

TI mmWave Training

xWR16xx mmWave Demo

NOTE: ES2.0 devices only

This version of the mmWave Demo lab will work only with xWR1642BOOST ES2.0 EVMs, which require **mmWave SDK version 2.0 or above**. These EVMs are marked with a sticker which says "ES2.0". EVMs which do not have this sticker have ES1.0 devices and require mmWave SDK version 1.xx.xx.xx. To download past versions of mmWave Industrial Toolbox which support ES1.0 EVMs, please follow the directions provided under **How to access previous Industrial Toolbox versions** at the bottom of the [Industrial Toolbox landing page](#)

Contents

- Overview
- Requirements
- Software setup
 - Pre-requisites
 - Downloading the Lab Project
 - Building the project
- Hardware setup
 - Preparing the EVM
 - Connecting the EVM
- Running the Demo

Lab Overview

- Configurable visualization tool for processed radar data
- The following plots are available:
 - Scatter Plot
 - Range Profile
 - Noise Profile
 - Range Azimuth Heat Map
 - Range Doppler Heat Map
 - Statistics



1. Requirements

- Software

- **Pre-requisites**

- [TI mmWave SDK](#) specified in Lab release notes, and all related dependencies installed as mentioned in the mmWave SDK release notes.

- [Google Chrome](#) with TI Cloud Agent Extension

- For running the mmWave Demo Visualizer
 - Download from [TI Cloud Agent](#) or install when accessing the [demo](#)

- mmWave SDK Demo

- Download from [TI Resource Explorer](#)

- UniFlash

- For flashing firmware images onto
 - Download from [TI.com/tool/uniflash](#)

- XDS110 Drivers

- For EVM XDS device support
 - Included with CCS Installation, or standalone through [TI XDS Emulation Software](#)

- Hardware

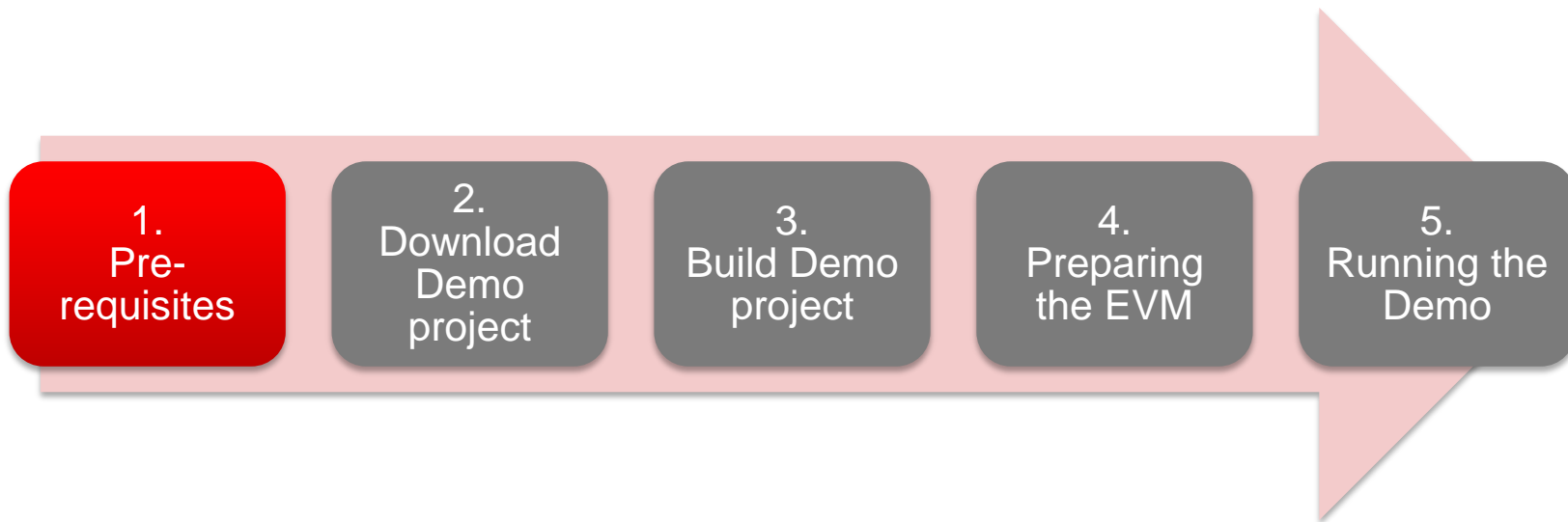
- xWR16xx ES2.0 EVM

- Micro USB cable (included in the EVM package)

- 5V/2.5A Power Supply

- [Purchase from Digikey](#)

Steps



1. Pre-requisites

1. Install Pre-requisites

2

3

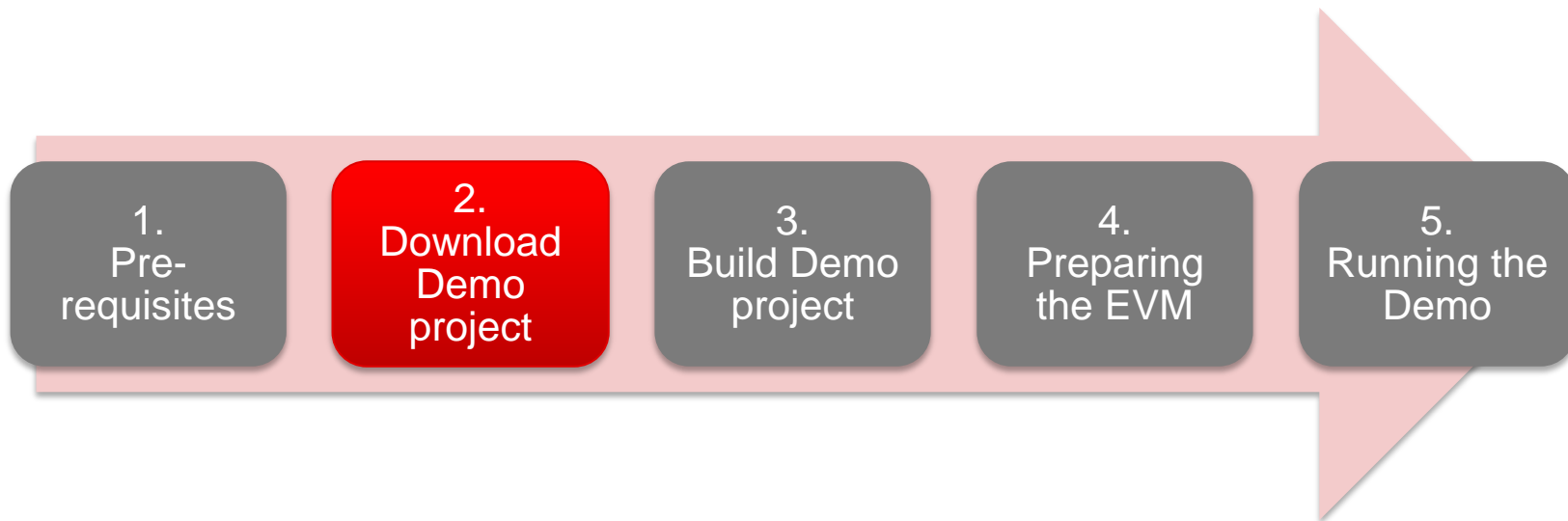
4

5

- It is assumed that you have the mmWave SDK specified in Lab release notes and all related dependencies installed as mentioned in the mmWave SDK release notes.
 - The mmWave SDK release notes include the links for downloading the required tools.
 - Helpful Tips
 - Beginning with SDK 1.1.0.2, the mmwave SDK installer automatically installs the correct versions of the required tools (except CCS)
 - Beginning with SDK 1.1.0.2, PERL and crc.pm are no longer required
- If you have already installed the mmWave SDK and all the required tools, you can move on to the next step i.e. downloading the lab on to your machine.

