– mmWave Radar Sensor –

Auto Radar Apps Webinar: Vehicle Occupancy Detection

Please note, this webinar is being recorded and will be made available to the public.

Audio Dial-in info:
Phone #: 1-972-995-7777
Passcode: **98425354#**

Vehicle Occupancy Detection Webinar

Dave Woodall
Senior Software Engineer, Auto Radar Apps
TI Dallas

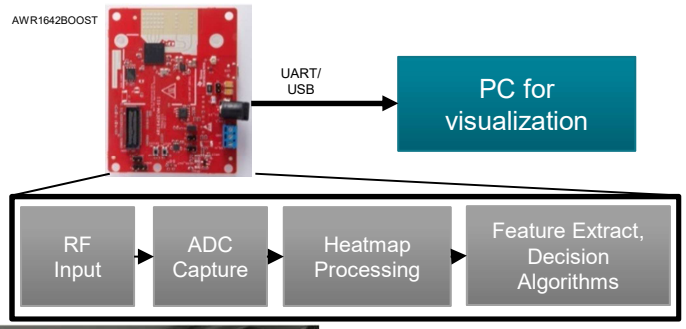
Audio Dial-in info:
Phone #: 1-972-995-7777
Passcode: **98425354#**

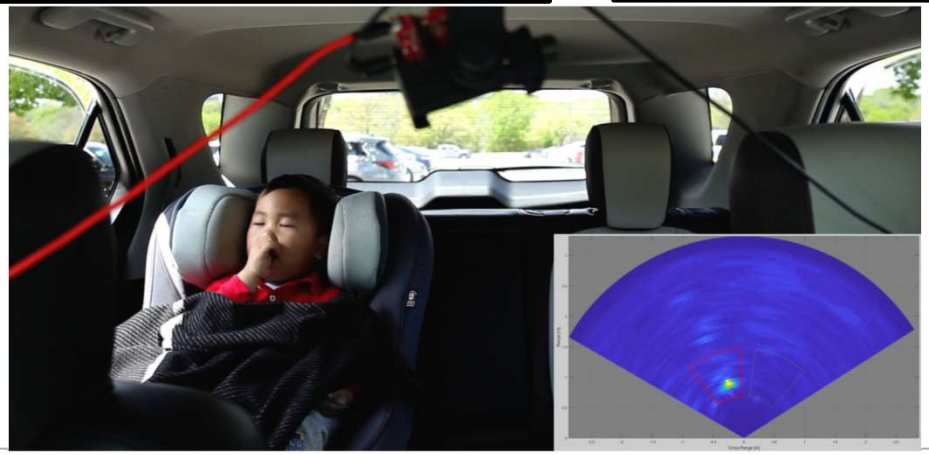
Vehicle Occupancy Detection TI Design

- TI Design Number: TIDEP-01001: www.ti.com/tool/tidep-01001

- AWR1642 – TI's FMCW mmWave (76-81GHz) **Single chip Radar** with 2Tx/4Rx RF front End, ADC, DSP(C674x) and MCU(Cortex-R4F)
- Complete RADAR data processing on-board the device.
- Device outputs heatmap, feature and zone decision information.
- Chirping pattern provides ability to detect small movements, such as breathing.

Demonstration using AWR1642 EVM

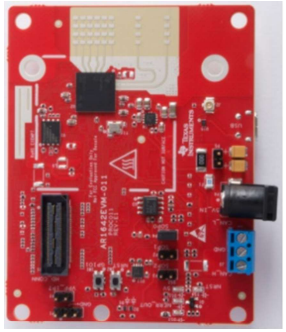




Implementation Details

Hardware

- AWR1642
  - 4 RX, 2TX
  - ES 1.0, 2.0
  - RF, firmware improvements



Software

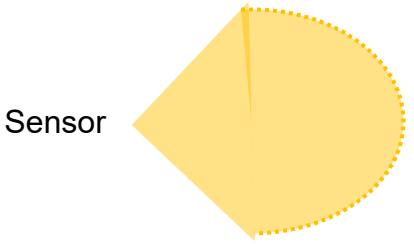
- mmwave_automotive_toolbox_x_x_x
  - Available in TI-REX (within CCS)
  - 1.x for ES1.0, 2.x for ES2.0
- mmWave SDK
  - Installer here: www.ti.com/tool/mmwave-sdk
  - 1.x for ES1.0, 2.x for ES2.0

