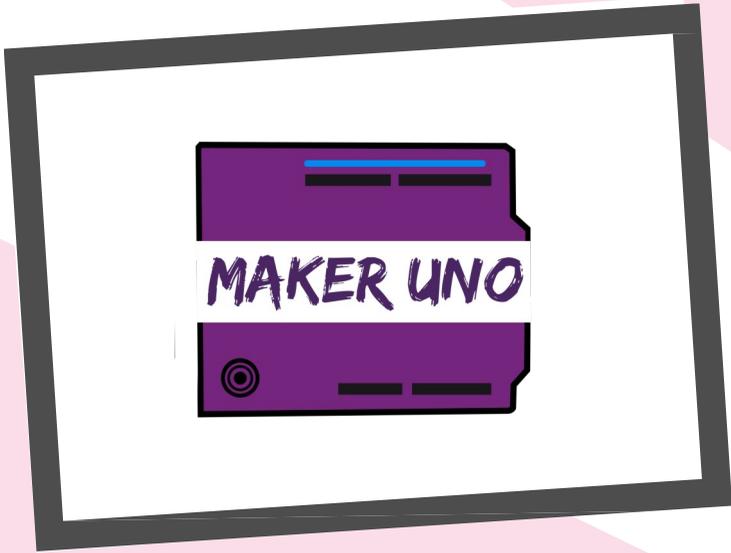


# Teacher Guidebook

Maker-UNO /

Arduino



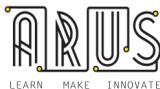
**기술 설계 커리큘럼 규격(DSKP)에  
따라 구체적으로 작성됨**

# 서문

본 모듈은 말레이시아 교육부가 발행한 양식 2의 기술 디자인 과목(RBT)에 대한 커리큘럼 규격(DSKP)을 바탕으로 구체적으로 작성되었다. 본 모듈에서는 강의의 전자적 요소를 가르칠 수 있는 올바른 지식과 기술로 교사를 강화하고 준비시키는 것을 목표로 한다. 이 모듈은 프로그래밍과 마이크로 컨트롤러 사용에 대한 사전 지식이 없는 교사를 위해 특별히 작성되었다. 이 모듈은 제조업체 UNO 마이크로 컨트롤러를 위해 특별히 설계되었지만 예를 들어 CT-UNO와 같은 아두이노 UNO 폼 팩터와 호환되는 마이크로 컨트롤러에도 적용된다. 이 모듈에는 교사 모듈과 함께 사용할 수 있는 학생 모듈도 함께 제공된다. 이 모듈은 사이트론 테크놀로지스(Cytron Technologies)와 아루스 아카데미(Arus Academy)의 공동 협력인 교사들이 개발하고 작성했다. 아루스 아카데미(Arus Academy)는 말레이시아 부킷 메르타렘 주 알마에 본사를 둔 사회적 기업이다. 네 명의 중등학교 교사들이 특권 이하의 아이들에게 STEM 교육에 높은 품질과 관련성을 제공한다는 사명을 가지고 공동 설립되었다. 아두이노는 오픈 소스의 정신과 원리로 작동한다. 이 모듈은 같은 원리를 염두에 두고 개발되었다. 사용자는 모듈에 포함된 모든 정보를 공유, 수정 및 재배포할 것을 권장한다. 우리가 요구하는 것은 출처 자료를 인정하고 적절한 크레딧을 주는 것이다.

본 모듈은 Creative Commons Accreditation -Share Alike 4.0 International에 따라 보호됨 (CC BY-SA 4.0).

본 문서의 한글 번역본은 ㈜인투피온의 소중한 자산이며, 무단으로 개제 및 배포 시 법률상 불이익을 받을 수 있습니다.





본 문서는 Curtentation-ShareAlike 4.0 International에 따라 라이선스가 부여된 Creative Commons mons.

## 허가 및 조건

- 공유성 - 매체 또는 형식으로 자료를 복사 및 재 배포 가능
- 적응성 - 어떤 목적 으로든 상업적으로도 재료를 리믹스, 변형 및 제작할 수 있습니다.

## 다음 조건하에서

- 귀속 - 적절한 크레딧을 부여하고, 라이선스에 대한 링크를 제공하고, 변경 여부를 표시해야 한다. 어떤 합리적인 방법으로든 그렇게 할 수 있지만, 허가자가 사용을 승인한다는 것을 뜻하지는 않는다.
- 똑같이 공유 - 자료를 리믹스, 변환 또는 빌드하는 경우, 원본과 동일한 라이선스로 기여금을 분배해야 한다.
- 추가 제한 없음 — 다른 사람이 라이선스 허가가 허락하는 것을 법적으로 제한하는 법률 용어 또는 기술적 조치를 적용하지 않을 수 있다.

## 게시 요약

Bahasa에서 2018년 24월 24일 첫 출간  
2018년 20월 5일 영어로 두 번째 발행  
- 개정 1: 독자의 피드백에 따른 수정 오타/번역 오류

정보 / 수정 / 제안은 다음으로 문의하십시오.

[enquiry@arusacademy.org.my](mailto:enquiry@arusacademy.org.my)

# 목차

권장하는 교육 및 학습 시간 할당	6
준비	7
1 단원 - 마이크로컨트롤러 이론	8
1.1 - 마이크로컨트롤러와 마이크로프로세서란?	9
1.2 마이크로컨트롤러의 구성 요소	12
1.3 - 블록 다이어그램	14
1.4 - 회로도 소개	15
2 단원 - 출력 (Output)	18
2.1-출력 프로그래밍 소개	19
2.2-출력 장치	30
2.3-출력 회로 연결 소개	34
2.4 - 출력 회로 시뮬레이션	42
3 단원 - 입력 (Input)	49
3.1-입력 프로그래밍 소개	50
3.2- 입력 장치	54
3.3-입력 회로 연결 소개	59
3.4 - 입력 회로 시뮬레이션	63



4 단원 - 입력 / 출력 조합	68
4.1 - 조건부 프로그래밍 구조 도입	69
4.2 - 출력 및 입력 회로 연결	76
4.3 - 출력 및 입력 회로 시뮬레이션	78
4.4 - 프로젝트 만들기	83
부록 1 - 안드로이드 스마트폰으로 프로그래밍 하는 법	84
부록 2 - 추가 참조	85
모듈 수업 문제지	87
1 단원- 마이크로컨트롤러	88
2 단원 - 출력 (Output)	92
3 단원 - 입력 (Input)	103
4 단원 - 입력과 출력	111
모듈 수업 문제 답안지	
1 단원- 마이크로컨트롤러	118
2 단원 - 출력 (Output)	121
3 단원 - 입력 (Input)	131
4 단원 - 입력과 출력	138



# 권장하는 교육 및 학습 시간 할당

단원	주제	시간 할당	비고
1 단원	마이크로컨트롤러 이론	2 시간	
2 단원	출력 (Output)	3 시간	특정 주제에 따라 PC 또는 휴대 전화 사용 필요
3 단원	입력 (Input)	2 시간	
4 단원	입력 / 출력 조합	3 시간	

# 준비

## 아두이노 마이크로컨트롤러

- 인투피온 공식 웹스토어 또는 Cytron 웹스토어에서 Maker-UNO 세트를 구입할 수 있습니다.
- 또한 온라인 상점에서 아두이노 세트 및 모듈과 센서도 구입할 수도 있습니다.

<http://m.intopion.com/main/index>

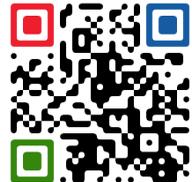


인투피온 웹스토어 QR

## 아두이노 소프트웨어

- 소프트웨어는 아두이노 공식 홈페이지를 통해 다운로드가 가능합니다.

<https://www.Arduino.cc/en/Main/Software>



아두이노 소프트웨어 QR

# 1 단원

# 마이크로컨트롤러

## 학습 표준

- 2.4.1 마이크로컨트롤러 및 마이크로프로세서의 의미 설명
- 2.4.2 마이크로컨트롤러 내부의 부품 설명
- 2.4.3 마이크로컨트롤러에 대한 회로도 스케치

## 평가 표준

- 1 단계 - 마이크로컨트롤러의 의미와 부품 설명
- 2 단계 - 마이크로컨트롤러에서 각 하드웨어의 기능 설명
- 3 단계 - 마이크로컨트롤러를 사용하여 회로도 스케치

## 부 단원

- 단원 1.1 - 마이크로컨트롤러와 그 기능
- 단원 1.2 - 마이크로컨트롤러의 부품
- 단원 1.3 - 블록 다이어그램
- 단원 1.4 - 회로도 소개

## 권장 수업 시간

단원 1.1	30분
단원 1.2	
단원 1.3	30분
단원 1.4	60분

# 단원 1.1

## 마이크로컨트롤러와 마이크로프로세서란?

### 학습 목표

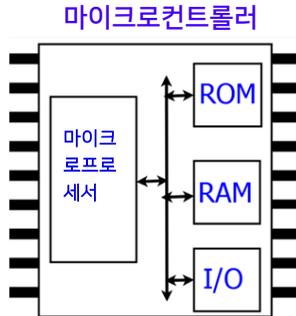
이 단원에서 학생은 마이크로프로세서뿐만 아니라 마이크로컨트롤러의 의미를 말할 수 있을 것이다.

### 성공 기준

학생들은 마이크로프로세서뿐만 아니라 마이크로프로세서의 의미를 구두 또는 서면으로 설명할 수 있다.

**마이크로컨트롤러**는 프로그램 가능한 하드웨어이다.

이 칩 안에는 마이크로프로세서를 중앙 처리 장치(CPU), 랜덤 액세스 메모리(RAM), 읽기 전용 메모리(ROM), 입출력(I/O) 포트를 사용한다. CPU, RAM, ROM 및 I/O 포트는 단일 칩 안에 있다.



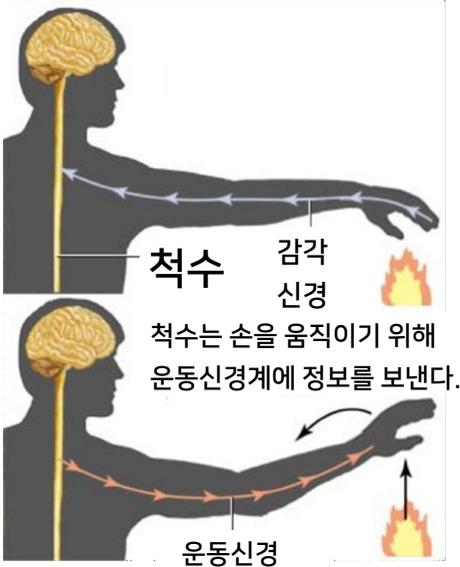
마이크로컨트롤러는 텔레비전, 에어컨, 자동차 등. 우리 주변의 다양한 기기에서 자주 사용된다.

# 단원 1.1

마이크로컨트롤러 입력을 수신하고, 수신한 정보를 처리하며, 원하는 출력을 정확하게 생성한다. 마이크로컨트롤러는 작은 컴퓨터나 뇌로 간주될 수 있다.

도표 1.1(b)과 비교할 수 있다. 마이크로컨트롤러는 우리 몸의 신경계와 같은 방식으로 작동한다. 신경수용체는 척수로 정보를 보내 뇌에서 처리하게 한다. 그러면 뇌는 근육에 적절한 반응을 보낼 것이다.

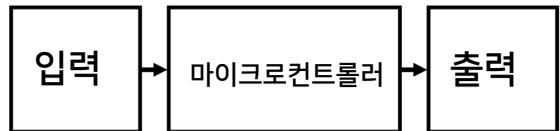
마이크로컨트롤러는 프로그래밍을 통해 특정 작업을 실행하도록 지시 받을 수 있다.



1.1(b) 도표 : 인체신경계 도표

지시를 받은 마이크로컨트롤러는 컴퓨터에 연결할 필요 없이 지시를 저장한다.

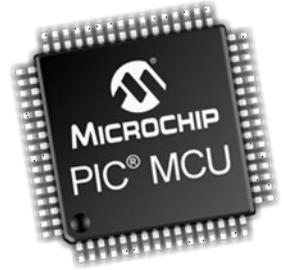
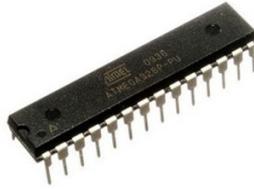
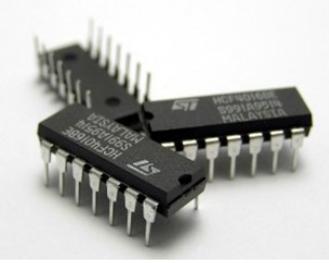
결론적으로 마이크로컨트롤러는 입력으로부터 정보를 수신하고 메모리 내의 프로그램에 따라 반응하는 도표 1.1(c)에 따라 작동한다.



