Elliptec Thorlabs

ELLx Resonant Piezo Motor Communication Protocol

Date: July-2023

Contents

Introduction

1. Purpose and Scope

This document describes the low-level communications protocol and commands used between the host controller and OEM modules within the ELLx family. The information contained in this document is intended to help third party system developers to write their own applications to interface to the Elliptec Thorlabs range of OEM modules without the constraints of using a particular operating system or hardware platform. The commands described here are those which are necessary to control movement; there is an additional set of commands, used for calibration or test, which will not be detailed as these are not required for the external system developer.

2. Electrical interface

The ELLx family of OEM module provides a multidrop TTL RS-232 interface (point to point TTL RS232, and RS485 are available on request) to communicate with the host PC.
TTL digital lines are also available (INM for in motion state, JOG, FWD and BWD for motion).
TTL voltage digital levels are 0V or 3.3V, exceeding these values may damage modules.

2.1 USB Interface adapter

HOSTREQ_HOME "ho

The USB interface adapter uses a Future Technology Devices International (FTDI), type FT232BM USB peripheral chip to communicate with the host PC. This is a USB2.0 compliant USB1.1 device. This USB interfacing chip provides a serial port interface to the embedded system and USB interface to the host control PC.
FTDI supply device drivers and interfacing libraries (for Windows, Linux and other platforms) used to access the USB chip. Before any PC USB communication can be established with an ELLx module, the client program is required to set up the necessary FTDI chip serial port settings used to communicate to the APT controller embedded system.
Here an example for Windows on how to establish a communication via the USB remote control adapter (169892 Elliptec Demo Board).

```
// Set baud rate to 9600.
ftStatus = FT_SetBaudRate(m_hFTDevice, (ULONG)uBaudRate);

// 8 data bits, 1 stop bit, no parity
ftStatus = FT_SetDataCharacteristics(m_hFTDevice, FT_BITS_8, FT_STOP_BITS_1,
FT_PARITY_NONE);

// Pre purge dwell 50ms.
Sleep(uPrePurgeDwell);

// Purge the device.
ftStatus = FT_Purge(m_hFTDevice, FT_PURGE_RX | FT_PURGE_TX);

// Post purge dwell 50ms.
Sleep(uPostPurgeDwell);

// Reset device.
ftStatus = FT_ResetDevice(m_hFTDevice);

// No flow control
ftStatus = FT_SetFlowControl(m_hFTDevice, FT_FLOW_NONE, 0, 0);
```

2.2 USB Device Enumeration

Enumeration is made via RS232 interface, USB adapter acts just as electrical adapter.

2.3 TTL RS-232 Interface

The multidrop TTL RS-232 interface uses the 8-way Picoflex male SMD connector on PCB or the rear panel, marked 'INTERCONNECT'.
Communications parameters are fixed at:

- 9600 bits/sec
- 8 data bits, 1 stop bit
- No parity
- No handshake

The multidrop bus is based on open drain signals so 10k ohm 1/8 watt pull-up resistor to 3.3V should be used to ensure proper integrity. Signal must be in 0-3.3V volt range only.

3. Overview of the Communications Protocol

The communications protocol used in the ELLx Elliptec Thorlabs module is based on the message structure that always starts with a fixed length, 3-byte *message header* which, in some cases, is followed by a variable length *data packet*. For simple commands, the 3-byte message header is sufficient to convey the entire command.
Some commands require parameters so 3-byte packet must be followed by the data bytes. The number of data bytes depends on command and length is now part of the packet is implicit (see command detail to see length).

Note that in the section below describing the various byte sequences, the C-type of notation will be used for hexadecimal values (e.g., 0x55 means 55 hexadecimal) and logical operators (e.g., | means logic bitwise OR). Values that are longer than a byte follow the Motorola big-endian format.

Data packets coming from modules are terminated with carriage return CR (0xD) first and then line feed LS (0xA). Each module has a 2 second time out such that the discard packet is discarded if time between each byte sent is higher longer than 2 seconds. Alternatively, carriage return CR can be used to clear receiving state machine and exit from a time out error or cancel a command not completed.
Error must be cleared reading module status see "gs".

4. Description of the message header

The 3 bytes in the message header are shown below:

Byte:	byte 1	byte 2	byte 3	byte 4 (optional) .. byte N
Meaning	ADDRESS	COMMAND ID		HEX ASCII DATA

The meaning of some of the fields depends on whether or not the message is followed by a data packet.

ADDRESS:	Is the address (0-F range) default value is 0”
COMMAND:	two bytes mnemonic command i.e., “re” for *reboot.*
HEX ASCII DATA:	Data are in ASCII format in hexadecimal notation. i.e., for a char parameter “0A” ascii value 0x30 0x61 mean decimal value 10.

The type of messages used in the communications exchange between the host and the sub-modules can be divided into 2 general categories:

(a) Host issues a command, “lower case” mnemonic commands “re”.

(b) Module reply or send state spontaneously “upper case” mnemonic command “GS”.

5. Format Specifiers

format	encoding
word	Unsigned 16-bit integer (2 bytes) in the Motorola (big-endian) format for example, decimal 12345 (3039H) is encoded as the byte sequence 30,39
short	Signed 16-bit integer (2 bytes) in 2's compliment format for example, decimal -1 is encoded as the byte sequence FF, FF
dword	Unsigned 32-bit integer (4 bytes) in the Motorola (big-endian) format for example, decimal 123456789 (75BCD15H) is encoded as the byte sequence 07,5B, CD,15
long	Signed 32-bit integer (4 bytes) in 2's compliment format for example, decimal -1 is encoded as the byte sequence FF, FF 4 bytes in the Motorola (big-endian) format for example, decimal -123456789 (FFFFFFFFF8A432EBH) is encoded as the byte sequence F8, A4, 32, EB
char	1 byte (2 digits)
char[N]	string of N characters

Most of data sent via the following command packet are unsigned types apart from any "position" related commands (home offset, move relative etc.)

