Getting started with the X-CUBE-MEMS-XT1 sensor and DSP algorithm software expansion for STM32Cube

Introduction

As well as the on-board sensors supported by the X-CUBE-MEMS1 expansion software package for STM32Cube, the extended X-CUBE-MEMS-XT1 version also supports devices connected via the DIL24 socket.

X-CUBE-MEMS-XT1 runs on the STM32 and includes drivers that recognize the sensors and collect temperature, humidity, pressure and motion data from the A3G4250D, AIS328DQ, AIS3624DQ, H3LIS331DL, HTS221, IIS2DLPC, IIS2MDC, ISM303DAC, ISM330DLC, LIS2DH12, LIS2DW12, LIS2MDL, LIS3MDL, LPS22HB, LPS22HH, LPS25HB, LPS33HW, LSM303AGR, LSM6DS0, LSM6DS3, LSM6DSL, LSM6DSO, LSM6DSR devices.

The expansion is built on STM32Cube software technology to ease portability across different STM32 microcontrollers. The software comes with sample implementations of the drivers running on an X-NUCLEO-IKS01A1 or X-NUCLEO-IKS01A2 expansion board connected to a NUCLEO-F401RE or NUCLEO-L476RG development board.

1 What is STM32Cube?

1.1 What is STM32Cube?

STM32Cube™ represents the STMicroelectronics initiative to make developers' lives easier by reducing development effort, time and cost. STM32Cube covers the STM32 portfolio.

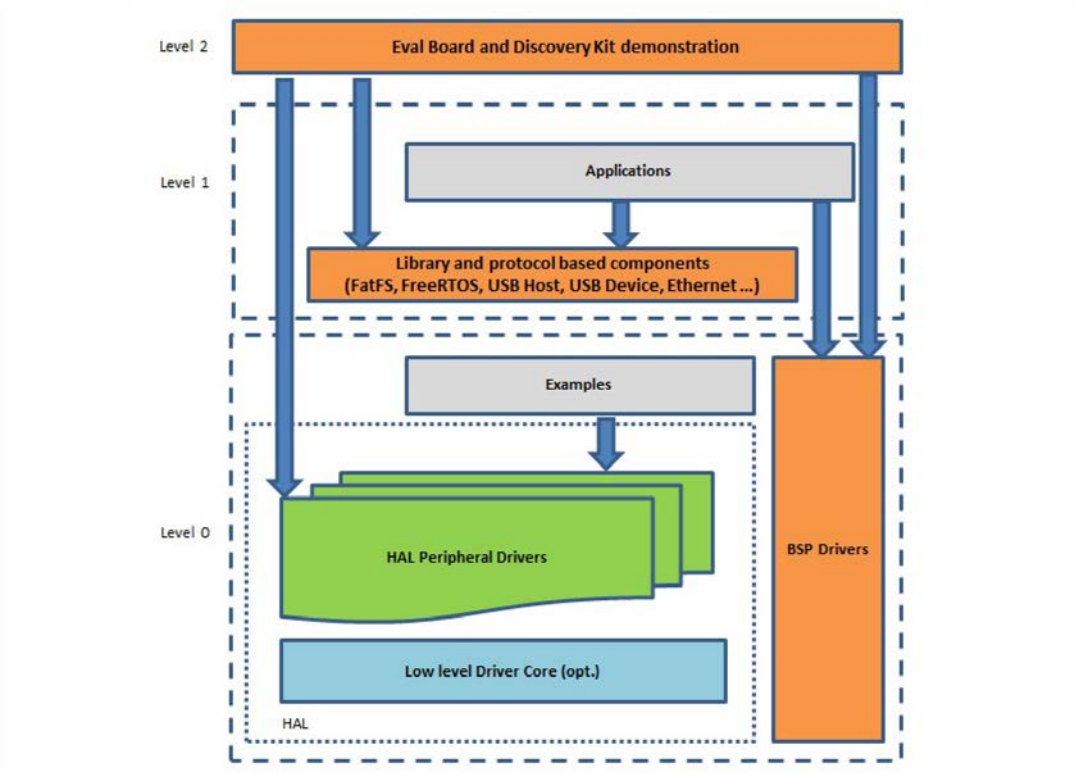
STM32Cube version 1.x includes:

- STM32CubeMX, a graphical software configuration tool that allows the generation of C initialization code using graphical wizards.
- A comprehensive embedded software platform specific to each series (such as the STM32CubeF4 for the STM32F4 series), which includes:
  - the STM32Cube HAL embedded abstraction-layer software, ensuring maximized portability across the STM32 portfolio
  - a consistent set of middleware components such as RTOS, USB, TCP/IP and graphics
  - all embedded software utilities with a full set of examples

1.2 STM32Cube architecture

The STM32Cube firmware solution is built around three independent levels that can easily interact with one another, as described in the diagram below.

Figure 1. Firmware architecture



Level 0: This level is divided into three sub-layers:

- Board Support Package (BSP): this layer offers a set of APIs relative to the hardware components in the hardware boards (Audio codec, IO expander, Touchscreen, SRAM driver, LCD drivers. etc…); it is based on

modular architecture allowing it to be easily ported on any hardware by just implementing the low level routines. It is composed of two parts:

  – Component: is the driver relative to the external device on the board and not related to the STM32, the component driver provides specific APIs to the external components of the BSP driver, and can be ported on any other board.
  – BSP driver: links the component driver to a specific board and provides a set of easy to use APIs. The API naming convention is BSP_FUNCT_Action(): e.g., BSP_LED_Init(), BSP_LED_On().

- Hardware Abstraction Layer (HAL): this layer provides the low level drivers and the hardware interfacing methods to interact with the upper layers (application, libraries and stacks). It provides generic, multi-instance and function-oriented APIs to help offload user application development time by providing ready to use processes. For example, for the communication peripherals (I²C, UART, etc.) it provides APIs for peripheral initialization and configuration, data transfer management based on polling, interrupt or DMA processes, and communication error management. The HAL Drivers APIs are split in two categories: generic APIs providing common, generic functions to all the STM32 series and extension APIs which provide special, customized functions for a specific family or a specific part number.
- Basic peripheral usage examples: this layer houses the examples built around the STM32 peripherals using the HAL and BSP resources only.

Level 1: This level is divided into two sub-layers:

- Middleware components: set of libraries covering USB Host and Device Libraries, STemWin, FreeRTOS, FatFS, LwIP, and PolarSSL. Horizontal interaction among the components in this layer is performed directly by calling the feature APIs, while vertical interaction with low-level drivers is managed by specific callbacks and static macros implemented in the library system call interface. For example, FatFs implements the disk I/O driver to access a microSD drive or USB Mass Storage Class.
- Examples based on the middleware components: each middleware component comes with one or more examples (or applications) showing how to use it. Integration examples that use several middleware components are provided as well.

Level 2: This level is a single layer with a global, real-time and graphical demonstration based on the middleware service layer, the low level abstraction layer and basic peripheral usage applications for board-based functions.

2 X-CUBE-MEMS-XT1 software expansion for STM32Cube

2.1 Overview

The X-CUBE-MEMS-XT1 software package expands STM32Cube functionality.

The key features are:

- Complete software to build applications using temperature and humidity sensors (HTS221 for both X-NUCLEO-IKS01A1 and X-NUCLEO-IKS01A2), pressure sensor (LPS25HB for X-NUCLEO-IKS01A1 and LPS22HB for X-NUCLEO-IKS01A2) and motion sensors (LIS3MDL and LSM6DS0 for X-NUCLEO-IKS01A1 and LSM303AGR and LSM6DSL for X-NUCLEO-IKS01A2), as per X-CUBE-MEMS1
- In addition, the following sensors available through the DIL24 adapter are supported: pressure sensor (LPS22HB for X-NUCLEO-IKS01A1; LPS22HH and LPS33HW for X-NUCLEO-IKS01A2) and motion sensors (LSM6DS3, LSM6DSL and LSM303AGR for X-NUCLEO-IKS01A1; H3LIS331DL, LIS2MDL, LIS2DH12 and LIS2DW12 for both X-NUCLEO-IKS01A1 and X-NUCLEO-IKS01A2; A3G4250D, AIS328DQ, AIS3624DQ, IIS2MDC, ISM303DAC, ISM330DLC, IIS2DLPC, LSM6DSO and LSM6DSR for X-NUCLEO-IKS01A2)
- Easy portability across different MCU families thanks to STM32Cube
- Free user-friendly license terms
- Three sample implementations to transmit real time sensor data to a PC including the Unicleo-GUI application and terminal application support
- Integrated Fast Fourier Transform (FFT) algorithm for vibration analysis
- Signal processing (MotionSP) middleware for vibration analysis in time and frequency domain
- Sample implementation of extended features like FIFO usage, detection of 6D orientation, free-fall, pedometer, single/double tap, tilt, wake-up, sensor hub and self-test

This software gathers temperature, humidity, pressure and motion sensor drivers for the A3G4250D, AIS328DQ, AIS3624DQ, H3LIS331DL, HTS221, IIS2DLPC, IIS2MDC, ISM303DAC, ISM330DLC, LIS2DH12, LIS2DW12, LIS2MDL, LIS3MDL, LPS22HB, LPS22HH, LPS25HB, LPS33HW, LSM303AGR, LSM6DS0, LSM6DS3, LSM6DSL, LSM6DSO and LSM6DSR devices running on STM32 Nucleo.

The package contains DataLogCustomFreeFall6D, DataLogCustomLite and DataLogExtended sample applications that enable sensor data logging on a PC; just visit www.st.com and download the Unicleo-GUI application and the associated documentation.

