jogging that axis. Note that any jog command that includes a speed value other than 0 will automatically stop any automated command. Therefore, initiating a jog can be used to terminate an automated move, including a preset tour. Four commands provide zoom and focus jog control for both cameras. These commands embed both speed and direction for zoom and focus. Zoom and focus can be adjusted independently of pan & tilt jog.

Since camera response data can be embedded in the status response, the response packet is variable length. The user should cue on the reception of a valid ETX to end the packet. If the platform has no camera response data to return the camera byte counts will be 0. However, if they are greater than 0, the host should read the additional bytes when parsing the packet. Note that the camera byte count reflects the actual length of the camera response and does not include a count of any ESC characters that may have been inserted to keep the return data from conflicting with any control characters.

- Jog/Status command

Data	Format	Bytes	7	6	5	4	3	2	1	0
Cmd Num	31H	1								
Cmd	Bitset	1	PDIR[1]	TDIR[2]	0	PSLO	TSLO	OSL	STOP	RES
Pan Jog Cmd	Bitset	1	Pan Speed (0-255)							
Tilt Jog Cmd	Bitset	1	Tilt Speed (0-255)							
Zoom 1 Jog	Bitset	1	Zoom Speed (0-127)							Dir[3]
Focus 1 Jog	Bitset	1	Focus Speed (0-127)							Dir[4]
Zoom 2 Jog	Bitset	1	Zoom Speed (0-127)							Dir[3]
Focus 2 Jog	Bitset	1	Focus Speed (0-127)							Dir[4]

[1]1 = CW/0 = CCW, 0 Speed = No Movement
[2]1 = UP/0 = DWN, 0 Speed = No Movement
[3]1 = Zoom Out/0 = Zoom In, 0 Speed = No Movement
[4]1 = Focus Out/0 = Focus In, 0 Speed = No Movement

Note 1: (Extension one)

- Jog/Status Extended command for 1024 linear speed steps

Data	Format	Bytes	7	6	5	4	3	2	1	0
Cmd Num	31H	1								
Cmd	Bitset	1	1	1	1	PSLO	TSLO	OSL	STOP	RES
Extension Type	Bitset	1	This specifies extension type. [1]							
Pan Jog Cmd	Bitset	2	Pan Speed (+/- 0-1023) [2]							
Tilt Jog Cmd	Bitset	2	Tilt Speed (+/- 0-1023) [3]							
Zoom 1 Jog	Bitset	1	Zoom Speed (0-127)							Dir[4]
Focus 1 Jog	Bitset	1	Focus Speed (0-127)							Dir[5]
Zoom 2 Jog	Bitset	1	Zoom Speed (0-127)							Dir[4]
Focus 2 Jog	Bitset	1	Focus Speed (0-127)							Dir[5]

[1]1 = 0-1023 linear steps
[2]+ve value = CW/ -ve value = CCW, 0 Speed = No Movement (LSB first)
[3]+ve value = UP/-ve value = DWN, 0 Speed = No Movement (LSB first)
[4]1 = Zoom Out/0 = Zoom In, 0 Speed = No Movement
[5]1 = Focus Out/0 = Focus In, 0 Speed = No Movement

- Response

Data	Format	Bytes	7	6	5	4	3	2	1	0
Cmd Num	31H	1								
PAN Coord	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00°							
TILT Coord	Int	3	TILT = -18000 to +18000 = -180.00° to +180.00°							
PAN Status	Bitset	1	CWSL	CCWSL	CWHL	CCWHL	TO	DE	0	0
TILT Status	Bitset	1	USL	DSL	UHL	DHL	TO	DE	0	0
Gen Status	Bitset	1	CON[2]	EXEC	DES[1]	OSLR	CWM	CCWM	UPM	DWNM
Zoom 1 Pos	Byte	1	0-255							
Focus 1 Pos	Byte	1	0-255							
Zoom 2 Pos	Byte	1	0-255							
Focus 2 Pos	Byte	1	0-255							
Cam 1 Count	Byte	1	0-80 for number of bytes of camera data to follow							
Cam 2 Count	Byte	1	0-80 for number of bytes of camera data to follow							
Cam 1 Data	Bytes	0-80	Camera return string in native format (only present if Cam Count 1 > 0)							
Cam 2 Data	Bytes	0-80	Camera return string in native format (only present if Cam Count 2 > 0)							

[1]DES bit is clear if coordinates are current, set if coordinates are the destination of a MOVE TO command.
[2]Indicates the platform is continuous rotation. Pan soft/hard limits should be ignored.

2.3 Automated "Move To" Commands:

Any "Move To" command should only be repeated until an acknowledgement has been received from the PTCR (echo of the command number). The host should then revert back to the standard 31H "Get Status/Jog" query. The PTCR will set the EXEC (executing) bit in the general status bit set to indicate that the command is being carried out. This bit will clear once the move has been completed.

The PTCR response to any automated "Move To" command will be identical to the standard status response with one exception. The response will echo the destination coordinates either as entered or retrieved from the preset table rather than the current coordinates. The setting of the destination bit DES, bit-5 of general status will indicate this. Status will then default back to current coordinates once the "Get Status/Jog" command/response resumes and the DES bit clears. The user may cue on the DES bit in order to fill a "Moving To" window with the destination coordinates. If the PTCR is detecting hard faults that will prohibit executing a "Move To" command it will echo the current position as the destination coordinates. This should act as a reminder for the user to check the fault status.

Movement will start once the PTCR has parsed the "Move To" coordinates. The setting of the appropriate axis MOVE bits in the status response will indicate this. As each axis arrives on station the respective MOVE bit will be cleared. The host may assume the move has been completed when all MOVE bits and the EXEC bit have cleared. If a fault occurs on **any** axis **all** motors will stop. The fault will be set and all MOVE bits and the EXEC bit will clear.

Setting the STOP bit in any following "Get Status/Jog" command will immediately terminate any automated "Move To" operation. *At this point it is intended to provide jog, camera and aux output control during tour. PTCR will ignore commands other then Jog, Camera and Aux during tour mode.*

2.3.1 "Move To Preset" Command:

The PTCR can retain up to 255 (0-254) position entries in a non-volatile preset table that are frequently used by the operator. Each position in this preset table represents following entities,

- Abs. Pan angle w.r.t PAN CENTER,
- Abs. Tilt angle w.r.t. TILT CENTER,
- Abs. Zoom / Focus/ Iris positions for CAMERA 1 (provided the lens has analog controls for Z/F/I)
- Abs. Zoom / Focus/ Iris positions for CAMERA 2 (provided the lens has analog controls for Z/F/I)

This command is used to move the platform to a preset position defined in this preset table. Further information on setting up the preset table is provided under command 40H.
(Note: Currently we support only 32)

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	32H	1								
Preset	xxH	1	Preset Number 0-254							

- Response

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	32H	1								
PAN Coord	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00°							
TILT Coord	Int	3	TILT = -18000 to +18000 = -180.00° to +180.00°							
PAN Status	Bitset	1	CWSL	CCWSL	CWHL	CCWHL	TO	DE	0	0
TILT Status	Bitset	1	USL	DSL	UHL	DHL	TO	DE	0	0
Gen Status	Bitset	1	CON	EXEC	1	OSLR	CWM	CCWM	UPM	DWNM
Zoom 1 Pos	Byte	1	0-255							
Focus 1 Pos	Byte	1	0-255							
Zoom 2 Pos	Byte	1	0-255							
Focus 2 Pos	Byte	1	0-255							

2.3.2 "Move To Entered Coordinate" Command:

This command is used to move the platform to a specific set of manually entered coordinates. The coordinate must consist of the desired position to 1/100th degree multiplied by 100, i.e., +90.00° should be sent as 9000. The user can force an axis to remain in position by sending its current position back to the PTCR. However, the coordinate value 99999 (+999.99°) can also be sent in order to prohibit movement of a specific axis. For example, if the user wishes to only move PAN, send 99999 (+999.99°) as the "Move To" coordinate for TILT and the TILT axis will remain stationary.

The PTCR will perform a range check for the input coordinates and abort the move if a coordinate is out of range. Allowable pan range is defined as -180.00° + pan angle offset to +180.00° + pan angle offset. Allowable tilt range is -90.00° + tilt angle offset to +90.00° + tilt angle offset. For an angle offset of 0°/0°, the range would be -180.00°/+180.00° and -90.00°/+90.00°. If a pan angle correction of +90.00° is entered, the allowable pan angle range would shift to -90.00° to 270.00°. If a tilt angle correction of -20.00° is entered, the allowable tilt angle range would shift to -110.00° to 70.00°.

- Command

Data	Format	Bytes	
Command	33H	1	
PAN Coord	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00° or 99999 for no move
TILT Coord	Int	3	TILT = -18000 to +18000 = -180.00° to +180.00° or 99999 for no move

- Response

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	33H	1								
PAN Coord	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00°							
TILT Coord	Int	3	TILT = -18000 to +18000 = -180.00° to +180.00°							
PAN Status	Bitset	1	CWSL	CCWSL	CWHL	CCWHL	TO	DE	0	0
TILT Status	Bitset	1	USL	DSL	UHL	DHL	TO	DE	0	0
Gen Status	Bitset	1	CON	EXEC	1	OSLR	CWM	CCWM	UPM	DWNM
Zoom 1 Pos	Byte	1	0-255							
Focus 1 Pos	Byte	1	0-255							
Zoom 2 Pos	Byte	1	0-255							
Focus 2 Pos	Byte	1	0-255							

2.3.3 "Move To Delta Coordinates " Command:

As opposed to moving to specific coordinates, the "Move To Delta Coordinate" command allows the user to move the platform a specific angular distance from the current position. The coordinate must consist of the desired position to 1/100th degree multiplied by 100, i.e., -20.00° should be sent as -2000. The user can force an axis to remain stationary by sending 0 to the PTCR for that axis.

- Command

Data	Format	Bytes	
Command	34H	1	
PAN Coord	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00°
TILT Coord	Int	3	TILT = -18000 to +18000 = -180.00° to +180.00°

- Response

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	34H	1								
PAN Coord	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00°							
TILT Coord	Int	3	TILT = -18000 to +18000 = -180.00° to +180.00°							

PAN Status	Bitset	1	CWSL	CCWSL	CWHL	CCWHL	TO	DE	0	0
TILT Status	Bitset	1	USL	DSL	UHL	DHL	TO	DE	0	0
Gen Status	Bitset	1	CON	EXEC	1	OSLR	CWM	CCWM	UPM	DWNM
Zoom 1 Pos	Byte	1	0-255							
Focus 1 Pos	Byte	1	0-255							
Zoom 2 Pos	Byte	1	0-255							
Focus 2 Pos	Byte	1	0-255							

2.3.4 "Move To Absolute 0/0" Command:

The pan & tilt unit encoders are initially aligned with the platform centered and level. This command will return the platform to that stored center position. This is a convenient method for returning the platform to factory center for maintenance or hard limit switch alignment.

Data	Format	Bytes
Command	35H	1

- Response

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	35H	1								
PAN Coord	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00°							
TILT Coord	Int	3	TILT = -18000 to +18000 = -180.00° to +180.00°							
PAN Status	Bitset	1	CWSL	CCWSL	CWHL	CCWHL	TO	DE	OL	PRF
TILT Status	Bitset	1	USL	DSL	UHL	DHL	TO	DE	OL	TRF
Gen Status	Bitset	1	CON	EXEC	1	OSLR	CWM	CCWM	UPM	DWNM
Zoom 1 Pos	Byte	1	0-255							
Focus 1 Pos	Byte	1	0-255							
Zoom 2 Pos	Byte	1	0-255							
Focus 2 Pos	Byte	1	0-255							

2.3.5 "Move To Home" Command:

A special preset position number 31, referred to as "Home", may be entered and stored by the PTCR. This command requires no preset number or coordinate input and will always return the pan and tilt unit to this "Home" position.

- Command:

Data	Format	Bytes
Command	36H	1

- Response

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	36H	1								
PAN Coord	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00°							
TILT Coord	Int	3	TILT = -18000 to +18000 = -180.00° to +180.00°							
PAN Status	Bitset	1	CWSL	CCWSL	CWHL	CCWHL	TO	DE	OL	PRF
TILT Status	Bitset	1	USL	DSL	UHL	DHL	TO	DE	OL	TRF
Gen Status	Bitset	1	CON	EXEC	1	OSLR	CWM	CCWM	UPM	DWNM
Zoom 1 Pos	Byte	1	0-255							
Focus 1 Pos	Byte	1	0-255							
Zoom 2 Pos	Byte	1	0-255							
Focus 2 Pos	Byte	1	0-255							

2.3.6 "Move To Entered Pan/Tilt/Zoom/Focus" Command:

This command is used to move the platform to a specific set of manually entered pan, tilt, zoom and focus coordinates. The coordinate for pan and tilt must consist of the desired position to 1/100th degree multiplied by 100, i.e., +90.00° should be sent as 9000. The user can force an axis to remain in position by sending its current position back to the PTCR. However, the coordinate value 99999 (+999.99°) can also be sent in order to prohibit movement of a specific axis. For example, if the user wishes to only move PAN, send 99999 (+999.99°) as the "Move To" coordinate for TILT and the TILT axis will remain stationary. The zoom and focus coordinates should be sent as a value between 0 and 255. Reference command 33H for range checking performed.

Connected lens systems may not be able to move across the full range of coordinates. If a zoom or focus position is entered that cannot be attained by the peripheral the unit will be moved as close as possible, then a timeout will allow the command to complete without generating an error.