6. Single Precision Floating Point Format

Single-precision floating-point format is a computer number format that occupies 4 bytes (32 bits) in computer memory and represents a wide dynamic range of values by using a floating point.
Where message parameters use floating point variables, the system uses the IEEE 754 standard.

7. Conversion between position, velocity and acceleration values in standard physical units and their equivalent APT parameters.

To convert between the position and encoder counters in the stage being driven, and real world units, (e.g. mm) the system uses certain conversion (scaling) factors. These conversion factors differ depending on the stage being driven and the controller being used.

The information packet contains all the scaling factor information: stage travel in mm or degrees (depending on ELL module type) and encoder pulses per measurement unit.

Model	TRAVEL	NOTE	PULSES/Rev	LOOP TYPE	Motors
ELL6 SHUTTER	31	31 mm	0	Closed: indexed	1
ELL7 LINEAR STAGE	26	28 mm	1024	Closed: linear 1024 imp/mm	2
ELL8 ROTARY STAGE	360°	Continuous rotation	262144 (40000H)	Closed: linear 2048 imp/mm	2
ELL9 MULTI-POSITION SHUTTER	31	31 mm	0	Closed: indexed	2
ELL10 LINEAR STAGE	60	60 mm	1024	Closed: linear 1024 imp/mm	2
ELL14 ROTATION STAGE	360°	Continuous rotation	262144 (40000H)	Closed: linear	2
ELL17 LINEAR STAGE	28	30 mm	1024	Closed: linear 1024 imp/mm	2
ELL18 ROTARY STAGE	360°	Continuous rotation	262144 (40000H)	Closed: linear 2048 imp/mm	2
ELL20 LINEAR STAGE	60	63.5 mm	1024	Closed: linear 1024 imp/mm	2
ELL15 MOTORIZED IRIS	10.5	10.5 mm	1000	Closed: linear	2

Control Messages

Introduction

The messages described here are listed in random manner, and may be system control messages, generic messages which apply to all module types, or messages which apply only to specific modules. Unless stated otherwise, a message applies to all modules.

Please also see the list of controller specific commands for details on applicability to a specific module type.

_DEVGET_INFORMATION “IN”

Function: Instruct hardware unit to identify itself giving information about model, serial number, firmware and hardware releases, travel, encoder pulses per measurement unit.

USER REQUEST
Command structure (3 bytes):

0	1	2
header only		
A	i	n

Example: Identify shutter at default address “0”

TX “0in”

DEVICE REPLY
Command structure (33 bytes):

0	1	2	3-4	5-12	13-16	17-18	19-20	21-24	25-32
header			Data						
A	“i”	“n”	ELL	SN	YEAR	FW rel	HW rel.	TRAVEL mm/deg	PULSES/M.U.

Example: Identify shutter at default address “0”

RX “0, IN, **06,**12345678,**2015,**01,**81,**001F,**00000001**”

Details:

0	Default Address
IN	Get Information
06	ELL6 bi-positional slider
12345678	Serial Number
2015	Year of manufacturing
01	Firmware release, i.e., 0.1
81	The most significant bit signifies thread type. (1 is imperial, 0 is metric), the remaining 7 bits signify the Hardware release, e.g. 81 (1000,0001) means imperial stage, hardware release 1
001F	31mm travel
00000001	1 pulse per position (bi-positional only)

_HOSTREQ_STATUS “gs”
_HOSTREQ_INFORMATION “in”_DEV
_DEVGET_STATUS “GS”

Function: Instruct hardware unit to get module status or error value. Once read the error value is cleared.

USER REQUEST status
Command structure (3 bytes):

0	1	2
header only		
A	g	s

Example: Get status of shutter at its default address “0”

TX “0gs”

DEVICE REPLY
Command structure (5 bytes):

0	1	2	3	4
header only			*Data*	
A	G	S	Error/Status	

Example: Shutter status value when shutter is at its default address “0”

RX “0GS**00**”
Means the shutter is OK

Status/Error code value	**Meaning**
0	OK, no error
1	Communication time out
2	Mechanical time out
3	Command error or not supported
4	Value out of range
5	Module isolated
6	Module out of isolation
7	Initializing error
8	Thermal error
9	Busy
10	Sensor Error (May appear during self-test. If code persists there is an error)
11	Motor Error (May appear during self-test. If code persists there is an error)
12	Out of Range (e.g., stage has been instructed to move beyond its travel range).
13	Over Current error
14-255	Reserved

Status/Error table

_HOSTREQ_SAVE_USER_DATA “us”

Function: Instruct hardware unit to save motor parameters (such as forward or backward frequencies) and other reserved or user parameters.

USER REQUEST save data
Command structure (3 bytes):

0	1	2
header only		
A	u	s

Example: Save motor and user parameter of shutter at default address “0”

TX “0us”

DEVICE REPLY
See _DEVGET_STATUS packet “GS”.
Device reply with OK Status Packet when command is accepted.

_HOSTREQ_CHANGEADDRESS “ca”

Function: Instruct hardware unit to change to the address of another device. The address is entered in the range 0-F. The default value is a 0.

USER REQUEST Change Address
Command structure (3 bytes):

0	1	2	3
header only			*Data*
A	c	a	New Address

Example: Change a motor at address 0 to address “A”

TX “0 ca A”

DEVICE REPLY
See _DEVGET_STATUS packet “GS”.
Device reply with OK Status Packet from new address when command is accepted.

_HOSTREQ_MOTOR1INFO “i1”
_DEVGET_MOTOR1INFO “I1”

Function: Instruct hardware unit to obtain motor parameters (such as forward or backward frequencies) and other reserved or user parameters.

