



# **ORT/RWT Series Transducer Communication Protocol**

**Revision 7 - December 2018**

## Table of Contents

<b>Introduction</b> .....	<b>4</b>
<b>Compatible Models</b> .....	<b>4</b>
<b>RS232</b> .....	<b>4</b>
<b>USB</b> .....	<b>5</b>
<b>Protocol Description</b> .....	<b>6</b>
Binary Format.....	6
ASCII Format.....	8
<b>Torque Modes</b> .....	<b>9</b>
Peak Torque.....	9
Peak Torque with AutoReset.....	9
Peak Torque CW.....	9
Peak Torque CCW.....	9
PeakMinMax.....	9
<b>Speed Modes</b> .....	<b>10</b>
Slow.....	10
Fast.....	10
<b>Temperature Sensors</b> .....	<b>11</b>
Ambient.....	11
Shaft.....	11
Internal.....	11
<b>Unit Key</b> .....	<b>11</b>
<b>Technology Family Key</b> .....	<b>12</b>
<b>Command Set</b> .....	<b>13</b>
Get Transducer ID.....	14
Get Transducer Information.....	15
Get Torque.....	16
Get Peak Torque.....	16
Get Peak Torque Auto Reset.....	16
Get Peak Torque CW.....	17
Get Peak Torque CCW.....	17
Get PeakMinMax Max.....	17
Get PeakMinMax Min.....	18
Get PeakMinMax.....	18
Get Torque - Convert Units To.....	18
Get Peak Torque - Convert Units To.....	19
Get Peak Torque Auto Reset - Convert Units To.....	19
Get Peak Torque CW - Convert Units To.....	20
Get Peak Torque CCW - Convert Units To.....	20
Get PeakMinMax Max - Convert Units To.....	21
Get PeakMinMax Min - Convert Units To.....	21
Get PeakMinMax - Convert Units To.....	22
Get Speed.....	22
Get Power.....	23
Get Temperature Ambient.....	23
Get Temperature Shaft.....	23
Get SlowCap Speed.....	24
Get FastCap Speed.....	24
Get SlowCap Power in Watts.....	24
Get FastCap Power in Watts.....	25
Get SlowCap Power in HP.....	25
Get FastCap Power in HP.....	25
Reset Specified Peaks.....	26
Reset ALL Peak Torque Values.....	28

Reset ALL Peak Values.....	28
Reset System Values .....	28
Peak Torque Reset .....	29
Peak Torque Auto Reset - Reset.....	29
Zero Transducer with Average .....	29
Zero Transducer .....	29
Get PeakMinMax & Reset .....	30
Set Torque Filter.....	30
Get Torque Filter .....	31
Set Speed Filter.....	31
Get Speed Filter .....	31

### **Contact Details**

Sensor Technology Ltd,  
Apollo Park,  
Ironstone Lane,  
Wroxton,  
BANBURY,  
OX15 6AY,  
United Kingdom.

### **Sales**

Email: [stlsales@sensors.co.uk](mailto:stlsales@sensors.co.uk)  
Tel: +44 (0)1869 238400

### **Technical Support**

Email: [software@sensors.co.uk](mailto:software@sensors.co.uk)  
Tel: +44 (0)1869 238400

**Introduction**

The RS232 and USB interfaces on the ORT/RWT Series Transducers provide a method of extracting digital operational data from the Transducer. Functions for controlling aspects of the Transducers operation are also present.

The protocol used is the same for both RS232 and USB, but due to USB’s more complex nature, it is recommended that the STCOMM DLL be used to interface your own application to the Transducer.

If development is being done on the Windows platform you may want to consider using the STCOMM DLL. The DLL simplifies the use of the USB and RS232 interfaces by providing a unified interface to access transducers connected via either method; it takes care of the low-level driver access, protocol negotiation and data manipulation.

**Compatible Models**

The protocol described in this manual is compatible with transducers from the advanced ORT and RWT family of products. Transducers must be running firmware version 3 or higher and have digital communications enabled.

The table below lists the models that are compatible:

Transducer Family	Model Range	Models
Optical (ORT)	ORT240	ORT240/ORT241
Rayleigh Wave (RWT)	RWT320	RWT320/RWT321/RWT322
	RWT340	RWT340/RWT341/RWT342
	RWT420	RWT420/RWT421/RWT422
	RWT440	RWT440/RWT441/RWT442

Compatible transducers can be identified by the presence of a status LED and serial number greater than 12200.

**RS232**

The RS232 interface provides a full-duplex communication channel; each byte of data is transmitted in a packet of 10bits. The data packet consists of one start bit, 8 data bits, no parity and one stop bit.

*Data Packet Format (D0 – Least Significant Bit)*

Start Bit	D0	D1	D2	D3	D4	D5	D6	D7	Stop Bit
-----------	----	----	----	----	----	----	----	----	----------

The RS232 interface can operate at 3 different baud rates, 9600, 38400 and 115200bps (default). The baud rate can be changed by using the “Transducer Control” program which accompanies our ORT/RWT Series Transducers.

