AN4594
Application note

Getting started with the STM32 Nucleo and the sensor expansion board X-NUCLEO-IKS01A1

Introduction

This document describes how to develop a sensor-based application using the STM32 Nucleo platform and the X-NUCLEO-IKS01A1 sensor expansion board, in the STM32Cube software environment. The sensor expansion board is a multi-sensor board with temperature, humidity, pressure and motion MEMS measuring sensors.

Contents

1 **Acronyms and abbreviations 3**

2 **Getting started 4**

2.1 Hardware description 4

2.1.1 STM32F401RE Nucleo 4

2.1.2 Sensor expansion board 4

2.2 Software description 5

2.2.1 STM32Cube 6

2.2.2 Firmware for STM32 Nucleo with X-NUCLEO-IKS01A1 expansion board 6

2.3 Hardware and software setup 6

2.3.1 Hardware setup 6

2.3.2 Software setup 6

2.3.3 System setup guide 7

3 **Software overview 8**

3.1 Software architecture 8

3.2 STM32Cube HAL 8

3.3 Board support package (BSP) 9

4 **Guide for writing applications 10**

4.1 Relevant APIs for sensor applications 10

4.1.1 Initialization 10

4.2 Application description 11

5 **Using the Sensors_Datalog application 12**

6 **Revision history 15**

1 Acronyms and abbreviations

Table 1. Acronyms and abbreviations

Acronym	Description
BSP	Board support package
GUI	Graphical user interface
HAL	Hardware abstraction layer
I^2C	Inter integrated circuit
IDE	Integrated development environment
LED	Light emitting diode

2 Getting started

2.1 Hardware description

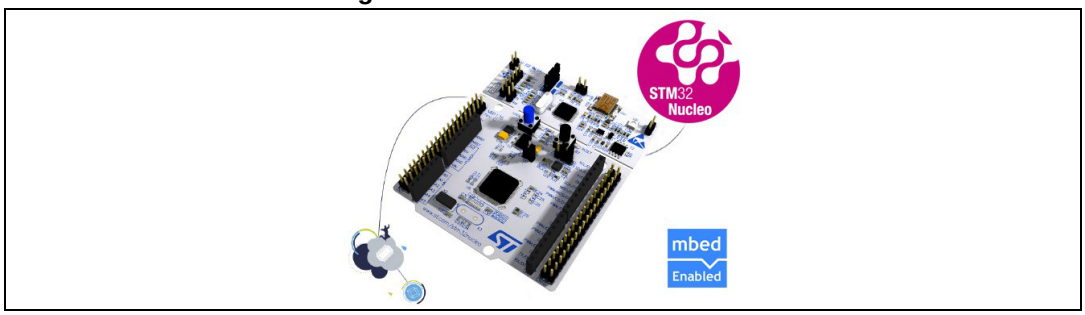
This section describes the individual hardware components required for developing a sensor-based application.

2.1.1 STM32F401RE Nucleo

STM32F401RE Nucleo board belongs to the STM32F401xD/xE range which is based on the high-performance ARM®Cortex™-M4 32-bit RISC core operating at a frequency of up to 84 MHz. Its Cortex-M4 core features a single-precision Floating Point Unit (FPU) which supports all ARM single-precision data-processing instructions and data types. It also implements a full set of DSP instructions and a memory protection unit (MPU) which enhances application security. It provides an affordable and flexible way for users to try new ideas and build prototypes with any STM32 microcontroller lines. The Arduino™ connectivity support and ST Morpho headers make it easy to expand the functionality of the Nucleo open development platform with a wide range of specialized expansion boards to choose from. The STM32 Nucleo board does not require any separate probes as it integrates the ST-LINK/V2-1 debugger/programmer. The STM32 Nucleo board comes with the comprehensive STM32 HAL software library, together with various packaged software examples.

