Welcome

Thank you for choosing Freenove products!

Get Started

Please follow this tutorial to set up the camera.

Get Support

Encounter problems? Don't worry! Refer to “TroubleShooting.pdf” or contact us.

When there are packaging damage, quality problems, questions encountering in use, etc., just send us an email. We will reply to you within one working day and provide a solution.

support@freenove.com

About

Freenove provides open source electronic products and services.

Freenove is committed to helping customers learn programming and electronic knowledge, quickly implement product prototypes, realize their creativity and launch innovative products. Our services include:

- Kits for learning programming and electronics
- Kits compatible with Arduino®, Raspberry Pi®, micro:bit®, etc.
- Kits for robots, smart cars, drones, etc.
- Components, modules and tools
- Design and customization

To learn more about us or get our latest information, please visit our website:

http://www.freenove.com

Copyright

All the files, materials and instructional guides provided are released under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. A copy of this license can be found in the folder containing the Tutorial and software files associated with this product.All the files provided in the ZIP file are released under. You can find a copy of the license in the ZIP file.

This means you can use these resource in your own derived works, in part or completely, but NOT for the intent or purpose of commercial use.
Freenove® brand and logo are trademarks of Freenove Creative Technology Co., Ltd and cannot be used without written permission.

FREENOVE™

Other registered trademarks and their owners appearing in this document:

Arduino® is a trademark of Arduino LLC (https://www.arduino.cc/).
Raspberry Pi® is a trademark of Raspberry Pi Foundation (https://www.raspberrypi.org/).
micro:bit® is a trademark of Micro:bit Educational Foundation (https://www.microbit.org/

Get Started

Note (Important)

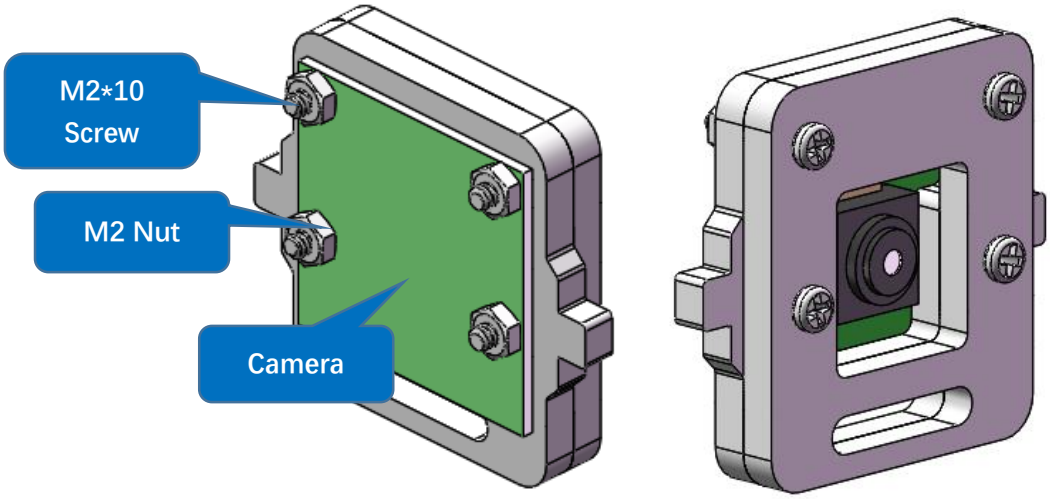
Please note that our FNK0056 prodcut, which comes with the camera model OV5647, is only compatible with Raspberry Pi; while the product FNK0056B, with camera model IMX219, can work with both Raspberry Pi and Jetson Nano.

