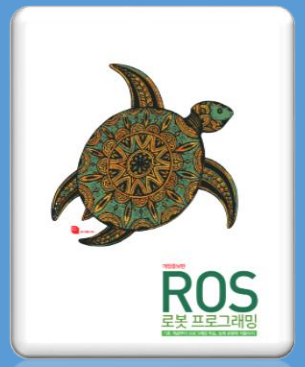
임베디드 시스템

ROBOTIS

Open Source Team

Yoonseok Pyo

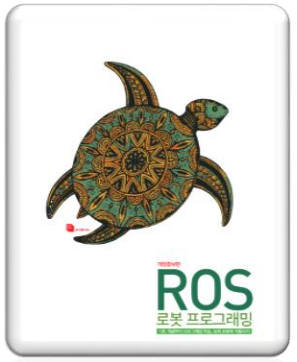


온라인강좌

YouTube Subscribe

교재
P. 244~291

Contents



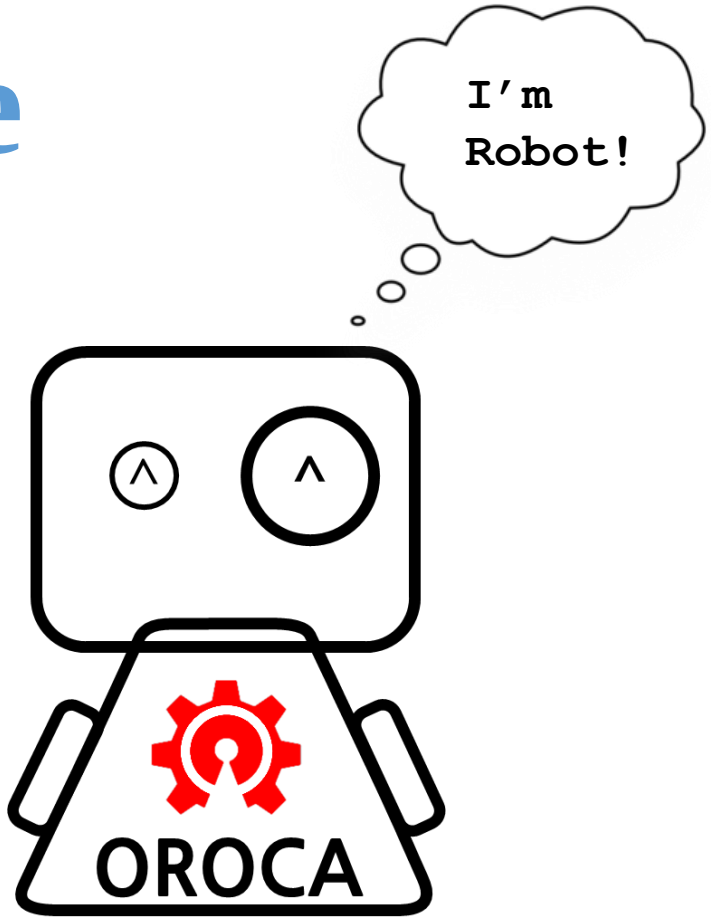
온라인강좌

교재
P. 244~291

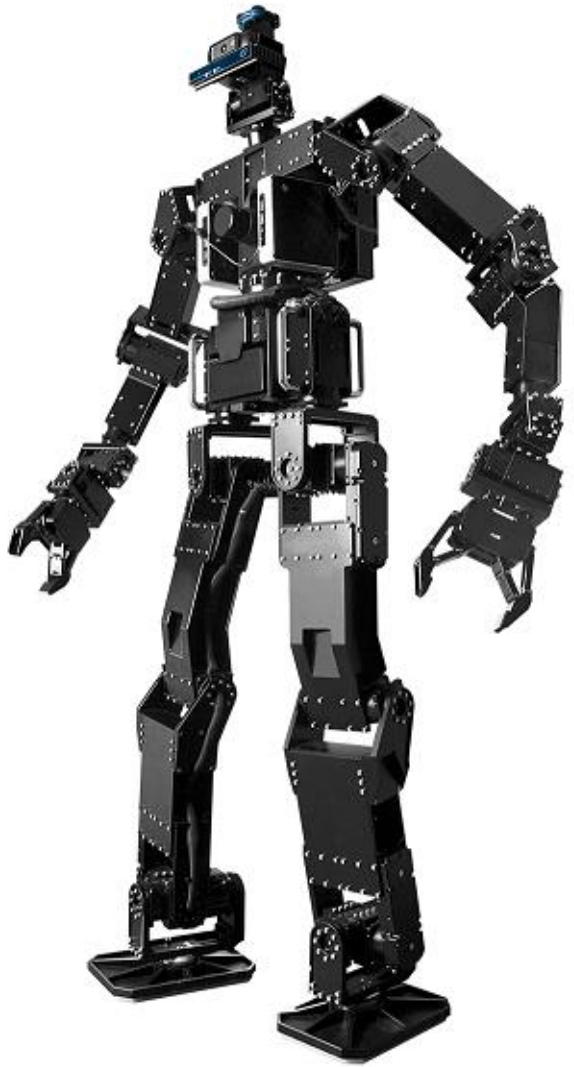
What is the Best Computing Resource for Your Robot?

CPU?

MCU?



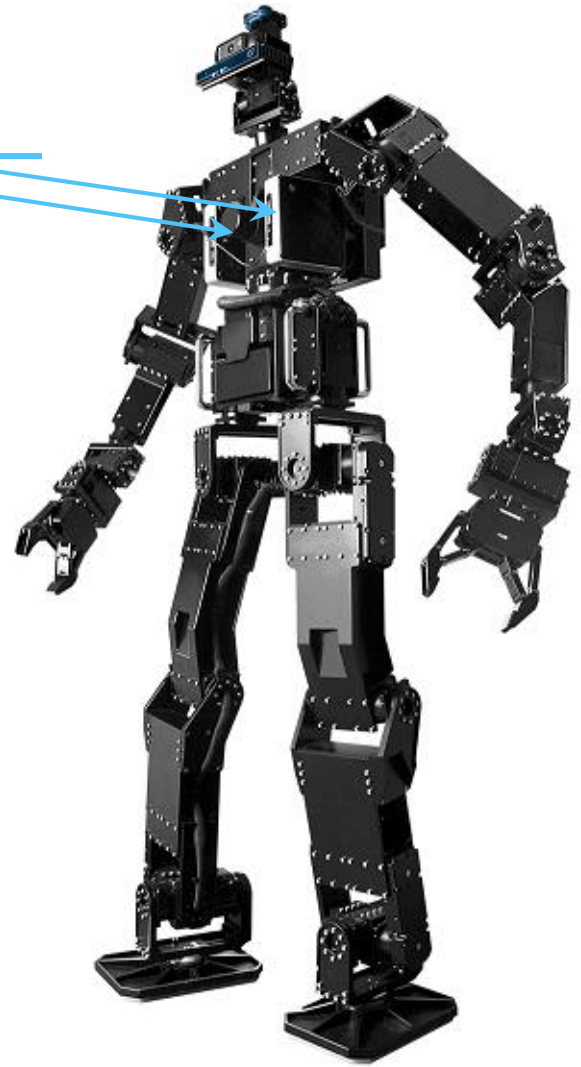
로봇에서 임베디드 시스템이 차지하는 비중



THORMANG3

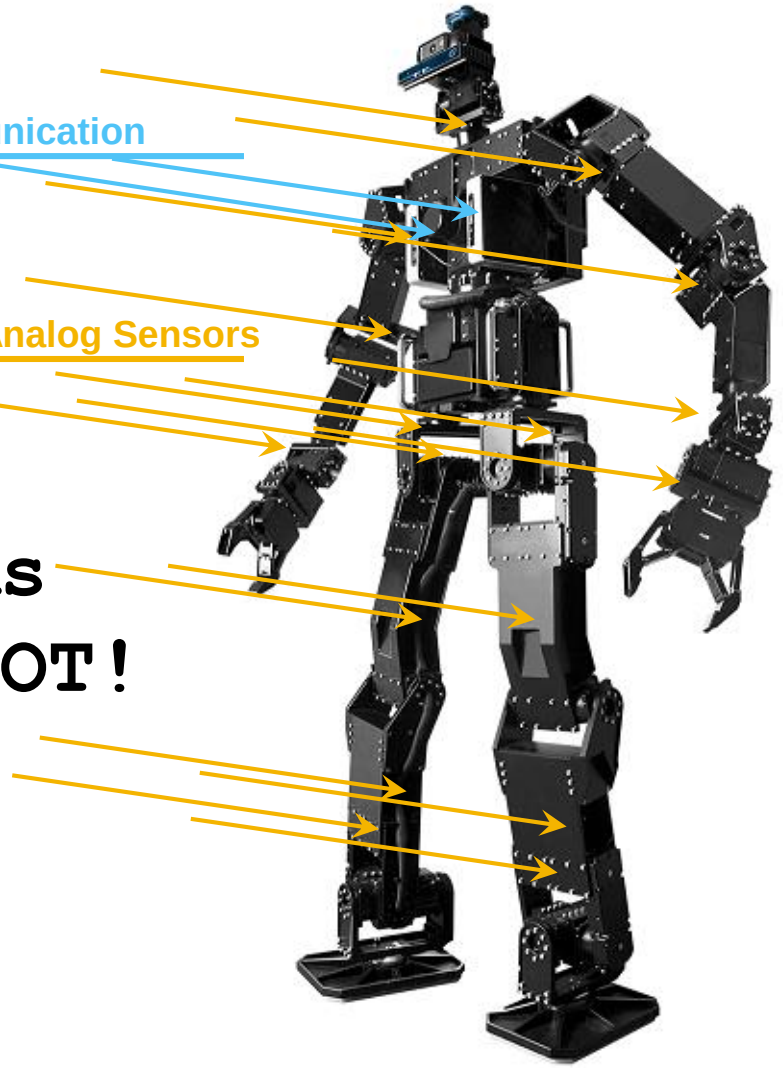
로봇에서 임베디드 시스템이 차지하는 비중

Intel NUC
for Sensor and Communication



THORMANG3

로봇에서 임베디드 시스템이 차지하는 비중



Small embedded systems are everywhere in ROBOT!

