

DCA1000EVM

Test Procedure Document

CHANGE HISTORY			
Version .No	Date	Change Description	Remarks
0.1	03-Oct-17	Initial Draft	
0.2	09-Nov-17	Updated Raw Mode Use Cases	
0.3	13-Nov-17	Updated Radar Studio Procedure	
0.4	17-Nov-17	Updated internal review comments	
0.5	01-Dec-17	Updated use case Post processing	
0.6	15-Dec-17	Updated Multi mode and playback test procedures	
0.7	22-Dec-17	Updated Raw Mode use case and added capture card help menu details	
0.8	12-Jan-18	Updated Use Case details	
0.9	22-Jan-18	Updated Playback use case details	
1.0	02-Feb-18	Updated EVM number and Multi mode use case	
1.1	06-Mar-18	Updated DCA1000EVM Ethernet Configuration, command format and data format	
1.2	10-Apr-18	Updated Post processing for Packet Re-ordering & Zero Filling	
1.3	19-Apr-18	Updated Post processing procedure	
1.4	20-Jul-19	Updated FPGA Programmer Link	

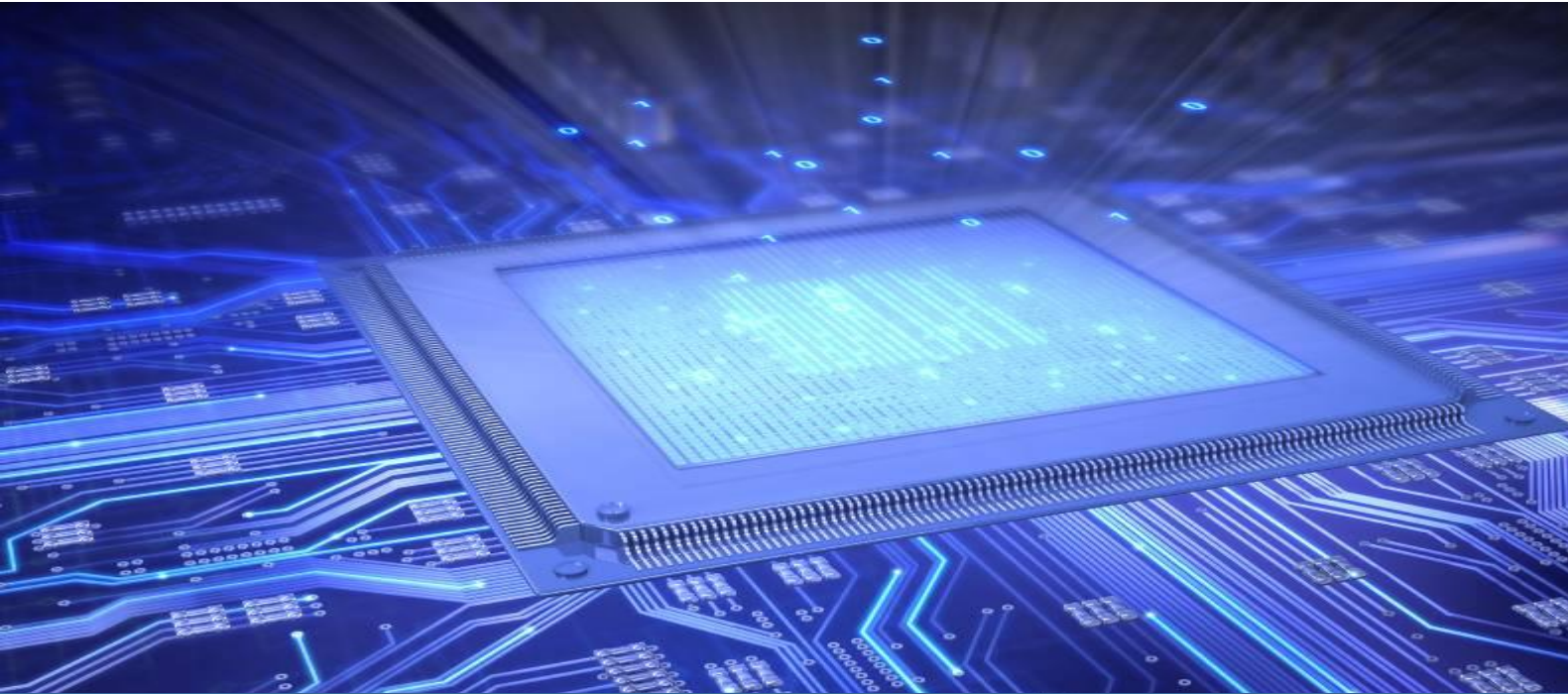


Table of Contents

1 Introduction	4
1.1 PURPOSE.....	5
1.2 TERMS / ACRONYMS / ABBREVIATIONS.....	5
1.3 BIBLIOGRAPHY	5
2 DCA1000EVM and Accessories	6
2.1 ACCESSORIES FOR BOARD TESTING	8
2.2 SOFTWARE REQUIREMENTS FOR TESTING	9
2.3 TEST PACKAGE.....	9
2.4 DEFAULT SWITCH & PUSH BUTTON SETTINGS ON DCA1000EVM	9
2.5 DEFAULT LED STATUS ON DCA1000EVM.....	11
3 FPGA Programming	12
3.1 HARDWARE SETUP.....	13
3.2 FPGA - SPI FLASH PROGRAMMING	13
4 FTDI- FT4232 EEPROM.....	16
4.1 FT4232HL(REFERENCE: U3) - EEPROM PROGRAMMING.....	16
5 DCA1000EVM to Radar EVM Board to Board to connection.....	19
6 System Testing.....	21
6.1 TEST SETUP:	21
6.2 CONFIGURING RADAR EVM BY USING RADAR STUDIO	24
6.2.1 Fixed test pattern generation	24
6.2.2 Real Time LVDS Data generation	29
6.3 DCA1000EVM ETHERNET CONFIGURATION DATA.....	30
6.4 DCA1000EVM COMMAND AND DATA FORMAT	32
6.4.1 DCA1000EVM command format	32
6.4.2 DCA1000EVM data format.....	33
6.5 DCA1000EVM GUI PROCEDURE	34
6.6 USE CASE 1: RECORD - RAW MODE/ETHERNET STREAM	39
6.6.1 Hardware switch configuration Mode	39
6.6.2 Software GUI Configuration Mode	41
6.7 USE CASE 2: RECORD - MULTI MODE/ETHERNET STREAM	43
6.7.1 Hardware switch configuration Mode	43
6.7.2 Software GUI Configuration Mode	44
6.8 USE CASE 3: RECORD - RAW MODE/SD CARD STORAGE	46
6.8.1 Hardware switch configuration Mode	46
6.8.2 Software GUI Configuration Mode	47
6.9 USE CASE 4: RECORD - MULTI MODE/SD CARD STORAGE	48
6.9.1 Hardware switch configuration Mode	48
6.9.2 Software GUI Configuration Mode	49
6.10 USE CASE 5: PLAYBACK.....	50
6.10.1 Hardware switch configuration Mode.....	50
6.10.2 Software GUI Configuration Mode.....	51
7 Errors	52
8 Troubleshooting	53
9 Limitations	53

List of Tables

Table 1 : Acronyms 5

Table 2 : Accessories..... 8

Table 3 : Power Selection Switch Settings 9

Table 4 : Push button Settings..... 9

Table 5 : User Switch1 Settings 9

Table 6 : Switch2 Settings10

Table 7 : LED Status11

Table 8 : FPGA Configuration Mode12

Table 9 : Default value stored in FPGA registers30

Table 10 : Theoretical Throughput53

List of Figures

Figure 1: DCA1000EVM - TOP View	6
Figure 2: DCA1000EVM - BOTTOM View.....	7
Figure 3: Screenshot of the device Manager.....	12
Figure 4: FPGA Config Switch	12
Figure 5: Hardware Setup	13
Figure 6: FTDI USB Cable Detection	13
Figure 7: FPGA Device Scan	14
Figure 8: Select JTAG option and bit file	14
Figure 9: FPGA SPI Flash Programming	15
Figure 10: FT_Prog Application Window	16
Figure 11: Scan Device.....	16
Figure 12: Apply Template	17
Figure 13: Program Device.....	17
Figure 14: Write successful	18
Figure 15: Test Setup Block Diagram	22
Figure 16: Board Test Setup.....	23
Figure 17: COM ports	24
Figure 18: Radar Studio Tool.....	25
Figure 19: Radar Studio Connection Settings.....	26
Figure 20: Radar Studio StaticConfig Settings	26
Figure 21: Radar Studio DataConfig Settings.....	27
Figure 22: Radar Studio ContStream Settings.....	27
Figure 23: Run Script in Lua Shell	28
Figure 24: Radar Studio TestSource Settings.....	29
Figure 25: Radar Studio SensorConfig Settings.....	29
Figure 26: DCA1000EVM GUI - SYSTEM WINDOW	34
Figure 27: DCA1000EVM GUI - CONFIG WINDOW	35
Figure 28: DCA1000EVM GUI - RECORD WINDOW	36
Figure 29: DCA1000EVM GUI - PLAYBACK WINDOW.....	37
Figure 30: DCA1000EVM GUI - HELP MENU	38

1 Introduction

1.1 Purpose

This document brings details of test setup and accessories required for testing DCA1000EVM along with test procedure for board bring up & Functional Testing.

1.2 Terms/Acronyms / Abbreviations

Acronym	Definition
BOM	Bill of Materials
BUB	Board Bring Up
DNI	Do Not Install
FPGA	Field Programmable Gate Array
JTAG	Joint Test Action Group
GUI	Graphical User Interface

Table 1: Acronyms

1.3 Bibliography

- DCA1000EVM Schematics and BOM
- DCA1000EVM board files
- FPGA requirements_V3

2 DCA1000EVM and Accessories

DCA1000EVM - TOP View

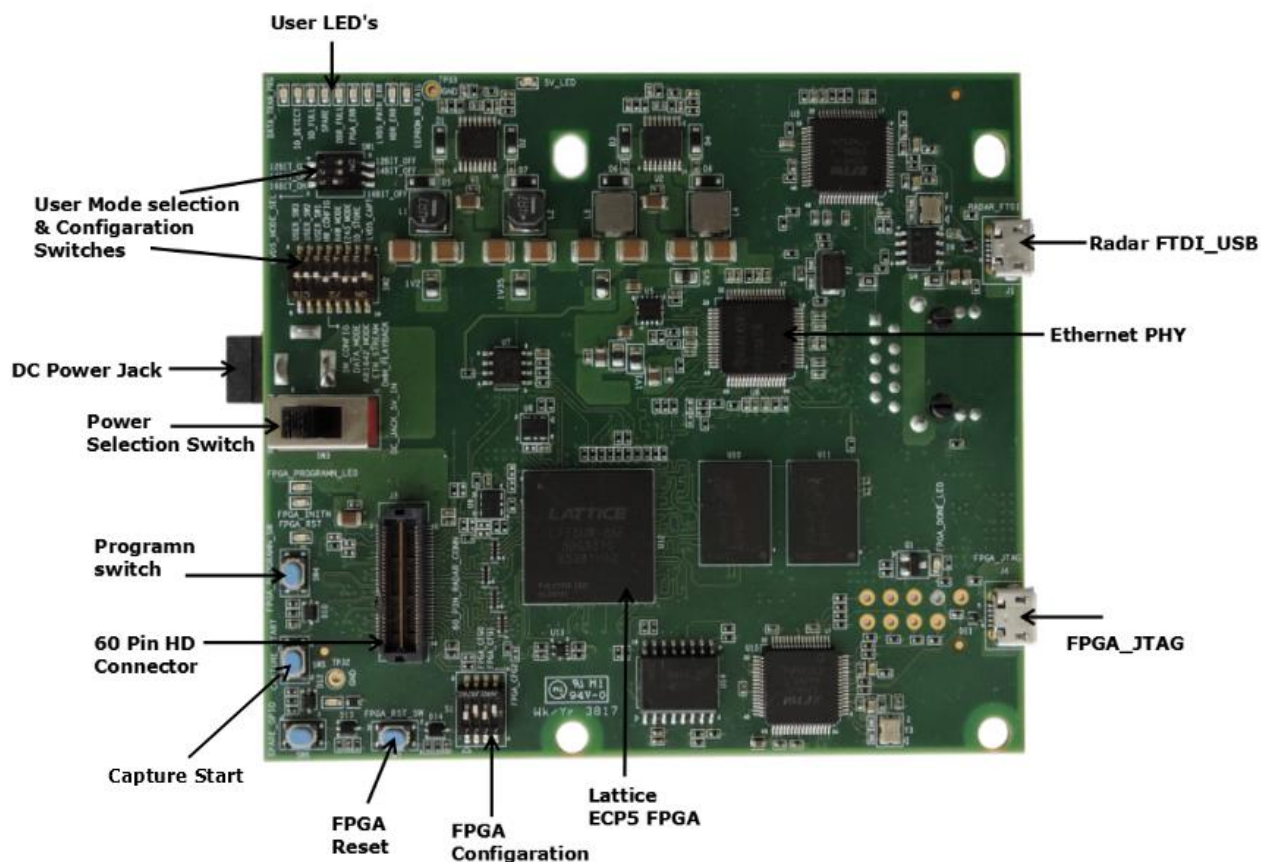
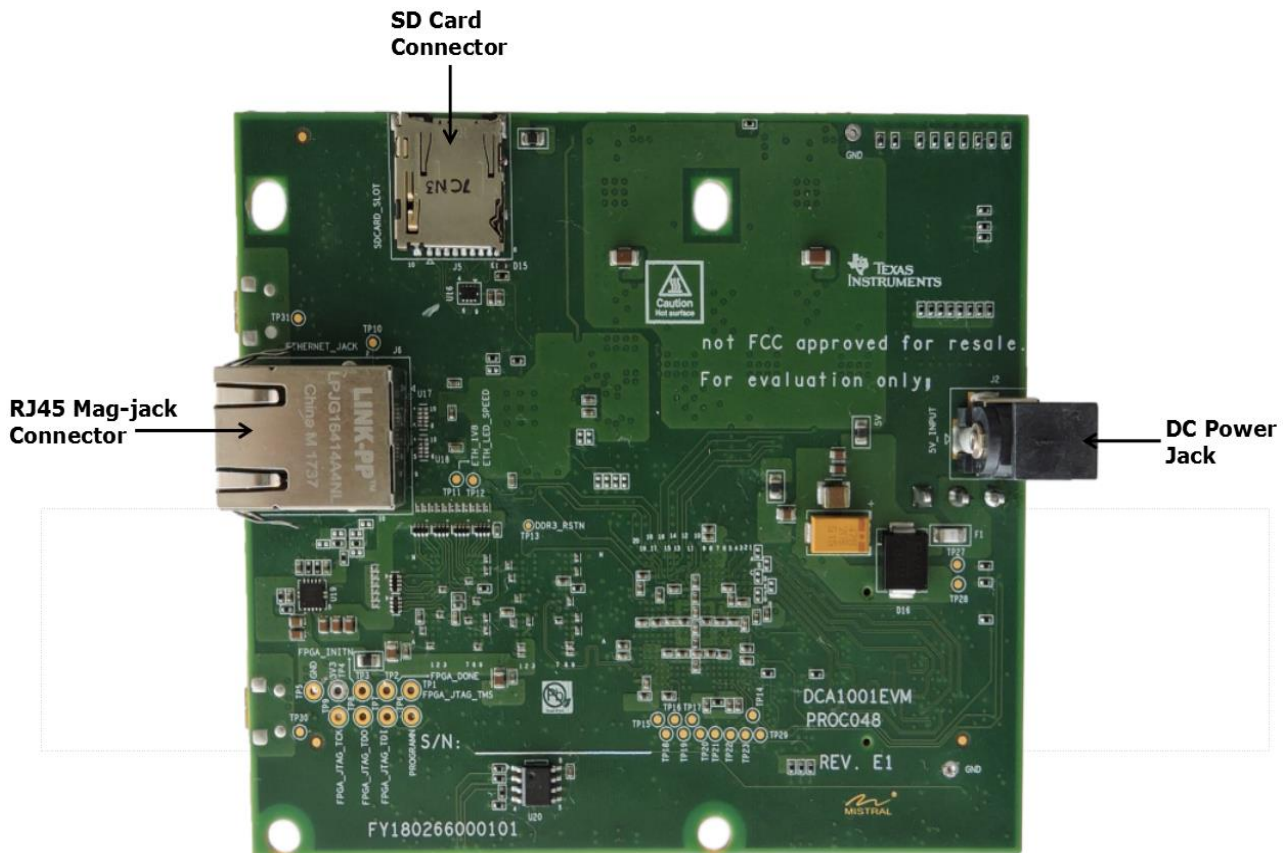






Figure 1: DCA1000EVM - TOP View

DCA1000EVM - BOTTOM View**Figure 2: DCA1000EVM - BOTTOM View**

2.1 Accessories for Board Testing

SL NO	ACCESSORIES		QUANTITY
Table 2:			
1	5V,2A Power Adapter		1
2	Micro SD card (SDSDB-008G-B35)		1
3	Micro USB-B Cable		1
4	Ethernet Cable * (N201-005-GY)		1
5	Samtec coax micro ribbon cable * (HQCD-030-02.00-SEU-TBR-1)		1
6	xWR1xxx EVM		1
7	MSO		1
8	Multimeter		1
9	A. Spacers, Screws, Washers.* B. L-Brackets		1 Set

* This will be available in DCA1000EVM Kit

2.2 Software Requirements for testing

1. Lattice Diamond Programmer tool
2. FTDI - FT_Prog tool
3. Radar Studio tool
4. Windows Test PC

2.3 Test Package

1. DCA1000EVM - FPGA Programming files
2. FTDI EEPROM Programming files
3. Radar EVM Binaries
4. API Frame-work DLL
5. DCA1000EVM GUI

2.4 Default Switch & push button Settings on DCA1000EVM

Reference	Usage	Description
SW3.1 Table 3	Power selection Switch	When the switch is closed to position 1, then the external 5-V power is given through a DC Jack.
SW3.2 Power	Power selection Switch	When the switch is closed to position 3, then the 5-V power is derived from the xWR1xxx EVM through the 60 pin connector. Note: To get 5-V power from the xWR1xxx EVM, the R34 (0Ω) Resistor should be mounted in the xWR1xxx EVM.

