

# TI mmWave ROS Driver Setup Guide

---

## Table of Contents

Purpose .....	3
Objectives .....	3
Requirements.....	3
Hardware .....	3
Software.....	3
Host machine .....	3
mmWave EVM .....	3
Compatibility.....	3
Why ROS?.....	4
Installing ROS .....	4
Using the ti_mmwave_rospkg ROS driver .....	4
Building the TI mmWave ROS driver.....	4
User permissions.....	7
Running the driver .....	8
Visualizing the data .....	9
Reconfiguring the chirp profile .....	11
How it works .....	12
Nodes .....	12
ti_mmwave_rospkg.....	12
mmWaveQuickConfig .....	13
Services .....	13
mmWaveCLI .....	13
Messages.....	13
Frequently Asked Questions (FAQ).....	14
Tips for usage with an Ubuntu Virtual Machine .....	14
I see serial port errors when I start the driver or I do not see the mmWave EVM serial ports at /dev/ttyACM0 and /dev/ttyACM1 .....	15
How do I Re-flash the TI mmWave EVM with Out-of-Box demo? (only if needed).....	15

## Purpose

This document exists to aid TI mmWave Evaluation Module (EVM) users who wish to leverage the Robot Operating System (ROS) framework for their applications.

## Objectives

- Bring up ROS on a Linux machine
- Build the `ti_mmwave_rospkg` package in the ROS workspace
- Launch the `ti_mmwave_rospkg` node to start the mmWave EVM
- Visualize the Radar point cloud using Rviz
- Reconfigure the mmWave EVM with `mmWaveQuickConfig` node (optional)

## Requirements

### Hardware

- Host PC (laptop or desktop)
- TI mmWave 1443 or 1642 EVM
- USB to microUSB cable
- 5VDC power supply (center positive barrel jack, 2.5A)

### Software

#### Host machine

- Ubuntu 16.04 LTS operating system (native Linux machine preferred but will also work with Linux virtual machine running under Windows 7/Windows 10)
- Robot Operating System (ROS)
- `ti_mmwave_rospkg` source code

#### mmWave EVM

- TI mmWave Out-of-Box Demo firmware

## Compatibility

The `ti_mmwave_rospkg` source code has been tested with ROS Kinetic Kame on an x86\_64 machine with Ubuntu 16.04 LTS. It has also been tested under an Ubuntu 16.04 LTS VirtualBox Virtual Machine (VM) under Windows 7 and Windows 10. Please follow all of the instructions in the section titled “Tips for Usage with a Virtual Machine” if you are using an Ubuntu Virtual Machine.

## Why ROS?

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.

From a high level perspective the TI mmWave Out-of-Box demo provides these functions:

- Be able to specify desired FMCW Radar chirp profile through command line interface (CLI) on a UART port
- Do 1D, 2D, CFAR, Azimuth and Elevation processing and stream out intensity, relative radial velocity and three spatial coordinates (x,y,z) of the detected objects in real-time. Please note that the TI mmWave 1443 has an antenna array that can support both Elevation and Azimuth (x,y,z) estimation whereas the TI mmWave 1642 EVM antenna array only supports Azimuth (x,y).

Because the ROS framework and libraries support point cloud data, ROS is useful for developing and prototyping with the mmWave EVMs for robotics applications.

## Installing ROS

Before installing ROS make sure you have a properly working Ubuntu 16.04 LTS host machine. If you are using an Ubuntu virtual machine please go to the section titled “Tips for Usage with a Virtual Machine” and make sure to read/follow all of the instructions there before continuing.

To install ROS on your machine follow the instructions provided on the following webpage. Please select the full desktop (ros-kinetic-desktop-full) version.

<http://wiki.ros.org/kinetic/Installation/Ubuntu>

After installation, if you are new to ROS it will be helpful to read and complete the tutorials found at the following link to familiarize yourself with the framework.

<http://wiki.ros.org/ROS/Tutorials>

## Using the ti\_mmwave\_rospkg ROS driver

### Building the TI mmWave ROS driver

1. Follow the instructions on the following webpage to create a new ROS workspace:

<http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment>

NOTE: For convenience, the commands to source the required ROS environment setup scripts (“source /opt/ros/kinetic/setup.bash” and “source <workspace\_dir>/devel/setup.bash”) can be placed at the end of your ~/.bashrc file so they are executed automatically every time you open a new Terminal/shell. Otherwise, you will

need to source them each time you open a new Terminal/shell. The following commands can be used to add the “source” commands to the end of your ~/.bashrc file. In the second command below make sure to change “<workspace\_dir>” to your actual workspace directory.

Caution: Open this doc in a stand-alone PDF reader (i.e. Adobe Acrobat, etc.) if you are going to copy and paste commands directly into Linux. Copying the commands from a web browser sometimes adds extra newline characters that will cause the commands to fail when pasted into Linux.

```
$ echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
$ echo "source <workspace_dir>/devel/setup.bash" >> ~/.bashrc
```

After you run these commands to add the scripts to your ~/.bashrc file, you must close and re-open your Terminal/shell for them to take effect.

Please note that the script in the devel directory is specific to the workspace and you would need to change your ~/.bashrc file and re-open your shell in order to work in a different workspace.

2. Install the ROS ‘serial’ package:

- a) Use GIT to download (clone) the ROS serial driver into your <workspace\_dir>/src directory:


```
$ cd <workspace_dir>/src
$ git clone https://github.com/wjwwood/serial.git
```

