Application Report
SWRA553A – May 2017 – Revised February 2020

Programming Chirp Parameters in TI Radar Devices

Vivek Dham

ABSTRACT

This application report provides information on how to select the right chirp parameters in a fast FMCW Radar device based on the end application and use case, and program them optimally on TI's radar devices.

Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

Frequency Modulated Continuous Wave (FMCW) mmWave radar sensors are becoming increasingly popular for multiple automotive and industrial applications. The system requirements and care-abouts in each of these applications could be very different. Range requirement, range resolution, max velocity requirement, sensor field of view, data memory, processor MIPS, and so forth are some of the aspects that need to be analyzed based on the end application. Understanding the relationships between the FMCW chirp configuration and system performance parameters helps in selecting the right chirp configurations.

TI's mmwave radar devices (MMIC) provide large flexibility in configuring chirp parameters and also allow multiple chirp configurations in a single frame. The timing parameters are accurately controlled by the digital timing engine and a built-in radio processor without heavy real-time software interference. This document describes the programming of chirp parameters and explains the various system considerations that determine the values for these parameters.

2 Impact of Chirp Configuration on System Parameters

In linear FMCW radars, the transmit (TX) signal is a single tone with its frequency changing linearly with time. This sweep in frequency is commonly referred to as a "chirp". A set of these chirps form a "Frame" and this can be used as the observation window for the radar processing. The various parameters of the chirp ramp (like frequency slope, sweep bandwidth, and so forth) impact the system performance.

Figure 1 depicts a single chirp and the associated timing parameters. Figure 2 shows frame structure that consists of a series of chirps followed by inter frame time. This represents 'Fast FMCW' modulation, where each chirp is typically 10's of µs in duration.

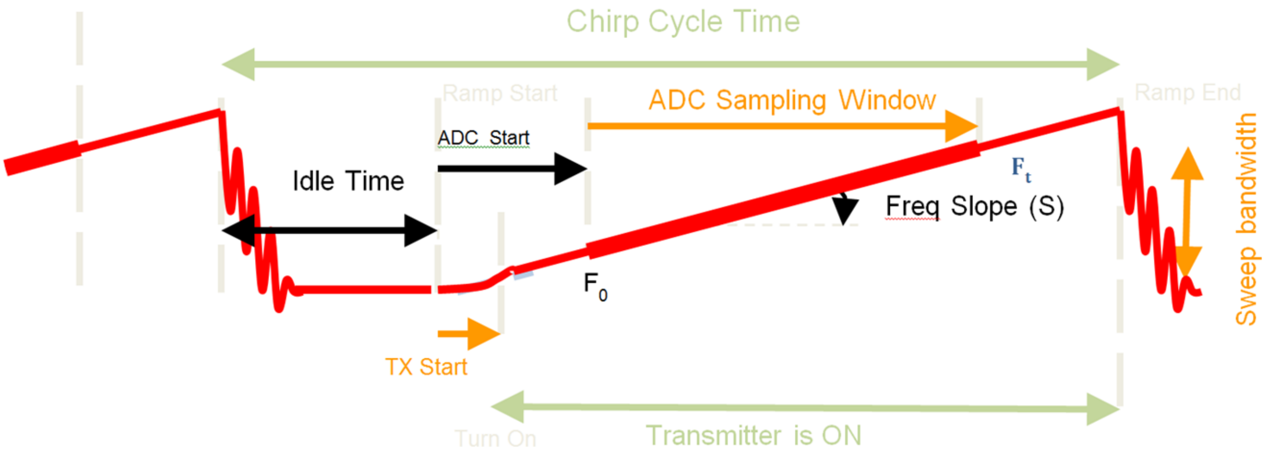


Figure 1. Typical FMCW Chirp

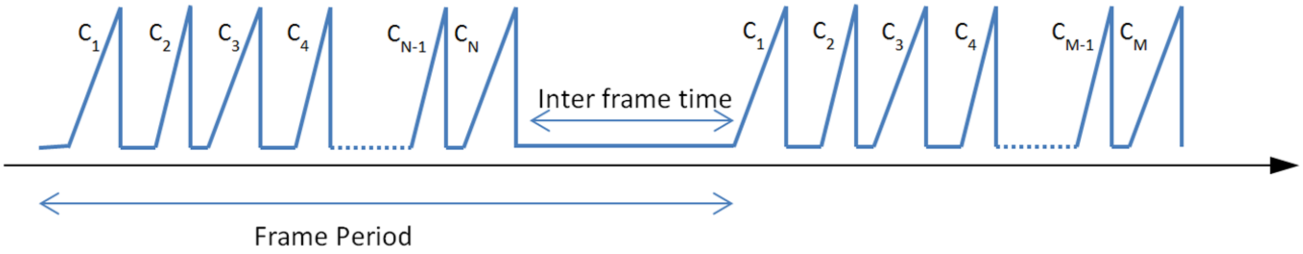


Figure 2. Typical Frame Structure

The following sections list key system performance parameters that are typically considered in any radar application and how the chirp configuration impacts each one of them.

2.1 Measurement Range and Range Resolution

The maximum and minimum distance over which a radar sensor can detect objects is an important parameter for a radar sensor. Also, the range resolution (ability to distinguish two nearby objects) is another important metric.

2.1.1 Maximum Range

In applications like automotive adaptive cruise control (ACC), it's important to be able to view a far off object (>150m). Detecting a far-off object can be limited by either the SNR of the received signal or the IF bandwidth supported by the Radar device.

