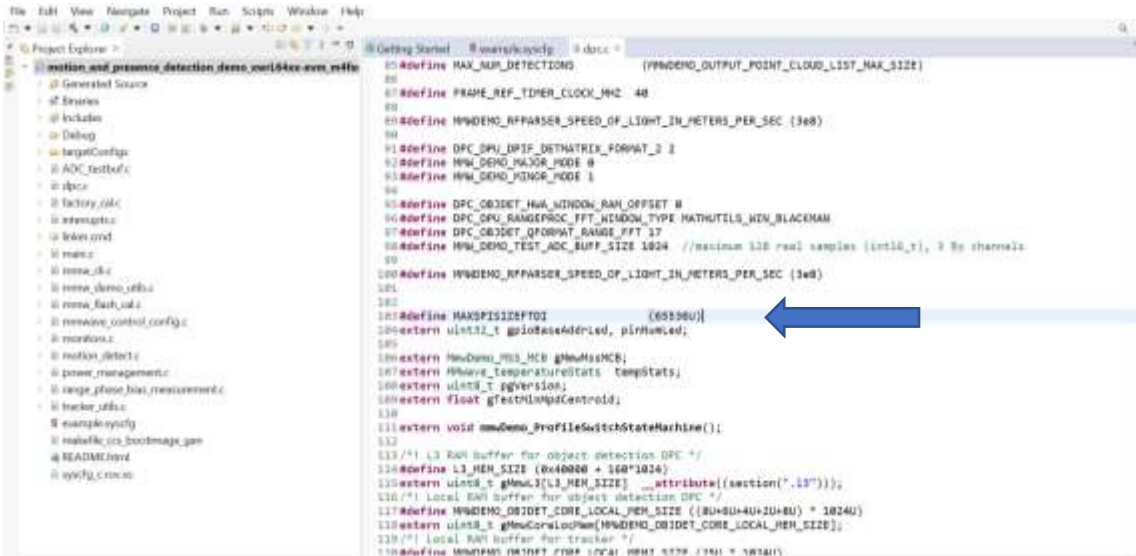


Steps for Raw ADC Data Streaming in IWR16432:

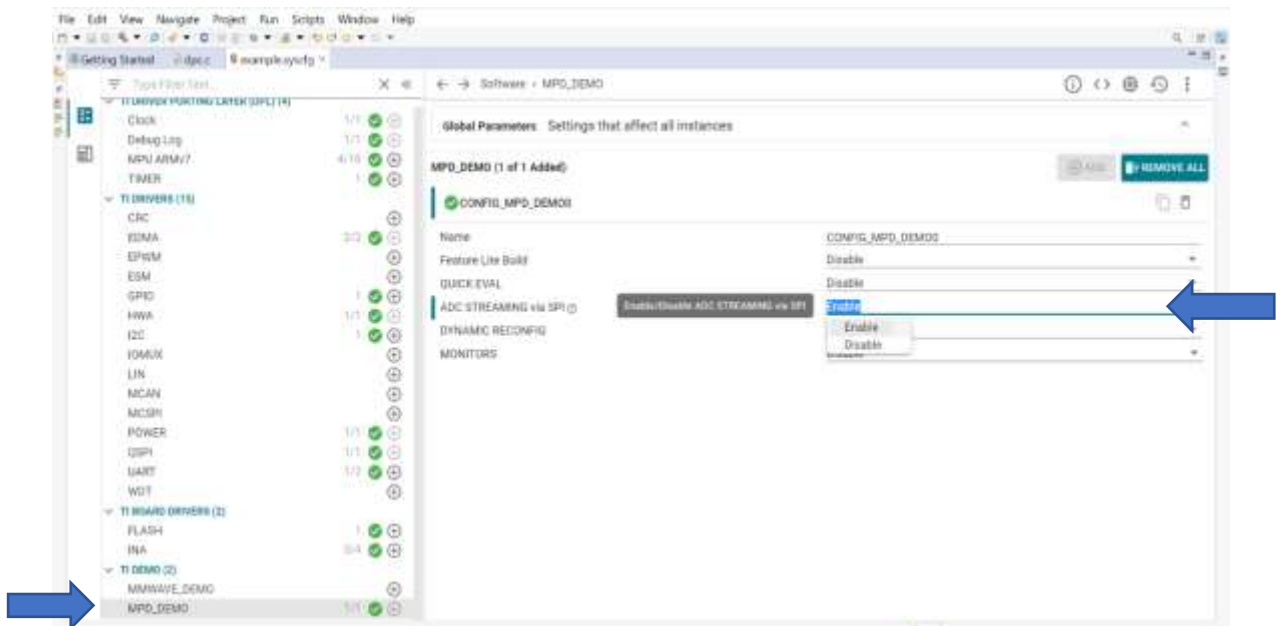
1. In dpc.c file, add these two lines of code.

```
#define MAXSPISIZEFTDI (65536U)  
extern uint32_t gpioBaseAddrLed, pinNumLed;
```



```
File Edit View Navigate Project Run Scripts Window Help  
Project Explorer  
motion_and_presence_detection_demo_v00164x-evm_m3fa  
  Generated Source  
  Binaries  
  Includes  
  targetConfig  
  ADC_testbuf.c  
  dpc.c  
  factory.o/c  
  streamguts.c  
  iwr16432  
  iwr16432.o/c  
  memc.c  
  memc.o/c  
  memc_demo.o/c  
  memc_flash.o/c  
  memc_wake_control.config.c  
  memc_wake.o/c  
  motion_detect.c  
  power_management.c  
  range_finder_hm_resumemem.c  
  tracker.o/c  
  example.syscfg  
  makefile_cc_bootmake.gen  
  README.html  
  syscfg.crc32  
Getting Started example.syscfg dpc.c  
05 #define MAX_NUM_DETECTIONS (MPWDEMO_OUTPUT_POINT_CLOUD_LIST_MAX_SIZE)  
06  
07 #define FRAME_REF_TIMER_CLOCK_MHZ 48  
08  
09 #define MPWDEMO_RFFPARSER_SPEED_OF_LIGHT_IN_METERS_PER_SEC (3e8)  
10  
11 #define DFC_CPU_DP1F_DETHATRIX_FORMAT_2_1  
12 #define MPW_DEMO_MAJOR_MODE 0  
13 #define MPW_DEMO_MINOR_MODE 1  
14  
15 #define DFC_OBJDET_HWA_WINDOW_RAM_OFFSET 8  
16 #define DFC_CPU_RANGEPROC_FFT_WINDOW_TYPE MATHUTILS_MIN_BLACKMAN  
17 #define DFC_OBJDET_QFORMAT_RANGE_FFT 17  
18 #define MPW_DEMO_TEST_ADC_BUFF_SIZE 1024 //maximum 138 real samples (int16_t), 3 Bx channels  
19  
20 #define MPWDEMO_RFFPARSER_SPEED_OF_LIGHT_IN_METERS_PER_SEC (3e8)  
21  
22  
23 #define MAXSPISIZEFTDI (65536U)  
24 extern uint32_t gpioBaseAddrLed, pinNumLed;  
25  
26 extern MemDemo_HLS_MCB gMemHlsMcb;  
27 extern #hwave_temperatureStats tempStats;  
28 extern uint8_t sgversion;  
29 extern float gTechLampCentroid;  
30  
31 extern void memDemo_HyoFileSwitchStateMaximize();  
32  
33 /* L3 RAM buffer for object detection DFC */  
34 #define L3_MEM_SIZE (0x40000 + 160*1024)  
35 extern uint8_t gMem3[L3_MEM_SIZE] __attribute__((section(".L3")));  
36 /* Local RAM buffer for object detection DFC */  
37 #define MPWDEMO_OBJDET_CORE_LOCAL_MEM_SIZE ((20+20+40+20+20) * 1024U)  
38 extern uint8_t gMemCoreLocalMem[MPWDEMO_OBJDET_CORE_LOCAL_MEM_SIZE];  
39 /* Local RAM buffer for tracker */  
40 #define MPWDEMO_OBJDET_CORE_LOCAL_MEM_SIZE ((20 + 20 + 40 + 20 + 20) * 1024U);
```

