XCORE-VOICE SOLUTION - Quick Start Guide

Release: 1.0.0
Publication Date: 2023/03/20

Table of Contents

1 Product Description

The XCORE-VOICE Solution consists of example designs and a C-based SDK for the development of audio front-end applications to support far-field voice use cases on the xcore.ai family of chips (XU316). The XCORE-VOICE design is currently based on FreeRTOS, leveraging the flexibility of the xcore.ai platform and providing designers with a familiar environment to customize and develop products.

XCORE-VOICE example designs provide turn-key solutions to enable easier product development for smart home applications such as light switches, thermostats, and home appliances. xcore.ai's unique architecture providing powerful signal processing and accelerated AI capabilities combined with the XCORE-VOICE framework allows designers to incorporate keyword, event detection, or advanced local dictionary support to create a complete voice interface solution.

2 Key Features

The XCORE-VOICE Solution takes advantage of the flexible software-defined xcore-ai architecture to support numerous far-field voice use cases through the available example designs and the ability to construct user-defined audio pipeline from the SW components and libraries in the C-based SDK.

These include:

Voice Processing components

- Two PDM microphone interfaces
- Digital signal processing pipeline
- Full duplex, stereo, Acoustic Echo Cancellation (AEC)
- Reference audio via I^{2}S with automatic bulk delay insertion
- Point noise suppression via interference canceller
- Switchable stationary noise suppressor
- Programmable Automatic Gain Control (AGC)
- Flexible audio output routing and filtering
- Support for Wanson or other 3rd party Automatic Speech Recognition (ASR) software

Device Interface components

- Full speed USB2.0 compliant device supporting USB Audio Class (UAC) 2.0
- Flexible Peripheral Interfaces
- Programmable digital general-purpose inputs and outputs

Example Designs utilizing above components

- Far-Field Voice Local Command
- Far-Field Voice Assistance

Firmware Management

- Boot from QSPI Flash
- Default firmware image for power-on operation
- Option to boot from a local host processor via SPI
- Device Firmware Update (DFU) via USB or other transport

Power Consumption

- Typical power consumption 300-350mW
- Low power modes down to 55mW

3 Obtaining the Hardware

The XK-VOICE-L71 DevKit and Hardware Manual can be obtained from the XK-VOICE-L71 product information page.

The XK-VOICE-L71 is based on the: XU316-1024-QF60A

Learn more about the The XMOS XS3 Architecture

4 Obtaining the Software

4.1 Development Tools

It is recommended that you download and install the latest release of the XTC Tools. XTC Tools 15.1.4 or newer are required. If you already have the XTC Toolchain installed, you can check the version with the following command:

```
xcc --version
```

4.2 Application Demonstrations

If you only want to run the example designs, pre-built firmware and other software can be downloaded from the XCORE-VOICE product information page.

4.3 Source Code

If you wish to modify the example designs, a zip archive of all source code can be downloaded from the XCORE-VOICE product information page.

See the Programming Guide for information on:

- Prerequisites
- Instructions for building, running, and debugging the example designs
- Details on the software design and source code

4.3.1 Cloning the Repository

Alternatively, the source code can be obtained by cloning the public GitHub repository.

Note: Cloning requires a GitHub account configured with SSH key authentication.

Run the following *git* command to clone the repository and all submodules:

```
git clone --recurse-submodules git@github.com:xmos/sln_voice.git
```

5 Example Designs

5.1 Far-field Voice Local Command

5.1.1 Overview

These are the XCORE-VOICE far-field local control example designs demonstrating:

- 2-microphone far-field voice control with I^2C or UART interface
- Audio pipeline including interference cancelling and noise suppression
- 16-phrase English language speech recognition

5.1.2 Example designs

5.1.2.1 Low-power Wake-up Demonstration

This is the low-power far-field voice local command (FFD) example design with Wanson speech recognition and local dictionary.

While inactive, low-power mode uses a fraction of energy otherwise required by normal operations while awaiting and processing speech.

When a wake-up phrase is followed by an command phrase, the application will output an audio response and a discrete message over I^2C and UART.

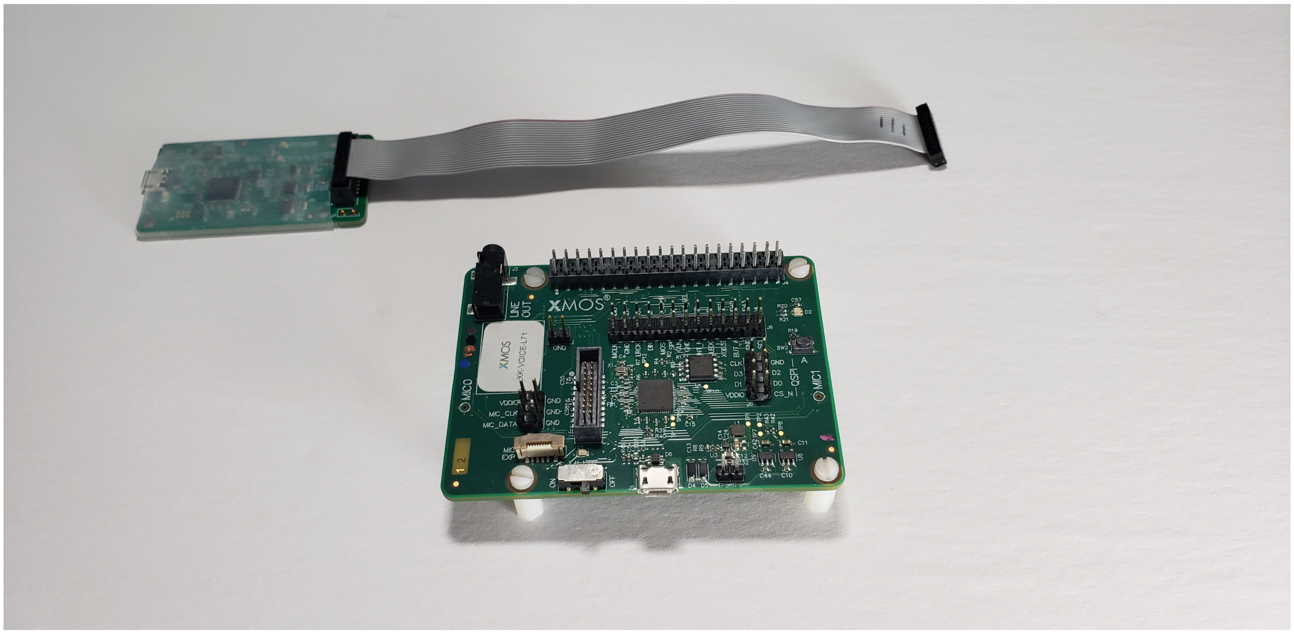
This software is an evaluation version only. It includes a mechanism that limits the maximum number of recognitions to 50. You can reset the counter to 0 by restarting or rebooting the application. The application can be rebooted by power cycling or pressing the SW2 button.

