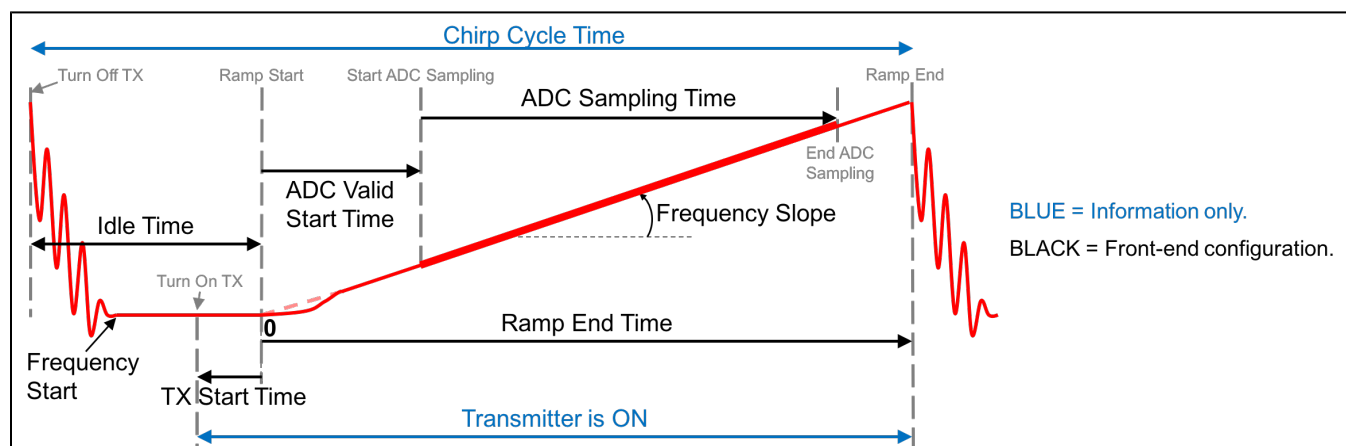




Converting configuration from older SDK release to current SDK release

As new versions of SDK releases are available, there are usually changes to the configuration commands that are supported in the new release. Now, users may have some hand crafted config file which worked perfectly well on older SDK release version but will not work as is with the new SDK release. If user desires to run the same configuration against the new SDK release, then there is a script `mmwDemo_<platform>_update_config.pl` provided in the `mmwave_sdk_<ver>\packages\ti\demo\<platform>\mmw\profiles` directory that they can use to convert the configuration file from older release to a compatible version for the new release. Refer to the perl file for details on how to run the script. Note that users will need to install perl on their machine (there is no strict version requirement at this time). For any new commands inserted by the script, there will be a comment preceding that line which is similar to something like this: "Inserting new mandatory command. Check users guide for details."

Most of the parameters described below are the same as the mmwavelink API specifications (see doxygen `mmwave_sdk_<ver>\packages\ti\control\mmwavelink\docs\doxygen\html\index.html`.) Additionally, users can refer to the chirp diagram below to understand the chirp and profile related parameters or the appnote <http://www.ti.com/litv/pdf/swra553>




3. 4. 0. 0. 1 Figure 2: Chirp Diagram

Configuration command	Command details	Command Parameters	Usage in mmW demo xwr16xx/xwr18xx/xwr64xx /xwr68xx
dfeDataOutputMode	<p>The values in this command should not change between sensorStop and sensorStart.</p> <p>Reboot the board to try config with different set of values in this command</p> <p>This is a mandatory command.</p>	<p><modeType></p> <p>1 - frame based chirps 2 - continuous chirping 3 - advanced frame config</p>	only option 1 and 3 are supported
channelCfg	<p>Channel config message to RadarSS. See mmwavelink doxygen for details.</p> <p>The values in this command should not change between sensorStop and sensorStart.</p> <p>Reboot the board to try config with different set of values in this command</p> <p>This is a mandatory command.</p>	<p><rxChannelEn></p> <p>Receive antenna mask e.g for 4 antennas, it is 0x1111b = 15</p>	4 antennas supported

