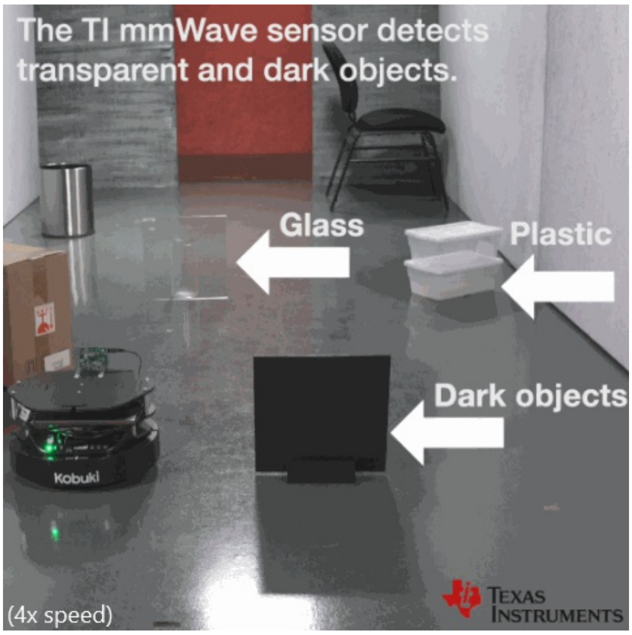
Autonomous Robotics with ROS for mmWave

This lab allows for the TI mmWave sensor to be used with popular mapping and navigation libraries in the Robot Operating System (ROS) environment, familiar to many robotics engineers. The lab uses the Octomap server and move_base libraries with TI's mmWave ROS Driver Package software to interface to the TI mmWave sensor. The lab supports use of IWR6843ISK or IWR6843AOP ES2.0 EVMs. With this TI driver and the software from the ROS community (ros.org) engineers may evaluate robot navigation and object avoidance quickly and easily.



Requirements

Prerequisite

> **Run Out of Box Demo**
>
> Before continuing with this lab, users should first run the out of box demo for the EVM. This will enable users to gain familiarity with the sensor's capabilities as well as the various tools used across all labs in the mmWave Industrial Toolbox.

Required and Supported mmWave Evaluation Modules (EVM)

ISK module with MMWAVEICBOOST

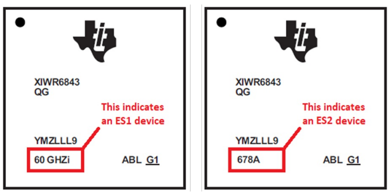
Quantity	Item
1	Antenna Module Board: IWR6843ISK
1	OPTIONAL: Industrial mmWave Carrier Board for CCS based debugging

OR

AOP EVM

Quantity	Item
1	IWR6843AOPEVM
1	OPTIONAL: Industrial mmWave Carrier Board for CCS based debugging

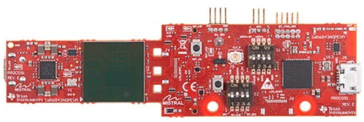
> **IWR6843 ES2.0 Only**
>
> This lab is only compatible with ES2.0 version of IWR6843.
>
> On ISK or ODS, check the device version on your IWR6843 using the on-chip device markings as shown below
>
> 1. If line 4 reads `678A`, you have an ES2 device. In this case, this lab is compatible with your EVM.
> 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/IWR6843ISK-ODS boards are orderable from the EVM link above.



On AOP, the EVM must be Rev F or later. This can be distinguished by the shape of the EVM if it is as shown 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.



Additional Hardware Requirements

Quantity	Item	Details
1	Robot	TurtleBot2 with plate and standoff kit
1	Computer (preferably laptop)	Running Linux Ubuntu 16.04. Used for remote operation and visualization
1	Laptop*	Used on Turtlebot2. Running Linux Ubuntu 16.04.
1	USB 2.0 printer-style cable (A-Male to B-Male)	Used to connect laptop to Turtlebot2
1	Micro USB Cable	Used to connect laptop to the mmWave EVM (cable comes with the EVM and should be connected to ‘XDS110 USB’ port on EVM)
1	12V to 5V DC to DC converter	Must be able to output at least 2.5Amps at 5V. Used to allow the EVM to be powered from the Turtlebot2 (this converter is required since the normal 5V output port on the TurtleBot2 cannot supply 2.5Amps)
1	2-pin miniFit JR connector/cable	Used to go from Turtlebot2 12V output port to the 12V input on the converter (for example, Molex cable part number 245135-0210 or 245135-0220 can be used by cutting it in half so the connector end goes to the TurtleBot2 12V output port and the cut wire end goes to the 12V input of the converter)
1	2.1mm barrel jack connector	Center positive with cable/wire to go from the 5V output on the converter to the EVM
	Misc. small bolts and nuts and brackets	For mounting mmWave sensor and DC converter to TurtleBot platform (not included with EVM or TurtleBot)

