
# Conexant AudioSmart™ 4-Mic Development Kit for Amazon AVS

## User Guide

CONEXANT®

EXPERIENCE SMART™

www.conexant.com

005UGR01

# Revision History

| Document No. | Release Date | Change Description |
|---|---|---|
| 005UGR01 | 09/19/17 | **Updated:**<br>• Step 2 in "Writing Raspbian OS to the microSD Card" on page 6. |
| 005UGR00 | 05/03/17 | Initial release |

# Table of Contents

# List of Figures

# Scope

This documentation provides step-by-step instructions on setting up the **Conexant AudioSmart™ 4-Mic Development Kit for Amazon AVS**. The document covers how to make the necessary hardware connections, install the driver, flash the firmware (only when needed), configure the Raspberry Pi2 (RPi 2), and set up the Amazon Alexa Voice Service (AVS).

***Note:*** Only use the RPi 2 with this configuration. Do not use the Raspberry Pi 3 (RPi 3).

# Target Audience

This document is intended for manufacturers and developers for creating Smart Home device prototypes that utilize the **Conexant AudioSmart 4-Mic Development Kit and the RPi 2 (not provided with the kit)** to offer an ideal Voice Control experience via Amazon's Alexa Voice Service.

# References

| Subject | Description | Location |
|---|---|---|
| $I^2C$ | $I^2C$ Specification | http://www.nxp.com/documents/user_manual/UM10204.pdf |
| ACPI | Advanced Configuration and Power Interface | http://www.acpi.info/DOWNLOADS/ACPIspec50.pdf |
| RPi 2 | An open source code single-board computer | https://www.raspberrypi.org/products/raspberry-pi-2-model-b/ |
| ALSA | The Advanced Linux Sound Architecture (ALSA) provides audio and MIDI functionality to the Linux operating system. | http://www.alsa-project.org/main/index.php/Main_Page |
| ASoC | ALSA system on Chip for $I^2S$ codecs | http://www.alsa-project.org/main/index.php/ASoC |
| RPi 2 | Kernel building | https://www.raspberrypi.org/documentation/linux/kernel/building.md |

# Definitions, Acronyms, and Abbreviations

| Name of Document | Description |
|---|---|
| AVS | Alexa Voice Service |
| DSDT | Differentiated System Description Table |
| GPIO | General-Purpose Input/Output |
| $I^2C$ | Inter-Integrated Circuit |
| RPi 2 | Raspberry Pi 2 |

# Overview

The **Conexant AudioSmart 4-Mic Development Kit for Amazon AVS** contains the following:

- CX20924 EVK
- Microphone/LED module (positioned on top of the CX20924 when shipped)
- USB cable
- Cable assembly (colored wires)
- +5V power supply

**Note:**

- **Micro SD card of at least 16GB is required.**
- **Powered speakers, RPi 2, and micro SD card are all mandatory for the set up but are NOT included in the Conexant AudioSmart 4-Mic Development Kit for Amazon AVS.**

Proper speaker selection will enhance overall performance. Download *External Loudspeaker Guidelines and Recommendation for Smart Speaker Applications* (008DGR0x) document for additional info.