		<p><txChannelEn> Transmit antenna mask</p>	<p>Refer to the antenna layout on the EVM/board to determine the right Tx antenna mask needed to enable the desired virtual antenna configuration.</p> <p>For example, in IWR6843 ISK, the 2 azimuth antennas can be enabled using bitmask 0x5 (i.e. tx1 and tx3). The azimuth and elevation antennas can both be enabled using bitmask 0x7 (i.e. tx1, tx2 and tx3).</p> <p>For example, in xWR1642BOOST, the 2 azimuth antennas can be enabled using bitmask 0x3 (i.e. tx1 and tx2).</p>
		<p><cascading> SoC cascading, not applicable, set to 0</p>	n/a
adcCfg	<p>ADC config message to RadarSS. See mmwavelink doxygen for details.</p> <p>The values in this command should not change between sensorStop and sensorStart.</p> <p>Reboot the board to try config with different set of values in this command</p> <p>This is a mandatory command.</p>	<p><numADCBits> Number of ADC bits (0 for 12-bits, 1 for 14-bits and 2 for 16-bits)</p> <p><adcOutputFmt> Output format : 0 - real 1 - complex 1x (image band filtered output) 2 - complex 2x (image band visible))</p>	<p>only 16-bit is supported</p> <p>only complex modes are supported</p>
adcbufCfg	<p>adcbuf hardware config. The values in this command can be changed between sensorStop and sensorStart.</p> <p>This is a mandatory command.</p>	<p><subFrameIdx> subframe Index</p> <p><adcOutputFmt> ADCBUF out format 0-Complex, 1-Real</p> <p><SampleSwap> ADCBUF IQ swap selection: 0-I in LSB, Q in MSB, 1-Q in LSB, I in MSB</p> <p><ChanInterleave> ADCBUF channel interleave configuration: 0 - interleaved(only supported for XWR14xx), 1 - non-interleaved</p> <p><ChirpThreshold> Chirp Threshold configuration used for ADCBUF buffer to trigger ping/pong buffer switch.</p> <p>Valid values:</p> <p>0-8 for demos that use DSP for 1D FFT and LVDS streaming is disabled</p> <p>only 1 for demos that use HWA for 1D FFT</p>	<p>For legacy mode, that field should be set to -1.</p> <p>For advanced frame mode, it should be set to either the intended subframe number or -1 to apply same config to all subframes.</p> <p>only complex modes are supported</p> <p>only option 1 is supported</p> <p>only option 1 is supported</p> <p>xwr16xx demo: Values 0-8 are supported since it uses DSP for 1D FFT. However, only value of 1 is supported when LVDS streaming is enabled.</p> <p>xwr64xx/xwr68xx/xwr18xx: only value of 1 is supported since these demos use HWA for 1D FFT</p>



profileCfg	<p>Profile config message to RadarSS and datapath. See mmwavelink doxygen for details. The values in this command can be changed between sensorStop and sensorStart.</p> <p>This is a mandatory command.</p> <div data-bbox="480 321 940 531" style="border: 1px solid orange; padding: 5px;"> <p>txCalibEnCfg Field</p> <p>This CLI command doesn't expose the txCalibEnCfg field in the mmwavelink structure. User should follow the mmwavelink documentation and update the CLI profileCfg handler function accordingly. The current handler sets the value to 0 for this field (backward compatible mode)</p> </div> <div data-bbox="480 552 940 905" style="border: 1px solid #ccc; padding: 5px;"> <p> Combination of numAdcSamples in profileCfg (and numRangeBins), numDopplerChirps = total number of chirps/(num TX in MIMO mode) in frameCfg or subFrameCfg, number of TX and RX antennas in channelCfg and chirpCfg determine the size of Radarcube and other internal buffers/heap in the demo. It is possible that some combinations of these values result in out of memory conditions for these heaps and demo will reject such configuration. Refer to demo and DPC doxygen to understand the data buffer layout and use the system printf's on sensorStart in CCS console window to understand the exact heap usage for a given configuration.</p> </div>		<div data-bbox="954 205 1170 489"> <p><profileId> profile Identifier</p> <p>Legacy frame (dfeOutputMode=1): could be any allowed value but only one valid profile per config is supported</p> <p>Advanced frame (dfeOutputMode=3): could be any allowed value but only one profile per subframe is supported. However, different subframes can have different profiles</p> </div> <div data-bbox="954 499 1170 720"> <p><startFreq> "Frequency Start" in GHz (float values allowed)</p> <p>Examples: 77 61.38</p> <p>Refer to the chirp diagram shown above to understand the relation between various profile parameters and inter-dependent constraints.</p> </div> <div data-bbox="954 730 1170 951"> <p><idleTime> "Idle Time" in u-sec (float values allowed)</p> <p>Examples: 7 7.15</p> <p>Refer to the chirp diagram shown above to understand the relation between various profile parameters and inter-dependent constraints.</p> </div> <div data-bbox="954 961 1170 1182"> <p><adcStartTime> "ADC Valid Start Time" in u-sec (float values allowed)</p> <p>Examples: 7 7.34</p> <p>Refer to the chirp diagram shown above to understand the relation between various profile parameters and inter-dependent constraints.</p> </div> <div data-bbox="954 1192 1170 1413"> <p><rampEndTime> "Ramp End Time" in u-sec (float values allowed)</p> <p>Examples: 58 216.15</p> <p>Refer to the chirp diagram shown above to understand the relation between various profile parameters and inter-dependent constraints.</p> </div> <div data-bbox="954 1423 1170 1476"> <p><txOutPower> Tx output power back-off code for tx antennas</p> <p>only value of '0' has been tested within context of mmW demo</p> </div> <div data-bbox="954 1486 1170 1539"> <p><txPhaseShifter> tx phase shifter for tx antennas</p> <p>only value of '0' has been tested within context of mmW demo</p> </div> <div data-bbox="954 1549 1170 1791"> <p><freqSlopeConst> "Frequency slope" for the chirp in MHz/usec (float values allowed)</p> <p>Examples: 68 16.83</p> <p>any value greater than 0 as per mmwavelink doxygen/device datasheet but represented in MHz /usec.</p> <p>Refer to the chirp diagram shown above to understand the relation between various profile parameters and inter-dependent constraints.</p> </div>
------------	---	--	--

