

RPino GOGO PLUS

사용자설명서 v1.0

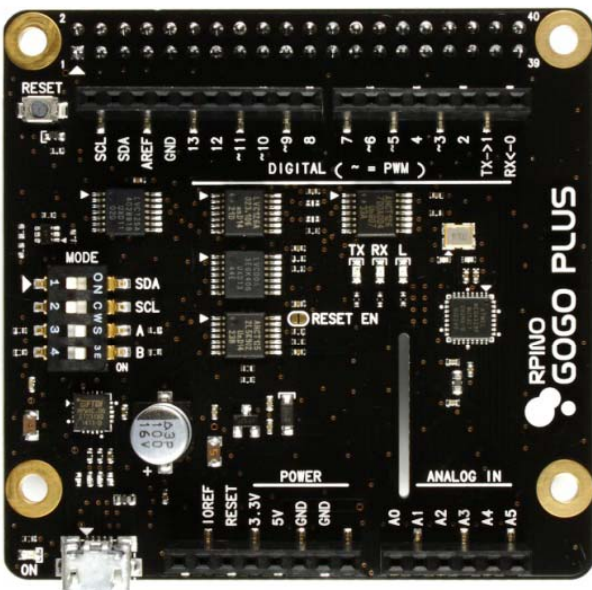


Copyright © NulSom Inc. All Rights Reserved.

■ 제품특징 및 사양

- 라즈베리파이(RaspberryPi Model B+)에 장착하여 PC 없이 사용 가능한 확장 보드
- 아두이노(Arduino) 부트로더 내장
- 라즈베리파이를 통한 아두이노 스케치 업로드 가능 (Arduino IDE 사용 가능)
- Micro USB B 포트 제공 (스마트폰 Micro 5핀 케이블 사용 가능)
 - 스케치 업로드 (단독 사용 가능)
 - 라즈베리파이, RPino GOGO PLUS 전원 공급
- 아두이노 / 라즈베리파이 Model B+ GPIO 핀 배열 호환
- 간편한 통신 모드 설정 지원
 - I2C Enable / Disable (RPI ↔ RPino)
 - UART Mode Select (All open, PC ↔ RPi, PC ↔ RPino, RPino ↔ RPi)
- 상태 표시 LED 지원 (POWER, LOAD, RX, TX)
- 전원 보호용 퓨즈 장착 (라즈베리파이 1100mA / RPino GOGO PLUS 750mA)
- 자동 전압 레벨 변환 회로 채용으로 안정성 향상 (3.3V ↔ 5V 신호레벨)
- 라즈베리파이와 결합시에도 RPi-Camera 사용 용이

제품명	RPino GOGO PLUS
MCU	ATmega328P
동작 속도	16MHz (External Crystal)
동작 전압	5V
컨넥터	Micro USB B Type
Digital I/O 핀	14 Pin
Analog Input 핀	6 Pin
Flash Memory	32 KB - 0.5 KB used by bootloader
스위치	MODE, RESET



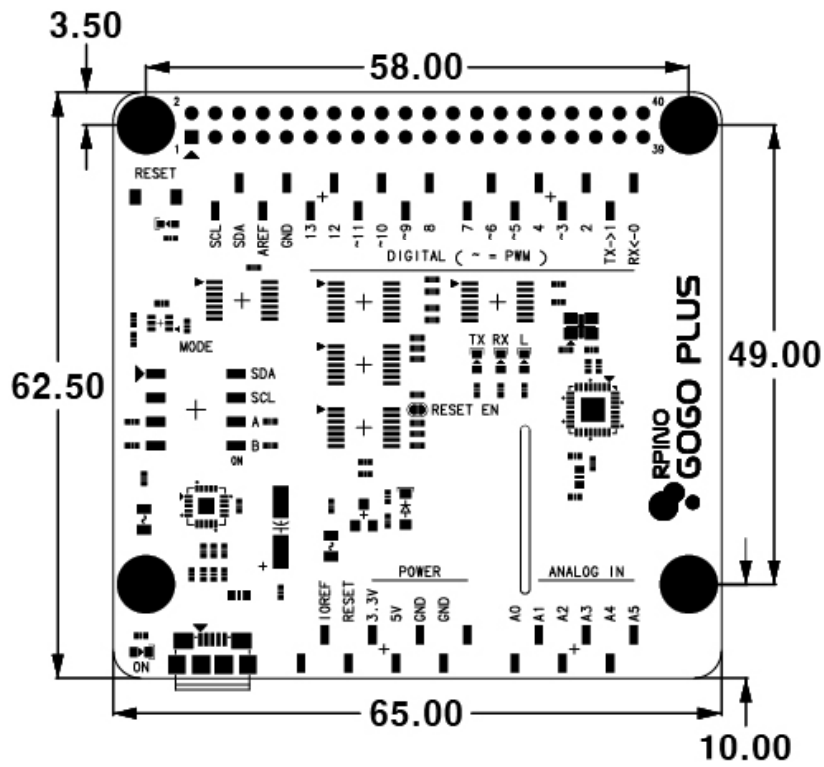
< RPino GOGO PLUS 앞면 >



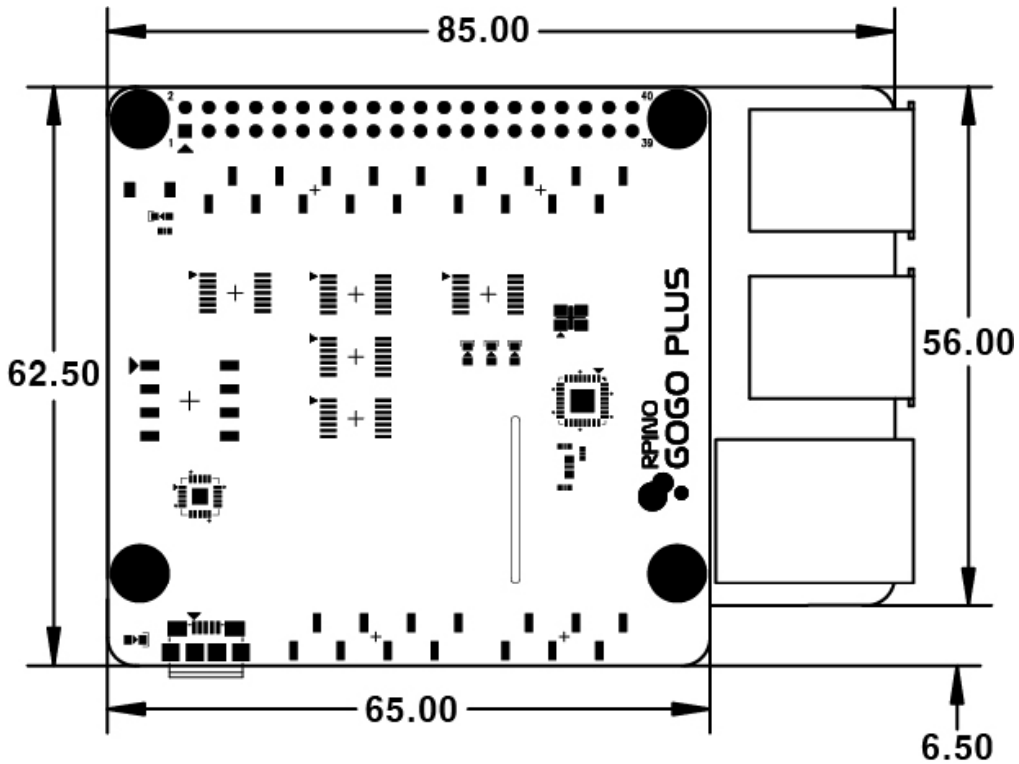
< RPino GOGO PLUS 뒷면 >

■ 제품크기 정보

(단위 : mm)