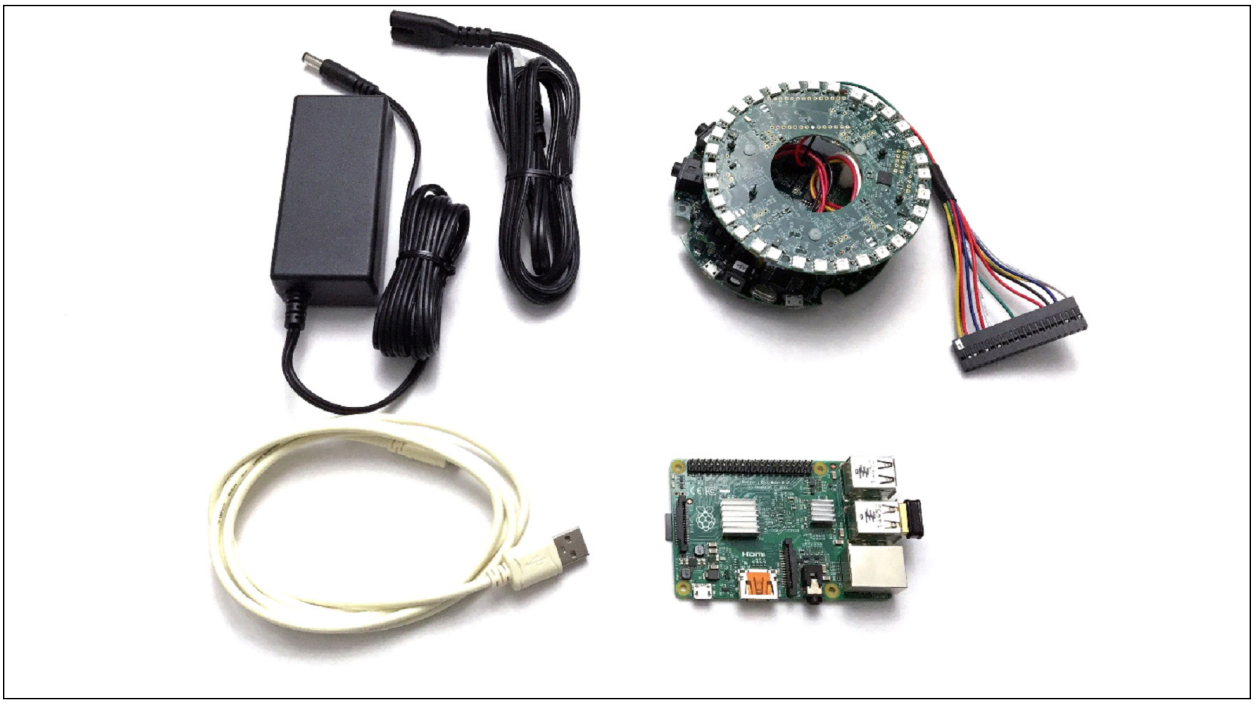


**Figure 1: Development Kit Components**

# CX20924 EVK

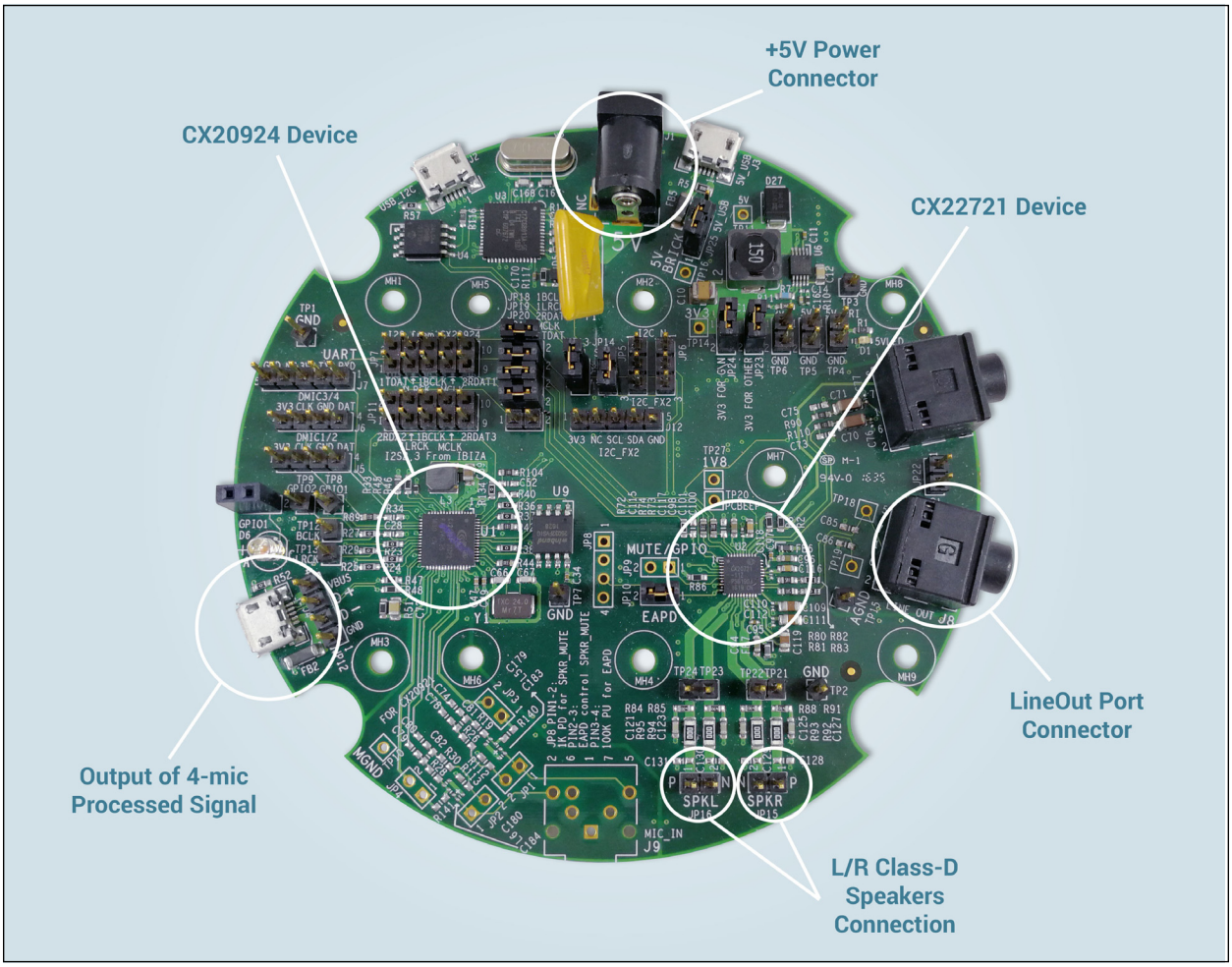



**Figure 2: CX20924 EVK: Connections, Interfaces, and Devices**

# Microphone/LED Module

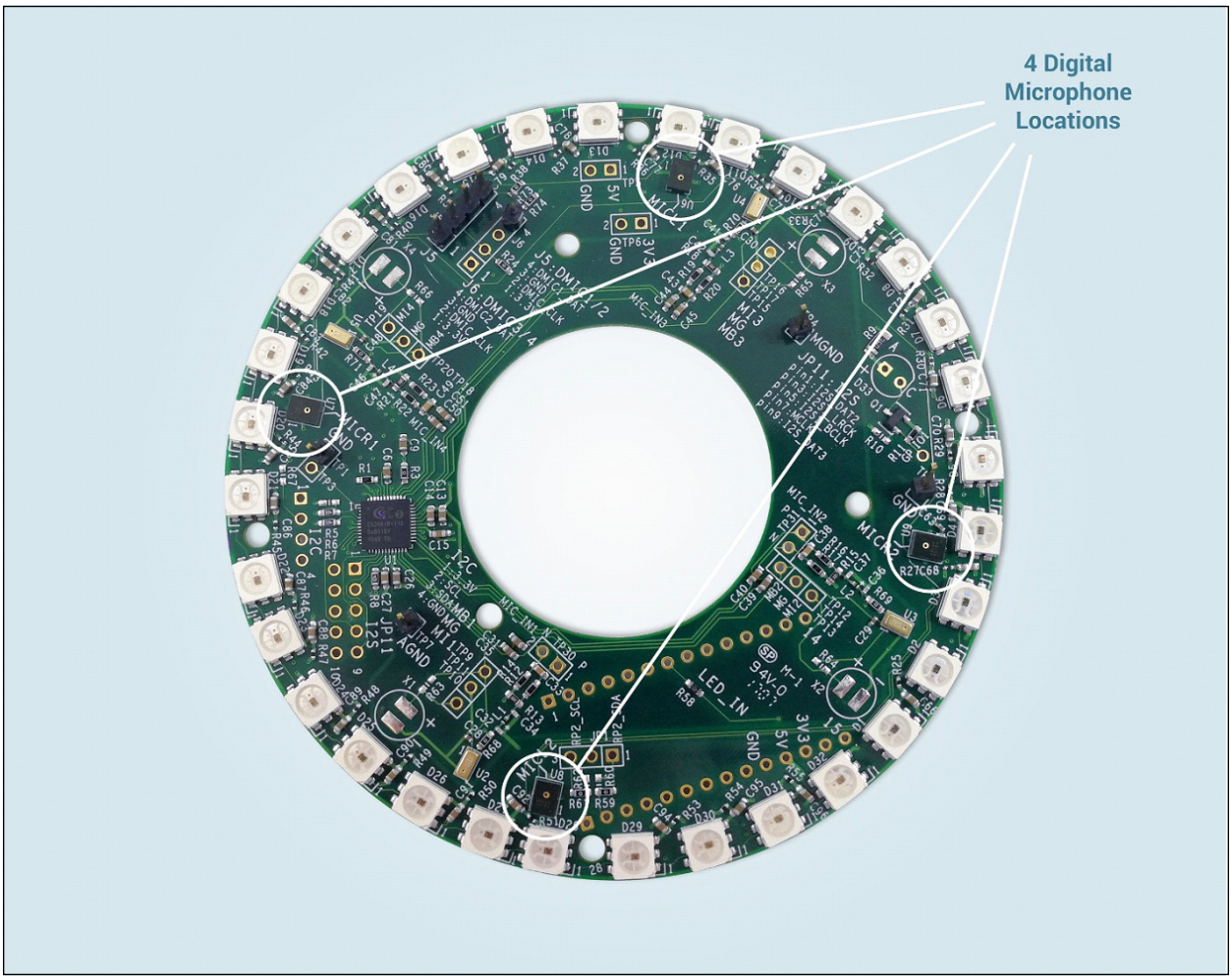


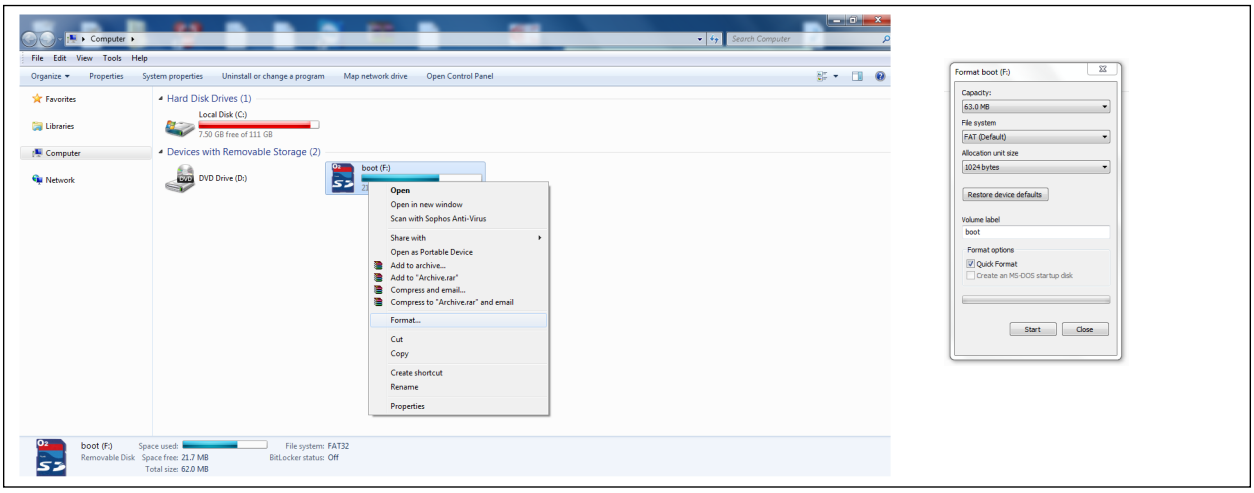

**Figure 3: Microphone/LED Module: Four Digital Microphone Locations**

# Step-by-Step Setup Summary

1. Write a Raspbian Image to the micro SD. Load the current Raspbian OS using Linux kernel version 4.4. See "Writing Raspbian OS to the microSD Card" on page 6.
2. Connections between the CX20924 EVK and the Microphone / LED module are already established. To set up the hardware connection between these two boards and the RPi 2 (not provided with the kit) refer to "Connecting the EVK and Microphone/LED Module to the RPi 2" on page 7.
3. Set up the AVS. See "AVS Setup" on page 10.
4. Build and install the Linux kernel. See "Building and Installing the Linux Kernel" on page 14.
5. Once the hardware and software setup is complete, refer to the following to run the 4-mic setup: "Running LED Server, Node.js Service, Sample App, Wake Word Engine, and Recording Agent" on page 18.
6. Refer to the following to connect to an Amazon account: "AVS Setup: Amazon Account Login" on page 20.
7. Verify the setup. See "Verifying the Setup" on page 23.
8. Conduct Cypress Siena USB-to-I2C device driver installation if required. This step is optional. See: "Installing the Cypress Siena USB-to-I2C Device Driver" on page 24.
9. Flashing New Firmware. This step is optional. Upgrade the firmware for the CX20924 device. Note that the CX20924 evaluation board is pre-flashed with the firmware required so this step is only necessary if an updated CX20924 firmware is provided. See "Flashing New Firmware" on page 26.

# Writing Raspbian OS to the microSD Card

1. Format the SD card. This can be done by going to the 'Computer' folder and right clicking on the SD card. Click **Format…** to display the format options. When reformatting, confirm that the File system is FAT32 or FAT, as shown below.

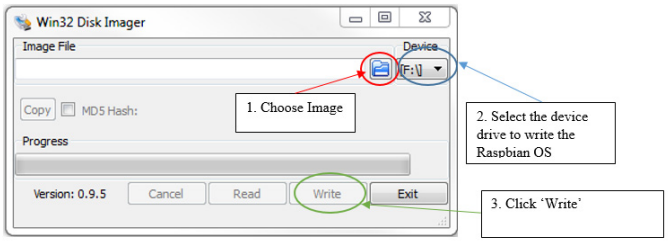



2. Download the Raspbian Jessie OS. The zip file is located at the following link:

   http://downloads.raspberrypi.org/raspbian/images/raspbian-2017-07-05/

3. Unzip and write the downloaded image to the SD card using Win32DiskImager. This tool can be found at the following link: https://sourceforge.net/projects/win32diskimager/

4. Run the Win32DiskImager to write the image to the SD card.




   - Browse to the subdirectory icon and select the image file.
   - Select the device drive in the **Device** drop down menu to write the Raspbian OS.
   - Click **Write** to write the image to the SD card.

5. After the image has finished writing, insert the microSD card into the RPi 2.

# Connecting the EVK and Microphone/LED Module to the RPi 2

Make the following hardware connections between the CX20924 EVK, the Microphone/LED module, and the RPi 2 (connections between the CX20924 EVK, Microphone/LED module and cable assembly are already established). Make sure the microSD card with the Raspbian image and AVS sample app is inserted into the RPi microSD card slot.

1. Connect the cable assembly to the GPIO pins on the RPi 2. Align the sticker labeled '1' with physical pin 1 on the RPi.

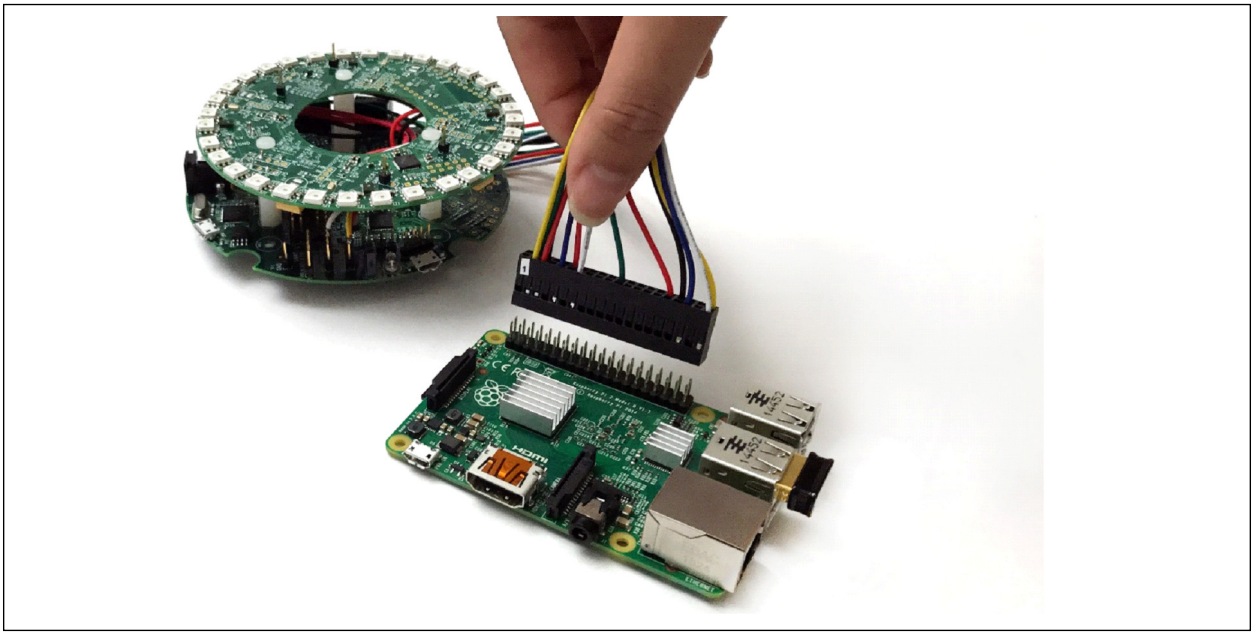


**Figure 4: Connecting Cable Assembly to RPi GPIO Pins**

2. Connect the output of the 4-mic processed signal.
   a. Use the USB cable to connect J4 on the CX20924 EVK to a USB port on the RPi.

