# **Quick Start Guide** for Development kit for Amazon AVS with Synaptics 2Mic and NXP PICO-PI-IMX 7D



Synaptics 2 Microphone Module

NXP PICO-PI-IMX7          Synaptics CX20921 Evaluation Board

This development kit available through Arrow consists of a **Synaptics AudioSmart™ 2-Mic Development Kit for Amazon AVS** and a **NXP PICO-PI-IMX7** development board for the NXP i.MX 7D processor. This guide provides step-by-step instructions for setting up the development kit. It demonstrates how to access and test Amazon AVS, Amazon's C++ device SDK (running on the PICO-PI-IMX7) and a third-party wake word engine that responds to "Alexa" (running the low-power Sensory wake word engine or the high-performance Sensory wake word engine with limited license).

When finished, you'll have a fully functioning Amazon AVS prototype that uses the Synaptics AudioSmart™ 2-Mic Development Kit as an audio front end, and the PICO-PI-IMX7 i.MX 7D development board as the processor handling the "Alexa" wake word recognition and interface to Amazon's AVS service. The processor system is also available as the PICO-IMX7 module, a production intent System on Module (SoM) for use in consumer products.

# Contents

# Section 1: Hardware & Setup

The **development kit for Amazon AVS with Synaptics 2Mic and NXP PICO-PI-IMX 7D** is available at:
https://www.arrow.com/en/products/synapticsnxp2micavs/arrow-development-tools

The kit includes

1. PICO-PI-IMX7D, SOM + development board.
2. Synaptics AudioSmart™ 2-Mic Development Kit for Amazon AVS, including;
   - CX20921 evaluation board, pre-flashed with firmware
   - Microphone module with two omnidirectional mics
   - Microphone holder board
   - Stereo 3.5mm male-to-male audio cable
   - Micro-USB cable
   - Type A to Type B USB cable
   - Cable assembly (colored wires)
3. +5V power supply for CX20921 evaluation board
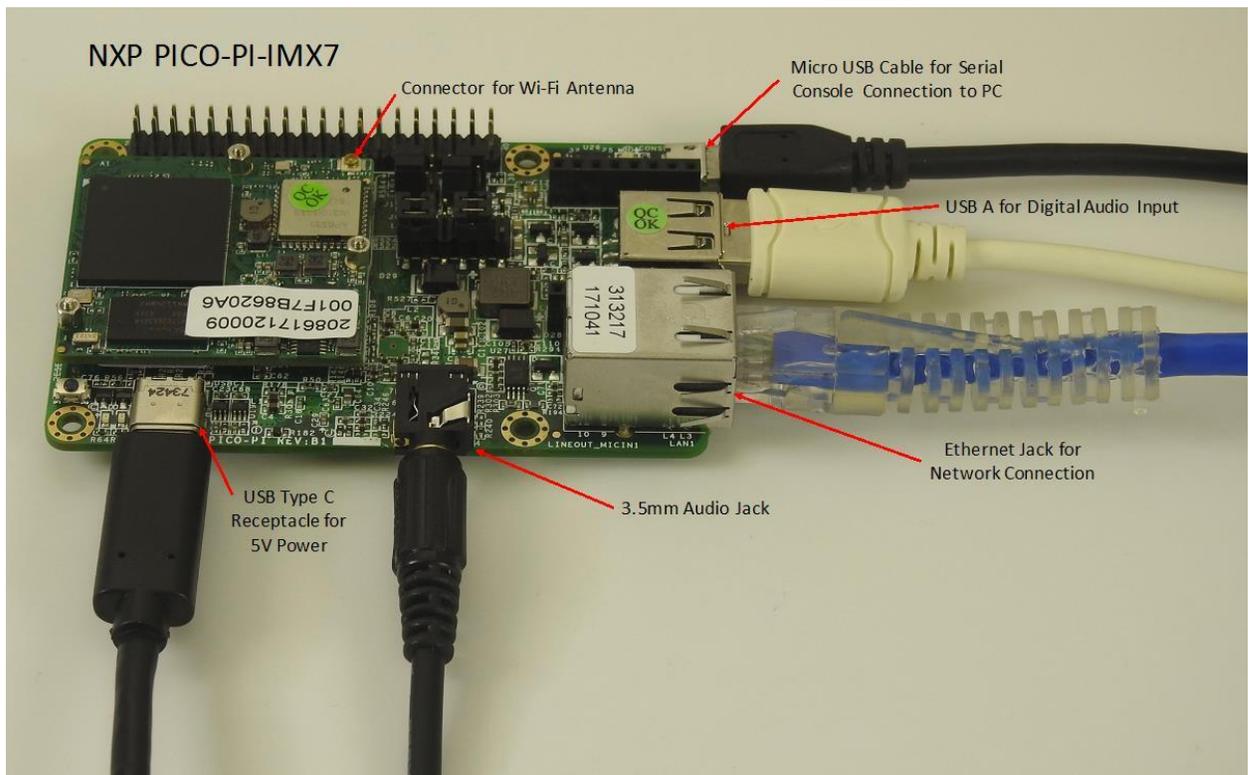


In addition to the kit, you will also require:

4. An external powered speaker with 3.5mm audio cable
   - Synaptics provides an External-Loudspeaker-Guidelines document at
     http://www.conexant.com/avs-support/External-Loudspeaker-Guidelines.pdf
5. PC with internet connection and a USB port
6. USB A to USB Type C cable

7.  Optional USB 5V power adaptor (or you can use your PC)

## Connect CX20921 evaluation board to PICO-PI-IMX7 board

1.  Plug the 3.5mm audio cable into the Microphone and Headphone jack on the PICO-PI-IMX7 board, and the Audio Input jack on the CX20921 evaluation board.

