

OS3D

MINIATURE 3D ORIENTATION SENSOR



User's Manual
Rev. 1.3

The **Inertial Labs OS3D** is a multi-purpose miniature 3D orientation sensor – Attitude and Heading Reference System, designed for use in real-time orientation tracking applications. It includes three types of sensing elements: tri-axial MEMS Gyroscopes, tri-axial MEMS Accelerometers, and tri-axial Magnetometers. The **OS3D** also comes equipped with an onboard processor and embedded orientation algorithms allowing for direct integration into systems without interfacing a PC (onboard calculation is available on OSv3 only, see OS3D Interface Control Document, clause 4.2 GetIden command for OS3D sensor types). Additionally, **OS3D** comes with hard and soft 2D and 3D magnetic calibration algorithms make the **OS3D** eliminate the magnetic influence in the magnetic interference environment.



Applications:

- Agriculture Robots
- Motion Capture Systems
- Training, Simulation and Head Tracking Systems
- Helmets Orientation Systems
- Quadrotor Helicopters and micro UAV
- Small Unmanned Underwater Vehicles
- Small gimbals and EOS
- Marine antenna stabilization system

KEY FEATURES AND FUNCTIONALITY

- Real-time Heading, Pitch and Roll orientation information
- Small size, lightweight and low power consumption
- Static Accuracy better than 0.2 deg in Pitch/Roll and 0.5 deg in Heading
- State-of-the-art algorithms for different dynamic motions of Robots, micro UAV, small UUV, small Gimbals and Antennas stabilization
- Ideal solution for Free Space Tracking Systems
- Gyro-Stabilized Slaved Magnetic Heading
- Embedded 2D and 3D magnetic calibration on hard and soft iron
- Up to 2000Hz data update rate
- OEM version available

Data from the Gyroscopes, Accelerometers, and Magnetometers, as well as the internal temperature sensor are gathered and processed by the on-board digital signal processor (DSP). The fusion algorithm processes these data and outputs the final orientation solution directly from the sensor. Data of the following types can be requested: raw inertial sensors data and/or quaternion data.

Each **OS3D** module is individually calibrated in a special non-magnetic laboratory where reference accelerations, angular rates, and magnetic fields are applied to the device and measured at constant temperature. Additionally, temperature cycling is performed to obtain temperature calibration parameters for the gyro and accelerometer elements. Once fielded, **OS3D** is able to be customer calibrated against soft- and hard-iron interference present in the end application.

