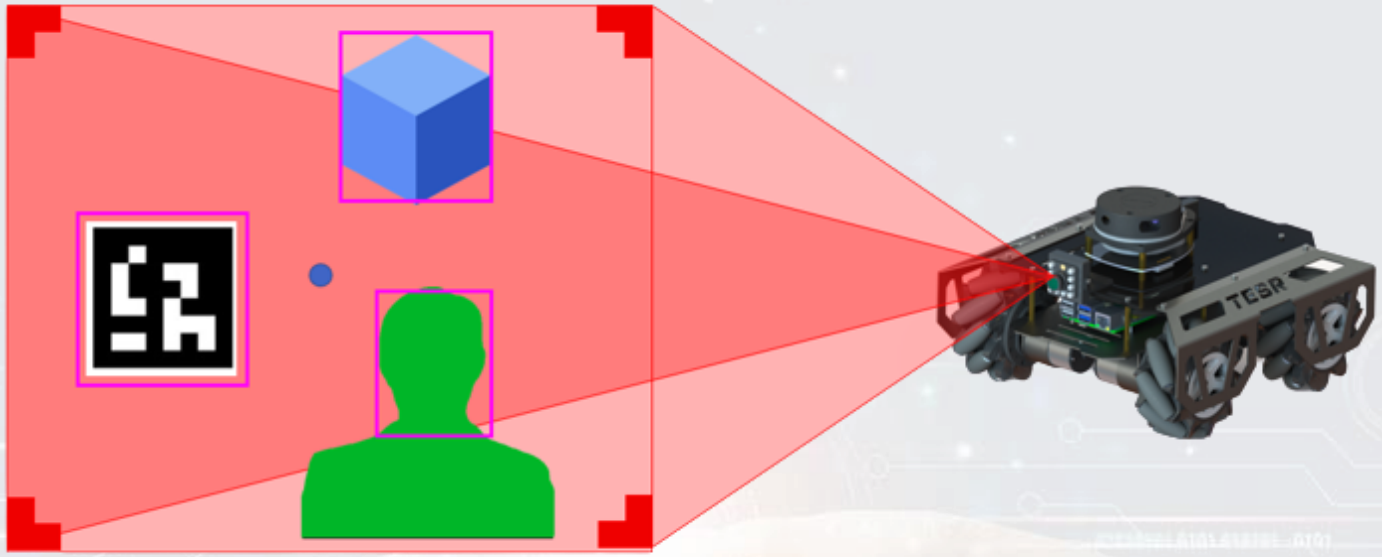


iron-X's image processing using 2D Camera By TESR

ROS2

Thai Embedded

2D Camera basic on ROS2

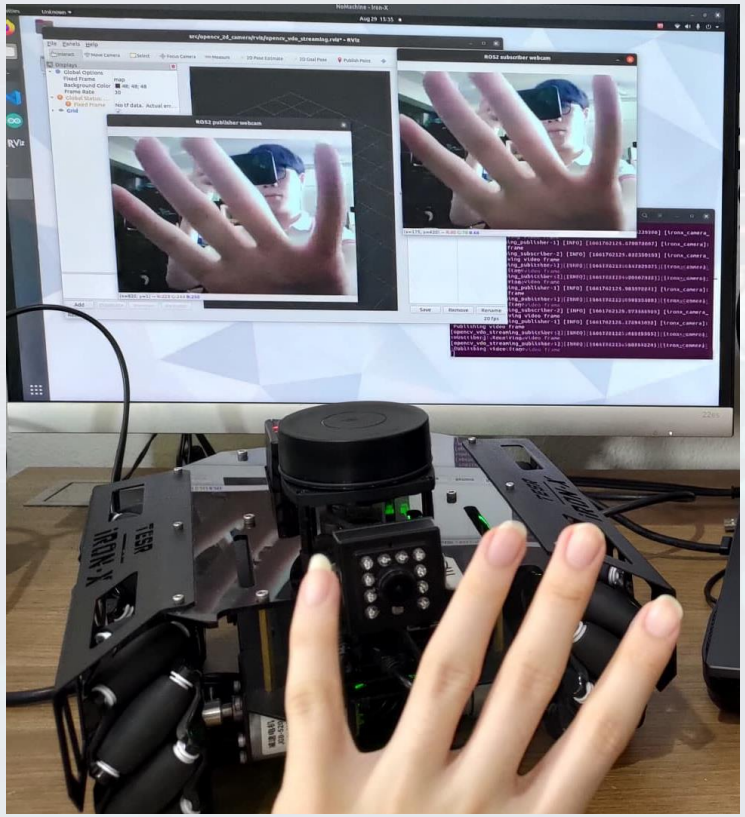
- Camera Streaming diagram

Camera Streaming

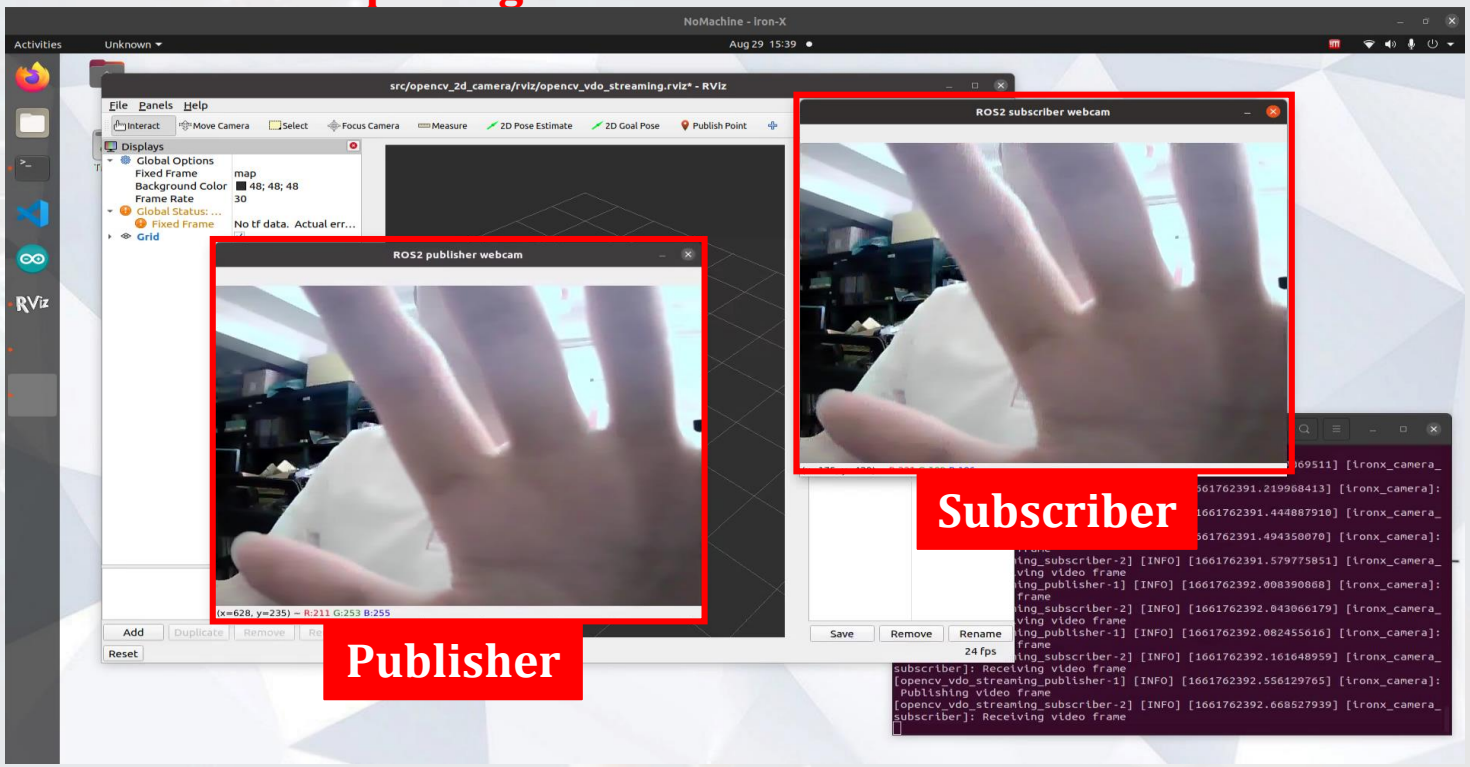
- You can run the example of camera streaming using:

```
ros2 launch opencv_2d_camera opencv_vdo_streaming.launch.py
```

