

TMS Toolkit v1.2.3  
**TMS9000 Torque Measurement**

*Rev C: Nov 2006*  
*008-0689-00*

## Overview

The **TMS Toolkit** is a software tool that allows communication with the TMS 9000 system using the ASCII XP protocol.

This software can be used to:

- See a list of supported parameters and commands
- Read back the values of readable parameters
- Write values to writeable parameters
- Execute commands
- Retrieve a list of readable parameter names and values which can then be copied to a file
- Calibrate the device
- Perform simple repetitive communications requests

You would use this software to:

- Test communications with devices
- Configure devices
- Calibrate devices

## System Requirements

- The **TMS Toolkit** software will work on Windows 95, 98, ME, NT, 2000 and XP.
- Windows 95 and NT will require Internet Explorer 5.0 or higher installed and NT will require service pack 5 or higher installed
- The screen must support and be set for a resolution of 800 x 600 or higher.
- NOTE – Ensure that NT/Win2000/WinXP users have “write” permissions for the relevant program files folder (usually \Program Files\TMS Toolkit)

## **“Calibration” user mode**

The TMS Toolkit offers two modes of user access, namely the “normal” user mode and the “calibration” (or “cal”) user mode. These modes differ in the level of accessibility to system functions such as calibration and scaling.

The “normal” user is the default mode, and in this mode, the user can set up communications with a TMS 9000 system and can also set zero and change filter settings.

The “calibration” user mode requires a special command to gain access and then all commands and settings become available for edit/recall.

For ease of reference, all of the functions associated with the “calibration” user are arranged at the end of this document. All of the sections at the beginning of this document apply equally to both user modes.

NOTE – this dual-mode functionality is intended as a means to reduce the incidence of unintentional access to key parameters and is not intended as nor should it be used as or substituted for a high security protection system.

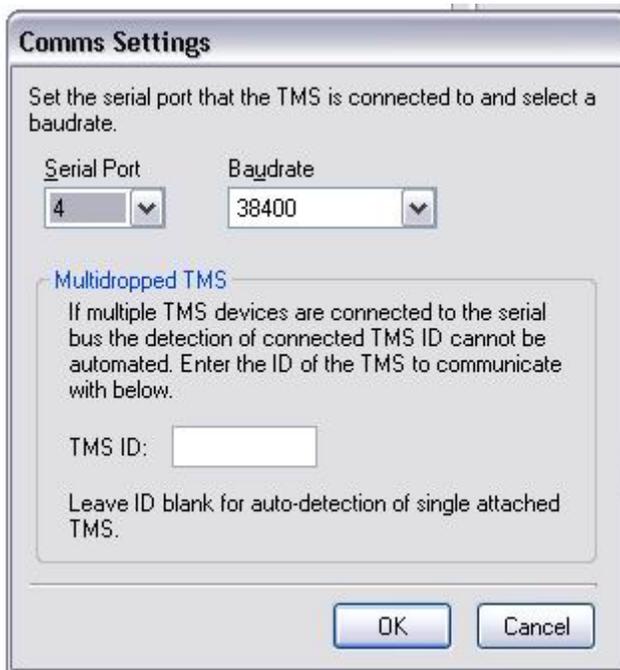
## Installation

The TMS Toolkit is supplied on CD-R or may be downloaded from [www.honeywell.com/sensing](http://www.honeywell.com/sensing). If using the CD-R version, the installation program will run automatically when the disk is loaded. In case it does not, then the SETUP program can be located using My Computer or Windows Explorer. In the case of a download from the website, the program can be saved locally and then run from the local copy, or it can be run from the website location with equal success. Before installing, be sure to remove any earlier versions by using the Uninstall programs that were provided with the earlier versions. Note – some earlier versions were named SPM Toolkit.

## Initial Configuration

To test the serial port setting and the TMS 9000 itself, connect the TMS 9000 to one of the PCs serial ports. If using a PC, which does not have a RS232 serial port, then a high quality USB-to-RS232 converter can be used instead. Note that such converters may have limitations such as the use of short cable lengths or special cable types to ensure reliable communications at 38400 baud. If in doubt, use a PC that has a true RS232 serial port built in to confirm communications and then try a laptop afterwards.

This example will assume that there is only one TMS 9000 connected to the serial bus. See Multi-Dropped TMS section for more information when using more than one TMS 9000 system on the same serial bus.



Click the **SETTINGS** button. This will open the **Comms Settings** dialog window as shown.

Select the **serial port** that the TMS is connected to (can be found from Windows Control Panel, System, Hardware, Ports) and select the **baudrate**. The default baudrate for standard TMS 9000 systems is 38400 (non-standard systems will be delivered with appropriate special instructions).

Ensure that the **TMS ID** field is left blank unless there are multiple TMS 9000 devices connected to the same serial bus, in which case the ID must be set to correspond with the ID of the device of interest.

Click **OK** when changes have been made, then click the **CONNECT** button.

The TMS Toolkit searches the designated serial port to find the connected device. The software takes care of configuring the port to suit the requirements of the TMS 9000.

## Window Sections

### Communications



**Settings** - This button will allow the serial port and baudrate to be configured.

**Connect** - This will automatically connect to the local **TMS** device if the TMS ID can be determined.

**Disconnect** - This will stop the communication between the connected device and the serial port.

Note that all of the windows can be re-sized by clicking and dragging on the dividing borders.

### Device Information



After successfully connecting, the Toolkit will display the serial number of the connected system along with the model, software version and default parameter name. The default parameter is the one that will be selected for display on the **TEST COMMS** tab until over-ridden by a selection from the drop down parameter table.

Clicking the button on the header will copy the information to the clipboard.

### Packet Counts

Packet Counts			
Packets Out	Packets In	In Per Sec	Error Count
5338	5338	0	188

Each time the software communicates with the remote device the Packet Counts section is updated. This lists the number of packets sent out and received, and also indicates the number of packets received per second and the number of errors that have occurred. When an error occurs, details are given in the **ERROR LIST**.

Clicking the button on the header will reset the counts.

## Traffic



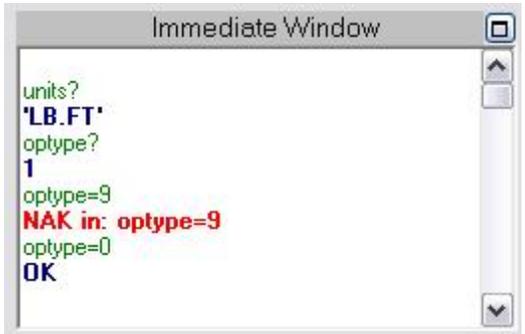
All communications to and from the remote device are listed in the Traffic window. The outgoing data is colored green and the incoming data is colored blue.

Note that when packets are retried this is a hidden function of the TMS device and only one packet will be displayed in this window.

This is very useful for inspecting the construction of data packets when writing your own communications software. It can also serve as a short-term data logging facility because the data can be copied to the clipboard and pasted into Excel for analysis.

Clicking the button on the header will clear the window.

## Immediate Window



The immediate window allows you to communicate directly with a connected remote device. Just type the data directly into the window, without the need for using any special addressing or special characters, then press the enter/return key. The list of valid commands/queries can be found in Appendix A.

If the device responds the data will be displayed here in blue and any errors will be indicated in red.

**Example:** To query the current torque value, enter

**VALUE?**

To set the current torque value to zero, enter

**ZERONOW**

Clicking the button on the header will clear the window.

## Error List



Any errors that are encountered when sending commands or queries, and errors in communications such as timeouts etc will be reported in the Error List in red.

Clicking the button on the header will clear the window.