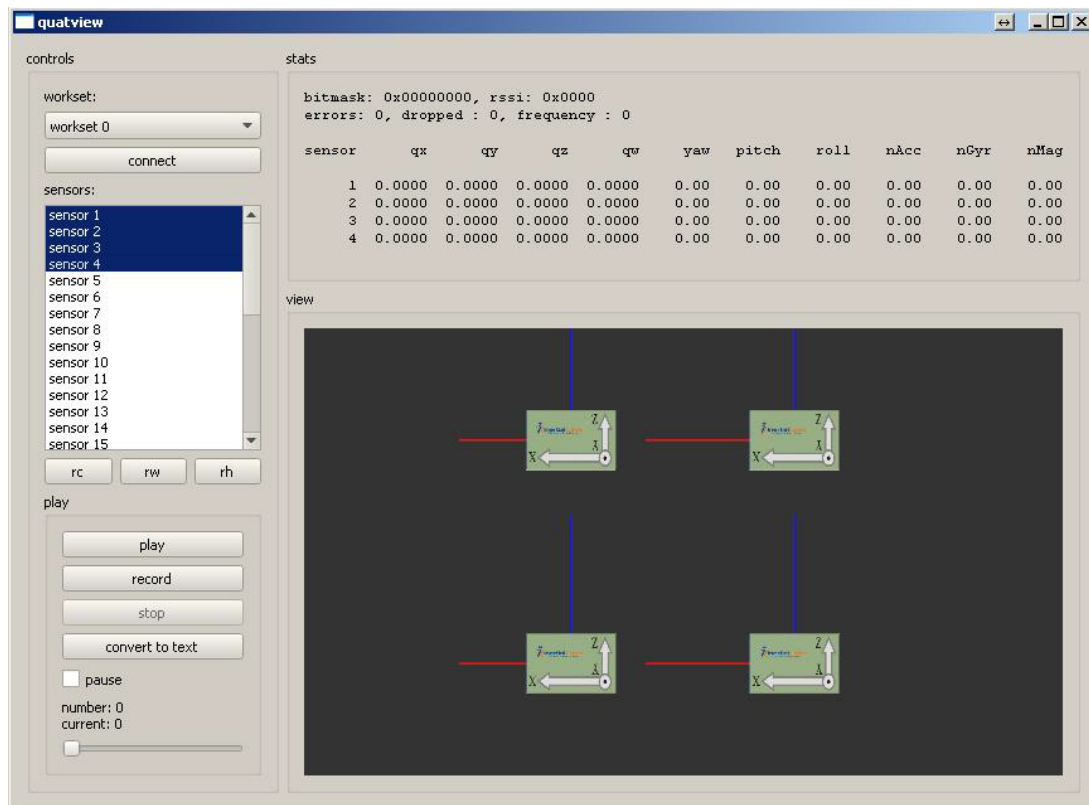
OS3D SDK Lite Demo Program

The Software Development Kit (SDK) Lite provides the ability for developers to create third party applications that are able to operate using the **OS3D** to provide orientation data within the application. One such application is the SDK Lite Demo Program (**QuatView**) by Inertial Labs.

From within **QuatView** it is possible to:

- Estimate the sensor's performance;
- Record data from sensors;
- Playback the saved data

QuatView Demo Software is intended for OS3D-family sensors with the SB-CU-USB device as control unit (or the USB-RS485 adaptor if single sensor is used). QuatView can be used to test the Inertial Labs OS3D sensor functionality and performance. Within the interface it continuously shows the output quaternion data along with Euler angles for both global and zero-stated coordinate bases. It also shows a representative 3D-block with the sensor axes drawn on it for easier visualization of sensor movements on the screen.



In addition to its use with single sensor configurations, QuatView is designed to be able to accept input of up to 31 simultaneously connected sensors whose data are passed through the connected SB-CU-USB. In multi-sensor configurations it is possible to view representative cubes for each sensor simultaneously.

1 Requirements

The SDK runs on Windows XP/Vista/Windows 7 and Linux x86/x64. Passing of sensor data to the PC must be done using one of the available OS3D Sensor Bus Control units: USB-RS485 adaptor (single sensor only) or SB-CU-USB (up to 31 sensors).

2 Software Installation

QuatView is an executable that does not require installation, it can be started from its location anywhere on the PC. Just copy the whole folder `inertiallabs_sdk_lite_api` to the desired location on your hard drive.

3 Sensor Driver

The Inertial Labs **OS3D** sensor data is passed to the PC via the sensor bus control units: USB-RS485 adaptor (single sensor only) or SB-CU-USB. Each of these is based on a commercial FTDI chip. In order for proper operation the FTDI driver must be installed prior to utilization of the SDK. The driver creates a virtual serial port (VCP) which is used by the SDK to communicate with the sensor(s). For VCP driver installation procedure please refer to the following link: <http://www.ftdichip.com/Support/Documents/InstallGuides.htm>

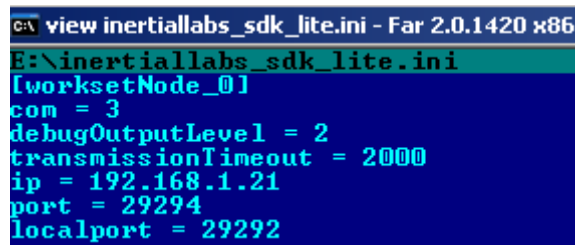
4 Sensor Connection and Start Up

Make sure you have the **OS3D** sensor(s) connected correctly to the SB-CU-USB or to USB-RS485 adaptor (for single sensor only) using the cable provided with shipment. Make sure that the control unit is also properly connected to the PC, the USB driver installed, and the COM port for the system noted (see item 3 “Sensor Driver Installation” above). Make sure the SB-CU-USB is powered up, if you use SB-CU-USB as the control unit. Please see the **Appendix D** for sensor's connection.

NOTE: User can access the OS3D data via various interfaces like RS232, RS422 and TTL UART using the Inertial Labs multiple interface cable. The cable also allows to use the wider supply voltage range. Please see the **Appendix D** for sensor's connection.