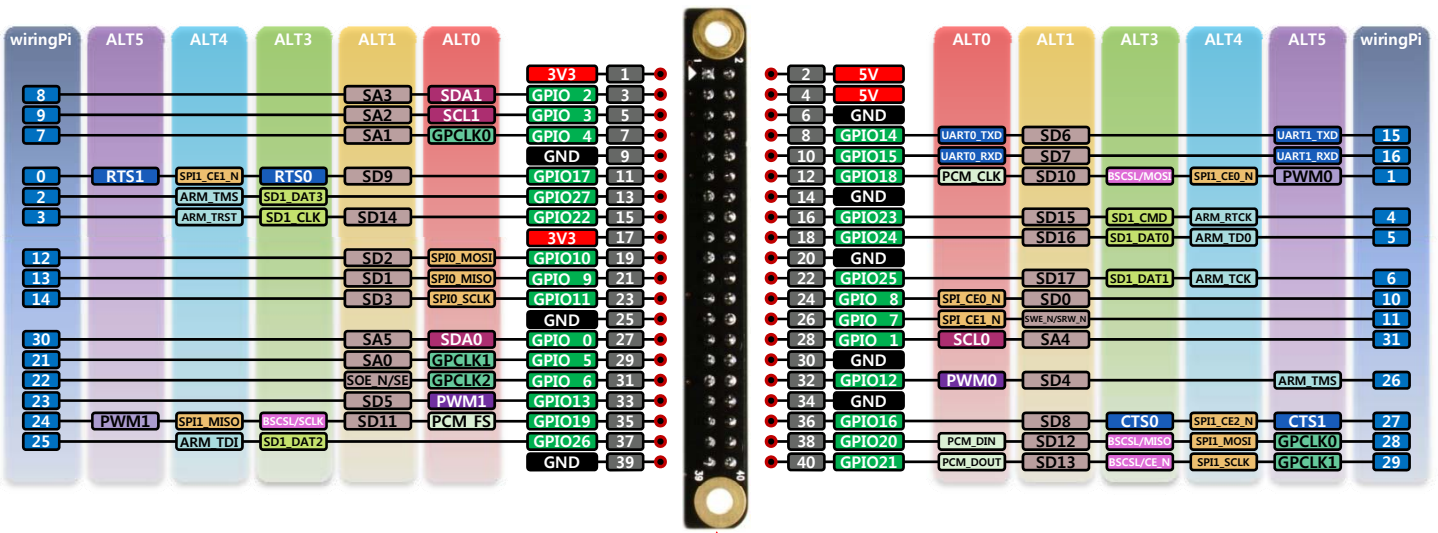


< RPino GOGO PLUS >

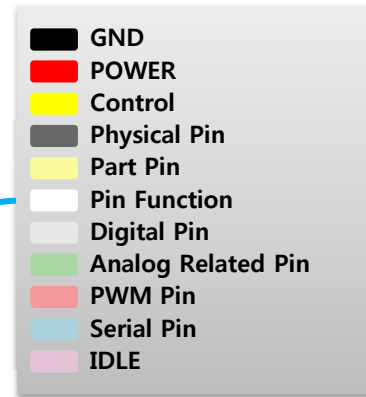


< Raspberry Pi에 RPino GOGO PLUS 결합시 >

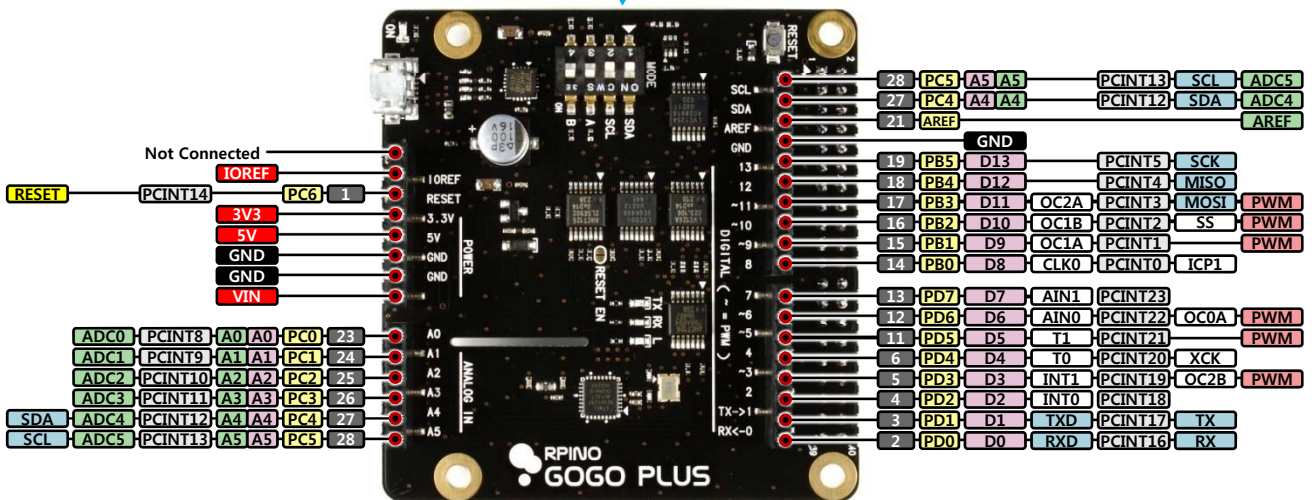
■ 핀배열



라즈베리파이 Model B+

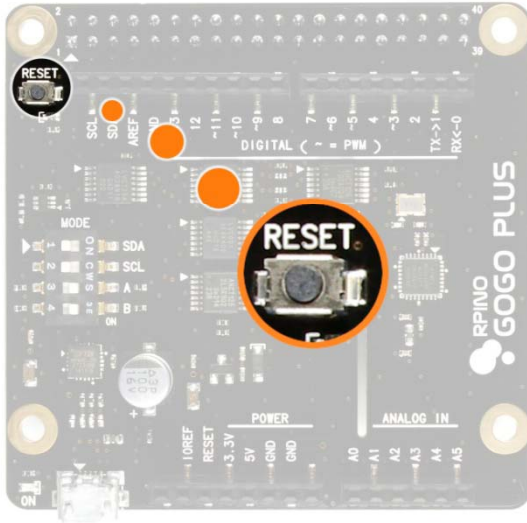


아두이노



■ 스위치별 기능

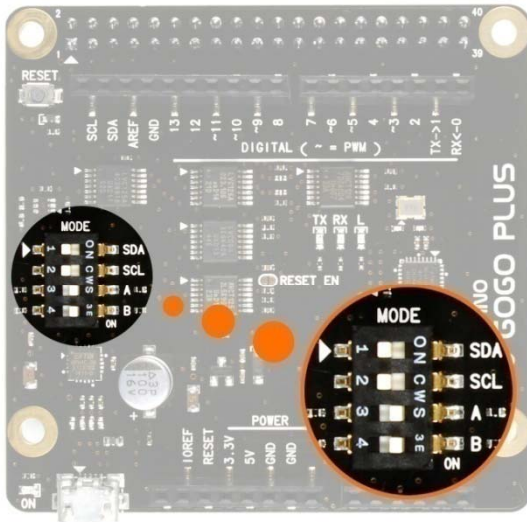
1. RESET 스위치



- **RESET 스위치 PUSH 동작**
- RPino GOGO PLUS Reset

※ **RPino GOGO PLUS의 MCU만 RESET 하기 때문에 라즈베리 파이의 작동과는 무관합니다.**

2. MODE 스위치



No.	핀 이름	기능
1	SDA	I2C Enable
2	SCL	
3	A	UART Mode Select
4	B	

- 스위치를 사용하여 기능 ENABLE / DISABLE
- 스위치 왼쪽으로 이동시 동작 : **OFF**
- 스위치 오른쪽으로 이동시 동작 : **ON**

- I2C - 라즈베리파이와 RPino GOGO PLUS I2C 연결 ON/OFF

- A, B 스위치 UART 연결 설정

모드 명	A	B	내용
MODE 0	0	0	모든 UART 통신 비활성화
MODE 1	0	1	PC ↔ Raspberry Pi
MODE 2	1	0	PC ↔ RPino GOGO PLUS
MODE 3	1	1	Raspberry Pi ↔ RPino GOGO PLUS

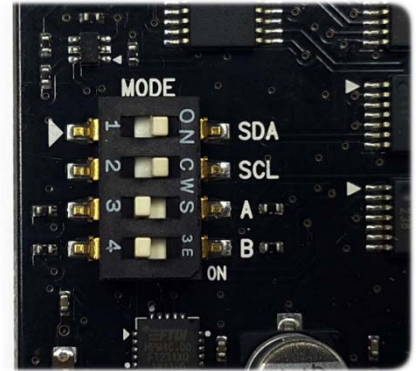
■ MODE 스위치 설정

1. I2C 통신 설정

- SDA, SCL을 ON 상태로 하여 I2C Enable

< MODE 스위치 >

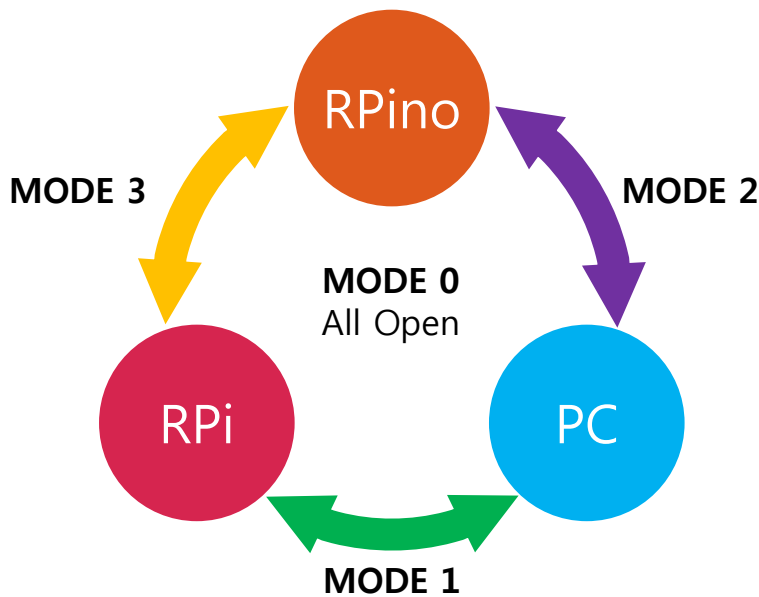
	OFF	ON		
1			SDA	ON
2			SCL	ON
3	X		A	Don't Care
4	X		B	Don't Care



- Arduino용 5V 레벨 I2C 디바이스 사용 가능
 - DAC / ADC
 - 초음파 센서
 - 온도 센서 등

2. UART 통신 설정

- 3-상태 버퍼 사용으로 오픈 시 포트 상호간섭을 없앴



- **MODE 0 : 모든 UART 통신 비활성화**
 - 연결되어 있는 모든 UART 통신 모드를 비활성화 시킵니다.
(아두이노 핀에 있는 TX->1, RX<-0 핀은 사용자의 스케치 프로그램에 따라 사용 가능)
- **MODE 1 : PC ↔ Raspberry Pi**
 - RPino GOGO PLUS에 부착된 Micro USB B 컨넥터를 이용하여 PC와 Raspberry Pi의 UART 통신을 연결 합니다.
- **MODE 2 : PC ↔ RPino GOGO PLUS**
 - Micro USB B 컨넥터를 이용하여 PC와 RPino GOGO PLUS의 UART 통신을 연결합니다.
- **MODE 3 : Raspberry Pi ↔ RPino GOGO PLUS**
 - Raspberry Pi Model B+의 GPIO 핀을 이용하여 RPi와 RPino GOGO PLUS의 UART 통신을 연결합니다.

