

1. OpenCM9.04 IDE(S/W) 사용법

① ROBOTIS OpenCM 다운로드

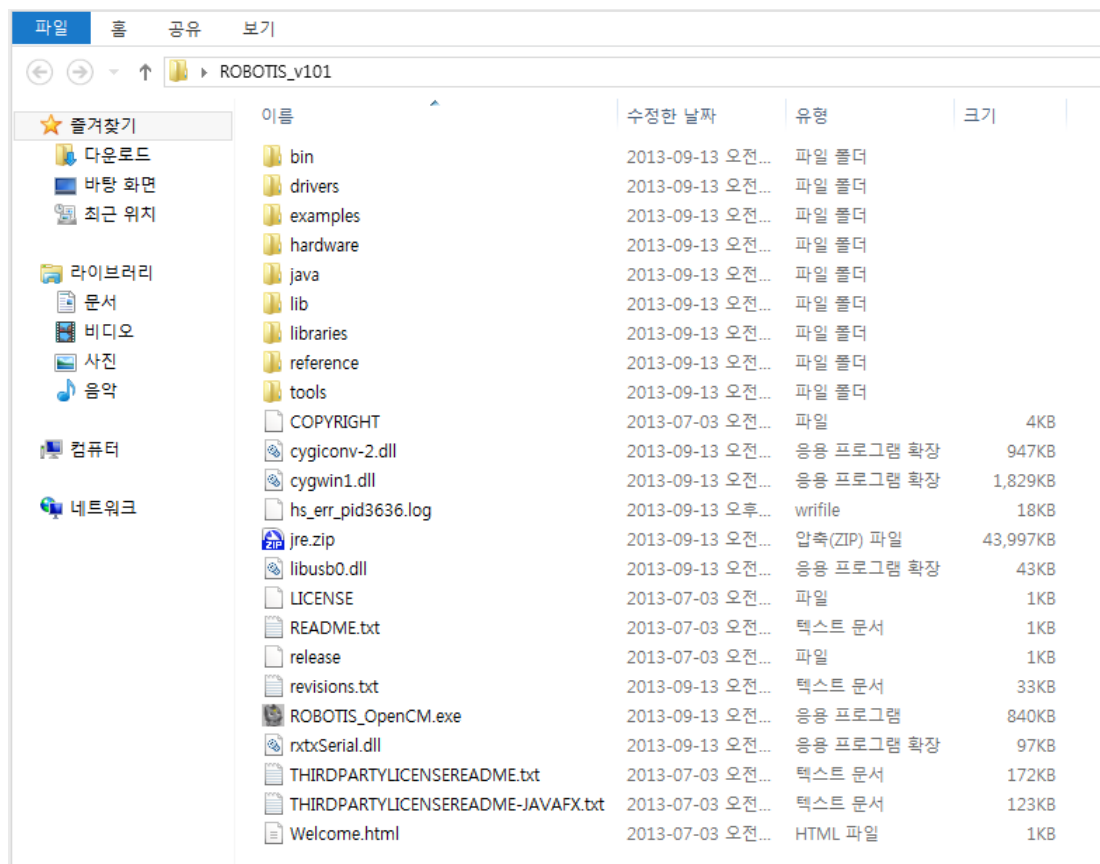
OpenCM9.04를 프로그래밍하기 위해서는 ROBOTIS OpenCM 프로그램이 필요하고 OpenCM은 로보티즈 E-manual를 통해서도 다운로드 받을 수 있습니다. OpenCM9.04 시리즈는 로봇소스를 통해 공개하기 때문에 최신 자료를 공유하기 위해서는 되도록

로봇소스를 통해서 받습니다.

<http://support.robotis.com>

② ROBOTIS OpenCM 구조

다운로드 받은 압축 파일을 풀면 아래와 같은 디렉토리 구조를 확인합니다..



이름	수정된 날짜	유형	크기
bin	2013-09-13 오전...	파일 폴더	
drivers	2013-09-13 오전...	파일 폴더	
examples	2013-09-13 오전...	파일 폴더	
hardware	2013-09-13 오전...	파일 폴더	
java	2013-09-13 오전...	파일 폴더	
lib	2013-09-13 오전...	파일 폴더	
libraries	2013-09-13 오전...	파일 폴더	
reference	2013-09-13 오전...	파일 폴더	
tools	2013-09-13 오전...	파일 폴더	
COPYRIGHT	2013-07-03 오전...	파일	4KB
cygconv-2.dll	2013-09-13 오전...	응용 프로그램 확장	947KB
cygwin1.dll	2013-09-13 오전...	응용 프로그램 확장	1,829KB
hs_err_pid3636.log	2013-09-13 오후...	wrfile	18KB
jre.zip	2013-09-13 오전...	압축(ZIP) 파일	43,997KB
libusb0.dll	2013-09-13 오전...	응용 프로그램 확장	43KB
LICENSE	2013-07-03 오전...	파일	1KB
README.txt	2013-07-03 오전...	텍스트 문서	1KB
release	2013-07-03 오전...	파일	1KB
revisions.txt	2013-09-13 오전...	텍스트 문서	33KB
ROBOTIS_OpenCM.exe	2013-09-13 오전...	응용 프로그램	840KB
rxTxSerial.dll	2013-09-13 오전...	응용 프로그램 확장	97KB
THIRDPARTYLICENSEREADME.txt	2013-07-03 오전...	텍스트 문서	172KB
THIRDPARTYLICENSEREADME-JAVAFX.txt	2013-07-03 오전...	텍스트 문서	123KB
Welcome.html	2013-07-03 오전...	HTML 파일	1KB

A. Drivers : Windows용 USB 드라이버(Inf)가 들어 있습니다.

ROBOTIS_v101 > drivers				
이름	수정된 날짜	유형	크기	
Robotis_cdc_drv.inf	2013-09-13 오전...	설치 정보	2KB	

B. Examples : ROBOTIS OpenCM의 파일메뉴 아래의 예제항목에서 나타나
는 스케치 코드들이 들어 있습니다.

ROBOTIS_v101 > examples				
이름	수정된 날짜	유형		
01. Basics	2013-09-13 오전...	파일 폴더		
02. Digital	2013-09-13 오전...	파일 폴더		
03. Communication	2013-09-13 오전...	파일 폴더		
04. Analog	2013-09-13 오전...	파일 폴더		
05. Dynamixel	2013-09-13 오전...	파일 폴더		
06. Interrupt	2013-09-13 오전...	파일 폴더		

C. Hardware : OpenCM 시리즈의 실제 C/C++ 코드들과 ARM 컴파일러가
포함되어 있습니다.

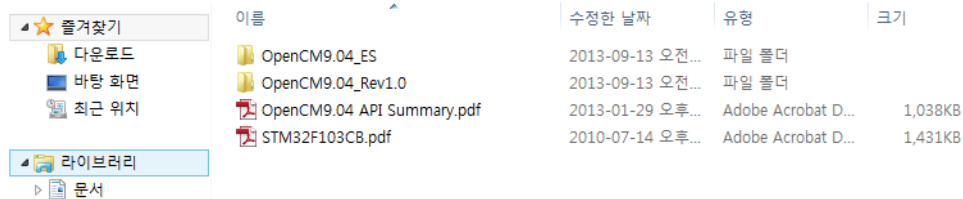
ROBOTIS_v101 > hardware				
이름	수정된 날짜	유형		
robotis	2013-09-13 오전...	파일 폴더		
tools	2013-09-13 오전...	파일 폴더		

robotis폴더에는 OpenCM9.04의 기본 API가 구현된 Core Library 폴더
가 있습니다.

ROBOTIS\hardware\robotis\cores\robotis

ROBOTIS_v101 > hardware > robotis > cores > robotis				
이름	수정된 날짜	유형	크기	
objCacheOpenCM9.04	2013-09-13 오후...	파일 폴더		
OpenCM9.04	2013-09-13 오전...	파일 폴더		
adc.c	2013-09-13 오전...	C 파일	5KB	
adc.h	2013-09-13 오전...	H 파일	14KB	
Arduino-compatibles.cpp	2013-09-13 오전...	CPP 파일	5KB	
Arduino-compatibles.h	2013-09-13 오전...	H 파일	14KB	
bit_constants.h	2013-09-13 오전...	H 파일	13KB	
bitband.h	2013-09-13 오전...	H 파일	5KB	
bits.h	2013-09-13 오전...	H 파일	2KB	
boards.cpp	2013-09-13 오전...	CPP 파일	6KB	
boards.h	2013-09-13 오전...	H 파일	6KB	
CM900.cpp	2013-09-13 오전...	CPP 파일	5KB	
CM900.h	2013-09-13 오전...	H 파일	4KB	
CM904.cpp	2013-09-13 오전...	CPP 파일	4KB	
CM904.h	2013-09-13 오전...	H 파일	3KB	
common.inc	2013-09-13 오전...	INC 파일	7KB	
coxabi-compat.cpp	2013-09-13 오전...	CPP 파일	1KB	
delay.h	2013-09-13 오전...	H 파일	1KB	
descriptors.c	2013-09-13 오전...	C 파일	8KB	
descriptors.h	2013-09-13 오전...	H 파일	8KB	
dxl.c	2013-09-13 오전...	C 파일	9KB	
dxl.h	2013-09-13 오전...	H 파일	3KB	
dxl_constants.h	2013-09-13 오전...	H 파일	6KB	

- D. Java : Windows에서 실행하기 위한 JRE(Java Runtime Environment)가 포함되어 있습니다.
- E. Lib : ROBOTIS OpenCM 리소스 파일이 포함되어 있습니다.
- F. Libraries : 별도의 스케치 라이브러리가 포함되어 있습니다.
- G. Reference : OpenCM 시리즈의 데이터시트 API 문서가 들어 있습니다.



이름	수정된 날짜	유형	크기
OpenCM9.04_ES	2013-09-13 오전...	파일 폴더	
OpenCM9.04_Rev1.0	2013-09-13 오전...	파일 폴더	
OpenCM9.04 API Summary.pdf	2013-01-29 오후...	Adobe Acrobat D...	1,038KB
STM32F103CB.pdf	2010-07-14 오후...	Adobe Acrobat D...	1,431KB

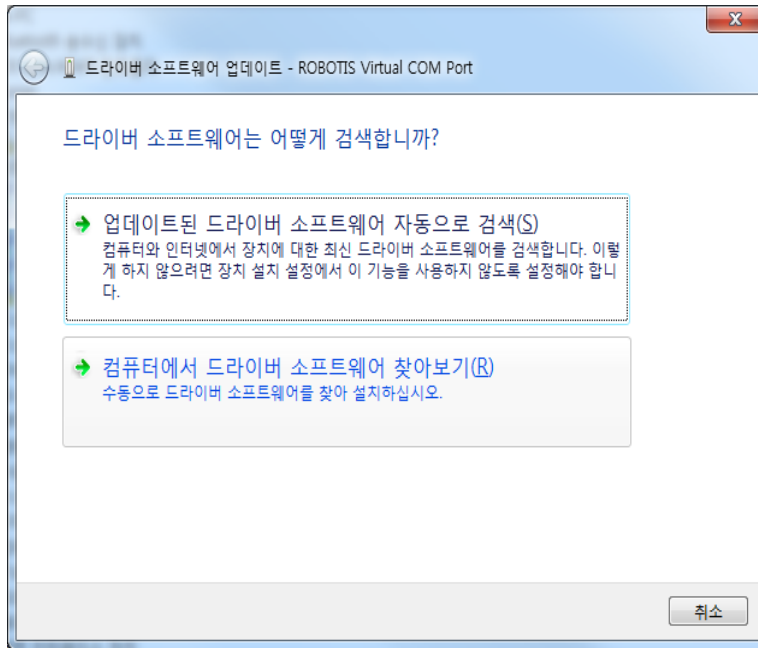
- H. Tools : ROBOTIS OpenCM의 기반인 Processing 관련 툴입니다.
- I. ROBOTIS_OpenCM.exe : ROBOTIS OpenCM 프로그램을 실행하기 위한 파일입니다.

여기까지는 Windows 패키지의 내용입니다. Linux/Mac OS X는 다음에 추가 합니다.

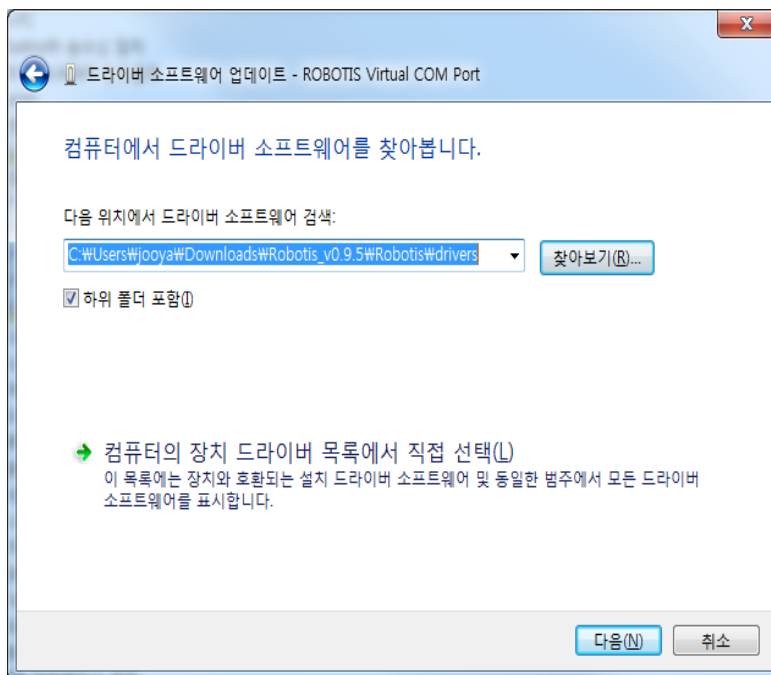
③ USB 드라이버 설치

OpenCM9.04를 ROBOTIS OpenCM을 통해 프로그래밍하기 위해서는 반드시 USB 드라이버가 있어야 합니다. 단, Windows OS 사용자분들만 아래 과정을 진행하고 자동으로 드라이버가 잡히는 Linux와 Mac OS X 이용자분들은 진행하지 않으셔도 됩니다.