Note: Due to the hardware design, SW2 is only functional when in full-power operation.

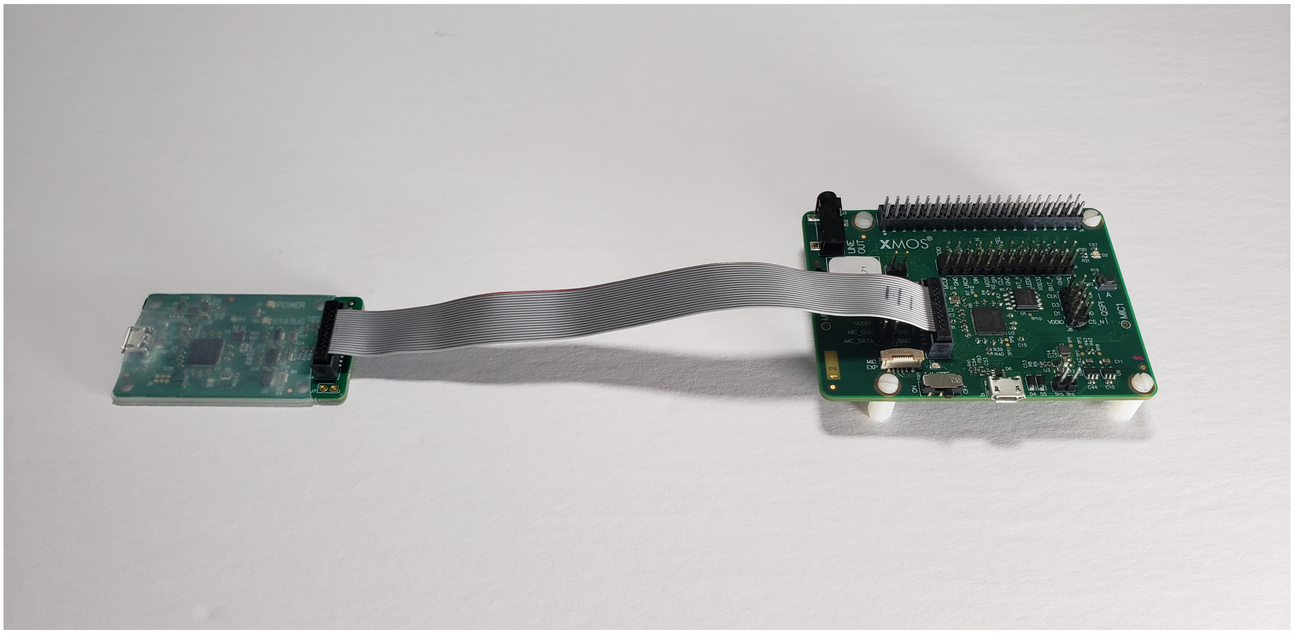
Requirements

- XK-VOICE-L71 board
- Powered speaker(s) with 3.5mm jack connection (OPTIONAL)

