

Blueinno start kit

Copyrights © blueinno. All rights reserved

블루이노 사용하기

[기본 개념]

- 블루이노 = 아두이노 x 블루투스 4.0
- 아두이노 = 하드웨어(HW) 보드 + 통합개발 환경(IDE) + 오픈소스 그룹
 - 하드웨어보드 = (디지털 + 아날로그) x (입력 + 출력)
 - 통합개발 환경 = 스케치작업(코딩) + 컴파일 + 업로드
 - 소스코드 = 스케치 (아두이노는 소스 프로그램을 '스케치'라 합니다.)
 - 컴파일 = 스케치를 마이크로 컨트롤러가 알아듣게 바꾸는 작업
 - 업로드 = 컴파일 된 것을 USB 케이블로 아두이노 보드에 옮기는 작업
 - 오픈소스 그룹 = 블로그 + 카페 + 커뮤니티 + 행사를 통한 정보 교류
- 블루투스 4.0 = 클래식 + 하이 스피드 + 저전력의 기능을
가진 새로운 프로토콜의 블루투스
- 소스 저장하는 곳 = 스케치 북

[사용 순서]



순서1 : PC 연결 및 보드 설치하기



➤ 연결 :

그림과 같이 USB 케이블을 보드와 PC를 연결합니다.

➤ USB 장치 드라이브 (USB Serial Converter) 설치하기

- USB 케이블을 보드에 연결하면, 자동으로 PC에 설치되고
그 이후 업데이트된 내용이 반영이 되도록 PC를 다시 시작합니다.
- 만약에 자동으로 드라이브가 설치되지 않는다면, 아래의 사이트에서
다운로드 받아서 설치를 합니다.

<http://www.ftdichip.com/Drivers/VCP.htm>

Currently Supported VCP Drivers:		Processor Architecture	
Operating System	Release Date	x86 (32-bit)	x64 (64-bit)
Windows*	2014-09-29	Available as setup executable Contact support1@ftdichip.com if looking to create cusomised drivers	
Linux	2009-05-14	1.5.0	1.5.0
Mac OS X	2012-08-10	2.2.18	2.2.18
Windows CE 4.2-5.2**	2012-01-06	1.1.0.20	-
Windows CE 6.0/7.0	2012-01-06	1.1.0.20 CE 6.0 CAT CE 7.0 CAT	-

순서2-1 : PC에 SW 개발환경 만들기

➤ 아두이노 통합개발환경(IDE) 설치

- 아두이노 홈페이지에 방문하여, 해당 OS별로 다운로드 하여 설치합니다.

<http://arduino.cc/en/Main/Software>



Download the Arduino Software

Category	Links
ARDUINO 1.0.6	Windows Installer, Windows ZIP file for non admin install, Mac OS X, Linux 32 bits, Linux 64 bits, Release Notes
ARDUINO 1.5.8 BETA	Windows Installer, ZIP file non admin, Mac OS X for Java 6, for Java 7, Linux 32 bit, 64 bit, Source
ARDUINO 1.0.x / 1.5.x PREVIOUS RELEASES	Download the previous versions of the Arduino Software. Available for Windows, Linux, and Mac OS X.
ARDUINO IDE INTEL GALILEO INTEL EDISON	Download the Arduino IDE that supports the Intel Galileo and the Intel Edison boards. Available for Windows, Linux, and Mac OS X.

- 윈도우 사용자는 Arduino 1.5.x Windows Installer를 다운로드하여 원하는 위치에 설치합니다. (예: C:\Program Files(x86)\Arduino) 이 때 PC의 바탕화면에 Arduino 바로 가기가 생성됩니다.

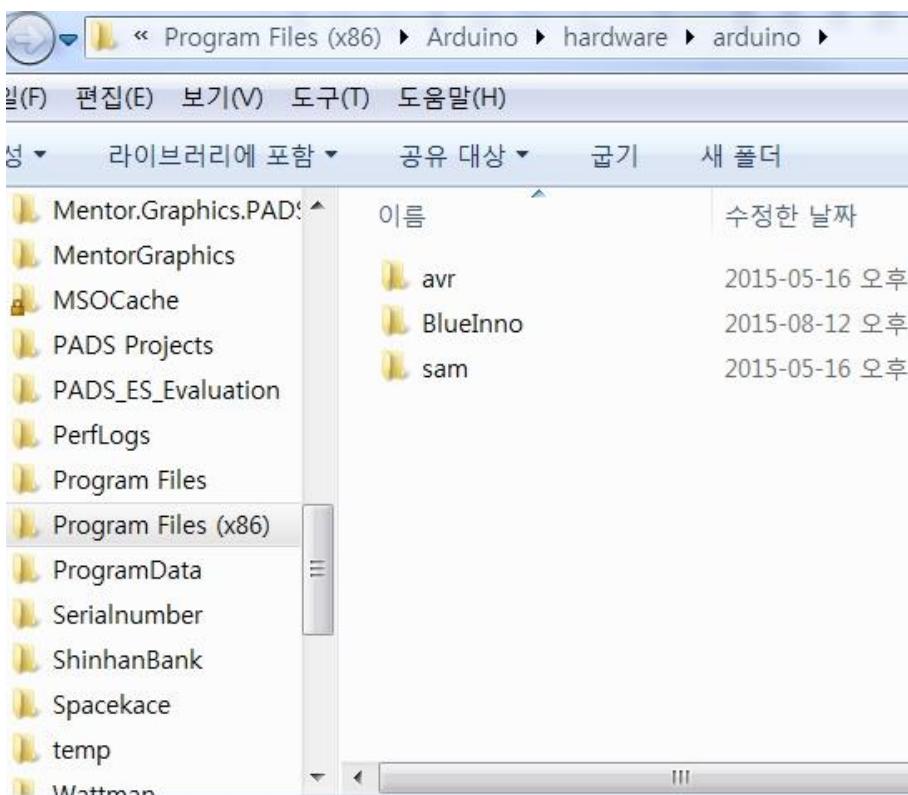
※ 주의 : 여기서 Arduino 1.0.x 버전은 사용할 수 없으며, 지속적으로 아두이노의 업그레이드로 호환이 안 될 수 있으니, 1.5.8 BETA 버전을 추천합니다.

순서2-2 : PC에 SW 개발환경 만들기

➤ 블루이노 라이브러리 설치

- 스케치에서 블루이노 예제를 활용하려면, 먼저, 블루이노 라이브러리를 다운로드 합니다.
- 라이브러리는 카페 (<http://cafe.naver.com/arduinoplusble>) 자료실에서 Blueinno.7z 파일들을 다운받아서 7Z로 압축을 풀고, 아두이노 스케치 프로그램이 설치된 **arduino** 폴더 내에 아래와 같이 설치합니다.

(예 C:\Program Files(x86)\Arduino\hardware\arduino\Blueinno)



➤ 프로젝트 예제 설치

- 카페 (<http://cafe.naver.com/arduinoplusble>) 자료실에서 BlueinnoSKPT.zip 파일들을 다운받아서 ALzip으로 압축을 풀고 아두이노 스케치 프로그램이 설치된 **libraries** 폴더 내에 설치합니다.
- (예 C:\Program Files\Arduino\libraries)

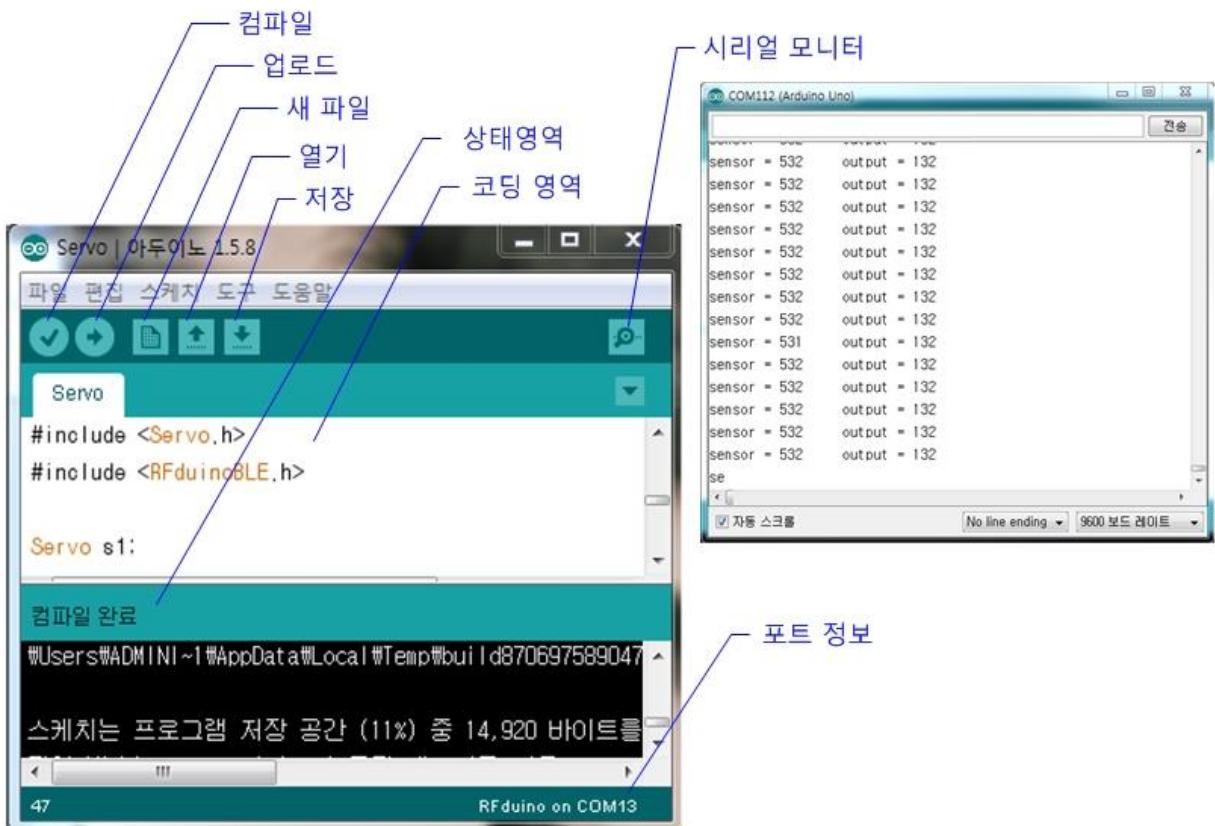
순서3-1 : SW 코딩을 위한 스케치 환경 만들기

➤ 스케치 실행 :

- 바탕화면에서 아두이노 스케치 아이콘 클릭



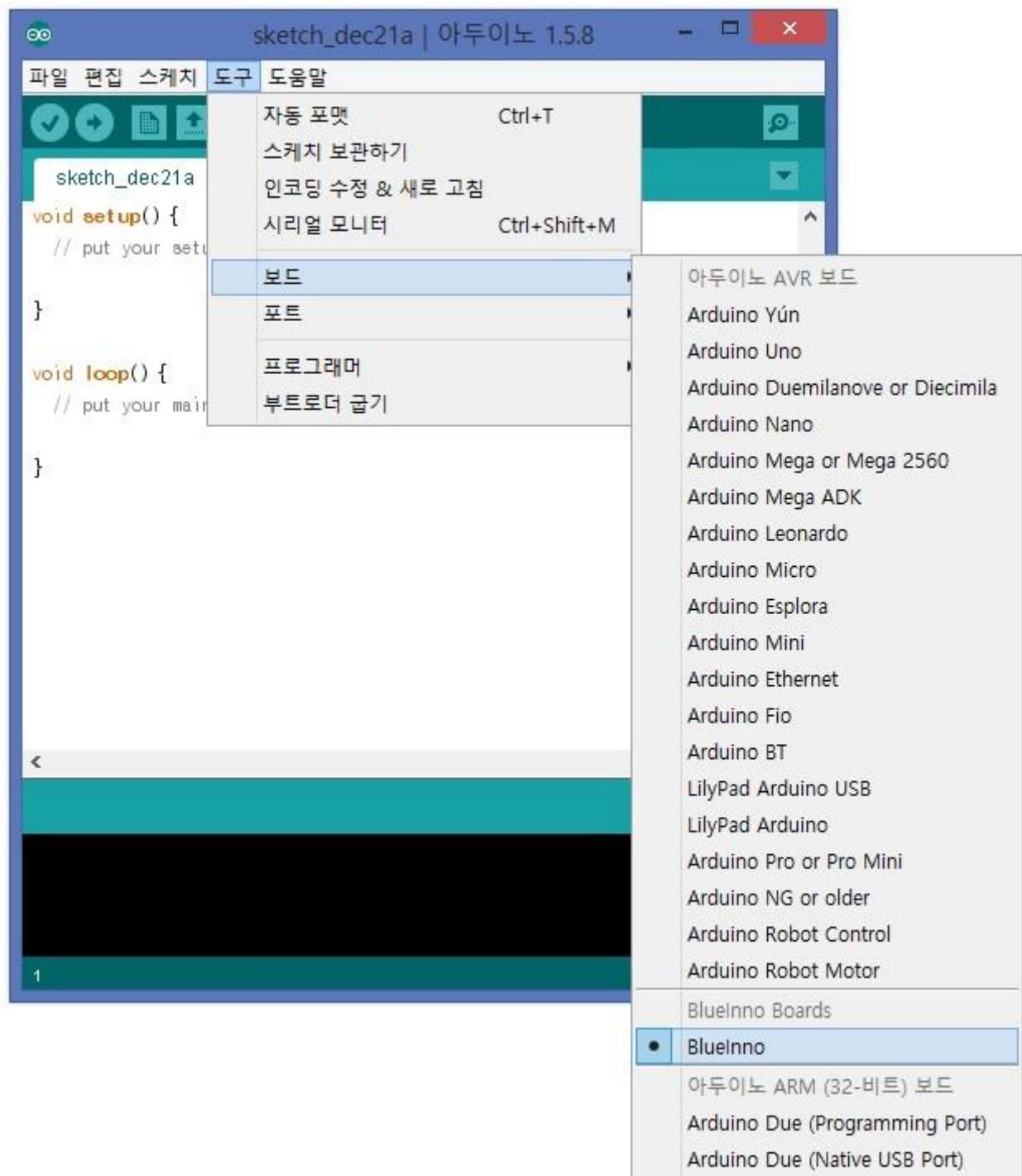
➤ 스케치 사용하기 :



- 컴파일 : 코드의 오류를 확인
- 업로드 : 보드로 코드를 보드로 업로드 함
- 새 파일 : 새로운 스케치를 만듦
- 열기 : 스케치북에서 스케치를 가져옴
- 저장 : 스케치를 저장함
- 상태영역 : 상태 정보를 표시 영역
- 코딩영역 : 스케치 직접 작성 및 편집하는 영역
- 시리얼 모니터 : 시리얼 모니터를 열어봄
- 포트 정보 : PC USB 연결된 포트 정보

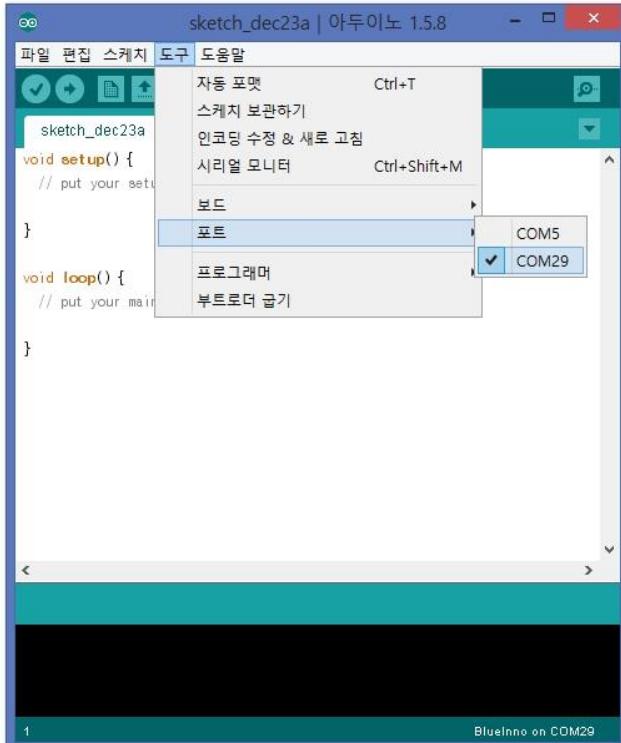
순서3-2 : SW 코딩을 위한 스케치 환경 만들기

- 보드 선택 : 도구 > 보드 > Blueinno 선택

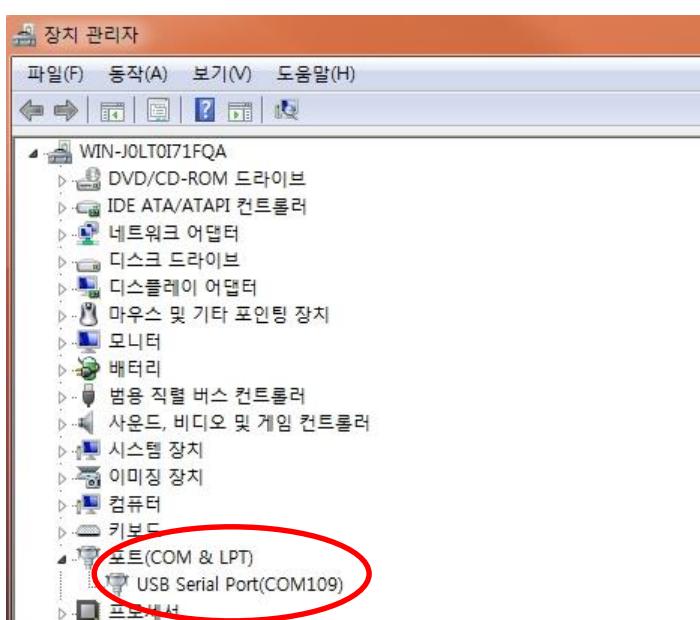


순서3-3 : SW 코딩을 위한 스케치 환경 만들기

- 포트 선택 : 도구 > COM xxx 선택
; 사용자의 PC의 환경에 따라서, 다른 번호가 표시될 수 있습니다.

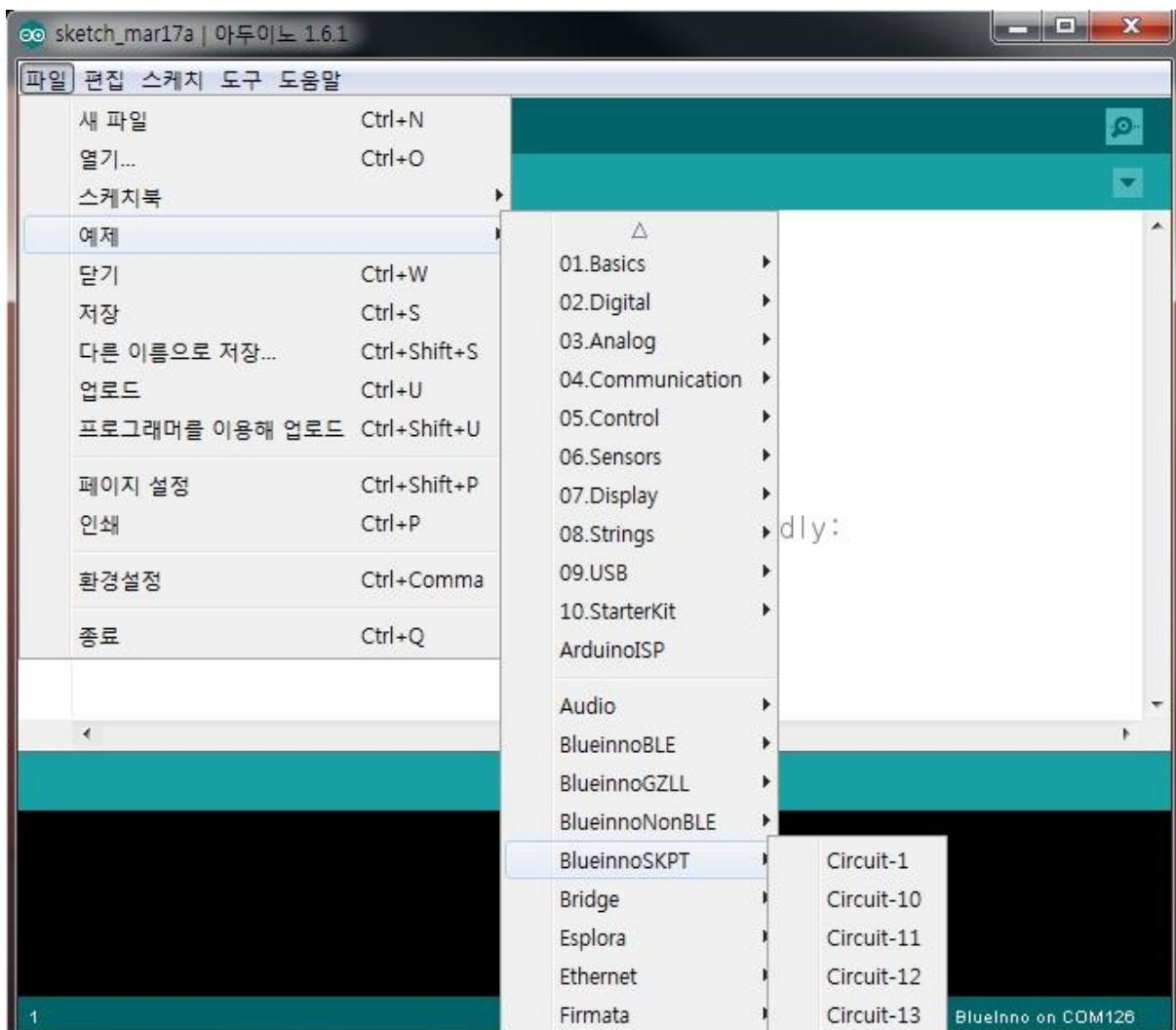


(포트 선택은 PC -> 제어판 -> PC 장치관리자에서 해당 포트를 확인)



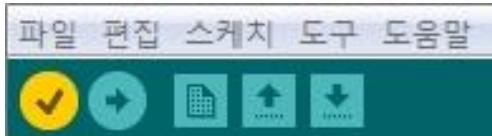
순서3-4 : SW 코딩을 위한 스케치 환경 만들기

- 예제 코드 불러오기



순서3-5 : SW 컴파일과 업로드 하기

- 화면 왼쪽 상단의 ✓버튼을 눌러 불러온 예제 프로그램 소스를 컴파일합니다.



- 완료되면, 왼쪽 하단에 '컴파일 완료'라고 나타납니다.

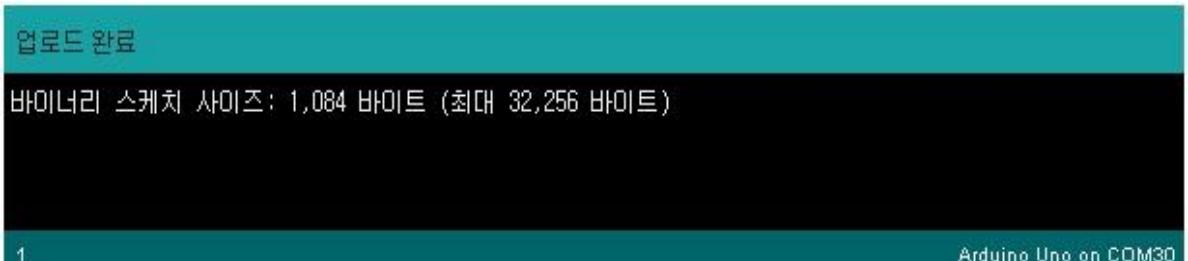


컴파일 에러 발생시에는 불러온 프로그램 소스가 제대로인지 아니면 소스 수정을 했을 경우, 수정된 구문의 문법이 정확한지를 확인하고 다시 컴파일을 해야 합니다.

- 화면 왼쪽 상단의 ⇒버튼을 눌러, 컴파일 완료된 예제 프로그램 소스를 업로드(다운로드)합니다.



실행 후 완료 시 100% (Success)라고 나타납니다.



그러면, 보드는 스케치의 명령에 따라 동작을 합니다.

프로젝트 목차 – 스마트폰과 연결없이 사용하기

- 프로젝트 1 - 목표 : Digital Out 이해
- 내용 : LED 켜기
- 프로젝트 2 - 목표 : Digital In and Digital Out 이해
- 내용 : 푸시 버튼 입력으로 LED 켜기
- 프로젝트 3 - 목표 : Analog Out (주파수) 이해
- 내용 : 간단한 소리 만들기
- 프로젝트 4 - 목표 : Analog Out (PWM) 이해
- 내용 : 푸시 버튼 입력에 의한 LED 밝기 바꾸기
- 프로젝트 5 - 목표 : Analog In and Digital Out 이해
- 내용 : 주변 밝기 변화에 따른 LED 제어하기
- 프로젝트 6 - 목표 : Analog In and Analog Out 이해
- 내용 : 가변 저항으로 서보 모터 각도 제어하기
- 프로젝트 7 - 목표 : Analog In and Analog Out 이해
- 내용 : 가변 저항으로 DC 모터 속도 제어하기
- 프로젝트 8 - 목표 : Analog In to Serial Monitor 이해
- 내용 : 온도 값을 PC 모니터에 표시하기
- 프로젝트 9 - 목표 : LCD 동작원리 이해
- 내용 : LCD 에 문자 표시하기

프로젝트 목차 – 스마트폰과 연결해서 사용하기

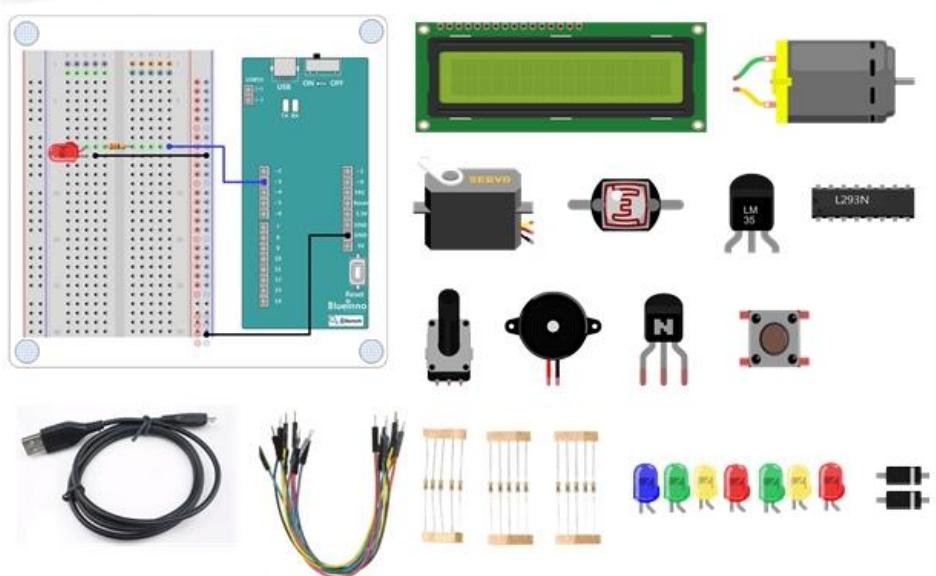
- 프로젝트 10 - 목표 : 블루투스 4.0 이해하기
 - 내용 : 아이비콘(iBeacon) 만들기
- 프로젝트 11 - 목표 : 블루투스 4.0 이해하기
 - 내용 : 스마트 폰으로 LED 켜기
- 프로젝트 12 - 목표 : 블루투스4.0 이해하기
 - 내용 : 스마트 폰으로 LED 색깔 변경하기
- 프로젝트 13 - 목표 : 블루투스4.0 이해하기
 - 내용 : 스마트 폰으로 서보 모터 각도 제어하기

블루이노 스타트 키트 구성품

➤ 블루이노2 스타트 키트 교재

➤ 부품박스

- 블루이노2 기본형 + 브레드 보드 + 고정판
- LCD (16*2열) + DC Motor (프로펠러) + 서보 모터
- 조도센서 + 온도센서 + DC Motor IC + IN4148
- 가변저항 + 피에조 스피커 + NPN 트랜지스터 + 푸시 버튼
- 리더 저항 4종 (10, 330, 1K, 10K ohm)
- LED 4종 (R, G, B, Y) + RGB LED
- 점퍼선 10색 20개 + USB 케이블



아두이노(=블루이노) 프로그래밍의 기초

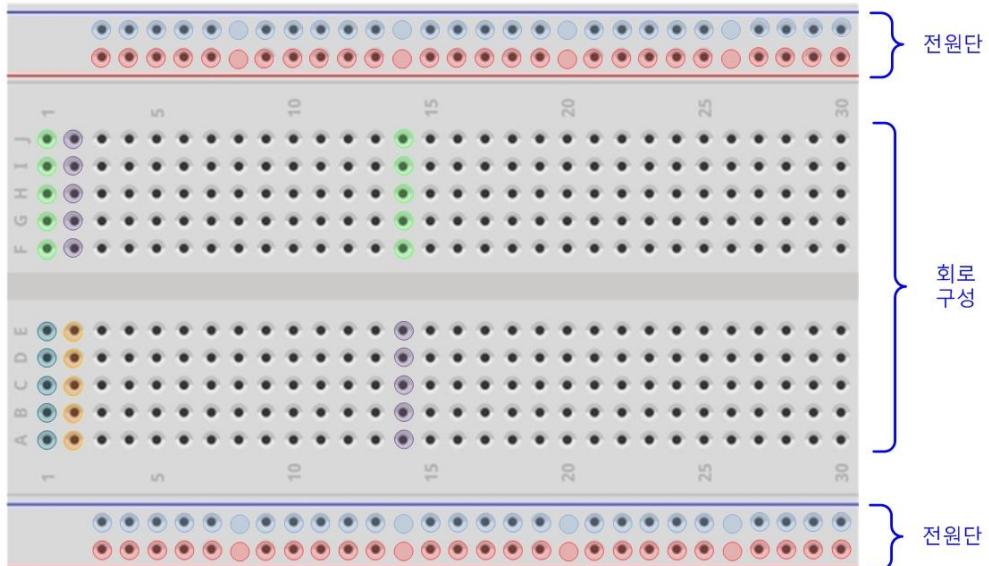
- 아두이노 언어는 C언어와 유사하지만, 더 간단합니다.
아두이노 라이브러리는 C언어로 작성되고, 통합개발환경은 Java로 만들어 졌습니다. 그 프로그램을 스케치(Sketch)라고 부릅니다.
- 프로그램은 3가지 요소 = 변수 + 구조 + 함수
 - 문법
 - `//` 두줄 슬래쉬 뒤는 한 라인의 설명문, 코드 실행 안됨
 - `/*` 사이는 여러 라인의 설명문, 코드 실행 안됨 `*/`
 - `{` 중괄호 사이는 코드 블록 `}`
 - `;` 한 줄의 코드 끝에는 세미콜론을 붙임
 - 변수
 - `int` : 4Byte(32bit) 정수 (-2,147,483,648 ~ +2,147,483,647)
 - `long` : 4Byte(32bit) 정수 (-2,147,483,648 ~ +2,147,483,647)
 - `boolean` : 1bit 참(True) or 거짓(False)
 - `float` : 4Byte 소수 (-3.4E+38 ~ +3.4E38)
 - `char` : 1바이트, ASCII 코드 'A'=65, 문자열은 "ABC"로 표현
 - 구조 = `void setup () {}` + `void loop () {}`
 - `void setup () {` // 이 곳에 있는 코드는 한번만 실행됨 `}`
 - `void loop () {` // 이 곳에 있는 코드는 반복해서 실행됨 `}`
 - 함수
 - 산술 연산자 (수학적 계산)
 - 1) `=` : 대입
 - 2) `+` : 덧셈
 - 3) `-` : 뺏셈
 - 4) `*` : 곱셈
 - 5) `/` : 나눗셈
 - 6) `%` : 나누고 남은 값 (예 : $7\%4 = 3$)

아두이노(=블루이노) 프로그래밍의 기초

- 비교 연산자 (논리적 비교)
 - 1) `==` : 같은가?
 - 2) `!=` : 같지 않은가?
 - 3) `<` : 작은가?
 - 4) `>` : 큰가?
- 제어구조 ; 프로그램은 코드를 차례대로 실행하나, 제어문에 의해서 바꿉니다.
 - 1) `If (조건) { }`
`else if (조건) { }`
`else { }`
 - 2) `for (int i=0; i<10; i++){ } // i=0에서 9까지, {} 을 10회 반복 실행합니다.`
- 디지털 함수
 - 1) `pinMode(pin, mode)` : 해당 핀과 입/출력으로 정합니다.
 - 2) `digitalWrite(pin, value)` : 디지털 출력 (해당 핀, 출력 값(High/Low))을 정합니다.
 - 3) `int digitalRead(pin)` : 디지털 입력 값을 해당 핀에서 2Byte 정수로 읽습니다.
- 아날로그 함수
 - 1) `pinMode(pin, mode)` : 해당 핀과 입/출력으로 정합니다.
 - 2) `analogWrite(pin, value)` : 아날로그 출력 (해당 핀, 출력 값)을 정합니다.
 - 3) `int analogRead(pin)` : 아날로그 입력 값을 해당 핀에서 2 Byte 정수로 읽습니다.

