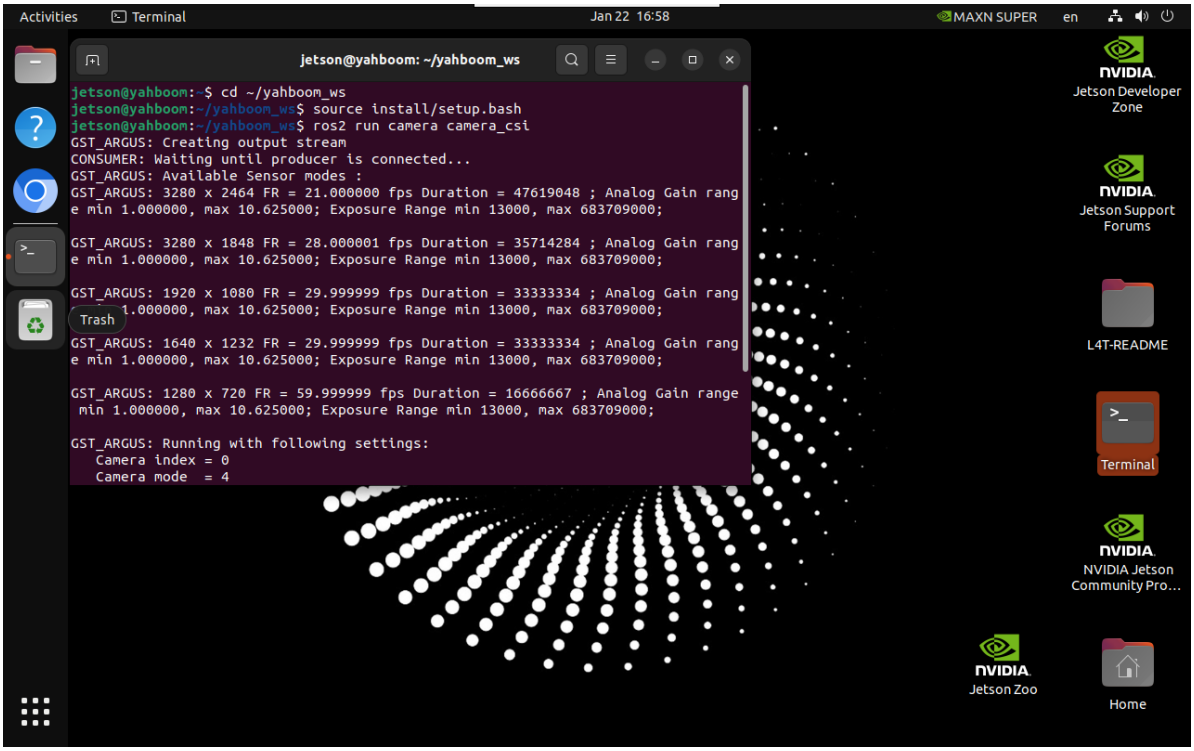
Camera calibration (CSI)

