Week_1

목차:

개념을 설명하는 장 부분과 코드를 설명하는 부분으로 나누어서 구성해 보았습니다.
학습 목적으로 작성한 것이라 주관적인 요소도 많이 들어가있고 각색을 한 부분도 많이 있습니다.
이를 감안해서 참고해주시면 되겠습니다.

▶1장 ROS2 소개

학습 소스: ROS2로 시작하는 로봇 프로그래밍
(표윤석, 임태훈 지음)

1. ROS2는 운영 체제일까?

ROS은 Robot Operating System의 약자로, 말 그대로 운영 체제라는 의미이다.
하지만 ROS는 정말로 컴퓨터 하드웨어와 사용자 애플리케이션 간의 인터페이스를 제공하는 OS(Operating System)일까?

답은 아니다. ROS는 컴퓨터의 여러 자원들을 관리한다기보다, 필요한 여러 기능들을 라이브러리의 형태로 제공하는 미들웨어, 또는 메타 운영체제에 더 가깝다고 볼 수 있다.

2. ROS2의 구성

ROS2는 28p의 그림 1-1에 나와있는 그림처럼 400여 개의

ROS2 공통 패키지(Common Packages)로 이루어져 있다.

Common Packages에는

- **빌드 시스템**
- ROS인터페이스 관리 패키지
- **토픽, 서비스,액션**에 사용되는 인터페이스
- 시뮬레이션
- 로봇 소프트웨어 개발 툴,
- **C++,python**과 같은 다양한 프로그래밍 언어를 지원하기 위한 **RCL패키지** 등이 포함되어 있다.

3. ROS2의 역사

ROS는 개발된지 오래되지 않았다.

2007년 5월 스탠퍼드 대학의 Morgan Quigley박사가 개발한 Switchyard 시스템에서 시작되었다.

ROS는 **BSD**와 **Apache License 2.0** 라이선스를 기반으로 하고있어 **누구나 수정,재사용,배포가 가능하다.**

> 💡 ROS 모임으로는 **ROScon**이 매년 열리고 있고, **Ros Meetup**이라는 이름의 다양한 커뮤니티 모임도 있다고 한다.

- ROSCon 2023은 2023년 10월 18일부터 20일까지 미국 The New Orleans Hyatt Regency에서 개최되었다.

4. ROS 또는 ROS2의 버전과 버전 주기

ROS는 2010년 1월에 ROS 1.0이 개발된 이후로 빈번한 업데이트가 이루어졌는데, 이에 새로운 버전 주기를 제안하는 사용자들이 생겨나게 되었고, 사용자들의 의견을 수렴하여 2013년 Hydro Meduda버전부터는 1년 주기로 정식 버전을 릴리즈하기로 하였다. 그 시기는 5월 23일이다.

▶2장 ROS2기반 로봇 개발

1.커뮤니티 중심의 개발 문화

ROS는 커뮤니티 중심의 개발 문화를 지향한다.

- 관련 문서는 위키 형태로 되어 있다.

→ 이러한 개발 문화는 쉽게 참여한 수 있다는 장점이 있지만

→ 흩어져 있는 코드와 문서, 자료로 학습하기 어려울 수 있다는 단점이 있다고 할 수 있다.,

2.ROS2 소스 코드

ROS의 최대 장점은 오픈 소프트웨어라는 점이다.

만약에 소스코드를 이해하고, 수정하고 싶으면 ROS2를 설치할 때에

바이너리가 아닌 소스 코드의 형태로 빌드하여 사용하는 방법도 있다.

3.ROS2 문서와 커뮤니티 게시판

ROS Discourse는 ROS 커뮤니티 게시판으로

- ROS 각 버전별 정규 릴리즈
- 신규 패키지 소식
- ROS TSC(Technical Steering Committee)소식과 회의
- 신규 ROS프로젝트
- 교육 및 강좌
- 구인/구직

등의 다양한 정보들을 얻을 수 있다.

▶3장 ROS2 개발환경 구축

1.개발환경

사용한 개발 환경은 다음과 같다.

- 기본 운영체제
- 로봇 운영체제
- 통합 개발환경
- 프로그래밍 언어
- 시뮬레이터

- Ubuntu 24.04 LTS
- ROS2 Foxy
- VSCODE
- C++
- Gazebo

ROS2의 설치 방법은 2가지가 있다.

- 소스 빌드
- 미리 빌드된 파일 사용
- 데비안 패키지 사용

데비안 패키지를 이용하면 필요한 종속성을 자동으로 설치해 주기 때문에 편리하다.