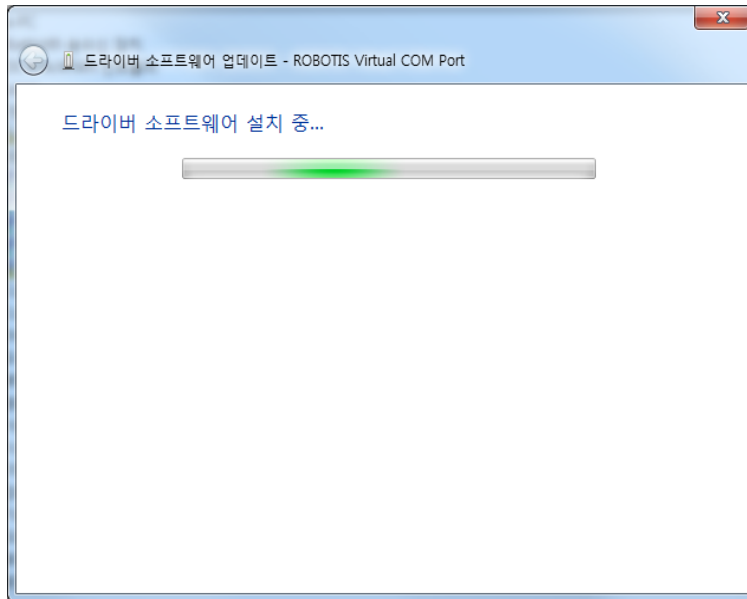
OpenCM9.04을 PC와 연결하면 장치관리자에 ROBOTIS Virtual COM Port 라는 장치가 나타납니다. **“마우스 오른쪽 버튼 -> 드라이버 소프트웨어 업데이트”**를 선택하면 아래와 같은 화면이 나타납니다.



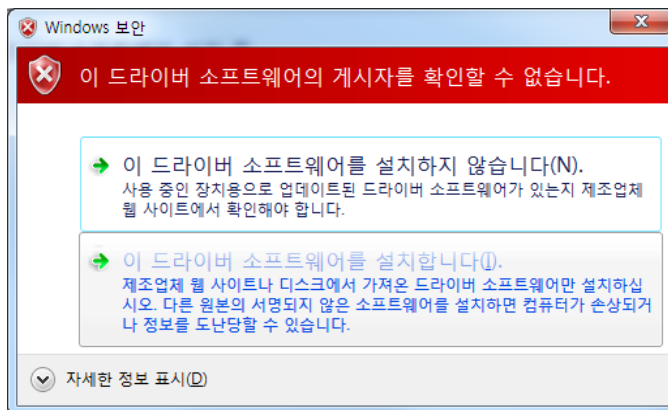
“컴퓨터에서 드라이버 소프트웨어 찾아보기”를 선택합니다.



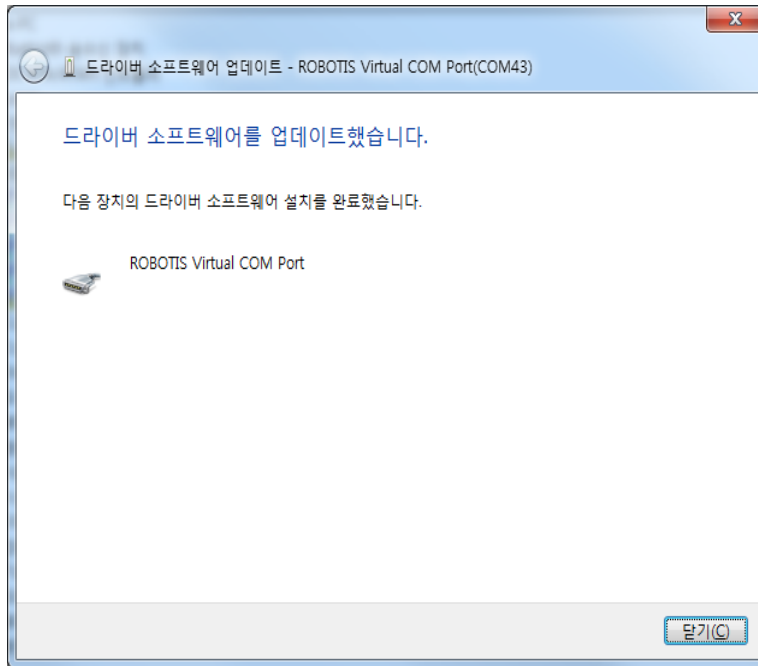
“찾아보기”를 클릭해서 위에서 압축을 해제한 디렉토리(ROBOTIS\drivers)를 선택합니다.



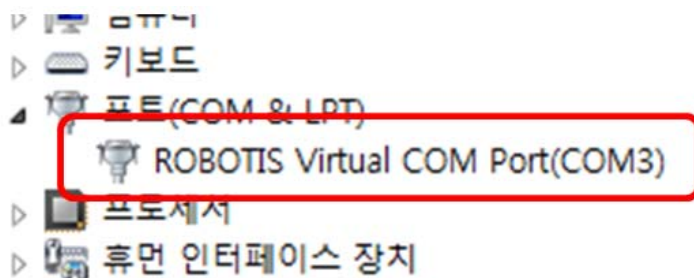
드라이버 소프트웨어 설치를 진행하다가 아래와 같은 보안 경고화면이 나타나면 2번째 항목의 **“이 드라이버 소프트웨어를 설치합니다.”**를 선택합니다.



성공적으로 USB 드라이버를 설치하셨다면 아래와 같이 **“드라이버 소프트웨어를 업데이트했습니다”**라는 창이 나타납니다.



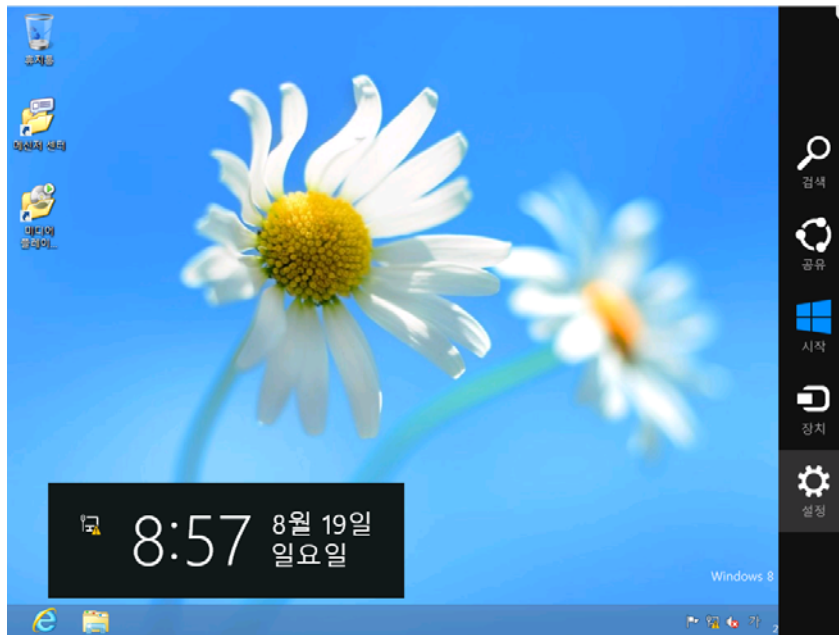
중요한 점은 장치관리자에서 방금 설치하셨던 ROBOTIS Virtual COM Port가 COM 포트 몇 번인지 확인해야 합니다. PC의 다른 USB의 포트에 연결하면 COM포트의 번호가 변경될 수 있으니 주의 하세요



Windows 8 USB 드라이버 설치 문제 해결 방법

윈도우 8의 경우 고급 시작 옵션에서 시작설정의 드라이버 서명적용 안 함으로 재부팅 하신 후 위의 OpenCM9.04 USB 드라이버 설치 작업을 진행하면 됩니다.

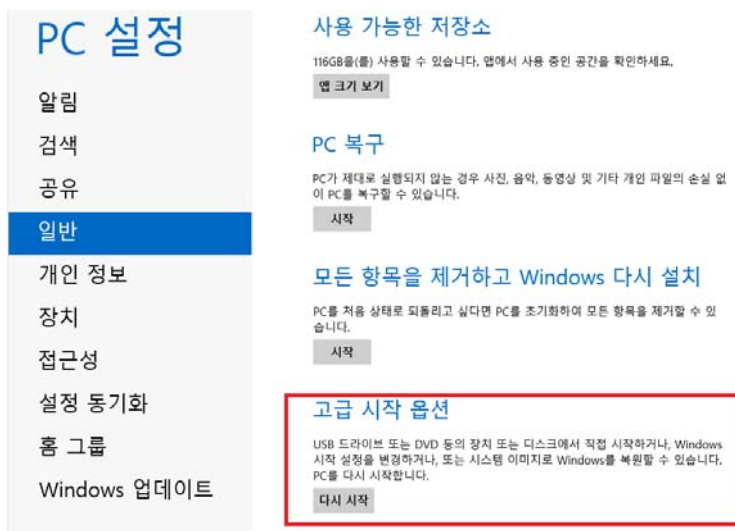
아래의 설정 메뉴 클릭하신 후



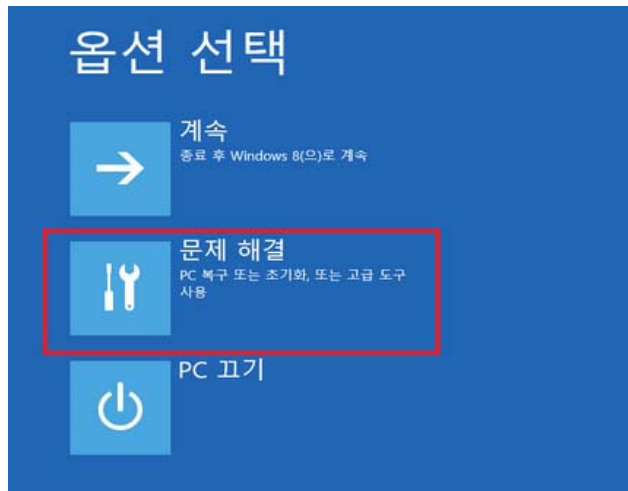
우측 하단의 PC 설정 변경을 클릭합니다.



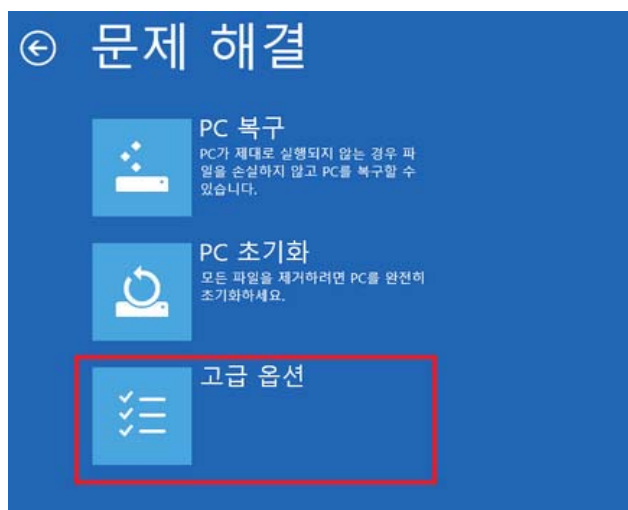
고급 시작 옵션을 선택 합니다.



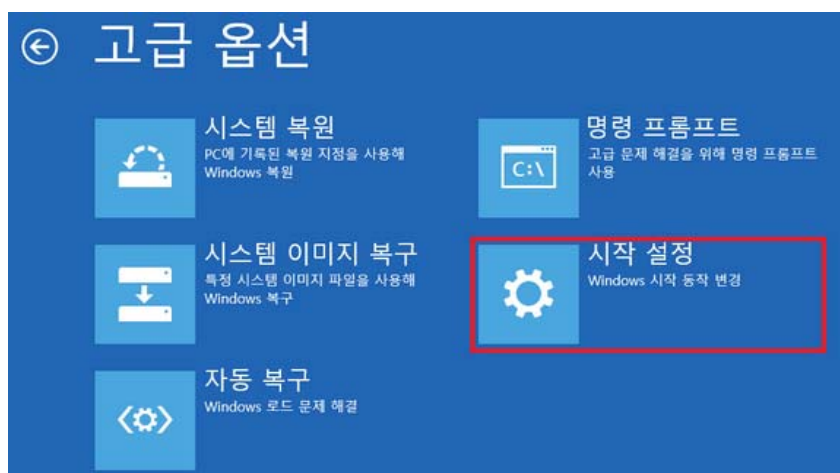
문제 해결 버튼을 선택합니다.



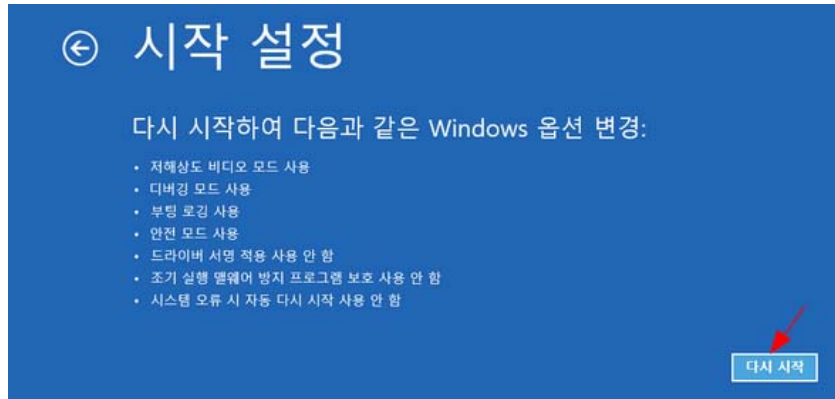
고급 옵션을 선택합니다.



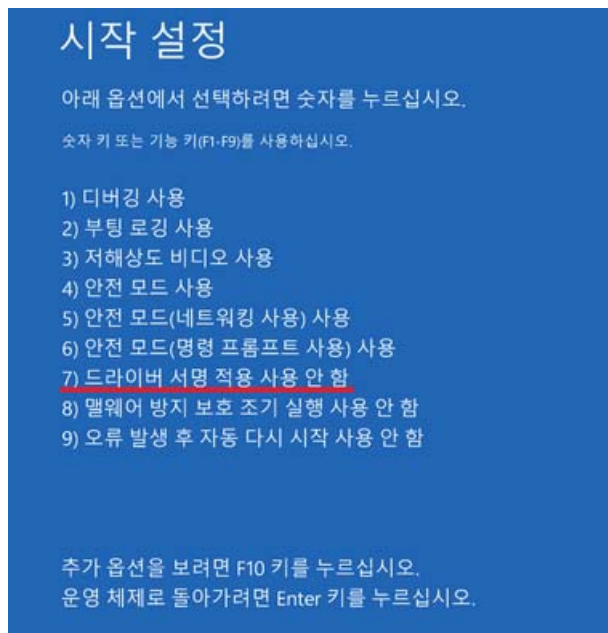
시작 설정을 선택합니다.



우측 하단의 다시 시작을 선택합니다



7번 드라이버 서명 적용 사용 안 함을 선택해서 재부팅 합니다.