1. Start the camera

```
cd ~/yahboom_ws
```

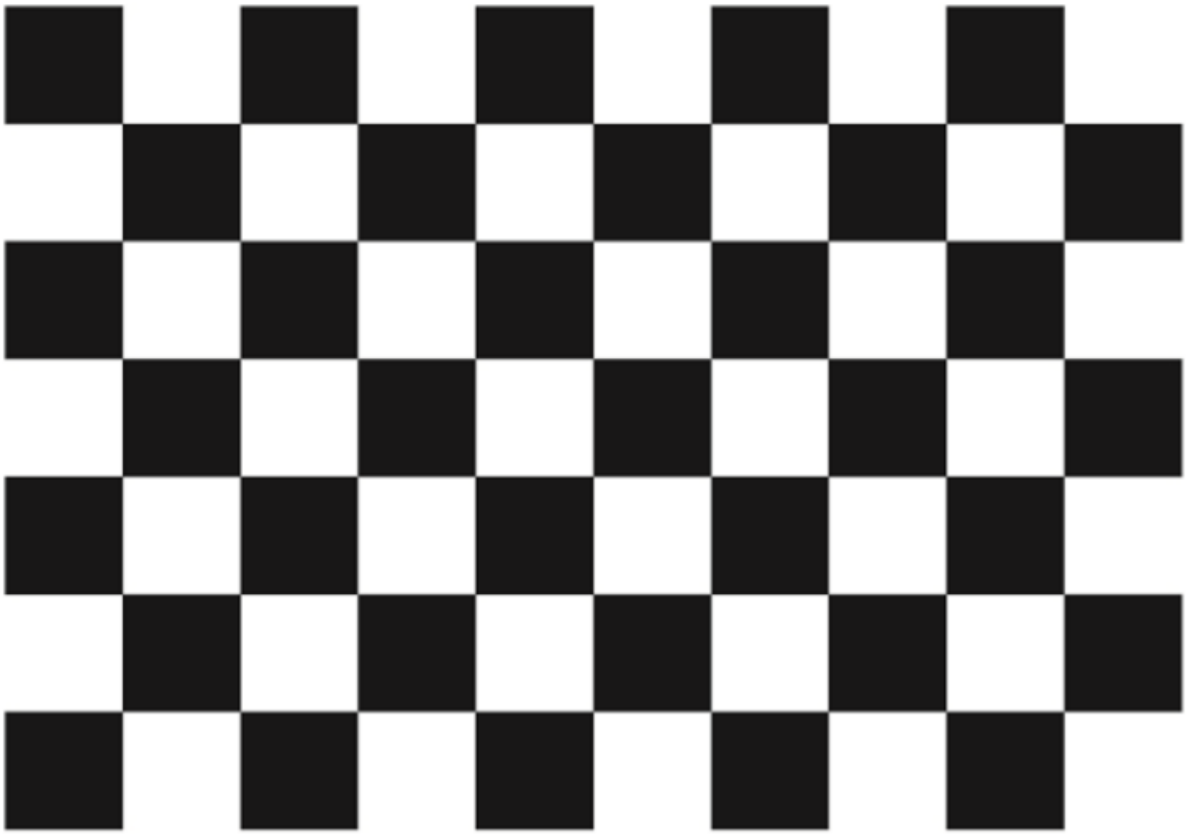
```
source install/setup.bash
```

```
ros2 run camera camera_csi
```



