Getting started with the X-CUBE-ISPU software package for STM32Cube

Introduction

The X-CUBE-ISPU expansion software package for STM32Cube runs on the STM32. It includes drivers that recognize the sensors and collect temperature, humidity, pressure, motion, and ISPU data.

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-IKS01A3/X-NUCLEO-IKS02A1/X-NUCLEO-IKS4A1 expansion boards connected to a featured STM32 Nucleo development board.

The software is also available on GitHub, where the users can signal bugs and propose new ideas through [**Issues**] and [**Pull requests**] tabs.

The software provides sample applications for communication with the PC Unicleo-GUI graphical user interface.

The software also provides template and example code for the ISPU, and binary libraries with sample code for integration in the ISPU.

Related links

Visit the STM32Cube ecosystem web page on www.st.com for further information

1 License

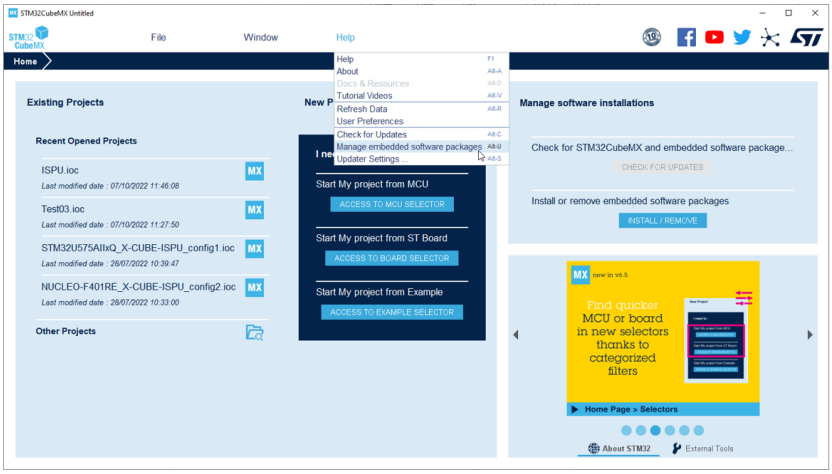
The software provided in this package is licensed under Software License Agreement SLA0095.

2 Installing the X-CUBE-ISPU pack in STM32CubeMX

After downloading (from www.st.com), installing and launching the STM32CubeMX, the X-CUBE-ISPU pack can be installed in a few steps.

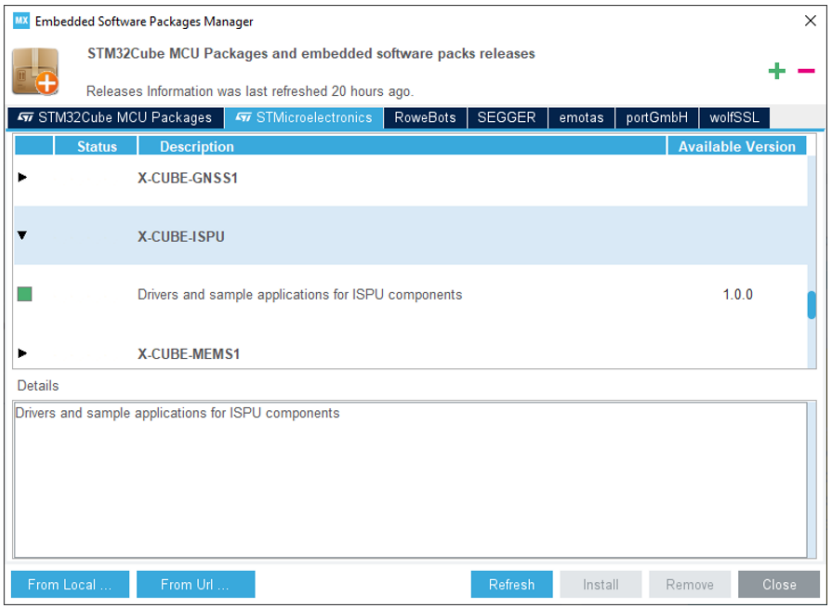
Step 1. From the menu, select [**Help**]>[**Manage embedded software packages**].

Figure 1. Managing embedded software packs in STM32CubeMX



Step 2. From the [**Embedded Software Packages Manager**] window, press the [**Refresh**] button to get an updated list of the add-on packs. Go to the [**STMicroelectronics**] tab to find the X-CUBE-ISPU pack.

Figure 2. Installing the X-CUBE-ISPU pack in STM32CubeMX



Step 3. Select it by checking the corresponding box and install it by pressing the [**Install Now**] button.

Once the installation is completed, the corresponding box becomes green, the [**Close**] button can be pressed, and the configuration of a new project can start.

3 Sample applications

The X-CUBE-ISPU contains sample applications, which:

- are ready-to-use projects that can be generated through the STM32CubeMX for any STM32 Nucleo development board, using the X-NUCLEO-IKS01A3, X-NUCLEO-IKS02A1 or the X-NUCLEO-IKS4A1 expansion board
- are ready-to-use projects that can be generated through the STM32CubeMX for any board equipped with an STM32 MCU, using supported MEMS components and MEMS devices with an ISPU core
- show the users how to use the APIs to initialize and use correctly the ST MEMS components and MEMS devices with an ISPU core

3.1 ISPU_DataLogExtended sample application

This application shows how to use the X-NUCLEO-IKS01A3, X-NUCLEO-IKS02A1, X-NUCLEO-IKS4A1 or a custom expansion board with MEMS devices and MEMS components with an ISPU core to send sensor data from an STM32 Nucleo development board, using a UART to a connected PC and displaying it in the Unicleo-GUI application.

The user can view the data from all on-board environmental sensors (temperature, humidity, and pressure sensors on the X-NUCLEO-IKS01A3 and the X-NUCLEO-IKS4A1) and from the on-board inertial sensors (magnetometer on the X-NUCLEO-IKS01A3, X-NUCLEO-IKS02A1 and the X-NUCLEO-IKS4A1), as well as data from one of the supported MEMS devices with an ISPU core (accelerometer and gyroscope), connected to the DIL24 socket or on the custom board.

The data can be displayed as text or in graphical plots and can be stored as a CSV file. The ISPU can be programmed with any algorithm stored in a UCF file.

3.2 ISPU_NEAI_AnomalyDetection sample application

This application shows how to integrate the anomaly detection library for the ISPU core generated by NanoEdgeAIStudio into the STM32 project. The output from the STM32 Nucleo development board is sent using a UART to a connected PC and displayed in a terminal application.

4 Hardware configuration

The X-NUCLEO-IKS01A3, X-NUCLEO-IKS02A1 or the X-NUCLEO-IKS4A1 interfaces with the STM32 microcontroller via the I²C bus. Therefore, no particular hardware modification is needed to interface the X-NUCLEO-IKS01A3, X-NUCLEO-IKS02A1 or the X-NUCLEO-IKS4A1 expansion board with a 64-pin STM32 Nucleo (for example, a NUCLEO-F401RETx) or a 144-pin STM32 Nucleo (for example, a NUCLEO-F429ZITx).

Figure 3. 64-pin STM32 Nucleo and X-NUCLEO-IKS02A1

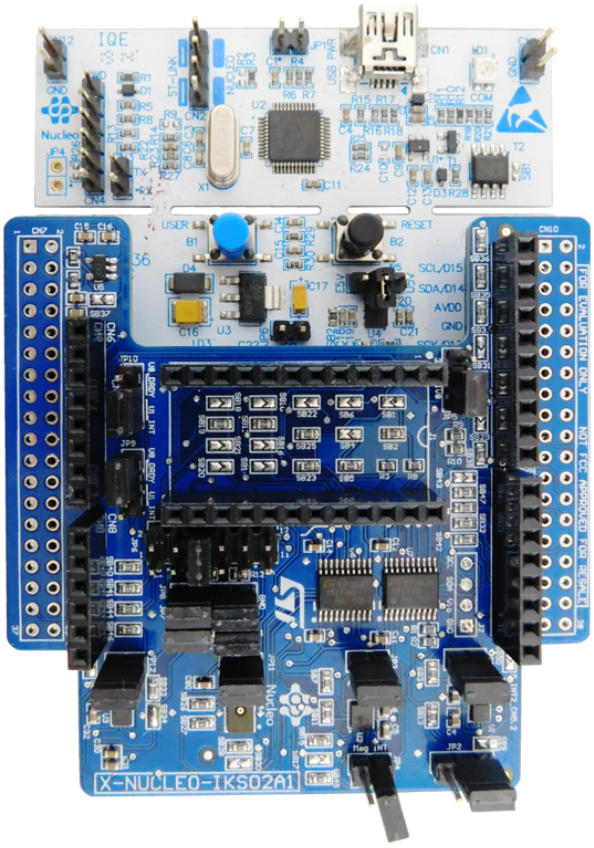
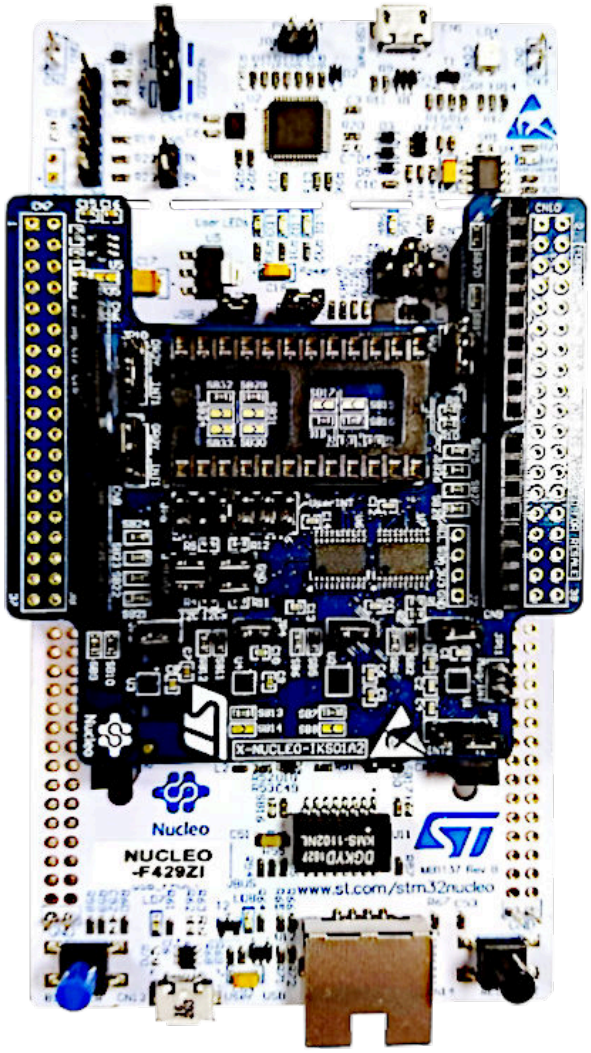


