UM1901

User manual

Getting started with the software package for digital MEMS microphones in X-CUBE-MEMSMIC1 expansion for STM32Cube

Introduction

The X-CUBE-MEMSMIC1 software provides the complete STM32 middleware to build applications using digital MEMS microphones. The software expands the STM32Cube range of solutions and is easily ported across different MCU families.

The package contains sample applications for the acquisition of PDM signals from up to four digital MEMS microphones, PDM to PCM conversion and real time streaming of audio data to a PC via a standard USB audio-input driver.

The package includes some advanced processing application examples which exploit the ST AcousticBF and AcousticSL libraries to provide specific examples of beamforming and sound source localization functions.

The software provides implementation examples for STM32 Nucleo platforms equipped with the X-NUCLEO-CCA02M1 expansion board, featuring two on-board MEMS microphones (MP34DT01-M) as well as headers to connect additional microphones via ST coupon boards based on the digital MEMS microphone(see STEVAL-MKI129Vx and STEVAL-MKI155Vx data briefs on www.st.com).

1 Acronyms and abbreviations

Table 1. List of acronyms

Acronym	Description
PDM	Pulse density modulation
PCM	Pulse code modulation
USB	Universal serial bus
SPI	Serial peripheral interface
I²S	Integrated interchip sound
BSP	Board support package
HAL	Hardware abstraction layer
IDE	Integrated development environment

2 X-CUBE-MEMSMIC1 software expansion for STM32Cube

2.1 Overview

The X-CUBE-MEMSMIC1 software package expands STM32Cube functionality with the following key features:

- Complete middleware to build applications using MEMS digital microphones (MP34DT01-M)
- Easy portability across different MCU families thanks to STM32Cube
- Audio input class USB driver to allow the recognition of the device as a standard USB microphone and enable audio streaming
- PC-based streaming using third-party standard audio editors
- Free, user-friendly license terms
- Microphone acquisition sample implementation available on the X-NUCLEO-CCA02M1 board when connected to a NUCLEO-F401RE, NUCLEO-L476RG or NUCLEO-F746ZG board
- Advanced processing applications based on ST acoustic libraries for NUCLEO-F401RE, including AcousticBF (real time beamforming) sample and AcousticSL (real time sound source localization) sample

This software enables the acquisition of up to four digital MEMS microphones through I²S, SPI or DFSDM peripherals and performs PDM to PCM format conversion, the main standard for audio communication and processing.

Exploiting the capabilities of the included audio-input USB driver, the device is recognized as a standard multichannel USB microphone by Windows® or any Unix-like system; it can perform signal streaming to a host system for data recording and further processing using any standard audio editor, even if any software with a standard USB audio interface can be used to interact with the device.

2.2 Architecture

This STM32Cube expansion enables development of applications using digital MEMS microphones.

It is based on the STM32CubeHAL hardware abstraction layer for the STM32 microcontroller and extends STM32Cube with a specific board support package (BSP) for the microphone expansion board and middleware components for audio processing and USB communication with a PC.

The software layers used by the application software to access and use the microphone expansion board are:

- STM32Cube HAL layer: provides a generic, multi-instance set of APIs to interact with the upper layers (the application, libraries and stacks). It consists of generic and extension APIs based on a common architecture which allows other layers like the middleware layer to function without specific Microcontroller Unit (MCU) hardware configurations. This structure improves library code reusability and guarantees easy device portability.
- Board Support Package (BSP) layer: is a set of APIs which provides a programming interface for certain board specific peripherals (LED, user button etc.). This interface also helps in identifying the specific board version and provides support for initializing required MCU peripherals and reading data.

For the microphone expansion board, it provides the interface for MP34DT01-M digital MEMS microphones.