Tool	Version	Download Link
mmWave SDK	2.0.0.4	download link
CCS	7.4 or later	download link
TI SYS/BIOS	6.53.02.00	Included in mmwave sdk installer
TI ARM Compiler	16.9.6.LTS	Included in mmwave sdk installer
TI CGT Compiler	8.1.3	Included in mmwave sdk installer
XDC	3.50.04.43	Included in mmwave sdk installer
C64x+ DSPLIB	3.4.0.0	Included in mmwave sdk installer
C674x DSPLIB	3.4.0.0	Included in mmwave sdk installer
C674x MATHLIB (little-endian, elf/coff format)	3.1.2.1	Included in mmwave sdk installer
mmwave Radar device support packages	1.5.9 or later	Upgrade to the latest using CCS update process (see SDK user guide for more details)
TI Emulators package	7.0.188.0 or later	Upgrade to the latest using CCS update process (see SDK user guide for more details)
Uniflash	latest	Uniflash tool is used for flashing xWR1xxx devices Cloud version (Recommended): https://dev.ti.com/uniflash Offline version: http://www.ti.com/tool/uniflash
mmWave Demo Visualizer	latest	TI Gallery APP for configuring mmWave sensors and visualizing the point cloud objects generated by the mmWave SDK demo https://dev.ti.com/mmWaveDemoVisualizer

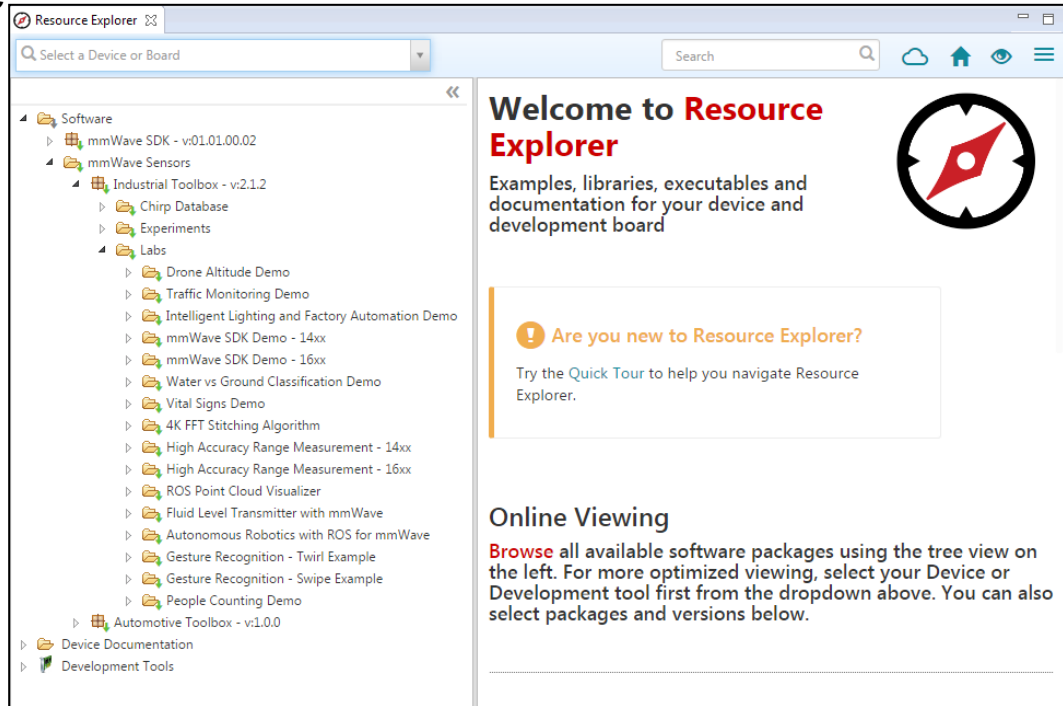
Steps



2. Download the Lab project




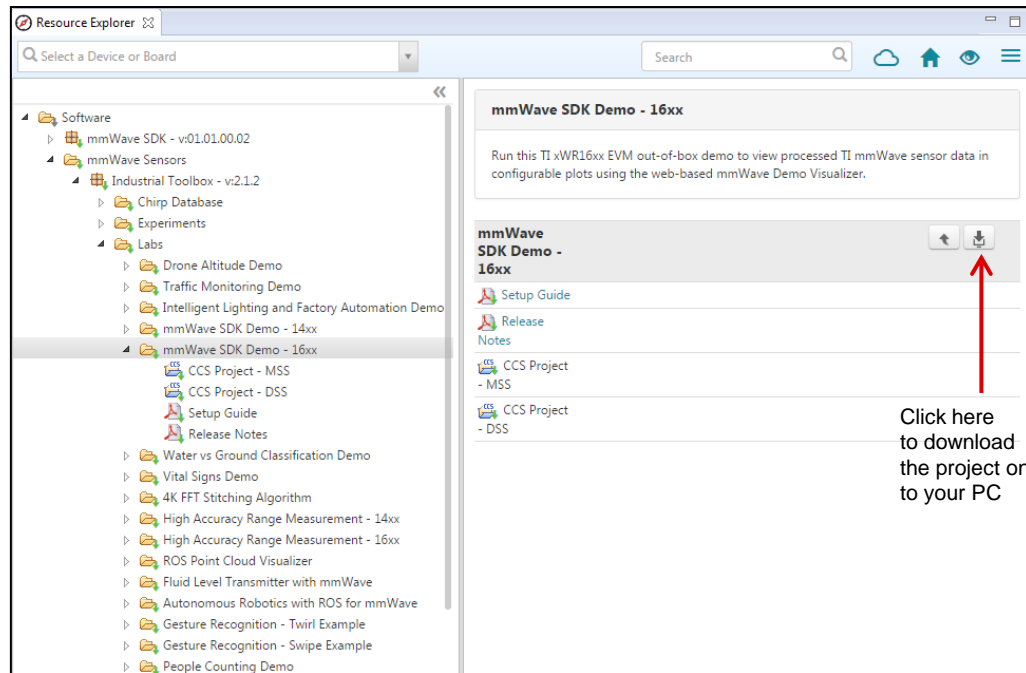
- The mmWave projects are available under **mmWave Sensors** ► **Industrial Toolbox** in CCS Resource Explorer.
- To download the mmWave demo, start CCS v7.2 (or later) and select **View** ► **Resource Explorer** to open the Resource Explorer.
- In the Resource Explorer Window, select **Software** ► **mmWave Sensors** ► **Industrial toolbox** ► **Labs**.