USER REQUEST Motor Data
Command structure (3 bytes):

0	1	2
header only		
A	i	1

Example: Get motor and user parameter data of shutter at default address “0”

TX “0i1”

DEVICE REPLY
Command structure (23 bytes):

0	1	2	3	4	5	6	7	8	9	10	11	12
header			*Data*									
A	I	1	Reserved	Motor	Current				Reserved			

13	14	15	16	17	18	19	20	21	22	23	24
Data											
Reserved				Forward Period (FwP)				Backward Period (BwP)			

Data Structure

field	description	format
Reserved		
Motor	The state of the motor (1 = ON, 0 = OFF)	char
Current	1866 points is 1 amp	word
Reserved		
Reserved		
Forward Period	Forward period value	word
Backward Period	Backward period value	word

Where:

Period=14740000/frequency for backward and forward motor movements
and 1 Amp of current is equal to 1866 points (1point is 0.54mA circa)

Example: Get motor 1 info at default address “0”

RX “0I1100428FFFFFFFF00BD008B”

Details:

0	default address of shutter
I1	Info packet for motor 1
1	Loop is ON
0	Motor 1 is not working
0428	Last current measurement value was 0x0428 (0.57A)
FFFF	ramp up (0xFFFF not defined)
FFFF	ramp down (0xFFFF not defined)
00BD	forward period (lower frequency, 78kHz)
008B	backward period (higher frequency, 106kHz)

_HOSTSET_FWP_MOTOR1 "f1"

Function: Due to load, build tolerances and other mechanical variances, the default resonating frequency of a particular motor may not be that which delivers best performance. This message allows the operating frequencies for forward movement to be adjusted.

USER SET fwp for motor 1
Command structure (7 bytes):

0	1	2	3	4	5	6
header only			*Data*			
A	f	1	Forward Period (FwP)			

Data Structure

field	description	format
Forward Period	The forward period value	word

Where: Period=14740000/frequency

Example: Set motor 1 forward frequency of a shutter at its default address "0" to 78kHz

TX "0f1804E"

Notes:
The most significant bit of the period (Bit 3) must be set to '8', e.g., 0f1**8**04E above.
User must save new working parameters using "us" command.
To restore the forward frequency factory setting value, send FFF, i.e., 0f18FFF.

DEVICE REPLY
See _DEVGET_STATUS packet "GS".
Device reply with OK Status Packet when command is accepted.

_HOSTSET_BWP_MOTOR1 "b1"

Function: Due to load, build tolerances and other mechanical variances, the default resonating frequency of a particular motor may not be that which delivers best performance. This message allows the operating frequencies for backwards movement to be adjusted.

USER SET BwP for motor 1
Command structure (7 bytes):

0	1	2	3	4	5	6
header only			*Data*			
A	b	1	Backward Period (BwP)			

Data Structure

field	description	format
Backward Period	The backward period value	word

Where: Period=14740000/frequency

Example: Set motor 1 backward frequency of a shutter at its default address "0" to 109kHz

TX "0f1806D"

Notes:
The most significant bit of the period (Bit 3) must be set to '8', e.g., 0f1**8**06D above.
User must save new working parameters using "us" command.
To restore the backward frequency factory setting value, send FFF, i.e., 0b18FFF.

DEVICE REPLY
See _DEVGET_STATUS packet "GS".
Device reply with OK Status Packet when command is accepted.

_HOSTREQ_SEARCHFREQ_MOTOR1 "s1"

Function: Due to load, build tolerances and other mechanical variances, the default resonating frequency of a particular motor may not be that which delivers best performance. This message requests a frequency search be performed to optimize the operating frequencies for backward and forward movement.

For ELL14, ELL17, ELL18 and ELL20 devices
After calling this message, the device can be "fine-tuned" further by calling the OPTIMIZE_MOTORS message.

USER REQUEST search frequency motor 1
Command structure (3 bytes):

0	1	2
header only		
A	s	1

Example: Search for best forward/backward frequencies for motor 1 at address "A"

TX "As1"

Note:
User must save new working parameters using "us" command.

DEVICE REPLY
See _DEVGET_STATUS packet "GS".
Device reply with OK Status Packet when command is accepted and completed.
Moving part may move using this command.

_HOSTREQ_SCANCURRENTCURVE_MOTOR1 “c1”

Function: Instruct hardware unit to scan the current curve for motor 1.

USER REQUEST scancurrentcurve motor 1
Command structure (3 bytes):

0	1	2
header only		
A	c	1

Example: Scan the current curve for motor 1 at address “A”

TX “Ac1”

DEVICE REPLY
See _DEVGET_STATUS packet “GS”.
Device reply with OK Status Packet when command is accepted and completed.
Moving part may move using this command.

_DEVGET_CURRENTCURVEMEASURE_MOTOR1 “C1”

Function: Once the ScanCurrentCurve request has been sent, this message is used to obtain the current curve measurement data from the hardware unit.

DEVICE REPLY
Command structure (522 bytes):

0	1	2	3-4	5-8	9-521
	header			Data	
A	C	1	Period	Current	87 points from 70 kHz up

Where each point is a couple Period (2bytes)-Current (4 bytes):

Period=14740000/frequency for backward and forward motor movements (2 bytes value) and
1 Amp of current is equal to 1866 points (4 bytes value in little endian format)

Each measurement takes around 12 seconds starting from 70 kHz to 120kHz circa.

_HOST_ISOLATEMINUTES “is”

Function: Isolate the device, such that it will not reply for the specified time period. The isolation time is specified in number of minutes.

USER REQUEST home
Command structure (4 bytes):

0	1	2	3
header only			*Data*
A	i	s	Minutes

Example: Isolate the device at address “0” for 60 minutes.

TX “0is3C” (0x3C = 60 minutes)

_HOSTREQ_MOTOR2INFO “i2”
_DEVGET_MOTOR2INFO “I2”

THIS MESSAGE APPLIES ONLY TO DEVICES WITH 2 MOTORS. IT IS NOT APPLICABLE TO SINGLE MOTOR DEVICES SUCH AS THE BI-POSITIONAL SLIDER.

