Design Guide: TIDEP-01000

People Tracking and Counting Reference Design Using mmWave Radar Sensor

Description

The TIDEP-01000 reference design demonstrates the use of the IWR6843, a single-chip, mmWave radar sensor from TI with integrated DSP, for an indoor and outdoor people-counting application. This reference design uses all three variants of the IWR6843: ISK + ICB (Industrial Carrier Board), ODS + ICB, and AOP (Antenna on Package) and integrates a complete radar-processing chain onto the IWR6843 device. This solution is shown to localize people in 3D space out to 15 m, with an location accuracy of +- 10cm and a counting density of 1 person per square meter.

Resources

TIDEP-01000	Design Folder
3D People Counting	3D People Counting TI Resource Explorer Folder
IWR6843	Product Folder
IWR6843ISK	Tool Folder for ISK Board

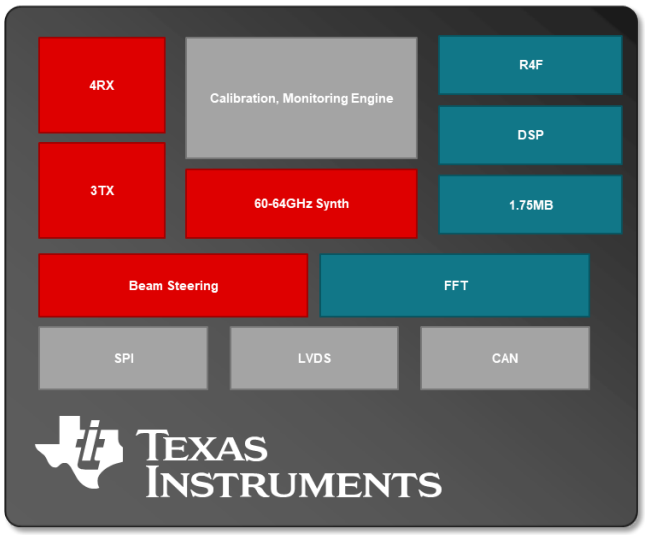
Ask our TI E2E™ support experts

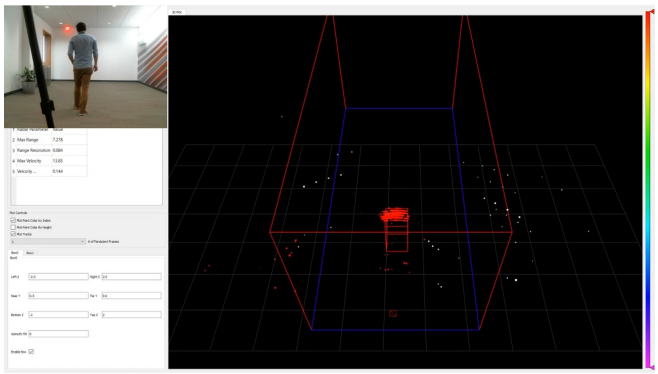
Features

- Demonstration hardware and software using IWR6843 and IWR6843AOP, mmWave radar sensor for people counting
- mmWave technology provides range, velocity, and angle information
- All detection and signal processing handled onboard IWR6843, point cloud and object information streamed to PC for visualization
- Approximate azimuth field of view of 120 degrees and elevation field of view of 40 degrees across distance of 15 m
- Examples of implementation of capon beamforming signal chain and group tracking algorithms

Applications

- People counting
- Motion detector
- Automated doors and gates
- IP network camera
- Lighting sensors
- Safety guards





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

NOTE: This design previously used the IWR1642 in 2Tx Mode. The source code for the IWR1642 People Counting Demo is available on TIREX.

1 System Description

The TIDEP-01000 provides a reference for building a people-tracking application that can detect and track humans up to 15 meters away. This design is a custom data processing demo application built to work on the IWR6843, an integrated single-chip frequency modulated continuous wave (FMCW) radar sensor capable of operation in the 60 to 64 GHz band.

Human monitoring has become an important area of exploration, due to its potential for understanding people's activities, intents, and health issues. The ability to continuously and consistently monitor human motion is an important function in numerous applications, including surveillance, appliance control, and analysis. Accuracy and precision plays an important role in these applications. While sensors such as passive infra-red (PIR) and time of flight (TOF) are in use, they suffer from limitations in accuracy, false alarms, and environmental changes such as darkness, brightness, and smoke.

Radarprovides accurate position and velocity measurement of people in the area of interest. They are relatively immune to environmental conditions such as the effects of rain, dust, or smoke. Additionally, they can work in complete darkness or in bright daylight. They are therefore useful for building automation applications such as people counting, motion detection, IP network cameras, and safety guards.

The design provided is an introductory-demo application developed on the IWR6843ISK and IWR6843ISK-ODS to accurately count and track the number of people in the area of interest. The radar data processing chain implements the static clutter removal, range azimuth heat map, elevation estimation, and Doppler extraction algorithms for obtaining 4 dimensional point cloud information. This signal chain follows the mmWave SDK DPM structure all the way through tracker and data output. After viewing this design, the customer should be able to better understand the implementation of advanced algorithms on the IWR6843 device with the mmWave SDK architecture.

Table 1. Key Performance Specifications

PERFORMANCE PARAMETER	SHORT RANGE CONFIGURATION
Device	IWR6843AOP, IWR6843ISK, IWR6843ODS
Field of view	120° horizontal, 40° vertical
Maximum range	8.1 meters
Range resolution	8.4 cm
Maximum velocity	16.2 km/h
Velocity resolution	0.324 km/h
System power consumption	2 Watts

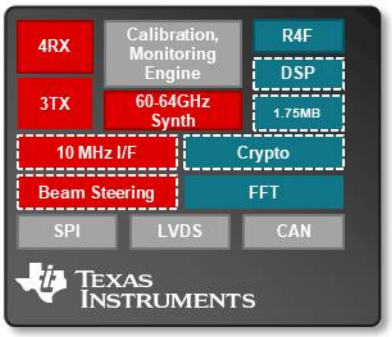
2 System Overview

2.1 Block Diagram

2.1.1 Hardware Block Diagram

The TIDEP-01000 is implemented on the IWR6843 ISK EVM see Figure 1. The EVM is connected to a host PC through a universal asynchronous receiver-transmitter (UART) for visualization.

Figure 1. Hardware Block Diagram

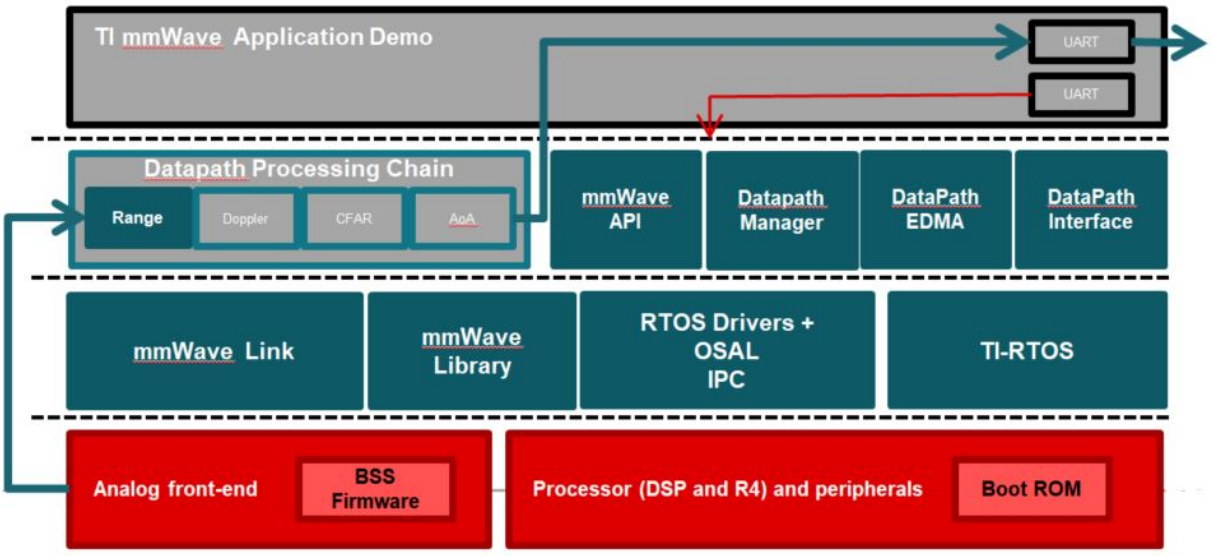


2.1.2 Software Block Diagram

2.1.2.1 mmWave SDK Software Block Diagram

The mmWave software development kit (SDK) enables the development of mmWave sensor applications using the IWR6843 SOC and EVM see Figure 2. The SDK provides a foundational structure for developing radar processing software. This structure includes a Data-Path Manager (DPM) which handles execution of the DataPath - Processing Chain (DPC). The DPC is made of Data Path Units (DPUs). This demo implements custom DPUs within a custom DPC to achieve the people coutning software.

Figure 2. mmWave SDK Software Block Diagram



2.1.2.2 Software Block Diagram of People-Counting Application

As shown in Figure 3, the implementation of the people-counting application demo on the IWR6843 consists of a signal chain running on the C674x DSP, and the tracking module running on the ARM® Cortex®-R4F processor.