2. Download - continued




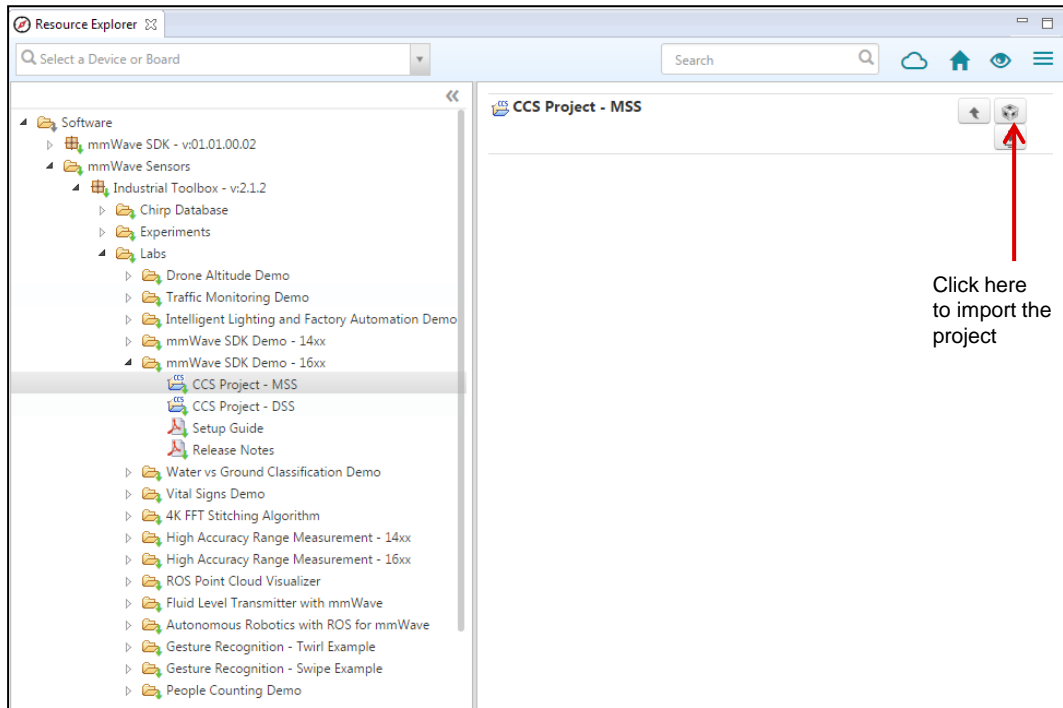
- Select the **16xx mmWave SDK** demo in the left view.
- The right view shows the contents of the Lab which contains the CCS Project and the PC GUI.
- Click on the **Download and Install** button  in the top right corner as shown.
- Select the **Make Available Offline** option from the drop down to start downloading the Lab.



2. Download - continued

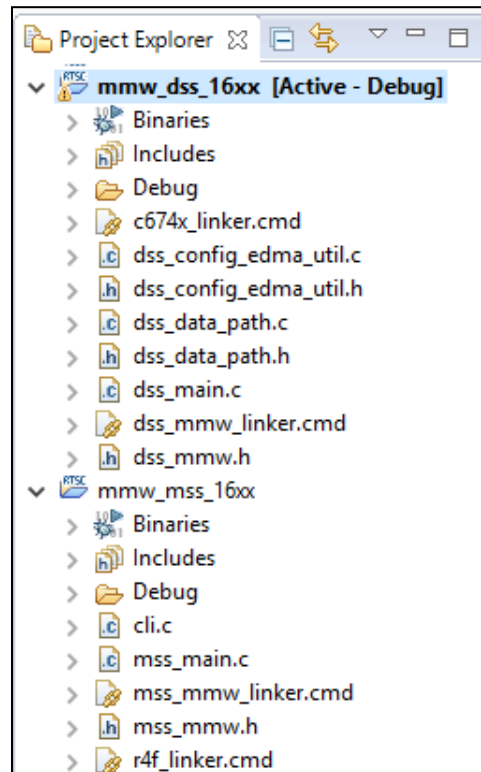
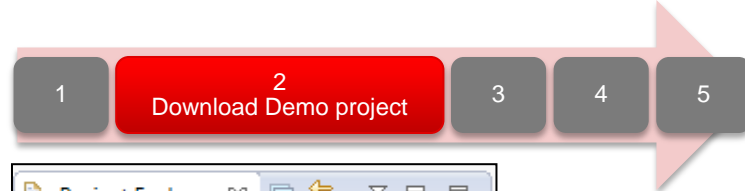


- The xWR16xx mmWave Demo consists of two CCS projects, one for the R4F core and one for the C674x DSP core
- The project will be downloaded in C:\ti\mmwave_industrial_toolbox
- Select the CCS Project MSS file in the left view
- Click on the **Import to IDE**  button which should be visible in the right side view after a successful download.
- This copies the project in the user's workspace and imports it into the CCS project explorer.
 - It is important to note that the copy created in the workspace is the one that gets imported in CCS. The original project downloaded in mmwave_industrial_toolbox is not modified.
- Repeat with the CCS Project DSS file

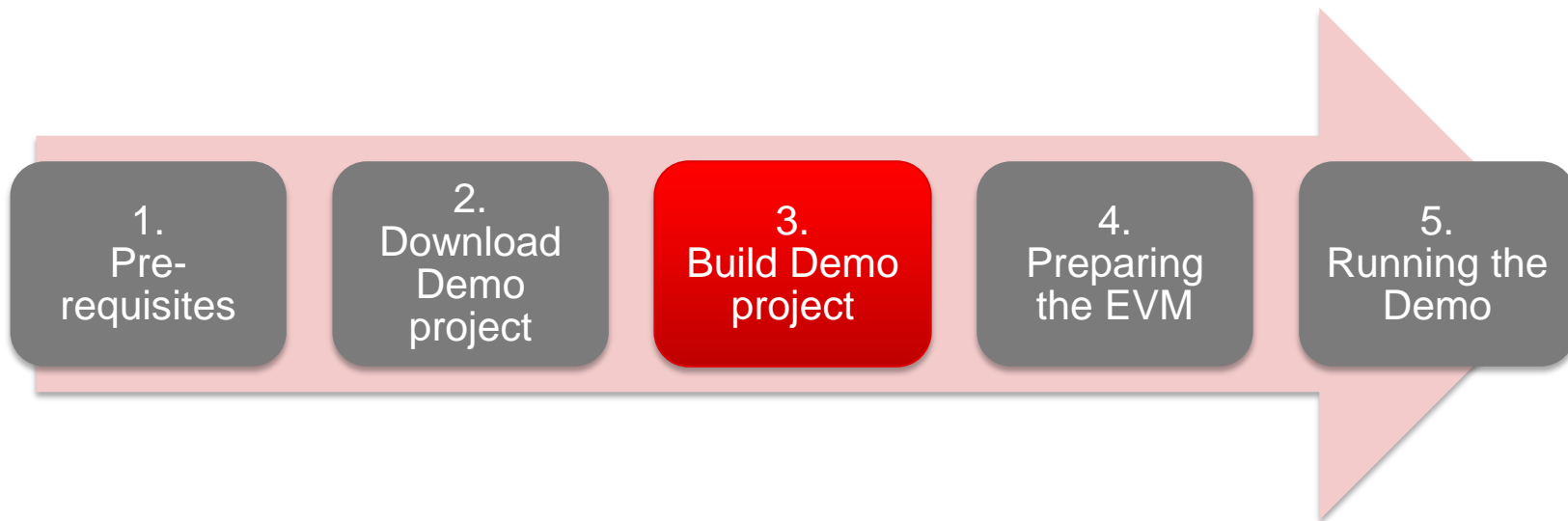


2. Download - continued

- After successfully completing the **Import to IDE** operation, the both projects should be visible in CCS Project Explorer as shown here.
- At this point, we have successfully downloaded the mmWave demo and imported it in CCS.
- We are ready to move on to the next step i.e. Building the projects.