1.1(b) 도표 : 입력에서 부터 마이크로컨트롤러를 거쳐서 출력하는 과정을 보여준다

# 단원 1.1

## 마이크로컨트롤러 단원 (칩셋)



## 일반적으로 사용되는 마이크로컨트롤러 보드들



1.1. (e) 도표 : 인텔 에디슨 (Intel Edison)



1.1. (f) 도표 : 아두이노 우노 (Arduino Uno)



1.1. (g) 도표 : 메이커 우노 (Maker Uno)



1.1. (h) 도표 : 비글본 블랙 (Beaglebone Black)

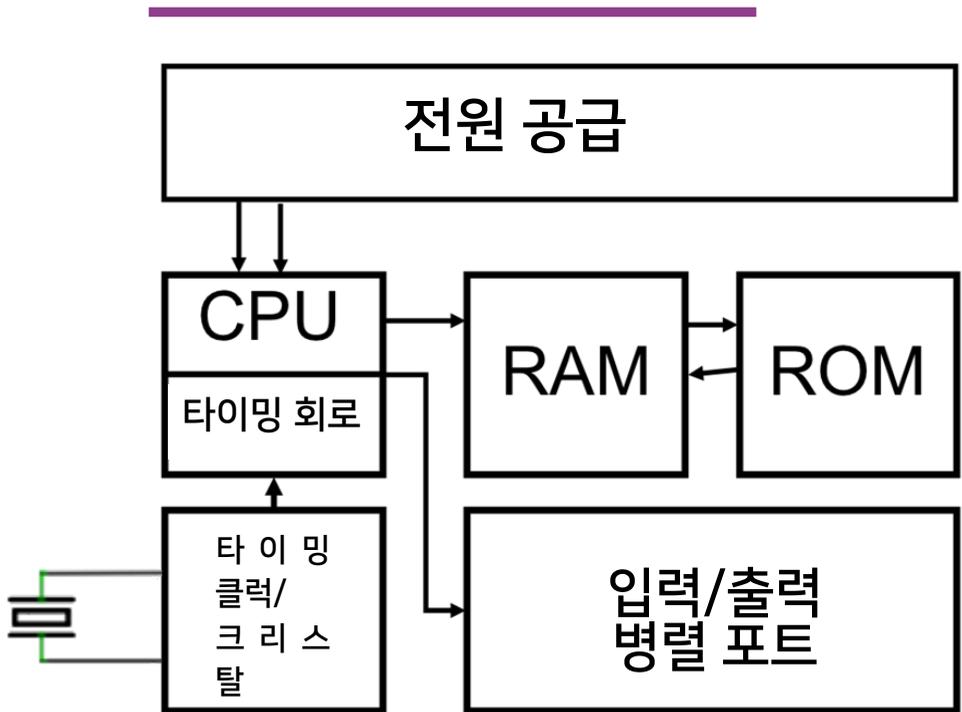
## 단원 1.2 마이크로컨트롤러의 구성 요소

### 학습 목표

이 단원에서는 학생이 마이크로컨트롤러에서 다양한 구성 요소를 설명할 수 있다.

### 성공 기준

학생들은 마이크로컨트롤러의 다양한 구성 요소를 구두 또는 서면으로 설명할 수 있다.



1.2(a) 도표 : 마이크로컨트롤러의 여러 가지 구성 요소를 보여준다.

## 단원 1.2

부품	기능
CPU	정보 및 명령어를 수신하고 입력 및 출력을 처리합니다.
RAM & ROM	명령어를 저장하고 실행할 메모리 공간입니다.
I/O 병렬 포트	LED, 모터 및 센서와 같은 입출력 장치에 연결합니다. 입력 신호와 출력 신호의 두 가지 유형이 있습니다 - 아날로그 신호와 디지털 신호
타이밍 회로	마이크로컨트롤러의 내부 사이클 타이밍을 유지하기 위해 사용됩니다.
타이밍 크리스탈	타이밍 회로의 기준으로 일정한 진동 주파수를 생성합니다.
전원 공급	마이크로컨트롤러에 전력을 제공합니다.

# 단원 1.3

## 블록 다이어그램

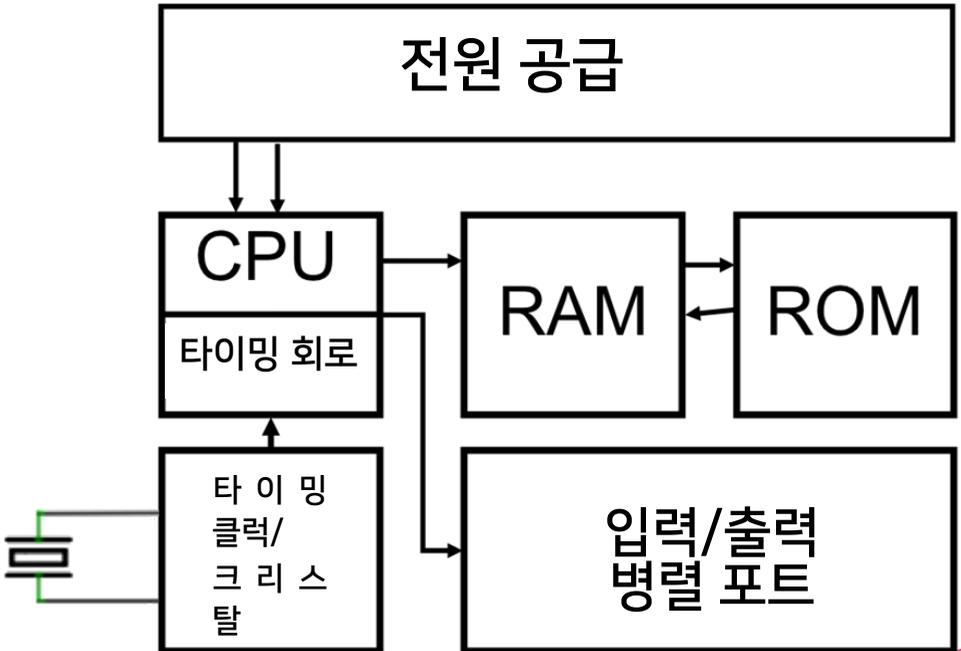
### 학습 목표

이 단원에서 학생은 블록 다이어그램을 그릴 수 있다.

### 성공 기준

학생들은 마이크로컨트롤러, 전원공급장치, 입출력 등이 포함된 블록 다이어그램을 그릴 수 있다..

다음 도표는 블록 다이어그램의 예를 보여준다.



1.3(a) 도표 : 마이크로컨트롤러의 블록 도표이다.

# 단원 1.4

## 회로도 소개

### 학습 목표

이 단원에서 학생은 회로도를 그릴 수 있을 것이다.

### 성공 기준

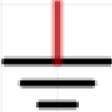
학생들은 도면 기준을 바탕으로 회로도를 정확하게 그릴 수 있다

---

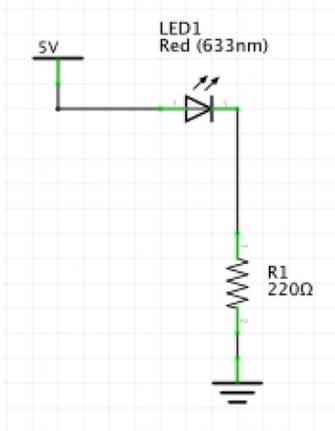
표준화된 회로도를 그리는 지침입니다.

1. 그릴 선이 모두 직선인지 확인한다.
2. 모든 선이 화살표가 아니도록 하여야 한다.
3. 모든 선을 수평 또는 수직으로만 그릴 수 있도록 한다.
4. 혼동을 피하기 위하여 교차하는 선을 최소화한다.
5. 각 구성 요소를 나타내기 위하여 표준 기호를 사용한다.
6. 회로도의 모든 구성요소에 라벨이 부착되도록 한다.

# UNIT 1.4

부품	기능
 <p>접지 (GND)</p>	배터리나 전원 공급 장치의 음극 단자입니다. 전자 회로는 일반적으로 GND로 종료됩니다.
 <p>전원 공급 5V</p>	전원 공급원 회로의 시작점입니다.

다음은 전자 회로의 회로도의 예다.



다음은 전자 회로도의 좋은 예를 보여준다

- 그릴 선이 모두 직선인지 확인한다.
- 모든 선이 화살표가 아니도록 하여야 한다.
- 모든 선을 수평 또는 수직으로만 그릴 수 있도록 한다.
- 혼동을 피하기 위하여 교차하는 선을 최소화한다.
- 각 구성 요소를 나타내기 위하여 표준 기호를 사용한다.
- 회로도의 모든 구성요소에 라벨이 부착되도록 한다.

# UNIT 1.4

다음은 전자 회로의 회로도의 예다.

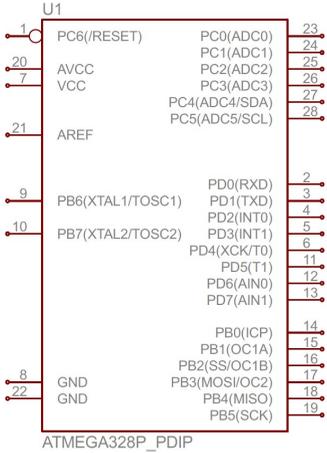


도표 1.4(b): ATMEGA328P 칩에 대한 회로도. (아두이노 UNO용 마이크로컨트롤러)

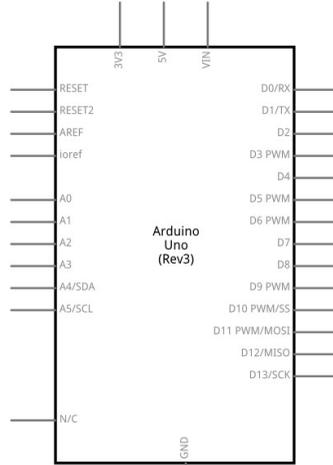


도표 1.4. © : Arduino UNO 개발 보드의 회로도.

다음 다이어그램은 마이크로컨트롤러를 사용하는 전자회로의 회로도다.

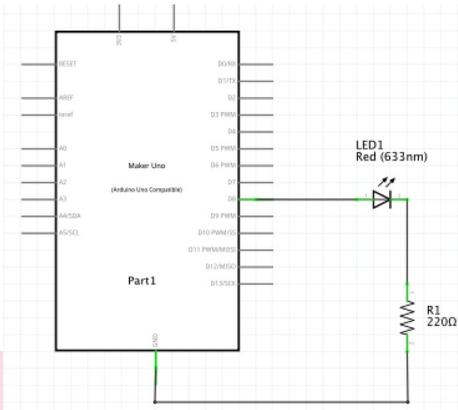


그림 1.4. (d) : Pin D8에서 GND로 연결된 LED 회로의 회로도.

# 단원 2

## 출력

### 학습 표준

2.4.4 전용 소프트웨어로 작동 시뮬레이션 회로를 제작한다.

2.4.5 마이크로컨트롤러의 입력 및 출력 회로 연결

2.4.6 입력 및 출력 회로에 기반한 간단한 프로그램 작성

### 평가 표준

4 단계 마이크로컨트롤러를 포함하는 회로의 기능을 시험한다.

### 부 단원

단원 2.1—출력 프로그래밍 소개

단원 2.2—출력 장치

단원 2.3—출력 회로 연결 소개

단원 2.4 - 출력 회로 시뮬레이션

## 권장 수업 시간

단원 2.1	60분
단원 2.2	60분
단원 2.3	60분
단원 2.4	60분

## 준비

- 1) 학생 한 명당 아두이노 / 메이커 UNO가 충분한지 확인한다.
- 2) 단원 2.1은 마이크로컨트롤러를 프로그래밍하기 위해 컴퓨터나 스마트폰을 사용해야 한다.  
(부록 1 참조)
- 3) 단원 2.4는 컴퓨터와 인터넷이 필요하다.
- 4) 학생 모듈과 함께 사용 가능

# 단원 2.1

## 출력 프로그래밍 소개

### 학습 목표

이 단원에서 학생은 출력 회로를 제어하기 위한 간단한 프로그램을 작성할 수 있다.

### 성공 기준

학생들은 Maker UNO / 아두이노에서 LED 내장 제어 가능

마이크로컨트롤러는 마이크로컨트롤러의 출력 핀으로 송신하여 출력 회로를 제어할 수 있다. 이를 통해 핀에 연결된 모든 구성 요소는 프로그래밍을 통해 제어될 수 있다.

일반적으로 마이크로컨트롤러는 아날로그 신호와 디지털 신호라는 두 가지 다른 유형의 신호를 보내지 않도록 만들어질 수 있다.

디지털 신호는 두 개의 상태-ON(회로에서 흐르는 전류) 또는 OFF(회로에서 전류 없음)를 가지고 있다. 핀 5에 ON 디지털 신호를 보내면 핀에 연결된 모든 전자 부품을 활성화할 수 있다. 마찬가지로 핀 5에 OFF 디지털 신호를 보내면 핀 5에 연결된 모든 구성 요소를 비활성화하거나 끌 수 있다.

아날로그 신호는 두 개의 상태-ON(회로에서 흐르는 전류) 또는 OFF(회로에서 전류가 흐르지 않음)를 가진다. 핀 5로 ON 디지털 신호를 보내면 핀에 연결된 모든 전자 부품을 작동시킬 수 있다. 마찬가지로 핀 5에 OFF 디지털 신호를 보내면 핀 5에 연결된 구성 요소를 비활성화하거나 끌 수 있다.



