



Automotive
Energy & Power Analysis
Aerospace & Defense
Transportation
General Test & Measurement

The Xsens Sensor Plug-in

Technical reference manual



ISO9001

Re-inventing Data Acquisition



Document History

Version	Date	Author	Changes
1.0	2013-09-05	Milan Svarny	Creation
1.01	2013-10-31	Christian Schmid Berthold Hinterplattner	Formatting
2.0	2014-03-07	Milan Svarny	Update new plugin version

Contents

1.	Plug-in Pre-Adjusts	5
1.1	Xsens Driver Instalation	5
1.2	Xsens Library registration	7
1.3	Plugin activation in DeweSoft	8
2.	Plug-in Description	9
2.1	Hardware Setup Screen	9
2.2	Channel Setup Screen.....	10
2.3	Measurement Screen	12
3.	Supported Hardware.....	13
4.	Limitation	14
5.	Overview of data output	15

Copyright © DEWETRON elektronische Messgeraete Ges.m.b.H.

This document contains information which is protected by copyright. All rights are reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

All trademarks and registered trademarks are acknowledged to be the property of their owners.

1. Plug-in Pre-Adjusts

The Plug-in is working with DEWESoft™ version 7.1 b114 and higher. It consists of one file and Xsens library files.

XsensUSB.dll

This file must be copied into DEWESoft's Add-ons folder, usually found under:
C:\DEWESoft7\Bin\V7_0\Addons

And Xsens Library.

The Installation of Xsens Library is composed from two part.

- Xsens Driver Installation
- Xsens Library registration

1.1 Xsens Driver Instalation

1. Extract driver files from „Xsens MTi 4th Generation driver.ZIP“
2. Start “SetupDriver.exe” (32 bit) or “SetupDriver_x64.exe” (64 bit)



Figure 1.0 Driver Installation

Connect Xsens to device and check if the device appears correctly in Windows device manager.

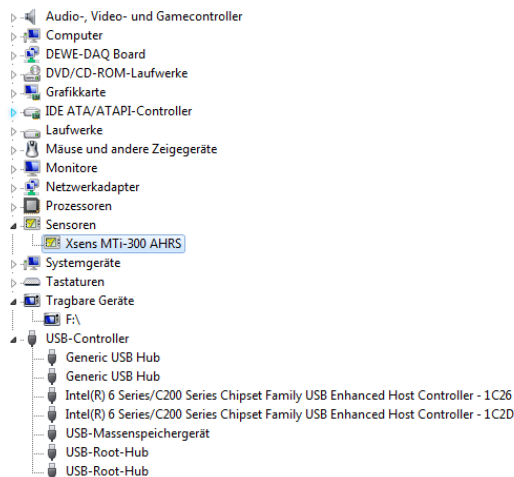


Figure 2.0 Windows Device Manger

XSENS PLUG-IN

Then Install Redistributable Package installs runtime components of Visual C++ Libraries required to run applications developed with Visual C++ on a computer that does not have Visual C++ installed.

32 bit system: “vcredist_x86.exe” (32 bit) or

64 bit system: “vcredist_x64.exe” (64 bit)

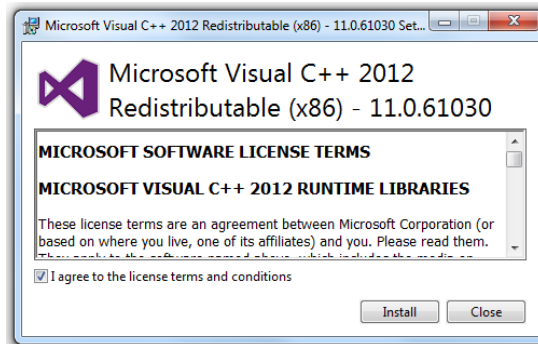


Figure 3.0 Redistributable Package installs runtime components of Visual C++

Copy the folder “Xsens” into DEWESoft Addons folder.

