ROS 2 Interface Guide for RFT Series (UART Model)

This guide provides instructions for interfacing the **RFT sensor**, which uses **UART communication**, within a **ROS 2** environment.

Steps 2 through 4 require a total of **three terminals**.
Each step must be executed in a **separate terminal**.

System Requirements

- ROS 2 must be installed
- Verified in a virtual environment using **Ubuntu 24.04** with **ROS 2 Jazzy**

1.Serial Port Low Latency Setting

- For stable communication with the sensor, the serial port should be configured with the **low_latency** option.

1-1) Identify the Serial Device

- First, identify the serial device assigned to the sensor.
- Use the following command to list connected serial devices:

```
ls -l /dev/ttyUSB*
```

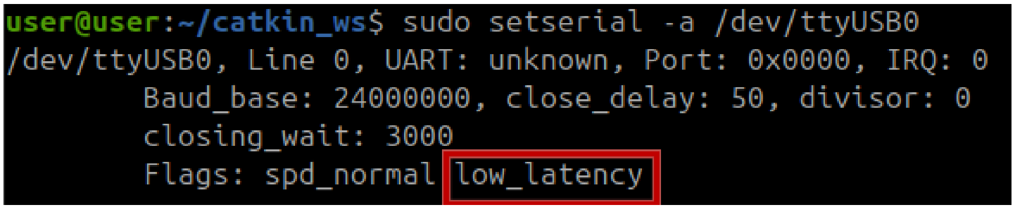
- Determine which device corresponds to the sensor (for example: /dev/ttyUSB0).

1-2) Check Serial Port Status

- Check the current status of the identified serial device(You must use the actual device name instead of /dev/ttyUSB0).

```
sudo setserial -a /dev/ttyUSB0
```

- Check the **Flags** field in the output.
- If **low_latency** is present, no further action is required. Proceed to **Step 2**.
- If **low_latency** is not present, complete **Steps 1-3 through 1-6**, then proceed to **Step 2**



```
user@user:~/catkin_ws$ sudo setserial -a /dev/ttyUSB0
/dev/ttyUSB0, Line 0, UART: unknown, Port: 0x0000, IRQ: 0
        Baud_base: 24000000, close_delay: 50, divisor: 0
        closing_wait: 3000
        Flags: spd_normal low_latency
```

1-3) Install setserial (if not installed)

- If the **setserial** command is not available, install it using the following command:

  sudo apt update

  sudo apt install setserial

- After installation, recheck the port status:

  sudo setserial -a /dev/ttyUSB0

1-4) Enable low_latency

- Apply the **low_latency** option to the serial device(You must use the actual device name instead of /dev/ttyUSB0).

  sudo setserial /dev/ttyUSB0 low_latency

(Replace /dev/ttyUSB0 with the actual device name.)

1-5) Verify Configuration

- Verify that the configuration has been applied:

  sudo setserial -a /dev/ttyUSB0

- Confirm that **low_latency** appears in the **Flags** field.

1-6) Notes

- Enabling low_latency improves serial communication responsiveness.
- Device names such as ttyUSB* or ttyACM* may vary by system.
- The configuration may be reset after a system reboot.
- Reapply these steps if communication issues occur again.

2. ROS 2 Workspace Setup

2-1) Create Your Workspace

- Create a directory that will contain your ROS 2 workspace.

```
mkdir -p ~/{your_workspace}
cd ~/{your_workspace}
```

2-2) Create the ‘src’ Folder

- The `src` folder is not generated automatically. You need to create it manually to store ROS 2 packages.

```
mkdir src
```

2-3) Unzip the ROS 2 Package into the ‘src’ Folder

- Copy ‘rft_sensor_serial.zip’ into `src` and extract it.

```
cp ~/downloads/rft_sensor_serial.zip ~/{your_workspace}/src/
cd ~/{your_workspace}/src
unzip rft_sensor_serial.zip
```

2-4) Build the Workspace

- Build all packages in the 'src' folder using 'colcon'.

```
cd ~/{your_workspace}
colcon build
```

- After the build completes, verify that the following directories are created:
  - build/
  - install/
  - log/

2-5) Source the ROS 2 Environment

- Source the ROS 2 setup script to load the core environment, including required paths and command-line tools.

```
source /opt/ros/setup.bash
```

2-6) Source the Workspace

- To use the newly built packages, source the setup script.

```
source ./install/setup.bash
```

2-7) Launches the `rft_sensor_serial` package

- Launch the sensor node and begin publishing RFT sensor data to ROS 2 topics.

```
ros2 launch rft_sensor_serial rft_sensor_launch.py
```

2-8) Notes

- To change the sensor USB port, edit the YAML file located in the config directory of the **rft_sensor_serial** package.
- Update the port value to match the device path where the sensor is detected.

3. Check and Display FT Sensor Data

3-1) Open a new terminal and run the following three commands:

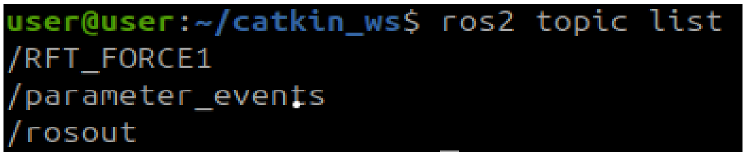
```
cd ~/{your_workspace}
```

```
source /opt/ros/setup.bash
```

```
source ./install/setup.bash
```

3-2) List Available ROS 2 Topics

- List all topics currently published by ROS 2 nodes:



```
user@user:~/catkin_ws$ ros2 topic list
/RFT_FORCE1
/parameter_events
/rosout
```

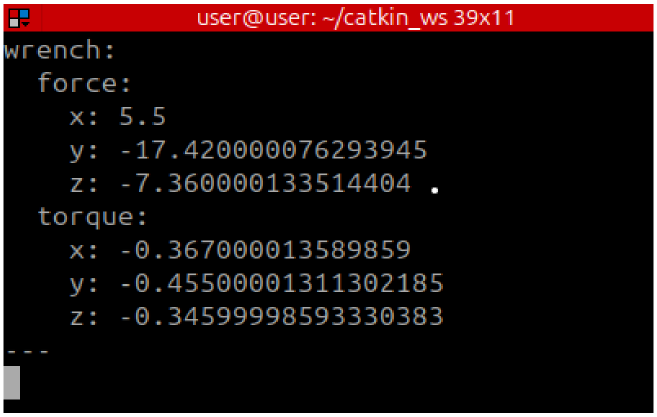
```
ros2 topic list
```

- This command displays all topics currently published by ROS 2 nodes.
- Identify the FT sensor topic (e.g., ‘/RFT_FORCE1’).

