
Getting started with the X-CUBE-IKA02A1 multifunctional software expansion for STM32Cube

Introduction

The X-CUBE-IKA02A1 expansion software package for STM32Cube runs on the STM32 microcontrollers and can be used to read the analog gas concentration value, reference voltage and temperature from the STLM20 sensor. The software library sets the appropriate sensor sensitivity and performs the compensation over the temperature range.

The expansion is built on STM32Cube software technology to ease portability across different STM32 microcontrollers. It is compatible with the P-NUCLEO-IKA02A1 development kit.

The software comes with a sample implementation to help you get started.

Contents

1	What is STM32Cube?	4
1.1	How does this software complement STM32Cube?.....	4
1.2	STM32Cube architecture	4
2	X-CUBE-IKA02A1 software expansion for STM32Cube.....	7
2.1	Overview	7
2.2	Architecture	7
2.3	Folder structure	8
2.4	APIs.....	8
2.5	Gas sensor sensitivity setting.....	9
2.6	Gain setting	10
2.7	Gas concentration reading sample.....	10
2.8	Unicleo GUI sample	10
3	Revision history	12

List of figures

Figure 1: Firmware architecture	5
Figure 2: X-CUBE-IKA02A1 software architecture	8
Figure 3: X-CUBE-IKA02A1 package folder structure	8
Figure 4: Data matrix reading from TGS5141	9
Figure 5: P-NUCLEO-IKA02A1 expansion board connected to Tera Term: firmware output with no gas presence	10
Figure 6: Unicleo Custom Fields window.....	11

1 What is STM32Cube?

STM32Cube™ is an STMicroelectronics initiative that helps you reduce 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

1.1 How does this software complement STM32Cube?

This software is based on the STM32CubeHAL hardware abstraction layer for the STM32 microcontroller. The package extends STM32Cube by providing a board support package (BSP) for the gas sensing expansion board.

Several sample projects in the Projects\Multi\Examples directory are available for developers to use and start experimenting with the code.

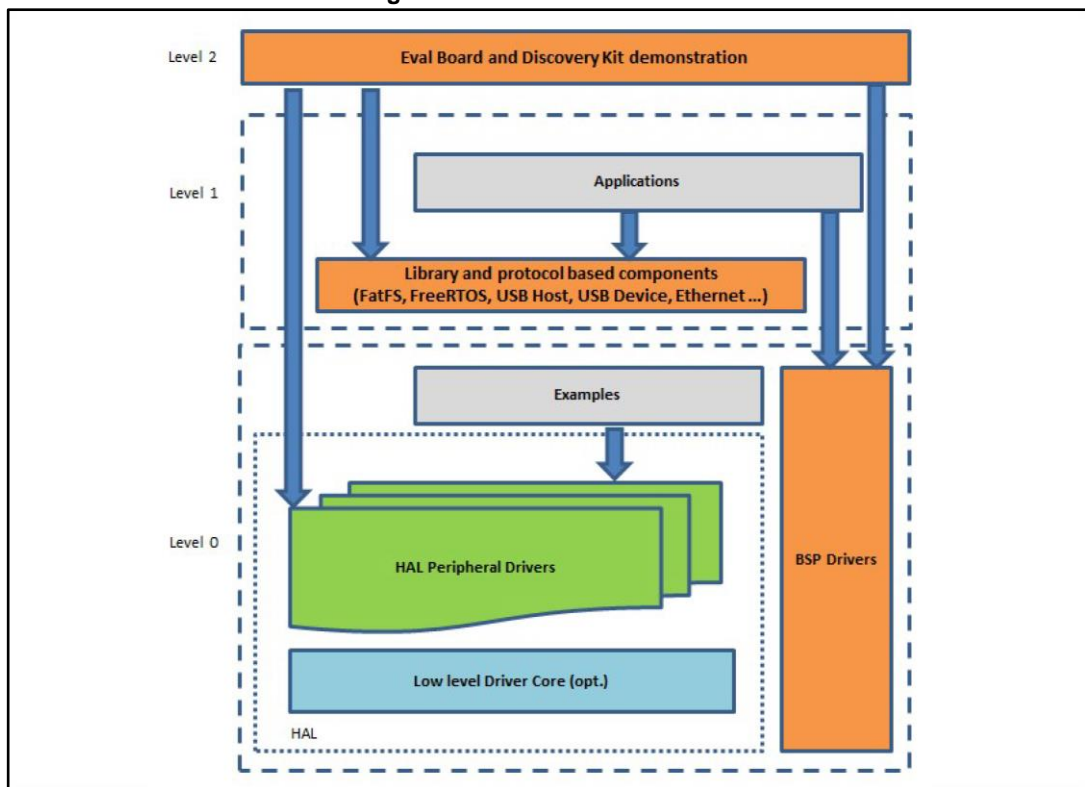
These samples are ready to be compiled using Keil (MDK-ARM) or IAR (EWARM):

