

<HW1>

과제 : QT gui 이미지 출력

1. 과제 조건

- image_recongnition -> usb_camera 패키지 활용하여 카메라 데이터 ROS 토픽으로 서브스크라이브

- QLabel 에 이미지 출력. QLabel 사이즈 640x480 으로 고정

- QPixmap 활용하여 이미지 데이터 해상도에 상관없이 640x480 크기인 QLabel 에 출력. 이미지 원본 비율 유지할 것

예시) 이미지 원본 해상도 1920x1080 -> 640x360 16:9 비율 유지한 상태로 이미지 출력(QLabel 사이즈는 반드시 고정)

keepaspectratio 사용

2. 개발과정 및 코드 구조

2 – 1 image_subscriber.cpp

이 파일은 ros2 노드로, USB 카메라의 영상을 수신해 OpenCV 창에 표시하는 역할을 한다.

usb_camera 패키지의 기본 이미지 토픽을 서브스크라이브 해서 usb 카메라 노드가 실행되고 있을때에 이미지를 받아올 수 있다.

코드 작동방식은 다음과 같다.

먼저 usb_camera 패키지의 기본 이미지 토픽을 서브스크라이브한다.

그 후에 image_callback 함수(이미지 메시지를 수신할 때마다 호출되는 함수)에서 ROS 이미지 메시지를 OpenCV 이미지로 변환한 후에 화면에 띄운다.

2 – 2 main_window.cpp

이 파일은 메인 ui 를 담당하고, 수신한 이미지를 QLabel 에 표시하는 역할을 한다.

먼저 생성자에서 label 의 크기를 주어진 크기인 640x480 로 고정시킨다.후에 rosShutdown

시그널을 close() 슬롯에 연결하여, ROS2 가 종료되면 Qt 창도 닫히도록 설정하고, imageReceived

시그널을 updateCameraImage 슬롯에 연결하여 새로운 이미지가 수신될 때마다 이미지를

업데이트한다.

updateCameraImage 함수에서는 QImage 형식으로 전달된 이미지를 QPixmap 형식으로 변환하고, 변환된 QPixmap 이미지를 KeepAspectRatio 를 사용해 QLabel 크기에 맞춰 비율을 유지하며 스케일링 한다. 그후에 스케일링 된 QPixmap 이미지를 QLabel 에 설정해 화면에 표시한다.

***QImage:** 픽셀 기반의 이미지 표현 형식으로, 이미지의 개별 픽셀을 수정하거나 접근하는 데 유리하다.

***QPixmap:** Qt 에서 주로 화면에 표시하기 위한 형식으로, 그래픽 하드웨어에 최적화되어 있어 QLabel 같은 위젯에 쉽게 사용할 수 있다.

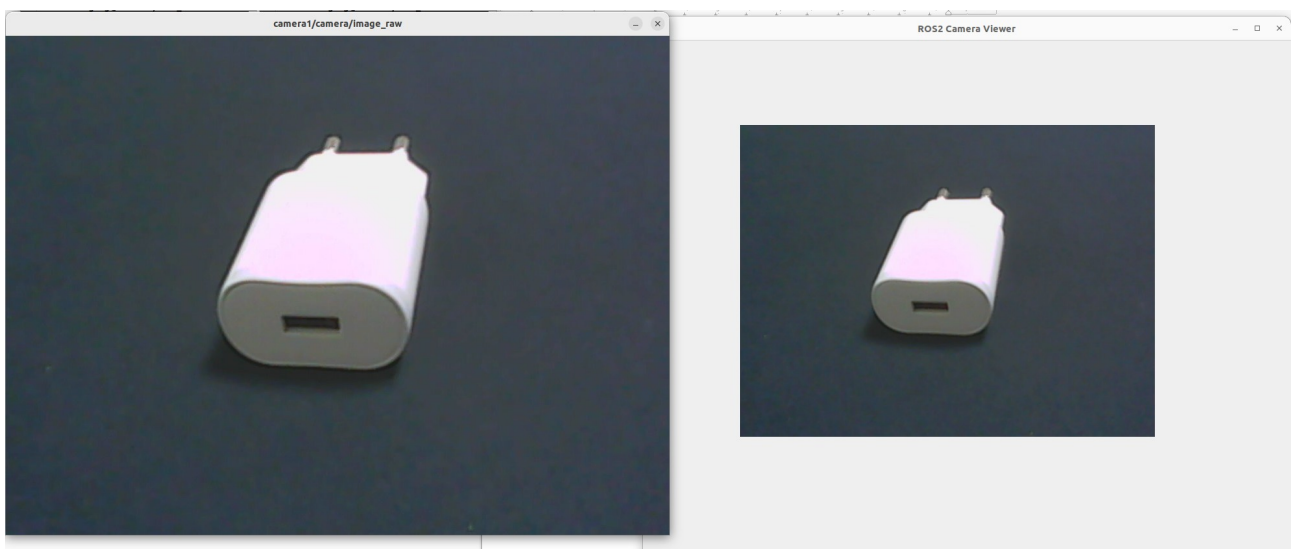
2 – 3 qnode.cpp

이 파일은 ROS2 와 Qt 간의 연결을 담당하는 파일로, Ros2 메시지를 구독하고 이미지를 변환해서 Qt 신호로 보낸다.

먼저 Ros2 를 초기화하고 노드를 생성한다. 그 후에 image_raw 토픽을 서브스크라이브해서 카메라에서 이미지를 수신한다. 이 수신된 이미지를 cv_bridge 를 이용해 변환하고, imageReceived 시그널로 방출해서 Qt UI 로 전달하게 된다.

3. 실행

실행 화면은 이와 같다.



4. 배운점

여러 이미지 처리 관련 함수에 대해 배울 수 있었고, 이미 생성되어 있는 패키지에 구독해서 불러온 정보들을 활용할 수 있는 법에 대해 알게 되었다.