## Multi-dropped TMS

Multiple TMS 9000's can be connected to the same serial bus and port when that port is either an RS485 port or has an RS232 to RS485 converter attached.

When more than one TMS 9000 is present the auto detection of TMS ID will not work. It is necessary to declare the ID of the TMS 9000 device that the communication is intended to be routed to.

This is done in the COMMS SETTINGS window by entering the ID of the TMS 9000 system (in the form XXXXXX) in the field labeled TMS ID.

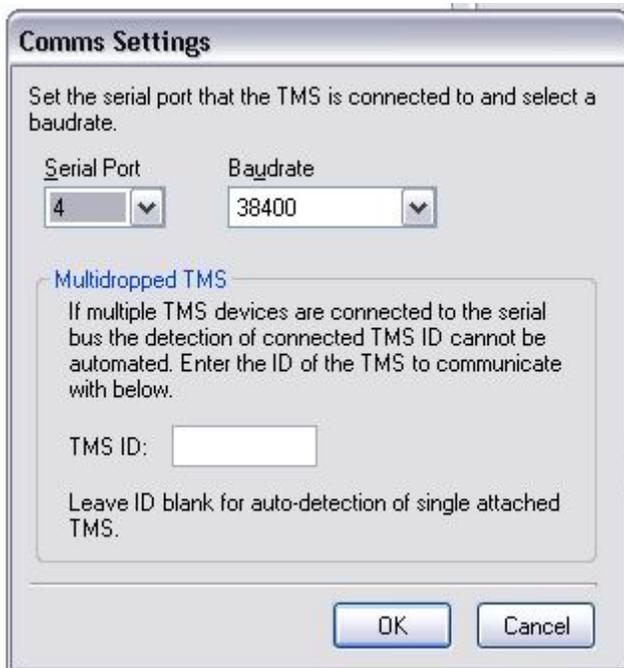


**Settings** - This button will allow the serial port and baudrate to be configured.

**Connect** - This will automatically connect to the local **TMS** device if the TMS ID can be determined.

**Disconnect** - This will stop the communication between the connected device and the serial port.

Note that all of the windows can be re-sized by clicking and dragging on the dividing borders.



Click the **SETTINGS** button. This will open the **Comms Settings** dialog window as shown.

Select the **serial port** that the TMS is connected to (can be found from Windows Control Panel, System, Hardware, Ports) and select the **baudrate**. The default baudrate for standard TMS 9000 systems is 38400 (non-standard systems will be delivered with appropriate special instructions).

Enter the **TMS ID** (printed on the yellow label attached to the motherboard inside the signal processing module) in the form FFFFFFFF, to correspond with the ID of the device of interest.

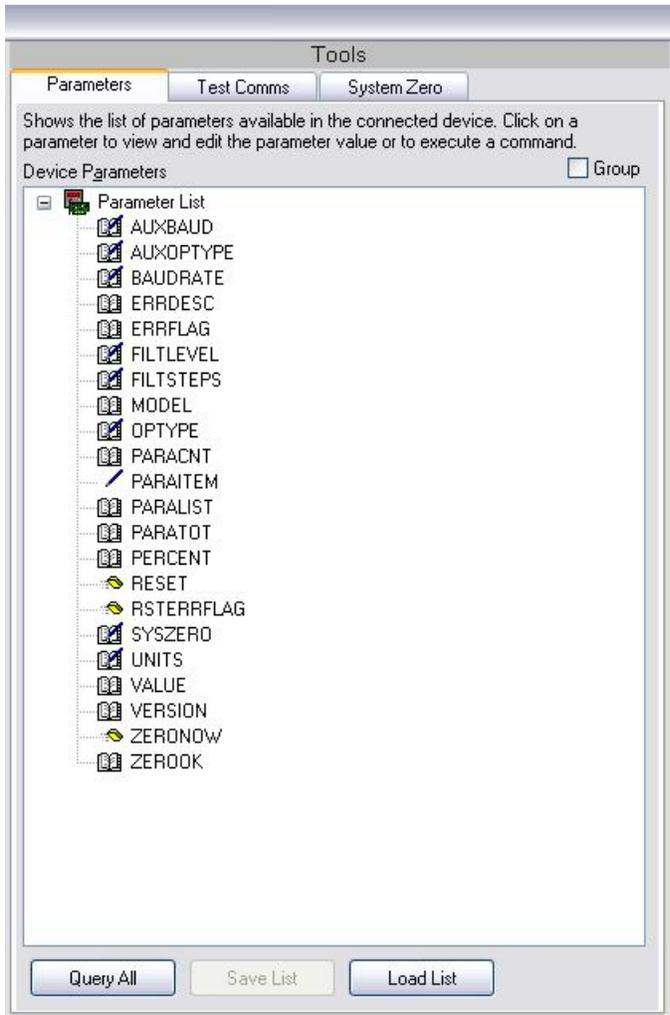
Click **OK** when changes have been made, then click the **CONNECT** button.

The TMS Toolkit searches the designated serial port to find the connected device. The software takes care of configuring the port to suit the requirements of the TMS 9000.

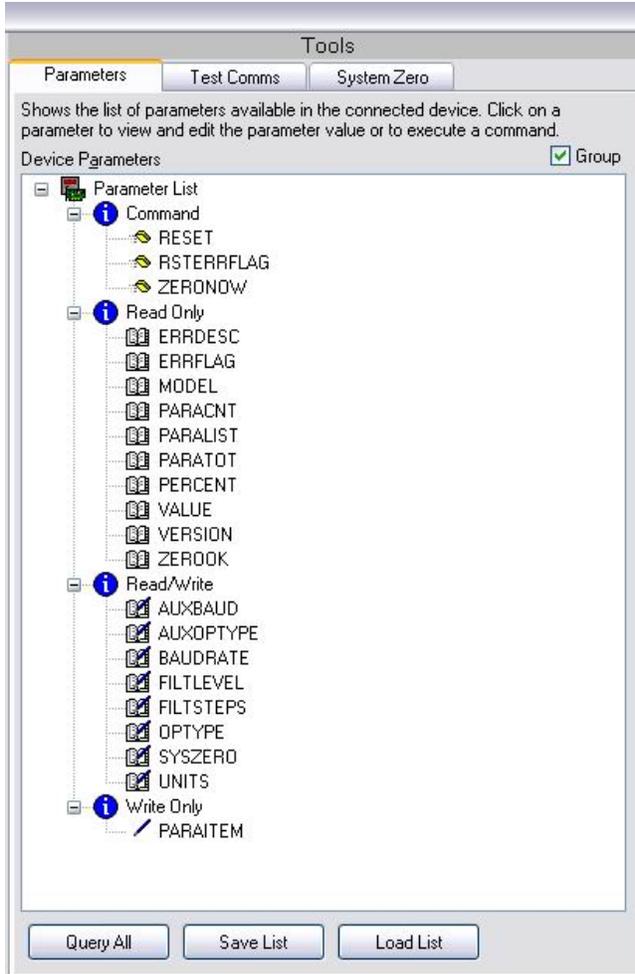
## Parameters tab

This task tab shows a list of commands and parameters that the connected device supports.

When the **TMS Toolkit** first connects to a device it does not recognize it will ask if you want it to query the device for a list of parameters. Clicking **OK** to this will cause the device to be queried. Once a successful query has taken place the toolkit will display the parameter list automatically the next time a connection is made to the same device. Note that the display is not refreshed dynamically so it is necessary to either click **QUERY ALL** periodically or double click on the parameter of interest to refresh it with the current value.



The **Group** checkbox at the top right will sort the parameters and commands into groups of similar types when checked or just alphabetically when unchecked.



