

자율주행 자동차

자이카 ROS2 Y모델 사용설명서

2025년

자이카 Y모델

실습 장비 – 자율주행 모형차

실습 장비 – 자이카 Y모델



자이카 Y모델 소개

- 고성능 AMD 프로세서 기반 자율주행 자동차



자이카 Y모델 스펙

Processor Module

프로세서	AMD Ryzen 64bits
메모리	32GB DDR4 RAM
유선랜	2.5Gbps RJ-45 LAN
USB	A-type x4, C-type x2
저장장치	250GB

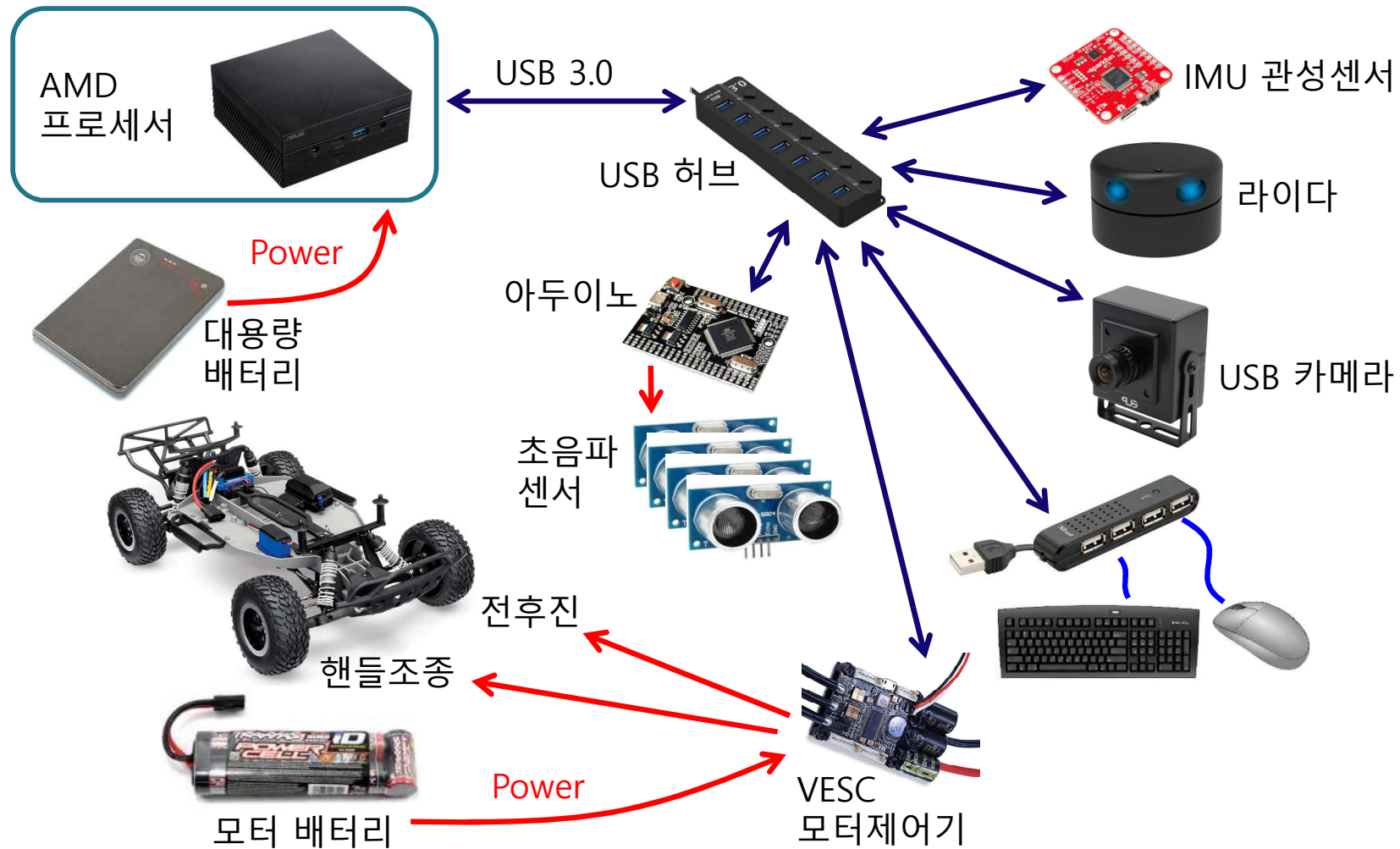
Software

운영체제	Ubuntu 22.04
ROS플랫폼	ROS2 Humble
그래픽	Radeon Graphics
프로그래밍	Python, C++
영상처리	OpenCV

Sensors + Peripherals

차체프레임	1/10 RC Car Body Frame, 55x30x25cm
구동모터	Brushless Motor
USB허브	USB 3.0 Hub, 7 Ports
모터제어기	VESC (Open Source ESC)
IMU관성센서	가속도 센서+자이로 센서+지자기 센서
USB카메라	170° 어안렌즈, UVC 1.1, 640x480 30fps
라이다	2D LIDAR, 12m Range, 5,000sample/s
무선랜	802.11ac WLAN, 외장안테나
초음파센서	HC-SR04, 2~400cm, Resolution 0.3cm
무선조종기	안드로이드 스마트폰 앱 (와이파이 접속)
메인배터리	리튬이온 Li-ion 12~19V, >20,000mAh
모터배터리	니켈수소 NiMH 8.4V, >3,000mAh