The max range relationship with the IF bandwidth is shown in Equation 1. TI's AWR2243 radar device provides a large 20-MHz bandwidth, and AWR1243 provides 15-MHz bandwidth, allowing more flexibility in the slope that can be used, which indirectly helps increase the max velocity as will be seen later.

$$\text{Range}_{\text{max}} = \frac{\text{IF}_{\text{max}} \times c}{2 \times S} \tag{1}$$

IF_{max} → Maximum IF bandwidth supported
c → Speed of light
S → Slope of the transmitted chirp

Note that the IF_{max} is also dependent on the ADC sampling frequency ($\text{ADC}_{\text{sampling}}$) used. In a complex 1x sampling mode, the IF bandwidth is limited to 0.9* ($\text{ADC}_{\text{sampling}}$). In case of complex 2x and real sampling modes, the IF bandwidth is limited to 0.9* ($\text{ADC}_{\text{sampling}}$)/2 . The maximum ADC sampling frequency in the TI's radar devices is 45 MHz (AWR22xx) and 37.5 MHz (AWR1xxx).

The other aspect that could limit the max range is the signal to noise ratio (SNR) of signal received by the receiver. This depends on:

- RF performance of the Radar device, like TX output power, RX noise figure, as well as chirp parameters like chirp duration and number of chirps in the frame.
- Antenna parameters like the TX and RX antenna gain in the direction of interest.
- Object characteristics like Radar Cross Section (RCS). RCS is a measure of the amount of energy the object reflects back. This decides how detectable the object is with a radar sensor.
- Minimum SNR required by the detection algorithm to detect an object.

$$\text{Range}_{\text{max}} \text{ based on SNR} = \sqrt[4]{\frac{PtG_{Rx}G_{Tx}c^2\sigma NT_r}{f_C{}^2\,(4\pi)^3 \times kT \times NF \times SNR_{det}}} \tag{2}$$

Pt → Tx output power
G_{Rx} ,G_{Tx} → RX and TX Antenna gain
σ → RCS of the object
N → Number of chirps
T_r → Chirp time
NF → Noise figure of the receiver
SNR_{det} → Minimum SNR required by the algorithm to detect an object
k → Boltzman constant
T_{det} → Ambient temperature

2.1.2 Range Resolution

In many applications it is important to be able to resolve two closely spaced objects as two separate objects, rather than detect them as one. The smallest distance between two objects that allows them to be detected as separate objects is referred to as range resolution. This primarily depends on the chirp sweep bandwidth that the radar sensor can provide. The larger the sweep bandwidth, the better the range resolution. TI's radar devices support a 4 GHz sweep bandwidth that allows a range resolution of as low as approximately 4cm.

$$\text{Range}_{\text{resolution}} = \frac{c}{2 \times B} \tag{3}$$

c → Speed of light
B → Sweep bandwidth of FMCW chirp

Better range resolution also helps in detecting very close by objects, hence, improving a minimum detection range.

2.2 Measurement Velocity and Velocity Resolution

2.2.1 Maximum Velocity

Along with the distance, the relative velocity of the object is another critical parameter of interest. The maximum measurable velocity in Fast FMCW modulated radars depends on the chirp cycle time, that is, the time difference between the start of two consecutive chirps. This in turn depends on how fast the frequency sweep can be performed and the minimum inter-chirp time allowed.

The faster the MMIC can ramp the frequency, the higher the maximum unambiguous velocity. TI's MMIC allows a fast ramp of 100 MHz/µs. Also the closed loop PLL is designed to support a very fast settling of the frequency ramp. Hence, the time taken for the VCO to jump from the end of the ramp frequency to restart the next ramp is very low and allows for a smaller idle time (as low as 2 µsec). For minimum idle time computation, see Section 5.

$$\text{Unambiguous max velocity} = \frac{\lambda}{4T_C} \tag{4}$$

T_c → Total Chirp time ,which inlcudes chirp time+idle time
λ → Wavelength of the signal used

The actual measurable max velocity can be extended beyond the unambiguous max velocity using higher level algorithm.

2.2.2 Velocity Resolution

In applications, like park assist, you might need to separate out objects with small velocity differences, for which good velocity resolution is needed. Velocity resolution mostly depends on the transmit frame duration, that is, increasing the number of chirps in a frame improves the velocity resolution.

$$\text{Unambiguous max velocity} = \frac{\lambda}{2NT_C} \tag{5}$$

N_c → Number of Chirp in a frame

2.3 Angular Range and Resolution

In order to locate the object in the 2D space, the angle of the object is also required along with the distance. In a radar system, the angle is estimated by receiving the reflected signal from the object using multiple receivers that are spaced apart with a distance 'd'. The signal arriving at each of the successive receivers is delayed by d*sin(θ) and this "delay" causes a phase shift of $\frac{2\pi d \sin(\theta)}{\lambda}$. This phase shift between each of the receivers is used to estimate the angle (θ) of the object.

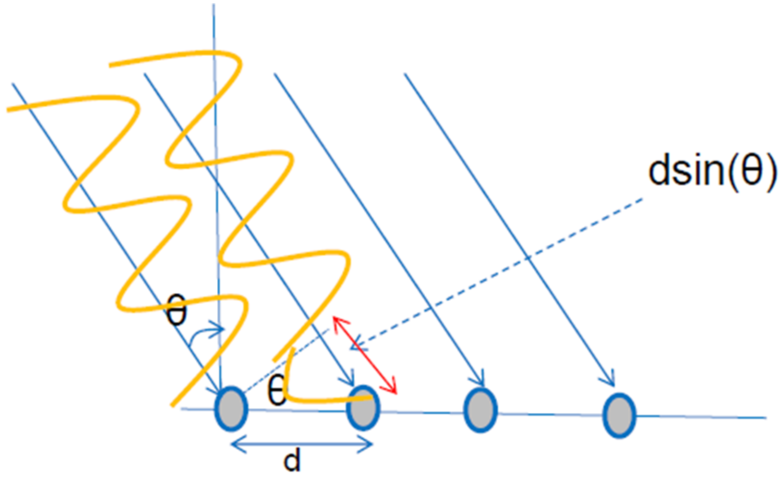


Figure 3. Angle Estimation Basics

The measurable unambiguous angular field of view from the MMIC perspective depends on the spacing between the receivers (d).

$$\text{Max unambiguous angular range} = \sin^{-1}\left(\frac{\lambda}{2d}\right) \tag{6}$$

d → Spacing between receiver antennas
ƛ → Wavelength

So for the widest angular field of view, the spacing of the receiver antenna should be ƛ/2 , theoretically giving ±90° viewing range.