With this utility, you can choose between various sensors available on the STM32 Nucleo expansion board and the sensor data can be viewed in plots and logged in a user selected file.

Table 1. X-CUBE-MEMS-XT1 sensor support and availability

Sensor name	Availability on X-NUCLEO-IKS01A1	Availability on X-NUCLEO-IKS01A2	Sensor type
LPS25HB	ON-BOARD	N/A	Pressure + Temperature
LPS22HB	DIL24	ON-BOARD	Pressure + Temperature
LPS22HH	N/A	DIL24	Pressure + Temperature
LPS33HW	N/A	DIL24	Pressure + Temperature
HTS221	ON-BOARD	ON-BOARD	Humidity + Temperature
LSM6DS0	ON-BOARD	N/A	Accelerometer + Gyroscope
LSM6DS3	DIL24	N/A	Accelerometer + Gyroscope
LSM6DSL	DIL24	ON-BOARD	Accelerometer + Gyroscope
LSMDSO	N/A	DIL24	Accelerometer + Gyroscope
LSM6DSR	N/A	DIL24	Accelerometer + Gyroscope

Sensor name	Availability on X-NUCLEO-IKS01A1	Availability on X-NUCLEO-IKS01A2	Sensor type
ISM330DLC	N/A	DIL24	Accelerometer + Gyroscope
LIS2DH12	DIL24	DIL24	Accelerometer
LIS2DW12	DIL24	DIL24	Accelerometer
H3LIS331DL	DIL24	DIL24	Accelerometer
AIS328DQ	N/A	DIL24	Accelerometer
AIS3624DQ	N/A	DIL24	Accelerometer
IIS2DLPC	N/A	DIL24	Accelerometer
LSM303AGR	DIL24	ON-BOARD	Accelerometer + Magnetometer
ISM303DAC	N/A	DIL24	Accelerometer + Magnetometer
LIS2MDL	DIL24	DIL24	Magnetometer
LIS3MDL	ON-BOARD	N/A	Magnetometer
IIS2MDC	N/A	DIL24	Magnetometer
A3G4250D	N/A	DIL24	Gyroscope

Note: *See Section 3.3.3.2 How to use LSM303AGR or LIS2MDL together with LIS3MDL on X-NUCLEO-IKS01A1, Section 3.3.3.3 How to use the LIS2MDL, IIS2MDC or ISM303DAC magnetometers on an X-NUCLEO-IKS01A2 expansion board and Section 3.3.3.4 How to use LPS22HH, LSM6DSO or LSM6DSR on an X-NUCLEO-IKS01A2 expansion board.*

2.2 Architecture

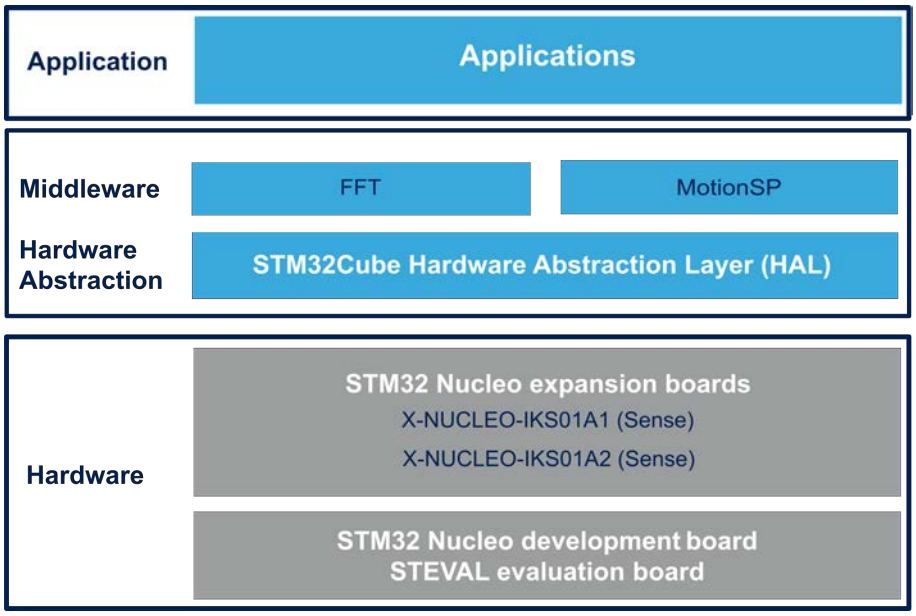
This software is a fully compliant expansion for STM32Cube, enabling development of applications using inertial and environmental sensors.

The software is based on the hardware abstraction layer for the STM32 microcontroller, STM32CubeHAL. The package extends STM32Cube by providing a Board Support Package (BSP) for the sensor expansion board and a sample application for serial communication with a PC.

The software layers used by the application software to access the sensor expansion board are:

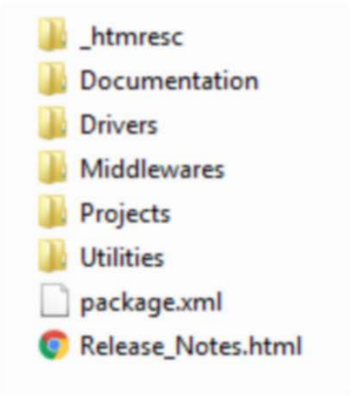
- **STM32Cube HAL layer:** consists of simple, generic and multi-instance set of APIs (application programming interfaces) which interact with the upper layer applications, libraries and stacks. These generic and extension APIs are based on a common framework so that overlying layers like middleware can function without requiring specific microcontroller unit (MCU) hardware information. This structure improves library code reusability and guarantees easy portability across other devices.
- **Board Support Package (BSP) layer:** provides software support for the STM32 Nucleo board peripherals, excluding the MCU. These specific APIs provide a programming interface for certain board specific peripherals like LEDs, user buttons, etc., and can also be used to fetch individual board version information. It also provides support for initializing, configuring and reading data.

Figure 2. **X-CUBE-MEMS-XT1 software architecture**



2.3 Folder structure

Figure 3. **X-CUBE-MEMS-XT1 package folder structure**



The following folders are included in the software package:

- _htmresc: contains resource files for Release_Notes.html.
- Documentation: contains a compiled HTML file detailing the software components and APIs.
- Drivers: contains the HAL drivers, the board specific drivers for each supported board or hardware platform (including the on-board components) and the CMSIS vendor-independent hardware abstraction layer for the Cortex-M processor series.
- Projects: contains a sample application to access sensor data using the NUCLEO-L476RG and NUCLEO-F401RE development boards for the IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM) and SW4STM32 System Workbench for STM32 integrated development environments.
- Utilities: contains the link to Unicleo-GUI PC application necessary for all DataLog-based sample applications.

2.4 APIs

Full function and parameter descriptions for the user APIs can be found in a the compiled X_CUBE_MEMS_XT1.chm HTML file in the package Documentation folder.

2.5 Sample application description

Ready-to-use sample applications for multiple IDEs using the X-NUCLEO-IKS01A1 or X-NUCLEO-IKS01A2 expansion board with the NUCLEO-F401RE or NUCLEO-L476RG development board are provided in the "Projects" directory.

The following table summarizes the sample applications with respect to supported sensors and corresponding X-NUCLEO expansion board.

Table 2. X-CUBE-MEMS-XT1 sample application support

Sample name	Supported sensors on X-NUCLEO-IKS01A1	Supported sensors on X-NUCLEO-IKS01A
DataLogCustomFreeFall6D DataLogExtended DataLogTerminal	H3LIS331DL, HTS221, LIS2DH12, LIS2DW12, LIS2MDL, LIS3MDL, LPS22HB, LPS25HB, LSM303AGR, LSM6DS0, LSM6DS3, LSM6DSL	A3G4250D, AIS328DQ, AIS3624DQ, H3LIS331DL, HTS221, IIS2MDC, IIS2DLPC, ISM303DAC, ISM330DLC, LIS2DH12, LIS2DW12, LIS2MDL, LPS22HB, LPS22HH, LPS33HW, LSM303AGR, LSM6DSL, LSM6DSO, LSM6DSR
DataLogCustomLite	Sensor/Shield-independent	Sensor/Shield-independent
6DOrientation	LSM6DS3	LSM6DSL
FFT_Demo	LIS2DH12, LSM303AGR, LSM6DS0, LSM6DSL	LIS2DH12, LSM303AGR, LSM6DSL
FIFOBypass2ContMode	LSM6DS3	LSM6DSL
FIFOContinuousMode	LSM6DS3	LSM6DSL
FIFODecimation	LSM6DS3	LSM6DSL
FIFOLowPower	LIS2DH12, LSM6DS3	LIS2DH12, LSM6DSL
FIFOMode	LPS22HB, LSM6DS3	LPS22HB, LSM6DSL
FreeFallDetection	LSM6DS3	LSM6DSL
Pedometer	LSM6DS3	LSM6DSL
SelfTest	LSM6DS3	LSM6DSL
SensorHub	N/A	LSM6DSL, LSM303AGR
SingleDoubleTap	LSM6DS3	LSM6DSL
TiltDetection	LSM6DS3	LSM6DSL
VibrationMonitoring	N/A	LSM6DSL
WakeUpDetection	LSM6DS3	LSM6DSL

Note: *See Section 3.3.3.2 How to use LSM303AGR or LIS2MDL together with LIS3MDL on X-NUCLEO-IKS01A1, Section 3.3.3.3 How to use the LIS2MDL, IIS2MDC or ISM303DAC magnetometers on an X-NUCLEO-IKS01A2 expansion board and Section 3.3.3.4 How to use LPS22HH, LSM6DSO or LSM6DSR on an X-NUCLEO-IKS01A2 expansion board.*

In the DataLogCustomFreeFall6D, DataLogCustomLite, DataLogExtended and DataLogTerminal firmware examples, the real-time sensor data is transmitted via serial port to a PC with the `HAL_UART_ Transmit ( )` system call. Transmitted sensor data can be viewed with the Unicleo-GUI application (DataLogCustomFreeFall6D, DataLogCustomLite and DataLogExtended examples) and any terminal application like Teraterm, Hyperterminal, etc. (all the above mentioned examples plus DataLogTerminal example).