자이카 Y모델 시스템 구성도



자이카 Y모델 소프트웨어 구성



Self
Driving

Tensorflow

C++

Python

OpenCV

ROS
Packages

Speed/Steering Control

Camera

LIDAR

IMU

Ultrasonic

ROS 2

Open Source Robot OS



Linux

자이카 소프트웨어

실습 장비 – 자율주행 모형차

자이카 소프트웨어 구성



Self
Driving

Machine Learning

Python

OpenCV

ROS
Packages

Speed/Steering Control

Camera

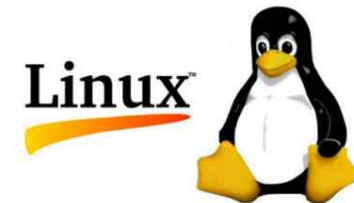
LIDAR

IMU

Ultrasonic

ROS 2

Open Source Robot OS



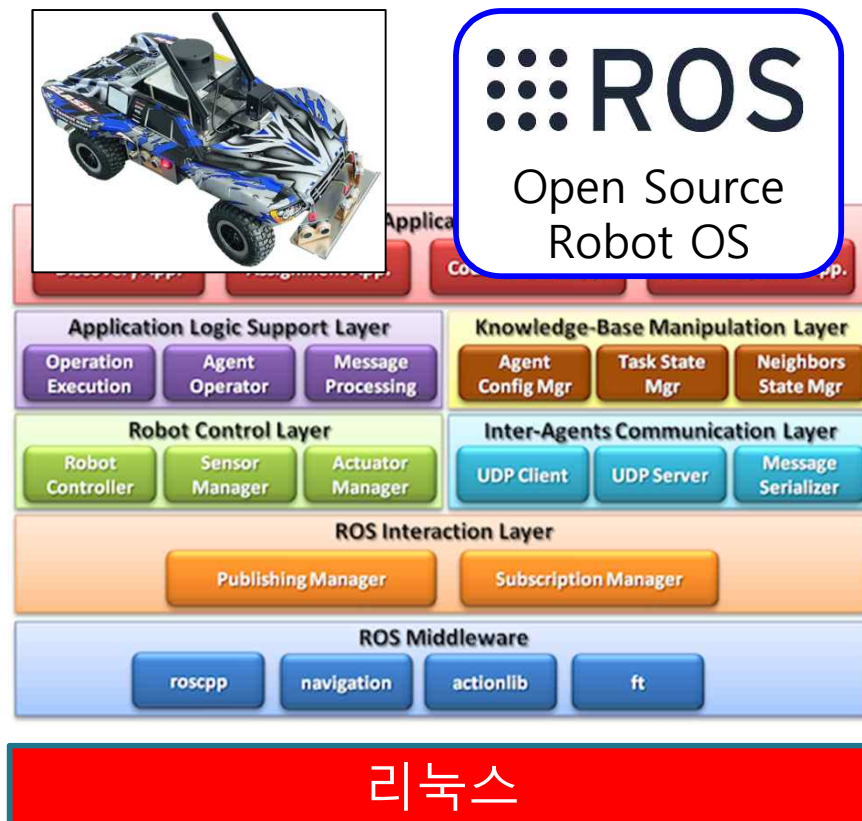
자이카 운영체제는 리눅스

- 운영체제로 리눅스 설치
 - 우분투 리눅스 (Ubutu Linux)
 - 22.04

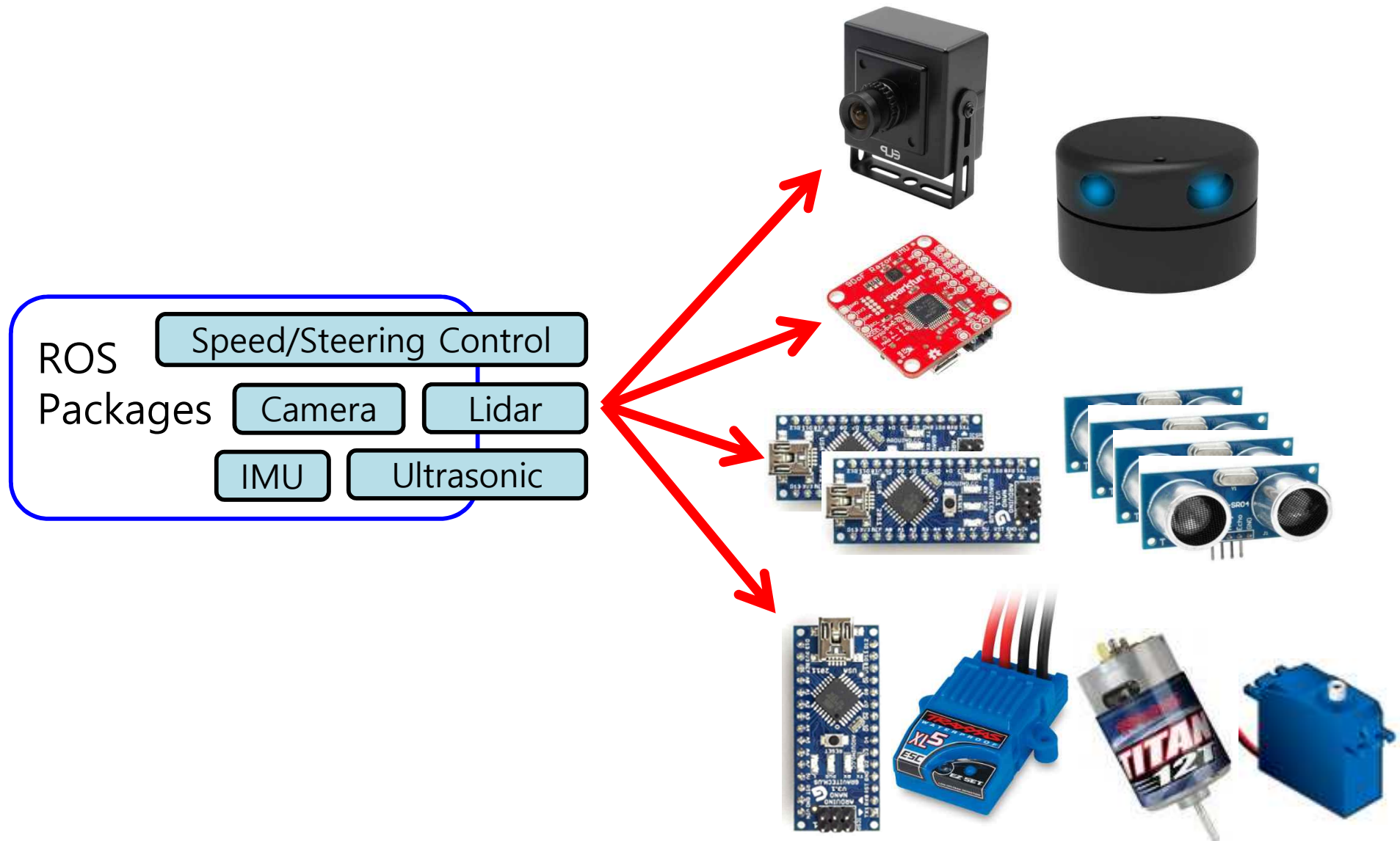


자이카 장치제어 미들웨어는 ROS

- 각종 센서와 모터제어를 위한 미들웨어
 - ROS (Robot OS) 설치



자이카 장치제어는 ROS Package



자율주행 SW 프로그래밍은 파이썬으로

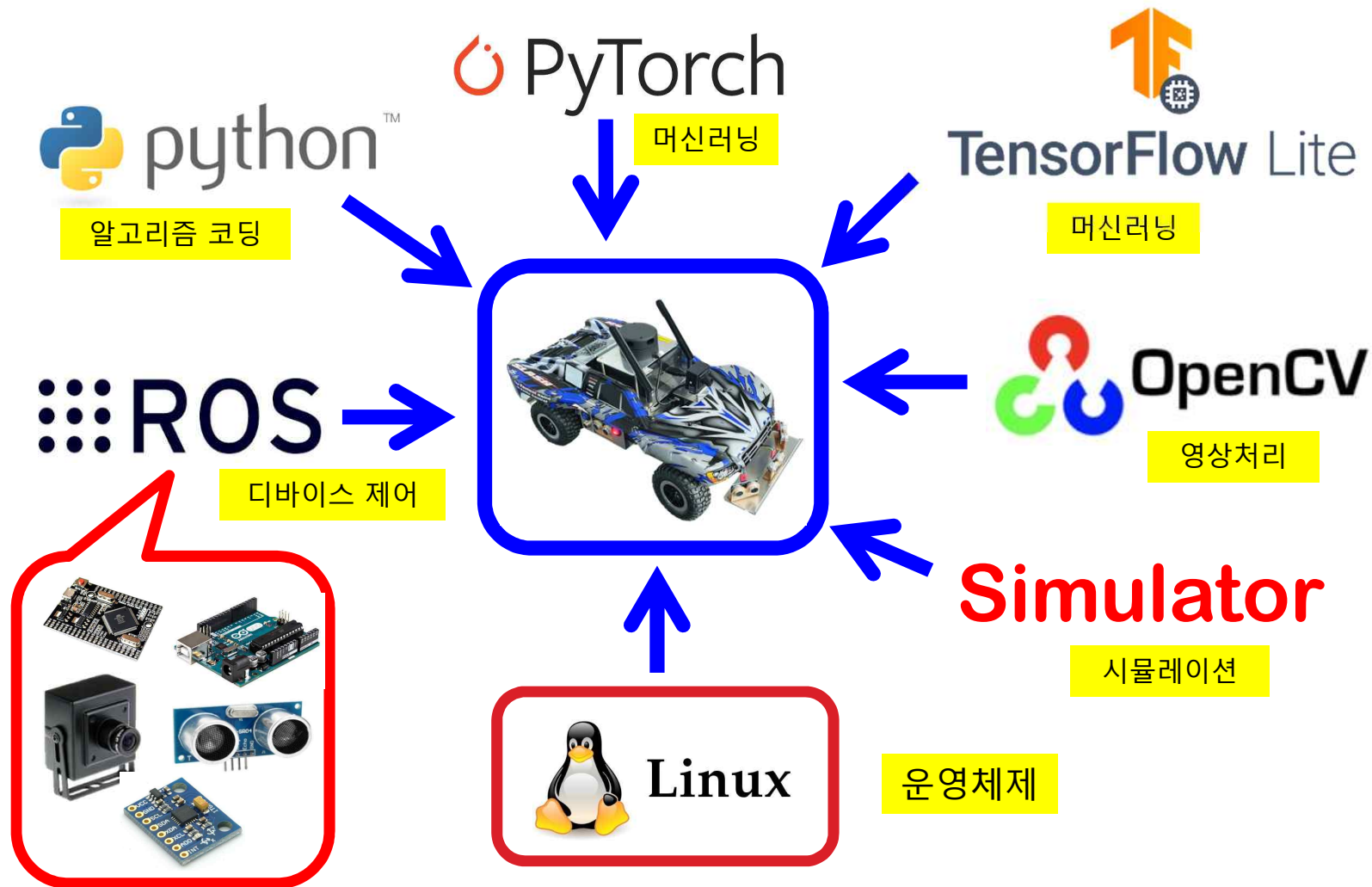
- 오픈소스 라이브러리 활용

자율주행SW



자이카 소프트웨어

- 자율주행 SW 플랫폼



실습장비 기동하기

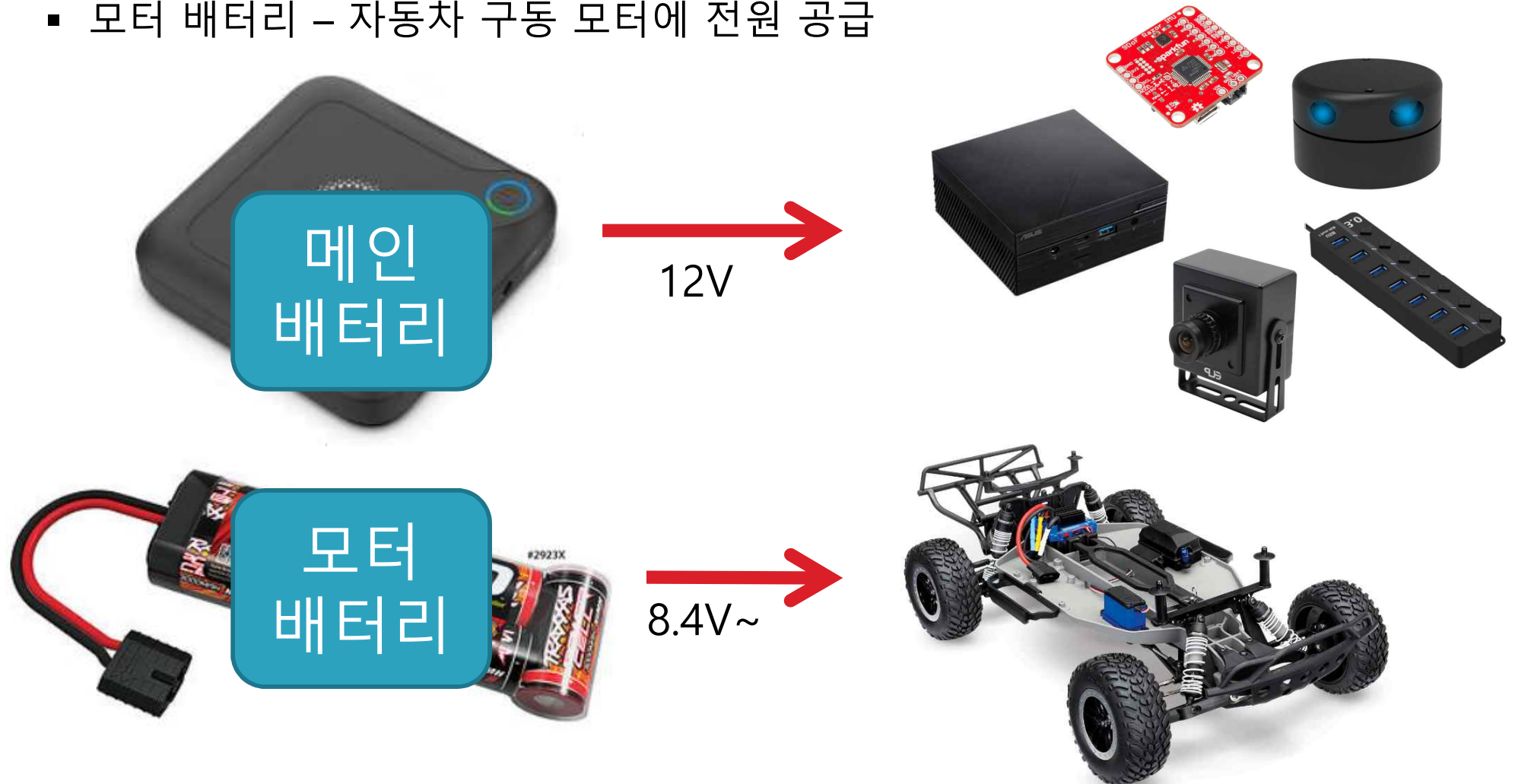
자이카 처음 전원 켜서 동작시키기

자이카 시동 걸기



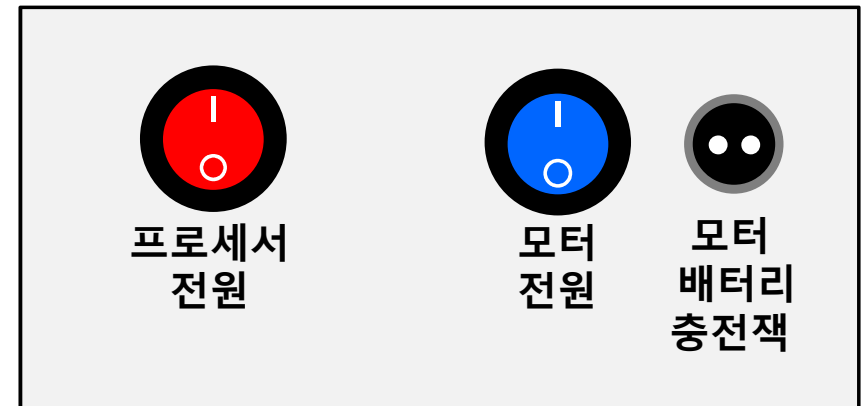
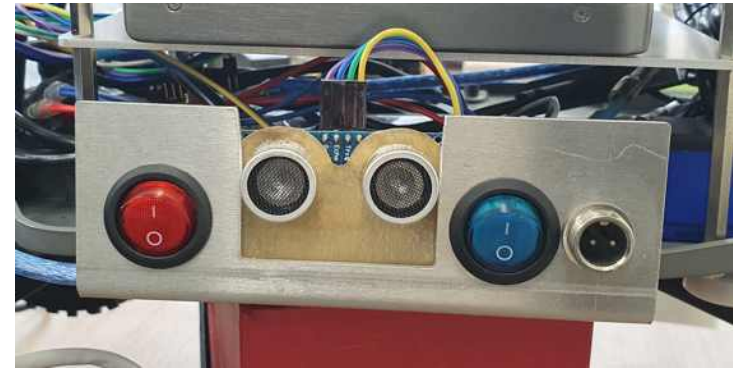
배터리

- 배터리 2개
 - 메인 배터리 – 프로세서 보드와 센서 장치에 전원 공급
 - 모터 배터리 – 자동차 구동 모터에 전원 공급



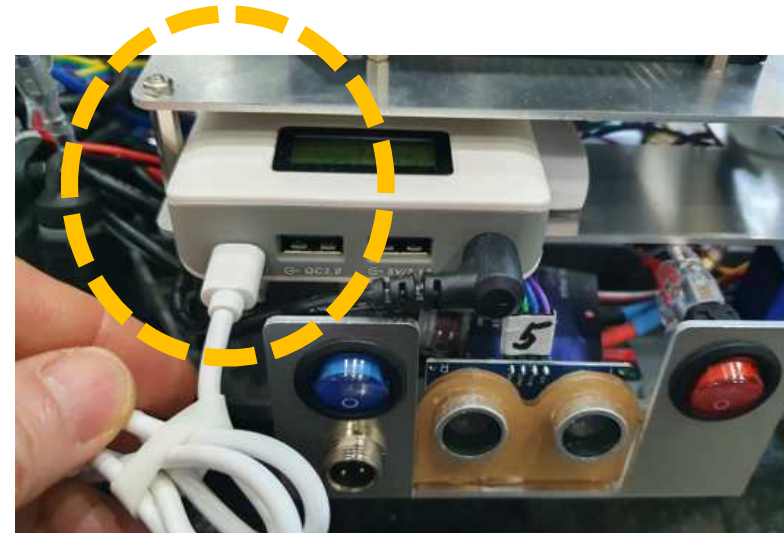
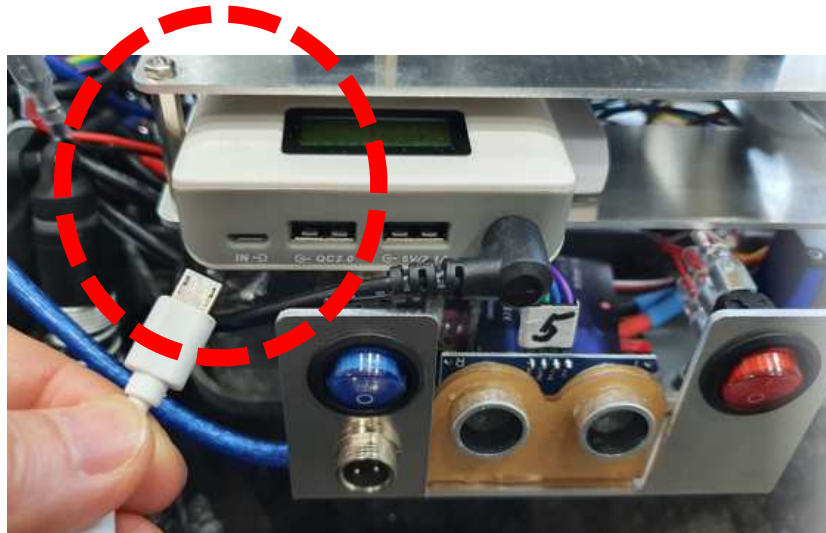
전원 켜기

