Part 6 About Aruco

- 핑크봇과 통신 및 연결하기 위한 세팅작업 안내

Part 6 - 목차

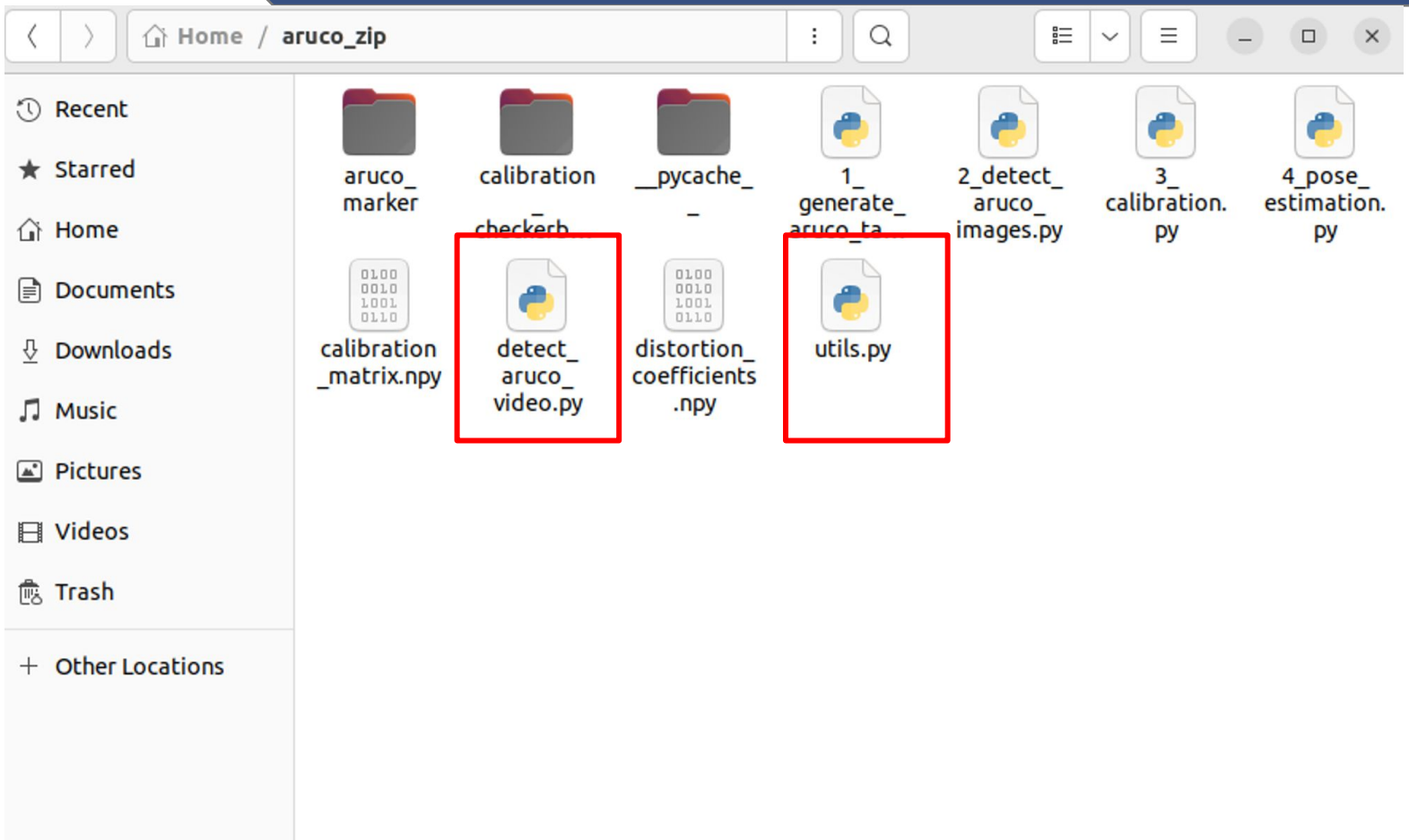
라이브러리 설치

- 해당 내용은 Ubuntu 22.04 기준입니다.
- numpy, opencv등 필요 라이브러리 설치

```
pip install numpy opencv-python
pip install opencv-contrib-python==4.6.0.66
```