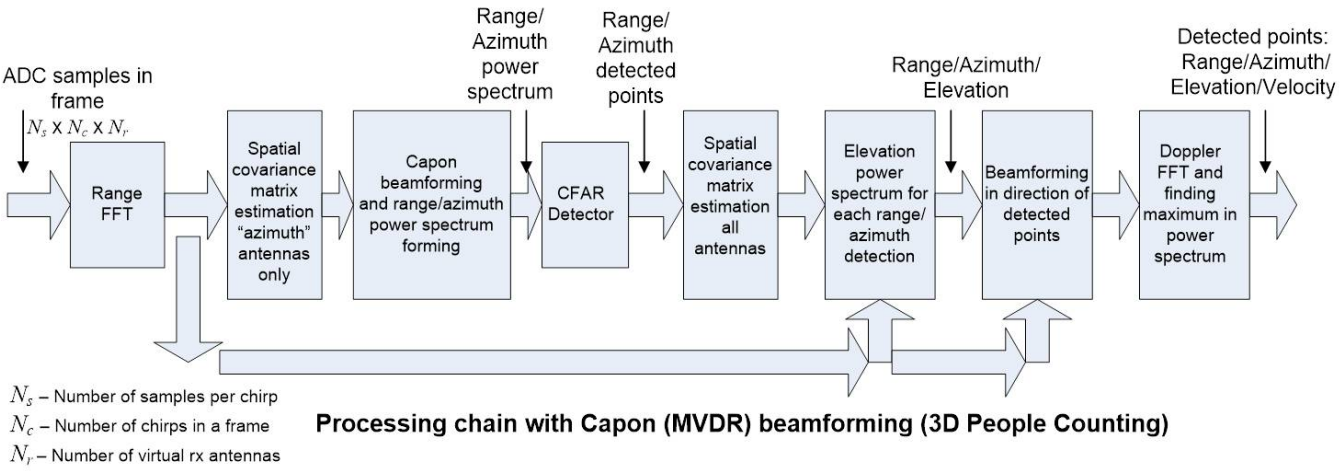
At its core, the demo does two things:

1. Use the radar data to produce a point cloud with each point containing an X, Y and Z coordinate, radial velocity, and SNR (signal to noise ratio)
2. Finds and tracks clusters in the point cloud.

The point cloud generation uses a unique angle of arrival algorithm, called Capon Beamforming, which has only been implemented by TI in this demo. The tracking algorithm used is developed by TI and used in a variety of our applications. The rest of the document will explain how the signal chain and tracker function.

- 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
  - Implemented on HWA and Cortex R4F
- Capon Beamforming (BF):
  - Static clutter removal
  - Covariance matrix generation, angle spectrum generation, and integration is performed
  - Outputs range-angle heat map
  - Implemented on c674 DSP
- CFAR detection algorithm:
  - Two-pass, constant false-alarm rate
  - First pass cell averaging smallest of CFAR-CASO in the range domain, confirmed by second pass cell averaging smallest of CFAR-CASO in the angle domain, to find detection points.
  - Implemented on c674 DSP
- Elevation Beamforming
  - Capon BF algorithm is applied again for each point detected in Range-Azimuth heatmap
  - 1-D Elevation Spectrum is generated and strongest signal is taken as the detected angle
  - Implemented on c674 DSP
- Doppler estimation:
  - For each detected [range, azimuth] pair from the detection module, estimate the Doppler by filtering the range bin using Capon beam-weights, and then run a peak search over the FFT of the filtered range bin.
  - Implemented on c674 DSP
- Tracking:
  - Operates on point cloud
  - Searches for clusters in Cartesian + Doppler Space
  - Predicts movement of clusters to maintain a track of unique objects such as people
  - Output of the tracker is a set of trackable objects with certain properties like position, velocity, physical dimensions, and point density
  - Implemented on Cortex R4F

Figure 3. People Counting Application Block Diagram

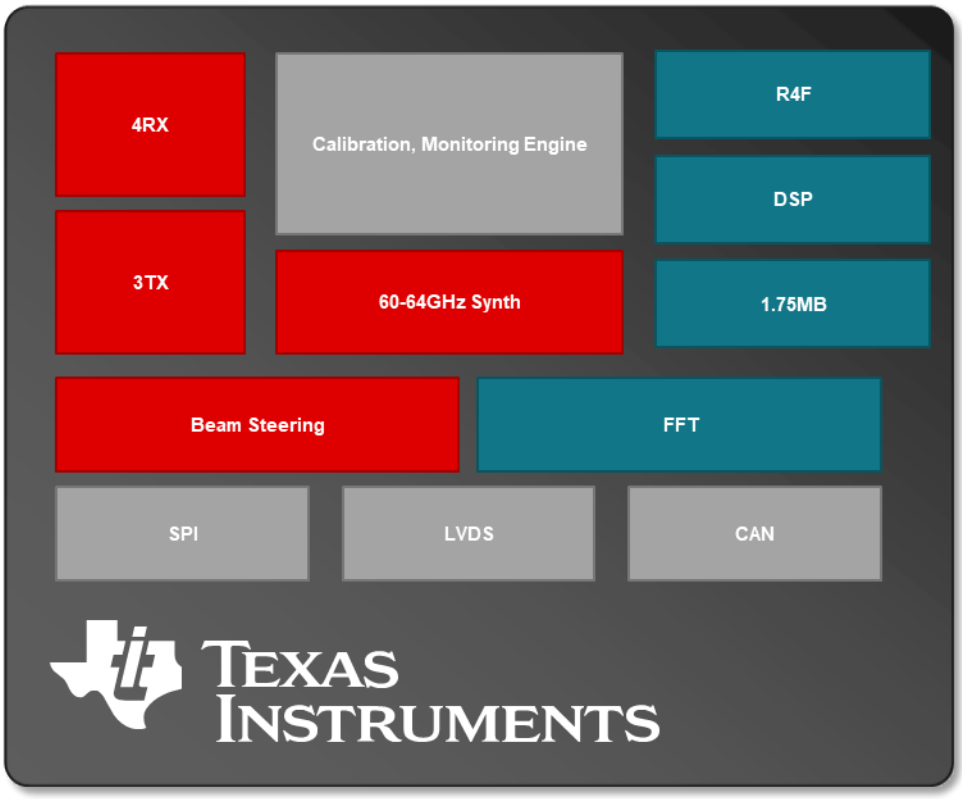


2.2 Highlighted Products

2.2.1 IWR6843

The IWR6843 is an integrated, single-chip, frequency modulated continuous wave (FMCW) sensor capable of operation in the 60 to 64 GHz band. The sensor is built with the low-power, 45nm, RFCMOS process from TI and enables unprecedented levels of integration in an extremely small form factor. The IWR6843 is an ideal solution for low-power, self-monitored, ultra-accurate radar systems in the industrial space.

Figure 4. IWR6843 Block Diagram



The IWR6843 has the following features:

- FMCW transceiver
  - Integrated PLL, transmitter, receiver, baseband, and A2D
  - 60 to 64GHz coverage, with 4GHz available bandwidth
  - Four receive channels
  - Three transmit channels
  - Ultra-accurate chirp (timing) engine based on fractional-N PLL
  - TX power
    - 12 dBm
  - RX noise figure
    - 12 dB (60 to 64 GHz)
  - Phase noise at 1 MHz
    - –92 dBc/Hz (60 to 64 GHz)
- Built-in calibration and self-test (monitoring)
  - ARM Cortex-R4F-based radio control system

  - Built-in firmware (ROM)
  - Self-calibrating system across frequency and temperature
- C674x DSP for FMCW-signal processing
  - On-chip memory: 1.75MB
- Cortex-R4F MCU for object detection, and interface control
  - Supports autonomous mode (loading the user application from QSPI flash memory)
- Integrated peripherals
  - Internal memories with ECC
  - Up to six ADC Channels
  - Up to two SPI Channels
  - Up to two UARTs
  - CAN interface
  - I^2C
  - GPIOs
  - Two-lane LVDS interface for raw ADC data and debug instrumentation

2.2.2 mmWave SDK

The mmWave SDK is the foundational software component of the mmWave projects and includes the following smaller components:

- Drivers
- OSAL
- mmWaveLink
- mmWaveLib
- mmWave API
- Data Path Manager
- BSS firmware
- Board setup and flash utilities

  For more information, see the mmWave SDK user's guide.

2.3 System Design Theory

2.3.1 Use-Case Geometry and Sensor Considerations

The IWR6843 is a radar-based sensor that integrates a fast, FMCW, radar front end, with both an integrated Arm Cortex-R4F MCU and a TI C674x DSP for advanced signal processing. The configuration of the IWR6843 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.

The key performance parameters at issue follow with brief descriptions:

- Maximum Range - maximum range at which a human can be detected
- Range Resolution - minumum distance required between two objects for the radar to detect them as separate objects
- Maximum Velocity - the maximum unambiguous velocity that can be detected. Objects moving faster than this may have incorrect velocity measurements
- Velocity resolution - minimum velocity difference between two objects for the radar to detect them as separate objects

When designing the frame and chirp configuration for a people-counting use case, start by considering the geometry of the scene specifically the range of interest. Table 2 lists example chirp configurations with good range and Doppler resolution and memory for a 8m indoor people counting application.

Table 2. Performance Parameters of Two Example Chirp Designs on the IWR6843

KEY INPUT PARAMETERS	
PERFORMANCE PARAMETER	6-METER RANGE
~~! 60.6~~	
Starting frequency (GHz)	60.75
Maximum range, R_{max} (m)	8.1
Range resolution (cm)	8.4
Maximum velocity (km/h)	16.2
Velocity resolution (km/h)	0.324
Idle time (µs)	30
ADC valid start time (µs)	25
Periodicity (ms)	55
Valid sweep bandwidth (MHz)	1780
Ramp slope (MHz/µs)	57
Sampling frequency (Msps)	2.95
Number of samples per chirp	96
Ramp end time (µs)	62
Chirp repetition period (µs)	184.0
Number of chirps	288
Active frame time (ms)	25.5

