Introduction

The X-CUBE-MEMS1 expansion software package for STM32Cube runs on the STM32 and includes drivers that recognize the sensors and collect temperature, humidity, pressure and motion data from the HTS221, LPS22HB, LSM6DSL and LSM303AGR devices.

The expansion is built on STM32Cube software technology to ease portability across different STM32 microcontrollers.

The software comes with a sample implementation of the drivers running on the X-NUCLEO-IKS01A2 expansion board connected to a featured STM32 Nucleo development board.

The software provides sample applications and advanced motion libraries (MotionAC, MotionAR, MotionAT, MotionAW, MotionCP, MotionEC, MotionFA, MotionFD, MotionFX, MotionGC, MotionGR, MotionID, MotionMC, MotionPE, MotionPM, MotionPW, MotionSD, MotionSM and MotionTL).
1. **What is STM32Cube?**

1.1 **STM32Cube overview**

STMCube™ is an STMicroelectronics initiative aimed at making life easier for the developer by reducing development effort, time and cost for the STM32 range of products. Version 1.x includes:

- The STM32CubeMX graphical software configuration tool to generate C initialization code using graphical wizards.
- A comprehensive embedded software platform for each series (e.g., STM32CubeF4 for the STM32F4 series)
  - the embedded STM32Cube HAL abstraction layer software which maximizes portability across the STM32 portfolio
  - a consistent set of middleware components, including RTOS, USB, TCP/IP and graphics
  - all the embedded software utilities come with a full set of examples


1.2 **STM32Cube architecture**

The STM32Cube firmware solution is based on three independent levels that freely interact with each other, as shown below:

![Figure 1. Firmware architecture](image)

**Level 0** is divided into three sub-layers:
The Board Support Package (BSP) layer offers a set of board hardware APIs (audio codec, IO expander, touchscreen, SRAM driver, LCD drivers, etc.) based on modular architecture which can be rendered compatible with any hardware by simply running the low-level routines. The BSP has two parts:

- **component**: the driver associated with the external device on the board (not the STM32); the component driver provides specific APIs to the BSP driver external components and can be ported to any other board.
- **BSP driver**: links the component driver to a specific board and provides a set of user-friendly APIs. The naming rule of the APIs is `BSP_FUNCT_Action()`: ex. `BSP_LED_Init()`, `BSP_LED_On()`

The Hardware Abstraction Layer (HAL) 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 which render user applications unnecessary by providing ready to use processes. For example, it provides APIs for the communication peripherals (I²S, UART, etc.) for initialization and configuration, data transfer management based on polling, interrupts or DMA processes, and management of any communication errors. There are two types of HAL driver APIs:

- **generic APIs** which provide common and generic functions to the entire STM32 series
- **extension APIs** which provide specific, customized functions for a particular family or a certain part number

Basic peripheral usage examples: this layer includes the examples built for the STM32 peripheral using the HAL and BSP resources only.

**Level 1** is divided into two sub-layers:

- **Middleware components**: a set of libraries covering USB host and device libraries STemWin, FreeRTOS, FatFS, LwIP, and PolarSSL. Horizontal interaction between layer components is handled directly by calling the feature APIs, while vertical interaction with the low level drivers is managed through specific callbacks and static macros implemented in the library system call interface. For example, the FatFs accesses the microSD drive or the USB mass storage class via the disk I/O driver.
- **Middleware examples (or applications)** for individual components as well as integration examples across several middleware components are provided.

**Level 2** is a single layer providing a global, real-time and graphical demonstration based on the middleware service layer, the low-level abstraction layer and basic peripheral usage applications involving board functions.
2 X-CUBE-MEMS1 software expansion for STM32Cube

2.1 Overview

The X-CUBE-MEMS1 software package expands the STM32Cube functionality. The key features are:

- Complete software to build applications for X-NUCLEO-IKS01A2 using temperature and humidity sensors (HTS221), pressure sensor (LPS22HB) and motion sensors (LSM303AGR and LSM6DSL)
- Several examples to show the innovative inertial and environmental sensors
- Sample application to transmit real-time sensor data to a PC
- Compatible with the Unicleo-GUI graphical user interface to display sensor data and configure outputs
- Sample implementation available on the X-NUCLEO-IKS01A2 board connected to a NUCLEO-F401RE, NUCLEO-L152RE, NUCLEO-L476RG or NUCLEO-L053R8 development board
- Advanced motion libraries with sample applications
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

This software gathers the temperature, humidity, pressure and motion sensor drivers for the HTS221, LPS22HB, LSM6DSL and LSM303AGR devices running on STM32 Nucleo.