		<p><txStartTime> "TX Start Time" in u-sec (float values allowed)</p> <p>Examples:</p> <p>1 2.92</p>	<p>any value as per mmwavelink doxygen/device datasheet but represented in usec.</p> <p>Refer to the chirp diagram shown above to understand the relation between various profile parameters and inter-dependent constraints.</p>
		<p><numAdcSamples> number of ADC samples collected during "ADC Sampling Time" as shown in the chirp diagram above</p> <p>Examples:</p> <p>256 224</p>	<p>any value as per mmwavelink doxygen/device datasheet but only values greater than 64 have been validated.</p> <p>Refer to the chirp diagram shown above to understand the relation between various profile parameters and inter-dependent constraints.</p>
		<p><digOutSampleRate> ADC sampling frequency in ksp/s.</p> <p>($\frac{\text{<numAdcSamples>}}{\text{<digOutSampleRate>}} = \text{"ADC Sampling Time"}$)</p> <p>Examples:</p> <p>5500</p>	<p>any value as per mmwavelink doxygen/device datasheet.</p> <p>Refer to the chirp diagram shown above to understand the relation between various profile parameters and inter-dependent constraints.</p>
		<p><hpfCornerFreq1> HPF1 (High Pass Filter 1) corner frequency 0: 175 KHz 1: 235 KHz 2: 350 KHz 3: 700 KHz</p>	<p>any value as per mmwavelink doxygen/device datasheet</p>
		<p><hpfCornerFreq2> HPF2 (High Pass Filter 2) corner frequency 0: 350 KHz 1: 700 KHz 2: 1.4 MHz 3: 2.8 MHz</p>	<p>any value as per mmwavelink doxygen/device datasheet</p>
		<p><rxGain> OR'ed value of RX gain in dB and RF gain target (See mmwavelink doxygen for details)</p>	<p>any value as per mmwavelink doxygen/device datasheet</p>
chirpCfg	<p>Chirp config message to RadarSS and datapath. See mmwavelink doxygen for details.</p> <p>The values in this command can be changed between sensorStop and sensorStart.</p> <p>This is a mandatory command.</p>	<p>chirp start index</p> <p>chirp end index</p> <p>profile identifier</p> <p>start frequency variation in Hz (float values allowed)</p> <p>frequency slope variation in kHz/us (float values allowed)</p> <p>idle time variation in u- sec (float values allowed)</p> <p>ADC start time variation in u-sec (float values allowed)</p>	<p>any value as per mmwavelink doxygen</p> <p>any value as per mmwavelink doxygen</p> <p>should match the profileCfg- >profileId</p> <p>only value of '0' has been tested within context of mmWV demo</p> <p>only value of '0' has been tested within context of mmWV demo</p> <p>only value of '0' has been tested within context of mmWV demo</p> <p>only value of '0' has been tested within context of mmWV demo</p>

		tx antenna enable mask (Tx2,Tx1) e.g (10)b = Tx2 enabled, Tx1 disabled.	See note under "Channel Cfg" command above. Individual chirps should have either only one distinct Tx antenna enabled (MIMO) or same TX antennas should be enabled for all chirps
lowPower	Low Power mode config message to RadarSS. See mmwavelink doxygen for details. The values in this command should not change between sensorStop and sensorStart. Reboot the board to try config with different set of values in this command. This is a mandatory command.	<don't_care> ADC Mode 0x00 : Regular ADC mode 0x01 : Low power ADC mode	set to 0 use value of '0' or '1' (depending on profileCfg->digOutSampleRate)
frameCfg	frame config message to RadarSS and datapath. See mmwavelink doxygen for details. dfeOutputMode should be set to 1 to use this command The values in this command can be changed between sensorStop and sensorStart. This is a mandatory command when dfeOutputMode is set to 1.	chirp start index (0-511) chirp end index (chirp start index-511) number of loops (1 to 255) number of frames (valid range is 0 to 65535, 0 means infinite) frame periodicity in ms (float values allowed) trigger select 1: Software trigger 2: Hardware trigger. Frame trigger delay in ms (float values allowed)	any value as per mmwavelink doxygen but corresponding chirpCfg should be defined any value as per mmwavelink doxygen but corresponding chirpCfg should be defined any value as per mmwavelink doxygen/device datasheet but greater than or equal to 4. For xwr16xx/xwr68xx demos where DSP version of Doppler DPU is used, the Doppler chirps (i.e. number of loops) should be a multiple of 4 due to windowing requirement. Note: If value of 2 is desired for number of Doppler Chirps, one must update the demo/object detection DPC source code to use rectangular window for Doppler DPU instead of Hanning window. any value as per mmwavelink doxygen any value as per mmwavelink doxygen and represented in msec. However frame should not have more than 50% duty cycle (i.e. active chirp time should be <= 50% of frame period). Also it should allow enough time for selected UART output to be shipped out (selections based on guiMonitor command) else demo will assert if the next frame start trigger is received from the front end and current frame is still ongoing. User can use the output of stats TLV to tune this parameter. only option for Software trigger is supported any value as per mmwavelink doxygen and represented in msec.
advFrameCfg	Advanced config message to RadarSS and datapath. See mmwavelink doxygen for details. The dfeOutputMode should be set to 3 to use this command. See profile_advanced_subframe.cfg profile in the mmW demo profiles directory for example usage. The values in this command can be changed between sensorStop and sensorStart.	<numOfSubFrames> Number of sub frames enabled in this frame <forceProfile> Force profile	any value as per mmwavelink doxygen only value of 0 is supported