## USB

The USB interface on the transducer is a USB 2.0 Full-Speed device running at 12Mbps. Communication is conducted via bulk transfers using two endpoints. A third endpoint is used for high speed mode, but its use requires optimised code and is not discussed in this manual.

On Windows the STCOMM DLL should be used to connect to USB and RS232 transducers as it greatly simplifies access.

It is beyond the scope of this manual to discuss the intricacies of the USB interface, the information provided below is for users with knowledge of the interface and for those wishing to use a platform not supported by the DLL.

The protocol is exactly the same for both RS232 and USB, ignoring the underlying USB transactions which should be controlled by the operating system, driver and interface library.

On Windows our software uses the libusb-win32 driver and library, on Linux you should use libusb which is what the Windows library was derived from.

### **Vendor ID (VID)**

Sensor Technology – 0x26B4

Early versions of our transducers used Vendor ID's and Product ID's provided by various silicon manufacturers (Silicon Laboratories – 0x10C4). These transducers used proprietary static libraries compiled into the transducer firmware. These static libraries were accompanied by a driver and DLL library.

We have since transitioned to a unified generic USB driver (libusb-win32), unfortunately, the transducers which use the proprietary libraries need some coercion to work correctly. A setup routine written for libusb-win32 can be provided if required.

### **Product ID (PID)**

The table below list the different Product ID's and associated Vendor ID's.

Vendor ID	Product ID	Model
0x26B4	0x0010	ORT 220 Series Torque Transducer (R5)
0x26B4	0x0011	ORT 240 Series Torque Transducer (R5)
0x26B4	0x0012	ORT 260 Series Torque Transducer (R5)
0x10C4	0x82FB	RWT320/340/360 Series Torque Transducer (R3)
0x26B4	0x0001	RWT 320 Series Torque Transducer (R4)
0x26B4	0x0002	RWT 340 Series Torque Transducer (R4)
0x26B4	0x0003	RWT 360 Series Torque Transducer (R4)
0x26B4	0x0004	RWT 420 Series Torque Transducer (R4/R5)
0x26B4	0x0005	RWT 440 Series Torque Transducer (R4/R5)
0x26B4	0x0006	RWT 460 Series Torque Transducer (R4/R5)

**Endpoints**

Communication is conducted by IN and OUT bulk endpoints. At a high level the IN/OUT channels can be thought of in the same terms as the RX and TX lines of the RS232 interface.

Endpoint Type	Direction	ID
Bulk	IN (Receive)	0x82
Bulk	Out (Transmit)	0x02

**Protocol Description**

The Transducer uses a simple request and send protocol. Data is transferred using either a binary or ASCII (Firmware v4.2) format. Firmware version 4.2 introduced a ASCII format alongside the original and efficient binary format. The ASCII format was added to provide compatibility with machines that had trouble decoding the binary data.

**Binary Format**

The binary format is a fast and efficient method of extracting data from the transducer. It has a much lower overhead compared to the ASCII format. Commands are one byte in length and either request data or switch Transducer functions on or off.

To request data, transmit a byte equal to the command number of the function you want to request, the Transducer will then reply with the relevant data or action your request. Some commands require additional parameters, in these cases the parameter data should follow the request byte, refer to the command descriptions for more information.

The data returned from request commands will be output in various formats, the format used depends upon the type of data requested. Multi-byte number types are output with the least significant byte (LSB) first, as with Little-Endian systems.

The data types used are C type variables, Int type variables are 2 bytes in size.

The following outlines the variable types used.

**Float Data Type (4 bytes):** IEEE-754 standard floating point number format.

Floating-point format:

SEEE EEEE EMMM MMMM MMMM MMMM MMMM MMMM  
 S – Sign Bit, E – Exponent, M – Mantissa.

<b>LSB</b> Byte 0	Byte 1	Byte 2	<b>MSB</b> Byte 3
----------------------	--------	--------	----------------------

**Unsigned Long Data Type (4 bytes):** Long type unsigned integer.

<b>LSB</b> Byte 0	Byte 1	Byte 2	<b>MSB</b> Byte 3
----------------------	--------	--------	----------------------

**Unsigned Int Data Type (2 bytes):** Unsigned integer variable.

<b>LSB</b> Byte 0	<b>MSB</b> Byte 1
----------------------	----------------------

**Unsigned Char (1 byte):** Single byte unsigned integer variable.

**Strings:** C has no string variable as such; strings output from the Transducer are in the format of an array of Char type (1 byte) integer values terminated with a NULL character.