재부팅이 완료되면 OpenCM9.04를 PC와 연결하신 후 장치관리자에서 다시 위의 USB 설치과정을 시작합니다.

④ Linux

ROBOTIS OpenCM9 SW Linux release를 다운로드 합니다.

사용하고 계시는 Linux가 32bit라면 Linux 32 bit 패키지를 다운로드 받으시고 64bit이면 Linux 64 bit 패키지를 e-Manual 에서 다운로드 받으시면 됩니다.

<http://www.robotis.com/xe/download/638505>

[Windows XP, Vista, 7, 8]

[ROBOTIS_v0,9,8_win.zip](#)

[Mac OS X] Tested in OS X 10,6,8

[ROBOTIS_v0,9,8_osx.dmg](#)

[Linux 64bit] Tested in Ubuntu 12,04

[ROBOTIS_v0,9,8_linux64.tar](#)

[Linux 32bit] Tested in Ubuntu 10,10

[ROBOTIS_v0,9,8_linux32.tar](#)

[Windows XP, Vista, 7, 8]

[ROBOTIS_v0,9,8_win.zip](#)

[Mac OS X] Tested in OS X 10,6,8

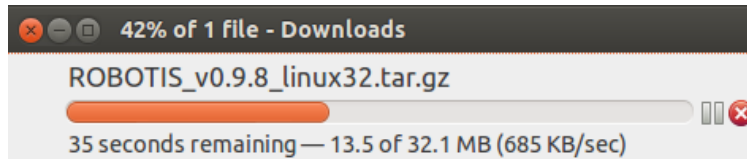
[ROBOTIS_v0,9,8_osx.dmg](#)

[Linux 64bit] Tested in Ubuntu 12,04

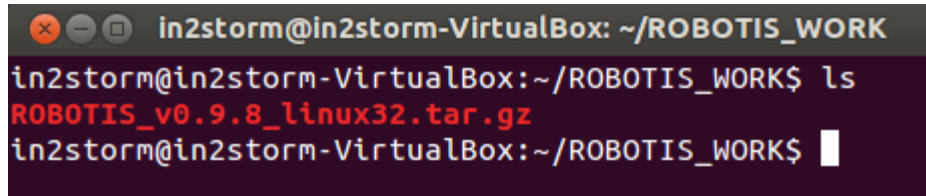
[ROBOTIS_v0,9,8_linux64.tar](#)

[Linux 32bit] Tested in Ubuntu 10,10

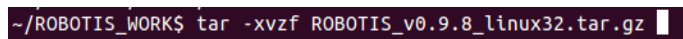
[ROBOTIS_v0,9,8_linux32.tar](#)



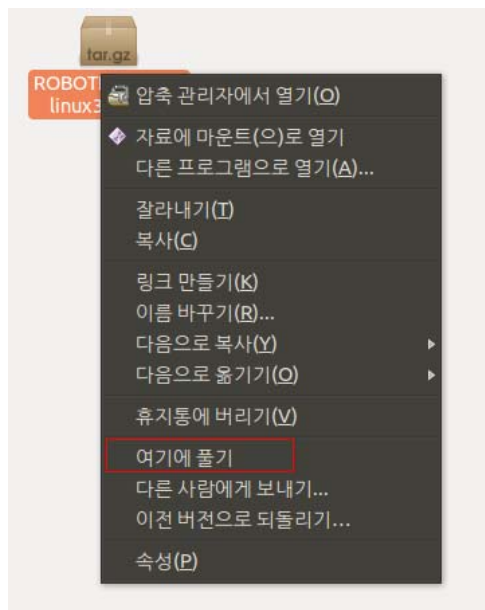
다운로드 받고 난 뒤 아래의 커맨드로 압축을 해제하면 됩니다.



```
~$tar -xvzf ROBOTIS_v0.9.8_linux32.tar.gz
```



아니면 아래와 같이 우분투 UI를 통해 압축해제 합니다.(마우스 오른쪽 버튼)



압축을 풀면 아래와 같이 ROBOTIS 폴더가 생깁니다.



A. JRE가 설치되어 있는지 확인합니다.

설치가 되어 있으면 다음 진행하고 안되어 있으면 apt-get을 통해 JRE 패키지 설치 진행합니다.

설치 확인은 아래와 같이 터미널에서 `java -version`을 입력합니다.

```
in2storm@in2storm-VirtualBox:~/ROBOTIS_WORK$ java -version
'java' 프로그램은 다음 꾸러미들에서 찾을 수 있습니다:
* default-jre
* gcj-4.6-jre-headless
* gcj-4.7-jre-headless
* openjdk-7-jre-headless
* openjdk-6-jre-headless
다음을 실행해 보십시오: sudo apt-get install <선택한 꾸러미>
in2storm@in2storm-VirtualBox:~/ROBOTIS_WORK$
```

위와 같은 응답이 나온다면 JAVA JRE(Java Runtime Environment) 또는 JDK를 설치해야 합니다.

여기서는 `openjdk-7-jre-headless`를 설치해보겠습니다.

`$sudo apt-get install openjdk-7-jre`

```
/ $ sudo apt-get install openjdk-7-jre
```

Y를 눌러서 계속 진행합니다.

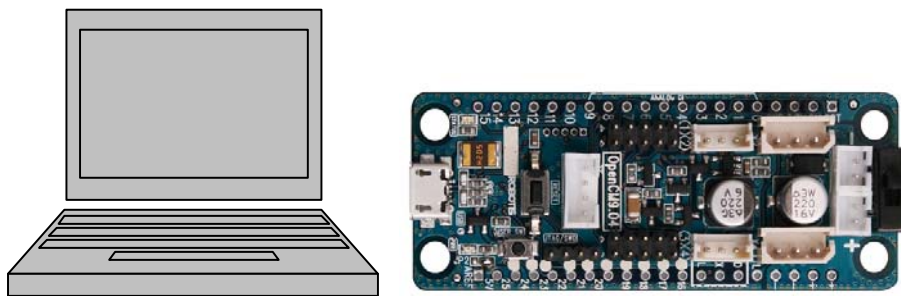
설치가 완료되면 아래와 같이 터미널에 다시 `java -version` 이라는 명령어를 입력합니다.

```
in2storm@in2storm-VirtualBox:~/ROBOTIS_WORK$ java -version
java version "1.7.0_15"
OpenJDK Runtime Environment (IcedTea7 2.3.7) (7u15-2.3.7-0ubuntu1~12.10.1)
OpenJDK Server VM (build 23.7-b01, mixed mode)
in2storm@in2storm-VirtualBox:~/ROBOTIS_WORK$
```

Java JRE가 성공적으로 설치되었으며 이제 ROBOTIS OpenCM을 실행할 수 있습니다.

B. OpenCM-9.04를 PC와 연결합니다.

아래와 같이 OpenCM9.04와 PC를 USB 케이블을 통해 연결합니다.



<OpenCM9.04를 PC와 연결>

단 USB 허브에서 다수의 USB 장치와 함께 연결하는 경우는 되도록 피해주고 **PC와 직접 연결하는 방법을 추천합니다.** 간혹 허브에서 전류가 충분하지 않으면 다운로드 실패하는 경우가 발생합니다. 따라서 반드시 OpenCM9.04에 충분한 전류를 공급할 수 있는 USB 포트에 연결하기 바랍니다.

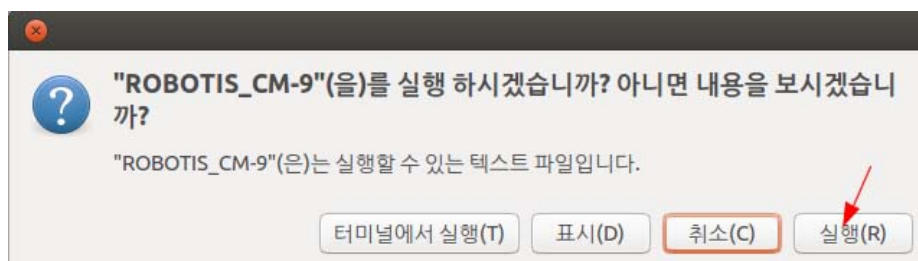
C. ROBOTIS OpenCM을 실행합니다.

아래와 같이 ROBOTIS OpenCM을 더블 클릭하거나

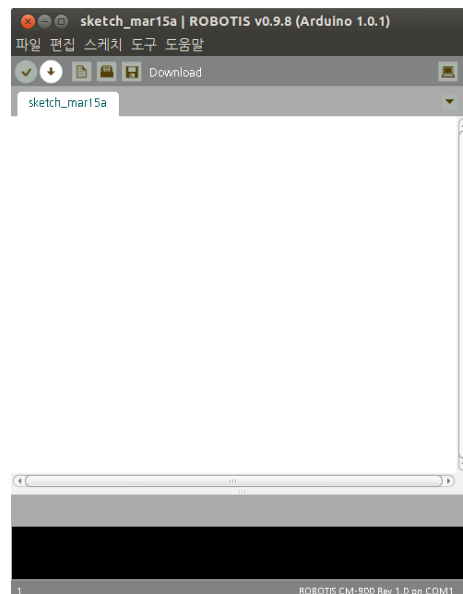
터미널에서 ./ROBOTIS OpenCM을 입력 후 **Enter 치면** 실행됩니다.



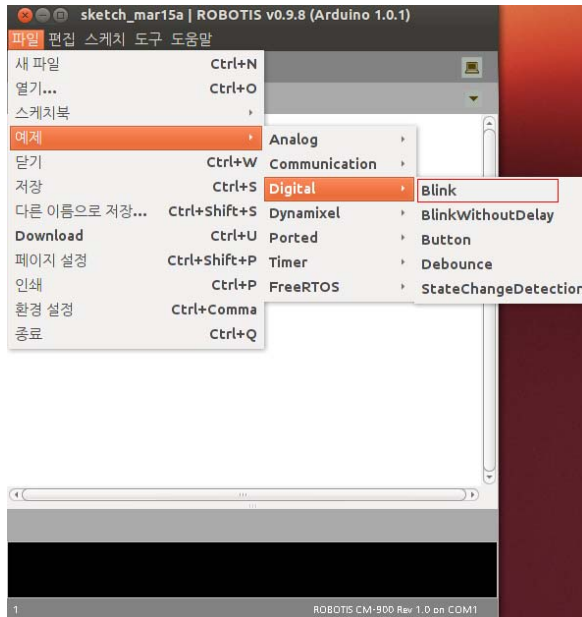
실행파일을 클릭합니다.



아래와 같이 실행됩니다



D. Blink 예제를 Open합니다.



E. 보드를 선택합니다.

ROBOTIS OpenCM9.04를 선택합니다.

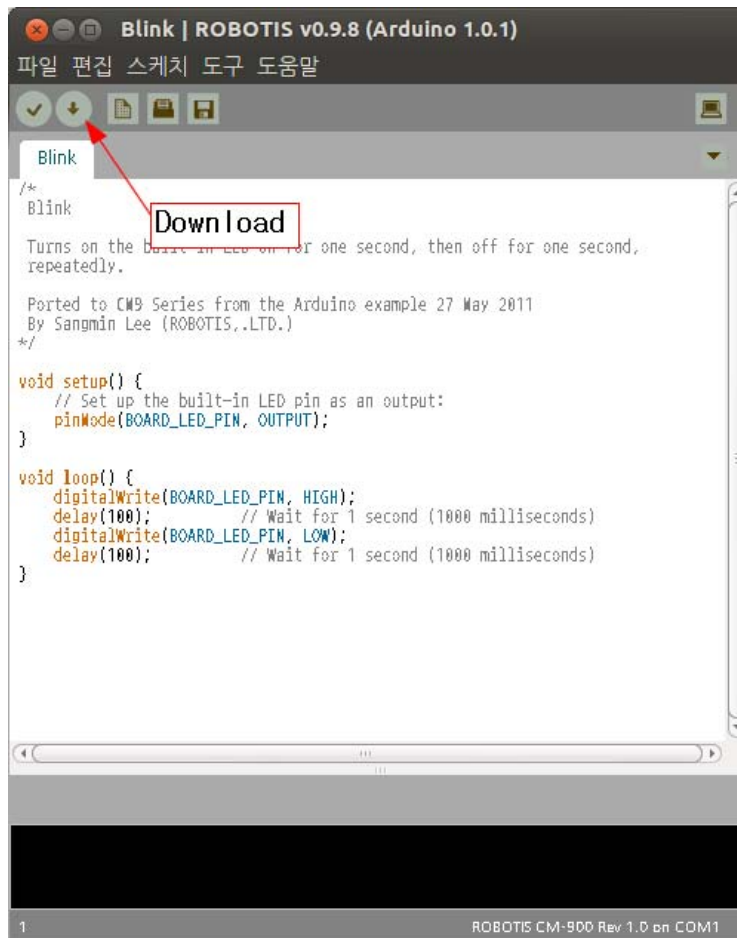


F. 시리얼 포트를 선택합니다.

시리얼 포트는 ttyACMX로 표시되는데 X는 사용자 PC에 따라 다른 숫자로 표시됩니다.



G. 다운로드를 수행합니다.



⑤ Mac OS X

A. ROBOTIS OpenCM SW Mac OS X release를 다운로드 합니다.

e-Manual 에서 Mac OS X용 설치 이미지 파일을 다운로드 받습니다.

<http://www.robotis.com/xe/download/638505>

[Windows XP, Vista, 7, 8]

[ROBOTIS_v0,9,8_win.zip](#)

[Mac OS X] Tested in OS X 10,6,8

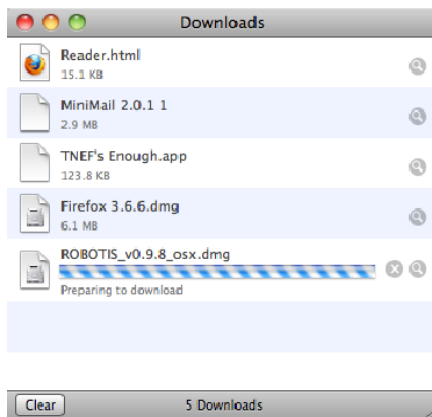
[ROBOTIS_v0,9,8_osx.dmg](#)

[Linux 64bit] Tested in Ubuntu 12,04

[ROBOTIS_v0,9,8_linux64.tar](#)

[Linux 32bit] Tested in Ubuntu 10,10

[ROBOTIS_v0,9,8_linux32.tar](#)



다운로드가 완료되면 아래 dmg파일을 더블클릭해서 Mount를 진행합니다.