Selection Switch Settings

Reference	Usage	Description
SW4 on	PROGRAMN	Whenever SPI Flash is re-flashed or re-programmed, press the PROGRAMN (SW4) button to program the FPGA from SPI Flash. Note: A Binary file should be present in the SPI Flash.
SW5 Set	Capture start	Whenever DCA1000EVM is in hardware configuration mode, Press Capture start(SW5) button to initiate the data transfer. Press the button(SW5) again to stop the data transfer.
SW6 n	SPARE_GPIO_SW	Future Use
SW7 g	FPGA_RST	Press the FPGA_RST button to reset the FPGA.

LVDS bit Setting Switch	SW1 (3 Position Switch)			Description
	1	2	3	
LVDS 12-bit Mode	ON	OFF	OFF	LVDS 12 bit data is captured from xWR1xxx EVM
LVDS 14-bit Mode	OFF	ON	OFF	LVDS 14 bit data is captured from xWR1xxx EVM
LVDS 16-bit Mode	OFF	OFF	ON	LVDS 16 bit data is captured from xWR1xxx EVM

Table 4: User Switch1 Settings

Reference	Usage	Description
SW2.1	LVDS_CAPTURE	When positioned at 1(Pin1), the DCA1000EVM is in Capture mode. In this mode, data is received over LVDS and streamed through Ethernet to the PC.
	DMM_PLAYBACK	When positioned at the other side (Pin16), the DCA1000EVM is in Play Back mode
SW2.2	SD_STORE	When positioned at 2 (Pin2), the DCA1000EVM is in Capture mode, and data is saved into the SD Card
	ETH_STREAM	When positioned at the other side (Pin15), the DCA1000EVM is in capture mode, and data is streamed over the network to the Host PC
SW2.3	1243_MODE	When positioned at 3 (Pin3), the DCA1000EVM captures 4-lane LVDS data
	AR1642_MODE	When positioned at the other side (Pin14), the DCA1000EVM captures 2-lane LVDS data
SW2.4	RAW_MODE	When positioned at 4 (Pin4), the DCA1000EVM is in raw mode. In this mode all LVDS data is captured as it is.
	DATA_MODE	When positioned at the other side (Pin13), the DCA1000EVM is in Data Separated mode. In this mode , the user can add specific headers to different data types, and FPGA separates out different data types based on the header.
SW2.5	CONFIG_VIA_HW	When positioned at 5 (Pin5), DCA1000EVM is configured as per the settings of Hardware DIP switch SW2 . In Hardware configuration mode, Software configurations are ignored
	CONFIG_VIA_SW	When positioned at the other side (Pin12), The DCA1000EVM is configured through the Software configuration commands over Ethernet. In Software configuration mode, Hardware DIP settings are ignored.
SW2.6	USER_SW1 (Via EEPROM Configuration)	When positioned at 6(Pin6), the Ethernet configuration data is read from EEPROM and the same is used for Ethernet communication.
	GND (Via FPGA Configuration)	When positioned at the other side (Pin11), the Ethernet configuration data hardcoded in FPGA is used for Ethernet communication.
SW2.7	USER_SW2	Future use
	GND	
SW2.8	USER_SW3	Future use
	GND	

Table 5: Switch2 Settings

2.5 Default LED Status on DCA1000EVM

Reference	Usage	Description	Color
LD1	DATA_TRANS_PROG_LED1	This LED indicates the progress of data transfer, either through the network or saving into the SD Card	Green
LD2	SD_DETECT_LED5	When the SD card is mounted into the DCA1000EVM, this LED indicates whether the SD card is detected or not.	Green
LD3	SD_FULL_LED0	This LED indicates the full condition of the SD Card	Red
LD4	SPARE_LED8	Future Use	Green
LD5	FPGA_ERR_LED2	When LVDS data is coming at high rate, the internal FPGA buffer gets full. In that case, this LED glows.	Red
LD6	DDR_FULL_LED7	This LED indicates the full condition of the 2GB DDR3 memory.	Red
LD7	LVDS_PATH_ERR_LED3	This LED indicates whether the xWR1xxx EVM is sending LVDS data or not.	Red
LD8	HEADER_ERR_LED4	This LED indicates the failure of the header when the DCA1000EVM is in Data Separated mode	Red
LD9	EEPROM_RD_FAIL_LED6	This LED indicates an EEPROM read access failure	Red
LD10	5V	This LED indicates the presence of the 5-V supply in the DCA1000EVM	Green
LD11	PROGRAMN	This LED indicates assertion of the program to FPGA when the PROGRAMN button is pressed.	Red
LD12	INITN	This LED indicates that the FPGA is ready to be configured	Red
LD13	FPGA_RST	This LED indicates the assertion of the FPGA Reset	Red
LD14	DONE	When FPGA is programmed from SPI flash, this LED glows.	Green
LD15	FTDI_JTAG_ACT	This LED indicates activity on the FPGA JTAG connectivity.	Green

Table 6: LED Status

3 FPGA Programming

For FPGA programming, install Lattice Diamond Programmer Standalone tool in the PC as per the following steps.

1. Download the Lattice Diamond programmer tool, Windows version (Programmer Standalone 3.12 SP1 64-bit for Windows) from the following link, <https://www.latticesemi.com/en/Products/DesignSoftwareAndIP/FPGAandLDS/LatticeDiamond>.
2. A new Folder named Iscc will be created at the path of installation.

Post installation steps:

1. Connect the USB cable to the FPGA JTAG USB port (**J4**) and power on the board.
2. The FTDI cable driver is installed automatically. Otherwise, manually select the cable driver from the installation path.
1. The device manager will recognize two COM ports as shown in [Figure 3](#).

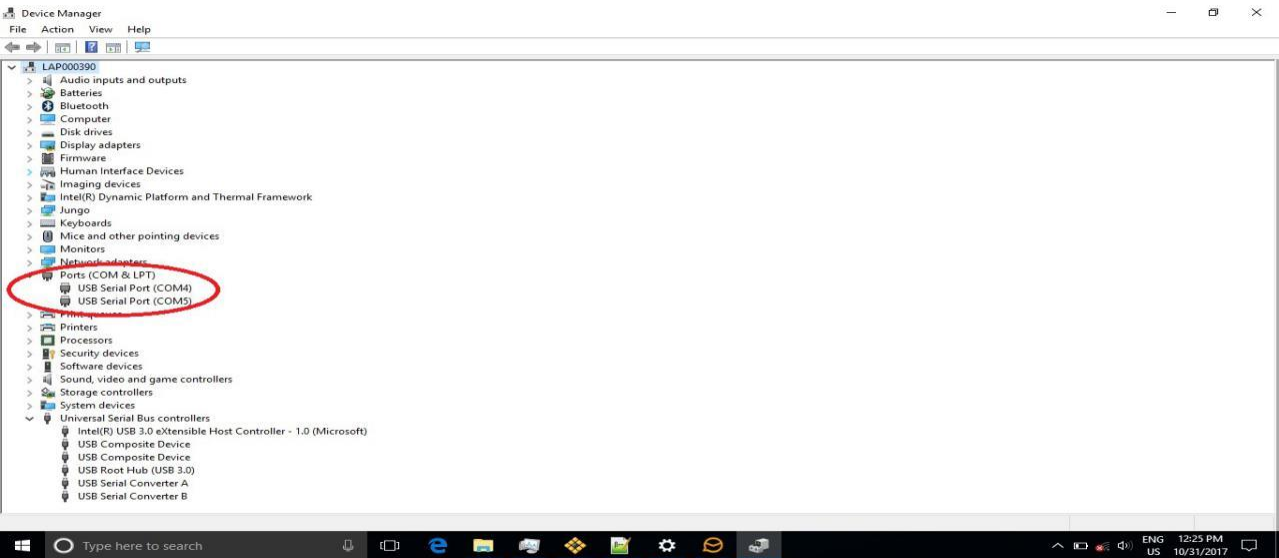


Figure 3: Screenshot of the device Manager

The DCA1000EVM supports four DIP switch (**S1**) options for the FPGA programming configuration, as listed in [Table 8](#).

Table 7: FPGA Configuration Mode

Configuration Mode	CFG [2:0]	S1.3	S1.2	S1.1
Master SPI	010	Pin5 (FPGA_CFG2)	Pin6 (FPGA_CFG1)	Pin7 (FPGA_CFG0)



Figure 4: FPGA Config Switch

3.1 Hardware Setup

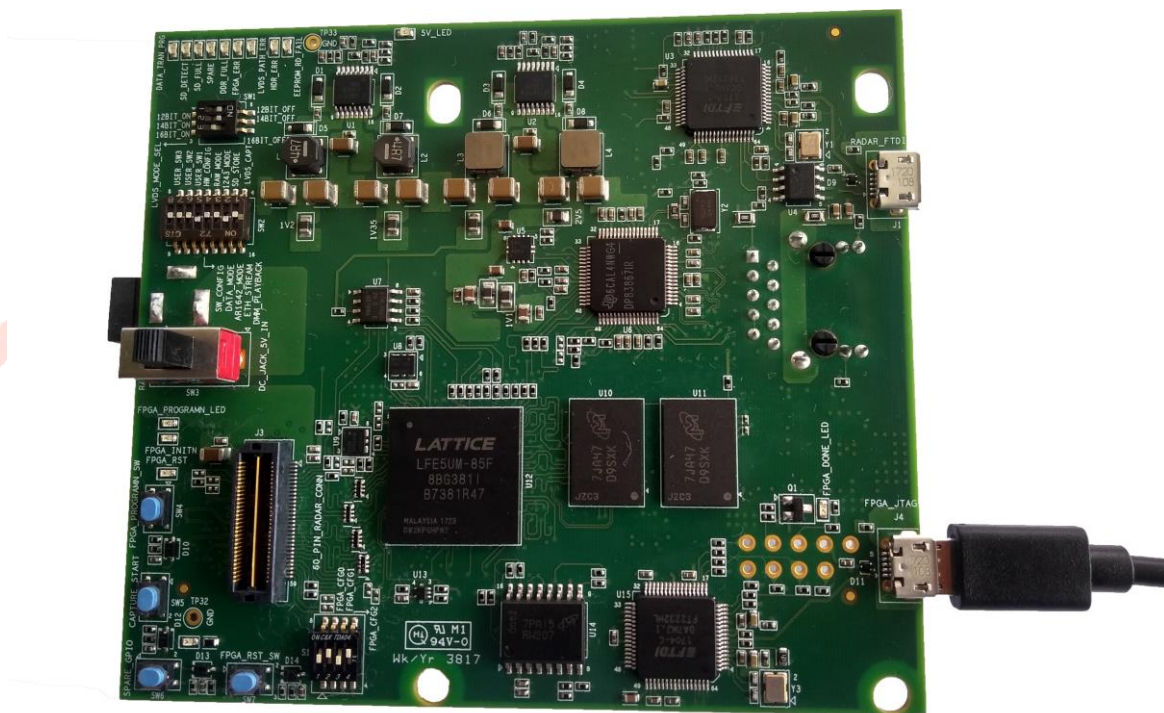


Figure 5: Hardware Setup

3.2 FPGA - SPI Flash programming

1. Open Lattice Diamond programmer tool from the Start menu and select Detect cable. Click OK as shown in [Figure 6](#),

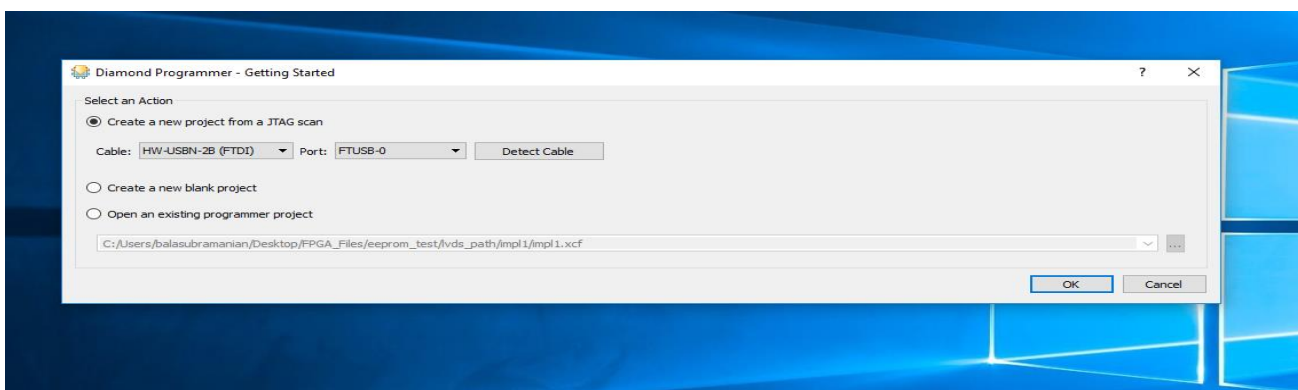


Figure 6: FTDI USB Cable Detection

2. FPGA device(**LFE5UM-85F**) is detected in the programmer window as shown in [Figure 7](#),

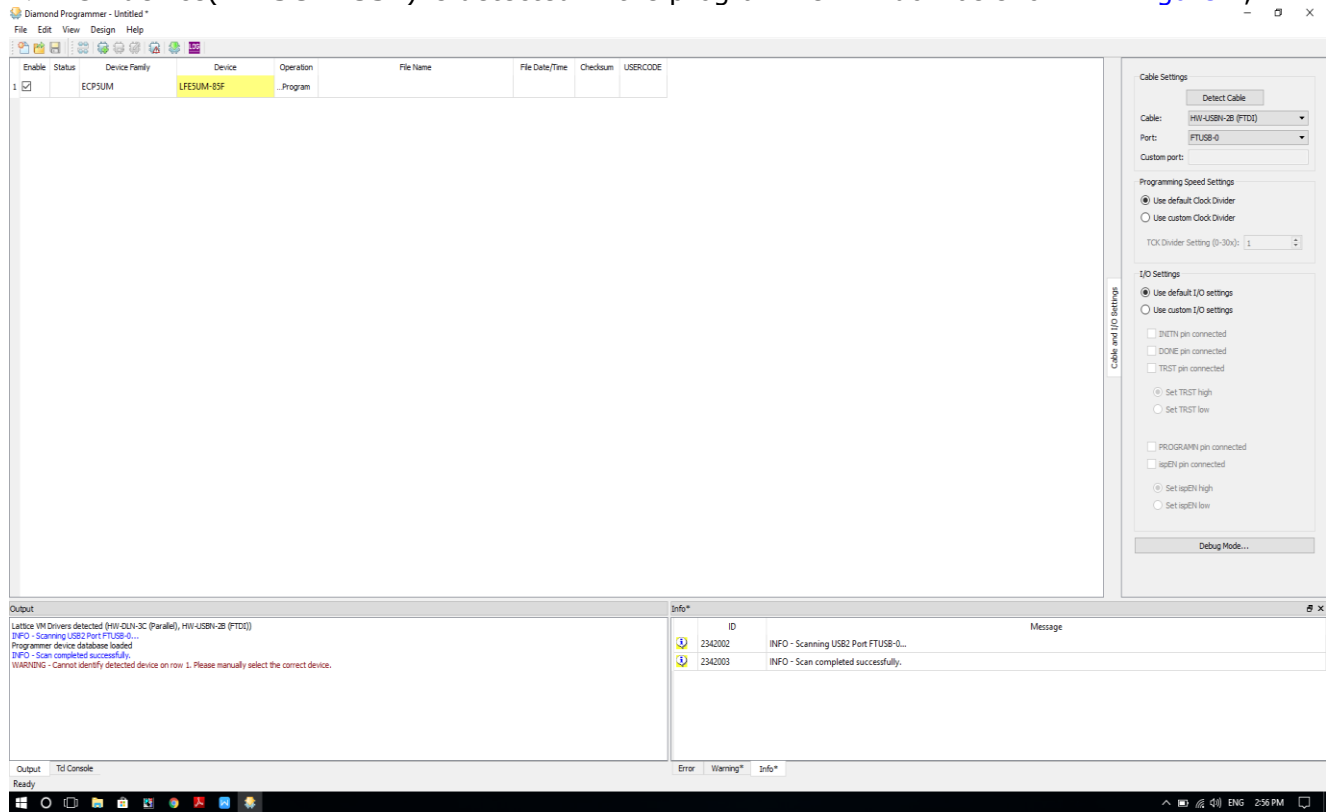


Figure 7: FPGA Device Scan

3. If the command line prompts to select the device manually, then click on the Device column and select the LFE5UM-85F device.

4. Click operation tab and select the Access mode options as JTAG 1532 mode; and for Operation select "Erase Only" for the Erase the bit file in the FPGA, as shown in [Figure 8](#).

5. Click the "Program" option to complete the process.

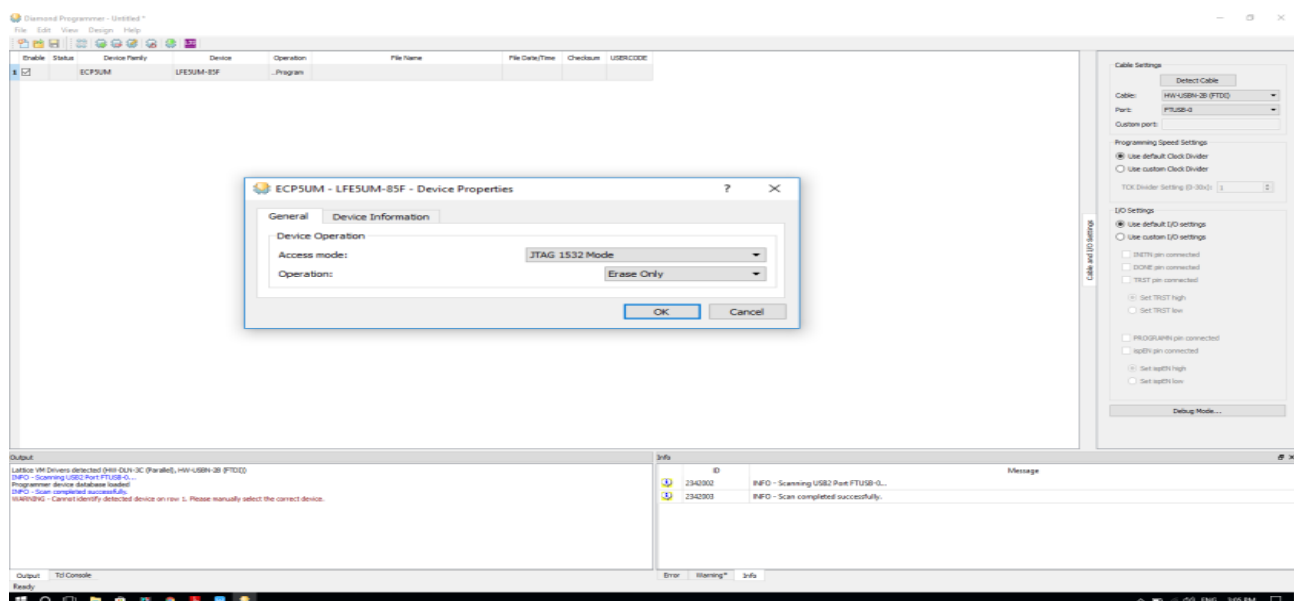


Figure 8: Select JTAG option and bit file

6. Click the operation tab to open the window, as shown in [Figure 9](#),
 - a) Set the Access mode option as **SPI Flash Background Programming**
 - b) Set the Operation as **"SPI Flash Erase, Program, Verify"** for bit file programming.
 - a) Select the bit file from the following path,
/DCA1000EVM_TP_REL/FPGA/DCA1000_FPGA_RECORD_xx.yy.bit file in programming file option. (xx -> Version No, yy -> Release Date)
 - c) Select SPI Flash device as (**Family: SPI Serial Flash, Vendor: Micron, Device: SPI-N25Q128A, Package: 16-pin SO16**)
 - d) Click OK
7. Click the "Program" option to complete the process.