3-3) Display FT Data

- Display the FT (Force/Torque) data published by the sensor:

```
ros2 topic echo /RFT_FORCE1
```



user@user: ~/catkin_ws 39x11

```
wrench:
  force:
    x: 5.5
    y: -17.420000076293945
    z: -7.360000133514404
  torque:
    x: -0.367000013589859
    y: -0.45500001311302185
    z: -0.34599998593330383
---
```

- This command continuously outputs FT sensor data to the terminal.
- Only FT data messages are displayed.

4. Sensor Control via ROS 2 Services

- Sensor control commands are sent using **ROS 2 services**.
- A service client is required to send commands to the sensor.

4-1) Launch the Service Client

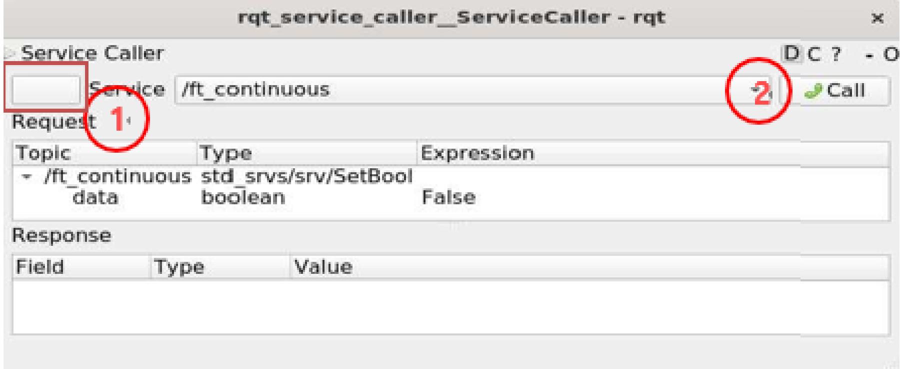
- Open a new terminal and source the environment:

```
cd ~/{your_workspace}

source /opt/ros/setup.bash

source ./install/setup.bash
```

- Launch the service client using rqt_service_caller:

```
ros2 run rqt_service_caller rqt_service_caller
```

- Once the window opens, click the Update button (1), then select the desired service from the list (2).



Service /ft_continuous

Request

Topic	Type	Expression
/ft_continuous	std_srvs/srv/SetBool	
data	boolean	False

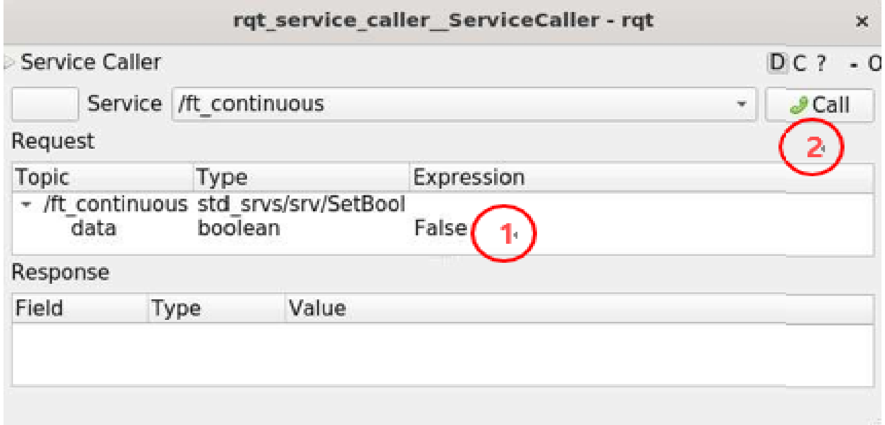
Response

Field	Type	Value

- Before using any service other than FT output, /ft_continuous must be set to 'False'.
- Other functions will not operate correctly while FT data output is active.
- Output messages from read/set services appear in the terminal used in **Section 2-7**.
- Start/Stop FT output messages appear in the terminal used in **Section 3-3**.
- Parameters returned by read services follow the tables in Sections **4-10 to 4-13**.

4-2) Stop / Start Data Output

- To stop FT data output:
    - Set the **expression** parameter of /ft_continuous to 'False' (1)
    - Click **Call** (2)
- To start FT data output:
    - Set the parameter to 'True' (1)
    - Click **Call** (2)



Service /ft_continuous

Request

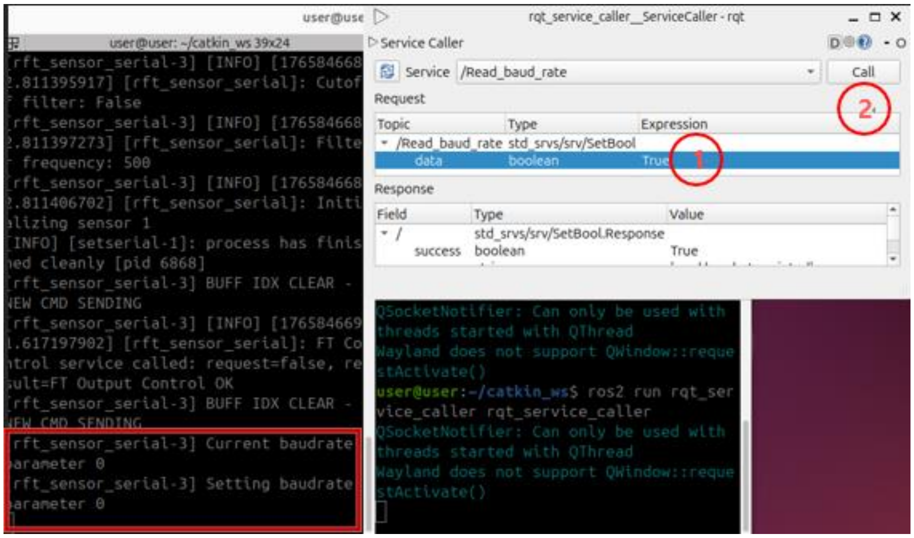
Topic	Type	Expression
/ft_continuous	std_srvs/srv/SetBool	
data	boolean	False

Response

Field	Type	Value

4-3) Read Baud-rate

- Select the /Read_baud_rate service.
- Set the **expression** parameter to 'True' (1).
- Click **Call** (2).