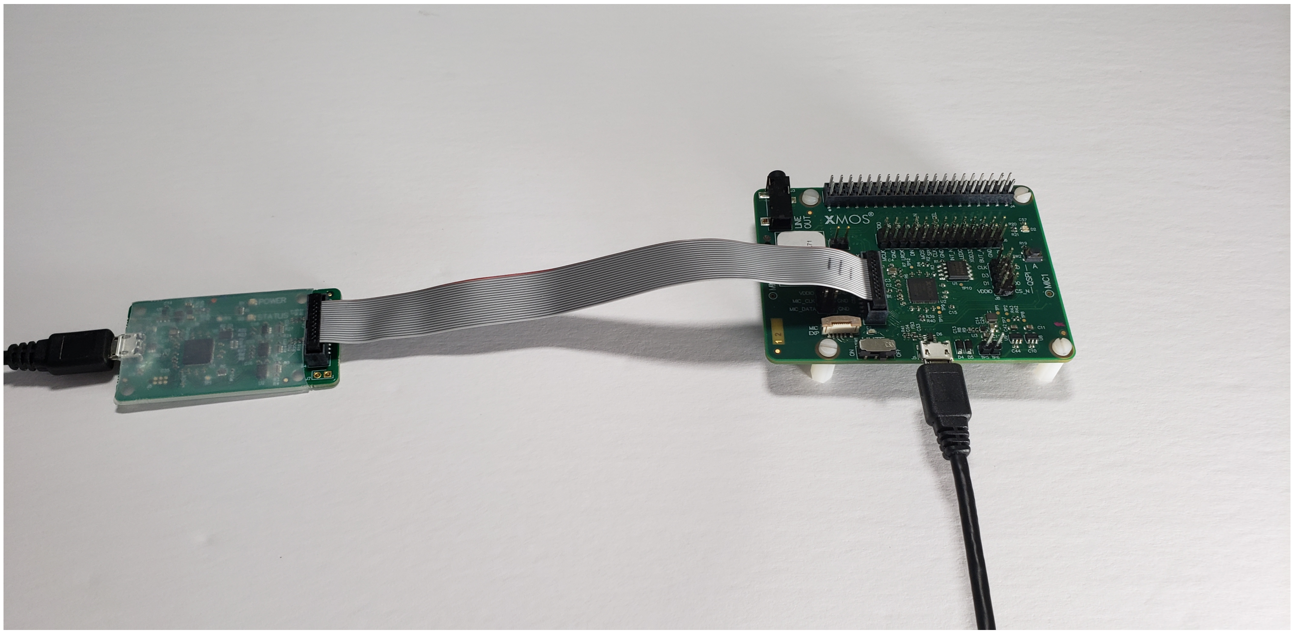
Hardware Setup This example design requires an XTAG4 and XK-VOICE-L71 board.



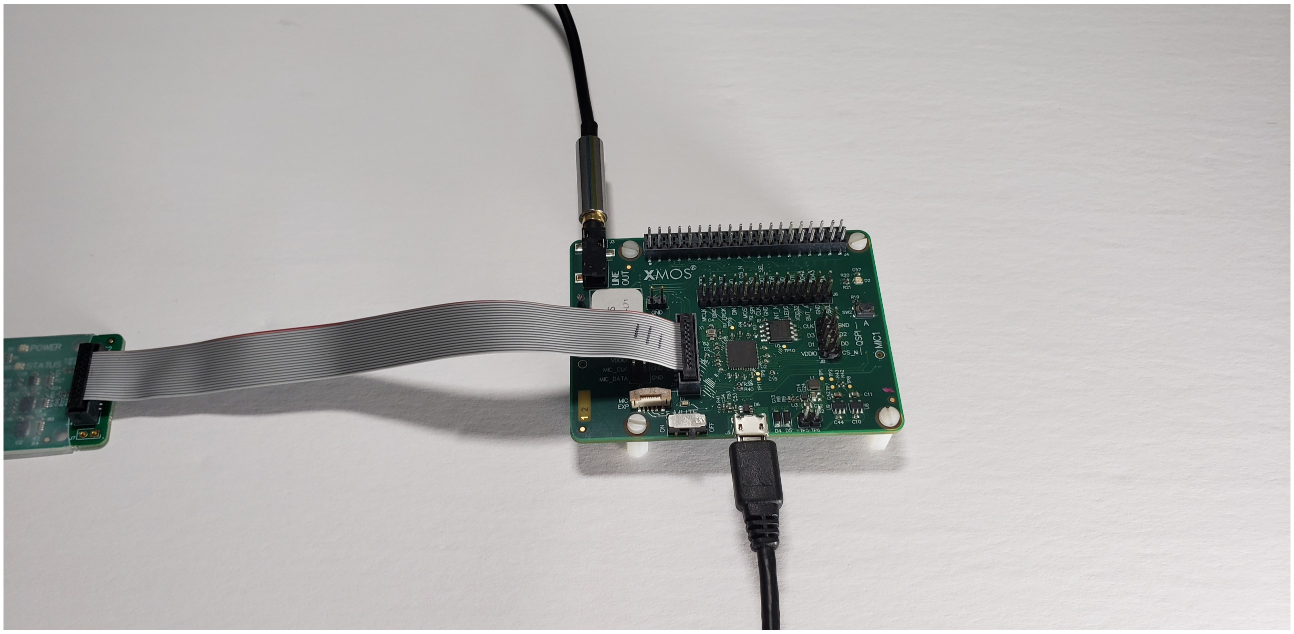
Connect the xTAG to the debug header, as shown below.



Connect the micro USB XTAG4 and micro USB XK-VOICE-L71 to the programming host.



Speakers (OPTIONAL) This example application features audio playback responses. Speakers can be connected to the LINE OUT on the XK-VOICE-L71.



Running the Demonstration

Flashing the Firmware Connect the XTAG4 via USB to the host computer running the XTC tools, and power on the board directly via USB.

On the host computer, open a `XTC Tools Command Prompt`.

```
xflash --quad-spi-clock 50MHz --factory example_ffd.xe --boot-partition-size 0x100000 --
↪data example_ffd_data_partition.bin
```

Being returned to the prompt means flashing has completed, and the XTAG4 may be disconnected.

Speech Recognition Speak one of the wakewords followed by one of the commands from the lists below.

There are three LED states:

- Flashing Green = Full Power, Waiting for Wake Word
- Solid Red & Green = Full Power, Waiting for Command
- Solid Red = Low Power

The application rests in low-power mode (solid red) until the audio pipeline detects audio, thereby entering full-power mode (flashing green) to begin wake-up phrase recognition. Upon recognizing 'Hello XMOS,' waiting begins for a command (solid red & green). After a period of inactivity, low-power mode resumes.

