Setup Guide for AWR6843 Child Presence Detection w/Classification

Summary

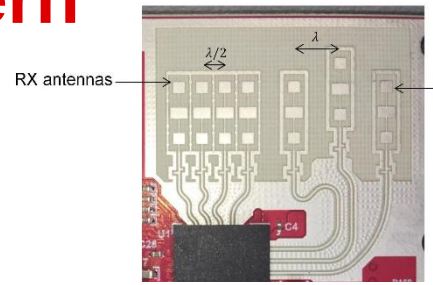
- The child presence detection sensing demo is designed for the following TI 60GHz mmWave devices
  - AWR6843 : https://www.ti.com/product/AWR6843
  - AWR6843AOP: https://www.ti.com/product/AWR6843AOP

  And can be downloaded at: My secure SW link
- In this presentation,
  - We will show how to use the child presence detection sensing demo with sensor in different mounting position and different antenna patterns inside a car/vehicle
  - All the CLI commands listed in this presentation are programmable through configuration file.

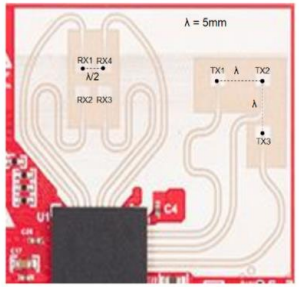
Configuration to Support Different Antenna Patterns

Support Different Antenna Pattern

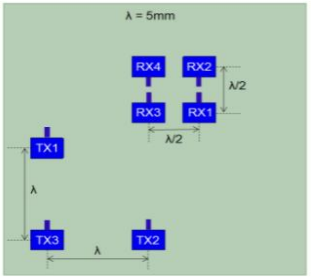
- The CPD w/Classification demo has built in the flexibility to support different antenna patterns. The antenna pattern is input through the following CLI commands
  - antGeometry0
  - antGeometry1
  - antPhaseRot
- Antenna geometry parameters antGeometry0 and antGeometry1 define the virtual antenna's physical location index (0, -1, -2, …) in the azimuth and elevation domains, respectively.
- The phase rotation parameter antPhaseRot defines the phase rotation introduced in the board design.
  - For example, on AWR6843 ISK board, there is no phase rotation between different antennas; on ODS and AOP board, some antenna patterns are fed from the opposite side as compared to others. Therefore, the phase rotation of 180 degrees (i.e., -1) is needed to apply.
- Next, we will provide examples to show how users can derive this values for different antenna pattern.



AWR6843 ISK



AWR6843 ODS



AWR6843 AOP

Support AWR6843 AOP Antenna

The Virtual antenna array for AOP is listed blow.

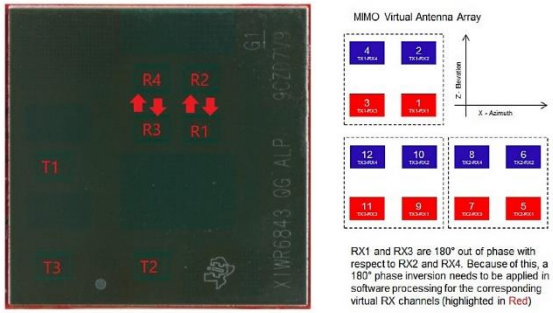
TX1-RX4 TX1-RX2
TX1-RX3 TX1-RX1
TX3-RX4 TX3-RX2 TX2-RX4 TX2-RX2
TX3-RX3 TX3-RX1 TX2-RX3 TX2-RX1

If the TX1, TX2 and TX3 is transmit in order, defined in chirp shown in below

chirpCfg 0 0 0 0 0 0 0 1
chirpCfg 1 1 1 0 0 0 0 2
chirpCfg 2 2 1 0 0 0 0 4

The visual antenna index becomes (as shown in figure)

	Azimuth Index			
Elevation Index	0	-1	-2	-3
0	4	2		
-1	3	1		
-2	12	10	8	6
-3	11	9	7	5



•Then users can obtain the azimuth index and elevation index for antenna index 1 – 12

Azimuth Index for virtual antenna 1

Azimuth Index for virtual antenna 12

–AzimuthIndexArray = [-1 -1 0 0 -3 -3 -2 -2 -1 -1 0 0]
–ElevationIndexArray = [-1 0 -1 0 -3 -2 -3 -2 -3 -2 -3 -2]

•This is how you program antGeometry0 and antGeometry1

– antGeometry0 -1 -1 0 0 -3 -3 -2 -2 -1 -1 0 0
– antGeometry1 -1 0 -1 0 -3 -2 -3 -2 -3 -2 -3 -2

•For the phaseRotation, based on the information provided, the phase of RX1/RX3 is 180 phase inversion from RX2/RX4

– antPhaseRot -1 1 -1 1 -1 1 -1 1 -1 1 -1 1

Support AWR6843 ODS Antenna

The visual antenna array for ODS is listed blow.