The package includes several sample applications that the developer can use to start experimenting with the code. A sample application has been developed to enable sensor data logging on a PC; a Windows PC utility (Unicleo-GUI) is available on www.st.com, to allow the developer choose among various sensors available on the expansion board and set the appropriate delay/interval among consecutive data points.

2.2 Architecture

This software is a fully compliant expansion for STM32Cube enabling development of applications using inertial and environmental sensors. See the previous chapter for a brief explanation of the STM32Cube architecture.

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:

- The STM32Cube HAL driver layer provides a simple, generic and multi-instance set of APIs (application programming interfaces) to interact with the upper layers (application, libraries and stacks). It includes generic and extension APIs and is based on a generic architecture which allows the layers built on it (such as the middleware layer) to implement their functionalities without dependence on the specific hardware configuration of a given Microcontroller Unit (MCU). This structure improves library code reusability and guarantees high portability across other devices.
- The Board Support Package (BSP) layer provides supporting software for the peripherals on the STM32 Nucleo board, except for the MCU. It has a set of APIs to provide a programming interface for certain board-specific peripherals (e.g. the LED, the user button etc.) and allow identification of the specific board version. For the sensor expansion board, it provides the programming interface for various inertial and environmental sensors and provides support for initializing and reading sensor data.
- The Middleware provides advanced motion libraries. The motion libraries include MotionAC (accelerometer calibration library), MotionAR (activity recognition library), MotionAT (active time library), MotionAW (activity recognition for wrist library), MotionCP (carrying position library), MotionEC (eCompass library), MotionFA (fitness activities library), MotionFD (fall detection library), MotionFX (sensor fusion library), MotionGC (gyroscope calibration library), MotionGR (gesture recognition library), MotionID (intensity detection library), MotionMC (magnetometer calibration library), MotionPE (pose estimation library), MotionPM (pedometer library), MotionPW (pedometer for wrist library), MotionSD (standing and sitting desk detection library), MotionSM (sleep monitoring library) and MotionTL (tilt sensing library).
Figure 2. X-CUBE-MEMS1 software architecture

<table>
<thead>
<tr>
<th>Application</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Middleware</td>
<td>MotionAC</td>
</tr>
<tr>
<td></td>
<td>MotionAR</td>
</tr>
<tr>
<td></td>
<td>MotionAT</td>
</tr>
<tr>
<td></td>
<td>MotionAW</td>
</tr>
<tr>
<td></td>
<td>MotionCP</td>
</tr>
<tr>
<td></td>
<td>MotionEC</td>
</tr>
<tr>
<td></td>
<td>MotionFA</td>
</tr>
<tr>
<td></td>
<td>MotionFD</td>
</tr>
<tr>
<td></td>
<td>MotionFX</td>
</tr>
<tr>
<td></td>
<td>MotionGC</td>
</tr>
<tr>
<td></td>
<td>MotionGR</td>
</tr>
<tr>
<td></td>
<td>MotionID</td>
</tr>
<tr>
<td></td>
<td>MotionMC</td>
</tr>
<tr>
<td></td>
<td>MotionPE</td>
</tr>
<tr>
<td></td>
<td>MotionPM</td>
</tr>
<tr>
<td></td>
<td>MotionPW</td>
</tr>
<tr>
<td></td>
<td>MotionSD</td>
</tr>
<tr>
<td></td>
<td>MotionSM</td>
</tr>
<tr>
<td></td>
<td>MotionTL</td>
</tr>
</tbody>
</table>

Hardware Abstraction

STM32Cube Hardware Abstraction Layer (HAL)

2.3 Folder structure

Figure 3. X-CUBE-MEMS1 package folder structure

The following folders are included in the software package:

- The **Documentation** folder contains a compiled HTML file generated from the source code and detailed documentation regarding the software components and APIs.
- The **Drivers** folder contains the HAL drivers, the board-specific drivers for each supported board or hardware platform, including those for the on-board components and the CMSIS layer, which is a vendor-independent hardware abstraction layer for the Cortex-M processor series.
• The **Middleware** folder contains the motion libraries, a platform-independent software layer provided in binary format for the Cortex-M4 processor series.