Front-view of iron-X



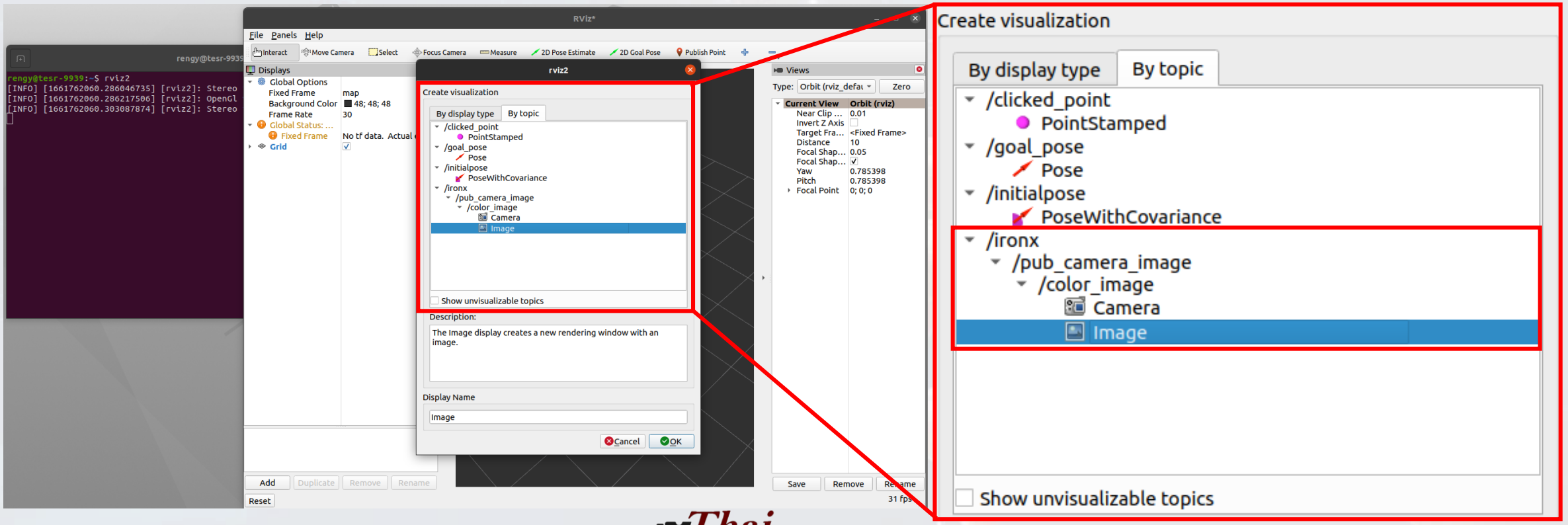
***Remote Desktop using NoMachine**



Camera Streaming

- Since, the image is streaming on the ROS server. So, you can also see the result from camera thought the Rviz on your **PC/Laptop** using:

```
rviz2
```



Thai Embedded

RESULT

Remote Desktop view → PC/Laptop view

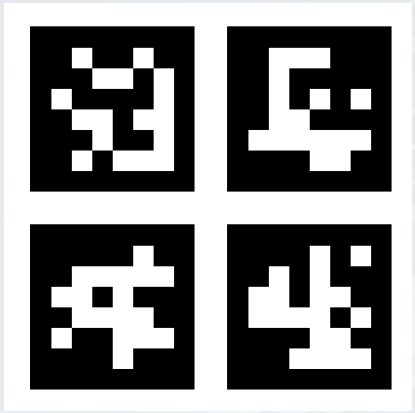
Camera Streaming

- And the RosGraph can be show by using:

```
rqt_graph
```

iron-X's image processing

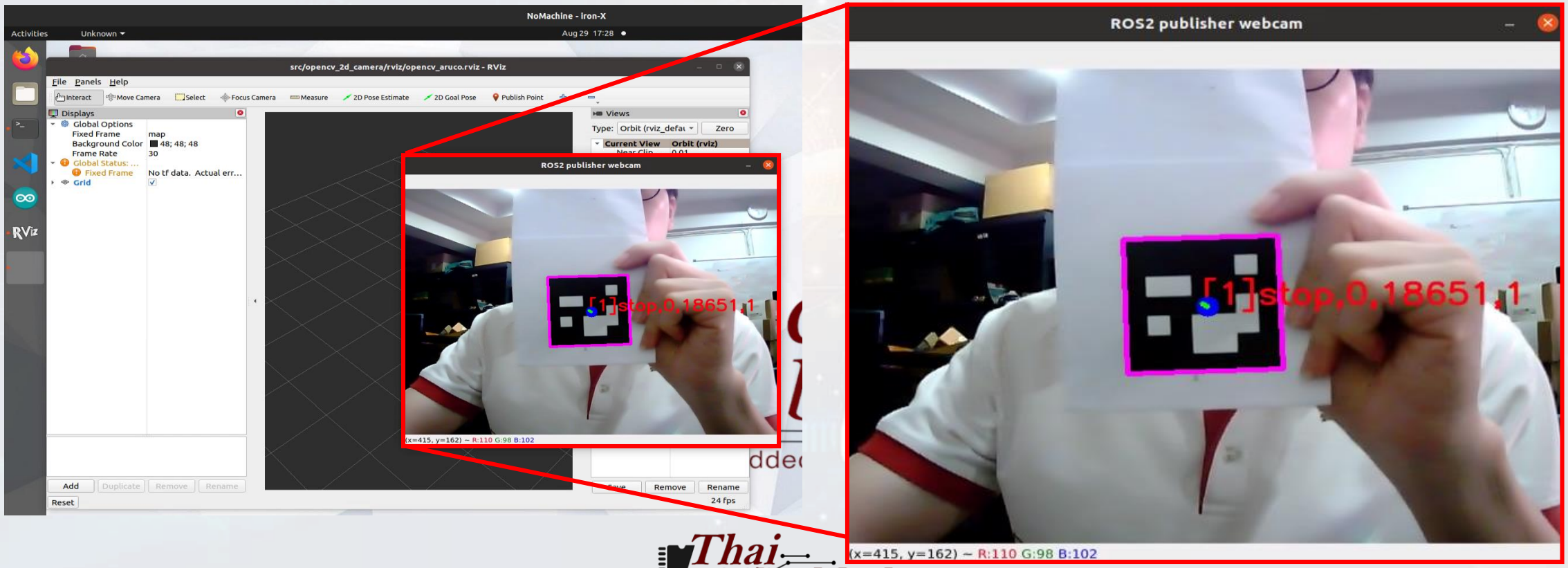
- iron-X has 4 example of image processing. Following as;
  - ARUCO detection
  - Face detection
  - Color detection
  - Lower-body detection



