Overview

- This lab exercise demonstrates the ability of IWR6843 TI-mmWave sensor to measure body displacements due to breathing and heart beat.
- Typical body surface displacement parameters due to breathing and hear-beat are

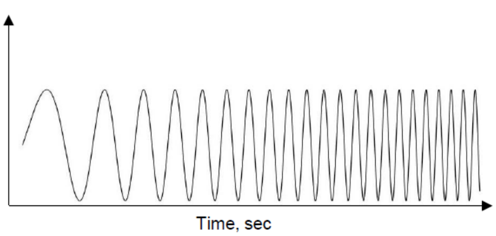
		From Front	From Back
Vital Signs	**Frequency**	**Amplitude**	**Amplitude**
Breathing Rate (Adults)	0.1 – 0.5 Hz	~ 1- 12 mm	~ 0.1 – 0.5 mm
Heart Rate (Adults)	0.8 – 2.0 Hz	~ 0.1 – 0.5 mm	~ 0.01 – 0.2 mm

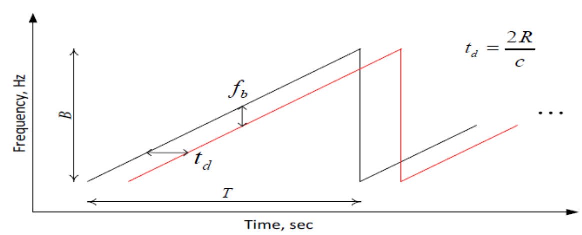
- To measure these small scale vibrations/displacements, we measure the change in phase of the FMCW signal with time at the target range bin $\Delta\phi_b = \frac{4\pi}{\lambda}\Delta R$
  - $\Delta\phi_b$ corresponds to the change in phase when the target moves a distance ΔR
  - Note that a smaller wavelength λ will give better displacement sensitivity
- Code Composer Studio (CCS) project along with source code is provided for this lab
- Pre-built binary files are also provided that can be loaded on to the IWR6843 EVM

Theory

FMCW Basics

- Periodic linearly-increasing frequency chirps (known as Frequency-Modulated Continuous Wave (FMCW)) are transmitted by radar towards the object





- Transmitted FMCW signal is given by $s(t) = e^{j\left(2\pi f_c t + \pi \frac{B}{T} t^2\right)}$

- Signal at the receiver is a delayed version of the transmitted signal $r(t) = e^{j\left(2\pi f_c (t - t_d) + \pi \frac{B}{T} (t - t_d)^2\right)}$

- The received signal from an object at range R after mixing and filtering is given by

$$s(t).r(t) \approx e^{j\left(4\pi \frac{BR}{cT} t + \frac{4\pi}{\lambda} R\right)} = e^{j(f_b t + \phi_b)}$$

FMCW Radar - Vital Signs Measurements

- Note that for a single object, the beat signal *b(t)* is a sinusoidal and has both frequency fb and phase ϕ_b

$$b(t) = e^{j\left(\underbrace{4\pi \frac{BR}{cT}}_{f_b} t + \underbrace{\frac{4\pi}{\lambda} R}_{\phi_b}\right)} = e^{j(f_b t + \phi_b)}$$

- To measure small scale vibrations, we measure the change in phase of the FMCW signal with time at the object range bin. If an object moves a distance ΔR then the change in phase between consecutive measurements is given by

$$\Delta\phi_b = \frac{4\pi}{\lambda}\Delta R$$

As an example at λ=4 mm when we have displacements as small as ΔR = 1mm, the corresponding phase change is $\Delta\phi_b = \pi$

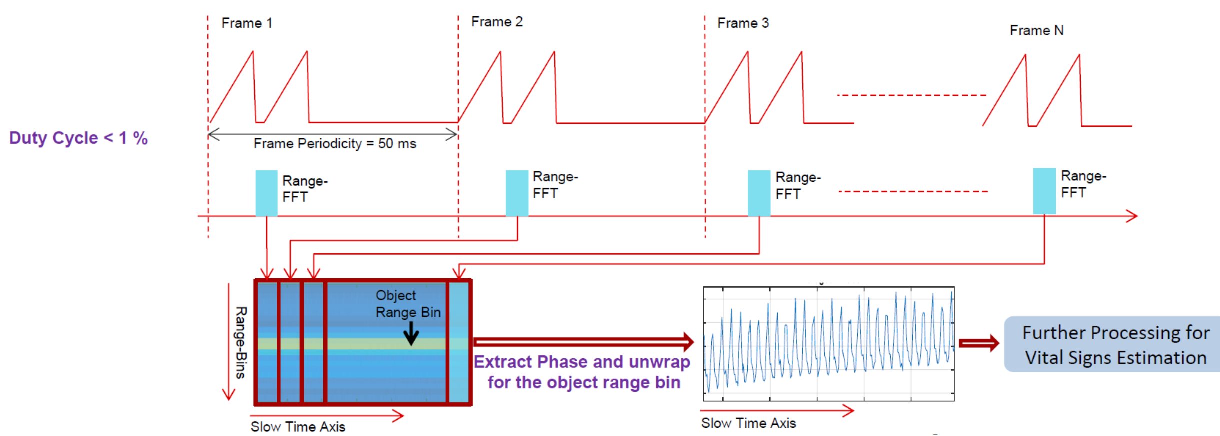
- Phase can be measured by taking the FFT of the beat signal b(t) and computing the phase at the object range-bin.
- Suppose we take the FFT and the object is at range-bin m, then the vibration signal x(t) can be extracted by measuring the phase at range-bin m at time indices nTs , where n is the chirp index and Ts is the time between consecutive measurements

$$x(m, nT_s) = \frac{\lambda}{4\pi}\phi_b(m, nT_s)$$

Note that we are assuming that the vibrations *x(t)* are small so that the object remains in the same range-bin during the duration of the measurements

