# XCORE-VOICE SOLUTION - Quick Start Guide

Publication Date: 2024/12/18
Document Number: v2.3.0

IN THIS DOCUMENT

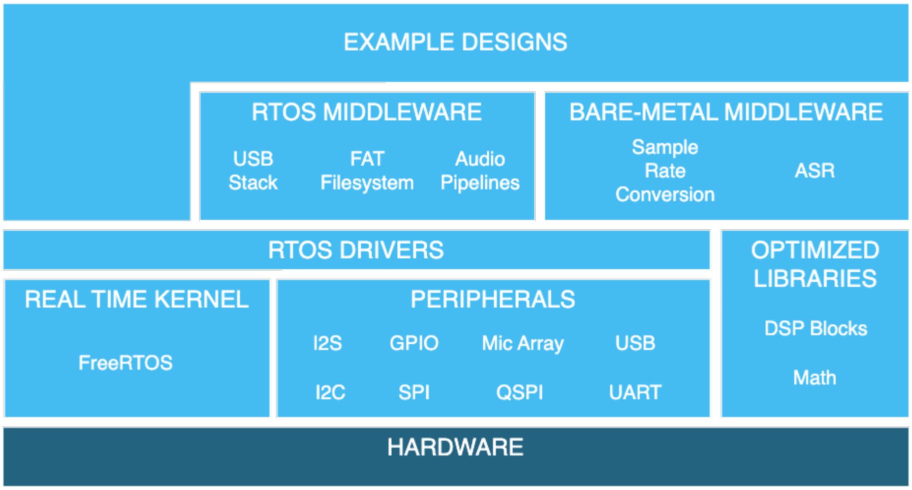
# 1 Product Description

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 examples are currently based on FreeRTOS or bare-metal, leveraging the flexibility of the xcore.ai platform and providing designers with a familiar environment to customize and develop products.

XCORE-VOICE example designs include 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. Bridging designs including PDM microphone to host aggregation are also included showcasing the use of xcore.ai as an interfacing and bridging solution for deployment in existing systems.

The C SDK is composed of the following components:

- Peripheral IO libraries including; UART, I$^2$C, I$^2$S, SPI, QSPI, PDM microphones, and USB. These libraries support bare-metal and RTOS application development.
- Libraries core to DSP applications, including vectorized math and voice processing DSP. These libraries support bare-metal and RTOS application development.
- Libraries for speech recognition applications. These libraries support bare-metal and RTOS application development.
- Libraries that enable multi-core FreeRTOS development on xcore including a wide array of RTOS drivers and middleware.
- Pre-build and validated audio processing pipelines.
- Code Examples - Examples showing a variety of xcore features based on bare-metal and FreeRTOS programming.
- Documentation - Tutorials, references and API guides.



## 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 Sensory, Cyberon 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
- Low Power 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 I$^2$C

**Power Consumption**

- FFD/FFVA: 300-350mW (Typical)
- Low Power FFD: 110mW (Full-Power), 54mW (Low-Power), <50mW possible with Sensory's LPSD under certain conditions.

# 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

The XCORE-AI-EXPLORER DevKit and Hardware Manual used in the Microphone Aggregation example can be obtained from the XK-VOICE-L71 product information page.

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 15.3.0 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
```

If you have previously cloned the repository or downloaded a zip file of source code, the following commands can be used to update and fetch the submodules:

```
git pull
git submodule update --init --recursive
```

# 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$^2$C 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 Demonstration

This is the far-field voice local command (FFD) example design. Two examples are provided: both examples include speech recognition and a local dictionary. One example uses the Sensory TrulyHandsfree™ (THF) libraries, and the other one uses the Cyberon DSPotter™ libraries.

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

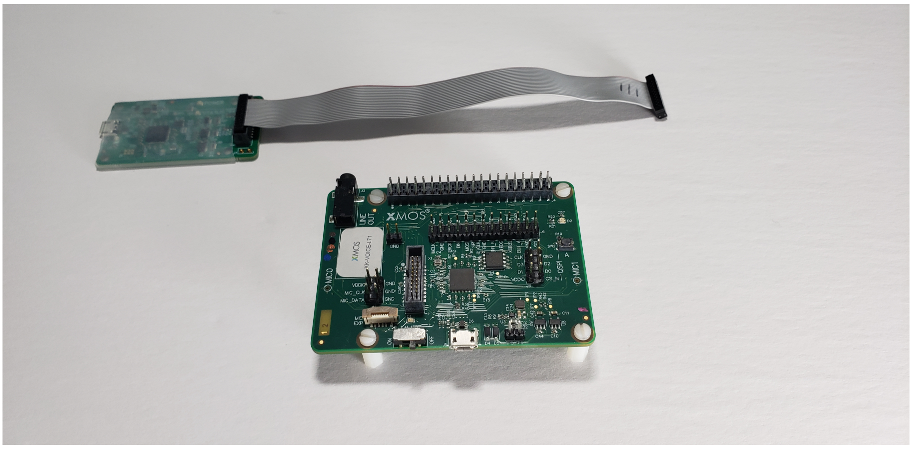
Sensory's THF and Cyberon's DSpotter™ libraries ship with an expiring development license. The Sensory one will suspend recognition after 11.4 hours or 107 recognition events, and the Cyberon one will suspend recognition after 100 recognition events. After the maximum number of recognitions is reached, a device reset is required to resume normal operation. To perform a reset, either power cycle the device or press the SW2 button.

Production software runs on a special device. Contact Cyberon, Sensory or XMOS sales for information about production use of the device.

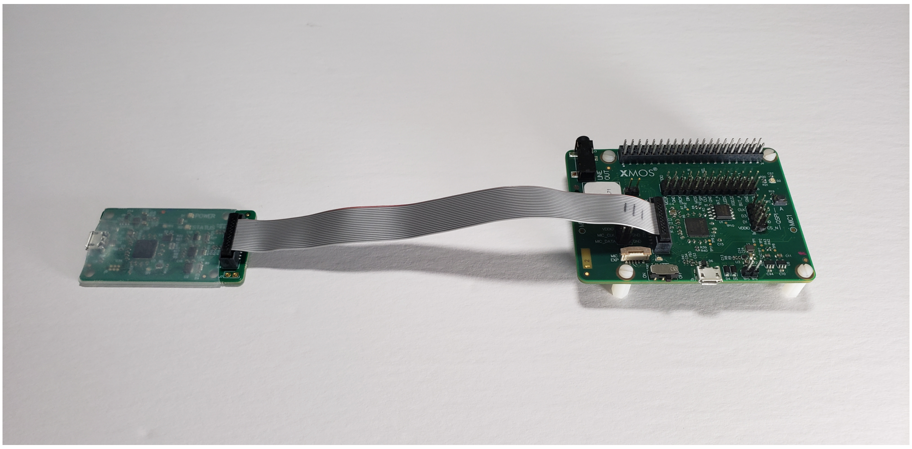
#### Requirements

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

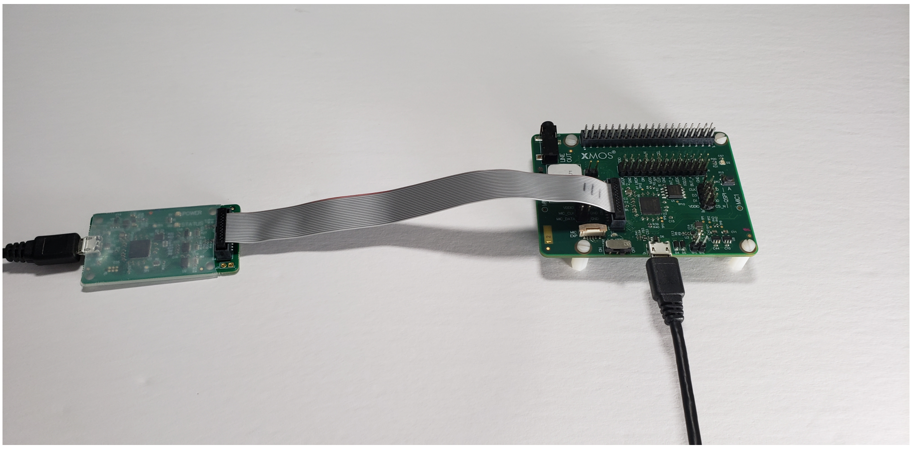
**Hardware Setup** This example design requires an XTAG4 and XK-VOICE-L71 board.



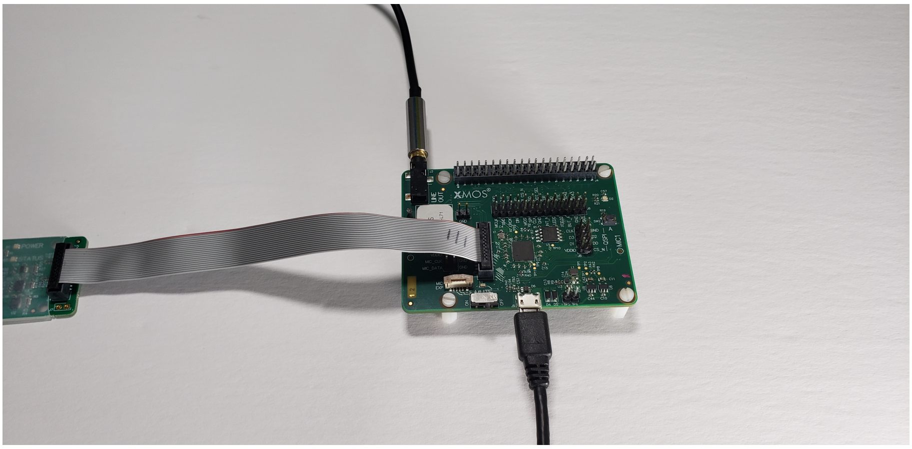
Connect the xTAG to the debug header, as shown below.