• **MODE 0 – 모든 UART 통신 비활성화**

< MODE 스위치 >

	OFF	ON		
1	X		SDA	Don't Care
2	X		SCL	Don't Care
3			A	OFF
4			B	OFF

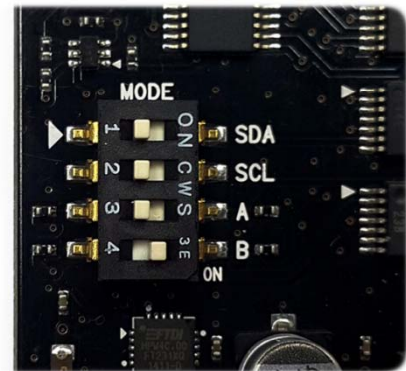


• **MODE 1 – PC ↔ Raspberry Pi**

- USB포트를 이용한 라즈베리파이 UART 모드 (PC를 이용한 시리얼 모니터)

< MODE 스위치 >

	OFF	ON		
1	X		SDA	Don't Care
2	X		SCL	Don't Care
3			A	OFF
4			B	ON



• **MODE 2 – PC ↔ RPino GOGO PLUS**

- USB포트를 이용한 RPino GOGO PLUS UART 모드 (PC를 이용한 스케치 업로드 또는 시리얼 모니터)

< MODE 스위치 >

	OFF	ON		
1	X		SDA	Don't Care
2	X		SCL	Don't Care
3			A	ON
4			B	OFF



※ 최초 사용 및 장치 드라이버 인식 실패시 17페이지를 참고하세요.

• **MODE 3 – Raspberry Pi ↔ RPino GOGO PLUS**

- 라즈베리파이와 RPino GOGO PLUS간의 UART 모드 (RPi를 이용한 스케치 업로드 또는 시리얼 모니터)

< MODE 스위치 >

	OFF	ON		
1	X		SDA	Don't Care
2	X		SCL	Don't Care
3			A	ON
4			B	ON



※ 최초 사용시 개발 환경 구축은 9페이지를 참고하세요.

■ 라즈베리파이와 결합하여 사용



1. 라즈베리파이 2x20 GPIO 컨넥터에 RPino GOGO PLUS 컨넥터를 연결
2. 이용 방법에 따라 MODE 스위치를 설정
 - 라즈베리파이를 이용한 스케치 업로드
 - MODE 3 (Raspberry Pi ↔ RPino GOGO PLUS)
 - PC를 이용한 스케치 업로드
 - MODE 2 (PC ↔ RPino GOGO PLUS)



3. Micro USB 케이블을 RPino GOGO PLUS USB 컨넥터에 연결
 - ※ 연결시 라즈베리파이의 전원도 함께 공급됩니다.
 - * 스마트폰 Micro 5핀 USB 케이블 사용 가능 (5V / 2A 이상 어댑터 사용 권장)
4. 사용 환경에 따라 9페이지의 “개발 환경 구축”을 참고하여 Arduino 설치 및 RPino 설정

■ RPino GOGO PLUS 단독 사용 (PC를 이용한 스케치 업로드)



1. RPino GOGO PLUS을 서포트로 고정 (권장)
 - 10mm이상의 서포트 사용 권장

Female Header
(H : 8.5mm)

서포트 (H : 10mm)



2. PC 스케치 업로드를 위한 MODE 스위치 설정
 - MODE 2 (PC ↔ RPino GOGO PLUS)



3. Micro USB 케이블을 RPino GOGO PLUS USB 컨넥터에 연결
4. 17페이지의 “개발 환경 구축”을 참고하여 장치 드라이버 및 “Arduino IDE” 설치

■ 개발 환경 구축 (라즈베리파이 – Arduino IDE)

※ 진행하기 전에 반드시 인터넷이 연결 되어 있어야 합니다.

• arduino 패키지(Arduino IDE) 설치 및 RPino 설정

1. 라즈베리파이 업데이트

```
pi@raspberrypi ~ $ sudo apt-get update
pi@raspberrypi ~ $ sudo apt-get upgrade
```

2. arduino 패키지(Arduino IDE) 설치

```
pi@raspberrypi ~ $ sudo apt-get install arduino
```

3. RPino 설정 파일 다운로드

```
pi@raspberrypi ~ $ wget http://nulsom.com/source/rpino.tar.gz
```

4. 설정 파일 압축 해제

```
pi@raspberrypi ~ $ tar xvfz rpino.tar.gz
```

5. 디렉토리 이동 및 RPino 설정

```
pi@raspberrypi ~ $ cd rpino
pi@raspberrypi ~/rpino $ ./setup_rpino
[ ok ] the original "/usr/bin/avrdude" file backup completed
[ ok ] the original "/usr/bin/avrdude.org" file copy completed
[ ok ] an "/usr/bin/autoreset" file copy completed
[ ok ] an "/usr/bin/avrdude-autoreset" file copy completed
[ ok ] avrdude file link completed
[ ok ] an "/etc/udev/rules.d/80-rpino.rules" file copy completed
[ ok ] the original "/boot/cmdline.txt" file backup completed
[ ok ] an "/boot/cmdline.txt" file copy completed
[ ok ] the original "/etc/inittab" file backup completed
[ ok ] an "/etc/inittab" file copy completed
[ ok ] the original "/usr/share/arduino/hardware/arduino/boards.txt"
file backup completed
[ ok ] an "/usr/share/arduino/hardware/arduino/boards.txt" file copy
completed
pi@raspberrypi ~/rpino $ _
```

※ 설정 과정 중 [warning] 메시지는 .org 파일이 존재하는 경우 발생 (이전 백업 파일 존재)

6. /etc/inittab 파일 수정

※ 소스 편집기는 vi 대신 nano를 사용하셔도 됩니다.

```
pi@raspberrypi ~/rpino $ sudo vi /etc/inittab
```

- 하단에 위치한 다음 내용을 수정

수정 전

```
#Spawn a getty on Raspberry Pi serial line
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

T0:23:respawn 앞쪽에 #을 추가하여 주석처리

수정 후

```
#Spawn a getty on Raspberry Pi serial line
#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

7. /boot/cmdline.txt 파일 수정

```
pi@raspberrypi ~/rpino $ sudo vi /boot/cmdline.txt
```

수정 전

```
dwc_otg.lpm_enable=0 console=ttyAMA0, 115200 console=tty1
root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
```

console=ttyAMA0, 115200 부분을 삭제

※ 라즈베리파이의 커널 버전에 따라 다음과 같이 표시 될 수 있으며 해당 내용을 모두 지워주세요.
console=ttyAMA0, 115200 kgdboc=ttyAMA0,115200

수정 후

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4
elevator=deadline rootwait
```

8. 라즈베리파이 재부팅

```
pi@raspberrypi ~ $ sudo shutdown -r now
```

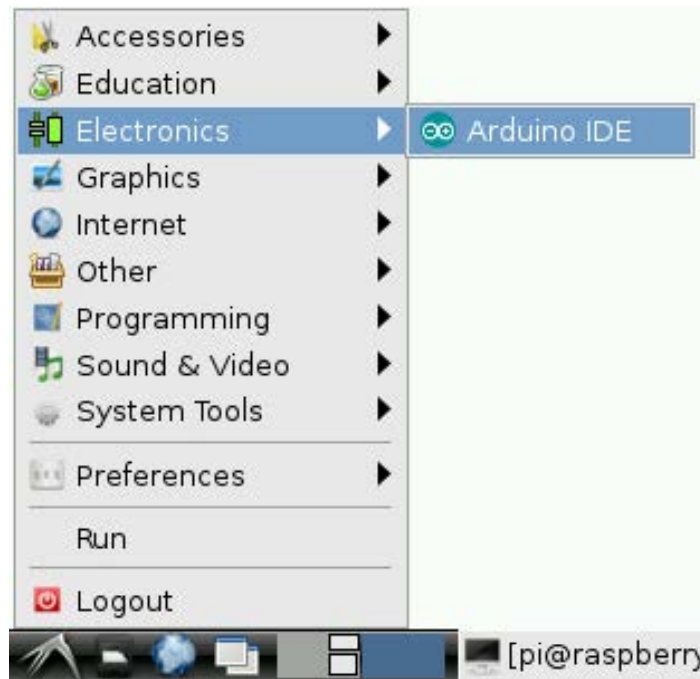
• Arduino IDE 사용 방법

1. 부팅 후 X-window 실행

