IA611 Xplained Pro - Demo Software User Guide

External Use

Document #AUD-ESP-00460-V1.0

This document describes the software architecture and methodology of the Knowles IA611 Xplained Pro - Demo Extension Kit (PN IA611-RDI-01). It also describes the APIs used in the Atmel Studio development environment. The extension kit is the IA611 Reference Design for IoT designed to be used with the Microchip SAM D21 Explained Pro MCU.

Filename: IA611_Xplained_Pro_Demo_Software_User_Guide.docx

Table of Contents

List of Figures

Chapter 1: Introduction

The Knowles **IA611 Reference Design for IoT** is based on IA611 Smart Mic. The IA611 enables ultra-low power voice and event detection including voice UI supporting keywords and voice commands and acoustic event detection. To achieve low power and compact design, the reference design is intended to be implemented with a readily available MCU board. The implementation discussed is this guide is the **IA611 Xplained Pro** development board designed to be used with the Microchip SAM D21 Explained Pro MCU.

This document will explain the software sample code and project supplied to run Knowles VoiceQ keyword and command detection on the IA611 Smart Mic.

This version of the IA611 Xplained Pro software is the demo version which excludes the IA611 binaries required to build the example project. This version includes prebuild binaries for SAMD21 Xplained Pro board to demonstrate the VoiceQ example using 1 OEM Wake Keyword + 3 commands. There pre-built binaries can be found under folder Binfiles/SAMD21_bin. Select appropriate ".hex" files to flash on SAMD21 according to the IA611 Host interface selected by jumpers on board (Refer to Chapter 2: HW Setup).

RDI_VQ-SAMD21_UART_V_2_0_3.hex ➔ UART Interface

RDI_VQ-SAMD21_I2C_V_2_0_3.hex ➔ I2C Interface

RDI_VQ-SAMD21_SPI_V_2_0_3.hex ➔ SPI Interface

Full version of the IA611 Xplained Pro Software:

Full version of the IA611 Xplained Pro software can be obtained from https://solutions.knowles.com/. The Full version provides three example samples based on Knowles VoiceQ algorithm.

1. **Voice Q Example:** 1 OEM Wake Keyword + 3 Commands
2. **Voice ID Example:** 1 User Trained Voice ID + 3 Commands
3. **UTK Example:** 1 User Trained Keyword + 3 commands

WakeUp Keyword (one of following)	LED0 Blinks	Command	LED0 Blinks
1. Hello VoiceQ 2. Hello VoiceQ (user trained VoiceID) 3. User Trained Keyword	1 time	Switch the Light	2 times
		Next Song	3 times
		Biadu-Yi Xia	4 times

An Android phone Application is (not included in demo version) to create user trained Voice ID and UTK models. You can find this app in "Tools" folder in full version of example project. Follow the instructions from *VoiceID-UTK Trainer App User Guide* document to install and use the app to generate the keyword models.

For all of the above three project configurations, the IA611 sample project can be built for one of three different hardware interfaces: UART, I2C and SPI.

1.1 Document Scope

This document is intended to demonstrate the IA611 smart phone functionality with single use case (1 OEM Wake Keyword + 3 Command Words). The solution provides the developer a very low touch example of programming and interfacing with the IA611 smart microphone with integrated DSP.

1.2 Product Overview

1.2.1 Hardware Components

The IA611 Xplained Pro development board connects to the SAM D Xplained Pro series of development boards. The IA611 Xplained Pro board contains an ID pin to identify the board and help integrate the solution into the Studio 7 ecosystem.

1.2.2 Software Components

Please install Atmel Studio 7 from:

https://www.microchip.com/avr-support/atmel-studio-7

Studio 7 is an integrated development platform for developing and debugging all AVR and SAM microcontroller applications.

1.2.3 Additional Hardware Requirements

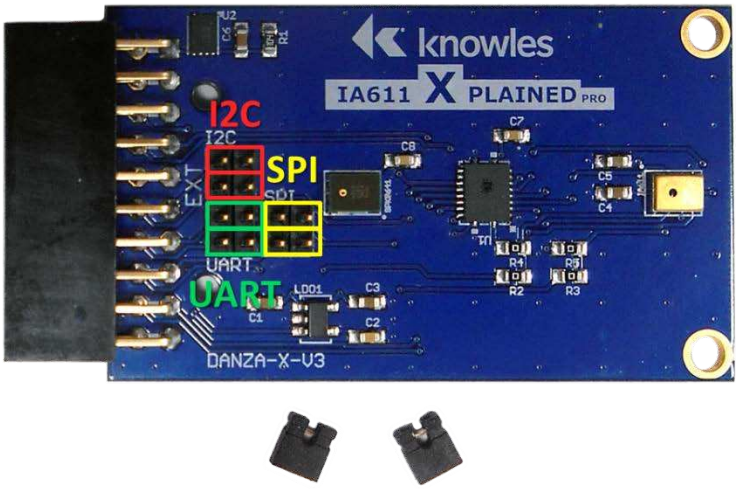
For this application, the SAM D21 Xplained Pro Evaluation Kit is used. This kit is available from Microchip or an authorized Microchip distributor. The part number for this kit is: ATSAM D21-XPRO and is available from distributors such as Arrow, Digi-Key, Newark and Mouser.

Chapter 2: Hardware Setup

This section will explain the simple task of setting up the hardware to get your project setup, built and debugged using Studio 7 integrated with the SAM D21 development kit.

