TI mmWave Labs

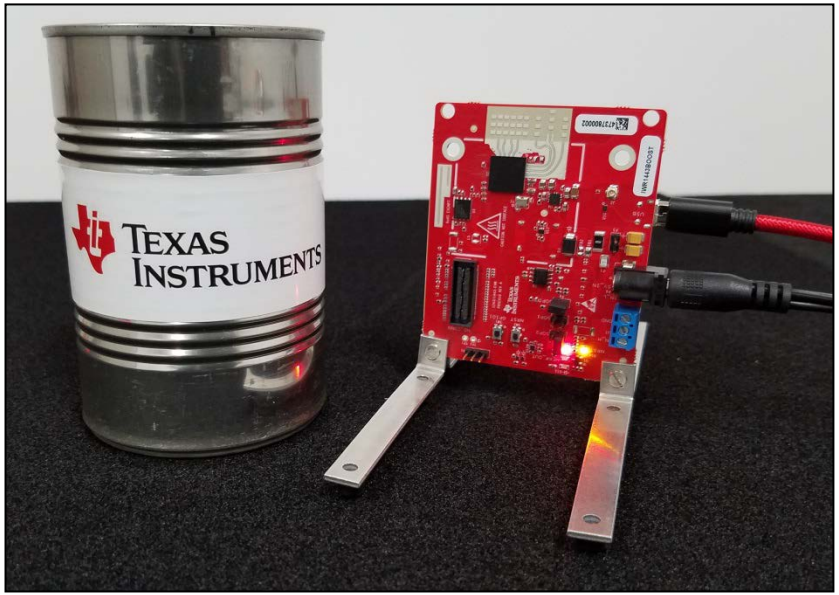
High Accuracy Range Measurement – 16xx

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

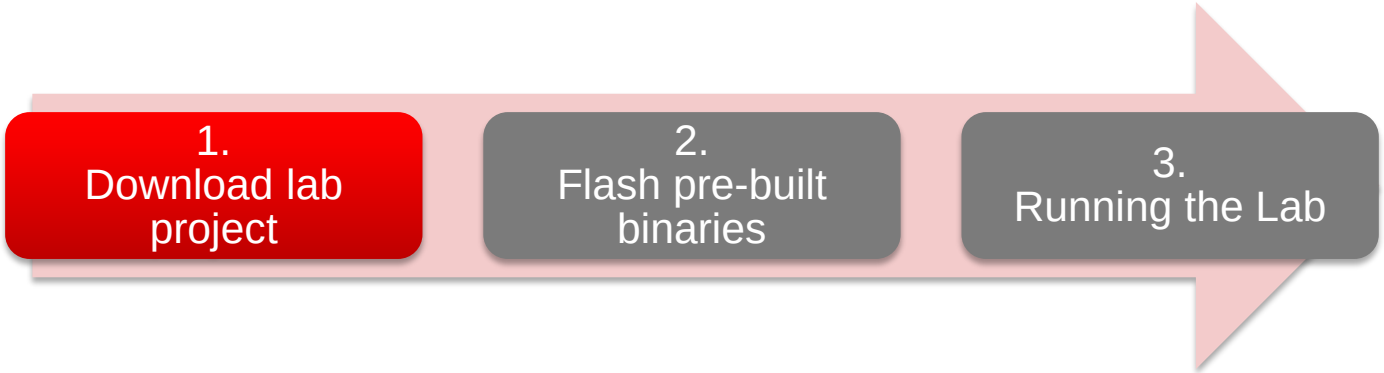
Contents

Lab Overview

- This lab demonstrates the implementation of zoom FFT to detect object within sub-mm accuracy.
- The mmWave sensor xwr1642 EVM is used for lab demonstration
- The lab demonstrates the capability of detecting single peak within range specified by user input
- Accuracy <0.1mm can be achieved with detection SNR ~57dB
- GUI visualizer tool for lab demonstration



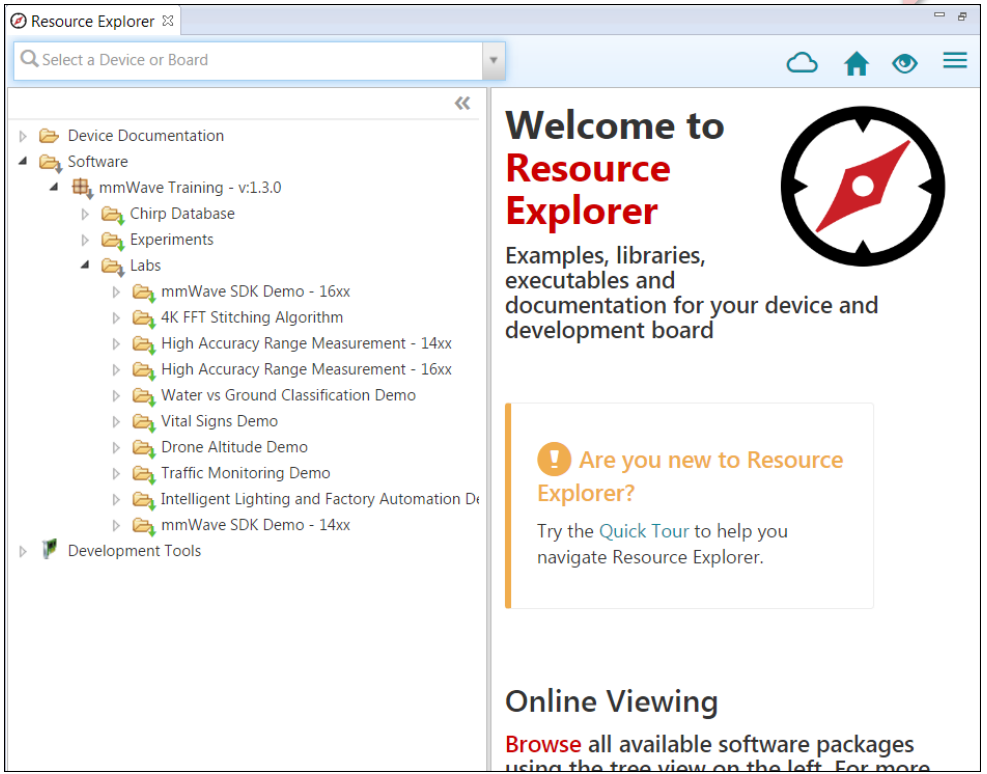
Quick Start Steps



- Required Hardware
  - xWR1642 EVM
  - Micro USB cable (included in the EVM package)
  - 5V/2.5A Power Supply
    - Purchase from Digikey

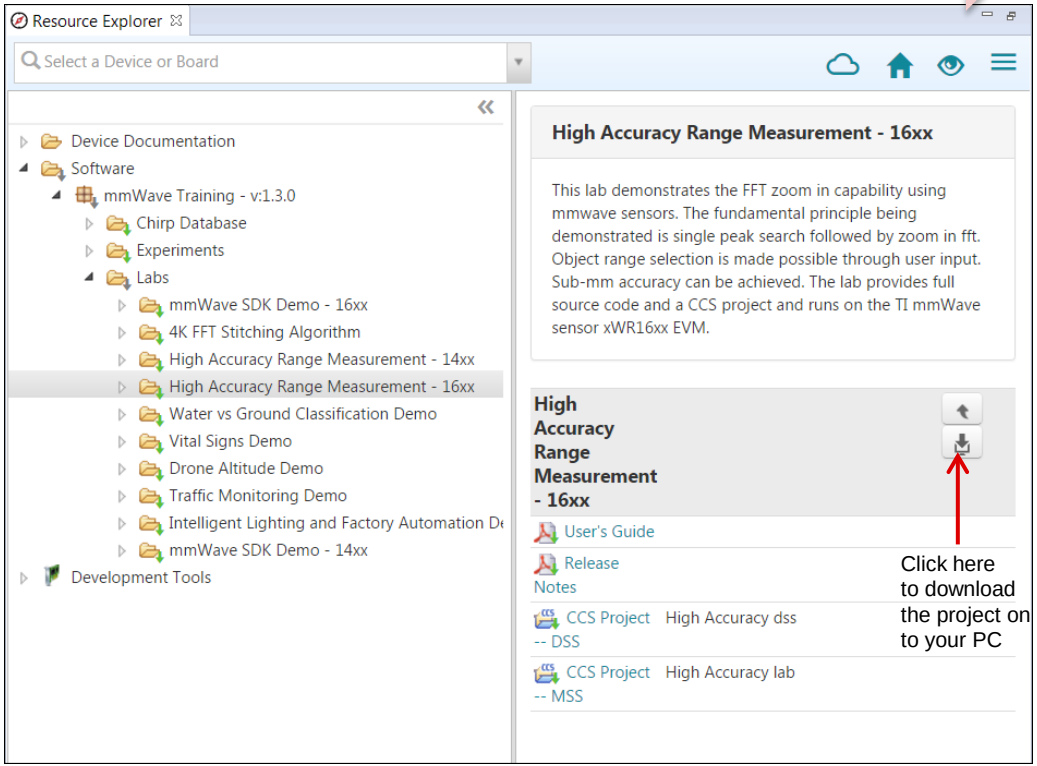
QUICK START 1. Download the lab

- The mmWave Lab projects are available under **mmWave Industrial Toolbox** or **mmWave Training** in CCS Resource Explorer.
- To download the high accuracy range measurement - 16xx Lab, start CCS v7.1 (or later) and select **View ► Resource Explorer** to open the Resource Explorer.
- In the Resource Explorer Window, select **Software ► mmWave Training ► Labs**.

