ROS Programming

(file #1 / ?) ver1.0 Kinetic / NOETIC

Yongseok Chi

Schedule

1. 준비사항 – C language, Linux

2. Schedule

<table>
<tr><th>교과목명</th><th>시간</th><th>프로젝트 교과목</th><th>시간</th></tr>
<tr><td rowspan="2">절차지향 프로그래밍
<table><tr><th>세부내용</th></tr><tr><td>- C 프로그래밍</td></tr><tr><td>- 메모리 구조 이해</td></tr><tr><td>- 자료구조 및 알고리즘 구현</td></tr><tr><td>- 시스템 프로그래밍</td></tr><tr><td>- Image Processing(Basic)-MFC</td></tr><tr><td>- Image Processing(Basic)-Python</td></tr></table></td><td>70Hr</td><td rowspan="2">ADAS 자율주행차 시스템 SW 개발(리눅스서버)
<table><tr><th>세부내용</th></tr><tr><td>- ROS 이해 및 설치환경</td></tr><tr><td>- Linux 시스템</td></tr><tr><td>- ROS 명령어</td></tr><tr><td>- SLAM(라이다 기반 네비게이션)</td></tr><tr><td>- Navigation</td></tr><tr><td>- 카메라 기반 자율주행</td></tr></table></td><td rowspan="2">50Hr</td></tr>
<tr><td>펌웨어 프로그래밍
<table><tr><th>세부내용</th></tr><tr><td>- 8bit AVR CPU이해 및Peripheral 실습</td></tr><tr><td>- Sub Project 1~4 (프로그래밍)</td></tr><tr><td>- 32bit Cortex-M7 CPU 이해 및 Peripheral 실습</td></tr><tr><td>- Sub Project 5 (프로그래밍)</td></tr></table></td><td>75Hr</td></tr>
</table>

- 수업시작 30분 : Quiz

- 수업 종료 30분전 : Q&A

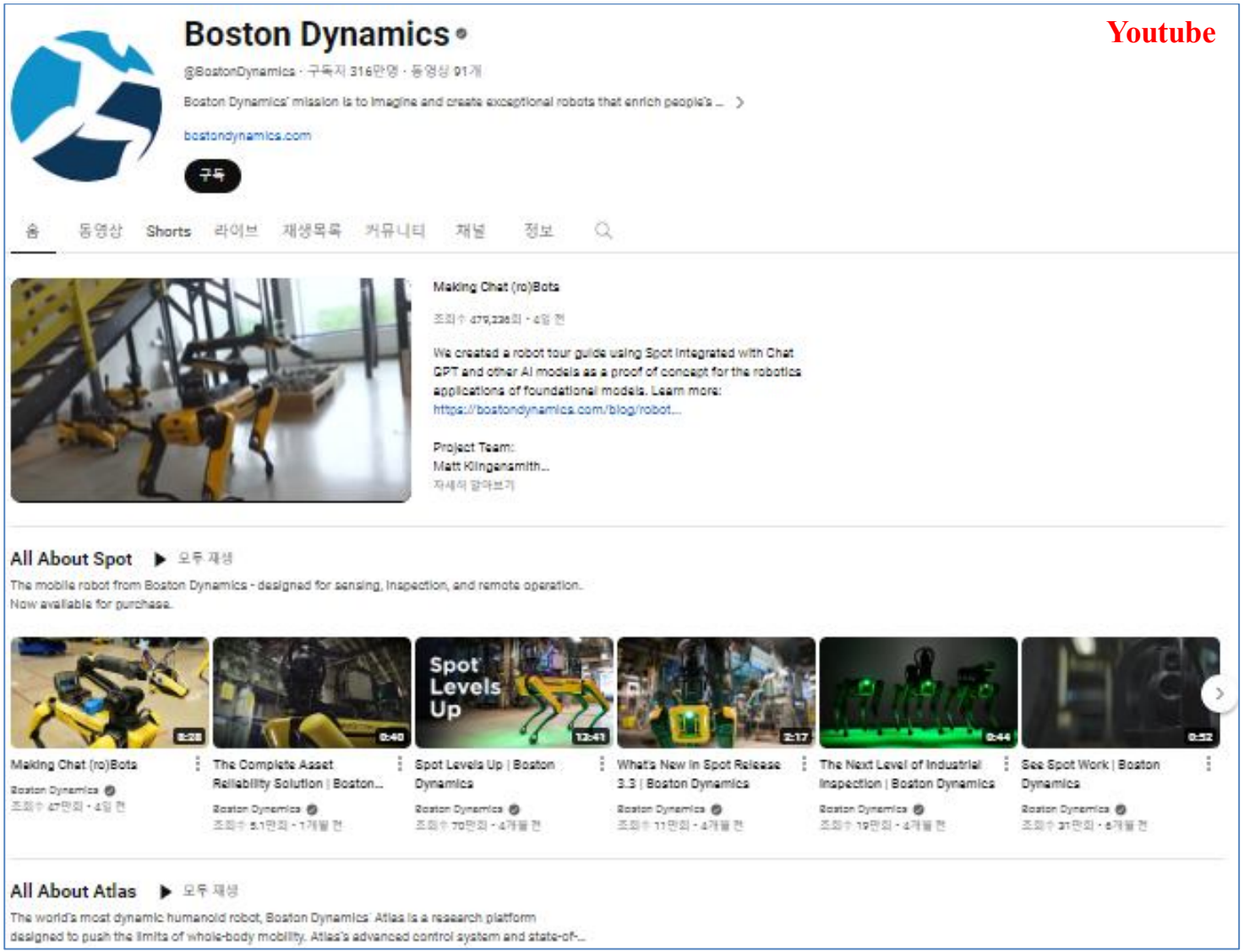
- 수업 중 : 왜 그럴까? 무슨 의미일까? 어떻게 할까? 끊임없이 생각

- 수업 후 복습

Schedule

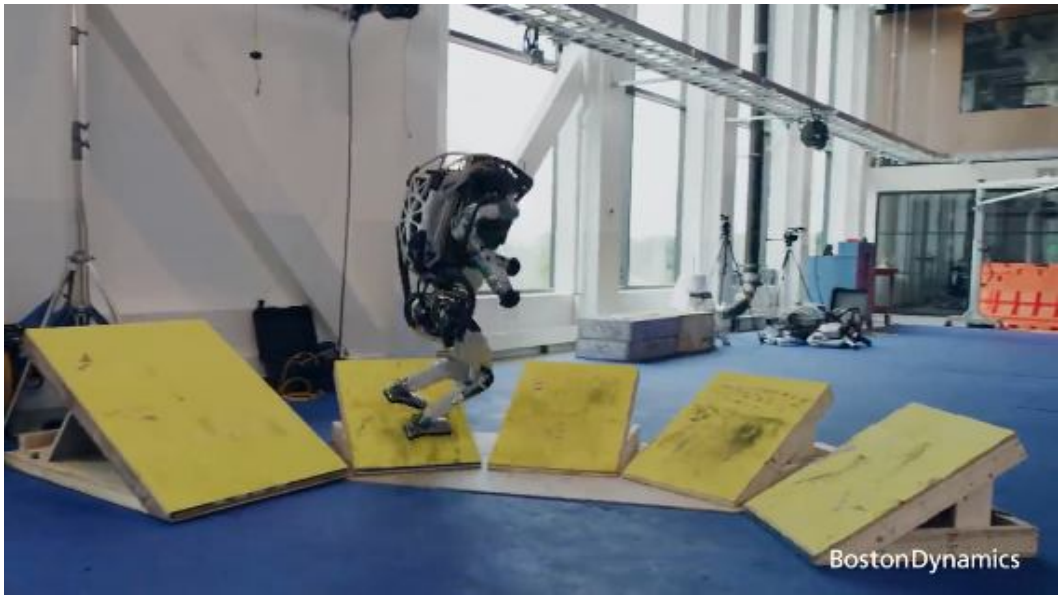
1. 절차지향 프로그래밍
   : C-Programming
      - Visual studio 2022 응용프로그램
   : Image Processing
      - MFC Programming / Jupyter Notebook(Python)
2. Firmware (Embedded) Programming
   : 8bits AVR Core (Atmega128)
      - Sub project 1~4
   : 32bits Cortex-M7 Core (STM32F767)
      - Sub project 5
3. Project – ADAS 자율주행 시스템 SW
   : 자율주행 자동차 CAN 통신
   : ROS(Robot Operating System) - Noetic
      - Turtlebot3 SLAM & Navigation
      - Turtlebot3 Autonomous Driving

Boston Dynamics



Boston Dynamics

2021년 8월 17일 : https://youtu.be/tF4DML7FIWk



2021년 8월 17일 https://youtu.be/EezdinoG4mk

수업목표

로봇 소프트웨어 플랫폼

(하드웨어 추상화, 하위디바이스 제어, 로보틱스 센싱, 인식, 위치추정, 지도 작성 , 내비게이션 등) 중심의 기술과

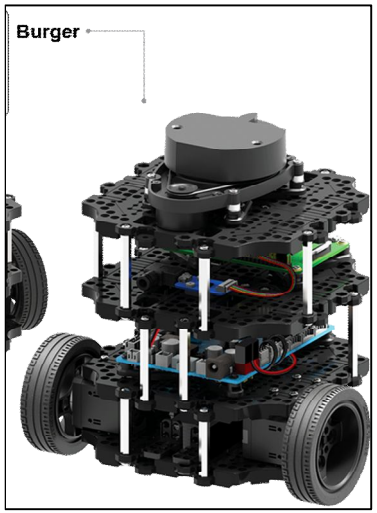
로봇 소프트웨어 플랫 폼(ROS)이 가져올 미래 가치와 기술 선점을 위한 교육을 바탕으로

다양한 영상과 라이다 기반의 로봇 자율주행 기술 교육을 통해 학생들의 첨단 기술습득을 목표로 함.

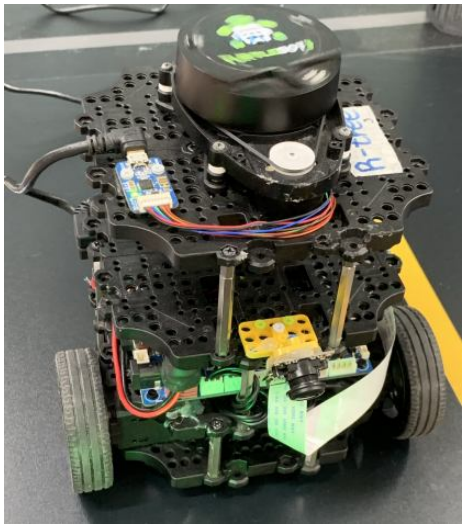
수업방법

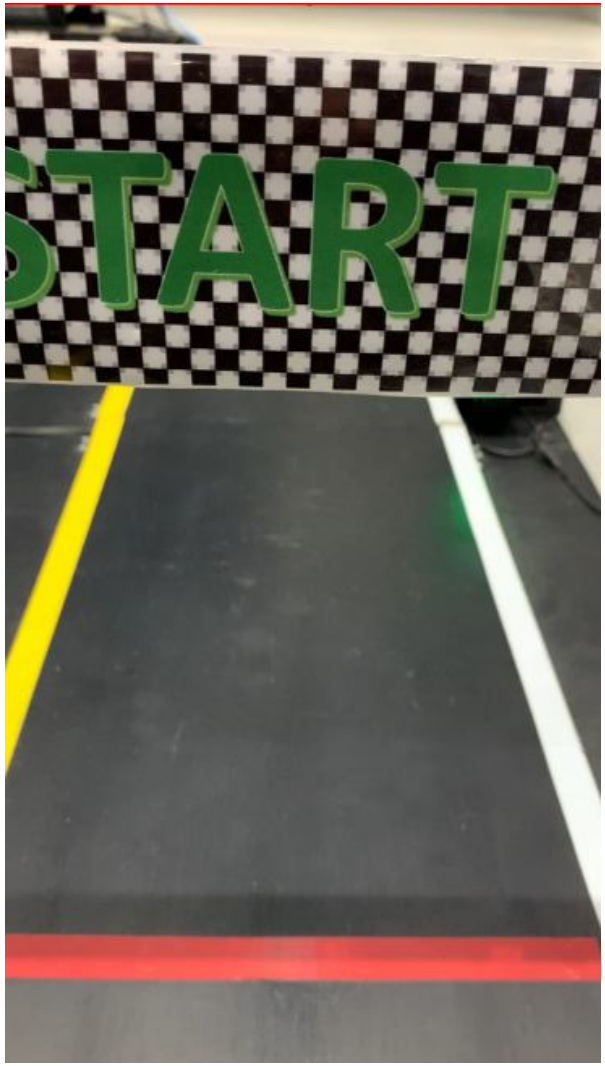
: Ubuntu 기반의 ROS 프로그래밍

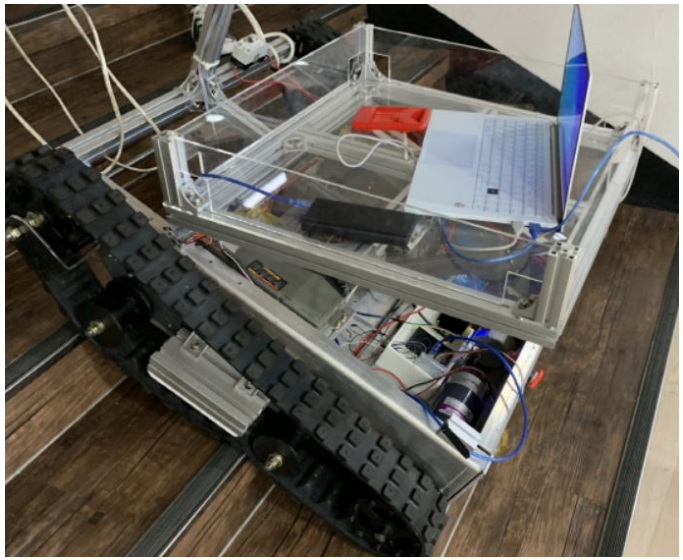
: 하드웨어 플랫폼(터틀봇) 운영



Mission



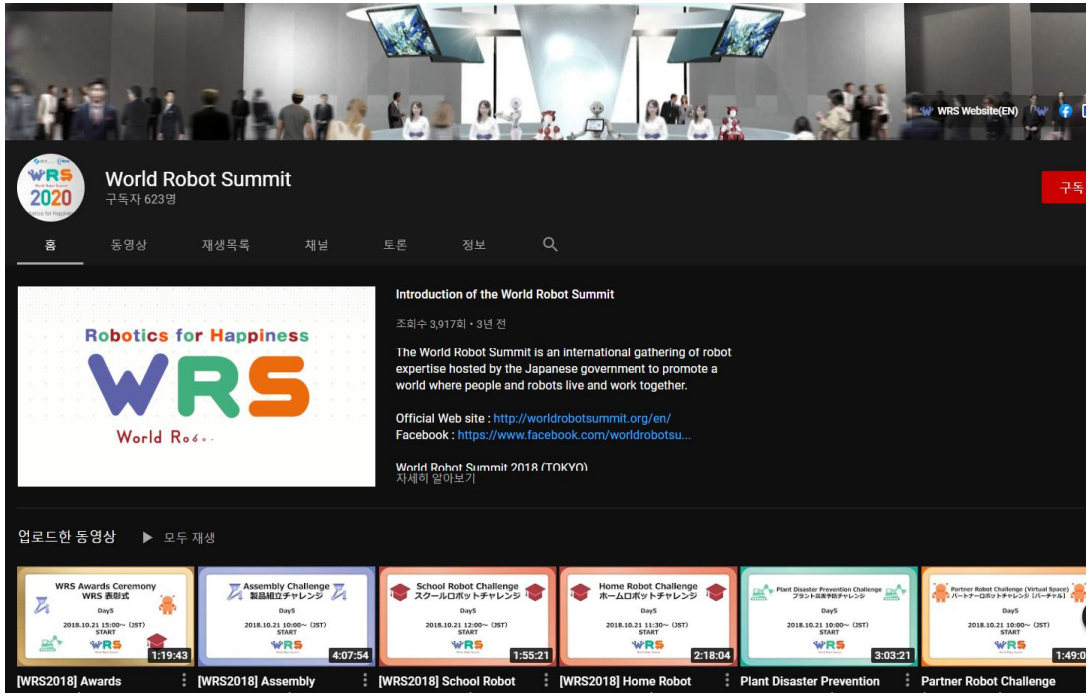




World robot summit

World Robot Summit

Youtube : World Robot Summit



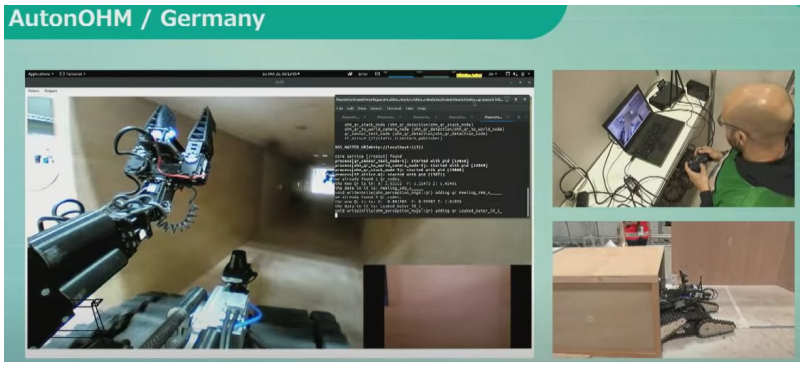
World Robot Summit

: Standard Disaster Robot Challenge

https://youtu.be/r2ZLAu4mTSI

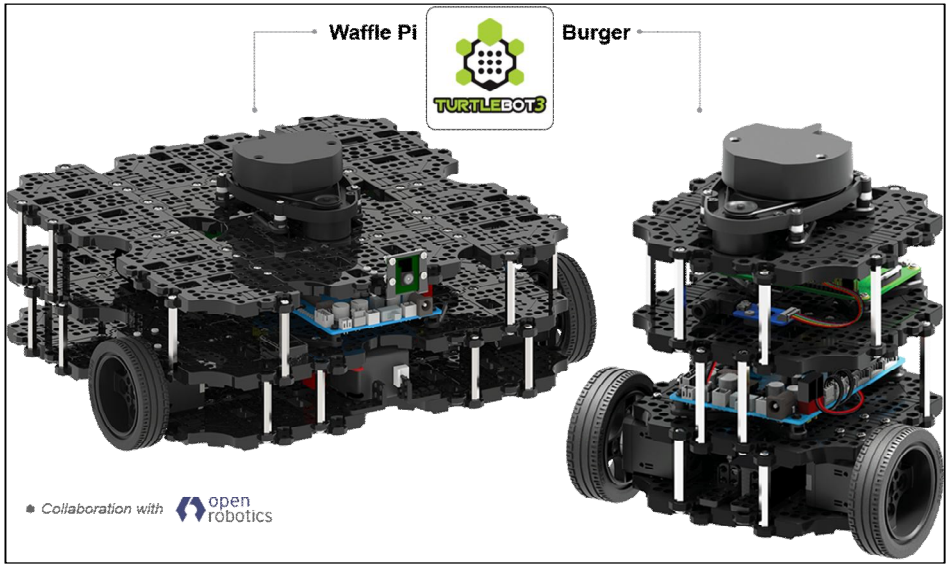
: Plant Disaster Prevention Challenge

https://youtu.be/GoUN_k90h_A



실험 재료 :

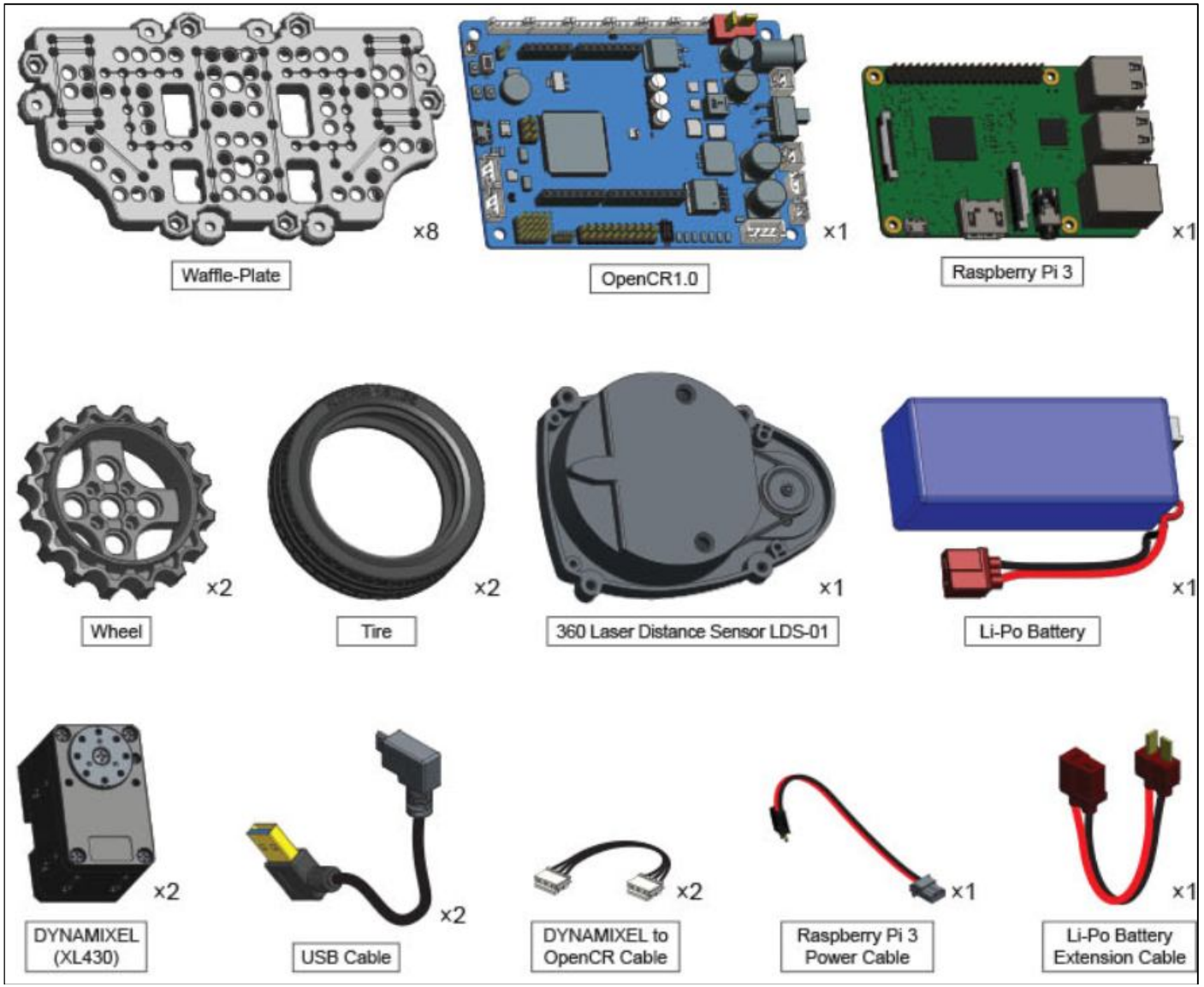

ROBOTIS
TURTLEBOT3
BURGER
TURTLEBOT3


Waffle Pi
TURTLEBOT3
Burger
Collaboration with
open robotics

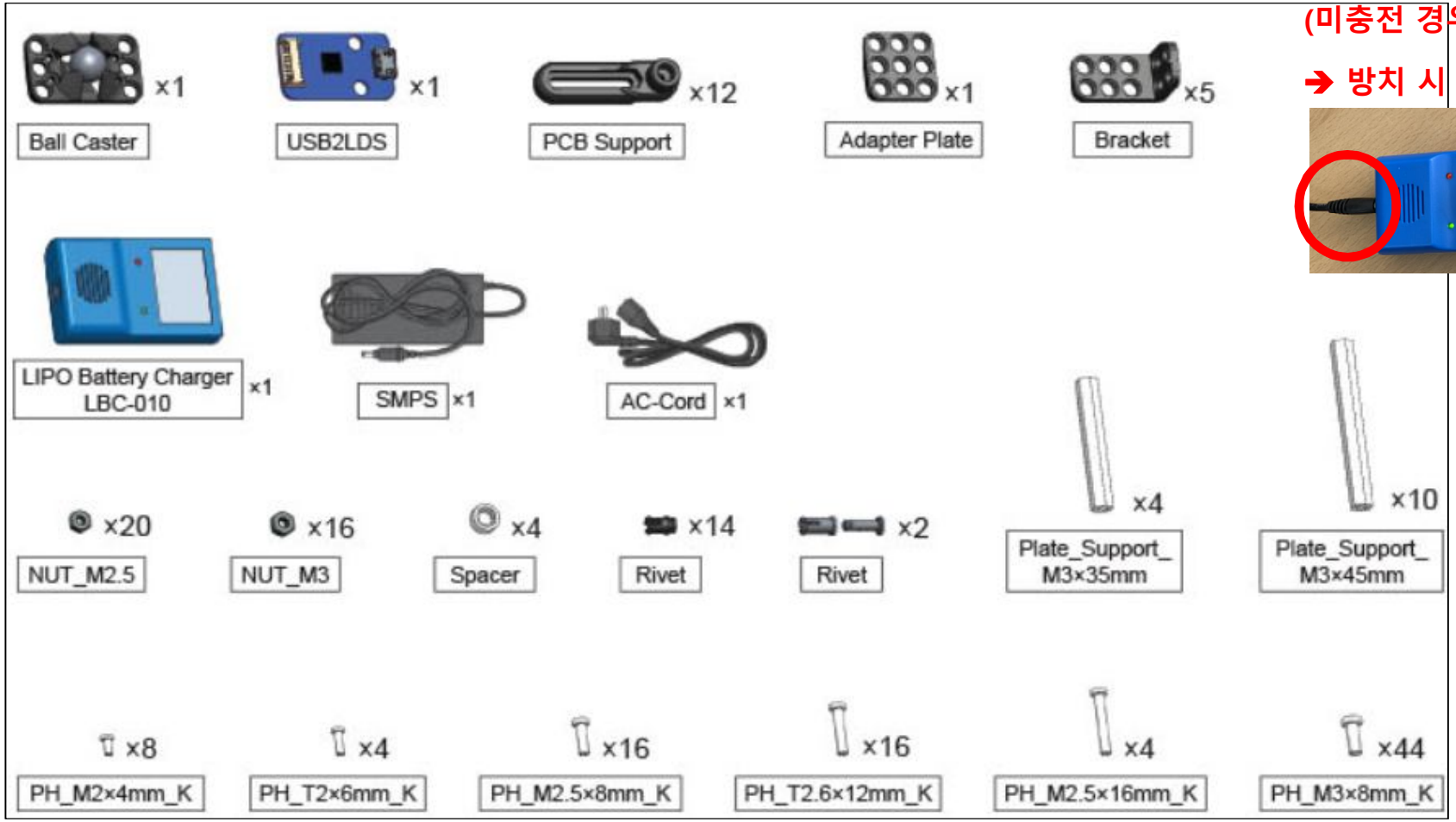

1
ROBOTIS
WWW.ROBOTIS.COM
Transcend
16GB
Raspberry Pi 3
Model B+
2


3
ROBOTIS
WWW.ROBOTIS.COM
4
ROBOTIS
WWW.ROBOTIS.COM
TURTLEBOT3

실험 재료 :


x8
Waffle-Plate
x1
OpenCR1.0
x1
Raspberry Pi 3
x2
Wheel
x2
Tire
x1
360 Laser Distance Sensor LDS-01
x1
Li-Po Battery
x2
DYNAMIXEL (XL430)
x2
USB Cable
x2
DYNAMIXEL to OpenCR Cable
x1
Raspberry Pi 3 Power Cable
x1
Li-Po Battery Extension Cable

실험 재료 :