Chirp Configuration for Demo

- 100 ADC Samples per chirp. Chirp duration is 50 ms based on the IF sampling rate of 2 MHz
- Each frame is configured to have 2 chirps. However only the 1st Chirp in the frame is used for processing
- A single TX-RX antenna pair is currently used for processing (Although all the RX antennas are enabled)
- Vital signs waveform is sampled along the “slow time axis” hence the vital signs sampling rate is equal to the Frame-rate of system



Requirements

Hardware and Software Requirements

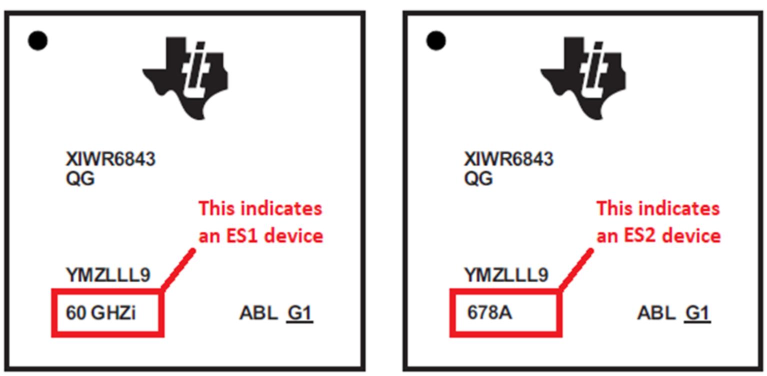
Hardware

Item	Details
Device	Industrial mmWave Carrier Board (http://www.ti.com/tool/MMWAVEICBOOST) and either one of the following antenna modules
	IWR6843ISK Antenna Module (http://www.ti.com/tool/IWR6843ISK) **OR** IWR6843ISK_ODS Antenna Module (http://www.ti.com/tool/IWR6843ISK-ODS) **OR** IWR6843AOPEVM (http://www.ti.com/tool/IWR6843AOPEVM).
	Note: The rest of this document will refer to the above board combination as **EVM**
Computer	PC with Windows 7 or 10. If a laptop is used, please use the 'High Performance' power plan in Windows.
Micro USB Cable	Provided with the Industrial Radar Carrier Board (http://www.ti.com/tool/MMWAVEICBOOST)
Power Supply	5V, 3A with 2.1-mm barrel jack (center positive). The power supply can be wall adapter style or a battery pack with a USB to barrel jack cable.

IWR6843 ES2.0 Only

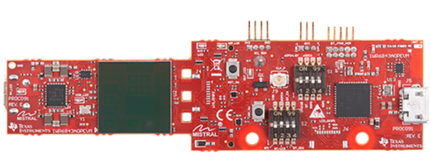
This lab is only compatible with ES2.0 version of IWR6843. Check the device version on your IWR6843ISK using the on-chip device markings as shown below:

1. If line 4 reads `678A`, you have an ES2 device. In this case, this lab is compatible with your EVM.
2. If line 4 reads `60 GHZi`, you have an older ES1 device. In this case, the lab is NOT compatible with your EVM. ES2 IWR6843ISK boards are orderable from the EVM link above.



AoP ES2.0 EVM only

The IWR6843 AoP version of this lab is only compatible with ES2.0 silicon and the corresponding EVM. Please ensure your EVM is the same as in the below image.

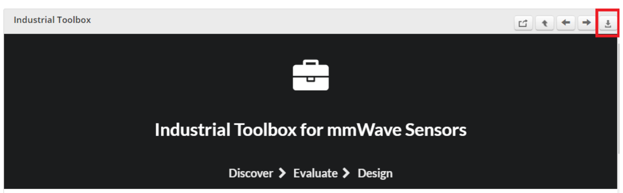


Software

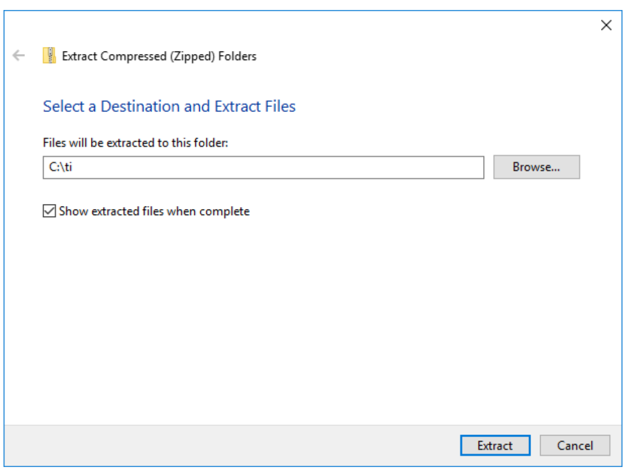
Tool	Version	Download Link
mmWave Industrial Toolbox	Latest	mmWave Industrial Toolbox (http://dev.ti.com/tirex/explore/node?node=AJoMGA2ID9pCPWEKPi16wg__VLyFKFf__LATEST)
TI mmWave SDK	3.5.x.x	TI mmWave SDK 3.5 (http://software-dl.ti.com/ra-processors/esd/MMWAVE-SDK/03_04_00_03/index_FDS.html) and all the related tools are required to be installed as specified in the mmWave SDK release notes (http://software-dl.ti.com/ra-processors/esd/MMWAVE-SDK/latest/exports/mmwave_sdk_release_notes.pdf)
Uniflash	Latest	Uniflash tool is used for flashing TI mmWave Radar devices. Download offline tool (http://www.ti.com/tool/UNIFLASH) or use the Cloud version (https://dev.ti.com/uniflash/#!/)

Expand for mmWave Industrial Toolbox installation without Code Composer Studio

1. Navigate to the TI Resource Explorer (http://dev.ti.com/tirex/explore/node?node=AJoMGA2ID9pCPWEKPi16wg__VLyFKFf__LATEST)
2. Click the download button. A .zip file will be downloaded.