**PeakMinMax:** A structure containing two Float data types

<b>Torque Peak Max</b>			
<b>LSB</b> Byte 0	Byte 1	Byte 2	<b>MSB</b> Byte 3
<b>Torque Peak Min</b>			
<b>LSB</b> Byte 4	Byte 5	Byte 6	<b>MSB</b> Byte 7





## **Torque Modes**

The primary purpose of the ORT/RWT Series Transducers is to measure Torque, the Torque value that is output from the Transducer is run through several processes within the firmware, these processes include a filter (if enabled), frequency to Torque rescaling, temperature correction and zero offset adjustment. The filter is a running average with a standard deviation cut off to remove spurious readings, the running average enables the sample throughput to be unaffected by filter size.

Once the final Torque value is computed it is run through a peak Torque capture algorithm. The peak Torque algorithm monitors the incoming data and compares it against a set of stored values using various criteria. If the value matches the criteria, that value replaces the stored value. In most cases the criterion is related to whether the captured value is greater than the stored value.

Peak values assume a reset position on start-up, when peak values are reset they are set to zero, PeakMinMax values are set to the current Torque value.

The peak Torque algorithm is run on every valid Torque reading captured, ensuring that no peak value is missed.

The Torque value unless specified will always be scaled in the native unit of measurement for the Transducer.

The following subsections describe the different types of peak Torque.

### ***Peak Torque***

The peak Torque value indicates the highest Torque applied to the transducer in either direction. The value is signed to indicate the direction that the Torque was applied in.

### ***Peak Torque with AutoReset***

The peak Torque with auto reset value is similar to the Peak Torque feature, it works in the same way by recording the maximum Torque, but automatically resets to zero when the current Torque value drops below a configured percentage of the peak value. When the reset triggers the peak is held for a few seconds before it is zeroed.

The default auto reset percentage is 80%; the percentage can be configured using the "Transducer Control" program which accompanies our ORT/RWT Series Transducers.

### ***Peak Torque CW***

The peak Torque CW value records the highest Torque value measured in the clockwise direction.

### ***Peak Torque CCW***

The peak torque CCW value records the highest Torque value measured in the counter-clockwise direction.

### ***PeakMinMax***

The PeakMinMax feature monitors the captured Torque values and records the lowest and highest value from a reference position. This reference is given via a reset command and assumes zero on power on. An example of the PeakMinMax feature is as follows: if the reference is set to 10, then the torque value goes up by 10 and down by 12, Max would be 20 and Min would be -2.

**Speed Modes**

Speed is decoded from a square wave signal, produced by a shaft mounted grating passing through an optical sensor. The frequency of the square wave indicates the rotational speed of the shaft. The transducer uses two methods for the measurement of Speed, both methods run simultaneously, offer good accuracy, but differ in measurement time. Speed is always measured in revolutions per minute (RPM).

**Slow**

The slow method uses a frequency count. Rising edges of the square wave are counted over a period of a second, after each second the count is converted into RPM. As the name suggests this method is slow, measurements will be produced at a rate of 1 a second. This method is good if you have a fluctuating drive speed and wish to filter the captured speed value.

**Fast**

The fast method uses a period count. The period count measures the time between rising edges of the square wave, then computes the RPM by turning the time into frequency. The fast methods measurement rate is variable and is directly related to the rotational speed of the transducer. When the rotational speed of the shaft rises above 2000 RPM the fast method will increase the number of rising edges over which time is measured, this is done to preserve measurement accuracy.

The fast methods measurement rate can be calculated from the following tables. The measurement rate differs between the different hardware revisions due to differing capabilities. The calculations shown are based on a standard 60 line grating.

*RWT320/340 (MKII)*

Rotational Speed (RPM)		Update Rate (Hz)
From	To	
0		1 Hz
1	2000	RPM / 2
2000	4000	$((\text{RPM} - 2000) \times 0.3227) + 650$
4000	8000	$((\text{RPM} - 4000) \times 0.196) + 800$
8000	16000	$((\text{RPM} - 8000) \times 0.1117) + 850$
16000	32000	$((\text{RPM} - 16000) \times 0.058) + 900$

*ORT240/RWT420/440 (MKIII)*

Rotational Speed (RPM)	Update Rate (Hz)
0	1 Hz
< 2000	RPM
> 2000	$\text{RPM} \times (1 / (\lfloor (\text{RPM} - 1) / 2000 \rfloor + 1))$

**Temperature Sensors**

The Transducer monitors Temperature from three different sensors, these are defined as ambient, shaft and internal. The shaft Temperature is the only one which is used for compensation; the other two are for monitoring only. The Transducer measures Temperature in degrees Celsius.

On some models not all sensors may be present; when a sensor is absent the value returned will be the shaft temperature.

**Ambient**

The ambient sensor is mounted in free air, stood off from the PCB it is mounted to.

**Shaft**

The shaft sensor is an infra-red device which is pointed directly at the centre of the shaft.

**Internal**

The internal sensor is part of the communications processor on the main processing board.

**Unit Key**

Some of the commands use a number to represent Transducer units or to indicate which units to convert to. The table below shows which number represents each unit, e.g. 7 = N.m.