Follow the below steps to begin working with the unit after you connected it to PC:

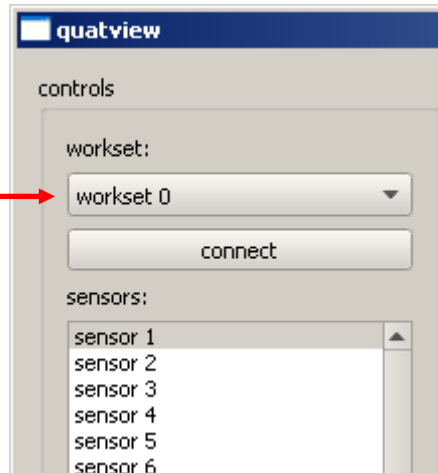
- 4.1 Update the “InertialLabs_SDK_Lite.ini” file located in the `..\inertiallabs_sdk_lite_api\bin\x32\` or `..\inertiallabs_sdk_lite_api\bin\x64\` directory. This file can be edited with any standard text editor. In the second line in the file it defines the COM port being used for “WorksetNode_0”. This COM port should match that which was identified during driver installation. See **Appendix C - Checking the serial port number**.



```
cmd view inertiallabs_sdk_lite.ini - Far 2.0.1420 x86
E:\inertiallabs_sdk_lite_api\bin\x32\inertiallabs_sdk_lite.ini
[worksetNode_0]
com = 3
debugOutputLevel = 2
transmissionTimeout = 2000
ip = 192.168.1.21
port = 29294
localport = 29292
```

4.2 Start QuatView by selecting the QuatView.exe executable from within the directory
..\inertiallabs_sdk_lite_api\bin\x32\ or ..\inertiallabs_sdk_lite_api\bin\x64\.

4.3 To get started it is needed to select the appropriate workset. Place the mouse cursor over the dropdown list in the upper left corner of the **QuatView** screen where the different potential worksets are available. From the list select “Workset 0” using the left mouse button. Press the “Connect” button. Allow several seconds for the sensor to get started.