Function: Instruct hardware unit to obtain motor parameters (such as forward or backward frequencies) and other reserved or user parameters.

USER REQUEST Motor Data
Command structure (3 bytes):

0	1	2
	header only	
A	i	2

Example: Get motor and user parameter data of stage at default address “0”

TX “0i2”

DEVICE REPLY
Command structure (23 bytes):

0	1	2	3	4	5	6	7	8	9	10	11	12
	header						*Data*					
A	I	2	Reserved	Motor		Current				Reserved		

13	14	15	16	17	18	19	20	21	22	23	24
					Data						
	Reserved				Forward Period (FwP)				Backward Period (BwP)		

Data Structure

field	description	format
Reserved		
Motor	The state of the motor (1 = ON, 0 = OFF)	char
Current	1866 points is 1 amp	word
Reserved		
Reserved		
Forward Period	Forward period value	word
Backward Period	Backward period value	word

Where:

Period=14740000/frequency for backward and forward motor movements
and 1 Amp of current is equal to 1866 points (1point is 0.54mA circa)

Example: Get motor 2 info at default address “0”

RX “0I2100428FFFFFFFF00BD008B”

Details:

0	default address of shutter
I2	Info packet for motor 2
1	Loop is ON
0	Motor 2 is not working
0428	Last current measurement value was 0x0428 (0.57A)
FFFF	ramp up (0xFFFF not defined)
FFFF	ramp down (0xFFFF not defined)
00BD	forward period (lower frequency, 78kHz)
008B	backward period (higher frequency, 106kHz)

_HOSTSET_FWP_MOTOR2 “f2”

THIS MESSAGE APPLIES ONLY TO DEVICES WITH 2 MOTORS. IT IS NOT APPLICABLE TO SINGLE MOTOR DEVICES SUCH AS THE BI-POSITIONAL SLIDER.

Function: Set the forward period for motor 2.

USER SET fwp for motor 2
Command structure (7 bytes):

0	1	2	3	4	5	6
header only			*Data*			
A	f	2	Forward Period (FwP)			

Data Structure

field	description	format
Forward Period	The forward period value	word

Where: Period=14740000/frequency

Example: Set motor 2 forward frequency of a stage at its default address “0” to 78kHz

TX “sf1004E”

Note:
User must save new working parameters using “us” command.

DEVICE REPLY
See _DEVGET_STATUS packet “GS”.
Device reply with OK Status Packet when command is accepted.

_HOSTSET_BWP_MOTOR2 "b2"

THIS MESSAGE APPLIES ONLY TO DEVICES WITH 2 MOTORS. IT IS NOT APPLICABLE TO SINGLE MOTOR DEVICES SUCH AS THE BI-POSITIONAL SLIDER.

Function: Set the backward period for motor 2.

USER SET BwP for motor 2
Command structure (7 bytes):

0	1	2	3	4	5	6
header only			*Data*			
A	b	2	Backward Period (BwP)			

Data Structure

field	description	format
Backward Period	The backward period value	word

Where: Period=14740000/frequency

Example: Set motor 2 backward frequency of a stage at its default address "0" to 109kHz

TX "0f1006D"

Note:
User must save new working parameters using "us" command.

DEVICE REPLY
See _DEVGET_STATUS packet "GS".
Device reply with OK Status Packet when command is accepted.

_HOSTREQ_SEARCHFREQ_MOTOR2 "s2"

THIS MESSAGE APPLIES ONLY TO DEVICES WITH 2 MOTORS. IT IS NOT APPLICABLE TO SINGLE MOTOR DEVICES SUCH AS THE BI-POSITIONAL SLIDER (ELL6 and ELL9).

Function: Due to load, build tolerances and other mechanical variances, the default resonating frequency of a particular motor may not be that which delivers best performance. This message requests a frequency search be performed to optimize the operating frequencies for backward and forward movement.

For ELL14, ELL17, ELL18 and ELL20 devices
After calling this message, the device can be "fine-tuned" further by calling the OPTIMIZE_MOTORS message.

USER REQUEST search frequency motor 2
Command structure (3 bytes):

0	1	2
header only		
A	s	2

Example: Search for best forward/backward frequencies for motor 2 at address "A"

TX "As2"

Note:
User must save new working parameters using "us" command.

DEVICE REPLY
See _DEVGET_STATUS packet "GS".
Device reply with OK Status Packet when command is accepted and completed.
Moving part may move using this command.

_HOSTREQ_SCANCURRENTCURVE_MOTOR2 "c2"

THIS MESSAGE APPLIES ONLY TO DEVICES WITH 2 MOTORS. IT IS NOT APPLICABLE TO SINGLE MOTOR DEVICES SUCH AS THE BI-POSITIONAL SLIDER.

Function: Instruct hardware unit to scan the current curve for motor 2.

USER REQUEST scancurrentcurve motor 2
Command structure (3 bytes):

0	1	2
header only		
A	c	2

Example: Scan the current curve for motor 2 at address "A"

TX "Ac2"

DEVICE REPLY
See _DEVGET_STATUS packet "GS".
Device reply with OK Status Packet when command is accepted and completed.
Moving part may move using this command.

_DEVGET_CURRENTCURVEMEASURE_MOTOR2 “C2”

THIS MESSAGE APPLIES ONLY TO DEVICES WITH 2 MOTORS. IT IS NOT APPLICABLE TO SINGLE MOTOR DEVICES SUCH AS THE BI-POSITIONAL SLIDER.

Function: Once the ScanCurrentCurve request has been sent, this message is used to obtain the current curve measurement data from the hardware unit.

