This document will provide detailed instructions on how to capture data over LVDS using mmWave SDK (v3.4) and DCA1000EVM.

Radar in use : AWR1642 - ES 2.0 (SN: 5637200025)

Know that LVDS streaming functionality over CLI is only provided by the following radars:

- **xWR16xx**
- **xWR18xx**
- **xWR68xx**

1. Positioning switches SW1 and SW2 on DCA1000EVM
   a. SW1

SW1.1	12bit_OFF
SW1.2	14bit_OFF
SW1.3	16bit_ON

   b. SW2

SW2.1	LVDS_CAPTURE	(0)
SW2.2	ETH_STREAM	(1)
SW2.3	AR1642_MODE	(1)
SW2.4	RAW_MODE	(0)
SW2.5	SW_CONFIG	(1)

2. Flashing radar using UniFlash:
   a. Use jumpers to short SOP0 and SOP2 pins. This SOP mode is only used when flashing firmware to the device with UniFlash.
   b. Power cycle the radar by holding SW2 for 1 second.
   c. Allow UniFlash to automatically detect the device. Ensure that the detected device is correct, and the icon is red. (The black icon is for chip only flashing)
   d. Select the correct COM Port (User/UART) under the “Settings & Utilities” Tab.
   e. On the “Program” Tab, for **Meta Image 1**, choose “Browse” to locate the demo binary. This binary will be located in the SDK installation path. For example:

      C:\ti\mmwave_sdk_03_04_00_03\packages\ti\demo\xwr16xx\mmw\xwr16xx_mmw_demo.bin

   f. Press the “Load Image” button. Once process is complete, remove ONLY the SOP2 jumper (leave SOP0 shorted)

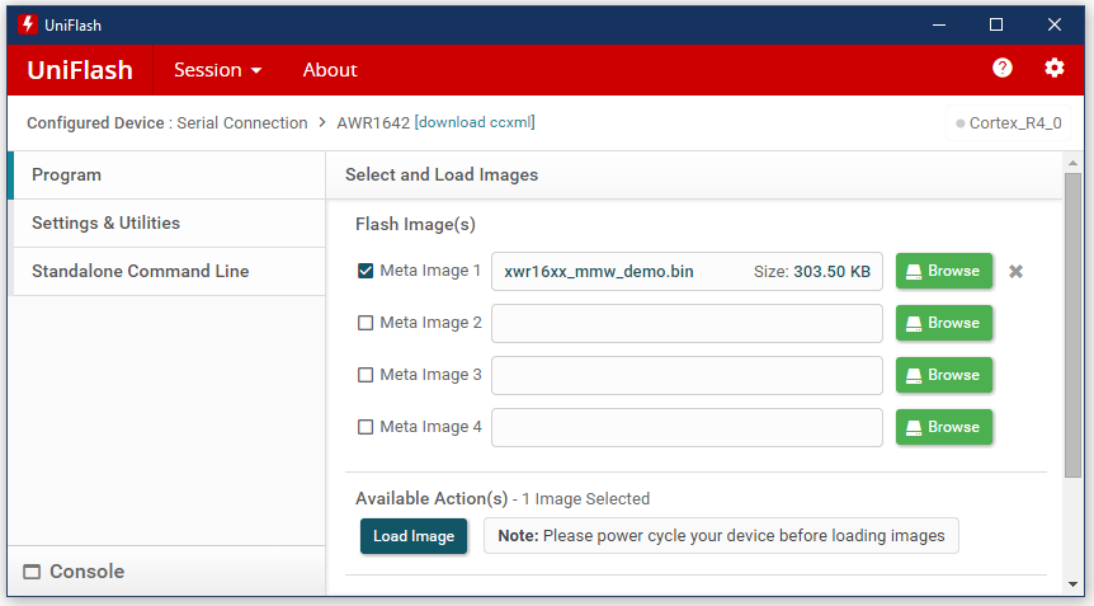

Figure 1 – UniFlash

3. Next, run the TI mmWave Demo Visualizer available online to confirm operation of radar. Once radar is confirmed working, select desired "scene selection" parameters and download the configuration file by pressing "**Save Config to PC**". I suggest renaming the configuration file to something distinct such as "awr1642_profile.cfg".

