



NovellaTM

Document MTAN002, Revision B, 15 Aug 2023

Xsens MTi for Mobile Warehouse Robotics

IMU, VRU, AHRS



Revision	Date	By	Changes
A	13 Sep 2022	SGL, AKO	First release
B	15 Aug 2023	SGL	Included MTi-320

© 2021-2023, Movella Inc. All rights reserved. Information in this document is subject to change without notice. All other trademarks are the property of their respective owners.

Contents

- 1 Introduction 4
- 2 Xsens MTi: Series and Functionalities 5
- 3 Getting started with the MTi Development/Starter Kit 8
- 4 Best practices – Hardware integration 14
- 5 Best practices – Device configuration and Outputs 16
- 6 Best practices – Software integration 21
- 7 Technical Support 23
- Appendix A: Sample data 24
- Appendix B: ROS driver modifications 27

1 Introduction

Xsens has a long history in developing industrial grade motion tracking modules. Already in 2007, Xsens released the MTi-G, the first fully integrated but compact GPS/INS device, combining high-quality MEMS-based inertial sensors with satellite positioning for a full plug-and-play navigation solution. Now, several generations and many years of experience in sensor fusion development later, Xsens offers a wide variety of motion tracking products in terms of form factor, accuracy, and pricing. Like the MTi-G, all of these are plug-and-play solutions. However, the Xsens MTi Series is also small, lightweight, and low in terms of power consumption. This makes the Xsens MTi Series highly attractive for AGV and AMR applications.

In this application note, we will focus on the use of Xsens MTi products in AGV and AMR applications. In chapter 2, we will discuss the key benefits of Motion Trackers and why they should be part of AGVs and AMRs. We will also introduce the MTi portfolio and explain which products are most suitable for AGV and AMR applications.

We will continue in chapter 3 with an introduction to the Xsens MTi Development Kit – explaining the basics of the included hardware and software. We will demonstrate how easy it is to get started.

In chapters 4 to 6, we will discuss best practices for the use of MTi products in AGV and AMR applications. How should your Motion Tracker be installed, and what are the optimal settings for AGV and AMR use cases? Should your device be calibrated or not? And which outputs are most useful? How to get the most out of the MT Software Development Kit?

All documentation and ways to access Xsens’ free-of-charge, global technical support will be discussed in chapter 7.

Finally, we conclude this application note with a basic implementation of all the information presented. Appendix A contains sample data from an actual robotic ground vehicle. Appendix B describes how the data was recorded using a ROS driver and a Raspberry Pi.



Figure 1: The Xsens MTi series line-up, showing (from left to right) the MTi 1-series, MTi 300-series, MTi 600-series and MTi 100-series.

2 Xsens MTi: Series and Functionalities

2.1 Why add an Xsens MTi to your AGV/AMR?

AGVs and AMRs can be equipped with a wide variety of sensors, including cameras, lidars and other devices that can sense the environment of the AGV/AMR. This allows the vehicle to detect objects, targets, beacons, and other points of interest, and helps it to localize itself. In addition, Inertial Measurement Units (IMUs) are often used to measure the dynamics and orientation of the vehicle itself. Visual, positional, and inertial sensors have proven to be a very powerful combination in the field of robotics. So-called SLAM (Simultaneous Localization And Mapping) algorithms, which make optimal use of combining various sensors, allow AGVs and AMRs to navigate autonomously even in challenging industrial environments.

Xsens's industrial grade Motion Trackers, called the MTi series, are small, lightweight, and low in terms of power consumption, especially compared to visual sensors and the computational power required to process their data. This makes the Xsens MTi series a highly attractive sensor module that can be included in AGV and AMR designs in order to increase their intelligence, and to improve their navigational performance and positioning accuracy, especially at higher velocities.

Xsens Motion Trackers are fully configurable, and can provide your AGV/AMR with any combination of the following data:

- ✓ 3D Angular Velocity
- ✓ 3D (Free) Acceleration
- ✓ 3D Magnetic Field
- ✓ 3D orientation (Roll/Pitch/Yaw, Quaternion or Rotation Matrix)
- ✓ Temperature
- ✓ Barometric Pressure
- ✓ Status information
- ✓ 3D Position, 3D Velocity and Time (GNSS/INS models only)

Compared to standard IMUs, Xsens Motion Trackers are:

- ✓ Fully calibrated for a wide variety of sensor characteristics, including but not limited to: sensor bias, temperature, non-linearity and misalignment;
- ✓ Available with different functionalities (e.g. IMU, VRU/AHT, AHRS, GNSS/INS), including on-board sensor fusion algorithms that are specifically designed for robotic applications, such as the Active Heading Stabilization feature;
- ✓ Available with various hardware interfaces (e.g. CAN, RS232, I²C, SPI, UART, USB);
- ✓ Available in different form factors (e.g. surface-mount, OEM or encased);
- ✓ Easily combined with other sensors using various synchronization features.

Finally, Xsens Motion Trackers are easy to test, evaluate and integrate:

- ✓ Hardware Development Kits available
- ✓ A ROS driver is available and actively maintained
- ✓ Free-of-charge MT Software Suite available, including:
 - Easy to use Graphical User Interface (MT Manager)
 - (Open source) Device API
 - Programming examples for C++, C#, Python, MATLAB
 - Magnetic Field Mapper software, for tackling magnetic disturbances
 - Firmware Updater
- ✓ Free worldwide technical support by local Product Specialists

2.2 Xsens MTi products

Xsens' MTi Series portfolio consists of various functionalities; all based on an Inertial Measurement Unit (IMU) but with varying degrees of on-board sensor fusion algorithms. Figure 2 summarizes the functionality of each variant.

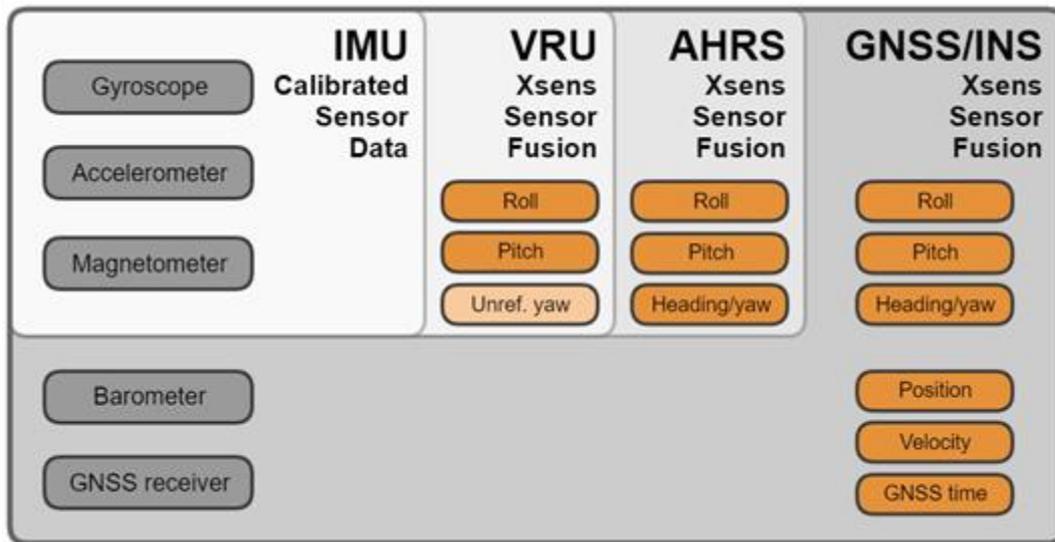


Figure 2: From IMU to GNSS/INS

IMU

The IMU variant is an Inertial Measurement Unit that measures 3D acceleration and 3D rate of turn with the addition of 3D magnetic field data and, depending on the product, barometric pressure. It does not fuse sensor data to deliver orientation estimates. The IMU can also be configured to output data generated by the strapdown integration algorithm (orientation increments Δq and velocity increments Δv).