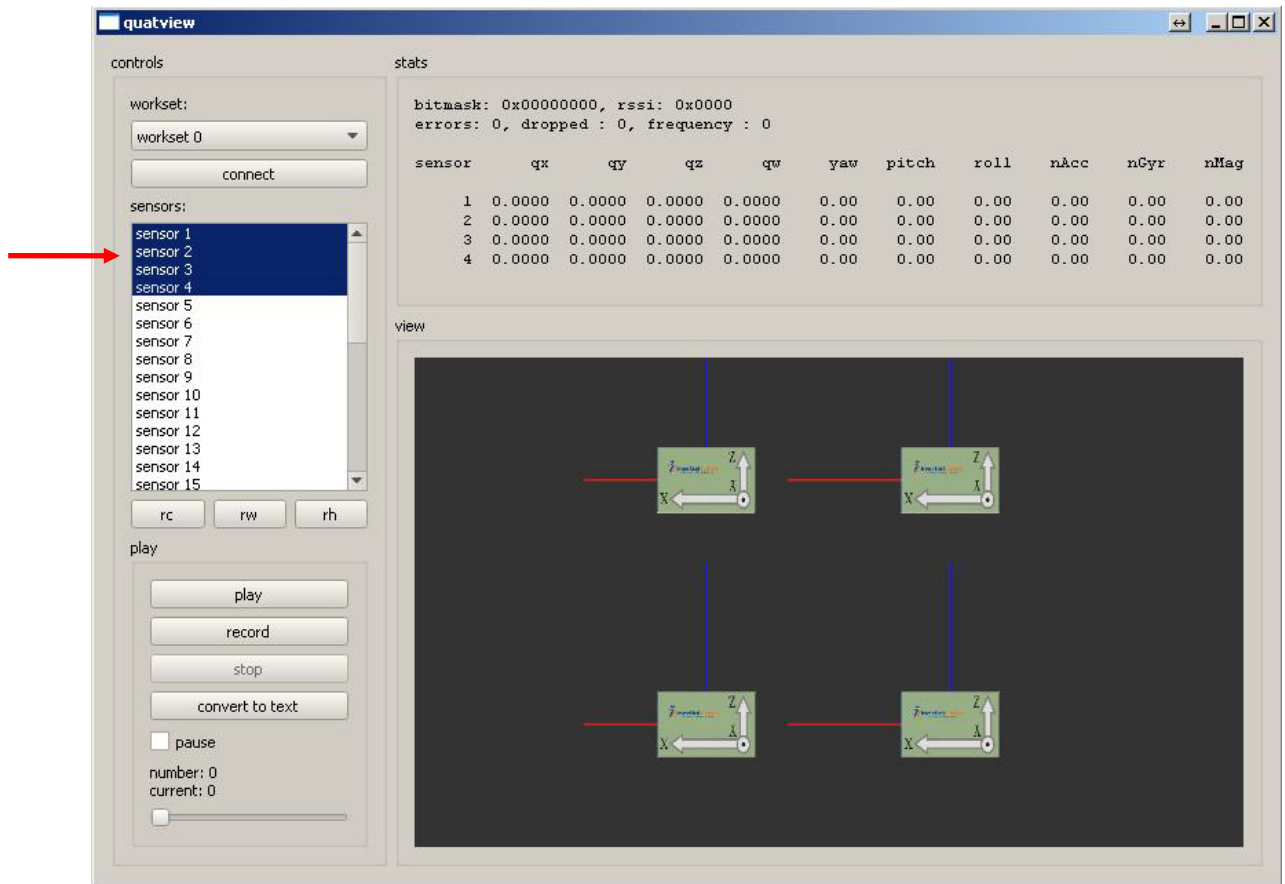


NOTE: If the sensor does not start after several seconds, there is likely an error in the COM port setting in the INI file. See step 1.4.1.

4.4 After several seconds data should be visible in the interface and the representative 3D cube should display the orientation of the sensor.

NOTE: In order to get accurate orientation it is required to have the sensor sitting still when you click the “Workset 0” button and for several seconds thereafter. Within these first several seconds the sensor goes through an initial alignment getting its initial orientation and checking the biases of the gyroscopes. If you see odd behavior of the sensor after start up, there is a good chance that this is because of either a poor initial alignment or starting the sensor in a very poor magnetic environment.

4.5 To show multiple sensors, it is possible within **QuatView** to select any number of sensors up to 31 to display simultaneously in the interface. In the sensors box under the selected workset, select the desired sensors to show by pressing either the Shift or Ctrl key and click on each desired sensor. The software will show blocks for each selected sensor. The orientation data are shown for only the most recently selected sensor.



5 Record/Playback Data

QuatView allows for the recording and playback of data from different demonstration sessions. This is a very useful feature for users who want to subject sensors to controlled tests in order to later check the same exact tests using different sensor parameters. Follow the below steps in order to record and playback data:

- 5.1 Check for the playback workset within the "InertialLabs_SDK_Lite.ini" file located in the ..\inertiallabs_sdk_lite_api\bin\x32\ or ..\inertiallabs_sdk_lite_api\bin\x64\ directory. This file can be opened with any standard text editor.
- 5.2 To record data, after you have successfully started the sensor, select the "Record" option from the interface, select the file to save data in. From that moment data is being recorded.
- 5.3 Once you have collected the desired data select "Stop".
- 5.4 To playback data from the recorded sequence, select "Play". This will open a new window allowing you to browse your PC for the data file you desire to review. Select the desired file. Playback can be fast-forwarded or rewound using the playback slider under the file save buttons.

6 Saved data format

- 6.1** The recorded data file is binary file that contains the sequence of frames saved as `ilInertialLabs_SDK_Lite_DataFrame`. See **OS3D-FG SDK-lite** document for details.
- 6.2** The button “Convert to text” allows user to convert the saved data from binary format to text. You need to choose the folder where a data was saved in a binary form, and the new text file will be created at the same folder. If you work with multiple sensors, the separate file will be created for every sensor. The name of the text file corresponds to the sensor's short address. The text data file is a space-delimited ASCII file. The data structure is as follows:

Line 1: explication, it shows the data columns assignment.

Example (shown here with word wrap):

counter 1, raw 9 (acc xyz, gyr xyz, mag xyz), calibrated 10 (acc xyz, gyr xyz, mag xyz, t), quaternion 4 (xyzw), angles 3 (yaw, pitch, roll)

Line 2 and the rest lines: data of the first and subsequent frames.

See **Appendix A - Sensor Data File Structure** for details.

7 Reset buttons

The reset buttons have the following functionality:

rc – cold reset – scans the sensor bus and restarts the orientation algorithm;

rw – warm reset - restarts the orientation algorithm without scanning the sensor bus,

rh – hot reset – resets the heading according to current values obtained from magnetometer.

8 Disconnecting the sensor

To stop the sensor, press the “Disconnect” button and simply close the QuatView application. This will close the connection to the serial port. Although technically the sensor can be disconnected physically at any time, the application must be restarted in order to reconnect to the sensor.

Appendix A - Sensor Data File Structure

Data format:

Space-delimited ASCII decimal numbers (integer or fixed-point 1.15 format).