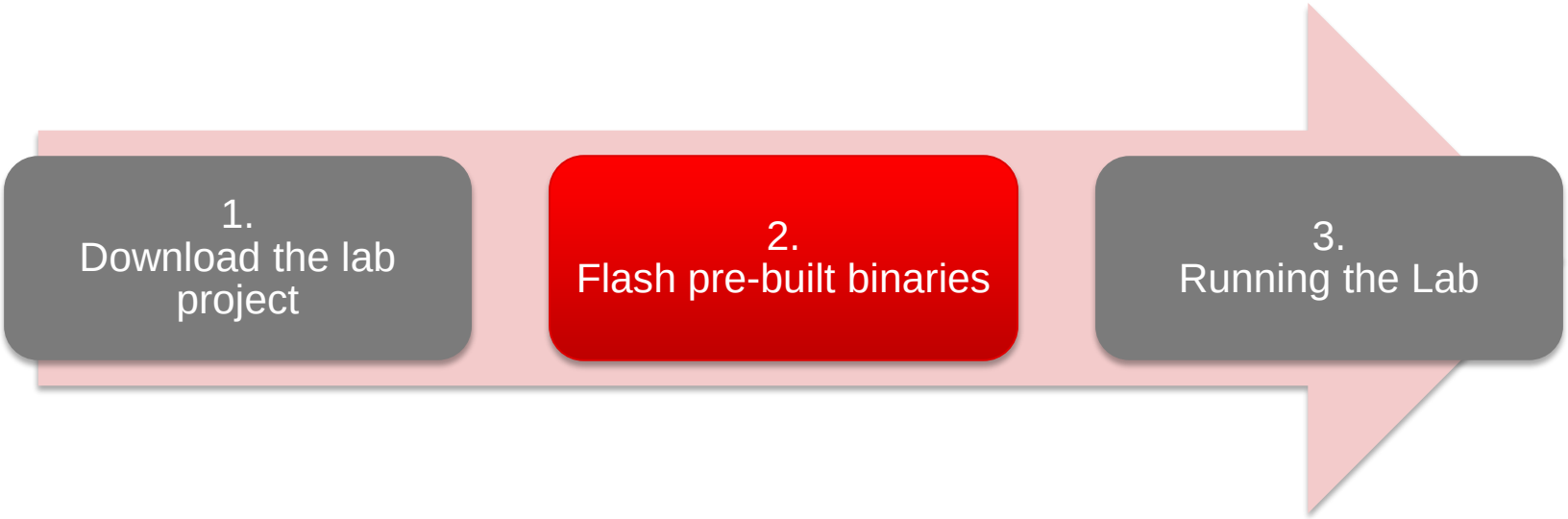


QUICK START 1. Download

- Select the **High Accuracy Range Measurement - 16xx** in the left view.
- The right view shows the contents of the Lab which contains the CCS Project
- 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.

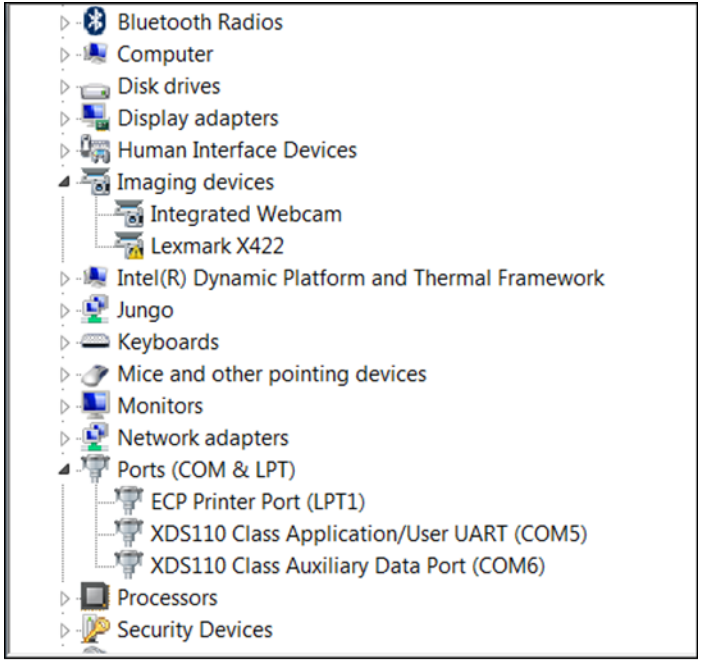


Quick Start Steps



Quick Start 2. Flash the binaries

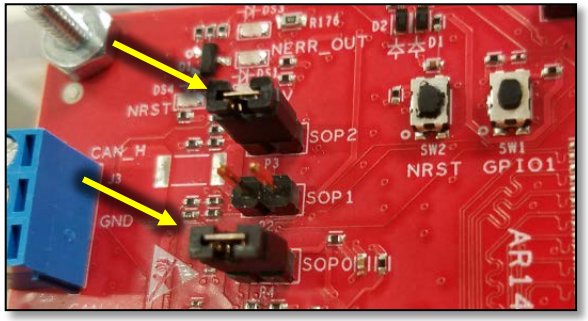
- Power on the EVM using a 5V/2.5A power supply.
- Connect the EVM to your PC and check the COM ports in Windows Device Manager
- The EVM exports two virtual COM ports as shown below:
  - XDS110 Class Application/User UART (COM_{UART}):
    - Used for passing configuration data and firmware to the EVM
  - 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.

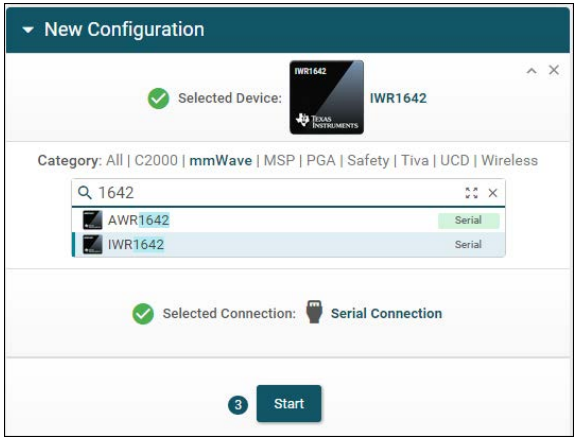


COM_{UART}: COM5 **COM_{AUX}**: COM6 The actual port numbers on your machine may be different

Quick Start 2. Flash – continued

1. Put the EVM in flashing mode by connecting jumpers on SOP0 and SOP2 as shown in the image.
2. Open the **UniFlash** tool
   - Download from TI.com/tool/uniflash
3. In the **New Configuration** section, locate and select the appropriate device (AWR1642 or IWR1642)
4. Click **Start** to proceed





Quick Start 2. Flash – continued

1. In the **Program** tab, browse and locate binary file shown below:

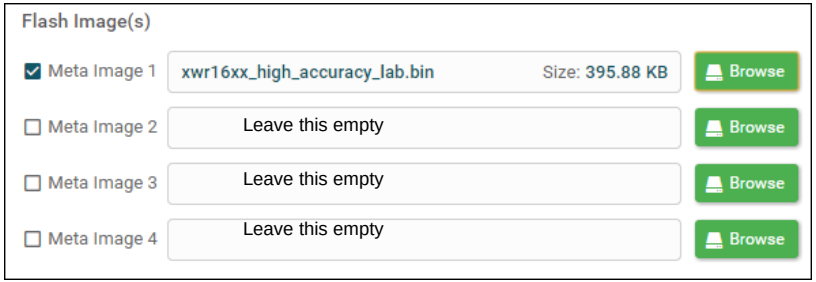
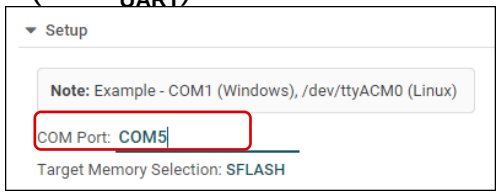