Apart from the antenna spacing, the measurable distance at different angles would also depend on the antenna gain pattern. Typically antennas would have a peak gain at one angle (mostly at 0°, that is, directly facing the front of the antenna) and then the gain would reduce as the angle increases. Figure 4 shows an example antenna pattern where the gain at 90° angle is > 15 dB lower than what it is at 0° angle.

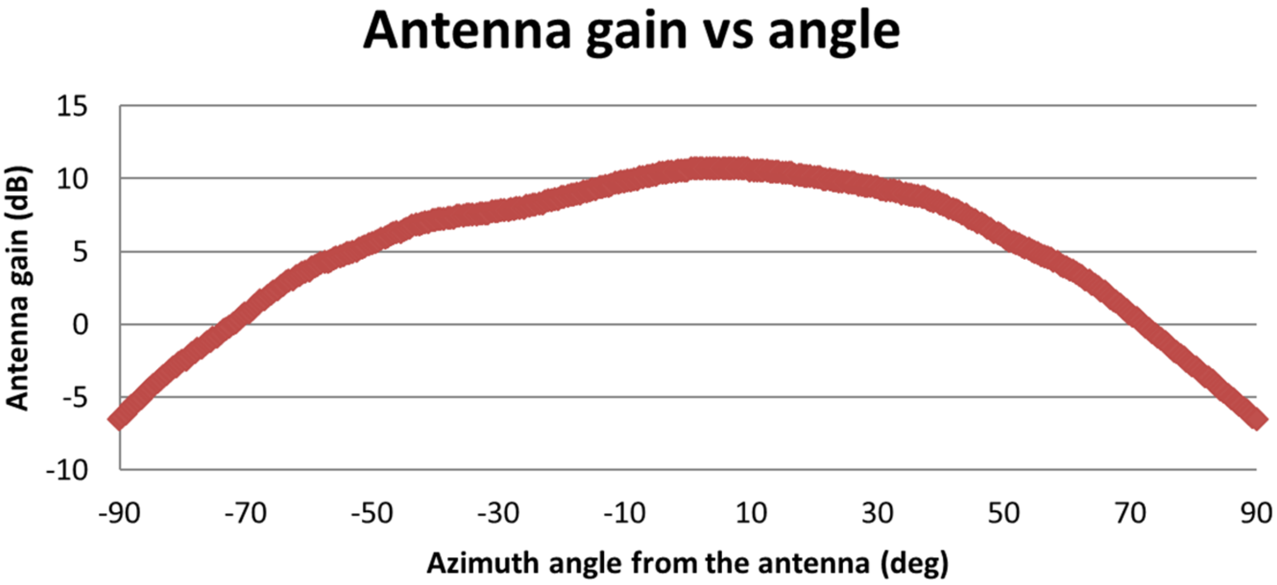


Figure 4. Example Antenna Gain Pattern

2.3.1 Angular Resolution

Apart from the angular field of view, it might also be important to resolve two objects at close by angles, that is, have good angular resolution. For example, in an automotive radar use case, it would be important to detect two cars far off in two different lanes rather than detect them as one single car. In general, the angular resolution measurement depends on the number of receiver antennas available. The larger the number of antennas, the better the resolution.

$$\text{Angular resolution (deg)} = \frac{\lambda}{d \times N_{RX} \cos\theta} \times \frac{180}{\pi} \tag{7}$$

θ_c → Angle of interest, that is, the angle at which the objects are present
N_{Rx} → Number of receiver antennas

The angular resolution can be further improved using multiple transmitters. For more details, see MIMO Radar. If there are multiple transmitters available, then the transmit antennas can be spaced in such a way that each of the transmitters paired with the set of receivers together create a virtual receive array. For example, if there are 3 TX and 4 RX, then a MIMO radar system can produce the equivalent angular resolution of 12 virtual channels.

$$\text{Angular resolution} = \frac{\lambda}{d \times N_{RX} N_{TX} \cos\theta} \times \frac{180}{\pi} \tag{8}$$