2.1 Jumpers

The IA611 Xplained Pro board supports three interface types, UART, I2C and SPI. Jumpers are provided for setting the interface type. These jumpers are used in conjunction with setting the interface type in the IA61x_config.h.



UART			
TX	X1-1	RX	X1-2
P4	X1-3	P5	X1-4

I2C			
SCL	X2-1	SDA	X2-2
P4	X2-3	P5	X2-4

SPI			
SS	X3-1	MOSI	X3-2
P4	X3-3	P5	X3-4

Figure 1 *IA611 Xplained Pro Interface Jumpers*

The IA611 Xplained Pro board's default configuration is UART. To configure the board for either I2C or SPI, with power disconnected, remove the two jumpers and insert them for the desired interface. Be sure to orient the jumpers as shown.

2.2 Connections

The SAM D21 has only 2 connections. First, plug the IA611 Xplained Pro board into **EXT1** and connect a micro USB cable to the connector labeled **Debug USB**. The USB cable should also be connected to a USB port on the computer you will be using to build and debug the solution.

Place the solution flat on a workspace as shown in the figure that follows.

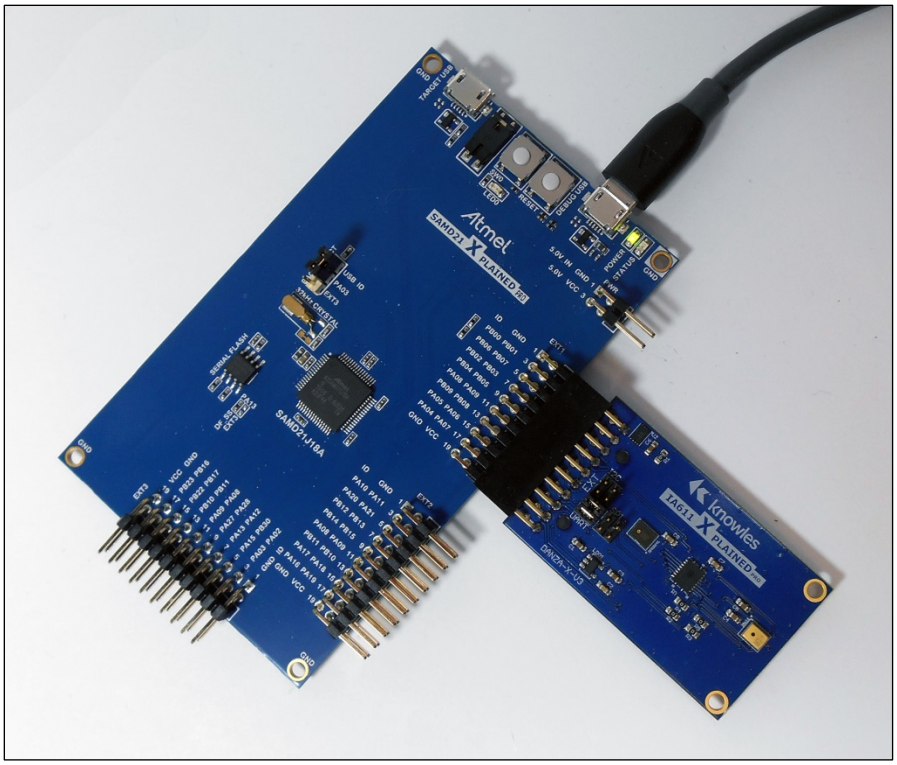


Figure 2 *IA611 Reference Design for IoT Connected to SAM D21 Developer Board*

Chapter 3: IA611 Sample Software Setup

3.1 Sample Project

After obtaining an Extension package of the sample project from Microchip gallery, install the package on your PC. It will install under the default location of the Atmel Studio 7 folder. To verify that IA611 RDI sample extension installed successfully, launch Atmel Studio 7 and then select **Extensions and Updates… from** the **Tools** dropdown. The **IA611-RDI-01 Xplained Pro - Demo** Extension should appear as shown in below.

