개발환경 구축 방법

1. 개요
2. 운영체재
3. Firmware, Middleware, Agent, Simulation
4. Ground Control
5. 영상정보처리 도구

연세대학교
Space-Y 항공팀 제어부
2025/09/25

1. 개요

해당 문서는 '항공팀 개발 환경 통일안'에서 정한 개발 환경을 구축하는 방법을 기술한다. '항공팀 개발 환경 통일안'은 작성 시점(2025/09/25) 기준 필요 요소의 가장 최신 버전을 채택하였으며 해당 문서는 각 요소의 가장 최신 버전을 설치하는 방법을 기술한다. 따라서 이후 새로운 버전이 출시되거나 개발 환경 최신화가 이루어질 경우 해당 문서의 내용이 정확하지 않을 수 있다.

운영체재(OS)	Ubuntu 22.04
Firmware	PX4 v1.16
Middleware	ROS2 Humble
Agent	Micro-XRCE-DDS-Agent v2.4.3
Compiler (C++)	g++ v11.4.0
CMake	v3.22.1
Python	Python v3.10.12
OpenCV	v4.5.4
Simulation	Gazebo Simulator "Harmonic"
Ground Control	QGC v5.0.7

<표 1> 개발환경 통일안(2025/09/25)

2. 운영체재

2-1. 개요

비행체의 제어 프로그램은 ROS2의 Node 형태로 작성된다. 따라서 현재 채택된 ROS2 Humble의 권장 실행 환경인 Ubuntu 22.04을 개발 환경의 운영체재로 활용한다.

Ubuntu는 초보자도 쉽게 사용할 수 있는 Linux Distro로, 가장 많이 사용되는 Linux Distro 중 하나이다.

2-2. 설치하기

아래의 링크를 참고하여 설치를 진행하면 된다.

https://ubuntu.com/tutorials/install-ubuntu-desktop#1-overview

'3. Creat a bootable USB stick'에서 USB의 Flash를 완료 후 Windows에서 해당 USB (일반적으로 F:\와 G:\ 디스크)를 사용할 수 없으니 포맷하라는 내용의 메시지를 띄운다. 이때 USB를 포맷하지 않도록 주의한다.

'4. Boot from USB flash drive'에서 Ubuntu를 설치하고자 하는 컴퓨터에 USB를 연결하고 컴퓨터 부팅을 진행하게 된다. 이때 자동으로 Ubuntu 설치가 진행되지 않으면 재부팅 중 부트 메뉴를 띄운다. 일반적으로 F2 혹은 F12를 활용하면 부트 메뉴에 접속할 수 있다. 이후 부팅 우선순위의 1순위를 연결한 USB로 바꿔준다.

3. Firmware, Middleware, Agent, Simulation

3-1. 시스템 개요

비행체의 제어는 Flight Controller(FC)에서 담당하게 된다. FC로는 오픈소스 제어 시스템인 Pixhawk를 사용한다. Pixhawk에는 ROS2 Node를 활용해 topic의 형태로 여러 명령을 전달할 수 있다. 이때 PX4에는 uXRCE-DDS-client가, 제어 프로그램이 탑재된 Companion Computer(CC)에는 uXRCE-DDS-agent가 실행중이어야 ROS2와 PX4 사이의 통신이 가능해진다.

개발 중인 제어 프로그램은 실(實)기체에 탑재시 기체 파손 및 사고의 가능성이 있기 때문에 시뮬레이션을 활용한다. 기체 제어의 시뮬레이션을 위해서는 FC를 개발 환경에서 재현해야하므로 개발 환경에 PX4 Firmware 및 uXRCE-DDS-agent를 설치해야한다.

3-2. 설치하기

아래의 링크를 참고하여 설치를 진행하면 된다. 터미널은 Ctrl+Alt+T로 띄울 수 있다.

https://docs.px4.io/main/en/ros2/user_guide.html#install-px4

`$ git clone https://...` 명령어를 입력할 때 직접 입력하면 github 사용자 ID를 입력하라는 문구가 뜰 수도 있다. 복사-붙이기를 활용하면 github 사용자 ID를 물어보지 않기 때문에 git 명령어 사용시 복사-붙이기를 활용할 것을 권장한다. Ubuntu 22.04 터미널에서 붙이기 단축키는 Ctrl+Shift+V이다.

