
FT602-D-SM

Digital Wind Sensor User Manual



FT TECHNOLOGIES LTD.
SUNBURY HOUSE
BROOKLANDS CLOSE
SUNBURY-ON-THAMES
MIDDLESEX TW16 7DX



TEL: +44 (0)20 8943 0801
FAX: +44 (0)20 8943 3283
WEB: www.fttechnologies.com
E-MAIL: info@fttechnologies.com

9000937 Rev A01

August 2026

The FT and Acu-Res logos are registered trademarks of FT Technologies Ltd.

Copyright © 2024 FT Technologies Ltd. All rights reserved.

Contents

Product Symbols.....	4
Safety Instructions	5
1 Quick Start Guide.....	6
2 INTRODUCTION.....	8
2.1 Product Overview	8
2.2 Labelling	8
2.3 Scope of Use	9
2.4 Disclaimer.....	9
3 FUNCTIONAL DESCRIPTION	10
3.1 Technical Performance	10
3.2 Wind Speed Calibration	11
3.3 Wind Speed and Direction Filtering.....	11
3.4 Compass & Declination Angle Operation	12
3.4.1 Compass Convention	12
3.4.2 Compass Calibration	12
3.5 Inclinator Zeroing	12
3.5.1 Pitch and Roll Conventions	13
3.6 Error Detection.....	13
4 MECHANICAL & ELECTRICAL INSTALLATION.....	14
4.1.1 Connector Details.....	17
5 SERVICE, CONFIGURATION & TESTING	18
5.1 Inspection.....	18
5.2 Fault Finding & Troubleshooting	20
5.3 Returns	20
6 SENSOR COMMUNICATION.....	21
6.1 Introduction	21
6.2 Conventions used in this manual	21
6.3 Data Transmission.....	22
6.4 Message Format	22
6.5 Listener and Talker Identifiers	23
6.6 Calculating the Message Checksum	24
6.7 Disabling the Checksum	25
6.8 Communicating via Command Line Terminal Programs (Including TeraTerm)	26
7 Communication Over CANBus	28
7.1 Mandatory PGNs	28

7.2	Configurable PGNs	28
8	PARAMETER SETTINGS	29
8.1	Command Types	29
8.1.1	Set Commands	29
8.1.2	Query Commands	30
8.2	Timing Constraints	31
8.3	Command Parameters	32
8.3.1	AM: Set or Query Anemometer Mount Orientation	32
8.3.2	AT.1: Query the Acoustic Temperature	33
8.3.3	AT.2: Set or Query the Acoustic Temperature Units	34
8.3.4	AT.3 Set or Query the Acoustic Temperature Filter Length	35
8.3.5	BM: Set or Query the Barometric Pressure	36
8.3.6	BR: Set or Query the Serial Interface Baud Rate	37
8.3.7	CF: Set or Query the Wind Compass Settings	38
8.3.8	CF.2: Set or Query the Tilt Measurements	40
8.3.9	CI: Set or Query the Communication Interface (RS485, RS422, UART)	41
8.3.10	CIC: Set or Query the CANbus Interface	42
8.3.11	CICP: Set or Query the NMEA PGNs	43
8.3.12	CU: Set or Query the Continuous Update Setting	44
8.3.13	DF: Set or Query the Wind Velocity Data Format	45
8.3.14	DL: Set or Query the Command Delay Interval	47
8.3.15	ER: Query or Reset the Error Report	48
8.3.16	FL.1: Set or Query General Filter Settings	49
8.3.17	FL.2: Set or Query Filter Lengths	50
8.3.18	ID: Set or Query the Listener & Talker Identifiers	51
8.3.19	MM: Reset or Query the Min/Max Recorded Wind Speed	52
8.3.20	SN: Query the Serial Number and Platform Version	53
8.3.21	SV: Query the Software Version	54
8.3.22	WV Polar: Query the Wind Velocity Reading	55
8.3.23	WV NMEA: Query the Wind Velocity Reading	57

Product Symbols

The following symbols may be used upon the product and within the manual.

Meaning / Description	Symbol
Warning/ Caution An appropriate safety instruction should be followed or caution to a potential hazard exists	
DC Current only Equipment operates under Direct Current (DC) supply only.	
Product Disposal In accordance with European directive 2012/19/EU on Waste Electrical and Electronic Equipment (WEEE), these product components must be recycled. This should be done by returning the product to FT Technologies or by using an appropriate waste disposal company. This product should not be disposed of in general waste of landfill. This product complies with the RoHS2 (2011/65/EU) directive.	

Safety Instructions

English

- To ensure the safe installation and operation of this product the equipment must be installed and integrated:
 - Using suitably qualified and trained personnel
 - In accordance with any regional electrical codes
 - In accordance with the instructions set out in this manual, observing all information, warnings and instructions
 - In accordance with any other instructions or guidance FT Technologies provide
- The equipment must only be operated within the range of the specified technical data and used for the purposes for which it was designed.
- The equipment should always be transported in packaging which is appropriate, that will prevent any accidental damage from occurring.
- Always ensure that any failures or errors from the product cannot cause any damage to any other equipment or property or cause any other consequential effects.

1 Quick Start Guide

Alongside your FT602 sensor you will find an FT063 Acu-Test cable (Figure 1) with a connector to the sensor at one end and USB connector at the other end. This cable can be used alongside FT Technologies' Acu-Vis software to temporarily communicate with your sensor and make any configuration changes you require. Users are given access to download Acu-Vis when they purchase a sensor, if you are having issues accessing Acu-Vis contact techsupport@fttechnologies.com for assistance.

Acu-Vis can be installed on any computer running Windows 7, 8, 8.1, 10 or 11. The sensor should be connected to the Acu-Test cable then connected to the PC via a USB port. Once connected, open Acu-Vis and select the green 'Connect' button at the bottom of the screen.

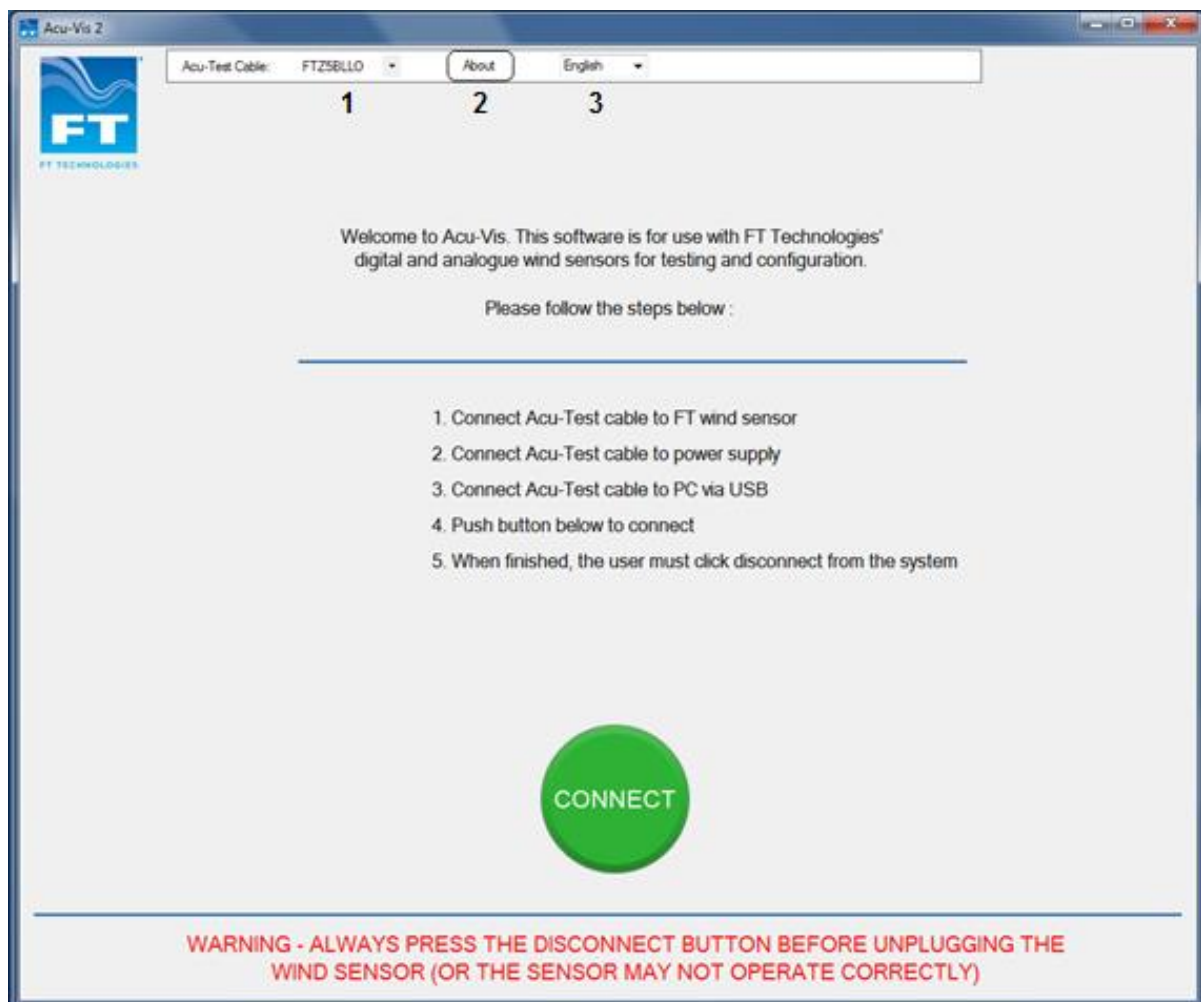


Figure 1: The Acu-Vis start screen and Acu-Test cable

Once connected the software will default to the 'Wind Readings' tab. This screen displays live wind speed, direction and temperature measurements from the sensor as well as some key details and settings the sensor is currently programmed with. The 'Settings' tab allows the user to alter some commonly altered settings on the sensor, including the averaging filter lengths.



Figure 2: The Wind Readings (left) and Settings (right) tabs in Acu-Vis

Once you have finished using Acu-Vis select the disconnect button in the top right of the screen to ensure any changes to sensor settings take effect. For further guidance on using Acu-Vis including how to unlock more advanced features, contact techsupport@fttechnologies.com.

2 INTRODUCTION

2.1 Product Overview

The FT602-D-SM is a solid-state ultrasonic wind sensor, using a patented Acoustic Resonance airflow sensing technique to measure accurately both wind speed and direction. The sensors have been specifically designed to operate in harsh environments including offshore areas. The wind sensor has no moving parts to degrade or wear and is designed for control applications requiring high reliability.

Mounting and aligning the sensor is very simple. A wind datum feature mark can be used to align the sensor to a reference point, while the integrated compass allows reference to magnetic North.

The FT602-D-SM allows communication via either RS422 (full-duplex), RS485 (half-duplex) UART or NMEA 2000 over CANBus.

2.2 Labelling

Each sensor is identified by its unique serial number found on the label as shown in Figure 3



Figure 3: Sensor Label

2.3 Scope of Use

The sensor is designed, manufactured and optimised for high availability.

No promise in part or full can be given to guarantee a sensor's continuous operation, as exceptional circumstances can occur that may result in the failure of the output from a sensor. Exceptional circumstances can include:

- Poor installation
- Inadequate inspection
- Power supply failures
- Poor quality electrical connections
- Lightning exposure
- Problematic environmental conditions or combination of conditions
- Physical Damage

Typically, higher levels of wind speed and wind direction data availability are achieved through the use of an additional FT sensor or alternative sensor. Control strategies or algorithms, which compensate in whole or in part, for any temporary interruption of data from individual sensors should also be applied. The choice and implementation of such methods is entirely the Purchaser's responsibility.

2.4 Disclaimer

There are no warranties, representations or conditions, expressed or implied of any kind given in this manual for any particular design application. The Purchaser should independently undertake sufficient testing to confirm validity and suitability of any design. The Purchaser assumes all risks and liability in conjunction with the use of the information given.

Any warranty given by FT Technologies in respect of the Equipment is conditional upon the sensor being handled, installed, integrated and operated in accordance within the guidelines given in this manual.

FT Technologies can take no responsibility for the effectiveness of any sensor lightning protection scheme implemented. The wind sensor has passed a wide range of EMC tests but FT Technologies does not warrant the sensor to survive lightning strikes.

Information supplied by FT Technologies Ltd. shall not be construed as permission to license to operate under, or recommendation to infringe any existing or pending patent, patent applications or trademarks.

3 FUNCTIONAL DESCRIPTION

3.1 Technical Performance

Sensor Performance^{1 & 2}

Measurement Principle Acoustic Resonance (compensated against variations in temperature, pressure and humidity)

Wind Speed Measurement

Range 0-75m/s
Resolution 0.1m/s
Accuracy $\pm 1\%$ (at 25 m/s at Datum)

Wind Direction Measurement

Range 0 to 360°
Sensor Accuracy $\pm 1^\circ$ (at 25 m/s at Datum)
Compass Accuracy 5° RMS
Resolution 1°

Acoustic Temperature Measurement

Range -20 to +60°C
Resolution 0.1°C
Accuracy $\pm 2^\circ\text{C}$ (at 21°C)

Pressure Measurement

Range 260 - 1260hPa
Resolution 0.0001hPa
Accuracy 0.5hPa

Tilt Measurement

Range -40 - +40°
Resolution 0.1°
Accuracy $\pm 7.5^\circ$

Environment

Temperature Range -20 to +60°C (operating)
-40 to +60°C (storage)
Humidity 0-100%
Altitude 0-4000m

Data I/O

Interface RS422 (full-duplex), RS485 (half-duplex), UART or CANBus.
Format ASCII, polled or Continuous Update modes, NMEA 0183 or NMEA 2000
Data Update Rate Up to 10 measurements per second (10Hz)

Power Requirements³

Supply Voltage 24VDC nominal (5-24 VDC range).
Sensor Current 28mA nominal (at 24VDC), 40mA (at 5VDC)

Physical

Weight 170g
Material UV stabilised polycarbonate blend
I/O Connector 12-way connector (PX0412/12P)
Mounting Method Surface-Mount with adaptor for pipe-mounting

Notes:

1. All specifications subject to change without notice.
2. Specifications calculated with the default settings and filters enabled.
3. See safety instruction requirements (page 5 and 6).