2.  Plug the USB A end of the USB A to micro USB cable into the USB A receptacle on the PICO-PI-IMX7 board, and the micro USB end into the CX20921 evaluation board USB Device port.

3.  Connect the powered speaker using a 3.5mm audio cable to the Lineout or Headphone Output jacks on the CX20921 evaluation board.

4.  Connect the Microphone board to the CX20921 evaluation board by plugging the 3.5mm cable into the Microphone Input jack.
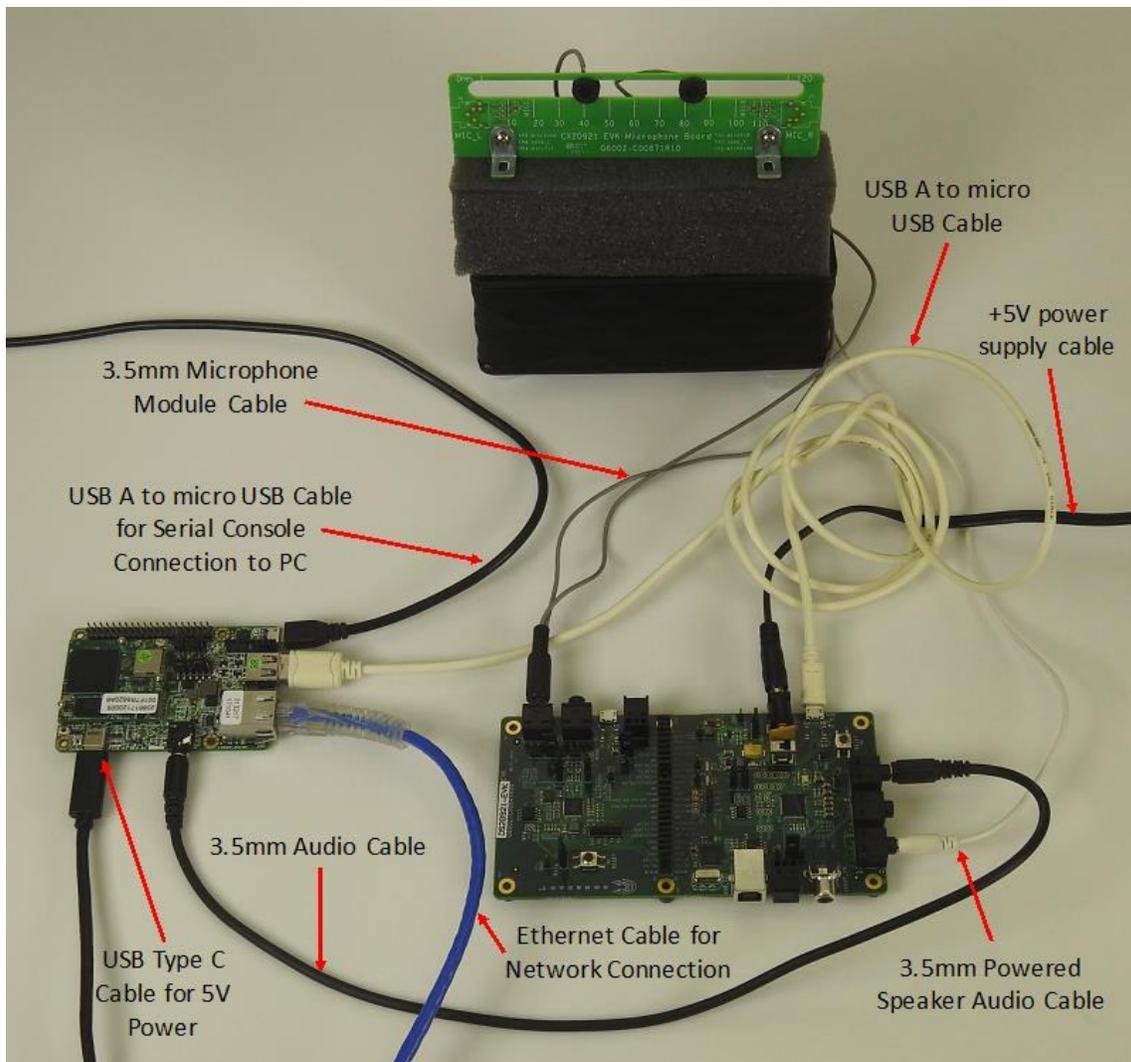