THORMANG3

컴퓨터 자원의 종류와 ROS 지원





	8/16-bit MCU	32-bit MCU		ARM A-class	x86
		"small" 32-bit MCU	"big" 32-bit MCU		
Example Chip	Atmel AVR	ARM Cortex-M0	ARM Cortex-M7	Samsung Exynos	Intel Core i5
Example System	Arduino Leonardo	Arduino M0 Pro	SAM V71	ODROID	Intel NUC
MIPS	10's	100's	100's	1000's	10000's
RAM	1-32 KB	32 KB	384 KB	a few GB (off-chip)	2-16 GB (SODIMM)
Max power	10's of mW	100's of mW	100's of mW	1000's of mW	10000's of mW
Peripherals	UART, USB FS, ...	USB FS	Ethernet, USB HS	Gigabit Ethernet	USB SS, PCIe

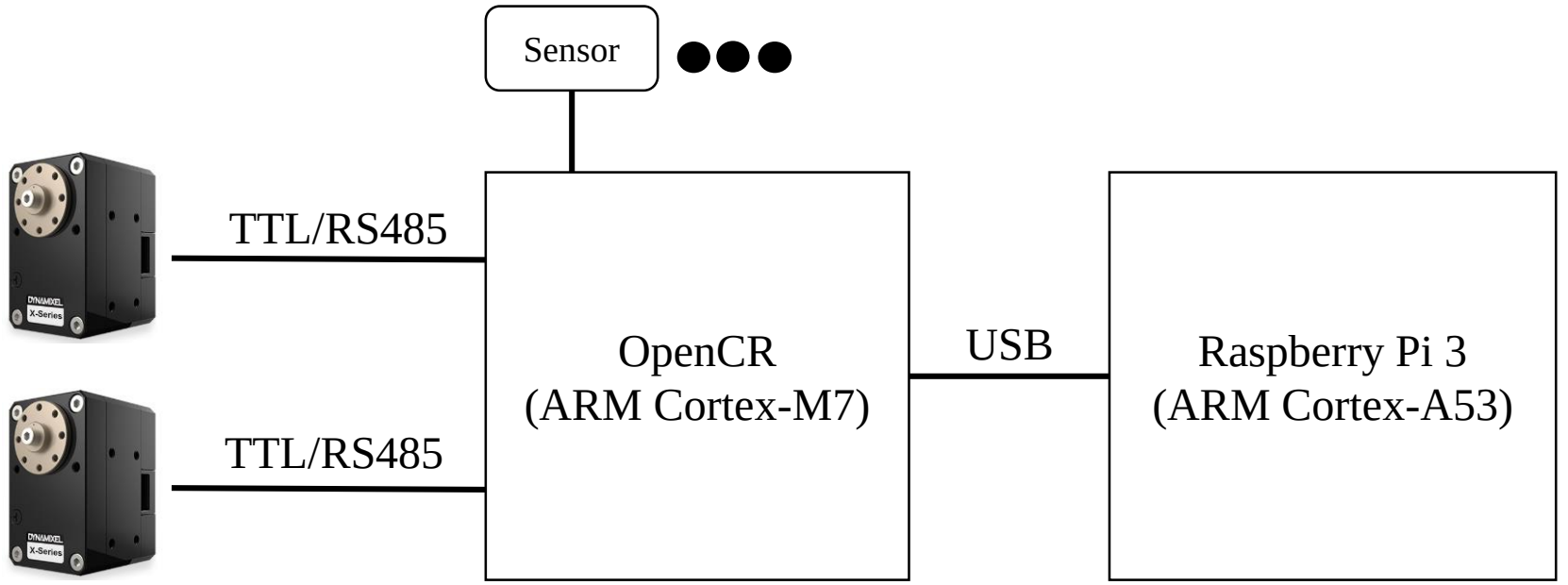
ROS 설치 불가능

ROS 설치 가능

Morgan Quigley (OSRF) "ROS 2 on “small” embedded systems", ROSCon2015

ROS에서 임베디드 시스템

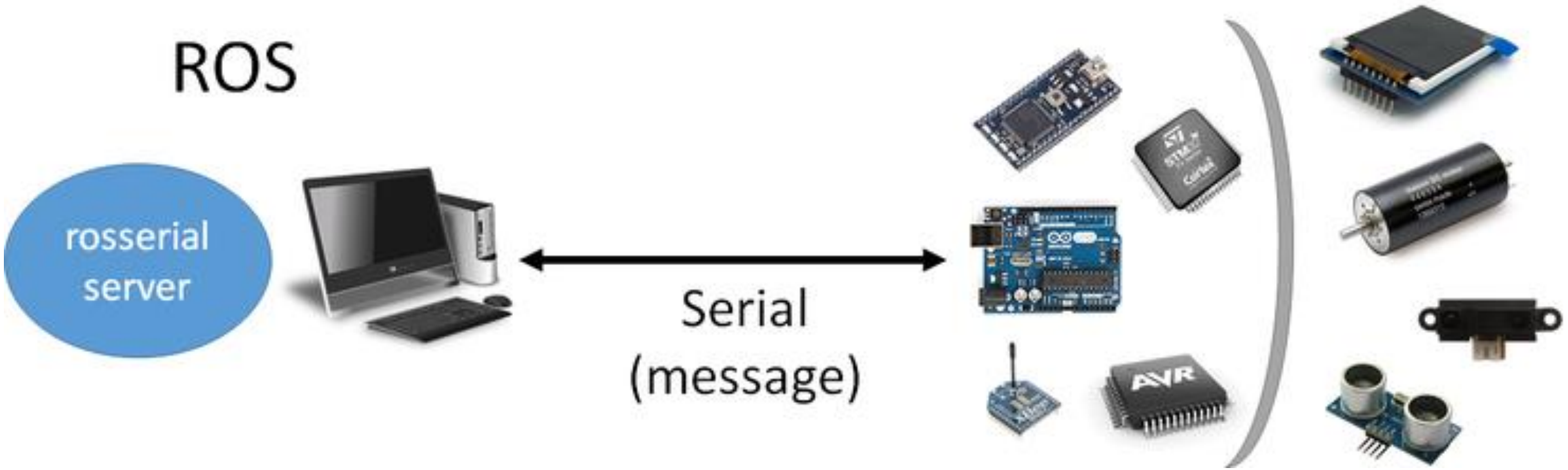
- PC와는 달리 임베디드 시스템에서는 ROS 설치가 불가능
- 실시간성 확보 및 하드웨어 제어를 위해서는
  ROS가 설치된 PC와 임베디드 시스템 간의 연결이 요구됨
- ROS에서는 이를 위해 rosserial 이라는 패키지를 제공!



[터틀봇3에서의 PC와 임베디드 시스템의 구성]

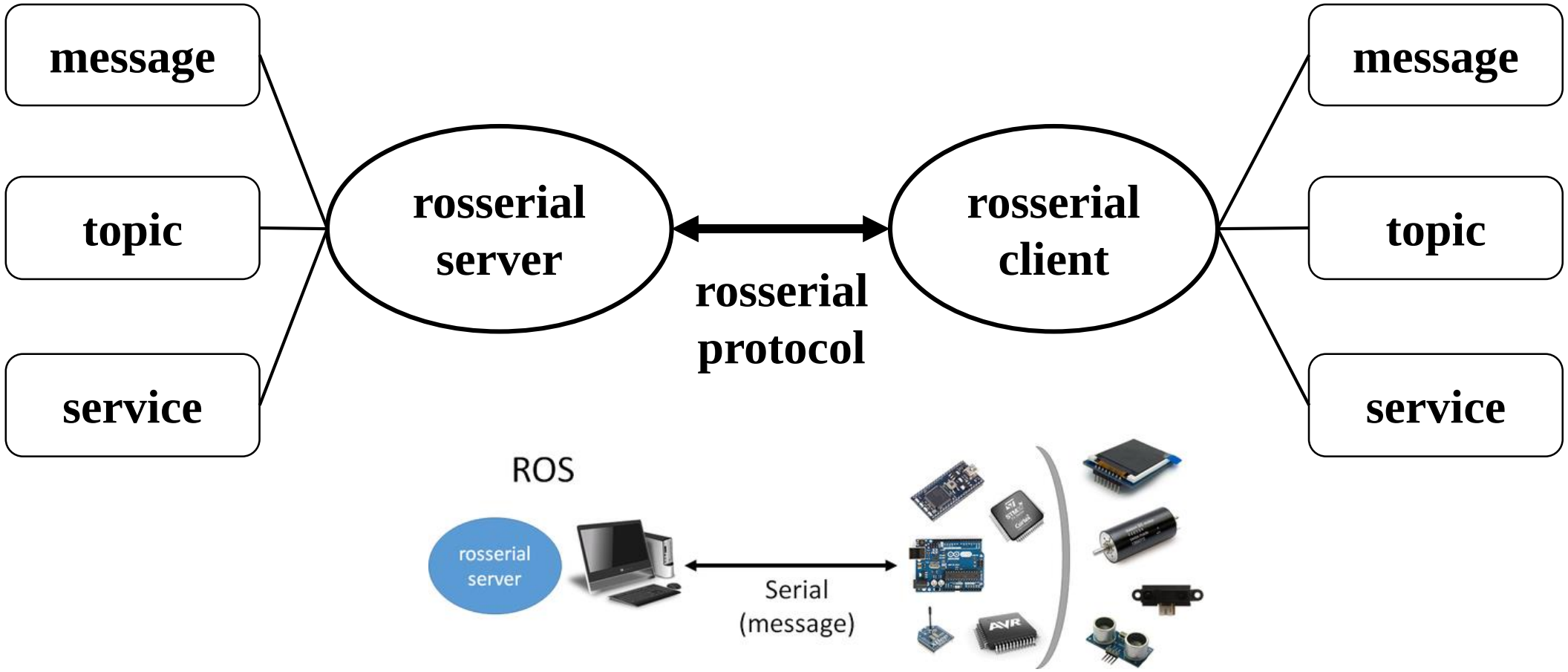
rosserial

- PC와 제어기 간의 메시지 통신을 위해 중계자 역할을 수행하는 ROS 패키지
  - 예) 제어기 → 시리얼(rosserial 프로토콜) → PC(ROS 메시지로 재전송)
  - 예) 제어기 ← 시리얼(rosserial 프로토콜) ← PC(ROS 메시지를 시리얼로 변경)