• The **Projects** folder contains several examples and applications for the NUCLEO-L053R8, NUCLEO-L152RE, NUCLEO-L476RG and NUCLEO-F401RE platforms to show the use of sensor APIs and the use of the motion libraries, provided with three development environments (IAR Embedded Workbench for ARM, RealView MDK-ARM Microcontroller Development Kit, System Workbench for STM32).

• The **Utilities** folder contains a "PC_software" subfolder containing a link for downloading the Unicleo-GUI, a Windows PC utility which shows real time sensor data.

### 2.4 APIs

Detailed technical information about the APIs available to the user can be found in the compiled HTML file "X_CUBE_MEMS1.chm" in the "Documentation" folder of the software package, where all the functions and parameters are fully described.

### 2.5 DataLogExtended application

The DataLogExtended sample application using the X-NUCLEO-IKS01A2 expansion board with the NUCLEO-F401RE, NUCLEO-L053R8,NUCLEO-L152RE or NUCLEO-L476RG board is provided in the "Projects" directory. Ready-to-use projects are available for multiple IDEs.

In the DataLog application, real-time sensor data are transmitted via serial port to a PC using the HAL_UART_Transmit() system call.

Transmitted sensor data can be viewed through Unicleo-GUI, a PC-based application developed by STMicroelectronics, available on www.st.com, which can be used to read and show data from the sensors 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, for example, would require the following steps:

- Initialization: IKS01A2_ENV_SENSOR_Init(...);
- Sensor temperature reading: IKS01A2_ENV_SENSOR_GetValue(...)
- Data serialization: Serialize();
- 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 Tera Term, the user can use the blue button on the STM32 Nucleo expansion board to start and stop the data log.

After connection has been established, the user can view data from all on-board environment sensors (temperature, humidity and pressure sensors) and all on-board inertial sensors (accelerometer, gyroscope and magnetometer sensors) and organize data in graphs (using Unicleo-GUI).

In addition, the user can change the sensor output data rate (ODR) and full scale (FS) settings using the Unicleo-GUI scroll-down menu without modifying the firmware and also get or set any sensor register value on a specific address. The application serial settings are: baud rate 921600 bps; 8 data bits; No parity; 1 stop bit; no hard flow control.
2.6 Unicleo-GUI data logging utility

The X-CUBE-MEMS1 expansion for STM32Cube contains a web link to download a utility for Windows PCs called "Unicleo-GUI", available on www.st.com.

Before using this utility, ensure that the STM32 Nucleo development board plus expansion board assembly is connected to the PC.

**Step 1.** Check the Windows Device Manager for the ST COM port number.

In the example below, it is COM7.
Step 2. Launch Unicleo-GUI application and ensure the COM port number for the current Nucleo board is correct.

Step 3. Select between various sensors (e.g., pressure, temperature, humidity, accelerometer, gyroscope, magnetometer) available on the expansion board and set appropriate delay/interval in milliseconds between consecutive data points; the default is 100 ms.
Step 4. Press "Start" to display the data.

Figure 7. Unicleo-GUI Utility sensor and interval selection

Step 5. Press "Motion MEMS" to display inertial sensor data.

Figure 8. Unicleo-GUI data plot
Step 6. Press "Environmental" to display environmental sensor data.
Step 7. Press the right mouse button to choose between recording in a file or removing the data from the panel.
2.7 DataLogTerminal application

This application shows how to use the X-NUCLEO-IKS01A2 to send sensor data from an STM32 Nucleo board using UART to a connected PC and display it on generic applications like Tera Term. After connection has been established, the user can view the data from all on-board environment sensors (temperature, humidity and pressure sensors) and all on-board inertial sensors (accelerometer, gyroscope and magnetometer sensors) using a hyper terminal. The application serial settings are: baud rate 115200 bps; 8 data bits; No parity; 1 stop bit; no hard flow control.