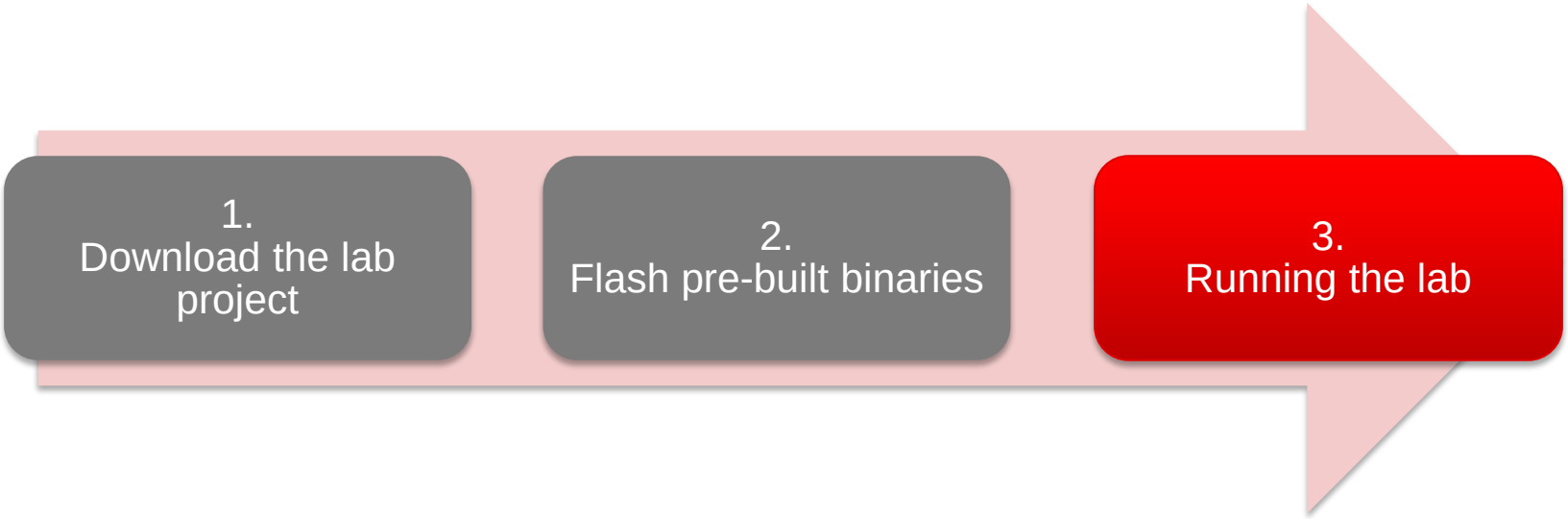
Image	Location
Meta Image 1	C:\ti\mmwave_industrial_toolbox_<ver>\labs\lab0005-high-accuracy-16xx\lab0005_high_accuracy_16xx_pjt\prebuilt_binaries**xwr16xx_high_accuracy_lab.bin**

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



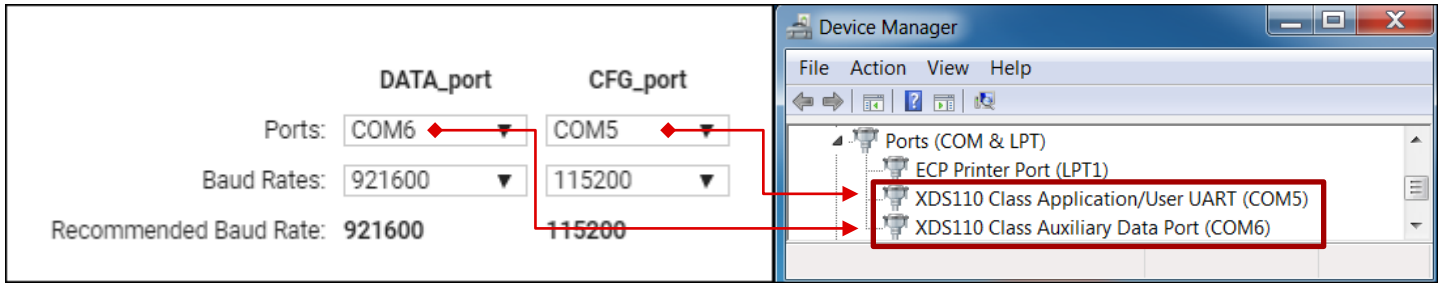
3. Return to the **Program** tab, power cycle the device and click on **Load Images**
4. When the flash procedure completes, UniFlash's console should indicate: [SUCCESS] Program Load completed successfully
5. Power off the board and remove the jumper from only header **SOP2** (this puts the board back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

Quick Start Steps



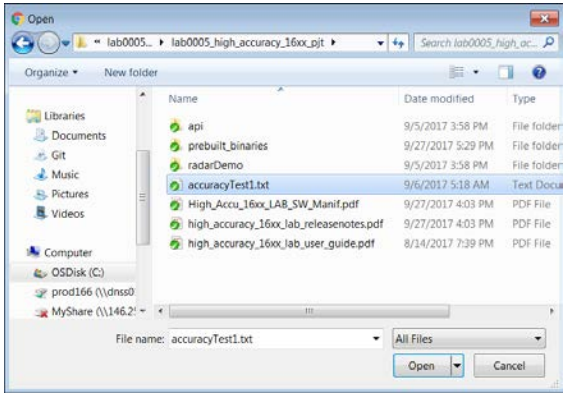
Quick Start 3. Running the Lab GUI

1. Using Google Chrome, navigate to the following URL:
   https://dev.ti.com/gallery/view/1359600/High_Accuracy_Visualizer/
   Alternatively, go to https://dev.ti.com/gallery and search for “High Accuracy 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

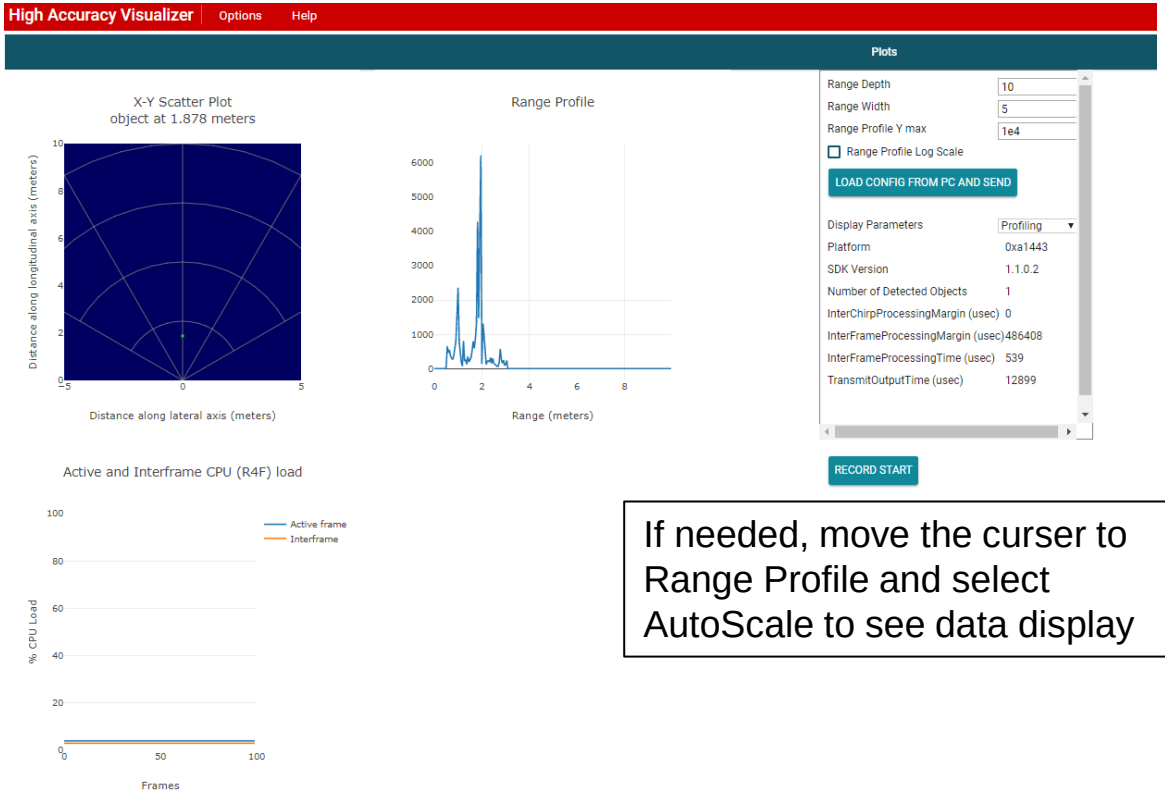


Quick Start 3. GUI - continued

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. Press “LOAD CONFIG FROM PC AND SEND” button and select accuracyTest1.txt under C:\ti\mmwave_industrial_toolbox_<ver>\labs\lab0005-high-accuracy-16xx\lab0005-high-accuracy-16xx_pjt
7. Move a highly reflective object in front of the EVM and see how the demo responds



Quick Start 3. GUI - continued



If needed, move the curser to Range Profile and select AutoScale to see data display

Quick Start 3. Limit the range

8. Use CLI command to disable or enable the range limited function.
9. User can change the parameter to suit their own application.
10. Command is RangeLimitCfg <numRangeBinZoomIn> <dis/enable><min range limited><max range limited>
    - <numRangeBinZoomIn>
      - numRangeBinZoomIn is for one side, with value of 2, the total number of bins of zoom-in is 5
      - <numRangeBinZoomIn> is an optimized number and alteration is not recommended
    - <dis/enable>: disable 0, enable 1
    - <min range limited> in unit of meters
    - <max range limited> in unit of meters
11. Example: RangeLimitCfg 2 1 0.5 3.0
    - Enable the range limited function and the min range is 0.5m, max range is 3.0m

Quick Start 3. Record the result

12. While using GUI to show the detected result, user can use record function to record it to a CSV file.
13. Press “LOAD CONFIG FROM PC AND SEND” button and select the accuracyTest1.txt under C:\ti\mmwave_training_<ver>\labs\lab0005-high-accuracy-16xx\lab0005-high-accuracy-16xx_pjt
14. When the application is running, press “Record Start” button and select the save file name in your disk. Record is running.
15. When stop recording, please press “Record Stop” button to stop it.
16. The .csv file format is shown on the right.