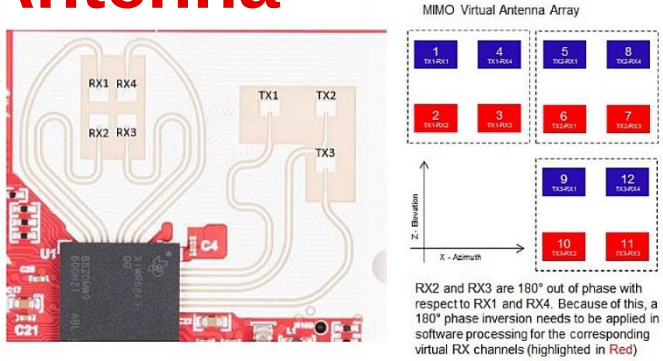
TX1-RX1 TX1-RX4 TX2-RX1 TX2-RX4
TX1-RX2 TX1-RX3 TX2-RX2 TX2-RX3
TX3-RX1 TX3-RX4
TX3-RX2 TX3-RX3

If the TX1, TX2 and TX3 is transmit in order, defined in chirp shown in below

chirpCfg 0 0 0 0 0 0 0 1
chirpCfg 1 1 1 0 0 0 0 2
chirpCfg 2 2 1 0 0 0 0 4

The visual antenna index becomes

Elevation Index \ Azimuth Index	0	-1	-2	-3
0	1	4	5	8
-1	2	3	6	7
-2			9	12
-3			10	11



•Then users can obtain the azimuth index and elevation index for antenna index 1 – 12

Azimuth Index for virtual antenna 1

Azimuth Index for virtual antenna 12

–AzimuthIndexArray = [0 0 -1 -1 -2 -2 -3 -3 -2 -2 -3 -3]
–ElevationIndexArray = [0 -1 -1 0 0 -1 -1 0 -2 -3 -3 -2]

•This is how you program antGeometry0 and antGeometry1
– antGeometry 0 0 0 -1 -1 -2 -2 -3 -3 -2 -2 -3 -3
– antGeometry 1 0 -1 -1 0 0 -1 -1 0 -2 -3 -3 -2

•For the phaseRotation, based on the information provided, the phase of RX2/RX3 is 180 phase inversion from RX1/RX4
– antPhaseRot 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1

Support AWR6843 ISK Antenna

- The visual antenna array for ISK is listed blow.

				TX2-RX1	TX2-RX2	TX2-RX3	TX2-RX4
TX1-RX1	TX1-RX2	TX1-RX3	TX1-RX4	TX3-RX1	TX3-RX2	TX3-RX3	RX3-RX4

- If the TX1, TX2 and TX3 is transmitted in order, then the visual antenna index becomes

			5	6	7	8	
1	2	3	4	9	10	11	12

- Define the index for azimuth and elevation direction:

		Azimuth Index							
		0	-1	-2	-3	-4	-5	-6	-7
Elevation Index	0			5	6	7	8		
	-1	1	2	3	4	9	10	11	12

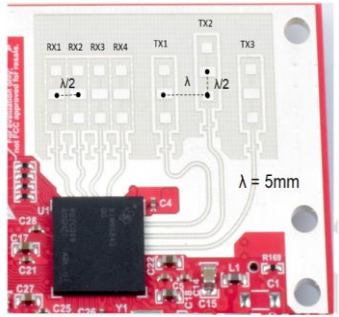
- Then you can program antGeometry0 and antGeometry1

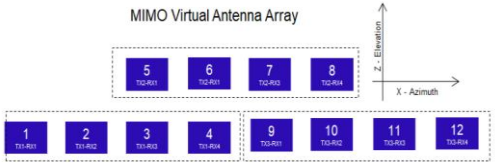
Azimuth Index for virtual antenna 1

Azimuth Index for virtual antenna 12

  - antGeometry 0 0 -1 -2 -3 -2 -3 -4 -5 -4 -5 -6 -7
  - antGeometry 1 -1 -1 -1 -1 0 0 0 0 -1 -1 -1 -1