2. Image calibration

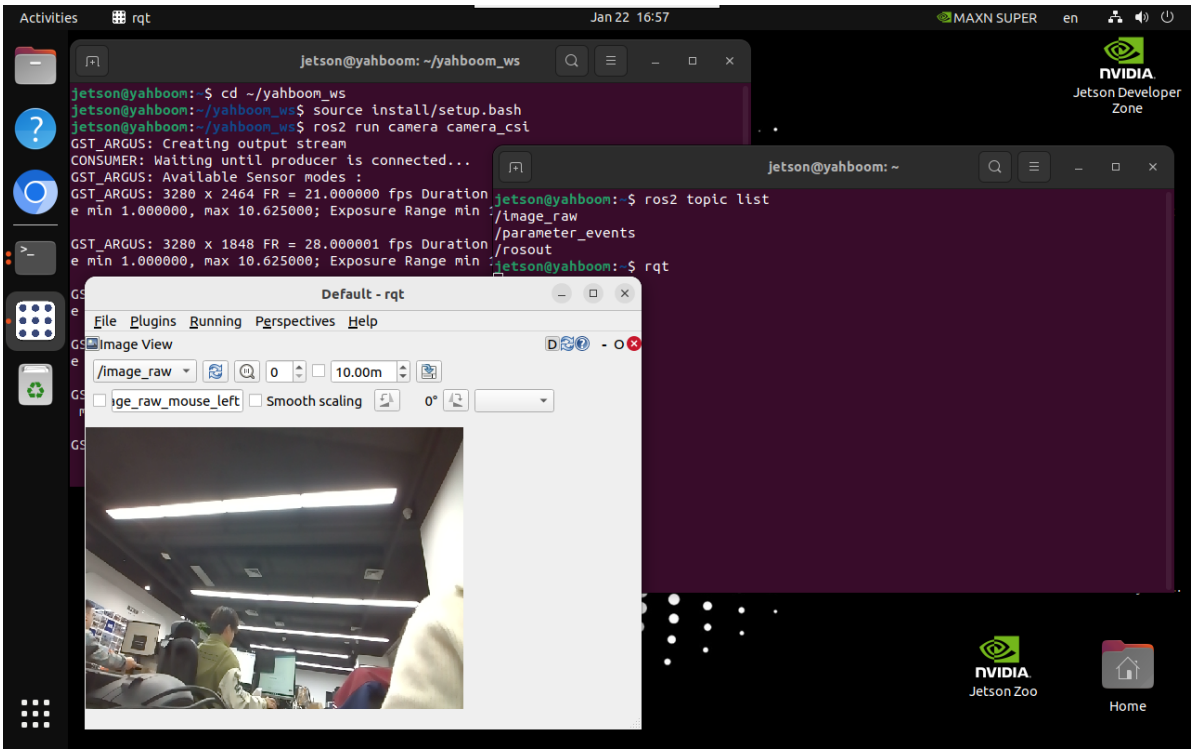
2.1. Calibration board



7×10 | Size: 20mm

2.2. View topics

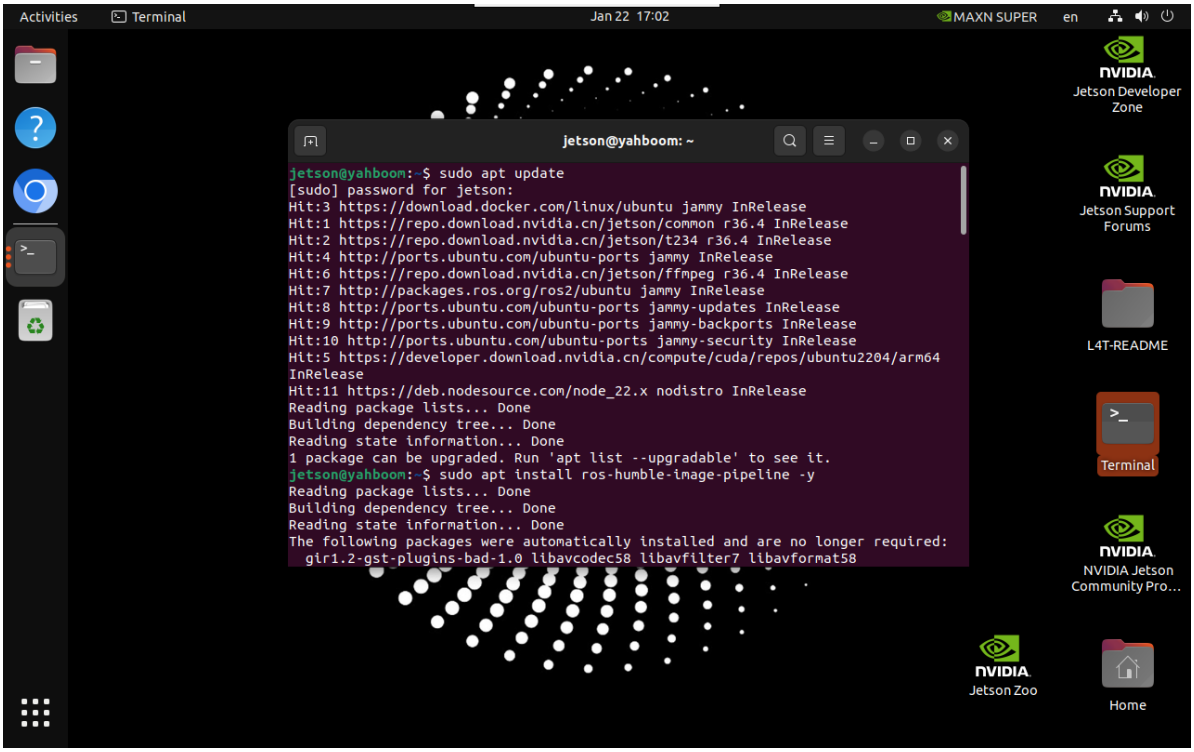
```
ros2 topic list
```



2.3. Image calibration

2.3.1. Install the image calibration tool

```
sudo apt update
sudo apt install ros-humble-image-pipeline -y
```



2.3.2. Start the image calibration tool

Start the ROS2 camera calibration tool `camera_calibration` and perform camera calibration: Before calibration, you need to start the camera first

```
ros2 run camera_calibration cameracalibrator --size 9x6 --square 0.02 --ros-args
--remap /image:=/image_raw
```