4. Modify the **lvdsStreamCfg** command in the "awr1642_profile.cfg" file according to settings shown in the Figure 2 - lvdsStreamCfg Table below. If **lvdsStreamCfg** requires modifying other commands, do so using the correct SDK user guide. (Ex: See how enabling "CP_ADC_CQ data" requires modifying command "analogMonitor")

Modify the **chirpThreshold in adcBufCfg to 1**. Otherwise, an exception will occur on line 1814 of *mss_main.c* stating that the number of chirps is not as expected. Ex: Change **adcbufCfg -1 0 1 1 0** to **adcbufCfg -1 0 1 1 1**

5. Create a copy of the DCA1000EVM's configuration file for a CLI unique version. This file is located on your mmWaveStudio installation path and by default has name **cf.json**

   C:\ti\mmwave_studio_02_01_01_00\mmWaveStudio\PostProc\cf.json

   Name the copy something distinct such as **awr1642_cf.json**

6. Modify **awr1642_cf.json**:

   a. Modify **dataLoggingMode:**
      i. If <enableHeader> is disabled (0) in lvdsStreamCfg, change "dataLoggingMode" to **"raw"**
      ii. If <enableHeader is enabled (1) in lvdsStreamCfg, change "dataLoggingMode" to **"multi"**

<enableHeader> is the **2nd** digit of **lvdsStreamCfg** (be sure to include the quotations)

b. Change **lvdsMode** to **2** (all supported devices only have 2 LVDS lanes)
c. Change **dataFormatMode** to **3** (all supported devices only allow 16-bit capture)
d. All ethernet settings should remain the same to allow for easy switching of the DCA1000EVM between radars using CLI and mmWaveStudio.
e. Modify **captureConfig:**
    i. **fileBasePath** changes where the saved data file is stored.
    ii. **filePrefix** changes the name of the saved data file.
    iii. **captureStopMode** changes the duration of the measurement. **"infinite"** will measure until there is no data present on the LVDS lanes or until the radar is told to stop. **"bytes", "frames", and "duration"** are used in combination with the next 3 lines (in the .json file) to tell the DCA1000EVM when to stop measuring data.
f. Modify **dataFormatConfig:**
    i. **MSBToggle** should be set to **0**.
    ii. **reorderEnable** should be set to **1**.
    iii. **dataPortConfig** should have all dataTypes set to **"complex"**. This is due to mmWave demo binary using all CBUFF/LVDS sessions as complex.

7. Running command to capture data with DCA1000EVM's CLI utility:
    a. Change directory to the modified (awr_1642.json) file. If in the same directory as the original cf.json file, the command to change directory will look like this:

```
cd "C:\ti\mmwave_studio_02_01_01_00\mmWaveStudio\PostProc\"
```

    b. Now use the 'DCA1000EVM_CLI_Control.exe' utility to start the capture by running the 'fpga', 'record', and 'record_start' commands, in this respective order. Doing so will look like the following command lines:

```
.\DCA1000EVM_CLI_Control.exe fpga AWR1642_cf.json
.\DCA1000EVM_CLI_Control.exe record AWR1642_cf.json
.\DCA1000EVM_CLI_Control.exe start_record AWR1642_cf.json
```

    c. To see a list of all the available commands in the DCA1000 CLI Utility, use the -h command:

```
.\DCA1000EVM_CLI_Control.exe fpga AWR1642_cf.json
```

    d. Once the start_record command is sent, the radar must begin data transmission within 30 seconds, otherwise the DCA1000EVM will run the stop_record command. Any data sent by the radar after the stop_record command will not be captured until the start_record command is run again.
    e. To properly end data capture, TI states that it is better to run the stop_record command before issuing the sensorStop command to the radar. I have found that if there is around 3 seconds of no incoming LVDS data during a capture, the DCA1000EVM will automatically issue the stop_record command. This means that anytime you issue sensorStop to the radar, then you are guaranteed that within 3 seconds the data capture will automatically end on the DCA1000EVM as well.

Figure 2 - lvdsStreamCfg Table

lvdsStreamCfg	Enables the streaming of various data streams over LVDS lanes. **When this feature is enabled, make sure chirpThreshold in adcbufCfg is set to 1.** The values in this command can be changed between sensorStop and sensorStart. This is a mandatory command.	<subFrameIdx> subframe Index	For legacy mode, this field should be set to -1 For advanced frame mode, it should be set to either the intended subframe number or -1 to apply same config to all subframes.
		<enableHeader> Enable/disable HSI header for all active data streams	0 - Disable 1 - Enable
		<dataFmt> Controls HW streaming. Specifies the HW streaming data format.	0-HW STREAMING DISABLED 1-ADC 4-CP_ADC_CQ When choosing CP_ADC_CQ, please ensure that CQRxSatMonitor and CQSigImgMonitor commands are provided with appropriate values and these monitors are enabled using analogMonitor command.
		<enableSW> Enable/disable user data (SW session)	0 - Disable 1 - Enable <enableHeader> should be set to 1 when this field is enabled.

Example: Change **lvdsStreamCfg -1 0 0 0** to **lvdsStreamCfg -1 0 1 0**