	<p>This is a mandatory command when dfeOutputMode is set to 3.</p>	<p><numFrames> Number of frames to transmit (1 frame = all enabled sub frames)</p>	<p>any value as per mmwavelink doxygen</p>
		<p><triggerSelect> trigger select 1: Software trigger 2: Hardware trigger.</p>	<p>only option for Software trigger is supported</p>
		<p><frameTrigDelay> Frame trigger delay in ms (float values allowed)</p>	<p>any value as per mmwavelink doxygen and represented in msec.</p>
subFrameCfg	<p>Subframe config message to RadarSS and datapath. See mmwavelink doxygen for details.</p> <p>The dfeOutputMode should be set to 3 to use this command. See profile_advanced_subframe.cfg profile in the mmW demo profiles directory for example usage</p> <p>The values in this command can be changed between sensorStop and sensorStart.</p> <p>This is a mandatory command when dfeOutputMode is set to 3.</p>	<p><subFrameNum> subframe Number for which this command is being given</p> <p><forceProfileIdx> Force profile index</p> <p><chirpStartIdx> Start Index of Chirp</p> <p><numOfChirps> Num of unique Chirps per burst including start index</p> <p><numLoops> No. of times to loop through the unique chirps</p> <p><burstPeriodicity> Burst periodicity in msec (float values allowed) and meets the criteria $\text{burstPeriodicity} \geq ((\text{numLoops}) * (\text{Sum total of time duration of all unique chirps in that burst})) + \text{InterBurstBlankTime}$</p> <p><chirpStartIdxOffset> Chirp Start address increment for next burst</p> <p><numOfBurst> Num of bursts in the subframe</p> <p><numOfBurstLoops> Number of times to loop over the set of above defined bursts, in the sub frame</p>	<p>value of 0 to RL_MAX_SUBFRAMES-1</p> <p>ignored as <forceProfile> in advFrameCfg should be set to 0</p> <p>any value as per mmwavelink doxygen but corresponding chirpCfg should be defined</p> <p>any value as per mmwavelink doxygen but corresponding number of chirpCfg should be defined</p> <p>any value as per mmwavelink doxygen but greater than or equal to 4</p> <p>For xwr16xx/xwr68xx demos where DSP version of Doppler DPU is used, the Doppler chirps (i.e. number of loops) should be a multiple of 4 due to windowing requirement.</p> <p>Note: If value of 2 is desired for number of Doppler Chirps, one must update the demo/object detection DPC source code to use rectangular window for Doppler DPU instead of Hanning window.</p> <p>any value as per mmwavelink doxygen and represented in msec but subframe should not have more than 50% duty cycle and allow enough time for selected UART output to be shipped out (selections based on guiMonitor command)</p> <p>set it to 0 since demo supports only one burst per subframe</p> <p>set it to 1 since demo supports only one burst per subframe</p> <p>set it to 1 since demo supports only one burst per subframe</p>