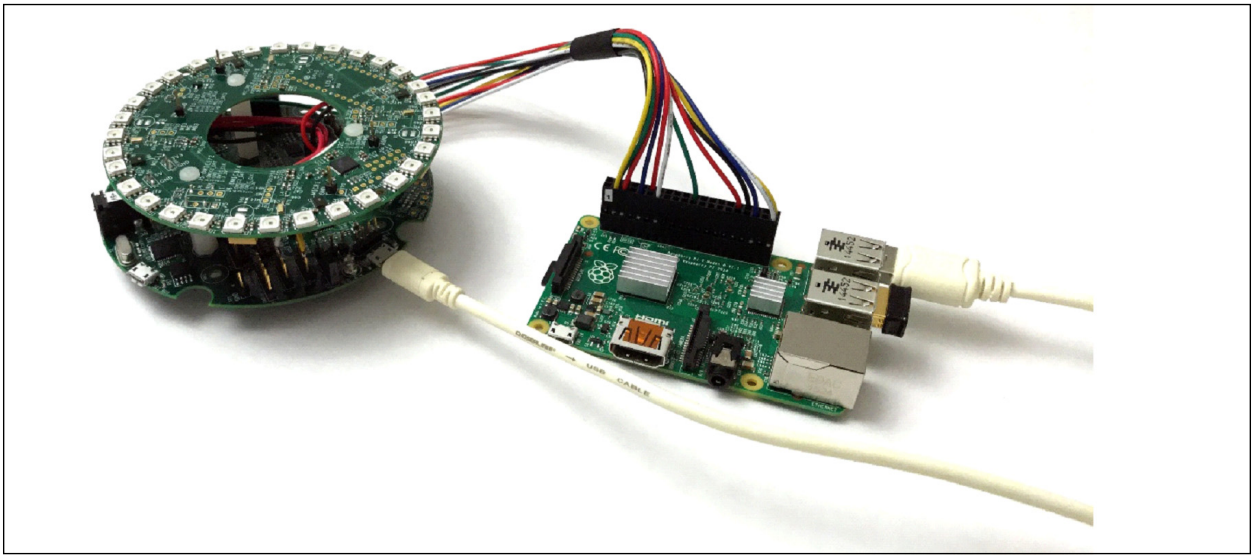


**Figure 5: Connecting Output of 4-Mic Processed Signal**

3. Connect the powered speakers to the LINEOUT port.
   a. Connect the 3.5mm jack from the powered speakers to J8 (labeled LINEOUT) on the CX20924 EVK.

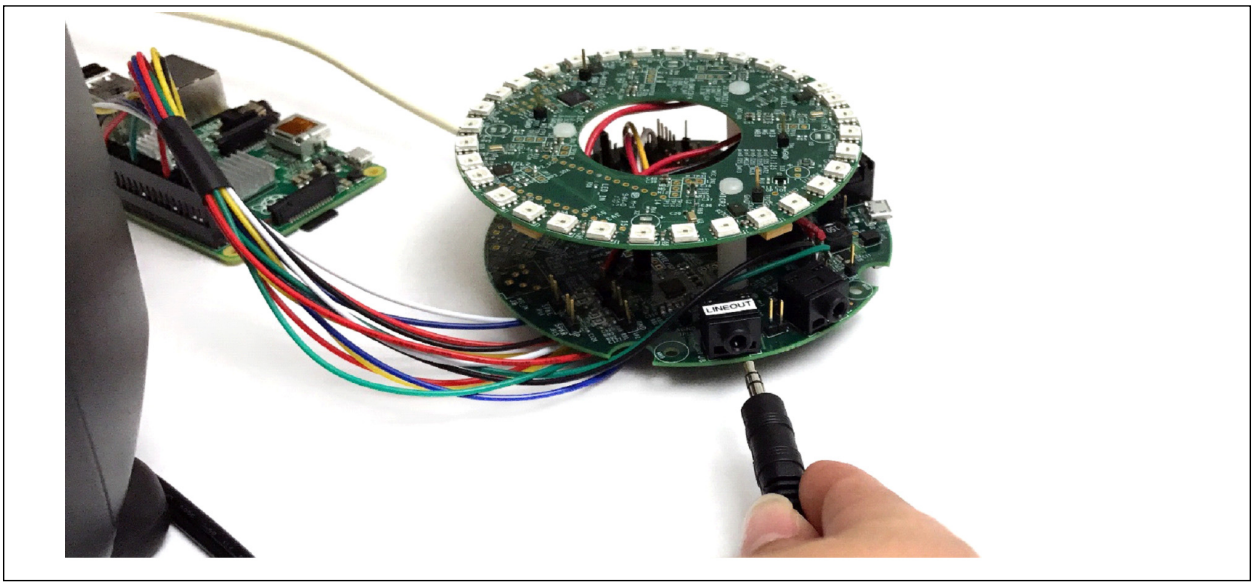



**Figure 6: Connecting Powered Speakers to the LINEOUT Port**

4. Using the HDMI and USB ports on the RPi, connect the RPi to the monitor, keyboard, and mouse.
5. Connect the 5V supply to the J1 connector on the CX20924 EVK. Use the +5V power supply included with the kit to power the CX20924 EVK, Microphone/LED module, and the RPi 2.

**Note: Power to the RPi 2 is provided from the CX20924 EVK and the Micro-USB power on the RPi 2 is not used.**

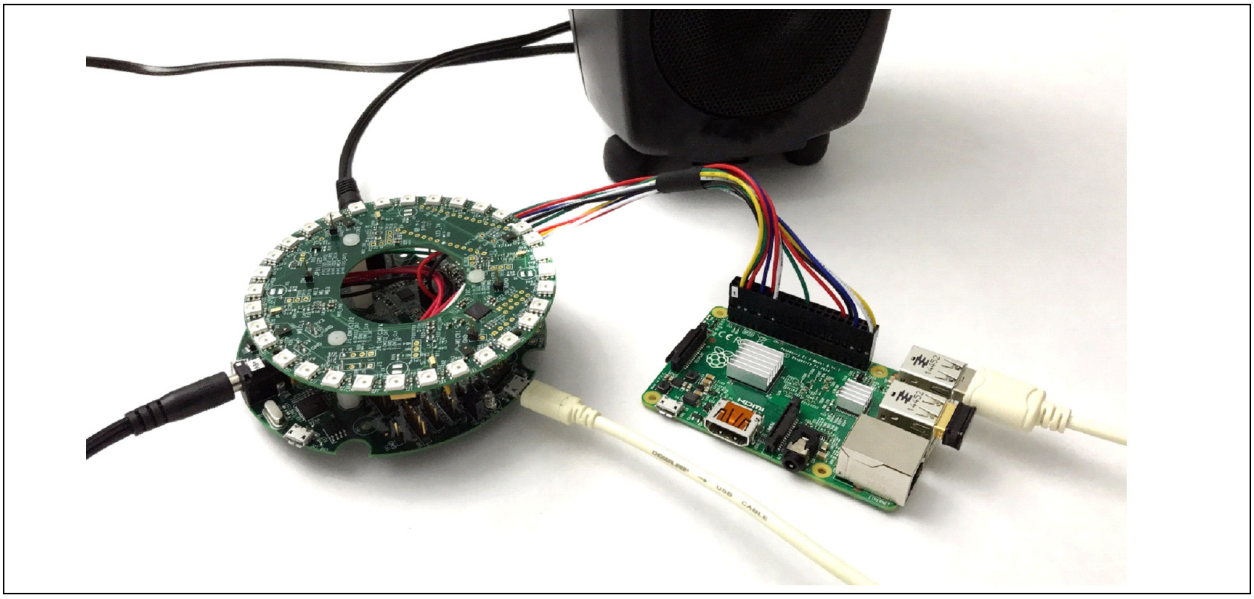


**Figure 7: Connecting 5V Power Supply to J1 Connector on the EVK**

The unit should be mounted on top of the speakers, separated by a cushioned material such as putty or foam, to minimize vibrations from the speaker.

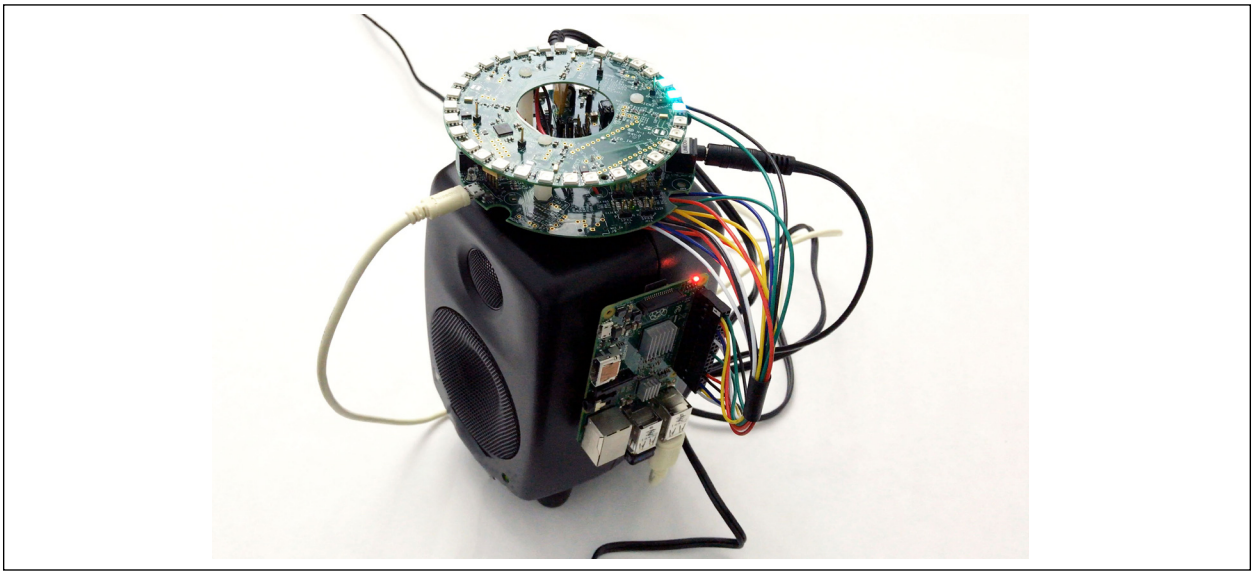


**Figure 8: DS20924 Mounted on Speaker**

# AVS Setup

Prior to completing the steps described in this section, the following tasks should be completed:

- Raspbian OS loaded onto the microSD card
- HDMI monitor, keyboard, and mouse are connected
- LAN or WiFi connection has been established
- Hardware setup complete

1. Send the following command to clone the repository from the Conexant GitHub:

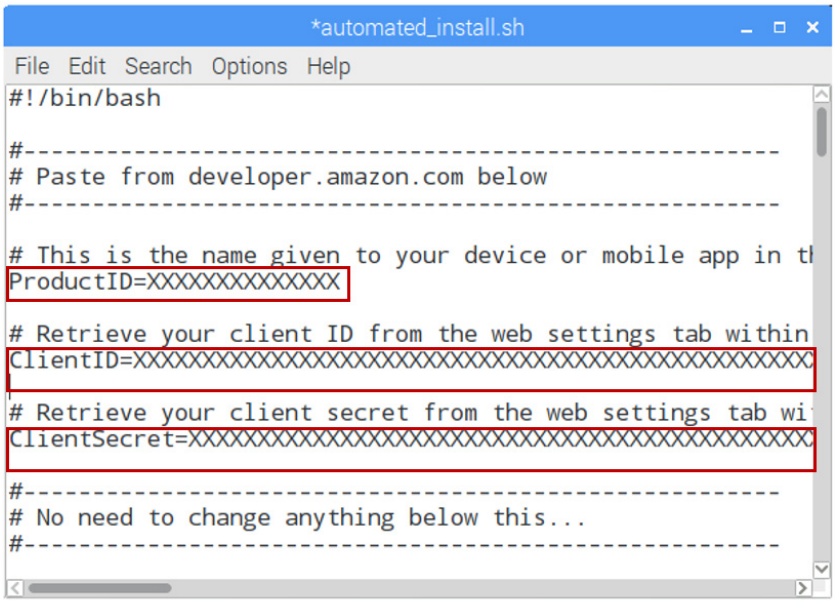
   ```
   $ git clone https://github.com/conexant/alexa-avs-sample-app.git
   ```

2. Refer to the following instructions to create a developer account with Amazon:

   https://github.com/alexa/alexa-avs-sample-app/wiki/Create-Security-Profile

   Before starting AVS installation, the automated_install.sh will require several parameters to be changed. The following credentials will need to be copied from the developer account (developer.amazon.com)

   ***Note:*** These parameters can be found under Alexa -> Alexa Voice Service -> Edit, on the developer.amazon.com site. The Device Type ID can be found in the Device Type Info and the Client ID and Client Secret can be found in the Security Profile tab. A representative example is shown below.



