MN00105 5/20/2010

Protocol Manual

PRODUCT FAMILY: SENTRY SERIES USING PTCR-95
PRODUCT NAME: PTCR-95 EMBEDDED CONTROLLER
PRODUCT MODELS: All PTCR-95 9K Encoder Stepper Models

MOOG QUICKSET
3650 WOODHEAD DRIVE
NORTHBROOK, IL 60062
TEL: 800 247-6563 • 847 498-0700 • FAX: 847 498-1258
WWW.QUICKSET.COM • E-MAIL: SALES@QUICKSET.COM

PTCR-95 Embedded Controller Protocol Rev N4

Table of Contents

1. PAN & TILT COMMUNICATIONS PROTOCOL OVERVIEW: 3

1.1 AUTOBAUD: 3
1.2 STORAGE OF COORDINATE INFORMATION: 4
1.3 CAMERA AND LENS CONTROL: 5
1.4 GENERAL COMMUNICATIONS STRUCTURE: 5
1.5 CONTROL CHARACTERS: 6
1.6 CALCULATING THE LRC CHECKSUM: 6
1.7 PASSING DATA THAT MATCHES CONTROL CHARACTER VALUES (ESC/BIT-7 SET): 7
1.8 PASSING 16-BIT, 24-BIT, AND 32-BIT INTEGER VALUES: 7
1.9 THE PROGRAMMER'S RESPONSIBILITY FOR INPUT RANGE AND FORMAT 8
1.10 STATUS DEFINITIONS: 8

2. PTCR COMMAND SET: 9

2.1 NUMERIC COMMAND LIST: 9
2.2 "GET STATUS/JOG" & "KEEP ALIVE" COMMAND 9
2.3 AUTOMATED "MOVE TO" COMMANDS: 10
2.3.1 "Move To Preset" Command: 11
2.3.2 "Move To Entered Coordinate" Command: 11
2.3.3 "Move To Delta Coordinates" Command: 12
2.3.4 "Move To Absolute 0/0" Command: 13
2.3.5 "Move To Home" Command: 14
2.3.6 "Move To Entered Pan/Tilt/Zoom/Focus" Command: 14
2.4 THE PRESET TABLE: 15
2.4.1 "Retrieve Preset Table Entry" Command: 15
2.4.2 "Save Coordinates As Preset Table Entry" Command: 15
2.4.3 "Save Current Position As Preset Table Entry" Command: 16
2.4.4 "Save Current Zoom/Focus Positions To Preset Table Entry" Command: 16
2.5 THE PRESET TOUR: 16
2.5.1 "Start Preset Tour" Command: 17
2.5.2 "Flush Preset Tour" Command: 17
2.5.3 "Query Preset Tour" Command: 17
2.5.4 "Append To Preset Tour" Command: 18
2.5.5 "Insert Into Preset Tour" Command: 18
2.5.6 "Delete From Preset Tour" Command: 19
2.5.7 "Replace In Preset Tour" Command: 19
2.5.8 "Get Tour Size" Command: 19
2.6 CAMERA/LENS PARAMETERS AND CONTROL 19
2.6.1 "Get/Set Camera Comm Parameters" Command: 20
2.6.2 "Get/Set Lens Parameters" Command: 20
2.6.3 "Command Camera" Command: 21
2.6.4 "Get/Set Camera Select/Power/Video" Command: 22
2.6.5 "Get/Set Camera Response Timeout" Command: 22
2.6.6 "Get/Set Aux Control Outputs" Command 23
2.7 PTCR OPERATING PARAMETERS: 23
2.7.1 Setting Pan & Tilt Angle Corrections: 23
2.7.2 "Get Pan & Tilt Angle Correction" Command: 23
2.7.3 "Set Pan & Tilt Angle Correction" Command: 24
2.7.4 "Align To Center" Command: 24
2.7.5 "Align To Coordinate" Command: 24
2.7.6 "Clear Angle Corrections" Command: 25

