Infineon Voice over GATT Profile (VoGP) for Bluetooth® Low Energy

About this document

Scope and purpose

This user guide specifies the Infineon Voice over GATT Profile (VoGP) for Bluetooth® Low Energy.The specification allows for a custom implementation of a voice-enabled remote control solution as an alternative to other proprietary solutions.

Intended audience

The intended audience is developers using AIROC™ Bluetooth® devices to implement a voice-enabled remote control solution using Infineon's ModusToolbox™ software.

Table of contents

1 Infineon Voice over GATT Profile

This specification describes the custom Infineon Voice over GATT Profile (VoGP). The following sample applications demonstrating the use of this protocol between Peripheral and Central devices are available for ModusToolbox™:

- Peripheral: IFX-Voice LE Remote Control Sample Application
- Central: IFX-Voice LE Host Sample Application

Upon LE link establishment, the capabilities are exchanged between the Central and Peripheral; audio can then be streamed from the Peripheral to the Central, based on the agreed capabilities.

1.1 Audio service

The Audio service is hosted on a voice remote. It allows a client to discover capabilities and configure the way the voice remote sends audio to the client. The Audio service uses the GATT protocol to transmit the proprietary audio control and data frames.

1.1.1 Transport dependencies

Infineon Audio service must be used over Bluetooth® Low Energy transport only.

1.1.2 Declaration

The Infineon Audio service must be instantiated as a primary service. A device may host only one instance of the Infineon Audio service.

The sample apps available for ModusToolbox™ set the Audio service UUID to 16-bit UUID 0x0000. The products must not be shipped to market using this sample UUID; a custom 16-bit UUID must be obtained from the Bluetooth® SIG.

The 128-bit UUIDs are used for the IFX_AUDIO_DATA, IFX_AUDIO_CTL_TX, and IFX_AUDIO_CTL_RX service characteristics as defined in Table 1.

Table 1 Description of UUIDs

Name	UUID[1]	Descriptions	GATT properties	Attribute length	
IFX_AUDIO_SERVICE	0000	Infineon voice assistant service	User must get the UUID from the Bluetooth® SIG. A dummy UUID of “0000” is used in the sample code.		
IFX_AUDIO_DATA	7EA3D3F3-F6C3-4B40-8CF8-F1169CF2B9DC	Transmits the audio data	Read Notify	ADPCM	166
				OPUS	90
IFX_AUDIO_CTL_TX	F5481A5E-318B-4AEC-8C21-5670F1466E6A	Sends the control data from Central to Peripheral	Write Write_Without_Response	7	
IFX_AUDIO_CTL_RX	FFD1C1B9-099D-4FB1-AFB4-D90553A2E339	Sends control data from Peripheral to Central	Read Notify	7	

[1] The UUID defined is temporary UUID for testing use only. For production, the UUID must be replaced by a real UUID purchased from the Bluetooth® SIG.

The IFX_AUDIO_DATA attribute is used to stream audio data from the Peripheral to Central. Before the audio can be streamed, the Central and Peripheral use the IFX_AUDIO_CTL_TX and IFX_AUDIO_CTL_RX attributes to send messages to exchange information such as version number and capabilities. In addition, the message is also used to control the audio (for example, to start or to stop audio streaming). The Central uses the IFX_AUDIO_CTL_TX attribute to send commands to the Peripheral. On the other hand, the Peripheral uses the IFX_AUDIO_CTL_RX attribute to respond to the command or send a request to the Central.

1.2 Control messages

Each message contains one byte of message type and a variable length data.

Table 2 Control message summary

Message	Type	Data	Description	Sender
IFX_AUDIO_CAP	0x01	version [2] cap [4]	Central sends its own capability and requesting for peripheral capability	Central
IFX_AUDIO_CAP_RESP	0x02	Version [2] cap [4]	Peripheral receives host capability and responds with its own capabilities	Peripheral
IFX_AUDIO_RESP	0x03	reason [1]	Response for a command or request	Central/ Peripheral
IFX_AUDIO_START_REQ	0x04	cfg [6]	Peripheral request to start audio	Peripheral
IFX_AUDIO_START	0x05	cfg [6]	Central command to start audio	Central
IFX_AUDIO_STOP_REQ	0x06	reason [1]	Peripheral request to stop audio	Peripheral
IFX_AUDIO_STOP	0x07	reason [1]	Central command to stop audio	Central

1.3 Link up initialization (paired and connected)

Immediately upon establishing the connection, the Central sends the IFX_AUDIO_CAP command to the Peripheral to initiate the exchange of capability information. Both IFX_AUDIO_CAP and IFX_AUDIO_CAP_RESP commands contain parameters for the version number and audio capability information.