5.  Place the Microphone board on top of the powered speakers using foam or putty for audio isolation as recommended in the Synaptics External-Loudspeaker-Guidelines document.
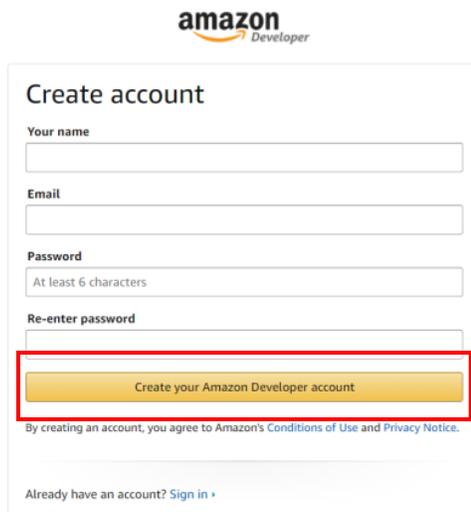
Microphone Module on top of Powered Speaker with Isolating Foam

6. Connect a USB A to micro USB cable to the Serial Console connector on the PICO-PI-IMX7 board and your PC

7. Connect +5V power supply included with the 2Mic Audio kit to the J1 connector on the CX20921 EVK board.

8. Provide 5V power to the PICO-PI-IMX7 board at the USB Type C connector from your PC or a USB power adaptor.

   Connect the PICO-PI-IMX7 board to a network via Ethernet

# Section 2: Amazon Account Setup

1. Create a free developer account at developer.amazon.com. You should review the **AVS Terms and Agreements** here. Create Alexa Voice Service (AVS) project.



2. Create a device and security profile. Follow the steps here to register your product and create a security profile.

Make note of the following parameters.

- ProductID (also known as Device Type ID),
- ClientID, and
- ClientSecret

**Important:** Make sure your **Allowed Origins** and **Allowed Return URLs** are set under **Security Profile > Web Settings** (see Create a device and security profile):



- **Allowed Origins:** http://localhost:3000
- **Allowed Return URLs:** http://localhost:3000/authresponse

# Section 3: Software Setup

The PICO-PI-IMX7 board comes pre-loaded with a Yocto Linux image, requiring the user to only run the scripts needed to build and run the app. To ensure your board is running the latest Yocto Linux image, download the image from http://download.technexion.net/files/avs-conexant.

9. Open a serial console (Putty or Tera Term) on your PC and select the board's COM port number enumerated by the host system. Use the following port settings:
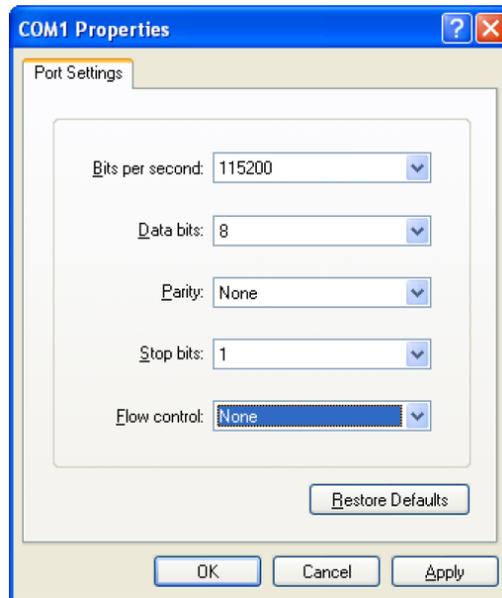


10. Power on the board. Make sure everything is connected before the system boots up.

11. On your serial console, login by typing "root"



```
Freescale i.MX Release Distro 4.1.15-1.2.0 pico-imx7 /dev/ttymxc4

pico-imx7 login: root
```

12. Set the date by using the following command
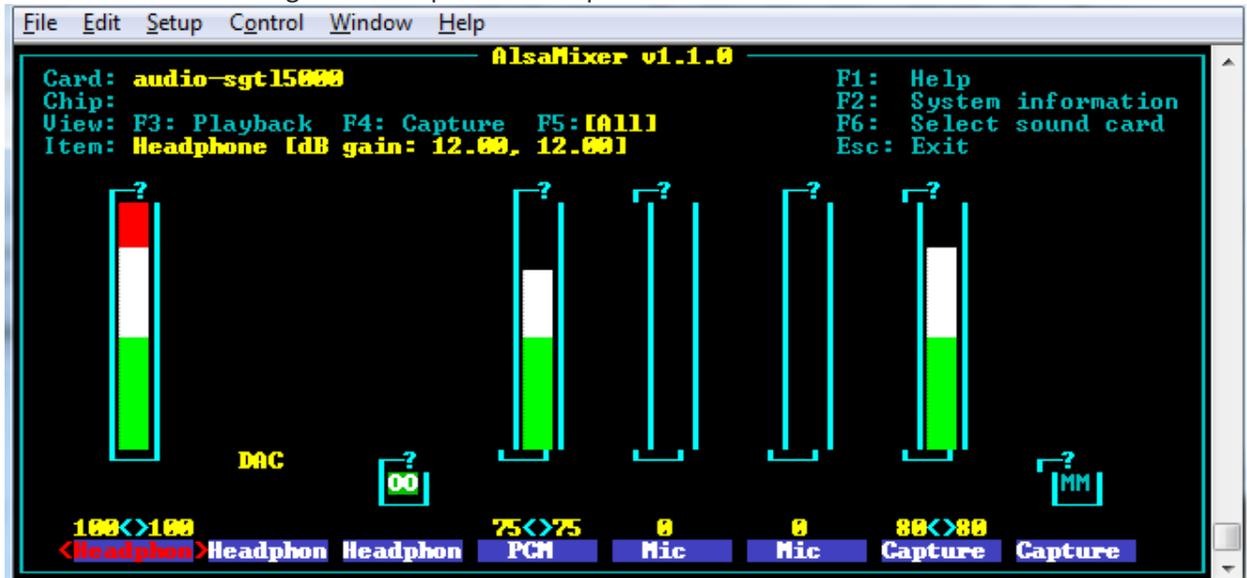
"./home/root/Alexa_SDK/Scripts/setUTCTime.sh"