File structure:

The sensor data file consists of bus data frames. A data frame contains a snapshot of inertial data acquired from the sensor bus plus the processed orientation quaternion for each sensor present on the bus at the moment.

Bus data frame structure:

Line 1: explication, it shows the data columns assignment.

Example (shown here with word wrap):

counter 1, raw 9 (acc xyz, gyr xyz, mag xyz), calibrated 10 (acc xyz, gyr xyz, mag xyz, t), quaternion 4 (xyzw), angles 3 (yaw, pitch, roll)

Line 2: data of the first frame.

Example (shown here with word wrap):

```
1156 -0.0595703125000000 -0.0336914062500000 6.0976562500000000 -0.0029296875000000
0.0087890625000000 -0.0131835937500000 -0.1113281250000000 -0.5068359375000000
-0.5087890625000000 -0.0065908099992434 0.0057210992923314 1.0047446346250228
0.0112287766574175 -0.0006608007364095 0.0087612850313011 0.0608967589355951
0.3160917396727632 -0.9256772444307091 -8.6484375000000000 -0.0030875882479594
-0.0034508701820759 -0.0852578111533969 -0.9963481640125250 9.7805791227234469
0.3862368139627212 0.3638420237790572
```

Line 3 and the rest lines: data of the first and subsequent frames.

Units of measurement:

Column 1 - counter. 16-bit unsigned integer roll-over counter of the frames.

Columns 2 to 4 - accelerometer's raw data. Unprocessed data from X, Y and Z channels of the accelerometer. Multiply these values by factor to have them in the same scale as specified in the Table 4.3.2 GetDataR of the OS3D Interface Control Document. The factor is 32768 for the OSv3 and OSv4 types of the sensors, and 2048 for the OSv6.

Columns 5 to 7 - gyroscope's raw data. Unprocessed data from X, Y and Z channels of the gyroscope. Multiply these values by factor to have them in the same scale as specified in the Table 4.3.2 GetDataR of the OS3D Interface Control Document. The factor is 32768 for the OSv3 and OSv4 types of the sensors, and 2048 for the OSv6.

Columns 8 to 10 - magnetometer's raw data. Unprocessed data from X, Y and Z channels of the magnetometer. Multiply these values by factor to have them in the same scale as specified in the Table 4.3.2

GetDataR of the OS3D Interface Control Document. The factor is 32768 for the OSv3 and OSv4 types of the sensors, and 2048 for the OSv6.

Columns 11 to 13 - accelerometer's calibrated data. Calibrated data from X, Y and Z channels of the accelerometer (acceleration along the X,Y and Z axis). Physical quantity measurement unit is G, i.e. 1 = 1G.

Columns 14 to 16 - gyroscope's calibrated data. Calibrated data from X, Y and Z channels of the gyroscope (angular velocity around the X, Y and Z axis). Physical quantity measurement unit is rad/sec, i.e. 1 = 1rad/sec.

Columns 17 to 19 - magnetometer's calibrated data. Calibrated data from X, Y and Z channels of the magnetometer (magnetic induction along the X, Y and Z axis). Physical quantity measurement unit is half of Gauss, i.e. 1 = 0.5Gs.

Column 20 - temperature. Temperature measured by the internal temp. sensor. Multiply these values by 256 to have them in the same scale as specified in the Table 4.3.6 GetDataD of the OS3D Interface Control Document or use the following formulas for conversion to Celsius scale:

For OSv3: $T(^{\circ}\text{C}) = -1.109375 * T + 12$;

For OSv4: $T(^{\circ}\text{C}) = -0.9375 * T + 40$;

For OSv6: $T(^{\circ}\text{C}) = 0.753125 * T + 33$;

Column 21 to 24 - quaternion. The estimated attitude quaternion. The last part is the scalar value. The attitude is given as the body frame with respect to the local East North Up (ENU) frame.

Columns 25 to 27 - Euler's angles. The estimated attitude Yaw, Pitch, and Roll angles measured in degrees. The attitude is given as a 3,1,2 Euler angle sequence describing the body frame with respect to the local East North Up (ENU) frame.

Appendix B - Inertiallabs_sdk_lite.ini file structure

The INI file consists of workset nodes. Each workset node consists of the communication parameters and the sensor-specific parameters. The file structure is as follows:

```
[WorksetNode_n]
<Communication parameters>
<Default filter parameters>
[WorksetNode_n.Sensor_j]
<sensor-specific filter parameters for the sensor #(j+1) in the WorksetNode #n>
```

The sensor-specific parameters, if defined in the sensor-specific sections, will be used in place of the default parameters. If the sensor-specific sections are empty, the default values for the workset node will be used.

