



RobotEye UDP Communications Specification V3.5

UDP Packet descriptions for communicating
with an Ocular Robotics RobotEye

Ocular Robotics

28 March 2025

Revision Table

Date	Notes
11/11/2015	• Version 3.0 initial public release
14/2/2017	• Updated RRBE command with new parameter/value pairs
30/11/2018	• Added RESEAT command • Added RETTRG command • Added EBHPTG broadcast packet
29/1/2021	• Added RESVEL and REGVEL commands
4/11/2024	• Updated RERRBE, RESRBE & ERBBEP for universal range, bearing, elevation sensor use.
28/3/2025	• Added REIIOG, REIIOS, REIIOT, REIOSE and REIOGE commands • Added REEENB, REESTS, REEDIS, REECCF, REERST, REERTM, REESTM and REEGTM commands

Contents

Revision Table.....	i
Basic Specification:	1
Eye Model Limitations.....	1
Common Arguments:	2
Standard Angle Representation.....	2
Conversion Factors	2
Useful Identities	2
Standard Speed Representation	2
Error Codes:.....	3
General Command Set:	4
CQRE – Seek Robot Eyes	4
SIPC – Set IP Configuration	4
Basic Motion Control Set:.....	5
HOME – Home Robot Eye	5
STOP – Stop Robot Eye	5
SEAC – Set Acceleration Limit	5
GEAA – Get Aperture Angles	5
SEAA – Set Aperture Angles.....	6
TRAA – Track Aperture Angles	6
Scan Pattern Commands:	7
SFFS – Set Full Field Scan	7
SBES – Set Bounded Elevation Scan.....	7
SRES – Set Region Scan	8
Laser Management Commands:	9
RRBE – Run Range-Bearing-Elevation Sensor	9
SRBE – Stop Range-Bearing-Elevation Sensor	9
Stabilisation Opcodes – REXXX-ST ONLY:	10
SRUN – Run Stabilisation	10
SSTP – Stop Stabilisation	10
SSTG – Stabilised Set Target	10
SGTG – Stabilised Get Target	10
SGRL – Stabilised Get Roll	11
SSSP – Stabilised Set Speed	11

High Performance Trigger Commands.....	12
SEAT – Set Aperture Angles with Trigger.....	12
TTRG – Test High Performance Trigger.....	12
Velocity Control Mode commands.....	13
SVEL – Set Aperture Velocity	13
GVEL – Get Aperture Velocity.....	13
Eye Broadcast Packets.....	14
RBEP – Eye Broadcast Range Bearing Elevation Packet.....	14
HPTG – Eye Broadcast High Performance Trigger Metadata.....	14
Isolated I/O Commands.....	15
IIOG – Get Isolated I/O.....	15
IIOS – Set Isolated I/O	15
IIOT – Toggle Isolated I/O	15
IOGE – Get Isolated I/O Capture.....	16
Event Manager Commands.....	17
EENB – Event Enable	17
ESTS – Get Event Status	17
EDIS – Disable Event	17
ERST – Reset Event Manager	17
ECCF – Configure Action Command.....	18
ERTM – Run Timer	18
ESTM – Stop Timer.....	18
EGTM – Get Timer.....	18

Basic Specification:

Interface to the Robot Eye is through UDP datagrams. The Robot Eye will only ever speak when spoken to, with the exception of specific, defined 'broadcast' packets. It will respond either to the source port and IP from which the packet originated, or in a broadcast packet depending upon the instruction. At present, an IANA port allocation request is pending for the designated port number 4365, and this is the only port on which the eye will respond to commands. The eye will only respond to unicast UDP packets with a limited number of exceptions documented on the specific opcodes.

All communication with the Robot Eye begins with a 6-character ASCII human-readable opcode. The first two characters of a transaction define the type of transaction sent to the Robot Eye as follows (where <XXXX> indicates a 4-character ASCII opcode and <YYYY> indicates a 4-character data type identifier):

- RE<XXXX>: Robot Eye <XXXX> – Command from a host to a Robot Eye
- EA<XXXX>: Eye Acknowledge <XXXX> - Acknowledgement of the receipt/completion of command <XXXX>
- ER<XXXX>: Eye Response <XXXX> - Response to a one-off request for information
- EE<XXXX>: Eye Error <XXXX> - Response to an erroneous command, followed by a single-byte binary error code.
- EB<YYYY>: Eye Broadcast <YYYY> - Unsolicited broadcast of data of specified type.

EB, EE and ER transactions **shall** be followed by one or more arguments in a binary format. RE transactions **may** have one or more arguments passed in a binary format. EA transactions are **never** followed by arguments.