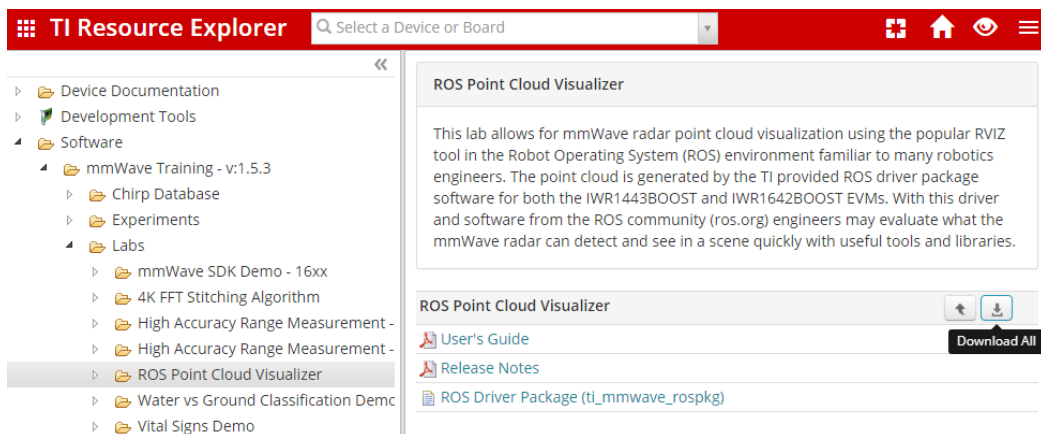
- b) More documentation on this library can be found at: <http://wjwwood.io/serial/>

3. Download the TI mmWave ROS Driver (ti\_mmwave\_rospkg) and copy it to your <workspace\_dir>/src directory:

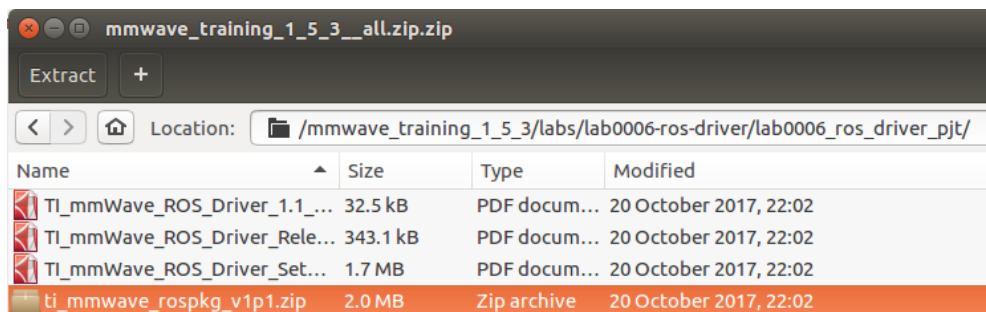
- a) Point your web browser to the following link and click on the “ROS Point Cloud Visualizer” lab in the table in the center or on the name of the folder on the left.

<http://dev.ti.com/tirex/#/?link=Software%2FmmWave%20Training%2FLabs>

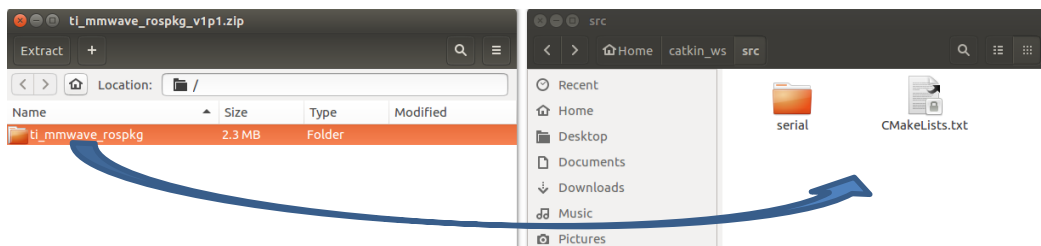
- b) Click on the “Download All” button  on the right side as shown in the following image and accept the user license agreement when prompted. Choose to open the downloaded zip file if prompted or save it to disk first and then open it by double-clicking it from the downloaded location.



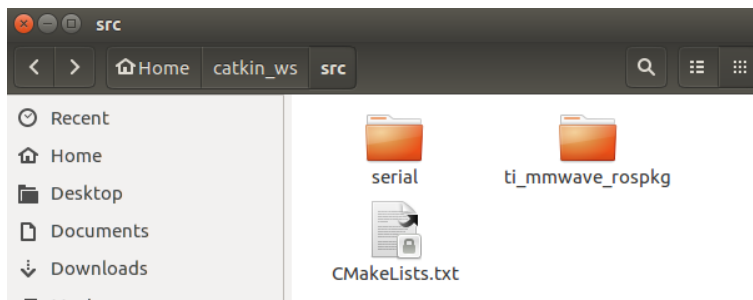
- c) Navigate into the folder structure of the opened zip file to the path shown in the following figure and then double-click on the `ti_mmwave_rospkg_<ver>.zip` file to open it.



- d) After opening the `ti_mmwave_rospkg_<ver>.zip` file, you should see the `ti_mmwave_rospkg` folder as shown in the left window in following screenshot. Open a new file browser and navigate to your `<workspace_dir>/src` directory as shown in the right window in the screenshot. Copy the `ti_mmwave_rospkg` folder to your `<workspace_dir>/src` directory by dragging it from the source (left) window to the destination (right) window.



- e) You should now see the `ti_mmwave_rospkg` folder in your `<workspace_dir>/src` directory as shown in the following screenshot.



#### 4. Build your project:

In the Terminal/shell, go to your `<workspace_dir>` directory and build the driver:

```
$ cd <workspace_dir>
$ catkin_make
```

If all of the installation steps were followed and both ROS environment scripts were sourced, the driver should build successfully and you should see the following at the end of the output.

```
[100%] Linking CXX executable
/home/username/catkin_ws/devel/lib/ti_mmwave_rospkg/ti_mmwave_rospkg
[100%] Built target ti_mmwave_rospkg
Scanning dependencies of target ti_mmwave_rospkg_generate_messages
[100%] Built target ti_mmwave_rospkg_generate_messages
```

### User permissions

Your user account must be a member of the “dialout” unix group in order to have access to the serial ports on Linux. To check if you are a member of this group run the “groups” command.

If you are already a member of the “dialout” group the output should list “dialout” (in addition to other groups).

If you are not a member of the “dialout” group, run the following command to add your user account to that group and then logout and login to Linux to make the change take effect:

```
$ sudo adduser <your_username> dialout
```

After executing this command please logout and login to Linux so the change will take effect.