Wakewords

- Hello XMOS

Dictionary Commands

- Switch on the TV
- Switch off the TV
- Channel up
- Channel down
- Volume up
- Volume down
- Switch on the lights
- Switch off the lights
- Brightness up
- Brightness down
- Switch on the fan
- Switch off the fan
- Speed up the fan
- Slow down the fan
- Set higher temperature
- Set lower temperature

Test Wake-up and Low-power Functionality

1. Once flashing is complete, the application is now running on the board.
2. Observe application state. While not detecting sufficient acoustic activity, the demo enters low-power mode. Observe the solid red LED.

3. Say, “Far field voice local control”. The demo enters full-power mode, waiting for the wake-up phrase. Observe the flashing green LED.
4. Speak the wake-up phrase, “Hello XMOS”. The demo plays a recognition tone and awaits a command for a time. Observe the solid red and green LEDs.
5. Say, “Switch on the lights”. The demo recognizes this command, and replies an acknowledgement over speakers, I^{2}C and UART.
6. The demo awaits more commands. Say, “Volume up”. After another acknowledgement, the board will continue to wait for commands.
7. After a period of inactivity, a power-down tone plays and low-power mode resumes.

5.2 Far-field Voice Assistant

5.2.1 Overview

These are the XCORE-VOICE far-field voice assistant example designs demonstrating:

- 2-microphone far-field voice assistant front-end
- Audio pipeline including echo cancelation, interference cancelling and noise suppression
- Stereo reference input and voice assistant output each supported as I^{2}S or USB (UAC2.0)

This application can be used out of the box as a voice processor solution, or extended to run local wakeword engines.

These applications features a full duplex acoustic echo cancellation stage, which can be provided reference audio via I^{2}S or USB audio. An audio output ASR stream is also available via I^{2}S or USB audio.

5.2.2 Example designs

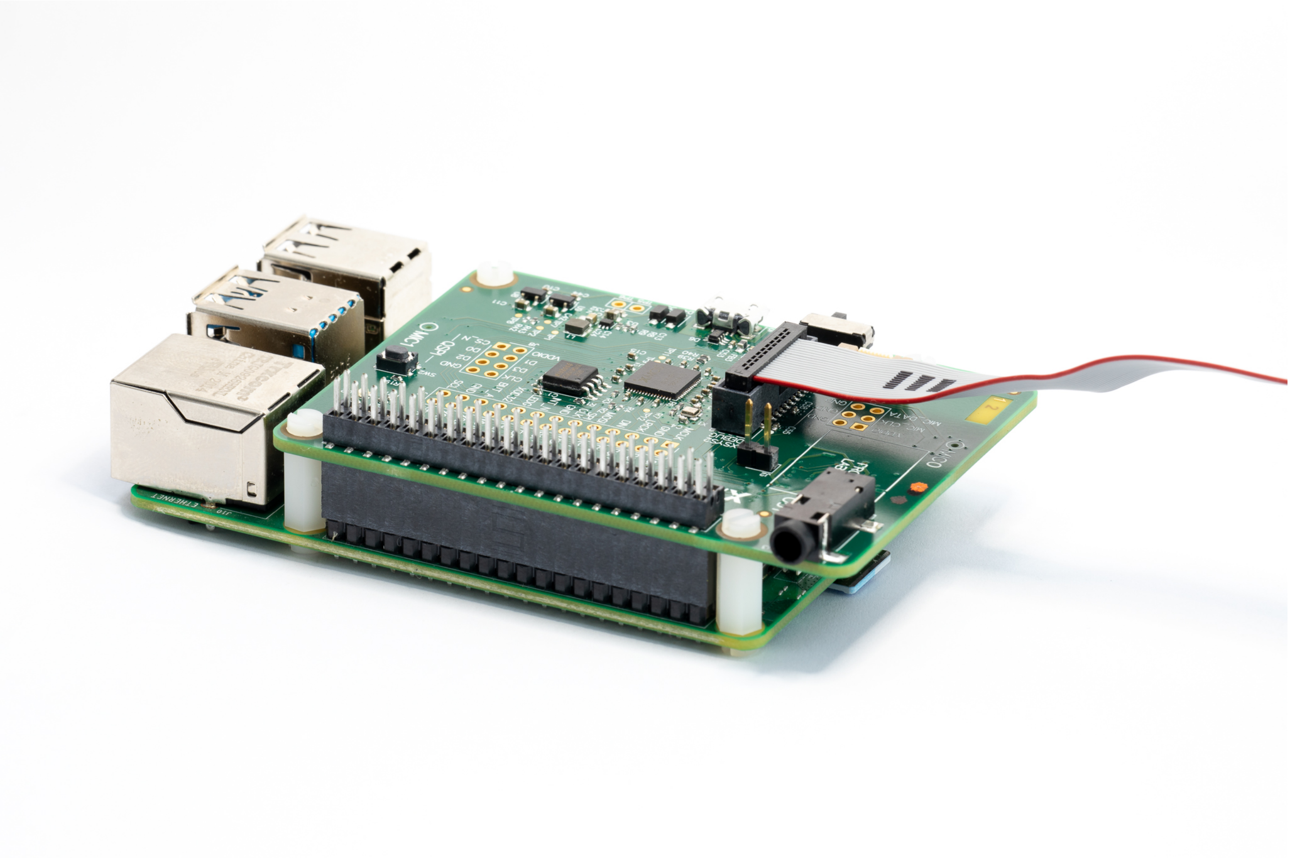
5.2.2.1 USB Audio Demonstration

Direct connection over USB to the host PC allowing signal analysis and evaluation.

