
Getting started with X-CUBE-LocalVUI

Introduction

The X-CUBE-LocalVUI STM32Cube Expansion Package enables local voice recognition on the STM32H7471-DISCO board.

It provides the following features:

- Cloudless voice user interface (UI)
- Audio capture from board microphone
- Integration of automatic speech recognition (ASR) software
 - Sensory TrulyHandsfree™ (THF) software component
 - Sensory TrulyNatural™ (TNL) software component
 - Picovoice Porcupine and Rhino software components
- Capability to build and integrate a customized voice UI vocabulary model
 - Vocabulary model built with Sensory VoiceHub
 - Vocabulary model built with Picovoice console
- Support for wake word detection only, or wake word and command mode
- Detected command logged on Virtual COM port
- Possibility to connect the board as a USB device to record the microphone audio capture

It contains:

- Software components to acquire audio on microphone and interface with the third-party ASR
- Software components to trace detected commands and send captured audio through USB for debug
- Third-party ASR components
- One application based on Sensory THF technology (binary and source code)
- One application based on Sensory TNL technology (binary and source code)
- One application based on Picovoice Porcupine and Rhino combined technologies (binary and source code)

The applications show:

- How to perform an audio capture
- How to interface with voice recognition APIs
- How to integrate new vocabulary models generated with the Sensory VoiceHub or Picovoice console tools

The X-CUBE-LocalVUI Expansion Package also provides a few tools to monitor and debug voice applications.



1 General information

The X-CUBE-LocalVUI Expansion Package runs on the STM32H7 microcontrollers based on the Arm[®] Cortex[®]-M7 processor.

The [Table 1](#) defines the terms used in this document.

Table 1. Acronyms, abbreviations, and definitions

Term	Definition
ASR	Automatic speech recognition
IDE	Integrated development environment
MCU	Microcontroller unit
Porcupine	Picovoice Porcupine wake word engine
PC	Personal computer
Rhino	Picovoice Rhino intent understanding engine
THF	Sensory TrulyHandsfree™
TNL	Sensory TrulyNatural™
UART	Universal asynchronous receiver transmitter
USB	Universal serial bus
Voice UI	Voice user interface

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All other trademarks are the property of their respective owners.



2 Hardware description

2.1 Standard kit contents

The STM32H747I-DISCO Discovery kit is based on the STM32H747XI microcontroller. It provides the required connectivity ports. The introductory documents to both products are available from the product pages at www.st.com:

- STM32H747I-DISCO: *Discovery kit with STM32H747XI MCU data brief (DB3608)*
- STM32H747XI: *Dual 32-bit Arm® Cortex®-M7 up to 480MHz and -M4 MCUs, up to 2MB Flash, 1MB RAM, 46 com. and analog interfaces, SMPS, DSI datasheet (DS12930)*

2.2 User-provided items

- A Windows® development PC, with a free USB port
- At least one USB cable with a Micro-B connector

The PC is used to:

- power hardware
- program firmware
- display the STM32 firmware traces and results onto a serial terminal

The STSW-LINK009 ST-LINK USB driver must be installed on the PC. It is available from the STMicroelectronics website at www.st.com. STSW-LINK009 is required to program the firmware onto the STM32 microcontroller.

2.3 Additional hardware references

Table 2. STMicroelectronics documents

Product	Document type	Document title	Reference
STM32H747I-DISCO	User manual	Discovery kit with STM32H747XI MCU	UM2411
STM32H747XI	Reference manual	STM32H745/755 and STM32H747/757 advanced Arm®-based 32-bit MCUs	RM0399
	Programming manual	STM32F7 Series and STM32H7 Series Cortex®-M7 processor programming manual	PM0253
	Application note	STM32H745/755 and STM32H747/757 lines dual-core architecture	AN5557

Table 3. Arm Limited documents

Document title	Download from
ARMv7-M Architecture Reference Manual	developer.arm.com ⁽¹⁾
Arm Cortex-M7 Processor Technical Reference Manual	

1. This URL belongs to a third party. It is active at document publication. However, STMicroelectronics shall not be liable for any change, move, or inactivation of the URL or the referenced material.

3 Development environment setup

3.1 Programmer

The STM32CubeProgrammer (STM32CubeProg) can be used to program the application binaries to the target. It is available for download from its home page in the STMicroelectronics website at www.st.com.

3.2 Toolchains and IDEs

The X-CUBE-LocalVUI Expansion Package provides support for two toolchains:

- STMicroelectronics STM32CubeIDE: It consists in an Eclipse®-based user interface and the pre-integrated GNU Arm compiler collection.
STM32CubeIDE is available for download from its home page in the STMicroelectronics website at www.st.com.
- IAR Systems® IAR Embedded Workbench® for Arm® (EWARM).
EWARM is available for download from the IAR Systems® website at www.iar.com.

Note: Eclipse is a registered trademark of the Eclipse foundation.

IAR Systems is a registered trademark owned by IAR Systems AB.