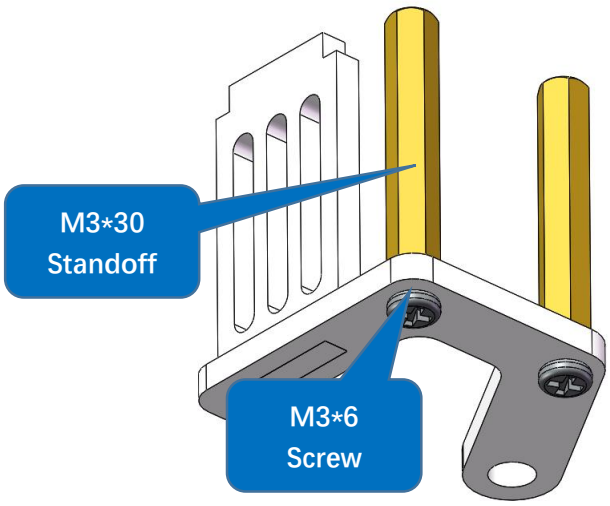
This tutorial introduces the use of product FNK0056B (Camera IMX219) on Jetson Nano.

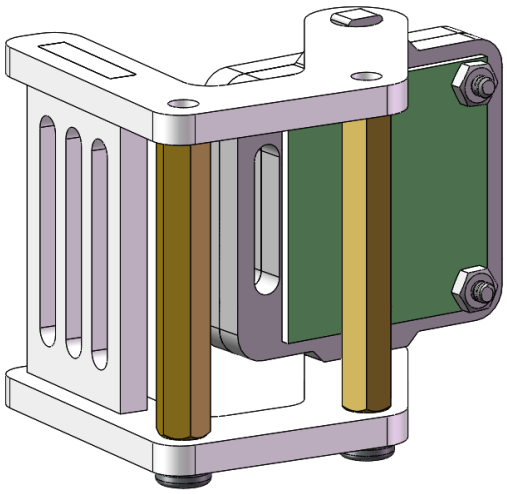
Product SKU	Camera Model	Work with Raspberry Pi	Wrok with Jetson Nano
FNK0056	OV5647	Yes	No
FNK0056B	IMX219	Yes	Yes(this tutorial)