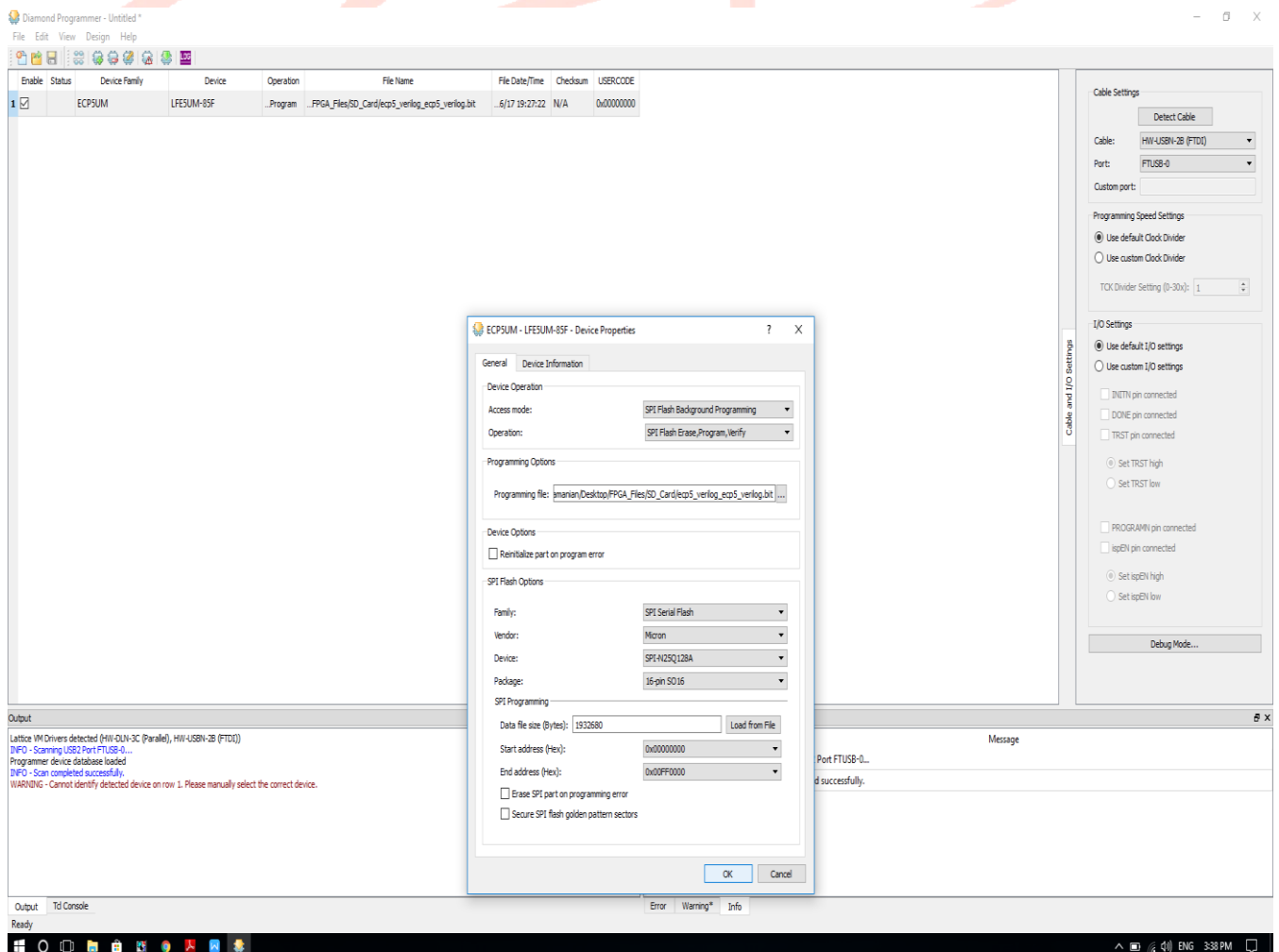


Figure 9: FPGA SPI Flash Programming

8. When the programming is done, press PROGRAMN (**SW4**) button (or) power cycle the board to load the FPGA bit file from SPI Flash.
9. FPGA DONE LED (**LD14**) glows to confirm the successful loading of the FPGA bit file from SPI Flash.

4 FTDI- FT4232 EEPROM

4.1 FT4232HL (Reference: U3) - EEPROM Programming

1. To load the FTDI template download the "FT Prog" tool from the FTDI website (http://www.ftdichip.com/Support/Utilities.htm#FT_PROG) and load the attached template.
2. Connect Micro USB cable to DEBUG USB connector (J1- RADAR FTDI) & other side to PC,
3. Run FT4232 FT_Prog application on Test PC,

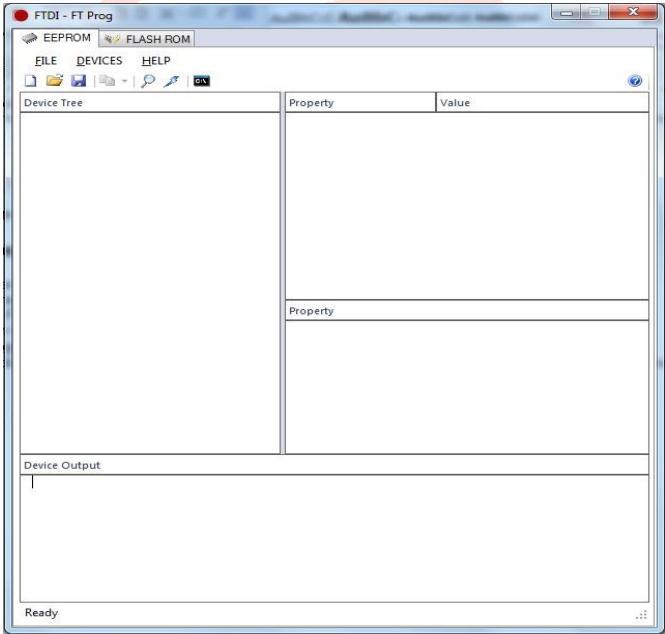


Figure 10: FT_Prog Application Window

4. Select Scan and Parse button,  Blank device will be displayed as shown in Figure 11,

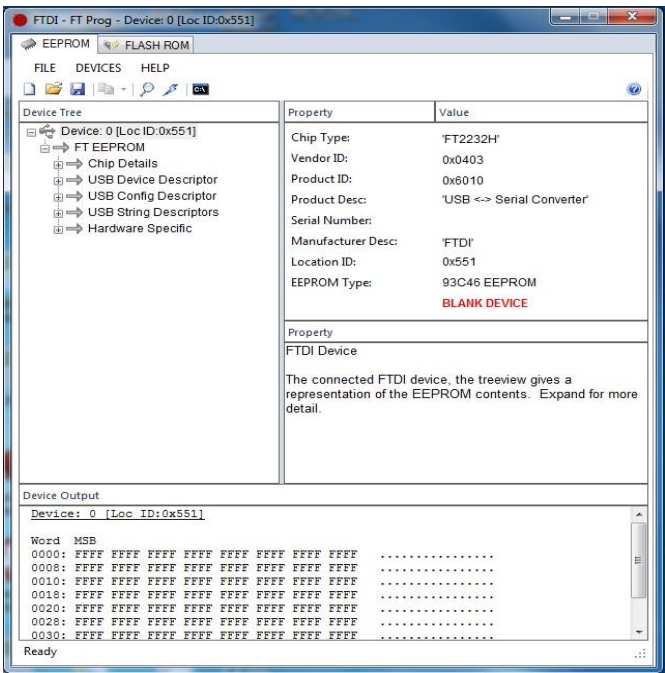


Figure 11: Scan Device

5. Click File -> Open Template and select the template AR-DevPack-EVM-012.xml available in the test package
6. Right click on Device: 0[Loc ID:0xXXX] -> Apply Template -> Select the release folder for the Template file and select the **AR-DevPack-EVM-012.xml** as shown below,

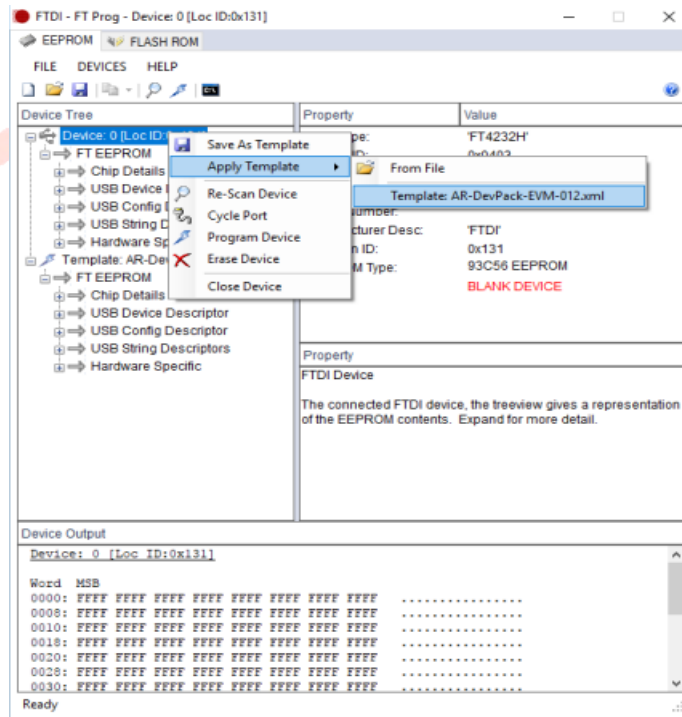


Figure 12: Apply Template

7. Now, select Device: 0[Loc ID:0xXXX] -> Program Device

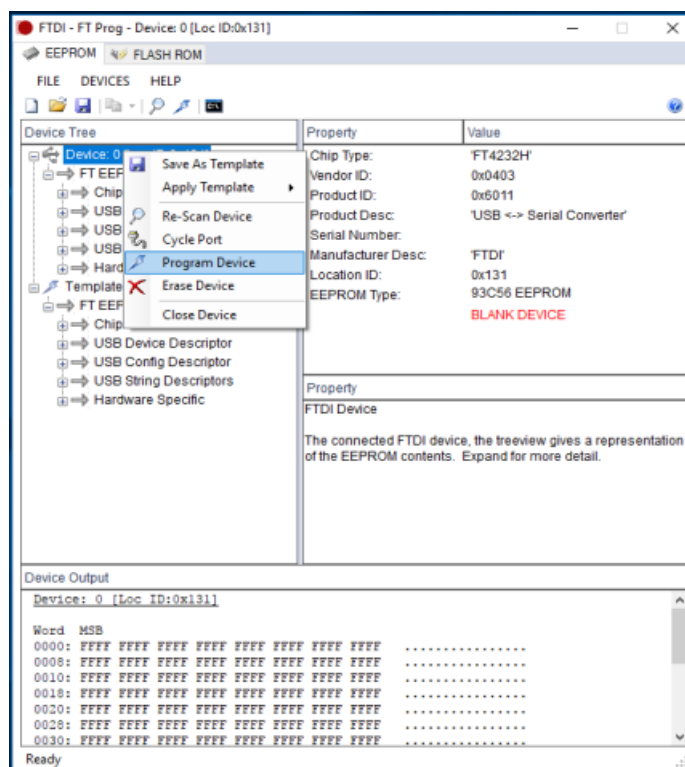


Figure 13: Program Device

8. Check the write successful status as highlighted in [Figure 14](#),

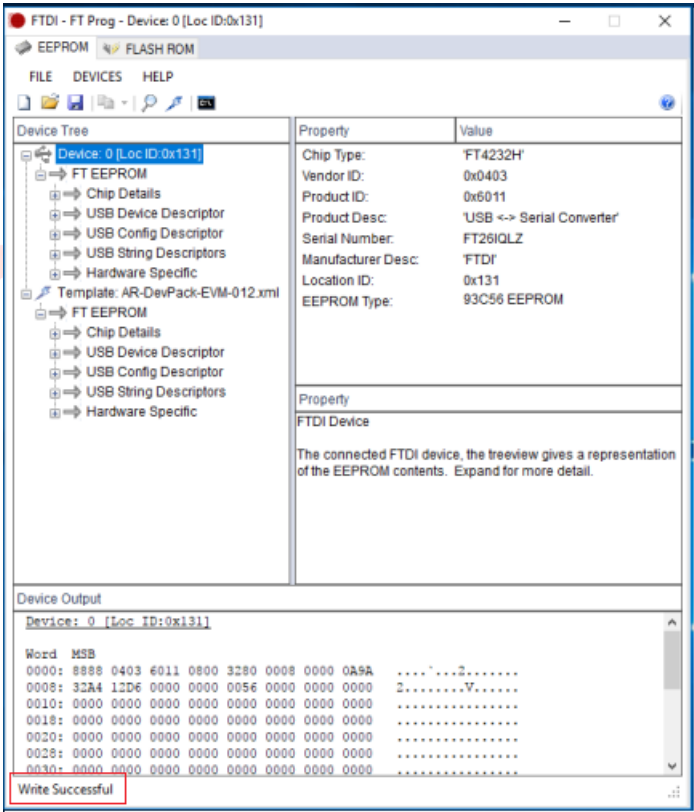
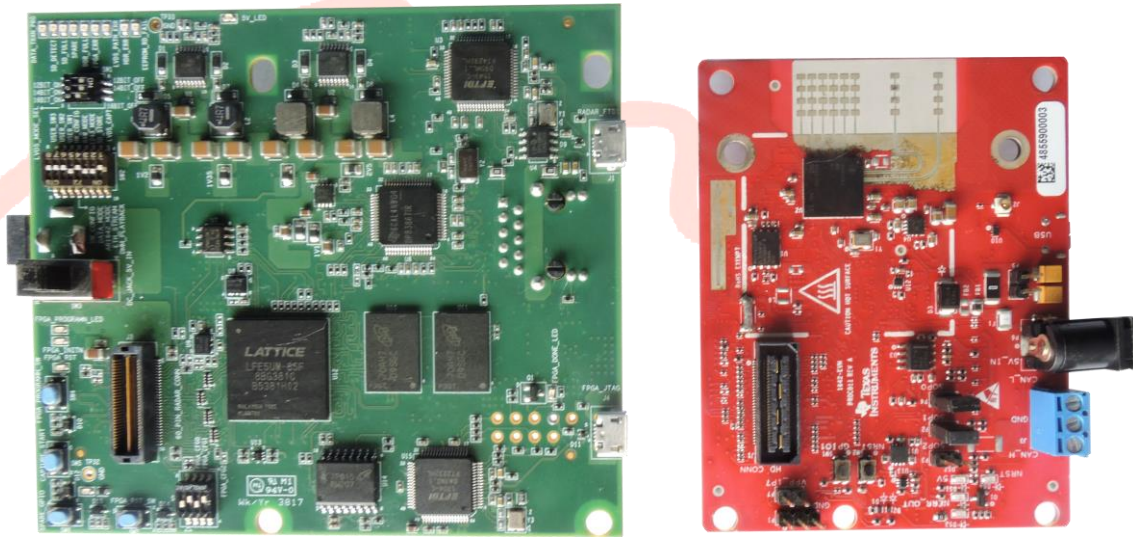


Figure 14: Write successful

5 DCA1000EVM to Radar EVM Board to Board to connection

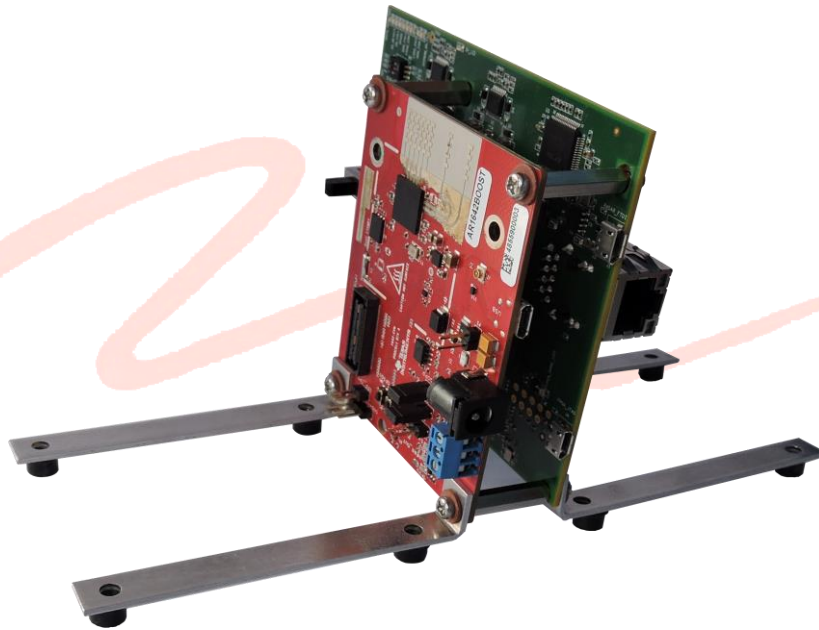
Step 1: Align the DCA1000EVM and Radar EVM boards as shown below



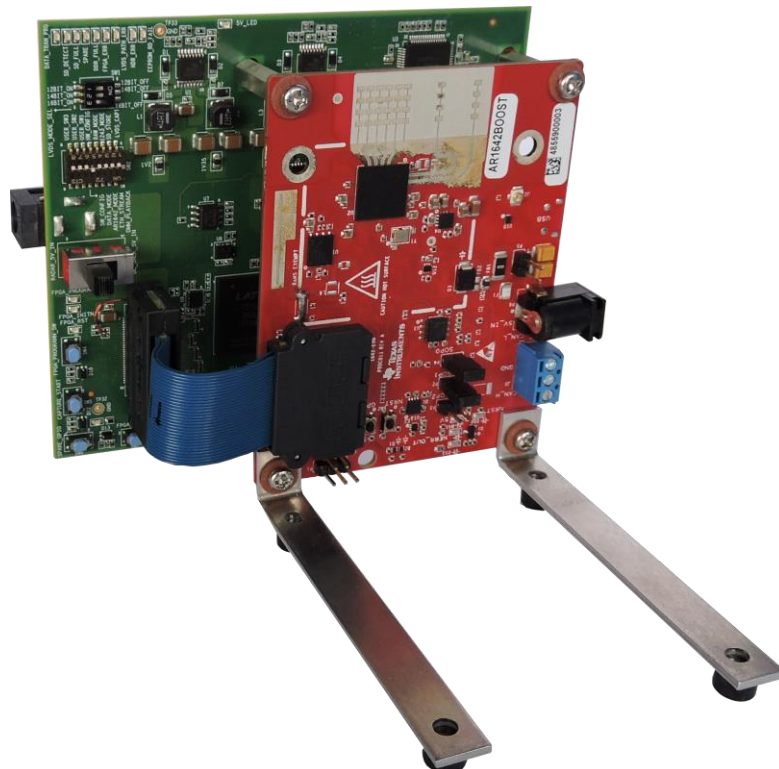
Step 2: Connect the DCA1000EVM with screw and stands as shown below



Step 3: Connect the DCA1000EVM with Radar EVM board as shown below



Step 4: Connect the Samtec Ribbon Cable as shown below.