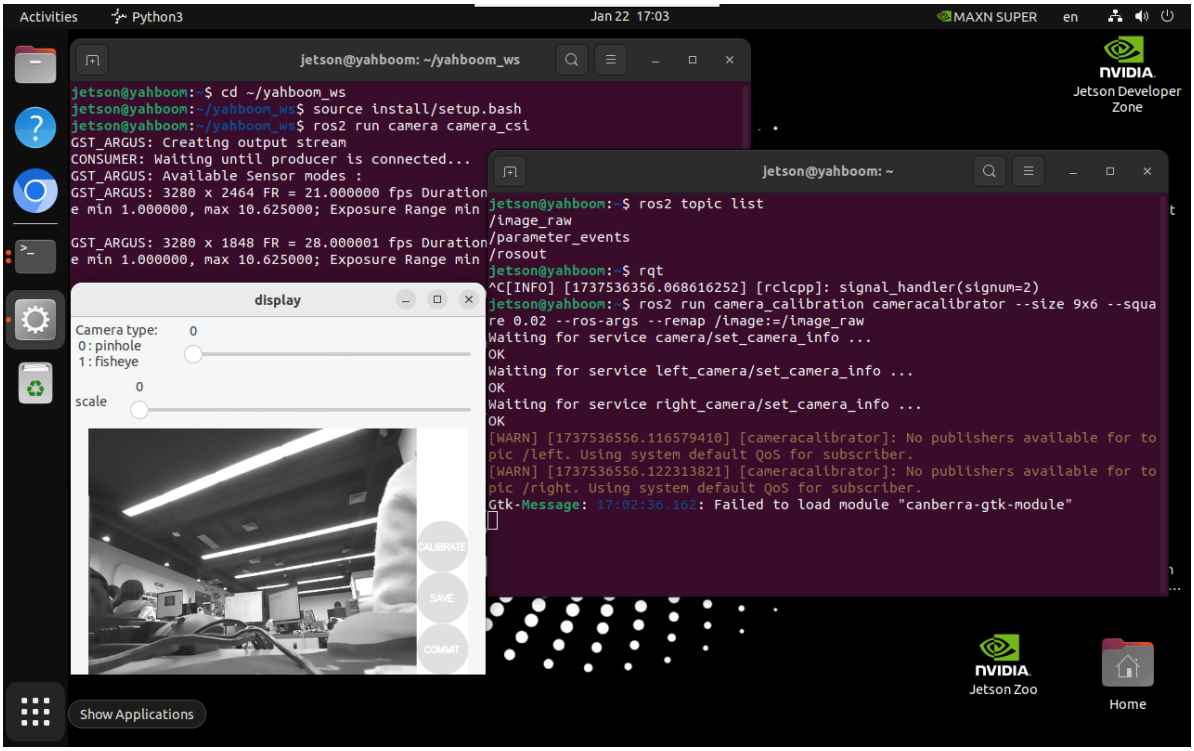
> **Parameter Description**

`--size`: The number of inner corner points of the chessboard → 9 rows and 6 columns of inner corner points

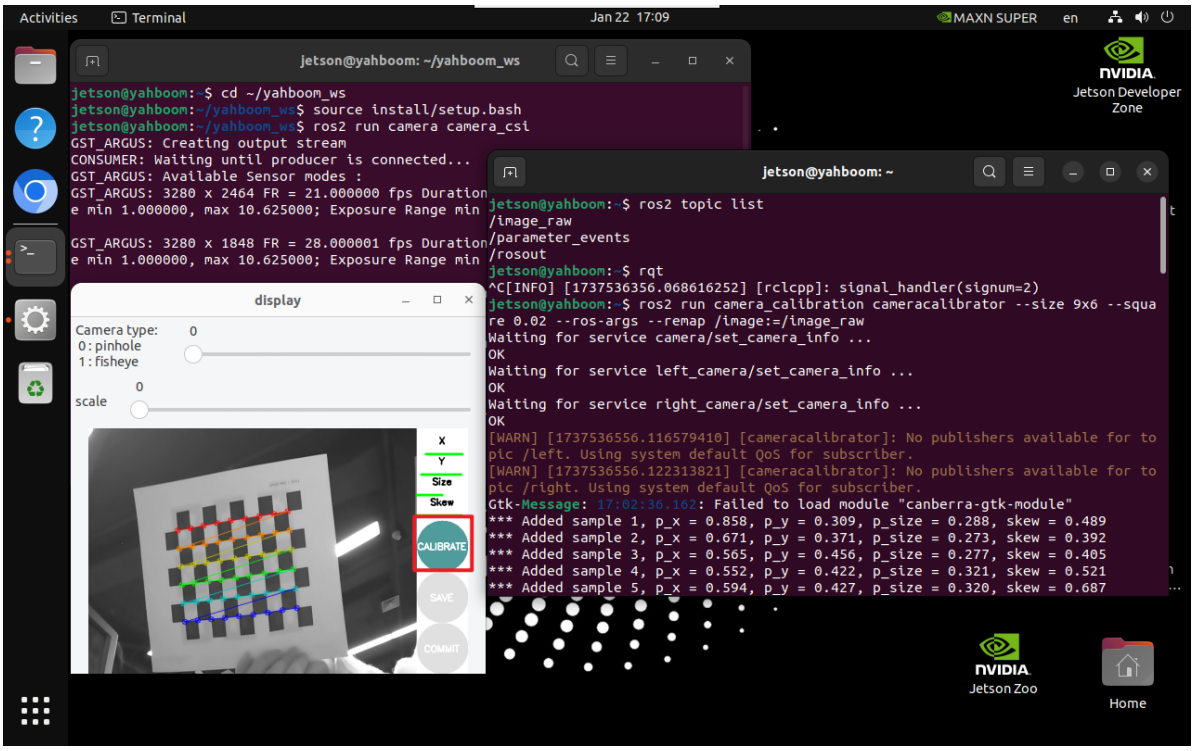
`--square`: The side length of the chessboard → 0.02 meters