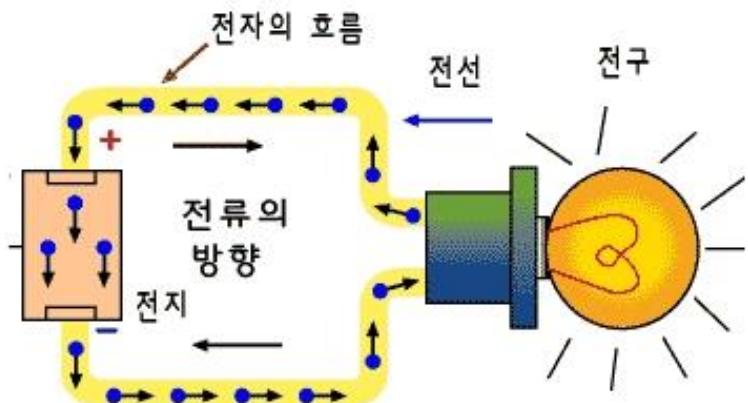
전자회로 기초 상식

▶ 브레드 보드



- 납땜 없이 자유롭게 회로 구성이 가능한 만능 보드입니다.
- 윗면과 아래면은 전원 연결용으로 가로로 연결되어 있습니다.
- 중간 면은 회로구성용으로 세로로 연결되어 있습니다.

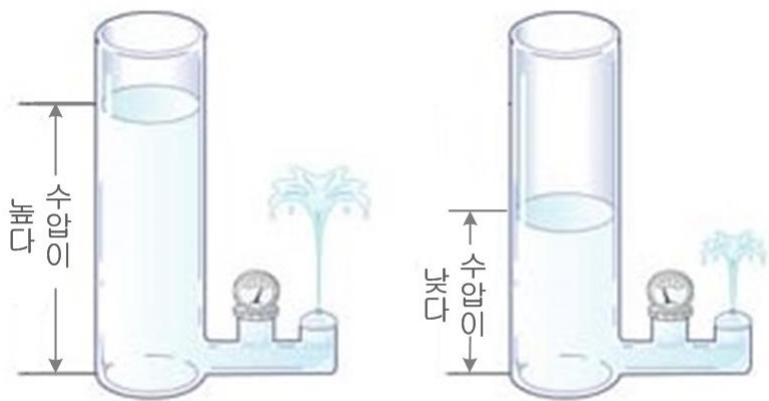
▶ 기본 전자 회로



- 전기는 전자의 흐름입니다. 전기는 $(+)$ \Rightarrow $(-)$ 로 흐릅니다.
- 전기를 가지고 있는 것은 전지이며, 이를 흘르게 하는 것은 전선입니다.
- 전기는 조명을 켜거나 소리를 내거나 모터를 움직이게 합니다.

전자회로 기초 상식

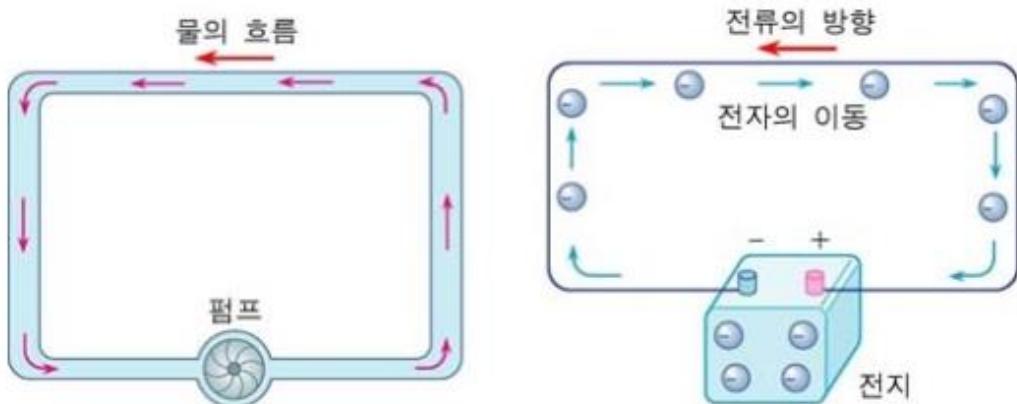
➤ 전압 :



[수압과 물줄기의 세기]

- 전기적인 위치에너지(전위, 높이)의 차이입니다.
- 물에 비유하면, 수위(물의 높이), 즉 수압으로 생각할 수 있습니다.
- 전압은 일반 건전지나 전압 생성장치(충전기, 파워공급장치)를 통해서 얻어집니다.

➤ 전류 :

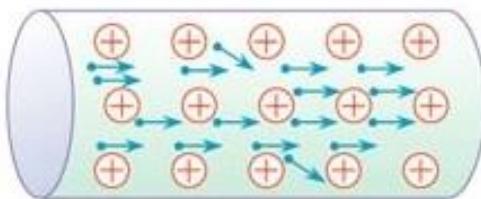


[물의 흐름과 도선 속의 전자의 이동 및 전류의 방향 비교]

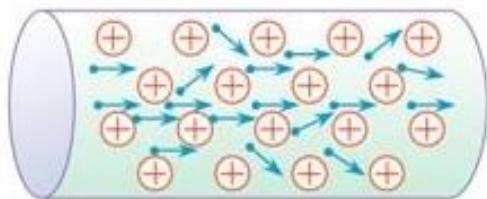
- 전자가 가지고 있는 전하의 흐름으로, 전선과 같은 도체에서 일어납니다.
- 전류의 세기는 전압의 세기와 저항에 의해서 영향을 받습니다.

전자회로 기초 상식

➤ 저항 :



(a) 저항이 작은 물질

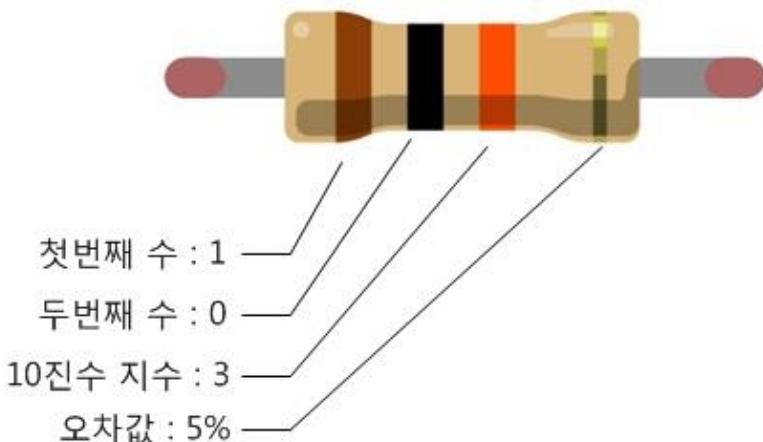


(b) 저항이 큰 물질

[금속 내에서 전자의 운동]

- 전류의 흐름을 방해하는 정도를 나타내는 수치를 저항이라고 합니다.
- 저항 값이 낮을수록 전류의 흐름을 적게 방해합니다.

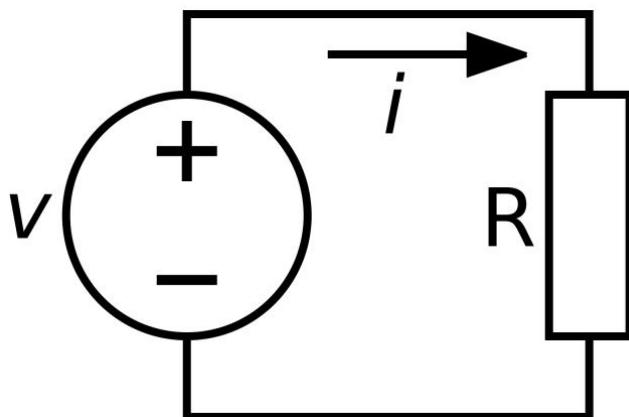
➤ 저항표



$$10 \times 10^3 = 10\text{K}\Omega \text{ 편차 } 5\%$$

첫번째, 두번째, 세번째 색띠의 숫자										네 번째 색띠		
흑	갈	적	황	노	녹	청	보	회	백	금	은	없음
0	1	2	3	4	5	6	7	8	9	5%	10%	20%

➤ 옴의 법칙



- 도체(저항)의 두 지점 사이에 나타나는 전위차에 의해 흐르는 전류가 일정한 법칙에 따르는 것을 말합니다.
- 전압(V) = 전류(I) X 저항(R)
 - 1) 전압의 단위는 볼트(V, volt)
 - 2) 전류의 단위는 암페어(A, ampere)
 - 3) 저항의 단위는 옴(Ω , ohm)
- 도체 양단에 전위차가 존재할 때, 도체의 저항의 크기와 전류는 반비례 관계를 갖습니다.
- 저항은 전류를 제한하는 요소로 작용합니다.

프로젝트 1 : LED 켜기

➤ 목적 :

- 디지털 출력(Digital Out)의 개념 이해를 위하여, LED 켜기 / 끄기의 예제를 사용합니다.
- 디지털은 0과 1로 표현됩니다.
- 블루이노에서는 VCC 입력전압이 3.3V이므로, High는 3.3V입니다.
- 보드 출력은 High 신호인 3.3V와 Low 신호인 0V를 출력할 수 있습니다.



➤ 기술 설명 – LED

- LED는 Light Emitting Diode의 약자로 화합물 반도체의 특성을 이용해 전기에너지를 광(빛) 에너지로 변환시켜 주는 반도체 소자를 말합니다.

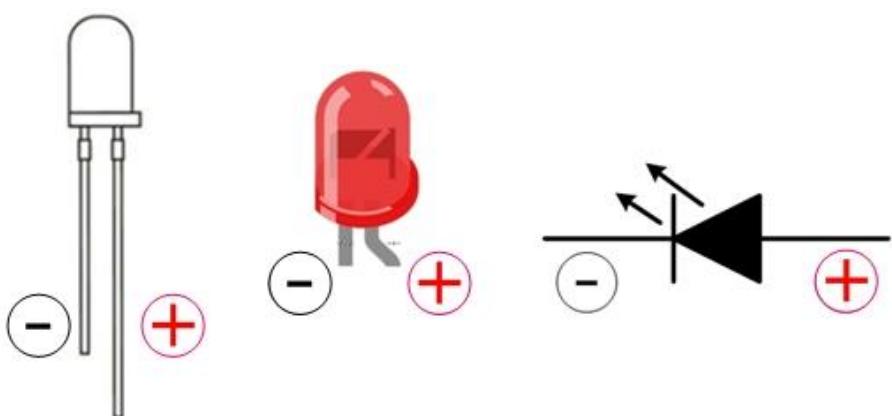


프로젝트 1 : LED 켜기

- P형, N형 반도체를 접합한 다이오드에 순방향 전압을 인가하면 전자와 정공이 결합하면서 빛이 발광되는 원리로 빛이 발생합니다.



- LED는 극성이 있으므로, 연결 시 극성을 잘 확인해야 합니다.



프로젝트 1 : LED 켜기

➤ 준비물 목록

No	부품명	수량	자세한 설명
1	스타트키트 기본세트	1	프로젝트 기본구성
2	LED (R/Y/G/B)	1	빛을 발생하는 부품
3	저항 330Ω	1	전류량 조절 부품

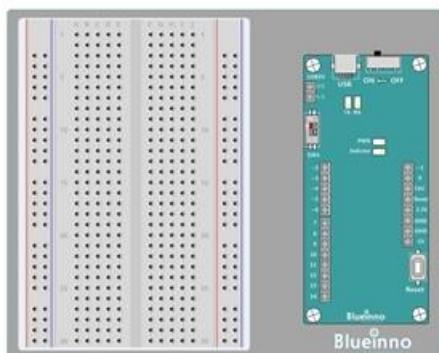
스타트키트 기본세트



점프 케이블



USB 케이블



브레드보드 + 블루이노 기본형



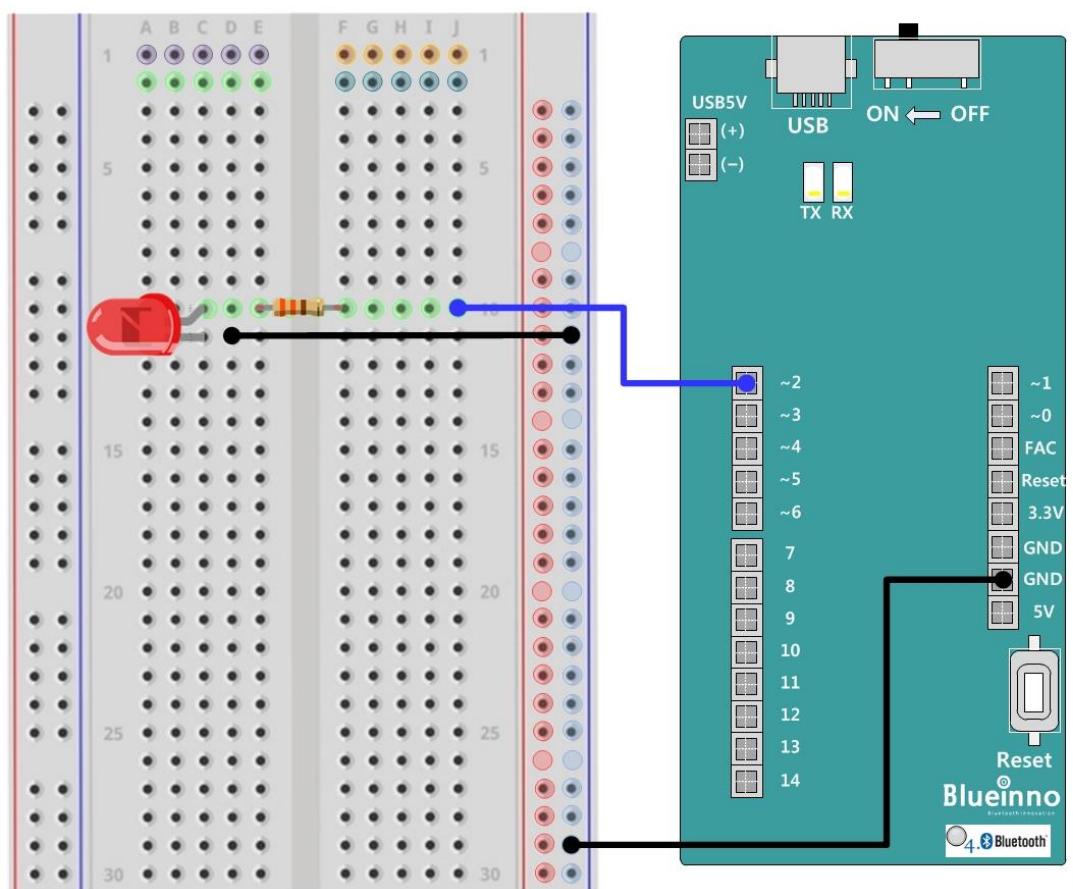
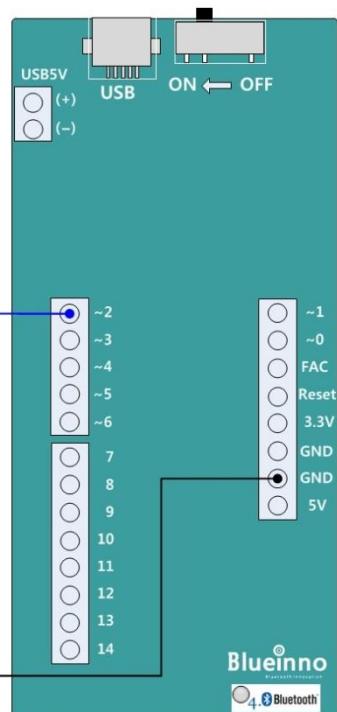
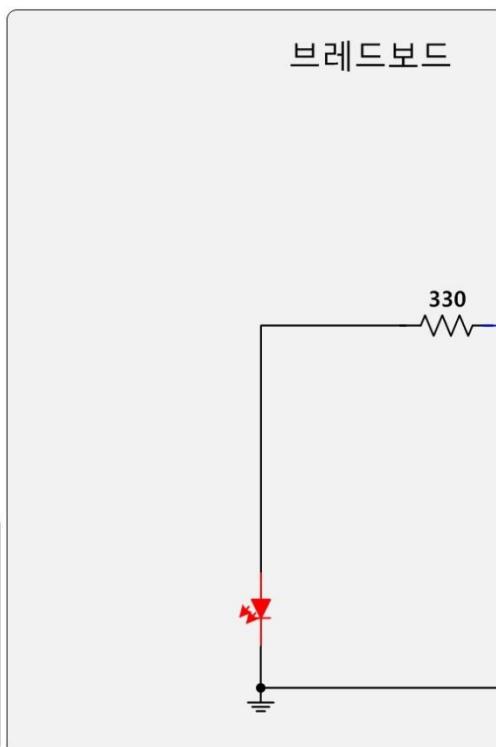
LED



저항 330Ω

프로젝트 1 : LED 켜기

➤ 회로도



➤ 회로 설명 :

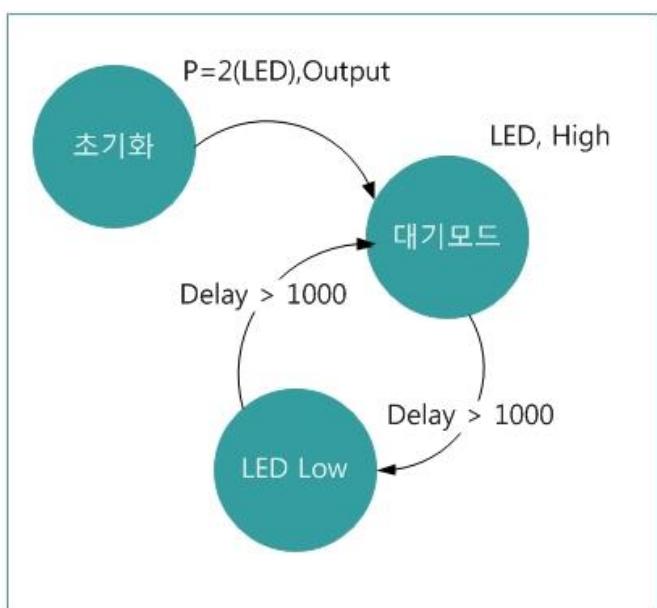
- LED 밝기는 저항 값을 통해 조절할 수가 있습니다.
보통은 전류를 약 4mA ((3.3-2.0V) / 330Ω) 정도 흘려 주면 됩니다.
- 포트의 최대 디지털 출력은 5mA입니다.
- 여러 포트로 동시에 출력시 최대합이 15~20mA 전류를 직접 공급이 가능합니다만, 더 이상의 전류공급을 희망하신다면, 별도의 회로를 구현하여야 합니다.

➤ 동작 :

- 전원이 공급되면, 빨강색 LED가 켜짐(On)=> 꺼짐(Off) => 켜짐(On) 을 반복합니다.

➤ 소프트웨어 구현 방향

- 하나의 포트를 출력으로 세팅합니다.
- 1초 간격으로 On/Off를 반복합니다.



- 예제 >BlueinnoSKPT > Circuit - 1

```
/*
LED 깜빡이기 ; 설명문
*/
// 초기 한번 설정
void setup() {
    pinMode(2, OUTPUT); // 2번 핀을 출력으로 설정 (설명문)
}
// 반복 실행
void loop() {
    digitalWrite(2, HIGH); // 디지털출력으로 2번 핀을 High로 출력
    delay(1000); // 1000ms 동안 상태 유지

    digitalWrite(2, LOW); // 디지털출력으로 2번 핀을 Low로 출력
    delay(1000); // 1000ms 동안 상태 유지
}
```

프로젝트 2 : 푸시 버튼으로 LED 켜기

➤ 목적 :

- 디지털 입력(Digital In)의 개념 이해를 위하여,
푸시 버튼을 활용한 LED 켜기 / 끄기의 예제를 사용합니다.
- 보드 입력은 3.3V ~ 2.6V(3.3V x 80%)까지는 High로 인식하며,
0V ~ 0.6V(3.3V x 20%)까지는 Low로 인식을 합니다.

➤ 기술 설명

▪ Key의 동작원리

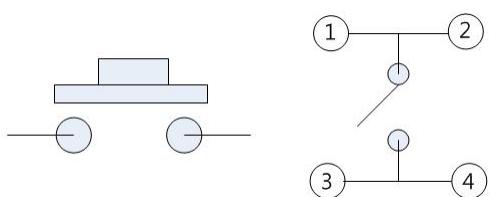
- 푸시 버튼 스위치나 텍트 스위치는 두 개의 지점을 연결시키는 전자부품입니다.
- 텍트 스위치는 누를 때만 연결합니다. 푸시 버튼 스위치는 누를 때만 연결되는 부품과 계속 유지되는 되는 부품 형태도 있습니다.
- 여기서 키(Key)는 누를 때만 연결되는 푸시 버튼 스위치를 사용합니다.



PUSH Switch



TACT Switch



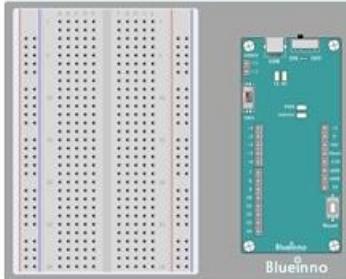
▪ Key의 응용

- 키를 디지털 입력으로 사용할 수가 있습니다.
- 회로에 따라서, 키 입력을 High (+3.3V)에서 인식하거나,
Low (0V)에서 인식하게 디지털 입력단을 설정할 수가 있습니다.
- 디지털 신호는 보드의 마이크로 컨터롤러에 인가되는 전압(VCC)에 따라서,
+5V, +3.3V 처럼 달라질 수 있습니다.

프로젝트 2 : 푸시 버튼으로 LED 켜기

➤ 준비물 목록

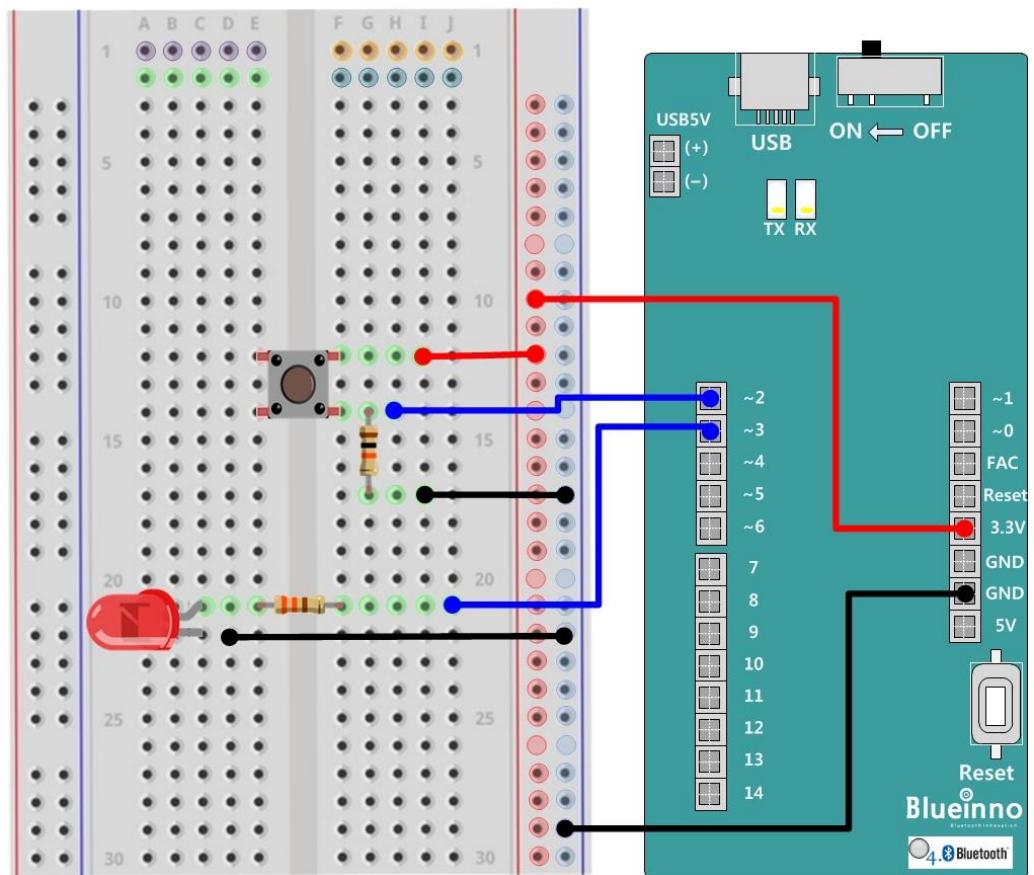
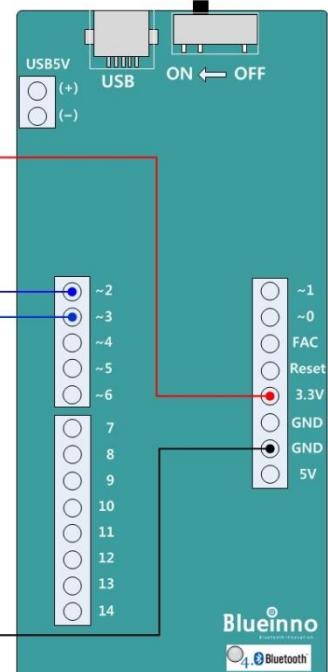
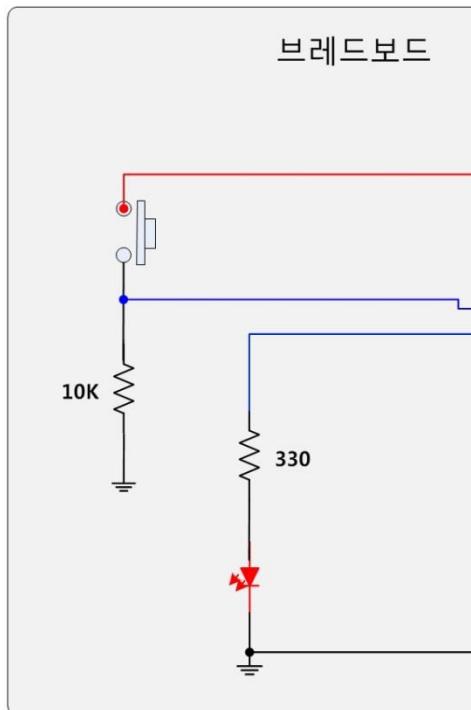
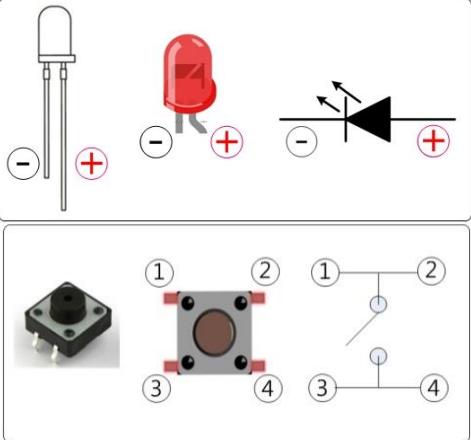
No	부품명	수량	자세한 설명
1	스타트키트 기본세트	1	프로젝트 기본구성
2	LED (R/Y/G/B)	4	빛을 발생하는 부품
3	저항 330Ω	4	전류량 조절 부품
4	저항 10KΩ	1	스위치 회로 안정화
5	푸시 스위치	1	스위치 부품

스타트키트 기본세트			
			
점프 케이블	USB 케이블	브레드보드 + 블루이노 기본형	
			
LED 저항 330Ω 저항 10KΩ 푸시스위치			

프로젝트 2 : 푸시 버튼으로 LED 켜기

➤ 회로도

[부품 방향 주의]



프로젝트 2 : 푸시 버튼으로 LED 켜기

➤ 회로설명 :

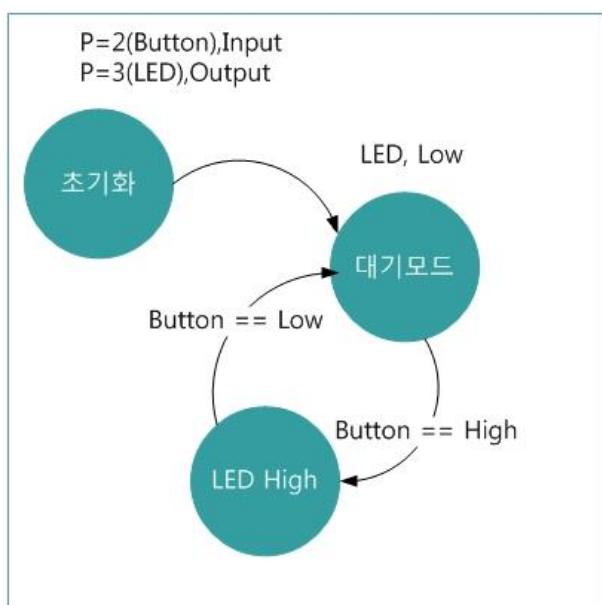
- 포트 2번은 초기에는 0V (Low)로 유지되다가, 버튼을 누르면 3.3V가 공급되어 High로 인식합니다.
- 외부 입력이 없을 때, 불안한 동작을 하지 않도록 하는 $10k\Omega$ 저항을 GND에 병렬로 연결합니다. 이것을 Pull-Down 저항이라고 합니다.

➤ 동작 :

- 전원이 공급되면, 모든 LED가 꺼져 있다가, 푸시 버튼을 누르면 빨강색 LED 켜지고, 누르지 않으면 꺼진 상태를 유지합니다.

➤ 소프트웨어 구현 방향

- 하나의 포트는 입력포트(버튼)로 또 하나의 포트는 출력포트(LED)로 설정합니다.
- 버튼을 누르는 동안 LED가 켜지고, 버튼을 놓으면 LED가 깨지게 합니다.



■ 예제 >BlueinnoSKPT > Circuit - 2

```
/* 버튼을 누르는 동안 LED켜기 */
int button = 2; // 버튼 핀을 2번으로 설정
int LED = 3; // LED핀을 3번으로 설정
int buttonState = 0; // 입력 핀의 상태를 읽기 위한 변수
```

void setup()

```
pinMode(LED , OUTPUT); // LED 핀을 출력으로 설정
pinMode(button, INPUT); // 버튼 핀을 입력으로 설정
```

void loop()

```
buttonState = digitalRead(button);
// 디지털입력 핀인 버튼이 눌려졌는지를 읽음
if(buttonState == HIGH) { // 버턴의 상태가 High 이면
    digitalWrite(LED , HIGH); // LED 핀을 High로 출력
}
else { // 버턴의 상태가 High가 아니면
    digitalWrite(LED , LOW); // LED 핀을 Low로 출력
}
```

프로그램 설명

- if() else 조건문 : if (함수의 조건)이 맞으면, if { }안의 루틴을 돌고, 틀렸을 때는 else{ } 안의 루틴을 돌게 하는 조건문입니다.

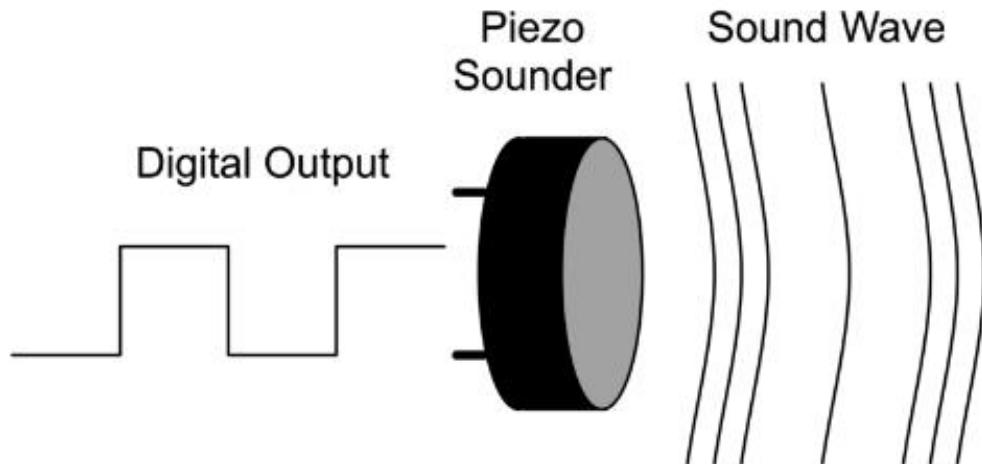
프로젝트 3 : 간단한 소리 만들기

➤ 목적 :

- 디지털 출력(Digital Out)의 개념 이해를 위하여, 톤(주파수) 출력하기의 예제를 사용합니다.

