TOPPAN ToF
senSPure™ SDK
ROS 2 Reference Manual

TOPPAN 3D ToF Camera

TOPPAN Holdings Inc.

Revision 1.13

September 9, 2025

Contents

1. Overview

1-1. Purpose of this manual

This manual explains how to build an environment of *senSPure*™ SDK ROS 2 package (Hereinafter, this package or TOPPAN ToF SDK) using ToF camera compatible with *senSPure*™ SDK and the communication specifications between nodes.

The following cameras are currently supported for operation.

Table 1. Supported camera device

Model	Product code	Camera firmware
C11U	TPSC1AS1Z	3.1.0 or higher

1-2. Definitions of terms and abbreviations

Table 2. Terms and abbreviations

Term	Description
SDK	Software Development Kit
ROS	Robot Operating System

1-3. Related documents

When referring to this document, please also refer to the following related documents.

Table 3. Related documents

Related Documents	Contents
***senSPure*™ C11U User's Guide**	*senSPure*™ C11U 3D ToF camera User's Guide
TOPPAN ToF *senSPure*™ SDK Environment Setup Guide	Configuration of TOPPAN ToF *senSPure*™ SDK software environment
TOPPAN ToF *senSPure*™ SDK API Reference Manual	API specifications of the TOPPAN ToF *senSPure*™ SDK library

1-4. ROS 2 version

Table 4. ROS2 version

Environment	System version	ROS 2 version
Host computer OS	Windows10	ROS 2 Humble Hawksbill

	Windows11	
	Ubuntu 20.04LTS	
	Ubuntu 22.04LTS	
SoC NVIDIA Jetson AGX Orin	JetPack 5.0.1 (Ubuntu 20.04LTS 64bit)	
	JetPack 6.0 (Ubuntu 22.04LTS 64bit)	

1-5. Recommended environment for the host computer

1-5-1. Standalone environment (launch_alone.py)

The recommended environment on the host PC to run this package is described below.

1-5-1-1. Required system requirements

Table 5. Standalone: Required system requirements

Hardware	Required environment
CPU	4-core, 2 GHz or higher, 64 bit CPU
RAM	8GB or higher
Physical I/F	1x USB3.1(Gen1) port

1-5-1-2. Recommended system requirements

Table 6. Standalone: Reccomended system requirements

Hardware	Recommended environment
CPU	8-core, 3 GHz or higher, 64 bit CPU
RAM	8GB or higher
Physical I/F	1x USB3.1(Gen1) port

1-5-2. Rviz environment (launch_rviz2.py)

1-5-2-1. Required system environments

Table 7. rviz environment: Required system requirements

Hardware	Required environment
CPU	8-core, 3 GHz or higher, 64 bit CPU
RAM	8GB or higher
Physical I/F	1x USB3.1(Gen1) port

1-5-2-2. Recommended system environments

Table 8. rviz environment: Reccomended system requirements

Hardware	Recommended environment
CPU	8-core, 4 GHz or higher, 64 bit CPU

RAM	8GB or higher
Physical I/F	1x USB3.1(Gen1) port

1-5-3. Sample Viewer environment (launch_viewer.py)

1-5-3-1. Required system environments

Table 9. Sample Viewer environment: Required system requirements

Hardware	Required environment
CPU	8-core, 3 GHz or higher, 64 bit CPU
RAM	8GB or higher
Physical I/F	1x USB3.1(Gen1) port
Monitor	Full-HD or higher

1-5-3-2. Recommended system environments

Table 10. Sample Viewer environment: Reccomended system requirements

Hardware	Recommended environment
CPU	8-core, 4 GHz or higher, 64 bit CPU
RAM	8GB or higher
Physical I/F	1x USB3.1(Gen1) port
Monitor	Full-HD or higher

2. ROS 2 environment setup

2-1. Provided Package

The directory structure of the software provided by this package and the ROS 2 package included in it are shown below. {ROS2 path} is replaced with the directory where the provided software was extracted. (Windows: C: ¥Users ¥{User} ¥ros2; Linux:/home/{User}/ros2)
Note that PostFilter is not included in the SDK, but because it is placed in the same directory as the SDK and used, it is described as part of the SDK in this manual for convenience.

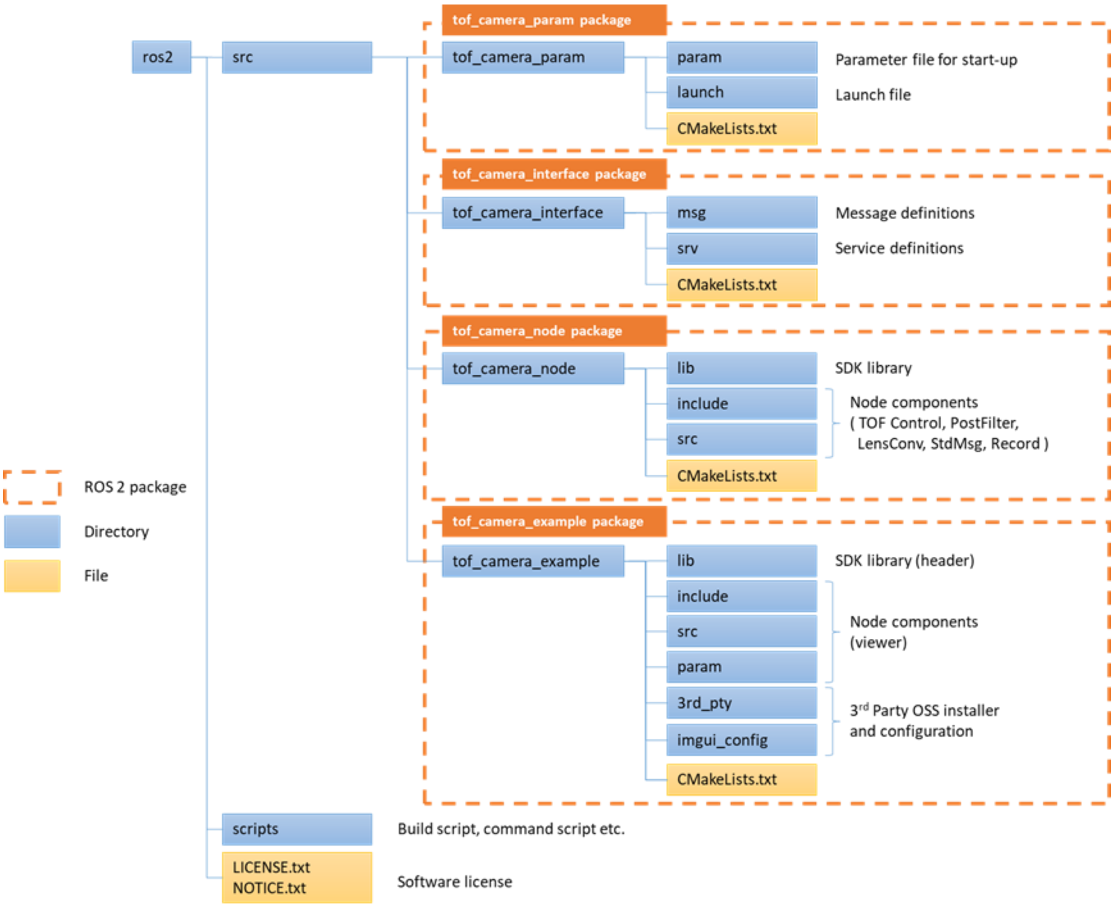


Figure 1. Provided package overview

Note: Do not extract to a directory containing double-byte characters. Also, for a Windows environment, do not extract to a directory containing more than 256 characters, including internal directories.

Table 11. ROS2 package overview

Directory		Contents
scripts		Scripts that build an environment for ROS 2, build an environment for this package, and instruct each Node.
src		Source code
	tof_camera_param	Startup parameters used by Nodes in tof_camera_node and Launch file to start Node.
	tof_camera_interface	Definition of types to communicate with each Node in

		tof_camera_node.
	tof_camera_node	Nodes for TOF camera and various conversion processing.
	tof_camera_example	Sample Viewer using Nodes in tof_camera_node
LICENSE.txt		License Terms
NOTICE.txt		Third-party Notices: OSS License Terms

2-2. Install ROS 2

Install ROS2 Humble Hawksbill (https://docs.ros.org/en/humble/index.html). If ROS 2 Foxy Fitzroy is already installed, please uninstall Python 3.8.3 and “Chocolatey”. Python can be uninstalled via the standard Windows Control Panel, and “Chocolatey” can be uninstalled by deleting its installation folder located at (C:¥ProgramData¥chocolatey).

Before proceeding, please refer to the “*TOPPAN ToF SDK Environment Setup Guide*,” which is located in the “***1-3. Related documents***” and follow the instructions for setting up the execution environment.

2-2-1. Windows

2-2-1-1. Download ROS 2 package

Navigate to ROS 2 Humble Hawksbill-Patch Release 8 (https://github.com/ros2/ros2/releases/tag/release-humble-20240222) via the website (https://github.com/ros2/ros2/releases), and download the following files from 'Assets'.

Download the following file from “Assets”

```
ros2-humble-20240222-windows-release-amd64.zip
```

If it does not exist, create the following directory using File Explorer:

```
C:¥dev
```

If the folder “C:¥dev¥ros2-windows” already exists, delete it or rename it. Extract the downloaded zip file into the above directory and ensure that the following file is present:

```
C:¥dev¥ros2-windows¥local_setup.bat
```

2-2-1-2. Obtaining zlib1.dll for Rviz2

The ROS 2 Humble binary package is missing “zlib1.dll”, which is required for “rviz2.exe” and may prevent it from launching.

Download the following file from the site (https://www.zlatkovic.com/pub/libxml/64bit/):

```
zlib-1.2.8-win32-x86_64.7z
```

Then, extract the archive and copy the contained zlib1.dll to the C:¥dev¥ros2-windows¥bin directory where the rviz2.exe of ROS2 Humble Hawksbill is located.

2-2-1-3. Uninstalling OpenSSL v1.1.1t

Open “Programs and Features (Uninstall or change a program)” in Windows, and if OpenSSL v1.1.1t is installed, please uninstall it. A new version of OpenSSL will be installed by chocolatey.

2-2-1-4. Installing Python

Start a command prompt with administrator privileges and install python using the following command.

```
cd {ROS2 path}
call .¥scripts¥windows¥install_python.bat
```

After execution is completed, the command prompt window will be closed.

From the Windows search, search for system environment variables, and then click variables for system environment variables. Make sure that the following environment variables are included in the system environment variable Path. If the environment variables are not included, add them to the beginning of Path.

Table 12. System environment variable settings (Python)

Variables	Value (Path)
Path	C:¥Python38
	C:¥Python38¥Scripts¥

2-2-1-5. System environment variable settings

From the Windows search, search for System Environment Variables and click System Environment Variable Variables. Add the following environment variables to the system environment variables.

Table 13. System environment variable settings

Variables	Value (Path)
Path	C:¥Program Files¥OpenSSL-Win64¥bin¥
	C:¥Program Files¥CMake¥bin
	C:¥tools¥opencv¥build¥x64¥vc15¥bin
	C:¥Program Files¥Graphviz¥bin
OPENSSL_CONF	C:¥Program Files¥OpenSSL-Win64¥bin¥openssl.cfg
OpenCV_DIR	C:¥tools¥opencv¥build

2-2-1-6. Installing dependent packages

Download the dependent packages for ROS 2 from Assets at the following website.
https://github.com/ros2/choco-packages/releases/tag/2022-03-15

- asio.1.12.1.nupkg
- bullet.3.17.nupkg
- cunit.2.1.3.nupkg
- eigen3.3.4.nupkg
- tinyxml-usestl.2.6.2.nupkg
- tinyxml2.6.0.0.nupkg

Place the downloaded file in the following location.

```
{ROS2 path}¥scripts¥windows¥choco_packages
```

2-2-1-7. Installing the ROS 2-related package

Start the command prompt with administrator privileges and install the other related packages using the following commands.

```
cd {ROS2 path}
call .¥scripts¥windows¥install_ros2.bat
```

When executing the command in a Proxy environment, rewrite the following parts of "install_ros2.bat", and then execute "install_ros2.bat". Change {Proxy Address} according to the execution environment.

(Before modification)

```
rem set https_proxy={Proxy address}
```

(After modification)

```
set https_proxy={Proxy address}
```

After the execution is completed, the command prompt window is closed. The files placed in the following locations before installation are not used and can be deleted.

```
{ROS2 path}¥scripts¥windows¥choco_packages
```

2-2-2. Installing ROS2 (Ubuntu/Jetson AGX Orin)

2-2-2-1. Source Set

For Humble Hawksbill, binary packages are not available for Ubuntu Linux 20.04, so it's necessary to build and install it from source. This installation script automates the build and installation process from source, following the procedures outlined in the official ROS 2 documentation. Note that in an Ubuntu 22.04 environment, installation via binary packages is also possible in addition to building from source. For detailed instructions, please refer to the ROS 2 official documentation notes [Note].

For questions regarding the installation or environment setup of ROS itself, please contact the official ROS community or refer to their documentation.

- src/eclipse-iceoryx
- src/ros-perception/image_common/polled_camera
- src/ros/ros_tutorials/ros_tutorials
- src/ros/ros_tutorials/roscpp_tutorials
- src/ros/ros_tutorials/rospy_tutorials
- src/ros2/rosbag2/rosbag2_storage_evaluation

Note: ROS 2 official documentation: https://docs.ros.org/en/humble/Installation.html

2-2-2-2. Installation Procedure

Launch a terminal and execute the installation using the following commands:

```
cd {ROS2 path}
./scripts/ubuntu/install_ros2.sh
```

If you are running in a proxy environment, rewrite the following part of "install_ros2.sh" and then execute "install_ros2.sh". Please change the {Proxy Address} part according to your execution environment.

(Before modification)

```
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o
/usr/share/keyrings/ros-archive-keyring.gpg
```

(After modification)

```
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -x {Proxy Address} -o
/usr/share/keyrings/ros-archive-keyring.gpg
```

ROS 2 Humble Hawksbill is installed under the "/opt/ros/humble" directory. The source code used for building is saved in "/opt/ros/humble-ws", but this directory can be deleted after installation.

2-3. Build the SDK library

2-3-1. Preparing the SDK Library

Prepare the pre-built SDK library and header files described in the provided software section of the "TOPPAN ToF *senSPure*™ SDK Environment Setup Guide" in "***1-3. Related documents***".