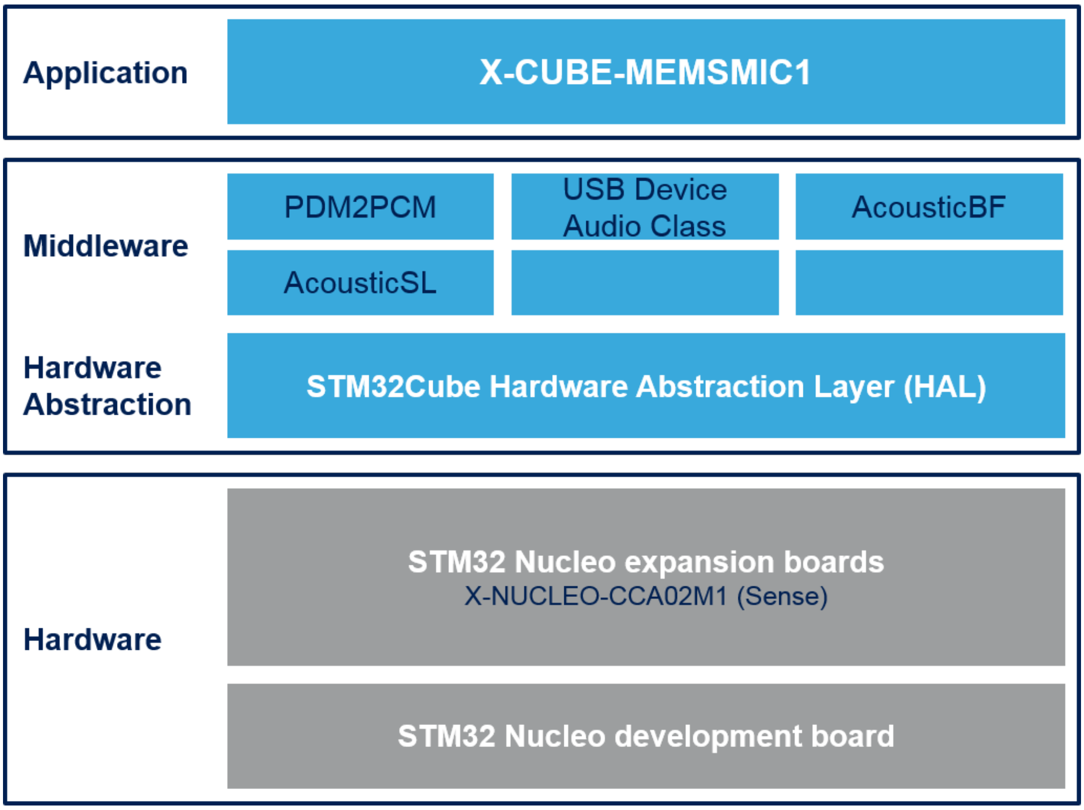


Figure 1. X-CUBE-MEMSMIC1 software architecture

2.3 Folder structure

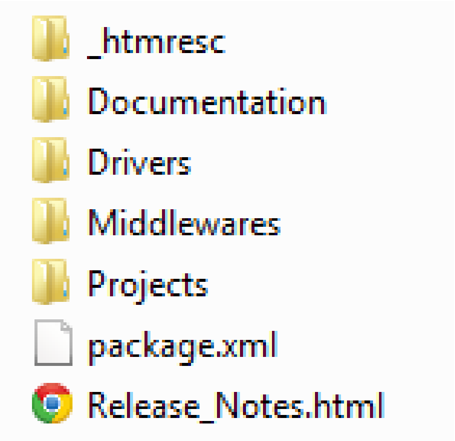

_htmresc
Documentation
Drivers
Middlewares
Projects
package.xml
Release_Notes.html

Figure 2. X-CUBE-MEMSMIC1 package folder structure

The following folders are included in the software package:

- **Documentation**: contains a compiled HTML file generated from the source code and detailed documentation of the software components and APIs
- **Drivers**: contains the HAL drivers and the board-specific drivers for supported board and hardware platforms, including those for the on-board components and the CMSIS vendor-independent hardware abstraction layer for the ARM® Cortex®-M processor series
- **Middlewares**: contains libraries and protocols for the PDM to PCM conversion, beamforming (AcousticBF), sound source localization (AcousticSL) and the audio-input USB driver

- **Projects**: contains a sample application for a NUCLEO-F401RE, a NUCLEO-L476RG or a NUCLEO-F746ZG platform to access microphone data, with three development environments, IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM-STM32) and System Workbench for STM32 (SW4STM32)

2.4 APIs

Detailed descriptions of all the functions and parameters of the user APIs user can be found in a compiled HTML file located in the Documentation folder.

2.5 Sample application description

The package includes several microphone-based applications covering a range of use cases, from the simple microphone acquisition to the advanced audio processing. The following applications are provided in the **Projects** directory with ready-to-build projects for multiple IDEs:

- microphone acquisition and streaming via USB, supported by the X-NUCLEO-CCA02M1 expansion board with a NUCLEO-F401RE, NUCLEO-L476RG or NUCLEO-F746ZG board.
- AcousticBF sample: real-time adaptive beamforming application using the PDM signals acquired via two digital MEMS microphones and the AcousticBF middleware to create a virtual directional microphone pointing to a fixed direction. It is supported by an X-NUCLEO-CCA02M1 expansion board with a NUCLEO-F401RE board.
- AcousticSL sample: real-time sound source localization application using four signals acquired via digital MEMS microphones and the AcousticSL middleware to estimate the arrival direction of an audio source. It is supported by the X-NUCLEO-CCA02M1 expansion board with a NUCLEO-F401RE board.

2.5.1 Microphone streaming via USB

This section describes the microphone streaming application via USB. Section 2.5.1.1 Audio acquisition strategy provides an overview of the MEMS microphone acquisition strategies and principles adopted. It can simplify the comprehension of the firmware structure and utilization.

2.5.1.1 *Audio acquisition strategy*

A digital MEMS microphone can be acquired via different peripherals, such as SPI, I²S, GPIO or DFSDM. It requires an input clock and it outputs a PDM stream at the same frequency of the input clock. This PDM stream is further filtered and decimated for conversion into PCM standard for audio transmission.

Two different digital MEMS microphones can be connected on the same data line, configuring the first to generate valid data on the rising edge of the clock and the other on the falling edge, by setting the L/R pin of each microphone accordingly. On the X-NUCLEO-CCA02M1 expansion board, two microphones share the same data line and can be managed differently, depending on the MCU capabilities.

If a digital filter for sigma delta modulators (DFSDM) peripheral is available, the microphones can be routed to DFSDM channels, otherwise use the I²S peripheral for the first and the second microphone and the SPI peripheral for the third and the fourth.

2.5.1.1.1 DFSDM microphone acquisition

The DFSDM peripheral generates the precise clock needed by the microphones and reads the data on the rising and falling edges of each PDM line. Acquired signals are then input to DSFDM filters for hardware filtering and decimation to generate standard PCM streams.

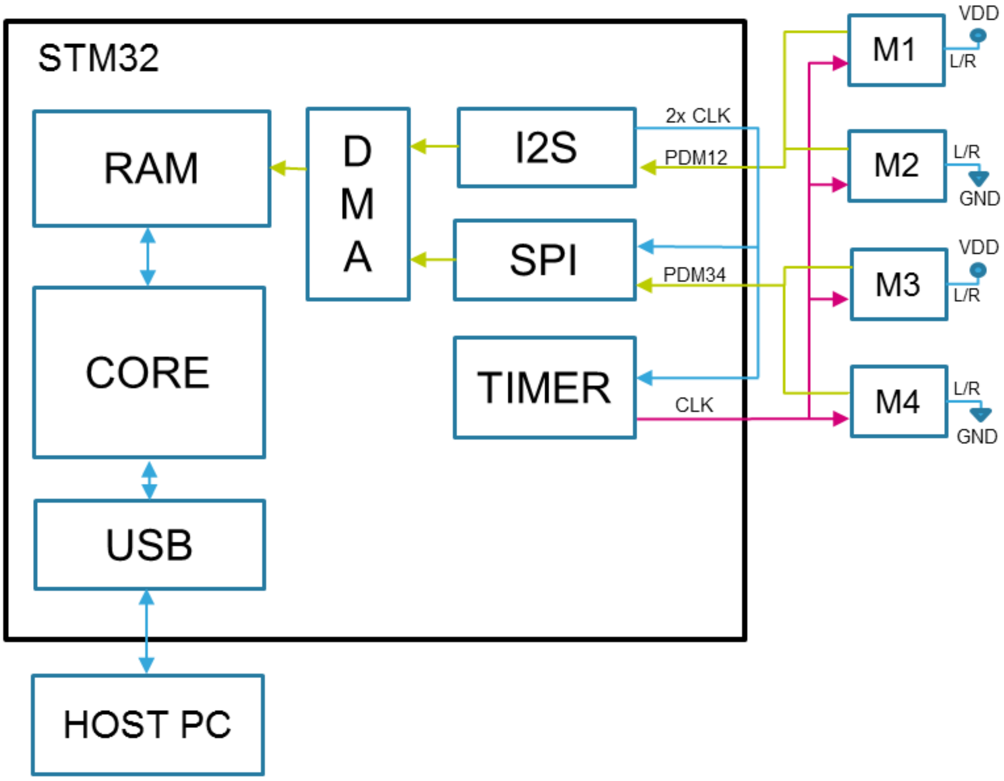
An additional software high pass filtering stage removes any DC offset in the output stream. DMA is used to reduce MCU load.

