User's Guide for ADC Data Capture using OOB Demo and DCA1000EVM CLI application

MMWAVE-SDK OOB (out-of-box) demo supports raw ADC data streaming through the LVDS interface. When the radar EVM board connected with the DCA1000EVM board. Users can use the DCA1000EVM CLI application to capture the raw data without starting the mmWaveStudio GUI interface.

The DCA1000EVM CLI application is provided for Windows and can be recompiled for other platforms. Users can find the source code and user guide below (or any MMWAVE-STUDIO release package) C:\ti\mmwave_studio_02_01_01_00\mmWaveStudio\ReferenceCode\DCA1000

This feature is also supported for low-power devices xWRL6432/xWRL1432. The MMWAVE-L-SDK supports raw ADC data streaming through the RFID interface.

In this package, we provide an example (through MATLAB executable) to use the MMWAVE-SDK/MMWAVE-L-SDK demos and the DCA1000CLI interface to get the raw ADC data captured. This tool configures and starts the sensor, communicates with the DCA1000 board, parses the configuration file to understand the frame structure, and finally parses the raw data to plot the 2D FFT output.

1. Flash the out-of-box demo binary onto the target

xWR6843	xWRL6432/xWRL1432
Follow the instruction below to flash the OOB to first-generation devices like xWR6843/xWR1642/xWR1843 https://dev.ti.com/tirex/explore/node?node=A__AOJgwAYscJOUPEaB-c-y7A__radar_toolbox__1AslXXD__LATEST	Follow the instruction below to flash the OOB (out of box) demo binary. <MMWAVE_L_SDK_05_03_00_02_DIR>/docs/MMWAVE_Radar_Visualizer(mmWave_Low_Power_Devices).pdf
A prebuilt binary is provided at: .\ prebuilt_binaries\xwr68xx_mmw_demo.bin The matching configuration file is provided as .\chirp_configs\xWR6843_inCabinSensing.cfg	A prebuilt binary is provided .\ prebuilt_binaries\ motion_and_presence_detection_demo.release.appimage The matching configuration file is provided as .\ chirp_configs\ xWRL6432_inCabinSensing_5104.cfg

Notes: When a different version of the SDK OOB demo is used, then users need to adjust the configuration file accordingly because a different version of the release binary can require a slightly different radar configuration file.

2. Hardware Setup

DCA1000EVM should be connected to Host PC via an Ethernet cable to access the CLI and Data Transfer process. DCA1000EVM should be connected to TI Radar EVM via 60 pin HD Connector by using 60 pin SAMTEC ribbon cable. DCA1000EVM power input should be connected either from DC Jack or TI Radar EVM power output (from 60 pins HD connector) by selecting the switch SW3.

Figure 1 shows the hardware setup with DCA1000 + MMWAVEICBOOST+ xWR6843 EVM.

Figure 2 shows the hardware setup with DCA1000 + xWR6843 EVM standalone.

Figure 3 shows the hardware setup with DCA1000 + xWRL6432 EVM, the same switch setting should be applied to xWRL1432 EVM.

Please pay close attention to the switch settings in yellow highlight.

Similar to the data capture using the MMWAVE-STUDIO GUI application, users have to set the PC to static IP and disable WIFI to be able to communicate with the DCA1000EVM board. The instruction can be found at the online DCA1000 MMWAVESTUDIO user guide: https://dev.ti.com/tirex/explore/node?node=A__AGTrhNYW8jE6cMxbovlfaA__radar_toolbox__1AslXXD__LATEST. Users can navigate to the session “confirming hardware detecting” to find the instruction to set the static IP address.