VRU/AHT

The Vertical Reference Unit (VRU) or Active Heading Tracker (AHT) adds the first layer of algorithms which uses gravity as a reference for roll and pitch calculations. Essentially it delivers the same data as the AHRS, except for the yaw. The yaw estimate of a VRU/AHT product is unreferenced, which means that it is computed without any geographic/magnetic reference, though still superior to just gyroscope integration (e.g., when using the gyro bias estimation techniques). All data outputs from the IMU are also available in this product version. The Active Heading Stabilization feature is also available in this product variant.

AHRS

This is the full Attitude and Heading Reference System (AHRS). It provides various outputs: roll, pitch and heading (magnetic referenced yaw). In addition, all functionalities of the IMU and VRU/AHT are also available in this product variant.

GNSS/INS

The GNSS/INS variant is a product with an interface to an external or internal (RTK) GNSS receiver as well as a barometer. It provides roll, pitch, yaw/heading, as well as 3D position, 3D velocity and time data. In addition, all data outputs of the IMU, VRU/AHT and AHRS are also available in this product variant.

In this application note we will focus only on the **VRU/AHT** and **AHRS** functionalities, which are used most often for ground vehicles and other indoor use cases. Xsens' GNSS/INS models, which combine inertial sensing with global positioning, will not be discussed here. If you would like to learn more about the GNSS/INS products, we recommend reading our Application Note [Xsens GNSS/INS: Supercharge your GNSS Receiver](#).

Table 1 shows a compact overview of the VRU and AHRS models offered by Xsens at this moment.

Table 1: Available Xsens VRU/AHT and AHRS models

Product series	Functionality	Product code	Form factor		Interfaces
MTi 1-series	VRU/AHT AHRS	MTi-2 MTi-3	SMD		UART, SPI, I ² C
MTi 300-series (IP51)	VRU/AHT	MTi-320	Module		RS232
MTi 600-series (IP51)	VRU/AHT AHRS	MTi-620 MTi-630	Module		RS232, CAN, UART
MTi 600-series (IP68)	VRU/AHT AHRS	MTi-620R MTi-630R	Robust		RS232, CAN
MTi 100-series	VRU/AHT AHRS	MTi-200 MTi-300	Robust		RS232/RS422, USB

2.3 VRU/AHT or AHRS?

As described earlier, the main difference between the VRU/AHT and AHRS model is the behaviour of the Yaw output of the device: AHRS models provide an absolute Yaw output that is referenced by the Earth's magnetic field, whereas VRU/AHT models provide a relative Yaw output that is initialized at 0 degrees when the device powers up.

VRU/AHT models are used most often for indoor ground vehicles, because indoors the Earth's magnetic field can be disturbed, possibly leading to a decreased Yaw accuracy when using an AHRS model. Examples of local magnetic disturbances are magnets, engines, high-current cables and ferromagnetic objects such as shelves. Also, the global heading of the vehicle is often not important for indoor use cases.

In contrast, outdoor vehicles typically use an AHRS because they require a global heading, and because there are less external magnetic disturbances.

Fortunately, all AHRS models can be downgraded through software such that they will behave exactly as a VRU/AHT device. This way, an AHRS Development/Starter Kit, which we will discuss in the next chapter, can be used to evaluate both the VRU/AHT and AHRS functionalities.

3 Getting started with the MTi Development/Starter Kit

3.1 Overview of the Development/Starter Kits

The **MTi Development/Starter Kit** is a very easy to use developer's kit that allows for fast and easy integration of the MTi in any user scenario. A Development/Starter Kit is available for the following IMU/VRU/AHT/AHRS products:

- MTi-3 AHRS (also covers MTi-1-IMU and MTi-2-VRU/AHT)
- MTi-320 VRU/AHT
- MTi-630 AHRS (also covers MTi-610-IMU and MTi-620-VRU/AHT)
- MTi-630R AHRS (also covers MTi-610R-IMU and MTi-620R-VRU/AHT)
- MTi-100 IMU
- MTi-200 VRU/AHT
- MTi-300 AHRS

Starter Kits include all necessary cables and antennas, allowing users to easily connect the MTi over various interfaces, including USB. The MTi-3 Development Kit and MTi-630 Development Kit also include a Development Board, as shown in Figure 3.



Figure 3: MTi-3 and MTi-630 Development Boards

Software and documentation

The MTi series Development Kits are supported by the MT Software Suite, which includes the following software components:

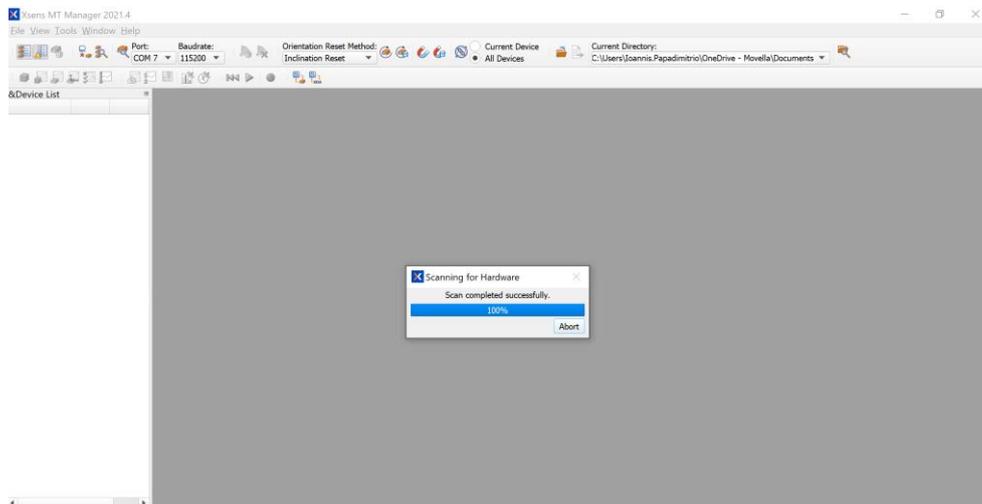
- MT Manager
- Magnetic Field Mapper
- MT SDK including programming examples and documentation

Additionally, the latest firmware for the MTi series can be downloaded and updated using the Firmware Updater which is separately available.

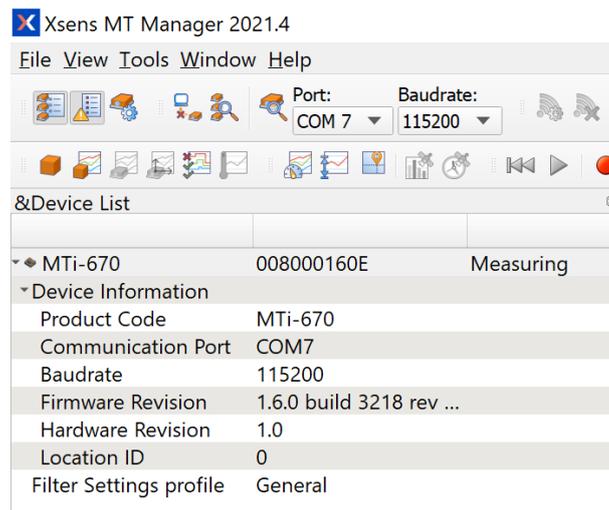
All software components can be downloaded from the Xsens website: <https://www.xsens.com/software-downloads>.