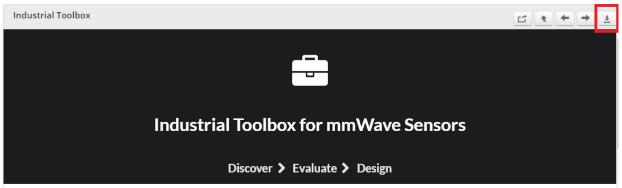


3. Navigate to the .zip file. Right click and then select **Extract All...**. Do NOT use the default path. The path must be `C:\ti`.



Expand for mmWave Industrial Toolbox installation using Code Composer Studio

1. Open CCS
2. In the top toolbar, navigate to **View** → **Resource Explorer**
3. In the **Resource Explorer** side panel (not the main panel with "Welcome to.."), navigate to Industrial Toolbox at **Software** → **mmWave Sensors** → **Industrial Toolbox - <ver>**
4. With Industrial Toolbox selected, the main panel should show the Industrial toolbox landing page. Click on the **Download icon** in the right corner of panel.



Quickstart

The quickstart guide will cover setting up the EVM, flashing firmware, and running the demo.

1. Setup the EVM for Flashing Mode

- For MMWAVEICBOOST + IWR6843ISK/ODS/AOP setup: Follow the instructions for Hardware Setup for Flashing in MMWAVEICBOOST Mode (../../../common/docs/hardware_setup/hw_setup_mmwaveicboost_mode_flashing.html)

2. Flash the EVM using Uniflash

Flash the binary listed below using UniFlash. Follow the instructions for using UniFlash (../../../common/docs/software_setup/using_uniflash_with_mmwave.html)

BIN Name	Board	Location
vital_signs_demo_68xx.bin	IWR6843ISK OR IWR6843ISK-ODS OR IWR6843AOPEVM	<INDUSTRIAL_TOOLBOX_INSTALL_DIR>\mmwave_industrial_toolbox_<VER>\labs\vital_signs\68xx_vital_signs\prebuilt_binaries

3. Setup the EVM for Functional Mode

- For MMWAVEICBOOST + IWR6843ISK/ODS/AOP setup: Follow the instructions for Hardware Setup of MMWAVEICBOOST + Antenna Module for Functional Mode (../../../common/docs/hardware_setup/hw_setup_mmwaveicboost_mode_functional.html)

At this point, the EVM should be powered, connected to the PC, flashed with the demo, and put in functional mode. The hardware setup is now complete.