Folder contains:

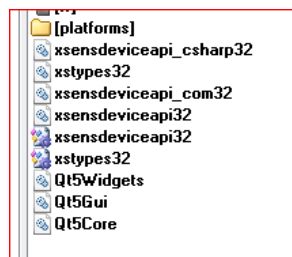


Figure 4.0 “Drive”:\DEWESoft7\Bin\V7_0\Addons\Xsens folder

1.2 Xsens Library registration

The xsensdeviceapi_com32.dll has to be registered with „regsvr32” command.
Start windows command line as administrator and register it.
Goto “Drive”:\DEWESoft7\Bin\U7_0\Addons\Xsens and type:

32 bit:

Regsvr32 xsensdeviceapi_com32.dll

64 bit:

C:\Windows\SysWOW64\Regsvr32.exe xsensdeviceapi_com32.dll

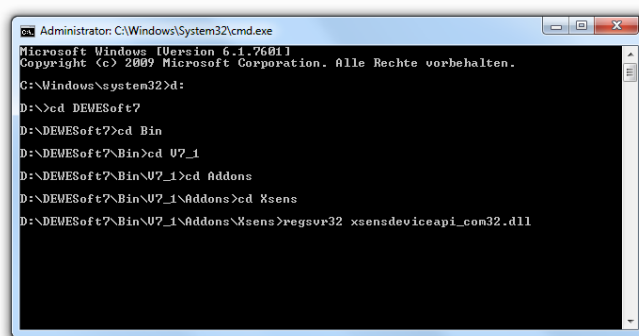


Figure 5.0 Xsens Lib Registering 32 Bit OS example

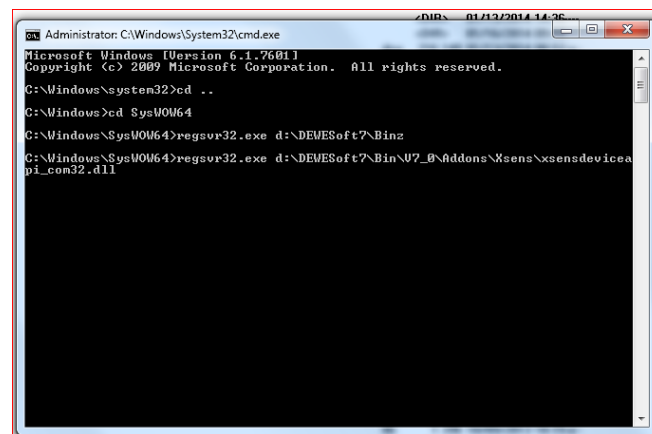


Figure 6.0 Xsens Lib Registering 64 Bit OS example

The successful executing command will push up information window.

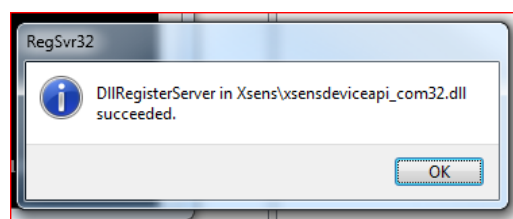


Figure 7.0 Xsens Lib Registered.

1.3 Plugin activation in DeweSoft

After all previous steps are done the plug-in has to be registered inside Dewesoft. This has to be done in DEWESoft's hardware setup.

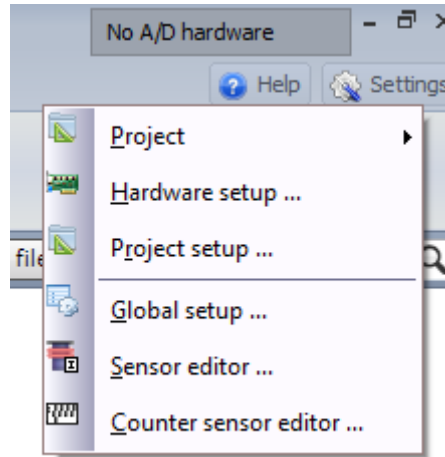


Figure 8.0 Settings

When the hardware setup has been opened, please press the *Reg.Plugins* button. Afterwards restart DEWESoft™ and the plug-in will be listed in the plug-in menu.