Figure 4. **144-pin STM32 Nucleo and X-NUCLEO-IKS01A3**



Specific configuration for ISPU_DataLogExtended and ISPU_NEAI_AnomalyDetection:

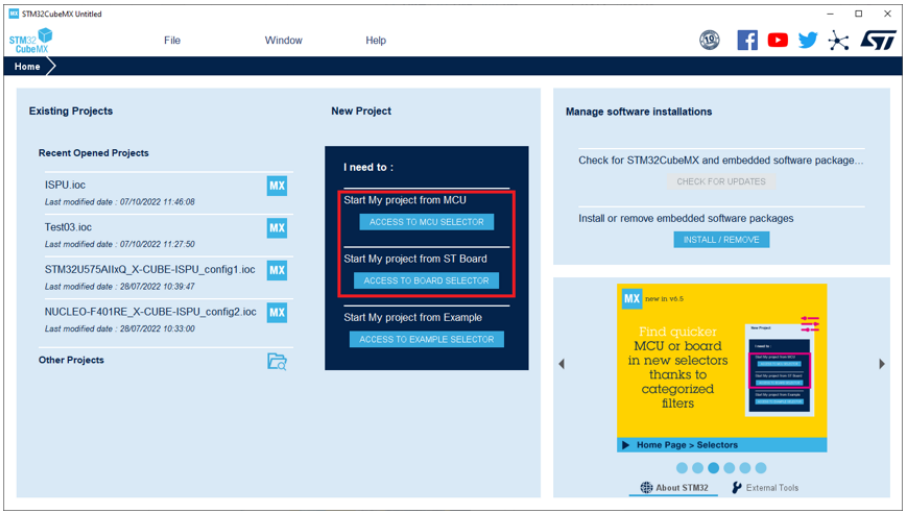
- **X-NUCLEO-IKS01A3**
  Ensure that the JP6 jumper is connected to the position #7 on the X-NUCLEO-IKS01A3 expansion board in order to connect the USER_INT pin that you selected on your MCU (in our example PA10) to the INT2 pin of the ISPU device.
- **X-NUCLEO-IKS02A1**
  Ensure that the JP6 jumper is connected to the position #3 on the X-NUCLEO-IKS02A1 expansion board in order to connect the USER_INT pin that you selected on your MCU (in our example PA10) to the INT2 pin of the ISPU device.
- **X-NUCLEO-IKS4A1**
  Ensure that the J6 jumper is connected to the position #3 on the X-NUCLEO-IKS4A1 expansion board in order to connect the USER_INT pin that you selected on your MCU (in our example PA10) to the INT2 pin of the ISPU device.
- **Custom board**
  Ensure that the USER_INT pin that you selected on your MCU (in our example PA10) is connected to the INT2 pin of the ISPU device.

5 Project generation

5.1 MCU/board selection

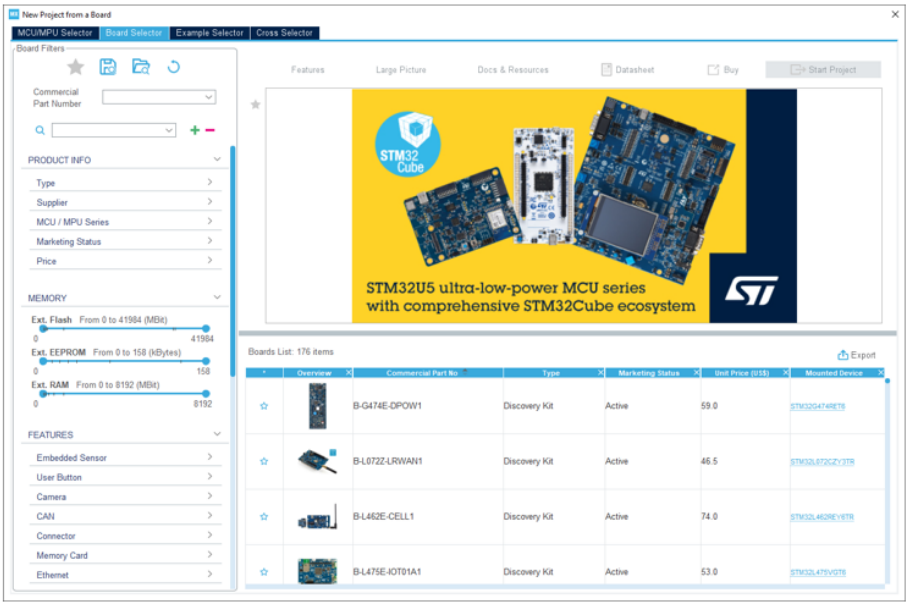
The project generation starts with the MCU or board selection. After launching the STM32CubeMX, you can choose whether to start a [**New Project**] from the [**MCU Selector**] or from the [**Board Selector**].

Figure 5. STM32CubeMX main page



The [**MCU/Board selector**] window pops up. From this window, you can select the STM32 MCU or board.

Figure 6. STM32CubeMX MCU/board selector window



After selecting the MCU or the board, you can configure the MCU peripherals, pinout, clock, add one or more software packs and set up the generated project.

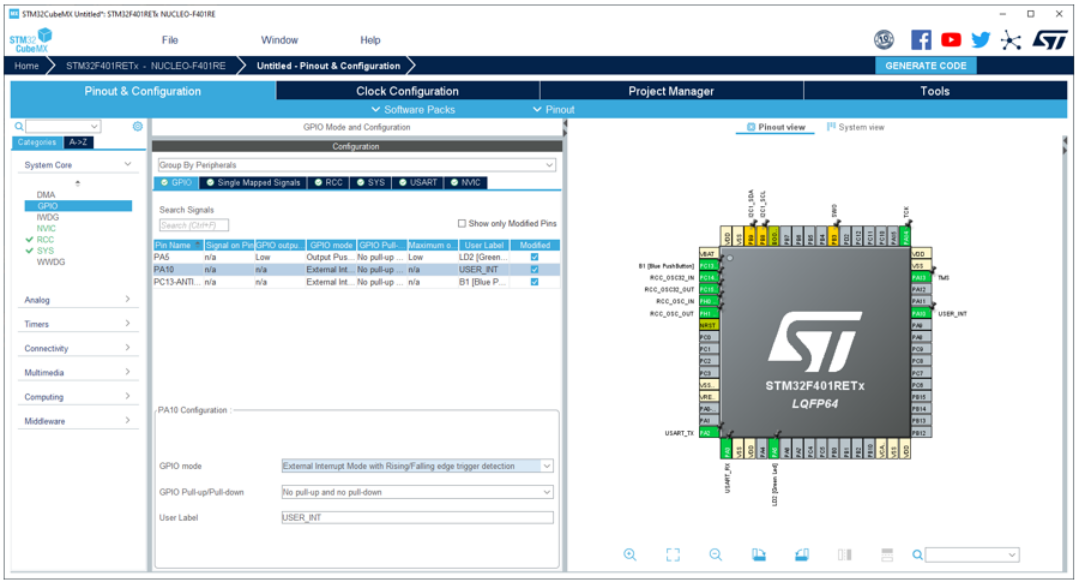
5.2 Pin configuration

The typical pin configuration for the STM32 Nucleo-64 development boards is:

- PB8 (SDA) and PB9 (SCL) - I²C bus for MEMS components
- PA10 - EXTI input for the user interrupt
  - For ISPU_DataLogExtended can be renamed to USER_INT (not mandatory)
    - GPIO Mode: External Interrupt Mode Rising/Faling edge trigger detection
  - For ISPU_NEAI_AnomalyDetection can be renamed to DIL24_INT2 (not mandatory)
    - GPIO Mode: External Interrupt Mode Rising edge trigger detection
- PC0 - EXTI input for the user interrupt
  - For ISPU_DataLogExtended is not needed
  - For ISPU_NEAI_AnomalyDetection can be renamed to DIL24_INT1 (not mandatory)
    - GPIO Mode: External Interrupt Mode Rising edge trigger detection
- PA3 (Rx), PA2 (Tx) - USART2 for the communication with a PC
- PA5 - GPIO output for the LED pin
- PC13 - EXTI input for the user button

From the [**Pinout & Configuration**] tab, click on the [**System Core**] category and then on the GPIO item to configure the GPIO pins according to following figure.