		<p><subFramePeriodicity> subFrame periodicty in msec (float values allowed) and meets the criteria subFramePeriodicity >= Sum total time of all bursts + InterSubFrameBlankTime</p>	<p>set to same as <burstPeriodicity> since demo supports only one burst per subframe</p>
guiMonitor	<p>Plot config message to datapath. The values in this command can be changed between sensorStop and sensorStart.</p> <p>This is a mandatory command.</p>	<p>All parameters below are flags (1 to enable and 0 to disable)</p> <p><subFrameIdx> subframe Index</p> <p><detected objects> 1 - enable export of point cloud (x,y,z, doppler) and point cloud sideinfo (SNR, noiseval) 2 - enable export of point cloud (x,y,z,doppler) 0 - disable</p> <p><log magnitude range> 1 - enable export of log magnitude range profile at zero Doppler 0 - disable</p> <p><noise profile> 1 - enable export of log magnitude noise profile 0 - disable</p> <p><rangeAzimuthHeatMap> range-azimuth heat map related information 1 - enable export of zero Doppler radar cube matrix, all range bins, all antennas to calculate and display azimuth heat map. 0 - disable (the GUI computes the FFT of this to show heat map)</p> <p><rangeDopplerHeatMap> range-doppler heat map 1 - enable export of the whole detection matrix. Note that the frame period should be adjusted according to UART transfer time. 0 - disable</p> <p><statsInfo> statistics (CPU load, margins, etc) 1 - enable export of stats data. 0 - disable</p>	<p>For legacy mode, that field should be set to -1 whereas for advanced frame mode, it should be set to either the intended subframe number or -1 to apply same config to all subframes.</p> <p>all values supported</p> <p>all values supported</p> <p>all values supported</p> <p>all values supported</p> <p>all values supported</p>
cfarCfg	<p>CFAR config message to datapath.</p> <p>The values in this command can be changed between sensorStop and sensorStart and even when the sensor is running.</p> <p>This is a mandatory command.</p>		



		<p><subFrameIdx> subframe Index</p>	<p>For legacy mode, that field should be set to -1 whereas for advanced frame mode, it should be set to either the intended subframe number or -1 to apply same config to all subframes.</p>
		<p><procDirection> Processing direction: 0 – CFAR detection in range direction 1 – CFAR detection in Doppler direction</p>	<p>all values supported; 2 separate commands need to be sent; one for Range and other for doppler.</p>
		<p><mode> CFAR averaging mode: 0 - CFAR_CA (Cell Averaging) 1 - CFAR_CAGO (Cell Averaging Greatest Of) 2 - CFAR_CASO (Cell Averaging Smallest Of)</p>	<p>all values supported</p>
		<p><noiseWin> noise averaging window length: Length of the one sided noise averaged cells in samples</p>	<p>supported</p>
		<p><guardLen> one sided guard length in samples</p>	<p>supported</p>
		<p><divShift> Cumulative noise sum divisor expressed as a shift. Sum of noise samples is divided by $2^{\text{<divShift>}}$. Based on <mode> and <noiseWin>, this value should be set as shown in next columns. The value to be used here should match the "CFAR averaging mode" and the "noise averaging window length" that is selected above. The actual value that is used for division (2^x) is a power of 2, even though the "noise averaging window length" samples may not have that restriction.</p>	<p>CFAR_CA: $\text{<divShift>} = \text{ceil}(\log_2(2 \times \text{<noiseWin>}))$ CFAR_CAGO/_CASO: $\text{<divShift>} = \text{ceil}(\log_2(\text{<noiseWin>}))$ In profile_2d.cfg, value of 3 means that the noise sum is divided by $2^3=8$ to get the average of noise samples with window length of 8 samples in CFAR -CASO mode.</p>
		<p>cyclic mode or Wrapped around mode. 0- Disabled 1- Enabled</p>	<p>supported</p>
		<p>Threshold scale in dB using float representation. This is used in conjunction with the noise sum divisor (say x). the CUT comparison for log input is: $\text{CUT} > (\text{Threshold scale converted from dB to Q8}) + (\text{noise sum} / 2^x)$ For example: 15 10.75</p>	<p>Detection threshold is specified in dB scale. Maximum value allowed is 100dB</p>