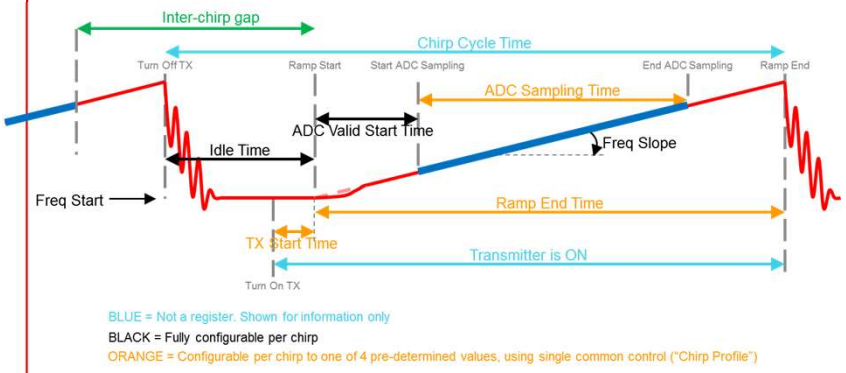
Key Performance Specifications

Parameter	Value
Max range	3.3 m
Max velocity	2.78 m/s
Range resolution	4.69cm
Velocity resolution	0.022 m/s
Field of View	120° Horizontal
Bandwidth	3.920 GHz

VOD Chirp Configuration Features

- TDM 2x4 MIMO
  - Suitable for applications which require a high angular resolution
- 512 Chirps/frame for high velocity resolution to detect slow movement like breathing
  - 256 TX1 chirps interlaced with
  - 256 TX2 chirps.
  - Large chirp cycle time =340us
- Maximize range resolution while fitting in a 512KB radar cube:
  - num samples = 64
  - sample rate = 2200ksps
  - freq slope = 98 Mhz/us





Example VOD Configuration File

- This is one of the provided VOD chirp/demo configurations:

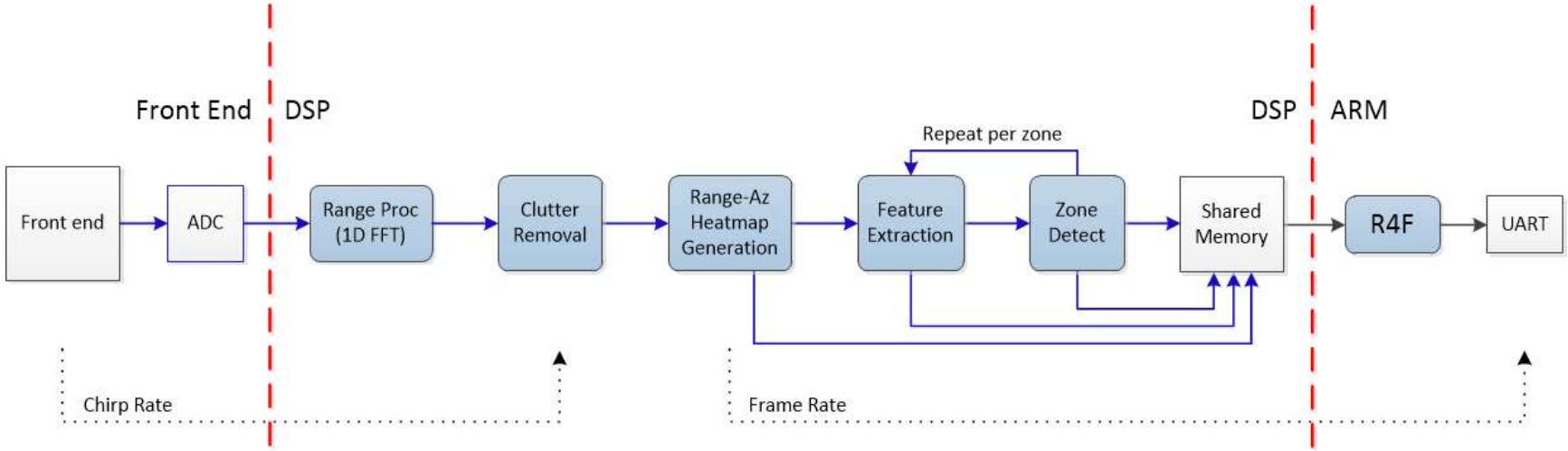
- Lines 3-13 configure the radar front end.