3.2 Wind Speed Calibration

The wind sensor is calibrated in our wind tunnels before dispatch. As the sensor has no moving parts, there is no need to recalibrate a sensor over its lifetime as no measurement degradation will occur. The sensor's compact strong monolithic shape is designed to prevent accidental transducer movement or damage. FT Technologies' calibration procedure and wind tunnels are designed to give a calibration profile that is within the accuracy limits set in the product technical specification (see Section 3.1). Periodically the accuracy of FT's wind tunnels are compared with the accuracy of an independent wind tunnel to ensure that no drift has occurred.

In exceptional circumstances users may wish to apply additional calibration factors. The sensor has an option to set a User Calibration Table, which can modify the reported wind speed. For further information on applying a User Calibration Table, please contact techsupport@fttechnologies.com.

3.3 Wind Speed and Direction Filtering

The sensor has optional internal filtering available. This is a digital finite impulse response (FIR) filter, which works by calculating the moving average of a fixed number of previous readings. If filtering is being applied externally, the sensor's output filters can be disabled. If filtering inside the sensor is preferred, the length of the averaging for both wind speed and direction can be independently set (See Sections 8.3.16 and 8.3.17).

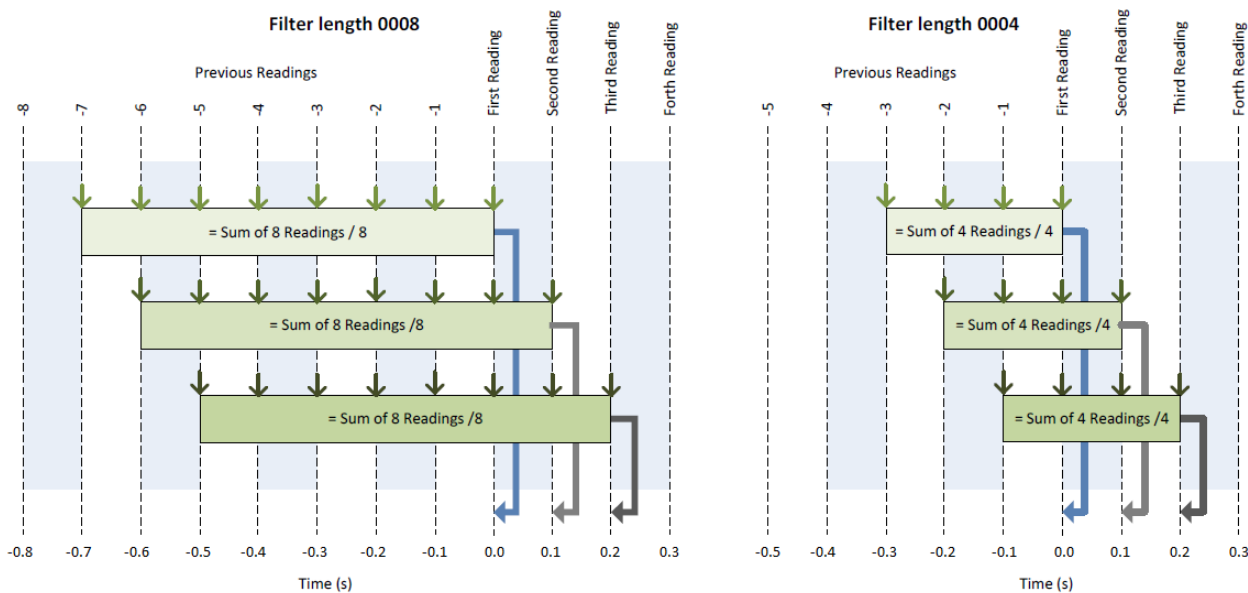


Figure 4: Examples of FIR Filtering



It is recommended to use average readings to reduce the effects of air turbulence.

3.4 Compass & Declination Angle Operation

1. The FT602-D-SM is fitted with a solid-state electronic compass so that the wind direction can be measured relative to magnetic North. There are three modes for the compass: Enabled (E), Standalone (S) and Disabled (D). When the compass is in Enabled mode, both wind direction and heading are reported relative to magnetic North. In Standalone mode, wind direction is decoupled from the heading and the sensor will report wind direction relative to the sensor datum mark on the wind sensor body (as per Figure 8). The compass heading can also be read back at any time using the CF command. See Section 8.3.7 for full details of the CF command. All data formats provide wind direction data relative to North (if the compass function is enabled). The NMEA 0183 message also has a field that indicates whether the compass module is enabled or disabled.

The effects of fixed local magnetic fields (i.e. those caused by magnetised ferrous material close to the compass) can be compensated for by performing an in-situ compass calibration. It is important that the calibration is performed with the FT602-D-SM mounted in its final position within the host system. The calibration can only compensate for fixed magnetic fields generated by the host system. Varying fields caused, for example, by ferrous objects passing near the FT602-D-SM (say by a car or truck) or high current electric circuits will not be corrected. Should the magnetic signature of the host system change significantly (because component parts have been added or removed), then a recalibration may be required.

3.4.1 Compass Convention

The compass heading convention further enhances the functionality of the FT602-D-SM. It specifies how the sensor reports compass headings based on its orientation and the Anemometer Mount setting:

- The sensor reports increasing compass heading (from 0° to 360°) when rotated clockwise while oriented normally (connector pointing down) and the Anemometer Mount set to Normal.
- Conversely, it reports decreasing compass heading when the Anemometer Mount is set to Inverted.

3.4.2 Compass Calibration

Compass calibration should be performed away from any external stray magnetic fields. To calibrate the FT602-D-SM compass proceed as follows:

- 1) Mount the FT602-D-SM in its final position within the host system
- 2) Ensure the compass is enabled (send the CFE 'enable compass' command, see Section 8.3.7).
- 3) Send the CFC 'enable calibration' command (see Section 8.3.7). Important: No further commands should be sent to the FT602-D-SM after the CFC command has been sent until the calibration is completed.
- 4) Slowly rotate the host system 1 complete revolution in the horizontal plane. The revolution should take between 40 and 60 seconds.
- 5) To complete the calibration send the CFE 'enable compass' command (see Section 8.3.7).
- 6) Store the new calibration Parameters in the compass module using the CFMS command. The save command, for example \$01,CFMS*//<cr><lf>, will save the new compass calibration parameters calculated during the calibration process to the Interface Board Flash memory. This command is required.
- 7) The sensor shall then be either reset via RSU command or power cycled to finish the calibration process.

3.5 Inclinometer Zeroing

The FT602-D-SM features a built-in inclinometer cable of measuring pitch and roll between +40° and -40°. Before using this feature, the user should zero the inclinometer once fixed in place in the user's system. The sensor should be mounted as close to level as possible to ensure the majority of the +40° to -40° operating range is available for use.

- 1) Enable the inclinometer using the CFE command.
- 2) With the sensor and system held still in its upright position send the CFTCS command to begin calibration.
- 3) Without moving the sensor wait for a minimum of 6.4 seconds before sending the CFTCF command to complete the calibration.

3.5.1 Pitch and Roll Conventions

In addition to compass functionality, the FT602-D-SM also incorporates conventions for measuring pitch and roll. The inclinometer convention specifies how the sensor reports pitch and roll based on its orientation:

- Positive pitch is indicated when the sensor is in its nominal orientation (connector facing downward) and tilted with the arrow mark facing up.
- Negative pitch is indicated when tilted when the arrow mark is facing down.
- Positive roll is reported when tilted to the right in nominal orientation, while negative roll is reported when tilted to the left.

These conventions ensure accurate reporting of the sensor's orientation, which is crucial for applications requiring precise measurements.

3.6 Error Detection

The sensor has a self-checking mechanism which can detect if a reading is invalid. On very rare occasions where an invalid reading may have been detected, this is signalled to the computer or data logger by setting a status flag character within the wind velocity output message (see Sections 8.3.22 and 8.3.23). The table below shows the possible flags and their meanings.

Status Flag	Description
0	Valid Data – The sensor is operating correctly and reporting valid wind data.
1	Invalid Data – The sensor is not reporting valid wind data, the sensor will attempt to recover. If this flag persists power cycle the sensor or contact FT Technologies for further advice.
2	Overspeed Warning – The sensor is operating correctly, however the measured wind speed exceed the operating limits of the sensor (75 m/s). The accuracy of this measurement cannot be guaranteed.
8	Signal Path Failure – The sensor has suffered a hardware failure in the measurement circuit. Results from this sensor should no longer be trusted and the sensor should be returned to FT Technologies for repair.

Figure 5: Wind measurement status flags and their meaning

It is important that error flags are not ignored. Data associated with a status flag other than 0 should not be processed as valid wind data. It is recommended that errors are monitored and logged. If the frequency of errors has recently increased, then inspection of the sensor for physical blockages may be required (see Section 5.1).

4 MECHANICAL & ELECTRICAL INSTALLATION

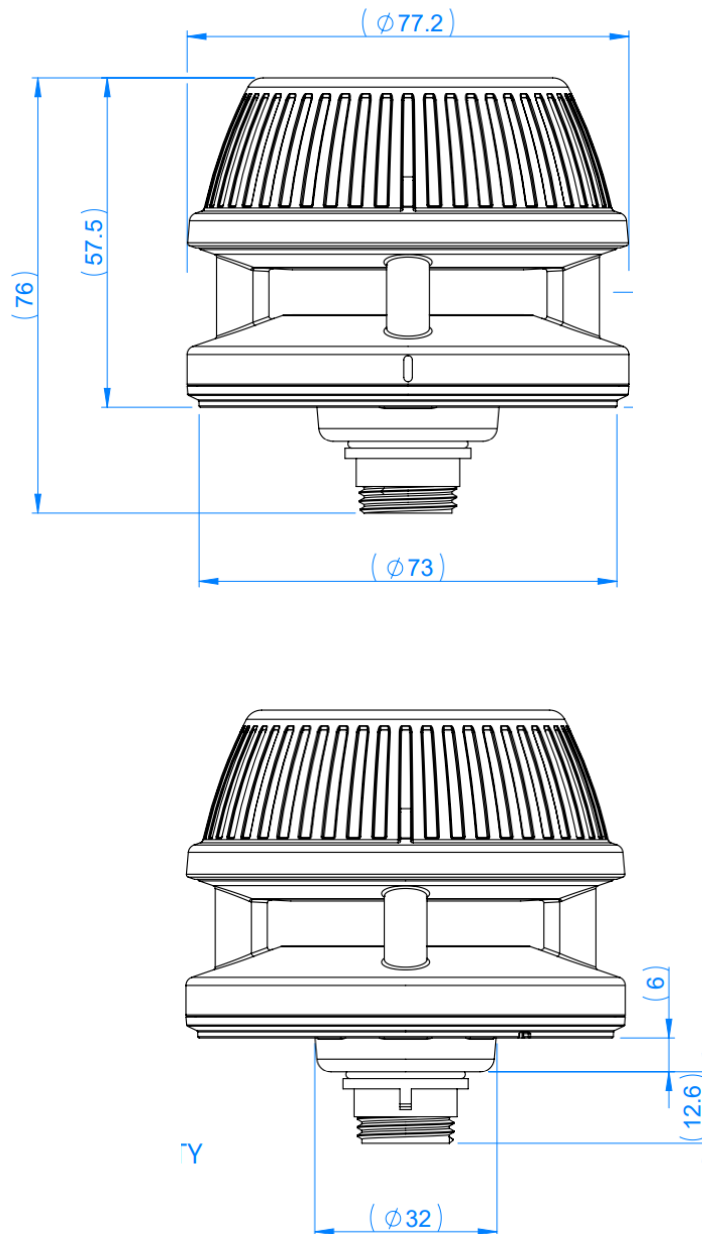


Figure 6: Profile and Measurements of the FT602

The FT602 is designed to be mounted to a flat surface using three M3 screws. The mounting holes have an internal thread depth of 5 mm, and 3 screws are provided with the sensor. Depending on the thickness of your mounting surface it may be appropriate to use longer screws. When selecting a mounting location, care should be taken to ensure airflow to the sensor is not obstructed or influenced by nearby objects.

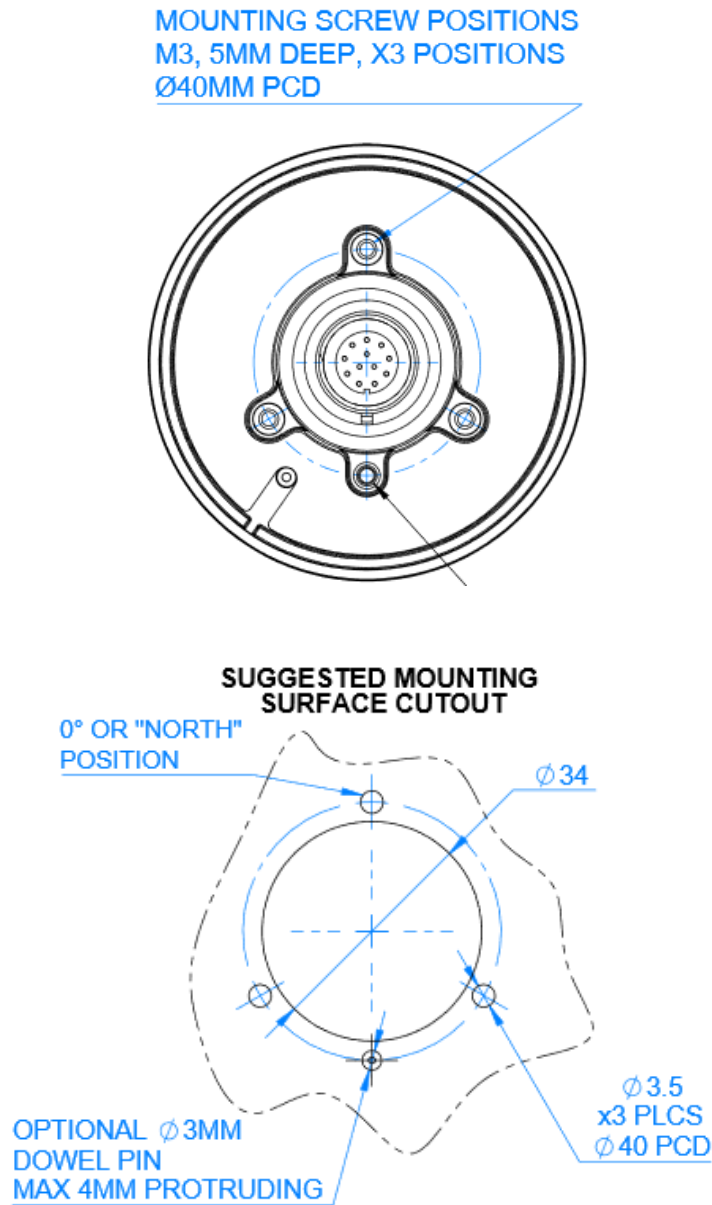


Figure 7: View of the bottom face (left) and base (right) including mounting screw positions, optional dowel pin and breather hole.