2.3.2 Low-Level Processing

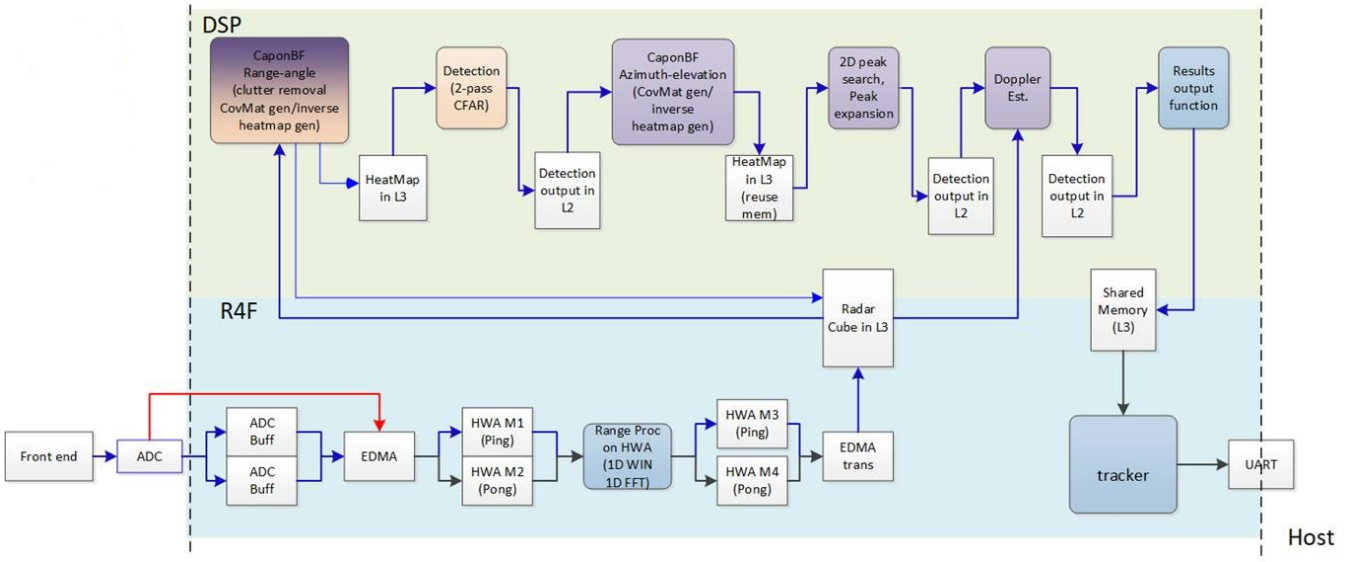
An example of a processing chain for people counting is implemented on the IWR6843 EVM.

The processing chain is implemented on the DSP and Cortex-R4F together. Table 3 lists the several physical memory resources used in the processing chain.

Table 3. Memory Configuration

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	16	Layer one data cache caches data accesses to any other section configured as cacheable. LL2, L3, and HSRAM are configured as cache-able.
L1P SRAM	28	28	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	4	4	Layer one cache caches program accesses to any other section configured as cacheble. LL2, L3, and HSRAM are configured as cache-able.
L2	256	239	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	768	720	Higher latency memory for DSP accesses primarily stores the radar cube and the range-azimuth heat map. It also stored system code not required to be executed at high speed.
HSRAM	32	20	Shared memory buffer used to store slow, non-runtime code.

Figure 5. Processing Chain Flow: Detection Tracking Visualization

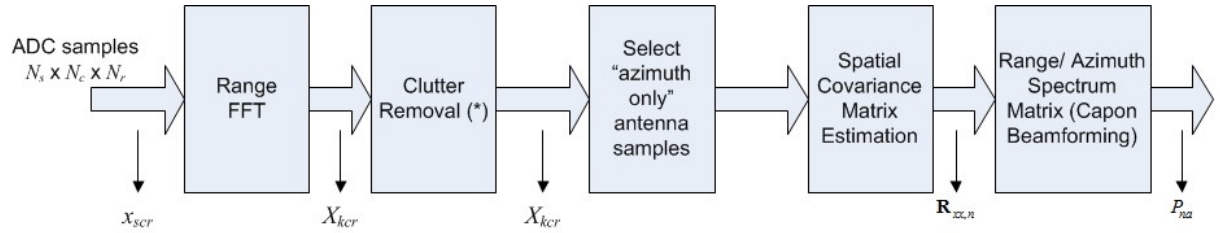


As shown in Figure 5, the implementation of the people-counting example in the signal-processing chain consists of the following blocks implemented on both the DSP and Cortex R4F. In the following section we break this process into the following smaller blocks:

1. Range FFT through Range Azimuth Heatmap with Capon BF
2. Object Detection with CFAR and Elevation Estimation
3. Doppler Estimation

1. Range FFT through Range Azimuth Heatmap with Capon BF
   - As shown in the block diagram, Raw Data is processed with a 1-D FFT (Range Processing) and Static Clutter Removal is applied to the result. Then Capon Beamforming is used to generate a range-azimuth heatmap. These are explained in depth below.

Figure 6. Range FFT through Range-Azimuth Heatmap



N_s - Number of samples per chirp
N_c – Number of chirps
N_r – Number of virtual receive antennas
$N_{r,a}$ – Number of virtual azimuth antennas

- Range processing
  - For each antenna, EDMA is used to move samples from the ADC output buffer to the FFT Hardware Acceleratorrator (HWA), controlled by the Cortex R4F. A 16-bit, fixed-point 1D windowing and 16-bit, fixed-point, 1D FFT are performed. EDMA is used to move output from the HWA 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 occurs each frame, except where noted, during the idle time between the active chirp time and the end of the frame.
- Static Clutter Removal
  - Once the active chirp time of the frame is complete, the interframe processing can begin, starting with static clutter removal. 1D FFT data is averaged across all chirps for a single Virtual Rx antenna. This average is then subtracted from each chirp from the Virtual Rx antenna. This cleanly removes the static information from the signal, leaving only the signals returned from moving objects. The formula is

$$X_{nr} = \frac{1}{N_c}\sum_{c=1}^{Nc} X_{ncr}$$
$$X_{ncr} = X_{ncr} - X_{nr} \quad (1)$$

  - With N_c = Number of chirps; N_r = Number of recieve antennas; X_{nr} = Average samples for a single receive antenna across all chirps; X_{ncr} = Samples of a Single Chirp from a receive antenna
- Capon beamforming
  - The Capon BF algorithm is split into two components: 1) the Spatial Covariance Matrix computation and 2) Range-Azimuth Heatmap Generation. The final output is the Range-Azimuth heatmap with beamweights. This is passed to the CFAR algorithm.
  - Spatial Covariance Matrix is calculated as the following:

- First, the spacial covariance matrix is estimated as an average over the chirps in the frame as $R_{xx,n}$ which is 8x8 for ISK and 4x4 for ODS:

$$R_{xx,n} = \frac{1}{N_c}\sum_{c=1}^{N_c} X_{nc} X_{nc}^H$$
$$X_{nc} = \left[X_{nc1}, \ldots, X_{ncNr}\right]^T \quad (2)$$

- Second, diagonal loading is applied to the R matrix to ensure stability

$$R_{xx,n} = R_{xx,n} + \alpha \frac{tr(R_{xx,n})}{N_r} I_{N_r} \quad (3)$$

– Range-Azimuth Heatmap Generation
  - First, the Range-Azimuth Heatmap P_{na} is calculated using the following equtaions
  - Subscript a indicates values across azimuth bins

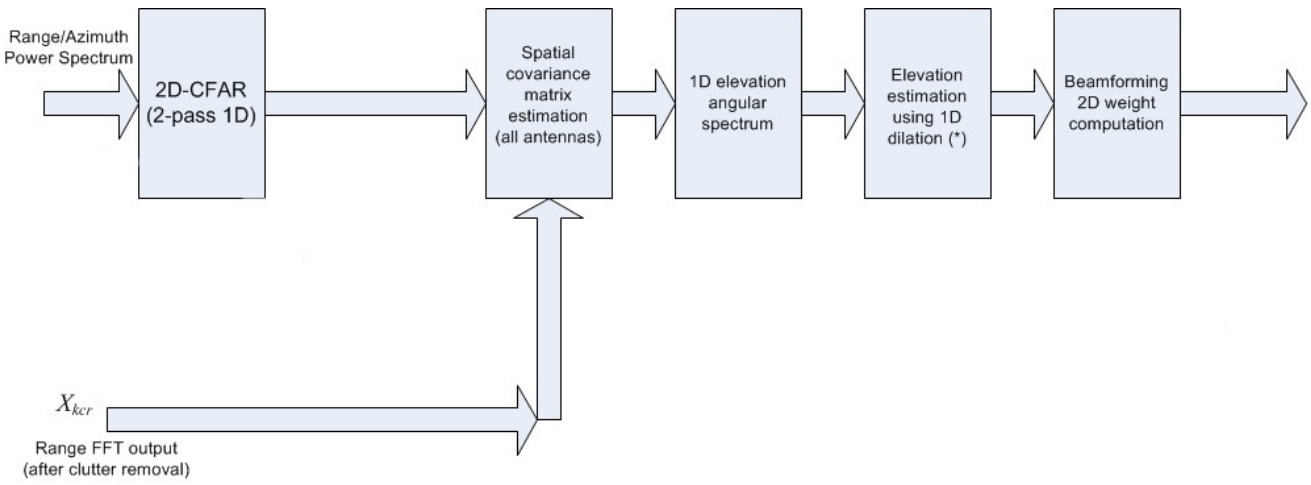
$$P_{na} = \frac{1}{a_a^H R_{xx,n}^{-1} a_a}$$
$$a_a = \left[1, e^{j\mu_a}, \ldots, e^{j(N_r-1)\mu_a}\right]^T$$
$$\mu_a = 2\pi \frac{d}{\lambda}\sin(\theta_a) \quad (4)$$

  - Then, the beamforming weights are calculated as

$$w_a = \frac{R_{xx,n}^{-1} a_a}{a_a^H R_{xx,n}^{-1} a_a} \quad (5)$$

2. Object Detection with CFAR and Elevation Estimation
   - Using the heatmap generated in the above steps, 2 Pass CFAR is used to generated detected points in the Range-Azimuth spectrum. For each detected point, Capon is applied to generate a 1D elevation angular spectrum, which is used to determine the elevation angle of the point

Figure 7. CFAR and Elevation Estimation



- Object detection
  – Two pass CFAR algorithms is used on the range azimuth heat map to perform the object detection using the CFAR "smallest of" method. First pass is done per angle bin along the range domain. Second pass in the angle domain is used confirm the detection from the first pass. The output detected point list is stored in L2 memory.
- Elevation Estimation with Capon BF
  – Full 2D 12 antenna Capon Beamforming is performed at the azimuth of each detected point. This is done following the same steps used to generate the range-azimuth heatmap; 1) generate spacial covariance matrix and 2) generate 1D elevation angle spectrum (similar to the heatmap)