N_{Tx} → Number of transmit antennas

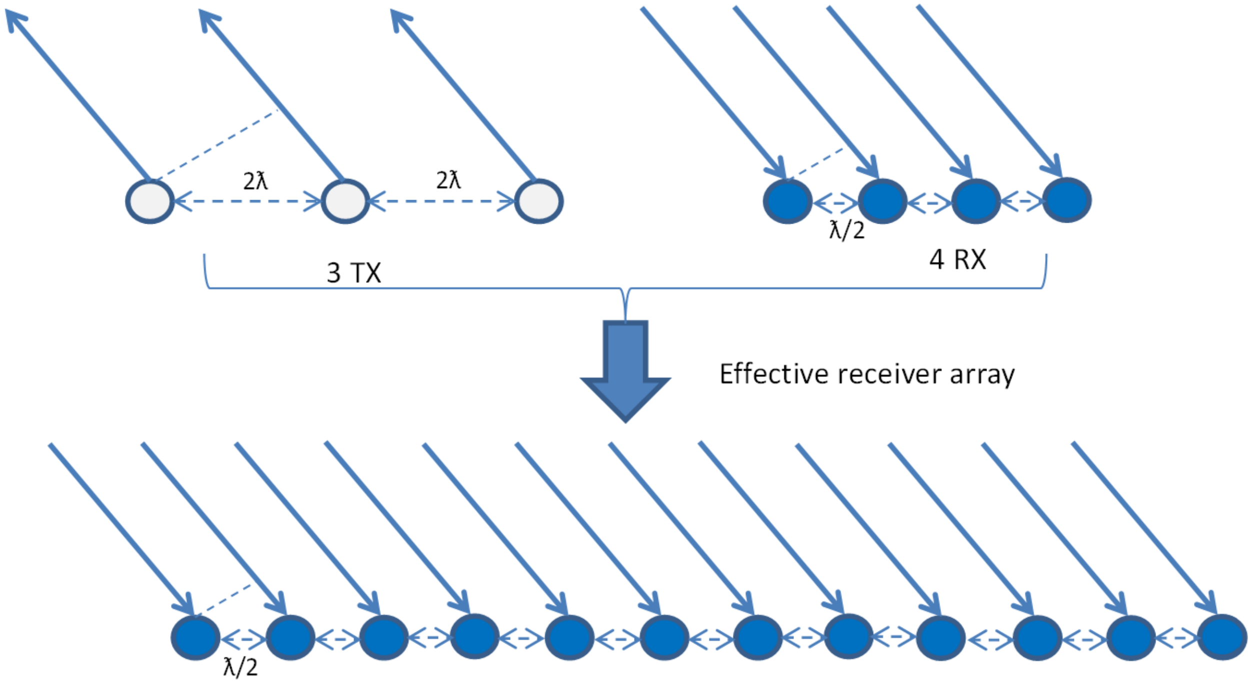


Figure 5. Effective Recevier Channels in MIMO Radar

3 Chirp Configurations for Common Applications

The most common applications for radar in the automotive context include short range radar (typically mounted in the corners) and mid or long-range radar (typically front facing).

This section shows chirp configurations for a 22m USRR, 45m SRR, 125m MRR and a 225m LRR use-case. It is important to note that these are just generic example configurations and it is possible to change the parameters based on the customer's specific system performance requirements. Sample chirp configurations and use-cases as applicable to various TI's mmwave radar devices are shown in the device-specific overview white papers.

Table 1. Example Chirp Configurations for Typical Applications

Parameter	Units	LRR	MRR	SRR	USRR
Max unambiguous range	m	225	125	45	22.5
Sweep bandwidth	MHz	300	540	750	1500
Ramp slope	MHz/us	10	12	15	30
Inter-chirp duration	us	8	10	12	50
Number of chirps	-	256	128	128	128
Range resolution	m	0.50	0.28	0.20	0.1
Chirp duration	us	30	45	50	50
Max umambiguous relative velocity [(1)]	kmph	92.28	63.75	56.56	35.3
Max beat frequency	MHz	15	10	4.5	4.5
ADC sampling rate (complex)	Msps	16.67	11.11	5.00	5.00
Number of samples per chirp		500	500	250	250
Range FFT size	-	512	512	256	256
Frame time (total)	ms	9.728	7.04	7.94	12.8
Frame time (active)	ms	7.68	5.76	6.4	6.4
Radar data memory required	KB	**2048**	**1024**	**512**	**512**

[(1)] The maximum velocity can be increased beyond the max unambiguous velocity by using higher level algorithms.

4 Configurable Chirp RAM and Chirp Profiles

The TI radar devices allow you to control the parameters of chirps in a frame by defining chirp profiles, and variations on top of these profiles through a chirp configuration RAM. Chirp profiles are basic chirp timing templates, useful in defining chirp variants with significant differences in one or more defining parameters (start frequency, slope, idle time, and so forth). The radar devices allow you to program four different chirp profiles. In addition, up to 512 unique chirps can be pre-programmed and stored in the chirp configuration RAM. Each chirp definition entry in the RAM can belong to one of the four profiles, and can optionally differ from their parent profile by small dithers in some of the parameter values. A frame would then consist of a sequence of chirps from a start index to an end index in the chirp configuration RAM which can be looped over up to 255 times.

The parameters that are controllable per profile are:

- Start frequency
- Frequency slope
- Idle time
- ADC start time
- Ramp end time

Each chirp in the chirp configuration RAM can have small dither values that are added to the profile parameters defined in the profile RAM. These are:

- Start frequency variable
- Frequency slope variable
- Idle time variable
- ADC start time variable

The chirp RAM and profile RAMs are depicted in Figure 6.