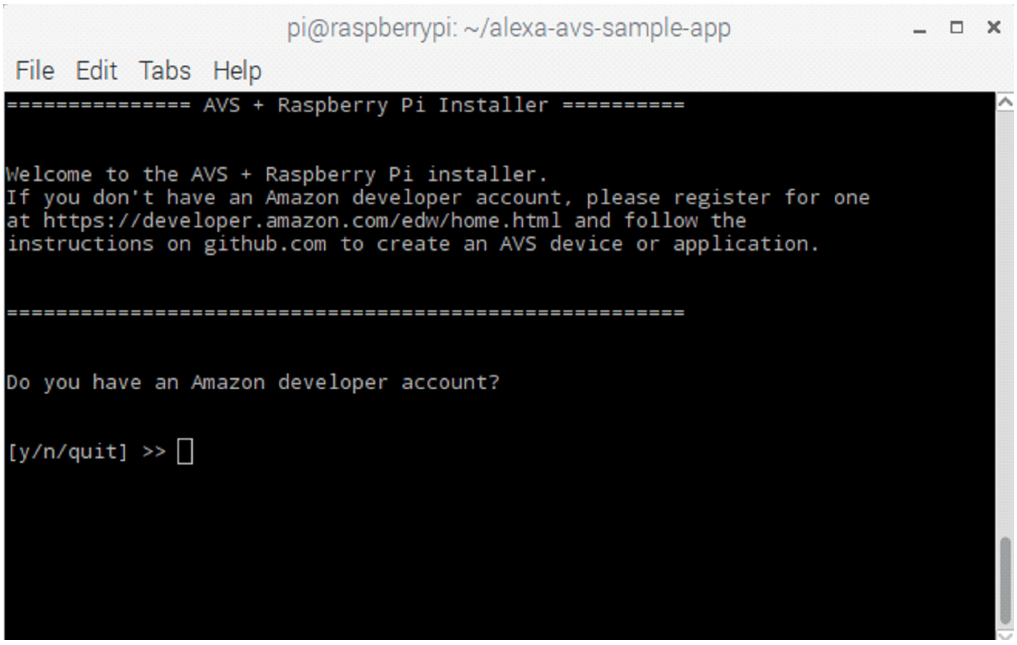

3. Save the file. Then give the necessary file permissions and run the process with the following command:

   ```
   chmod +x automated_install.sh
   . automated_install.sh
   ```

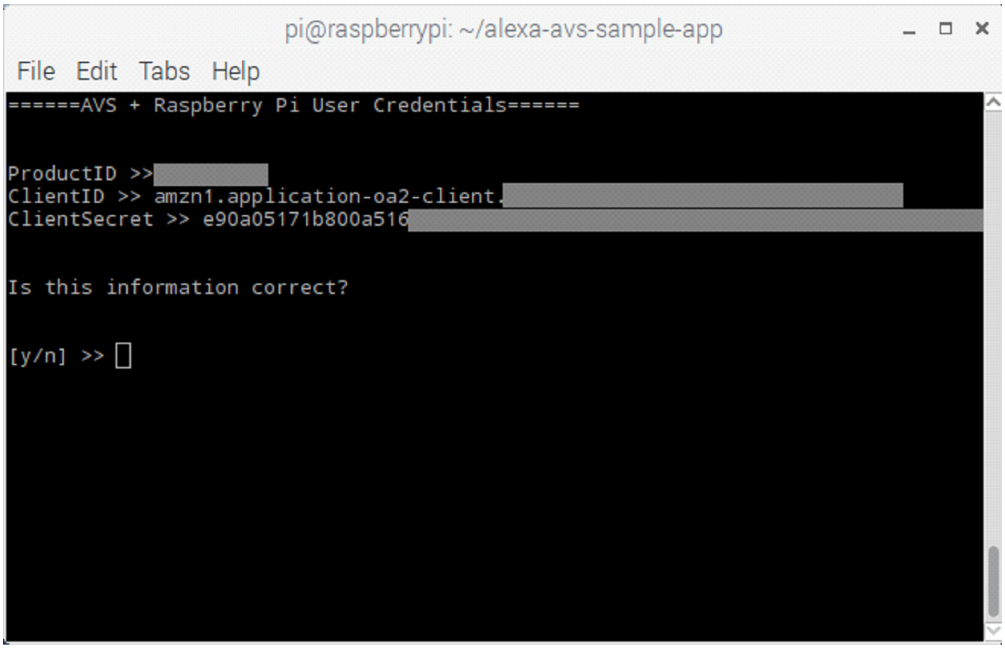
   After the installer is started, several screens will display. Follow the screen instructions shown below to download the AVS sample app.

This requires agreeing to the terms and conditions as shown in the following screen:

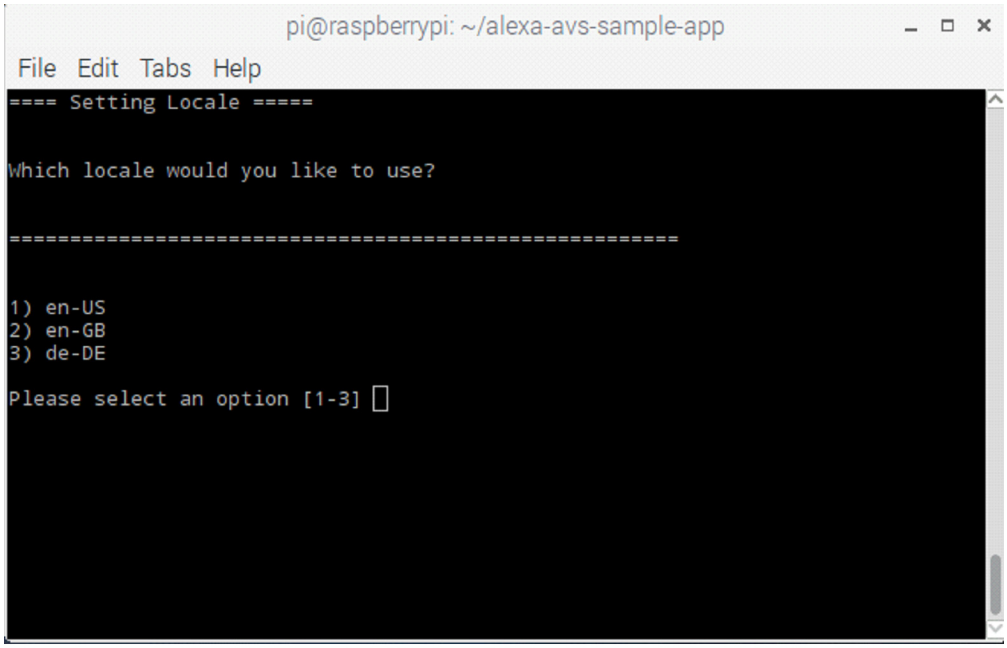

4. Type **y** and press **Enter**. The following screen enables you to validate your Amazon developer status.



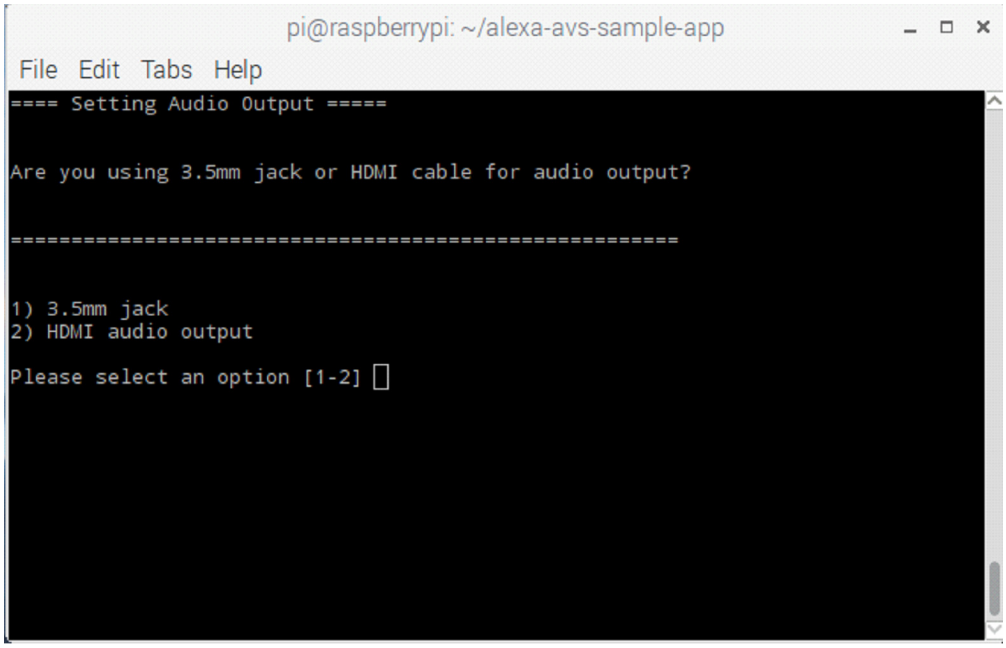

5. Type **y** and press **Enter**. The following screen enables you to set up your RPi 2 credentials, enter your **ProductID**, **ClientID**, and **ClientSecret**.