ARUCO detection

- You can run the example of ARUCO detection using:

```
ros2 launch opencv_2d_camera opencv_aruco.launch.py
```

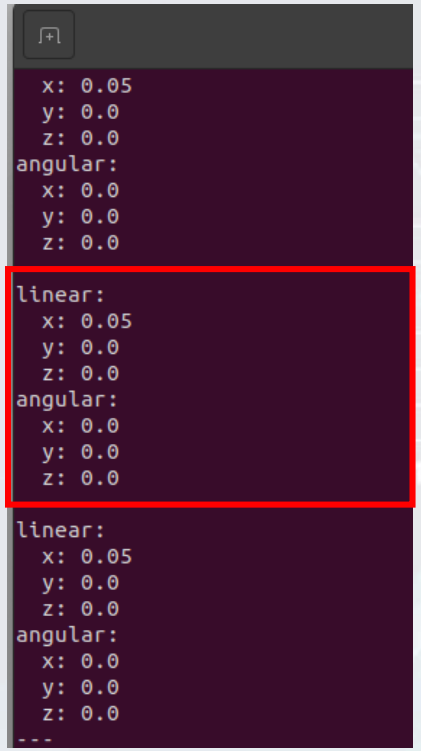


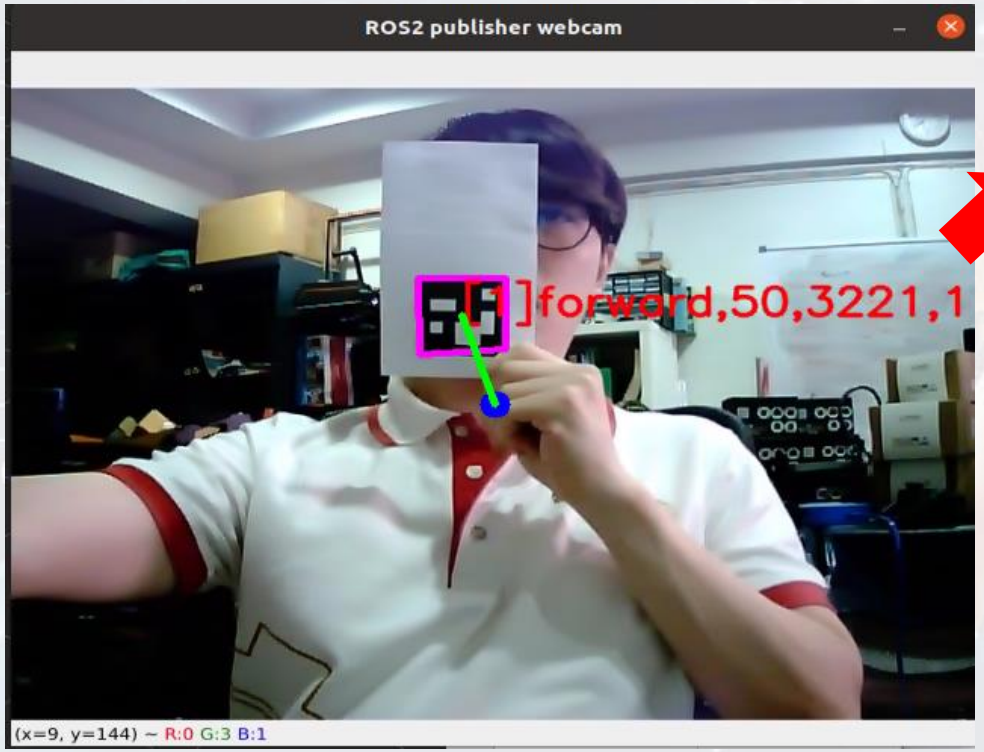
Thai Embedded

ARUCO detection

- The image processing's result transfer to /cmd_vel you can see by using:

```
ros2 topic echo /cmd_vel
```





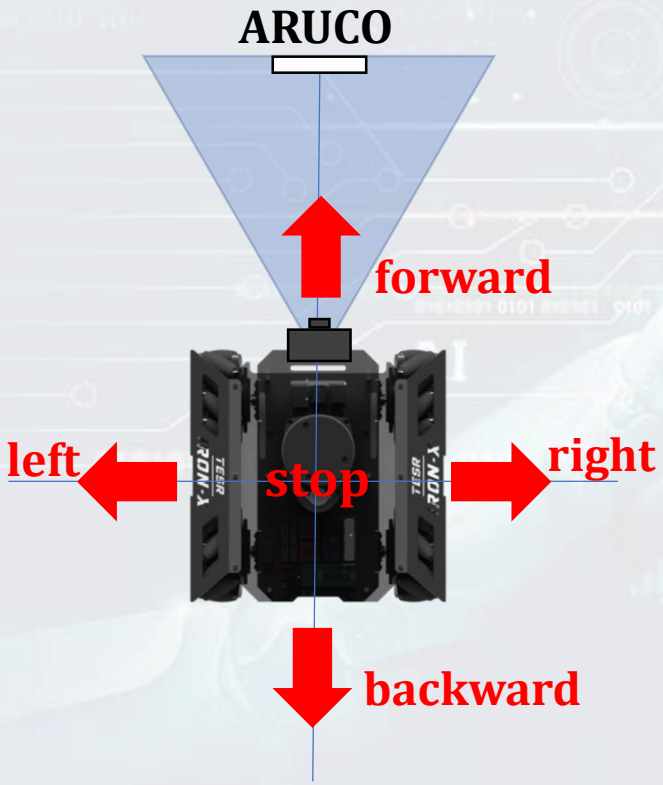
No.	Direction	Velocity	Size of interest	No.
1	forward	50	3221	1