3.2 Configuring the MTi

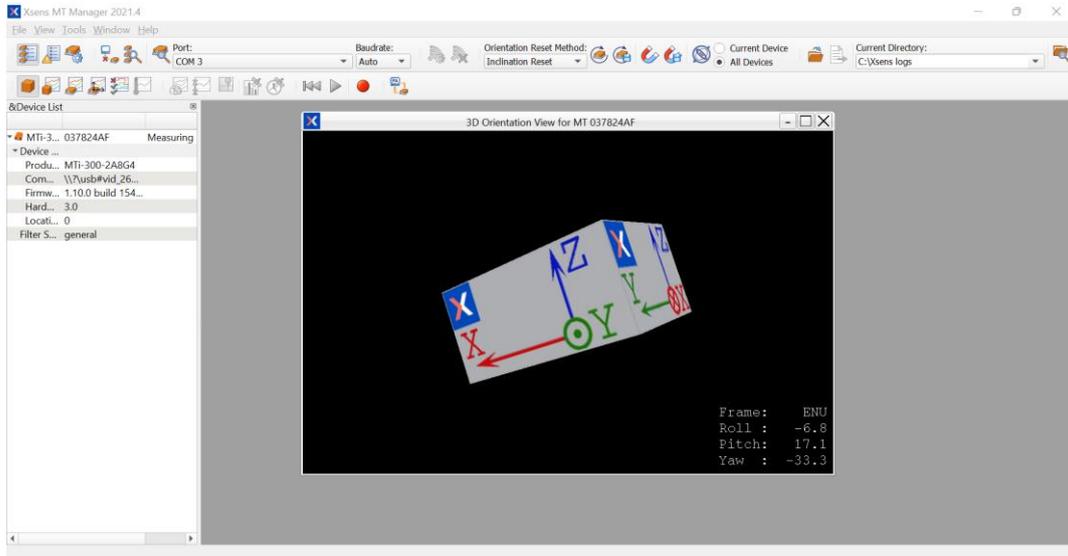
The MTi-series can be directly configured by means of MT Manager. Connect the MTi to your PC using the supplied USB cable. Upon start-up of the MT Manager, it automatically scans the available COM-ports and/or MTi USB devices on the PC for connected devices.



The MTi is displayed in the Device List with the respective unique MT device ID number.



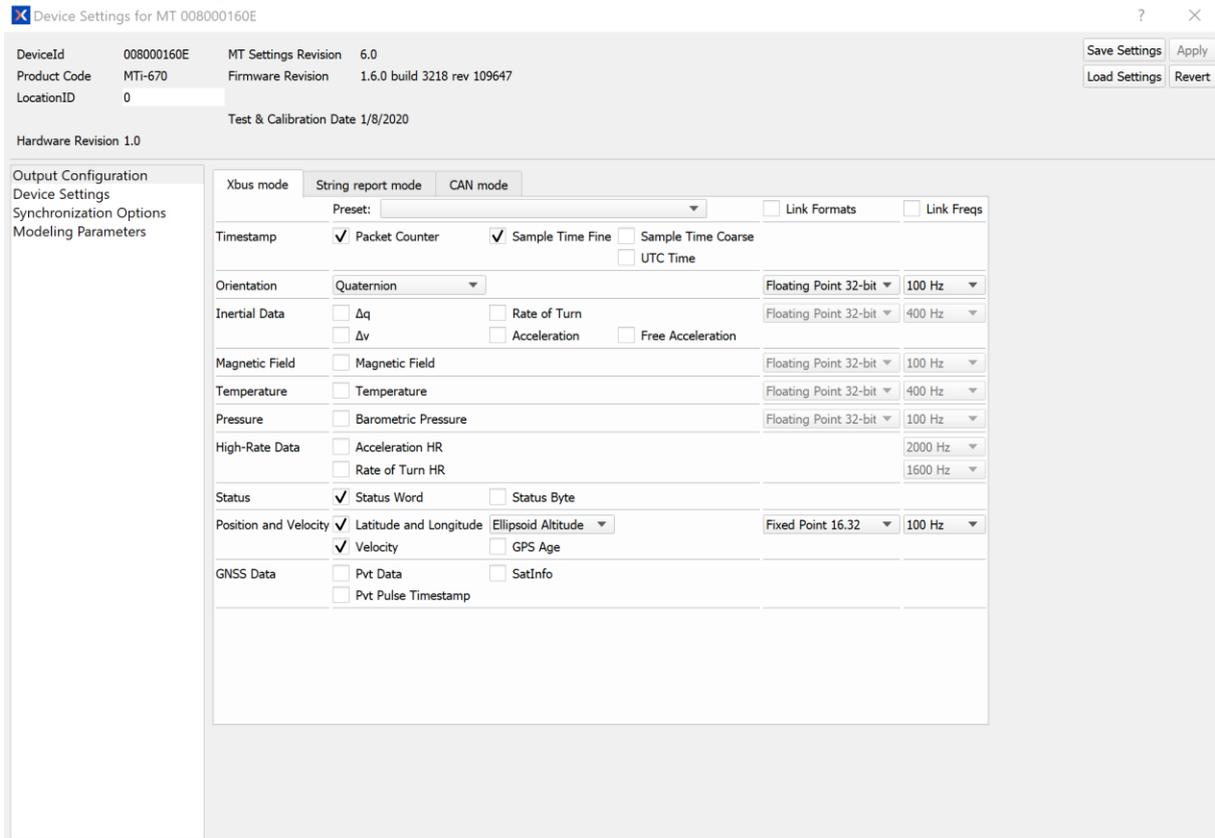
To see a real time 3D visualization of the orientation of the MTi, click the 3D View icon . The inertial data , orientation data in Euler angles , status data , position , altitude  and velocity  can be visualized by clicking their respective icons. In order to open these windows, the corresponding data outputs of the MTi need to be configured. Otherwise, the window icons will be grey.



In MT Manager click the Device Settings button  to open the Output Configuration dialog.

Output Configuration tab

In the Output Configuration tab you can choose the desired outputs of the MTi:



Once you have selected your desired outputs, click Apply to save the output settings in the MTi's internal memory.

Device Settings tab

In the Device Settings tab, you will find options to set up the MTi's communication interfaces and it is possible to change the behavior of the sensor fusion algorithms of the MTi. Under Filter Settings, you will find the available filter profiles, which can be used to set the VRU/AHT or AHRS behavior of your device. For further details on filter profiles, see chapter 5.

Device Settings for MT 008000160E

DeviceId: 008000160E MT Settings Revision: 6.0
 Product Code: MTI-670 Firmware Revision: 1.6.0 build 3218 rev 109647
 LocationID: 0 Test & Calibration Date: 1/8/2020
 Hardware Revision: 1.0

Save Settings Apply
 Load Settings Revert

Output Configuration
Device Settings
 Synchronization Options
 Modeling Parameters

COM port name: COM7

Port Configuration: RS232
 Protocol: Xbus Baudrate: 115200 Protocol: Xbus Baudrate: 115200
 CAN: Enable 250k Baudrate

Filter Settings: XDA: 21.2 General Onboard: 21.2 General Manage

Option Flags: In-Run Compass Calibration (ICC) - Beta
 Orientation Smoother

GNSS Configuration: GPS & GLONASS

u-blox GNSS Platform: Portable

GNSS Receiver Type Config.	Receiver Model	Baud Rate	Input Rate (Hz)	Receiver Options
u-Blox MAX-M8Q		115200	4	0

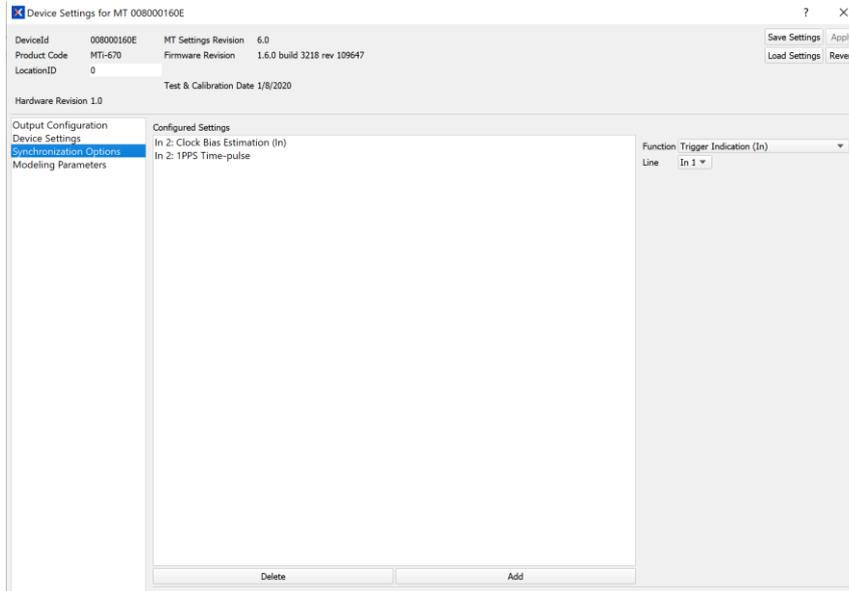
Position: Latitude [deg]: 52.24049169 Longitude [deg]: 6.838216771 Altitude [m]: 0

UTC Time: 1/1/1970 12:00 AM Get Set current

RotSensor			RotLocal		
1	0	0	1	0	0
0	1	0	0	1	0
0	0	1	0	0	1

Synchronization Option tab

In the Synchronization Option tab you can change and add Synchronization options in your MTi. For detailed explanations of the functions and how to use them, refer to the [Low Level Communication Protocol Documentation](#) and [Synchronization with the MTi](#).