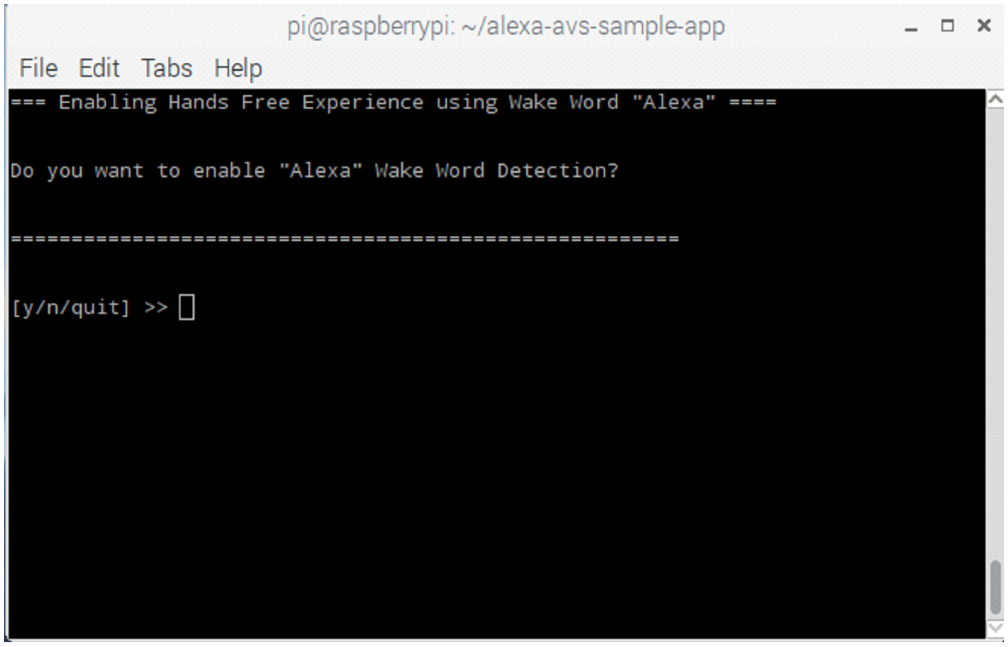

6. Type **y** and press **Enter**. The following screen enables you to set up your locale based on language and country. For example, en (English)-US (USA), en (English)-GB (Great Britain). Locales are a framework to switch between multiple languages and allow users to use their language, country, characters, collation order, etc.




7. Type **1**, **2**, or **3** depending on your language and locale, and press **Enter**. The following screen enables you to select the method of audio output you are using.




8. For audio output, select option **1**. By default, the $I^2S$ interface is used. Thus, the HDMI audio output selection has no impact.




9. Type **y** and press **Enter**. The following screen enables you to enable Alexa Wake Word Detection.

   This starts the installation process. The installer should take approximately one hour to complete.

# Building and Installing the Linux Kernel

Prior to completing the steps described in this section, the following tasks should be completed:

- Before installing the kernel, the microSD card should have completed the AVS setup.

Steps 1-9 must be completed on the host machine (which is running the Ubuntu environment and which the kernel will be built on). Steps 10 and 11 will be completed on the RPi 2.

1. Build and install the Linux kernel for the RPi 2. The following instructions refer to cross compiling the driver on an Ubuntu environment.
   - Prior to building the Linux driver, the following packages must be installed: Git, bc, libncurses5-dev, libncursesw5-dev, and gcc-arm-linux-gnueabihf.
   - All these packages can be installed with the following command:

     ```
     $ sudo apt-get install git bc libncurses5-dev libncursesw5-dev gcc-arm-linux-gnueabihf
     ```

2. The following instructions create a 4.4.50 kernel running on the RPi 2. If you don't have the toolchain installed, use the following command to install the toolchain:

   ```
   $ git clone https://github.com/raspberrypi/tools
   ```

   ***Note:*** For convenience, the sub-directory, "bin", can be added to $PATH

   ```
   $ PATH=$PATH:~/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/bin
   ```

3. Clone the following RPI kernel tree:

   ```
   $ git clone --depth 10 -n --branch cnxt-rpi-4.4.y --single-branch https://github.com/conexant/rpi-linux.git
   ```

   ```
   $ KERNEL=kernel7
   ```

   ```
   $ cd rpi-linux
   ```

   ```
   $ git checkout
   ```

4. Build sources

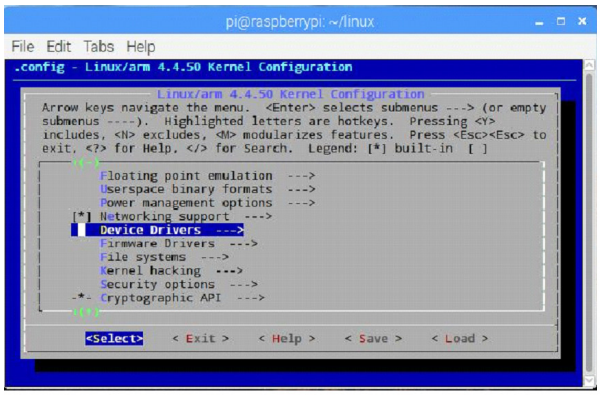
   ```
   $ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- bcm2709_defconfig
   ```

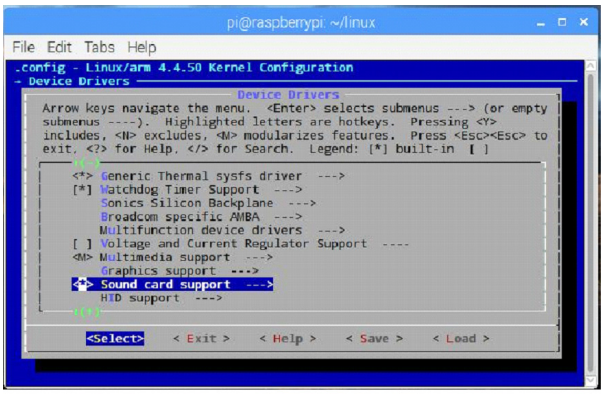
5. Enable the driver options

   ```
   $ make ARCH=arm menuconfig
   ```

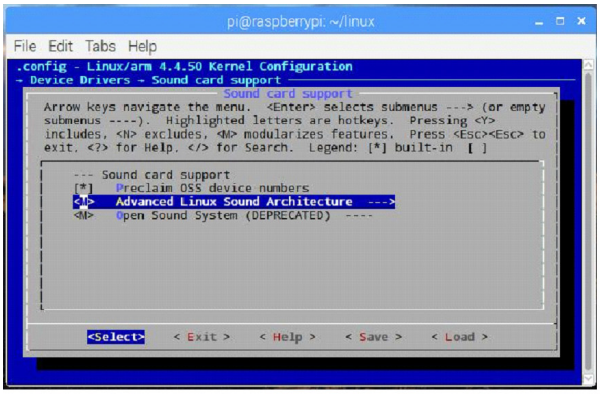
6. In the Linux/arm 4.4.50 Kernel Configuration menu, go to: [Device Drivers] => [Sound card support] => [Advanced Linux Sound Architecture] => [ALSA for SoC audio support]. Check the following two boxes:
   - [Support for Smart Speaker Pi add on soundcard (USB)]
   - [Support for Smart Speaker Pi add on soundcard (I2S)]

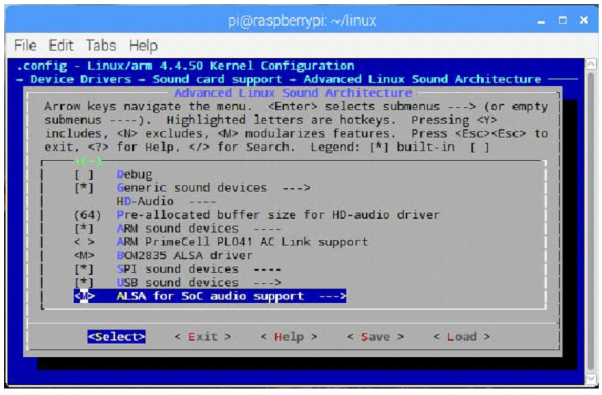
The following screens provide references:



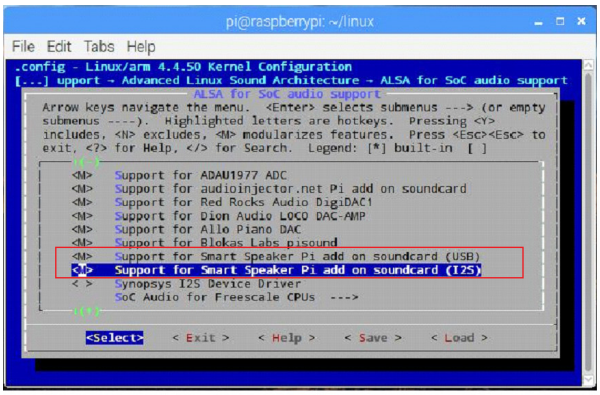



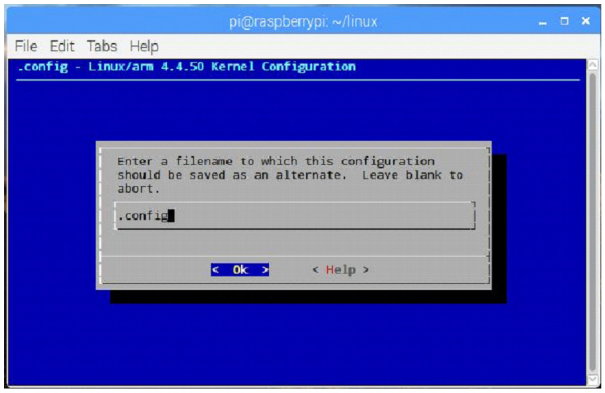

:



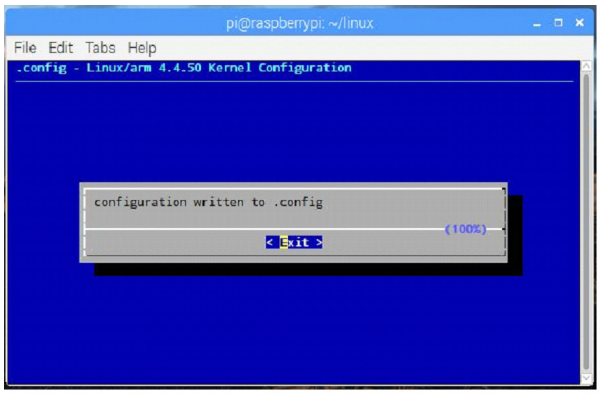




:



:




7. When this process is complete, exit back to the terminal.
8. Send the following commands to build the kernel:

   **Note: The compilation can be sped up by using the '-j n' option, with n equal to the number of processors * 1.5. This will reduce the compilation time significantly.**

   ```
   $ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- zImage modules dtbs
   ```

9. Once the kernel has finished building, power down the 4-mic development kit, remove the SD card (with the AVS already loaded), and use an SD card adapter to connect the card to the host. Identify the FAT partition (boot) and the ext4 partition (root). This will be necessary to install the kernel to the SD card. See the section *Install Directly onto the SD card* at the following link for details:

   https://www.raspberrypi.org/documentation/linux/kernel/building.md

10. After the installation of the kernel is complete, edit the /boot/config.txt file. On the RPi 2, send the following command to edit the file:

    ```
    $ sudo pico /boot/config.txt
    ```

    In the text editor, add the following options:

    ```
    dtoverlay=rpi-cxsmartspk-usb
    dtoverlay=i2s-mmap
    dtparam=i2c_arm=on
    program_usb_boot_mode=1
    ```

11. To ensure audio playback is through the driver just built/installed, the default audio device should be disabled. Make the following change to the /boot/config.txt file.

    ```
    dtparam=audio=off
    ```

    Save the file and exit.