2-3-2. Integration of the SDK Library

Copy all the files under the directory containing the built SDK library to the directory shown below.

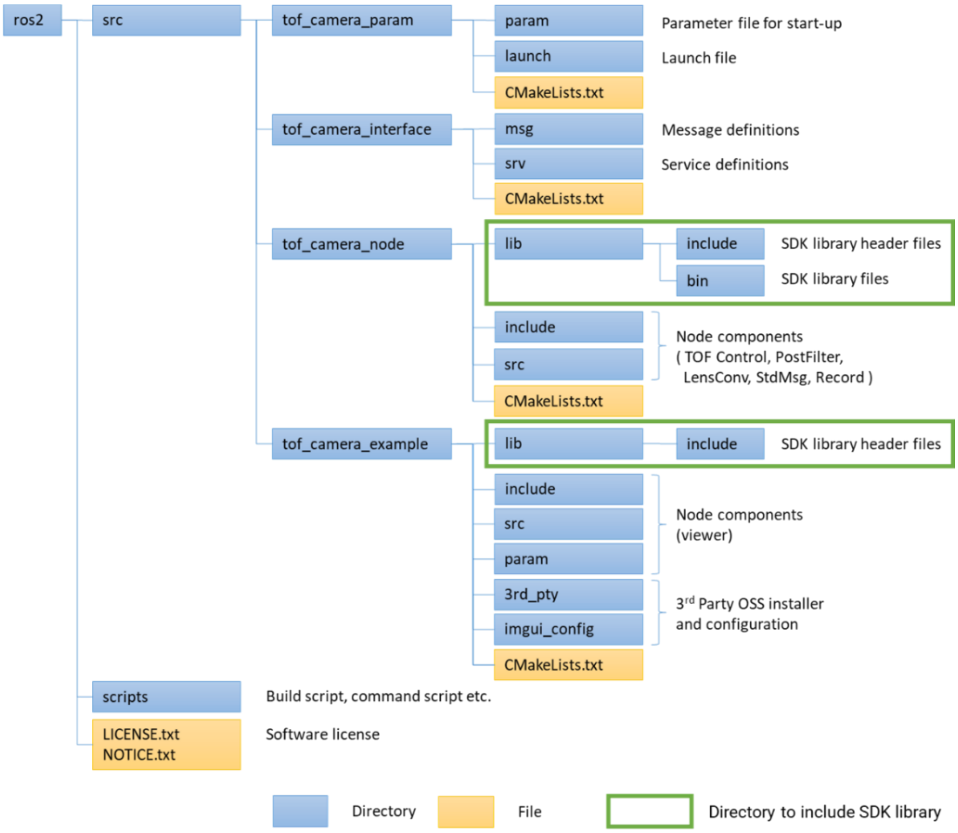


Figure 2. SDK integration

Table 14. TOPPAN ToF SDK integration

OS	Copy source	Copy destination
Windows	{SDK path}note1¥build¥include	{ROS2 path}note2¥src¥tof_camera_node¥lib¥include
		{ROS2 path}¥src¥tof_camera_example¥lib¥include
	{SDK path}¥build¥bin¥Release	{ROS2 path}¥src¥tof_camera_node¥lib¥bin
Linux/Ubuntu	{SDK path}/build/include	{ROS2 path}/src/tof_camera_node/lib/include
		{ROS2 path}/src/tof_camera_example/lib/include
	{SDK path}/build/bin	{ROS2 path}/src/tof_camera_node/lib/bin

***Note*[1]**: Replace {SDK path} with the directory where the SDK library was extracted.
***Note*[2]**: Do not specify a directory containing multi-byte characters for {ROS2 path}.

2-3-3. Building command (Windows)

2-3-3-1. Installing OSS for Sample Viewer

Launch a command prompt and execute the installation using the following commands:

```
cd {ROS2 path}¥src¥tof_camera_example¥3rd_pty
call .¥install¥windows¥install_glfw_imgui.bat
```

2-3-3-2. Build Procedure

Launch a command prompt and execute the build using the following commands:

```
cd {ROS2 path}
call .¥scripts¥windows¥build.bat
```

2-3-4. Building command (Ubuntu/Jetson AGX Orin)

2-3-4-1. Installing OSS for Sample Viewer (Ubuntu)

Launch a terminal and execute the installation using the following commands:

```
cd {ROS2 path}/src/tof_camera_example/3rd_pty
./install/ubuntu/install_glfw_imgui.sh
```

If the OpenGL driver is not installed on your host PC environment, additionally install the OpenGL driver using the following command:

```
./install/ubuntu/install_gl_mesa.sh
```

2-3-4-2. Installing OSS for Sample Viewer (Jetson AGX Orin)

Launch a terminal and execute the installation using the following commands:

```
cd {ROS2 path}/src/tof_camera_example/3rd_pty
./install/ubuntu/install_glfw_imgui.sh
```

2-3-4-3. Build Procedure (Ubuntu/Jetson AGX Orin)

Execute the build using the following commands:

```
cd {ROS2 path}
./scripts/ubuntu/build.sh
```

3. Node Structure

3-1. Node Structure overview

The diagram of Node structure provided by this package "tof_camera_node" is shown below.

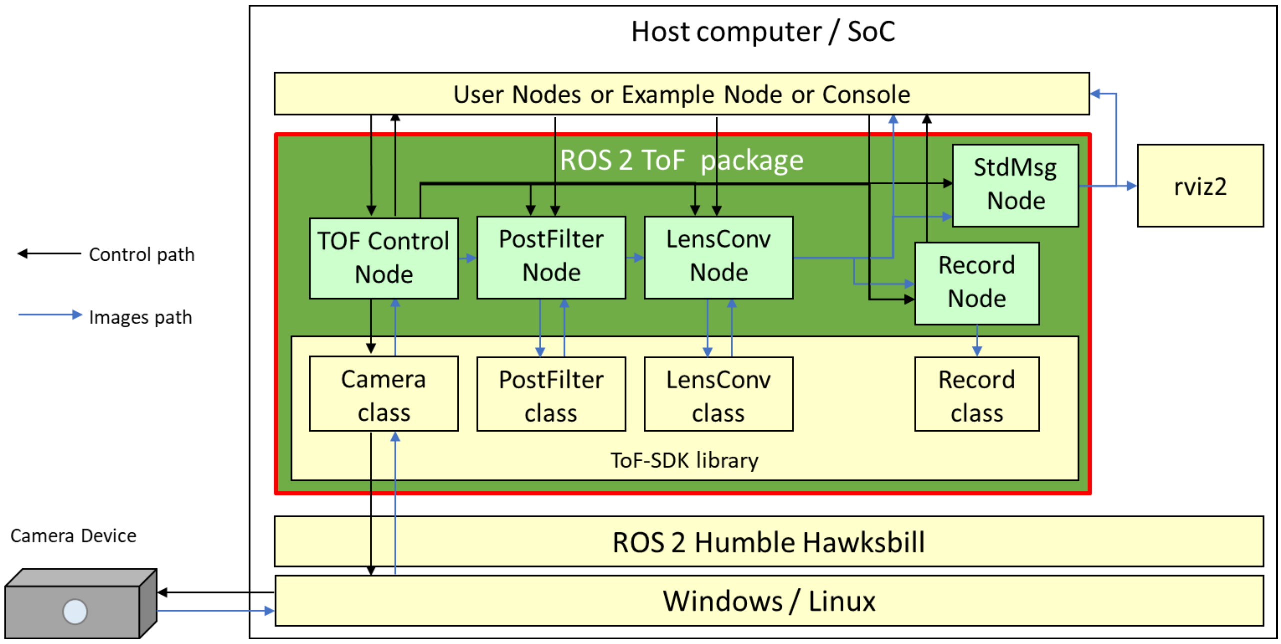


Figure 3. Node structure diagram

Table 15. Node structure overview

Node name	Overview
TOF Control	Depth, IR, and Raw images are received from the Camera Device and sent as Topics.
PostFilter	Depth and IR images output from TOF Control are smoothed and processed for abnormal points and sent as Topics.
LensConv	Cartesian coordinate transformation, distortion correction, and point group transformation are performed on the Depth image and IR image output from the PostFilter, and the Depth image, IR image, and Point Cloud data are sent as Topics.
StdMsg	The Depth image, IR image, Raw image, and Point Cloud data output from LensConv are converted into ROS standard Topics and sent as Topics.
Record	The Depth image, IR image, and Raw image output from LensConv are saved as files.
Example	Sample Viewer Node included in the tof_camera_example package.
rviz2	ROS2 standard monitoring viewer.

3-2. Inter-Node communications

Inter-Node communication diagram of this package (tof_camera_node) is shown below.

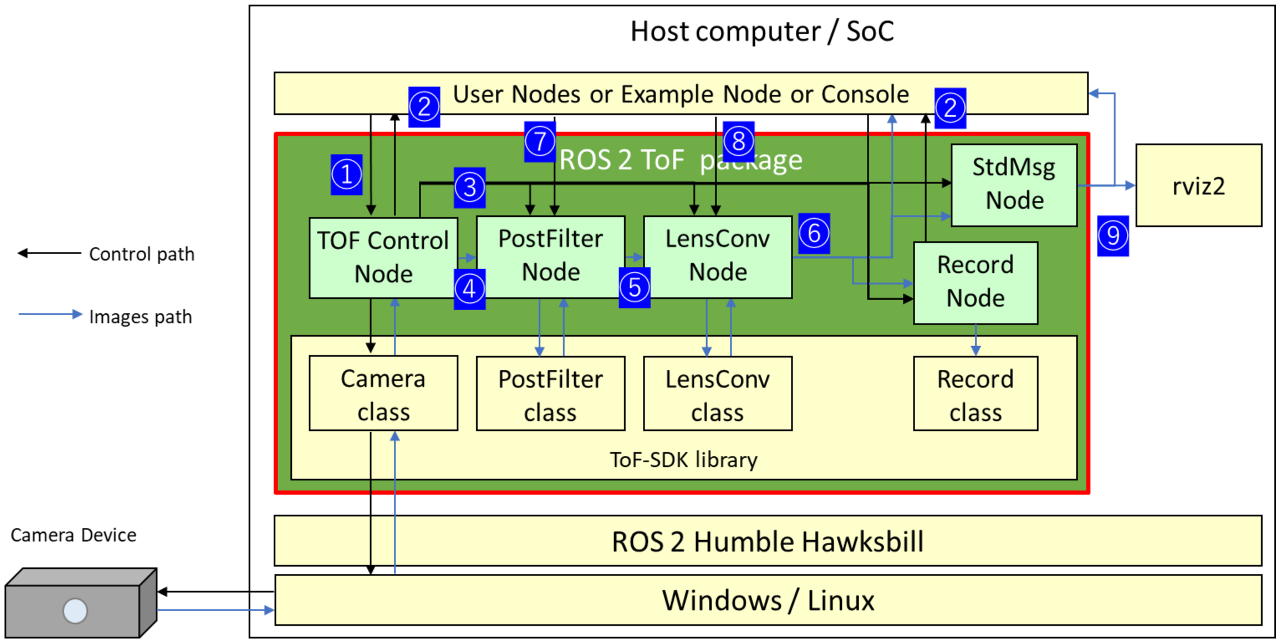


Figure 4. Communication between nodes

Table 16. Node communication topics

No.	Topic Name	Communication Type	Src Node	Dst Node	Overview
①	krm/get_dev_list	GetDevList (Service)	User Example	TOF Control	Get Device List
	krm/open_dev	OpenDev (Service)			Open Camera Device
	krm/close_dev	CloseDev (Service)			Close Camera Device
	krm/get_dev_info	GetDevInfo (Service)			Get Device Information
	krm/get_fov	GetFov (Service)			Get FoV Information
	krm/get_ext_trigger_type	GetExtTriggerType (Service)			Get External Trigger Type Information
	krm/get_ext_trigger_offset	GetExtTriggerOffset (Service)			Get External Trigger Signal Offset Information
	krm/get_mode_list	GetModeList (Service)			Get Operation Mode List Information
	krm/get_mode	GetMode (Service)			Get Current Operation Mode Information
	krm/get_img_kinds	GetImgKinds (Service)			Get Current Output Image kind Information
	krm/get_img_format	GetImgFormat (Service)			Get Current Image Format Information
	krm/get_post_filt_info	GetPostFiltInfo (Service)			Get Current PostFilter Processing Information
	krm/get_lens_info	GetLensInfo (Service)			Get Current Lens System Conversion Processing Parameters
	krm/get_light_times	GetLightTimes (Service)			Get Current Light Emission Count Information
	krm/get_ae_state	GetAEState (Service)			Get Current AE Function State Information
	krm/get_ae_interval	GetAEInterval (Service)			Get Current AE Function Control Interval Information
	krm/get_raw_sat_threshold	GetRawSatThreshold (Service)			Get Current RAW Saturation Threshold Information
	krm/get_ir_dark_thr	GetIrDarkThreshold			Get Current IR Invalid Threshold

	eshold	(Service)			Information
	krm/get_int_supp_info	GetIntSuppInfo (Service)			Get Current Interference Prevention Function Information
	krm/set_ext_trigger_type	SetExtTriggerType (Service)			Set External Trigger Type
	krm/set_ext_trigger_offset	SetExtTriggerOffset (Service)			Set External Trigger Signal Offset
	krm/set_mode	SetMode (Service)			Set Operation Mode
	krm/set_img_kinds	SetImgKinds (Service)			Set Output Image Type
	krm/set_light_times	SetLightTimes (Service)			Set Light Emission Count
	krm/set_ae_state	SetAEState (Service)			Set AE Function State
	krm/set_ae_interval	SetAEInterval (Service)			Set AE Function Control Interval
	krm/set_int_supp	SetIntSupp (Service)			Set RAW Saturation Threshold Information
	krm/set_ir_dark_threshold	SetIrDarkThreshold (Service)			Set IR Invalid Threshold
	krm/set_raw_sat_threshold	SetRawSatThreshold (Service)			Set Interference Prevention Function
	krm/tof_ctrl	TofCtrl (Service)			Control Image Output from Camera
	krm/get_play_target	GetPlayTarget (Service)			Get File Playback Target Directory Information
	krm/get_play_time	GetPlayTime (Service)			Get Current Playback Time of File Playback
	krm/get_play_status	GetPlayStatus (Service)			Get File Playback Status
	krm/set_play_target	SetPlayTarget (Service)			Set File Playback Target Directory
	krm/set_play_ctrl	SetPlayCtrl (Service)			Control File Playback Function
②	krm/notify	Notify (Message)	TOF Control Record	User Example	Abnormal Notification, State Transition Notification
③	krm/event_chg_prop	EventChgProp (Message)	TOF Control	PostFilter LensConv StdMsg Record	Notification of Camera-Specific Information Changes
	krm/event_chg_fmt	EventChgFmt (Message)			Notification of Format Information Changes
④	krm/tof_out	FrameData (Message)	TOF Control	PostFilter	Notify Depth Image, IR Image, and Sensor RAW Image Output from Camera
⑤	krm/post_filt_out	FrameData (Message)	PostFilter	LensConv	Notify Depth Image and IR Image After Filter Conversion
⑥	krm/lens_out	FrameData (Message)	LensConv	User Example StdMsg Record	Notify Depth Image, IR Image, and Point Cloud Data After Lens System Conversion Processing
⑦	krm/psbl_post_filt	PsblPostFilt (Service)	User Example	PostFIlter	Get Usability Information of Each PostFilter Function
	krm/set_post_filt	SetPostFilt (Service)			Set Usage of Each PostFilter Function
	krm/set_post_filt_prm	SetPostFiltPrm (Service)			Set PostFilter Parameters
⑧	krm/psbl_lens_conv	PsblLensConv (Service)	User Example	LensConv	Get Usability Information of Each LensConv Function
	krm/set_lens_conv	SetLensConv (Service)			Set Usage of Each LensConv Function
	krm/set_pcd_pos	SetPcdPos (Service)			Set Origin Position and Rotation Information of World Coordinate Transformation
	krm/set_pcd_color	SetPcdColor (Service)			Set Color Information
⑨	krm/record_ctrl	RecordCtrl (Service)	User Example	Record	Control File Saving Function
⑩	krm/std/cam_info	sensor_msgs/CameraInfo (Message)	StdMsg	rviz2 etc.	Notify ROS Standard Camera Information
	krm/std/depth	sensor_msgs/Image			Notify ROS Standard Depth Image