http://wiki.ros.org/rosserial

rosserial server와 client



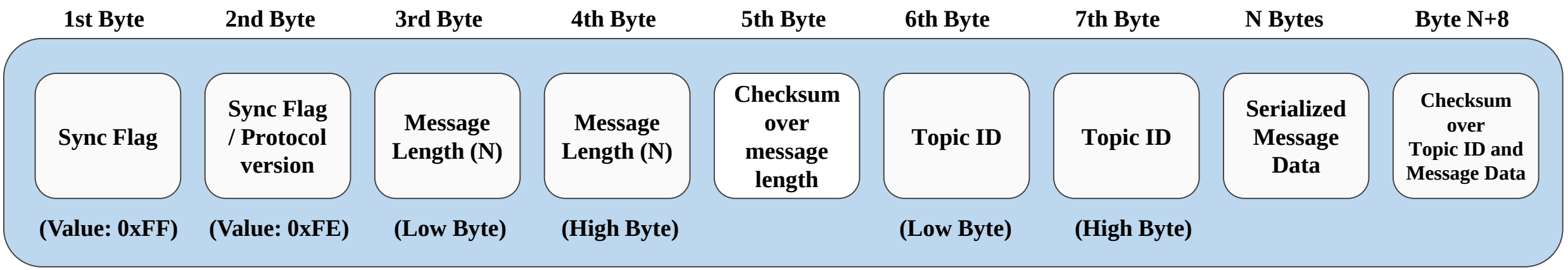
rosserial server와 client

```
$ sudo apt install ros-kinetic-rosserial ros-kinetic-rosserial-server ros-kinetic-rosserial-arduino
```

- **rosserial server**
  - **rosserial_python**: Python 언어 기반의 rosserial server, 매우 많이 사용됨
  - **rosserial_server**: C++ 언어 기반의 rosserial server, 일부 기능 제약 있음
  - **rosserial_java**: Java 언어 기반의 rosserial server, 안드로이드 SDK와 함께 사용

- **rosserial client**
  - **rosserial_arduino**: Arduino와 Leonardo 지원, OpenCR은 수정하여 사용 중
  - **rosserial_embeddedlinux**: 임베디디용 리눅스용 라이브러리
  - **rosserial_windows**: 윈도우 운영체제 지원, 윈도우의 응용프로그램과 통신 지원
  - **rosserial_mbed**: ARM사의 mbed 지원
  - **rosserial_tivac**: TI사의 Launchpad 지원

rosserial 프로토콜 (http://wiki.ros.org/rosserial/Overview/Protocol)



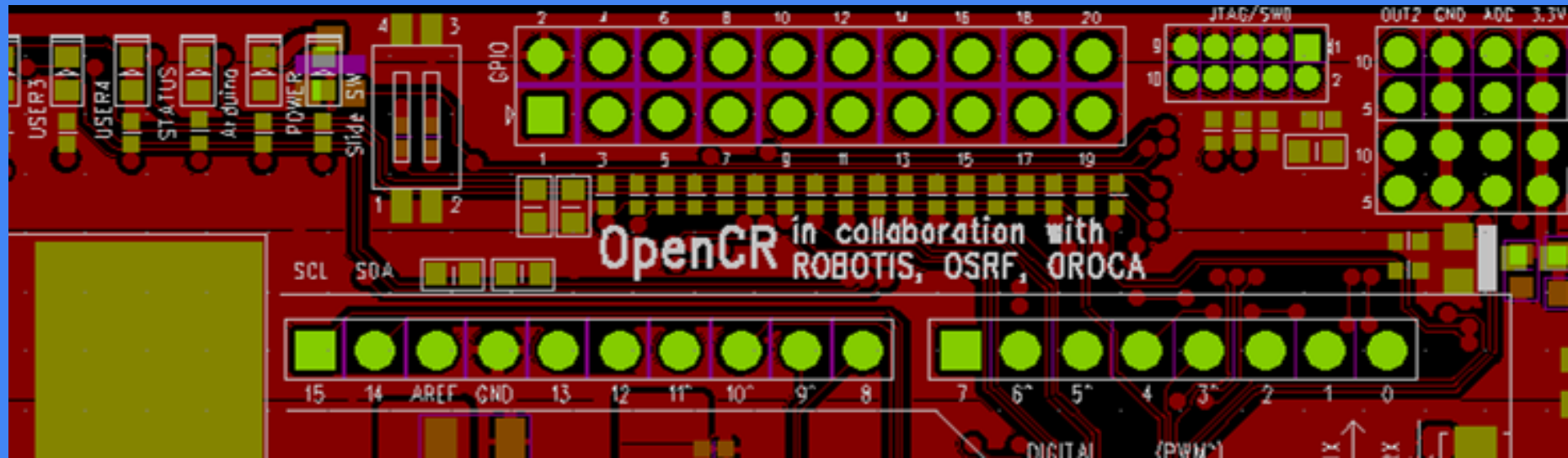
Sync Flag	패킷의 시작 위치를 알기 위한 헤더로 항상 0xFF 이다.
Sync Flag / Protocol version	프로토콜 버전으로 ROS Groovy는 0xFF이고, ROS Hydro, Indigo, Jade 그리고 Kinetic은 0xFE 이다.
Message Length (N)	메시지의 데이터 길이, 2 Byte = Low Byte + High Byte, Low 바이트가 먼저 전송되고 이어서 High 바이트가 전송된다.
Checksum over message length	메시지 길이 헤더의 유효성 검증을 위한 체크섬 Checksum = 255 - ((Message Length Low Byte + Message Length High Byte) %256)
Topic ID	메시지의 형태를 구분하기 위한 ID이다. 2 Byte = Low Byte + High Byte ID_PUBLISHER=0, ID_SUBSCRIBER=1, ID_SERVICE_SERVER=2, ID_SERVICE_CLIENT=4, ID_PARAMETER_REQUEST=6, ID_LOG=7, ID_TIME=10, ID_TX_STOP=11
Serialized Message Data	송/수신 메시지를 시리얼 형태로 전송하기 위한 데이터이다. 예) IMU, TF, GPIO 데이터
Checksum over Topic ID and Message Data	Topic ID와 메시지 데이터의 유효성 검증을 위한 체크섬 Checksum = 255 - ((Topic ID Low Byte + Topic ID High Byte + data byte values) % 256)

rosserial 제약사항

- **메모리**
  - 퍼블리셔, 서브스크라이버 개수 및 송신, 수신 버퍼의 크기를 미리 정의해야 함
- **Float64**
  - 마이크로컨트롤러는 64비트 실수연산을 지원하지 않아 32비트형으로 변경함
- **Strings**
  - 문자열 데이터를 String 메시지 안에 저장하지 않고 외부에서 정의한 문자열 데이터의 포인터 값만 메시지에 저장함
- **Arrays**
  - 메모리 제약사항으로 배열의 크기를 지정해서 사용
- **통신 속도**
  - UART 같은 경우 115200bps와 같은 속도로는 메시지의 개수가 많아지면 응답 및 처리속도가 느려 짐

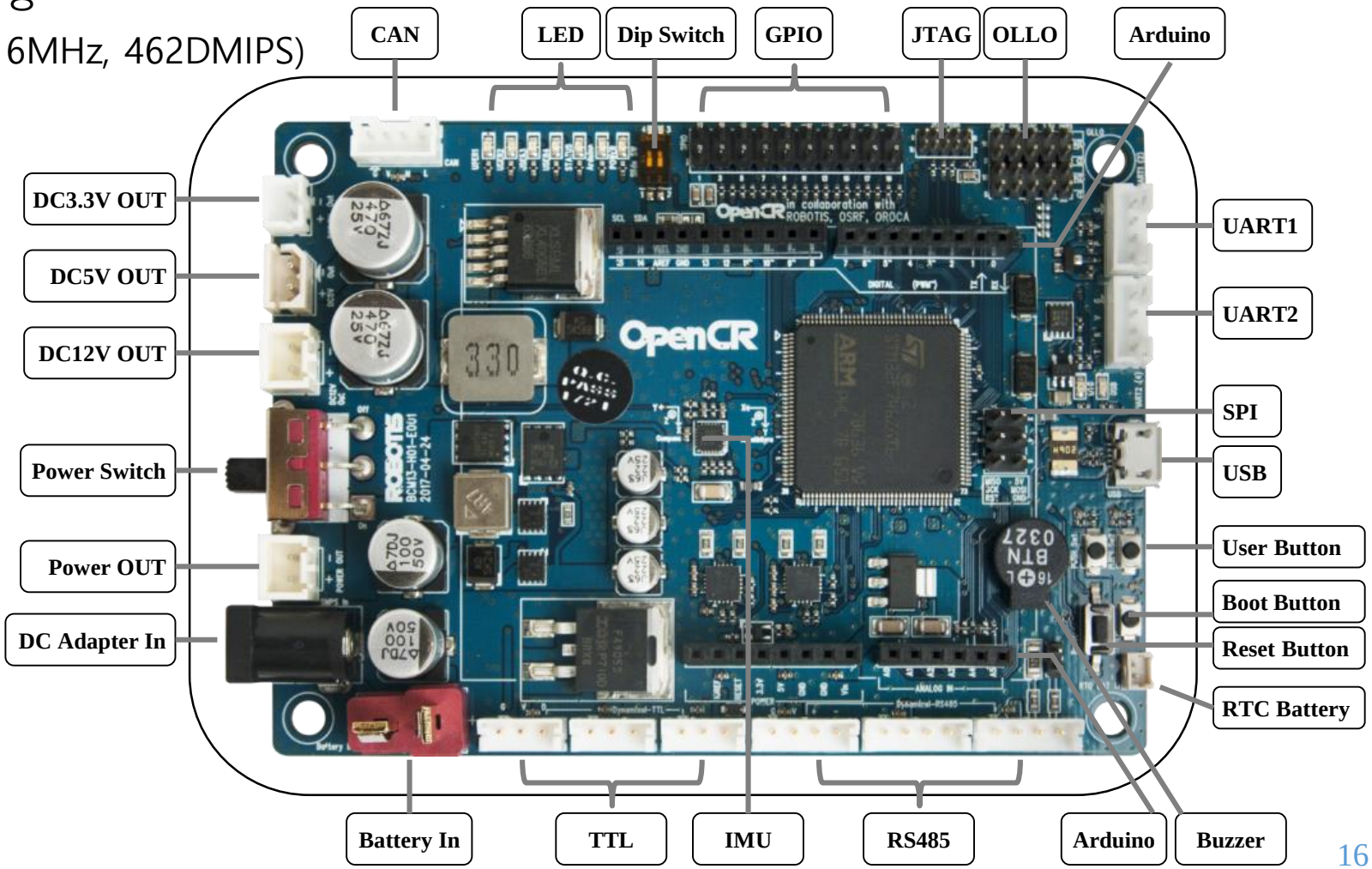
OpenCR; **Open**-source Control module for ROS