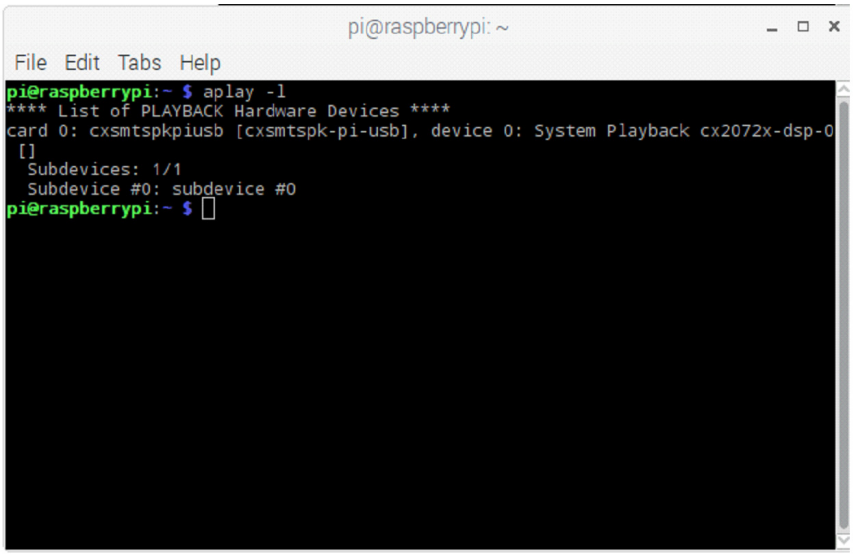
    Now the hardware connections should be established and the RPi 2 should be correctly configured. Refer to "Running LED Server, Node.js Service, Sample App, Wake Word Engine, and Recording Agent" on page 18 for running the 4-mic setup.

12. To verify that the driver is installed, open a terminal on the RPi 2 and send the following command:

    ```
    $ aplay -l
    ```

    The following device should display:

:

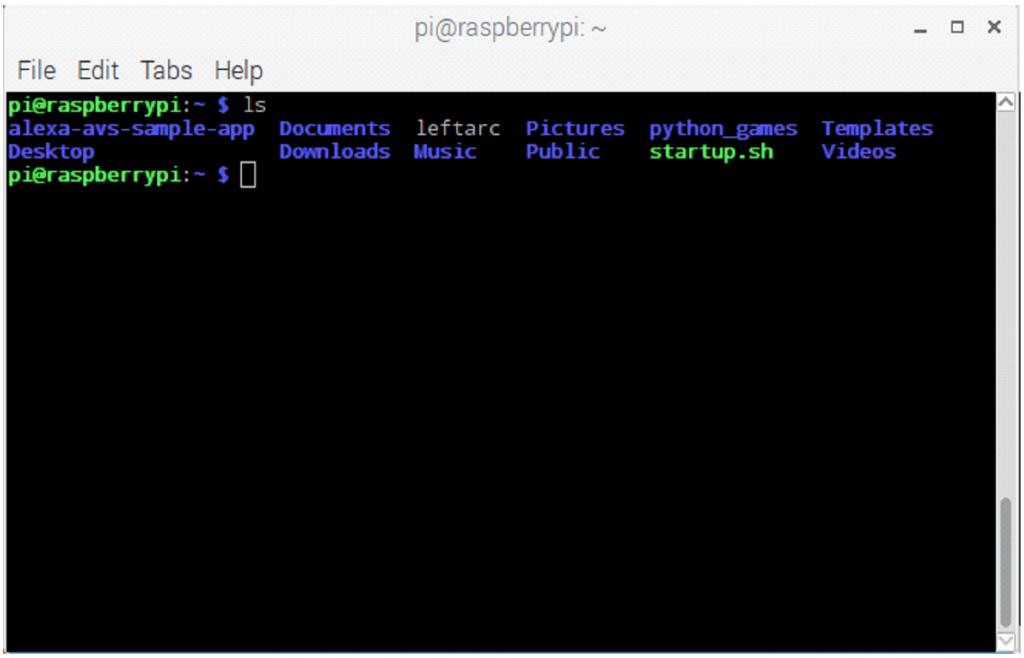

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: cxsmtspkpiusb [cxsmtspk-pi-usb], device 0: System Playback cx2072x-dsp-0
[]
Subdevices: 1/1
Subdevice #0: subdevice #0
pi@raspberrypi:~ $

# Running LED Server, Node.js Service, Sample App, Wake Word Engine, and Recording Agent

Once the installation is complete, you are ready to try out the Conexant AudioSmart 4-Mic Development Kit for Amazon AVS.

On the home directory, run the startup.sh script to open all necessary windows at once.

```
./startup.sh
```




Alternatively, these windows can be opened manually. See the following for more information.

**Note: It is recommended that these commands be run in the following order.**

- Terminal Window 1: for LED functionality on the 4-mic setup
- Terminal Window 2: to run the web service for authorization
- Terminal Window 3: to run the sample app to communicate with AVS
- Terminal Window 4: to run the wake word engine which allows you to start an interaction using the phrase "Alexa"
- Terminal Window 5: to run the recordingAgent, which sends audio content continuously to the client

## Terminal Window 1

Open a new terminal window and type the following command to run the LED server for the 4-mic setup:

```
cd ~/alexa-avs-sample-app/samples/tLED && sudo python tLEDServer.py
```

## Terminal Window 2

Open a new terminal window and type the following commands to bring up the web service which is used to authorize your sample app with AVS:

```
cd ~/alexa-avs-sample-app/samples/companionService && npm start
```

The server is now running on port 3000 and you are ready to start the client.

## Terminal Window 3

1. Configure the .asoundrc file with the correct settings.

   **Note: This step is necessary to ensure correct audio playback and recording.**

   ```
   cd ~/alexa-avs-sample-app/samples
   cp ~/leftarc ~/.asoundrc
   ```

2. Open a new terminal window and type the following commands to run the sample app, which communicates with AVS:

   ```
   cd ~/alexa-avs-sample-app/samples/javaclient && mvn exec:exec
   ```

See "AVS Setup: Amazon Account Login" on page 20 for more information.

## Terminal Window 4

**Note: Skip this step to run the same app without a wake word engine.**

This project supports two versions of Sensory's TrulyHandsFree wake word engine. The `-e` option is used to select the low-power embedded wake word engine running on the Conexant development kit or the high-performance wake word engine running on the RPi 2.

1. Open a new terminal window and use the following commands to bring up the wake word engine from Sensory. The wake word engine will allow you to initiate interactions using the phrase "Alexa".
2. To run the embedded low-power Sensory wake word engine (on the Conexant CX20924), type:

   ```
   cd ~/alexa-avs-sample-app/samples/wakeWordAgent/src
   sudo ./wakeWordAgent -e gpio
   ```

   **Note: For the low-power Sensory wake word engine, the wake word agent must be run as admin.**

3. To run the high-performance Sensory wake word engine (on the RPi 2), type:

   ```
   cd ~/alexa-avs-sample-app/samples/wakeWordAgent/src && ./wakeWordAgent -e sensory
   ```

## Terminal Window 5

Open a new terminal window and type the following command to run the recordingAgent, which sends the processed microphone signals to the javaclient:

```
cd ~/alexa-avs-sample-app/samples/recordingAgent && ./run.sh
```

You now have a working hands-free AVS prototype.

## AVS Setup: Amazon Account Login

An Amazon account will need to be connected to use the Conexant AudioSmart 4-Mic Development Kit for Amazon AVS. A monitor, keyboard, and mouse is required for this procedure.

1. Using the HDMI and USB ports on the RPi, connect the RPi to the monitor, keyboard, and mouse.
2. After the RPi boots up and the startup.sh is run, the following window will popup. Note that the 'Bearer Token' is empty.

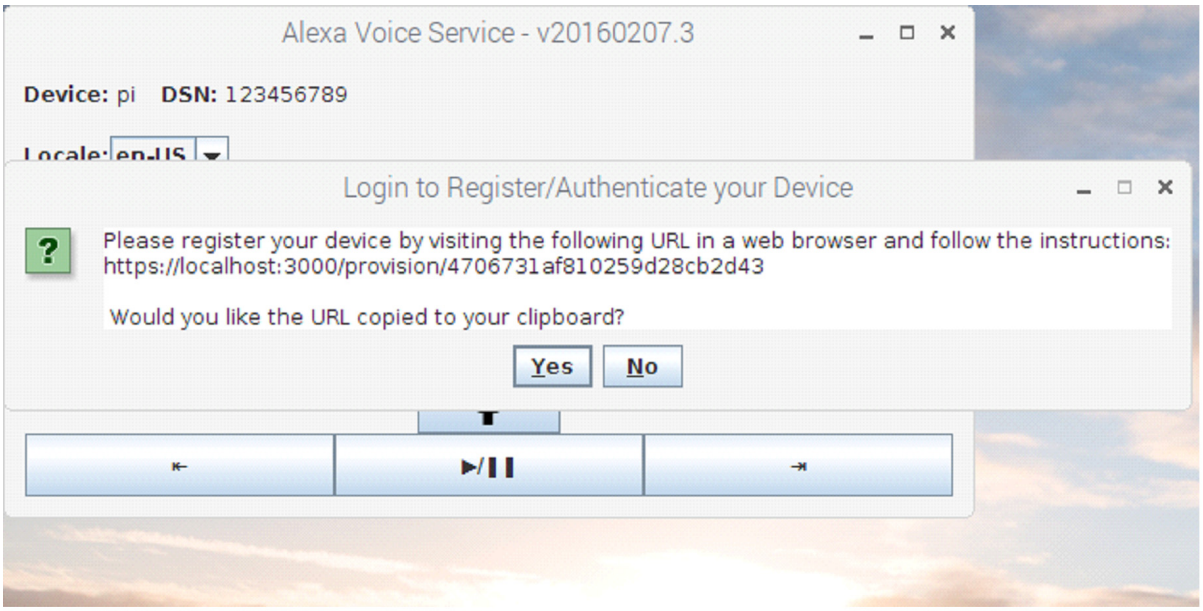
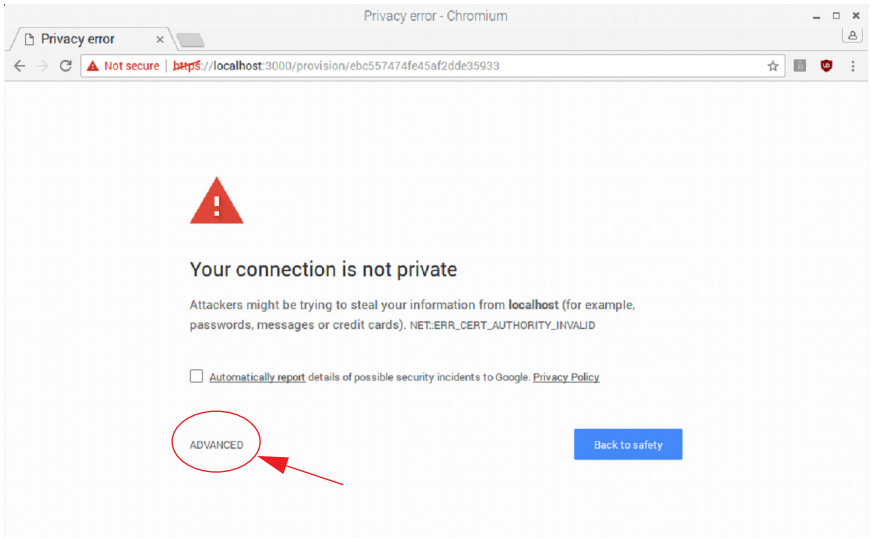
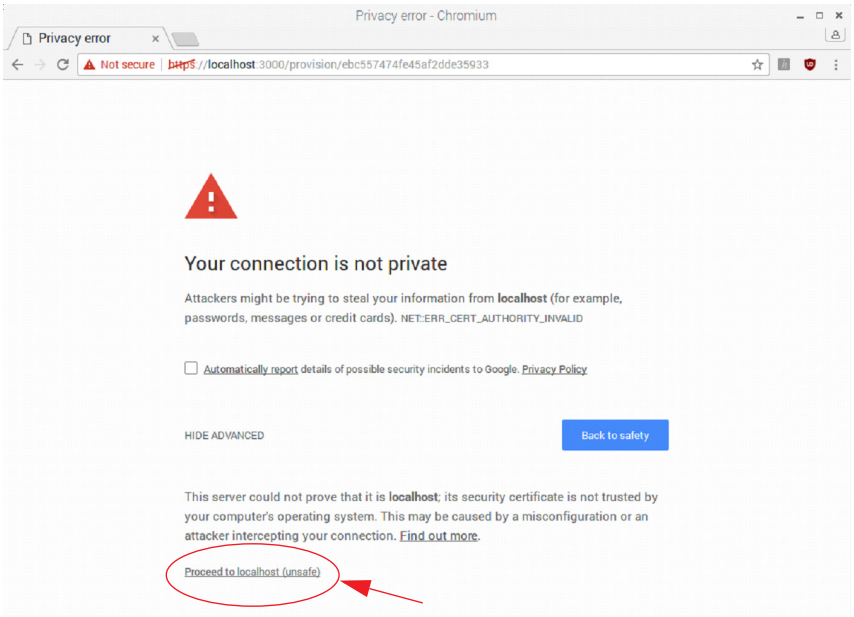