- For the phaseRotation, based on the information provided, there is no phase jump between different antennas.
  - antPhaseRot 1 1 1 1 1 1 1 1 1 1 1 1

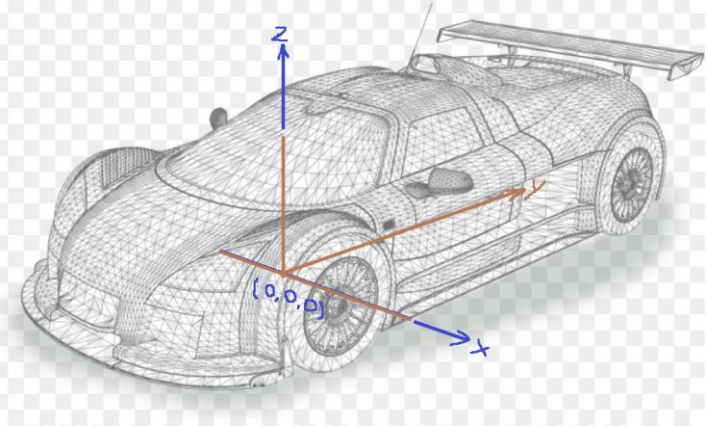




Configuration to Support different Sensor Mounting

Car Frame of Reference and Point Cloud Transformation

- The detected point cloud is all relative to the sensor.
  - The sensor position can change, but the seating zone for a car is fixed given a car.
  - For simplicity, in the visualizer, we define the seating zone based on the car coordinates, and transform the point cloud from sensor coordinates to the car coordinates.
- To support different position and different mounting angle, "sensorPosition" CLI command is used
  - Indicates the mounting offset in (x, y, z) and mounting rotation angle in y-z plane, x-y plane and x-z plane



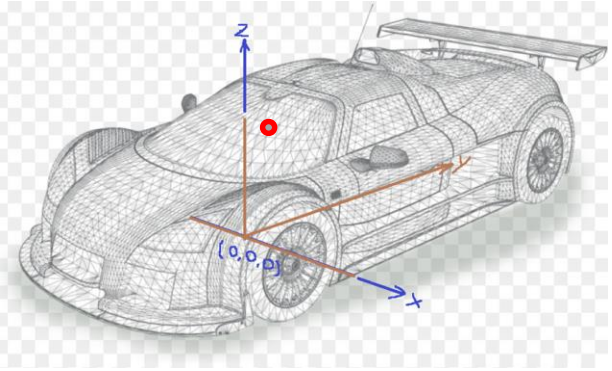
CLI command	Parameters (in command order)
sensorPosition	offset in x direction, in meter
	offset in y direction, in meter
	offset in z direction, in meter
	Clockwise rotation angle in y-z plane, in degree
	Clockwise rotation angle in x-y plane, in degree
	Clockwise rotation angle in x-z plane, in degree

Mounting example 1: Front mounting

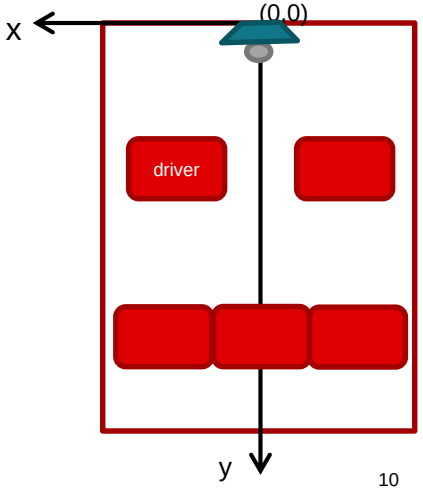
- When the sensor is mounted on the back of the rear mirror and facing straight forward, the sensor position should be programmed as

  sensorPosition 0 0.1 0.8 5 0 0

- This indicates the sensor is side mounted at (x=0, y= 0.1, z = 0.8m) and facing forward with slightly tilting down angle of 5 degree, i.e., 5 degree clockwise rotation in y-z plane.