user@use
rqt_service_caller__ServiceCaller - rqt
user@user: ~/catkin_ws 39x24
Service Caller
Call
2
1

```
[rft_sensor_serial-3] [INFO] [176584668
2.811395917] [rft_sensor_serial]: Cutof
f filter: False
[rft_sensor_serial-3] [INFO] [176584668
2.811397273] [rft_sensor_serial]: Filte
r frequency: 500
[rft_sensor_serial-3] [INFO] [176584668
2.811406702] [rft_sensor_serial]: Initi
alizing sensor 1
[INFO] [setserial-1]: process has finis
hed cleanly [pid 6868]
[rft_sensor_serial-3] BUFF IDX CLEAR -
NEW CMD SENDING
[rft_sensor_serial-3] [INFO] [176584669
1.617197902] [rft_sensor_serial]: FT Co
ntrol service called: request=false, re
sult=FT Output Control OK
[rft_sensor_serial-3] BUFF IDX CLEAR -
NEW CMD SENDING
[rft_sensor_serial-3] Current baudrate
parameter 0
[rft_sensor_serial-3] Setting baudrate
parameter 0
```

Service /Read_baud_rate

Request

Topic	Type	Expression
/Read_baud_rate	std_srvs/srv/SetBool	
data	boolean	True

Response

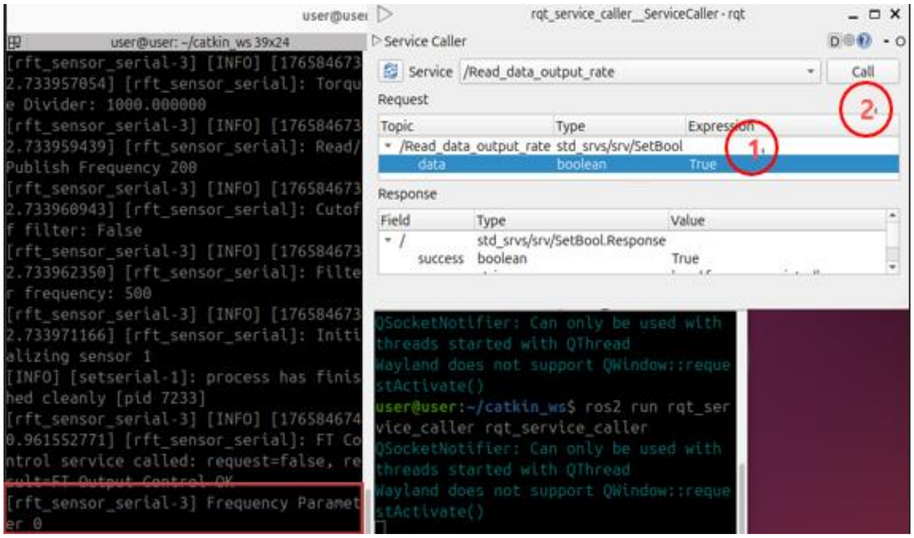
Field	Type	Value
/	std_srvs/srv/SetBool.Response	
success	boolean	True

```
QSocketNotifier: Can only be used with
threads started with QThread
Wayland does not support QWindow::reque
stActivate()
user@user:~/catkin_ws$ ros2 run rqt_ser
vice_caller rqt_service_caller
QSocketNotifier: Can only be used with
threads started with QThread
Wayland does not support QWindow::reque
stActivate()
```

- The current baud rate and configured baud rate are displayed.

4-4) Read Data Output Rate

- Select the /Read_data_output_rate service.
- Set the **expression** parameter to ‘True’(1).
- Click **Call** (2).



```
user@user: ~/catkin_ws 39x24
[rft_sensor_serial-3] [INFO] [176584673
2.733957054] [rft_sensor_serial]: Torqu
e Divider: 1000.000000
[rft_sensor_serial-3] [INFO] [176584673
2.733959439] [rft_sensor_serial]: Read/
Publish Frequency 200
[rft_sensor_serial-3] [INFO] [176584673
2.733960943] [rft_sensor_serial]: Cutof
f filter: False
[rft_sensor_serial-3] [INFO] [176584673
2.733962350] [rft_sensor_serial]: Filte
r frequency: 500
[rft_sensor_serial-3] [INFO] [176584673
2.733971166] [rft_sensor_serial]: Initi
alizing sensor 1
[INFO] [setserial-1]: process has finis
hed cleanly [pid 7233]
[rft_sensor_serial-3] [INFO] [176584674
0.961552771] [rft_sensor_serial]: FT Co
ntrol service called: request=false, re
sult=FT Output Control OK
[rft_sensor_serial-3] Frequency Paramet
er 0
```

rqt_service_caller__ServiceCaller - rqt

Service Caller

Service: /Read_data_output_rate

Request

Topic	Type	Expression
/Read_data_output_rate	std_srvs/srv/SetBool	
data	boolean	True

Response

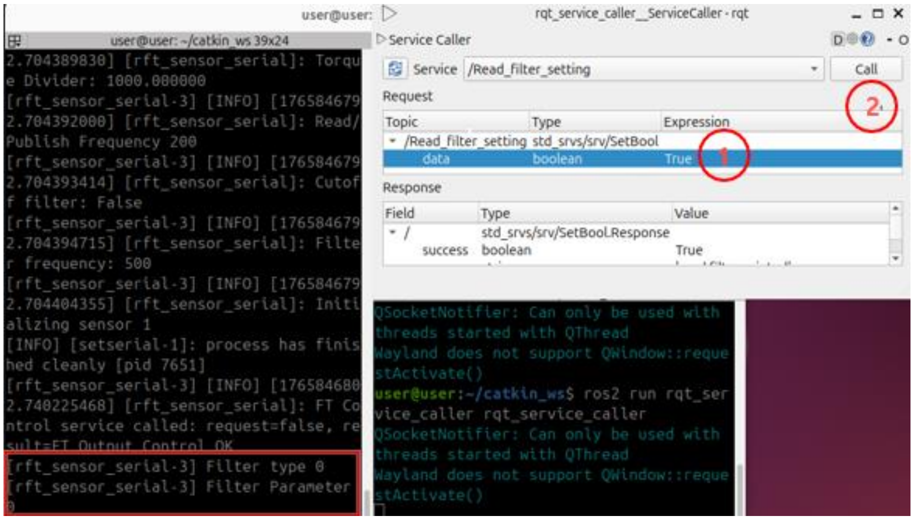
Field	Type	Value
/	std_srvs/srv/SetBool.Response	
success	boolean	True

```
QSocketNotifier: Can only be used with
threads started with QThread
Wayland does not support QWindow::reque
stActivate()
user@user:~/catkin_ws$ ros2 run rqt_ser
vice_caller rqt_service_caller
QSocketNotifier: Can only be used with
threads started with QThread
Wayland does not support QWindow::reque
stActivate()
```

- The current data output rate parameter is displayed.