Analog Signal



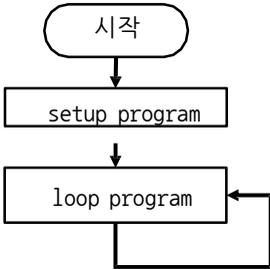
Digital Signal

반면에 아날로그 신호는 두 개 이상의 다른 신호를 사용한다. 아두이노의 경우, 0부터 255까지의 256개의 값이 있다. 0은 회로에 전류가 없음을 의미하며(전압값 0) 255는 회로의 최대 전압이다. 아날로그 값(0 ~ 255)은 출력 핀으로 전송될 전압 값에 정비례한다(예를 들어, 127은 최대 전압 값의 50%를 송신한다).

즉, LED를 아날로그 핀으로 연결하면 0이 핀을 끄고 127이 LED를 50% 밝기로 켜는 반면 100은 100% 밝기로 켜는 것이다.

# 단원 2.1

마이크로 컨트롤러는 두 가지 주요 기능 설정(setup) 기능과 루프(loop) 기능을 실행합니다. 설정(setup) 기능은 Arduino를 켜거나 재설정 한 후에 한 번만 실행되는 반면 루프(loop) 기능은 Arduino가 꺼질 때까지 계속 실행됩니다.



Arduino를 프로그래밍하기 위해 Arduino IDE (Intergrated Development Environment)를 사용할 수 있습니다. 이 소프트웨어는 Arduino 웹 사이트 [www.arduino.cc](http://www.arduino.cc)에서 다운로드 할 수 있습니다.

Arduino IDE 소프트웨어를 연 후 코드를 쉽게 읽을 수 있도록 코드의 줄 번호를 표시하는 것이 좋습니다. 이렇게 하려면 파일-> 기본 설정을 클릭하고 표시 줄 번호 옵션을 선택한 다음 확인을 클릭하십시오.

도표 2.1(b): 아두이노의 순서도

Arduino IDE의 사용자 인터페이스는 그림 2.1 (c)의 사용자 인터페이스와 비슷합니다. 두 기능 (설정 및 루프)을 쉽게 사용할 수 있는 방법에 주목하십시오.

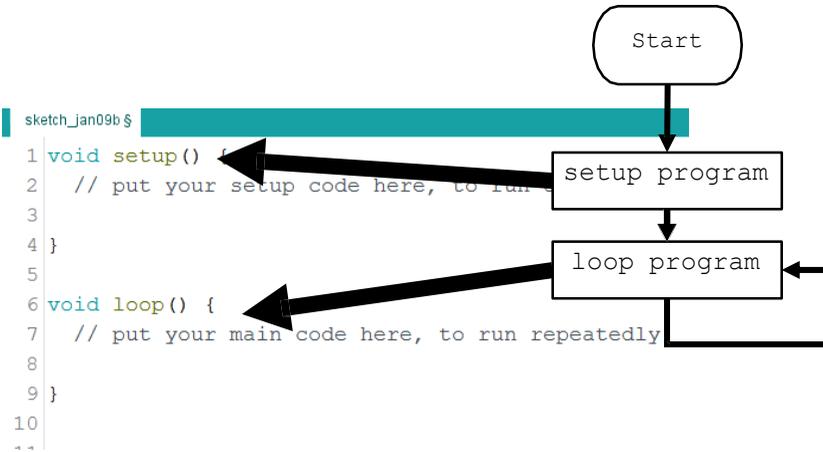


도표 2.1(b): 아두이노의 순서도와 Arduino IDE 프로그래밍 비교

# 단원 21

Arduino 프로그램 작성 단계의 좋은 예

## pinMode in Void setup()

핀 번호와 사용할 모드(입력 모드 또는 출력 모드)를 식별하고 지정하십시오.

## 프로그램 계획

프로그램으로 무엇을 성취하고 싶은지 생각해보십시오.  
필요한 경우 순서도 스케치

## "Void loop()"로 코드를 작성하십시오

루프 함수로 코드를 작성하십시오. 코드가 출력 신호를 보내는 것을 포함 할 수 있습니다.  
입력 신호를 기반으로 어떤 출력 신호를 보낼지 알 뿐만 아니라  
입력 신호를 수신하고 읽습니다.

아두이노 프로그래밍의 규칙.

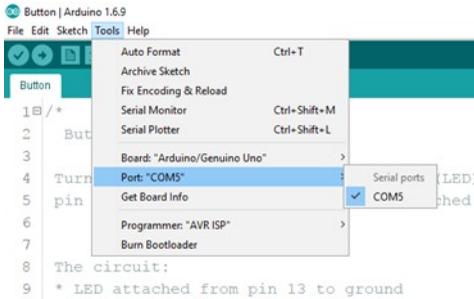
- 각 코드 줄은 세미콜론 기호로 끝나야합니다. ;
- 키워드의 철자가 올바른지 확인하십시오 (주황색 / 파란색).
- ()와 {}가 쌍을 이루어야 합니다.

# 단원 2.1

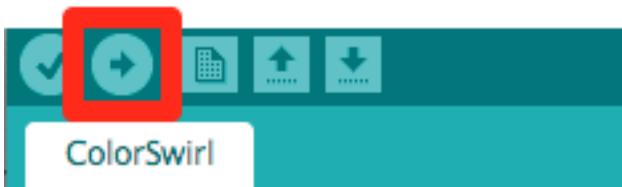
Arduino 프로그램 작성 단계의 좋은 예

## USB 케이블로 Arduino를 컴퓨터에 연결

USB 케이블로 Arduino 보드를 컴퓨터에 연결한 다음 올바른 포트를 선택하십시오.



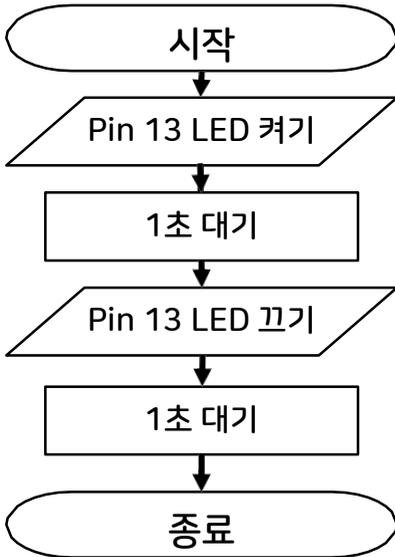
업로드 아이콘(화살표)을 누르고 적절한 이름으로 파일 저장



# 단원 21

코드가 업로드 되면 코드는 아두이노 메모리에 남아 있습니다.  
 프로그램을 지우거나 덮어 쓰기 전까지는 계속 유지 됩니다.  
 프로그램을 작성하기 전에 순서도를 사용하여 프로그램을 계획 해야합니다.

다음과 같은 프로그램을 작성합니다



이것은 분기를 포함하지 않는 순차적인 구조의 프로그램입니다. 다른 종류의 코드를 나타내는데 다른 모양이 어떻게 사용되는지 확인하십시오.

 코드의 시작 또는 끝을 나타냄

 알고리즘의 입력 또는 출력을 나타냄

 알고리즘에서 단계/절차를 나타냄

# 단원 2.1

코드의 순서도는 다음과 같습니다.

```

1 void setup() {
2   pinMode(13,OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(13,HIGH);
7   delay(1000);
8   digitalWrite(13,LOW);
9   delay(1000);
10 }
--

```

각 한 Line당 기능은 다음과 같습니다

Line	프로그래밍 문구	기능
1	<code>void setup() {</code>	<p>setup 함수의 시작을 표시합니다. setup 함수는 일반적으로 사용할 핀의 모드를 정의하는 데 사용됩니다. void와 setup의 색상이 어떻게 다른지 확인하십시오. 이는 Arduino 프로그래밍에서 특수 키워드임을 나타냅니다.</p> <p>{ 기호는 setup 함수에 대한 코드의 시작을 나타냅니다.</p>

# 단원 21

Line	프로그래밍 문구	기능
2	<code>pinMode(13,OUTPUT);</code>	<p>핀이 입력 핀인지 출력 핀인지 여부를 설정합니다.</p> <p>이 선은 핀 13을 출력 핀으로 설정합니다</p>
3	<code>}</code>	<p><code>}</code> 기호는 설정 기능의 끝을 표시합니다. 라인 1의 <code>{</code> 기호부터 시작하여 설정 기능에는 <code>{</code> 와 <code>}</code> 사이의 코드만 포함되며, 이 코드는 라인 1에서 3까지만 포함됩니다.</p>
5	<code>void loop() {</code>	<p>이 라인은 루프 기능의 시작을 나타냅니다. 루프 기능은 아두이노가 꺼질 때까지 무기한으로 실행되는 기능입니다. <code>void</code>와 <code>loop</code>라는 단어는 Arduino에서도 키워드이므로 서로 다른 색상으로 표시 됩니다.</p> <p><code>{</code> 기호는 루프 기능의 시작을 나타냅니다.</p>

# 단원 21

Line	프로그래밍 문구	기능
6	<code>digitalWrite(13, HIGH);</code>	<p><code>digitalWrite</code>는 디지털 신호를 지정된 핀으로 보냅니다. 이 라인은 핀 13에 HIGH 신호를 보내며 핀 13에 연결된 모든 구성 요소가 켜집니다.</p> <p>Arduino는 대소 문자를 구분하며 <code>digitalWrite</code>로 써야하며 HIGH는 모두 대문자로 써야합니다. 그렇지 않으면 키워드로 인식되지 않으며 다른 색으로 표시됩니다.</p>
7	<code>delay(1000)</code>	<p><code>delay</code>는 지정된 시간 동안 기다립니다. 괄호 안에 쓰여진 값은 밀리 초 단위로 측정됩니다. 예를 들어 1000 밀리 초는 1 초에 해당합니다.</p> <p>따라서 <code>delay(1000)</code>는 시스템이 정지하고 1 초 동안 대기 함을 의미합니다.</p>
8	<code>digitalWrite(13, LOW)</code>	<p>이 라인은 LOW를 PIN 13으로 보냅니다. 핀 13에 연결된 구성 요소를 끄거나 비활성화합니다.</p>

## 단원 2.1

Line	프로그래밍 문구	기능
9	<code>delay(1000)</code>	<p>이 라인없이 핀 13을 끈 후에 또 다른 delay를 기입해야합니다. 라인 8이 실행 된 후 프로그램 흐름은 라인 6으로 돌아갑니다.</p> <p>핀 13이 꺼져 있지만 꺼짐 시간이 너무 짧으며 육안으로는 감지 할 수 없으므로 켜져 있지 않은 것처럼 보입니다.</p>
10	<code>}</code>	이 기호는 루프 기능의 끝을 나타냅니다. 라인 5의 {기호에서 시작하여 루프 기능에는 라인 5와 라인 10 사이의 라인 만 포함됩니다.

## 단원 2.1

`analogWrite`를 사용하여 아날로그 신호를 보낼 수도 있습니다. 아날로그핀은 보드에 ~ 기호가 있는 핀만 아날로그 신호를 생성합니다. 아날로그 핀은 Pin 3,5,6,7, 9, 10 및 11입니다.

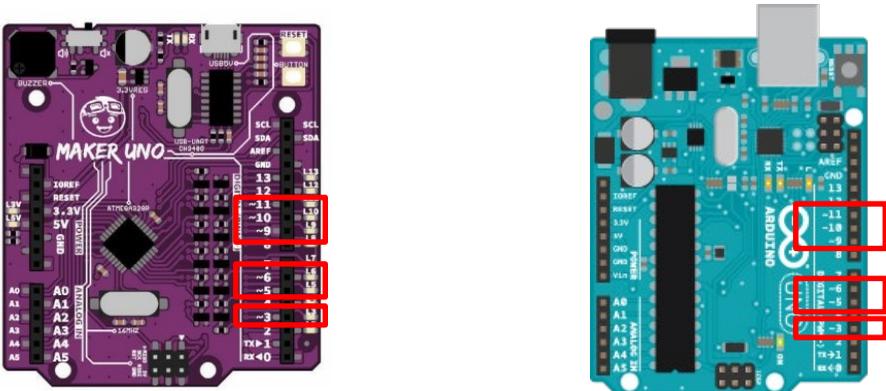


도표 2.1. (d) : 아날로그 핀의 위치 (PWM 핀이라고도 함)

`analogWrite`는 다음과 같이 작동합니다.

`analogWrite` (핀, 값) 값의 범위는 0에서 255 사이입니다.

예를 들어,

`analogWrite (9, 255)`는 핀 9에 최대 전압 인 255를 전송합니다. 핀 9에 연결된 모든 구성 요소는 최대 전압을 받습니다. 핀 9에 LED가 있으면 100 % 밝기로 켜집니다.

`analogWrite (9, 127)`는 핀 9에 127 (255의 절반)을 전송합니다. 핀 9에 연결하면 최대 전압의 50 %가 수신됩니다. 핀 9에 LED가 있으면 LED 50 % 밝기로 켜집니다.

`analogWrite (9, 0)`는 핀 9에 0을 보냅니다. 핀 9에 연결된 모든 구성 요소는 0 전압을 받습니다. 핀 9에 LED가 있으면 전혀 켜지지 않습니다.

# 단원 2.1

다음 섹션은 Maker UNO만을 위한 섹션입니다.  
다른 유형의 Arduino에서는 LED를 회로에 연결해야 합니다.