Table 3 Capability exchange messages

Name	Type, data	Description	Sender
IFX_AUDIO_CAP	0x01, version [2] cap [4]	Central sends its own capability and requests for Peripheral capability	Central
IFX_AUDIO_CAP_RESP	0x02, version [2] cap [4]	Peripheral receives the Central capability and responds with its own capabilities	Peripheral

Table 4 Capability exchange message data

version [2]	major [1] minor [1]
cap [4]	cap_codec [1] cap_data_unit [1] cap_sampling_rate [1] audio_limit [1]
major [1]	0x00[2]
minor [1]	0x02[3]
cap_codec [1]	[0b] Reserved [1b] ADPCM capable [2b] OPUS capable [3:7b] Set to '0'
cap_data_unit [1]	[0b] 8 bits [1b] 16 bits [2:7b] Set to '0'
cap_sampling_rate [1]	[0b] 8 kHz [1b] 16 kHz [2:7b] Set to '0'
resp_timeout_in_sec [1]	Respond timeout value in seconds. '0' is not a valid value. When '0' is passed, it will be replaced with the default value of '2'.

1.4 Audio streaming control

Start/stop of audio streaming is controlled by the Central; however, the Peripheral can optionally send a request to start or stop audio streaming. The message exchange sequence is shown in Figure 1.

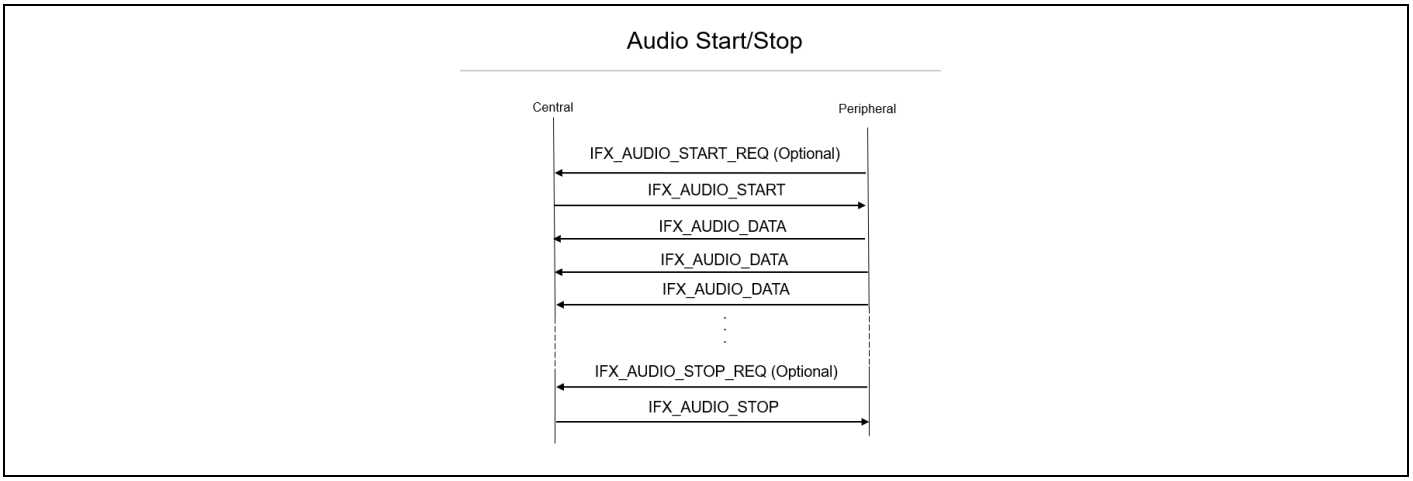


Figure 1 Message exchange sequence

[2] Indicates this specification version, 002-39100.

[3] Indicates this specification version, 002-39100.

When the Central receives the IFX_AUDIO_START_REQ request from the Peripheral, the Central should get ready to receive the audio data and send the IFX_AUDIO_START command to start audio streaming. If the Central cannot receive the audio, it should send a response with the reason to reject the request.

The Peripheral should also handle the case when the Central is not responding. The request should expire in the time specified in the capability parameter (resp_timeout_in_sec).

When the Peripheral receives the IFX_AUDIO_START command and but is unable to start audio, it should respond with IFX_AUDIO_RESP with a reason.

When the Central expects but does not receive audio data for the time specified in the parameter, it should time out and send the IFX_AUDIO_STOP command to end the audio streaming.

The maximum time audio can be streamed is specified in the configuration (see the streaming_timeout_in_sec parameter in the Audio control message data in Table 5. Once the audio streaming reaches the limit, the Central should send the IFX_AUDIO_STOP command to stop audio streaming. The Peripheral also should stop audio streaming after the specified time limit even without the IFX_AUDIO_STOP command.