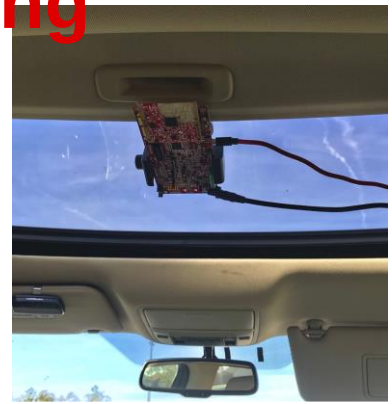
CLI command	Parameters (in command order)	Example Values for Front mounting
sensorPosition	offset in x direction, in meter	0
	offset in y direction, in meter	0.1
	offset in z direction, in meter	0.8
	Clockwise rotation angle in y-z plane, in degree	5
	Clockwise rotation angle in x-y plane, in degree	0
	Clockwise rotation angle in x-z plane, in degree	0

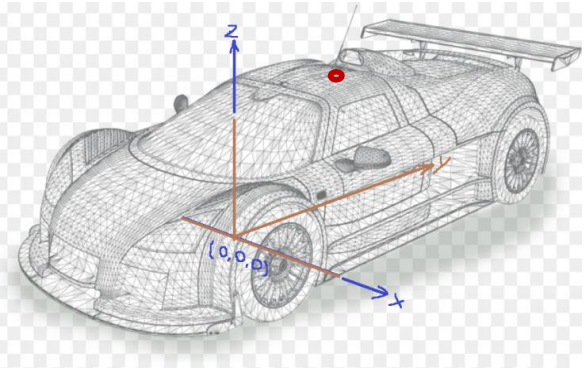


Note: All the pictures are taking from the second row looking towards the first row.

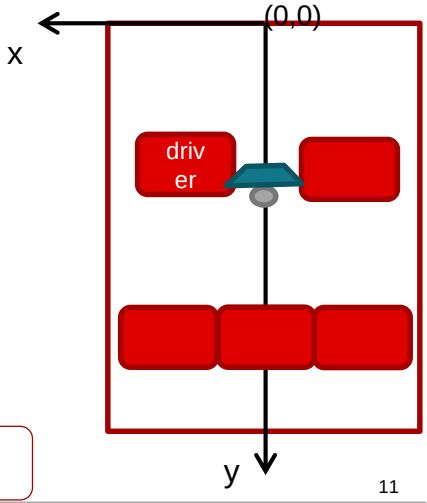
Mounting example 2: Overhead Mounting

- when the sensor is mounted around the center of the roof facing downward, the sensor position should be programmed as
- sensorPosition 0 1.2 1.25 90 0 0
- This indicates the sensor is mounted at (x = 0, y = 1.2m, z = 1.25m) and rotated 90 degree to face the floor, i.e., 90 degree clockwise rotation in y-z domain.





CLI command	Parameters (in command order)	Example Values for Front mounting
sensorPosition	offset in x direction, in meter	0
	offset in y direction, in meter	1.2
	offset in z direction, in meter	1.25
	Clockwise rotation angle in y-z plane, in degree	90
	Clockwise rotation angle in x-y plane, in degree	0
	Clockwise rotation angle in x-z plane, in degree	0



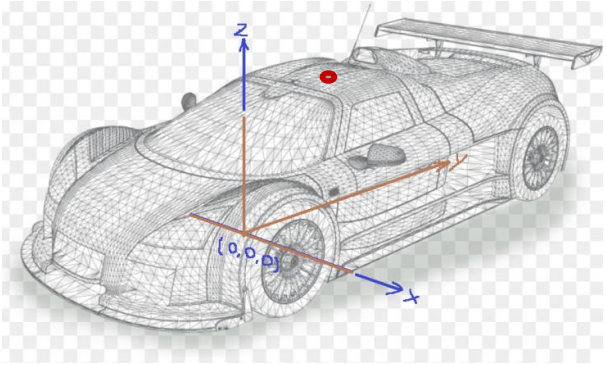
Note: All the pictures are taking from the second row looking towards the first row.
The USB cable is on the passenger side