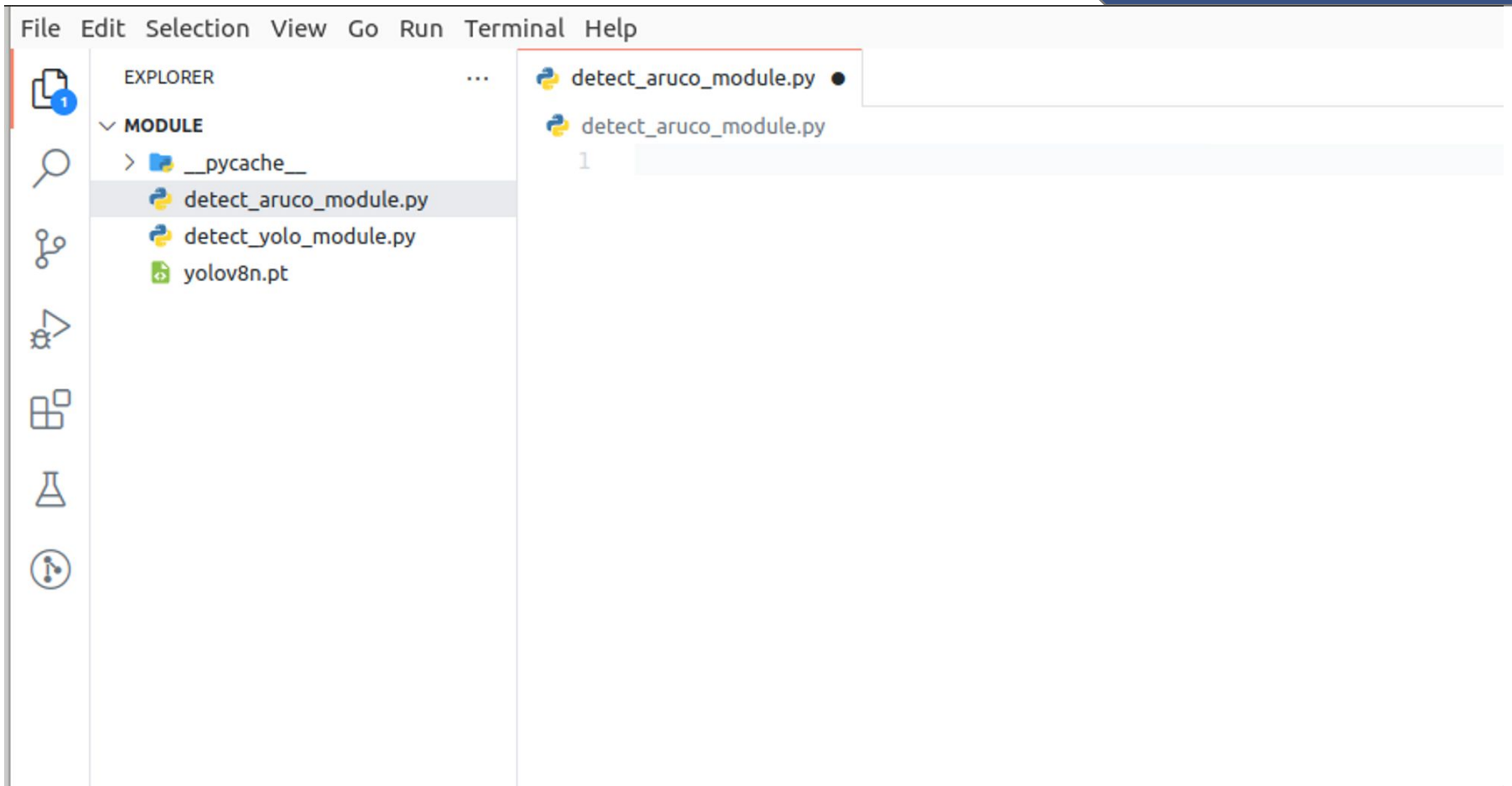
코드 Module로 만들기

'pose_QrUCo.Zip'안의 'utils.py', 'detect_aruco_video.py' 의 코드 사용



코드 Module로 만들기

각자의 workspace에 만들기



코드 Module로 만들기

필요한 모듈 불러오기

```
import rclpy

from rclpy.node import Node
from std_msgs.msg import Int32
from std_msgs.msg import Int32MultiArray
from sensor_msgs.msg import Image
from sensor_msgs.msg import CompressedImage
from rclpy.duration import Duration
from cv_bridge import CvBridge, CvBridgeError

import cv2
import sys