- Command

Data	Format	Bytes	
Command	3AH	1	
PAN Coord	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00° or 99999 for no move
TILT Coord	Int	3	TILT = -18000 to +18000 = -180.00° to +180.00° or 99999 for no move
Zoom 1 Pos	Byte	1	1-255 (01H-FFH) or 0 to hold position
Focus 1 Pos	Byte	1	1-255 (01H-FFH) or 0 to hold position
Zoom 2 Pos	Byte	1	1-255 (01H-FFH) or 0 to hold position
Focus 2 Pos	Byte	1	1-255 (01H-FFH) or 0 to hold position

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	3AH	1								
PAN Coord	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00°							
TILT Coord	Int	3	TILT = -18000 to +18000 = -180.00° to +180.00°							
PAN Status	Bitset	1	CWSL	CCWSL	CWHL	CCWHL	TO	DE	0	0
TILT Status	Bitset	1	USL	DSL	UHL	DHL	TO	DE	0	0
Gen Status	Bitset	1	CON	EXEC	1	OSLR	CWM	CCWM	UPM	DWNM
Zoom 1 Pos	Byte	1	0-255							
Focus 1 Pos	Byte	1	0-255							
Zoom 2 Pos	Byte	1	0-255							
Focus 2 Pos	Byte	1	0-255							

2.3.7 Move To Entered Pan/Tilt/Zoom/Focus with max speed limit

This command is used to move the platform to a specific set of manually entered pan, tilt, zoom and focus coordinates. The coordinate for pan and tilt must consist of the desired position to 1/100th degree multiplied by 100, i.e., +90.00° should be sent as 9000. The user can force an axis to remain in position by sending its current position back to the PTCR. However, the coordinate value 99999 (+999.99°) can also be sent in order to prohibit movement of a specific axis. For example, if the user wishes to only move PAN, send 99999 (+999.99°) as the "Move To" coordinate for TILT and the TILT axis will remain stationary. The zoom and focus coordinates should be sent as a value between 0 and 255. Reference command 33H for range checking performed.

Connected lens systems may not be able to move across the full range of coordinates. If a zoom or focus position is entered that cannot be attained by the peripheral the unit will be moved as close as possible, then a timeout will allow the command to complete without generating an error.

- Command:

Data	Format	Bytes	
Command	3BH	1	
PAN Coord	Int	3	PAN = -360000 to +360000 = -360.000° to +360.000° or 999999 for no move

TILT Coord	Int	3	TILT = -180000 to +180000 = -180.000° to +180.000° or 999999 for no move
Zoom 1 Pos	Byte	1	1-255 (01H-FFH) or 0 to hold position
Focus 1 Pos	Byte	1	1-255 (01H-FFH) or 0 to hold position
Iris 1 Pos	Byte	1	1-255 (01H-FFH) or 0 to hold position
Zoom 2 Pos	Byte	1	1-255 (01H-FFH) or 0 to hold position
Focus 2 Pos	Byte	1	1-255 (01H-FFH) or 0 to hold position
Iris 2 Pos	Byte	1	1-255 (01H-FFH) or 0 to hold position
Pan Max Speed	Int	2	Pan Speed in Deg/S (0.00 to +/- 320.00)
Tilt Max Speed	Int	2	Tilt Speed in Deg/S (0.00 to +/- 320.00)

The Pan and Tilt Max Speed entered here will be valid only for the move. Once move is over, these speeds are reset back to configured speed (or default speed if not configured).

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	3BH	1								
PAN Coord	Int	3	PAN = -360000 to +360000 = -360.000° to +360.000°							
TILT Coord	Int	3	TILT = -180000 to +180000 = -180.000° to +180.000°							
PAN Status	Bitset	1	CWSL	CCWSL	CWHL	CCWHL	TO	DE	0	0
TILT Status	Bitset	1	USL	DSL	UHL	DHL	TO	DE	0	0
Gen Status	Bitset	1	CON	EXEC	1	OSLR	CWM	CCWM	UPM	DWNM
Zoom 1 Pos	Byte	1	0-255							
Focus 1 Pos	Byte	1	0-255							
Iris 1 Pos	Byte	1	0-255							
Zoom 2 Pos	Byte	1	0-255							
Focus 2 Pos	Byte	1	0-255							
Iris 2 Pos	Byte	1	0-255							

2.3.8 Move Motor:

Even though the unit may have resolution of 0.01 deg, depending on stepper motor, driver micro-steps and gear ratio, pay load can move with much higher resolution. This command is intended to move motor directly for fine tuning purpose on long range camera. **(For Oracle, it is 256 microsteps per full step of motor)**

- Command:

Data	Format	Bytes								
Command	3DH	1								
Cmd	Bitset	1	PDIR[1]	TDIR[2]	0	0	0	0	0	0
PAN Motor	Byte	1	No of Steps (Maximum 255)							
TILT Motor	Byte	1	No of Steps (Maximum 255)							

[1] 1 = CW/0 = CCW, 0 Steps = No Movement
[2] 1 = UP/0 = DWN, 0 Steps = No Movement
Unit will always move with 32 micro-step (This may change later on). So with 20:1 gear ratio (like Mercury), 32 micro steps and 1.8 deg motor, one step will move motor approximately 0.0028 deg.

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	3DH	1								
PAN Coord	Int	3	PAN = -360000 to +360000 = -360.000° to +360.000°							
TILT Coord	Int	3	TILT = -180000 to +180000 = -180.000° to +180.000°							
PAN Status	Bitset	1	CWSL	CCWSL	CWHL	CCWHL	TO	DE	0	0
TILT Status	Bitset	1	USL	DSL	UHL	DHL	TO	DE	0	0
Gen Status	Bitset	1	CON	EXEC	1	OSLR	CWM	CCWM	UPM	DWNM

2.4 The Preset Table:

The PTCR can retain up to 32 (0-31) preset positions in non-volatile memory that are frequently used by the operator. The user can store, retrieve, or move to these coordinates by modifying and using the preset table as outlined below. Preset zoom and focus positions can also be stored.

A special preset, referred to as "Home" position can be directly driven to without referencing a preset number using command 36H. It should be saved at preset position 31. This position can also be assigned as a generic preset and can be included in a preset tour.

Any command to store or retrieve a preset entry will echo back the preset's coordinates. If a preset number greater than 31 is entered the preset number will be echoed as FFH and the remainder of the data will be 0's.

2.4.1 "Retrieve Preset Table Entry" Command:

The operator may retrieve the stored coordinate position and zoom/focus byte values for any preset.

- Command:

Data	Format	Bytes	
Command	40H	1	
Preset Num	xxH	1	0-254 (0-FEH)

- Response:

Data	Format	Bytes	
Command	40H	1	
Preset Num	xxH	1	0-254 (0-FEH) or FFH if preset out of range
Preset Pan	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00°
Preset Tilt	Int	3	TILT = -18000 to +18000 = -180.00° to +180.00°
Zoom 1	xxH	1	0-255 (00H-FFH)
Focus 1	xxH	1	0-255 (00H-FFH)
Zoom 2	xxH	1	0-255 (00H-FFH)
Focus 2	xxH	1	0-255 (00H-FFH)

2.4.2 "Save Coordinates As Preset Table Entry" Command:

This command allows the user to load a specific set of coordinates to the preset table. The PTCR will perform a range check for the input coordinates and abort saving if a coordinate is out of range. Allowable pan range is defined as -180.00° + pan angle offset to +180.00° + pan angle offset. Allowable tilt range is -90.00° + tilt angle offset to +90.00° + tilt angle offset. For an angle offset of 0°/0°, the range would be -180.00°/+180.00° and -90.00°/+90.00°. The user may save coordinates that exceed both soft and hard limits. Zoom and focus settings cannot be arbitrarily entered and should be determined by actually zooming and focusing the cameras using jog. Commands 42H and 43H below will allow storage of zoom and focus settings.

- Command:

Data	Format	Bytes	
Command	41H	1	
Preset Num	xxH	1	0-254 (0-FEH)
Preset Pan	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00°
Preset Tilt	Int	3	TILT = -18000 to +18000 = -180.00° to +180.00°

- Response:

Data	Format	Bytes	
Command	41H	1	
Preset Num	xxH	1	0-254 (0-FEH) or FFH if preset out of range
Preset Pan	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00°

Preset Tilt	Int	3	TILT = -18000 to +18000 = -180.00° to +180.00°
Zoom 1	xxH	1	0-255 (00H-FFH)
Focus 1	xxH	1	0-255 (00H-FFH)
Zoom 2	xxH	1	0-255 (00H-FFH)
Focus 2	xxH	1	0-255 (00H-FFH)

2.4.3 "Save Current Position As Preset Table Entry" Command:

This command allows the user to store the platform's current position and zoom/focus settings as a preset table entry.

- Command:

Data	Format	Bytes	
Command	42H	1	
Preset Num	xxH	1	0-254 (0-FEH)

- Response:

Data	Format	Bytes	
Command	42H	1	
Preset Num	xxH	1	0-254 (0-FEH) or FFH if preset out of range
Preset Pan	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00°
Preset Tilt	Int	3	TILT = -18000 to +18000 = -180.00° to +180.00°
Zoom 1	xxH	1	0-255 (00H-FFH)
Focus 1	xxH	1	0-255 (00H-FFH)
Zoom 2	xxH	1	0-255 (00H-FFH)
Focus 2	xxH	1	0-255 (00H-FFH)

2.4.4 "Save Current Zoom/Focus Positions To Preset Table Entry" Command:

This command allows the user to append the platform's current zoom and focus positions to an existing preset table entry. This allows the user to set a pan & tilt preset, manually jog the lenses until zoom and focus is set, then add the valid zoom and focus positions to the preset.

- Command:

Data	Format	Bytes	
Command	43H	1	
Preset Num	xxH	1	0-254 (0-FEH)

- Response:

Data	Format	Bytes	
Command	42H	1	
Preset Num	xxH	1	0-254 (0-FEH) or FFH if preset out of range
Preset Pan	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00°
Preset Tilt	Int	3	TILT = -18000 to +18000 = -180.00° to +180.00°
Zoom 1	xxH	1	0-255 (00H-FFH)
Focus 1	xxH	1	0-255 (00H-FFH)
Zoom 2	xxH	1	0-255 (00H-FFH)
Focus 2	xxH	1	0-255 (00H-FFH)

2.5 The Preset Tour:

The PTCR can hold three 63-step (0-62) preset tours. Preset tours are built only from assigned presets and allow the pan and tilt unit to sequentially move to a preset in the tour, wait a defined period of time, move to the next preset in the tour, wait a defined period of time, etc. Optionally, the user may also switch video signals when the preset position is reached (see the CV1 and CV2 bits) and blank the returned video during execution of a step (see the VBL bit.) Tours will be continuously executed until a "STOP" or jog command is received or a command other than a status query is received. **If a fault occurs or a soft or hard limit is reached the tour will stop executing.**

Tours are built by first flushing the existing tour. This will reset the tour pointer to 0. The user then sequentially adds preset numbers (0-31) and wait times (0-99 secs) to each stop in the tour. Once built, the tour can be started using the "Start Preset Tour" command, selecting which tour to execute.

The user may query both the number of steps in a tour and an actual entry's values using the "Get Tour Size" and "Query Preset Tour" commands. The user also has the capability to edit the preset tour using the "Append To Preset Tour", "Insert Into Preset Tour", "Delete From Preset Tour", and "Replace In Preset Tour" commands. The first command simply adds a preset to the end of the tour. The second allows the user to insert a preset into the tour while retaining the presets that follow. The third allows the user to remove a preset while retaining the presets that follow. In both cases, the presets that follow will be shifted up or down as required to keep the tour complete. The fourth command allows the user to replace a tour entry without disturbing the remaining tour entries. Note that the special "Home" preset 31 can be included in a tour.

Note that, as each step begins execution, the PTCR will return a response similar to the standard "Get Status/Jog" response but with the DES bit set, indicating that the coordinates returned are the actual destination for the move.

During random tour unit will randomly select path to go among specified paths. *(In Theory there is no random, it is all math so if somebody figure out the math at last, don't blame it on me.*)

Tours 0,1 and 2 are regular preset build tours corrosponds to legacy Quickset products and have 64 steps in each tour. Tour 3 is auto scan and it is two position tour to scan between two specified preset points.

2.5.1 "Start Preset Tour" Command:

This command allows the user to execute any one of the three preset tours. If the tour is empty FFH will be returned. See each type below for further tour information.

- Command:

Data	Format	Bytes	
Command	37H	1	
Tour Num	xxH	1	0-2, 3,4 and 5

Tours 0,1 and 2 are regular preset tours same as legacy PTCR96 system.

Tour 3 is Auto Scan between two presets. It uses only firt two steps of tour 3 and runs between them.

Tour 4 is random preset tour runs among specified tour steps programmed in tour 4 but selection is random.

Tour 5 is preset scan starting preset 0. Dwell time is fixed at 1 second.

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	37H	1								
Tour Num	xxH	1	0-2, 3,4 and 5 or FFH if selected tour is empty							

2.5.2 "Flush Preset Tour" Command:

The operator may completely clear a tour and ready it for building by using this command. This command should also be used to clear a tour that is found to be corrupt.

- Command:

Data	Format	Bytes	
Command	50H	1	
Tour Num	xxH	1	0-2, 3 and 4.