Maker UNO 는 모든 디지털 핀에 LED가 있습니다.  
프로그래밍을 작성하여 모든 핀의 LED를 제어할 수 있습니다.

예를 들어 다음과 같습니다.

```
digitalWrite(9, LOW);  
digitalWrite(8, HIGH);  
delay(1000); digitalWrite  
rite(9, HIGH); digita  
lWrite(8, LOW); delay  
(1000);
```

이 프로그램을 실행하면 pin 9와 pin 8의 LED가 번갈아 깜박입니다.

# 단원 2.2

## 출력 장치

### 학습 목표

이 단원에서는 학생들이 다양한 출력 장치를 사용하는 방법을 배울 수 있게 될 것이다.

### 성공 기준

학생들은 출력 장치의 이름과 기능을 나열할 수 있을 것이다.

출력 장치는 연결된 핀이 Arduino에서 주는 신호를 실행합니다. Arduino는 5V를 제공하여 각 핀에 연결된 장치를 켤 수 있습니다. 프로젝트 제작에 일반적으로 사용되는 출력 장치는 발광 다이오드 (LED) 및 부저입니다. 이 출력 장치는 직류 모터 12v, LED 조명 12v 등과 같은 외부 전원 공급 장치없이 켜집니다.

## 저항

- 저항 값이 높을수록 통과 저항이 높아짐.
- 저항은 장치가 타는 것을 방지하거나 전압 분배기 역할을 하기 때문에 중요.
- 저항기의 색상 밴드를 보면 저항기 값을 읽을 수 있음.
- 회로도의 저항 기호는 다음과 같다.  $\text{---}\omega\text{---}$



도표 2.2.(b) : 10,000 (10k) Ohm 저항

www.resistorguide.com

Color	Significant figures	Multiply	Tolerance (%)	Temp. Coeff. (ppm/K)	Fail Rate (%)
Bad	black	0 0 0	x 1	250 (U)	
Beer	brown	1 1 1	x 10	1 (F)	100 (S)
Reds	red	2 2 2	x 100	2 (G)	50 (R)
Our	orange	3 3 3	x 1K		15 (P)
Young	yellow	4 4 4	x 10K		25 (Q)
Guns	green	5 5 5	x 100K	0.5 (D)	20 (Z)
But	blue	6 6 6	x 1M	0.25 (C)	10 (T)
Vodka	violet	7 7 7	x 10M	0.1 (B)	5 (M)
Goes	grey	8 8 8	x 100M	0.05 (A)	1 (K)
Well	white	9 9 9	x 1G		
Get	gold	3rd digit only for 5 and 6 bands	x 0.1	5 (J)	
Some	silver		x 0.01	10 (K)	
Now!	none			20 (M)	

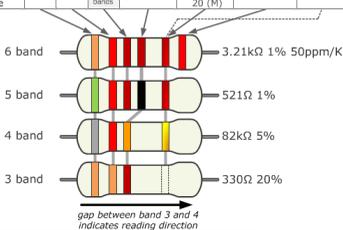


도표 2.2.(a) : 저항 값을 읽는 방법 (출처: Resistorguide.com)

## 단원 2.2

# 발광 다이오드 (LED)

- 발광 다이오드는 전류가 통과 할 때 빛을 방출하는 다이오드이다.
- LED는 장식용 조명으로 사용할 수 있으며 타지 않도록 저항과 함께 사용해야 한다.
- LED에는 극성이 있습니다. 양극과 음극이 있다.
- Arduino의 모든 핀에는 내장 LED가 있다
- LED 기호는 다음과 같다.

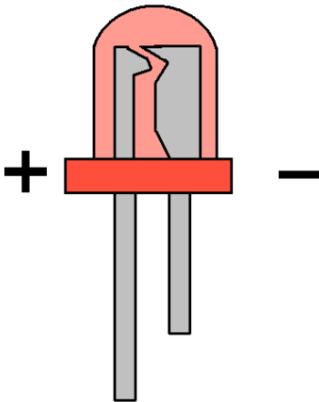
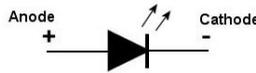


도표 2.2(c):  
LED의 양극 및 음극 다리를 식별하는 방법

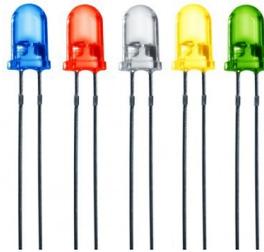


도표 2.2(d):  
발광 다이오드 (LED)

## 단원 2.2

### 부저

- 부저들은 전기 신호를 소리로 바꾸는 방식으로 작동한다.
- LED처럼 부저에도 극성이 있다.
- 부저를 알람이나 멜로디로 사용할 수 있다.
- 부저 기호는 다음과 같다.



도표 2.2(e):  
다양한 타입의 부저들

Maker Uno에 내장 부저가 있습니다.

이 부저는 핀 8에 연결되며 슬라이딩 스위치로 켤 수 있습니다.  
사용하기 전에 스위치를 밀어 부저를 활성화 해야합니다.

스위치를 밀어 부저를 활성화

부저

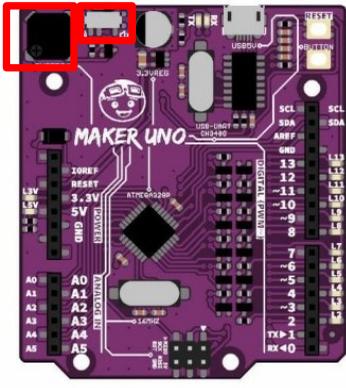


도표 2.2(f):  
부저와 부저 스위치의 위치

# 단원 2.2

## 직류 (DC) 모터

- 3v, 6v, 9v, 12v 등과 같이 전압 요구 사항이 다른 모터에는 여러 가지 유형이 있습니다.
- 전류가 흐르면 DC 모터 샤프트가 회전합니다.
- 일반적으로 모터에는 많은 양의 전류가 필요하므로 드라이버 또는 트랜지스터 및 다이오드에 연결해야 합니다.
- 모터에 연결된 드라이버를 사용하면 시계 방향 또는 시계 반대 방향으로 모터 회전 방향을 제어 할 수 있습니다.
- 표준 Maker UNO 학습 키트를 사용하는 경우 학습 키트에 3V DC 모터가 포함되어 있습니다. 메이커 UNO는 5V 이상의 전압이 필요한 모터를 켤 수 없습니다.
- Arduino는 모터를 켜기에 충분한 전류를 보낼 수 없으므로 트랜지스터와 함께 모터를 사용해야 합니다.
- DC 모터의 회로도는 다음과 같습니다. 



도표 2.2(g):  
각기 다른 타입의 모터들

## 단원 2.3

### 출력 회로 연결

#### 학습 목표

이 단원에서는 학생들이 회로도를 읽고 회로도를 기반으로 회로를 연결할 수 있다.

#### 성공 기준

학생들은 회로도에 따라 적어도 하나의 회로를 연결할 수 있다.

점퍼 와이어와 브레드 보드를 사용하여 출력 회로를 연결할 수 있습니다.

브레드 보드는 점퍼 와이어를 사용하거나 납땀하지 않고 회로에 연결하여 회로를 테스트 할 수 있는 프로토 타입 보드입니다.

시작하기 전에 다음 구성 요소에 대해 더 자세히 알아 보시다.

### 점퍼 와이어

- 점퍼 와이어는 마이크로 컨트롤러의 다른 구성 요소를 마이크로 컨트롤러에 연결하는데 사용됩니다.
- 비공식적으로 빨간색 점퍼 와이어는 전원 공급 장치로 가는 연결에 사용되며 검은 점퍼 와이어는 접지로 연결하는 데 사용됩니다.



도표 2.3(a) : 점퍼와이어

# 단원 2.3

## 브레드 보드

- 브레드 보드는 납땀을 사용하지 않고 프로토 타입 회로를 테스트하고 생산하는데 사용됩니다.
- 구멍은 도표 2.3 (a)와 같이 빨간색 선을 따라 연결됩니다.  
구성 요소 또는 와이어가 동일한 회선에 있는 경우 연결됩니다.

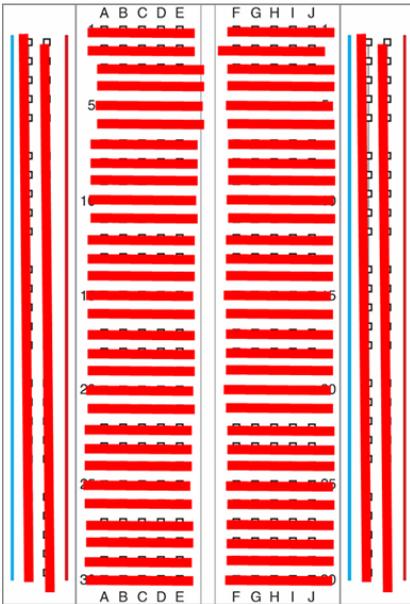


도표 2.3. (a):  
빨간색 선은 브레드 보드의 구멍  
사이의 연결점을 보여줍니다.

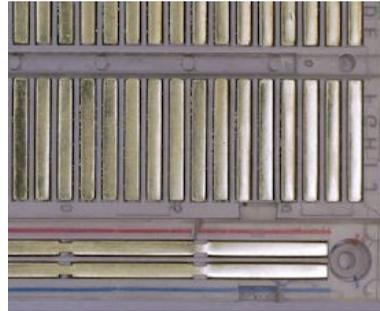


도표 2.3. (b):  
브레드 보드의 구멍 밑면의  
사이의 연결 선상을 보여줍니다.

# 단원 2.3

## 브레드 보드 회로 연결

### LED 회로 연결

아래 회로는 3V 배터리 및 브레드 보드를 사용하는 간단한 직렬 및 병렬 회로이며, 회로도 및 연결을 보여줍니다.

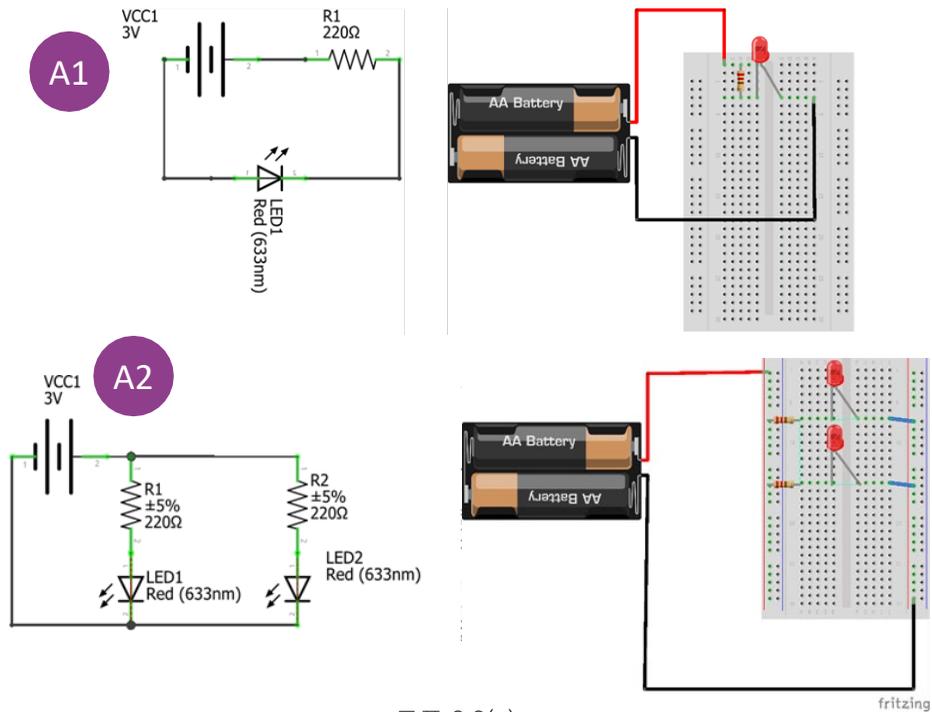
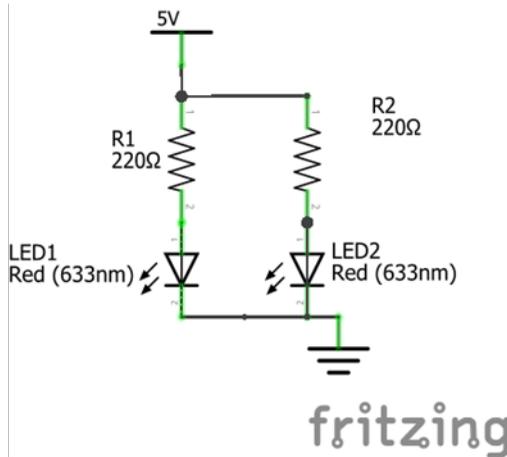
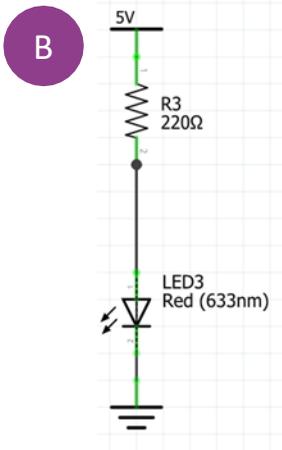