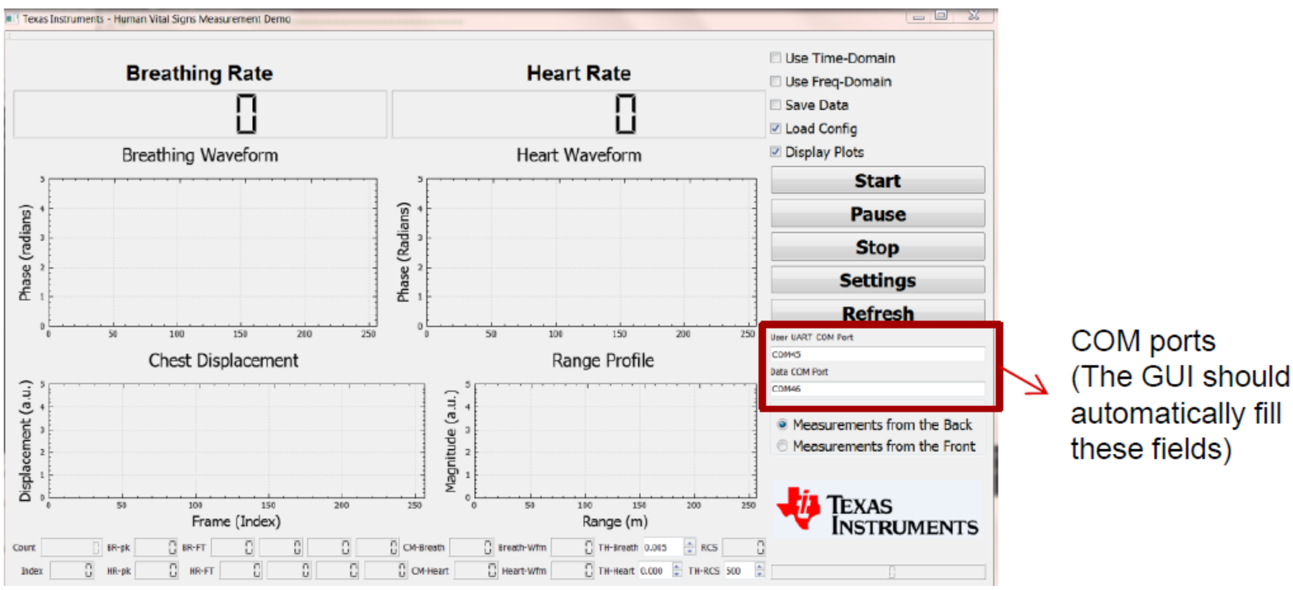
4. Run the Lab

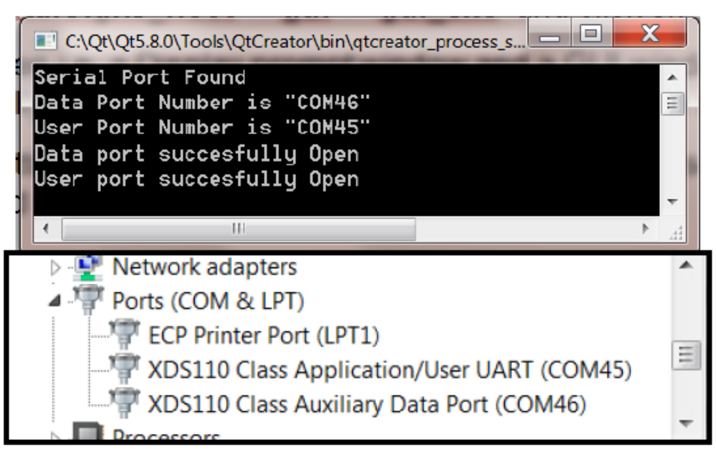
To run the lab, launch and configure the visualizer which displays the detection and tracked object data received via UART.

> **Note:**
>
> The visualizer can only run with the following:
> Industrial mmWave Carrier Board (http://www.ti.com/tool/MMWAVEICBOOST) + IWR6843ISK Antenna Module (http://www.ti.com/tool/IWR6843ISK)
> **OR**
> + IWR6843ISK-ODS Antenna Module (http://www.ti.com/tool/IWR6843ISK-ODS)
> **OR**
> + IWR6843AOPEVM (http://www.ti.com/tool/IWR6843AOPEVM).

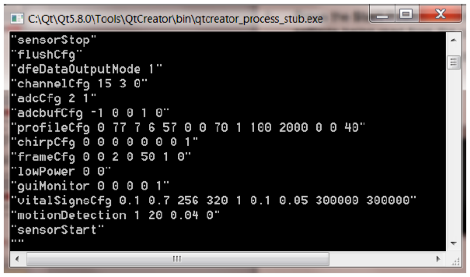
Launch the visualizer:

- Navigate to folder `<mmwave_industrial_toolbox_install_dir>\labs\vital_signs\68xx_vital_signs\gui\gui_exe`
- Run `VitalSignsRadar_Demo.exe`
- Two windows should open i.e. a Display prompt window and a GUI window. If the EVM is connected to the PC, then the display prompt window should successfully open the COM ports (to double check, make sure they match with the port numbers on the Device Manager).
- In the GUI window, the **User UART COM Port** and **Data COM Port** fields should automatically be filled with the correct port numbers (Make sure that no other EVM is connected to the USB ports of the PC)
- If the GUI does not open you might need the vc runtime which can be downloaded here (https://support.microsoft.com/en-us/help/2977003/the-latest-supported-visual-c-downloads)



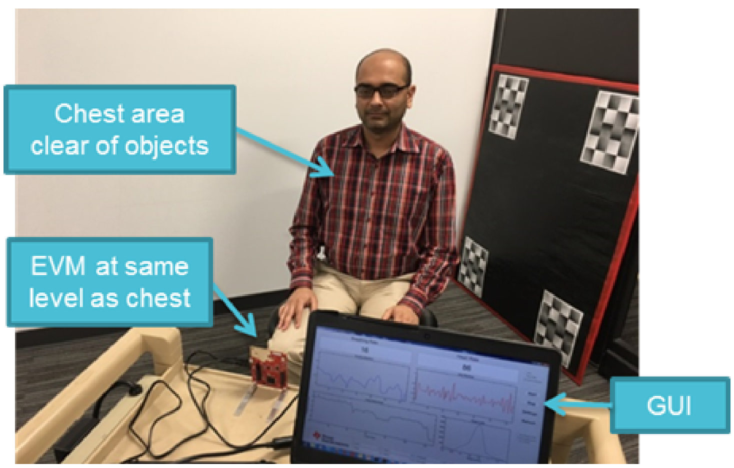


- Press the **Start** Push button in the GUI. In the Display Prompt window you should see the configuration settings being read from the configuration text file and sent through the UART to the EVM
- As soon as the **sensorStart** command is sent, the GUI should start displaying the data

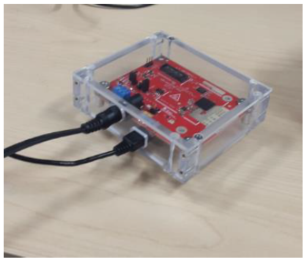


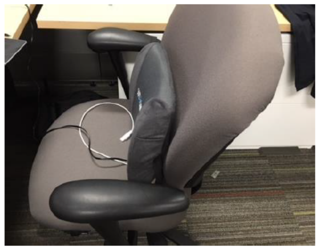
Sensor Placement:

- For **Sensor in Front** the EVM shuld be mounted upright while subject should sit facing in a chair about .3-.8m in front.
- For initial measurement subject should stay still for 10-15 seconds to allow the application to calibrate. For subsequent measurements the subject must stay still for 5-10 seconds.



- For **Sensor in Back**, the EVM can be placed in the back of a car seat or in a seat cushion. This can be done by placing the EVM in an enclosure as shown below.