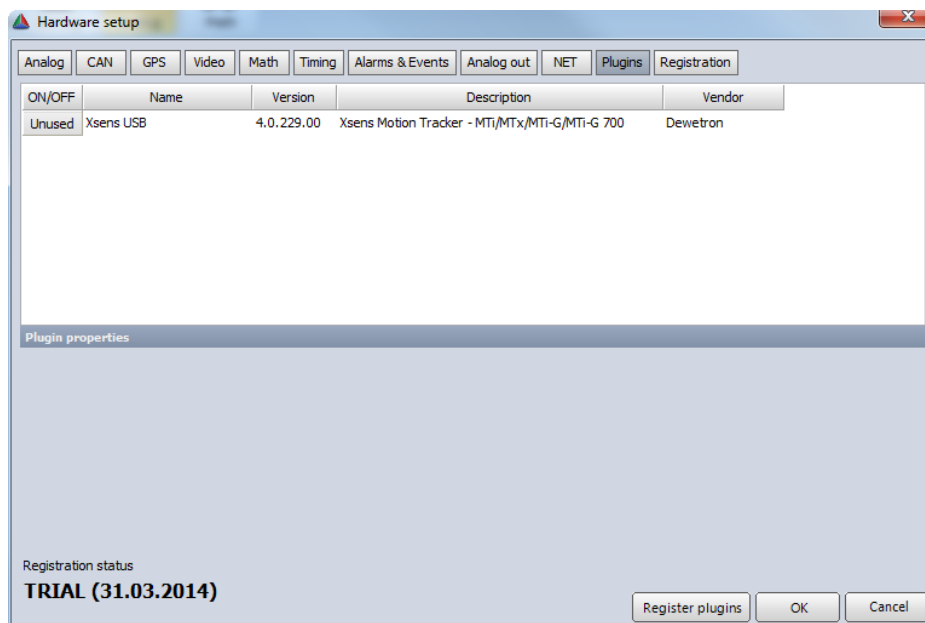


Figure 9.0 Hardware setup

Then it is possible to scan the presented Xsens sensors. When the plug-in is enabled a scan is done automatically.

The Rescan button can be used for checking presentation of the Xsens device. If the device is visible on the form then it is ready for measurement.

2. Plug-in Description

The plug-in controls the Xsens devices and collects measured data.

The control of the devices is based on Xsens SDK 4.2.1.

The measured values are set by the plug-in according to the type of device.

The plug-in can handle maximum two Xsens devices at the same time and the devices may be different type.

The Xsens device has a two states:

- Configuration
- Measurement

In config state various settings can be read and written according to the user request and the device type, in measurement state the MT will output its data message which contains data depending on the current configuration and provide the data to DEWESoft™.