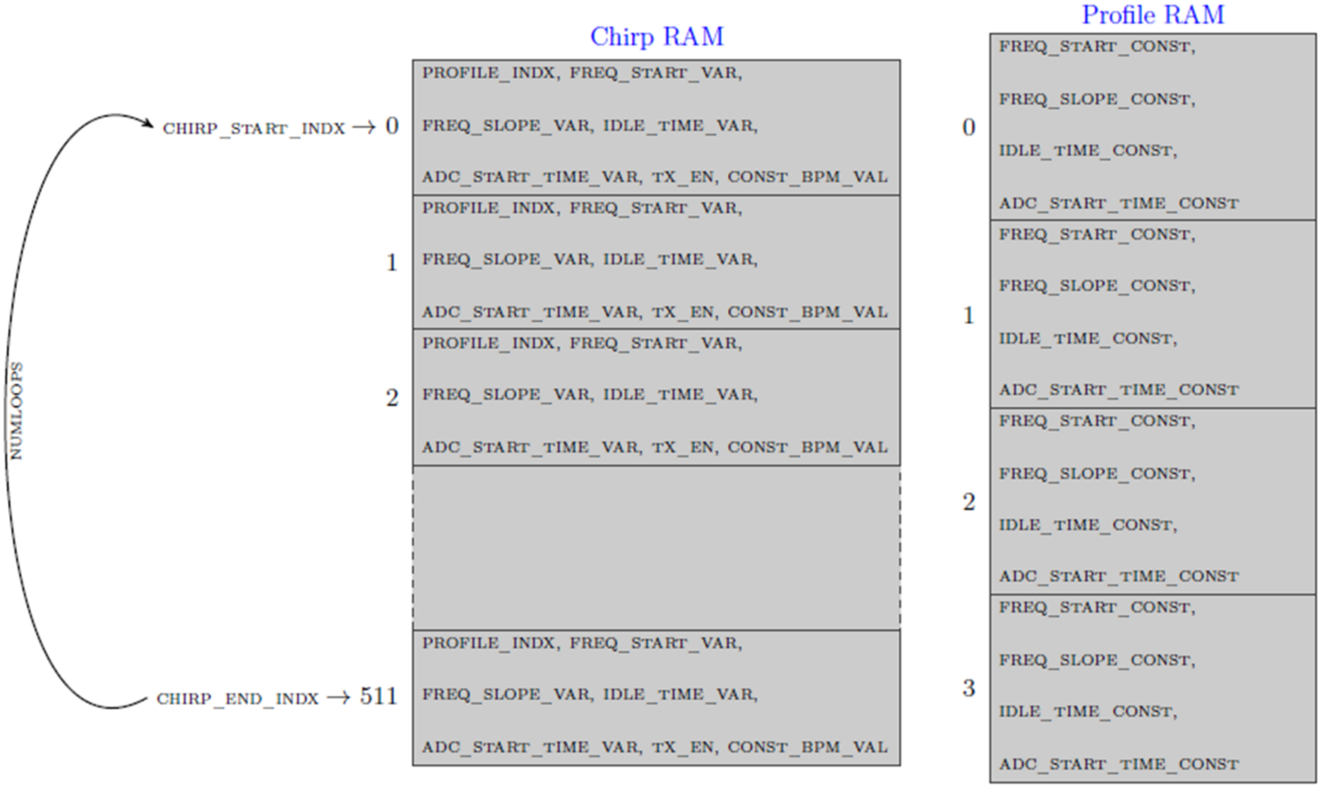


Figure 6. Chirp and Profile RAM Memory Allocation

5 Chirp Timing Parameters

Table 2 provides a list of the parameters that govern the chirp timing. It also shows the relationship of these parameters to form a chirp.

Table 2. Chirp Timing Parameters

Sl. No.	Parameter name	Description
1	Idle time	The time between the end of previous chirp and start of next chirp.
2	TX start time	The time from the start of the ramp at which transmitter is turned on.
3	ADC start time	The time from the start of the ramp when the ADC starts sampling the data.
4	Ramp end time	The time from the start of the ramp until the chirp continues ramping. After this time the synthesizer frequency is reset to the start frequency of the next chirp.

The following sections list the system considerations that need to be taken to determine the value for each of these parameters. The actual numbers for each of these parameters can be derived using a chirp timing parameter configuration calculator utility in Radar Studio application available as part of the DFP package.

5.1 Idle Time

The minimum required idle time is determined predominantly by the synthesizer ramp down settling time, which is a function of ramp down bandwidth. The synthesizer ramp down settling times for some typical bandwidths are tabulated in Table 3.

Table 3. Typical Synthesizer Ramp Down Times for Different Modulation Bandwidths

Ramp Bandwidth	Synthesizer Ramp Down Time (µs)
< 1 GHz	2
> 1 GHz and < 2 GHz	3.5
> 2 GHz and < 3 GHz	5
> 3 GHz	7

Table 3 can directly be used to set the minimum idle time, for cases where the ADC sampling rate is 5 Msps or higher. However, for low sampling rates (< 5 Msps), there is also another constraint that needs to be kept in mind when programming the minimum idle time. This comes from digital pipeline delays in the sigma-delta ADC decimation chain. Due to this, the minimum idle time might need to be slightly higher than the values shown in Table 3. The chirp parameter configuration calculator utility (part of the Radar Studio tool, www.ti.com/tool/mmwave-dfp) takes care of this constraint as well and provides the overall recommended idle time.

5.2 ADC Start Time

The ADC start time does NOT have a hard minimum requirement for correct chip functionality. However, there is a trade-off between the value of this parameter and the quality of “beginning of chirp” signal envelope settling, dependent on the following factors:

- Synthesizer PLL ramp-up settling time, which is a function of ramp slope
- HPF step response settling, which is a function of HPF corner frequencies
- IF/DFE LPF settling time, which is a function of DFE output mode and sampling rate.

The chirp timing parameter configuration calculator utility takes these aspects into account in computing the recommended value.

The user may also have other considerations such as maximum round-trip delay of reflections from the target based on the end use-case that may be added to the recommended ADC start time from the utility. The customer may also choose to use a smaller value than the recommended value, in order to reduce the total inter-chirp gap, by trading off the quality of settling.

5.3 Ramp End Time

The ramp end time is the sum of (a) the ADC start time, (b) the ADC sampling time and (c) the excess ramping time at the end of the ramp. There is no hard requirement on excess ramping time. However, there is again a trade-off between the amount of excess ramping and the “end of chirp” signal envelope settling performance. The primary factor determining the excess ramping required is IF/DFE filtering latency, which is again a function of DFE output mode and sampling rate.