4-5) Read Filter Setting

- Select the /Read_filter_setting service.
- Set the **expression** parameter to ‘True’(1).
- Click **Call** (2).



```
user@user: ~/catkin_ws 39x24
2.704389830] [rft_sensor_serial]: Torqu
e Divider: 1000.000000
[rft_sensor_serial-3] [INFO] [176584679
2.704392000] [rft_sensor_serial]: Read/
Publish Frequency 200
[rft_sensor_serial-3] [INFO] [176584679
2.704393414] [rft_sensor_serial]: Cutof
f filter: False
[rft_sensor_serial-3] [INFO] [176584679
2.704394715] [rft_sensor_serial]: Filte
r frequency: 500
[rft_sensor_serial-3] [INFO] [176584679
2.704404355] [rft_sensor_serial]: Initi
alizing sensor 1
[INFO] [setserial-1]: process has finis
hed cleanly [pid 7651]
[rft_sensor_serial-3] [INFO] [176584680
2.740225468] [rft_sensor_serial]: FT Co
ntrol service called: request=false, re
sult=FT Output Control OK
[rft_sensor_serial-3] Filter type 0
[rft_sensor_serial-3] Filter Parameter
0
```

rqt_service_caller__ServiceCaller - rqt

Service Caller

Service: /Read_filter_setting

Request

Topic	Type	Expression
/Read_filter_setting	std_srvs/srv/SetBool	
data	boolean	True

Response

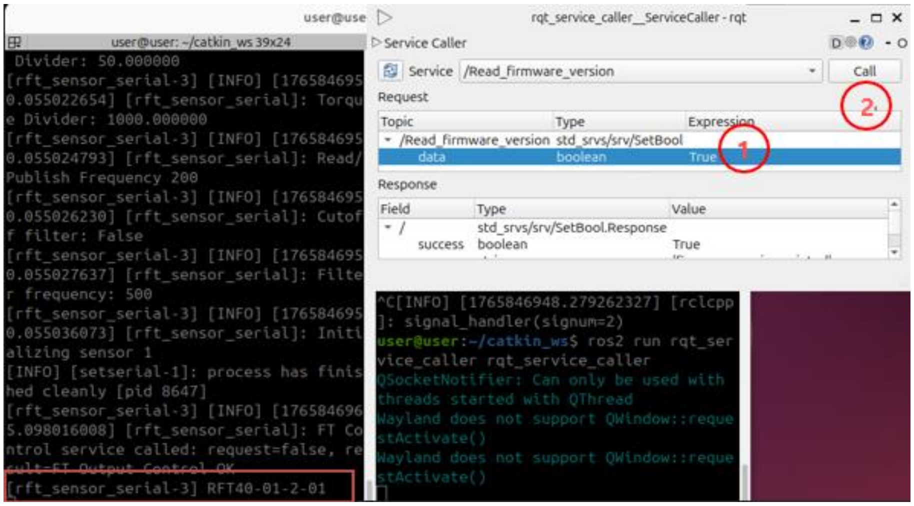
Field	Type	Value
/	std_srvs/srv/SetBool.Response	
success	boolean	True

```
QSocketNotifier: Can only be used with
threads started with QThread
Wayland does not support QWindow::reque
stActivate()
user@user:~/catkin_ws$ ros2 run rqt_ser
vice_caller rqt_service_caller
QSocketNotifier: Can only be used with
threads started with QThread
Wayland does not support QWindow::reque
stActivate()
```

- The current filter setting parameter is displayed.

4-6) Read Firmware Version

- Select the /Read_firmware_version service.
- Set the **expression** parameter to ‘True’ (1).
- Click **Call** (2).


user@use
user@user: ~/catkin_ws 39x24
rqt_service_caller__ServiceCaller - rqt
Service Caller
Call
2
1

```
 Divider: 50.000000
[rft_sensor_serial-3] [INFO] [176584695
0.055022654] [rft_sensor_serial]: Torqu
e Divider: 1000.000000
[rft_sensor_serial-3] [INFO] [176584695
0.055024793] [rft_sensor_serial]: Read/
Publish Frequency 200
[rft_sensor_serial-3] [INFO] [176584695
0.055026230] [rft_sensor_serial]: Cutof
f filter: False
[rft_sensor_serial-3] [INFO] [176584695
0.055027637] [rft_sensor_serial]: Filte
r frequency: 500
[rft_sensor_serial-3] [INFO] [176584695
0.055036073] [rft_sensor_serial]: Initi
alizing sensor 1
[INFO] [setserial-1]: process has finis
hed cleanly [pid 8647]
[rft_sensor_serial-3] [INFO] [176584696
5.098016008] [rft_sensor_serial]: FT Co
ntrol service called: request=false, re
sult=FT Output Control OK
[rft_sensor_serial-3] RFT40-01-2-01
```

Service /Read_firmware_version

Request

Topic	Type	Expression
/Read_firmware_version	std_srvs/srv/SetBool	
data	boolean	True

Response

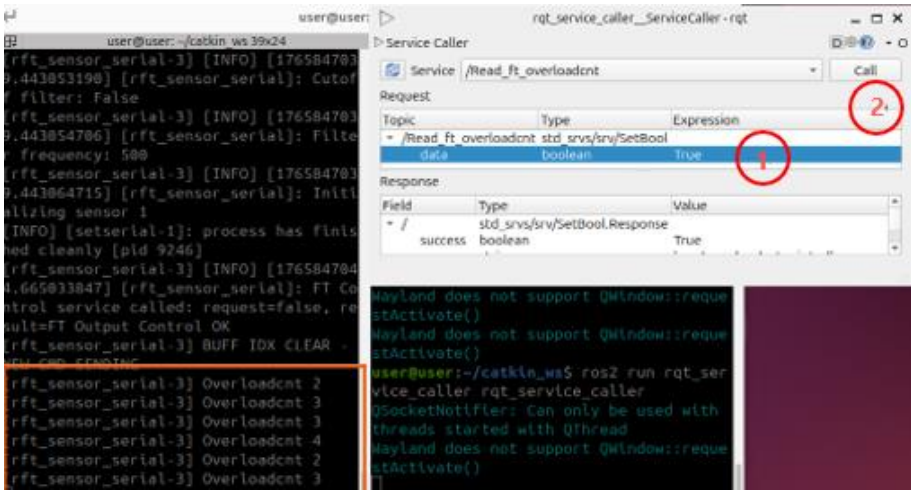
Field	Type	Value
/	std_srvs/srv/SetBool.Response	
success	boolean	True

```
^C[INFO] [1765846948.279262327] [rclcpp
]: signal_handler(signum=2)
user@user:~/catkin_ws$ ros2 run rqt_ser
vice_caller rqt_service_caller
QSocketNotifier: Can only be used with
threads started with QThread
Wayland does not support QWindow::reque
stActivate()
Wayland does not support QWindow::reque
stActivate()
```

- The firmware version is displayed.