- Response:

Data	Format	Bytes	
Command	50H	1	
Tour Num	xxH	1	0-2, 3 and 4.

2.5.3 "Query Preset Tour" Command:

The operator may examine the sequential steps of a tour by using this command. If the response returns FFH for the step number, the requested step does not exist in the tour.

- Command:

Data	Format	Bytes	
Command	51H	1	
Tour Num	xxH	1	0-2, 3 and 4.
Step Num	xxH	1	0-62 (0-3EH)

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	51H	1								
Tour Num	xxH	1	0-2, 3 and 4.							
Step Num	xxH	1	0-62 (0-3EH) or FFH for all if the step does not exist							
Preset Num	xxH	1	CV2[1]	CV1[1]	0-31 (0-1FH)					
Wait Time	xxH	1	VBL[2]	0-99 (0-FFH) seconds						

[1]0 = No Automatic Camera Switching, CV1 Set = Switch to Cam1 Video, CV2 Set = Switch to Cam2 Video
[2]0 = No Video Blanking Between Moves, 1 = Video Blanking Between Moves

2.5.4 "Append To Preset Tour" Command:

The operator may append a preset to the tour by using this command. The step number returned by the response represents the tour position where the preset was saved. If FFH is returned the tour is full and the preset was not accepted or the tour is corrupt.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	52H	1								
Tour Num	xxH	1	0-2,3 and 4							
Preset Num	xxH	1	CV2	CV1	0-31 (0-1FH)					
Wait Time	xxH	1	VBL	0-99 (0-FFH) seconds						

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	52H	1								
Tour Num	xxH	1	0-2, 3 and 4							
Step Num	xxH	1	0-62 (0-3EH) or FFH for all if the current tour is full							
Preset Num	xxH	1	CV2	CV1	0-31 (0-1FH)					
Wait Time	xxH	1	VBL	0-99 (0-FFH) seconds						

2.5.5 "Insert Into Preset Tour" Command:

The operator may insert a preset into the tour by using this command. Any steps that follow the insertion will be moved out one step. If a step number of FFH is returned the tour is full and the preset was not accepted. If the step number returned is less than the step number sent the step did not originally exist and the new entry was appended to the tour.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	53H	1								
Tour Num	xxH	1	0-2, 3 and 4							
Step Num	xxH	1	0-62 (0-3EH)							
Preset Num	xxH	1	CV2	CV1	0-31 (0-1FH)					
Wait Time	xxH	1	VBL	0-99 (0-FFH) seconds						

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	53H	1								
Tour Num	xxH	1	0-2, 3 and 4.							
Step Num	xxH	1	0-62 (0-3EH) or FFH for all if the current tour is full							
Preset Num	xxH	1	CV2	CV1	0-31 (0-1FH)					
Wait Time	xxH	1	VBL	0-99 (0-FFH) seconds						

2.5.6 "Delete From Preset Tour" Command:

The operator may delete a preset in the tour by using this command. Any steps that follow the deletion will be moved back one step. If a step number of FFH is returned the tour did not contain the step to be deleted.

- Command:

Data	Format	Bytes	
Command	54H	1	
Tour Num	xxH	1	0-2,3 and 4.
Step Num	xxH	1	0-62 (0-3EH)

- Response:

Data	Format	Bytes	
Command	54H	1	
Tour Num	xxH	1	0-2, 3 and 4.
Step Num	xxH	1	0-62 (0-3EH) or FFH for all if the step does not exist

2.5.7 "Replace In Preset Tour" Command:

The operator may replace a preset in the tour by using this command. All other existing steps will be unaltered. If a step number of FFH is returned the tour did not contain the step to be replaced.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	55H	1								
Tour Num	xxH	1	0-2, 3 and 4.							
Step Num	xxH	1	0-62 (0-3EH)							
Preset Num	xxH	1	CV2	CV1	0-31 (0-1FH)					
Wait Time	xxH	1	VBL	0-99 (0-FFH) seconds						

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	55H	1								

Tour Num	xxH	1	0-2,3 and 4.		
Step Num	xxH	1	0-62 (0-3EH) or FFH for all if the step does not exist		
Preset Num	xxH	1	CV2	CV1	0-31 (0-1FH)
Wait Time	xxH	1	VBL	0-99 (0-FFH) seconds	

2.5.8 "Get Tour Size" Command:

The operator can query a tour to find out the number of steps stored. If the tour is corrupt this procedure will return FFH or 255. This is a good method of determining if the tour is corrupt before attempting to modify it.

- Command:

Data	Format	Bytes	
Command	56H	1	
Tour Num	xxH	1	0-2, 3 and 4

- Response:

Data	Format	Bytes	
Command	56H	1	
Tour Num	xxH	1	0-2, 3 and 4.
Step Count	xxH	1	0-62 (0-3EH) or FFH if the tour is corrupt

2.5.9 "Get / Set PTZF Tolerance for Tour" Command: (Factory use only)

Some time operating conditions does not allow PT/Camera to reach destination positions as specified in tour definition. For ex. hysteresis in zoom / focus lens position, windy condition requires higher stop band in belt driver unit to prevent oscillation around destination.

When such things happen, tour function waits for the step to complete (i.e to reach destination). This function allows user to adjust tolerances for each required positions to overcome such operating hurdles and run smooth tour.

- Command:

Data	Format	Bytes		
Command	5AH	1		
Pan Tol.	xxH	1	Q	0-127 (0.00 to 1.27 deg)
Tilt Tol.	xxH	1	0-127 (0.00 to 1.27 deg)	
Zoom 1 Tol.	xxH	1	0-99	
Focus 1 Tol.	xxH	1	0-99	
Iris 1 Tol.	xxH	1	0-99	
Zoom 2 Tol.	xxH	1	0-99	
Focus 2 Tol.	xxH	1	0-99	
Iris 2 Tol.	xxH	1	0-99	

- Response:

Data	Format	Bytes		
Command	5AH	1		
Pan Tol.	xxH	1	0	0-127 (0.00 to 1.27 deg)
Tilt Tol.	xxH	1	0-127 (0.00 to 1.27 deg)	
Zoom 1 Tol.	xxH	1	0-99	
Focus 1 Tol.	xxH	1	0-99	
Iris 1 Tol.	xxH	1	0-99	
Zoom 2 Tol.	xxH	1	0-99	
Focus 2 Tol.	xxH	1	0-99	
Iris 2 Tol.	xxH	1	0-99	

2.6 Camera/Lens Parameters and Control

The PTCR will act as a transparent conduit for camera commands. Any command destined for the camera should be completely built including any control characters and checksums. It should then be placed inside a pan & tilt command wrapper (see command 62H.) Up to 80 bytes may be sent in one string to the camera. The PTCR will be made aware of the camera's required serial parameters by the setting of its own non-volatile parameters (see command 60H.) The PTCR will use this data to configure a transceiver and UART and send the string to the camera.

The PTCR-95 provides communications with up to 2 cameras. However, this cannot be done simultaneously. The user must select the camera prior to sending a command string to it (see command 62H.) The user may also control camera input power and select which camera returns video output via command 63H.

Normally, the PTCR will respond to a command 31H status query with standard status data and camera byte counts of 0. When a serial byte is received from a camera a timer will be started and the PTCR will wait for additional bytes to arrive (see command 65H.) As each byte is received it will be placed in a queue and the timer will be restarted. Once the timer expires the PTCR assumes the camera response is complete. The PTCR will tack the byte count and returned camera data onto the end of the next status response. Therefore, if a camera byte count greater than 0 is received the application should remove the camera bytes and use them as required.

Though a bit convoluted, this allows the PTCR to be transparent, negating the need for code changes to accommodate different camera types. This will also allow cameras that only operate with allow dedicated RS-232 or RS-422 links to participate in an RS-485 party line environment. However, the onus of building proper camera commands and parsing received data resides with the programmer.

2.6.1 "Get/Set Camera Comm Parameters" Command:

These parameters define the serial communications parameters and levels for the attached cameras. To determine proper settings for these values reference specific camera data. These parameters should be checked first if a camera is not responding to commands. Sending "0" as the baud rate will disable use of the camera serial port. Setting the Query bit will instruct the PTCR to return the current values without changing them. The secondary microcontrollers will be loaded with the new values once the parameters are stored. **These parameters should be properly configured before physically connecting the cameras for the first time or changing camera serial port levels.**

Currently, **only 8 bits, no parity may be set and any other setting will be ignored.** However, hooks have been left in to accommodate different formats in the future. The transmitted values will be ignored when simply querying the PTCR. Therefore, they do not need to be included.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	60H	1								
Camera 1	xxH	1	Query	LVL[1]		0-3 Parity/Bits[2]		0-7 baud[3]		
Camera 2	xxH	1	0	LVL[1]		0-3 Parity/Bits[2]		0-7 baud[3]		

[1]Camera 1/2 --> 00 = RS-232/ 01 = RS-422 / 10 = RS-485 4 W (for future) / 11 = RS-485 2W (for future)
[2]0 = 8/N, 1 = 7/N, 2 = 7/E, 3 = 7/O (Only 8/ None currently supported)
[3]0 = Disable, 1 = 9.6, 2 = 14.4, 3 = 19.2, 4 = 28.8, 5 = 38.4, 6 = 57.6, 7 = 115.2

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	60H	1								
Camera 1	xxH	1	0	LVL		0-3 Parity/Bits		0-7 baud		
Camera 2	xxH	1	0	LVL		0-3 Parity/Bits		0-7 baud		

2.6.2 "Get/Set Lens Parameters" Command:

Proper setting of these values allow the slave microcontrollers to know how the zoom and focus motors should be powered, if they are installed, and if the pot reading needs to be inverted. Setting the correct minimum speed value will keep the zoom and focus motors from stalling. Maximum speed is always considered the full speed available

from the motor driver. This parameter should be checked first if the zoom and focus functions are not working properly.

Setting the Query bit will instruct the PTCR to return the current value without changing it. Clearing the Query bit will actually load the new value into the EEPROM. The secondary microcontrollers will be loaded with the new values once the parameters are stored.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	61H	1								
Lens 1	Bitset	1	Query	[1]ZE	[2]ZR	[3]ZI	[4]FE	[5]FR	[6]FI	0
Lens 1 Z Min	xxH	1	0-255 (00H-FFH) minimum zoom speed							
Lens 1 F Min	xxH	1	0-255 (00H-FFH) minimum focus speed							
Lens 2	Bitset	1	0	ZE	ZR	ZI	FE	FR	FI	0
Lens 2 Z Min	xxH	1	0-255 (00H-FFH) minimum zoom speed							
Lens 2 F Min	xxH	1	0-255 (00H-FFH) minimum focus speed							

[1]1 = Enable the Zoom Function
[2]1 = Reverse the Zoom Motor's Normal Operation
[3]1 = Invert the Zoom Resolver Reading
[4]1 = Enable the Focus Motor
[5]1 = Reverse the Focus Motor's Normal Operation
[6]1 = Invert the Focus Resolver Reading

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	61H	1								
Lens 1	Bitset	1	0	ZE	ZR	ZI	FE	FR	FI	0
Lens 1 Z Min	xxH	1	0-255 (00H-FFH) minimum zoom speed							
Lens 1 F Min	xxH	1	0-255 (00H-FFH) minimum focus speed							
Lens 2	Bitset	1	0	ZE	ZR	ZI	FE	FR	FI	0
Lens 2 Z Min	xxH	1	0-255 (00H-FFH) minimum zoom speed							
Lens 2 F Min	xxH	1	0-255 (00H-FFH) minimum focus speed							

2.6.3 "Command Camera" Command:

This command acts as a transfer wrapper around a camera-specific command. In order to use this command the user should build an entire camera command string including any extra bytes such as control and framing characters and checksums the camera needs for communications. The PTCR-required STX, identity and command number should be prepended to the front and the ETX and LRC should be appended to the end. If the standard method of transmitting other commands is used (ESC insertion) any conflicting values in the camera string will be automatically converted to "safe" values for transmission to the PTCR.

The main processor of the PTCR will strip the camera command string out of this command, return any altered control character values to their original value, and transfer the string to the slave processor. The slave processor will then transfer it to the camera via the serial port. Note that the camera command string can be variable length **but cannot exceed 80 bytes**.

The PTCR will respond to this command, indicating that it was received correctly. Any data the camera has to return will be sent through the "Get Status/Jog" 31H command. Reference it for more information on receiving returned data from the cameras. Normally, camera communications is half duplex, i.e., the user should wait for a response before sending another camera command. Otherwise, a camera command string in the microcontroller buffer may be overwritten.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	62H	1								

Cam/Count	xxH	1	[1]CN	1 – 80 bytes to follow
Camera Cmd	xxH	1-80	Complete Camera Command String	

[1]0 = Camera 1/1 = Camera 2

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	62H	1								

Extended Part

- *Extended Command:*

Data	*Format*	*Bytes*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Command	*62H*	*1*								
Extend Cmd	*FFH*	*1*	*Fix*							
Board Addr.	*00H - 0FH*	*1*	*Destination board address in Oracle board set*							
Channel No.	*xxH*	*1*	*[1] Pkt Type*	*Bits 6:0 --> Channel number 0 to 7 (8 Max)*						
Pkt Length	*xxH*	*1*	*[2] Next Pkt*	*Bits 6:0 --> Packet length (Max 45)*						
Camera Cmd	*xxH*	*1-45*	*Complete Camera Command String*							

[1] Pkt Type --> 0 = Send Packet to Camera, 1 = Get Response from Camera
[2] Next Pkt --> 0 = No next packet, 1= Next packet is in raw.