- 자동차 측면 제어판넬에 있는 전원버튼을 누르면 된다
 - 빨간색 버튼 → 메인 배터리 전원스위치
 - 파란색 버튼 → 모터 배터리 전원스위치



메인 배터리 충전하기

- 메인 배터리
 - USB 충전기로 충전 가능
 - 배터리 용량은 20,000mAh
 - ▶ $20,000\text{mAh} = 20\text{Ah}$
 - ▶ $20\text{Ah} = 2\text{A} \times 10\text{h}$
 - ▶ 2A짜리 충전기로 충전하면 10시간 소요

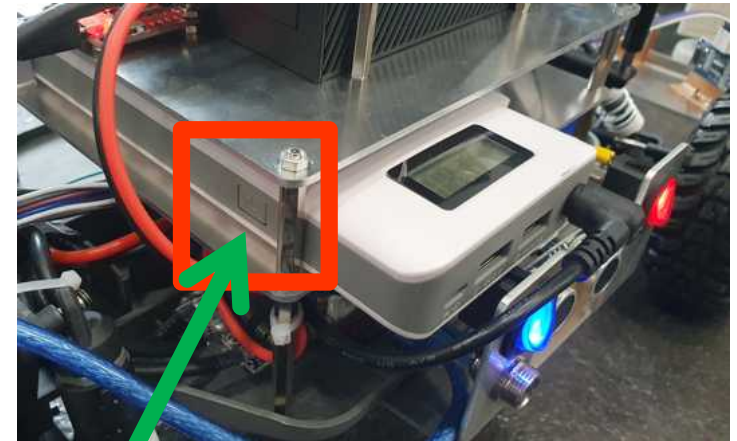


메인 배터리 켜기 & 잔량 확인

- 배터리에서 전기를 뽑아 쓰려면

배터리 위에 있는 시작버튼 눌러야 함

- ▶ 평상시는 꺼져 있음 (파워 세이프 모드)



시작 버튼

- 이걸 눌러야 전기가 나옴
- 배터리 잔량 확인도 가능함
- 전기를 뽑아 쓰지 않으면 자동으로 다시 꺼짐 (파워 세이프 모드)

모터 배터리 충전하기

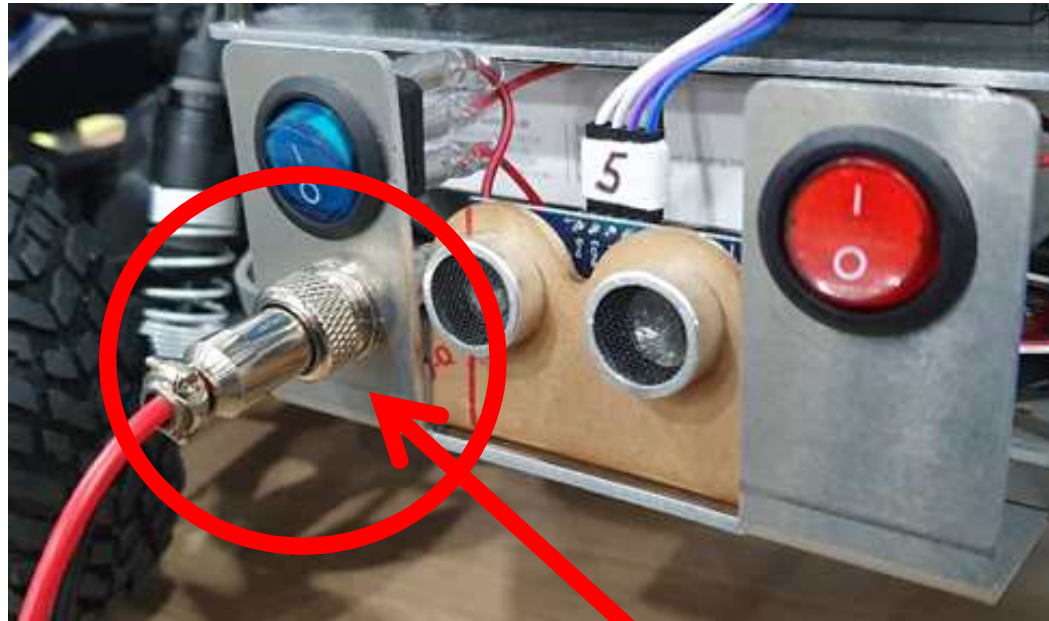
- 용량 3,000mAh
- 특별한 충전기를 사용하여 충전
 - 고속 충전이 가능 (1시간 이내)



충전기



모터 배터리 충전하기



8.4V 1.5A



모터 배터리 충전기 사용법

- 배터리 종류 선택



첫 번째 "Batt. Type" 버튼을
여러 번 눌러서 오른쪽과 같이
"NiMH BATT"을 선택한 후에
네 번째 "Start" 버튼을 누른다.

USER SET
PROGRAM->

PROGRAM SELECT
LiPo BATT

PROGRAM SELECT
NiMH BATT

PROGRAM SELECT
NiCd BATT

PROGRAM SELECT
Pb BATT

PROGRAM SELECT
SAVE DATA

PROGRAM SELECT
LOAD DATA

모터 배터리 충전기 사용법

- 충전 전압과 전류량 세팅



이 숫자가 깜빡이면 변경이 가능한 상태이다.

맨 오른쪽 "Start/Enter" 버튼을 한 번 누르면
 백 소리와 함께 숫자 부분이 깜빡인다.
 두번째와 세번째 버튼 "Dec.", "Inc." 버튼을 눌러서
 숫자(충전 전류값)가 1.5A가 되도록 조정한다.
 1.5A로 세팅이 완료되면 "Start/Enter" 버튼을 누른다.
 백 소리가 나면서 숫자 깜빡임이 없어진다. (세팅 완료)

모터 배터리 충전기 사용법

• 충전 시작하기



BATTERY CHECK
WAIT...

NiMH 1.5A 09.80V
CHG 005:12 00006

맨 오른쪽 "Start/Enter" 버튼을
2초 이상 누르고 있으면
뽁 소리와 함께 충전이 시작된다.

1.5A : 충전전류가 1.5A=1,500mA임을 표시
09.80V : 현재 배터리 전압이 9.8V임을 표시
005:12 : 충전시간이 5분12초 지났음을 표시

모터 배터리 충전기 사용법

- 충전 완료, 마무리



삐비빅!
삐비빅!
삐비빅!

충전이 완료되면 삐비빅~ 삐비빅~ 삐비빅~
알람소리와 함께 자동으로 충전이 종료된다.
자동차에 연결된 어댑터를 빼면 된다.



분리

메인 보드 전원 켜기

- 메인 배터리 On
 - 메인배터리에 있는 시작버튼을 먼저 한 번 누르고 (배터리 슬립모드 해제)
 - 빨간색 스위치를 켜다.



메인 보드 전원 켜기

- 빨간색 메인 버튼에 불이 켜지면
- 프로세서의 전원 버튼을 눌러 부팅시킨다.



길다란 버튼이
전원 버튼이다.
꾸욱~ 눌러준다.

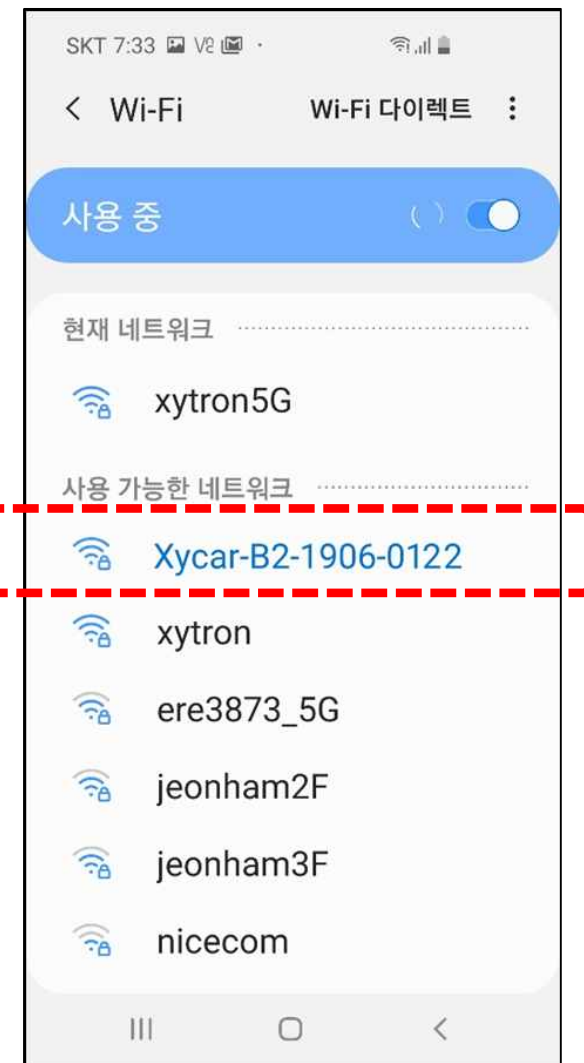


자이카 정상 기동 확인

- 스마트폰에서 와이파이 검색해 본다
 - 메인 보드에 노란색 라벨이 붙어있다.
같은 숫자의 와이파이가 잡히는지 확인한다.
 - 아래 예, Xycar-XXXXXXX 잡히면 OK



스마트폰 와이파이 검색



자이카 정상 기동 확인

- 자이카에 모니터를 붙인다.
 - 리눅스 부팅 화면이 표시되면 OK

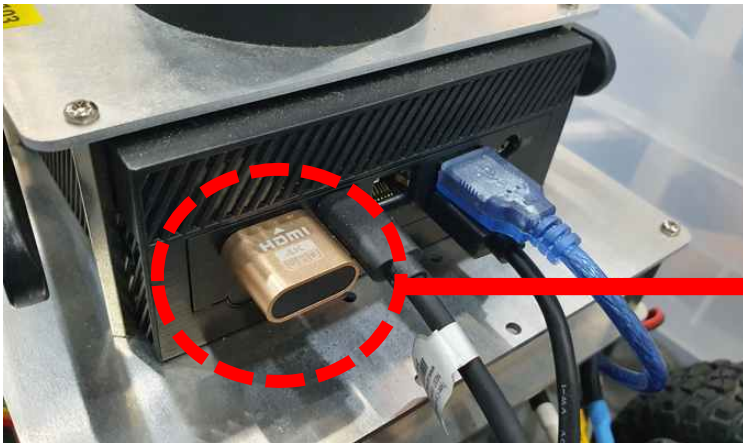


HDMI 모니터 케이블 연결

- 차량 프로세서 모듈에 HDMI 케이블 연결
 - (주의) 프로세서 부팅시 동글 또는 모니터 케이블이 꽂혀 있어야 함.