Result Template.csv

Frame	object at(m)
0	1.7998
1	1.8584
2	1.8584
3	1.8576
4	1.8576
5	1.7998
6	1.7998
7	1.8573
8	1.8573
9	1.8552
10	1.8552
11	1.7996
12	1.7996
13	1.7996
14	1.7996
15	1.8004

Zoom FFT Algorithm

- Configuration is designed to have 1 TX and 1 RX, multiple chirps per frame are supported
- Chirp signal is accumulated before coarse range FFT
- A size N FFT can be re-expressed as
  - Step1: N_1 number of size-N_2 FFTs,
  - Step 2: followed by additional twiddle multiplication,
  - Step 3: then N_2 number of size-N_1 FFTs, if N can be factorized as $N = N_1 \times N_2$.
- The following assumptions are made:
  - *N2 = round up to power of 2 of NsamplesPerChirp*
  - Zoom-in factor is the same as N2, meaning N1 = N2

For more details of the algorithm, see link below
https://en.wikipedia.org/wiki/Cooley%E2%80%93Tukey_FFT_algorithm#General_factorizations

Zoom FFT Algorithm – continued

- Memory optimized design algorithm:
  - Only peak object region has zoom FFT of size N, no need to generate complete N=N1*N2 number of twiddle factor
  - Generate 2 sets of twiddle: $\mathrm{fine}Tw$: $e^{-j2\pi k/N}$ and $\mathrm{Co}arseTw$: $e^{-j2\pi k/N_1}$ for *k=0…N_1-1*
  - The basic operation becomes finding the indices to the fineTw array and index to the CoarseTw array, then multiply a set of twiddles from table look-up and then multiply to the input signal.
  - Calculation of twiddle array indices is mainly AND and SHIFT operations
- Memory usage of high accuracy range processing:
  - L2 heap for twiddle factor storage and configuration setting
    - Allocated size is 4 * 2 * N2 * 4 + 52 bytes,
  - L2 Scratch for data
    - Allocated size is 2 * 2 * N2 * 4 bytes

Zoom FFT Code Implementation

- Mss_main.c :
  - Redefines the UART output data format
    - the detected object range is formatted to be 32 bits, with lower 20 bits for fractional part and upper 12 bits for integer part.
- Dss_main.c:
  - Memory initialization
  - Copy ADCbuffer data into L1 Memory, calls for highAccuRangeProcessing
  - Process and send output to mss
- RADARDEMO_highAccuRangeProc_priv.c
  - Chirp signal accumulation
  - Twiddle factor generation
  - Coarse FFT and coarse object detection within user defined range
  - Zoom FFT for fine freq detection

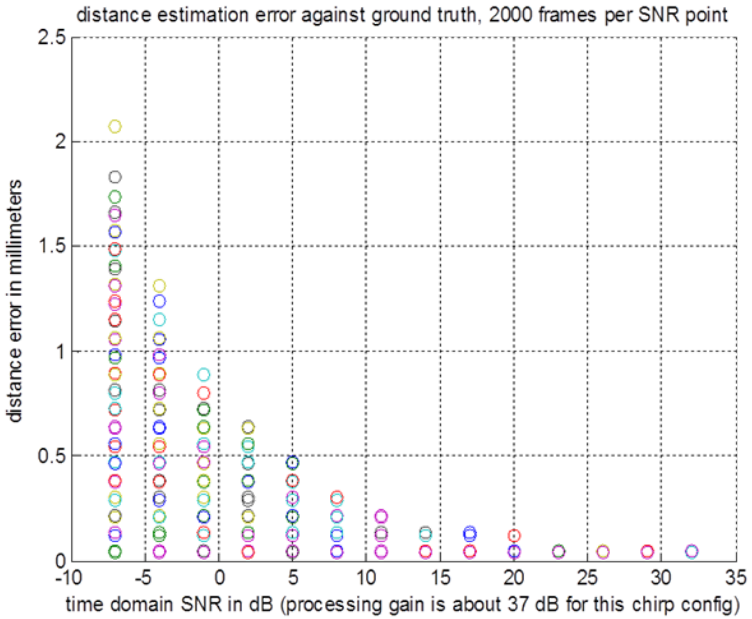
Trade-offs and Considerations

- Comparison of the high accuracy lab 16xx vs. 14xx:
  - Both 16xx and 14xx has the flexibility to select the range where highest peak is for zoom FFT
  - 16xx has much higher accuracy than14xx: 14xx HWA limits the max FFT size of 16K, 16xx process data with DSP, the example given in the lab is 512x512 (256K)
  - Higher accuracy also leads to higher power consumption and longer processing latency. 14xx will have much lower processing latency by using HWA, in addition to lower power consumption

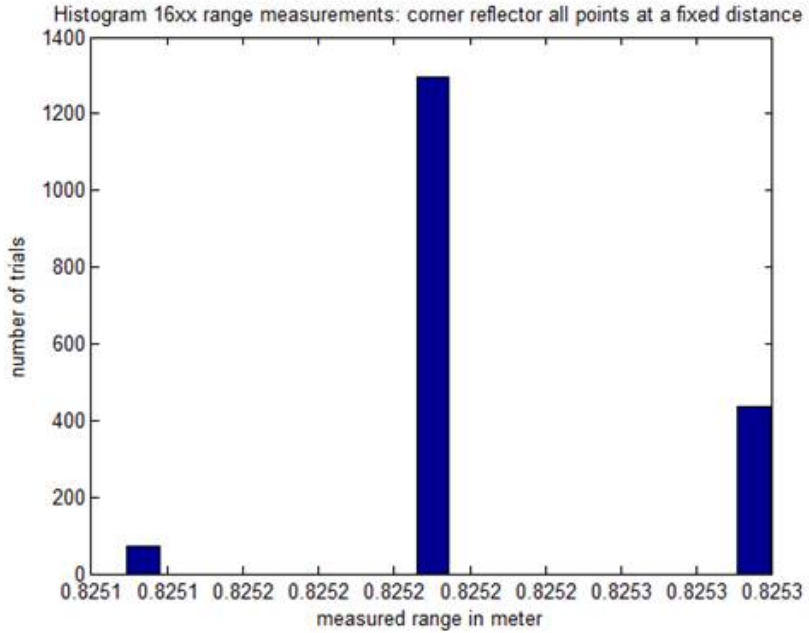
Simulation Result

- Error vs time domain SNR scatter plot
  - Can read of the max error for a given time domain SNR.
  - Processing gain to convert to Detection SNR is approximately 37 dB (10log10(512*10))
- Conclusions
  - We read the following results from the plot:

Target Accuracy	Required Detection SNR
< 0.1 mm	~ 57 dB
<0.5 mm	~ 42 dB
< 1 mm	~ 36 dB

