

Design Guide: TIDEP-01001

Vehicle Occupant Detection Reference Design



Description

This reference design demonstrates the use of the AWR6843 60GHz single-chip mmWave sensor with integrated DSP, as a Vehicle Occupant Detection (VOD) and Child Presence Detection (CPD) Sensor enabling the detection of life forms in a vehicle. This design provides a reference processing chain which runs on the C674x DSP, enabling the generation of a heat map to detect life forms in a Field of View (FOV) of ± 60 degrees.

Resources

TIDEP-01001	Design Folder
AWR6843	Product Folder
AWR6843ISK	Tool Folder
mmWaveSDK	Software Development Kit
VOD Application Software	Software Demonstration Source Code

Features

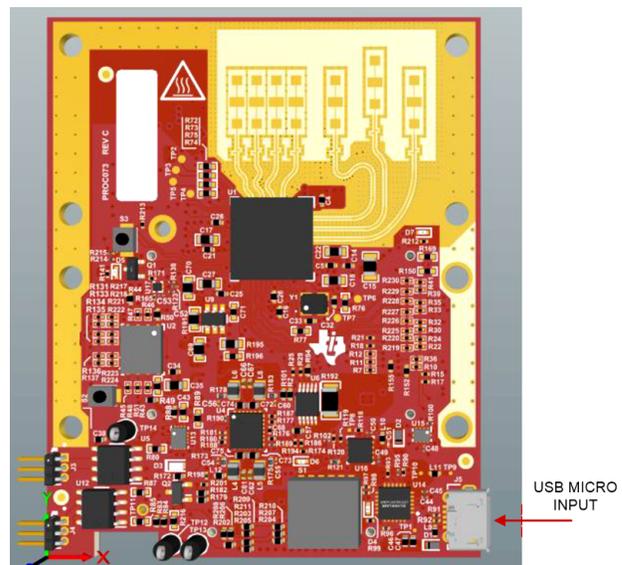
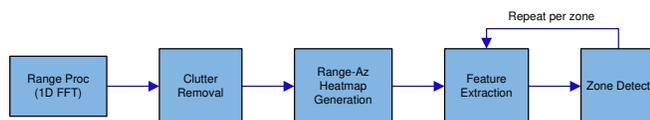
- Demonstration of mmWave sensor technology for robust detection of life forms (adults, children, pets) in a vehicle.
- Generate presence heat map with an FOV of ± 60 degrees.
- Source code for processing and detection based on the mmWave software development kit (SDK)
- Based on proven EVM hardware designs, enabling quick time to market and out-of-the-box demonstration.
- Radar front-end and detection configuration fully explained

Applications

- Child presence detection
- Vehicle occupant detection
- Intruder detection



Ask our TI E2E™ support experts



An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

1 System Description

Today's vehicles require robust and reliable information about the in-cabin occupancy. Smart airbag deployment systems, air condition controls, detecting children and disabled people left behind in vehicles relies upon this information.

The TIDEP-01001 provides a reference for creating a vehicle occupant detection application, using TI's AWR6843 sensor based on 60-GHz mmWave radio-frequency (or AWR1843 sensor based on 77-GHz mmWave radio-frequency) complementary metal-oxide semiconductor (RF-CMOS) technology.

TI's mmWave sensing devices integrate a 60-GHz to 64-GHz (or 76-GHz to 81-GHz) mmWave radar front end with ARM® microcontroller (MCU) and TI DSP core for single-chip systems.

This reference design demonstrates the suitability of the AWR6843/1843 for vehicle occupant detection applications. The design targets the implementation of a wide azimuth field of view ($\pm 60^\circ$) and close range (3 m) sensor configuration, which can detect life forms across multiple regions (zones) of interest, and can localize occupant signature within a region.

This reference design implements algorithms for generating an azimuth-range heat map, detection, and decision for an AWR6843/1843 device on a TI EVM module.

The design provides a list of required hardware, schematics, and foundational software to quickly begin vehicle occupancy detection product development. It describes the example usage case as well as the design principle, implementation details, and engineering tradeoffs made in the development of this application. High-level instructions for replicating the design are provided.

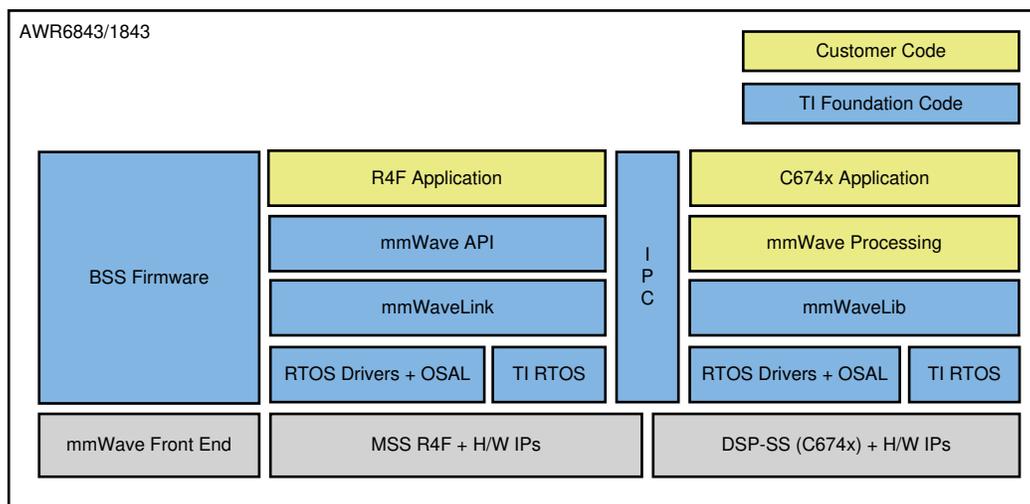
2 System Overview

The VOD reference design is built around the AWR6843/1843 EVM evaluation boards and the millimeter wave (mmWave) software development kit (SDK). The system is optimized and built for VOD/CPD applications to detect objects within a 3-m range.

2.1 Block Diagram

The mmWave software development kit (SDK) enables the development of mmWave sensor applications using the AWR6843/1843 SOC and EVM. The SDK provides foundational components that let end users focus on their applications. In addition, the SDK provides several demonstration applications, which serve as a guide for integrating the SDK into end-user mmWave applications. This reference design is a separate package installed in addition to the SDK package, and is the "application" portions of the block diagram.