- Then a single peak search is performed to find the elevation angle of each point. This step does not generate new detection points.
- Spatial Covariance matrix is similar to before, with input based on detections

$$R_{xx,m} = \frac{1}{N_c}\sum_{c=1}^{N_c} X_{kc}X_{kc}^H \qquad k = r_{det,m} \tag{6}$$

- With diagonal loading matrix

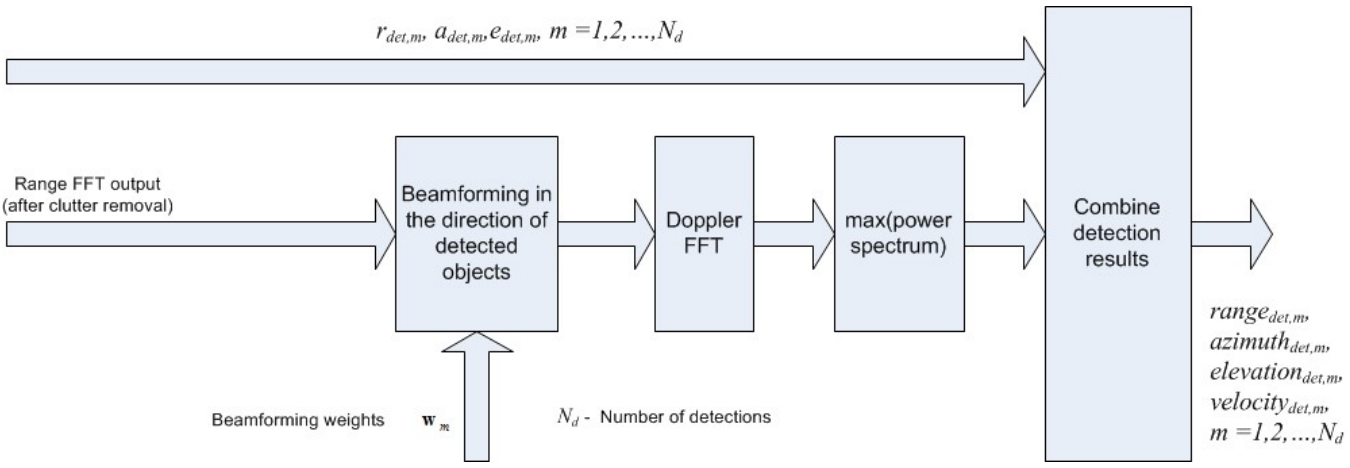
$$R_{xx,m} = R_{xx,m} + \alpha_2 \frac{tr(R_{xx,m})}{N_r} I_{N_r} \tag{7}$$

- 1D Elevation Spectrum is as follows

$$P_m = \frac{1}{a_m^H R_{xx,m}^{-1} a_m}$$
$$a_m = a(\mu_m, \upsilon_m) = a(\mu_m) \otimes a(\upsilon_m)$$
$$a(\mu_m) = [1, \quad e^{j\mu_m}, \ldots, e^{j(N_r-1)\mu_m}]^T$$
$$a(\upsilon_m) = [1, \quad e^{j\upsilon_m}, \ldots, e^{j(N_r-1)\upsilon_m}]^T \tag{8}$$

3. Doppler Estimation
   - For each detected point in range and azimuth(angle) space, Doppler is estimated using the capon beam weights and Doppler FFT. The output is stored in the L2 memory. This output is combined with the point cloud produced during CFAR and Elevation Estimation, resulting in output for each point of:
     - Range
     - Azimuth
     - Elevation
     - Doppler
     - SNR

Figure 8. Doppler Estimation and Combination of Results



All the above processing except the range processing happens during inter-frame time. After DSP finishes frame processing, the results are written in shared memory (L3/HSRAM) for Cortex-R4F to input for the group tracker.

4. Group tracker
   - The tracking algorithm implements the localization processing. Tracker works on the point cloud data from DSP, and provide localization information. Tracker inputs the point cloud data, range, azimuth, elevation, doppler, and SNR for each point in the point cloud. It outputs a list of tracked objects - each object has position, velocity, and acceleration in 3D Cartesian (X, Y, Z) space.

Table 4 lists the results of benchmark data measuring the overall MIPS consumption of the signal processing chain running on the DSP. Time remaining assumes a 50 ms total frame time.

Table 4. MIPS Use Summary

!~USED TIME

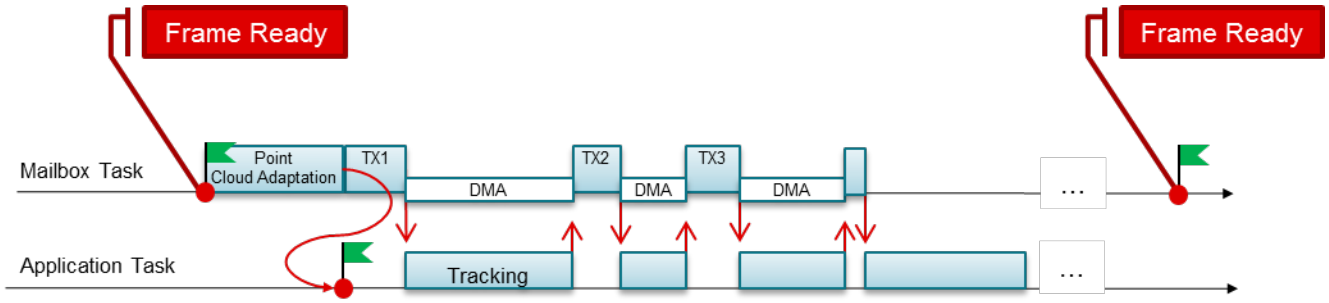
PARAMETER	Time Used (ms)	LOADING (Assuming 50 ms frame time)
!~Active chirp time15%14 µs92 µs7.2 ms		
Active Frame time	13.9	27.8%
Range-Azimuth Heatmap Generation	9.42	18.84%
2 Pass CFAR	1	2%
Elevation Estimation	11.7	23.4%
Doppler Estimation	2.15	4.3%
Total Processing Time	22.335	44.67%
Total Time	36.235	72.47%

2.3.3 High-Level Processing Details

2.3.3.1 DPM Model

Processing on the device is handled by the Data Path Manager (DPM). The DPM coordinates processing on the R4F and DSP. The group tracker (gtrack) is run on Cortex R4F, in a separate task. Gtrack is encapsulated in a DPU, so it can be initialized and configured by the DPM. During runtime, gtrack runs in a separate task. This task pends on a semaphore, which is released when the point cloud is received and stored from the DSP. This allows the Cortex R4F to transmit data on UART or process ADC data through the HWA while the tracker is running on the point cloud from the previous frame. The main task in mss is referred to as a Mailbox Task below.

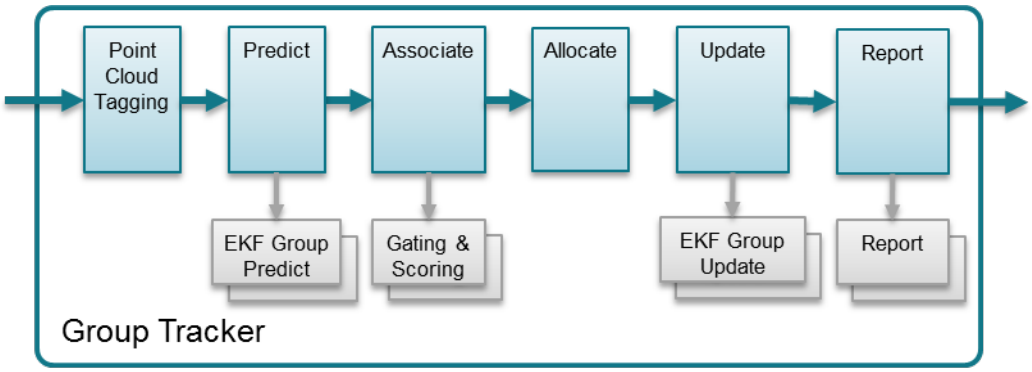
Figure 9. High-Level Processing Task Model



2.3.3.2 Group Tracker

The tracking algorithm is implemented as a library within a DPU. The DPM initalizes and configures the tracker DPU with configuration parameters that describe sensor, scenery, and behavior of radar targets. The algorithm is called once per frame from the application task context. It is possible to create multiple instances of group tracker. Figure 10 shows the steps algorithm goes during each frame call. The algorithm inputs measurement data in polar coordinates (range, azimuth, elevation, Doppler, SNR), and tracks objects in Cartesian space. Therefore, use the extended Kalman filter (EKF) process.

Figure 10. Group Tracking Algorithm



Point cloud input is first tagged based on scene boundaries. Some points may be tagged as *outside the boundaries*, and are ignored in association and allocation processes.

Each iteration of the tracker runs through the following steps:

- If a track exists, use a Kalman filter to predict the tracking group centroid for time n based on state and process covariance matrices, estimated at time n-1. Compute a-priori state and error covariance estimations for each trackable object. At this step, compute measurement vector estimations.
- If a track exists, the association function allows each tracking unit to indicate whether each measurement point is *close enough* (gating), and if it is, to provide the bidding value (scoring). The point is assigned to a highest bidder.
- Any points not assigned to a track go through an allocate function. During the allocation process,

points are first joined into a sets based on their proximity in 3D Cartesian + Doppler coordinates. Each set becomes a candidate for an allocation decision, and must pass a threshold for cumulative SNR and threshold for minimum number of points in the setto become a new track. When passed, the new tracking unit is allocated.

- During the update step, tracks are updated based on the set of associated points. Compute the innovation, Kalman gain, and a-posteriori state vector and error covariance. In addition to classic EKF, the error covariance calculation includes group dispersion in a measurement noise covariance matrix.
- The report function queries each tracking unit and produces the algorithm output.

2.3.3.3 Configuration Parameters

The configuration parameters are used to configure the tracking algorithm. They are adjusted to match the customer use case, based on particular scenery and target characteristics. Parameters are divided into mandatory, and optional (advanced). Mandatory parameters are described in Table 5.