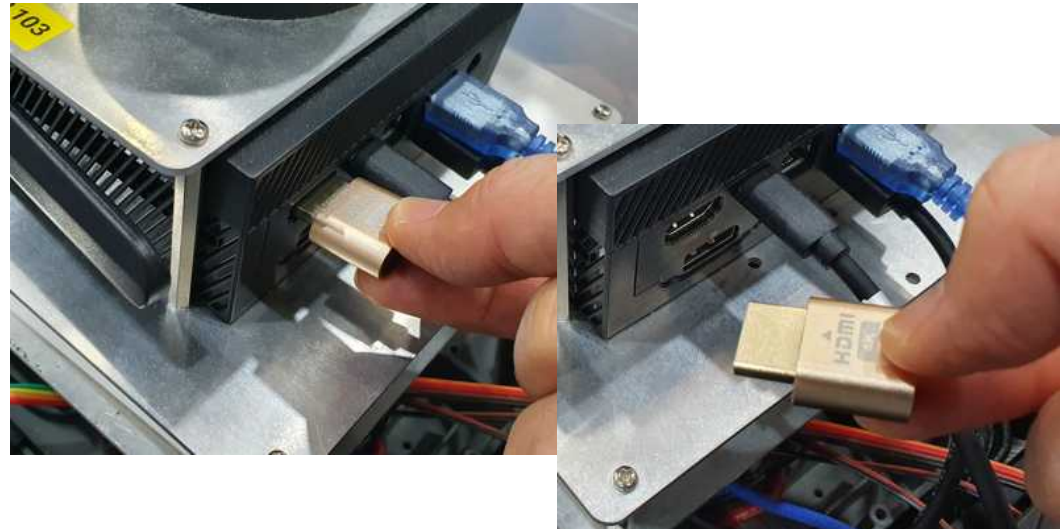
1

HDMI 케이블을
꽂을 수 있는
연결잭 제공



2

꽂혀 있는 동글을
빼서 잘 보관



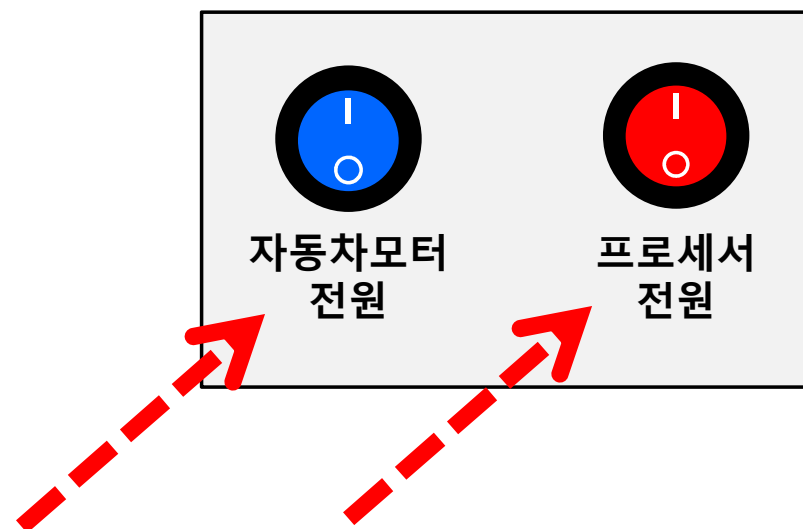
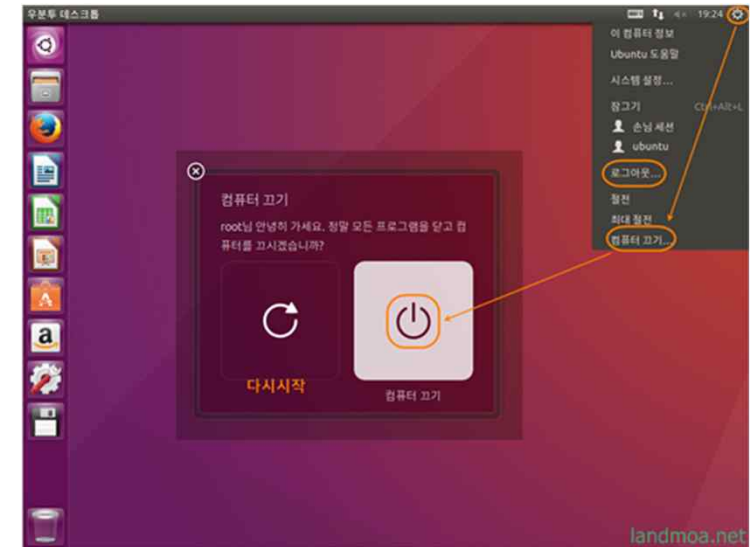
3

모니터 케이블 연결



사용 후 끝 때는...

- 가능하면 리눅스를 "전원끄기" 하고
 - 리눅스가 실행되는 프로세서 전원이 꺼진다
- 나중에 파란색 버튼을 끈다.
 - 모터 전원이 꺼진다



원격 접속 개발환경

자이카의 무선 와이파이 기능

자이카는 리눅스 컴퓨터

- 프로세서 + 저장공간 + 무선랜 + **키보드 + 마우스 + 모니터**



자이카에 모니터를 붙이면

- 자이카에 모니터를 연결한다.
 - 전원을 켜면 리눅스 부팅 화면이 나타남
 - 리눅스 PC로 사용 가능



자이카는 무선공유기 ?

- 자이카가 무선공유기, 즉 AP(Access Point)로 동작한다.
- 마치 무선공유기처럼,



자이카에 무선으로 접속

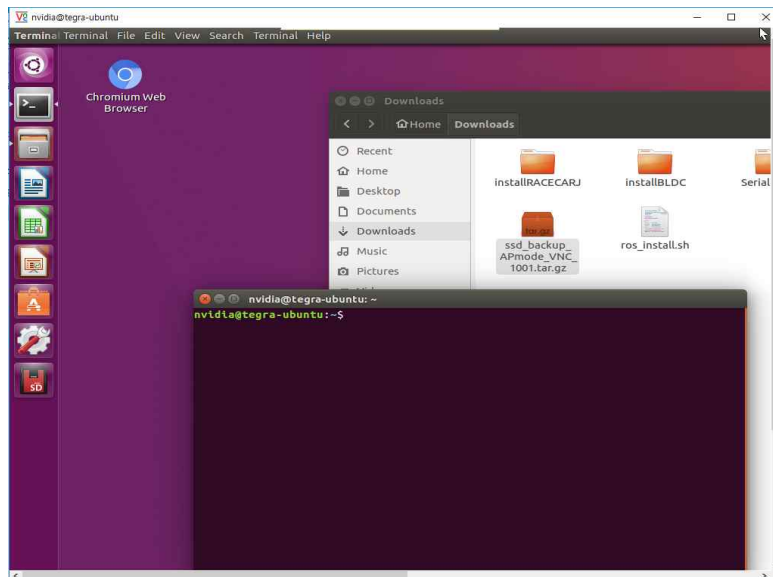
- 무선 와이파이로 자이카에 접속 가능
 - 차량과 무선으로 연결해서 SW 작업 가능
 - 차량에 장착된 카메라의 영상을 PC 또는 휴대폰으로 확인 가능



GUI 기반 원격접속



노트북/스마트폰에서
자이카로 원격접속



와이파이 접속

- 자이카 = 무선공유기



아이디

카메라 위에 적혀 있다

비밀번호



없음
(개방형 와이파이)



스마트폰에서 자이카 와이파이에 접속하기

- 와이파이 비번은 없다. (개방형 와이파이)



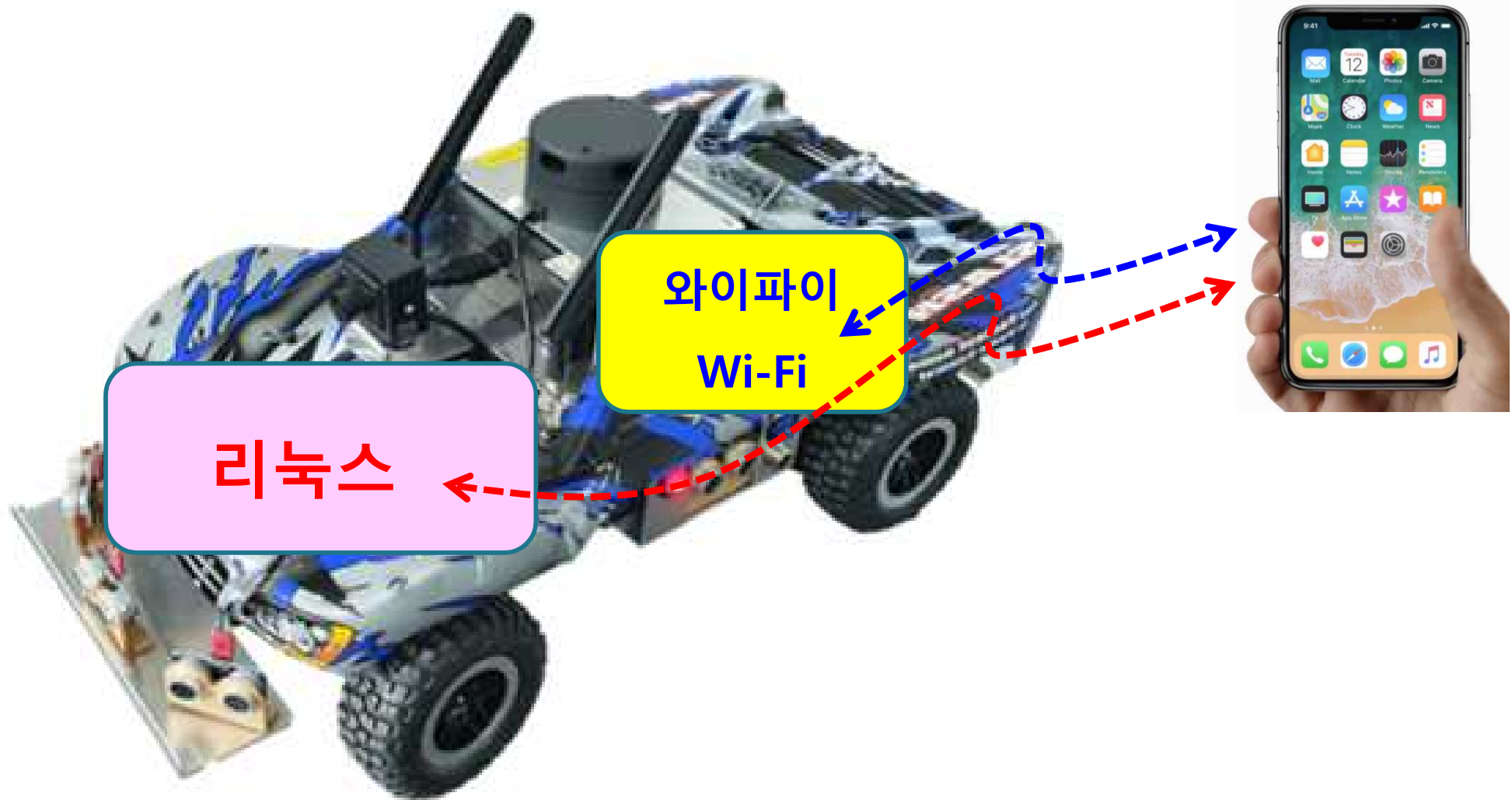
“인터넷 없음...” 에러? 에러가 아니다!



차량과 인터넷을 연결해 주는
장치가 바로 '모뎀'
→ 차량에 랜선이 연결되어 있지 않습니다.



와이파이 접속 후에 → 리눅스에 로그인 해야



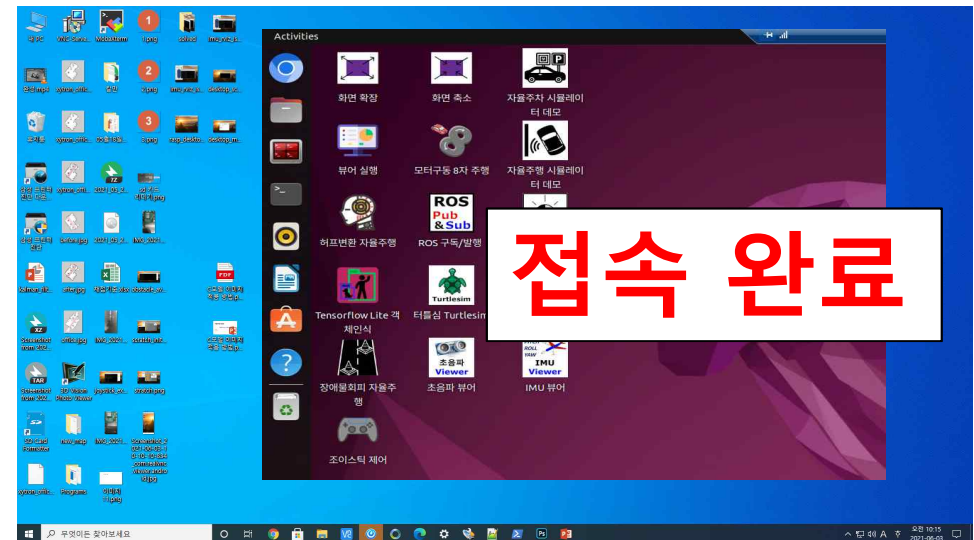
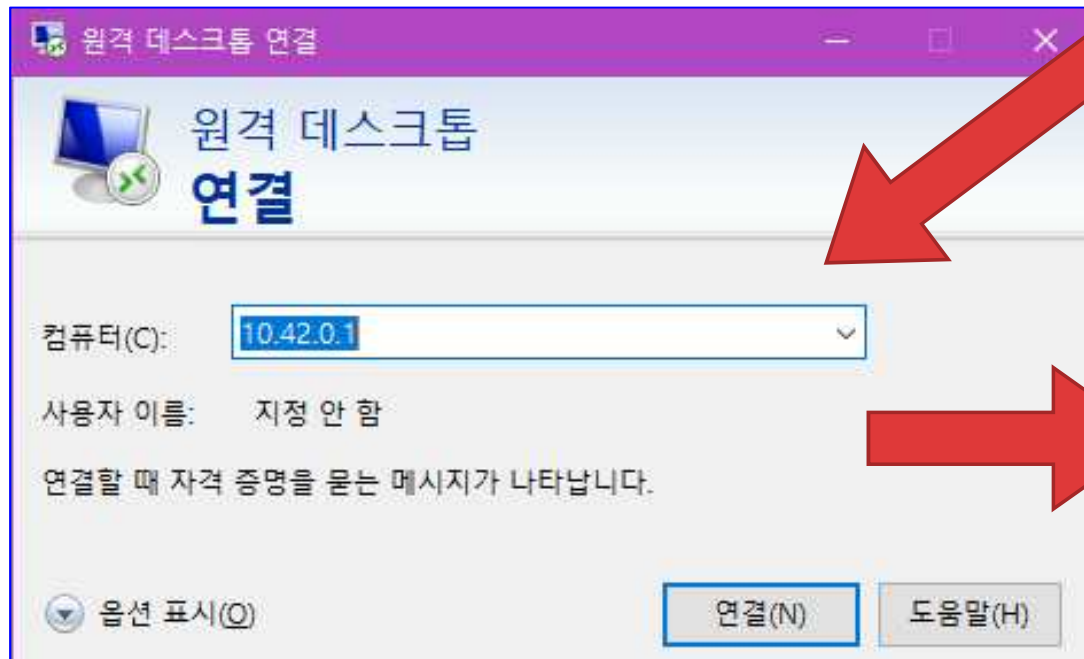
무선 원격접속 프로그램 - 원격 데스크톱

- Windows 원격 데스크톱 연결



- (1) PC에서 와이파이 검색해서 xy-car-*** 접속한 다음에,
- (2) “원격데스크톱” 실행해서 접속한다.