Eye Model Limitations

Not all commands are supported by all Ocular Robotics products. Unless otherwise documented, commands are supported by all robot eye products. Unexpected behaviours may occur if unsupported commands are executed on an Ocular Robotics product. The following is a brief list of single-model commands within this document:

REV25 (and variants) ONLY	RE05, RE08 ONLY
TRAA – Track Aperture Angles	RRBE – Run Range-Bearing-Elevation
	SRBE – Stop Range-Bearing-Elevation

REXXX-ST ONLY	
SRUN – Start Stabilising	
SSTP – Stop Stabilising	
SSTG – Stabilised Set Target	
SGTG – Stabilised Get Target	
SGRL – Stabilised Get Roll	
SSSP – Stabilised Set Speed	

Common Arguments:

NOTE: All arguments are written as [NAME-SIZE] where size is the size of the argument in bytes, all integer arguments are big-endian

Standard Angle Representation

All angles sent to and received from the RobotEye use a single standard representation. This is a 32-bit 0.32 fixed point notation, representing the angle as a fraction of a full revolution. Appropriate conversion factors and some useful identities are shown below.

Conversion Factors

	From			
To	-----	Degrees	Radians	0.32 Fixed Point
Degrees		$\theta_{DEG} = \theta_{DEG}$	$\theta_{DEG} = \theta_{RAD} \times \frac{180}{\pi}$	$\theta_{DEG} = \theta_{0.32} \times \frac{360}{2^{32}}$
Radians		$\theta_{RAD} = \theta_{DEG} \times \frac{\pi}{180}$	$\theta_{RAD} = \theta_{RAD}$	$\theta_{RAD} = \theta_{0.32} \times \frac{2\pi}{2^{32}}$
0.32 Fixed Point		$\theta_{0.32} = \theta_{DEG} \times \frac{2^{32}}{360}$	$\theta_{0.32} = \theta_{RAD} \times \frac{2^{32}}{2\pi}$	$\theta_{0.32} = \theta_{0.32}$

Note that all 0.32 representation results should be truncated to 32-bits, discarding any higher bits.

Useful Identities

In the 0.32 representation, the following identities hold:

- $0^\circ == 360^\circ == 720^\circ == -360^\circ$
- $+180^\circ == -180^\circ$

The essence of these identities is that an integer overflow in the representation represents a single complete revolution, which can be concisely expressed as follows:

$$\theta_{0.32} == (\theta \pm n \times 360^\circ)_{0.32} \quad \text{for any integer } n$$

It should be noted that the choice of signed or unsigned representation of the 0.32 angle for the conversion back to degrees or radians will determine whether the resultant angle is in the range $0-360^\circ$ or $\pm 180^\circ$

Standard Speed Representation

Any commands to the RobotEye requiring a speed argument use a 32-bit 16.16 fixed point notation in units of revolutions/second. Appropriate conversion factors are shown below:

	From			
To	-----	Degrees/Second	Radians/Second	16.16 Fixed Point
Degrees/Second		$\dot{\theta}_{DEG} = \dot{\theta}_{DEG}$	$\dot{\theta}_{DEG} = \dot{\theta}_{RAD} \times \frac{180}{\pi}$	$\dot{\theta}_{DEG} = \dot{\theta}_{16.16} \times \frac{360}{2^{16}}$
Radians/Second		$\dot{\theta}_{RAD} = \dot{\theta}_{DEG} \times \frac{\pi}{180}$	$\dot{\theta}_{RAD} = \dot{\theta}_{RAD}$	$\dot{\theta}_{RAD} = \dot{\theta}_{16.16} \times \frac{2\pi}{2^{16}}$
16.16 Fixed Point		$\dot{\theta}_{16.16} = \dot{\theta}_{DEG} \times \frac{2^{16}}{360}$	$\dot{\theta}_{16.16} = \dot{\theta}_{RAD} \times \frac{2^{16}}{2\pi}$	$\dot{\theta}_{16.16} = \dot{\theta}_{16.16}$

Error Codes:

Error codes are represented as a single 8-bit unsigned integer transmitted following an EA<XXXX> transaction, shown here in hexadecimal representation:

0x00 *No Error.*

0x01 *Invalid Argument Length*

A command was sent with the incorrect number of associated arguments. Test is performed on the number of bytes in the UDP packet in addition to the opcode, so typically indicates a poorly constructed UDP datagram.

0x02 *Argument Out Of Range*

One or more arguments passed in has exceeded the valid range for that argument

0x03 *Error Not Ready*

The eye is not in an appropriate state for the desired command.

0x04 *Error Not Homed*

The command requires the eye to have been homed since startup and this has not taken place.