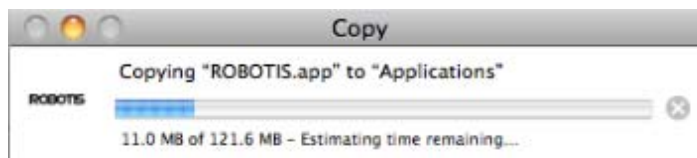


자동실행으로 아래와 같은 창이 뜨면 마우스로 ROBOTIS 아이콘을

Application 쪽으로 드래그 합니다.



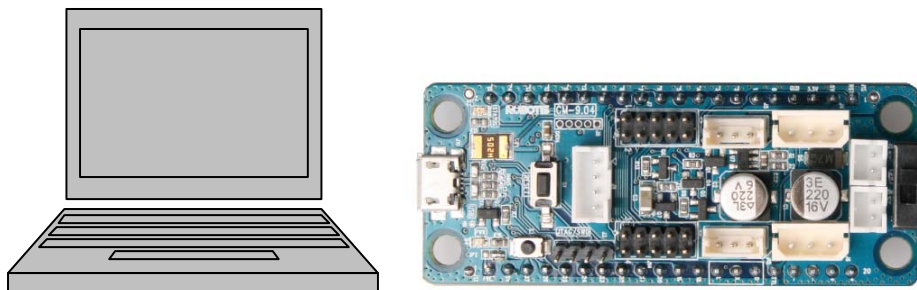
다음과 같이 설치가 진행됩니다.



설치가 끝나면 Application 폴더에 ROBOTIS.app가 생성되고, 이것을 **더블클릭하면** 실행됩니다.

B. OpenCM9.04를 PC와 연결합니다.

아래와 같이 OpenCM9.04와 PC를 USB 케이블을 통해 연결합니다.

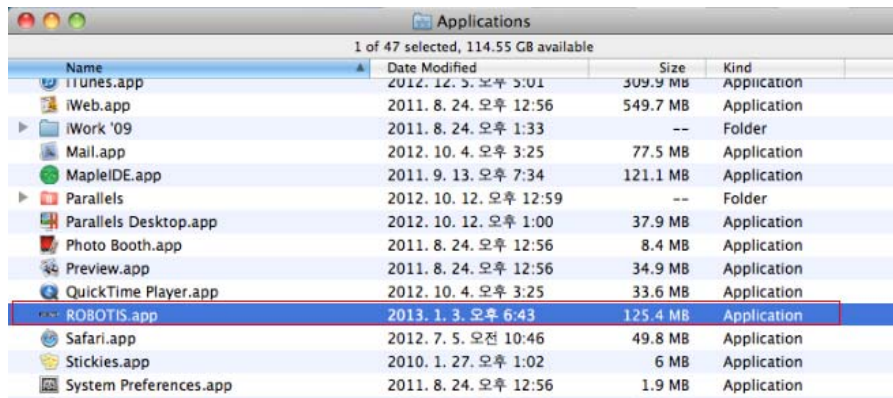


단 USB 허브에서 다수의 USB 장치와 함께 연결하는 경우는 되도록 피해주시고 **PC와 직접 연결하는 방법을 추천합니다**. 간혹 허브에서 전류가 충분하지 않으면 다운로드 실패하는 경우가 발생합니다. 따라서

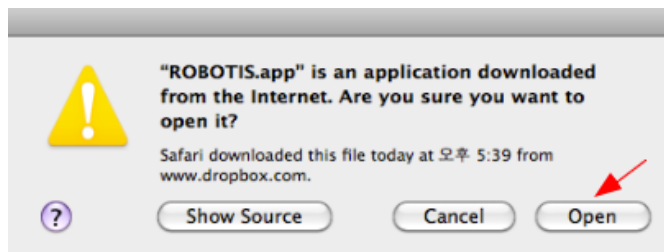
반드시 OpenCM9.04에 충분한 전류를 공급할 수 있는 USB 포트에 연결하기 바랍니다.

C. ROBOTIS OpenCM을 실행합니다.

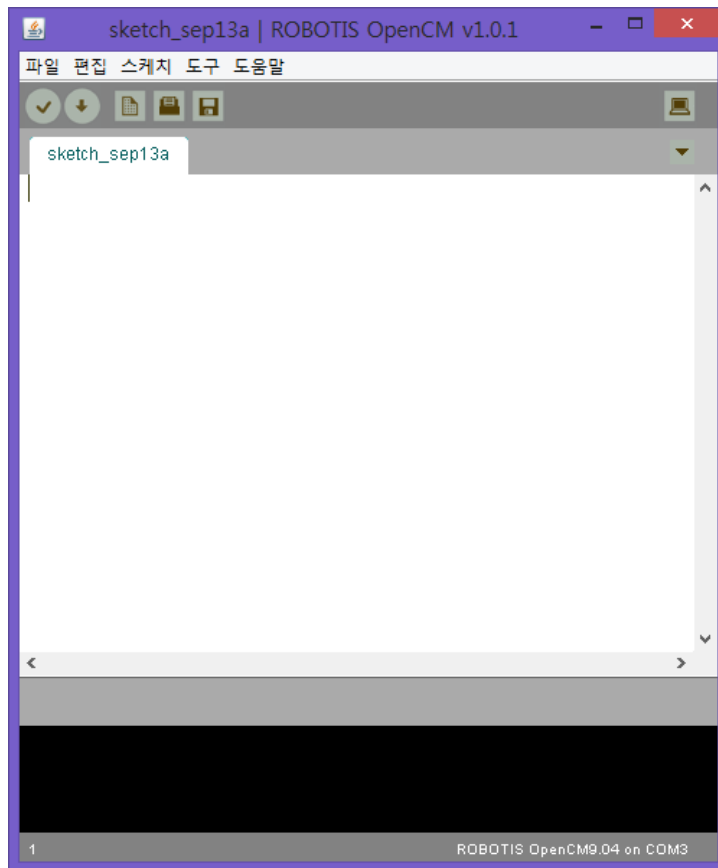
Finder를 이용해서 Application 폴더를 보면 ROBOTIS.app 응용프로그램 패키지가 있고 이것을 더블 클릭해서 실행합니다.



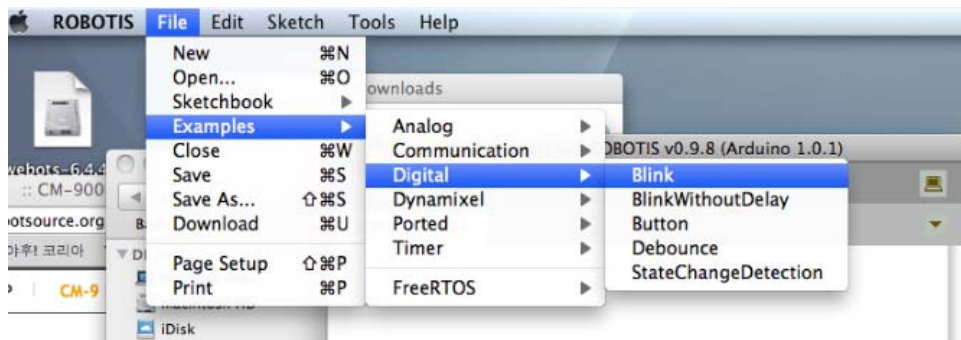
그냥 Open을 선택합니다.

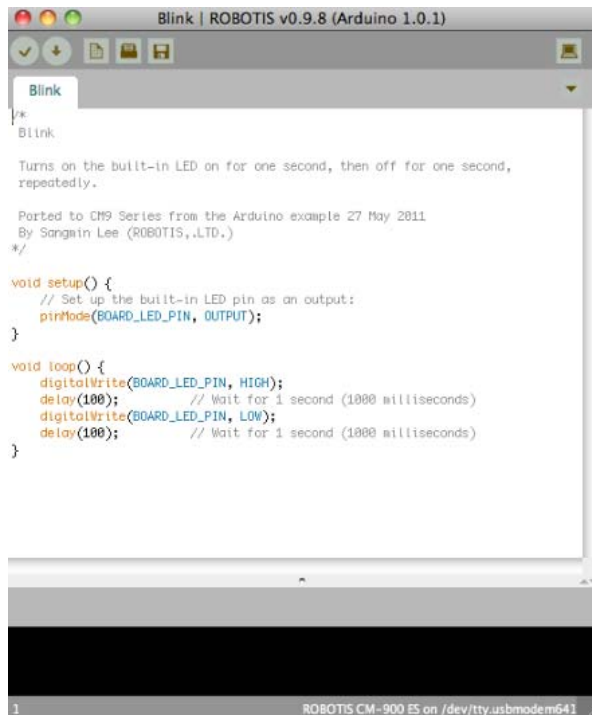


아래와 같이 ROBOTIS OpenCM이 실행됩니다.

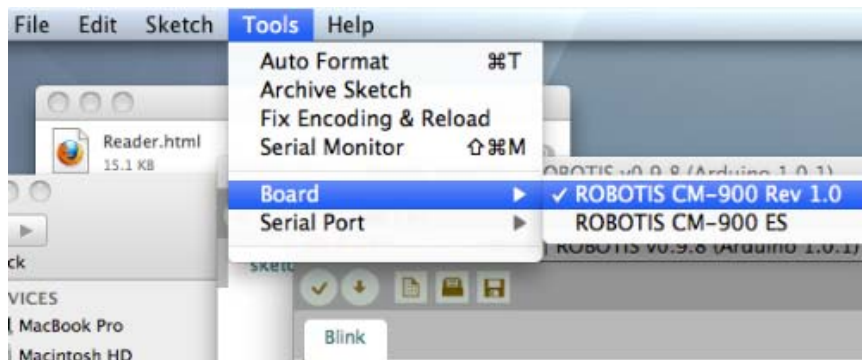


D. Blink 예제를 Open합니다.

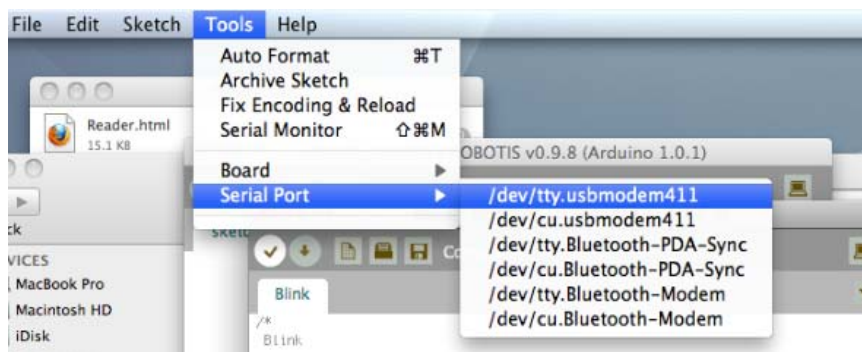




E. 보드를 선택합니다.

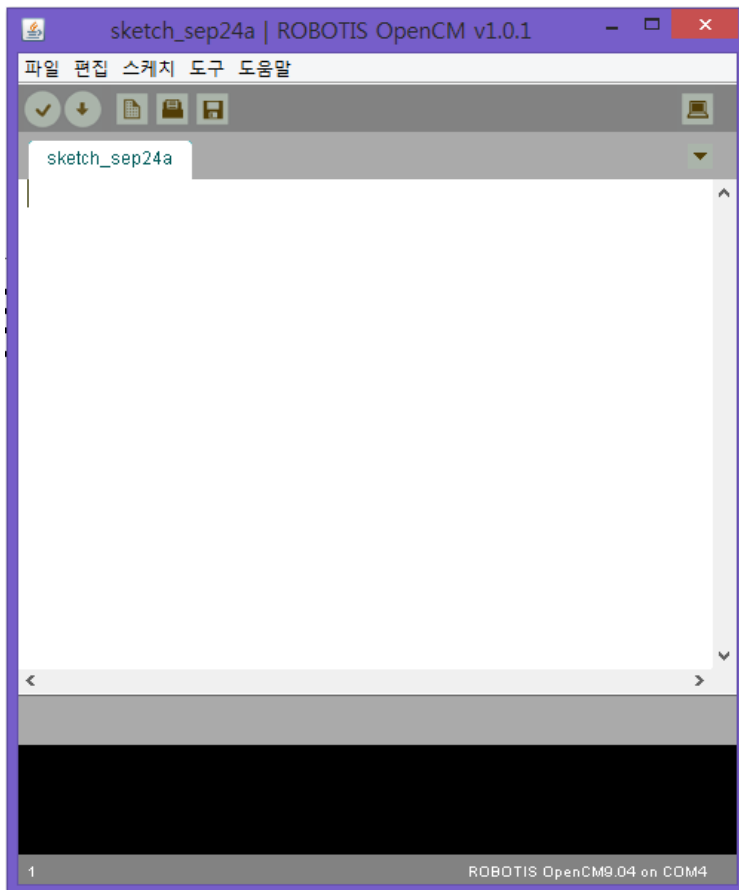


F. 시리얼 포트를 선택합니다.



⑥ SW 환경 설정

USB 드라이버 설치를 무사히 마치셨다면 ROBOTIS_OpenCM.exe를 실행해 주세요. Linux, Mac OS X 사용자는 바로 ROBOTIS OpenCM을 실행하면 됩니다. 아래와 같은 화면이 나타납니다.

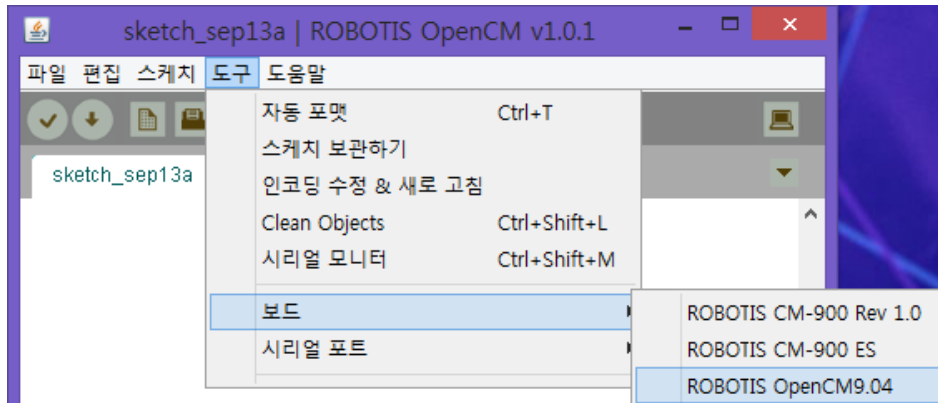


<ROBOTIS OpenCM 실행화면>

ROBOTIS OpenCM에서 반드시 선택해야 될 항목은 2가지, 보드 선택과 시리얼 포트 선택입니다. 환경 설정은 옵션입니다.