The Unicleo-GUI application developed by ST (refer to the user manual on www.st.com)) can be used to read, view and store data from the X-NUCLEO-IKS01A1 or X-NUCLEO-IKS01A2 expansion board connected to a PC via the STM32 Nucleo board. The firmware converts the sensor data into a readable format for the Unicleo-GUI utility.

Sending temperature sensor data via UART would, for example, require the following steps:

1. Initialization: `BSP_TEMPERATURE_ Init ( ...);`
2. Sensor temperature reading: `BSP_TEMPERATURE_Get_ Temp ( ...);`
3. Data serialization: `Serialize();`
4. Data transmission: `HAL_UART_Transmit ();`

The `Serialize( )` function converts the temperature data into a readable format for the Unicleo-GUI utility. Similarly, data from other sensors is also formatted and communicated to the utility. When connected via Teraterm, the user can use the blue button on the STM32 Nucleo expansion board to start and stop the data log.

Most of the DIL24 sensors are automatically detected by the firmware. If a sensor inserted into DIL24 has not a unique I²C address and a value of WHO_AM_I register, it cannot be automatically recognized: thus, the popup window notifies the user to select which sensor is actually connected.

All other firmware examples can be used as template to accelerate development and test every function associated with each sensor. These examples can be easily modified to suit specific scenarios and are designed for easy transfer from currently supported sensors to other sensors in actual applications.

2.5.1 DataLogCustomFreeFall6D and DataLogCustomLite sample application description

These sample applications are designed to explain how to:

- define custom data stream
- define the names and description of custom data displayed in the Unicleo-GUI
- send the data to the Unicleo-GUI for data logging and visualization

There are two types of DataLogCustom firmware:

1. **DataLogCustomFreeFall6D**: data from MEMS and environmental sensors together with custom data are sent to the Unicleo-GUI and demonstrate how to be used for a practical application like displaying FreeFall and 6D orientation data.
2. **DataLogCustomLite**: only custom data are sent to the Unicleo-GUI. This firmware is independent from MEMS and environmental sensors and is fully configurable.

2.5.1.1 *DataLogCustomFreeFall6D*

This sample application shows how to define simple user data and send them to Unicleo-GUI for data logging and visualization. It is designed to use the X-NUCLEO-IKS01A1 or X-NUCLEO-IKS01A2 expansion board with the on-board sensors and/or sensors in DIL24 socket.

The user can display the custom data in the Unicleo-GUI by clicking the **Custom Fields** button in the left vertical toolbar.

The figure below shows the LSM303AGR acceleration in X, Y, Z axes in the graph first section, Free Fall / 6D Orientation data from the LSM6DSL sensor in the **Logic Analyzer** section and Theta/Psi/ Phi tilt angles in the **Custom Int32 and Float Values** section.

It is also designed to show how to acquire data from additional sensor data.

Figure 4. **DataLogCustomFreeFall6D - Unicleo-GUI Custom Fields Plot window**

2.5.1.2 *DataLogCustomLite*

Differently from the previous example, this sample application is designed to be used on any attached expansion board. This allows the user to either use his own expansion board or no expansion board at all; thus, he can send and display any data from the STM32 to the PC with no need to create his own PC application.

Furthermore, it is possible to send binary, integer and float values from the PC to the STM32 microcontroller.

The user can define his own actions in the firmware and control the behavior or adjust parametrs in real-time.

Figure 5. **DataLogCustomLite - Unicleo-GUI main window**

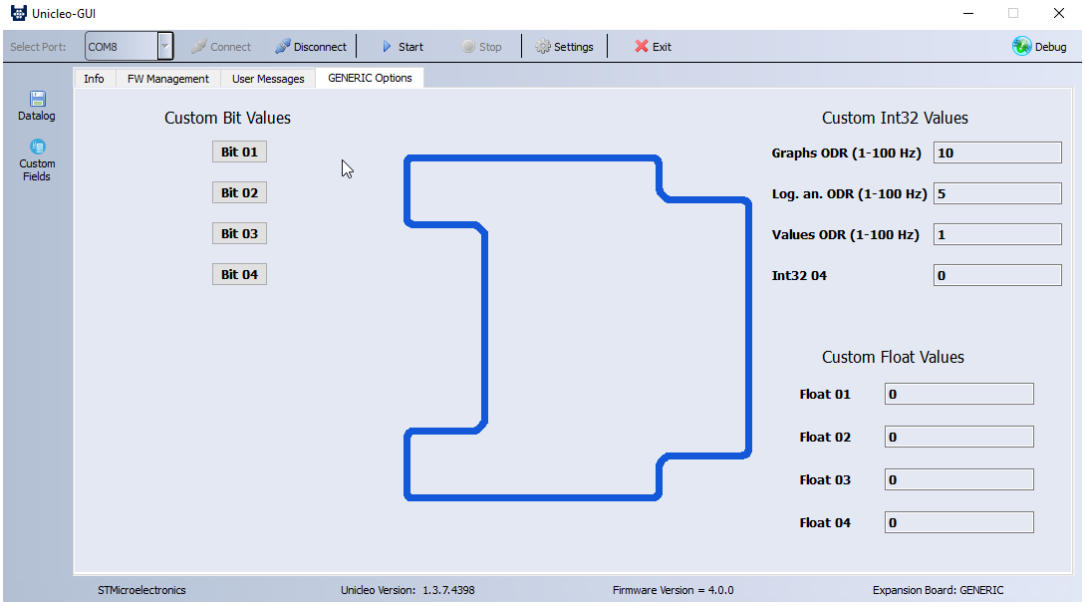
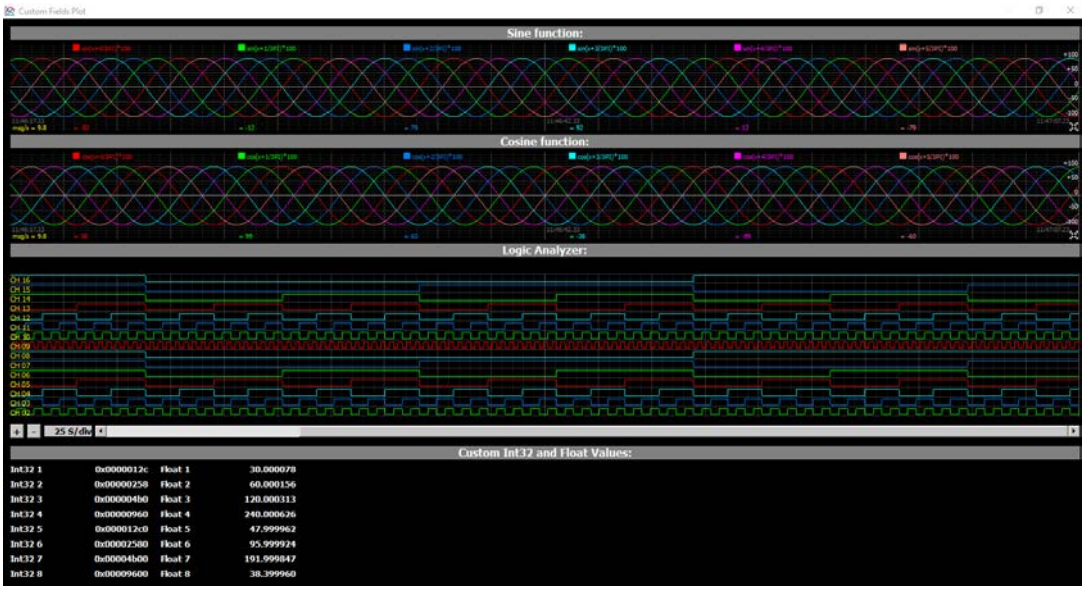


Figure 6. DataLogCustomLite - Unicleo-GUI Custom Fields Plot window shows three shifted sinus functions in the graph first section, the counter in the **Logic Analyzer** section and **Int32 and Float Values** section.

The Unicleo-GUI custom fields also include the **Custom Bit Values** buttons.

Custom Int32 and Float Values fields allow the user to communicate from the PC to the firmware during runtime. It is possible to set separately ODR graphs, ODR logic analyzer and ODR values in real-time using this feature.

Figure 6. **DataLogCustomLite - Unicleo-GUI Custom Fields Plot window**



2.5.1.3 *Custom data sections*

The following sections list the firmware mode capabilities.

2.5.1.3.1 DataLogCustomFreeFall6D

- Graph 1: Up to 3 axes of data visualization
- Graph 2: Up to 3 axes of data visualization
- Logic Analyzer: Up to 16 channels of binary data visualization

- Integers: Up to 4 integer values
- Floats: Up to 4 float values

2.5.1.3.2 DataLogCustomLite

From the firmware to the PC:

- Graph 1: Up to 6 axes of data visualization
- Graph 2: Up to 6 axes of data visualization
- Logic Analyzer: Up to 16 channels of binary data visualization
- Integers: Up to 8 integer values
- Floats: Up to 8 float values

From the PC to the firmware:

- Custom Bits: Up to 4 bit values
- Custom Floats: Up to 4 float values
- Custom Int32s: Up to 4 Int32 values

2.5.1.4 *Msg.Data stream structure*

Custom data has to be filled in an appropriate format into the `Msg.Data[]` stream structure to be sent to the Unicleo-GUI.