## Running the driver

1. Close all previously opened Terminal windows/shells (since they may have old/stale environment variables)
2. If you are using a brand new TI mmWave EVM, it should come pre-flashed with the Out-of-Box Demo. If the EVM has been re-programmed with a different flash image or if you are running into issues you can follow the steps in the section “How do I Re-flash the TI mmWave EVM with Out-of-Box demo?” to re-flash the EVM with the Out-of-Box Demo.
3. Make sure the SOP2 jumper on the TI mmWave EVM has been removed and then power it using the 5VDC power supply and connect the EVM via USB to your host machine
4. If you are using an Ubuntu virtual machine (VM) running under Windows, whenever you connect or power-cycle the mmWave EVM you will need to enable the VM to access the mmWave EVM serial ports by going to the Devices menu along the top of the VM window, going to Devices->USB and selecting the Texas Instruments XDS110 option as shown in the picture in the “Tips for usage with an Ubuntu Virtual Machine” section. This is not necessary for native Linux machines.
5. Open a new Terminal window/shell. If you did not add the commands to source the two ROS environment scripts used during the “Setup a ROS Workspace” section to your ~/.bashrc file, then they should be manually sourced now using the following commands (make sure to change <workspace\_dir> to your actual workspace directory):

```
$ source /opt/ros/kinetic/setup.bash
$ source <workspace_dir>/devel/setup.bash
```

6. Launch the TI mmWave ROS node and Rviz visualizer as follows. (Please note that the roslaunch command will automatically start a “roscore” ROS master process if there is not one started already.)

For the TI mmWave 1443 EVM use the following command:

3-D configuration with elevation info:

```
$ roslaunch ti_mmwave_rospkg rviz_1443_3d.launch
```

2-D configuration without elevation info:

```
$ roslaunch ti_mmwave_rospkg rviz_1443_2d.launch
```

For the TI mmWave 1642 EVM (only supports 2-D config) use the command:

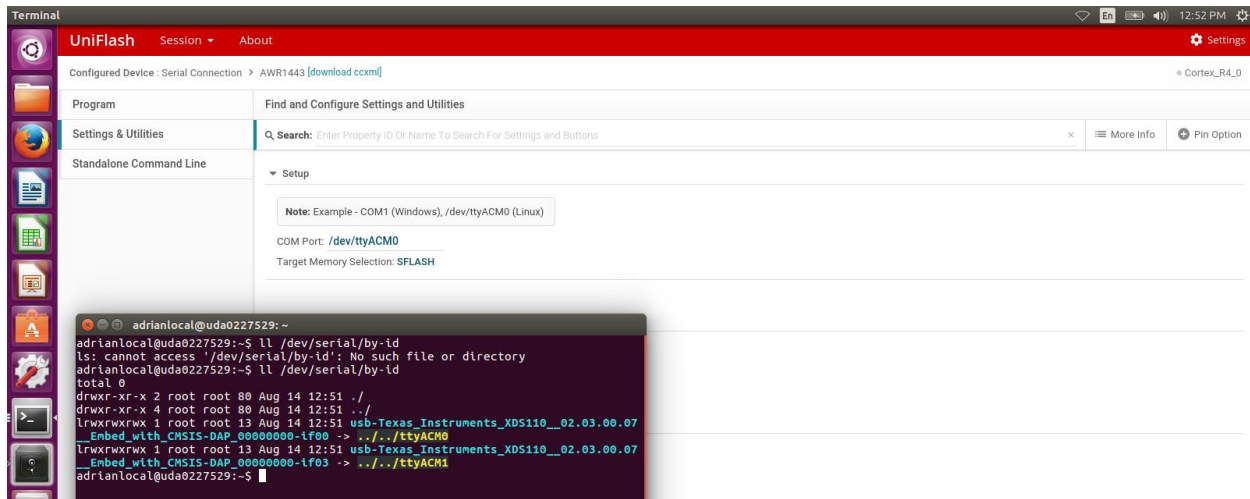
```
$ roslaunch ti_mmwave_rospkg rviz_1642_2d.launch
```

After up to 20 seconds the LED labeled ‘DS3’ on the EVM should light up (or pulse off then on if it was already lit up) and you should start seeing points being displayed in the Rviz visualizer window as shown in the pictures in the “Visualizing the Data” section. To stop the roslaunch command press CTRL-C in the Terminal window.



If the mmWave EVM was not running the Out-of-box demo firmware or did not successfully connect to the PC via the default Linux serial ports (/dev/ttyACM0 and /dev/ttyACM1) the command above will produce errors in the Terminal window and you will not see any points in the Rviz visualizer window. You will need to press CTRL-C in the Terminal window to stop the roslaunch command.

You can check that the mmWave EVM is connected and that both COM ports show up using the command “ll /dev/serial/by-id” from a Linux Terminal window as shown in the following image.



If you are already using the default (/dev/ttyACM0 and /dev/ttyACM1) ports for a different USB device or the TI mmWave EVM is connected to different ports you can open the file {\${WORKSPACE}}/src/ti\_mmwave\_rospkg/launch/ti\_mmwave\_sensor.launch and edit the following lines to use your own serial ports instead of the default ports:

```
<param name="command_port" value="/dev/ttyACM0" />
<param name="data_port" value="/dev/ttyACM1" />
```

Please see the “Frequency Asked Questions” section for more troubleshooting hints if needed.