**Figure 9: Alexa Voice Service User Interface**

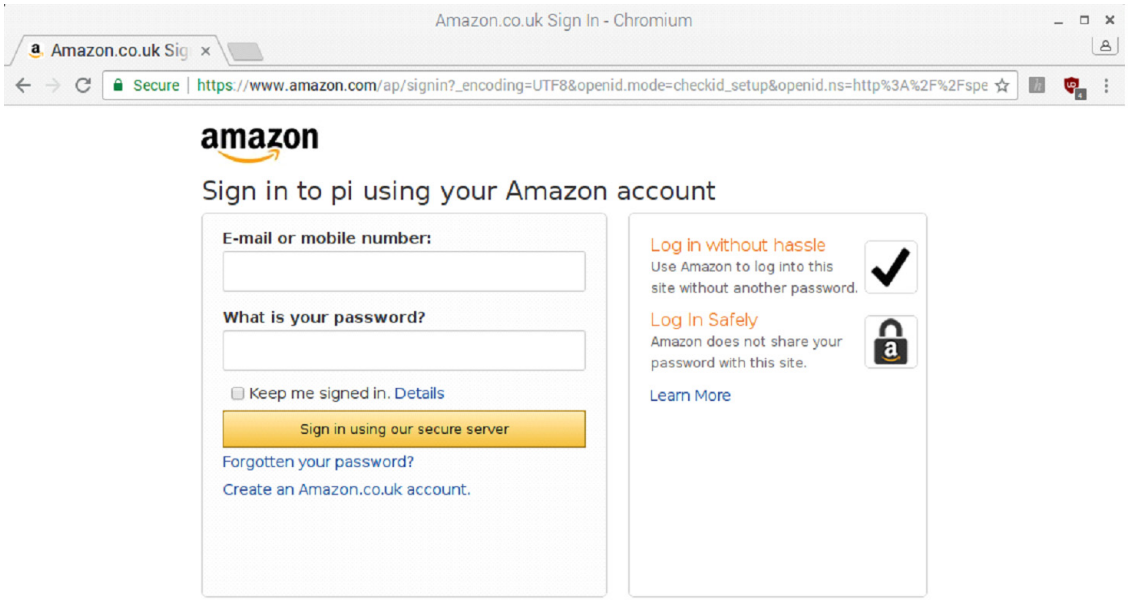
3. Click **Yes** and open a web browser. Go to the copied URL.

If a warning page appears, click **Advanced** and then click **Proceed to Localhost (unsafe)**. The following shows representative screens.

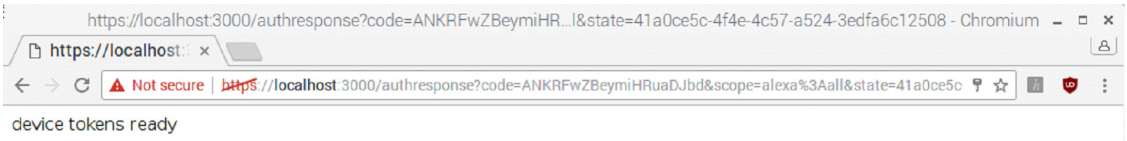


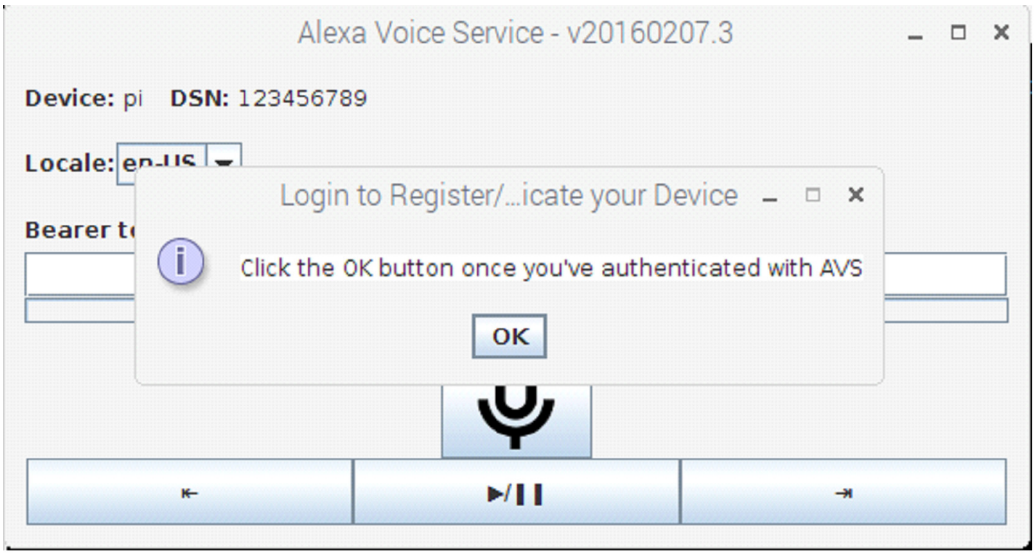
4. The URL will go to a page to sign in using an Amazon account. Login with your Amazon account credentials.




5. The next page should display the message 'device tokens ready' as shown below. The web browser can now be closed.



6. Click OK in the following window.




Now, the 'Bearer Token' field from Step 2 should be filled. The Amazon account is set up and the demo unit is ready to be used.

# Verifying the Setup

Once the RPi is correctly configured and the hardware connections are established, the unit is ready to be run when all LEDs turn off.

- Say **Alexa** and three LEDs will point in the direction of the talker.
- When the command is being processed, blue LEDs will flash around the board.
- While the response is played out, the LEDs on the device will brighten and dim.

# Installing the Cypress Siena USB-to-I2C Device Driver

A driver must be installed to enable the onboard Cypress Siena USB-to-I2C device. Once enabled, this device will allow the user to communicate with the CX20924 evaluation board over $I^2C$. This is necessary to flash firmware.

1. Connect the Type A to Type B USB cable between your laptop and the CX20924 evaluation board to connector **J2**.

   a. This cable will interface to a Cypress USB-to-$I^2C$ control device, allowing you to control parameters of the CX20924 evaluation board using a Conexant GUI.

   b. This cable can also be used to flash new Firmware to the CX20924 evaluation board if needed (see "Flashing New Firmware" on page 26).

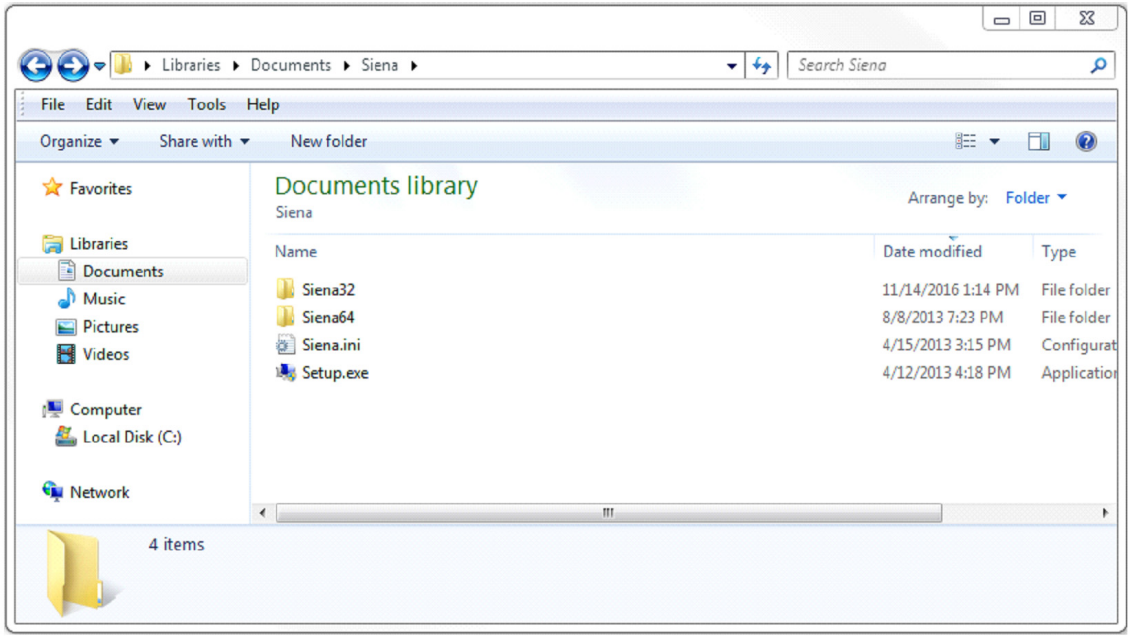
2. The Cypress Siena driver that enables USB-$I^2C$ communication between your laptop and the CX20924 evaluation board can be downloaded from:

   http://www.conexant.com/avs-support/Conexant_Siena_Driver.rar

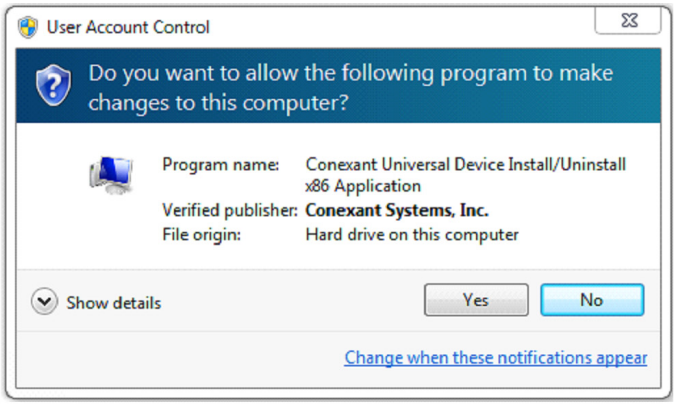
   a. Filename: **Conexant_Siena_Driver.rar**

   b. Extract the **Conexant_Siena_Driver.rar** file onto your laptop.