It should be done in five different handler functions in the firmware 'main.c' file, which handles the custom data for each format (`Custom_Graph_1_Handler`, `Custom_Graph_2_Handler`, `Logic_Analyzer_Handler`, `Custom_Ints_Handler` and `Custom_Ints_Handler`).

The structure is fixed as shown in the following tables.

Table 3. DataLogCustomFreeFall6D - Msg.Data stream structure

Bytes			Msg.Data stream
Length	From	To	
1	0	0	Destination address (same as received from GUI)
1	1	1	Source address (DEV_ADDR = 0x32)
1	2	2	Command
1	3	3	RTC hours
1	4	4	RTC minutes
1	5	5	RTC seconds (integer part)
1	6	6	RTC seconds (decimal part)
4	7	10	Pressure
4	11	14	Temperature
4	15	18	Humidity
4	19	22	Acceleration - X axis
4	23	26	Acceleration - Y axis
4	27	30	Acceleration - Z axis
4	31	34	Angular velocity - X axis
4	35	38	Angular velocity - Y axis
4	39	42	Angular velocity - Z axis
4	43	46	Magnetic field - X axis
4	47	50	Magnetic field - Y axis
4	51	54	Magnetic field - Z axis

Bytes			Msg.Data stream
Length	From	To	
4	55	58	Custom graph 1 axis 1
4	59	62	Custom graph 1 axis 2
4	63	66	Custom graph 1 axis 3
4	67	70	Custom graph 2 axis 1
4	71	74	Custom graph 2 axis 2
4	75	78	Custom graph 2 axis 3
1	79	79	Logic analyzer (channels 8 - 1)
1	80	80	Logic analyzer (channels 16 - 9)
4	81	84	Custom int32 value 1
4	85	88	Custom int32 value 2
4	89	92	Custom int32 value 3
4	93	96	Custom int32 value 4
4	97	100	Custom float value 1
4	101	104	Custom float value 2
4	105	108	Custom float value 3
4	109	112	Custom float value 4
1	113	113	New data flag bits 1 (bits 5-TEMP 4-HUM 3-PRES 2-MAG 1-GYR 0-ACC)
1	114	114	New data flag bits 2 (bits 4-FLOATS 3-INTS 2-LOG.AN 1-GRAPH2 0-GRAPH1)
		115	Stream Msg.Length

Table 4. DataLogCustomLite - Msg.Data stream structure

Bytes			Msg.Data stream
Length	From	To	
1	0	0	Destination address (same as received from GUI)
1	1	1	Source address (DEV_ADDR = 0x32)
1	2	2	Command
1	3	3	RTC hours
1	4	4	RTC minutes
1	5	5	RTC seconds (integer part)
1	6	6	RTC seconds (decimal part)
4	7	10	Custom graph 1 axis 1
4	11	14	Custom graph 1 axis 2
4	15	18	Custom graph 1 axis 3
4	19	22	Custom graph 1 axis 4
4	23	26	Custom graph 1 axis 5
4	27	30	Custom graph 1 axis 6
4	31	34	Custom graph 2 axis 1
4	35	38	Custom graph 2 axis 2

Bytes			Msg.Data stream
Length	From	To	
4	39	42	Custom graph 2 axis 3
4	43	46	Custom graph 2 axis 4
4	47	50	Custom graph 2 axis 5
4	51	54	Custom graph 2 axis 6
1	55	55	Logic analyzer (channels 8 - 1)
1	56	56	Logic analyzer (channels 16 - 9)
4	57	60	Custom int32 value 1
4	61	64	Custom int32 value 2
4	65	68	Custom int32 value 3
4	69	72	Custom int32 value 4
4	73	76	Custom int32 value 5
4	77	80	Custom int32 value 6
4	81	84	Custom int32 value 7
4	85	88	Custom int32 value 8
4	89	92	Custom float value 1
4	93	96	Custom float value 2
4	97	100	Custom float value 3
4	101	104	Custom float value 4
4	105	108	Custom float value 5
4	109	112	Custom float value 6
4	113	116	Custom float value 7
4	117	120	Custom float value 8
1	121	121	New data flag bits (bits 4-FLOATS 3-INTS 2-LOG.AN 1-GRAPH2 0-GRAPH1)
		122	Msg.Length

2.5.1.5 *Config and String structures*

Together with the custom data, the user has to define the configuration and description of data as graph and axis names, graph units, time axis full scale and position as well as the int and float value names and the logic analyzer channels.

This defining operation has to be done in the *sensor_commands.c* file by filling the predefined message strings called `custom_config[]`, `custom_names[]`, `custom_names2[]` and `custom_names3[]`. The message formats are mandatory and well commented in the code.

The following sections give an overview of the messages.

2.5.1.5.1 custom_config

`custom_config[]` message contains custom configuration numbers that allow to set the data displayed by the Unicleo-GUI. The last three configuration numbers define the parameters the firmware has to receive from the Unicleo-GUI (modified by the user in real-time) and influence the firmware behavior during runtime.

Table 5. DataLogCustomFreeFall6D and DataLogCustomLite - custom_config definition

Bytes			Msg.Data configuration	
Length	From	To	DataLogCustomFreeFall6D	DataLogCustomLite
1	0	0	Destination address (same as received from GUI)	
1	1	1	Source address (DEV_ADDR = 0x32)	
1	2	2	0x50 CMD_SENSOR	
1	3	3	Command	
1	4	4	Sensor type	
1	5	5	Number of graphs (0 - 2)	
1	6	6	Number of graph 1 axes	
			(0 - 3)	(0 - 6)
1	7	7	Number of graph 2 axes	
			(0 - 3)	(0 - 6)
1	8	8	Number of log. an. channels (0 - 16)	
1	9	9	Number of int32 values	
			(0 - 4)	(0 - 8)
1	10	10	Number of float values	
			(0 - 4)	(0 - 8)
1	11	11	Number of awaited bit values (0 - 4)	
1	12	12	Number of awaited int32 values (0 - 4)	
1	13	13	Number of awaited float values (0 - 4)	
		14	Msg.Length configuration	

2.5.1.5.2 custom_names

`custom_names[]` message string 1 contains custom names and parameters delimited by "|".

*Important: Each name or parameter has a variable size but the maximum length of the whole string is **250 bytes**.*

Table 6. DataLogCustomXXX and DataLogCustomLite - custom_names definition

Name or parameter (max. length 250 bytes)	
DataLogCustomFreeFall6D	DataLogCustomLite
Graph 1 name	
Graph 1 axis 1 name	
Graph 1 axis 2 name	
Graph 1 axis 3 name	
Leave empty	Graph 1 axis 4 name
Leave empty	Graph 1 axis 5 name
Leave empty	Graph 1 axis 6 name
Graph 1 units	
Graph 1 full scale	

Name or parameter (max. length 250 bytes)	
DataLogCustomFreeFall6D	**DataLogCustomLite**
Graph 1 integer values with time axis on: 0x01-bottom, 0x03-middle, 0x05-top Graph 1 float values with time axis on: 0x11-bottom, 0x13-middle, 0x15-top	
Graph 2 name	
Graph 2 axis 1 name	
Graph 2 axis 2 name	
Graph 2 axis 3 name	
Leave empty	Graph 2 axis 4 name
Leave empty	Graph 2 axis 5 name
Leave empty	Graph 2 axis 6 name
Graph 2 units	
Graph 2 full scale	
Graph 2 integer values with time axis on: 0x01-bottom, 0x03-middle, 0x05-top Graph 2 float values with time axis on: 0x11-bottom, 0x13-middle, 0x15-top	

2.5.1.5.3 custom_names2

`custom_names2[]` message string 2 contains custom names and parameters delimited by "|".

*Important: Each name or parameter has a variable size but the maximum length of the whole string is **250 bytes**.*

The DataLogCustomFreeFall6D and DataLogCustomLite name or parameter custom_names2 definition are:

- Log. an. CH 1 name
- Log. an. CH 2 name
- Log. an. CH 3 name
- Log. an. CH 4 name
- Log. an. CH 5 name
- Log. an. CH 6 name
- Log. an. CH 7 name
- Log. an. CH 8 name
- Log. an. CH 9 name
- Log. an. CH 10 name
- Log. an. CH 11 name
- Log. an. CH 12 name
- Log. an. CH 13 name
- Log. an. CH 14 name
- Log. an. CH 15 name
- Log. an. CH 16 name
- Awaited bit 1 name
- Awaited bit 2 name
- Awaited bit 3 name
- Awaited bit 4 name
- Awaited int32 1 name
- Awaited int32 2 name
- Awaited int32 3 name
- Awaited int32 4 name

- Awaited float 1 name
- Awaited float 2 name
- Awaited float 3 name
- Awaited float 4 name

2.5.1.5.4 custom_names3

`custom_names3[]` message string 3 contains custom names and parameters delimited by "|".