접속주소 : 10.42.0.1
Username : xytron
Password : xytron



무선 원격접속 프로그램 – Remote Desktop

- 스마트폰 앱



(1) 스마트폰에서 와이파이 검색해서

xycar-*** 접속한 다음에,

(2) 위 원격접속 앱을 실행하여 연결한다.

접속주소 : 10.42.0.1

Username : xytron

Password : xytron



접속 완료

원격 접속 완료



노트북과 스마트폰 여러 대 사용 가능

- 여러 장치들을 동시에 자이카에 접속시킬 수 있다

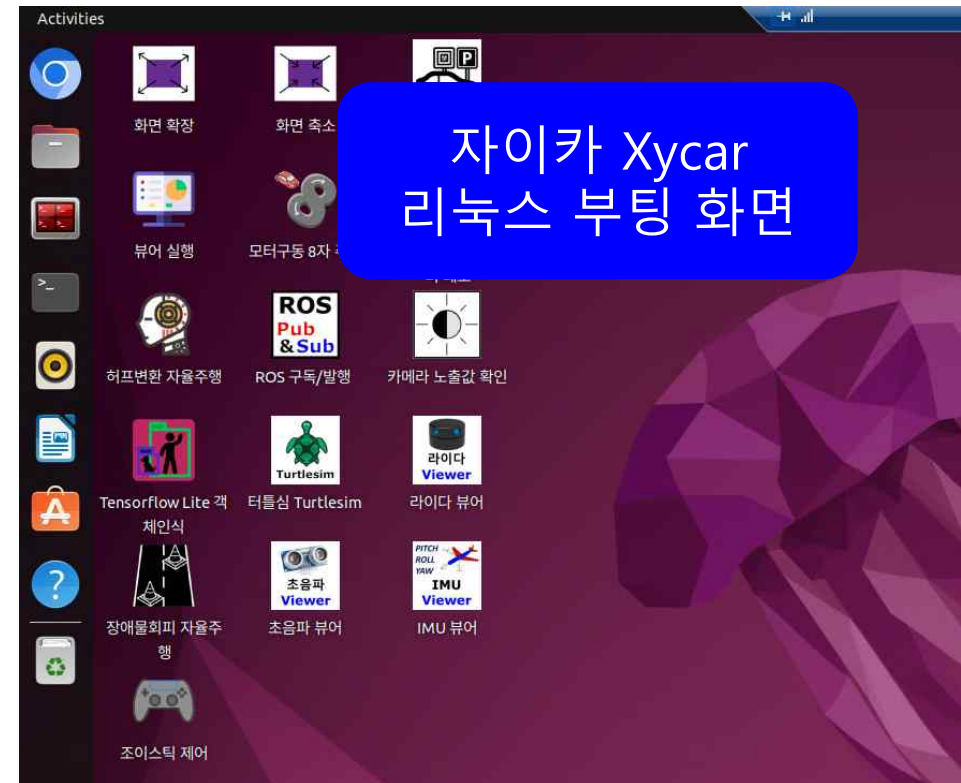


자이카 주행시킴

자이카 처음 움직여보기 &
여러가지 방법으로 주행시킴



전원 켜고 부팅시키면 바탕화면의 앱 실행 가능

- 리눅스 바탕화면
 - 다양한 SW 제공 → 아이콘 클릭해서 실행 가능 (노트북/스마트폰 이용)



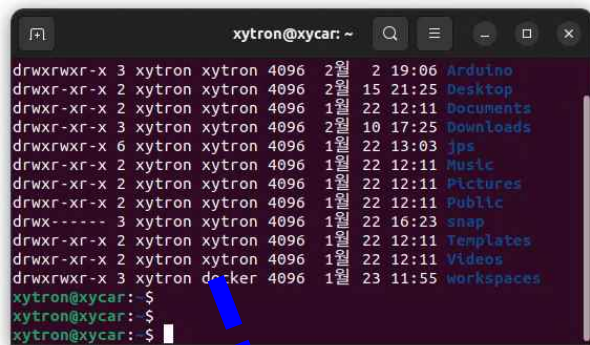
실행 아이콘

- 바탕화면 실행아이콘

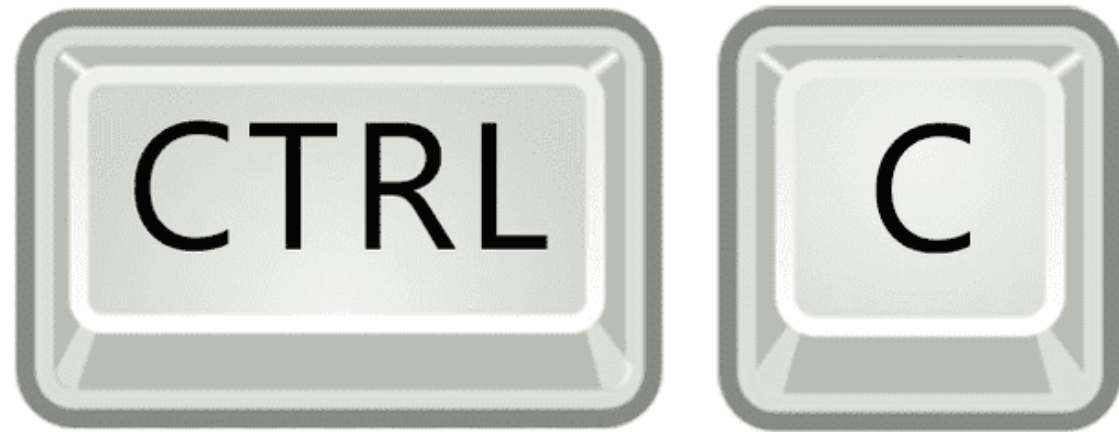
 <p>ROS Pub &Sub</p> <p>ROS 구독/발행</p>	 <p>Turtlesim</p> <p>터틀심 Turtlesim</p>	 <p>뷰어 실행</p>	 <p>IMU Viewer</p> <p>IMU 뷰어</p>	 <p>IMU Viewer</p> <p>IMU 뷰어</p>
 <p>장애물회피 자율주행</p>	 <p>카메라 노출값 확인</p>	 <p>모터구동 8자 주행</p>	 <p>초음파 Viewer</p> <p>초음파 뷰어</p>	 <p>라이다 Viewer</p> <p>라이다 뷰어</p>

실행한 프로그램을 끝내고 싶을 때

- 열린 터미널에서 Ctrl+C 누르면 된다. 또는 터미널을 닫는다.
 - 실행되던 프로그램들이 모두 종료된다.



```
xytron@xycar: ~  
drwxrwxr-x 3 xytron xytron 4096 2월 2 19:06 Arduino  
drwxr-xr-x 2 xytron xytron 4096 2월 15 21:25 Desktop  
drwxr-xr-x 2 xytron xytron 4096 1월 22 12:11 Documents  
drwxr-xr-x 3 xytron xytron 4096 2월 10 17:25 Downloads  
drwxrwxr-x 6 xytron xytron 4096 1월 22 13:03 jps  
drwxr-xr-x 2 xytron xytron 4096 1월 22 12:11 Music  
drwxr-xr-x 2 xytron xytron 4096 1월 22 12:11 Pictures  
drwxr-xr-x 2 xytron xytron 4096 1월 22 12:11 Public  
drwx----- 3 xytron xytron 4096 1월 22 16:23 snap  
drwxr-xr-x 2 xytron xytron 4096 1월 22 12:11 Templates  
drwxr-xr-x 2 xytron xytron 4096 1월 22 12:11 Videos  
drwxrwxr-x 3 xytron docker 4096 1월 23 11:55 workspaces  
xytron@xycar: $  
xytron@xycar: $  
xytron@xycar: $
```



Ctrl + C

실행아이콘 - 뷰어 실행

- 각종 센서의 동작 상태를 확인하고자 할 때 사용
- 차량에 장착된 센서들을 모두 동작시키고
 - 센싱된 데이터를 보기 좋게 그래픽 등으로 표시해 준다.
 - 라이다, 카메라, IMU, 초음파센서의 데이터를 살펴볼 수 있다.



뷰어 실행

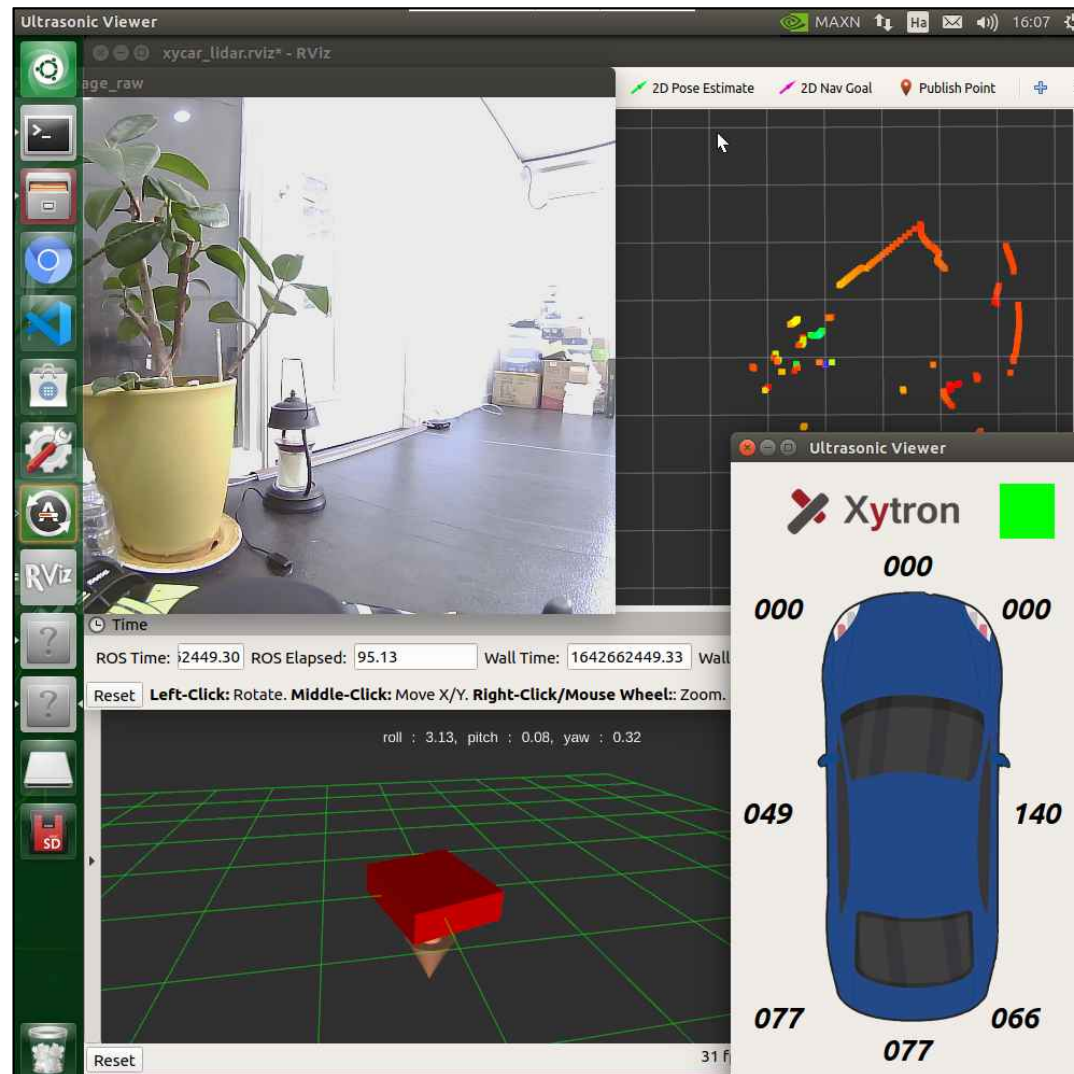


실행아이콘 – 뷰어 실행

- 여러 개의 센서데이터 뷰어가 동시에 실행된다.



