

ARPG Goggles - A radar-based velocity estimation ROS node

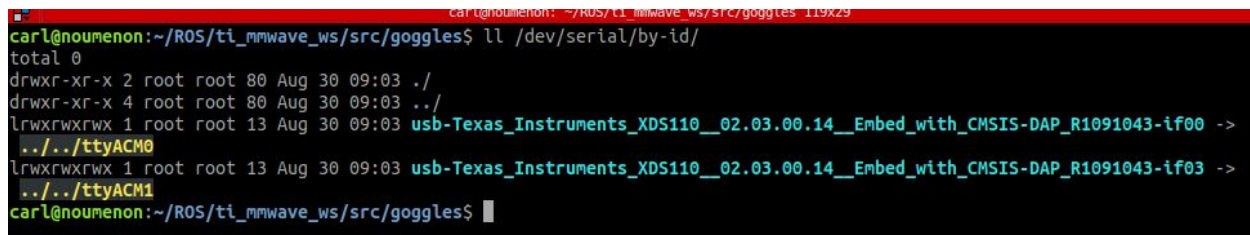
Carl Stahoviak & Andrew Kramer

carl.stahoviak@colorado.edu

andrew.kramer@colorado.edu

NOTE: These instructions assume that the TI AWR1843BOOST is recognized as a USB device. This can be verified by running the following command in the terminal:

```
ll /dev/serial/by-id
```



```
carl@noumenon: ~/ROS/ti_mmwave_ws/src/goggles: ll /dev/serial/by-id/
total 0
drwxr-xr-x 2 root root 80 Aug 30 09:03 ./
drwxr-xr-x 4 root root 80 Aug 30 09:03 ../
lrwxrwxrwx 1 root root 13 Aug 30 09:03 usb-Texas_Instruments_XDS110_02.03.00.14_Embed_with_CMSIS-DAP_R1091043-if00 ->
.././ttyACM0
lrwxrwxrwx 1 root root 13 Aug 30 09:03 usb-Texas_Instruments_XDS110_02.03.00.14_Embed_with_CMSIS-DAP_R1091043-if03 ->
.././ttyACM1
carl@noumenon: ~/ROS/ti_mmwave_ws/src/goggles$
```

If the AWR1843BOOST is not properly recognized (the board is not recognized as a Texas Instruments XDS110 device), please email Carl Stahoviak at carl.stahoviak@colorado.edu

The steps below take you through the following process:

1. Cloning the ARPG `ti_mmwave_ropkg`, `goggles` and `serial` ROS packages into a catkin workspace
2. Updating the firmware on the AWR1843BOOST. This firmware update patches a known bug in the mmWave SDK version 3.2.
3. Launching the Goggles Node. This will launch both the `/mmWaveDataHdl/RScan` (pointcloud2) topic and the `/mmWaveDataHdl/vel` (TwistWithCovarianceStamped)

Cloning the Goggles and Associated Repos

1. In the `src` folder of a catkin workspace, clone the following repos:

```
git clone https://github.com/arpag/goggles.git
git clone https://github.com/arpag/ti_mmwave_ropkg.git
git clone https://github.com/wjwood/serial.git
```

2. Before running `catkin_make`, install the following: `sudo apt install libceres-dev`

Updating the Demo Visualizer Firmware

1. Download and install TI Uniflash: <http://www.ti.com/tool/download/UNIFLASH>

1a. Click on the 'Linux installer for Uniflash' link to download the installer file
1b. Find the `uniflash_sl.5.1.0.XXXX.run` file in the Downloads folder and make it an executable: `chmod +x uniflash_sl.5.1.0.XXXX.run`
1c. Run the Uniflash installer file: `./uniflash_sl.5.1.0.XXXX.run`
1d. At this point, if you get an error (I don't have a screenshot of this error) related to a missing libusb library, this error can be resolved by installing the libusb library: `sudo apt install libusb-dev` (do NOT install Uniflash without resolving any errors that pop up during the installation process)