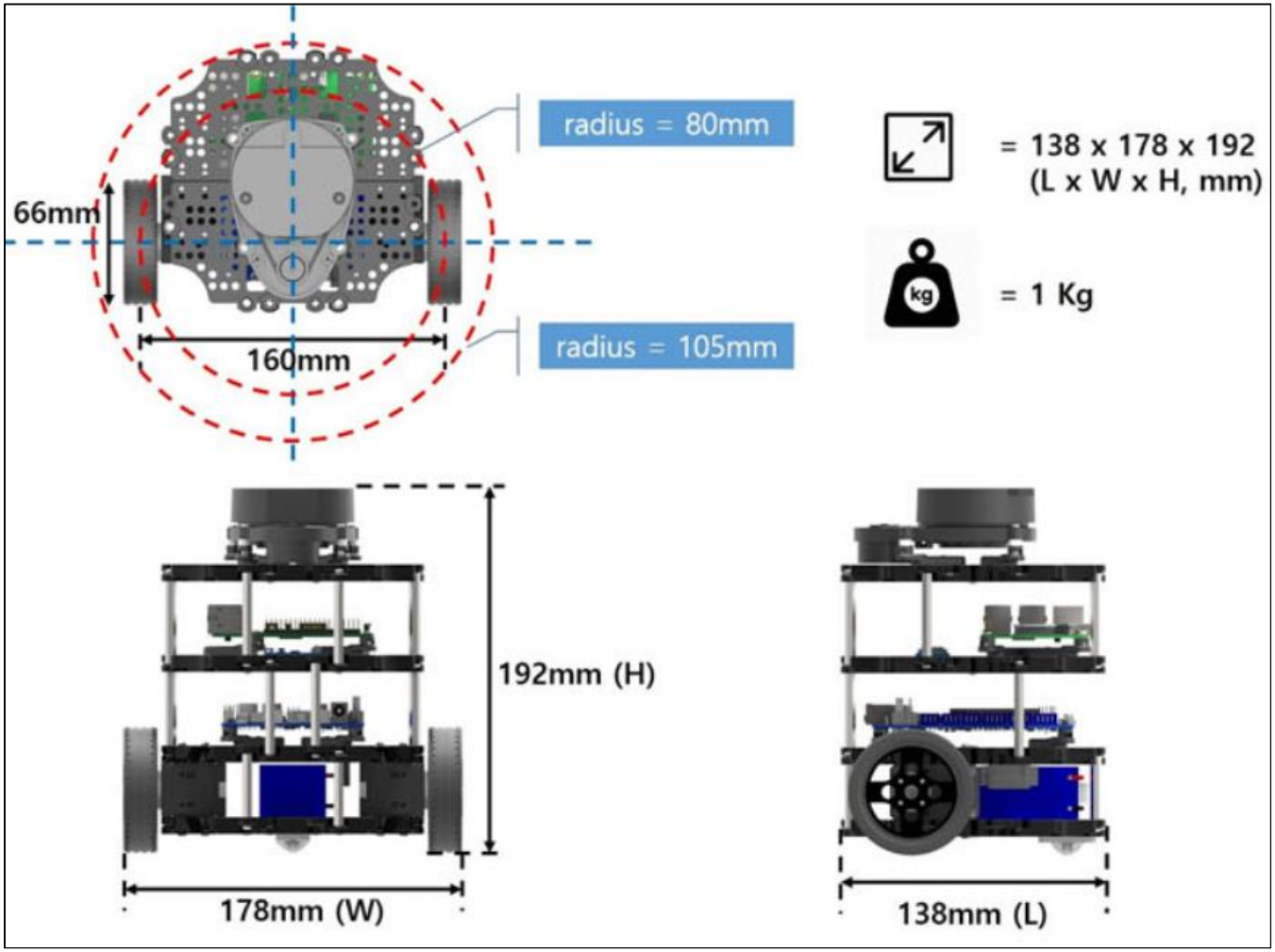


Battery 사전에 충전하기

(미충전 경우 Buzzer 소리 발생)

➔ 방치 시 완전 방전됨

실험 재료 :


radius = 80mm
66mm
160mm
radius = 105mm
= 138 x 178 x 192
(L x W x H, mm)
kg
= 1 Kg
192mm (H)
178mm (W)
138mm (L)

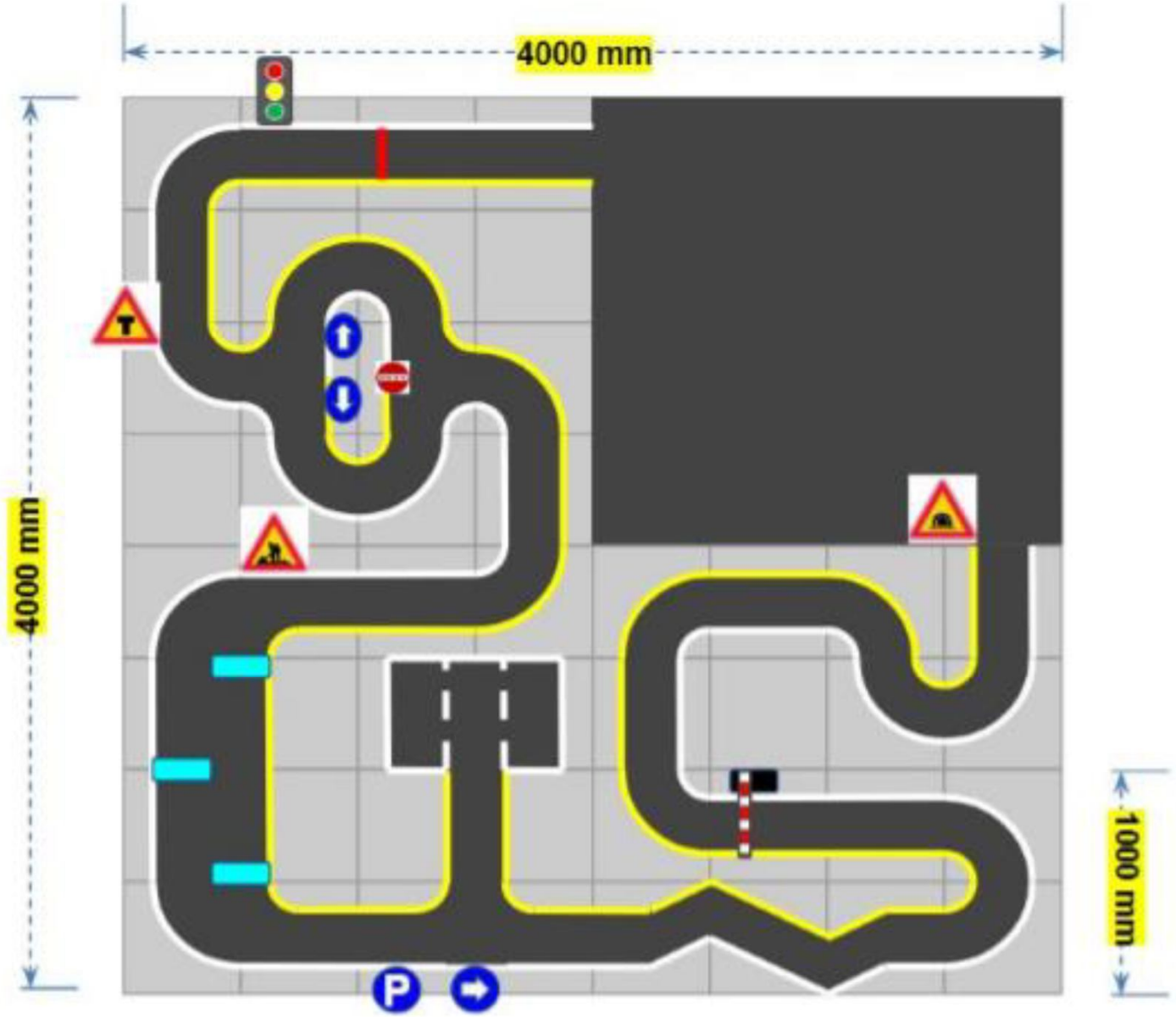
실험 재료 :

경기장 :

- red, yellow, blue 신호등 인식 후 출발
- 교차로 신호 인식 후 이동
- 장애물 회피 이동
- 비워 있는 주차공간에 주차
- 건널목 정지후 이동
- SLAM, Navigation 이동

경기장 : 4m x 4m (ref. 전자정보관 5층)

경기장 :



1	Traffic Mission
2	Intersection
3	Obstacles
4	Parking
5	Stop Bar
6	Tunnel

Reference books

1. Reference

(1) Source code : github.com의 ROBOTIS-GIT

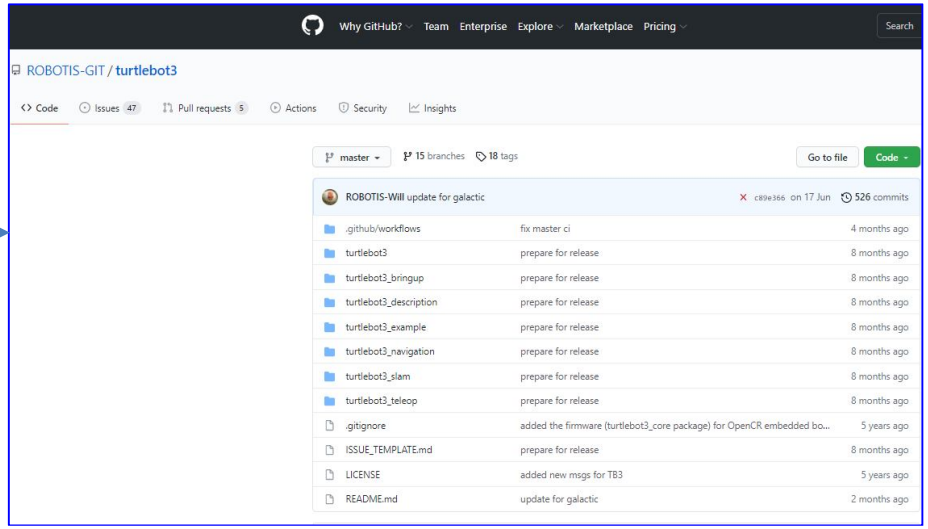
https://github.com/ROBOTIS-GIT/turtlebot3

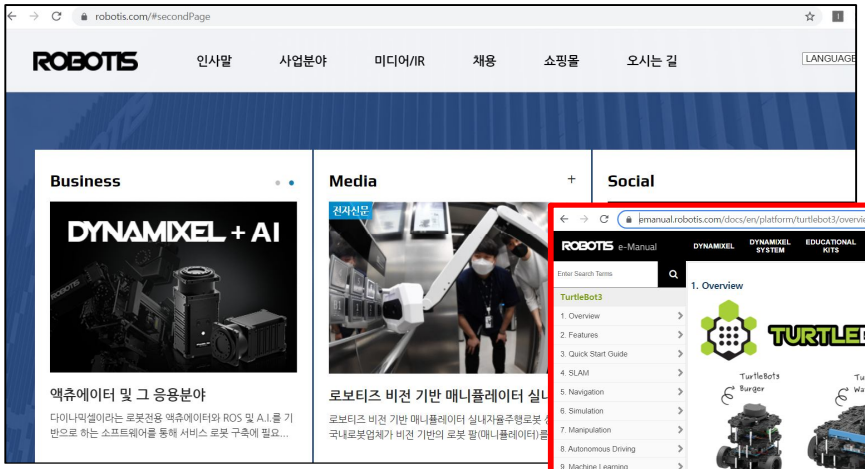
(2) OpenCR (Control module for ROS) : github.com의 ROBOTIS-GIT

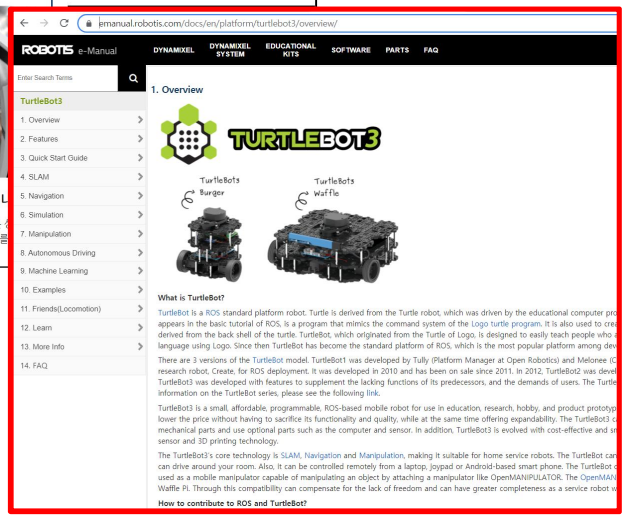
https://github.com/ROBOTIS-GIT/OpenCR

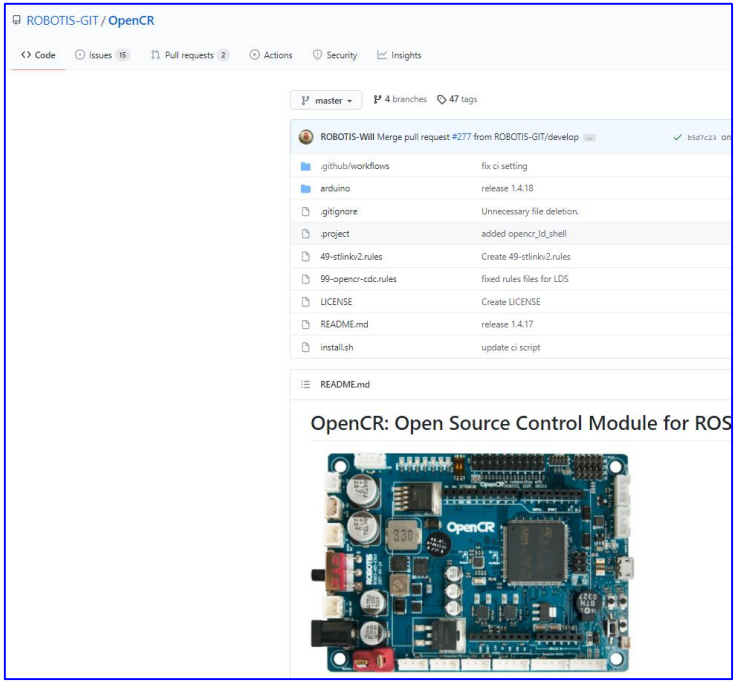
(3) Robotis e-manual : https://www.robotis.com

https://emanual.robotis.com/docs/en/platform/turtlebot3/overview









Reference books

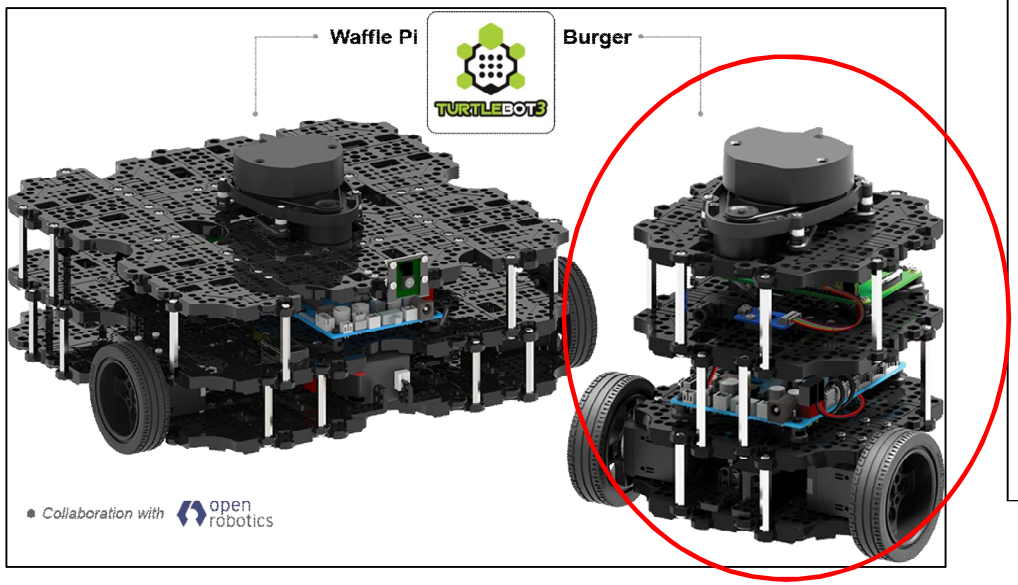
2. Reference books

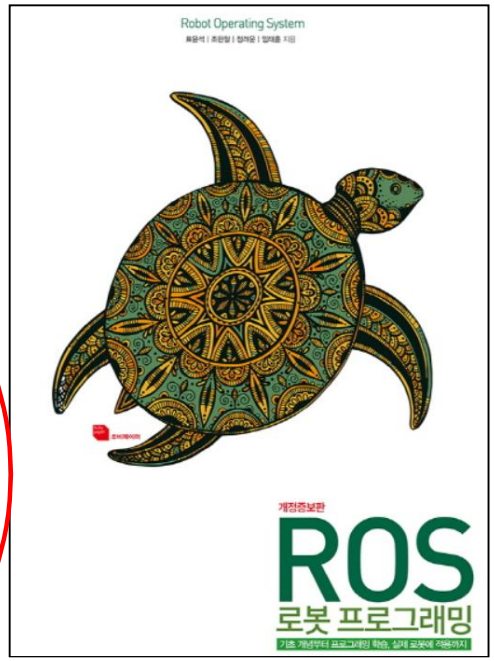
(1) 주교재

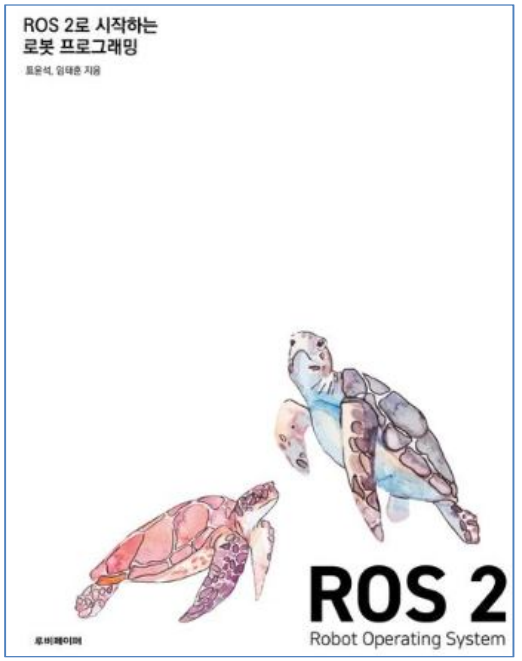
: ROS 로봇 프로그래밍 : Ruby paper

: Robotis e-manual : https://emanual.robotis.com/docs/en/platform/turtlebot3/overview

(2) 실험 재료 : Turtlebot3 burger







Reference books

3. Reference open source 

(1) source download 방법

❶ 리눅스에서 직접 내려받기 - Ubuntu 기반의 ROS 프로그래밍

: git 명령어를 이용하기 위해 git 를 설치

: 터미널 창을 열어 다음과 같이 git 를 설치

$ sudo apt-get install git

: ros_tutorials 패키지

$ git clone https:// github.com/ROBOTIS-GIT/ros_tutorials.git

❷ 예제 tutorial source

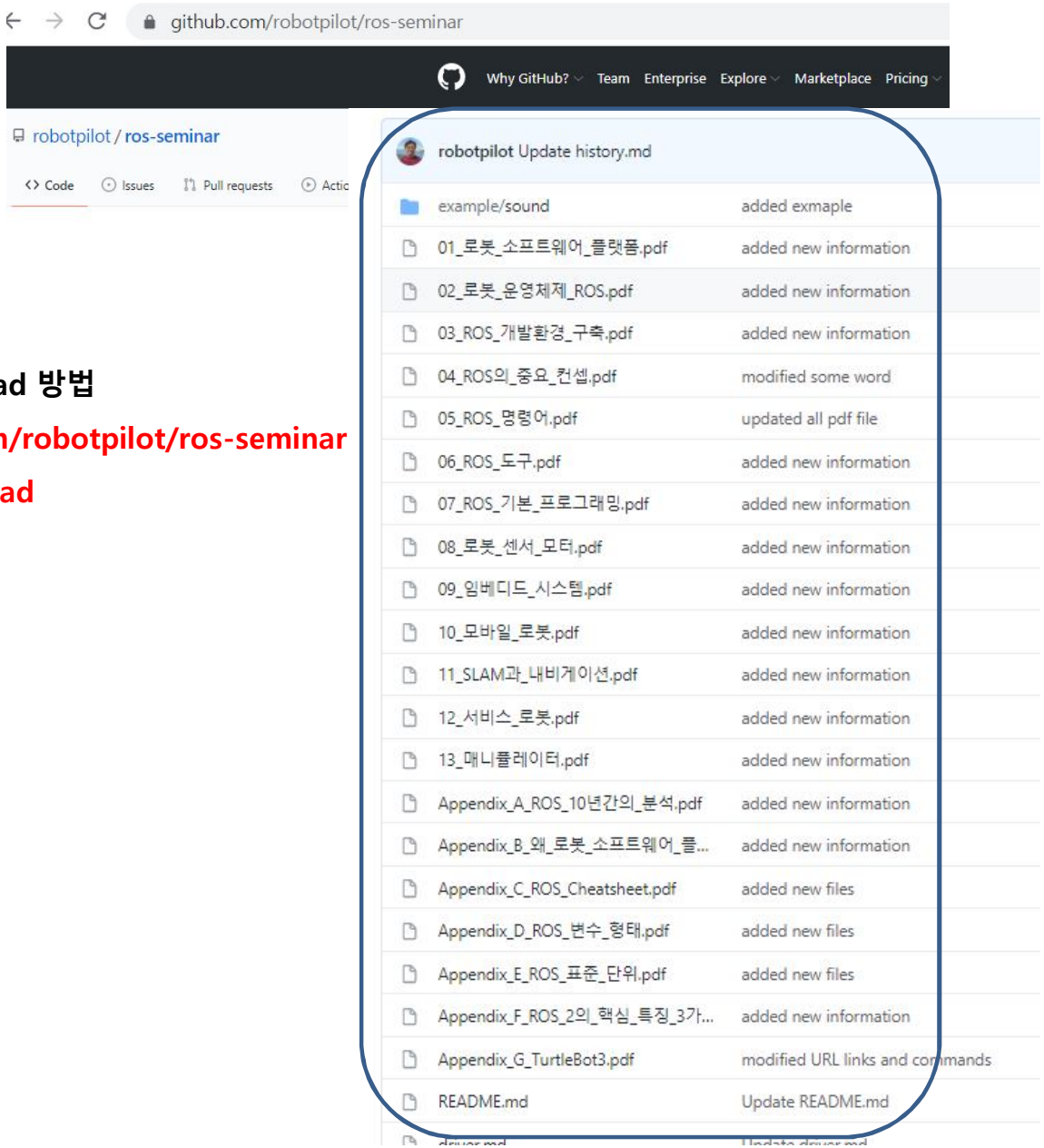
: https://github.com/ROBOTIS-GIT/ros_tutorials

4. Reference community

: 오픈 소스 로봇기술 공유 커뮤니티: 오로카 에서 공개 강좌와 공개 프로젝트에 대한 소개

- 오로카(OROCA) http://www.oroca.org
- 한국 ROS 페이스북 그룹 https://www.facebook.com/groups/427060150825583/
- 한국 ROS 사용자 구글 그룹 https://groups.google.com/forum/#!forum/korea-ros-users
- ROS Discourse 한국 채널 https://discourse.ros.org/c/local/korea
- ROS Discourse https://discourse.ros.org/
- ROS QnA http://answers.ros.org/questions/

Reference books



5. Reference documents

(1) documents download 방법

: https://github.com/robotpilot/ros-seminar

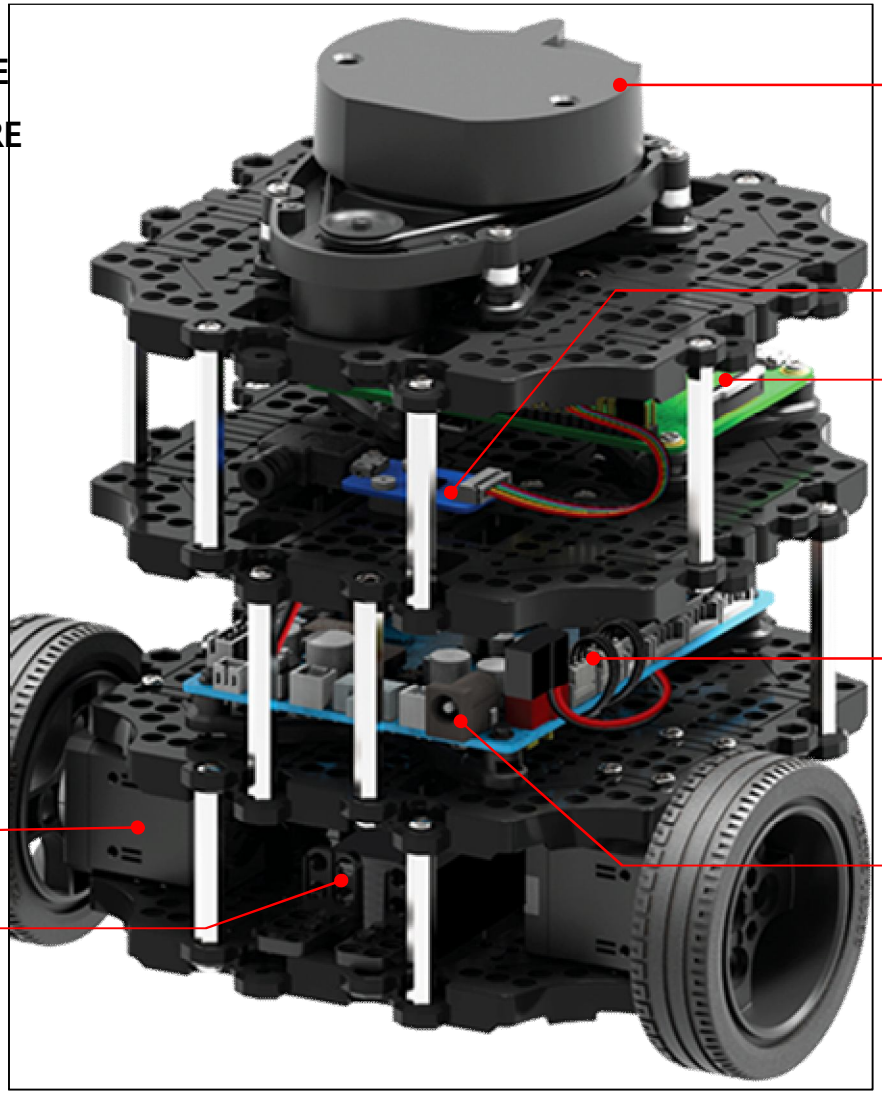
➔ 강의자료 download

Turtlebot3 burger : Standard Specification

1. Specification

(1) OPEN SOURCE SOFTWARE

(2) OPEN SOURCE HARDWARE



360° LiDAR
: HLS-LFCD LDS(Laser Distance Sensor)
: Detection distance 120mm~3500mm
: Angular Resolution 1 degree

: USB2LDS (115200 baudrate)

SBC(Single Board Computer)
: Raspberry PI 3 B+ / PI 4
- Broadcom BCM2837B0 / 2711
- Cortex-A53 64bits Quad-core
- 1.4GHz frequency

OpenCR(Control module for ROS)
: STM32F746
- ARM Cortex-M7 32bits RISC core
- 216MHz frequency

Input 12V 5A

DYNAMIXEL
: XL430-W350-T
: Stall Torque
- 4.1 [N.m] (at 12.0 [V], 2.3 [A])
: No Load Speed
- 46 [rev/min] (at 12.0 [V])

Li-Po battery 11.1V 1800mAh

Turtlebot3 burger : Standard Specification

: Ref. https://www.ros.org/reps/rep-0003.html#noetic-ninjemys-may-2020-may-2025

: ROS 1.0

=> Kinetic Kame

=> Noetic

Kinetic Kame (May 2016 - May 2021)

Required Support for:

- Ubuntu Wily (15.10)
- Ubuntu Xenial (16.04)

Ubuntu 16.04.7 LTS (Xenial Xerus)
: Raspberry PI 3 B+

Recommended Support for:

- Debian Jessie
- Fedora 23
- Fedora 24

Minimum Requirements:

- C++11
  - GCC 4.9 on Linux, as it's the version that Debian Jessie ships with
- Python 2.7
  - Python 3.4 not required, but testing against it is recommended
- Lisp SBCL 1.2.4
- CMake 3.0.2
  - Debian Jessie ships with CMake 3.0.2
- Boost 1.55
  - Debian Jessie ships with Boost 1.55

Exact or Series Requirements:

- Ogre3D 1.9.x
- Gazebo 7
- PCL 1.7.x
- OpenCV 3.x
- Qt 5.3.x
- PyQt5

Lunar Loggerhead (May 2017 - May 2019)

Required Support for:

- Ubuntu Xenial (16.04)
- Ubuntu Yakkety (16.10)
- Ubuntu Zesty (17.04)

Ubuntu 16.04.7 LTS (Xenial Xerus)

Melodic Morenia (May 2018 - May 2023)

Required Support for:

- Ubuntu Artful (17.10)
- Ubuntu Bionic (18.04)

Ubuntu 18.04.5 LTS (Bionic Beaver)

Noetic Ninjemys (May 2020 - May 2025)

Required Support for:

- Ubuntu Focal Fossa (20.04)