뷰어 실행

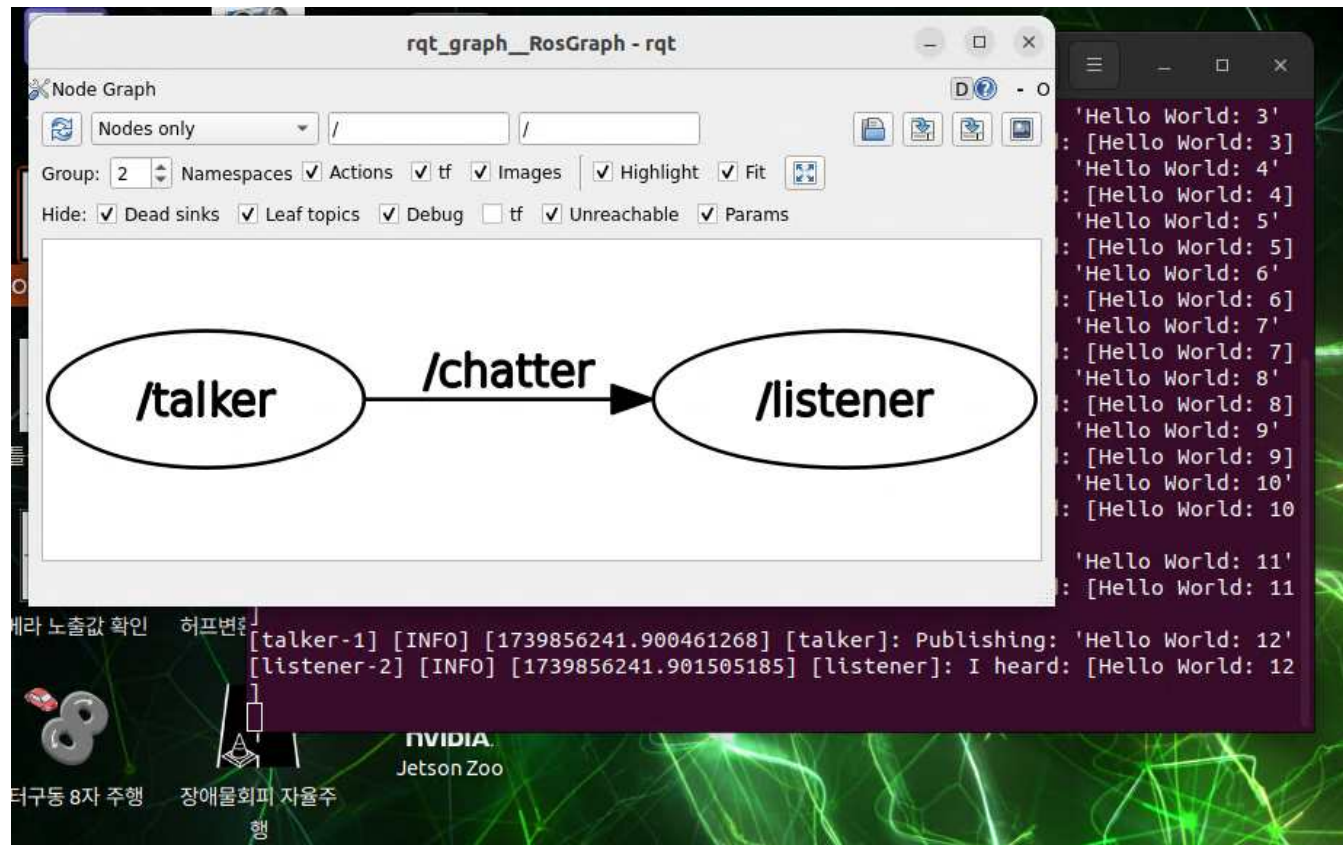


실행아이콘 – ROS2 발행자/구독자 예제

- 스트링 문자열을 담은 토픽을 발행하는 노드와 그걸 구독하는 노드를 구동한다.
- 함께 Rqt_Graph 프로그램도 실행시킨다.

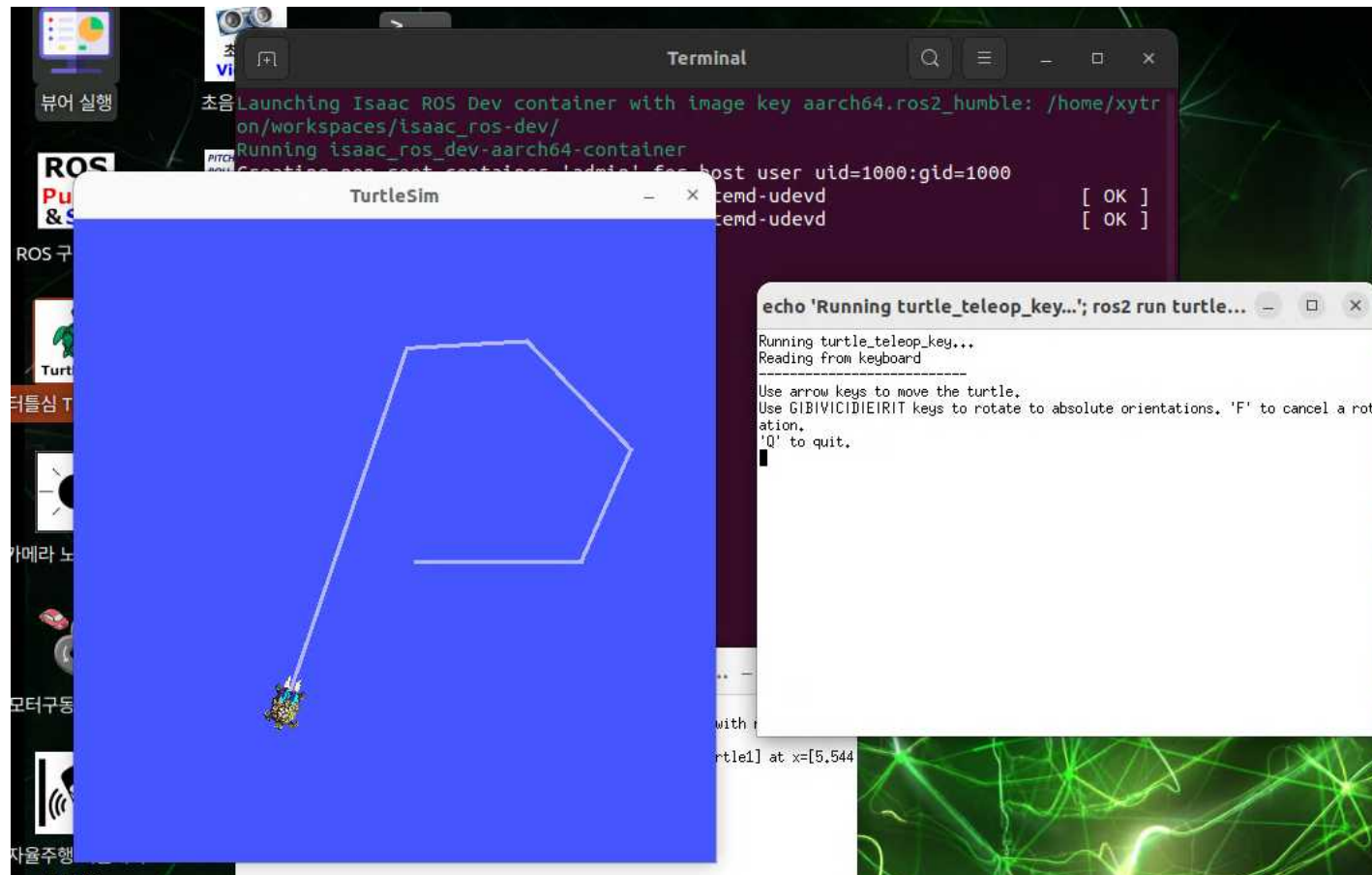
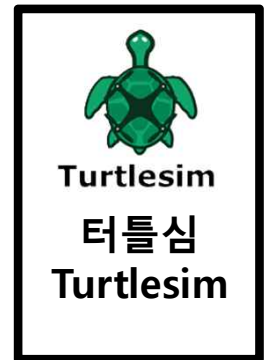
ROS
Pub
&
Sub

ROS
구독/발행



실행아이콘 – 모터구동 8자 주행

- ROS2 예제 코드인 터틀심 프로그램을 구동한다.
 - 2개의 노드가 실행된다. turtlesim_node + turtle_teleop_key
 - 키보드(방향키)를 이용해서 거북이를 움직일 수 있다.



실행아이콘 - 노출값 확인

- 카메라의 노출값 (Exposure 값)의 설정에 사용한다
 - 노출을 조정하여 화면을 환하게 또는 어둡게 만들 수 있다.
 - 차선과 물체 인식에 필요하다

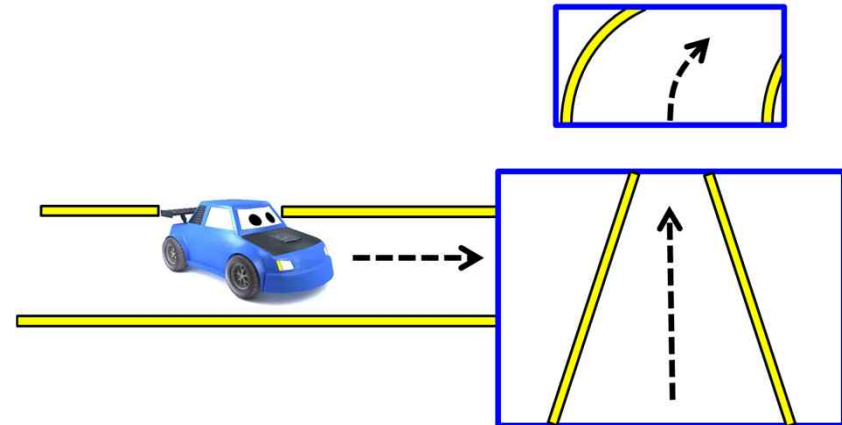


위아래 클릭하면
노출값을 바꿀 수 있다

찾은 노출값을 나중에
카메라 launch 파일에 반영하면
카메라로 촬영되는
영상의 밝기를 조정할 수 있다.

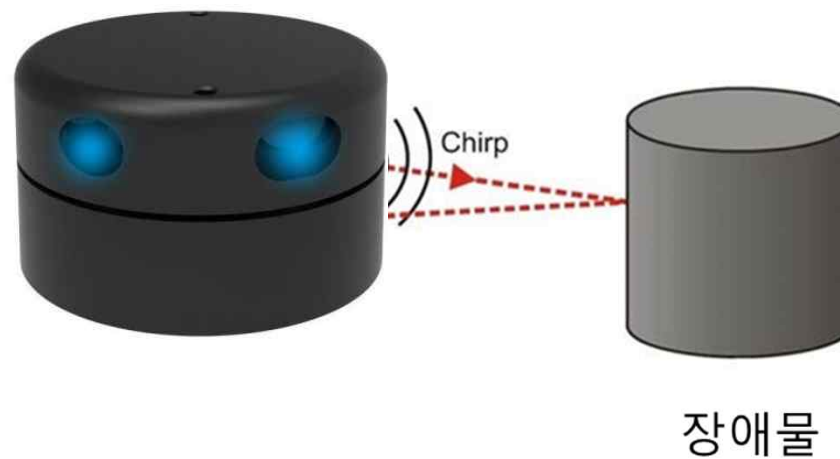
실행아이콘 – 자율주행

- 카메라 영상에서 차선을 찾아 차선을 따라 주행시킬 때 사용
- 카메라를 이용해 차선을 찾은 후 주행한다.
 - 허프 변환 알고리즘을 이용하여 차선을 찾는다.
 - 검은색 바닥에 흰색 차선이 있어야 한다.



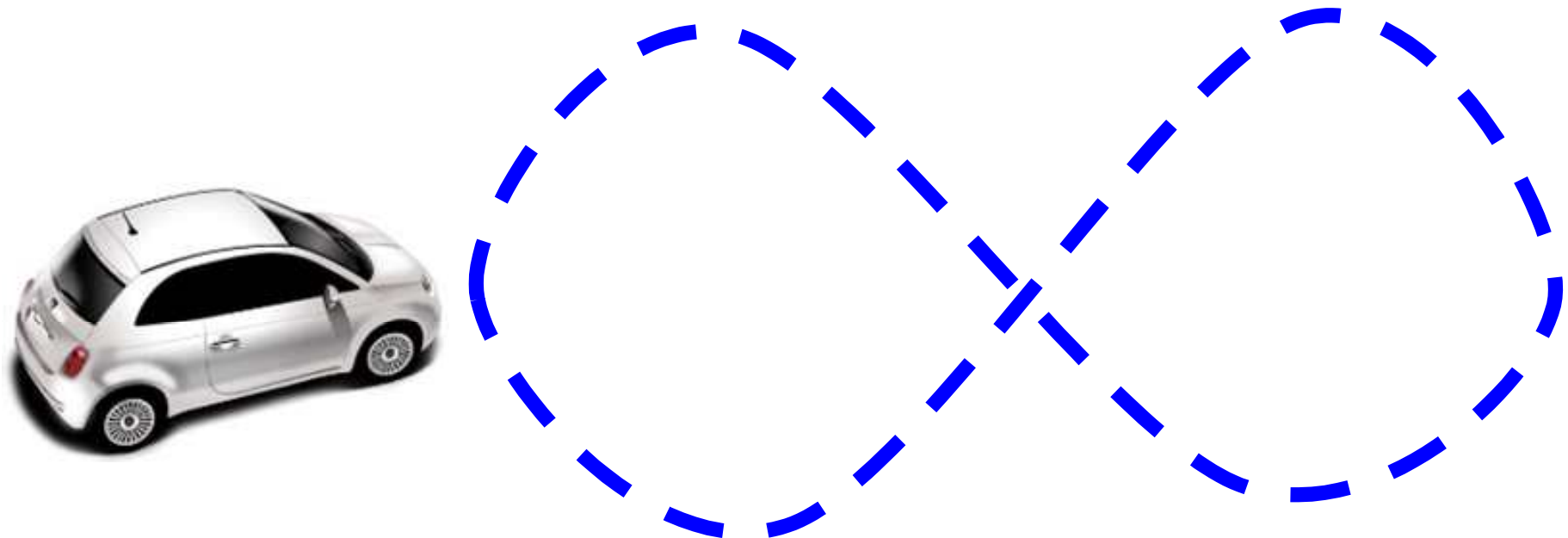
실행아이콘 – 장애물 회피 주행

- 카메라 대신 거리센서를 이용해서 주행시킬 때 사용
- 일단 차량이 직진하다가
 - 거리센서에 장애물이 감지되면
 - 방향을 바꾸어 장애물을 비켜 간다.