0x05 *Error Invalid Argument*

The command specified an invalid argument such as an invalid axis.

0x06 *Error Unknown Command*

The command is not a recognised Robot Eye opcode.

0x07 *Error Unsupported Command*

The command transmitted is not compatible with this Robot Eye product.

0x08 *Error Scan too Sparse*

The requested scan pattern has too few lines per scan. See eye model documentation for details on the scan density limitations of each eye model.

0x09 *Error Busy*

The eye is currently executing another command which prevents the requested command from executing.

0x0A *Error Bad Flash Page*

The requested user flash page is not available for the requested operation.

0x0B *Error Bad Flash Key*

The requested user flash page operation could not be performed, as the incorrect key has been written for the requested page and operation.

0x0C *Error Stabilisation Running*

The requested command is not available while stabilisation is active.

0xFF *Other Error*

General Command Set:

CQRE – Seek Robot Eyes

Command: **RECQRE**

Response: **ERCQRE**[Serial-N]

[Serial]: Robot-eye serial number, an ASCII encoded character string of variable length.

Notes: *This eye will respond to this command when received in a broadcast packet. This command is intended for use as a broadcast packet for eye discovery. All eyes on the network will respond with their associated serial number and IP address.*

SIPC – Set IP Configuration

Command: **RESIPC**[Serial-N][IP-4]

[Serial]: Robot-eye serial number, specifying the eye whose IP is to be set.

[IP]: Desired new Robot Eye IP address, sent as 4 8-bit integers.

Response: **EASIPC**

Notes: *This eye will respond to this command when received in a broadcast packet. This command enables the IP address of an eye to be specified. The serial number argument allows this packet to be sent as a broadcast without affecting more than the specified eye. Only the eye with the matching serial number will respond to this instruction, and the response will be transmitted from the old IP address immediately before transitioning to the new address.*

Basic Motion Control Set:

HOME – Home Robot Eye

Command: **REHOME**
Response: **EAHOME**
Errors: **EEHOME{0x01}**
Notes: This command will cause the eye to execute its homing routine. Typically performed at start-up, this routine enables the eye to identify the index locations and correctly localise its pointing direction. This command will cause the eye to point in the direction (0,0); the acknowledgement will be transmitted when the homing is complete.
If stabilisation is currently running, this command will stop stabilisation.

STOP – Stop Robot Eye

Command: **RESTOP**
Response: **EASTOP**
Errors: **EESTOP{0x01}**
Notes: This command will cause the eye to stop any currently executing scan or non-blocking motion (TRAA).
If stabilisation is currently running, this command will stop stabilisation.

SEAC – Set Acceleration Limit

Command: **RESEAC**[Accel-4]
[Accel]: Desired acceleration limit in revolutions/second². This parameter uses the same 16.16 fixed point notation as the speed parameter and the same conversions described on page 2.
Response: **EASEAC**
Errors: **ESEAC{0x01, 0x02}**
Notes: This command will set the acceleration limit to be used for any future motion commands.

GEAA – Get Aperture Angles

Command: **REGEAA**
Response: **ERGEAA**[Az-4][El-4]
[Az]: Current Eye Azimuth
[El]: Current Eye Elevation
Errors: **EEGEAA{0x01}**
Notes: Due to network and communications latency, this method is not recommended for high-frequency position measurements.

SEAA – Set Aperture Angles

Command:	RESEAA [Az-4][El-4][Spd-4] [Az]: Desired Eye Azimuth [El]: Desired Eye Elevation [Spd]: Movement speed limit
Response:	EASEAA This acknowledgement packet will be sent when the eye has completed its move and settled at the commanded position.
Errors:	EESEAA {0x01, 0x02, 0x03, 0x0C}
Notes:	The eye will respond to other commands which interrupt an SEAA move-in-progress, and if another move command is given, no acknowledgement for the previously executing move will be sent. The movement speed limit is an upper limit only. The peak attainable velocity is determined by the acceleration limits applied. Any value above this peak attainable velocity will guarantee a triangular velocity profile over the move, with the time taken and peak velocity determined by the acceleration limit. This command disabled while stabilisation is active in REV25-ST systems. Attempted execution will result in error code 0x0C.

TRAA – Track Aperture Angles

Command:	RETRAA [Az-4][El-4][Spd-4] [Az]: Desired Eye Azimuth [El]: Desired Eye Elevation [Spd]: Movement speed limit
Response:	EATRAA This acknowledgement packet will be sent immediately if the arguments are well-posed for the command. There is no acknowledgement when the eye has reached the desired position.
Errors:	EETRAA {0x01, 0x02, 0x03, 0x0C}
Notes:	This command is designed for small, high-frequency commanded position updates, such as those required to make small adjustments for target tracking or stabilisation applications. <i>It should not be used for large moves</i> , and has a longer settling time than an equivalent SEAA command. The movement speed limit is an upper limit only. The peak attainable velocity is determined by the acceleration limits applied. Any value above this peak attainable velocity will guarantee a triangular velocity profile over the move, with the time taken and peak velocity determined by the acceleration limit. This command disabled while stabilisation is active in REV25-ST systems. Attempted execution will result in error code 0x0C.