6 System Testing

6.1 Test Setup:

Pre-Requisites:

1. DCA1000EVM should be connected to Host PC via Ethernet cable to access the GUI and Data Transfer process.
2. DCA1000EVM should be connected to PC via USB Cable (J1-Radar FTDI) for configuring the RADAR EVM by using on board FTDI chip.
3. DCA1000EVM should be connected to PC via USB cable (J4-FPGA JTAG) for programming the FPGA (When card is in programming mode)
4. DCA1000EVM should be connected to TI Radar EVM via 60 pin HD Connector by using 60 pin Samtec ribbon cable.
5. DCA1000EVM power input should be connected either from DC Jack or TI Radar EVM power output (from 60 pin HD connector) by selecting the switch SW3.
6. Radar EVM should be connected to PC via USB Cable (J8) for UART connectivity.
7. For testing, Mistral has used following HOST PC configuration
 - a) Intel Core i5 2.5 GHz or above
 - b) 8.00 GB RAM or above
 - c) Windows 10 Pro operating system
 - d) IP Address - 192.168.33.30
 - e) 1-Gbps Ethernet Port
8. Pre-installed Radar Studio tool in Host PC along with the drivers mentioned in the following link, <http://www.ti.com/tool/mmwave-dfp>.
9. Check for "rfeval" folder in DFP package to find Radar studio tool (\\rf_eval\\radarstudio\\RunTime\\Radar Studio).

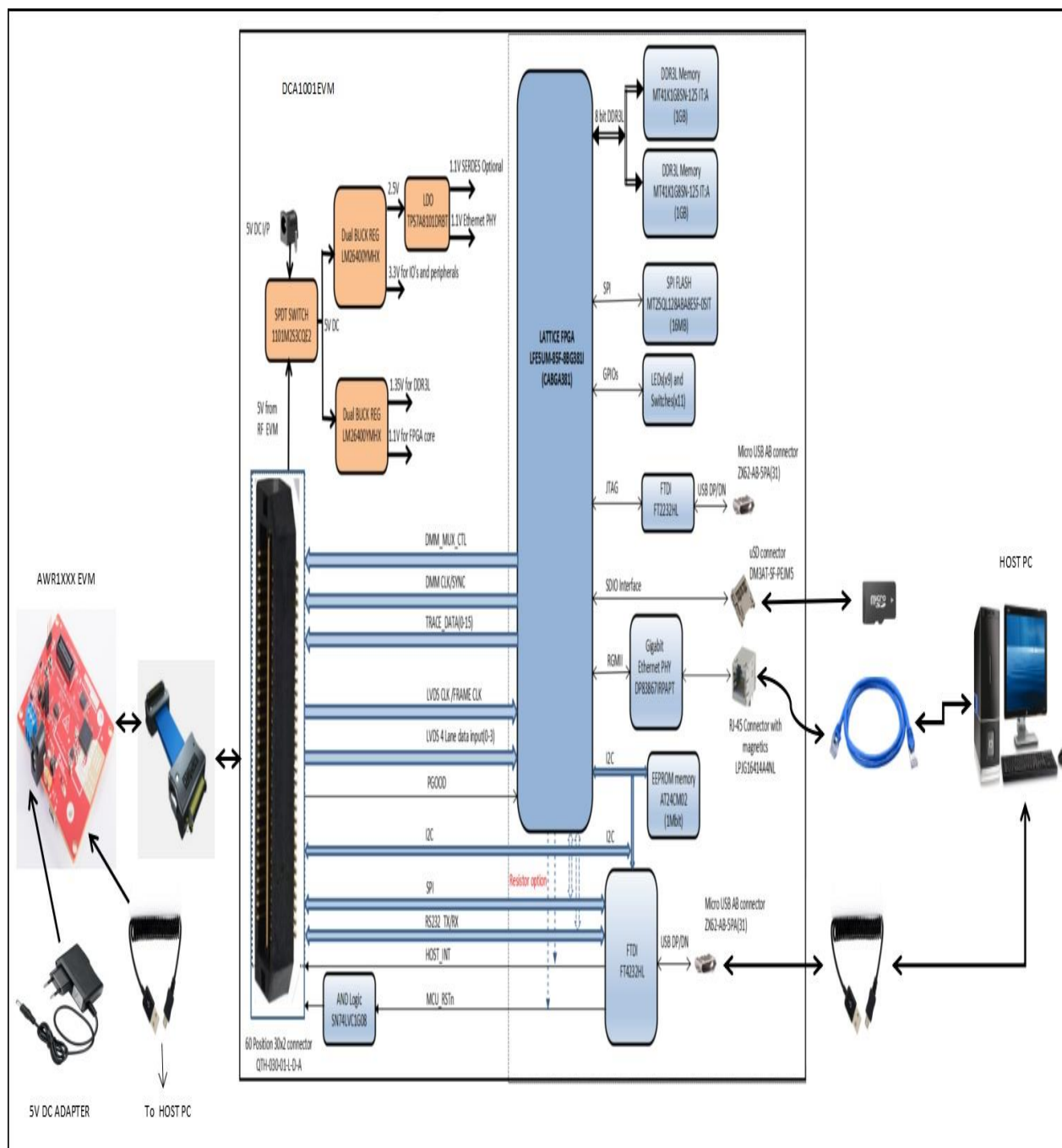


Figure 15: Test Setup Block Diagram

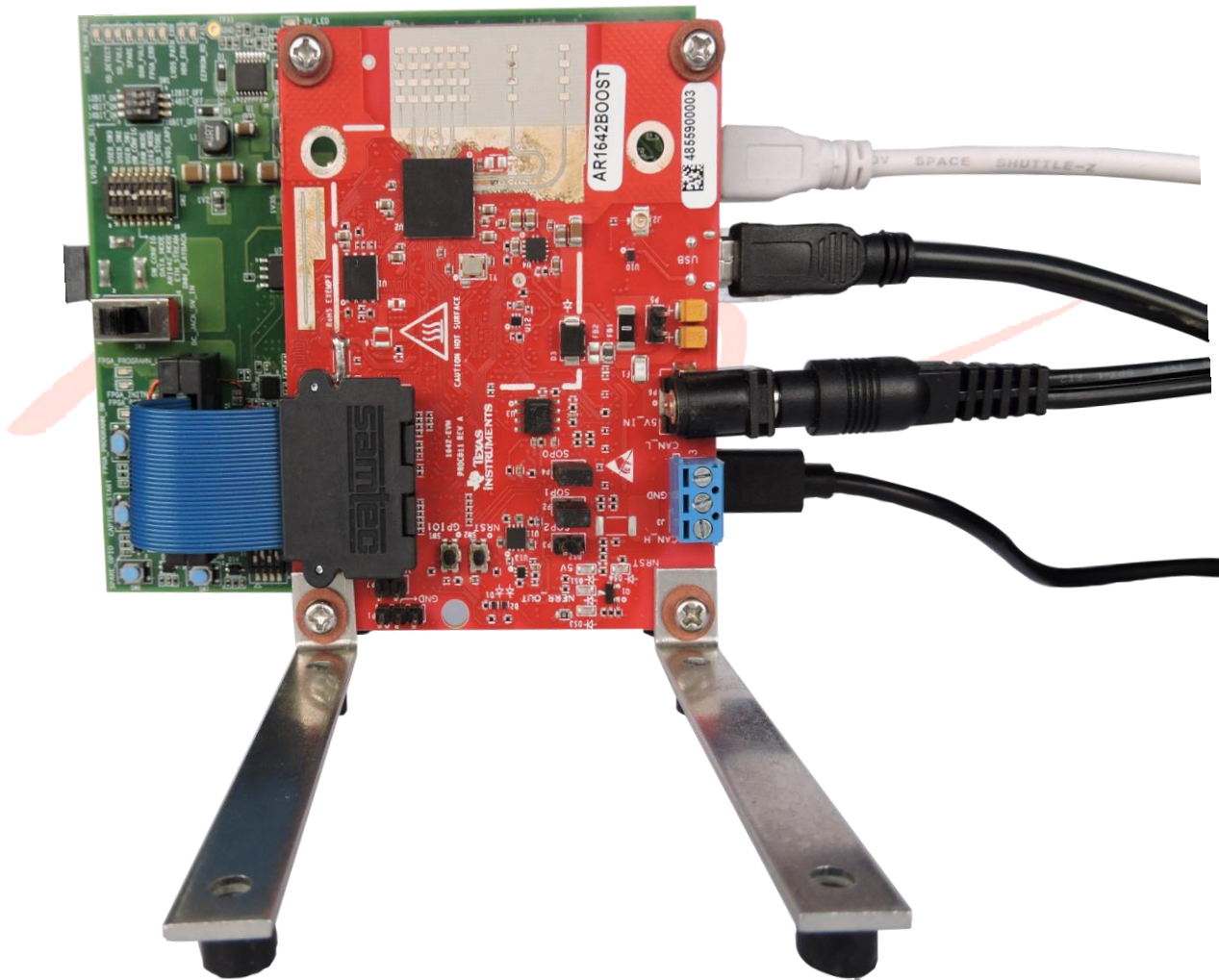


Figure 16: Board Test Setup

6.2 Configuring RADAR EVM by using Radar Studio

To configure Radar EVM to generate either Fixed test pattern generation or Real Time LVDS Data generation as described in 6.2.1 & 6.2.2.

6.2.1 Fixed test pattern generation

- Set the SOP jumper settings in Mode 2 (Development mode) in RADAR EVM.
(SOP2=0, SOP1=1, SPO0=1)
- Power On the board.
- Check the ports to be shown in "Device Manager" as per below [Figure 17](#),

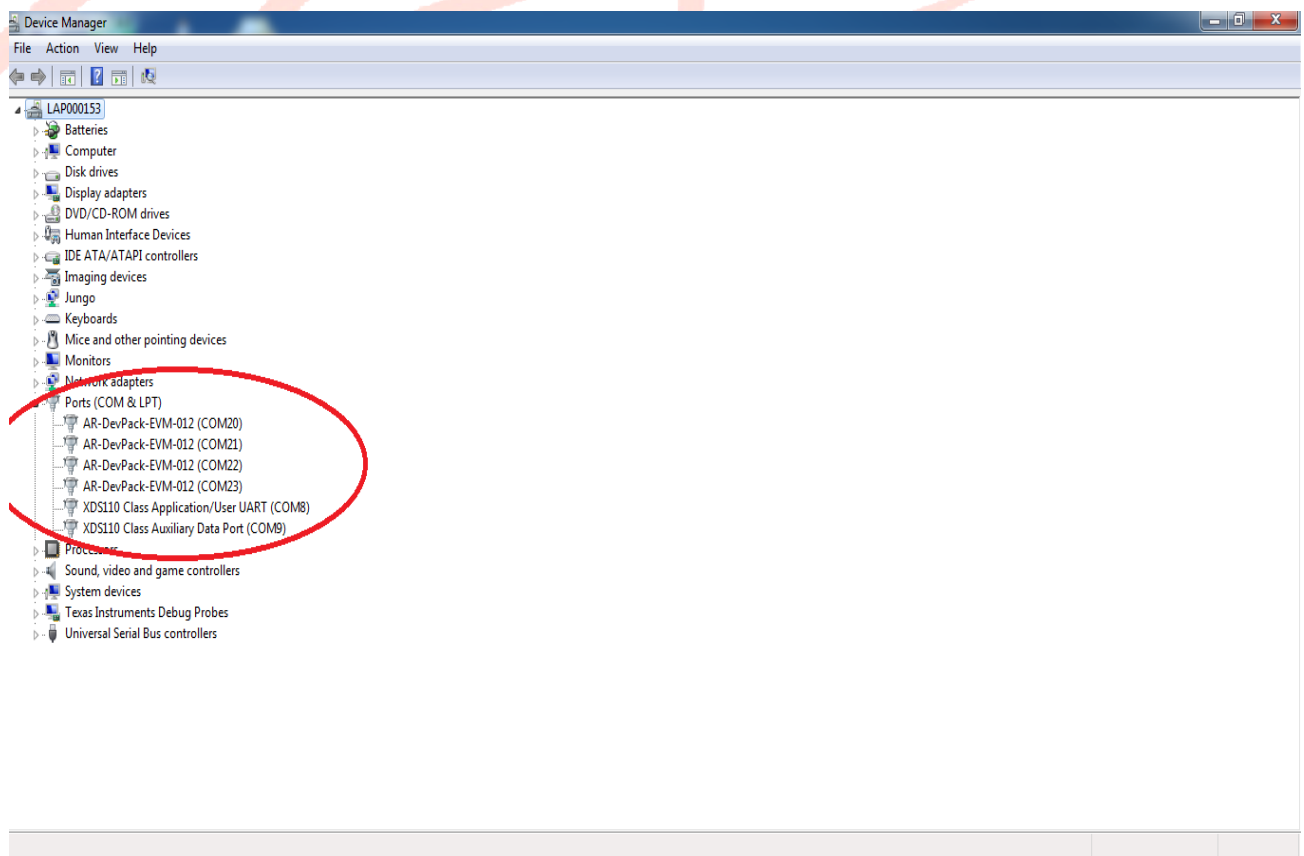


Figure 17: COM ports

- Run Radar Studio Application and Go to "Connection" tab in RadarAPI tab.

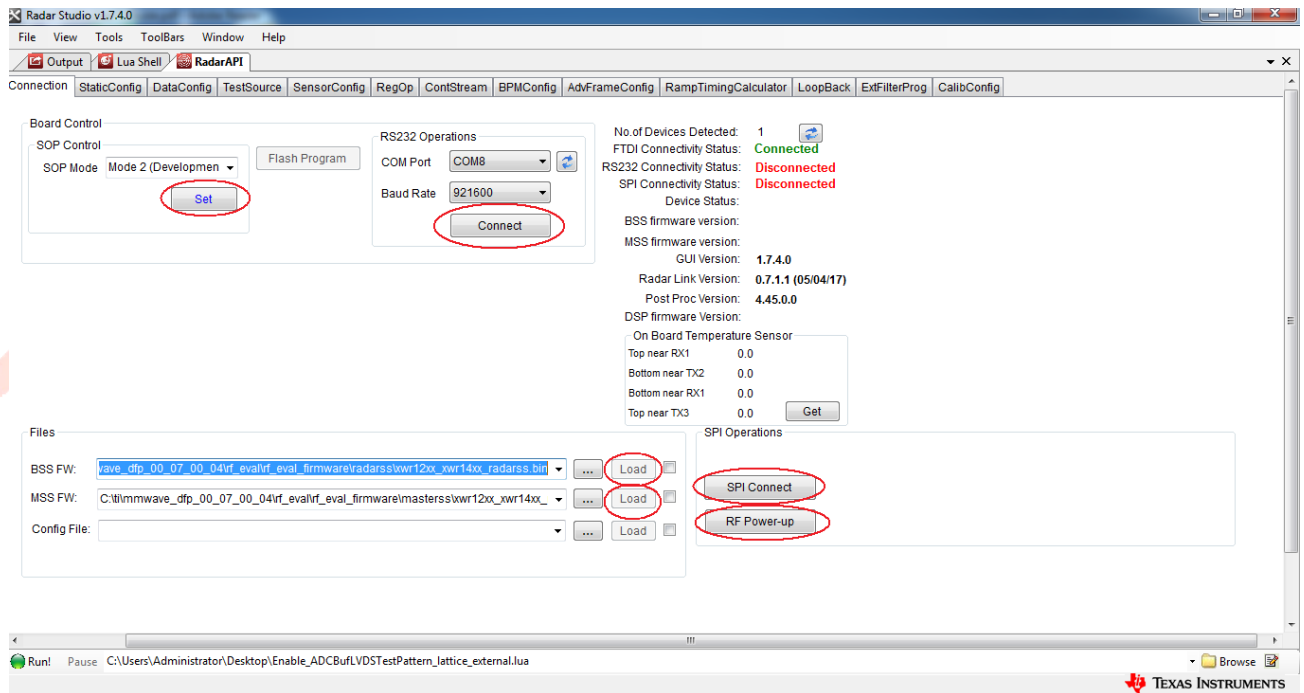


Figure 18: Radar Studio Tool

- Click set option in SOP control as "Mode 2 (Development mode)".
- Click Connect RS232 option (Set appropriate User UART COM port & Baud rate as 921600).
- Select BSS FW for particular AR Device (xWR1243/1443 or xWR1642) from Radar Studio installed folder and Load the file by clicking Load button.
- Select MSS FW for particular AR Device (xWR1243/1443 or xWR1642) from Radar Studio installed folder and Load the file by clicking Load button.
- Click "SPI Connect & RF Power-up" buttons and Check the connectivity completion as per below [Figure 19](#),

