search
DFRobot Products (DFR) Index
Sensor (SEN) Index
Telecom Module (TEL) Index
Motor Driver (DRI) Index
Fitting (FIT) Index
Robot (ROB) Index
micro:bit (MBT) Index
Kit (KIT) Index
Toy (TOY) Index
Servos (SER) Index
What is XX series
How to select series
How it works series
PinPong Library

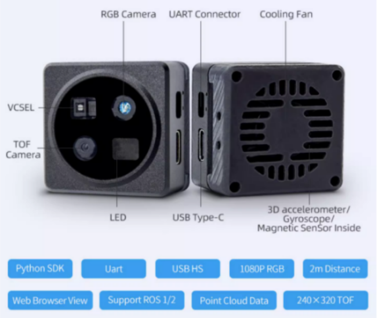
Introduction
Preparation
On-page preview
Interactive configuration
Save data
SSH login
Case
Custom development
Use ROS
Use ROS1
Use ROS2
More Documents

SKU:SEN0583

Introduction

The RGB-D camera is an advanced 3D time-of-flight (TOF) camera module equipped with RGB capabilities. It combines depth measurement and RGB image capture functionalities seamlessly. Utilizing a simple USB data connection, this camera module effortlessly delivers TOF depth data and RGB images at a remarkable speed of 30 frames per second. With the aid of a web-based control interface, users can swiftly preview depth maps, point cloud data, and experience real-time, vibrant 3D visualization. Moreover, it offers effortless plug-and-play functionality on Linux systems, eliminating the need for any additional drivers. It seamlessly integrates with ROS1/2 and provides open access to packages, sample codes, and a Python Software Development Kit (SDK). Leveraging the acquired point cloud data, various applications such as 3D facial recognition, robotic arm manipulation, and SLAM mapping can be realized with unparalleled precision.

An integrated, cost-effective, and user-friendly 3D Time-of-Flight (ToF) solution designed specifically for Internet of Things (IoT) applications. This solution encompasses various algorithms that enable distance measurement, multi-zone body positioning sensing, posture detection, posture control, and essential calibration functionalities. These features find wide-ranging applications in numerous fields, including white goods, laser televisions, smart projectors, intelligent lighting, smart parking systems, and home automation facilities.

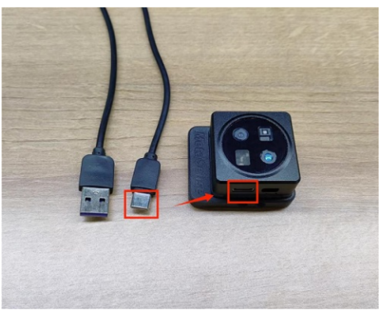


Preparation

To run preview module on Windows, install the driver.

Windows Installation driver tutorial:Click to view

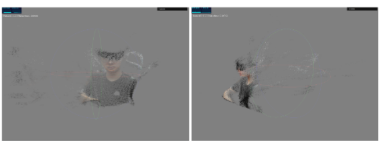
On-page preview



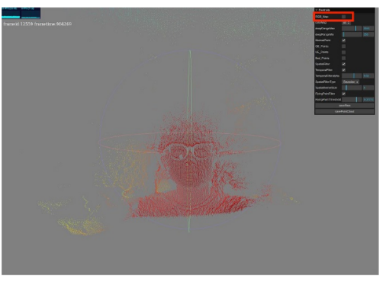
Before using this device, make sure that the 192.168.233.0/24 address segment is not occupied in your network environment, because the MS-A075V uses RNDIS and sets its IP address http://192.168.233.1.

Connect the module to the PC with power according to the figure above, then the built-in fan will start to work and the red light will be displayed at the lens. Select the browser and enter http://192.168.233.1 to preview the 3D point cloud image. After power-on, the system and program will be started 10s-15s delayed.

Quick preview using web upper computer (front and side) :

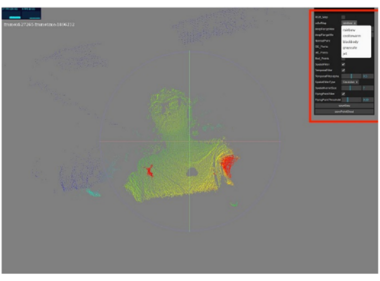


We can preview the depth pseudo-color point cloud map. Open the interaction panel in the upper right corner, uncheck RGB_Map in the first line.



Interactive configuration

The preview webpage contained many function configurations, we can change them to get different live preview result.



Here tells the functions of each widgets.