Mounting example 3: Overhead Mounting with x-y rotation

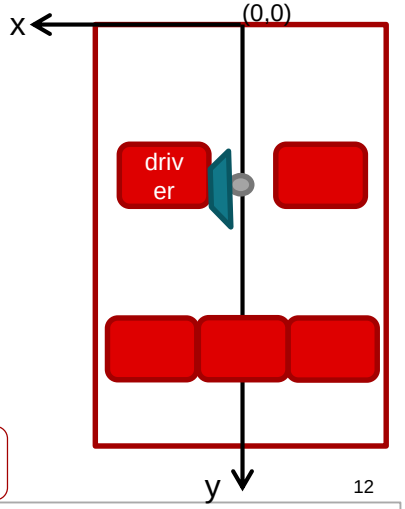
- Sensor mounted at the center of the roof facing downward, but rotated 90 degree in x-y domain,

  sensorPosition 0 1.05 1.25 90 -90 0

- This indicates the sensor is mounted at (x = 0, y = 1.05m, z = 1.1m) and rotated 90 degree in y-z domain to face the floor, then rotated 90 degree counter-clockwise in x-y domain



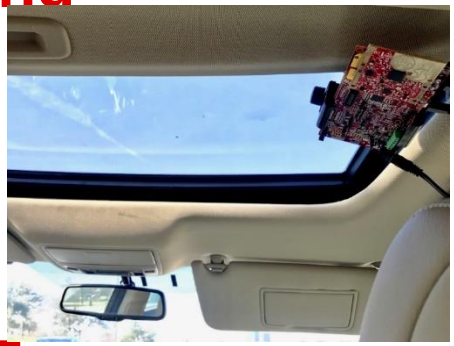
CLI command	Parameters (in command order)	Example Values for Front mounting
sensorPosition	offset in x direction, in meter	0
	offset in y direction, in meter	1.05
	offset in z direction, in meter	1.25
	Clockwise rotation angle in y-z plane, in degree	90
	Clockwise rotation angle in x-y plane, in degree	-90
	Clockwise rotation angle in x-z plane, in degree	0

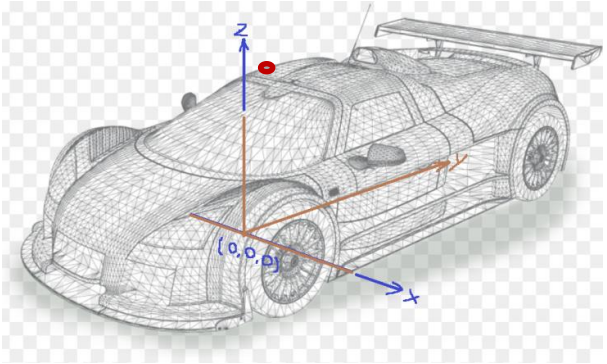


Note: All the pictures are taking from the second row looking towards the first row

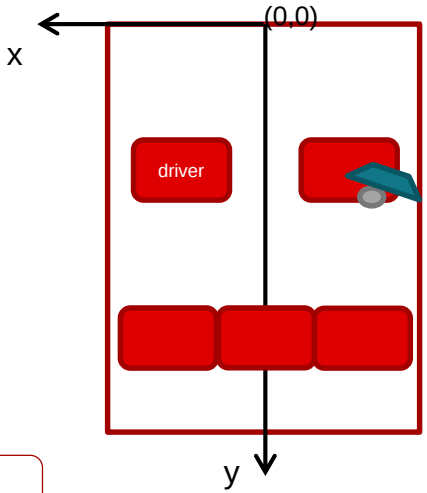
Mounting example 4: Overhead side mounting

- Sensor mounted at the passenger side of the roof, and tilting towards the driver side,
- sensorPosition -0.4 1.2 1.25 90 0 -25
- This indicates the sensor is mounted at (x = -0.4m, y = 1.2m, z = 1.25m) and rotated 90 degree in y-z domain to face the floor, then rotated 25 degree counter-clockwise in x-z plane towards the driver side





CLI command	Parameters (in command order)	Example Values for Front mounting
sensorPosition	offset in x direction, in meter	-0.4
	offset in y direction, in meter	1.2
	offset in z direction, in meter	1.25
	Clockwise rotation angle in y-z plane, in degree	90
	Clockwise rotation angle in x-y plane, in degree	0
	Clockwise rotation angle in x-z plane, in degree	-25



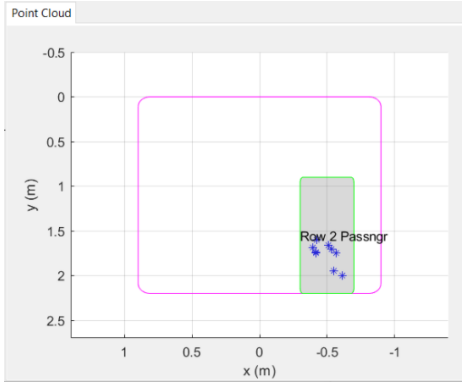
Note: All the pictures are taking from the second row looking towards the first row.