		peak grouping 0 - disabled 1 - enabled	supported
multiObjBeamForming	<p>Multi Object Beamforming config message to datapath.</p> <p>This feature allows radar to separate reflections from multiple objects originating from the same range/Doppler detection.</p> <p>The procedure searches for the second peak after locating the highest peak in Azimuth FFT. If the second peak is greater than the specified threshold, the second object with the same range/Doppler is appended to the list of detected objects. The threshold is proportional to the height of the highest peak.</p> <p>The values in this command can be changed between sensorStop and sensorStart and even when the sensor is running.</p> <p>This is a mandatory command.</p>	<p><subFrameIdx> subframe Index</p> <p><Feature Enabled> 0 - disabled 1 - enabled</p> <p><threshold> 0 to 1 – threshold scale for the second peak detection in azimuth FFT output. Detection threshold is equal to <thresholdScale> multiplied by the first peak height. Note that FFT output is magnitude squared.</p>	<p>For legacy mode, that field should be set to -1 whereas for advanced frame mode, it should be set to either the intended subframe number or -1 to apply same config to all subframes.</p> <p>supported</p> <p>supported</p>
calibDcRangeSig	<p>DC range calibration config message to datapath.</p> <p>Antenna coupling signature dominates the range bins close to the radar. These are the bins in the range FFT output located around DC.</p> <p>When this feature is enabled, the signature is estimated during the first N chirps, and then it is subtracted during the subsequent chirps.</p> <p>During the estimation period the specified bins (defined as [negativeBinIdx, positiveBinIdx]) around DC are accumulated and averaged. It is assumed that no objects are present in the vicinity of the radar at that time.</p> <p>This procedure is initiated by the following CLI command, and it can be initiated any time while radar is running. Note that the maximum number of compensated bins is 32.</p> <p>The values in this command can be changed between sensorStop and sensorStart and even when the sensor is running.</p> <p>This is a mandatory command.</p>	<p><subFrameIdx> subframe Index</p> <p><enabled> Enable DC removal using first few chirps 0 - disabled 1 - enabled</p> <p><negativeBinIdx> negative Bin Index (to remove DC from farthest range bins)</p> <p>Maximum negative range FFT index to be included for compensation. Negative indices are indices wrapped around from far end of 1D FFT.</p> <p>Ex: Value of -5 means last 5 bins starting from the farthest bin</p> <p><positiveBinIdx> positive Bin Index (to remove DC from closest range bins)</p> <p>Maximum positive range FFT index to be included for compensation</p> <p>Value of 8 means first 9 bins (including bin#0)</p>	<p>For legacy mode, that field should be set to -1 whereas for advanced frame mode, it should be set to either the intended subframe number or -1 to apply same config to all subframes.</p> <p>supported</p> <p>supported</p> <p>supported</p>

		<p><numAvg> number of chirps to average to collect DC signature (which will then be applied to all chirps beyond this).</p> <p>Value of 256 means first 256 chirps (after command is issued and feature is enabled) will be used for collecting (averaging) DC signature in the bins specified above. From 257th chirp, the collected DC signature will be removed from every chirp.</p>	The value must be power of 2, and must be greater than the number of Doppler bins.
clutterRemoval	<p>Static clutter removal config message to datapath.</p> <p>Static clutter removal algorithm implemented by subtracting from the samples the mean value of the input samples to the 2D-FFT</p> <p>The values in this command can be changed between sensorStop and sensorStart and even when the sensor is running.</p> <p>This is a mandatory command.</p>	<p><subFrameIdx> subframe Index</p> <p><enabled> Enable static clutter removal technique 0 - disabled 1 - enabled</p>	<p>For legacy mode, that field should be set to -1 whereas for advanced frame mode, it should be set to either the intended subframe number or -1 to apply same config to all subframes.</p> <p>supported</p>
aoaFovCfg	<p>Command for datapath to filter out detected points outside the specified range in azimuth or elevation plane</p> <p>The values in this command can be changed between sensorStop and sensorStart and even when the sensor is running.</p> <p>This is a mandatory command.</p>	<p><subFrameIdx> subframe Index</p> <p><minAzimuthDeg></p> <p><maxAzimuthDeg></p> <p><minElevationDeg></p> <p><maxElevationDeg></p>	<p>For legacy mode, that field should be set to -1 whereas for advanced frame mode, it should be set to either the intended subframe number or -1 to apply same config to all subframes.</p> <p>minimum azimuth angle (in degrees) that specifies the start of field of view</p> <p>maximum azimuth angle (in degrees) that specifies the end of field of view</p> <p>minimum elevation angle (in degrees) that specifies the start of field of view</p> <p>maximum elevation angle (in degrees) that specifies the end of field of view</p>
cfarFovCfg	<p>Command for datapath to filter out detected points outside the specified limits in the range direction or doppler direction</p> <p>The values in this command can be changed between sensorStop and sensorStart and even when the sensor is running.</p> <p>This is a mandatory command.</p>	<p><subFrameIdx> subframe Index</p> <p><procDirection> Processing direction: 0 – point filtering in range direction 1 – point filtering in Doppler direction</p>	<p>For legacy mode, that field should be set to -1 whereas for advanced frame mode, it should be set to either the intended subframe number or -1 to apply same config to all subframes.</p> <p>both values supported but this command should be given twice - one for range direction and other for doppler direction</p>