```
pi@raspberrypi ~ $ startx
```

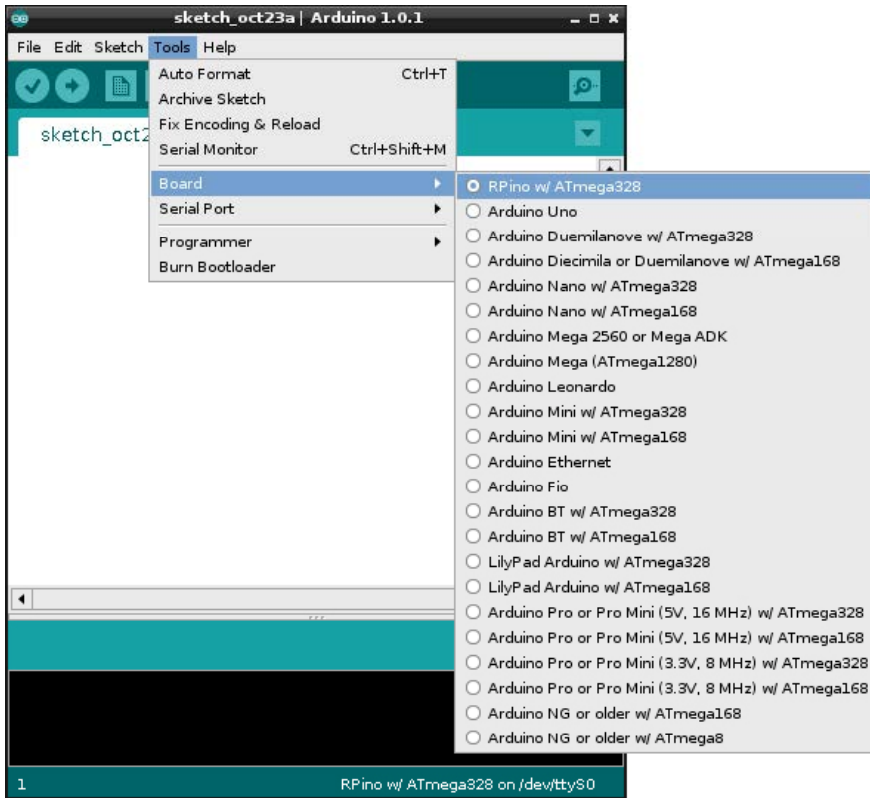
2. Arduino IDE 실행

-  버튼 클릭 > Electronics > Arduino IDE 선택



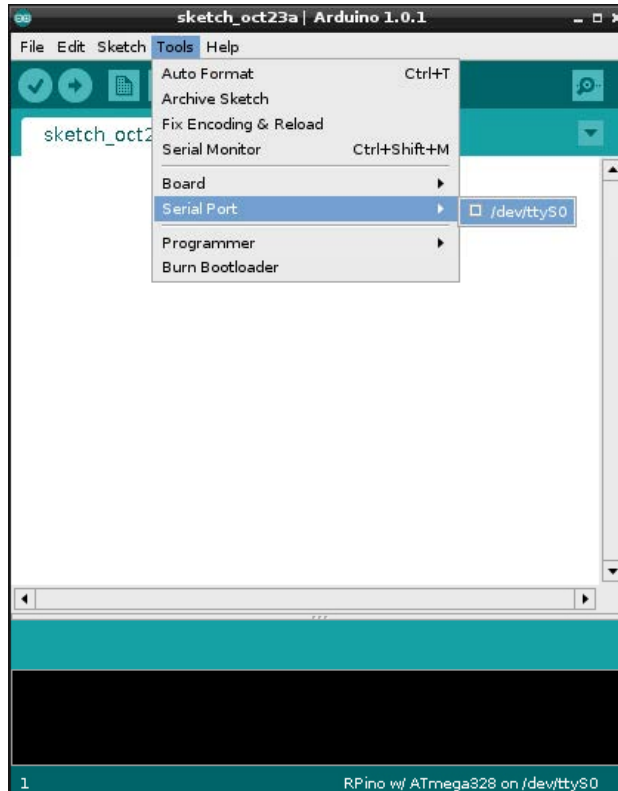
3. RPino GOGO PLUS 보드설정

- Tools > Board > RPino w/ ATmega328 선택



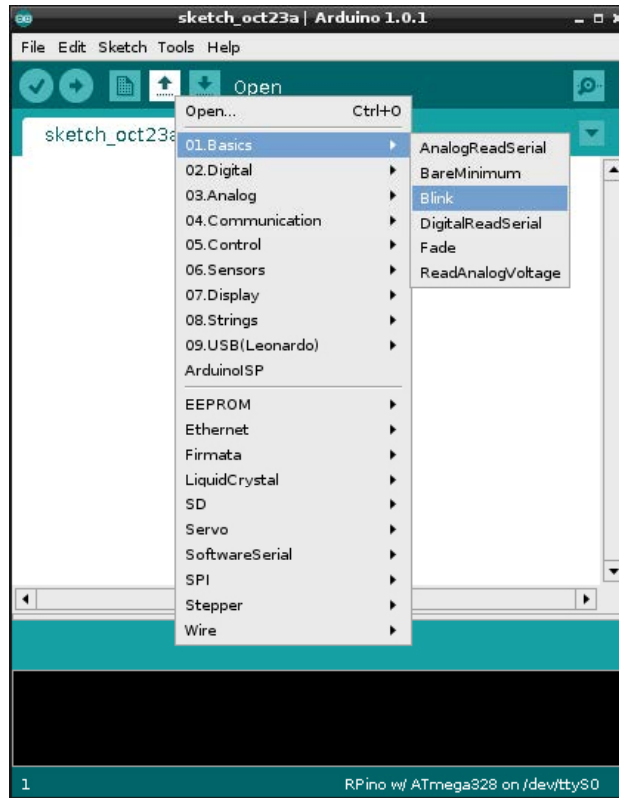
4. 시리얼포트 및 프로그래머 선택

- Tools > Serial Port > /dev/ttyS0 선택




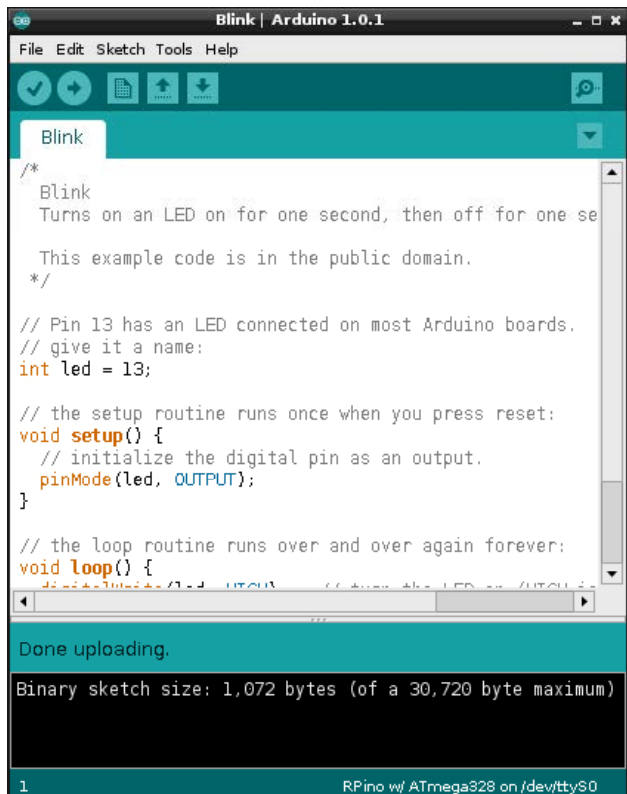
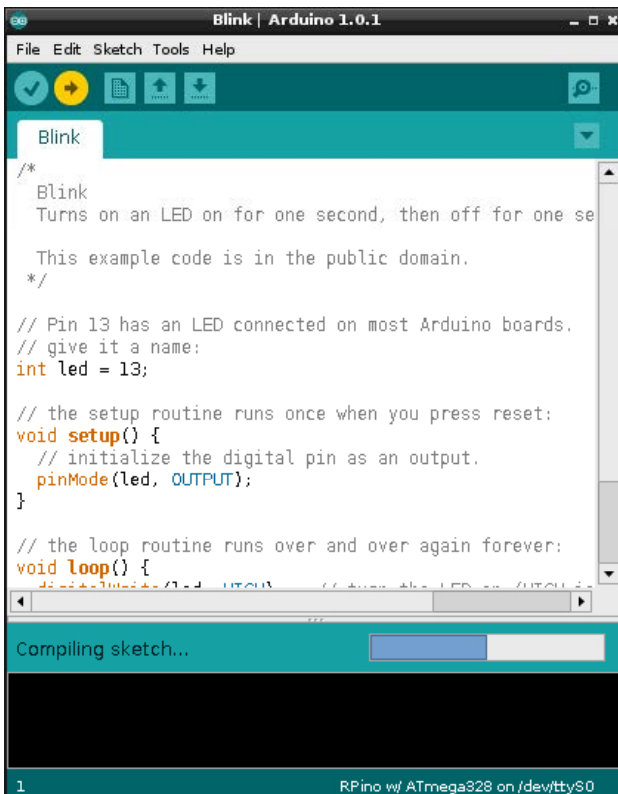
5. Blink 예제 프로그램 선택

-  아이콘 클릭 > 01.Basics > Blink 예제 선택



6. 프로그램 컴파일 및 업로드

-  아이콘 클릭 (컴파일 및 업로드)
- 상태 바 상단에 **Done uploading** 문구가 출력되면 정상적으로 업로드 완료



■ 개발 환경 구축 (라즈베리파이 – Command Line)

※ 진행하기 전에 반드시 인터넷이 연결 되어 있어야 합니다.

• arduino(Arduino IDE) & arduino-mk 패키지 설치 및 RPino 설정

1. 라즈베리파이 업데이트

```
pi@raspberrypi ~ $ sudo apt-get update
pi@raspberrypi ~ $ sudo apt-get upgrade
```

2. arduino(Arduino IDE) & arduino-mk 패키지 설치

```
pi@raspberrypi ~ $ sudo apt-get install arduino arduino-mk
```

※ GUI환경이 아닌 Command Line에서만 사용시에는 다음과 같이 설치 하시면 됩니다.