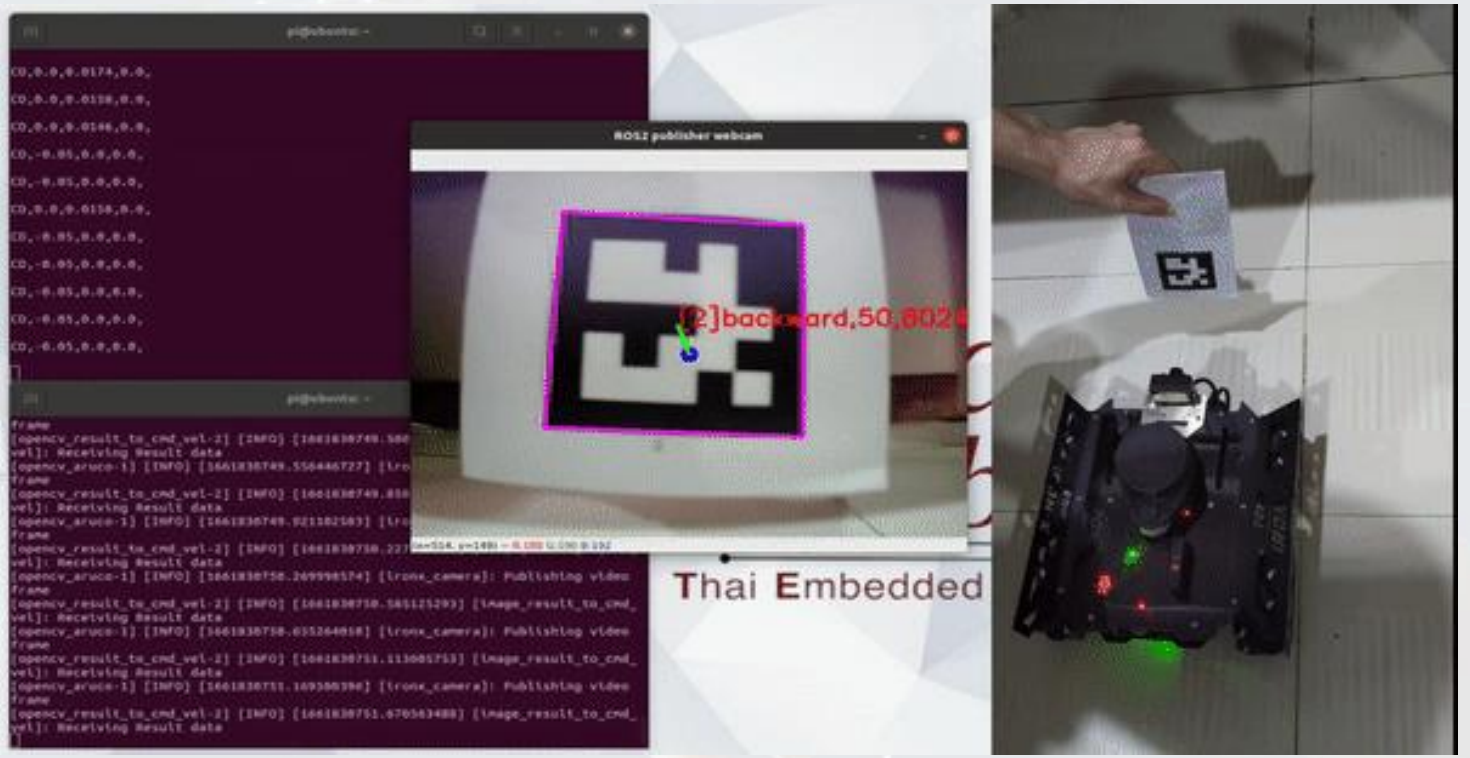
*Velocity is equal fixed value + additional speed that according to size of interest.

ARUCO detection

- Then, you can enable the iron-X's driver to move follow the /cmd_vel using:

```
ros2 run ironx_driver ironx_driver
```



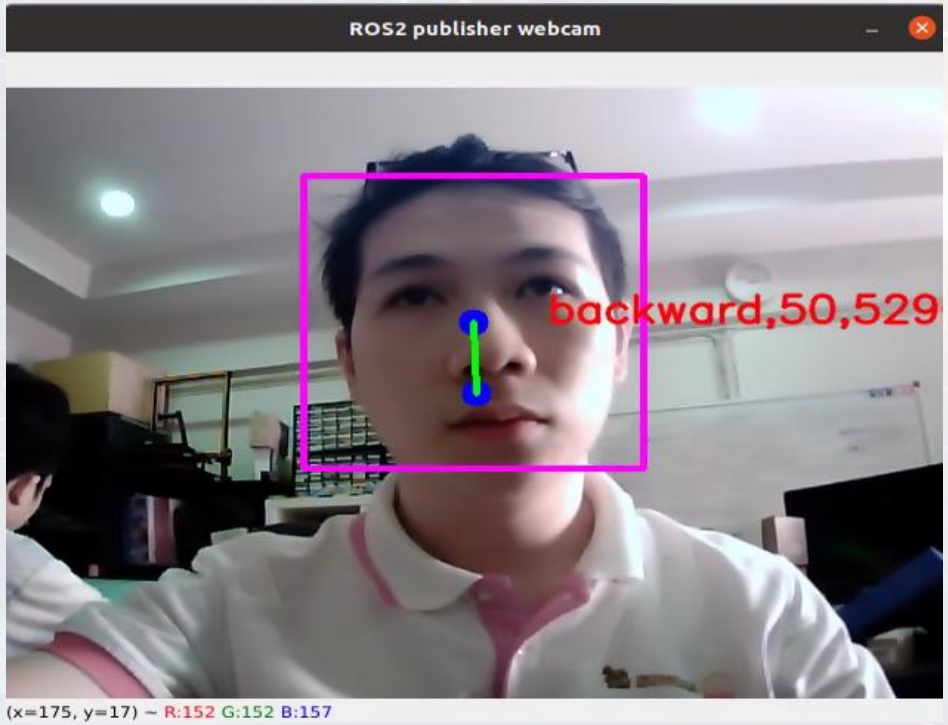


*Speed x4

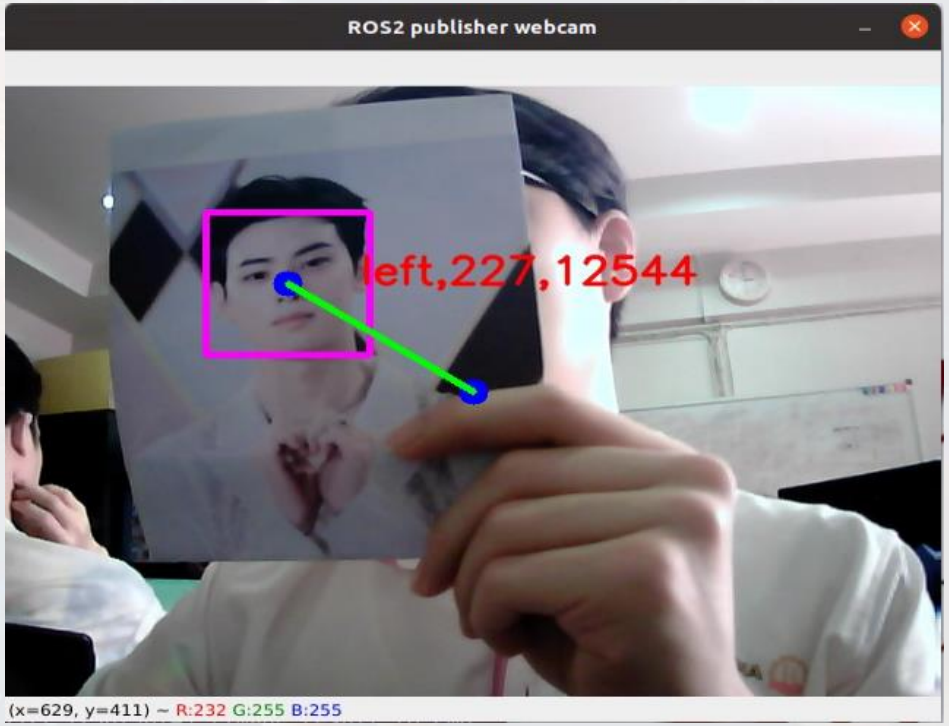
Face detection

- You can run the example of ARUCO detection using:

```
ros2 launch opencv_2d_camera opencv_face_detection.launch.py
```