`--ros-args`: Passing ROS parameters

`--remap`: Topic remapping → `/image_raw` as image input stream



2.3.3 Calibration process



> **Parameter Description**

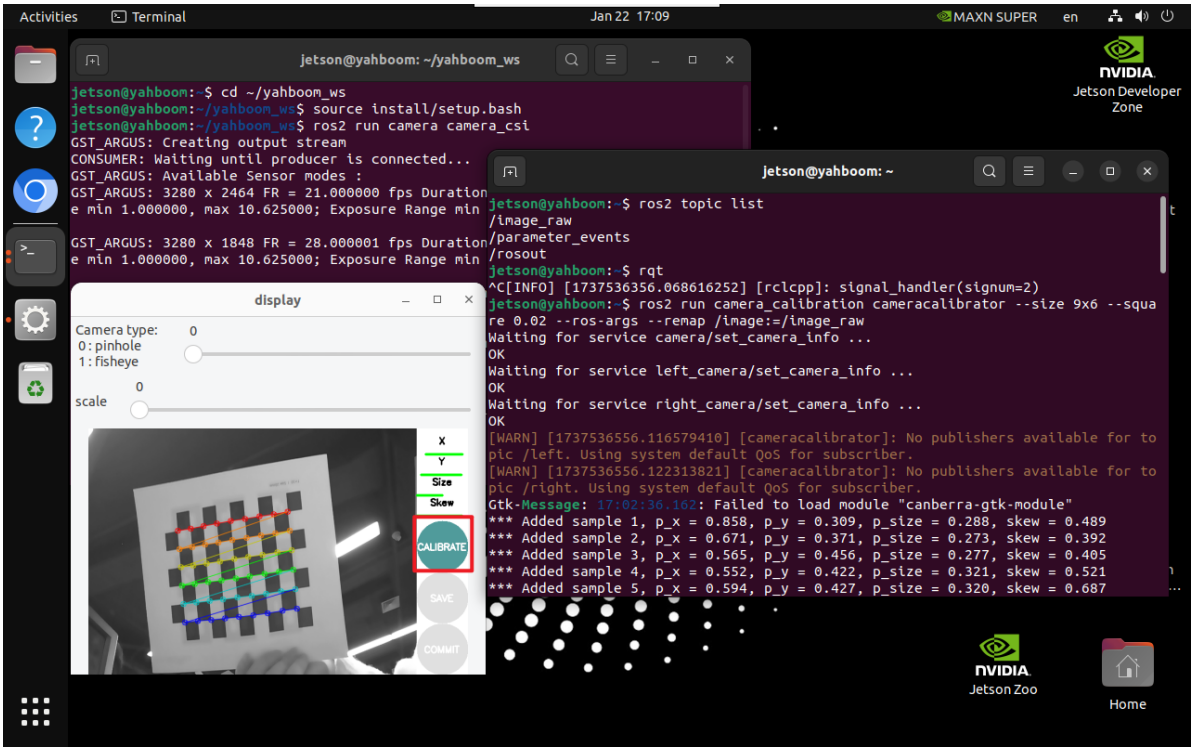
`X`: The chessboard moves left and right in the camera's field of view