## Visualizing the data

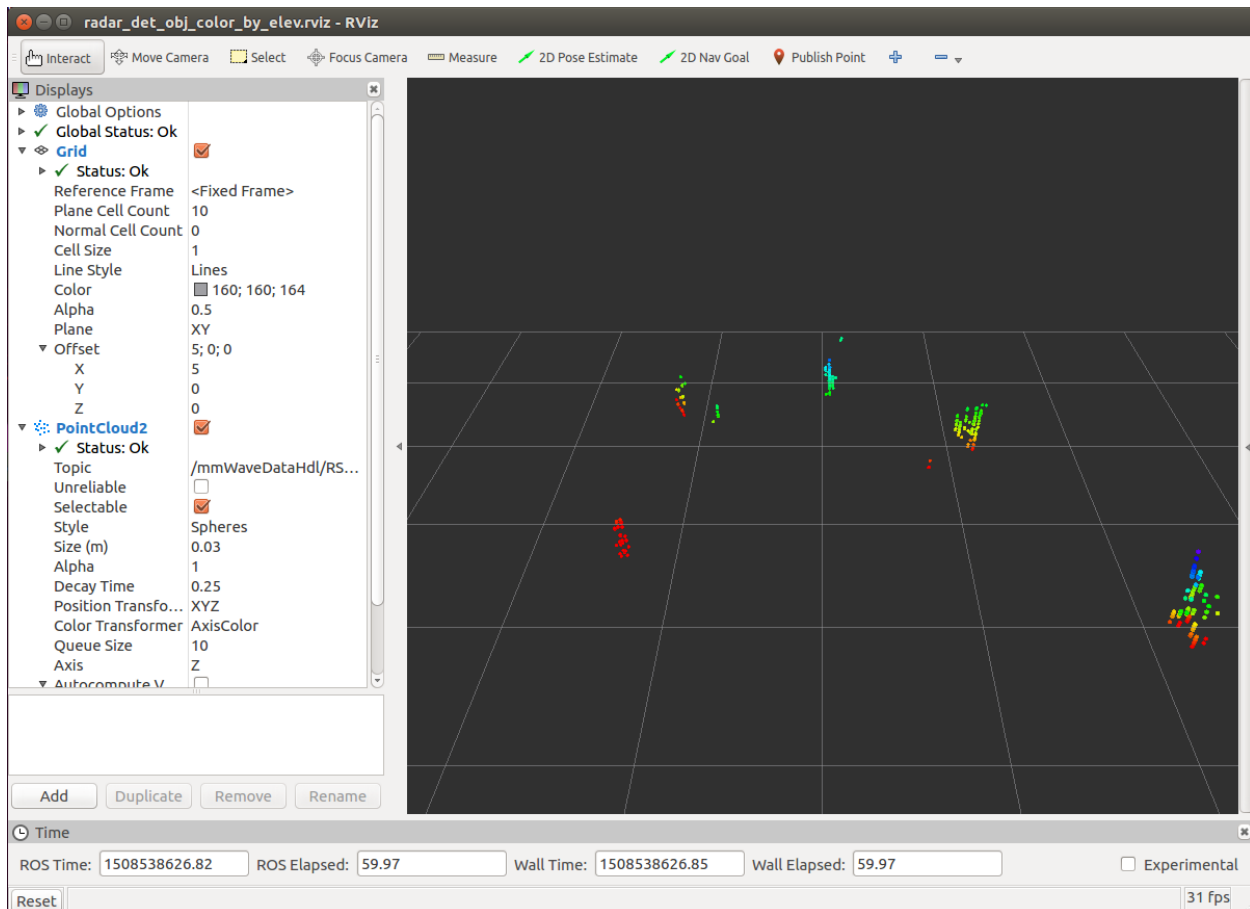
Once the sensor is running you should see the point cloud displayed in the Rviz visualizer window. The previous roslaunch command will automatically launch Rviz, but it can also be launched manually if needed using the following command in a new Terminal window/shell (after sourcing the ROS environment scripts).

```
$ rosrn rviz rviz
```

- Click Add->PointCloud2

- Select /mmWaveDataHdl/RScan from the “Topic” field dropdown for the PointCloud2 on the left hand panel

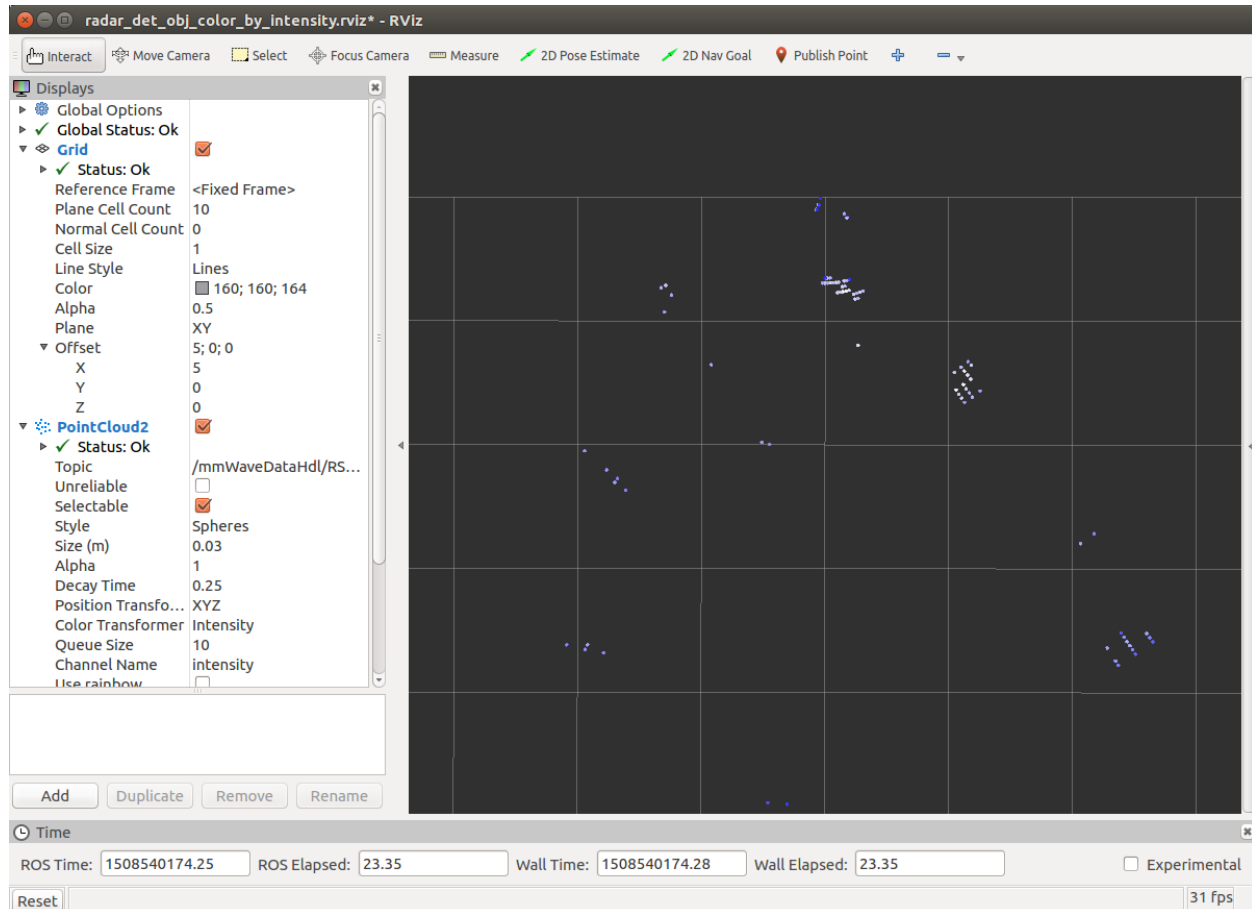
The following figure shows an example Rviz window for the 1443 sensor in a 3D configuration. In the initial perspective shown, the view is setup with the TI mmWave EVM in the back-middle edge of the grid facing toward you. The color of each point is based on its Z-axis elevation relative to the grid. You can move the perspective around by moving the mouse while holding down the left mouse button while the mouse is over the visualizer display. You can zoom in/out using the mouse scroll wheel or by holding down the right mouse button. Holding down the center scroll wheel allows panning.