4-7) Read Count of Overload Occurrence

- Select the /Read_ft_overloadcnt service.
- Set the **expression** parameter to ‘True’ (1).
- Click **Call** (2).


user@user:
user@user: ~/catkin_ws 39x24
rqt_service_caller__ServiceCaller - rqt
Service Caller
Call
2
1

```
[rft_sensor_serial-3] [INFO] [176584703
9.443853190] [rft_sensor_serial]: Cutof
f filter: False
[rft_sensor_serial-3] [INFO] [176584703
9.443854706] [rft_sensor_serial]: Filte
r frequency: 500
[rft_sensor_serial-3] [INFO] [176584703
9.443064715] [rft_sensor_serial]: Initi
alizing sensor 1
[INFO] [setserial-1]: process has finis
hed cleanly [pid 9246]
[rft_sensor_serial-3] [INFO] [176584704
4.665033847] [rft_sensor_serial]: FT Co
ntrol service called: request=false, re
sult=FT Output Control OK
[rft_sensor_serial-3] BUFF IDX CLEAR -
NEW CMD SENDING
[rft_sensor_serial-3] Overloadcnt 2
[rft_sensor_serial-3] Overloadcnt 3
[rft_sensor_serial-3] Overloadcnt 3
[rft_sensor_serial-3] Overloadcnt 4
[rft_sensor_serial-3] Overloadcnt 2
[rft_sensor_serial-3] Overloadcnt 3
```

Service /Read_ft_overloadcnt

Request

Topic	Type	Expression
/Read_ft_overloadcnt	std_srvs/srv/SetBool	
data	boolean	True

Response

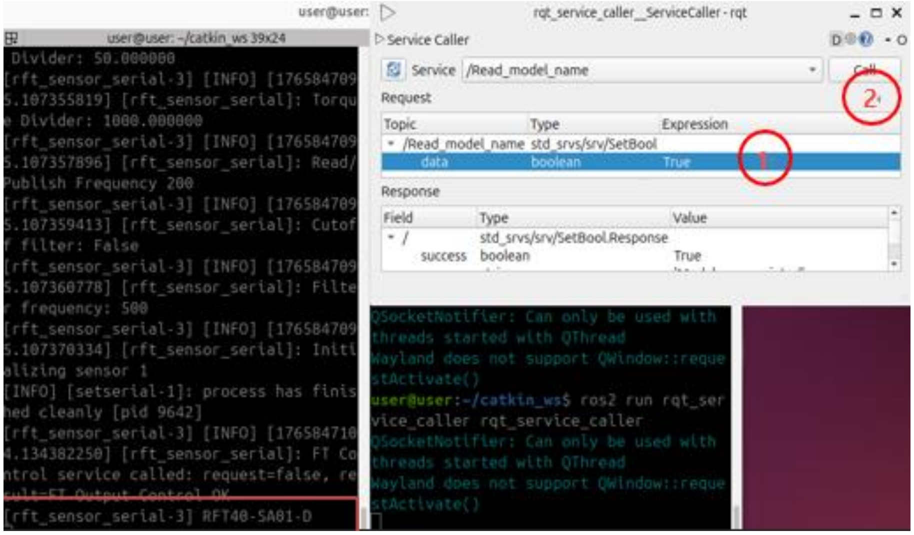
Field	Type	Value
/	std_srvs/srv/SetBool.Response	
success	boolean	True

```
Wayland does not support QWindow::reque
stActivate()
Wayland does not support QWindow::reque
stActivate()
user@user:~/catkin_ws$ ros2 run rqt_ser
vice_caller rqt_service_caller
QSocketNotifier: Can only be used with
threads started with QThread
Wayland does not support QWindow::reque
stActivate()
```

- Overload count values are displayed in the following order:
  - Fx, Fy, Fz, Tx, Ty, Tz

4-8) Read Model Name

- Select the /Read_model_name service.
- Set the **expression** parameter to ‘True’ (1).
- Click **Call** (2).


user@user:
user@user: ~/catkin_ws 39x24
rqt_service_caller__ServiceCaller - rqt
Service Caller
Call
2
1

```
Divider: 50.000000
[rft_sensor_serial-3] [INFO] [176584709
5.107355819] [rft_sensor_serial]: Torqu
e Divider: 1000.000000
[rft_sensor_serial-3] [INFO] [176584709
5.107357896] [rft_sensor_serial]: Read/
Publish Frequency 200
[rft_sensor_serial-3] [INFO] [176584709
5.107359413] [rft_sensor_serial]: Cutof
f filter: False
[rft_sensor_serial-3] [INFO] [176584709
5.107360778] [rft_sensor_serial]: Filte
r frequency: 500
[rft_sensor_serial-3] [INFO] [176584709
5.107370334] [rft_sensor_serial]: Initi
alizing sensor 1
[INFO] [setserial-1]: process has finis
hed cleanly [pid 9642]
[rft_sensor_serial-3] [INFO] [176584710
4.134382250] [rft_sensor_serial]: FT Co
ntrol service called: request=false, re
sult=FT Output Control OK
[rft_sensor_serial-3] RFT40-SA01-D
```

Service: /Read_model_name

Request

Topic	Type	Expression
/Read_model_name	std_srvs/srv/SetBool	
data	boolean	True

Response

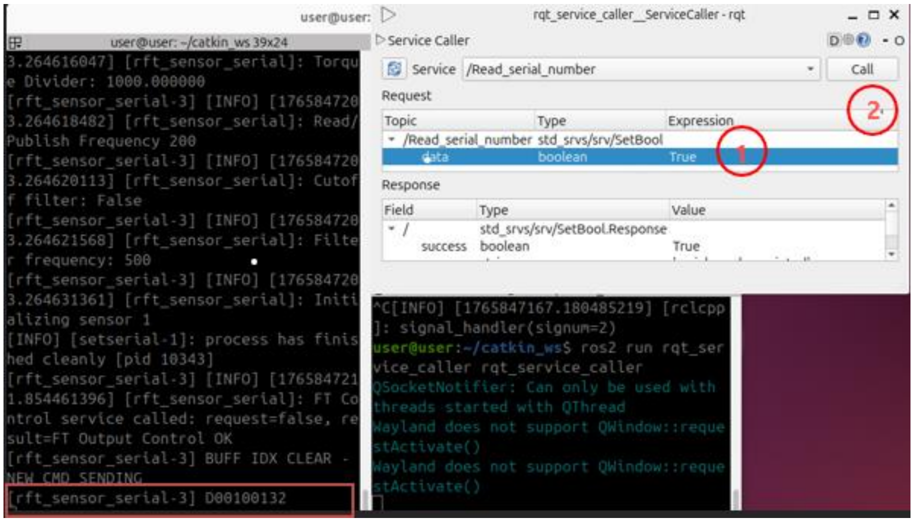
Field	Type	Value
/	std_srvs/srv/SetBool.Response	
success	boolean	True

```
QSocketNotifier: Can only be used with
threads started with QThread
Wayland does not support QWindow::reque
stActivate()
user@user:~/catkin_ws$ ros2 run rqt_ser
vice_caller rqt_service_caller
QSocketNotifier: Can only be used with
threads started with QThread
Wayland does not support QWindow::reque
stActivate()
```

- The sensor model name is displayed.