*Important: Each name or parameter has a variable size but the maximum length of the whole string is **250 bytes**.*

Table 7. DataLogCustomFreeFall6D and DataLogCustomLite - custom_names3 definition

Name or parameter (max. length 250 Bytes)	
DataLogCustomFreeFall6D	DataLogCustomLite
Cust. int32 value 1 name	
Cust. int32 value 1 units	
Cust. int32 value 2 name	
Cust. int32 value 2 units	
Cust. int32 value 3 name	
Cust. int32 value 3 units	
Cust. int32 value 4 name	
Cust. int32 value 4 units	
Leave empty	Cust. int32 value 5 name
Leave empty	Cust. int32 value 5 units
Leave empty	Cust. int32 value 6 name
Leave empty	Cust. int32 value 6 units
Leave empty	Cust. int32 value 7 name
Leave empty	Cust. int32 value 7 units
Leave empty	Cust. int32 value 8 name
Leave empty	Cust. int32 value 8 units
Cust. float value 1 name	
Cust. float value 1 units	
Cust. float value 2 name	
Cust. float value 2 units	
Cust. float value 3 name	
Cust. float value 3 units	
Cust. float value 4 name	
Cust. float value 4 units	
Leave empty	Cust. float value 5 name
Leave empty	Cust. float value 5 units
Leave empty	Cust. float value 6 name
Leave empty	Cust. float value 6 units
Leave empty	Cust. float value 7 name
Leave empty	Cust. float value 7 units

Name or parameter (max. length 250 Bytes)	
DataLogCustomFreeFall6D	DataLogCustomLite
Leave empty	Cust. float value 8 name
Leave empty	Cust. float value 8 units

2.5.2 Sensor Hub sample application description

This sample application is designed as a template showing how to set up and use the LSM6DSL **Sensor Hub** functionality.

It has been implemented for the X-NUCLEO-IKS01A2 MEMS Shield board for the simple hardware switching possibility. This example uses LSM6DSL and LSM303AGR on-board sensors to simulate the 9-axis device returning accelerometer, gyroscope and magnetometer data to any PC terminal application like Tera Term and similar.

The X-NUCLEO-IKS01A2 hardware configuration has to be changed from **Default** to **Sensor Hub** as shown in the following table:

Table 8. X-NUCLEO-IKS01A2 LSM6DSL Sensor Hub hardware configuration

Jumper	Default	Sensor Hub
JP7	1-2 3-4 (I2C1=I2C2, I2Cx=GND)	2-3 (I2C1=I2Cx)
JP8	1-2 3-4 (I2C1=I2C2, I2Cx=GND)	2-3 (I2C1=I2Cx)

2.5.3 FFT_Demo sample application description

The FFT_Demo provides frequency analysis of acceleration data using Fast Fourier Transform (FFT) algorithm, to detect vibration from devices such as motors, fans and pumps.

FFT_Demo project supports the following sensors:

- Accelerometer on the X-NUCLEO-IKS01A1 expansion board:
  - LSM6DS0
- Accelerometers on the X-NUCLEO-IKS01A2 expansion board:
  - LSM303AGR
  - LSM6DSL
- Evaluation boards plugged to X-NUCLEO-IKS01A1 or X-NUCLEO-IKS01A2 DIL24 socket:
  - STEVAL-MKI172V1 (LSM303AGR) (X-NUCLEO-IKS01A1).
  - STEVAL-MKI178V1 (LSM6DSL) (X-NUCLEO-IKS01A1 or X-NUCLEO-IKS01A2)
  - STEVAL-MKI168V1 (IIS2DH) (X-NUCLEO-IKS01A1 or X-NUCLEO-IKS01A2)
  - STEVAL-MKI135V1 (LIS2DH) (X-NUCLEO-IKS01A1 or X-NUCLEO-IKS01A2)
  - STEVAL-MKI151V1 (LIS2DH12) (X-NUCLEO-IKS01A1 or X-NUCLEO-IKS01A2)

There are two PC applications that can be used with the FFT_Demo: Unicleo-GUI or Terminal.

2.5.3.1 *Unicleo-GUI*

The FFT_Demo firmware is automatically detected by the Unicleo-GUI; dedicated options and a **Spectrum plot** are available to select a sensor and display data.

Figure 7. **FFT_Demo Unicleo-GUI - Options tab**

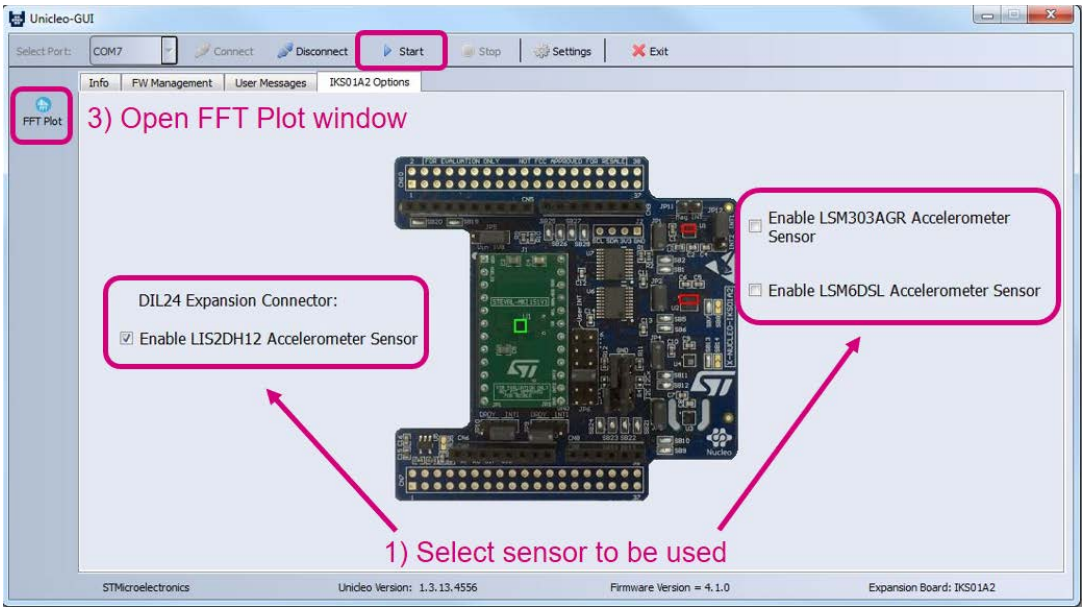
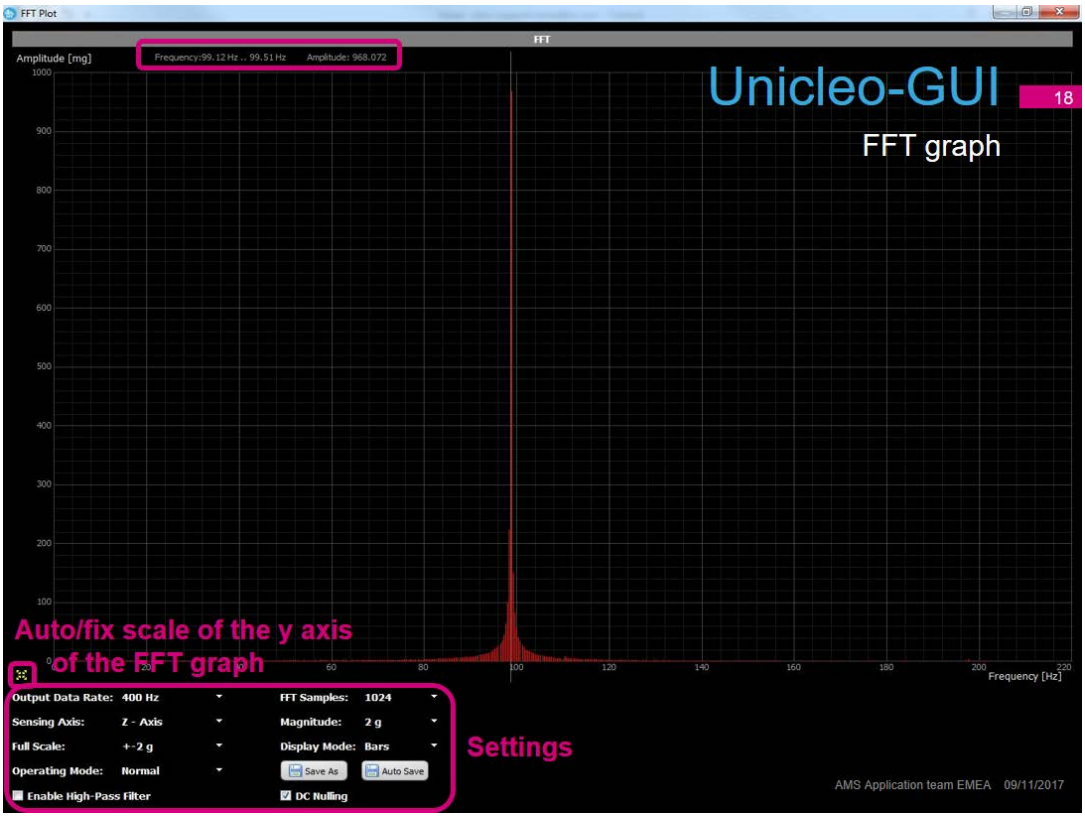
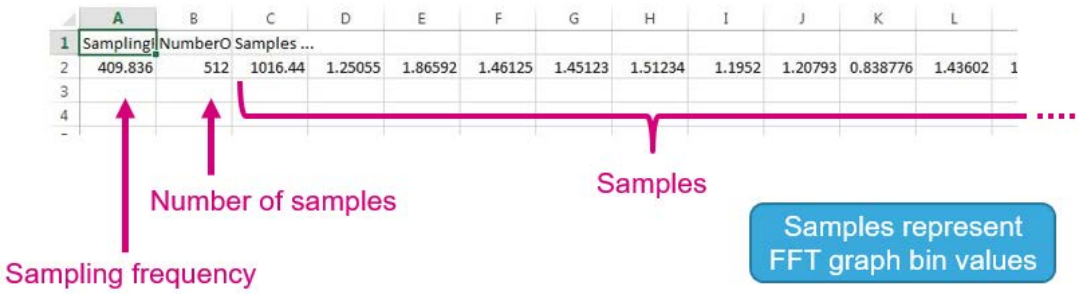


Figure 8. **FFT_Demo Unicleo-GUI - Spectrum plot**



Data can be saved as .csv file and directly opened in Excel (or similar).