Scan Pattern Commands:

SFFS – Set Full Field Scan

Command: **RESFFS** [Speed-4][NLines-4]
[Speed]: Scan Azimuth rate
[NLines]: Number of lines within scan pattern. This is represented in a 32-bit 16.16 fixed point representation. Non-integer scan line counts will result in an interleaved scan. Target NLines can be converted to 16.16 format by multiplying by 2^{16}

Response: **EASFFS**
Errors: **EESFFS**{0x01, 0x02, 0x08, 0x0C}
Notes: This command will cause the eye to begin executing a full-field scan. Any existing scan pattern will be stopped. The scan will start from minimum elevation and will transition from the current location to that elevation over 3 rotations of the head in azimuth. Note that if the eye has not been homed, results may not be as expected. This command disabled while stabilisation is active in REV25-ST systems. Attempted execution will result in error code 0x0C.

SBES – Set Bounded Elevation Scan

Command: **RESBES** [Speed-4][ElMin-4][ElMax-4][NLines-4]
[Speed]: Scan Azimuth rate in degrees/second.
[ElMin]: Lower Elevation Limit
[ElMax]: Upper Elevation Limit
[NLines]: Number of lines within scan pattern. This is represented in a 32-bit 16.16 fixed point representation. Non-integer scan line counts will result in an interleaved scan. Target NLines can be converted to 16.16 format by multiplying by 2^{16}

Response: **EASBES**
Errors: **EESBES**{0x01, 0x02, 0x05, 0x08, 0x0C}
Notes: This command will cause the eye to begin executing a bounded elevation scan. Any existing scan pattern will be stopped. The scan will start from minimum elevation and will transition from the current location to that elevation over 3 rotations of the head in azimuth. Note that if the eye has not been homed, results may not be as expected. This command disabled while stabilisation is active in REV25-ST systems. Attempted execution will result in error code 0x0C.

SRES – Set Region Scan

Command:	RESRES [AzMin-4][AzMax-4][ElMin-4][ElMax-4][Speed-4][NLines-4] [AzMin]: Lower Azimuth Limit [AzMax]: Upper Azimuth Limit [ElMin]: Lower Elevation Limit [ElMax]: Upper Elevation Limit [Speed]: Scan Azimuth rate limit in degrees/second. [NLines]: Number of lines within scan pattern. This is represented in a 32-bit 16.16 fixed point representation. Non-integer components of the NLines argument will presently be discarded, but may be implemented in future firmware revisions. Target NLines can be converted to 16.16 format by multiplying by 2 ¹⁶
Response:	EASRES
Errors:	EESRES {0x01, 0x02, 0x05, 0x08, 0x0C}
Notes:	This command will cause the eye to begin executing a region scan at the specified resolution within the specified bounds. The region scan shall take the shortest path between the upper and lower azimuth limits, giving a maximum azimuth width of 180 degrees. Any existing scan pattern will be stopped. The scan will start from minimum azimuth elevation and will transition from the current location to that location immediately respecting acceleration limits. Note that there may be some overshoot if the head is currently scanning at high speed due to acceleration limitations. The azimuth rate limit is only an upper bound. For many scan patterns, the peak acceleration will be the limiting factor with respect to the line rate. For more detail on the relationships between acceleration limits and line rates, see the ACRA 2012 paper “A Novel Approach to 3D Laser Scanning”, available from the ocular robotics website. This command disabled while stabilisation is active in REV25-ST systems. Attempted execution will result in error code 0x0C.

Laser Management Commands:

RRBE – Run Range-Bearing-Elevation Sensor

Command: **RERRBE**{[Param1-1][Value1-4][Param2-1][Value2-4]...[ParamN-1][ValueN-4]}

This command takes a set of Parameter+Value pairs as arguments. Any parameters not included will be left to device defaults. The number of parameter-value pairs can be 0 if all parameters are desired to be default.