- Lines 18-25 configure the VOD demo's algorithms.

- This file is sent via a UART when the demo is started.

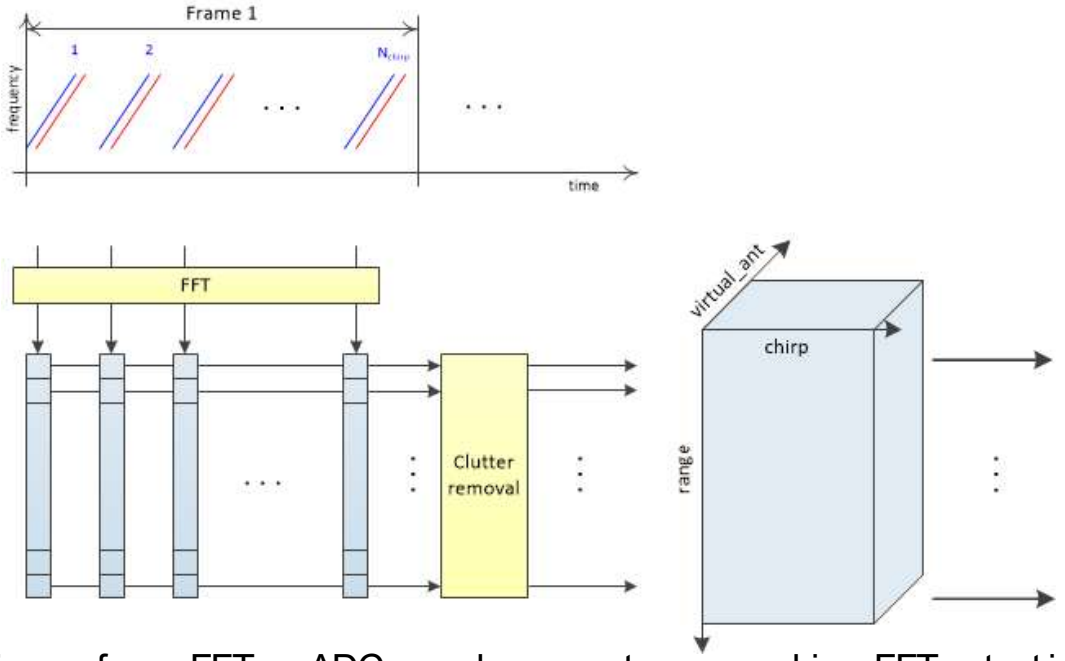
```
 1: sensorStop
 2: flushCfg
 3: dfeDataOutputMode 1
 4: channelCfg 15 3 0
 5: adcCfg 2 1
 6: adcbufCfg -1 0 0 1 1
 7: profileCfg 0 77 250 10 40 0 0 98 1 64 2200 0 0 40
 8: chirpCfg 0 0 0 0 0 0 0 1
 9: chirpCfg 1 1 0 0 0 0 0 2
10: chirpCfg 2 2 0 0 0 0 0 1
11: chirpCfg 3 3 0 0 0 0 0 2
12: frameCfg 0 3 128 0 160 1 0
13: lowPower 0 1
14: guiMonitor -1 0 1 1
15: calibDcRangeSig -1 0 -5 8 256
16:
17: %OD Demo commands:
18: zoneDef 2 17 16 8 15 17 16 26 15
19: coeffMatrixRow  0  -11.705674   -6.557091  -6.556105    2.793170    2.806355
20: coeffMatrixRow  1  -9.453702   -0.592442   -5.440082    3.204209   -5.838531
21: coeffMatrixRow  2  -9.492407   -5.426766   -0.588876   -5.904976    3.170924
22: coeffMatrixRow  3  -1.211744    3.400299    3.446914    8.541460    8.544744
23: meanVector  22.771068   22.756197   -3.955602   -3.970473    0.712385
24: stdVector   6.086866    6.070428    3.236808    3.262139    0.348652
25: oddemoParms  8  0.001
26: sensorStart
27:
```