The STM32F401RE Nucleo firmware and related documentation are available on st.com at http://www.st.com/web/en/catalog/mmc/FM141/SC1169/SS1577/LN1810/PF258797

Figure 1. STM32-F4 nucleo board



2.1.2 Sensor expansion board

The X-NUCLEO-IKS01A1 is a sensor expansion board usable with the STM32 Nucleo system. It is also compatible with the Arduino UNO R3 connector layout, and is designed around the STMicroelectronics Humidity (HTS221), Pressure (LPS25H) and Motion sensors (LIS3MDL and LSM6DS0). The X-NUCLEO-IKS01A1 interfaces with the STM32 MCU via the I^2C pin, but the user can change the default I^2C port and the device IRQ by changing one resistor on the expansion board.

Figure 2. X-NUCLEO-IKS01A1 sensor expansion board

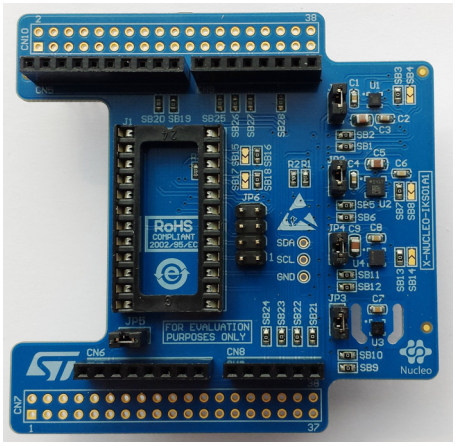
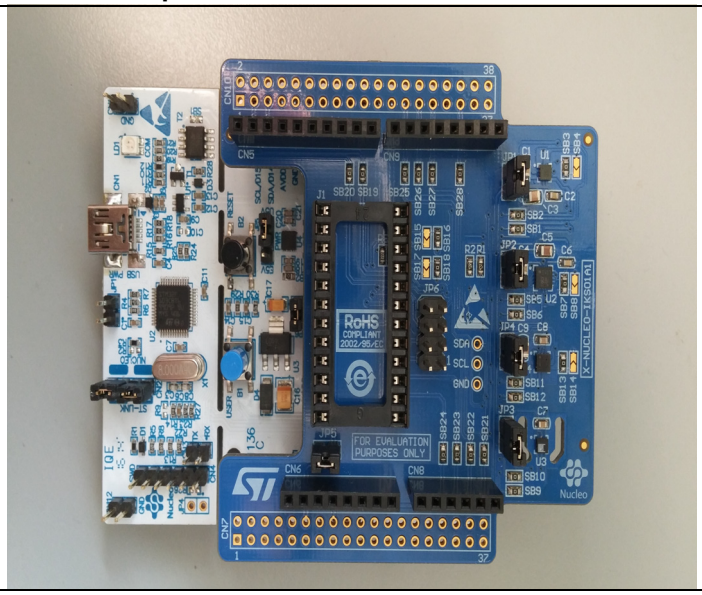


Figure 3. Sensor expansion board connected to STM32 Nucleo board



The X-CUBE-MEMS1 firmware and related documentation is available on st.com at http://www.st.com/x-nucleo.

2.2 Software description

The following software components are required to set up a suitable development environment for creating applications based on the sensor expansion board:

- STM32Cube environment and related firmware for the STM32Nucleo and sensor expansion board
- Development tool chain and compiler: IAR embedded workbench IDE is the development environment used for the applications described in this document.

2.2.1 STM32Cube

STM32Cube is a development framework which provides tools and libraries to develop C applications on STM32 series platforms. A full description of the STM32Cube framework can be found at http://www.st.com/stm32cube.

STM32Cube comprises the STM32CubeF4platform which includes the STM32Cube HAL (the STM32 abstraction layer embedded software which maximizes portability across STM32 portfolio), plus a consistent set of middleware components (such as RTOS, USB, FatFS and STM32 touch sensing). All embedded software utilities come with a full set of examples.

All the generic embedded software components required to develop an application on STM32F4 microcontrollers are gathered by STM32CubeF4 in a single package.

STM32CubeF4 is fully compatible with the STM32CubeMX code generator for initialization code. The package includes a low level hardware abstraction layer (HAL) that covers the microcontroller hardware, together with an extensive set of examples running on STMicroelectronics boards. The HAL is available in open-source BSD license for developer convenience.