- Extended Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	62H	1								
Extend Cmd	FFH	1	Fix							
Board Addr.	00H - 0FH	1	Destination board address in Oracle board set							
Channel No.	xxH	1	[1] Data Type	Bits 6:0 --> Channel number 0 to 7 (8 Max)						
Pkt Length	xxH	1	[2] Next Pkt	Bits 6:0 --> Packet length (Max 45)						
Camera Cmd	xxH	1-45	Complete Camera Command String							

[1] Data Type --> 0 = No Data attached to packet, 1 = Data attached to packet.
[2] Next Pkt --> 0 = No next packet, 1= Next packet is in raw.
Note: To Get Board ID and Channel number, read Oracle manual.

2.6.4 "Get / Set Video Switching" Command: (63H)

This command is reserved for future as currently we do not have any video switching on Oracle Board set.

2.6.5 "Get/Set Camera Select/Power/Video" Command:

Note: This command works only for two payload.
This command performs configuration of the camera video, turns on/off video and brings the OSD online and offline. There is one video per payload with OSD capability that can also be independently blanked. The following table shows the mode, OSD control, and blanking configurations possible with the onboard video selector.

The PTCR maintains two configuration states, initialize and operating. The initialize state will be the default configuration used when the unit is first powered on. The operating state may then be changed as required without impacting the initial state. Setting the Query bit with the STOR bit clear will return the current operating configuration. Setting the Query bit with the STOR bit set will return the stored default configuration. If Query is clear and the STOR bit is set the configuration byte will be stored in EEPROM and will be the new default configuration at power-up. Clearing both the Query bit and the STOR bit will change the current configuration but the default configuration will not change.

Mode	OSD	Payload 1 (VO1)	Payload 2 (VO2)	Payload 1 BLK (A)	Payload 2 BLK (B)
0	0	Video No OSD	Video No OSD	0	0
0	1	Video No OSD	Video No OSD	0	0
1	0	Video No OSD	Video No OSD	0	0
1	1	Video OSD	Video No OSD	0	0
2	0	Video No OSD	Video No OSD	0	0
2	1	Video No OSD	Video OSD	0	0
3	0	Video No OSD		0	0
3	1	Video OSD	Video OSD	0	0
X	X	No Video	X	1	X
X	X	X	No Video	X	1

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	64H	1								
Data	xxH	1	Query	STOR	[1]PAY LOAD1 BLK	[2]PAY LOAD2 BLK	[3]PAL / NTSC	Mode 0-3		[4]OSD Both Payload
Data	xxH	1	0	0	0	0	[5]Pay Load2 CAM PWR	[5]Pay Load1 CAM PWR	0	1

[5]1 = CAM PWR ON, 0 = CAM PWR OFF
[1]1 = PayLoad 1 Video Select 0 = Pay Load 1 Video Blanked
[2]1 = PayLoad 2 Video Select 0 = Pay Load 2 Video Blanked
[3]1 = Payload Camera is PAL, 0 = Payload Camera is NTSC
[4] 1 = Payload Main OSD On, 0 = Payload Main OSD off.

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	64H	1								
Data	xxH	1	Query	STOR	[1]PAY LOAD1 BLK	[2]PAY LOAD2 BLK	[3]PAL Payload 1	Mode 0-3		[4]OSD Payload 1
Data	xxH	1	0	0	[7]Pay Load2 PAL	0	[5]Pay Load2 CAM PWR	[5]Pay Load1 CAM PWR	0	0

2.6.6 "Get/Set Camera Response Timeout" Command:

As stated above, once a byte of data is received from the camera, the slave microcontroller will wait for a predefined time for additional bytes to arrive before flagging the main processor to gather and return them. The timer will be restarted every time an additional byte arrives. This method will reduce the number of partial string returns and parsing repeats for the host. However, different cameras will require differing amounts of time to complete their responses. This command allows adjusting the timeout/transfer data timer in multiples of 9ms for the 2 cameras. The extended command is more suited for Oracle Architecture.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	65H	1								
Timeout 1	xxH	1	Query	1 – 100 (* 9ms)						
Timeout 2	xxH	1	0	1 – 100 (* 9ms)						

Timeout 1 is for Payload 1 Camera 1 and Timeout 2 is for Payload 2 camera 1.

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	65H	1								
Timeout 1	xxH	1	0	1 – 100 (* 9ms)						
Timeout 2	xxH	1	0	1 – 100 (* 9ms)						

2.6.7 "Get/Set Aux Control Outputs" Command

Each camera interface connector provides two auxiliary control outputs. These may be used to activate wipers, lens washers, lights, etc. Each line provides a voltage set at the input voltage of the PTCR-95 and a switched return path. Maximum loading for these outputs is dictated by the capacity of the input power supply but should be restricted to 1.5 Amps each or less. The auxiliary lines are toggle on/toggle off and will remain in the mode set by the last 66H command. The lines will be off by default at power-up. Setting the Query bit will instruct the PTCR to return the current value without changing it. Clearing the Query bit will load the new value into the unit and activate the lines accordingly.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	66H	1								
Aux Bits	xxH	1	Query	0	0	0	AUX22	AUX21	AUX12	AUX11

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	66H	1								
Aux Bits	xxH	1	0	0	0	0	AUX22	AUX21	AUX12	AUX11

[1]1 = Output On/0 = Output Off for AUX (Camera Number)(Output Number)

2.7 OSD related Parameters:

The PTCR 1000 can provide on-board OSD through the Pay Load board. Command 64H turns on/ off OSD feature of the Pay Load board to connected camera video. The following commands can be used to further configure the ON screen display. OSD information is divided in to two type, Startup display that stays for 10 second after power up (if turned on) and continuous display that stays on after startup display till the unit is powered.

2.7.1 "Get/Set OSD Angle Format" Command: (Continuous OSD only). Standard PTCR 96 command.

A standard position display for the OSD is provided in the firmware. The format is (AZ: +nnn.nn EL:+nnn.nn This command allows the user to turn the display of this specific line on and off without altering an OSD-enabled video mode. The user may also select the line for display, 0-11. The SIGN bit determines if the angular display is signed or unsigned. A signed display shows -179.99° to +180.00° in pan and -90.00° to +90.00° in tilt. If the SIGN bit is clear the display will show +0.00° to +359.99° in pan and +0.00° to +180.00° in tilt. Setting the Query bit will instruct the PTCR to return the current values without changing them. (The display will change between 0.00/0.000 resolution depending on selected system resolution.)

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	70H	1								
Channel	xxH	1	Query	STOR	[1]SIGN	[2]DIS	Line Number 0-11			

[1]SIGN 0 --> unsigned, 1--> Signed
[2]DIS 0 --> OFF, 1--> ON for this line

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	70H	1								
Cmd Bits	xxH	1	Query	STOR	[1]SIGN	[2]DIS	Line Number 0-11			

2.7.2 "Get/Set OSD Site Name / Number format" Command: (Continuous)

A site number (unit identity) and name display for the platform is provided in the firmware. The basic format is UNIT nnn sssssssssss with "sss…" denoting a user entered string of up to 12 characters. This command allows the user to turn the display of this specific line on and off without altering an OSD enabled video mode. The user may also select the line for display, 0-11. Setting the Query bit will instruct the PTCR to return the current values without changing them. The 12 character string can be built using command 75H. This OSD line is configurable between startup and continuous display.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	71H	1								
Cmd Bits	xxH	1	Query	STOR		[1]DIS	Line 0-11			

[1]DIS --> 0 = Site Number and Name Display Off/ 1 = Site Number and Name Angle Display On
Same settings will be applied to both payload boards. Future command will provide separate controls.

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	71H	1								
Cmd Bits	xxH	1	Query	STOR	[2]TYPE	[1]DIS	Line 0-11			

PTCR will read settings from Payload board 1 and send that data as response.

2.7.3 "Get/Set OSD Name String" Command:

The OSD site name/number string displays the unit ID/address for the platform but, optionally, can also display a user defined name of up to 12 characters. When this command is executed the name string will be loaded into both the active display and the non-volatile memory for recall at initial power on. The OSD is preloaded with a font set of the standard ASCII characters from character 20H(32) to 7FH(127). This set includes all of the "printable"

characters, i.e., punctuation, numbers, and upper and lower case letters. Consult the ASCII table below for the conversion from hex value to character. Note that ALL 12 characters must be passed and spaces (20H) can be used for padding. This string is stored on the main board.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	75H	1								
Char 1	xxH	1	Query	ASCII code (20H - 7FH)						
Char 2-12	xxH	11	0	ASCII code (20H - 7FH) (For Query don't send this bytes or it will be ignored)						

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	75H	1								
Char 1-12	xxH	12	0	ASCII Code (20H-7FH)						

2.7.4 "Get/Set Status OSD format" Command: (Continuous only)

User can turn on/ off unit status display over video. This helps with PELCO-D kind of unidirectional protocol control interface. Status OSD will display errors like Hard Limit Fault or Time Out etc.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	76H	1								
Cmd Bits	xxH	1	Query	STOR	0	[1]DIS	Line 0-11			
Channel	FxH	1	Fix part FH to announce for extended command				Channel number 0 ~ 7			

[1]DIS --> 0 = Status Display Off/ 1 = Status Display On
If channel byte is missing then it is considered for channel 0 (Pay load board 1)

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	76H	1								
Cmd Bits	xxH	1	Query	STOR	0	[1]DIS	Line 0-11			
Channel	FxH	1	Fix part FH to announce for extended command				Channel number 0 ~ 7			

2.7.5 "Get/Set SW revision number format" Command: (Start up or continuous display)

User can turn on/ off unit SW number and it's revision display over video.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	77H	1								
Cmd Bits	xxH	1	Query	STOR	[2]TYPE	[1]DIS	Line 0-11			
Channel	FxH	1	Fix part FH to announce for extended command				Channel number 0 ~ 7			

[1]DIS --> 0 = SW revision number Off/ 1 = SW revision number On
[2]TYPE --> 0 = Start up / 1 = Continuous display
If channel byte is missing then it is considered for channel 0 (Pay load board 1)

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	77H	1								
Cmd Bits	xxH	1	Query	STOR	[2]TYPE	[1]DIS	Line 0-11			
Channel	FxH	1	Fix part FH to announce for				Channel number 0 ~ 7			

			extended command	

2.7.6 "Get/Set Serial and Model number format" Command: (Start up or continuous display)

User can turn on/ off unit Serial number and Model number display over video. This display is available as configurable between continuous display or as start up display.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	78H	1								
Cmd Bits	xxH	1	Query	STOR	[2]TYPE	[1]DIS	Line 0-11			
Channel	FxH	1	Fix part FH to announce for extended command				Channel number 0 ~ 7			

[1]DIS --> 0 = Serial and Model number Off/ 1 = Serial and Model number On
[2]TYPE --> 0 = Start up / 1 = Continuous display
If channel byte is missing then it is considered for channel 0 (Pay load board 1)

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	78H	1								
Cmd Bits	xxH	1	Query	STOR	[2]TYPE	[1]DIS	Line 0-11			
Channel	FxH	1	Fix part FH to announce for extended command				Channel number 0 ~ 7			

2.8 PTCR Operating Parameters:

Most parameters may be queried from the PTCR. A reduced set of PTCR parameters may also be modified through the host interface.

2.8.1 Setting Pan & Tilt Angle Corrections:

The pan & tilt unit is internally calibrated to reflect an absolute relationship between the bottom mounting plate, the enclosure, and the tilt frame. However, the situation may arise where the user wishes to offset the degree display. This may be a result of mounting orientation or the desired method of measuring. To correct the reading, the user may provide an offset value for the displayed pan and tilt coordinates. Entry of a positive offset will simply increase the respective degree display. Negative will decrease the degree display. Internal orientation will not change, only the displayed angle.

Sometimes it is beneficial to correct coordinate display for the platform relative to your point of reference. This can be manually performed by calculating and entering pan and tilt angle offsets. However, the "Align to Center" and "Align to Coordinate" commands listed below can be used to allow the PTCR to perform these calculations for you.

Note that any change in pan and tilt offset will also modify the displayed position of presets, soft limits, etc. The relative angles will be correct, however. For example, assume a unit has a 0° tilt offset, the tilt frame is level at 0° and the preset will move it to -20°. Executing the preset move will move the tilt frame to -20°. If a +10° tilt offset is loaded a unit with a level tilt frame will display +10° and, after moving to the preset, -10° will be displayed. The unit has still moved 20° relative to center. Only the displayed angle has been altered to accommodate for the offset. **Therefore, it is recommended that any modification of pan & tilt angle offset be followed by a reloading of the preset table and soft limits if normally displayed in your application.**

2.8.2 "Get Pan & Tilt Angle Correction" Command:

- Command:

Data	Format	Bytes
Command	85H	1

- Response:

Data	Format	Bytes	
Command	85H	1	
Pan Offset	Int	3	PAN = -18000 to +18000 = -180.00° to +180.00°
Tilt Offset	Int	3	TILT = -9000 to +9000 = -90.00° to +90.00°

2.8.3 "Set Pan & Tilt Angle Correction" Command:

This command allows manual entry of angle correction. The PTCR will check the input range of both pan and tilt angle corrections and will not save invalid entries.