Communication parameters:

By default SDK autodetects the device connected via a serial port. But user can point the type of device using the **device** variable. Also it is possible to change the communication interface and the type of requested data. This could be done by listing the desired parameters: **device = <device_type> <interface> <data_type>**

Name	Values	Description
device_type	dongle	USB-RS485 adaptor
	usbcu	SB-CU-USB control unit
	bsmain, emulators, optitrack	Other devices
interface	serial	Use serial communication port
	tcp	Use TCP port
	udp	Use UDP port
data_type	raw	Raw data from sensing elements
	quat	Orientation quaternion
	calibratedt	Calibrated data + temperature
	quadcalibratedt	Orientation quaternion + calibrated data + temperature

By default SDK communicates via the serial port number 1 at baudrate of 1M. User can change these parameters to desired ones by modification of the following fields:

Name	Description
com	Serial communication port number (ex. com = 2)
baudRate	Serial communication port baudrate (ex. baudRate = 4800)
ip	Desired ip for tcp or udp interface (ex. ip = 127.0.0.1)
port	Desired port for tcp or udp interface (ex. port = 2000)

Sensor-specific (filter) parameters:

KpINI = (0..1): the overall gravity reference weight in the filter. The lower the value is the less susceptible to linear accelerations the sensor is. The tradeoff is the settling time after sharp movement: the lower the value is the longer it takes to settle.

ThreshMagINI = (0..1): the threshold for turning the magnetic channel off after the magnitude of the local magnetic field differs from either initial magnitude or moving average magnitude (see `is_mag_new_norm_calc`) by more than this.

MagErrorComp = (0..1): determines how fast the gyroscopic heading gets back to the magnetic after the disturbance goes away. This portion of the difference between the magnetic heading and the gyroscopic is written off on each step. For example if the value is 0.1, that means the difference is multiplied by 0.9 each step. If the value is 0, the heading never gets back to the magnetic (may be useful for head tracking in bad magnetic environment, see `KprMagMaxINI`); if the value is 1, heading snaps to magnetic immediately after the disturbance is gone (see `DelayOnMagDisturbINI`).

DelayOnMagDisturbINI = (0..1000000): the time (in seconds) to keep the magnetic channel off even when magnetic field stabilizes or returns to the initial value after disturbance is detected. After this time passes, the heading gets back to magnetic (see `MagErrorComp` above and `ThreshGyroINI` and `KprMagMaxINI` below).

KprMagINI = (0..1): the magnetic reference weight relative to the gravity reference weight in the filter. If the value is 0, then the magnetic channel is off. If the value is 1 then the weight is the same as the weight of the gravity channel.

ThreshGyroINI = (0..2000): the angular rate (in deg/sec) above which the heading gets back to magnetic at different speed than when the motion is slow (see `KprMagMaxINI` below as compared to `MagErrorComp` above)

KprMagMaxINI = (0..1): same as the `MagErrorComp` but for angular rates above the `ThreshGyroINI`.

Frequency = (60 / 100 / 200 / 500): the filter update rate in frames per second.

TimeStartSec = (0..100): fusion algorithm start-up time in seconds. The gyroscopic biases are written off and the initial magnetic field is calculated during this period. The sensor is "shaking" to indicate that the filter is not yet working. It is recommended to keep the sensor still during this period. If the magnetic field is not updated later on (see `is_mag_new_norm_calc`) then it is a good idea to keep the sensor in a clean magnetic environment during start-up too.

GyroZeroAveragingTimeSec = (0..100): the averaging window (in seconds) for angular rates (to estimate the bias) and for magnetic field magnitude (to detect disturbances).

gyrThreshFormotionDetect = (0...TBD): the threshold of HF changes of angular rate in dps; the sensor is considered as motionless if the current value is lower than specified by this parameter.

magThreshFormotionDetect = (0..TBD): the threshold of HF changes of magnetic field; the sensor is considered as motionless if the current value is lower than specified by this parameter.

PredictionTime = (0..100): the orientation is predicted as far as this time (in milliseconds). Theoretically it can be any value, but it only makes sense as far as the angular rate does not change much. For human motion, the bandwidth is about 20 Hz, which makes 50 the practical limit for this application.

is_mag_channel = (0, 1): determines whether magnetometers are used. 0 – magnetometers are only used for initial setup, 1 – magnetometers are always used. If Head Tracking is limited to orientation (no head translation needed) and no real-world heading is necessary, then the perception may be improved in magnetically challenging areas by setting **is_mag_channel = 0**.