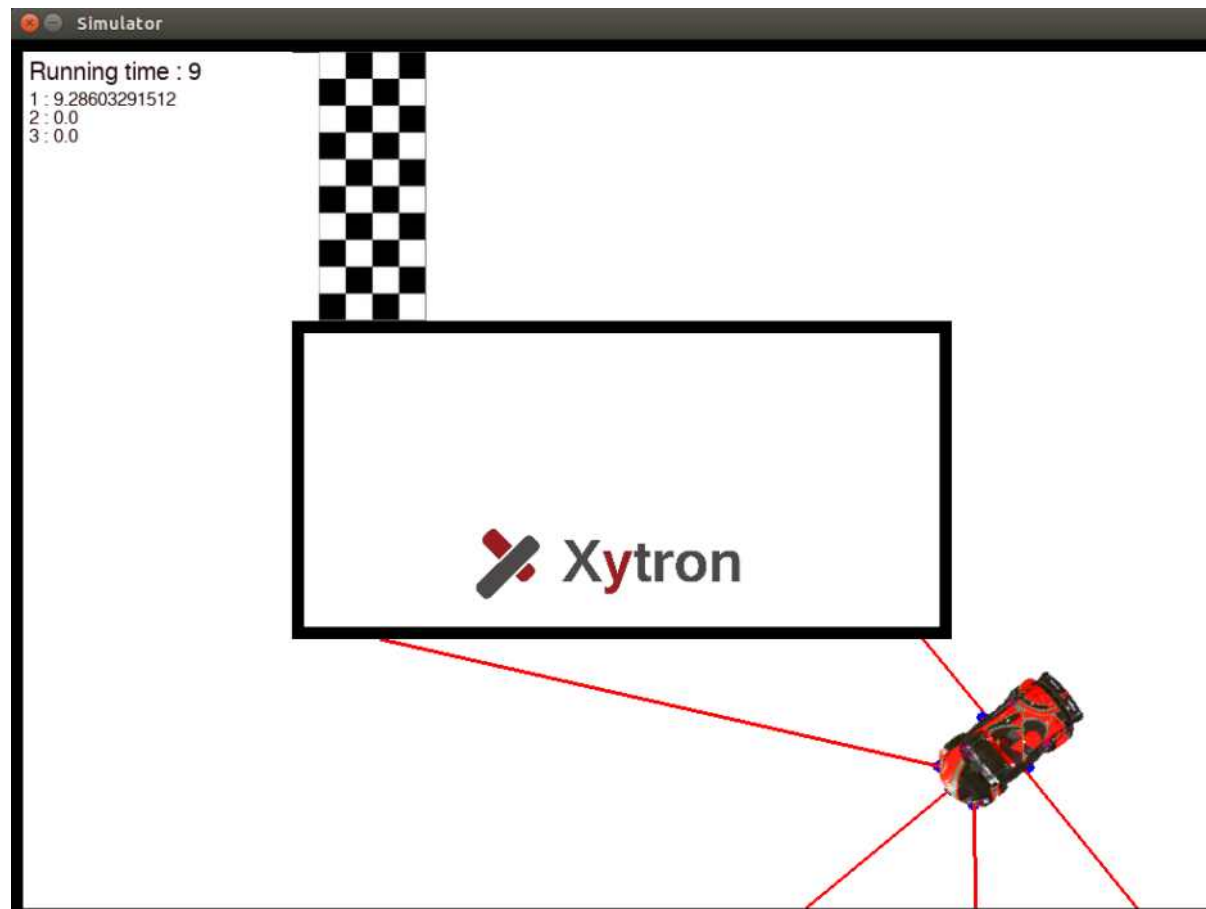
실행아이콘 – 모터구동 8자 주행

- 차량이 무조건 8자를 그리면 주행하게 함
 - 모터가 정상인지 확인하는데 적합
 - 센서를 전혀 사용하지 않으므로 장애물과 충돌 위험 있음.



실행아이콘 – 거리센서 시뮬레이터 주행 데모

- 2D 시뮬레이터에서 차량의 자율주행 모습을 시연
 - 5개의 거리센서 정보를 이용하여 벽과 충돌하지 않고 주행
 - 알고리즘을 이용한 충돌회피 자율주행 예제코드 실행



실행아이콘 – 자율주차 시뮬레이터 주행 데모

- 2D 시뮬레이터에서 자율주차(Auto parking) 모습을 시연
 - 차량의 위치와 주차영역의 거리, 방향, 각도를 이용한 자율주차
 - 경로생성과 경로추종 알고리즘을 이용한 자율주차 예제코드 실행



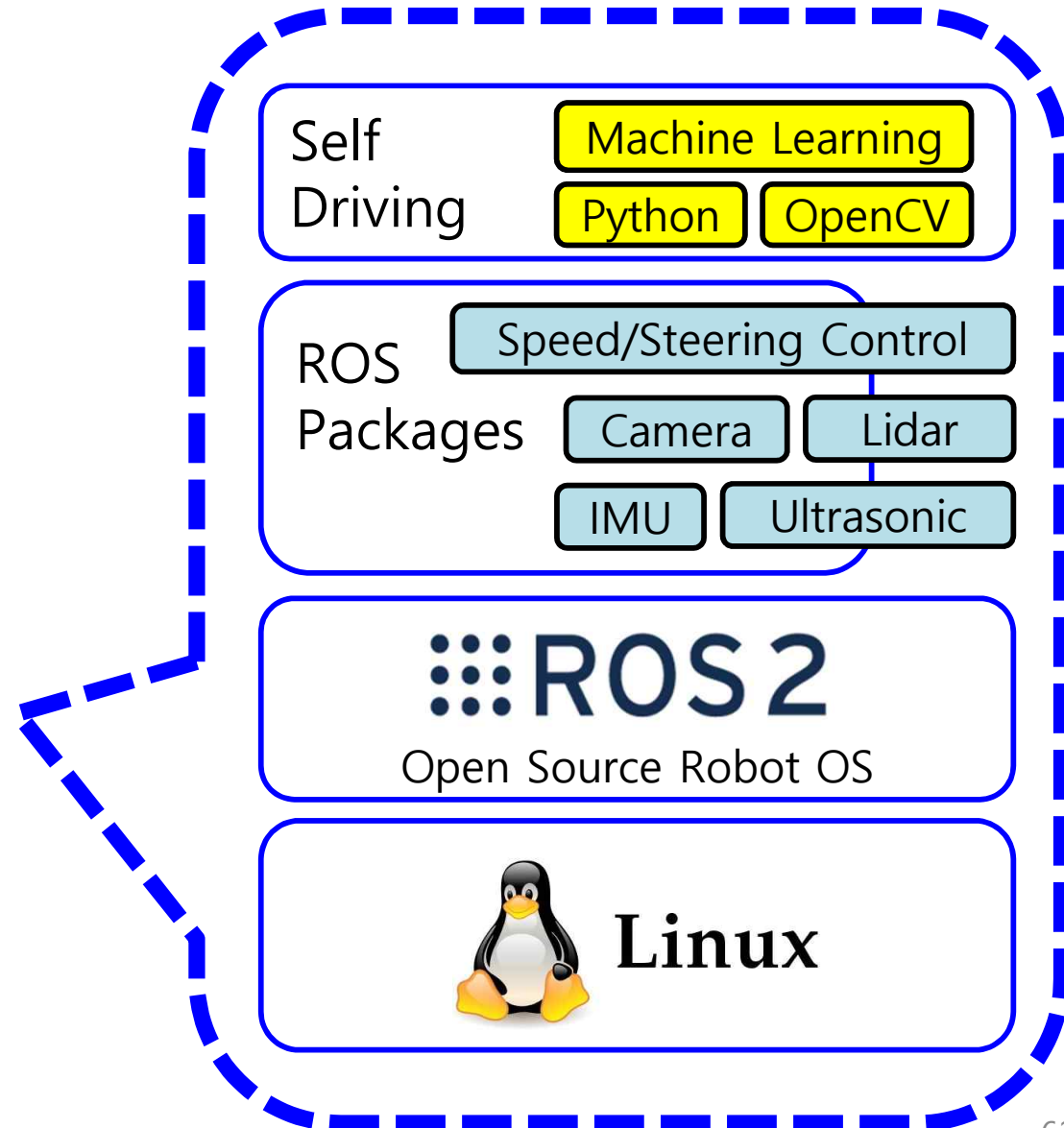
자이카 SW 구조

자이카 위에 탑재된 SW 플랫폼

자이카에 탑재되는 SW 구성도

• 시스템 SW

- ① 운영체제 – 리눅스
- ② 미들웨어 – ROS2
- ③ 라이브러리 – 영상처리



시스템SW - ① 리눅스

자이카 시스템SW – ① 운영체제 : 리눅스

- 운영체제 – 리눅스



Self
Driving

Machine Learning

Python

OpenCV

ROS
Packages

Speed/Steering Control

Camera

Lidar

IMU

Ultrasonic

ROS 2

Open Source Robot OS



Linux

자이카 = 리눅스 PC

- 자이카는 바퀴가 달려 움직일 수 있는 리눅스 PC



자이카 부팅

- 자이카에 모니터를 붙인다.
 - 전원을 켜면
 - 리눅스 부팅 화면이 나타남



DP cable



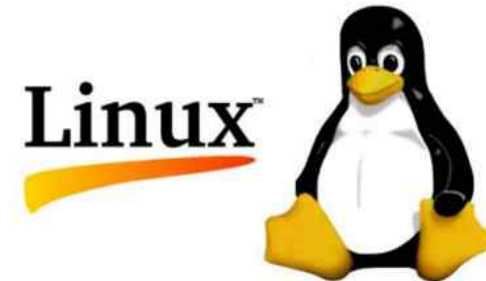
프로세서 보드와 모니터를
DP 케이블로 연결



프로세서 보드 위에 리눅스 설치

- 리눅스 관리자(root) 패스워드
 - 엔비디아 보드

아이디 : xytron
패스워드: xytron



시스템SW - ② ROS2

자이카 시스템 소프트웨어 – ② 미들웨어 : ROS

- 미들웨어 – ROS2 (Robot OS version 2)



Self
Driving

Machine Learning

Python

OpenCV

ROS

Packages

Speed/Steering Control

Camera

Lidar

IMU

Ultrasonic

ROS2

Open Source Robot OS



Linux

ROS (Robot Operating System)