2.5.1.1.2 I²S and SPI microphone acquisition

In this scenario, a precise clock signal is generated by the I²S peripheral while the SPI is configured in slave mode and is fed by the same timing signal generated by I²S. This clock is then halved by a timer and input to the microphones. The SPI and I²S peripherals operate at twice the microphone frequency to read the data on both the rising and falling edges of the microphone clock, thus reading the bits of two microphones each.

A software demuxing step separates the signal from the two microphones and allows further processing like PDM-to-PCM conversion. For further information regarding MEMS microphone and PDM-to-PCM decimation, please refer to application note AN3998 on www.st.com.

Figure 3. **General acquisition strategy using I²S and SPI**



For single microphone acquisition, the correct microphone timer is generated directly by I²S and one single microphone data line is read by the same peripheral.

2.5.2 Application description

The application example acquires the PDM outputs from the MEMS microphones, converts them into PCM format and streams the resulting signal to a host PC via USB. PDM to PCM conversion is performed using a software library when the system is based on I²S plus SPI; in other cases, DFSDM dedicated hardware is adopted. Synchronized acquisition of 1, 2 or 4 microphones is possible by configuring the solder bridges appropriately.

PDM and PCM data is available in the application before being sent to the USB to facilitate the development and testing of microphone-based audio processing algorithms.

STM32 I²S, SPI or DFSDM devices used for microphones acquisition are set up depending on the sampling frequency and number of channels. If available, the DFSDM peripheral is used to acquire up to four channels; otherwise, PDM signals are acquired through I²S (up to 2 microphones) or both I²S and SPI peripherals (up to 4 microphones). Moreover, the USB driver is initialized and the relative descriptor is configured to match the correct channels and sample frequency configuration.

After the configuration phase, acquisition commences and the device streams the signals to a host device as a standard multichannel USB microphone.

Note: The positions of solder bridges and jumpers must be compliant with the desired microphone configuration; please refer to Section 3.1.2 X-NUCLEO-CCA02M1 expansion board for more hardware setup details.

The audio-related firmware components are mainly concentrated in the `audio_application.c` and `usb_audio_if.c` files, which employ the dedicated X-NUCLEO-CCA02M1 BSP layer and, when necessary, the PDM-to-PCM decimation library middleware. In this version, STM32 peripheral setup via the BSP layer is driven by the USB driver to allow total device control from the host PC. A different approach could be used to decouple USB and BSP functions by removing BSP functions from the USB interface file (`usb_audio_if.c`) and calling them in the application itself.

To set up the system for microphone acquisition and streaming (you can trace them in the application):

- check the HW configuration for solder bridges and jumpers (see Section 3.1.2 X-NUCLEO-CCA02M1 expansion board)
- initialize USB descriptor using `USBD_AUDIO_Init_Microphone_Descriptor(…)` according to the number of channels to be streamed
- initialize USB core and start USB functionalities with the functions: `USBD_Init(…)`, `USBD_RegisterClass(…)`, `USBD_AUDIO_RegisterInterface(…)`, `USBD_Start(…)`. This allows the device to be recognized as a standard USB microphone with the requested configuration
- All the required peripherals and middleware must be configured depending on the number of channels to be streamed and the desired sampling frequency to be achieved; these steps are performed inside the `usbd_audio_if.c` file, whose functions are called in response to the USB operations that start when the device is connected to the PC (enumeration, dataInput…). The following BSP functions are used:
  - BSP Initialization: initializes the hardware peripherals and the PDM library depending on the desired acquisition settings
  - BSP Record: starts acquisition; a double buffer mechanism is implemented and an interrupt is generated every millisecond to allow data processing
  - BSP Stop: stops data acquisition
- The interrupt service routine for audio data acquisition is implemented as a callback in the `audio_application.c` file (`BSP_AUDIO_IN_Transfer_CallBack(…)`). When the system is based on I²S and SPI peripherals NUCLEO-F401RE and NUCLEO-F746ZG board, the software PDM-to-PCM conversion is performed in this function. To achieve this, the BSP layer provides a millisecond of PDM data, demuxed and arranged as required by the decimation library. Conversely, when the system employs DFSDM (NUCLEO-L476RG), PDM-to-PCM conversion is handled by hardware and the BSP layer provides the application PCM data, arranged as a standard multichannel PCM stream. The user can modify this function to, for example, add DSP functions to the audio before sending it to USB via the data transfer function. For further information about USB APIs, refer the compiled HTML file located in the ‘Documentation’ folder.

Alternatively, configuration and initialization of the audio peripherals can be managed independently of the USB flow by directly calling the audio-related BSP functions from the application space. In this case, the user calls the BSP Initialization, Record and Stop functions as needed.

Note: *Depending on the MCU used and the corresponding available resources, not all channels/sampling frequency combinations are allowed.*

The configurations supported by each Nucleo board are:

- NUCLEO-L476RG: 1, 2 or 4 microphone acquisition and streaming at 8 kHz, 16 kHz, 32 kHz, 48 kHz or 96 kHz
- NUCLEO-F401RE or NUCLEO-F746ZG: 1, 2 or 4 microphone acquisition and streaming at 8 kHz, 16 kHz, 32 kHz or 48 kHz

2.6 AcosuticBF example

The application is designed to acquire the two microphones soldered on the X-NUCLEO-CCA02M1 board, perform beamforming and stream two audio channels to a host PC via USB.