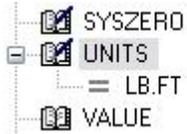
Double-clicking on a command in the tree list will cause a message dialog to appear requesting confirmation to whether the command should be sent to the device.

Double-clicking on a parameter will query the parameter value from the device then display the value in a new branch in the tree under the parameter name.

Every time the parameter name is double-clicked, the value will be re-queried and displayed. If the parameter is writeable, it can be changed by double-clicking on it.

The new value can then be input. Pressing Enter on the keyboard or clicking elsewhere on the tree list will then cause the new value to be written to the device then re-queried and displayed in the tree branch.

In all the above directions, the double click can be replaced by pressing **Enter** on the keyboard. In other words, cursor keys can be used to move the highlight around the tree list then press **Enter** to activate the branch.



## Button Functions

### Query All



**Query All** - Clicking this button will cause the available parameters to be queried from the device and displayed in the tree list. This will also enable the **Save List** button.

### Save List



**Save List** - Clicking this button, which is only available after a **Query All**, will prompt for a filename to which the parameter data will be saved.

The parameter values are saved to a text file (with the .ttp file extension) and can be used as a record of settings used within a device and also used to re-configure the same device at a later date or another device.

Note that calibration and scaling data will not be saved unless the TMS Toolkit is being operated in "Cal" user mode.

### Load List



**Load List** - Clicking this button will prompt for a filename (with the .ttp file extension) which will be used to load previously saved or edited data back into the system.

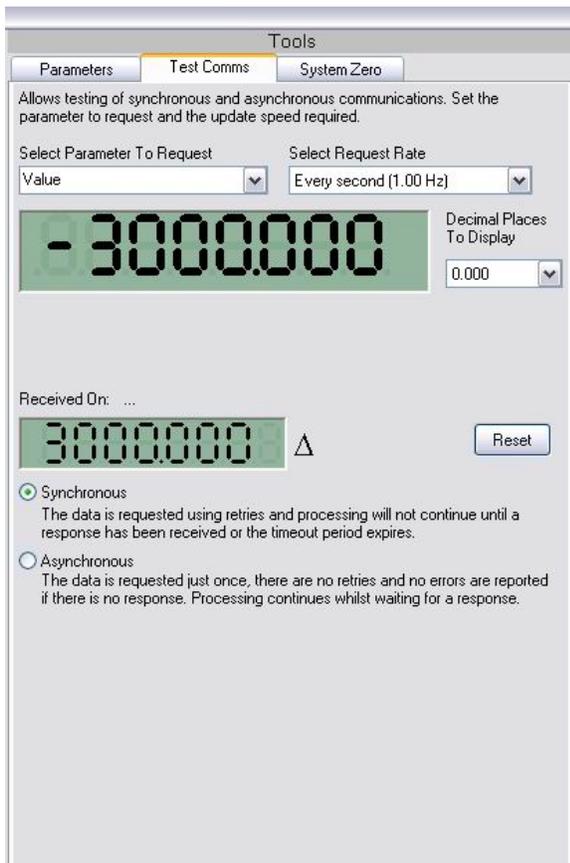
Note that calibration and scaling data will not be loaded unless the TMS Toolkit is being operated in "Cal" user mode.

Note that the parameter files are actually text files which can be edited, using a text editor such as NOTEPAD, to remove any parameters that are not required to be written to a device and also to modify parameter values. When using NOTEPAD, be sure to save the file with a .ttp file extension otherwise TMS Toolkit will not recognize it as being a valid parameter file.

## Test Comms tab

This task tab allows simple repetitive communications to be established. You can select a readable parameter from the **Select Parameter To Request** dropdown list and an update rate from the **Select Request Rate** dropdown list.

Note that you can also type directly into the **Select Parameter To Request** dropdown list and the color of the list will change to red indicating that the parameter to be requested is currently being changed. The communications will continue to request the previous parameter until either the Return/Enter key is pressed on the keyboard or focus moves to another control.



### Synchronous

The data is requested using retries (handled by the TMS) and processing will not continue until a response has been received or the timeout period expires. This option may not be able to request data at the rate selected above, because this depends on the speed of the PC and the quality of the transmission line.

The device response will be displayed in the simulated Liquid Crystal Display (LCD). This is only useful for numeric parameters (some characters such as M cannot be displayed in the simulated LCD so they will appear as blanks)

The required number of decimal places to display can be selected from the dropdown list next to the LCD.

Below the main LCD there is a smaller LCD (the **DELTA** window) which displays the change in the main reading LCD since the last reset. This display can be reset as required by clicking the **Reset** button.

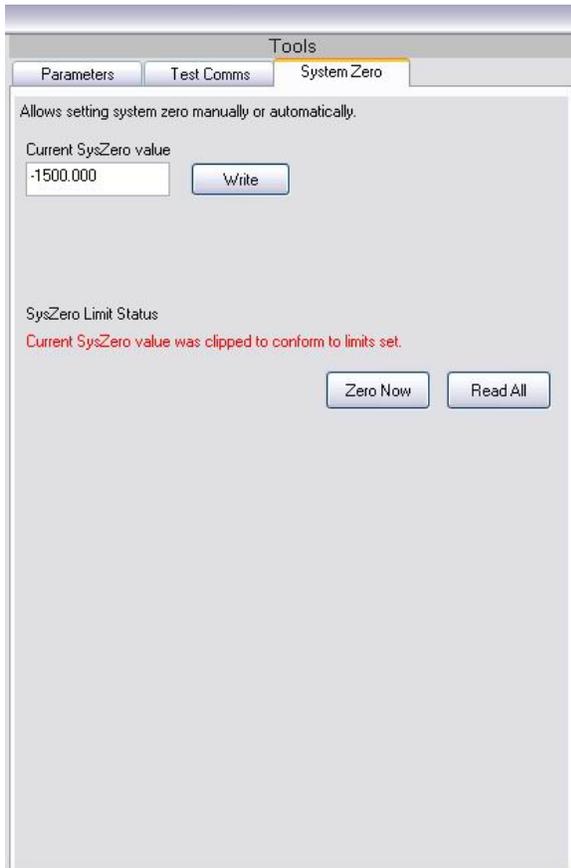
You can choose whether to send synchronous or asynchronous requests by clicking the appropriate option on the lower part of the task window.

### Asynchronous

The data is requested just once. There are no retries and no errors are reported if there is no response. If and when the response packet arrives, the packed ID (PID) is inspected and if it matches the PID in the request, the data will be displayed in the LCD. Processing continues while waiting for the data. In this mode, the 'Received On' label will indicate the date and time that the packet arrived. This option will usually appear to run faster than the synchronous option because it is not necessary to wait for a response before sending the next command. However, the data collected may contain gaps.

## System Zero tab

This task tab allows the system zero to be viewed and set.



**ZeroNow** - Click this button to perform automatic system zero.

Note that the action of the **ZeroNow** command is restricted by the hidden parameter called **ZeroLimit** as a means of preventing the repetitive use of the **ZeroNow** command in situations where physical damage is occurring within the torque sensor (for example overloading or shock loading). The value of the hidden parameter **ZeroLimit** is determined by the Calibration User during the calibration process.

In the event that a **ZeroNow** command is limited by **ZeroLimit**, then a warning message “**Current SysZero value was clipped to conform to limits set**” is displayed in red and the zero is adjusted as close to zero as is allowed by **ZeroLimit**.

**Current SysZero Value** - Shows the current value stored for the **SysZero** parameter, which is the difference between the current zero and the zero that was determined and stored during the calibration process. It is automatically updated by clicking on the **ZeroNow** button or it can be edited and written to the device by clicking the **Write** button next to the field.