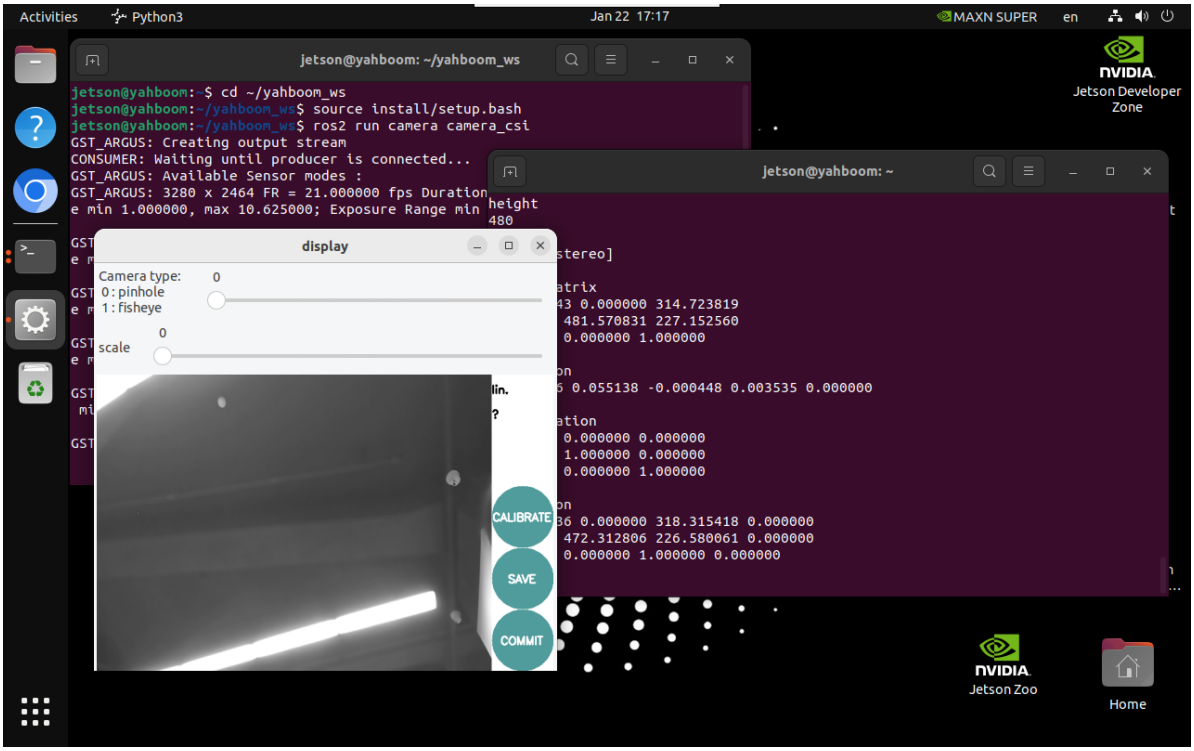
`Y`: The chessboard moves up and down in the camera's field of view