13. Set the speaker and microphone level using the "alsamixer" command
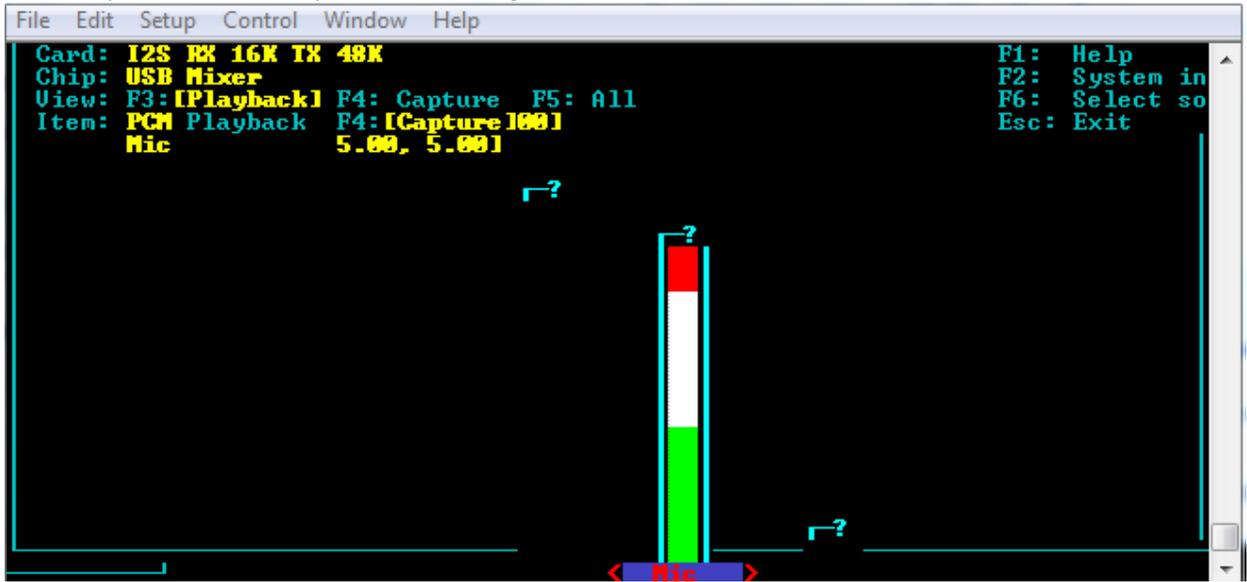


Use F6 to select audio-sgtl15000 chip to set headphone level to maximum



14. Register for an Amazon developer account if there isn't one. Refer to **Section 2** to setup the Amazon developer account.

15. Change directory to Alexa_SDK using "cd Alexa_SDK"

16. Run the setupAVS.sh script by typing "./setupAVS.sh". You will see output as following

```
root@imx7d-pico:~# cd ~/Alexa_SDK
root@imx7d-pico:~/Alexa_SDK# ./setupAVS.sh

==================================================================
 Welcome to Alexa SDK Image for NXP i.mx7D Pico Pi
 Let's setup your environemt...

 ** Please enable the Network access by Ethernet/Wifi **

 For using Wake Word Detection, please, accept the Sensory license...

==================================================================


Wed Oct  4 20:25:07 UTC 2017
Cloning into 'sensory-alexa'...
remote: Counting objects: 149, done.
remote: Total 149 (delta 0), reused 0 (delta 0), pack-reused 149
Receiving objects: 100% (149/149), 8.46 MiB | 207.00 KiB/s, done.
Resolving deltas: 100% (51/51), done.
Checking connectivity... done.
The Sensory TrulyHandsfree binaries are not licensed.
Press RETURN to review the license agreement and update the files.
```

Hit Enter to proceed through Sensory license agreement.

```
Sensory TrulyhandsfreeÖ Library License

You may not use the Sensory TrulyHandsfreeÖ library or code files except in
compliance with this license. The Sensory library and code is distributed on an
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.

THE FOLLOWING TERMS AND CONDITIONS (the end user license agreement, or
ôEULAö) SET FORTH THE RIGHTS BEING LICENSED TO YOU (THE ôLicenseeö) FOR
THE USE OF SENSORYÆS TRULYHANDSFREE(Ö) (herein referred to as "Software
Product" or "Software"). THESE RIGHTS ARE THE ONLY RIGHTS YOU HAVE TO THE
SOFTWARE, SO IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS EULA, DO
NOT USE THE SOFTWARE.


License and Copyright

Sensory grants Licensee a personal, non-commercial, non-transferable,
non-sublicensable, limited, and non-exclusive right to use the Software on a
single computer, tablet, or mobile device for use by a single concurrent user,
and solely provided that you adhere to all the terms and conditions of this
EULA. Modifying, translating, duplicating or copying all or part of the
Software is strictly prohibited. In addition, no right is granted to rent,
--More--
```