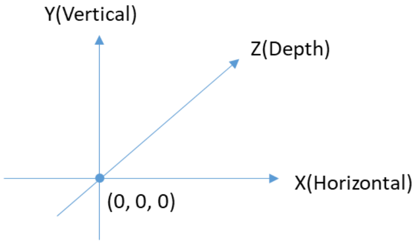
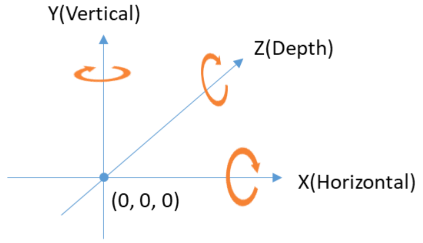
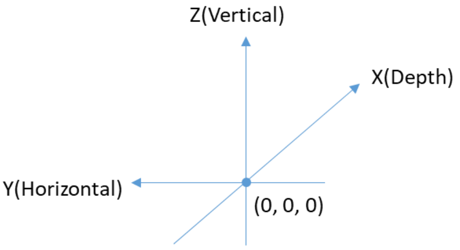
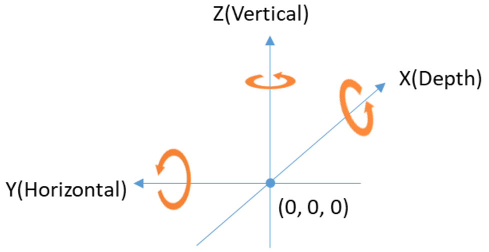
		(Message)			
	krm/std/ir	sensor_msgs/Image (Message)			Notify ROS Standard IR Image
	krm/std/raw1 krm/std/raw2 krm/std/raw3 krm/std/raw4	sensor_msgs/Image (Message)			Notify ROS Standard Sensor RAW Image (G1-G4)
	krm/std/pcd	sensor_msgs/PointCloud2 (Message)			Notify ROS Standard Point Cloud Data

3-2-1. Point cloud output of LensConv Node and StdMsg Node

The LensConv class outputs a point cloud of left-handed coordinate system, and each coordinate is in [unit: mm] as described in "***1-3. Related documents***".

As shown in the following table, the LensConv Node outputs a point cloud of left-handed coordinate system in mm, which is the same as the LensConv class. The StdMsg Node converts to a form that can be displayed by rviz2, and outputs the point cloud of right-handed coordinate system in [unit: m].

Table 17. Point cloud output of LensConv Node and StdMsg Node

Node	Coordinate system	Rotation direction
LensConv Node (Left-handed coordinate)		 (Counterclockwise around the axis)
StdMsg Node (Right-handed coordinate)		 (Clockwise around the axis)

Note that the coordinate system set for "lensconv_param.yaml" and "SetPcdPos" described later should be the left-handed coordinate system handled by LensConv Node.

In addition, the output point cloud data differs in part as follows:

Table 18. Point Cloud Output Data Format for LensConv Node and StdMsg Node

Node	Data format (sensor_msgs/PointCloud2)		
LensConv Node	PointField::UINT32	color	Invalid points contain UINT32_MAX. If PCD_COLOR_IR is set with SetPcdColor, valid points contain the IR value of the same pixel.
	PointField::FLOAT32	x	Point cloud data for the same number of pixels as the Depth image, including invalid points, is output.
	PointField::FLOAT32	y	
	PointField::FLOAT32	z	
StdMsg Node	PointField::FLOAT32	x	Point cloud data excluding invalid points is

	PointField::FLOAT32	y	output.
	PointField::FLOAT32	z	

Note: Starting from ROS2 package version 1.4.2, the storage order of point cloud data output from the StdMsg Node has been changed. If you are developing with a package version of 1.4.1 or earlier, or if you have output data from the StdMsg Node that was acquired with an older version, you must be careful when reading the data. However, if you are acquiring data and developing by referencing the PointField information, this change will not affect you. For details, please refer to "***10-3. PointField Information (Data Array Change)***".

3-2-2. QoS setting

The following QoS (Quality of Service) settings are specified for Topics sent as Messages. Create a subscriber with the same settings on the receiving side.

Table 19. QoS setting

Topic name	History	Depth	reliability	Durability
krm/event_chg_prop krm/event_chg_fmt	Keep last	1	Reliable	Volatile
krm/tof_out krm/post_filt_out krm/lens_out krm/notify krm/std/cam_info krm/std/depth krm/std/ir krm/std/raw1 krm/std/raw2 krm/std/raw3 krm/std/raw4 krm/std/pcd	Keep last	5	Reliable	Volatile

3-2-3. FrameID

The following string is set in the frame_id of std_msgs/Header, which is included in FrameData, sensor_msgs/Image, sensor_msgs/PointCloud2, and sensor_msgs/CameraInfo.

"krm" + "cam_index"(which described in "tof_param.yaml")

Expamle) When cam_index = 0, FrameID defines "krm0"

4. Message/Service type definition

This package defines the Message/Service type definition as follows.

4-1. Message type definition

4-1-1. Version

Table 20. Version definition

Description	• Version information.		
Parameters	**Type**	**Name**	**Remark**
	uint8	major	Major version
	uint8	minor	Minor version
	uint16	rev	Revision

4-1-2. MinMaxValue8

Table 21. MinMaxValue8 definition

Description	• Indicates the minimum, maximum, and current values of uint8 type parameters.		
Parameters	**Type**	**Name**	**Remark**
	uint8	min	Minimum value
	uint8	max	Maximum value
	uint8	value	Current value

4-1-3. MinMaxValue16

Table 22. MinMaxValue16 definition

Description	• Indicates the minimum, maximum, and current values of a uint16 type parameter.		
Parameters	**Type**	**Name**	**Remark**
	uint16	min	Minimum value
	uint16	max	Maximum value
	uint16	value	Current value

4-1-4. MinMaxValue32

Table 23. MinMaxValue32 definition

Description	• Indicates the minimum, maximum, and current values of the uint32 type parameter.		
Parameters	**Type**	**Name**	**Remark**
	uint32	min	Minimum value
	uint32	max	Maximum value
	uint32	value	Current value

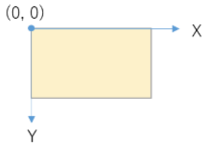
4-1-5. Range

Table 24. Range definition

Description	• Indicates distance range information.		
Parameters	Type	Name	Remark
	uint16	min	Minimum distance [mm].
	uint16	max	Maximum distance [mm].

4-1-6. Point2d

Table 25. Point2d definition

Description	• The position of the pixel in the image plane. • The pixel position is the one with the top-left as the origin. 		
Parameters	Type	Name	Remark
	uint16	x	X-axis coordinates [pixel].
	uint16	y	Y-axis coordinates [pixel].

4-1-7. CameraType

Table 26. CameraType definition

Description	• Indicates the type of camera device used.				
Parameters	Type	Name	Remark		
	uint8	type	Camera device type		
			C11_USB	0	C11U camera (USB Interface)
			PLAYBACK	1	For file playback

4-1-8. ConnDevice

Table 27. ConnDevice definition

Description	• Indicates information on the connected device.		
Parameters	Type	Name	Remark
	uint16	id	Device ID
	string	name	Device Name

4-1-9. DeviceInfo

Table 28. DeviceInfo definition

Description	• Indicates the equipment information of the camera device.		
Parameters	Type	Name	Remark
	uint32	hw_kind	HW model number

			Upper 16 bits: Sensor type number Lower 16 bits: Lens type number
	uint32	serial_no	Device serial number
	Version	map_ver	Camera configuration MAP version
	Version	firm_ver	Camera firmware version
	uint32	adjust_no	Adjustment number
	uint16	ld_wave	Light source wavelength [nm].
	uint16	ld_enable	Information of light source (Number of lights) The valid information of each light source is entered in order from the least significant bit. (0b: Disable, 1b: Enable) [0] LD1, [1] LD2, [2] LD3 ...
	uint16	correct_calib	Compensation Calibration Revision.

4-1-10. CamFov

Table 29. CamFov definition

Description	• Indicates the FOV information of the camera device.		
Parameters	**Type**	**Name**	**Remark**
	uint16	horz	Horizontal field of view [degree × 100]
	uint16	vert	Vertical field of view [degree × 100]

4-1-11. ExtTriggerType

Table 30. ExtTriggerType definition

Description	• Indicates the external trigger type of the camera device.				
Parameters	**Type**	**Name**	**Remark**		
	uint8	ext_trigger_type	External trigger type		
			EXT_TRG_STANDALONE	1	Standalone
			EXT_TRG_SLAVE	2	Slave (Secondary)
			EXT_TRG_MASTER	3	Master (Primary)

4-1-12. ImgOutKind

Table 31. ImgOutKind definition

Description	• Indicates information on the combination of image types output from the camera.				
Parameters	**Type**	**Name**	**Remark**		
	uint8	kind	Output image kind		
			OUT_IMG_DEPTH	0	Depth only
			OUT_IMG_IR	1	IR only
			OUT_IMG_DEPTH_IR	2	Depth + IR
			OUT_IMG_DEPTH_IR_RAW	3	Depth + IR + Gate RAW (Sensor)
			OUT_IMG_RAW	4	Gate RAW (Sensor)

4-1-13. ModeInfo

Table 32. ModeInfo definition

Description	• Indicates operating mode information.		
Parameters	**Type**	**Name**	**Remark**
	uint8	id	Operating mode ID
	string	description	Operation mode description
	ImgOutKind []	img_out	Output image type
	Range	dist_range	Measuring range [mm].
	uint16	fps	Received frame rate [fps × 100].
	uint8	thin_w	Horizontal thinning number (1 / thin_w)
	uint8	thin_h	Vertical thinning number (1 / thin_h)
	Point2d	crop	Image cut-out position from sensor pixels
	bool	light_times	Changeable number of “light times” true: Yes false: No
	uint16	range_calib	Range calibration revision

4-1-14. ImageFormat

Table 33. ImageFormat definition

Description	• Indicates the format information of the image data.		
Parameters	**Type**	**Name**	**Remark**
	uint16	width	Image data width [pixel].
	uint16	height	Image data height [pixel].
	Point2d	active_start	Active pixel start position [pixel].
	uint16	active_w	Active pixel width [pixel].
	uint16	active_h	Active pixel height [pixel].
	uint32	pixels	Number of image data pixels (width x height) [pixel].
	uint8	bpp	Size of one image data pixel [byte].
	uint64	size	Image data size [byte].

4-1-15. ImageFormats

Table 34. ImageFormats definition

Description	• Indicates image format information for all image types.		
Parameters	**Type**	**Name**	**Remark**
	ImageFormat[]	data	Image format information

4-1-16. PostFiltInfo

Table 35. PostFiltInfo definition

Description	• Indicates PostFilter processing information.