Requirements

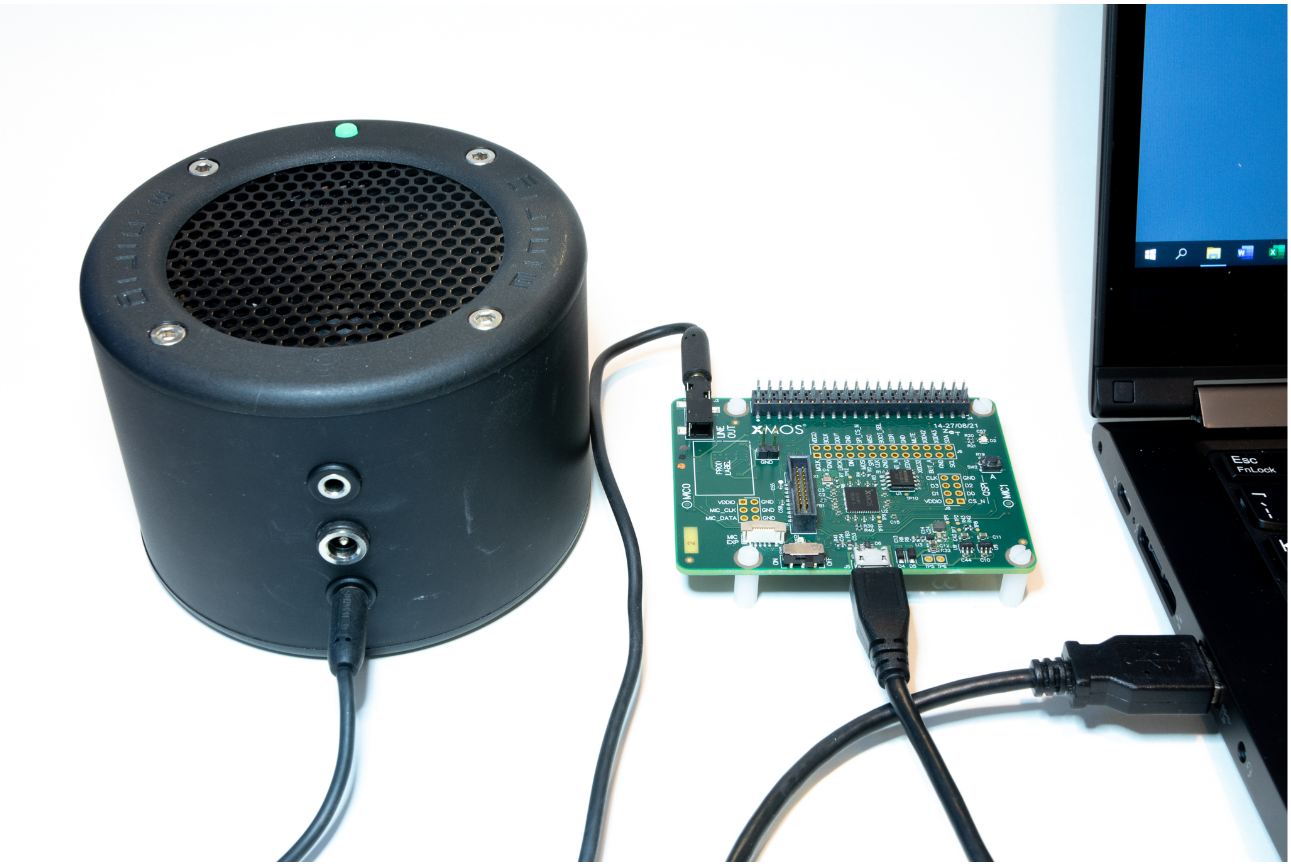
- XK-VOICE-L71 board
- Powered speaker(s) with 3.5mm jack connection
- Host system running Windows, macOS, Linux or Android
- USB A to Micro cable for connection to the host

Hardware Setup Connect either end of the ribbon cable to the XTAG4, and the other end to the XK-VOICE-L71 board as shown (Image shows piggybacked connection to RPi. Standalone operation is also supported):



Running the Demonstration

Configure the Hardware Connect the host system to the micro-USB socket, and the speakers to the jack plug as shown:



Either mono or stereo speakers may be used.

Flashing the Firmware Connect the XTAG4 via USB to the host computer running the XTC tools, and power on the board (either via RPi or directly via USB).

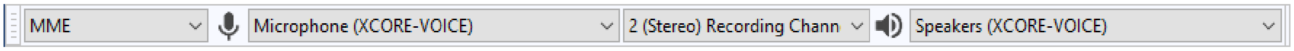
On the host computer, open a `XTC Tools Command Prompt`.

```
xflash --quad-spi-clock 50MHz --factory example_ffva_ua_adec.xe --boot-partition-size␣
↪0x100000 --data example_ffva_ua_adec_data_partition.bin
```

Being returned to the prompt means flashing has completed, and the XTAG4 may be disconnected.