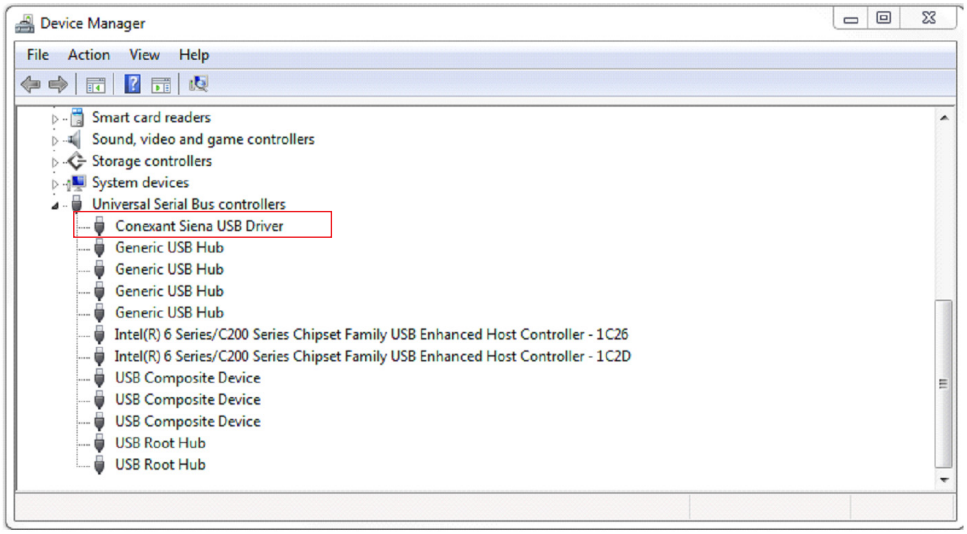
3. Run the **Setup.exe** file included in the **Conexant_Siena_Driver** folder as shown below:



4. When the following message appears. Click **Yes**.




In the Device Manager, you should shortly see **Conexant Siena USB Driver** appear under **Universal Serial Bus Controllers**.




The Cypress Siena device is now ready to use.

# Flashing New Firmware

The CX20924 evaluation board comes pre-flashed with firmware (FW). The following instructions are only needed if a new FW version needs to be loaded.

1. Connect the standard USB cable to allow Cypress Siena USB-I$^2$C control if not done already.
2. Firmware flashing on the CX20924 device is done from the laptop/PC through the onboard Cypress Siena CY7C68013A I$^2$C-to-USB converter device going to the CX20924 and the SPI flash device.
   a. FW update system flow: **Laptop/PC>Cypress CY7C68013A>CX20924>SPI Flash**.
   b. A driver is required to use the Cypress Siena CY7C68013A device, so make sure you have previously installed the Cypress Siena CX7C68013A driver as described in "Installing the Cypress Siena USB-to-I2C Device Driver" on page 24.

The FW release package is stored in a folder named **fcp**. This folder contains a number of subfolders as shown below.

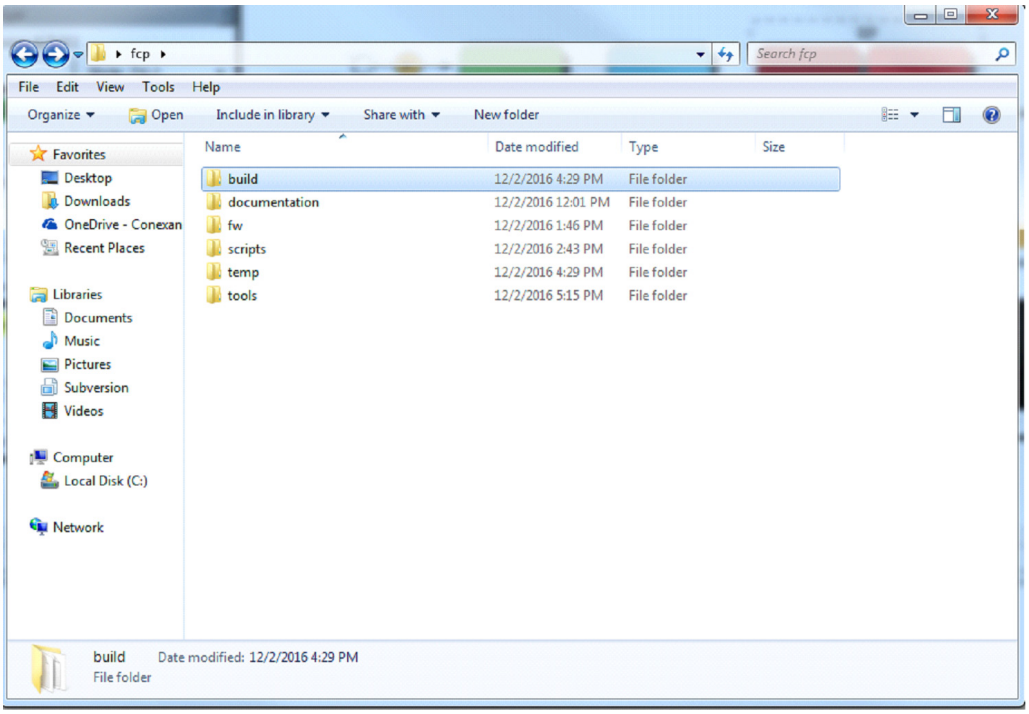



**Figure 10: Example FCP Folder**

3. Open the **build** folder which contains the ***.sfs** FW image file.

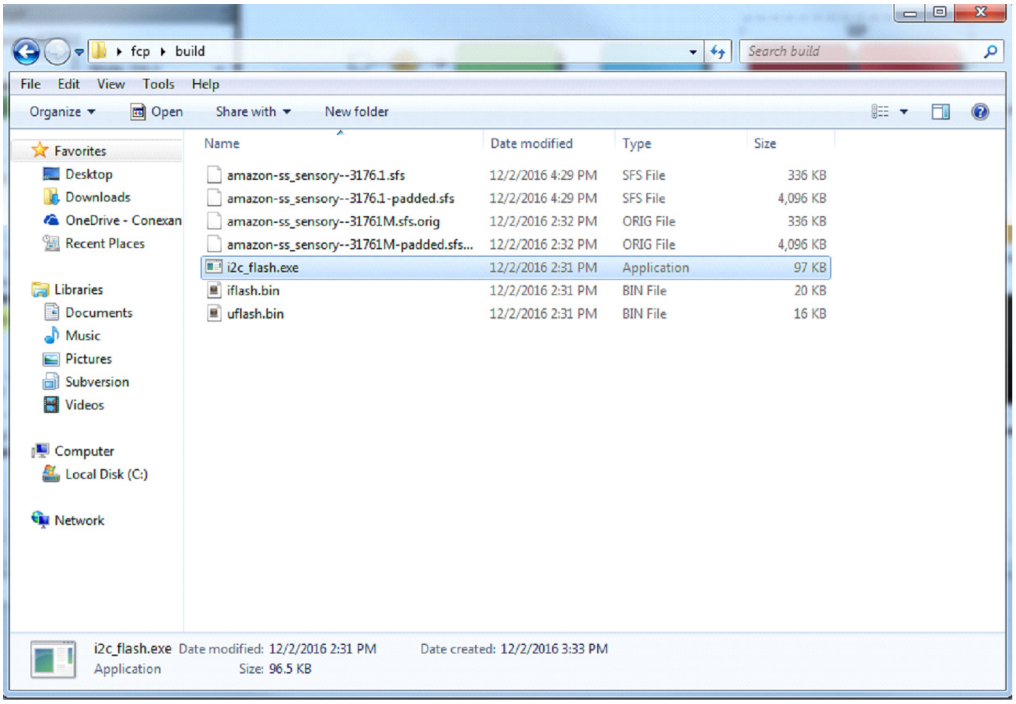
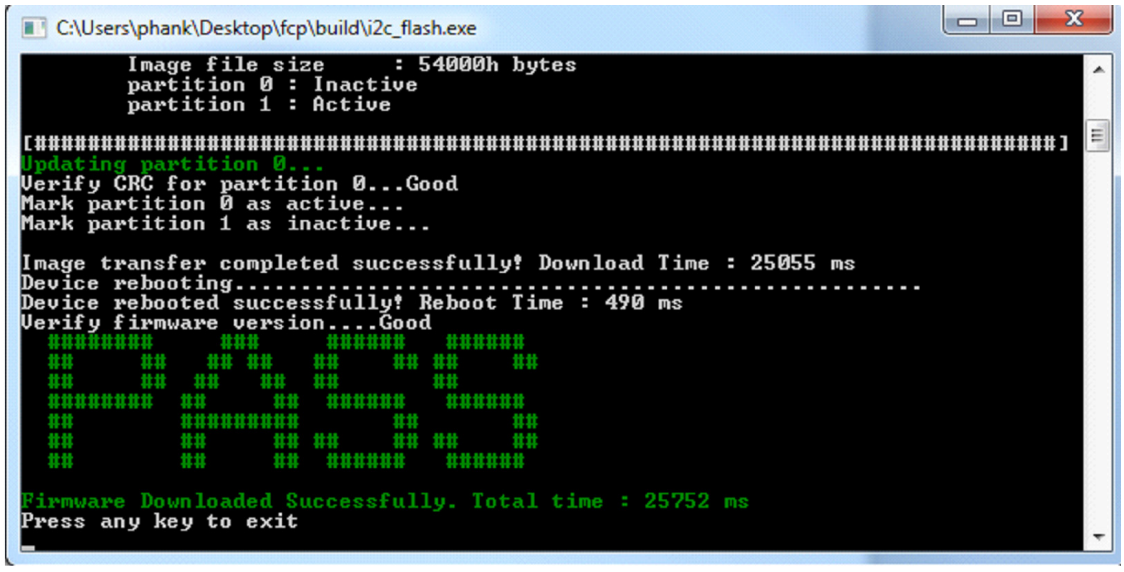



**Figure 11: Build Folder Contents**

4. Double click on the **i2c_flash.exe** file. FW updating will begin, and the following window will appear.




Wait for the green **PASS** message to appear, indicating that the FW was updated successfully. The FW upgrade should take less than 30 ms to complete.

5. Once the successful PASS message appears, close the window and cycle power on the board by unplugging and plugging back the power on the CX20924 evaluation board. The CX20924 device is ready to be used with the new updated FW.

**www.conexant.com**

Headquarters: 1901 Main Street, Suite 300 Irvine, CA,92614
General Information: U.S. and Canada: 888-855-4562 | International: 1 + 949-483-4600




**Reader Response:** Conexant strives to produce quality documentation and welcomes your feedback. Please send comments and suggestions to americasupport@conexant.com. For technical questions, contact your local Conexant sales office or field applications engineer.