Parameters	Type	Name	Remark
	bool	cam_med_filt	Median filter executed in the camera device
	bool	cam_bil_filt	Bilateral filter executed in the camera device
	bool	cam_fly_p_filt	Flying pixel filter executed in the camera device

4-1-17. LensInfo

Table 36. LensInfo definition

Description	• Indicates Lens system conversion parameters.		
Parameters	Type	Name	Remark
	uint16	sens_w	Sensor width [pixel].
	uint16	sens_h	Sensor height [pixel].
	uint32	focal_len	Focal length (Fixed point: 12 bit integer part, 20 bit decimal part)
	uint8	thin_w	Horizontal thinning number (1 / thin_w)
	uint8	thin_h	Vertical thinning number (1 / thin_h)
	Point2d	crop	Image cropping position from the active area
	bool	cam_dist	Distortion correction performed in the camera device true: implemented false: not implemented
	uint64 [9]	dist	Distortion correction Parameters [fx, fy, cx, cy, k1, k2, p1, p2, k3]. (Fixed-point: 1 bit for the sign part, 16 bit for the integer part, 47 bit for the decimal part)
	uint16	lens_calib	Lens calibration revision

4-1-18. IntSuppModeType

Table 37. IntSuppModeType definition

Description	• Indicates the type of interference suppression function mode.				
Parameters	Type	Name	Remark		
	uint8	int_supp_mode_type	Interference suppression function mode		
			INT_SUPP_MODE_OFF	0	Off
			INT_SUPP_MODE_MANUAL	1	Manual assign
			INT_SUPP_MODE_AUTO	2	Auto @test function

4-1-19. IntSuppInfo

Table 38. IntSuppInfo definition

Description	• Indicates information on the interference suppression function.		
Parameters	Type	Name	Remark
	IntSuppModeType	mode	Interference suppression function mode
	MinMaxValue8	prm_m	Manual assign number for Manual mode
	MinMaxValue8	prm_a1	Auto mode param 1. @test function
	MinMaxValue8	prm_a2	Auto mode param 2. @test function

	MinMaxValue8	prm_a3	Auto mode param 3. @test function

4-1-20. PlayTime

Table 39. PlayTime definition

Description	• Indicates playback time information.		
Parameters	**Type**	**Name**	**Remark**
	uint32	total	Total frame number
	uint32	current	Current frame number (starting from 0)

4-1-21. FrameInfo

Table 40. FrameInfo definition

Description	• Indicates additional information on the received frame.				
Parameters	**Type**	**Name**	**Remark**		
	uint32	number	Frame number		
	uint16	frm_err	Abnormal frame information The presence or absence of an abnormality is entered at each bit position. Mask and judge by the following values.		
			FRM_ERR_MASK_DROP	1bit	Frame discontinuity info. (0: Continuous, 1: Discontinuous)
			FRM_ERR_MASK_CRC	2bit	CRC Error info. (0: Non-error, 1: Error)
	uint16	temperature	Temperature information in the camera device (Fixed point: Integer part 10 bits, Decimal part 6 bits) (UINT16_MAX: invalid temperature value)		
	uint16	light_cnt	Light times value		
	uint8	conv_stat	Conversion status information (Fixed decimal point: 10 bits for integer part, 6 bits for decimal part) Mask and judge by the following values.		
			CONV_STAT_MASK_CRCT_DIST	1bit	Distortion correction
			CONV_STAT_MASK_FILT_MED	2bit	Median filter
			CONV_STAT_MASK_FILT_BIL	4bit	Bilateral filter
			CONV_STAT_MASK_FILT_FLY	8bit	Flying pixel filter

4-1-22. FrameImage

Table 41. FrameImage definition

Description	• The following image types are available • Depth image • IR image • Gate RAW image • In the case of Depth images, krm:: INVALID_DEPTH or krm:: SATURATION_DEPTH is set for invalid pixels. • The unit of the depth data value is [mm].

	Type	Name	Remark
Parameters	FrameInfo	info	Frame information
	sensor_msgs/Image	image	Image data

4-1-23. FrameData

Table 42. FrameData definition

Description	• Indicates the Depth image, IR image, sensor RAW image and Point Cloud data contained in one frame.				
Parameters	Type	Name	Remark		
	bool	stopped	Indicates that reception has stopped true : Reception stopped false : Reception is in progress. When true, parameters other than STOPPED cannot be referenced.		
	bool	discontinuity	Indicates discontinuity of received image true : discontinuous false : Continuous		
	uint16	rcv_fps	Received frame rate [fps × 100].		
	PlayTime	play_time	Playback position information (only valid for file playback)		
	FrameImage	depth	Depth image		
	FrameImage	ir	IR image		
	FrameImage	raw1	Sensor Gate RAW G1 image		
	FrameImage	raw2	Sensor Gate RAW G2image		
	FrameImage	raw3	Sensor Gate RAW G3image		
	FrameImage	raw4	Sensor Gate RAW G4image		
	sensor_msgs/PointCloud2	pcd	Point Cloud data		
	uint8	pcd_kind	Point Cloud data type		
			PCD_XYZ	0	PCD
			PCD_RGBXYZ	1	RGB PCD
			PCD_IRXYZ	2	IR PCD

4-1-24. Notify

Table 43. Notify definition

Description	• Indicates information on notification of irregularities.				
Parameters	Type	Name	Remark		
	uint8	notify	Notification information		
			ERR_PARAM	1	Startup parameter anomalies (tof_param.yaml)
			ERR_TIMEOUT	2	Camera error notification (Image reception abnormal stop)
			ERR_SYSTEM	3	System error notification

			PLAY_REACHED_EOF	10	File playback notification (End of file stop)
			REC_REACHED_EOF	11	File save notification (End of file stop)
			REC_ERR_SYSTEM	12	File save notification (File save error)

4-1-25. EventChgProp

Table 44. EventChgProp definition

Description	• Indicates camera-specific information (device information, operating mode information, Lens system conversion processing parameters, FOV and PostFilter processing information) used in information change notifications.		
Parameters	**Type**	**Name**	**Remark**
	DeviceInfo	dev_info	Device information
	ModeInfo	mode_info	Operation mode information
	LensInfo	lens_info	Lens system conversion processing parameters
	CamFov	fov	Camera viewing angle
	PostFiltInfo	post_filt_info	PostFilter processing information

4-1-26. EventChgFmt

Table 45. EventChgFmt definition

Description	• Indicates the image format information used in the format information change notification.		
Parameters	**Type**	**Name**	**Remark**
	ImageFormats	formats	Image format information for all image types.

4-1-27. PostFilterPrm

Table 46. PostFilterPrm definition

Description	• Indicates information on the PostFilter configuration.		
Parameters	**Type**	**Name**	**Remark**
	uint8	median_ksize	Median filter size (3 or 5)
	uint8	bil_ksize	Bilateral filter size (3 or 5)
	float64	bil_sigma_depth	Smoothing parameter for the Depth value of the bilateral filter
	float64	bil _sigma_ir	Smoothing parameter for the IR value of the bilateral filter
	float64	bil_sigma_space	Smoothing parameter for spatial direction of the Bilateral filter
	uint8	flyp_ksize	Flying pixel filter size (3 or 5)
	bool	flyp_log	Flying pixel filter processing method true : Ratio false : Differential @test function

	uint16	flyp_thr	Flying pixel filter edge threshold
	bool	flyp_fast_proc	Flying pixel filter processing method true : Speed priority false : Accuracy priority

4-1-28. PosOrgRotation

Table 47. PosOrgRotation definition

Description	• Indicates information on the LensConv world coordinate transformation settings.		
Parameters	**Type**	**Name**	**Remark**
	int16	offset_x	Origin position: offset in X-axis direction [mm].
	int16	offset_y	Origin position: offset in Y-axis direction [mm].
	int16	offset_z	Origin position: offset in Z-axis direction [mm].
	float32	rotation_pitch	Pitch rotation angle [degree].
	float32	rotation_yaw	Yaw rotation angle [degree].
	float32	rotation_roll	Roll rotation angle [degree].

4-2. Service type definition

4-2-1. GetDevList

Table 48. GetDevList definition

Description	• Used when getting the list of connected camera devices.		
Request structure	**Type**	**Name**	**Remark**
	CameraType	type	Camera type to be used
Response structure	**Type**	**Name**	**Remark**
	bool	result	true : Success false : Failure
	ConnDevice []	dev_list	Device information

4-2-2. OpenDev

Table 49. OpenDev definition

Description	• Used when opening the specified camera device. • Specify the device ID obtained with GetDevList for dev_id.		
Request structure	**Type**	**Name**	**Remark**
	CameraType	type	Camera type to be used
	uint16	dev_id	Device ID
Response structure	**Type**	**Name**	**Remark**
	bool	result	true : Success false : Failure

4-2-3. CloseDev

Table 50. CloseDev definition

Description	• Used when closing the specified camera device.		
Request structure	Type	Name	Remark
	uint8	reserved	reserved
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure

4-2-4. GetDevInfo

Table 51. GetDevInfo definition

Description	• Used when getting device information. • Fails if the camera device is not open.		
Request structure	Type	Name	Remark
	uint8	reserved	reserved
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure
	GetDevInfo	dev_info	Device information

4-2-5. GetFov

Table 52. GetFov definition

Description	• Used when getting FoV information. • Fails if the camera device is not open.		
Request structure	Type	Name	Remark
	uint8	reserved	reserved
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure
	CamFov	cam_fov	FoV information

4-2-6. GetExtTriggerType

Table 53. GetExtTriggerType definition

Description	• Used when getting external trigger type information. • Fails if the camera device is not open.		
Request structure	Type	Name	Remark
	uint8	reserved	reserved
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure
	ExtTriggerType	type	External trigger type

4-2-7. GetExtTriggerOffset

Table 54. GetExtTriggerOffset definition

Description	• Used when getting external trigger signal offset information. • Fails if the camera device is not open.		
Request structure	**Type**	**Name**	**Remark**
	uint8	reserved	reserved
Response structure	**Type**	**Name**	**Remark**
	bool	result	true : Success false : Failure
	uint8	offset	External trigger signal offset (pulse width)

4-2-8. GetModeList

Table 55. GetModeList definition

Description	• Used when getting the operation mode list information. • Fails if the camera device is not open.		
Request structure	**Type**	**Name**	**Remark**
	uint8	reserved	reserved
Response structure	**Type**	**Name**	**Remark**
	bool	result	true : Success false : Failure
	ModeInfo []	mode_list	Operating mode list information

4-2-9. GetMode

Table 56. GetMode definition

Description	• Used to get the current operation mode. • Fails if the camera device is not open.		
Request structure	**Type**	**Name**	**Remark**
	uint8	reserved	reserved
Response structure	**Type**	**Name**	**Remark**
	bool	result	true : Success false : Failure
	uint8	mode	Operation mode

4-2-10. GetImgKinds

Table 57. GetImgKinds definition

Description	• Used to get the current output image type information. • Fails if the camera device is not open.		
Request structure	**Type**	**Name**	**Remark**
	uint8	reserved	reserved
Response	**Type**	**Name**	**Remark**

structure	bool	result	true : Success false : Failure
	ImgOutKind	img_out	Output image type

4-2-11. GetImgFormat

Table 58. GetImgFormat definition

Description	• Used to get the current image format information. • Fails if the camera device is not open.		
Request structure	Type	Name	Remark
	uint8	reserved	reserved
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure
	ImageFormats	img_fmts	Image format information

4-2-12. GetPostFiltInfo

Table 59. GetPostFiltInfo definition

Description	• Used to get the current PostFilter processing information. • Fails if the camera device is not open.		
Request structure	Type	Name	Remark
	uint8	reserved	reserved
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure
	PostFiltInfo	post_filt_info	PostFilter processing information

4-2-13. GetLensInfo

Table 60. GetLensInfo definition

Description	• Used to get the current Lens conversion processing parameters. • Fails if the camera device is not open.		
Request structure	Type	Name	Remark
	uint8	reserved	reserved
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure
	LensInfo	lens_info	Lens conversion processing parameters

4-2-14. GetLightTimes

Table 61. GetLightTimes definition

Description	• Used when getting the current light emission count (Light Times) information. • Fails if the camera device is not open.

Request structure	Type	Name	Remark
	uint8	reserved	reserved
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure
	MinMaxValue32	light_times	“Light Times” information

4-2-15. GetAEState

Table 62. GetAEState definition

Description	• Used to get the current light emission count (Light times) information. • Fails if the camera device is not open.		
Request structure	Type	Name	Remark
	uint8	reserved	reserved
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure
	bool	enable	Status of AE function true : Enable false : Disable

4-2-16. GetAEInterval

Table 63. GetAEInterval definition

Description	• Used to get the current AE control interval information. • Fails if the camera device is not open.		
Request structure	Type	Name	Remark
	uint8	reserved	reserved
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure
	MinMaxValue16	interval	AE control interval information

4-2-17. GetRawSatThreshol

Table 64. GetRawSatThreshold definition

Description	• Used to get the current saturation threshold level (to RAW data) information. • Fails if the camera device is not open.		
Request structure	Type	Name	Remark
	uint8	reserved	reserved
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure
	MinMaxValue16	raw_sat_th	RAW saturation threshold level

4-2-18. GetIrDarkThreshold

Table 65. GetIrDarkThreshold definition

Description	• Used to get the current IR invalid threshold level information. • Fails if the camera device is not open.		
Request structure	Type	Name	Remark
	uint8	reserved	reserved
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure
	MinMaxValue16	ir_dark_th	IR invalid threshold level

4-2-19. GetIntSuppInfo

Table 66. GetIntSuppInfo definition

Description	• Used to get the current interference suppression function information. • Fails if the camera device is not open.		
Request structure	Type	Name	Remark
	uint8	reserved	reserved
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure
	IntSuppInfo	int_supp_info	Interference suppression information

4-2-20. SetExtTriggerType

Table 67. SetExtTriggerType definition

Description	• Used when setting external trigger type information. • Fails if the camera device is not open. • Fails if set when the current external trigger type is Slave (EXT_TRG_SLAVE).		
Request structure	Type	Name	Remark
	ExtTriggerType	type	External trigger type
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure

4-2-21. SetExtTriggerOffset

Table 68. SetExtTriggerOffset definition

Description	• Used when setting external trigger signal offset (Pulse width) information. • Fails if the camera device is not open.		
Request structure	Type	Name	Remark
	uint8	offset	External trigger signal offset
Response structure	Type	Name	Remark
	bool	result	true : Success

			false : Failure

4-2-22. SetMode

Table 69. SetMode definition

Description	• Used when setting the operation mode. • Specify the operation mode ID obtained with GetModeList. Fails if the camera device is not open. • Fails if set while image output is being performed from the camera device.		
Request structure	**Type**	**Name**	**Remark**
	uint8	mode	Operation mode
Response structure	**Type**	**Name**	**Remark**
	bool	result	true : Success false : Failure

4-2-23. SetImgKinds

Table 70. SetImgKinds definition

Description	• Used when setting the output image type. • Specify the output image type obtained with GetModeList. • Fails if the camera device is not open. • Fails if set while image output is being performed from the camera device.		
Request structure	**Type**	**Name**	**Remark**
	ImgOutKind	img_out	Output image type
Response structure	**Type**	**Name**	**Remark**
	bool	result	true : Success false : Failure

4-2-24. SetLightTimes

Table 71. SetLightTimes definition

Description	• Used when setting the number of light emissions (Light Times). • Set the number of light emissions within the minimum to maximum values obtained with GetLightTimes. • Fails if the camera device is not open. • If the operation mode is switched with SetMode, the number of light emissions is reset to the initial value for each operation mode.		
Request structure	**Type**	**Name**	**Remark**
	uint32	count	Light times
Response structure	**Type**	**Name**	**Remark**
	bool	result	true : Success false : Failure

4-2-25. SetAEState

Table 72. SetAEState definition

Description	• Used when setting the AE function state.