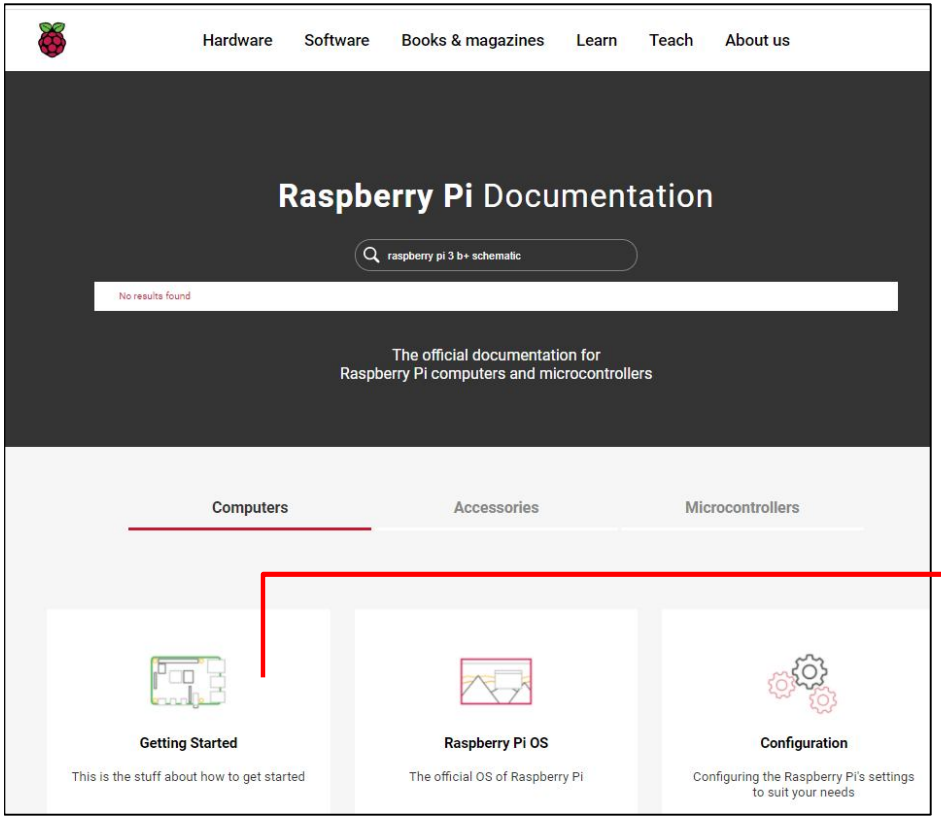
Ubuntu 20.04.3 LTS (Focal Fossa)
: Raspberry PI 4B

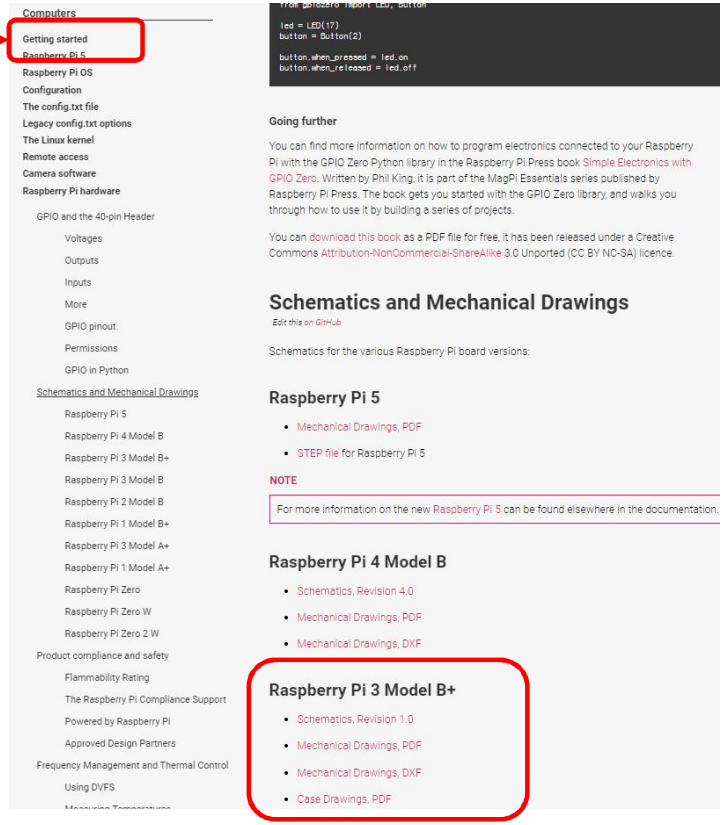
Turtlebot3 burger : Standard Specification

2. Specification of Parts

(1) SBC(Single Board Computer) : Raspberry PI 3 B+

- https://www.raspberrypi.org/documentation/
- https://www.raspberrypi.org/documentation/computers/raspberry-pi.html#raspberry-pi-3-model-b
- Broadcom BCM2837B0 / Cortex-A53 64bits Quad-core / 1.4GHz frequency



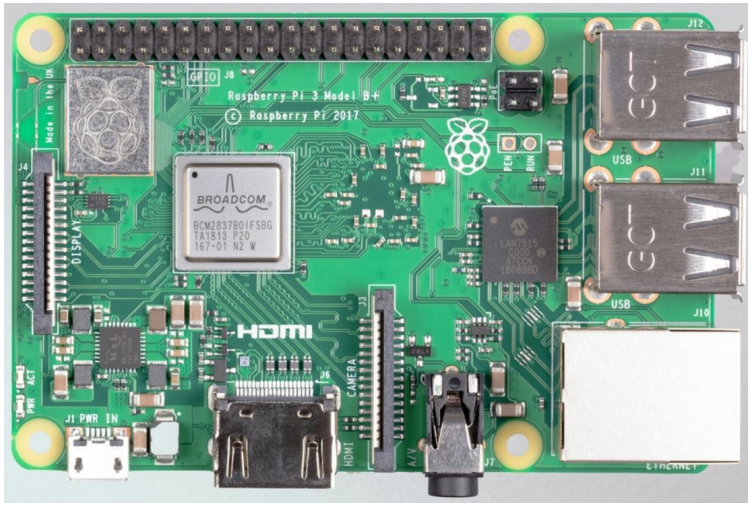


Turtlebot3 burger : Standard Specification

2. Specification of Parts

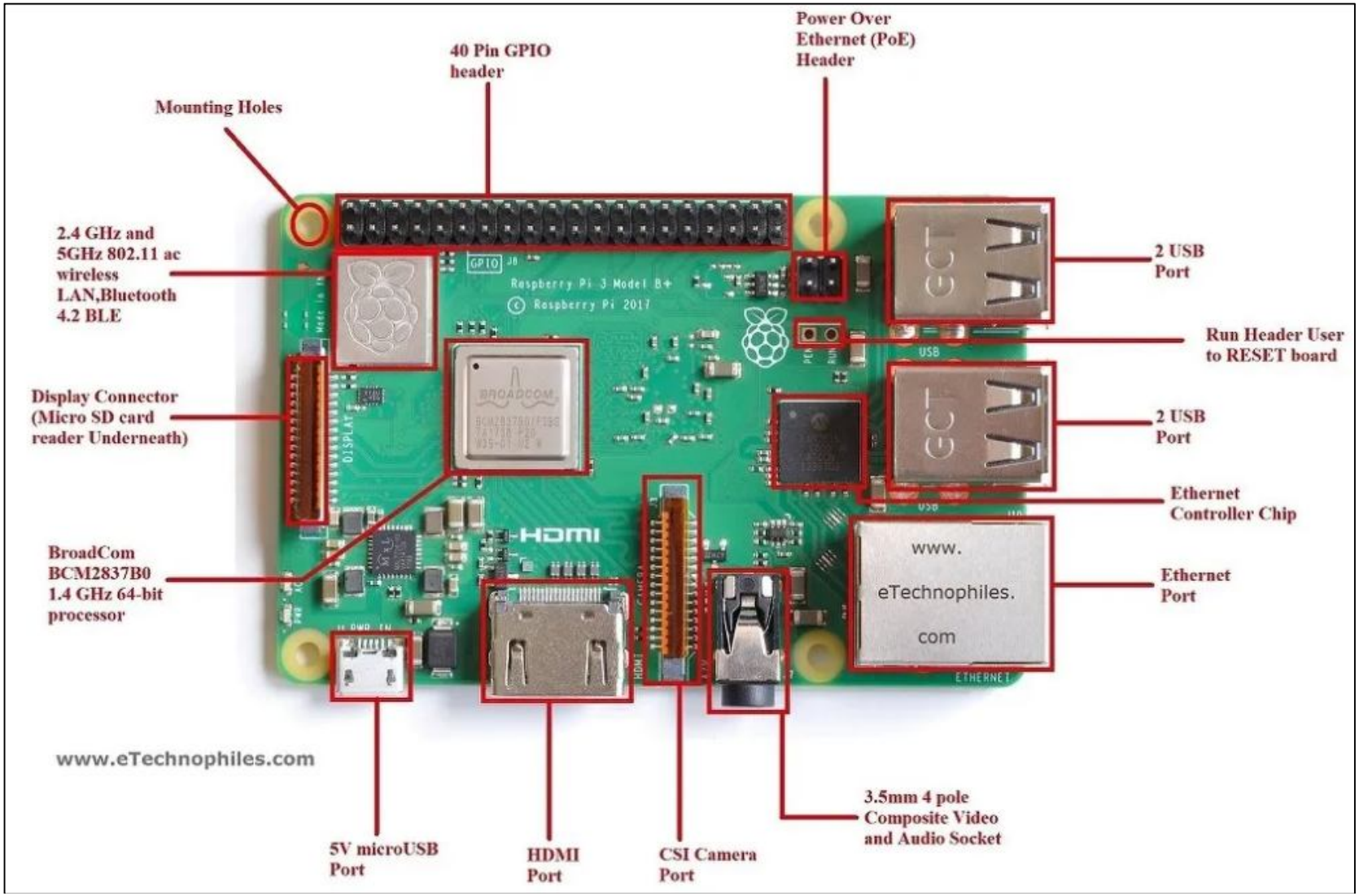
(1) SBC(Single Board Computer) : Raspberry PI 3 B+ : https://www.raspberrypi.org/forums/

Processor:	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory:	1GB LPDDR2 SDRAM
Connectivity:	■ 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE ■ Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) ■ 4 × USB 2.0 ports
Access:	Extended 40-pin GPIO header
Video & sound:	■ 1 × full size HDMI ■ MIPI DSI display port ■ MIPI CSI camera port ■ 4 pole stereo output and composite video port
Multimedia:	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card support:	Micro SD format for loading operating system and data storage
Input power:	■ 5V/2.5A DC via micro USB connector ■ 5V DC via GPIO header ■ Power over Ethernet (PoE)–enabled (requires separate PoE HAT)

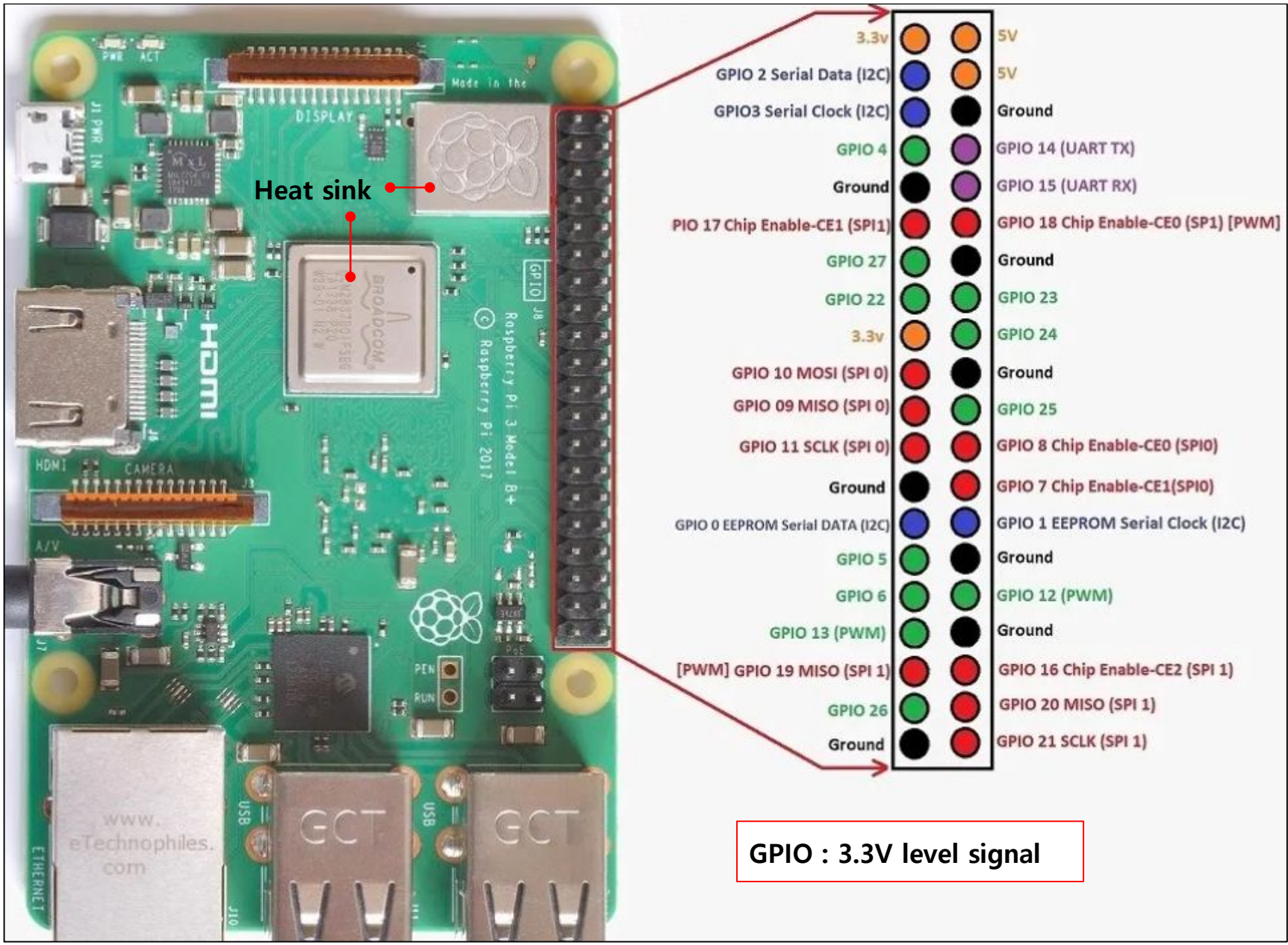


Turtlebot3 burger : Standard Specification

(1) SBC(Single Board Computer) : Raspberry PI 3 B+ : https://www.raspberrypi.org/forums/



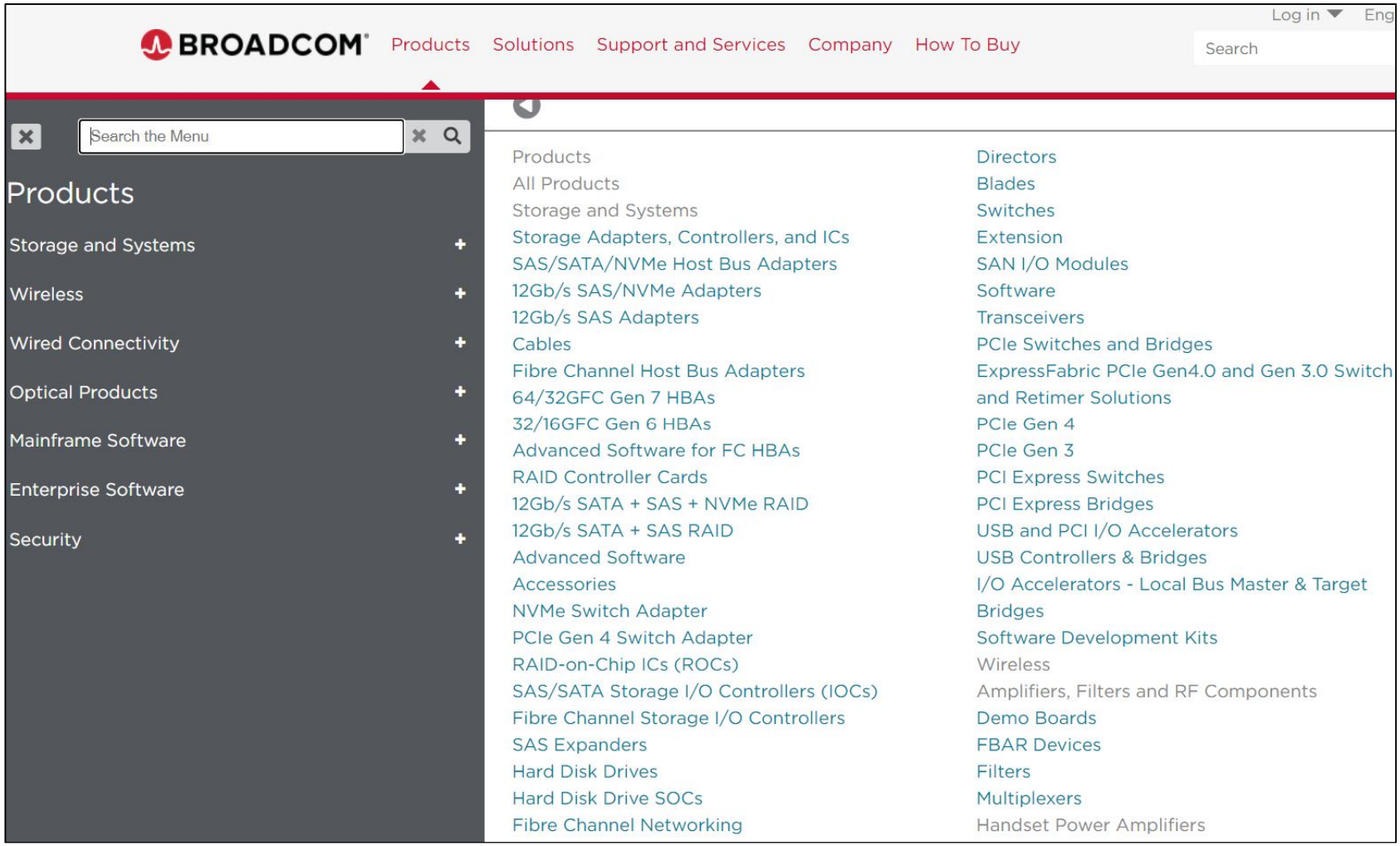
: Raspberry PI 3 B+ pin map



: **Broadcom BCM2837B0 / Cortex-A53 64bits Quad-core**

https://www.broadcom.com/

http://www.arm.com/products/processors/cortex-a/cortex-a53-processor.php

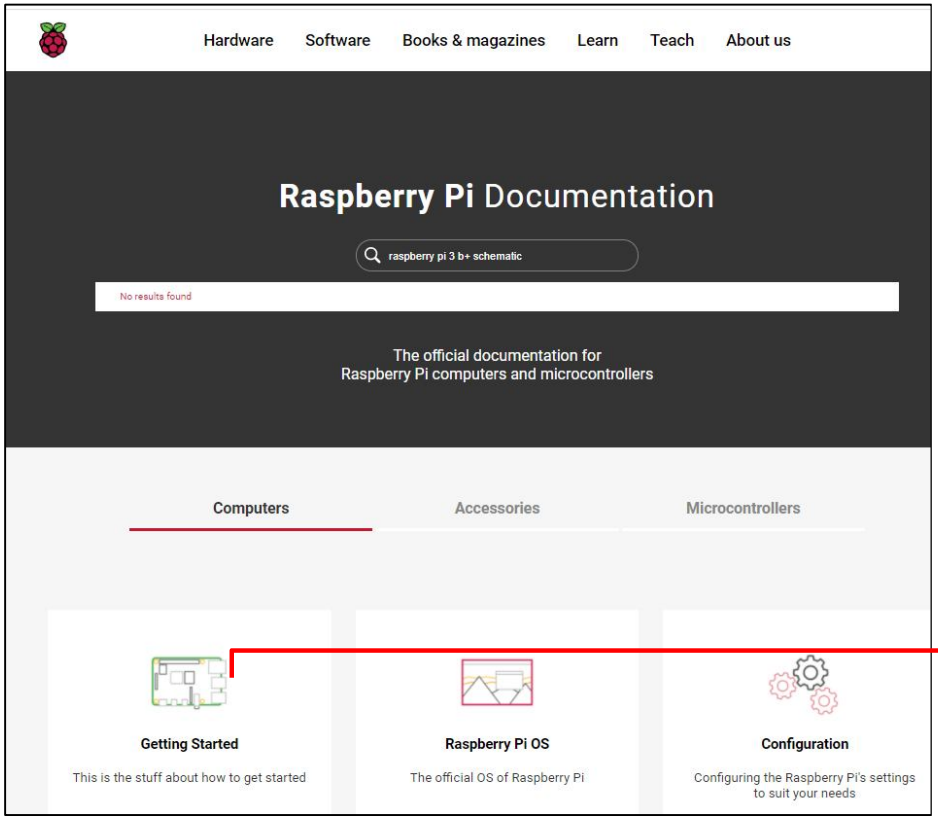


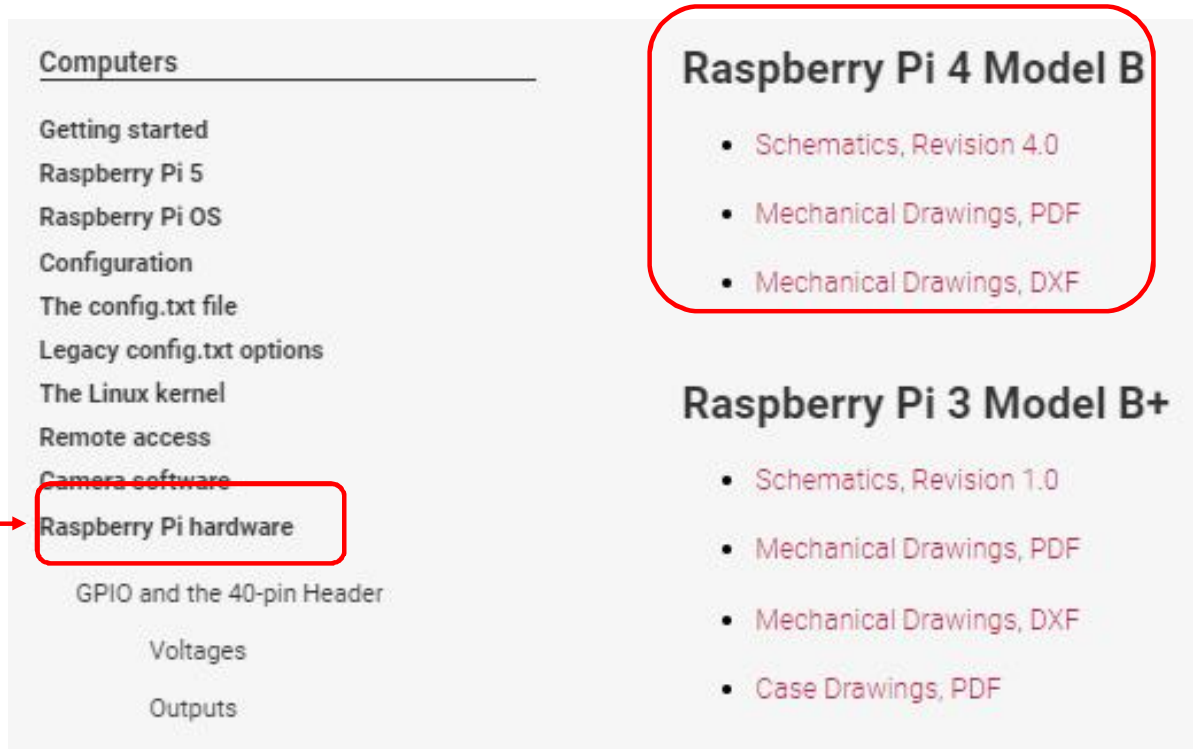
: Raspberry PI 3 B+

	Pi 1 Model A	Pi 1 Model A+	Pi 1 Model B	Pi 1 Model B+	Pi 2 Model B	Pi 3 Model B
판매금액	US$25	US$20	US$35			
Processor chipset	Broadcom BCM2835 700MHz Single-Core 32Bit ARM1176JZF-S 32bits				Broadcom BCM2836 900MHz Quad-Core 32Bit ARM Cortex-A7	64bits Broadcom BCM2837 1.2GHz Quad-Core 64Bit ARM Cortex-A53
Ethernet	No		10/100 BaseT Ethernet socket			
Wi-Fi	No					BCM43143 (802.11 b/g/n Wireless LAN)
Bluetooth	No					Bluetooth 4.1 (Bluetooth Classic and LE)
GPU	Dual Core VideoCore IV® Multimedia Co-Processor, Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30, H.264 high-profile decode, Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure					
Memory(SDRAM)	256 MB (GPU와 공유)		512 MB (GPU와 공유)		1GB LPDDR2	
USB 2.0	1 Port (direct from BCM2835 chip)		2 Ports (보드에서 3포트 USB 허브 지원)	4 Ports (보드에서 5포트 USB 허브 지원)		
Camera Connector	라즈베리 파이 카메라와 라즈베리파이 NoIR 카메라를 연결하기 위한 15-pin MIPI Camera Serial Interface (CSI-2)					
Video Output	HDMI (rev 1.3 & 1.4), PAL과 NTSC 기준에 맞는 640×350 부터 1920×1200 이상의 14개의 해상도, Composite RCA (PAL & NTSC)	HDMI (rev 1.3 & 1.4), PAL과 NTSC 기준에 맞는 640×350 부터 1920×1200 이상의 14개의 해상도, Composite RCA (PAL & NTSC, 오디오 출력도 겸함)				
Display Connector	Display Serial Interface (DSI) 15 way flat flex cable connector					
Audio Output	3.5mm jack을 통한 오디오, HDMI를 통한 디지털 오디오, I²S					
Storage	SD / MMC / SDIO 카드 슬롯 (카드에 3.3 V가 공급될때)	Micro SD	SD / MMC / SDIO 카드 슬롯	Micro SD		
GPIO Connector (Low-Level Peripherals)	26-pin (GPIO, UART, I²C, SPI, +3.3V, +5V, GND)	40-pin (GPIO, UART, I²C, SPI, +3.3V, +5V, GND)	26-pin (GPIO, UART, I²C, SPI, +3.3V, +5V, GND)	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip (GPIO, UART, I²C, SPI, +3.3V, +5V, GND)		
Power	5V 1.8A					5V 2.5A(2.4A)

(1) SBC(Single Board Computer) : Raspberry PI 4B

- https://www.raspberrypi.org/documentation/
- https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#raspberry-pi-4-model-b



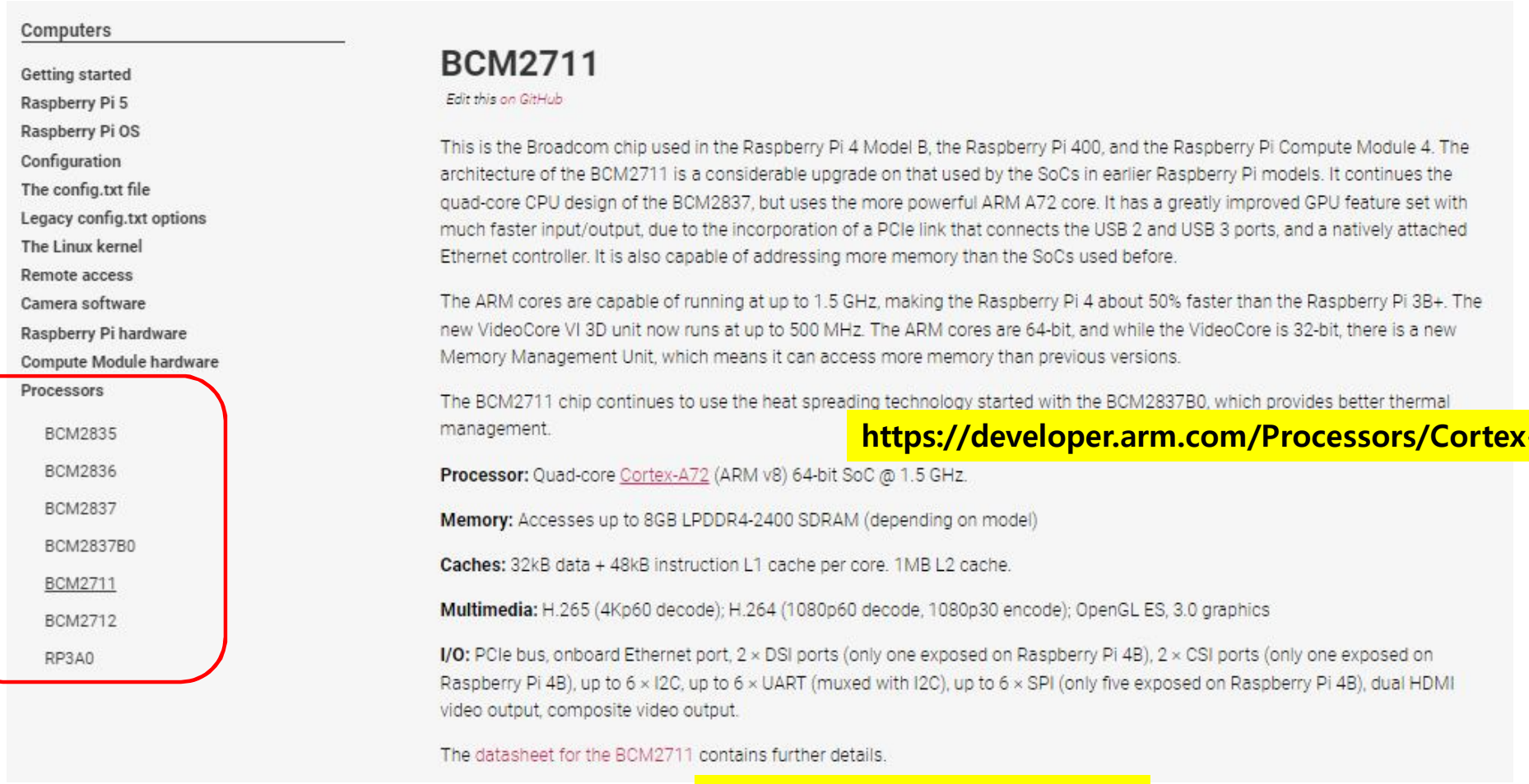


: Raspberry Pi 4 Broadcom BCM2711 quad-core Arm Cortex A72 processor @ 1.5GHz

: Raspberry Pi 5 Broadcom BCM2712 quad-core Arm Cortex A76 processor @ 2.4GHz

(1) SBC(Single Board Computer) : Raspberry PI processor

- https://www.raspberrypi.com/documentation/computers/processors.html#bcm2711



Computers

Getting started
Raspberry Pi 5
Raspberry Pi OS
Configuration
The config.txt file
Legacy config.txt options
The Linux kernel
Remote access
Camera software
Raspberry Pi hardware
Compute Module hardware
Processors

BCM2835
BCM2836
BCM2837
BCM2837B0
BCM2711
BCM2712
RP3A0

BCM2711

Edit this on GitHub

This is the Broadcom chip used in the Raspberry Pi 4 Model B, the Raspberry Pi 400, and the Raspberry Pi Compute Module 4. The architecture of the BCM2711 is a considerable upgrade on that used by the SoCs in earlier Raspberry Pi models. It continues the quad-core CPU design of the BCM2837, but uses the more powerful ARM A72 core. It has a greatly improved GPU feature set with much faster input/output, due to the incorporation of a PCIe link that connects the USB 2 and USB 3 ports, and a natively attached Ethernet controller. It is also capable of addressing more memory than the SoCs used before.