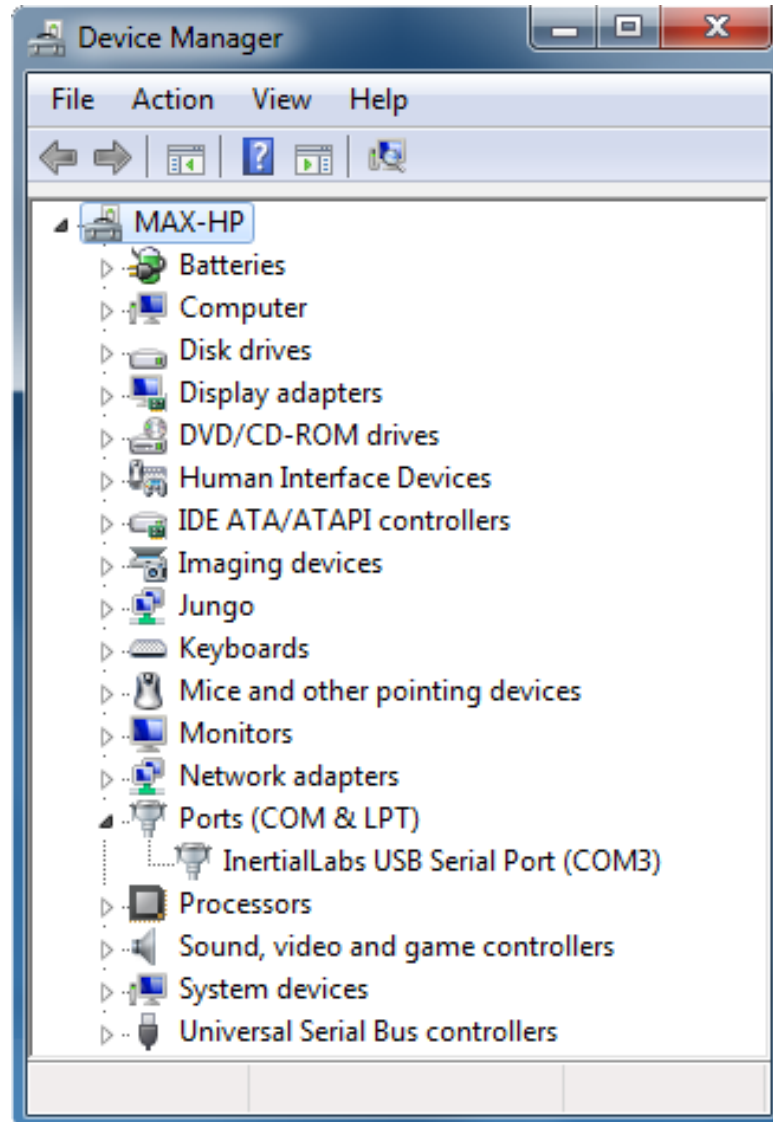
is_mag_new_norm_calc = (0, 1): determine whether long-term changes in magnetic field are taken into account. 0 – only the initial magnetic field is considered normal; 1 – the magnetic field magnitude is averaged over the window defined in GyroZeroAveragingTimeSec

smoothing_mode = (0, 1, 2): This parameter reduces accelerometer noise. This is useful for head tracking. The tradeoff is slower reaction to very dynamic (high-frequency) motion, which is uncommon for head movement. 0 – no smoothing, widest bandwidth; 1 – little smoothing, wide bandwidth; 2 – max smoothing, narrower bandwidth.

smoothingGain = (0..1): this parameter is reasonable if the **smoothing_mode** = 1. The lower values give the more smooth movement.

Appendix C - Checking the serial port number.

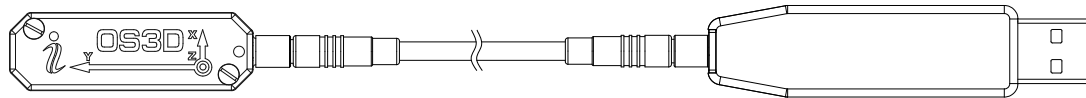
To know the number of the serial port to which the **OS3D** is connected, click "**START**" button, then select "**Control Panel**" and launch the "**Device manager**". In the opened "**Device Manager**" window (see the picture below) you will see the COM port which will be marked as "**InertialLabs USB Serial Port (COM3)**".