Note that any attempt to set a **SysZero** value which is beyond the limits set by **ZeroLimit** will result in the zero being adjusted as close to zero as is allowed by **ZeroLimit** and the display of a red warning message “**Current SysZero value was clipped to conform to limits set**”

**SysZero Limit Status** - Indicates whether the last attempt at writing **SysZero** (or clicking **ZeroNow**) was successful within preset limits.

**Read All** button - Reads back current device settings and refreshes the display.

## CALIBRATION USER mode

To access the higher-level functions such as calibration and scaling settings, it is necessary to start the TMS Toolkit in “Cal” mode. To do this, follow the instructions below.

### Create a new shortcut



Create a copy of the TMS Toolkit shortcut on the desktop by normal means (such as ctrl-click-drag-drop or by using the right mouse button and choosing Copy then Paste).

Right-click on the new shortcut and rename it to represent the “Calibration” user mode.

Right-click on the new shortcut and select **PROPERTIES**.

### Edit the new shortcut



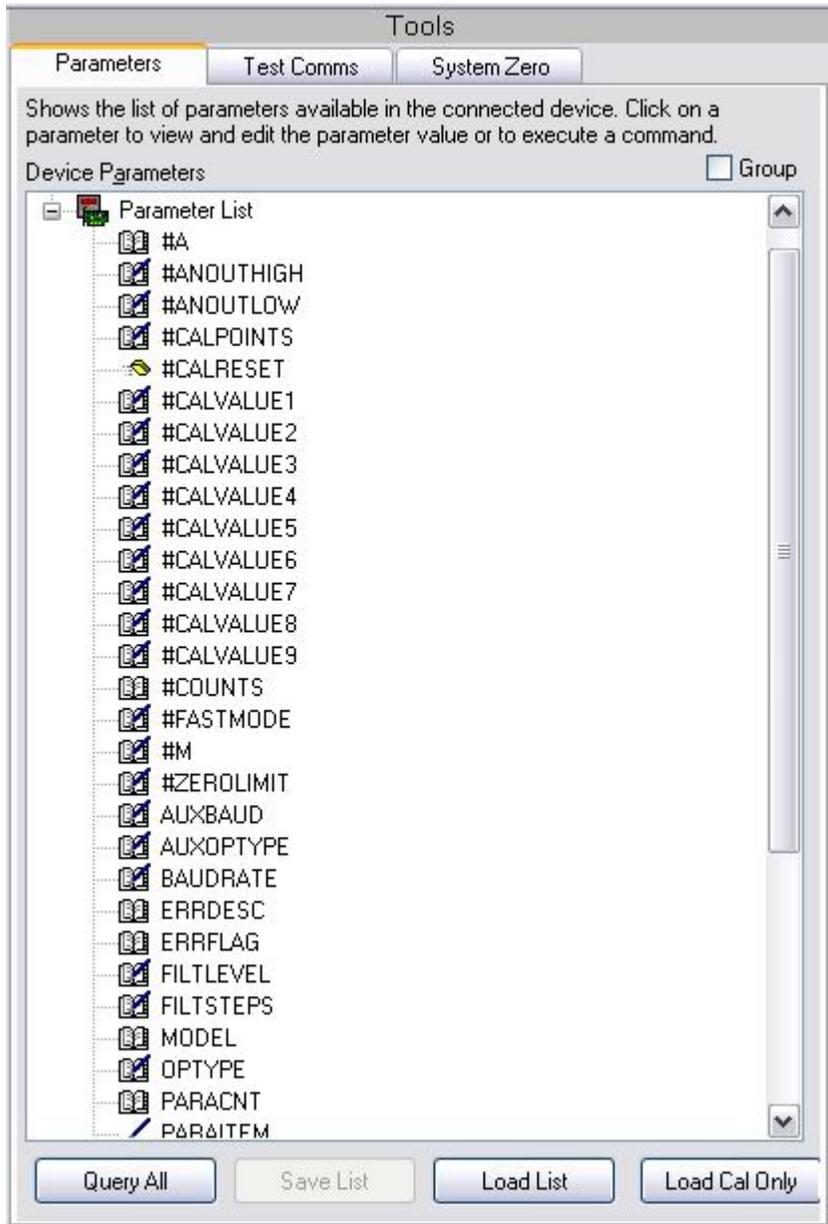
Edit the **TARGET** line by adding **/cal** after the quote marks (note – a space is required between the quote mark and the /cal text).

Click OK then test that the modified shortcut calls up the **CALIBRATION** mode of the TMS Toolkit. The header line which previously showed TMS Toolkit should now show **TMS Toolkit - CALIBRATION**

Note - If the header line continues to show TMS Toolkit then close TMS Toolkit and return to the shortcut properties dialogue box. Check that the space exists between the quote marks (") and the forward slant character (/).

## CAL user - Parameters tab

This task tab now shows a complete list of commands and parameters that the connected device supports. The parameters that are “reserved” for “Cal” user access are prefixed with the # character. Security Note - the # commands and parameters can be entered via the Immediate Window regardless of the mode of use of TMS Toolkit.



The “Cal” user mode exposes the calibration and scaling parameters which can then be accessed and edited as required.

The scaling parameters **#ANOUTHIGH** and **#ANOUTLOW** determine the scaling of the standard outputs (analog voltage, current or frequency), and operate independently of the calibration settings. Therefore, a sensor can be re-scaled at any time to represent a different full scale value.

The calibration parameters consist of the 9 calibration data values **#CALVALUE1** through **9** and the parameter **#CALPOINTS** that sets the number of calibration points that are active.

The **#CALRESET** command performs a purge of the calibration memories and should be used whenever the number of **#CALPOINTS** is changed.

**CAUTION NOTE** – all calibration and scaling data including the settings of the analog outputs will be erased whenever **#CALRESET** is performed.

## CAL user - Button Functions

### Cal user - Query All



**Query All** - Clicking this button will cause **all** parameters to be queried from the device and displayed in the tree list. This will also enable the **Save List** button.

### Cal user - Save List



**Save List** - Clicking this button, which is only available after a **Query All**, will prompt for a filename to which the parameter data will be saved. Note that **all** data including calibration and scaling data will be saved when in "Cal" user mode.

The parameter values are saved to a text file (ttp file extension) and can be used as a record of settings used within a device and also used to re-configure the same device at a later date or another device.

### Cal user - Load List



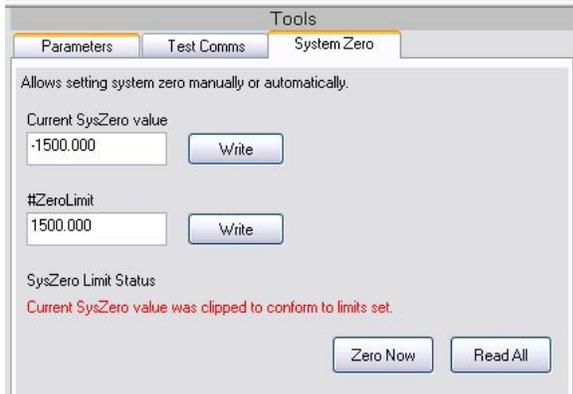
**Load List** - Clicking this button will prompt for a filename (ttp file extension) which will be used to load previously saved or edited data back into the system.

Note that a warning message will be displayed advising the user that the calibration data is about to be over-written.

Note that the parameter files are actually text files which can be edited, using a text editor such as NOTEPAD, to remove any parameters that are not required to be written to a device and also to modify parameter values. When using NOTEPAD, be sure to save the file with a .ttp file extension otherwise TMS Toolkit will not recognize it as being a valid parameter file.

## System Zero tab

This task tab allows the system zero to be viewed and set.