The sensor has a breather hole in the bottom face of the sensor to allow internal pressure to equilibrate with the ambient conditions and prevent pressure build-up. The breather hole features a gore vent that prevents moisture ingress to the sensor. It is important to ensure airflow to the breather hole. Failure to do so will result in accurate pressure measurements.

Also provided with the sensor is a pipe mount adaptor which allows the sensor to be mounted to a 25 mm external diameter pipe. The adaptor connects to the base of the sensor using the three M3 screws provided. The adaptor features a dowel pin to ensure the alignment marking on the adaptor match those on the sensor.

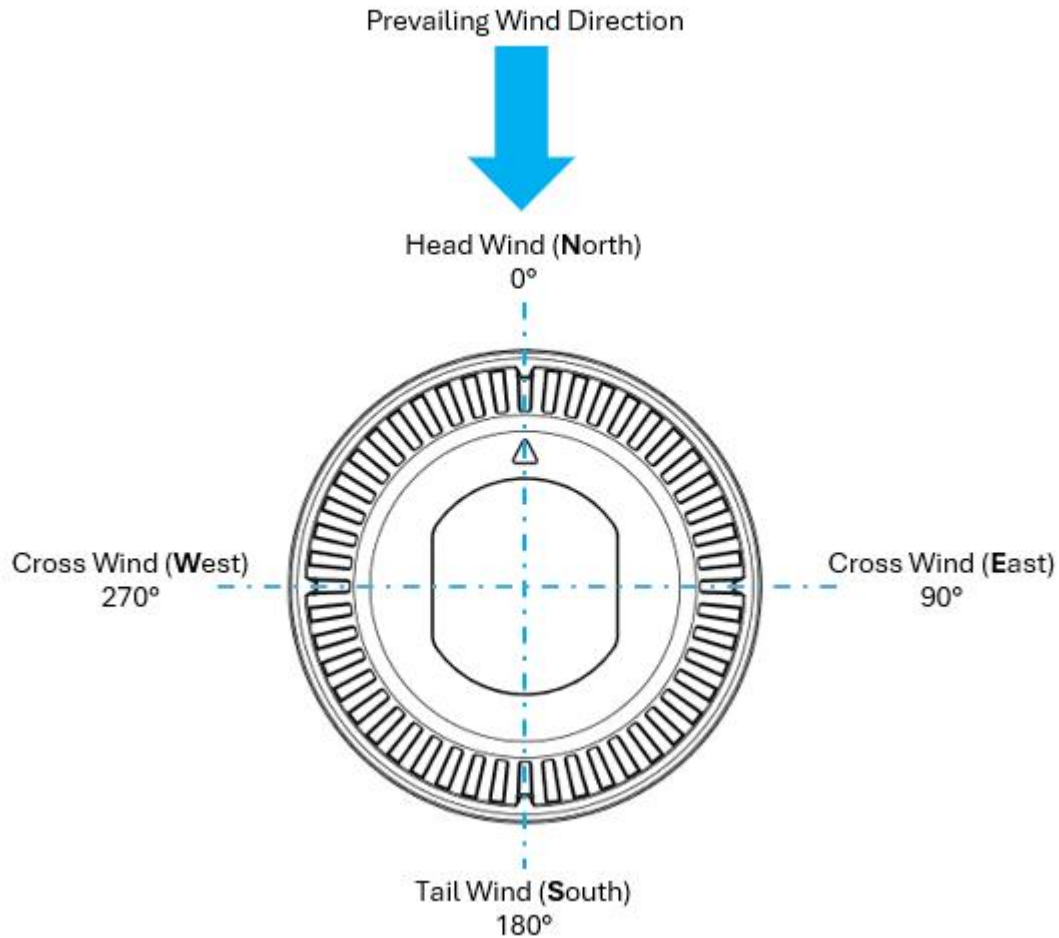
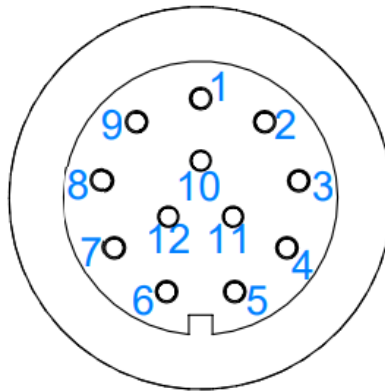


Figure 8: Orientation of the FT602 when the compass is disabled.

The sensor features a datum indicator on the top face of the sensor to aid with aligning the sensor. When the compass is disabled in the sensor, the sensor will report a wind direction of 0° when the wind is blowing along the axis of the indicator (as shown by the arrow in Figure 8). When the compass is enabled, the sensor will report wind direction relative to magnetic north.

4.1.1 Connector Details

All electrical connections are made to the digital sensor via a Bulgin 400 Series Buccaneer (PX0412-12S) 12-way connector located in the base of the wind sensor housing. The connector pin designations are shown in Figure 9. The connector conforms to IEC 61076-2-101.



Pin	Function	FT016 Core Colour
1	5-26VDC	Red
2	0V	Black
3	Data Ground	Grey
4	RS422 TX Data A -	Yellow
5	RS422 TX Data B +	Green
6	RS485/RS422 RX Data B +	Red-Blue
7	RS485/RS422 RX Data A -	Violet
8	UART RX	White
9	UART TX	Grey-Pink
10	N/C	Brown
11	CAN-Low	Blue
12	CAN-High	Pink

Figure 9: Sensor Connector Pin

When correctly mated with a Bulgin PX0410/12S/XXXX connector using SA3179 contacts. The connector is protected to IP67. For additional protection, a cold shrink or heat shrink sleeve can be applied over the mated connector.

The FT016 cable features the Bulgin connector and pins listed above and comes in a 1.5m cable length, this cable is capable of communication with all FT602-D-SM communication modes. Other lengths and specific communication cables are available from FT, check the website for the latest range.

5 SERVICE, CONFIGURATION & TESTING

5.1 Inspection

The following checks are required to identify any signs of corrosion or damage on the sensor which may hinder its performance. It is recommended that these checks be carried out annually.

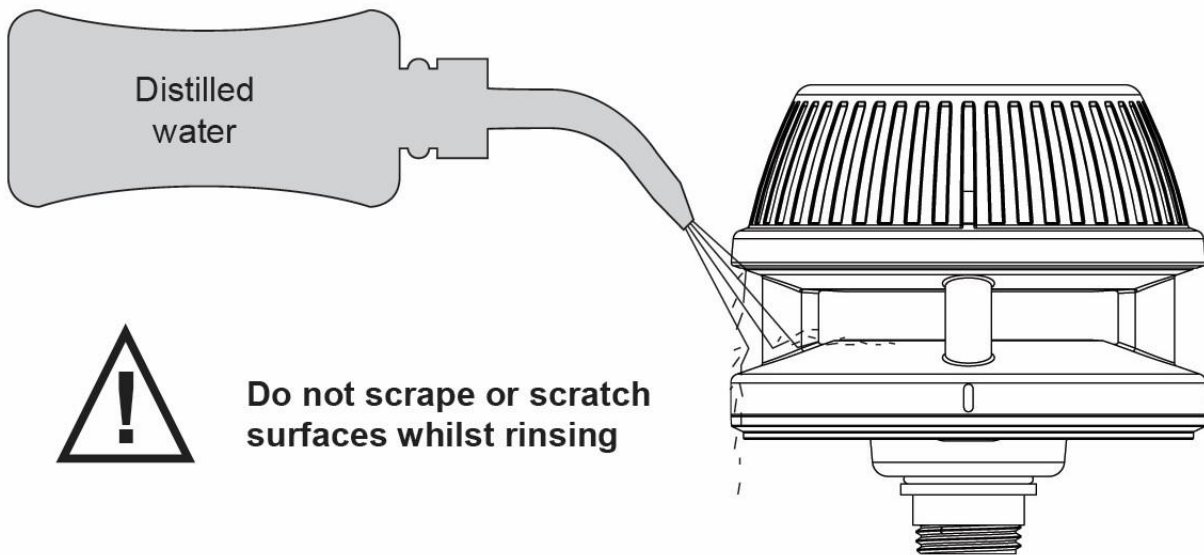


**DO NOT INSERT OBJECTS INSIDE CAVITY
INTERNAL SURFACES
MAY BE DAMAGED**

Mechanical damage: Check the sensor body for signs of damage, paying particular attention to the seals. Also inspect for signs of lightning damage which may appear as burns or scorch marks (or burnt smells). If damage has occurred replace the sensor immediately. Check the O-ring seal and fixings are in good condition, replace them if necessary for long-term protection.

Interconnection cable: Inspect the condition of the cable. If it has become frayed or damaged it should be replaced. Intermittent cable faults may not be visible and may show up as data errors. Check the intended network component values (termination resistors etc.).

Connector protective sleeve: If cable protection is used, check for any signs of damage or degradation.



Cleaning: If any debris is present this can be removed by gently rinsing the measurement cavity surface with distilled water using a laboratory wash bottle or similar. Please note excess water droplets can be removed by lightly blowing or shaking the sensor.

Do not scrape or scratch the surfaces whilst rinsing. **Under no circumstances** should objects be inserted inside the measurement cavity, as this can cause irreparable damage. The body of the sensor can be washed if required using the same method as described above. Whilst washing the sensor care must be taken not to get water in the breather hole or into the connector at the base of the sensor.

Do not use cleaning chemicals to clean the sensor. If washing a nearby item protect the sensor with a suitable cover. Ensure the cover is removed before re-enabling the wind data survey.

5.2 Fault Finding & Troubleshooting

To determine whether a sensor has a fault carry out the following steps:

- Follow the inspection procedure above to identify signs of physical damage.
- Remove any objects or insects lining the cavity or blocking the airflow.
- Reset the sensor (RSU command or power-cycle).

If there are signs of physical damage and/or the sensor is failing to communicate properly, it should be replaced. Sensors may be returned to FT Technologies for further analysis if required (see Section 5.3).



Warning – The sensor contains no user serviceable components. Do not attempt disassembly as damage may result and product warranties will be invalidated.

During extreme weather conditions there may be periods where data is temporarily unavailable. However there are ways to mitigate against these affects. The following steps should be taken to ensure the highest levels of data availability from the sensor:

- Check that the wind sensor data and status flag errors are being processed as per the advice in Sections 8.3.22 and 8.3.23.
- Ensure that any installed lightning and EMI protection (see Section 4) is in good condition and any cable shielding is terminated at both ends. All mating surfaces must be free of paint and corrosion.

Please contact FT Technologies for further information and advice if required.

5.3 Returns

If a sensor appears to be faulty, compile a detailed fault description for each sensor, then contact FT Technologies to request a Returns Materials Authorisation (RMA) form. Please complete the form and return as instructed. Returns cannot be accepted without prior approval via this authorisation form.

Units damaged by lightning or disassembled by the customer cannot typically be repaired, however an inspection fee may still apply.

6 SENSOR COMMUNICATION

6.1 Introduction

The sensor features an ASCII-based communication protocol transmitted over either RS422 (full-duplex), RS485 (half-duplex) or UART (full-duplex, 3V and 5V TTL compatible). The protocol incorporates checksum validation to ensure the integrity of all data transmissions. In addition to the FT Technologies proprietary protocol the sensor can output the common NMEA 0183 MWV (Wind Speed and Angle) sentence.

The FT602 is also capable of communicating via NMEA 2000 over CANBus. NMEA 2000 is enabled, disabled and configured via serial commands described in this section. The NMEA 2000 protocol is described in further detail in Communication Over CANBus.

6.2 Conventions used in this manual

All examples of sensor transmitted and received messages are printed in italic courier monospace font, e.g.

```
$<listenerID>,DFP*<checksum><cr><lf>
```

Angle brackets are used as placeholders for data (e.g. <wind speed>) or for non-printable ASCII characters (e.g. <cr> for carriage return). Figure 10 lists the various special characters and symbols which are used in the examples given in this Manual.

Symbol	Valid Characters	HEX Values	Definition
a	{A to Z} {0 to 9} {/}	41-5A, 30-39, 2F	Talker/listener address field characters
c	{A to Z} {0 to 9}	41-5A, 30-39	Fixed length field of alpha (upper case only) and numeric characters
h	{A to F} {0 to 9} {/}	30-39, 41-46, 2F	Checksum field validation characters
s	{ }	20	Space
x	0 to 9	30-39	Fixed length field of numeric characters
x.x	{0 to 9} {.	30-39, 2E	Fixed point numeric field (i.e. always with leading and trailing zeros)
±	{+ -}	2B, 2D	Polarity indicator. Where a value can take on both positive and negative values a polarity indicator (either + or -) is always sent as the first character in the field.
	{ \$ }	24	Start of message delimiter
	{ * }	2A	Checksum field delimiter
	{ , }	2C	Field delimiter
-	{ - }	2D	Dash
<cr>		0D	Carriage return
<lf>		0A	Line feed
<name>			Placeholder for data

Figure 10: Symbols used in this Handbook

6.3 Data Transmission

Data is transmitted and received via an asynchronous serial communication interface using ASCII characters. The interface operates with the following parameters:

Parameter	Setting
Baud Rate	1200, 2400, 4800, 9600 (factory default), 19200, 38400
Data Bits	8
Start Bits	1
Stop Bits	1
Parity	None

Figure 11: Data Transmission Parameters

To set the sensor baud rate use the BR command (Section 8.3.1)

The sensor does not use handshaking (either hardware or software) to control the flow of data to and from the host computer. It is important, therefore, that the serial interface of the host computer is set with handshaking/flow control disabled.