Read the agreement and enter "yes" if you agree and would like to proceed.

```
Copyright ⌐ Sensory, Inc. 2016. All Rights Reserved. (http://sensory.com/)
Do you accept this license agreement? [yes or no]:
```

Wait for the Sample app to be built.

```
License key applied.

[ 16%] Built target AVSCommon
[ 18%] Built target gmock_main
[ 19%] Built target gtest
[ 20%] Built target gtest_main
[ 21%] Built target AttachmentCommonTestLib
[ 22%] Built target AlexaClientSDKInitTest
[ 23%] Built target AttachmentManagerV2Test
[ 23%] Built target AttachmentReaderTest
[ 24%] Built target AttachmentTest
[ 25%] Built target AttachmentWriterTest
[ 26%] Built target CapabilityAgentTest
[ 26%] Built target DialogUXStateAggregatorTest
[ 26%] Built target ExceptionEncounteredSenderTest
[ 27%] Built target HandlerAndPolicyTest
[ 28%] Built target NamespaceAndNameTest
[ 29%] Built target ConfigurationNodeTest
[ 30%] Built target ExecutorTest
[ 31%] Built target JSONUtilTest
[ 32%] Built target LogEntryStreamTest
[ 33%] Built target LoggerTest
[ 34%] Built target RequiresShutdownTest
[ 34%] Built target SharedDataStreamTest
```

```
[ 92%] Built target AlexaDirectiveSequencerLibraryTest
[ 93%] Linking CXX executable AudioInputProcessorIntegrationTest
[ 93%] Built target AudioInputProcessorIntegrationTest
[ 93%] Linking CXX executable AudioPlayerIntegrationTest
[ 93%] Built target AudioPlayerIntegrationTest
[ 94%] Linking CXX executable ServerDisconnectIntegrationTest
[ 94%] Built target ServerDisconnectIntegrationTest
[ 94%] Linking CXX executable SpeechSynthesizerIntegrationTest
[ 95%] Built target SpeechSynthesizerIntegrationTest
[ 96%] Built target DefaultClient
[ 97%] Linking CXX executable SampleApp
[100%] Built target SampleApp


====================================================================
 Now we will setup your AVS Credentials
====================================================================


====================================================================
 Please provide the following info of your AVS device
 You can get your info from https://developer.amazon.com/home.html
====================================================================

Enter your Product ID[my_device]: █
```

Enter your Amazon credentials that were created in section 2.

```
[100%] Built target SampleApp


====================================================================
 Now we will setup your AVS Credentials
====================================================================


====================================================================
 Please provide the following info of your AVS device
 You can get your info from https://developer.amazon.com/home.html
====================================================================

Enter your Product ID[my_device]: picopi
Enter your Clent Id[amzn1.application-oa2-client.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxx]: amzn1.application-oa2-⌐
Enter your Client Secret[xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxx]: 83e66e6626c2b8c745
Are your Credential OK [Y/N]? █
```