The audio streams contain:

- the algorithm output (first channel)
- an omnidirectional microphone as a reference (second channel)

Note: *The positions of solder bridges and jumpers must be compliant with the 2 microphones configuration when using I²S (refer to Section 3.1.2 X-NUCLEO-CCA02M1 expansion board for more hardware setup details).*

The application performs this sequence of operations:

- initializes and starts the USB audio input driver and middleware: this allows a host PC to recognize the device as a standard multichannel USB microphone
- initializes microphone acquisition using the relevant BSP function

- initializes the AcousticBF library
- starts the audio acquisition to trigger the library execution
- transmits processed data and the omnidirectional microphone reference to the USB driver every millisecond

The audio-related firmware components are mainly located in the `audio_application.c` and `usb_audio_if.c` files, which employ the dedicated X-NUCLEO-CCA02M1 BSP layer and the AcousticBF middleware.

USB and microphones initialization are the same as shown in Section 2.5.2 Application description.

Further details about the library API can be found in the chm help file in the Documentation folder.

Note: This application is supported by the X-NUCLEO-CCA02M1 expansion board when connected to a NUCLEO-F401RE.

2.7 AcosuticSL example

The application is designed to perform:

- acquisition of the four external microphones mounted on the X-NUCLEO-CCA02M1 board
- sound source localization running based on those signals
- output of the localization result through the ST-LINK embedded virtual COM port
- streaming of four audio channels to a host PC via the X-NUCLEO-CCA02M1 USB connector

Note: The positions of solder bridges and jumpers must be compliant with the 4 microphones configuration when using I²S and SPI (please refer to Section 3.1.2 X-NUCLEO-CCA02M1 expansion board for more hardware setup details).

The application performs this sequence of operations:

- initializes and starts the USB audio input driver and middleware: this allows a host PC to recognize the device as a standard multichannel USB microphone
- initializes microphone acquisition using the relevant BSP function
- initializes the AcousticSL library
- starts the audio acquisition to trigger the library execution
- transmits the four omnidirectional microphone streams to the USB driver every millisecond
- sends the AcousticSL results to a host PC on each library running cycle through ST-LINK VCP.

The audio-related firmware components are mainly concentrated in the `audio_application.c` and `usb_audio_if.c` files, which employ the dedicated X-NUCLEO-CCA02M1 BSP layer and the AcousticSL middleware.

USB and microphones initialization are the same as shown in Section 2.5.2 Application description.

Further details about the library API can be found in the chm help file in the Documentation folder.

Note: This application is supported by the X-NUCLEO-CCA02M1 expansion board when connected to a NUCLEO-F401RE.

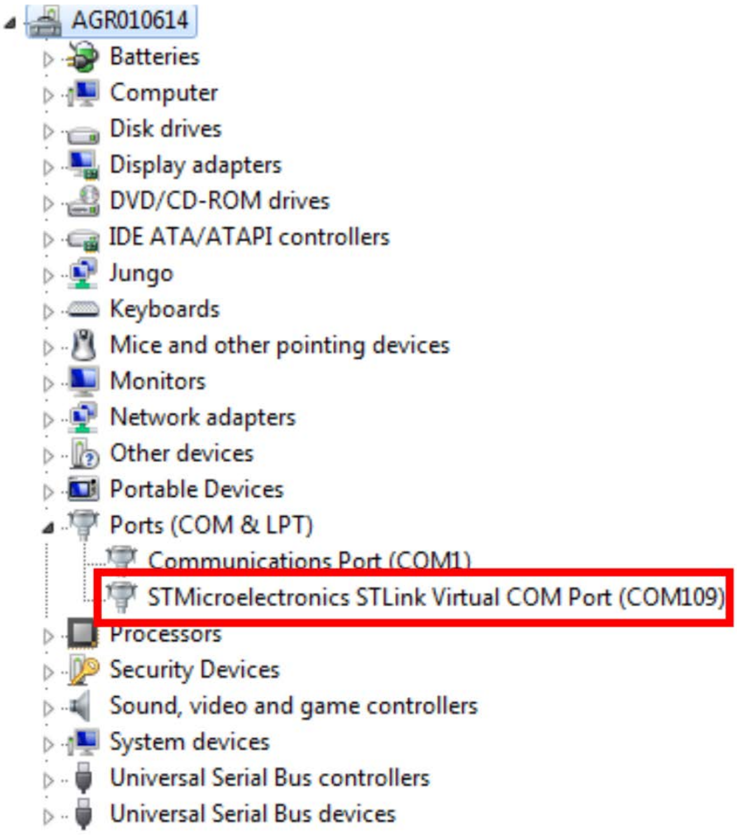
2.7.1 Program execution example

A third party serial communication software is required to be able to collect localization values. One possible solution is the free Putty software available at: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html.

To set up the demo:

Step 1. Connect the NUCLEO-F401RE mini USB cable (USB type A to mini-B USB cable)

Figure 4. **STLink Virtual COM port detection**



AGR010614
- Batteries
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- IDE ATA/ATAPI controllers
- Jungo
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Other devices
- Portable Devices
- Ports (COM & LPT)
  - Communications Port (COM1)
  - STMicroelectronics STLink Virtual COM Port (COM109)
- Processors
- Security Devices
- Sound, video and game controllers
- System devices
- Universal Serial Bus controllers
- Universal Serial Bus devices

Step 2. Check the COM Port number assigned to the board in Windows Device Manager

Step 3. Open the serial utility and ensure the connection parameters are set to:

– Baudrate: 9600

– Data bits: 8

– Stop bits: 1

– Parity: none

– Flow control: none

Step 4. Open the COM port

The localization results are shown.

Note: Below a certain audio energy threshold, the library is not executed; in this case, localization is not computed and nothing is sent to the VCP.

To record audio through the device, you need to install a third party software such as Audacity® freeware program to save or play the streamed signal.

Further information about Audacity® can be found at: http://audacityteam.org/?lang=en.