OpenCR; **Open**-source Control module for **ROS**



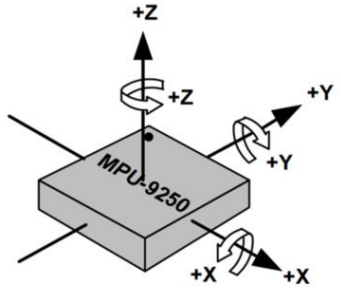
OpenCR (Open-source Control Module for ROS)

- **ROS를 지원하는 임베디드 보드**이며 **터틀봇3**에서 메인 제어기로 사용된다.
- **오픈소스 H/W, S/W** : 회로, BOM, 거버 데이터 등의 H/W 정보 및 OpenCR의 모든 S/W를 오픈소스로 공개
- **rosserial의 제약 사항을 극복**하기 위한 구성
  - 32-bit ARM Cortex-M7 with FPU (216MHz, 462DMIPS)
  - 1MB 플래시 메모리
  - 320KB SRAM
  - Float64 지원
  - UART가 아닌 USB 패킷 전송 사용
- SBC 계열의 컴퓨터와 다양한 센서와 함께 사용하기 위한 **전원 설계**
  - 12V@1A, 5V@4A, 3.3V@800mA
- **확장 포트**
  - 32 pins(L 14, R 18)
    *Arduino connectivity
  - OLLO Sensor module x 4 pins
  - Extension connector x 18 pins

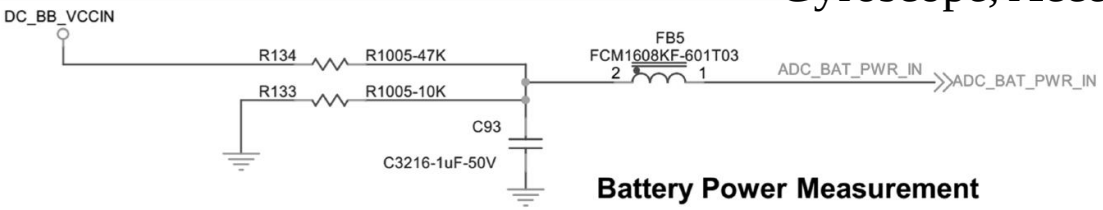


기본 장착 센서 및 통신 지원

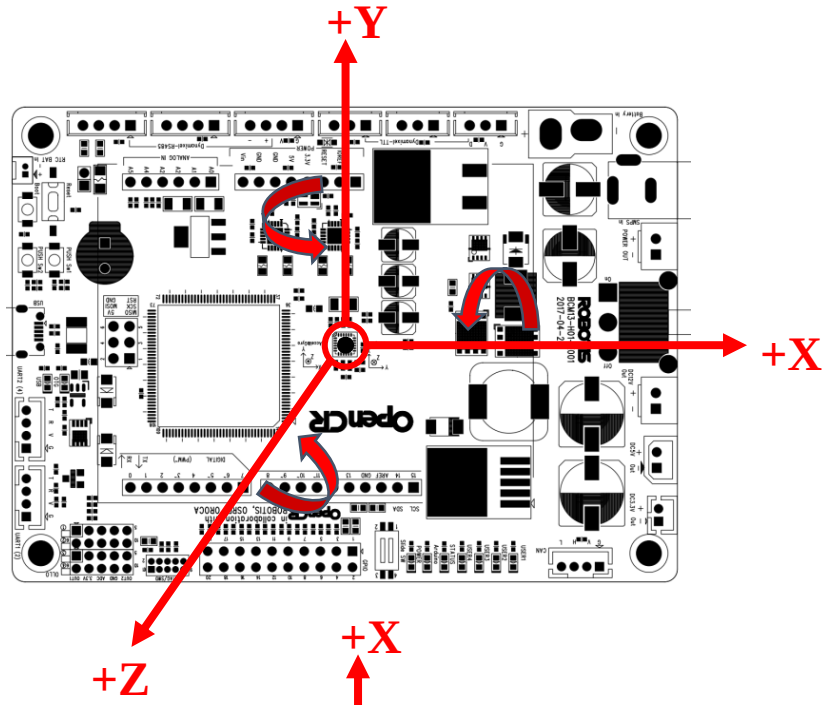
- 기본 장착 센서
  - Gyroscope 3Axis
  - Accelerometer 3Axis
  - Magnetometer 3Axis
  - 전압 측정 회로