The ARM cores are capable of running at up to 1.5 GHz, making the Raspberry Pi 4 about 50% faster than the Raspberry Pi 3B+. The new VideoCore VI 3D unit now runs at up to 500 MHz. The ARM cores are 64-bit, and while the VideoCore is 32-bit, there is a new Memory Management Unit, which means it can access more memory than previous versions.

The BCM2711 chip continues to use the heat spreading technology started with the BCM2837B0, which provides better thermal management.

Processor: Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5 GHz.

Memory: Accesses up to 8GB LPDDR4-2400 SDRAM (depending on model)

Caches: 32kB data + 48kB instruction L1 cache per core. 1MB L2 cache.

Multimedia: H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); OpenGL ES, 3.0 graphics

I/O: PCIe bus, onboard Ethernet port, 2 × DSI ports (only one exposed on Raspberry Pi 4B), 2 × CSI ports (only one exposed on Raspberry Pi 4B), up to 6 × I2C, up to 6 × UART (muxed with I2C), up to 6 × SPI (only five exposed on Raspberry Pi 4B), dual HDMI video output, composite video output.

The datasheet for the BCM2711 contains further details.

https://developer.arm.com/Processors/Cortex-A72

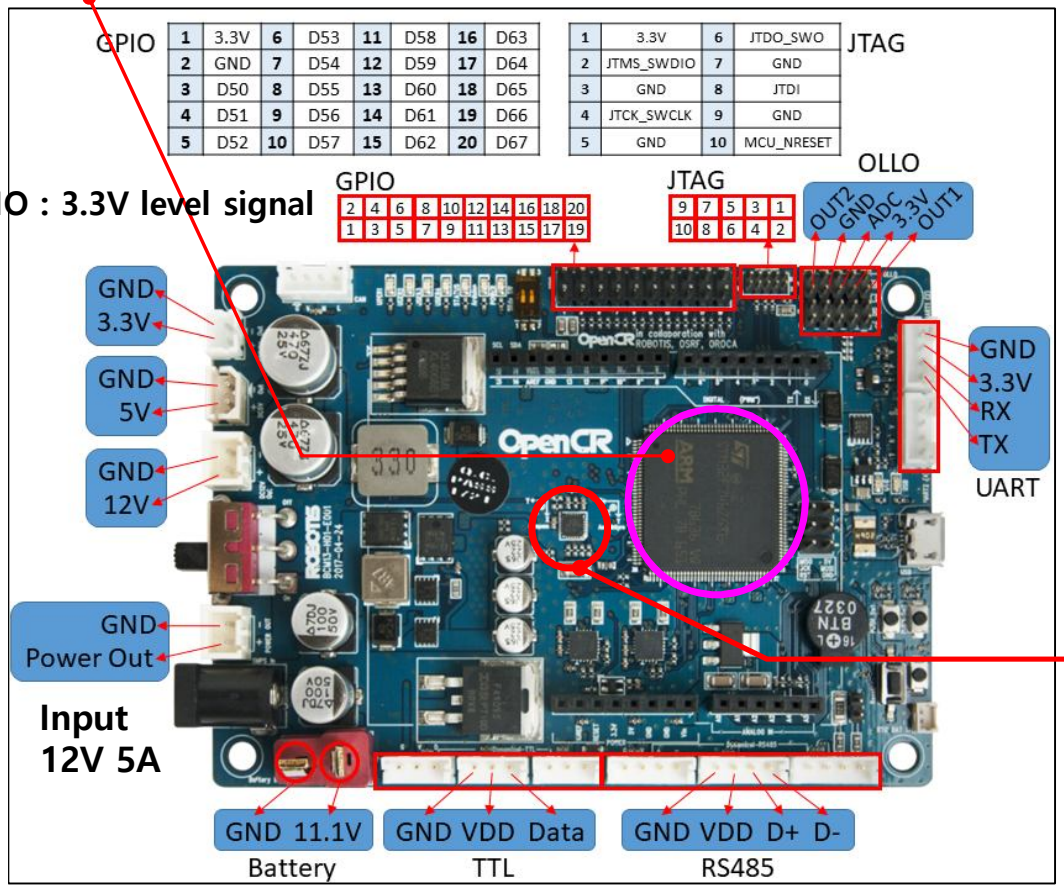
https://www.broadcom.com/

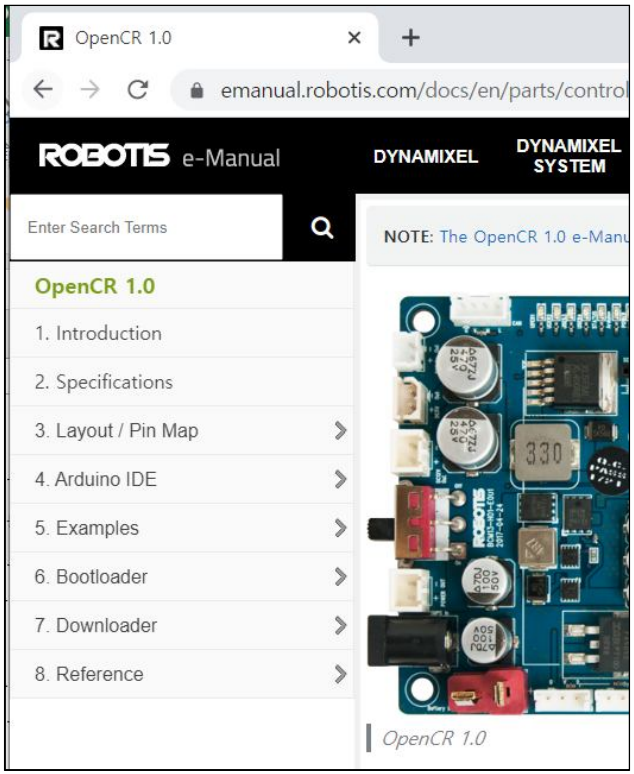
Turtlebot3 burger : Standard Specification

2. Specification of Parts

(2) OpenCR(Control module for ROS) : STM32F746

- https://emanual.robotis.com/docs/en/parts/controller/opencr10/
- ARM Cortex-M7 32bits RISC core / 216MHz frequency





IMU(Inertial Measurement Unit, 관성측정장치) :

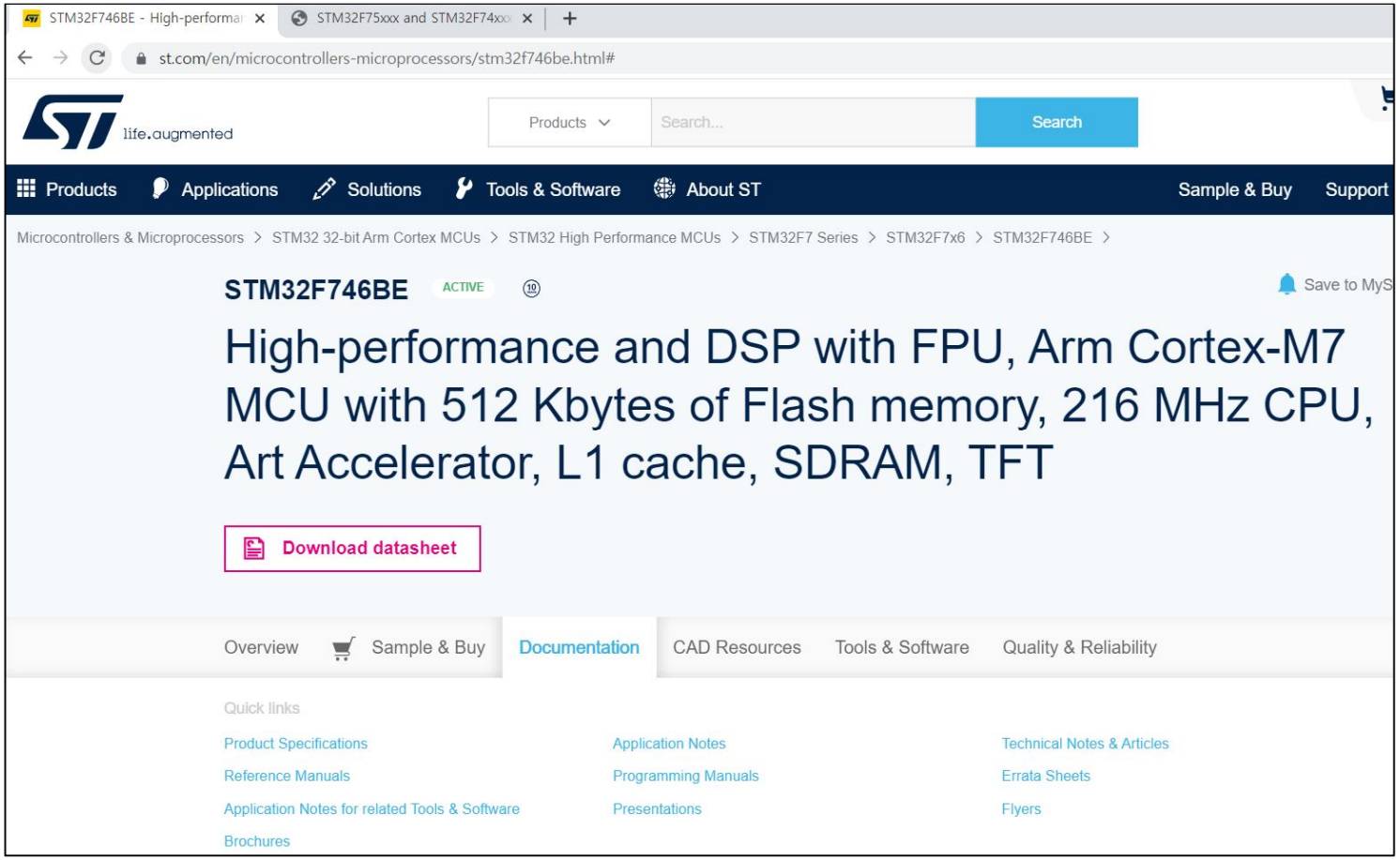
Gyroscope, Accelerometer, Magnetometer 3axis
: (Discontinued) MPU9250
: (New) ICM-20648

Turtlebot3 burger : Standard Specification

2. Specification of Parts

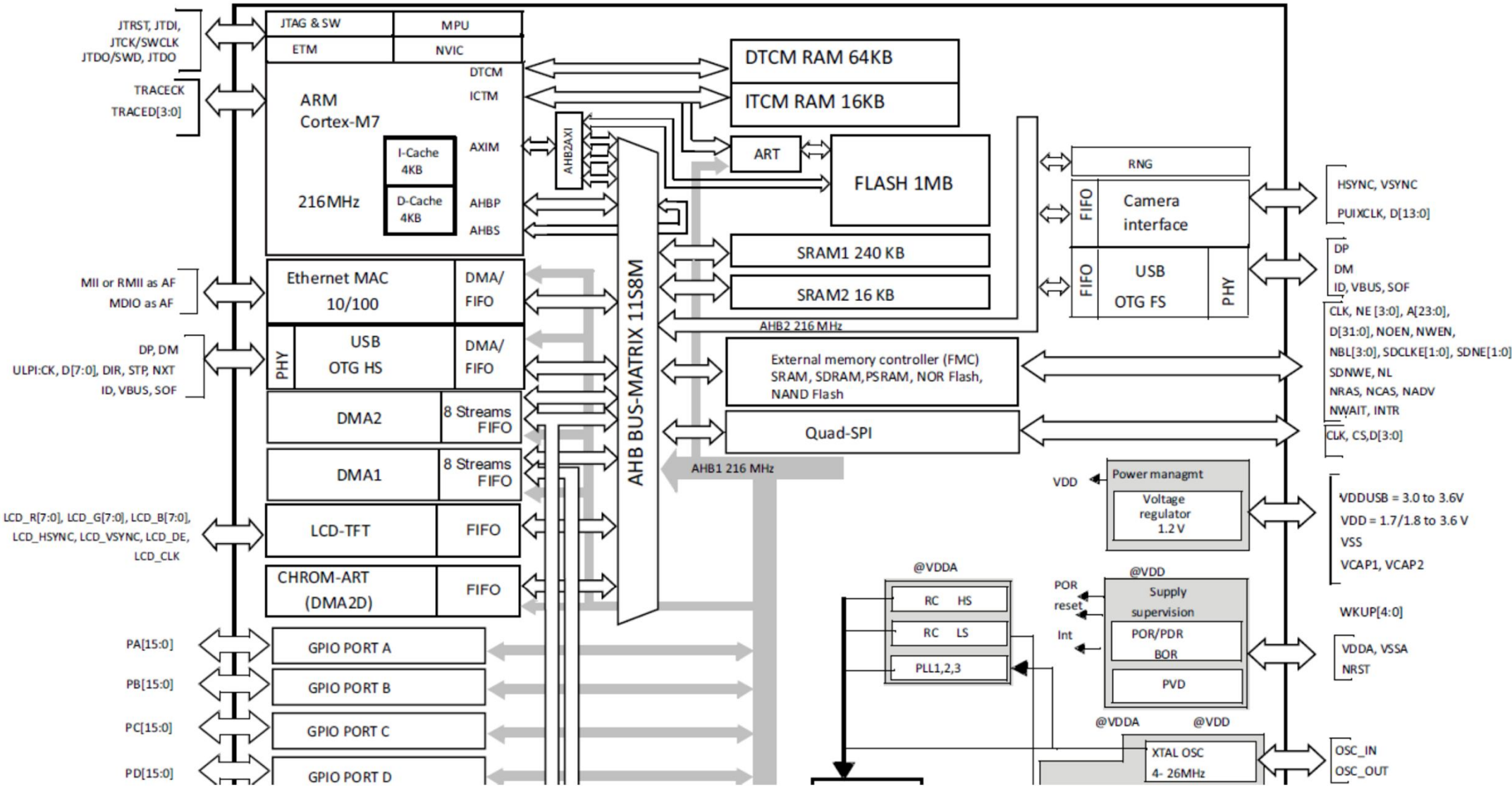
(2) OpenCR(Control module for ROS) : STM32F746 => st.com

- ARM Cortex-M7 32bits RISC core / 216MHz frequency



STM32F746

- ARM Cortex-M7 32bits RISC core / 216MHz frequency (pdf 16page)

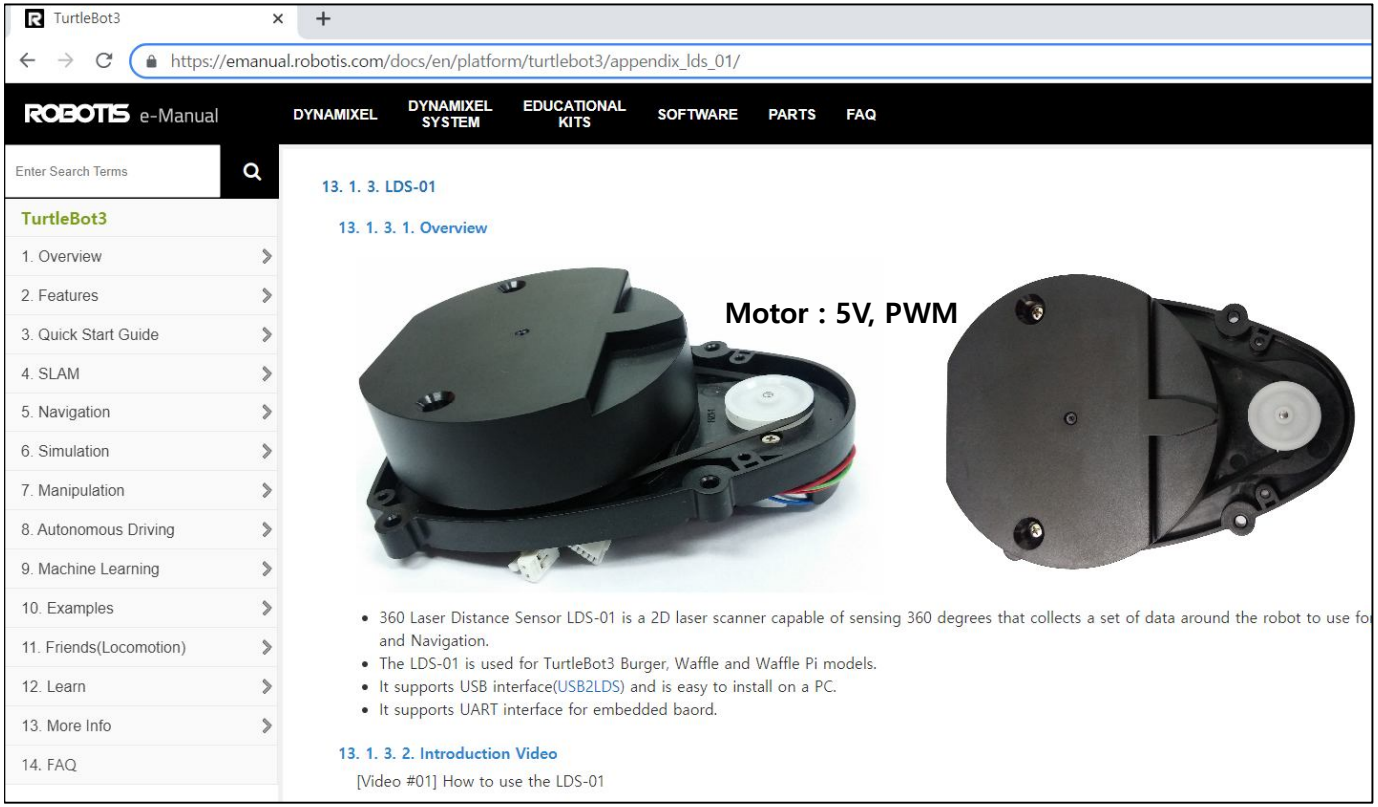


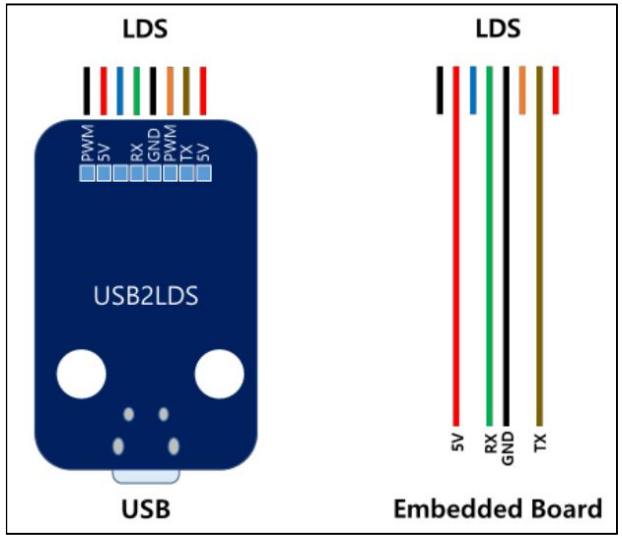
Turtlebot3 burger : Standard Specification

2. Specification of Parts

(3) 360° LiDAR : HLS-LFCD LDS(Laser Distance Sensor)

- Detection distance 120mm~3500mm
- Angular Resolution 1 degree / USB2LDS (115200 baudrate)
- https://emanual.robotis.com/docs/en/platform/turtlebot3/appendix_lds_01/





Turtlebot3 burger : Standard Specification

2. Specification of Parts

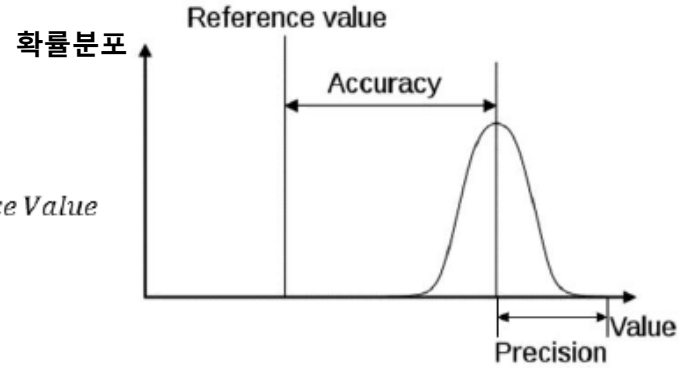
(3) 360° LiDAR : HLS-LFCD LDS(Laser Distance Sensor)

Items	Specifications
Operating supply voltage	5V DC ±5%
Light source	Semiconductor Laser Diode(λ=785nm)
LASER safety	IEC60825-1 Class 1
Current consumption	400mA or less (Rush current 1A)
Detection distance	120mm ~ 3,500mm
Interface	3.3V USART (230,400 bps) 42bytes per 6 degrees, Full Duplex option
Ambient Light Resistance	10,000 lux or less
Sampling Rate	1.8kHz
Dimensions	69.5(W) X 95.5(D) X 39.5(H)mm

Items	Specifications
Distance Range	120 ~ 3,500mm
Distance Accuracy (120mm ~ 499mm)	±15mm
Distance Accuracy(500mm ~ 3,500mm)	±5.0%
Distance Precision(120mm ~ 499mm)	±10mm
Distance Precision(500mm ~ 3,500mm)	±3.5%
Scan Rate	300±10 rpm
Angular Range	360°
Angular Resolution	1°

$$Accuracy = Distance\ Average - Reference\ Value$$

$$Precision = \frac{Max. - Min.}{2}$$



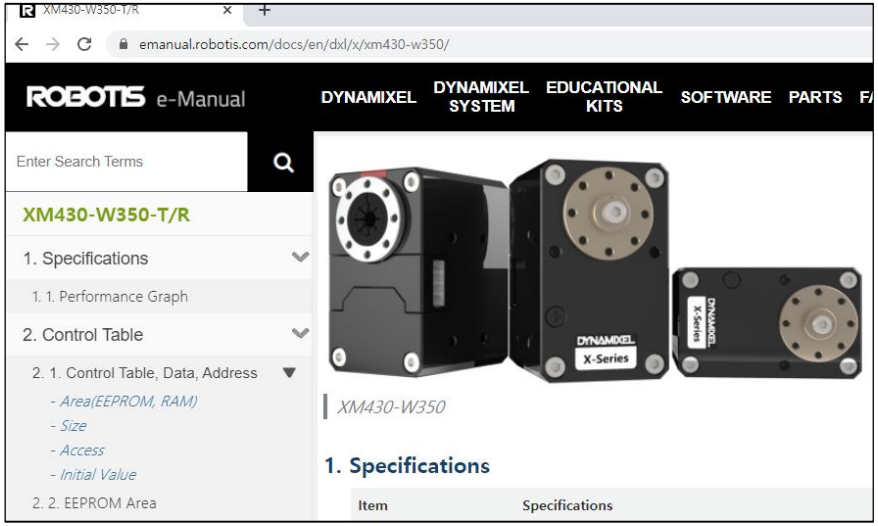
Turtlebot3 burger : Standard Specification

2. Specification of Parts

(4) DYNAMIXEL : XL430-W350-T

- https://emanual.robotis.com/docs/en/dxl/x/xm430-w350/
- Input Voltage 10.0 ~ 14.8 [V] (Recommended : 12.0 [V])

Item	Specifications
MCU	ARM CORTEX-M3 (72 [MHz], 32Bit)
Position Sensor	Contactless absolute encoder (12Bit, 360 [°]) Maker : ams(www.ams.com), Part No : AS5045
Motor	Coreless
Baud Rate	9,600 [bps] ~ 4.5 [Mbps]
Control Algorithm	PID control
Resolution	4096 [pulse/rev]
Backlash	15 [arcmin] (0.25 [°])
Operating Modes	Current Control Mode Velcoity Control Mode Position Control Mode (0 ~ 360 [°]) Extended Position Control Mode (Multi-turn) Current-based Position Control Mode PWM Control Mode (Voltage Control Mode)

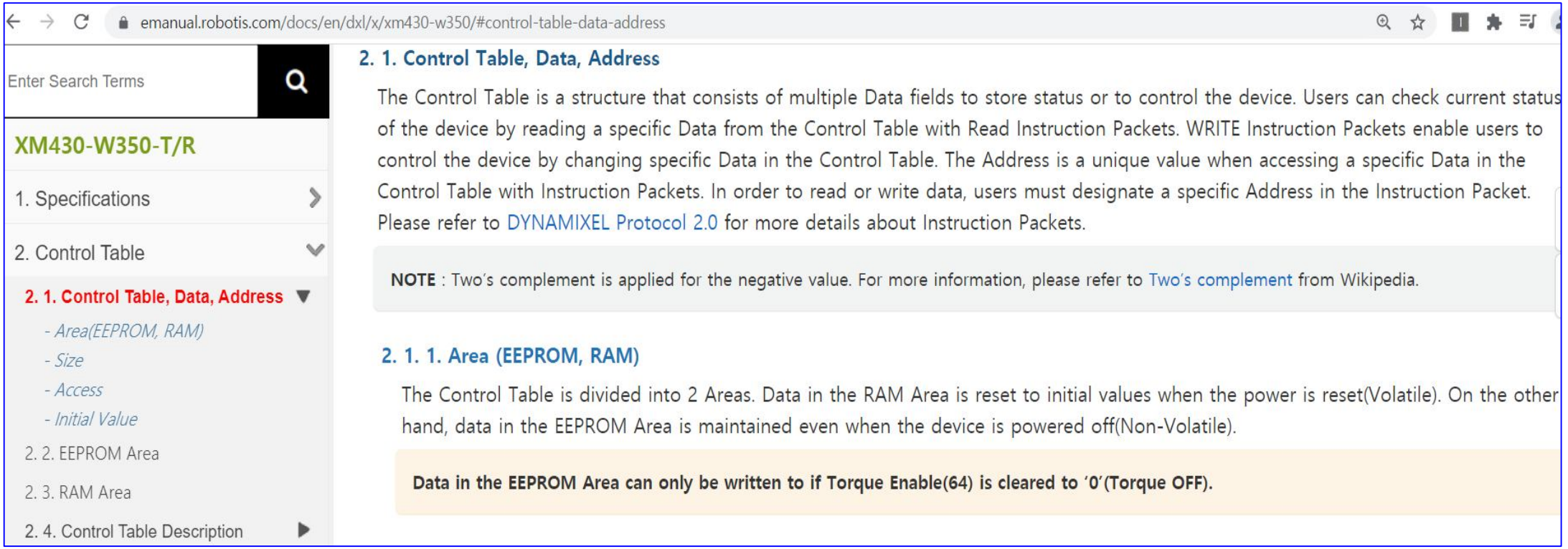


Weight	82 [g]
Dimensions (W x H x D)	28.5 x 46.5 x 34 [mm]
Gear Ratio	353.5 : 1
Stall Torque (순간최대토크)	3.8 [N.m] (at 11.1 [V], 2.1 [A]) **4.1 [N.m] (at 12.0 [V], 2.3 [A])** 4.8 [N.m] (at 14.8 [V], 2.7 [A])
No Load Speed	43 [rev/min] (at 11.1 [V]) **46 [rev/min] (at 12.0 [V])** 57 [rev/min] (at 14.8 [V])

Turtlebot3 burger : Standard Specification

2. Specification of Parts

(4) DYNAMIXEL : XL430-W350-T



Turtlebot3 burger : Standard Specification

2. Specification of Parts

(4) DYNAMIXEL : DYNAMIXEL Protocol 2.0

DYNAMIXEL Protocol 2.0

2. Instruction Packet

Instruction Packet is the command packet sent to the Device.