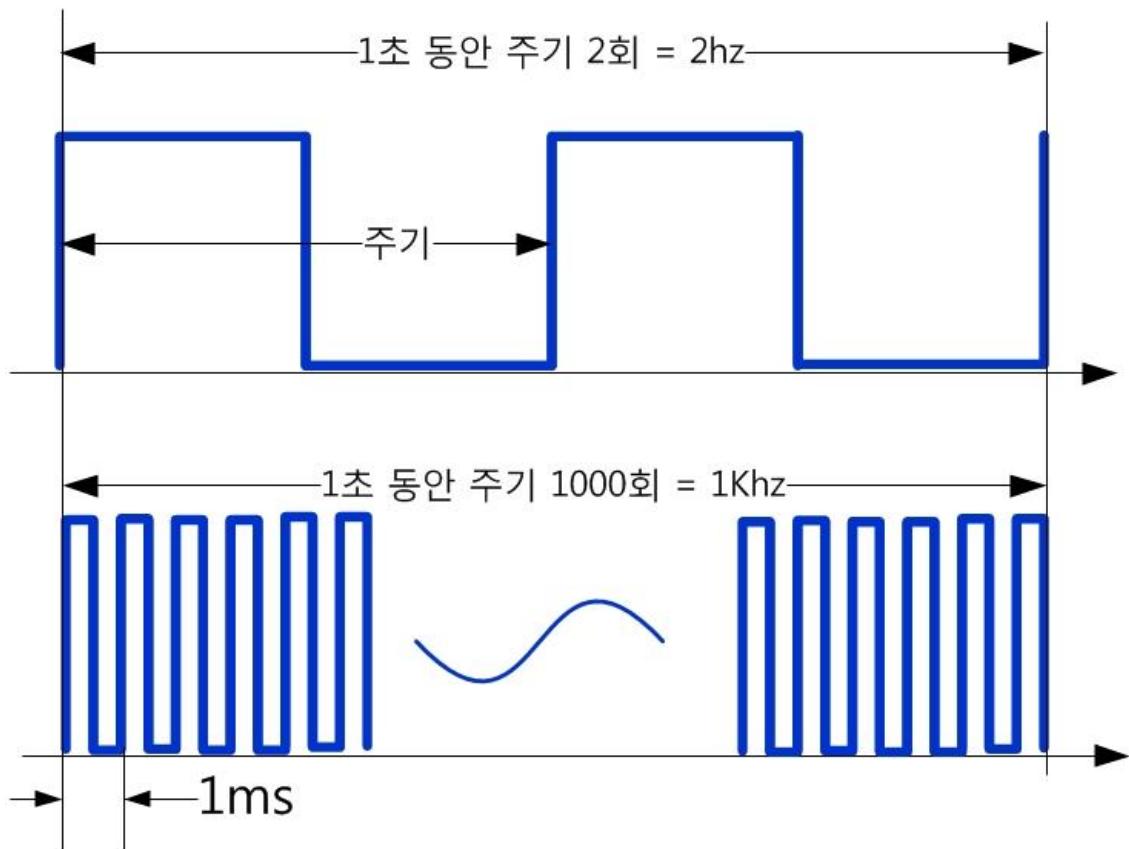
➤ 기술 설명 : 피에조 스피커

- 피에조 스피커(Piezoelectric Speaker)는 크기가 작고, 저렴하지만 간단하게 전기 신호를 주어서 소리를 내게 하는 전자 부품입니다.
- 피에조 필름에 진동이나 힘을 가하면, 힘에 비례하여, 전압이 발생하여 압전센서 역할을 하고, 반대로 필름에 전압을 넣을 경우 기계에서 소리가 납니다.



기술 설명 : 디지털 출력으로 주파수 만들기

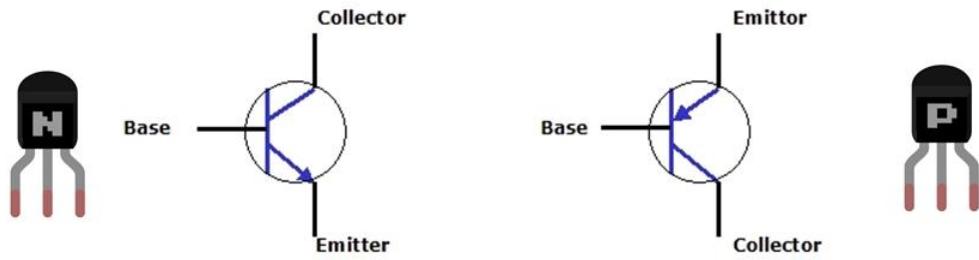
- 디지털 출력의 High / Low 신호가 일정한 간격으로 반복된다면, 1초 동안 반복된 회수를 주파수(Frequency)라 하고, 단위는 헤르츠(Hz)로 표기합니다.
- 희망하는 주파수에 맞는 High/Low 반복 비율로 제어해야만 소리를 만들 수가 있습니다. 만약에 1Khz 소리를 발생하려면, 0.5ms 단위로 High/Low를 반복하여야 합니다.



- 음표는 [C, D, E, F, G, A, B, C]로 표시하는데, 우리말로 [도, 레, 미, 파, 솔, 라, 시, 도]이며, 소리의 높낮이에 따라서, 주파수가 달라진다.
- C 음은 보통 277Hz로 정의 되어 있습니다.
- 좋은 멜로디의 소리가 출력되려면, 음(소리)의 높낮이와 박자(길이)가 조화롭게 되어야 합니다.

기술 설명 : 트랜지스터(Transistor)

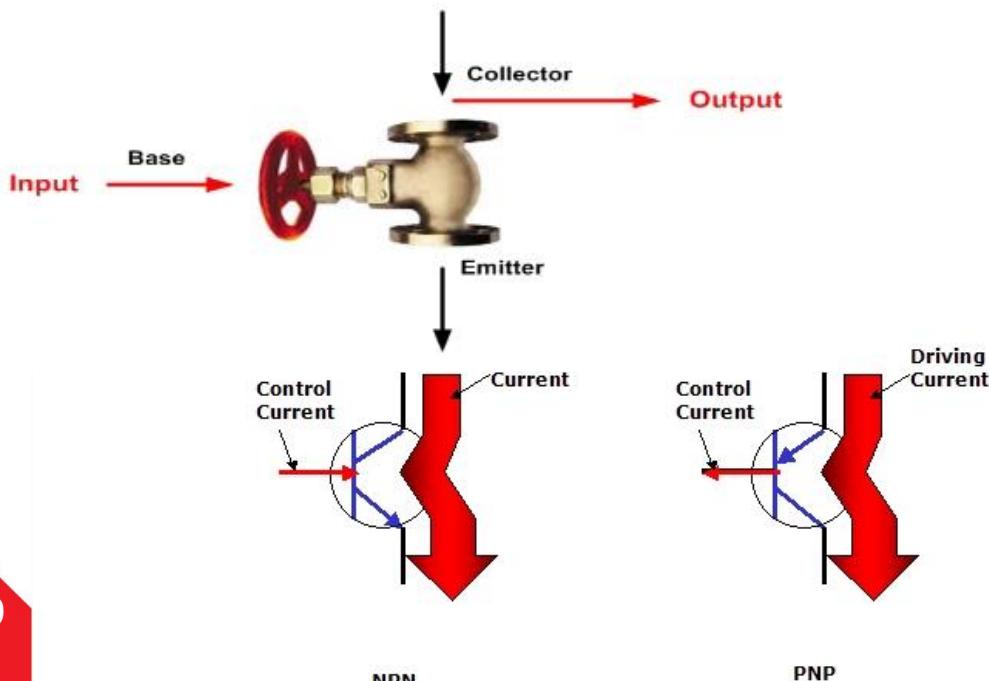
- 포트의 전류가 부족할 경우에는 외부에 트랜지스터를 활용해서, 많은 전류 공급이 가능합니다. 트랜지스터는 여러 종류가 있습니다.



NPN 트랜지스터

PNP 트랜지스터

- 트랜지스터에는 이미터(emitter), 콜렉터(collector), 베이스(base)로 세 개의 단자가 있습니다.
- NPN 트랜지스터의 기본적인 동작 원리는 작은 전류가 베이스로 흘러 들어가면, 콜렉터에서 이미터로 큰 전류가 흐르는 것입니다.
그래서, 이를 증폭기라고 표현하기도 합니다.
- PNP 트랜지스터의 기본적인 동작 원리는 이미터의 작은 전류가 베이스에서 흘러 들어가면, 이미터에서 콜렉터로 큰 전류가 흐르는 것입니다.



프로젝트 3 : 간단한 소리 만들기

➤ 준비물 목록

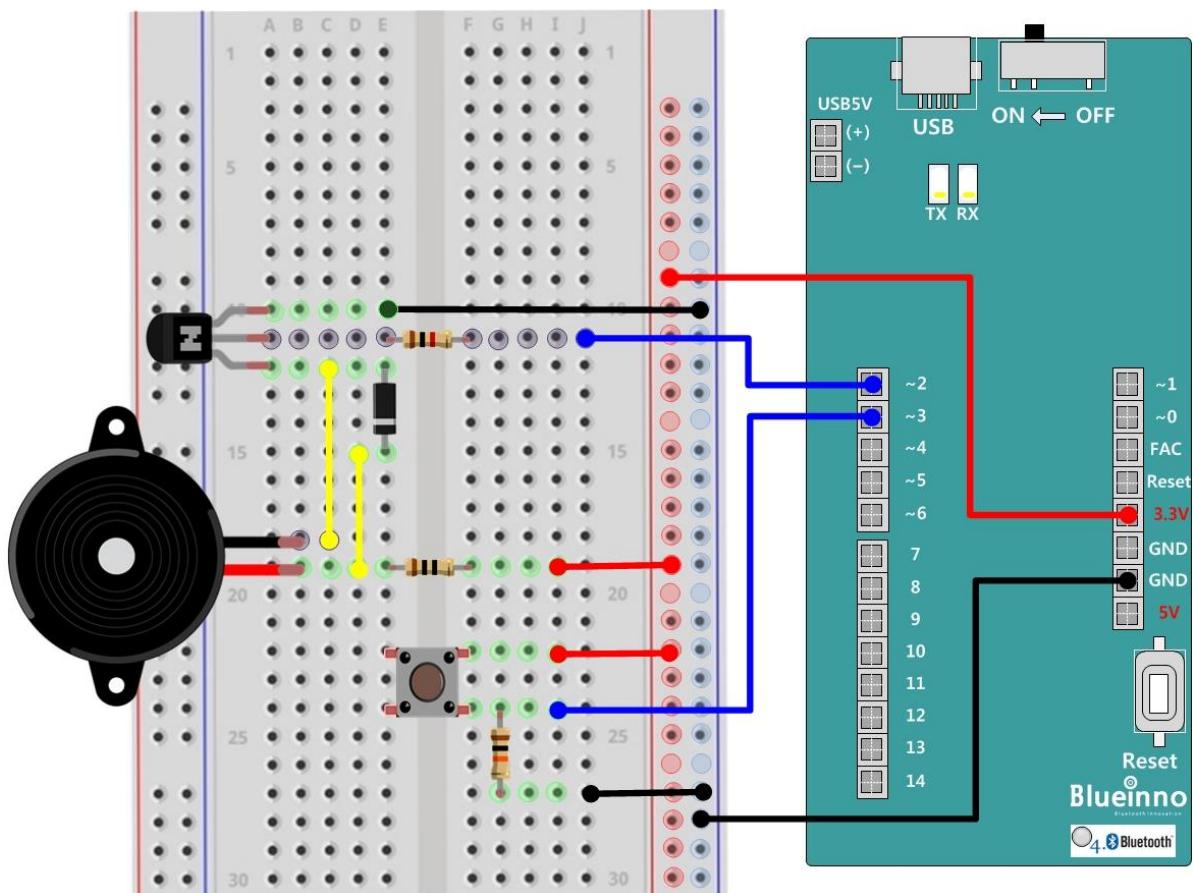
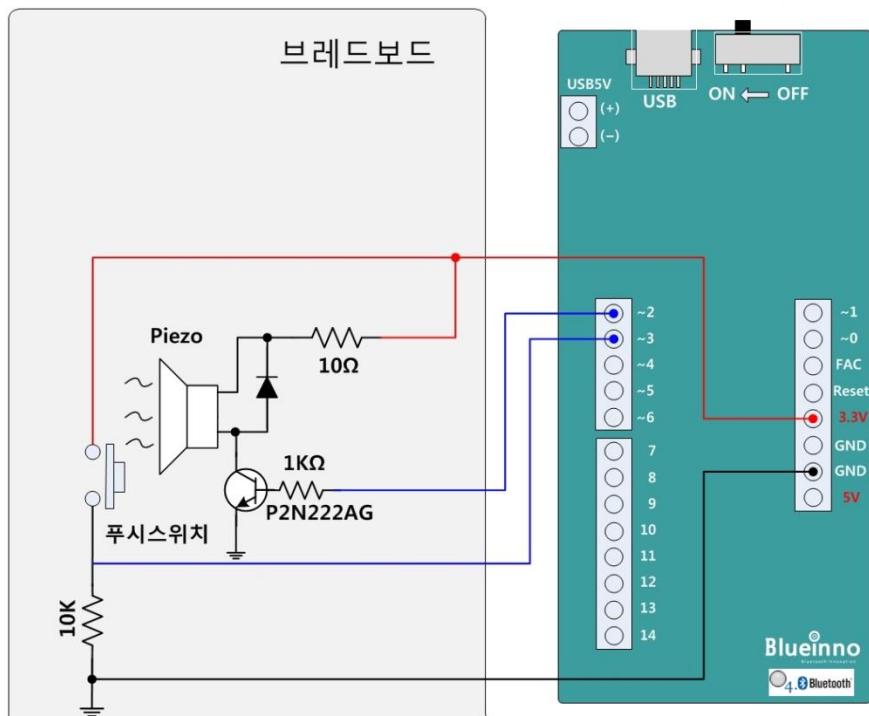
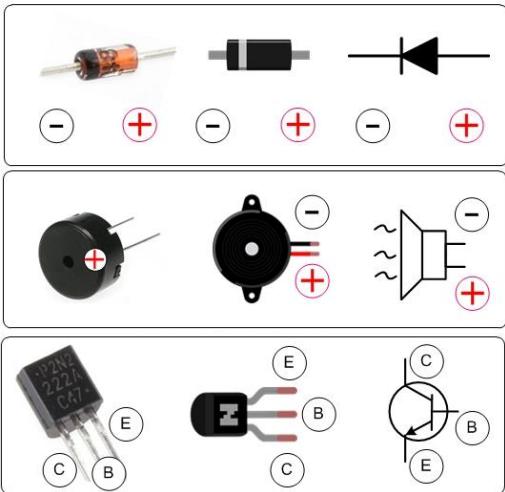
No	부품명	수량	자세한 설명
1	스타트키트 기본세트	1	프로젝트 기본구성
2	저항 10Ω	1	전류량 조절 부품
3	저항 1KΩ	1	전류량 조절 부품
4	저항 10KΩ	1	스위치 회로 안정화
5	푸시 스위치	1	스위치 부품
6	다이오드 1N4148	1	트랜지스터 보호 부품
7	트랜지스터 P2N222AG	1	전류를 증폭하는 부품
8	피에조 스피커	1	소리내는 부품

스타트키트 기본세트			
점프 케이블	USB 케이블	브레드보드 + 블루이노 기본형	
저항 10Ω	저항 1KΩ	저항 10KΩ	푸시스위치
다이오드	트랜지스터	피에조 스피커	

프로젝트 3 : 간단한 소리 만들기

➤ 회로도

[부품 방향 주의]



프로젝트 3 : 간단한 소리 만들기

➤ 회로 설명 :

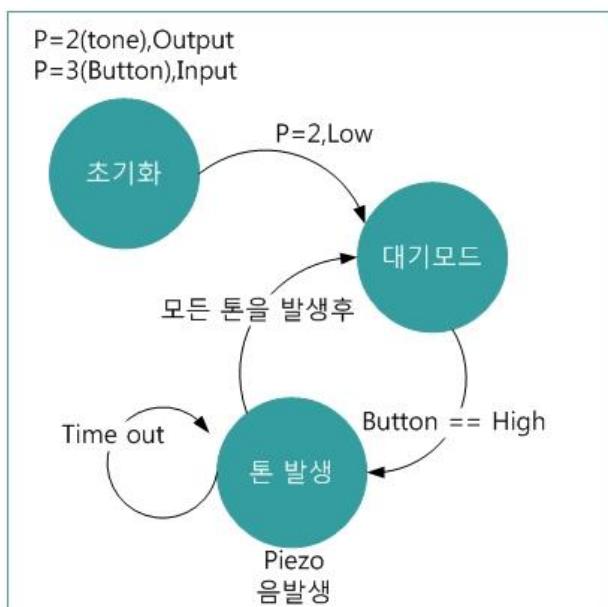
- 포트3번의 버튼이 눌러지면, 포트 2번을 통해 디지털 주파수를 출력하면, 트랜지스터를 도통시켜서, 피에조의 소리가 나게 합니다.
- 권장회로는 피에조의 동작 전류가 큰 소자를 기준으로 했으며, 일부 낮은 전류 (20mA이하)로 동작이 가능한 소자는 트랜지스터를 통하지 않고, 직접 포트에 연결해도 동작합니다.

➤ 동작 :

- 버튼을 누르면 디지털 주파수를 출력하고, 끝나면 다시 버튼을 누를 때까지 대기합니다.

➤ 소프트웨어 구현 방향

- 하나의 포트는 입력포트(버튼)로 또 하나의 포트는 출력포트(Piezo)로 설정합니다.
- 버튼을 누르면 배열에 해당하는 주파수 톤을 발생합니다.
- 모든 주파수 톤이 발생이 끝나면, 다음 버튼 입력을 기다립니다.



■ 예제 >BlueinnoSKPT > Circuit -3

```

/*
피예조 스피커에 톤(고정 주파수) 소리 출력하기
*/
int speakerPin = 2; // 스피커에 연결된 핀을 2번으로 설정
int button = 3; // 버튼에 연결된 핀을 3번으로 설정
int buttonState = 0; // 버튼 핀의 상태를 저장하기 위함

void setup() {
pinMode(speakerPin , OUTPUT); // 스피커 핀을 출력으로 설정
pinMode(button, INPUT); //버튼 핀을 입력으로 설정
}

void loop() {
buttonState = digitalRead(button); // 버튼이 눌려졌는지를 읽음
if(buttonState == HIGH) { // 조건; 버튼의 상태가 High 이면
tone(speakerPin, 2000, 1000); // 스피커에 2000hz의 음을 1초 동안 내기
delay(2000); // 2초간 대기
}
else { // 버튼의 상태가 High가 아니면
noTone(speakerPin); // 스피커 핀의 출력을 하지 않습니다.
}
}

```

➤ 프로그램 설명

- tone() : 피예조 스피커에 특정 주파수를 발생시킵니다.
- tone(pin, frequency, duration) : 피예조 스피커의 해당 핀, 특정 주파수, 일정시간 동안 출력을 합니다.

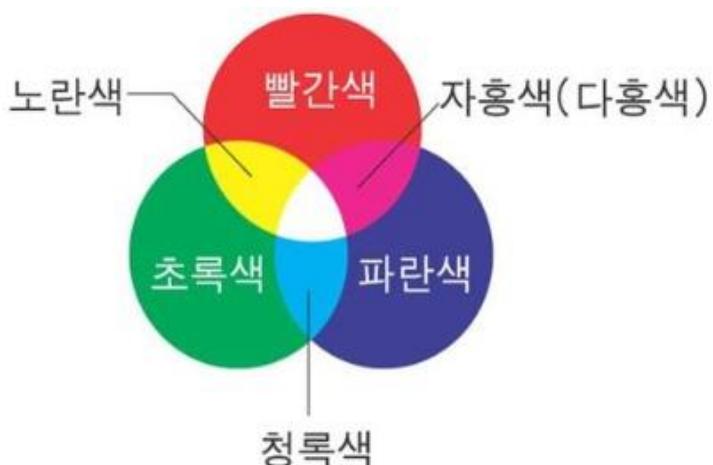
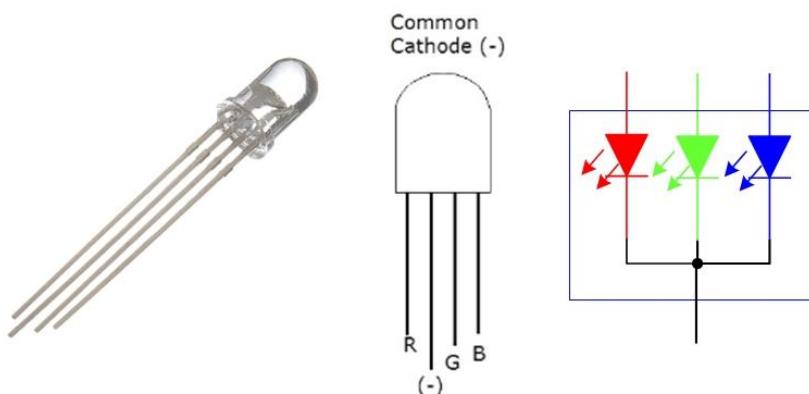
프로젝트 4 : 푸시 버튼 입력에 의한 LED 밝기 바꾸기

➤ 목적 :

- 아날로그 출력(Analog Out)의 개념 이해를 위하여, 3색 (Red/Green/Blue) LED의 밝기 바꾸기를 예제로 사용합니다.
- digitalWrite(255)는 가장 많은 전류를 공급하므로, 밝기가 가장 밝습니다.
- digitalWrite(0)은 가장 적은 전류를 공급하므로, 밝기가 가장 어둡습니다.

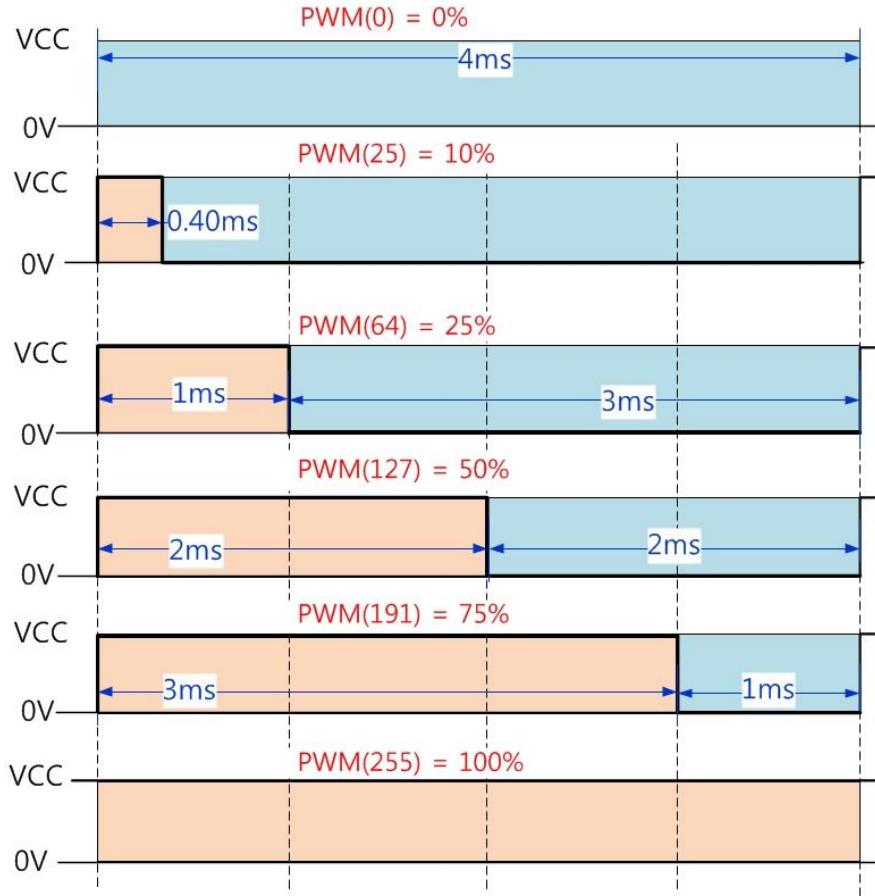
➤ 기술 설명 – 3색 (RGB ; Red/Green/Blue) LED

- 빛의 3원색인 빨간색(Red), 초록색(Green), 파란색(Blue)을 가지고, 3가지 색의 세기(혼합)에 따라서 다양한 색의 빛을 만들 수 있습니다.
- 그리고, 빛의 세기를 전류의 양(PWM)으로 조절할 수가 있습니다.



기술 설명 : 아날로그 출력 (PWM)

- Pulse Width Modulation (PWM)은 전원을 제어하는 기술입니다.
- 여기서는 LED의 밝기를 제어하기 위해 사용됩니다. 아래의 그림은 PWM핀에서 나오는 신호를 보여줍니다.



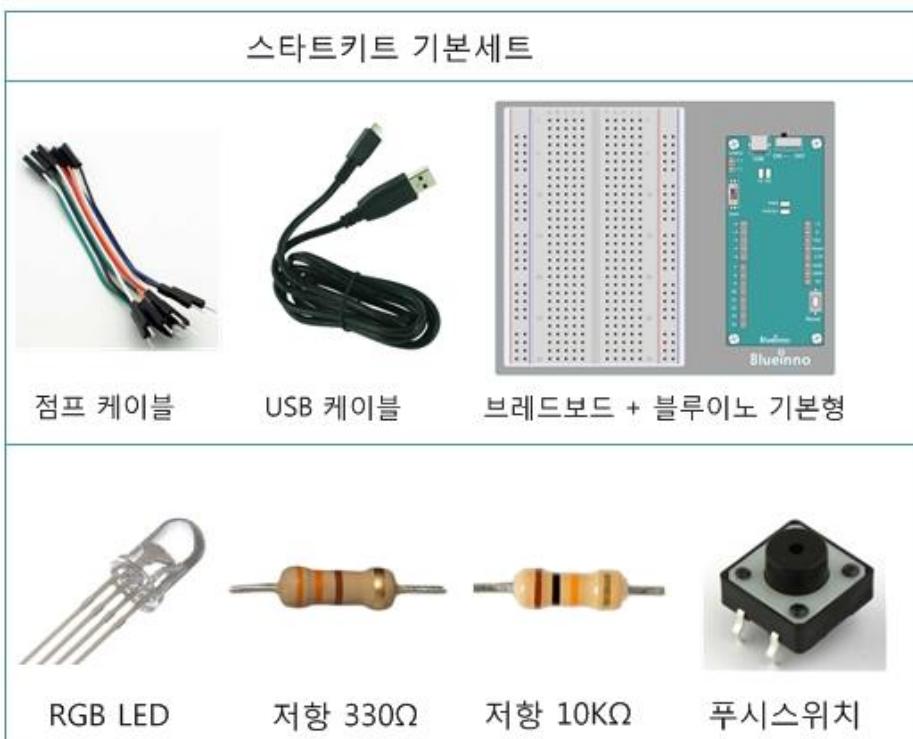
아날로그 신호 (출력) – 블루이노

- 대략 4ms(1/250 초)마다 PWM 출력은 펄스를 만들어냅니다. 이 펄스의 길이는 analogWrite 함수에 의해 제어됩니다. analogWrite(0)은 아무런 펄스를 만들지 않는 것을 의미하고 analogWrite(255)는 펄스를 계속 만들고 있는 것을 의미합니다.
- analogWrite에 넘겨주는 값을 지정함으로써 펄스를 만들 수 있고, 만약 펄스가 전체 펄스에서 5%만 나온다면 전체 전력에서 5%만 공급되는 것이라고 볼 수 있습니다.
- 실제적으로 LED는 pulse의 주기마다 계속 on/off되고 있지만 사람의 눈은 이런 속도로 깜박이는 것을 인식할 수 없고 단지 밝기가 변하고 있는 것만 인식할 수 있습니다.

프로젝트 4 : 푸시 버튼 입력에 의한 LED 밝기 바꾸기

➤ 준비물 목록:

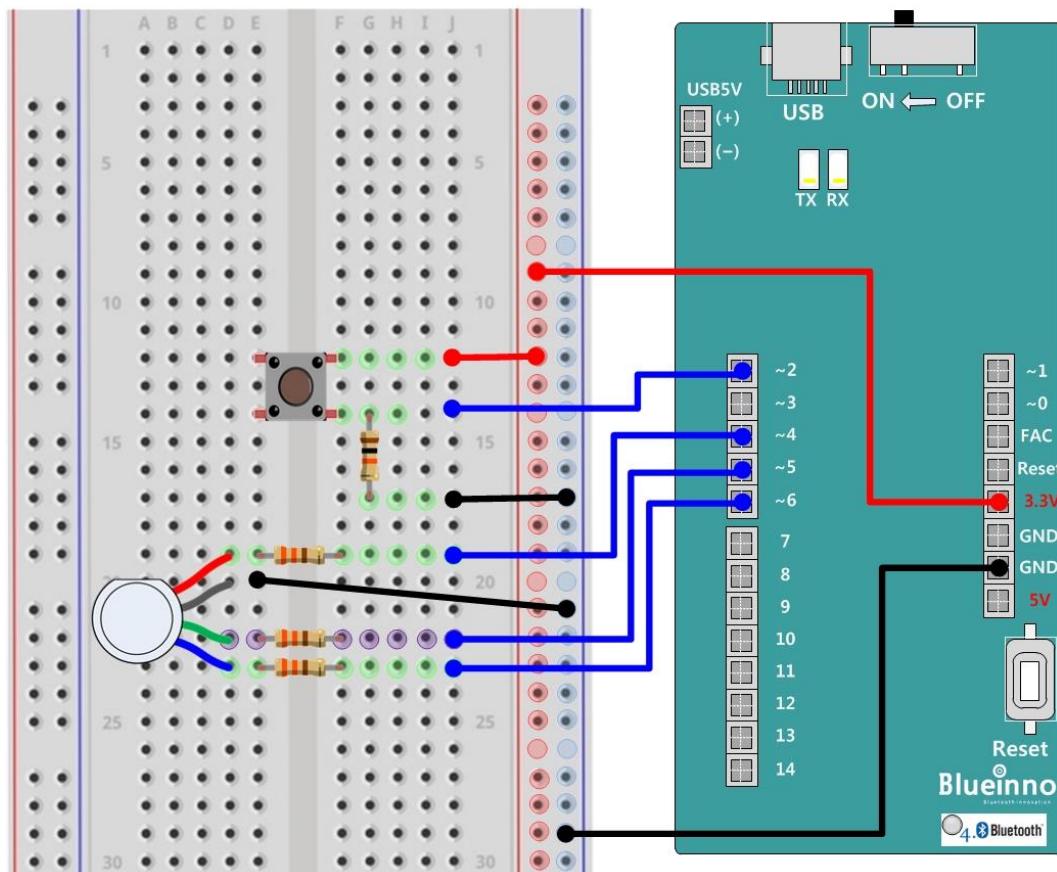
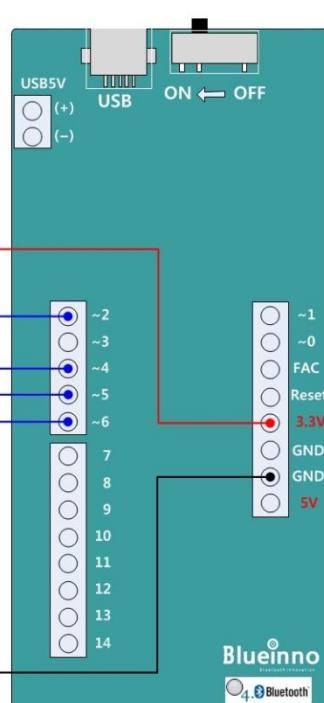
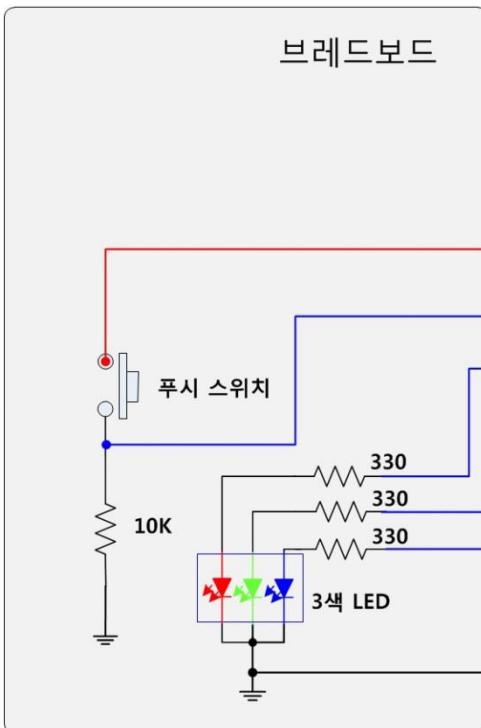
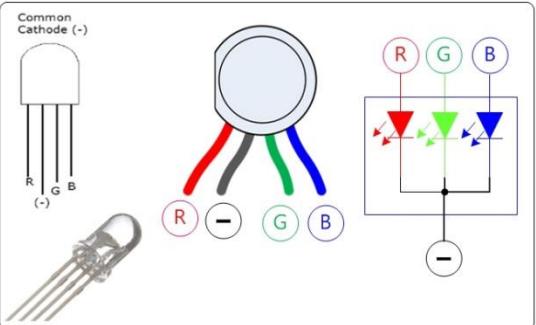
No	부품명	수량	자세한 설명
1	스타트키트 기본세트	1	프로젝트 기본구성
2	RGB LED	1	빛을 발생하는 부품
3	저항 330Ω	3	전류량 조절 부품
4	저항 10KΩ	1	스위치 회로 안정화
5	푸시 스위치	1	스위치 부품



프로젝트 4 : 푸시 버튼 입력에 의한 LED 밝기 바꾸기

➤ 회로도

[부품 방향 주의]



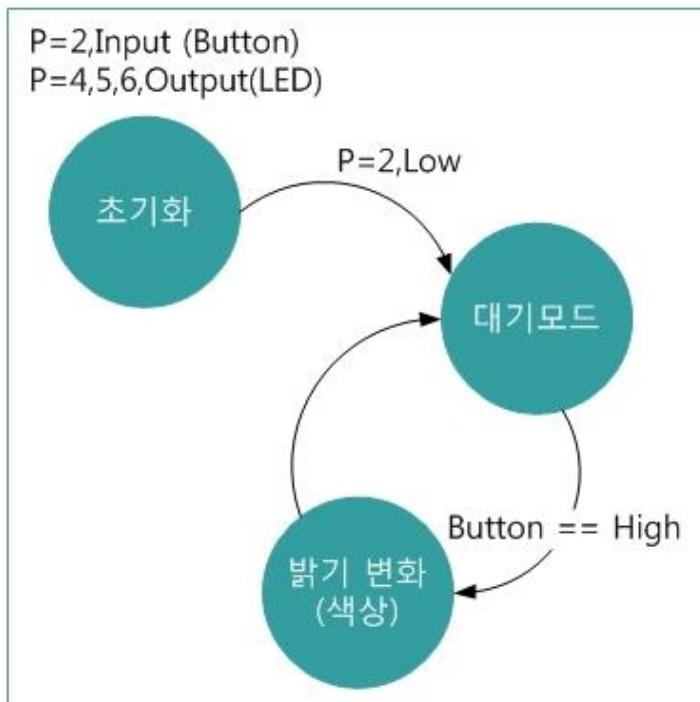
프로젝트 4 : 푸시 버튼 입력에 의한 LED 밝기 바꾸기

➤ 회로 설명 :

- 포트 2번은 초기에는 0V (Low)로 유지되다가, 버튼을 누르면 3.3V가 공급되어 High로 인식합니다.
- 동작 :
 - 전원이 공급되면, 모든 LED가 꺼져 있다가, 푸시 버튼을 누르면 3색 LED가 순차적으로 여러 가지 밝기로 변합니다.
 - 푸시 버튼을 놓으면, 3색 LED는 꺼집니다.