- Gas concentration reading sample:
 - reads the analog value of gas concentration, temperature and reference voltage
 - performs compensation over the temperature range
 - prints data to serial line
- Unicleo GUI sample:
 - performs the same readings and compensation of the gas concentration reading sample
 - shows data in graphs via the user-friendly GUI
 - is available at www.st.com

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-IKA02A1 software expansion for STM32Cube

2.1 Overview

The X-CUBE-IKA02A1 software package expands the functionality of STM32Cube.

The key features of package are:

- Complete middleware to build application sensing the concentration of different gases depending on the sensor used
- High portability across different MCU families thanks to STM32Cube
- A sample application to transmit real-time data to PC using serial communication
- A Windows PC applications to log sensor data
- Free, user-friendly license terms
- Sample implementations available for the P-NUCLEO-IKA02A1 development kit.

The package also includes a sample application that the developer can use to start experimenting with the code.

2.2 Architecture

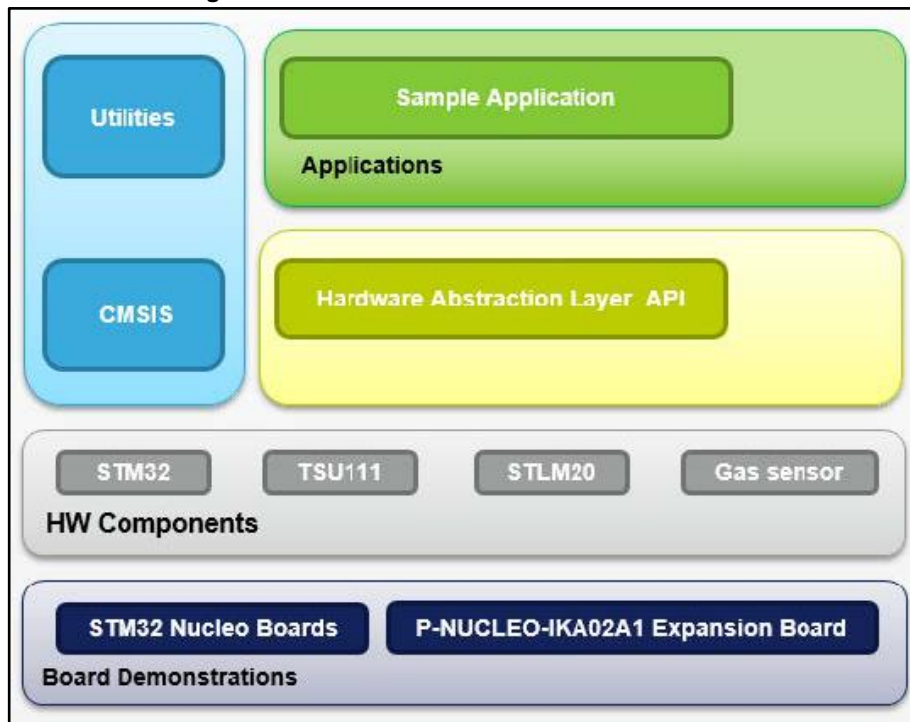
This software is a fully compliant expansion for STM32Cube that enables the development of applications using different gas sensors.

The software is based on the STM32CubeHAL hardware abstraction layer for the STM32 microcontroller. The package extends STM32Cube with 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:

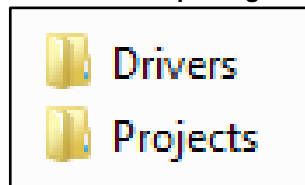
- The STM32Cube HAL driver layer provides a simple, generic and multi-instance set of APIs (application programming interfaces) to interact with the upper layers (application, libraries and stacks). It includes generic and extension APIs and is based on a generic architecture which allows the layers built on it (such as the middleware layer) to implement their functions without requiring specific microcontroller unit (MCU) configuration data. This structure improves library code re-usability 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.

Figure 2: X-CUBE-IKA02A1 software architecture



2.3 Folder structure

Figure 3: X-CUBE-IKA02A1 package folder structure



The following folders are included in the software package:

- 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 vendor-independent hardware abstraction layer for the Cortex-M processor series.
- The **Projects** folder contains a sample application for the NUCLEO-L053R8, NUCLEO-F401RE and NUCLEO-L476RG platforms to access sensor data; projects are provided for two development environments (IAR Embedded Workbench for ARM and RealView MDKARM Microcontroller Development Kit).