![Figure 12. DataLogTerminal application screenshot](image)

2.8 FIFO mode application for pressure sensor

This application shows how to use the X-NUCLEO-IKS01A2 to store pressure and temperature data in FIFO mode and send data from an STM32 Nucleo board via UART to a connected PC, displaying it on generic applications like Tera Term.

Note: This feature is only available for LPS22HB.

After connection has been established, press the user button to store pressure and temperature data in the FIFO mode and then view the data using a hyper terminal. The application serial settings are: baud rate 115200 bps; 8 data bits; No parity; 1 stop bit; no hard flow control.
2.9 6D orientation application for accelerometer sensor

This application shows how to use the X-NUCLEO-IKS01A2 expansion board to find out the 6D orientation and send data from an STM32 Nucleo board via UART to a connected PC, displaying it on generic applications like Tera Term.

Note: This feature is only available for LSM6DSL.

After connection has been established, the user can rotate the board to change the 6D orientation and then view the data using a hyper terminal or just push the user button to display the current 6D orientation.

The application serial settings are: baud rate 115200 bps; 8 data bits; No parity; 1 stop bit; no hard flow control.
2.10 FIFO continuous mode application for gyroscope sensor

This application shows how to use the X-NUCLEO-IKS01A2 expansion board to store gyroscope data in FIFO continuous mode and send data from an STM32 Nucleo board via UART to a connected PC, displaying it on generic applications like Tera Term.

**Note:** This feature is only available for LSM6DSL.

After connection has been established, the user can push the user button to launch the FIFO demo in continuous mode and then view the data using a hyper terminal. By pressing again the STM32 Nucleo board user button, FIFO continuous mode changes into FIFO bypass mode. If you press the user button once again, the FIFO demo restarts in continuous mode and so on.

The application serial settings are: baud rate 115200 bps; 8 data bits; No parity; 1 stop bit; no hard flow control.

![Figure 15. FIFO continuous mode application for gyroscope sensor screenshot](image)

2.11 FIFO low power mode application for accelerometer sensor

This application shows how to use the X-NUCLEO-IKS01A2 expansion board to store accelerometer data in FIFO continuous mode and send data from an STM32 Nucleo board via UART to a connected PC, displaying it on generic applications like Tera Term.

**Note:** This feature is only available for LSM6DSL.

After connection has been established, the user can push the user button to launch the FIFO low power demo and then view the data using a hyper terminal; afterwards, the component enters sleep mode. The user can press the user button to launch again the FIFO low power demo.

The application serial settings are: baud rate 115200 bps; 8 data bits; No parity; 1 stop bit; no hard flow control.
2.12 FIFO mode application for gyroscope sensor

This application shows how to use the X-NUCLEO-IKS01A2 expansion board to store gyroscope data in FIFO mode and send data from an STM32 Nucleo board via UART to a connected PC, displaying it on generic applications like Tera Term.

Note: This feature is only available for LSM6DSL.

After connection has been established, the user can push the user button to launch the FIFO mode demo and then view the data using a hyper terminal; press the user button to launch again the FIFO mode demo.

The application serial settings are: baud rate 115200 bps; 8 data bits; No parity; 1 stop bit; no hard flow control.
2.13 Free fall detection application for accelerometer sensor

This application shows how to detect the free fall event using the X-NUCLEO-IKS01A2 expansion board and an STM32 Nucleo board.

Note: This feature is only available for LSM6DSL.

After application starts, the user can try to let the STM32 Nucleo board falling; when the free fall event is detected, the Nucleo board LED is switched on for a while.

The Nucleo board user button can be used to enable/disable the free fall detection feature.

2.14 Multiple event application for accelerometer sensor

This application shows how to use the X-NUCLEO-IKS01A2 expansion board to detect free fall, tap, double tap, tilt, wake up, 6D Orientation and step events and send data from an STM32 Nucleo board via UART to a connected PC, displaying it on generic applications like Tera Term.