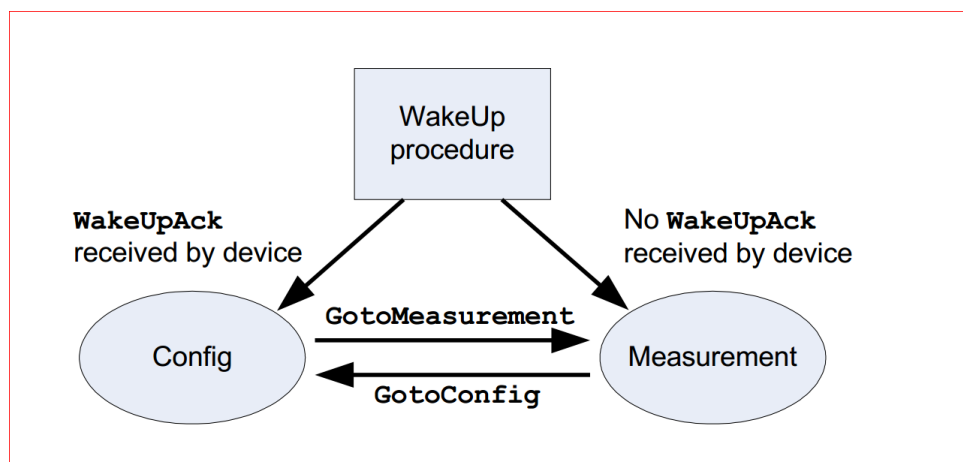


Figure 10.0: Xsens device states

The measurement process is event based. When the device has new data then Xsens SDK provides data by an event to the plug-in.

The plug-in stores the requested data with a time stamp to the list. When DEWESoft requests data then the list is provided to DCOM.

Measurement during “Setup screen” and “Measure screen” is the same procedure just data are provided once to “Setup screen” and to “Measure screen” on the other hand.

The plug-in has three main windows:

- Hardware setup screen
- Channel setup screen
- Measurement screen

2.1 Hardware Setup Screen

On this screen the plug-in detects devices and the main configuration of Xsens device is done. The Figure 2.0 shows two devices connected.

It is possible reconnect the Xsens devices in this time before starting the measurement. User can connect the correct Xsens device to the port and press “rescan port” or disable and enable the plug-in and the device will be enumerated, connected and the communication port opened.

2.2 Channel Setup Screen

On the Setup screen the user can set:

- Used channels
- Scenarios / Filters (defined by device)
- Delay
- Orientation data output
- Units
- Sampling Rate

The settings on setup screen can be applied for each device separately. The channels are mapped according to the device type. The user can choose which channels he would like use for measurement. The Scenarios/Filters are loaded from on-board Xsens device. Each device type can contain different types and versions of the scenario.

Delay represents time delay to measurement timestamp.

Orientation data output represents three types : Euler, Quaternion, Matrix.

Each channel is measured in the corresponding unit. For some value it could be recalculated to other type of unit or maximum and minimum limit value could be set.

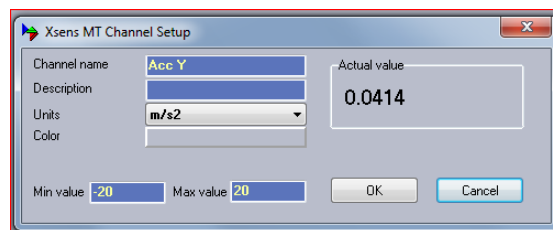


Figure 11.0: Channel Setup

The Sampling Rate could be set according data type in combo box list. The Combo box is activated immediately when user click on light gray box in Rate column.