2. Update the firmware on the AWR1843BOOST.

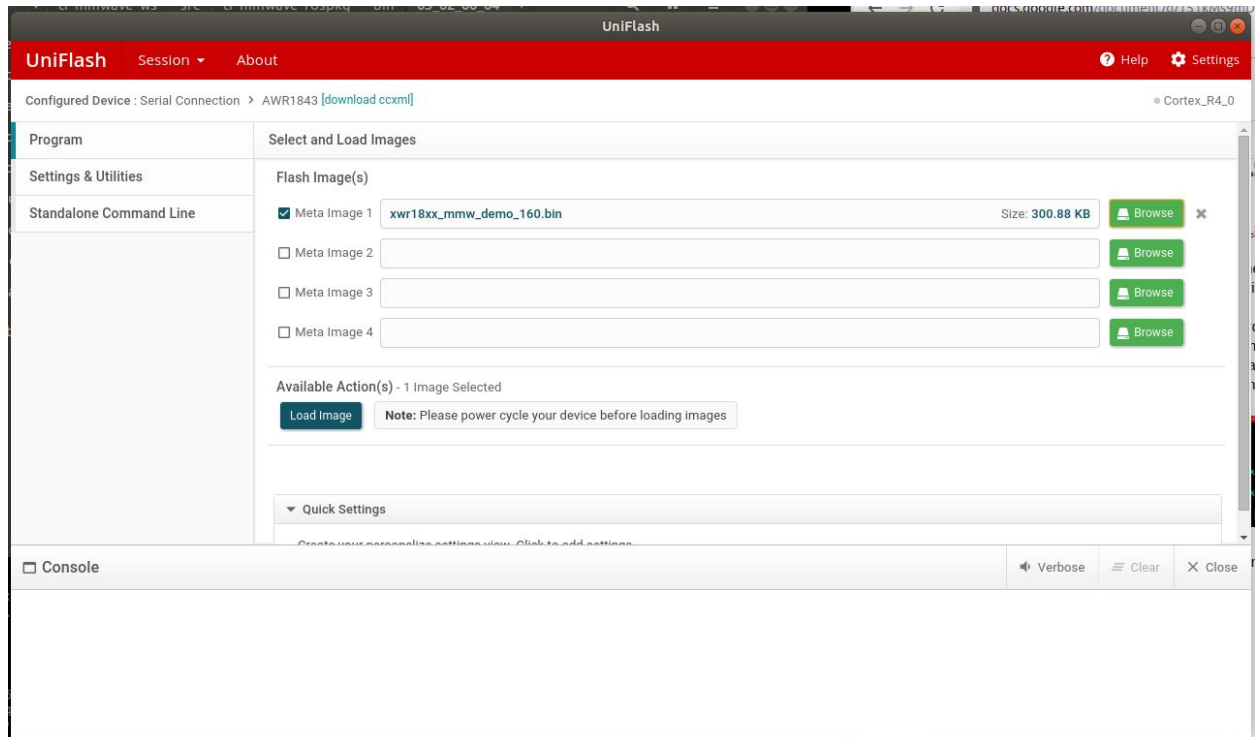
2a. If the board is plugged into the USB port and powered on, disconnect power from the board, and enable the SOP2 DIP switch (lower right hand corner of the board). This puts the board into programming mode.

2b. Reconnect power to the board and start Uniflash. The device should be autodetected, but if not, select AWR1843BOOST from the device menu.