Figure 9. **FFT_Demo interpretation of the saved data**



To calculate frequency of each bin use the following formula:

$$a_n = n \cdot \frac{Sampling\ frequency}{2 \cdot Number\ of\ samples} \quad (1)$$

where a_n is the *n*-th bin of the graph and

$$n = 0,1,2\ldots,(Number\ of\ samples - 1) \quad (2)$$

2.5.3.2 *Terminal*

The user can select a menu option by simply pressing its number on the PC keyboard.

Pressing **m** or **M** on the keyboard, the **Main menu** appears.

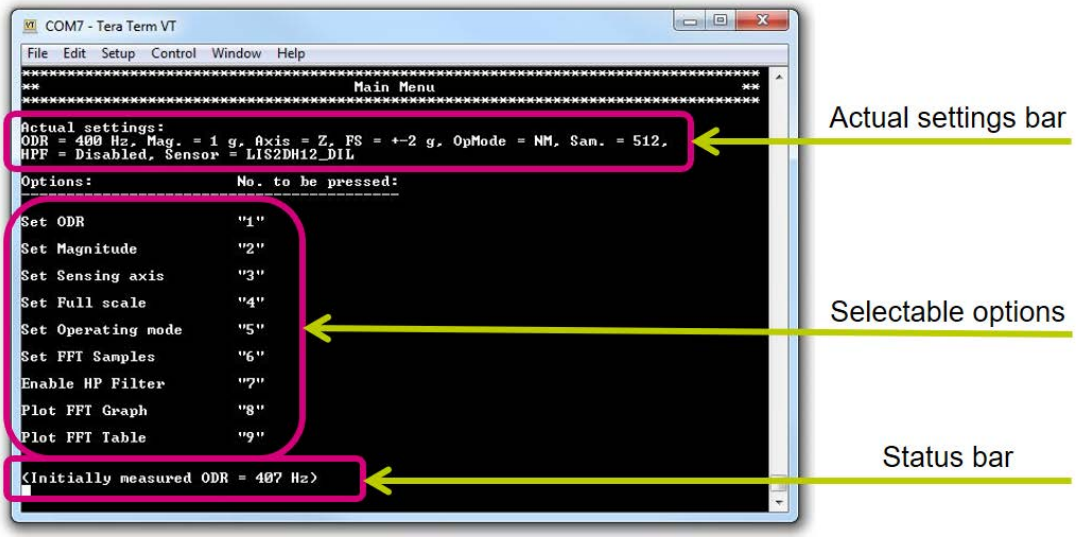
Pressing **h** or **H** on the keyboard the following two options are switched in the main menu:

- enable HP filter/disable HP filter (allows enabling or disabling the Terminal AXL internal high pass filter)
- enable DC nulling/disable DC nulling (allows enabling or disabling the nulling of 0 Hz part in the calculated frequency spectrum – earth gravity)

The sensor in use can be switched by pressing **s** or **S** on the keyboard.

Options not allowed in the actual sensor setup are disabled and marked by * in the menu.

Figure 10. **FFT_Demo Terminal mode - main menu**



The signal maximum frequency that can be displayed in the FFT graph is ODR/2.

The signal minumum magnitude that can be displayed is 1% of the maximum graph magnitude.

Under the FFT graph, there are the following info bars:

- The maximum calculated frequency in the whole displayable frequency range and its value
- The maximum frequency value calculated by using the Flat top window(1)
- The maximum calculated bin frequency displayed in the FFT graph(2)

- The input AC signal maximum amplitude(3)

1. *Only when the HP filter is enabled*
2. *Together with the bin range.*
3. *Maximum and minimum values.*

The number of FFT samples (256, 512, 1024) is transformed into 51 bins displayed in the graph.
The graph bin contains the interval of frequencies and displays only the frequency biggest magnitude in that interval.

$$Graph\ refresh\ rate \geq \frac{FFT\ samples}{ODR}\left(e.g.\ \frac{512}{400\ Hz} = 1.28\ seconds\right) \quad (5)$$

Figure 11. FFT_Demo Terminal mode - spectrum plot

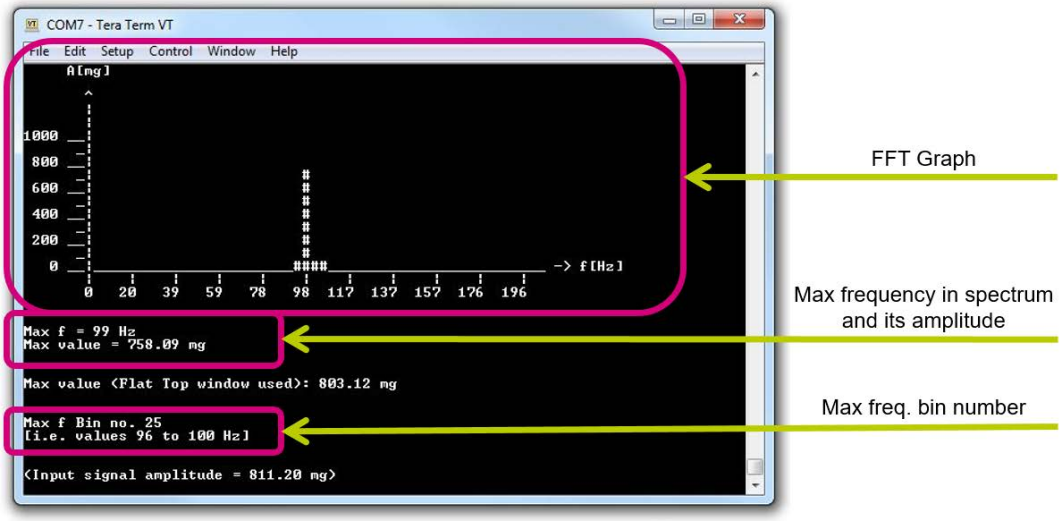


Figure 12. FFT_Demo Terminal mode - signal information

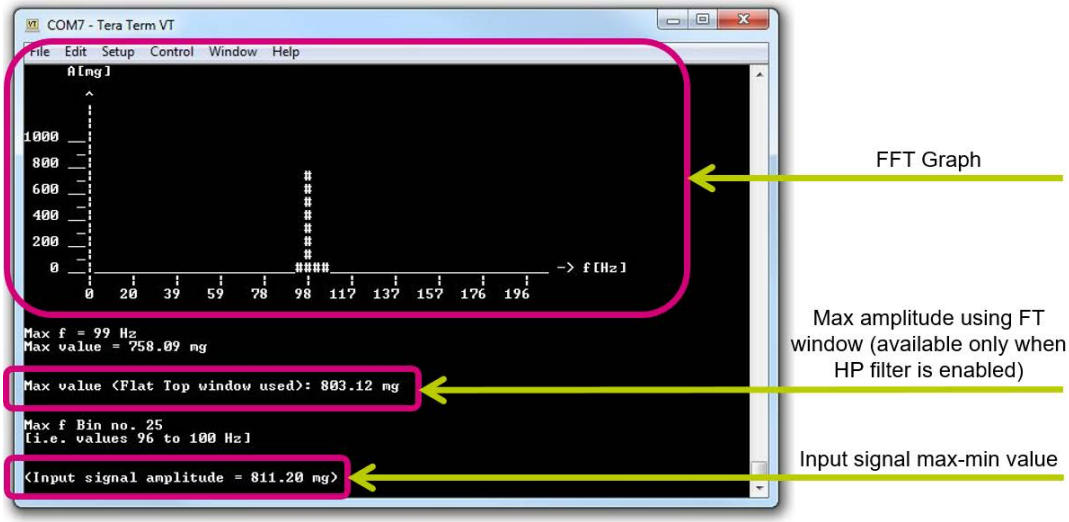


Figure 13. **FFT_Demo Terminal mode - output data**

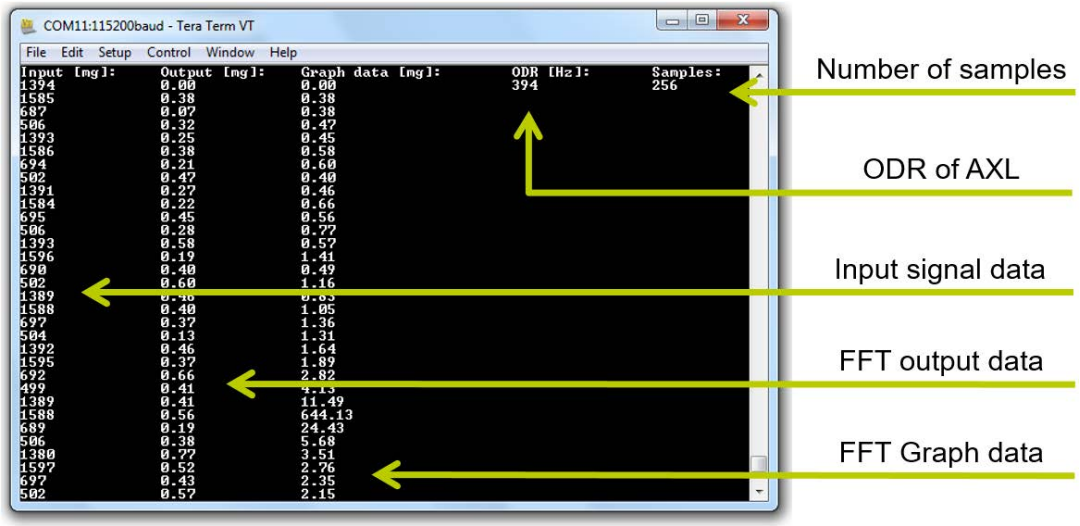


Figure 14. **FFT_Demo Terminal mode - sensor selection**

2.5.3.3 *Vibration monitoring sample application description*

This sample implementation uses Signal Processing (MotionSP) library and performs analysis of acceleration data to detect vibration from devices such as motors, fans and pumps.