Developer's Guide

Build the Firmware from Source Code

1. Software Requirements

Tool	Version	Download Link
mmWave Industrial Toolbox	Latest	mmWave Industrial Toolbox (http://dev.ti.com/tirex/explore/node?node=AJoMGA2ID9pCPWEKPi16wg__VLyFKFf__LATEST)
TI mmWave SDK	3.5.x.x	TI mmWave SDK 3.5 (http://software-dl.ti.com/ra-processors/esd/MMWAVE-SDK/03_04_00_03/index_FDS.html) and all the related tools are required to be installed as specified in the mmWave SDK release notes (http://software-dl.ti.com/ra-processors/esd/MMWAVE-SDK/latest/exports/mmwave_sdk_release_notes.pdf)
Code Composer Studio	8.3.1	Code Composer Studio v8.3.1 (http://processors.wiki.ti.com/index.php/Download_CCS#Code_Composer_Studio_Version_8_Downloads)

2. Import Lab Project

For the Vital Signs 68xx Lab, there are two projects, the DSS for the C674x DSP core and the MSS project for the R4F core, that need to be imported to CCS and compiled to generate firmware for the xWR6843.

- Start CCS and setup workspace as desired.
- Import the projects below to CCS using either TI Resource Explorer in CCS or CCS Import Projectspecs method:
    - **vital_signs_68xx_dss.projectspec**
    - **vital_signs_68xx_mss.projectspec**

Expand for details on importing via TI Resource Explorer in CCS

- In the top toolbar, navigate to **View** → **Resource Explorer**
- In the **Resource Explorer** side panel (not the main panel with "Welcome to.."), navigate to **Software** → **mmWave Sensors** → **Industrial Toolbox** → **Labs** → **Vital Signs - 68xx**
- Under the expanded **Vital Signs - 68xx** folder, there should be two CCS projects, **CCS Project - DSS** and **CCS Project - MSS**.

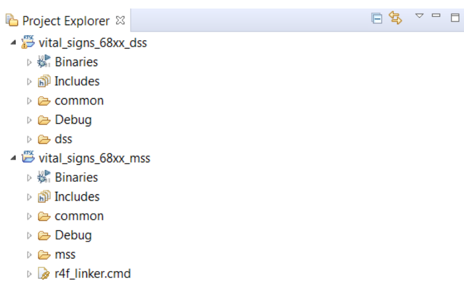
- For each of the two projects: Click on the project, which should open the project in the right main panel, and then click on the Import to IDE button  .

Expand for details on importing via CCS Import Projectspecs

- In the top toolbar, navigate to **Project > Import CCS Projects...**
- With the **Select search-directory** option enabled, click **Browse...**, navigate to the **lab0026_vital_signs_68xx** folder at `<mmwave_industrial_toolbox_install_dir>\labs\vital_signs\68xx_vital_signs` , and then click **OK**.
- Under **Discovered projects**, select **vital_signs_68xx_dss** and **vital_signs_68xx_mss** (ignore any other projects), then click **Finish**.

Successful Import to IDE

After using either method, both project should be visible in **CCS Project Explorer**



Project Workspace

When importing projects to a workspace, a copy is created in the workspace. All modifications will only be implemented for the workspace copy. The original project downloaded in mmWave Industrial Toolbox is not touched.

3. Build the Lab

Build DSS Project

The DSS project must be built before the MSS project.

With the **vital_signs_68xx_dss** project selected in **Project Explorer**, right click on the project and select **Rebuild Project**. Selecting **Rebuild** instead of **Build** ensures that the project is always re-compiled. This is especially important in case the previous build failed with errors.

Successful DSS Project Build

In the **Project Explorer** panel, navigate to and expand **vital_signs_68xx_dss** → **Debug** directory. The project has been successfully built if the following files appear in the **Debug** folder:

- vital_signs_68xx_dss.bin
- vital_signs_68xx_dss.xe674

Build MSS Project

After the DSS project is successfully built, select **vital_signs_68xx_mss** in **Project Explorer**, right click on the project and select **Rebuild Project**.

Successful MSS Project Build

In the **Project Explorer** panel, navigate to and expand **vital_signs_68xx_mss** → **Debug** directory. The project has been successfully built if the following files appear in the **Debug** folder:

- vital_signs_68xx_mss.xer4f
- vital_signs_demo_68xx.bin

Build Fails with Errors

If the build fails with errors, please ensure that all the prerequisites are installed as mentioned in Software Requirements.

4. Execute the Lab

There are two ways to execute the compiled code on the EVM:

- Deployment mode: the EVM boots autonomously from flash and starts running the bin image
  - Using Uniflash, flash the **vital_signs_demo_68xx_mss.bin** found at `<PROJECT_WORKSPACE_DIR>\vital_signs_68xx_mss\Debug\vital_signs_demo_68xx_mss.bin`
  - The procedure to flash the EVM is the same as detailed in the Flash the EVM section.
- Debug mode: enables connection with CCS while lab is running; useful during development and debugging

Expand for help with Debug mode:

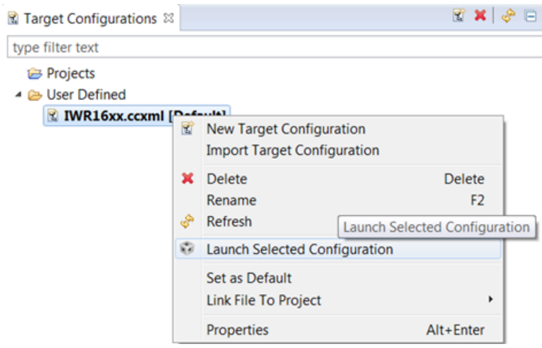
The CCS debug firmware (provided with the mmWave SDK) needs to be flashed once on the EVM.