The STM32CubeF4 firmware and related documentation is available on st.com at http://www.st.com/web/en/catalog/tools/PF259243

2.2.2 Firmware for STM32 Nucleo with X-NUCLEO-IKS01A1 expansion board

X-CUBE-MEMS1 is a package that contains firmware with the STM32 Cube framework that is compatible with the STM32 Nucleo and the X-NUCLEO-IKS01A1 boards. It has sample applications that can be used to obtain data from the sensors on the expansion board. The same sample applications are referenced in further sections of this document.

The firmware and related documentation are available on st.com at http://www.st.com/x-nucleo.

2.3 Hardware and software setup

This section describes hardware and software setup procedure for writing applications based on the sensor expansion board. It also describes the system setup needed for the above.

2.3.1 Hardware setup

The following hardware is required:

1. One STM32 Nucleo development platform (suggested order code: NUCLEO-F401RE)
2. One sensor expansion board (see *Figure 1*, order code: X-NUCLEO-IKS01A1)
3. One USB type A to mini-B USB cable to connect the Nucleo to the PC

2.3.2 Software setup

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

Development tool chains and compilers

IAR embedded workbench for ARM (EWARM) tool-chain V7.20

The IAR tool-chain has the following minimum requirements:

- PC with Intel® or AMD® processor running one of the following Microsoft® operating systems:
  - Windows XP SP3
  - Windows Vista
  - Windows7

GUI utility Sensors_Datalog

The Sensors_Datalog utility has following minimum requirements:

- PC with Intel or AMD processor running one of following Microsoft operating systems:
  - Windows XP SP3
  - Windows Vista
  - Windows 7
- At least 128 MB of RAM
- 2 X USB ports
- 40 MB of hard disk space

2.3.3 System setup guide

This section describes how to setup different hardware components before writing and executing an application on the STM32 Nucleo board with the sensor expansion board.

Sensors_Datalog GUI setup

The Sensors_Datalog GUI included in the software package is a graphical user interface that can be used to interact and obtain data from the sensors on the sensor expansion board.

This utility retrieves sensor data from the connected Nucleo board and displays it in a tabular or graphical form.

In order to use the Sensors_Datalog GUI, make sure you have correctly set up your hardware and software (Sensors_Datalog GUI installed).

STM32 Nucleo and sensor expansion board setup

The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer. The developer can download the relevant version of the ST-LINK/V2-1 USB driver (according to the MS Windows OS) by searching ST-SW-LINK008 or STSW-LINK009 on www.st.com.

The X-NUCLEO-IKS01A1 sensor expansion board can be easily connected to the Nucleo motherboard through the Arduino UNO R3 extension connector, see *Figure 3*. The sensor expansion board is capable of interfacing with the external STM32 microcontroller on Nucleo using Inter-Integrated Circuit (I^2C) transport layer.

3 Software overview

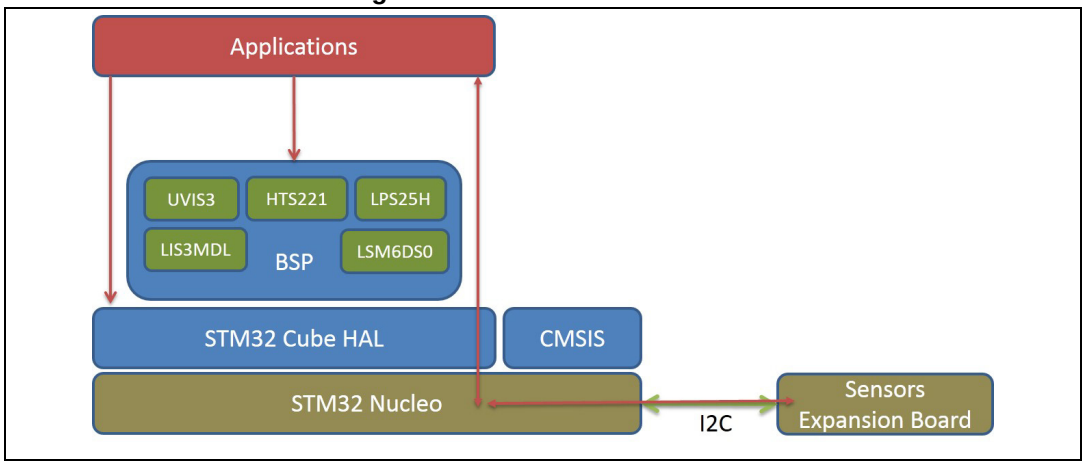
3.1 Software architecture

This chapter describes the following software layers which are used by the application software to access and use the sensor expansion board:

- STM32Cube HAL layer
- board support package (BSP) layer

The next figure outlines the layering of the software architecture that comprises the STM32 Nucleo and the BlueNRG expansion board:

Figure 4. Software architecture



3.2 STM32Cube HAL

The STM32Cube HAL is the hardware abstraction layer for the STM32 microcontroller which ensures maximum portability across STM32 platforms.

The HAL driver layer provides a simple, generic, multi-instance set of APIs (application programming interfaces) to interact with the upper layers (application, libraries and stacks). It is composed of generic and extension APIs and is based on generic architecture which allows the layers built upon it, such as the middleware layer, to implement their functions without dependence on the specific hardware configuration for a given Microcontroller Unit (MCU). This structure improves the library code reusability and facilitates portability to other devices.

For detailed information regarding the STM32Cube HAL drivers API, please refer to the document “Description of STM32F4xx HAL drivers”, User manual UM1749, available from st.com at http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00113898.pdf.

3.3 Board support package (BSP)

The board software package (BSP) includes support for the peripherals on the STM32 Nucleo board with a limited set of APIs that provides a programming interface for certain board-specific peripherals such as the LED, the user button, etc., and also facilitates identifying the specific board version.

For the sensor expansion board, it provides the programming interface for various sensors like temperature, pressure and UV sensors and support for initializing and obtaining sensor data.

4 Guide for writing applications

This chapter describes how to write applications which use sensors, based on the STM32 Nucleo board equipped with the sensor expansion board.

4.1 Relevant APIs for sensor applications

This section describes generic initialization and setup for writing sensor expansion board applications.

4.1.1 Initialization

Every application must first execute the following basic initialization steps in order to configure and set up the STM32 Nucleo with the sensor expansion board hardware and the software stack for correct operation.

Initializing STM32Cube HAL

The STM32Cube HAL library must be initialized so that the necessary hardware components are correctly configured.

- HAL_Init();
  This API initializes the HAL library: it configures Flash prefetch, Flash preread and Buffer Cache0; it also configures the time base source, vectored interrupt controller and low-level hardware.

Initializing Nucleo board peripherals and sensor expansion board

Some of the Nucleo on-board peripherals, hardware and the sensor expansion board must be configured before they are used. The functions to do this are:

- BSP_LED_Init(Led_TypeDef Led);
  This API configures the LED on the Nucleo.
- BSP_PB_Init(Button_TypeDef Button, ButtonMode_TypeDef Button_Mode);
  This API configures the user button in GPIO mode or in external interrupt (EXTI) mode.
- BSP_HUM_TEMP_isInitialized()
  This API configures the temperature sensor (hts221) on the sensor expansion board.
- BSP_PRESSURE_isInitialized()
  This API configures the pressure sensor (lps25h) on the sensor expansion board.
- BSP_IMU_6AXES_isInitialized()
  This API configures accelerometer and gyroscope sensor (LSM6DS0) on the sensor expansion board.
- BSP_MAGNETO_isInitialized
  This API configures magnetometer sensor (LIS3MDL) on the sensor expansion board.

Reading data from sensor expansion board

The BSP provides the API and the functionality for taking the data from sensors. Once the sensors are initialized we can get data from them using these functions:

- BSP_HUM_TEMP_GetTemperature(float * Temp)
  This API gives the access to temperature sensor data.
- BSP_HUM_TEMP_GetHumidity (float * Humid)
  This API gives the access to humidity sensor data.
- BSP_PRESSURE_GetPressure((float *)&Press)
  This API gives the access to pressure sensor data.
- BSP_IMU_6AXES_X_GetAxesRaw ((AxesRaw_TypeDef *)&ACC_Value)
  This API gives the access to accelerometer sensor data.
- BSP_IMU_6AXES_G_GetAxesRaw((AxesRaw_TypeDef *)&GYR_Value)
  This API gives the access to gyroscope sensor data.
- BSP_MAGNETO_M_GetAxesRaw((AxesRaw_TypeDef *)&MAG_Value)
  This API gives the access to magnetometer sensor data.