공식 홈페이지에 소개된 방법은 데비안 패키지를 이용하는 방법이다.

2.ROS의 설치

ROS의 설치는 지역 설정→ 소스 설정 →ROS2 패키지 설치 → ROS2 패키지 설치 확인 → ROS 개발 툴 설치 → ROS 빌드 테스트로 진행된다.

- 지역 설정 : $ sudo apt update && sudo apt install locales 외 명령 3줄
- 소스 설정 : $ sudo apt update && sudo install curl gnupg2 lsb-release 외 2줄

→ ROS2 패키지 설치

- → setup.bash로 환경 변수 설정과 각 터미널에서 talker과 listener 설정
- ROS 개발시 반드시 필요한 소프트웨어 설치

-build-essential 외 27개 패키지 설치

- ROS2 빌드 테스트

setup.bash로 환경 설정

→ $ mkdir -p ~/robot_ws/src

(여기서 robot_ws은 자신이 만들고 싶은 워크스페이스 이름이다)

(-p 옵션은 부모 디렉토리도 같이 생성한다)

→ 워크스페이스로 디렉토리 이동

→ colcon build - -symlink-install로 빌드

(여기서 심볼릭 링크를 생성하는 것의 의미는,

바이너리와 라이브러리 파일을 시스템의 설치 경로에 심볼릭 링크로 연결하여 여러 버전의 패키지를 동시에 설치하거나 업데이트할 때, 링크만 변경하면 설치나 업데이트가 가능하게 함)

3.Run command 설정

- 별도의 장으로 설명

4.통합 개발환경(IDE)와 Extensions 설치

Extension은 대표적으로 다음과 같은 것들이 있다.

- C/C++
- ROS
- URDF
- Colcon Tasks
- XML Tools
- YAML
- Markdown All inOne
- Highlight Trailing White Spaces
- Better Comments

vscode의 개발환경 설정

setting.json

- VSCode의 (사용자별) 글로벌 환경설정 파일이다.

c_cpp_properties.json

- C/C++ 관련 설정이다.

VSCode에는 외부 프로그램을 CLI와 연동하는 기능이 있고, 이를 Task라고 한다.

▶ code1-1 / Run commands 설정

요약 설명:

- bashrc에서 사용하는 명령어는 셸 명령어이다.
- alias: 사용자가 정의한 명령어를 다른 이름으로 사용할 수 있게 해주는 명령어이다.
- export명령어는 변수를 환경변수로 설정하는 명령어이다.
    - export 명령어를 .bashrc파일에서 사용하고 환경변수로 등록하면 여러 경로를 편리하게 환경 변수로 설정할 수 있다.
- source /opt/ros/<version>/setup.bash 파일을 환경변수에 등록하면 터미널 실행시 환경 변수가 자동으로 등록된다.
- 환경변수 등록 + 이 내용을 setup.bash의 끝에 붙혀넣은 후에 testpub, testsub 명령어를 실행하면 간단하게 talker와 listener을 바로 시연해볼 수 있다.

```
# export RCUTILS_CONSOLE_OUTPUT_FORMAT='[{severity}] [{name}]: {message}
# ({function_name}() at {file_name}:{line_number})'
    # 이 코드는 ROS2에서 로그를 출력할 때 어떤 형식으로 출력할지를 설정하는 코드이다.

export RCUTILS_CONSOLE_OUTPUT_FORMAT='[{severity}]: {message}'
    # ROS2에서는 로그를 출력할 때 버퍼링을 하지 않고 바로 출력한다.
    # export명령어는 변수를 환경변수로 설정하는 명령어이다.

export RCUTILS_COLORIZED_OUTPUT=1
    # ROS2에서 로그를 출력할 때 색을 입혀서 출력한다.

export RCUTILS_LOGGING_USE_STDOUT=0
    # ROS2에서 로그 메시지를 파일로 출력한다.
# export RCUTILS_LOGGING_USE_STDOUT=1
    # ROS2에서 로그 메시지를 터미널로 출력한다.

export RCUTILS_LOGGING_BUFFERED_STREAM=0
    # ROS2에서 로그 메시지를 파일로 출력한다.
# export RCUTILS_LOGGING_BUFFERED_STREAM=1
    # ROS2에서 로그 메시지를 터미널로 출력한다.

alias cw = 'cd ~/robot_ws'
    # cw라는 명령어를 입력하면 ~/robot_ws로 이동한다.
alias cs = 'cd ~/robot_ws/src'
    # cs라는 명령어를 입력하면 ~/robot_ws/src로 이동한다.
alias ccd = 'colcon_cd'
    # ccd라는 명령어를 입력하면 colcon_cd로 이동한다.

alias cb = 'cd ~/robot_ws' && 'colcon build --symlink-install'
    # cb라는 명령어를 입력하면 ~/robot_ws로 이동하고 colcon build --symlink-install를 실행한다.
alias cbs = 'colcon build --symlink-install'
    # 빌드하고 심볼릭링크를 생성한다.
alias cbp = 'colcon build --symlink-install --packages-select'
    # 선택한 패키지만 빌드하고 심볼릭링크를 생성한다.
alias cbu = 'colcon build --symlink-install --packages-up-to'
    # 선택한 패키지를 빌드하고 그 패키지에 의존하는 패키지도 빌드한다.
alias ct = 'colcon test'
    # 테스트를 실행한다.
alias ctp = 'colcon test --packages-select'
    # 선택한 패키지만 테스트를 실행한다.
alias ctr = 'colcon test-result'
    # 테스트 결과를 확인한다.
```

```
alias rt = 'ros2 topic list'
    # 토픽 목록을 확인한다.
alias re = 'ros2 topic echo'
    # 토픽의 내용을 확인한다.
alias rn = 'ros2 node list'
    # 노드 목록을 확인한다.

alias killgazebo = 'killall -9 gazebo & killall -9 gzserver & killall -9 gzclient'
    # gazebo를 종료한다.

alias af = 'ament_flake8'
    # = python에서 코드 스타일 가이드를 준수하는지 확인하는 도구이다.
alias ac = 'ament_cpplint'
    # = C++에서 구글의 코드 스타일 가이드를 준수하는지 확인하는 도구이다.

alias testpub = 'ros2 run demo_nodes_cpp talker'
    # talker 노드를 실행한다.
alias testsub = 'ros2 run demo_nodes_cpp listener'
    # listener 노드를 실행한다.
alias testpubimg = 'ros2 run image_tools cam2image'
    # cam2image 노드는 카메라 토픽을 구독해서 이미지 메시지로 변환하고
    # 이미지 메시지를 퍼블리시하는 노드이다.
alias testsubimg = 'ros2 run image_tools showimage'
    # showimage 노드는 이미지 메시지를 구독해서 이미지로 변환하고
    # 이미지를 화면에 출력하는 노드이다.
```

▶ code1-2 / 환경 설정

User setting 설정

settings.json

```
{
    "cmake.configureOnOpen": false,
▶이 설정을 활성화하면 CMake 프로젝트의 CMakeLists.txt 파일을 열 때마다
 자동으로 CMake 구성을 실행하여 프로젝트를 빌드할 수 있다.

    "editor.minimap.enabled": false,
▶Visual Studio Code 에디터에서 미니맵(minimap)을 사용하지 않는다.

    "editor.mouseWheelZoom": true,

    "editor.renderControlCharacters": true,
▶제어 문자(control characters)를 렌더링한다.
  탭(tab) 문자: →
  줄 바꿈(newline) 문자: $
  공백(space) 문자: · (온점)

    "editor.rulers" : [100],
▶텍스트 파일에서 100번째 글자 위치에 가로 줄자가 표시된다.

    "editor.tabSize": 2,
▶탭(tab) 문자의 크기를 지정한다.

    "files.association" : {
        "*.repos" : "yaml",
        "*.world" : "xml",
        "*.xacro" : "xml"
    },
▶*.repos 파일을 yaml으로 가져온다.
▶*.world 파일을 xml으로 가져온다.
▶*.xacro 파일을 xml으로 가져온다.

▶.repos : ROS 패키지의 의존성 정보를 담는다
▶.world : Gazebo의 모델 및 환경 정보를 담는다
▶.xacro : XML 매크로 파일이다.
  XML 매크로 파일에서 정의된 매크로는 다른 XML 파일에서 include할 수 있다.



    "files,insertFinalNewline": true,
    "files.trimTrailingWhitespace": true,
    "terminal.integrated.scrollback": 1000000,
    "workbench.iconTheme": "vscode-icons",
    "workbench.editor.pinnedTabSizing": "compact",
    "ros.distro": "foxy",
    "colcon,provideTask":
```