The previous roslaunch command loads a default Rviz configuration but feel free to modify the visualization settings in the Rviz window as desired. The PointCloud2 point size and shape can be changed by setting the Size (m) parameter and Style parameter. The points can be colored based on elevation (used by default in the 3-D configurations) or by intensity (used by default in the 2-D configurations). Changing the Decay option to a non-zero value will cause Rviz to accumulate and display points over a specified number of seconds (it is set to 0.25 sec in the provided default configurations). Try changing this parameter and note the change in the displayed output.

The following figure shows an example Rviz window for the 1443 sensor in a 2D configuration (which looks similar to the 1642 sensor which always uses 2D due to the 1642 EVM antenna design). The initial

perspective is a top-down (bird's eye) view with the sensor at the top-middle of the grid facing downward. The color of each point represents the intensity of the reflection where dark blue represents the lowest intensity points displayed and white represents the highest intensity points being displayed. This is configurable in the Rviz settings for the PointCloud2.



## Reconfiguring the chirp profile

At this point, the mmWave EVM will be running one of the selected FMCW Radar chirp configurations provided with the driver. If you would like to reconfigure the mmWave device's chirp profile with a different set of parameters, you can create a new configuration using the mmWave Demo Visualizer (<https://dev.ti.com/mmWaveDemoVisualizer>) and save the corresponding cfg file using the "Save Config to PC" button. Make sure the device and mmWave SDK version selected in the GUI match your EVM and Out-of-Box firmware. You can send the configuration to the mmWave device with the following command:

```
$ rosrn ti_mmwave_rospkg mmWaveQuickConfig /path/to/profile.cfg
```

## How it works

### Nodes

#### **ti\_mmwave\_rospkg**

This node is a nodelet manager which is loaded with the two nodelets listed below when initiated. It must be instantiated with the parameters listed below.

#### *Parameters*

command\_port : Linux port used for the mmWave User UART port

command\_rate: Baud rate for the command\_port (115200)

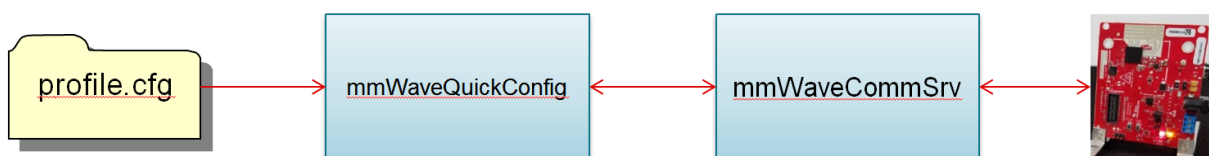
data\_port: Linux port used for the mmWave Data UART port

data\_rate: Baud rate for the data\_port (921600)

#### *mmWaveCommSrv Nodelet*

This nodelet provides the mmWave UART Command Line Interface as a ROS Service.

- Establishes a ROS Service which receives a CLI command and parameters in the form of a std\_msgs::string
- Uses 'serial' library (dependency) to open/read a COM port in Linux
- Sends the string out to the mmWave board over UART
- Returns the response from the EVM in the form of a std\_msgs::string
- mmWaveQuickConfig node uses this service to load a full .cfg file to the EVM

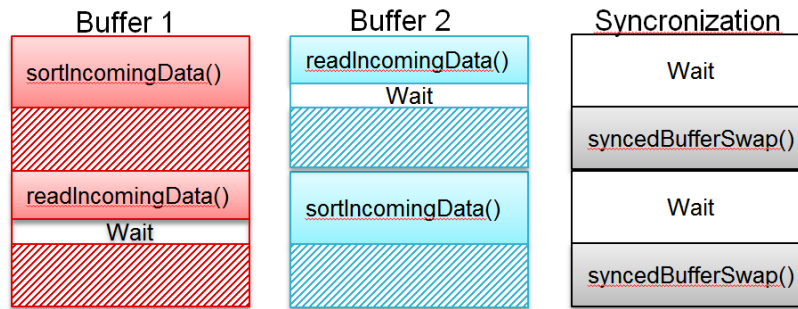


#### *mmWaveDataHdl Nodelet*

This nodelet reads and sorts the incoming radar data from the mmWave EVM.

- Uses 'serial' library to read incoming data from the mmWave device via a COM port on Linux
- Multi-Threaded Double Buffered read & sort of incoming packets
  1. readIncomingData thread : reads incoming data, looks for start-of-packet signal

2. sortIncomingData thread : sorts the incoming data based off of the package structure sent from the mmWave device
  3. syncedBufferSwap thread : when 1) and 2) have finished, swaps data buffers so the sorting thread can operate on new data
- Publishes output as a PointCloud<T> type ROS message



### mmWaveQuickConfig

This node will read a user specified radar chirp profile line-by-line and send each line to the mmWaveCommSrv nodelet to configure the mmWave EVM.