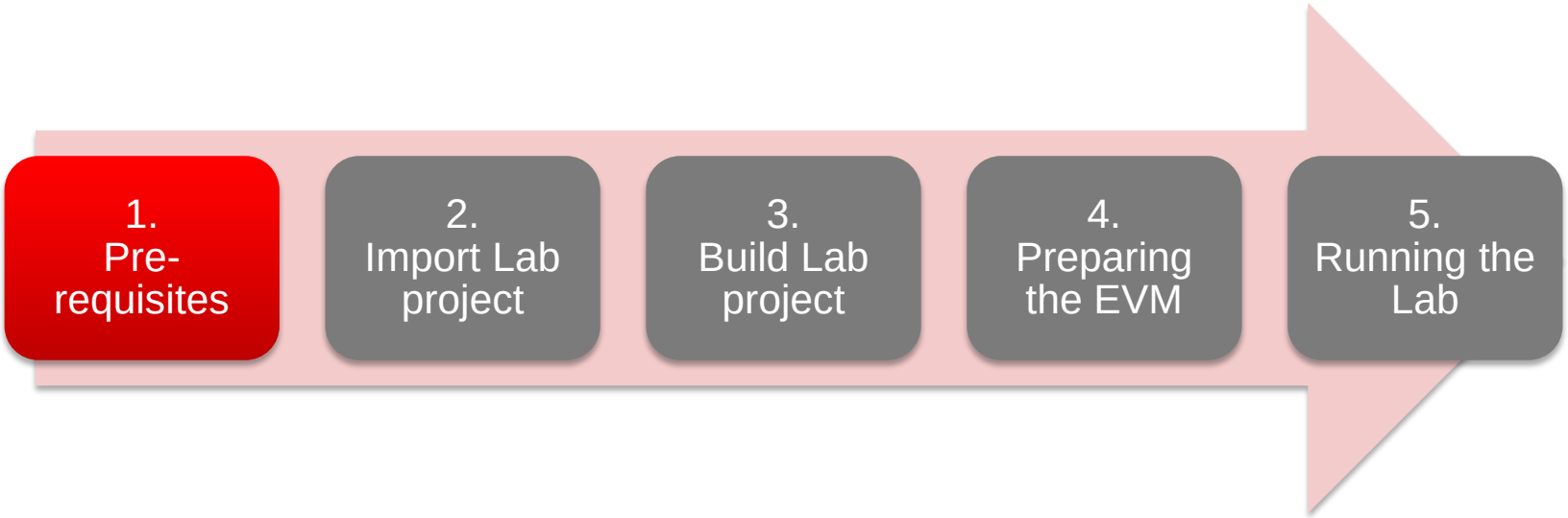


Lab test with corner reflector



- For a stationary target, the measurement is done 1800 times

Steps for Building from the Source Code and Run

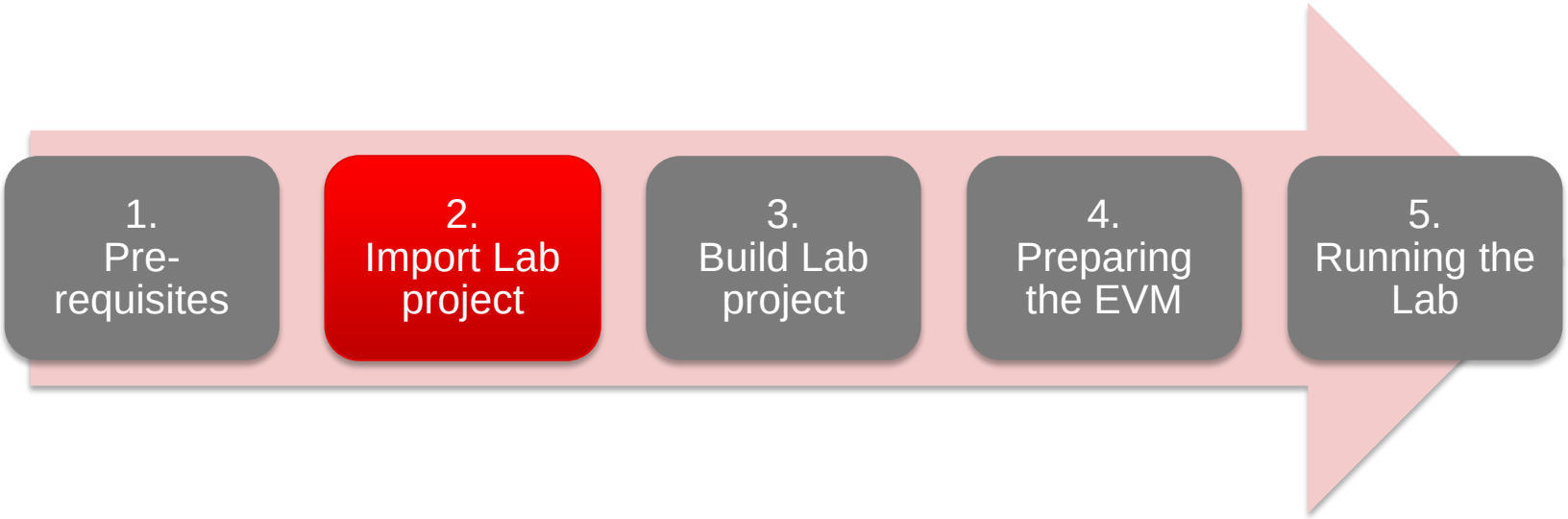


1. Pre-requisites

- It is assumed that you have the TI 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 versions of the above tools.
- 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.

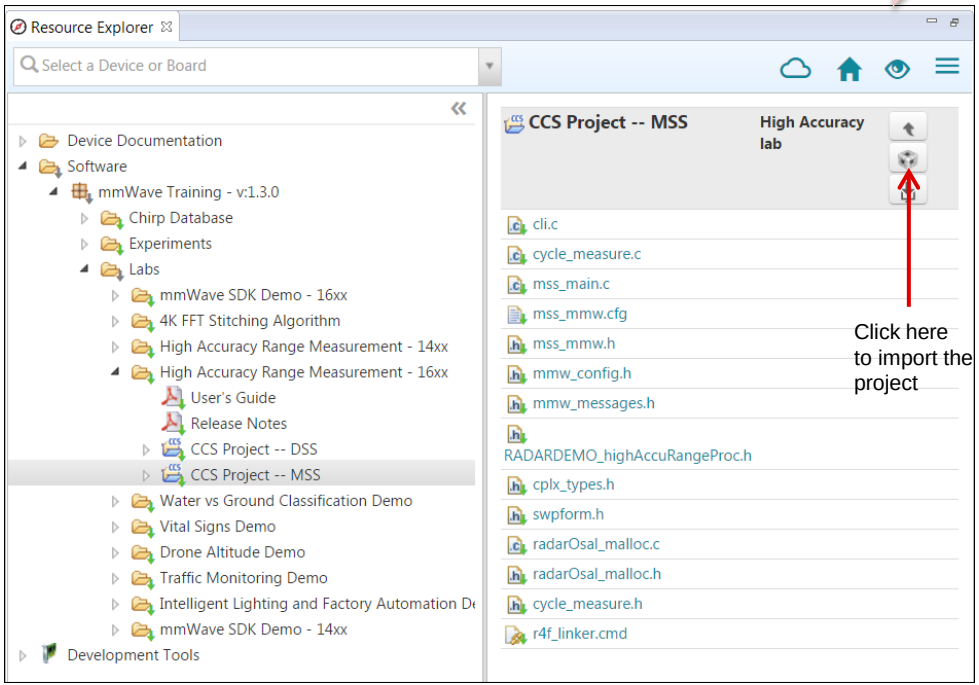
Tool	Version	Download Link
mmWave SDK	2.0.0.4	download link
CCS	7.4 or later	download link
TI SYS/BIOS	6.53.02.00	Included in mmwave sdk installer
TI ARM Compiler	16.9.6.LTS	Included in mmwave sdk installer
TI CGT Compiler	8.1.3	Included in mmwave sdk installer
XDC	3.50.04.43	Included in mmwave sdk installer
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


1.
Pre-
requisites
2.
Import Lab
project
3.
Build Lab
project
4.
Preparing
the EVM
5.
Running the
Lab

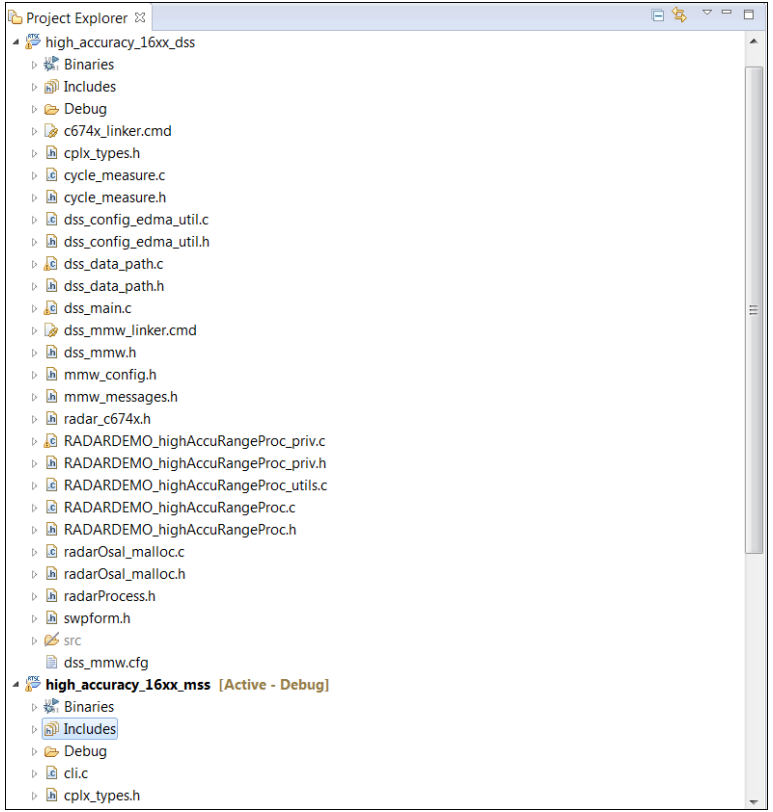
2. Import Lab Project

- The **High Accuracy Range Measurement - 16xx** Lab consists of two CCS projects, one for the R4F core and one for the C674x DSP core
- As shown in quick start section, the lab projects are downloaded and should be available from C:\ti\mmwave_training
- Select the **CCS project -- MSS** in the left view as shown.
- 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_training is not touched.
- Repeat with the CCS Project DSS file

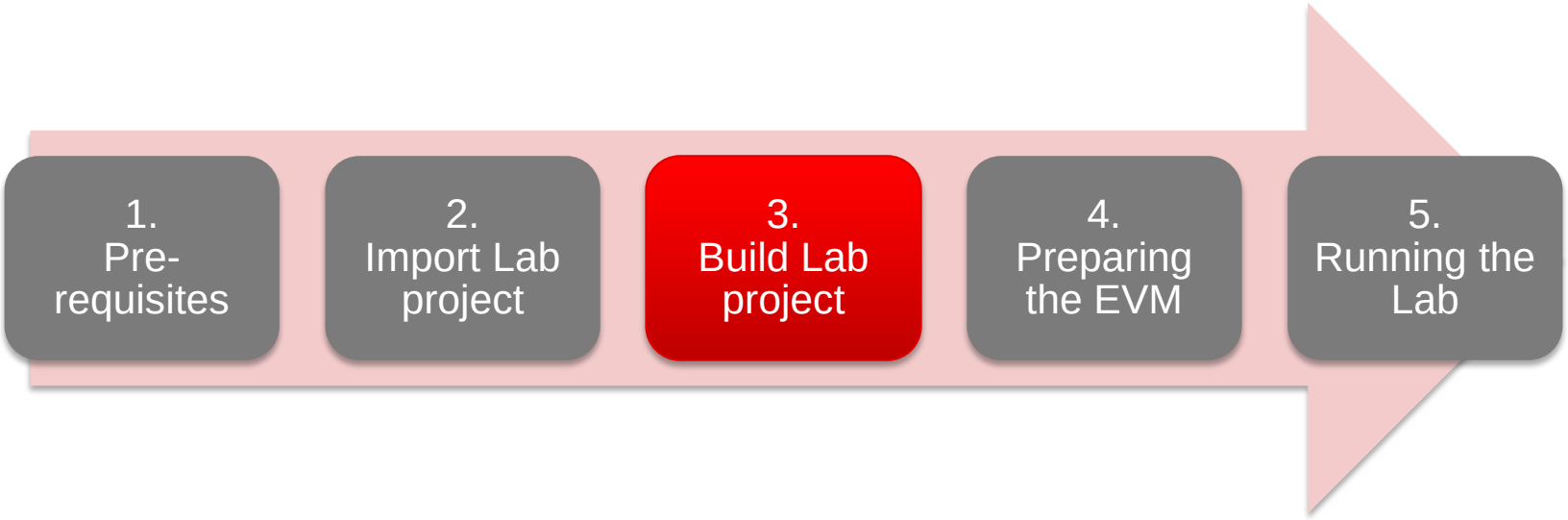


2. Import - continued

- After successfully completing the **Import to IDE** operation, the project should be visible in CCS Project Explorer as shown here.
- At this point, we have successfully downloaded the High Accuracy 16xx Dss and Mss and imported it in CCS.
- We are ready to move on to the next step i.e. Building the project.