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 = Waiting for Wake Word
- Solid Red & Green = Waiting for or Processing Command
- Fast Flashing Red = Evaluation period has expired

The application resets waiting for the wakeword (flashing green). Upon recognizing ‘Hello XMOS’ or ‘Hello Cyberon’ (DSpotter™ model only), waiting begins for a command (solid red & green). After a period of inactivity, or successful command processing the application returns to waiting for wakeword (flashing green).

Sensory TrulyHandsfree™ and Cyberon DSpotter™ models detect the same commands, as listed below.

### Wakewords

- Hello XMOS
- Hello Cyberon (DSpotter™ model only)

### 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

## 5.2 Low Power Far-field Voice Local Command

### 5.2.1 Overview

This is the XCORE-VOICE low power far-field local control example designs demonstrating:

- Low power control/handling
- Small wake word model in SRAM
- 2-microphone far-field voice control with I$^2$C or UART interface
- Audio pipeline including interference cancelling and noise suppression
- 16-phrase English language speech recognition

### 5.2.2 Example designs

**5.2.2.1 Demonstration** The low power far-field voice local command (Low Power FFD) example design targets low power speech recognition using Sensory's TrulyHandsfree™ (THF) speech recognition and local dictionary.

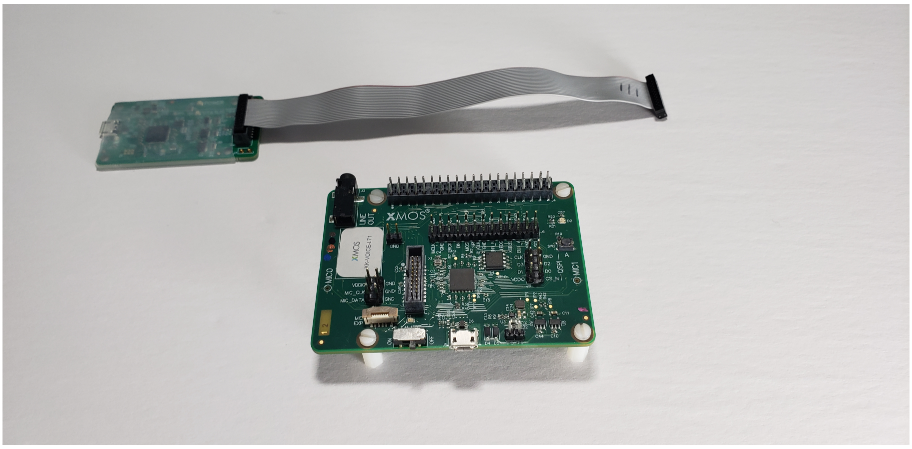