➤ 소프트웨어 구현 방향

- 하나의 포트는 입력포트(버튼)로 3개의 포트는 출력포트(LED)로 설정합니다.
- 버튼 입력을 하면 각 색깔의 LED의 밝기를 달리 조절합니다.
- 3색 밝기 조절에 따라 다양한 색깔을 표현할 수 있습니다.



■ 예제 > BlueinnoSKPT > Circuit - 4

```

/* 푸시 버튼에 의한 LED 밝기 바꾸기

int button = 2; // 버튼이 연결된 핀을 2번으로 설정
int R_LED = 4; // Red LED가 연결된 핀을 4번으로 설정
int G_LED = 5; // Green LED가 연결된 핀을 5번으로 설정
int B_LED = 6; // Blue LED가 연결된 핀을 6번으로 설정
int buttonState = 0; // 버튼 입력의 상태를 저장하기 위함

void setup() {
pinMode(R_LED , OUTPUT); // Red LED 핀을 출력으로 설정
pinMode(G_LED , OUTPUT); // Green LED 핀을 출력으로 설정
pinMode(B_LED , OUTPUT); // Blue LED 핀을 출력으로 설정
pinMode(button, INPUT); // button 핀을 입력으로 설정
}

void loop() {
buttonState = digitalRead(button); // 디지털 입력 핀인 버튼이 눌려졌는지를 읽음

if(buttonState == HIGH) { // 버튼이 High 이면 {} 를 실행합니다.

    analogWrite(R_LED, 255); // Red LED의 밝기를 최대로 합니다.
    delay(500); // 500ms 동안 대기합니다.

    analogWrite(R_LED, 123); // Red LED의 밝기를 중간으로 합니다.
    delay(500); // 500ms 동안 대기합니다.

    analogWrite(R_LED, 0); // Red LED의 밝기를 최소로 합니다.
    delay(500); // 500ms 동안 대기합니다.
}
}

```

■ 예제 >BlueinnoSKPT > Circuit -4

```
analogWrite(G_LED, 255); // Green LED의 밝기를 최대로 합니다.
```

```
delay(500); // 500ms 동안 대기합니다.
```

```
analogWrite(G_LED, 123); // Green LED의 밝기를 중간으로 합니다.
```

```
delay(500); // 500ms 동안 대기합니다.
```

```
analogWrite(G_LED, 0); // Green LED의 밝기를 최소로 합니다.
```

```
delay(500); // 500ms 동안 대기합니다.
```

```
analogWrite(B_LED, 255); // Blue LED의 밝기를 최대로 합니다.
```

```
delay(500); // 500ms 동안 대기합니다.
```

```
analogWrite(B_LED, 123); // Blue LED의 밝기를 중간으로 합니다.
```

```
delay(500); // 500ms 동안 대기합니다.
```

```
analogWrite(B_LED, 0); // Blue LED의 밝기를 최소로 합니다.
```

```
delay(500); // 500ms 동안 대기합니다.
```

```
}
```

```
}
```

➤ 프로그램 설명

- `analogWrite(pin, pwm)` : 아날로그 출력 핀에 pwm (밝기 0~255)의 값을 출력합니다

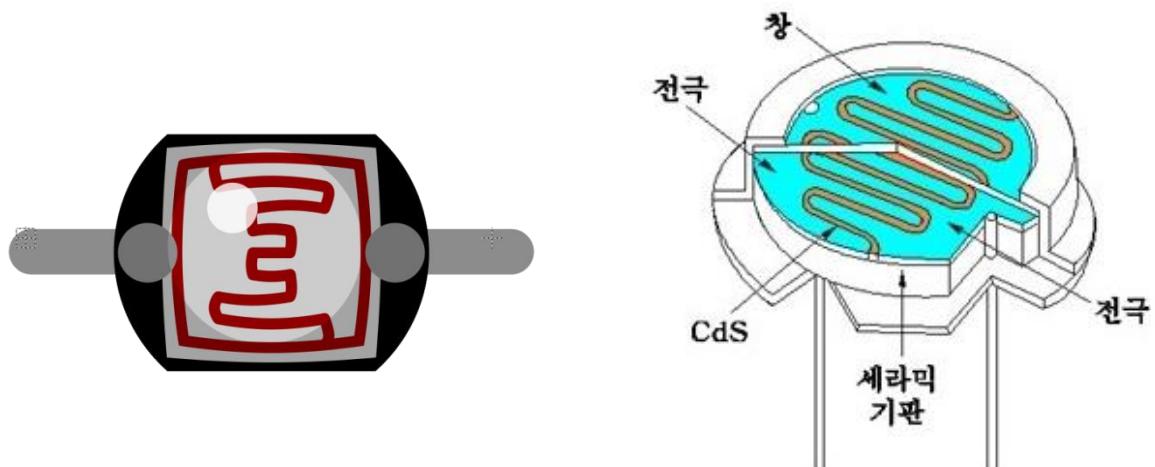
프로젝트 5 : 주변 밝기 변화에 따른 LED 제어하기

➤ 목적 :

- 아날로그 입력(Analog In)의 개념 이해를 위하여, CDS 조도센서를 활용합니다.

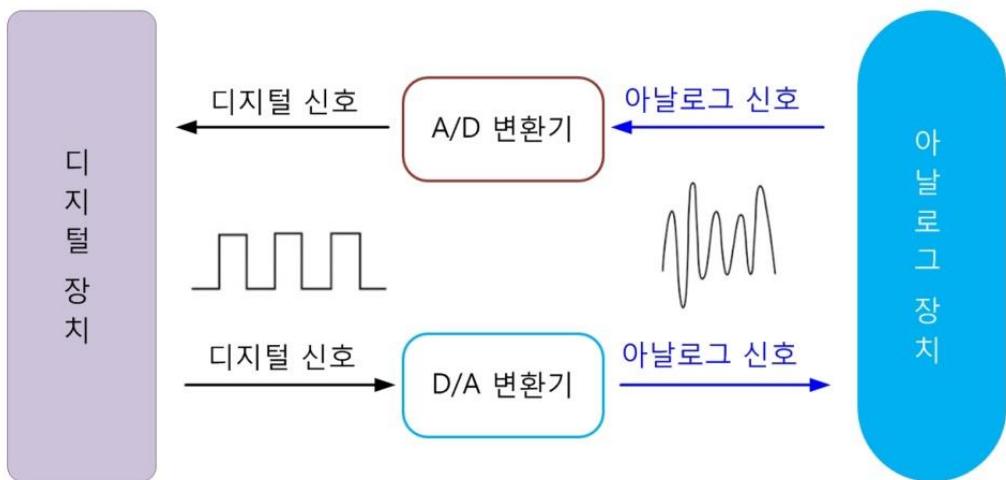
➤ 기술 설명 – CDS 센서

- CDS 조도센서는 저렴한 가격과 활용도 때문에 많이 사용하는 광도전 효과를 이용한 반도체 포토센서입니다.
- 주위가 밝으면 저항이 줄어들고, 주위가 어두우면 저항이 커지는 특성입니다.
- CDS라고 불리는 이유는 CDS Photoresistor를 만드는 주 재료가 카드뮴(CD), 황(S)의 화합물인 황화카드뮴(CDS)이기 때문입니다.

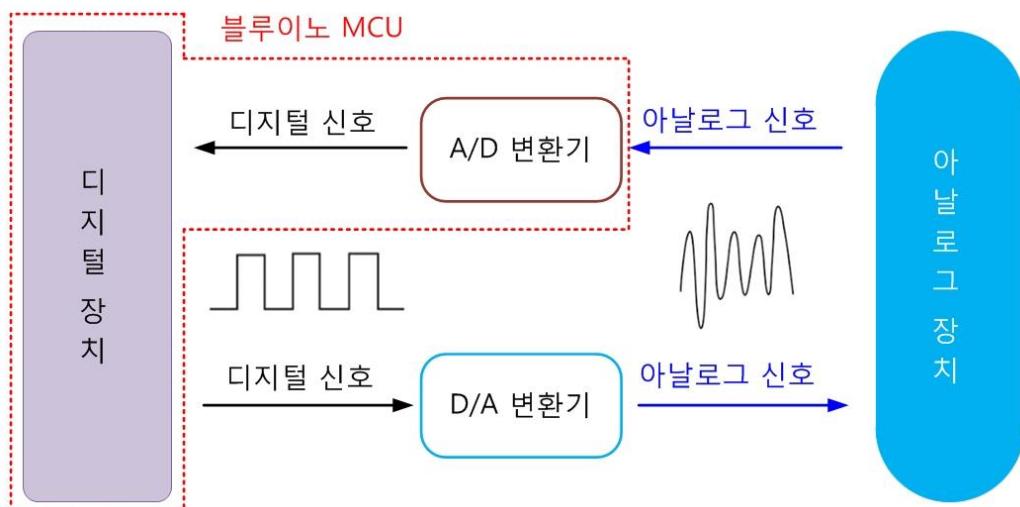


기술 설명 : A/D(ADC) & D/A(DAC) 변환기

- ADC (Analog Digital Converter) 란 연속적인 신호인 아날로그 신호를 부호화된 디지털 신호로 변환하는 장치를 뜻합니다.
이러한 신호는 온도, 압력, 음성, 영상신호, 전압 등을 실생활에서 연속적인 아날로그 신호를 측정하여, 그 신호를 컴퓨터로 입력하고 디지털로 변화하는 장치입니다.



* 블루이노 마이크로 컨터롤러(MCU) 내에는 ADC 10bit가 내장되어 있습니다.
그러므로, 외부에 별도의 ADC를 사용하지 않아도 됩니다.



- DAC (Digital Analog Converter) 란 ADC의 정반대의 개념입니다.

프로젝트 5 : 주변 밝기 변화에 따른 LED 제어하기

▶ 준비물 목록

No	부품명	수량	자세한 설명
1	스타트키트 기본세트	1	프로젝트 기본구성
2	LED (R/Y/G/B)	4	빛을 발생하는 부품
3	저항 330Ω	4	전류량 조절 부품
4	저항 1KΩ	1	조도 분배 전압 회로
5	조도 센서	1	밝기를 감지하는 센서

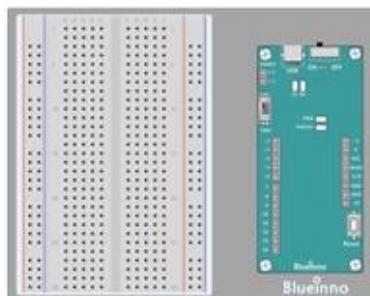
스타트키트 기본세트



점프 케이블



USB 케이블



브레드보드 + 블루이노 기본형



LED



저항 330Ω



저항 1KΩ

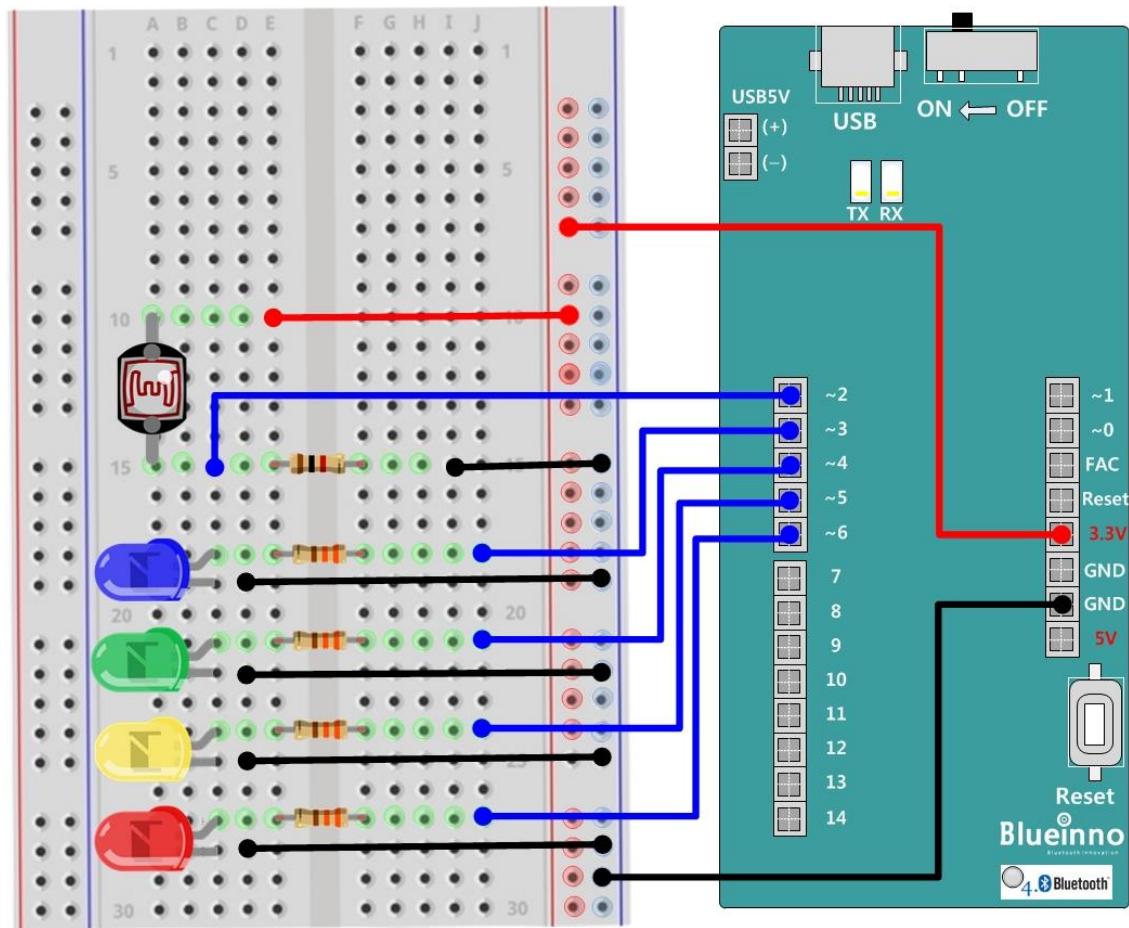
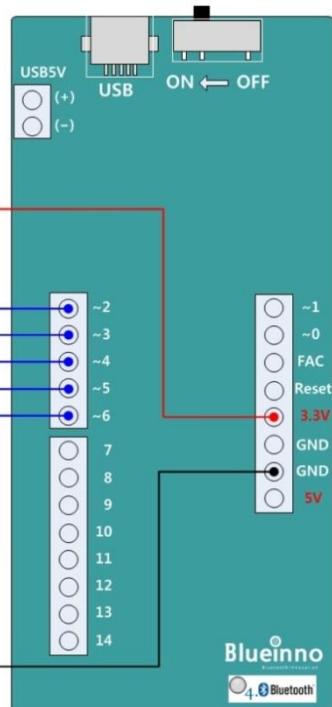
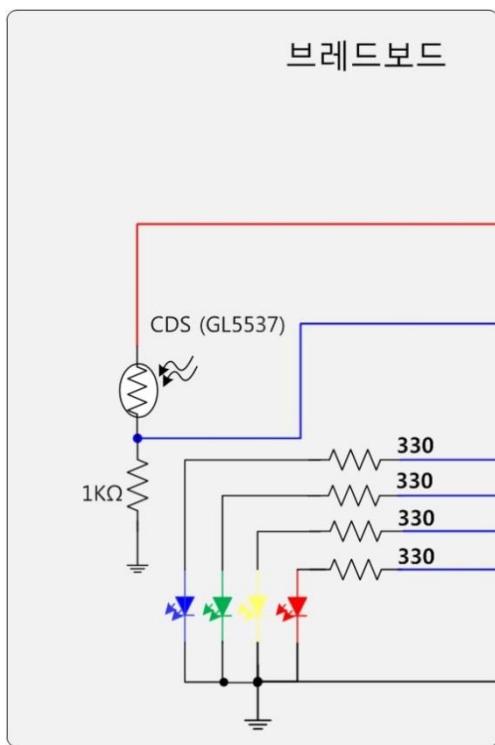
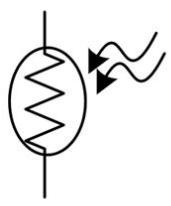


조도 센서

프로젝트 5 : 주변 밝기 변화에 따른 LED 제어하기

➤ 회로도

[부품 방향 없음]



프로젝트 5 : 주변 밝기 변화에 따른 LED 제어하기

➤ 회로 설명 :

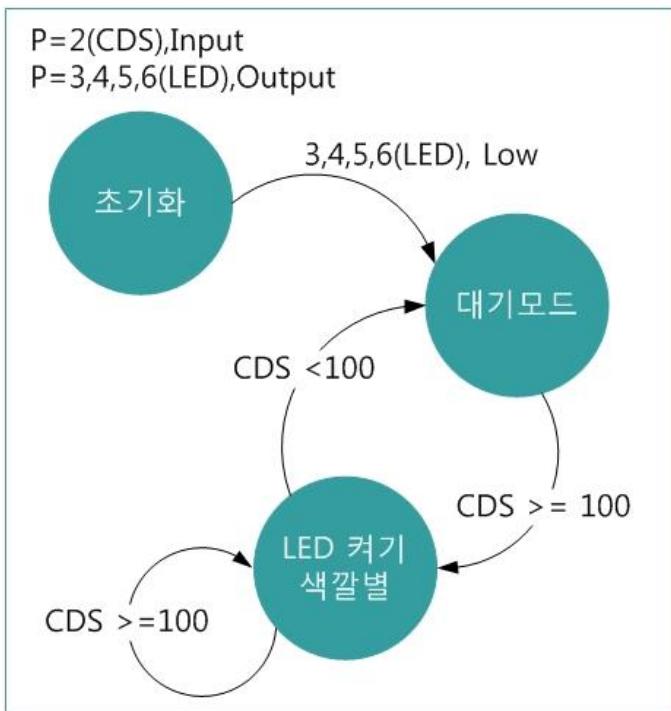
- 빛의 밝기에 따라서 저항 값이 변경되어, 여기에 연결된 5.0V 전압이 센서 저항과 1KΩ의 분배되는 전압이 포트 2번에 인가되면, 내부의 ADC를 통해 디지털 데이터 0~1023 사이의 값을 읽습니다.

➤ 동작 :

- 손으로 CDS 조도센서를 가리면, 밝기에 따라 ADC 값의 영역에 따라서 변화합니다.
- ADC 값에 해당하는 LED 포트 3번, 4번, 5번을 제어하여 켜줍니다.

➤ 소프트웨어 구현 방향

- ADC를 이용하여 CDS에 의해서 받아 들여지는 값에 따라 LED를 조정하게 합니다.
- ADC 값에 따라 각 색깔의 ON을 달리하도록 구성합니다.



▪ 예제 >BlueinnoSKPT > Circuit - 5

```
/* 주변 밝기 변화(CDS 조도 센서의 값)에 따른 LED 제어하기
```

```
*/
```

```
int CDS = 2; // CDS 조도센서가 연결된 핀을 2번으로 설정
```

```
int B_LED = 3; // Blue LED가 연결된 핀을 3번으로 설정
```

```
int G_LED = 4; // Green LED가 연결된 핀을 4번으로 설정
```

```
int Y_LED = 5; // Yellow LED가 연결된 핀을 5번으로 설정
```

```
int R_LED = 6; // Red LED가 연결된 핀을 6번으로 설정
```

```
int sensorValue = 0; // 센서의 값을 저장하기 위함
```

```
void setup() {
```

```
  pinMode(CDS, INPUT); // CDS 핀을 입력으로 설정
```

```
  pinMode(B_LED, OUTPUT); // Blue LED 핀을 출력으로 설정
```

```
  pinMode(G_LED, OUTPUT); // Green LED 핀을 출력으로 설정
```

```
  pinMode(Y_LED, OUTPUT); // Yellow LED 핀을 출력으로 설정
```

```
  pinMode(R_LED, OUTPUT); // Red LED 핀을 출력으로 설정
```

```
}
```

```
void loop() {
```

```
  sensorValue = analogRead(CDS); // 아날로그 CDS 센서 값을 읽음
```

```
  if ( (sensorValue >=100) && (sensorValue < 200)) { // 센서값이 100보다 같거나 크고
```

```
                                // 센서의 값이 200보다 작으면
```

```
    digitalWrite(B_LED, HIGH); // Blue LED 를 High로 출력
```

```
    digitalWrite(G_LED, LOW); // Green LED 를 Low 로 출력
```

```
    digitalWrite(Y_LED, LOW); // Yellow LED 를 Low 로 출력
```

```
    digitalWrite(R_LED, LOW); // Red LED 를 Low 로 출력
```

```
}
```

예제 >BlueinnoSKPT > Circuit - 5

```
else if( sensorValue >=200) && (sensorValue < 400) {  
    // 센서 값이 200보다 같거나 크고, 400보다 작으면  
    digitalWrite(B_LED, LOW); // Blue LED 를 Low 로 출력  
    digitalWrite(G_LED, HIGH); // Green LED 를 High 로 출력  
    digitalWrite(Y_LED, LOW); // Yellow LED 를 Low 로 출력  
    digitalWrite(R_LED, LOW); // Red LED 를 Low 로 출력  
}  
  
else if( sensorValue >=400) && (sensorValue < 550) {  
    // 센서 값이 400보다 같거나 크고, 550보다 작으면  
    digitalWrite(B_LED, LOW); // Blue LED 를 Low 로 출력  
    digitalWrite(G_LED, LOW); // Green LED 를 Low 로 출력  
    digitalWrite(Y_LED, HIGH); // Yellow LED 를 High 로 출력  
    digitalWrite(R_LED, LOW); // Red LED 를 Low 로 출력  
}  
  
else if( sensorValue >=600) {  
    // 센서 값이 600보다 같거나 크면  
    digitalWrite(B_LED, LOW); // Blue LED 를 Low 로 출력  
    digitalWrite(G_LED, LOW); // Green LED 를 Low 로 출력  
    digitalWrite(Y_LED, LOW); // Yellow LED 를 Low 로 출력  
    digitalWrite(R_LED, HIGH); // Red LED 를 High 로 출력  
}  
}
```

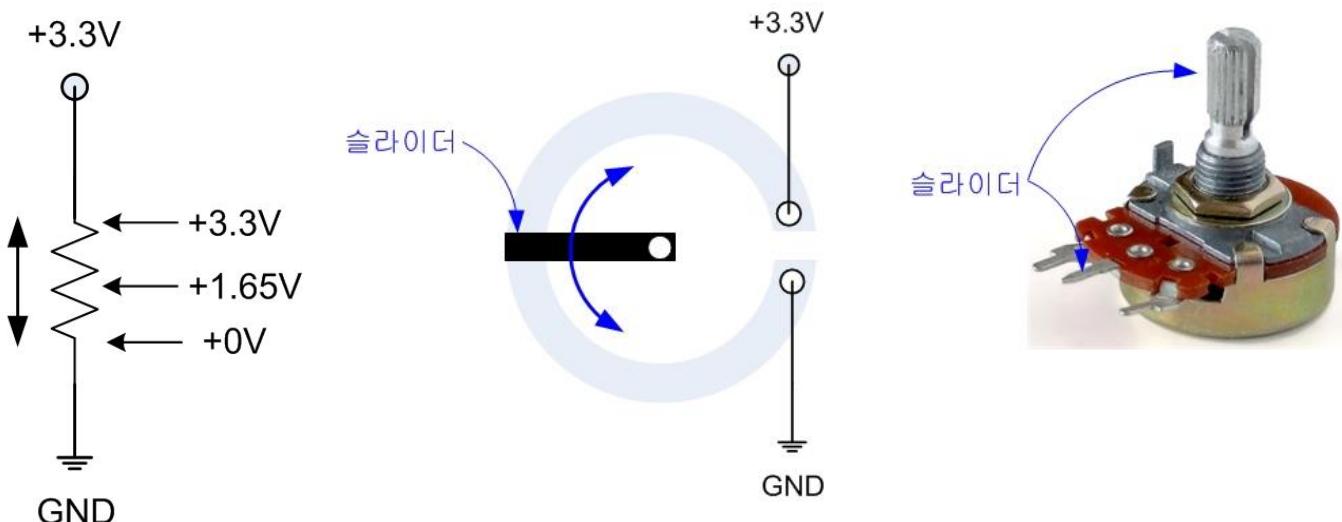
프로젝트 6 : 가변 저항으로 서보 모터 각도 제어하기

➤ 목적 :

- 아날로그 입력의 개념 이해를 위하여, 가변 저항기를 활용합니다.
- 아날로그 출력의 개념 이해를 위하여, 서보 모터를 활용합니다.

➤ 기술 설명 - 가변저항기

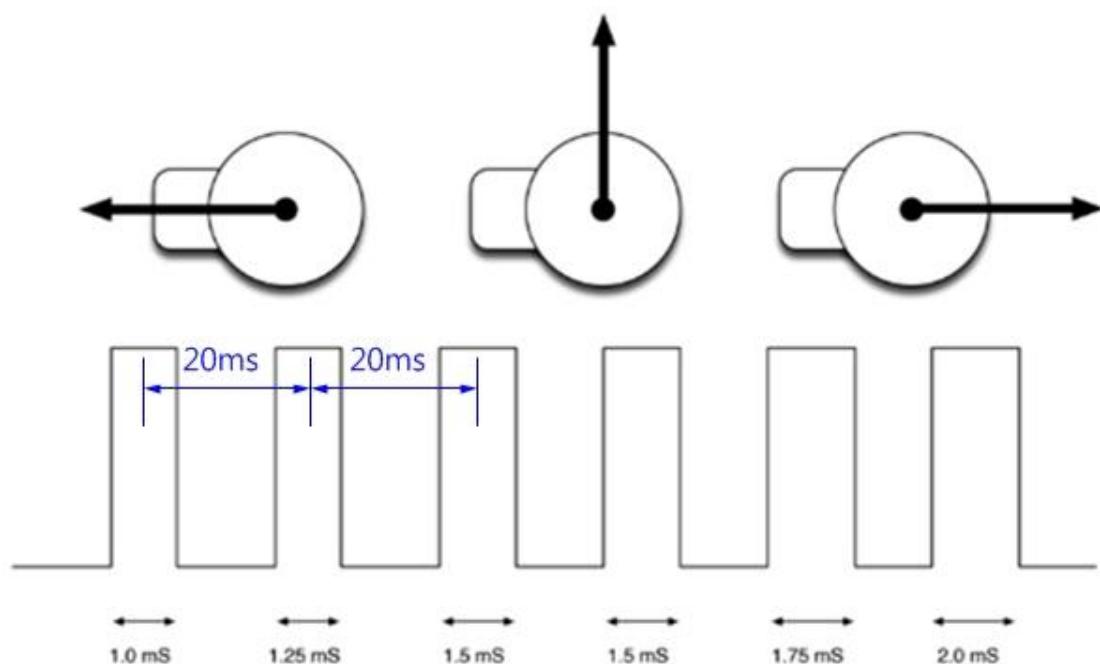
- 가변저항기는 둥글게 생긴 트랙이 있는데, 저항과 같은 역할을 합니다. 가변저항기가 일반저항과 다른 점은 슬라이더라 불리는 중간연결입니다. 이 슬라이더는 가변저항기를 돌리면 같이 돌아 가는데, 3.3V 쪽으로 돌리면 점차 3.3V가 되고 GND 쪽으로 돌리면 0V가 됩니다.



기술 설명 : 서보 모터

- 서보 모터는 다른 모터와 달리 계속해서 회전을 하지 못하지만, 대신, PWM 신호를 사용해서 특정 각도로 움직일 수가 있으며, 그 각도를 유지합니다.
- 모형자동차나 무선헬기에서 방향키로 많이 사용합니다.
- 서보 모터는 전원(+5.0V), 제어신호(PWM), GND로 3라인이 연결되어야 합니다.