Note: This feature is only available for LSM6DSL.

After connection has been established, the user can simulate all the events and then view the data using a hyper terminal or can push the user button to enable/disable all hardware features.

The application serial settings are: baud rate 115200 bps; 8 data bits; No parity; 1 stop bit; no hard flow control.

Figure 18. Multiple event application for accelerometer sensor screenshot

2.15 Pedometer application for accelerometer sensor

This application shows how to use the X-NUCLEO-IKS01A2 expansion board to count steps and send data from an STM32 Nucleo board via UART to a connected PC, displaying it on generic applications like Tera Term.

Note: This feature is only available for LSM6DSL.

After connection has been established, the user can shake the board to simulate the steps and then view the data using a hyper terminal or can push the user button to reset the step counter.

The application serial settings are: baud rate 115200 bps; 8 data bits; No parity; 1 stop bit; no hard flow control.
2.16 Self-test application for accelerometer and gyroscope sensors

This application shows how to use the X-NUCLEO-IKS01A2 expansion board to test accelerometer and gyroscope operation mode and send data from an STM32 Nucleo board via UART to a connected PC, displaying it on generic applications like Tera Term.

Note: This feature is only available for LSM6DSL.

After connection has been established, the user can push the user button to launch the self-test and then view the data using a hyper terminal. The application serial settings are: baud rate 115200 bps; 8 data bits; No parity; 1 stop bit; no hard flow control.
2.17 Single tap and double tap detection for accelerometer sensor

This application shows how to detect the single and double tap events using the X-NUCLEO-IKS01A2 expansion board and an STM32 Nucleo board.

*Note:* This feature is only available for LSM6DSL.

After application starts, the user can try to tap the STM32 Nucleo board; when the single tap event is detected, the Nucleo board LED is switched on for a while. The user can press the user button to pass from the single tap detection to the double tap detection feature; when the double tap event is detected, the LED is switched on twice for a while. The user can press again the Nucleo board user button to disable the single/double tap detection feature and so on.

2.18 Tilt detection for accelerometer sensor

This application shows how to detect the tilt event using the X-NUCLEO-IKS01A2 expansion board and an STM32 Nucleo board.

*Note:* This feature is only available for LSM6DSL.

After application starts, the user can try to tilt the STM32 Nucleo board; when the tilt event is detected, the Nucleo board LED is switched on for a while. The Nucleo board user button can be used to enable/disable the tilt detection feature.
2.19  **Wake up detection for accelerometer sensor**

This application shows how to detect the wake up event using the X-NUCLEO-IKS01A2 expansion board and an STM32 Nucleo board.

*Note:*  *This feature is only available for LSM6DSL.*

After application starts, the user can try to touch the STM32 Nucleo board; when the wake up event is detected, the Nucleo board LED is switched on for a while. The Nucleo board user button can be used to enable/disable the wake up detection feature.

2.20  **Sample applications for motion libraries**

Every motion library is provided together with a sample application that shows the main features of the library using the X-NUCLEO-IKS01A2 expansion board and an STM32 Nucleo board. For more information, refer to each motion library user manual available on www.st.com.
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 21. STM32 Nucleo board

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

3.1.2 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.
3.2 Software description

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

- **X-CUBE-MEMS1**: an STM32Cube expansion for sensor application development. The X-CUBE-MEMS1 firmware and associated documentation is available on www.st.com.
- Development tool-chain and compiler: The STM32Cube expansion software supports the three following environments:
  - IAR Embedded Workbench for ARM® (EWARM) toolchain + ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM) toolchain + ST-LINK
  - System Workbench for STM32 + ST-LINK

3.3 Hardware setup

The following hardware components are required:

1. One STM32 Nucleo development platform (suggested order code: NUCLEO-F401RE or NUCLEO-L053R8 or NUCLEO-L152RE or NUCLEO-L476RG)
2. One sensor expansion board (order code: X-NUCLEO-IKS01A2)
3. One USB type A to mini-B USB cable to connect the STM32 Nucleo to a PC

3.4 Software setup

To set up the SDK, run the sample testing scenario based on the GUI utility and customize applications, select one of the integrated development environments supported by the STM32Cube expansion software and follow the system requirements and setup information provided by the IDE provider.