Next step is to log-in with your Amazon account details.

```
============================================================
Now we will try to authenticate your AVS device

Please Open a VLC connection with IP = 192.168.1.140

Open FireFox on your VLC connection and copy/paste the below
URL to login with your Amazon AVS Developer user/pass

* Running on http://127.0.0.1:3000/ (Press CTRL+C to quit)
```
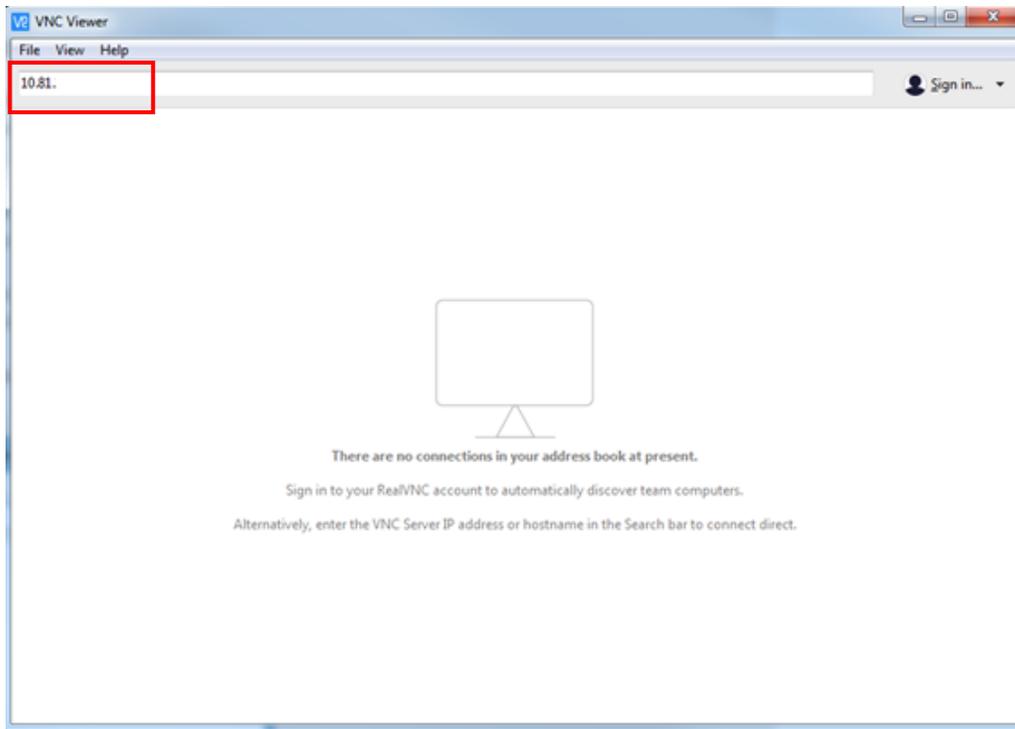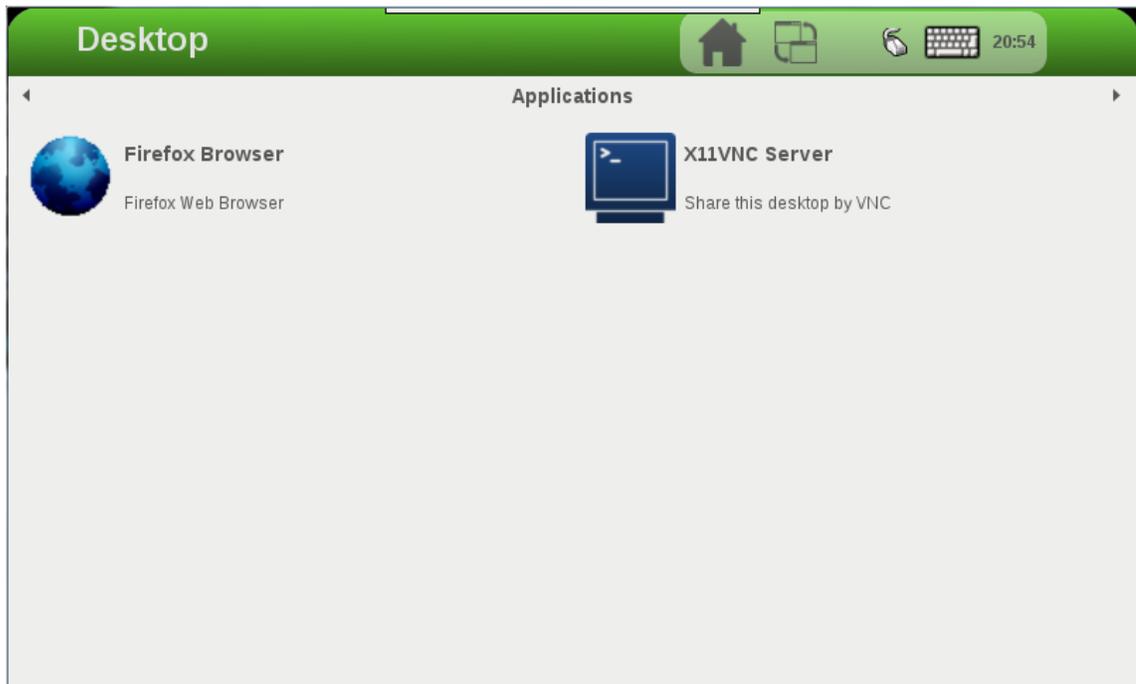
On you host machine install VNC viewer. You can download it from download VNC viewer (available for Mac, Linux, Windows, and more).