해당 링크에 ROS2 Humble 설치 안내까지 포함되어 있다. ROS2 Humble 공식 문서를 참고하고 싶으면 아래의 링크를 활용하면 된다.

https://docs.ros.org/en/humble/Installation.html

해당 문서 작성 시점(2025/09/25)에서는 PX4 v1.16이 가장 최근 버전이기에 특별한 버전 지정이 없어도 통일안에 맞게 설치가 가능하다. 가장 최근이 아닌 특정 버전을 설치하고자 하면 아래의 링크를 참고하면 된다.

https://docs.px4.io/main/en/contribute/git_examples.html#get-a-specific-release

처음 링크의 안내를 모두 따랐으면 Firmware, Middleware, Agent, Simulation 및 CMake, Python이 정상적으로 설치된다. CMake와 Python은 PX4 설치 시 자동으로 알맞은 버전이 설치된다. 만약 ROS2를 Humble이 아닌 지원이 끊긴 Foxy와 같은 버전을 사용할 경우 CMake, Python 등의 최신 버전과 호환되지 않아 과거 버전으로 내려야할 수도 있다.

컴퓨터 사양에 따라 Gazebo Simulator “Harmonic”의 실행이 원활하지 않을 수 있다. 이러한 경우 Gazebo의 버전을 내려 Gazebo Classic을 설치하면 조금 더 원활하게 시뮬레이션이 실행된다. 이는 아래의 링크를 참고하여 진행하면 된다.

https://docs.px4.io/main/en/sim_gazebo_classic/#installation

3-3. (선택사항) ROS2 참조하기

새 터미널에서 ROS2 관련 명령어를 사용하기 위해서는 항상 아래의 명령을 통해 ROS2 디렉토리를 참조해야한다.

```
$ source /opt/ros/humble/setup.bash
```

개발 과정에서 ROS2 관련 명령은 항상 사용하게 될 것이기에 위의 명령을 새로운 터미널을 띄울때마다 입력하기에는 버거울 수 있다. 이때 위의 명령을 .bashrc 파일에 추가해주면 ROS2 관련 명령을 사용할때마다 입력할 필요가 없어진다.

```
$ cd
$ nano .bashrc
```

위의 명령을 실행하면 .bashrc 파일을 수정할 수 있게 된다. 가장 마지막 줄에 "source /opt/ros/humble/setup.bash"를 추가해주면 된다. 이후 Ctrl+O, Enter, Ctrl+X를 통해 수정사항을 저장한다. 현재 터미널을 닫고 새로운 터미널을 다시 열면 .bashrc 파일의 수정사항이 반영된다.

3-4. (선택사항) px4_msgs workspace 구축하기

3-4-1. 개요

PX4 제어는 px4_msgs를 통해 이루어진다. 따라서 PX4 제어를 위한 ROS2 workspace에는 항상 px4_msgs가 있어야한다. 만약 PX4 workspace가 여러개 있으면 같은 내용(px4_msgs)을 가진 파일이 불필요하게 많을 뿐만 아니라, PX4 Firmware 업데이트시 모든 workspace의 px4_msgs를 최신화해줘야하는 버거움이 있다. 따라서 px4_msgs만 있는 workspace를 구축하고 해당 workspace를 참조하여 개발을 진행하면 더욱 쾌적한 개발 경험을 할 수 있다.