Face detected on real face



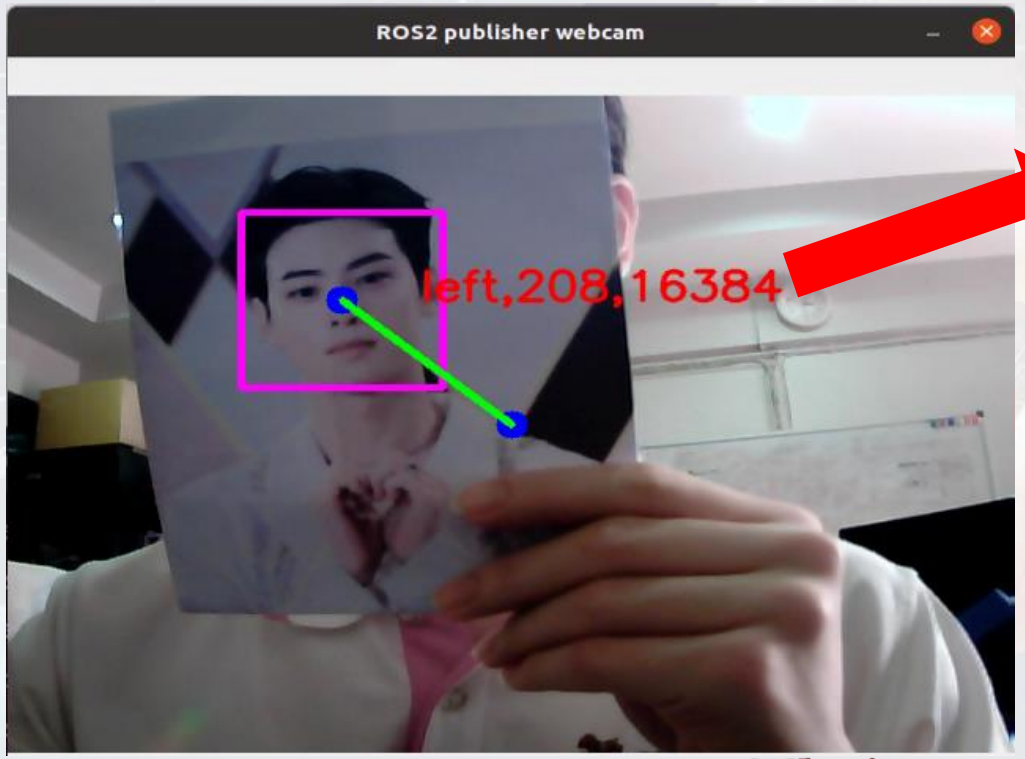
Face detected from picture

Face detection

- The image processing's result transfer to /cmd_vel you can see by using:

```
ros2 topic echo /cmd_vel
```

```
---
linear:
  x: 0.0
  y: 0.0414
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
linear:
  x: 0.0
  y: 0.0416
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
```



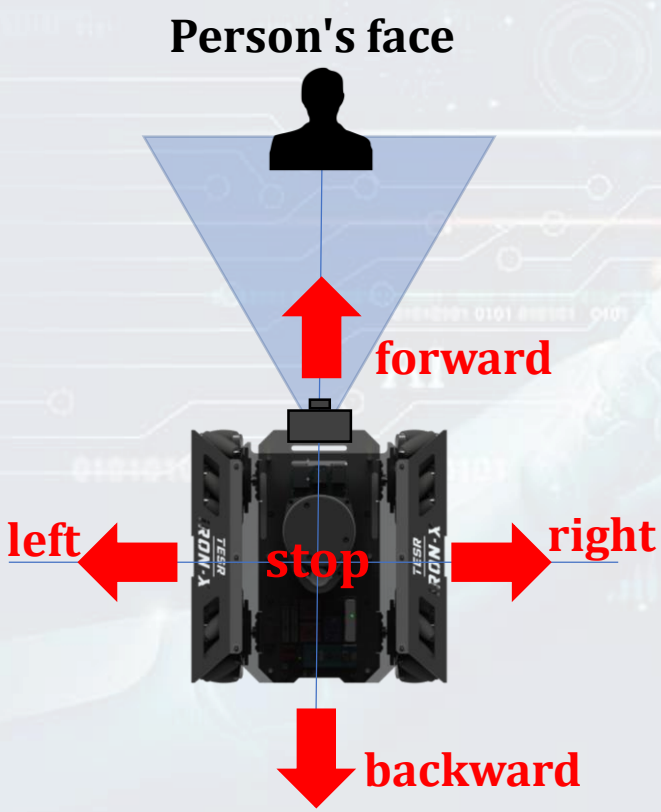
Direction	Velocity	Size of interest
left	208	16384

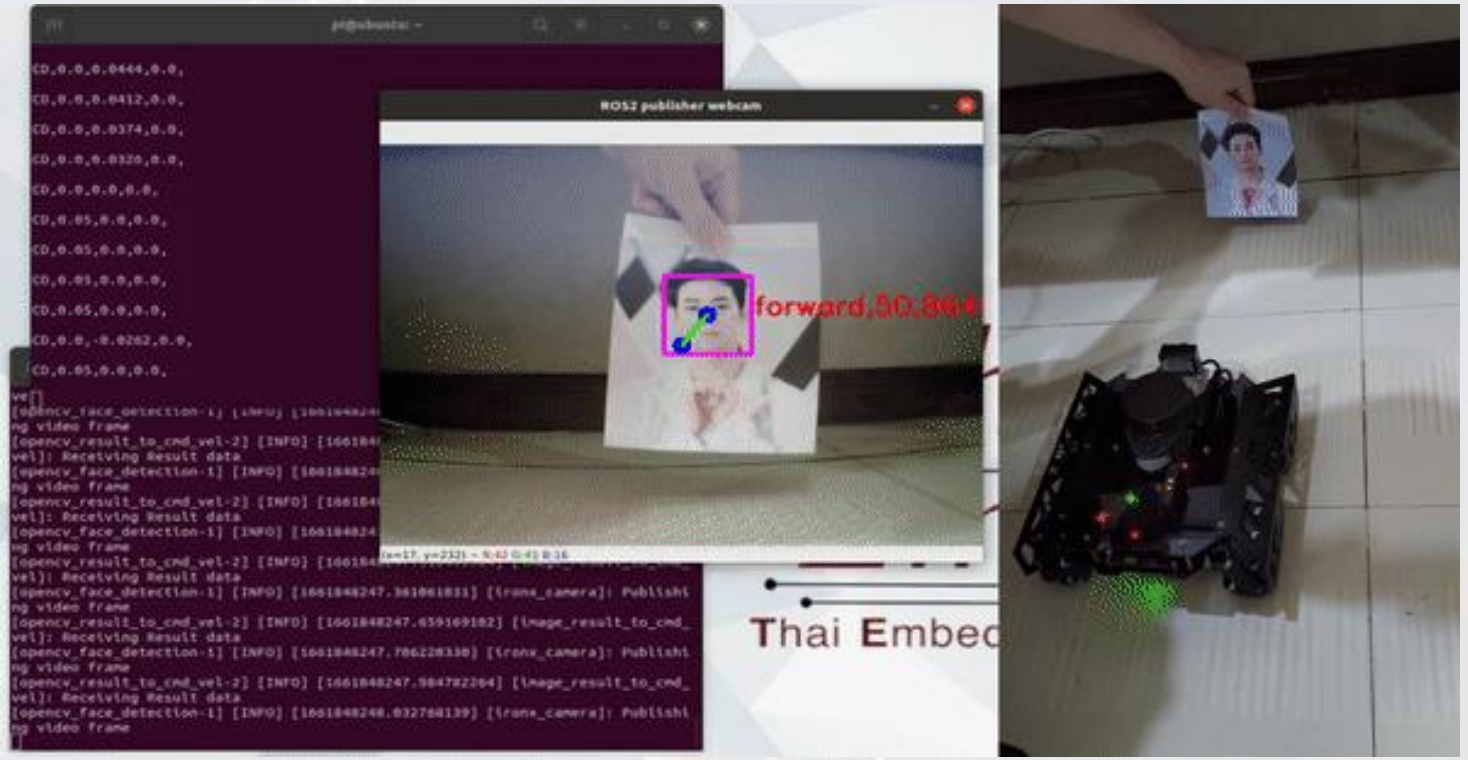
*Velocity is equal fixed value + additional speed that according to size of interest.