6.4 Message Format

Data communication between the sensor and the host computer is performed by the transmission of ASCII messages. Figure 12 shows the composition of the message. The same message format is used for both received and transmitted messages.

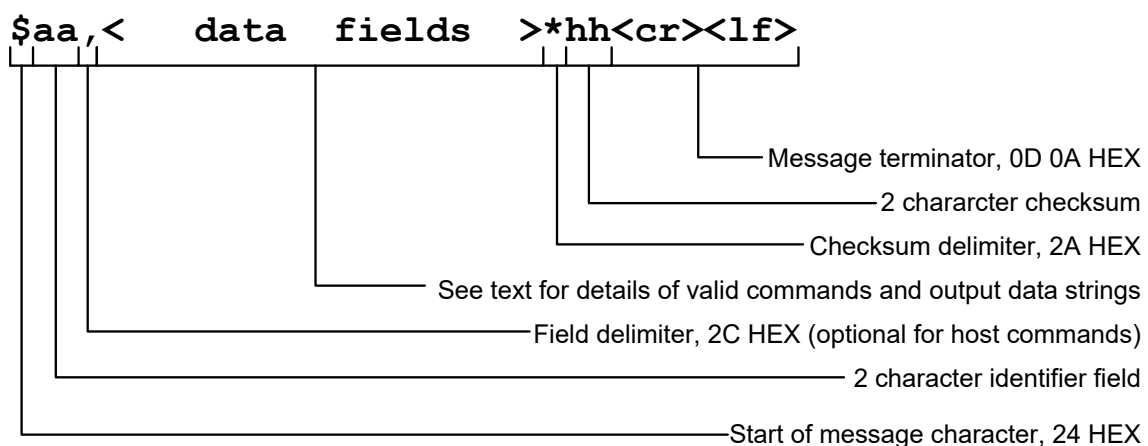


Figure 12: Message Format

All messages start with the '\$' start of message character, followed by the 2-character talker/listener identifier (Section 6.5) field.

Following the first delimiter is the main body of the message which comprises a variable number of data fields (dependent on the message being transmitted), each separated by the field delimiter character (','). Data fields may contain alpha, numeric, or alphanumeric data depending on the information content of the field.

Messages sent to the sensor will contain a command in <data fields> and messages transmitted from the sensor will contain output data in <data fields>.

The data field section of the message is terminated by the checksum delimiter character '*'. Following the checksum delimiter is the two-character checksum field. See Section 0 for information on how to compute the checksum and Section 0 if checksum message validation is not required.

All messages are terminated with a carriage return <cr> and line feed <lf>.

6.5 Listener and Talker Identifiers

The sensor is assigned with both a Listener and Talker identifier address that allows an individual sensor to be uniquely identified in a system comprising more than one sensor.

Whenever a message is sent to the sensor, the identifier field of the message (the 2 characters immediately following the '\$' start of message character) must correspond to the sensor Listener identifier address, otherwise the sensor will ignore the message. In applications where more than one sensor is connected to the RS422/RS485 bus, you should assign each sensor in the system a unique Listener ID. The host computer will then be able to address individually each sensor.

If you do not wish to use the Listener ID in messages sent from the host computer, you can replace the Listener ID with '//'. Sending '/' in place of the Listener ID will allow any sensor, irrespective of its Listener ID setting, to respond to the message. When using multiple sensors in a network, QUERY commands should not be sent using '/' since this will cause all sensor units to transmit data resulting in bus contention. The '/' address characters can be used to send a SET command simultaneously to all sensor units, although the ID command should not be used in this way.

Whenever a message is transmitted from the sensor, the identifier field of the message (the 2 characters immediately following the '\$' start of message character) will contain the Talker ID. The Talker ID is used as a message tag to identify which sensor has transmitted the message.

The factory default value for the Listener ID is 01 and for the Talker ID it is W1 (Weather Instrument). To change the Listener and/or Talker ID use the ID Command, Section 8.3.18.

6.6 Calculating the Message Checksum

All messages sent to, or received from, the sensor include a checksum field. Messages that are transmitted from the sensor always include a checksum value in the checksum field. Messages sent to the sensor by the host computer can either contain a checksum value or an 'ignore checksum identifier' in the checksum field.

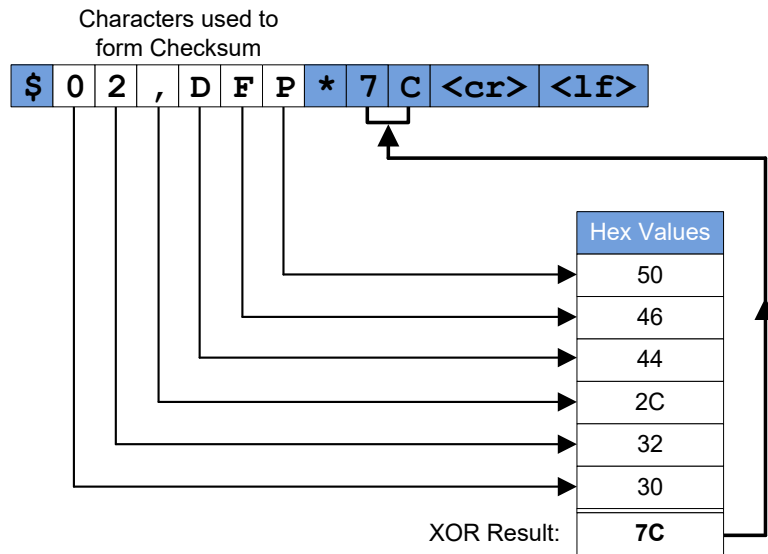


Figure 13: Checksum Example

The checksum value is calculated by Exclusive OR'ing (XOR'ing) all the bytes between (but not including) the '\$' and the '*' characters of the message. The resulting single byte value is then represented by 2 HEX characters in the message string. The most significant character is transmitted first.

Note: Since a message only contains ASCII characters (which have values in the range 0-7F) the checksum value will always be between 0 and 7F.

6.7 Disabling the Checksum

All messages which are sent to the sensor must contain a valid checksum value in the checksum field, otherwise the sensor will not process the incoming message.

Although it is recommended that a checksum value be computed for all messages which are sent to the sensor, in some cases this may not be convenient (i.e. when communicating with the sensor with a terminal).

To prevent the sensor from performing checksum validation of incoming messages, send the ASCII characters '/' in place of the checksum value.

Example:

Send a message to set the data output format to Polar using the DFP command (the sensor Listener ID in this example is set to 02)

With a checksum (sensor checksum validation automatically enabled):

```
$02,DFP*50<cr><lf>
```

Without a checksum (sensor checksum validation automatically disabled):

```
$02,DFP*//<cr><lf>
```

A checksum value is always transmitted by the sensor with every outgoing message. However, the checksum field can be ignored by the host computer if checksum validation for received messages is not required.

6.8 Communicating via Command Line Terminal Programs (Including TeraTerm)

The FT602 sensor can be communicated with using a command line terminal application using either the FT063 cable or a user created cable.

One terminal example is TeraTerm, an open-source terminal application, although any terminal application should be compatible. The below guide details how to configure TeraTerm to provide a user-friendly interface with the sensor.

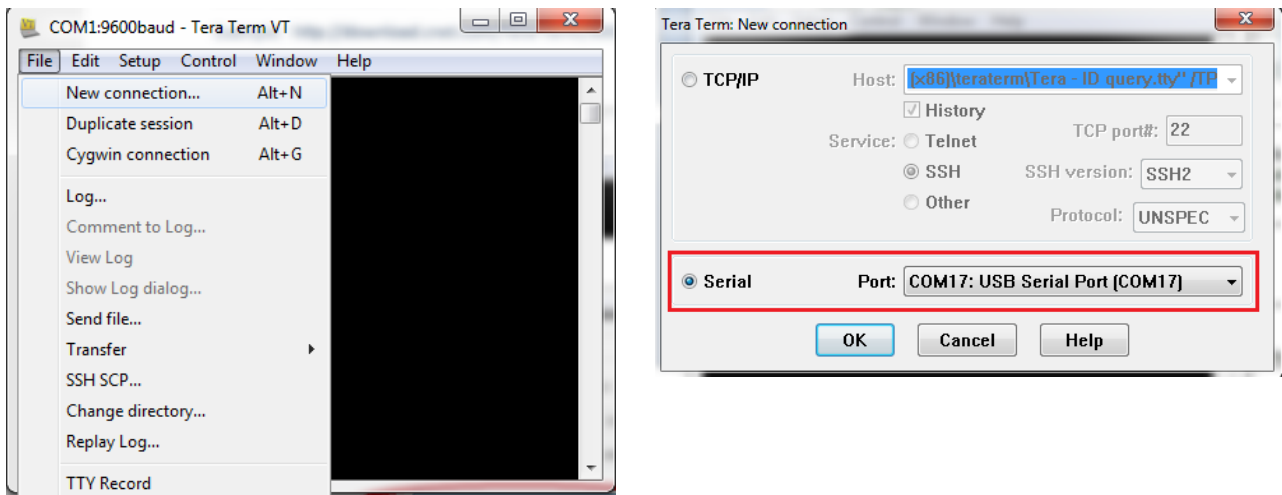


Figure 14: Creating a connection in TeraTerm

After opening TeraTerm, select the “New connection” option in the File Menu. Select serial connection and find the correct COM channel.

This will be automatically assigned by your computer and should be described as a “USB Serial Port”.

To make the terminal as simple to use as possible, we recommend making the following changes in the Setup>Terminal interface.

Set the New Line Transmit to “CR+LF” to prevent the need to type the end of line characters and enabling Local Echo as shown below.

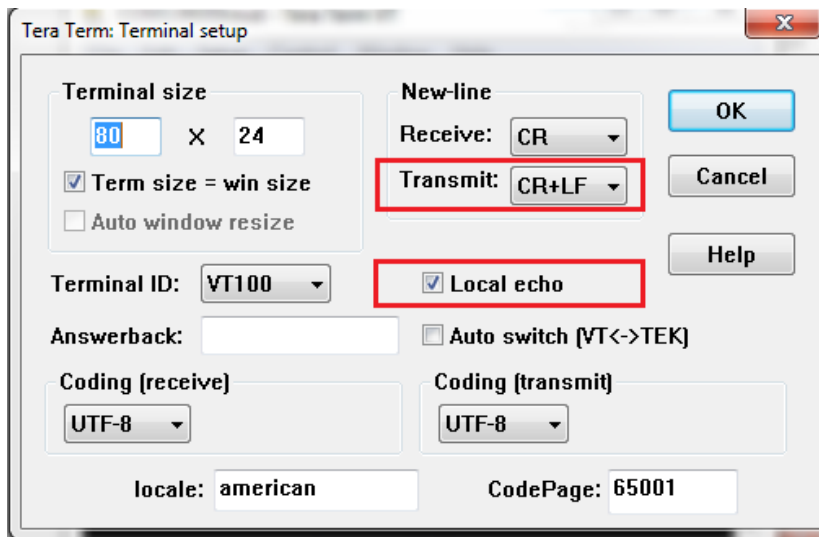


Figure 15: Modifying Terminal Settings

At this point commands and queries can now be sent to the sensor. For further details on what commands can be sent, see Section 8.3.

Please note; TeraTerm defaults to 9600 baud rate, if you alter the sensor baud rate you will need to change this in the Setup>Serial Port menu.

Below are some useful commands:

```
$01,ID?*// Query the sensor ID setting
$01,CU?*// Query the CU mode setting
$01,CUD*// Disable CU mode
$01,CUE00010*// Enable CU mode at 1 second intervals (1Hz).
```

The // line bypasses the checksum validation. Checksum validation is recommended for production datalogging to improve data integrity. See Section 6 and Section 8 for further details regarding command structures and replies.

Note: If a sensor is set to an ID setting other than 01, replace \$01 with the alternative ID. Starting a command with \$// will perform the relevant command on all connected sensors.

7 Communication Over CANBus

In addition to the ASCII based communication detailed in Section 6, the FT602 is capable of communicating over CANBus. In this communication mode the sensor communicates using the NMEA 2000 protocol only. This protocol includes a number of NMEA 2000 PGNs to communicate a variety of meteorological and navigation information.

In line with the NMEA 2000 protocol, all messages are transmitted at a baud rate of 250k. The FT602 is capable of transmission only over CANBus. All configuration of the FT602 must be performed over one of the ASCII communication modes detailed in 21. The sensor will transmit the selected PGNs at the frequencies defined by the NMEA 2000 standard.

7.1 Mandatory PGNs

PGN 126993	Heartbeat	Transmitted once per minute Contains equipment status – a fault value indicates the sensor has lost resonance and is trying to recover. If the fault persists contact FT for further advice.
------------	-----------	---

Figure 16: Mandatory PGNs

Figure 16 gives an overview of the mandatory PGNs that are always enabled in the FT602. For full details of the message structure(s) please consult the NMEA 2000 standard.

7.2 Configurable PGNs

PGN 127250	Vessel Heading	Transmitted 10 times per second Contains Heading, applied magnetic correction and heading reference (magnetic or true north)
127257	Attitude	Transmitted Once per second Contains Yaw, Pitch and Roll data.
PGN 130306	Wind Data	Transmitted 10 times per second Contains wind speed, wind direction, and wind reference (apparent or relative to compass heading)
PGN 130314	Atmospheric Pressure	Transmitted once every 2 seconds. Contains pressure in Pascals
PGN 130316	Temperature	Transmitted once every 2 seconds. Contains temperature in Kelvins
PGN 65302	WV Status	Transmitted 10 times per second Field three contains the wind data status flag, either 00, 01, 02 or 8 corresponding to status flags 0,1,2 and 8 respectively. All other fields should be ignored. For further details on interpreting each status flag see section 3.5.1.