```
pi@raspberrypi ~ $ sudo apt-get install arduino-core arduino-mk
```

3. RPino 설정 파일 다운로드

```
pi@raspberrypi ~ $ wget http://nulsom.com/source/rpino.tar.gz
```

4. 설정 파일 압축 해제

```
pi@raspberrypi ~ $ tar xvfz rpino.tar.gz
```

5. 디렉토리 이동 및 RPino 설정

```
pi@raspberrypi ~ $ cd rpino
pi@raspberrypi ~/rpino $ ./setup_rpino
[ ok ] the original "/usr/bin/avrdude" file backup completed
[ ok ] the original "/usr/bin/avrdude.org" file copy completed
[ ok ] an "/usr/bin/autoreset" file copy completed
[ ok ] an "/usr/bin/avrdude-autoreset" file copy completed
[ ok ] avrdude file link completed
[ ok ] an "/etc/udev/rules.d/80-rpino.rules" file copy completed
[ ok ] the original "/boot/cmdline.txt" file backup completed
[ ok ] an "/boot/cmdline.txt" file copy completed
[ ok ] the original "/etc/inittab" file backup completed
[ ok ] an "/etc/inittab" file copy completed
[ ok ] the original "/usr/share/arduino/hardware/arduino/boards.txt"
file backup completed
[ ok ] an "/usr/share/arduino/hardware/arduino/boards.txt" file copy
completed
pi@raspberrypi ~/rpino $ _
```

※ 설정 과정 중 [warning] 메시지는 .org 파일이 존재하는 경우 발생 (이전 백업 파일 존재)

6. /etc/inittab 파일 수정

※ 소스 편집기는 vi 대신 nano를 사용하셔도 됩니다.

```
pi@raspberrypi ~/rpino $ sudo vi /etc/inittab
```

- 하단에 위치한 다음 내용을 수정

수정 전

```
#Spawn a getty on Raspberry Pi serial line
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

TO:23:respawn 앞쪽에 #을 추가하여 주석처리

수정 후

```
#Spawn a getty on Raspberry Pi serial line
#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

7. /boot/cmdline.txt 파일 수정

```
pi@raspberrypi ~/rpino $ sudo vi /boot/cmdline.txt
```

수정 전

```
dwc_otg.lpm_enable=0 console=ttyAMA0, 115200 console=tty1
root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
```

console=ttyAMA0, 115200 부분을 삭제

※ 라즈베리파이의 커널 버전에 따라 다음과 같이 표시 될 수 있으며 해당 내용을 모두 지워주세요.
console=ttyAMA0, 115200 kgdboc=ttyAMA0,115200

수정 후

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4
elevator=deadline rootwait
```

8. 라즈베리파이 재부팅

```
pi@raspberrypi ~ $ sudo shutdown -r now
```

• arduino-mk 사용 방법

- GUI환경이 아닌 Command Line에서 컴파일 및 스케치 업로드를 할 수 있는 방법

1. 스케치 디렉토리 생성 및 소스 작성

```
pi@raspberrypi ~ $ sudo mkdir mysketch
pi@raspberrypi ~ $ sudo cd mysketch
pi@raspberrypi ~/mysketch $ sudo vi mysketch.ino
```

※ 디렉토리 명과 *.ino 파일 명이 반드시 일치해야 합니다.

- Blink 예제

```
void setup() {
    pinMode(13, OUTPUT);
}

void loop() {
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(1000);
}
```

2. Makefile 작성

```
pi@raspberrypi ~/mysketch $ sudo vi Makefile
```

- Makefile 내용 ※ RPino 설정 파일 안에 포함되어 있는 "Makefile"을 사용하셔도 됩니다.

```
ARDUINO_DIR = /usr/share/arduino
BOARD_TAG = rpino
ARDUINO_PORT = /dev/ttyS0
ARDUINO_LIBS = ''
include /usr/share/arduino/Arduino.mk
```

3. RPino GOGO PLUS에 스케치 업로드

```

pi@raspberrypi ~/mysketch $ sudo make upload
for STTYF in 'stty -F' 'stty --file' 'stty -f' 'stty <' ; \
  do $STTYF /dev/tty >/dev/null 2>&1 && break ; \
  done ; \
  $STTYF /dev/ttyS0 hupcl ; \
  (sleep 0.1 2>/dev/null || sleep 1) ; \
  $STTYF /dev/ttyS0 -hupcl
최초 컴파일의 경우 이 위치에 컴파일 메시지가 표시 됩니다. (생략)
/usr/bin/avrdude -q -V -p atmega328p -C /etc/avrdude.conf -c arduino -b
57600 -P /dev/ttyS0 \
-U flash:w:build-cli/mysketch.hex:i

avrdude-original: AVR device initialized and ready to accept instructions
avrdude-original: Device signature = 0x1e950f
avrdude-original: NOTE: FLASH memory has been specified, an erase cycle
will be performed
    To disable this feature, specify the -D option.
avrdude-original: erasing chip
avrdude-original: reading input file "build-cli/mysketch.hex"
avrdude-original: writing flash (1040 bytes):
avrdude-original: 1040 bytes of flash written

avrdude-original: safemode: Fuses OK
pi@raspberrypi ~/mysketch $ _

```

■ RPino 설정 파일 제거

- 사용자 혹은 기타 환경에 의한 오작동시 재설정 또는 RPino 설정 전 상태로 되돌리기 위한 방법

1. RPino 설정 파일 다운로드

※ 기존에 다운 받은 파일도 사용 가능합니다.

```
pi@raspberrypi ~ $ wget http://nulsom.com/source/rpino.tar.gz
```

2. 설정 파일 압축 해제

```
pi@raspberrypi ~ $ tar xvfz rpino.tar.gz
```

3. 디렉토리 이동

```
pi@raspberrypi ~ $ cd rpino
```

4. RPino 설정 파일 제거

```

pi@raspberrypi ~/rpino $ ./remove_rpino
[ ok ] remove "/usr/bin/avrdude" link file
[ ok ] remove "/usr/bin/avrdude-origianl" file
[ ok ] restore "/usr/bin/avrdude" original file
[ ok ] remove "/usr/bin/autoreset" file
[ ok ] remove "/usr/bin/avrdude-autoreset" file
[ ok ] remove "80-rpino.rules" file
[ ok ] restore "cmdline.txt" original file
[ ok ] restore "inittab" original file
[ ok ] restore "boards.txt" original file
pi@raspberrypi ~/rpino $ _

```

※ 원본 파일이 백업되어 있지 않은 경우

```
pi@raspberrypi ~/rpino $ ./remove_rpino
[ warning ] The original files not found!
May not be completely restore if you continue.
Do you want to continue? [y/n]
```

※ 위 메시지는 RPino 설정시 백업한 원본 파일 중 한 개 이상의 파일이 존재하지 않을 때 표시되고 사용자의 선택에 따라 제거 여부가 결정되며 백업이 되어있는 원본 파일만 복구 됩니다. 이 경우 동작상 오류가 발생할 가능성이 있기 때문에 페이지 하단 "arduino / arduino-core 패키지 재설치" 를 진행해 주시기 바랍니다.

예시) avrdude의 원본 파일이 백업되어 있지 않은 경우

```
Do you want to continue? [y/n] y
[ warning ] the original "/usr/bin/avrdude" file dosen't exist!
[ ok ] remove "/usr/bin/avrdude" link file
[ ok ] remove "/usr/bin/avrdude-original" file
[ ok ] remove "/usr/bin/autoreset" file
[ ok ] remove "/usr/bin/avrdude-autoreset" file
[ ok ] remove "/etc/udev/rules.d/80-rpino.rules" file
[ ok ] restore "/boot/cmdline.txt" original file
[ ok ] restore "/etc/inittab" original file
[ ok ] restore "/usr/share/arduino/hardware/arduino/boards.txt"
original file
pi@raspberrypi ~/rpino $ _
```

5. 라즈베리파이 재부팅

```
pi@raspberrypi ~ $ sudo shutdown -r now
```

■ arduino / arduino-core 패키지 재설치

1. arduino / arduino-core 패키지 삭제

- X Window 환경 또는 Arduino IDE를 설치한 사용자의 경우

```
pi@raspberrypi ~ $ sudo apt-get autoremove arduino
```

- Command Line 환경 또는 arduino-core 패키지를 설치하여 사용한 경우

```
pi@raspberrypi ~ $ sudo apt-get autoremove arduino-core
```

2 사용 환경에 따른 패키지 재설치

- X Window 환경 또는 arduino 패키지(Arduino IDE) 사용

```
pi@raspberrypi ~ $ sudo apt-get install arduino
```

- Command Line 환경 또는 arduino-core 패키지 설치

```
pi@raspberrypi ~ $ sudo apt-get install arduino-core
```

※ RPino 재설정은 최초 설정 방법과 동일하게 하시면 됩니다.