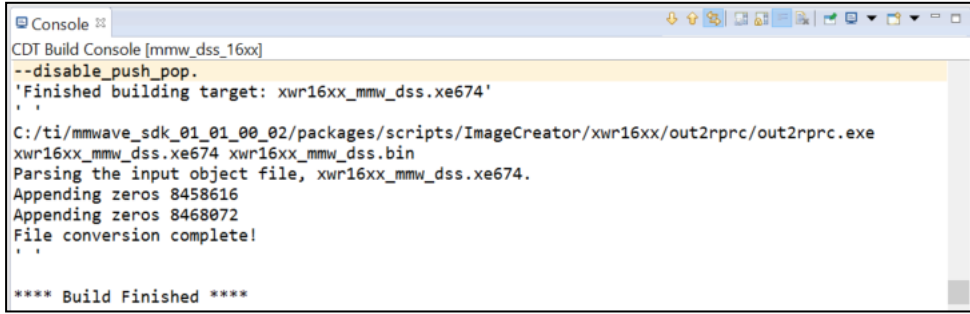
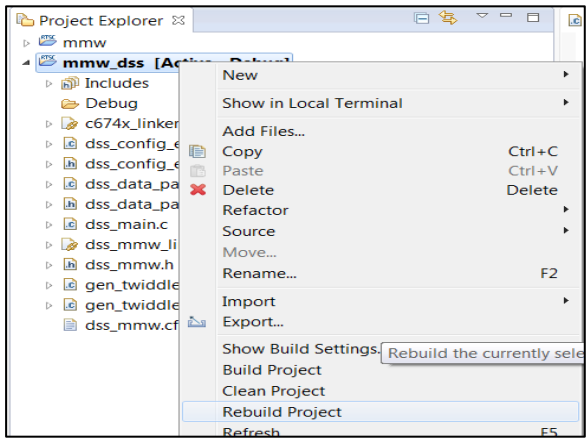


Steps



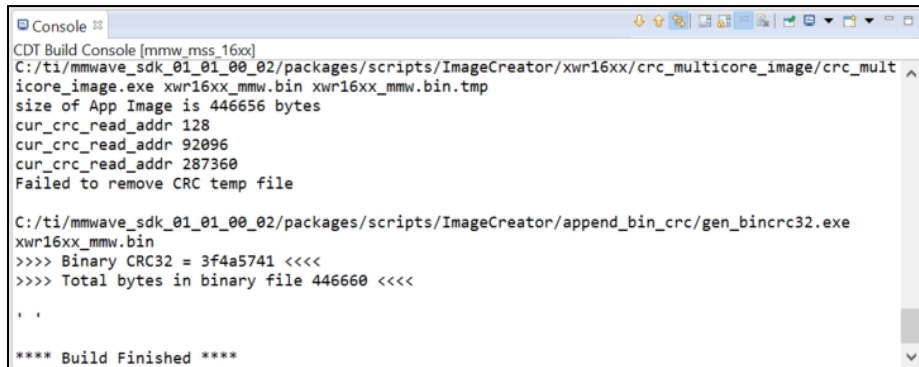
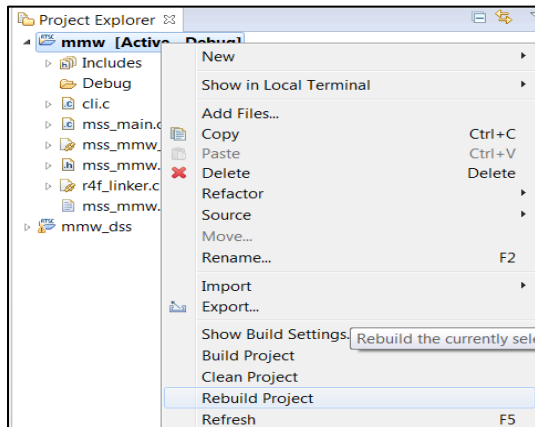
3. Build the Lab

- With the mmw_dss_16xx project selected in Project Explorer, right click on the project and select **Rebuild Project**.
 - Selecting **Rebuild** instead of **Build** ensures that the project is always re-compiled. This is especially important in case the previous build failed with errors.
- On successful completion of the build, you should see the output in CCS console as shown here and the following two files should be produced in the project debug directory
 - xwr16xx_mmw_dss.xe674
 - xwr16xx_mmw_dss.bin (note, this image is not flashed directly. It is merged into a combined meta image when building the MSS; shown on the next page).
- If the build fails with errors, please ensure that all the pre-requisites are installed as mentioned in the mmWave SDK release notes.
 - Please note that pre-built binary files are provided with the demo under mmwave SDK.
 - Look under <mmwave_sdk_install_dir>\packages\ti\demo\xwr16xx\mmw

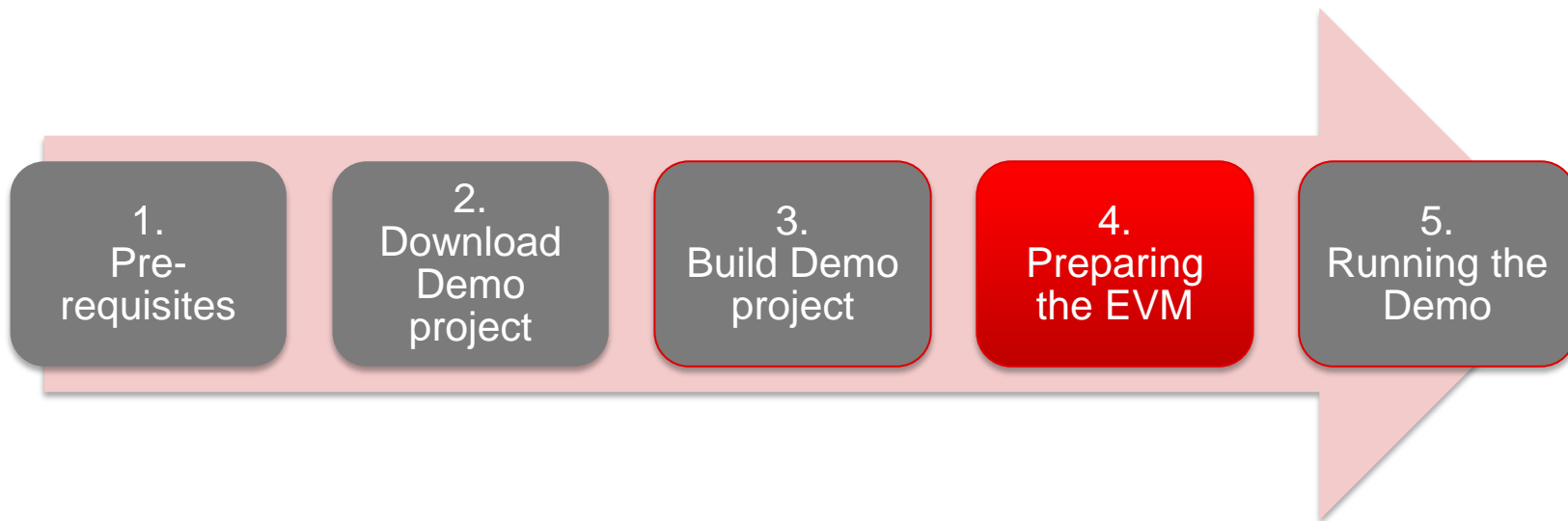


3. Build the Lab

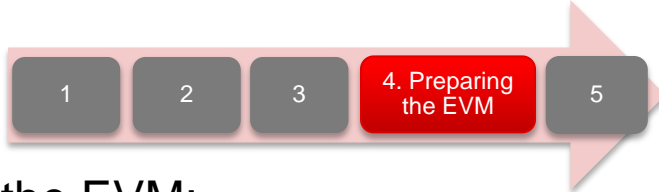
- The mmw_dss_16xx project must be built BEFORE the mmw_mss_16xx project is built.
- With the mmw_mss_16xx project selected in Project Explorer, right click on the project and select **Rebuild Project**.
- On successful completion of the MSS build, you should see the output in CCS console as shown here and the following three files should be produced in the project debug directory
 - xwr16xx_mmw_mss.xer4f
 - xwr16xx_mmw_mss.bin
 - xwr16xx_mmw.bin (this is the meta image to be flashed)
- If the build fails with errors, please ensure that all the pre-requisites are installed as mentioned in the mmWave SDK release notes.
 - Please note that pre-built binary files are provided with the demo under mmwave SDK.
 - Look under <mmwave_sdk_install_dir>\packages\ti\demo\xwr16xx\mmw