Steps

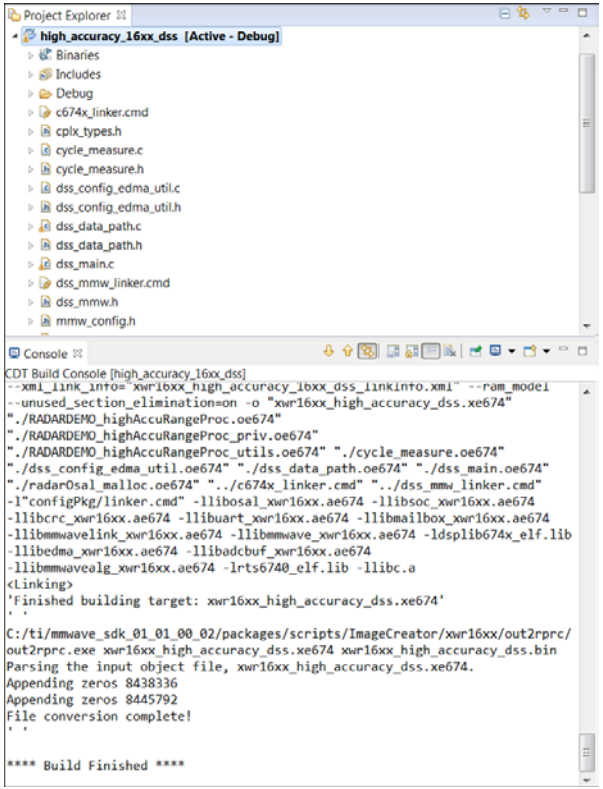

1.
Pre-
requisites
2.
Import Lab
project
3.
Build Lab
project
4.
Preparing
the EVM
5.
Running the
Lab

3. Build the Lab

- With the high_accuracy_16xx_dss 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.

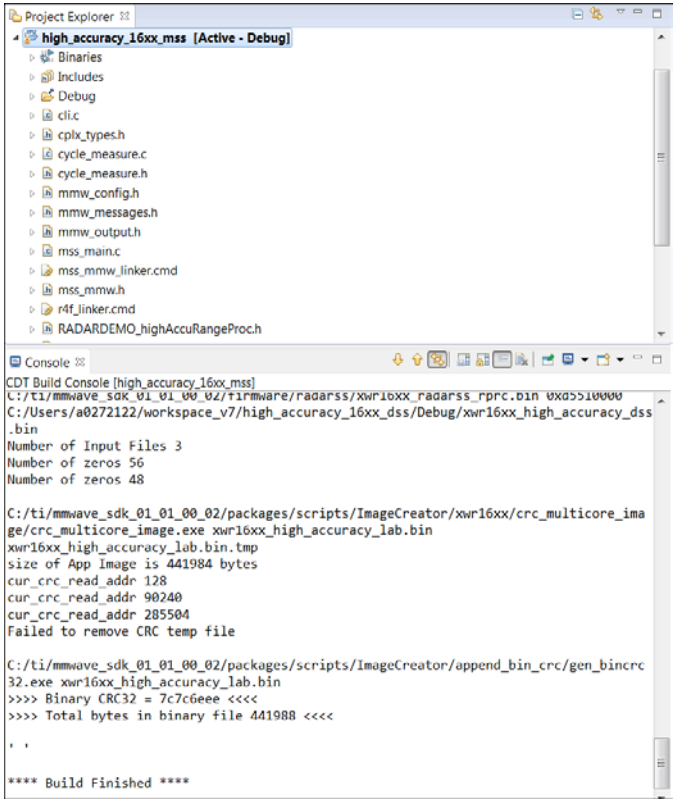
- On successful completion of the build, you should see the output in CCS console as shown here and the following two files should be produced in the project debug directory
  - xwr16xx_high_accuracy_dss.xe674
  - xwr16xx_high_accuracy_dss.bin

- If the build fails with errors, please ensure that all the pre-requisites are installed as mentioned in the mmWave SDK release notes.

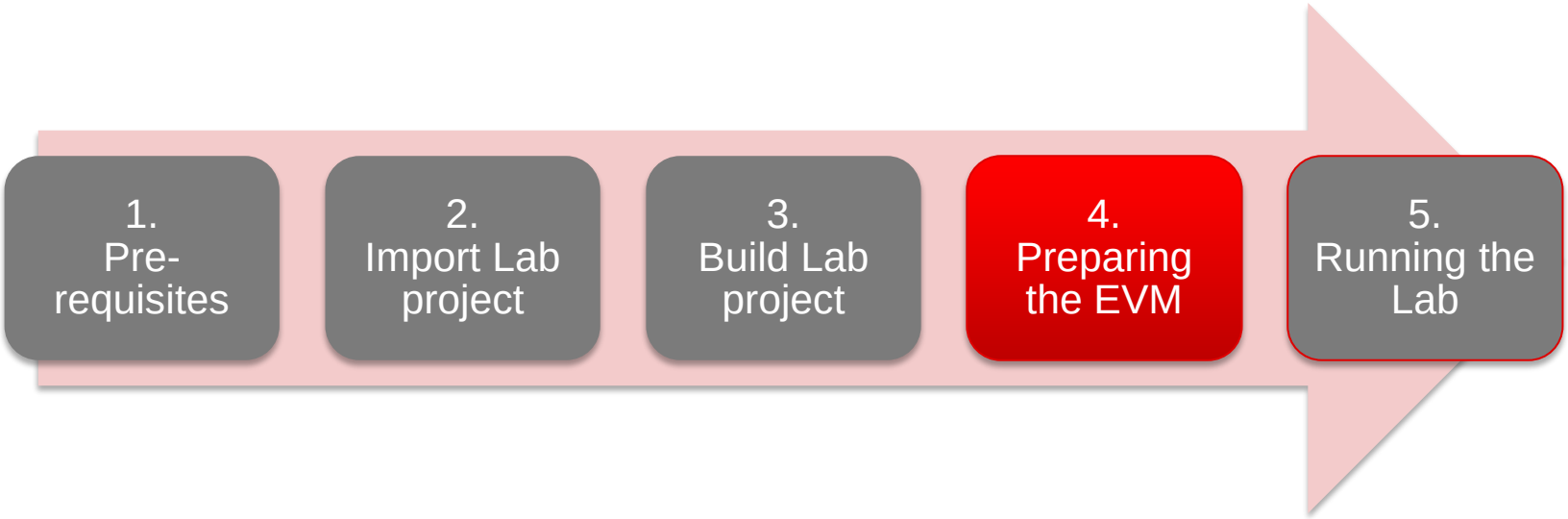


3. Build the Lab – continued

- The **high_accuracy_16xx_dss** project must be built BEFORE the **high_accuracy_16xx_mss** project is built.
- With the **high_accuracy_16xx_mss** project selected in Project Explorer, right click on the project and select **Rebuild Project**
- On successful completion of the 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_high_accuracy_lab.bin
  - xwr16xx_high_accuracy_mss.bin
  - xwr16xx_high_accuracy_mss.xer4f
- If the build fails with errors, please ensure that all the pre-requisites are installed as mentioned in the mmWave SDK release notes.