**ZeroNow** - Click this button to perform an automatic system zero.

Note that the action of the **ZeroNow** command is limited by the parameter **#ZeroLimit**, as a means of preventing the repetitive use of the **ZeroNow** command in situations where physical damage is occurring within the torque sensor (for example overloading or shock loading). The parameter **#ZeroLimit** is hidden from the “normal” user.

**Read All Button** - Reads back current device settings and refreshes the display.

**Current SysZero Value** - Shows the current value stored for the **SysZero** parameter, which is the difference between the current zero and the zero that was determined and stored during the calibration process. It is automatically updated by clicking on the **ZeroNow** button or it can be edited and written to the device by clicking the **Write** button next to the field.

**SysZero Limit Status** - Indicates whether the last attempt at writing **SysZero** was successful within preset limits.

In the event that a **Zero Now** or a **SysZero** value is limited by **#ZeroLimit**, then a warning message “**Current SysZero value was clipped to conform to limits set**” is displayed in red and the zero is adjusted as close to zero as is allowed by **#ZeroLimit**.

**#ZeroLimit** - Shows the limits (+ and -) within which the current value is allowed to be set to zero. This limit can be edited and written to the device by clicking the **Write** button next to the numeric field. The limit is set in engineering units (the same as the calibration units) and is used as a way of preventing repetitive use of the **ZeroNow** command in cases where physical damage may be occurring within the torque sensor (for example overloading or shock loading).

The factory setting for this parameter is 50% of the value used for the analog output (**#Anouthigh**) of the sensor and there is no test for polarity so the limit will apply either side of the “true” zero that was automatically determined by the system during the calibration process. The user may wish to set tighter limits after installing and exercising the sensor.

Note that **#ZeroLimit** is only displayed when operating TMS Toolkit at the “**Cal**” user access level.

## Troubleshooting

Problem	Possible Solutions
Cannot communicate.	<p>Check all wiring. If using the RS232 port, check that the Rx pin of the host computer is connected to the Tx pin of the TMS 9000 and vice versa. Check that the communications cable being used is of high quality or try a shorter length of cable (RS232 is sensitive to cable length and grounding issues, especially when used with laptop computers where grounding is uncertain).</p>
	<p>Check that the correct serial port is selected in the software or TMS Toolkit. When using Windows, the serial port in use can be found by using the CONTROL PANEL, SYSTEM, HARDWARE, DEVICE MANAGER, COM ports functions. On older desktop PC's, the COM1 port is already in use for the mouse, so a different COM port should be selected. If using a USB to Serial adapter, Windows assigns the COM port designations dynamically so they may change whenever the system is rebooted. The serial port settings are automatically modified by TMS Toolkit so there is no need to change any of the settings in Windows. The baud rate setting in TMS Toolkit should always be 38400 because that is the default baud rate of the TMS 9000. The "TMS ID" should be left blank because TMS Toolkit will search the connected port for any TMS device and will commence the communication automatically if present.</p>

## Error Messages – Cannot connect to Local TMS!

This message appears whenever TMS Toolkit attempts to connect to a TMS device and fails to receive a response. Possible reasons are as follows :



- No TMS device is connected
- A TMS device is connected, but is powered off
- A bad or broken connection exists at the PC end or at the TMS end
- Connected to the wrong communications port at the PC end (should be the same as was selected on the **Settings** tab) or at the TMS end (should be J7)
- If trying to connect for the first time, the connections may be incorrect (RS232 requires a crossover between devices, so Rx should connect to Tx and vice versa)

Check all of the above and try again.

## Error Messages – An error occurred while attempting to identify device!



This message appears whenever TMS Toolkit attempts to connect to a TMS device and receives an unexpected or invalid response. Possible reasons are as follows :

- The connected device is not a TMS
- The connection between the PC and the TMS is too long for reliable RS232 communications

Check all of the above and try again.

## Error Messages – Broadcast ID Information



This message appears whenever TMS Toolkit receives an invalid response. Possible reasons are as follows :

- No TMS device
- Faulty communications cable
- Sending a broadcast command (see multi-dropped TMS)

Check all of the above and try again.

## Appendix A – Commands and Parameter Descriptions

**Introduction** - The TMS 9000 system uses a simple ASCII-based communications protocol that is both easy to use and easy to memorize. Some repetitive tasks such as setting a new zero point can be performed with less key strokes by making use of the Immediate Window and keying the command directly.

The complete list of valid parameters and commands and their functions are included below. Note that any parameters or commands that are prefixed with # will not be displayed in the parameter list when in “normal” user mode. However, all commands and settings can be accessed via the Immediate Window in either the “normal” user mode or the “Cal” user mode.

If communicating with the TMS 9000 via HyperTerminal or some other communications channel, then the full ASCII-XP message protocol must be provided, as detailed in Appendix B.

Below is the list of commands and parameters supported by the TMS 9000 (corresponding to SPM firmware version v1.36 and above) :

Parameter Name	Description	Data Type
#A	Mode reporting access used by TMS Toolkit software to extract parameter information from the device.	Read only
#AnOutHigh	Sets or returns the value in engineering units applied to the input that will give 100% (maximum positive full scale) output on the analog outputs. To invert the output polarity, enter the required negative full scale output value.  <b>NOTE</b> – any change to #AnOutHigh will result in the cancellation of any zero offset value stored in #SysZero	Read/write
#AnOutLow	Sets or returns the value in engineering units applied to the input that will give 0% (minimum negative full scale) output on the analog outputs. To invert the output polarity, enter the required positive full scale output value.  <b>NOTE</b> – any change to #AnOutLow will result in the cancellation of any zero offset value stored in #SysZero	Read/write
AuxBaud	Not yet supported.	Read/write
AuxOPType	Not yet supported.	Read/write
BaudRate	Not yet supported.	Read only
*CalCnts1- *CalCnts9	Internal calibration data. This is read only via TMS Toolkit and is viewable only in “CAL” level access. These parameters are exposed to enable the saving and loading of calibration data only.	Read only
#CalPoints	Sets the number of calibration points in use. Value must be between 2 and 9. Any change in #CalPoints should be followed by a #CalReset command to clear the previous unwanted calibration data from the memories. For correct system operation, the number of active calibration values (#CalValue 1-9) must be equal to the number of calibration points that are defined by the #CalPoints parameter.  <b>NOTE</b> – all calibration data and all analog output setting data will be cleared by #CalReset. It is recommended to save the parameter list before invoking #CalReset.  <b>NOTE</b> – any change to #AnOutHigh will result in the cancellation of any zero offset value stored in #SysZero	Read/write
#CalReset	Resets all calibration information. When the reset command is issued, all calibration data and all settings of the analog outputs (#AnOutHigh and #AnOutLow) are cleared so no reliable output will be available until all of the calibration points specified by the #CalPoints parameter and the required values of #AnOutHigh and #AnOutLow have been entered.  <b>NOTE</b> – all calibration data and all analog output setting data will be cleared by #CalReset. It is recommended to save the parameter list before invoking #CalReset.  <b>NOTE</b> – any change to any calibration parameter such as caused by #CalReset will result in the cancellation of any zero offset value stored in #SysZero	Command

