SENSOR
TECHNOLOGY LTD

RWT420/440 Series Transducer Communication Protocol

Revision 5 - October 2014
RWT320/340 MKII (Firmware Revision: 3)
RWT420/440 MKIII (Firmware Revision: 4)

Table of Contents

Contact Details

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

Sales
Email: stlsales@sensors.co.uk
Tel: +44 (0)1869 238400
Fax: +44 (0)1869 238401

Technical Support
Email: software@sensors.co.uk
Tel: +44 (0)1295 730746
Fax: +44 (0)1295 738966

Introduction

The RS232 and USB interfaces on the RWT300/400 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 STCOMMDLL DLL be used to interface your own application to the Transducer. Further assistance can be provided if USB is needed on an unsupported platform, or the developer wishes to interface directly without the DLL.

RS232 Settings

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 RWT300/400 Series Transducers**.**

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.

LSB Byte 0	Byte 1	Byte 2	**MSB** Byte 3

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

LSB Byte 0	Byte 1	Byte 2	**MSB** Byte 3

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

LSB Byte 0	**MSB** 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.

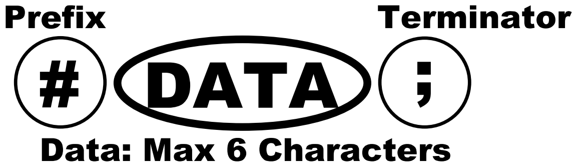
PeakMinMax: A structure containing two Float data types

Torque Peak Max			
LSB Byte 0	Byte 1	Byte 2	**MSB** Byte 3
Torque Peak Min			
LSB Byte 4	Byte 5	Byte 6	**MSB** Byte 7

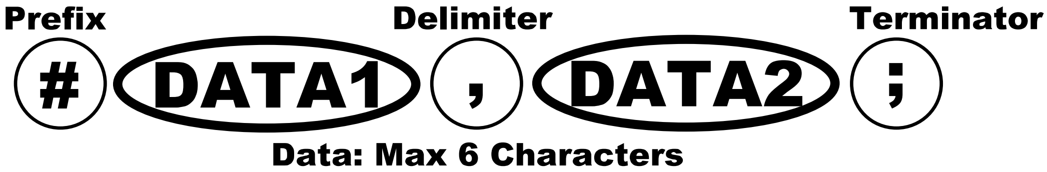
ASCII Format

The ASCII format is a straight forward way of exchanging data with a Transducer. All data exchanges are in a human readable format, which enables the use of a terminal program to interact with a Transducer. The draw back of using the ASCII format is the increased overhead of encoding the binary data, as well as the additional command formatting required.

All data exchanges consist of a data request and a response. Each message either to or from the Transducer should be formatted in the following way:



Each message should start with a "#" prefix and end with a ";" terminator, if additional values are to be transmitted or received, each value should be separated with "," delimiter.



To request data or an action, transmit a message with the first DATA field set to the command number of the function required, the Transducer will then reply with the relevant data or action your request, some commands may require extra parameters.

Each message transaction must be completed within 5 seconds, if it is not the input buffer will be purged and a "#NAK;" will be transmitted. The timer is started as soon as a "#" is received.

Requests that do not result in a data response are acknowledged with an "ACK;". Any formatting or validation errors detected in transmitted messages will receive a "#NAK;".

Requests do not need to be any specific length as long as each field does not exceed 6 characters or contain invalid characters. Command responses vary in length depending on the command, but each individual command will always respond with a fixed length response.

Torque, Speed, Power and Temperature requests all output the same number format. The format used is ±0000000.000, the first character is the sign, while this is irrelevant for Speed and Power values it is still output. The sign in Torque readings indicates Torque direction, negative is counter-clockwise, while positive is clockwise.

To illustrate how to request data from the transducer, the following example shows how to request Torque from a Transducer.

Request Torque:	#50;
Transducer Response:	#+0000000.390;

Torque Modes

The primary purpose of the 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 though 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 RWT300/400 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 RWT300 series and RWT400 series because of different implementations. 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	((RPM – 2000) x 0.3227) + 650
4000	8000	((RPM – 4000) x 0.196) + 800
8000	16000	((RPM – 8000) x 0.1117) + 850
16000	32000	((RPM – 16000) x 0.058) + 900

RWT420/440 (MKIII)

Rotational Speed (RPM)	**Update Rate (Hz)**
0	1 Hz
< 2000	RPM
> 2000	RPM x (1 / (⌊ (RPM - 1) / 2000 ⌋ + 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.

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.