Modeling Parameters tab

In the Modeling Parameters tab you can find the calibration parameters of the accelerometers, gyroscopes and magnetometers (offset, gain and misalignment; refer to the [MTi Family Reference Manual](#) for details).

	Accelerometers			Rate Gyros		
Offset	32667.6	32666.1	32560.2	32775.3	32772.2	32761.5
Gain	322.615	326.932	313.993	923.137	937.119	939.472
Misalignment	0.999992	-0.000286391	-0.00397574	0.999955	-0.00833611	-0.0051918
	-0.000489338	0.999958	0.00915082	0.00922184	0.999952	0.00422447
	0.00253361	-0.00921366	0.999955	0.0091406	-0.00307801	0.999956
	Magnetometers					
Offset	32448.8	32746.1	32961			
Gain	2092.69	2067.43	2078.62			
Misalignment	0.999716	0.0225834	-0.00757214			
	-0.016583	0.999786	0.0123431			
	-0.000344361	-0.0307013	0.999529			

3.3 Recording data

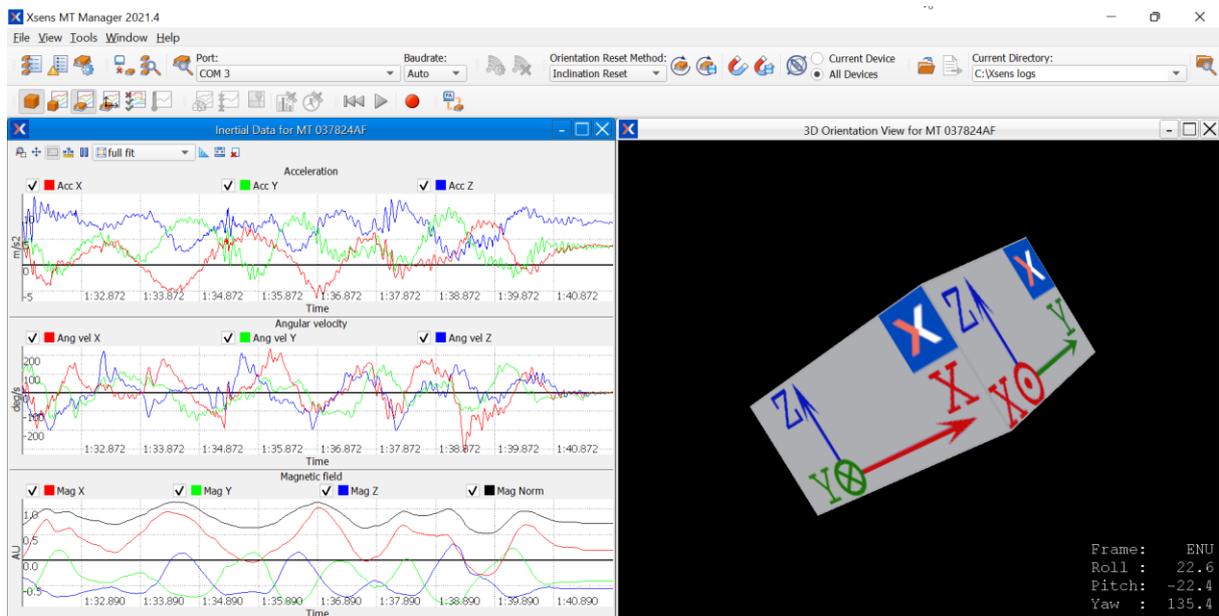
With MT Manager, it is possible to record data and export that data for use in other programs. Once you have configured the MTi to output the desired data, you can use the Record Function of the MT Manager in order to record data and generate log files.

First check if you receive data. Open, for example, the 3D view



and the inertial data graph





Logging data can be started and stopped by pressing the “Record / Stop Record” button:



The generated log file (.MTB file) will be written to the folder specified in the toolbar at the top of the main window under “Current directory”. By default, this is the root directory of the MT Manager installation (e.g. C:\Program Files\Xsens\MT Software Suite xxx.x\MT Manager).

These data can be later replayed with MT Manager and exported in different formats (ASCII and KMZ)

More information regarding the functions and features of MT Manager can be found in the [MT Manager User Manual](#).

4 Best practices – Hardware integration

In order to have your MTi product provide your AGV/AMR with the best data possible, it is important to start by making sure that the Motion Tracker is correctly installed. Incorrect installation may lead to erroneous or distorted sensor data as explained in the following paragraphs, which in turn can lead to a degraded sensor fusion performance. Next to the following paragraphs, we advise reading your MTi product series' [Hardware Integration Manual](#) for further guidance on connectors, cables, pinouts, etc.

4.1 Shocks and vibrations

For best results, it is recommended to mechanically isolate your MTi from shocks and vibrations as much as possible. Shocks and vibrations are measured by the accelerometers, which can make it difficult for the Motion Tracker to estimate the vehicles Roll/Pitch angles. They can also affect the gyroscope component, resulting in measurement errors. Because of this, the high-end MTi 100-series are equipped with vibration-rejecting gyroscopes, which are less affected by shocks and vibrations due to their mechanical design.

Xsens has tested a set of vibration dampeners on the MTi. Vibration dampeners are low-profile rubber cylinders that allow the MTi to be mounted on an object without a direct metal to metal connection that transduces vibrations from the object to the MTi. The vibration dampeners have been tested with frequencies up to 1200 Hz that caused aliasing when the MTi was mounted directly on the vibration table. These vibrations had no effect with the vibration dampeners fitted. The dampeners tested are manufactured by Norelem and have part number 26102-00800855 see Figure 4. Alternatively, vibrations and shocks can often be dampened by mounting the Motion Tracker on top of a thin piece of foam/rubber or using a thick piece of adhesive tape.



Figure 4: Vibration dampeners

4.2 Center of rotation

When using an MTi to measure the dynamics of a moving vehicle, it is best to position the MTi close to the centre of rotation (CR) of the vehicle. Any rotations around the centre of rotation translate into centripetal accelerations at any point outside the centre of rotation, which means that the accelerometers may be affected by quick turns.

4.3 Magnetic fields

When an AHRS MTi is placed close to or on an object that is either magnetic or contains ferromagnetic materials, the measured magnetic field is distorted (warped) and causes an error in the computed heading. The earth's magnetic field is altered by the presence of ferromagnetic materials, permanent magnets or power lines with strong currents (several amperes) in the vicinity of the device. The distance to the object and the amount of ferromagnetic material determines the magnitude of disturbance introduced. Errors in estimated yaw due to such distortions can be quite large, since the earth's magnetic field is very weak in comparison to the magnitude of the sources of distortion. If the local Earth magnetic field is temporarily disturbed, the on-board AHRS algorithms will initially track this disturbance instead of incorrectly assuming that the device has rotated. However, in case of continuous magnetic disturbances (>10 to 30 s, depending on the filter settings), the computed heading will slowly converge to a new solution using the 'new' local magnetic north. Note that the magnetic field has no direct effect on the inclination (Roll/Pitch) estimates.