17. Open a VNC Viewer window and enter your PicoPi's IP address.

18. You should be connected to your PicoPi.



19. Click once on the Firefox browser icon

20. Enter the address shown by the app as below



21. Login with your Amazon account details
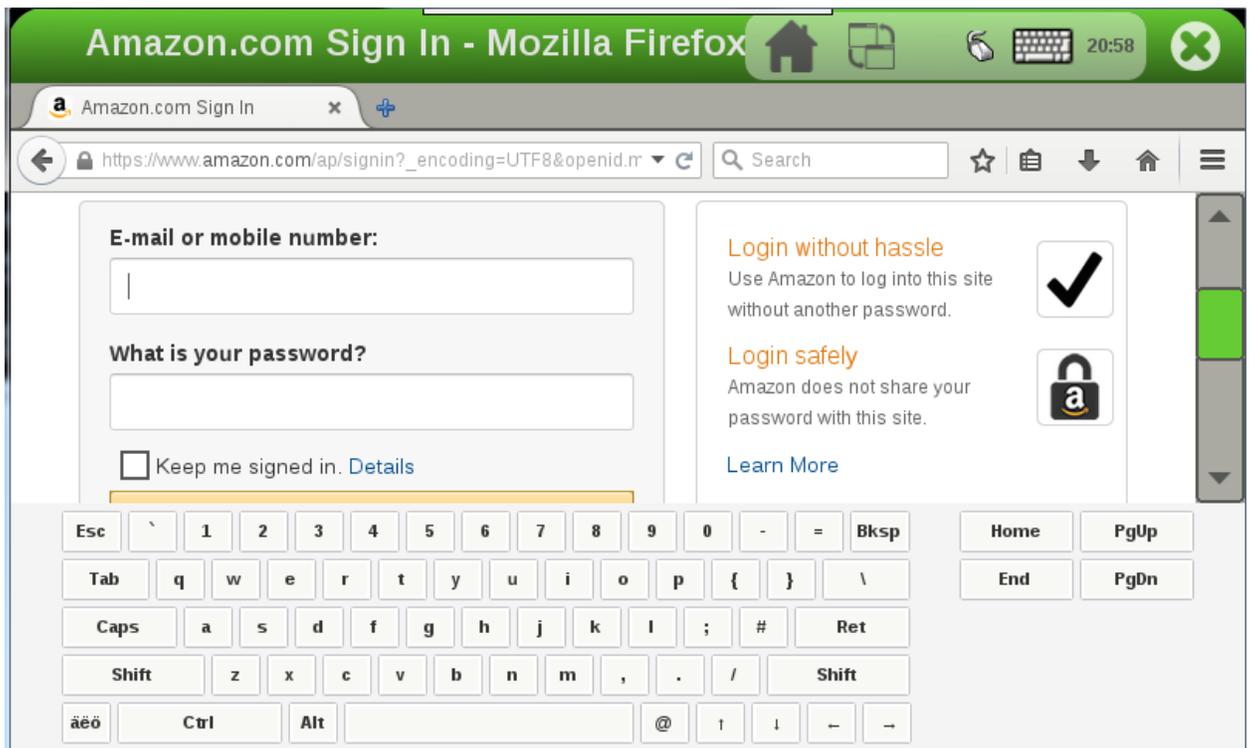
22. Once you see the output as below, shutdown Firefox and the VNC connection
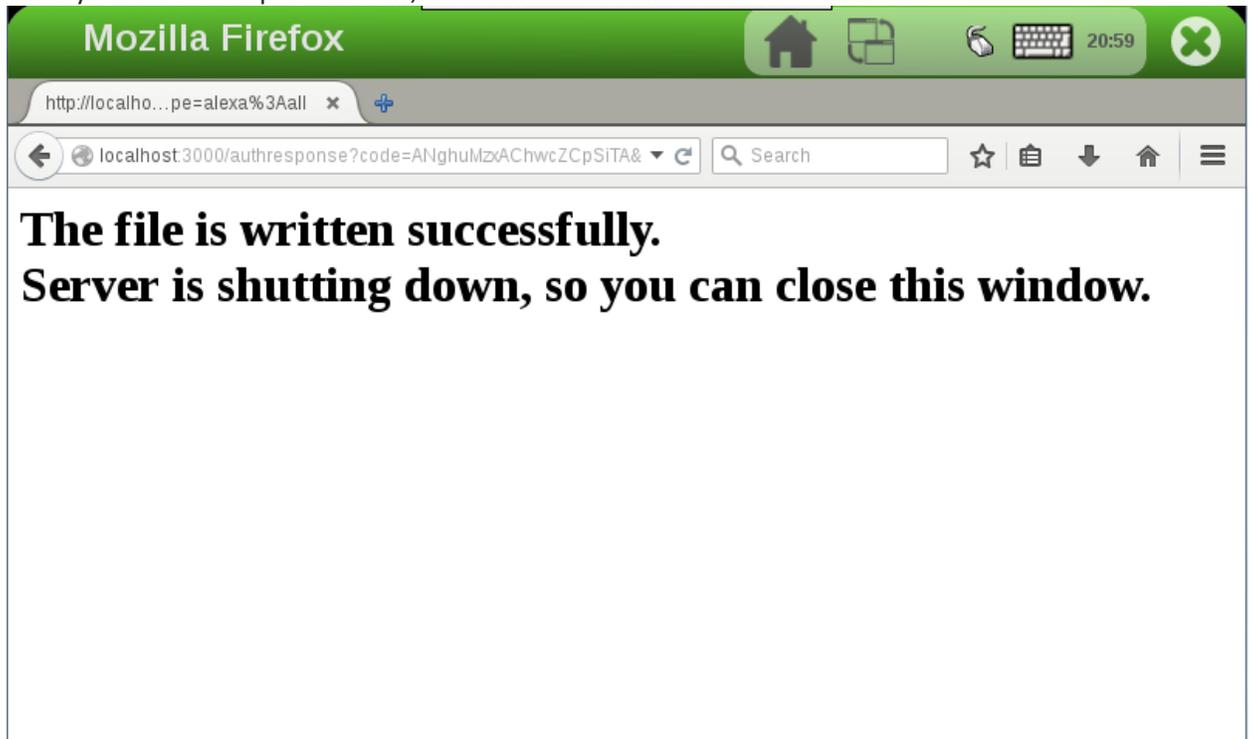


23. Your app should be ready now

```
================================================================
Done!!! You are now ready to start with Alexa SDK for NXP Pico Pi

To run the SampleApp:

  cd ~/Alexa_SDK/avs-sdk-client/SampleApp/src/

  TZ=UTC ./SampleApp ../../Integration/AlexaClientSDKConfig.json \
  ../../Integration/inputs/SensoryModels/ DEBUG9

NOTE: For make Alerts/Timers works properly on AVS, you need to
sync your date to UTC. You can do it by running next script:

  /home/root/Alexa_SDK/Scripts/setUTCTime.sh

Enjoy !!!
================================================================
```

24. Follow instructions on changing the directory and running the app as below
    cd ~/Alexa_SDK/avs-sdk-client/SampleApp/src/
    TZ=UTC ./SampleApp ../../Integration/AlexaClientSDKConfig.json
    ../../Integration/inputs/SensoryModels/ DEBUG9