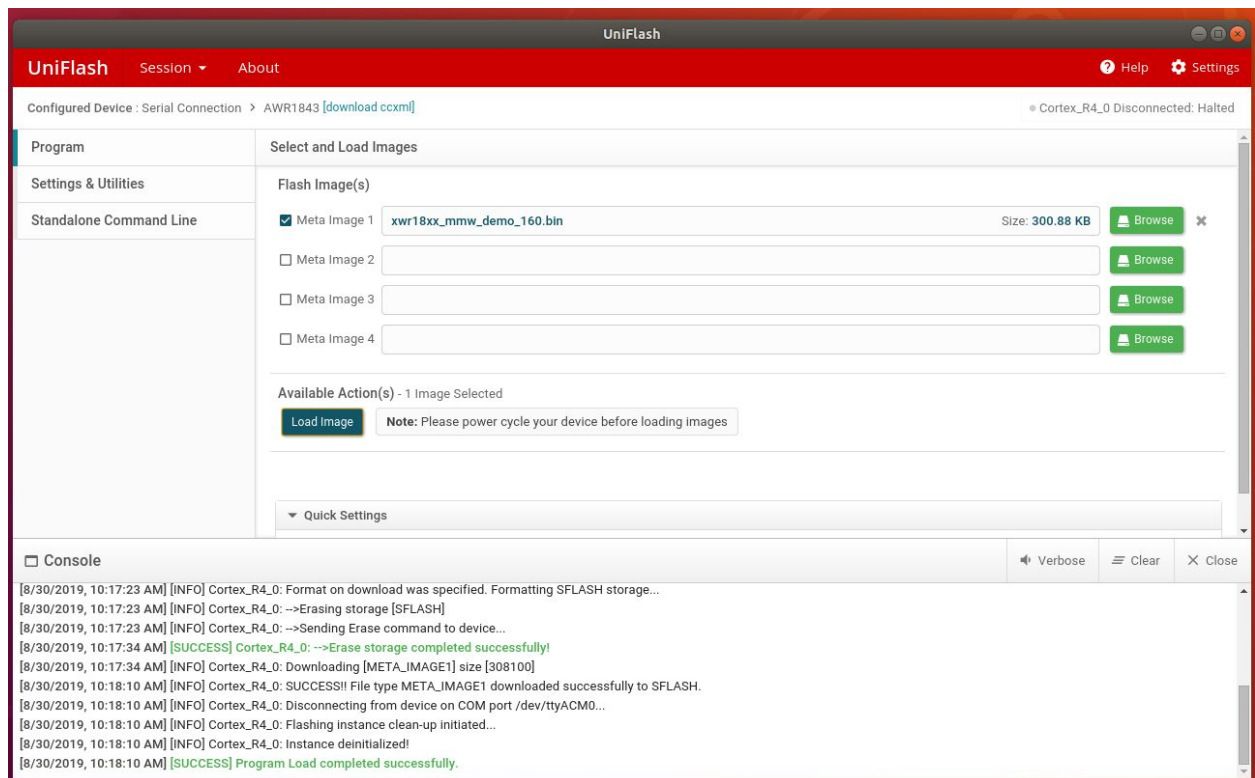
2c. Click on 'Settings & Utilities' and change the 'COM Port' to the port associated with the Command Port (if00). In the screenshot below, I would set the 'COM Port' to `/dev/ttyACM0` (but this can change is other usb devices are plugged in)

```
carl@noumenon: ~/ROS/ti_mmwave_ws/src/goggles 119x29
carl@noumenon:~/ROS/ti_mmwave_ws/src/goggles$ ll /dev/serial/by-id/
total 0
drwxr-xr-x 2 root root 80 Aug 30 09:03 ./
drwxr-xr-x 4 root root 80 Aug 30 09:03 ../
lrwxrwxrwx 1 root root 13 Aug 30 09:03 usb-Texas_Instruments_XDS110__02.03.00.14__Embed_with_CMSIS-DAP_R1091043-if00 ->
../ttyACM0
lrwxrwxrwx 1 root root 13 Aug 30 09:03 usb-Texas_Instruments_XDS110__02.03.00.14__Embed_with_CMSIS-DAP_R1091043-if03 ->
../ttyACM1
carl@noumenon:~/ROS/ti_mmwave_ws/src/goggles$
```

2d. Return to 'Program' screen and load the `xwr18xx_mmw_demo_160.bin` (`ti_mmwave_rospkg/bin/03_02_00_04/xwr18xx_mmw_demo_160.bin`) as 'Meta Image 1' and then click 'Load Image' (Note: This step can fail if the board has not been put into programming mode as described above)



When this has finished properly, you should see the following:



Launching the Goggles Node

1. The Goggles launch file, `goggles.launch`, accepts as inputs the command port and data port associated with the device. The default values for these arguments are the following:

```
<arg name="command_port" default="/dev/ttyACM0"/>
<arg name="data_port" default="/dev/ttyACM1"/>
```

Note: You may need to pass these values in as input arguments if the board is not mapped to `/dev/ttyACM0` and `/dev/ttyACM1`

2. Start the radar board and Goggles velocity estimation node with the following launch file
(Note that the board must NOT be in programming mode for the radar to launch)

```
roslaunch goggles goggles.launch
```

3. Optionally, the input argument `rviz` can be set to `true` to enable an RVIZ visualization of the radar pointcloud

```
roslaunch goggles goggles.launch rviz:=true
```