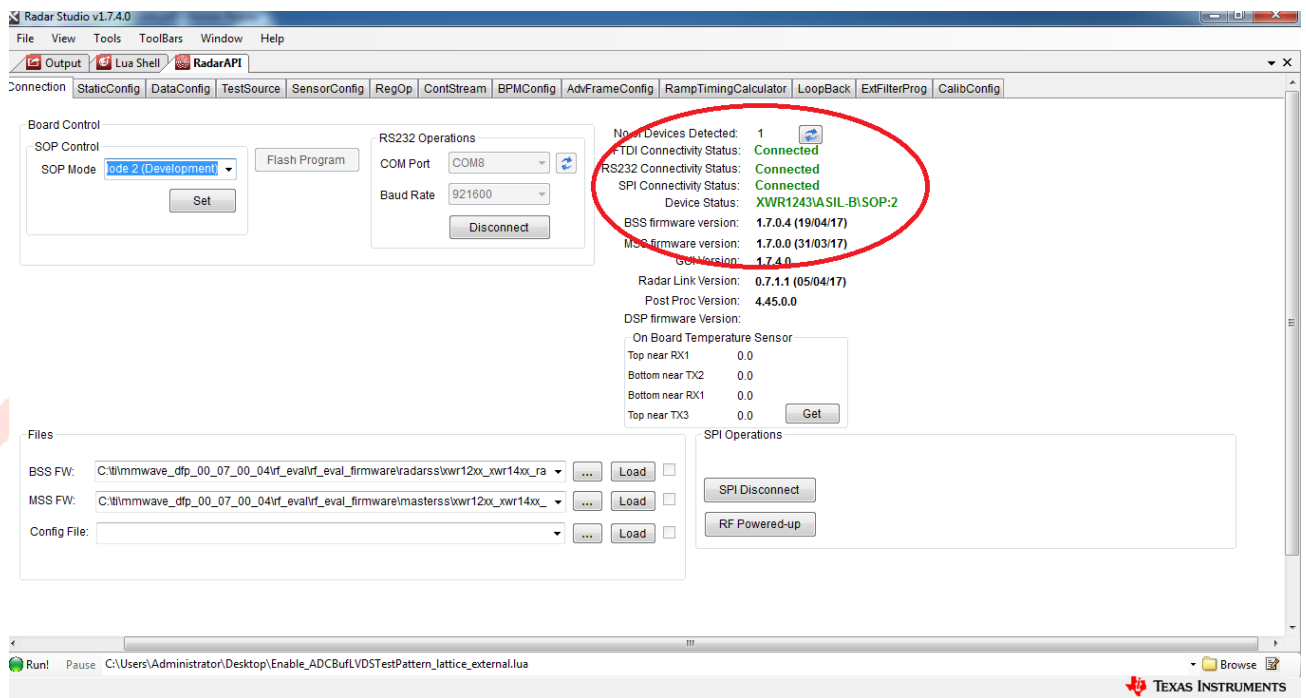


Figure 19: Radar Studio Connection Settings

- Click on "StaticConfig" tab
- Deselect all TX Channels
- Click on 'ADC Config' Set button
- In LP Mode, Set the LP ADC Mode as 'Regular ADC'
- Click on 'RF Init' button.

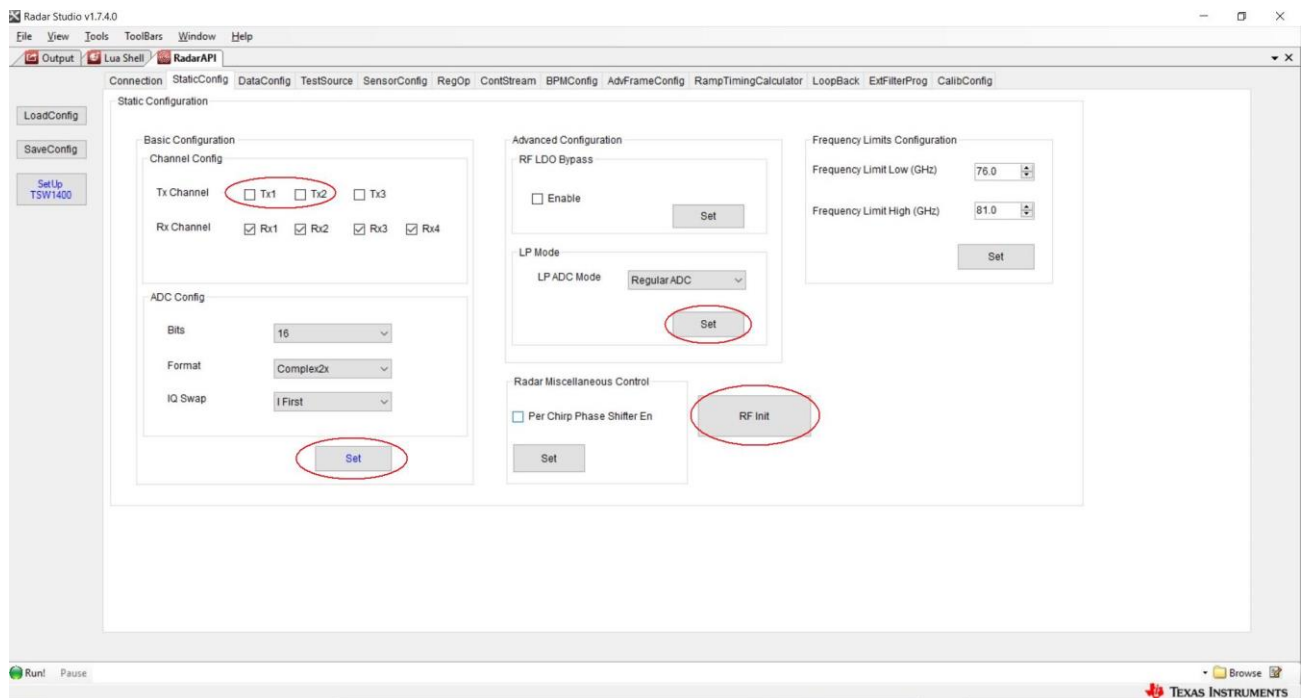


Figure 20: Radar Studio StaticConfig Settings

- Click on "DataConfig" tab,
- Click on 'Data path Configuration' Set button
- Set the Clock Configuration as
 - ◆ Lane Clock --> DDR Clock
 - ◆ Data rates --> 150 /225/300/400/450/600 (Mbps)
- Click on 'Clock Configuration' Set button
- Click on 'LVDS lane Configuration' Set button

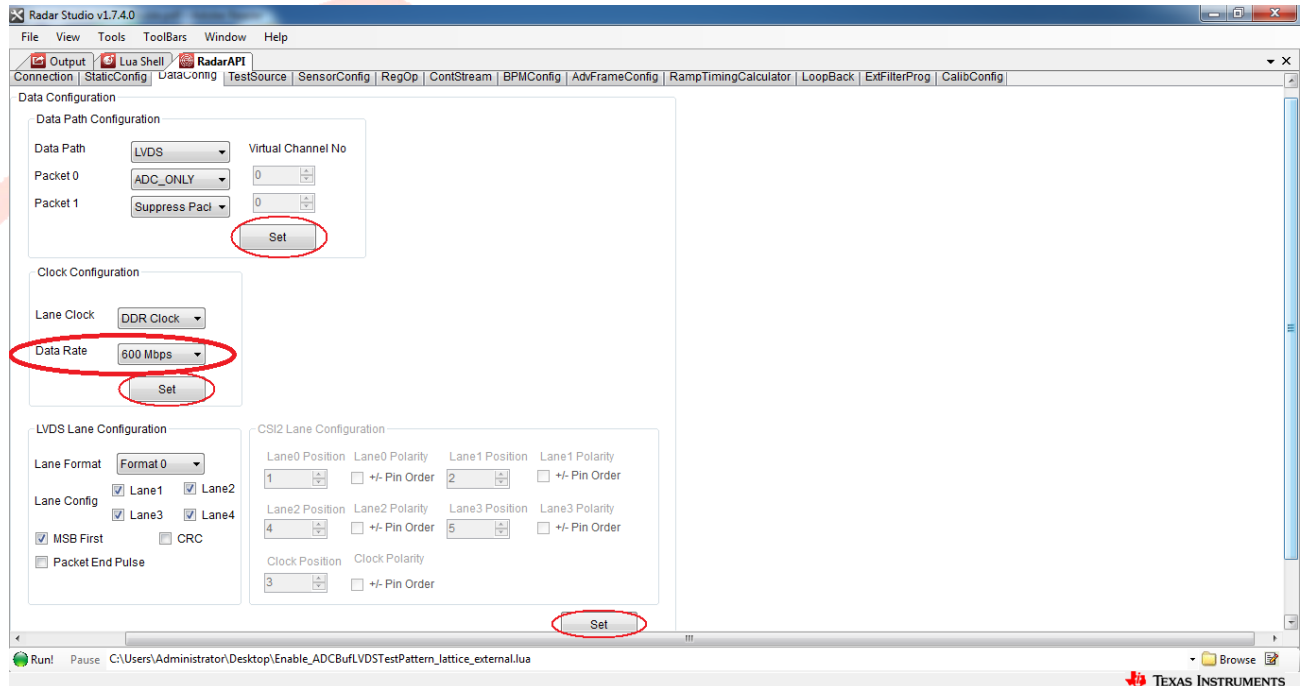


Figure 21: Radar Studio DataConfig Settings

- Click on "ContStream" tab,
- Click on 'ContStreaming' Set button to set default values,
- Click 'Enable' in the StreamEnable option.

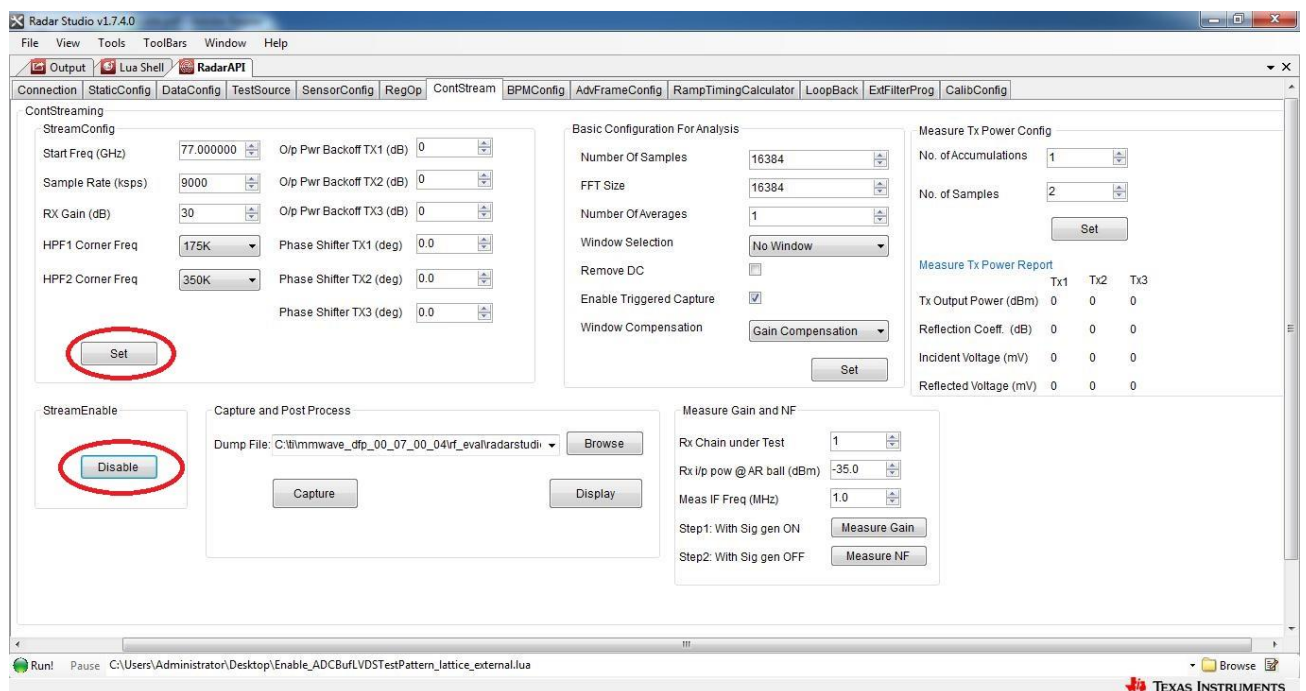


Figure 22: Radar Studio ContStream Settings

- Click on "Lua Shell" tab,
- Load the Lua Commands file (File name: Enable_ADCBufLVDS_TestPattern_lattice_external.lua) for generating the fixed pattern.
- Type "Enable_ADCBufLVDS_TestPattern ()" in the command line and Enter

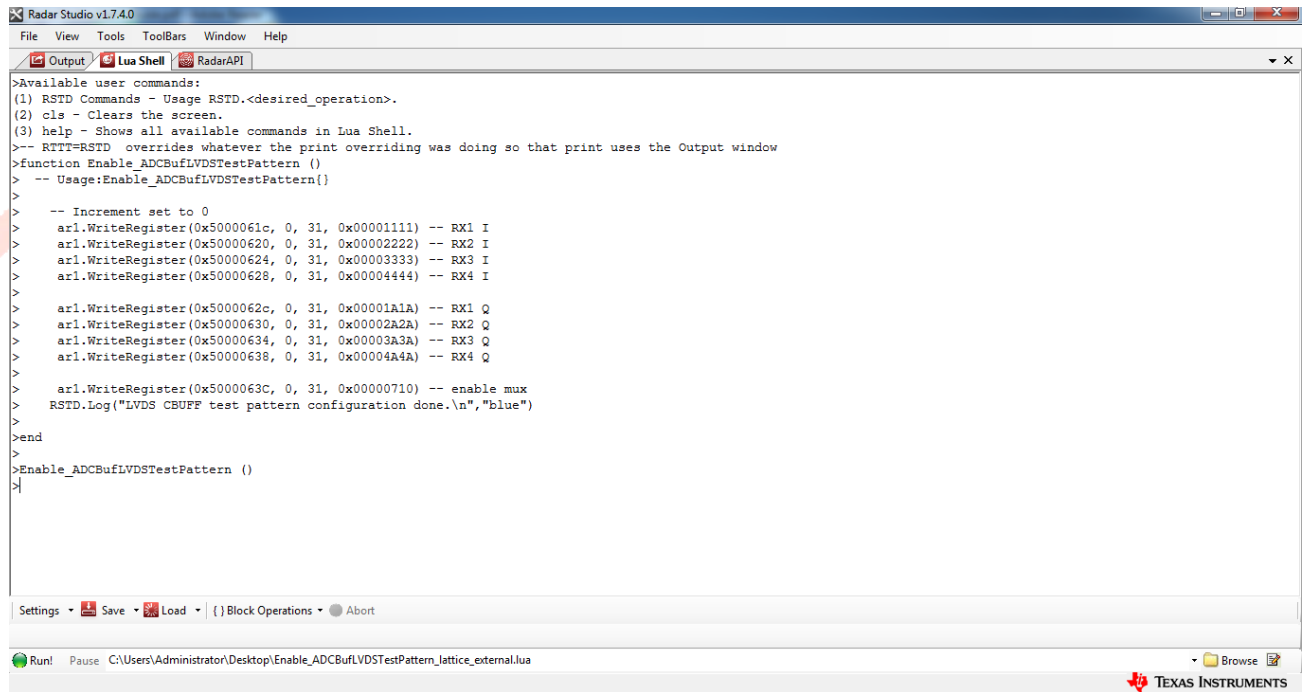


Figure 23: Run Script in Lua Shell

- Once above settings are completed, Radar EVM will initiate LVDS fixed pattern data transfer.

Note: When User want to stop the data transfer, Disable the StreamEnable in the 'ContStream' tab.

6.2.2 Real Time LVDS Data generation

- Follow the same procedure for Connection tab as mentioned in section 8.2.1 Fixed data pattern generation.
- Deselect only 'TX2 Channel' in the StaticConfig tab
- Set 'ADC Config' --> 'LP ADC mode' --> Click 'RF Init'
- Follow the same procedure for DataConfig tab as mentioned in section 8.2.1 Fixed data pattern generation.
- Click on "TestSource" tab
- Set the default TestSource values as mentioned in TestSource window

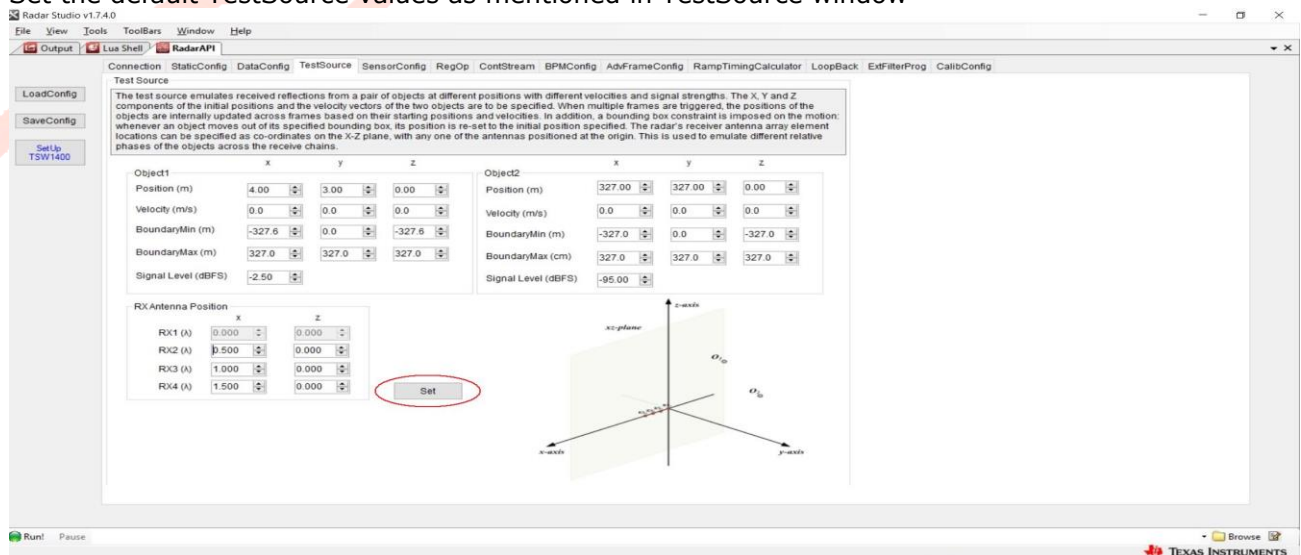


Figure 24: Radar Studio TestSource Settings

- Click on "SensorConfig" tab
- Set Sensor Configuration default values
- Set Chirp default values
- Select Test Source Enable
- Click the 'Trigger Frame' in the "Capture and Post Processing"

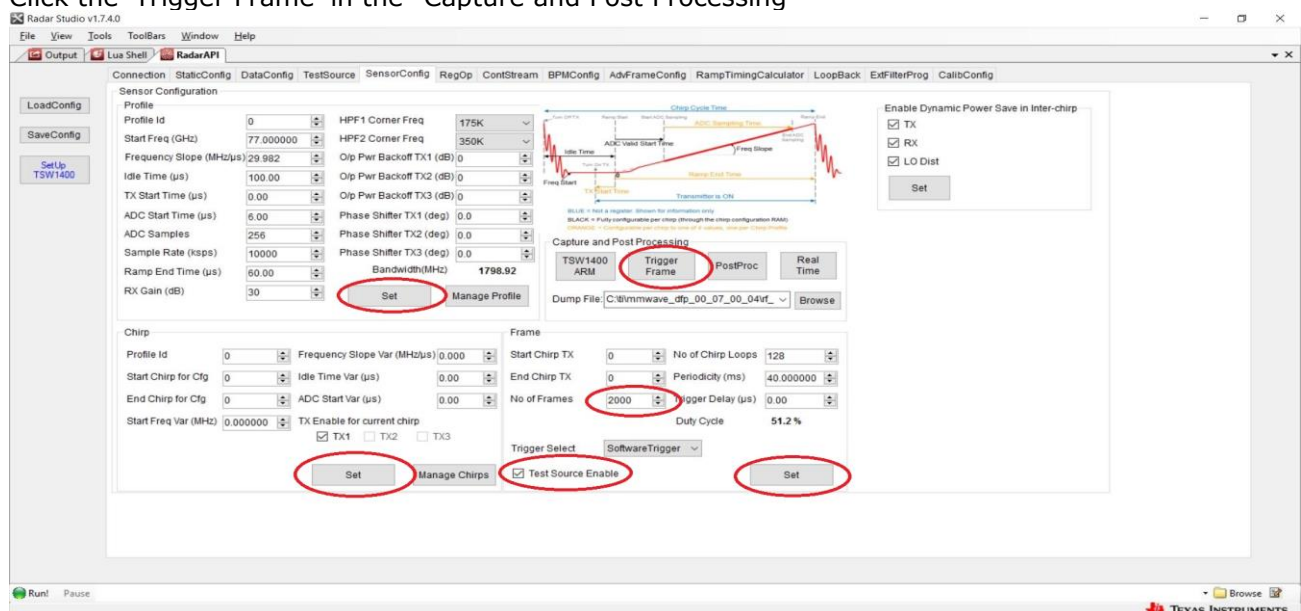


Figure 25: Radar Studio SensorConfig Settings

- Once above settings are completed, Radar EVM will generate Real Time LVDS Data transfer.