4-9) Read Serial Number

- Select the /Read_serial_number service.
- Set the **expression** parameter to ‘True’ (1).
- Click **Call** (2).


user@user:
user@user: ~/catkin_ws 39x24
rqt_service_caller__ServiceCaller - rqt
Service Caller
Call
2
1

```
3.264616047] [rft_sensor_serial]: Torqu
e Divider: 1000.000000
[rft_sensor_serial-3] [INFO] [176584720
3.264618482] [rft_sensor_serial]: Read/
Publish Frequency 200
[rft_sensor_serial-3] [INFO] [176584720
3.264620113] [rft_sensor_serial]: Cutof
f filter: False
[rft_sensor_serial-3] [INFO] [176584720
3.264621568] [rft_sensor_serial]: Filte
r frequency: 500
[rft_sensor_serial-3] [INFO] [176584720
3.264631361] [rft_sensor_serial]: Initi
alizing sensor 1
[INFO] [setserial-1]: process has finis
hed cleanly [pid 10343]
[rft_sensor_serial-3] [INFO] [176584721
1.854461396] [rft_sensor_serial]: FT Co
ntrol service called: request=false, re
sult=FT Output Control OK
[rft_sensor_serial-3] BUFF IDX CLEAR -
NEW CMD SENDING
[rft_sensor_serial-3] D00100132
```

Service: /Read_serial_number

Request

Topic	Type	Expression
/Read_serial_number	std_srvs/srv/SetBool	
data	boolean	True

Response

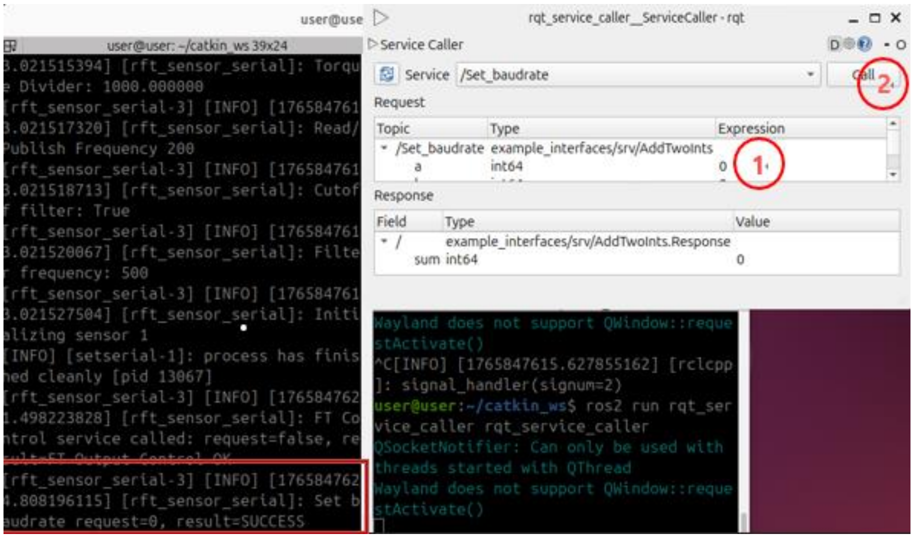
Field	Type	Value
/	std_srvs/srv/SetBool.Response	
success	boolean	True

```
^C[INFO] [1765847167.180485219] [rclcpp
]: signal_handler(signum=2)
user@user:~/catkin_ws$ ros2 run rqt_ser
vice_caller rqt_service_caller
QSocketNotifier: Can only be used with
threads started with QThread
Wayland does not support QWindow::reque
stActivate()
Wayland does not support QWindow::reque
stActivate()
```

- The sensor serial number is displayed.

4-10) Set Baud-rate

- Select the /Set_baudrate service.
- Set the desired baud rate value in ‘request.a’ (1).
- Click **Call** (2).



```
user@user: ~/catkin_ws 39x24
3.021515394] [rft_sensor_serial]: Torqu
e Divider: 1000.000000
[rft_sensor_serial-3] [INFO] [176584761
3.021517320] [rft_sensor_serial]: Read/
Publish Frequency 200
[rft_sensor_serial-3] [INFO] [176584761
3.021518713] [rft_sensor_serial]: Cutof
f filter: True
[rft_sensor_serial-3] [INFO] [176584761
3.021520067] [rft_sensor_serial]: Filte
r frequency: 500
[rft_sensor_serial-3] [INFO] [176584761
3.021527504] [rft_sensor_serial]: Initi
alizing sensor 1
[INFO] [setserial-1]: process has finis
hed cleanly [pid 13067]
[rft_sensor_serial-3] [INFO] [176584762
1.498223828] [rft_sensor_serial]: FT Co
ntrol service called: request=false, re
sult=FT Output Control OK
[rft_sensor_serial-3] [INFO] [176584762
4.808196115] [rft_sensor_serial]: Set b
audrate request=0, result=SUCCESS
```

Service /Set_baudrate

Request

Topic	Type	Expression
/Set_baudrate	example_interfaces/srv/AddTwoInts	
a	int64	0

Response

Field	Type	Value
/	example_interfaces/srv/AddTwoInts.Response	
sum	int64	0

```
Wayland does not support QWindow::reque
stActivate()
^C[INFO] [1765847615.627855162] [rclcpp
]: signal_handler(signum=2)
user@user:~/catkin_ws$ ros2 run rqt_ser
vice_caller rqt_service_caller
QSocketNotifier: Can only be used with
threads started with QThread
Wayland does not support QWindow::reque
stActivate()
```

- The result indicates whether the baud rate change was successful.