3-4-2. 구축방법

```
$ mkdir -p ~/px4_ros2_ws/src/
$ cd ~/px4_ros2_ws/src/
$ git clone https://github.com/PX4/px4_msgs.git
$ cd ..
$ source /opt/ros/humble/setup.bash ←3-3을 진행한 경우 입력하지 않아도 된다.
$ colcon build
$ cd
$ nano .bashrc
```

위의 명령을 실행하면 3-3에서와 같이 .bashrc 파일을 수정할 수 있다. 가장 마지막 줄에 "source ~/px4_ros2_ws/src/install/setup.bash"를 추가하고 저장하면 된다. 이후 터미널을 재실행하면 수정사항이 반영된다.

workspace 구축 중 colcon 명령이 존재하지 않는다는 메시지가 뜰 수도 있다. 설치해주면 된다.

```
$ sudo apt update
$ sudo apt install python3-colcon-common-extensions
```

4. Ground Control

4-1. 개요

비행체를 지상에서 제어하기 위해 Ground Control(GC)가 필요하다. 이는 QGroundControl(QGC)이라는 앱을 활용해 이루어진다. 시뮬레이션 실행에서 QGC의 버전은 크게 중요하지 않다. Gazebo Classic을 활용한 시뮬레이션에서는 QGC가 아예없어도 실행이 가능하지만, Gazebo Simulation "Harmonic" 이상의 버전에서는 QGC가 있어야 시뮬레이션에서 구현된 가상 비행체의 비행이 가능하다.

4-2. 설치하기

설치는 아래의 링크를 참고하여 진행하면된다.

https://docs.qgroundcontrol.com/master/en/qgc-user-guide/getting_started/download_and_install.html#ubuntu

설치 과정에서 "`$ chmod +x ....`"를 실행하는 단계가 있다. 이 명령은 다운 받은 QGroundControl-x86_64.AppImage와 같은 디렉토리에서 진행하거나, 해당 파일의 디렉토리를 지정 후 실행하면 된다. 대부분의 경우 다운된 파일은 ~/Downloads/에 있다.

5. 영상정보처리 도구

5-1. 개요

영상정보처리를 위한 많은 라이브러리가 있다. OpenCV는 Python을 활용한 영상정보처리에서 많이 사용되는 라이브러리 중 하나이다.

5-2. 설치하기

OpenCV를 설치하는 방법은 apt설치, source 설치, pip 설치 세가지가 있다. pip 설치가 유연성은 떨어지지만 가장 쉽기 때문에 pip 설치 방법만 기술한다.

```
$ pip install opencv-python==4.5.4.60
```

이후 설치 여부 확인을 위해 python shell을 실행한다.

```
$ python3
```

그러면 아래와 같은 문구가 뜬다.

```
Python 3.10.12 (main, Aug 15 2025, 14:32:43) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information
>>>
```

아래 명령을 실행하면 OpenCV의 버전을 확인할 수 있다.

```
>>> import cv2
>>> print(cv2.__version__)
```

"`4.5.4`"라는 문구가 뜨면 성공적으로 설치가 된 것이다. 이후 "`exit()`"을 입력하여 shell을 종료할 수 있다.

“import cv2” 실행 시 아래와 같은 문구가 뜰 수도 있다.

```
A module that was compiled using NumPy 1.x cannot be run in
NumPy 2.2.6 as it may crash. To support both 1.x and 2.x
ersions of NumPy, modules must be compiled with NumPy 2.0.
Some module may need to rebuild instead e.g. with ‘pybind11>=2.12’.

If you are a user of the module, the easiest solution will be to
downgrade to ‘numpy<2’ or try to upgrade the affected module.
We expect that some modules will need time to support NumPy 2.

Traceback (most recent call last): File “<stdin>”, line 1 in <module>
  File “/home/user/.local/lib/python3.10/sit-packages/cv2/__init__.py”, line 8
, in <module>
    from .cv2 import *
AttributError: _ARRAY_API not found
Traceback (most recent call last):
  File “<stdin>”, line 1, in <module>
  File “/home/user/.local/lib/python3.10/sit-packages/cv2/__init__.py”, line 8
, in <module>
    from .cv2 import *
ImportError: numpy.core.multiarray failed to import
```

이는 OpenCV v4.5.4가 예전 버전이기 때문에 최신 NumPy 버전인 v2.0 이상과 호환이 되지 않아 발생하는 문제이다. 아래 명령을 통해 NumPy 버전을 v2.0 미만으로 내리면 문제가 해결된다.

```
$ pip install “nump<2” --upgrade –force-reinstall
```