	• If the minimum and maximum values obtained with GetLightTimes are the same, the AE function is always disabled. • Fails if the camera device is not open. • Fails if set while image output is being performed from the camera device.		
Request structure	**Type**	**Name**	**Remark**
	bool	enable	Status of AE function true : Enable false : Disable
Response structure	**Type**	**Name**	**Remark**
	bool	result	true : Success false : Failure

4-2-26. SetAEInterval

Table 73. SetAEInterval definition

Description	• Used when setting the control interval of the AE function. • Fails if the camera device is not open. • Fails if set while image output is being performed from the camera device. • If the operation mode is switched with SetMode, the AE control interval before the operation mode switch is carried over.		
Request structure	**Type**	**Name**	**Remark**
	uint8	interval	AE control interval information
Response structure	**Type**	**Name**	**Remark**
	bool	result	true : Success false : Failure

4-2-27. SetRawSatThreshold

Table 74. SetRawSatThreshold definition

Description	• Used when setting the RAW saturation threshold level. • Set the threshold within the minimum to maximum values obtained with GetRawSatThreshold. • Fails if the camera device is not open. • If the operation mode is switched with SetMode, the threshold is reset to the initial value for each operation mode.		
Request structure	**Type**	**Name**	**Remark**
	uint16	raw_threshold	RAW saturation threshold level
Response structure	**Type**	**Name**	**Remark**
	bool	result	true : Success false : Failure

4-2-28. SetIrDarkThreshold

Table 75. SetIRDarkThreshold definition

Description	• Used when setting the IR invalid threshold.

	• Set the threshold within the minimum to maximum values obtained with GetIrDarkThreshold. • Fails if the camera device is not open. • If the operation mode is switched with SetMode, the threshold is reset to the initial value for each operation mode.		
Request structure	Type	Name	Remark
	uint16	ir_threshold	IR invalid threshold level
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure

4-2-29. SetIntSupp

Table 76. SetIntSupp definition

Description	• Used when setting the interference prevention function. • Set the values of each parameter within the minimum to maximum values obtained with GetIntSuppInfo. • Fails if the camera device is not open.		
Request structure	Type	Name	Remark
	IntSuppModeType	int_supp_mode	Interference suppression mode
	uint8	int_supp_prm_m	Manual mode parameter
	uint8	int_supp_prm_a1	Auto mode param1 @test mode
	uint8	int_supp_prm_a2	Auto mode param2 @test mode
	uint8	int_supp_prm_a3	Auto mode param3 @test mode
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure

4-2-30. TofCtrl

Table 77. TofCtrl definition

Description	• Used for controlling image output from the camera device. • Fails if the camera device is not open.				
Request structure	Type	Name	Remark		
	uint8	cmd	Control commands		
			CMD_START	0	Start output
			CMD_STOP	1	Stop output
Response structure	Type	Name	Remark		
	bool	result	true : Success false : Failure		

4-2-31. GetPlayTarget

Table 78. GetPlayTarget definition

Description	• Used when getting file playback target directory information.

	• Can only be used when file playback is selected as the camera device. • Fails if the camera device is not open.		
Request structure	**Type**	**Name**	**Remark**
	uint8	reserved	reserved
Response structure	**Type**	**Name**	**Remark**
	bool	result	true : Success false : Failure
	string	directory	File playback target directory

4-2-32. GetPlayTime

Table 79. GetPlayTime definition

Description	• Used when getting the current playback time of file playback. • Can only be used when file playback is selected as the camera device. • Fails if the camera device is not open.		
Request structure	**Type**	**Name**	**Remark**
	uint8	reserved	reserved
Response structure	**Type**	**Name**	**Remark**
	bool	result	true : Success false : Failure
	PlayTime	play_time	Current playback time information

4-2-33. GetPlayStatus

Table 80. GetPlayStatus definition

Description	• Used when getting the file playback status. • Can only be used when file playback is selected as the camera device. • Fails if the camera device is not open. playing_fps indicates the frame rate during playback. • In fast-forward or slow-motion playback, this is the frame rate after fast-forward or slow-motion relative to the original frame rate. (Example: If the original is 30fps data, it becomes 60fps for 2x fast-forward playback and 15fps for ½x slow-motion playback).				
Request structure	**Type**	**Name**	**Remark**		
	uint8	reserved	reserved		
Response structure	**Type**	**Name**	**Remark**		
	bool	result	true : Success false : Failure		
	uint8	state	File Playback States		
			STOPPED	0	Stopped state
			PLAYING	1	Playing state (normal speed playback)
			PAUSE	2	Paused state
			FAST	3	Fast-forward playback state (high-speed playback)
			SLOW	4	Slow-motion playback state (low-speed playback)

	uint16	playing_fps	Current frame rate [fps × 100]

4-2-34. SetPlayTarget

Table 81. SetPlayTarget definition

Description	• Used when setting the file playback target directory. • Can only be used when file playback is selected as the camera device. • Fails if the camera device is not open. • Fails if set while image output is being performed from the camera device. Fails if the specified directory or file does not exist, or if the file is outside the target version.		
Request structure	**Type**	**Name**	**Remark**
	string	directory	File playback target directory
Response structure	**Type**	**Name**	**Remark**
	bool	result	true : Success false : Failure

4-2-35. SetPlayCtrl

Table 82. SetPlayCtrl definition

Description	• Used to control the file playback function. • Can only be used when file playback is selected as the camera device. • Set the number of frames when specifying the following control commands: ▪ Playback time setting: Set the current playback time to the specified frame (within the total playback time range obtained with GetPlayTime). ▪ Fast-forward control: Fast-forward by the specified number of frames. ▪ Rewind control: Rewind by the specified number of frames. • If pause control is set while in a paused state, it returns to the playback state. The following control commands fail in a paused state: ▪ Fast-forward playback control ▪ Slow-motion playback control • The control command fails in the following cases: ▪ The camera device is not open. ▪ The open camera type is other than file playback (PLAYBACK). ▪ Image output is stopped. ▪ In playback time setting, if a time exceeding the total playback time is set. ▪ In fast-forward playback control, if it exceeds 4x speed or the received frame rate exceeds 120fps. ▪ In slow-motion playback control, if it is less than 1/4x speed or the received frame rate falls below 10fps.				
Request structure	**Type**	**Name**	**Remark**		
	uint8	cmd	Control commands		
			CMD_PLAY_TIME	0	Set playback time
			CMD_PAUSE	1	Pause playback
			CMD_FAST_PLAY	2	Fast-forward playback
			CMD_SLOW_PLAY	3	Slow-motion playback
			CMD_JUMP_FW	4	Jump forward

			CMD_JUMP_BW	5	Jump backward
	uint32	time	Frame Count • The frame count is valid only when using the CMD_PLAY_TIME, CMD_JUMP_FW, or CMD_JUMP_BW commands.		
Response structure	**Type**	**Name**	**Remark**		
	bool	result	true : Success false : Failure		

4-2-36. PsblPostFilt

Table 83. PsblPostFilt definition

Description	• Used when getting the usability information of each PostFilter function.				
Request structure	**Type**	**Name**	**Remark**		
	uint8	filt_type	Filter type		
			POST_FILT_MEDF	0	Median Filter
			POST_FILT_BILF	1	Bilateral Filter
			POST_FILT_FLYF	2	Flying Pixel Filter
Response structure	**Type**	**Name**	**Remark**		
	bool	result	true : Success false : Failure		
	bool	possible	Availability information true : Available false : Unavailable		

4-2-37. SetPostFilt

Table 84. SetPostFilt definition

Description	• Used when setting the usage of each PostFilter function. • The initial state follows the contents of "postfilter_param.yaml".				
Request structure	**Type**	**Name**	**Remark**		
	uint8	filt_type	Filter type		
			POST_FILT_MEDF	0	Median Filter
			POST_FILT_BILF	1	Bilateral Filter
			POST_FILT_FLYF	2	Flying Pixel Filter
	bool	enable	true : Enable false : Disable		
Response structure	**Type**	**Name**	**Remark**		
	bool	result	true : Success false : Failure		

4-2-38. SetPostFiltPrm

Table 85. SetPostFilterPrm definition

Description	• Used when setting PostFilter parameters. • The initial state follows the contents of "postfilter_param.yaml".

Request structure	Type	Name	Remark
	PostFilterPrm	param	PostFilter parameters
Response structure	Type	Name	Remark
	bool	result	true : Success false : Failure

4-2-39. PsblLensConv

Table 86. PsblLensConv definition

Description	• Used when getting the usability information of each LensConv function.				
Request structure	Type	Name	Remark		
	uint8	conv_type	Conversion type		
			LENS_CONV_DIST	0	Distortion correction
Response structure	Type	Name	Remark		
	bool	result	true : Success false : Failure		
	bool	possible	Availability information true : Available false : Unavailable		

4-2-40. SetLensConv

Table 87. SetLensConv definition

Description	• Used when setting the usage of each LensConv function. • The initial state follows the contents of “lensconv_param.yaml”.				
Request structure	Type	Name	Remark		
	uint8	conv_type	Conversion type		
			LENS_CONV_DIST	0	Distortion correction
			LENS_PCD_KIND	1	Point cloud transformation
	bool	enable	When conv_type is LENS_CONV_DIST	true : Enable false : Disable	
			When conv_type is LENS_PCD_KIND	PCD Transformation Method true: World Coordinate false: Camera Coordinate	
Response structure	Type	Name	Remark		
	bool	result	true : Success false : Failure		

4-2-41. SetPcdPos

Table 88. SetPcdPos definition

Description	• Used when setting the origin position and rotation information for LensConv world coordinate transformation. • The initial state follows the contents of “lensconv_param.yaml”.

Request structure	**Type**	**Name**	**Remark**
	PosOrgRotation	pos	Origin position and rotation information for world coordinate transformation
Response structure	**Type**	**Name**	**Remark**
	bool	result	true : Success false : Failure

4-2-42. SetPcdColor

Table 89. SetPcdColor definition

Description	• Used when setting color information after point cloud transformation. • The initial state is PCD_COLOR_NONE.				
Request structure	**Type**	**Name**	**Remark**		
	uint8	color	Color information		
			PCD_COLOR_NONE	0	Exclude color information
			PCD_COLOR_IR	1	Use IR data as color information
Response structure	**Type**	**Name**	**Remark**		
	bool	result	true : Success false : Failure		

4-2-43. RecordCtrl

Table 90. RecordCtrl definition

Description	• Used to control file recording of output images. • Only directory, save_frames, packing_frames, is_crct_dist, is_filt_med, is_filt_bil, and is_filt_fly_p are valid when CMD_REC_START is set for cmd. • The following cases fail when cmd is CMD_REC_START: • If either save_frames or packing_frames is 0. • If the remaining capacity after saving the number of recorded frames is predicted to fall below 1GB.				
Request structure	**Type**	**Name**	**Remark**		
	uint8	cmd	File recording control commands		
			CMD_REC_START	0	Start recording
			CMD_REC_STOP	1	Stop recording
	string	directory	Target recording directory path Do not specify paths containing multi-byte characters		
	uint32	save_frames	Number of frames to record		
	uint16	packing_frames	Number of frames to include in one file		
	bool	is_crct_dist	Whether distortion correction has been applied		
	bool	is_filt_med	Whether the median filter has been applied		
	bool	is_filt_bil	Whether the bilateral filter has been applied		
	bool	is_filt_fly_p	Whether the flying pixel filter has been applied		
Response structure	**Type**	**Name**	**Remark**		
	bool	result	true : Success false : Failure		

5. Static parameters

The “tof_camera_param” in this package contains a YAML file (.yaml) that is referenced as static parameters and is stored in the following location when the build is executed.

```
{ROS2 path}/install/tof_camera_param/share/tof_camera_param/param
```

The contents of the YAML file used by each Node as a static parameter are described below.

5-1. tof_param.yaml

TOF Control Node reads “tof _param.yaml” as a static parameter at startup sequence. The contents of “tof_param.yaml” are as follows.

Table 91. tof_param.yaml parameters

Parameter	Range	Default	Remark
cam_type	0~1	0	Camera type 0 : Use camera device 1 : Use PlayBack function
cam_index	0～	0	Index of the connected camera list (Valid only when cam_type is a camera device)
cam_mode	0～	0	Operation Mode ID (Valid only when cam_type is a camera device)
play_dir	-	-	Directory to be played when using PlayBack (Valid only when cam_type is PlayBack) *In Windows OS, specify a path separator as "¥¥" instead of "¥."
start	true/false	true	Image output status after starting Node true : Starts image output. false : Do not start image output.

5-2. postfilter_param.yaml

The PostFilter Node reads “postfilter _param.yaml” at startup as a static parameter. The contents of “postfilter_param.yaml” are as follows.

Table 92. postfilter_param.yaml parameters

Parameter	Range	Default	Description
median	true/false	true	Execution setting of median filter true : Executed false : Not executed
bilateral	true/false	true	Execution setting of bilateral filter true : Executed false : Not executed
flyp	true/false	true	Execution setting of the flying pixel filter true : Yes false : No

median_ksize	3 or 5	3	Median filter size
bilateral_ksize	3 or 5	3	Bilateral filter size
bilateral_sigma_depth	0.001 ... 1000.000	500.0	Depth value smoothing parameter for bilateral filter
bilateral_sigma_ir	0.001 ... 250.000	100.0	IR value smoothing parameter for bilateral filter
bilateral_sigma_color	0.001 ... 10.000	1.0	Spatial smoothing parameter for bilateral filter
flyp_ksize	3 or 5	3	Flying pixel filter size
flyp_log	true/false	true	Flying pixel filter judgment method true : Ratio false : Differential value (For debug use)
flyp_thr	uint16	130	Flying pixel judgment threshold for Flying pixel filter
flyp_fast_proc	true/false	true	Flying pixel filter processing method true : Speed priority false : Accuracy priority

5-3. lensconv_param.yaml

LensConv Node reads “lensconv _ param.yaml” as a static parameter at startup sequence. The contents of “lensconv_param.yaml” are as follows.