Figure 7. GPIO configuration

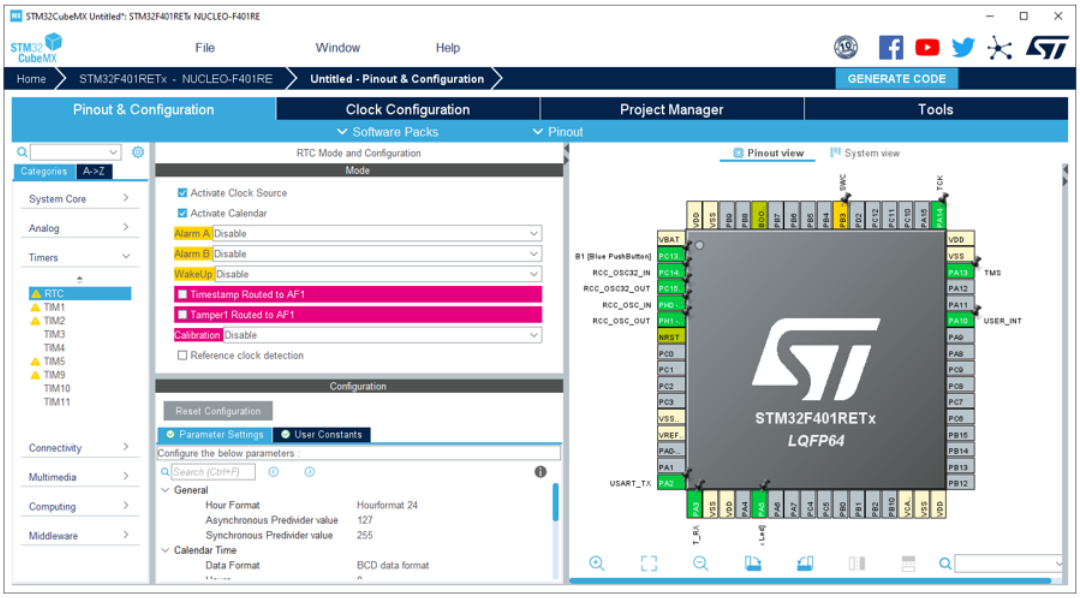


5.3 RTC configuration

Note: RTC is not mandatory for ISPU_NEAI_AnomalyDetection.

From the [**Pinout & Configuration**] tab, click on the [**[Timers]**] category and then enable Activate Clock Source and Activate Calendar option for the RTC peripheral.

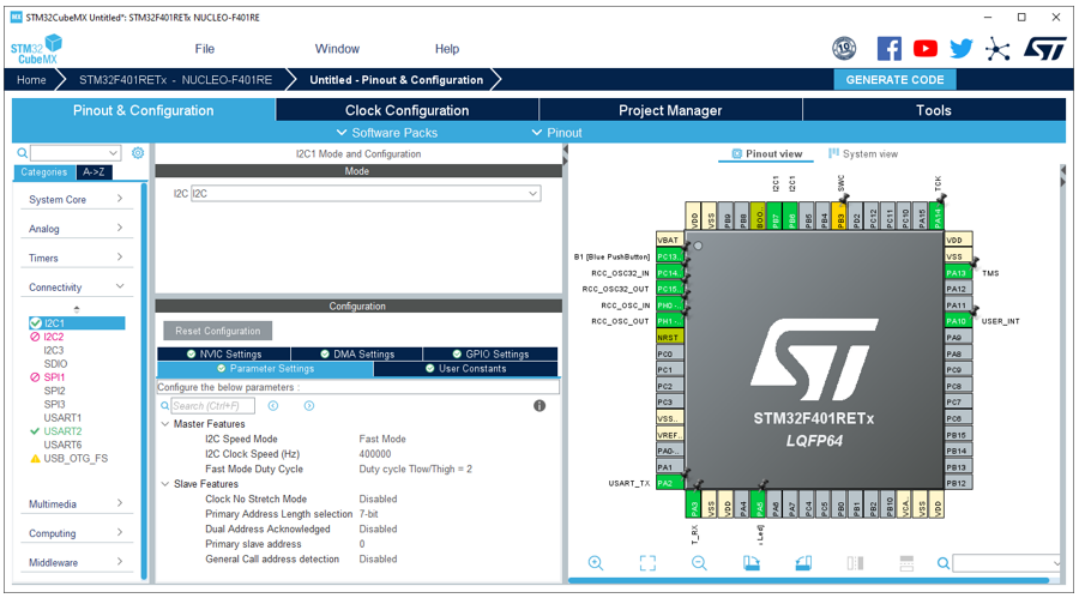
Figure 8. **RTC configuration**



5.4 I²C configuration

From the [**Pinout & Configuration**] tab, click on the [**Connectivity**] category and then enable the I2C1 item and set the I²C speed at 400 KHz (fast mode).

Figure 9. **I²C configuration**



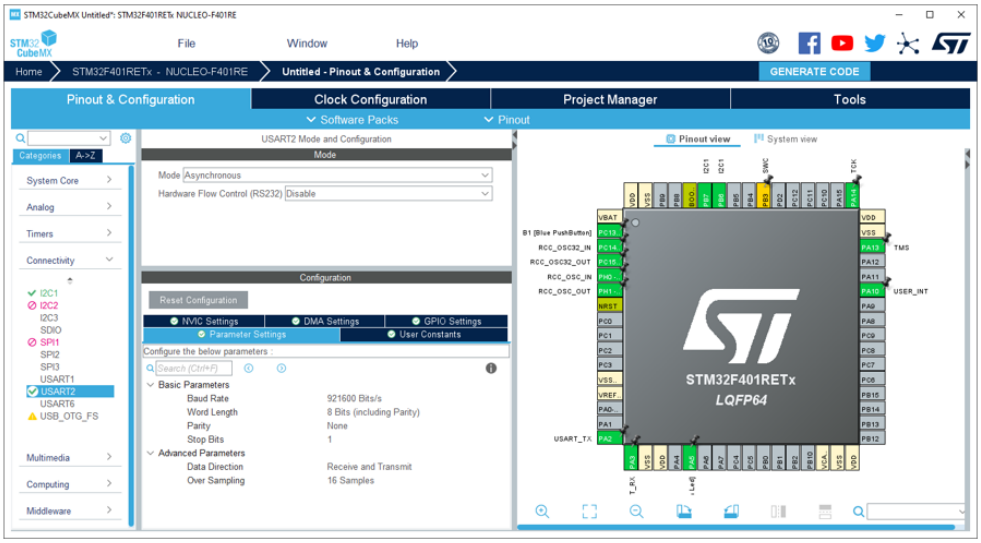
5.5 USART configuration

From the [**Pinout & Configuration**] tab, click on the [**Connectivity**] category and then configure the USART2 according to the configuration shown in the table below.

Table 1. USART configuration parameters

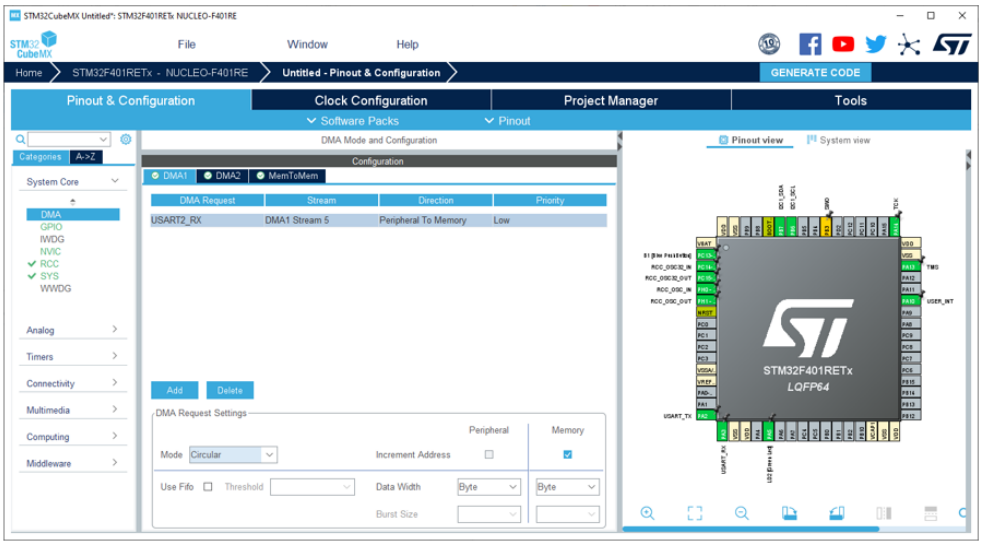
Parameter	Value
Baud rate	921600 bits/s - ISPU_DataLogExtended 115200 bits/s - ISPU_NEAI_AnomalyDetection
Word length	8 bits (including parity)
Parity	None
Stop bits	1

Figure 10. USART configuration



From the [**Pinout & Configuration**] tab, click on the [**System Core**] category and then on the DMA item. Then, set up the DMA for the USART2_RX in circular mode, incrementing the memory address as shown below.