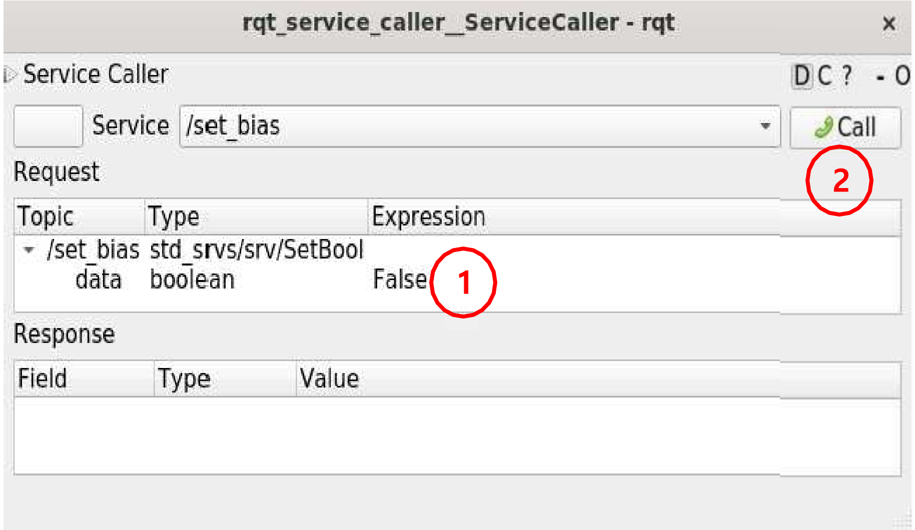
- Refer to the table below for supported baud rates:

Baud-rate parameter	Baud-rate (bps)
0(0x00)	115,200
1(0x01)	921,600
2(0x02)	460,800
3(0x03)	230,400
4(0x04)	115,200

4-11) Set Bias

- To disable bias:
  - Set /set_bias to ‘False’ (1).
  - Click **Call** (2).
- To enable bias:
  - Set the parameter to ‘True’ (1).

- Click **Call** (2).



Service /set_bias

Request

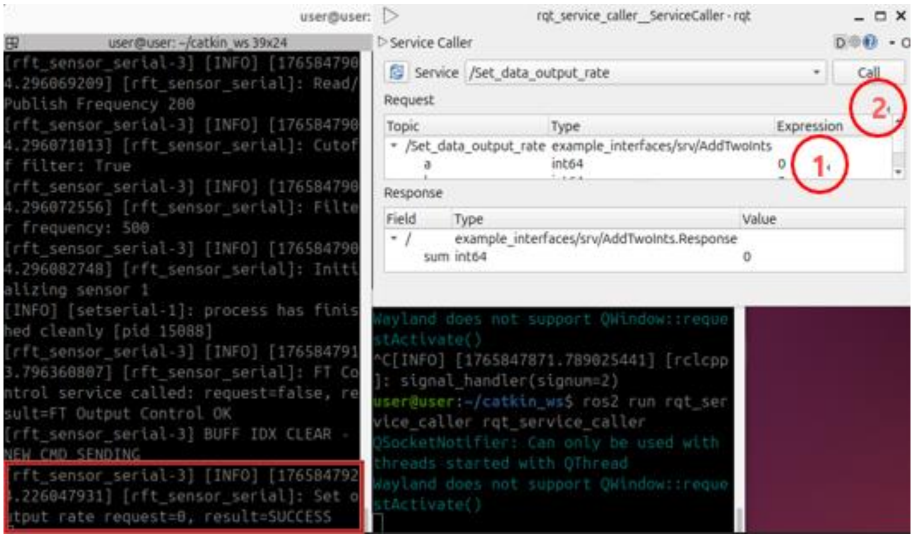
Topic	Type	Expression
/set_bias	std_srvs/srv/SetBool	
data	boolean	False

Response

Field	Type	Value

4-12) Set Data Output Rate

- Select the /Set_data_output_rate service.
- Set the desired output rate value in 'request.a' (1).
- Click **Call** (2).


user@user:
user@user: ~/catkin_ws 39x24
rqt_service_caller__ServiceCaller - rqt
Service Caller
Call
2
1

```
[rft_sensor_serial-3] [INFO] [176584790
4.296069209] [rft_sensor_serial]: Read/
Publish Frequency 200
[rft_sensor_serial-3] [INFO] [176584790
4.296071013] [rft_sensor_serial]: Cutof
f filter: True
[rft_sensor_serial-3] [INFO] [176584790
4.296072556] [rft_sensor_serial]: Filte
r frequency: 500
[rft_sensor_serial-3] [INFO] [176584790
4.296082748] [rft_sensor_serial]: Initi
alizing sensor 1
[INFO] [setserial-1]: process has finis
hed cleanly [pid 15088]
[rft_sensor_serial-3] [INFO] [176584791
3.796360807] [rft_sensor_serial]: FT Co
ntrol service called: request=false, re
sult=FT Output Control OK
[rft_sensor_serial-3] BUFF IDX CLEAR -
NEW_CMD_SENDING
rft_sensor_serial-3] [INFO] [176584792
.226047931] [rft_sensor_serial]: Set o
tput rate request=0, result=SUCCESS
```

Service /Set_data_output_rate

Request

Topic	Type	Expression
/Set_data_output_rate	example_interfaces/srv/AddTwoInts	
a	int64	0

Response

Field	Type	Value
/	example_interfaces/srv/AddTwoInts.Response	
sum	int64	0

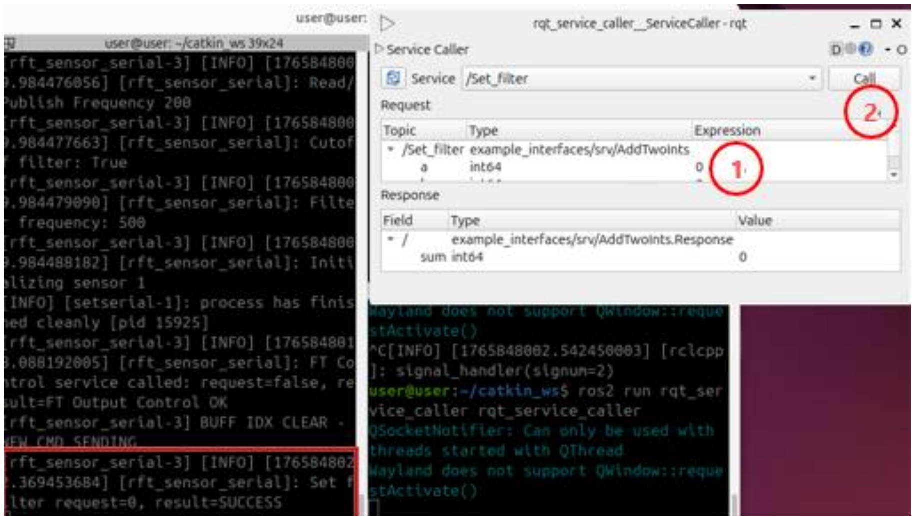
```
Wayland does not support QWindow::reque
stActivate()
^C[INFO] [1765847871.789025441] [rclcpp
]: signal_handler(signum=2)
user@user:~/catkin_ws$ ros2 run rqt_ser
vice_caller rqt_service_caller
QSocketNotifier: Can only be used with
threads started with QThread
Wayland does not support QWindow::reque
stActivate()
```