When the small wake word model running on tile 1 recognizes a wake word utterance, the device transitions to full power mode where tile 0's command model begins receiving audio samples, continuing the command recognition process. On command recognition, the application outputs a discrete message over I$^2$C and UART.

Sensory's THF software ships with an expiring development license. It will suspend recognition after 11.4 hours or 107 recognition events; after which, a device reset is required to resume normal operation. To perform a reset, either power cycle the device or press the SW2 button. Note that SW2 is only functional while in full power mode (this application is configured to hold the device in full-power mode on such license expiration events).

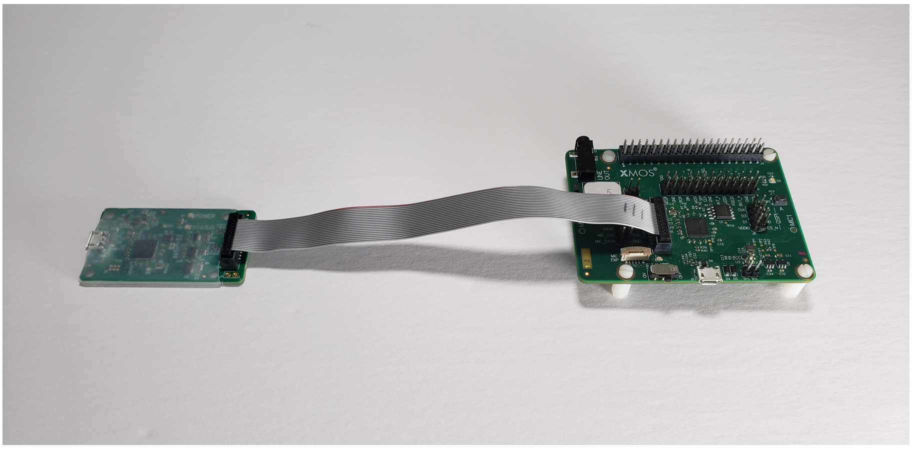
**Required Hardware**

- XK-VOICE-L71 board
- XTAG4 debug adapter
- 2x USB-Micro B cables
- Host computer for programming

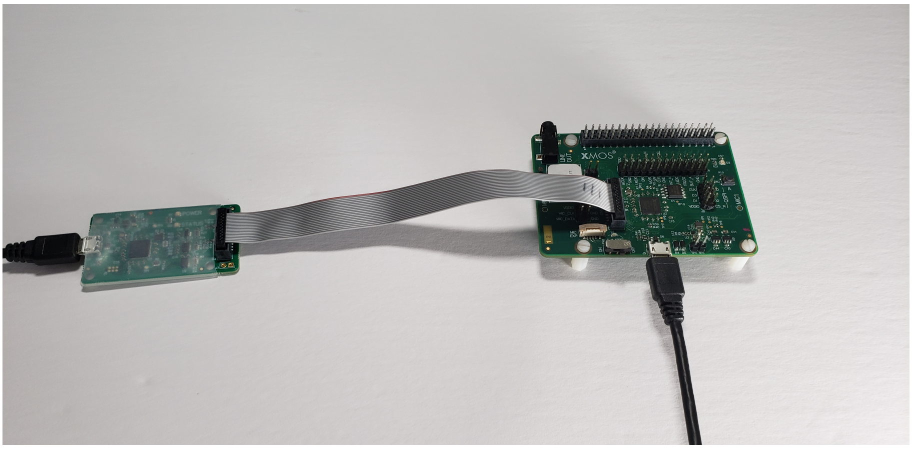
**Hardware Setup** This example design requires an XTAG4 and XK-VOICE-L71 board.



Connect the XTAG4 to the debug header, as shown below.



Connect the both USB Micro-B connections on the XTAG4 and XK-VOICE-L71 to the programming host computer.



### 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_low_power_ffd.xe --boot-partition-size 0x100000 --data example_
↪low_power_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 wake words followed by one of the commands from the lists below.

There are four LED states:

- Solid Red = Low Power. Waiting for wake word.