- **RGB_Map** checkbox, control RGB map. Display the deep pseudo-colored point map when checked, display the RGB-mapped point cloud map when unchecked.
- **colorMap** drop-down bar, provides several pseudo-color mapping options(cmap)， jet is recommended. Avaliable when RGB_Map is unchecked.
- **deepRangeMax** and **deepRangeMin** slide bar are used for setting the mapping range of cmap, the depth value between deepRangeMin and deepRangeMax are OK. Avaliable when RGB_Map is unchecked.
- **NormalPoint** checkbox, control the display of normal point(There maybe some invalidation points when TOF worked, need to do opposite action). Recommend checked.
- **OE_Points** checkbox, control the display of OE point. Recommend unchecked.
- **UE_Points** checkbox, control the display of UE point. Recommend unchecked.
- **Bad_Points** checkbox, control the display of bad point. Recommend unchecked
- **SpatialFilter** checkbox, control the Spatial filtering. Processing based on the spatialFilterSize value below and the algorithm specified by the SpatialFilterType.
- **TemporalFilter** checkbox, control the Temporal filtering, A time average is calculated based on the temporalfilteralpha values below.
- **TemporalFilteralpha** slide bar, set the time for Temporal filtering. Adjust it moderate, can be tested by yourself.
- **SpatialFilterType** drop-down bar, set the Spatial filtering algorithm, provides Gaussian filtering and Bilateral filtering. Bilateral filtering requires high performance, not recommended.
- **SpatialFilterSize** slide bar, set the range for Spatial Filter. Adjust it moderate, can be tested by yourself.
- **FlyingPointFilter** checkbox, control the flying point filter. Set

the the following FlyingPointThreshold value as the filtering threshold, those that exceed the threshold will be filtered out. Set it moderate, otherwise the validation points will be filtered out.

Save data

The webpage provides 2 buttons at the bottom of the control bar.

- **SaveRaw**：Save one frame raw data. If you want to use the depth data or IR or RGB data for development, you need to know the data struct of raw. We provide a detailed jupyter notebook about the data processing of raw for users and developers.
- **SavePointCloud**：Sava one frame 3D point map, and its saved format is pcd. Can be previewed via the script provided above.

Note: The data of raw can be get through open interface, which developers can do development on. While the pointcloud data does not provide any interface since it's calculated based on the raw data and camera internal parameters.

SSH login

In addition to using the web page to preview directly, we can also login using SSH with password root

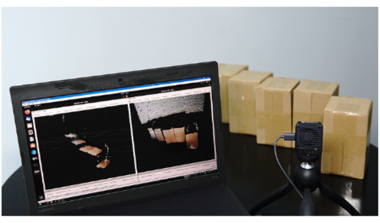
In the web preview page we know the ip address of this device is 192.168.233.1, with which we can login to this device.

```
ssh root@192.168.233.1
```

Case

Real shooting of distant, near and far point cloud

High-precision mapping of differences in object placement distances, point cloud maps can intuitively feel a more realistic visualization.



Custom development

python SDK

This is a SDK based on Python3. MS-A075V opens its http interface, we can get its origin data (Depth map, ir map, rgb map) through http request.

To help user understand the struct of data package and the relevant logic of decoding, we provide decoding related functions that encapsulate http requests and native data, based on which users can do custom development.

Get SDK:Click to download

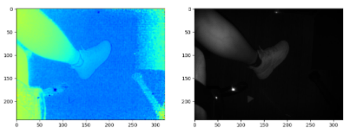
Method：Install jupyter, connect to the TOF module, then open the toturial.py file.

Decode and stream

After understanding the struct of data package and the relevant logic of decoding from Python SDK, we can do advanced development, continuously get, decode and call the third-party python image library, like matplotlib for live display. The toturial.py gives the implementation of getting onr frame data, and it can achieve live display by plt with loop.

Decoding and steraming:Click me to see content of stream.py

Methods：Run command python stream.py after installing all dependent packages.

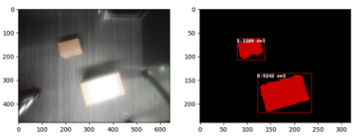


Detect volume

Based on third-party python library，and have understood the logic of getting and decoding data, we can do more development: Continuously display frames, roughly calculate the point cloud through the data from TOF module by SDK, do the accumulation to get the total volume. Limit: The top view should include all details except bottom.

Detect volume：Click to view calVolumes.py

Methods：Run command python calVolumes.py after installing all dependent packages, there will be notice after you run it.



Use ROS

To begin this, install ROS on your computer first.

Use ROS1

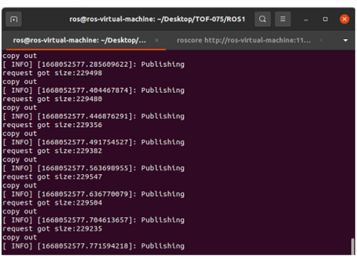
Preparation

Prepare a Linux enviroment for ROS.