Record Captured Voice

1. Open a music player on host PC, and play a stereo file.
2. Check music is playing through powered speakers.
3. Adjust volume using music player or speakers.
4. Open Audacity and configure to communicate with kit. Input Device: XCORE-VOICE Voice Processor and Output Device: XCORE-VOICE Voice Processor
5. Set recording channels to 2 (Stereo) in Device



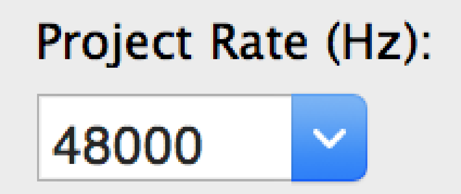
MME

Microphone (XCORE-VOICE)

2 (Stereo) Recording Chann

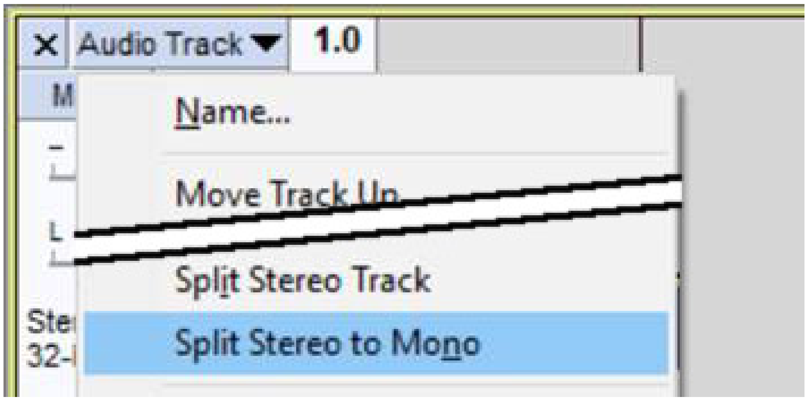
Speakers (XCORE-VOICE)

6. Set Project Rate to 48000Hz in Selection Toolbar.

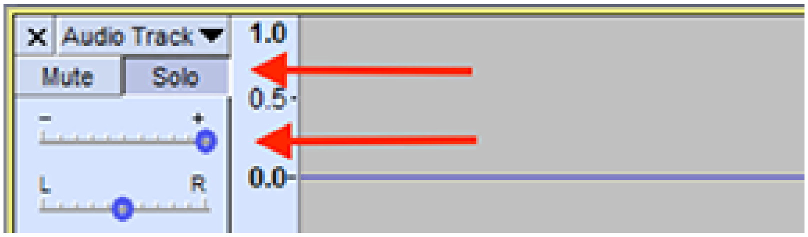


Project Rate (Hz): 48000

7. Click Record (press 'r') to start capturing audio streamed from the XCORE-VOICE device.
8. Talk over music; move around the room while talking.
9. Stop music player.
10. Click Stop (press space) to stop recording. Audacity records single audio channel streamed from the XCORE-VOICE kit including extracted voice signal.
11. Click dropdown menu next to Audio Track, and select Split Stereo To Mono.



12. Click Solo on left channel of split processed audio. Increase Gain slider if necessary.



13. Click Play (press space) to playback processed audio.

Only your voice is audible. Playback music is removed by acoustic echo cancellation; voice is isolated by interference canceller; background noise is removed by noise suppression algorithms.

5.2.2.2 Alexa Voice Service (AVS) Demonstration

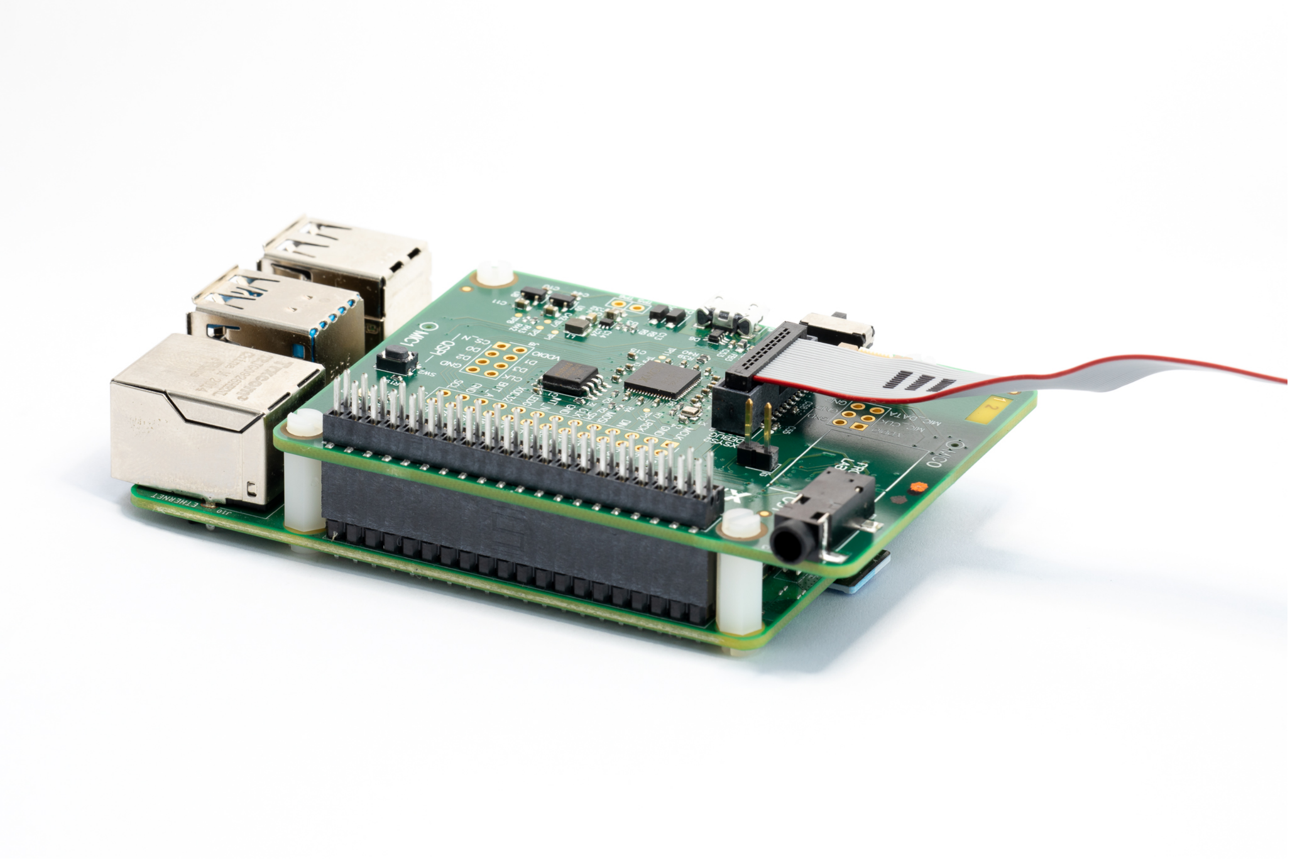
Integration into RPi system, using I2S, running an Alexa Voice Service (AVS) client.