Figure 1. Software Block Diagram



2.2 Vehicle Occupancy Detection Software Block Diagram

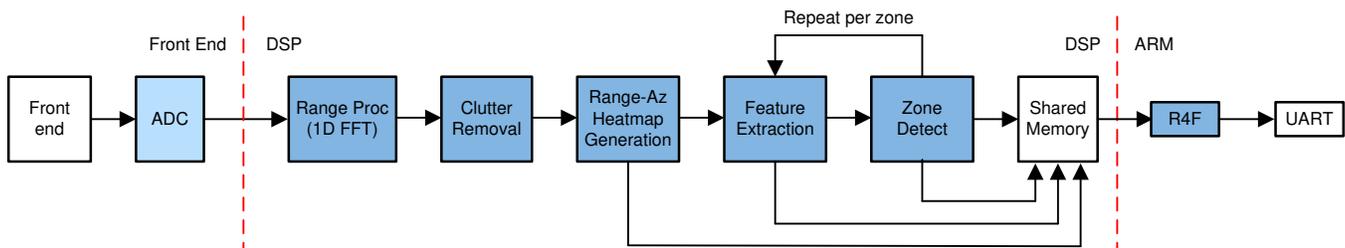
As described in [Figure 2](#), the implementation of the vehicle occupancy detection example in the signal-processing chain consists of the following blocks, implemented as DSP code executing on the C674x core in the AWR6843/1843:

- Range processing
 - For each antenna, 1D windowing, and 1D fast Fourier transform (FFT).
 - Range processing is interleaved with the active chirp time of the frame.
- Clutter removal
 - Estimate the DC component for each range bin, across chirps in a frame.
 - Subtract the estimated DC component for each range bin.
- Range-Azimuth heatmap generation
 - Perform Direction-of-Arrival (DOA) Spectral Estimation to calculate a 2D heatmap for the frame, indexed by range (N_r , rows) and azimuth (N_{az} , columns).
- Heatmap conditioning
 - Arc removal: For each range row, examine the power compared to normal empty FOV. When there is a large amount of MVDR leakage causing elevated power across the entire row (for example, an arc), use image processing techniques to drop the arc to the noise floor while maintaining any peak at its original strength.
 - Smoothing: On occasion, frame to frame clutter removal variations can cause a single frame drop in occupant signature strength, which in turn can affect decision quality. By using a smoothing function that compares the heatmap to the previous frame's heatmap, these drops can be

smoothed out and a more consistent signature is maintained.

- Feature Extraction
 - Each frame, scan the heatmap within each defined zone of interest and compute a feature vector:
 - Zone power. Originally, this was a simple average across the entire zone. But large zones suffer because the power of an occupant signature gets diluted due to the number of noise level cells in the zone. Now, the zone is searched for the maximum 5x5 region, and the average of the 5x5 is used for zone power.
 - Frame-averaged zone power (for L (window length), frames).
 - Frame-averaged power ratio (for L frames).
 - Correlation coefficient of zone power.
- Zone Detection
 - For N_z zones, there are 2^{N_z} possible occupancy states. Offline, define a matrix of decision parameters (coefficients) to represent targets of interest (adults, children, pets, and so on).
 - Perform matrix multiplication with the decision parameters and feature vector. This yields an array of flags, one flag per zone, a 1 indicating *zone occupied*, and 0 indicating *empty*.

Figure 2. Application Software Block Diagram

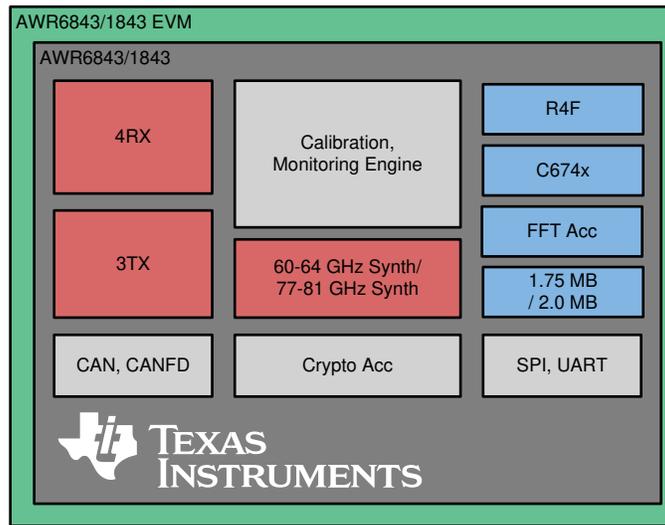


After the DSP finishes frame processing, the results are formatted and written in shared memory (HSRAM) for the R4F MCU to send to the host through a UART for visualization.

2.3 Highlighted Products

2.3.1 AWR6843/1843 Single-Chip Radar Solution

The AWR6843/1843 devices are integrated single-chip FMCW sensors capable of operation in the 60-GHz to 64-GHz and 76-GHz to 81-GHz bands. They are built with TI's low power 45-nm RFCMOS process, and enables unprecedented levels of integration in an extremely small form factor. They are an ideal solution for low power, self-monitored, ultra-accurate radar systems in the automotive and industrial space.

Figure 3. AWR6843/1843BOOST EVM Block Diagrams


The AWR6843/1843 EVMs have the following features:

- AWR6843 or AWR1843 mmWave radar device.
- Power management circuit to provide all the required supply rails from a single 5-V input.
- Three onboard TX antennas and four RX antennas.
- Onboard XDS110, which provides JTAG interface, UART1 for loading the radar configuration to the device, and UART2 to send the object data back to the PC.

For more details on the hardware, see the [AWR6843 Evaluation Module \(AWR6843ISK\) Single-Chip mmWave Sensing Solution](#). The schematics and design database can be found in the following documents:

- [AWR6843 Evaluation Board Design Database](#)
- [AWR6843ISK Schematic, Assembly, and BOM](#)

2.3.2 mmWaveSDK

The mmWave SDK is split in two broad components: the mmWave Suite and the mmWave demo. The mmWave Suite is the foundational software part of the mmWave SDK and includes smaller components:

- Drivers
- OSAL
- mmWaveLink (BSS interface API)
- mmWaveLib (C674 optimized library)
- mmWave API (High level control API)
- BSS firmware
- Board setup and flash utilities

The mmWave SDK provides a demonstration program that depicts the various control and data processing aspects of a typical mmWave radar application. Data visualization of the demonstration's output on a PC are provided via online visualization tools.

Note that SDK versions 3.x and later should be used, as earlier versions do not support AWR6843 or AWR1843.

2.3.3 mmWave Toolboxes

Also provided are two mmWave "toolboxes" that contain a wide variety of example labs and demonstration programs. The toolboxes may be accessed in the TI Resource Explorer, available in both Code Composer Studio (CCS) and via a browser [here](#).

2.4 System Design Theory

2.4.1 Use Case Geometry and Sensor Considerations

The AWR6843/1843 are radar-based sensors that integrate a fast FMCW radar front end with both an integrated ARM R4F MCU and the TI C674x DSP for advanced signal processing.

The configuration of the radar front end depends on the configuration of the transmit signal and the configuration and performance of the RF transceiver, the design of the antenna array, and the available memory and processing power. This configuration influences key performance parameters of the system.

When designing the frame and chirp configuration for a vehicle occupancy detection use case, start by considering increasing range resolution and velocity resolution over maximum range and velocity; because objects are within short range, defined zones will be relatively stationary.