Key Value	Unit
0	ozf.in
1	lbf.in
2	lbf.ft
3	gf.cm
4	Kgf.cm
5	Kgf.m
6	mN.m
7	N.m

### Technology Family Key

The protocol discussed in this manual covers multiple product families of different technologies. It is not necessary to know the actual technology as it does not affect the commands.

For information purposes the user may wish to decode the technology used, the table below shows which key values represent each technology.

Key Value	Family Designation	Technology Type
1	RWT	SAW based transducer
2	ORT	Optical based transducer
4	Strain Gauge	Strain Gauge based transducer
8	RWT External	SAW based transducer with external electronics
16	ORT External	Optical based transducer with external electronics

It is beyond the scope of this manual to discuss the different technologies and their merits, please refer to the individual product manuals for a technical description.

## Command Set

The table below outlines the commands available:

Command	Function	Parameters	Return Value
<b><i>Transducer Identification</i></b>			
0	Get Transducer ID	None	Transducer ID String
1	Get Transducer Information	None	Transducer Information
<b><i>Transducer Data</i></b>			
50	Get Torque	None	Torque
51	Get Peak Torque	None	Peak Torque
52	Get Peak Torque Auto Reset	None	Peak Torque Auto Reset
53	Get Peak Torque CW	None	Peak Torque CW
54	Get Peak Torque CCW	None	Peak Torque CCW
55	Get PeakMinMax Max	None	PeakMinMax Max
56	Get PeakMinMax Min	None	PeakMinMax Min
57	Get PeakMinMax	None	PeakMinMax
60	Get Torque Convert Unit To	Units	Torque
61	Get Peak Torque Convert Units To	Units	Peak Torque
62	Get Peak Torque Auto Reset Convert Units To	Units	Peak Torque Auto Reset
63	Get Peak Torque CW Convert Units To	Units	Peak Torque CW
64	Get Peak Torque CCW Convert Units To	Units	Peak Torque CCW
65	Get PeakMinMax Max Convert Units To	Units	PeakMinMax Max
66	Get PeakMinMax Min Convert Units To	Units	PeakMinMax Min
67	Get PeakMinMax Convert Units To	Units	PeakMinMax
100	Get Speed	None	Speed
101	Get Power	None	Power in Watts
102	Get Temperature Ambient	None	Temperature Ambient
103	Get Temperature Shaft	None	Temperature Shaft
110	Get SlowCap Speed	None	Speed
111	Get FastCap Speed	None	Speed
112	Get SlowCap Power in Watts	None	Power in Watts
113	Get FastCap Power in Watts	None	Power in Watts
114	Get SlowCap Power in HP	None	Power in HP
115	Get FastCap Power in HP	None	Power in HP
<b><i>Transducer Control</i></b>			
146	Reset specified Peaks	See Description.	
147	Reset all Peak Torque values	None	None
148	Reset all Peaks	None	None
149	Reset System Values	None	None
150	Peak Torque Reset	None	None
152	Peak Torque Auto Reset - Reset	None	None

155	Zero Transducer with Average	None	None
156	Zero Transducer	None	None
173	PeakMinMax Retrieve & Reset	None	PeakMinMax
180	Set Torque filter	Filter Setting	None
181	Get Torque filter	None	Filter Setting
182	Set Speed filter	Filter Setting	None
183	Get Speed filter	None	Filter Setting

## Transducer Identification

### *Get Transducer ID*

	Binary	ASCII
<b>Command</b>	0	#0;
<b>Description</b>	<p>Requests an ID string from the Transducer. The ID string contains the Transducer Model Name, Firmware Revision and Serial Number.</p> <p>The ID string has the following format:</p> <p style="text-align: center;">RWT321-DA - Firmware Revision: 2.1 Serial Number: 12345678</p>	
<b>Parameters</b>	None	
<b>Return Value</b>	Transducer ID String	
	Char [58]	#ID STRING;