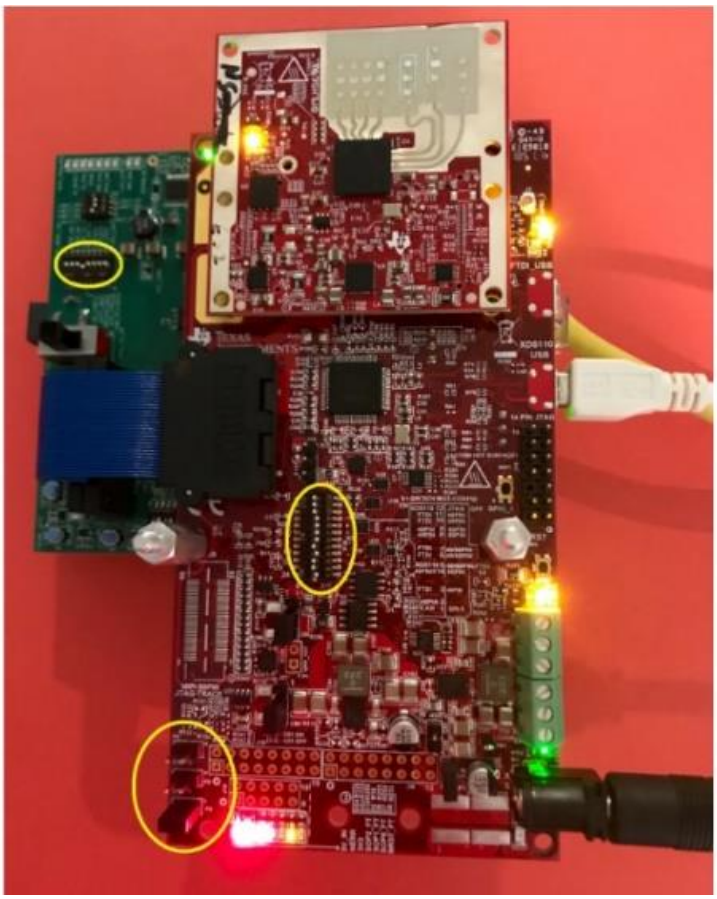

Figure 1 Hardware setting with DCA1000 + MMWAVEICBOOST + xWR6843ISK EVM

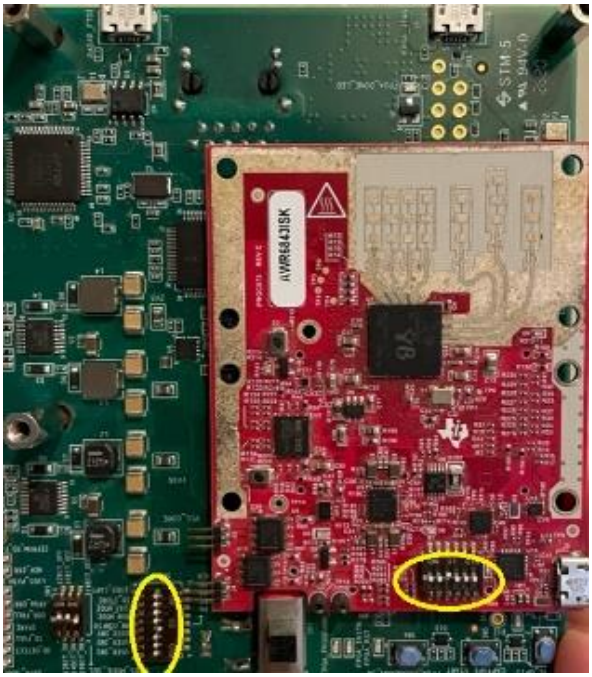

Figure 2 Hardware setup with DCA1000 + xwr6843 standalone EVM board (without MMWAVEICBOOST)