## Services

### mmWaveCLI

#### Requests

Std\_msgs::String comm

#### Response

Std\_msgs::String resp

## Messages

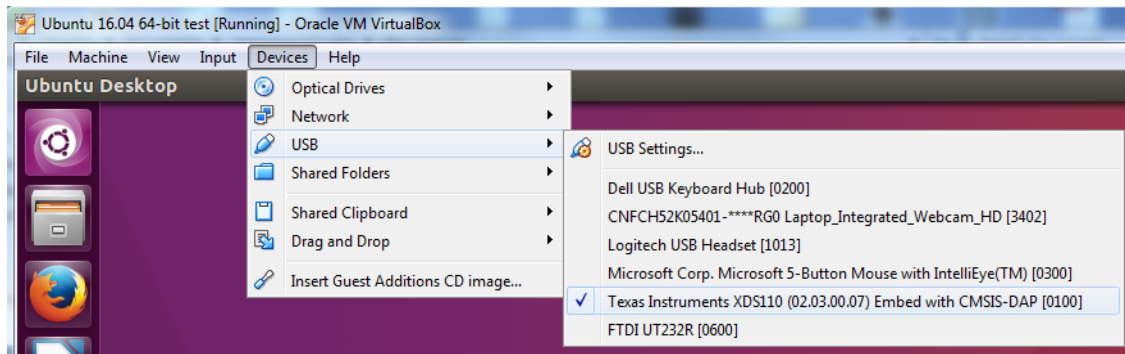
Sensor\_msgs::PointCloud2 /mmwaveDataHdl/RScan

## Frequently Asked Questions (FAQ)

### Tips for usage with an Ubuntu Virtual Machine

Here are some tips when using ROS and the TI mmWave ROS driver with an Ubuntu Virtual Machine (VM).

- The Windows PC should have more than 12GB of free hard drive space to install the VM and all required software on the VM
- The driver was successfully tested using an Ubuntu 16.04 LTS Virtual Machine running under Oracle Virtualbox ([www.virtualbox.org](http://www.virtualbox.org)) on Windows 7 and Windows 10. The following instructions are specific to Oracle Virtualbox but should be useful for other virtual machine managers as well.
- Download and install Oracle Virtualbox using the link above (to download click the Download button on the webpage and then click the download link for “Windows hosts”)
- Open Virtualbox and click the button to create a new virtual machine (VM)
- Create the VM in Virtualbox using the following settings:
  - Name: Ubuntu 16.04
  - If not already selected, use Type: “Linux” and Version: “Ubuntu (64-bit)” and click Next
  - Select at least 2048 MB for the RAM memory size and click Next
  - Choose “Create a Virtual Hard Disk now” when prompted and click Create
  - For the following prompts select “VDI”, “Dynamically allocated”, and increase the size to at least 12GB and then click Create
- Download (save to your hard drive) the Ubuntu 16.04 installer ISO image (1.5 GB) from the following link: <http://releases.ubuntu.com/16.04.3/ubuntu-16.04.3-desktop-amd64.iso>
- Start the newly created VM and point it to the downloaded ISO image when prompted for a startup disk
- Follow the prompts to install Ubuntu on the VM
  - It is not required to check the box to “Install third-party software”
  - Use the default selected option “Erase disk and install Ubuntu” when prompted
  - Press “Continue” when prompted to write changes to disk
  - Press “Restart Now” when prompted
- After logging into the Ubuntu VM using your newly created username and password you can open a Terminal window by clicking on the icon on the top of the left bar and searching for “Terminal” and then clicking on the Terminal icon (you can then right-click the Terminal icon on the left bar and choose to lock it to the Launcher bar if desired)
- Whenever you connect or power-cycle the mmWave EVM you will need to enable the VM to access the mmWave EVM serial ports by going to the Devices menu along the top of the VM window, going to Devices->USB and selecting the Texas Instruments XDS110 option as shown in the following picture:



- Enable the VM to access the internet by going to the “Devices->Network->Network Settings” menu at the top of the VM window and then select “Bridged Adapter” and pick your active network adapter so the VM can access the network
- If you are behind a corporate firewall, you may also need to setup your Linux proxy configurations to allow proper operation of GIT, web browsing and software installation using apt-get. Please contact your IT department for assistance if needed.
- Do not enable the “Display->Enable 3D Acceleration” option in the VM settings (it is normally disabled by default) or it may cause Rviz (ROS visualizer) to crash

### **I see serial port errors when I start the driver or I do not see the mmWave EVM serial ports at /dev/ttyACM0 and /dev/ttyACM1**

- Make sure the mmWave EVM is powered on and connected to the PC. It may take up to 20 seconds for Linux to completely initialize the serial ports for usage after the EVM is plugged in.
- If you are using a Virtual Machine, please make sure you have followed the steps listed in the section on tips for Virtual Machine usage regarding the USB serial port access.
- If the mmWave EVM is connected via different serial ports, follow the steps in the “Running the driver” section to modify the launch file that specifies the ports for the driver to use

### **How do I Re-flash the TI mmWave EVM with Out-of-Box demo? (only if needed)**

If you are using a brand new TI mmWave EVM, it should come pre-flashed with the Out-of-Box Demo. If the EVM has been re-programmed with a different flash image or if you are running into issues you can follow these steps to re-flash the EVM with the Out-of-Box Demo.