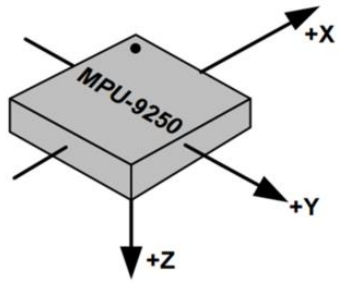
Gyroscope, Accelerometer



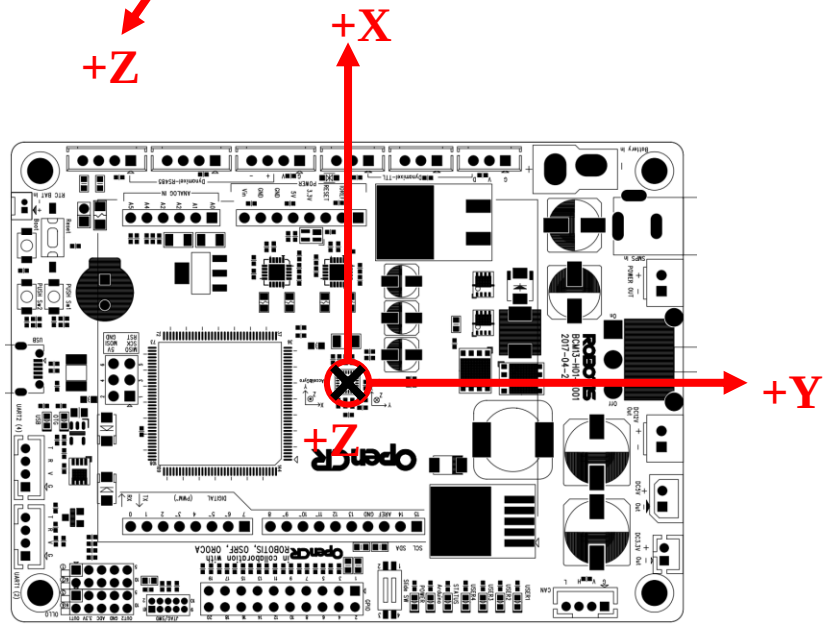
Battery Power Measurement



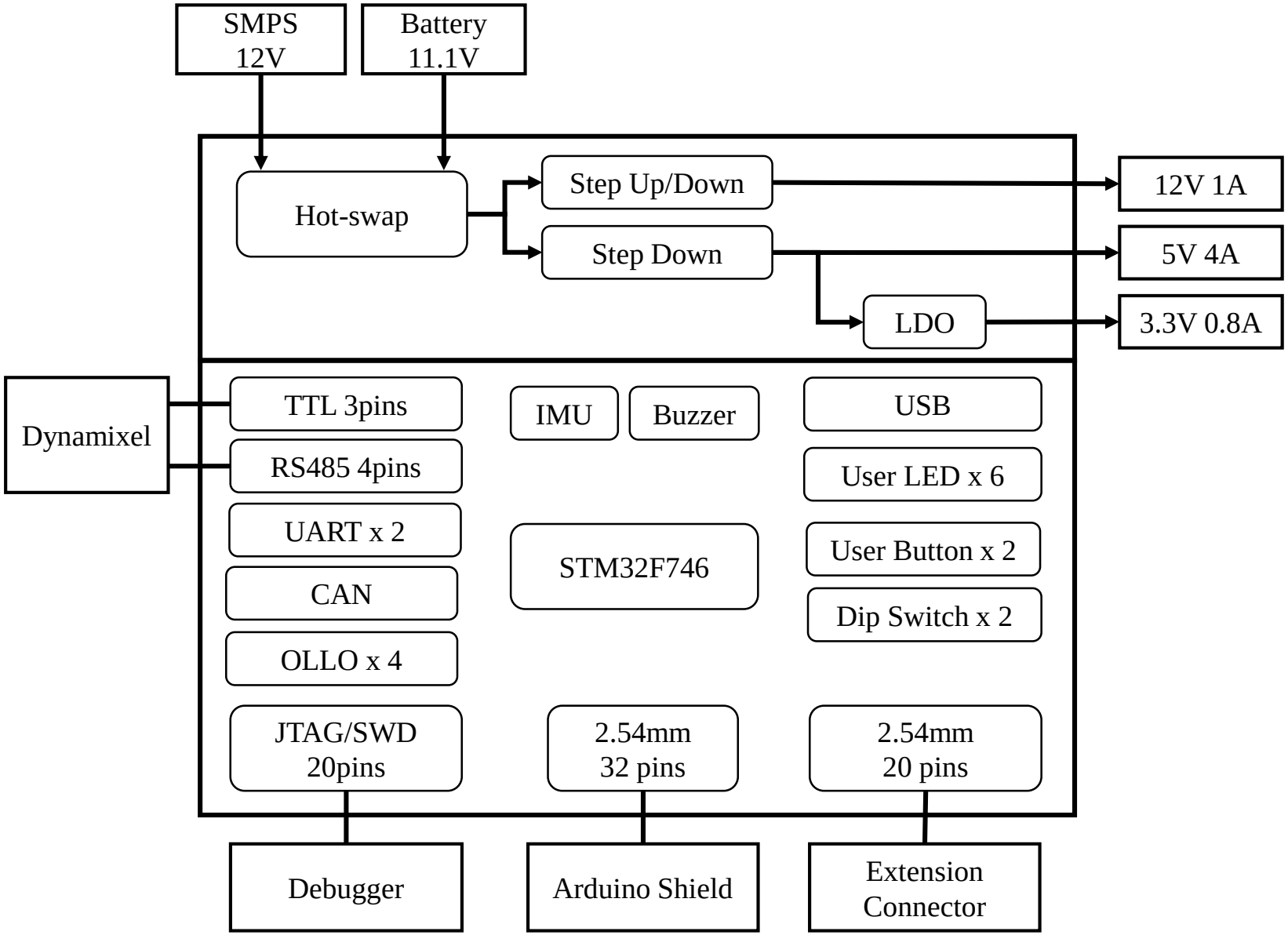
- 통신 지원
  - USB, SPI, I2C
  - TTL, RS485, CAN

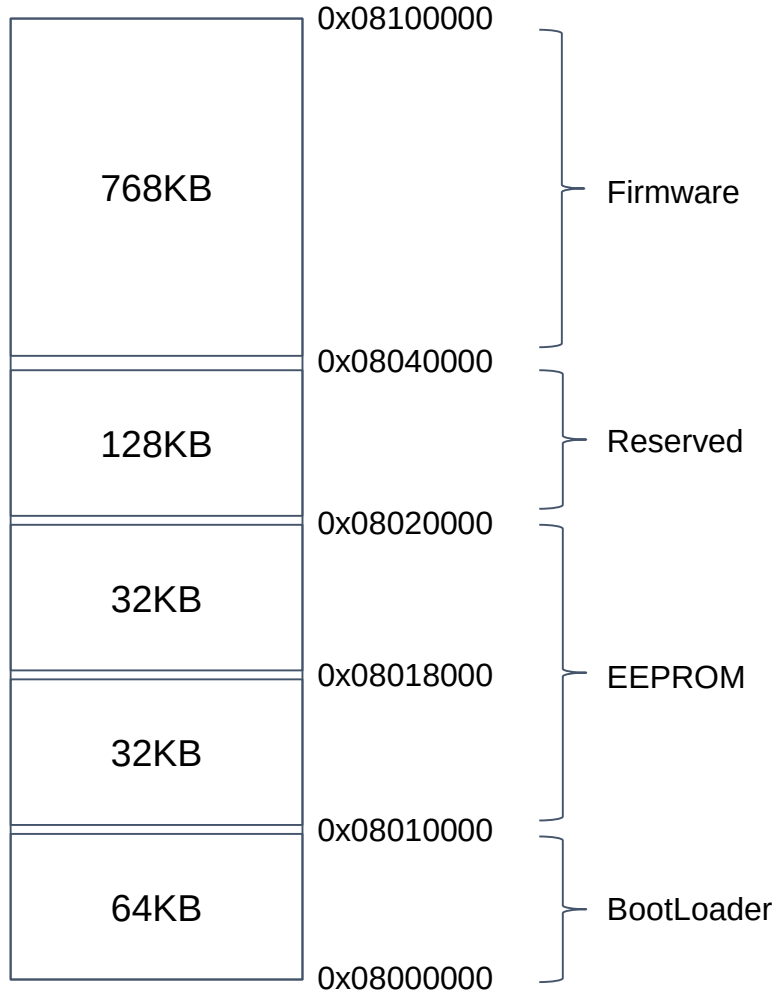


Magnetometer



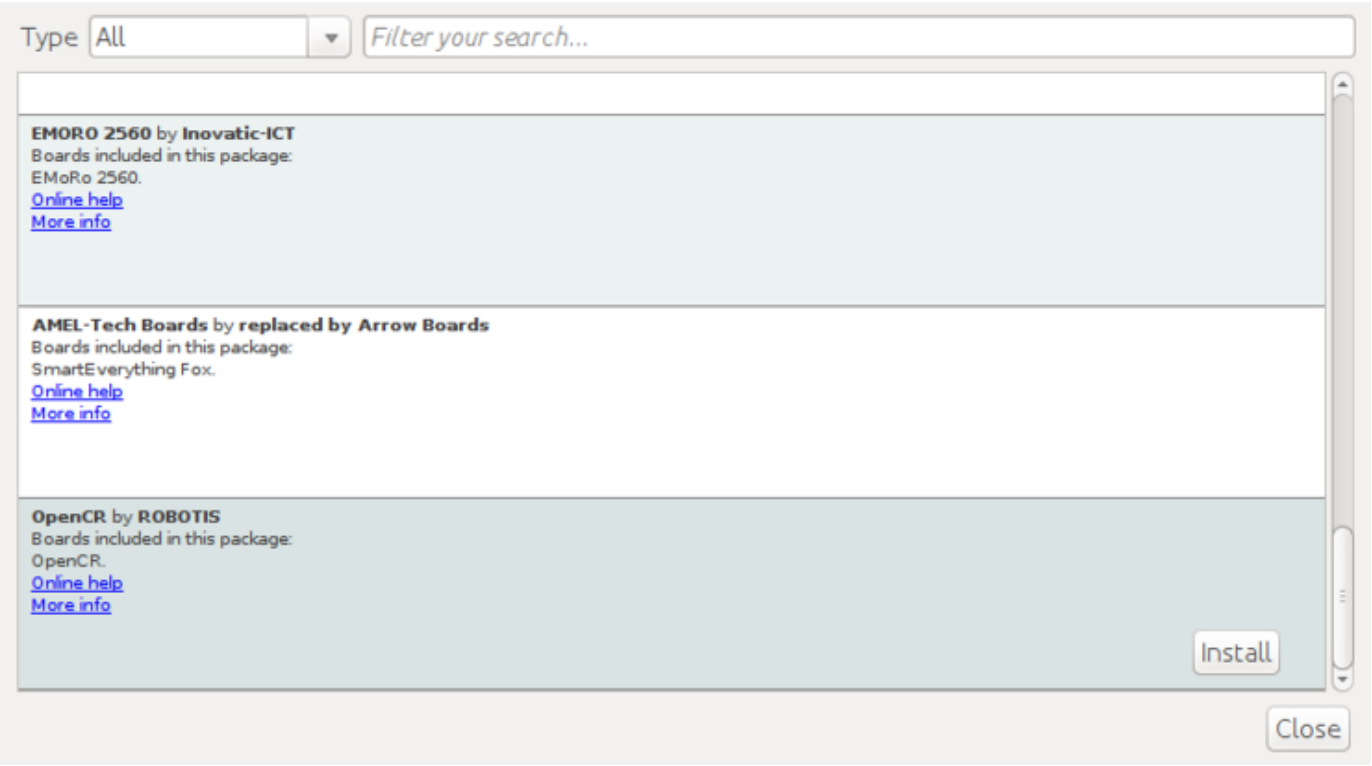
블록 다이어그램 및 플래시 메모리 맵

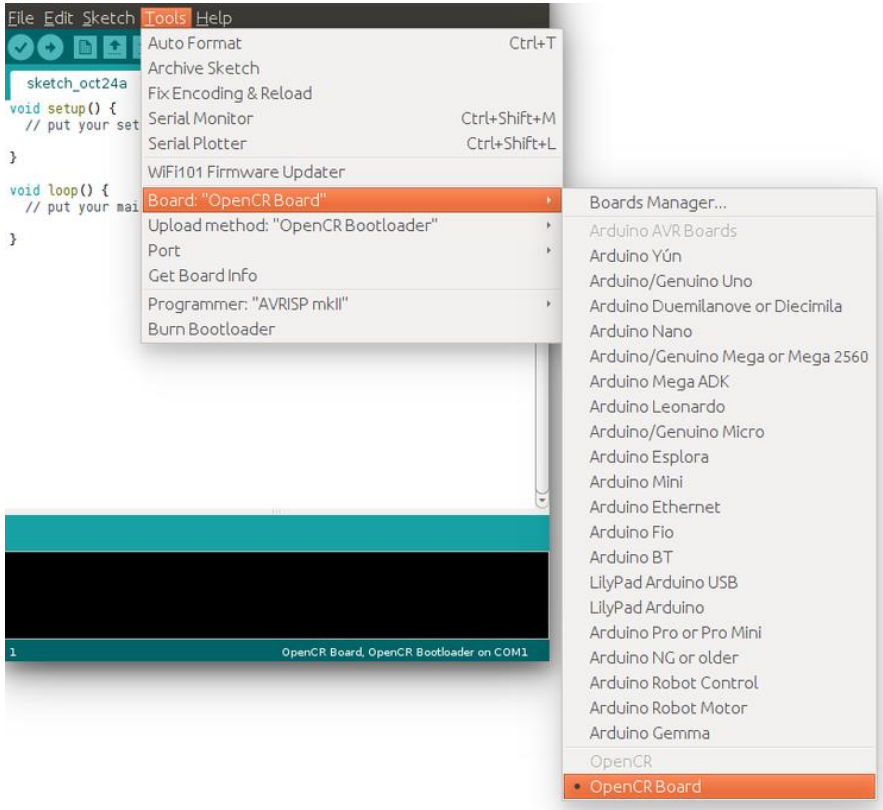




개발환경 구축

- OpenCR은 Arduino IDE를 지원함
- OpenCR 개발환경 구축 방법
  - http://emanual.robotis.com/docs/en/platform/turtlebot3/appendix_opencr1_0/
  - http://emanual.robotis.com/docs/en/parts/controller/opencr10/