4.2 Application description

An example application using the sensor expansion board with the Nucleo F4 board can be downloaded from [$BASE_DIR\Projects\STM32F4xx-Nucleo\Examples\EWARM\project.eww]. In this application, real-time sensor data is transmitted to PC via a serial port using system call HAL_UART_Transmit(). Transmitted sensor data can be viewed using Sensors_Datalog, a PC-based application developed by STMicroelectronics (refer to *Section 5* for further details). Sensors_Datalog can be used to read and visualize data from the sensor expansion board connected to the PC via the Nucleo board. The firmware provided formats the sensor data that can be read by the Sensors_Datalog utility. For example, sending temperature data through UART is typically handled by the following steps:

- BSP_HUM_TEMP_isInitialized();
- BSP_HUM_TEMP_GetTemperature();
- Serialize();
- HAL_UART_Transmit();

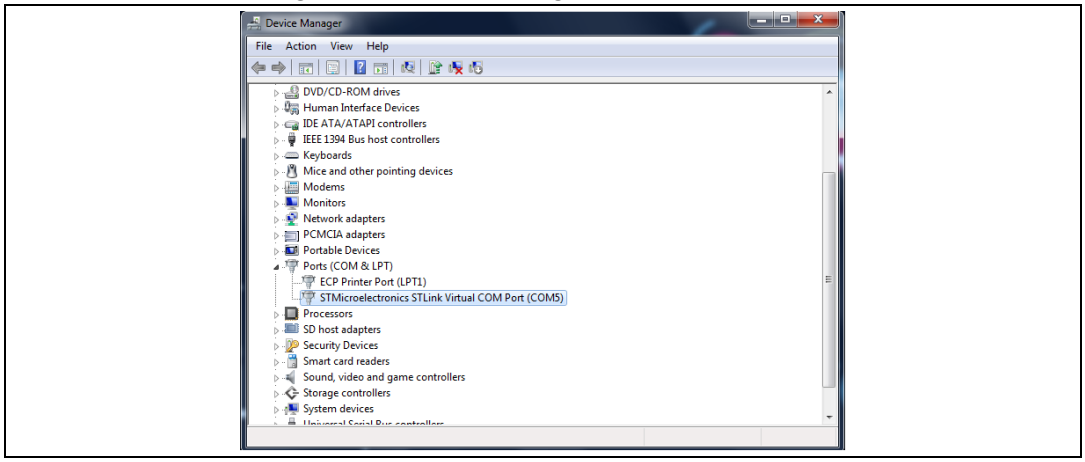
The Serialize() function formats the temperature data readable by the Sensors_Datalog utility. Similarly, data from other sensors is also formatted and communicated to the utility. When connected via Teraterm, the user can press the blue button on the Nucleo board to start and stop the data log.

5 Using the Sensors_Datalog application

This section describes the Sensors_Datalog utility. The user must first ensure that the necessary drivers described in previous sections are installed and the STM32 Nucleo with the expansion board is connected to a PC. Please follow the sequence below:

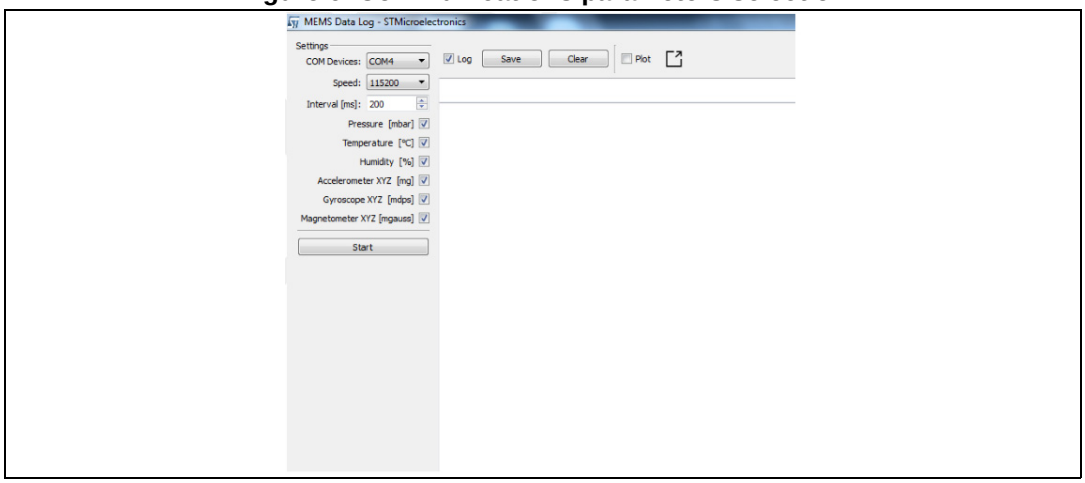
1. Check the windows “device manager” to determine the ST COM port; in the example *Figure 5*, the port is COM5.