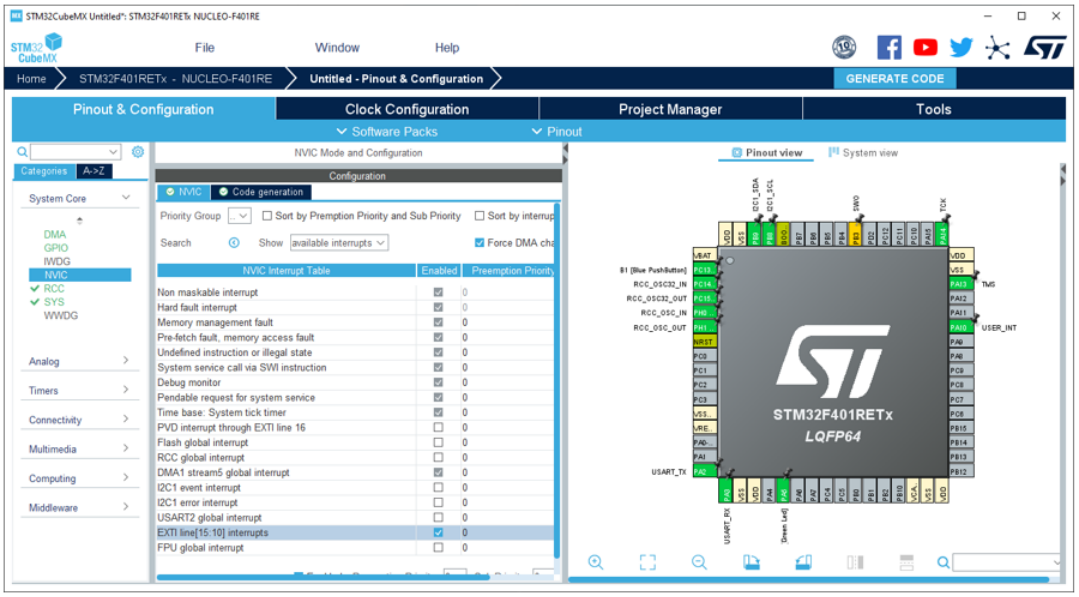
Figure 11. DMA configuration



5.6 NVIC configuration

From the [**Pinout & Configuration**] tab, click on the [**System Core**] category and then on the NVIC item to enable the EXTI line interrupt on the pin which is connected to the INT2 of the ISPU device.

Figure 12. NVIC configuration



5.7 Adding software components

To add software components from the X-CUBE-ISPU software pack to the project, click on the [**Software Packs**] button. Then, click on the [**Select Components**] item.

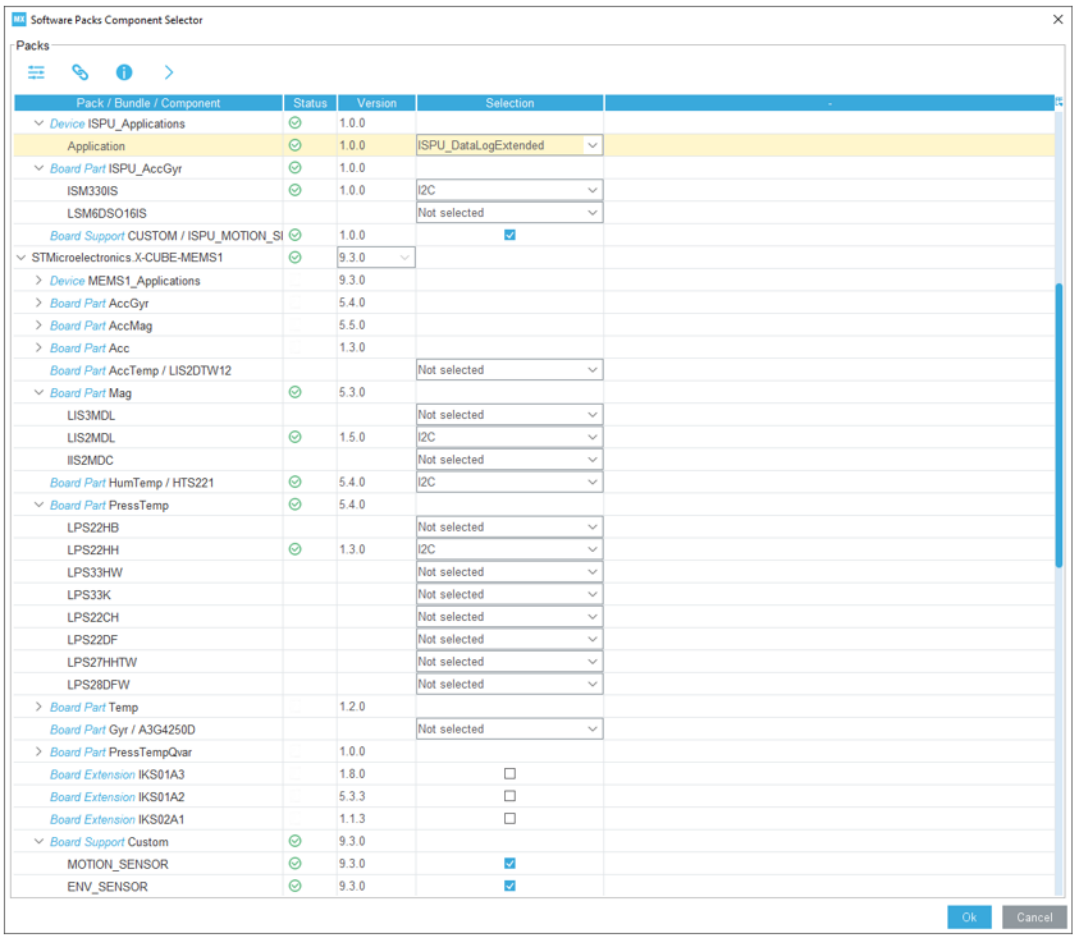
From the [**Software Packs Component Selector**] window, you can add the drivers for individual MEMS components only, as well as the BSP abstraction layer, or the fully functional applications.

5.7.1 Software components for X-NUCLEO-IKS01A3

This section outlines how to configure the software component for an X-NUCLEO-IKS01A3 expansion board that mounts environmental and motion MEMS sensors from the X-CUBE-MEMS1 pack and MEMS component with an ISPU core in a DIL24 socket from the X-CUBE-ISPU pack.

In this case, you can configure the MEMS with an ISPU device to be used via I²C bus (default for the expansion board) or via SPI bus (only four wires).

Figure 13. [Software Packs Component Selector] window for X-NUCLEO-IKS01A3

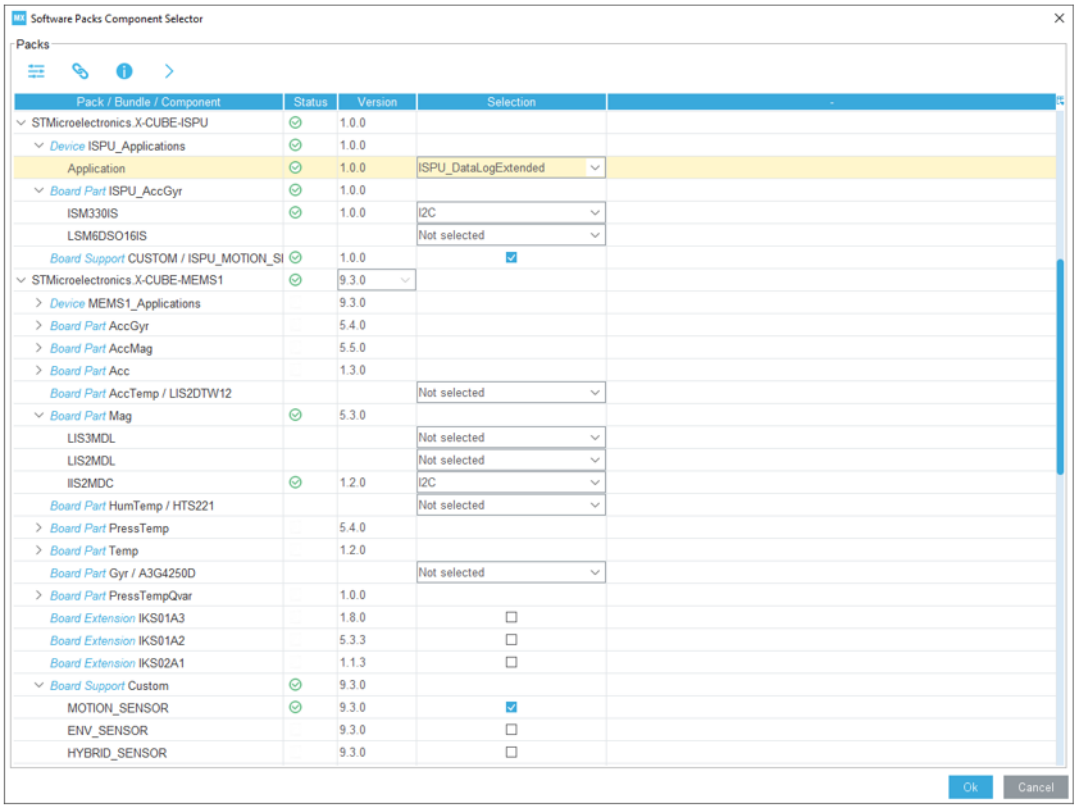


5.7.2 Software components for X-NUCLEO-IKS02A1

This section outlines how to configure the software component for an X-NUCLEO-IKS02A1 expansion board that mounts motion MEMS sensors from the X-CUBE-MEMS1 pack and MEMS component with an ISPU core in a DIL24 socket from the X-CUBE-ISPU pack.

In this case, you can configure the MEMS with an ISPU device to be used via I²C bus (default for the expansion board) or via SPI bus (only four wires).