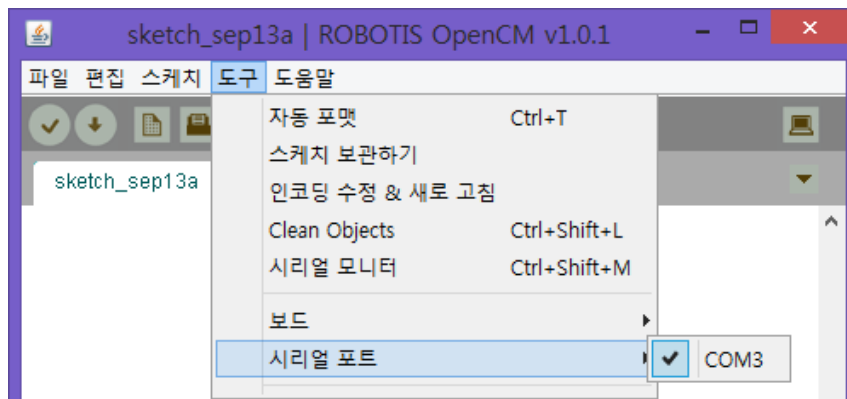
A. 보드 선택

OpenCM9.04의 H/W 버전에 맞게 반드시 선택하셔야 합니다.



B. 시리얼 포트 선택

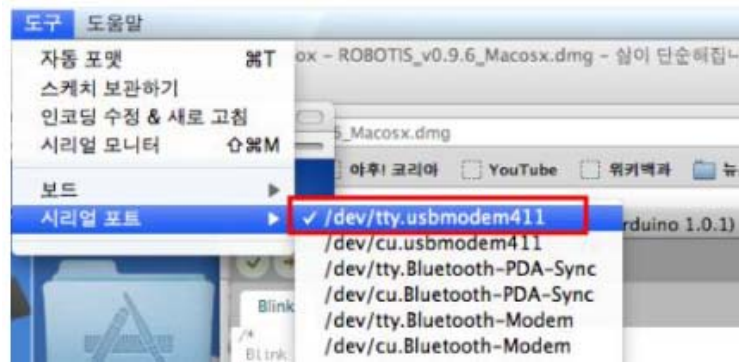
위에서 USB 드라이버를 설치하면서 확인했던 COM포트를 선택합니다.



Linux의 경우 /dev/ttyACMX를 선택합니다. 숫자 X는 여러분의 PC환경에 따라 다릅니다.

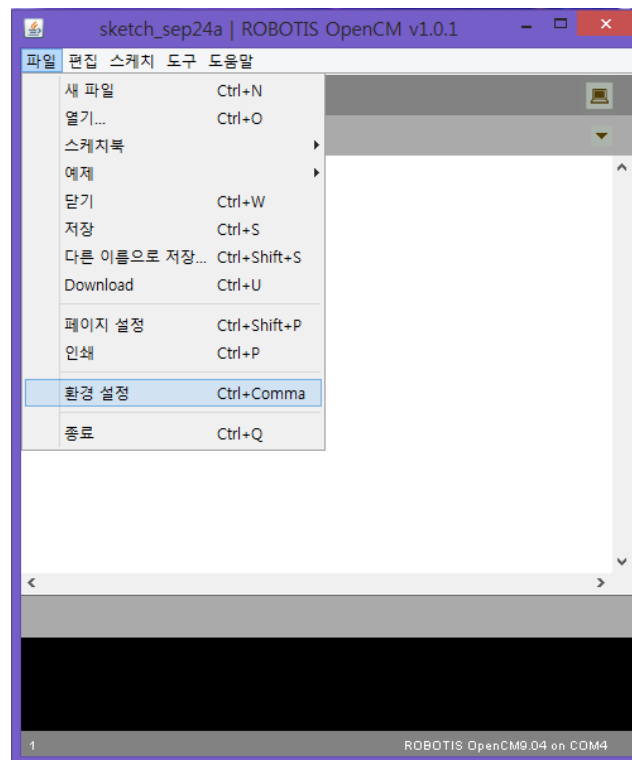


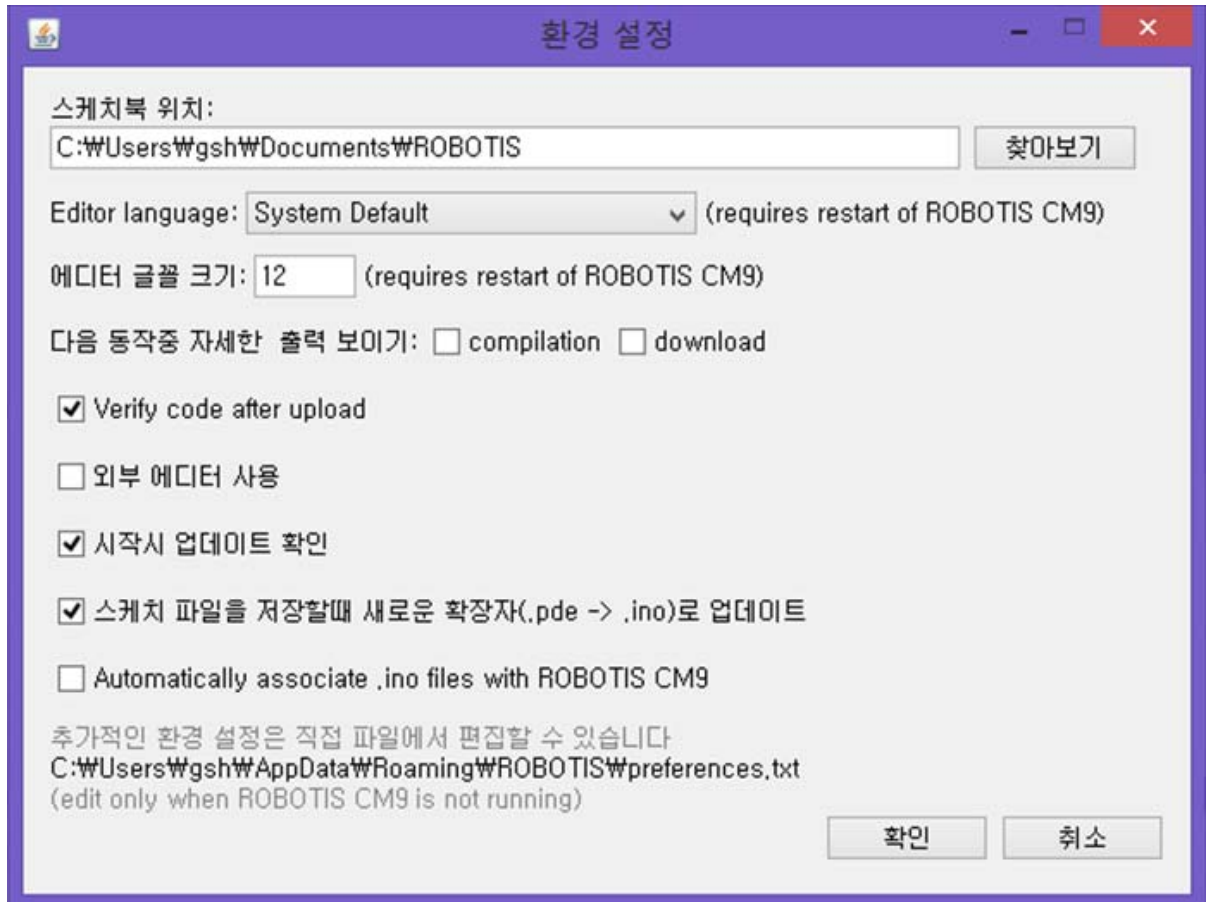
Mac OS X는 tty.usbmodemX11을 선택합니다. 마찬가지로 X는 여러분의 PC환경에 따라서 다릅니다.



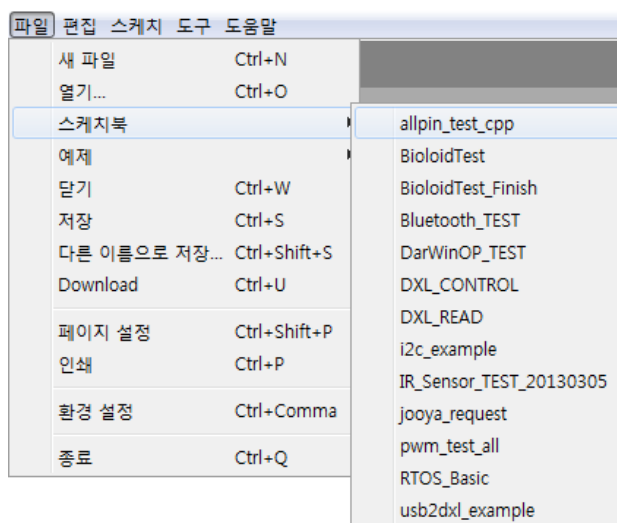
C. 환경설정

아래와 같이 파일 -> 환경설정을 통해 세부적인 변경을 할 수 있습니다.





- i. 스케치북 위치 : 사용자 기본 작업 디렉토리입니다. 기본적으로 스케치 코드를 저장하면 여기 경로에 저장되고 ROBOTIS OpenCM이 실행되면서 그 경로의 스케치 코드를 모두 읽어와서 파일 -> 스케치북 항목에서 모두 열거됩니다.



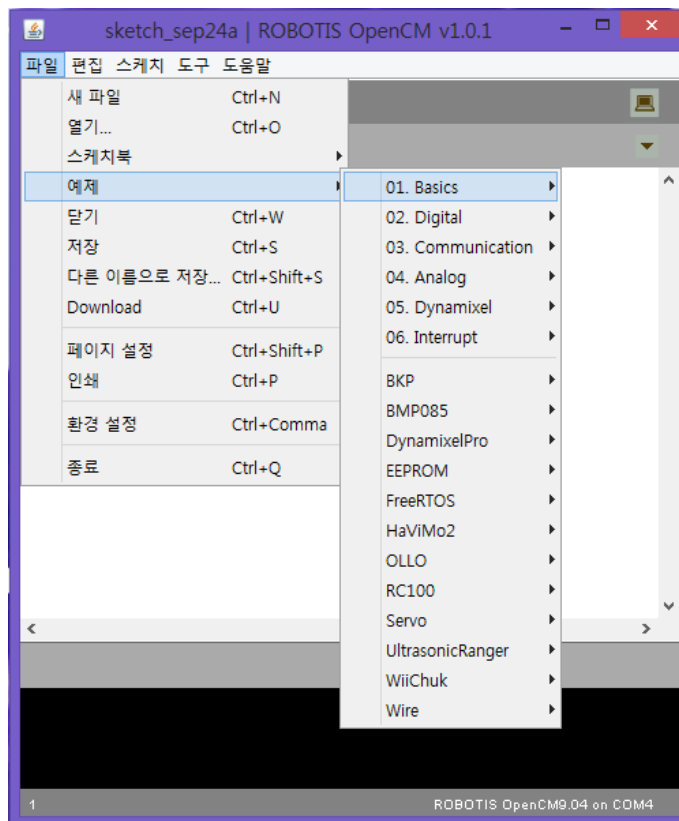
- ii. Editor language : 글꼴을 변경합니다.
- iii. 다음 동작중 자세한 출력 보이기 : compilation에 체크를 하면 컴파

일 과정중에 자세한 출력을 합니다. Download에 체크를 하면 컴파일 후 다운로드 과정을 자세히 출력합니다.

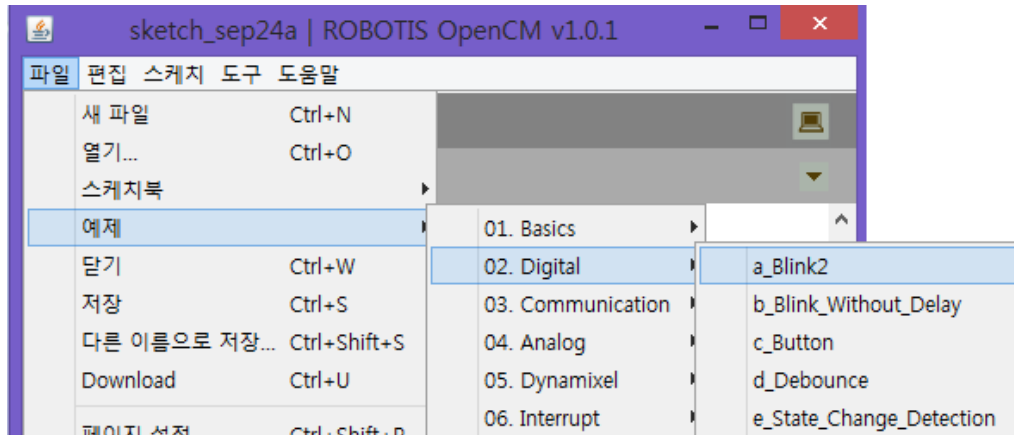
```
.C:\ROBOTIS\hardware\tools\arm\bin\arm-none-eabi-gcc -Os -g -mcpu=cortex-m3  
-mthumb -march=armv7-m -nostdlib -ffunction-sections -fdata-sections  
-Wl,--gc-sections -DBOARD_CM900_REV10 -DMCU_STM32F103C8 -DVECT_TAB_FLASH  
-DSTM32_MEDIUM_DENSITY -DERROR_LED_PORT=GPIOB -DERROR_LED_PIN=2  
1  
ROBOTIS CM-900 Rev 1.0 on COM36
```

⑦ 예제 다운로드

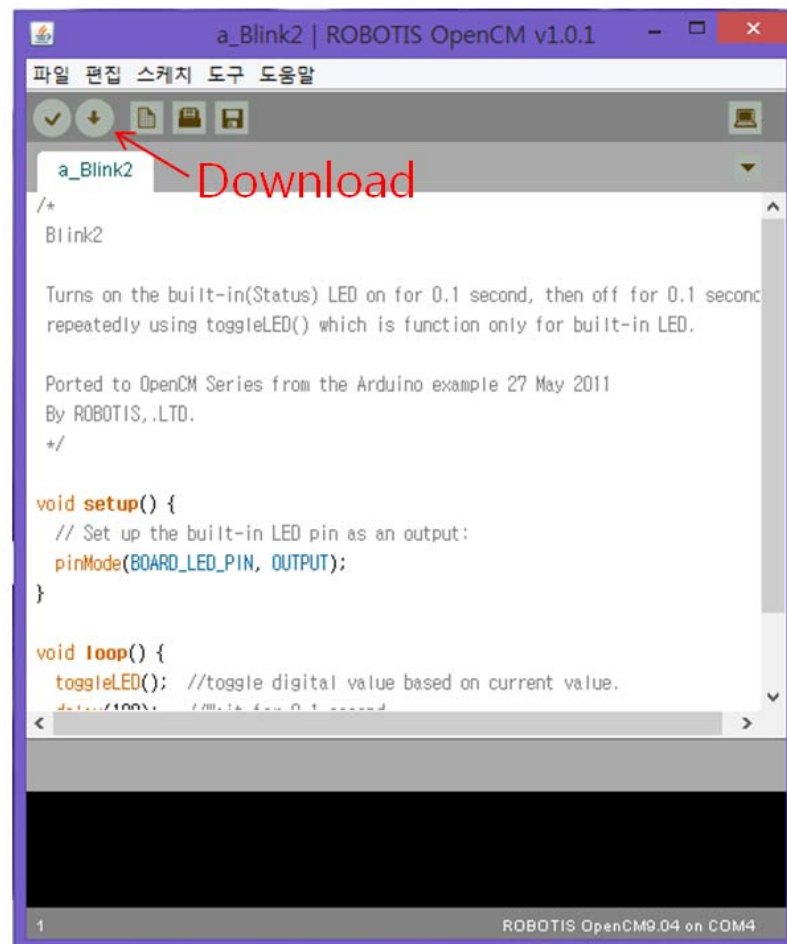
ROBOTIS OpenCM 프로그램은 개발에 필요한 예제를 파일 -> 예제 메뉴를 통해 손쉽게 접근할 수 있습니다.