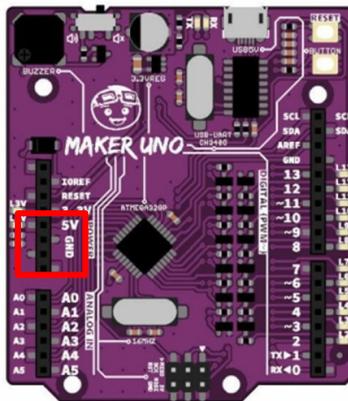
도표 2.2(a) :  
A1 (상단) 직렬 회로  
A2 (하단) 병렬 회로

## 단원 2.3

전원을 5V 전원으로 교체합니다 :



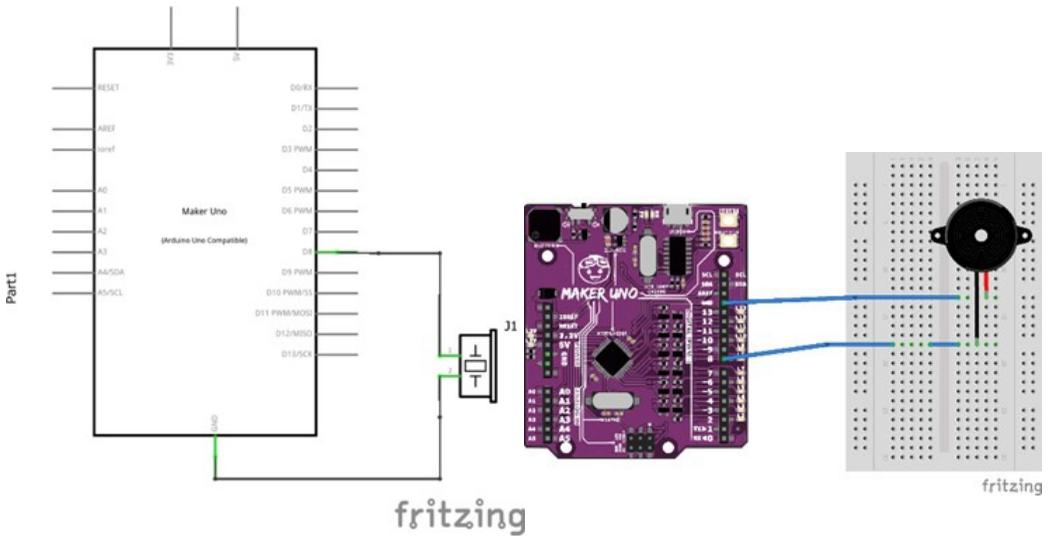
Arduino 또는 Maker UNO는 5V 핀 (+) 및 GND (-) 핀으로 배터리와 같은 5V 전압을 제공 할 수 있습니다.



## 단원 2.3

### 부저

부저는 LED와 유사한 극성 (빨간색 핀, 검은 색 접지)에 따라 연결해야 합니다. 다음 회로는 Arduino에 대한 부저 연결의 예입니다. Maker UNO의 경우 여기에는 핀 8에 내장 된 부저가 있습니다. 사용하기 전에 슬라이더 스위치를 눌러 부저를 활성화해야 합니다.



## 단원 2.3

```

1 void setup() {
2   // put your setup code here, to run once:
3   pinMode(8, OUTPUT);
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   tone(8, 165, 500);
9   delay(1000);
10  tone(8, 175, 500);
11  delay(1000);
12  tone(8, 196, 500);
13  delay(1000);
14 }
15
16

```

Line	프로그래밍 문구	기능
1	tone (8, 165, 500)	이 부저는 핀 8에서 500밀리초 동안 음악의 E3(미) 노트인 165Hz에서 소리를 재생 합니다. tone은 부저를 제어하는 데 사용되는 특수 기능입니다. 핀 번호, 재생할 사운드 주파수 및 재생할 사운드 지속 시간의 3 가지 매개 변수를 사용합니다.

## 단원 2.3

톤은 음을 생성 할 수 있습니다.

다음 목록은 음악 음표 (3 옥타브) 및 주파수 목록을 보여줍니다.

음계	주파수
E <b>미</b>	165
F <b>파</b>	175
G <b>솔</b>	196
A <b>라</b>	220
B <b>시</b>	247
C <b>도</b>	262
D <b>레</b>	294
E <b>미</b>	330
F <b>파</b>	349



전체 음계 목록을 보려면

<https://www.arduino.cc/en/Tutorial/ToneMelody?from=Tutorial.Tone>

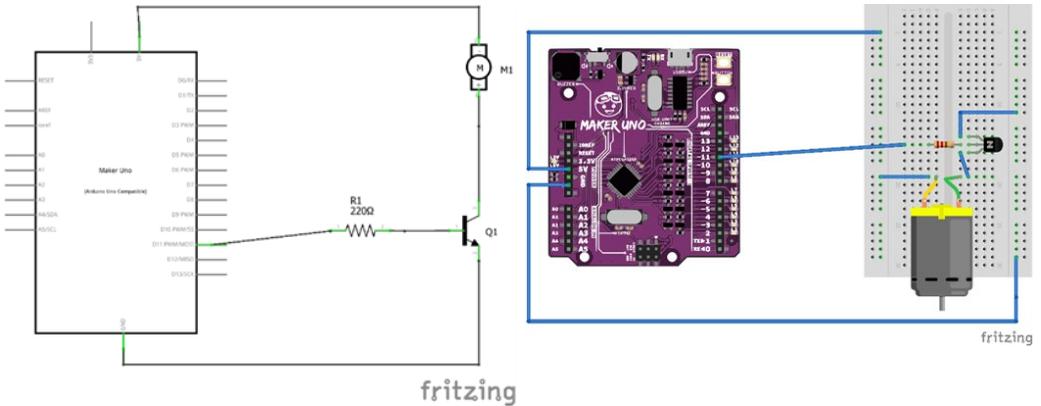
을 방문 해보세요.

노트 (옥타브 1 ~ 옥타브 8)

## 단원 2.3

### 직류 (DC) 모터

DC 모터는 트랜지스터와 함께 연결해야 합니다. 모터가 작동하려면 더 높은 전류가 필요하고 이 전류는 Arduino의 출력 핀으로 제공 할 수 없기 때문입니다. 회로 연결은 다음과 같습니다.



이 문제를 극복하기 위해서는 트랜지스터가 Hte 모터를 제어하는 스위치로서 기능할 필요가 있습니다.

모터를 켜려면 11번 핀으로 신호를 보낸 다음 트랜지스터를 열어 전류가 모터로 흐르도록 해야 합니다.

모터를 켜기 위한 프로그래밍은 다음과 같습니다.

```
sketch_jan03a §
1 void setup() {
2   // put your setup code here,
3   pinMode(11, OUTPUT);
4 }
5
6 void loop() {
7   // put your main code here,
8   digitalWrite(11, HIGH);
9 }
```

## 단원 2.4

### 출력회로 시뮬레이션

#### 학습 목표

이 단원에서 학생들은 전용 소프트웨어를 사용하여 출력 회로를 시뮬레이션 할 수 있다.

#### 성공 기준

학생들은 코드 시뮬레이션과 함께 하나 이상의 회로를 시뮬레이션 할 수 있다.

우리는 프리웨어인 Tinkerlab을 사용하여 출력 회로를 시뮬레이션 할 것입니다.

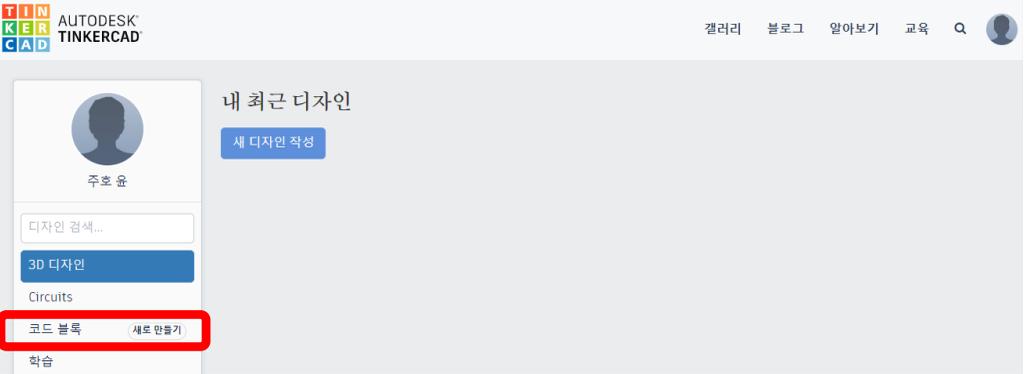
웹링크 : <https://www.tinkercad.com/>입니다.

Tinkercad에 액세스하는 단계는 다음과 같습니다.

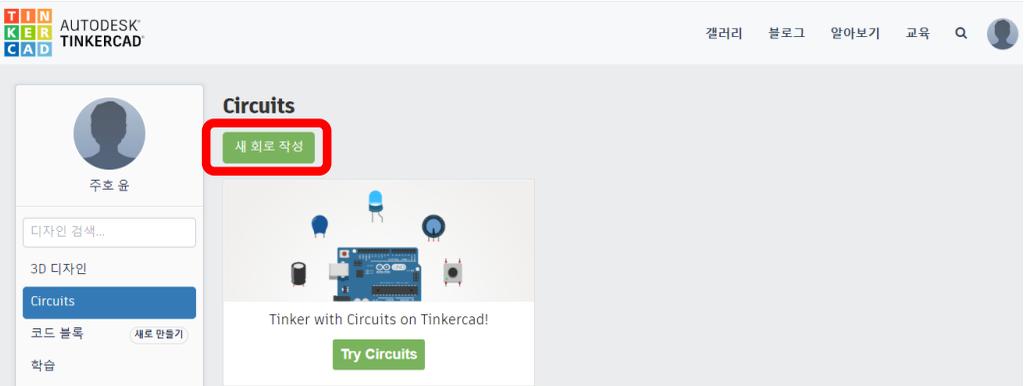


# 단원 2.4

로그인 후 Circuit란을 클릭 해주세요.

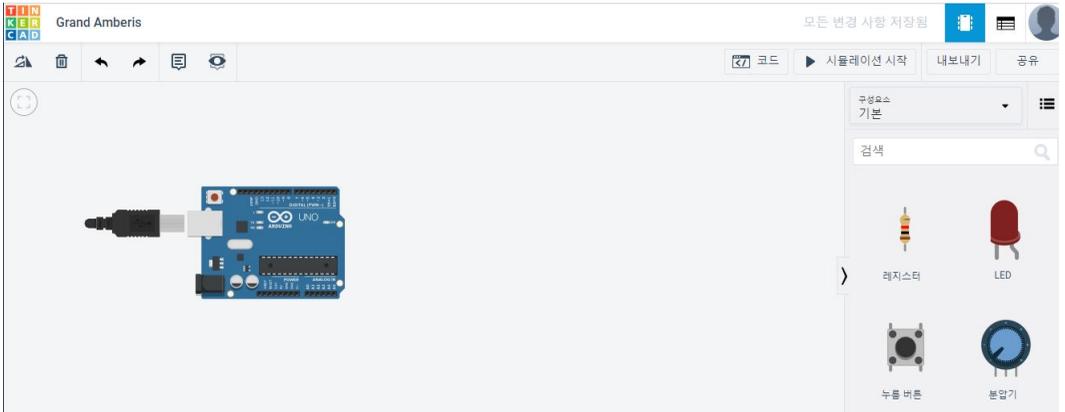


새 회로 작성을 클릭 해주세요.