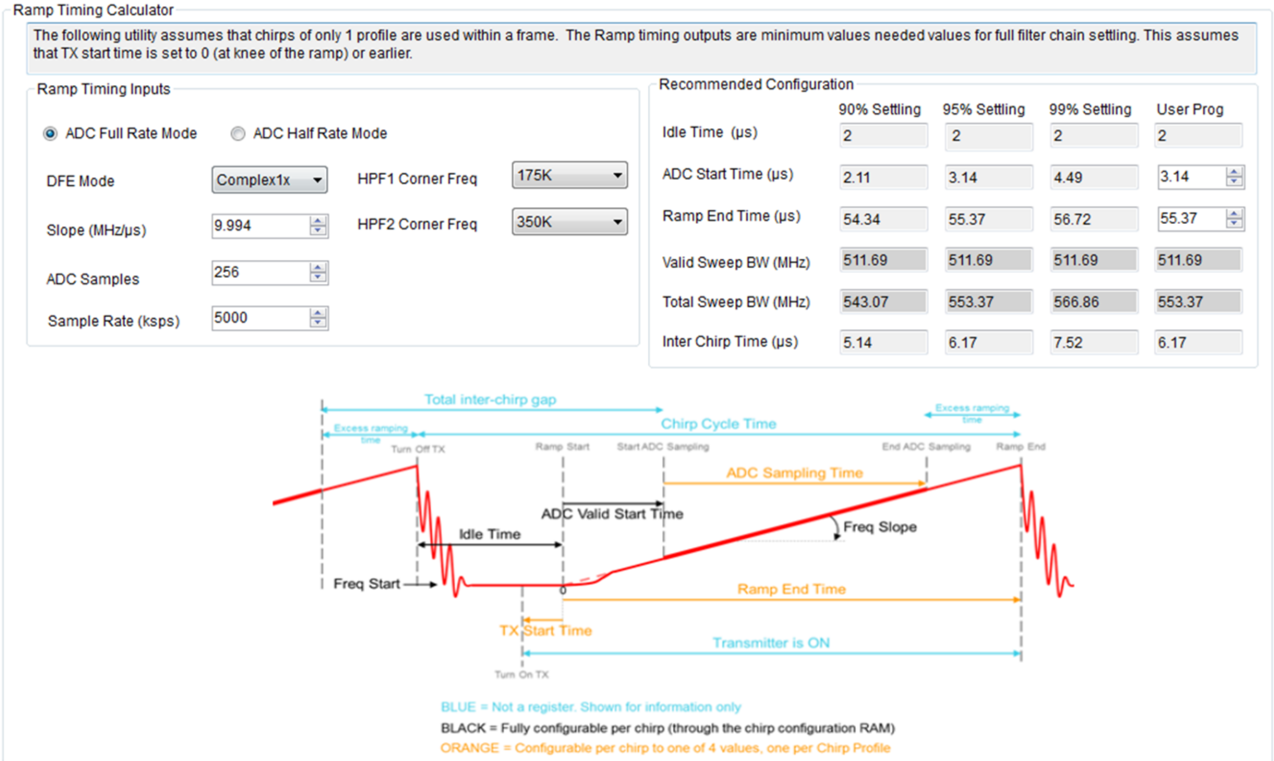
The chirp timing parameter configuration calculator utility takes these aspects into account in arriving at the recommended value for this parameter.

Table 4. Typical IF/DFE Filtering Latencies as a Function of DFE Mode and Output Sampling Rate

DFE Mode	DFE Sampling Rate (Msps)	IF/DFE Filtering Latencies (μs)
Complex1x	16.7	0.3
	10	0.5
	5	1.0
Real/Complex2x	33.3	0.1
	16.7	0.2
	10	0.4
	5	0.8

5.4 Example Timing Calculation Using the Calculator

The Ramp timing calculator available in the mmWave Studio tool (part of the DFP package: (www.ti.com/tool/mmwave-dfp) provides the right timing parameter value recommendations based on the user configurations (slope, ADC samples, sampling rate). These recommendations are based on the minimum requirements from TI's radar device.



Ramp Timing Calculator

The following utility assumes that chirps of only 1 profile are used within a frame. The Ramp timing outputs are minimum values needed values for full filter chain settling. This assumes that TX start time is set to 0 (at knee of the ramp) or earlier.

Ramp Timing Inputs

● ADC Full Rate Mode ○ ADC Half Rate Mode

DFE Mode: Complex1x

HPF1 Corner Freq: 175K

Slope (MHz/μs): 9.994

HPF2 Corner Freq: 350K

ADC Samples: 256

Sample Rate (ksps): 5000

Recommended Configuration

	90% Settling	95% Settling	99% Settling	User Prog
Idle Time (μs)	2	2	2	2
ADC Start Time (μs)	2.11	3.14	4.49	3.14
Ramp End Time (μs)	54.34	55.37	56.72	55.37
Valid Sweep BW (MHz)	511.69	511.69	511.69	511.69
Total Sweep BW (MHz)	543.07	553.37	566.86	553.37
Inter Chirp Time (μs)	5.14	6.17	7.52	6.17

BLUE = Not a register. Shown for information only

BLACK = Fully configurable per chirp (through the chirp configuration RAM)

ORANGE = Configurable per chirp to one of 4 values, one per Chirp Profile

Figure 7. Example Usage of the Ramp Timing Calculator

You may have further considerations over these based on the specific use case. These considerations need to be accounted for separately.

For example, if the raw ADC data is being set over the CSI or LVDS path, the idle time setting needs to make sure the transfer on the CSI/LVDS interface is complete before the next chirp data is available. The time available for this data transfer is = "ramp end time" + "idle time". In case of CSI, the overhead time due to the MIPI protocol also needs to be accounted for.

6 Advanced Chirp Configurations

6.1 Multi-Mode Radar Applications

As you have seen that based on the desired application the chirp configurations need to be set differently. But what if we need to support multiple modes, for example short range and mid-range, simultaneously using a single radar device? The advanced frame configuration available in TI's radar allows for large flexibility to have multiple chirp configurations in a single frame. The frame can be constituted using a sequence of "sub-frames" with each of these sub-frames representing a different radar mode. The rlSetAdvFrameConfig API helps enable this kind of configuration. For details, see the "mmWave Radar Interface Control Document" in the DFP package: www.ti.com/tool/mmwave-dfp).

To provide maximum flexibility of the chirps within a frame the advanced frame config API provides the ability to break a frame into different sub frames (up to 4). Each of the sub frames consists of multiple bursts of chirps (up to 512 bursts) . Each burst can consist of 512 unique chirps which are associated to one of the 4 profiles, and the start chirp index can be programmed to have a fixed offset from the previous burst. A set of bursts in a sub-frame can be further looped in software up to 64 times. The below figure is an example how a sub frame is formed.

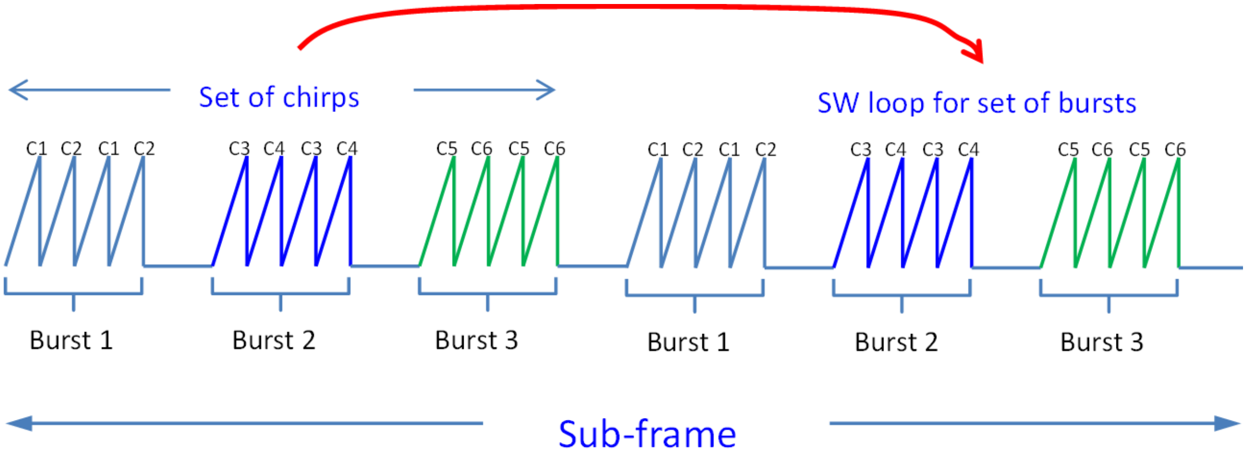


Figure 8. Sub Frame Structure Showing Three Bursts Looped Twice (start index having an offset of 2)

The frame is formed by a maximum of four such sub frames, and each of the sub frames can have a different set of chirps. The different chirps can also use different transmitters (which could possibly have different antenna configurations). Figure 9 shows an example of the different chirp profiles and sub frames that can form a frame.

The timing requirements for the advanced frame is as follows: Inter-burst time should be ≥ 50 µsec, inter sub-frame time should be ≥ 100 µsec and inter frame time should be ≥ 200 µsec.

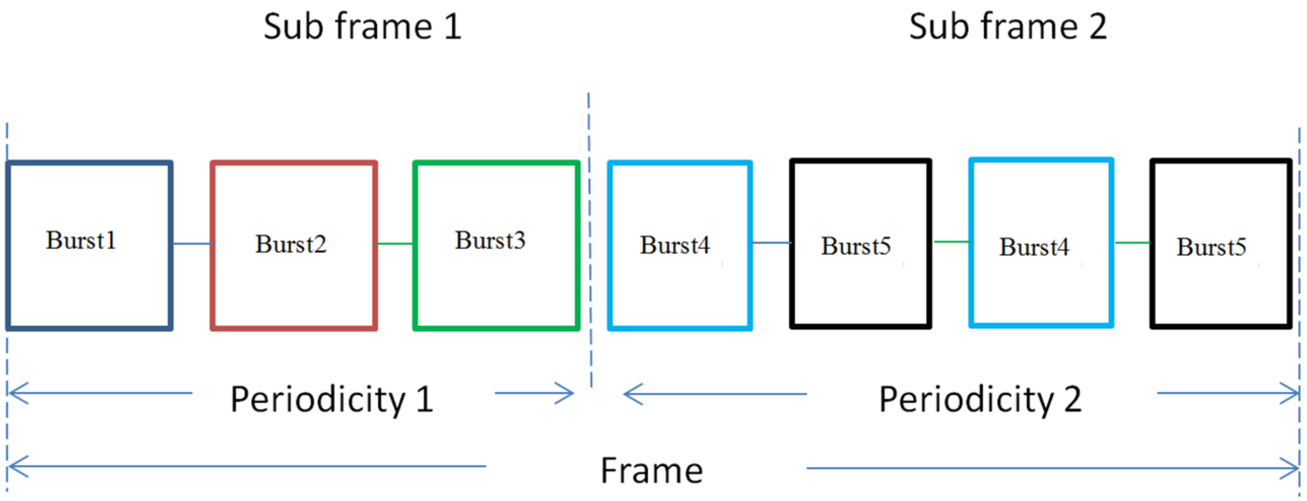


Figure 9. Example of Advanced Frame Configuration of Two Sub Frames

6.2 *Individual Transmitter Binary Phase Modulation (BPM)*

TI’s radar device allows each of the transmitter outputs to be modulated by a phase of 0° or 180°. This allows usage of multiple transmitters simultaneously by using mutually uncorrelated binary codes. hence, improving the effective SNR.