```

코드 Module로 만들기

클래스 생성 후 ArUco마커 딕셔너리들을 정의하기

```
class ArucoDetect(Node):
    def __init__(self):
        super().__init__('aruco_detect')

        self.ARUCO_DICT = {
                    "DICT_4X4_50": cv2.aruco.DICT_4X4_50,
                    "DICT_4X4_100": cv2.aruco.DICT_4X4_100,
                    "DICT_4X4_250": cv2.aruco.DICT_4X4_250,
                    "DICT_4X4_1000": cv2.aruco.DICT_4X4_1000,
                    "DICT_5X5_50": cv2.aruco.DICT_5X5_50,
                    "DICT_5X5_100": cv2.aruco.DICT_5X5_100,
                    "DICT_5X5_250": cv2.aruco.DICT_5X5_250,
                    "DICT_5X5_1000": cv2.aruco.DICT_5X5_1000,
                    "DICT_6X6_50": cv2.aruco.DICT_6X6_50,
                    "DICT_6X6_100": cv2.aruco.DICT_6X6_100,
                    "DICT_6X6_250": cv2.aruco.DICT_6X6_250,
                    "DICT_6X6_1000": cv2.aruco.DICT_6X6_1000,
                    "DICT_7X7_50": cv2.aruco.DICT_7X7_50,
                    "DICT_7X7_100": cv2.aruco.DICT_7X7_100,
                    "DICT_7X7_250": cv2.aruco.DICT_7X7_250,
                    "DICT_7X7_1000": cv2.aruco.DICT_7X7_1000,
                    "DICT_ARUCO_ORIGINAL": cv2.aruco.DICT_ARUCO_ORIGINAL,
                    "DICT_APRILTAG_16h5": cv2.aruco.DICT_APRILTAG_16h5,
                    "DICT_APRILTAG_25h9": cv2.aruco.DICT_APRILTAG_25h9,
                    "DICT_APRILTAG_36h10": cv2.aruco.DICT_APRILTAG_36h10,
                    "DICT_APRILTAG_36h11": cv2.aruco.DICT_APRILTAG_36h11
                    }
```

우리가 사용할 마커 종류

코드 Module로 만들기

이미지를 받는 subscriber와 detect 정보 publisher 만들기

```
41
42
43      self.aruco_subscriber = self.create_subscription(
44          CompressedImage,
45          'image_raw/compressed',
46          self.CvImage,
47          10 )
48
49      self.aruco_publisher = self.create_publisher(
50          Int32MultiArray,
51          '/detect_aruco_num',
52          10
53      )
54
55      self.bridge = CvBridge()
```

CvBridge도 같이 만들어 준다

코드 Module로 만들기

로봇의 카메라로부터 받아오는 이미지를 실행하는 함수 만들기

```
55          self.bridge = CvBridge()
56
57      def CvImage(self, msg):
58          try:
59              self.image = self.bridge.compressed_imgmsg_to_cv2(msg, "bgr8")
60          except CvBridgeError:
61              print("CvBridge is not working, check plz")
62
```