■ 개발 환경 구축 (PC – Arduino IDE)

※ Windows 7을 기준으로 작성되었습니다.

• Arduino IDE 다운로드

arduino.cc 사이트에서 상단 “Download”를 클릭하여 사용자의 운영체제에 맞게 다운로드

* 작성일 기준 버전 : 2014/10/23 – arduino-1.0.6 Version

- Windows

Installer - <http://downloads.arduino.cc/arduino-1.0.6-windows.exe>

Zip File - <http://downloads.arduino.cc/arduino-1.0.6-windows.zip>

• 장치 드라이버 다운로드

제품의 장치 드라이버는 FTDI chip 공식 홈페이지에서 다운로드 받을 수 있습니다.

www.ftdichip.com > Drivers > VCP Drivers > 운영체제에 맞는 드라이버 다운로드

* 작성일 기준 버전 : 2014/10/23 – 2.12.00 Version

- Windows

32/64Bit - <http://www.ftdichip.com/Drivers/CDM/CDM%20v2.12.00%20WHQL%20Certified.zip>

- Linux

32Bit - http://www.ftdichip.com/Drivers/VCP/Linux/ftdi_sio.tar.gz

64Bit - http://www.ftdichip.com/Drivers/VCP/Linux/ftdi_sio.tar.gz

• 장치 드라이버 및 Arduino IDE 설치

1. MODE 스위치를 “MODE 2”로 설정

< MODE 스위치 >

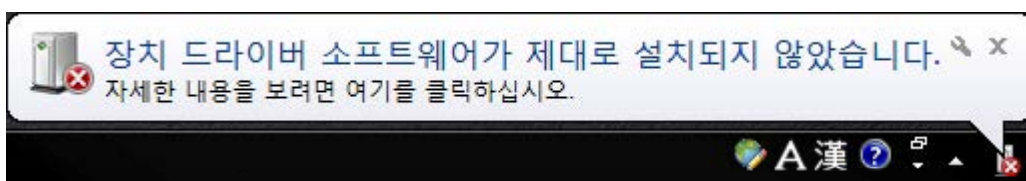
	OFF	ON		
1	X		SDA	Don't Care
2	X		SCL	Don't Care
3		X	A	ON
4	X		B	OFF



2. RPino GOGO PLUS를 PC의 USB와 연결



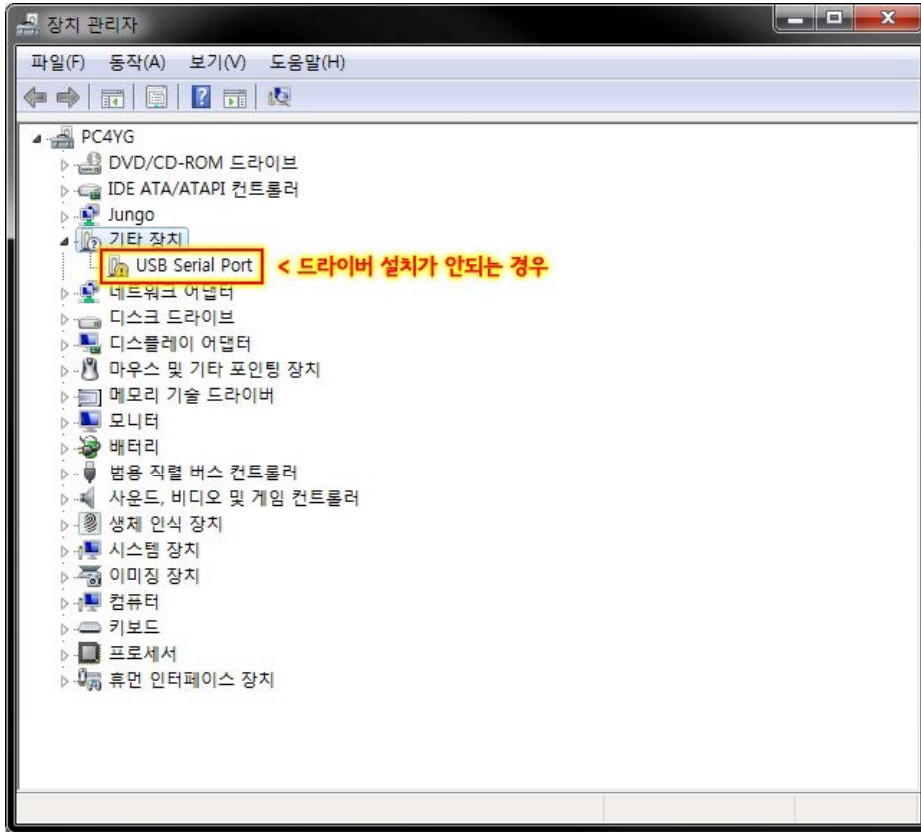
※ 위와 같이 드라이버 설치가 자동으로 되는 경우에는 따로 설치할 필요가 없습니다.



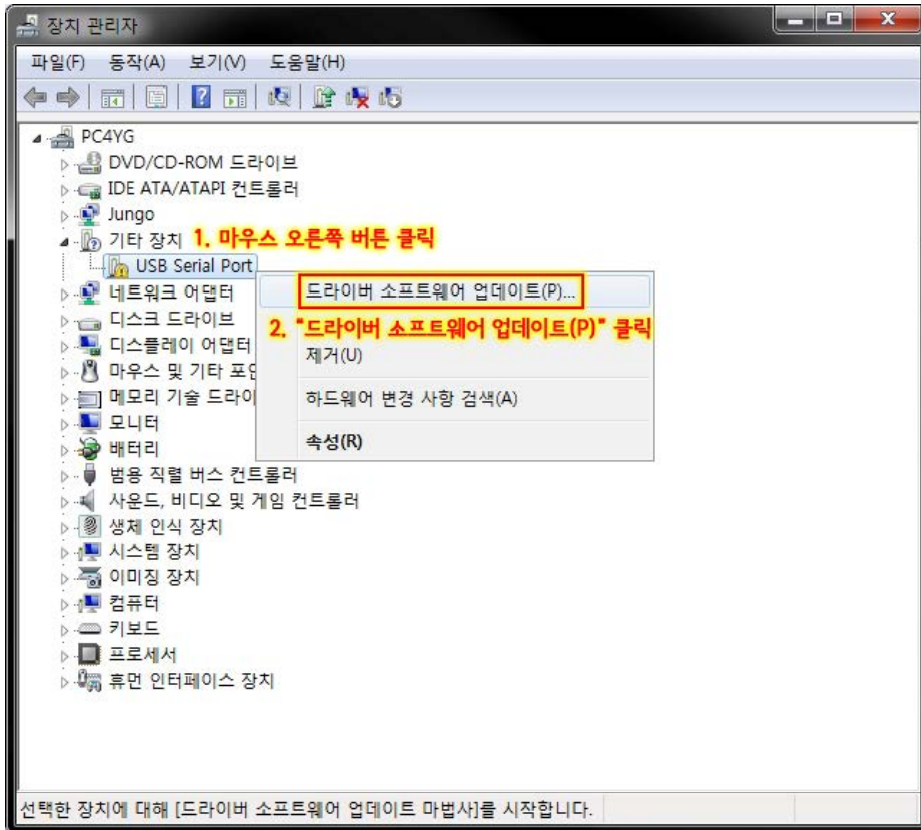
※ 자동 드라이버 설치에 실패한 경우에는 다음 페이지의 설치 방법을 참고해 주세요.

3. "작업관리자" 실행

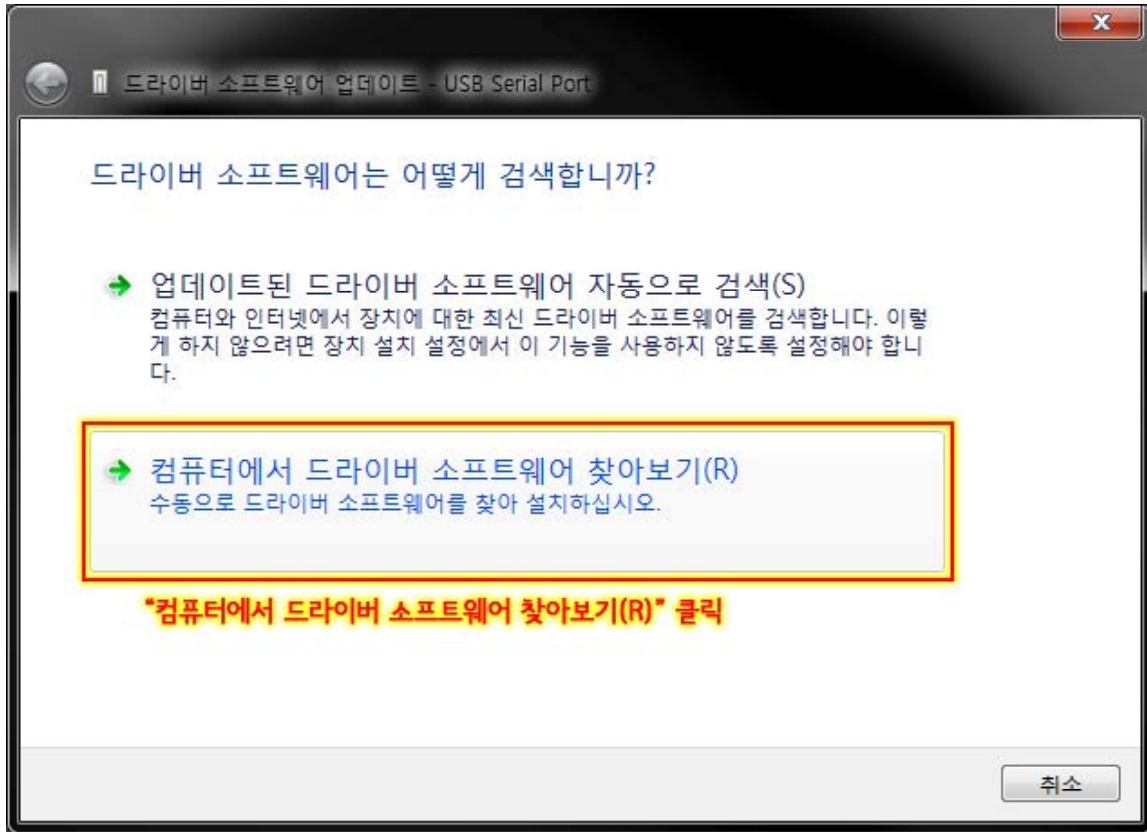
- 시작 > 실행 > "devmgmt.msc" 입력
- 시작 > 컴퓨터(오른쪽 마우스 클릭) > 속성 > "작업관리자" 클릭