After integrating an MTi into an AGV/AMR, constant magnetic disturbances will be present. Using a so-called 'Magnetic Field Mapping' (MFM, i.e. a 3D calibration for soft and hard iron effects), these magnetic disturbances can be completely calibrated for, allowing the MTi to be used as if it would not be secured to the object containing ferromagnetic materials. This process will be discussed further in chapter 5. Fluctuating magnetic disturbances, such as those generated by engines or wireless transmitters, cannot be compensated for. It is therefore highly recommended to position the MTi as far as possible from these sources.

Using MT Manager, you can easily identify whether magnetic disturbances are near your MTi, by monitoring the magnetic norm: [Using the magnetic norm to identify magnetic distortions](#)

4.4 Interfacing with a PC

When integrating an MTi into your vehicle, we recommend including the possibility for connecting with a PC in the future, for various reasons:

- It will be easier to carry out a magnetic calibration if needed (see chapter 5.5)
- It will be easier to update the firmware of your MTi
- It will be easier to inspect or change your device configuration using MT Manager
- It will be easier to inspect or record your device's output data in case you ever need technical support

Allowing for a PC connection is possible by:

- Exposing the UART pins of your MTi and connecting a UART-to-USB converter to those pins (MTi 1-series¹, MTi 600-series)
- Exposing the RS232 pins of your MTi and connecting an RS232-to-USB converter to those pins (MTi 300-series, MTi 600-series, MTi 100-series)
- Allowing to connect the CA-MP-MTI-12 cable in combination with the CA-USB-CONV converter (MTi 300-series, rugged MTi 600-series)
- Allowing to connect one of the available USB cables (MTi 100-series²)

NB: It is not recommended to integrate MTi 1-series or MTi 600-series [Development Boards](#) into a vehicle. Development Boards are designed for testing and evaluation only and are subject to future hardware modifications.

¹ Note that the UART interface of the MTi 1-series has to be enabled by pulling the PSEL lines to ground, e.g. using jumpers or a switch.

² Various USB cables are available in our [web shop](#).

5 Best practices – Device configuration and Outputs

5.1 Output configuration

Xsens Motion Trackers are suitable for a wide range of applications, from marine to aerospace and from autonomous vehicles to gimbal stabilization. Each of these applications may require a different set of output messages from the MTi. If you are going to use an Xsens Motion Tracker for an AGV/AMR use case, then it is also important to start by configuring the required outputs for your device. Fortunately, this is made very easy when using the Device Settings window in MT Manager as explained in chapter 3.

When deciding on the output configuration of your MTi, consider the following:

- Which data outputs are essential for my AGV/AMR?
- Which data outputs could be useful in the future, e.g. for debugging or additional functionality?
- Which output frequency is required by my AGV/AMR? Note that it is possible to configure the output rate for each individual output message. For instance, you can receive Rate of Turn data at 100 Hz and Temperature data at 1 Hz.

In any case, we recommend enabling the StatusWord output. This output provides valuable information regarding the real-time state of your MTi, including sensor clipping flags, filter state and synchronization flags. An overview of the StatusWord contents can be found in the [MT Low Level Communication Protocol Documentation](#).

For a general description of all available output messages, please see the [MTi Family Reference Manual](#).

Output rates versus Interface Bandwidth

Please note that the hardware interface of your device needs to have enough bandwidth to be able to transmit all output data of the MTi. For serial interfaces (e.g. UART, RS232, USB when using a converter) MT Manager will typically show “data overflow” warning messages at the bottom of the screen in case there is too much data to be transmitted, see Figure 5. In this case you will need to either:

- Increase your device’s baud rate, or;
- Decrease the amount of output messages, or;
- Decrease the output frequency of your output messages.

For CAN, I²C and SPI interfaces, similar “data overflow” warning messages will be sent out by the MTi in case the bus load is too high (CAN) or if data messages are not read out fast enough by the host (I²C and SPI).

```
&Message Window
04.08.2022 - 11:30:57.417 -> WARNING: Device: 0788005B: Data Overflow detected. Sample times may not be correct. Selected baudrate may be too low.
04.08.2022 - 11:30:57.417 -> WARNING: Device: 0788005B: Data Overflow detected. Sample times may not be correct. Selected baudrate may be too low.
04.08.2022 - 11:30:57.338 -> WARNING: Device: 0788005B: Data Overflow detected. Sample times may not be correct. Selected baudrate may be too low.
04.08.2022 - 11:30:57.261 -> WARNING: Device: 0788005B: Data Overflow detected. Sample times may not be correct. Selected baudrate may be too low.
```

Figure 5: MT Manager displaying data overflow warnings

5.2 Filter profiles

Xsens Motion Trackers offer various filter profiles, depending on their functionality and series (MTi 1-series, 600-series, 100-series). Filter profiles are used to tune the sensor fusion algorithms of the MTi for a specific use case, or to achieve a specific desired output behaviour. Filter profiles only affect the outputs of the sensor

fusion algorithms, such as the orientation (Roll/Pitch/Yaw) estimates. Sensor fusion inputs, such as Acceleration and Magnetic Field, are not affected by filter profiles.

For **VRU/AHT** devices, the amount of filter settings is limited. All VRU/AHT devices support a “VRU(-general)” profile, which is also the default setting. In addition, the user can enable Active Heading Stabilization (AHS), which is recommended for most AGV/AMR use cases. We will discuss Active Heading Stabilization in the next paragraph. Finally, the MTi-620(R) allows you to select a **Robust** profile which is the right choice for AGV/AMR applications³.

For **AHRS** products, the available filter profiles vary between products:

- For the MTi-3, we recommend using the **high_mag_dep** profile if the magnetic disturbances in your vehicle’s environment are very low. The **general** and **north reference** profiles are a better choice if magnetic disturbances are occasionally present.
- For the MTi-630(R), we recommend using the **Robust** profile for AGV/AMR applications, combined with either the **NorthReference** or **FixedMagRef** heading behaviour⁴.
- For the MTi-300, we recommend using the **high_mag_dep** profile if the magnetic disturbances in your vehicle’s environment are very low. The **general** and **low_mag_dep** profiles are a better choice if magnetic disturbances are occasionally present.

In all cases, for **AHRS** products please note the following:

- You will need to carry out a magnetic calibration after mounting the MTi into your AGV/AMR. This will compensate for static magnetic disturbances that are caused by the vehicle itself. Without a magnetic calibration, the Yaw estimates of your AHRS device will likely become erroneous. See also chapter 5.5.
- If the environment in which your vehicle is used is unreliable due to many (local) magnetic disturbances, and if your application does not require an absolute Yaw output, then we recommend using the VRU/AHT functionality instead.

Comparing different filter settings

If you are still in doubt about which filter settings to use, we recommend simply comparing the results for various filter settings. For this purpose, Xsens’ MT Manager offers a useful reprocessing feature: If you record data from the MTi with a specific output configuration called “XDA Processing”, then you can open the recording in MT Manager. MT Manager is then able to apply the sensor fusion algorithms instead of the MTi running them on-board. This way, you can record a single log file and compare how the MTi would have performed with various filter settings, but under exactly the same circumstances. This procedure is explained on BASE: [Recording an MTB file with XDA Processing](#)

After recording data, you can simply open multiple instances of MT Manager (each with the recording processed using a different filter setting) and compare the orientation estimates.

³ The *Robust* profile is tuned for slowly moving applications that experience few unexpected rotations and accelerations. This contrasts with the *Responsive* profile which is more suitable for drones and handheld applications. Finally, the *General* profile is a legacy setting that is no longer recommended for new designs.

⁴ The *NorthReference* and *FixedMagRef* heading behaviours allow the user to configure how the filter copes with locally disturbed magnetic fields. To learn more about the differences, please see [this tutorial video](#).