2.7.7 "Get/Set Pan & Tilt Soft Limits" Command: 25
2.8 PTCR INITIAL SETUP PARAMETERS: 26
2.8.1 Getting/Setting Pan & Tilt Center Position: 26
2.8.2 "Get Pan & Tilt Potentiometer/Encoder Center Position" Command: 26
2.8.3 "Set Pan & Tilt Potentiometer Center Position" Command: 26
2.8.4 "Get/Set Pan & Tilt Ramp Parameters" Command: 26
2.8.5 "Initialize Preset Table to 0/0" Command: 28
2.8.6 "Get/Set Motor and Potentiometer/Resolver Direction" Command: 28
2.8.7 "Get/Set Heater Configuration" Command: 29
2.8.8 "Get/Set Communication Timeout" Command: 29
2.8.9 "Get/Set/Store Maximum Speed" Command: 30
2.8.10 Encoder Alignment and Tracking: 30
2.8.11 "Initial Encoder Align" Command: 31
2.8.12 "Perform Homing Cycle" Command: 31
2.8.13 "Get/Set Identity Address" Command: 31
2.9 PTCR FACTORY SETUP PARAMETERS 32
2.9.1 Get and Set Step Rate Minimum and Maximum 32
2.9.2 "Get/Set Pan Step Rate Parameters" Command: 33
2.9.3 "Get/Set Tilt Step Rate Parameters" Command: 33
2.9.4 Configuring Resolver Parameters 33

3. CODE EXAMPLES 34

4. REVISION HISTORY: 36

4.1 REV A 06/02/05 36
4.2 REV B 06/21/05 36
4.3 REV C 02/23/06 36
4.4 REV N1 04/11/06 36
4.5 REV N2 05/02/06 36
4.6 REV N3 01/18/08 37
4.7 REV N4 05/20/09 37

1. Pan & Tilt Communications Protocol Overview:

The controller PCB mounted internal to the QPT-90 (PTCR-95) will be addressable via an dedicated RS-422, dedicated RS-232, or shared half- or full-duplex (2-wire/4-wire) RS-485 communications link provided by an attached remote host. The host will act as master and be responsible for initiating and maintaining communications with the PTCR. The PTCR will remain idle until the server sends a query/command. The PTCR will then parse the packet, check for integrity, and respond as required. The host should not send packets faster than 10 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.

1.1 Autobaud:

An automatic baud detection procedure (autobaud) has been implemented. Supported baud rates are 9600bps, 14.4kbps, 19.2kbps, 28.8kbps, 38.4kbps, and 57.6kbps. Structure is 8 data bits, 1 stop bit, and no parity. Autobaud 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. However, to change baud rates, the user may power the PTCR down, change host baud rate, then power the unit 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 resync the PTCR to the host, even without the autobaud feature. With autobaud the host must be able to transmit up to 125-150 bytes of data (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.

1.2 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 every 50us 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 initially configured at the factory, this index point is equated to a resolver value relative to absolute center in pan 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 corrected using the stored index value. 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.

Each axis is fitted with a 9000-line encoder. 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.

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 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°.

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.

1.3 Camera and Lens Control:

The PTCR actually contains three microcontrollers; the main microcontroller dedicated to controlling the pan & tilt unit and dual slave microcontrollers dedicated to camera and lens control. Several non-volatile parameters are set and stored for identifying the operation of the lens and for determining whether a camera has serial remote control capabilities 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.

Lens control is straightforward. A slave microcontroller controls two proportional-speed bi-directional motor drivers. One driver controls the focus motor and the second controls the zoom motor. Lenses are fitted with potentiometers for positional feedback. The potentiometer output for each axis is connected to the slave microcontroller ADC's 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 and focus on both cameras.

Camera control is more difficult due to the numerous control signals available for each camera type. Therefore, a special variable-length command 62H is used to wrap a complete camera command for transfer to the camera. The user should build a complete camera command string. It should then be prepended and appended with the appropriate PTCR control bytes for transfer to the main microcontroller. The main microcontroller will then strip the intact camera command out of the packet and transfer it to the slave microcontroller where it will then be sent to the camera. See command 62H, "Command Camera", below for more detail.

The PTCR can also return received camera data through the “Get Status/Jog” command and 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.

1.4 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.**

1.5 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

1.6 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.

1.7 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.

1.8 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 converts to 007FFFFFH as a 32-bit number. -8388608 converts to FF800000H.