예를 들어 Digital IO 입출력 관련 개발을 하기 전에 반드시 Blink예제를 열어서 분석 후에 OpenCM9.04에 다운로드를 해서 동작을 확인하면 더욱 빠른 개발을 진행할 수 있습니다.



아래처럼 Blink 예제를 열었습니다. 그리고 바로 아래 화살표 모양의 다운로드 버튼을 눌러서 다운로드를 진행합니다.

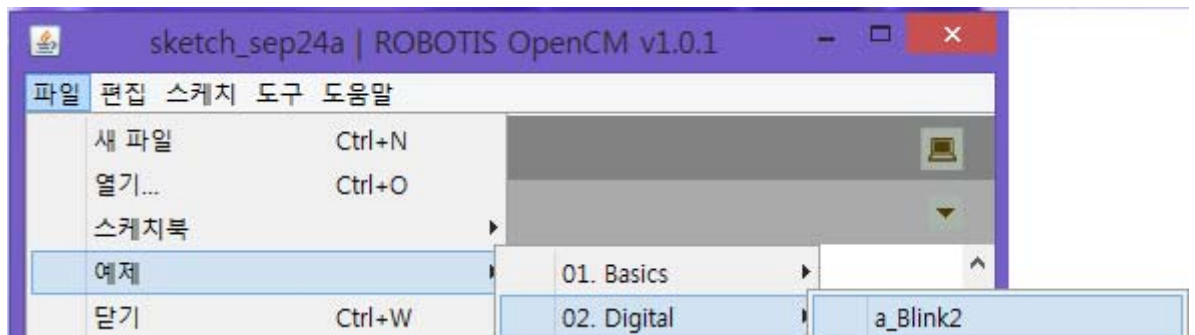


이제 ROBOTIS OpenCM SW에서 사용되는 함수들에 대한 자세한 설명은 4장 API Reference를 참고하시고 다양한 API 사용법은 5장 배우기 (Learning) 항목을 참고하시기 바랍니다.

⑧ Blink예제

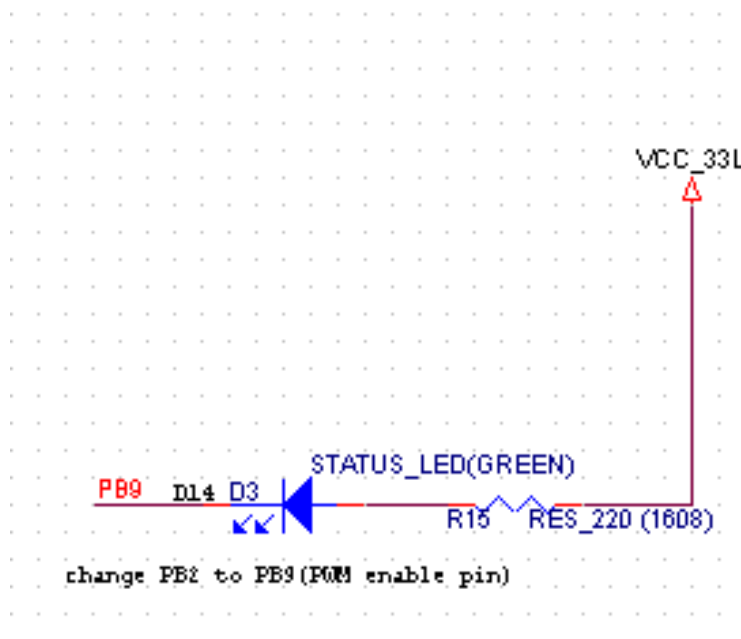
OpenCM9.04의 Blink 예제는 Arduino의 Blink 예제에서 포팅하였으며 라이선스는 Arduino의 정책을 따릅니다.

파일 -> 예제 -> Digital -> a_Blink2 순서로 예제를 열기합니다.



A. 회로도

OpenCM9.04의 Status LED는 아래와 같이 CPU의 D14(PB9)번 IO포트 를 통해서 on/off를 제어합니다.



전형적인 Current Sink 회로를 통해 LED를 제어하며, D14(PB9)가 HIGH이면 LED가 OFF되고 LOW이면 LED가 ON됩니다.

B. 스케치 코드

```
void setup() {
    // Set up the built-in LED pin as an output:
    pinMode(BOARD_LED_PIN, OUTPUT);
}

void loop() {
    digitalWrite(BOARD_LED_PIN, HIGH);
    delay(100);          // Wait for 1 second (1000 milliseconds)
    digitalWrite(BOARD_LED_PIN, LOW);
    delay(100);          // Wait for 1 second (1000 milliseconds)
}
```

OpenCM9.04의 어떤 한 IO포트를 사용하기 위해서는 반드시

pinMode(핀번호, 핀모드)함수를 통해 초기화를 시켜주어야 합니다.

여기서 핀 번호는 OpenCM9.04의 IO포트 실크 스크린을 참고하면 되며, BOARD_LED_PIN이라는 define은 미리 Core Library에 선언되어 있으므로 그냥 BOARD_LED_PIN이라고 쓰면 D16번 핀이 자동으로 할당됩니다. 아래의 경로에 있는 OpenCM9.04.h 헤더파일을 열어보면 미리 선언된 다른 define도 볼 수 있으며 원하는 define을 직접 선언해서 사용하셔도 됩니다.

ROBOTIS\hardware\robotis\cores\robotis\OpenCM

```
#ifndef OpenCM904_H_
#define OpenCM904_H_
#include "gpio.h"

#define CYCLES_PER_MICROSECOND 72
#define SYSTICK_RELOAD_VAL 71999 /* takes a cycle to reload */
/*
 * [ROBOTIS][CHANGE] CM-900 Do not have built-in button.
 * 2013-04-22
 */
#define BOARD_BUTTON_PIN 25//28
#define BOARD_LED_PIN 14//16
#define BOARD_PIN 20 // added
```

Blink 예제는 단순히 IO포트를 통해 HIGH/LOW의 디지털 신호만 출력하면 되기 때문에 핀 모드는 OUTPUT으로 설정하면 됩니다.

만약 setup()함수에서 핀 모드 설정이 되었다면 loop()함수 내에서

digitalWrite(핀번호, HIGH/LOW)를 통해 핀 출력 제어를 할 수 있으며 **delay(millisecond)**함수를 통해 CPU를 동작을 지연시키면서 LED가 on/off 되는 주기를 조절할 수 있습니다.

C. 데이터 확인

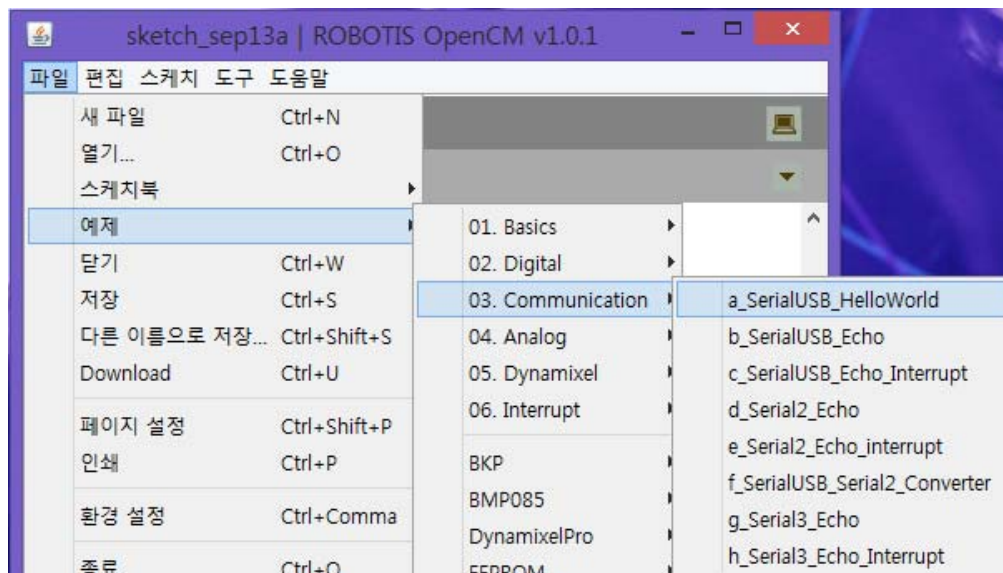
STATUS LED가 on/off 되는지 확인합니다.

⑨ SerialUSB_HelloWorld 예제

OpenCM9.04와 외부장치(예를 들면 PC)의 가장 간편한 연결 방법은 USB를 통해 통신을 수행하는 것입니다. 이를 위해 SerialUSB라는 클래스 인스턴스가 미리 선언되어 있으며, SerialUSB의 다양한 Method를 통해 USB 통신을 수행합니다.

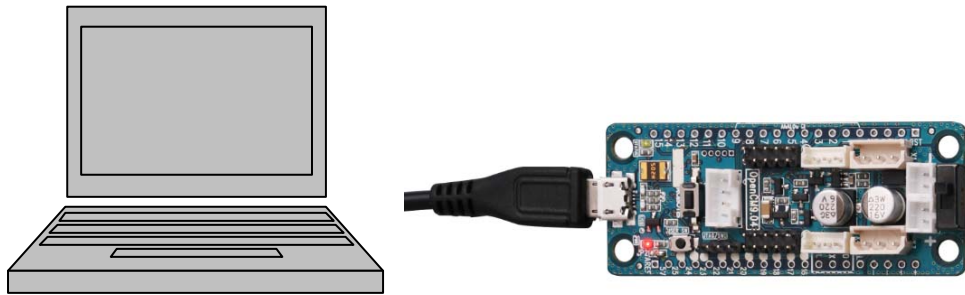
여기서는 SerialUSB_HelloWorld 예제를 통해 PC와 터미널 통신을 어떻게 수행하는지 배워봅니다.

파일 -> 예제 -> Communication -> SerialUSB_HelloWorld 예제를 엽니다.



A. OpenCM9.04와 PC 연결

다운로드를 할 때와 같이 제공된 Micro USB 케이블을 사용해서 PC와 OpenCM9.04를 연결합니다.



B. 스케치 코드

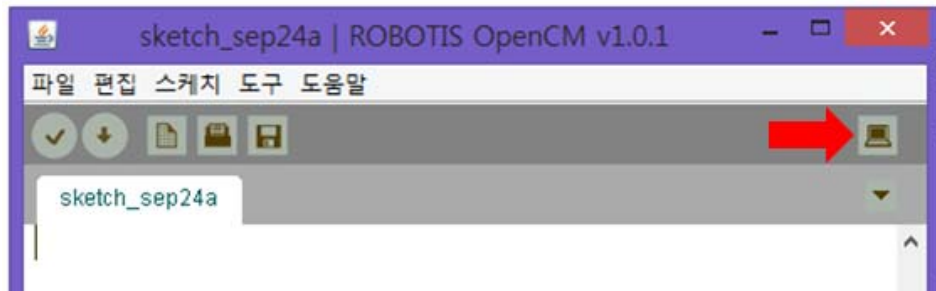
```
void setup() {  
    //Initialize USB Serial  
    SerialUSB.begin();  
}  
int nCount=0;  
void loop() {  
    //print "Hello World!!" to PC though USB Virtual COM port  
    SerialUSB.println("Hello World!!");  
    SerialUSB.print("nCount : "); // display nCount variable and increase  
    SerialUSB.println(nCount++);  
    delay(1000);  
}
```

Setup()에서 클래스 SerialUSB 인스턴스에 대한 초기화를 begin()메소드를 통해 수행합니다. 인자는 void타입이므로 아무것도 지정하지 않습니다. 다른 시리얼 장치(Serial1,2,3)와 사용법이 매우 유사하지만 SerialUSB.begin() 메소드는 Baud rate 인자가 필요 없습니다.

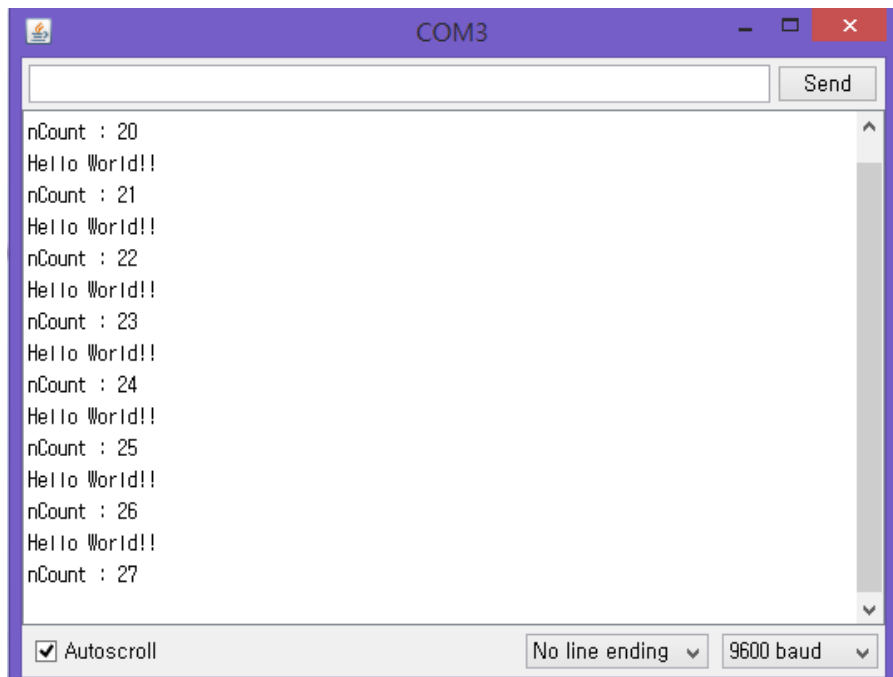
Loop()에서 SerialUSB.print() 또는 SerialUSB.println() 명령어를 통해 출력을 수행합니다. 차이점이라면 SerialUSB.println()명령어는 마지막에 줄 바꿈까지 수행합니다.