- Command:

Data	Format	Bytes	
Command	80H	1	
Pan Offset	Int	3	PAN = -18000 to +18000 = -180.00° to +180.00°
Tilt Offset	Int	3	TILT = -9000 to +9000 = -90.00° to +90.00°

- Response:

Data	Format	Bytes	
Command	80H	1	
Pan Offset	Int	3	PAN = -18000 to +18000 = -180.00° to +180.00°
Tilt Offset	Int	3	TILT = -9000 to +9000 = -90.00° to +90.00°

2.8.4 "Align To Center" Command:

"Align To Center" will automatically calculate the pan and tilt angle corrections required to realign the angular position display for the platform so that the current position is considered a center position displaying a pan and tilt angle of 0°. This command is useful if the user wishes to measure the relative angle between two objects. The user may jog to the first object then execute "Align To Center", changing the displayed angle to 0°/0°. Jogging to the next target will display the relative angle between the two objects.

- Command:

Data	Format	Bytes
Command	82H	1

- Response:

Data	Format	Bytes	
Command	82H	1	
Pan Offset	Int	3	PAN = -18000 to +18000 = -180.00° to +180.00°
Tilt Offset	Int	3	TILT = -9000 to +9000 = -90.00° to +90.00°

2.8.5 "Align To Coordinate" Command:

"Align to coordinate" allows entry of the desired position to display. For example, the platform is mounted on a southeast line, is jogged due east, and currently reads -45.0° in pan. The user wants this reading to be +90.00° reflecting a compass point. The user may calculate and manually enter an offset using Command 80H. Alternately, the user can simply enter +90.00° for pan using this command and the offset will be automatically calculated and stored.

- Command:

Data	Format	Bytes	
Command	83H	1	
PAN Coord	Int	3	PAN = -18000 to +18000 = -180.00° to +180.00°
TILT Coord	Int	3	TILT = -9000 to +9000 = -90.00° to +90.00°

- Response:

Data	Format	Bytes	
Command	83H	1	
Pan Offset	Int	3	PAN = -18000 to +18000 = -180.00° to +180.00°
Tilt Offset	Int	3	TILT = -9000 to +9000 = -90.00° to +90.00°

2.8.6 "Clear Angle Corrections" Command:

This command will clear any angular corrections to zero, realigning the platform angular display to the true 0°/0° position.

- Command:

Data	Format	Bytes
Command	84H	1

- Response:

Data	Format	Bytes	
Command	84H	1	
Pan Offset	Int	3	PAN = 0
Tilt Offset	Int	3	TILT = 0

2.8.7 "Get/Set Pan & Tilt Soft Limits" Command:

The PTCR can contain degree positions that, when exceeded, can stop platform travel. These values are referred to as software or soft limits. Soft limits act as redundant safety stops in addition to the hard limit switches. Soft limits are normally set just inside the hard limits, making the soft limit the primary stop and the hard limit the redundant stop.

Though it would be possible to allow setting the soft limits by entering coordinates through the remote interface this feature has not been included in the interests of safety. It is critical that the platform be observed while soft limits are being set in order to avoid collisions. If the user can physically jog to the soft limit point without hitting anything, it is a safe limit. Therefore, soft limits can only be set using a "move to and assign" method.

The user should jog the platform to the desired limit position, then send the command with the appropriate axis identified in order to set the soft limit. The user may override any existing soft limit by setting the OSL (Override Soft Limit) bit in the jog command. **This bit should only be used to assist in establishing soft limits.** The returned OSLR will show when this bit is set. Continuous rotation units will not observe soft limit settings and the values returned may be disregarded. Presence of the ENC bit in the "Get Status/Jog" response will cue the user that the attached unit is continuous rotation. Setting the Query bit will return the current value for an axis. Clearing the Query bit will actually store the position for the axis.

Note that any change in pan and tilt offset will also modify the displayed position of presets, soft limits, etc. The relative angles will be correct, however. Therefore, it is recommended that any modification of pan & tilt angle offset be followed by a reloading of the preset table and soft limits if normally displayed in your application.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	81H	1								
Axis Number	xxH	1	Query	0 = CW, 1 = CCW, 2 = Up, 3 = Down						

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	81H	1								
Axis Number	xxH	1	0	0 = CW, 1 = CCW, 2 = Up, 3 = Down						
Soft Limit	Int	3	PAN = -36000 to +36000, TILT = -18000 to +18000							

2.8.8 "Get all Pan & Tilt Soft Limits" Command:

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	86H	1								

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	86H	1								
Soft Limit	Int	3	PAN CW = -36000 to +36000							
Soft Limit	Int	3	PAN CCW = -36000 to +36000							
Soft Limit	Int	3	TILT UP = -18000 to +18000							
Soft Limit	Int	3	TILT DOWN = -18000 to +18000							

2.9 Get/Set Pan and Tilt Encoder corrections: (Factory Only)

PTZ uses encoders or potentiometers for position feedback. They may have some nonlinearity and inaccuracies. This command will allow configuring PTZ system for errors observed during factory calibration of encoder against known standard. To use these data, PTZ must have algorithm implemented for inaccuracy correction using these data.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	8AH	1								
data	xxH	1	Query							[1]P/T
Data	xxH	2	[2]Index number between o and 359							

data	xxH	1	+/- 128 correction counts

-[1]P/T = 0 means for PAN and 1 means for TILT.
-[2]Index is currently in 0-359 and represents measurement points at whole degree of standard reference.
-correction counts represents number between -127 to +127, that may represent 100th of degree or 1000th degree depending on resolution of encoder.

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	8AH	1								
data	xxH	1	Query							[1]P/T
data	xxH	2	Index							
data	xxH	1	+/- 128 counts							

-[1]P/T = 0 means for PAN and 1 means for TILT.
-Index is currently in 0-359 and represents measurement points at whole degree of standard reference.
-correction counts represents number between -127 to +127, that may represent 100th of degree or 1000th degree depending on resolution of encoder.

2.10 PTCR Initial Setup Parameters:

2.10.1 Getting/Setting Pan & Tilt Center Position:

This pan & tilt unit uses incremental encoders to track position. The 9000 line encoders are further decoded to detect 36000 transitions or steps in one rotation. Since the encoders are mounted directly to the pivot points of the platform, one rotation of the platform equals one rotation of the encoder. 36000 steps equal 360.00° of travel or 1 step equals 0.01°. The encoders do not return an absolute position but only indicate direction and steps. There is no absolute center for the encoder. We "align" the encoder to the platform by moving the platform to physical center in pan and tilt, then executing the "Set Pan & Tilt Center Position" command. This will adjust the encoder counts for both pan and tilt to 0. From this point forward any movement will result in an increment or decrement of the encoder count equal to the distance moved in hundredths of degrees.

Encoders should be configured with the index pulse ±10° from physical center but only to allow the platform to easily detect and realign its readings relative to the index pulse. An encoder will also illuminate the yellow pan and tilt center readings when the count is within 100 RU's of 0/0. Since an encoder is not an absolute resolver these will have little use during alignment but will indicate the point of alignment once a unit is properly setup.

The first command will allow the user to retrieve the current stored value for center for each axis in resolver units. This value only has significance for potentiometer-based systems. However, the command is retained for downward compatibility. For this encoder based system the value will always be 0. The second command will allow the user to command the PTCR to reset the encoder count to 0. The result will be re-aligned returned coordinates of 0°/0° (with companion 0°/0° angle offsets.)

2.10.2 "Get Pan & Tilt Potentiometer/Encoder Center Position" Command:

- Command:

Data	Format	Bytes
Command	90H	1

- Response:

Data	Format	Bytes	
Command	90H	1	
Pan Count	Int	3	PAN = -360000 to +360000 = -360.000° to +360.000°
Tilt Count	Int	3	TILT = -180000 to +180000 = -180.000° to +180.000°

PTCR1000 based system are different where index pulse is always at 0. Center is the offset from the index pulse. (Compared to PTCR95N and 96, where center is always 0 and index is offset from center).

2.10.3 "Set Pan & Tilt Potentiometer Center Position" Command:

- Command:

Data	Format	Bytes
Command	91H	1

- Response:

Data	Format	Bytes	
Command	91H	1	
Pan Count	Int	3	PAN = -360000 to +360000 = -360.000° to +360.000°
Tilt Count	Int	3	TILT = -180000 to +180000 = -180.000° to +180.000°

2.10.4 "Get/Set Pan & Tilt Ramp Parameters" Command:

These parameters determine the acceleration/deceleration rate and initial start/stop speeds for the pan and tilt axes. At the factory a minimum and a maximum step rate are set in the controller. The controller then calculates an array of interim speeds that create linear steps from minimum to maximum speed. One may approximate an array speed using the following formula:

If x = 1 then
    Step Freq[1] = Minimum Freq
Else if x > 8 then
    Step Freq[x] = (((Maximum Freq – Minimum Freq) /247) * (x – 8)) + Minimum Freq

Step Freq[2-8] is a special case. The range between step 1 and step 9 is divided into 8 smaller pieces to provide more precise low speed jogging capabilities. The interim speeds can be calculated as follows:
If x = 2-8
    Step Span = Step Freq[9] – Step Freq[1]
    Step Freq[x] = (Step Span * (x – 1) / 8) + Step Freq[1]

Speed[0] is always reserved for stopped. Minimum speed will be attained at Speed[1] and maximum at Speed[255].

Stepper motors require a linear speed progression (acceleration) and regression (deceleration) over time to reach speed without dropping "out of sync" and stalling. However, a stepper can normally be instantaneously started at some fraction of the maximum speed without requiring acceleration. The Start/Stop value sets this speed for each axis. Starting at this value rather than absolute minimum speed will speed completion of automated moves and the accel/ decel sequence. This value and the resulting frequency output will equate to "x" as shown in the formula above.

Once started, the motor will begin at the selected "safe" start speed, then accelerate as required to reach operational speed. The acceleration is performed by incrementing the array defined above by one count periodically. The Acc/Dec value defines the number of 50us delays to be introduced between each increment of speed. For example, an Acc/Dec value of 40 will equate to 40 * 50us = 2ms per speed increment. If safe Start/Stop speed was set at "50" then 255 – 50 or 205 speed increments would take place during acceleration. The platform would accelerate to full speed in 205 * 2ms or 410ms. Conversely, it will also decelerate in jog mode from full speed to a safe stop in the same 410ms time frame. When in jog mode, deceleration is the mirror of acceleration and the Start/Stop speed is also used to stop the motor without decelerating all the way down to absolute minimum speed.

When jogging accel and decel are purely functions of time. When performing automated moves, final position also must be considered when decelerating. The greatest efficiency is attained when the platform decelerates exactly to final position. This is performed by looking at the distance to final position in RU's and decelerating a speed step per "x/4" RU's. The ramp value determines the number of RU's per speed step. For example, we will assume top speed is 255 and safe Start/Stop speed is 55. From full speed we have 200 speed steps to a safe stop. If the ramp value is set to 1 we will begin decelerating 200 * 1/4 or 50 RU's from final position and reduce the speed four steps per RU. However, it may be found that this deceleration rate is too quick for the motor and drive train. Setting ramp to 8 will start deceleration at 200 * 8/4 or 400 RU's and reduce speed one step per 2 RU's. The objective of the user is to determine the proper accel/decel rate for jog, then attempt to emulate it in the ramp setting for automated moves.

Note that all values for each axis are platform, load, position, and direction dependent and must be derived by testing. It is recommended that the settings be tested when the load is moved from a position in a direction that requires the most torque from the axis. For tilt this would typically be an extreme "down" angle with a move up or vice versa. For pan this would typically be a move with the load as offset from center and unbalanced across the pan axis as possible. As a rule of thumb, if an axis immediately stalls when an automated move command is issued the Start/Stop speed is likely too high. If the axis starts but stalls at different points during acceleration the acceleration rate may be too low (accelerates too fast.) If the axis consistently overshoots the destination the ramp value may be to low. If the axis stalls fairly consistently at one speed it is possible the maximum step rate is too high for the load. In this case consult with the factory.

Setting the Query bit will instruct the PTCR to return the current values without changing them. Clearing the Query bit will actually load the new values into the EEPROM and update the speed parameters.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	92H	1								

P Start/Stop	xxH	1	Query	0-127
P Acc/Dec	xxH	1	0-255	
P Ramp	xxH	1	1-255	
P Reserve	xxH	1	0	0-127
T Start/Stop	xxH	1	0	0-127
T Acc/Dec	xxH	1	1-255	
T Ramp	xxH	1	1-255	
T Reserve	xxH	1	0	0-127

- Reply:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	92H	1								
P Start/Stop	xxH	1	0	0-127						
P Acc/Dec	xxH	1	0-255							
P Ramp	xxH	1	1-15							
P Reserve	xxH	1	0	0-127						
T Start/Stop	xxH	1	0	0-127						
T Acc/Dec	xxH	1	0-255							
T Ramp	xxH	1	1-15							
T Reserve	xxH	1	0	0-127						

2.10.5 "Initialize Preset Table to 0/0" Command:

This command initialize the entire preset table to 0°/0° (pan and tilt center) and set the zoom/focus bytes to 0. This can be used at initial setup to quickly clear the entire preset table. Reload the table into your application as required after issuing this command.

- Command:

Data	Format	Bytes
Command	94H	1

- Reply:

Data	Format	Bytes
Command	94H	1

2.10.6 "Get/Set Motor and Potentiometer/Resolver Direction" Command:

As the controller is fitted to different QPT types, motors and resolvers will be connected through different drive systems that may require reversing readings or direction of rotation in order to make them relate properly to the platform. This command allows altering these configurations.