Face detection

- Then, you can enable the iron-X's driver to move follow the /cmd_vel using:

```
ros2 run ironx_driver ironx_driver
```



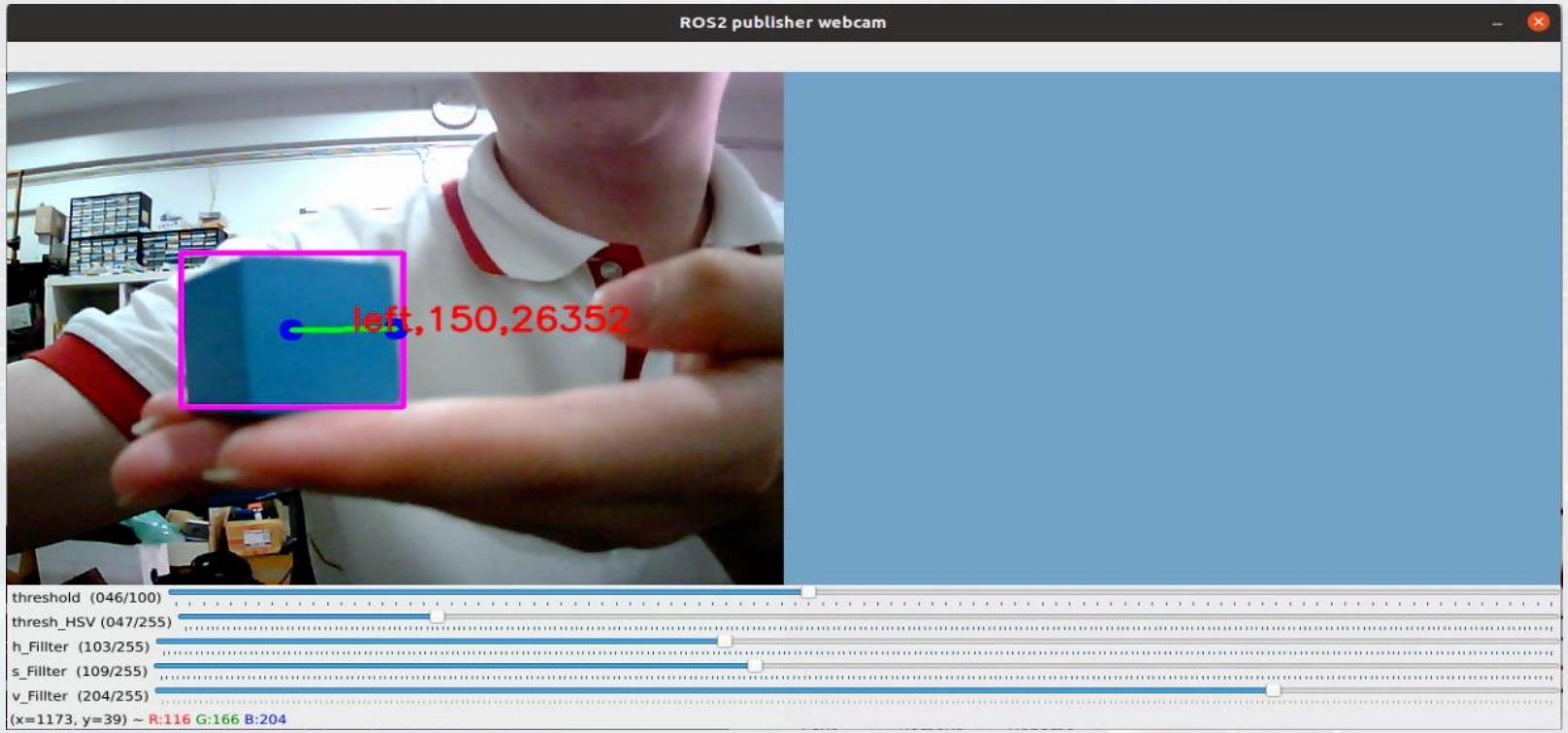


*Speed x2

Color detection

- You can run the example of ARUCO detection using:

```
ros2 launch opencv_2d_camera opencv_color_detection.launch.py
```

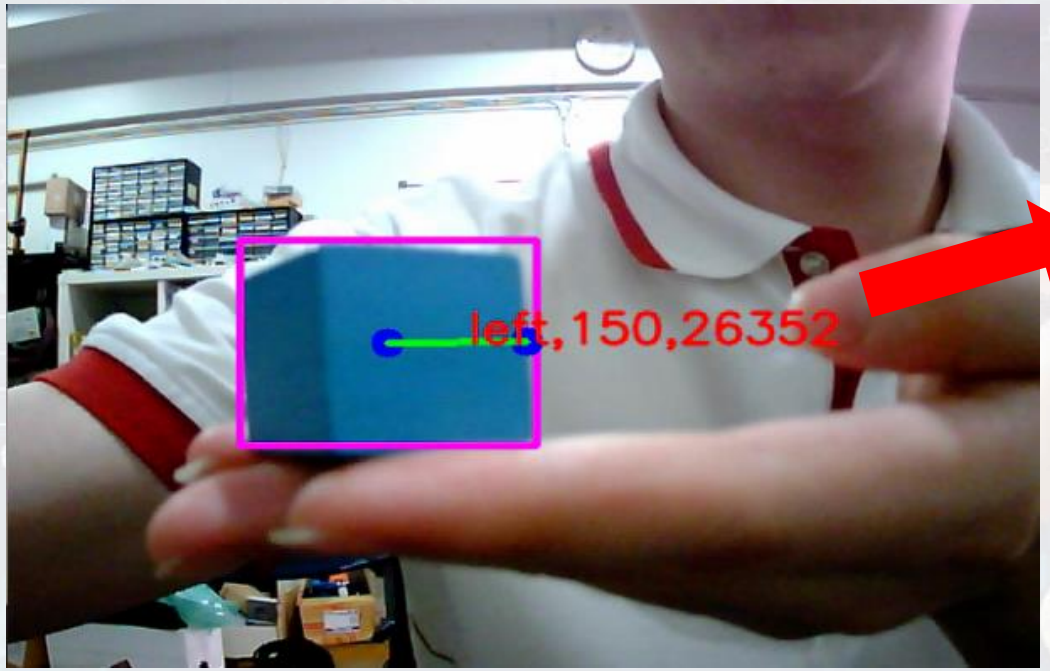


Color detection

- The image processing's result transfer to /cmd_vel you can see by using:

```
ros2 topic echo /cmd_vel
```

```
linear:
  x: 0.0
  y: 0.03
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
linear:
  x: 0.0
  y: 0.03
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
```