Steps

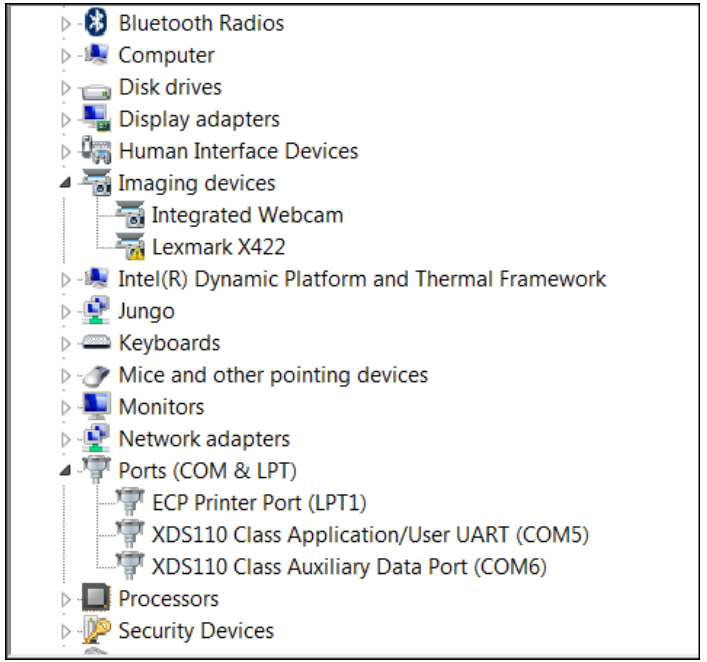

1.
Pre-
requisites
2.
Import Lab
project
3.
Build Lab
project
4.
Preparing
the EVM
5.
Running the
Lab

4. 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) 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.
- The Quick Start section has demonstrated the deployment mode
- The following 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. Connecting to the EVM

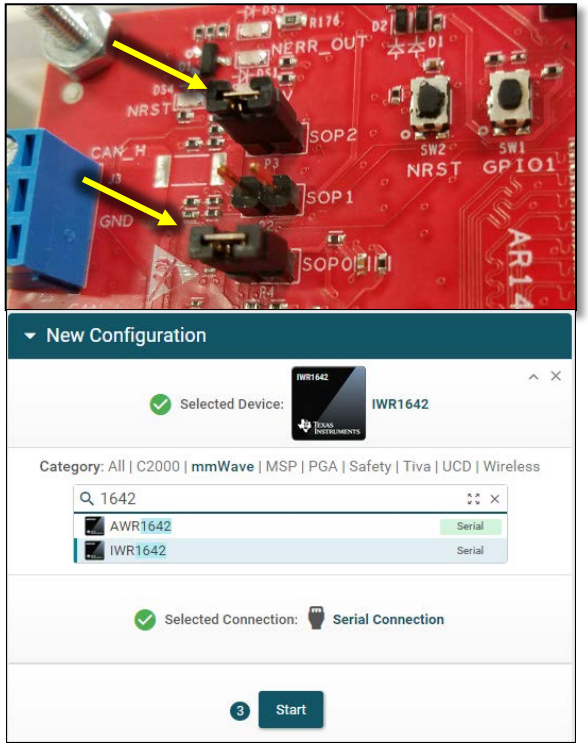
- Power on the EVM using a 5V/2.5A power supply.
- Connect the EVM to your PC and check the COM ports in Windows Device Manager
- The EVM exports two virtual COM ports as shown below:
  - XDS110 Class Application/User UART (COM_{UART}):
    - Used for passing configuration data and firmware to the EVM
  - 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}: COM5 **COM_{AUX}**: COM6 The actual port numbers on your machine may be different

4. Flashing CCS debug firmware

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 the appropriate device (IWR1642 or IWR1642)
4. Click **Start** to proceed



4. Flashing CCS debug firmware

1. In the **Program** tab, browse and locate the ccs debug image shown below:

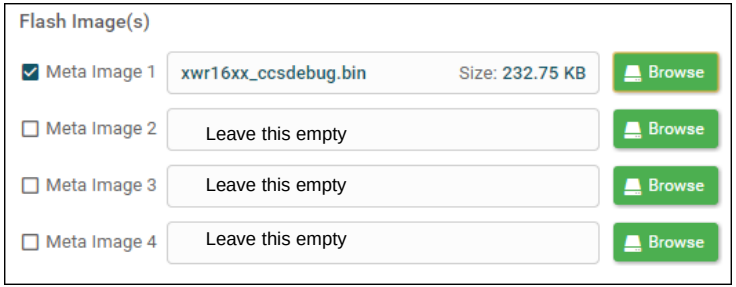
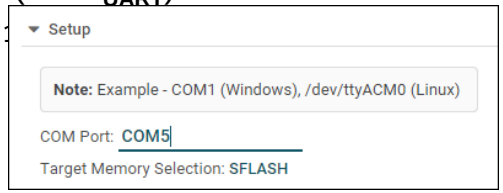


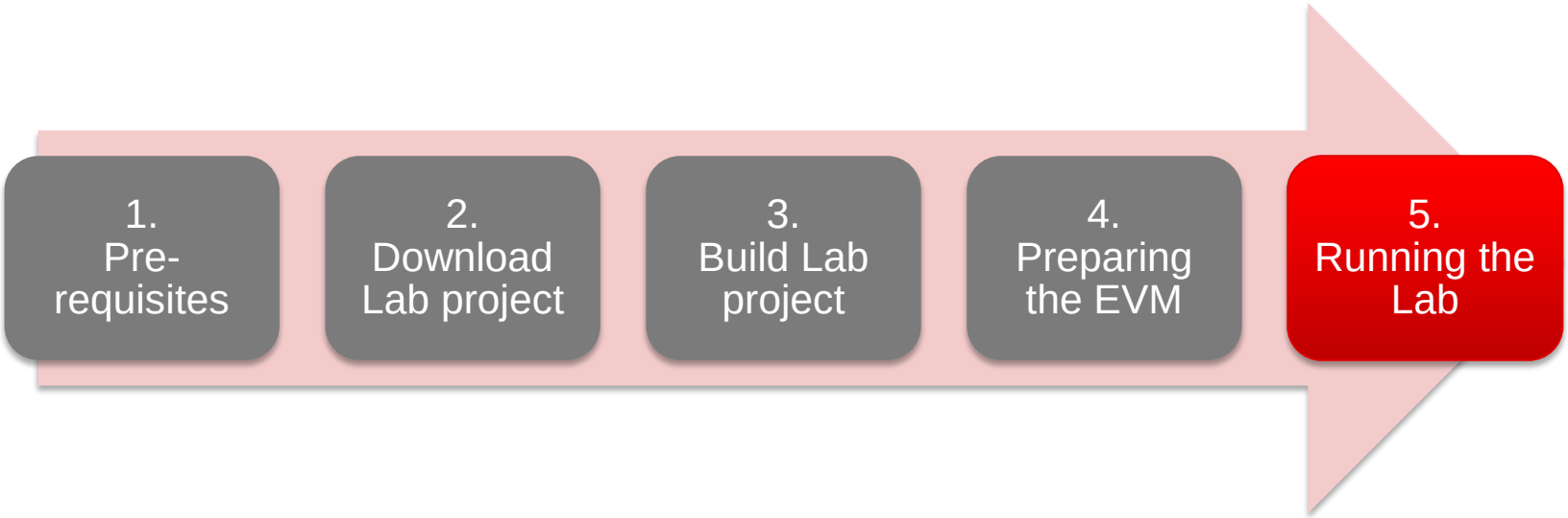
Image	Location
Meta Image 1	C:\ti\mmwave_sdk_<ver>\packages\ti\utils\ccsdebug\xwr16xx_ccsdebug.bin

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



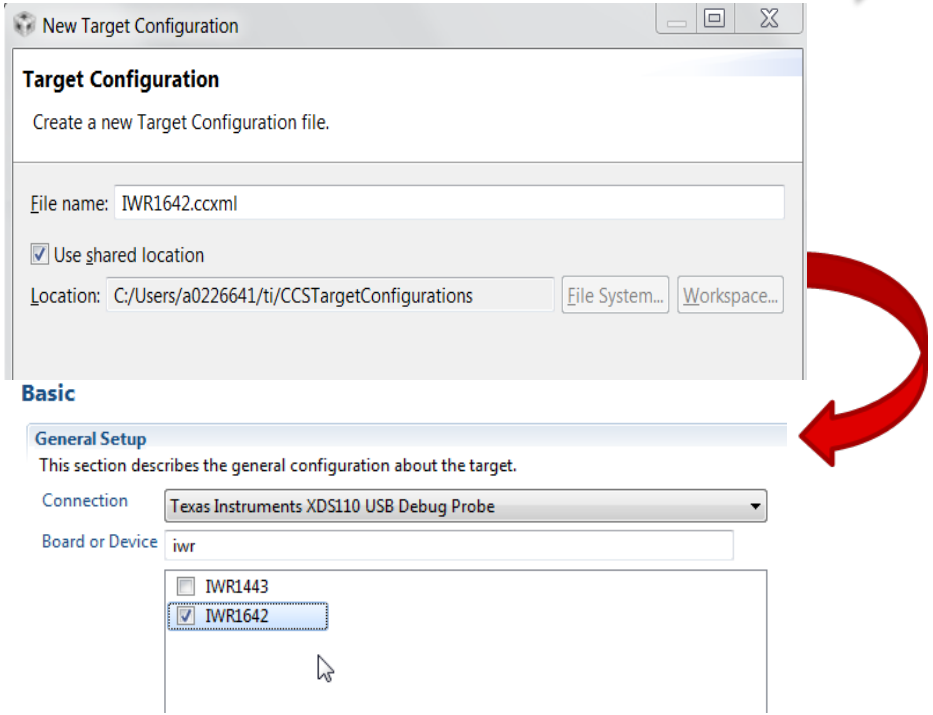
3. Return to the **Program** tab, power cycle the device and click on **Load Images**
4. When the flash procedure completes, UniFlash's console should indicate: [SUCCESS] Program Load completed successfully
5. Power off the board and remove the jumper from only header **SOP2** (this puts the board back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