2.8 PC audio recording utility example: Audacity

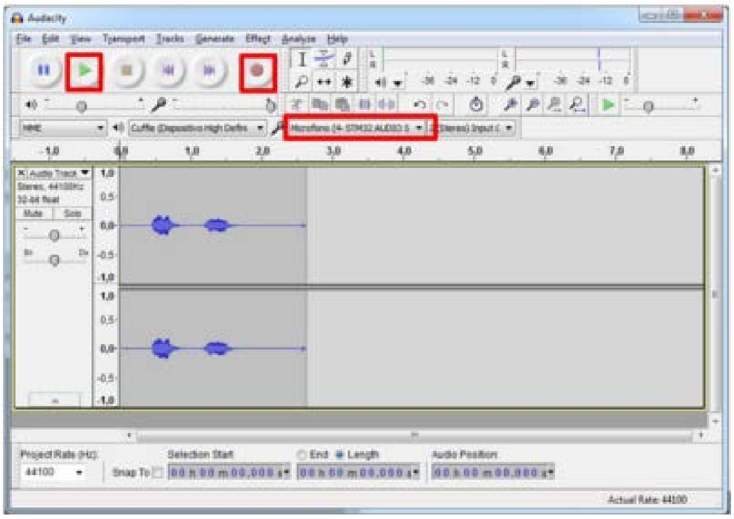
This section describes the use of the Audacity® application for multi-channel audio recording.

Audacity is a free, open source, cross-platform software package for recording and editing sounds and is available for Windows®, iOS®, GNU/Linux® other operating systems as a freeware audio editing environment (http://sourceforge.net/projects/audacity/).

In Windows 7, the released Audacity version is capable of recording up to 2 microphones; the proprietary ASIO driver is usually the best way to perform multi-channel recordings on Windows, but licensing restrictions prevent including ASIO support in released versions of Audacity. However, it can be compiled with ASIO support for private, non-distributable use.(For more information, refer to Audacity ASIO Audio interface wiki.)

To start audio recording, ensure that STM32 AUDIO Streaming in FS mode is the selected audio input device and proceed to record and play audio using from the interface shown below.

Figure 5. Audacity for Windows



MME
Cuffie (Dispositivo High Defin
Microfono (4- STM32 AUDIO S
2 (Stereo) Input C

Project Rate (Hz): 44100
Snap To □
Selection Start 00 h 00 m 00,000 s
○ End ● Length 00 h 00 m 00,000 s
Audio Position 00 h 00 m 00,000 s
Actual Rate: 44100

3 System setup guide

3.1 Hardware description

This section describes the hardware components needed for developing applications based on digital MEMS microphones. The following sub-sections describe the individual components.

3.1.1 STM32 Nucleo platform

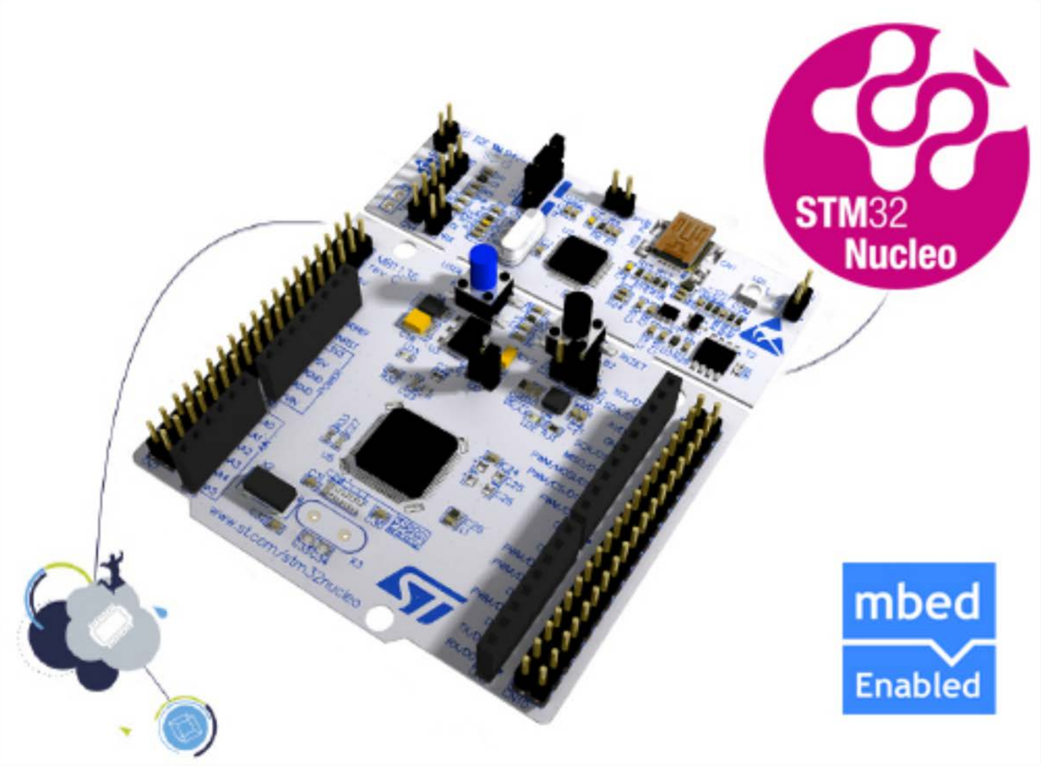
STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