**Get Transducer Information**

	Binary	ASCII																
<b>Command</b>	1	#1;																
<b>Description</b>	Requests information on the Transducers configuration. The information details the Transducers setup.																	
<b>Parameters</b>	None																	
<b>Return Value</b>	<p>The transducer returns a structure of values, either as a packed C structure for binary or a delimited list in ASCII. The returned data contains the following fields, dates have the following format, DD/MM/YYYY.</p> <p>Model Name - Transducer Model Name.                      Type - Transducer Family, (In binary mode see Family Key).                      FSD - Transducer Full Scale.                      Units - Transducer Torque Unit, (In binary mode see Unit Key).                      Maximum Speed - Maximum tested/rated speed.                      Serial Number - Transducer Serial Number.                      Manufacture Date - Transducer Manufacture Date.                      Calibration Date - Transducer Calibration Date.                      Options - Enabled features.</p> <p>The options value is a combination of binary flags, the table below shows its makeup. Bit=1 Enabled, Bit=0 Disabled</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th><b>Bit0 (L)</b></th> <th><b>Bit1</b></th> <th><b>Bit2</b></th> <th><b>Bit3</b></th> </tr> </thead> <tbody> <tr> <td>USB</td> <td>RS232</td> <td>Advanced User Control</td> <td>Current Output</td> </tr> <tr> <th><b>Bit4</b></th> <th><b>Bit5</b></th> <th><b>Bit6</b></th> <th><b>Bit7 (M)</b></th> </tr> <tr> <td>None</td> <td>Speed Encoder</td> <td>Angle Encoder</td> <td>IP65</td> </tr> </tbody> </table>		<b>Bit0 (L)</b>	<b>Bit1</b>	<b>Bit2</b>	<b>Bit3</b>	USB	RS232	Advanced User Control	Current Output	<b>Bit4</b>	<b>Bit5</b>	<b>Bit6</b>	<b>Bit7 (M)</b>	None	Speed Encoder	Angle Encoder	IP65
	<b>Bit0 (L)</b>	<b>Bit1</b>	<b>Bit2</b>	<b>Bit3</b>														
USB	RS232	Advanced User Control	Current Output															
<b>Bit4</b>	<b>Bit5</b>	<b>Bit6</b>	<b>Bit7 (M)</b>															
None	Speed Encoder	Angle Encoder	IP65															
	<pre>Struct { char Model_Name[10]; unsigned char Type; unsigned int FSD; unsigned char Units; unsigned long Max_Speed; char Serial_Number[9]; char Manufacture_Date[11]; char Calibration_Date[11]; unsigned char Options; }</pre>	<pre>#Model Name, Type, FSD, Units, Maximum Speed, Serial Number, Manufacture Date, Calibration Date, Options;  CR/LF added for readability.</pre>																

**Transducer Data**

***Get Torque***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	50	#50;
<b>Description</b>	Requests the current Torque value, if averaging has been enabled then this value is averaged.	
<b>Parameters</b>	None	
<b>Return Value</b>	Torque in the Transducers native units.	
	Float	#±0000000.000;

***Get Peak Torque***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	51	#51;
<b>Description</b>	Requests the current Peak Torque value.	
<b>Parameters</b>	None	
<b>Return Value</b>	Peak Torque in the Transducers native units.	
	Float	#±0000000.000;

***Get Peak Torque Auto Reset***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	52	#52;
<b>Description</b>	Requests the current Peak Torque Auto Reset value.	
<b>Parameters</b>	None	
<b>Return Value</b>	Peak Torque Auto Reset in the Transducers native units.	
	Float	#±0000000.000;



***Get Peak Torque CW***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	53	#53;
<b>Description</b>	Requests the current clockwise Peak Torque value.	
<b>Parameters</b>	None	
<b>Return Value</b>	Peak Torque value in the Transducers native units.	
	Float	#±0000000.000;

***Get Peak Torque CCW***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	54	#54;
<b>Description</b>	Requests the current counter-clockwise Peak Torque value.	
<b>Parameters</b>	None	
<b>Return Value</b>	Peak Torque value in the Transducers native units.	
	Float	#±0000000.000;

***Get PeakMinMax Max***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	55	#55;
<b>Description</b>	Requests the current Max value from the PeakMinMax data.	
<b>Parameters</b>	None	
<b>Return Value</b>	PeakMinMax Max Torque value in the transducers native units.	
	Float	#±0000000.000;

**Get PeakMinMax Min**

	Binary	ASCII
<b>Command</b>	56	#56;
<b>Description</b>	Requests the current Min value from the PeakMinMax data.	
<b>Parameters</b>	None	
<b>Return Value</b>	PeakMinMax Min Torque value in the transducers native units.	
	Float	#±0000000.000;

**Get PeakMinMax**

	Binary	ASCII
<b>Command</b>	57	#57;
<b>Description</b>	Requests the PeakMinMax data.	
<b>Parameters</b>	None	
<b>Return Value</b>	PeakMinMax data consisting of two Torque values in the transducers native units.	
	PeakMinMax structure	#Max,Min; #±0000000.000,±0000000.000;

**Get Torque - Convert Units To**

	Binary	ASCII
<b>Command</b>	60	#60, <b>UNITS</b> ;
<b>Description</b>	Requests the current Torque value and converts the native units to the selected units.	
<b>Parameters</b>	<b>UNITS</b> - The <b>UNITS</b> parameter specifies the Torque unit to convert the Torque value to. Use the Unit key to find the corresponding value for the unit required.	
	Unsigned Char	<b>UNITS</b> value as an additional parameter.
<b>Return Value</b>	Torque in the selected unit. Parameter is acknowledged in ASCII mode.	
	Float	#ACK,±0000000.000;