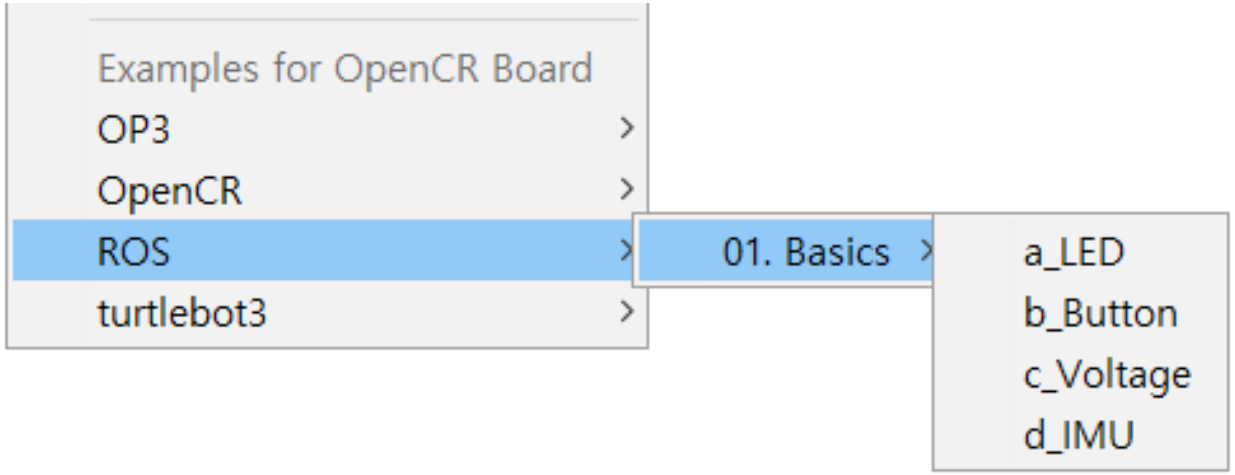




rosserial 예제 (LED 제어)

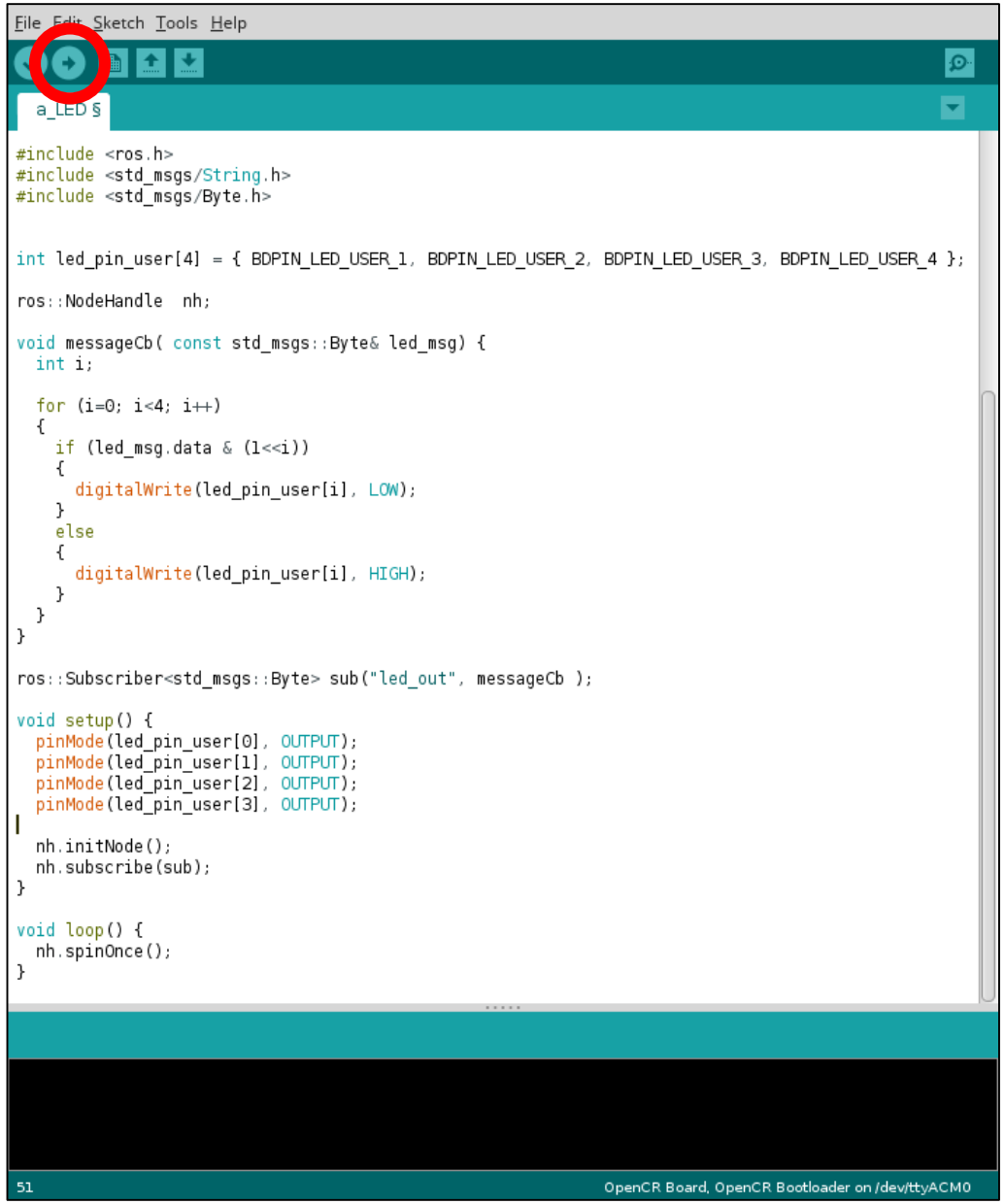
```
$ arduino
```

- Arduino를 실행 후, [File] > [Examples] > [ROS] > [01. Basics] > [a_LED]라는 기본 예제를 불러와 빌드하여 업로드하자.
- 이 예제는 LED 4개를 ROS 표준 데이터 타입인 std_msg/Byte를 이용하여 led_out 서브스크라이버를 정의 한다.
- 서브스크라이버의 콜백 함수가 호출되면 전달된 메시지 값의 0~3번 비트 값에 해당하는 LED 를 비트가 1이면 LED를 켜고 0이면 끈다.



rosserial 예제 (LED 제어)

```
#include <ros.h>
#include <std_msgs/String.h>
#include <std_msgs/Byte.h>

int led_pin_user[4] = { BDPIN_LED_USER_1, BDPIN_LED_USER_2,
BDPIN_LED_USER_3, BDPIN_LED_USER_4 };

ros::NodeHandle  nh;

void messageCb( const std_msgs::Byte& led_msg) {
  int i;

  for (i=0; i<4; i++)
  {
    if (led_msg.data & (1<<i))
    {
      digitalWrite(led_pin_user[i], LOW);
    }
    else
    {
      digitalWrite(led_pin_user[i], HIGH);
    }
  }
}

ros::Subscriber<std_msgs::Byte> sub("led_out", messageCb );

void setup() {
  pinMode(led_pin_user[0], OUTPUT);
  pinMode(led_pin_user[1], OUTPUT);
  pinMode(led_pin_user[2], OUTPUT);
  pinMode(led_pin_user[3], OUTPUT);

  nh.initNode();
  nh.subscribe(sub);
}

void loop() {
  nh.spinOnce();
}
```