Steps



4.1 Preparing the EVM



- There are two ways to execute the compiled code on the EVM:
 - Deployment mode: Flashing the binary (.bin image) on to the EVM serial flash
 - In this mode, the EVM boots autonomously from flash and starts running the bin image.
 - Debug mode: Downloading and running the executable (.xer4f image and .xe674) from CCS.
 - You will need to flash a small CCS debug firmware on the EVM (one time) to allow connecting with CCS. This debug firmware image is provided with the mmWave SDK.
 - As a recap, the build process in Step 3 produces the .bin .xer4f and .xe674 images.
- This presentation explains the second method i.e. Debug mode (CCS).
 - To prepare the EVM for debug mode, we start with flashing the CCS debug firmware image.
 - Please note that the same flashing process can be used to flash the Lab binary to run it in deployment mode.

4.2 Connecting to the EVM

1

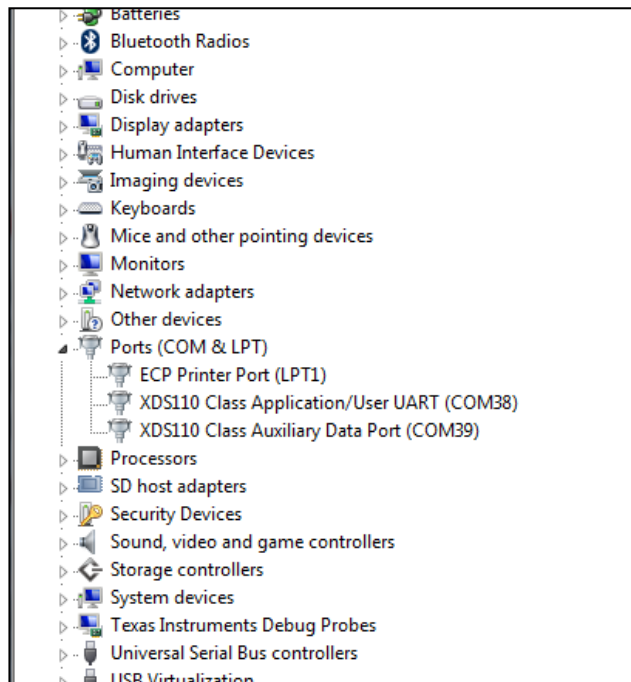
2

3

4. Preparing
the EVM

5

- Power on the EVM using a 5V/2.5A power supply.
- Connect the EVM to your PC and check the COM ports in Windows Device Manager
- The EVM exports two virtual COM ports as shown below:
 - XDS110 Class Application/User UART (COM_{UART}):
 - Used for passing configuration data and firmware to the EVM
 - XDS110 Class Auxiliary Data Port (COM_{AUX})
 - Used to send processed radar data output
- Note the COM_{UART} and COM_{AUX} port numbers, as they will be used later for flashing and running the Lab.



COM_{UART} : COM38 COM_{AUX} : COM39

- The actual port numbers on your machine may be different

4.3 Flashing CCS debug firmware

1

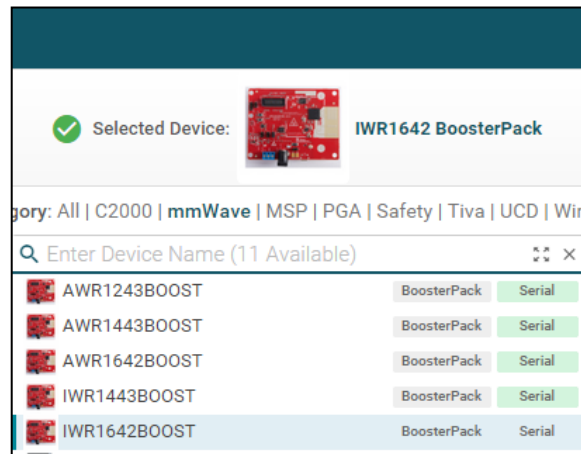
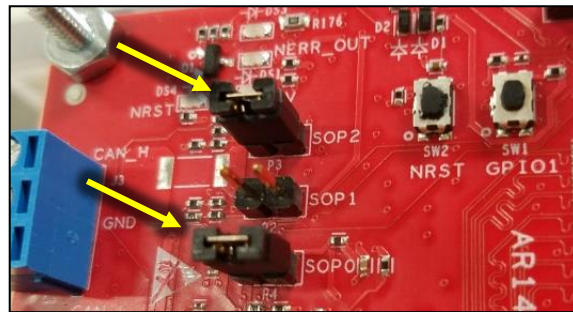
2

3

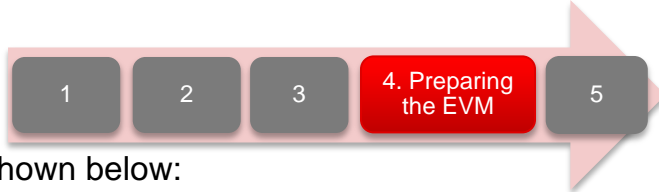
4. Preparing the EVM

5

1. Put the EVM in flashing mode by connecting jumpers on SOP0 and SOP2 as shown in the image.
2. Open the **UniFlash** tool
3. In the **New Configuration** section, locate and select IWR1642BOOST
4. Click **Start** to proceed