***Sitara AM572x processor can be used instead of a laptop for the Turtlebot2**

For instructions on how to implement the Sitara based alternative please refer to the Autonomous robotics reference design with Sitara™ processors and mmWave sensors using ROS. The reference design demonstrates an embedded robotic system where point-cloud data from the mmWave radar sensing is processed by the Sitara AM57x processor which runs Robot Operating System (ROS) and is the main processor for the overall system control.

Software

Tool	Version	Download Link
TI mmWave SDK	3.4.x.x	Link to Latest mmWave SDK. To access a previous version of the mmWave SDK scroll to the bottom of the table and click the link under "MMWAVE-SDK previous release". Repeat to continue stepping back to previous versions.

Tool	Version	Download Link
mmWave Industrial Toolbox	Latest	Download and install the toolbox. Go to Using TI Resource Explorer & the mmWave Industrial Toolbox for instructions.
Uniflash	Latest	Uniflash tool is used for flashing TI mmWave Radar devices. Download offline tool or use the Cloud version
Silicon Labs CP210x USB to UART Bridge VCP Drivers	Latest	Only needed for AOP EVM. https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers

Laptops

Both laptops need to have:

- Linux Ubuntu 16.04 natively installed (Ubuntu 16.04 Virtual Machine running on Windows can be used for remote control laptop if desired)
- ROS Kinetic Kame LTS with specified ROS packages
- TI mmWave ROS Driver
- Additional ROS packages supplied with this lab

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 + Antenna Module setup: Follow the instructions for Hardware Setup for Flashing in MMWAVEICBOOST Mode
- For IWR6843ISK in Standalone/Modular Mode: Follow the instructions for Hardware Setup of IWR6843ISK for Flashing Mode
- For AOP in Standalone/Modular Mode: Follow the instructions for Hardware Setup of IWR6843AOPEVM for Flashing Mode

2. Flash the EVM using Uniflash

Flash the binary listed below using UniFlash. Follow the instructions for using UniFlash

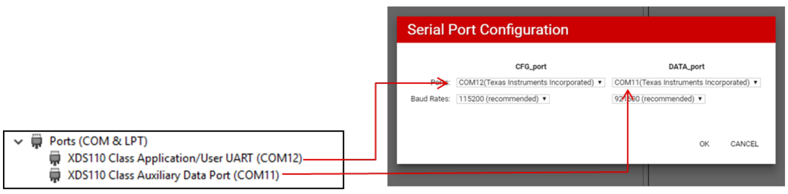
BIN Name	Board	Location
xwr68xx_mmw_demo.bin	IWR6843ISK	<INDUSTRIAL_TOOLBOX_INSTALL_DIR>\mmwave_industrial_toolbox_<VER>\out_of_box_demo\68xx_mmwave_sdk_dsp\prebuilt_binaries\xwr68xx_mmw_demo.bin
xwr64xxAOP_mmw_demo.bin	IWR6843AOPEVM	<INDUSTRIAL_TOOLBOX_INSTALL_DIR>\mmwave_industrial_toolbox_<VER>\out_of_box_demo\68xx_aop_mmwave_sdk_hwa\prebuilt_binaries\xwr64xxAOP_mmw_demo.bin

3. Setup the EVM for Functional Mode

- For MMWAVEICBOOST + Antenna Module setup: Follow the instructions for Hardware Setup of MMWAVEICBOOST + Antenna Module for Functional Mode
- For IWR6843ISK in Standalone/Modular Mode: Follow the instructions for Hardware Setup of IWR6843ISK for Functional Mode
- For AOP: follow the instructions for Hardware Setup of IWR6843AOPEVM for Functional Mode

4. Verification using online Visualizer

- Power up the EVM and connect it to the Windows PC with the provided USB cable (make sure that the SOP2 jumper is removed).
- Using Google Chrome, navigate to the following URL: https://dev.ti.com/mmWaveDemoVisualizer
- If prompted, follow the on-screen instructions for installing TI Cloud Agent (this is need the first time on a new PC)
- In the GUI menu, select Options → Serial Port
- In the serial port window, enter the appropriate port in each of the drop down menus based on your port numbers from the "flash the evm" section
- Click on Configure to connect the GUI to the EVM. The GUI Status bar should show **Conected:**