2. Open example.syscfg and click MPD_DEMO. In that enable the field ADC Streaming via SPI.

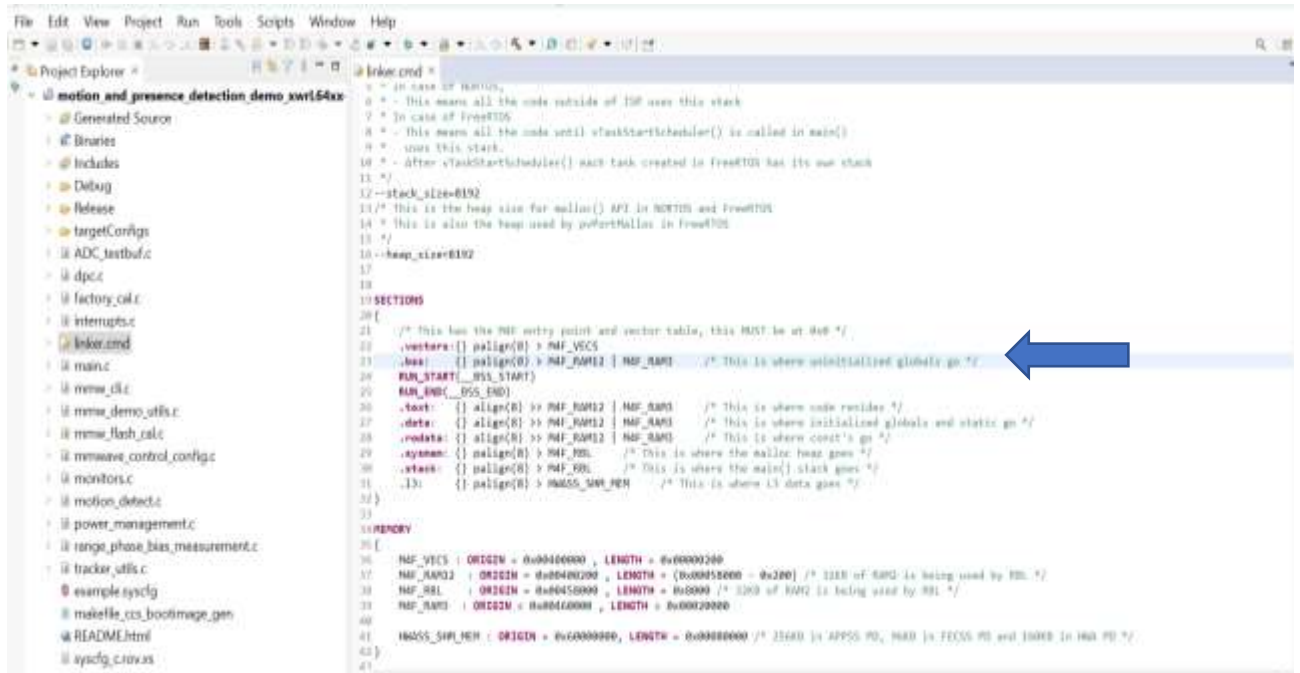


3. Disable the default configuration sent out via mmw_cli.c



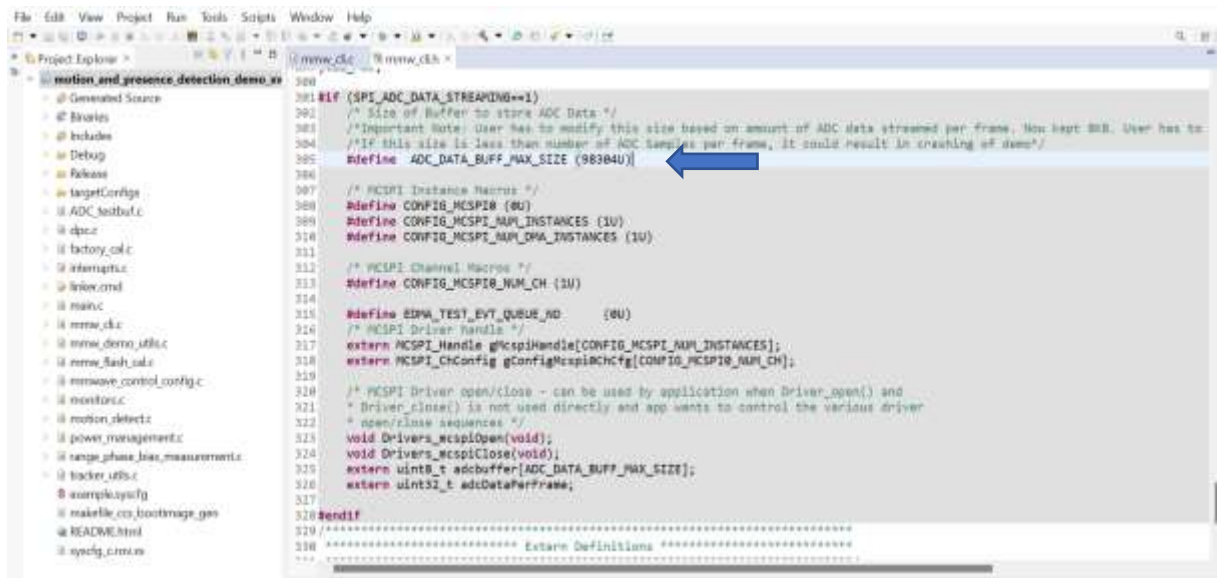
```
410 #define MAX_PRESENCE_CMD 16
411 #define MAX_RADAR_CMD 20
412 #define MAX_RADAR_CMD_ASP 21
413 char* radarCmdString[MAX_RADAR_CMD] =
414 {
415     #if !defined(SOC_XWR154XX)
416     "channelCfg 7 3 0 '\n'",
417     "chirpCoefCfg 18 0 0 126 4 30 0 '\n'",
418     "chirpTimingCfg 6.00 20 0.00 90 59.75 '\n'",
419     #else
420     "chirpTimingCfg 6.00 20 0.00 90 77 '\n'",
421     #endif
422     "frmsaCfg 8 0 403 1 210 0 0 '\n'",
423     "gulMonitor 0 0 0 0 0 1 0 0 0 1 '\n'",
424     "sigProcChainCfg 64 8 2 0 4 4 0 3 '\n'",
425     "cfarCfg 2 4 3 2 0 12.00 0 0 0 1 0 1 '\n'",
426     "rangeCoefCfg -40.00 50.00 -40.00 40.00 '\n'",
427     "clusterRemoval 3 '\n'",
428     "compRangeEliminPrxChanPhase 0.00 1.00 0.00 -1.00 0.00 1.00 0.00 -1.00 0.00 1.00 0.00 -1.00 0.00 '\n'",
429     "adcDataSource 0 adcData_1_000 bin '\n'",
430     "testLogging 0 '\n'",
431     "loadPowerCfg 0 '\n'",
432     "factoryCalibCfg 1 0 40 0 0x1f9000 '\n'",
433     "tpdBoundaryBox 1 -0.10 0.10 0.07 0.6 -0.10 0.10 '\n'",
434     "sensorPosition 0.00 0.00 0.00 0.00 0.00 '\n'",
435     "tdmnrStateCfg 5 4 40 0 4 30.00 4 4 '\n'",
436     "clusterCfg 1 0.50 2 '\n'",
437     "sensorStart 0 0 0 0 '\n'
438 #endif
439 };
```