C. 데이터 확인

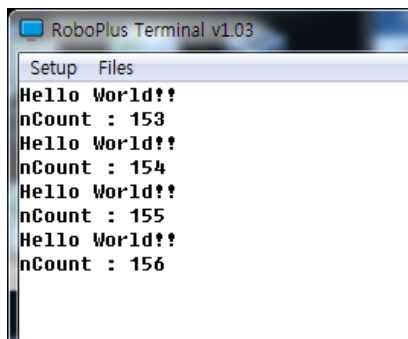
간단하게 ROBOTIS OpenCM SW를 통해 아래와 같이 시리얼 모니터로 간단히 확인가능하며, RoboPlus Terminal이나 하이퍼 터미널을 이용해서도 OpenCM9.04에서 출력하는 데이터를 확인할 수 있습니다.



우측 상단의 노트북 모양의 아이콘을 클릭하면 아래와 같이 시리얼 모니터가 USB COM포트를 통해 바로 연결됩니다.



아래는 RoboPlus Terminal로 연결한 화면입니다. 다운로드할 때와 같은 COM포트로 연결하면 되고 Baud rate는 아무거나 하셔도 관계 없습니다.



단, 이 예제는 오직 출력만 수행하며 터미널에서 입력은 지원하지 않습니다.

⑩ SerialUSB_Echo 예제

2번 SerialUSB_HelloWorld는 출력만 수행하는데 비해서 SerialUSB_Echo는 어떻게 USB를 통해 입출력을 수행하는지에 대한 자세한 방법을 알 수 있습니다.

A. 스케치 코드

```
void setup(){
  //USB Serial initialize
  SerialUSB.begin();
}
void loop(){
  // when you typed any character in terminal
  if(SerialUSB.available()){
    //print it out though USB
    SerialUSB.print((char)SerialUSB.read());
  }
}
```

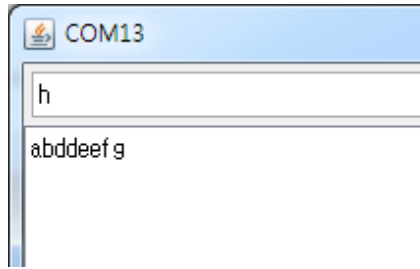
SerialUSB_HelloWorld와 같이 setup()에서 Baud rate 인자 없이 SerialUSB.begin() 메소드를 통해 초기화를 수행합니다.

Loop()에서 CPU에서 반복적으로 USB를 통해 데이터가 들어오는지 확인합니다. 만약에 USB를 통해 어떤 데이터가 입력되면 if문 안의 SerialUSB.available() 메소드가 1을 반환하고 없다면 계속 0을 반환합니다. 입력된 데이터가 있다면 if조건이 참(True)이 되면서 SerialUSB.read()함수를 통해 받은 1byte 문자를 바로 SerialUSB.print() 메소드의 인자로 넘겨줍니다. 만약 Char형 캐스팅 없이 그냥 넘겨주면 Hex값으로 데이터가 넘어가서 터미널에서 문자로 출력되지 않음을 유의하셔야 합니다.

B. 데이터 확인

SerialUSB_HelloWorld와 같이 ROBOTIS OpenCM SW의 시리얼 모니터를 통해 에코 입출력을 수행할 수 있으며 RoboPlus나 하이퍼터미널을 통해서도 확인 가능합니다. 연결이 되면 키보드를 통해 데이터 입력을 하면 OpenCM9.04가 되받아서 그대로 출력하기 때문에 에코 예제라고

합니다.

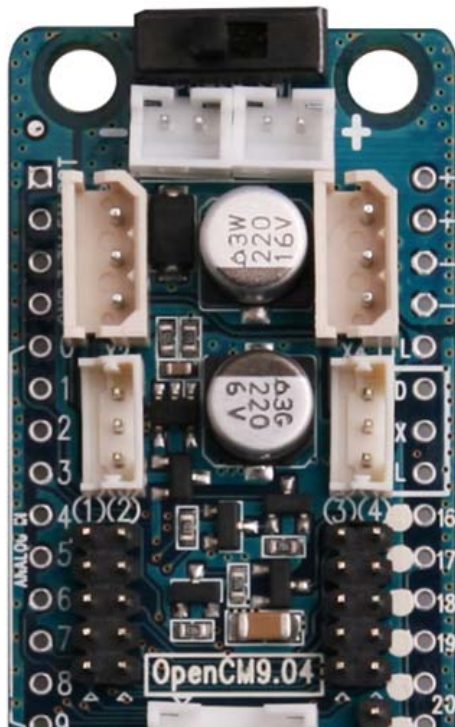


키보드로 문자입력을 해보면, 입력한 문자가 그대로 터미널에 출력되는 것을 확인할 수 있습니다.

⑪ 다이나믹셀 Basic 예제

OpenCM 9.04는 쉽고 편리하게 로봇을 제작할 수 있도록 다이나믹셀 연결 커넥터를 탑재한 제어기입니다. TTL 3핀과 XL-Series 3핀을 위한 마이크로 TTL 3핀을 통해 TTL 종류의 다이나믹셀을 제어할 수 있습니다. 단, 다이나믹셀 예제를 위해서는 DC 혹은 Battery 단자를 통해 제어하려는 다이나믹셀에 맞는 12V 또는 7.4V 전원을 반드시 공급해야 합니다.

다이나믹셀 Basic 예제는 기본적으로 Blink 예제와 동일한 구조이고 LED on/off 대신 다이나믹셀의 위치 제어를 통해 특정 위치를 왔다가 갔다가 하는 간단한 제어를 배워 봅니다.



Battery

TTL

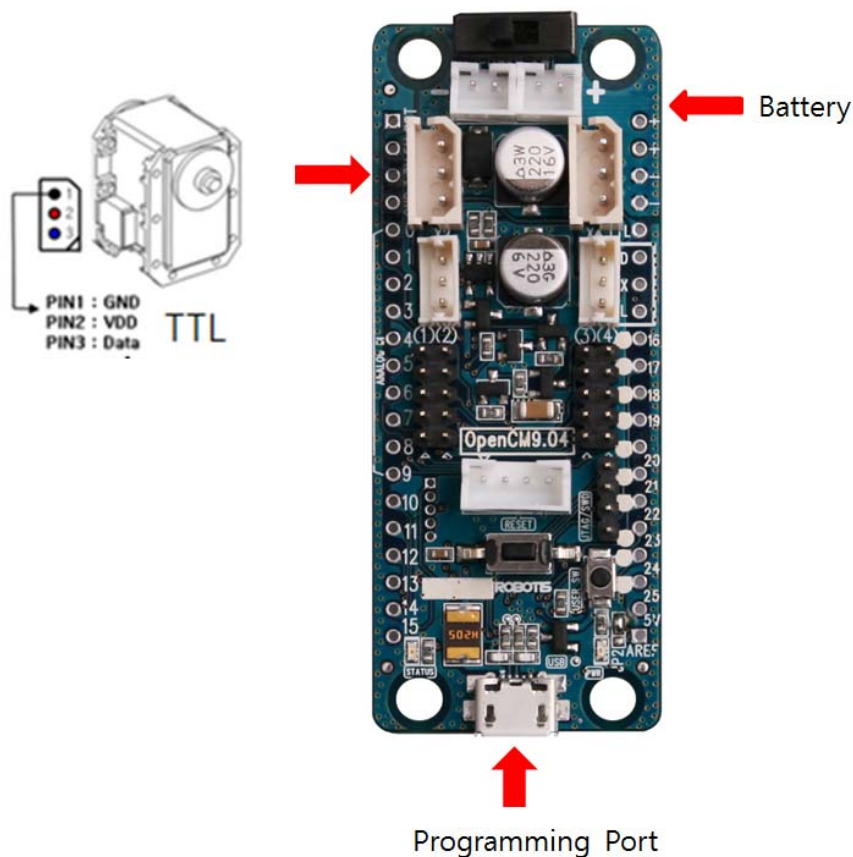
XL-Series

A. 다이내믹셀 연결도

아래와 같이 3핀 다이내믹셀을 연결합니다. 반드시 해당 다이내믹셀에 맞는 전원을 공급할 수 있는 DC SMPS 또는 Battery를 연결해야 합니다. 프로그래밍은 ROBOTIS OpenCM SW에서 USB 통신을 이용해 프로그래밍을 수행합니다.

반드시 연결된 다이내믹셀의 ID가 1번이고 통신 속도가 1Mbps로 설정되어 있음을 Dynamixel Wizard를 통해 확인하길 바랍니다.

OpenCM9.04는 다이내믹셀과의 통신을 Serial1을 통해 수행합니다. 따라서 다이내믹셀을 사용한다면 Serial1(USART1)은 사용하지 않습니다.



B. 스케치 코드

```
void setup() {
    // Initialize the dynamixel SDK:
    Dxl.begin(1);
}

void loop() {
    delay(1000);           // Wait for 1 second (1000 milliseconds)
    Dxl.writeWord(1, 30, 100); //Turn dynamixel ID 1 to position 100
    delay(1000);           // Wait for 1 second (1000 milliseconds)
    Dxl.writeWord(1, 30, 1000); //Turn dynamixel ID 1 to position 1000
}
```

다이나믹셀 제어를 수행하기 앞서서 반드시 다이나믹셀 버스를 초기화해야 합니다. Setup()에서 Dxl.begin(Baud rate인자)을 통해 다이나믹셀 버스를 초기화 합니다. OpenCM9.04는 연결된 모든 다이나믹셀은 여기서 초기화 했던 버스 위에서 움직이게 됩니다. Baud rate인자는 다이나믹셀 버스 속도를 초기화하는 인자이며, 여기서 1은 1Mbps속도로 초기화를 수행합니다. 자세한 정보는 API Reference또는 다이나믹셀 E-

manual을 참고 바랍니다.

Loop()함수 내에서는 Dxl.writeWord(ID, Address, Value)를 통해 위치 제어를 수행합니다. Address인자에 Goal Position(L) 주소를 넣고 Value에 1 Word에 해당하는 크기의 값을 넣어줍니다. 여기서는 100과 1000 위치를 왔다가 갔다하는 동작을 1초 지연간격으로 수행하고 있습니다.

참고로 Goal Position은 다이나믹셀 모델마다 범위가 다릅니다. 해당 예제는 0~ 1023(0x3FF)까지 사용가능한 다이나믹셀을 했을 때 이동하는 시간을 고려해서 1초 지연을 사용하였습니다. 0~4095(0xFFF)까지 위치 제어가 가능한 다이나믹셀 모델에 해당 예제에 수행했을 때 움직이는 각도가 매우 작을 수 있으며 더 작은 지연이 적당할 수도 있습니다. 해당 모델에 대한 자세한 Goal Position 제어는 반드시 E-manual을 참고 바랍니다.

C. 데이터 확인

이 예제에서 확인할 수 있는 데이터는 없습니다.

⑫ 다이나믹셀 ReadWrite예제

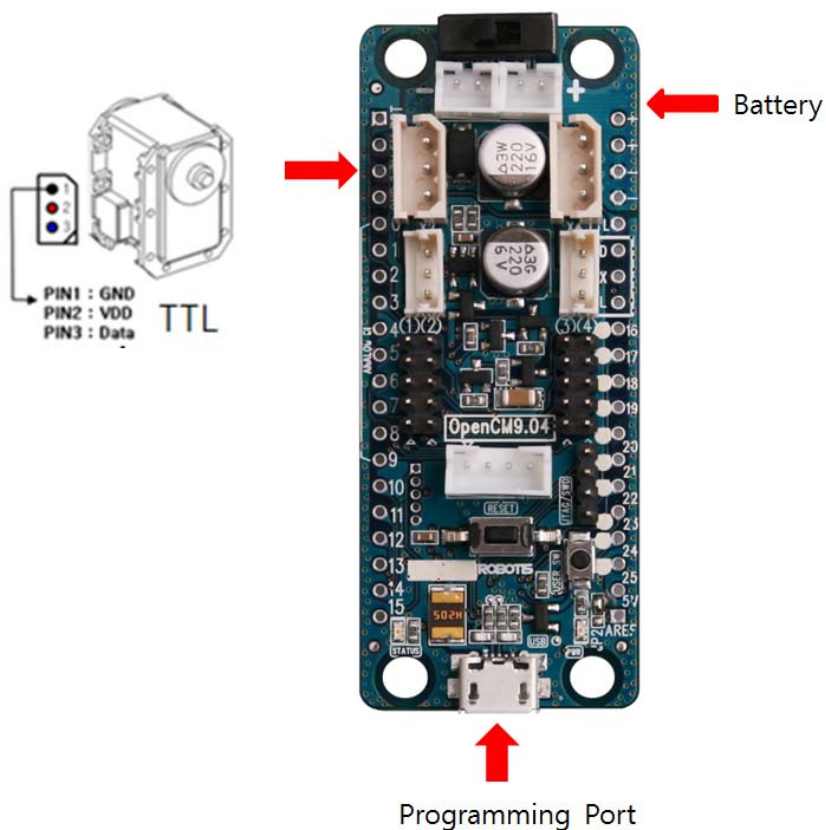
다이나믹셀은 읽기와 쓰기가 가능한 액추에이터입니다. 이 예제는 다이나믹셀의 움직임 유무를 확인하고, 움직임이 없을 때 이동방향을 바꾸어서 다른 위치로 이동하는 예제입니다. 이동이 끝나면 현재의 위치를 출력하고 다음 동작으로 넘어갑니다.

A. 다이나믹셀 연결도

다이나믹셀 Basic 예제와 연결방법은 동일 합니다. 반드시 다이나믹셀을 위한 별도의 전원이 필요합니다.

반드시 연결된 다이나믹셀의 ID가 1번이고 통신 속도가 1Mbps로 설정되어 있음을 Dynamixel Wizard를 통해 확인하길 바랍니다.

OpenCM9.04는 다이나믹셀과의 통신을 Serial1을 통해 수행합니다. 따라서 다이나믹셀을 사용한다면 Serial1(USART1)은 사용하지 않습니다.