- CCS Debug method is enabled by flashing the CCS Debug Firmware (provided with the mmWave SDK) using the methods covered in the Quickstart section.
- Use the following image instead

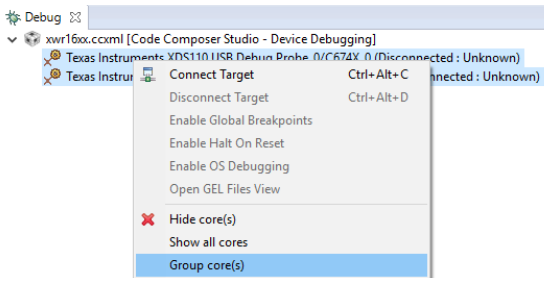
Image	**Location**	**Comment**
Meta Image 1	`C:\ti\mmwave_sdk_<ver>\packages\ti\utils\ccsdebug\xwr68xx_ccsdebug.bin`	Provided with the mmWave SDK

After the CCS debug firmware has been flashed, connect the EVM to CCS

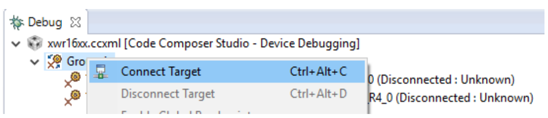
- Create a target configuration (skip to "Open the target..." if config already created previously in another lab for xwr68xx)
  - Go to **File** → **New** → **New Target Configuration File**
  - Specify an appropriate file name (ex: IWR68xx.ccxml) and check "**Use shared location**". Click **Finish**.
- In the configuration editor window:
  - Select **Texas Instruments XDS110 USB Debug Probe** for Connection
  - Select **IWR6843** device in the Board or Device text box.
  - Press the **Save** button to save the target configuration.
  - [Optional]: Press the **Test Connection** button to check the connection with the board.
- Open the target configuration window by going to **View** → **Target Configurations**.
  - Under **User Defined** configurations the target configuration previously created should appear.
  - Right click on the target configuration and select **Launch Select Configuration**. The target configuration will launch in the **Debug Window**.



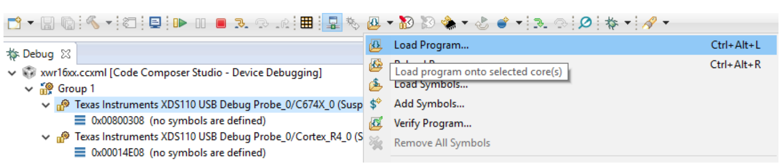
- Group cores and connect
  - Select both the **Texas Instruments XDS110 USB Debug probe/C674X_0** and **Texas Instruments XDS110 USB Debug probe/Cortex_R4_0** and then right click and select **Group core(s)**



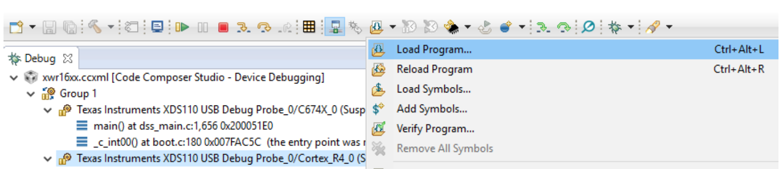
  - Select **Group 1** and then right click and select **Connect Target**



- Load the binary
  - Once both targets are connected, click on the C674X_0 target and then click **Load** button in the toolbar.



  - In the **Load Program** dialog, press the **Browse Project** button .
  - Select **vital_signs_demo_68xx_dss.xe674** found at `<PROJECT_WORKSPACE_DIR>\vital_signs_demo_68xx_dss\Debug\vital_signs_demo_68xx_dss.xe674` and press **Ok**.
  - Press **Ok** again in the **Load Program** dialog.
  - Repeat the above Load the Binary process for the Cortex_R4_0 target, selecting instead **vital_signs_demo_68xx_mss.xer4f** found at `<PROJECT_WORKSPACE_DIR>\vital_signs_demo_68xx_dss\Debug\vital_signs_demo_68xx_dss.xer4f`



- Run the binary
  - Select **Group 1**, press the **Run/Resume** button 

- The program should start executing and generate console output.

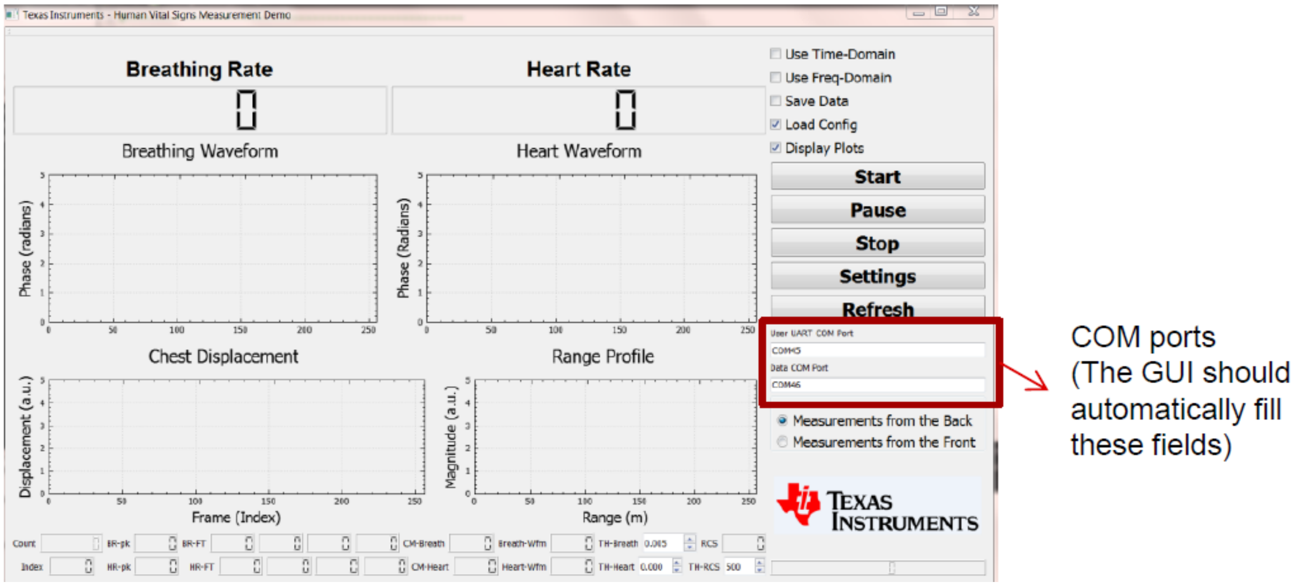
To run the lab, launch and configure the visualizer which displays the detection and tracked object data received via UART.