- Blinking Green = Full power. Waiting for command.
- Solid Red & Green = Full power. Processing command.
- Flickering Red = Full power. End of evaluation (device reset required).

On startup, the application enters low power mode and waits for the wake word. Upon wake word recognition, the device enters full power mode and waits for a command. Upon command recognition, the device will queue the command for processing. On each wake word or command recognition, a timer is reset (per tile). On expiration of the intent engine's timer, the device will request a transition to low power. The other tile may reject the request in cases where its timer has not expired or other application-specific reasons.

## Supported Wake Word

- Hello XMOS

## Supported 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

## 5.3 Far-field Voice Assistant

### 5.3.1 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.

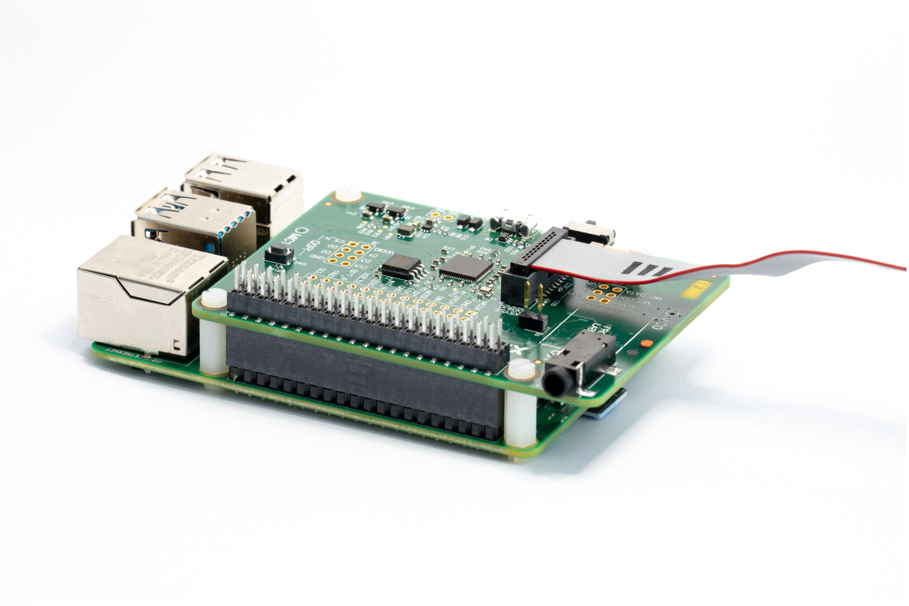
### 5.3.2 Example designs

**5.3.2.1 USB Audio Demonstration** Direct connection over USB to the host PC allowing signal analysis and evaluation.

#### Requirements

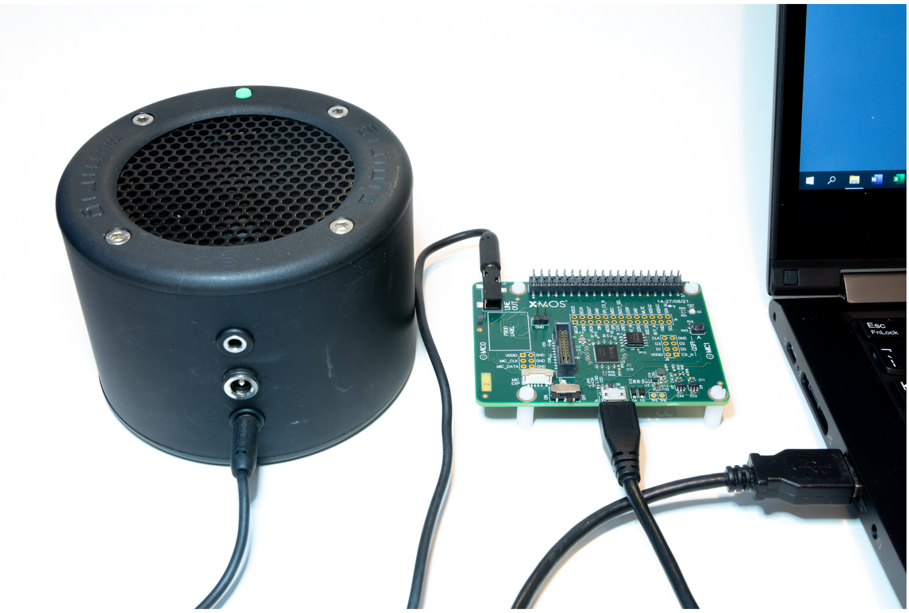
- XK-VOICE-L71 board
- Powered speaker(s) with 3.5mm jack connection
- Host system running Windows, macOS, Linux or Android
- USB A to Micro cable for connection to the host

**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).

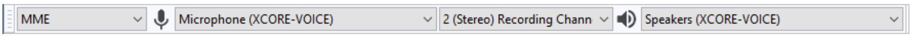
On the host computer, open a `XTC Tools Command Prompt`.