Figure 5. Device manager view on windows



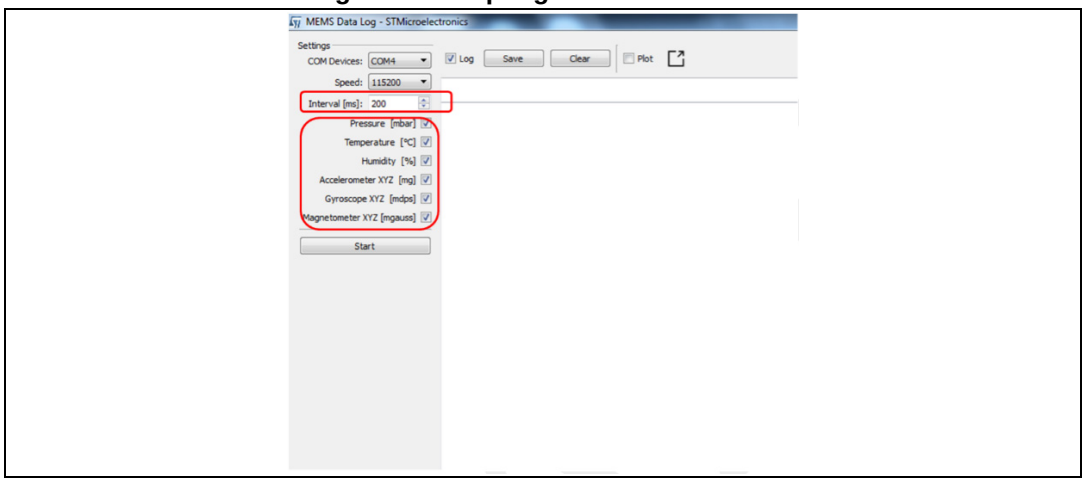
2. Launch Sensors_Datalog.exe and check if the COM Device number for the current expansion board is correct and the Speed is set to 115200, as shown in *Figure 6*.

Figure 6. Communications parameters selection



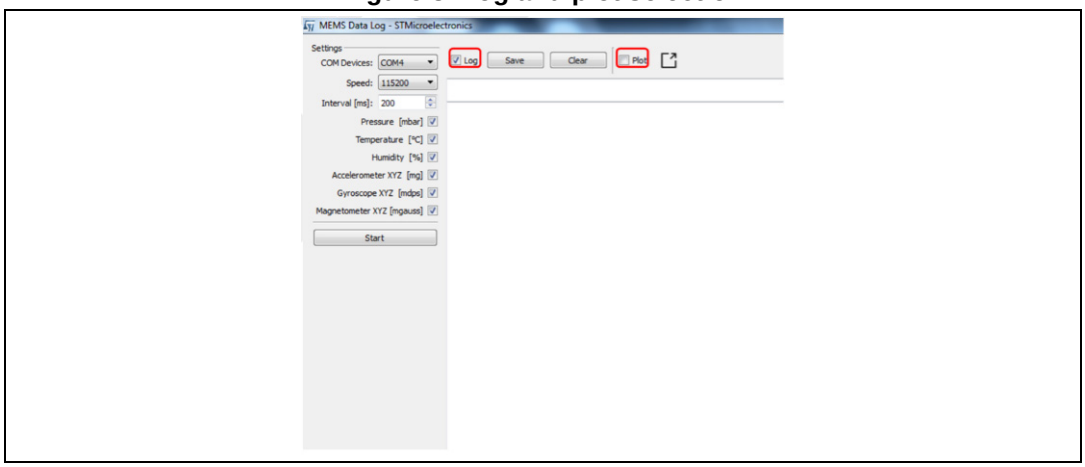
3. Select the desired sensors to obtain a data log. You can choose from the various sensors (pressure, temperature, humidity, accelerometer, gyroscope, magnetometer) available on the expansion board.
4. Set appropriate delay/interval in milliseconds between consecutive data points; the default is 200 ms.