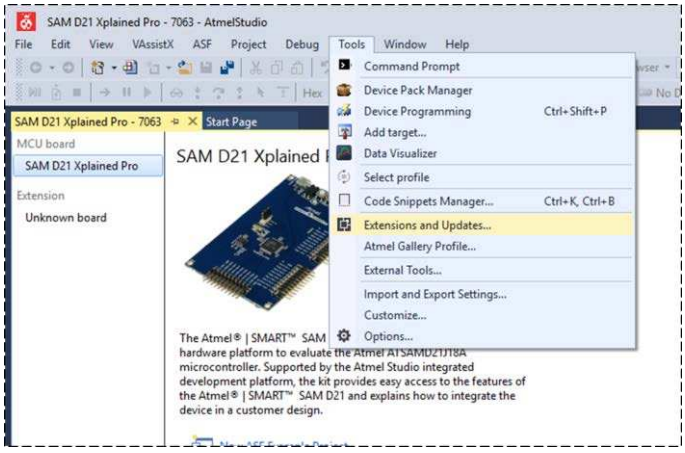


Figure 3 *IA611-RDI-01 Xplained Pro - Demo Extension*

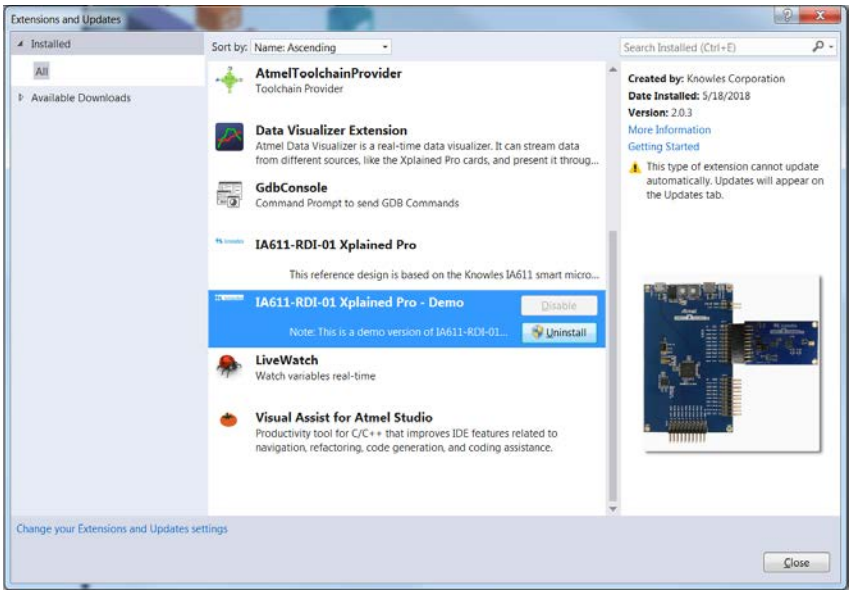


Figure 4 *IA611-RDI-01 Xplained Pro - Demo Extension*

3.2 Creating Example Project from Extension

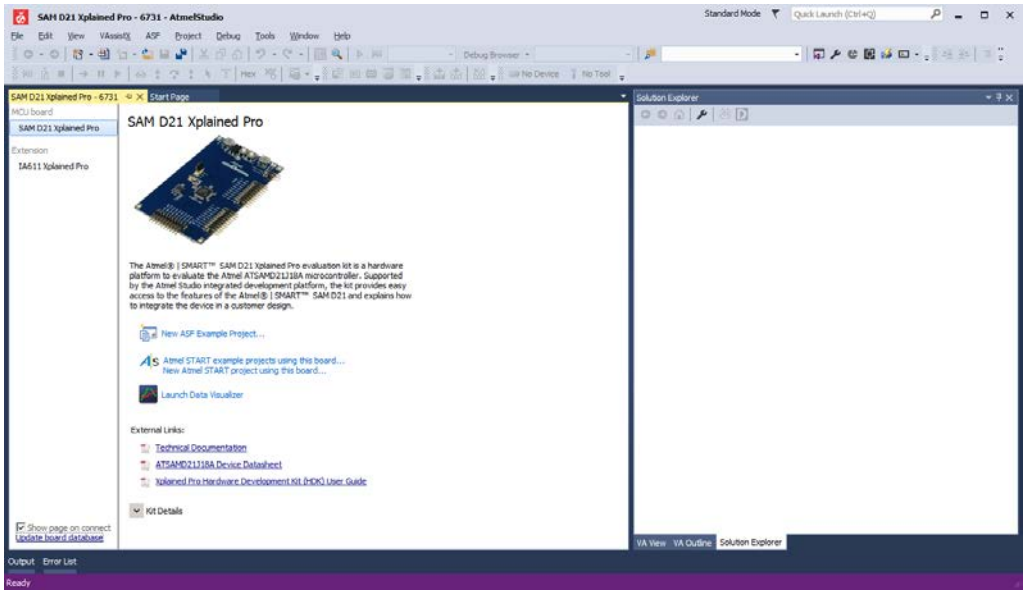


Figure 5 *Studio 7 Welcome Page*

1) Attach the IA611 Reference Design for IoT to SAM D21 Developer Board as shown in Figure 1 and connect a USB cable between board's Debug USB port and the PC.

2) Launch Studio 7 and create the example project by clicking on **File** and selecting **New▶ Example Project** from the dropdown.