Header 1	Header 2	Header 3	Reserved	Packet ID	Length 1	Length 2	Instruction	Param	Param	Param	CRC 1	CRC 2
0xFF	0xFF	0xFD	0x00	ID	Len_L	Len_H	Instruction	Param 1	...	Param N	CRC_L	CRC_H

2. 1. Header

The field that indicates the start of the Packet

2. 2. Reserved

Uses 0X00 (Note that Reserved does not use 0XFD). The Reserved functions the same as Header.

See the next image of a table of the Packet Details of the DYNAMIXEL Wizard 2.0, which shows that the Reserved (0x00) are included in the Header field.

Header				ID	Length		Inst	Param			CRC	
FF	FF	FD	00	01	06	00	03	40	00	01	DB	66

A table of Packet Details of DYNAMIXEL Wizard 2.0

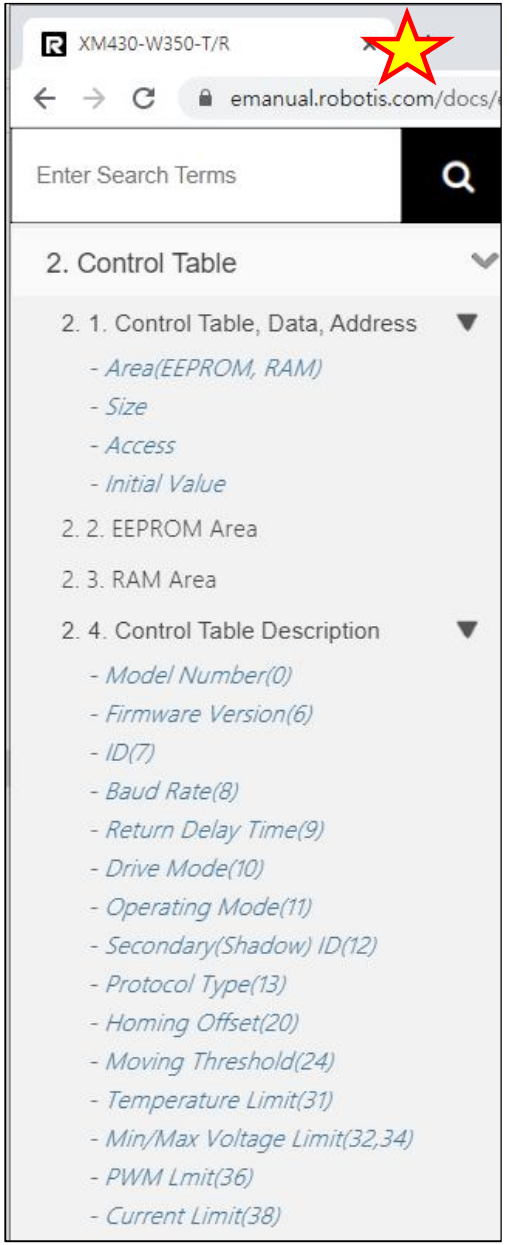
Turtlebot3 burger : Standard Specification

2. Specification of Parts

(4) DYNAMIXEL : XL430-W350-T

2. 2. Control Table of EEPROM Area

Address	Size(Byte)	Data Name	Access	Initial Value	Range	Unit
0	2	Model Number	R	1,020	-	-
2	4	Model Information	R	-	-	-
6	1	Firmware Version	R	-	-	-
7	1	ID	RW	1	0 ~ 252	-
8	1	Baud Rate	RW	1	0 ~ 7	-
9	1	Return Delay Time	RW	250	0 ~ 254	2 [µsec]
10	1	Drive Mode	RW	0	0 ~ 5	-
11	1	Operating Mode	RW	3	0 ~ 16	-
12	1	Secondary(Shadow) ID	RW	255	0 ~ 252	-
13	1	Protocol Type	RW	2	1 ~ 2	-
20	4	Homing Offset	RW	0	-1,044,479 ~ 1,044,479	1 [pulse]
24	4	Moving Threshold	RW	10	0 ~ 1,023	0.229 [rev/min]
31	1	Temperature Limit	RW	80	0 ~ 100	1 [°C]
32	2	Max Voltage Limit	RW	160	95 ~ 160	0.1 [V]
34	2	Min Voltage Limit	RW	95	95 ~ 160	0.1 [V]
36	2	PWM Limit	RW	885	0 ~ 885	0.113 [%]

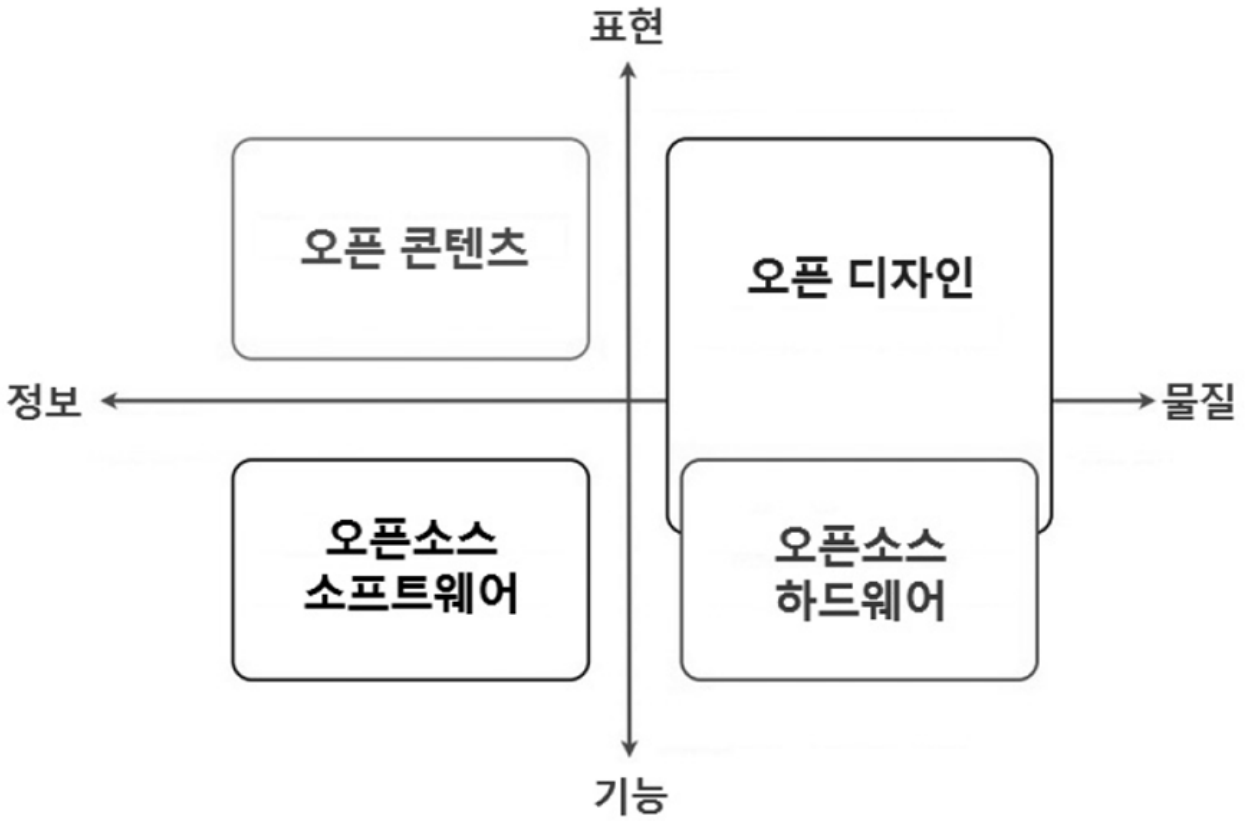


1. Open Source Hardware(OSHW) 플랫폼

1-1. Open Source Hardware, OSHW 기반의 실험

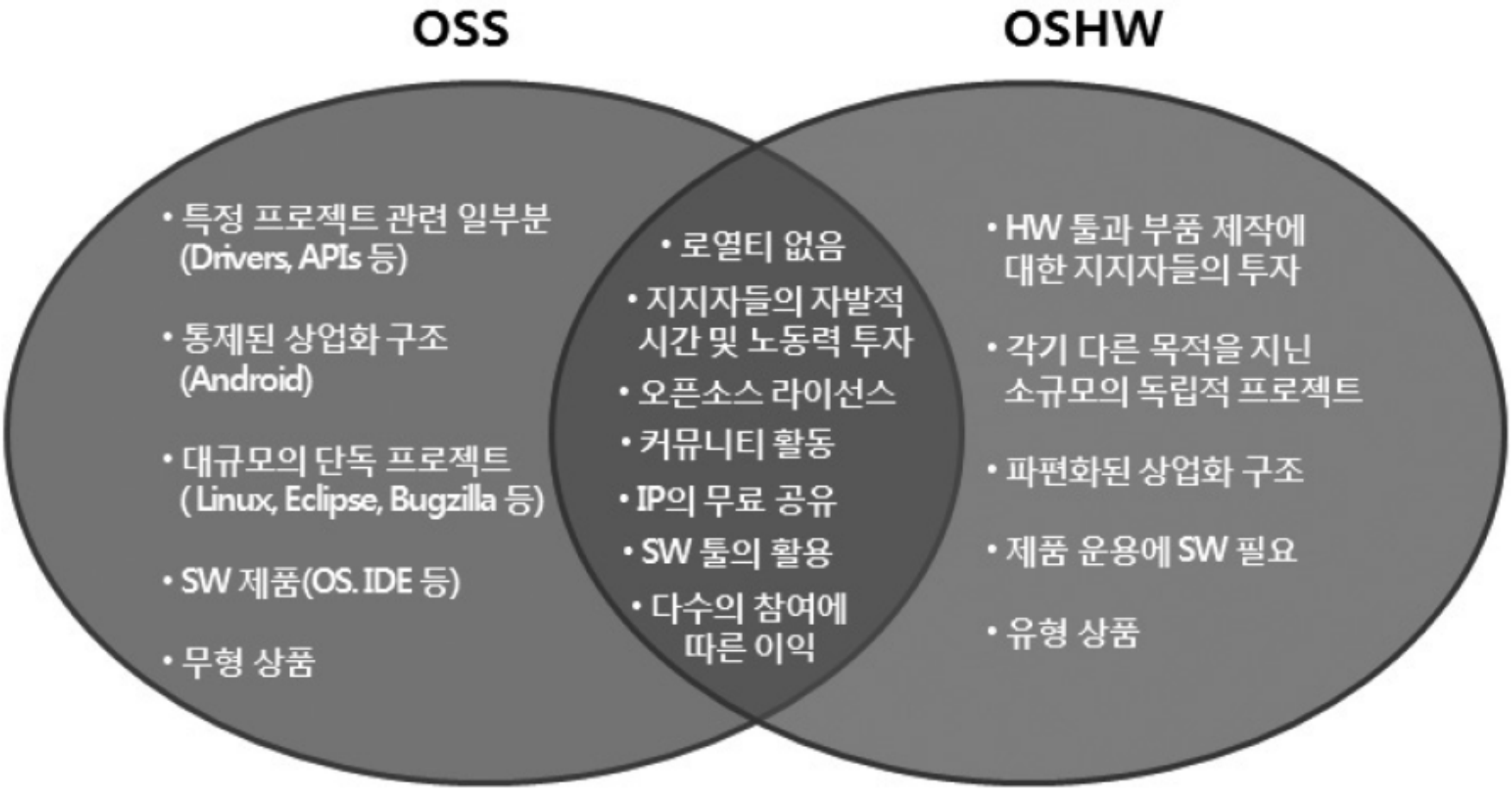
- **Open Source Hardware(OSHW) & Open Source Software(OSS)**

  : 기존 산업구조와 비교해 저렴한 하드웨어 비용(?)과 수많은 소스•예제를 공유하고 아이디어를 제안하는 글로벌 커뮤니티,
  그리고 일반인도 쉽게 배우고 직접 만들어 볼 수 있는 '낮은 진입장벽'



1. Open Source Hardware(OSHW) 플랫폼

- Open Source Hardware(OSHW) & Open Source Software(OSS)



1. Open Source Hardware(OSHW) 플랫폼

1-2. Open Source Hardware(OSHW) 플랫폼 : Arduino

: 아두이노(Arduino)

	MCU	Board
AVR	ATmega168	Pro(168), Mini(168), LilyPad (168V)
	ATmega328	UNO, Fio, Nano, Pro(328), Mini(328, Rev5, 5V), Pro Mini, LilyPad (328V)
	ATmega2560	Mega 2560, Mega ADK
	ATmega32U4	Yún, Leonardo, Esplora, Micro
	ATtiny85	GEMMA
ARM	Cortex-M0+	Zero, Zero PRO, M0, M0 PRO
	Cortex-M3	Due

1. Open Source Hardware(OSHW) 플랫폼

1-3. Open Source Hardware(OSHW) 플랫폼 : Beagle Board

: 비글보드(Beagle Board)

➔ ARM Cortex-A 시리즈 코어를 갖춘 저전력 Texas Instrument(TI) 프로세서를 사용하며, 저비용이고, 팬이 없는 소형 단일 보드 컴퓨터.

➔ 오픈소스 하드웨어로써 SoC인 OMAP(Open Multimedia Applications Platform) 3530을 기반 리눅스를 비롯해 안드로이드, 우분투(Ubuntu) 등 다양한 OS를 지원.

<비글보드>

<비글보드-xM>

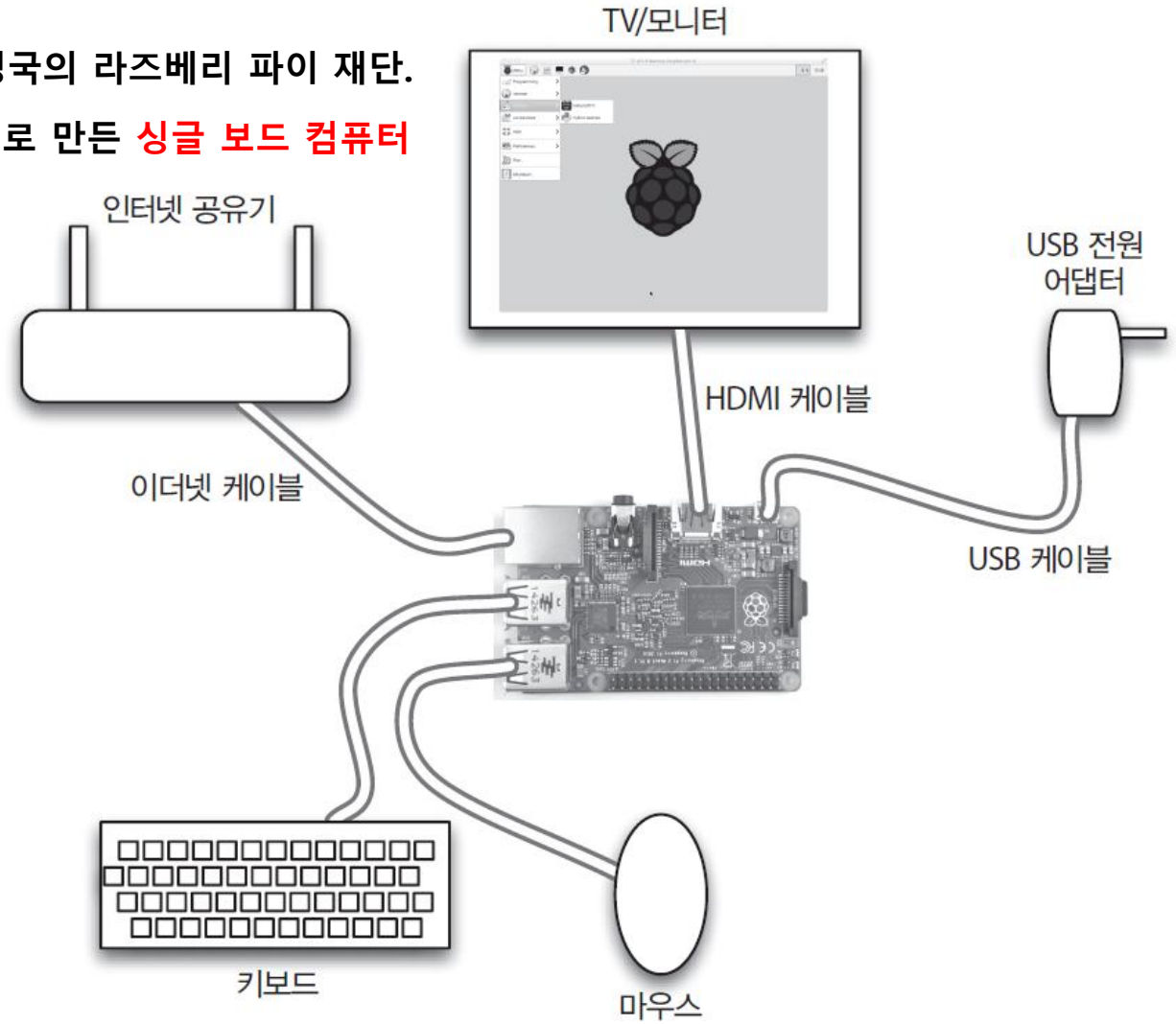
<비글본>

<비글본 블랙>

1. Open Source Hardware(OSHW) 플랫폼

1-4. Open Source Hardware(OSHW) : Raspberry Pi

: 라즈베리 파이(Raspberry Pi)는 2012년 2월, 영국의 라즈베리 파이 재단.
학교에서 컴퓨터 및 과학 교육의 증진을 목적으로 만든 **싱글 보드 컴퓨터**



- **1.2GHz 64-bit quad-core ARMv8 CPU**
  - ➔ **Broadcom BCM2837 : raspberry pi3**
- **802.11n Wireless LAN**
- **Bluetooth 4.1**
- **Bluetooth Low Energy (BLE)**

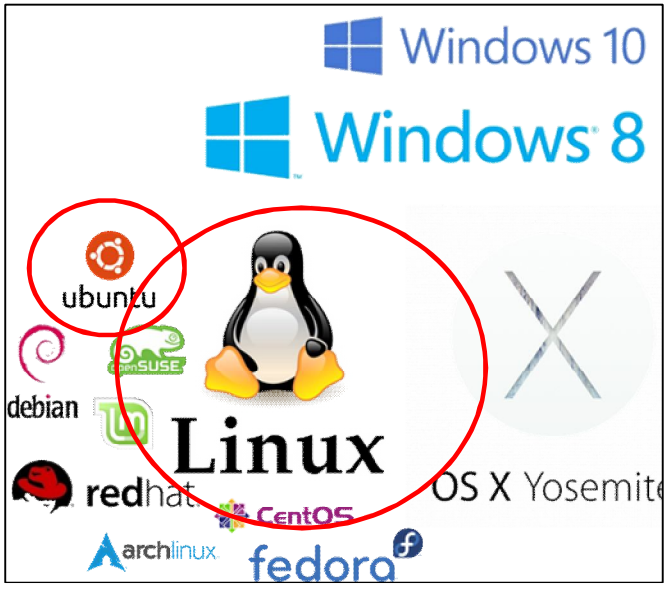
2. Robot Software Platform

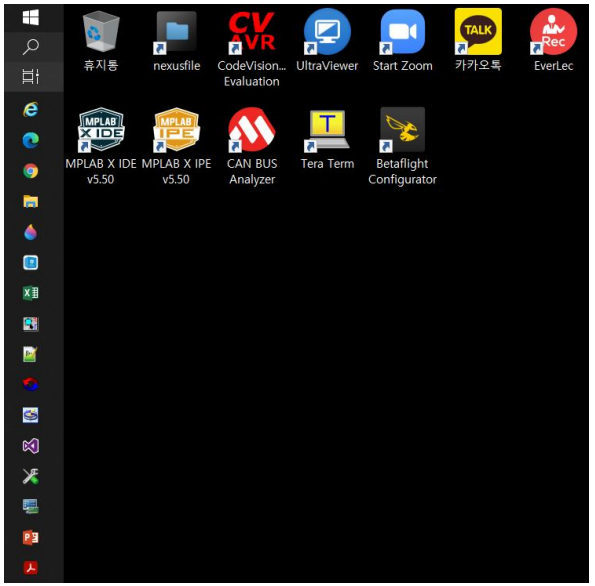
2-1. Software Platform

- **Platform 구성요소**

  : Conventional, 특정 회사에서 개발된 하드웨어의 전용 펌웨어(Firmware)로
  특정 하드웨어 장치의 구동을 위해 소프트웨어를 탑재하였고
  제조사에서 제공한 서비스만을 이용할

  : **Hardware + Operating System(운영체제) + Application(응용 프로그램) + User**





2. Robot Software Platform

2-2. Robot Software Platform

- Robot Platform 구성요소

  : 소프트웨어 플랫폼과 하드웨어 플랫폼

- 로봇 소프트웨어 플랫폼

  : 로봇 응용프로그램을 개발할 때 필요한 하드웨어 추상화, 하위 디바이스 제어, 로보틱스에서 많이 사용되는 센싱, 인식, 동시적 자기 위치 추정과 지도 작성(SLAM), 내비게이션(Navigation), 매니퓰레이션(Manipulation) 등의 기능 구현과 프로그램 패키지 관리, 개발환경에 필요한 라이브러리와 다양한 개발, 디버깅 도구를 의미

  : 로봇 운영체제

  - 미국 ROS(Robot Operating System), 일본 오픈 로보틱스 테크놀로지 미들웨어(OpenRTM), 유럽 실시간 제어 중심의 OROCOS, 한국의 OPRoS (Open Platform for Robot Services)

  : 로봇 기능 공유 그룹

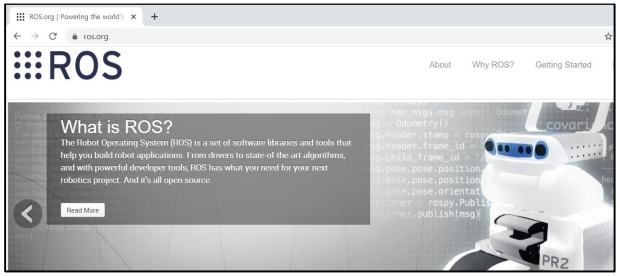
  - 얼굴인식과 물체인식 기능의 로봇베이스(Robotbase)
  - 모바일 로봇 기능의 유진로봇

  : ROS 산업 컨소시엄 구성의 ROS-I(ROS Industrial Consortium)

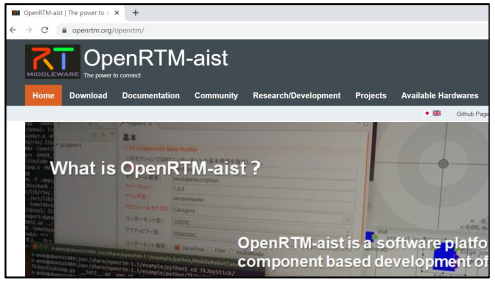
2. Robot Software Platform

: 로봇 운영체제 ★

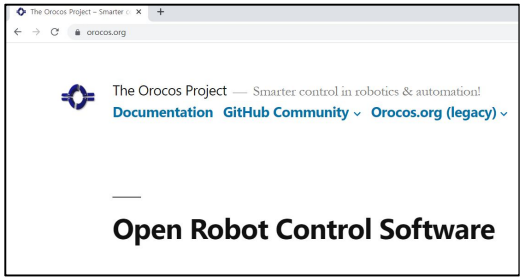
http://www.ros.org



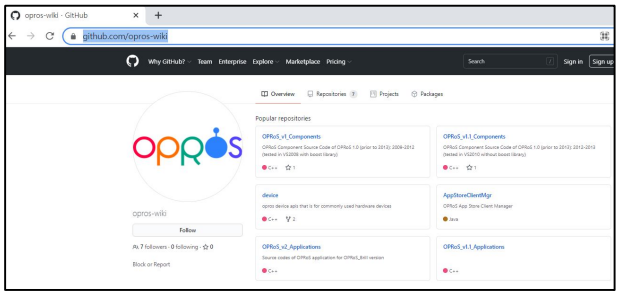
http://openrtm.org



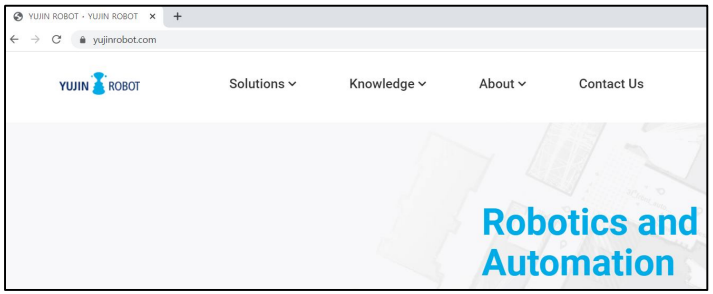
http://www.orocos.org



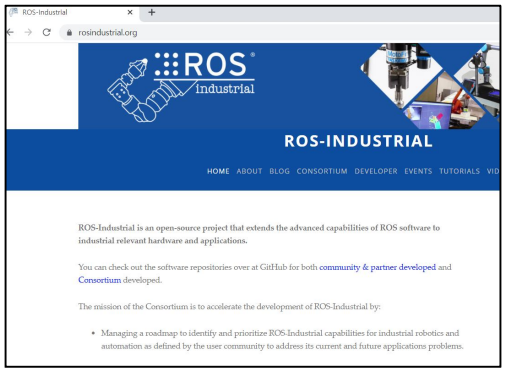
https://github.com/opros-wiki



http://www.yujinrobot.com



http://rosindustrial.org/



2. Robot Software Platform

2-3. Why ROS ?