- Refer to the table below for available output rate parameters:

Communication		Output Rate Parameter (Output rate, Hz)								
Baud-rate Parameter		0 (200)	1 (10)	2 (20)	3 (50)	4 (100)	5 (200)	6 (333)	7 (500)	8 (1000)
0	115,200bps	O	O	O	O	O	O	O	X	X
1	921,600bps	O	O	O	O	O	O	O	O	O
2	460,800bps	O	O	O	O	O	O	O	O	X
3	230,400bps	O	O	O	O	O	O	O	O	X
4	115,200bps	O	O	O	O	O	O	O	X	X
5	57,600bps	O	O	O	O	O	O	X	X	X

4-13) Set Filter

- Select the /Set_filter service.
- Enter the filter parameter value (not the filter type) in request.a (1).
- Click **Call** (2).



```
user@user: ~/catkin_ws 39x24
[rft_sensor_serial-3] [INFO] [176584800
9.984476056] [rft_sensor_serial]: Read/
Publish Frequency 200
[rft_sensor_serial-3] [INFO] [176584800
9.984477663] [rft_sensor_serial]: Cutof
f filter: True
[rft_sensor_serial-3] [INFO] [176584800
9.984479090] [rft_sensor_serial]: Filte
r frequency: 500
[rft_sensor_serial-3] [INFO] [176584800
9.984488182] [rft_sensor_serial]: Initi
alizing sensor 1
[INFO] [setserial-1]: process has finis
hed cleanly [pid 15925]
[rft_sensor_serial-3] [INFO] [176584801
3.088192005] [rft_sensor_serial]: FT Co
ntrol service called: request=false, re
sult=FT Output Control OK
[rft_sensor_serial-3] BUFF IDX CLEAR -
NEW CMD SENDING
[rft_sensor_serial-3] [INFO] [176584802
2.369453684] [rft_sensor_serial]: Set f
ilter request=0, result=SUCCESS
```

rqt_service_caller__ServiceCaller - rqt

Service Caller

Service /Set_filter

Request

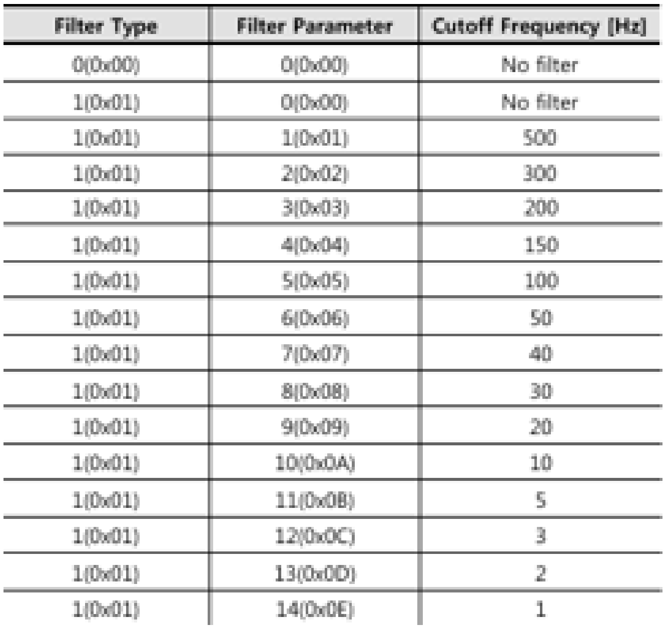
Topic	Type	Expression
/Set_filter	example_interfaces/srv/AddTwoInts	
a	int64	0

Response

Field	Type	Value
/	example_interfaces/srv/AddTwoInts.Response	
sum	int64	0

```
Wayland does not support QWindow::reque
stActivate()
^C[INFO] [1765848002.542450003] [rclcpp
]: signal_handler(signum=2)
user@user:~/catkin_ws$ ros2 run rqt_ser
vice_caller rqt_service_caller
QSocketNotifier: Can only be used with
threads started with QThread
Wayland does not support QWindow::reque
stActivate()
```

- Refer to the table below for filter parameters and cutoff frequencies:



Filter Type	Filter Parameter	Cutoff Frequency [Hz]
0(0x00)	0(0x00)	No filter
1(0x01)	0(0x00)	No filter
1(0x01)	1(0x01)	500
1(0x01)	2(0x02)	300
1(0x01)	3(0x03)	200
1(0x01)	4(0x04)	150
1(0x01)	5(0x05)	100
1(0x01)	6(0x06)	50
1(0x01)	7(0x07)	40
1(0x01)	8(0x08)	30
1(0x01)	9(0x09)	20
1(0x01)	10(0x0A)	10
1(0x01)	11(0x0B)	5
1(0x01)	12(0x0C)	3
1(0x01)	13(0x0D)	2
1(0x01)	14(0x0E)	1

5. Configuration Guide for ‘rft_sensor_serial’ ROS 2 Package

- The ‘rft_sensor_serial’ package uses a YAML configuration file to define UART and sensor parameters.

5-1) Configuration Parameters

```
rft_sensor_serial:
        ros__parameters:
                DEV_NAMES: [“/dev/ttyUSB0”]
                BAUD: 115200
                FORCE_DIVIDER: 50.0
                TORQUE_DIVIDER: 1000.0
                FREQUENCY: 200
                SET_CUTOFF_FILTER: True
                FILTER_FREQUENCY: 500
```

5-2) Parameter Description

- **DEV_NAMES**
  Serial device path connected to the RFT sensor.
- **BAUD**
  UART communication speed (bps).
- **FORCE_DIVIDER**
  Scaling factor for force values (N).
- **TORQUE_DIVIDER**
  Scaling factor for torque values (Nm).
- **FREQUENCY**
  Sensor data publish rate (Hz).
- **SET_CUTOFF_FILTER**
  Enables or disables the low-pass filter.
- **FILTER_FREQUENCY**
  Cutoff frequency of the low-pass filter (Hz).

5-3) Notes

- Ensure **DEV_NAMES** matches the actual device path.
- Adjust divider values according to the sensor model.
- **FILTER_FREQUENCY** is effective only when **SET_CUTOFF_FILTER** is True.