		<p><min (meters or m/s)> the units depends on the value for <procDirection> field above. meters for Range direction and meters/sec for Doppler direction</p>	<p>minimum limits for the range or doppler below which the detected points are filtered out</p>
		<p><max (meters or m/s)> the units depends on the value for <procDirection> field above. meters for Range direction and meters/sec for Doppler direction</p>	<p>maximum limits for the range or doppler above which the detected points are filtered out</p>
compRangeBiasAndRxChanPhase	<p>Command for datapath to compensate for bias in the range estimation and receive channel gain and phase imperfections. Refer to the procedure mentioned here</p> <p>The values in this command can be changed between sensorStop and sensorStart and even when the sensor is running.</p> <p>This is a mandatory command.</p>	<p><rangeBias> Compensation for range estimation bias in meters</p> <p><Re(0,0)> <Im(0,0)> <Re(0,1)> <Im(0,1)> ... <Re(0,R-1)> <Im(0,R-1)> <Re(1,0)> <Im(1,0)> ... <Re(T-1,R-1)> <Im(T-1,R-1)></p> <p>Set of Complex value representing compensation for virtual Rx channel phase bias in Q15 format. Pairs of I and Q should be provided for all Tx and Rx antennas in the device</p>	<p>supported</p> <p>For xwr1843, xwr6843 and xwr6443 demos: 12 pairs of values should be provided here since the device has 4 Rx and 3 Tx (total of 12 virtual antennas)</p> <p>For xwr1642 demo: 8 pairs of values should be provided here since the device has 4 Rx and 2 Tx (total of 8 virtual antennas)</p>
measureRangeBiasAndRxChanPhase	<p>Command for datapath to enable the measurement of the range bias and receive channel gain and phase imperfections. Refer to the procedure mentioned here</p> <p>The values in this command can be changed between sensorStop and sensorStart and even when the sensor is running.</p> <p>This is a mandatory command.</p>	<p><enabled> 1 - enable measurement. This parameter should be enabled only using the profile_calibration.cfg profile in the mmW demo profiles directory 0 - disable measurement. This should be the value to use for all other profiles.</p> <p><targetDistance> distance in meters where strong reflector is located to be used as test object for measurement. This field is only used when measurement mode is enabled.</p> <p><searchWin> distance in meters of the search window around <targetDistance> where the peak will be searched</p>	<p>supported</p> <p>supported</p> <p>supported</p>



extendedMaxVelocity	<p>Velocity disambiguation config message to datapath. A simple technique for velocity disambiguation is implemented. It corrects target velocities up to (2*vmax). The output of this feature may not be reliable when two or more objects are present in the same range bin and are too close in azimuth plane.</p> <p>The values in this command can be changed between sensorStop and sensorStart and even when the sensor is running.</p> <p>This is a mandatory command.</p>	<subFrameIdx> subframe Index	For legacy mode, that field should be set to -1 whereas for advanced frame mode, it should be set to either the intended subframe number or -1 to apply same config to all subframes.
		<enabled> Enable velocity disambiguation technique 0 - disabled 1 - enabled	supported.
CQRxSatMonitor	<p>Rx Saturation Monitoring config message for Chirp quality to RadarSS and datapath. See mmwavelink doxygen for details on rIRxSatMonConf_t.</p> <p>The enable/disable for this command is controlled via the "analogMonitor" CLI command.</p> <p>The values in this command can be changed between sensorStop and sensorStart.</p> <p>This is a mandatory command</p>	<profile> Valid profile Id for this monitoring configuration. This profile ID should have a matching profileCfg.	any value as per mmwavelink doxygen but corresponding profileCfg should be defined
		<satMonSel> RX Saturation monitoring mode	any value as per mmwavelink doxygen
		<priSliceDuration> Duration of each slice, 1LSB=0.16us, range: 4 - number of ADC samples	any value as per mmwavelink doxygen
		<numSlices> primary + secondary slices ,range 1-127. Maximum primary slice is 64.	any value as per mmwavelink doxygen
		<rxChanMask> RX channel mask, 1 - Mask, 0 - unmask	any value as per mmwavelink doxygen
CQSigImgMonitor	<p>Signal and image band energy Monitoring config message for Chirp quality to RadarSS and datapath. See mmwavelink doxygen for details on rISigImgMonConf_t.</p> <p>The enable/disable for this command is controlled via the "analogMonitor" CLI command.</p> <p>The values in this command can be changed between sensorStop and sensorStart.</p> <p>This is a mandatory command</p>	<profile> Valid profile Id for this monitoring configuration. This profile ID should have a matching profileCfg	any value as per mmwavelink doxygen but corresponding profileCfg should be defined
		<numSlices> primary + secondary slices, range 1-127. Maximum primary slice is 64.	any value as per mmwavelink doxygen
		<numSamplePerSlice> Possible range is 4 to "number of ADC samples" in the corresponding profileCfg. It must be an even number.	any value as per mmwavelink doxygen
analogMonitor	<p>Controls the enable/disable of the various monitoring features supported in the demos.</p> <p>The values in this command can be changed between sensorStop and sensorStart.</p> <p>This is a mandatory command.</p>	<rxSaturation> CQRxSatMonitor enable /disable 1:enable 0: disable	"enable" supported only for xwr16xx and xwr18xx demo /device. This feature is not yet supported on xwr68xx device /RadarSS, however the command must still be provided with value set to disabled.
		<sigImgBand> CQSigImgMonitor enable /disable 1:enable 0: disable	"enable" supported only for xwr16xx and xwr18xx demo /device. This feature is not yet supported on xwr68xx device /RadarSS, however the command must still be provided with value set to disabled.