Table 5. Mandatory Configuration Parameters

PARAMETER	DEFAULT	DIM	DESCRIPTION
trackerEnabled	1	-	Enables or disables the tracker. 1 is enabled.
maxNumPoints	1000	—	Maximum number of detection points per frame
maxNumTracks	20	—	Maximum number of targets to track at any given time
maxRadialVelocity	67	m/s * 10	Maximum absolute radial velocity reported by sensor
radialVelocityResolution	105	m/s * 1000	Minimal non-zero radial velocity reported by the sensor
deltaT	50	ms	Frame rate - dependent on chirp configuration

2.3.3.3.1 Advanced Parameters

Advanced parameters are divided into a few sets. Each set can be omitted, and defaults are used by an algorithm. The customer must modify the necessary parameters to achieve better performance.

2.3.3.3.1.1 Scenery Parameters

This set of parameters describes the scene. It allows the user to configure the tracker with expected boundaries and scene entrances. These effect tracker behavior. The tracker does not track clusters outside of the boundaries set by the rightWall and leftWall parameters, and tracker behavior can be tuned differently for the areas defined by the lowerEntrance and upperEntrance parameters. Table 6 lists example parameters.

Table 6. Scenery Parameters

PARAMETER	DEFAULT	DIM	DESCRIPTION
lefWall	–6	m	Position of the left wall, in meters, set to -100 if no wall. Points behind the wall will be ignored
rightWall	6	m	Position of the right wall, in meters, set to 100 if no wall. Points behind the wall will be ignored
Near Entrances	0.5	m	Wall/Entrance closest to EVM. The distance along the Y axis from the EVM where you want to start tracking.
Far Entrance	7	m	Wall/Entrance far from the EVM. The distance along the Y Axis where you want to start/stop tracking.
Floor	-2	m	Position of the ground in meters.
Ceiling	2	m	Position of the ceiling in meters.

2.3.3.3.1.2 Allocation Parameters

The reflection points reported in the point cloud are associated with existing tracking instances. Points that are not associated are subjects for the allocation decision. Each candidate point is clustered into an allocation set. To join the set, each point must be within maxDistance and maxVelThre from the set's centroid. When the set is formed, it must have more than setPointsThre members, and pass the minimal velocity and SNR thresholds. Table 7 lists these parameters.

Table 7. Allocation Parameters

PARAMETER	DEFAULT	DIM	DESCRIPTION
SNR threshold	0	—	Minimum total SNR for the allocation set, linear sum of power ratios
SNR Obscured threshold	800	-	Minimum total SNR if the allocation set is behind another track with respect to the radar module.
Velocity threshold	0.1	m/s	Minimum radial velocity of the allocation set centroid
Points threshold	15	—	Minimum number of points in the allocation set
maxDistanceThre	1	m^2	Maximum squared distance between candidate and centroid to be part of the allocation set
maxVelThre	2	m/s	Maximum velocity difference between candidate and centroid to be part of the allocation set

2.3.3.3.1.3 State Transition Parameters

Each tracking instance can be in either FREE, DETECT, or ACTIVE state. Once per frame, the instance can get HIT (have non-zero points associated to a target instance) or MISS (no points associated) event.

When in FREE state, the transition to DETECT state is made by the allocation decision. See Table 7 for the allocation decision configuration parameters. When in DETECT state, use the det2active threshold for the number of consecutive hits to transition to ACTIVE state, or det2free threshold of number of consecutive misses to transition back to FREE state. When in ACTIVE state, the handling of the MISS (no points associated) is as follows:

- If the target is in the static zone *and* the target motion model is close to static, then assume that the reason for no detection is because they were removed as static clutter. In this case, increment the miss count, and use the static2free threshold to extend the life expectation of the static targets.
- If the target is outside the static zone, then no points were associated because that target is exiting. In this case, use the exit2free threshold to quickly free the exiting targets.
- If the target is in the static zone, but has non-zero motion in radial projection, then the lack of detections occurs when the target is obscured by other targets. In this case, continue target motion according to the model, and use the active2free threshold.
- If the target is in a static zone and is only associated with static points (0 Doppler), then the sleep2Free threshold is used.

Table 8 lists the parameters used to set this behavior.

Table 8. State Transition Parameters

PARAMETER	DEFAULT	DIM	DESCRIPTION
det2activeThre	10	—	In DETECT state; how many consecutive HIT events needed to transition to ACTIVE state
det2freeThre	5	—	In DETECT state; how many consecutive MISS events needed to transition to FREE state
active2freeThre	10	—	In ACTIVE state and NORMAL condition; how many consecutive MISS events needed to transition to FREE state
static2freeThre	100	—	In ACTIVE state and STATIC condition; how many consecutive MISS events needed to transition to FREE state
exit2freeThre	5	—	In ACTIVE state and EXIT condition; how many consecutive MISS events needed to transition to FREE state
sleep2freeThre	600	-	In ACTIVE state and sleep condition, how many misses are needed to transition to FREE state.

2.3.3.3.1.4 Gating Parameters

The gating parameters set is used in the association process to provide a boundary for the points that can be associated with a given track. These parameters are target-specific. Table 9 lists each of these parameters.

Table 9. Gating Function Parameters

PARAMETER	DEFAULT	DIM	DESCRIPTION
Gain	3	—	Gating gain
LengthLimit	1.5	m	Gating limit in length
WidthLimit	1.5	m	Gating limit in width
HeightLimit	2	m	Gating limit in height
VelocityLimit	20	m/s	Gating limit in velocity (0 – no limit)

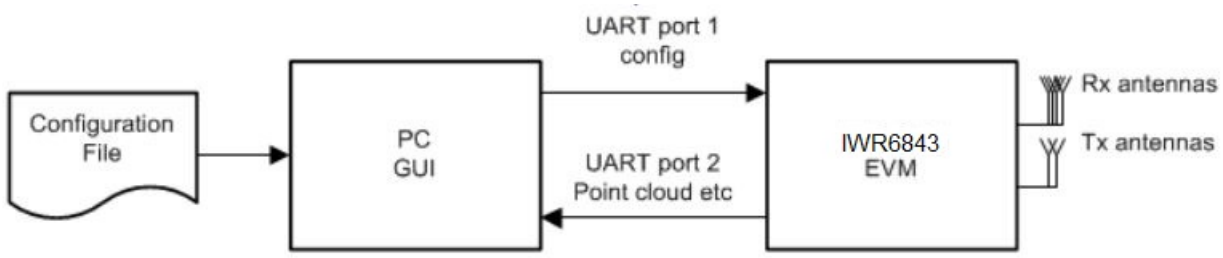
The gating gain is a factor by which the gating volume can be increased to search for points to associate with the track. The gating volume can be estimated as the volume of the ellipsoid, computed as $V = \frac{4\pi}{3}abc$, where a, b, and c are the expected target dimensions in range (m), angle (rad), and doppler (m/s).

For example, consider a person as a radar target. For the target center, we could want to reach ±0.45 m in length (a = 0.9), ±3 degree in azimuth (b = 6π / 180), and ±5.18 m/s in radial velocity (c = 10.36), resulting in a volume of approximately 4.

In addition to setting the volume of the gating ellipsoid, the limits can be imposed to protect the ellipsoid from overstretching. The limits are the function of the geometry and motion of the expected targets. For example, setting the WidthLimit to 8 m does not allow the gating function to stretch beyond 8 m in width.

2.3.4 Output Through UART

Figure 11. IWR6843 UART Communication



As shown in Figure 11, the example processing chain uses one UART port to receive input configuration from 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 software package for details on the format of the input configuration and output results.