5.3 Active Heading Stabilization

Active Heading Stabilization (AHS) is a software component within the sensor fusion engine designed to increase the Yaw accuracy of VRU/AHT devices, even in a disturbed magnetic environment. For the MTi 600-series, the AHS feature is embedded in the filter profile “VRUAHS”. For the MTi 1-series, MTi 300-series and MTi 100-series, AHS is a functionality that can be enabled separately. For those devices, it is recommended to use the vru_general filter profile when enabling AHS.

Active Heading Stabilization aims to reduce the accumulation of errors in the Yaw output (also known as drift) which normally arises through the integration of gyroscope data over longer time periods. It does so by making use of magnetic field measurements; not as a reference, as is the case with AHRS profiles, but as an additional source of information on angular motion.

Because of the way the Active Heading Stabilization algorithm uses magnetic field data, it is recommended to avoid fluctuating (time dependent) magnetic disturbances, such as those generated by engines or wireless transmitters. It is therefore highly recommended to position the MTi as far as possible from these sources.

For more details and a tutorial video on AHS, we refer to [BASE](#).

5.4 Manual Gyro Bias Estimation

Manual Gyro Bias Estimation, or MGBE in short, is a feature that can be used to improve the MTi's internal estimate of biases on the gyroscope data. Together with sensor noise, gyroscope biases are the main source of orientation estimation errors when relying on gyroscope readings. For a general introduction to sensor biases and their influence, we recommend reading [this article](#). The MGBE feature can significantly improve the orientation output performance, particularly for VRU/AHT but also for AHRS devices. MGBE is highly recommended to be used with MTi 1-series devices, as those tend to show larger gyroscope biases than MTi 300-series, MTi 600-series and MTi 100-series devices due to their small form factor and mechanical stress that may arise during hardware integration.

MGBE can be used whenever the MTi and the vehicle in which it is integrated are motionless for a short duration. At that point the user (or the vehicle's motion controller) can send a command to the MTi such that it will estimate its gyroscope biases as accurately as possible, within a couple seconds; the duration is configurable.

For AGV and AMR use cases it is highly recommended to program the vehicle controller such that it sends out the command once after power up, assuming that the vehicle does not start moving immediately. The orientation output performance can be further optimized by sending the command periodically (e.g. once per XX minutes), or by sending the command automatically whenever the vehicle's controller knows that the vehicle will be standing still for a short period of time, which will result in a higher success rate). Sending the command more often than once per 15 minutes is not necessary

For more details and a tutorial video on Manual Gyro Bias Estimation we refer to [BASE](#).

5.5 Magnetic calibration

As explained in chapter 4.3, carrying out a magnetic calibration is essential when using an AHRS device. Without a proper magnetic calibration, the magnetic field that is sensed by the magnetometers will likely be distorted continuously, resulting in Yaw estimation errors.

For VRU/AHT devices, magnetic calibrations are recommended when Active Heading Stabilization is enabled, but not required. This is because the Active Heading Stabilization does not make use of the magnetic field as a reference, but purely as an aiding source in addition to the gyroscopes. With AHS disabled, a VRU/AHT device will not make any use of magnetometer data in its sensor fusion algorithms and therefore a magnetic calibration is not needed.

During a magnetic calibration, you will need to gather magnetic field data by slowly rotating your vehicle (with the MTi mounted) in as many orientations as possible. For AGVs/AMRs, a horizontal plane calibration (only rotating around the Z-axis, about 3 full circles) is sufficient; this is called a 2D calibration. However, if your vehicle regularly faces inclinations such as ramps, it is recommended to include these dynamics in the magnetic calibration as well. After a magnetic calibration the newly computed magnetometer calibration parameters will be stored in the memory of the MTi. This means that it is not required to repeat the magnetic calibration if the MTi is mounted at the same location on the AGV/AMR, and as long as no structural changes have been made to the AGV/AMR.

The best (and recommended) way of carrying out a magnetic calibration is by using the Magnetic Field Mapper, which is included as a separate tool in the MT Software Suite. A typical calibration result for this tool can be seen in Figure 6. The Magnetic Field Mapper does however require a PC connection (Windows/Linux)⁵. Alternative to the Magnetic Field Mapper GUI, an MFM SDK is available including C++ example codes. If a direct PC connection is not possible after integrating the MTi into the AGV/AMR, then there are several alternative magnetic calibration methods available:

1. **Off-line Magnetic Field Mapping⁶:** In this case, measurement data will first be recorded on the AGV/AMR itself, and then transferred to a PC for use with the Magnetic Field Mapper. The calibration results can be written back to the device using a single binary command.
2. **Copying calibration parameters from one vehicle to another:** In this case, the Magnetic Field Mapper will need to be used at least once. The calibration results can be written to any other vehicle's MTi using a single binary command. Of course, it is essential that the ferromagnetic design of each vehicle is identical, and the MTi should be mounted at the same location as in the vehicle that was used with the Magnetic Field Mapper.
3. **In-Run Compass Calibration / Representative Motion (beta):** In this case, the magnetic calibration algorithm is not running on a PC, but the calibration is performed in real-time by the processor of the MTi. This functionality is less accurate/powerful than the Magnetic Field Mapper, but it eliminates the need for a host PC as it only requires sending a couple binary commands. For more information on this functionality, please refer to BASE: [In-run Compass Calibration \(ICC\) and Representative Motion](#)

⁵ ARM architectures (e.g. Raspberry Pi, NVIDIA Jetson) are not supported by the MFM GUI/SDK at the moment.

⁶ Only possible with serial interfaces (RS232, UART, USB) because of the format required by the Magnetic Field Mapper.

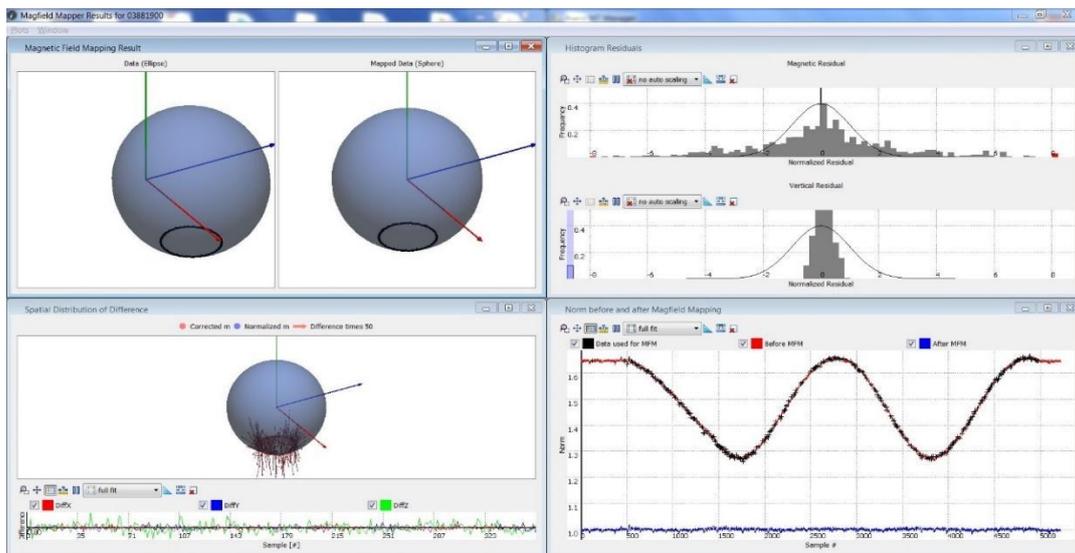


Figure 6: Example report of the Magnetic Field Mapper GUI for a 2D calibration. The application is a wheeled robotic ground vehicle consisting of a large metal structure with batteries and motors nearby. During the calibration, the vehicle drove two full circles. The magnetic field measurements have been mapped – in other words, the newly computed calibration parameters have compensated for magnetic disturbances caused by the vehicle itself.