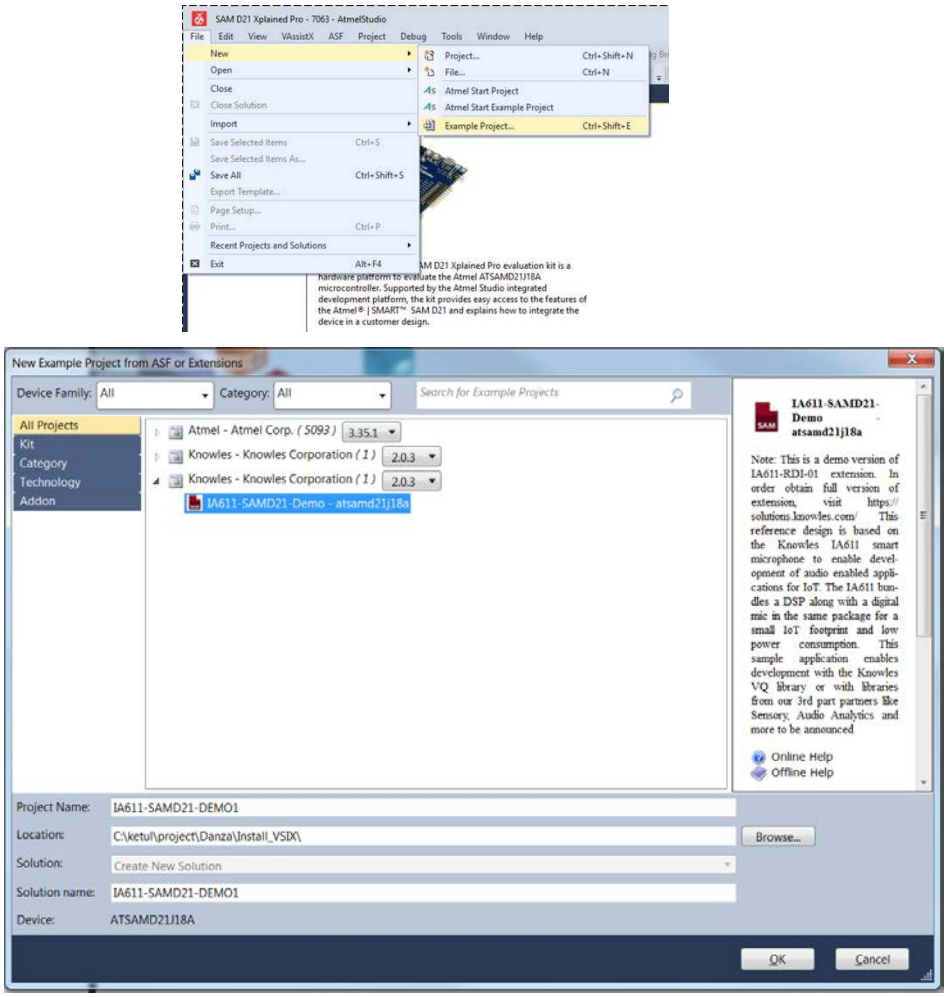


Figure 6 *Create Example Project*

3) Expand the **Knowles – Knowles Corporation** project and select **IA611-SAMD21-Demo – atsamd21j18a**.

   Note *Type **IA611** in search box to locate the sample project if you cannot find the project.*

4) **Project Name**, **Location**, and **Solution Name** will autofill but, if desired, may be changed. Click **OK.**

5) Check the **I accept the license agreement** checkbox and click **Finish**.

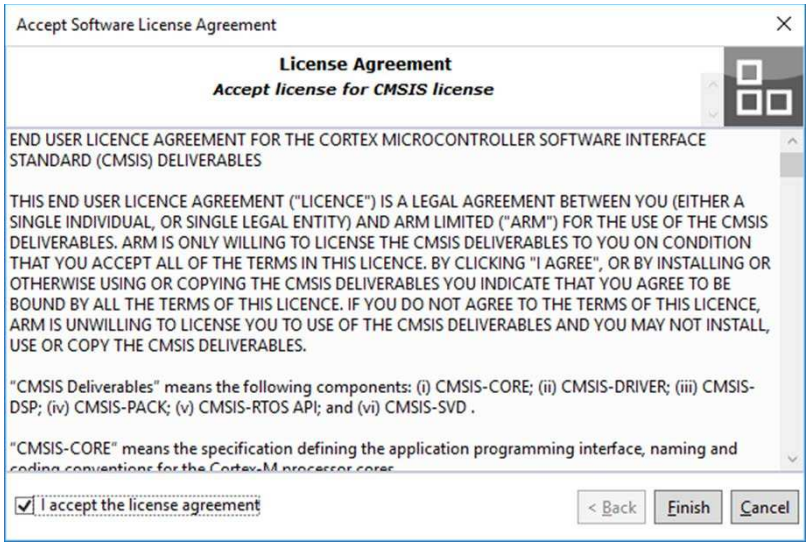


Figure 7 *License Agreement*

Once the project has been created, the project files will appear in the workspace's Solution Explorer window under **IA611-SAMD212**. The directories include

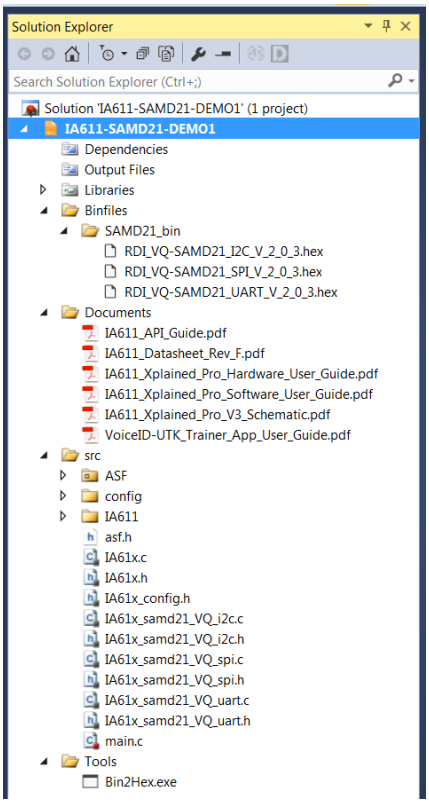


Figure 8 *Solution Explorer*

- **Src**: source code needed to build the solution. Chapter 4 provides an explanation of the files within the src directory. This version of source includes the dummy files for IA611 firmware and keyword models.
- **Binfiles ▶ DAMD21_bin**: Prebuilt executables for SAMD21 for three different IA611 hardware interface configurations (UART, I2C and SPI).
- **Documents**: Software User Guide (this document), Hardware User Guide, IA611 Datasheet, IA611 API Guide, and VoiceID-UTK Trainer App User Guide.
- **Tools**: Bin2Hex utility to convert IA611 bin files to header file (refer to section 4.6)

Chapter 4: IA611-SAMD21-Sample

4.1 Host Software Architecture:

IA611 Xplained Pro Host (SAMD21) software consists of following components:

4.1.1 Host Application

Example code to call sample API's of IA611 interface to initialize IA611 instance, to load firmware, sysconfig and keyword models, setup audio route and application code to handle the keyword detect events.