Signal Processing Chain: Block Diagram



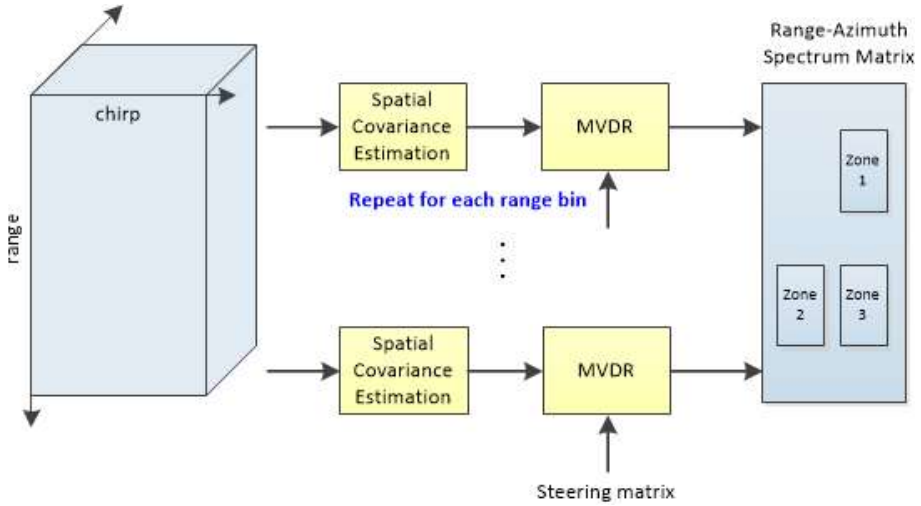
- Low level signal processing chain is implemented on DSP (C674x) of AWR1642.
- Low level signal processing chain outputs heatmap, feature parameters and zone decisions.
- R4F (MSS ARM) processor transfers outputs from shared memory to UART.
- All algorithms are fully documented in the design document. Go to www.ti.com/tool/tidep-01001.