Figure 14. [Software Packs Component Selector] window for X-NUCLEO-IKS02A1

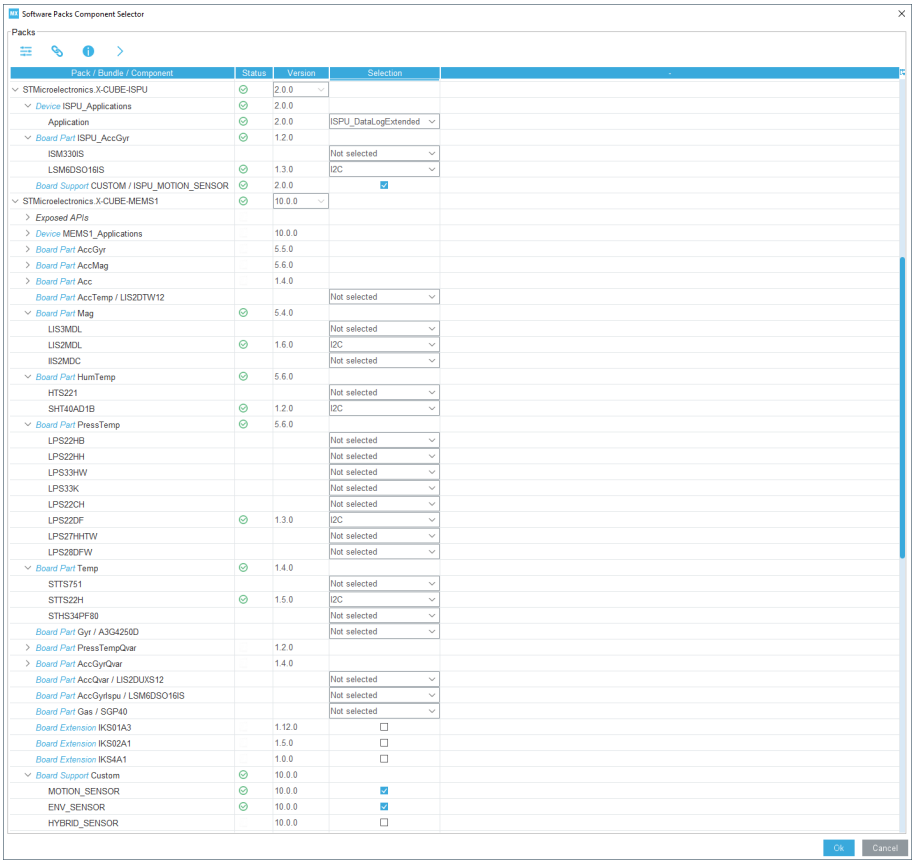


5.7.3 Software components for X-NUCLEO-IKS4A1

his section outlines how to configure the software component for an X-NUCLEO-IKS4A1 expansion board that mounts environmental and motion MEMS sensors from the X-CUBE-MEMS1 pack and MEMS component with an ISPU core from the X-CUBE-ISPU pack.

In this case, you can configure the MEMS with an ISPU device to be used via I²C bus (default for the expansion board) or via SPI bus (only four wires).

Figure 15. [Software Packs Component Selector] window for X-NUCLEO-IKS4A1

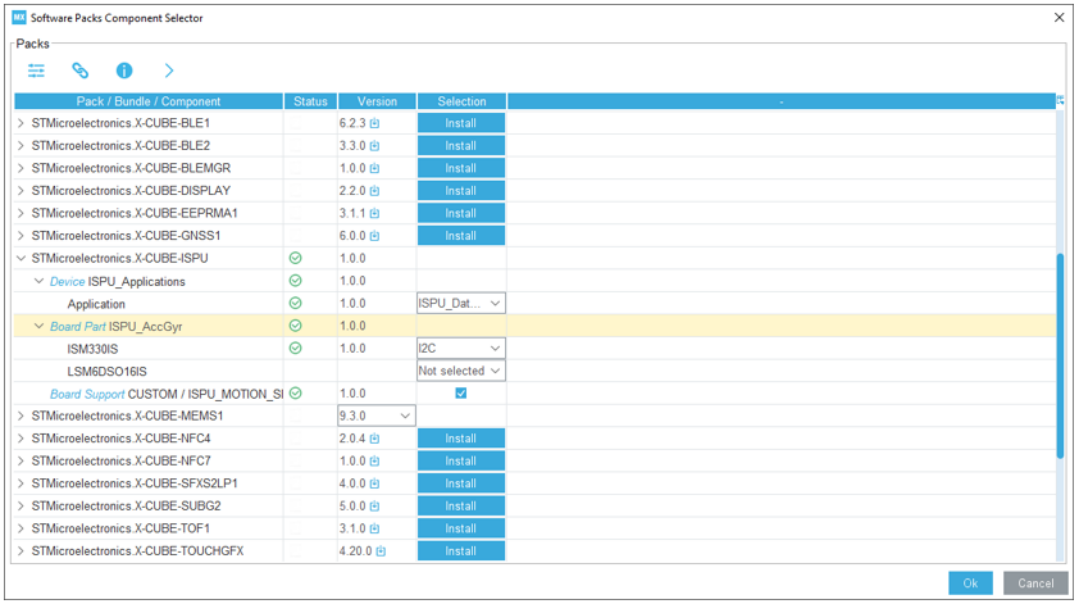


5.7.4 Software components for a custom board

This section outlines how to configure the software component for a custom board that mounts a MEMS device with an ISPU core.

In this case, you can configure the MEMS with an ISPU device to be used via I²C bus or via SPI bus (only four wires).

Figure 16. [Software Packs Component Selector] window for a custom board



5.8 Configuration of the software components

From the [**Pinout and Configuration**] tab, click on the [**Software Packs**] category. Then, press the [**STMicroelectronics.X-CUBE-ISPU.x.x.x**] item. Enable all the [**Board Part**] elements, the [**Board Support Custom**], and the [**Device ISPU Applications**] checkboxes from the [**Mode**] view. Set the following platform settings from the [**Configuration**] view, according to the selected sensor. Assign the configured peripherals to the required peripherals by the selected application. The figure below shows a configuration example.

Figure 17. Parameter settings for ISPU_DataLogExtended (ISM330IS)

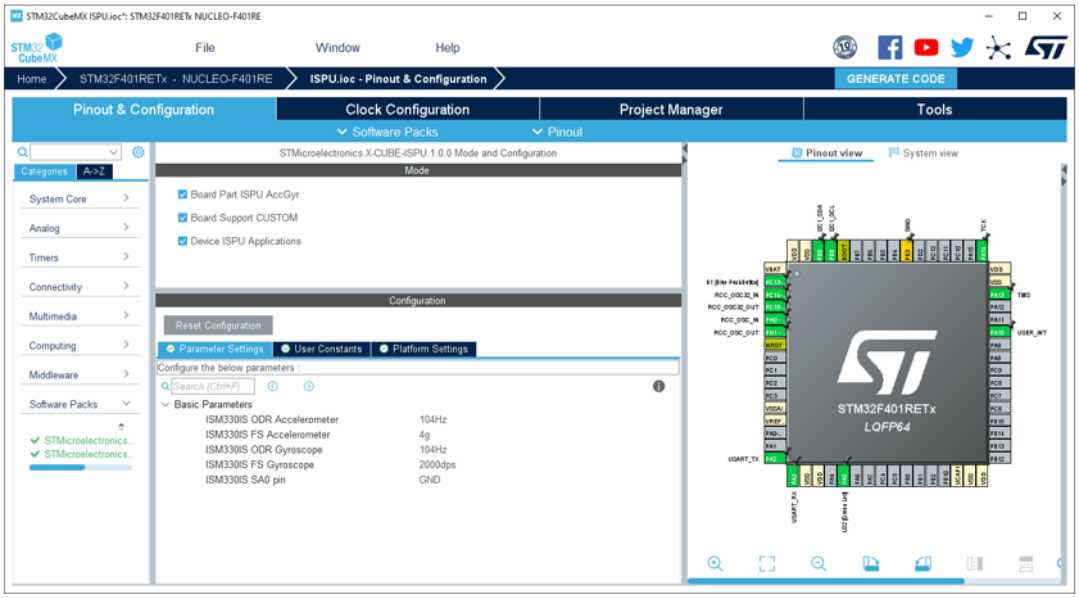
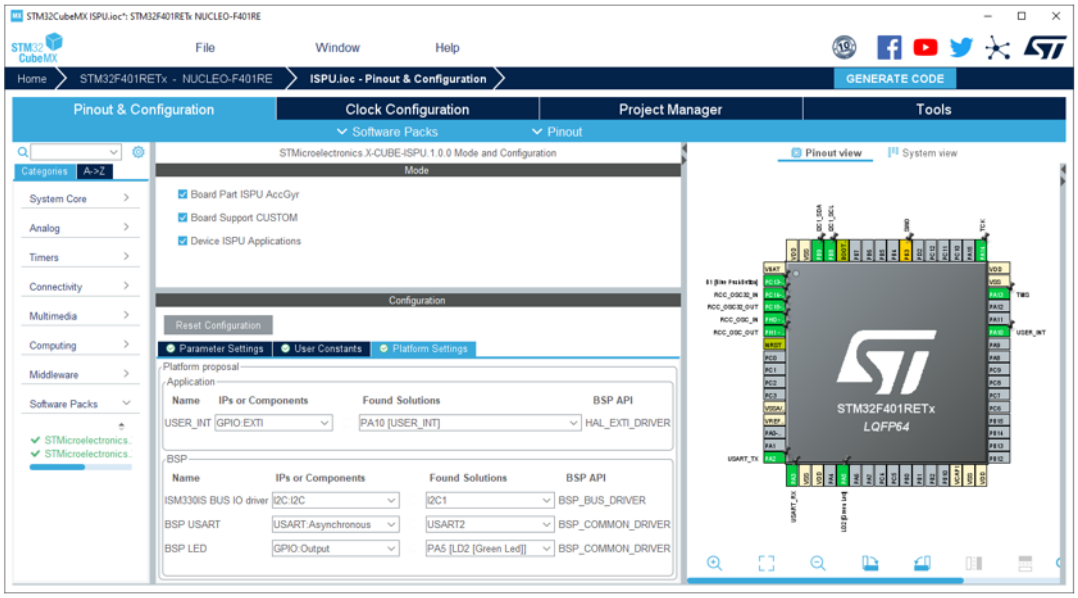


Figure 18. Platform settings for ISPU_DataLogExtended



5.8.1 Configuration for X-NUCLEO-IKS01A3

From the [**Pinout and Configuration**] tab, click on the [**Software Packs**] category. Then, press the [**STMicroelectronics.X-CUBE-MEMS1.x.x.x**] item. Enable all the [**Board Part**] elements, the [**Board Support Custom**] from the [**Mode**] view.