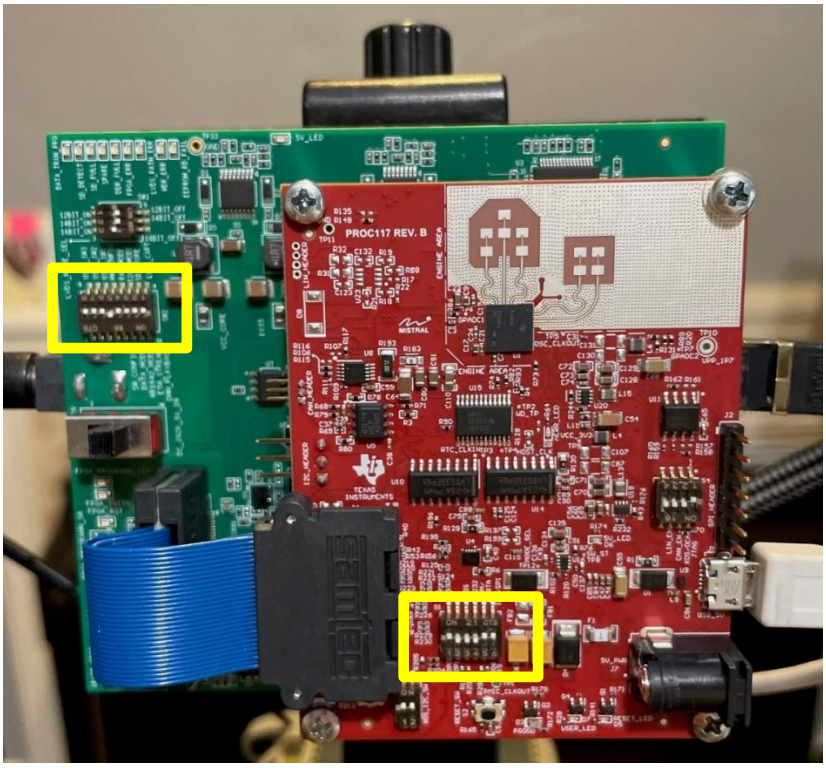

Figure 3 Hardware setting with DCA1000 + xWRL6432 EVM

3. Run the MATLAB data capture executable

3.1 Install MATLAB RUNTIME ENGINE

Note that you will have to install the right MATLAB runtime library before running MATLAB executable. The runtime library information is listed in readme.txt. The source code of this executable is included in .\gui directory.

3.2 Configuration

Double-click on rawDataCaptureGUI_DCA1000CLI.exe, and you will see the following configuration GUI pop out after about 10 seconds. The last used configuration port number, configuration file name, and raw data file name will be saved at “settings.txt”, and automatically reloaded next time for ease of use.

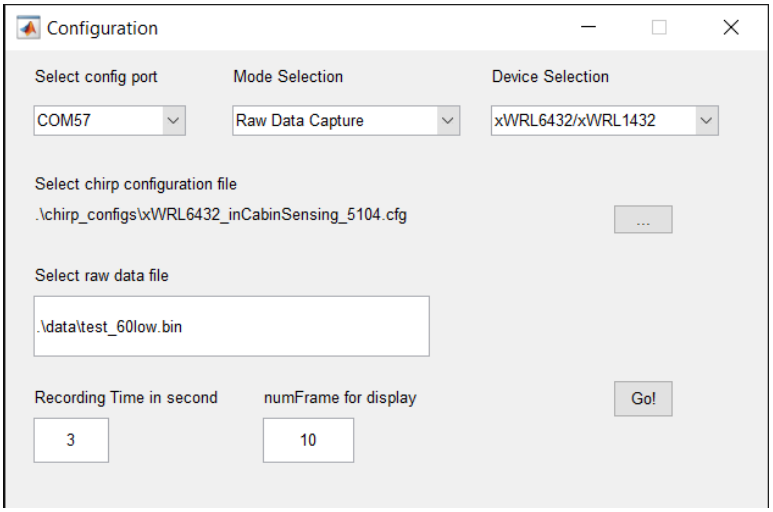

Figure 4 Raw ADC data capture with DCA1000CLI.

Notes: the users need to choose the correct device group to start the corresponding data parsing function.

From the configuration GUI, you will be able to

- Choose the COM configuration port;
- Choose the mode: capture mode or replay mode;
- Choose the device type: xWR6843/1843/1642 or xWRL6432/1432;
- Choose the configuration file;
- Choose the raw data file name (check section 4 for more details);
- Choose the total capture time in second;
- Choose the number of frames for display: non-coherent combined (across antennas) 2D FFT output is displayed for each frame, which serves as a quick check for the captured data. An example is shown in the figure below;
- After that, click on Go! Button to start configuration and data capture.

3.3 Post process after capture is done.

After the capture is done, non-coherent combined (across antennas) 2D FFT output is displayed for each frame, which serves as a quick check for captured data.

- The output example with xWR6843 EVM is shown in the figure below. Different color curves represent different Doppler bins.