<b>#CalValue1</b> to <b>#CalValue9</b>	<p>These values are written in engineering units when the appropriate load is applied. Each of the nine parameters can be written at any time. See Calibration section later in this document. <b>NOTE</b> – the values entered <b>MUST</b> be in ascending order, starting with <b>#CalValue1</b> (negative values count as lower than positive values). The number of calibration points entered must be equal to the number of calibration points activated by <b>#CalPoints</b>.</p> <p><b>NOTE</b> – the existing calibration data is overwritten by any new input of <b>#CalValue1~9</b>. It is recommended to save the parameter list before entering new values.</p> <p><b>NOTE</b> – any change to <b>#CalValue1~9</b> will result in the cancellation of any zero offset value stored in <b>#SysZero</b></p>	Read/write										
<b>#Counts</b>	Returns the raw A-D counts value derived from the ADC on the rotating sensor.	Read only										
<b>ErrFlag</b>	<p>The <b>ErrFlag</b> parameter will indicate any errors that have occurred by returning a numeric value that is comprised of binary values representing the various error states. i.e. the binary values for each error are added together to produce the <b>ErrFlag</b> value.</p> <p>The error states are not retained between power cycles.</p> <table border="1" data-bbox="365 636 735 762"> <thead> <tr> <th>Decimal Value</th> <th>Error Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No error.</td> </tr> <tr> <td>1</td> <td>Power cycled.</td> </tr> <tr> <td>2</td> <td>Output clamped.</td> </tr> <tr> <td>4</td> <td>Watchdog reset.</td> </tr> </tbody> </table> <p>Once the errors have been read they can be reset using the <b>RstErrFlag</b> command.</p>	Decimal Value	Error Description	0	No error.	1	Power cycled.	2	Output clamped.	4	Watchdog reset.	Read only
Decimal Value	Error Description											
0	No error.											
1	Power cycled.											
2	Output clamped.											
4	Watchdog reset.											
<b>#FastMode</b>	<p>Used to initiate raw throughput of data without scaling or filtering. Set this parameter to 1 to enable fast mode. This setting is volatile so the device will revert to normal mode after the next Reset command or after the next power-up. Set to zero to disable fast mode. When in fast mode, the internal raw A-D counts results are fed directly to the analog output (voltage or current) without any scaling or filtering, giving a data throughput rate of 8.8 kHz (regardless of <b>FiltLevel</b> or <b>FiltSteps</b> settings).</p> <p><b>NOTE</b> - the frequency output is not supported in fast mode. The fast mode can be scaled in the user's data acquisition system by using the shunt cal facility, and is intended to be used for dynamic measurements only.</p>	Read/write										
<b>FiltLevel</b>	<p>Used to set the threshold of operation of the digital filter. Values are set as parts per 10000, meaning that to set a threshold of 10% of the sensor rated capacity, then <b>FiltLevel</b>=1000</p> <p>For a % step change in input which is greater than (<b>FiltLevel</b> / 10000 * 100%), the new input value will be passed immediately to the output. For a step change in input which is below the threshold set by <b>FiltLevel</b>, the output is filtered according to the setting of <b>FiltSteps</b>. Permissible values are 1 through 10000.</p> <p><b>NOTE</b> – when <b>FiltLevel</b> is set to 1, digital filtering is disabled and the unfiltered data is passed to the analog outputs at a rate of 2.2 kHz. When <b>FiltLevel</b> is set to 2, the digital filter update rate is set to be 1.1 kHz and the filter characteristic is set by <b>FiltSteps</b>. Factory default value is 100, representing a threshold of 1% of sensor rated capacity (refer to the factory calibration data sheet or the rating plate attached to the sensor to confirm the rated capacity).</p>	Read/write										
<b>FiltSteps</b>	<p>Used to set the response time of the digital filter. Used in conjunction with <b>FiltLevel</b> to control the digital filtering behaviour. Permissible values are 1 through 10000.</p> <p>Filtering takes the form of an RC equivalent, where a change in input value which is greater than the threshold set by <b>FiltLevel</b> causes the output value to be incremented in the number of steps set by <b>FiltSteps</b>. The filter refresh rate is 1100 Hz.</p> <p>Factory default setting is 10, which in conjunction with the factory default setting of <b>FiltLevel</b>=100 provides for an output increment in the form of (x/2, x/3, x/4, x/5, x/6 .... x/10) where x=step change in input of more than ((<b>FiltLevel</b>/10000)*rated capacity). Given the filter update rate of 1100 Hz, the settling time to 63% final value will be 9 ms (Update rate x <b>FiltSteps</b>) and to 1% final value will be 45 ms (Update rate x <b>FiltSteps</b> x 5). The settling time to 0.1% final value will be 64 ms (Update rate x <b>FiltSteps</b> x 7).</p>	Read/write										
<b>#M</b>	Mode reporting access used by TMS Toolkit software to extract parameter information from the device.	Read only										
<b>Model</b>	Returns the model name (TMS)	Read only string										
<b>OpType</b>	<p>Sets or returns the currently selected analog output where 0=current; 1=voltage; 2=frequency 10 kHz; 3=frequency 60 kHz, 4=current and frequency 10 kHz, 5=voltage and frequency 10 kHz, 6=current and frequency 60 kHz, 7=voltage and frequency 60 kHz..</p> <p><b>NOTE</b> – data throughput performance can be affected by using multiple outputs. For high speed dynamic data analysis applications (&gt;500Hz), select the voltage output (<b>OpType</b>=1).</p>	Read/write										
<b>ParaCnt</b>	Returns the number of parameters in the device.	Read										

<b>Paraltem</b>	Set to the index number of the required parameter.	Write												
<b>ParaList</b>	<p>Returns the information on the parameter indexed by <b>Paraltem</b>. Format: '<b>index,paramname,type</b>' a string.</p> <p>The value of <i>type</i> indicates the parameter's properties by the addition of the following numerical values:</p> <table border="0"> <tr><td>1</td><td>Readable</td></tr> <tr><td>2</td><td>Writeable</td></tr> <tr><td>4</td><td>Command</td></tr> <tr><td>32</td><td>String</td></tr> <tr><td>64</td><td>Numeric</td></tr> <tr><td>128</td><td>Boolean</td></tr> </table> <p>Example: '<b>1,MODEL,33</b>' In the above example where <i>type</i> = 33 the parameter <i>MODEL</i> is a readable string.</p>	1	Readable	2	Writeable	4	Command	32	String	64	Numeric	128	Boolean	Read string
1	Readable													
2	Writeable													
4	Command													
32	String													
64	Numeric													
128	Boolean													
<b>Percent</b>	Returns the value of the applied torque in percentage terms (0-100) where this range is the selected range over which the analog outputs work and is set by <b>#AnOutLow</b> and <b>#AnOutHigh</b> .	Read only												
<b>#ReScale</b>	Internal system command, for factory use only	Command												
<b>Reset</b>	<b>Reset</b> command to restart device and to implement parameter changes that require a reset.	Command												
<b>RstErrFlag</b>	Reset all error flags.	Command												
<b>#ScScale</b>	Scaling factor applied to the Shunt Calibration value. Can be used to force the shunt calibration value to any desired value. A setting of 1 causes the shunt calibration value to be delivered at its raw level, which could be any value (refer to calibration data sheet for the relevant sensor for the actual value), but is generally set during manufacture to be between 50% and 99% of the sensor capacity.	Read/Write												
<b>SysZero</b>	Allows manual setting of the current <b>Value</b> or querying of the current zero offset being applied. The returned value is the amount of zero offset being applied to the true <b>Value</b> . To zero the system, this parameter should be set in engineering units to the value read when the system is supposed to be displaying zero. The action of <b>SysZero</b> may be limited by <b>#ZeroLimit</b> as described above, in which case the flag <b>ZeroOK</b> will be set to 0. Note that when any calibration parameter ( <b>#CalValue1-9</b> , <b>#CalReset</b> ) is changed, the value of <b>SysZero</b> is set to 0 and any zero offset is cancelled. This function, when used with <b>#ZeroLimit</b> , allows the current <b>Value</b> to be offset to any desired level (remember to consider the dynamic loading range of the sensor itself when applying large offsets)	Read/write												
<b>Units</b>	Text memo field in which the name of the engineering units used for calibration can be stored for recall later. Note that when reading some characters via a 7-segment display (TMS Toolkit uses a virtual 7-segment display), some characters such as M will not display correctly.	Read/Write												
<b>Usr1-9</b>	Free format, text memo fields for general purpose. Example is the storage of calibration data and place information, the name of the calibration technician, etc. Allowable characters are alphanumeric only.	Read/Write												
<b>Value</b>	Returns the value of the applied torque in calibrated engineering units.	Read only												
<b>Version</b>	Returns the software version	Read only												
<b>ZeroNow</b>	Sets the current <b>Value</b> to zero unless limited by <b>#ZeroLimit</b> as described above. The action performed by <b>ZeroNow</b> is to clear any previous zero offset then compare the true <b>Value</b> to <b>#ZeroLimit</b> , then, to the extent allowable by <b>#ZeroLimit</b> , write the true <b>Value</b> to <b>SysZero</b> , resulting in a new current <b>Value</b> of zero	Command												
<b>ZeroOK</b>	Returns indication of 1 if the previous <b>ZeroNow</b> command was successful in setting the current <b>Value</b> to zero and returns 0 if the action was limited by <b>#ZeroLimit</b> .	Read only												
<b>#ZeroLimit</b>	The limit in engineering units at which the <b>ZeroNow</b> command will be allowed to operate, relative to the computed <b>Value</b> at zero load that was stored during the calibration process. Therefore <b>#ZeroLimit</b> represents the maximum allowable difference between the "calibration" zero and the "current" zero. If the <b>ZeroNow</b> command is issued when the current <b>Value</b> is greater than <b>#ZeroLimit</b> , then the current <b>Value</b> will be moved to the extent allowed by <b>#ZeroLimit</b> and the flag <b>ZeroOK</b> will be set to 0. Factory setting is 50% of the calibrated range. Note that <b>#ZeroLimit</b> is a bipolar setting, so it will be applied to both directions (+ and -) around the calibration zero value.	Read/Write												
<b>*ZeroPVal</b>	Internal system zero data. This is read only via TMS Toolkit and is viewable only in "CAL" level access. This parameter is exposed to enable the saving and loading of calibration and zero data.	Read only												