lvdsStreamCfg	<p>Enables the streaming of various data streams over LVDS lanes. When this feature is enabled, make sure chirpThreshold in adcbufCfg is set to 1.</p> <p>The values in this command can be changed between sensorStop and sensorStart.</p> <p>This is a mandatory command.</p>	<subFrameldx> subframe Index	For legacy mode, that field should be set to -1 whereas for advanced frame mode, it should be set to either the intended subframe number or -1 to apply same config to all subframes.
		<enableHeader> 0 - Disable HSI header for all active streams 1 - Enable HSI header for all active streams	supported
		<dataFmt> Controls HW streaming. Specifies the HW streaming data format. 0-HW STREAMING DISABLED 1-ADC 4-CP_ADC_CQ*	supported CP_ADC_CQ is not yet supported on xwr68xx device/RadarSS 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> 0 - Disable user data (SW session) 1 - Enable user data (SW session) <enableHeader> should be set to 1 when this field is enabled.	supported
bpmCfg	<p>BPM MIMO configuration.</p> <p>Every frame should consist of alternating chirps with pattern TXA+TxB and TXA-TXB where TXA and TXB are two azimuth TX antennas. This is alternate configuration to TDM-MIMO scheme and provides SNR improvement by running 2Tx simultaneously. When using this scheme, user should enable both the azimuth TX in the chirpCfg. See profile_2d_bpm.cfg profile in the xwr16xx mmW demo profiles directory for example usage.</p> <p>This config is supported and mandatory only for demos that use Doppler DSP DPU (xwr16xx/xwr68xx). This config is not supported and is not needed for demos that use Doppler HWA DPU (xwr18xx/xwr64xx).</p>	<subFrameldx> subframe Index	For legacy mode, that field should be set to -1 whereas for advanced frame mode, it should be set to either the intended subframe number or -1 to apply same config to all subframes.
		<enabled> 0-Disabled 1-Enabled	supported This feature cannot be set to enabled xwr68xx demo as it is not yet supported in xwr68xx RadarSS.
		<chirp0Idx> BPM enabled: If BPM is enabled in previous argument, this is the chirp index for the first BPM chirp. It will have phase 0 on both TX antennas (TXA+ , TXB+). Note that the chirpCfg command for this chirp index must have both TX antennas enabled. BPM disabled: If BPM is disabled, a BPM disable command (set phase to zero on both TX antennas) will be issued for the chirps in the range [chirp0Idx.. chirp1Idx]	any value as per mmwavelink doxygen but corresponding chirpCfg should be defined



		<p><chirp1Idx></p> <p>BPM enabled: If BPM is enabled, this is the chirp index for the second BPM chirp. It will have phase 0 on TXA and phase 180 on TXB (TXA+, TXB-). Note that the chirpCfg command for this chirp index must have both TX antennas enabled.</p> <p>BPM disabled: If BPM is disabled, a BPM disable command (set phase to zero on both TX antennas) will be issued for the chirps in the range [chirp0Idx..chirp1Idx].</p>	any value as per mmwavelink doxygen but corresponding chirpCfg should be defined
sensorStart	<p>sensor Start command to RadarSS and datapath. Starts the sensor. This function triggers the transmission of the frames as per the frame and chirp configuration. By default, this function also sends the configuration to the mmWave Front End and the processing chain.</p> <p>This is a mandatory command.</p>	<p>Optionally, user can provide an argument 'doReconfig' 0 - Skip reconfiguration and just start the sensor using already provided configuration.</p> <p><any other value> - not supported</p>	supported
sensorStop	<p>sensor Stop command to RadarSS and datapath. Stops the sensor. If the sensor is running, it will stop the mmWave Front End and the processing chain. After the command is acknowledged, a new config can be provided and sensor can be restarted or sensor can be restarted without a new config (i.e. using old config). See 'sensorStart' command.</p> <p>This is mandatory before any reconfiguration is performed post sensorStart.</p>		supported
flushCfg	<p>This command should be issued after 'sensorStop' command to flush the old configuration and provide a new one.</p> <p>This is mandatory before any reconfiguration is performed post sensorStart.</p>		
%		Any line starting with '%' character is considered as comment line and is skipped by the CLI parsing utility.	

3. 4. 0. 0. 1Table 1: mmWave SDK Demos - CLI commands and parameters

3. 5. Running the prebuilt unit test binaries (.xer4f and .xe674)

Unit tests for the drivers and components can be found in the respective test directory for that component. See section "[mmWave SDK - TI components](#)" for location of each component's test code. For example, UART test code that runs on TI RTOS is in [mmwave_sdk_<ver>/packages/ti/drivers/uart/test/<platform>](#). In this test directory, you will find .xer4f and .xe674 files (either prebuilt or build as a part of instructions mentioned in "[Building drivers/control components](#)"). Follow the instructions in section "[CCS development mode](#)" to download and execute these unit tests via CCS.

