Using CCS Debug for Development

This mode should be used when debugging with CCS is involved and/or developing an mmWave application where the .bin files keep changing constantly and frequent flashing of image onto the board is not desirable. This mode allows you to flash once and then use CCS to download a different image to the device's RAM on every boot.

0. Requirements

- PC with:
  - Recommended OS: Windows 7 or 10
  - Code Composer Studio (version as specified in demo User's Guide)
  - mmWave SDK installed (version as specified in demo User's Guide)
- The EVM should already be setup for **Flashing Mode** according to the appropriate hardware setup guides for your EVM:

EVM	Guide
IWR6843ISK, IWR6843ODS, or IWR6843AOPEVM + MMWAVEICBOOST Carrier Board	Hardware Setup for Flashing in MMWAVEICBOOST Mode (../hardware_guides/Evm_Setup_Operational_Modes.html)
IWR1443BOOST, IWR1642BOOST, or IWR1843BOOST	Hardware Setup of IWRXXXXBOOST for Flashing Mode (../hardware_guides/Evm_Setup_Operational_Modes.html)

Note: The IWR6843AOPEVM cannot be used by itself for CCS debug mode. It must be used as an antenna module in conjunction with the MMWAVEICBOOST carrier board.

1. Flash the CSS debug firmware using Uniflash

The debug binary is provided in the mmWave SDK. Flash the appropriate binary according to the instructions for using UniFlash (using_uniflash_with_mmwave.html)

Device	BIN Name	Location
IWR14XX	xwr14xx_ccsdebug.bin	`C:\ti\mmwave_sdk_<VER>\packages\ti\utils\ccsdebug`
IWR16XX	xwr16xx_ccsdebug.bin	`C:\ti\mmwave_sdk_<VER>\packages\ti\utils\ccsdebug`
IWR18XX	xwr18xx_ccsdebug.bin	`C:\ti\mmwave_sdk_<VER>\packages\ti\utils\ccsdebug`
IWR68xx or IWR68xx AOP	xwr68xx_ccsdebug.bin	`C:\ti\mmwave_sdk_<VER>\packages\ti\utils\ccsdebug`

2. Setup the EVM for Functional Mode

Put the EVM in functional mode.

EVM	Guide
IWR6843ISK, IWR6843ODS, or IWR6843AOPEVM + MMWAVEICBOOST Carrier Board	Hardware Setup of MMWAVEICBOOST + Antenna Module for Functional Mode (../hardware_guides/Evm_Setup_Operational_Modes.html)
IWR1443BOOST, IWR1642BOOST, or IWR1843BOOST	Hardware Setup of IWRXXXXBOOST for Functional Mode (../hardware_guides/Evm_Setup_Operational_Modes.html)

3. Connect the EVM to CCS using JTAG

3.1 Create Target Configuration File (CCXML)

Creation of the CCXML file is needed for each device type, but only the first time CCS debug mode is used. In subsequent sessions, this step can be skipped and the existing CCXML file can be used directly.

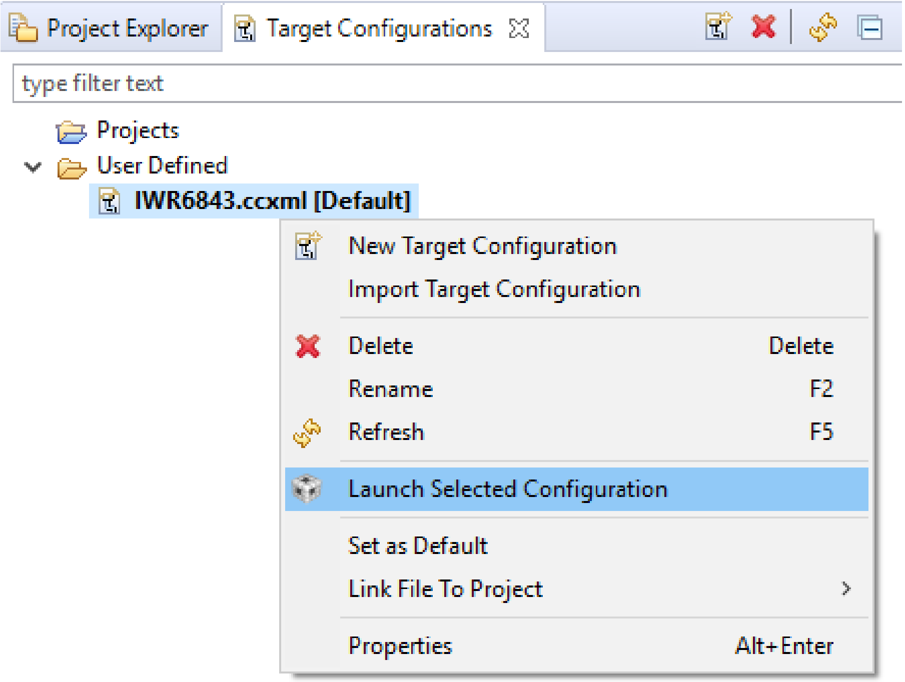
To create the CCXML file:

1. Launch CCS

2. If CCS does not already show "Target Configurations" window, go to **View** → **Target Configurations**.
3. This will show the "Target Configurations" window, right click in the window and select "New Target Configuration"
4. Specify an appropriate file name (ex: IWR68xx.ccxml) for the particular EVM being used and check "Use shared location". Click **Finish**.
5. The configuration editor window will open.
6. In the window, select **Texas Instruments XDS110 USB Debug Probe** for Connection and select the appropriate mmWave device in the **Board or Device** text box.
7. Press the "Save" button to create and save the target configuration.
8. [Optional]: Press the "Test Connection" button to check the connection with the board.

3.2 Connect to EVM using Target Configuration File and Load Binaries

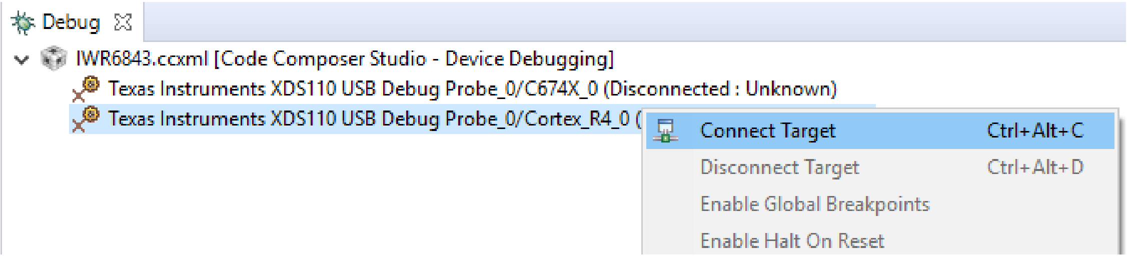
- If needed, open the target configuration window by going to **View** → **Target Configurations**.
- Under **User Defined** configurations the target configuration file previously created should appear.
- Right click on the target configuration and select **Launch Select Configuration**. The target configuration will launch in the **Debug Window** and show all cores present on the device. The following images may vary slightly depending on device type.



- The necessary core(s) to connect to will depend on what is utilized by the demo code. This can be determined by examining which debug binaries are provided by the demo. The table below maps the file to the appropriate core. The demo may use only one or both.

Debug binary file extension	Connect and load to:
*.xer4f	Cortex_R4_0
*.xe674	C674X_0

- For each necessary core, select and right click on the to select **Connect Target**. An example is shown below. When connected, the message "(Disconnected: Unknown)" changes to "(Suspended)".


Debug
IWR6843.ccxml [Code Composer Studio - Device Debugging]
Texas Instruments XDS110 USB Debug Probe_0/C674X_0 (Disconnected : Unknown)
Texas Instruments XDS110 USB Debug Probe_0/Cortex_R4_0 (
Connect Target Ctrl+Alt+C
Disconnect Target Ctrl+Alt+D
Enable Global Breakpoints
Enable Halt On Reset

- For each debug binary, click on the appropriate core so that it is highlighted then click the **Load** button in the toolbar. In the **Load Program** dialog, press the **Browse Project** button. Select the debug binary listed in the demo user's guide, ensuring that the correct file is used for the core as listed in the table above. Press **Ok** until you are returned to the debug window. The program will now load.

  Note: If the demo has **.xer4f* and **.xe674* binaries, complete the connect to target and load program steps for each core/binary.

3.3 Run

- For each core a binary was loaded to, select the core so that it is highlighted then press the **Run/Resume** button
- The program should start executing and generate console output.

The systems is now setup for CCS debugging.

Return to User's Guide.

Need More Help?

- For further tips on real-time debugging, consult "Section 6.15. CCS Debugging of real time application" of the mmWave SDK (http://software-dl.ti.com/ra-processors/esd/MMWAVE-SDK/latest/exports/mmwave_sdk_user_guide.pdf)
- For more information on the Code Composer Studio (CCS) debug environment consult "Section 7. Debug" of the CCS User's Guide (http://software-dl.ti.com/ccs/esd/documents/users_guide/index_debug.html#debug)

- Search for your issue or post a new question on the mmWave E2E forum (https://e2e.ti.com/support/sensor/mmwave_sensors/f/1023)