The library features:

- time and frequency domain analysis
- acceleration RMS, speed RMA and acceleration peak value calculation
- analyses performed for all 3 axes at the same time
- selectable window for FFT analysis

Frequency domain analysis uses Fast Fourier Transform (FFT) algorithm and displays the result in separate plots for X, Y and Z axis.

Time domain analysis calculates Acceleration RMS, Acceleration Peak and Speed RMS values which are displayed below FFT plots. Acceleration Peak can be reset by pressing the red X button next to the value to restart calculating the result from zero.

You can set the accelerometer output data rate and full scale, number of FFT samples, magnitude, etc., and also choose whether to leave the input data unchanged by selecting rectangular window or to multiply the acceleration data by either hanning, hamming or flat top window to avoid spectral leakage.

Vibration Monitoring sample application supports only the LSM6DSL sensor on the X-NUCLEO-IKS01A2 expansion board together with Unicleo-GUI Windows PC application.

Figure 15. Vibration Monitoring - Unicleo-GUI spectrum plots, settings and calculated results

3 System setup guide

3.1 Hardware description

3.1.1 STM32 Nucleo platform

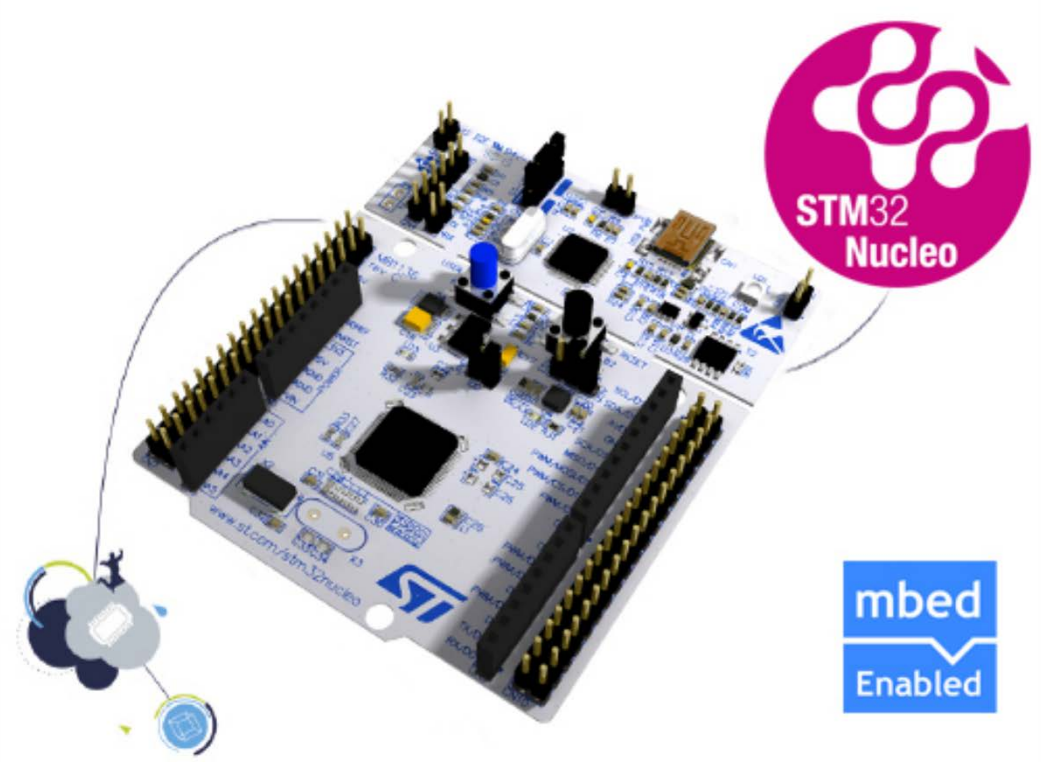
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

The Arduino™ connectivity support and ST morpho connectors make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide range of specialized expansion boards to choose from.

The STM32 Nucleo board does not require separate probes as it integrates the ST-LINK/V2-1 debugger/ programmer.

The STM32 Nucleo board comes with the comprehensive STM32 software HAL library together with various packaged software examples.

Figure 16. STM32 Nucleo board



Information regarding the STM32 Nucleo board is available at www.st.com/stm32nucleo

3.1.2 X-NUCLEO-IKS01A1 expansion board

The X-NUCLEO-IKS01A1 is a sensor expansion board for the STM32 Nucleo board. It is also compatible with Arduino UNO R3 connector layout and is designed around humidity (HTS221), pressure (LPS25HB) and motion (LIS3MDL and LSM6DS0) sensing devices. The X-NUCLEO-IKS01A1 interfaces with the STM32 MCU via the I²C pin, and the user can change the default I²C port and the device IRQ by changing a resistor on the evaluation board.

You can attach one among H3LIS331DL, LIS2DH12, LPS22HB, LSM303AGR, LSM6DS3 or LSM6DSL DIL24 expansion components to be used in place of the on-board sensors.

See Section 3.3.3.2 How to use LSM303AGR or LIS2MDL together with LIS3MDL on X-NUCLEO-IKS01A1, Section 3.3.3.3 How to use the LIS2MDL, IIS2MDC or ISM303DAC magnetometers on an X-NUCLEO-IKS01A2 expansion board and Section 3.3.3.4 How to use LPS22HH, LSM6DSO or LSM6DSR on an X-NUCLEO-IKS01A2 expansion board.

Figure 17. X-NUCLEO-IKS01A1 expansion board

3.1.3 X-NUCLEO-IKS01A2 expansion board

The X-NUCLEO-IKS01A2 is a motion MEMS and environmental sensor expansion board for STM32 Nucleo.

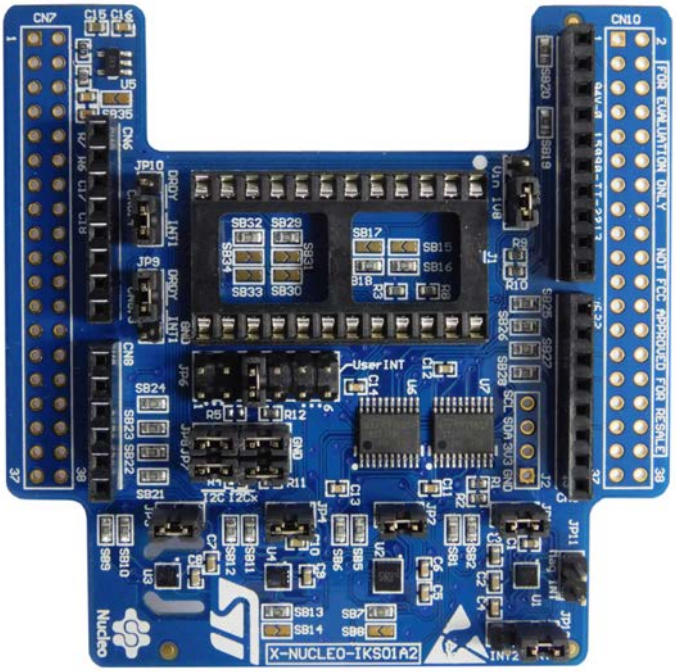
It is compatible with the Arduino UNO R3 connector layout, and is designed around the LSM6DSL 3D accelerometer and 3D gyroscope, the LSM303AGR 3D accelerometer and 3D magnetometer, the HTS221 humidity and temperature sensor and the LPS22HB pressure sensor.

The X-NUCLEO-IKS01A2 interfaces with the STM32 microcontroller via the I²C pin, and it is possible to change the default I²C port.

You can attach one of the A3G4250D, AIS328DQ, AIS3624DQ, H3LIS331DL, IIS2DLPC, IIS2MDC, ISM303DAC, ISM330DLC, LIS2DH12, LIS2DW12, LIS2MDL, LPS22HH, LPS33HW, LSM6DSL, LSM6DSO and LSM6DSR DIL24 expansion components to be used in place of the on-board sensors.

See Section 3.3.3.2 How to use LSM303AGR or LIS2MDL together with LIS3MDL on X-NUCLEO-IKS01A1, Section 3.3.3.3 How to use the LIS2MDL, IIS2MDC or ISM303DAC magnetometers on an X-NUCLEO-IKS01A2 expansion board and Section 3.3.3.4 How to use LPS22HH, LSM6DSO or LSM6DSR on an X-NUCLEO-IKS01A2 expansion board.