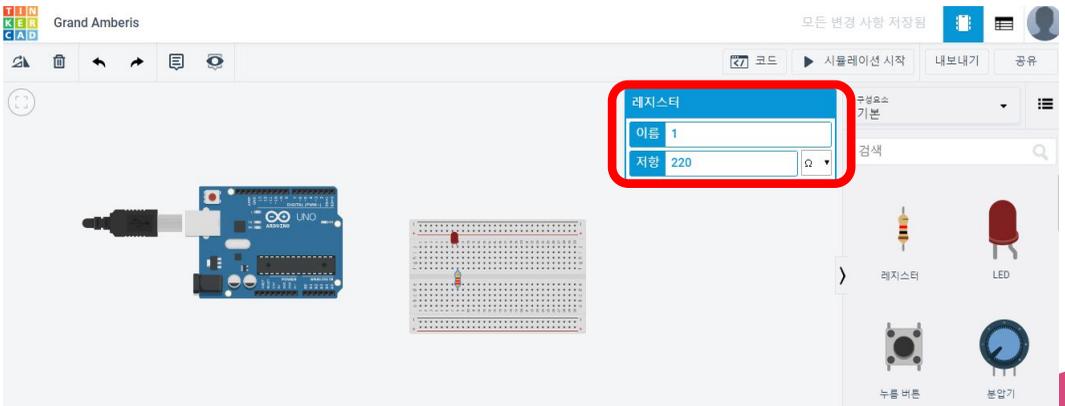


# 단원 2.4

LED, 저항, 아두이노 및 브레드보드 추가 해주세요

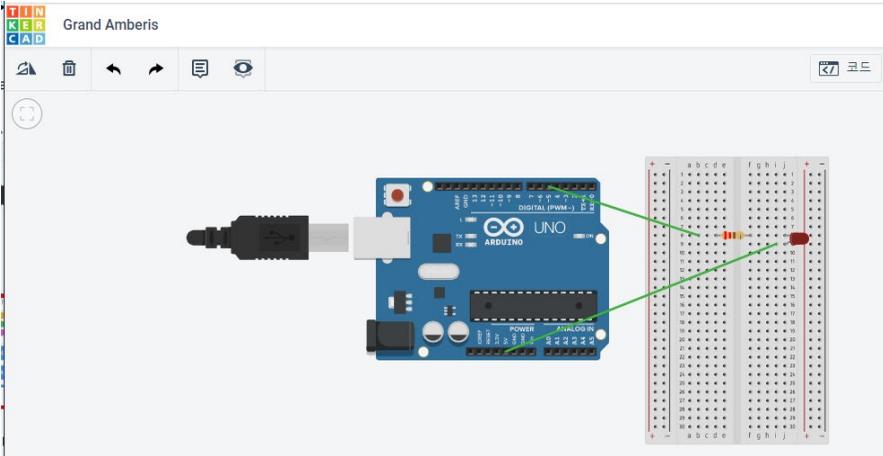


저항 추가 시 레지스터(저항)란에 알맞은 저항 값을 기입 해주세요

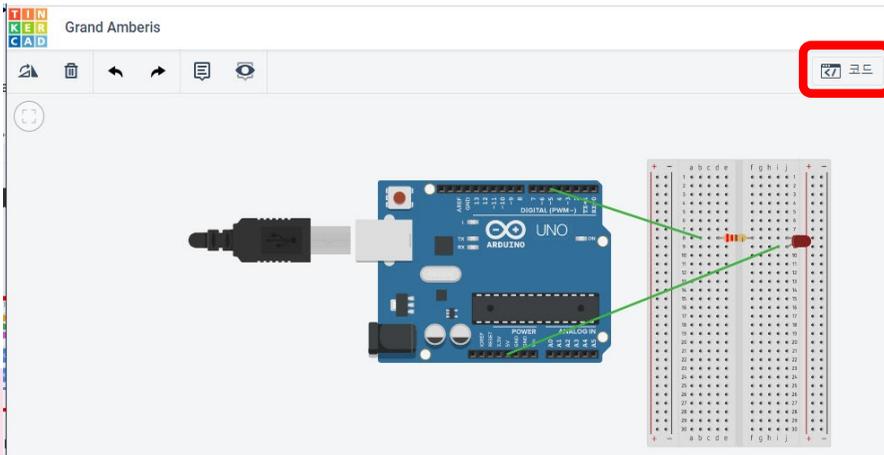


# 단원 2.4

브레드보드와 장치들을 클릭하여 연결해주세요.  
Pin 5 -> 저항, 5V -> LED (+)



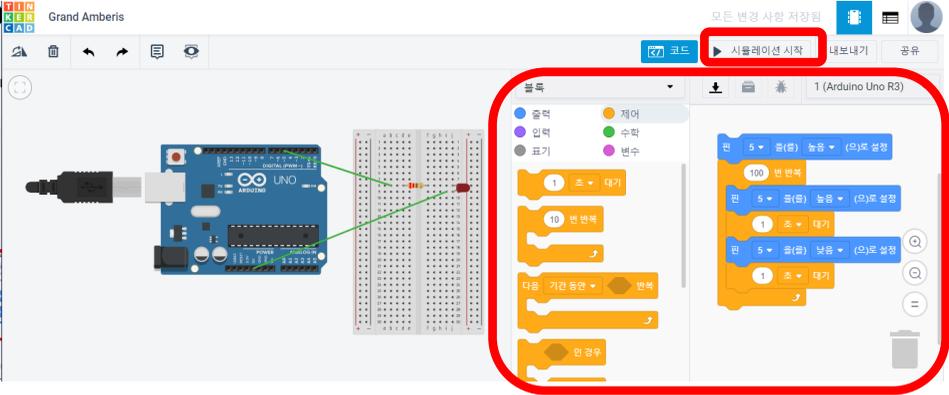
코드 버튼을 클릭하여 코딩을 해보세요



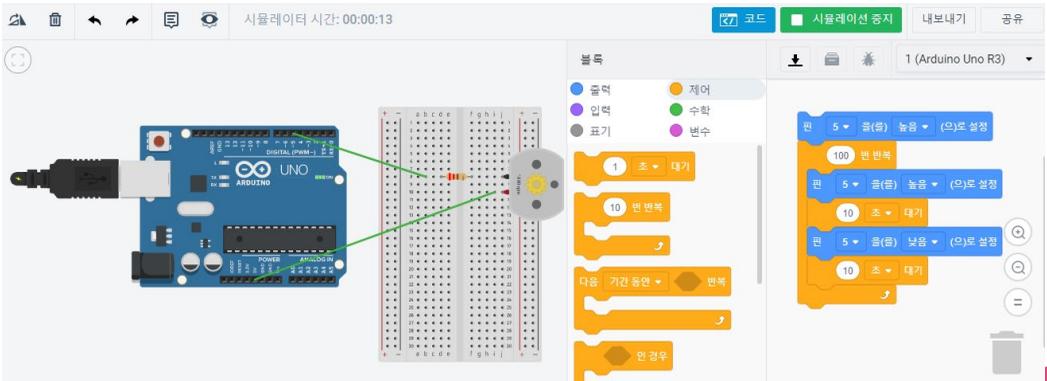
# 단원 2.4

빨간 상자 안에 관련 블록을 끌어다 놓아서  
pin 5에 연결된 LED를 깜박이는 코드를 만들어 보세요

코드를 만들어 보았다면, 시뮬레이션 시작 버튼을 눌러 확인해 보세요



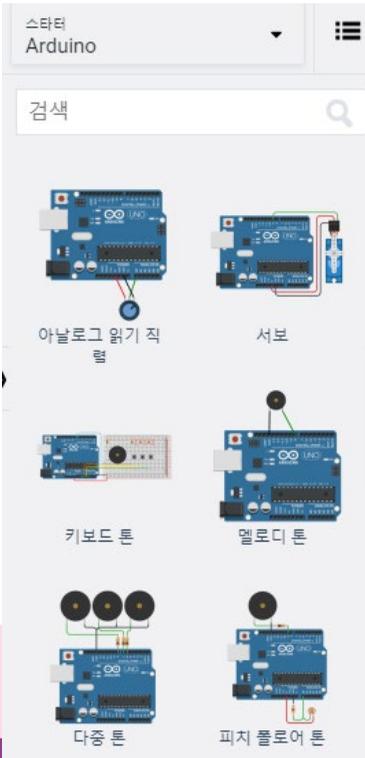
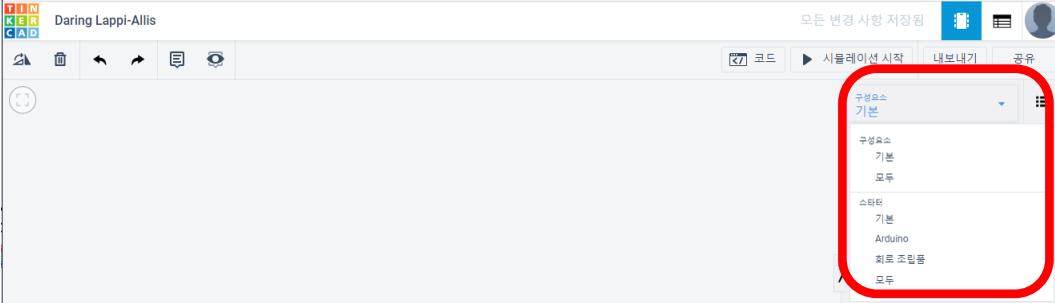
1개 이상의 LED, 부저 또는 모터로 출력 회로를 시뮬레이션 해보세요.



# 단원 2.4

## 구성요소에서 스타트업 기능 사용하기

오른쪽의 구성요소 탭을 클릭하고 스타터 그룹안에 있는 아두이노 란을 클릭해주세요



미리 준비된 구성에 맞춰서 코딩을 연습을 해봅시다.

## 단원 2.4

기본, 아두이노, 회로 조립품 까지 모든 과정을 수행하면서 하드웨어 회로와 코딩에 대한 이해에 도움이 될 수 있습니다.



# 단원 3

## 출력

### 학습표준

2.4.4 전용 소프트웨어로 기능하는 시뮬레이션 회로 구축.

2.4.5 마이크로컨트롤러의 입력 및 출력 회로 연결

2.4.6 입력 및 출력 회로를 기반으로 간단한 프로그램 작성

### 평가표준

4 단계 마이크로컨트롤러를 포함하는 회로의 기능을 테스트한다.

### 부 단원

단원 3.1 입력 회로 프로그래밍 소개

단원 3.2 입력 장치의 유형

단원 3.3 입력 회로 연결 소개

단원 3.4 입력 회로 시뮬레이션

### 권장 수업 시간

단원 3.1 60분

단원 3.2 30분

단원 3.3

단원 3.4 30분

### 준비

- 1) 학생 1명당 아두이노/Maker UNO가 충분한지 확인한다.
- 2) 단원 3.1은 마이크로컨트롤러를 프로그래밍하기 위해 컴퓨터나 스마트폰을 사용해야 한다. (부록 1)
- 3) 단원 3.4는 컴퓨터 및 인터넷 필요
- 4) 학생 모듈과 함께 사용 가능

## 단원 3

# 입력 회로 프로그래밍 소개

### 학습 목표

이 단원에서 학생들은 입력 회로를 제어하는 간단한 프로그래밍을 작성할 수 있다

### 성공 기준

학생들은 회로에서 입력 값을 읽고 표시하기 위해 프로그래밍을 작성할 수 있다.

마이크로 컨트롤러는 핀을 통해 신호를 수신 할 수 있습니다. 마이크로 컨트롤러에 연결된 센서는 디지털 및 신호를 마이크로 컨트롤러로 보낼 수 있습니다.

디지털 신호의 값은 1 (켜기) 또는 0 (끄기)입니다. 모든 숫자 핀 (3-13)은 프로그램 명령 `digitalRead`를 사용하여 디지털 신호를 수신 할 수 있습니다. 디지털 신호를 사용하는 입력 장치의 예로는 스위치 (눌림 / 안눌림), 빛물감지 센서 (물 / 물 없음) 등 두 가지 상태가 있는 장치가 있습니다.

반면 아날로그 신호는 두 개 이상의 값을 포함하며 범위는 0 ~ 1023입니다. 아날로그 신호는 프로그램 명령 `analogRead`를 사용하여 아날로그 입력 핀 (핀 A0 ~ A5)에서만 유효합니다. 아날로그 출력 장치의 예는 포텐서미터 (노브 회전에 따른 값 변경), 광 센서 (매우 어둡거나 매우 밝음), 마이크 등과 같은 다양한 정보를 감지하는 센서입니다.

입력 장치를 사용하기 전에 Setup 기능에서 입력 핀을 INPUT 모드로 설정해야 합니다.

```
sketch_jan07a $
```

```
1 void setup() {
2   // put your setup code here
3   pinMode(4, INPUT);
4 }
5
```

## 단원 3.1

2번 핀에는 메이커 UNO 용 내장 스위치가 있습니다.  
핀을 사용하려면 모드를 `INPUT_PULLUP` 으로 설정해야 합니다

```
sketch_jan07a $
1 void setup() {
2   // put your setup code here,
3   pinMode(2, INPUT_PULLUP);
4 }
5
```

값을 사용하려면 `analogRead` 또는 `digitalRead`를 사용하여 값을 읽고 변수에 저장해야 합니다.

```
~
6 void loop() {
7   // put your main code here,
8   int x = digitalRead(2);
9   int y = analogRead(A2);
10 }
```

라인 8은 핀 2에서 디지털 신호를 읽고, 이를 변수 x에 유지합니다. 여기서 라인 9는 핀 A2에서 아날로그 신호를 읽고 y라는 변수에 저장합니다.

또한 컴퓨터와의 직렬 통신을 열어 디버깅 목적으로 읽은 값을 검색하고 표시 할 수 있습니다. 그렇게 하려면 컴퓨터와 Arduino 간의 통신 채널을 `Serial.begin(9600)` 코드로 시작해야 합니다.

```
1 void setup() {
2   // put your setup code
3   pinMode(A0, INPUT);
4   Serial.begin(9600);
5 }
```

9600은 Arduino와 컴퓨터 간의 일반적인 데이터 통신 전송 속도입니다.  
값이 높을수록 Arduino에서 컴퓨터로 더 많은 정보를 보낼 수 있습니다.

## 단원 3.1

그런 다음 `Serial.println`을 사용하여 입력 장치의 값을 표시 할 수 있습니다  
(=는 L의 소문자 입니다).

```
1 void setup() {  
2  
3   pinMode(A0, INPUT);  
4   Serial.begin(9600);  
5 }  
6  
7 void loop() {  
8  
9   int x = analogRead(A2);  
10  Serial.println(x);  
11 }
```

코드가 업로드되면 시리얼 모니터 (CTRL + SHIFT + M)를 열거나 돋보기 기호를 눌러 Arduino가 읽은 값을 볼 수 있습니다.