4. Update the linker.cmd file as described below.



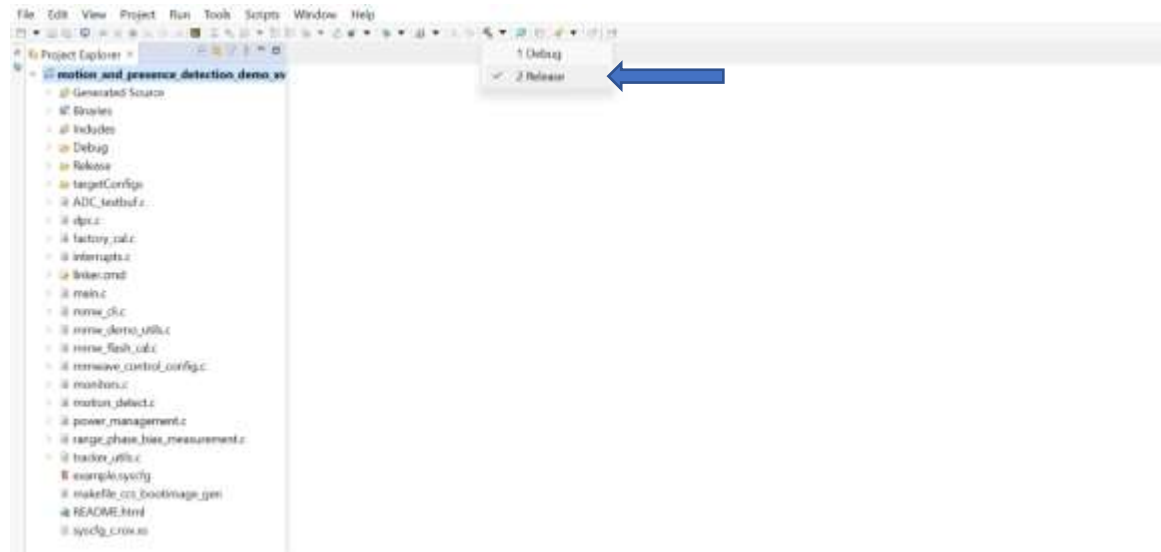
```
11 * This means all the code outside of ISF uses this stack
12 * In case of FreeRTOS
13 * This means all the code until vTaskStartScheduler() is called in main()
14 * starts this stack.
15 * After vTaskStartScheduler() each task created in FreeRTOS has its own stack
16 */
17 --stack_size=0192
18 /* This is the heap size for malloc() API in NORTOS and FreeRTOS
19 * This is also the heap used by pvPortMalloc in FreeRTOS
20 */
21 --heap_size=0192
22
23 SECTIONS
24 {
25     /* This has the PEI entry point and vector table, this MUST be at 0x0 */
26     .vectors: {} ALIGN(8) > MIF_VECTS
27     .bss: {} ALIGN(8) > MIF_RAM12 | MIF_RAM3 /* This is where uninitialized globals go */
28     .RAM_START_0SS_START:
29     RAM_START_0SS_END
30     .text: {} ALIGN(8) >> MIF_RAM12 | MIF_RAM3 /* This is where code resides */
31     .data: {} ALIGN(8) >> MIF_RAM12 | MIF_RAM3 /* This is where initialized globals and static go */
32     .rodata: {} ALIGN(8) >> MIF_RAM12 | MIF_RAM3 /* This is where const's go */
33     .system: {} ALIGN(8) > MIF_RAM /* This is where the malloc heap goes */
34     .system: {} ALIGN(8) > MIF_RAM /* This is where the malloc heap goes */
35     .I3: {} ALIGN(8) > IM055_SMP_MEM /* This is where I3 data goes */
36 }
37
38 MEMORY
39 {
40     MIF_VECTS : ORIGIN = 0x00400000 , LENGTH = 0x0000200
41     MIF_RAM12 : ORIGIN = 0x00400200 , LENGTH = (0x00050000 - 0x200) /* 11KB of SRAM is being used by PEI */
42     MIF_RAM3 : ORIGIN = 0x00450000 , LENGTH = 0x00000000 /* 1KB of RAM is being used by PEI */
43     MIF_RAM3 : ORIGIN = 0x00460000 , LENGTH = 0x00020000
44
45     IM055_SMP_MEM : ORIGIN = 0x6A000000 , LENGTH = 0x00000000 /* 256KB is APP50, IM40 is PECS50 and IM000 is IM40 */
46 }
47
```

5. Update the ADC buffer size as per requirement in mmw_cli.h



```
300
301 #if (SPI_ADC_DATA_STREAMING==1)
302 /* Size of Buffer to store ADC Data */
303 /*Important Note: User has to modify this size based on amount of ADC data streamed per frame. Max kept 8192. User has to
304 /*If this size is less than number of ADC samples per frame, it could result in crashing of demo*/
305 #define ADC_DATA_BUFF_MAX_SIZE (983040)
306
307 /* MCSPI Instance Macro */
308 #define CONFIG_MCSPI0 (0U)
309 #define CONFIG_MCSPI_NUM_INSTANCES (1U)
310 #define CONFIG_MCSPI_NUM_DMA_INSTANCES (1U)
311
312 /* MCSPI Channel Macro */
313 #define CONFIG_MCSPI0_NUM_CH (1U)
314
315 #define EDMA_TEST_EVT_QBUS_ID (0U)
316 /* MCSPI Driver Handle */
317 extern MCSPI_Handle gMcspiHandle[CONFIG_MCSPI_NUM_INSTANCES];
318 extern MCSPI_ChConfig gConfigMcspiChCfg[CONFIG_MCSPI0_NUM_CH];
319
320 /* MCSPI Driver open/close - can be used by application when Driver_open() and
321 * Driver_close() is not used directly and app wants to control the various driver
322 * open/close sequences */
323 void Drivers_mcspiOpen(void);
324 void Drivers_mcspiClose(void);
325 extern uint8_t adcbuffer[ADC_DATA_BUFF_MAX_SIZE];
326 extern uint32_t adcDataPerFrame;
327
328 #endif
329
330 ***** External Definitions *****
331
```