코드 Module로 만들기

아로커마커 정보 화면에 그려주는 함수 만들기

```python


    def aruco_display(self, corners, ids, rejected, image):

        if len(corners) > 0:
            # ArUco 마커의 ID 목록을 1차원 배열로 변환
            ids = ids.flatten()

            for (markerCorner, markerID) in zip(corners, ids):
                # 마커 모서리 추출
                corners = markerCorner.reshape((4, 2))
                (topLeft, topRight, bottomRight, bottomLeft) = corners
                topRight = (int(topRight[0]), int(topRight[1]))
                bottomRight = (int(bottomRight[0]), int(bottomRight[1]))
                bottomLeft = (int(bottomLeft[0]), int(bottomLeft[1]))
                topLeft = (int(topLeft[0]), int(topLeft[1]))
                # 마커 모서리 그리기
                cv2.line(image, topLeft, topRight, (0, 255, 0), 2)
                cv2.line(image, topRight, bottomRight, (0, 255, 0), 2)
                cv2.line(image, bottomRight, bottomLeft, (0, 255, 0), 2)
                cv2.line(image, bottomLeft, topLeft, (0, 255, 0), 2)
                # 마커 중심점 그리기
                cX = int((topLeft[0] + bottomRight[0]) / 2.0)
                cY = int((topLeft[1] + bottomRight[1]) / 2.0)
                cv2.circle(image, (cX, cY), 4, (0, 0, 255), -1)
                # 마커 ID 번호 쓰기
                cv2.putText(image, str(markerID),(topLeft[0], topLeft[1] - 10),
                            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
        return image
```

코드 Module로 만들기

아로커마커 인식하고 ID를 ros2 topic으로 보내는 함수 만들기

```python

def aruco_detect(self, camera, video, aruco_type):
    # ArUco 타입이 맞지 않으면 종료하기
    if self.ARUCO_DICT.get(aruco_type, None) is None :
        print(f"ArUCo tag type '{aruco_type}' is not supproted")
        sys.exit(0)

    aruco_dict = cv2.aruco.Dictionary_get(self.ARUCO_DICT[aruco_type])
    aruco_params = cv2.aruco.DetectorParameters_create()

    h, w, _ = self.image.shape

    width = 1000
    height = int(width * (h / w))
    frame = cv2.resize(self.image, (width, height), interpolation=cv2.INTER_CUBIC)
    corners, ids, rejected = cv2.aruco.detectMarkers(frame, aruco_dict,
                                                     parameters=aruco_params)

    # 마커가 인식된 경우에만 Print 하기
    if ids is not None and len(ids) > 0:
        for markerID in ids :
            print("Detect ArUco marker ID: {}".format(markerID))

            # detect_aruco_num Topic으로 보내기
            msg = Int32MultiArray()
            msg.data = [int(markerID)]
            self.aruco_publisher.publish(msg)

    # 마커 정보 화면에 그리기
    detected_markers = aruco_display(self, corners, ids, rejected, frame)
    return detected_markers
```

코드 Module로 만들기

opencv로 화면에 띄우기

```
122
123         last_iamge = aruco_detect(self, camera=True, video=None, aruco_type="DICT_4X4_50")
124
125         cv2.imshow("image_raw/compressed", last_iamge)
126         cv2.waitKey(1)
127
```

코드 Module로 만들기

마지막으로 노드 실행 함수 만들기

```python

def main(args=None):

    rclpy.init(args=args)

    aruco_detect = ArucoDetect()
    rclpy.spin(aruco_detect)
    rclpy.shutdown()


if __name__ == "__main__":
    main()

```

코드 실행

직접 코드를 실행해 보자

- [원격으로 접속해서] 로봇기동

```
ros2 launch minibot_bringup bringup_robot.launch.py
```

- [원격으로 접속해서] 카메라 실행

```
ros2 run v4l2_camera v4l2_camera_node --ros-args -p image_size:="[400,256]"
```

위의 2가지 작동한 다음

- [원격 접속 아님]위의 2번에서 생성한 .py파일 실행

```
python3 detect_aruco_module.py
```

코드 실행

직접 코드를 실행해 보자

dongu@raspdu: ~ 51x10

```
[ros2_control_node-2] [INFO] [1707148087.966863406]
 [minibot_io_controller]: on_activate...
[ros2_control_node-2] [INFO] [1707148087.967866628]
 [minibot_io_controller]: command_interface_configu
ration...
[ros2-6] Sucessfully loaded controller minibot_io_c
ontroller into state active
[INFO] [ros2-6]: process has finished cleanly [pid
2081]
```