Figure 17: Configurable PGNs

Figure 17 give a brief overview of the configurable PGNs available in the FT602. All 6 commands can be enabled or disabled as required using the CACP command. For full details of the message structure(s) please consult the NMEA 2000 standard.

8 PARAMETER SETTINGS

8.1 Command Types

8.1.1 Set Commands

Figure 18 lists the commands that may be sent to the sensor from the host computer that are used to SET configuration options for the wind sensor.

Command	Mnemonic	Configuration Options	Factory Default	Section
Anemometer Mount	AM	Normal Inverted	Normal	8.3.1
Acoustic Temperature Units	AT	C (Celsius) F (Fahrenheit) K (Kelvin) Filter Length	C (Celsius) 1 minute	8.3.2 8.3.3 8.3.4
Barometric Pressure	BM	Enabled or disabled	Disabled	8.3.5
Serial interface baud rate	BR	1200, 2400, 4800, 9600, 19200, 38400	9600	8.3.6
Compass Settings	CF	Enabled or disabled 000.0° to 359.9° Declination angle	Disabled 0.000°	8.3.7
Communication Interface	CI	RS485, RS422 or UART	RS485	8.3.9
CANBus Interface	CIC	Enabled or Disabled	Disabled	8.3.10
NMEA PGNs	CICP	Various PGNs see 7.2 Enabled or Disabled	Disabled	8.3.11
Continuous Update	CU	Enable or Disable Update interval, 0.1-6000 seconds	Disabled	8.3.12
Wind velocity data format	DF	Polar, Combined, 'T' format or NMEA	Polar	8.3.13
Command delay interval	DL	00 to 20	01	8.3.14
Clear Error Report	ER	Reset	000000000000	8.3.15
Wind Data filter	FL	Enabled or Disabled Speed filter length Direction filter length	Enabled 0016 0016	8.3.16 8.3.17
Listener and talker identifiers	ID	Listener ID = xx Talker ID = xx	Listener ID = 01 Talker ID = WI	8.3.18
Min/Max wind speed	MM	Reset	999.9,000.0	8.3.19

Figure 18: Set Commands

When a valid message is recognised by the sensor, the sensor will carry out the command contained in the message. To verify that the command has been successfully carried out, an associated QUERY command can be sent after most SET commands (see Section 8.1.2 for the list of parameters which may be queried).

All user parameter settings are stored in non-volatile memory and are retained when the sensor is switched off. When the sensor is next switched on (or a user reset command is sent) the sensor will revert to these settings. The sensor can therefore be configured prior to final installation if required.

8.1.2 Query Commands

Figure 19 lists the commands that may be sent to the sensor from the host computer that are used to QUERY the wind sensor's latest readings or configuration.

Command	Mnemonic	Sensor Data Returned	Section
Anemometer Mount	AM	Normal Inverted	8.3.1
Acoustic Temperature	AT	Temperature Temperature Units (°C, °F, K) Filter time period	8.3.2 8.3.3 8.3.4
Barometric Pressure	BM	Barometric Pressure Barometer Status	8.3.5
Serial interface baud rate	BR	1200, 2400, 4800, 9600, 19200, 38400	8.3.6
Compass Settings	CF	Enabled, Standalone or Disabled Compass status 000.0° to 359.9°	8.3.7
Communication Interface	CI	RS485, RS422 or UART	8.3.9
CANBus Interface	CIC	Enabled or Disabled	8.3.10
NMEA PGNs	CICP	Enabled or Disabled	8.3.11
Continuous update	CU	Enabled or Disabled Update interval (0.1 - 6000 seconds)	8.3.12
Wind velocity data format	DF	Polar, Combined, 'T' format or NMEA	8.3.13
Command delay interval	DL	00 to 20	8.3.14
Error report	ER	Factory Report	8.3.15
Wind Data filter	FL	Enabled or Disabled Speed filter length, 1-64 Direction filter length, 1-64	8.3.16 8.3.17
Listener and talker identifiers	ID	Listener ID = xx Talker ID = xx	8.3.18
Min/Max wind speed	MM	Min and Max Speeds Recorded	8.3.19
Serial Number	SN	Serial Number	8.3.20
Software Version	SV	Software Version	8.3.21
Wind velocity reading	WV	Wind Speed, Direction and Sensor Status	8.3.22 8.3.23

Figure 19: Query Commands

8.2 Timing Constraints

When a valid command is received by the sensor input buffer, there will be a time delay whilst the command is being processed. The actual command latency depends on exactly when the last character of the command is received within the sensor internal processing cycle. The sensor can process only one SET or QUERY command at a time.

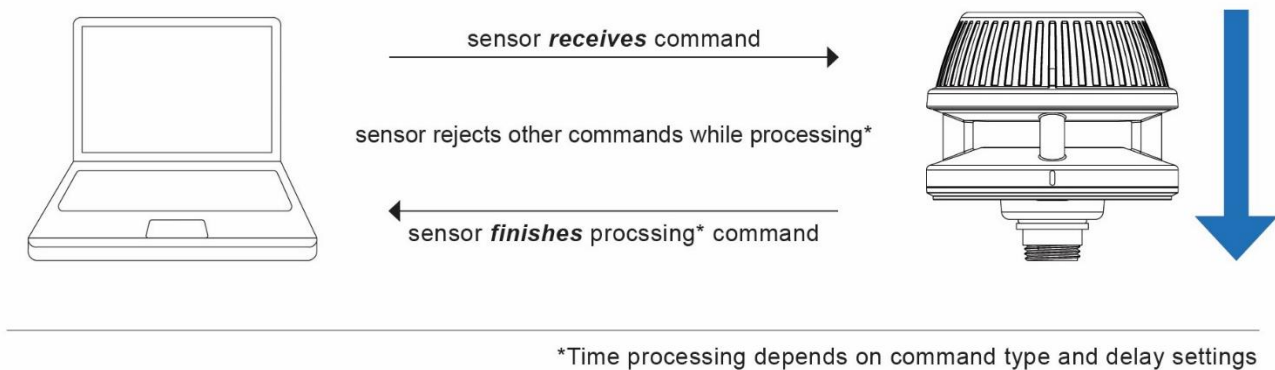


Figure 20: Command Processing

Once a SET command has been received by the sensor, it can take approximately 400ms for the command to be processed and any setting change implemented. If other commands are sent during this period, they may be ignored by the wind sensor. Therefore, all SET commands must be separated by a period of at least 500ms before further commands are sent.

Once a QUERY command has been received by the sensor, it takes up to 50ms for the command to be processed. The sensor will then wait for a predefined delay before sending a response. This delay time is programmable in increments of 50ms.

See Section 8.3.14 for details of how the delay between the sensor receiving a command and transmitting a reply can be adjusted. An additional latency is recommended depending on the time taken for the computer to switch between transmit and receive modes.

If the delay time has been extended using the DL command, then the next QUERY command sent to the wind sensor must be separated by the maximum QUERY command processing time (50ms) and at least the DL interval (50ms default.) It is therefore recommended that the frequency of any QUERY command does not exceed 10Hz, i.e. 10 commands per second.

Message Example:

For example, to set the wind velocity output to Polar format and verify that the command has been accepted, send the following commands:

Set the wind reading format to polar:

```
$//DFP*//<cr><lf>
```

Then wait 500ms for the SET command to be implemented by the sensor.

A QUERY can then be sent to confirm command has been carried out:

```
$//DF?*//<cr><lf>
```

Then wait 50-100ms for the sensor to send a response:

```
$WI,DF=P*5D<cr><lf>
```

Please note the above example assumes the sensor has a factory default time delay of 50-100ms (DL01).

8.3 Command Parameters

Each command, and its usage, is described in the following Sections. All examples, other than where stated, assume that the sensor Listener ID is set to 01, and the sensor TalkerID is set to WI, (Weather Instrument).

8.3.1 AM: Set or Query Anemometer Mount Orientation

Command Parameter	AM									
Command Syntax	SET Sensor:	<code>\$<listenerID>,AM<orientation>*<checksum><cr><lf></code> <code>\$aa,AMx*hh<cr><lf></code>								
	QUERY Sensor:	<code>\$<listenerID>,AM?*<checksum><cr><lf></code> <code>\$aa,AM?*hh<cr><lf></code>								
	Sensor output:	<code>\$<talkerID>,AM=<orientation>*<checksum><cr><lf></code> <code>\$aa,AM=x*hh<cr><lf></code>								
Parameters	<orientation> N I Set Normal orientation for mounting the sensor in ‘regular’ orientation Set Inverted orientation for mounting the sensor ‘upside down’									
Examples	Example 1 Set the anemometer mount orientation to inverted and verify the new setting: <table><tr><td><u>Message</u></td><td><u>Comment</u></td></tr><tr><td><code>\$01,AMI*//<cr><lf></code></td><td>Set Inverted orientation</td></tr><tr><td><code>\$01,AM?*//<cr><lf></code></td><td>Query orientation setting</td></tr><tr><td><code>\$WI,AM=I*4A<cr><lf></code></td><td>Inverted orientation setting</td></tr></table>		<u>Message</u>	<u>Comment</u>	<code>\$01,AMI*//<cr><lf></code>	Set Inverted orientation	<code>\$01,AM?*//<cr><lf></code>	Query orientation setting	<code>\$WI,AM=I*4A<cr><lf></code>	Inverted orientation setting
	<u>Message</u>	<u>Comment</u>								
<code>\$01,AMI*//<cr><lf></code>	Set Inverted orientation									
<code>\$01,AM?*//<cr><lf></code>	Query orientation setting									
<code>\$WI,AM=I*4A<cr><lf></code>	Inverted orientation setting									
Description	Use the AM command if changing the mounting of the anemometer between right way up and upside down. In inverted orientation the sensor’s compass and wind direction readings will be reversed to provide correct direction and heading output.									

8.3.2 AT.1: Query the Acoustic Temperature

Command Parameter	AT	
Command Syntax	SET Sensor:	N/A
	QUERY Sensor:	<pre>\$<listenerID>,AT?*<checksum><cr><lf> \$aa,AT?*hh<cr><lf></pre>
	Sensor output:	<pre>\$<talkerID>,AT=<temp>,<units>,<status>*<checksum> <cr><lf> \$aa,AT=±xxx.x,c,c*hh<cr><lf></pre>
Parameters	<temp> -077.8 to +368.3	Read only parameter that returns the current acoustic temperature reading in the range: -61.0°C to +95.1°C -77.8°F to +203.2°F 212.2K to 368.3K
	<units> C F K	The Acoustic Temperature is output in degrees Celsius The Acoustic Temperature is output in degrees Fahrenheit The Acoustic Temperature is output in Kelvin
	<status> A V	The acoustic temperature reading is valid An acoustic temperature reading is being acquired. The output may not be valid during this time
Examples	Example 1 – Query Temperature °C	
	Query the latest acoustic temperature reading – a valid reading in degrees Celsius.	
	<u>Message</u>	<u>Comment</u>
	\$01,AT?*/<cr><lf>	Query acoustic temperature
	\$WI,AT=+023.9,C,A*15<cr><lf>	Sensor response
	Example 2 – Query Temperature °K	
	Query the latest acoustic temperature reading..	
	<u>Message</u>	<u>Comment</u>
	\$01,AT?*/<cr><lf>	Query acoustic temperature
	\$WI,AT=+297.2,K,V*0C<cr><lf>	Sensor response
	Example 3 - CU AT DFC Mode	<u>Comment</u>
	\$WI,WVC=000.0,333,0,+026.4,C,A*4B	CUE mode with AT (DFC Mode)
	Example 4 – WV DFC Mode	<u>Comment</u>
	\$WI,WVC=000.0,323,0,+026.3,C,A*4D	WV response (DFC Mode)
Description	The AT query command returns the current acoustic temperature reading in the selected units. An acoustic temperature reading will be returned even if the sensor is in the process of trying to acquire a valid reading. The acoustic temperature reading is filtered with a one-minute time constant by default.	

8.3.3 AT.2: Set or Query the Acoustic Temperature Units

Command Parameter	AT (units)	
Command Syntax	SET Sensor:	<code>\$<listenerID>,ATU<units>*<checksum><cr><lf></code> <code>\$aa,ATUc*hh<cr><lf></code>
	QUERY Sensor:	<code>\$aa,AT?U*hh<cr><lf></code>
	Sensor output:	<code>\$<talkerID>,AT=<units>*<checksum><cr><lf></code> <code>\$aa,AT=c*hh<cr><lf></code>
Parameters	<units>	
	C	The Acoustic Temperature is to be output in degrees Celsius
	F	The Acoustic Temperature is to be output in degrees Fahrenheit
	K	The Acoustic Temperature is to be output in Kelvin
Examples	<u>Example 1</u>	
	Set the Acoustic Temperature reading output units to degrees Fahrenheit.	
	<u>Message</u>	<u>Comment</u>
	<code>\$01,ATUF*//<cr><lf></code>	Set units to Fahrenheit
	<code>\$01,AT?U*//<cr><lf></code>	Query temperature units
	<code>\$WI,AT=F*5C<cr><lf></code>	Sensor response
Description	Use this command to change the measurement units of the acoustic temperature output readings. The available measurement units are degrees Celsius, Fahrenheit and Kelvin.	