**Get Peak Torque - Convert Units To**

	Binary	ASCII
<b>Command</b>	61	#61, <b>UNITS</b> ;
<b>Description</b>	Requests the current Peak Torque value and converts the native units to the selected units.	
<b>Parameters</b>	<b>UNITS</b> - The <b>UNITS</b> parameter specifies the Torque unit to convert the Torque value to. Use the Unit key to find the corresponding value for the unit required.	
	Unsigned Char	<b>UNITS</b> value as an additional parameter.
<b>Return Value</b>	Peak Torque value in the selected unit. Parameter is acknowledged in ASCII mode.	
	Float	#ACK,±0000000.000;

**Get Peak Torque Auto Reset - Convert Units To**

	Binary	ASCII
<b>Command</b>	62	#62, <b>UNITS</b> ;
<b>Description</b>	Requests the current Peak Torque Auto Reset value and converts the native units to the selected units.	
<b>Parameters</b>	<b>UNITS</b> - The <b>UNITS</b> parameter specifies the Torque unit to convert the Torque value to. Use the Unit key to find the corresponding value for the unit required. Parameter is acknowledged in ASCII mode.	
	Unsigned Char	<b>UNITS</b> value as an additional parameter.
<b>Return Value</b>	Peak Torque Auto Reset value in the selected unit. Parameter is acknowledged in ASCII mode.	
	Float	#ACK,±0000000.000;

**Get Peak Torque CW - Convert Units To**

	Binary	ASCII
<b>Command</b>	63	#63, <b>UNITS</b> ;
<b>Description</b>	Requests the current clockwise Peak Torque value and converts the native units to the selected units.	
<b>Parameters</b>	<b>UNITS</b> - The <b>UNITS</b> parameter specifies the Torque unit to convert the Torque value to. Use the Unit key to find the corresponding value for the unit required.	
	Unsigned Char	<b>UNITS</b> value as an additional parameter.
<b>Return Value</b>	Peak Torque value in the selected unit. Parameter is acknowledged in ASCII mode.	
	Float	#ACK,±0000000.000;

**Get Peak Torque CCW - Convert Units To**

	Binary	ASCII
<b>Command</b>	64	#64, <b>UNITS</b> ;
<b>Description</b>	Requests the current counter-clockwise Peak Torque value and converts the native units to the selected units.	
<b>Parameters</b>	<b>UNITS</b> - The <b>UNITS</b> parameter specifies the Torque unit to convert the Torque value to. Use the Unit key to find the corresponding value for the unit required.	
	Unsigned Char	<b>UNITS</b> value as an additional parameter.
<b>Return Value</b>	Peak Torque value in the selected unit. Parameter is acknowledged in ASCII mode.	
	Float	#ACK,±0000000.000;

**Get PeakMinMax Max - Convert Units To**

	Binary	ASCII
<b>Command</b>	65	#65, <b>UNITS</b> ;
<b>Description</b>	Requests the current Max value from the PeakMinMax data and converts the native units to the selected units.	
<b>Parameters</b>	<b>UNITS</b> - The <b>UNITS</b> parameter specifies the Torque unit to convert the Torque value to. Use the Unit key to find the corresponding value for the unit required.	
	Unsigned Char	<b>UNITS</b> value as an additional parameter.
<b>Return Value</b>	PeakMinMax Max Torque value in the selected unit. Parameter is acknowledged in ASCII mode.	
	Float	#ACK,±0000000.000;

**Get PeakMinMax Min - Convert Units To**

	Binary	ASCII
<b>Command</b>	66	#66, <b>UNITS</b> ;
<b>Description</b>	Requests the current Min value from the PeakMinMax data and converts the native units to the selected units.	
<b>Parameters</b>	<b>UNITS</b> - The <b>UNITS</b> parameter specifies the Torque unit to convert the Torque value to. Use the Unit key to find the corresponding value for the unit required.	
	Unsigned Char	<b>UNITS</b> value as an additional parameter.
<b>Return Value</b>	PeakMinMax Min Torque value in the selected unit. Parameter is acknowledged in ASCII mode.	
	Float	#ACK,±0000000.000;

**Get PeakMinMax - Convert Units To**

	Binary	ASCII
<b>Command</b>	67	#67, <b>UNITS</b> ;
<b>Description</b>	Requests the PeakMinMax data and converts the native units to the selected units.	
<b>Parameters</b>	<b>UNITS</b> - The <b>UNITS</b> parameter specifies the Torque unit to convert the Torque value to. Use the Unit key to find the corresponding value for the unit required.	
	Unsigned Char	<b>UNITS</b> value as an additional parameter.
<b>Return Value</b>	PeakMinMax data consisting of two Torque values in the selected unit. Parameter is acknowledged in ASCII mode.	
	PeakMinMax structure	#ACK,Max,Min;  #ACK, ±0000000.000, ±0000000.000;  CR/LF added for readability.