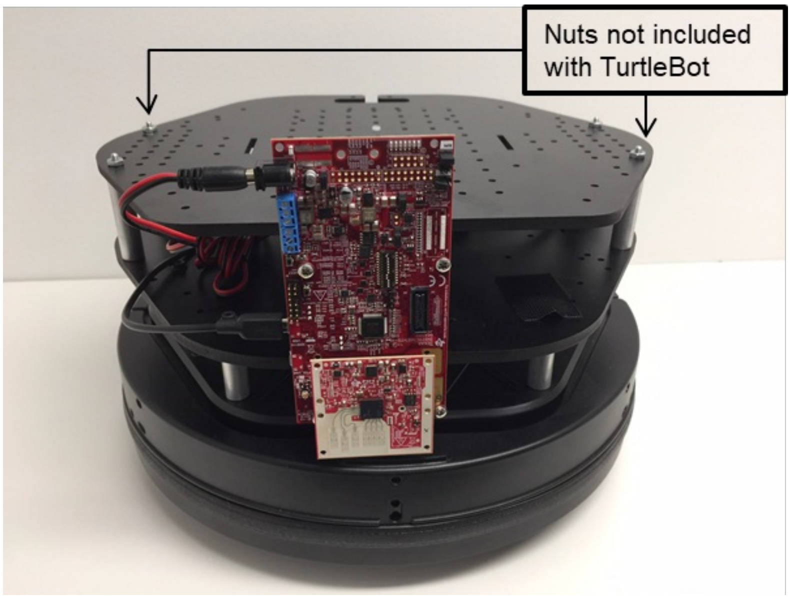
EVM COM Ports in Windows Device Manager

mmWave Demo Visualizer: Serial port configuration

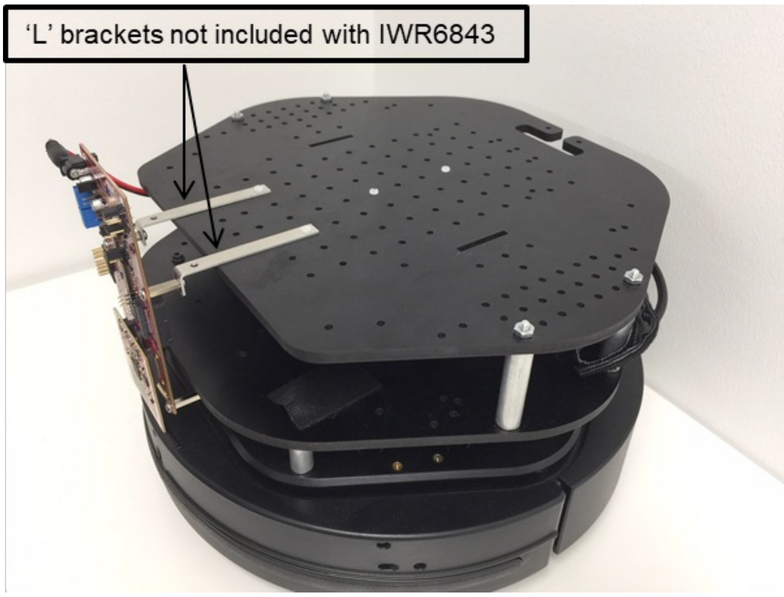
4. Setting up the TurtleBot2

- The TurtleBot2 is a low-cost, personal robotics platform that is well supported within the ROS community. There are many existing demos that work out-of-the-box with the TurtleBot2 including teleoperation (remote control), mapping, and navigation.
- In this guide we will have a look at how to modify these demos to integrate the TI mmWave sensor as the 3-D sensor.
- The TurtleBot2 should be assembled and mmWave EVM mounted as shown in the following pictures to work with this lab. There may be slight variation in mounting depending on EVM option.
- In the example shown, the 12V to 5V converter is mounted underneath the center of the top plate.
- The Turtlebot and EVM is connected to the laptop using USB cables

IWR6843ISK with MMWAVEICBOOST carrier board – front view (shown without required laptop):



IWR6843ISK with MMWAVEICBOOST carrier board – side view (shown without required laptop):



5. Installing ROS and the TI mmWave ROS Driver

- Please follow the instructions in the TI mmWave ROS Driver Setup Guide (available on the **TI Resource Explorer under Labs > TI mmWave ROS Driver**) to install ROS and the TI mmWave ROS Driver on each laptop before continuing.
- ROS must be installed on both the TurtleBot laptop and the Remote Control laptop.
- It is a good idea to test out the installation on each laptop by connecting the TI mmWave EVM and trying out the point cloud visualization.