DEVICE REPLY
Command structure (522 bytes):

0	1	2	3-4	5-8	9-521
header			Data		
A	C	2	Period	Current	87 points from 70 kHz up

Where each point is a couple Period (2bytes)-Current (4 bytes):

Period=14740000/frequency for backward and forward motor movements (2 bytes value) and
1 Amp of current is equal to 1866 points (4 bytes value in little endian format)

Each measurement takes around 12 seconds starting from 70 kHz to 120kHz circa.

_HOSTREQ_HOME “ho”
_DEVGET_POSITION “GS or PO”

THIS MESSAGE DOES NOT APPLY TO OPTIC SLIDER DEVICES

Function: Instruct hardware unit to move to the home position.
For rotary stages only, byte 3 details the homing direction, 0 for clockwise and 1 for CCW.
For other stages, byte 3 is ignored.

USER REQUEST home
Command structure (4 bytes):

0	1	2	3
header only			
A	h	o	dir

Example: Request the rotary stage at address “0” to move to home in CW direction.

TX “0ho0”

DEVICE REPLY
If the move is still incomplete, the device will send a status message (see GetStatus (GS) message). If the move is complete, the device will send the current position, e.g., 0x3000 (RX "APO00003000").

Command structure (5 bytes or 11 bytes):

0	1	2	3	4
header only			*Data*	
A	G	S	Status	

or

0	1	2	3	4	5	6	7	8	9	10
header only			*Data*							
A	P	O	Position							

Example: Rotator status value at default address “0”

RX “0PO**00000000**”
Means the stage is homed and the move is complete.

Position is a long type (32 bit signed, 2’s complement)

_HOSTSET_AUTOHOMING "ah"

THIS MESSAGE IS ONLY APPLICABLE TO THE ELL15 MOTORIZED IRIS

Function: Instruct the hardware unit to move to home position at start up. For ELL15 only, byte 3 details whether auto-homing is activated or deactivated. '0' for auto-homing off, '1' for auto-homing on.

USER REQUEST auto-home

Command structure (4 bytes):

0	1	2	3
header only			
A	a	h	1

Example: Request the device at address "0" to deactivate auto-homing.

TX "0ah0"

DEVICE REPLY

If the move is still being performed, the device will send a status message (see GetStatus (GS) message). If the move is complete, the device will send the home position, i.e., 11.5.

Command structure (5 bytes or 11 bytes):

0	1	2	3	4
header only			*Data*	
A	G	S	Status	

or

0	1	2	3	4	5	6	7	8	9	10
header only				*Data*						
A	P	O		Position						

Example: Iris value at default address "0".

RX "0PO000002CEC"

Meaning the stage is homed and the move is complete.
Position is a long type (32 bit signed, 2's complement).

_HOSTREQ_MOVEABSOLUTE "ma"
_DEVGET_POSITION "GS or PO"

THIS MESSAGE DOES NOT APPLY TO OPTIC SLIDER DEVICES

Function: Instruct hardware unit to move to a specified absolute position. The position to move is specified in bytes 3 to 10 in encoder pulses (2048 encoder pulses per mm).

USER REQUEST move absolute
Command structure (11 bytes):

0	1	2	3	4	5	6	7	8	9	10
header only			*Data*							
A	m	a	Position							

Example: Request a linear stage at address "A" to move at position 4mm. Linear stage has 2048 encoder pulses per mm, hence 4 mm 8192 pulses (0x2000 in hexadecimal).

TX "Ama00002000"

Note:
Position is a long type (32 bit signed, 2's complement)
Use "in" command to get number of pulses per engineering units (mm or degrees).

DEVICE REPLY
If the move is still incomplete, the device will send a status message (see GetStatus (GS) message). If the move is complete, the device will send the current position, e.g., 0x3000 (RX "APO00003000").

Command structure (5 bytes or 11 bytes):

0	1	2	3	4
header only			*Data*	
A	G	S	Status	

or

0	1	2	3	4	5	6	7	8	9	10
header only				*Data*						
A	P	O		Position						

Example: Linear stage at default address "A" is at position 4 mm

RX "APO00002000"
Meaning the linear stage is at 4mm of travel (-0.5um of error)
Position is a long type (32 bit signed, 2's complement)

_HOSTREQ_MOVERELATIVE “mr”
_DEVGET_POSITION “GS or PO”

THIS MESSAGE DOES NOT APPLY TO OPTIC SLIDER DEVICES

Function: Instruct hardware unit to move to a specified relative position. The position to move is specified in bytes 3 to 10 in encoder pulses (2048 encoder pulses per mm).

USER REQUEST move relative
Command structure (11 bytes):

0	1	2	3	4	5	6	7	8	9	10
header only			*Data*							
A	m	r	Position							

Example: Request a linear stage at address “A” to move 2mm from the present position.
Linear stage has 2048 encoder pulses per mm, hence 2 mm 4096 pulses (0x1000 in hexadecimal).

TX “Amr00001000”

Note:
Position is a long type (32 bit signed, 2’s complement)
Use “in” command to get number of pulses per engineering units (mm or degrees).

DEVICE REPLY
If the move is incomplete, the device will send a status message (see GetStatus (GS) message). If the move is complete, the device will send the current position, e.g., 0x3000 (RX "APO00003000").

Command structure (5 bytes or 11 bytes):

0	1	2	3	4
header only			*Data*	
A	G	S	Status	

or

0	1	2	3	4	5	6	7	8	9	10
header only			*Data*							
A	P	O	Position							

Position is a long type (32 bit signed, 2’s complement)

Example: Linear stage at default address “A” is at absolute position 6 mm

RX “APO00003000”
Meaning the linear stage is at 6mm of travel (-0.5um of error)

_HOSTREQ_HOMEOFFSET “go”
_DEVGET_HOMEOFFSET “HO”
_HOSTSET_HOMEOFFSET “so”

THIS MESSAGE DOES NOT APPLY TO OPTIC SLIDER DEVICES

Function: This message sets/returns the distance of the Home position from the absolute limit of travel.
The position is specified in bytes 3 to 10 in encoder pulses (2048 encoder pulses per mm).