**Get Speed**

	Binary	ASCII
<b>Command</b>	100	#100;
<b>Description</b>	Requests the current Speed value. The slow speed capture mode is used.	
<b>Parameters</b>	None	
<b>Return Value</b>	Speed in RPM.	
	Float	#±0000000.000;

***Get Power***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	101	#101;
<b>Description</b>	Requests the current Power value in Watts. The slow speed capture mode is used.	
<b>Parameters</b>	None	
<b>Return Value</b>	Power in Watts.	
	Float	#±0000000.000;

***Get Temperature Ambient***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	102	#102;
<b>Description</b>	Requests the Transducers internal ambient Temperature.	
<b>Parameters</b>	None	
<b>Return Value</b>	Temperature in degrees C (°C).	
	Float	#±0000000.000;

***Get Temperature Shaft***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	103	#103;
<b>Description</b>	Requests the Transducers shaft Temperature.	
<b>Parameters</b>	None	
<b>Return Value</b>	Temperature in degrees C (°C).	
	Float	#±0000000.000;

***Get SlowCap Speed***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	110	#110;
<b>Description</b>	Requests the current Speed value from the slow speed capture system.	
<b>Parameters</b>	None	
<b>Return Value</b>	Speed in RPM.	
	Unsigned Long	#±0000000.000;

***Get FastCap Speed***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	111	#111;
<b>Description</b>	Requests the current Speed value from the fast speed capture system.	
<b>Parameters</b>	None	
<b>Return Value</b>	Speed in RPM.	
	Unsigned Long	±0000000.000

***Get SlowCap Power in Watts***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	112	#112;
<b>Description</b>	Requests the current Power value in Watts, using the current torque and speed value from the slow speed capture system.	
<b>Parameters</b>	None	
<b>Return Value</b>	Power in Watts.	
	Float	#±0000000.000;



***Get FastCap Power in Watts***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	113	#113;
<b>Description</b>	Requests the current Power value in Watts, using the current torque and speed value from the fast speed capture system.	
<b>Parameters</b>	None	
<b>Return Value</b>	Power in Watts.	
	Float	#±0000000.000;

***Get SlowCap Power in HP***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	114	#114;
<b>Description</b>	Requests the current Power value in Horse Power (HP), using the current torque and speed value from the slow speed capture system.	
<b>Parameters</b>	None	
<b>Return Value</b>	Power in HP.	
	Float	#±0000000.000;

***Get FastCap Power in HP***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	115	#115;
<b>Description</b>	Requests the current Power value in Horse Power (HP), using the current torque and speed value from the fast speed capture system.	
<b>Parameters</b>	None	
<b>Return Value</b>	Power in HP.	
	Float	#±0000000.000;

## Transducer Control

### *Reset Specified Peaks*

	Binary	ASCII
<b>Command</b>	146	#146, <b>FLAGS</b> ;
<b>Description</b>	<p>Resets the stored peak values or zero's the transducer in accordance with binary flags specified as an additional parameter.</p> <p>The ASCII implementation of this command is a straight forward request with a single parameter.</p> <p>The binary implementation requires some handshaking which differs from the standard request.</p> <p>The parameter value is an Unsigned Int which is 2 bytes, because of this some handshaking is required to ensure that no overflows occur in the Transducer's internal processor.</p> <p>The procedure for transmitting the value is outlined below:</p> <ol style="list-style-type: none"> <li>1. Transmit the command byte (Unsigned Char) with a value of 146.</li> <li>2. Receive a byte (Unsigned Char); the byte will have a value of 145, the value has no significance.</li> <li>3. Transmit the 2 byte (Unsigned Int) reset request parameter, LSB first.</li> <li>4. Receive a byte (Unsigned Char), this second byte acts as a confirmation, its value will be 145, again the value has no significance.</li> </ol>	

<b>Parameters</b>	<p><b>FLAGS</b> - The <b>FLAGS</b> parameter specifies which stored values should be reset. The value is made up of binary flags, each flag signifying a value to reset.</p> <p>The input value is calculated by adding together or OR'ing the flag values. The table below shows the reset flags and their respective values.</p>		
	<b>Flag Value</b>	<b>Value to be reset</b>	<b>Description</b>
	0x01	Transducer Zero	Zero's the Transducer.
	0x02	Transducer Zero with Average	Zero's the Transducer with an average value.
	0x04	Peak Torque	Resets the Peak Torque to zero.
	0x08	Peak Torque Auto Reset	Resets the Peak Torque Auto Reset to zero.
	0x10	Peak Torque CW	Resets the Peak Torque CW to zero.
	0x20	Peak Torque CCW	Resets the Peak Torque CCW to zero.
	0x40	PeakMinMax	Resets the Min and Max values to the current torque value.
	0x80	Peak FastCap Speed	Resets the Peak FastCap Speed value to zero.
	0x100	Peak SlowCap Speed	Resets the Peak SlowCap Speed value to zero.
	0x200	Peak FastCap Power	Resets the Peak FastCap Power value to zero.
0x400	Peak SlowCap Power	Resets the Peak SlowCap Power value to zero.	
	<p><i>Example:</i></p> <p>To reset all the torque values (Peak Torque: 0x04, Peak Torque Auto Reset: 0x08, Peak Torque CW: 0x10, Peak Torque CCW 0x20, PeakMinMax: 0x40), the input value would be 0x7C.</p> <p><i>Input Value:</i> 0x7C = 0x04 + 0x08 + 0x10 + 0x20 + 0x40.</p>		
	Unsigned Int	<b>FLAGS</b> value as an additional parameter.	
<b>Return Value</b>	See Description.	#ACK;	