Figure 4. Example Car Configurations

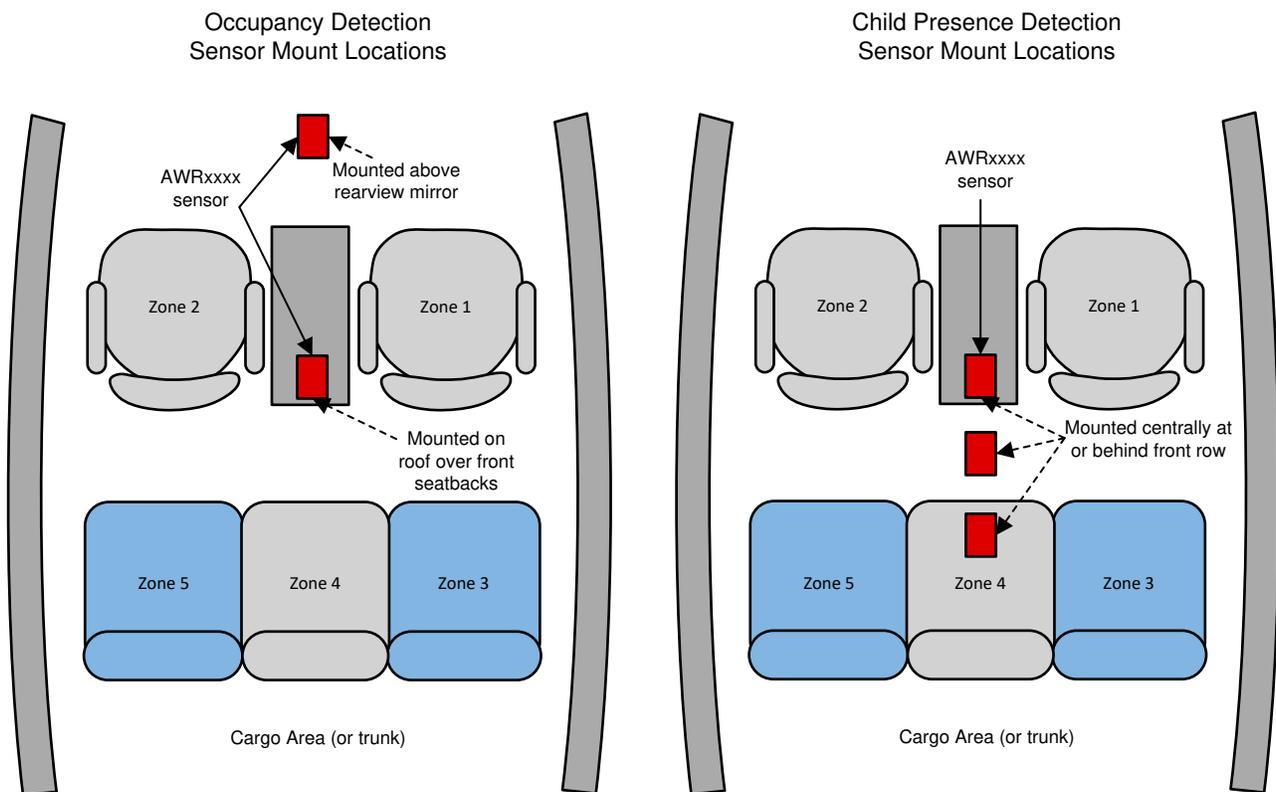


Figure 4 shows the zone locations and sensor mounting positions for the example configuration use cases. This design makes use of the two azimuth antennas on the device, and currently does not use the elevation antenna (and so does not generate elevation data). As such, our primary data structure for decision making is the range-azimuth heatmap. For VOD applications, a forward mounted antenna is preferable so that good coverage of the two seat rows are obtained. Sensor angle is not so important, but is set to optimize line of sight and centering of zones as much as possible. Typically, sensor angle is 10 to 15 degrees from vertical.

For CPD, it is important to have good coverage of the second row footwell, so a more overhead mounting position is desirable. However, depending on the vehicle's interior geometries, this may limit view of the front row seats. Also, it is usually found that a mounting angle parallel to the floor causes undesirable reflections. So a small tilt towards the rear of the vehicle is recommended. This tilt is typically 10 to 15 degrees from horizontal.

2.4.2 Chirp Configuration and System Performance

An example configuration for VOD/CPD applications is given in [Table 1](#).

Table 1. Example Configuration

PARAMETER	SPECIFICATION
Idle time (μs)	250
ADC start time (μs)	10
Ramp end time (μs)	40
Num ADC samples per chirp	64
Freq slope (MHz/μs)	98
Starting frequency (GHz)	60 or 77
ADC sampling freq (ksps)	2200
Num chirps per frame	512 (128 × 4)
MIMO (1 => Yes)	1
Chirp cycle time (μs)	340
Bandwidth (MHz)	3920
Frame periodicity (ms)	160
Radar cube requirements (KB)	512

The configurations in [Table 1](#) are just illustrative, and can be tailored according to user requirements.

2.4.3 Configuration Profile

The VOD demo application lets the user push the radar configuration using a “Profile Configuration” file over the CLI (or User) UART to the mmWave device. The mmWave SDK user guide (included in the mmWave SDK distribution) describes the semantics of the general chirp and profile commands in detail. These are listed at the top of all VOD demo configuration files. Following the chirp and profile commands, the VOD demo specific commands are listed. The VOD demo user guide describes these commands in detail. The command 'sensorStart' is normally the last command sent, since this command causes chirp and frame processing to begin.

Figure 5. VOD Profile Configuration File

```

1  sensorStop
2  flushCfg
3  dfeDataOutputMode 1
4  channelCfg 15 5 0
5  adcCfg 2 1
6  adcbufCfg -1 0 0 1 1
7  profileCfg 0 60 250 10 40 4 0 98 1 64 2200 0 0 40
8  chirpCfg 0 0 0 0 0 0 1
9  chirpCfg 1 1 0 0 0 0 4
10 chirpCfg 2 2 0 0 0 0 1
11 chirpCfg 3 3 0 0 0 0 4
12 frameCfg 0 3 128 0 160 1 0
13 lowPower 0 1
14 %heatmap size MUST be 32 for data collection mode
15 guiMonitor 0 1 16
16
17 %VOD Demo commands:
18
19 %please refer to VOD demo chirp commands
20
21 sensorStart

```

The profile configuration (profileCfg), defines the profile of a single chirp (as shown in [Figure 5](#)). Subsequently, four chirp configurations are defined; each one inheriting the same profile but associated with TX1 and TX2 alternately. Finally, a frame config message constructs a frame with transmissions alternating between TX1 and TX2.

2.4.4 Processing Chain

An example processing chain for vehicle occupancy detection using a short range chirp and frame design is implemented on the mmWave device.

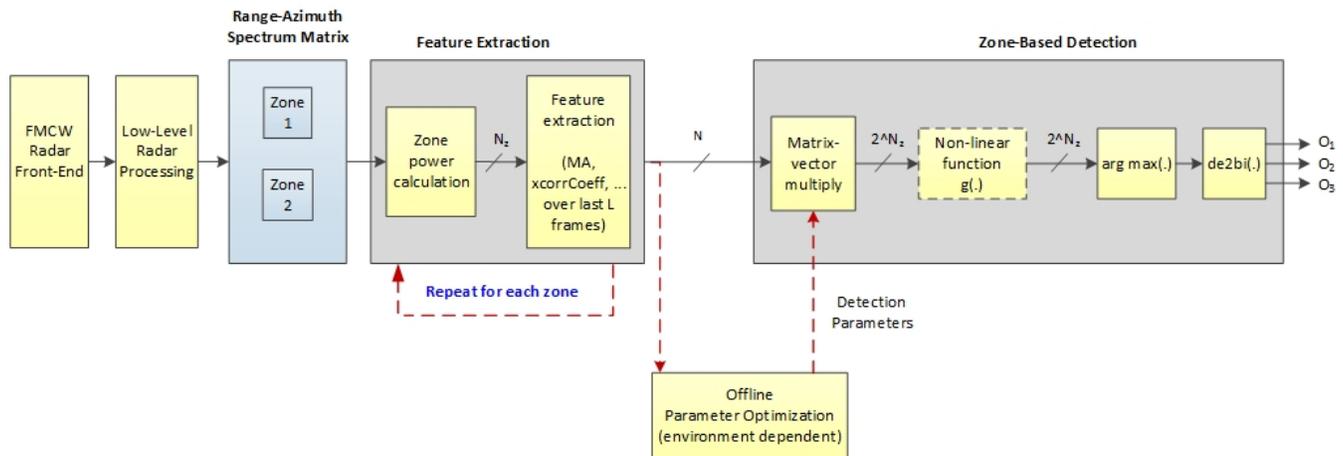
The main processing elements involved in the processing chain consist of the following:

- Front end – Represents the antennas and the analog RF transceiver implementing the FMCW transmitter and receiver and various hardware-based signal conditioning operations. This must be properly configured for the chirp and frame settings of the usage case.
- ADC – The ADC is the main element that interfaces to the DSP chain. The ADC output samples are buffered in ADC output buffers for access by the digital part of the processing chain.
- EDMA controller – This is a user-programmed DMA engine employed to move data from one memory location to another without using another processor. The EDMA can be programmed to trigger automatically, and can also be configured to reorder some of the data during the movement operations.
- C674 DSP – This is the digital signal processing core that implements the configuration of the front end and executes the main signal processing operations on the data. This core has access to several memory resources, as noted further in the design description.
- ARM R4F – This ARM MCU can execute application code, including further signal processing operations and other higher level functions. In this application, the ARM R4F primarily relays visualization data to the UART interface. There is a shared memory visible to both the DSP and the R4F.

The processing chain is implemented on the DSP. There are several physical memory resources used in the processing chain, which are described in [Table 2](#).

Table 2. DSP Memory Resources

Section Name	Size (KB) as Configured	Memory Used (KB)	Description
L1D SRAM	16	16	Layer one data static RAM is the fastest data access for DSP and is used for most time-critical DSP processing data that can fit in this section.
L1D cache	16	Used as cache	Layer one data cache caches data accesses to any other section configured as cacheable. The LL2, L3, and HSRAM are configured as cacheable.
L1P SRAM	16	4	Layer one program static RAM is the fastest program access RAM for DSP and is used for most time-critical DSP program that can fit in this section.
L1P cache	16	Used as cache	Layer one cache caches program accesses to any other section configured as cacheable. The LL2, L3, and HSRAM are configured as cacheable.
LL2	256	240	Local layer two memory is lower latency than layer three for accessing and is visible only from the DSP. This memory is used for most of the program and data for the signal processing chain.
L3	640	595	Higher latency memory for DSP accesses primarily stores the radar cube and the range-Doppler power map. It is a less time sensitive program. Data can also be stored here.
HSRAM	32	32	Shared memory buffer between the DSP and the R4F relays visualization data to the R4F for output over the UART in this design.

Figure 6. Processing Chain Flow


As shown in [Figure 6](#), the implementation of the VOD and CPD use cases in the signal-processing chain consists of the following blocks, implemented as DSP code executing on the C674x core in the mmWave device:

- **Range processing** — For each antenna, EDMA is used to move samples from the ADC output buffer to DSP's local memory. A 16-bit, fixed-point 1D windowing and 16-bit, fixed-point 1D FFT are performed. EDMA is used again to move output from DSP local memory to the radar cube storage in layer three (L3) memory. Range processing is interleaved with active chirp time of the frame. All other processing happens each frame, except where noted, during the idle time between the active chirp time and the end of the frame (the "inter-frame" period).
- **Clutter removal** — In clutter-rich environments, especially indoor, detecting objects with small RCS (radar cross section), such as pedestrian and life-form objects, is a challenge. Some clutters with large RCS, such as building structure and furniture, can dominate the received signal. As a result, objects with small RCS can be buried under the strong interference from the clutters and become difficult to detect. When the radar sensor is stationary, clutters and all stationary objects have zero Doppler, which is exploited to improve the signal condition for the remaining radar signal processing chain.
- **Range-Azimuth heatmap generation** — The Range-Azimuth heatmap is generated using high resolution direction of arrival (DoA) spectral estimation, based on spatial covariance. This is done each frame using the clutter removed, 1D FFT outputs. To avoid numeric instability issues, a small value is added along the diagonal of the covariance estimate matrix. The "heatmap" is actually a velocity map, with larger magnitude values showing more movement of non-static objects.
- **Range-Azimuth heatmap conditioning** — To help overcome issues with clutter removal and DoA, the heatmap is further processed to suppress MVDR arcs, and and to smooth single-frame drops caused by frame variations in clutter removal.
- **Feature extraction** — Using the frame's Range-Azimuth heatmap (and several previous frame's heatmaps), several features are calculated using zone definitions pre-computed and loaded during configuration. The features are computed using only the heatmap cells within the defined zones. These features are average power, average power for the past N frames, power ratios, and the correlation coefficient.
- **Zone detection** — For N_z zones, there 2^{N_z} are possible occupancy states for the area of interest. Matrix multiplications are performed with the feature vectors and offline-generated training coefficients. The result of these multiplications is an array of decisions, 1 (occupied) or 0 (empty) for each zone, representing the possible occupancy states.

After DSP finishes frame processing, the results are formatted and written in shared memory for the MSS R4F to send to host using UART for visualization. The PC GUI executable then decodes each frame's data and updates the heatmap and zone detections accordingly.

2.4.5 Heatmap Generation Algorithms

This section explains a functional block for high-resolution DoA spectral estimation based on spatial covariance. The spatial covariance is estimated within a radar frame. Although an MVDR (minimum variance distortion less response) based DoA estimation approach is presented, other covariance-based high-resolution DoA methods can be used as well using the same radar processing signal chain.

Notation:

$X_{n,k,p}$: the output of range processing and clutter removal.

n, k , and p are the range bin index, the chirp index, and the virtual receive antenna index, respectively.

The following describes a DoA spectral estimation:

For the n -th range bin:

1. Spatial covariance is estimated as follows:

$$\hat{R}_n = \frac{1}{N_c} \sum_{k=1}^{N_c} X_{n,k} X_{n,k}^H$$

where

- $x_{n,k}$ is called a dimensional spatial vector for n -th range bin and the k -th chirp, which is formed from the data cube by stacking samples across the virtual antennas (1)

For example:

$$x_{n,k} = [X_{n,k,1}, X_{n,k,2}, \dots, X_{n,k,N_a}]^T \quad (2)$$

To reduce the amount of computation, a smaller number of chirps less than N_c may be used in estimating the spatial covariance.

DoA spectral estimation is performed as follows: An MVDR (also known as Capons' beamforming) based approach is shown here. First, denote $a(\theta)$ a steering vector for an azimuth angle θ as follows: for a linear one-dimensional virtual receiver antenna array,

$$a(\theta) = [1, \exp(j2\pi d \sin(\theta)), \dots, \exp(j2\pi(N_a - 1)d \sin(\theta))]^T$$

where

- d is the inter-antenna spacing normalized by the wavelength. (3)

The steering vector represents phase difference on each of virtual receive antennas when the incident radar radio-frequency (RF) signal comes from azimuth angle θ . The angular spectrum for the n -th range bin is given by:

$$P_n(\theta) = \frac{1}{a(\theta)^H \hat{R}_n^{-1} a(\theta)} \quad (4)$$

And the corresponding beamforming vector is given as:

$$w_n(\theta) = \frac{\hat{R}_n^{-1} a(\theta)}{a(\theta)^H \hat{R}_n^{-1} a(\theta)} \quad (5)$$

The DoA spectrum for each range bin is stacked into a matrix form, where the n -th row is populated with the DoA spectrum of the n -th range bin. This matrix is called range-azimuth spectrum matrix S , where the (n, m) element is given by:

$$s_n = \beta \bar{P}_n$$

where

- the azimuth angle θ_m for an azimuth angle index m (6)

2.4.6 Diagonally-Loading Method

In some situations, the covariance estimate is singular or near-singular, which may cause numeric stability issues in doing the matrix inversion as a part of the DoA spectral estimation. One way to avoid the numeric stability issues is to add a small value along the diagonal of the covariance estimate matrix:

$$P_n(\theta) = \frac{1}{d(\theta)^H [\hat{R}_n + \alpha_n I]^{-1} d(\theta)} \quad (7)$$

The diagonally loading factor α_n may be determined using the noise variance estimate as follows:

$$\alpha_n = \beta \bar{P}_n \quad (8)$$

for a constant β , where $\bar{P}_n = 1/N_u \text{trace}\{R_n\}$ is the average of the diagonal terms of the covariance estimate matrix.

2.4.7 Heatmap Conditioning Algorithms

Arc Removal: Arc removal requires knowing what the average noise floor is for each range row of the heatmap. This data is collected automatically by the demo software when "rowNoise" CLI commands are not provided to the DSP at startup. In this case, the software assumes that the entire field of view (FOV) is completely free of any moving objects and will collect 60 frames, generating the average noise floor for each row, and when done, will output the rowNoise commands containing each range row's average noise value N_r . During normal runtime, the Arc removal algorithm operates on each frame's heatmap, one row at a time as described by the following pseudo-code using image processing translate and scale operations.

```

for each range row:
    if (range_row_min_val > Nr)
        row_range = range_row_max_val - range_row_min_val
        scale_factor = 1.0 + ((range_row_min_val - Nr) / row_range)

    for each cell in the range row:
        //Translate the row to zero and rescale
        rescaled_val = (cell - range_row_min_val) * scale_factor
        //Translate the scaled row to the noise floor
        cell = rescaled_val + Nr
    
```

Heatmap Smoothing: Heatmap smoothing requires keeping the previous frame's heatmap in memory for comparison to the current heatmap. Each cell of the current heatmap is compared to the corresponding cell in the previous heatmap times a threshold. The threshold is simply a percentage (ie. 0.5 = 50%). The pseudo-code is:

```

for each cell in the heatmap:
    cell_thresh = previous_cell * threshold

    if (current_cell < cell_thresh)
        current_cell = cell_thresh
    
```

So the effect is that sudden spikes in power will appear instantly, while sudden drops in power will fade over time, depending on the threshold level. The higher the threshold (up to 99%) the slower the fade. A threshold of zero disables smoothing, as heatmap values are always positive.

2.4.8 Feature Extraction Algorithms

Notations:

- N_z is the number of zones.
- r_n (for $n = 1, 2, \dots, N_r$) is the range value for the range bin index n , and θ_m (for $m = 1, 2, \dots, N_{az}$) is the azimuth angle for the azimuth angle index m , where N_r is the number of range bins and N_{az} is the number of azimuth angle bins.
- $S[t]$ is the $N_r \times N_{az}$ range-azimuth spectrum matrix at radar frame t , whose (n,m) -th element is denoted as $S_{n,m}[t]$.

2.4.9 Definition of Zones

Zones may be defined in range and azimuth-angle domain, and depend on the mounting location and attitude of the radar sensor. For example, in-cabin applications for automotive vehicles, each driver/passenger seats can be defines as zones. A zone can be defined as follows with four parameters.

$$Z_i = \{(n, m): r_L^i \leq r_n < r_u^i, \theta_L^i \leq \theta_m < \theta_U^i\} \quad (9)$$

Z_i is the set of all the range-azimuth angle grids within the i -th zone (a rectangular-shaped boundary in range-azimuth domain) which is defined by four parameters, r_L^i , r_U^i , θ_L^i , and θ_U^i .

2.4.10 Features

For each zone, several features used in determining the occupancy state are extracted. The features may be derived from the average power for each zone (referred to as 'zone power'). For zone i at a radar frame index t , the average zone-power is defined as follows:

$$Q_i[t] = \frac{1}{|Z_i|} \sum_{(n,m) \in Z_i} S_{n,m}[t] \quad (10)$$

Features are defined as follows:

Moving-averaged zone power: for zone i at frame t ,

$$\bar{Q}_i[t] = \frac{1}{L} \sum_{l=t-L+1}^t Q_i[l]$$

where

- L is the window length for the moving-average. (11)

- Moving-averaged power ratio: for zone i at frame t ,

$$q_i[t] = \frac{\bar{Q}_i[t]}{\sum_{i=1}^{N_z} \bar{Q}_i[t]} \quad (12)$$

- Correlation coefficient of zone power: between zone i and zone j ,

$$\rho_{i,j}[t] = \frac{1}{L-1} \sum_{l=t-L+1}^t \left(\frac{Q_i[l] - \bar{Q}_i[l]}{\sigma_i[t]} \right) \left(\frac{Q_j[l] - \bar{Q}_j[l]}{\sigma_j[t]} \right)$$

where

- $\sigma_i[t]$ is the standard deviation of $Q_i[t]$ for zone i , which is defined as follows: (13)

$$\sigma_i[t] = \left[\frac{1}{L} \sum_{l=t-L+1}^t (Q_i[l] - \bar{Q}_i[l])^2 \right]^{\frac{1}{2}} \quad (14)$$

The feature set for a radar frame is formed by stacking all the features. For example, when $N_z = 3$, the feature set vector is 9-dimensional vector $x[t]$ as follows:

$$x[t] = [\bar{Q}_1[t], \bar{Q}_2[t], \bar{Q}_3[t], q_1[t], q_2[t], q_3[t], \rho_{1,2}[t], \rho_{2,3}[t], \rho_{3,1}[t]]^T \quad (15)$$

In the following sections, the element of the feature vector is sometimes also denoted as $x[t] = [x_1[t], x_2[t], \dots, x_N[t]]^T$, where N is the number of features.

2.4.11 Zone Detection Algorithms

For N_z zones, there are 2^{N_z} possible occupancy states for the area of interest. The binary occupancy state for zone i is denoted by $O_i[t] \in \{0,1\}$. $O_i[t] = 0$ and $O_i[t] = 1$ represents that the zone i at radar frame t is 'empty' and 'occupied', respectively. For example, when N_z the occupancy status for the all the zones are:

$$(O_1[t], O_2[t], O_3[t]) \in \{(0,0,0), (1,0,0), (0,1,0), (1,1,0), (0,0,1), (0,1,1), (1,1,1)\} \quad (16)$$

For each of possible occupancy decision states, we define $(N+1)$ dimensional decision weight vector (also called decision parameters):

$$W_{(o_1, o_2, \dots, o_{N_z})} = [W_{(o_1, o_2, \dots, o_{N_z})}^0, W_{(o_1, o_2, \dots, o_{N_z})}^1, \dots, W_{(o_1, o_2, \dots, o_{N_z})}^{N_z}]^T$$

where

- N is the number of element in feature vector $x[t]$, w_0 is for bias term. (17)

The probability (or likelihood) of having a particular occupancy state is defined as follows:

$$\begin{aligned} \hat{p}_{(o_1, o_2, \dots, o_{N_z})}[t] &= g \left(W_{(o_1, o_2, \dots, o_{N_z})}^T \tilde{x}[t] \right) \\ &= g \left(\sum_{i=0}^N W_{(o_1, o_2, \dots, o_{N_z})}^i r_i^x[t] \right) \end{aligned}$$

where

- $\tilde{x}[t] = [1; x[t]] = [x_0[t], x_1[t], \dots, x_{N_z}[t]]^T$ (obtained by pre-appending '1' to the feature vector $x[t]$. Note that $x_0[t] = 1$ always), and $g(\cdot)$ is a non-linear function, e.g., logistic function, (18)

$$g(z) = \frac{1}{1 + e^{-z}} \quad (19)$$

Zone occupancy state is estimated as follows:

$$(\hat{O}_1[t], \hat{O}_2[t], \dots, \hat{O}_{N_z}[t]) = \arg_{(o_1, o_2, \dots, o_{N_z}) \in \{0,1\}^{N_z}} \max \hat{p}_{(o_1, o_2, \dots, o_{N_z})}[t] \quad (20)$$

Because $g(\cdot)$ is an increasing function, the occupancy decision making is equivalent to:

$$(\hat{O}_1[t], \hat{O}_2[t], \dots, \hat{O}_{N_z}[t]) = \arg_{(o_1, o_2, \dots, o_{N_z}) \in \{0,1\}^{N_z}} \max W_{(o_1, o_2, \dots, o_{N_z})}^T \tilde{x}[t] \quad (21)$$

Furthermore, the zone detection can be implemented with a matrix-vector multiplication as shown in [Figure 3](#), by stacking $w_{(o_1, o_2, \dots, o_{N_z})}$ into a $2^{N_z} \times (1 + N)$ matrix W as follows:

$$W = \begin{bmatrix} W_{(0,0, \dots, 0)}^T \\ W_{(1,0, \dots, 0)}^T \\ \vdots \\ W_{(1,1, \dots, 1)}^T \end{bmatrix} \quad (22)$$

Then the zone-based decision may be made as follows:

$$(\hat{O}_1[t], \hat{O}_2[t], \dots, \hat{O}_{N_z}[t]) = \text{de2bi} [\arg \max g(W\tilde{x}[t])] \quad (23)$$

where

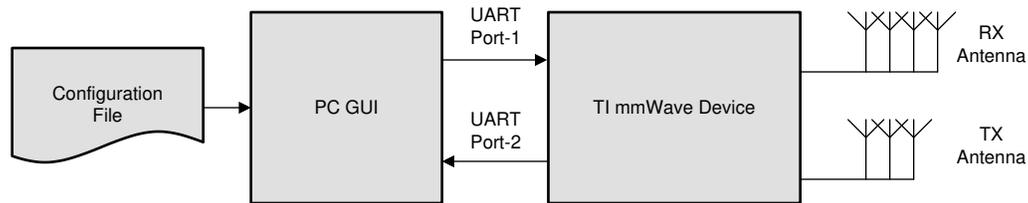
- the non-linear function $g(\cdot)$ is element-wise, and $\arg \max(\cdot)$ gives the index of element that gives the maximum in a vector, and $\text{de2bi}[\cdot]$ is a converter from a decimal value to a binary vector. (23)

For example, when $N_z = 3$, $\text{de2bi}[\cdot]$ functions as shown in the following table:

index	$(O_1[t], O_2[t], O_3[t]) = \text{de2bi}[\text{index}]$
0	(0, 0, 0)
1	(1, 0, 0)
2	(0, 1, 0)
3	(1, 1, 0)
4	(0, 0, 1)
5	(1, 0, 1)
6	(0, 1, 1)
7	(1, 1, 1)

2.4.12 Output through UART

Figure 7. mmWave Device UART Communication



As illustrated in [Figure 7](#), the example processing chain uses one UART port to receive input configuration to the front end and signal processing chain, and uses the second UART port to send out processing results for display. See the information included in the user guide for detailed information on the format of the input configuration and output results.

3 Hardware, Software, Testing Requirements, and Test Results

3.1 Required Hardware and Software

The AWR6843 and AWR1843 EVMs from Texas Instruments are easy-to-use evaluation boards for mmWave sensing. The VOD radar application runs on the EVM and connects to a visualization tool running on a PC connected to the EVM over USB.

For details regarding usage of this board, see the [AWR6843 Evaluation Module \(AWR6843ISKM\) Single-Chip mmWave Sensing Solution](#).

For details regarding the VOD GUI visualization tool, see the user guide for this demo.

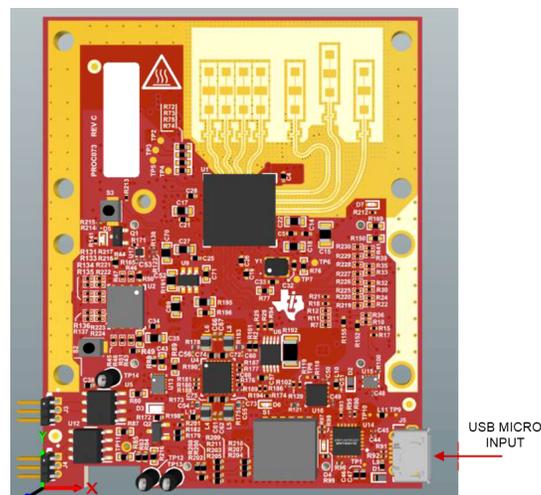
3.1.1 Hardware

The AWR6843 core design includes:

- AWR6843 device: A single-chip, 60-GHz radar device with an integrated DSP
- Power management network using a low-dropout linear regulator (LDO) and power management integrated circuit (PMIC) DC/DC supply (TPS7A88, TPS7A8101-Q1, and LP87524B-Q1)

The EVM also hosts a device to assist with onboard emulation and UART emulation over a USB link with the PC.

Figure 8. AWR6843BOOST EVM



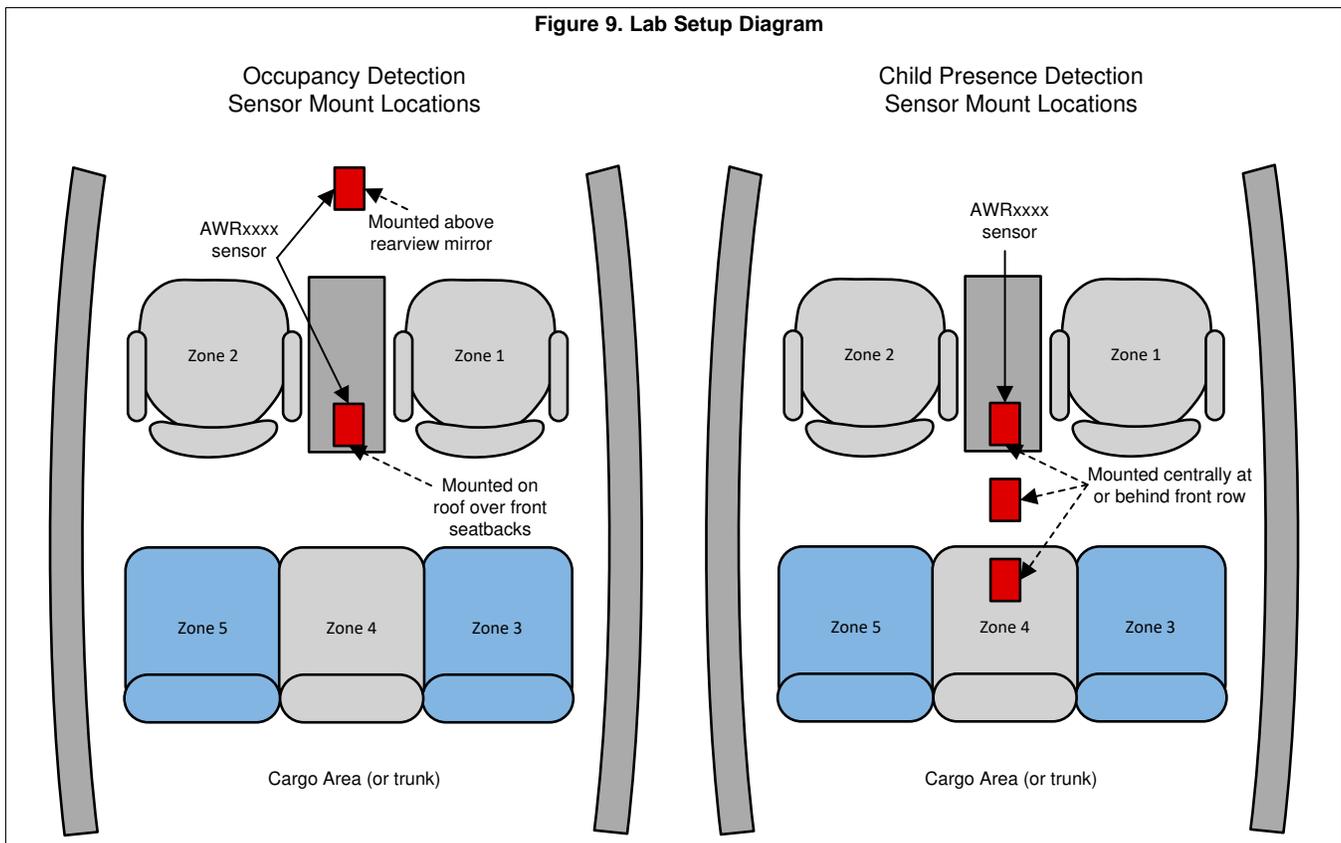
3.1.2 Software and GUI

- The mmWave SDK can be downloaded from the [mmWave software development kit \(SDK\)](#). The installation program will also install all required tool components.
- Download the [vehicle occupant detection application](#) software.
- Details on how to run the pre-built binaries and how to rebuild the demonstration application are provided in the VOD user guide in TI-Rex.

3.2 Testing and Results

3.2.1 Test Setup

Two use cases are supported by this design, VOD and CPD. Please refer to subsection 2.4.1 for a description of sensor mounting and zone configuration. The demo project provides example chirp configurations for each use case and supported device. These files are located in the /chirp_configs folder of the demo.



3.2.2 Test Results

The VOD demo GUI is instrumented with a counting widget that when started, counts frames received from the device and counts positive (occupied) detections in each zone. No averaging or smoothing is performed; it is a simple frame count. Each test was allowed to run for at least 30 seconds (approximately 200 frames at 6.25 fps), and the results are recorded in [Table 3](#). Since a large movement such as entering or exiting the vehicle causes a large spike in all zones, occupants were allowed to sit in the zones for several frames before the counting starts.

Table 3. Scenario Config file: od_demo_68xx_4zone.cfg

TEST (1)	TOTAL FRAMES	Front Left	Front Right	Back Left	Back Right	ERROR (%)
All zones empty	227	0	0	0	0	0
Static object	216	0	0	0	0	0
Front Left occupied	243	243	11	21	0	8.6%
Front right occupied	228	0	228	0	0	0
Back left occupied	217	0	0	217	0	0
Back right occupied	232	0	0	0	232	0
All zones occupied	249	249	249	249	249	0

1. Front mounted sensor test results

Test Notes: Detection software that functions well in the lab environment will encounter two obstacles when moving to a vehicle. Both of these difficulties are caused by the confined environment of the vehicle interior, and both cause extra unwanted energy in the heatmap. First, azimuthal leakage of the MVDR algorithm is increased by reflections from the sides of the vehicle. Second, multipath from all interior surfaces creates extra energy that typically co-locates with second row seats. Extra measures need to be taken to reject this extra energy, and are currently under study. Also, both of these conditions are amplified when any large motion occurs, such as when an occupant enters or exits the vehicle, changes seats, bounces in the seat, and so on. This causes widespread energy spikes that must be ignored.

Other tests were also performed with the mmWave sensor mounted in the vehicle. The first test was a child seated in a carseat in one of the zones. [Figure 10](#) shows the child in the carseat, and a capture from the VOD demo GUI during testing.

Figure 10. Child in a Car Seat



Another test illustrates an intruder approaching the vehicle. [Figure 11](#) shows the intruder and the corresponding heatmap image. A positive (occupied) detection was not gathered, because the intruder was still outside the defined zones.

Figure 11. Intruder Approaching Vehicle



4 Design Files

To download the software files, see the design files at [TIDEP-1001](#).

4.1 Schematics

To download the schematics, see the design files at [TIDEP-01001](#).

4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at [TIDEP-01001](#).

4.3 PCB Layout Recommendations

4.3.1 Layout Prints

To download the layer plots, see the design files at [TIDEP-01001](#).

4.4 Altium Project

To download the Altium Designer® project files, see the design files at [TIDEP-01001](#).

4.5 Gerber Files

To download the Gerber files, see the design files at [TIDEP-01001](#).

4.6 Assembly Drawings

To download the assembly drawings, see the design files at [TIDEP-01001](#).

5 Software Files

Download the [vehicle occupant detection application](#) software.

6 Related Documentation

1. Texas Instruments, [AWR6843 Evaluation Module \(AWR6843ISK\) Single-Chip mmWave Sensing Solution](#)
2. Texas Instruments, [Programming Chirp Parameters in TI Radar Devices](#)
3. Texas Instruments, [AWR6843 Single-Chip 60-GHz FMCW Radar Sensor](#)
4. Texas Instruments, [AWR68xx Technical Reference Manual](#)
5. Texas Instruments, [AWR6843 Evaluation Board Design Database](#)
6. Texas Instruments, [AWR6843ISK Schematic, Assembly, and BOM](#)
7. Texas Instruments, [mmWave SDK User's Guide](#)
8. Texas Instruments, [AWR6843 mmWave sensor: 60–64-GHz radar-on-chip for short-range radar applications](#)

6.1 Trademarks

TI E2E is a trademark of Texas Instruments.

ARM is a registered trademark of ARM Ltd.

Altium Designer is a registered trademark of Altium LLC or its affiliated companies.

All other trademarks are the property of their respective owners.

Revision History

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

Changes from Original (April 2018) to A Revision	Page
• Changed AWR1642 to AWR6843 throughout document	1
• Added (VOD) and Child Presence Detection (CPD)	1
• Added Software Demonstration Source Code.....	1
• Changed <i>child left behind in car</i> to <i>child presence detection</i>	1
• Changed board image	1
• Changed 2 to <i>sensor</i>	2
• Changed 70 to 60	2
• Changed (or AWR1843 sensor based on 77-GHz mmWave radio-frequency)	2
• Changed 76 GHz to 81 GHz to 81 GHz mmWave radar front end with 57-GHz to 60-GHz (or 76-GHz to 81-GHz) mmWave radar front end with	2
• Changed <i>two regions of interest. This can be extended to multiple regions detection.</i> to <i>multiple regions (zones) of interest, and can localize occupant signature within a region.</i>	2
• Changed Updated System Description section.....	2
• Changed Figure 1 Software Block Diagram	3
• Changed <i>Average zone power</i> to <i>Zone power. Originally, this was a simple average across the entire zone. But large zones suffer because the power of an occupant signature gets diluted due to the number of noise level cells in the zone. Now, the zone is searched for the maximum 5x5 region, and the average of the 5x5 is used for zone power.</i>	4
• Changed <i>Moving average...</i> to <i>Frame-averaged</i>	4
• Added 57-GHz to 60-GHz and	4
• Changed Figure 3 : AWR6843/1843BOOST EVM Block Diagrams.....	5
• Changed mmWave Toolboxes	5
• Changed <i>Memory</i> to <i>Radar cube</i>	7
• Changed by adding VOD demo and added additional text	7
• Changed Figure 5 : VOD Profile Configuration File	7
• Changed Table 3 title from <i>Memory Resources</i> to <i>DSP Memory Resources</i>	8
• Changed <i>vehicle occupancy detection example</i> to <i>VOD and CPD use cases</i>	9
• Added (the "inter-frame" period).....	9
• Added The "heatmap" is actually a velocity map, with larger magnitude values showing more movement of non-static objects.	9
• Added Range-Azimuth heatmap conditioning — To help overcome issues with clutter removal and DoA, the heatmap is further processed to suppress MVDR arcs, and and to smooth single-frame drops caused by frame variations in clutter removal.	9
• Added subtitle: Heatmap Conditioning Algorithms	11
• Changed <i>Two test scenarios were used to test the VOD application software: a lap setup and a vehicle setup. The lab setup attempted to replicate the seating area of a vehicle, but without the confinement of door and roof panels. The vehicle setup created a zone in each of the back seats of a common vehicle. when the test setup is replicated, the demo application can be run as described in the user guide, using the appropriate configuration file.</i> to <i>Two use cases are supported by this design, VOD and CPD. Please refer to subsection 2.4.1 for a description of sensor mounting and zone configuration. The demo project provides example chirp configurations for each use case and supported device. These files are located in the /chirp_configs folder of the demo.</i>	16
• Changed Figure 9 : Lab Setup Diagram.....	16
• Deleted Figure 10 : Lab Test Setup; Figure 11 : Vehicle Test Measurements; and Figure 12 : Vehicle Test Setup.....	16
• Deleted <i>To create a testing criteria...</i>	16
• Deleted <i>For each scenario, all possible occupied and empty zone combinations were tested.</i>	16
• Changed 1000 frames to 30 seconds.....	16
• Changed 75 minutes at 6 frames/seconds), and the resulting counts are recorded in the following tables. These tests required that the test subjects are in place when the frame counting begins, and nothing enters an "empty" zone during the test period. to 200 frames at 6.25 fps), and the results are recorded in Table 3 . Since a large movement such as entering or exiting the vehicle causes a large spike in all zones, occupants were allowed to sit in the zones for several frames before the counting starts.	16
• Changed title of Table 4 from <i>od_demo_carseats_302_3p0.cfg</i> to <i>Scenario Config file: od_demo_68xx_4zone.cfg</i>	17

• Changed data in Table 4	17
• Added Table Note 1	17
• Changed Figure 13: Child in a Car Seat.....	17
• Changed Figure 14: Intruder Approaching Vehicle.....	18

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), 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, regulatory 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](#) 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.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2022, Texas Instruments Incorporated