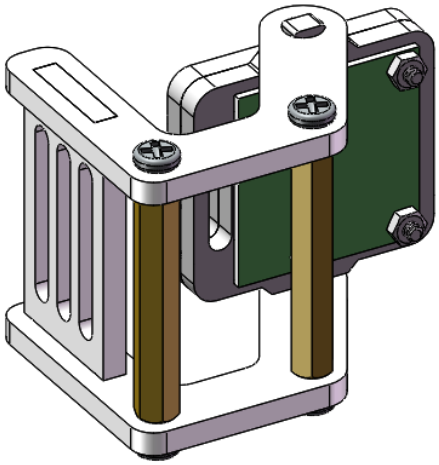
Step 1 Assembly

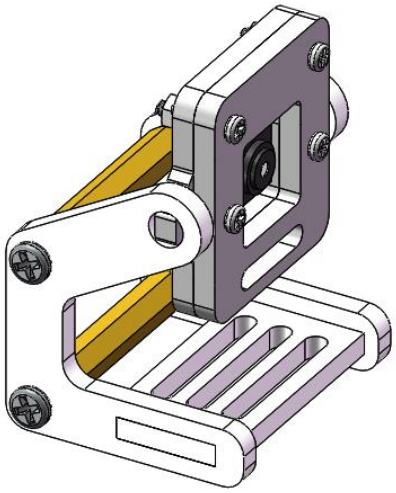
Assemble the camera support



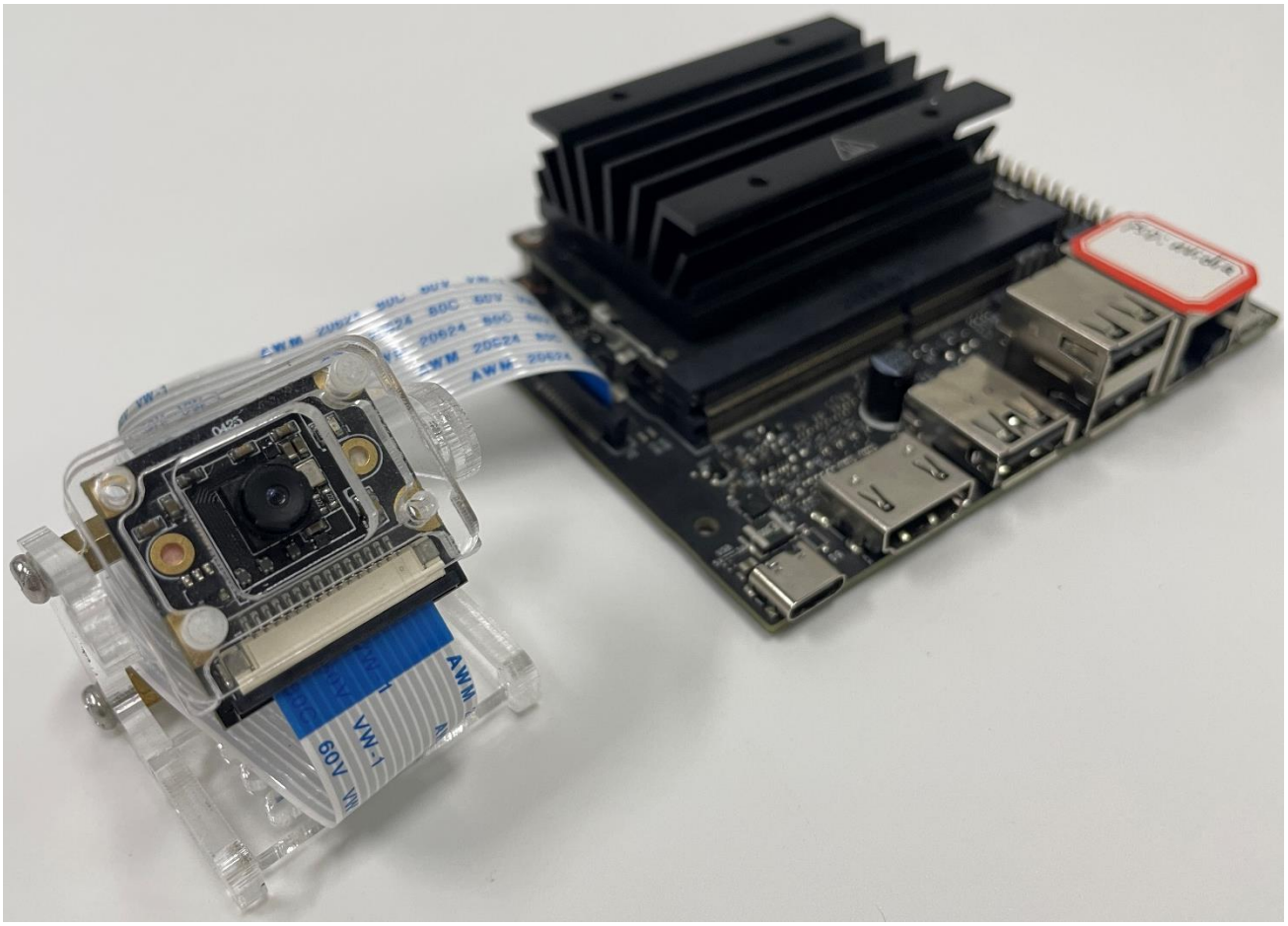

M3*30
Standoff
M3*6
Screw







Step 2 Camera Cable Connection



Step 3 Run the Code

Official Ubuntu OS has configured all necessary libraries for JetsonNano. You only need to download and run the code.

Open Terminal,

1. Download tutorial and code with the command:

```
git clone https://github.com/Freenove/Freenove_Camera_Module_for_Raspberry_Pi
```