For more information you can refer to Unicleo-GUI user manual available on www.st.com.
3.5 STM32 Nucleo and sensor expansion board 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 by searching STSW-LINK008 or STSW-LINK009 (according to your version of Windows) on www.st.com.

The X-NUCLEO-IKS01A2 sensor expansion board 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 23. Sensor expansion board plugged to STM32 Nucleo board

3.6 Unicleo-GUI setup

The Unicleo-GUI retrieves sensor data from the connected STM32 Nucleo board and displays it in tables and graphs.

To use the Unicleo-GUI, ensure the relevant hardware and software has been correctly set up. The Unicleo-GUI installer for a Windows PC can be downloaded from www.st.com. "Utilities\PC_Software" folder contains a link to the download page.
## Revision history

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>19-Feb-2015</td>
<td>1</td>
<td>Initial release</td>
</tr>
<tr>
<td>17-Jun-2015</td>
<td>2</td>
<td>Add support to L1</td>
</tr>
<tr>
<td>30-Sep-2015</td>
<td>3</td>
<td>Removed Middlewares folder and added support for L4</td>
</tr>
<tr>
<td>21-Dec-2015</td>
<td>4</td>
<td>Updated Figure 2: &quot;X-CUBE-MEMS1 software architecture&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Section 2.5: &quot;Sample application description&quot;</td>
</tr>
<tr>
<td>26-Apr-2016</td>
<td>5</td>
<td>Updated Section 2.1: Overview</td>
</tr>
<tr>
<td>04-Nov-2016</td>
<td>6</td>
<td>Text and formatting changes throughout document</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added new board compatibility information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added Section 3.1.3: &quot;X-NUCLEO-IKS01A2 expansion board&quot;</td>
</tr>
<tr>
<td>22-Mar-2017</td>
<td>7</td>
<td>Updated Section &quot;Introduction&quot;, Section 2.1: &quot;Overview&quot;, Section 2.2: &quot;Architecture&quot;, Section 2.3: &quot;Folder structure&quot;, Section 2.5: &quot;DataLog application&quot;, Section 2.6: &quot;Unicleo-GUI data logging utility&quot;, Section 2.2.2: &quot;Software setup&quot; and Section 2.2.5: &quot;Unicleo-GUI setup&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added Section 2.7: &quot;DataLogExtended application&quot;, Section 2.8: &quot;DataLogTerminal application&quot;, Section 2.9: &quot;FIFO mode application for pressure sensor&quot;, Section 2.10: &quot;6D orientation application for accelerometer sensor&quot;, Section 2.11: &quot;FIFO continuous mode application for gyroscope sensor&quot;, Section 2.12: &quot;FIFO low power mode application for accelerometer sensor&quot;, Section 2.13: &quot;FIFO mode application for gyroscope sensor&quot;, Section 2.14: &quot;Free fall detection application for accelerometer sensor&quot;, Section 2.15: &quot;Multiple event application for accelerometer sensor&quot;, Section 2.16: &quot;Pedometer application for accelerometer sensor&quot;, Section 2.17: &quot;Self-test application for accelerometer and gyroscope sensors&quot;, Section 2.18: &quot;Single tap and double tap detection for accelerometer sensor&quot;, Section 2.19: &quot;Tilt detection for accelerometer sensor&quot;, Section 2.20: &quot;Wake up detection for accelerometer sensor&quot; and Section 2.21: &quot;Sample applications for motion libraries&quot;</td>
</tr>
<tr>
<td>20-Sep-2017</td>
<td>8</td>
<td>Updated Section ● Introduction, Section 2.2 Architecture and Figure 2. X-CUBE-MEMS1 software architecture</td>
</tr>
<tr>
<td>10-Jul-2018</td>
<td>9</td>
<td>Updated Introduction, Section 2.1 Overview, Section 2.2 Architecture, Section 2.3 Folder structure and Section 2.5 DataLogExtended application.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Removed references to X-NUCLEO-IKS01A1 throughout the document.</td>
</tr>
</tbody>
</table>
# Contents