Front End and Range (Chirp) Processing



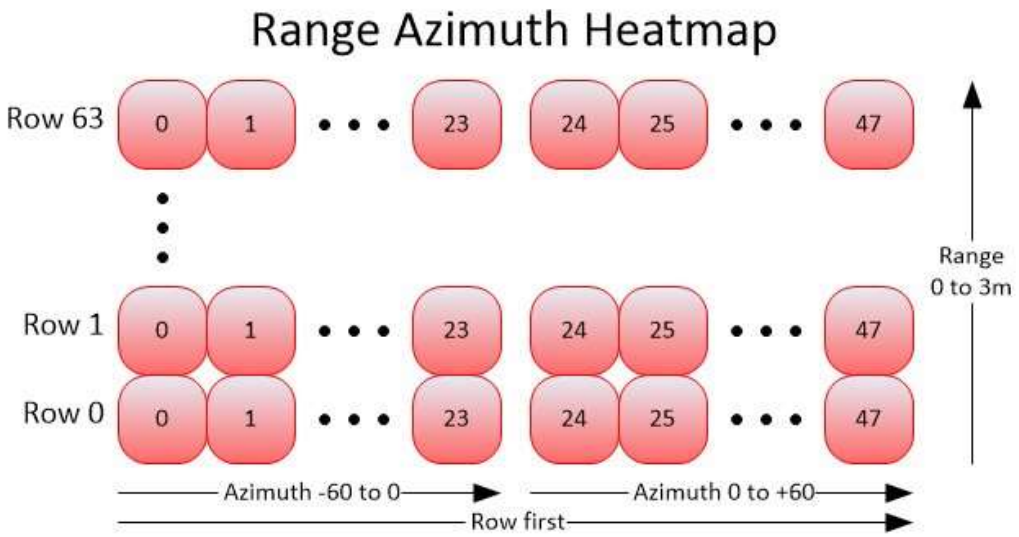
- Range processing performs FFT on ADC samples per antenna per chirp. FFT output is a set of range bins.
- Perform static clutter removal by subtracting the estimated DC component from each range bin.
- Range processing results in local scratch buffers are EDMA'd to the L3 radar cube with transpose.

Range-Azimuth Heatmap Processing



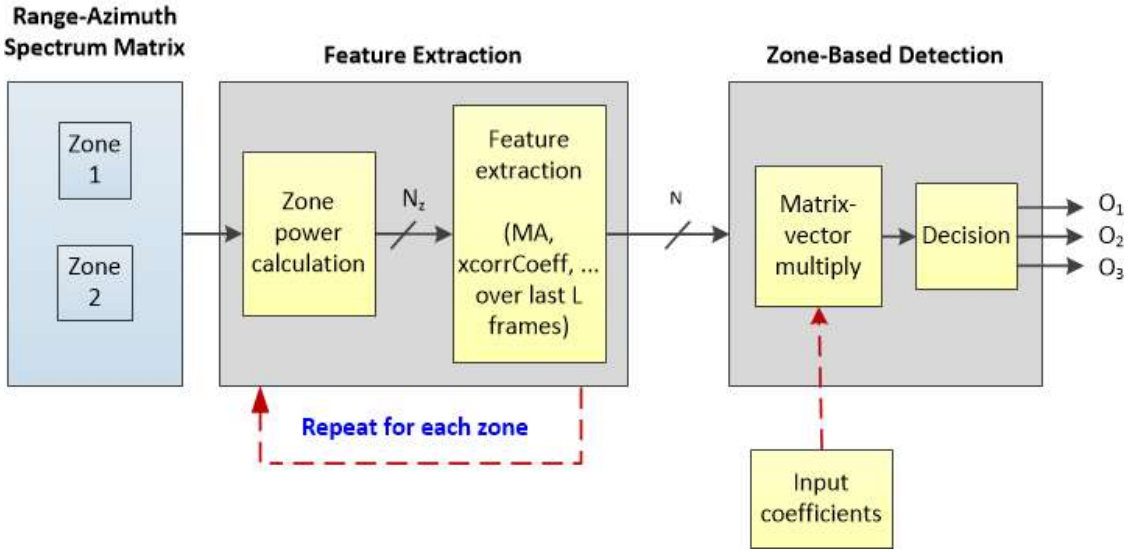
- Perform a high resolution Direction-of-Arrival (DOA) Spectral Estimation to calculate a 2D heatmap for the frame:
  - Spectral Covariance estimation is performed by stacking range bins across antennas, per n^{th} range bin and k^{th} chirp.
  - DOA estimation uses MVDR (minimum variance distortionless response, or Capon's beamforming) to compute the angular spectrum for each range bin.