2. Enter JetsonNano folder with cd command.

```
cd ~/Freenove_Camera_Module_for_Raspberry_Pi/JetsonNano
```

3. Run the code

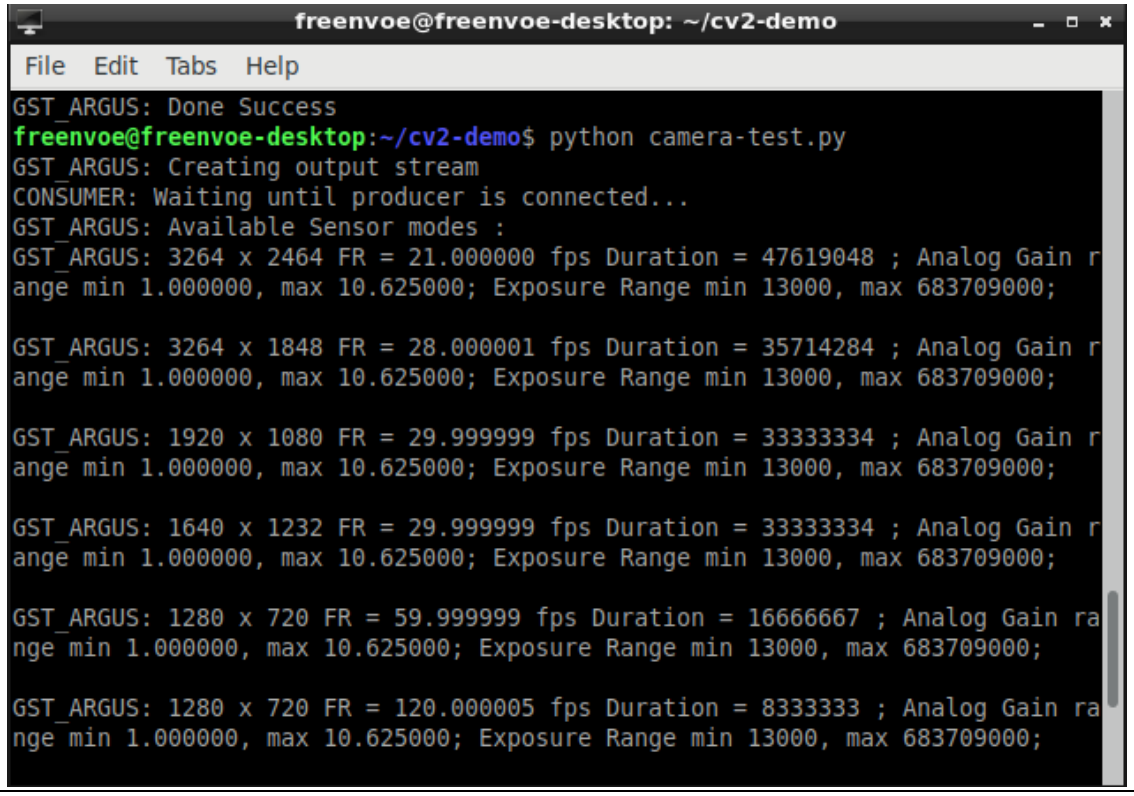
```
python camera-test.py
```

After the code runs successfully, a real-time video will be displayed on a new window.