Table 93. lensconv_param.yaml parameters

Parameter	Range	Default	Remark
distortion	true/false	true	Setting for distortion correction on Depth and IR images true : Yes false : No
pcd_origin	true/false	false	Conversion method for output Point Cloud data true : Convert to world coordinate origin false : Convert to camera coordinate origin
pcd_offset_x	int16	0	Origin position: Offset (X axis direction) [mm] [Note]
pcd_offset_y	int16	0	Origin position: Offset (Y axis direction) [mm] [Note]
pcd_offset_z	int16	0	Origin position: Offset (Z axis direction) [mm] [Note]
pcd_rotation_x	float32	0.0	Pitch rotation angle [degree] [Note]
pcd_rotation_y	float32	0.0	Yaw rotation angle [degree] [Note]
pcd_rotation_z	float32	0.0	Roll rotation angle [degree] [Note]

Note: Initial settings for the world coordinate origin and camera rotation angles used in point cloud transformation (world coordinates). These settings are valid only when pcd_origin is set to true and can be reconfigured through the krm/tof_lens_conv service.

6. Operation sequence

The operation sequence of this package is as follows.

6-1. Initialization sequence

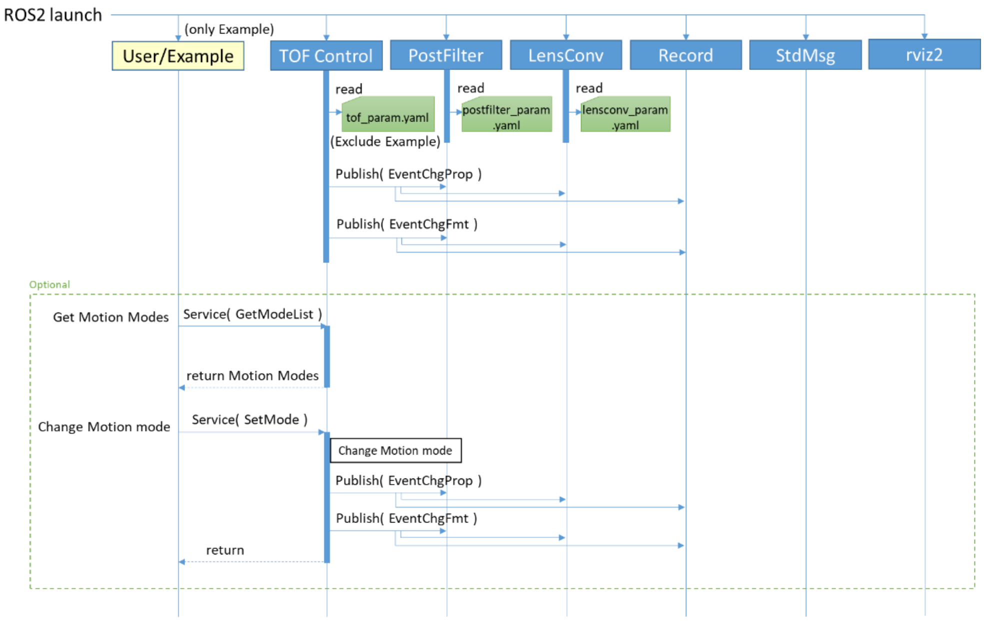


Figure 5. Initialization sequence

6-2. Image receiving sequence

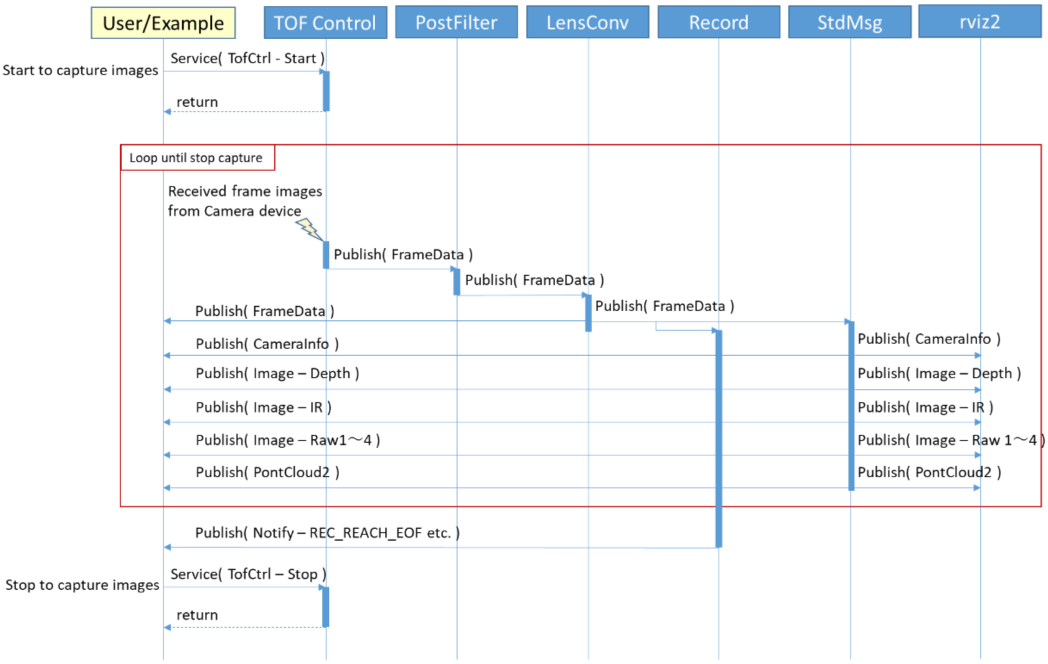


Figure 6. Image receiving sequence

7. Execution

7-1. Launch files

This package uses the Launch file to start each Node. The following Launch files are provided according to the Node configuration.

Table 94. Launch files

Node configuration	Launch file name
TOF Control + PostFilter + LensConv	launch_alone.py
TOF Control + PostFilter + LensConv + Record + Sample Viewer	launch_viewer.py
TOF Control + PostFilter + LensConv + StdMsg + rviz2	launch_rviz2.py
TOF Control + PostFilter + LensConv + Record + StdMsg + rviz2	launch_rviz2_record.py

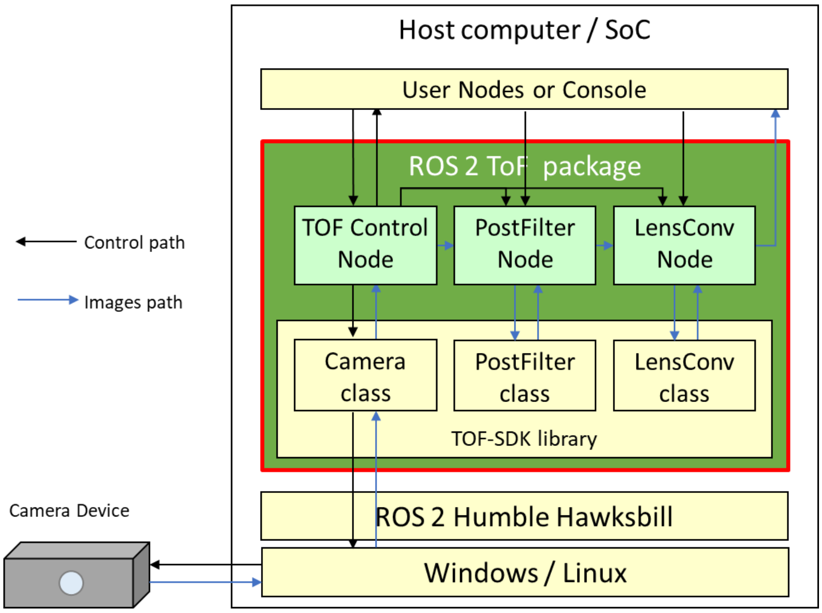


Figure 7. Node structure (launch_alone.py)

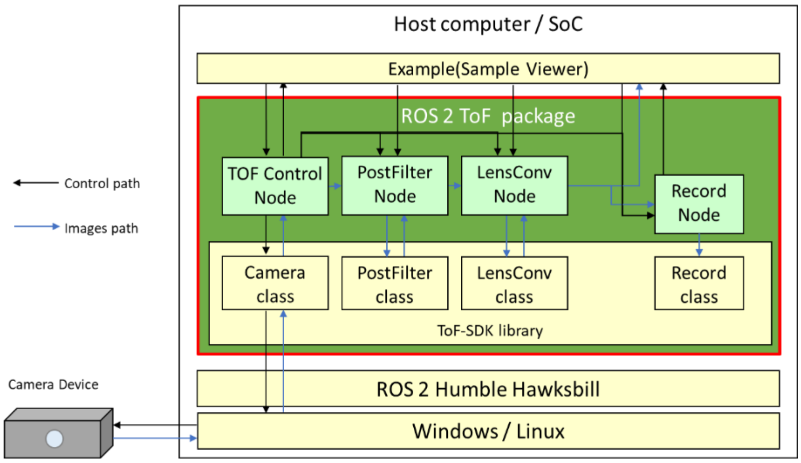


Figure 8. Node structure (launch_viewer.py)

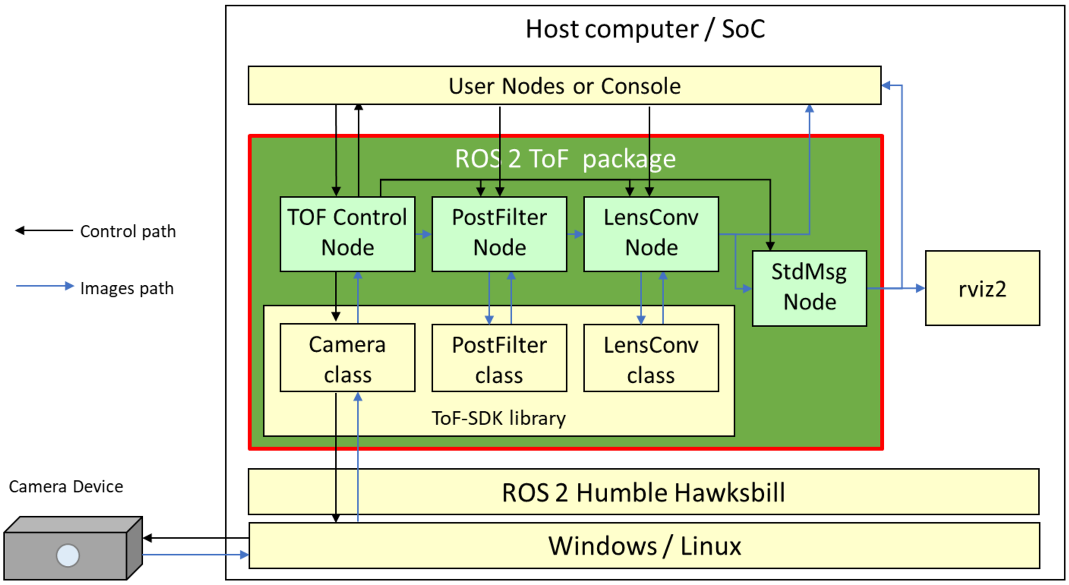


Figure 9. Node structure (launch_rviz2.py)

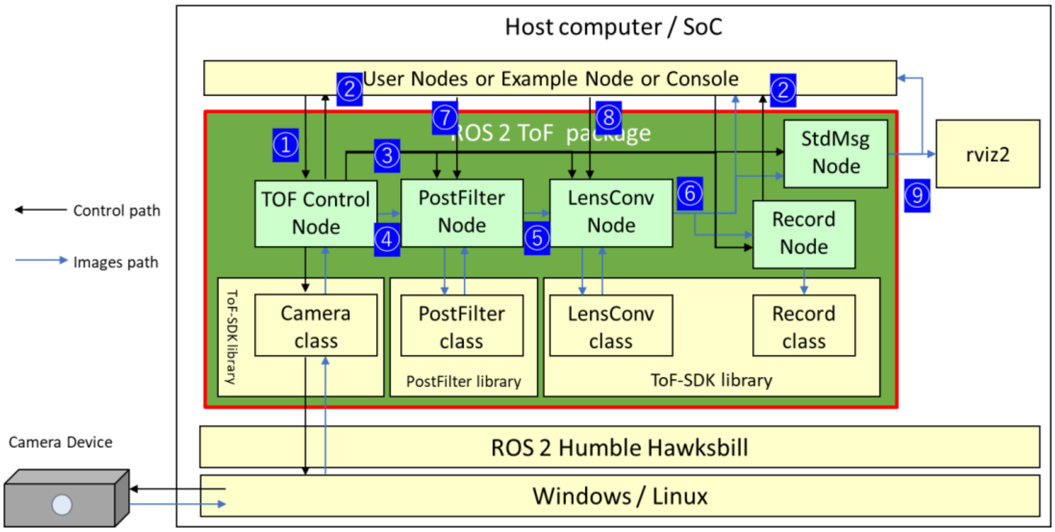


Figure 10. Node structure (launch_rviz2_record.py)

7-2. About domain IDs

ROS uses domain IDs to distinguish communication on the same network. To distinguish communication with different parties on the same network, it is necessary to switch domain IDs. In all script files provided by this package, the domain ID is set to 10. To switch domain IDs, change the value of ROS_DOMAIN_ID in each script file.

Table 95. Target files for changing domain ID

Environment	Target file
Windows	{ROS2 path}¥scripts¥windows¥run_launch.bat
	{ROS2 path}¥scripts¥windows¥env.bat
Linux/Ubuntu	{ROS2 path}/scripts/ubuntu/env.sh

7-3. Execution for Windows

This section describes how to execute this package and how to control it from the command prompt. For subsequent execution methods, start the command prompt and execute from the command prompt.

7-3-1. Start Node

Use the Launch file to start each Node. Execute the following command. Replace {Launch file} with the name of the Launch file according to the configuration to be executed.

```
cd {ROS2 path}
call .¥scripts¥windows¥run_launch.bat {Launch file}
```

7-3-2. Control Using CUI (launch_alone.py/launch_rviz2.py/launch_rviz2_record.py)

When using "launch_alone.py/launch_rviz2.py/launch_rviz2_record.py", control is performed on the command prompt. When using launch_viewer.py, perform operations on the GUI.

```
cd {ROS2 path}
call .¥scripts¥windows¥run_launch.bat {Launch file}
```

7-3-2-1. Environment configuration

If you want to control from a command prompt different from the one where you launched the Node, execute the following beforehand:

```
cd {ROS2 path}
call .¥scripts¥windows¥env.bat
```

7-3-2-2. Obtaining Various Information

Execute the following commands according to the information to be obtained:

```
cd {ROS2 path}
(Command according to the content to be obtained)
```

For the parameter specification content and obtainable content described in {} in the table, refer to the contents of each Service type definition. For parameter specification, specify numbers as decimal numbers, bool type as 0 (false)/1 (true), and string type as strings (ASCII characters only).