2.4 APIs

The X-CUBE-IKA02A1 software package provides simple APIs to read the analog values from the P-NUCLEO-IKA02A1 expansion board.

Solder bridges must be in line with the header file *x_nucleo_ika02a1.h*^a.

^a Refer to UM2247 at www.st.com for solder bridge setting.

Basic functions, such as converting analog to digital values using STM32 ADC converters, are handled by *stm32fxxx_nucleo_ika02a1.c*.

The BSP layer for the P-NUCLEO-IKA02A1 expansion board is provided by *x_nucleo_ika02a1.c*.

Some functions directly provide gas concentration readings in ppm value, temperature and function to perform compensation over temperature:

- to read temperature in °C: `Get_Temperature()`
- to get gas concentration in ppm: `Get_Gas_concentration(...)`
- to calculate compensated concentration: `Get_comp_gas_concentration(...)`

To provide appropriate compensation over temperature the lookup table, *sensor_calibration_TGS5141.h*, with the related sensitivity must be filled in.



By default, all the parameters are set to provide the best performance with the TGS5141 sensor.

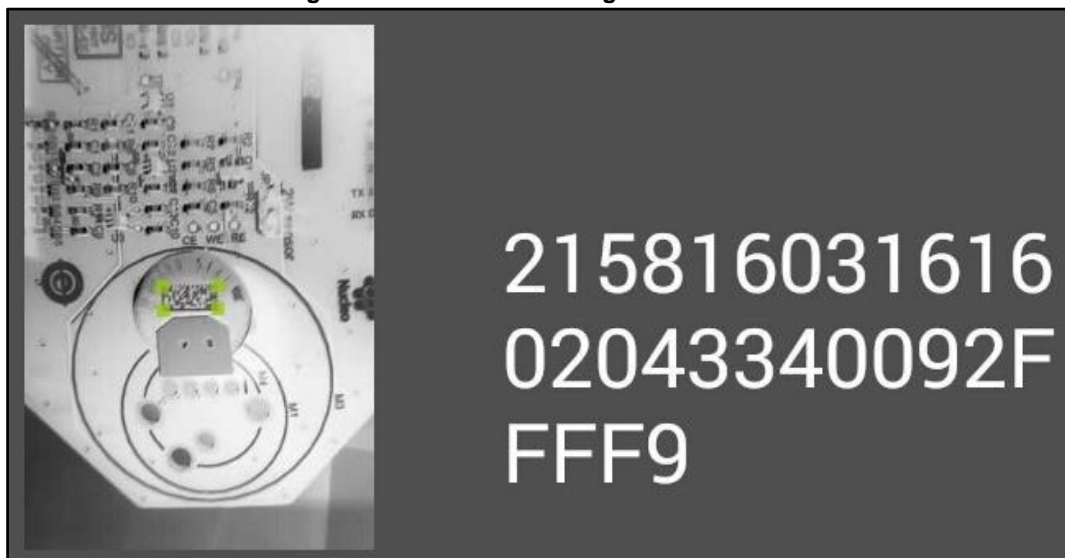
2.5 Gas sensor sensitivity setting

The sensitivity of any electrochemical sensor is unique to that sensor. Even devices produced consecutively on the same day can vary greatly in sensitivity. For the TGS5141, the output current per CO concentration can vary from 1.2nA/ppm to 3.2nA/ppm. The sensor sensitivity must therefore be calibrated in the gas chamber before embedding the sensor in the final application.

The P-NUCLEO-IKA02A1 comes with the Figaro TGS5141 sensor, which has a data matrix with calibration information that can be read by a smartphone with an appropriate application; the first four digits represent the sensor sensitivity in pA/ppm.

Sample routines are provided in the X-CUBE-IKA02A1 package which count values while compensating for different sensitivity parameters that can be easily modified and even stored in the STM32 microcontroller Flash.

Figure 4: Data matrix reading from TGS5141



2.6 Gain setting

The P-NUCLEO-IKA02A1 development kit for gas sensing allows the gain to be set with ease^a. The X-CUBE-IKA02A1 package contains routines that compensate for gain as well as sensitivity during counting.