6. Ensure that the build is Release version and rebuild the project.



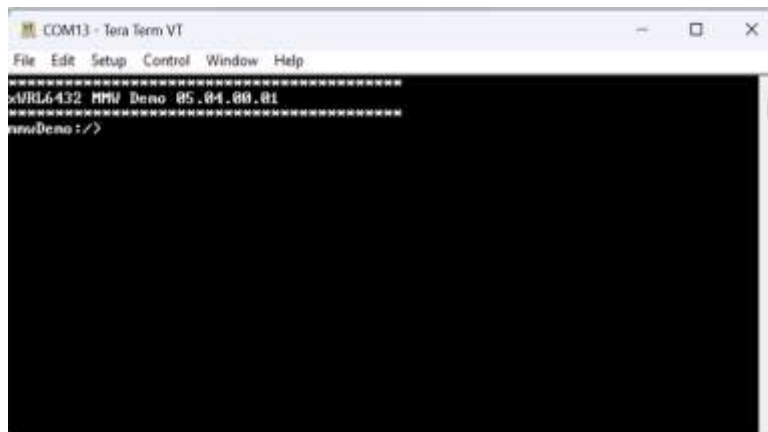
- Flash the release appimage built in the previous step.
- Setup the device for SPI transfer by ensuring switch S1.1 and S1.6 are ON.
- Open tera term and choose the corresponding COM Port.



- Setup the serial port with baud rate 115200.



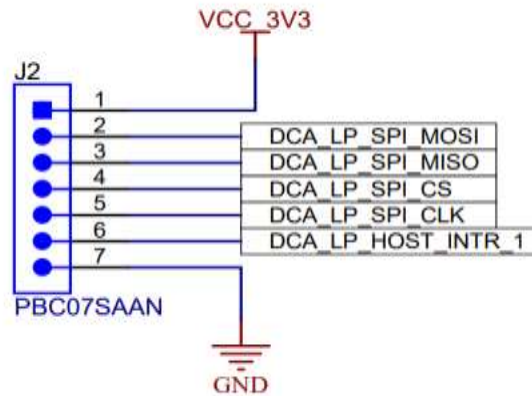
- Press reset switch from FCCSP. Once pressed, tera term should look like this.



13.1 Connection Table of SPI Interface of xWRL6432 with C232HM-DDHSL-0 Cable

XWRLx4XX FCCSP Device	C232HM-DDHSL-0 Cable
MOSI	YELLOW WIRE
MISO	GREEN WIRE
CHIP SELECT	BROWN WIRE
SPI CLOCK	ORANGE WIRE
SPI BUSY	GREY WIRE
GROUND	BLACK WIRE