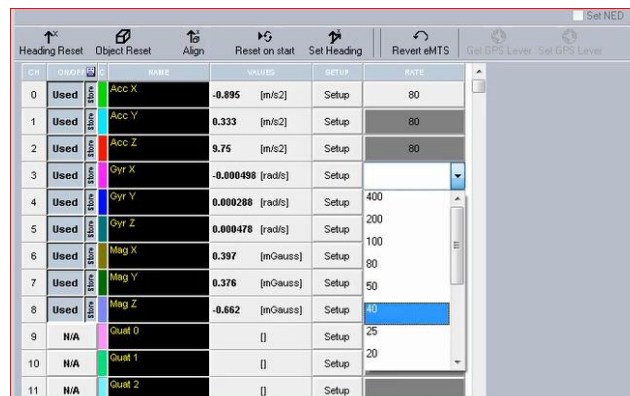


Figure 12.0: Sample Rate Setup

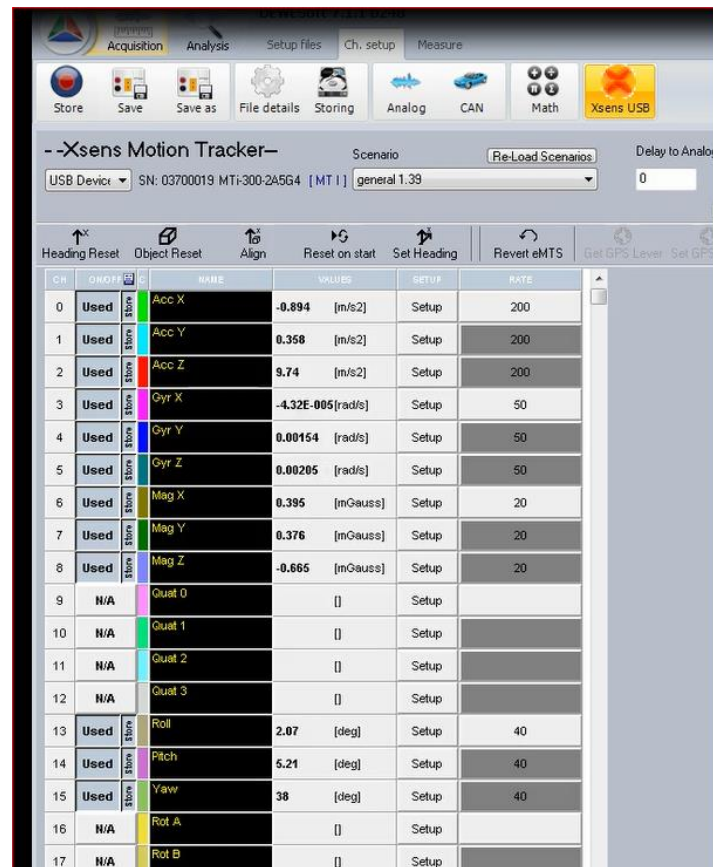


Figure 13.0: Channels Setup Screen

NOTE: The channels “Acc...” and “Gyr...” have one common sample rate which is defined by the “Acc” sample rate.

The setup of channels, colors and so on can be stored and reloaded again. The setup to load is used only if the number and type of device matches.

In case of inconsistency for one parameter the setup is refused by the plug-in.

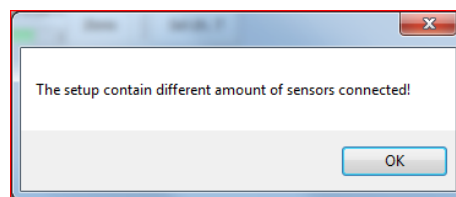


Figure 14.0 Setup Warning

Or

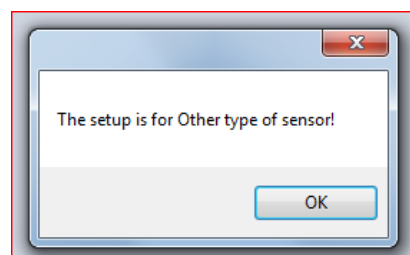


Figure 15.0 Setup Warning

Each device can be reset into these modes:

- XRM_StoreAlignmentMatrix - Store the current object alignment matrix to persistent memory
- XRM_Heading - Brief Reset the heading (yaw)
- XRM_Object - Reset the attitude (roll, pitch)
- XRM_Alignment - Reset heading and attitude. This effectively combines the XRM_Heading and XRM_Object

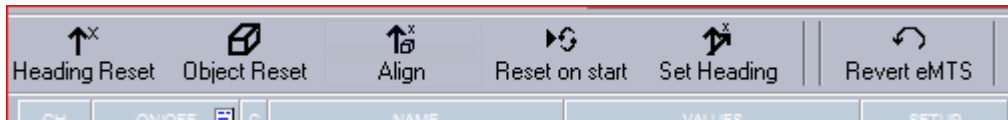


Figure 16.0: Reset type of device

Each reset mode is possible for all devices. When more than one sensor is connected then the window below is displayed (Figure8) where it is possible to select the device for applying the action.