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

Command Set

The table below outlines the commands available:

Command	Function	Parameters	Return Value
Transducer Identification			
0	Get Transducer ID	None	Transducer ID String
1	Get Transducer Information	None	Transducer Information
Transducer Data			
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
Transducer Control			
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**
Command	0	#0;
Description	Requests an ID string from the Transducer. The ID string contains the Transducer Model Name, Firmware Revision and Serial Number. The ID string has the following format: RWT321-DA - Firmware Revision: 2.1 Serial Number: 12345678	
Parameters	None	
Return Value	Transducer ID String	
	Char [58]	#ID STRING;

Get Transducer Information

<table>
<tr><th></th><th>Binary</th><th>ASCII</th></tr>
<tr><td><b>Command</b></td><td>1</td><td>#1;</td></tr>
<tr><td><b>Description</b></td><td colspan="2">Requests information on the Transducers configuration. The information details the Transducers setup.</td></tr>
<tr><td><b>Parameters</b></td><td colspan="2">None</td></tr>
<tr><td rowspan="2"><b>Return Value</b></td><td colspan="2">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.

Model Name - Transducer Model Name.
Type - Transducer Technology, reserved for future use.
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.

The options value is a combination of binary flags, the table below shows its makeup. Bit=1 Enabled, Bit=0 Disabled
<table>
<tr><th><i>Bit0 (L)</i></th><th><i>Bit1</i></th><th><i>Bit2</i></th><th><i>Bit3</i></th></tr>
<tr><td>USB</td><td>RS232</td><td>Advanced User Control</td><td>Current Output</td></tr>
<tr><td><b><i>Bit4</i></b></td><td><b><i>Bit5</i></b></td><td><b><i>Bit6</i></b></td><td><b><i>Bit7 (M)</i></b></td></tr>
<tr><td>None</td><td>Speed Encoder</td><td>Angle Encoder</td><td>IP65</td></tr>
</table>
</td></tr>
<tr><td>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;
}</td><td>#Model Name,
Type,
FSD,
Units,
Maximum Speed,
Serial Number,
Manufacture Date,
Calibration Date,
Options;

CR/LF added for readability.</td></tr>
</table>

Transducer Data

Get Torque

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

Get Peak Torque

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

Get Peak Torque Auto Reset

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

Get Peak Torque CW

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

Get Peak Torque CCW

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

Get PeakMinMax Max

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

Get PeakMinMax Min

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

Get PeakMinMax

	Binary	**ASCII**
Command	57	#57;
Description	Requests the PeakMinMax data.	
Parameters	None	
Return Value	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**
Command	60	#60,***UNITS***;
Description	Requests the current Torque value and converts the native units to the selected units.	
Parameters	***UNITS*** - The ***UNITS*** 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	***UNITS*** value as an additional parameter.
Return Value	Torque in the selected unit. Parameter is acknowledged in ASCII mode.	
	Float	#ACK,±0000000.000;

Get Peak Torque - Convert Units To

	Binary	**ASCII**
Command	61	#61,***UNITS***;
Description	Requests the current Peak Torque value and converts the native units to the selected units.	
Parameters	***UNITS*** - The ***UNITS*** 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	***UNITS*** value as an additional parameter.
Return Value	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**
Command	62	#62,***UNITS***;
Description	Requests the current Peak Torque Auto Reset value and converts the native units to the selected units.	
Parameters	***UNITS*** - The ***UNITS*** 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	***UNITS*** value as an additional parameter.
Return Value	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**
Command	63	#63,***UNITS***;
Description	Requests the current clockwise Peak Torque value and converts the native units to the selected units.	
Parameters	***UNITS*** - The ***UNITS*** 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	***UNITS*** value as an additional parameter.
Return Value	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**
Command	64	#64,***UNITS***;
Description	Requests the current counter-clockwise Peak Torque value and converts the native units to the selected units.	
Parameters	***UNITS*** - The ***UNITS*** 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	***UNITS*** value as an additional parameter.
Return Value	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**
Command	65	#65,***UNITS***;
Description	Requests the current Max value from the PeakMinMax data and converts the native units to the selected units.	
Parameters	***UNITS*** - The ***UNITS*** 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	***UNITS*** value as an additional parameter.
Return Value	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**
Command	66	#66,***UNITS***;
Description	Requests the current Min value from the PeakMinMax data and converts the native units to the selected units.	
Parameters	***UNITS*** - The ***UNITS*** 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	***UNITS*** value as an additional parameter.
Return Value	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**
Command	67	#67,***UNITS***;
Description	Requests the PeakMinMax data and converts the native units to the selected units.	
Parameters	***UNITS*** - The ***UNITS*** 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	***UNITS*** value as an additional parameter.
Return Value	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**
Command	100	#100;
Description	Requests the current Speed value. The slow speed capture mode is used.	
Parameters	None	
Return Value	Speed in RPM.	
	Float	#±0000000.000;

Get Power

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

Get Temperature Ambient

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

Get Temperature Shaft

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

Get SlowCap Speed

	Binary	**ASCII**
Command	110	#110;
Description	Requests the current Speed value from the slow speed capture system.	
Parameters	None	
Return Value	Speed in RPM.	
	Unsigned Int	#±0000000.000;

Get FastCap Speed

	Binary	**ASCII**
Command	111	#111;
Description	Requests the current Speed value from the fast speed capture system.	
Parameters	None	
Return Value	Speed in RPM.	
	Unsigned Int	±0000000.000