6. Installing the mmWave Mapping and Navigation Demo Packages

After installing ROS and the TI mmWave ROS Driver, follow the steps below on both laptops for interchangeability. Install the Required Dependent ROS Packages

1. Install the following ROS packages which are required dependencies to run the demos:

```
$ sudo apt-get install ros-kinetic-navigation
```
```
$ sudo apt-get install ros-kinetic-hector-slam
```
```
$ sudo apt-get install ros-kinetic-octomap-server
```
```
$ sudo apt-get install ros-kinetic-kobuki
```
```
$ sudo apt-get install ros-kinetic-octomap-rviz-plugins
```
```
$ sudo apt-get install ros-kinetic-vision-opencv
```
```
$ sudo apt-get install ros-kinetic-depth-image-proc
```
```
$ sudo apt-get install ros-kinetic-joy
```

2. Download the source code for the specific version of the ROS fake_localization package required to run the demos:

```
$ mkdir -p ~/catkin_ws/src/navigation/fake_localization
```
```
$ cd ~/catkin_ws/src/navigation/fake_localization
```
```
$ wget https://raw.githubusercontent.com/ros-planning/navigation/1.14.2/fake_localization/fake_localization.cpp
```
```
$ wget https://raw.githubusercontent.com/ros-planning/navigation/1.14.2/fake_localization/CMakeLists.txt
```
```
$ wget https://raw.githubusercontent.com/ros-planning/navigation/1.14.2/fake_localization/package.xml
```

7. Download the TurtleBot mmWave Mapping and Navigation Packages

1. Download the **ti_mmwave_ros_map_nav_** <ver>**.tar.gz** Linux archive file from the Robotics / Autonomous Robotics with ROS for mmWave folder and copy it to your catkin <workspace_dir>/src directory.

> **Note**
>
> Downloading and extracting the TurtleBot mmWave Mapping and Navigation Packages should be done on both laptops

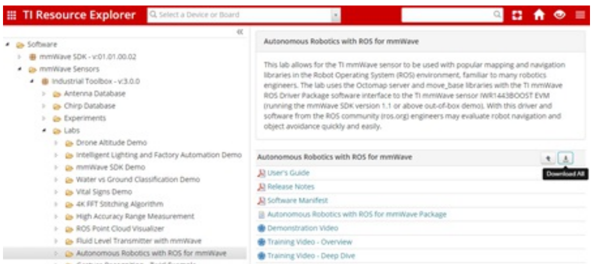
a) Point your web browser to the mmwave sensor software package: https://dev.ti.com/tirex/#/?link=Software

b) Click on "mmWave Sensors" and then click on "Industrial Toolbox"

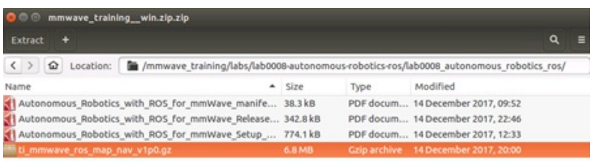
c) Click on the link for "Labs"

d) Click on the "Autonomous Robotics with ROS for mmWave" lab in the table in the center

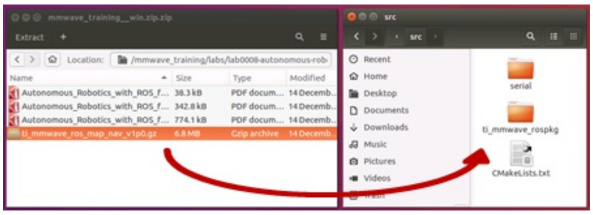
e) Click on the "Download All" button on the right side as shown in the following image and accept the user license agreement when prompted. Choose to save the downloaded zip file if prompted or save it to disk first and then open it by doubleclicking it from the downloaded location.