6.3 DCA1000EVM Ethernet configuration data

- The DCA1000EVM stores default Ethernet configuration data (such as DCA1000EVM IP, DCA1000EVM MAC, System IP, Configuration port and Data port) in FPGA internal registers, also there is a provision to store the same into the EEPROM using update EEPROM command. Please refer Radar studio user guide from the following link <http://www.ti.com/tool/mmwave-dfp> to get more details about 'update EEPROM' feature.
- Default Ethernet configuration data stored in FPGA internal registers are listed in Table 9,

Parameter	Default value(decimal)
FPGA IP	192.168.33.180
FPGA MAC	12-34-56-78-90-12
System IP	192.168.33.30
Configuration Port	4096
ADC/CP_ADC/ADC_CP/CP_ADC_CQ Data	4098
CP Data	4099
CQ Data	4100
R4F Debug Data	4101
DSP Debug Data	4102

Table 8 : Default value stored in FPGA registers

- DCA1000EVM has option of loading Ethernet configuration data from EEPROM or FPGA internal register based on **SW2.6**.
- Ethernet configuration data will be loaded from the EEPROM when **SW2** positioned at 6 (pin6) else FPGA register values will be loaded.

- Loading Ethernet configuration data from EEPROM
 - When **SW2** positioned at 6 (pin6) DCA1000EVM will load Ethernet configuration data from EEPROM.
 - In general, EEPROM is blank when DCA1000EVM comes out of factory. EEPROM will remain blank until update EEPROM command is given to DCA1000EVM through Ethernet port.
 - When update EEPROM command is given to DCA1000EVM, Ethernet configuration data are stored into EEPROM along with header and footer.
 - DCA1000EVM requires power cycle/hardware reset to load updated Ethernet configuration data from EEPROM.
 - During power ON/after hardware reset, DCA1000EVM FPGA loads Ethernet configuration data from EEPROM and checks for header and footer. If header and footer are found, data loaded from EEPROM will be used for Ethernet communication else FPGA uses default values stored in internal registers.
 - Ethernet configuration data updated in EEPROM should be used in Ethernet packets which are going as command to DCA1000EVM, for further communication. Otherwise DCA1000EVM will not respond to the command.
- Loading Ethernet configuration data from FPGA internal registers
 - When **SW2** positioned at (pin11) DCA1000EVM will load default Ethernet configuration data from FPGA internal registers and the same will be used for Ethernet communication.

Note: 1) For data separated mode, DCA1000EVM FPGA receives only one data port number from update EEPROM command. This port is considered as ADC data port and other data type's port number are incremented by one.

2) Whenever Switch 2.6 position is changed from one option to another option, DCA1000EVM requires power cycle /Hardware reset.

6.4 DCA1000EVM command and data format

DCA1000EVM follows UDP protocol and it supports 14 predefined commands. Configuration and status of DCA1000EVM are communicated through configuration port. Data port is used to transfer raw mode/data separated mode data.

6.4.1 DCA1000EVM command format

DCA1000EVM receives command from host through configuration port and sends back the response in same port.

- Command Request protocol consists of following:
 1. Header
 - a. Header
 - b. Command code
 - c. Data size
 2. Data
 3. Footer

Header (2 bytes)	Command code (2 bytes)	Data size (2 bytes)	Data (Min - 0 bytes Max - 504 bytes)	Footer (2 bytes)
---------------------	---------------------------	------------------------	--	---------------------

- Command Response protocol consists of following:
 1. Header
 - a. Header
 - b. Command code
 2. Status
 3. Footer

Header (2 bytes)	Command code (2 bytes)	Status (2 bytes)	Footer (2 bytes)
---------------------	---------------------------	---------------------	---------------------

- Commands supported in DCA1000EVM

Command	Command Code
RESET_FPGA_CMD_CODE	0x01
RESET_AR_DEV_CMD_CODE	0x02
CONFIG_FPGA_GEN_CMD_CODE	0x03
CONFIG_EEPROM_CMD_CODE	0x04
RECORD_START_CMD_CODE	0x05
RECORD_STOP_CMD_CODE	0x06
PLAYBACK_START_CMD_CODE	0x07
PLAYBACK_STOP_CMD_CODE	0x08
SYSTEM_CONNECT_CMD_CODE	0x09
SYSTEM_ERROR_CMD_CODE	0x0A
CONFIG_PACKET_DATA_CMD_CODE	0x0B
CONFIG_DATA_MODE_AR_DEV_CMD_CODE	0x0C
INIT_FPGA_PLAYBACK_CMD_CODE	0x0D
READ_FPGA_VERSION_CMD_CODE	0x0E

- Example: RESET_FPGA_CMD_CODE

1. Reset FPGA Command request from host

Name	Data Type	Number of bytes	Default Value	Min value	Max value	Description
Header	UINT16	2	0xA55A	-	-	0xA55A always. Start bits of packet
Command code	UINT16	2	0x01	-	-	Command code
Size	UINT16	2	0	-	-	Data Size
Footer	UINT16	2	0xEEAA	-	-	0xEEAA always. Stop bits of packet

2. Command response from DCA1000EVM

Name	Data Type	Number of bytes	Default Value	Min value	Max value	Description
Header	UINT16	2	0xA55A	-	-	0xA55A always. Start bits of packet
Command code	UINT16	2	0x01	-	-	Command code
Status	UINT16	2	0	0	1	0 – Success 1 – Failure
Footer	UINT16	2	0xEEAA	-	-	0xEEAA always. Stop bits of packet

6.4.2 DCA1000EVM data format

DCA1000EVM sends raw mode/data separated mode data to host through data port.

- **Raw mode data format:** DCA1000EVM sends raw mode data in the following format.

Sequence number (4 bytes)	Byte count (6 bytes)	Raw mode data (Min - 48 bytes Max - 1462 bytes)
------------------------------	-------------------------	---

- **Data separated mode data format:** DCA1000EVM sends data separated mode data in the following format.

Sequence number (4 bytes)	Byte Count (6 bytes)	Data separated mode identifier (8 bytes)	Data separated mode length field (8 bytes)	Data separated mode data (Min - 48 bytes Max - 1446 bytes)
------------------------------	-------------------------	---	---	--

6.5 DCA1000EVM GUI Procedure

The Release folder shall contain the Test Package. The Test package release folder shall contain the DCA1000EVM GUI in the below folder path

/DCA1000EVM _TP_REL/SW/GUI/DCA1000EVM .EXE

To Open DCA1000EVM GUI, double-click on the GUI application (DCA1000EVM .EXE)

Set Static IP Address to Host PC as 192.168.33.30 as default.

➤ SYSTEM Window

1. This window shall configure System IP Address, FPGA IP Address, FPGA MAC Address, Configuration port, Record/Playback port.
2. User can configure default settings in SYSTEM window.
3. Click 'Connect' option to connect DCA1000EVM with Host PC DCA1000EVM GUI.

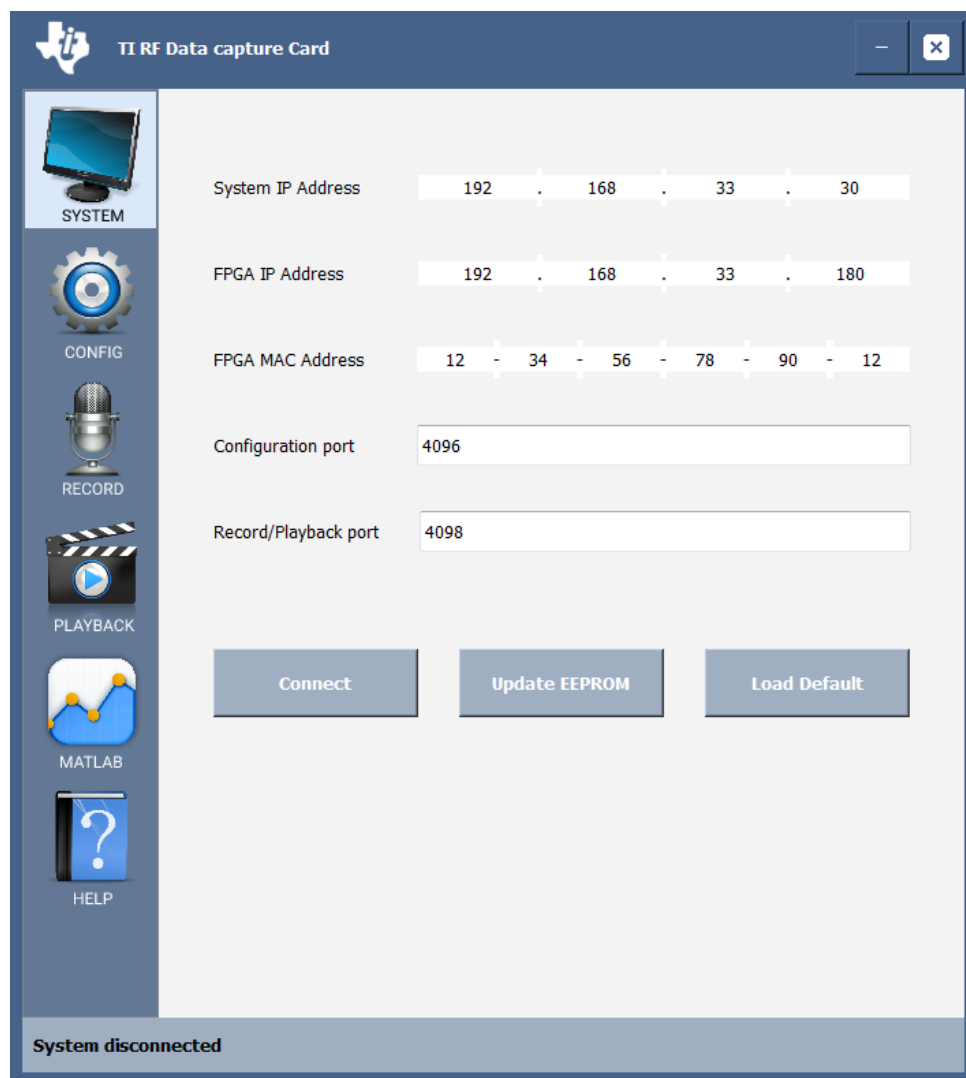


Figure 26: DCA1000EVM GUI - SYSTEM WINDOW

➤ CONFIG Window

1. User can select following different modes of operations as provided in the below figure (When SW2.5 selected as SW_Config option)
 - a) Data Logging Mode : RAW/Multi
 - b) LVDS Mode : AR1243/AR1642
 - c) Data Transfer Mode : LVDS Capture/DMM Playback
 - d) Data Capture Mode : SD Card Storage/Ethernet Stream
 - e) Data Format Mode : 12/14/16 bit
2. Click 'Configure FPGA' to configure the modes of operation
3. User can reset the FPGA on DCA1000EVM by clicking on 'Reset FPGA'
4. User can reset the Radar EVM by clicking on 'Reset AR Device'
5. User can configure Ethernet packet delay in DCA1000EVM GUI by changing the values from 5us to 500us for file capture depends on Host PC configuration and capabilities.

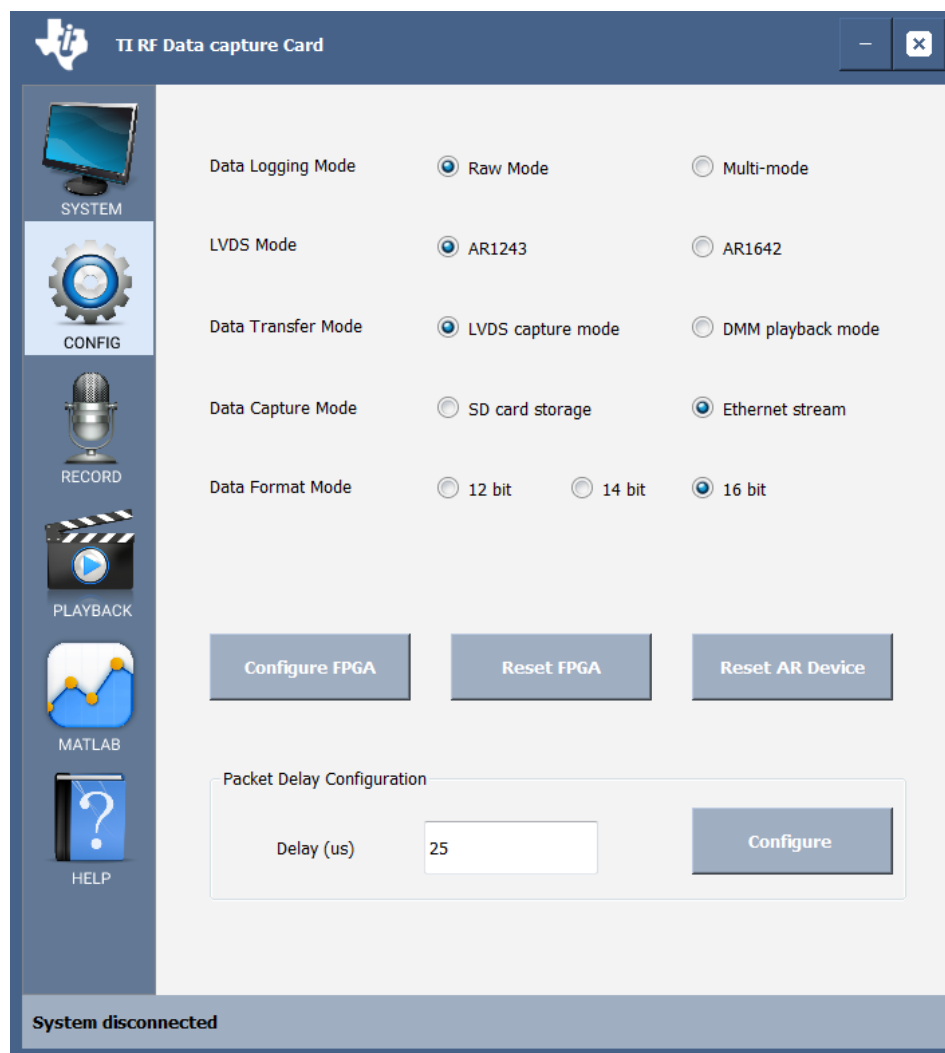


Figure 27: DCA1000EVM GUI - CONFIG WINDOW

➤ **RECORD Window**

1. User can select the file name along with folder name in this window.
2. Click 'Record' button to start the data capture into a file.
3. Click 'Stop' button to stop the data capture.