Get SlowCap Power in Watts

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

Get FastCap Power in Watts

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

Get SlowCap Power in HP

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

Get FastCap Power in HP

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

Transducer Control

Reset Specified Peaks

	Binary	**ASCII**
Command	146	#146,***FLAGS***;
Description	Resets the stored peak values or zero's the transducer in accordance with binary flags specified as an additional parameter. The ASCII implementation of this command is a straight forward request with a single parameter. The binary implementation requires some handshaking which differs from the standard request. 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. The procedure for transmitting the value is outlined below: 1. Transmit the command byte (Unsigned Char) with a value of 146. 2. Receive a byte (Unsigned Char); the byte will have a value of 145, the value has no significance. 3. Transmit the 2 byte (Unsigned Int) reset request parameter, LSB first. 4. Receive a byte (Unsigned Char), this second byte acts as a confirmation, its value will be 145, again the value has no significance.	

<table>
<tr>
<td rowspan="2"><b>Parameters</b></td>
<td colspan="2">
<b><i>FLAGS</i></b> - The <b><i>FLAGS</i></b> parameter specifies which stored values should be reset. The value is made up of binary flags, each flag signifying a value to reset.

The input value is calculated by adding together or OR'ing the flag values. The table below shows the reset flags and there respective values.

<table>
<tr><th>Flag Value</th><th>Value to be reset</th><th>Description</th></tr>
<tr><td>0x01</td><td>Transducer Zero</td><td>Zero's the Transducer.</td></tr>
<tr><td>0x02</td><td>Transducer Zero with Average</td><td>Zero's the Transducer with an average value.</td></tr>
<tr><td>0x04</td><td>Peak Torque</td><td>Resets the Peak Torque to zero.</td></tr>
<tr><td>0x08</td><td>Peak Torque Auto Reset</td><td>Resets the Peak Torque Auto Reset to zero.</td></tr>
<tr><td>0x10</td><td>Peak Torque CW</td><td>Resets the Peak Torque CW to zero.</td></tr>
<tr><td>0x20</td><td>Peak Torque CCW</td><td>Resets the Peak Torque CCW to zero.</td></tr>
<tr><td>0x40</td><td>PeakMinMax</td><td>Resets the Min and Max values to the current torque value.</td></tr>
<tr><td>0x80</td><td>Peak FastCap Speed</td><td>Resets the Peak FastCap Speed value to zero.</td></tr>
<tr><td>0x100</td><td>Peak SlowCap Speed</td><td>Resets the Peak SlowCap Speed value to zero.</td></tr>
<tr><td>0x200</td><td>Peak FastCap Power</td><td>Resets the Peak FastCap Power value to zero.</td></tr>
<tr><td>0x400</td><td>Peak SlowCap Power</td><td>Resets the Peak SlowCap Power value to zero.</td></tr>
</table>

<i>Example:</i>

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.

<i>Input Value:</i> 0x7C = 0x04 + 0x08 + 0x10 + 0x20 + 0x40.
</td>
</tr>
<tr>
<td>Unsigned Int</td>
<td><b><i>FLAGS</i></b> value as an additional parameter.</td>
</tr>
<tr>
<td><b>Return Value</b></td>
<td>See Description.</td>
<td>#ACK;</td>
</tr>
</table>

Reset ALL Peak Torque Values

	Binary	**ASCII**
Command	147	#147;
Description	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.	
Parameters	None	
Return Value	None	#ACK;

Reset ALL Peak Values

	Binary	**ASCII**
Command	148	#148;
Description	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.	
Parameters	None	
Return Value	None	#ACK;

Reset System Values

	Binary	**ASCII**
Command	149	#149;
Description	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.	
Parameters	None	
Return Value	None	#ACK;

Peak Torque Reset

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

Peak Torque Auto Reset - Reset

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

Zero Transducer with Average

	Binary	**ASCII**
Command	155	#155;
Description	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.	
Parameters	None	
Return Value	None	#ACK;

Zero Transducer

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

Get PeakMinMax & Reset

	Binary	**ASCII**
Command	173	#173;
Description	Requests the PeakMinMax data and resets the stored values to the current Torque reading.	
Parameters	None	
Return Value	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**
Command	180	#180,***FILTER***;
Description	Enables and configures the Torque filtering system.	
Parameters	***FILTER*** - 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	***FILTER*** value as an additional parameter.
Return Value	None	#ACK;

Get Torque Filter

	Binary	**ASCII**
Command	181	#181;
Description	Retrieves the current Torque filter setting.	
Parameters	None	
Return Value	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**
Command	182	#182,***FILTER***;
Description	Enables and configures the Speed filtering system.	
Parameters	***FILTER*** - 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	***FILTER*** value as an additional parameter.
Return Value	None	#ACK;

Get Speed Filter

	Binary	**ASCII**
Command	183	#183;
Description	Retrieves the current Speed filter setting.	
Parameters	None	
Return Value	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;