NOTES

The Home offset is set at the factory to ensure that all stages home consistently. Before changing the offset value, it is good practice to note the factory default value to allow easy resetting in case of problems.

The home offset value can be adjusted to change the home position to a required value (e.g., 0mm). For good operation the home offset value must be greater than 500um for Linear stages and 0.7 degrees for rotation stages.

Increasing the home offset value may reduce the maximum achievable travel.

USER REQUEST home offset
Command structure (3 bytes):

0	1	2
header only		
A	g	o

Example: Request the home offset for the device at address “A”

TX “Ago”

DEVICE REPLY
The device reply consists of a 3-byte message header followed by an 8 byte data packet as follows:

0	1	2	3	4	5	6	7	8	9	10
header only			*Data*							
A	H	O	Offset Distance							

Example: Get the home offset for a stage at address “l”.
The stage has 2048 encoder pulses per mm. For example, a home offset of 0.25mm will be returned as 0x200 (i.e., 512 encoder pulses).

RX “AHO00000200”

Note:
Position is a long type (32 bit signed, 2’s complement)
Use “in” command to get number of pulses per engineering units (mm or degrees).

USER SET home offset

Command structure (11 bytes):

0	1	2	3	4	5	6	7	8	9	10
header only			*Data*							
A	s	o	Offset Distance							

Example: Set the home offset for a stage at address “l”.
The stage has 2048 encoder pulses per mm. For example, a home offset of 0.25mm will be set as 0x200 (i.e., 512 encoder pulses).

TX “Aso00000200”

_HOSTREQ_JOGSTEPSIZE “gj”
_DEVGET_JOGSTEPSIZE “GJ”
_HOSTSET_JOGSTEPSIZE “sj”

THIS MESSAGE DOES NOT APPLY TO OPTIC SLIDER DEVICES

Function: This message sets/returns the distance to move when a jog command is initiated. The jog move is initiated by calling the Forward (fw) or Backward (bw) message.
The jog step size is specified in bytes 3 to 10 in encoder pulses (2048 encoder pulses per mm).

ELL14 Units Only Setting the step size to zero will make the unit move continuously when the Forward (fw) or Backward (bw) message. To stop the motion, a STOP (st) command must be sent, otherwise the device will reply with status "Busy". When operating in continuous motion, the velocity must be set between 50%-70% using the "sv" command otherwise the unit could overheat.

USER REQUEST jog step size
Command structure (3 bytes):

0	1	2
header only		
A	g	j

Example: Request the jog step size for the device at address “A”
TX “Agj”

DEVICE REPLY
The device reply consists of a 3-byte message header followed by an 8-byte data packet:

0	1	2	3	4	5	6	7	8	9	10
header only			*Data*							
A	G	J	Offset Distance							

Example: Get the jog step size for a stage at address “A”.
The stage has 2048 encoder pulses per mm. For example, a jog step size of 1.0 mm will be returned as 0x800 (i.e., 2048 encoder pulses).

RX “AGJ00000800”

Note:
Position is a long type (32 bit signed, 2’s complement)
Use “in” command to get number of pulses per engineering units (mm or degrees).

USER SET jog step size
Command structure (11 bytes):

0	1	2	3	4	5	6	7	8	9	10
header only			*Data*							
A	s	j	Offset Distance							

Example:

Set the jog step size for a stage at address “A”.
The stage has 2048 encoder pulses per mm. For example, a jog step size of 0.25mm will be set as 0x200 (i.e., 512 encoder pulses).
Position is a long type (32 bit signed, 2’s complement)

TX “Asj00000200”

Example [Ell14]:

Setting the step size to zero will make the unit move continuously by calling the Forward (fw) or Backward (bw) message.

TX “Asj00000000”

_HOST_FORWARD “fw”
_HOST_BACKWARD “bw”

Function: This message instructs the hardware unit to move the specified motor forward or backwards by the distance set in the JogStepSize message.

Ell14 Units Only Setting the step size to zero will make the unit move continuously when the Forward (fw) or Backward (bw) message. To stop the motion, a STOP (st) command must be sent, otherwise the device will reply with status "Busy". When operating in continuous motion, the velocity must be set between 50%-70% using the "sv" command otherwise the unit could overheat.

USER REQUEST forward
Command structure (3 bytes):

0	1	2
header only		
A	f	w

Example: Move the motor at default address “A” forward

TX “Afw”

DEVICE REPLY
If the move is still incomplete, the device will send a status message (see GetStatus (GS) message). If the move is complete, the device will send the current position, e.g., 0x3000 (RX "APO00003000").

Command structure (5 bytes or 11 bytes):

0	1	2	3	4
header only			*Data*	
A	G	S	Status	

or

0	1	2	3	4	5	6	7	8	9	10
header only			*Data*							
A	P	O	Position							

Example: Linear stage at default address “A” is at absolute position 6 mm
The current position is returned in encoder pulses. There are 2048 encoder pulses per mm. For example, a returned value of 0x3000 is equivalent to 12,288 pulses, i.e., 6 mm.

RX “APO00003000”
Meaning the linear stage is at 6mm of travel (-0.5um of error)

Position is a long type (32 bit signed, 2’s complement)

_HOST_MOTIONSTOP "st"

Function: **This message is applicable only to ELL14 units** and instructs the hardware to stop its motion when operating in continuous motion. See Host_Forward, Host_Backward and SetJogStepsize for information to set unit into continuous motion. Units other than the Ell14 will answer with a command error or not supported message.

USER REQUEST motion stop
Command structure (3 bytes):

0	1	2
header only		
A	s	t

Example: Stop the motor at default address "A" from continuous motion

TX "Ast"

DEVICE REPLY
The device will stop and send a status message (see GetStatus (GS) message).