rosserial server 실행 및 LED 제어를 위한 토픽 퍼블리시

- roscore를 실행한 후, rosserial sever를 실행시킨다.

```
$ roscore
```

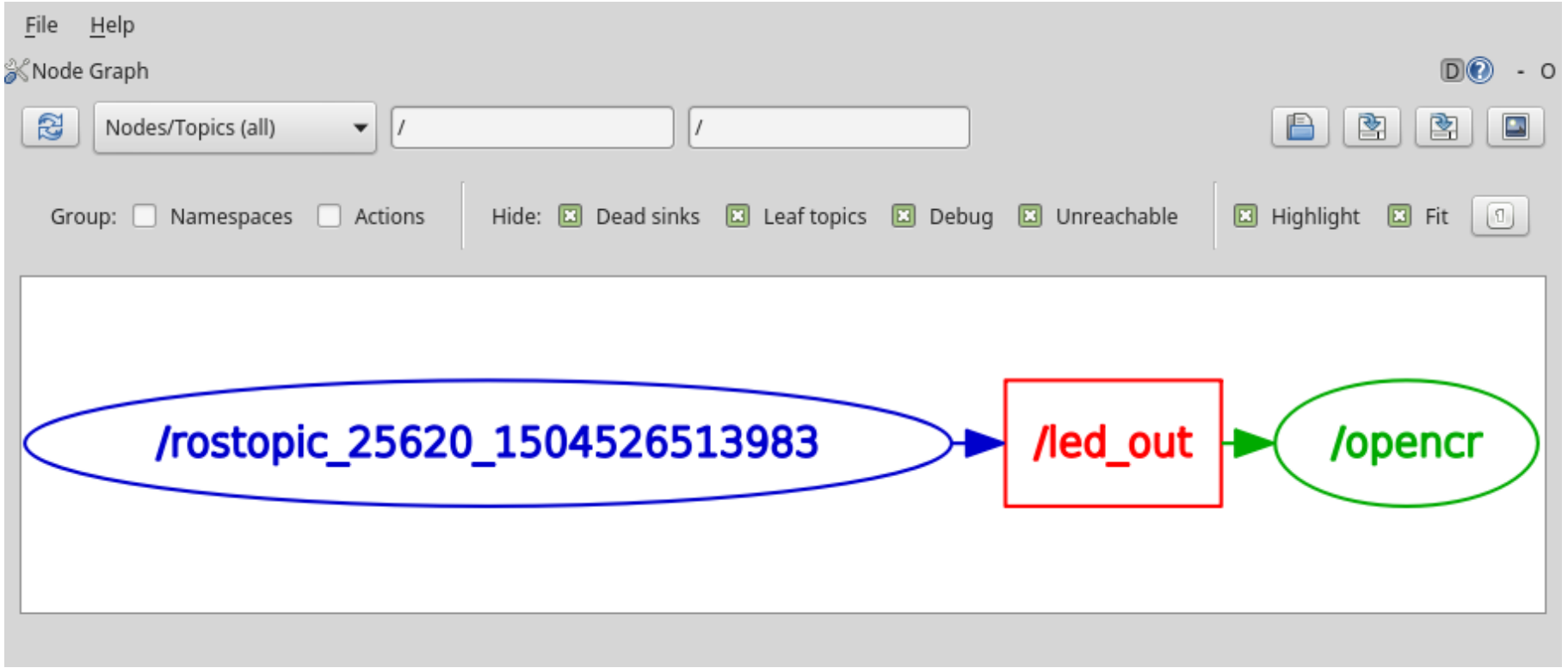
```
$ rosrun rosserial_python serial_node.py __name:=opencr _port:=/dev/ttyACM0 _baud:=115200
[INFO] [1495609829.326019]: ROS Serial Python Node
[INFO] [1495609829.336151]: Connecting to /dev/ttyACM0 at 115200 baud
[INFO] [1495609831.454144]: Note: subscribe buffer size is 1024 bytes
[INFO] [1495609831.454994]: Setup subscriber on led_out [std_msgs/Byte]
```

- rostopic pub을 이용하여 led_out에 값을 입력하여 LED를 제어해보자.

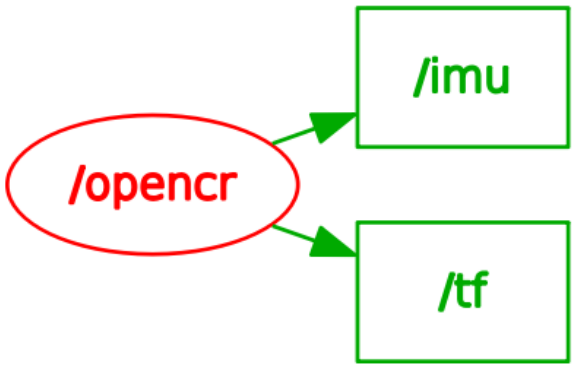
```
$ rostopic pub -1 led_out std_msgs/Byte 1        → USER1 LED On
$ rostopic pub -1 led_out std_msgs/Byte 2        → USER2 LED On
$ rostopic pub -1 led_out std_msgs/Byte 4        → USER3 LED On
$ rostopic pub -1 led_out std_msgs/Byte 8        → USER4 LED On
$ rostopic pub -1 led_out std_msgs/Byte 0        → LED Off
```

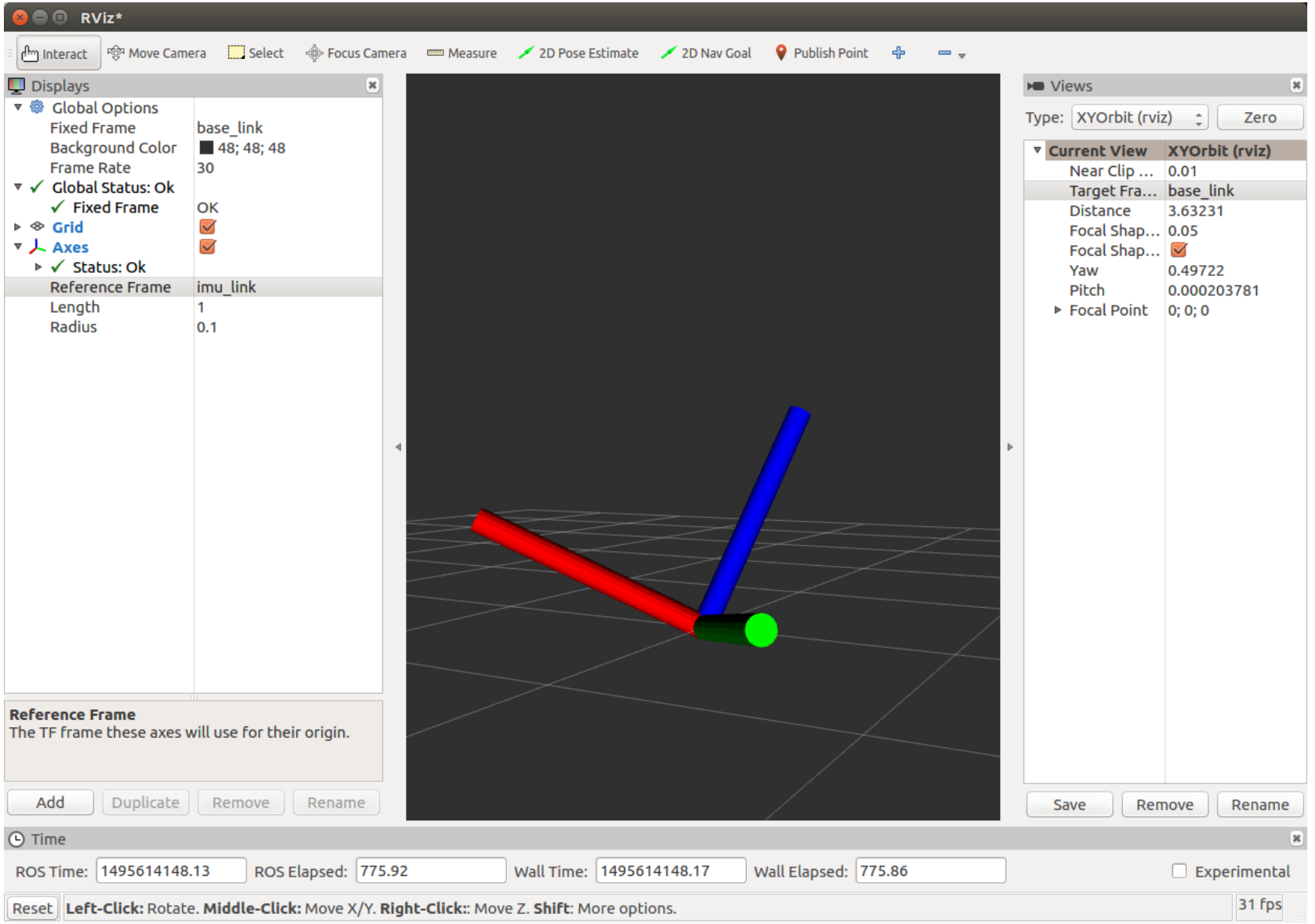
LED 제어를 위한 퍼블리셔 노드와 서브스크라이버 노드

- rqt_graph를 실행해 보자.
- rostopic 명령어가 퍼블리셔 노드로 opencr(rosserial server)가 서브스크라이버로 동작하고 있으며, 두 노드간에 '/led_out' 이라는 토픽명으로 정보가 송수신되고 있는 것을 확인할 수 있다.