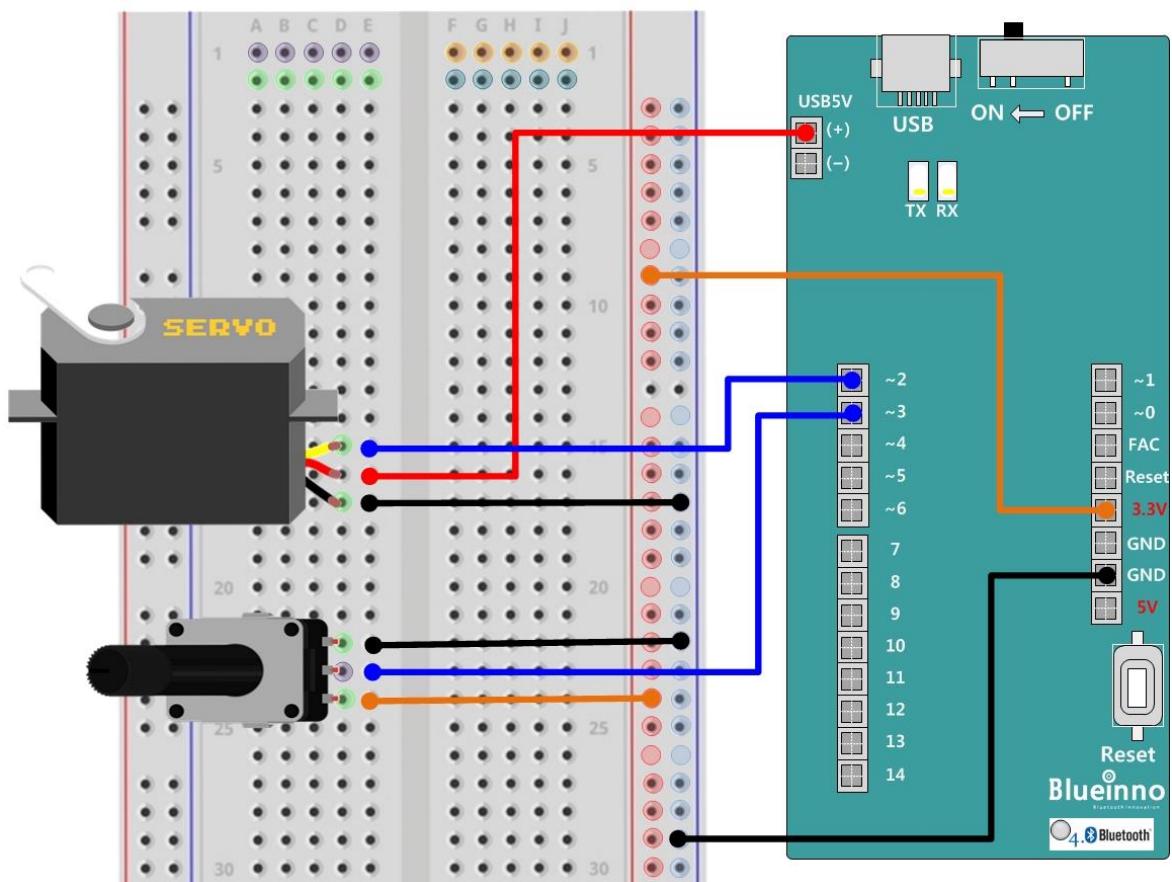
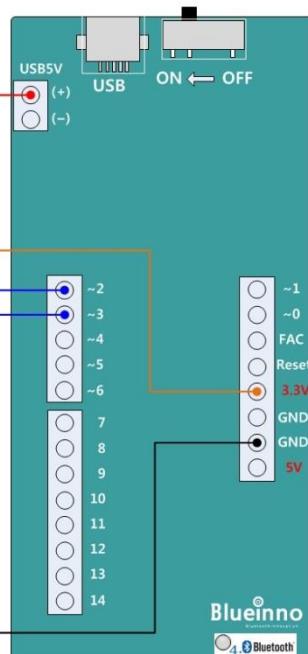
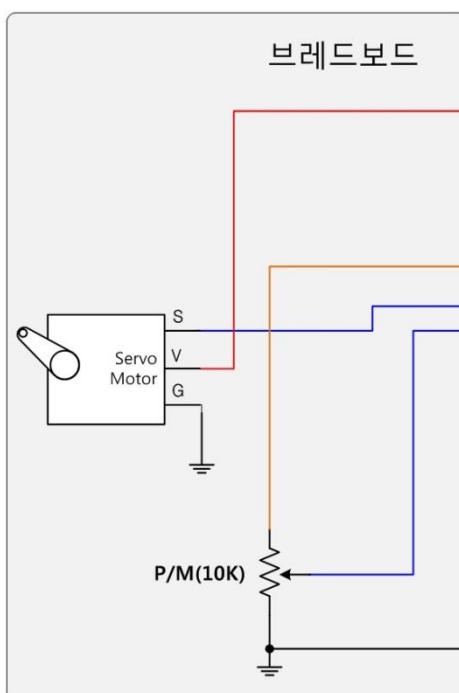
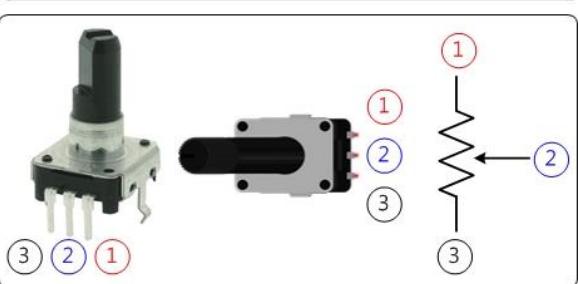
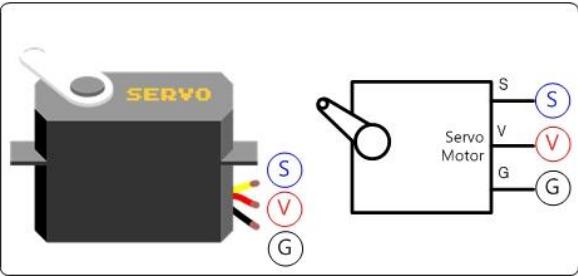
- 서보 모터의 제어신호(PWM)가 표준화되어 있습니다.
- 펄스 폭에 따라서 서보 모터의 각도가 정해집니다.
- 값의 영역은 554 ~ 2,400이며, 단위는 μs 이며,
- 보통은 0~180도 사이를 사용합니다.
 - 1.0ms(1000)의 펄스 폭은 Servo 0도를 설정합니다.
 - 1.5ms(1500)의 펄스 폭은 Servo 90도를 설정합니다.
 - 2.0ms(2000)의 펄스 폭은 Servo 180도를 설정합니다.
 - 현재의 각을 유지하려면, 20ms 간격으로 동일한 펄스 폭을 출력하여 주어야 합니다.



프로젝트 6 : 가변 저항으로 서보 모터 각도 제어하기

회로도

[부품 방향 주의]



프로젝트 6 : 가변 저항으로 서보 모터 각도 제어하기

➤ 회로 설명 :

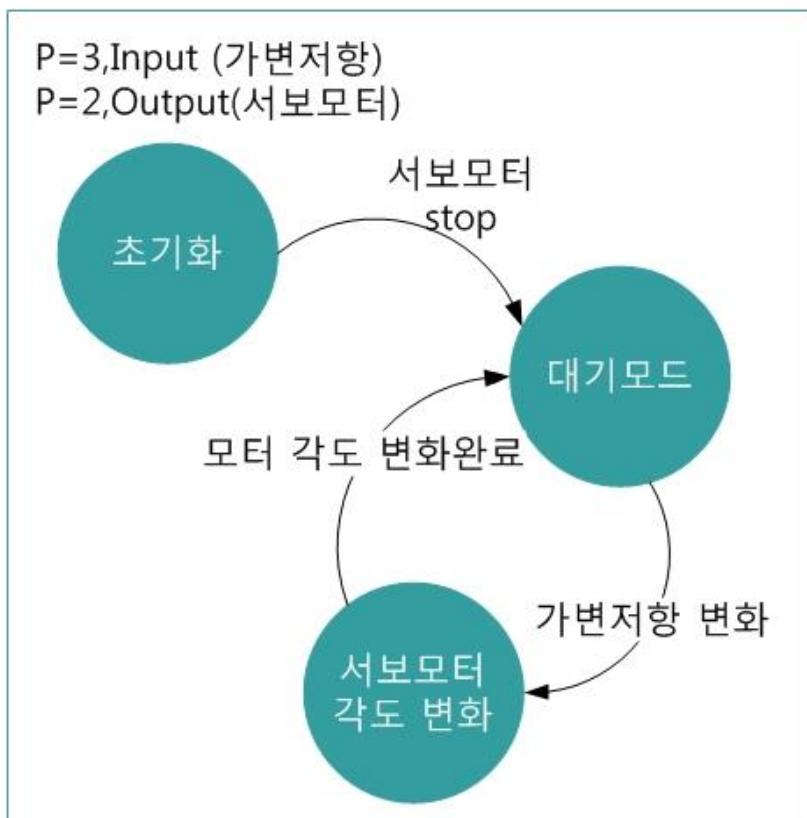
- 가변저항기의 슬라이드 위치에 따라서, 분배되는 전압이 포트 3번에 인가되면 내부의 ADC 를 통해 디지털 데이터 0~1023 사이의 값을 읽습니다.
- 그리고, 이와 연동된 서보 모터의 각도 값을 포트 2번에서 PWM으로 출력합니다.

➤ 동작 :

- 손으로 가변저항기를 돌리면, 이와 연동되는 서보 모터의 각도가 같이 변합니다.

➤ 소프트웨어 구현 방향

- 가변 저항의 값을 읽어서 각도 값으로 변경하여 줍니다.
- 해당 각도를 서보 모터에 전달하여 원하는 각도로 움직이게 합니다.



■ 예제 > BlueinnoSKPT > Circuit -6

```

/* 가변 저항으로 서보 모터 각도 제어하기
*/
#include <Servo.h> // 서보모터 라이브러리 사용을 위해, 헤더 파일을 추가 함
int Potent = 3; // 가변 저항이 연결된 핀을 3번으로 설정

Servo s1; // 서보모터 제어 라이브러리를 사용하기 위한 변수 선언
int sensorValue = 0; // 센서의 값을 저장하기 위함

void setup() {
    s1.attach(2); // 서보모터에 펄스를 공급해줄 포트를 2번으로 설정
    pinMode(Potent, INPUT); // 가변저항 핀을 입력으로 설정
}

void loop() {
    sensorValue = analogRead(Potent); // 아날로그 가변저항기의 저항 값을 읽음

    s1.write(map(sensorValue, 0, 1023, 0, 179));
    delay(200);
}

```

➤ 프로그램 설명

- #include <Servo.h> : 서보 모터 라이브러리를 사용하기 위해서 반드시 추가해야 합니다. 아두이노 개발환경에서 인클루드 헤더 파일은 대소문자를 구분하므로, 유의해서 코드를 작성해야 합니다.
- s1.write(map(입력 값, 시작범위, 끝 범위, 출력 시작 값, 출력 끝값)) : 가변 저항의 입력된 값을 0~1023 범위의 값을 읽어서, 0~179 범위의 출력 값을 서보 모터를 제어하기 위해 2번 핀으로 보냅니다.

프로젝트 7 : 가변 저항으로 DC 모터 속도 제어하기

➤ 목적 :

- 아날로그 입력의 개념 이해를 위하여, 가변저항기를 활용합니다.
- 아날로그 출력의 개념 이해를 위하여, DC 모터를 활용합니다.

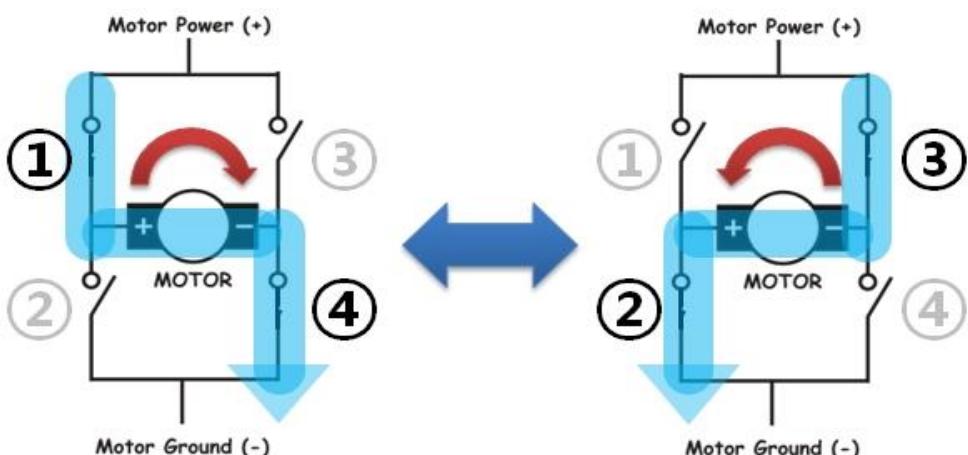
➤ 기술 설명 : DC 모터

- DC 모터는 2라인(전원, GND)만 연결하면 됩니다.
- DC 모터는 자동차의 뒷바퀴 회전체나 헬기의 프로펠러 회전체로 많이 활용됩니다.

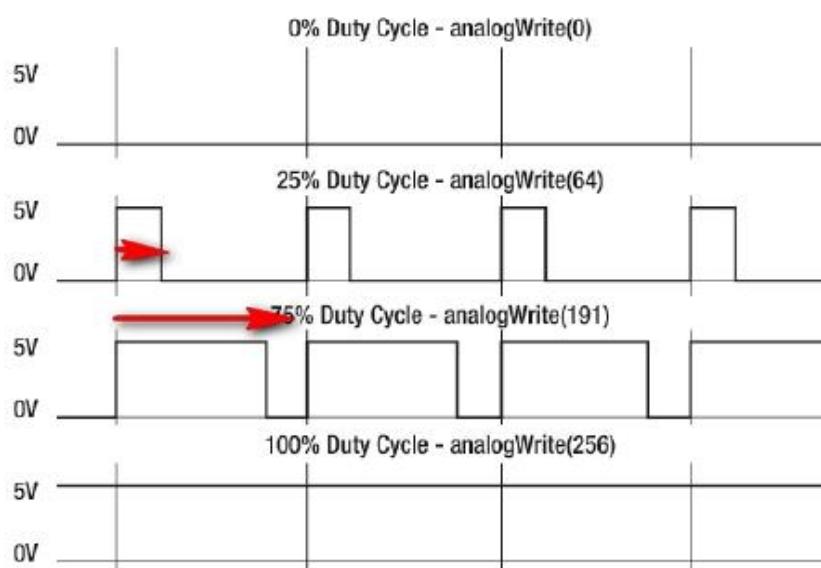


기술 설명 : DC 모터

- 인가하는 전원의 방향에 따라서 방향이 순방향과 역방향으로 변경됩니다. 보통은 방향전환을 위해서 Tr이나 FET를 사용할 수 있지만, 이 회로에는 잘못 제어 시 부품이나 회로가 손상될 위험이 도사리고 있으므로, L293 같은 H브리지 직접회로(IC)를 사용하는 것이 간편하고 안전합니다. 생김새가 H 모양으로 H 브리지(Bridge) 회로 혹은, 이것을 직접회로로 만든 부품이 전류의 소모 용량에 따라서, 여러 부품(L298, L293)들이 있습니다.



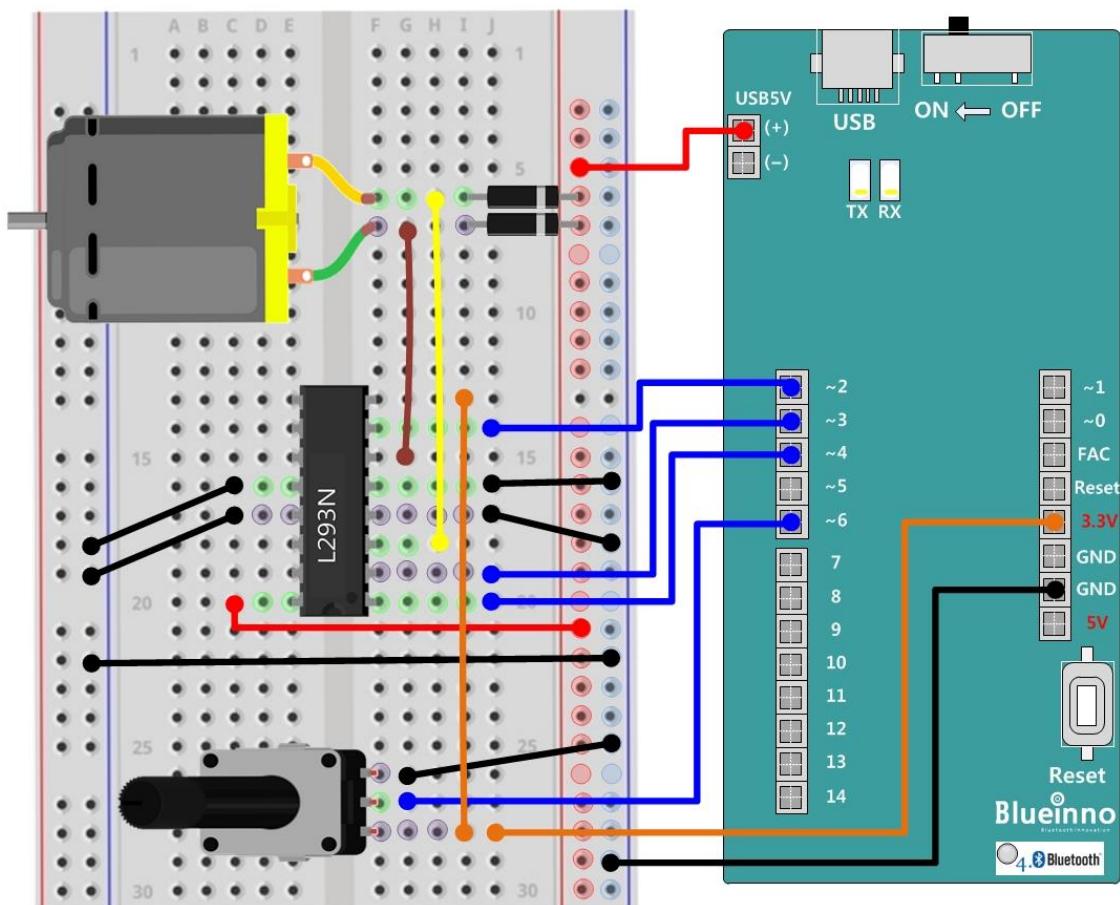
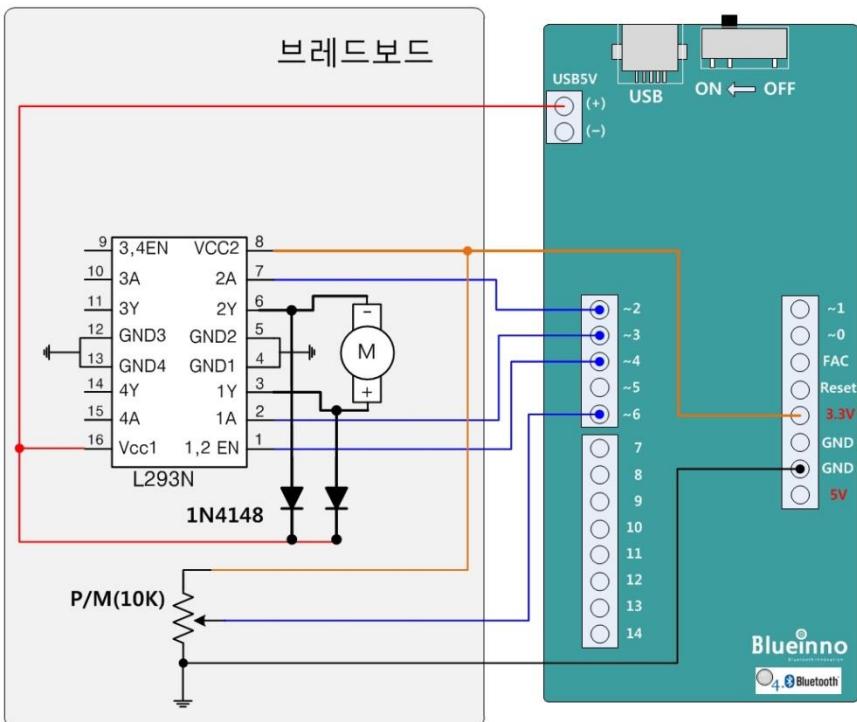
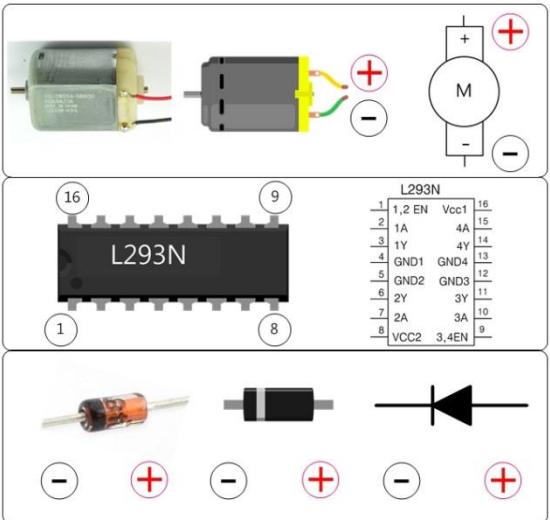
- 속도제어는 PWM 신호로 Duty 비에 의해서 제어됩니다. 0%일 때 거의 움직이지 않으며, 100%일 때 가장 빠르게 회전합니다.



프로젝트 7 : 가변 저항으로 DC 모터 속도 제어하기

➤ 회로도

[부품 방향 주의]



프로젝트 7 : 가변 저항으로 DC 모터 속도 제어하기

➤ 회로 설명 :

- 가변저항기의 슬라이드 위치에 따라서, 분배되는 전압이 포트 6번에 인가되면 내부의 ADC를 통해 디지털 데이터 0~1023 사이의 값을 읽습니다.
- 그리고, 이와 연동된 DC 모터의 속도 값을 포트 4번에서 PWM으로 출력합니다.
- 포트 2번과 3번의 출력 값으로 모터의 방향을 결정합니다.

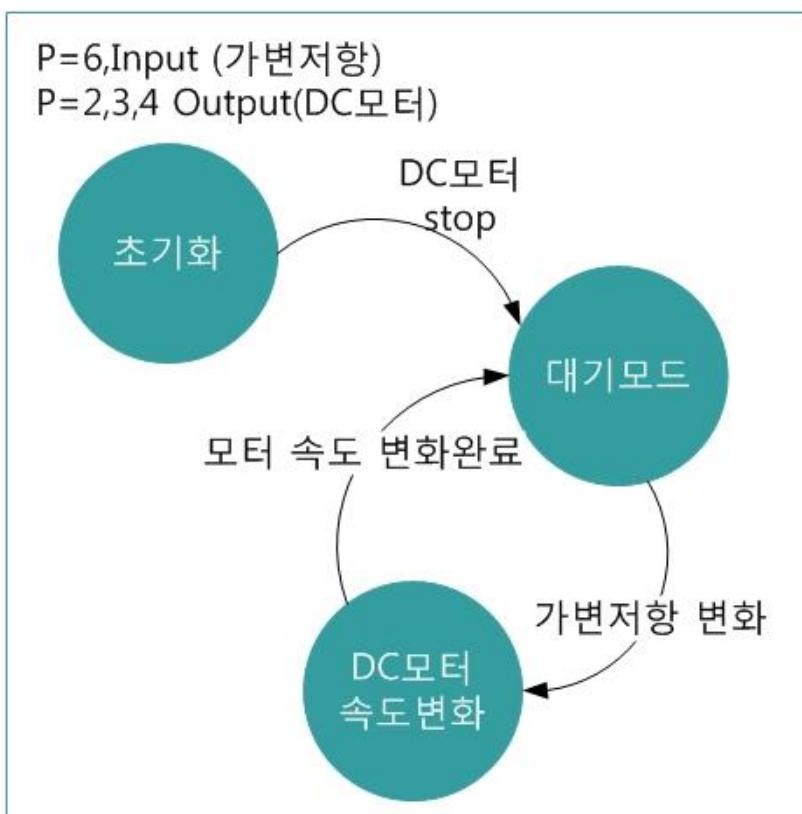
❖ **주의사항 :** 보드에 공급되는 전원이 DC 5V, 300mA 정도이며, 다른 DC 모터를 사용시 과도한 전류 사용으로 시스템이 제대로 동작하지 않을 수 있으므로 DC 모터 사용시 전기적 사양을 확인 바랍니다.

➤ 동작 :

- 손으로 가변저항기를 돌리면, 이와 연동되는 DC 모터의 속도가 같이 변합니다.

➤ 소프트웨어 구현 방향

- 가변 저항의 값을 읽어서 DC모터의 속도 값으로 변경하여 줍니다.
- 해당 값을 DC모터에 전달하여 원하는 속도로 움직이게 합니다.



■ 예제 >BlueinnoSKPT > Circuit - 7

```

/* 가변 저항으로 DC 모터의 속도 제어하기
*/
int Enable = 4; // 동작(속도) 스위칭이 연결된 핀을 4번으로 설정
int In1 = 3; // 모터 순방향 제어 핀이 연결된 핀을 3번으로 설정
int In2 = 2; // 모터 역방향 제어 핀이 연결된 핀을 2번으로 설정
int Potent = 6; // 가변저항기가 연결된 핀을 6번으로 설정

int sensorValue = 0; // 센서에서 나오는 값을 저장하기 위함

void setup() {
    pinMode(Enable, OUTPUT); // 동작(속도) 스위칭 핀을 출력으로 설정
    pinMode(In1, OUTPUT); // 순방향 제어 핀을 출력으로 설정
    pinMode(In2, OUTPUT); // 역방향 제어 핀을 출력으로 설정
    pinMode(Potent, INPUT); // 가변 저항의 연결 핀을 입력을 설정
}

void loop() {
    sensorValue = analogRead(Potent); // 가변저항 값을 읽음
    int speed = map(sensorValue, 0, 1023, 50, 255);
    // 속도는 가변저항 값을 0 ~ 1023까지 읽고, PWM 출력을 50~ 255로 정합니다.

    analogWrite(Enable, speed); // 동작(속도) 스위칭을 스피드로 출력합니다.
    digitalWrite(In1, HIGH); // 순방향으로 돌아가게 High 출력을 합니다.
    digitalWrite(In2, LOW); // 순방향으로 돌아가게 Low 출력을 합니다.
}

```

➤ 프로그램 설명

- `analogWrite(speed)` 값을 0 ~255 까지 설정할 수가 있으나, DC 모터의 특성상 너무 값이 낮으면 회전이 안될 경우가 있습니다. 여기서는 50부터 시작합니다.

프로젝트 8 : 온도 값을 PC 모니터에 표시하기

➤ 목적 :

- 시리얼 모니터 프로그램을 이용하여, 보드의 센서 정보를 PC 화면에 표시할 수 있습니다.

➤ 기술 설명 : 시리얼 통신

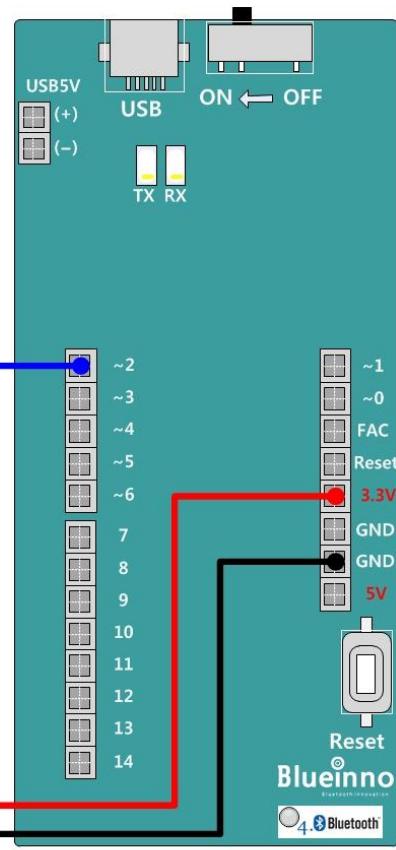
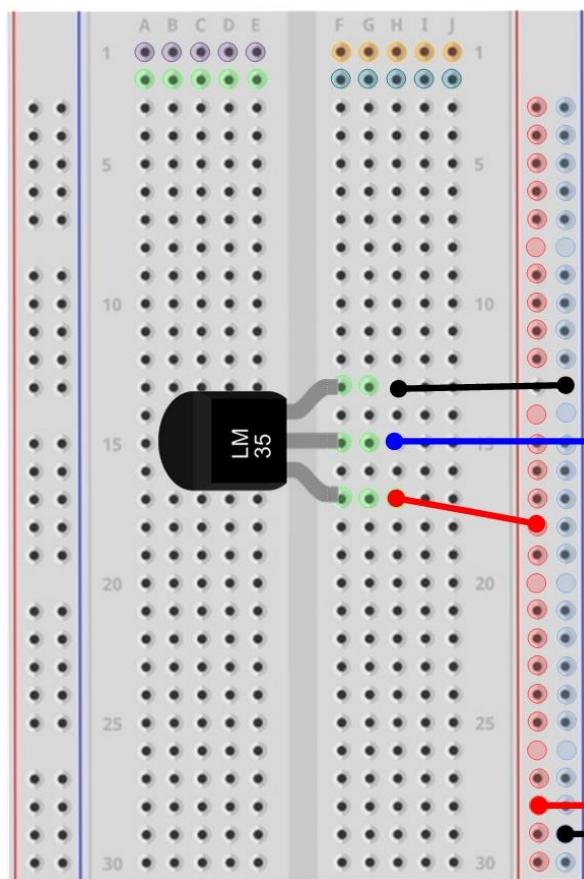
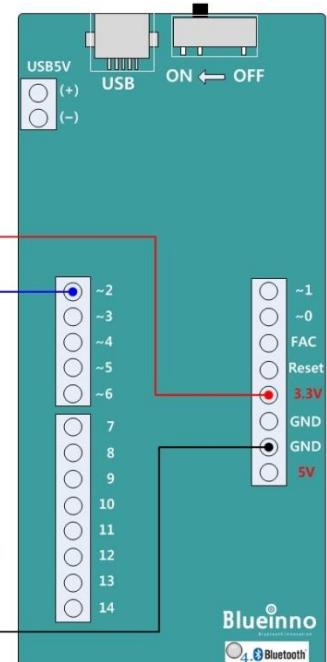
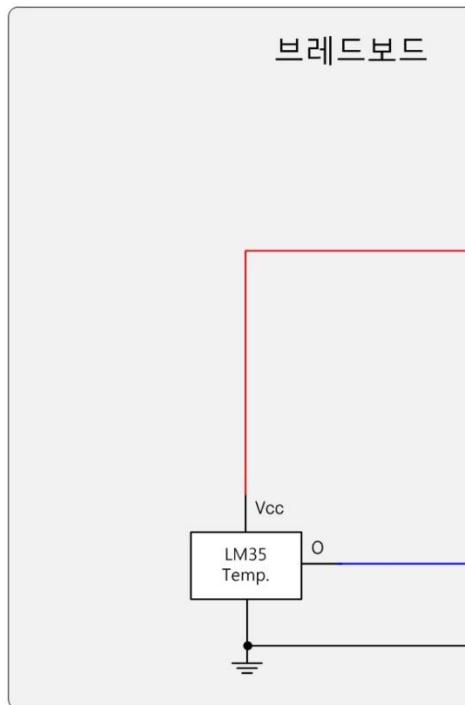
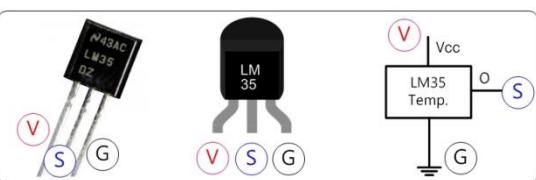
- PC와 보드간의 정보 교환을 위해 물리적으로 연결된 것을 시리얼 통신이라고 합니다.
 - 보드에 입력된 센서의 값을 PC와 연결된 USB 케이블을 통해 메시지를 주고 받을 수 있습니다.
 - PC에서 글자나 명령을 전달해서, 보드가 명령을 수행하게 할 수 있습니다.
 - 보드의 상태를 디버깅하기 쉽고, 보드를 컴퓨터에서 제어할 수 있게 하여 줍니다.
 - PC의 USB 포트를 잘 설정해 주어야 하며, 보드간의 전송 속도도 맞추어야만 통신이 가능합니다.



프로젝트 8 : 온도 값을 PC 모니터에 표시하기

➤ 회로도

[부품 방향 주의]



프로젝트 8 : 온도 값을 PC 모니터에 표시하기

➤ 회로설명 :

- 온도센서는 주변의 온도를 감지하여, 전압으로 포트 2번으로 입력됩니다.
- 내부 ADC에 의해서, 디지털 데이터 0~1023 사이의 값으로 읽고, 마이크로 컨터롤러에서 온도로 계산하여, USB 케이블 포트로 온도 값을 전송합니다.

➤ 동작 : PC 모니터에 표시 하기

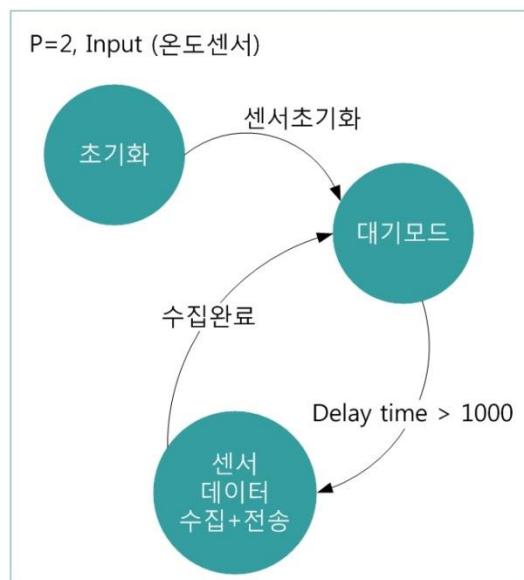
- USB 케이블로 보드와 PC 간의 시리얼로 연결되어, 온도 센서의 값들이 표현됩니다.