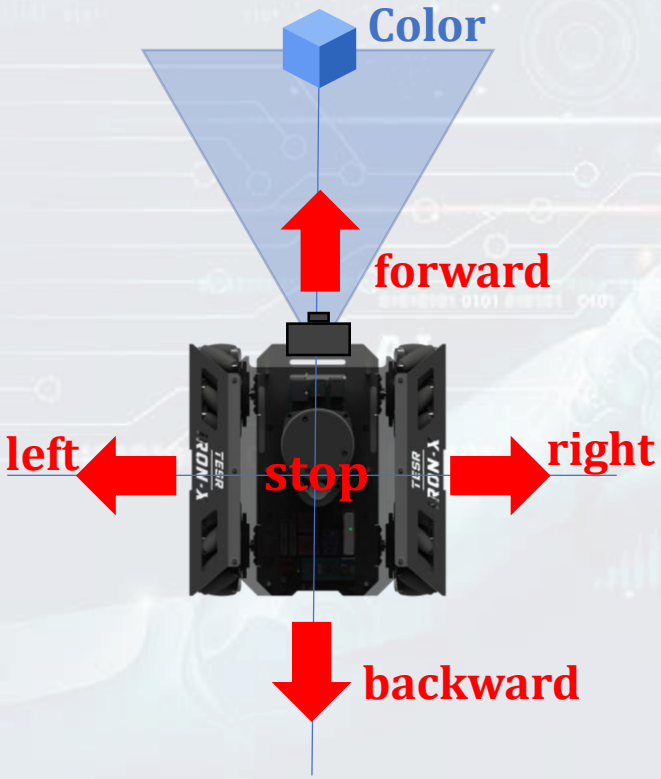
Direction	Velocity	Size of interest
left	150	26352

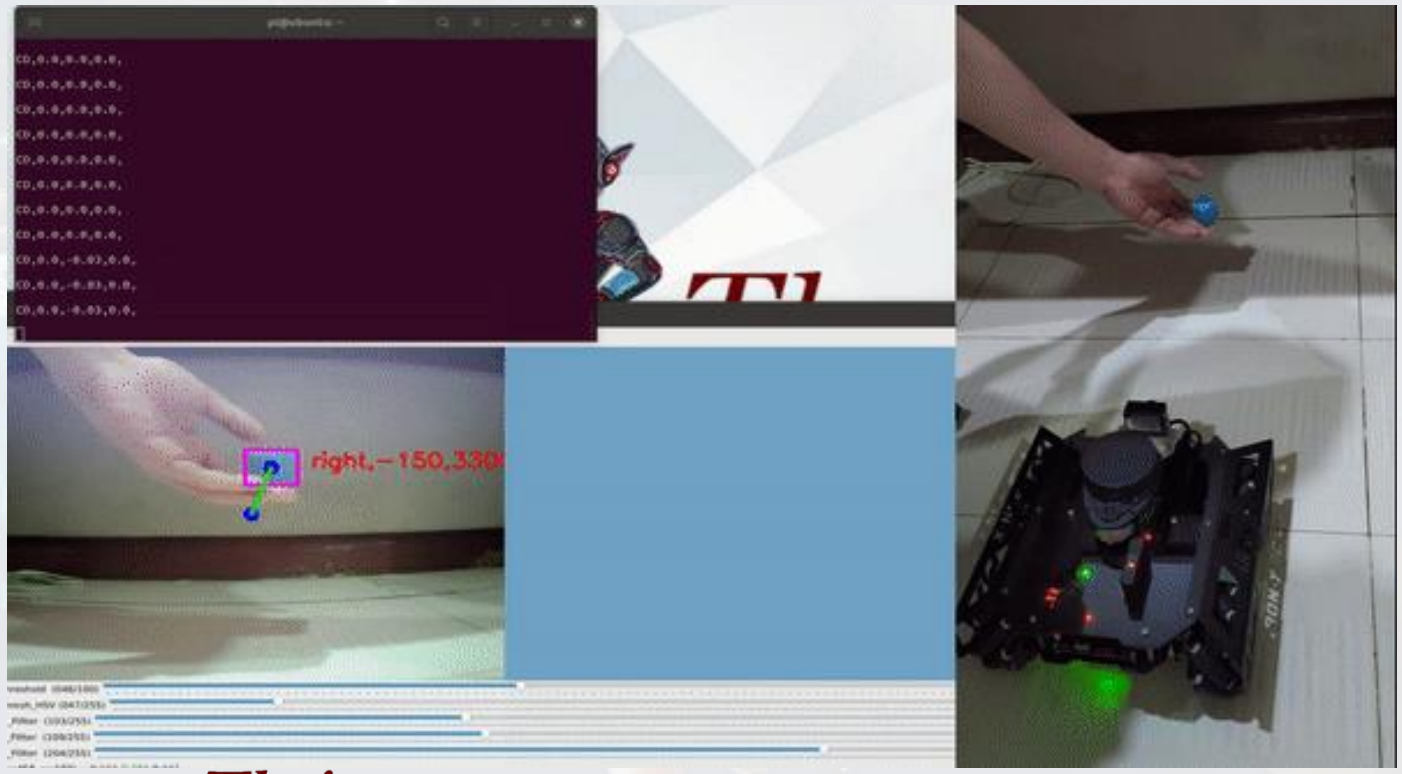
*Velocity is equal fixed value + additional speed that according to size of interest.

Color detection

- Then, you can enable the iron-X's driver to move follow the /cmd_vel using:

```
ros2 run ironx_driver ironx_driver
```





*Speed x3

Lower-body detection

- You can run the example of ARUCO detection using:

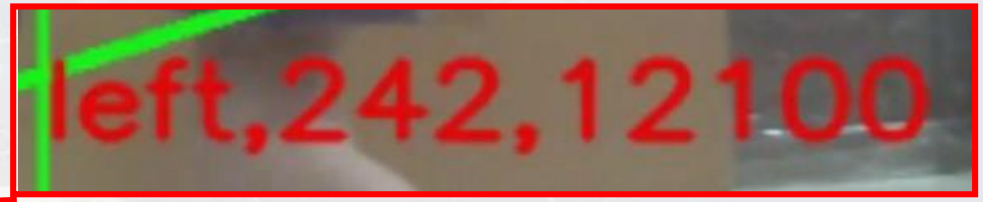
```
ros2 launch opencv_2d_camera opencv_lowerbody_detection.launch.py
```