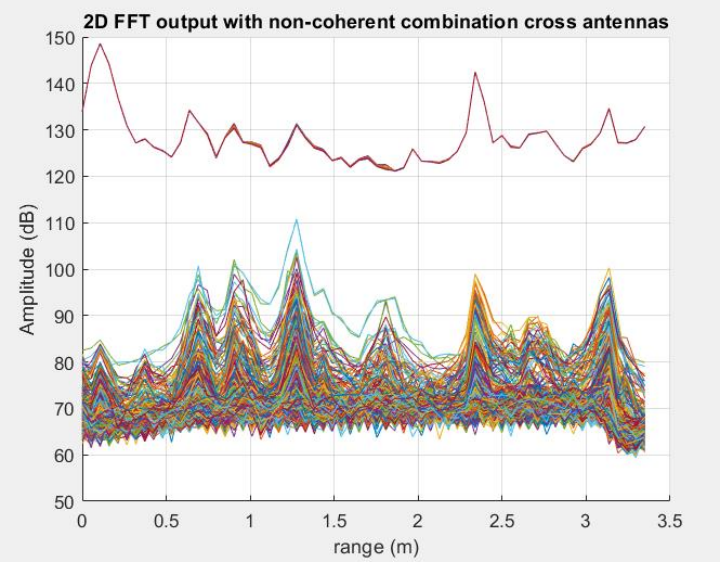

Figure 5 2D FFT output (different color is different doppler bins)

- The output example with xWRL6432 EVM is shown in the figure below. Note that since xWRL6432 only supports real mode, the range FFT output is symmetric, and only the first half of the range FFT output will be used.

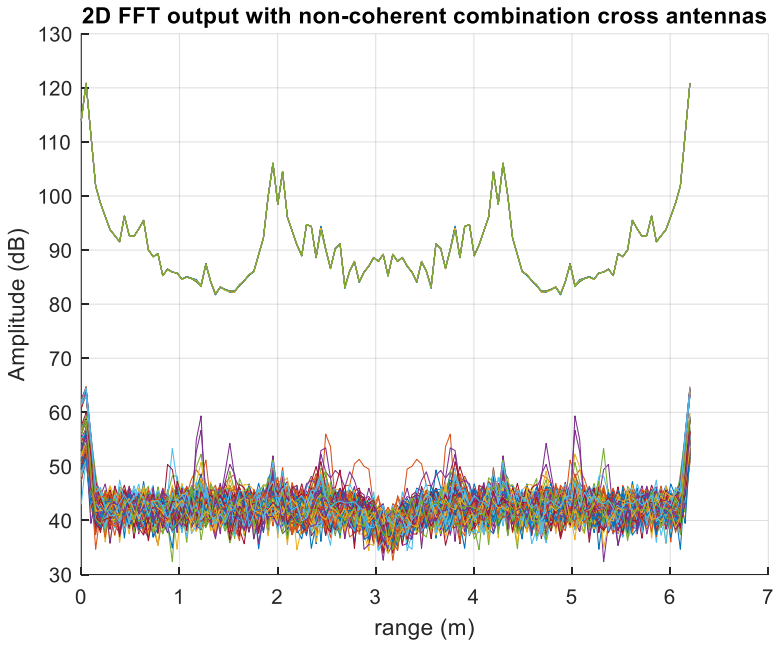

Figure 6 2D FFT output (different color is different doppler bins)

4. Saved raw data name

Through GUI, users can choose the raw data file name, for example '.\data\test.bin'. However, the capture tool will actually save the raw data into multiple data files as:

‘.\data\test_Raw_0.bin’

‘.\data\test_Raw_1.bin’

…

The number of files depends on the total data size and the maximum allowed file size setting in “datacard_config.json”. For example, if the customer sets the capture time to be long and the total data size goes beyond 1GHz. Then it will result in multiple data files. Note that the "maxRecFileSize_MB" is set to 1024MB inside “datacard_config.json”.

5. Easy ways to adapt to the new use case

It is common that users will come up with their own chirp configuration. Users are encouraged to change the profile/chirp/frame configuration but keep the rest of the settings.

6. Bugs and Limitations

For the xWRL6432 device, this package is tested and works with MMWAVE_L_SDK_05_01_00_04. The data format has changed since MMWAVE_L_SDK_05_02_00_02. We are still working on modifying the data parsing to support the newer version of SDK L.

This capture tool is not as flexible as the radar studio, because it has the same limitation in the MMWAVE-SDK OOB demo. For example, the MMWAVE-SDK OOB demo does not support CW signal, does not support multiple profiles, does not support radar cube sizes larger than 768Kbytes and etc. The benefit is that you receive the exact ADC data as you run the MMWAVE-SDK demo.