



Week 6

1. ROS의 시간

- ROS2는 여러 노드들이 서로 통신하며 다양한 정보들을 주고받기 때문에 해당 정보들이 발간된 정확한 시간이 필수적임
- ROS2에서 사용하는 기본 시계는 System Clock이며, 협정 세계시로 표시되어 전세계 어디서든 사용이 가능함.

System Time

- System Clock을 사용한 시간을 말함.
- `sudo ntpdate ntp.ubuntu.com` 명령어를 입력하여 특정 서버의 시간으로 동기화 할 수 있음

ROS Time

- 시뮬레이션 환경에서 시간을 조절하기 위해 많이 사용함
- `use_sim_time`이 True로 설정된 노드는 `/clock` 토픽을 구독할 때까지 시간을 0으로 초기화함

Time API

- ROS2에서 제공하는 시간과 관련된 API는 크게 `time`, `duration`, `rate`가 있음.

2. ROS2의 파일 시스템

- ROS2의 파일시스템은 ROS패키지 및 소스 코드를 검색하거나 메시지 파일, 실행 파일, 파라미터 설정 및 환경설정 파일 등을 이용할 때 사용자에게 일관된 경험을 제공하기 위해 동일 구조를 가지고 있음.

패키지와 메타패키지

- ROS2에서 소프트웨어 구성을 위한 기본 단위가 패키지로서 ROS의 응용프로그램은 패키지 단위로 개발되고 관리됨.
- 패키지는 ROS의 최소 단위의 실행 프로세서인 노드를 하나 이상 포함하거나 다른 노드를 실행하기 위한 런치와 같은 실행 및 설정 파일들을 포함하게 됨

```

• /src C/C++ 코드용 폴더
• /include C/C++ 헤더 파일용 폴더 (폴더 안에는 각 패키지 이름별 폴더로 패키지별 헤더를 구분함)
• /param 파라미터 파일용 폴더
• /launch roslaunch에 사용되는 launch 파일용 폴더
• /패키지_이름_폴더 Python 코드용 폴더
• /test 테스트 코드 및 테스트 데이터용 폴더
• /msg 메시지 파일용 폴더
• /srv 서비스 파일용 폴더
• /action 액션 파일용 폴더
• /doc 문서용 폴더
• package.xml: 패키지 설정 파일 (REP-0140, REP-0149 참고)
• CMakeLists.txt: C/C++ 빌드 설정 파일
• setup.py: 파이썬 코드 환경 설정 파일
• README: 사용자 문서, github 리포지토리의 메인에 표시된다.
• CONTRIBUTING: 해당 패키지 개발에 공헌하는 방법을 기술하는 파일
• LICENSE: 이 패키지의 라이선스를 기술하는 파일
• CHANGELOG.rst: 이 패키지의 버전별 변경 사항 모음 파일 (REP-0132 참고)

```

3. ROS2의 빌드 시스템과 빌드 툴

- ROS는 코드의 재사용성을 위하여 패키지와 노드 단위로 구성되어 있고 각 패키지는 다른 패키지와 상호 호환성을 위하여 의존성을 갖게 됨.

빌드 시스템

- ROS2에서는 새로운 빌드 시스템인 ament를 사용함.
- ament_python은 CMake를 사용하지 않음

빌드 툴

- colcon : ROS1과 ROS2 모두를 지원하기 위하여 통합된 빌드 툴
- ROS2의 기본 빌드 툴로 사용중

패키지 생성

- `ros2 pkg create [패키지명] - - build-type (빌드타입) - -dependencies (패키지1)`

4. ROS2의 패키지 파일

package.xml

- 패키지 이름, 저작자, 라이선스, 의존성 패키지 등을 포함
- 사용되는 빌드 툴, 의존성 패키지들이 모두 기술됨

CMakeLists.txt

- 파이썬패키지에는 없음
- ROS패키지를 멀티 플랫폼에서 빌드할 수 있게 해줌

setup.py

- ROS2 파이썬 패키지에서만 사용하는 배포를 위한 설정파일
- CMakeLists.txt는 사용하지 않고 이 파일을 사용함

setup.cfg

- ROS2 파이썬 패키지에서는 이 파일에 develop과 install 옵션을 설정