RE05

Parameter		Valid Value Range	Description
PORT	0x00	0 : 65535	Target UDP port for data streaming. (Default 4366)
MEASUREMENT FREQUENCY	0x03	0 : 10000	Standard Ranging Mode
		30000	Fast Ranging Mode
MEASUREMENT AVERAGING	0x04	1 : 30000	Measurements to average per output range
GAIN	0x05	0:3	Laser receiver gain
BROADCAST	0x06	0:1	Broadcast 1; Unicast 0

RE08

Parameter		Valid Value Range	Description
PORT	0x00	0 : 65535	Target UDP port for data streaming. (Default 4366)
MODE	0x01	0 : 2	0x00 - High Penetration
			0x01 - Fast Mode
			0x02 - High Speed Mode (Default)
MULTIPLE PULSE BEHAVIOUR	0x02	0 : 4	0x00 - All Targets (Default)
			0x01 - First Target Only
			0x02 - Last Target Only
			0x03 - Highest Amplitude Target Only
			0x04 - Highest Reflectance Target Only

Response: **ERRRBE**

Errors: **EERRBE**{0x01, 0x02, 0x04, 0x05, 0xFF}

Notes: This command will begin the laser fusion process. The laser will be sampled at the specified frequency, packets of multiple laser returns will be constructed, and these shall be transmitted as broadcast packets at the port number specified in the argument. For details on the broadcast packet, see the documentation for EBRBEP (Eye Broadcast Range Bearing Elevation Points).

If the Robot Eye is presently streaming data, the Robot Eye will return an error code of 0x09 indicating that it is busy. If the new client wishes to disconnect the currently streaming client, then this can be accomplished by sending the SRBE opcode to stop the running fusion process before restarting the streaming with RRBE.

SRBE – Stop Range-Bearing-Elevation Sensor

Command: **RSRBE**

Response: **EASTLS**

Notes: This command will halt the RE08 laser data streaming. Any buffered laser data will be discarded when this command is received. This command will stop ANY running streaming, and could potentially cause the disconnection of another connected client.

Stabilisation Opcodes – REXXX-ST ONLY:

SRUN – Run Stabilisation

Command: **RESRUN**
Response: **EASRUN**
Errors: **EESRUN**{0x04, 0x07}
Notes: This command will activate stabilisation on the target REV25-ST. The aperture pointing direction in the world-frame when this command is received is taken as the current stabilised target. To make incremental movements, the SRUN command should be followed by a SGTG command to extract the current pointing position of the eye in the stabilised world co-ordinate frame. This will also enable checking of the state of stabilisation, as until the stabilisation startup has completed, the SGTG command will return error code 0x03 = UDP_ERROR_NOT_READY.

SSTP – Stop Stabilisation

Command: **RESSTP**
Response: **EASSTP**
Errors: **EESTOP**{0x07}
Notes: This command will stop any running stabilisation.

SSTG – Stabilised Set Target

Command: **RESSTG**[Yaw-4][Pitch-4][Roll-4]
[Yaw-4]: The world-frame yaw between the target location and world-frame north.
[Pitch-4]: The world-frame pitch between the target location and the horizon.
[Roll-4]: The world-frame roll of the target about the target vector.
Response: **EASSTG**
Errors: **EESSTG**{0x01, 0x02, 0x03, 0x04, 0x07}
Notes: When working in stabilised mode, all commands are managed in the world reference frame. The angles transmitted with an SSTG command represent the required co-ordinate frame transformation which, when applied to the world-frame, will result in a reference frame with the X-axis directed at the target, the Y-axis pointing to the right in the image, and the Z-axis pointing down.

SGTG – Stabilised Get Target

Command: **RESGTG**
Response: **ERSGTG**[Yaw-2][Pitch-2][Roll-2]
[Yaw-4]: The world-frame yaw between the target location and world-frame north.
[Pitch-4]: The world-frame pitch between the target location and the horizon.
[Roll-4]: The world-frame roll of the target about the target vector.
Errors: **EESGTG**{0x03, 0x07}
Notes: See SSTG for explanation of the stabilisation reference frame.
Be aware that when working in the Euler angle space, the problem of ‘gimbal lock’ can become present for commanded pitch angles close to $\pm 90^\circ$. Near these pitch angles, the values returned by SGTG may become unreliable, however accurate and reliable pointing commands may still be issued using the SSTG command.

SGRL – Stabilised Get Roll

Command: **RESGRL**

Response: **ERSGRL**[ImageRoll-4]

[ImageRoll-4]: The roll angle required to be applied to an image taken at the instant the request was made in order for that image to align with the reference frame as defined in the SSTG command.

Errors: **EESGRL**{0x03, 0x07}

Notes: This command will extract the roll transform which, when applied to images from the camera attached to the REV25-ST, provides for the third axis of stabilisation. It is not necessarily required that the image be rotated according to this angle, if computer-vision based target tracking or other algorithms are being applied to the output images, then it will almost certainly be faster to deal with the image rotation after the target tracks have been extracted, rather than applying an affine transform to the entire image.