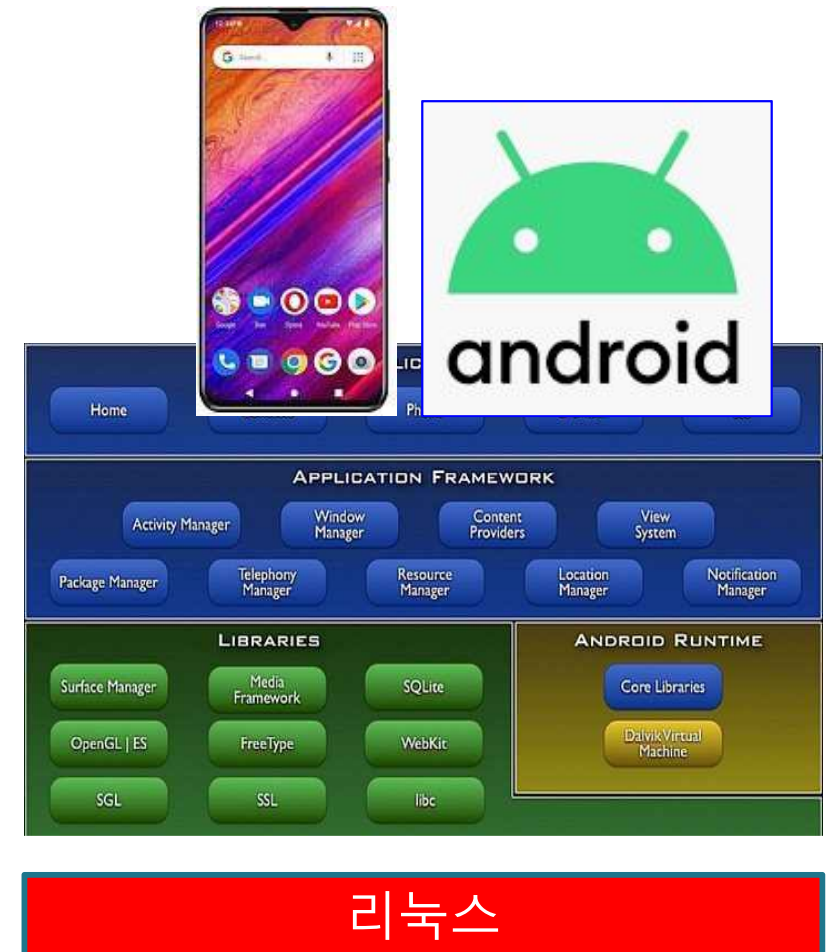
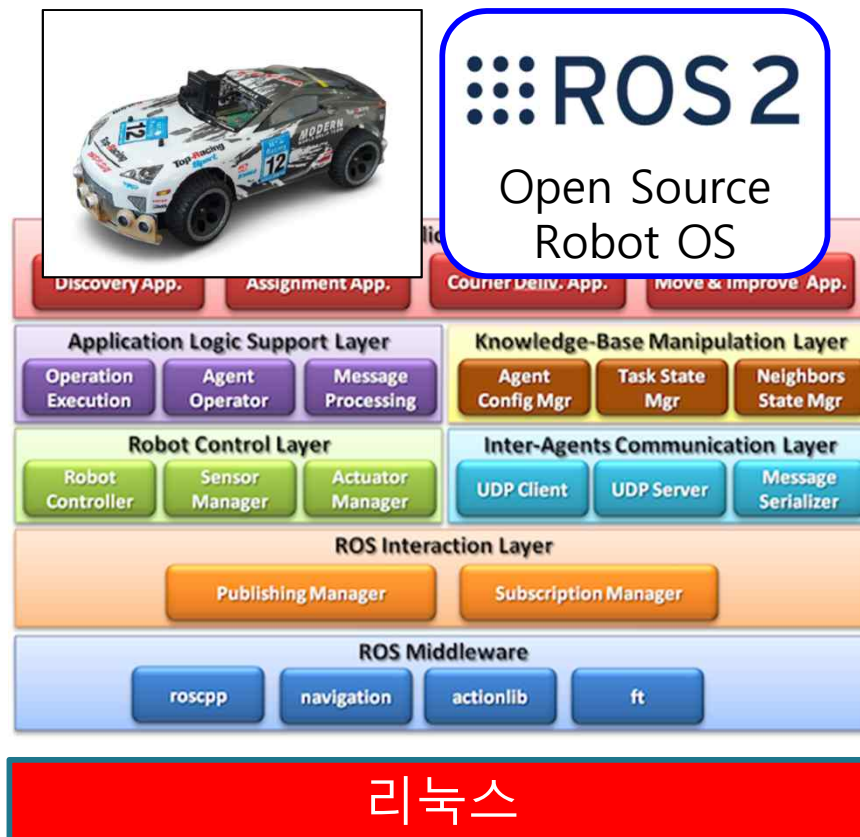
- 오픈소스 로봇 운영체제
 - 소스 무료 공개
 - 개방형 구조
 - 활발한 커뮤니티 활동



- 로봇 소프트웨어를 개발하는 데 필요한 소프트웨어의 집합체
 - 소프트웨어 프레임워크 (Software Framework)
- 메타 운영체제 (Meta OS), 미들웨어 (Middleware)
 - 소프트웨어 모듈 + 라이브러리 집합 + 도구 집합

ROS는 자동차(로봇의 일종) 제어를 위한 미들웨어

- 각종 센서와 모터를 프로그래머가 편하게 사용할 수 있도록 지원
 - 스마트폰에는 Android가... 자율주행차에는 ROS가...



ROS2 버전

ROS 2

Documentation for Active ROS Distributions



Previous Long Term Support ROS 2 Distro

[platform support]



Supported ROS 2 Distro

Latest Stable
[platform support]



Latest Long Term Support ROS 2 Distro

Latest LTS, Recommended for Robot Development
[platform support]



Development ROS 2 Distro

Recommended for Package Development
[platform support]

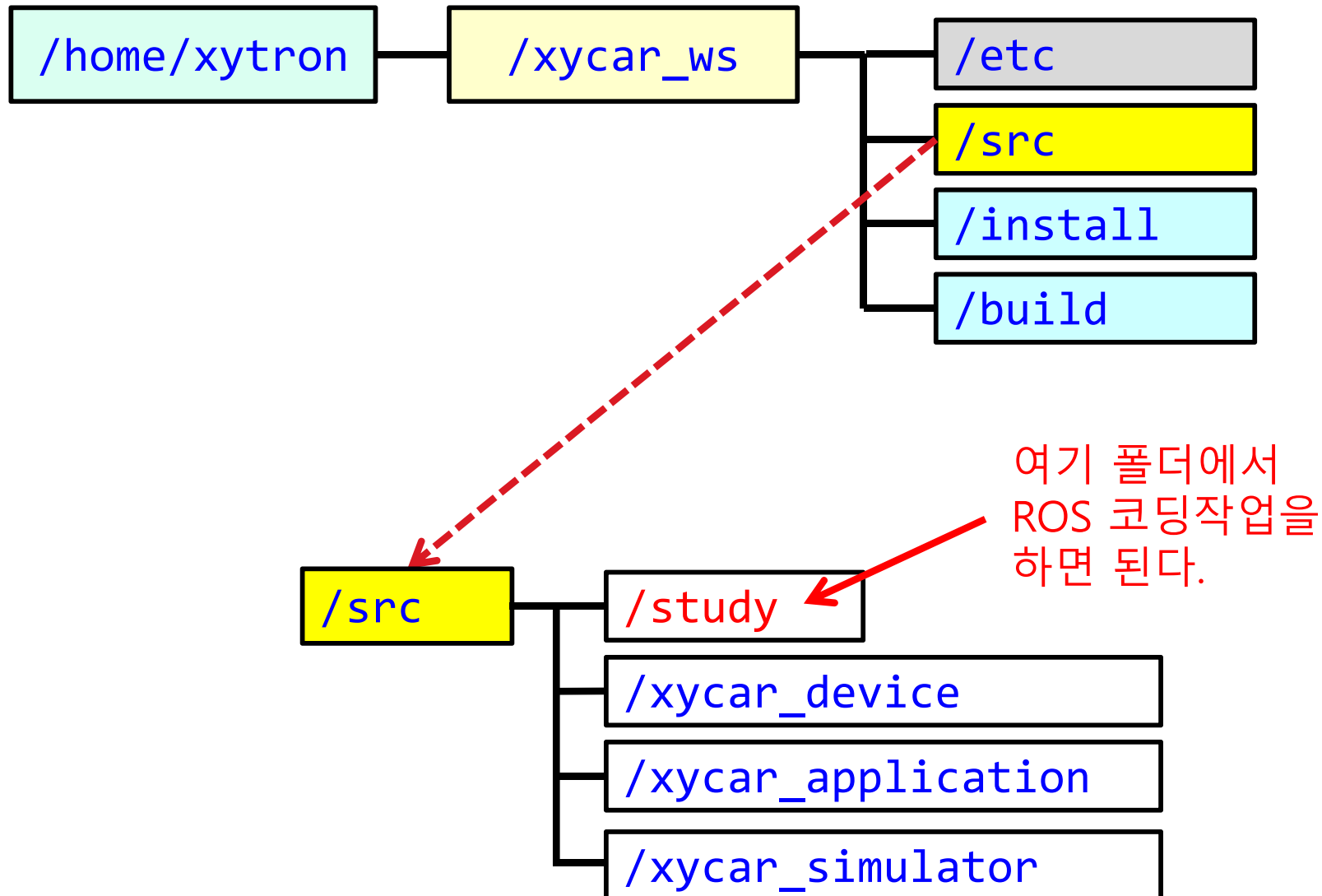


Long Term Support ROS 1 Distro

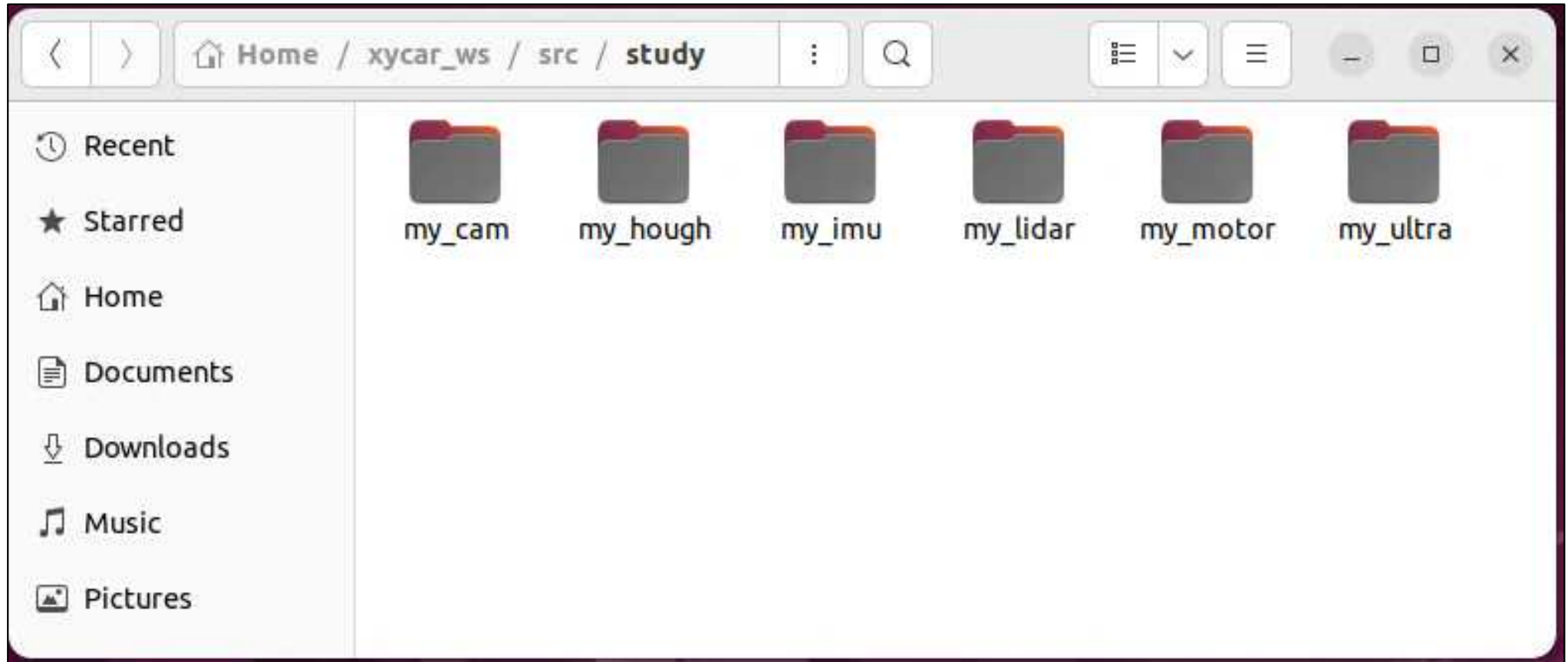
Recommended for ROS 1
[platform support]

자이카에 설치된 ROS 폴더 구조

- ROS 작업폴더



예제 코드 – study 폴더



예제 코드 – study 폴더

- 자이카에 장착되어 있는 각종 센서와 모터를 제어하기 위한 예제 제공
 - my_cam : 카메라 노드로부터 사진 토픽을 받아 화면에 표시하는 예제
 - my_hough : 카메라 사진에서 차선을 찾아 자율주행 하는 예제
 - my_imu : IMU 노드로부터 토픽을 받아 기울임정보를 표시하는 예제
 - my_lidar : 라이다 노드로부터 토픽을 받아 거리정보를 표시하는 예제
 - my_motor : 모터로 토픽을 보내 차량을 전진시키는 예제
 - my_ultra : 초음파 노드로부터 토픽을 받아 거리정보를 표시하는 예제

코딩 후 빌드 작업

- 2개의 Alias 명령을 사용할 수 있습니다.

```
$ cbs
```

이 명령어는 ~/xycar_ws/xycar_ws/src 아래에 있는 모든 ROS 패키지를 colcon 빌드하는 명령어입니다. 이 명령을 실행하기 전에 ~/xycar_ws/build/ 폴더와 ~/xycar_ws/install/ 폴더 아래에 있는 내용을 모두 삭제하는 것이 좋습니다.

```
$ cbp AAAA
```

이 명령어는 AAAA라는 특정한 ROS 패키지 하나만을 별도로 colcon 빌드하는 명령어입니다. AAAA 패키지 안에 있는 파이썬 파일이나 런치 파일 등을 수정했을 때 잊지말고 한번씩 실행하면 됩니다.



Q&A