Requirements

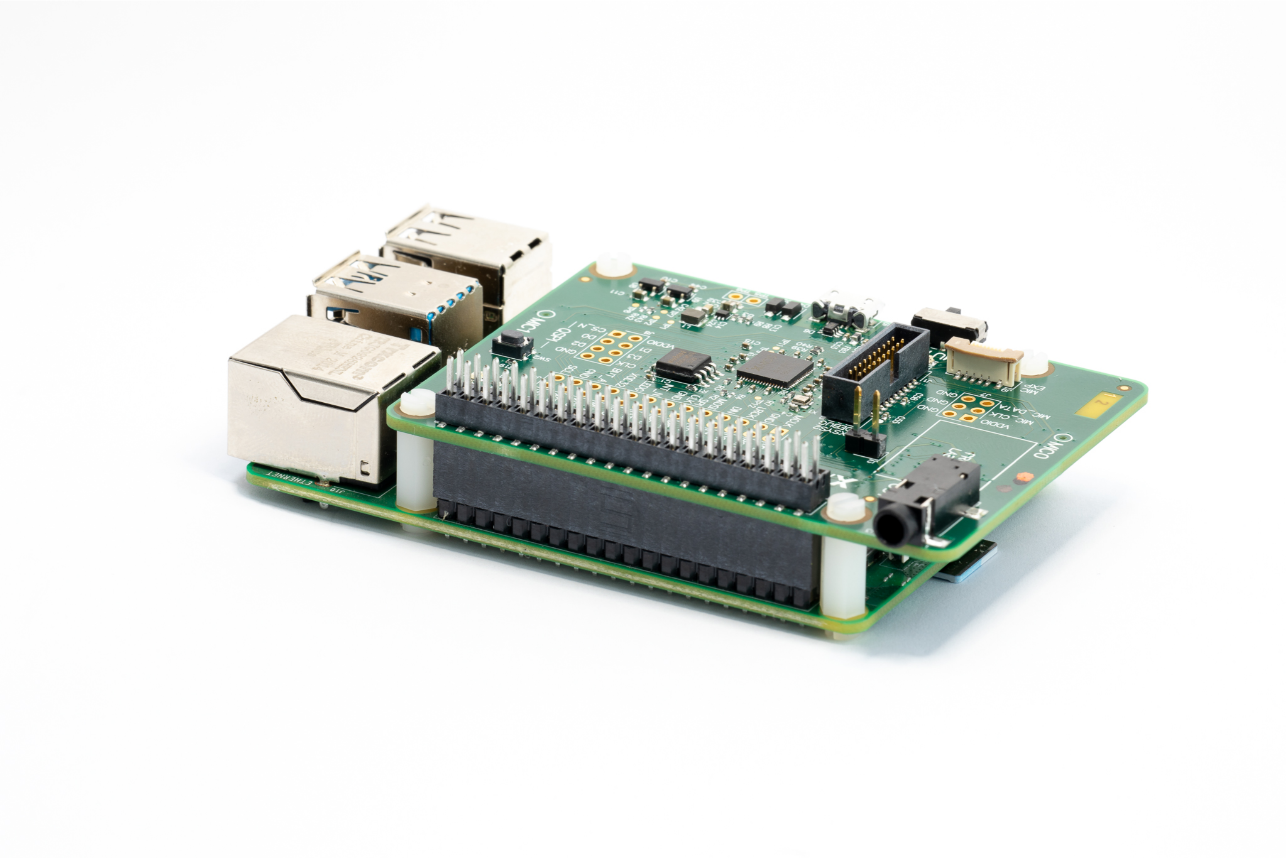
- XK-VOICE-L71 board
- Powered speaker(s) with 3.5mm jack connection
- Raspberry Pi model 3 or 4 with power unit
- HDMI monitor, USB keyboard and mouse

- SD card (minimum 16GB size)
- Amazon Developer Account

Hardware Setup Connect either end of the ribbon cable to the XTAG4, and the other end to the XK-VOICE-L71 board as shown (Image shows piggybacked connection to RPi. Standalone operation is also supported):



Connect the XV-VOICE-L71 to the Raspberry Pi ensuring that the connector fully lines up, as shown below.



Running the Demonstration

Flashing the Firmware Connect the XTAG4 via USB to the host computer running the XTC tools, and power on the board (either via RPi or directly via USB).

On the host computer, open a `XTC Tools Command Prompt`.

Navigate to the root directory of the sln_voice repository.

Run the following commands to flash the firmware.

```
xflash --quad-spi-clock 50MHz --factory example_ffva_int_adec.xe --boot-partition-size␣
↪0x100000 --data example_ffva_int_adec_data_partition.bin
```

Being returned to the prompt means flashing has completed, and the XTAG4 may be disconnected.