14. SPI Header for FTDI Interface (DCA_LP_HOST_INTR_1 is SPI BUSY SIGNAL)



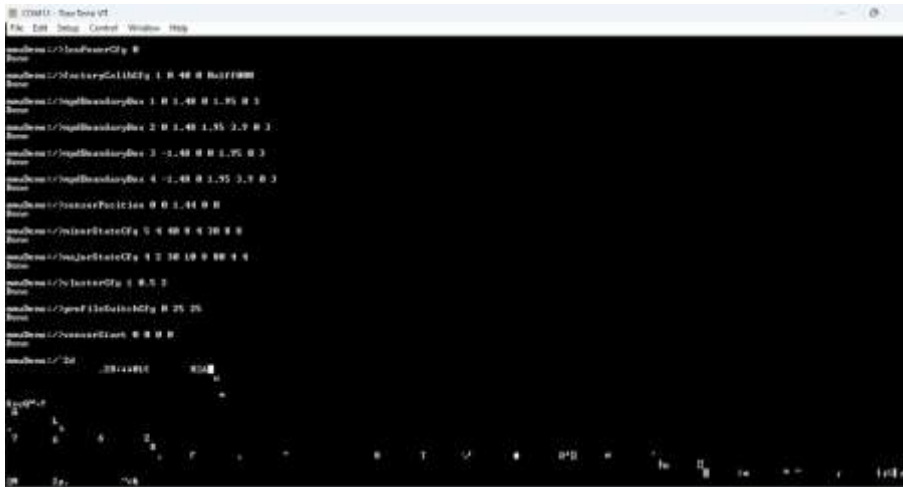
15. Open the adcDataSPIFTDI application and provide all the required parameters.

```

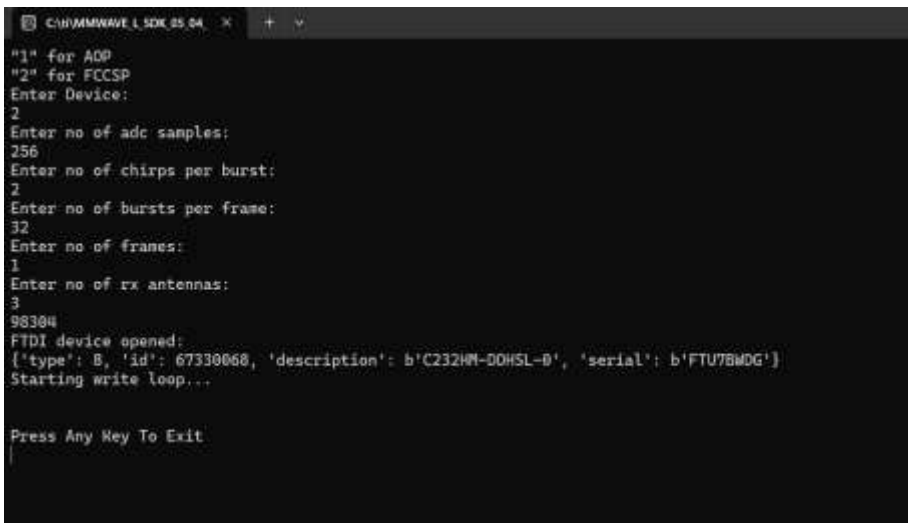
C:\MMMWAVE_L5DK_05_04 x + v
"1" for ADP
"2" for FCCSP
Enter Device:
2
Enter no of adc samples:
256
Enter no of chirps per burst:
2
Enter no of bursts per frame:
32
Enter no of frames:
1
Enter no of rx antennas:
3
98384
FTDI device opened:
{'type': 8, 'id': 67238068, 'description': b'C232HM-DDHSL-0', 'serial': b'FTU7Bw0G'}
Starting write loop...

```

16. Now, Issue sensor start command from tera term.



17. Open adcDataSPIFTDI application and press any key to exit.




18. In the same directory, adc data would be stored in a text file.

Name	Date modified	Type	Size
source	4/23/2024 4:24 PM	File folder	
adcddata	5/13/2024 11:13 AM	Text Source File	232 KB
adcDataSPIFTDI	3/4/2024 7:42 PM	Application	21,323 KB

Post Processing of Captured Data:

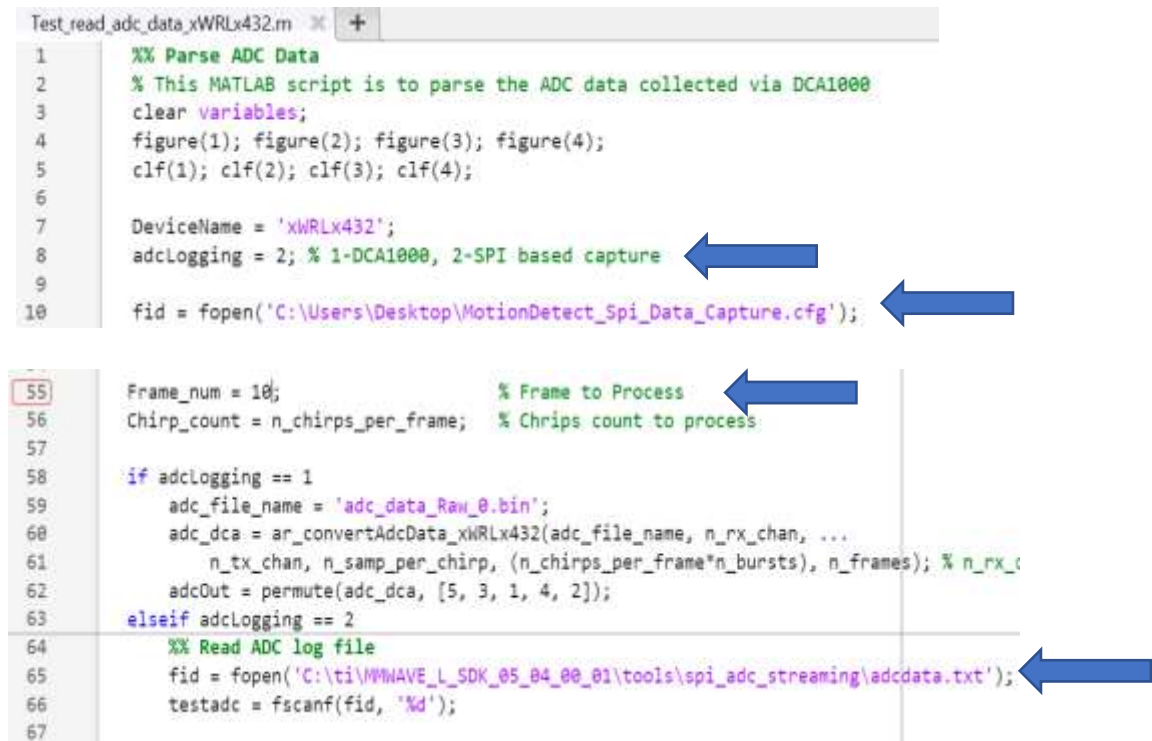
1. Data Format of the captured values in adcddata.txt file : Frame1 Chirp1 Antenna1, Frame1 Chirp1 Antenna2 and so on.

Example : 259 – Frame 1 Chirp1 Antenna1; 246 – Frame 1 Chirp1 Antenna2;



```
adcddata.txt
1 259
2 246
3 252
4 264
5 261
6 247
7 271
8 306
9 318
10 284
11 262
12 261
13 265
14 257
15 259
16 284
17 308
```

2. The captured data can be used to perform post processing with the Matlab script provided in {SDK_INSTALL_DIR}/tools/ADC_parser/Test_read_adc_data_xWRLx432.m
3. Once the above mentioned script is opened in Matlab, ensure that adcllogging is 2 in the script. Also, provide the path for adcddata.txt file and configuration file used. Only one frame could be processed with the script. The specific frame number to be processed should be indicated in the script.



```
Test_read_adc_data_xWRLx432.m
1 %% Parse ADC Data
2 % This MATLAB script is to parse the ADC data collected via DCA1000
3 clear variables;
4 figure(1); figure(2); figure(3); figure(4);
5 clf(1); clf(2); clf(3); clf(4);
6
7 DeviceName = 'xWRLx432';
8 adcllogging = 2; % 1-DCA1000, 2-SPI based capture
9
10 fid = fopen('C:\Users\Desktop\MotionDetect_Spi_Data_Capture.cfg');

55 Frame_num = 10; % Frame to Process
56 Chirp_count = n_chirps_per_frame; % Chirps count to process
57
58 if adcllogging == 1
59     adc_file_name = 'adc_data_Raw_0.bin';
60     adc_dca = ar_convertAdcData_xWRLx432(adc_file_name, n_rx_chan, ...
61         n_tx_chan, n_samp_per_chirp, (n_chirps_per_frame*n_bursts), n_frames); % n_rx_c
62     adcOut = permute(adc_dca, [5, 3, 1, 4, 2]);
63 elseif adcllogging == 2
64     %% Read ADC log file
65     fid = fopen('C:\ti\MMWAVE_L_SDK_05_04_00_01\tools\spi_adc_streaming\adcddata.txt');
66     testadc = fscanf(fid, '%d');
67
```

4. Run the script to view the output figures.

Sample Output Figures:

