参考資料

TI Designs: TIDEP-01001
車両の乗員検出のリファレンス・デザイン

TEXAS INSTRUMENTS

概要

このリファレンス・デザインでは、DSP内蔵のシングル・チップmmWaveセンサであるAWR1642を車両の乗員検出センサとして使用し、車内の生命体を検出する方法を紹介します。このデザインでは、±60°の視野角(FOV)内に存在する生命体を検出するためのヒートマップの生成を可能にする、C674x DSP上で実行される参照用処理チェーンを提供します。

リソース

TIDEP-01001	デザイン・フォルダ
AWR1642	プロダクト・フォルダ
AWR1642BOOST	ツール・フォルダ
mmWaveSDK	ソフトウェア開発キット

TI E2E™ Community

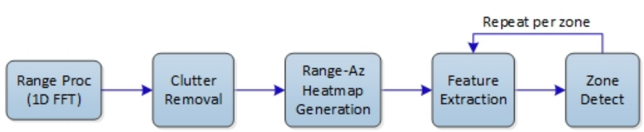
E2E™エキスパートに質問

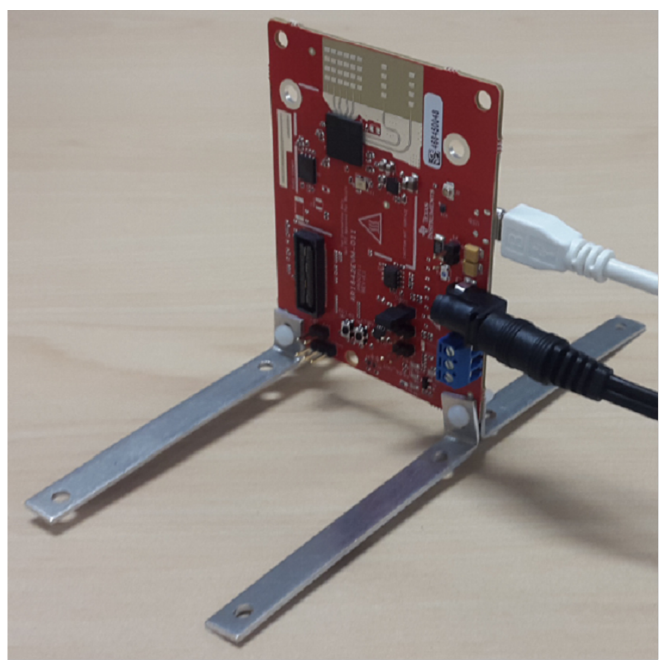
特長

- mmWaveセンサ・テクノロジによる車内の生命体(大人、子供、ペット)の確実な検出のデモ
- ±60°のFOVを持つ存在ヒートマップの生成
- mmWaveソフトウェア開発キット(SDK)を基礎とする処理および検出のソース・コード
- 実績のあるEVMハードウェア設計を基礎とし、短期間での市場投入と、即時のデモンストレーションを実現
- レーダー・フロント・エンドと検出回路構成に関する詳細

アプリケーション

- 車内に取り残された子供の検出
- 車両の乗員検出
- 侵入者検出





TIDUE95 翻訳版 — 最新の英語版資料 http://www-s.ti.com/sc/techlit/TIDUE95

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 AWR1642 based on 77-GHz mmWave radio-frequency complementary metal-oxide semiconductor (RF-CMOS) technology.

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

This reference design demonstrates the suitability of the AWR1642 for vehicle occupant detection applications. This design targets the implementation of a wide, azimuth field of view (±60°), close range (3 m) sensor configuration, which can detect life forms across two regions of interest. This can be extended to multiple regions detection.

This TI Design implements algorithms for generating an azimuth-range heat map, detection, and decision for an AWR1642 device on a TI EVM module.

The design provides a list of required hardware, schematics, and foundational software to quickly begin traffic monitoring 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 TI design is built around the AWR1642EVM evaluation board and the millimeter wave (mmWave) SDK demo application. The system is optimized and built for VOD 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 AWR1642 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 TI Design is a separate package installed on top of the SDK package.

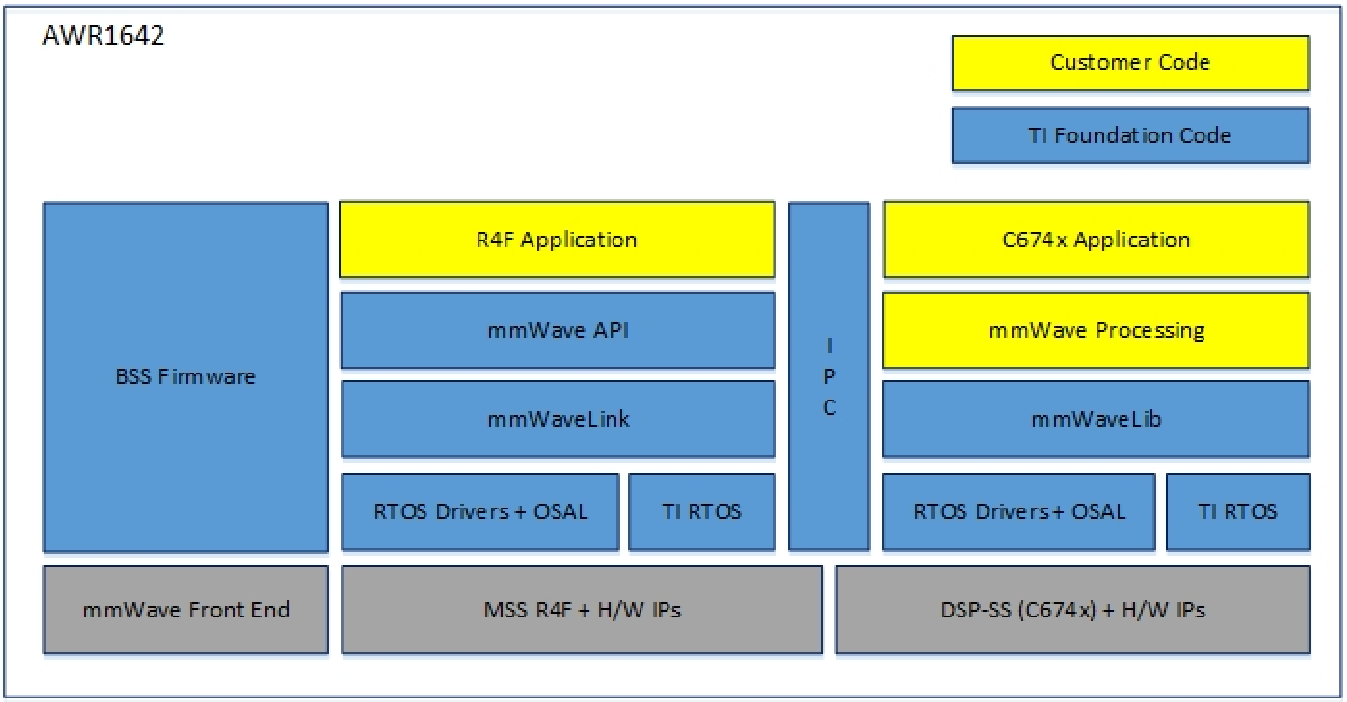


図 1. Software Block Diagram

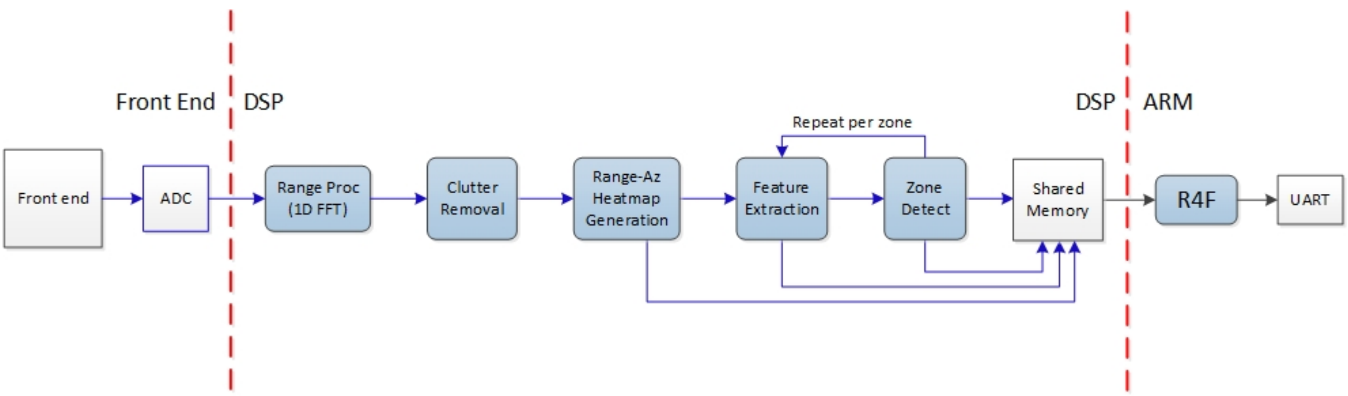
2.2 Vehicle Occupancy Detection Software Block Diagram

As described in 図 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 AWR1642:

- 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).
- Feature Extraction
  - Each frame, scan the heatmap within each defined zone of interest and compute a feature vector:
    - Average zone power
    - Moving-average zone power (for L (window length), frames)
    - Moving-average power ratio (for L frames)
    - Correlation coefficient of zone power
- Zone Detection
  - For N_z zones, there are 2^{Nz} possible occupancy states. Offline, define a matrix of decision parameters (coefficients) to represent targets of interest (adults, children, pets, etc).
  - 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'.

図 2. Application Software Block Diagram



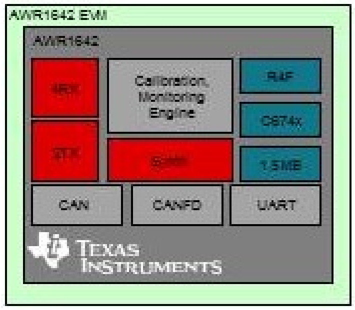
After the DSP finishes frame processing, the results are formatted and written in shared memory (HSRAM) for the R4F MCU to send to the host through a UART for visualization.

2.3 Highlighted Products

2.3.1 AWR1642 Single-Chip Radar Solution

The AWR1642 is an integrated single-chip FMCW sensor capable of operation in the 76-GHz to 81-GHz band. It is built with TI's low power 45-nm RFCMOS process, and enables unprecedented levels of integration in an extremely small form factor. The AWR1642 is an ideal solution for low power, self-monitored, ultra-accurate radar systems in the automotive and industrial space.

図 3. AWR1642BOOST EVM Block Diagram



The AWR1642 EVM has the following features:

- AWR1642 radar device
- Power management circuit to provide all the required supply rails from a single 5-V input.
- Two onboard TX antennas and four RX antennas
- Onboard XDS110, which provides JTAG interface, UART1 for loading the radar configuration on the AWR1642 device, and UART2 to send the object data back to the PC.

  For more details on the hardware, see the *AWR1642 Evaluation Module (AWR1642EVM) Single-Chip mmWave Sensing Solution*. The schematics and design database can be found in the following documents: the *AWR1642 Evaluation Board Design Database* and the *AWR1642EVM Schematic, Assembly, and BOM* .

2.3.2 mmWaveSDK

The mmWave SDK is split in two broad components: the mmWave Suite and mmWave demos. 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 demos provide a suite of demonstrations that depict the various control and data processing aspects of an mmWave application. Data visualization of the demonstration's output on a PC is provided as part of these demonstrations:

- mmWave processing demonstration
- ADC data streaming demonstration

2.4 System Design Theory

2.4.1 Use Case Geometry and Sensor Considerations

The AWR1642 is a radar-based sensor that integrates 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 AWR1642 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

図 4. Example Zone Geometry

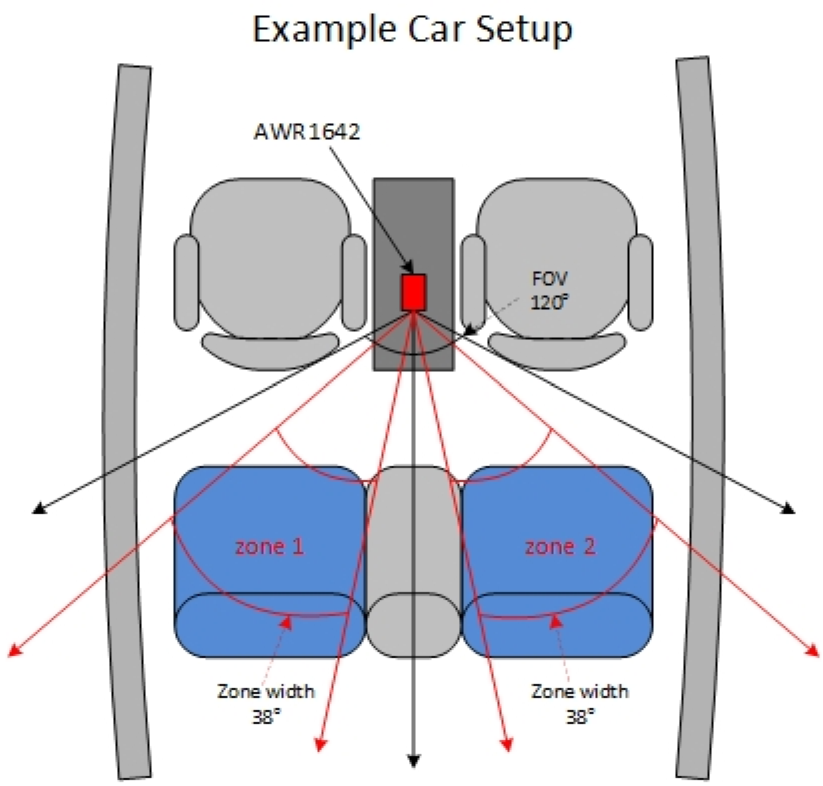


図 4 shows the zone geometry for the example configurations. In these examples, the primary field of zone coverage begins at about 61 mm and continues to 136 mm. The zone widths are roughly 38 degrees each. Tuning of the zones is completely configurable, as described in the user guide. There is a small amount of spectral leakage from cell to cell, so an object entering a heatmap cell adjacent to a defined zone cell could cause enough averaged energy in the zone to cause a transient positive detection.

2.4.2 Chirp Configuration used and System Performance

An example configuration for the VOD application is given in 表 1.

表 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

表 1. Example Configuration (continued)

PARAMETER	SPECIFICATION
Starting frequency (GHz)	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
Memory requirements (KB)	512

The above configurations are just illustrative, and can be tailored according to user requirements

2.4.3 Configuration Profile

The demo applications in the mmWave SDK distribution let the user push the radar configuration using a “Profile Configuration” file over UART to the AWR1642 EVM. The mmWave SDK user guide (included in the mmWave SDK distribution) describes the semantics of the following commands in detail. In the configuration files for the vehicle occupancy demo, additional commands are created that allow configurability of the specialized algorithms. These commands are described in the demo’s user guide.

図 5. VOD Profile Configuration File

```
sensorStop
flushCfg
dfeDataOutputMode 1
channelCfg 15 3 0
adcCfg 2 1
adcbufCfg -1 0 0 1 1
profileCfg 0 77 250 10 40 0 0 98 1 64 2200 0 0 40
chirpCfg 0 0 0 0 0 0 0 1
chirpCfg 1 1 0 0 0 0 0 2
chirpCfg 2 2 0 0 0 0 0 1
chirpCfg 3 3 0 0 0 0 0 2
frameCfg 0 3 128 0 160 1 0
lowPower 0 0
guiMonitor -1 0 1 1
calibDcRangeSig -1 0 -5 8 256

%OD Demo commands:
zoneDef 2 13 16 9 15 13 16 27 15
coeffMatrixRow  0  -14.409613   -8.187467   -8.019457    3.833826    4.045485   -0.539210
coeffMatrixRow  1  -12.465002   -2.228748   -5.612140    4.324446   -7.912656    0.528747
coeffMatrixRow  2  -9.210626   -3.004206   -0.082298   -2.949700    7.538751    0.246658
coeffMatrixRow  3  -1.917373    2.299849    2.389683    7.035545    6.675194   -0.099186
meanVector  23.992751   23.689970   -3.388539   -3.691319    0.816150
stdVector   8.495360    8.497394    2.198640    2.432388    0.196337
oddemoParms  12  0.001
sensorStart
```

The profile configuration (profileCfg), defines the profile of a single chirp (as seen in 図 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 AWR1642 EVM.

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 programed 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 表 2.

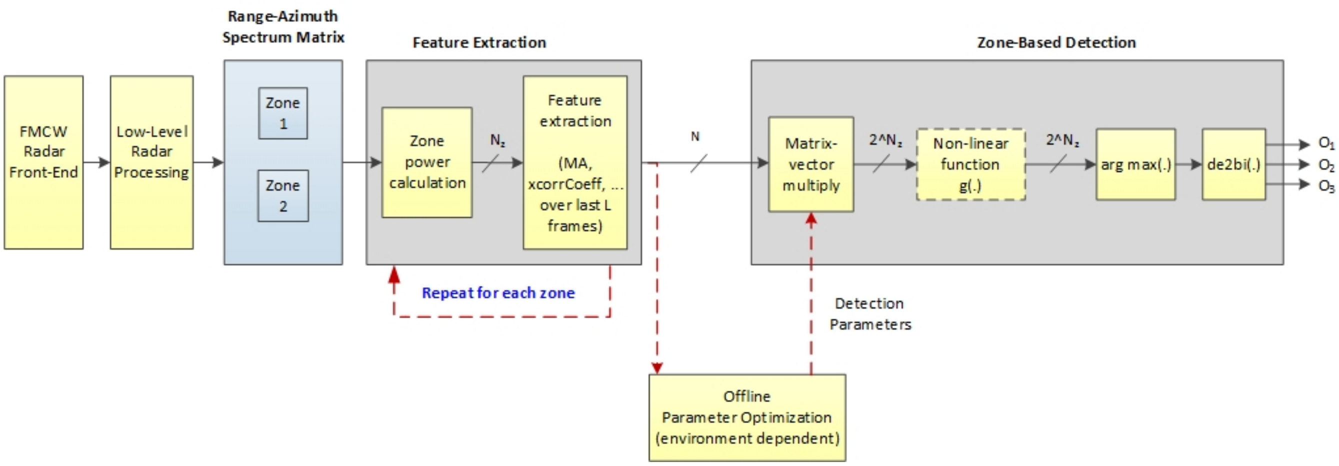
表 2. Memory Resources

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.

表 2. Memory Resources (continued)

Section Name	Size (KB) as Configured	Memory Used (KB)	Description
HSRAM	32	32	Shared memory buffer between the DSP and theR4F relays visualization data to the R4F for output over the UART in this design.

図 6. Processing Chain Flow



As shown in 図 6, 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 AWR1642:

- 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 to move output from DSP local memory to 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.
- 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.
- 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:

$$x_{n,k} = \left[x_{n,k,1},\ x_{n,k,2},\ \ldots\ x_{n,k,N_a}\right]^T \quad (2)$$

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 α(θ) a steering vector for an azimuth angle θ as follows: for a linear one-dimensional virtual receiver antenna array,

$$\alpha(\theta) = \left[1,\ exp\left(j2\pi d\sin(\theta)\right),\ \ldots\ exp\left(j2\pi\left(N_a - 1\right)d\ \sin(\theta)\right)\right]^T$$

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:

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

And the corresponding beamforming vector is given as:

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

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 (see 図 1), whose (n, m) element is given by:

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

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:

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

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 $P_n = 1/N_\alpha$ trace{R_n}) is the average of the diagonal terms of the covariance estimate matrix.

2.4.7 Feature Extraction Algorithms

Notations:

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

2.4.8 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^i_L, r^i_U, θ^i_L, and θ^i_U.

2.4.9 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:

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

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:

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

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

2.4.10 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 $O_i[t]$ E {0,1}. $O_i[t]$ = 0andOi[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):

$$w_{(o_1,\ o_2,\ \ldots,\ o_{N_Z})} = \left[w_{(o_1,\ o_2,\ \ldots,\ o_{N_Z})}^{0},\quad w_{(o_1,\ o_2,\ \ldots,\ o_{N_Z})}^{1},\quad \ldots,\quad w_{(o_1,\ o_2,\ \ldots,\ o_{N_Z})}^{N}\right]^T$$

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:

$$\hat{p}_{(o_1, o_2, \ldots, o_{N_z})}[t] = g\left(w^T_{(o_1, o_2, \ldots, o_{N_z})}\tilde{x}[t]\right) = g\left(\sum_{i=0}^{N} w_{(o_1, o_2, \ldots, o_{N_z}), i}\, x_i[t]\right)$$

where

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

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

Zone occupancy state is estimated as follows:

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

Because *g(.)* is an increasing function, the occupancy decision making is equivalent to:

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

Furthermore, the zone detection can be implemented with a matrix-vector multiplication as shown in **Figure 3**, by stacking $w_{(o1, o2, \ldots, oNz)}$ into a 2^{Nz} x (1 + N)matrix W as follows:

$$W = \begin{bmatrix} w^T_{(0,0, \ldots, 0)} \\ w^T_{(1,0, \ldots, 0)} \\ w^T_{(1,1, \ldots, 1)} \end{bmatrix} \quad (22)$$

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

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

where

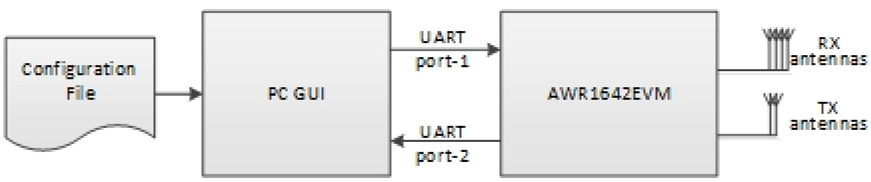
- the non-linear function g(.) is element-wise, and arg max(.) gives the index of element that gives the maximum in a vector, and de2bi[.] is a converter from a decimal value to a binary vector. (23)

For example, when N_z = 3, de2bi[.] functions as shown in the following table:

index	$(\hat{O}_1[t], \hat{O}_2[t], \hat{O}_3[t])$ = de2bi[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)

2.4.11 Output through UART

図 7. AWR1642BOOST UART Communication



As illustrated in 図 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 AWR1642EVM from Texas Instruments is an easy-to-use evaluation board for mmWave sensing devices. The VOD radar application runs on the AWR1642 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 *AWR1642 Evaluation Module (AWR1642EVM) Single-Chip mmWave Sensing Solution*.

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

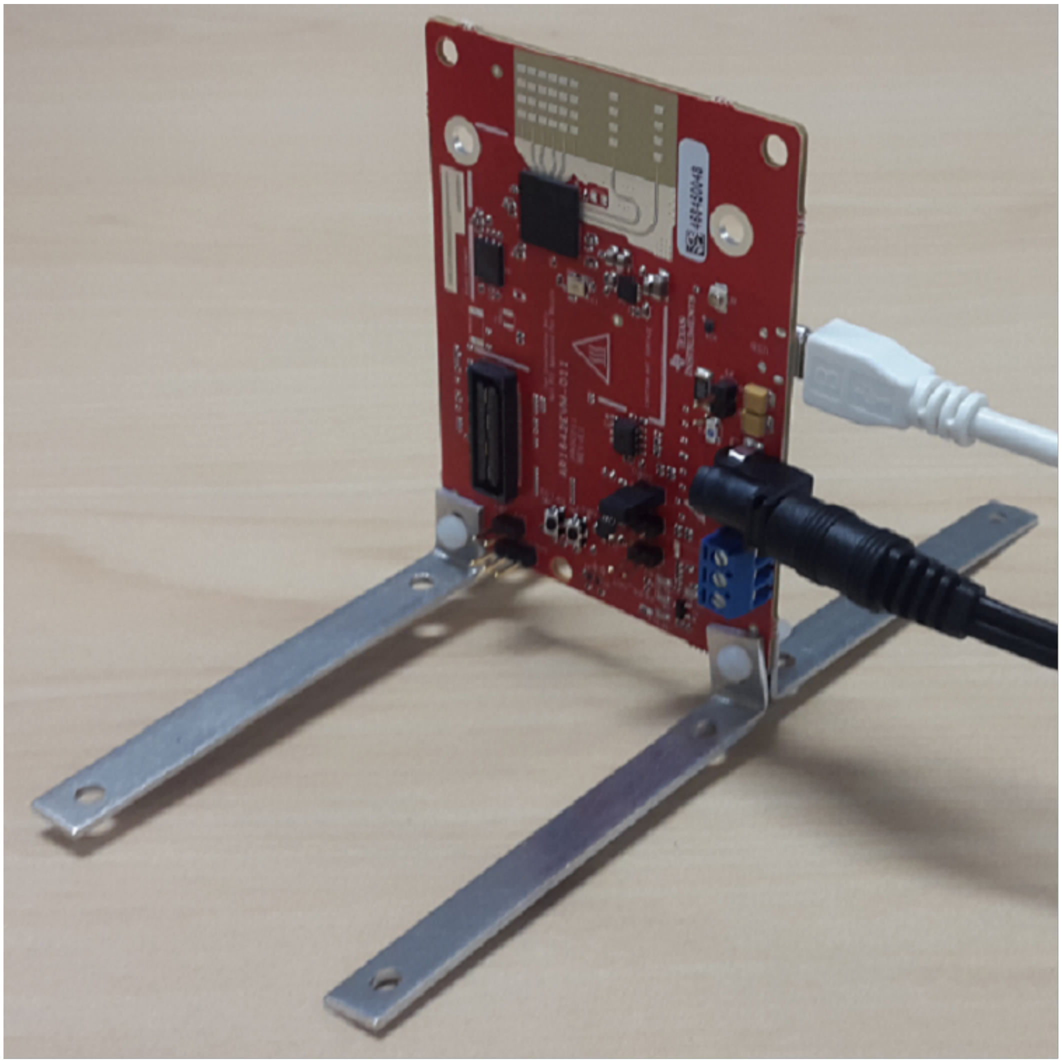
3.1.1 Hardware

The AWR1642 core design includes:

- AWR1642 device: A single-chip, 77-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.

図 8. AWR1642BOOST EVM



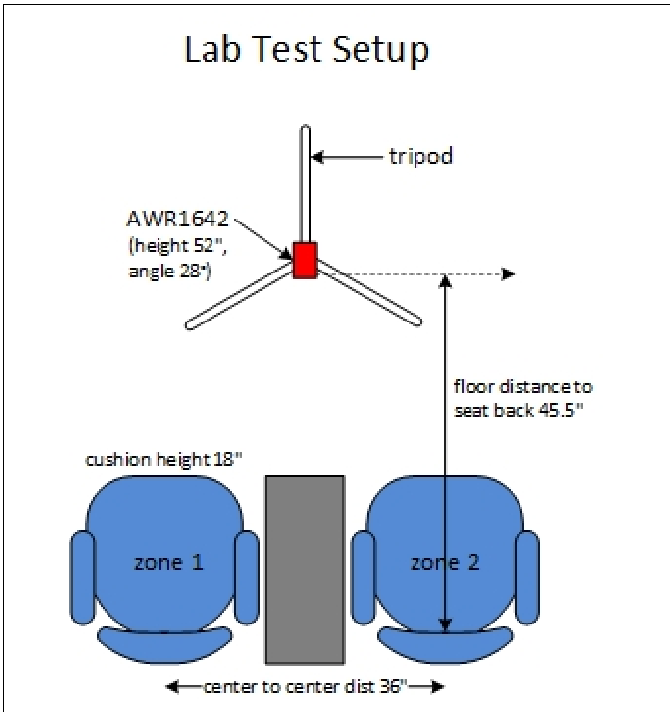
3.1.2 Software and GUI

- The mmWave SDK can be downloaded from here. The installation program will also install all required tool components.
- To download the vehicle occupant detection application software, use the following TI Resource Explorer (TI Rex) here.
- 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.

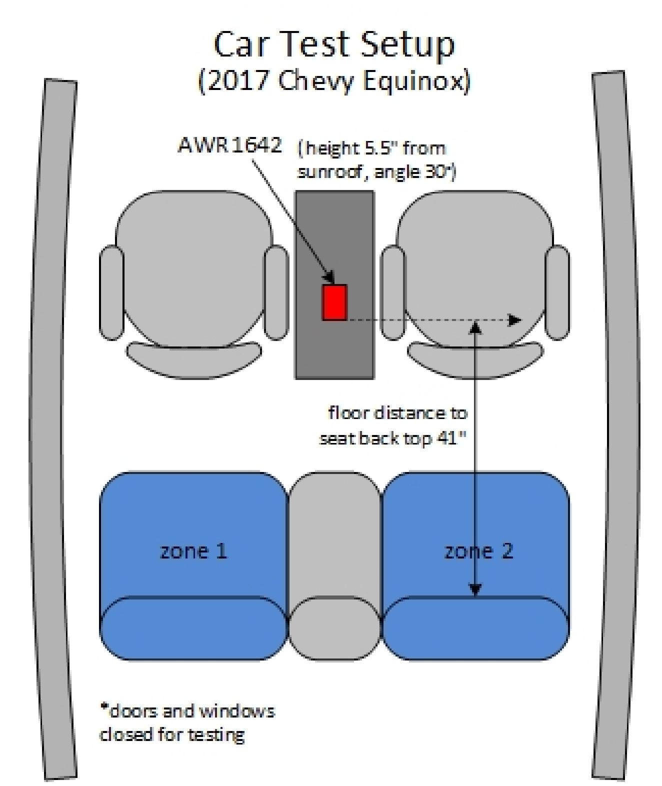
3.2 Testing and Results

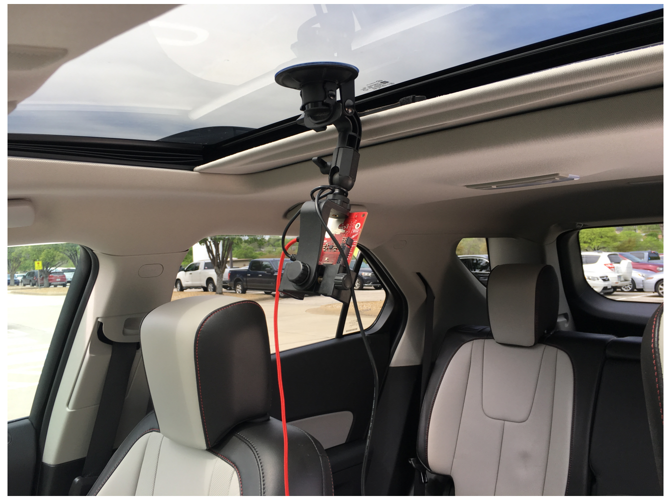
3.2.1 Test Setup

Two test scenarios were used to test the VOD application software: a lab 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.



TIDUE95 翻訳版 — 最新の英語版資料 http://www-s.ti.com/sc/techlit/TIDUE95





3.2.2 Test Results

To create a testing criteria, the VOD demo GUI was instrumented with a counting widget that when started, counts frames received from the AWR1642, and counts positive (occupied) detections in each zone per frame. No averaging or smoothing is performed; it is a simple frame count. For each scenario, all possible occupied and empty zone combinations were tested. Each test was allowed to run for at least 1000 frames (approximately 2.75 minutes at 6 frames/second), and the resulting counts are recorded in the following tables. These tests require that the test subjects are in place when the frame counting begins, and nothing enters an “empty” zone during the testing period.

表 3. Scenario Config file: od_demo_carseats_0302_3p0.cfg

Test	Total Frames	Zone1 Count	Zone1Error (%)	Zone2 Count	Zone2 Error (%)
Zone1: emptyZone2: empty	1004	0	0	0	0
Zone1: occupiedZone2: empty	1018	1018	0	0	0
Zone1: emptyZone2: occupied	1034	2	0.19	1034	0
Zone1: occupiedZone2: occupied	1025	1025	0	1011	1.3

表 4. Scenario Config file: od_demo_ car_0318_1p0.cfg

Test	Total Frames	Zone1 Count	Zone1Error (%)	Zone2 Count	Zone2 Error (%)
Zone1: emptyZone2: empty	1011	0	0	0	0

表 4. Scenario Config file: od_demo_ car_0318_1p0.cfg (continued)

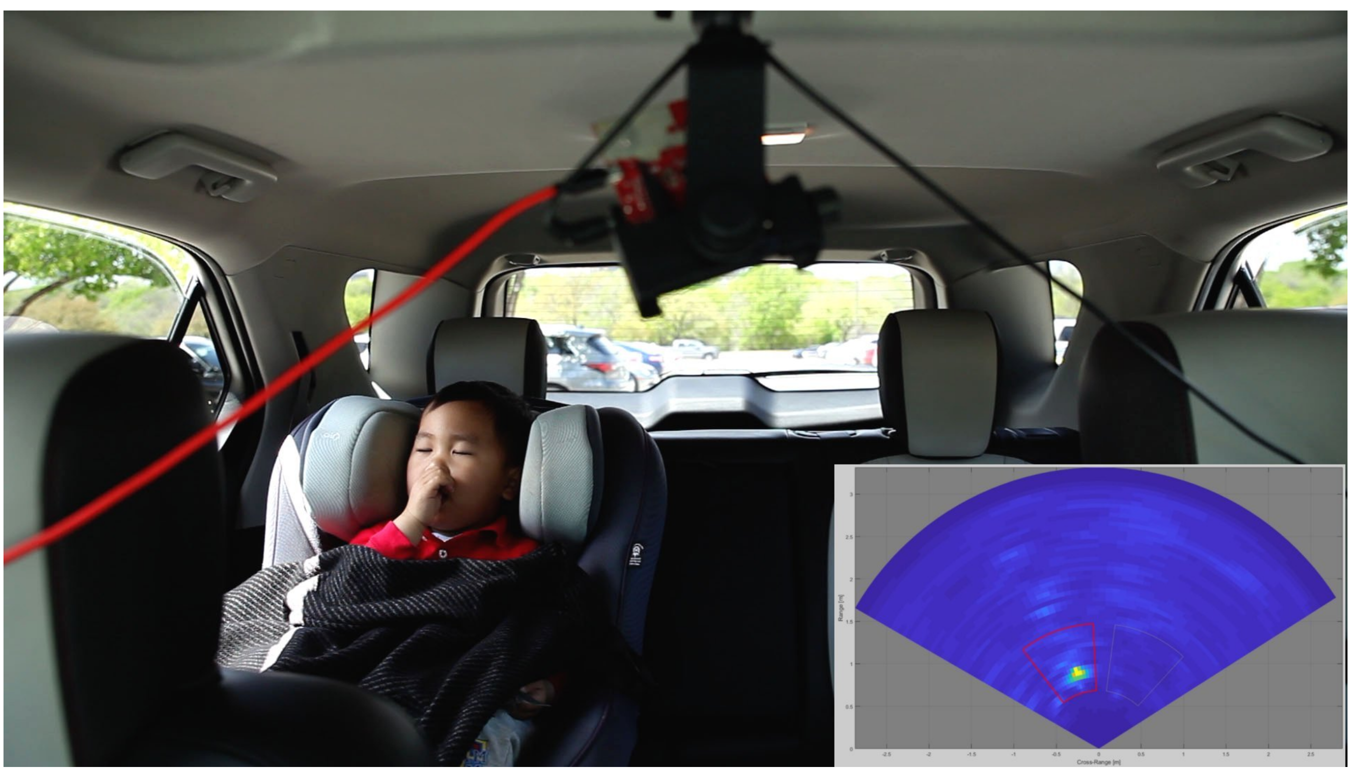
Test	Total Frames	Zone1 Count	Zone1Error (%)	Zone2 Count	Zone2 Error (%)
Zone1: occupiedZone2: empty	1008	1008	0	0	0
Zone1: emptyZone2: occupied	1007	0	0	1007	0
Zone1: occupiedZone2: occupied	1012	1012	0	1012	0

表 5. Scenario Config file: od_demo_ car_0318_3p0.cfg

Test	Total Frames	Zone1 Count	Zone1Error (%)	Zone2 Count	Zone2 Error (%)
Zone1: emptyZone2: empty	1004	0	0	0	0
Zone1: occupiedZone2: empty	1012	1012	0	0	0
Zone1: emptyZone2: occupied	1006	0	0	1006	0
Zone1: occupiedZone2: occupied	1008	1008	0	1002	0.6

Other tests were also performed with the AWR1642 sensor mounted in the vehicle. The first test was a child seated in a carseat in one of the zones. 図 13 shows the child in the carseat, and a capture from the VOD demo GUI during testing.

図 13. Child in a Car Seat



Another test illustrates an intruder approaching the vehicle. 図 14 shows the intruder and the corresponding heatmap image. A positive (occupied) detection was not gathered, because the intruder was still outside the defined zones.

図 14. 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

To download the software files, see the design files at TIDEP-01001 .