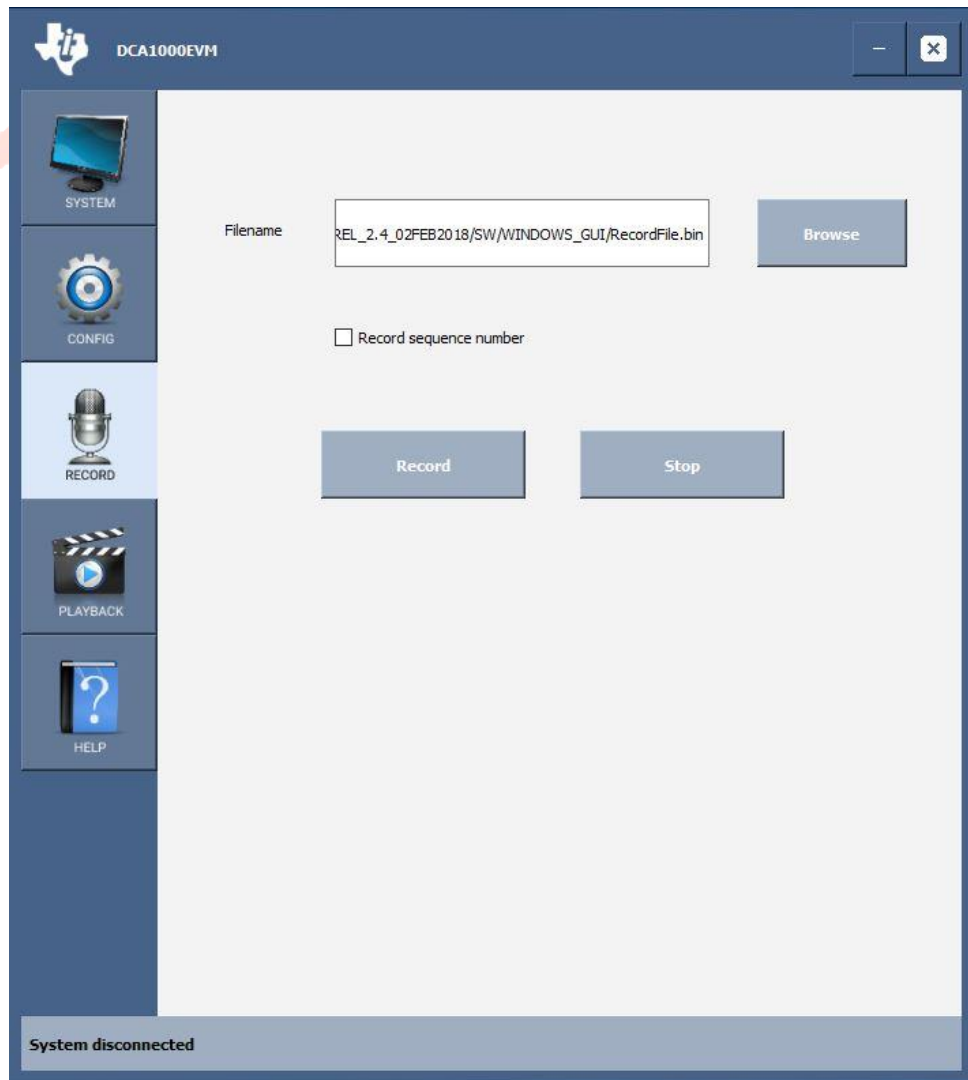


Figure 28: DCA1000EVM GUI - RECORD WINDOW

➤ **PLAYBACK Window**

1. User can select the file name along with folder name in this window.
2. Click 'Playback' button to start the data transfer and playback.
3. Click 'Stop' button to stop the data transfer and playback.
4. User can configure Ethernet packet delay in DCA1000EVM GUI by changing the values from 0us to 500us for playback depends on Host PC configuration and capabilities.

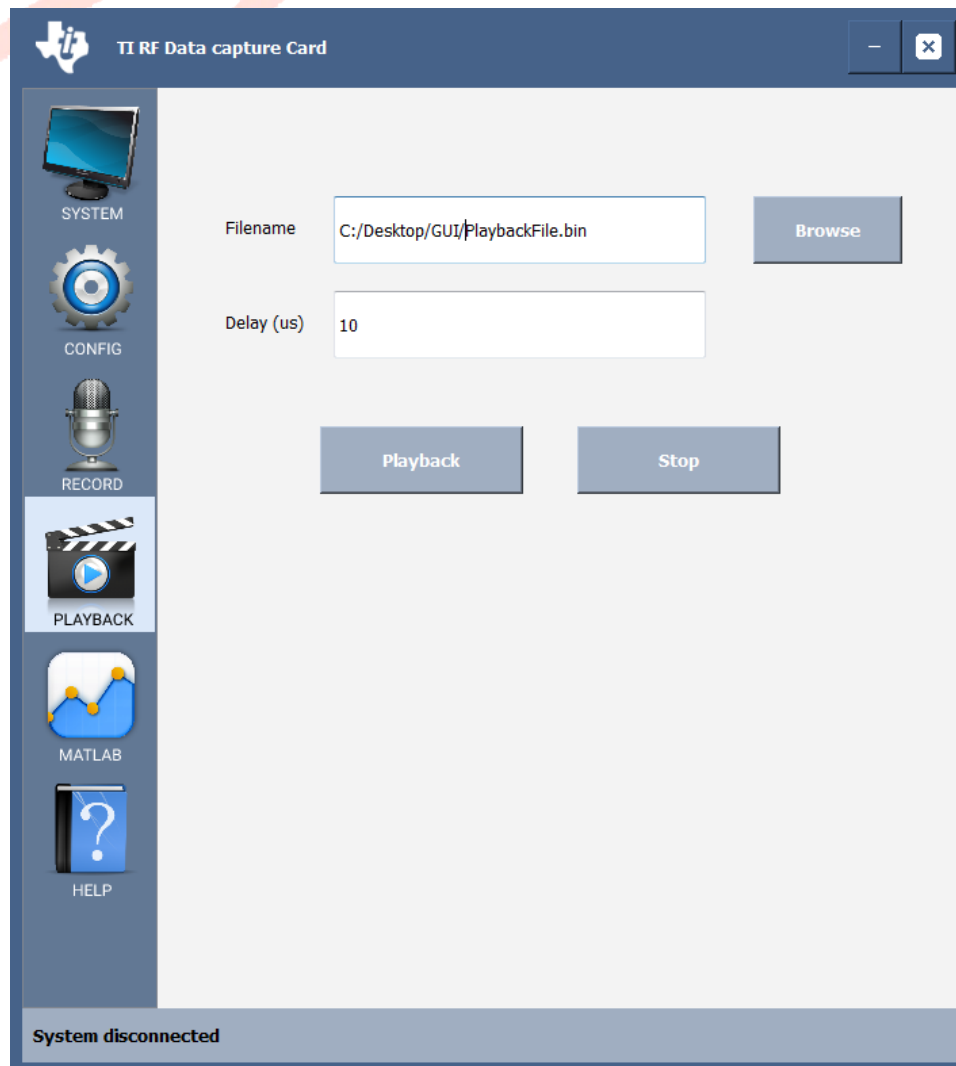


Figure 29: DCA1000EVM GUI - PLAYBACK WINDOW

➤ Help Menu

- Help menu gives GUI and FPGA version details.
- GUI version is displayed once the GUI is opened.
- FPGA version will be updated once the Capture card gets connected with GUI.
- FPGA version gives version details and the feature supported in the binary either record or playback.

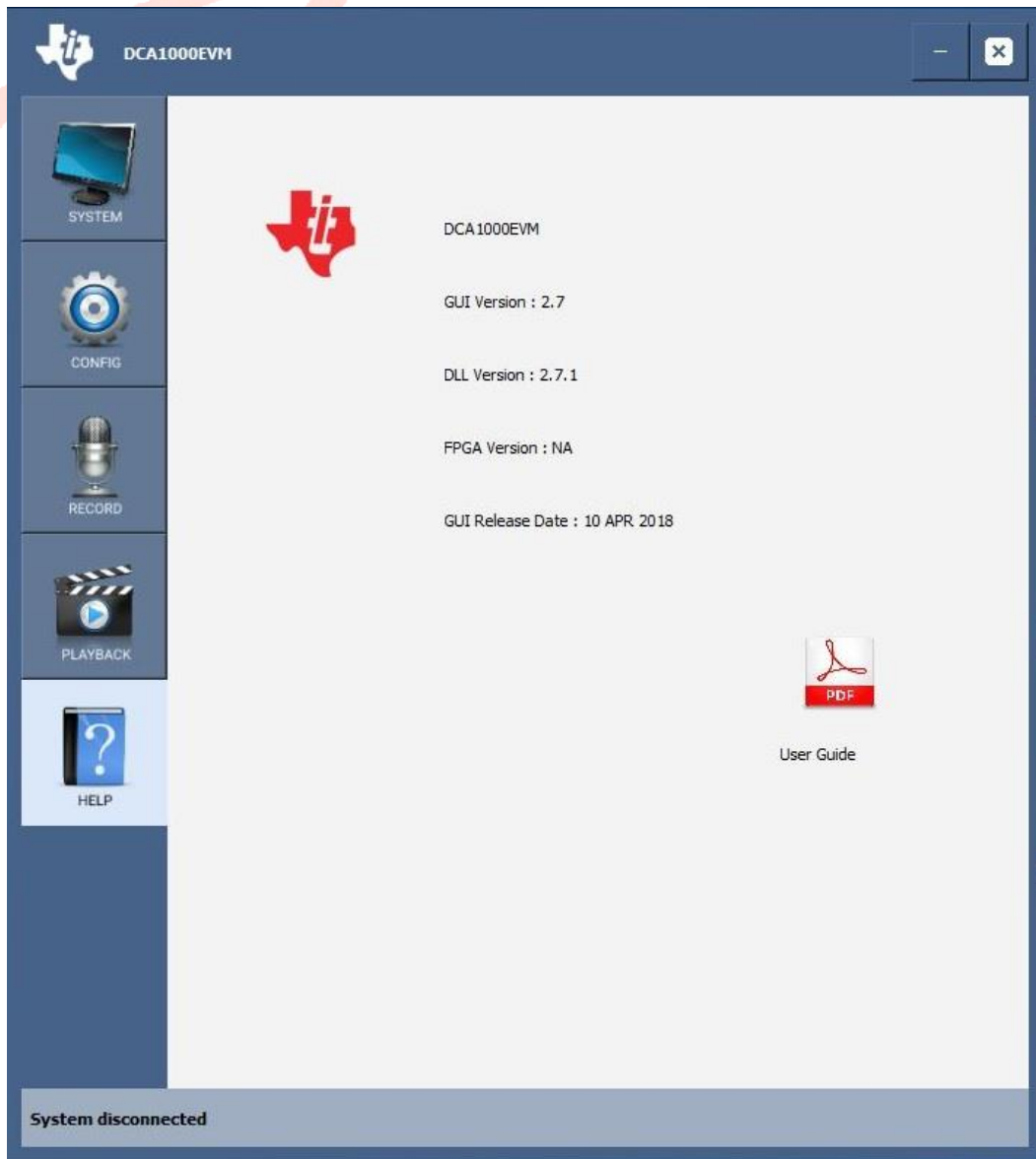


Figure 30: DCA1000EVM GUI - HELP MENU

6.6 Use case 1: Record - RAW Mode/Ethernet Stream

RAW Mode recording can be tested either through Hardware switch configuration mode or Software GUI configuration mode.

User can configure Ethernet packet delay in DCA1000EVM GUI by changing the values from 5us to 500us for file capture depends on Host PC configuration and capabilities.

Note 1: If LVDS data rate is changed in Radar Studio, DCA1000EVM requires Hardware/Software reset to lock with incoming clock.

Note 2: Number of ADC samples configured for Radar EVM should be multiple of 256 bytes else residue data will be left inside DCA1000EVM LVDS buffer. This is due to DDR3 controller implementation in DCA1000EVM.

6.6.1 Hardware switch configuration Mode

Test Procedure

- Configure the Radar EVM with appropriate settings through Radar studio as mentioned in section 5.2.
- **DCA1000EVM switch settings:**
 1. Keep switch SW2.5 in HW_CONFIG position.
 2. Keep switch SW2.4 in RAW_MODE.
 3. Keep switch SW2.1 in LVDS_CAPTURE position.
 4. Keep switch SW2.2 in ETH_STREAM position.
 5. Keep switch SW2.3 in AR1243_MODE for 4 Lane LVDS interface from **xWR1243BOOST** or in AR1642_MODE (Other side) for 2 Lane LVDS interface from **xWR1642BOOST**.
 6. Set SW1.1 to 12BIT_ON (or) SW1.2 to 14BIT_ON (or) SW1.3 to 16BIT_ON according to the data size selected in RADAR EVM.

Note: Only one switch option should be in ON position at any time in SW1

- **DCA1000EVM GUI settings:**
 - Run DCA1000EVM GUI by double-clicking on "DCA1000EVM_GUI.exe"
 - Open "System" tab and click on "Connect" button
 - Open "Record" tab and browse to a valid .bin file
 - Click on "Record" button to start recording of data
 - Click on "Stop" button to stop recording of data
- **Indications:**
 - When data transfer starts **DATA_TRANS_PRG** LED (LD1) will start toggling and 'Record in progress' message will be shown in GUI.
 - When record is completed **DATA_TRANS_PRG** LED (LD1) will glow and 'Record is Completed' message will be shown in GUI.
 - If xWR1xxx EVM is not sending LVDS data then **LVDS_PATH_ERR** LED (LD7) will glow and 'No LVDS Data' message will be shown in GUI
 - When DDR gets full **DDR_FULL** LED (LD6) will glow and 'DDR is Full' message will be shown in GUI.
 - Whenever FPGA internal buffer gets full **FPGA_ERR** LED (LD5) will glow and 'LVDS Buffer Full' message will be shown in GUI.

- **Output File Name :**

- User can create any file name to capture the data. GUI will create a file in the format of '**XX_Raw_Y.bin**' in the stored location.
(**XX** --> User configurable filename. **Y** --> Incremental number starts with 0)

- **Post Processing**

1. Packet Re-ordering & Zero Filling:

Step 1:

Please make sure that Packet_Reorder_Zerofill.exe should be present in the released GUI folder & also where the files are to be post processed

Step 2:

Navigate to above folder and run the exe. To run the exe, type the following command in the console

> Packet_Reorder_Zerofill.exe <Input_File_Name> <Output_FileP_Name> <log.txt> and press ENTER

For eg: > Packet_Reorder_Zerofill.exe RecordFile.bin Record.bin log.txt

Step 3:

- a. Once post processed, < Output file name > <log.txt> will be generated.

- b. Log.txt file contains the post processed information as follows,

For eg:

- i. Reordering required: YES/NO
- ii. Packet loss: YES/NO
- iii. Number of packets received: 100000
- iv. Number of zero filled packets: 0
- v. Number of zero filled bytes: 0

- c. Then use RawFileFormat.exe for formatting the file.

2. Raw File Formatting:

Run RawFileFormat.exe (It will be available in the test release GUI package folder) for formatting the data.

Note: To run the script, run the following command in the console

> RawFileFormat.exe <No of Lanes> <Input_File_Name> <Output_File_Name>

For eg:

- 1) For formatting 4 lanes data,
> RawFileFormat.exe 4 "RawFile_0.bin" "Raw0.bin"
- 2) For formatting 2 lanes data,
> RawFileFormat.exe 2 "RawFile_0.bin" "Raw0.bin"

- **Analysis**

- Check the formatted file size and compare with expected file size in bytes.
- Open Matlab, load log file from Radar studio postproc folder and formatted file.
- Check 2D FFT plot in matlab to validate the data loss or data corruption.

6.6.2 Software GUI Configuration Mode

Test Procedure

- Configure the Radar EVM with appropriate settings through Radar studio as mentioned in section 5.2.
- **DCA1000EVM switch settings:**
 - Keep switch SW2.5 in SW_CONFIG position.
- **DCA1000EVM GUI settings:**
 1. Open DCA1000EVM GUI
 2. Open "System" tab and click on "Connect" button
 3. Open "Config" tab, Click on "Configure FPGA" button after configuring the modes with following options

a) Data Logging Mode	--> Raw Mode
b) LVDS Mode	--> AR1243(4 Lane)/AR1642(2 Lane)
c) Data Transfer Mode	--> LVDS Capture Mode
d) Data Capture Mode	--> Ethernet Stream
e) Data Format Mode	--> 12/14/16 bit
 4. Open "Record" tab and browse to a valid .bin file
 5. Click on "Record" button to start data transfer and recording
 6. Click on "Stop" button to stop data transfer and recording
- **Indications:**
 - When data transfer starts **DATA_TRANS_PRG** LED (LD1) will start toggling and '**Record in progress**' message will be shown in GUI.
 - When record is completed **DATA_TRANS_PRG** LED (LD1) will glow and '**Record is Completed**' message will be shown in GUI.
 - If xWR1xxx EVM is not sending LVDS data then **LVDS_PATH_ERR** LED (LD7) will glow and '**No LVDS Data**' message will be shown in GUI
 - When DDR gets full **DDR_FULL** LED (LD6) will glow and '**DDR is Full**' message will be shown in GUI.
 - Whenever FPGA internal buffer gets full **FPGA_ERR** LED (LD5) will glow and '**LVDS Buffer Full**' message will be shown in GUI.
- **Output File Name:**
 - User can create any file name to capture the data.GUI will create a file in the format of '**XX_Raw_Y.bin**' in the stored location.
(XX --> User configurable filename.Y --> Incremental number starts with 0)
- **Post Processing**
 1. Packet Re-ordering & Zero Filling:

Step 1:
Please make sure that Packet_Reorder_Zerofill.exe should be present in the released GUI folder & also where the files are to be post processed

Step 2:
Navigate to above folder and run the exe. To run the exe, type the following command in the console
> Packet_Reorder_Zerofill.exe <Input_File_Name> <Output_FileP_Name> <log.txt> and press ENTER

For eg: > Packet_Reorder_Zerofill.exe RecordFile.bin Record.bin log.txt

Step 3:

a. Once post processed, < Output file name > <log.txt> will be generated.

b. Log.txt file contains the post processed information as follows,

For eg:

- i. Reordering required: YES/NO
- ii. Packet loss: YES/NO
- iii. Number of packets received: 100000
- iv. Number of zero filled packets: 0
- v. Number of zero filled bytes: 0

c. Then use RawFileFormat.exe for formatting the file.

2. Raw File Formatting:

Run RawFileFormat.exe (It will be available in the test release GUI package folder) for formatting the data.

Note: To run the script, run the following command in the console

> RawFileFormat.exe <No of Lanes> <Input_File_Name> <Output_File_Name>

For eg:

- 1) For formatting 4 lanes data,
> RawFileFormat.exe 4 "RawFile_0.bin" "Raw0.bin"
- 2) For formatting 2 lanes data,
> RawFileFormat.exe 2 "RawFile_0.bin" "Raw0.bin"

● **Analysis**

- Check the formatted file size and compare with expected file size in bytes.
- Open Matlab, load log file from Radar studio postproc folder and formatted file.
- Check 2D FFT plot in matlab to validate the data loss or data corruption.

6.7 Use case 2: Record - Multi Mode/Ethernet Stream

Multi-Mode recording can be tested either through Hardware switch configuration mode or Software GUI Configuration Mode.

User can configure Ethernet packet delay in DCA1000EVM GUI by changing the values from 5us to 500us for file capture depends on Host PC configuration and capabilities.

Note 1: DCA1000EVM requires hardware/software reset after power cycle or LVDS data rate change.

Note 2: Number of ADC samples configured for Radar EVM should be multiple of 256 bytes else residue data will be left inside DCA1000EVM LVDS buffer. This is due to DDR3 controller implementation in DCA1000EVM.