Zone Design Example

Occupancy Display

- Some limitations in the visualizer
  - totNumRows: support 2 or 3 rows
- The limitation for the car occupancy display (as shown on the top right side)
  - First row supports two seats
  - Second row supports three seats
  - Third row supports three seats
  - Users can use zone definition to disable some of these seats
- No limitation on the point cloud display (as shown on the bottom right), users can define any other area, like footwell, intruder area and etc. These area will not display on the car occupancy display, but will show in the point cloud display.

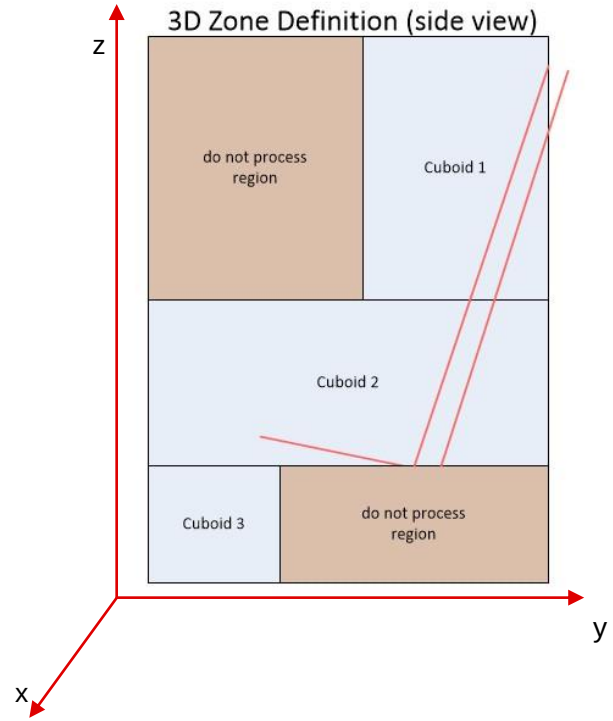
Occupancy View



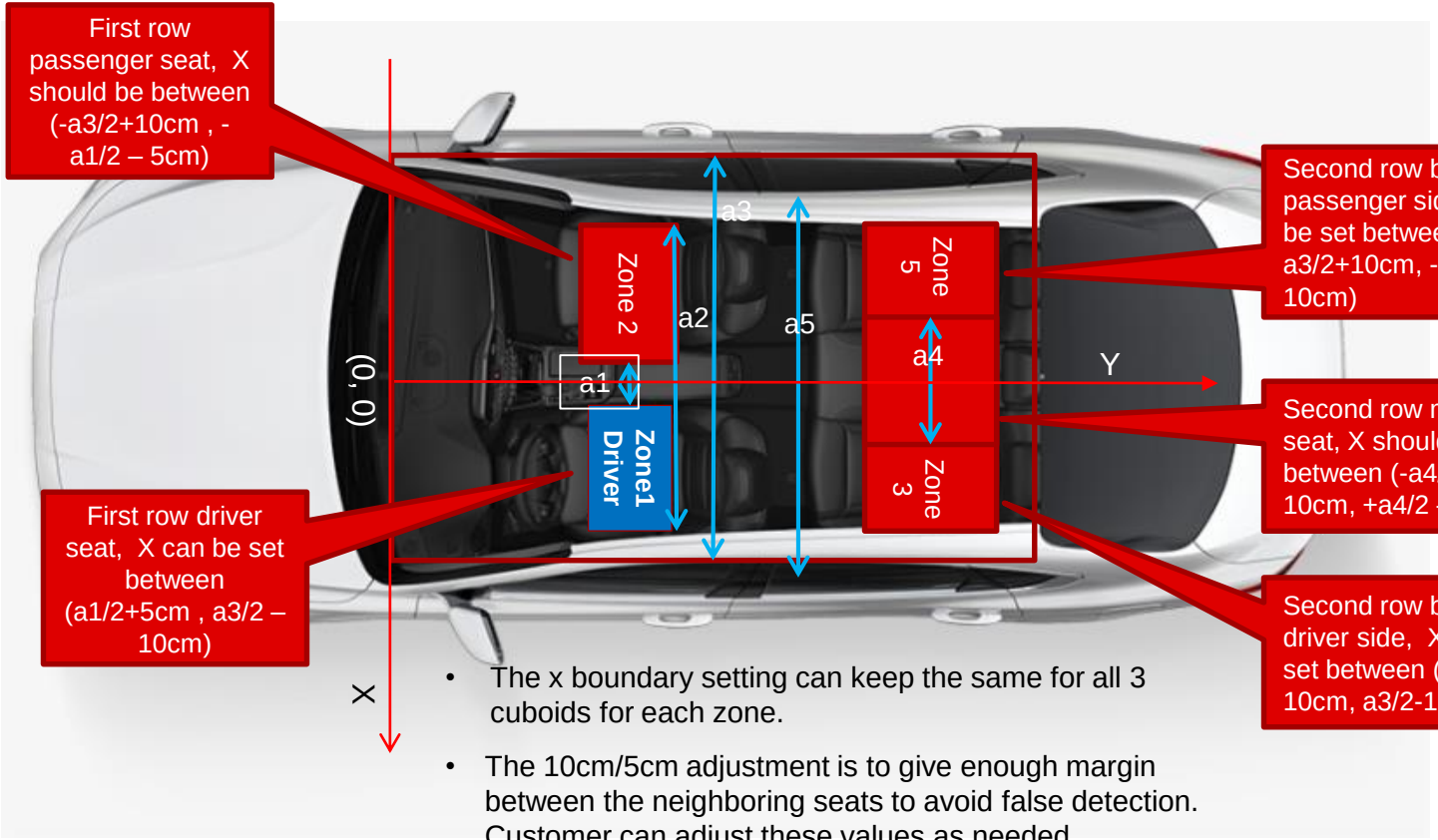
Point Cloud Display

Zone Definition

- The zone is defined with a number of cuboids, which are simply rectangular volumes, having depth (y direction), width (x direction), and height (z direction). These cuboids will approximate the space where want occupants may be located.