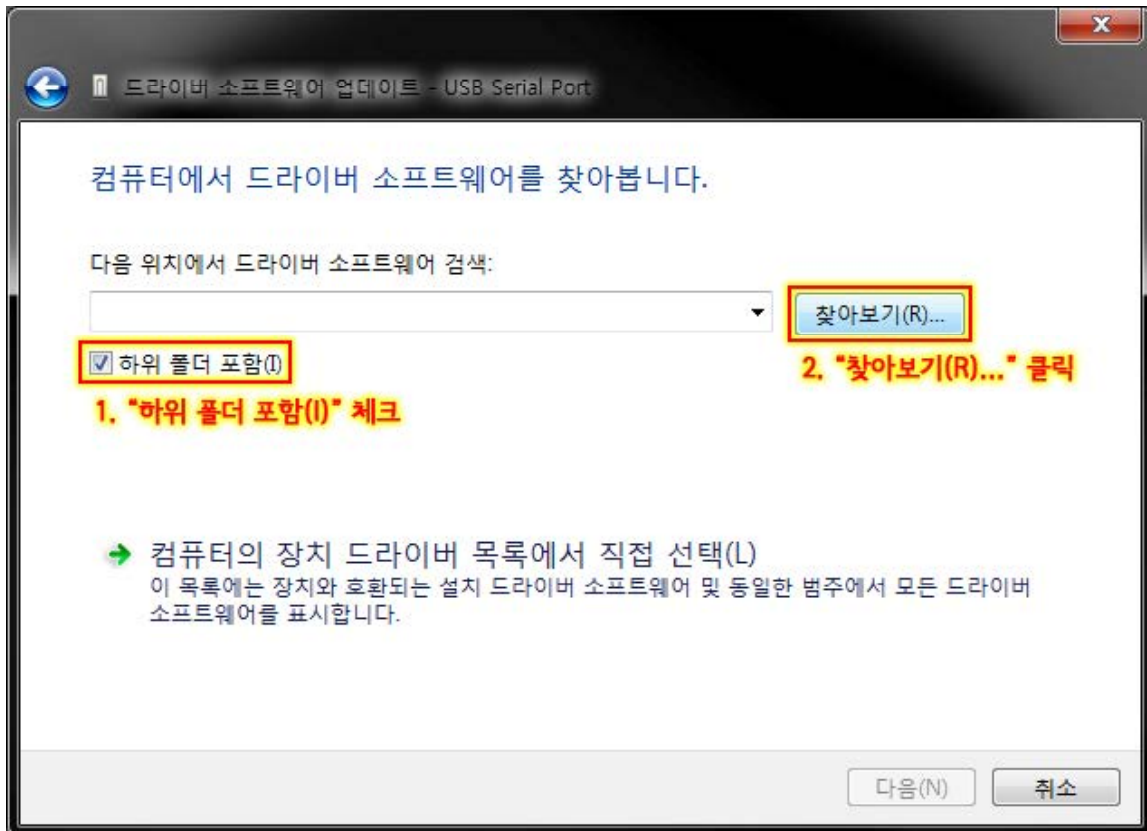
4. "드라이버 소프트웨어 업데이트(P)..." 클릭



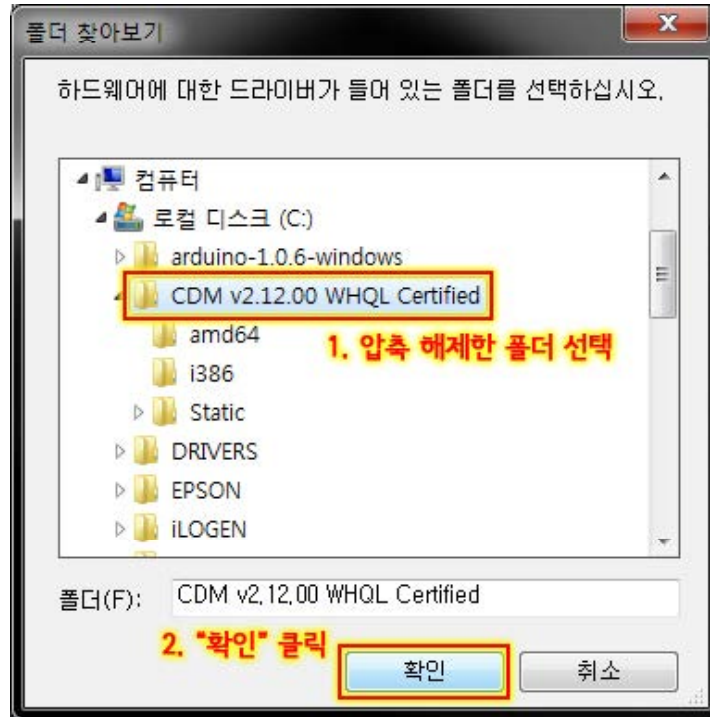
5. "컴퓨터에서 드라이버 소프트웨어 찾아보기(R)" 클릭



6. "하위 폴더 포함(I)" 체크 후 "찾아보기(R)..." 클릭

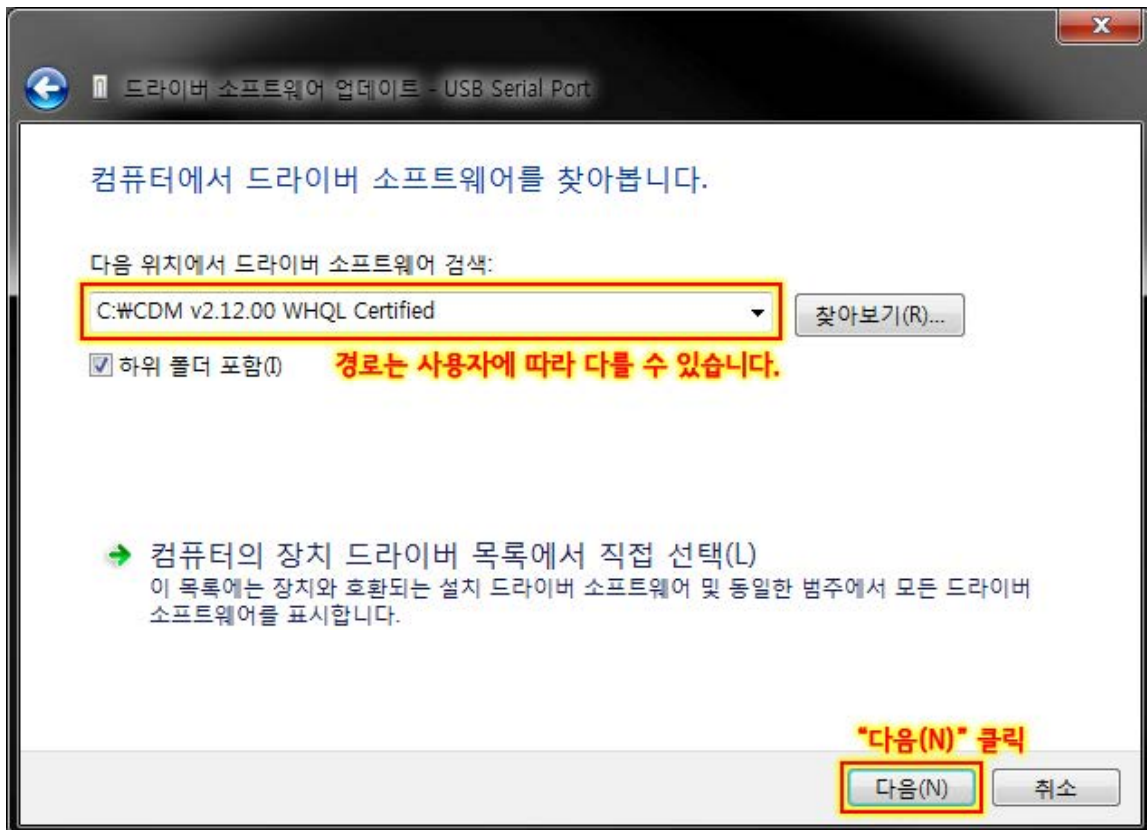


7. "압축 해제한 폴더" 선택

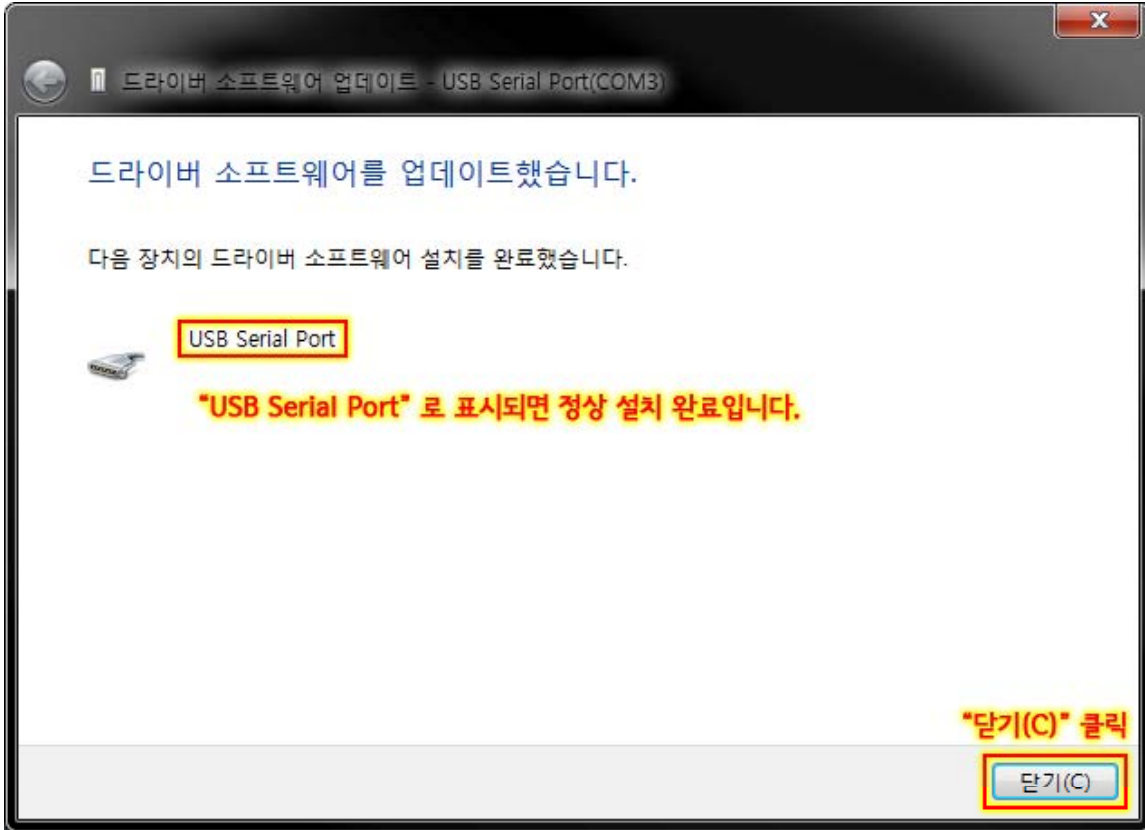


※ 폴더명, 위치는 사용자에게 따라 다를 수 있습니다.

8. "다음" 클릭



9. 정상적으로 설치가 완료되면 아래와 같은 창이 뜹니다. “닫기” 클릭

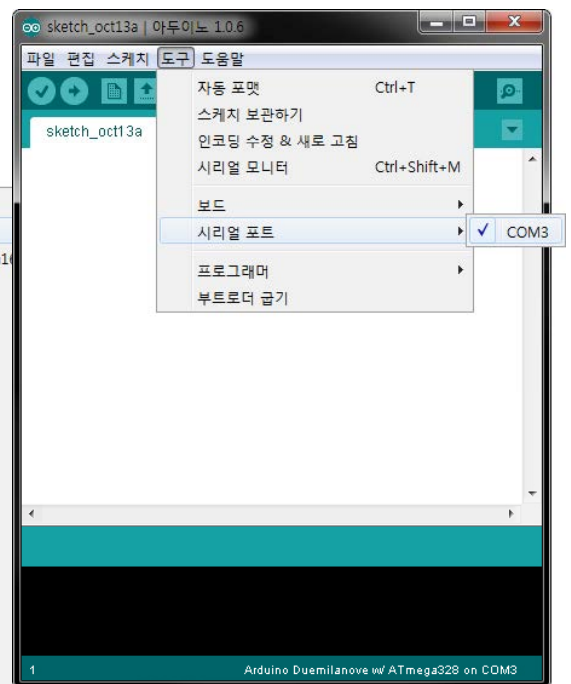
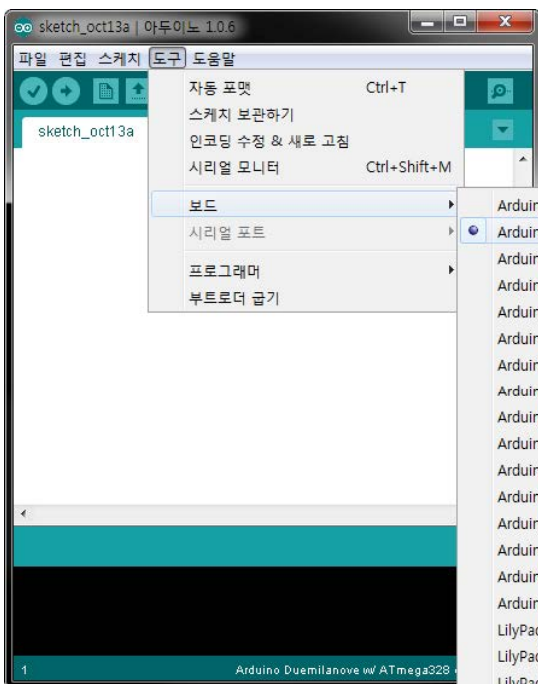


※ FTDI chip을 사용하기 때문에 포트명은 제품명이 아닌 USB Serial Port 로 표시 됩니다.