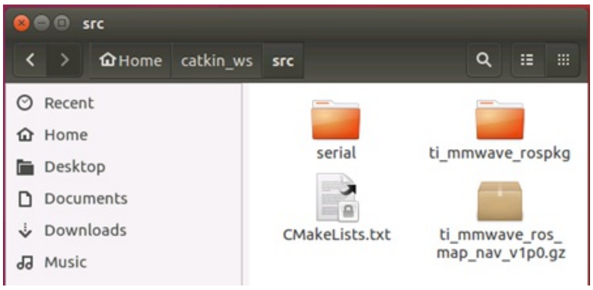
f) Navigate into the folder structure of the opened zip file to the path shown in the following figure. (Note that the path may start with mmwave_sensors/industrial_toolbox_ depending on package version.)



g) Open a new file browser window and navigate to your catkin <workspace_dir>/src directory as shown in the right window in the following screenshot. Copy the **ti_mmwave_ros_map_nav_** <ver>**.tar.gz** file to your <workspace_dir>/src directory by dragging it from the source (left) window to the destination (right) window.



h) You should now see the **ti_mmwave_ros_map_nav_**`<ver>`**.tar.gz** file in your `<workspace_dir>/src` directory as shown in the following screenshot.



1. Extract the turtlebot, navigation, and turtlebot_mmwave_launchers folders (as well as a custom mmWave chirp config file which gets placed in the ti_mmwave_rospkg/cfg directory) from the archive (.tar.gz) file using the following command executed from the `<workspace_dir>/src` directory. Change to match the actual filename.

```
$ tar xzf ti_mmwave_ros_map_nav_<ver>.tar.gz
```

1. Go back to your catkin `<workspace_dir>` directory and build the workspace:

```
$ 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 [100%] on the lines at the end of the build output.

8. Networking

ROS is a distributed system, meaning that it can communicate over a local network with other ROS components. For this demo, both laptops mentioned above must be on the exact same network and must be able to ping (icmp) each other by IP address. The Remote Control laptop must also be able to ssh (tcp/ip) into the TurtleBot laptop by IP address. You may need to install ssh on the laptops using the following command:

```
$ sudo apt-get install ssh
```

For more information regarding ROS's networking visit the link: http://wiki.ros.org/ROS/NetworkSetup

Additionally a ROS system may only have one "roscore" across all machines. In order for machines to recognize this "roscore" they must have an environment variable defined which specifies the IP address of the "roscore" machine.

On the TurtleBot machine

Edit your **~/.bashrc** file to include the following lines at the bottom:

```
export ROS_MASTER_URI=http://localhost:11311
export ROS_IP=<IP_OF_THIS_MACHINE>
```

You can check your IP by running `$ ifconfig` on the command line. Note that the line exporting the ROS_IP environment variable may not be required if your network is setup where each machine can contact/ping the other by hostname. You must close and re-open the shell for the updated **~/.bashrc** file to take effect.

On the Remote machine

Edit your **~/.bashrc** file to include the following lines at the bottom:

```
export ROS_MASTER_URI=http://<IP_OF_TURTLEBOT_MACHINE>:11311
export ROS_IP=<IP_OF_THIS_MACHINE>
```

You can check your IP by running `$ ifconfig` on the command line. Note that if your network is setup where each machine can contact/ping the other by hostname then you can use

instead of in the first line and the line exporting the ROS_IP environment variable may not be required. You must close and re-open the shell for the updated **~/.bashrc** file to take effect.

9. Remote Control Demo (TurtleBot Bring-up and Teleoperation)

These steps must take place from the remote machine, "ssh-ing" into the TurtleBot laptop when necessary.

TurtleBot Bring-up

To start the TurtleBot, open a terminal window on the remote machine, ssh into the TurtleBot laptop and run the following command.

- For IWR6843ISK:


```
$ roslaunch turtlebot_bringup minimal.launch mmwave_device:=6843ISK
```

- For IWR6843AOP:


```
$ roslaunch turtlebot_bringup minimal.launch mmwave_device:=6843AOP
```

If the EVM was not in a good state the roslaunch command will fail. Try resetting the EVM by pressing the 'NRST' button on the EVM and then run the desired roslaunch command again. After the Turtlebot and mmWave sensor are configured, you may see periodic "Kobuki : malformed subpayload detected" errors. These errors appear to come from the Turtlebot driver and do not affect the operation of the demo.

Teleoperation (Remote Control)

Open a new terminal window on the remote machine and run the following command to bring up the teleoperation (remote control) of the TurtleBot:


```
$ roslaunch turtlebot_teleop keyboard_teleop.launch
```

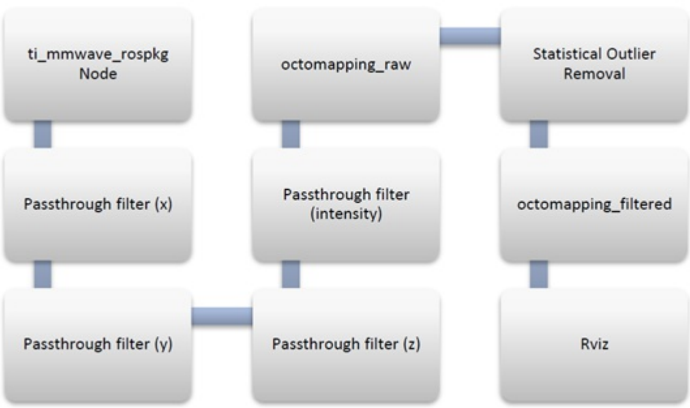
Follow the instructions shown in the window to control the TurtleBot. You can exit out of the remote control application by pressing `CTRL-C`.

10. Mapping Demo

The Mapping Demo is an example of how to use TI's mmWave Radar EVMs within the ROS framework on a robot to build a map.

- The demo runs the octomap_server package in ROS. There are several filters that have been brought up for use: Pass Through filters for all the point cloud fields, and a Statistical Outlier Removal filter for filtering the raw map.
  - **Pass Through Filters**: remove values outside a certain range for a given field (X, Y, Z, intensity)
  - **Statistical Outlier Removal**: removes values based off their distance from their closest neighbors
- The parameters/limits for these filters can be found and modified in the **turtlebot_mmwave_launchers/launch/radar_limit_filters.launch** file. The **radar_mapping.launch** file mentioned below must be re-started after changing the **radar_limit_filters.launch** file in order for the new parameters to take effect.

The processing graph for the incoming point cloud data is shown below:



Start-up

To run the mapping demo, first follow the TurtleBot Bring-up and Remote Control instructions in the "TurtleBot Bring-up and Teleoperation (Remote Control)" section to bring up and remote control the TurtleBot. Then, to run the mapping demo open a new terminal window, ssh into the TurtleBot laptop and run the following command:


```
$ roslaunch turtlebot_mmwave_launchers radar_mapping.launch
```

You may see a few warnings immediately after the launch file is run. It may also output a warning saying "Nothing to publish, octree is empty" whenever there are no objects detected in front of the mmWave sensor.

Visualization

To view the TurtleBot, Radar data, and map data in Rviz open a new terminal window on the remote machine and run the following command which will open a pre-defined Rviz configuration customized for the mapping demo (the command is all one line):

```
$ rosrun rviz rviz -d ~/catkin_ws/src/turtlebot_mmwave_launchers/launch/mapping_visualization.rviz
```

Alternatively, you can run the following command to open a blank Rviz screen and then manually add the same topics to the visualization as follows:

```
$ rosrun rviz rviz
```

- Once Rviz has started, add the radar data by selecting Add-> PointCloud2 and selecting /mmWaveDataHdl/RScan under the Topic dropdown menu for the PointCloud2.
    - Make the following changes to improve the visualiation:
        - Size to 0.03
        - Style to Spheres
        - Decay to 0.25
- To visualize the TurtleBot on the screen select Add->Robot Model and Rviz will automatically detect the robot description and display the TurtleBot.
- To view the path the TurtleBot is taking select Add->Path and choose /trajectory under the Topic dropdown for the Path.
- To visualize the octomap output in Rviz, select Add->PointCloud2 and select `<raw_or_filtered>_point_cloud_centers`
    - To color points by elevation change ColorTransformer to AxisColor.
    - Changing the Size to 0.03 and Style to Spheres will improve the visualization.

You can also save a custom Rviz configuration and load it in the future for convenience.

Saving a Map

- The TurtleBot laptop is now running the Radar and TurtleBot drivers, as well as all the filtering and mapping nodes shown in the flow chart above.
- Move the TurtleBot around your environment manually until you are satisfied with your map.
- Then open a new terminal window on the remote machine, ssh into the TurtleBot laptop, and run the following command to save the map:

```
$ rosrun octomap_server octomap_saver -f <your_file_name>.bt
```

Viewing a Previously Saved Map

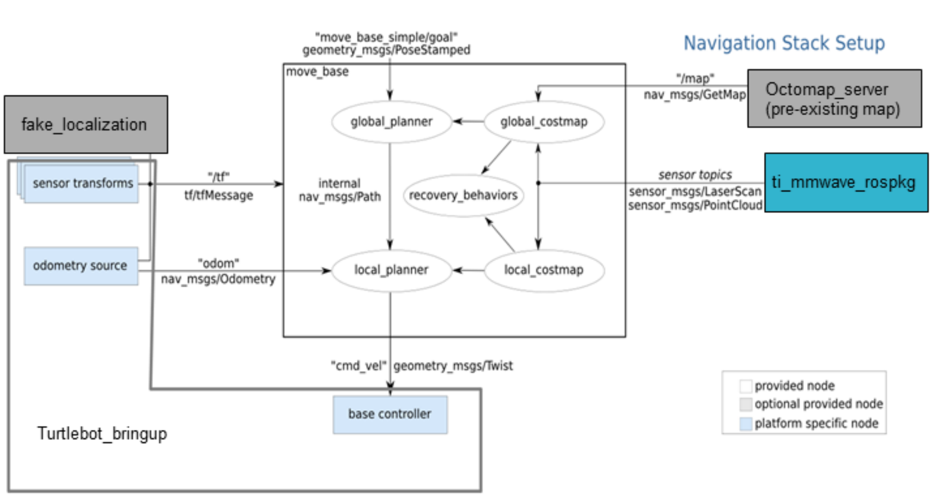
- To view a saved `<map>.bt` file, you must first shutdown the mapping demo launch file if it was launched by pressing CTRL-C in the window where the radar_mapping.launch file was launched.
- Rviz will need to be opened and configured as mentioned in the Visualization section above if it is not already open.
- Then, open a new terminal window on the remote machine, ssh into the TurtleBot laptop, and run the following command to serve the saved map (the command is all one line):