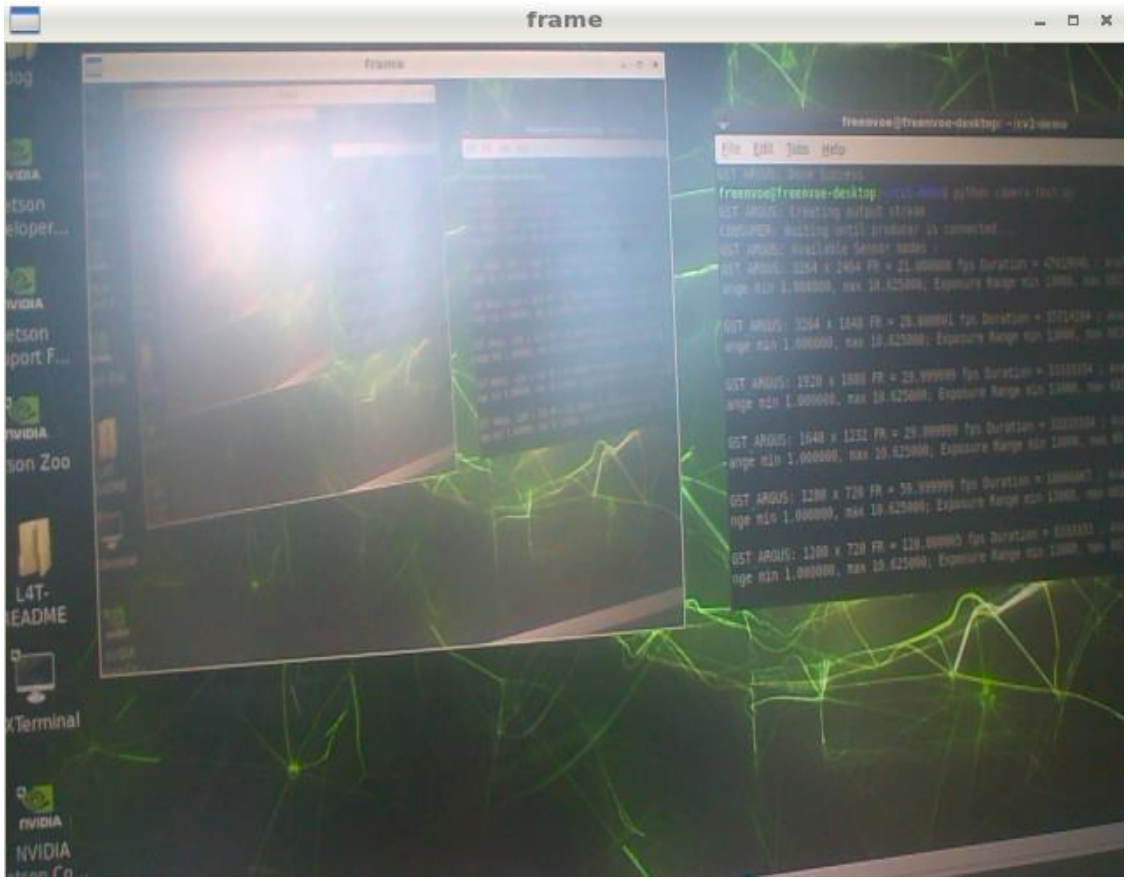
You can press any key on the video window or press Ctrl-C on Terminal to exit the program.

As shown below:

Terminal:



Camera:



Code Analysis

Here is the complete code:

```
#!/usr/bin/env python3
################################################################################
# @file camera-test.py
# @author suhayl@freenove (support@freenove.com)
# @brief Buzzer service.
# @version v1.0.0
# @date 2023-3-29
#
# @copyright Copyright (c) 2023. Freenove corporation.
################################################################################

import cv2

def gstreamer_pipeline(
    capture_width=640,
    capture_height=480,
    display_width=640,
    display_height=480,
    framerate=30,
    flip_method=0,
    ):
    return (
        "nvarguscamerasrc ! "
        "video/x-raw(memory:NVMM), "
        "width=(int)%d, height=(int)%d, "
        "format=(string)NV12, framerate=(fraction)%d/1 ! "
        "nvvidconv flip-method=%d ! "
        "video/x-raw, width=(int)%d, height=(int)%d, format=(string)BGRx ! "
        "videoconvert ! "
        "video/x-raw, format=(string)BGR ! appsink"
        % (
            capture_width,
            capture_height,
            framerate,
            flip_method,
            display_width,
            display_height,
        )
    )

```

```
cap = cv2.VideoCapture(gstreamer_pipeline(flip_method=2), cv2.CAP_GSTREAMER)
while (cap.isOpened()):

    ret, frame = cap.read()
    cv2.imshow('frame', frame)
    if cv2.waitKey(50) >= 0:
        break

cap.release()
cv2.destroyAllWindows()

```

Camera resolusion and size of the display window can be set with gstreamer_pipeline function.

```
    capture_width=640,
    capture_height=480,
    display_width=640,
    display_height=480,
```

Troubleshooting

If any error occurs, please check the version of OPENCV Library, which needs to be later than 3.3.1.

```
python
```

```
import cv2
```

```
cv2.__version__
```