4.1.2 IA611 Sample API and HW Interface Abstraction Layer

IA611 interface definition and interface specific implementation for UART, SPI and I2C interfaces.

4.1.3 IA611 Components

Included as header (.h) file in the Host code.

- **IA611 Firmware:** IA611 Firmware with Audio algorithm. Different for each hardware interface.
- **IA611 Sysconfig:** IA611 firmware configuration file. Different for each hardware interface
- **Keyword Model files:** Keyword model files for i.e., *Hello VoiceQ*, *Switch the Light*, *Next Song*.

4.1.4 Bin2Hex Utility

Executable that converts the binary files to 16 bit array. In case of change in Config file or keyword models, this utility converts input binary file to header file in 16bit array. This header file then needs to be replaced at appropriate location in IA611 sample project and build.

4.1.5 Voice ID/UTK Trainer Application

Mobile application for Android phone to generate the user trained keywords as well as user activated Voice IDs. Generated model then needs to be copied from mobile

phone to Host PC to be generated to .h file using Bin2Hex utility and then included in sample project.

Refer to *VoiceID-UTK Trainer App User Guide* for more details on how to use this application and generate UTK/Voice ID keywords.

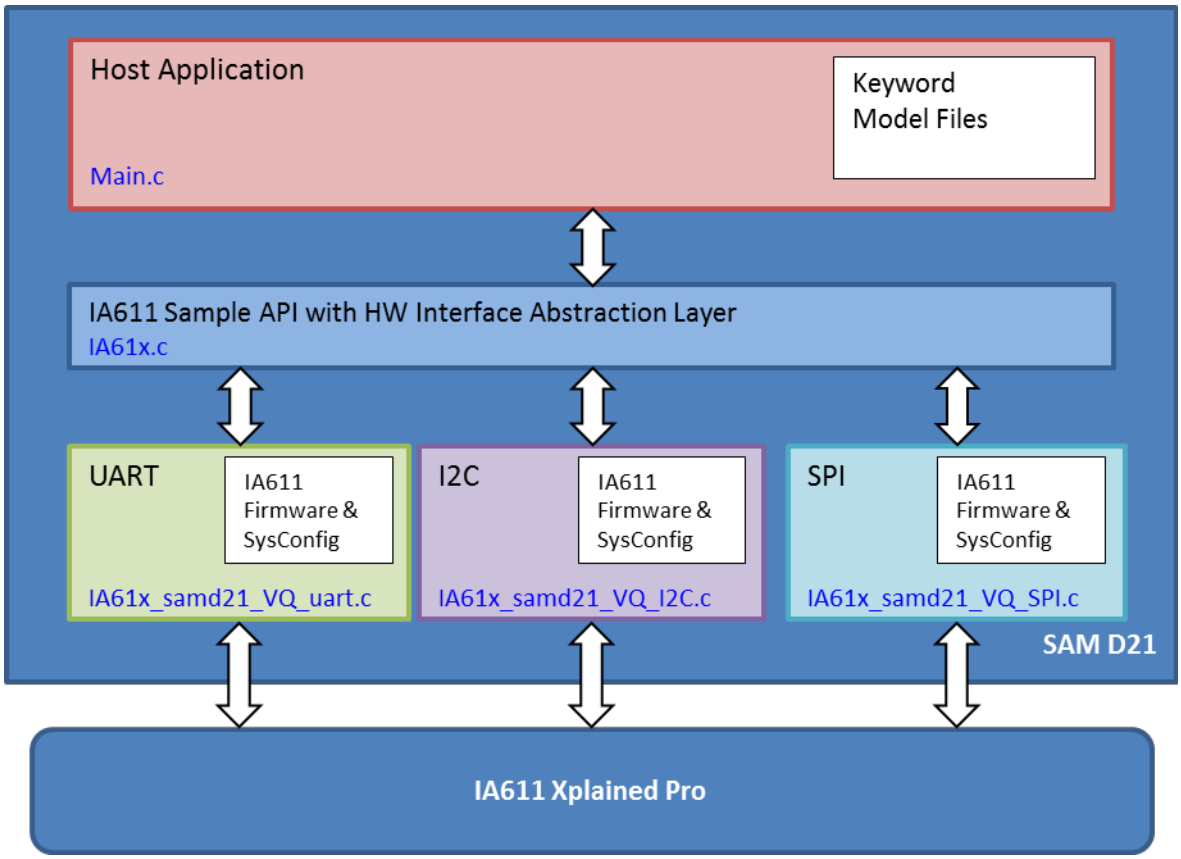


Figure 9 *Host Software Architecture*

4.2 Source Files

4.2.1 src/IA61x_config.h

This file defines the IA611 and Host interface layer to be included in sample project. Enable appropriate #define to select specific interface from UART, SPI or I2C.

`IA61x_SAMD21_VQ_UART`: For UART interface (Default is UART interface)

`IA61x_SAMD21_VQ_I2C`: For I2C interface

`IA61x_SAMD21_VQ_SPI`: For SPI interface

4.2.2 src/IA61x.c, IA61x.h

This file defines abstract API's to access the IA61x interface specific API and common #defines.

IA61x_init: Creates IA61x interface instance to access the IA611 interface API.