4.3 Flashing CCS debug firmware



5. In the **Program** tab, browse and locate the Radar SS and MSS images shown below:

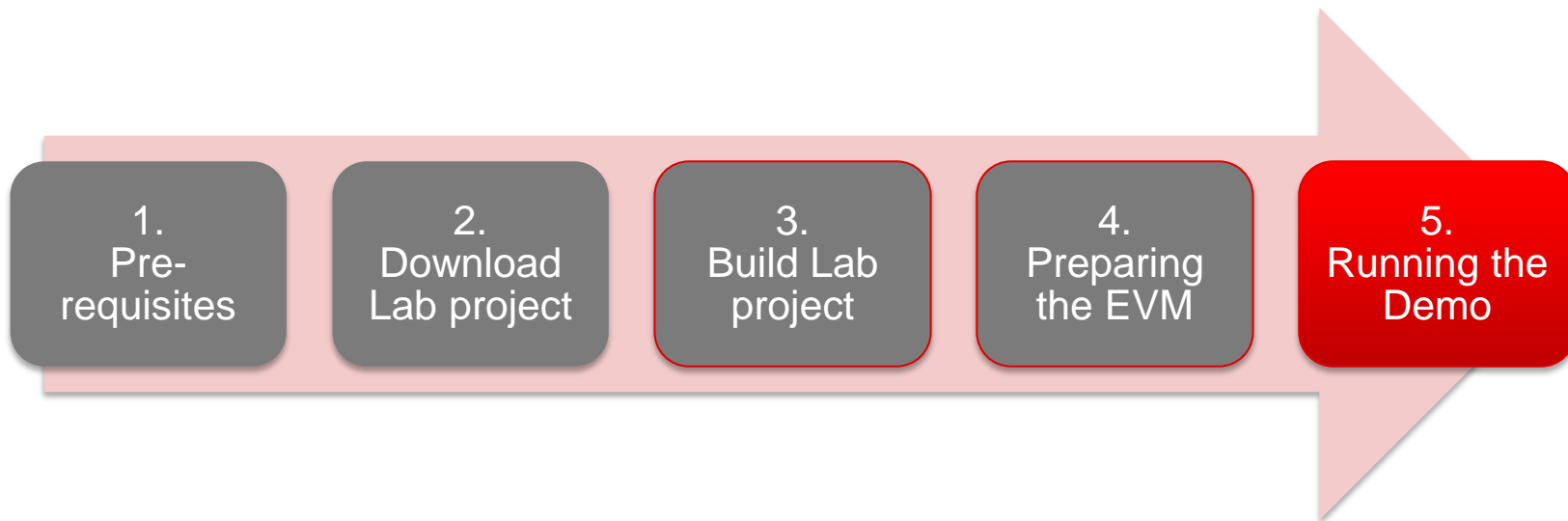
Image	Location
Meta Image 1	C:\ti\mmwave_sdk_<ver>\packages\ti\utils\ccsdebug

6. In the **Settings & Utilities** tab, fill the **COM Port** text box with the Application/User UART COM port number (**COM_{UART}**) noted earlier

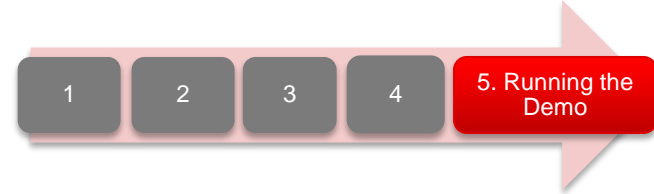


7. Return to the **Program** tab, power cycle the device and click on **Load Images**
8. When the flash procedure completes, UniFlash's console should indicate: [SUCCESS] Program Load completed successfully
9. Power off the board and remove the jumper from only header **SOP2**. Power the board back on (this puts the board back in functional mode)

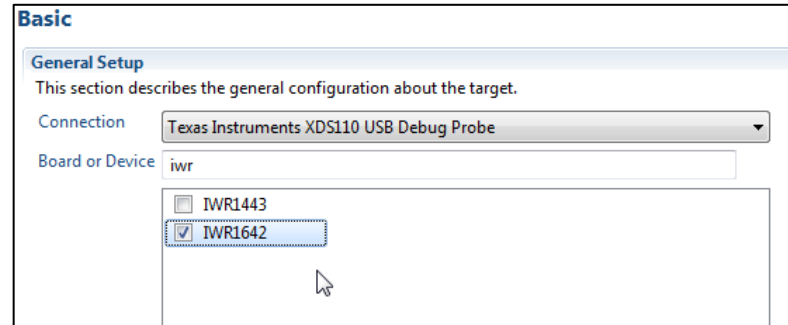
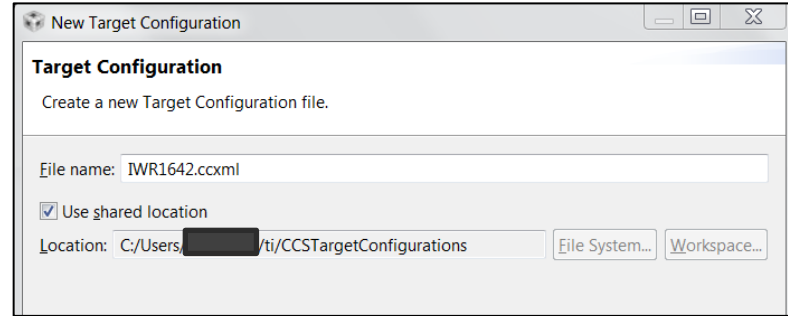
Steps





5.1 Connecting EVM to CCS

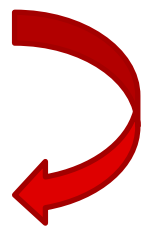
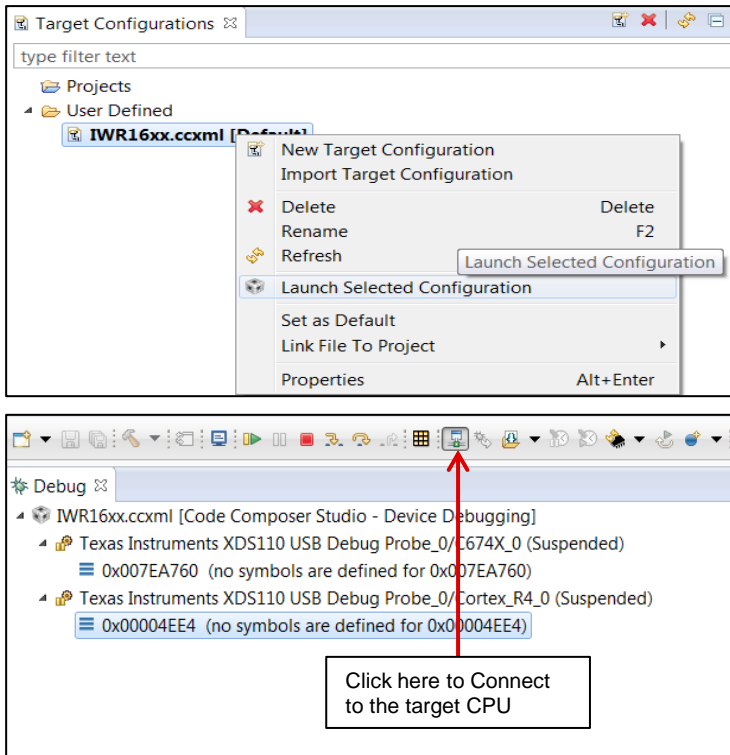


- It is assumed that you were able to download and build the Lab in CCS (completed steps 1, 2 and 3)
- To connect the Radar EVM to CCS, we need to create a target configuration
 - Go to File ► New ► New Target Configuration File
 - Name the target configuration accordingly and check the “Use shared location” checkbox. Press Finish
 - In the configuration editor window:
 - Select “Texas Instruments XDS110 USB Debug Probe” for **Connection**
 - Select **IWR1642** or **AWR1642** in the **Board or Device** list
 - Press the **Save** button to save the target configuration.
 - You can press the **Test Connection** button to check the connection with the board.