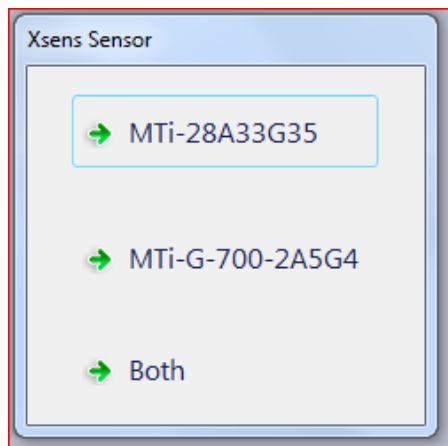


Figure 17.0 Sensor Choose

2.3 Measurement Screen

On this screen data is just provided according channel setup. The acquisition is done same way as for “Channel setup screen”. It is done in a periodical event from Xsens SDK.

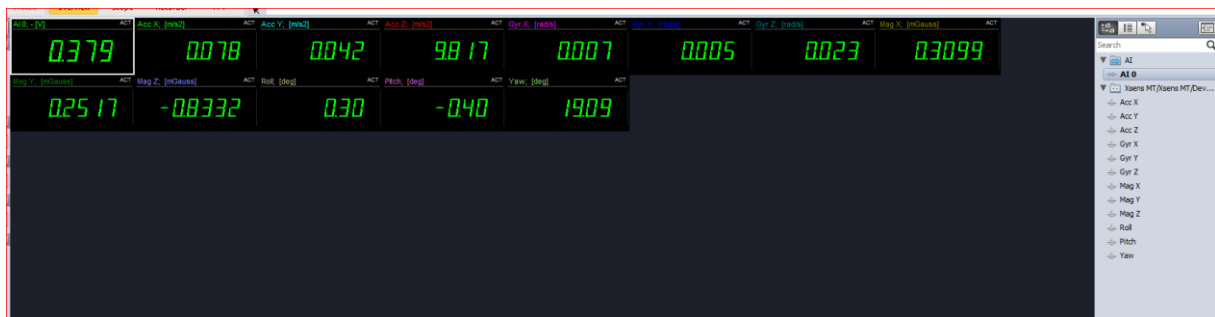


Figure 18.0 Measurement screen

3. Supported Hardware

The listings below show the supported devices for Plug-in versions > 4.0.

- MTi: MTi and MTx legacy Motion Trackers (3rd generation)
- MTi-G: MTi-G legacy Motion Tracker with GPS (3rd generation)
- 10: MTi-10 IMU (4th generation)
- 20: MTi-20 VRU (4th generation)
- 30: MTi-30 AHRS (4th generation)
- 100: MTi-100 IMU (4th generation)
- 200: MTi-200 VRU (4th generation)
- 300: MTi-300 AHRS (4th generation)
- 700: MTi-G-700 GPS/INS (4th generation)

Plug-in versions < 3.0 do NOT support 4th generation USB devices like for example the 700 series!!

4. Limitation

1.

The Xsens plug-in ver. 3.0 doesn't contain some values from MTi-G devices as Xsens Plug-in ver. 2.XX. The reason is different Xsens SDK. Plug-in ver. 3.0 use Xsens SDK 4.1.

From Xsens Support Email:

"First of all the legacy devices are not fully supported in XDA. It is not a 100% tested yet. This is probably the reason why your GPS PVT message is not fully functional. I'm sorry but this can only be solved by changing XDA legacy compatibility. I will give this information to our product manager.

For XDA the GPS age is not needed as the GPS comes in at its own frequency. So there is no GPS Age for the 700. For pressure age this is the same.
"

GPS PVT message contains missing data for MTi-G legacy device.

The missing data for MTi-G legacy are: Standard deviation (h, v, s) and GPS Age.

For the MTi-G 700 the missing values are GPS age and Pressure AGE.

Solution:

Xsens released a new revision of SDK 4.1. Unfortunately, it is not official up to now and not all files are necessary to import into Delphi environment is released by now.