`Size`: The chessboard moves back and forth in the camera's field of view

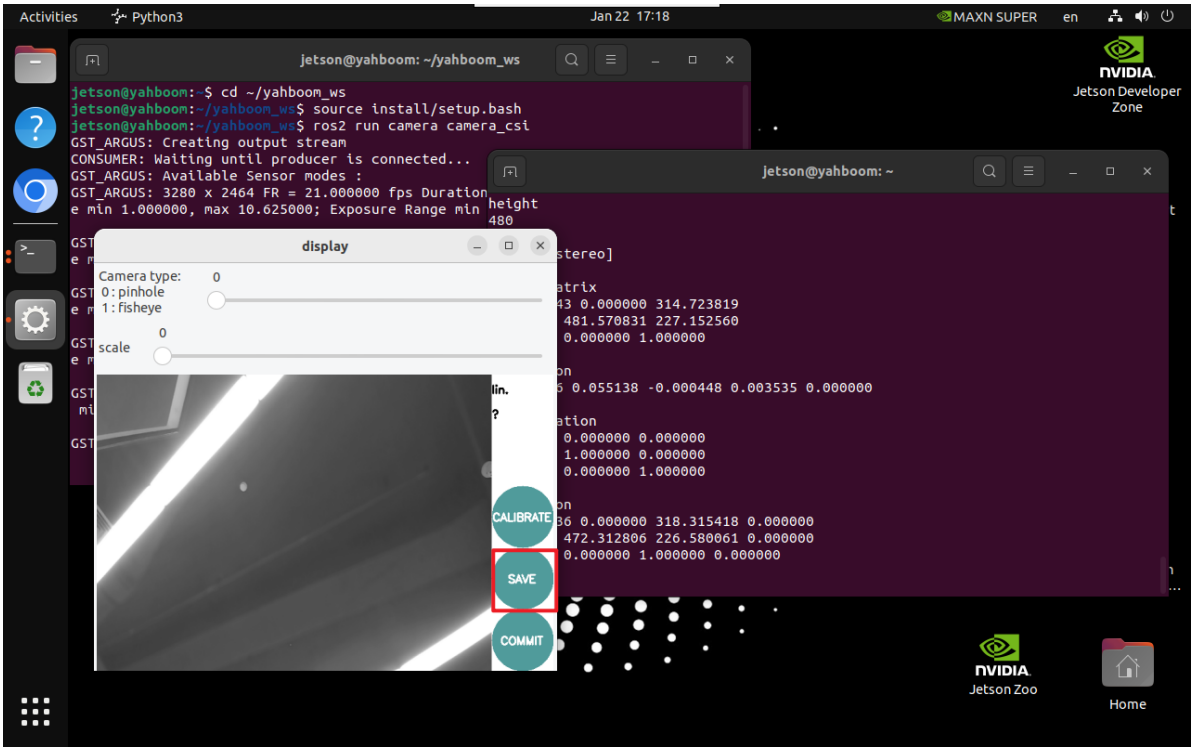
`Skew`: Tilt and rotate of the chessboard in the camera's field of view

When `X`, `Y`, `Size`, and `Skew` turn green, you can click `CALIBRATE` to calculate the camera internal parameters: The more pictures you calibrate, the longer it takes, and the program may get stuck.





After calibration, click `SAVE` to save the result!



Close the program after successful saving!

2.3.4, Calibration results

Default save location and file name of calibration results:

```
/tmp/calibrationdata.tar.gz
```

Copy and decompress the calibration results:

```
sudo mv /tmp/calibrationdata.tar.gz ~
mkdir calibrationdata
tar -xzvf calibrationdata.tar.gz -C ~/calibrationdata
```

The decompressed file contains the calibrated `*.png`, `ost.yaml` and `ost.txt` files. We mainly copy the data in the `ost.yaml` file to the `camera_usb.yaml` file.

> **ost.yaml**

Replace the `camera_usb.yaml` file data with the `image_width`, `image_height`, `data` in `camera_matrix`, `data` in `projection_matrix`, and `data` in `distortion_coefficients`.

```
image_width: 640
image_height: 480
camera_name: narrow_stereo
camera_matrix:
  rows: 3
  cols: 3
  data: [583.32121,   0.     , 290.8562 ,
          0.     , 783.65447, 223.15165,
          0.     ,   0.     ,   1.     ]
distortion_model: plumb_bob
distortion_coefficients:
  rows: 1
  cols: 5
```

```
  data: [0.106696, -0.116255, -0.018194, -0.011949, 0.000000]
rectification_matrix:
  rows: 3
  cols: 3
  data: [1., 0., 0.,
         0., 1., 0.,
         0., 0., 1.]
projection_matrix:
  rows: 3
  cols: 4
  data: [600.07017,   0.     , 284.95284,   0.     ,
           0.     , 807.38252, 216.33741,   0.     ,
           0.     ,   0.     ,   1.     ,   0.     ]
```

> **camera_csi.yaml**

Modified file:

```
%YAML:1.0
---
image_width: 1920
image_height: 1080
camera_name: camera
camera_matrix: !!opencv-matrix
  rows: 3
  cols: 3
  dt: d
  data: [583.32121,   0.     , 290.8562 ,
           0.     , 783.65447, 223.15165,
           0.     ,   0.     ,   1.     ]
distortion_model: plumb_bob
distortion_coefficients: !!opencv-matrix
  rows: 1
  cols: 5
  dt: d
  data: [0.106696, -0.116255, -0.018194, -0.011949, 0.000000]
rectification_matrix:
  rows: 3
  cols: 3
  data: [1., 0., 0.,
         0., 1., 0.,
         0., 0., 1.]
projection_matrix:
  rows: 3
  cols: 4
  data: [600.07017,   0.     , 284.95284,   0.     ,
           0.     , 807.38252, 216.33741,   0.     ,
           0.     ,   0.     ,   1.     ,   0.     ]
```