6 Related Documentation

- Texas Instruments, *AWR1642 Evaluation Module (AWR1642EVM) Single-Chip mmWave Sensing Solution*
- Texas Instruments, *Programming Chirp Parameters in TI Radar Devices*
- Texas Instruments, *AWR1642 Single-Chip 77- and 79-GHz FMCW Radar Sensor*
- Texas Instruments, *AWR14xx/16xx Technical Reference Manual*
- Texas Instruments, *AWR1642 Evaluation Board Design Database*
- Texas Instruments, *AWR1642BOOST Schematic, Assembly, and BOM*
- Texas Instruments, *mmWave SDK User's Guide*
- Texas Instruments, *AWR1642 mmWave sensor: 76–81-GHz radar-on-chip for short-range radar applications*

6.1 商標

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.
すべての商標および登録商標はそれぞれの所有者に帰属します。

TIの設計情報およびリソースに関する重要な注意事項

Texas Instruments Incorporated ("TI")の技術、アプリケーションその他設計に関する助言、サービスまたは情報は、TI製品を組み込んだアプリケーションを開発する設計者に役立つことを目的として提供するものです。これにはリファレンス設計や、評価モジュールに関係する資料が含まれますが、これらに限られません。以下、これらを総称して「TIリソース」と呼びます。いかなる方法であっても、TIリソースのいずれかをダウンロード、アクセス、または使用した場合、お客様(個人、または会社を代表している場合にはお客様の会社)は、これらのリソースをここに記載された目的にのみ使用し、この注意事項の条項に従うことに合意したものとします。

TIによるTIリソースの提供は、TI製品に対する該当の発行済み保証事項または免責事項を拡張またはいかなる形でも変更するものではなく、これらのTIリソースを提供することによって、TIにはいかなる追加義務も責任も発生しないものとします。TIは、自社のTIリソースに訂正、拡張、改良、およびその他の変更を加える権利を留保します。

お客様は、自らのアプリケーションの設計において、ご自身が独自に分析、評価、判断を行う責任がお客様にあり、お客様のアプリケーション(および、お客様のアプリケーションに使用されるすべてのTI製品)の安全性、および該当するすべての規制、法、その他適用される要件への遵守を保証するすべての責任をお客様のみが負うことを理解し、合意するものとします。お客様は、自身のアプリケーションに関して、(1) 故障による危険な結果を予測し、(2) 障害とその結果を監視し、および、(3) 損害を引き起こす障害の可能性を減らし、適切な対策を行う目的での、安全策を開発し実装するために必要な、すべての技術を保持していることを表明するものとします。お客様は、TI製品を含むアプリケーションを使用または配布する前に、それらのアプリケーション、およびアプリケーションに使用されているTI製品の機能性を完全にテストすることに合意するものとします。TIは、特定のTIリソース用に発行されたドキュメントで明示的に記載されているもの以外のテストを実行していません。

お客様は、個別のTIリソースにつき、当該TIリソースに記載されているTI製品を含むアプリケーションの開発に関連する目的でのみ、使用、コピー、変更することが許可されています。明示的または黙示的を問わず、禁反言の法理その他どのような理由でも、他のTIの知的所有権に対するその他のライセンスは付与されません。また、TIまたは他のいかなる第三者のテクノロジまたは知的所有権についても、いかなるライセンスも付与されるものではありません。付与されないものには、TI製品またはサービスが使用される組み合わせ、機械、プロセスに関連する特許権、著作権、回路配置利用権、その他の知的所有権が含まれますが、これらに限られません。第三者の製品やサービスに関する、またはそれらを参照する情報は、そのような製品またはサービスを利用するライセンスを構成するものではなく、それらに対する保証または推奨を意味するものでもありません。TIリソースを使用するため、第三者の特許または他の知的所有権に基づく第三者からのライセンス、もしくは、TIの特許または他の知的所有権に基づくTIからのライセンスが必要な場合があります。

TIのリソースは、それに含まれるあらゆる欠陥も含めて、「現状のまま」提供されます。TIは、TIリソースまたはその仕様に関して、明示的か暗黙的かにかかわらず、他のいかなる保証または表明も行いません。これには、正確性または完全性、権原、続発性の障害に関する保証、および商品性、特定目的への適合性、第三者の知的所有権の非侵害に対する黙示の保証が含まれますが、これらに限られません。

TIは、いかなる苦情に対しても、お客様への弁護または補償を行う義務はなく、行わないものとします。これには、任意の製品の組み合わせに関連する、またはそれらに基づく侵害の請求も含まれますが、これらに限られず、またその事実についてTIリソースまたは他の場所に記載されているか否かを問わないものとします。いかなる場合も、TIリソースまたはその使用に関連して、またはそれらにより発生した、実際的、直接的、特別、付随的、間接的、懲罰的、偶発的、または、結果的な損害について、そのような損害の可能性についてTIが知らされていたかどうかにかかわらず、TIは責任を負わないものとします。

お客様は、この注意事項の条件および条項に従わなかったために発生した、いかなる損害、コスト、損失、責任からも、TIおよびその代表者を完全に免責するものとします。

この注意事項はTIリソースに適用されます。特定の種類の資料、TI製品、およびサービスの使用および購入については、追加条項が適用されます。これには、半導体製品(http://www.ti.com/sc/docs/stdterms.htm)、評価モジュール、およびサンプル(http://www.ti.com/sc/docs/sampterms.htm)についてのTIの標準条項が含まれますが、これらに限られません。