A device set for reverse operation is not necessarily an indication of miss wiring or incorrect installation. It may simply be that the particular QPT type requires a motor or resolver to operate in the opposite direction due to design. For example, a non-continuous pan unit fitted with a potentiometer may increase its voltage output as the platform moves CW. A continuous rotation unit fitted with an encoder may actually decrease its count while moving clockwise. Both are correct. However, both will also return different directions of change when moving in the same physical direction. Configuring one unit as "Normal" and the other as "Inverted" in pan will correct this difference. A unit fitted with a higher or lower speed motor/gearbox combination may rotate the output shaft in the opposite direction. Therefore, though "Normal" mode may be the standard, this unit may operate in "Reverse" mode. Setting a bit to 0 will configure that device for "Normal" mode. Setting the bit to "1" will configure it for "Reverse/Inverse" mode. These values are set at the factory and should not normally be altered.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	95H	1								

Pan/Tilt	xxH	1	Query	0	0	0	TRES	PRES	TMTR	PMTR

- Reply:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	95H	1								
Pan/Tilt	xxH	1	0	0	[1]CON-TILT	[1]CON-PAN	TRES	PRES	TMTR	PMTR

[1]1 = Continuous Rotation, 0 = Non-continuous Rotation

2.10.7 "Get/Set Heater Configuration" Command:

Some QPT units are fitted with heaters. As the PTCR unit is normally used with low voltage DC systems the user may be limited as to the total power available for operating the unit, especially if operating from batteries. This command will allow the user to select an operating mode for the heater to match power availability as required. Legacy command provides only mode control of main board heaters.

No Heat Mode(Option 0): Turns off all heating option.
Share Power Mode(Option 1): Cycles Heater automatically based on on-board thermal sensor. The heater is always turned off when motors moving. It is power back depending on on-board thermal sensor. This limits instantaneous current draw.
Full Power Mode(Option 2): Cycles Heater automatically based on on-board thermal sensor.

Extended command will provide user to adjust temperature sensor set points for different heating operation.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	97H	1								
Config	xxH	1	Query	0 = No Heat, 1 = Share, 2 = Full Heat						

- Reply:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	97H	1								
Config	xxH	1	0	0 = No Heat, 1 = Share, 2 = Full Heat						

2.10.8 Get/Set Unit Resolution Type (only in new units)

New range of QPT units support extended protocol commands. In order to use extended command set, configure the unit to use extended command set using this command. The setting will be used after next power recycle.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	98H	1								
Encoder Type	xxH	1	Query	X	[1]Tilt Encoder Type			[1]Pan Encoder type		
Pan Encoder Total Count	xxH	3	Up to 999999							
Tilt Encoder Total Count	xxH	3	Up to 999999							

- Reply:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	98H	1								
Encoder Type	xxH	1	Query	X	[1]Tilt Encoder Type			[1]Pan Encoder type		
Pan Encoder Total Count	xxH	3	Up to 999999							

Tilt Encoder Total Count	xxH	3	Up to 999999

Pan and Tilt encoder types are currently not used as this rev of software is for incremental encoder. If the encoders are absolute then init encoder and homing etc. are not required. So setting these values will not have any effect.

[1]Tilt Encoder Type or [1]Pan Encoder type: 000 = incremental with index pulse. 001=incremental without index pulse, 010= absolute, 111= Reserved

Pan encoder and Tilt encoder values user can set depending on encoder installed. by default unit will use 36000 as encoder counts.

2.10.9 "Get/Set Communication Timeout" Command:

The embedded controller's sole method of operating and providing feedback is via the communication interface. Timely return of position and status information is important for many applications. However, some applications exist where feedback is of little or no use. For example, if a QPT unit is used to simply move a camera through a series of presets for viewing (a tour), constant return of positional information may not be needed. The user is only interested in the video returned by the camera. In this case, the user may wish to simply load a tour into the unit, start execution of the tour, then remove the communication connection and allow the unit to "free run." As two-wire RS-485 starts being introduced into the PTCR line it may not be possible or desired for a user's communications software to address all of the units in the communications daisy chain quickly. The capability to adjust or defeat the communication timeout value allows the user to assign a priority to the importance of constant communication.

Normally, a communication timeout is considered a fault of sufficient weight to stop any automated movement of the platform. If the data returned to the user's computer is critical, especially in determining the next move, the timeout should be set to a fairly low level (1-2 seconds.) If the user's software must share processing time and cannot service the QPT unit quickly a higher level can be set. If the user wishes the QPT to operate autonomously without the requirement for constant communication the user can set the timeout value for 0, defeating any stop due to a communication fault. Of course, all other faults will still remain active.

Setting the Query bit will instruct the PTCR to return the current value without changing it. Clearing the Query bit will actually load the new value into the EEPROM and update the fault timer.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	96H	1								
Timeout	xxH	1	Query	0(defeat) - 120 seconds						
LRC	xxH	1								
ETX	03H	1								

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	96H	1								
Timeout	xxH	1	0	0(defeat) - 120 seconds						
LRC	xxH	1								
ETX	03H	1								

2.10.10 Get firmware/s (backward comaptible)

The command will inquire about current firmware revision and date. PTCR95N/96 compatible command will only return main board firmware revision and date. However PTCR1000 extended command will return SW and its revision number for all the connected board in the system.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9AH	1								

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0

Command	9AH	1	
Major	xxH	1	00-99
Minor	xxH	1	00-99
Date	xxH	1	1-31
Month	xxH	1	1-12
Year	xxH	1	0-99

Extended Part

- Extended Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9AH	1								
Dummy	FFH	1								

- Extended Response:

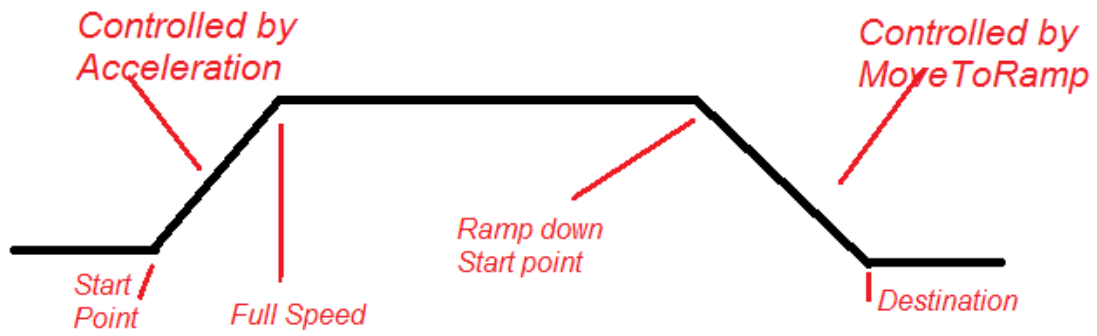
Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9AH	1								
Main Board										
Major	xxH	1	00-99							
Minor	xxH	1	00-99							
Date	xxH	1	1-31							
Month	xxH	1	1-12							
Year	xxH	1	0-99							
SW Number	Int	2	00-65535							
Pan Driver										
Major	xxH	1	00-99							
Minor	xxH	1	00-99							
SW Number	Int	2	00-65535							
Tilt Driver										
Major	xxH	1	00-99							
Minor	xxH	1	00-99							
SW Number	Int	2	00-65535							
Payload 1										
Major	xxH	1	00-99							
Minor	xxH	1	00-99							
SW Number	Int	2	00-65535							
Payload 2										
Major	xxH	1	00-99							
Minor	xxH	1	00-99							
SW Number	Int	2	00-65535							

The response may get extended depending on how many boards are connected.

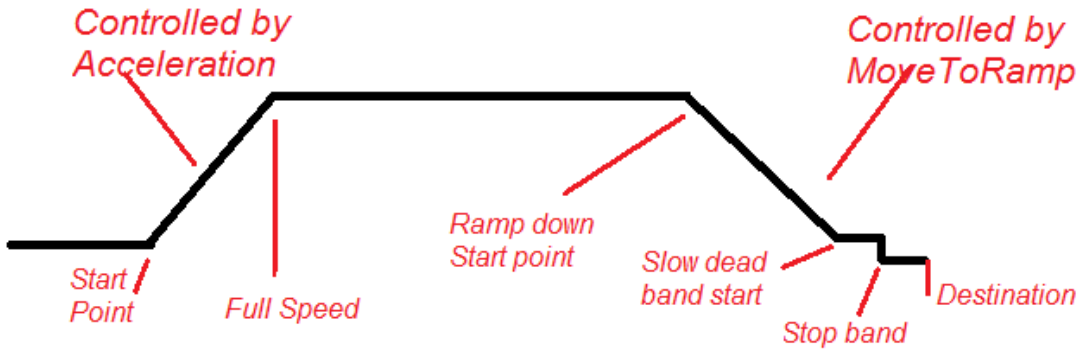
2.10.11 Get/Set Pan and Tilt Ramp - II parameters:

This command provides capability to adjust secondary ramp parameters. These parameters are useful in fine tuning of belt driver unit or wind load acting on payload and disturbing automated move.

Automated move with trapezoidal profile: (Without additional Ramp parameters)



Automated move with trapezoidal profile: (With additional Ramp Parameter)



- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9BH	1								
Pan Stop Band	xxH	1	Query	0-20 counts of encoder						
Pan Slow Band	xxH	1	0-30 counts of encoder							
Tilt Stop Band	xxH	1	0-20 counts of encoder							
Tilt Slow Band	xxH	1	0-30 counts of encoder							
LRC	xxH	1								
ETX	03H	1								

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9BH	1								
Pan Stop Band	xxH	1	0	0-20 counts of encoder						
Pan Slow Band	xxH	1	0-30 counts of encoder							
Tilt Stop Band	xxH	1	0-20 counts of encoder							
Tilt Slow Band	xxH	1	0-30 counts of encoder							
LRC	xxH	1								
ETX	03H	1								

These settings are persistent. By moving the unit at slow speed for few encoder count, system allows kinetic energy to die out on load side so when motor stop moving, system stops and do not experience jerk.

2.10.12 Get/Set Maximum Automated Move Speed:

The command will save maximum speed to be used during automated move if the commands do not have any separate specifications.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9CH	1								
Query	xxH	1	Query	0	0	0	0	0	0	0
Pan Max	Byte	1	Max Speed 0-255							
Tilt Max	Byte	1	Max Speed 0-255							

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9CH	1								
Pan Max	Byte	1	Max Speed 0-255							
Tilt Max	Byte	1	Max Speed 0-255							

2.10.13 Encoder Alignment and Tracking:

HARD LIMITS MUST BE SET PROPERLY BEFORE EXECUTING ANY ALIGNMENT COMMAND

MOTOR/RESOLVER NORMAL/REVERSE AND CONTINUOUS PAN MUST BE SET BEFORE EXECUTION

As the encoders used for tracing position are incremental, not absolute, position is maintained by constantly saving counts in non-volatile memory. While the unit is powered up the stored values will stay accurate but, if the unit loses power during a high speed move or is bumped when powered down, it is possible that some accuracy may be lost due to undetected, unsaved transitions of the encoders. Each encoder provides an index pulse at one position in the axis' full rotation, typically very close to the 0/0 position. An offset for each axis is stored in memory for each encoder's index count offset from physical center. Whenever an index pulse is crossed going CW for pan or UP for tilt, the PTCR will reload this offset into the respective encoder's count, maintaining full accuracy.

This offset is set at the factory using the "Initial Encoder Align" command. The user should align the platform by jogging to physical pan and tilt center. This command should then be issued. THE PLATFORM WILL STOP RESPONDING TO ANY CONNECTED HOST WHILE THE ALIGNMENT IS TAKING PLACE. The platform will automatically move in pan, then in tilt, looking for the index pulses. Once found, the offsets from physical center will be calculated and the counts will be stored and returned to the host. Communications will also resume. In the future, whenever the index pulses are crossed the encoder counts will be corrected relative to the stored values.

After initial alignment and when first powered up, the platform should be within a few counts of the correct reading, depending on how the unit was handled during the power down period. As the unit is operated and the index pulse is crossed in each axis the stored count offset will be automatically loaded, increasing the reading to full accuracy. Optionally, the user may also force a realignment by executing a "Perform Homing Cycle" command. The platform will automatically move through the index pulses in both axes, will update the count, then return to a corrected 0/0 position. THE PLATFORM WILL STOP RESPONDING TO ANY CONNECTED HOST WHILE THE ALIGNMENT IS TAKING PLACE. The TIND bit indicates that the tilt index was found. The PIND bit indicates that the pan index was found. If the procedure returns with either bit clear the encoders should be inspected for proper operation.

2.10.14 "Initial Encoder Align" Command:

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9DH	1								

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9DH	1								
Pan Index	Int	3	0							
Tilt Index	Int	3	0							

Index pulse is always at zero. Physical center may be at other then 0.

2.10.15 "Perform Homing Cycle" Command:

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9EH	1								

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9EH	1								
Pan/Tilt	xxH	1	0	0	0	0	0	0	[1]TIND	[2]PIND

Extended Part

Use this extended command to set power on homing mode and/or homing destination. By default homing destination is display 0/0 for Quickset Remote Emulator GUI that is absolute 0/0 and any offset set by user.

- Extended Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9EH	1								
Extension	FFH	1								
SubCmd	XXH	1	Q	X	[1]P-ON	[3]SET CODE			[2]HOMING MODE	

[1]P-ON:

0 = No power on homing
1 = perform homing at power on.