5.6 Orientation resets

The MTi series (IMU models excluded) offer various orientation reset methods, allowing the user to reset the Roll, Pitch and/or Yaw output of the MTi to 0 degrees using a single command. These resets can also be stored in the memory of the MTi, allowing the changes to remain valid after repowering the device. The following reset methods are available:

- **Inclination reset:** Redefines the sensor-fixed frame such that Roll and Pitch outputs become 0 degrees;
- **Heading reset:** Redefines the local Earth-fixed frame such that the Yaw (Heading) output becomes 0 degrees;
- **Alignment reset:** A combination of the Inclination reset and Heading reset;

Care has to be taken that the Inclination reset also affects the calibrated sensor data, as these are presented in the (redefined) sensor-fixed frame. Similarly, the Heading reset also affects the Free Acceleration output which is presented in the (redefined) local Earth-fixed frame.

For further technical details regarding orientation resets, we refer to BASE: [Changing or Resetting the MTi reference co-ordinate systems](#)

For AGV/AMR use cases, the Inclination reset is especially useful to compensate for constant errors that arise if the MTi is not perfectly levelled after integration, or if it is mounted at an angle. The Heading reset is especially useful to “nullify” Yaw drift whenever a known reference point is reached, such as a docking/charging station.

6 Best practices – Software integration

6.1 Communication options and example codes

Once you have found the right settings for your device and once you are content with the data received, it is time to start incorporating the MTi into your software. As explained in chapter 2.1, the MTi can offer various hardware interfaces. Depending on the hardware interface of your choice, Xsens provides various programming examples and other tools that will help you get started. These can be found in the MT SDK folder of your MT Software Suite installation folder.

Besides MT Manager, which is designed for first-time use, configuration and evaluation, there are two primary ways of communicating with your MTi device which we will discuss next.



Figure 7: Impressions of the Xsens Device API (left) and low-level communication (right).

1. **The (public) Xsens Device API⁷** allows your host to communicate with your MTi through an extensive API, supporting C++, C#, Python and MATLAB development. The (public) Xsens Device API is fully documented in the Xsens Device API library⁸.

Example codes are available for C++, C#, Python and MATLAB. These demonstrate how to:

- Connect with your MTi
- Configure its outputs
- Receive and record its data
- Extract data from a recording
- Carry out a magnetic calibration (MFM SDK)

The example codes can be easily extended with your preferred configuration settings, synchronization options, etc. using the Xsens Device API library.

2. **Low-level communication (Xbus)⁹** allows for raw (binary) communication between the MTi and the host, no drivers or libraries needed. The Xbus protocol covers all functionality of the MTi and is fully documented in the MT Low-Level Communication Protocol document.

⁷ Not compatible with the I²C, SPI and CAN interfaces.

⁸ The Xsens Device API library can be found inside the Documentation folder of your MT Software Suite.

⁹ Not compatible with the CAN interface. The CAN interface of the MTi 600-series follows the CAN output messages as described in the [MT CAN Protocol Documentation](#). Example codes are not available, but a [.dbc file is provided](#) that will help you to translate all incoming CAN data messages.

C-based example codes are available which allow you to communicate with the MTi 1-series over the SPI and I²C interface using an ST Nucleo board. These demonstrate:

- The basics of communicating with the device over SPI and I²C
- How to receive and translate data from your MTi
- How to carry out a firmware update without access to a PC

For a complete overview of the MT SDK programming examples and their location, please see BASE:

[Introduction to the MT SDK programming examples for MTi devices](#)

In addition to the example codes, guides are available on BASE that explain how to communicate with common hardware such as [NVIDIA Jetson](#), [Raspberry Pi](#) and [Arduino](#).

6.2 Robotic Operating System (ROS)

As AGVs and especially AMRs are becoming more intelligent and equipped with different types of sensors, the need for combining sensor drivers and algorithms is increasing. The Robotic Operating System (ROS) allows developers to easily combine data from various sources (sensors, engines, batteries, etc.), perform sensor fusion (e.g. SLAM algorithms) and decision making, and control/actuate the robot – all using a single open-source software framework.

Xsens offers a ROS driver as part of the Linux MT SDK. It is written in C++ and based on the public Xsens Device API. Because of that, it can also be used with ARM-based architectures that are popular among robotics developers, such as the Raspberry Pi and NVIDIA Jetson.

Documentation for the ROS driver and installation instructions are available online: http://wiki.ros.org/xsens_mti_driver

The Xsens ROS driver formed the basis for a 3rd party **ROS2** which has been published in collaboration with Xsens. It is available on [Github](#).

Both Xsens ROS drivers offer the following functionality:

- Scan for and connect with an Xsens MTi (serial interfaces only, e.g. RS232, UART, USB)
- Receive data packets from the MTi
- Publish the data as separate topics
- Logging to .mtb (optional)
- 3D visualization in Rviz (optional)

Next to that, because of the use of the public Xsens Device API, the driver can be easily extended with your preferred configuration settings, synchronization options, etc. using the Xsens Device API library¹⁰.

In the Appendix of this application note, we will demonstrate the use of the (slightly modified) ROS driver to gather data from an MTi-630 using a Raspberry Pi.

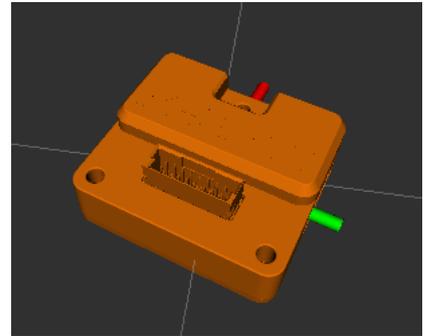


Figure 8: Rviz visualization of an MTi 600-series device

¹⁰ The Xsens Device API library can be found inside the Documentation folder of your MT Software Suite.

7 Technical Support

7.1 Online documentation

All the latest documentation is available at: www.xsens.com/xsens-mti-documentation

The table below provides an overview of all documents listed on this page, categorized by MTi product series.

Table 2: Overview of documentation available for the MTi series.

MTi 1-series	MTi -320	MTi 600-series	MTi 10/100-series
MTi Family Reference Manual			MTi User Manual
MTi 1-series Datasheet	MTi-320 Datasheet	MTi 600-series Datasheet	
MTi 1-series DK User Manual	MTi-320 SK User Manual	MTi 600-series DK User Manual	
MTi 1-series Hardware Integration Manual	n/a	MTi 600-series Hardware Integration Manual	
		MT CAN Protocol Documentation	
MT Manager Manual			
Magnetic Calibration Manual			
MT Low Level Communication Protocol Documentation			
Firmware Updater User Manual			

7.2 BASE by Xsens

BASE by Xsens is the right place to go for additional information, such as best practices, tutorials, troubleshooting and third-party integrations. Our documentation often refers to pages on BASE for further reading.



BASE consists of the following three components:

- A public technical support forum, maintained by our Product Specialists
- Articles, guides and FAQs
- Tutorial videos

7.3 Technical support

Finally, please feel free to contact our global team of Product Specialists. We are available 24/5 for technical support, free of charge! To get in contact, please fill in [the web form on BASE](#) or send an e-mail to support@xsens.com.

Appendix A: Sample data

The sample data set presented in this appendix can be downloaded by clicking [here](#). The log file can be opened using MT Manager, part of the MT Software Suite. This software package is available free to download from: <https://www.xsens.com/software-downloads>

This Indoor Robot sample data package contains a log file of the MTi-630 AHRS, though the exact same behaviour can be achieved with the MTi-620 VRU/AHT based on filter profile selection (see chapter 5.2). We have used the Robust/VRUAHS filter settings.

The 30-minute data set has been recorded by a Raspberry Pi 4B running the Xsens ROS driver, while the MTi was mounted on top of a vacuum cleaner robot (see Figure 9). At the start of the recording, the robot was positioned at its docking station. The robot then drove around and returned to its docking station. The ROS driver was slightly modified such that it would automatically send out a Manual Gyro Bias Estimation command for 5 seconds upon connecting with the MTi, and periodically (every 30 seconds) after that. However, because the robot was driving around nearly continuously, only the initial MGBE call succeeded and all periodic calls were rejected, as can be seen in Figure 10. The modifications to the ROS driver will be presented in Appendix B.