1. **What is STM32Cube?** .............................................................. 2  
   1.1 STM32Cube overview .......................................................... 2  
   1.2 STM32Cube architecture ........................................................ 2  

2. **X-CUBE-MEMS1 software expansion for STM32Cube** ................. 4  
   2.1 Overview ..................................................................... 4  
   2.2 Architecture ................................................................... 4  
   2.3 Folder structure ................................................................ 5  
   2.4 APIs ......................................................................... 6  
   2.5 DataLogExtended application .................................................... 6  
   2.6 Unicleo-GUI data logging utility ................................................... 7  
   2.7 DataLogTerminal application .................................................... 11  
   2.8 FIFO mode application for pressure sensor ............................... 12  
   2.9 6D orientation application for accelerometer sensor ....................... 13  
   2.10 FIFO continuous mode application for gyroscope sensor ............... 13  
   2.11 FIFO low power mode application for accelerometer sensor ............ 14  
   2.12 FIFO mode application for gyroscope sensor .............................. 15  
   2.13 Free fall detection application for accelerometer sensor ............... 15  
   2.14 Multiple event application for accelerometer sensor .................... 16  
   2.15 Pedometer application for accelerometer sensor ......................... 16  
   2.16 Self-test application for accelerometer and gyroscope sensors .......... 17  
   2.17 Single tap and double tap detection for accelerometer sensor .......... 18  
   2.18 Tilt detection for accelerometer sensor .................................... 18  
   2.19 Wake up detection for accelerometer sensor ............................. 18  
   2.20 Sample applications for motion libraries ................................. 19  

3. **System setup guide** ............................................................... 20  
   3.1 Hardware description .......................................................... 20  
      3.1.1 STM32 Nucleo platform ................................................. 20  
      3.1.2 X-NUCLEO-IKS01A2 expansion board ............................. 20  
   3.2 Software description .......................................................... 21
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>Hardware setup</td>
<td>21</td>
</tr>
<tr>
<td>3.4</td>
<td>Software setup</td>
<td>21</td>
</tr>
<tr>
<td>3.5</td>
<td>STM32 Nucleo and sensor expansion board setup</td>
<td>21</td>
</tr>
<tr>
<td>3.6</td>
<td>Unicleo-GUI setup</td>
<td>22</td>
</tr>
</tbody>
</table>

**Revision history**

23
List of tables

Table 1. Document revision history ............................................................. 23
List of figures

Figure 1. Firmware architecture ...............................................................2
Figure 2. X-CUBE-MEMS1 software architecture ...................................................5
Figure 3. X-CUBE-MEMS1 package folder structure .................................................5
Figure 4. Unicleo-GUI: DataLogExtended application screenshot ..................................7
Figure 5. Windows Device Manager .........................................................................8
Figure 6. Unicleo-GUI main page ..........................................................................8
Figure 7. Unicleo-GUI Utility sensor and interval selection .........................................9
Figure 8. Unicleo-GUI data plot ..........................................................................9
Figure 9. Unicleo-GUI Motion Sensor Plot ..........................................................10
Figure 10. Unicleo-GUI Environmental Sensor Plot .............................................11
Figure 11. Unicleo-GUI right click Menu ..........................................................11
Figure 12. DataLogTerminal application screenshot ...............................................12
Figure 13. FIFO mode application for pressure sensor screenshot .............................13
Figure 14. 6D orientation application for accelerometer sensor screenshot ...............13
Figure 15. FIFO continuous mode application for gyroscope sensor screenshot ...............14
Figure 16. FIFO low power mode application for accelerometer sensor screenshot ..........14
Figure 17. FIFO mode application for gyroscope sensor screenshot ..............................15
Figure 18. Multiple event application for accelerometer sensor screenshot ...............16
Figure 19. Pedometer application for accelerometer sensor screenshot .......................17
Figure 20. Self-test application for accelerometer and gyroscope sensors screenshot .........18
Figure 21. STM32 Nucleo board .......................................................................20
Figure 22. X-NUCLEO-IKS01A2 MEMS and environmental sensor expansion board ........21
Figure 23. Sensor expansion board plugged to STM32 Nucleo board .......................22