

STM32Cube function pack for full-duplex voice streaming over Bluetooth low energy using Opus compression

Application	FP-AUD-BVLINK2		
Middleware	BLE	BlueVoice OPUS	PDM Lib
	USB Device	OPUS	
Hardware Abstraction	STM32Cube Hardware Abstraction Layer (HAL)		
Hardware	STM32 Nucleo Expansion boards X-NUCLEO-IDB05A1 (Connect) X-NUCLEO-CCA02M1 (Sense)		STEVAL-STLKT01V1 or STEVAL-BCNKT01V1 evaluation board
	STM32 Nucleo development board		



Features

- Complete firmware to implement full-duplex speech communication over Bluetooth low energy (BLE) using Opus compression
- A BlueVoiceOPUS customized profile for audio over BLE, including an easy-to-use set of APIs to exploit advanced Opus functionality (source code available)
- Third-party Opus v1.2.1 (downloadable from <https://www.opus-codec.org>) middleware: an open, royalty-free and highly versatile audio codec that is standardized by the Internet Engineering Task Force (IETF) as [RFC 6716](#)
- Digital audio signal acquisition and processing
- Audio out playback through USB or jack connector
- Sample implementation available for [X-NUCLEO-IDB05A1](#) plus [X-NUCLEO-CCA02M1](#) connected to a [NUCLEO-F446RE](#) or [NUCLEO-L476RG](#), for [SensorTile \(STEVAL-STLKT01V1\)](#) and [BlueCoin \(STEVAL-BCNKT01V1\)](#)
- Compatibility with [STBLESensor](#) app (v 4.2.0 or higher) for Android, to perform audio streaming at 16 kHz
- Easy portability across different MCU families thanks to [STM32Cube](#)
- Free, user-friendly license terms

Description

FP-AUD-BVLINK2 is an [STM32Cube](#) function pack that performs voice streaming over BLE in a full-duplex configuration using the advanced Opus compression algorithm. The application runs on the [STM32 Nucleo](#) and includes drivers and middleware for BLE ([BlueNRG-MS](#)) and digital MEMS microphones.

The Peripheral module can also stream audio to an Android™ device running the [STBLESensor](#) app v 4.2.0 or higher

The software with the suggested combination of STM32 and ST devices can be used, for example, to develop wireless audio communication systems in smart home or wearable applications.

Thanks to the low bitrates you can achieve while maintaining high voice quality with Opus, combined with the low power features of [BlueNRG](#), you can develop applications featuring very low consumption.

The same audio codec can be used to implement high quality stereo music streaming.

The software runs on the STM32 microcontroller and includes all the necessary drivers to use the devices on the [STM32 Nucleo](#) development board and expansion boards, as well as on the [STEVAL-BCNKT01V1](#) and [STEVAL-STLKT01V1](#) evaluation boards.

It also includes the Opus audio codec as the [STM32Cube](#) middleware.

Product summary	
STM32Cube function pack for full-duplex voice streaming over Bluetooth Low Energy via Opus compression	FP-AUD-BVLINK2
Bluetooth low energy expansion board based on SPBTLE-RF module for STM32 Nucleo	X-NUCLEO-IDB05A1
Digital MEMS microphone expansion board based on MP34DT01-M for STM32 Nucleo	X-NUCLEO-CCA02M1
SensorTile development kit	STEVAL-STLKT01V1
BlueCoin starter kit	STEVAL-BCNKT01V1

1 Detailed description

1.1 What can you do with STM32Cube function packs?

The [STM32Cube](#) function packs leverage the modularity and interoperability of STM32 Nucleo and X-NUCLEO boards, and STM32Cube and X-CUBE software, to create function examples, embodying some of the most common use cases, for each application area.

These software function packs are designed to exploit as much as possible the underlying [STM32 ODE](#) hardware and software components to best fit the requirements of final users' applications.

Moreover, function packs may include additional libraries and frameworks which do not present the original X-CUBE packages, thus enabling new functionalities and creating a real and usable system for developers.

1.2 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

1.2.1 How does this STM32Cube function pack 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 [BlueNRG-MS](#), MEMS microphone expansion boards, SensorTile and BlueCoin; middleware components for audio acquisition, communication with other BLE devices, USB streaming of recorded signals and a dedicated profile for full-duplex speech transmission over BLE ([BlueVoiceOPUS](#)).

The third party Opus (v1.2.1) middleware is included in the function pack.

The [BlueVoiceOPUS](#) profile defines a BLE service which includes one characteristic for audio transmission and one for optional control message. In a full-duplex system, both sides of the communication (central and peripheral) can act as a server of information. Periodic notifications containing compressed audio data are sent from the central node acting as a server to the peripheral node acting as a client, and vice versa.

The [BlueVoiceOPUS](#) middleware is responsible for audio encoding and periodic data transmission on the server side and for decoding of received data on the client side.

The drivers abstract low-level hardware details and allow the middleware components and applications to access the devices in a hardware-independent fashion.

The package includes a sample application that developers can use to start experimenting with the code. It enables audio acquisition, compression and transmission over BLE from the module acting as a transmitter to the module acting as a receiver. The receiver is responsible for audio decompression and USB streaming of audio data to a PC. The system is recognized by the PC as a standard microphone, and any freeware or commercial audio recording software can be used to interface with it. Depending on the hardware used, the audio can be played through a jack connector. Both the central and the peripheral modules can act as a transmitter and a receiver at the same time, enabling full-duplex voice streaming.

The Peripheral module can also stream audio @16kHz to an Android device running the [STBLESensor](#) app (v 4.2.0 or higher).

Revision history

Table 1. Document revision history

Date	Version	Changes
06-Mar-2017	1	Initial release.
11-Jan-2018	2	Added compatibility information for STBLESensor app.

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.

© 2019 STMicroelectronics – All rights reserved