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 6. STM32 Nucleo board

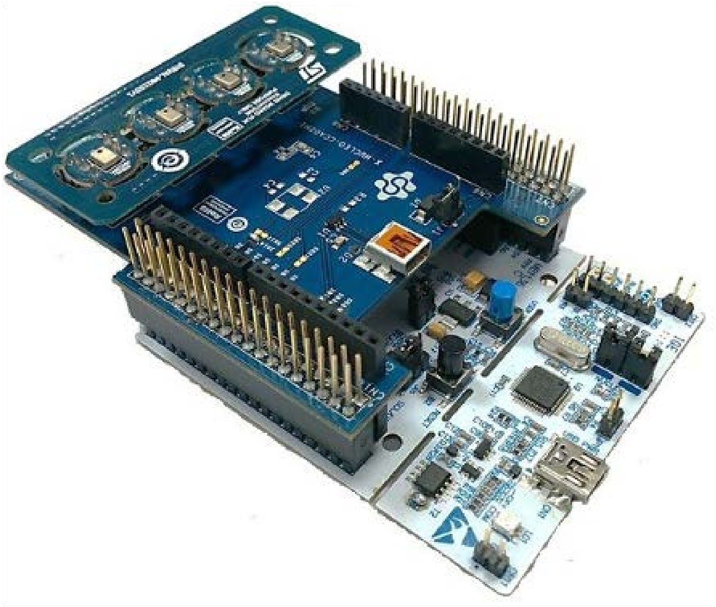


Information regarding the STM32 Nucleo board is available at www.st.com/stm32nucleo

3.1.2 X-NUCLEO-CCA02M1 expansion board

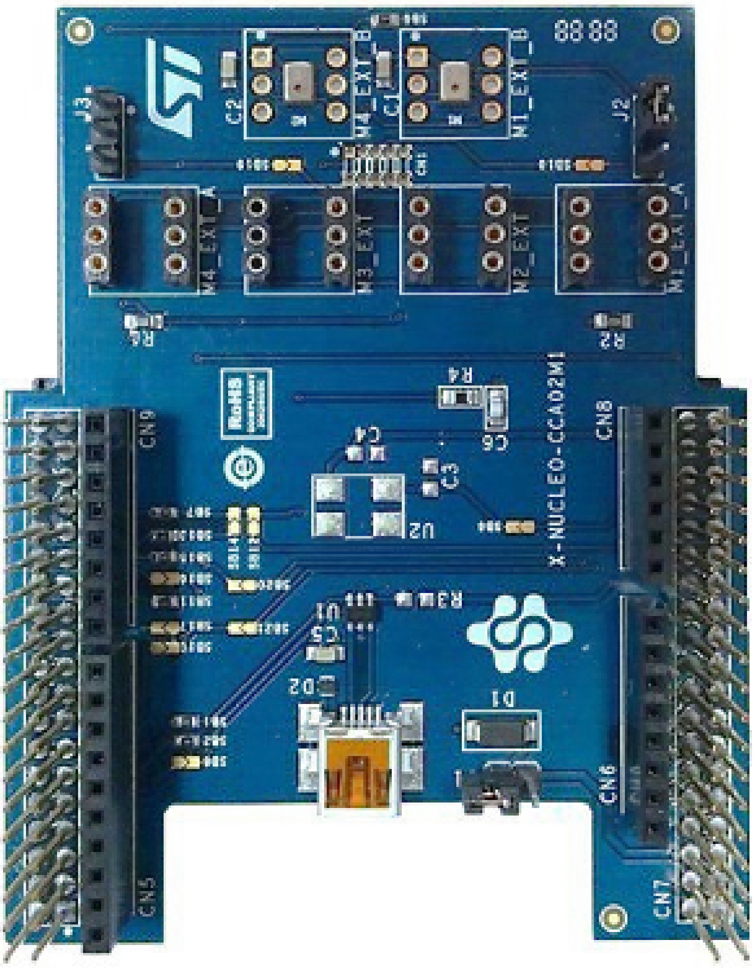
X-NUCLEO-CCA02M1 is an evaluation board based on digital MEMS microphones. It is compatible with the ST morpho connector layout and is designed around STMicroelectronics MP34DT01-M digital microphones. It has two microphones soldered on the board and it offers the possibility of connecting additional microphones using MP32DT01-based coupon evaluation boards (STEVAL-MKI129Vx or STEVAL-MKI155Vx).

Figure 7. **STEVAL-MKI155V1 plugged on X-NUCLEO-CCA02M1**



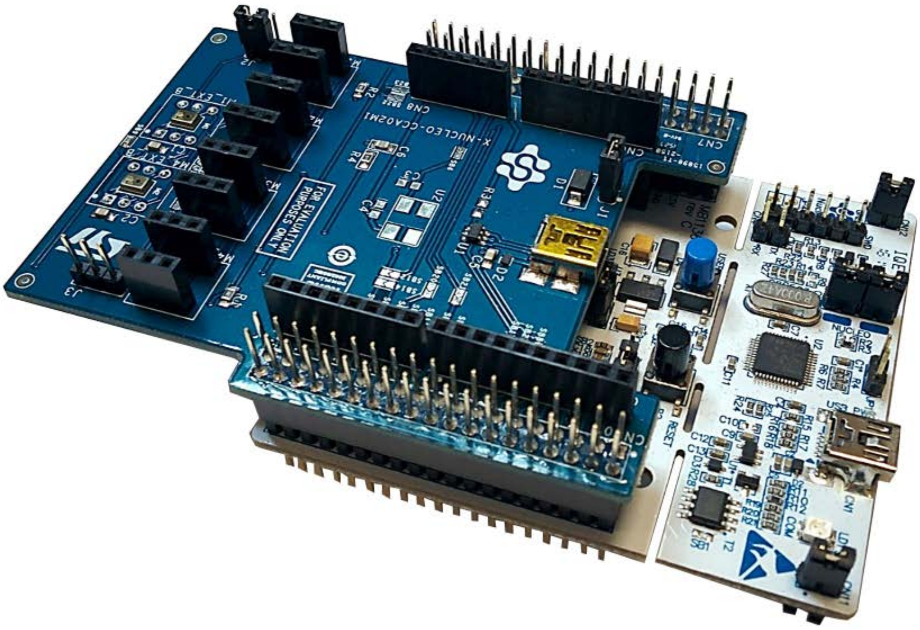
X-NUCLEO-CCA02M1 allows the acquisition and streaming of up to two microphones using the I²S bus and up four coupon microphones using I²S and SPI together. It represents an easy and fast solution for the development of microphone-based applications, as well as a starting point for audio algorithm implementation. When available, the DFSDM peripheral is employed to handle up to 4 microphones.

Figure 8. X-NUCLEO-CCA02M1 expansion board



Information about the X-NUCLEO- CCA02M1 expansion board is available on www.st.com at http://www.st.com/x-nucleo.