- 실증 예제

  : 미국 NASA 국제 우주 정거장 운영 로봇

  Robonaut2 (https://robonaut.jsc.nasa.gov/R2/)에 유럽 실시간 제어 중심의 OROCOS 혼용

https://www.ros.org/

- 프로그램의 재사용 및 Permissive Licensing

  : ROS 혼용

- 통신 기반의 프로그램

  : 센싱, 인식, 동작의 프로그램을 components화, 즉 node화하여

  각 node별 최소 실행 단위 통신 운영

- 개발도구 지원

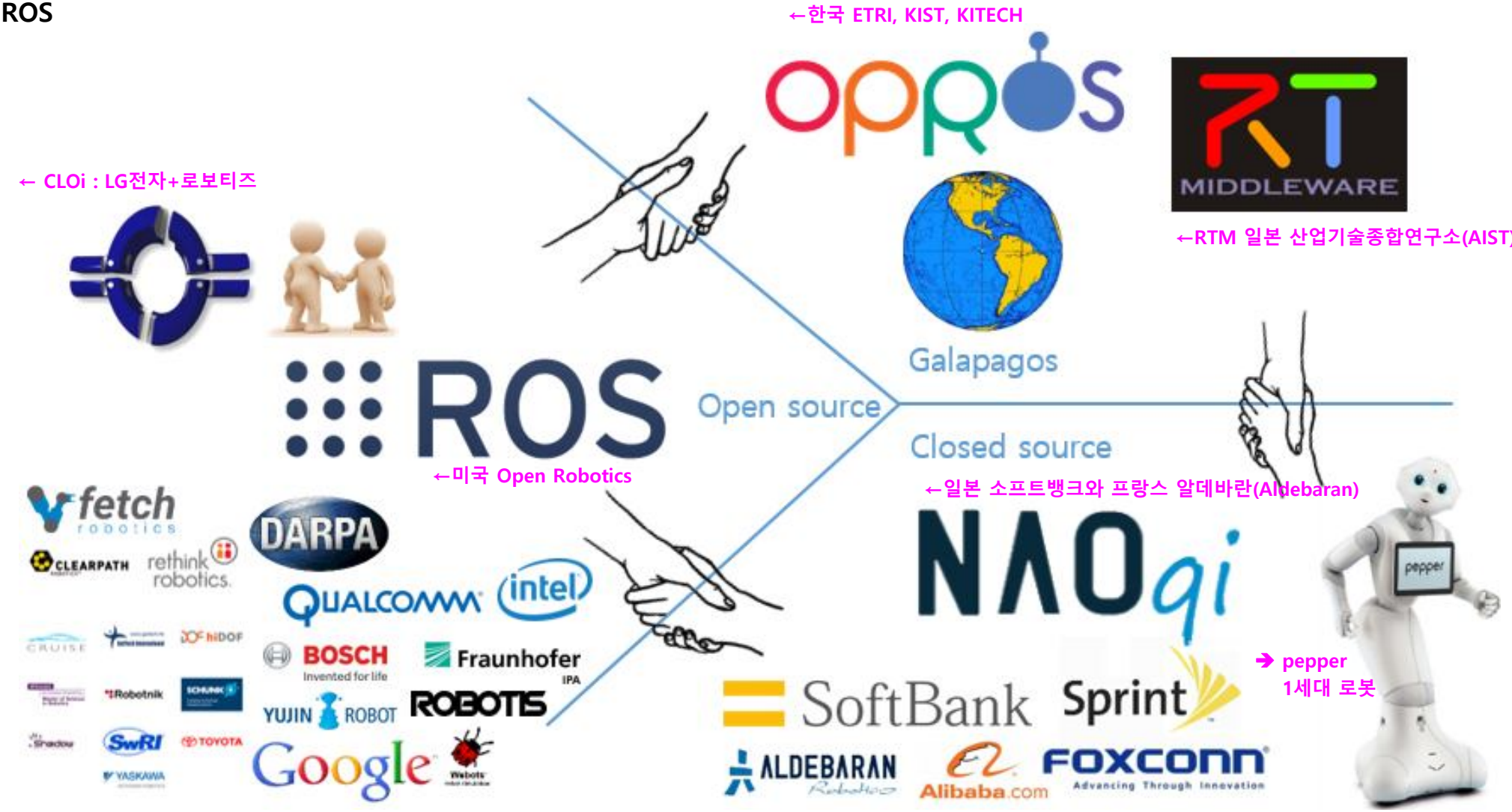
  : GUI도구 rqt, 3차원 시각화 도구 Rviz 제공

- 활성화 된 ROS community

  : 22,000 wiki pages

2. Robot Software Platform

2-4. ROS



LG CLOi ServeBot

https://www.lge.co.kr/kr/business/product/it/lg-LDLIM10

: 서울대학교병원 설치

https://youtu.be/rPMhQzw7peE

: 2021 CES LG전자 CLOi

3. ROS Robot Operating System

3-1. 운영체제 Operating System

- UNIX 운영체제(Operating System)

  : 1970년대 초에 AT&T 벨 연구소에서 개발된 이후로 지속적으로 발전

  스마트폰, PC, 서버, 슈퍼컴퓨터에까지 사용. 소프트웨어 경쟁력의 핵심

  : Ken Thompson이 어셈블리어로 개발함, D. Ritchie가 C 언어로 다시 작성함

  ➔ C 언어는 Unix를 작성하기 위한 언어

- UNIX 기반의 운영체제(Operating System)

  : 안드로이드(Android) OS

  : iOS

  : 맥(Mac) OS X

  : 리눅스(Linux)

  : BSD 유닉스(Berkeley Standard Distribution Unix)

  : 시스템 System V

  : Sun 솔라리스(Solaris)

  : IBM AIX : HP HP-UX : Cray 유니코스(Unicos)

3. ROS Robot Operating System

3-1. 운영체제 Operating System

- UNIX 운영체제 특징

  : 단순성(Simplicity)

  - 운영체제 자체를 가능한 최소한의 필요 기능만 제공
  - 자원에 대한 일관된 관점 제공

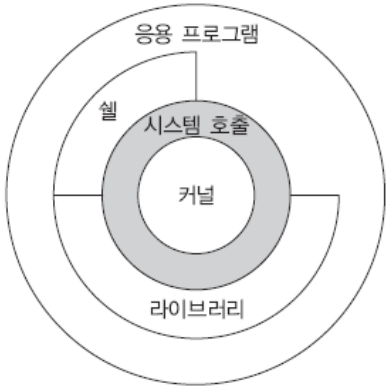
  : 이식(Portability)

  - 이식을 위해 C 언어로 작성
  - 다양한 플랫폼에 이식 가능
  - 스마트폰, PC, 서버, 슈퍼컴퓨터 등

  : 개방성

  - 소스 코드 공개와 같은 개방성

- UNIX 운영체제 구조



  : 운영체제

  - 컴퓨터 하드웨어 자원 운영 관리,
    프로그램을 실행 환경을 제공

  : 커널(kernel)

  - 운영체제의 핵심으로 하드웨어 운영 및 관리
  - 하드웨어와 운영체제의 다른 부분 사이의 중재자 역할

  : 시스템 호출(system call)

  - 커널이 제공하는 서비스에 대한 프로그래밍 인터페이스

  : 쉘(shell)

  - 사용자와 운영체제 사이의 인터페이스
  - 사용자로부터 명령어 입력 받아, 해석 수행해주는 명령어 해석기

3. ROS Robot Operating System

3-1. 운영체제 Operating System

- UNIX History

: System V

- 벨 연구소에서 개발된 버전이 발전하여 시스템 V가 됨
- 유닉스 버전 중의 최초의 대표적인 성공 사례
- 여러 유틸리티가 공개되면서 일반 사용자들에 확산
- 다양한 상업용 버전으로 발전

➔ IBM의 AIX, Sun의 Solaris, HP의 UP-UX

: BSD(Berkeley Standard Distribution)

- 공개 소스코드를 기반으로 버클리대학교에서 개선
- 주요 기능 개선

➔ 메모리 관리 기능 향상

➔ 네트워킹 기능 추가 - TCP/IP 네트워킹, 소켓(Socket) 등

- 상업용 운영체제의 기초

➔ 썬 OS(Sun OS), 맥 OS(Mac OS)

: Linux

- PC를 위한 효율적인 유닉스 시스템
- 1991년 헬싱키 대학의 Linus B. Torvalds에 의해 개발됨
- 소스코드 공개

➔ 인터넷 상에서 개발자들에 의해 기능 추가 및 확장됨

➔ 공용 도메인 상의 무료 OS

- 다양한 Platform에 맞게 porting 가능

➔ PC, 워크스테이션, 서버 등

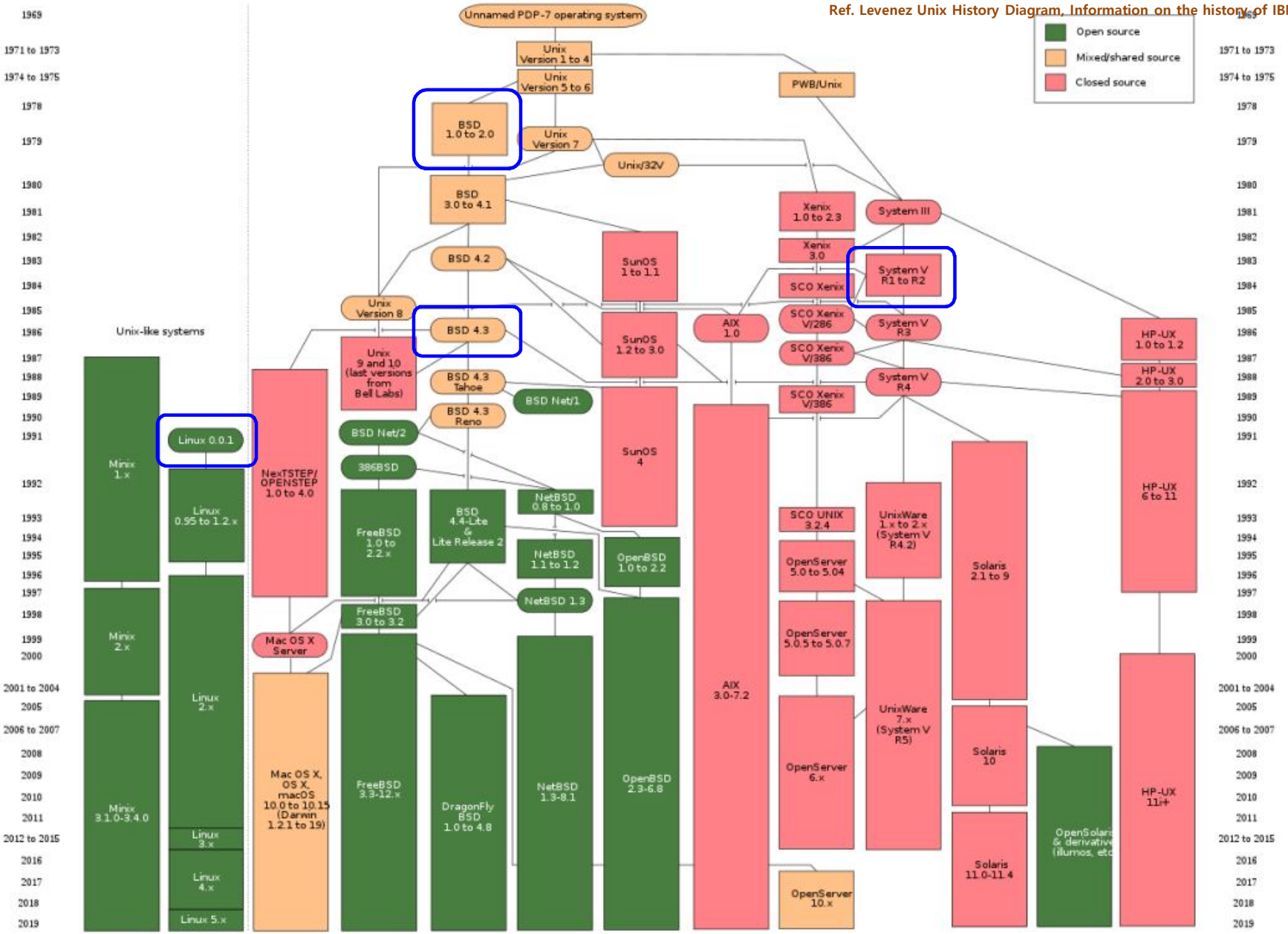
- GNU(General Public License) 소프트웨어와 함께 배포

➔ "GNU is Not UNIX"

➔ 레드햇(RedHat), 우분투(Ubuntu), 데비안(Debian), 페도라(Fedora), CentOS

- **UNIX History Tree**

Ref. Levenez Unix History Diagram, Information on the history of IBM's AIX on ibm.com



3. ROS Robot Operating System

3-1. 운영체제 Operating System

- **Linux**

: GNU('그누'로 발음)는 GNU 프로젝트를 통하여 개발한 유닉스 계열 컴퓨터 운영 체제

궁극적으로는 "완전한 유닉스 호환 소프트웨어 시스템"이 되는 것이 목표

: GNU는 "GNU는 유닉스가 아니다."란 의미를 갖는 문장 "GNU's Not UNIX"의 약자

: GNU 일반 공중 사용 허가서 (GNU General Public License, GPL)

: GNU 약소 일반 공중 사용 허가서 (The GNU Lesser General Public License, LGPL)

: GNU 자유 문서 사용 허가서 (GNU Free Documentation License, GFDL)

: Linux 장점

- 풍부하고 다양한 하드웨어를 효과적으로 지원

➔ 대부분의 하드웨어를 지원하는 추세. PC, 워크스테이션, 서버 등

- 놀라운 성능 및 안정성, 인터넷에 맞는 강력한 네트워크 구축

- 다양한 응용 프로그램 개발됨

- 무료 배포판

➔ 우분투(Ubuntu) / 페도라(Fedora) / CentOS

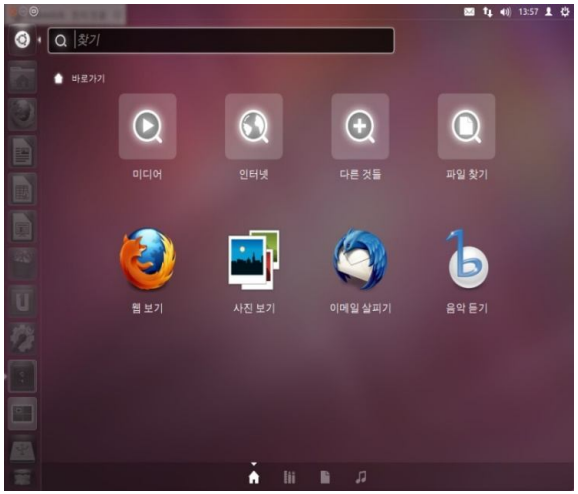
3. ROS Robot Operating System

3-1. 운영체제 Operating System

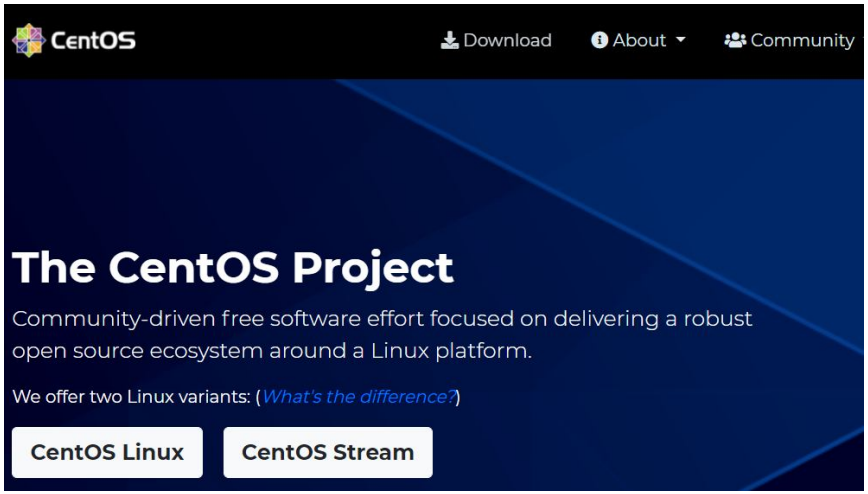
- **Linux 무료 배포판**

  : 우분투(Ubuntu) / 페도라(Fedora) / CentOS

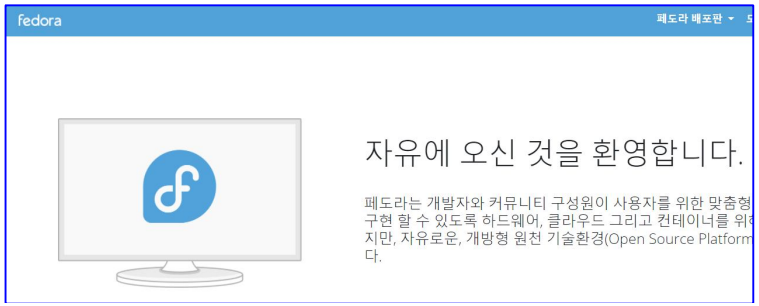
http://www.ubuntu.com



http://www.centos.org



https://getfedora.org/ko



3. ROS Robot Operating System

3-2. About ROS

The **Robot Operating System (ROS 로봇 운영체제)** is **a flexible framework for writing robot software**. It is a **collection of tools, libraries**, and conventions that aim **to simplify the task of creating complex and robust robot behavior** across a wide variety of robotic platforms.

Why? Because creating truly robust, general-purpose robot software is *hard*. From the robot's perspective, problems that seem trivial to humans often vary wildly between instances of tasks and environments. Dealing with these variations is so hard that no single individual, laboratory, or institution can hope to do it on their own.

As a result, ROS was built from the ground up **to encourage *collaborative* robotics software development**. For example, **one laboratory might have experts in mapping indoor environments, and could contribute a world-class system for producing maps. Another group might have experts at using maps to navigate,** and **yet another group might have discovered a computer vision approach that works well for recognizing small objects** in clutter.

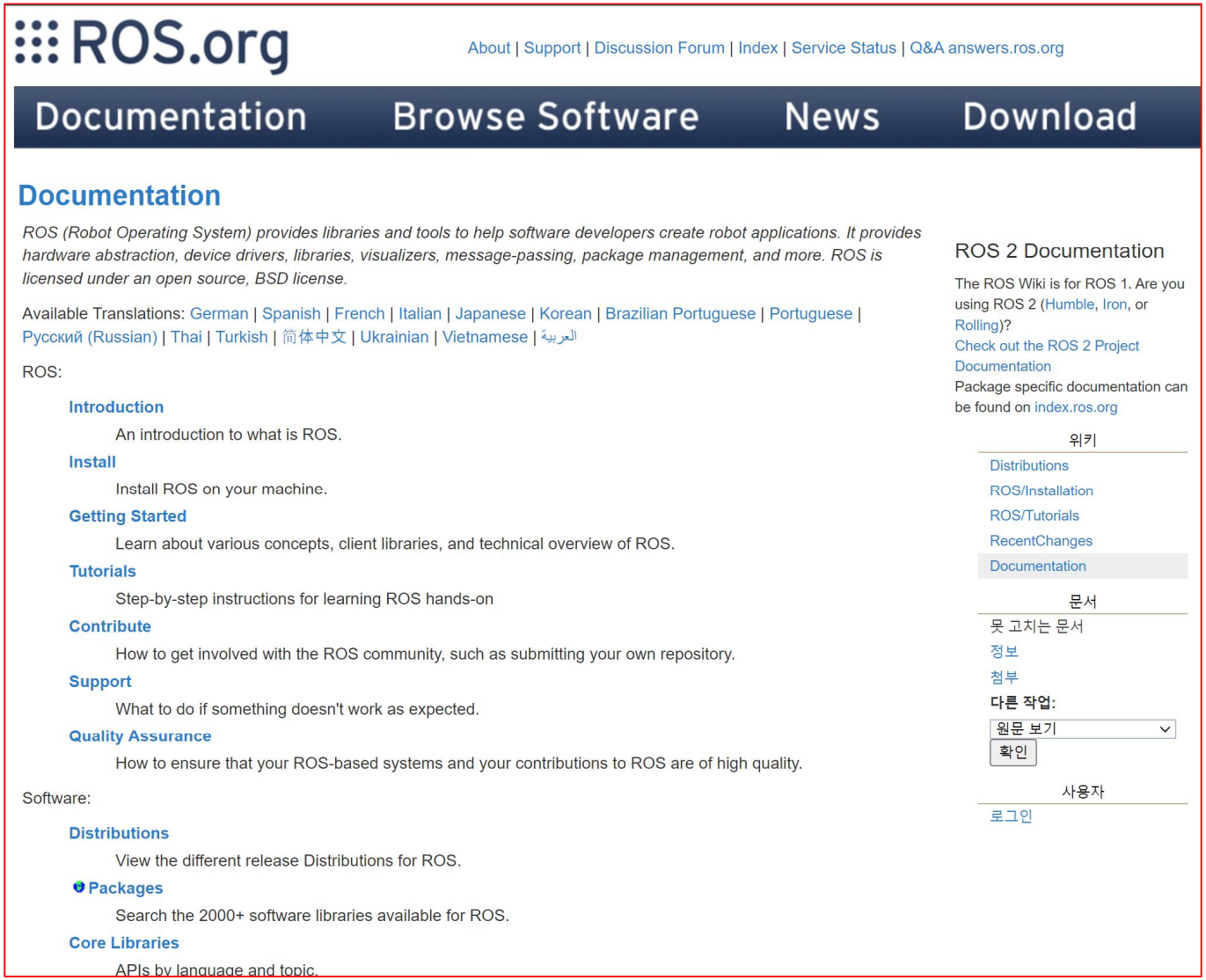
ROS was designed specifically for groups like these to collaborate and build upon each other's work, as is described throughout this site.

Ref. ros.org/about-ros/

3. ROS Robot Operating System

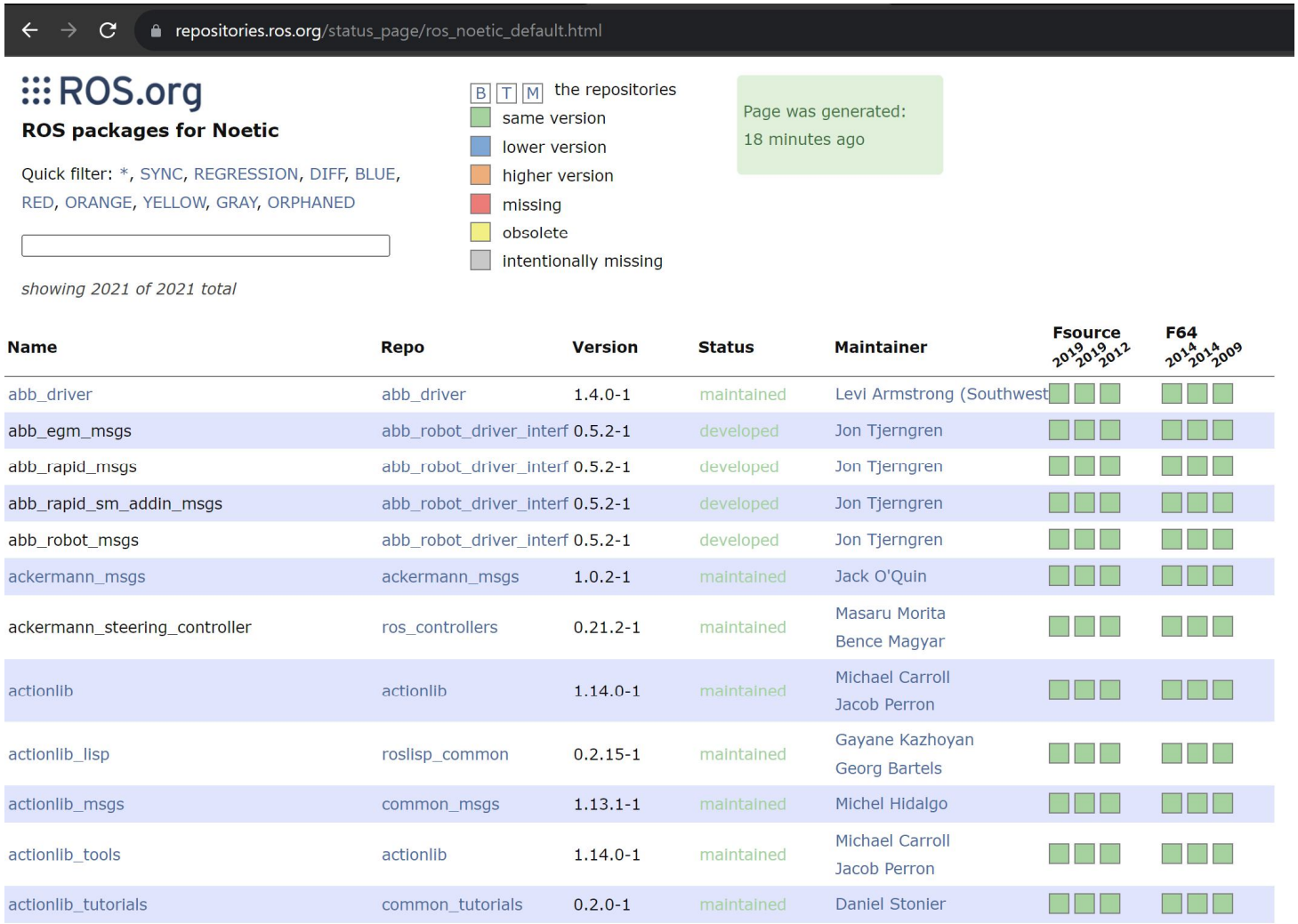
3-2. About ROS

http://wiki.ros.org/



3. ROS Robot Operating System

ROS noetic package http://repositories.ros.org/status_page/ros_noetic_default.html



repositories.ros.org/status_page/ros_noetic_default.html

ROS.org

ROS packages for Noetic

Quick filter: *, SYNC, REGRESSION, DIFF, BLUE, RED, ORANGE, YELLOW, GRAY, ORPHANED

showing 2021 of 2021 total

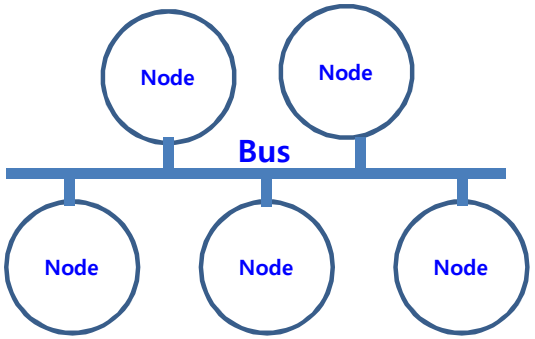
B T M the repositories
same version
lower version
higher version
missing
obsolete
intentionally missing

Page was generated:
18 minutes ago

Name	Repo	Version	Status	Maintainer	Fsource 2019 2019 2012	F64 2014 2014 2009
abb_driver	abb_driver	1.4.0-1	maintained	Levi Armstrong (Southwest		
abb_egm_msgs	abb_robot_driver_interf	0.5.2-1	developed	Jon Tjerngren		
abb_rapid_msgs	abb_robot_driver_interf	0.5.2-1	developed	Jon Tjerngren		
abb_rapid_sm_addin_msgs	abb_robot_driver_interf	0.5.2-1	developed	Jon Tjerngren		
abb_robot_msgs	abb_robot_driver_interf	0.5.2-1	developed	Jon Tjerngren		
ackermann_msgs	ackermann_msgs	1.0.2-1	maintained	Jack O'Quin		
ackermann_steering_controller	ros_controllers	0.21.2-1	maintained	Masaru Morita Bence Magyar		
actionlib	actionlib	1.14.0-1	maintained	Michael Carroll Jacob Perron		
actionlib_lisp	roslisp_common	0.2.15-1	maintained	Gayane Kazhoyan Georg Bartels		
actionlib_msgs	common_msgs	1.13.1-1	maintained	Michel Hidalgo		
actionlib_tools	actionlib	1.14.0-1	maintained	Michael Carroll Jacob Perron		
actionlib_tutorials	common_tutorials	0.2.0-1	maintained	Daniel Stonier		

3. ROS Robot Operating System

3-2. About ROS



- Meta-Operating System

  : ROS는 Ubuntu, Fedora, CentOS 등 앞에서 언급했던 운영체제 개념과 상이함

  : 애플리케이션과 분산 컴퓨팅 자원 간의 가상 layer.