***Reset ALL Peak Torque Values***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	147	#147;
<b>Description</b>	Resets the entire memory bank of stored peak values related to Torque, values are reset to zero, except for the PeakMinMax values which are reset to the current Torque value.	
<b>Parameters</b>	None	
<b>Return Value</b>	None	#ACK;

***Reset ALL Peak Values***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	148	#148;
<b>Description</b>	Resets the entire memory bank of stored peak values related to Torque, Speed and Power. Values are reset to zero, except for the PeakMinMax values which are reset to the current Torque value.	
<b>Parameters</b>	None	
<b>Return Value</b>	None	#ACK;

***Reset System Values***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	149	#149;
<b>Description</b>	Resets the entire memory bank of stored peak values related to Torque, Speed and Power, then zero's the transducer using an averaged zero. All subsequent Torque values are offset by the zero value. Peak values are reset to zero, except for the PeakMinMax values which are reset to the current Torque value.	
<b>Parameters</b>	None	
<b>Return Value</b>	None	#ACK;

***Peak Torque Reset***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	150	#150;
<b>Description</b>	Resets the stored Peak Torque value to zero.	
<b>Parameters</b>	None	
<b>Return Value</b>	None	#ACK;

***Peak Torque Auto Reset - Reset***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	152	#152;
<b>Description</b>	Resets the stored Peak Torque AutoReset value to zero.	
<b>Parameters</b>	None	
<b>Return Value</b>	None	#ACK;

***Zero Transducer with Average***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	155	#155;
<b>Description</b>	Zero's the Transducer Torque value; all subsequent Torque readings will be offset by an averaged torque value. When the command is sent to the transducer, the firmware will average over the next 32 Torque samples, the averaged Torque value is then stored as the zero offset.	
<b>Parameters</b>	None	
<b>Return Value</b>	None	#ACK;

***Zero Transducer***

	<b>Binary</b>	<b>ASCII</b>
<b>Command</b>	156	#156;
<b>Description</b>	Zero's the Transducer Torque value; all subsequent Torque readings will be offset by the torque value present when zeroed.	
<b>Parameters</b>	None	
<b>Return Value</b>	None	#ACK;

**Get PeakMinMax & Reset**

	Binary	ASCII
<b>Command</b>	173	#173;
<b>Description</b>	Requests the PeakMinMax data and resets the stored values to the current Torque reading.	
<b>Parameters</b>	None	
<b>Return Value</b>	PeakMinMax data consisting of two Torque values in the transducers native units.  The reset is acknowledged in ASCII mode.	
	PeakMinMax structure	#Max,Min,ACK;  #±0000000.000, ±0000000.000, ACK;  CR/LF added for readability.

**Set Torque Filter**

	Binary	ASCII
<b>Command</b>	180	#180, <b>FILTER</b> ;
<b>Description</b>	Enables and configures the Torque filtering system.	
<b>Parameters</b>	<b>FILTER</b> - Torque filter setting. Disables or sets the filtering sample level.  Valid filter settings:  0 – OFF, 2, 4, 8, 16, 32, 64, 128, 256 (In binary format send 255 for 256).	
	Unsigned Char	<b>FILTER</b> value as an additional parameter.
<b>Return Value</b>	None	#ACK;

**Get Torque Filter**

	Binary	ASCII
<b>Command</b>	181	#181;
<b>Description</b>	Retrieves the current Torque filter setting.	
<b>Parameters</b>	None	
<b>Return Value</b>	Torque filter setting. The value returned indicates the filter level. If zero is received the filter is disabled. In binary format a filter level of 256 is output as 255.	
	Unsigned Char	#000;

**Set Speed Filter**

	Binary	ASCII
<b>Command</b>	182	#182, <b>FILTER</b> ;
<b>Description</b>	Enables and configures the Speed filtering system.	
<b>Parameters</b>	<b>FILTER</b> - Speed filter setting. Disables or sets the filtering sample level.  Valid filter settings: 0 – OFF, 2, 4, 8, 16, 32, 64, 128, 256 (In binary format send 255 for 256).	
	Unsigned Char	<b>FILTER</b> value as an additional parameter.
<b>Return Value</b>	None	#ACK;

**Get Speed Filter**

	Binary	ASCII
<b>Command</b>	183	#183;
<b>Description</b>	Retrieves the current Speed filter setting.	
<b>Parameters</b>	None	
<b>Return Value</b>	Speed filter setting. The value returned indicates the filter level. If zero is received the filter is disabled. In binary format a filter level of 256 is output as 255.	
	Unsigned Char	#000;