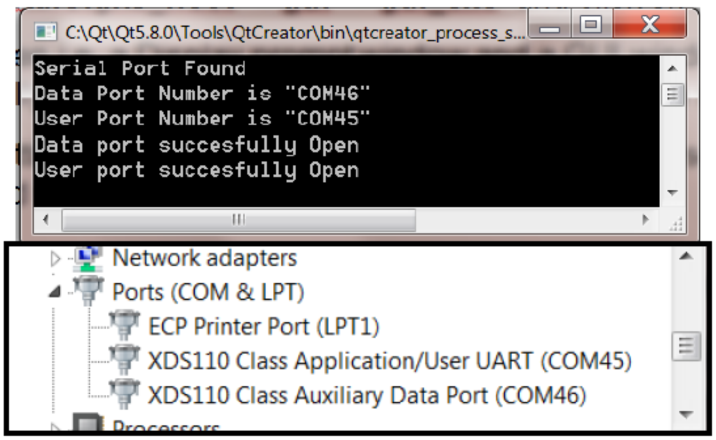
Note:

The visualizer can only run with the following:
Industrial mmWave Carrier Board (http://www.ti.com/tool/MMWAVEICBOOST) + IWR6843ISK Antenna Module (http://www.ti.com/tool/IWR6843ISK)
OR
+ IWR6843ISK-ODS Antenna Module (http://www.ti.com/tool/IWR6843ISK-ODS)
OR
+ IWR6843AOPEVM (http://www.ti.com/tool/IWR6843AOPEVM).

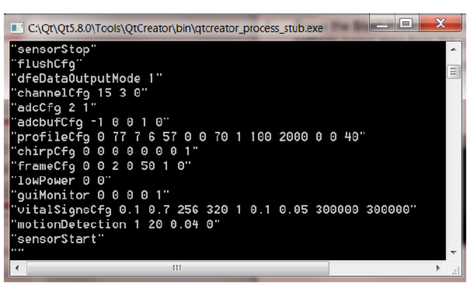
Launch the visualizer

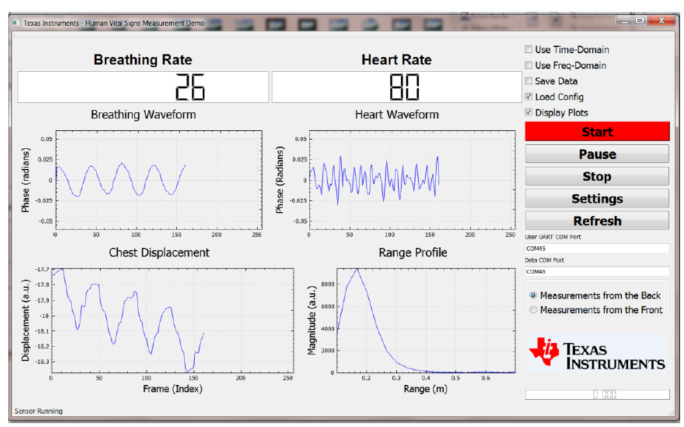
- Navigate to folder `<mmwave_industrial_toolbox_install_dir>\labs\vital_signs\68xx_vital_signs\gui\gui_exe`
- Run `VitalSignsRadar_Demo.exe`
- Two windows should open i.e. a Display prompt window and a GUI window. If the EVM is connected to the PC, then the display prompt window should successfully open the COM ports (to double check, make sure they match with the port numbers on the Device Manager).
- In the GUI window, the **User UART COM Port** and **Data COM Port** fields should automatically be filled with the correct port numbers (Make sure that no other EVM is connected to the USB ports of the PC)
- If the GUI does not open you might need the vc runtime which can be downloaded here (https://support.microsoft.com/en-us/help/2977003/the-latest-supported-visual-c-downloads)





- Press the **Start** Push button in the GUI. In the Display Prompt window you should see the configuration settings being read from the configuration text file and sent through the UART to the EVM
- As soon as the **sensorStart** command is sent, the GUI should start displaying the data





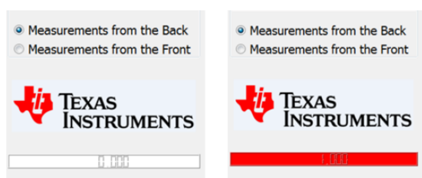
Configuring the GUI

- The mmWave sensor device and algorithm configurations are set through the configuration text file. These files are located in `<mmwave_industrial_toolbox_install_dir>\labs\vital_signs\68xx_vital_signs\gui\profiles`
- Configuration commands relevant to the vital signs algorithm are **vitalSignsCfg** and **motionDetection**
- **vitalSignsCfg**