IA61x_uninit: Uninitializes IA611 interface

download_config: Downloads Sysconfig file

download_program: Downloads Firmware file

download_keyword: Downloads keyword model. Internally this function takes care of incrementing and adding the keyword ID

VoiceWake: Sets up the usecase specific audio route

wait_keyword: Checks if any keyword detected by IA611

cmd: Sends command to IA611 and get response

put: Sends command to IA611

get: Reads response from IA611

4.2.3 src/main.c

Main host application file consists of sample code to do the following:

- Initialize IA611 interface
- Download firmware/sysconfig/keyword models
- Setup specific audio route
- Wait for keyword detection and application action for specific keyword/command detected.

The sample application demonstrates simple use case of one wake keyword and three command words.

For each keyword/command word detected, LED0 will blink for a number of times equal to keyword ID. LED0 is located on the development board just below the pushbutton switch SW0.

ID	Keyword or Command	LED0 Blinks
1	Hello VoiceQ or UTK	1 time
2	Switch the Light	2 times
3	Next Song	3 times
4	Biadu-Yi Xia	4 times

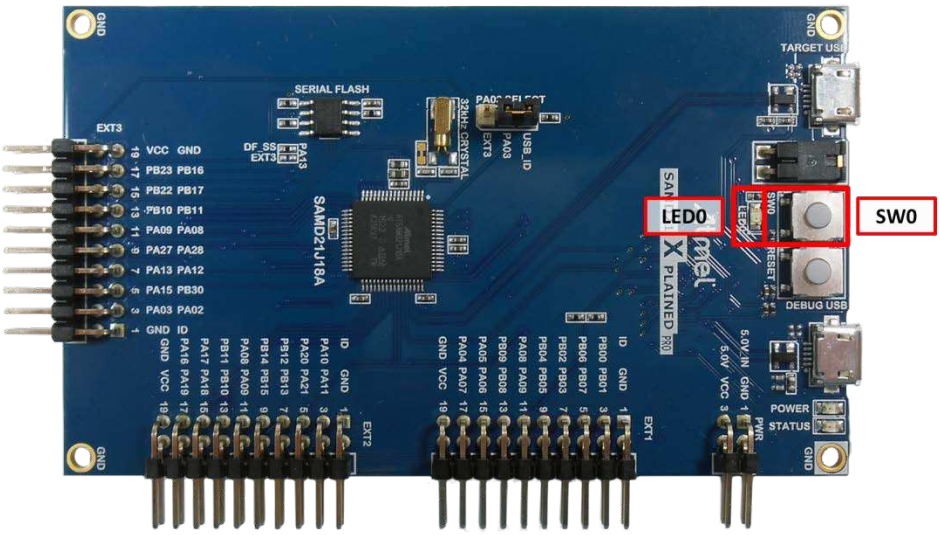


Figure 10 *LED0 on SAM D21 Development Board*

"Main" is the function that is first run after initialization of the hardware is complete. This function will initialize the board, configure LED0, and initialize the Debug UART before beginning to initialize the IA611 microphone. After the MCU is able to communicate with the microphone, two files are downloaded: the system config file, followed by the IA61x firmware binary.

After successfully downloading the two binaries, each of the keywords files are downloaded to the microphone. In the example code, one keyword and three commands are downloaded to the IA611 microphone using the function `IA61x->download_keyword()`.

Once successfully downloading keywords and commands, the IA611 can start processing audio with the `IA61x->open()` function. This will set the audio routes up and put the microphone into a low power voice detection mode.

The main function will now wait for the keyword to be uttered by the user. `IA61x->wait_keyword()` function will return a value of the keyword that was spoken.

Once the microphone has detected the first keyword, it immediately begins looking for a command using the same `IA61x->wait_keyword ()` function.

If a command is not received within a given time (timeout condition), the microphone will reset to the original state, waiting for the first keyword to be spoken.

If the microphone detects a command, the `wait_keyword()` function will return the number corresponding to the command that was downloaded in order.

4.3 src/IA61x_SAM D21_VQ_uart.c

This file contains all the required functions to initialize the UART on the SAM D21 as well as read and write to the UART. It also contains the basic functions to

download code to the IA611, download keywords and commands and initialize the audio route used.

The IA611 Smart Mic on the Xplained Pro Extension board will be always listening for wake words and commands. Upon detection of the correct phrases, the IA611 will send an interrupt to the MCU using P5 / UART_Tx (Pin 15 TX on the Level Shifter) as IRQ, interrupt to host to signal a wake word detection. Current example implements Host UART interface is implemented as polling mode hence it will detect the IA611 interrupt as some data on UART_Rx line and check for the event detection.

4.4 src/IA61x_SAM D21_VQ_i2c.c

This file contains all the required functions to initialize the I2C on the SAM D21 as well as read and write to the I2C. It also contains the basic functions to download code to the IA611, download keywords and commands and initialize the audio route used.

The IA611 Smart Mic on the Xplained Pro Extension board will be always listening for wake words and commands. Upon detection of correct phrases, the IA611 will send an interrupt to the MCU using P3 / I2C_addr2/irq (Pin 17 MISO on IA611 Xplained Pro connector) as IRQ, interrupt to host to signal a wake word detection. The MCU will then take actions according to interrupt information received.

4.5 src/IA61x_SAM D21_VQ_spi.c

This file contains all the required functions to initialize the SPI on the SAM D21 as well as read and write to the SPI. It also contains the basic functions to download code to the IA611, download keywords and commands and initialize the audio route used.