It will be fixed in new revision most probably.

2.

The Plug-in can control maximum two Xsens devices at the same time.

3.

The measurement buffer time stamp is taken from computer system time.

5. Overview of data output

Group Name	Type Name	XDA type name	Hex Value	Valid for MTi product						
				10	20	30	100	200	300	700
Temperature		XDI_TemperatureGroup	08x0	•	•	•	•	•	•	•
	Temperature	XDI_Temperature	081y	•	•	•	•	•	•	•
Timestamp		XDI_TimestampGroup	10x0	•	•	•	•	•	•	•
	UTC Time	XDI_UtcTime	1010	•	•	•	•	•	•	•
	Packet Counter	XDI_PacketCounter	1020	•	•	•	•	•	•	•
	Integer Time of Week (ITOW)	XDI_Itow	1030	•	•	•	•	•	•	•
	GPS Age	XDI_GpsAge	1040							•
	Pressure Age (legacy)	XDI_PressureAge	1050							
	Sample Time Fine	XDI_SampleTimeFine	1060	•	•	•	•	•	•	•
	Sample Time Coarse	XDI_SampleTimeCoarse	1070	•	•	•	•	•	•	•
	Frame Range	XDI_FrameRange	1080							
Orientation Data		XDI_OrientationGroup	20xy		•	•		•	•	•
	Quaternion	XDI_Quaternion	201y		•	•		•	•	•
	Rotation Matrix	XDI_RotationMatrix	202y		•	•		•	•	•
	Euler Angles	XDI_EulerAngles	203y		•	•		•	•	•
Pressure		XDI_PressureGroup	30xy				•	•	•	•
	Baro Pressure	XDI_BaroPressure	301y				•	•	•	•
Acceleration		XDI_AccelerationGroup	40xy	•	•	•	•	•	•	•
	Delta V	XDI_DeltaV	401y	•	•	•	•	•	•	•
	Acceleration	XDI_Acceleration	402y	•	•	•	•	•	•	•
	Free Acceleration	XDI_FreeAcceleration	403y		•	•		•	•	•
Position		XDI_PositionGroup	50xy							•
	Altitude MSL	XDI_AltitudeMsl	501y							•
	Altitude Ellipsoid	XDI_AltitudeEllipsoid	502y							•
	Position ECEF	XDI_PositionEcef	503y							•
	LatLon	XDI_LatLon	504y							•
Angular Velocity		XDI_AngularVelocityGroup	80xy							
	Rate of Turn	XDI_RateOfTurn	802y	•	•	•	•	•	•	•
	Delta Q	XDI_DeltaQ	803y	•	•	•	•	•	•	•
GPS		XDI_GpsGroup	88x0							•
	DOP	XDI_GpsDop	8830							•
	SOL	XDI_GpsSol	8840							•
	Time UTC	XDI_GpsTimeUtc	8880							•
	SV Info	XDI_GpsSvInfo	88A0							•
Sensor Component Readout (SCR)		XDI_RawSensorGroup	A0x0	•	•	•	•	•	•	•
	ACC + GYR + MAG + temperature	XDI_RawAccGyrMagTemp	A010	•	•	•	•	•	•	•
	Gyro temperatures	XDI_RawGyroTemp	A020	•	•	•	•	•	•	•
Magnetic		XDI_MagneticGroup	C0xy	•	•	•	•	•	•	•
	Magnetic Field	XDI_MagneticField	C02y	•	•	•	•	•	•	•
Velocity		XDI_VelocityGroup	D0xy							•
	Velocity XYZ	XDI_VelocityXYZ	D01y							•
Status		XDI_StatusGroup	E0x0	•	•	•	•	•	•	•
	Status Byte	XDI_StatusByte	E010	•	•	•	•	•	•	•
	Status Word	XDI_StatusWord	E020	•	•	•	•	•	•	•
	RSSI	XDI_Rssi	E040							