3 Hardware, Software, Testing Requirements, and Test Results

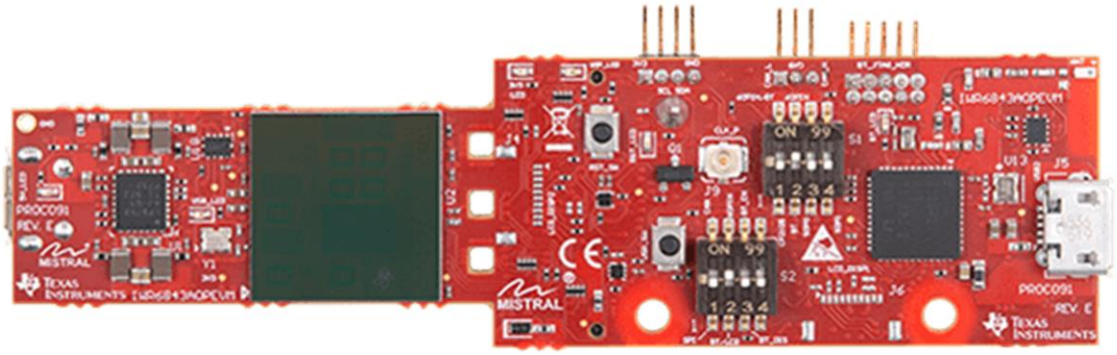
3.1 Required Hardware and Software

3.1.1 Hardware

The user has a choice of any mmWave IWR6843 Device. We provide support for the

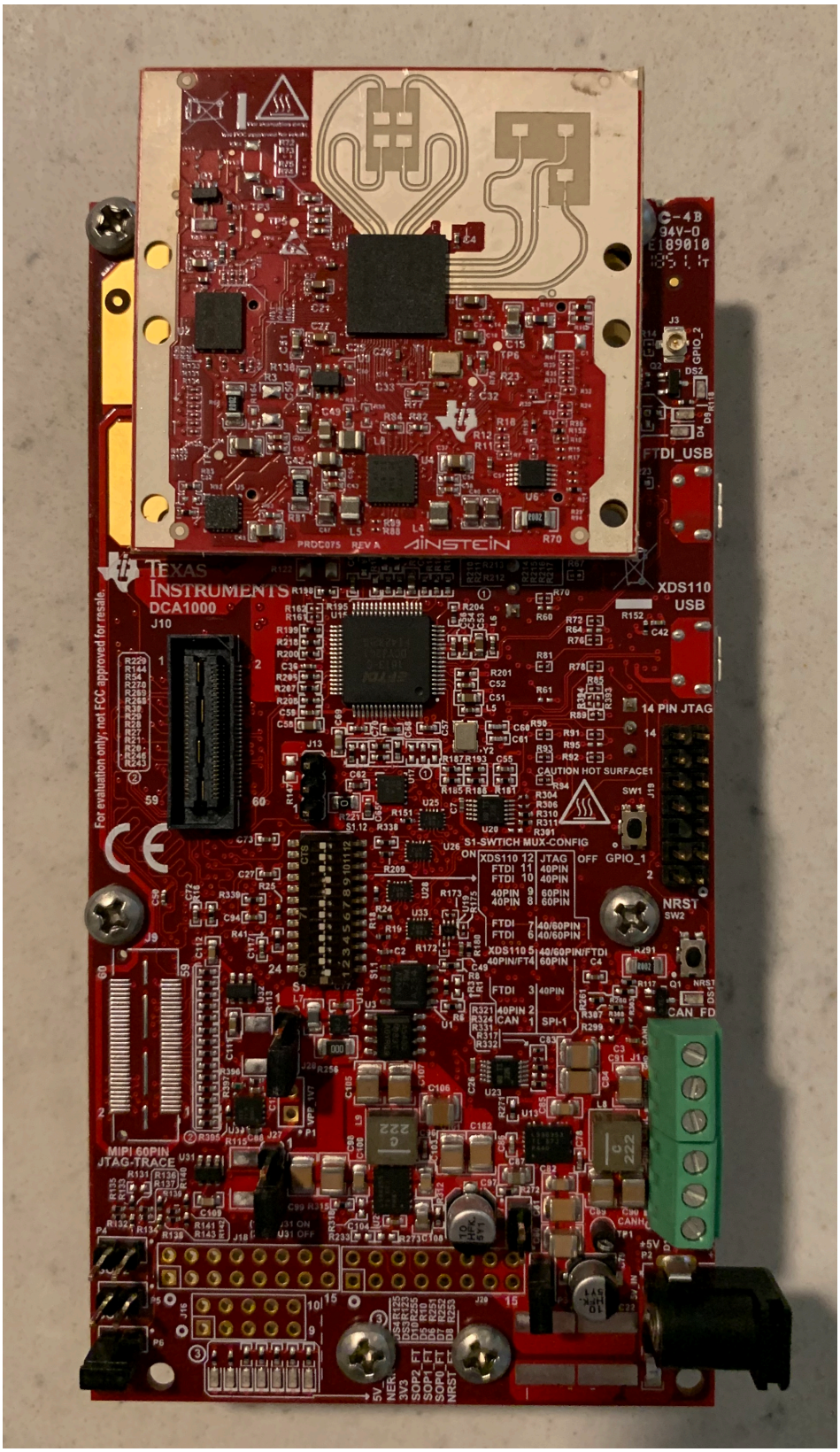
- IWR6843 Intelligent mmWave Sensor Antenna-On-Package (AoP) Evaluation Module (Antenna on Package)

Figure 12. IWR6843 Intelligent mmWave Sensor Antenna-On-Package (AoP) Evaluation Module Board Image



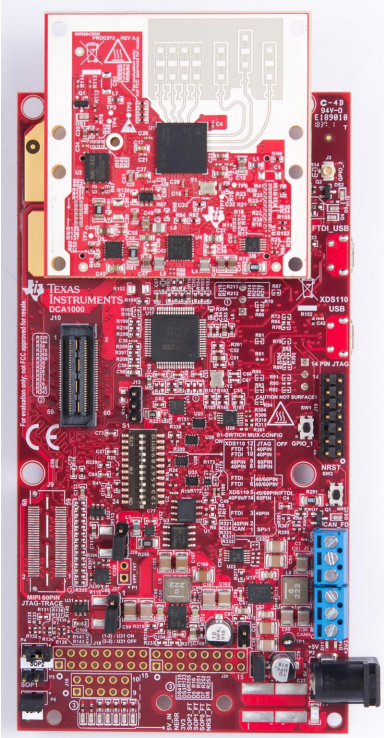
- IWR6843 Intelligent mmWave Overhead Detection Sensor (ODS) Antenna Plug-In Module

Figure 13. WR6843 Intelligent mmWave Overhead Detection Sensor (ODS) Antenna Plug-In Module Board Image



- IWR6843 ISK

Figure 14. IWR6843ISK EVM



3.1.2 Software

- mmWave SDK
- People-Counting Application Project

3.2 Testing and Results

This section will discuss the performance testing technique and results. During the development of the demo, TI developed a new technique to measure the accuracy of the tracker. As a result, the tracker accuracy values will be very different and much more detailed than previously published results for other TI People Counting demos.

3.2.1 Test Setup

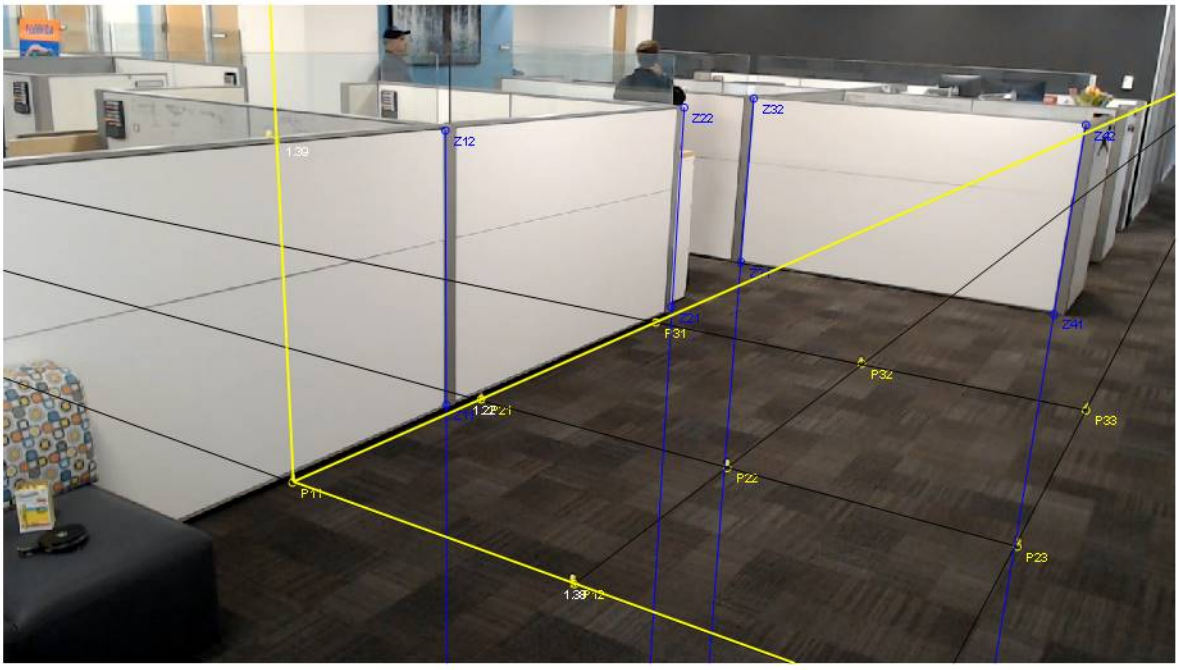
To determine the real performance of the People Counting and Tracking software, test scenarios and test areas have to be created so that the absolute location of people in the room can be compared to the reported location of people in the room. Here, reported location refers to the location reported by the mmWave device. This can be accomplished by carefully marking the test area, then recording each test. This is done as follows:

3.2.1.1 Marking the test environment.

The scene must be marked so that video analysis can accurately estimate the location of people in the room. Once the test environment is decided upon, the tester must create a 3x3 grid on the ground of highly visible markers. The markers should form parallel lines. Setting one of these markers as the origin with location (0, 0, 0), the position of the other 8 markers should be measured. Once the ground plane is complete, another set of markers must be placed above the ground plane. In our example, we use 4 markers, placed directly above markers on the ground plane, so that the X and Y coordinates are the same. Otherwise, the location of the ground directly below these elevated markers must be known. The final scene should look something like this. Note the bright yellow dots on the ground.

Once the markers are placed, the camera position must be selected to see all of the points and the whole intended test area. Once placed, the camera should not be moved. With the location of all markers and the camera known, vanishing lines can be drawn, and the position of people in the scene can be determined.

Figure 15. Processing Marked Locations in a Scene



3.2.1.2 Recording the Test Vector

For each test vector, a camera is used to record the base truth values in the scene, and radar data is captured with the IWR6843ISK and DCA1000. Generally test vectors are 1 to 2 minutres in lenght. During this process, the location and angle of the camera and radar must not change. These should be saved together for later analysis.