➤ 소프트웨어 구현 방향

- analogRead함수를 이용하여 온도센서의 값을 읽습니다.
- Serial함수를 사용하여 측정된 값을 시리얼 모니터를 통하여 출력합니다.

■ 예제 >BlueinnoSKPT > Circuit -8



// 온도 값을 PC 모니터에 표시하기

```
const int temperaturePin = 2; // 온도센서가 연결된 핀을 2번으로 설정
```

```
void setup()
```

```
{
```

```
pinMode(temperaturePin, INPUT); // 온도센서 핀을 입력으로 설정
```

```
Serial.begin(9600); // 시리얼 데이터를 주고 받을 수 있게 준비함
```

```
}
```

// 속도=9600(bps)

```
void loop()
```

```
{
```

```
int reading = analogRead(temperaturePin); // 아날로그 입력 온도 값을 읽음
```

```
Serial.print((3.0*reading)/1024.0); // LM35 온도 계산법 =
```

// 온도 = (3.0*reading*100.0)/1024.0

```
Serial.println("C"); // 컴퓨터에 "C" 문자를 시리얼 통신으로 보냄
```

```
Delay(1000); // 시리얼 포트의 버퍼를 방지하기 위함
```

```
}
```

프로그램 설명

- Serial.begin(speed) : 시리얼 데이터를 주고 받기 위해 통신 속도를 설정합니다.
- Serial.print(data) : 시리얼포트로 데이터를 전송

프로젝트 9 : LCD에 문자 표시하기

➤ **목적 :**

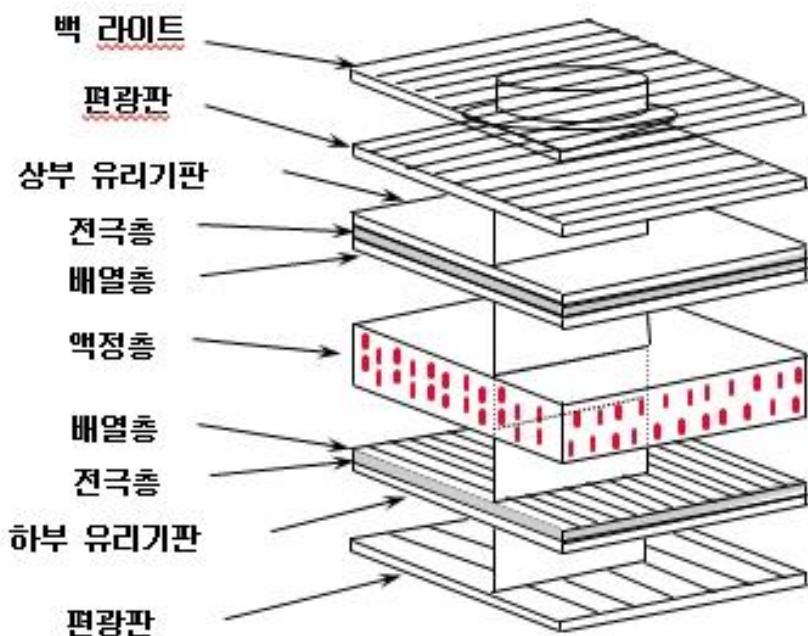
- LCD 소자를 통한 여러 가지 정보(글자나 그림)를 표시하는 방법을 이해합니다.

➤ **기술 설명 : LCD (Liquid Crystal Display)**

- LCD는 우리 생활의 많은 부분에 사용되는 부품으로 손목시계 부터 뉴스와 드라마를 보는 TV, 지하철역의 광고판까지 다양합니다.

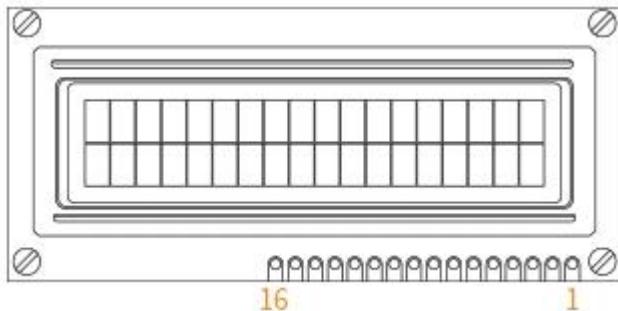


- LCD란 액체와 고체 사이의 중간적 특성을 갖는 액정의 전기, 광학적 성질을 표시 장치에 응용한 것입니다. 액체처럼 유동성을 갖는 유기분자인 액정이 결정처럼 규칙적으로 배열되는 상태를 이용한 것으로, 이 분자 배열이 외부 전극에 의해 변화되는 성질을 이용하여 표시소자로 만든 것을 ‘액정 디스플레이’라고 합니다.
- LCD 구조입니다.



기술 설명 : LCD(Liquid Crystal Display)

- LCD는 복잡 인터페이스를 가지고 있습니다. 이 말은 화면을 조정하기 위해서는 동시에 여러 개의 인터페이스 핀을 조작해야 한다는 것입니다.



[LCD 핀맵]

1	VSS	GND
2	VDD	5V 전원
3	V0	LCD 밝기 조절 핀
4	RS	레지스터 선택 핀
5	R/W	읽기 / 쓰기 모드 선택핀
6	E	레지스터에 쓰기 위한 핀
7	DB0	데이터 입출력 (4bit or 8bit)
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	LED+	LED (BackLight) 전원
16	LED-	LED Gnd 핀

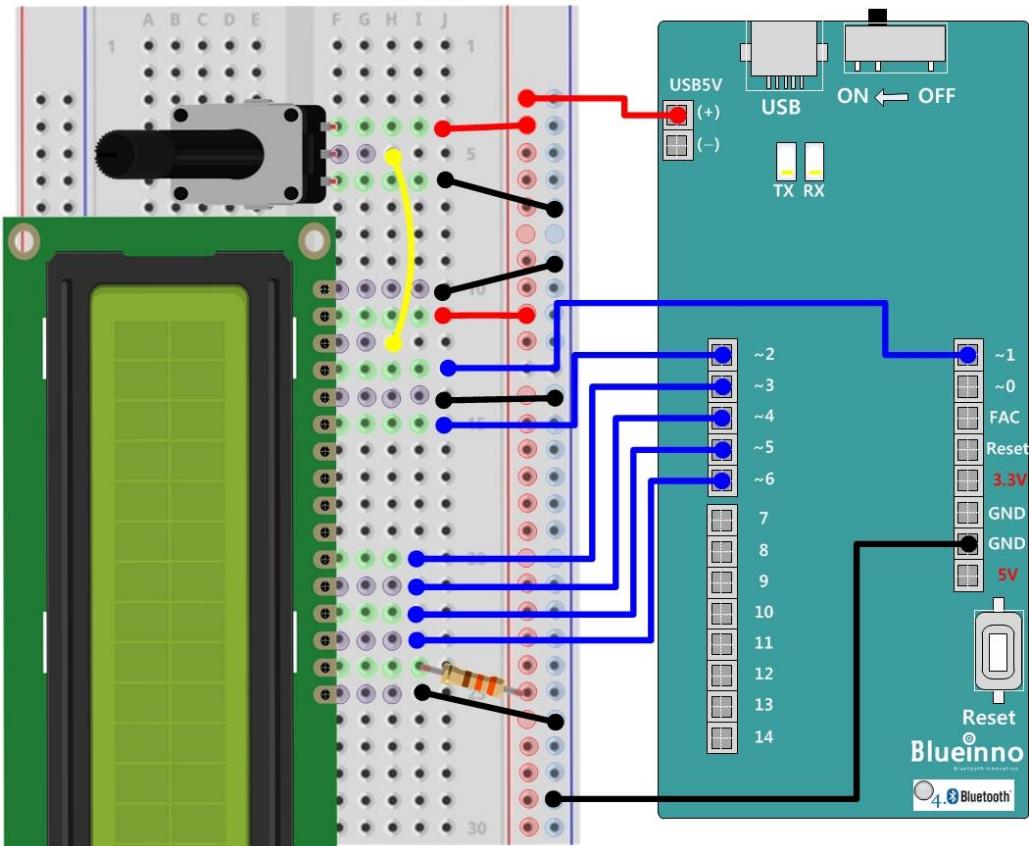
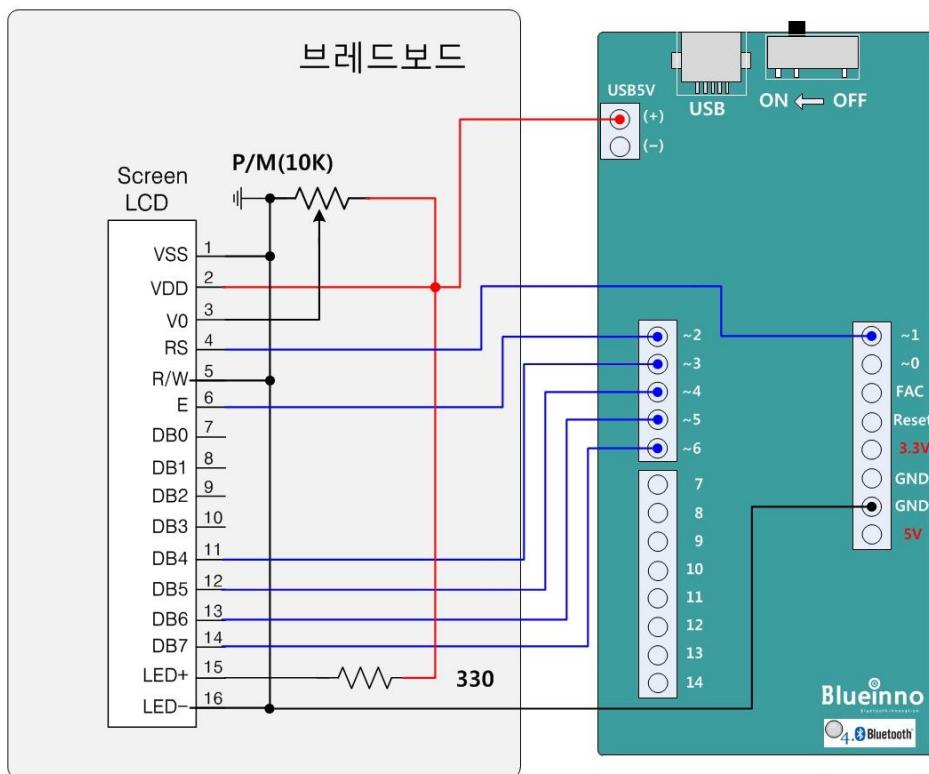
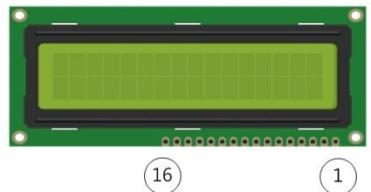
- 화면을 조정하는 과정은 화면에 쓰고자 하는 data를 data register에 입력하고, 그런 다음 명령 register에 명령을 주는 과정으로 이루어져 있습니다. 그러나 Library를 이용하면 작업이 쉬워지고 저번에 깔려 있는 명령에 대해서는 몰라도 되게 해줍니다.

프로젝트 9 : LCD에 문자 표시하기

➤ 회로도

[부품 방향 주의]

VSS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
VDD																
RS																
R/W																
E																
DB0																
DB1																
DB2																
DB3																
DB4																
DB5																
DB6																
DB7																
LED+																
LED-	16															



프로젝트 9 : LCD에 문자 표시하기

➤ 회로 설명

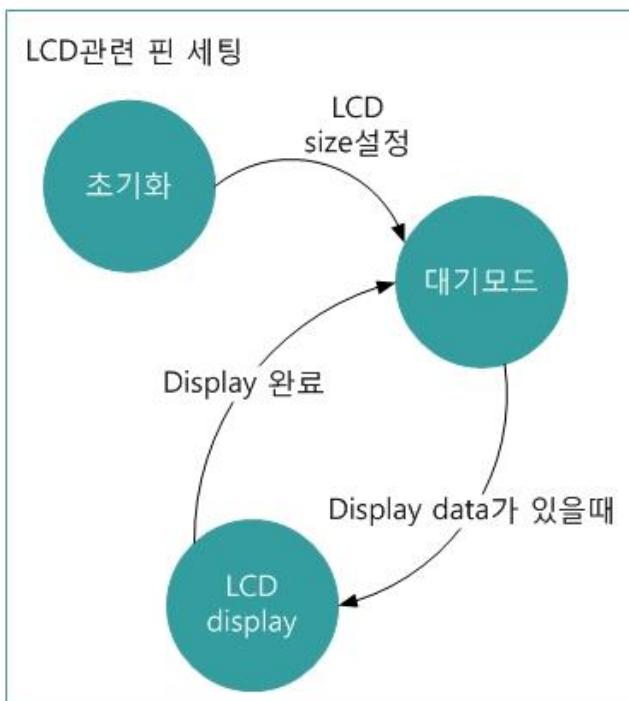
- LiquidCrystal library는 Hitachi HD44780 driver와 호환되는 LCD를 제어할 수 있게 합니다.
- Hitachi-compatible LCDs는 두 가지 모드로 동작하는데 4-bit 혹은 8-bit입니다.
- 4-bit 모드는 블루이노의 일곱 개의 핀이 필요하고 8-bit 모드의 경우 11개의 핀이 필요합니다. 예제는 4-bit 모드입니다.

➤ 동작 : LCD에 표시 하기

- 전원이 공급되면, LCD에 "Hello Blueinno!" 를 반복적으로 출력하게 하는 것입니다.

➤ 소프트웨어 구현 방향

- LiquidCrystal.h를 사용하여 LCD 구현에 필요한 정보를 알아봅니다.
- Lcd함수의 핀 설정, 초기화, 출력에 관한 내용을 사용하여 원하는 문자를 LCD에 출력하게 합니다.



▪ 예제 >BlueinnoSKPT > Circuit - 9

```
/* LCD 에 문자 표시하기
*/
#include <LiquidCrystal.h> // LCD 라이브러리를 사용하기 위한 헤더 파일을 추가
LiquidCrystal lcd(1, 2, 3, 4, 5, 6); // LCD와 보드의 핀 번호를 적어 줍니다.
                                         // LCD (RS, Enable, Data Bit 4, 5, 6, 7 )

void setup()
{
}

void loop()
{
    lcd.setCursor(0,1); // LCD에 Display할 좌표를 정해 줍니다.
    lcd.begin(16, 2); // LCD 데이터를 2행 16열로 표시합니다.
    lcd.clear(); // LCD 화면을 지웁니다.
    lcd.print("hello Blueinno!"); // LCD 화면에 " Hello Blueinno!"를 표시합니다.
    delay(1000); //1초간 유지
    lcd.clear(); // LCD 화면을 지웁니다.
    delay(1000); // 1초간 유지
}
```

프로젝트 10 : 아이비콘(iBeacon) 만들기

➤ 목적 :

- 스마트폰과 보드 간의 무선 통신, 블루투스4.0 통신을 이해합니다.

➤ 회로 설명 :

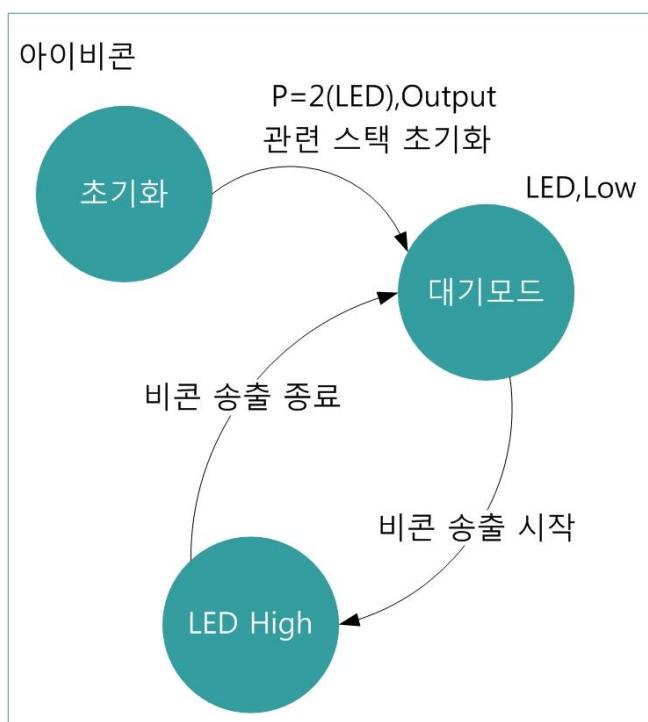
- 일정시간 간격으로 보드는 비콘 데이터를 송출하며, 이때 LED는 계속 켜져 있습니다.

➤ 동작 : **스마트폰 응용앱 사용하기(86 페이지)** 를 참조하세요!

- 스마트폰과 보드간에 연결(페어링) 절차가 필요 없습니다.
- 스마트폰은 보드의 아이비콘 관련 데이터를 받아서, 이와 관련된 정보를 스마트폰에 표시합니다. (예 ; UUID, Major, Minor, RSSI)

➤ 소프트웨어 구현 방향

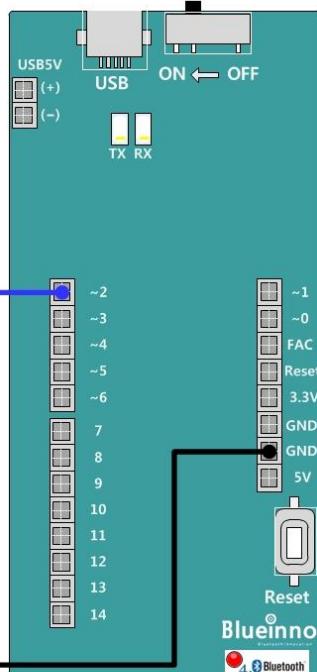
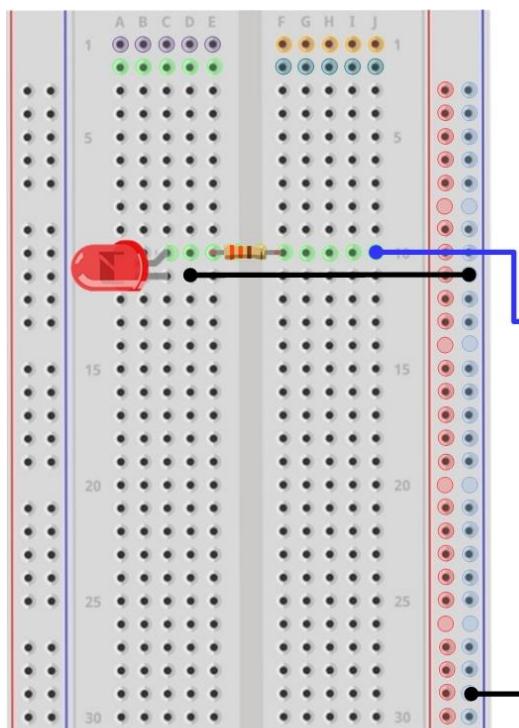
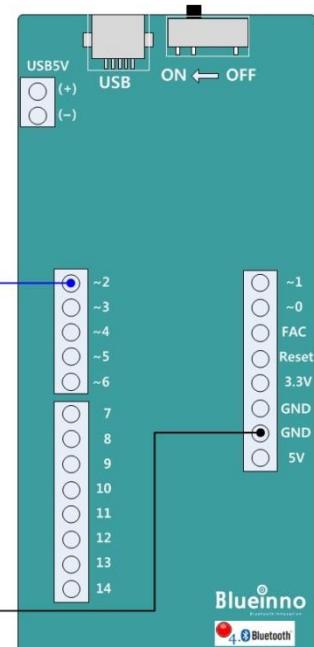
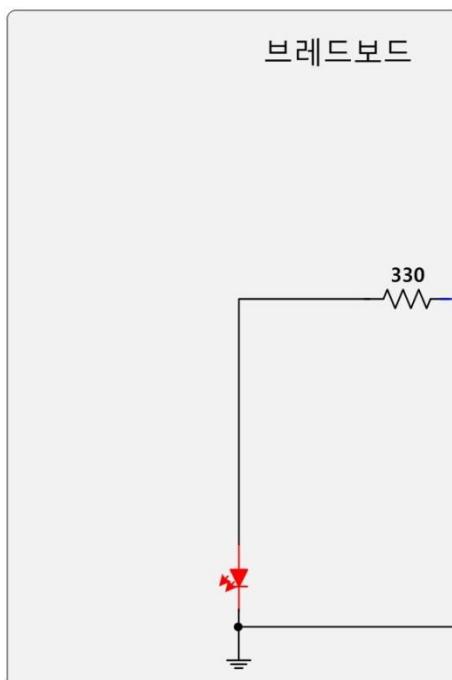
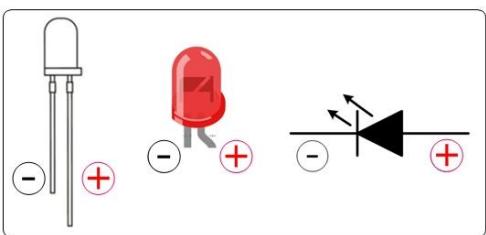
- 아이비콘을 사용하기 위한 헤더파일 추가(RFduinoBLE.h) 및 관련 함수에 대하여 설정합니다.
- RFduinoBLE_onAdvertisement()함수를 이용하여 비콘이 발생할 때의 상태를 확인하여 LED 제어를 합니다.



프로젝트 10 : 아이비콘(iBeacon) 만들기

➤ 회로도

[부품 방향 주의]

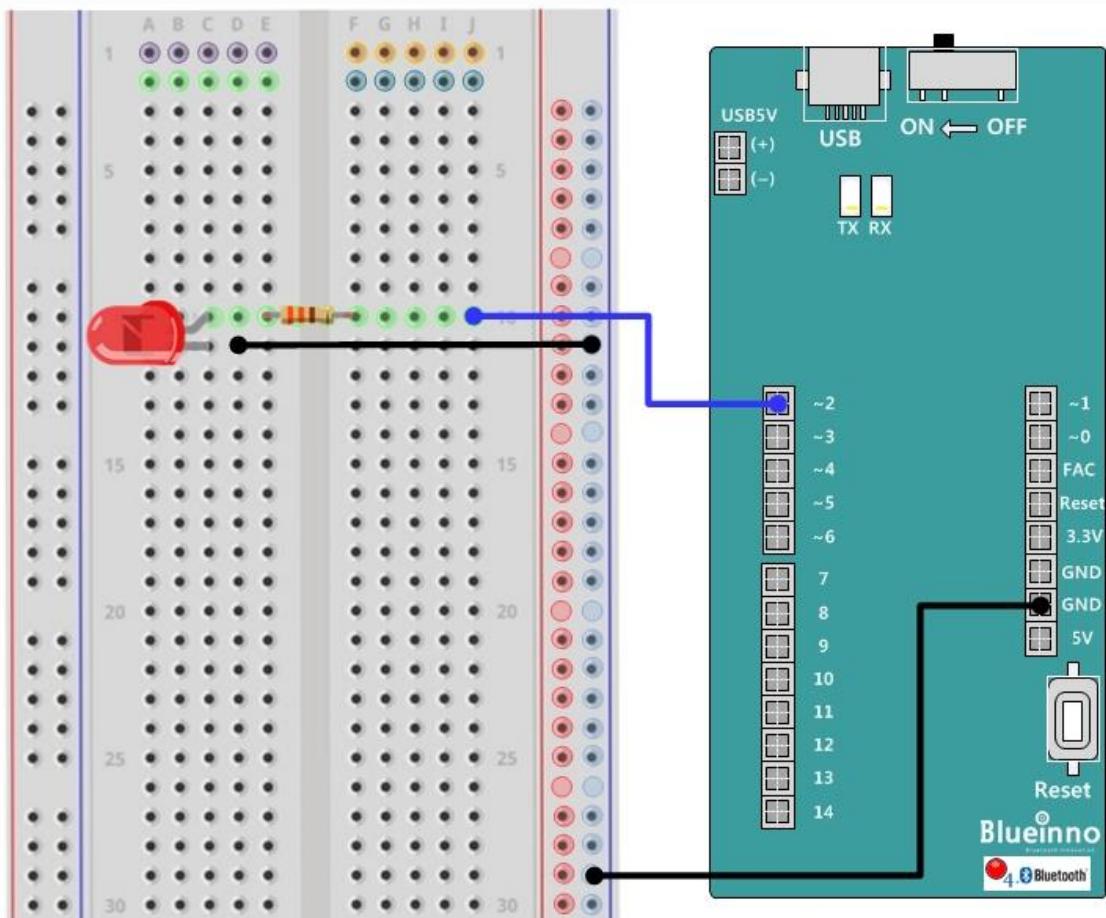


Bluetooth 4.0

프로젝트 10 : 아이비콘(iBeacon) 만들기

▶ 동작 설명

- 아이비콘에 설정된 파라메터를 정기적으로 송신합니다.
 - 송신시마다 LED는 켜지며, 여기에 해당하는 정보가 스마트폰 응용앱의 아이비콘 정보창에 표시됩니다.



UUID
Major
Minor
Rssi



The logo consists of a blue lightning bolt icon above the text "Bluetooth 4.0".

▪ 예제 >BlueinnoSKPT > Circuit - 10

```
/* 아이비콘 만들기
*/
#include <RFduinoBLE.h> // 블루투스 라이브러리를 사용하기 위해 헤더 파일 추가
int led = 2; // LED 연결된 핀을 2번으로 설정
void setup() {
pinMode(led, OUTPUT); // LED를 출력으로 설정
RFduinoBLE.iBeacon = true; // 아이비콘을 사용한다는 것을 의미합니다.
RFduinoBLE.begin(); // 블루투스 (BLE) 스택의 시작함을 의미하며
// 이를 통하여 관련 서비스를 할 수 있습니다.
}

void loop() {
}

void RFduinoBLE_onAdvertisement(bool start)
// Advertisement(비콘)을 시작하면
{
if (start)
digitalWrite(led, HIGH); // LED를 켜고
else
digitalWrite(led, LOW); // 그렇지 않으면, LED를 끕니다.
}
```

4. Bluetooth®

프로젝트 11 : 스마트폰으로 LED 켜기

➤ **목적 :**

- 스마트폰과 보드 간의 무선 통신, 블루투스4.0 통신을 이해합니다.

➤ **회로 설명 :**

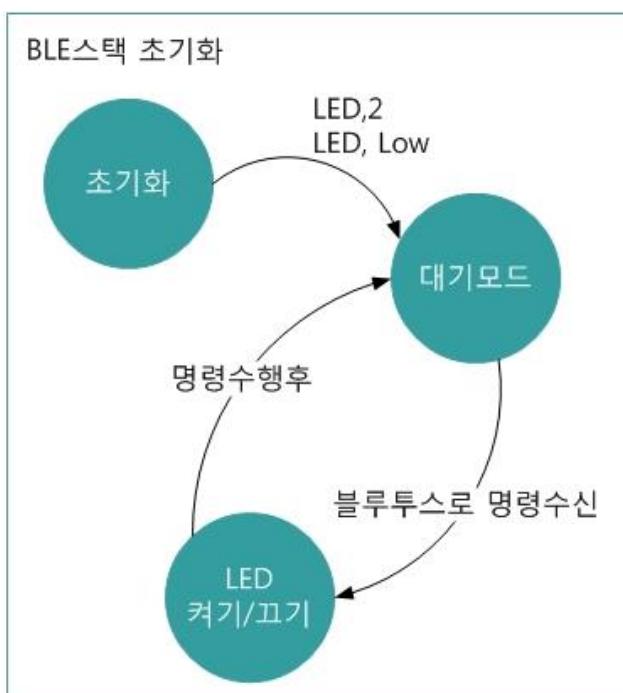
- 스마트폰과 보드 간의 블루투스 연결(페어링)로 서로 통신할 수 있게 합니다.
- 스마트폰 응용앱을 통해서, 미리 정해진 데이터 포맷에 맞추어, LED 켜기 / 끄기의 제어 명령을 보드로 전송하면, 보드는 LED 켜기 / 끄기를 제어합니다.

➤ **동작 :**

- 스마트폰과 보드 간의 블루투스를 연결합니다.
- 그 다음 스마트폰 응용앱을 통해서, 보드의 LED 켜기 / 끄기를 제어합니다.

➤ **소프트웨어 구현 방향**

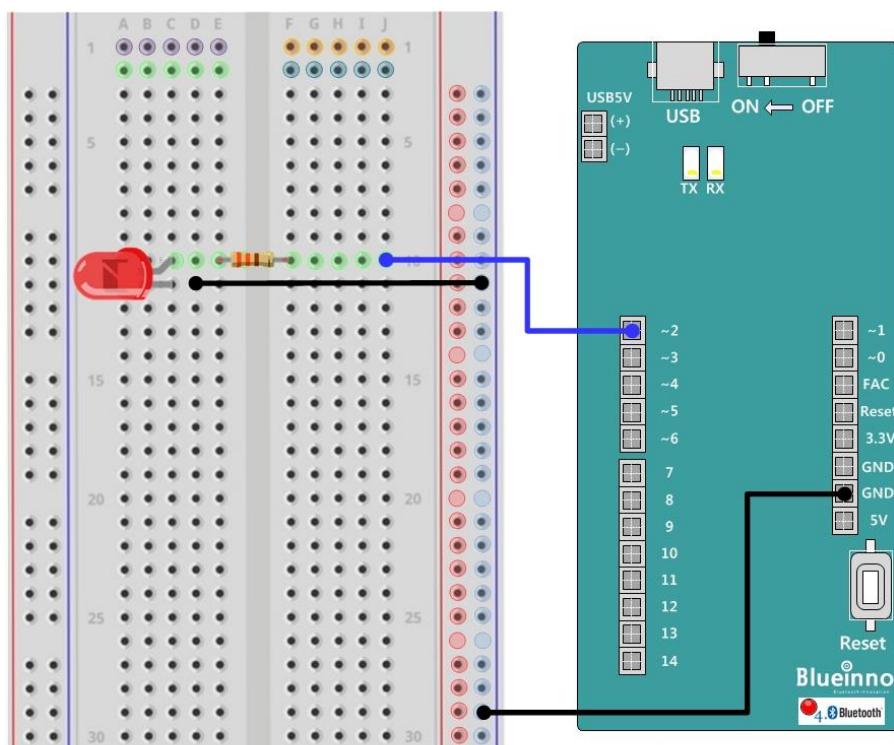
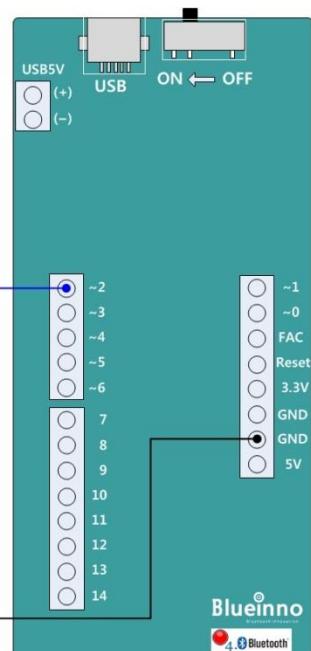
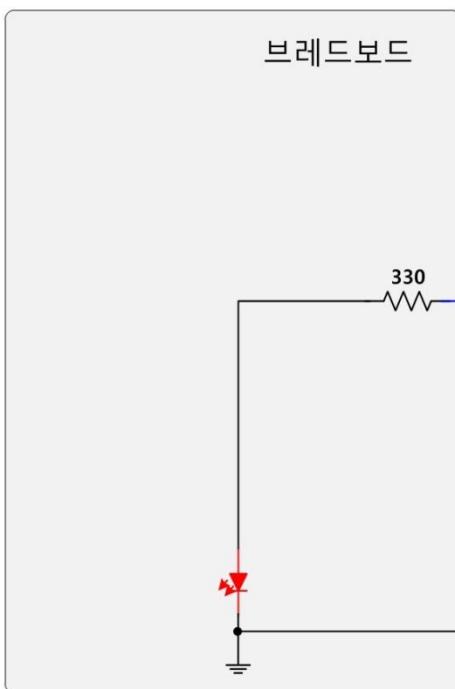
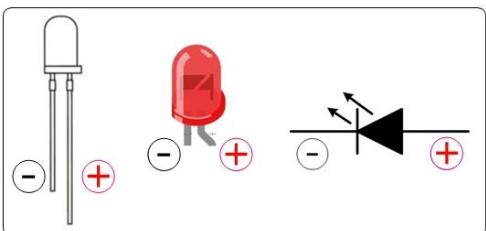
- 아이비콘일 때와 같이 헤더파일 및 초기화 작업을 합니다.
- 폰에서 데이터를 수신하는 함수 RFduinoBLE_onReceive()를 통하여 수신 데이터에 따라 LED 컨트롤이 되도록 구성합니다.