Important: When unzipping, the extraction path must not be too long, or the toolchain may report errors while building. In Windows®, it is advised to substitute a drive letter to the installation directory, and to open the project files from this new virtual drive. For instance:

```
> subst P: C:\Users\john\Downloads\X-CUBE-LocalVUI_v1.0.0
```

3.3 Other software required to develop and debug applications

- Serial terminal software emulator to read the console log from the STM32 board via the USB COM serial port (such as Tera Term).
Configure the serial port as follows:
 - 115 200 bauds
 - 8-bit data
 - no parity
 - 1 stop bit
 - no flow control
- Audio capture software, if there is a need, to record the board audio input (such as Audacity®).

4 Hardware setup

- Connect the USB cable from the ST-LINK USB CN2 board connector to the development computer.
The ST-LINK is an in-circuit debugger and programmer soldered on the STM32H7471-DISCO board. It is used for power supply, programming the embedded application in the flash memory, and interacting with the application through the serial Virtual COM port.
- Additionally, it is possible to connect another USB cable from the USB_OTG_HS CN1 board connector. This connection can be used to record the board audio capture on the PC.

5 Use of precompiled applications

5.1 Sensory THF application

The THF precompiled binary is available as X-CUBE-LocalVUI\Bin\STM32H747I-DISCO_xCubeLocalVuiThf.hex.

To use the THF precompiled application, follow these steps:

1. Open STM32CubeProgrammer (STM32CubeProg)
 - a. Find the device serial number ending with “STM32H747I-DISCO” available for connection
 - b. Connect
 - c. Navigate to the tab **[Erase and Programming]>[Download]**
 - d. Choose the file to program:
X-CUBE-LocalVUI\Bin\STM32H747I-DISCO_xCubeLocalVuiThf.hex
 - e. Click on **[Start Program]**
2. Open Tera Term
 - a. Navigate to **[File]>[New Connection]>[Serial]**
 - b. Find the COM port “COMx: STMicroelectronics STlink virtual COMPort (COMx)”
 - c. Click on **[Open]**
3. Reset the board
4. Observe that the traces on the UART are like in the example below:

```
Tslice Version = 6.8.0
Calling SensoryProcessInit
STM32 & THF ready
Wake words =
- sensory
List of available commands =
- turn_on_the_air_conditioning
- turn_the_air_conditioning_off
- turn_on_the_light
- turn_the_light_off
- room_temperature
- outside_temperature
- outside_humidity
- get_the_lights
- make_it_colder
- make_it_hotter
- turn_off_the_air_conditioning
- turn_the_air_conditioning_on
- turn_off_the_light
- turn_the_light_on
----- Listening to wake word ...
```

5. Exercise the application and check in the traces whether the recognition was successful:

```
----- Listening to wake word
... ==> sensory 3
----- Listening to commands ..
==> turn_on_the_air_conditioning
```

5.2 Sensory TNL application

The TNL precompiled binary is available as X-CUBE-LocalVUI\Bin\STM32H747I-DISCO_xCubeLocalVuiTnl.hex.

The steps to program the precompiled application and start the UART traces are the same as in [Section 5.1 Sensory THF application](#).

The traces simply list the wake word, not the commands (refer to the TNL readme file for the set of available commands):

```
Rejection is set to 0.300000
Available vocabulary:
1: "sensory"
STM32 & TNL ready
----- Listening to wake word or commands ...
```

The following traces are displayed when the recognition is successful:

```
Sensory trigger found 9
"small cappuccino"
Printing Intent informations:
nlu : { orderBeverage { size small } { beverage cappuccino } }
orderBeverage : { size small } { beverage cappuccino }
size : small
beverage : cappuccino
```

5.3 Picovoice application

The Picovoice precompiled binary is available as `STM32H747I-DISCO_xCubeLocalVuiPicovoice.hex`.

To use the Picovoice precompiled application, follow these steps:

1. Open STM32CubeProgrammer ([STM32CubeProg](#))
 - a. Find the device serial number ending with “*STM32H747I-DISCO*” available for connection
 - b. Connect
 - c. Navigate to the tab **[Erase and Programming]>[Download]**
 - d. Choose the file to program:
`STM32H747I-DISCO_xCubeLocalVuiPicovoice.hex`
 - e. Click on **[Start Program]**
2. Open Tera Term
 - a. Navigate to **[File]>[New Connection]>[Serial]**
 - b. Find the COM port “*COMx: STMicroelectronics STlink virtual COMPort (COMx)*”
 - c. Click on **[Open]**
3. Reset the board