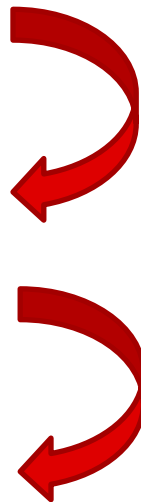
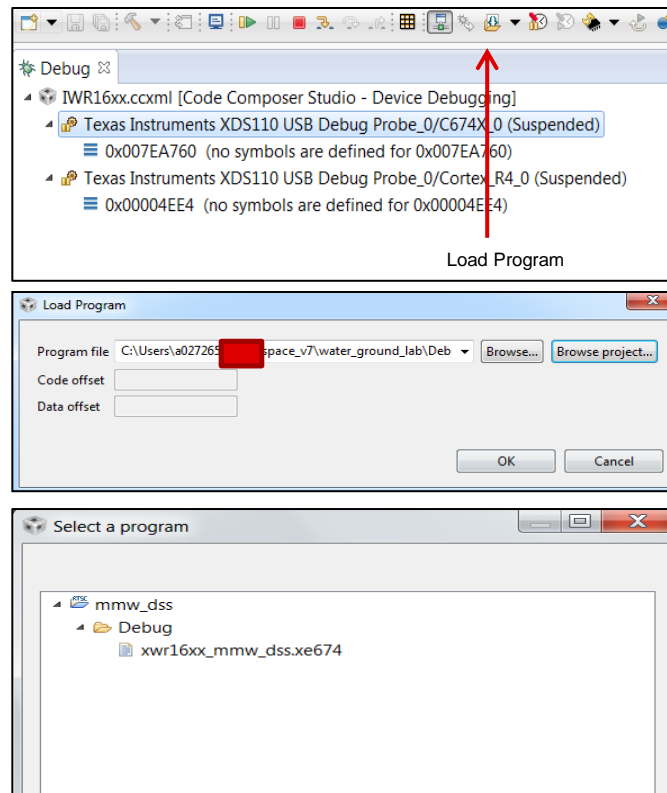
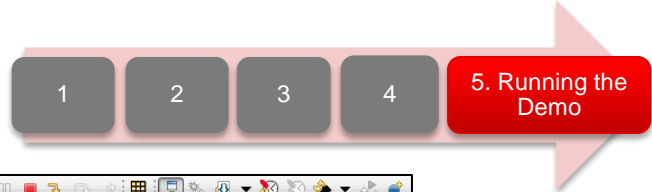
5.1 Connecting - continued

- Go to **View ► Target Configurations** to open the target configuration window.
- You should see your target configuration under **User Defined** configurations.
- With the board powered on, right click on the target configuration and select **Launch Select Configuration**.
- This will launch the target configuration in the debug window.
- Select the Texas Instruments XDS110 USB Debug probe/C674X_0 and press the **Connect Target** button 
- Select the Texas Instruments XDS110 USB Debug probe/Cortex_R4_0 and press the **Connect Target** button 



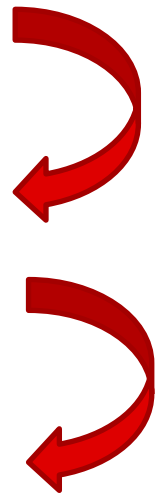
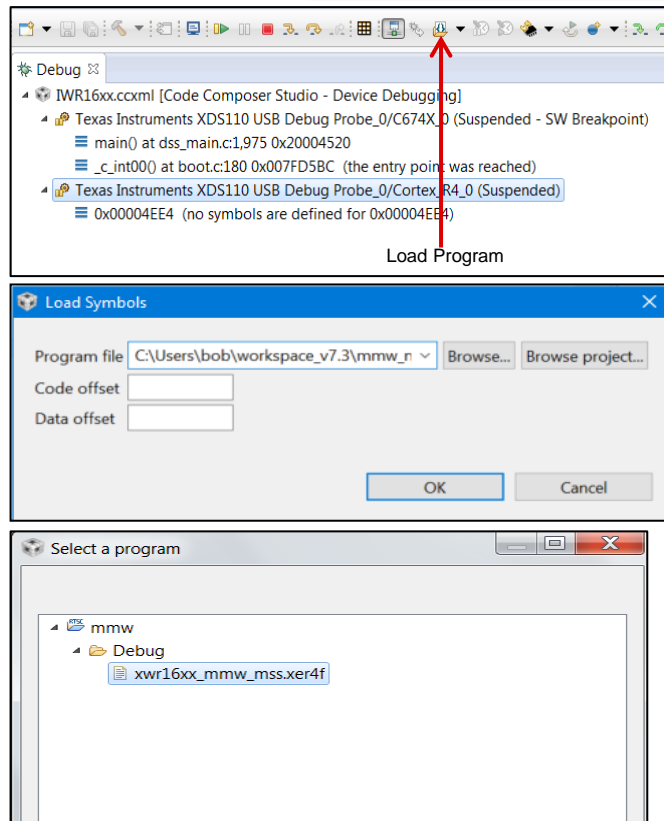
5.2 Loading the binary

- Once both targets are connected, select the C674X_0 target, and click on the **Load** button in the toolbar.
- In the **Load Program** dialog, press the **Browse Project** button .
- Select the lab executable (.xe674) found in the mmw_dss project as shown, and press OK.
- Press OK again in the **Load Program** dialog.

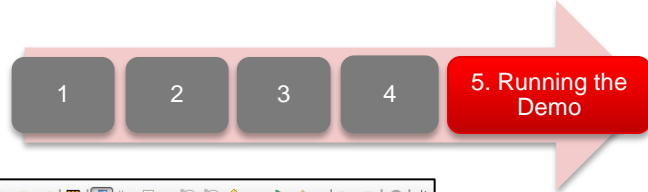



5.2 Loading the binary

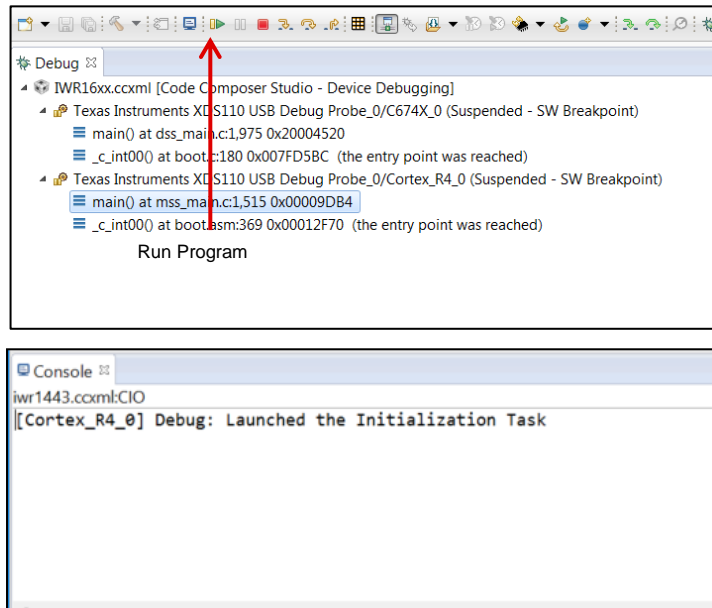
- Now select the Cortex_R4_0 target, and click on the **Load** button in the toolbar.
- In the **Load Program** dialog, press the **Browse Project** button .
- Select the lab executable (.xer4f) found in the mmw project as shown, and press OK.
- Press OK again in the **Load Program** dialog.