  분산 컴퓨팅 자원을 활용하여 스케줄링 및 로드, 감시, 에러 처리 등을 실행하는 시스템을 의미

  : Ubuntu 운영체제를 설치하고 그 위에 추가로 ROS 설치하여

  프로세스 관리 시스템, 파일 시스템, 유저 인터페이스, 프로그램 유틸리티(컴파일러, Thread) 등 사용

  : 다수의 이기종 하드웨어 간의 데이터 송수신, 스케줄링, 에러 처리 등 로봇 응용프로그램에 필요한 필수 기능 라이브러리 제공

3. ROS Robot Operating System

3-2. About ROS

- ROS 구성

: Client Libraries

- 다양한 프로그래밍 언어(C, Python)를 지원하기 위함

- libraries roscpp, rospy

: 하드웨어 인터페이스

- 하드웨어 제어를 위함

: 데이터 통신을 위한 커뮤니케이션

: Robotics Application Framework

- 다양한 로보틱스 응용프로그램 작성 위함

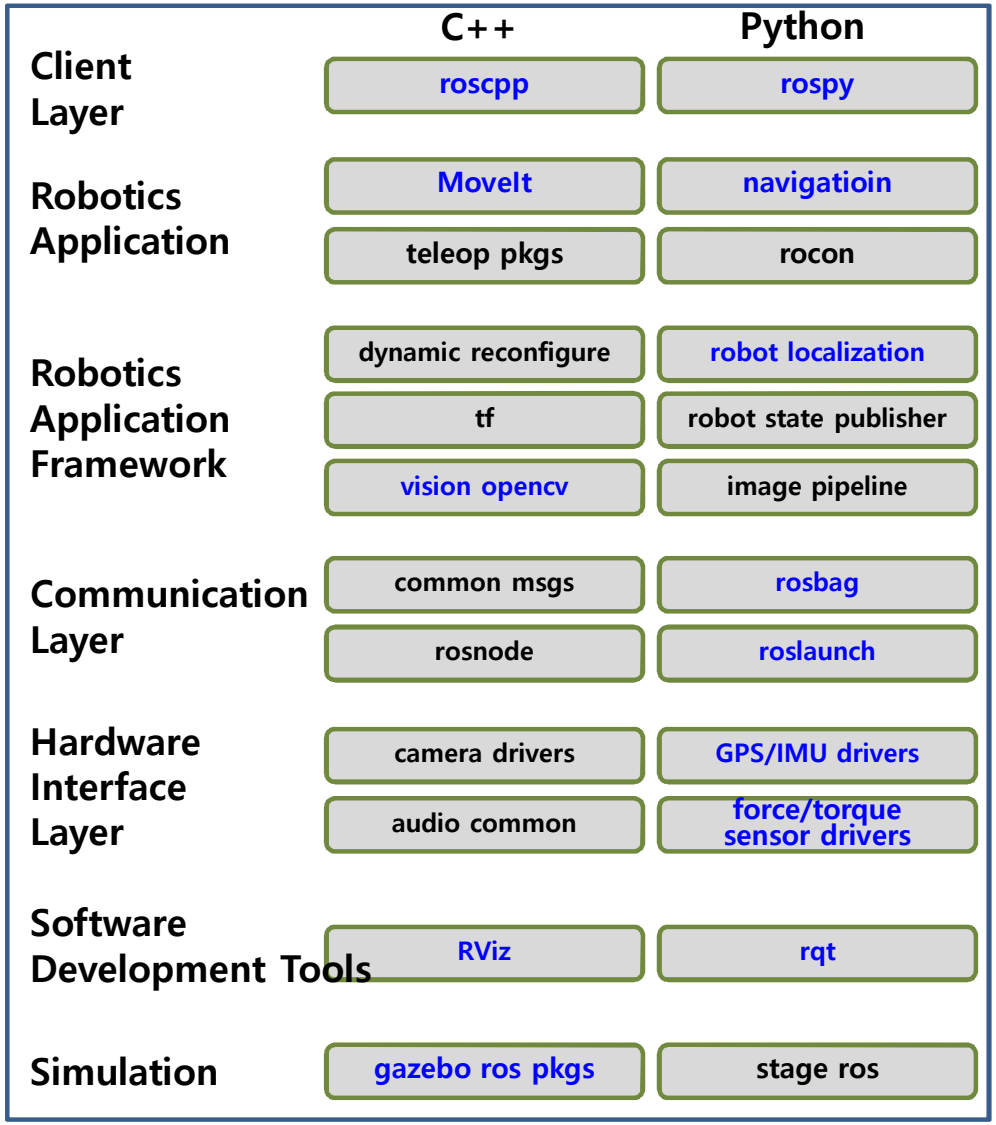
: Robotics Application

- Robotics Application Framework 기반의 서비스용 응용프로그램

: Simulation

- 가상의 공간에서 로봇을 제어해 볼 수 있는 시뮬레이션

: Software Development Tool



- **ROS 구성**

API	ROS	C++	Python
ROS	ROS	roscpp	rospy
Basic Datatypes	common_msgs	common_msgs	common_msgs
Manipulating message streams	topic_tools	message_filters	message_filters
Drivers	joystick_drivers, camera_drivers, laser_drivers, audio_common (aka sound_drivers), imu_drivers	joystick_drivers, camera_drivers, laser_drivers, audio_common, imu_drivers	
Driver Implementation	driver_common	driver_common	
Filtering data		filters	
3D processing	laser_pipeline, perception_pcl	laser_pipeline, perception_pcl	
Image processing		image_common, image_pipeline, vision_opencv	vision_opencv
Transforms/Coordinates		tf, tf_conversions, robot_state_publisher	tf, tf_conversions
Actions	actionlib	actionlib	actionlib
Executive/Task Manager	executive_smach		executive_smach
Navigation	navigation	via actionlib	via actionlib
Simulation (2D)	simulator_stage	simulator_stage	
Simulation (3D)	simulator_gazebo	simulator_gazebo	
Robot Model		robot_model	
Realtime Controllers	pr2_controller_manager	pr2_controller_interface, realtime_tools	
Motion planning (arms)	ompl, chomp_motion_planner, sbpl	actionlib through move_arm	actionlib through move_arm
Humanoid walk	walk_msgs	walk_interfaces	

Ref. http://wiki.ros.org/APIs

3. ROS Robot Operating System

- ROS 구성

Ref. https://roscon.ros.org/2023/

APP

- ❑ 5,000이상의 패키지
- ❑ 2,818개의 공식 패키지 제공 (Indigo /March, 2017)
- ❑ 13,441,711 .deb 패키지 다운로드 (July 2017)
- ❑ 18,839 Wiki페이지 (July 2017)

ROS

- ❑ 애플리케이션 — fetch beer, elevator ...
- ❑ 시뮬레이터 — gazebo, player/stage, STDR Simulator ...
- ❑ 지능모듈 — navigation, action, grasping ...
- ❑ 라이브러리 — tf, PCL, OpenCV, OpenRave ...
- ❑ 디바이스드라이버 — camera_drivers, urg_node ...
- ❑ 디버그 툴 — rviz, rqt_graph, rosbag, rostopic ...
- ❑ 메시지통신 — rosmaster, rosmsg, rosservice ...
- ❑ 실행 툴 — rosrun, roslaunch ...
- ❑ 컴파일 툴 — catkin_make, rosbuild ...
- ❑ 파일시스템 — roscd, rosls
- ❑ 설치 툴 — rosinstall

ROBOT, SENSOR

- ❑ 90종류 이상의 로봇, 80종류 이상의 센서 지원

3. ROS Robot Operating System

- ROS 구성

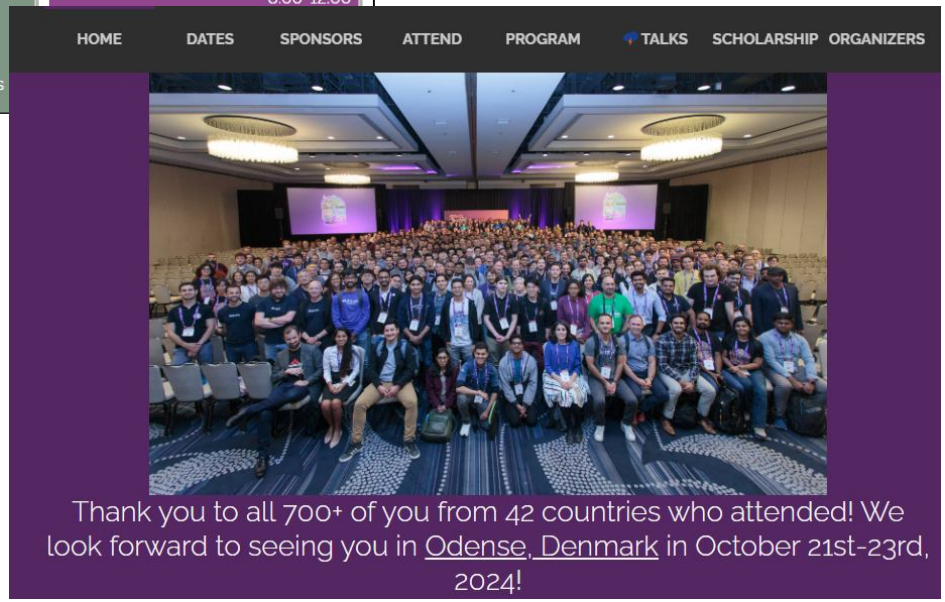
  : ROSCON(ROS Conference)

  - developers conference

  - https://roscon.ros.org/2023/

St. Paul, Minnesota, USA
19-20 May 2012

Madrid, Spain
September 29th-30th, 2018



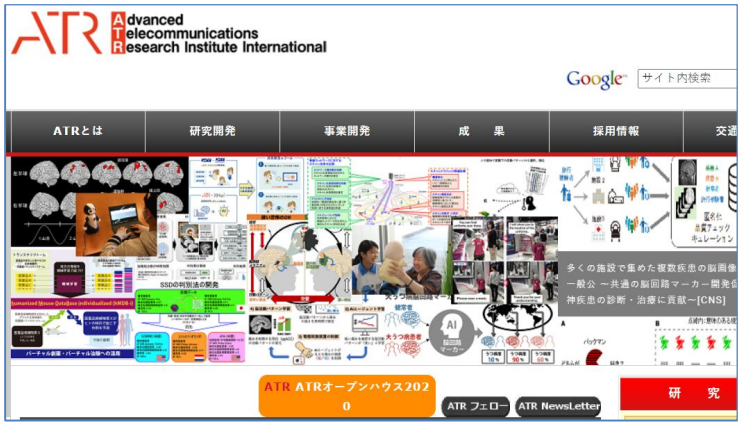
3. ROS Robot Operating System

- ROS 구성

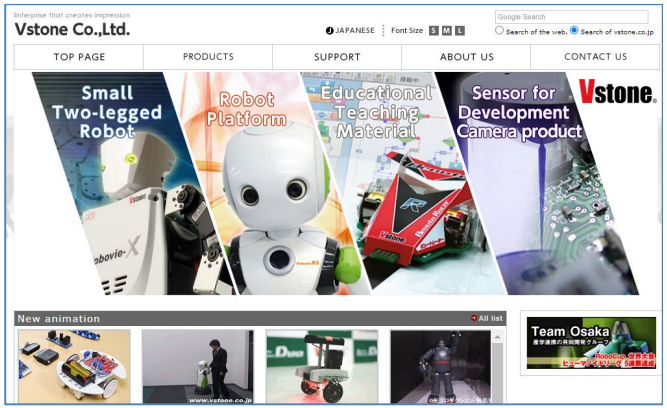
  : 센서 및 로봇분야

  - ATR(Advanced Telecommunications Research institute international)

    https://www.atr.jp



    https://www.vstone.co.jp/english/index.html



  - Robot Shop

    https://www.vstone.co.jp/robotshop/

3. ROS Robot Operating System

3-2. About ROS

- ROS 특징

: 분산 프로세스

- 최소 단위의 실행 가능한 프로세스(Node) 형태로 프로그램, 각 프로세스는 독립적으로 실행되면서 데이터를 주고 받음
- Node 간의 송/수신되는 데이터인 메시지를 저장하고 재사용 가능

: 패키지 단위 관리

- 같은 목적을 갖는 다양한 프로세스를 패키지 단위로 관리하기 때문에 개발, 공유, 수정 후 재배포 쉬움

: 공개 repository

- 개발자가 선호하는 공개 repository (GitHub 등) 등에 패키지를 공개 및 각 라이선스도 배포

: API(Application Programming Interface) 형태

- ROS를 활용한 프로그램을 개발할 때 단순히 API를 불러와 자신이 사용하던 코드에 쉽게 삽입할 수 있도록 설계
- ROS 프로그래밍과 C++, Python 프로그램

: 복수의 프로그래밍 언어 지원

- 다양한 언어를 지원하기 위해 Client Library를 제공.
- Python, C++ 등 로봇 분야에서 많이 사용하는 프로그래밍 언어와 JAVA, C# 등의 언어에서도 사용가능

3. ROS Robot Operating System

3-2. About ROS

- ROS 특징

: 로봇에 대한 표준 메시지 정의

- 카메라, IMU, 레이저, 센서와 경로 및 지도 등의 내비게이션 데이터 등 표준 메시지 정의

➔ 모듈화, 협업 작업을 유도, 효율성 향상

: 로봇 기하학 라이브러리 제공

- 진단 시스템, 센싱과 인식 레벨의 라이브러리

: 내비게이션

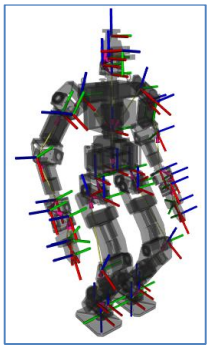
- 로봇에서 많이 사용되는 로봇의 포즈(위치/자세) 추정, 지도내의 자기 위치 추정 제공
- 지도 작성에 필요한 SLAM, 작성된 지도 내에서 목적지를 찾아가는 Navigation 라이브러리를 제공

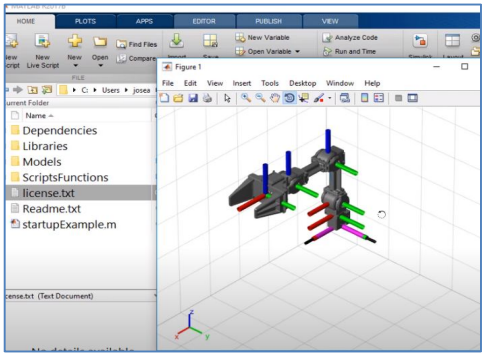
: Manipulation (운동학: kinematics)

- 로봇 암에 사용되는 IK(Inverse kinematics: Task -> Joint), FK(forward kinematics : J->T) 제공
- Pick and Place(평면상의 물체 위치 선정과 객체를 집는 과정)를 지원

➔ Manipulation algorithm (https://youtu.be/5DnKot3mMSc) tools제공(Moveit)

Moveit ➔ 운동 계획과 제어를 위한 도구 toolkit



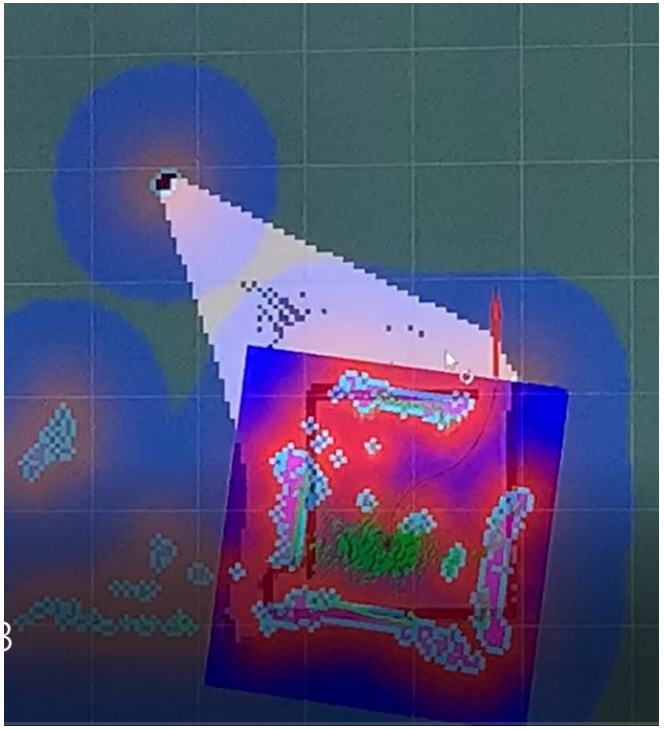


3. ROS Robot Operating System

- ROS 특징

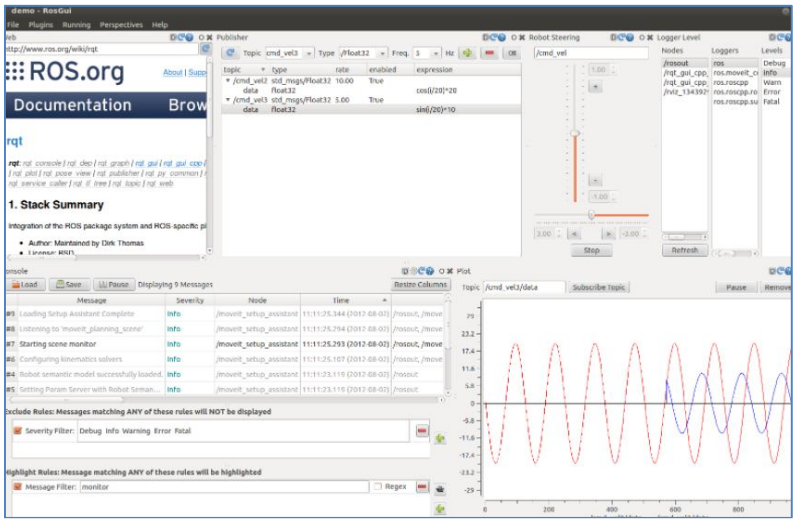
: Rviz

- ROS에서 제공하는 3D visualization tool



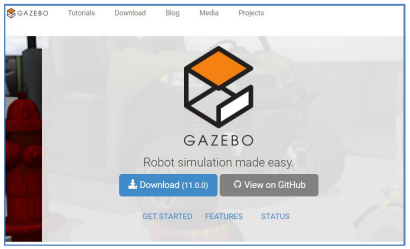
: RQT

- Qt(Qt Project : open collaboration)기반의 framework for GUI
- 3parts ➔ 노드와 그들 사이의 연결 정보 표시(rqt_graph)
  ➔ 인코더, 전압, 시간에 따라 변화하는 숫자를 플롯(rqt_plot)
  ➔ 데이터를 메시지 형태로 기록하고 재생(rqt_bag)



: Gazebo (http://gazebosim.org)

- 물리 엔진을 탑재
- 로봇, 센서, 환경 모델 등을 3차원 시뮬레이터



3. ROS Robot Operating System

3-2. About ROS

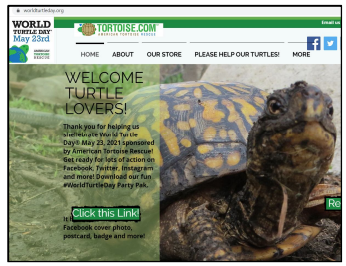
- ROS version

: 2010년 ROS 1.0 release

- Box Turtle, C Turtle, Diamondback, Electric Emys, Fuerte, Groovy Galapagos, Hydro Medusa, Indigo Igloo, Jade Turtle, Kinetic Kame, Lunar Loggerhead ...

: 각 버전마다 일러스트와 거북이 아이콘(https://www.worldturtleday.org)

- 거북이 아이콘은 turtlesim이라는 ROS 공식 tutorial 시뮬레이션 ➔ Turtlebot 유례



: ROS version 주기는 ROS가 공식적으로 지원하는 운영체제 우분투(Ubuntu) release 주기와 동일

: ROS 서포트 기간은 각 버전 별로 release 후 2년간 지원

: 2년에 한 번씩 release 되는

Ubuntu LTS(Long Term Support) version에 맞추어 나오는 ROS version은 LTS 서비스가 종료되는 5년간 지원

: 2016년의 Ubuntu 16.04 LTS를 지원하는 ROS 버전인 ROS Kinetic Kame 버전은 2021년 5월까지 지원

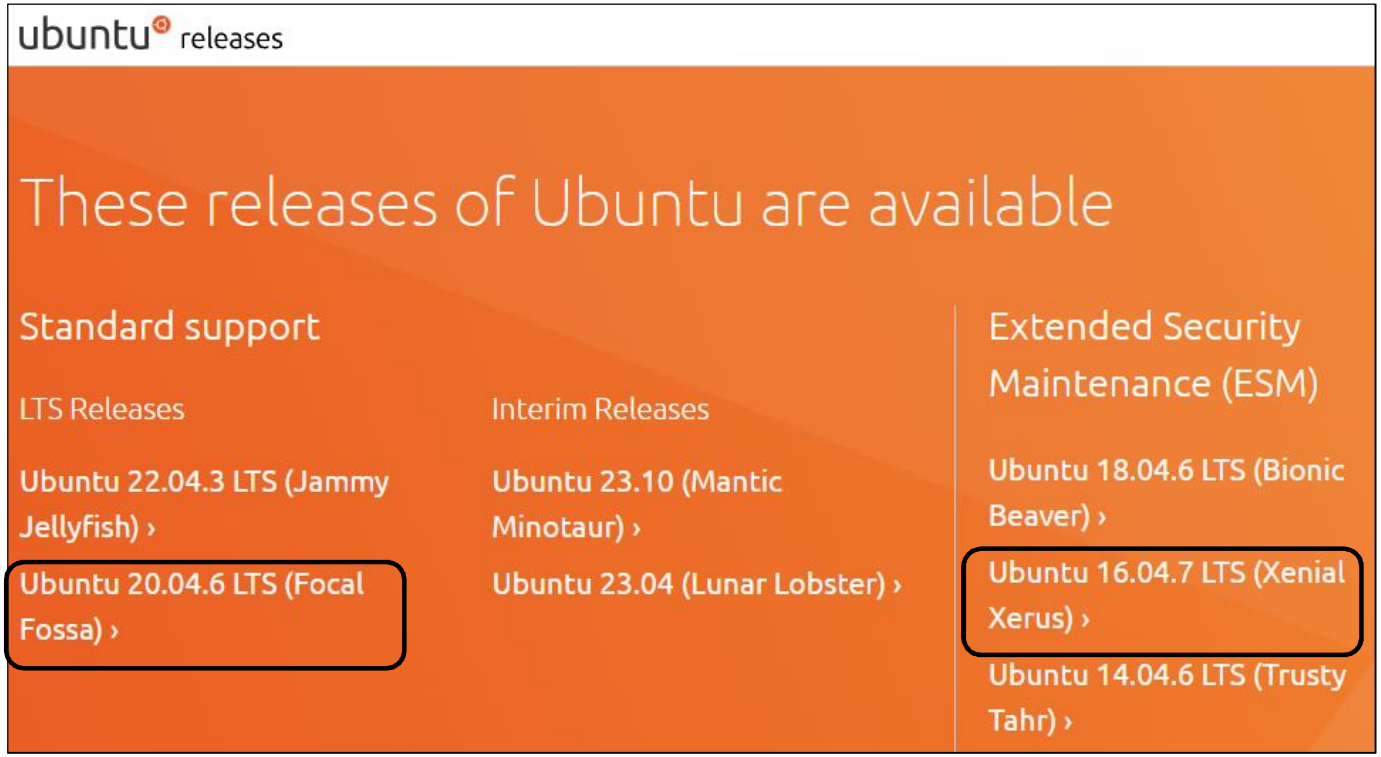
➔ Ubuntu 16.04 LTS (Xenial Xerus) 지원기간 연장(https://ubuntu.com / https://releases.ubuntu.com)

" Ubuntu 16.04 LTS is supported until 2024 through Canonicals' Extended Security Maintenance (ESM) product."

3. ROS Robot Operating System

[Ubuntu]

: https://ubuntu.com / https://releases.ubuntu.com



Ubuntu 20.04.3 LTS (Focal Fossa)

: Raspberry PI 4B

: ROS 1.0 Noetic

Ubuntu 16.04.7 LTS (Xenial Xerus)

: Raspberry PI 3 B+

: ROS 1.0 Kinetic Kame

3. ROS Robot Operating System

: Ref. https://www.ros.org/reps/rep-0003.html#noetic-ninjemys-may-2020-may-2025

: ROS 1.0

=> Kinetic Kame

=> Noetic

Kinetic Kame (May 2016 - May 2021)

Required Support for:

- Ubuntu Wily (15.10)
- Ubuntu Xenial (16.04)

Ubuntu 16.04.7 LTS (Xenial Xerus)
: Raspberry PI 3 B+

Recommended Support for:

- Debian Jessie
- Fedora 23
- Fedora 24

Minimum Requirements:

- C++11
  - GCC 4.9 on Linux, as it's the version that Debian Jessie ships with
- Python 2.7
  - Python 3.4 not required, but testing against it is recommended
- Lisp SBCL 1.2.4
- CMake 3.0.2
  - Debian Jessie ships with CMake 3.0.2
- Boost 1.55
  - Debian Jessie ships with Boost 1.55

Exact or Series Requirements:

- Ogre3D 1.9.x
- Gazebo 7
- PCL 1.7.x
- OpenCV 3.x
- Qt 5.3.x
- PyQt5

Lunar Loggerhead (May 2017 - May 2019)

Required Support for:

- Ubuntu Xenial (16.04)
- Ubuntu Yakkety (16.10)
- Ubuntu Zesty (17.04)

Ubuntu 16.04.7 LTS (Xenial Xerus)

Melodic Morenia (May 2018 - May 2023)

Required Support for:

- Ubuntu Artful (17.10)
- Ubuntu Bionic (18.04)

Ubuntu 18.04.5 LTS (Bionic Beaver)

Noetic Ninjemys (May 2020 - May 2025)

Required Support for:

- Ubuntu Focal Fossa (20.04)

Ubuntu 20.04.3 LTS (Focal Fossa)
: Raspberry PI 4B

3. ROS Robot Operating System

3-2. About ROS

- ROS version

  : 2010년

  ROS 1.0 release

  : 2017년

  ROS 2.0 release

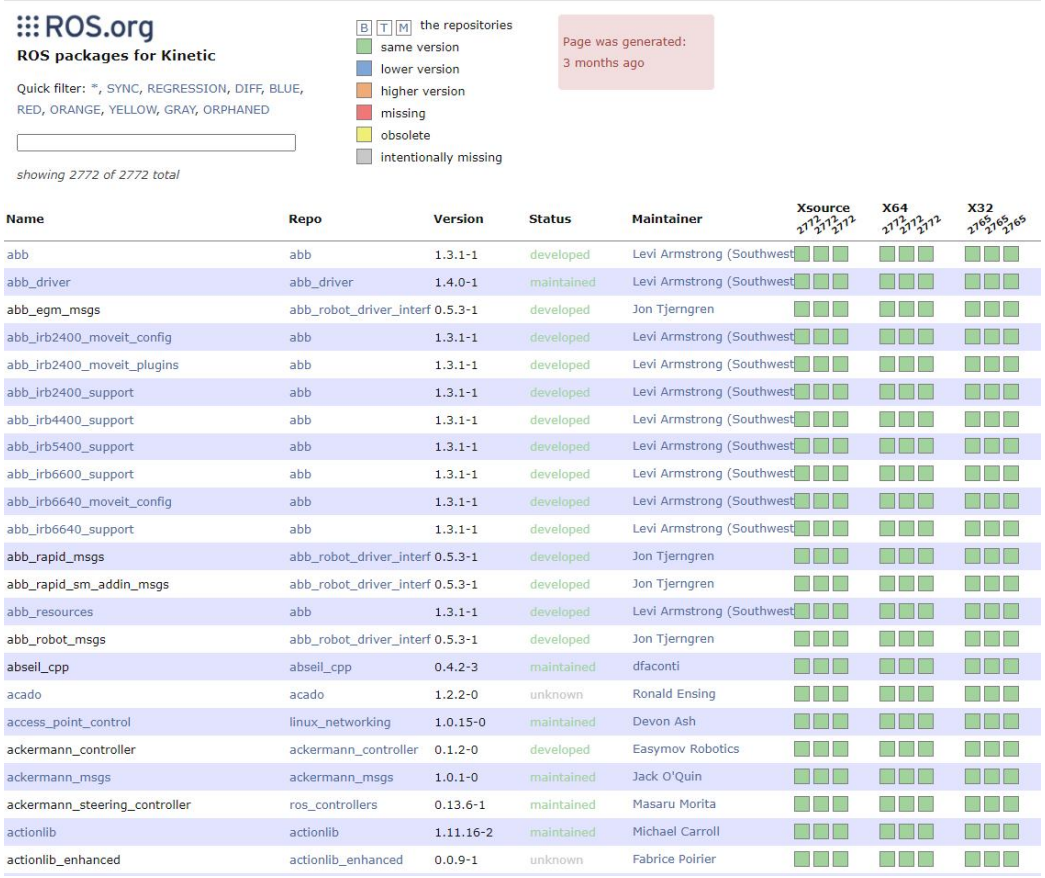
Distro	Release date	Poster	*Turtle*, turtle in tutorial	EOL date
ROS Noetic Ninjemys (Recommended)	May 23rd, 2020			May, 2025 (Focal EOL)
ROS Melodic Morenia	May 23rd, 2018			May, 2023 (Bionic EOL)
ROS Lunar Loggerhead	May 23rd, 2017			May, 2019
ROS Kinetic Kame	May 23rd, 2016			April, 2021 (Xenial EOL)
ROS Jade Turtle	May 23rd, 2015			May, 2017
ROS Indigo Igloo	July 22nd, 2014			April, 2019 (Trusty EOL)
ROS Hydro Medusa	September 4th, 2013			May, 2015
ROS Groovy Galapagos	December 31, 2012			July, 2014
ROS Fuerte Turtle	April 23, 2012			--
ROS Electric Emys	August 30, 2011			--
ROS Diamondback	March 2, 2011			--
ROS C Turtle	August 2, 2010			--
ROS Box Turtle	March 2, 2010			--

3. ROS Robot Operating System

- **ROS version**

  : **ROS packages for Kinetic**

  - **http://repositories.ros.org/status_page/ros_kinetic_default.html**



ROS.org

ROS packages for Kinetic

Quick filter: *, SYNC, REGRESSION, DIFF, BLUE, RED, ORANGE, YELLOW, GRAY, ORPHANED

showing 2772 of 2772 total

B T M the repositories
same version
lower version
higher version
missing
obsolete
intentionally missing

Page was generated: 3 months ago

Name	Repo	Version	Status	Maintainer	Xsource 2772 2772 2772	X64 2772 2772 2772	X32 2765 2765 2765
abb	abb	1.3.1-1	developed	Levi Armstrong (Southwest			
abb_driver	abb_driver	1.4.0-1	maintained	Levi Armstrong (Southwest			
abb_egm_msgs	abb_robot_driver_interf	0.5.3-1	developed	Jon Tjerngren			
abb_irb2400_moveit_config	abb	1.3.1-1	developed	Levi Armstrong (Southwest			
abb_irb2400_moveit_plugins	abb	1.3.1-1	developed	Levi Armstrong (Southwest			
abb_irb2400_support	abb	1.3.1-1	developed	Levi Armstrong (Southwest			
abb_irb4400_support	abb	1.3.1-1	developed	Levi Armstrong (Southwest			
abb_irb5400_support	abb	1.3.1-1	developed	Levi Armstrong (Southwest			
abb_irb6600_support	abb	1.3.1-1	developed	Levi Armstrong (Southwest			
abb_irb6640_moveit_config	abb	1.3.1-1	developed	Levi Armstrong (Southwest			
abb_irb6640_support	abb	1.3.1-1	developed	Levi Armstrong (Southwest			
abb_rapid_msgs	abb_robot_driver_interf	0.5.3-1	developed	Jon Tjerngren			
abb_rapid_sm_addin_msgs	abb_robot_driver_interf	0.5.3-1	developed	Jon Tjerngren			
abb_resources	abb	1.3.1-1	developed	Levi Armstrong (Southwest			
abb_robot_msgs	abb_robot_driver_interf	0.5.3-1	developed	Jon Tjerngren			
abseil_cpp	abseil_cpp	0.4.2-3	maintained	dfaconti			
acado	acado	1.2.2-0	unknown	Ronald Ensing			
access_point_control	linux_networking	1.0.15-0	maintained	Devon Ash			
ackermann_controller	ackermann_controller	0.1.2-0	developed	Easymov Robotics			
ackermann_msgs	ackermann_msgs	1.0.1-0	maintained	Jack O'Quin			
ackermann_steering_controller	ros_controllers	0.13.6-1	maintained	Masaru Morita			
actionlib	actionlib	1.11.16-2	maintained	Michael Carroll			
actionlib_enhanced	actionlib_enhanced	0.0.9-1	unknown	Fabrice Poirier			