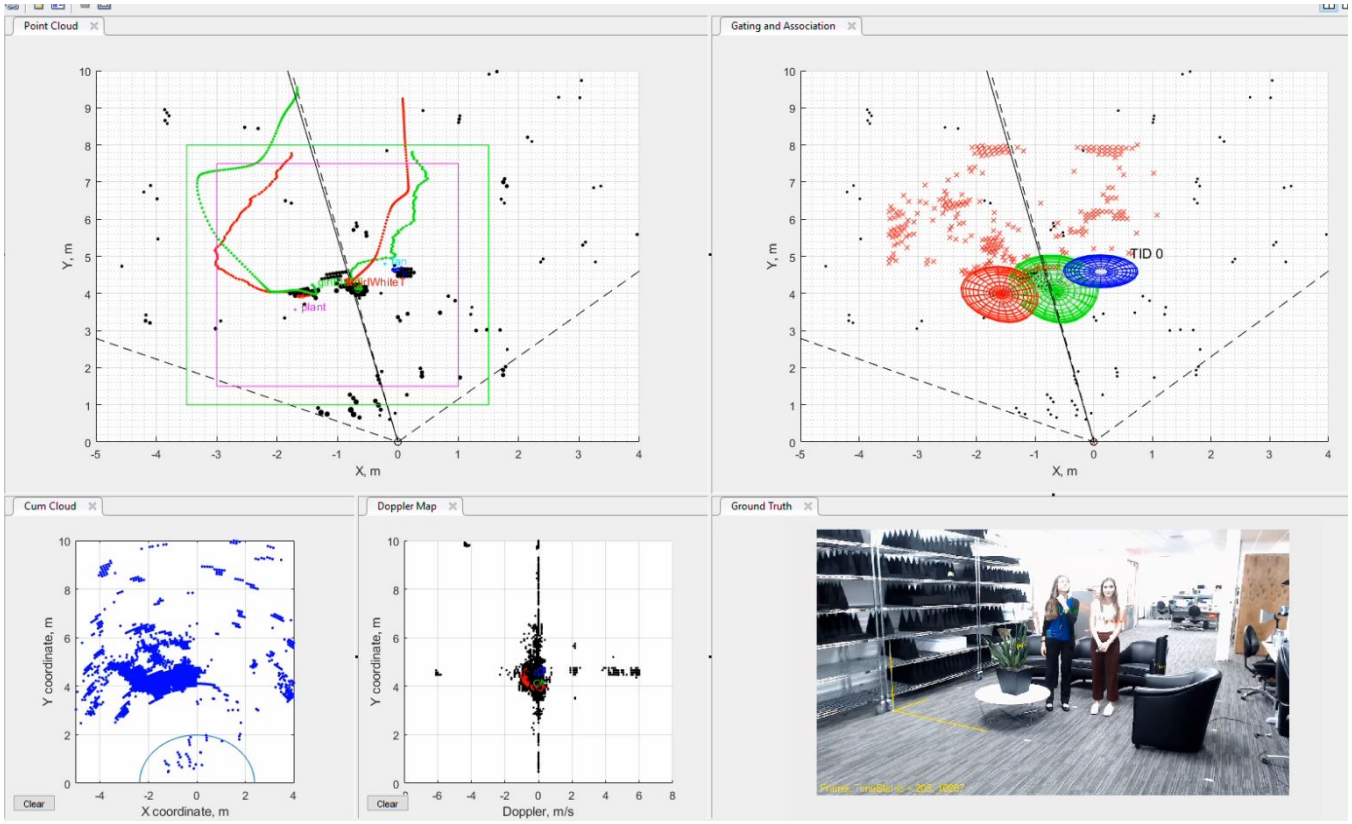
3.2.1.3 Analyzing the Test Data

Once all of the data is collected, the location of each person can be determined from the video footage, and compared to the tracker output. To determine location from the video, the location of two pixels must be know

1. The pixel near the center of the person's chest
2. The pixel where the person's feet touch the ground, below their chest

TI has manually marked the locations as well as algorithmically marked the locations of people. The test results in Figure 16 use data marked by a human.

Figure 16. Data Analysis Example



The test results contain 5 metrics. These include:

- Good Frame Rate (GoodFR or GFR) - this is the percentage of frames with no errors. If you capture 100 frames and a person is missed in 7 frames, the GFR is 93%. If 1 person is missed in 6 frames and 2 people are missed in 1 frame giving 7 frames total with errors, the GFR is still 93%.
- GoodFR1 - Good Frame Rate allowing 1 error. If you collect 100 frames, with 7 frames where exactly 1 person is not tracked, GFR1 is 100% while GFR is 93%
- GoodFR2 - Good Frame Rate allowing 2 errors.
- Miss Detection Rate (MDR) - ratio of missed people vs the total number of people. First, each time a target is present but not detected, increment the miss. You can have more than one miss per frame. Divide this value by the sum of targets in all frames to get MDR
- False Detection Rate (FDR) - ratio of false detection vs the total number of targets detected by the radar module. Each detected target not associated with a real target increments this counter. Divide the counter by the total number of detected targets summed across frames to get this value.

- Positional Error - this is the average error between the absolute position and measured position in meters.

3.2.2 Test Description

Tracker performance was measured across 14 different test vectors in two different scenes.

- Open Area
- Conference Table

In the scenes, people walked, stood, and sat at different locations. There is a complete description in Table 10. We started at test vector three as test 1 and 2 are used to ensure proper setup.

Figure 17. Example Test Scenario

Table 10. Example Test Scenario Data

Test Number	Test Description	Total People
3	Single person enters FOV from the boresight, approaching radially, makes an "8", makes and "O" (~half the FoV), and leaves the FOV	1
4	Single person enters FOV from the boresight, approaching radially, stops at 3m distance for 10s, turns and leaves the FOV	1
5	Two persons entering separately, coming closer to each other, standing still, leaving the scene separetly	2
6	Two persons entering together, getting separated by about 1m, standing still, leaving the scene separetly	2
7	Two persons entering together, coming to about the same spot, standing still, leaving the scene separately	2

Table 10. Example Test Scenario Data (continued)

8	Two persons entering together, coming to about the same spot, standing still, leaving the scene together	2
9	Single person enters FOV from the side, moving tangetially, back and forward, then leaving the scene	1
10	5 persons entering the scene, standing still at approx 1m appart in a group for 10s or so, and then leaving	5
11	Single person enters FOV from the boresight, approaching radially, makes rounds around the table and leaves the FOV	1
12	Single person enters FOV from the boresight, approaching radially, sits at 3m distance for 10s, stands, and leaves the FOV	1
13	3 persons entering the scene, coming to a table, talking, taking sits, all leaving	3
14	4 persons entering the scene, coming to a table, talking, taking sits, 3 leaving	4
15	5 persons entering the scene, coming to a table, talking, taking sits, 3 leaving	5
16	6 persons entering the scene, coming to a table, talking, taking sits, 4 leaving	6

3.2.2.1 Test Results

See Table 11 for testing results.

Table 11. Test Results

Test Number	Number of People	GFR (%)	GFR1 (%)	GFR2 (%)	MDR (%)	FDR (%)	Postional Error (m)
3	1	99.22	NA	NA	1.24	0	0.31
4	1	99.84	NA	NA	0.21	0	0.18
5	2	61.56	100	NA	24.97	0	0.18
6	2	97.50	100	NA	0.46	1.35	0.20
7	2	97.19	100	NA	1.10	0.70	0.23
8	2	99.84	100	NA	0.11	0.00	0.21
9	1	98.59	NA	NA	2.33	0.00	0.24
10	5	29.06	52.81	100	31.85	0.19	0.29
11	1	99.06	NA	NA	1.19	0.00	0.35
12	1	79.38	NA	NA	23.57	0.00	0.35
13	3	40.16	62.19	78.28	47.31	0.82	0.34
14	4	33.44	78.28	99.53	24.68	0.00	0.25
15	5	79.69	87.19	100	8.02	0.00	0.22
16	6	41.41	60.63	69.06	25.16	0.00	0.28

From these results, we can see that the software performs well when the targets are moving. The static clutter removal algorithm discussed previously removes any noisy points generated from non-human objects that don't interest us. This leaves each person as a singular, isolated cluster of points. As we add more people to the scene, the tracker accuracy slowly decreases, due to complexities introduced in the point cloud when multiple people interact with eachother. The tracker performance is also poor when many people in the room are sitting, as they become static and are no longer detected by the point cloud. This can be rectified either through point cloud improvements, tracker improvements, or both. Otherwise, the people counting and tracking algorithm shows that the mmWave IWR6843 device is more than capable of tracking people in complex, indoor environments.

4 Design Files

4.1 Schematics

To download the schematics, see the design files at IWR6843.

4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at IWR6843.

4.3 Altium Project

To download the Altium Designer® project files, see the design files at IWR6843.

5 Software Files

To download the software files, see the design files at IWR6843.

6 Related Documentation

1. Texas Instruments, *IWR6843 Data Sheet*
2. Texas Instruments, *IWR68xx/16xx/14xx Industrial Radar Family*, technical reference manual
3. Texas Instruments, *mmWave SDK*, tools folder

6.1 Trademarks

TI E2E is a trademark of Texas Instruments.
Altium Designer is a registered trademark of Altium LLC.
ARM, Cortex are registered trademarks of Arm Limited.
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 C Revision (November 2019) to D Revision	Page
• Added in 3D space	1
• Changed 16m to 15m	1
• Changed greater than 90% to of +- 10cm	1
• Changed 3 people to 1 person	1
• Added ODS + ICB, and AOP (Antenna on Package)	1
• Added 3D People Counting TI Resource Explorer Folder	1
• Changed /ICB Bundle to Board	1
• Deleted Ideal for Environmental Effects	1
• Added and Elevation field of View of 44 Degrees	1
• Changed 16m to 15m	1
• Deleted Which can Extend to 14 m with Different Chirp Configuration	1
• Changed Azimuth to Approximate azimuth	1
• Changed 44 degrees to 40 degrees	1
• Changed Status Clutter to capon beamforming signal chain	1
• Changed 16 to 15	2
• Changed *counting* to *tracking*	2
• Deleted count	2
• Added appliance control	2
• Changed *Radars allow an accurate measurement of distances, relative velocities of people, and other objects* to *Radar provides accurate position and velocity measurement of people in people in the area of interest.*	2
• Added elevation estimation,	2
• Added 4 dimensional	2
• Added The signal chain follows the mmWave SDK DPM structure	2
• Changed IWR6843 ISK to IWR6843ISK	2
• Added *and IWR6843ISK-ODS*	2
• Changed *...structure. This information is passed on to the group tracker for object localization.* to *...structure all the way through tracker and data output.*	2
• Added with the mmWave SDK architecture	2
• Changed 30° to 40°	2
• Changed 6.3 to 8.1	2
• Changed 5.5 to 8.4	2
• Changed 23.6 to 16.2	2
• Changed 0.3246 to 0.324	2
• Deleted foundational components that help end users focus on their applications. The SDK also 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 on top of the SDK package	3
• Added a foundational structure for developing radar processing software. This structure includes a Data-Path Manager (DPM) which handles execution of the DataPath - Processing Chain (DPC). The DPC is made of Data Path Units (DPUs). This demo implements custom DPUs within a custom DPC to achieve the people coutning software.	3
• Added Implemented on HWA and Cortex R4E	4
• Changed beam forming to Beamforming (BF)	4
• Changed inverse angle spectrum generation to angle spectrum generation	4
• Added Implemented on c674 DSP	4
• Added Implemented on c674 DSP	4
• Deleted Perform target localization, and report the results.	4
• Added Implemented on Cortex R4E	4
• Deleted The mmWave SDK is divided into two broad components: mmWave Suite and mmWave Demos.	7
• Added The mmWave SDK is the foundational software components of the mmWave projects and includes the following smaller components	7