The IA611 Smart Mic on the Xplained Pro Extension board will be always listening for wake words and commands. Upon detection of correct phrases, the IA611 will send an interrupt to the MCU using P1 / pdm_sdo/irq (Pin 7 P1 on IA611 Xplained Pro connector)) as IRQ, interrupt to host to signal a wake word detection. The MCU will then take actions according to interrupt information received.

4.6 Project Configurations and Flashing Instructions

This version of the IA611 Xplained Pro software provides pre built three example samples based on Knowles VoiceQ algorithm.

Voice Q Example: 1 OEM Wake Keyword + 3 Commands for 3 different Host interfaces.

1. RDI_VQ-SAMD21_UART_V_2_0_3.hex ➔ UART Interface
2. RDI_VQ-SAMD21_I2C_V_2_0_3.hex ➔ I2C Interface
3. RDI_VQ-SAMD21_SPI_V_2_0_3.hex ➔ SPI Interface

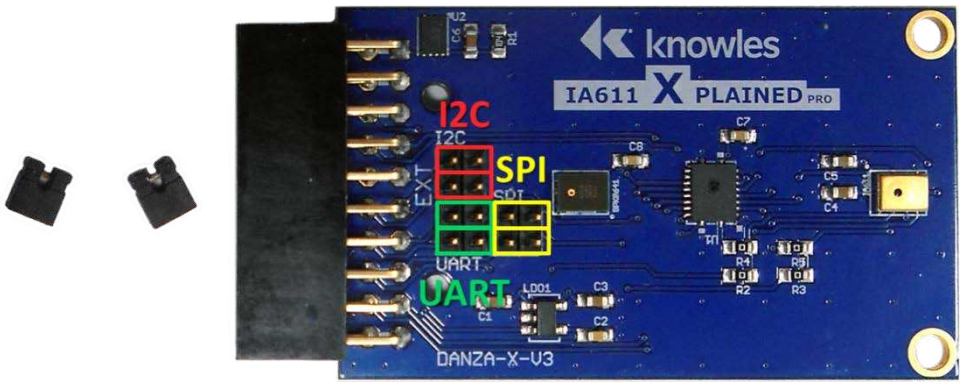


Figure 11 *UART, I2C and SPI Interface Jumper Positions*

Note *By default the UART interface is enabled. Only enable one interface of the three options at a time. To use I2C or SPI, attach the jumper in the orientation shown.*

The default bin files are built with configuration for the VoiceQ example (1 OEM Wake Keyword + 3 commands). To flash one of the examples bin files follow instructions below.

4.6.1 Flash VoiceQ Example

1) Launch Atmel Studio 7 and select **File->Open->Project/Solution…**
2) Locate and open the **IA611-SAMD21.atsln** solution.
3) Make sure that your HW setup is configured correctly as described in Chapter 2.
4) Go to **Tools ➔ Device Programming** and it will open Device programming dialog box.
5) Select the listed Hardware from **Tools** dropdown menu. If you have more than one SAMD21 boards connected, then select the correct hardware based on the serial number you intend to flash.

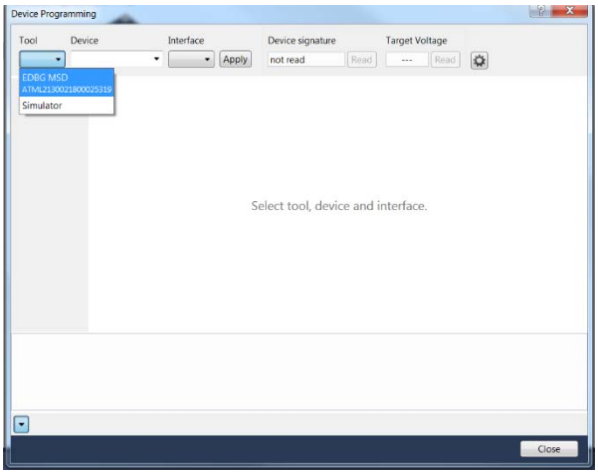


Figure 12 *Select Device for programming*

6) Click on "**Apply**" to apply settings.

7) Click on "**Read**" to read device signature to confirm that Studio 7 is able to communicate with HW.

8) Click on "**Memories**" to open device flashing page. Select "RDI_VQ-SAMD21_UART_V_2_0_3.hex" file from the project folder \Binfiles\SAMD21_bin. Click on Program to start flashing.