← 분량이 방대하여, 실질적인 개발에 도움이 됨

3. ROS Robot Operating System

- **ROS version**

  : **ROS packages for Noetic**

  - **http://repositories.ros.org/status_page/ros_noetic_default.html**

ROS.org

ROS packages for Noetic

Quick filter: *, SYNC, REGRESSION, DIFF, BLUE, RED, ORANGE, YELLOW, GRAY, ORPHANED

showing 2012 of 2012 total

B T M the repositories
same version
lower version
higher version
missing
obsolete
intentionally missing

Page was generated:
2 minutes ago

Name	Repo	Version	Status	Maintainer	Fsource 2010 2010 2007	F64 2007 2007 2004
abb_driver	abb_driver	1.4.0-1	maintained	Levi Armstrong (Southwe...		
abb_egm_msgs	abb_robot_driver_interf	0.5.2-1	developed	Jon Tjerngren		
abb_rapid_msgs	abb_robot_driver_interf	0.5.2-1	developed	Jon Tjerngren		
abb_rapid_sm_addin_msgs	abb_robot_driver_interf	0.5.2-1	developed	Jon Tjerngren		
abb_robot_msgs	abb_robot_driver_interf	0.5.2-1	developed	Jon Tjerngren		
ackermann_msgs	ackermann_msgs	1.0.2-1	maintained	Jack O'Quin		
ackermann_steering_controller	ros_controllers	0.21.2-1	maintained	Masaru Morita Bence Magyar		
actionlib	actionlib	1.14.0-1	maintained	Michael Carroll Jacob Perron		
actionlib_lisp	roslisp_common	0.2.15-1	maintained	Gayane Kazhoyan Georg Bartels		
actionlib_msgs	common_msgs	1.13.1-1	maintained	Michel Hidalgo		

← 분량이 방대하여, 실질적인 개발에 도움이 됨

3. ROS Robot Operating System

- ROS 2

  : 2018.07.02 - ROS 2 2nd release (Bouncy Bolson)

  : 2017.12.08 - ROS 2 1st release (Ardent Apalone)

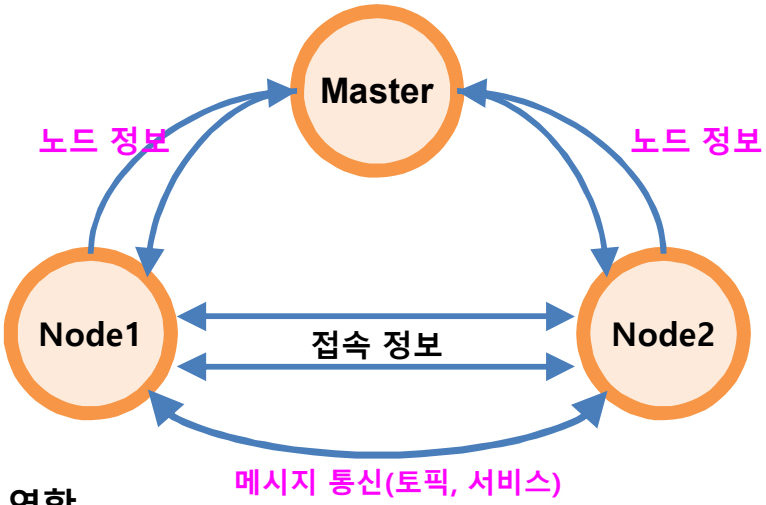
	ROS 1	ROS 2
Platforms	Ubuntu, OS X	Ubuntu, OS X, Windows
Communication	XMLRPC + ROSTPC	DDS
Languages	C++03, Python 2	C++11, Python 3 (3.5+)
Build system	rosbuild → catkin	ament
Messages, Services	*.msg, *.srv	new *.msg, *.srv, *.msg.idl, *.srv.idl
roslaunch	XML	written in Python
multiple nodes	one node in a process	multiple nodes in a process
real-time	external frameworks like Orocos	real-time nodes when using a proper RTOS with carefully written user code
Graph API	remapping at startup time only	remapping at runtime
Embedded systems	rosserial (UART)	FreeRTPS (UART, Ethernet, WiFi, et. al.)
		No non-isolated build
		No devel space

3. ROS Robot Operating System

3-3. ROS 용어

: http://wiki.ros.org

- ROS Wiki. ROS의 기본적인 사용법과 각 패키지에 대한 설명, 사용되는 parameter, 저작자, 라이선스, 홈페이지, repository, tutorial 등이 기술



- ROS Master (http://wiki.ros.org/Master)

: 실행 명령어 roscore

: Node와 Node 사이의 연결과 메시지 통신을 위한 네임 서버와 같은 역할

: Master가 실행하면 각 node의 이름을 등록, 필요에 따라 정보를 받음

- Master 없이, node 간의 접속, topic 서비스와 같은 메시지 통신을 할 수 없음

: Master는 http 기반의 프로토콜인 XMLRPC(XML-Remote Procedure Call)를 이용 slave과 통신

- ROS 구동하면, master는 사용자가 정해 놓은 ROS_MASTER_URI 변수에 기재된 URI 주소와 포트를 갖음

3. ROS Robot Operating System

3-3. ROS 용어

- **ROS Node** (http://wiki.ros.org/Nodes)

  : ROS에서 실행되는 최소 단위의 프로세서를 의미. 즉 하나의 실행 가능한 프로그램

  : 하나의 목적에 하나의 node를 작성(각 프로그램 세분화)

  - 센서 드라이브, 센서 데이터를 이용한 변환, 장애물 판단, 모터 구동, Encoder 입력, 내비게이션 등 세분화된 node

  : node 구동 후 master에 node name, topic 및 service name, message type(ROS naming사용), URI address, Port를 등록

  - for publisher, subscriber, service server, service client에서 사용하기 위해

  : 각 node는 node간에 topic과 service를 이용하여 message를 주고받음

  - node는 master와 통신할 때 XMLRPC(eXtensible Markup Language-Remote Procedure Call) 이용

  - node 간의 통신에서는 XMLRPC나 TCP/IP 통신 계열의 TCPROS(ROS message 통신)를 이용

  ➔ http://wiki.ros.org/ROS/TCPROS

  ➔ node간의 접속 요청과 응답은 XMLRPC를 사용하며, 메시지 통신은 노드와 노드 간의 TCPROS를 이용

  : URI 주소와 Port는 현재 노드가 실행 중인 컴퓨터에 저장된 ROS_HOSTNAME 환경변수 값을 URI 주소로 사용

3. ROS Robot Operating System

3-3. ROS 용어

- **ROS Package** (http://wiki.ros.org/Packages)

  : ROS 응용프로그램은 Package 단위로 개발

  Package는 최소한 하나 이상의 node를 포함하거나 다른 Package의 node를 실행하기 위한 설정 파일을 포함

  : ROS Kinetic package http://repositories.ros.org/status_page/ros_kinetic_default.html

  : Metapackage

  - 공통된 목적을 지닌 패키지들의 집합(ROS의 file system concepts)

- **ROS Message** (http://wiki.ros.org/Messages)

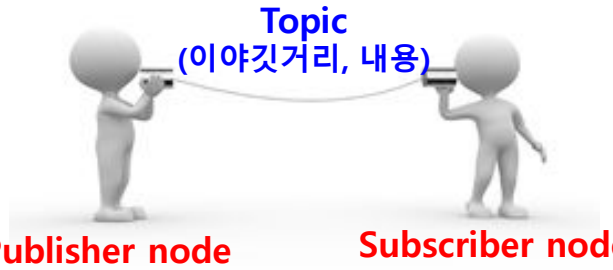
  : 노드 간의 데이터를 주고 받음. 메시지는 integer, floating point, Boolean 같은 변수 형태

Ref. http://www.dreamstime.com/illustration/people-talk-listen-tin-can-phone-communication.html

3. ROS Robot Operating System

3-3. ROS 용어

- **ROS Topic** (http://wiki.ros.org/Topics)



Publisher node

① topic과 자신의 정보를 master 등록

② topic을 message형태로 송신

➔ Publish 라고 함

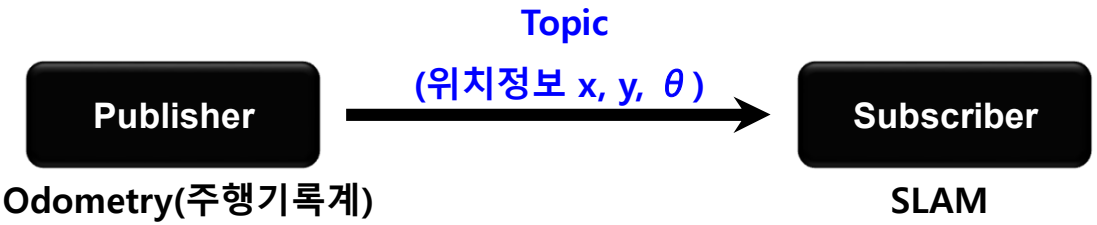
Subscriber node

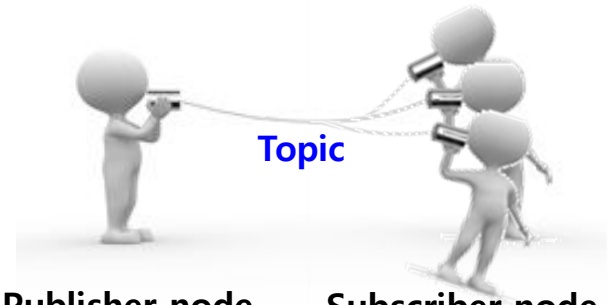
① topic과 자신의 정보를 master 등록

② topic을 전송하는 publisher node정보를 master로 받음

③ publisher node와 접속하여 message를 받음

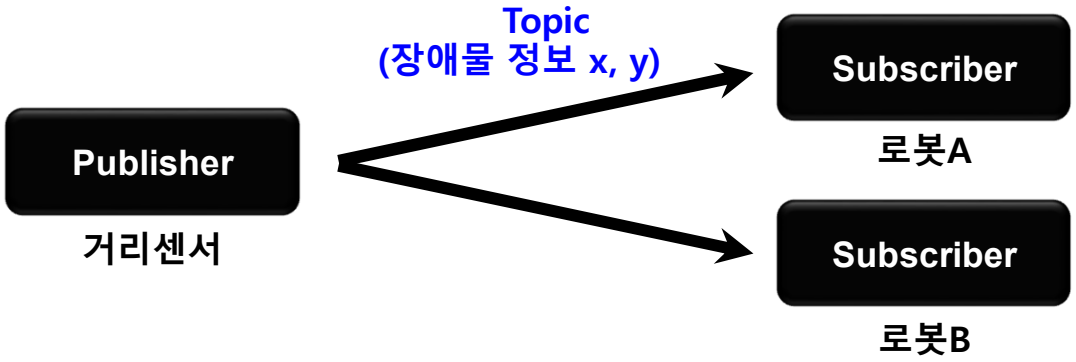
➔ Subscribe 라고 함





Publisher node Subscriber node

1:N, N:1, N:N 통신



Ref. http://www.dreamstime.com/illustration/people-talk-listen-tin-can-phone-communication.html

3. ROS Robot Operating System

3-3. ROS 용어

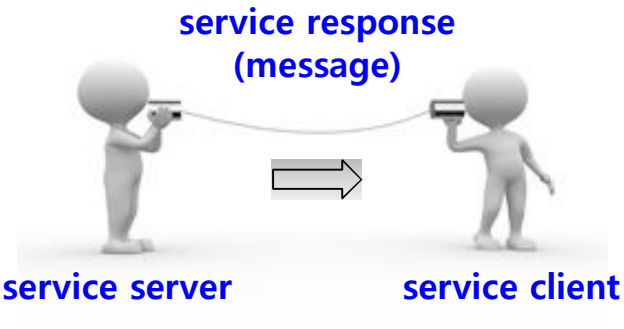
- **ROS Service (http://wiki.ros.org/Services)**

  : ROS topic 통신 방식은 비동기 방식(필요에 따라 data 송수신)

  : ROS는 필요에 따라 동기 방식 사용 ➔ Service 명칭

  - 요청이 있을 때 응답 하는 Service server, 요청하고 응답 받는 Service client로 구분
  - Service는 토픽과는 달리 일회성 메시지 통신. 서비스의 요청과 응답이 완료되면 연결된 두 노드의 접속은 끊김

① 서비스 요청을 받으면 지정된 서비스를 수행
② 수행한 결과를 서비스 클라이언트에 전달
➔ 서비스 서버는 정해진 명령을 받아 수행하는 노드에 사용



① 서비스 요청을 서비스 서버에 전달
② 서비스 결과값을 수신
➔ 서비스 클라이언트는 정해진 명령을 지시하고 결과값을 받는 노드에 사용

3. ROS Robot Operating System

Ref. http://www.dreamstime.com/illustration/people-talk-listen-tin-can-phone-communication.html

3-3. ROS 용어

- **ROS Action** (http://wiki.ros.org/actionlib)

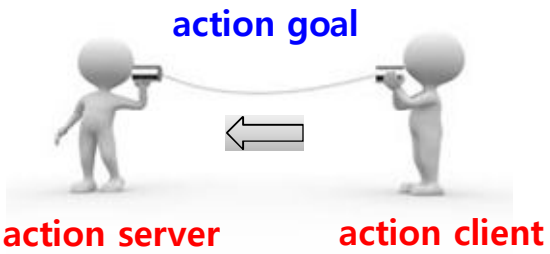
  : 요청 처리 후 응답까지 오랜 시간 필요하고

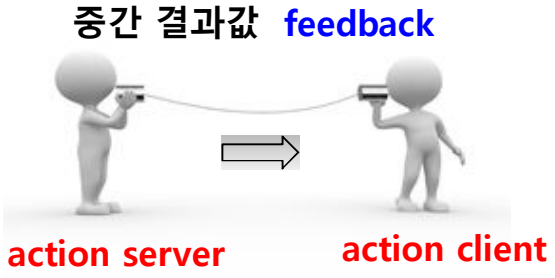
  중간 결과값이 필요한 경우에 사용되는 message 통신 방식

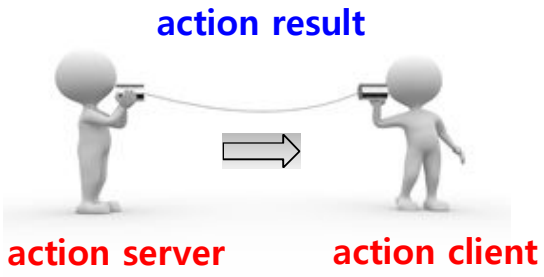
  : 요청과 응답에 해당되는 goal과 result, 중간 결과값의 feedback 추가

  - Action의 goal을 정하는 action client
  - goal에 맞춰 일을 수행하고 action result, feedback 전달하는 action server

  : 비동기 방식 양방향 메시지 통신







3. ROS Robot Operating System

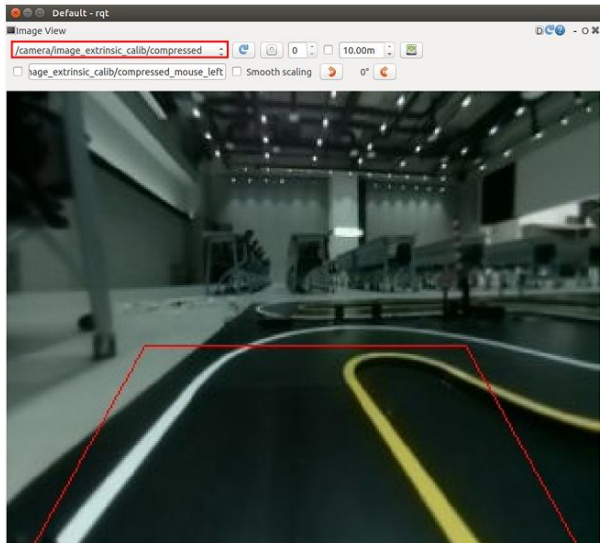
3-3. ROS 용어

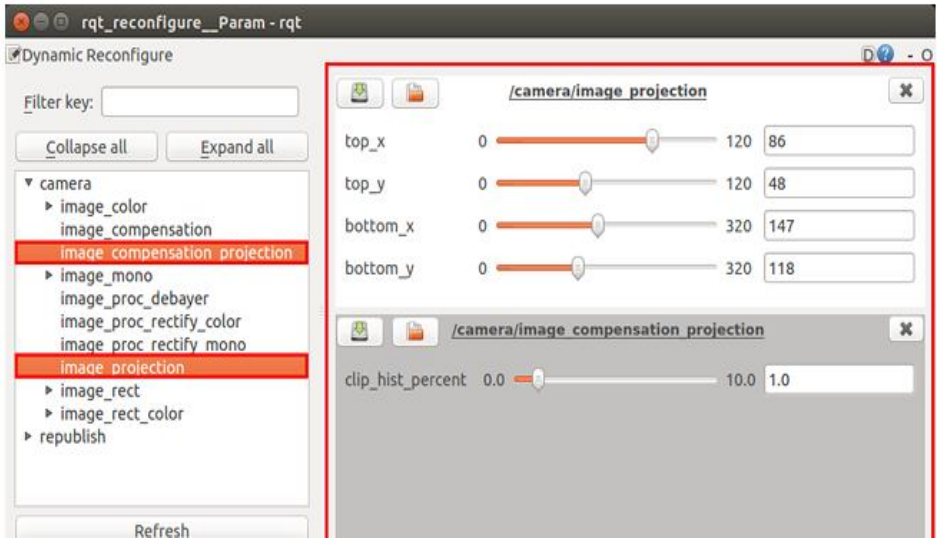
- **ROS Parameter** (http://wiki.ros.org/Parameter%20Server#Parameters)

  : Parameter는 default로 설정 값들이 지정되어, 필요에 따라 외부에서 읽거나 쓸 수 있음

  특히 외부에서 쓰기 기능을 이용하여 상황에 따라 설정 값을 실시간으로 바꿀 수 있음

  : 예를 들어 RPi Camera 설정





- **ROS Parameter server** (http://wiki.ros.org/Parameter%20Server)

  : Parameter 사용할 때, 각 Parameter를 등록하는 서버

Ref. http://www.dreamstime.com/illustration/people-talk-listen-tin-can-phone-communication.html

3. ROS Robot Operating System

3-3. ROS 용어

- **catkin (http://wiki.ros.org/catkin)**

  : low-level build system macros and infrastructure for ROS.

  : ROS와 관련된 빌드, 패키지 관리, 패키지 간의 의존관계를 편리하게 사용하기 위함

  : ROS의 빌드 시스템은 CMake(Cross Platform Make)를 이용, 패키지 폴더에 CMakeLists.txt 에 빌드 환경을 기술

  ➔ ROS에서 CMake를 ROS에 맞게 수정하여, ROS에 특화된 catkin build system

- ~~rosbuild (http://wiki.ros.org/rosbuild)~~

  ~~: ROS Groovy 이전 version에 적용된 build system~~

- **roscore (http://wiki.ros.org/roscore)**

  : ROS master를 구동하는 명령어

  : roscore는 같은 네트워크에서 하나만 구동 됨.

  - ROS를 구동하면 사용자가 정해놓은 ROS_MASTER_URI 변수에 기재된 URI 주소와 포트를 사용

3. ROS Robot Operating System

3-3. ROS 용어

- **rosrun** (http://wiki.ros.org/rosbash#rosrun)

  : ROS의 기본 실행 명령어

  : package에서 하나의 node를 실행하는데 사용

  - 노드가 사용하는 URI 주소와 포트는 현재 노드가 실행 중인 컴퓨터에 저장된 ROS_HOSTNAME 환경 변수 값을 URI 주소로 사용

- **roslaunch** (http://wiki.ros.org/roslaunch)

  : 여러 node를 실행하는데 사용. 명령어를 통해 하나 그 이상의 정해진 노드를 실행

  : node를 실행할 때, package의 parameter나 node name 변경, node name space 설정, ROS_ROOT 및 ROS_PACKAGE_PATH 설정, 환경변수 변경 등 node 실행에 특화된 ROS 명령어

  : roslaunch는 *.launch 파일을 사용하여 실행 node에 대한 설정을 해주며

  ➔ XML(Extensible Markup Language)에 기반, XML 태그 형태의 다양한 옵션을 제공

- **bag** (http://wiki.ros.org/Bags)

  : ROS에서 주고받는 메시지의 데이터를 저장하는 파일 포맷을 bag 라고 하며 확장자로 *.bag 사용

  : ROS에서는 이 bag를 이용하여 메시지를 저장하고 필요할 때 이를 재생하여 이전 상황을 그대로 재현

3. ROS Robot Operating System

3-3. ROS 용어

- repository

  : 공개된 package가 저장된 URL 주소.

  : svn, hg, git 등의 소스 관리 시스템을 이용하여 이슈, 개발, 내려 받기 등을 관리

  : 제공되는 ROS 패키지의 많은 수가 github에 각 소스 코드의 repository로 사용됨

- graph

  : node, topic, publisher, subscriber 관계는 graph를 통해 시각적으로 나타냄

  : 실행은 rqt_graph package의 rqt_graph node를 실행

  - 실행 명령에는 rqt_graph과 rosrun rqt_graph

- name

  : node, topic, parameter, service는 모두 name을 지님

  : name을 마스터에 등록하고 각 node의 topic, parameter, service를 사용할 때 이름을 기반으로 검색, 메시지를 전송

3. ROS Robot Operating System

3-3. ROS 용어

- client library (http://wiki.ros.org/Client%20Libraries)

  : ROS는 사용되는 언어의 의존성을 낮추기 위해 다양한 언어 지원

  : roscpp, rospy, roslisp, rosjava, roslua, roscs, roseus, PhaROS, rosR 등

- URI(Uniform Resource Identifier)

  : 통합 자원 식별자. 인터넷에 있는 자원을 나타내는 유일한 주소. 인터넷 프로토콜에서 식별자

- MD5(Message Digest algorithm5)

  : 128bit 암호화 Hash function(임의의 길이를 갖는 임의의 data를 고정 길이 data로 mapping)

  : 주로 프로그램이나 파일이 원본 그대로인지를 확인하는 무결성 검사

- RPC(Remote Procedure Call)

  : 멀리 떨어져(Remote) 있는 컴퓨터상의 프로그램이 다른 컴퓨터 내에 있는 서브 프로그램(Procedure)을 불러내는(Call) 것을 의미

- XML(Extensible Markup Language)

  : 태그 등을 이용하여 데이터의 구조를 명기하는 언어. ROS에서는 *.launch, *.urdf, package.xml 사용

3. ROS Robot Operating System

3-3. ROS 용어

- XMLRPC(eXtensible Markup Language - Remote Procedure Call)

  : RPC 프로토콜의 일종

  : 인코딩 형식에서는 XML을 채택

  전송 방식에서는 접속 상태를 유지하지 않고 & 점검하지 않는 요청과 응답 방식의 HTTP 프로토콜을 사용

- TCP(Transmission Control Protocol)

  : 전송 제어 프로토콜. TCP/IP라 부름.

  ➔ 이는 인터넷 프로토콜 계층의 시각에서 보면 IP(Internet Protocol)를 기반으로 전송 제어 프로토콜인 TCP를 사용하여 데이터의 전달을 보증하고 보낸 순서대로 송수신

  : 메시지 및 서비스에서 사용되는 TCP/IP 기반의 메시지 방식을 TCPROS

- CMakeLists.txt (http://wiki.ros.org/catkin/CMakeLists.txt)

  : ROS 빌드 시스템 catkin은 기본적으로 CMake 이용 ➔ package 폴더에 CMakeLists.txt 파일에 빌드 환경 기재

- package.xml

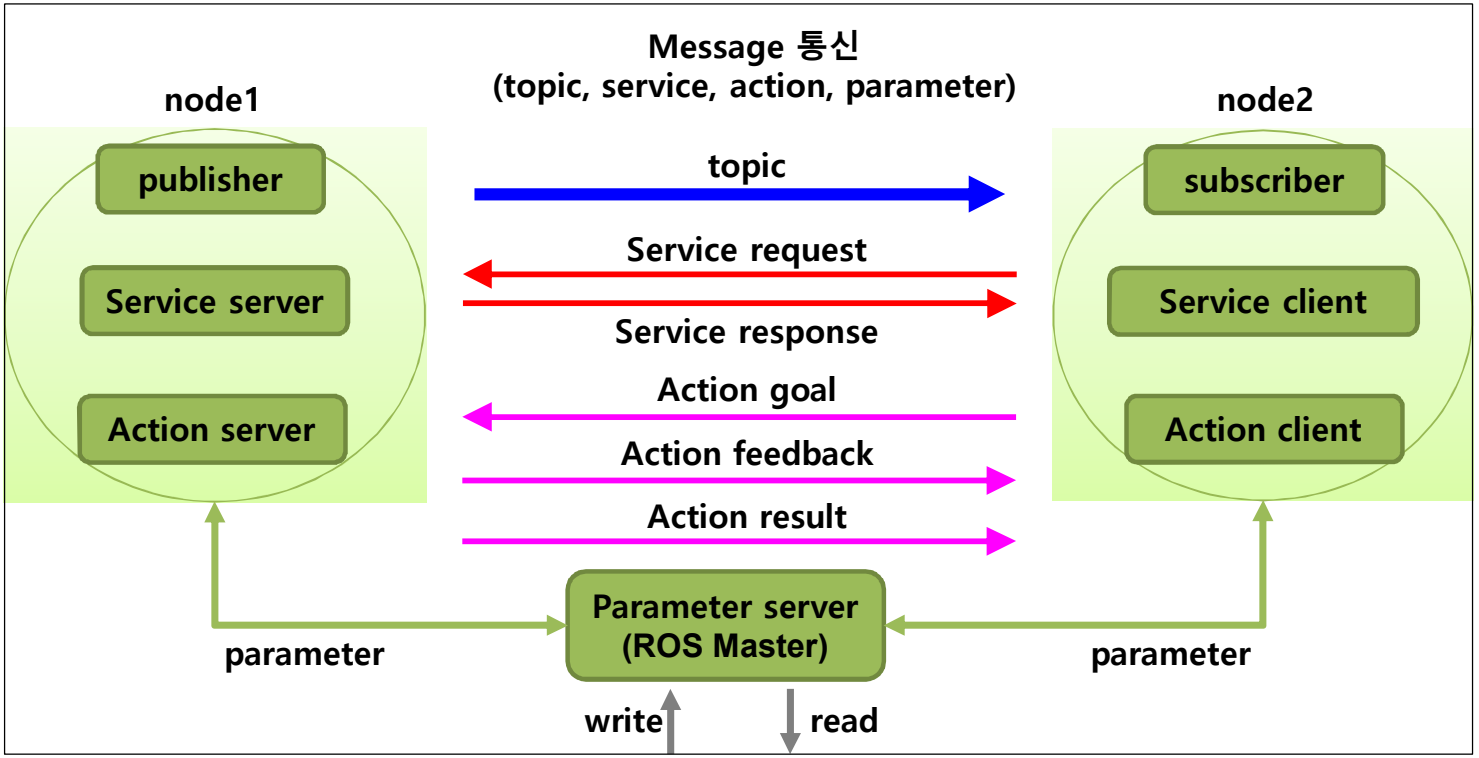
  : 패키지의 정보를 담은 XML 파일. 패키지의 이름, 저작자, 라이선스 등을 기술

3. ROS Robot Operating System

3-4. ROS Message 통신

: 최소 단위 실행 프로그램 node는 또 다른 node와 message를 통해 데이터를 주고받음 ➔ node 간의 메시지 통신

- topic : 단방향 메시지 송수신 방식
- service : 양방향 메시지 request/response 방식
- action : 양방향 메시지 goal/feedback/result 방식
- node 안에서 사용되는 parameter는 외부에서 변경 가능



감사합니다