- In this demo, 1, 2 or 3 (up to 3) cuboids per zone are defined.
  - A detected point in the point cloud residing any of these cuboids will be included in the calculations for determining occupancy.
  - If your zone is a cargo area, it may be needed to define only one cuboid to approximate the space.
  - For normal vehicle seats, foot well areas are included to check for children and pets. Hence three cuboids are defined
    - Cuboid 1: Head and chest area
    - Cuboid 2: Lap area
    - Cuboid 3: Foot well
- Following slides show how to define zone areas



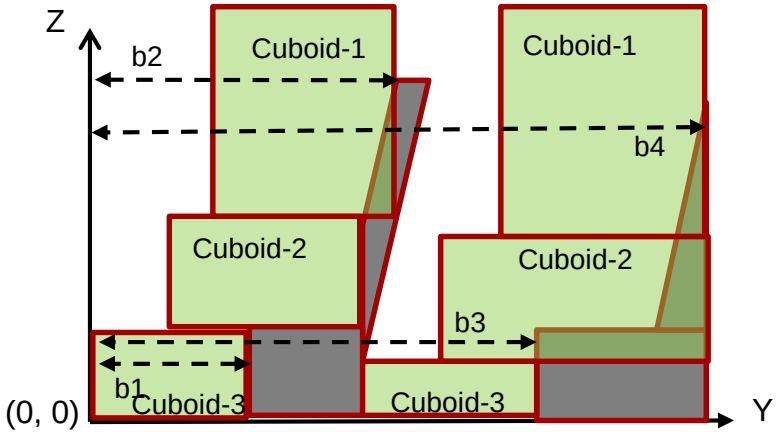
Zone area definition - X direction – measure a1~ a5



- The x boundary setting can keep the same for all 3 cuboids for each zone.
- The 10cm/5cm adjustment is to give enough margin between the neighboring seats to avoid false detection. Customer can adjust these values as needed.