dongu@raspdu: ~ 51x10

```
[INFO] [1707148084.490553785] [v4l2_camera]: camera
 calibration URL: file:///home/dongu/.ros/camera_in
fo/mmal_service_16.1.yaml
[ERROR] [1707148084.491398269] [camera_calibration_
parsers]: Unable to open camera calibration file [/
home/dongu/.ros/camera_info/mmal_service_16.1.yaml]
[WARN] [1707148084.491731892] [v4l2_camera]: Camera
 calibration file /home/dongu/.ros/camera_info/mmal
_service_16.1.yaml not found
```

seok@seok-B550M-AORUS-PRO-P: ~/dev_ws/module 54x22

```
seok@seok-B550M-AORUS-PRO-P:~/dev_ws/module$ python3 d
etect_aruco_module.py
Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. U
se QT_QPA_PLATFORM=wayland to run on Wayland anyway.
```

코드 실행

ros2 topic 확인

- [원격 접속 아님]Local에서 확인

```
ros2 topic list
```

- detect_aruco_num 확인

```
seok@seok-B550M-AORUS-PRO-P: ~ 54x22
seok@seok-B550M-AORUS-PRO-P:~$ ros2 topic list
/base_controller/cmd_vel_out
/base_controller/cmd_vel_unstamped
/base_controller/odom
/base_controller/transition_event
/camera_info
/detect_aruco_num
/dynamic_joint_states
/image_raw
/image_raw/compressed
/image_raw/compressedDepth
/image_raw/theora
/joint_state_broadcaster/transition_event
/joint_states
/minibot_io_controller/enable_motor
/minibot_io_controller/range
/minibot_io_controller/robot_state
/minibot_io_controller/set_lamp
/minibot_io_controller/transition_event
/parameter_events
/robot_description
/rosout
```

코드 실행

ros2 topic 받기

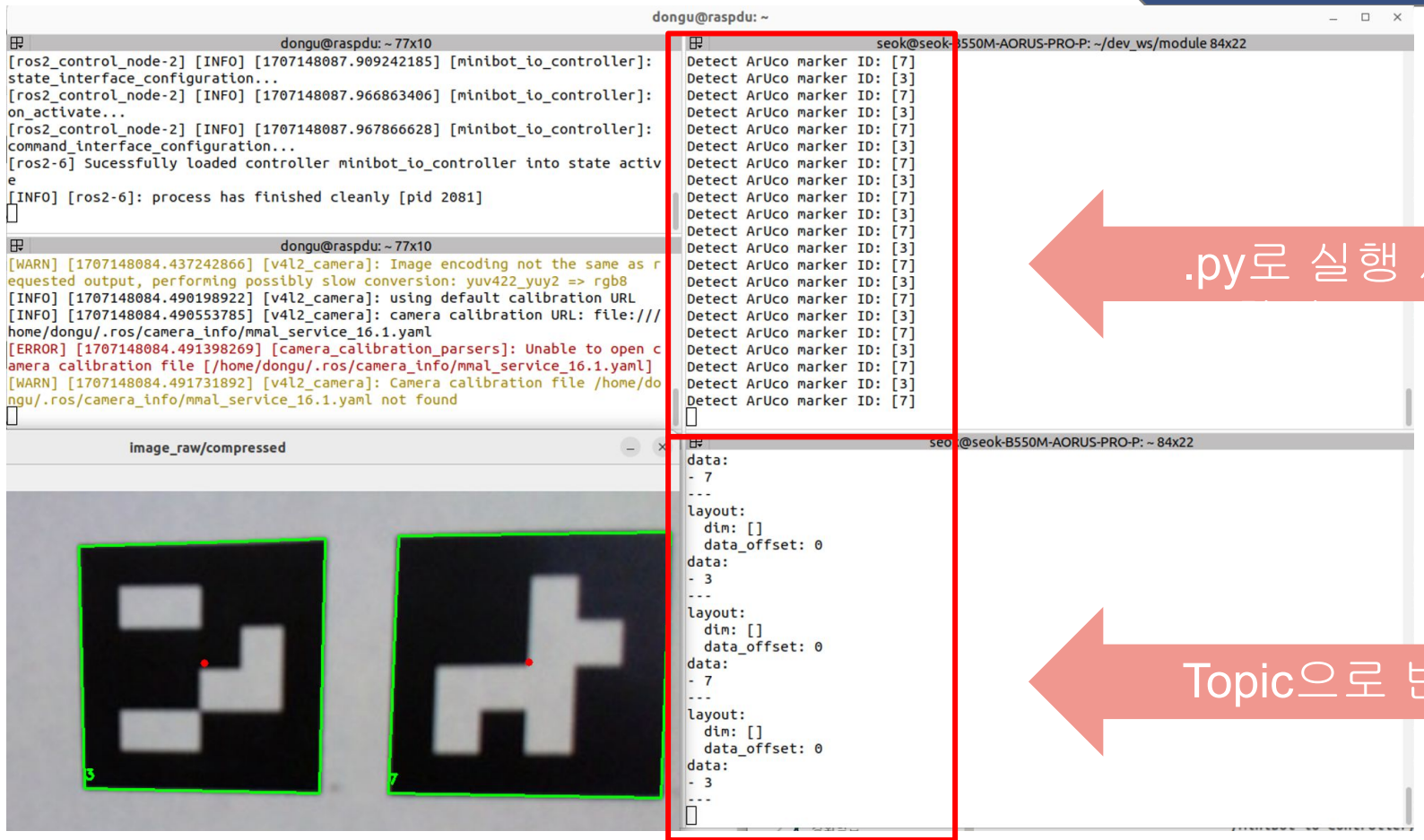
- [원격 접속 아님]

```
ros2 topic echo /detect_aruco_num
```