Appendix D - The OS3D possible interfaces

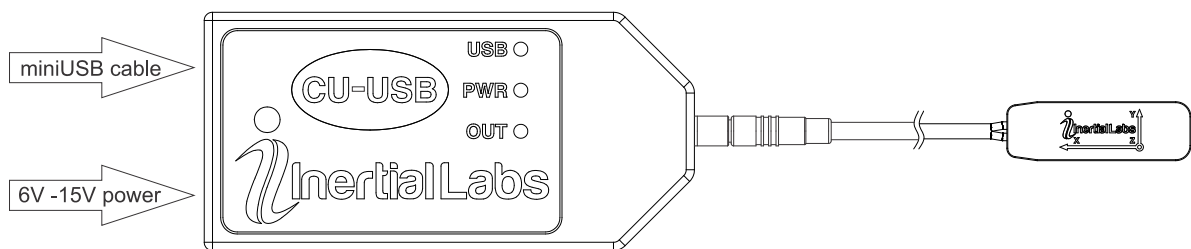
The OS3D sensor has built-in TIA/EIA-485A half-duplex transceiver. In order to connect it to PC the following devices can be used:

- In case of one sensor only, it is recommended to use the USB-RS485 adaptor by Inertial Labs. See the picture below for the connection.



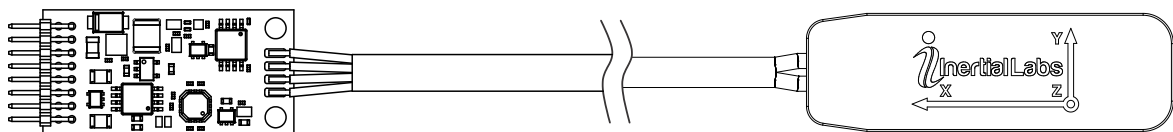
OS3D in aluminum case connected to USB-RS485 adaptor

- In case of multiple sensors connected in a chain, it is recommended to use the SB-CU-USB control unit by Inertial Labs. In this case the separate power supply is required (6V to 15V). It is also provided by Inertial Labs. See the picture below for the connection.



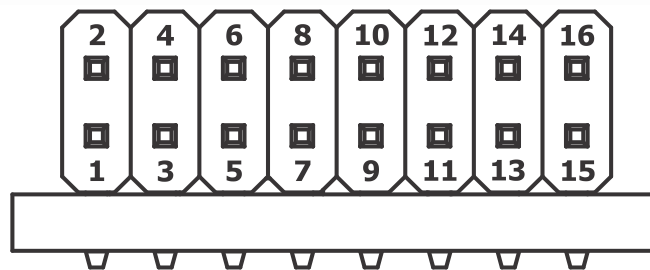
OS3D in plastic case connected to SB-CU-USB control unit

- User can also access the **OS3D** data using the other interfaces like RS232, RS422 and TTL UART using the **Inertial Labs multiple interfaces cable**. The cable also allows to use the wider supply voltage range. In this case the separate power supply is required (5.5V to 42V). User is responsible for providing the power and applying the correct voltage levels to the data lines. See the picture below for the connection.



OS3D in plastic case connected to multiple interfaces cable

No.	Name	Value	Parameters
1	GND	ground, shield	-
2	VIN	input voltage	5.5V to 42V
3	GND	ground, shield	-
4	+5V0	output voltage	5V0@300mA
5	RSVD	Reserved	-
6	485-B	B-RS-485	120 Ohm
7	RSVD	reserved	-
8	485-A	A-RS-485	120 Ohm
9	TTL-TX	TTL UART output	3.3V
10	422-Y	Y-RS-422	120 Ohm
11	TTL-RX	TTL UART input	3.3V max
12	422-Z	Z-RS-422	120 Ohm
13	232-TX	RS232 output	± 15V
14	422-B	B-RS-422	120 Ohm
15	232-RX	RS232 input	± 30V max
16	422-A	A-RS-422	120 Ohm



Multiple interfaces cable pinout (2mm pitch rectangular header)

The rectangular 1.27mm pitch header is used as the connector on this cable. The mating parts are:

- Sullins LPPB082CFFN - Through hole female socket;
- Sullins LPPB082NFSP - Surface mount female socket;
- Samtec FFSD-08-D-06.00-01-N - Female socket to female socket ribbon cable assembly.