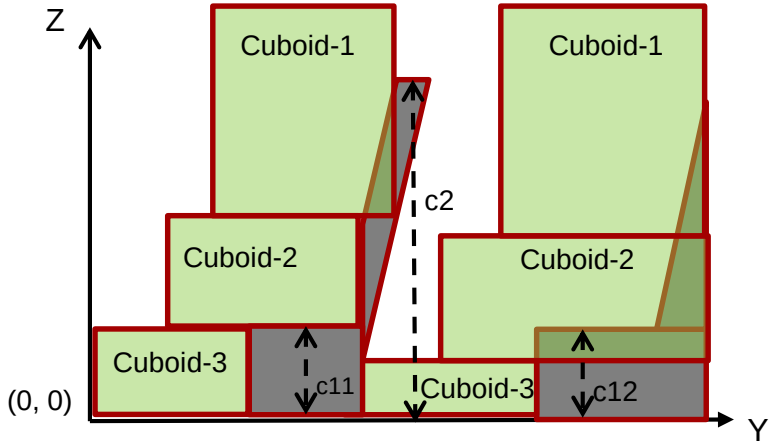
Y Direction – Measure b1~ b4

- First row
  - cuboid-1 can be set to (40cm, b2)
  - cuboid-2 set to (20, b2-20cm);
  - cuboid-3 set to (0, b1);
- Second row
  - cuboid-1 can be set to between (b2 + 40cm, b4)
  - cuboid-2 set to between (b2+20cm, b4)
  - cuboid-3 set to between (b1+40, b3).
- In Y direction, we can keep the same setting for all the seats in the same row.
- The adjustment 20cm, 40cm is to leave enough margin between the first row and second row and the same time, trying to cover enough foot-well area in case baby needs to be detected for foot-well. Customer can adjust these value as needed



Z direction – measure c1~ c2

- C1 = max (C11, C12)
- First row
  - cuboid-1 can be set to (c1+40cm, c2+20cm)
  - cuboid-2 set to (c1, c1+40cm)
  - cuboid-3 set to (-10cm, c1)
- Second row
  - cuboid-1 can be set to between (c1 + 30cm, c2+20cm)
  - cuboid-2 set to between (c1-10, c1+30)
  - cuboid-3 set to between (-10cm, c1-10).
- In Z direction, we can keep the same setting for all the seats in the same row.
- 10cm, 20cm, 30cm, 40cm above are just some rough adjustments. It was used to leave enough margin between the first row and second row and the same time, trying to cover enough foot-well area in case baby needs to be detected for foot-well.
- Customer can adjust these value as needed.



Example Zone Definition and Sensor Orientation Commands

Zone parameters	Measurement in cm
a1	30
a2	125
a3	160
a4	40
a5	130
b1	70
b2	110
b3	160
b4	210
c1	40
c2	90
Sensor Position-X	0
Sensor Position-Y	120
Sensor Position-Z	126

```
% Sensor position
% <xOffset> <yOffset> <zOffset> <yzTilt>,<xyTilt>,<xzTilt>
sensorPosition 0 1.2 1.25 90 0 0

% Origin = -X to psngr, +X to driver, yStart, yEnd (0 = brake panel), zStart, zEnd (z: 0 start of the floor)

% zone 1 (driver) cuboids
cuboidDef 1 1   0.2 0.70    0.4 1.1   0.8 1.1
cuboidDef 1 2   0.2 0.70    0.2 0.9    0.4  0.8
cuboidDef 1 3   0.2 0.70    0 0.7    -0.1  0.4
% zone 2 (front passenger) cuboids
cuboidDef 2 1  -0.70 -0.2   0.4 1.1   0.8  1.1
cuboidDef 2 2  -0.70 -0.2   0.2 0.9    0.4  0.8
cuboidDef 2 3  -0.70 -0.2   0 0.7    -0.1  0.4
% zone 3 (2nd row driver side) cuboids
cuboidDef 3 1   0.30 0.70    1.5 2.1    0.7  1.1
cuboidDef 3 2   0.30 0.70    1.3 2.1    0.3  0.7
cuboidDef 3 3   0.30 0.70    0.8 1.6    -0.1  0.3
% zone 4 (2nd row middle) cuboids
cuboidDef 4 1  -0.10 0.10   1.5 2.1    0.7 1.1
cuboidDef 4 2  -0.10 0.10   1.3 2.1    0.3  0.7
cuboidDef 4 3  -0.10 0.10   1.1 1.6    -0.1 0.3
% zone 5 (2nd row passenger side) cuboids
cuboidDef 5 1  -0.70 -0.30   1.5 2.1    0.7  1.1
cuboidDef 5 2  -0.70 -0.30   1.3 2.1    0.3  0.7
cuboidDef 5 3  -0.70 -0.30   1.1 1.6    -0.1  0.3
```