4. The wake word is “Picovoice”. The traces on the UART simply list the commands:

```

STM32 & PICOVOICE ready
List of available commands =
context:
expressions:
orderBeverage:
- "$beverage:beverage"
- "$numberOfShots:numberOfShots $beverage:beverage"
- "$size:size $beverage:beverage"
- "$numberOfShots:numberOfShots $size:size $beverage:beverage"
- "$size:size $numberOfShots:numberOfShots $beverage:beverage"
- "[brew, can I get, can I have, I want, get me, give me, I'd like, make me, may
I have, I'll have, I'll take, I'll get] [a, an] $beverage:beverage"
- "[brew, can I get, can I have, I want, get me, give me, I'd like, make me,
may I have, I'll have, I'll take, I'll get] [a, an] $numberOfShots:numberOfShots
$beverage:beverage"
- "[brew, can I get, can I have, I want, get me, give me, I'd like, make me, may
I have, I'll have, I'll take, I'll get] [a, an] $size:size $beverage:beverage"
- "[brew, can I get, can I have, I want, get me, give me, I'd like, make me,
may I have, I'll have, I'll take, I'll get] [a, an] $numberOfShots:numberOfShots
$size:size $beverage:beverage"
- "[brew, can I get, can I have, I want, get me, give me, I'd like,
make me, may I have, I'll have, I'll take, I'll get] [a, an] $size:size
$numberOfShots:numberOfShots $beverage:beverage"
slots:
beverage:
- "americano"
- "cappuccino"
- "coffee"
- "espresso"
- "latte"
- "mocha"
numberOfShots:
- "single shot"
- "double shot"
- "triple shot"
size:
- "small"
- "medium"
- "large"

----- Listening to wake word ...

```

6 Applications build from sources

All the applications can be built from their sources with either the STM32CubeIDE or the EWARM toolchains:

1. Double-click on the project in the selected toolchain
2. Compile, program the target, and debug as usual.

Table 4. Sensory THF application projects

Toolchain	Project path
STM32CubeIDE	X-CUBE-LocalVUI\Projects\STM32H747I-Discovery\Applications\sensory_thf\STM32CubeIDE\xCubeLocalVuiThf\project
EWARM	X-CUBE-LocalVUI\Projects\STM32H747I-Discovery\Applications\sensory_thf\EWARM\Project.eww

Table 5. Sensory TNL application projects

Toolchain	Project path
STM32CubeIDE	X-CUBE-LocalVUI\Projects\STM32H747I-Discovery\Applications\sensory_tnl\STM32CubeIDE\xCubeLocalVuiTnl\project
EWARM	X-CUBE-LocalVUI\Projects\STM32H747I-Discovery\Applications\sensory_tnl\EWARM\Project.eww

Table 6. Picovoice application projects

Toolchain	Project path
STM32CubeIDE	X-CUBE-LocalVUI\Projects\STM32H747I-Discovery\Applications\picovoice\STM32CubeIDE\xCubeLocalVuiPicovoice\project
EWARM	X-CUBE-LocalVUI\Projects\STM32H747I-Discovery\Applications\picovoice\EWARM\Project.eww

7 Change of grammar and vocabularies

Sensory

Both Sensory applications can be upgraded with customized grammar and vocabularies. It can be achieved in a few steps:

1. Define a custom vocabulary model from the Sensory VoiceHub portal at voicehub.sensory.com
2. Build from VoiceHub
3. Download the model
4. Uncompress the files and copy them to the proper folder

The way to integrate new vocabularies is described in detail in the readme file of each application.

The Sensory THF application example supports both use cases:

- wake word only
- wake word plus commands

Picovoice

The Picovoice application can be upgraded with customized grammar and vocabularies. It can be achieved in a few steps:

1. Define a custom vocabulary model from the Picovoice console tools at console.picovoice.ai (Porcupine for the wake word, Rhino for the commands)
2. Build the model from the console
3. Download the model
4. Copy the model to the proper folder

The readme file of the Picovoice application describes the way to integrate new vocabularies in detail.

8 Record of board audio input on PC

This feature is provided as a debug tool. It helps to check the correct capture of the board microphone that is sent to the ASR.

Once a USB cable connects the USB_OTG_HS CN1 board connector with the PC, the Audacity® software displays a device called the “*Microphone (STM32 Headset)*”.

Select this device as input to record the audio as captured by the microphone of the board.

Revision history

Table 7. Document revision history

Date	Revision	Changes
27-Apr-2022	1	Initial release.
17-Jun-2022	2	Added the integration of Picovoice products: <ul style="list-style-type: none"> <li data-bbox="730 510 1054 539">• Added Picovoice application <li data-bbox="730 544 1453 595">• Updated Introduction, Applications build from sources, and Change of grammar and vocabularies

Contents

1	General information	2
2	Hardware description	3
2.1	Standard kit contents	3
2.2	User-provided items	3
2.3	Additional hardware references	3
3	Development environment setup	4
3.1	Programmer	4
3.2	Toolchains and IDEs	4
3.3	Other software required to develop and debug applications	4
4	Hardware setup	5
5	Use of precompiled applications	6
5.1	Sensory THF application	6
5.2	Sensory TNL application	7
5.3	Picovoice application	8
6	Applications build from sources	10
7	Change of grammar and vocabularies	11
8	Record of board audio input on PC	12
	Revision history	13
	List of tables	15

List of tables

Table 1.	Acronyms, abbreviations, and definitions	2
Table 2.	STMicroelectronics documents	3
Table 3.	Arm Limited documents	3
Table 4.	Sensory THF application projects	10
Table 5.	Sensory TNL application projects	10
Table 6.	Picovoice application projects	10
Table 7.	Document revision history	13

IMPORTANT NOTICE – 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 acknowledgment.

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. For additional information about ST trademarks, refer to www.st.com/trademarks. 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.

© 2022 STMicroelectronics – All rights reserved