Table 96. Get command list (Windows)

Get information	Command
Camera Device List (**GetDevList**)	call .¥scripts¥windows¥get_dev_list.bat {type.type}
Device information (**GetDevInfo**)	call .¥scripts¥windows¥get_dev_info.bat
FoV Information (**GetFov**)	call .¥scripts¥windows¥get_fov.bat
External Trigger Type Information (**GetExtTriggerType**)	call .¥scripts¥windows¥get_ext_trigger_type.bat
External Trigger Signal Offset Info. (**GetExtTriggerOffset**)	call .¥scripts¥windows¥get_ext_trigger_offset.bat
Operation Mode List Information (**GetModeList**)	call .¥scripts¥windows¥get_mode_list.bat

Operation Mode (**GetMode**)	call .¥scripts¥windows¥get_mode.bat
Output Image Kinds (**GetImgKinds**)	call .¥scripts¥windows¥get_img_kinds.bat
Image Format (**GetImgFormat**)	call .¥scripts¥windows¥get_img_format.bat
PostFilter Processing Information (**GetPostFiltInfo**)	call .¥scripts¥windows¥get_post_filt_info.bat
Lens Conversion Processing Parameters (**GetLensInfo**)	call .¥scripts¥windows¥get_lens_info.bat
Number of "Light times" (**GetLightTimes**)	call .¥scripts¥windows¥get_light_times.bat
AE Function State (**GetAEState**)	call .¥scripts¥windows¥get_ae_state.bat
AE Function Control Interval (**GetAEInterval**)	call .¥scripts¥windows¥get_ae_interval.bat
RAW Saturation Threshold (**GetRawSatThreshol**)	call .¥scripts¥windows¥get_raw_sat_threshold.bat
IR Invalid Threshold (**GetIrDarkThreshold**)	call .¥scripts¥windows¥get_ir_dark_threshold.bat
Interference Prevention Function Info. (**GetIntSuppInfo**)	call .¥scripts¥windows¥get_int_supp_info.bat
File Playback Target Directory (**GetPlayTarget**)	call .¥scripts¥windows¥get_play_target.bat
Current Playback Time of File Playback (**GetPlayTime**)	call .¥scripts¥windows¥get_play_time.bat
File Playback Status (**GetPlayStatus**)	call .¥scripts¥windows¥get_play_status.bat
Each PostFilter Function state (**PsblPostFilt**)	call .¥scripts¥windows¥psbl_post_filt.bat {filt_type}
Each LensConv Function state (**PsblLensConv**)	call .¥scripts¥windows¥psbl_lens_conv.bat {conv_type}

7-3-2-3. Changing and Controlling Various Settings

Execute the following commands according to the settings to be changed or the control content:

```
cd {ROS2 path}
(Command according to the settings to be changed or the control content)
```

For the parameter specification content and obtainable content described in {} in the table, refer to the contents of each Service type definition. For parameter specification, specify numbers as decimal numbers, bool type as 0 (false)/1 (true), and string type as strings (ASCII characters only).

Table 97. Setting and Controlling command list (Windows)

Setting/Control information	Command
Camera Device Open Processing **(OpenDev)**	call .¥scripts¥windows¥open_dev.bat {type.type} {dev_id}
Camera Device Close Processing **(CloseDev)**	call .¥scripts¥windows¥close_dev.bat
External Trigger Type **(SetExtTriggerType)**	call .¥scripts¥windows¥set_ext_trigger_type.bat {type.ext_trigger_type}
External Trigger Signal Offset **(SetExtTriggerOffset)**	call .¥scripts¥windows¥set_ext_trigger_offset.bat {offset}
Operation Mode **(SetMode)**	call .¥scripts¥windows¥set_mode.bat {mode}
Output Image Kinds **(SetImgKinds)**	call .¥scripts¥windows¥set_img_kinds.bat {img_out.img_out_kind }

Number of "Light **times**" **(SetLightTimes)**	call .¥scripts¥windows¥set_light_times.bat {count}
AE Function State **(SetAEState)**	call .¥scripts¥windows¥set_ae_state.bat {enable}
AE Function Control Interval **(SetAEInterval)**	call .¥scripts¥windows¥set_ae_interval.bat {interval}
RAW Saturation Threshold **(SetRawSatThreshold)**	call .¥scripts¥windows¥set_raw_sat_threshold.bat {raw_threshold}
IR Invalid Threshold **(SetIrDarkThreshold)**	call .¥scripts¥windows¥set_ir_dark_threshold.bat {ir_threshold}
Interference Prevention Function Info. **(SetIntSupp)**	call .¥scripts¥windows¥set_int_supp_info.bat {int_supp_mode. int_supp_mode_type} {int_supp_prm_m} {int_supp_prm_a1} {int_supp_prm_a2} {int_supp_prm_a3}
File Playback Target Directory **(SetPlayTarget)**	call .¥scripts¥windows¥set_play_target.bat {directory}
File Playback Function Control **(SetPlayCtrl)**	call .¥scripts¥windows¥set_play_ctrl.bat {cmd} {time}
Setting of Each PostFilter Function **(SetPostFilt)**	call .¥scripts¥windows¥set_post_filt.bat {filt_type} {enable}
PostFilter Parameters **(SetPostFiltPrm)**	call .¥scripts¥windows¥set_post_filt_prm.bat {param.median_ksize} {param.bil_ksize} {param.bil_sigma_depth} {param.bil_sigma_ir} {param.bil_sigma_space}{param.flyp_ksize} {param.flyp_log} {param.flyp_thr}{param.flyp_fast_proc}
Setting of Each LensConv Function **(SetLensConv)**	call .¥scripts¥windows¥set_lens_conv.bat {conv_type} {enable}
Origin Position and Rotation Information of LensConv World Coordinate Transformation **(SetPcdPos)**	call .¥scripts¥windows¥set_pcd_pos.bat {pos.offset_x} {pos.offset_y} {pos.offset_z} {pos.rotation_pitch} {pos.rotation_yaw} {pos.rotation_roll}
Color Information After Point Cloud Transformation **(SetPcdColor)**	call .¥scripts¥windows¥set_pcd_color.bat {color}

* Do not specify a directory containing multi-byte characters.

7-3-2-4. Start image output (capture)

Use Service of TofCtrl to start image output. Execute the following command in a command prompt different from the command prompt used to start Node.

```
cd {ROS2 path}
call .¥scripts¥windows¥start_capture.bat
```

7-3-2-5. Stop image output (capture)

Use the Service of TofCtrl to stop image output. Execute the following command in a command prompt that is different from the command prompt from which Node was started.

```
cd {ROS2 path}
call .¥scripts¥windows¥stop_capture.bat
```

7-3-2-6. Start recording image output

Start saving output images using the RecordCtrl Service. Execute the following command. For parameter specification, specify numbers as decimal numbers, bool type as 0 (false)/1 (true), and string type as strings (multi-byte characters are not allowed).

```
cd {ROS2 path}
```

```
call .¥scripts¥windows¥start_record.bat {directory} {save_frames} {packing_frames} {is_crct_dist}
{is_filt_med} {is_filt_bil} {is_filt_fly_p}
```

7-3-2-7. Stop recording image output

Stop recording output images using the RecordCtrl Service. Execute the following command.

```
cd {ROS2 path}
call .¥scripts¥windows¥stop_record.bat
```

7-3-3. Exit

Exit with the "Ctrl + C" key on the terminal used to start Node using the Launch file.

7-4. Execution for Linux/Ubuntu

This section describes how to execute this package and how to control it from the terminal. For the execution methods described below, launch a terminal and execute them on the terminal.

7-4-1. Start Node

Use the Launch file to start each Node. Execute the following command. Replace {Launch-file} with the name of the Launch file according to the configuration to be executed.

```
cd {ROS2 path}
./scripts/ubuntu/run_launch.sh {Launch file}
```

7-4-2. Control Using terminal (launch_alone.py/launch_rviz2.py/launch_rviz2_record.py)

When using "launch_alone.py/launch_rviz2.py/launch_rviz2_record.py", control is performed on the terminal. When using launch_viewer.py, perform operations on the GUI.

7-4-2-1. Obtaining Various Information

Execute the following commands according to the information to be obtained:

```
cd {ROS2 path}
(Command according to the content to be obtained)
```

For the parameter specification content and obtainable content described in {} in the table, refer to the contents of each Service type definition. For parameter specification, specify numbers as decimal numbers, bool type as 0 (false)/1 (true), and string type as strings (ASCII characters only).

Table 98. Get command list (Linux/Ubuntu)

Get information	Command
Camera Device List (**GetDevList**)	./scripts/ubuntu/get_dev_list.sh {type.type}
Device information (**GetDevInfo**)	./scripts/ubuntu/get_dev_info.sh
FoV Information (**GetFov**)	./scripts/ubuntu/get_fov.sh

External Trigger Type Information (**GetExtTriggerType**)	./scripts/ubuntu/get_ext_trigger_type.sh
External Trigger Signal Offset Info. (**GetExtTriggerOffset**)	./scripts/ubuntu/get_ext_trigger_offset.sh
Operation Mode List Information (**GetModeList**)	./scripts/ubuntu/get_mode_list.sh
Operation Mode (**GetMode**)	./scripts/ubuntu/get_mode.sh
Output Image Kinds (**GetImgKinds**)	./scripts/ubuntu/get_img_kinds.sh
Image Format (**GetImgFormat**)	./scripts/ubuntu/get_img_format.sh
PostFilter Processing Information (**GetPostFiltInfo**)	./scripts/ubuntu/get_post_filt_info.sh
Lens Conversion Processing Parameters (**GetLensInfo**)	./scripts/ubuntu/get_lens_info.sh
Number of “Light times” (**GetLightTimes**)	./scripts/ubuntu/get_light_times.sh
AE Function State (**GetAEState**)	./scripts/ubuntu/get_ae_state.sh
AE Function Control Interval (**GetAEInterval**)	./scripts/ubuntu/get_ae_interval.sh
RAW Saturation Threshold (**GetRawSatThreshol**)	./scripts/ubuntu/get_raw_sat_threshold.sh
IR Invalid Threshold (**GetIrDarkThreshold**)	./scripts/ubuntu/get_ir_dark_threshold.sh
Interference Prevention Function Info. (**GetIntSuppInfo**)	./scripts/ubuntu/get_int_supp_info.sh
File Playback Target Directory (**GetPlayTarget**)	./scripts/ubuntu/get_play_target.sh
Current Playback Time of File Playback (**GetPlayTime**)	./scripts/ubuntu/get_play_time.sh
File Playback Status (**GetPlayStatus**)	./scripts/ubuntu/get_play_status.sh
Each PostFilter Function state (**PsblPostFilt**)	./scripts/ubuntu/psbl_post_filt.sh {filt_type}
Each LensConv Function state (**PsblLensConv**)	./scripts/ubuntu/psbl_lens_conv.sh {conv_type}

7-4-2-2. Changing and Controlling Various Settings

Execute the following commands according to the settings to be changed or the control content:

```
cd {ROS2 path}
(Command according to the settings to be changed or the control content)
```

For the parameter specification content and obtainable content described in {} in the table, refer to the contents of each Service type definition. For parameter specification, specify numbers as decimal numbers, bool type as 0 (false)/1 (true), and string type as strings (ASCII characters only).

Table 99. Setting and Controlling command list (Linux/Ubuntu)

Setting/Control information	Command
Camera Device Open Processing (**OpenDev**)	./scripts/ubuntu/open_dev.sh {type.type} {dev_id}
Camera Device Close Processing (**CloseDev**)	./scripts/ubuntu/close_dev.sh

External Trigger Type (**SetExtTriggerType**)	./scripts/ubuntu/set_ext_trigger_type.sh {type.ext_trigger_type}
External Trigger Signal Offset (**SetExtTriggerOffset**)	./scripts/ubuntu/set_ext_trigger_offset.sh {offset}
Operation Mode (**SetMode**)	./scripts/ubuntu/set_mode.sh {mode}
Output Image Kinds (**SetImgKinds**)	./scripts/ubuntu/set_img_kinds.sh {img_out.img_out_kind}
Number of “Light times” (**SetLightTimes**)	./scripts/ubuntu/set_light_times.sh {count}
AE Function State (**SetAEState**)	./scripts/ubuntu/set_ae_state.sh {enable}
AE Function Control Interval (**SetAEInterval**)	./scripts/ubuntu/set_ae_interval.sh {interval}
RAW Saturation Threshold (**SetRawSatThreshold**)	./scripts/ubuntu/set_raw_sat_threshold.sh {raw_threshold}
IR Invalid Threshold (**SetIrDarkThreshold**)	./scripts/ubuntu/set_ir_dark_threshold.sh {ir_threshold}
Interference Prevention Function Info. (**SetIntSupp**)	./scripts/ubuntu/set_int_supp_info.sh {int_supp_mode. int_supp_mode_type} {int_supp_prm_m} {int_supp_prm_a1} {int_supp_prm_a2} {int_supp_prm_a3}
File Playback Target Directory (**SetPlayTarget**)	./scripts/ubuntu/set_play_target.sh {directory}
File Playback Function Control (**SetPlayCtrl**)	./scripts/ubuntu/set_play_ctrl.sh {cmd} {time}
Setting of Each PostFilter Function (**SetPostFilt**)	./scripts/ubuntu/set_post_filt.sh {filt_type} {enable}
PostFilter Parameters (**SetPostFiltPrm**)	./scripts/ubuntu/set_post_filt_prm.sh {param.median_ksize} {param.bil_ksize} {param.bil_sigma_depth} {param.bil_sigma_ir} {param.bil_sigma_space}{param.flyp_ksize} {param.flyp_log} {param.flyp_thr}{param.flyp_fast_proc}
Setting of Each LensConv Function (**SetLensConv**)	./scripts/ubuntu/set_lens_conv.sh {conv_type} {enable}
Origin Position and Rotation Information of LensConv World Coordinate Transformation (**SetPcdPos**)	./scripts/ubuntu/set_pcd_pos.sh {pos.offset_x} {pos.offset_y} {pos.offset_z} {pos.rotation_pitch} {pos.rotation_yaw} {pos.rotation_roll}
Color Information After Point Cloud Transformation (**SetPcdColor**)	./scripts/ubuntu/set_pcd_color.sh {color}

* Do not specify a directory containing multi-byte characters.