SSSP – Stabilised Set Speed

Command: **RESSSP**[Speed-4]

[Speed-2]: The maximum speed the eye is allowed to move at in degrees-per-second.

Response: **EASSSP**

Errors: **EASSSP**{0x01, 0x02, 0x03, 0x04, 0x07}

Notes: This command will set the motion speed limit used for the stabilisation. In most situations this parameter will not need to be changed from its default.

High Performance Trigger Commands

SEAT – Set Aperture Angles with Trigger

Command:	RESEAT [Az-4][El-4][Spd-4][Hld-4][Exp-4] [Az]: Desired Eye Azimuth [El]: Desired Eye Elevation [Spd]: Movement speed limit [Hld]: Hold-off time before triggering. 32-bit unsigned integer in microseconds. [Exp]: Pulse-width of trigger pulse. 32-bit unsigned integer in microseconds.
Response:	EASEAT This acknowledgement packet will be sent when the eye has completed its move and settled at the commanded position.
Errors:	ESEAT {0x01, 0x02, 0x03, 0x07, 0x0C}
Notes:	This command will cause the robot eye to execute an SEAA move followed by generating a trigger pulse on the high-performance trigger line. For details on the motion pattern, please see entry for SEAA – Set Aperture Angles on Page 6. The trigger pulse will be generated with a rising edge [Hld] microseconds after the SEAA motion complete event has occurred, and with a pulse-width of [Exp] microseconds. The EASEAT acknowledgement packet will be sent at the completion of the motion pattern. The completion of the exposure will be notified with the broadcast of an EBHPTG packet, which contains all the metadata associated with the exposure which has just taken place. The precision of the HPT is to within 0.03% of the total time, i.e. the sum of hold-off and exposure. This may lead to unexpected results if, for example, a holdoff of the order of milliseconds is commanded followed by an exposure of single-digit microseconds.

TTRG – Test High Performance Trigger

Command:	RETTRG [Hld-4][Exp-4] [Hld]: Hold-off time before triggering. 32-bit unsigned integer in microseconds. [Exp]: Pulse-width of trigger pulse. 32-bit unsigned integer in microseconds.
Response:	EATTRG This acknowledgement packet will be sent upon receipt of the TTRG command, and does not indicate completion of the desired exposure.
Errors:	EETTRG {0x07}
Notes:	As with the SEAT command, at the completion of the exposure an EBHPTG packet will be transmitted. The precision of the HPT is to within 0.03% of the total time, i.e. the sum of hold-off and exposure. This may lead to unexpected results if, for example, a holdoff of the order of milliseconds is commanded followed by an exposure of single-digit microseconds.

Velocity Control Mode commands

SVEL – Set Aperture Velocity

Command: **RESVEL** [AzVel-4][ElVel-4]
[Az]: Desired Eye Azimuth Velocity
[El]: Desired Eye Elevation Velocity

Response: **EASVEL**

Errors: **EESVEL**{0x02, 0x04, 0x07, 0x0C}

Notes: This command will cause the RobotEye to enter controlled velocity mode. In this mode, the aperture will move at the velocities specified in the packet. As the aperture approaches its limits of motion, the elevation velocity will be linearly reduced to prevent a hard-stop. The actual current elevation velocity can be obtained using the EAGVEL command. The commanded elevation velocity will only be reduced if it is towards the limit. When moving away from the limit of motion, it will be the full commanded value.

GVEL – Get Aperture Velocity

Command: **REGVEL**

Response: **ERGVEL**[AzVel-4][ElVel-4]

Errors: **EEGVEL**{0x03, 0x07, 0x0C}

Notes: This command will read the current velocity mode set-point from the RobotEye. It will include the elevation velocity scaling that occurs as the aperture approaches one of the end-stops.
Note that this will only return a velocity when using the Controlled Velocity Mode, it will return 0x03 if called when the eye is in any other mode.

Eye Broadcast Packets

RBEP – Eye Broadcast Range Bearing Elevation Packet

Header: **EBRBEP**

Payload **RE05**: {[tStamp-4] [Az-4] [El-4] [Range-4] [Intensity-2] [Reserved-2] [Reserved-2]} *repeated to end of packet.*

Payload **RE08**: {[tStamp-4] [Az-4] [El-4] [Range-4] [Amplitude-2] [Reflectance-2] [PSD-2]} *repeated to end of packet.*

[tStamp]:	The time at which the point was taken. Measured in 100's of nano-seconds since powerup. Will overflow approximately every 429.5 seconds
[Az]:	The azimuth at which the range sample was taken.
[El]:	The elevation at which the range sample was taken.
[Range]:	The range in millimeters as an unsigned 32-bit integer.
[Intensity]:	The signal strength (0-126) of the return
[Amplitude]:	The amplitude of the return as a 16-bit signed integer in units of 0.01 dB
[Reflectance]:	The reflectance of the return as a 16-bit signed integer in units of 0.01 dB
[PSD]:	The deviation in shape of the received pulse as compared to the transmitted pulse. 0 indicates no deviation, values below 15 can typically be considered as 'good' for most applications. For more detail on the meaning of the Amplitude, Reflectance and Pulse Shape Deviation parameters, please see the RE08 User's Manual.