---

c_cpp_properties.json

```
{
    "configuration" : [
        {
            "name" : "Linux",
▶개발 환경의 이름
            "includePath" : [
▶추가 헤더 파일 경로
                "${default}",
                "${workspaceFolder}/**",
                "/opt/ros/foxy/include/**"
            ],
            "define" :[],
▶전처리 지시문을 포함하는 배열
            "compilerPath" : "/usr/bin/g++",
▶컴파일러 경로
            "cStandard" : "c99",
▶C++표준
            "cppStandard" : "c++14",
            "intelliSenseMode" : "linux-gcc-x64"
        }
    ],
    "version" : 4
}
```

---

task.json

```
{
    "version": "2.0.0",
    "tasks": [
▶task에는 3개의 객체가 있다.
        {
            "label": "colcon: build",
▶label: 해당 명령의 이름이다.
            "type": "shell",
            "command": "colcon build --cmake-args -DCMAKE_BUILD_TYPE=Debug",
▶실행할 명령어이다.
            "problemMatcher": [],
            "group": {
                "kind": "build",
                "isDefault": true
            }
        },
        {
            "label": "colcon: test",
            "type": "shell",
            "command": "colcon test && colcon test-result"
        },
        {
            "label": "colcon: clean",
            "type": "shell",
            "command": "rm -rf build install"
        }
    ]
}
```

launch.json

```
{
    "version": "0.2.0",
    "configurations" : [
▶디버그 구성이다.
        {
            "name" : "Debug-rclpy(debugpy)",
▶python 코드 디버그
            "type" : "python",
▶구성의 이름, 유형이다.
            "request" : "launch",
▶요청의 종류이다. launch는 디버그 시작 요청이다.
            "program" : "${file}",
▶디버그 시작 파일 경로이다.
            "console" : "integratedTerminal"
▶콘솔, 이 경우 vscode 내부 터미널 이용
        },
        {
            "name" : "Debug-rclpy(gbd)",
▶C++ 코드 디버그
            "type" : "cppdbg",
            "request" : "launch",
            "program" : "${workspaceFolder}/install/${input:package}/lib/${input:package}/${input:node}",
            "args" : [],
▶전달할 인자
            "preLaunchTask" : "colcon : build",
            "stopAtEntry" : true,
▶디버그 시작시 일시 정지할지
            "cwd" : "${workspaceFolder}",
▶현재 작업 디렉토리
            "externalConsole" : false,
            "MIMode" : "gdb",
▶디버그 프로토콜
            "setupCommands" : [
▶디버거 초기화 명령어
                {
                    "description" : "Enable pretty-printing for gdb",
                    "text" : "-enable-pretty-printing",
                    "ignoreFailures" : true
                }
            ]

        }
    ],
    "inputs" : [
▶입력 필드이다, configuration 필드에 사용되는 값들을 설정한다.
▶${input:package}와 ${input:node}는 변수로 inputs 필드에서 정의된 값으로 대체된다.
        {
            "id" : "package",
▶package를 정의한다.
            "type" : "promptString",
▶입력 필드의 유형이다. 문자열 값을 받는다.
            "description" : "Package name",
            "default" : "topic_service_action_rclcpp_example"
▶미입력시
        },
        {
            "id" : "node",
▶node를 정의한다.
            "type" : "promptString",
```

```
            "description" : "Node name",
            "default" : "argument"
        }
    ]

}
```

4장 이제는 ROS2로

1.ROS 버전

2020년 5월 23일 마지막 ROS 버전인 Noetic이 발표되었고

2025년 5월이 Noetic의 EOL으로

현재 시점인 2023년 4월 12일 기준으로 2년 남았다.

우분투 20.04기반으로 동작되었던 첫번째 ROS2 버전인 Foxy 는 이제 EOL을 맞을 것이고

우분투 22.04 기반으로 동작되는 ROS2 Humble을 가장 많이 사용하고 있는것 같다.

올해 5월 23일에는

- Windows 10 (Visual Studio 2019)와
- Ubuntu 22.04 (Jammy)

을 Tier1 platform으로 지원하는

ROS 2 Iron Irwini가 발표할 예정이다.

첫번째 ROS2 버전인 Foxy이 EOL을 두 번째 ROS2버전인 Humble이 출시된지 1년이 되었고

마지막 ROS 버전인 Noetic의 EOL이 2년 남았으며

ROS2의 새 버전이 발표되는 지금 시점이

ROS1에서 ROS2로 갈아탈 가장 좋은 시점이 아닌가 하고 생각해본다.

감사합니다.