## 단원 3.1

다음 코드는 내장 스위치 (Maker UNO 용)에서 값을 읽고 값을 표시합니다. 이 예에서는 Maker UNO의 브레드 보드와 연결할 회로가 필요하지 않지만 다른 Arduino 모델의 경우 외부 푸시 버튼 연결이 필요합니다. 푸시 버튼을 pin 2에 연결하고 핀을 `pinMode (2, INPUT)`으로 설정해야 합니다.

```
sketch_jan07a$  
1 void setup() {  
2   // put your setup code here,  
3   pinMode(2, INPUT_PULLUP);  
4   Serial.begin(9600);  
5 }  
6  
7 void loop() {  
8   // put your main code here,  
9   int x = digitalRead(2);  
10  Serial.println(x);  
11 }  
12  
13
```

# 단원 3.2

## 입력장치

### 학습 목표

이 단원에서 학생들은 다양한 유형의 입력 장치를 배울 수 있다

### 성공 기준

학생들은 입력 장치의 이름과 기능을 나열하고 진술 할 수 있다.

입력 장치는 주변 환경에서 감지 된 값을 읽고 Arduino의 핀을 통해 읽은 정보를 보냅니다. 모든 핀을 프로그램 명령 digitalRead로 디지털 신호를 읽는 데 사용할 수 있습니다.

아날로그 신호는 핀 A0 ~ A5 인 아날로그 핀을 사용해서 만 감지 할 수 있습니다. 스위치와 같이 사용할 수 있는 다른 입력 장치가 있습니다.

포텐셔미터, 광 저항 저항기 (광 센서), 온도 센서, 인간 센서 (PIR 센서) 등

### 스위치

- 마이크로 스위치, 토글 스위치 및 푸시 버튼과 같은 다양한 유형의 스위치가 있습니다.
- 시중에는 여러 유형의 스위치가 있지만 모두 회로가 완전한지 여부와 같은 원리로 작동합니다.
- 다음은 가장 일반적으로 사용되는 스위치입니다.
- 푸시 버튼

▶ 스위치를 누르고있는 동안 "ON"상태가됩니다. 누르지 않으면 OFF 상태로 돌아갑니다.

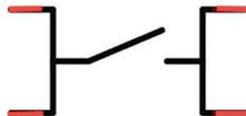


도표 3.1.(a): 푸시버튼

도표 3.1.(b): 푸시버튼의 회로도 기호

## 단원 3.2

### 스위치

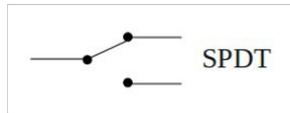
- 토글 스위치
  - ▶ STST 타입 -Single Pole, Single Throw



ON - OFF 동작만 가능

SPST에는 두 개의 터미널이 있으며 연결은 내부가 완전한 지 또는 불완전한 지를 결정합니다. SPST 연결은 보다 명확한 판독을 위해 풀업 / 풀다운 회로가 필요합니다. 노브의 위치에 따라 ON 또는 OFF 모드로 유지됩니다.

- ▶ STDT 타입 -Single Pole, Double Throw



ON - ON / ON - OFF 동작 모두 가능

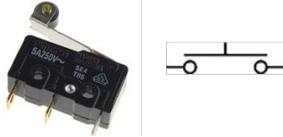
SPDT 스위치에는 3 개의 공통 단자, NO (일반적으로 열림) 및 NC (일반적으로 닫힘)가 있습니다.

SPDT 스위치에는 풀업 / 풀다운 회로가 필요하지 않습니다. 노브 위치에 따라 ON 또는 OFF 모드로 유지됩니다.

# 단원 3.2

## 스위치

- 마이크로 스위치



마이크로 스위치는 작은 움직임이나 가벼운 압력에 반응합니다. 예를 들어 냉장고의 전등 스위치입니다. 냉장고가 열리면 (마이크로 스위치에서 압력이 해제 됨) 표시등이 켜집니다.

## 포텐셔미터

포텐셔미터는 고정 저항 값이 없는 저항입니다. 노브를 돌려 저항 값을 변경할 수 있습니다.

이 저항 변화는 아날로그 입력 핀을 통해 Arduino가 읽을 수 있습니다.



도표 3.1.(c): 포텐셔미터

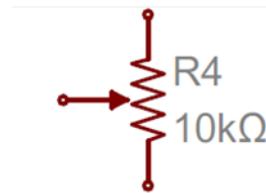


도표 3.1.(c): 포텐셔미터의 회로도 기호

## 단원 3.2

### 광 센서 / 광 증속 저항

- 저항 값은 주변 환경의 밝기에 따라 변합니다. 밝을수록 Arduino에서 읽을 수 있는 값이 높아집니다.

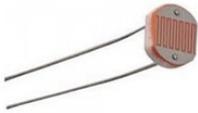


도표 3.1.(e): 광센서

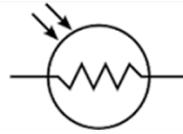


도표 3.1.(e): 광센서의 회로도 기호

### 온도센서

- 시장에는 다양한 유형의 온도 센서가 있으며 가장 일반적인 것은 LM35입니다.
- LM35는 전압 판독 형태로 입력을 생성합니다. 온도를 섭씨로 측정 하려면 이 입력을 추가 처리해야 합니다.



도표 3.1.(g): LM35

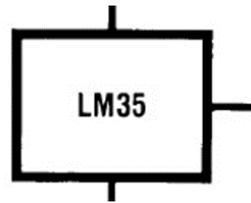


도표 3.1.(g): LM35의 직접회로 (IC) 의 회로도 기호

# 단원 3.2

## 온도센서

- 사람의 움직임은 사람이 발산하는 적외선을 감지하는 PIR 센서 (Passive Infrared)로 감지 할 수 있습니다.



도표 3.1.(e) : PIR 센서

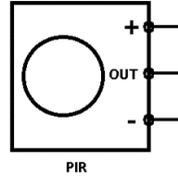


도표 3.1.(f) : PIR 센서의 회로도 기호

# 단원 3.3

## 입력 회로 연결

### 학습 목표

이 단원에서 학생들은 회로도를 읽고 회로도를 기반으로 회로를 연결할 수 있다.

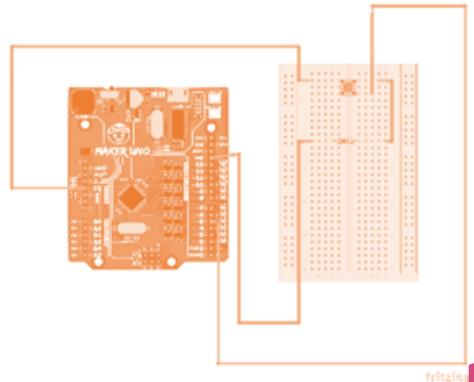
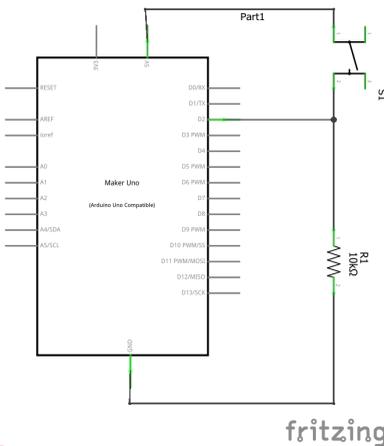
### 성공 기준

학생들은 회로도를 기반으로 하나 이상의 회로를 만들 수 있다

출력 회로와 마찬가지로 입력 회로 연결은 점퍼 와이어와 브레드 보드도 사용합니다.

### 푸시버튼

- Maker UNO에는 핀 2에 연결된 푸시 버튼이 있습니다. 이를 사용하려면 설정에서 pinMode를 INPUT\_PULLUP으로 설정하십시오.
- 푸시 버튼을 연결하는 회로도와 연결도는 다음과 같습니다.



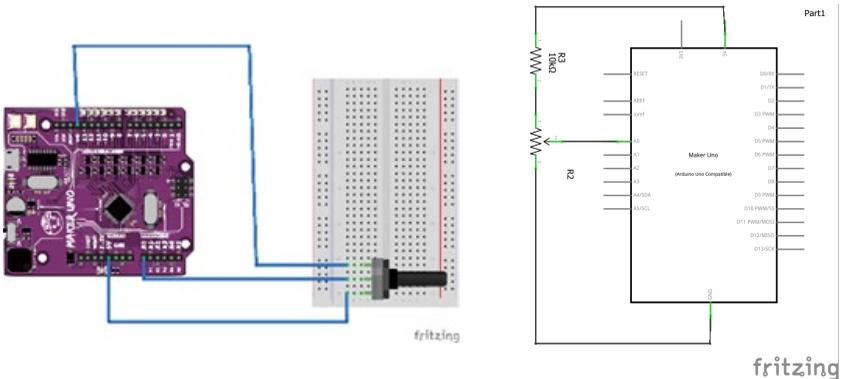
# 단원 3.3

## 푸시버튼

- 디지털 입력 장치이므로 **digitalRead**를 사용하여 값을 읽습니다.
- 풀다운 회로를 만들려면 10k 저항이 필요합니다. 풀다운 회로는 푸시 버튼을 누르지 않을 때 모든 부동 전하를 접지로 끌어내어 판독 된 값의 정확성을 보장합니다.

## 포텐서미터

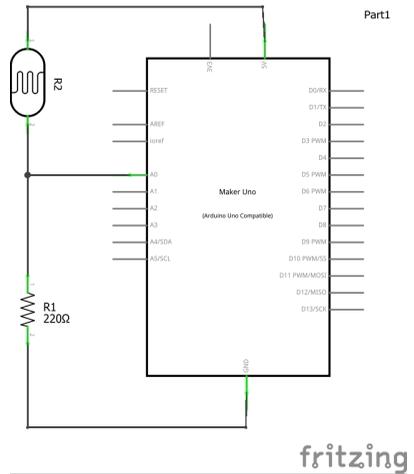
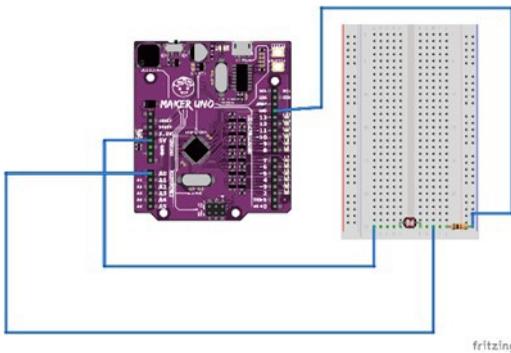
- 포텐서미터 연결은 저항이므로 저항이 없으므로 극성이 없습니다.
- 포텐서미터의 왼쪽과 오른쪽은 5V 또는 접지에 연결되며 중간 핀은 원하는 핀에 연결됩니다.
- 포텐서미터는 아날로그 입력 장치 유형입니다. 아날로그 장치의 경우 핀 A0 ~ A5와 프로그램 명령 **analogRead**를 사용하여 값을 읽어야 합니다.



# 단원 3.3

## 광센서 (LDR)

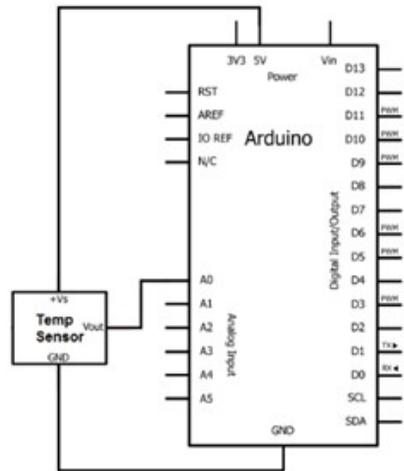
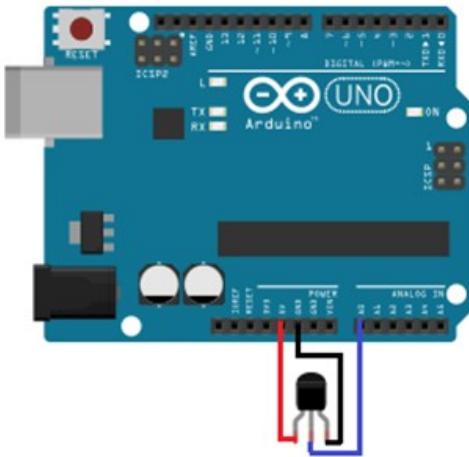
- 포텐셔미터와 마찬가지로 LDR (조명 저항)에는 극성이 없습니다.
- 푸시 버튼 연결과 유사하게 LDR은 깨끗한 판독을 위해 풀다운 회로가 필요합니다.
- LDR은 아날로그 입력 장치이므로 핀 AQ-A5를 사용하여 연결하고 프로그래밍 명령 `analogRead`를 사용하여 값을 읽습니다.



## 단원 3.3

### 온도센서

- 온도 센서 LM35 연결은 왼쪽 및 오른쪽 단자가 5V 및 접지에 연결된 포텐서 미터와 유사합니다. 중간 단자를 핀에 연결합니다.
- LM35는 아날로그 입력 장치입니다. 따라서 핀 A0 ~ A5와 프로그래밍 명령 `analogRead`를 사용하여 값을 읽습니다.