1. Download Uniflash, TI’s flashing software, via the following link:

[http://processors.wiki.ti.com/index.php/Category:CCS\\_UniFlash](http://processors.wiki.ti.com/index.php/Category:CCS_UniFlash)

2. Make sure the file “/lib/x86\_64-linux-gnu/libudev.so.0” exists. If it does not exist execute the following command to generate a symbolic link necessary for Uniflash to work:

```
$ sudo ln -sf /lib/x86_64-linux-gnu/libudev.so.1 /lib/x86_64-linux-  
gnu/libudev.so.0
```

3. Navigate to the download directory, make the file executable and run the Uniflash installer:

```
$ cd ~/Downloads  
$ chmod +x uniflash_sl.4.2.1490.run  
$ ./uniflash_sl.4.2.1490.run
```

**Note:** Your Uniflash executable filename may differ once a newer version becomes available

4. Download the Linux version of the mmWave SDK via the following link:

<http://www.ti.com/tool/mmwave-sdk>

5. Install the following mmWave SDK dependency:

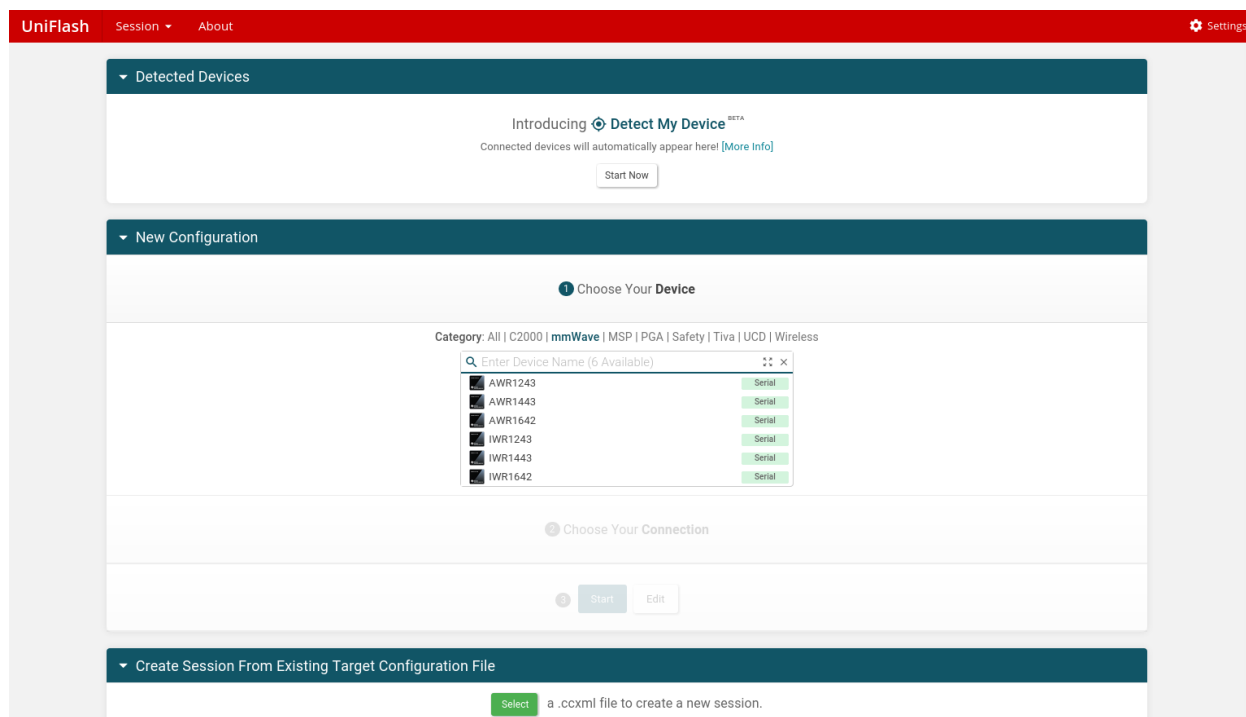
```
$ sudo apt-get install libc6:i386
```

6. Navigate to the download directory, make the file executable and run the mmWave SDK installer:

```
$ cd ~/Downloads  
$ chmod +x mmwave_sdk_01_00_00_05-Linux-x86-Install.bin  
$ ./mmwave_sdk_01_00_00_05-Linux-x86-Install.bin
```

7. Make sure there is no power to the mmWave EVM and place a jumper on the SOP2 header
8. Run Uniflash and you should see the following screen:





Click **mmWave** and in the list select your device. (We will proceed using the AWR1443 device.)

## 9. Specify the following flash images:

For xWR1443 EVM:

Meta Image 1/RadarSS:

`{$MMWAVE_SDK_INSTALL_DIR}/firmware/radarss/xwr12xx_xwr14xx_radarss.bin`

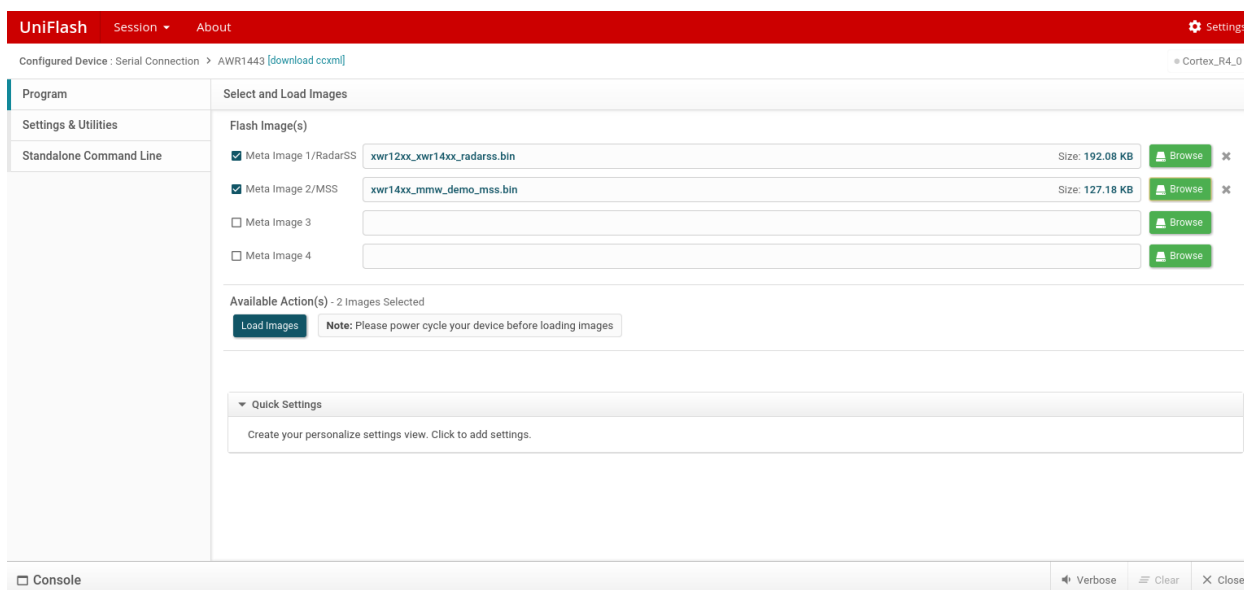
Meta Image 2/MSS:

`{$MMWAVE_SDK_INSTALL_DIR}/packages/ti/demo/{$YOUR_DEVICE}/mmw/xwr14xx_mmw_demo_mss.bin`

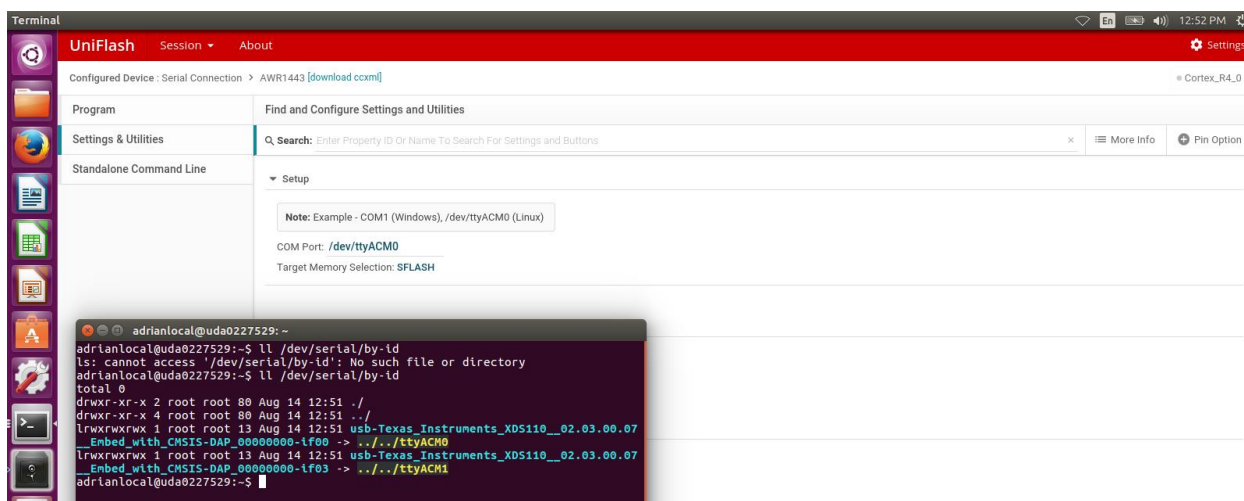
For xWR1642 EVM:

Meta Image 1:

`{$MMWAVE_SDK_INSTALL_DIR}/packages/ti/demo/{$YOUR_DEVICE}/mmw/xwr16xx_mmw_demo.bin`



10. Supply power to your mmWave EVM and connect the device to the computer with the microUSB cable.
11. Check that the mmWave EVM is connected and that both COM ports show up using the command “ll /dev/serial/by-id” from a Linux Terminal window as shown in the following image.



12. On the left-hand pane in Uniflash, select **Settings and Utilities**.
  - a. Enter the serial port corresponding to the first COM port (typically /dev/ttyACM0)
13. Select **Program** in the left-hand pane and then click the “Load Images” button. When finished, the console should report “Program Load completed successfully” as in the following image.

**UniFlash**
Session
About
Settings

Configured Device : Serial Connection > AWRI1443 [\[download ccmf\]](#)
» Cortex\_R4\_0 Disconnected: Halted

Program
Settings & Utilities
Standalone Command Line

Select and Load Images

Flash Image(s)

☒ Meta Image 1/RadarSS  Size: 192.08 KB

☒ Meta Image 2/MSS  Size: 127.18 KB

☐ Meta Image 3

☐ Meta Image 4

Available Action(s) - 2 Images Selected

Note: Please power cycle your device before loading images

Quick Settings

Create your personalize settings view. Click to add settings.

☐ Console

Verbose
Clear
Close

```

[8/14/2017, 1:05:31 PM] [INFO] Cortex_R4_0: Format on software was successful. Formatting on flash storage...
[8/14/2017, 1:05:31 PM] [INFO] Cortex_R4_0: ->Erasing storage [SFLASH]
[8/14/2017, 1:05:31 PM] [INFO] Cortex_R4_0: ->Sending Erase command to device...
[8/14/2017, 1:05:41 PM] [SUCCESS] Cortex_R4_0: ->Erase storage completed successfully!
[8/14/2017, 1:05:41 PM] [INFO] Cortex_R4_0: Downloading [RadarSS_BUILD] size [196692]
[8/14/2017, 1:06:24 PM] [INFO] Cortex_R4_0: SUCCESS!! File type RadarSS_BUILD downloaded successfully to SFLASH.
[8/14/2017, 1:06:24 PM] [INFO] Cortex_R4_0: Downloading [MSS_BUILD] size [130228]
[8/14/2017, 1:06:51 PM] [INFO] Cortex_R4_0: SUCCESS!! File type MSS_BUILD downloaded successfully to SFLASH.
[8/14/2017, 1:06:51 PM] [INFO] Cortex_R4_0: Downloading [CONFIG_INFO] size [164]
[8/14/2017, 1:06:51 PM] [INFO] Cortex_R4_0: SUCCESS!! File type CONFIG_INFO downloaded successfully to SFLASH.
[8/14/2017, 1:06:51 PM] [INFO] Cortex_R4_0: Disconnecting from device on COM port /dev/ttyACM0...
[8/14/2017, 1:06:51 PM] [INFO] Cortex_R4_0: Flashing instance clean-up initiated...
[8/14/2017, 1:06:51 PM] [INFO] Cortex_R4_0: Instance deinitialized!
[8/14/2017, 1:06:51 PM] [SUCCESS] Program Load completed successfully.

```

**Note:** If Uniflash stops responding after clicking “Load Images”, re-start it and make sure EVM is power cycled after Uniflash is started.

- After Uniflash has finished flashing the device remove the power supply, remove the SOP2 jumper and reapply power to the EVM.