Figure 7. Sampling interval selection



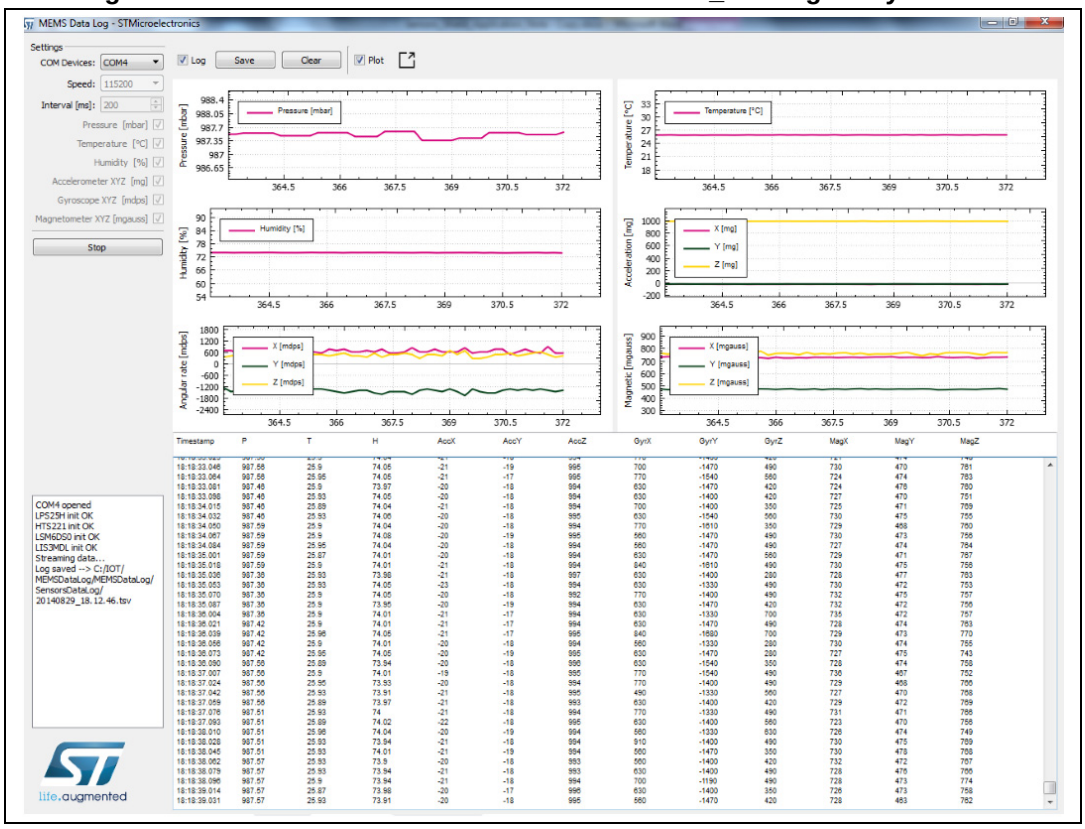
5. You can save the log to a file that you can find in a folder called “SensorsDataLog”; or you can clear the log by pressing the “ClearLog” button.
6. Check the “Plot” CheckBox to see a data log for the sensors that you have selected.

Figure 8. Log and plot selection



7. Press Start, and the data is displayed as shown in *Figure 8*

Figure 9. Sensor data visualization in the Sensors_Datalog utility window



6 Revision history

Table 2. Document revision history

Date	Revision	Changes
19-Nov-2014	1	Initial release.