## Appendix B – ASCII XP Protocol

**Introduction** - Communications with the TMS 9000 is carried out using a protocol, **ASCIIXP (ASCII eXtended Protocol)**, which has been developed for multi-drop telemetry products.

Each TMS 9000 device has a unique device ID that must be used for all communications. The ID is fixed at production and is referred to in hexadecimal format. The range covered is from **000001** to **FFFFFF** which gives an address range of **1** to **16777215** decimal. With 16 million addresses, no two devices will have the same ID.

Device ID **000000** is reserved as a '**Broadcast**' ID. Data transmitted to this ID will be acted on by all receiving devices. However, devices will **not** respond with data to a broadcast command.

All devices communicate using the **ASCIIXP** protocol which has been designed to be humanly readable. Another advantage of this protocol is that with its optional parameter structure, it can just as easily be used from a simple serial terminal program such as HyperTerminal (typed manually in real-time) as well as a high level device such as a PC or PLC. A driver is available for use with any programming system that can utilise OCX's. Source code, written in Visual Basic, is also available.

### ASCIIXP Protocol

ASCIIXP is an ASCII protocol designed to be routable over radio, Ethernet or serial links and is easily programmed and parsed (split into sections). Only ASCII characters are used and the suffix 'Carriage Return' (ASCII value 13, hexadecimal value 0xD) is used as the framing character. This enables ASCIIXP to be used with common serial terminal applications such as HyperTerminal. The 'Enter/Return key' on a standard keyboard is used to write the framing character. In the following document **<13>** will be used to represent the ASCII character 13.

---

A checksum can be included for integrity but it is optional. If a checksum exists in a received data packet then a checksum will be included in the returned packet.

---

The full syntax for the ASCIIIXP protocol is:

**ToID**[:**FromID**[:**PID**]]:**Data**[:**Checksum**]<13>

Or, in its simplest form :

**ToID**:**Data**<13>

Note that the packet is divided into fields by the colons. Square bracketed sections indicate optional parameters. The first field is the address, the second is the data and the third (optional) field is the checksum.

All packets must have a **ToID**, i.e. the ID of the intended target device, but can also have an optional **FromID**, (the ID of the device that transmits the packet) and a **PID** packet identifier (see below). It is not necessary to include the **FromID** in the transmitted packet unless multiple TMS 9000 devices reside on the same serial port. In this case the **FromID** is used to indicate which TMS 9000 should handle the data. In single TMS 9000 systems, the TMS 9000 device can extract this data from the underlying transport protocol since all devices include their **FromID** in all their transmitted data resulting in unambiguous identification of the sender.

The **PID** is also optional. This is the **P**acket **I**Dentifier and can be up to 6 alphanumeric characters. If a **PID** is used, then a **FromID** should be used or at least a placeholder for a **FromID** i.e. **FFFF;;abcd:xxxxxxxxxxx**. The function of the **PID** is to enable a received packet to be matched with the packet that requested the data. When programming a PC to talk to a device a **PID** should be used. This enables the user to determine that the device has responded as a result of a specific request by checking that the **PID** in the response matches the **PID** of the request. This inclusion of a **PID** is even more important in an environment where a device is periodically pushing data out to a PC due to one or more of its **Events** being triggered. In this case the data arriving as a result of an **Event** occurring would have a **PID** of **Ev<sub>n</sub>** where **n** is the number of the event that sent the data. Thus, by including a **PID** in a request for data, the **PID** of incoming data packets can be checked to determine if the data is a response to this request or alternatively, an unrequested transmission that has been triggered by a remote device's **Event**.

Spaces should **not** be used in the ASCIIIXP protocol.

When devices communicate with each other they **always** include a **FromID** and a **PacketID**.

Checksums are optional. In order to **not** use a checksum, the colon preceding the checksum and the checksum itself must be omitted i.e. **Address:command<13>**  
If no checksum is present in a request, the reply will be sent without a checksum.

Examples of valid packet constructions:

**ToID**;**FromID**;**PID**:**Data**:**Checksum**<13>  
**ToID**;;**PID**:**Data**:**Checksum**<13>  
**ToID**:**Data**:**Checksum**<13>  
**ToID**;**FromID**;**PID**:**Data**<13>  
**ToID**;;**PID**:**Data**<13>  
**ToID**:**Data**<13>

## Definitions

ToID	A 6 character maximum address, which is a HEX representation of the device address. All telemetry devices will be manufactured with a unique 3 byte address. Example: FEDCAB.
FromID	<i>Optional.</i> A 6 character hex address, which is the sender's device address. This is only required when multiple TMS 9000 devices are connected to the same serial bus.
PID	<i>Optional.</i> Up to six characters of numbers or letters to identify the packet. If the PID is prefixed with a '!' character the packet will be sent asynchronously i.e. there will be no retries on transmission and no ACK is required on any underlying transport protocols. The programmer will be responsible for catching the returned data packet and handling it. This is useful for periodically 'prompting' a device for a value without having to wait and see if it has acknowledged the prompt. When the result arrives it can be identified by inspecting its PID.
Data	Data is the information sent to the remote device and can be for requesting data, changing a parameter value or executing a command.
Checksum	<i>Optional.</i> Is a 2 character ASCII representation of the hex value of <b>ALL</b> the XOR'ed bytes in the transmitted packet from the first address byte up to and including the colon preceding the checksum digits. A typical checksum may be C7
<13>	The single byte, ASCII value 13 always suffixes a data packet.