8.3.4 AT.3 Set or Query the Acoustic Temperature Filter Length

Command Parameter	AT (filter length)	
Command Syntax	SET Sensor:	<code>\$<listenerID>,ATF<filter length>*<checksum><cr><lf></code> <code>\$aa,ATFxxc*hh<cr><lf></code>
	QUERY Sensor:	<code>\$<listenerID>,AT?F*<checksum><cr><lf></code> <code>\$aa,AT?F*hh<cr><lf></code>
	Sensor output:	<code>\$<talkerID>,AT=<filter length>*<checksum><cr><lf></code> <code>\$aa,AT=xxc*hh<cr><lf></code>
Parameters	<filter length> 00S to 50S, 01M to 10M	Time constant of the acoustic temperature filter in seconds (S) or minutes (M). Supported settings are 00S (to disable the filter), then ten second increments from 10S up to 50S, then one minute increments from 01M to 10M. (Factory Default Setting = 01M)
Examples	<u>Example 1</u> Set the filter time constant to 40 seconds. Verify that the command has been accepted.	
	<u>Message</u> <code>\$01,ATF40S*//<cr><lf></code> <code>\$01,AT?F*//<cr><lf></code> <code>\$WI,AT=40S*4D<cr><lf></code>	<u>Comment</u> Set 40s time constant Query filter time constant Sensor response
	<u>Example 2</u> Set the filter time constant to 2 minutes. Verify that the command has been accepted.	
	<u>Message</u> <code>\$01,ATF02M*//<cr><lf></code> <code>\$01,AT?F*//<cr><lf></code> <code>\$WI,AT=02M*55<cr><lf></code>	<u>Comment</u> Set 2 minute time constant Query filter time constant Sensor response
Description	Use this command to alter the time constant of the acoustic temperature filter. The filter can be disabled by setting the time constant to zero (00S). Valid time constants are ten second steps from 10S to 50S and then one minute steps from 01M to 10M.	

8.3.5 BM: Set or Query the Barometric Pressure

Command Parameter	BM	
Command Syntax	SET Sensor:	<code>\$<listenerID>,BM<mode>*<checksum><cr><lf></code> <code>\$aa,BMm*hh<cr><lf></code>
	QUERY Sensor:	<code>\$<listenerID>,BM?*<checksum><cr><lf></code> <code>\$aa,BM?*hh<cr><lf></code>
	Sensor output:	<code>\$<talkerID>,DF=<mode>,<status>,<barometric pressure>,<barometer temperature>*<checksum><cr><lf></code> <code>\$aa,BM=m,s,pppp.pppp,ttt.tt*hh<cr><lf></code>
Parameters	<mode>	
	D	Disabled
	E	Enabled
	C	Calibration
	<status>	
	D	Disabled
	V	Valid
	<barometric pressure>	
	660.0000 to 1260.0000	Measured Barometric Pressure in hPa
	<barometer temperature>	
	-20.00 to +60.00	Measured Temperature of the Barometer in °C
Examples	<u>Example 1</u>	
	Enable the Barometer, then query the barometer. The example shows the sensor output with a pressure of 1013.2500 hPa and a barometer temperature of 25.00 °C	
	<u>Message</u>	<u>Comment</u>
	<code>\$01,BME*//<cr><lf></code>	Enabled Barometer
	<code>\$01,BM?*//<cr><lf></code>	Query Barometer
	<code>\$WI,BM=E,V,1013.2500,+25.00*17<cr><lf></code>	Sensor response

8.3.6 BR: Set or Query the Serial Interface Baud Rate

Command Parameter	BR													
Command Syntax	SET Sensor:	$\$ \langle listenerID \rangle, BR \langle baudrate \rangle * \langle checksum \rangle \langle cr \rangle \langle lf \rangle$ $\$ aa, BRx * hh \langle cr \rangle \langle lf \rangle$												
	QUERY Sensor:	$\$ \langle listenerID \rangle, BR ? * \langle checksum \rangle \langle cr \rangle \langle lf \rangle$ $\$ aa, BR ? * hh \langle cr \rangle \langle lf \rangle$												
	Sensor Output:	$\$ \langle talkerID \rangle, BR = \langle baudrate \rangle * \langle checksum \rangle \langle cr \rangle \langle lf \rangle$ $\$ aa, BR = x * hh \langle cr \rangle \langle lf \rangle$												
Parameters	<baudrate> <table><tr><td>0</td><td>Set the baud rate to 38400 baud</td></tr><tr><td>1</td><td>Set the baud rate to 19200 baud</td></tr><tr><td>2</td><td>Set the baud rate to 9600 baud (Factory Default Setting)</td></tr><tr><td>3</td><td>Set the baud rate to 4800 baud</td></tr><tr><td>4</td><td>Set the baud rate to 2400 baud</td></tr><tr><td>5</td><td>Set the baud rate to 1200 baud</td></tr></table>		0	Set the baud rate to 38400 baud	1	Set the baud rate to 19200 baud	2	Set the baud rate to 9600 baud (Factory Default Setting)	3	Set the baud rate to 4800 baud	4	Set the baud rate to 2400 baud	5	Set the baud rate to 1200 baud
0	Set the baud rate to 38400 baud													
1	Set the baud rate to 19200 baud													
2	Set the baud rate to 9600 baud (Factory Default Setting)													
3	Set the baud rate to 4800 baud													
4	Set the baud rate to 2400 baud													
5	Set the baud rate to 1200 baud													
Examples	<u>Example 1</u> Set the baud rate to 19200 baud, verify the new setting and send a user reset command to activate the new baud rate <table><tr><td><u>Message</u></td><td><u>Comment</u></td></tr><tr><td>$\\$ 01, BR1 * // \langle cr \rangle \langle lf \rangle$</td><td>Set baud rate to 19200</td></tr><tr><td>$\\$ 01, BR ? * // \langle cr \rangle \langle lf \rangle$</td><td>Query baud rate setting</td></tr><tr><td>$\\$ WI, BR = 1 * 2E \langle cr \rangle \langle lf \rangle$</td><td>Sensor Output</td></tr><tr><td>$\\$ 01, RSU * // \langle cr \rangle \langle lf \rangle$</td><td>Send user reset</td></tr></table>		<u>Message</u>	<u>Comment</u>	$\$ 01, BR1 * // \langle cr \rangle \langle lf \rangle$	Set baud rate to 19200	$\$ 01, BR ? * // \langle cr \rangle \langle lf \rangle$	Query baud rate setting	$\$ WI, BR = 1 * 2E \langle cr \rangle \langle lf \rangle$	Sensor Output	$\$ 01, RSU * // \langle cr \rangle \langle lf \rangle$	Send user reset		
<u>Message</u>	<u>Comment</u>													
$\$ 01, BR1 * // \langle cr \rangle \langle lf \rangle$	Set baud rate to 19200													
$\$ 01, BR ? * // \langle cr \rangle \langle lf \rangle$	Query baud rate setting													
$\$ WI, BR = 1 * 2E \langle cr \rangle \langle lf \rangle$	Sensor Output													
$\$ 01, RSU * // \langle cr \rangle \langle lf \rangle$	Send user reset													
Description	Use the BR command to change the sensor serial interface baud rate. The new baud rate setting will only come into effect when the sensor is next powered-up or after a Reset command (RSU) has been received. If the baud rate is changed, you will only be able to communicate with the sensor if the host computer's baud rate is set to the same baud rate. If you do not know what the current setting of the sensor baud rate is you will need to try each baud rate in turn until you establish communication.													

8.3.7 CF: Set or Query the Wind Compass Settings

Command Parameter	CF(heading)	
Command Syntax	SET Sensor:	<pre> \$<listenerID>,CF<mode>*<checksum><cr><lf> \$aa,CFc*hh<cr><lf> \$<listenerID>,CF<heading>*<checksum><cr><lf> \$aa,CFxxx.x*hh<cr><lf> </pre>
	QUERY Sensor:	<pre> \$<listenerID>,CF?*<checksum><cr><lf> \$aa,CF?*hh<cr><lf> </pre>
	FT602-D-SM output:	<pre> \$<talkerID>,CF=<mode>,<status>,<heading>,<dec>*<checksum><cr><lf> \$aa,CF=C,C,xxx.x,xxx.x*hh<cr><lf> </pre>
Parameters	<heading>	000.0 to 359.9 Compass heading in degrees
	<mode>	C Enable calibration mode D Disable compass function E Enable compass function S Enable Standalone Compass function MS Save compass calibration
	<status>	D Compass module either disabled or not present V Valid compass reading
	<dec>	000.0 to 359.9 Declination angle. Standard convention requires that westward declination angles are subtracted from magnetic heading in order to derive true heading. Conversely eastward declination angles are added to magnetic heading. However, since the declination angle on the FT602-D-SM is provided in the form 0 to 359.9°, the user must write westerly declination in the form (360 – westerly declination). For example, a westerly declination of 10° would be sent to the FT602-D-SM as 350°.
Examples	Example 1 Set the declination angle to 5° East and read the compass heading and declination parameters:	
	Message <pre> \$01,CF005.0*//<cr><lf> \$01,CF?*//<cr><lf> \$WI,CF=D,D,005.0,005.0*26<cr><lf> </pre>	Comment Set declination angle to 5° Query compass parameters FT602-D-SM output
	Example 2 The datum direction of the FT602-D-SM is pointing 40° to the East – set the declination angle a further 5° East and read the compass heading and declination parameters:	
	Message <pre> \$01,CF005.0*//<cr><lf> \$01,CF?*//<cr><lf> \$WI,CF=E,V,045.0,005.0*31<cr><lf> </pre>	Comment Set declination angle to 5° Query parameters FT602-D-SM output
	Example 3 Enable, query and disable the compass:	
	Message <pre> \$01,CFE*//<cr><lf> \$01,CF?*//<cr><lf> \$WI,CF=E,V,000.0,000.0*35<cr><lf> </pre>	Enable compass Query compass setting FT602-D-SM output (enabled) Disable compass

	\$01,CFD*//<cr><lf>	
	Example 4 Enable the calibration procedure and save the results \$01,CFC*//<cr><lf> \$01,CFE*//<cr><lf> \$01,CFMS*//<cr><lf>	Enable compass calibration Disable compass calibration Save compass calibration
Description	Use the CF command to access the compass related features of the FT602-D-SM. See Section 3.4 for further details about the compass, declination angle and the calibration procedure.	

8.3.8 CF.2: Set or Query the Tilt Measurements

Command Parameter	CF(tilt)	
Command Syntax	SET Sensor:	<code>\$<listenerID>,CFT<mode>*<checksum><cr><lf> \$aa,CFTm*hh<cr><lf></code>
	QUERY Sensor:	<code>\$<listenerID>,CF?T*<checksum><cr><lf> \$aa,CF?T*hh<cr><lf></code>
	FT602-D-SM output:	<code>\$<talkerID>,CF=<status>,<pitch>,<roll>*<checksum><cr><lf> \$aa,CF=c,xxxx.x,xxxx.x*hh<cr><lf></code>
Parameters	<pitch> +040.0 to -040.0	Pitch of the sensor in degrees
	<roll> +040.0 to -040.0	Roll of the sensor in degrees
	<mode> TCS TCF	Start tilt zero calibration Finish tilt zero calibration
	<status> D V I	Compass module disabled Valid inclinometer reading Invalid inclinometer reading
Examples	<u>Example 1</u> CF?T query and response. Pitch is +20.4° and roll is -13.2° parameters:	
	<u>Message</u> \$01,CF?T*//<cr><lf> \$WI,CF=V,+020.4,-013.2*26<cr><lf>	<u>Comment</u> Query tilt data FT602-D-SM output
	<u>Example 2</u> Start tilt zero calibration and finish calibration.	
	<u>Message</u> \$01,CFTCS*//<cr><lf> \$01,CFTCF*//<cr><lf>	<u>Comment</u> Start tilt calibration Finish tilt calibration
Description	The CF?T command will only provide valid data when the compass module is enabled. See Section 7.3.7 for details on how to enable the compass.	
	For further guidance on performing tilt zero calibration see Inclinometer Zeroing.	

8.3.9 CI: Set or Query the Communication Interface (RS485, RS422, UART)

Command Parameter	CI	
Command Syntax	SET Sensor:	$\$ \langle \text{listenerID} \rangle, CI \langle \text{interface} \rangle * \langle \text{checksum} \rangle \langle \text{cr} \rangle \langle \text{lf} \rangle$ $\$aa, CIc * hh \langle \text{cr} \rangle \langle \text{lf} \rangle$ or $\$aa, CIcc * hh \langle \text{cr} \rangle \langle \text{lf} \rangle$
	QUERY Sensor:	$\$ \langle \text{listenerID} \rangle, CI ? * \langle \text{checksum} \rangle \langle \text{cr} \rangle \langle \text{lf} \rangle$ $\$aa, CI ? * hh \langle \text{cr} \rangle \langle \text{lf} \rangle$
	Sensor output:	$\$ \langle \text{talkerID} \rangle, CI = \langle \text{interface string} \rangle * \langle \text{checksum} \rangle \langle \text{cr} \rangle \langle \text{lf} \rangle$ $\$aa, CI = ccccc * hh \langle \text{cr} \rangle \langle \text{lf} \rangle$
	QUERY RS422 Transmitter Mode:	$\$ \langle \text{listenerID} \rangle, CI ? 2 * \langle \text{checksum} \rangle \langle \text{cr} \rangle \langle \text{lf} \rangle$ $\$aa, CI ? 2 * hh \langle \text{cr} \rangle \langle \text{lf} \rangle$
	Sensor output:	$\$ \langle \text{talkerID} \rangle, CI = \langle \text{RS422 tx mode} \rangle * \langle \text{checksum} \rangle \langle \text{cr} \rangle \langle \text{lf} \rangle$ $\$aa, CI = c * hh \langle \text{cr} \rangle \langle \text{lf} \rangle$
Parameters	$\langle \text{interface} \rangle$	
	8	Set the RS485 interface (Factory Default Setting)
	2A	Set the RS422 interface with the transmitter always enabled.
	2I	Set the RS422 interface with the transmitter disabled when not transmitting.
	U	Set the UART communication interface
	$\langle \text{interface string} \rangle$	
	RS485	The RS485 Interface is enabled.
	RS422	The RS422 Interface is enabled.
	UART	The UART Interface is enabled.
	$\langle \text{RS422 tx mode} \rangle$	
	A	The RS422 transmitter is always enabled.
	I	The RS422 transmitter is disabled when not transmitting.
Examples	Example 1 Enable the RS485 interface. Verify that the command has been accepted.	
	Message $\$01, CI8 * // \langle \text{cr} \rangle \langle \text{lf} \rangle$ $\$01, CI ? * // \langle \text{cr} \rangle \langle \text{lf} \rangle$ $\$WI, CI = RS485 * 3D \langle \text{cr} \rangle \langle \text{lf} \rangle$	Comment Enable RS485 Query communications interface Sensor response
	Example 2 Enable the RS422 interface with the transmitter always enabled. Verify that the interface and transmitter settings.	
	Message $\$01, CI2A * // \langle \text{cr} \rangle \langle \text{lf} \rangle$ $\$01, CI ? * // \langle \text{cr} \rangle \langle \text{lf} \rangle$ $\$WI, CI = RS422 * 30 \langle \text{cr} \rangle \langle \text{lf} \rangle$ $\$01, CI ? 2 * // \langle \text{cr} \rangle \langle \text{lf} \rangle$ $\$WI, CI = A * 44 \langle \text{cr} \rangle \langle \text{lf} \rangle$	Comment Enable RS422 with transmitter on Query communications interface Sensor response Query RS422 transmitter mode Sensor response
Description	Use the CI command to switch between communications interface modes. Only one communications interface can be used at a time, however the sensor is able to receive host commands through any of the three supported interface modes. When selecting the RS422 interface, the sensor's RS422 transmitter may be set to remain permanently enabled (or only enabled during message transmission).	