1.9 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 user 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.

1.10 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
31H	Get Status/Jog	63H	Get/Set Select Camera/Power/Video
32H	Move To Preset	65H	Get/Set Camera Timeouts
33H	Move To Entered Coordinates	66H	Get/Set Aux Control Outputs
34H	Move To Delta Coordinates		
35H	Move To Absolute 0/0	80H	Set Pan & Tilt Angle Correction
36H	Move To Home	81H	Get/Set Soft Limit
37H	Start Preset Tour	82H	Align Angles To Center
3AH	Move To Entered Pan/Tilt/Zoom/Focus	83H	Align Angles To Coordinates
		84H	Clear Angle Correction
40H	Retrieve Preset Table Entry	85H	Get Pan & Tilt Angle Correction
41H	Save Coordinates As Preset Table Entry		
42H	Save Current Position As Preset Table Entry	90H	Get Center Position in RU's
43H	Add Current Z/F To Preset Table Entry	91H	Set Center Position
		92H	Get/Set Pan and Tilt Ramp Parameters
50H	Flush Preset Tour	94H	Initialize Preset Table to 0/0
51H	Query Preset Tour	95H	Get/Set Motor and Resolver Direction
52H	Append To Preset Tour	96H	Get/Set Communication Timeout
53H	Insert Into Preset Tour	97H	Get/Set Heater Power Sharing
54H	Delete From Preset Tour	9CH	Get/Set/Store Maximum Speed
55H	Replace In Preset Tour	9DH	Initial Encoder Align
56H	Get Tour Size	9EH	Perform Homing Cycle
		9FH	Get/Set Identity Address
60H	Get/Set Camera Comm Parameters		
61H	Get/Set Lens Parameters	A1H	Get/Set Pan Step Rate Parameters
62H	Command Selected Camera	A2H	Get/Set Tilt Step Rate Parameters

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.

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

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 bitset 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 or setting jog speed to a value other than 0 in any following "Get Status/Jog" command will immediately terminate any automated "Move To" operation. Sending any command other than the 31H "Get Status/Jog" command during an automated move will also terminate the automated "Move To" command.

2.3.1 “Move To Preset” Command:

The PTCR can retain up to 32 position entries in a non-volatile table that are frequently used by the operator. 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 below. If a preset number greater than 31 is entered no move will occur.

Data	Format	Bytes	
Command	32H	1	
Preset	xxH	1	Preset Number 0-31

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°.

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

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.

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°

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

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.

Data	Format	Bytes
Command	36H	1

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.

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

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

2.4 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.

Data	Format	Bytes	
Command	40H	1	
Preset Num	xxH	1	0-31 (0-1FH)

Data	Format	Bytes	
Command	40H	1	
Preset Num	xxH	1	0-31 (0-1FH) 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.

Data	Format	Bytes	
Command	41H	1	
Preset Num	xxH	1	0-31 (0-1FH)
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°

Data	Format	Bytes	
Command	41H	1	
Preset Num	xxH	1	0-31 (0-1FH) 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.

Data	Format	Bytes	
Command	42H	1	
Preset Num	xxH	1	0-31 (0-1FH)

Data	Format	Bytes	
Command	42H	1	
Preset Num	xxH	1	0-31 (0-1FH) 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.

Data	Format	Bytes	
Command	43H	1	
Preset Num	xxH	1	0-31 (0-1FH)

Data	Format	Bytes	
Command	43H	1	
Preset Num	xxH	1	0-31 (0-1FH) 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.

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.

Data	Format	Bytes	
Command	37H	1	
Tour Num	xxH	1	0-2

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	37H	1								
Tour Num	xxH	1	0-2 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.

Data	Format	Bytes	
Command	50H	1	
Tour Num	xxH	1	0-2

Data	Format	Bytes	
Command	50H	1	
Tour Num	xxH	1	0-2

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.

Data	Format	Bytes

Command	51H	1	
Tour Num	xxH	1	0-2
Step Num	xxH	1	0-62 (0-3EH)

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	51H	1								
Tour Num	xxH	1	0-2							
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.

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

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	52H	1								
Tour Num	xxH	1	0-2							
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.

