**ASSIGNMENT 4 - ROS Hardware Integration**

Nathan Ramos & Quinn Frady

**Description of the problem**

The goal of this assignment is to interface the roscore with the STM32CubeIDE using USART1 via the USBC port to transmit and visualize our ultrasonic data. We will use this integration to perform the following tasks.

**Task 1**

The goal of task 1 is to use our program from Assignment 1, Part 1 and USART DMA functionality to integrate the HC-SR04 US sensor to measure distance. We will program this node into the STM32 Core Board via a topic range\_node of type sensor\_msgs/Range names /us\_range to the host machine. We will visualize the data in RVIZ.

Task 1 wanted the following ros nodes and topics:

* Nodes
  + Range\_Node (Declared in STM32F103RCT6)
    - Topics
      * Publishes - /us\_range of type sensor\_msgs::Range

To execute this assignment, we created our package da4\_1 in STMCubeIDE that references ros, std\_msgs, sensor\_msgs, string, stdio, and stdlib. Afterward we combined the necessary components of rosserial with our previous ultrasonic sensor code to create a package on the STM board capable of communicating with ros. It initiates our range\_node that determines ultrasonic data and communicates this data across the topic us\_range. We created this topic with a radiation type of 1 (sound), a header.frame\_id of us\_range, a minimum range of 0, and a maximum range of 150. Our field of view is 0.5 radians which is 30 degrees. Finally, we displayed this data in RVIZ with the steps outlined below.

To execute this task you must run the following steps

1. Upload STM32CubeIDE Code to STM32F103RCT6
2. Terminal 1
   1. Roscore
3. Terminal 2
   1. CD to DesignAssignments/da4
   2. Catkin\_make
   3. Roslaunch da4\_1 begin.launch
4. View data in RVIZ

**Task 1 Source Code Directory**

DesignAssignments/da4/src/da4\_1/Core/Src/main.cc

**Task 1 Video**

<https://youtu.be/wEbIRuEQTj0>

**Task 2**

The goal of Task 2 was to user our program from Assignment 1, Part 2 and USART DMA functionality to integrate the on-board ICM20607 IMU to measure the roll pitch and yaw of our core board. We will program this node into the STM32 Core Board via a topic of type std\_msgs/Float32MultiArray. We will create a ROS Node of type sensor\_msgs/Imu to convert this data and publish it for RVIZ to visualize.

Task 2 wanted the following ros nodes and topics

* Nodes
  + RPY\_Node (Declared in the STMF103RCT6)
    - Topics
      * Publishes - /rpy of type std\_msgs::Float32MultiArray
  + IMU\_Node
    - Topics
      * Subscribes - /rpy
      * Publishes - /imu of type sensor\_msgs::Imu

To execute this assignment, we created our package da4\_2 in STMCubeIDE that references ros, std\_msgs, sensor\_msgs, and geometry\_msgs. Afterward we combined the components of rosserial with our previous IMU code to create a package on the STM Board capable of communicating with ros. It initiates our rpy\_node that determines the IMU orientation and communicates this data across the topic /rpy. We created a topic /imu to read and convert this data to quaternion. Finally, we displayed this data in RVIZ.

To execute this task you must run the following steps

1. Upload STM32CubeIDE Code to STM32F103RCT6
2. Terminal 1
   1. Roscore
3. Terminal 2
   1. CD to DesignAssignments/da4
   2. Catkin\_make
   3. Roslaunch da4\_2 begin.launch
4. View data in RVIZ

**Task 2 Source Code Directory**

DesignAssignments/da4/src/da4\_2/Core/Src/main.cc

And

DesignAssignments/da4/src/dar\_2/src/

**Task 2 Video**

<https://youtu.be/yR2_QYh5zL4>