Figure 9. **Microphone expansion board plugged to STM32 Nucleo board**



3.2 Software description

The following software components are needed in order to set up the development environment for creating applications for the STM32 Nucleo equipped with a MEMS microphone expansion board:

- X-CUBE-MEMSMIC1: an STM32Cube expansion for audio application development; the X-CUBE-MEMSMIC1 firmware and related documentation is available on st.com
- development tool-chain and compiler: the STM32Cube expansion software supports the following environments:
  - IAR Embedded Workbench for ARM® (IAR-EWARM) toolchain + ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM-STM32) toolchain + ST-LINK
  - System Workbench for STM32 (SW4STM32) + ST-LINK

3.3 Hardware and software setup

This section describes the hardware and software setup procedures, and the corresponding system setup.

3.3.1 Hardware setup

The following hardware components are needed:

1. an STM32 Nucleo development platform (suggested order code: NUCLEO-F401RE)
2. a microphone expansion board (order code: X-NUCLEO-CCA02M1)
3. a USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC
4. a USB type A to Mini-B USB cable to connect the X-NUCLEO-CCA02M1 to the PC for USB streaming

3.3.2 External microphone connection

The X-NUCLEO-CCA02M1 expansion board is designed to support up to four external digital microphones based on the ST coupon daughterboard concept (part number: STEVAL-MKI129Vx or STEVAL-MKI155Vx, refer to 4. for further details).

For this purpose, four headers (`M1_EXT_A`, `M2_EXT`, `M3_EXT`, `M4_EXT_A`) allow operation along a linear array concept. Footprints for additional headers (`M1_EXT_B` and `M4_EXT_B`) can be used as an alternative to `M1_EXT_A` and `M4_EXT_A` to create a square-shaped microphone array.

Note: M1_EXT_A, M1_EXT_B and M4_EXT_A, M4_EXT_B share the same signals, so be careful not to plug both M4_EXT_A and M4_EXT_B or M1_EXT_A and M1_EXT_B at the same time to avoid possible microphone damage.

3.3.3 Jumper configuration

The X-NUCLEO-CCA02M1 expansion board offers various solutions for microphone acquisition and USB streaming. Depending on your needs, you can choose to acquire on-board microphones or external microphones from one to four channels.

Each configuration requires the correct solder-bridge setup. Different scenarios with corresponding hardware configurations are described below.

Different acquisition scenarios can be achieved depending on the number of microphones and configuration. JP2 and JP3 are used to choose between external and internal microphones, while solder jumpers are used to connect the right signals to the MCU devices (SPI, Timer, I²S and DFSDM).

The configurations depend on the acquisition peripherals adopted: DFSDM or I²S plus SPI.

Note: For each hardware configuration, the correct firmware initialization parameters must be used.

3.3.3.1 *Jumper settings for DFSDM-based systems*

1 or 2 microphone acquisition

The clock is generated by DFSDM peripheral and the PDM line of the first and second microphone is routed to the MCU.

Table 2. **Solder bridge configuration for 1 or 2 microphone acquisition**

SB	Status
SB7	Open
SB8	Open
SB9	Open/Close
SB10	Open
SB11	Open
SB12	Close
SB13	Open
SB14	Open
SB15	Open
SB16	Close
SB17	Open
SB18	Open
SB19	Open
SB20	Open
SB21	Open

In addition, J2 must be placed in position 1-2 for on-board microphone acquisition or 2-3 when using an external microphone while J3 must be left open. When acquiring on-board microphones, close SB9 to acquire both of them.

4 microphone acquisition

In this case, the PDM line of the third and fourth microphone is also routed to the MCU.

Table 3. **Solder bridge configuration for 4 microphone acquisition**

SB	Status
SB7	Open
SB8	Open
SB9	Open
SB10	Open
SB11	Open
SB12	Close
SB13	Open
SB14	Open
SB15	Open
SB16	Close
SB17	Close
SB18	Open
SB19	Open
SB20	Open
SB21	Open

In addition, J2 and J3 must be placed in position 2-3 for external microphone acquisition.

3.3.3.2 *Jumper settings for I²S-plus-SPI-based systems*

1 microphone acquisition

The I²S peripheral is used directly to give the right clock to the microphone and thus acquires the same microphone. For this use case, you need the following SB configuration.

Table 4. **Solder bridge configuration for 1 microphone acquisition**

SB	Status
SB7	Open
SB8	Open
SB9	Open
SB10	Open
SB11	Close
SB12	Open
SB13	Close
SB14	Close
SB15	Open
SB16	Open
SB17	Open

SB	Status
SB18	Open
SB19	Open
SB20	Open
SB21	Open

In addition, J2 must be placed in position 1-2 for on-board microphone acquisition or 2-3 when using an external microphone, while J3 must be left open. If using external microphones, do not plug anything in `M2_EXT` header.

2 microphone acquisition

In this case, the I²S peripheral is used to generate twice the frequency needed by the microphones. In this scenario, the clock is then halved by the timer and routed to the microphones to give them the right clock. I²S therefore reads values from both edges of the merged PDM lines. For this use case you need the following SB configuration:

Table 5. **Solder bridge configuration for 2 microphone acquisition**

SB	Status
SB7	Close
SB8	Open
SB9	Open /Close
SB10	Open
SB11	Close
SB12	Open
SB13	Close
SB14	Open
SB15	Close
SB16	Open
SB17	Open
SB18	Open
SB19	Open
SB20	Open
SB21	Open

In addition, J2 must be placed in position 1-2 for on-board microphone acquisition or 2-3 when using external microphones, while J3 must be left open. When acquiring on-board microphones, close SB9 to acquire both of them.

4 external microphone acquisition

In this case, the I²S peripheral is used to generate a clock frequency that is twice the frequency needed by the microphones, and SPI is configured in slave mode in order to use such timing. As in the previous case, the clock is then halved by the timer and routed to the microphones to give the right clock. I²S and SPI read values from both the edges of the merged PDM lines. For this use case you need the following SB configuration:

Table 6. **Solder bridge configuration for 4 microphone acquisition**

SB	Status
SB7	Close
SB8	Close
SB9	Open
SB10	Close
SB11	Close
SB12	Open
SB13	Close
SB14	Open
SB15	Close
SB16	Open
SB17	Open
SB18	Open
SB19	Open
SB20	Open
SB21	Open

In addition, J2 and J3 must be placed in position 2-3 for external microphone acquisition.