Configuration	Parameters	Values	Comments
vitalSignsCfg	Start Range (meters)	0.1	The subject/person is expected to be within the Start Range and End Ranges. The program searches for the maximum peak within these ranges and assumes that peak corresponds to the subject.
	End Range (meters)	0.7	
	Breathing Waveform Size	256	Specifies the number of points within the waveforms. As an example, given a frame-rate of 20 Hz and 256 number of samples in the waveform then the time duration of the waveform would be = 256/20 ~ 12.8 seconds. In general, larger the time duration, better the frequency resolution after the FFT and higher the FFT processing gain. However, due to the inherent time-frequency resolution tradeoff, we lose the ability to measure instantaneous changes in the heart-rate and breathing-rate
	Heart-rate Waveform Size	320	
	Rx-Antenna to Process	1	Rx receiver number to process. Data from a single RX antenna is processed in the current implementation
	Alpha filter value for Breathing waveform energy computation	0.1	Alpha filter values for recursive averaging of the waveform energies based on the equation below where x(n) is the current waveform value while E(n) is the energy. $E(n) = \alpha x^2(n) + (1-\alpha)E(n-1)$
	Alpha filter value for heart-beat waveform energy computation	0.05	
	Scale Factor for breathing waveform	300000	Scaling factors to convert waveform values in floating points to 32 bit integers required for the FFT.
	Scale Factor for heart-beat waveform	300000	

- **motionDetection** The purpose of this block is to discard the data segments that might be corrupted by large amplitude movements. The heart waveform is divided into segment of L samples. If the energy within this data segment exceeds a user-defined threshold ETh then all the samples are discarded from the time-domain heart waveform.

Configuration	Parameters	Values	Comments
motionDetection	Enable	1	0: Disable the Block 1: Enable the Block
	Data segments Length (L)	20	Data segment over which the energy is computed
	Threshold (E_{TH})	0.04	Energy threshold value. If the energy in the data segment length exceeds this value then the data segment is discarded
	Gain Control	0	0: Disable Gain Control 1: Enable Gain Control



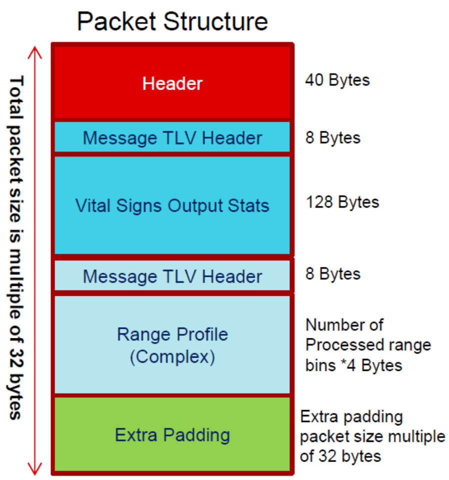
Data segment valid

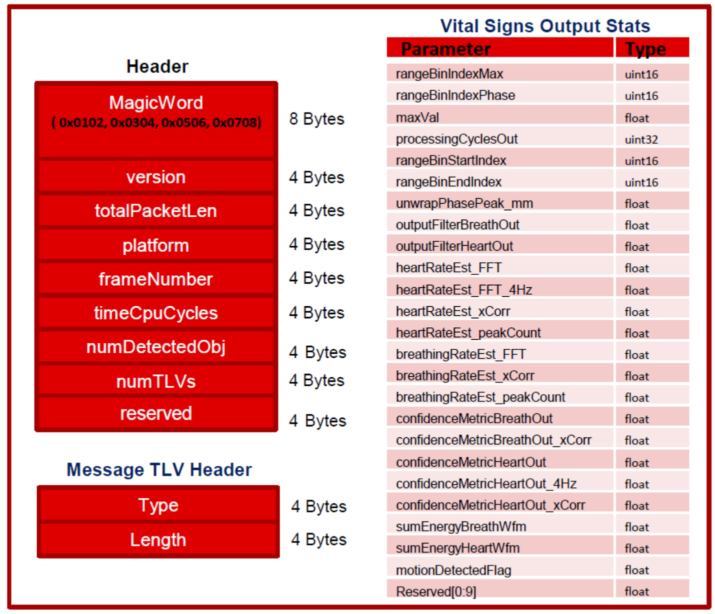
Data segment discarded

Note : The display panel below the TEXAS INSTRUMENTS icon will turn RED if the current data segment is discarded due to motion corruption.

Output Packet

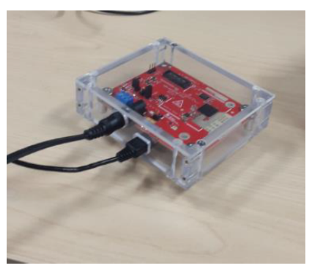
The format of the UART Data Stream is below:

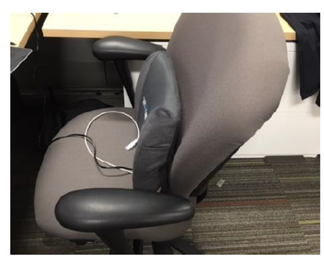




Sensor Placement

- For **Sensor in Front** the EVM shuld be mounted upright while subject should sit facing in a chair about .3-.8m in front.
- For initial measurement subject should stay still for 10-15 seconds to allow the application to calibrate. For subsequent measurements the subject must stay still for 5-10 seconds.
- For **Sensor in Back**, the EVM can be placed in the back of a car seat or in a seat cushion. This can be done by placing the EVM in an enclosure as shown below.



Visualizer Source Code

Source files are located at `<mmwave_industrial_toolbox_install_dir>\labs\vital_signs\68xx_vital_signs\gui\gui_source` .

Need More Help?

- Search for your issue or post a new question on the mmWave E2E forum (https://e2e.ti.com/support/sensor/mmwave_sensors/f/1023)