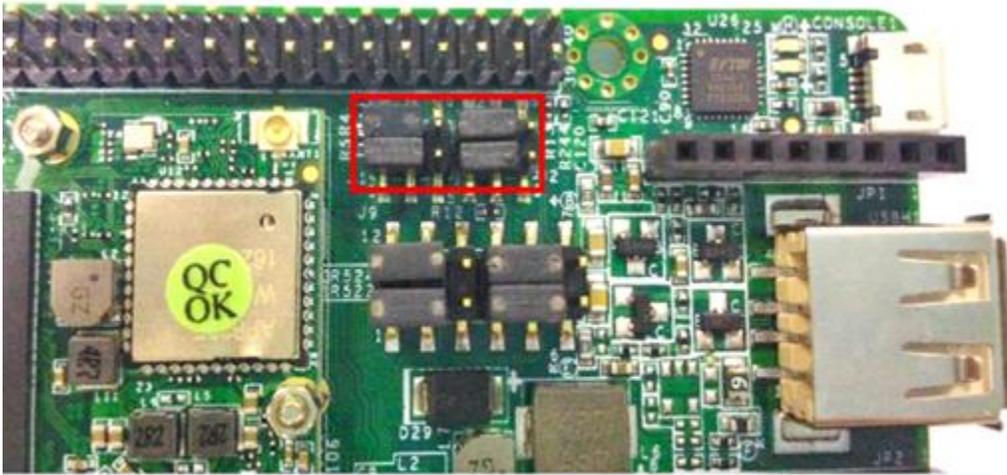25. You should see an output as below



26. Talk to Alexa.
27. You can now talk to Alexa by simply using the wake word "Alexa". Try the following –

    Ex: "Alexa ,what's the time?"

    Or "Alexa, what's the weather in Seattle?"

# Section 4: Board Re-flashing

1. Change J2 jumpers to put the Pico-Pi in serial download mode, as shown below.
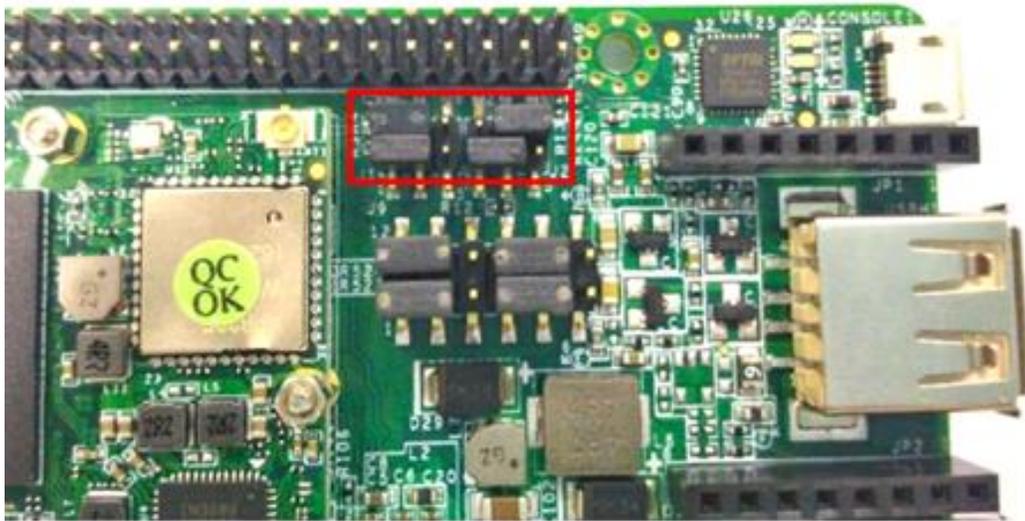


Jumper Setup (Download Mode)

2. Connect the Pico-Pi to your Linux PC using the provided USB-A to USB-C cable.
3. Download the bootbomb software from here:
   ftp://ftp.technexion.net/development_resources/development_tools/installer/pico-imx7-imx6ul-imx6ull_otg-installer_20170112.zip
4. Extract the zip file to your Linux PC.
5. Change directory to the unzipped folder.
   ex: "cd pico-imx7-imx6ul-imx6ull_otg-installer_20170112/"
6. Go to the linux folder: "cd linux" .
7. Change permissions for all the following files
   a. "chmod 0777 ./ imx_usb"
   b. "chmod 0777 ./ imx_usb_32"
8. Execute the following command: "sudo ./imx_usb ../pico-imx7d_bootbomb_20170112.imx".
9. You will see your board appear as a mass storage device. You can confirm this by using following command: "cat /proc/partitions"

10. DD the "$name_of_Alexa_PicoPi_image.sdcard" file to your mass storage device
    "pv $name_of_Alexa_PicoPi_image.sdcard |sudo dd of=/dev/sdb bs=1M && sync"
11. Wait for the transfer to complete, you will see the output as below

---

3.56GB 0:03:37 [16.8MB/s] [================================>] 100%

0+29184 records in

0+29184 records out

3825205248 bytes (3.8 GB) copied, 217.703 s, 17.6 MB/s

---

12. Disconnect USB-C cable and change J2 jumpers back to allow the board to boot from eMMC, as
    shown below.



Jumper Setup (Boot Mode)