```
xflash --quad-spi-clock 50MHz --factory example_ffva_ua_adec_altarch.xe --boot-partition-size 0x100000 --data example_ffva_ua_adec_altarch_data_partition.bin
```

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

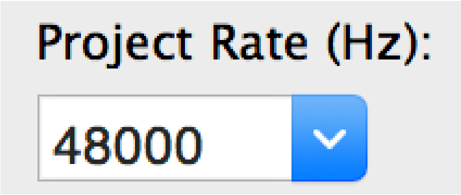
### Record Captured Voice

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



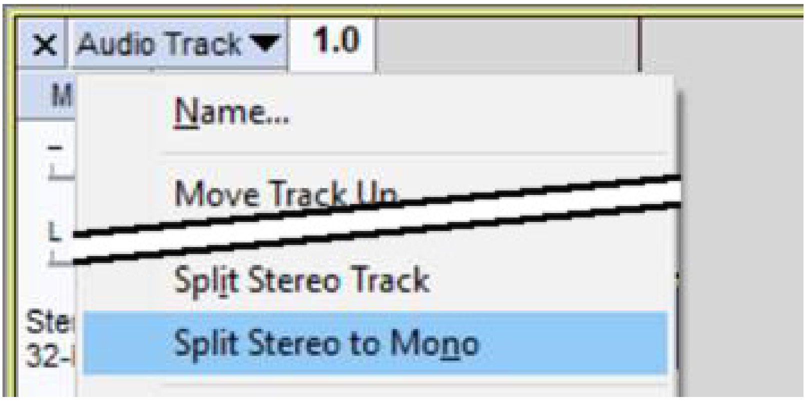
MME | Microphone (XCORE-VOICE) | 2 (Stereo) Recording Chann | Speakers (XCORE-VOICE)

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

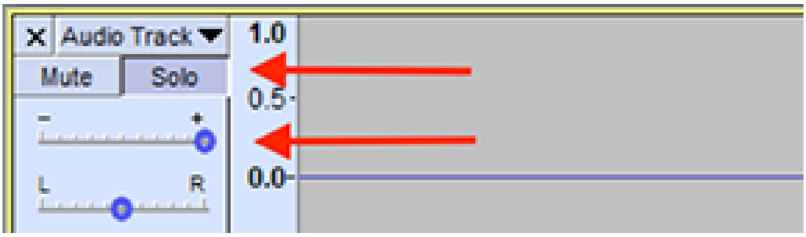


Project Rate (Hz): 48000

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.

# 6 Licenses

## 6.1 XMOS

All original source code is licensed under the XMOS License.

## 6.2 Third-Party

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

Table 1: Third Party Module Copyrights & Licenses

| Module | Copyright & License |
|---|---|
| dr_wav | Copyright (C) 2022 David Reid, licensed under a public domain license |
| FatFS | Copyright (C) 2017 ChaN, licensed under a BSD-style license |
| FreeRTOS | Copyright (c) 2017 Amazon.com, Inc., licensed under the MIT License |
| Sensory TrulyHandsfree™ | The Sensory TrulyHandsfree™ speech recognition library is *Copyright (C) 1995-2022 Sensory Inc.* and is provided as an expiring development license. Commercial licensing is granted by Sensory Inc. |
| Cyberon DSpotter™ | For any licensing questions about Cyberon DSpotter™ speech recognition library please contact Cyberon Corporation. |
| TinyUSB | Copyright (c) 2018 hathach (tinyusb.org), licensed under the MIT license |

# 7 Other examples

Where no quickstart guide exists such as for Microphone Aggregation, Asynchronous Sample Rate Conversion and Automatic Speech Recognition with Cyberon library, please consult the Programming Guide which contains setup information for these applications.

Xmos Ltd. is the owner or licensee of this design, code, or Information (collectively, the “Information”) and is providing it to you “AS IS” with no warranty of any kind, express or implied and shall have no liability in relation to its use. Xmos Ltd. makes no representation that the Information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.