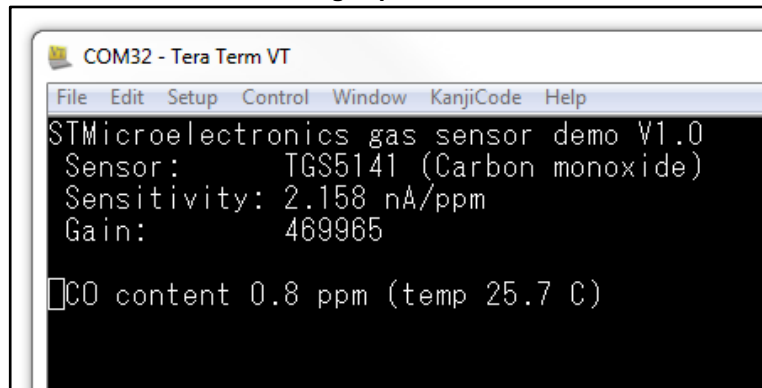
2.7 Gas concentration reading sample

The P-NUCLEO-IKA02A1 expansion board is pre-programmed with basic firmware and calibration values stored in the Flash memory.

To read these data, you only have to connect the board to the PC with installed drivers for STM32 Nucleo boards ([STSW-LINK009](#)) and a terminal application.

Once the virtual COM port is detected, you can open it by setting up 115200 baud rate, 8 bits and no parity.

Figure 5: P-NUCLEO-IKA02A1 expansion board connected to Tera Term: firmware output with no gas presence



The P-NUCLEO-IKA02A1 board can host different sensor types and the user can modify the gain.

The default application allows changing the sensitivity, gain and new value storage in the Flash memory.

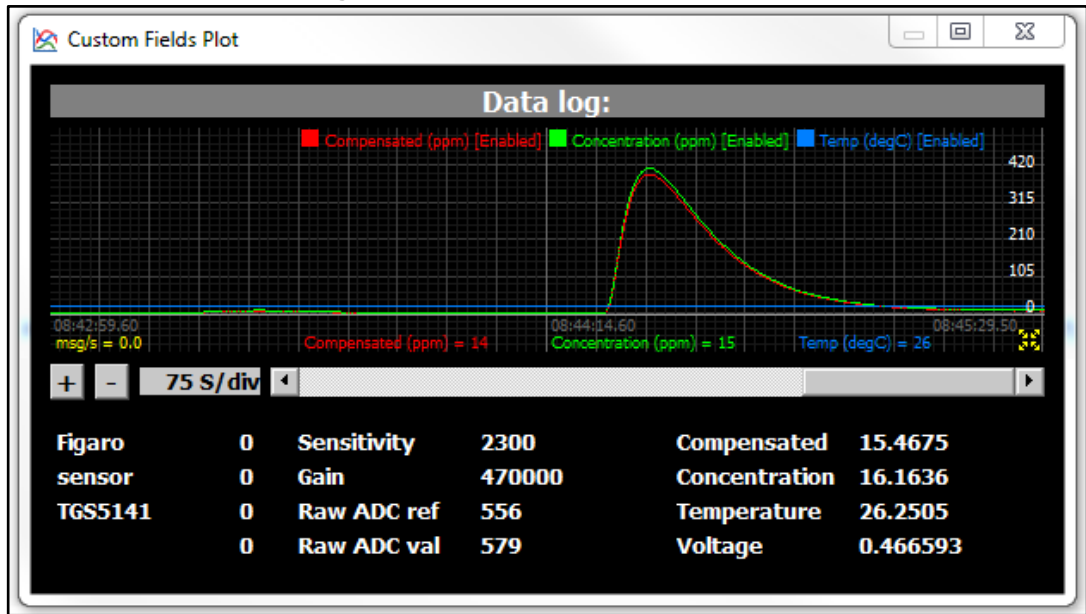
The dialog can be invoked by sending "s" to adjust the sensitivity or "g" to adjust the gain.

2.8 Unicleo GUI sample

The ST GUI application, Unicleo, available at www.st.com, contains a **Custom Fields** window which fully supports the P-NUCLEO-IKA02A1 expansion board to show the measured values in a graph.

^a See the P-NUCLEO-IKA02A1 hardware user manual for more information.

Figure 6: Unicleo Custom Fields window



3 Revision history

Table 1: Document revision history

Date	Version	Changes
05-Jul-2017	1	Initial release.
24-Oct-2017	2	Updated Section 1.1: "How does this software complement STM32Cube?" , Section 2.3: "Folder structure" and Section 2.4: "APIs" . Added Section 2.7: "Gas concentration reading sample" and Section 2.8: "Unicleo GUI sample" .

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2017 STMicroelectronics – All rights reserved