Figure 9: The vacuum robot connected to its docking station. Mounted on top are the encased Raspberry Pi 4B, a power bank and the MTi-630. All connections (not shown on the photo) were established through USB.



Figure 10: The Status Word output of the MTi-630 showing periodical Manual Gyro Bias Estimation calls. However only the first call (before the robot took off from its docking station) succeeded. All other calls were automatically rejected by the MTi-630 because the device was not motionless.

During the experiments the robot also bumped into many walls and objects, even resulting in several "clipping events" (see Figure 11) in which the device's accelerometer range was exceeded. Nevertheless, the resulting Yaw drift (difference in Yaw output between start and end) is only 2.4 degrees (Figure 12).

As explained in chapter 5.6, since the robot has returned to a known reference point (its docking station) this is an ideal point at which a heading reset command could be sent to "nullify" the Yaw drift before taking off again.

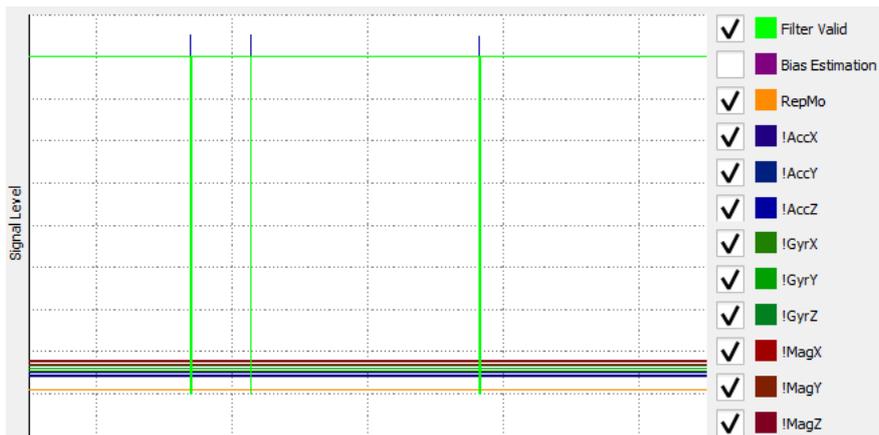


Figure 11: Several occurrences of z-axis accelerometer clipping, caused by bumping into walls and/or objects, were reported by the MTi-630 during the 30-minute trial. Consequently, the Filter Valid flag was dropped temporarily to indicate a potential decrease in orientation accuracy.

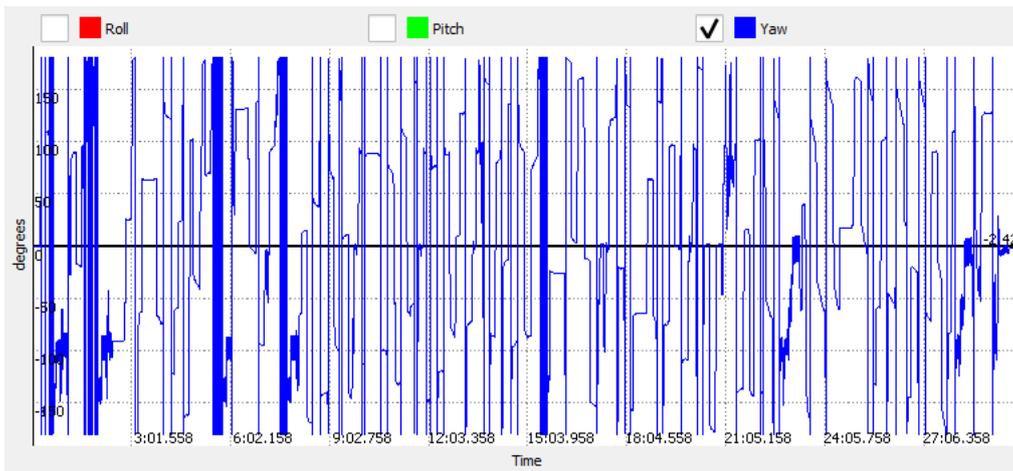


Figure 12: Yaw output of the MTi-630 during the 30-minute trial, showing nearly continuous dynamics. The error in Yaw after returning to the docking station is -2.42 degrees.

Appendix B: ROS driver modifications

In this section we describe the changes (highlighted in yellow) made to the Xsens ROS driver in order to generate the sample data that was presented in Appendix A. We take the ROS driver that is part of the MT Software Suite 2021.4 for Linux as a starting point. See chapter 6.2 for more details. We decided to extend this ROS driver such that it automatically sends out Manual Gyro Bias Estimation commands to the connected MTI. As explained in chapter 5.4, this helps reducing Yaw drift over time when especially when using the VRU/AHT profiles.

We start by declaring a new function, which will trigger a Manual Gyro Bias Estimation call, in *xdainterface.cpp*. For that we make use of the Xsens Device API library. It already provides API calls that can be used to construct and send the MGBE command, with a user-provided duration as input. Additionally, we declare a ROS callback function which will always initiate a Manual Gyro Bias Estimation of 3 seconds when called:

```
...
bool XdaInterface::manualGyroBiasEstimation(uint16_t duration)
{
    XsMessage snd(XMID_SetNoRotation, sizeof(uint16_t));
    XsMessage rcv;
    snd.setDataShort(duration);
    if (!m_device->sendCustomMessage(snd, true, rcv, 1000))
        return false;

    ROS_INFO("Manual Gyro Bias Estimation initiated.");
    return true;
}

void XdaInterface::callbackGyroBiasEstimation(const ros::TimerEvent& event)
{
    manualGyroBiasEstimation(3);
}
...
```

These functions also need to be defined in *xdainterface.h*:

```
...
class XdaInterface
{
public:
    ...
    void callbackGyroBiasEstimation(const ros::TimerEvent& event);
    bool manualGyroBiasEstimation(uint16_t duration);
    ...
};
```

Finally, we can modify *main.cpp* to implement the desired function calls. In this case we decide to call the function once after connecting with the device, and after that periodically (every 30 seconds) using `ros::Timer` and our callback function. Ideally, the function would be called whenever the controller knows that the vehicle is standing still, e.g. based on actuator control inputs, however the robotic setup we used for this experiment did not support this functionality.

```
...
int main(int argc, char *argv[])
{
    ros::init(argc, argv, "xsens_driver");
    ros::NodeHandle node;

    XdaInterface *xdaInterface = new XdaInterface();

    xdaInterface->registerPublishers(node);

    if (!xdaInterface->connectDevice())
        return -1;

    if (!xdaInterface->prepare())
        return -1;

    //Perform bias estimation for 5 seconds after initialization
    if (!xdaInterface->manualGyroBiasEstimation(5))
        return -1;

    //Perform bias estimation every 30 seconds
    ros::Timer timer =
node.createTimer(ros::Duration(30), &XdaInterface::callbackGyroBiasEstimation, xdaIn
terface);

    while (ros::ok())
    {
        xdaInterface->spinFor(milliseconds(100));

        ros::spinOnce();
    }

    xdaInterface->close();

    return 0;
}
```

Recording data to a .mtb file such that you can review the data in MT Manager does not require any modifications to the ROS driver itself. Simply open the parameter file *xsens_mti_node.yaml* and uncomment the *log_file* parameter. We also recommend uncommenting the *port* and *baudrate* parameters and specifying the correct values for your MTi device, in order to avoid port scanning issues:

```
## Device settings, provide one of the following:
## - device_id (most secure if you use multiple devices)
## - port + baudrate information
## - nothing: the driver scans for devices and selects the first found.
# device_id: '077007EF' # uppercase hex string
port: '/dev/ttyUSB0'
baudrate: 921600 # non necessary for some devices

## Log file (optional), placed in ~/.ros/ otherwise use absolute path
log_file: 'log.mtb'
...
```