From the [**Parameter Settings**] tab, you can change some parameters of the MEMS devices. For example, you can decide if the SA0 pin is connected to VDD or GND, in order to associate the correct I²C address to the MEMS device.

The figure below shows the software pack settings of the ISPU_DataLogExtended application, when using the X-NUCLEO-IKS01A3.

Figure 19. Parameter settings - X-NUCLEO-IKS01A3 (MEMS1)

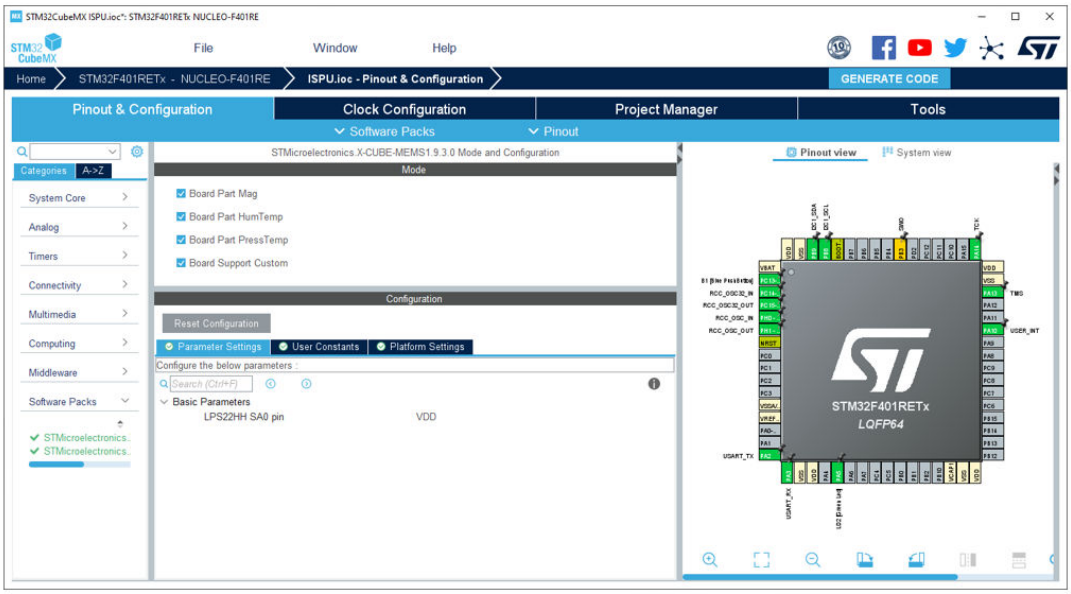
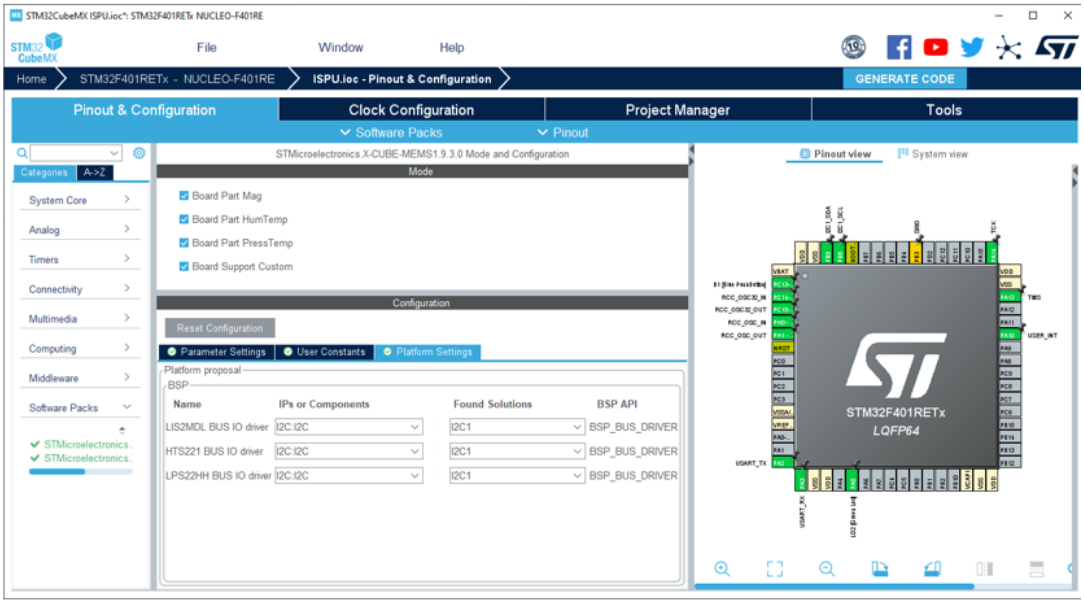


Figure 20. Platform settings - X-NUCLEO-IKS01A3 (MEMS1)



5.8.2 Configuration for X-NUCLEO-IKS02A1

From the [**Pinout and Configuration**] tab, click on the [**Software Packs**] category. Then, press the [**STMicroelectronics.X-CUBE-MEMS1.x.x.x**] item. Enable all the [**Board Part**] elements, the [**Board Support Custom**] from the [**Mode**] view.

From the [**Parameter Settings**] tab, you can change some parameters of the MEMS devices. For example, you can decide if the SA0 pin is connected to VDD or GND, in order to associate the correct I²C address to the MEMS device.

The figure below shows the software pack settings for the ISPU_DataLogExtended application, when using the X-NUCLEO-IKS02A1 (MEMS1).

Figure 21. Parameter settings - X-NUCLEO-IKS02A1 (MEMS1)

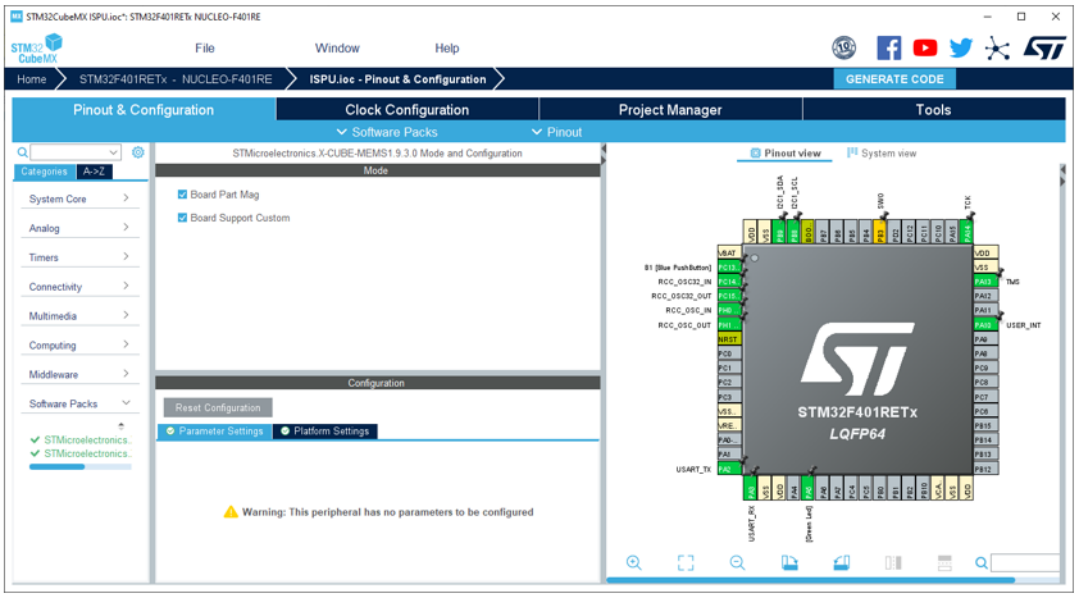
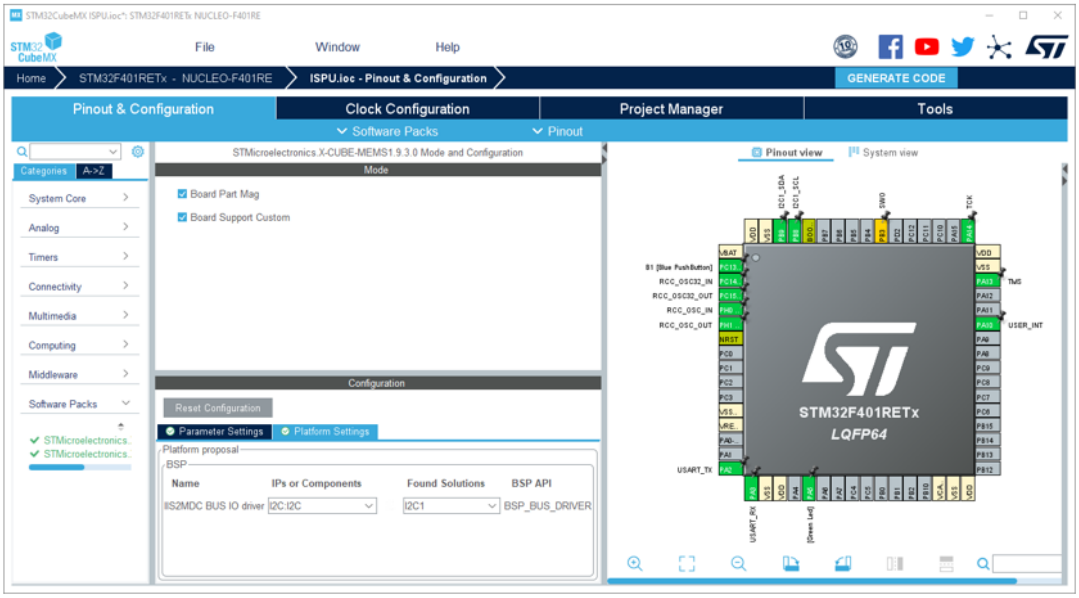


Figure 22. Platform settings - X-NUCLEO-IKS02A1



5.8.3 Configuration for X-NUCLEO-IKS4A1

From the [Pinout and Configuration] tab, click on the [Software Packs] category. Then, press the [STMicroelectronics.X-CUBE-MEMS1.x.x.x] item. Enable all the [Board Part] elements, the [Board Support Custom] from the [Mode] view.