Lower-body detection

- The image processing's result transfer to /cmd_vel you can see by using:

```
ros2 topic echo /cmd_vel
```

```
---
linear:
  x: 0.0
  y: 0.0414
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
linear:
  x: 0.0
  y: 0.0416
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
```



Direction	Velocity	Size of interest
left	242	12100

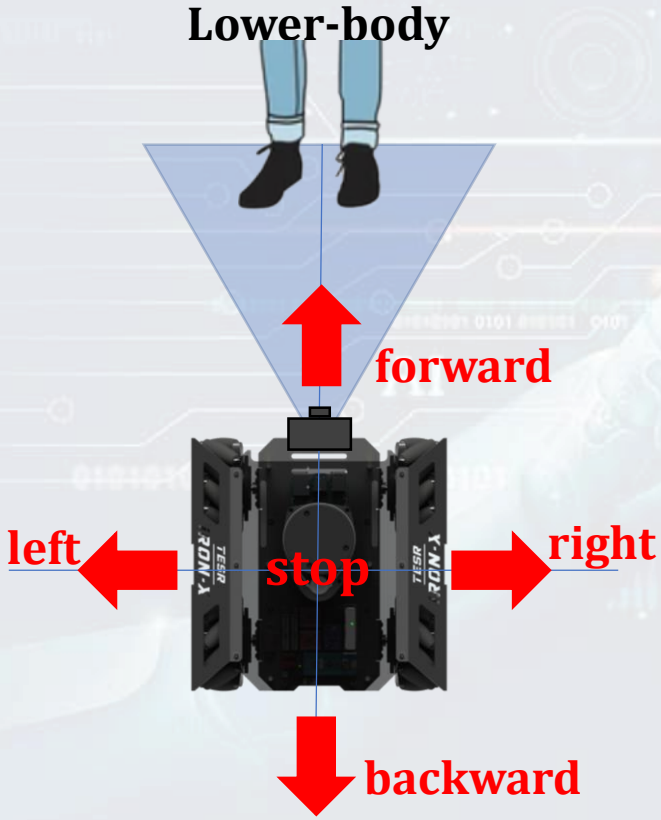
*Velocity is equal fixed value + additional speed that according to size of interest.

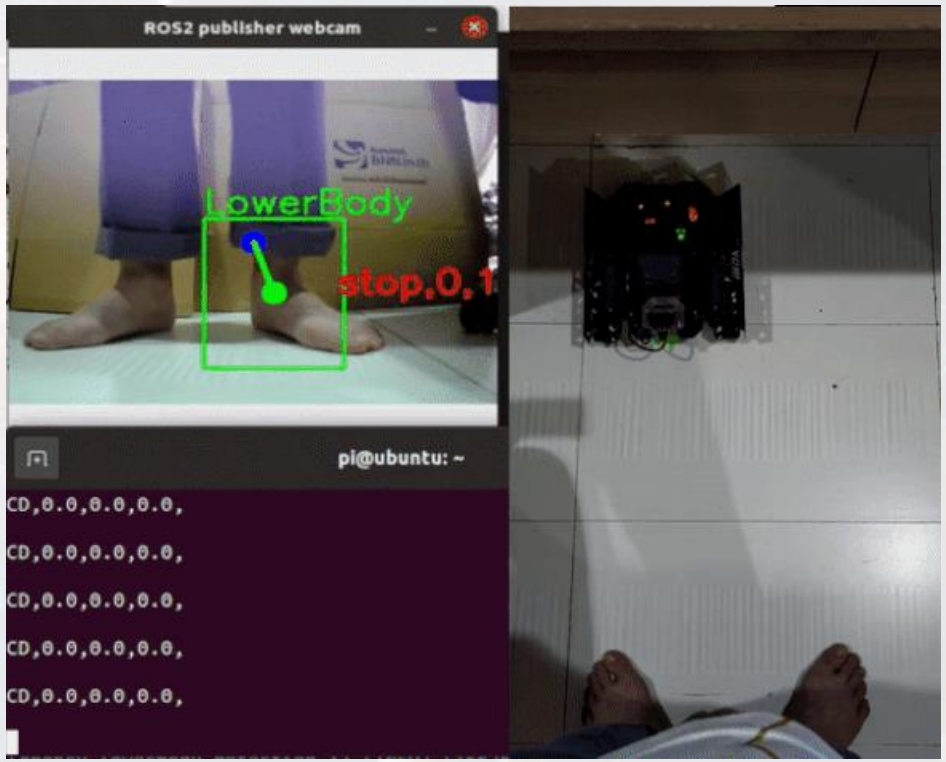
Thai Embedded

Lower-body detection

- Then, you can enable the iron-X's driver to move follow the /cmd_vel using:

```
ros2 run ironx_driver ironx_driver
```





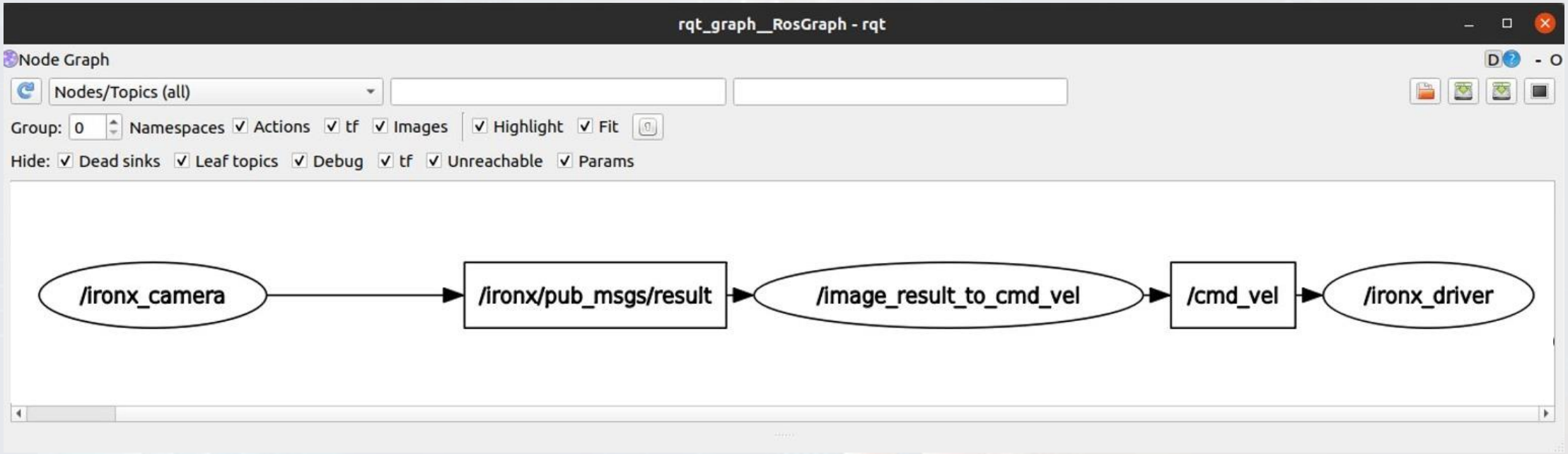
*Speed x4

Thai Embedded

RosGraph iron-X's image processing

- Let's see the RosGraph of image processing by using:

```
rqt_graph
```



Contact Us

Email: tesrshop@gmail.com
Line official Account: @ion1900z
Facebook fanpage: TESR
Tel. 082-983-7768