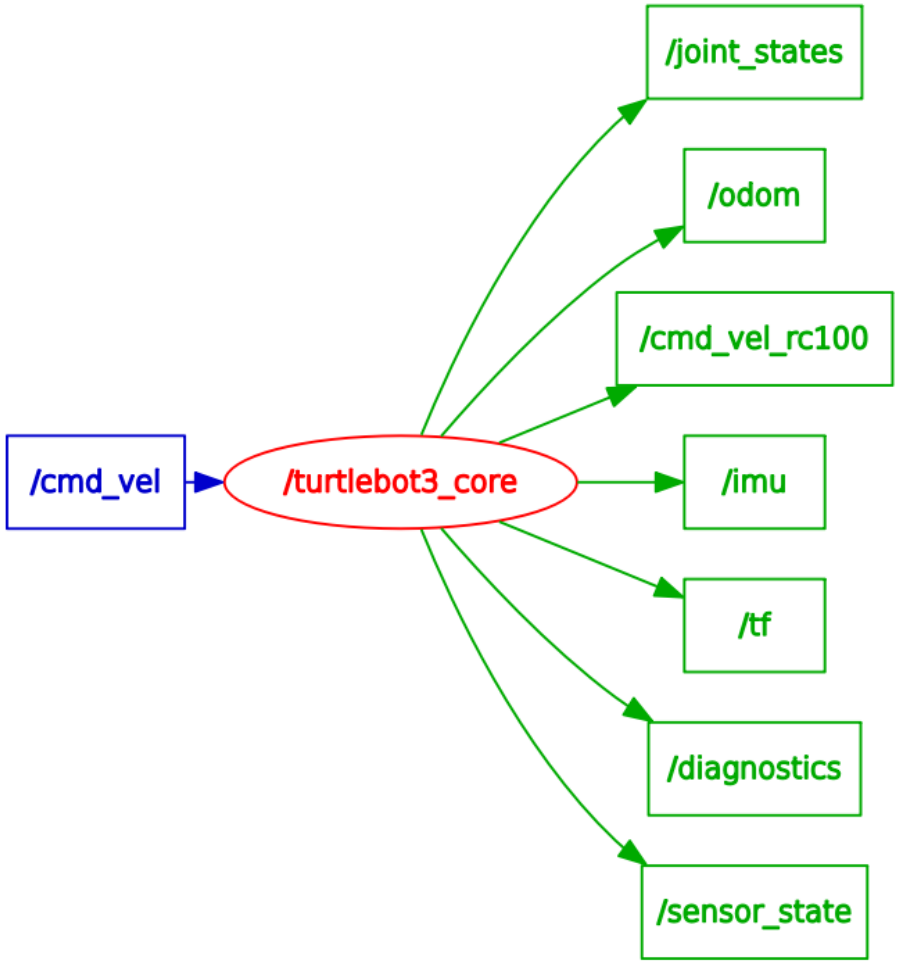


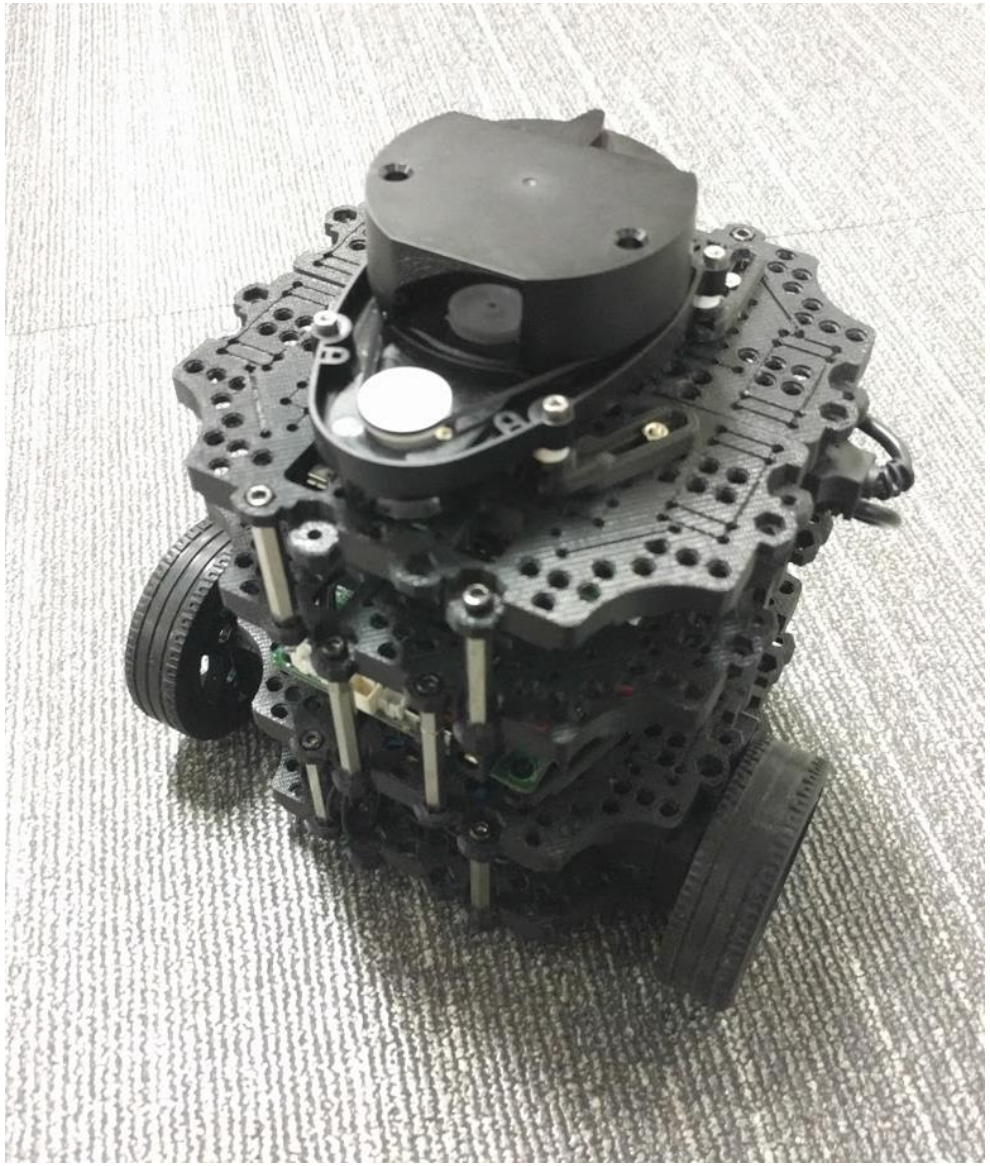
rosserial 예제 (IMU제어)





rosserial 예제 (TurtleBot3 Burger)





질문
대환영!

* 온라인 상의 질문이라면
오르카 및 로열로를 이용해주세요!

여기서! 광고 하나 나가요~

개정 증보판

✔ 한국어판 구매 링크

✔ 4개 언어로 출판!

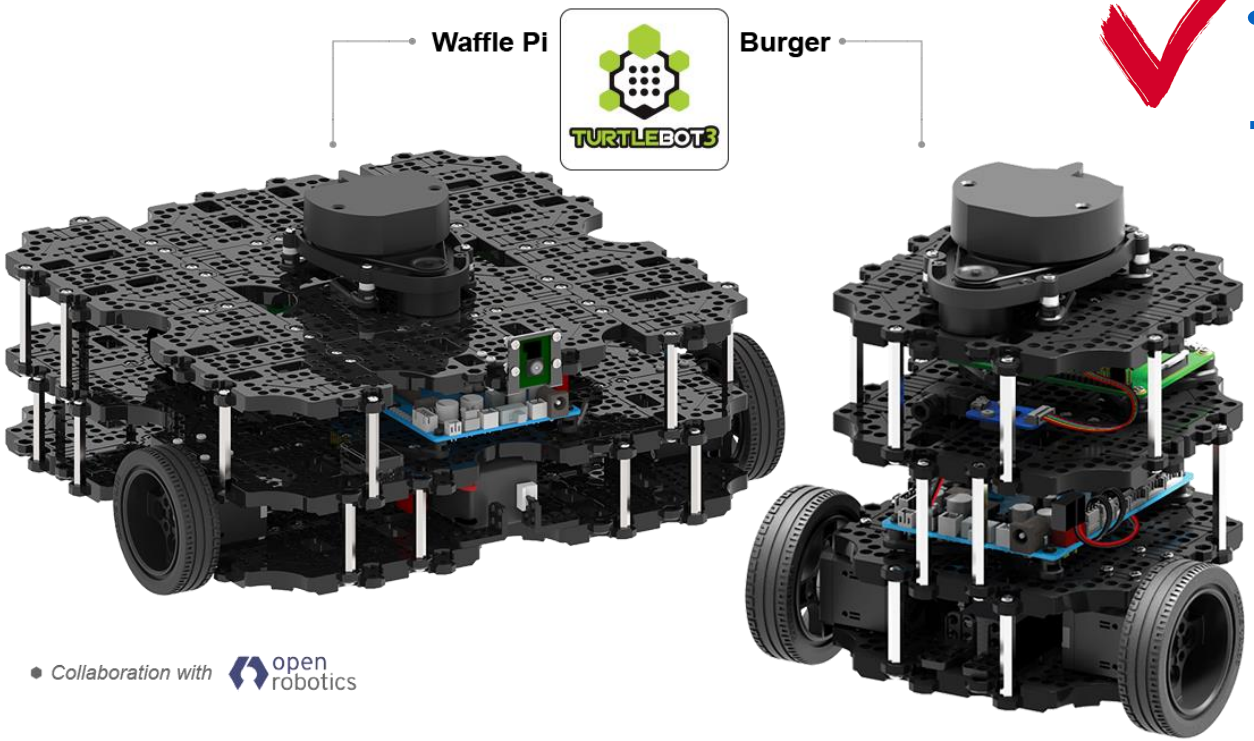
국내 유일! 최초! ROS 참고서!
ROS 공식 플랫폼 **TurtleBot3** 개발팀이
직접 저술한 바이블급 ROS 책

여기서! 광고 둘 나가요~

TURTLEBOT3

인공지능(AI) 연구의 시작,
ROS 교육용 공식 로봇 플랫폼

터틀봇3는 ROS기반의 저가형 모바일 로봇으로 교육, 연구, 제품개발, 취미 등 다양한 분야에서 활용할 수 있습니다.



Direct Link

여기서! 광고 셋 나가요~

오로카

- 오로카
- www.oroca.org
- 오픈 로보틱스 지향
- 풀뿌리 로봇공학의 저변 활성화
- 공개 강좌, 세미나, 프로젝트 진행

- 로봇공학을 위한 열린 모임 (KOS-ROBOT)
- www.facebook.com/groups/KoreanRobotics
- 로봇공학 통합 커뮤니티 지향
- 일반인과 전문가가 어울러지는 한마당
- 로봇공학 정보 공유
- 연구자 간의 협력

Robot Source

- RobotSource
- www.robotsource.org
- 글로벌 로보틱스 커뮤니티 지향
- 로봇공학 정보 공유
- 자신의 로봇 프로젝트 공유
- DIY 로봇 프로젝트 진행

혼자 하기에 답답하시다고요?

커뮤니티에서 함께 해요~

끝.

표윤석

Yoonseok Pyo
pyo@robotis.com
www.robotpilot.net

www.facebook.com/yoonseok.pyo