Range-Azimuth Heatmap Format



- Heatmap contains 64 range rows, each with 48 azimuth (angle) columns
- Row 0 is nearest range, row 63 is furthest (3 meters)
- Column 0 is -60 degrees, column 47 is +60 degrees.

Zone Processing (Feature Extract and Detection)



- 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.
- Perform matrix multiplication with the input coefficients and the feature vector. This yields an array of flags, one flag per zone, a '1' indicating 'zone occupied', and '0' indicating 'empty'.

Software Integration into mmWave Demo

- VOD demo is based on the mmWave demo.
- Standard mmWave demo frame processing is replaced with three calls to perform these algorithms:
  - ODDemo_Heatmap_aoaEstCaponBF()
  - ODDemo_Feature_extract()
  - ODDemo_Decision_process()
- This code (shown on the right) is found in dss_data_path.c.
- The generated heatmap is not specific to this demo and can easily be reused for other applications.
- Feature extraction and zone decision processing are performed only if selected for output via the guiMonitor command.

```
void MmwDemo_interFrameProcessing(MmwDemo_DSS_DataPathObj *obj)
{
    uint32_t rangeIdx;
    MmwDemo_GuiMonSel *guiMonSel = &gMmwDssMCB.cliCfg[0].guiMonSel;

    // Occupancy Detection Inter Frame Processing

    oddemo_dataPathObj.swapBuf = oddemo_swapBuf;

    for (rangeIdx = 0; rangeIdx < ODDEMO_MAX_RANGE; rangeIdx++)
    {
      Re16bitIm16bit_swap(oddemo_dataPathObj.inputAntSamples,
                          oddemo_swapBuf,
                          CHIRP_STEP);

      ODDemo_Heatmap_aoaEstCaponBF(&oddemo_dataPathObj);

      oddemo_dataPathObj.inputAntSamples += CHIRP_STEP;
      oddemo_dataPathObj.rangeAzimuthHeatMap += ODDEMO_MAX_AZIMUTH;
    }

    oddemo_dataPathObj.inputAntSamples = (cmplx16ReIm_t *)&gMmwL3[0];
    oddemo_dataPathObj.rangeAzimuthHeatMap = &oddemo_rangeAzimuthHeatMap[0];

    if ((guiMonSel->featureVector == 1) ||
        (guiMonSel->decision == 1))
    {

      ODDemo_Feature_extract(ODDEMO_MAX_ZONES,
                             oddemo_rangeAzimuthHeatMap,
                            &oddemo_feature);

      if (guiMonSel->decision == 1)
      {
        uint8_t *dptr = (uint8_t *)&oddemo_decision;

        ODDemo_Decision_process(oddemo_coeffMatrix,
                                &oddemo_feature,
                                 dptr);
      }
    }
```

Live Demonstrations

- TI Resource Explorer (TIREX)
- Locate the VOD demo in TIREX
- Import the VOD demo to CCS Projects
- Build the demo executables
- Run the demo
- Demo video

- Questions?

Q & A

Learn more about TI mmWave Sensors

- Learn more about xWR1x devices, please visit the product pages
  - AWR1642: http://www.ti.com/product/AWR1642
- Get started evaluating the platform with xWR1x EVMs, purchase EVM at
  - AWR1642 EVM: http://www.ti.com/tool/AWR1642BOOST

- VOD Video: https://training.ti.com/vehicle-occupant-detection-using-mmwave-sensors
- VOD Blog: http://e2e.ti.com/blogs_/b/behind_the_wheel/archive/2018/04/30/detecting-vehicle-occupancy-with-mmwave-sensors

- Automotive toolbox on TIREX: http://dev.ti.com/tirex/#/DevTool/AWR1642%20Automotive%20EVM/?link=Software%2FmmWave%20Sensors%2FAutomotive%20Toolbox
- Ask questions on TI's E2E forum @ http://e2e.ti.com