From the [Parameter Settings] tab, you can change some parameters of the MEMS devices. For example, you can decide if the SA0 pin is connected to VDD or GND, in order to associate the correct I²C address to the MEMS device.

The figure below shows the software pack settings of the ISPU_DataLogExtended application, when using the X-NUCLEO-IKS4A1.

Figure 23. Parameter settings - X-NUCLEO-IKS4A1 (MEMS1)

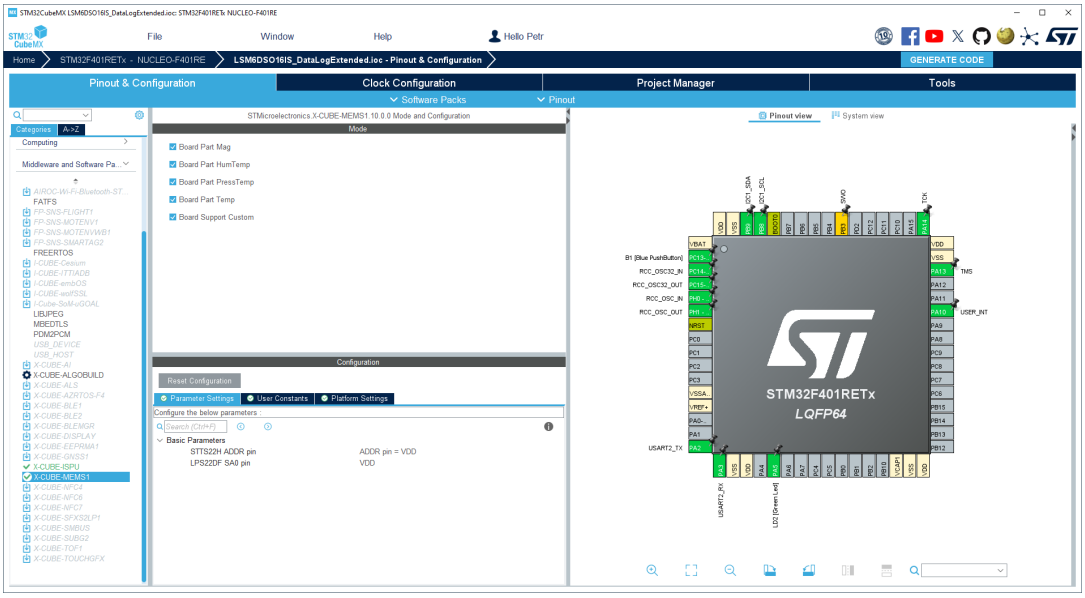
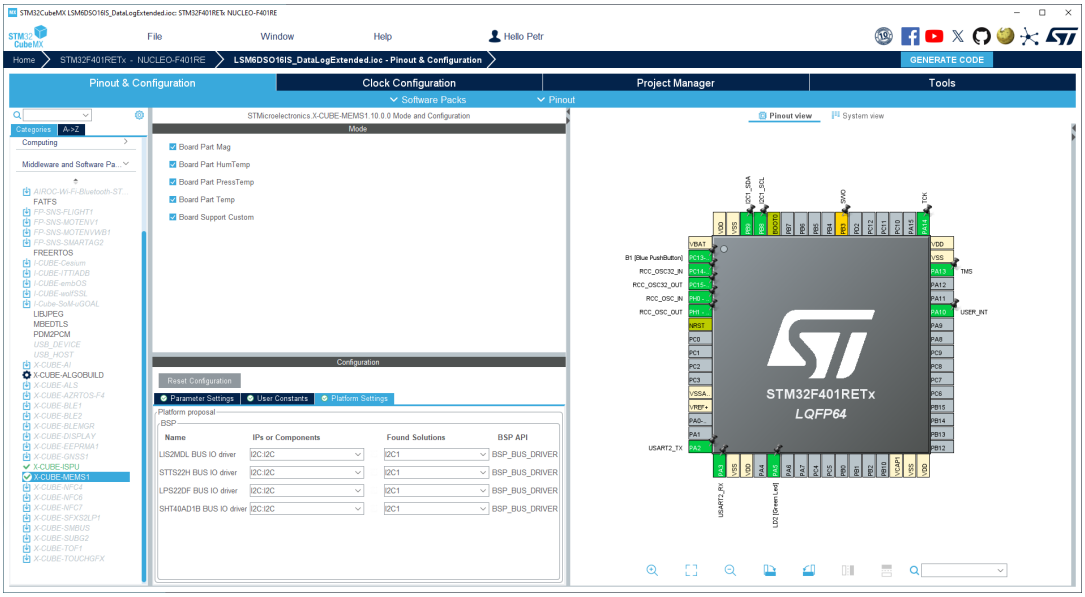


Figure 24. Platform settings - X-NUCLEO-IKS4A1 (MEMS1)



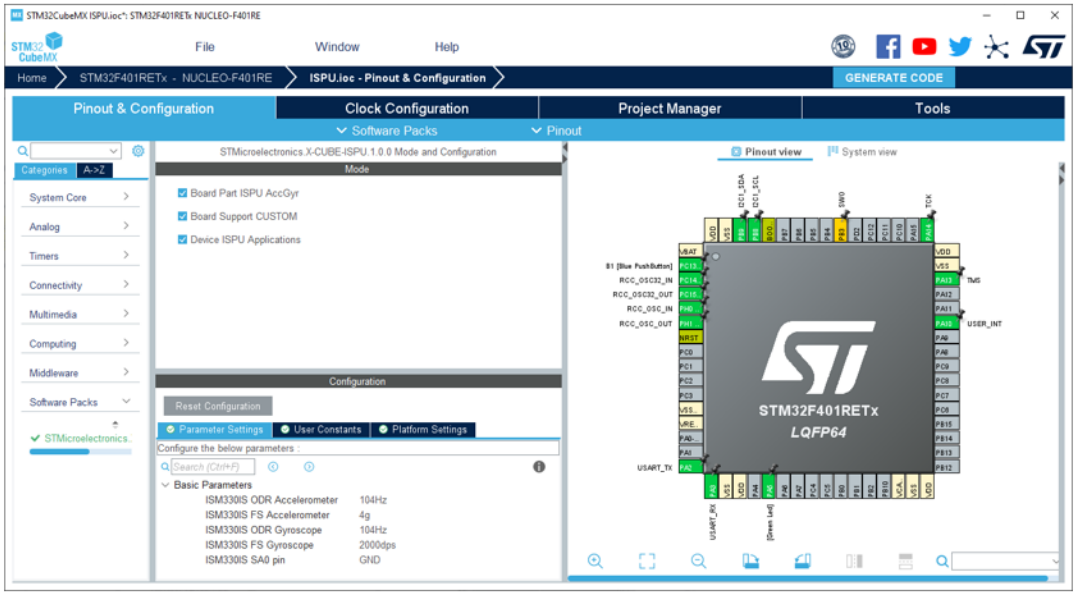
5.8.4 Configuration for the custom board

From the [**Pinout and Configuration**] tab, click on the [**Software Packs**] category. Then, press the [**STMicroelectronics.X-CUBE-MEMS1.x.x.x**] item. Enable all the [**Board Part**] elements, the [**Board Support Custom**] from the [**Mode**] view.

From the [**Parameter Settings**] tab, you can change some parameters of the MEMS devices. For example, you can decide if the SA0 pin is connected to VDD or GND, in order to associate the correct I²C address to the MEMS device.

The figure below shows the parameter settings of the ISPU_DataLogExtended application for a custom board.

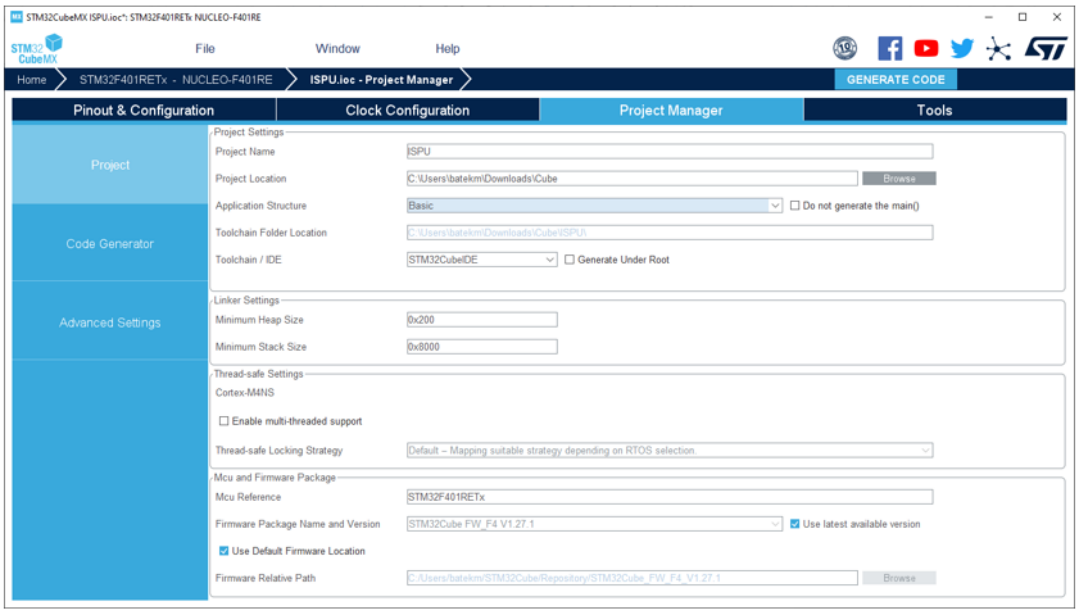
Figure 25. Parameter settings - custom board



5.9 Project generation

Once all the above-described steps have been performed, in the **Project Manager** tab increase the Minimum Stack Size to 0x8000, select the preferred Toolchain and the sample application using the **X-CUBE-ISPU** software can be generated clicking the “GENERATE CODE” button.