HPTG – Eye Broadcast High Performance Trigger Metadata

Header: **EBHPTG**

Payload: [Count-4][tStamp-4][Az-4][El-4][Exp-4]

[Count]:	An integer count of the number of successful HPT trigger events which have occurred since powerup.
[tStamp]:	The 32-bit timestamp at which the exposure occurred. This is in units of ticks of a 66 MHz clock since powerup, and will therefore overflow approximately every 65 seconds.
[Az]:	The azimuth at which the exposure was taken.
[El]:	The elevation at which the exposure was taken.
[Exp]:	The exposure time for the image, in units of microseconds.

Notes: The precision of the HPT is to within 0.03% of the total time, i.e. the sum of hold-off and exposure. This may lead to unexpected results if, for example, a holdoff of the order of milliseconds is commanded followed by an exposure of single-digit microseconds.

Isolated I/O Commands

IIOG – Get Isolated I/O

Command: **REIIOG**
Response: **ERIIOG**[Outputs-1][Inputs-1]
[Outputs]: 8-bit bit-field specifying which output bits are currently enabled. Output1 is defined at 0x01, Output2 is defined at 0x02. Bits with value 1 have their associated output enabled. Bits with value 0 have their associated output disabled.
[Inputs]: 8-bit bit-field specifying the current state of the digital inputs. Input1 is defined at 0x01, Input2 is defined at 0x02. A value of 1 means that the associated input value is high. A value of 0 means the associated input value is low.
Errors: **EIIOG**{0x07}
Notes: This command enables the capture of the current state of the digital outputs, and the current state of the digital inputs.

IIOS – Set Isolated I/O

Command: **REIIOS**[Outputs-1]
[Outputs]: 8-bit bit-field specifying which value to apply to the output bits. Output1 is defined as 0x01, Output2 is defined as 0x02. Bits with value 1 will have their associated output set to high. Bits with value 0 will have their associated output set to low.
Response: **ERIIOS**
Errors: **EIIOS**{0x01, 0x07}
Notes: This command sets the desired state on the isolated I/O outputs.

IIOT – Toggle Isolated I/O

Command: **REIIOT**[ToggleBits-1]
[Outputs]: 8-bit bit-field specifying which output bits to toggle. Output1 is defined as 0x01, Output2 is defined as 0x02. Bits with value 1 will have their output state toggled. Bits with value 0 will remain unchanged.
Response: **ERIIOT**
Errors: **EIIOT**{0x01, 0x07}
Notes: This command causes the specified output bits to be toggled from their current state. Output channels whose bit is 0 in ToggleBits will remain unchanged.

IOSE – Set Isolated I/O Capture

Command: **REIOSE**[Inputs-1][Edge-1]
[Inputs]: 8-bit bit-field specifying which input to use as the event manager and capture trigger. Input0 is defined at 0x00, Input1 is defined at 0x01.
[Edge]: 1-byte specifying which edge or edges of the digital input to use as the capture trigger, and which edge or edges are capable of acting as an event source. 0x00 – No edge. Use to disable event and capture generation. 0x01 – Rising edge. Capture will occur on the rising edge of the input, and EVENTSOURCE_ISOLATEDIN_RISING event source will be triggered. 0x02 – Falling edge. Capture will occur on then falling edge of the input, and EVENTSOURCE_ISOLATEDIN_FALLING event source will be triggered. 0x03 – Both edges. Capture will occur on both the edges of the input, and both EVENTSOURCE_ISOLATEDIN_RISING and EVENTSOURCE_ISOLATEDIN FALLING event sources will be triggered.

Response: **ERIOSE**

Errors: **EEIOSE**{0x01, 0x07}

Notes: This command allows the configuration of the input event to trigger an event source for the event manager, and a capture of the aperture position on the specified edge. Note that the event manager has independent event for rising and falling edges on the selected isolated input. These event sources will only fire if the appropriate edge is enabled through this command.