Steps


1.
Pre-
requisites
2.
Download
Lab project
3.
Build Lab
project
4.
Preparing
the EVM
5.
Running the
Lab

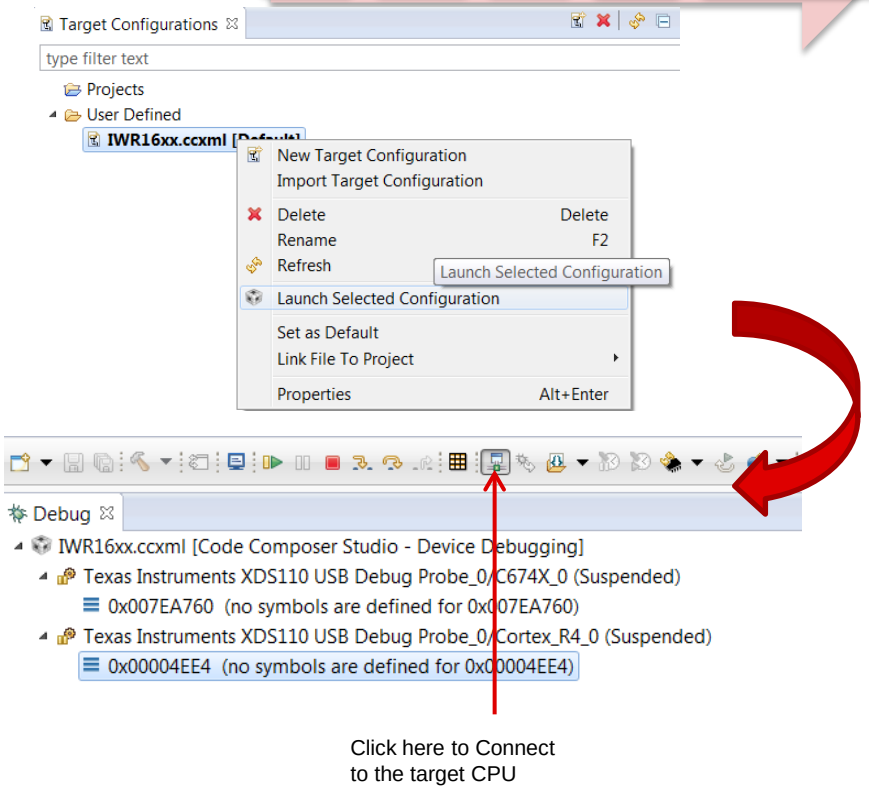
5. Connecting EVM to CCS

- 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 AWR1642 or IWR1642 device as appropriate in the **Board or Device** text box.
    - Press the **Save** button to save the target configuration.
    - You can press the **Test Connection** button to check the connection with the board.



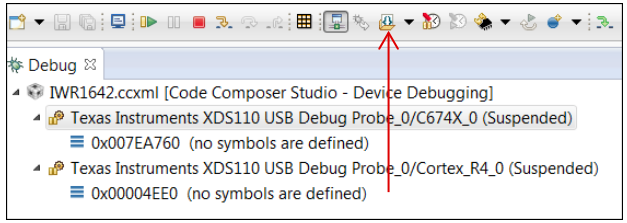
5. Connecting - continued

- Go to **View ► Target Configurations** to open the target configuration window.
- You should see your target configuration under **User Defined** configurations.
- 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_0and press the **Connect Target** button 

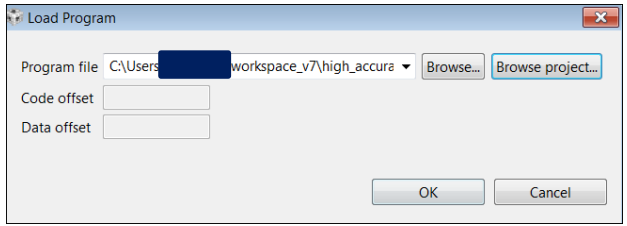


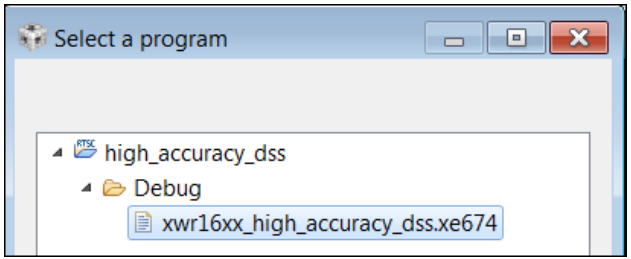
5. Loading the binary

- 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 high_accuracy_16xx_dss project as shown, and press OK.
- Press OK again in the **Load Program** dialog.



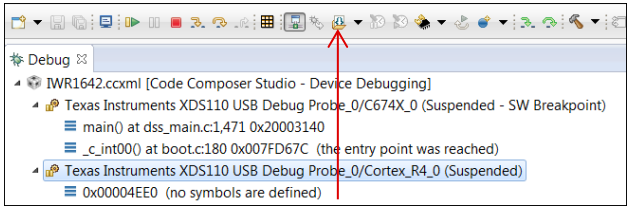
Load Program



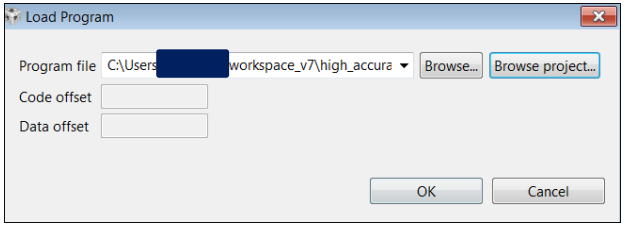


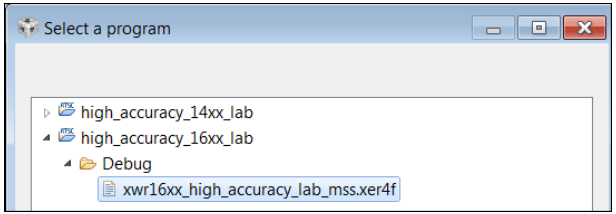
5. Loading the binary

- 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 high_accuracy_16xx_mss project as shown, and press OK.
- Press OK again in the **Load Program** dialog.



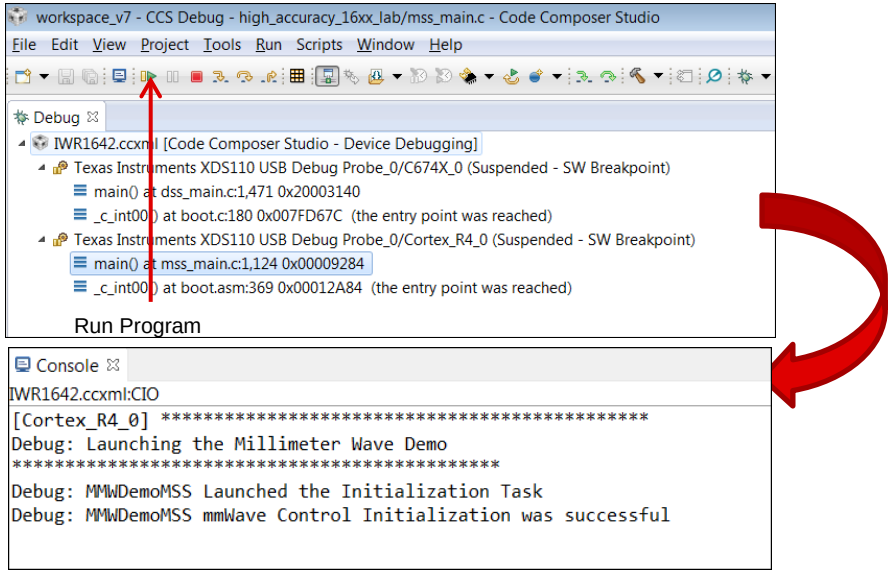
Load Program





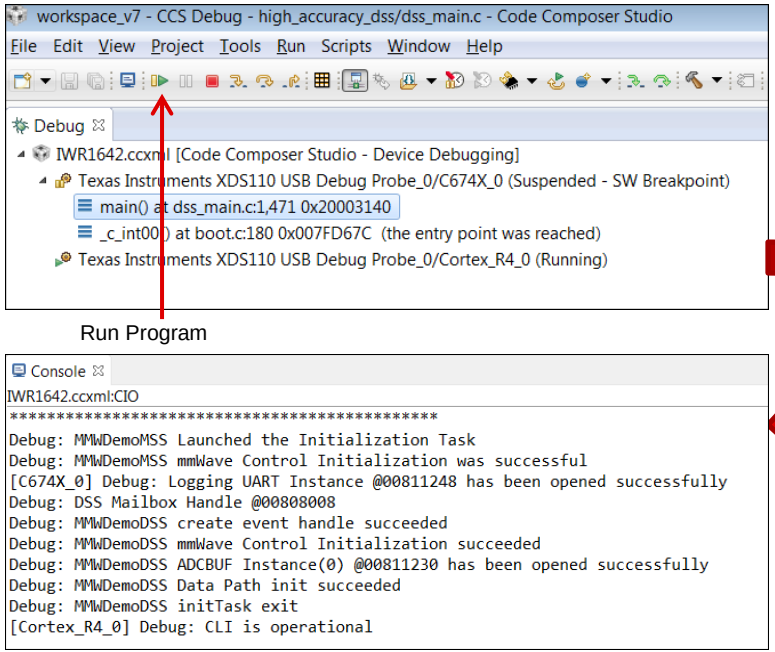
5. Running the binary

- 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.
- If everything goes fine, you should see the “MMWDemoMSS mmWave Control Initialization was successful” message which indicates that the program is waiting for the DSS to be started



5. 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. Follow Quick Start Section 3 for more details.



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 @ http://e2e.ti.com