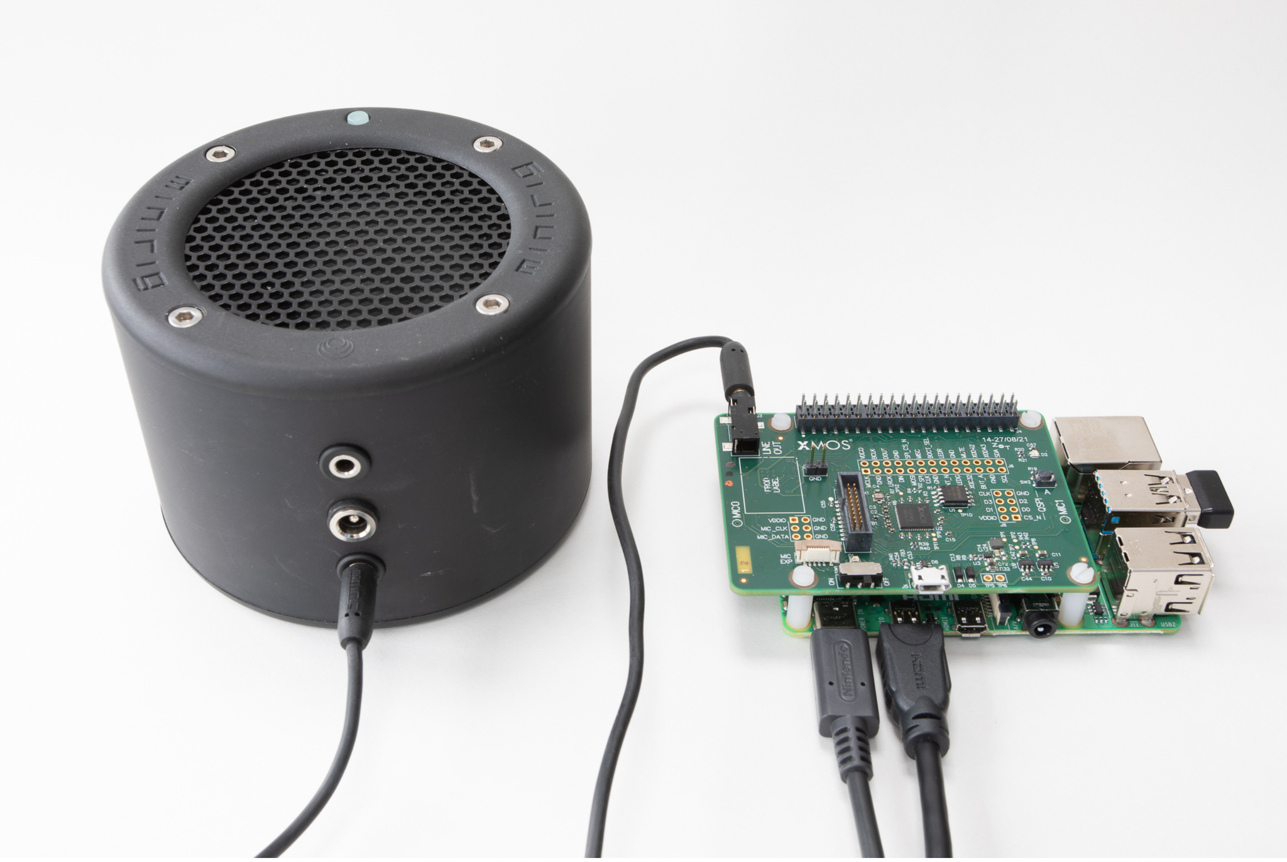
Prepare the Raspberry Pi System

Note: The FFVA-INT firmware is compatible with XVF3610-INT software, therefore instructions for installing the XVF3610-INT pi software can be followed for this AVS demo. The “Firmware Upgrade” section may be dismissed, as your FFVA-INT firmware is already updated per the above section of this guide.

Note: While ‘apt update’ will break the install, the user can and must ‘apt-get update’ before the auto_install.sh script is called, or package installation will fail.

Prepare the Raspberry Pi System image on the SD card by following the instructions for XVF3610-INT as described on github

Connect the System Connect the speakers (into the XV-VOICE-71), HDMI monitor cable, and mouse as shown:



Install and Configure Install the Amazon Alexa SDK and configure the Raspberry Pi Audio by following the instructions here:

AVS Setup Instructions

Run Demo and Test Virtual Assistant Functionality

1. Once installation is complete, avsrun will run on startup. Otherwise, run the demo by typing `avsrun` in a terminal. The demo will now operate as an Alexa virtual assistant.
2. After successful device authorization, avsrun always displays 'Alexa is currently idle!' while waiting for the user.
3. Tap to talk can be used to simulate wakeword functionality. Type 't' and hit Enter, and avsrun displays 'Listening…'
4. The device waits a short time for a query. Try speaking "What is the weather?" and avsrun displays 'Thinking…'
5. Soon after, the virtual assistant should audibly answer over speaker as avsrun displays 'Speaking…'

6. Barge-in may also be tested. Before the assistant has finished speaking, Type ‘t’ and hit Enter. The assistant will stop and await a new command.

6 Copyright & Disclaimer

7 Licenses

7.1 XMOS

All original source code is licensed under the XMOS License.

7.2 Third-Party

Additional third party code is included under the following copyrights and licenses:

Table 7.1: Third Party Module Copyrights & Licenses

Module	Copyright & License
dr_wav	Copyright (C) 2022 David Reid, licensed under a public domain license
Wanson Speech Recognition Library	The Wanson speech recognition library is Copyright 2022. Shanghai Wanson Electronic Technology Co. Ltd ("WANSON") and is subject to the Wanson Restrictive License