  ```
  $ rosrun octomap_server octomap_server_node /path/to/<map>.bt octomap_point_cloud_centers:=filtered_point_cloud_centers
  ```

  You should see the saved map displayed in Rviz. Note that the other topics in Rviz may show a warning/error since they are not active.

11. Navigation Demo

The TurtleBot navigation demo runs on the nodes mentioned in the image below.



Here are the steps to run the navigation demo.

1. Close all previous terminal windows if any were open
2. Open a new terminal window on the remote machine, ssh into the TurtleBot laptop and bring up the TurtleBot and mmWave EVM with the following command.
   - For IWR6843ISK:

     Select text

     ```
     $ roslaunch turtlebot_bringup minimal.launch mmwave_device:=6843ISK
     ```

   - For IWR6843AOP:

     Select text

     ```
     $ roslaunch turtlebot_bringup minimal.launch mmwave_device:=6843AOP
     ```

If the EVM was not in a good state the roslaunch command will fail. Try resetting the EVM by pressing the 'NRST' button on the EVM and then run the desired roslaunch command again.

The configuration files are included in the ti_mmwave_ros_map_nav_v1p4.tar.gz package ...\ti_mmwave_ros_map_nav_v1p4\ti_mmwave_rospkg\cfg. To change the parameter, edit and save the file using a text editor. For example, to update the demo to run using the IWR6843ISK-ODS and configuration file used with the ODS point cloud demo

- Edit the "6843ISK_3d.cfg" configuration file to one that works with the ODS point cloud demo.
- Flash the IWR6843ISK-ODS with the ODS point cloud binary.
- Run the demo as documented for 6843ISK EVM.

1. To bring up the move_base and fake_localization nodes and load a prebuilt map using the octomap_server, open a new terminal window on the remote machine, ssh into the TurtleBot laptop and run the following command:

   Select text

   ```
   $ roslaunch turtlebot_mmwave_launchers radar_navigation.launch
   ```

   Note 1: By default, this launch file loads a specific prebuilt map file containing a map of a simple rectangular space roughly 4ft x 6ft which can be used to make the robot stay within a space of that size for demo and testing purposes.

   - To load your own map, edit the radar_navigation.launch file and change the map filename shown in bold below to your own saved map file.
     *node name="octomap_server" pkg="octomap_server" type="octomap_server_node" args=***"$(find turtlebot_mmwave_launchers)/launch/map_4ft_by_6ft_border_large.bt** *projected_map:=map" />*
   - Alternatively, this static map can be completely disabled/removed as follows if desired to allow the robot to plan paths freely without any artificial boundaries.
     - Remove the line containing "costmap_2d::StaticLayer" in the global_costmap_params.yaml and local_costmap_params.yaml param files
     - Remove the line containing "octomap_server_node" in the ~catkin_ws/src/turtlebot_mmwave_launchers/launch/radar_navigation.launch file
     - The size of the global and local costmaps can also be increased in the global_costmap_params.yaml and local_costmap_params.yaml param files to allow setting goals that are farther away

   Note2: You may see "octree is empty" warnings when there are no objects detected in front of the mmWave sensor.

2. Open a new terminal window on the remote machine and run the following command which will open a pre-defined Rviz configuration customized for the navigation demo.
   - To show the inflation layer which is used for path planning (the command is all one line):

     Select text

     ```
     $ rosrun rviz rviz -d ~/catkin_ws/src/turtlebot_mmwave_launchers/launch/navigation_visualization.rviz
     ```

   - To not show the inflation layer which is used for path planning (the command is all one line):

     Select text

     ```
     $ rosrun rviz rviz -d ~/catkin_ws/src/turtlebot_mmwave_launchers/launch/navigation_visualization_2.rviz
     ```

- Alternatively, you can run the following command to open a blank Rviz screen and then manually add the same topics to the visualization as follows:

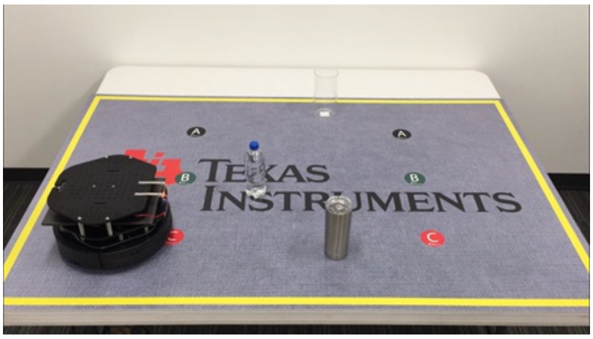
```
$ rosrun rviz rviz
```

  - "Add" three Map displays, one PointCloud2, the Robot Model, and one PosewithCovariance. Use the Topic dropdowns on the displays you added in Rviz to select topics for each one.
    - For the maps, choose "map" for one and the local and global cost maps for the other two.
    - Select /initialpose for PosewithCovariance.
    - You can select /mmWaveDataHdl/RScan or /xyzi_filt_out for the PointCloud2.
      - Changing the Size to 0.03, Style to Spheres, and Decay to 0.25 will improve the PointCloud2 visualization.
    - You can also add two Path topics (one for /move_base/DWAPlannerROS/local_plan and one for /move_base/DWAPlannerROS/global_plan) and a Polygon for /move_base/local_costmap/footprint.
- There is also an alternate pre-defined Rviz configuration file as an additional visualization example located at "turtlebot_mmwave_launchers/launch/navigation_visualization_2.rviz" that adds a PointCloud for the voxel grid (/move_base/local_costmap/obstacle_layer/marked_cloud) and disables the costmaps.

3. Start the navigation by first providing an initial pose estimate. Select 2D Pose Estimate (along the top of screen) and click on the location where the TurtleBot is within the map and drag in the direction it is facing. You should see the TurtleBot appear on the same spot you click immediately after releasing. Note: Please do not choose a starting position or goal that is too close to the boundaries of the current map as the navigation stack will not be able to create what it considers to be a valid path.

4. Now, give the TurtleBot a navigation goal by selecting 2D Nav Goal and clicking the location you would like the TurtleBot to navigate towards and dragging in the direction you would like it to face. The TurtleBot should then begin navigation to its goal. If you need to stop it, terminate the radar_navigation.launch roslaunch command by clicking on the terminal window that was used to launch it and pressing CTRL-C.

5. As an alternative to manually specifying the initial pose and goal using Rviz, there is also an example shell script in the "turtlebot_mmwave_launchers/scripts" directory that you can call from the Linux command prompt to set the initial pose and goal. It is intended to be used with the default 4ft x 6ft prebuilt map where you start the robot on one of three starting points (a, b, or c) at one end of the rectangular space facing the opposite end and then send it to one of the three goal points at the other end. This is useful for demonstrating obstacle avoidance in a 4ft x 6ft space as shown in the following picture. To run the script, go to the "turtlebot_mmwave_launchers/scripts" directory and type "./start_nav.sh" at the Linux prompt.



Need More Help?

- Fort Turtlebot2 issues and questions: http://wiki.ros.org/Robots/TurtleBot#Robots.2BAC8-TurtleBot.2BAC8-kinetic.TurtleBot2
- For ROS related questions: https://answers.ros.org/questions/
- Additional resources in the documentation of the mmWave SDK (note hyperlinks will only work if the mmWave SDK has been installed on PC):
  - mmWave SDK Module Doc located at <mmwave_sdk_install_dir>/docs/mmwave_sdk_module_documentation.html
  - mmWave SDK User's Guide located at <mmwave_sdk_install_dir>/docs/mmwave_sdk_user_guide.pdf
  - mmWave SDK Release Notes located at <mmwave_sdk_install_dir>/docs/mmwave_sdk_release_notes.pdf
- Find answers to common questions on mmWave E2E FAQ
- Search for your issue or post a new question on the mmWave E2E forum