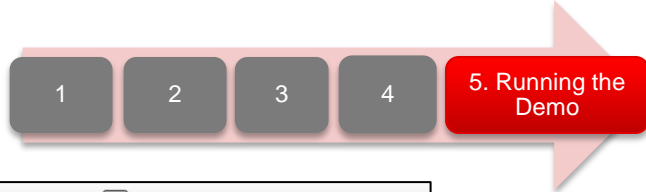
5.3 Running the binary




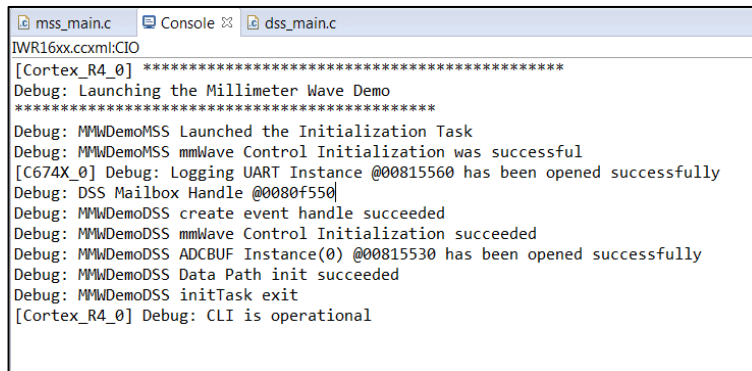
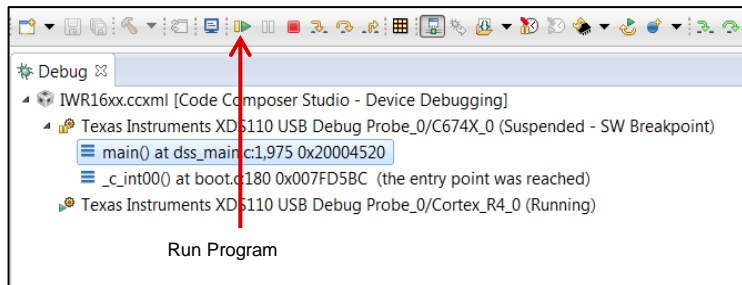
- With both executables loaded, select `mss_main.c`, as shown, and press the Run/Resume button 
- The program should start executing and generate console output as shown.



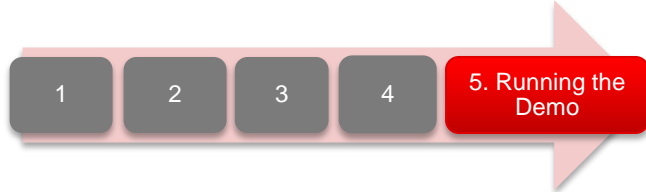
5.3 Running the binary



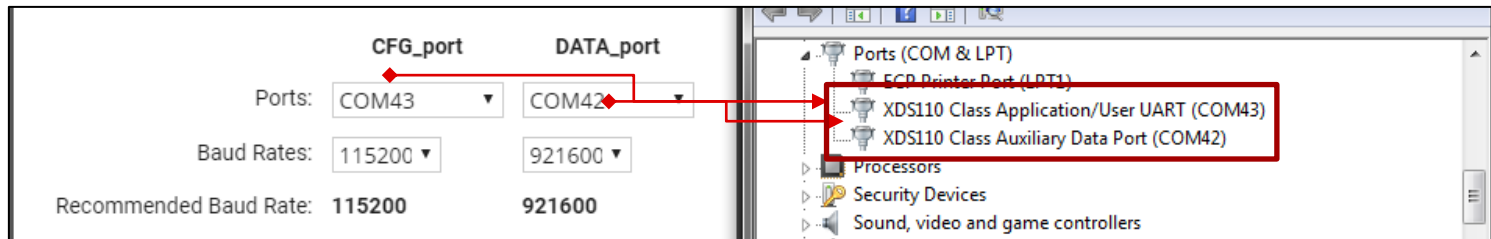
- Select `dss_main.c`, as shown, and press the Run/Resume button 
- Further console output should be generated as shown.
- You should see the “CLI is operational” message which indicates that the program is ready and waiting for the sensor configuration
- The sensor configuration is sent using the web GUI



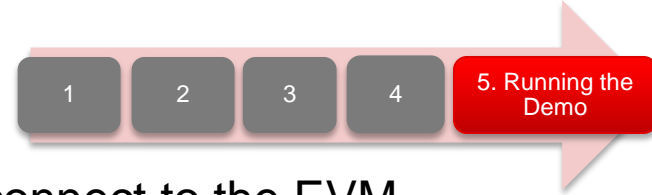
5.4 Running the Lab GUI





1. Using Google Chrome, navigate to the following URL:
<https://dev.ti.com/mmWaveDemoVisualizer>
 - Alternatively, go to <https://dev.ti.com/gallery> and search for “mmWave Demo Visualizer”
2. If prompted, follow the on-screen instructions for installing [TI Cloud Agent](#)
3. Once the demo is loaded, go to **Options** → **Serial Port**
4. In the serial port window, enter the appropriate port in each of the drop down menus based on your port numbers from Step 2



5.4 Running GUI - continued



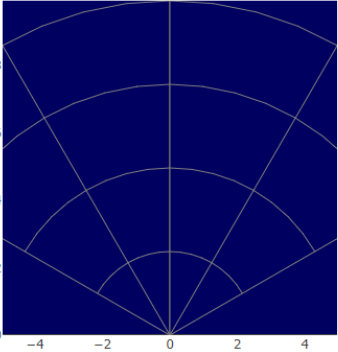
5. Click on **Configure** and the demo will automatically connect to the EVM
 - Not connected:  Connected: 
 - If the connection fails, try clicking on the connection icon in the bottom left corner
6. Select the appropriate mmWave device from the **Platform** dropdown menu
7. Use the available options to create the desired configuration
 - Additional details about the configuration parameters can be found in the [mmWave Demo Visualizer User Guide](#)
8. When ready to send the configuration, click on **Send Config To mmWave Device**
9. Click on the **Plots** tab to view the plots that were selected to be shown
10. Move a highly reflective object in front of the EVM and see how the demo responds

5.4 Running GUI - continued

mmWave Demo Visualizer | Options Help

Configure Plots

X-Y Scatter Plot



Distance along lateral axis (meters)

Range Profile for zero Doppler



Relative Power (dB)

Range (meters)

Range Depth: 9.93 Range Width: 4.96

Range Profile Y max: 2e6 Range Profile Log Scale

File Size Max (MB): 1

Record time max (s): 10

SENSOR STOP RECORD START


LOAD CONFIG FROM PC AND SEND

Display Parameters Profiling


Platform

SDK Version

Doppler-Range Plot



Active and Interframe CPU (R4F) load



Active frame Interframe

Powered By GUI Co

COM4:115200, COM6:921600 Hardware Connected. TEXAS INSTRUMENTS 29

Learn more about TI mmWave Sensors

- Learn more about xWR1x devices, please visit the product pages
 - IWR1443: <http://www.ti.com/product/IWR1443>
 - IWR1642: <http://www.ti.com/product/IWR1642>
 - AWR1443: <http://www.ti.com/product/AWR1443>
 - AWR1642: <http://www.ti.com/product/AWR1642>
- Get started evaluating the platform with xWR1x EVMs, purchase EVM at
 - IWR1443 EVM: <http://www.ti.com/tool/IWR1443BOOST>
 - IWR1642 EVM: <http://www.ti.com/tool/IWR1642BOOST>
 - AWR1443 EVM: <http://www.ti.com/tool/AWR1443BOOST>
 - AWR1642 EVM: <http://www.ti.com/tool/AWR1642BOOST>
- Download mmWave SDK @ <http://www.ti.com/tool/MMWAVE-SDK>
- Ask question on TI's E2E forum mmWave Sensors forum @ https://e2e.ti.com/support/sensor/mmwave_sensors/



© Copyright 2017 Texas Instruments Incorporated. All rights reserved.

This material is provided strictly “as-is,” for informational purposes only, and without any warranty.
Use of this material is subject to TI’s **Terms of Use**, viewable at TI.com