Note: Other configurations are available, based on the MCU used.

Note: When acquiring 4 microphones using an STM32 F746Nucleo-144 development board, JP6 jumper on the board must be opened.

3.3.3.3 *Board power supply jumper*

X-NUCLEO-CCA02M1 expansion board has an on-board USB connector to power the underlying STM32 Nucleo board. For this purpose, close JP1 on the X-NUCLEO-CCA02M1 expansion board and configure jumper JP5 on the STM32 Nucleo board on the E5V side.

When a NUCLEO-F746ZG is used, a micro-USB OTG connector is available on the Nucleo board itself.

Note: Do not connect the X-NUCLEO connector and the Nucleo connector to the PC simultaneously.

3.3.3.4 *USB connection solder bridges*

X-NUCLEO-CCA02M1 USB connector is connected to DM and DP pins of the STM32 Nucleo board through two solder jumpers: SB1 and SB2. These solder bridges must be closed if you want to use USB communication, otherwise you can leave them open.

When a NUCLEO-F746ZG is used, SB1 and SB2 can be left open and the micro USB OTG connector on the Nucleo board can be used to connect to the PC.

3.3.4 Software setup

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

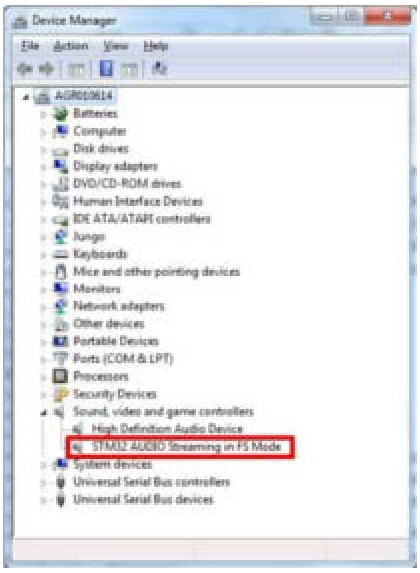
3.3.4.1 *Development tool-chains and compilers*

Select one of the integrated development environments supported by the STM32Cube expansion software and read the system requirements and setup information for the selected IDE provider.

3.3.4.2 *Recognition of the device as a standard USB microphone in Windows 7*

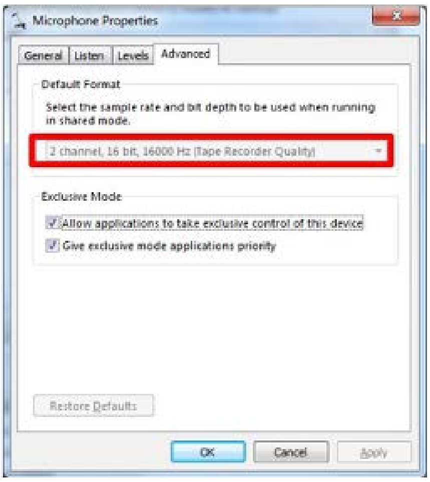
The sample application for audio acquisition and streaming includes an audio input USB driver that allows the device to be recognized as a standard USB microphone. After firmware download to MCU flash, jumper setup and X-NUCLEO-CCA02M1 connection to PC via USB, check the device manager to ensure it has been recognized correctly, as shown in the figure below.

Figure 10. **STM32 microphone in device manager**



The final step is to right-click on the volume icon in the Windows task bar and choose ‘Recording device’. Now select STM32 microphone and click on ‘Properties’. In the ‘Advanced’ tab, there is a summary of the current device setup in terms of sampling frequency and number of channels. Select the right configuration in order to be able to record and save audio.

Figure 11. **Advanced configuration setup**



Select the sample rate and bit depth to be used when running in shared mode.

2 channel, 16 bit, 16000 Hz (Tape Recorder Quality)

☑ Allow applications to take exclusive control of this device

☑ Give exclusive mode applications priority

3.3.5 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 MEMS microphone expansion board.

3.3.5.1 *STM32 Nucleo and microphone 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 by searching STSW-LINK008 or STSW-LINK009 on www.st.com.

The microphones expansion board X-NUCLEO-CCA02M1 can be easily connected to the STM32 Nucleo through the ST morpho extension connector on the board. The microphone expansion board is capable of interfacing with the external STM32 microcontroller on STM32 Nucleo via I²C, I²S and USB.

4 References

1. MP34DT01-M datasheet DS10384 - *MEMS audio sensor omnidirectional digital microphone* on www.st.com.
2. Application note AN3998 - *PDM audio software decoding on STM32 microcontrollers* on www.st.com.
3. Audacity ASIO Audio interface wiki.
4. Data briefs for STEVAL-MKI129Vx series and data briefs for STEVAL-MKI155Vx on www.st.com

Revision history

Table 7. **Document revision history**

Date	Version	Changes
09-Jun-2015	1	Initial release.
27-Jan-2016	2	Throughout document added reference to NUCLEO-L476RG board and DFSDM peripheral option Updated Section 2.5: "Sample application description" Updated Section 3.3.3: "Jumper configuration"
11-Jul-2016	3	Updated Nucleo board compatibility information Updated Section 4.3.3.3: "Board power supply jumper" Updated Section 4.3.3.4: "USB connection solder bridges"
11-May-2017	4	Updated Section Introduction, Section 2.1 Overview , Section 2.2 Architecture, Section 2.3 Folder structure, Section 2.5 Sample application description, Section 2.5.1.1 Audio acquisition strategy and Section 2.5.2 Application description. Added Section 2.5.1 Microphone streaming via USB, Section 2.6 AcosuticBF example, Section 2.7 AcosuticSL example and Section 2.7.1 Program execution example.
19-Apr-2018	5	Updated Figure 1. X-CUBE-MEMSMIC1 software architecture. Removed references to NUCLEO-F072RB and NUCLEO-L053R8 boards.

Contents

List of tables

List of figures

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.

© 2018 STMicroelectronics – All rights reserved