Figure 26. Project manager configuration



5.10 Project usage

The generated project can be opened in the selected toolchain and compiled. After compilation, the target device can be programmed with the generated binary output.

ISPU_DataLogExtended firmware cooperates with Unicleo-GUI applications. So, you need to use this application to test the ISPU device.

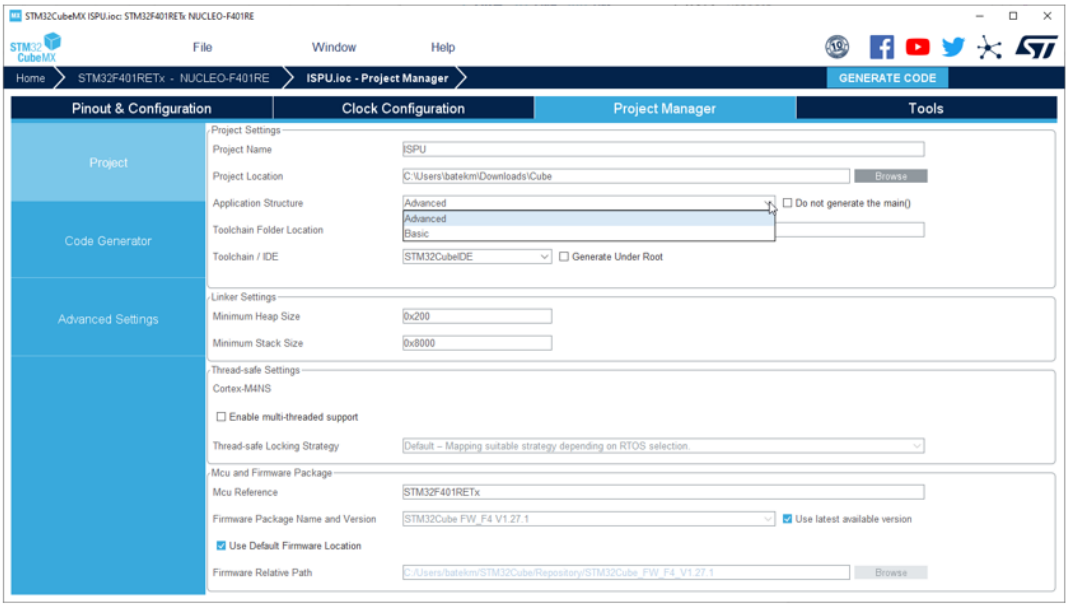
ISPU_NEAI_AnomalyDetection firmware can be tested with any serial terminal application.

6 Generated folder structure

When generating a project, you can adopt two models of folder structure if you are using a high-level firmware component (a middleware in the STM32CubeMX MCU package):

- **Basic tructure**, which is often used with the HAL examples and single package projects. This structure consists in having the IDE configuration folder at the same level of the sources (organized in the *Inc* and *Src* subfolders).
- **Advanced structure**, which provides a more efficient and organized folder model that allows an easy middleware application integration when several packages are used.
  In the advanced mode, *Src* and *Inc* are generated under the *Core* folder.
  For each package, the list of the generated files is under <*ModuleName*> (*ISPU* for the X-CUBE-ISPU pack). They are at the same level of the *Core* and contain the *App* and the *Target* subfolders.

Figure 27. STM32CubeMX application structure configuration



7 Known limitations and workarounds

On the dual-core STM32 series, this expansion software can be used on both cores but exclusively.

The projects for the MDK-ARM toolchain can be generated by default with the MicroLIB disabled for some STM32 families. In this case, the MEMS application, which sends data to the VCOM, does not work properly. To avoid this issue, check that the [**Use MicroLIB**] option is always enabled in the MDK-ARM project options.

Revision history

Table 2. **Document revision history**

Date	Revision	Changes
01-Aug-2022	1	Initial release.
24-Oct-2022	2	Updated *Section 2 Installing the X-CUBE-ISPU pack in STM32CubeMX, Section 4 Hardware* *configuration, Section 5.1 MCU/board selection, Section 5.2 Pin configuration, Section 5.4 I²C* *configuration, Section 5.5 USART configuration, Section 5.6 NVIC configuration, Section 5.7 Adding* *software components, Section 5.7.1 Software components for X-NUCLEO-IKS01A3,* *Section 5.7.2 Software components for X-NUCLEO-IKS02A1, Section 5.7.4 Software components* *for a custom board, Section 5.8 Configuration of the software components,* *Section 5.8.1 Configuration for X-NUCLEO-IKS01A3, Section 5.8.2 Configuration for X-NUCLEOIKS02A1,* *Section 5.8.4 Configuration for the custom board, Section 5.9 Project generation, and* *Section 6 Generated folder structure.* Added *Section 3.2 ISPU_NEAI_AnomalyDetection sample application, Section 5.3 RTC* *configuration,* and *Section 5.10 Project usage.*
07-Dec-2023	3	Updated Section Introduction, Section 3 Sample applications, Section 3.1 ISPU_DataLogExtended sample application, Section 4 Hardware configuration, Section 5.2 Pin configuration and Section 5.7.2 Software components for X-NUCLEO-IKS02A1. Added Section 5.7.3 Software components for X-NUCLEO-IKS4A1 and Section 5.8.3 Configuration for X-NUCLEO-IKS4A1.

Contents

1 License . . . 2
2 Installing the X-CUBE-ISPU pack in STM32CubeMX . . . 3
3 Sample applications . . . 4
3.1 ISPU_DataLogExtended sample application . . . 4
3.2 ISPU_NEAI_AnomalyDetection sample application . . . 4
4 Hardware configuration . . . 5
5 Project generation . . . 7
5.1 MCU/board selection . . . 7
5.2 Pin configuration . . . 8
5.3 RTC configuration . . . 8
5.4 I²C configuration . . . 9
5.5 USART configuration . . . 10
5.6 NVIC configuration . . . 11
5.7 Adding software components . . . 12
5.7.1 Software components for X-NUCLEO-IKS01A3 . . . 12
5.7.2 Software components for X-NUCLEO-IKS02A1 . . . 13
5.7.3 Software components for X-NUCLEO-IKS4A1 . . . 14
5.7.4 Software components for a custom board . . . 15
5.8 Configuration of the software components . . . 16
5.8.1 Configuration for X-NUCLEO-IKS01A3 . . . 17
5.8.2 Configuration for X-NUCLEO-IKS02A1 . . . 18
5.8.3 Configuration for X-NUCLEO-IKS4A1 . . . 19
5.8.4 Configuration for the custom board . . . 20
5.9 Project generation . . . 21
5.10 Project usage . . . 21
6 Generated folder structure . . . 22
7 Known limitations and workarounds . . . 23
Revision history . . . 24
List of tables . . . 26
List of figures . . . 27

List of tables

Table 1. USART configuration parameters 10
Table 2. Document revision history 24

List of figures

Figure 1. Managing embedded software packs in STM32CubeMX . . . 3
Figure 2. Installing the X-CUBE-ISPU pack in STM32CubeMX . . . 3
Figure 3. 64-pin STM32 Nucleo and X-NUCLEO-IKS02A1 . . . 5
Figure 4. 144-pin STM32 Nucleo and X-NUCLEO-IKS01A3 . . . 6
Figure 5. STM32CubeMX main page . . . 7
Figure 6. STM32CubeMX MCU/board selector window . . . 7
Figure 7. GPIO configuration . . . 8
Figure 8. RTC configuration . . . 9
Figure 9. I²C configuration . . . 9
Figure 10. USART configuration . . . 10
Figure 11. DMA configuration . . . 10
Figure 12. NVIC configuration . . . 11
Figure 13. [**Software Packs Component Selector**] window for X-NUCLEO-IKS01A3 . . . 12
Figure 14. [**Software Packs Component Selector**] window for X-NUCLEO-IKS02A1 . . . 13
Figure 15. [**Software Packs Component Selector**] window for X-NUCLEO-IKS4A1 . . . 14
Figure 16. [**Software Packs Component Selector**] window for a custom board . . . 15
Figure 17. Parameter settings for ISPU_DataLogExtended (ISM330IS) . . . 16
Figure 18. Platform settings for ISPU_DataLogExtended . . . 16
Figure 19. Parameter settings - X-NUCLEO-IKS01A3 (MEMS1) . . . 17
Figure 20. Platform settings - X-NUCLEO-IKS01A3 (MEMS1) . . . 17
Figure 21. Parameter settings - X-NUCLEO-IKS02A1 (MEMS1) . . . 18
Figure 22. Platform settings - X-NUCLEO-IKS02A1 . . . 18
Figure 23. Parameter settings - X-NUCLEO-IKS4A1 (MEMS1) . . . 19
Figure 24. Platform settings - X-NUCLEO-IKS4A1 (MEMS1) . . . 19
Figure 25. Parameter settings - custom board . . . 20
Figure 26. Project manager configuration . . . 21
Figure 27. STM32CubeMX application structure configuration . . . 22