8.3.10 CIC: Set or Query the CANbus Interface

Command Parameter	CIC	
Command Syntax	SET Sensor:	<i>\$<listenerID>,CIC<CANBus>*<checksum><cr><lf> \$aa,CICc*hh<cr><lf></i>
	QUERY Sensor:	<i>\$<listenerID>,CI?C*<checksum><cr><lf> \$aa,CI?C*hh<cr><lf></i>
	Sensor output:	<i>\$<talkerID>,CI=<CANBus>,<CAN Status>*<checksum><cr><lf> \$aa,CI=x,x*hh<cr><lf></i>
Parameters	<CANBus> E D	Enabled Disabled
	<CAN Status> E D	Enabled. Disabled
Examples	Example 1 Enable the CAN interface. Verify that the command has been accepted.	
	Message <i>\$01,CICE*//<cr><lf></i> <i>\$01,CI?C*//<cr><lf></i> <i>\$WI,CI=E,E*3D<cr><lf></i>	Comment Enable CANBus Query CAN interface Sensor response
Description	Use the CIC command to enable or disable the CANBus between communications interface. When enabled the sensor will begin outputting all enabled NMEA 2000 PGNs at the rates defined in Section 7. When the CIC query is sent to the sensor, the sensor should always respond with matching responses in the <CANBus> and <CAN Status> fields. If the responses to these fields don't match this indicates a fault with the sensor.	

8.3.11 CICIP: Set or Query the NMEA PGNs

Command Parameter	CICIP	
Command Syntax	SET Sensor:	<i>\$<listenerID>,CICIP<PGN><status>*<checksum><cr><lf> \$aa,CICPcc*hh<cr><lf></i>
	QUERY Sensor:	<i>\$<listenerID>,CI?CP<PGN>*<checksum><cr><lf> \$aa,CI?CPc*hh<cr><lf></i>
	Sensor output:	<i>\$<talkerID>,CI=<PGN>,<status>*<checksum><cr><lf> \$aa,CI=x,x*hh<cr><lf></i>
Parameters	<PGN> 1 PGN 130306 (Wind Data) 2 PGN 127250 (Vessel Heading) 3 PGN 127257 (Attitude) 4 PGN 130314 (Atmospheric Pressure) 5 PGN 130316 (Temperature) 6 PGN 65305 (WV Status Proprietary PGN)	
	<status> E Enabled. D Disabled	
Examples	Example 1 Enable the Wind Data PGN interface. Verify that the command has been accepted.	
	Message <i>\$01,CICP1E*//<cr><lf></i> <i>\$01,CI?CP1*//<cr><lf></i> <i>\$WI,CI=1,E*3D<cr><lf></i>	Comment Enable Wind Data PGN Query Wind Data PGN Status Sensor response
Description	Use the CICIP command to enable or disable the configurable PGNs available in the FT602. When the CANBus interface is enabled all enabled PGNs will be transmitted at the update rate defined in Section 7.	

8.3.12 CU: Set or Query the Continuous Update Setting

Command Parameter	CU
Command Syntax	SET Sensor: <pre>\$<listenerID>,CU<cont.update>,<interval>*<checksum> <cr> <lf> \$aa,CUCxxxxx*hh<cr><lf></pre> QUERY Sensor: <pre>\$<listenerID>,CU?*<checksum><cr><lf> \$aa,CU?*hh<cr><lf></pre> Sensor Output: <pre>\$<talkerID>,CU=<cont.update>,<interval>*<checksum><cr> <lf> \$aa,CU=c,xxxxx*hh<cr><lf></pre>

WARNING: Do not use the continuous update mode if there are other talkers connected to the data bus. Only one active talker is allowed on the data bus at any one time otherwise bus contention will occur

8.3.13 DF: Set or Query the Wind Velocity Data Format

Command Parameter	DF	
Command Syntax	SET Sensor:	<code>\$<listenerID>,DF<format>*<checksum><cr><lf></code> <code>\$aa,DFc*hh<cr><lf></code> or <code>\$aa,DFcc*hh<cr><lf></code>
	QUERY Sensor:	<code>\$<listenerID>,DF?*<checksum><cr><lf></code> <code>\$aa,DF?*hh<cr><lf></code>
	Sensor output:	<code>\$<talkerID>,DF=<format>*<checksum><cr><lf></code> <code>\$aa,DF=c*hh<cr><lf></code>
Parameters	<format>	
	P	Set the data format to Polar (wind speed and direction) (Factory Default Setting)
	N	Set the data format to NMEA 0183 with wind speed in m/s
	NN	Set the data format to NMEA 0183 with wind speed in knots
	NK	Set the data format to NMEA 0183 with wind speed in km/h
	C	Set the data format to Combined (Polar data & acoustic temperature)
	T	Set the data format to T Format (Polar data, temperature, barometric pressure and magnetometer)
X	Set the data format to X Format (Polar data, temperature and barometric pressure)	
Examples	Example 1	
	Set the wind velocity output data format to NMEA with wind speed in m/s and verify the new setting.	
	Message	Comment
	<code>\$01,DFN*//<cr><lf></code>	Set format to NMEA (m/s)
	<code>\$01,DF?*//<cr><lf></code>	Query format setting
	<code>\$WI,DF=N*43<cr><lf></code>	Sensor response
	Example 2	
	Set the wind velocity output data format to NMEA with wind speed in knots and verify the new setting.	
	Message	Comment
	<code>\$01,DFNN*//<cr><lf></code>	Set format to NMEA (knots)
	<code>\$01,DF?*//<cr><lf></code>	Query format setting
	<code>\$WI,DF=NN*0D<cr><lf></code>	Sensor response
	Example 3	
	Set the wind velocity output data format to 'Combined'	
	Message	Comment
<code>\$01,DFC*//<cr><lf></code>	Set format to Combined	
<code>\$01,DF?*//<cr><lf></code>	Query format setting	
<code>\$WI,DF=C*4E<cr><lf></code>	Sensor Response = C	
Example 4		
WV? Command with DFC enabled		
Message	Comment	
<code>\$01,WV?*//</code>	Query WV wind data	
<code>\$WI,WVC=000.0,323,0,+026.3,C,A*4D</code>	Sensor Response	
Example 5		
CU Mode example with DFC enabled		
Message	Comment	
<code>\$01,CUE00010*//</code>	Enable CU mode (1Hz)	
<code>\$WI,WVC=000.0,333,0,+026.4,C,A*4B</code>	CU DFC Response (1Hz)	

Description	Use the DF command to set the required format of the wind velocity readings. See command WV (Sections 8.3.22 and 8.3.23) for a description of the sensor output for each of the format types. When a DF Set command is sent to the sensor, a reset of the minimum and maximum readings to their default values is automatically performed. Polar Format: The sensor returns the magnitude of the wind speed (m/s only) and the wind direction (0-359 degrees). NMEA 0183 Format: The sensor returns the wind angle (0-359 degrees, Relative) and wind speed (m/s, knots or km/h). The sensor TalkerID is always set to WI when NMEA format is selected irrespective of any value that may have been set with the ID command. Combined Format: The sensor returns the polar wind data (m/s) alongside the acoustic air temperature. T Format: The sensor returns polar wind data (m/s), acoustic temperature, barometric pressure and magnetometer data.
-------------	---

8.3.14 DL: Set or Query the Command Delay Interval

Command Parameter	DL	
Command Syntax	SET Sensor:	<code>\$<listenerID>,DL<delay>*<checksum><cr><lf></code> <code>\$aa,DLxx*hh<cr><lf></code>
	QUERY Sensor:	<code>\$<listenerID>,DL?*<checksum><cr><lf></code> <code>\$aa,DL?*hh<cr><lf></code>
	Sensor output:	<code>\$<talkerID>,DL=<delay>*<checksum><cr><lf></code> <code>\$aa,DL=xx*hh<cr><lf></code>
Parameters	<delay> 00 to 20 (delay interval, in 50ms increments) (Factory Default Setting = 01)	
Examples	<u>Example 1</u> Set the command delay interval to 250ms and verify the new setting.	
	<u>Message</u> \$01,DL05*//<cr><lf> \$01,DL?*//<cr><lf> \$WI,DL=05*02<cr><lf>	<u>Comment</u> Set delay to 250ms Query delay setting Sensor response
Description	Use the DL command to set the delay interval from when the sensor receives a command to when the command is executed. The DL command is primarily intended for use where a time delay may be required to allow the RS422/RS485 interface to switch from transmit to receive mode.	
	For example, if the delay interval is set to 250ms then the sensor will commence outputting the wind velocity data between 250-300ms after receiving a WV query command. If any further commands are sent to the sensor before the delay interval has elapsed they will be discarded.	

8.3.15 ER: Query or Reset the Error Report

Command Parameter	ER		
Command Syntax	SET Sensor:	$\$<listenerID>,ER<reset>*<checksum><cr><lf>$ $\$aa,ERc*hh<cr><lf>$	
	QUERY Sensor:	$\$<listenerID>,ER?*<checksum><cr><lf>$ $\$aa,ER?*hh<cr><lf>$	
	Sensor output:	$\$<talkerID>,ER=<error\ report>*<checksum><cr><lf>$ $\$aa,ER=xxxxxxxxxxxxxxxx*hh<cr><lf>$	
Parameters	<reset> <i>R</i>	Resets the historical log section of the error report to all 0's	
	<error report> <i><error report></i>	Sensor error report string	
Examples	<u>Example 1</u> Query the error report		
	<u>Message</u> $\$01,ER?*//<cr><lf>$ $\$WI,ER=0000000000000000*28<cr><lf>$	<u>Comment</u> Query error report Sensor response	
Description	The error report contains information on errors that have occurred during the operation of the sensor. The output string is always comprised of 15 ASCII characters (all shown as '0', ASCII 30(HEX) in the above example).		
	01234567891011121314 ↑ Most recent Least recent Current Status Error Report Locations The first character in the data field represents the current operational status of the sensor. '0' (ASCII 30(HEX)) indicates that the sensor is functioning correctly any other character indicates that an error condition exists. The status is cleared once the ER command is executed. The next 14 locations contain an historical log of the last 14 errors with the most recent error being recorded in the leftmost position. Each error condition is assigned an ASCII character. The historical log is stored in Flash and is retained when the power is switched off or the sensor software reset. This report can be sent back to the FT factory for analysis if there are problems with the sensor.		

Currently the historical error log is only used for factory diagnostic purposes.



8.3.16 FL.1: Set or Query General Filter Settings

Command Parameter	FL (enable/ disable)									
Command Syntax	SET Sensor:	<code>\$<listenerID>,FL<filter>*<checksum><cr><lf></code> <code>\$aa,FLc*hh<cr><lf></code>								
	QUERY Sensor:	<code>\$<listenerID>,FL?*<checksum><cr><lf></code> <code>\$aa,FL?*hh<cr><lf></code>								
	Sensor output:	<code>\$<talkerID>,FL=<filter>*<checksum><cr><lf></code> <code>\$aa,FL=c*hh<cr><lf></code>								
Parameters	<filter> E Filter enabled (Factory Default Setting) D Filter disabled									
Examples	Example 1 Enable the filter. Verify that the command has been accepted. <table><tr><td>Message</td><td>Comment</td></tr><tr><td><code>\$01,FLE*//<cr><lf></code></td><td>Enable filtering</td></tr><tr><td><code>\$01,FL?*//<cr><lf></code></td><td>Query filter setting</td></tr><tr><td><code>\$WI,FL=E*40<cr><lf></code></td><td>Sensor response</td></tr></table>		Message	Comment	<code>\$01,FLE*//<cr><lf></code>	Enable filtering	<code>\$01,FL?*//<cr><lf></code>	Query filter setting	<code>\$WI,FL=E*40<cr><lf></code>	Sensor response
	Message	Comment								
<code>\$01,FLE*//<cr><lf></code>	Enable filtering									
<code>\$01,FL?*//<cr><lf></code>	Query filter setting									
<code>\$WI,FL=E*40<cr><lf></code>	Sensor response									
	Example 2 Disable the filter. Verify that the command has been accepted. <table><tr><td>Message</td><td>Comment</td></tr><tr><td><code>\$01,FLD*//<cr><lf></code></td><td>Disable filtering</td></tr><tr><td><code>\$01,FL?*//<cr><lf></code></td><td>Query filter setting</td></tr><tr><td><code>\$WI,FL=D*41<cr><lf></code></td><td>Sensor response</td></tr></table>		Message	Comment	<code>\$01,FLD*//<cr><lf></code>	Disable filtering	<code>\$01,FL?*//<cr><lf></code>	Query filter setting	<code>\$WI,FL=D*41<cr><lf></code>	Sensor response
Message	Comment									
<code>\$01,FLD*//<cr><lf></code>	Disable filtering									
<code>\$01,FL?*//<cr><lf></code>	Query filter setting									
<code>\$WI,FL=D*41<cr><lf></code>	Sensor response									
Description	Use the FL command to enable or disable moving average filtering of the wind speed and wind direction readings (see Section 3.3).									