[2]HOMING MODE:

00,01= after getting index go to abs 0/0. This mode is default when not configured any thing.
10 = after getting index go to disp 0/01
11 = after getting index go to Preset 31.

[3]SET CODE: Fix code 111.

- Extended Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9EH	1								
Pan/Tilt	xxH	1	1	1	1	1	1	1	1	1
SubCmd	xxH	1	X	X	[1]P-ON	1	1	1	[2]HOMING MODE	

2.10.16 "Get/Set Identity Address" Command:

The identity address is used to uniquely identify a unit in a daisy-chained RS-485 environment. When the identity address is sent only the unit with a matching identity address will parse the incoming data, execute the command, seize the host receive line, respond, then release the line. The absence of responses from the other units allows a clear path for data return on the receive data line.

This command can be used to initially set or change the unit's identity address. If the user knows the current address of the unit to modify the current address should be sent as the Identity with the new address sent as the New Identity data byte. The unit will respond with the current address as the Identity and the new address as the New Identity response byte. **From this point forward the unit's identity address has been changed and it will only respond to the new address.**

The identity can always be retrieved or changed by sending the 00 "broadcast" as the identity address. Any command will result in the return of the current identity in the response. Note, however, that any and all units in the network will act upon and respond to a broadcast command. **Therefore, the unit to modify must be isolated from the other units by either disconnecting the other units from the daisy chain or by connecting the unit to modify directly to a dedicated host.**

When operating in a dedicated RS-232 or RS-422 mode the identity should be set for 00. This will cause the dedicated unit to seize and hold the host's receive line full time.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9FH	1								
New Identity	xxH	1	Query	1-99 (01H-63H) for RS-485 identity or 00 for dedicated						

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9FH	1								
New Identity	xxH	1	0	New Identity = 00-99 (00H-63H)						

2.11 PTCR Factory Setup Parameters

Factory setup parameters are selected for the specific type of platform being controlled. Alteration of these parameters without guidance from Quickset will likely result in damage to the platform or injury to personnel.

2.11.1 Get and Set Step Rate Minimum and Maximum:

The pan and tilt step rate parameters determine the speed range for each axis. The platform can provide 255 distinct speeds. This command allows the user to define the minimum and maximum step rate for each axis. The controller will then calculate the 255 step rates from minimum to maximum. Pan and tilt speed ranges can be set completely independent of each other. One may approximate an array speed using the following formula:

Step Freq[x] = ((Maximum Freq – Minimum Freq) /255) * x + Minimum Freq

Speed[0] is always reserved for stopped. Minimum speed will be attained at Speed[1] and maximum at Speed[255].

2.11.2 "Get/Set Min/Max Pan Motor Speed" Command:

The command sets motor speed in d/s. Payload speed may be different depending on gearing.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	A1H	1								
Query	xxH	1	Query	0	0	0	0	0	0	[1]PCON
Pan Min	Int	2	>= 1 d/s							
Pan Max	Int	2	<= 20000 d/s							

[1]PCON --> 1 = Pan Continuous Rotation, 0 = Non-continuous Pan Rotation

Note: Continuous or non-continuous depends on underlying hardware. The Step Rate commands shows motor minimum and maximum speed in d/s. (not the actual step rate)

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	A1H	1								
Pan Min	Int	2	>= 1 d/s							
Pan Max	Int	2	<= 20000 d/s							

2.11.3 "Get/Set Min/Max Tilt Motor Speed" Command:

The command sets motor speed in d/s. Payload speed may be different depending on gearing.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	A2H	1								
Query	xxH	1	Query	0	0	0	0	0	0	[1]TCON
Tilt Min	Int	2	>= 1 d/s							

Tilt Max	Int	2	<= 20000 d/s

[1]TCON --> 1 = Tilt Continuous Rotation, Non-continuous Pan Rotation

Note: Continuous or non-continuous depends on under laying hardware. The Step Rate commands shows motor minimum and maximum speed in d/s. (not the actual step rate)

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	A2H	1								
Tilt Min	Int	2	>= 1 d/s							
Tilt Max	Int	2	<= 20000 d/s							

2.11.4 "Set Serial Number / Model number" Command:

Factory only command. Restricted for authorized use only. This command is used to save serial number and model number of the unit. The data is then used to display serial number / model number on OSD as well user can retrieve the number in case he lost the paper work. Automatic report generation tool uses this serial number so that Quickset Service department can quickly trace the unit's paper work without going back to customer. Serial number and model number are unique for each model.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	A4H	1								
Serial number	Bytes	13	Data (all bytes should be ASCII numbers between 0-9. Hex code 30H to 39H). Filling all bytes is mandatory.							
Model number	Bytes	15	ASCII data including number, character and special characters. Use 20H (Fill unused bytes with space. This is mandatory)							

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	A4H	1								
Serial number	Bytes	13	Data (all bytes should be ASCII numbers between 0-9. Hex code 30H to 39H)							
Model Number	Bytes	15	ASCII data including number, character and special characters. Use 20H (Fill unused bytes with space. This is mandatory)							

2.11.5 "Set ID for Oracle Board" Command: (Oracle Board Set Related Command)

Factory only command. Restricted for authorized use only. This command is used to assign ID to Motor driver board and payload boards as their firmware is common and use depends on ID byte saved. There should only be one board powered on the Oracle IBUS other then Main Control Node in order to set ID for the board.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	ABH	1								
Board ID	Byte	1	ID for the board must be between 03H to 0FH							

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	ABH	1								
Board ID	Byte	1	ID for the board must be between 03H to 0FH							

Ex: To make a Motor Driver Board, a PAN Axis Motor Driver Board it needs ID 03H. A newly manufactured and flashed board do not have ID. Connect the Motor Driver Board to Main Controller Node. Make sure that the Motor Driver Board is powered on and the only board on Oracle IBUS besides Main Controller Node. Establish

communication with Main Controller Node and then send the command 02H 00H ABH 1BH 83H A8H 03H (ID of 03H requires escape sequence). Once the command is acknowledged by Main Controller Node, the Motor Driver Board is ready to work as Pan Axis Motor Driver Board. To see ID list check Oracle Document.

2.12 Stabilization Feature Handling:

These features are available only if the PTZ is equipped with stabilization module.

2.12.1 Stabilization Status get/ set Command:

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Cmd Num	B1H	1								
Stab cmd	Bitset	1	Query						Stab On/Off	Gyro Init

- Set Stab On/ Off = 1 to turn stabilization on.
- Set Stab On/ Off = 0 to turn stabilization off.
- Set Query bit to get status. When Query bit is set, all other bits are ignored.
- During stabilized mode, system will acknowledge but not execute gyro initialization command

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Cmd Num	B1H	1								
Stab status	Bitset	1					PTZ Err		Stab	Gyro

- Gyro = 1 reports gyro sensor ok otherwise gyro sensor has error.
- Stab = 1 stabilization is on.
- Stab = 0 reports stabilization is off.
- PTZ Err: This bit, when set, indicates there is a communication link error between PTZ and Stab system.

(Restricted access for factory use)

2.12.2 Stabilization PID parameter get/ set command: (set query bit to get value)

- Command:

Data	Format	Bytes								
Command	B2H	1								
Parameter	Bitset	1	Query	Query Gain/ Range	0	0	0	0	0	0
PAN P Gain	Int	2	0 to max range							
PAN I Gain	Int	2	0 to max range							
PAN D Gain	Int	2	0 to max range							
TILT P Gain	Int	2	0 to max range							
TILT I Gain	Int	2	0 to max range							
TILT D Gain	Int	2	0 to max range							

- Set Query bit and clear Query Gain/Range bit to get status.
- Set both bits to get max ranges for PID parameters from the unit. (This is useful to scale the software from generic 0 to 100 values).

- Response:

Data	Format	Bytes	
Command	B2H	1	
PAN P Gain	Int	2	0 to max range OR max range
PAN I Gain	Int	2	0 to max range OR max range
PAN D Gain	Int	2	0 to max range OR max range
TILT P Gain	Int	2	0 to max range OR max range
TILT I Gain	Int	2	0 to max range OR max range

TILT D Gain	Int	2	0 to max range OR max range

- P, I, D gains are value of P, I, D parameters of PID loop.
- Default values used with ApolloStab50 Rev 1 is P = 400, I = 1200 and D=0 for PAN and P = 600, I = 800 and D=0 for TILT.

2.12.3 Gyro drift compensation Get / Set command: (set query bit to get value):

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	B5H	1								
SeqNo	Bitset	1	Query	0	0	0	0	0	0	On/ Off

- If Query bit is set, unit will return current status of Gyro drift.
- Clear Query bit and set On/Off bit to turn on Gyro drift compensation.
- Clear Query and On/Off bits to turn off Gyro drift compensation.

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	B5H	1								
SeqNo	Bitset	1	0	0	0	0	0	0	0	On/ Off

- On/Off bit will indicate current status of Gyro drift compensation.

2.13 Geo Referencing Features:

These features provides user with ability to use geo referencing coordinates for PT positioning. It may use GPS connected to any serial port to the PT or PT geo reference position set by user.

2.13.1 Get / Set PTCR's Geo Position Command:

The command is used to get or set unit's current geo reference positions. Latitude and Longitude are defined as Deg * 1000000 and Altitude is defined as mtr * 10 above MSL.

This command also defines that the unit, either uses fixed user defined geo location as PTCR's GPS position or uses live GPS data from any incoming NMEA GPGGA information either on UART 1 or on UART2 connection.

For ex. 42 deg 08.93338' N is 42.148889 deg and 42 deg 08.93338' S is -42.148889 deg. use +ve for East and -ve for West.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Cmd Num	BAH	1								
Cmd	Bitset	1	[1]Query				[2]Sub Command			
Latitude	Bytes	4	Deg * 1000000. (+ve for N and -ve for S) or "USE GPS" code							
Longitude	Bytes	4	Deg * 1000000. (+ve for E and -ve for W) or "USE GPS" code							
Altitude	Bytes	4	Mtrs * 10 or "USE GPS" code							

[2]Sub Command = 0001 for PTCR geo position mode
= 0101 for PTCR configured GPS location
= 1011 for PTCR current GPS location, volatile.

When [1]Query bit is set, PTCR ignores next 12 byte if entered by user. User can safely omit them from command.

Get Information from Unit:

To Retrieve PTCR's configured GPS location:
Set [1]Query bit and use [2]Sub Command = 0101. This location is used in absence of GPS or when the unit is set to use fixed geo location. (Cmd byte = 0x85)

To Retrieve PTCR's current GPS location:
Set [1]Query bit and use [2]Sub Command = 1011. (Cmd byte = 0x8B)

To Retrieve PTCR's Geo position mode:
Set [1]Query bit and use [2]Sub Command = 0001. (Cmd byte = 0x81)

Set Information to Unit: (Configure Unit):

Configure PTCR's GPS location:
Clear [1]Query bit and set [2]Sub Command = 0101. (Cmd byte = 0x05)
Next 12 bytes represens Latitude (4 bytes), Longitude(4 bytes) and Altitude(4 bytes) should contain valid values. Valid values are Latitude between and including +/- 90.000000 deg, Longitude between and including +/- 180.000000, Altitude between and including -1000.0 to +50000.0 mtrs. This location is used in absence of GPS. (Default is Northbrook, IL)

Set PTCR's geo position mode:
Clear [1]Query bit and [2]Sub Command = 0001. (cmd byte = 0x01)
Set next 12 bytes to *code 0x99 (Hex 99) each* to use GPS based location. GPS need to be connected to UART1 or UART2 and must use NMEA protocol. Unit uses GPGGA information to find out current geo location.

Or

Set next 12 bytes to *code 0x33 (Hex 33) each* to use Fixed configured location.

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Cmd Num	BAH	1								
Cmd	Bitset	1	Q	[3]GPSA		[4]GPSC	[2]Sub Command			
Latitude	Bytes	4	Deg * 1000000. (+ve for N and -ve for S)							
Longitude	Bytes	4	Deg * 1000000. (+ve for E and -ve for W)							
Altitude	Bytes	4	Mtrs * 10							
NoOfSats	Byte	1	No of Satellites (only when GPSA=1 and subcommand=1011(0x0B))							
HDOP	Byte	2	Quality of Fix (*100).							

[3]GPSA = 0 no Fix for GPS/ GPS not available.
= 1 GPS available with Fix.
= 2 DGPS available with Fix or higher. (Differential GPS)

[4]GPSC = 0 meaning unit not configured to use GPS.
= 1 Unit is configured to use GPS.

Retrieve PTCR's current GPS location response:
PTCR sets [2]Sub Command = 1011. PTCR also sets correct GPS available and configure status bits. Next 12 bytes indicates PTCR's current GPS location in form of Latitude, Longitude and Altitude. This may come from GPS or PTCR's NVM (Non volatile Memory) depending of GPS is configured and available.

Configure PTCR's GPS location response:
PTCR sets [2]Sub Command = 0101. PTCR also sets correct GPS available and configure status bits. Next 12 bytes indicates PTCR's configured GPS location in form of Latitude, Longitude and Altitude. This may differ from commanded location depending on validity.

Retrieve PTCR's geo position mode:
PTCR sets [2]Sub Command = 0001. PTCR also sets correct GPS available and configure status bits. Next 12 bytes indicates PTCR's current GPS location in form of Latitude, Longitude and Altitude. This may come from GPS or PTCR's NVM depending of GPS is configured and available.