Command structure (5 bytes):

0	1	2	3	4
header only			*Data*	
A	G	S	Status	

Example: The user set the unit to continuous mode by setting the jog velocity to 50% and the jog step size to 0:
TX Asv32
RX "AGS00"
TX "Asj00000000"
RX "AGS00"
Started the motion with
TX "Afw" or "Abw"
Stop the motion with
TX "Ast"
RX "AGS09" and/or RX "AGS00"

.

_HOST_GETPOSITION “gp”
_DEV_GETPOSITION “PO”

Function: This message returns the current position of the specified motor

USER REQUEST save data
Command structure (3 bytes):

0	1	2
header only		
A	g	p

Example: Request the current position of the motor at default address “A”

TX “Agp”

DEVICE REPLY
The device will send the current position,
e.g., 0x0FFF.

0	1	2	3	4	5	6	7	8	9	10
header only			*Data*							
A	P	O	Position							

Position is a long type (32 bit signed, 2’s complement)

Example: Linear stage at default address “A” is at absolute position 6 mm

RX “APO00003000”
Meaning the linear stage is at 6mm of travel (-0.5um of error)

_HOSTREQ_VELOCITY “gv”
_DEVGET_VELOCITY “GV”
_HOSTSET_VELOCITY “sv”

THIS MESSAGE DOES NOT APPLY TO OPTIC SLIDER DEVICES

Function: Velocity control of Elliptec products is achieved by adjusting the drive power. This message sets/returns the velocity compensation to be applied to a move when a command is initiated. The move is initiated by calling the Forward (fw) or Backward (bw) message. The velocity is specified in bytes 4 and 5 as percentage of max velocity. Note that depending on the load, velocity less than 25% to 45% of max may cause the device to stall.

ELL14 Units Only Setting the step size to zero will make the unit move continuously when the Forward (fw) or Backward (bw) message. To stop the motion, a STOP (st) command must be sent, otherwise the device will reply with status "Busy". When operating in continuous motion, the velocity must be set between 50%-70% using the "sv" command otherwise the unit could overheat.

USER REQUEST Get Velocity
Command structure (3 bytes):

0	1	2
header only		
A	g	v

Example: Request the velocity compensation used for the device at address “A”
TX “Agv”

DEVICE REPLY
The device reply consists of a 3-byte message header followed by a 2-byte data packet:

0	1	2	3	4
header only			*Data*	
A	G	V	Velocity	

Example: Get the velocity compensation for a stage at address “A”.

RX “AGV64”, i.e., the velocity compensation is 100% of the maximum available.

USER SET Velocity
Command structure (5 bytes):

0	1	2	3	4
header only			*Data*	
A	s	v	Velocity	

Example: Set the velocity compensation for a stage at address “A” to 50%.

TX “Asv32”

_DEVGET_BUTTONSTATUS “BS”
_DEVGET_BUTTONPOSITION “BO”

Function: These messages are similar to the GetStatus (GS) and GetPosition responses but are sent automatically when a move or position change is demanded via the hardware FW, BW and JOG buttons. The Get_BUTTONSTATUS message is sent while the device is in motion, whereas the GET_BUTTONPOSITION message is sent once the move has completed.

GET_BUTTONSTATUS DEVICE REPLY

Command structure (5 bytes):

0	1	2	3	4
header only			*Data*	
A	B	S	Error/Status	

Example: Shutter status value when shutter is at its default address “0”

RX “0BS**00**”
Means the shutter is OK

Status/Error code value	**Meaning**
0	OK, no error
1	Communication time out
2	Mechanical time out
3	Command error or not supported
4	Value out of range
5	Module isolated
6	Module out of isolation
7	Initializing error
8	Thermal error
9	Busy
10	Sensor Error (May appear during self-test. If code persists there is an error)
11	Motor Error (May appear during self-test. If code persists there is an error)
12	Out of Range (e.g., stage has been instructed to move beyond its travel range).
13	Over Current error
14-255	Reserved

Status/Error table

GET_BUTTONPOSITION DEVICE REPLY

The device will send the current position once the move is complete,
e.g., 0x3000.

0	1	2	3	4	5	6	7	8	9	10
header only			*Data*							
A	B	O	Position							

Example: Linear stage at default address "A" is at absolute position 6 mm

RX "ABO00003000"
Meaning the linear stage is at 6mm of travel (-0.5um of error)
Position is a long type (32 bit signed, 2's complement)

_HOST_GROUPADDRESS “ga”

Function: This message allows several devices to be addressed simultaneously as a temporary group, such that their movement can be synchronized. This is achieved by instructing a module(s) to respond to a different address for a particular operation. Once the motion has been completed the device returns to its original address.
For example, suppose there are 3 sliders at the addresses 0, 1, and 2, and we want to synchronize the sliders at addresses 0 and 2. Sending message 2ga0 instructs the slider at address 2 to listen to address 0. Slider 2 would reply 0gs00 from the new address. The user can now ask for a synchronized move of both the sliders using 0fw. Both slider 0 and slider 2 would move to forward position in the same usec, replying:
0gp0000001F 2gp0000001F
Because motion is synchronized, data sent by the module is handled with a priority (otherwise they would overlap), i.e., address 0 has the highest priority sending data, while address F has the lowest. So, address 0 comes first even for simultaneous moves in the same usec. Error messages behave in a similar way.
It is possible to use a combination of "ca" and "ga" to synchronize more movements, but message would come only from address set by "ca".
For manual operation via the handset, synchronization is hard wired. If multiple devices are connected to the comms bus, then all devices will move when the FWD and BWD buttons are pressed.