Figure 18. **X-NUCLEO-IKS01A2 MEMS and environmental sensor expansion board**



3.2 Software description

The following software components are required for a suitable development environment for creating applications with the STM32 Nucleo development board equipped with the sensor expansion board:

- X-CUBE-MEMS-XT1: an STM32Cube expansion for sensor application development. The X-CUBE-MEMS-XT1 and associated documentation is available on www.st.com
- One of the following supported development tool-chains and compilers:
  - IAR Embedded Workbench for ARM® (IAR-EWARM) toolchain + ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM-STM32) toolchain + ST-LINK
  - SW4STM32 System Workbench for STM32 + ST-LINK

3.3 Hardware and software setup

This section describes the hardware and software setup procedures. It also describes the system setup needed for the above.

3.3.1 Hardware setup

The following hardware components are required:

1. One STM32 Nucleo development platform (order code: NUCLEO-F401RE or NUCLEO-L476RG)
2. One sensor expansion board (order code: X-NUCLEO-IKS01A1 or X-NUCLEO-IKS01A2)
3. One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC

3.3.2 Software setup

This section lists the minimum requirements for the developer to setup the SDK, run the sample testing scenario based on the Unicleo-GUI utility and customize applications.

3.3.2.1 *Development Tool-chains and compilers*

Select one of the Integrated Development Environments supported by the STM32Cube expansion software and follow the system requirements and setup information provided by the same IDE provider.

3.3.2.2 *PC utility*

The Unicleo-GUI application for PC has following minimum requirements:

- PC running Windows 7 or higher
- At least 2 GB RAM
- USB port
- 40 MB HD space

3.3.3 System setup guide

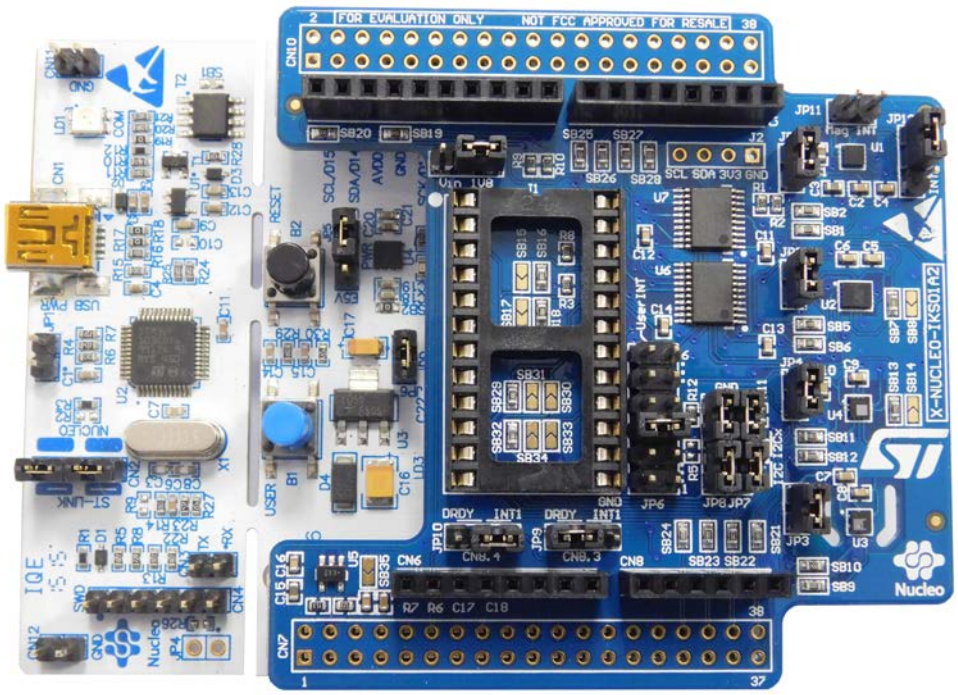
This section describes how to setup different hardware components before writing and executing an application on theSTM32 Nucleo board with the sensors expansion board.

3.3.3.1 *STM32 Nucleo and sensor expansion boards setup*

The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer. Developers can download the relevant version of the ST-LINK/V2-1 USB driver at STSW-LINK007 or STSW-LINK009 (according to the Microsoft Windows OS version).

The X-NUCLEO-IKS01A1 and X-NUCLEO-IKS01A2 sensor expansion boards can be easily connected to the STM32 Nucleo board through the Arduino UNO R3 extension connector and can interface with the external STM32 microcontroller on STM32 Nucleo via the Inter-Integrated Circuit (I²C) transport layer.

Figure 19. X-NUCLEO-IKS01A2 expansion board on STM32 Nucleo development board



3.3.3.2 *How to use LSM303AGR or LIS2MDL together with LIS3MDL on X-NUCLEO-IKS01A1*

Using the LSM303AGR or LIS2MDL magnetometer in DIL24 socket together with the X-NUCLEO-IKS01A1 expansion board on-board LIS3MDL magnetometer requires the following procedure.

Step 1. Disconnect the SB3 solder bridge on the X-NUCLEO-IKS01A1.

Step 2. Connect the SB4 solder bridge on the X-NUCLEO-IKS01A1.

Step 3. Change the `LIS3MDL_MAG_I2C_ADDRESS_HIGH` value to `LIS3MDL_MAG_I2C_ADDRESS_LOW` in the *\Drivers\BSP\X_NUCLEO_IKS01A1\ x_nucleo_iks01a1_magneto.c* file inside the `BSP_LIS3MDL_MAGNETO_Init()` function.

Step 4. Compile and download the firmware to the STM32 Nucleo development board.

3.3.3.3 *How to use the LIS2MDL, IIS2MDC or ISM303DAC magnetometers on an X-NUCLEO-IKS01A2 expansion board*

Using the LIS2MDL, IIS2MDC or ISM303DAC magnetometers in DIL24 socket on an X-NUCLEO-IKS01A2 expansion board requires SB1 and SB2 solder bridge disconnection.

The LSM303AGR on-board sensor has to be disconnected from the I²C bus, because it has the same I²C address with the above mentioned magnetometers.

3.3.3.4 *How to use LPS22HH, LSM6DSO or LSM6DSR on an X-NUCLEO-IKS01A2 expansion board*

Using the LPS22HH, LSM6DSO or LSM6DSR sensors in DIL24 socket on an X-NUCLEO-IKS01A2 expansion board requires pulling the INT1 pin of the device to the GND during power up (connecting the STM32 Nucleo board to the USB) to enable I²C communication (instead of I^3C).

This can be done by connecting jumper wire between any GND pin (e.g. Arduino connector CN6.6, CN6.7 or CN5.7) and INT1 pin (Arduino connector CN9.5).

After power up, this connection has to be removed.

3.3.3.5 *Unicleo-GUI setup*

The Unicleo-GUI is a graphical user interface that can be used to interact and obtain data from sensors on the expansion board, and display acquired sensor data in tables and graphs; download the user manual from www.st.com for installation and usage instructions.

Revision history

Table 9. **Document revision history**

Date	Version	Changes
01-Dec-2016	1	Initial release.
12-Jun-2016	2	Updated Section "Introduction", Section 2.1: "Overview", Figure 2: "X-CUBE-MEMS-XT1 software architecture", Section 2.3: "Folder structure", Section 2.5: "Sample application description" Added Section 2.5.1: "DataLogCustomDemo, DataLogCustomFreeFall6D and DataLogCustomLite sample application description" and Section 2.5.2: "Sensor Hub sample application description"
15-Nov-2017	3	Updated title, Section 2.1: "Overview", Figure 2. "X-CUBE-MEMS-XT1 software architecture", Section 2.5: "Sample application description", Section 2.5.1: "DataLogCustomFreeFall6D and DataLogCustomLite sample application description". Added DataLogCustomDemo, Section 2.5.1.1: "DataLogCustomFreeFall6D", Section 2.5.1.2: "DataLogCustomLite", Section 2.5.1.3: "Custom data sections", Section 2.5.1.3.1: "DataLogCustomFreeFall6D", Section 2.5.1.3.2: "DataLogCustomLite", Section 2.5.1.4: "Msg.Data stream structure", Section 2.5.1.5: "Config and String structures", Section 2.5.1.5.1: "custom_config", Section 2.5.1.5.2: "custom_names", Section 2.5.1.5.3: "custom_names2", Section 2.5.1.5.4: "custom_names3", Section 2.5.3: "FFT_Demo sample application description", Section 2.5.3.1: "Unicleo-GUI" and Section 2.5.3.2: "Terminal"
26-Jan-2018	4	Updated Section Introduction, Section 2.1 Overview, Section 2.5 Sample application description, Section 2.5.1 DataLogCustomFreeFall6D and DataLogCustomLite sample application description, Section 2.5.1.1 DataLogCustomFreeFall6D, Section 3.1.2 X-NUCLEO-IKS01A1 expansion board and Section 3.1.3 X-NUCLEO-IKS01A2 expansion board. Added Section 3.3.3.3 How to use LSM303AGR or LIS2MDL together with LIS3MDL on X-NUCLEO-IKS01A1 and Section 3.3.3.2 How to use the LIS2MDL, IIS2MDC or ISM303DAC magnetometers on an X-NUCLEO-IKS01A2 expansion board.
12-Sep-2018	5	Updated Section Introduction, Section 2.1 Overview, Figure 2. X-CUBE-MEMS-XT1 software architecture, Section 2.5 Sample application description, Section 3.1.2 X-NUCLEO-IKS01A1 expansion board and Section 3.1.3 X-NUCLEO-IKS01A2 expansion board. Added Section 2.5.3.3 Vibration monitoring sample application description and Section 3.3.3.4 How to use LPS22HH, LSM6DSO or LSM6DSR on an X-NUCLEO-IKS01A2 expansion board.

Contents

List of tables

List of figures