1. build package
- copy source to the ROS source forder (~/catkin_ws/src/)
- build

```
cd ~/catkin_ws
catkin_make
```

2. launch roscore

```
roscore
```

3. set parameters of serial node “rft_sensor_serial”
- set the port, baudrate, force/torque divider value
- ROS parameters of the node should be set to the same value as the set value of the sensor.
(Refer to the sensor's manual for the setting value.)
- If not set, the following values are set as defaults while the node is running. If the default value and the value to be set are the same, there is no need to set the corresponding parameter.
- Parameters and Default value

| Parameter | Description | Default value |
|---|---|---|
| /RFT_COM_PORT | Name of serial device | /dev/ttyUSB0 |
| /RFT_COM_BAUD | Baud-rate of serial port | 115200 |
| /RFT_FORCE_DEVIDER | Force divider | 50 |
| /RFT_TORQUE_DEVIDER | Torque divider | 2000 |

-For example, to set the serial port to “/dev/ttyUSB1” and the torque divider to 1000, use the ROS command as follows.

```
rosparam set /RFT_COM_PORT /dev/ttyUSB0
rosparam set /RFT_TORQUE_DEVIDER  1000
```

4. start to run serial node “rft_sensor_serial”

```
rosrun rft_sensor_serial rft_sensor_serial
```

5. A service caller is needed to send an action command to the sensor. The following is an example of using “rqt_service_caller”.

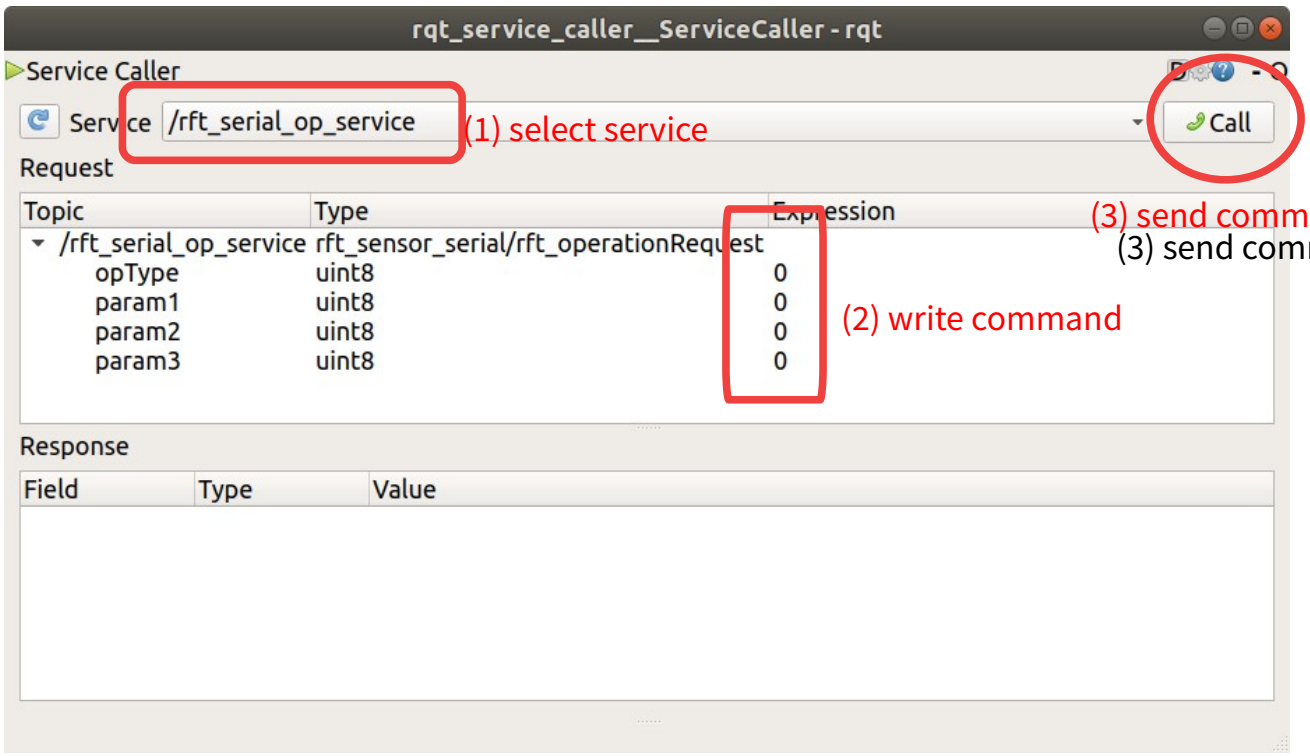
```
rosrun rqt_service_caller rqt_service_caller
```

-After setting the command(opType) and parameter values(param1, param2, param3) by referring to the manual, you can control the operation and setting of the sensor by pressing the call button.

- For example,

| **Command** | **opType** | **param1** | **param2** | **param3** |
|---|---|---|---|---|
| Start F/T data out | 11 | DC | DC | DC |
| Stop F/T data out | 12 | DC | DC | DC |
| Set bias | 17 | 1 | DC | DC |
| Set un-biased | 17 | 0 | DC | DC |

(Note) DC means “Don’t care’ → Any value doesn't matter.




6.you can see the force/torque data using rqt_plot

```
rqt_plot /RFT_FORCE/wrench/force
rqt_plot /RFT_FORCE/wrench/torque
```