B. 스케치 코드

아래와 같이 미리 define을 통해 컨트롤 테이블을 참조해서 제어하려는 레지스터의 주소를 미리 선언하면 좋습니다.

```
#define P_GOAL_POSITION_L    30
#define P_PRESENT_POSITION_L 36
#define P_MOVING             46

word Position;
word wPresentPos;
byte INDEX = 0;
byte bMoving, CommStatus;
byte id = 1;
word GoalPos[2] = {0, 1023};

void setup() {
    Dxl.begin(1);
    //print to USB port
    SerialUSB.begin();
}
```

```

void loop() {
  bMoving = Dxl.readByte( id, P_MOVING);
  CommStatus = Dxl.getResult();
  if( CommStatus == COMM_RXSUCCESS ){
    if( bMoving == 0 ){
      // Change goal position
      if( INDEX == 0 )
        INDEX = 1;
      else
        INDEX = 0;
      // Write goal position
      Dxl.writeWord( id, P_GOAL_POSITION_L, GoalPos[ INDEX] );
    }
    // Read present position
    wPresentPos = Dxl.readWord( id, P_PRESENT_POSITION_L );
    SerialUSB.print("Goal Position : ");
    SerialUSB.println(GoalPos[ INDEX]);
    SerialUSB.print("Present position :");
    SerialUSB.println(wPresentPos);
    SerialUSB.println("Success");
  }else {
    SerialUSB.println("Fail");
  }
  delay(1000);
}

```

bMoving = Dxl.readByte(id, P_MOVING);

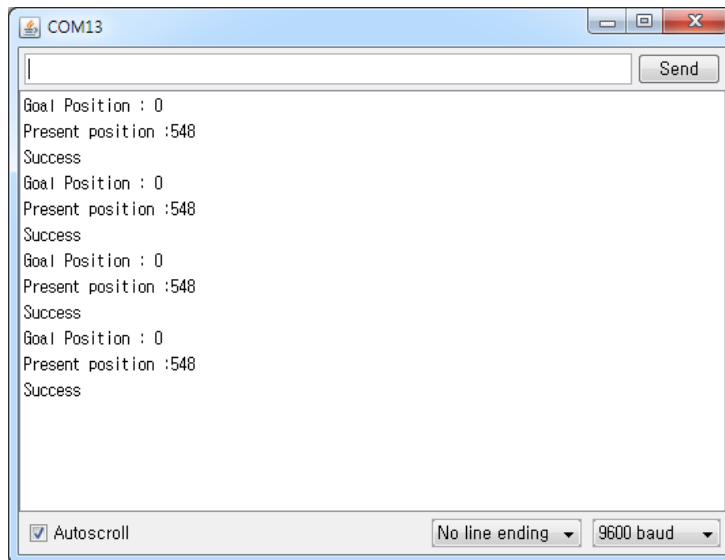
을 통해 다이나믹셀의 움직임이 있으면 1을 할당하고 움직임이 없으면 0을 할당합니다. Dxl.getResult()를 통해 성공적으로 전송이 되었다면 bMoving 값이 0이면 Goal Position의 Index값을 바꾸고 Dxl.writeWord(id, P_GOAL_POSITION, GoalPos[INDEX])의 GoalPos 배열을 이용해서 변경된 Goal Position에 대한 명령을 다이나믹셀로 전송합니다. 그리고 마지막으로 GoalPos[INDEX] 값을 USB를 통해 출력하고 아래의 커맨드를 이용해 다이나믹셀의 현재 위치를 확인합니다.

wPresentPos = Dxl.readWord(id, P_PRESENT_POSITION_L);

C. 데이터 확인

SerialUSB의 예제와 같이 ROBOTIS OpenCM SW의 시리얼 모니터를 통

해 바로 다이내믹셀의 GoalPos[INDEX]값과 현재 위치값을 확인하실 수 있습니다.



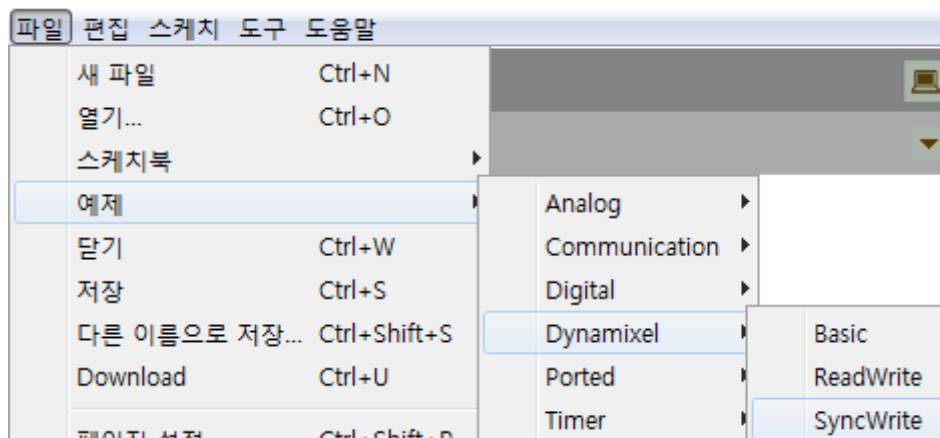
⑬ 다이내믹셀 SyncWrite 예제

다이내믹셀은 Broadcast ID를 통해 버스에 연결된 모든 다이내믹셀을 동시에 제어할 수 있습니다. 이러한 복수의 다이내믹셀에 대한 동시제어를 Sync Write라고 합니다.

이 예제를 통해서 5개의 다이내믹셀을 동시에 제어하는 Syncwrite 패킷을 만드는 방법을 배웁니다. 참고로 Syncwrite에 대한 기본지식은 반드시 아래의 E-manual을 통해서 학습하고 제어하기 바랍니다.

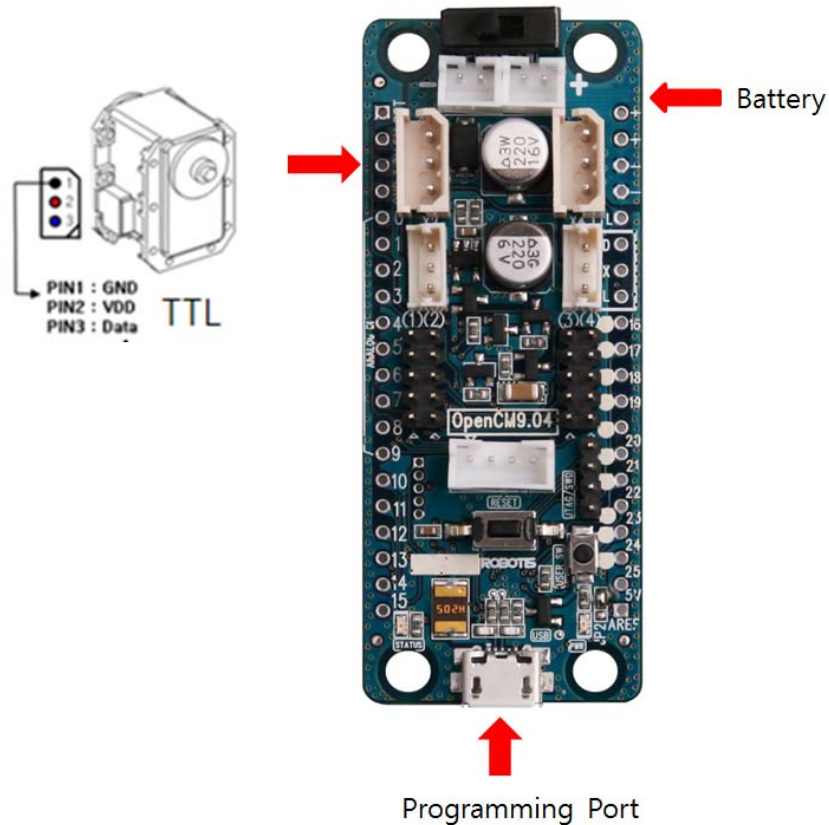
http://support.robotis.com/ko/e-manual_kor.htm#product/dynamixel/communication/dxl_instruction.htm

예제는 파일 -> 예제 -> Dynamixel -> SyncWrite 순서로 선택합니다.



A. 다이내믹셀 연결도

아래와 같이 ID=1번부터 5번까지 설정된 다이내믹셀 5개를 3핀 또는 4핀 포트를 통해 연결합니다. 마찬가지로 기본 통신 속도는 1Mbps로 설정합니다. 이렇게 연결된 다이내믹셀들은 하나의 버스위에 연결되어 있고 각각 제어 또는 동시제어를 수행할 수 있습니다.



OpenCM9.04는 다이내믹셀과의 통신을 Serial1을 통해 수행합니다. 따라서 다이내믹셀을 사용한다면 Serial1(USART1)은 사용하지 않습니다.

B. 스케치 코드

아래와 같이 다이내믹셀 제어에 필요한 컨트롤 레지스터 주소값을 E-manual을 참고해서 미리 define 해놓습니다.


참고로 1 Word(2byte) 크기의 컨트롤 레지스터는 하위 LOW 주소만으로 충분히 제어할 수 있습니다.

```
#define P_GOAL_POSITION_L 30
#define P_GOAL_SPEED_L 32

#define NUM_ACTUATOR 5 // Number of actuator
#define MAX_POSITION 1023
```

AmpPos는 연결된 모든 다이나믹셀들의 초기 위치값입니다.

```
word AmpPos = 512;
word wPresentPos;
word GoalPos = 0;
byte id[NUM_ACTUATOR];
byte CommStatus;
byte i;
```



setup에서 다이나믹셀 버스 초기화를 위해 Dxl.begin(1)을 선언하고 다이나믹셀에서 읽은 정보를 표시하기 위해 SerialUSB.begin()도 선언합니다. 그리고 나머지는 다이나믹셀의 속도와 초기 위치를 설정합니다.

```
void setup() {
  Dxl.begin(1);
  SerialUSB.begin();
  //Insert dynamixel ID number to array id[]
  for(i=0; i<NUM_ACTUATOR; i++){
    id[i] = i+1;
  }
  // Set goal speed
  Dxl.writeWord( BROADCAST_ID, P_GOAL_SPEED_L, 0 );
  // Set goal position
  Dxl.writeWord( BROADCAST_ID, P_GOAL_POSITION_L, AmpPos );
  delay(1000);
}
```

Loop()에서 Syncwrite를 위한 패킷을 만드는 부분과 다이나믹셀 통신 결과를 표시하는 부분으로 나눌 수 있습니다.

Packet을 만드는 방식은 아래의 형식에 맞추어서 하지만 자세한 이론적인 설명은 반드시 다이나믹셀 Instruction에 대한 E-manual을 참고합니다.

ID 0xFE

Length $(L+1) \times N + 4$ (L:RX-64별 Data Length, N:RX-64의 개수)

Instruction 0x83

Parameter1 Data를 쓰고자 하는 곳의 시작 Address

Parameter2 쓰고자 하는 Data의 길이 (L)

Parameter3 첫 번째 RX-64의 ID

Parameter4 첫 번째 RX-64의 첫 번째 Data

Parameter5 첫 번째 RX-64의 두 번째 Data

...

Parameter L+3 첫 번째 RX-64의 L번째 Data

Parameter L+4 두 번째 RX-64의 ID

Parameter L+5 두 번째 RX-64의 첫 번째 Data

Parameter L+6 두 번째 RX-64의 두 번째 Data

...

Parameter 2L+4 두 번째 RX-64의 L번째 Data



일반적으로 동시 제어 가능한 다이나믹셀은, 1개의 명령 패킷이 4바이트인 경우 26개까지 가능합니다.
다이나믹셀의 수신버퍼 용량이 143byte 이므로 Packet의 길이가 143byte를 초과하지 않도록 하십시오.

Dynamixel을 위한 Packet을 만들 때 주의할 점은 워드 단위의 Packet 이라고 하위바이트만 접근하지 않고 하위 바이트, 상위 바이트 모두 명시해야 합니다.

```

void loop() {
// Make syncwrite packet
Dxl.setTxPacketId(BROADCAST_ID); 1
Dxl.setTxPacketInstruction(INST_SYNC_WRITE); 2
Dxl.setTxPacketParameter(0, P_GOAL_POSITION_L); 3
Dxl.setTxPacketParameter(1, 2); 4

for( i=0; i<NUM_ACTUATOR; i++){
    Dxl.setTxPacketParameter(2+3*i, id[i]); 5
    Dxl.setTxPacketParameter(2+3*i+1, Dxl.getLowByte(GoalPos)); 6
    Dxl.setTxPacketParameter(2+3*i+2, Dxl.getHighByte(GoalPos)); 6

    SerialUSB.println(GoalPos); 7
}
Dxl.setTxPacketLength((2+1)*NUM_ACTUATOR+4); 8
Dxl.txrxPacket(); 9

CommStatus = Dxl.getResult();
//SerialUSB.print("CommSatus = ");SerialUSB.println(CommStatus);
if( CommStatus == COMM_RXSUCCESS ){
    PrintCommStatus(CommStatus);
}
else{
    PrintErrorCode();
}

GoalPos += 100; Report result of CommStatus

if( GoalPos > MAX_POSITION )
    GoalPos -= MAX_POSITION;
delay(CONTROL_PERIOD);

}

```

예제에서는 E-manual과는 다르게 Parameter가 1부터가 아닌 0부터 시작함에 유의하세요.

1번: Syncwrite Packet의 첫머리는 반드시 Broadcast ID로 시작합니다.

2번: Instruction Sync Write를 의미하는 0x83을 나타냅니다.

3번: Parameter 0번을 의미하는 Goal Position에 대한 컨트롤 테이블 주소입니다.

4번: Goal Position만 제어하기 때문에 2바이트(1워드)를 지정합니다.

5번: 제어하려는 ID와 Parameter 번호를 지정합니다. 계산식은 아래를 참조하세요
(데이터 길이 +1)*(인덱스값 i=0,1,2,...) + 2(BROADCAST_ID, INST_SYNC_WRITE)

6번: Goal Position에 대한 위치값을 하위 1바이트 상위 1바이트 순서로 지정합니다.
워드 단위의 데이터이지만 반드시 하위, 상위 바이트로 나누어서 접근합니다.

7번: 목표 위치값을 USB를 통해 출력합니다.

8번: Packet의 길이를 계산합니다. 계산식은 아래와 같습니다.

Length (L+1) X N + 4 (L:RX-64별 Data Length, N:RX-64의 개수)

9번: 만들어진 Packet을 실제로 Dxl.txrxPacket() 메소드 수행 후에 전송이 됩니다.

C. 데이터 확인

SerialUSB의 예제와 같이 ROBOTIS OpenCM SW의 시리얼 모니터를 통해 바로 다이나믹셀의 GoalPos[INDEX]값과 현재 위치값을 확인하실 수 있습니다.