Table 5 Audio control messages

Name	Type, data	Description	Sender
IFX_AUDIO_RESP	0x03, reason [1]	The response for command or request	Central/Peripheral
IFX_AUDIO_START_REQ	0x04, cfg [6]	Peripheral request to start audio	Peripheral
IFX_AUDIO_START	0x05, cfg [6]	Central command to start audio	Central
IFX_AUDIO_STOP_REQ	0x06 reason [1]	Peripheral is requesting to stop audio	Peripheral
IFX_AUDIO_STOP	0x07 reason [1]	Central command to stop audio	Central

Table 6 Audio control message data

cfg [6]	codec [1] data_unit [1] speed [1] limit [1] len [2]
codec [1]	0x00: PCM raw data[4] 0x01: Reserved 0x02: ADPCM 0x03: OPUS
data_unit [1]	0x00: 8-bit[5] 0x01: 16-bit
speed [1]	0x00: 8 kHz[6] 0x01: 16 kHz
streaming_timeout_in_sec [1]	0x00: No time limit (both Central and Peripheral must be '0'). Any other value to specify the maximum time for streaming audio. When both Central and Peripheral value does not agree, it will use the lower value of the two.
len [2]	Audio data frame length in little endian format: i.e., 166 for ADPCM, 90 for OPUS
reason [1]	0x00: User request 0x01: Not capable 0x02: Timeout 0x03: Invalid config 0x04: Capability data too long 0x05: Capability data too short 0x06: Audio not active 0x07: Busy 0x08: Audio stopped 0x09: Invalid data length

[4] Because of the bandwidth constraints, this is only good for testing purpose; not capable to stream live audio but only good to send an audio file.

[5] Not supported; firmware will reject.

[6] This option should be used if the ACL connection interval is not fast enough to stream 16 kHz audio data. That is, for example, when ADPCM codec is used, if the ACL connection interval is greater than 20 ms, this option should be used to avoid data overrun.

2 MTU size and ACL connection interval requirement

To support live audio streaming, the MTU size and ACL connection interval must meet the following for the bandwidth.

Codec	PCM data buffer	Frame size	Max ACL connection interval
ADPCM	320 samples	166	20 ms
OPUS	320 samples	90	20 ms

3 Audio format

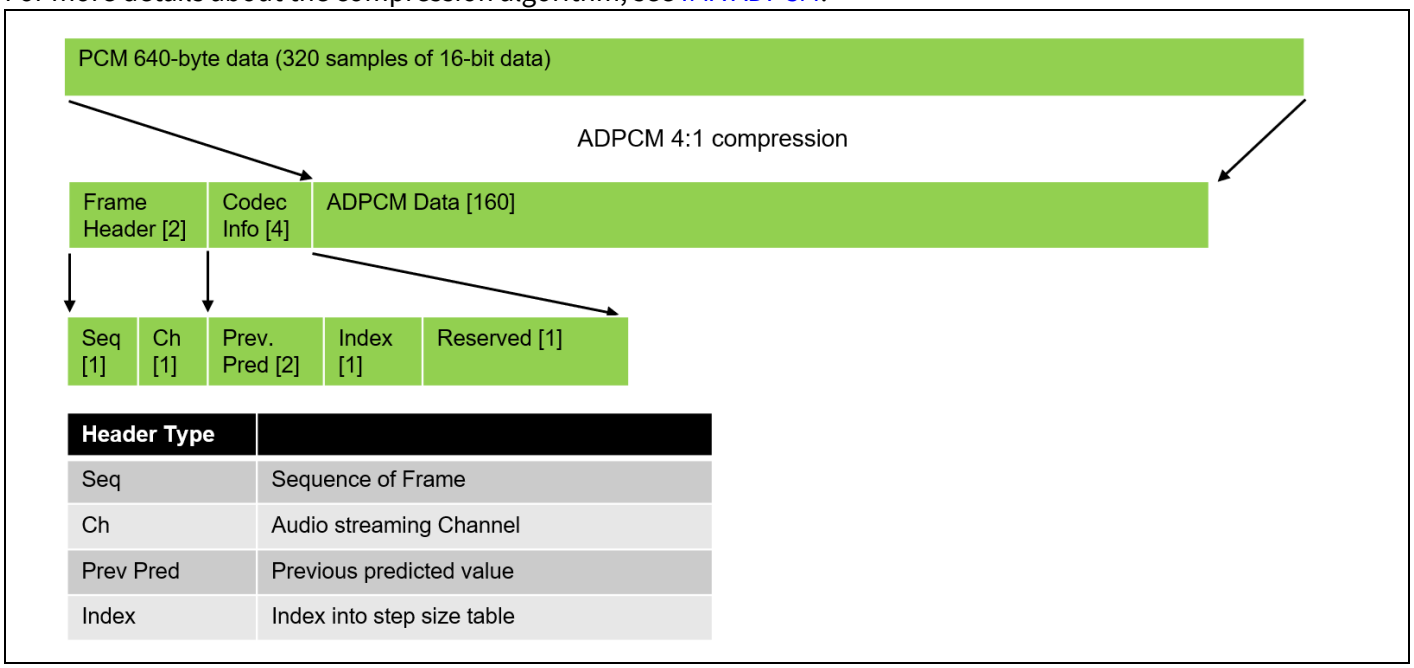
The audio data is 16 bits with a sample rate of 16 kHz PCM raw data. It can be compressed using ADPCM or OPUS encoding method. Each audio frame contains a 2-byte header: Sequence and Channel, followed by the Codec Info and the encoded codec data.