프로젝트 11 : 스마트폰으로 LED 켜기

➤ 회로도

[부품 방향 주의]

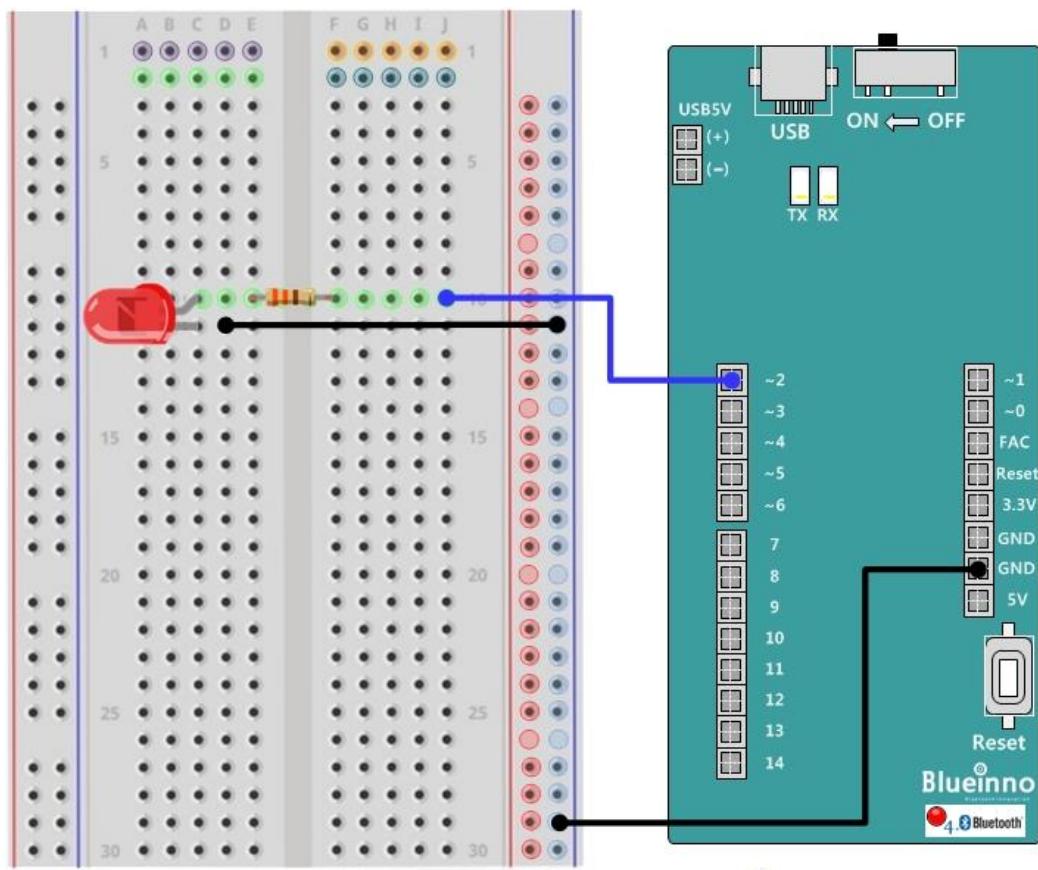


Bluetooth 4.0
LED

프로젝트 11 : 스마트폰으로 LED 켜기

➤ 동작 설명

- 보드와 스마트폰은 서로 연결(페어링)을 하여, 데이터 통신이 가능하도록 합니다.
- 보드는 미리 정의된 블루투스 프로토콜에 맞게 제어 데이터를 수신 대기합니다.
- 스마트폰 응용앱에서 LED 켜기/끄기를 제어합니다.



▪ 예제 >BlueinnoSKPT > Circuit - 11

```
// 스마트폰으로 LED 켜기
#include <RFduinoBLE.h> // 블루투스 라이브러리를 사용하기 위해 헤더 파일 추가
int led = 2; // LED에 연결된 핀을 2번으로 설정

void setup() {
pinMode(led, OUTPUT); // LED를 출력으로 설정
RFduinoBLE.advertisementData = "ledbtn"; // 블루투스 통신 데이터를
// ledbtn 포맷 적용

RFduinoBLE.begin(); // 블루투스 (BLE) 스택의 시작함을 의미하며,
// 이를 통하여 관련 서비스를 할 수 있습니다.
}

void loop() {

}

void RFduinoBLE_onReceive(char *data, int len) // 블루투스 수신 데이터를 처리 함수
// char data 란에 데이터(0~F), len 란에는 길이 표시
{
if (data[0]) // 수신한 데이터가 참(True, char 인 경우 = 1~ F) 이면
digitalWrite(led, HIGH); // LED를 켜기 (High)

else // 수신한 데이터가 거짓(False, 0)이면
digitalWrite(led, LOW); // LED를 끄기 (Low)
}
```

프로젝트 12 : 스마트폰으로 LED 색깔 변경

➤ 목적 :

- 스마트폰과 보드 간의 무선 통신, 블루투스4.0 통신을 이해합니다.

➤ 회로 설명 :

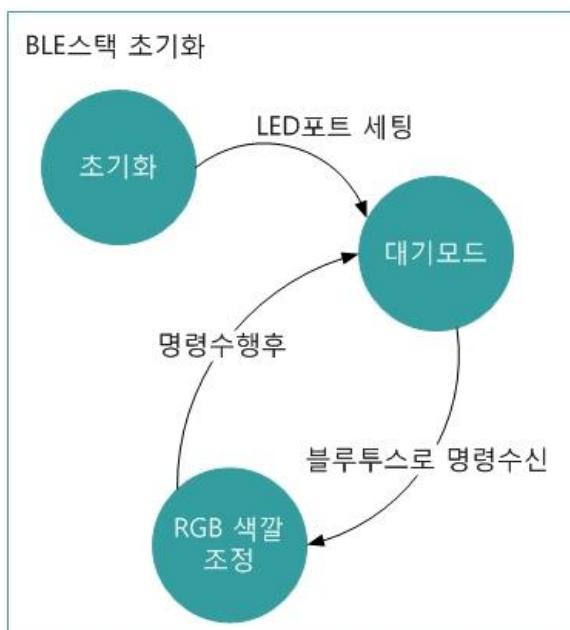
- 스마트폰과 보드 간의 블루투스 연결(페어링)로 서로 통신할 수 있게 합니다.
- 스마트폰 응용앱을 통해서, 미리 정해진 데이터 포맷에 맞추어, 원하는 LED 색깔 정보를 보드로 전송합니다.
- 보드에서는 이에 해당하는 포트 2번, 3번, 4번의 PWM 출력 값들의 조합으로 원하는 LED 색깔을 표시합니다.

➤ 동작 :

- 스마트폰과 보드 간의 블루투스를 페어링 합니다.
- 그 다음 스마트폰 응용앱을 통해서, 보드의 LED 색깔을 제어합니다.

➤ 소프트웨어 구현 방향

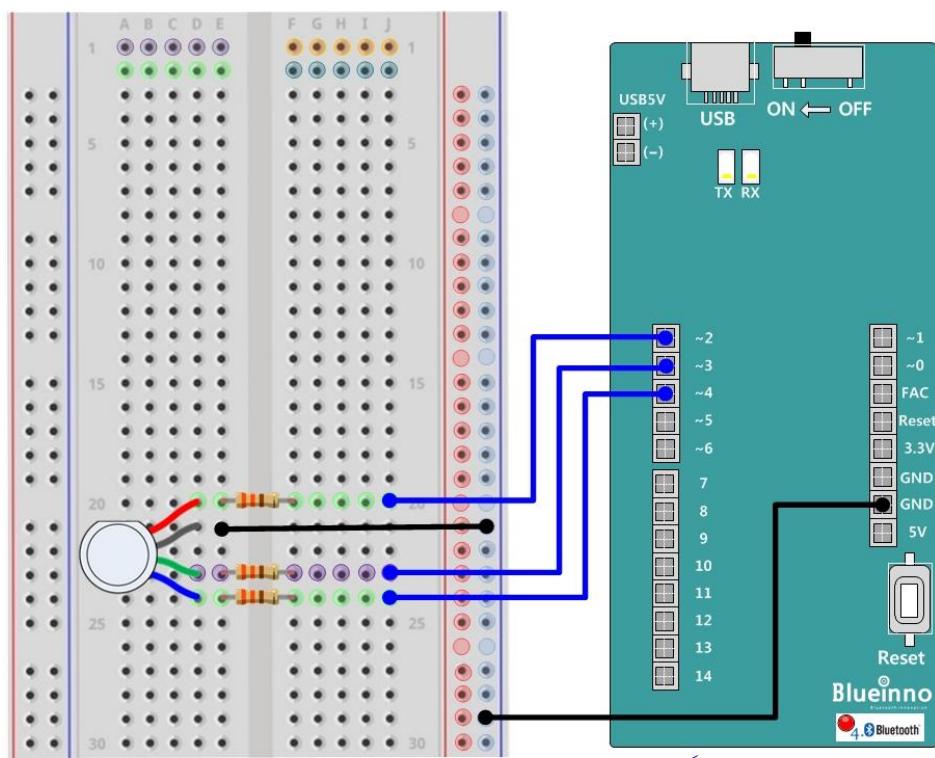
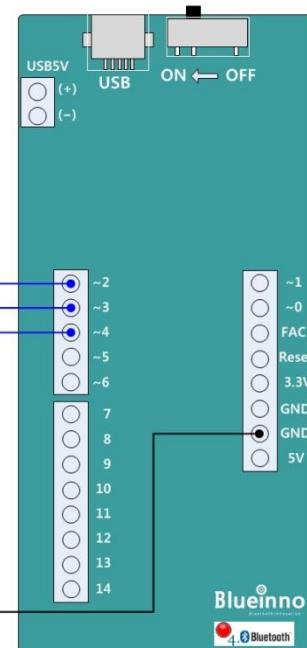
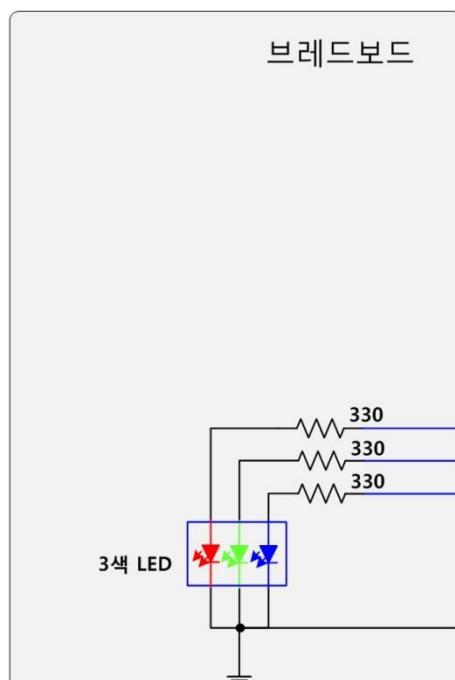
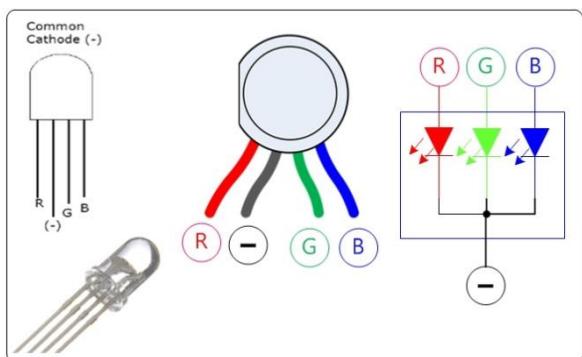
- 스마트폰과 페어링이 될 때와 안 될 때에 대한 상태를 표시합니다.
- 수신하는 함수 RFduinoBLE_onReceive()의 data를 RGB 값에 맞게 분리하여 각각을 변경할 수 있도록 합니다.



프로젝트 12 : 스마트폰으로 LED 색깔 변경

➤ 회로도

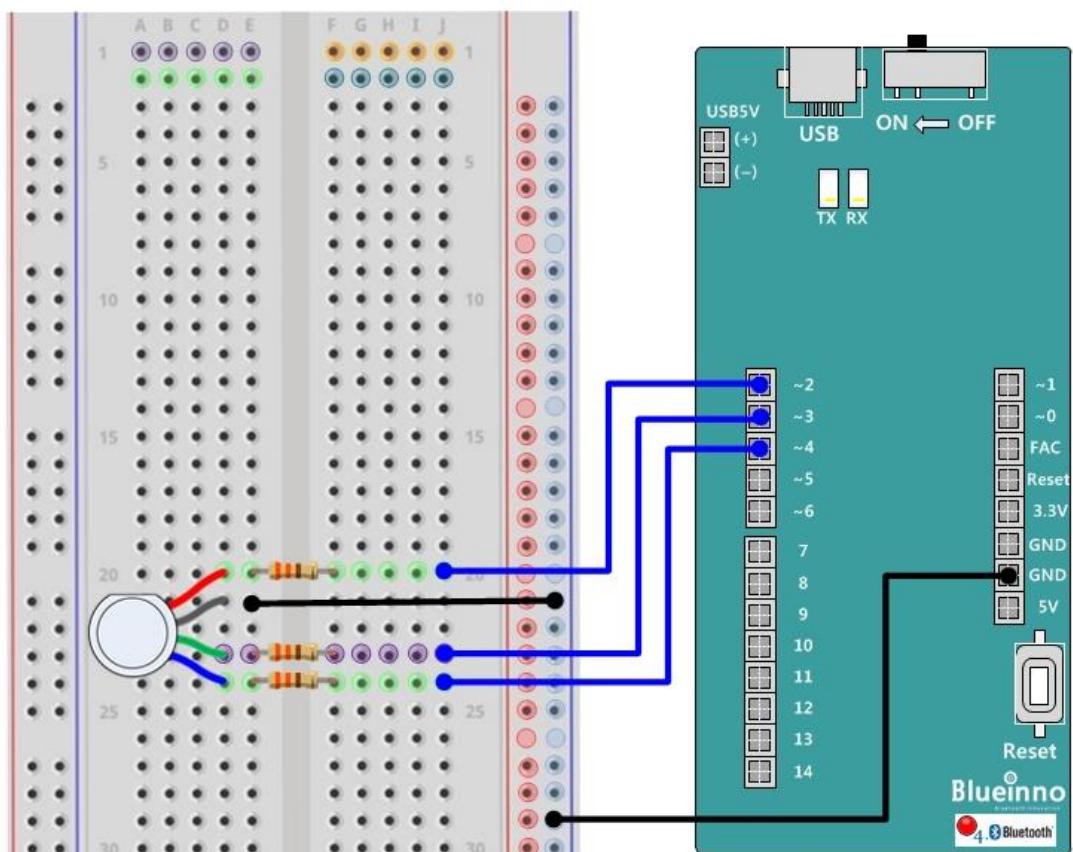
[부품 방향 주의]



프로젝트 12 : 스마트폰으로 LED 색깔 변경

▶ 동작 설명

- 보드와 스마트폰은 서로 페어링을 하여, 데이터 통신이 가능하도록 합니다.
- 보드는 미리 정의된 블루투스 프로토콜에 맞게 제어 데이터를 수신 대기합니다.
- 스마트폰 응용앱에서 RGB 제어를 통해 색깔을 제어합니다.



▪ 예제 >BlueinnoSKPT > Circuit -12

```
/* 스마트폰으로 LED 색깔 변경
*/
#include <RFduinoBLE.h> // 블루투스 라이브러리를 사용하기 위해 헤더 파일 추가
int led1 = 2;           // LED1이 연결된 핀을 2번으로 설정
int led2 = 3;           // LED2이 연결된 핀을 3번으로 설정
int led3 = 4;           // LED3이 연결된 핀을 4번으로 설정

void setup() {
    pinMode(led1, OUTPUT);    // LED 1을 출력으로 설정
    pinMode(led2, OUTPUT);    // LED 2를 출력으로 설정
    pinMode(led3, OUTPUT);    // LED 3을 출력으로 설정
    RFduinoBLE.advertisementData = "rgb"; // 블루투스 통신데이터를 rgb 포맷 적용
    RFduinoBLE.begin();       // 블루투스 (BLE) 스택의 시작함을 의미하며,
                            // 이를 통하여 관련 서비스를 할 수 있습니다.
}

void loop() {

}

void RFduinoBLE_onConnect() { // 블루투스 연결상태일 경우, 흰색
    analogWrite(led1, 255);   // 아날로그 LED1은 255 출력
    analogWrite(led2, 255);   // 아날로그 LED2는 255 출력
    analogWrite(led3, 255);   // 아날로그 LED3은 255 출력
}
```

4. Bluetooth®

예제 >BlueinnoSKPT > Circuit - 12

```

void RFduinoBLE_onDisconnect() { // 블루투스 연결이 끊어진 상태일 경우
    digitalWrite(led1, LOW); // 아날로그 LED1은 0 출력
    digitalWrite(led2, LOW); // 아날로그 LED2은 0 출력
    digitalWrite(led3, LOW); // 아날로그 LED3은 0 출력
}

void RFduinoBLE_onReceive(char *data, int len){ // 블루투스 수신 데이터를 처리 함수
    // char data 란에 데이터(0~F), len 란에는 길이 표시

    if (len >= 3) // 수신된 데이터의 길이가 3보다 같거나 크다면
    {
        uint8_t r = data[0]; // 8 bit R은 데이터의 첫 번째를 읽음
        uint8_t g = data[1]; // 8 bit G는 데이터의 두 번째를 읽음
        uint8_t b = data[2]; // 8 bit B는 데이터의 세 번째를 읽음

        analogWrite(led1, r); // LED1에 R 값을 출력
        analogWrite(led2, g); // LED2에 G 값을 출력
        analogWrite(led3, b); // LED3에 B 값을 출력
    }
}

```

➤ 프로그램 설명

- RFduinoBLE_onConnect() : 폰과 페이링이 되어졌을 때 불리어지는 함수입니다.
- RFduinoBLE_onDisconnect() : 폰과 페이링이 끊어졌을 때 불리어지는 함수입니다.

프로젝트 13 : 스마트폰으로 서보 모터 각도 제어

➤ 목적 :

- 스마트폰과 블루이노 간의 블루투스4.0 통신을 통해서, 서보 모터의 각도 제어를 이해합니다.

➤ 회로 설명 :

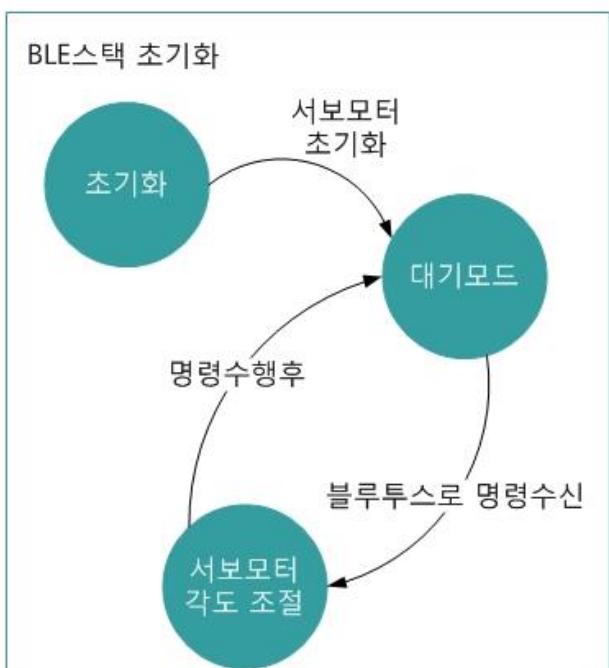
- 스마트폰과 보드 간의 블루투스 페어링으로 서로 통신할 수 있게 합니다.
- 스마트폰 응용앱을 통해서, 미리 정해진 데이터 포맷에 맞추어 서보 모터의 방향 정보를 보드로 전송합니다.
- 보드에서는 포트 2번에서 이에 해당하는 각도의 디지털 신호를 출력하여, 서보 모터의 방향을 제어합니다.

➤ 동작 :

- 스마트폰과 블루이노 보드 간의 블루투스를 페어링 합니다.
- 그 다음 스마트폰 응용앱을 통해서, 서보 모터의 각도를 제어합니다.

➤ 소프트웨어 구현 방향

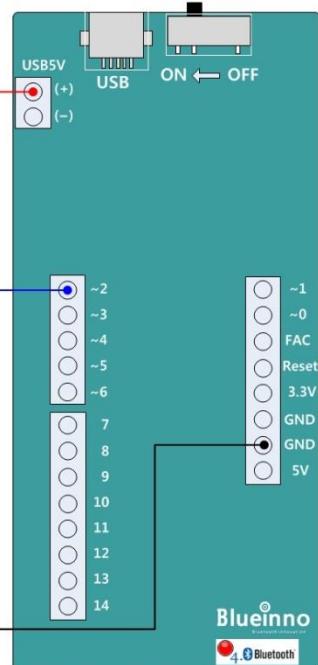
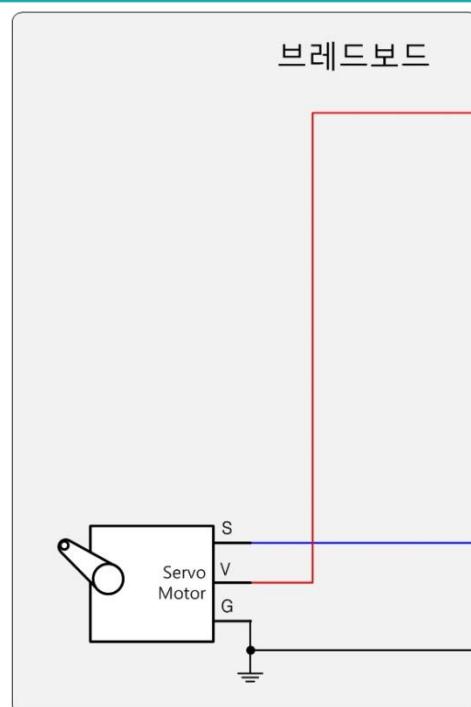
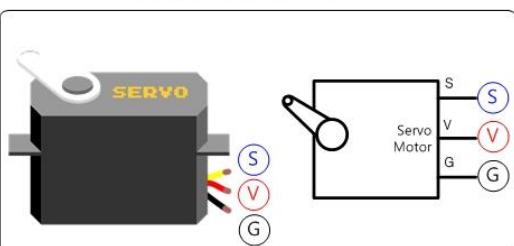
- 이전 프로젝트에서 구현해본 서보 모터 세팅을 사용합니다.
- 수신하는 함수 RFduinoBLE_onReceive()의 data를 각 서보 모터에 맞게 각도로 변경하여 서보 모터를 제어합니다.



프로젝트 13 : 스마트폰으로 서보 모터 각도 제어

➤ 회로도

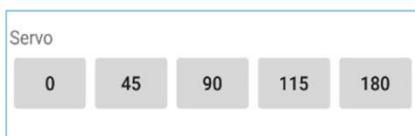
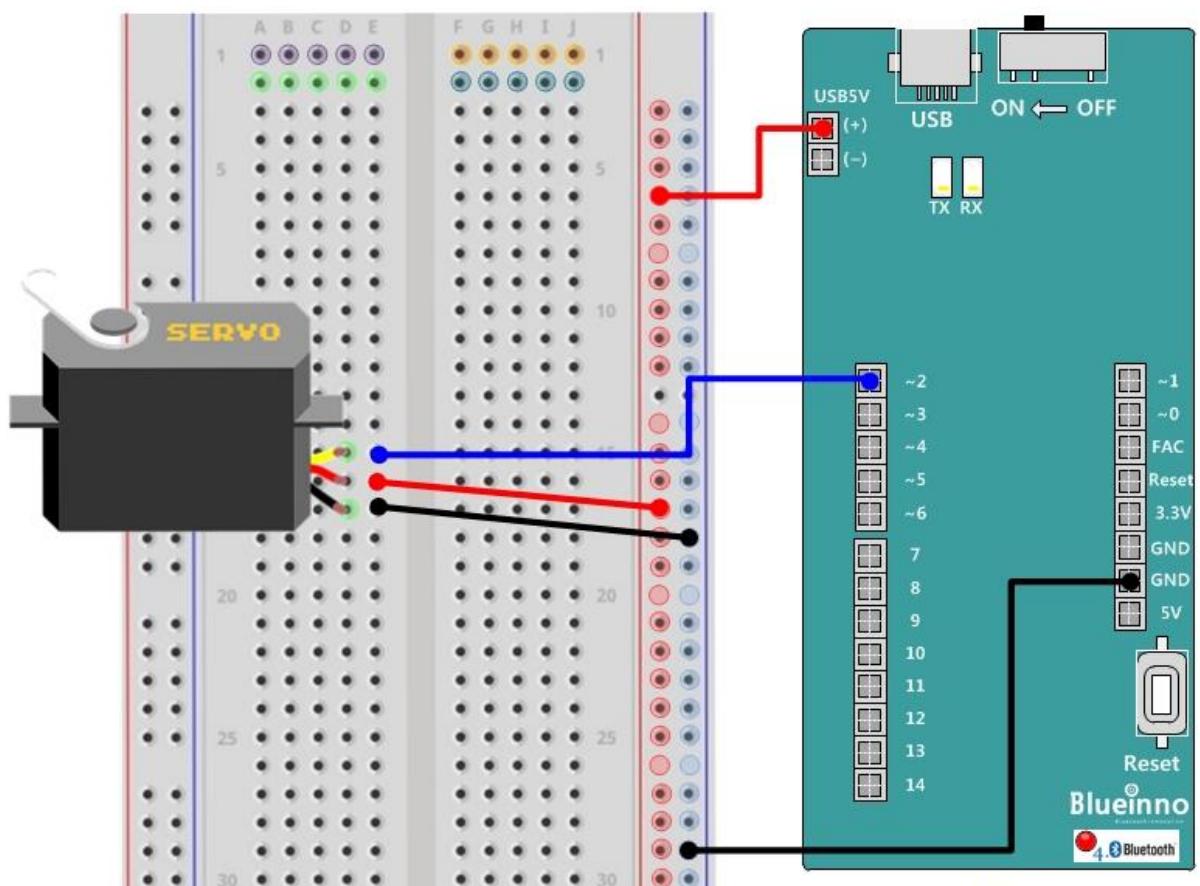
[부품 방향 주의]



프로젝트 13 : 스마트폰으로 서보 모터 각도 제어

▶ 동작 설명

- 보드와 스마트폰은 서로 페어링을 하여, 데이터 통신이 가능하도록 합니다.
- 보드는 미리 정의된 블루투스 프로토콜에 맞게 제어 데이터를 수신 대기합니다.
- 스마트폰 응용앱에서 서보 모터의 각도를 제어합니다.



예제 >BlueinnoSKPT > Circuit - 13

```

/*스마트폰으로 서보 모터 각도 제어

*/
#include <Servo.h> // 서버 모터 라이브러리를 사용하기 위해 헤드 파일을 추가
#include <RFduinoBLE.h> // 블루투스 라이브러리를 사용하기 위해 헤드파일 추가
Servo s1; // 서보 모터를 s1에 설정

void setup() {
    s1.attach(2); // 서보 모터 s1을 2번 핀에 설정
    RFduinoBLE.advertisementInterval = 500; // 블루투스 전송 간격을 500ms로 유지
    RFduinoBLE.advertisementData = "servo"; // 블루투스 데이터 전송을
                                                // servo 포맷을 적용
    RFduinoBLE.begin(); // 블루투스 (BLE) 스택의 시작함을 의미하며,
} // 이를 통하여 관련 서비스를 할 수 있습니다.

void loop() {

}

void RFduinoBLE_onReceive(char *data, int len){ // 블루투스 수신 데이터를 처리 함수
    // char data 란에 데이터(0~F), len 란에는 길이 표시
    int servo = data[0]; // 첫 번째 데이터는 서보 모터 번호를 읽음
    int degree = data[1]; // 두 번째 데이터는 서보 모터의 각도를 읽음

    if (servo==0x01); // 서보 모터 번호가 1이면,
        s1.write(degree); // 서보 모터 1번의 각도를 출력
}

```

블루투스 4.0 = BLE (Bluetooth Low Energy)

- **블루투스 4.0의 탄생 배경**
 - 최적의 디바이스를 만들기 위해, 특정 프로파일만을 지원하도록, 블루투스 무선 기술의 전체기능을 구현할 필요가 없이 적합하게 개발된 것입니다.
 - 블루투스 프로파일이란 블루투스 애플리케이션을 구현할 때, 특정 애플리케이션마다 사용해야 할 프로토콜의 종류와 그 구조 및 사용방법을 규정한 것입니다.

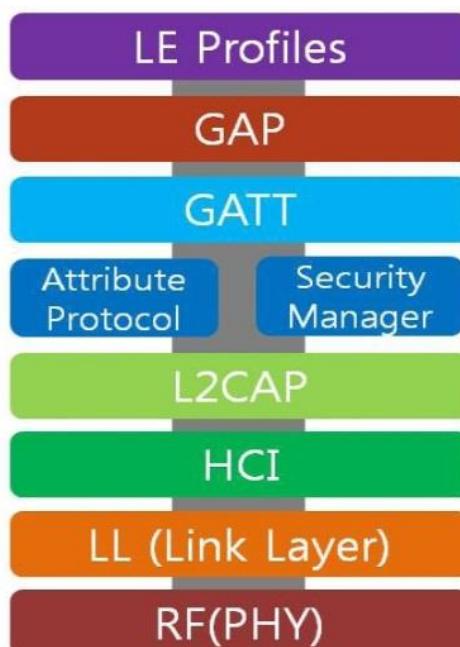
- **블루투스 4.0의 특징**
 - 낮은 가격, 낮은 대기 시간
 - 낮은 Duty Cycle, 낮은 데이터 전송률
 - 낮은 전력소모

- **블루투스 4.0과 기존 블루투스의 비교**



기술사양	BR/EDR	LE
RF Channels 수	79	40
데이터 전송 속도	1~3 Mbps	1 Mbps
토플로지	스캐터넷	스타버스
음성지원	지원	지원하지 않음
데이터 전송시간	100 ms	< 6 ms
파워소모	< 30 mA	<15 mA
주 사용 애플리케이션	휴대폰, 헤드셋, PC, 핸드헬드 디바이스 등	휴대폰, 시계, 스포츠, 헬스케어, 센서, 기기제어 등

- **블루투스 프로파일 구조**



블루투스 4.0 응용 사례 – 사물인터넷, 웨어러블 기기

- 스마트폰 ~ 무선 RC , 장난감



- 스마트폰 ~ 웨어러블 기기



- 스마트폰 ~ 사물인터넷의 소형 디바이스 제품

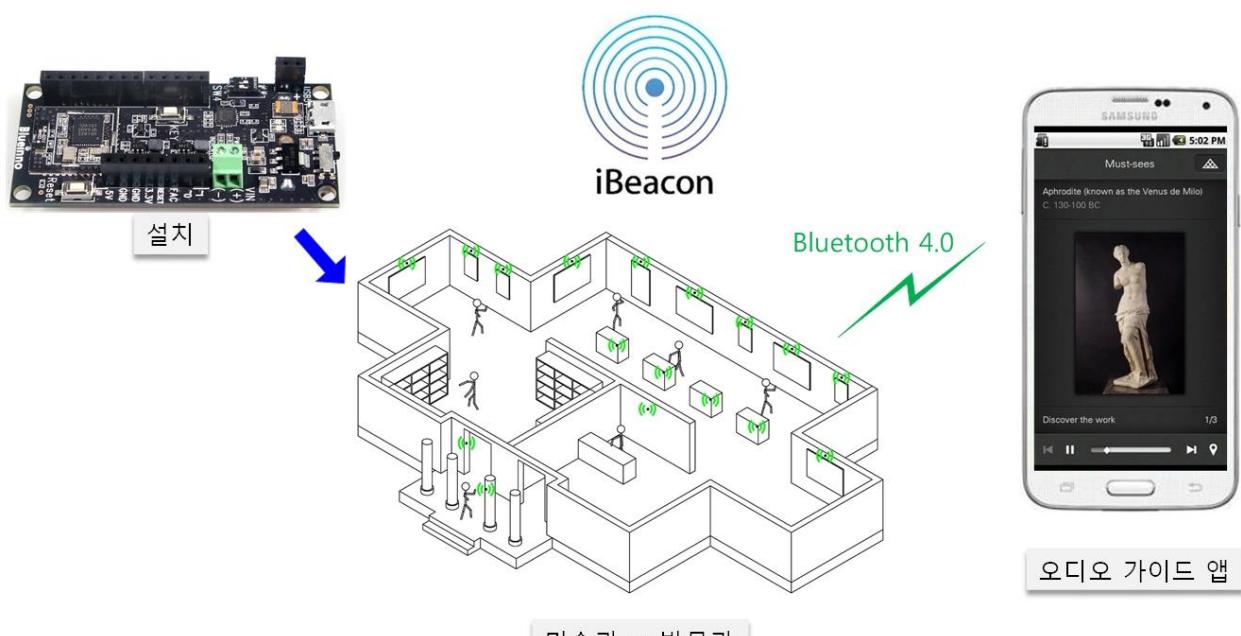


블루투스 4.0 응용 사례 - 아이비콘

- 다양한 응용 서비스가 가능한데, 대표적인 서비스인 아이비콘(iBeacon)이 있습니다.
- 아이비콘은 실내 측위시스템을 위한 애플의 등록 상표로서, 디바이스를 가진 자신의 위치를 알릴 수 있는 새로운 형태의 저전력, 저비용의 전파 발신장치입니다.
- 아이비콘은 식별자(UUID등)를 정보를 발신하며, 송신주기와 송신 강도는 배터리 소모량에 영향을 크게 미치므로 미리 설정된 설정값을 가지고 있으며, 이 중 major와 minor를 포함한 일부 설정값을 개발자가 변경할 수 있습니다.