The phase configuration is done by the “rlSetBpmChirpConfig” API. For details on BPM and MIMO radar, see MIMO Radar .

7 Basic Chirp Configuration Programming Sequence

TI’s mmwave link APIs provide simple interface to the MMIC and offer full flexibility to configure all the chirp parameters based on the application and requirement. The details of each of the APIs and the parameter descriptions are available in the “mmWave Radar Interface Control Document: www.ti.com/tool/mmwave-dfp). Figure 10 shows a typical sequence to be followed to configure the chirp and frame.

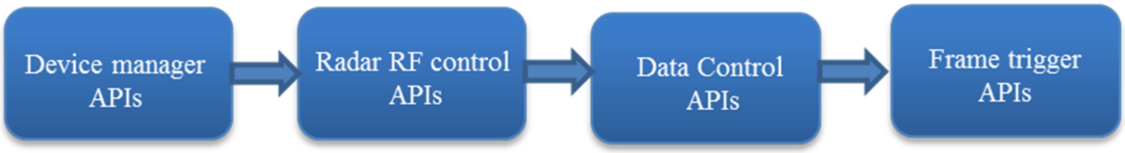


Figure 10. Radar Configuration Sequence

7.1 Device Manager APIs

These APIs are used to power on and initialize the sensor:

- *rlDevicePowerOn*

  This function initializes the driver and does the necessary resource allocation for the driver. It initializes the host protocol driver by creating the necessary OS services like semaphore, mutex, queues and so forth. It also brings radar devices (multiple devices in case of cascade) out of reset and opens the communication channel (SPI, Mailbox, and so forth) with these devices.

- *DeviceFileDownload*

  This function downloads a binary file from the host to the internal RAM of the radar device. This file could be a firmware patch file, application code, calibration data or configuration data.

- *rlDeviceRfStart*

  This function initializes the RF (BIST) subsystem in the radar device. The function returns immediately and RF initialization completion is indicated by the asynchronous event (RL_EVENT_AR_DEVICE_START_COMPLETE). User application should wait for this event before invoking any Radar Sensor Control APIs.

7.2 Radar RF Control APIs

These APIs are used to configure the RF parameters, chirp profiles and frame configurations.

- *rlSetChannelConfig*

  This API allows configuring of the number of TX antennas (out of 3) and number of RX antennas (out of 4) to be used. It also allows the selection of whether they are using the sensor in standalone mode or cascade mode.

- *rlSetAdcOutConfig*

  This API allows configuring of the number of bits per sample (12/14/16). It also allows the choice of whether the ADC data should be real only, Complex -1x or Complex- 2x.

- *rlSetLowPowerModeConfig*

  This API allows the setting of a low power ADC mode for power saving. In this mode, the maximum ADC sampling rate is limited.

- *rlSetProfileConfig*

  This API allows configuring the chirp “profile”, which defines a template for a chirp. These configurations include chirp start frequency, idle time, ADC start time, chirp slope, chirp duration, TX power level in the chirp, number of ADC samples per chirp, the ADC sampling rate, High pass filter (HPF) cutoff frequencies and RX gain setting. Up to four profiles can be defined and in a particular frame (which can have up to 512 unique chirps) the chirps in the frame can belong to any of these four profiles. Since a single profile may not suffice all the intended use cases of the sensor, having flexibility to have multiple profiles allows usage of a single sensor in multiple scenarios/use cases.

- *rlSetChirpConfig*

  Once the profiles are defined, each of the unique chirps can be associated to one of these profiles. Apart from this, the API also allows limited chirp to chirp variations (beyond the profiles) in some of the important parameters like start frequency, idle time and ADC start time. The API also allows selecting the transmitters to be used in that particular chirp.

- *rlSetFrameConfig*

This API allows selecting the sequence of the chirps that form the frame, number of frames that need to be transferred and the periodicity of the frames. The periodicity would define the inter-frame time (periodicity – chirptime), hence, the duty cycle of the transmission. The frame could be software API triggered (via rlSensorStart API) or externally triggered using the SYNC_IN signal. In case of the HW trigger option, a programmable delay can be set from the SYNC_IN edge.

7.3 Radar Data Control APIs

The ADC data captured during the chirps is transferred out of the device on the high speed debug interface or CSI. The data control APIs allow configuring the ADC data that needs to be transferred out and the high speed interface (LVDS/CSI) configurations. Along with the ADC data, some additional information related to the quality of the chirp and chirp parameters (that are referred to as CQ (chirp quality) and CP (chirp parameter), respectively) can also be transferred out. The LVDS/ CSI configurations, lane configurations, and so forth can be done using these APIs. For details on these APIs and the parameters, see the “mmWave Radar Interface Control Document” (part of the DFP package: www.ti.com/tool/mmwave-dfp).

7.4 Frame Trigger API

Once the chirps and frames are configured, they can be triggered either via a software API or hardware triggered using the digital SYNC_IN signal. The software API to trigger the frame is rlSensorStart.

8 References

MIMO Radar

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Original (May 2017) to A Revision	Page
• Updated Maximum Range section.	3

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265