USER REQUEST Create Group
Command structure 4 bytes):

0	1	2	3
header only			*Data*
A	g	a	New Address

Example: Instruct a motor at address 0 to listen to address “A”

TX “0 ga A”

DEVICE REPLY
Command structure (5 bytes):

0	1	2	3	4
header only			*Data*	
A	G	S	Error/Status	

Example: Status value when a motor at address 0 is listening to address “A”

RX “AGS**00**”
Means the shutter is OK

Please see the Forward and Backward (FW and BW) messages for information on the move message format.

_HOST_OPTIMIZE_MOTORS "om"
_DEV_STATUS "GS"

THIS MESSAGE APPLIES ONLY TO ELL14, ELL15, ELL17, ELL18 and ELL20 DEVICES

Function: Due to load, build tolerances and other mechanical variances, the default resonating frequency of a particular motor may not be that which delivers best performance.
This message fine tunes the frequency search performed by the SEARCHFREQ messages. When this message is called, the SEARCHFREQ message is called first automatically to optimize the operating frequency. After completion, another frequency search is performed, and the mechanical performance is monitored to further optimize the operating frequencies for backward and forward movement.

NOTE. The operation described can take several minutes to perform and during that time, the bus is occupied almost exclusively by this message. This may have an effect on the performance of any other units connected to the same bus.

USER REQUEST Optimize Motors
Command structure (3 bytes):

0	1	2
header only		
A	o	m

Example: Optimize the motors for the device at address "A"
TX "Aom"

DEVICE REPLY
If the optimization is still being performed, the device will send a "busy (09)" status message (see GetStatus (GS) message). If the cycle is complete, the device will send "no Errors (0)".

Command structure (5 bytes):

0	1	2	3	4
header only			*Data*	
A	G	S	Status	

_HOST_CLEAN_MECHANICS "cm"
_DEV_STATUS "GS"

THIS MESSAGE APPLIES ONLY TO ELL14, ELL17, ELL18 and ELL20 DEVICES

Function: Calling this message starts a cleaning cycle, during which the device will move backwards and forwards over its full range of travel for a few minutes, in order to remove any dust or debris from the bearing rails and motor contacts. If an attempt is made to send another message while the cleaning process is being performed, the device will reply "busy (09)" in the status bits.

NOTE. The operation described can take several minutes to perform and during that time, the bus is occupied almost exclusively by this message. This may have an effect on the performance of any other units connected to the same bus.

USER REQUEST Get Velocity
Command structure (3 bytes):

0	1	2
header only		
A	c	m

Example: Clean the device at address "A"
TX "Acm"

DEVICE REPLY
If the clean cycle is still being performed, the device will send a "busy (09) status message (see GetStatus (GS) message). If the cycle is complete, the device will send "no Errors (0)".

Command structure (5 bytes):

0	1	2	3	4
header only			*Data*	
A	G	S	Status	

_HOST_MOTIONSTOP "st"
_DEV_STATUS "GS"

THIS MESSAGE APPLIES ONLY TO ELL14, ELL15, ELL17, ELL18 and ELL20 DEVICES

Function: This message stops (aborts) the optimization or cleaning process initiated by the OPTIMIZE_MOTORS and CLEAN_MECHANICS messages described previously.

USER REQUEST Stop
Command structure (3 bytes):

0	1	2
header only		
A	s	t

Example: Stop the current process for the device at address "A"
TX "Ast"

DEVICE REPLY
When the stop instruction is complete, the device will send "no Errors (0)".

Command structure (5 bytes):

0	1	2	3	4
header only			*Data*	
A	G	S	Status	

_HOST_ENERGIZE_MOTOR “e1”
_DEV_STATUS “GS”

THIS MESSAGE APPLIES ONLY TO ELL5 PIEZO DRIVER DEVICES

Function: Calling this message energizes the motor at the address specified in bit 0. The energizing frequency is specified as a 4-bit hex string in bits 3 to 6, in the range 230 Hz to 2 MHz.

USER SET
Command structure (7 bytes):

0	1	2	3	4	5	6
header only			*Data*			
A	f	1	Forward Period (FwP)			

Data Structure

field	description	format
Forward Period	The forward period value	word

Where: Period=14,740,000/frequency

Example: Energize the motor at address “0” to 78kHz

TX “0e100BC”

14,740,000/78,000 = 188 = 00BC

DEVICE REPLY
The motor will remain energized until a Halt command is received (see HALT_MOTOR message) and the device will send “no Errors (0)”.

Command structure (5 bytes):

0	1	2	3	4
header only			*Data*	
A	G	S	Status	

_HOST_HALT_MOTOR “h1”
_DEV_STATUS “GS”

THIS MESSAGE APPLIES ONLY TO ELL5 PIEZO DRIVER DEVICES

Function: This message stops (HALTS) the motor at the address specified in bit 0, previously initiated by the ENERGIZE_MOTORS message described previously.

USER REQUEST Stop
Command structure (3 bytes):

0	1	2
header only		
A	h	1

Example: Stop the motor at address “0”
TX “0h1”

DEVICE REPLY
When the stop instruction is complete, the device will send “no Errors (0)”.

Command structure (5 bytes):

0	1	2	3	4
header only			*Data*	
A	G	S	Status	

ERROR MESSAGES

Function: These messages are returned by the GetStatus and Get_BUTTONSTATUS messages when an error condition exists.

Status/Error code value	**Meaning**
0	OK, no error
1	Communication time out
2	Mechanical time out
3	Command error or not supported
4	Value out of range
5	Module isolated
6	Module out of isolation
7	Initializing error
8	Thermal error
9	Busy
10	Sensor Error (May appear during self-test. If code persists there is an error)
11	Motor Error (May appear during self-test. If code persists there is an error)
12	Out of Range (e.g., stage has been instructed to move beyond its travel range).
13	Over Current error
14-255	Reserved

Status/Error table

Messages Applicable to Shutter Devices

Messages Applicable to Rotation Devices

Messages Applicable to Linear Devices

Messages Applicable to Iris Devices