▶ 오디오 가이드 서비스

- 전시장이나 박물관에 아이비콘 디바이스를 전시물마다 설치하고, 그 고유 ID를 데이터베이스합니다. 관람객이 전시 작품 근처에 다가가면 편리하게 작품에 대한 자세한 설명을 스마트폰을 통해서 서비스를 받을 수 있습니다.



블루투스 4.0 응용 사례 - 아이비콘

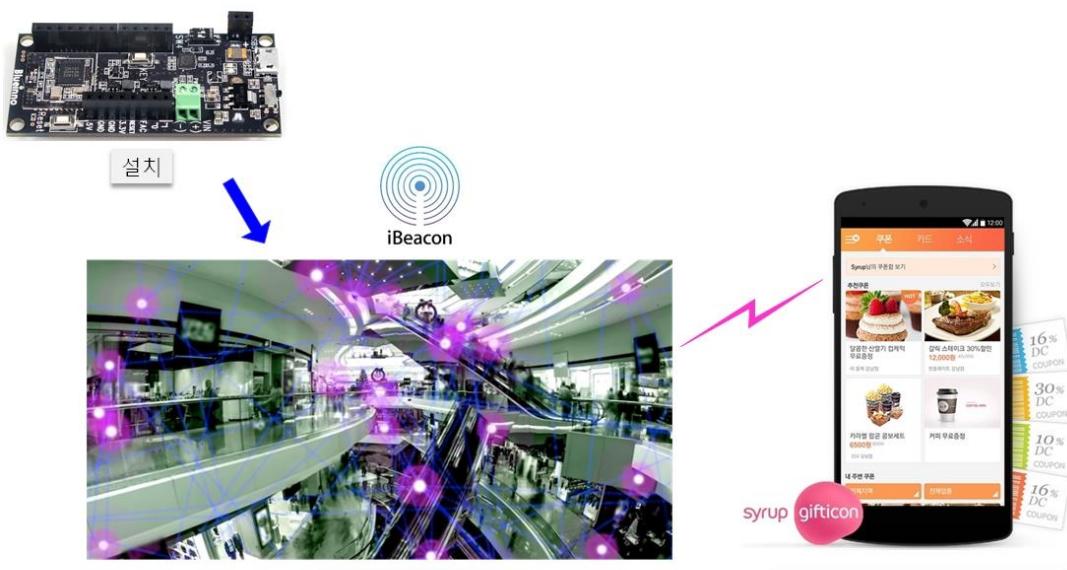
실내 내비게이션 서비스

- 복잡한 건물 (전시장, 빌딩, 병원) 내에 아이비콘을 활용한 실내 내비게이션 서비스를 구현할 수 있습니다.



전자쿠폰 서비스

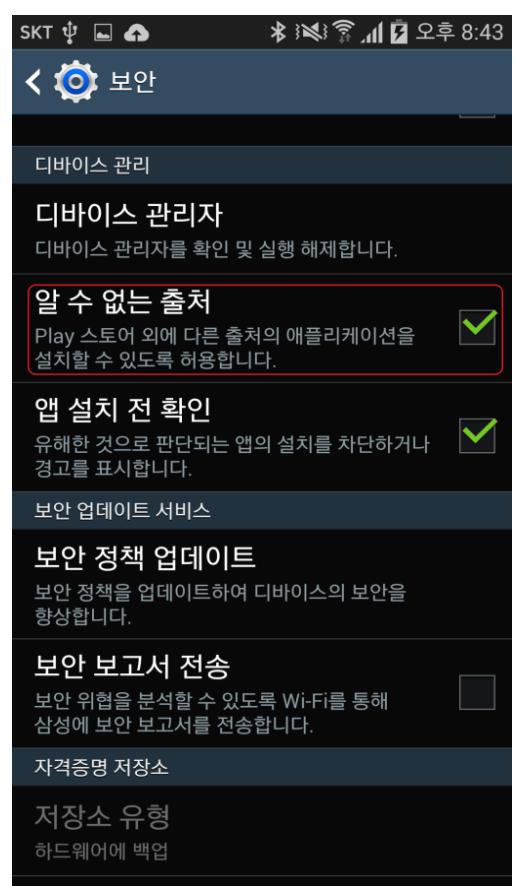
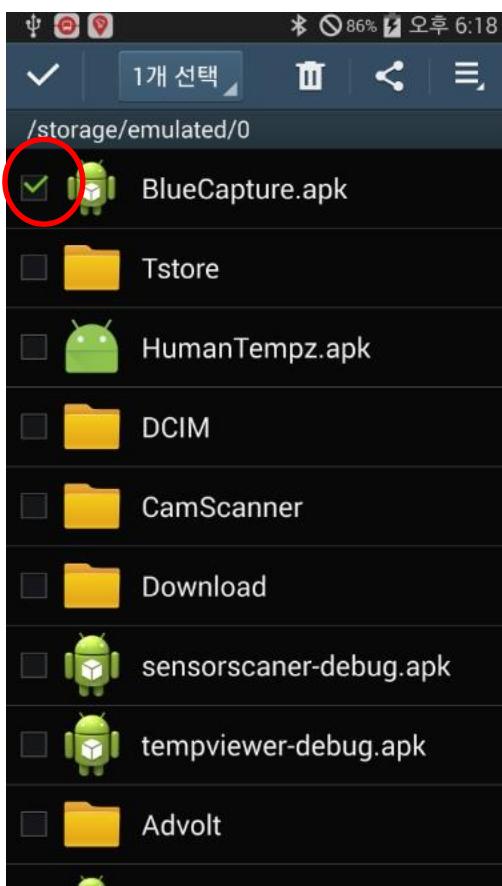
- 매장이나 쇼핑몰의 고객들에게 전자쿠폰이나 제품 할인정보를 제공하는 서비스를 구현할 수 있습니다.



스마트 폰 응용앱 사용하기(안드로이드 OS용)

▪ 통합 응용앱 설치하기

- Bluetooth 4.0(BLE)이 지원되는 스마트폰이어야 합니다.
(삼성 스마트폰의 경우, Galaxy S3 이상)
- 네이버카페(<http://cafe.naver.com/arduinoplusble>) 자료실에서 BlueCapture.apk 파일을 다운받습니다.
- 다운받은 BlueCapture.apk 파일을 안드로이드폰의 내장메모리
(또는 외장메모리)의 적당한 위치에 복사합니다.

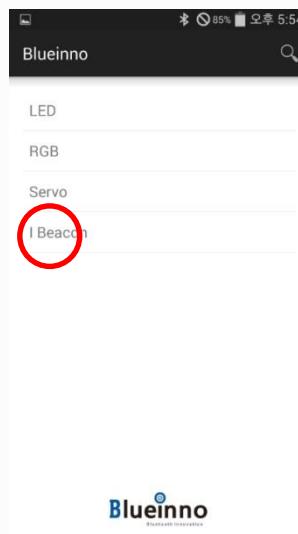
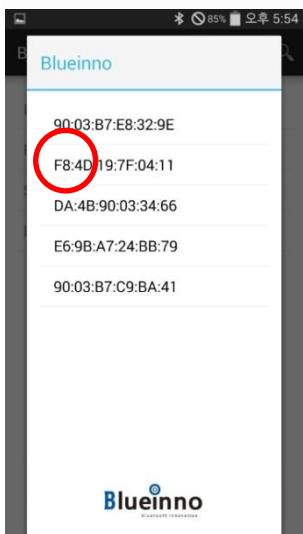
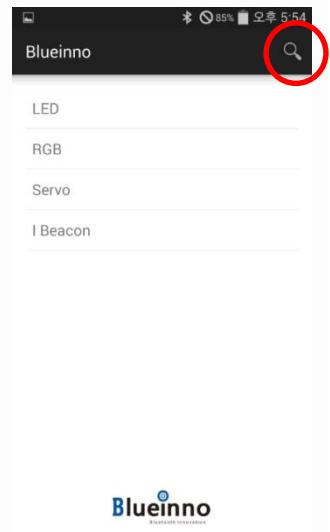
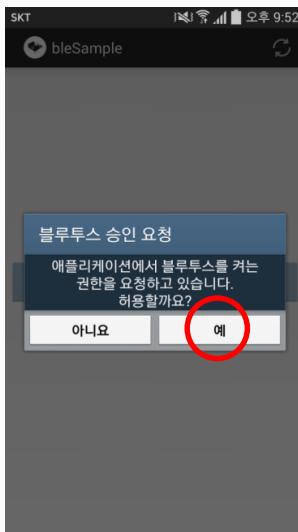


- 안드로이드폰에서 파일관리자 앱을 실행하고 복사한 BlueCapture.apk 파일을 선택해서 설치합니다.
(이 때 스마트폰 설정→보안→알 수 없는 출처가 체크되어야 함)

스마트폰 응용앱 사용하기 (안드로이드 OS용)

■ 앱 사용하기

- 스마트폰의 설치된 앱을 실행하면, Find(🔍)로 기기 검색 (블루투스가 켜져 있어야 함)
- 검색된 기기를 선택 => 검색된 서비스를 선택 => 보드 정보 및 제어하기

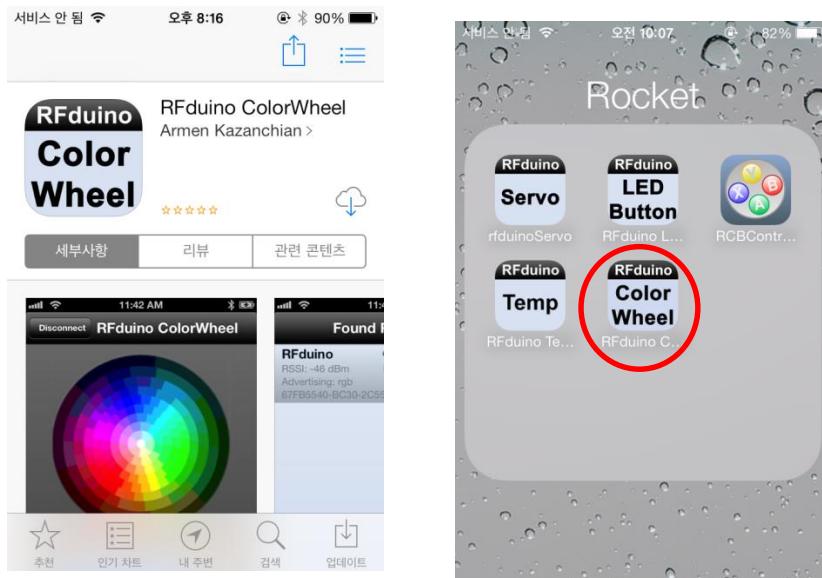


- I Beacon : UUID, Major, Minor, RSSI(신호세기)를 읽을 수 있습니다.
- LED : LED 켜기 / 끄기 제어를 할 수 있습니다.
- RGB : 3색 LED의 색깔을 변경할 수 있습니다.
- Servo : 서보 모터의 각도를 제어할 수 있습니다.
- ❖ I Beacon은 검색없이 실행되지만, LED, RGB, Servo 어플은 기기 검색을 통한 해당 ID 선택으로 연결하여야만 제어가 가능합니다.

스마트폰 응용앱 사용하기(iOS용 스마트폰)

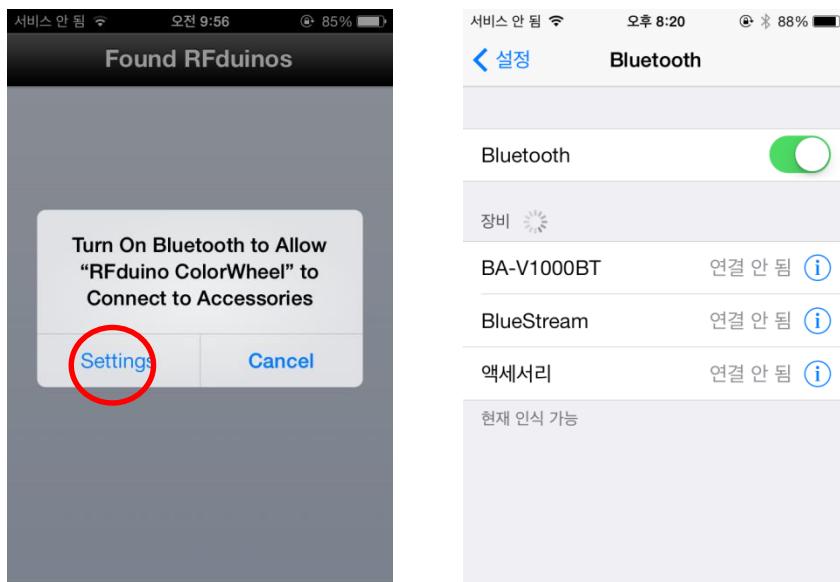
■ 앱 설치하기 (예제 ; RGB ColorWheel 사용하기)

- Bluetooth 4.0(BLE)이 지원되는 아이폰은 버전 4S 이상, iOS 8.2 이상이어야 합니다.
- 아래의 그림과 같이 앱스토어에 연결되어 다운로드를 선택하면 설치 완료 후 바탕화면에 바로 가기가 생성됩니다.



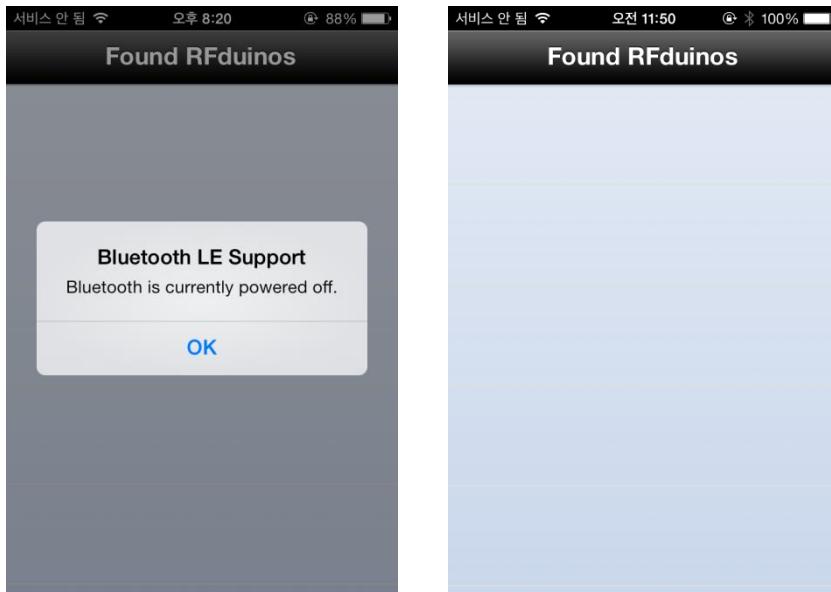
■ 앱 사용하기

- 아이폰 바탕화면에서 설치된 ColorWheel 앱을 선택하면 아래 그림과 같이 Bluetooth를 On하라는 팝업 메시지가 나타나고 Settings를 선택하여 Bluetooth를 On 시킵니다.

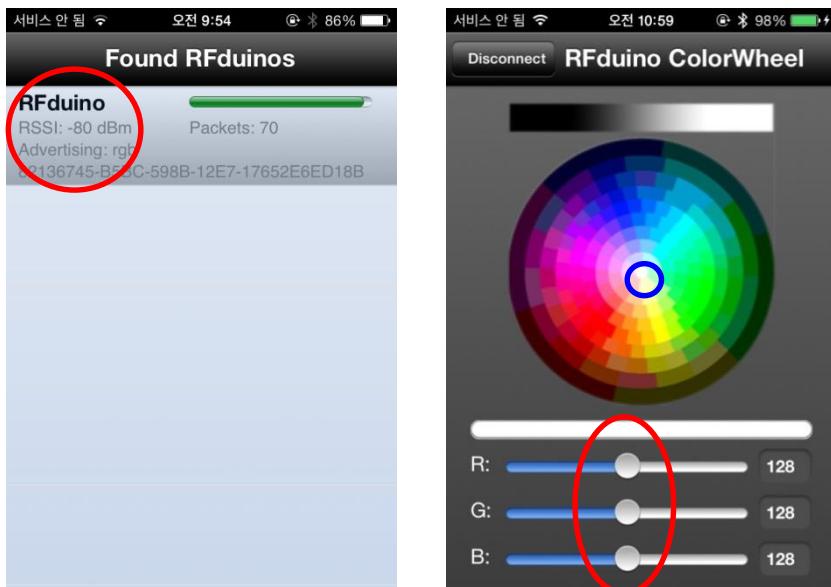


스마트폰 응용앱 사용하기(iOS용 스마트폰)

- 만약 블루이노 보드의 전원이 꺼져있으면, Bluetooth On 상태에서 ColorWheel 앱을 실행해도 아래 그림과 같이 팝업 메시지가 나타나고 화면상에 연결된 기기가 아무것도 안 보입니다.



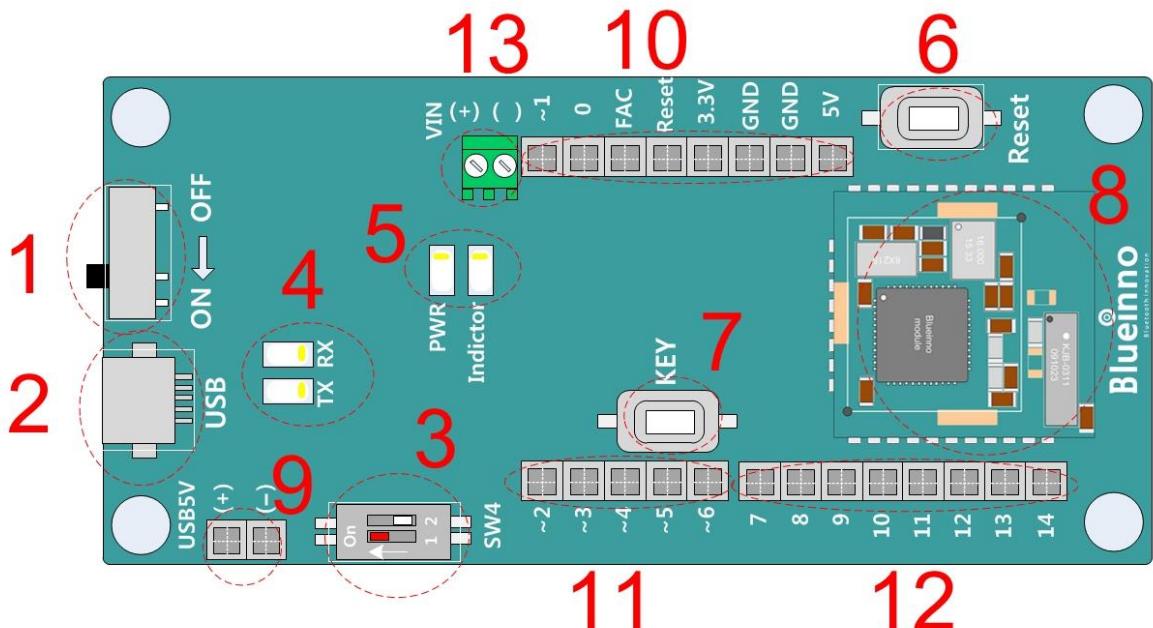
- 블루이노 보드의 전원이 켜져 있고 아이폰의 Bluetooth On 상태에서 ColorWheel 앱을 실행하면, 그림과 같이 연결(페어링)된 기기가 화면에 나타나고 선택하여 실행합니다.



ColorWheel 앱에서 임의의 색깔을 터치하거나 또는 R, G, B 바를 각각 드래그하면, 3색 LED가 해당 색깔로 변경됩니다.

블루이노2 기본형 보드 설명

➤ 주요부위 설명

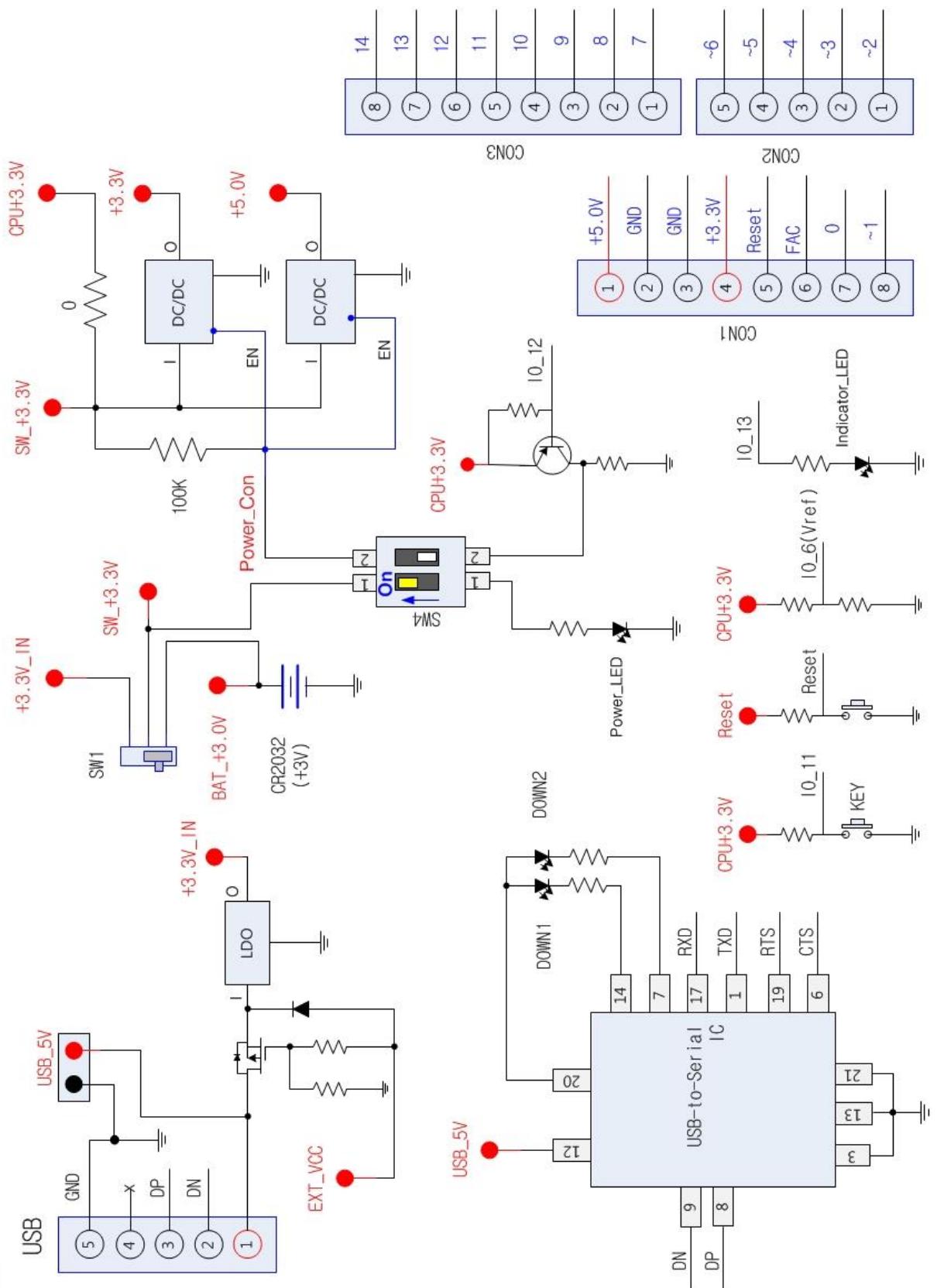


- 1번 : 전원 On/Off 스위치
- 2번 : Micro USB 컨넥터 단자 (USB 통신 + 전원공급)
- 3번 : 선택 스위치
 - 1) 1번 전원상태 표시 (LED On) 선택 , default = on
 - 2) 2번 Power Save 동작 여부 선택 , default = off
- 4번 : 시리얼 통신 표시 LED
- 5번 : PWR : 전원 LED, Indicator : LED 제어 13번 핀
- 6번 : 시스템 리셋 버튼
- 7번 : 내부 입력용 버튼
- 8번 : 블루이노 모듈 (Bluetooth 4.0 Soc + Arduino Sketch)
- 9번 : USB 전원 (+5.0V, 최대 500mA)
- 10번 : Con 1
- 11번 : Con 2
- 12번 : Con 3
- 13번 : VIN, 외부 인가 전압 (DC 4.5 ~ 15V)

블루이노2 기본형 보드 설명

Con.	Name	설명
Con 1	+5.0V	전원 DC 5V (최대 80mA)
	GND	Ground
	GND	Ground
	+3.3V	전원 DC 3.3V (최대 150mA)
	RESET	System reset.
	FAC	공장 초기화
	0	Digital In / Out
	~1	Analog In / PWM Out, Digital In / Out
Con 2	~2	Analog In / PWM Out, Digital In / Out
	~3	Analog In / PWM Out, Digital In / Out
	~4	Analog In / PWM Out, Digital In / Out
	~5	Analog In / PWM Out, Digital In / Out
	~6	Analog In / PWM Out, Digital In / Out
Con 3	7	Digital In / Out
	8	Digital In / Out
	9	Digital In / Out
	10	Digital In / Out
	11	Digital In / Out
	12	Digital In / Out
	13	Digital In / Out
	14	Digital In / Out
UART 통신	Serial.begin(baud) ; USB 포트 출력 , PC와 통신시 사용 Serial.begin(baud, RX pin, TX pin) ; GPIO 핀에 사용자 할당	
I2C 통신	Wire.begin() ; 기본설정 SCL = 6번, SDA = 5번 Wire.beginOnPins(SCL pin, SDA pin) ; GPIO 핀에 사용자 할당	
SPI 통신	기본설정 MISO = 3번, SCK = 4번 , MOSI = 5번 , SS/CS = 6번 사용자 할당시는 variant.h 파일을 수정해야 함	
SoftwareSerial 함수	#include <SoftwareSerial.h> // 헤더 파일 추가 SoftwareSerial (rx_pin, tx_pin) // rx_pin 1 ~ 7 까지, Interrupt pin(1~7) 사용 begin(9600) // 최대 9,600bps까지	

블루이노2 기본형 보드 회로도



블루이노2 기본형 제품 사양

Item	Contents
RF Spec	Bluetooth 4.0 + 2.4Ghz RF
	TX Power = +4dbm (10.5mA)
	RX Sensitivity = -93dBm (13mA)
CPU	16MHz ARM Cortex-M0
Flash, RAM	256KB, 16KB
I/O (15ea)	- Digital I/O
	- Analog PWM Out
	- Analog ADC In (10bit) – 6ea
UART	Serial 115,200bps
	I2C , SPI
SIZE	30 x 65 x 14 mm
Input Power	micro USB 5V or Battery 3V
Coin Battery	CR2032(3V, 240mAh)

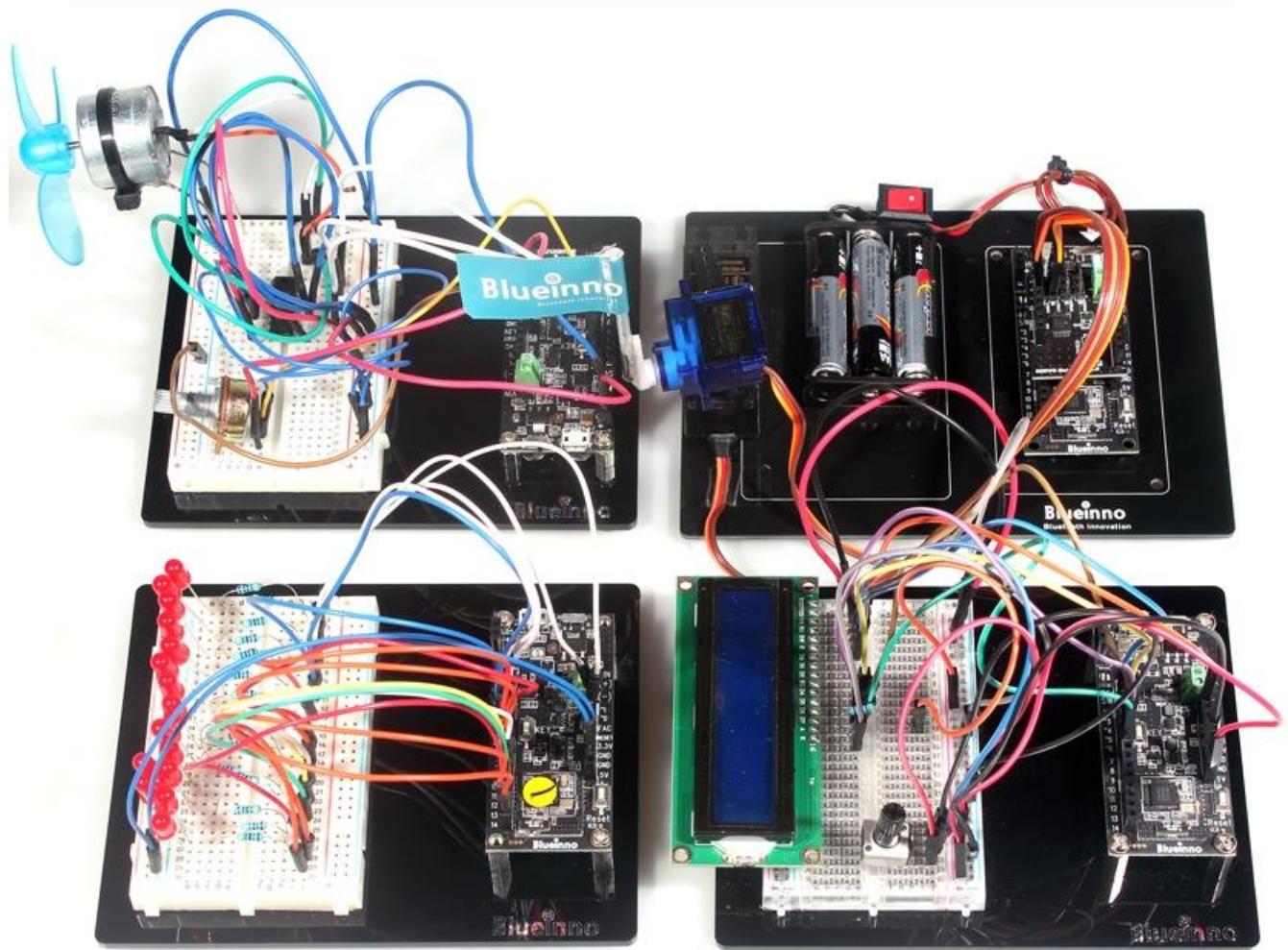
NAVER 카페

블루이노

검색

<http://cafe.naver.com/arduinoplusble>

스타트 키트 수업 예제



블루이노 고객센터



아두이노와 같은 오픈소스 하드웨어 제품입니다.
특히, 사물인터넷, 웨어러블 디바이스, 아이비콘 등등 제작이
쉽습니다.

153-715 서울특별시 금천구 벚꽃로 298, 514호
(가산동, 대륭 포스트타워 6차)
514, Daeryung Post Tower 6-cha, Gasan-dong, Geumcheon-gu,
Seoul, Korea [Zip:153-715]



Tel : 070 - 4288 - 8187

Fax : 02 - 2083 - 8188

Cell : 010 – 9139 – 5839

이메일 dwlee@blueinno.co.kr
홈페이지 www.blueinno.co.kr

[네이버 카페]

<http://cafe.naver.com/arduinoplusble>

자세한 정보는 네이버카페(블루이노)에서 참조하세요!

NAVER 카페

블루이노

검색

한세대학교 류대현 교수, 블루이노 공동 지음

Blueinno start kit