Install and RUN

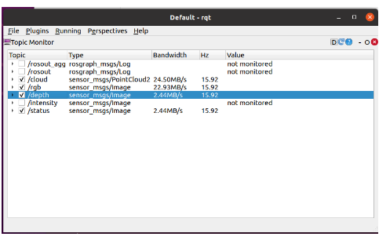
Access package download.

```
#Extract sipeed_tofv075-ros.zip, and open its path
source /opt/ros/*/setup.sh
catkin_make
source devel/setup.sh
rosrun sipeed_tof_cpp publisher
#Then the terminal continuously refreshes the command line
```


ros@ros-virtual-machine: ~/Desktop/TOF-075/ROS1
ros@ros-virtual-machine: ~/Desktop/...
roscore http://ros-virtual-machine:11...

```
copy out
[ INFO] [1668052577.285609622]: Publishing
request got size:229498
copy out
[ INFO] [1668052577.404467874]: Publishing
request got size:229480
copy out
[ INFO] [1668052577.446876291]: Publishing
request got size:229356
copy out
[ INFO] [1668052577.491754527]: Publishing
request got size:229382
copy out
[ INFO] [1668052577.563698955]: Publishing
request got size:229547
copy out
[ INFO] [1668052577.636770079]: Publishing
request got size:229504
copy out
[ INFO] [1668052577.704613657]: Publishing
request got size:229235
copy out
[ INFO] [1668052577.771594218]: Publishing
```

View frames in RQT



Topic	Type	Bandwidth	Hz	Value
☐ /rosout_agg	rosgraph_msgs/Log			not monitored
☐ /rosout	rosgraph_msgs/Log			not monitored
☑ /cloud	sensor_msgs/PointCloud2	24.50MB/s	15.92	
☑ /rgb	sensor_msgs/Image	22.93MB/s	15.92	
☑ /depth	sensor_msgs/Image	2.44MB/s	15.92	
☐ /intensity	sensor_msgs/Image			not monitored
☑ /status	sensor_msgs/Image	2.44MB/s	15.92	

4. RVIZ preview

Open rviz2, in the bottom left interface choose Add->By topic->PointCloud2 or/depth ->Image add ->Display/Global Options/Fixed Frame, changed it into tof, in this way it displays point cloud normally. According to the added content, the Image displays in the left and the point cloud display in the center.

Use ROS2

Preparation

Prepare a Linux enviroment for ROS.

Install and RUN

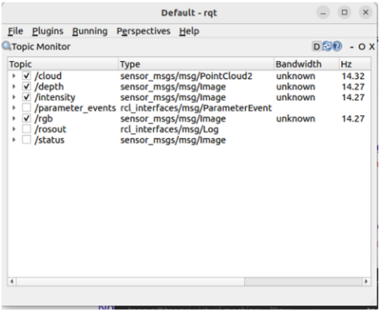
We have provided the functional package for ROS2, users need compile and run it on the system with ROS2.Access package download.

```
#Extract sipeed_tofv075_ros2.zip, and open its path
source /opt/ros/*/setup.sh
```

```
colcon build #If it indicates missing colcon, use command sudo
source install/setup.sh
ros2 run sipeed_tof_cpp publisher
#Then the terminal continuously refreshes the command line and
```

View frames by RQT

Open RQT and select Plugins->Topics->Topic Monitor.

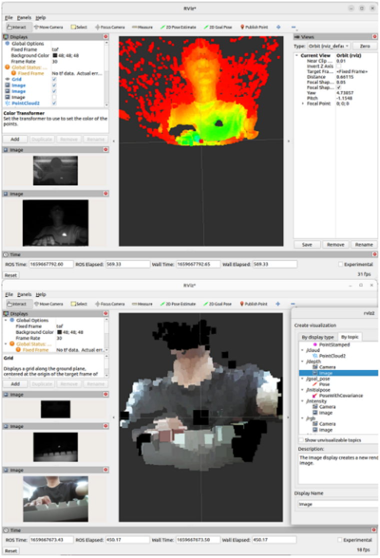


Topic	Type	Bandwidth	Hz
☑ /cloud	sensor_msgs/msg/PointCloud2	unknown	14.32
☑ /depth	sensor_msgs/msg/Image	unknown	14.27
☑ /intensity	sensor_msgs/msg/Image	unknown	14.27
☐ /parameter_events	rcl_interfaces/msg/ParameterEvent		
☑ /rgb	sensor_msgs/msg/Image	unknown	14.27
☐ /rosout	rcl_interfaces/msg/Log		
☐ /status	sensor_msgs/msg/Image		

RVIZ2 Priview

Open rviz2, in the bottom left interface choose Add->By topic->PointCloud2 or/depth ->Image add ->Display/Global Options/Fixed Frame, changed it into tof, in this way it displays point cloud normally. According to the added content, the Image displays in the left and the point cloud display in the center.

The result of mixing Pseudo-colored point clouds and RGB:



More Documents

Downloaded from Arrow.com.