- Added components 7
- Added For more information, see the mmWave SDK user's guide. 7
- Deleted mmWave processing demonstration mmWave Demos ADC data capture demonstration and ADC data streaming demonstration from the SDK provides a suite of demonstrations that depict the various control and data processing aspects of an mmWave application. Data visualization of the output of a demonstration on a PC is provided as part of these demonstrations 7
- Deleted sentence in first bulleted list 8
- Changed 6 m and 14 m to 8m 9
- Changed 60.6 to 60.75 9
- Changed 6.3 to 8.1 9
- Changed 5.5 to 8.4 9
- Changed 23.6 to 16.2 9
- Changed 0.36 to 0.324 9
- Changed 10 to 25 9
- Changed 50 to 55 9
- Changed 2713.66 to 1780 9
- Changed 53 to 57 9
- Changed 2.50 to 2.95 9
- Changed 128 to 96 9
- Changed 128 to 288 9
- Changed 2.355 to 2.55 9
- Changed 24 to 28 10
- Changed 176 to 239 10
- Changed 640 to 768 10
- Changed 600 to 720 10
- Added *Doppler power map* to *-azimuth heat map.* 10
- Changed *It is a less time sensitive program. Data can also be stored here* to *It also stored system code not required to be executed at high speed.* 10
- Changed *Currently unused* to *20* 10
- Changed *Shared memory buffer between the DSP and the R4F relays visualization data to the R4F for output over the UART in this design* to *Shared memory bugger used to store slow, non-runtime code.* 10
- Changed *...implemented as DSP code executing on the C674x core in the IWR6843.*to *...implemented on both the DSP and Cortex R4F. In the following section, we break this process into the following smaller blocks:* 11
- Changed *Range Azimuth*to *Range FFT through Range Azimuth* 11
- Added image title *Range FFT through Range-Azimuth Heatmap* 11
- Changed *local memory of the DSP* to *FFT Hardware Accelerator (HWA) controlled by the Cortex R4F.* 11
- Changed DSP to HWA 11
- Changed *1D FFT data is averaged then subtracted from each sample.*to *Once the active chirp time of the frame is complete, the interframe processing can begin, starting with static clutter removal. 1D FFT data is averaged across all chirps for a single Virtual Rx antenna. This average is then subtracted from each chirp from the Virtual Rx antenna.* ... 11
- Added With N_s = Number of samples per chirp; N_c = Number of chirps; N_r = Number of recieve antennas; X_{nr} = Average sample for a single recieve antenna across all chirps; X_{ncr} = Single Chirp from a receive antenna 11
- Changed Single Chirp... to Samples of a Single Chirp... 11
- Changed Capon beam former to Capon beam forming 11
- Added The Capon BF algorithm is split into two components: 1) the Spatial Covariance Matrix and 2) Range-Azimuth Heatmap Generation. The final output is the Range-Azimuth heatmap with beamweights. This is passed to the CFAR algorithm. 11
- Added computation 11
- Deleted Let s(t) be the incoming waves after mixing to baseband. The sensor array signal to be processed is given by: X(t) = A(θ)s(t) + n(t). Where: A(θ) = (a(θ 1), …a(θ M)) is the steering matrix. a(θ) = (e j2πy 1 sin(θ) , …, e j2πy N sin(θ)) is the steering vector. M is the number of angle bins. y n is the sensor position normalized by wavelength. The Capon BF approach is: θcapon = argminθ{trace(A(θ) × Rn-1 × A(θ)H }, where Rn is the spatial covariance matrix. 11
- Added na 12
- Added Then, the beamforming weights are calculated as 12
- Added with CFAR 12
- Added image title *CFAR and Elevation Estimation* 12

- Added using the CFAR smallest of method. 12
- Added This output is combined with the point cloud produced during CFAR and Elevation Estimation, resulting in output for each point of: 13
- Added image title *Doppler Estimation and Combination of Results* 13
- Deleted which can be used by classification layers (currently not implemented in this example of a people counting application). 13
- Added range, azimuth, elevation, doppler, and SNR for each point in the point cloud. It outputs a list of tracked objects - each object has position, velocity, and acceleration in 3D Cartesian (X, Y, Z) space. 13
- Deleted performs target localization, and reports the results (a target list). therefore, the ourpur of the tracker is a set of trackable objects with certain properties (like position, velocity, physical dimensions, point density, and other features. 13
- Changed ...and memory consumption of the processing chain, up to and including the tracking. to ...consumption of the signal processing chain running on the DSP. Time remaining assmumes a 50 ms total frame time. 13
- Changed MIPS Use Summary table and added the following rows Range-Azimuth Heatmap Generation, 2 Pass CFAR, Elevation Estimation, Doppler Estimation, Total Processing Time, and Total Time 14
- Changed *Task Model* to *DPM Model* 15
- Deleted High-level processing is implemented with two tasks: higher priority mailbox task, and lower priority application task. When the system is configured, the mailbox task is pending on a semaphore, waiting for the frame ready message from DSP. When awakened, the mailbox task copies the relevant point cloud data from the shared memory into TCM, and posts the semaphore to an application task to run. It then creates the transport frame header, and initiates a DMA process for each part (TLV) of the frame. While DMA started sending data over UART, the mailbox task yields to the lower priority application task. When the DMA process completes, additional DMA can be scheduled (for example, TM2 and TM3). To achieve parallelism between the task processing and DMA, the transmit task sends the current (Nth) point cloud TLV with the previous (N-1)th target list and target index TLVs. 15
- Added within a DPU 15
- Changed *The application task creates an algorithm instance...* to *The DPM initializes and configures the tracker DPU...* 15
- Changed *angle, Doppler...* tto *azimuth, elevation, Doppler, SNR...* 15
- Added Each iteration of the tracker runs through the following steps: 15
- Changed *The predict function estimates the...* to *If a track exists, use a Kalman filter to predict the tracking...* 15
- Changed *The association function...* to *If a track exists, the association...* 15
- Changed *Points not assigned go through...* to *Any points not assigned to a track go through...* 15
- Changed *measurement* to *3D Cartesian + Doppler* 15
- Changed *...multiple tests to become* to *a threshold for cumulative SNR and threshold for minimum number of points in the set to become a new track.* 15
- Added list item 16
- Added trackerEnabled 16
- Added 1 16
- Added - 16
- Added Enables or disables the tracker. 1 is enabled. 16
- Changed 250 to 1000 16
- Changed NA to 67 16
- Added * 10 16
- Changed NA to 105 16
- Added * 1000 16
- Added - dependent on chirp configuration 16
- Changed -1.5 to -6 16
- Changed -1.5 to 6 16
- Changed lower Entrance to Near Entrances 16
- Changed 1 to 0.5 16
- Changed *Entrance area lower boundary, in meters, set to) if not defined* to *Wall/Entrance closest to EVM. The distance along the Y axis from the EVM where you want to start tracking.* 16
- Changed upperEntrance to Far Entrance 16
- Changed 4.5 to 7 16
- Changed *Entrance areas lower boundary, in meters, set to 100 if not defined.* to *Wall/Entrance far from the EVM. The distance along the Y Axis where you want to start/stop tracking.* 16
- Added two rows to the table (Floor and Ceiling) 16
- Changed 100 to 0 17
- Added table row: SNR Obscured threshold 17

- Changed 5 to 15 17
- Added If the target is in a static zone and is only associated with static points (0 Doppler), then the sleep2Free threshold is used. 17
- Added table row: sleep2freeThre 17
- Changed Volume to Gain 18
- Changed 4 to 3 18
- Changed 3 to 1.5 18
- Changed 2 to 1.5 18
- Added table row: HeightLimit 18
- Changed 0 to 20 18
- Added *The gating gain is a factor by which the gating volume can be increased to search for points to associate with the track.* 18
- Changed range to length 18
- Added This lab can also run on the ODSISK or AOP module. 21
- Changed *IWR6843 EVM*to *IWR6843ISK EVM* 21
- Added This section will discuss the performance testing technique and results. During the development of the demo, TI developed a new technique to measure the accuracy of the tracker. As a result, the tracker accuracy values will be very different and much more detailed than previously published results for other TI People Counting demos. 22
- Deleted The people counting system was set up for five different test cases, to characterize and demonstrate the capabilities of the system. 22
- Deleted This test scenario shows the ability to change the detection and tracking range of the people counting system by modifying the chirp configuration used. A person is shown to be detected at 14 m, and the history of the person's movement is tracked as he moves through the scene. 26
- Added table column *Number of People* 26
- Changed all 100's to NA in this table 26
- Added From these results, we can see that the software performs well when the targets are moving. The static clutter removal algorithm discussed previously removes any noisy points generated from non-human objects that don't interest us. This leaves each person as a singular, isolated cluster of points. As we add more people to the scene, the tracker accuracy slowly decreases, due to complexities introduced in the point cloud when multiple people interact with eachother. The tracker performance is also poor when many people in the room are sitting, as they become static and are no longer detected by the point cloud. This can be rectified either through point cloud improvements, tracker improvements, or both. Otherwise, the people counting and tracking algorithm shows that the mmWave IWR6843 device is more than capable of tracking people in complex, indoor environments. 26

Changes from B Revision (November 2018) to C Revision **Page**

- Changed 57 GHz to 60 GHz in Section 1 2
- Changed 57 GHz to 60 GHz throughout Section 2.2.1 6
- Changed 7-GHz available bandwidth to 4-GHz available bandwidth in Section 2.2.1 6
- Changed 84 GHz to 64 GHz in Section 2.2.1 6
- Deleted NS - Number of samples per chirp; 11

Changes from A Revision (April 2018) to B Revision **Page**

- Changed all instances of IWR1642 to IWR6843 1
- Changed all instances of 77 GHz to 60 GHz in reference to the IWR6843 1
- Changed information in Description 1
- Added note regarding IWR1642 in 2T Mode. 2
- Changed information in Section 1 2
- Changed data in Table 1 2
- Changed information in Section 2.1.2.2 4
- Changed information in Section 2.2.1 6
- Changed data in Table 2 9

Changes from Original (March 2018) to A Revision	Page

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