Seq	Ch	Codec info	Encoded codec data

3.1 ADPCM

Each 166-byte audio frame contains a 2-byte frame header and 4-byte codec info along with 160 bytes of compressed audio data. The compression ratio is 4:1; therefore, each frame contains 640 bytes of audio raw data, that is, 320 samples, which is 20 ms of audio data at 16 kHz sampling rate. It means that the ACL connection interval must be less than 20 ms to avoid data overrun.

For more details about the compression algorithm, see IMA ADPCM.



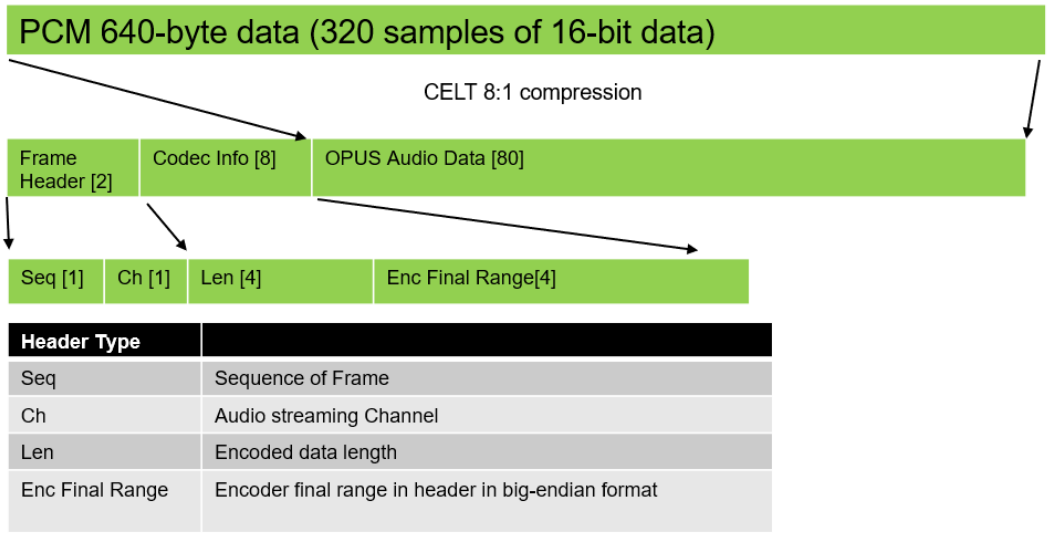
Header Type	
Seq	Sequence of Frame
Ch	Audio streaming Channel
Prev Pred	Previous predicted value
Index	Index into step size table

Figure 2 Compression using ADPCM

3.2 OPUS

Each audio frame contains a 2-byte frame header and 8-byte Codec Info along with 80 bytes of compressed audio data. OPUS uses the Constrained Energy Lapped Transform (CELT) algorithm to compress the 640-byte PCM data (320 samples) to 80-byte OPUS data. 320 samples are equivalent to 20 ms when using the sampling rate of 16 kHz. Therefore, the ACL connection interval must be less than 20 ms.

For more details about the OPUS codec, see Opus (audio format) Wikipedia page.



Header Type	
Seq	Sequence of Frame
Ch	Audio streaming Channel
Len	Encoded data length
Enc Final Range	Encoder final range in header in big-endian format

Figure 3 Compression using OPUS codec

Revision history

Document revision	Date	Description of changes
**	2023-11-30	Initial release

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.
The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc., and any use of such marks by Infineon is under license.

Edition 2023-11-30

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

002-39100 Rev. **

Important notice

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie")

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.