8.3.17 FL.2: Set or Query Filter Lengths

Command Parameter	FL (lengths)	
Command Syntax	SET Sensor:	<pre>\$<listenerID>,FLL<speedLen>,<dirLen>*<checksum><cr> <lf> \$aa,FLLxxxx,xxxx*hh<cr><lf></pre>
	QUERY Sensor:	<pre>\$<listenerID>,FL?L*<checksum><cr><lf> \$aa,FL?L*hh<cr><lf></pre>
	Sensor output:	<pre>\$<talkerID>,FL=<speedLen>,<dirLen>*<checksum><cr><lf> f> \$aa,FL=xxxx,xxxx*hh<cr><lf></pre>
Parameters	<speedLen> 0001 to 0064	Sample size of the wind speed filter. Number of previous readings# used to calculate the latest wind speed reading, 0001 is equivalent to disabling the filter (Factory Default Setting = 0016)
	<dirLen> 0001 to 0064	Sample size of the wind direction filter. Number of previous readings# used to calculate the latest wind direction reading, 0001 is equivalent to disabling the filter; (Factory Default Setting = 0016)
Examples	<u>Example 3</u> Modify the filter's dimensions. Verify that the command has been accepted.	
	<u>Message</u> \$01,FLL0001,0032*//<cr><lf> \$01,FL?L*//<cr><lf> \$WI,FL=0001,0032*29<cr><lf>	<u>Comment</u> Set speed filter length to 1 and direction filter length to 32. Query filter's length settings. Sensor response.
Description	Use this command to modify the speed and direction filter lengths. When the filter is enabled, speed and direction readings are independently averaged by calculating the mean of the previous number of readings# set by the filter lengths <speedLen> and <dirLen>.	
	Please note the previous readings filter length by definition includes the most up to date wind reading at position 1. So setting a filter to a length of 0001 gives just the current reading (see Section 3.3). The sensor's internal memory is large enough to retain 64 previous speed and direction readings, allowing for maximum filter length of 6.4 seconds.	

8.3.18 ID: Set or Query the Listener & Talker Identifiers

Command Parameter	ID								
Command Syntax	SET Sensor: $\\$<listenerID>, ID<RxID><TxID>*<checksum><cr><lf>$ $\\$aa, ID=cccc*hh<cr><lf>$ QUERY Sensor: $\\$<listenerID>, ID?*<checksum><cr><lf>$ $\\$aa, ID?*hh<cr><lf>$ Sensor output: $\\$<talkerID>, ID=<RxID><TxID>*<checksum><cr><lf>$ $\\$aa, ID=cccc*hh<cr><lf>$								
Parameters	<RxID> 00 to ZZ The sensor 2 digit listener address identifier (Factory Default RxID = 01) <TxID> 00 to ZZ The sensor 2 digit talker address identifier (Factory Default TxID = W1)								
Examples	<u>Example 1</u> Set the sensor listener address identifier to A1 and the talker address identifier to B1. Verify that the command has been accepted. <table> <tr> <td><u>Message</u></td><td><u>Comment</u></td></tr> <tr> <td>$\\$01, IDA1B1*//<cr><lf>$</td><td>Set address ID's</td></tr> <tr> <td>$\\$A1, ID?*//<cr><lf>$</td><td>Query ID settings</td></tr> <tr> <td>$\\$B1, ID=A1B1*6C<cr><lf>$</td><td>Sensor response</td></tr> </table> Note: the ID? command must use the new listener ID otherwise the command will not be recognised.	<u>Message</u>	<u>Comment</u>	$\$01, IDA1B1*//<cr><lf>$	Set address ID's	$\$A1, ID?*//<cr><lf>$	Query ID settings	$\$B1, ID=A1B1*6C<cr><lf>$	Sensor response
<u>Message</u>	<u>Comment</u>								
$\$01, IDA1B1*//<cr><lf>$	Set address ID's								
$\$A1, ID?*//<cr><lf>$	Query ID settings								
$\$B1, ID=A1B1*6C<cr><lf>$	Sensor response								
Description	Use the ID command to set the listener and talker address identifiers. See Section 6.5 for information on listener and talker address identifiers. From software V7.5.1 the ID command can also be used to modify the Talker ID for the NMEA 0183 data format (see Section 8.3.23). Please note that the NMEA 0183 data standard format does not officially support alternative talker IDs and may result in failure to communicate or interpret data.								

8.3.19 MM: Reset or Query the Min/Max Recorded Wind Speed

Command Parameter	MM		
Command Syntax	SET Sensor:	<code>\$<listenerID>,MM<setting>*<checksum><cr><lf></code> <code>\$aa,MMc*hh<cr><lf></code>	
	QUERY Sensor:	<code>\$<listenerID>,MM?*<checksum><cr><lf></code> <code>\$aa,MM?*hh<cr><lf></code>	
	Sensor output:	<code>\$<talkerID>,MM=<MinSpeed>,<MaxSpeed>*<checksum><cr></code> <code><lf></code> <code>\$aa,MM=xxx.x,xxx.x*hh<cr><lf></code>	
Parameters	<setting>	R Resets the min/max readings to their default (<MinSpeed> to 999.9 and <MaxSpeed> to 000.0) until the first reading	
	<MinSpeed>	000.0 to 999.9	Minimum detected wind speed in current unit (m/s, knots or km/h)
	<MaxSpeed>	000.0 to 999.9	Maximum detected wind speed in current unit (m/s, knots or km/h)
Examples	<u>Example 1</u> Query the min/max wind speed readings		
	<u>Message</u>		<u>Comment</u>
	<code>\$01,MM?*/<cr><lf></code> <code>\$WI,MM=005.1,034.2*22<cr><lf></code>		Query the min/max readings Sensor response
Description	Use the MM command to query the minimum and maximum wind speed readings that the sensor has recorded since it was last switched on. The minimum and maximum readings are set to their default values when an MMR, an RS or a DF set command is executed.		

8.3.20 SN: Query the Serial Number and Platform Version

Command Parameter	SN
Command Syntax	SET Sensor: NA
	QUERY Sensor: <code>\$<listenerID>,SN?*<checksum><cr><lf></code> <code>\$aa,SN?*hh<cr><lf></code>
	Sensor Output: <code>\$<talkerID>,SN=<SerialNumber>,<PlatformVersion>*<checksum><cr><lf></code> <code>\$aa,SN=xxxxxx-xxx,xxsss*hh<cr><lf></code>
Parameters	<SerialNumber> 00000-000 to 99999-999 Unique serial number of the sensor
	<PlatformVersion> 00-99 Platform version (issue) of the sensor design. The 3 spaces after the 2 digit number are reserved for future use.
Examples	<u>Example 1</u> Read the sensor serial number and platform version <u>Message</u> <code>\$01,SN?*/<cr><lf></code> <code>\$WI,SN=09000-130,24 *3E<cr><lf></code> <u>Comment</u> Query serial number Sensor response
Description	The SN command returns the serial number of the sensor and also the platform version of the sensor. The serial number format starts with a 5-digit batch code, followed by a 3 digit number which identifies a sensor within a particular batch. The overall number is the unique serial number identifier for the sensor.

8.3.21 SV: Query the Software Version

Command Parameter	SV		
Command Syntax	SET Sensor:	NA	
	QUERY Sensor:	\$<listenerID>,SV?*<checksum><cr><lf> \$aa,SV?*hh<cr><lf>	
	Sensor Output:	\$<talkerID>,SV=<SoftwareVersion>*<checksum><cr><lf> \$aa,SV=xx.xx.xs*hh<cr><lf>	
Parameters	<SoftwareVersion> 01.00.0s to Software version of the sensor, where s is a space. 99.99.9s		
Examples	<u>Example 1</u>		
	Read the software version number		
	<u>Message</u>		<u>Comment</u>
	\$01,SV?*17<cr><lf>		Query software version
	\$WI,SV=07.13.0 *16<cr><lf>		Sensor response 7.13.0 format
Description	The SV command returns the software version of the sensor.		
	There is one blank space following software version.		

8.3.22 WV Polar: Query the Wind Velocity Reading

Command Parameter	WV (Polar)
-------------------	------------

Command Syntax	SET Sensor:	N/A
	QUERY Sensor:	<code>\$<listenerID>,WV?*<checksum><cr><lf></code> <code>\$aa,WV?*hh<cr><lf></code>
	Sensor output:	<code>\$<talkerID>,WVP=<speed>,<angle>,<status>*<checksum></code> <code><cr><lf></code> <code>\$aa,WVP=xxx.x,yyy,z*hh<cr><lf></code>
		For DFC Mode: <code>\$<talkerID>,WVP=<speed>,<angle>,<status>,<temperature>,<temp.units>,<temp.status>*<checksum><cr><lf></code> <code>\$aa,WVC=www.w,xxx,y,zzzz.z,a,b*hh<cr><lf></code> For DFT Mode: <code>\$<talkerID>,WVT=<speed>,<angle>,<status>,<temperature>,<temp.units>,<temp.status>,<magnetometer output>,<magnetometer temp>,<pressure>*<checksum><cr><lf></code> <code>\$aa,WVT=www.w,xxx,y,zzzz.z,a,b,mxmxmymymzmt,mtmt,pp pp.pp*hh<cr><lf></code> For DFX Mode: <code>\$<talkerID>,WVT=<speed>,<angle>,<status>,<temperature>,<temp.units>,<temp.status>,<pressure>,<barometer status>*<checksum><cr><lf></code> <code>\$aa,WVT=www.w,xxx,y,zzzz.z,a,b,pppp.pppp,b*hh<cr><lf></code>

Parameters	<speed> 000.0 to 075.0	Measured wind speed in metres per second
	<angle> 000 to 359	Measured wind direction in degrees relative to sensor datum
	<status> 0 to Z	Indicates whether an error condition was detected by the sensor operating system. A status value of 0 indicates no issues have been detected (ASCII 30(HEX)).
		If the sensor detects an error condition, the status character will be set according to section 3.5.1
	<Temperature> -020.0 to +060	-20°C to +60°C
	<Temp Units> C	Degrees Celsius
	<Temp Status> A or V	A (Valid temperature data) or V (Acquiring data – not yet valid)
	<Magnetometer Output> ASCII Coded Hex Values	ASCII coded hex values for magnetometer output in X, Y and Z axes.

<Magnetometer Temp> <i>ASCII Coded Hex Value</i>	ASCII coded hex value for magnetometer temperature.
<Pressure> <i>660.00 to 1260.00</i>	Measured Barometric Pressure in hPa
<Barometer Status> <i>D or V</i>	D (disabled) or V (valid)

Examples	<u>Example 1</u> The following example illustrates the DFP (Polar) wind velocity data format. The example shows the sensor output with a wind speed of 20m/s and a wind angle of 45°.	
	<u>Message</u> \$01,WV?*13<cr><lf> \$WI,WVP=020.0,045,0*73<cr><lf>	<u>Comment</u> Query the wind velocity Sensor response (DFP)
	<u>Example 2</u> The following examples illustrates the DFC (Combined) wind velocity data format. The example shows the sensor output with a wind speed of 0m/s, a wind angle of 323° and acoustic temperature of +26.3°C.	
	<u>Message</u> \$01,WV?*13<cr><lf> \$WI,WVC=000.0,323,0,+026.3,C,A*4D	<u>Comment</u> Query the wind velocity Sensor Response (DFC)
	<u>Example 3</u> The following example illustrates the DFT wind velocity data. The example shows the sensor output with a wind speed of 20m/s and a wind angle of 45°.	
	<u>Message</u> \$01,WV?*//<cr><lf> \$//,WVT=020.0,045,0,+025.4,C,A,FF190088FF2E,17C0,1020.23*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>1020.2300*<checksum>	

8.3.23 WV NMEA: Query the Wind Velocity Reading

Command Parameter	WV (NMEA)	
Command Syntax	SET Sensor:	N/A
	QUERY Sensor:	<code>\$<listenerID>,WV?*<checksum><cr><lf></code> <code>\$aa,WV?*hh<cr><lf></code>
	Sensor output:	<code>\$WIMWV,<angle>,<reference>,<speed>,<units>,<status>*</code> <code><checksum><cr><lf></code> <code>\$WIMWV,xxx,R,xxx.x,c,A*hh<cr><lf></code>
Parameters	<angle> 000 to 359	Measured wind direction in degrees relative to sensor datum
	<speed> 000.0 to 075.0 000.0 to 145.8 000.0 to 270.0	Measured wind speed (in metres per second). Measured wind speed (in knots). Measured wind speed (in km per hour).
	<reference> R T	Relative (compass disabled) True (compass enabled)
	<units> M N K	Indicates the wind speed is presented in metres/second Indicates the wind speed is presented in knots Indicates the wind speed is presented in kilometres/hour
	< status > 0 to Z	Indicates whether an error condition was detected by the operating system, such as out of range wind speed or incorrect signal level. Any character other than 'A' (ASCII 41(HEX)) = error.
Examples	Example 1	
	The following example illustrates the NMEA wind velocity data format. The example shows the sensor output in m/s with a wind speed of 20m/s and a wind angle of 45°.	
	<u>Message</u>	<u>Comment</u>
	<code>\$01,WV?*13<cr><lf></code>	Query the wind velocity
	<code>\$WIMWV,045,R,020.0,M,A*3D<cr><lf></code>	Sensor NMEA response
	Example 2	
	The following example illustrates the NMEA wind velocity data format. The example shows the sensor output in knots with a wind speed of 30.6 knots and a wind angle of 9°.	
	<u>Message</u>	<u>Comment</u>
	<code>\$01,WV?*13<cr><lf></code>	Query the wind velocity
	<code>\$WIMWV,009,R,030.6,N,A*31<cr><lf></code>	Sensor NMEA response
Description	The WV command returns the wind velocity value in the currently selected format and units. Polar or NMEA formats are available. Use the DF command, Section 8.3.13, to select the required output format and units.	
	Polar Format: The sensor returns the magnitude of the wind speed (m/s) and the wind direction (0-359°) (see WV Polar, Section 8.3.22).	
	NMEA 0183 Format: The sensor returns the NMEA 0183 Wind Speed and Angle sentence MWV. The sensor returns the wind direction (0-359°) and wind speed (m/s, knots or km/h) using the MWV Wind Speed and Angle sentence. The sensor Talker ID can be modified by the ID command (be aware that changing the Talker ID may result in incompatible NMEA 0183 data formatting)	

End of Manual – Back to **Contents**