---

NOTE: The TMS 9000 uses the ASCIIXP protocol, which can be utilised by radio devices and contains routing data, which consists of the sender ID and the receiver ID. The TMS 9000 does not require a gateway device to enable connection to a serial port so the ToID and the FromID will be identical!

---

## Communications Examples

For the next section of this document we will assume that no checksum is used to simplify explanation. Each device supports a range of parameters that are either readable, writeable, read/writeable or commands. The next simple examples will assume that the transmitted data is sent to AAAAAA

### To read a parameter synchronously:

**AAAAAA:Value?<13>**

**Value** is the parameter name followed by a question mark.

The response would be

**AAAAAA;AAAAAA:123.456<13>**

If the parameter returns a string value it may look like this... Note the single quote delimiters around the string data.

**AAAAAA;AAAAAA:'TMS 9000'<13>**

If the device does not recognize the command then it will return

**AAAAAA;AAAAAA:?<13>**

### To read a parameter asynchronously:

**AAAAAA;!pid:Value?<13>**

**Value** is the parameter name followed by a question mark. We have now specified a **PID** and indicated that this is an asynchronous transmission by prefixing the **PID** with a **!**

An asynchronous transmission allows the driver to not wait for a response and also not apply any retries in the low level transport protocol. This feature can be used to regularly transmit a request for data where the receiving device may or may not be activated. The user is responsible for receiving the response when it occurs and to determine how to handle the data by inspecting the **PID**.

The response would be:

**AAAAAA;AAAAAA;!pid:123.456<13>**

If the parameter returns a string value it may look like this:

**AAAAAA;AAAAAA;!pid:'TMS 9000'<13>**

Note the single quote delimiters around the string data.

If the device does not recognize the command then it will return

**AAAAAA;AAAAAA;!pid:?<13>**

### To write a parameter

**AAAAAA:CalValue1=123.456<13>**

Where **CalValue1** is the parameter name followed by an **equals** sign then the **data** in engineering units. To write to a parameter that expects a string value you must enclose the string in single quotes:

**AAAAAA:StringVal='GROSS'<13>**

The device will respond with:

**AAAAAA;AAAAAA:OK<13>**

If the device does not recognize the command or the data is invalid then it will return:

**AAAAAA;AAAAAA:?<13>**

# TMS 9000 Torque Measurement System

October 2006  
008-0689-00

## To issue a command

**AAAAAA:Reset<13>**

Where **Reset** is the command name.

The device will respond with:

**AAAAAA;AAAAAA:OK<13>**

If the device does not recognize the command or the data is invalid then it will return:

**AAAAAA;AAAAAA?:<13>**

## Multiple parameters

Multiple commands or requests must be separated with semicolons i.e.

**FFFFFF:Model?;#CalValue1=99;Reset<13>**

The device will respond with all the values normally returned but separated by semicolons i.e.

**AAAAAA;AAAAAA:'TMS 9000';OK;OK<13>**

## Appendix C - Software License

**Introduction** - The TMS Toolkit software package was developed by Mantracourt Electronics Limited on behalf of Honeywell Lebow Products Inc.

The use of the TMS Toolkit software is governed by the license conditions contained in the text below. The software installation procedure repeats the agreement text as below and requires that the license conditions are accepted before the software will load.

THIS IS A LEGAL AGREEMENT BETWEEN YOU AND MANTRACOURT ELECTRONICS LTD. READ IT CAREFULLY BEFORE USING THE SOFTWARE. IT PROVIDES A LICENCE TO USE THE SOFTWARE AND CONTAINS WARRANTY INFORMATION AND LIABILITY DISCLAIMERS. BY INSTALLING AND USING THE SOFTWARE, YOU ARE CONFIRMING YOUR ACCEPTANCE OF THE SOFTWARE AND AGREEING TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO BE BOUND BY THESE TERMS, THEN UNINSTALL THE SOFTWARE.

**Definition** - The terms: product, software or control are used interchangeably and represent all executables, ActiveX controls, documentation, Help files, and any other file included in the downloadable package.

**License** - The Product is protected by intellectual property laws, copyright law and international copyright treaty provisions. You are granted a worldwide, non-exclusive, non-transferable, license to use this product on multiple computers.

**License Restrictions** - You may not de-compile, reverse engineer, disassemble, modify, translate or otherwise reduce the Software to a human-perceivable form. You may not change the product in any way that removes or changes the requirement of a license file or registration. You may not rent, lease, or make a charge for the Software. You are not allowed to distribute our software or constituent controls as constituents of other controls or to allow any third party developer to use its features in design mode. You are not permitted to distribute any Registry Keys, license files (.lic), or serials inserted by our products to any other party and to take necessary care to prevent unauthorized use of our controls by a third party in a development environment.

**Distribution** - This agreement allows you to distribute the software freely, without royalty payments, to an unlimited number of computers as long as the controls are not being made available as development controls within a development environment.

**Disclaimer of Warranty** - This software and any accompanying files are provided 'AS IS' without warranty of any kind either expressed or implied, including but not limited to any warranties of merchantability or fitness for a particular purpose. We disclaim any warranty of non-infringement to third parties rights.

**Limitation of Liability** - Your use of this software is AT YOUR OWN RISK. Under no circumstances shall we be liable to you or any other person for any direct, indirect, special, incidental, exemplary, or consequential damages of any kind, including but not limited to damages for loss of profits, data, computer failure or malfunction or any commercial damages or loss arising from your use of this software, even if we have been advised of such damages.

**⚠ WARNING**

**PERSONAL INJURY**

DO NOT USE these products as safety or emergency stop devices or in any other application where failure of the product could result in personal injury.

**Failure to comply with these instructions could result in death or serious injury.**

**⚠ WARNING**

**MISUSE OF DOCUMENTATION**

- The information presented in this product sheet is for reference only. Do not use this document as product installation guide.
- Complete installation, operation, and maintenance information is provided in the instructions supplied with each product.

**Failure to comply with these instructions could result in death or serious injury.**

**WARRANTY/REMEDY**

Honeywell warrants goods of its manufacture as being free of defective materials and faulty workmanship. Honeywell's standard product warranty applies unless agreed to otherwise by Honeywell in writing; please refer to your order acknowledgement or consult your local sales office for specific warranty details. If warranted goods are returned to Honeywell during the period of coverage, Honeywell will repair or replace, at its option, without charge those items it finds defective. **The foregoing is buyer's sole remedy and is in lieu of all other warranties, expressed or implied, including those of merchantability and fitness for a particular purpose. In no event shall Honeywell be liable for consequential, special, or indirect damages.**

While we provide application assistance personally, through our literature and the Honeywell web site, it is up to the customer to determine the suitability of the product in the application.

Specifications may change without notice. The information we supply is believed to be accurate and reliable as of this printing. However, we assume no responsibility for its use.

**SALES AND SERVICE**

Honeywell serves its customers through a worldwide network of sales offices, representatives and distributors. For application assistance, current specifications, pricing or name of the nearest Authorized Distributor, contact your local sales office or:

**E-mail:** [sales@sensotec.com](mailto:sales@sensotec.com)

**Internet:** [www.honeywell.com/sensotec](http://www.honeywell.com/sensotec)

**Phone and Fax:**

Tel: 614-850-5000

Fax: 614-850-1111

**Automation and Control Solutions**

Sensing and Control

Honeywell

Lebow Products

1985 Douglas Drive North

Minneapolis, MN 55422

[www.honeywell.com/sensing](http://www.honeywell.com/sensing)

Printed in USA  
November 2006

Copyright © 2006 Honeywell International Inc. All rights reserved.

**Honeywell**