전북경진대회

코드 실행

아로커마커 인식하기



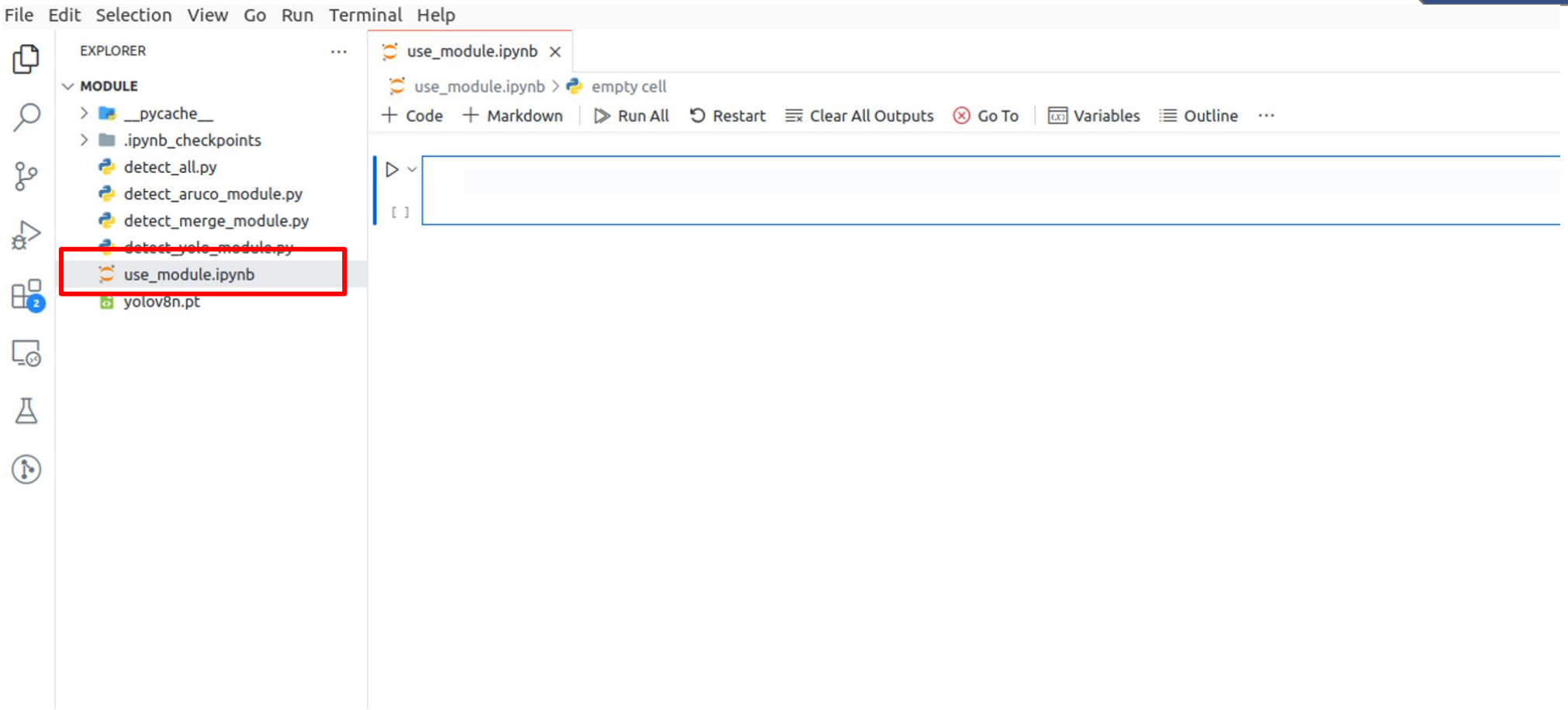
.py로 실행 시킨 화면으로

Topic으로 번호 확인

전북경진대회

.ipynb로 실행

name.ipynb



전북경진대회

.ipynb로 실행

ROS_DOMAIN_ID 설정하기

```
!echo $ROS_DOMAIN_ID
```

```
import os

# 원하는 Domain ID로 설정
domain_id = 7274  # 예시로 7274를 사용하였습니다. 원하는 값으로 변경해주세요.

# ROS_DOMAIN_ID 환경 변수 설정
os.environ['ROS_DOMAIN_ID'] = str(domain_id)
```

다른 팀과 겹치면 안된다

```
!echo $ROS_DOMAIN_ID
```

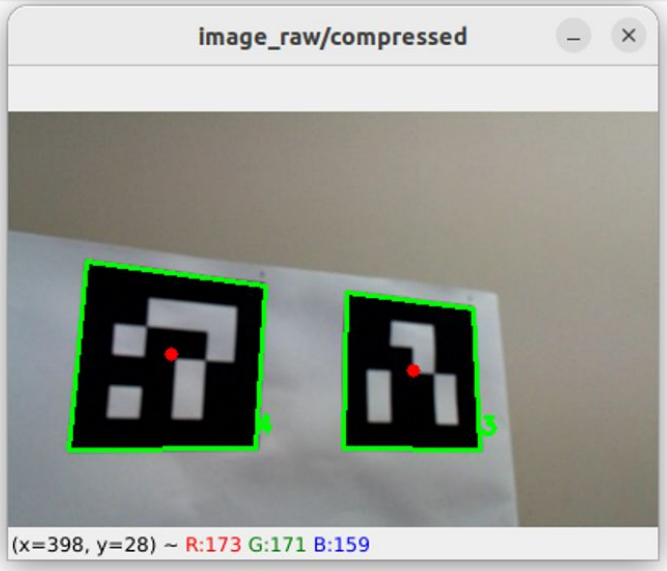
```
7274
```

.ipynb로 실행

아로커마커 모듈 불러와서 실행하기

```python
import rclpy
from rclpy.node import Node
from std_msgs.msg import Int32MultiArray
from rclpy.duration import Duration
from cv_bridge import CvBridge

from detect_aruco_module import ArucoDetect

def main():
    rclpy.init()

    aruco_detect = ArucoDetect()
    rclpy.spin(aruco_detect)

    rclpy.shutdown()

if __name__ == "__main__":
    main()
```

46.2s




```
qt.qpa.plugin: Could not find the Qt platform plugin "wayland" in "/home/seok/.local/lib/python3.10/site-packages/cv2/qt/plugins"
Detect ArUco marker ID: [3]
Detect ArUco marker ID: [3]
Detect ArUco marker ID: [4]
Detect ArUco marker ID: [3]
Detect ArUco marker ID: [4]
Detect ArUco marker ID: [3]
Detect ArUco marker ID: [4]
```