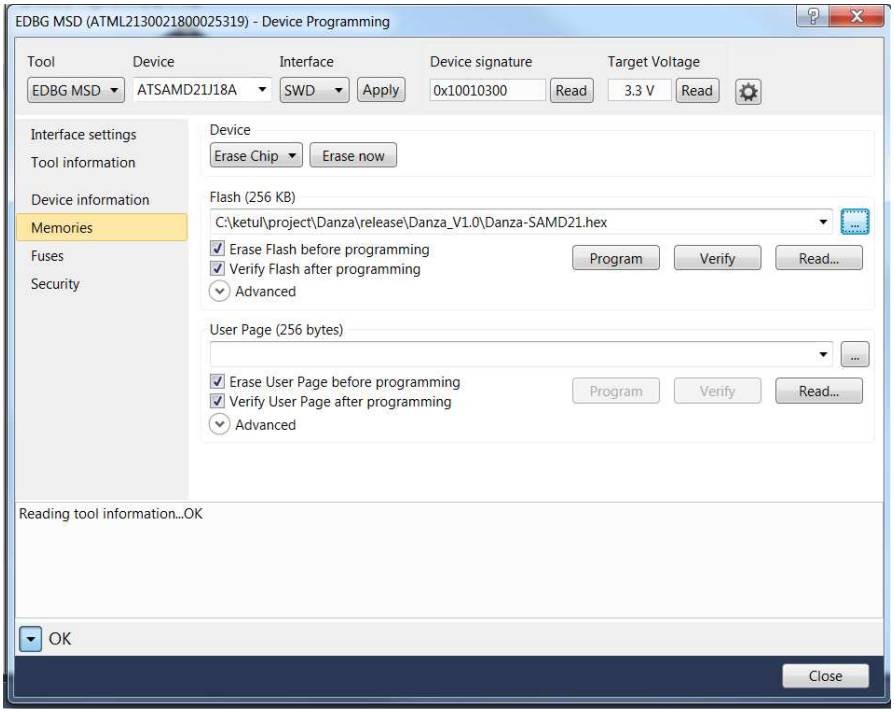


Figure 13 *Select .hex file and program*

9) Once Flashing is over, review the message at bottom of the dialog box to confirm that flashing is successful.

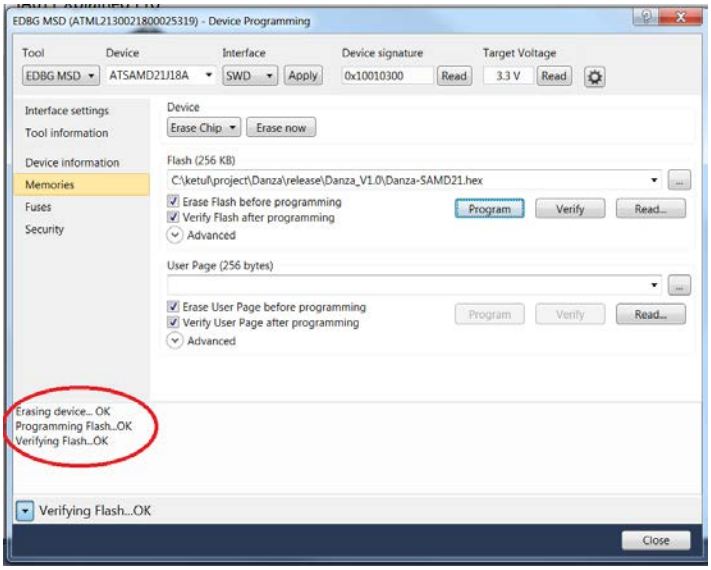


Figure 14 *Flash .hex file to device*

10) Wait for SAMD21 to boot. Once booted successfully, LED 0 on SAMD21 board will blink 4 times.

11) Say the wakeup keyword phrase "Hello VoiceQ" followed by one of the command phrases below and observe the state of LED0. Alternately, use a terminal program that supports DTR flow control and the settings shown to monitor the SAM D21's EDBG Virtual COM Port.

NOTE *If the command phrase is not said within six seconds, the keyword will time out as indicated by LED0 blinking six times and on the terminal.*

WakeUp Keyword	LED0 Blinks	Command	LED0 Blinks
Hello VoiceQ	1 time	Switch the Light	2 times
		Next Song	3 times
		Biadu-Yi Xia	4 times

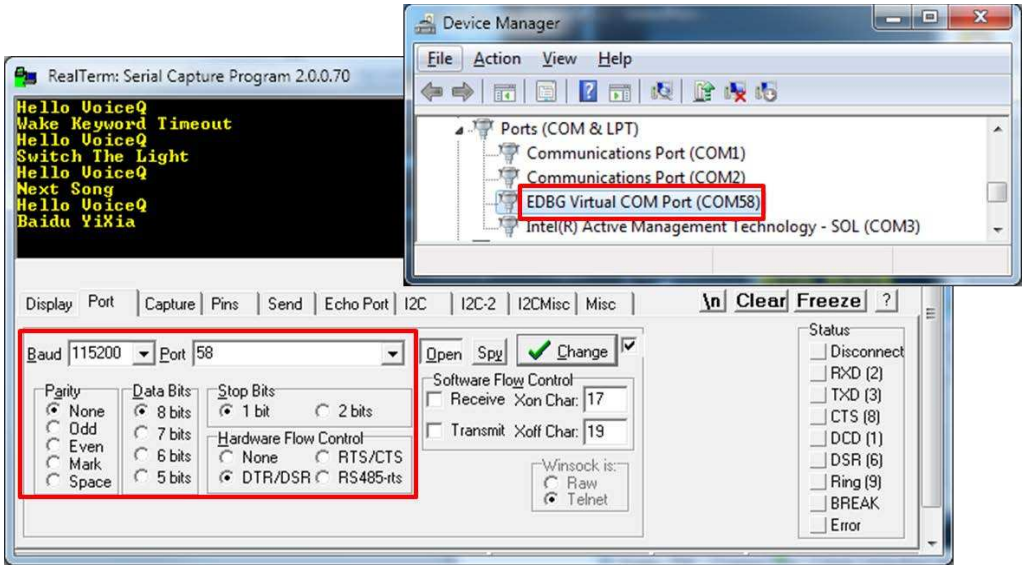


Figure 15 *Terminal Setup and Output*

12) Repeat the keyword plus command for the other command phrases.

Revision History

The following lists the changes made since the previous version of this document.

Version	Revision Date	Author(s)	Comments
1.0	05/18/2018	Ketul	Initial Release

Knowles Intelligent Audio
331 Fairchild Drive
Mountain View, California 94043
USA
Tel: +1 650.254.2800
Fax: +1 650.254.1440

www.knowles.com

Rev. 5/21/18