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	53H	1								
Tour Num	xxH	1	0-2							
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						

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	53H	1								
Tour Num	xxH	1	0-2							
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.

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

Data	Format	Bytes	
Command	54H	1	
Tour Num	xxH	1	0-2
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.

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	55H	1								
Tour Num	xxH	1	0-2							
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						

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	55H	1								
Tour Num	xxH	1	0-2							
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.

Data	Format	Bytes	
Command	56H	1	
Tour Num	xxH	1	0-2

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

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.

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

[1]Camera 1/2 0 = RS-232/1 = RS-422
[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

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

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

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.

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

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

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

This command performs configuration of the camera video selection and power on or off for each camera. If the baud parameters have not been initialized (see Command 60H) it is possible the serial line driver could connect an RS-232 level signal to a camera expecting an RS-422 level signal or vice versa. It is recommended that the cameras not be connected until these levels are set properly via Command 60H.

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.

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	63H	1								
Data	xxH	1	Query	STOR	0	0	[1]C2	[1]C1	[2]CV2	[2]CV1

[1]1 = Power On/0 = Power Off for Cameras 1/2
[2]1 = Camera 1/2 Video Select or both 0 = Video Blanked

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	63H	1								
Data	xxH	1	0	STOR	0	0	C2	C1	CV2	CV1

2.6.5 "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.

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)						

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)						

2.6.6 "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.

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

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 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.

2.7.1 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.**

2.7.2 "Get Pan & Tilt Angle Correction" Command:

Data	Format	Bytes
Command	85H	1

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°

2.7.3 "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.

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°

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°

2.7.4 "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.

Data	Format	Bytes
Command	82H	1

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°

2.7.5 "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.

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°

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°

2.7.6 "Clear Angle Corrections" Command:

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

Data	Format	Bytes
Command	84H	1

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

2.7.7 "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.

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						

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 PTCR Initial Setup Parameters:

2.8.1 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.)

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

Data	Format	Bytes
Command	90H	1

Data	Format	Bytes	
Command	90H	1	
Pan Count	Int	3	PAN = 0
Tilt Count	Int	3	TILT = 0

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

Data	Format	Bytes
Command	91H	1

Data	Format	Bytes	
Command	91H	1	
Pan Count	Int	3	PAN = 0
Tilt Count	Int	3	TILT = 0

2.8.4 "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. Conversly, 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.

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						

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						

2.8.5 "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.

Data	Format	Bytes
Command	94H	1

Data	Format	Bytes
Command	94H	1

2.8.6 "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 miswiring 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.

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

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	95H	1								
Pan/Tilt	xxH	1	0	0	0	0	TRES	PRES	TMTR	PMTR

2.8.7 "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. Option 0 will prohibit heater operation, reducing overall current draw. Option 1 and 2 will automatically cycle the heater as dictated by the on-board thermal sensor. However, in option 1 mode, a heater that is on will be turned off whenever the axis motors are operating, then be powered back on once motion has stopped. This limits instantaneous current draw. Option 2 will operate the heater as required concurrently with motor operation.

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						

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						

2.8.8 "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.

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								

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								

2.8.9 "Get/Set/Store Maximum Speed" Command:

The PTCS-20 operates within the speed ranges defined by the factory set minimum speed and the factory set maximum speed. As defined in the "Get/Set Pan & Tilt Ramp Parameters" command, the range of speed is divided into 255 steps from minimum to maximum speed and each step speed can be calculated. The platform can be moved at any of the speeds using the jog command but, by default, an automated movement will peak at maximum speed in both axes. Using this command, the user may limit the maximum speed in either or both axes for automated moves. This includes moves to defined coordinates, moves to presets, and moves executed during a tour.

Setting the Query bit will return the current volatile values for maximum speed. Clearing the Query bit will write the sent values into the volatile variables. Setting the STOR and Query bits will return the user-set default values. Setting the STOR bit and clearing Query bit will write the values to non-volatile memory. When the unit is powered on these values will be automatically loaded.

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

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

2.8.10 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, unstored 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.

2.8.11 "Initial Encoder Align" Command:

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

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	9DH	1								
Pan Index	Int	3	PAN = -18000 to +18000 = -180.00° to +180.00°							
Tilt Index	Int	3	TILT = -9000 to +9000 = -90.00° to +90.00°							

2.8.12 "Perform Homing Cycle" Command:

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

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

2.8.13 "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.