6.7.1 Hardware switch configuration Mode

Test Procedure

- Set Radar EVM jumper settings (SOP0 and SOP2) for Flash programming mode and download Radar EVM binary using Uniflash.
- Set Radar EVM jumper setting (SOP0) to Functional mode to test Multi mode feature.
- **DCA1000EVM switch settings:**
 1. Keep switch SW2.5 in HW_CONFIG position.
 2. Keep switch SW2.4 in DATA_MODE.
 3. Keep switch SW2.1 in LVDS_CAPTURE position.
 4. Keep switch SW2.2 in ETH_STREAM position.
 5. Keep switch SW2.3 in AR1243_MODE for 4 Lane LVDS interface from **xWR1243BOOST** or in AR1642_MODE (Other side) for 2 Lane LVDS interface from **xWR1642BOOST**.
 6. Set SW1.1 to 12BIT_ON (or) SW1.2 to 14BIT_ON (or) SW1.3 to 16BIT_ON according to the data size selected in RADAR EVM.

Note: Only one switch option should be in ON position at any time in SW1

- **DCA1000EVM GUI settings:**
 - Open DCA1000EVM GUI
 - Open "System" tab and click on "Connect" button
 - Open "config" tab
 - Click "Reset AR Device" to reset Radar EVM
 - Open "Record" tab and browse to a valid .bin file
 - Click on "Record" button to start recording
 - Run Radar EVM script in console to send data for multi mode
 - Click on "Stop" button once record completed status is received

- **Indications:**

- When data transfer starts **DATA_TRANS_PRG** LED (LD1) will start toggling and '**Record in progress**' message will be shown in GUI.
- When record is completed **DATA_TRANS_PRG** LED (LD1) will glow and '**Record is Completed**' message will be shown in GUI.
- If xWR1xxx EVM is not sending LVDS data then **LVDS_PATH_ERR** LED (LD7) will glow and '**No LVDS Data**' message will be shown in GUI
- If there is unknown header detecting by FPGA logic then GUI will indicate '**NO Header**' message as well as **HDR_ERR** LED (LD8) will glow.
- When DDR gets full **DDR_FULL** LED (LD6) will glow and '**DDR is Full**' message will be shown in GUI.
- Whenever FPGA internal buffer gets full **FPGA_ERR** LED (LD5) will glow and '**LVDS Buffer Full**' message will be shown in GUI.

- **Output File Name:**

- User can create any file name to capture the data. GUI will create a file in 5 different data types as mentioned below
 - **XX_CP_Y.bin**
 - **XX_CQ_Y.bin**
 - **XX_DSP_Y.bin**
 - **XX_R4F_Y.bin**
 - **XX_Raw_Y.bin**
 (XX --> User configurable filename. Y --> Incremental number starts with 0)

- **Post Processing**

- Check captured file size with transferred bytes.
- Open captured file and check the data.

6.7.2 Software GUI Configuration Mode

Test Procedure

- Set Radar EVM jumper settings (SOP0 and SOP2) for Flash programming mode and download Radar EVM binary using Uniflash.
- Set Radar EVM jumper setting (SOP0) to Functional mode to test Multi mode feature.
- **DCA1000EVM switch settings:**
 - Keep switch SW2.5 in SW_CONFIG position.
- **DCA1000EVM GUI settings:**
 1. Open DCA1000EVM GUI
 2. Open "System" tab and click on "Connect" button
 3. Open "config" tab
 4. Click "Reset AR Device" to reset Radar EVM
 5. In "Config" tab, Click on "Configure FPGA" button after configuring the modes with following options. This step is required only after power ON or DCA1000EVM reset.
 - a) Data Logging Mode --> Multi Mode
 - b) LVDS Mode --> AR1243(4 Lane)/AR1642(2 Lane)
 - c) Data Transfer Mode --> LVDS Capture Mode
 - d) Data Capture Mode --> Ethernet Stream
 - e) Data Format Mode --> 12/14/16 bit
 6. Open "Record" tab and browse to a valid .bin file
 7. Click on "Record" button to start data transfer and recording
 8. Run Radar EVM script in console to send data for multi-mode
 9. Click on "Stop" button to stop data transfer and recording

Indications:

- When data transfer starts **DATA_TRANS_PRG** LED (LD1) will start toggling and '**Record in progress**' message will be shown in GUI.
- When record is completed **DATA_TRANS_PRG** LED (LD1) will glow and '**Record is Completed**' message will be shown in GUI.
- If xWR1xxx EVM is not sending LVDS data, then **LVDS_PATH_ERR** LED (LD7) will glow and '**No LVDS Data**' message will be shown in GUI
- If there is unknown header detecting by FPGA logic, then GUI will indicate '**NO Header**' message as well as **HDR_ERR** LED (LD8) will glow.
- When DDR gets full **DDR_FULL** LED (LD6) will glow and '**DDR is Full**' message will be shown in GUI.
- Whenever FPGA internal buffer gets full **FPGA_ERR** LED (LD5) will glow and '**LVDS Buffer Full**' message will be shown in GUI.

● Output File Name :

- User can create any file name to capture the data. GUI will create a file in 5 different data types as mentioned below
 - **XX_CP_Y.bin**
 - **XX_CQ_Y.bin**
 - **XX_DSP_Y.bin**
 - **XX_R4F_Y.bin**
 - **XX_Raw_Y.bin**
- (XX --> User configurable filename. Y --> Incremental number starts with 0)

● Post Processing

- Check captured file size with transferred bytes.
- Open captured file and check the data.

6.8 Use case 3: Record - Raw Mode/SD Card Storage

RAW Mode recording can be tested either through Hardware switch configuration mode or Software GUI Configuration Mode.

User can configure Ethernet packet delay in DCA1000EVM GUI by changing the values from 5us to 500us for file capture depends on Host PC configuration and capabilities.

Note 1: If LVDS data rate is changed in Radar Studio, DCA1000EVM requires Hardware/Software reset to lock with incoming clock.

Note 2: Number of ADC samples configured for Radar EVM should be multiple of 256 bytes else residue data will be left inside DCA1000EVM LVDS buffer. This is due to DDR3 controller implementation in DCA1000EVM.

6.8.1 Hardware switch configuration Mode

Test Procedure

- Configure the Radar EVM with appropriate settings through Radar studio as mentioned in section 5.2.
- **DCA1000EVM switch settings:**
 1. Keep switch SW2.5 in HW_CONFIG position.
 2. Keep switch SW2.4 in RAW_MODE
 3. Keep switch SW2.1 in LVDS_CAPTURE position.
 4. Keep switch SW2.2 in SD_STORE position.
 5. Keep switch SW2.3 in AR1243_MODE for 4 Lane LVDS interface from **xWR1243BOOST** or in AR1642_MODE (Other side) for 2 Lane LVDS interface from **xWR1642BOOST**.
 6. Set SW1.1 to 12BIT_ON (or) SW1.2 to 14BIT_ON (or) SW1.3 to 16BIT_ON according to the data size.

Note: Only one switch option should be in ON position at any time in SW1

- **DCA1000EVM GUI settings:**
 - Open DCA1000EVM GUI
 - Open "System" tab and click on "Connect" button
 - Open "Record" tab and browse to a valid .bin file
 - Click on "Record" button to start data capture and store it in SD card
 - Click on "Stop" button to stop data capture.
- **Post Processing**
 - TBD
- **Analysis**
 - TBD

6.8.2 Software GUI Configuration Mode

Test Procedure

- Configure Radar EVM with appropriate settings through Radar studio as mentioned in section 5.2.
- **DCA1000EVM switch settings:**
 - Keep switch SW2.5 in SW_CONFIG position.
- **DCA1000EVM GUI settings:**
 1. Open DCA1000EVM GUI
 2. Open "System" tab and click on "Connect" button
 3. Open "Config" tab, Click on "Configure FPGA" button after configuring the modes with following options
 - a) Data Logging Mode --> Multi Mode
 - b) LVDS Mode --> AR1243(4 Lane)/AR1642(2 Lane)
 - c) Data Transfer Mode --> LVDS Capture Mode
 - d) Data Capture Mode --> SD Card Storage
 - e) Data Format Mode --> 12/14/16 bit
 4. Open "Record" tab and browse to a valid .bin file
 5. Click on "Record" button to start data capture and store it in SD card
 6. Click on "Stop" button to stop data capture
- **Post Processing**
 - TBD
- **Analysis**
 - TBD

6.9 Use case 4: Record - Multi Mode/SD Card Storage

Multi-Mode recording can be tested either through Hardware switch configuration mode or Software GUI Configuration Mode.

User can configure Ethernet packet delay in DCA1000EVM GUI by changing the values from 5us to 500us for file capture depends on Host PC configuration and capabilities.

Note 1: Once the board is powered on, DCA1000EVM requires Hardware/Software reset to lock with incoming clock.

Note 2: Number of ADC samples configured for Radar EVM should be multiple of 256 bytes else residue data will be left inside DCA1000EVM LVDS buffer. This is due to DDR3 controller implementation in DCA1000EVM.

6.9.1 Hardware switch configuration Mode

Test Procedure

- Set Radar EVM jumper settings (SOP0 and SOP2) for Flash programming mode and download Radar EVM binary using Uniflash.
- Set Radar EVM jumper setting (SOP0) to Functional mode to test Multi mode feature.
- **DCA1000EVM switch settings:**
 1. Keep switch SW2.5 in HW_CONFIG position.
 2. Keep switch SW2.4 in DATA_MODE
 3. Keep switch SW2.1 in LVDS_CAPTURE position.
 4. Keep switch SW2.2 in SD_STORE position.
 5. Keep switch SW2.3 in AR1243_MODE for 4 Lane LVDS interface from **xWR1243BOOST** or in AR1642_MODE (Other side) for 2 Lane LVDS interface from **xWR1642BOOST**.
 6. Set SW1.1 to 12BIT_ON or SW1.2 to 14BIT_ON or SW1.3 to 16BIT_ON according to the data size.

Note: Only one switch option should be in ON position at any time in SW1

- **DCA1000EVM GUI settings:**
 - Open DCA1000EVM GUI
 - Open "System" tab and click on "Connect" button
 - Open "Record" tab and browse to a valid .bin file
 - Click on "Record" button to start data transfer and recording
 - Click on "Stop" button to stop data transfer and recording
- **Post Processing**
 - TBD
- **Analysis**
 - TBD

6.9.2 Software GUI Configuration Mode

Test Procedure

- Set Radar EVM jumper settings (SOP0 and SOP2) for Flash programming mode and download Radar EVM binary using Uniflash.
- Set Radar EVM jumper setting (SOP0) to Functional mode to test Multi mode feature.
- **DCA1000EVM switch settings:**
 - Keep switch SW2.5 in SW_CONFIG position.
- **DCA1000EVM GUI settings:**
 1. Open DCA1000EVM GUI
 2. Open "System" tab and click on "Connect" button
 3. Open "Config" tab, click on "Configure FPGA" button after configuring the modes with following options
 - a) Data Logging Mode --> Multi Mode
 - b) LVDS Mode --> AR1243(4 Lane)/AR1642(2 Lane)
 - c) Data Transfer Mode --> LVDS Capture Mode
 - d) Data Capture Mode --> SD Card Storage
 - e) Data Format Mode --> 12/14/16 bit
 4. Open "Record" tab and browse to a valid.bin file
 5. Click on "Record" button to start data capture and store it in SD card
 6. Click on "Stop" button to stop data capture
- **Post Processing**
 - TBD
- **Analysis**
 - TBD

6.10 Use case 5: Playback

Playback can be tested either through Hardware switch configuration mode or Software GUI Configuration Mode.

User can configure Ethernet packet delay in DCA1000EVM GUI by changing the values from 0us to 500us for playback depends on Host PC configuration and capabilities.

6.10.1 Hardware switch configuration Mode

Test Procedure

- **DCA1000EVM switch settings:**
 1. Keep switch SW2.5 in HW_CONFIG position.
 2. Keep switch SW2.1 in DMM_PLAYBACK position.
- **DCA1000EVM GUI settings:**
 - Open DCA1000EVM GUI
 - Open "System" tab and click on "Connect" button
 - Open "Playback" tab and browse to a valid .bin (refer below note) file
 - Click on "Playback" button to start data transfer and playback
 - Click on "Stop" button to stop data transfer and playback

Note: Select a valid .bin file with incremental pattern. Playback has been tested with 16-bit incremental data for DMM data lines and all other control lines are driven high from GUI.

Indications:

- When data transfer starts **DATA_TRANS_PRG** LED (LD1) will start toggling and '**Playback in progress**' message will be shown in GUI.
 - When playback is completed **DATA_TRANS_PRG LED** (LD1) will glow and '**Playback is Completed**' message will be shown in GUI.
 - If the packet received from DCA1000EVM GUI is not in sequence, then **FPGA_ERR** LED (LD5) will glow and '**Packet is in out of sequence**' message will be shown in GUI
- **Analysis**
 - Check the data transmitted from DCA1000EVM GUI to Radar EVM in Lattice reveal analyzer. Incremental data should be seen in reveal analyzer without any breaks. Visual observation.
 - Probe DMM data lines using oscilloscope. Since DMM interface is running at 100MHz clock, divided version of 100MHz clock should be observed in all data lines. For example: 0th bit should be 50MHz, 1st bit should be 25MHz, 2nd bit should be 12.5MHz and so on. Probe on CRO and confirm the results.

6.10.2 Software GUI Configuration Mode

Test Procedure

- **DCA1000EVM switch settings:**
 1. Keep switch SW2.5 in SW_CONFIG position.
 2. Keep switch SW2.1 in DMM_PLAYBACK position.
- **DCA1000EVM GUI settings:**
 - Open DCA1000EVM GUI
 - Open "System" tab and click on "Connect" button
 - Open "Playback" tab and browse to a valid .bin file
 - Click on "Playback" button to start data transfer and playback
 - Click on "Stop" button to stop data transfer and playback

Indications:

- When data transfer starts **DATA_TRANS_PRG** LED (LD1) will start toggling and '**Record in progress**' message will be shown in GUI.
 - When playback is completed **DATA_TRANS_PRG LED** (LD1) will glow and '**Playback is completed**' message will be shown in GUI.
 - If the packet received from DCA1000EVM GUI is not in sequence, then **FPGA_ERR** LED (LD5) will glow and '**Packet is in out of sequence**' message will be shown in GUI
- **Analysis**
 - Check the data transmitted from DCA1000EVM GUI to Radar EVM in Lattice reveal analyzer. Incremental data should be seen in reveal analyzer without any breaks. Visual observation.
 - Probe DMM data lines using oscilloscope. Since DMM interface is running at 100MHz clock, divided version of 100MHz clock should be observed in all data lines. For example: 0th bit should be 50MHz, 1st bit should be 25MHz, 2nd bit should be 12.5MHz and so on. Probe on CRO and confirm the results.

7 Errors

Error indication	Description	Troubleshooting
Out of sequence packet	Host PC might miss some Ethernet packets/might receive jumbled packets from DCA1000EVM. This may be due to either Ethernet fast streaming, UDP protocol, Host PC configuration and Network Card (NIC).	User can change Ethernet packet delay options to avoid out of sequence. When increasing Ethernet packet delay, Streaming Throughput will be decreased due to slower data capturing.
FPGA_ERR_LED	This LED indicates FPGA internal LVDS buffer overflow error. This error occurs when LVDS data rate is faster than Ethernet data rate	User can change Periodicity in Radar studio Sensorconfig settings to avoid LVDS buffer overflow condition
DDR_FULL_LED	This LED indicates DDR3 full error. This condition occurs when LVDS data rate is faster than Ethernet data rate	User can change Periodicity in Radar studio Sensorconfig settings to avoid DDR3 full condition
LVDS_PATH_ERR_LED	This LED indicates NO LVDS DATA received from Radar EVM within timeout period.	<ul style="list-style-type: none"> This will occur if xWR1xxx EVM is not sending LVDS data to DCA1000EVM. Check whether Radar EVM is configured and enabled to send data Check cabling assembly is proper between xWR1xxx EVM & DCA1000EVM
HEADER_ERR_LED	This LED glows when NO HEADER found in LVDS data.	<ul style="list-style-type: none"> This will occur if xWR1xxx EVM is not sending LVDS data to DCA1000EVM. Check whether Radar EVM is configured and enabled to send data separated mode data Check cabling assembly is proper between xWR1xxx EVM & DCA1000EVM
EEPROM_RD_FAIL_LED	This LED indicates EEPROM read failure status. This error can happen if any problem exists with EEPROM hardware connectivity or EEPROM addressing problem.	<ul style="list-style-type: none"> Power cycle DCA1000EVM Check EEPROM connectivity and address lines on hardware

8 Troubleshooting

- To confirm Ethernet connectivity between the DCA1000EVM and Host PC, Use Wireshark tool to check the Ethernet packet transfer.
- Ensure that UDP is enabled in the Host PC for UDP Data Transmission and Reception.
- When providing Hardware reset or Power cycle the on board, before you start connecting the Radar Studio with the DCA1000EVM, check whether the Host PC Ethernet connectivity established or not.
- Ensure that the Samtec cable is properly connected between xWR1xxx EVM and DCA1000EVM to avoid signal discontinuity.
- The default Ethernet delay is pre-programmed as 25µs in both FPGA and software. The user can configure the delay from 5µs to 500µs depending on Host PC configuration and capabilities, to avoid packet loss/packets out of sequence.
- When the LVDS data rate is changed in Radar Studio, the DCA1000EVM requires a Hardware/Software reset to lock the PLL with the incoming clock in the FPGA.

9 Limitations

- This release supports LVDS to Ethernet Streaming for raw mode & data segregated mode data Capture.
- The SD Card Storage and Playback feature is not supported in this release.
- DCA1000EVM FPGA needs minimum delay of 12ms required between Bit clock starts and actual LVDS Data start to lock the LVDS PLL IP.
- In default conditions, Ethernet throughput varies up to **325 Mbps** speed in a 25-µs Ethernet packet delay. The User can change the Ethernet packet delay from 5µs to 500µs to achieve different throughputs, as explained in [Table 10](#), and by changing the Periodicity in the Radar studio Sensorconfig settings (refer the following link for more details, <http://www.ti.com/tool/mmwave-dfp>) to achieve different Ethernet throughputs.

Ethernet delay	Packet	Theoretical throughput (Mbps)
5us		~706
10us		~545
25us		~325
50us		~193

Table 9 Theoretical Throughput



USA	INDIA
43092 CHRISTY STREET, Fremont, CA 94538, USA Tel: +1-408-705-2240 Fax: +1-408-987-9665 E-mail: usa@mistralsolutions.com	No.60, 'Adarsh Regent', 100 Ft. Ring Road, Domlur Extension, Bangalore - 560 071. Tel: +91-80-3091 2600 Fax: +91-80-2535 6440 E-mail: info@mistralsolutions.com