IOGE – Get Isolated I/O Capture

Command: **REIOGE**

Response: **ERIOGE**[Input-1][Edge-1]
[Input]: 8-bit bit-field specifying which input to use as the event manager and capture trigger. Input0 is defined at 0x00, Input1 is defined at 0x01.
[Edge]: 1-byte specifying which edge or edges of the digital input are configured as the capture trigger, and which edge or edges are configured to act as an event source. 0x00 - No edge. Indicates event and capture generation are disabled . 0x01 - Rising edge. Configured for capture to occur on the rising edge of the input, and EVENTSOURCE_ISOLATEDIN_RISING event source to be triggered. 0x02 – Falling edge. Configured for capture to occur on the falling edge of the input, and EVENTSOURCE_ISOLATEDIN_FALLING event source to be triggered. 0x03 - Both edges. Configured for capture to occur on the both edges of the input, and both EVENTSOURCE_ISOLATEDIN_RISING, and EVENTSOURCE_ISOLATEDIN_FALLING event sources to be triggered.

Errors: **EEIOGE**{0x07}

Notes: This command allows to read the current configuration of the event input for the event manager

Event Manager Commands

The Event Manager API is a firmware subsystem designed to provide extremely responsive reactions to a range of input conditions. The Input condition could be a timer, external synchpulse or internal event such as movement complete. The action range from simple notification to capture of aperture angles or execution of pre-configured UDP commands. This command resets the event manager, disabling any existing events, and stopping the event manager's internal timer.

EENB – Event Enable

Command: **REIOSE**[Source-1][Action-1][Repeat-1]
[Source]: 1-byte unsigned integer. The desired source for this event.
[Action]: 1-byte unsigned integer. The desired action for this event.
[Repeat]: 1-byte unsigned integer. Whether or not to repeat this event.
Boolean represented as: 0 means one-shot, other values indicate repeat

Response: **ERIOSE**

Errors: **EEIOSE**{0x01}

Notes: This command enables the desired event. If one-shot mode is selected, the event will self-disable after a single source occurrence.

ESTS – Get Event Status

Command: **REESTS**[Source-1][Action-1]
[Source]: 1-byte unsigned integer. The event source.
[Action]: 1-byte unsigned integer. The event action.

Response: **ERESTS**[Status-1]
[Status]: 1-byte unsigned integer. 0x00 - EVENTMANAGER_EVENT_DISABLED, 0x01 - EVENTMANAGER_EVENT_ENABLED_ONESHOT, 0x02 - EVENTMANAGER_EVENT_ENABLED_REPEAT.

Errors: **EEESTS**{0x01, 0x05}

Notes: This command retrieves the current status of the event as a 1-byte unsigned integer

EDIS – Disable Event

Command: **REEDIS**[Source-1][Action-1]
[Source]: 1-byte unsigned integer. The event source.
[Action]: 1-byte unsigned integer. The event action.

Response: **EREDIS**

Errors: **EEEDIS**{0x01, 0x03, 0x05}

Notes: This command disables the specified event.

ERST – Reset Event Manager

Command: **REERST**

Response: **ERERST**

Errors: **EEERST**{0x03}

Notes: This command resets the event manager, disabling any existing events, and stopping the event manager's internal timer.

ECCF – Configure Action Command

Command: **REECF**[Buffer-1][Command_1-1][Command_2-1][Command_3-1]...[Command_n-1]
[Buffer]: The buffer to load, from 0x80 to 0x85
[Command_1]: 1-byte
[Command_2]: 1-byte
[Command_3]: 1-byte
⋮
[Command_n]: 1-byte

} The UDP command to load into the buffer, including opcode and all required arguments.

Response: **EECCF**
Errors: **EECCF**{0x03}
Notes: This command loads the specified buffer with the desired UDP command. There is no error checking of arguments to commands. There are two validity checks performed: Length of the command argument is at least 6 bytes(opcode length) First two characters of the command argument are RE(header for all RobotEye UDP commands)

ERTM – Run Timer

Command: **REERTM**[Frequency-2]
[Frequency]: 16-bits, The frequency of operation for the timer in Hz.
Response: **ERERTM**
Errors: **EEERTM**{0x01, 0x03}
Notes: This command starts the timer running at the desired frequency, or updates the currently running frequency if the timer is already running.

ESTM – Stop Timer

Command: **REESTM**
Response: **ERESTM**
Errors: **EEESTM**{0x03}
Notes: This command stops the timer if it's currently running.

EGTM – Get Timer

Command: **REEGTM**
Response: **REGTM**[Frequency-2]
[Frequency]: The frequency of operation for the timer in Hz, or 0 if it's not running.
Errors: **EEEGTM**{0x03}
Notes: This command obtains the frequency that the timer is running at.