• Arduino IDE 사용 방법

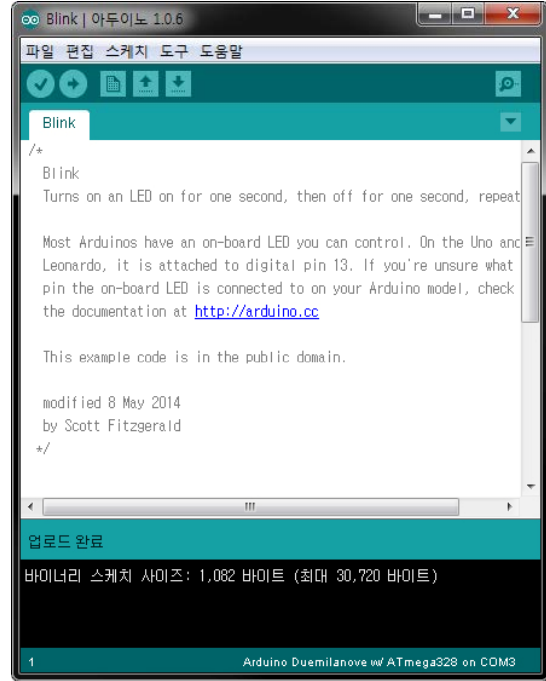
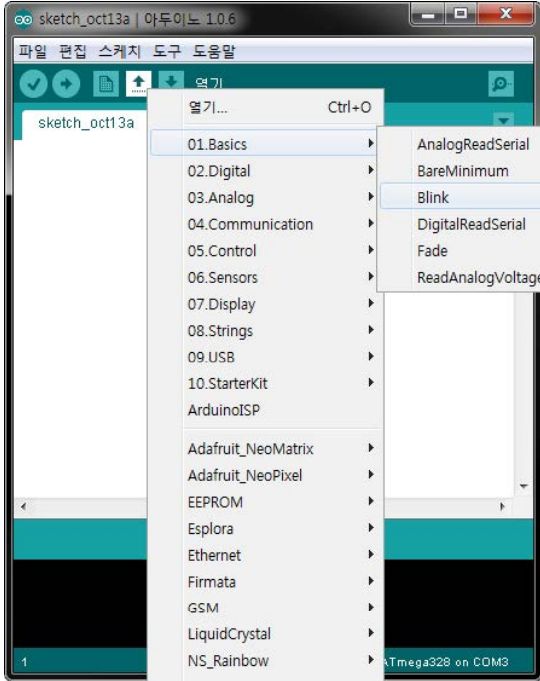
1. 보드 및 시리얼 포트 선택

- 도구 > 보드 > Arduino Duemilanove w/ ATmega328 선택
- 도구 > 시리얼 포트 > COMxx 선택 (RPino GOGO PLUS가 인식되어 있는 COM 포트를 선택)



2. Blink 예제 프로그램 컴파일 및 업로드

-  아이콘 클릭 > 01.Basics > Blink 예제 선택
-  아이콘 클릭 (컴파일 및 업로드)
- 상태 바 상단에 업로드 완료 문구가 출력되면 정상적으로 업로드 완료



■ 제품문의

주 소 : 대전광역시 유성구 테크노1로 11-3 N214호(관평동 1337)
 대표번호 : 042-710-2580
 이 메 일 : support@nulsom.com

홈페이지 : www.nulsom.com
 블 로 그 : nulsom.tistory.com