7-4-2-3. Start image output (capture)

Start image output using the TofCtrl Service. Execute the following command.

```
cd {ROS2 path}
./scripts/ubuntu/start_capture.sh
```

7-4-2-4. Stop image output (capture)

Stop image output using the TofCtrl Service. Execute the following command.

```
cd {ROS2 path}
./scripts/ubuntu/stop_capture.sh
```

7-4-2-5. Start recording image output

Start recording output images using the RecordCtrl Service. Execute the following command. For parameter specification, specify numbers as decimal numbers, bool type as 0 (false)/1 (true), and string type as strings (multi-byte characters are not allowed).

```
cd {ROS2 path}
./scripts/ubuntu/start_record.sh {directory} {save_frames} {packing_frames} {is_crct_dist} {is_filt_med}
{is_filt_bil} {is_filt_fly_p}
```

7-4-2-6. Stop recording image output

Stop recording output images using the RecordCtrl Service. Execute the following command.

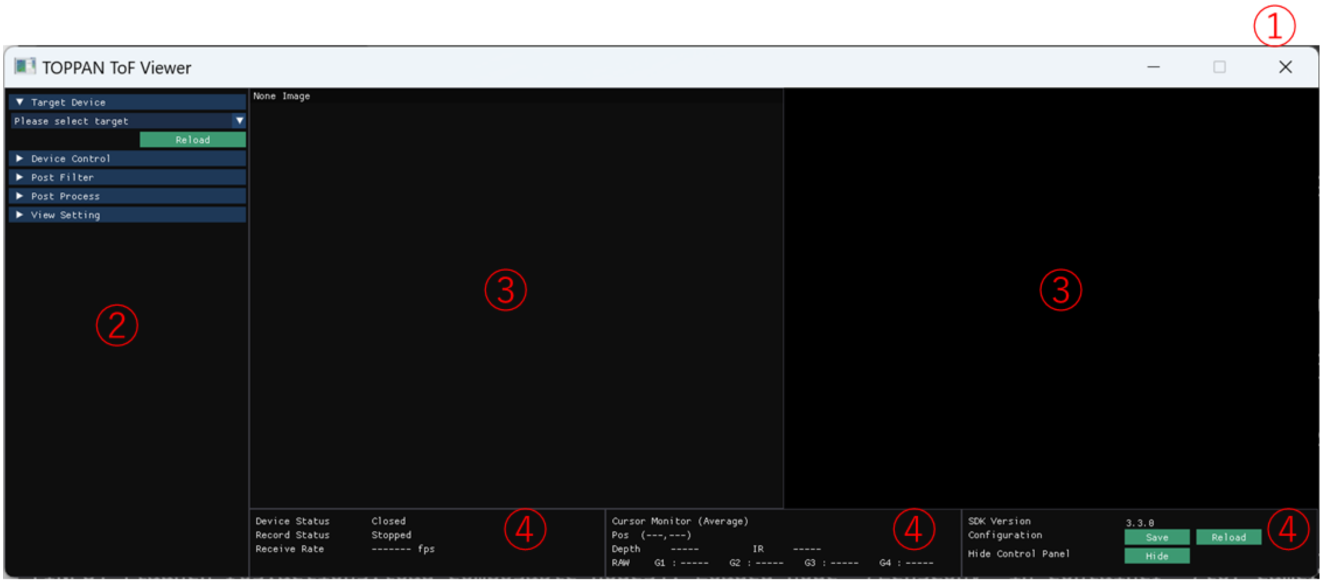
```
cd {ROS2 path}
./scripts/ubuntu/stop_record.sh
```

7-4-3. Exit

Exit with the “Ctrl + C” key on the terminal that started Node using the Launch file.

8. Run Sample Viewer

When launching the Sample Viewer (TOPPAN ToF Camera Viewer) with "launch_viewer.py", the following window screen will be displayed. For application function descriptions and operating procedures, please refer to the relevant chapters in the "***senSPure™ C11U User's Guide***" and others described in "***1-3. Related documents***".



```
Device Status    Closed
Record Status    Stopped
Receive Rate     ------- fps
```

```
Cursor Monitor (Average)
Pos  (---,---)
Depth     -----        IR     -----
RAW    G1 : -----    G2 : -----    G3 : -----    G4 : -----
```

```
SDK Version            3.3.0
Configuration
Hide Control Panel
```

Figure 11. Viewer window

Table 100. Viewer function

No.	Name	Description
①	Main Screen	Window for drawing each panel.
②	Control Panel	Allows operation of various controls and settings.
③	Image Display Panel	Displays various images or point clouds.
④	Status Display Panel	Displays various status information, received frame rate, etc.

9. Run rviz2

When “launch_rviz2.py” is specified as the Launch files, rviz2 starts up as shown below and displays the depth image, IR image, and Point Cloud data received from the camera.

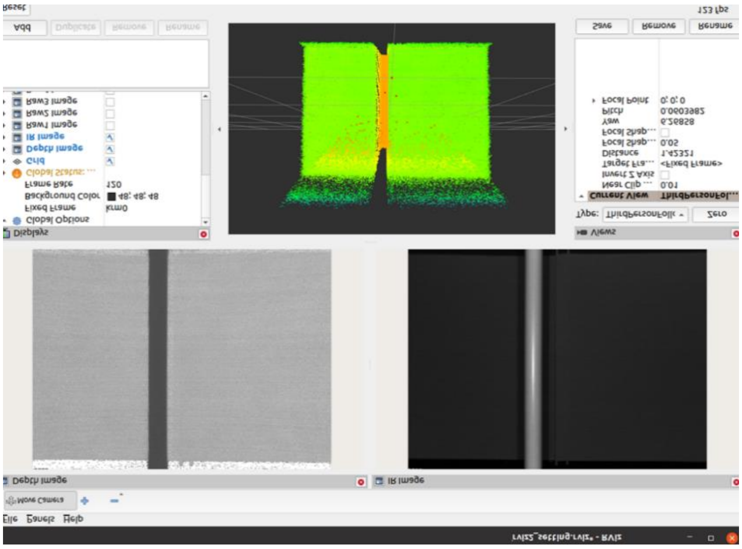


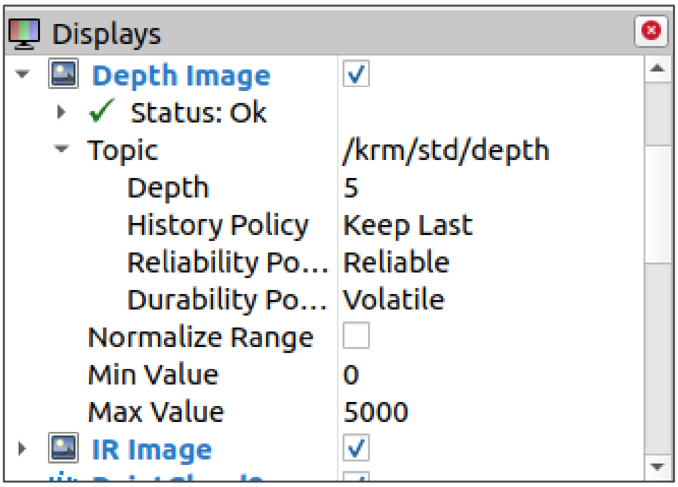
Figure 12. rviz2 window

9-1-1. FrameID setting

By default, the rviz2 display target FrameID is set to “krm 0”. Set the FrameID specified for Displays - Global Options - Fixed Frame according to the cam_index value of “tof _ param.yaml”.

9-1-2. View color setting for Depth image

By default, the depth image range set for display in rviz2 is 0~5000 [mm]. Because it is not linked to the distance range of the operation mode information, change the value of Displays - Depth Image - Min Value/Max Value according to the distance range of the operation mode and the range you want to display.



Displays
- Depth Image ☑
  - ✓ Status: Ok
  - Topic /krm/std/depth
    - Depth 5
    - History Policy Keep Last
    - Reliability Po... Reliable
    - Durability Po... Volatile
  - Normalize Range □
  - Min Value 0
  - Max Value 5000
- IR Image ☑

Figure 13. View color setting for Depth image

10. Constraints on use

10-1. Data output constraints

When a C11U camera captures images with a long interval between frames, it may output frame data that contains a high level of noise components from dark current*Note1*. To remove frames affected by dark current, the first two frames immediately following the start of frame output are discarded internally by the Camera class. However, this does not apply when the external trigger type is Secondary (Slave, EXT_TRG_SLAVE) and there are long intervals between external triggers, in which case frames affected by dark current may appear.

Depending on the host computer's OS and operating environment, frame acquisition may be unstable, so the first frame acquired after frame data output begins may not be the third frame.

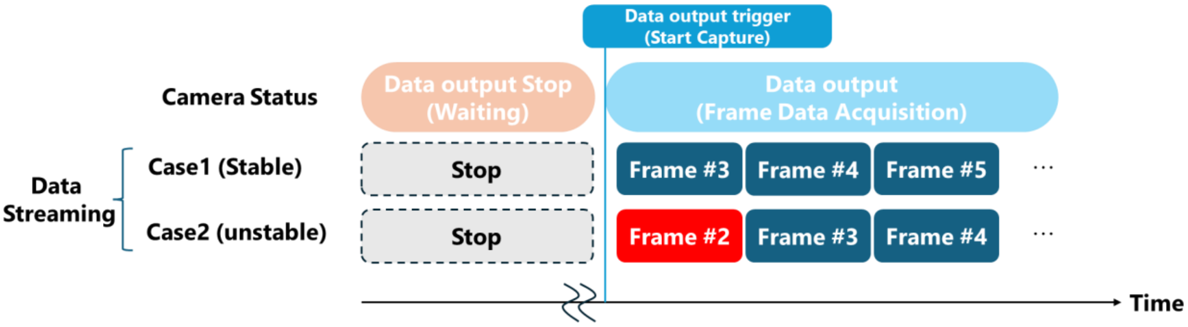


Figure 14. Data Output Constraints

***Note*[1]**:

Dark current is a type of noise that occurs even when the light-receiving element (pixel) is not exposed to light. This dark current noise mixes with the optical signal, which represents the actual distance information, and degrades the accuracy of distance measurement. The effect of dark current noise can be particularly pronounced in frame data acquired after the pixels have been idle for a long period.

10-2. Constraints for external trigger type Secondary (Slave) mode

When the external trigger type is Secondary (Slave, EXT_TRG_SLAVE), if the system waits for an image for a long period without receiving one, it may place a heavy load on the host PC. Because prolonged high-load operation can lead to malfunctions or failure, we recommend canceling the wait state if it will be a long period of waiting for reception.

10-3. PointField Information (Data Array Change)

If you are using the old TOPPAN ToF SDK ROS 2 package (version 1.4.1 or earlier), the point cloud data obtained from the StdMsg Node does not have its byte storage order consistent with the x, y, z order defined in the PointField information. When using point cloud data, please pay attention to the byte storage order. In the latest TOPPAN ToF SDK ROS 2 package (version 1.4.2 or later), the byte storage order has been unified with the x, y, z definitions in the PointField information, so you can now acquire the coordinates by simply referencing the definitions.

The table below shows the byte storage order for a coordinate of (x, y, z) = (1, 2, 3) output from the LensConv Node.

Table 101. The byte storage order for a coordinate of (x, y, z) = (1, 2, 3) output

	Output Method	Output Result: x	Output Result: y	Output Result: z
StdMsg ver.1.4.1 or earlier	Byte order	-1	2	3
	Referencing PointField info.	3	- 1	2
StdMsg ver.1.4.2 or later	Byte order	3	-1	2
	Referencing PointField info.	3	-1	2

- Byte order (Byte array): Refers to the contiguous sequence of bytes (8-bit units) when point cloud data is stored in memory or a file. The coordinates for each point (x, y, z) are stored sequentially in bytes.
- PointField: This is information that defines the element name, data type, and storage location for each attribute (e.g., x, y, z) of the PointCloud data. By referencing this information, you can correctly extract attribute values from the byte array.

Note that there is no specification changes related to the data output from the LensConv Node across different versions. The storage order of the byte array and the PointField information consistently match in the x, y, z order.

11. Terms of Use and Disclaimer

Please refer to the “C11U User’s Guide”, “TOPPAN ToF *senSPure*™ SDK API Reference Manual”, and other related documents for the terms and conditions of use for products from TOPPAN Holdings Inc. and TOPPAN Inc. (hereinafter referred to as "the Company").

- Unauthorized copying, reproduction, or reprinting of any part or all of this document is strictly prohibited.
- The contents of this document are subject to change without notice.
- While the Company takes every measure to provide accurate information, it does not assume responsibility for errors or omissions. Furthermore, the Company is not liable for any damages resulting from the use of the information contained in this document.
- The Company shall not be liable for any damages, including but not limited to data loss, loss of opportunity, loss of profit, and other incidental, indirect, or consequential damages, arising from the use of this product.
- Product names and company names, and other proper nouns mentioned in this document and related documents, belong to their respective companies or individuals. In this document, the TM (™) and R (®) marks may be omitted. These names are used for identification and explanation purposes within this document only. The Company has no intention to infringe on any of these rights.

Document history

Date	Version	Comment
2024/09/11	1.00	Initial release
2025/03/19	1.10	1. Release For C11U ES version
2025/04/23	1.11	1. Modified minor typos
2025/06/10	1.12	1. 2-3-4-3. Build Procedure (Ubuntu/Jetson AGX Orin); Modified build path. 2. 1-4. ROS 2 version; Added NVIDIA Jetson and jetpack version.
2025/09/09	1.13	1. 1-4 Operating Environment Updated: Updated to include Ubuntu 22.04LTS and JetPack 6.0. 2. 10. Constraints on use: Added. 3. 2-2-2-1. Source Set: Wording corrected to indicate that package exclusion will not occur. 4. 3-2-1. Point Cloud Output of LensConv Node and StdMsg Node: Corrected the coordinate system diagram for the StdMsg Node. 5. Modified minor typos.

TOPPAN

TOPPAN ホールディングス株式会社 TOF 事業推進センター
TOF Business Development Center, TOPPAN Holdings Inc.

TOPPAN 株式会社 エレクトロニクス事業本部
Electronics Division, TOPPAN Inc.

Location
(日本語) 〒108-8539 東京都港区芝浦 3-19-26 トッパン芝浦ビル
(English) 3-19-26, Shibaura, Minato-ku, Tokyo, 108-8539
E-mail electronics@toppan.co.jp
Website https://www.toppan.com/ja/electronics/device/tof/ (TOPPAN Inc.)

ToF camera product support center
For support related to ToF camera products, please contact the designated support center.
E-mail btop_support@toppan.co.jp