Set PTCR's geo position mode response:
PTCR sets [2]Sub Command = 0001. PTCR also sets correct GPS available and configure status bits. Next 12 bytes indicates PTCR's current GPS location in form of Latitude, Longitude and Altitude. This may come from GPS or PTCR's NVM depending of GPS is configured and available.

2.13.2 Get/Set North position Command:

To follow GPS based target, unit need to know where the North is w.r.t. its center position. This command provides user the capability of setting North either by entering PAN North offset w.r.t. its center position or moving unit to North and setting current position as North.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	BBH	1								
cmd	xxH	1	Q							**[1]Mode**
PAN North offset	Int	3	PAN = -18000 to +18000 = -180.00° to +180.00°							

[1]Mode = 0 means use user specified offset to set PTCR's North position from PAN center.
= 1 means set current position as North position.

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	BBH	1								
cmd	xxH	1	Q							
PAN North offset	Int	3	PAN = -36000 to +36000 = -360.00° to +360.00°							

Returns the North offset from Pan Center position.

2.13.3 Move To GPS Target location Command: (Entered Location / Preset / Last Visited Location)

To follow GPS based target, PTCR provides 3 different options using this command.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	BCH	1								
cmd	xxH	1	X	[1]Dest	Preset Number 0-31					
Latitude	Bytes	4	Deg * 1000000. (+ve for N and -ve for S) , 0x99999999 to move to last visited GPS target							
Longitude	Bytes	4	Deg * 1000000. (+ve for E and -ve for W), 0x99999999 to move to last visited GPS target							
Altitude	Bytes	4	Mtrs * 10, 0x99999999 to move to last visited GPS target							

[1]Dest = 0 means move to GPS Preset location as specified.
= 1 means move to entered GPS location.

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	BCH	1								
cmd	xxH	1	X	[1]Dest	Preset Number 0-31					
Latitude	Bytes	4	Deg * 1000000. (+ve for N and -ve for S), 0x99999999 for not valid target latitude							
Longitude	Bytes	4	Deg * 1000000. (+ve for E and -ve for W), 0x99999999 for not valid target long.							
Altitude	Bytes	4	Mtrs * 10, 0x99999999 for not valid target altitude.							

2.13.4 Get / Set / Clear GPS Preset Command:

GPS based Preset Table:
The PTCR can retain up to 32 (0-31) GPS based target position entries in a non-volatile preset table that are frequently used by the operator. Each position in this preset table represents following entities,

- Latitude of object point,
- Longitude of object point,
- Altitude of object point,

This command is used to Set preset position / Query preset position or Clear preset position or table.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	BDH	1								
Preset	xxH	1	Q	Preset Number 0-31, 93 (0x5D) to clear preset table						
Action	xxH	1	0xAA = Set Preset 0x55 = Clear Preset 0xFF = Clear Preset Table							
Latitude	Bytes	4	Deg * 1000000. (+ve for N and -ve for S) , 0x99999999 to set preset to last visited target Latitude							
Longitude	Bytes	4	Deg * 1000000. (+ve for E and -ve for W), 0x99999999 to set preset to last visited target Longitude							
Altitude	Bytes	4	Mtrs * 10, 0x99999999 to set preset to last visited target Altitude							

- If Q bit is set, "Action" byte has no effect.
- For Clear Preset and Clear Preset Table Action, next 12 bytes are ignored and user can safely omit it. Clear Preset requires valid preset number and code 0x55. Clear Preset Table requires Preset Number of 127 and code of 0xFF.
- For SET PRESET command user can specify GPS Target location or last visited GPS target location as described in command.

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	BDH	1								
Preset	xxH	1	Q	Preset Number 0-31, 93 (0x5D) for Preset Table Cleared.						
Action	xxH	1	0xAA = Preset Set to following coordinates 0x55 = Preset Cleared 0xFF = Clear Preset Table 0x00 = Preset number not valid.							
Latitude	Bytes	4	Deg * 1000000. (+ve for N and -ve for S) , 0x99999999 for latitude not set							
Longitude	Bytes	4	Deg * 1000000. (+ve for E and -ve for W), 0x99999999 for Longitude not set							
Altitude	Bytes	4	Mtrs * 10, 0x99999999 for Altitude not set							

Clear Preset and Clear Preset Table reply will show Latitude, Longitude and Altitude not set.

2.13.5 Query / Clear GPS target History points Command:

PTCR maintains history from last visited GPS target locations from "Move To Entered GPS Target location " command. User can retrieve or clear history using this command. The history is persistent and FIFO type.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Cmd Num	BEH	1								
Cmd	Bitset	1	Query	History point 0-19. Set to 127 to clear history table without Query bit						

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Cmd Num	BEH	1								
Cmd	Bitset	1	Query	= 127 indicates history cleared.						
Latitude	Bytes	4	Deg * 1000000. (+ve for N and -ve for S) or H99999999 for no history							
Longitude	Bytes	4	Deg * 1000000. (+ve for E and -ve for W) or H99999999 for no history							
Altitude	Bytes	4	Mtrs * 10 or H99999999 for no history							

2.14 Debug and Health Monitoring:

2.14.1 Get/Reset Debug Data

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	C0H	1								
Cmd	Byte	1	Reset[1]	6	5	4	3	2	1	0

[1]Reset = 1, resets all accumulated debug data

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	C0H	1								
Encoder 0 Errors	Bytes	4	Long word (stored to NVRAM when there is an error)							
Encoder 1 Errors	Bytes	4	Long word (stored to NVRAM when there is an error)							
Machine Run Hour	Bytes	4	Long word In seconds (stored every second)							
Pan Motor Move Hour	Bytes	4	Long word in seconds (stored every accumulated second)							
Tilt Motor Move Hour	Bytes	4	Long word in seconds (stored every accumulated second)							
UART Errors	Bytes	4	Long word (Stored to NVRAM when there is an error)							

2.15 PTCR User Configuration Parameters - Special

2.15.1 "Get Serial Number / Model Number" Command:

This command provides user the ability to retrieve unit's serial number and model number to report back or reorder.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	D0H	1								

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	D0H	1								
Serial number	Bytes	13	Data							
Model number	Bytes	15	Data							

2.15.2 "Get/Set Serial Type and Terminate" Command:

This command allows users to choose between different serial communication modes and allow to form a serial line network. The two COM bit defines the serial line interface and TERM bit defines the termination resistor position. TERM is automatically ignored in case of RS-232 is choosen.

- 00 for RS-232, 01 for RS-422, 10 for RS-485 2-wire half duplex, 11 for RS-485 4-wire full duplex

The PTCR-1000 can operate either as a singly controlled unit or as a member of a chain of networked units on serial line interface. An identity byte is always included in both transmit and receive packets, allowing a host to specifically address a single unit. When operating in a dedicated host-to-platform mode it may be preferable to keep the platform's TX line at a known level during idle time. In this case, the unit should normally be configured for "full duplex" mode. Many line powered RS-422/RS-232 converters require that both TX and RX lines be held high during idle to produce parasitic power for the converter to use. In this instance the unit should also be configured for "full duplex" mode.

A 120Ω termination resistor is installed across the host-to-platform RX line and can be connected by setting the TERM bit. The termination resistor will provide known impedance to the line to reduce reflection and improve reliability and noise immunity. The termination resistor should normally be activated on any one-to-one unit or on a single unit furthest from the host in a network. Set bit 1 to terminate the RX line. Setting the Query bit returns the current setting while clearing the bit instructs the PTL-11 to both activate the setting and write it to FRAM as an initial setting at power up. Requirement of termination resistor depends on many parameters including cable type, distance and baud rate.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	D1H	1								
Query	xxH	1	Query	0	[1]Port		TERM	[2]COM		

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	D1H	1								
Query	xxH	1	0	0	[1]Port		TERM	[2]COM		

[1]00 = Port 0, 01 = Port 1, 10 = PORT 2, 11 = PORT 3 (For Multi port user interface like Oracle MCC)
[2]000 = RS-232, 001 = RS-422, 010 = RS-485 2-wire half duplex, 011 = RS-485 4-wire full duplex.

The command is extended to provide auto baud v/s fix selected baud capability to system. To set baud rate use following extension. This is persistence setting and on4ce fix baud is selected, unit will not respond to other baud rates until changed.

- Command:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	D1H	1								
Query	xxH	1	Query	1	[1]Port		[2]Baud Rate			

- Response:

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	D1H	1								
Query	xxH	1	Query	1	[1]Port		[2]Baud Rate			

[1]00 = Port 0, 01 = Port 1, 10 = PORT 2, 11 = PORT 3 (For Multi port user interface like Oracle MCC)

[2]Baud Rate = 0 for Auto baud (Between 2400, 4800, 9800, 14400, 19200, 28800, 38400, 57600, 115200)
= 1 for 1200
= 2 for 2400
= 3 for 4800
= 4 for 9600
= 5 for 14400
= 6 for 19200
= 7 for 28800
= 8 for 38400
= 9 for 57600
= 10 for 115200
> 10 not used.

3. Code Examples

The following snippets from the PTCR microcontroller code provide some examples of how to implement efficient communications with the PTCR. Though the code is specifically written for a microcontroller compiler the unfamiliar commands should be self-explanatory. The snippets include LRC calculation and ESC/Bit-7 Set handling. Both transmit and receive procedures are interrupt based in these examples. **Note that, for the host's end, the handling of STX and ACK should be reversed.**

```
// Serial Communications Constants
#define  STX         0x02
#define  ETX         0x03
#define  ACK         0x06
#define  NAK         0x15
#define  ESC         0x1B


// Serial Transmit Interrupt Triggered on TX Buffer Empty
#int_tbe
tbe_isr() {
   if( tx_ptr == tx_len ) {
      putc( ETX );                        // send ETX and
      disable_interrupts(INT_TBE);        // buffer empty so disable transmit
   }
   else {
      switch( tx_buff[tx_ptr] ) {         // if a control character
         case STX :
         case ETX :
         case ESC :
         case ACK :
         case NAK :  putchar( ESC );                  // send and escape
                     bit_set( tx_buff[tx_ptr], 7 );   // and set bit 7 of data byte
                     break;
         default  :  putchar( tx_buff[tx_ptr] );      // else just handle normally
                     tx_ptr++;
                     break;
      }
   }
}


// Serial Receive Interrupt
// Reception of an ETX must also follow setting an "STX found" flag so we
// will not try to parse a partial reception of data.
#int_rda
rda_isr() {
   int   temp_rx;
   temp_rx = getc();                      // get character from serial port
   switch( temp_rx ) {
      case STX :  rx_ptr = 0;             // realign to front of buffer
                  rx_done = FALSE;        // new stream coming, last corrupt
                  esc_flag = FALSE;       // clear possible escape flag
                  found_id = FALSE;       // clear identity found flag
                  found_stx = TRUE;       // and indicate an STX was found
                  break;
      case ETX :  if( found_stx ) {       // if we started with an STX
                     rx_done = TRUE;      // flag done
                     found_stx = FALSE;   // clear to find another STX
                  }
```

```
                  found_id = FALSE;       // clear to find another identity
                  esc_flag = FALSE;       // and clear possible escape flag
                  break;
      case ESC :  esc_flag = TRUE;        // escape found so flag
                  break;
      default  :  if( esc_flag ) {        // if last char was an escape
                     bit_clear( temp_rx, 7 );   // clear bit 7
                     esc_flag = FALSE;          // and clear flag
                  }
                  if(( found_STX ) && (!found_id)) {  next byte must be ID
                     rx_id = temp_rx;
                     found_id = TRUE;
                  }
                  else {
                     rx_buff[ rx_ptr ] = temp_rx;     // save byte
                     if( rx_ptr == 50 )               // prevents buffer overrun
                        rx_ptr = 0;
                     else
                        rx_ptr++;                     // and increment pointer
                  }
                  break;
   }
}

// Calculates and returns transmit LRC
// length is actually number of bytes, not top subscript
int Calc_LRC( int length ) {
   int temp_LRC = 0, index;
   for( index = 0; index < length; index++ )
      temp_LRC ^= tx_buff[index];
   return( temp_LRC );
}

// Calculates and returns TRUE if LRC was good or FALSE if LRC failed
short Check_LRC(void) {
   int temp_LRC, index;
   temp_LRC = rx_id;                       // prime with receive identity
   for( index = 0; index < rx_ptr; index++ )
      temp_LRC ^= rx_buff[index];
   return( temp_lrc == 0 );
}


// Send data
void send_data( int cmd_num ) {
   tx_buff[0] = unit_id;
   tx_buff[1] = cmd_num;
   tx_buff[2] = somedata;
   tx_buff[3] = moredata;
   tx_buff[4] = Calc_LRC( 4 );
   tx_len = 5;
   tx_ptr = 0;
   putchar(ACK);
   enable_interrupts(INT_TBE);      // triggers on TX buffer empty
}

void main( void ) {

disable_interrupts(INT_TBE);
```

```
enable_interrupts(INT_RDA);

< Code >

if( rx_done ) {                     // if we have received a packet
   rx_done = FALSE;                 // reset for next packet whether good or bad
   if( Check_LRC()) {               // check the received LRC
      process data;                 // if good, process the data
      act_on_data;
      gather_return_data;
      send_data( echo_command );    // and send the response
   }
}
```

4. Revision History:

4.1 Rev 1.0.0 06/20/2012

Initial Release.