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						

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)						

2.9 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.**

2.9.1 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. Minimum speed cannot be less than 15 Hz. Maximum speed cannot be more than 30,000 Hz. Pan and tilt speed ranges can be set completely independent of each other. 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].

Command A1H is used for pan. Command A2H is used for tilt. 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 array. The status of the CON continuous rotation bit will actually be returned in the standard 31H status response. The 31H command should be executed if the user needs to check the continuous/non-continuous rotation status. The CON bit will be ignored if the Query bit is set.

2.9.2 "Get/Set Pan Step Rate Parameters" 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]CON
Pan Min	Int	2	>= 15 Hz							
Pan Max	Int	2	<= 30,000 Hz							

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

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	A1H	1								
Pan Min	Int	2	>= 15 Hz							
Pan Max	Int	2	<= 30,000 Hz							

2.9.3 "Get/Set Tilt Step Rate Parameters" 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	0
Tilt Min	Int	2	>= 15 Hz							
Tilt Max	Int	2	<= 30,000 Hz							

Data	Format	Bytes	7	6	5	4	3	2	1	0
Command	A2H	1								
Tilt Min	Int	2	>= 15 Hz							
Tilt Max	Int	2	<= 30,000 Hz							

2.9.4 Configuring Resolver Parameters

The PTCR-95N controller is designed for a platform fitted with 9000 line encoders for pan and for tilt. Therefore, scaling is fixed and resolver selection is encoder only.

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 A 06/02/05

Initial Release

4.2 Rev B 06/21/05

Initial Release

4.3 Rev C 02/23/06

Change in accel/decel algorithm. See command 92H.

4.4 Rev N1 04/11/06

This protocol is for initial development of the PTCR-96 protocol. All angles will be returned as 24-bit values providing resolution to 1/100th degree. The unit will be used with ONLY a 9000 line encoder providing 1 step per .01 deg when 4X decoded. Therefore, all variables and commands for scaling are removed.

4.5 Rev N2 05/02/06

Change in command 31H to allow transmission of full 0-255 jog speeds for each axis. Jog direction has now been moved up into the Command bitset. Separate “Jog Slow” bits have also been provided for each axis. Setting the “Jog Slow” bit for an axis will reduce the speed by a factor of 8, not 64 as in past revisions.

The minimum stepper frequency settable via commands A1H/A2H has been reduced from 115 Hz to 15 Hz, an 8 fold drop. This will allow low end speed control in jog similar to the previous divide-by-64 "Jog Slow" mode but with better top end speed when in "Jog Slow" mode.

Step rate calculation has been modified to create a linear speed progression from step 1, then from step 9 to step 255. Steps 2-8 are a linear progression between step 1 and step 9 and provide the user with a wider range of very low end jog speeds. Since automated moves always start from speeds much higher than the very low end jog speeds this lack of linearity at the bottom end will never be seen when accelerating or decelerating. Information on the updated algorithm has been included for both command 92H and command A1H/A2H.

All commands that return angles have been changed to show 3 bytes rather than 2 for each returned angle.

Updated algorithms have also been provided for accel/decel and move-to ramp down rates for command 92H.

Return data for command 9EH "Homing Cycle" has been altered. The 9DH "Initial Align" and 9EH "Homing Cycle" procedures will execute much more quickly than previous versions. **The user must confirm prior to issuing the commands that hard limits are all set and detected properly, that motor/resolver normal/reverse settings are correct, and that the platform has been correctly set for continuous or non-continuous pan operation. Otherwise, damage may occur to the platform.**

4.6 Rev N3 01/18/08

Clean up. Added command 3A, "Move To Entered Pan/Tilt/Zoom/Focus" Command, section 2.3.6 .

4.7 Rev N4 05/20/09

Added command 9CH to allow setting of a maximum speed for each axis when executing an automated move. The value represents which step rate between 1-255 is used when any automated move is executed, be it a "move to" or tour execution. Each axis can be set separately and the value sent can be either used immediately (RAM) or stored as a default power-up value (FRAM).