# 단원 3.4

## 입력 회로 시뮬레이션

### 학습 목표

이 단원에서 학생들은 전용 소프트웨어에서 입력 회로를 시뮬레이션 할 수 있다.

### 성공 기준

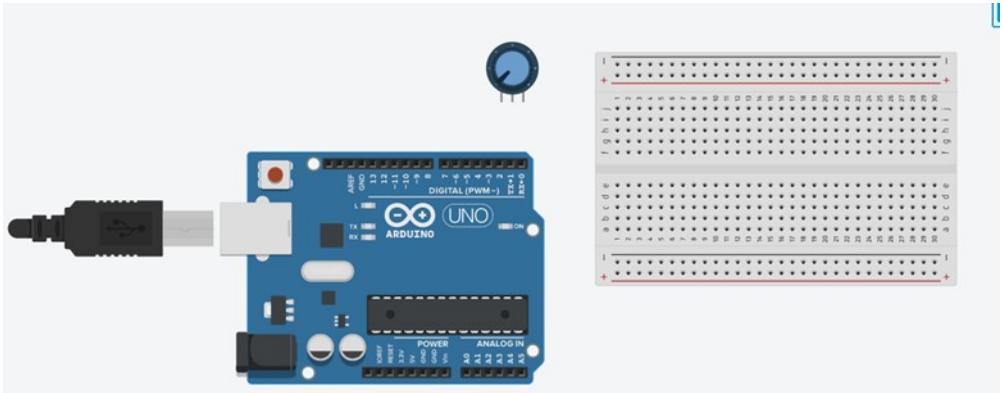
학생들은 프로그래밍을 통해 최소 하나의 회로 시뮬레이션을 시뮬레이션 할 수 있다.

무료 웹 애플리케이션 Tinkercad를 사용하여 입력 회로를 시뮬레이션 할 수 있습니다.

Tinkercad 웹 링크 : [www.tinkercad.com](http://www.tinkercad.com)

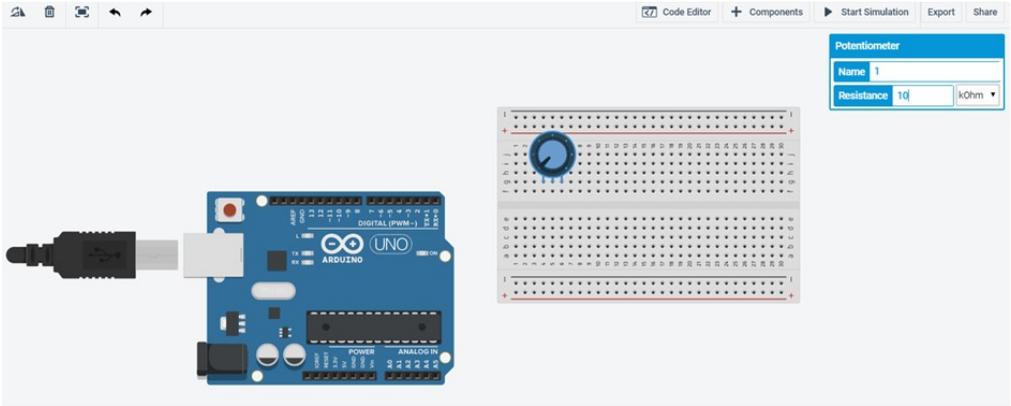
### 포텐셔미터 회로 시뮬레이션

아두이노 , 포텐셔미터(분압기) , 브레드보드 추가

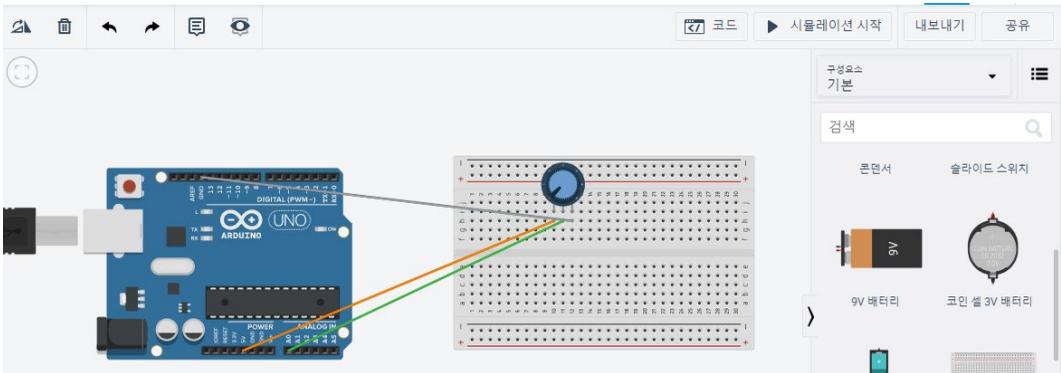


## 단원 3.4

포텐서미터 저항 값을 클릭하고 값을 10k 또는 100k로 변경하여 포텐서미터 저항 값이 올바르게 설정되어 있는지 확인하십시오

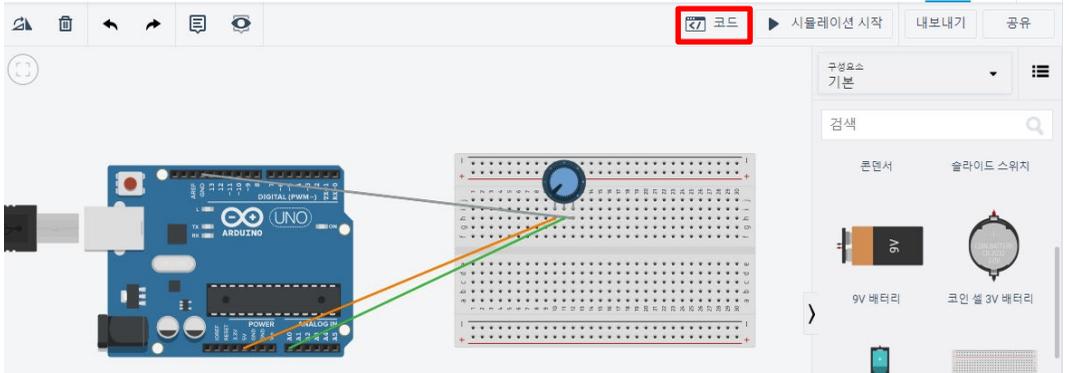


아두이노와 포텐서미터를 브레드보드 각각 연결



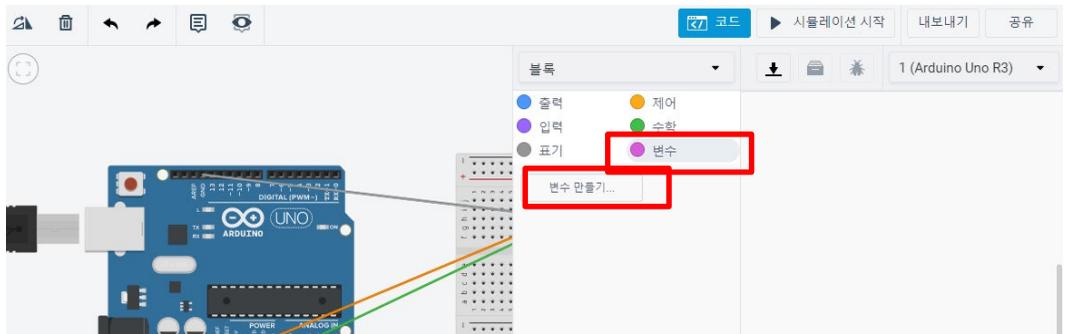
# 단원 3.4

화면 상단의 코드를 열어 프로그래밍을 시작



관련 블록을 끌어다 놓아 A0에 연결된 포텐서미터의 값을 읽고 표시하는 프로그래밍을 생성합니다.

새로운 변수 만들기



www.tinkercad.com 내용:

변수이름은 꼭 영문으로 할것

새 변수 이름:

확인

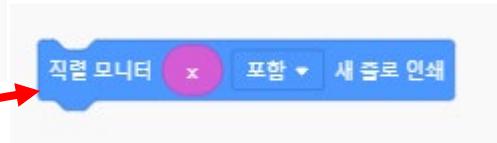
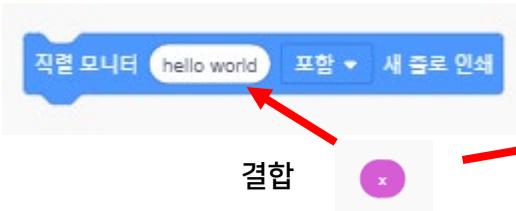
취소

# 단원 3.4

만든 변수를 입력란의  
아날로그 핀 읽기와 결합

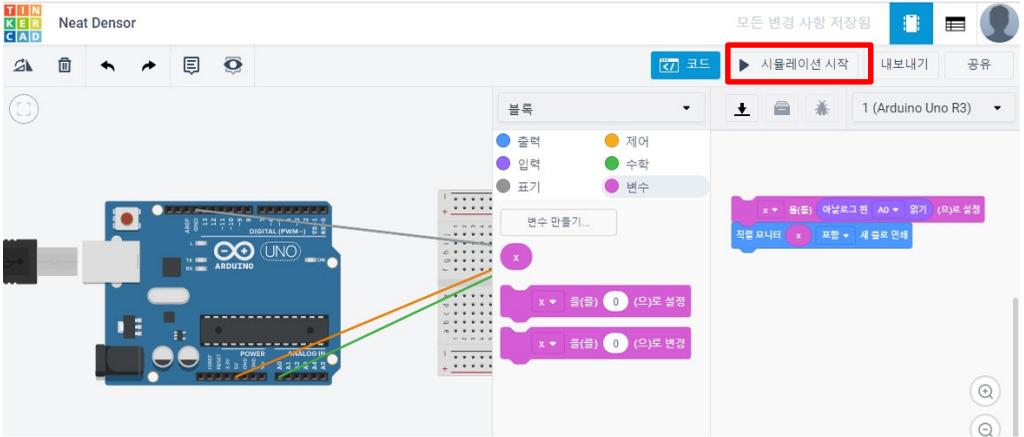


출력란에 직렬모니터 포함 새 줄로 인쇄 블록과 변수 이름 X를 결합

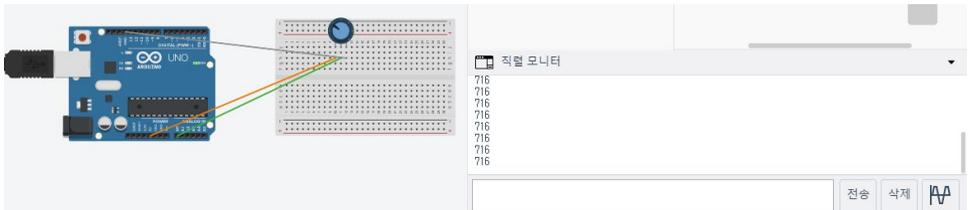


# 단원 3.4

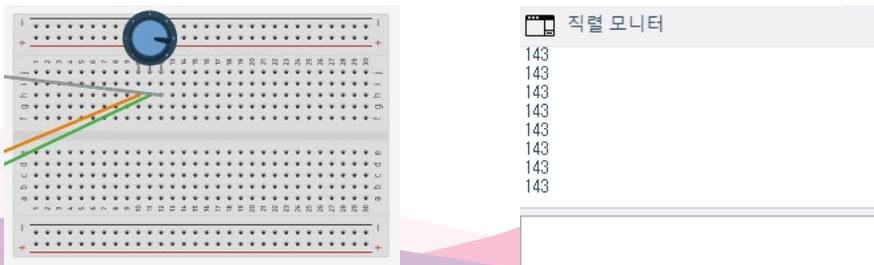
## 시뮬레이션 시작 누르기



블록 메뉴 쪽 제일 하단에 있는 **직렬 모니터** 눌러 포텐서미터 출력 값 확인하기



포텐서미터의 노브를 돌리면서 직렬모니터에 변하는 출력 값을 확인해보세요



# 단원 4

## 입력과 출력의 조합

### 학습표준

- 2.4.4 전용 소프트웨어로 기능적인 시뮬레이션 회로를 구축
- 2.4.5 마이크로 컨트롤러의 입력 및 출력 회로 연결
- 2.4.6 입력 및 출력 회로를 기반으로 간단한 프로그램 작성
- 2.4.7 전자 회로의 기능 시험 및 평가
- 2.4.8 전자 회로의 개선을 권장

### 평가표준

- 4 단계 마이크로 컨트롤러가 포함 된 회로의 기능을 테스트
- 5 단계 문제를 해결하기 위해 입력 및 출력을 위한 프로그래밍 제어 구조를 정당화
- 6 단계 작동하는 마이크로 컨트롤러 회로를 구성

### 부 단원

- 단원 4.1 Arduino의 조건부 프로그래밍 구조 소개
- 단원 4.2 결합 입력 및 출력 회로
- 단원 4.3 입력 및 출력 회로 조합 시뮬레이션
- 단원 4.4 프로젝트 제작

### 권장 수업 시간

단원 3.1	60분
단원 3.2	30분
단원 3.3	
단원 3.4	90분

### 준비

- 1) 각 학생에게 충분한 Arduino / Maker UNO가 있는지 확인하십시오.
- 2) Unit 4.1에서는 컴퓨터 또는 스마트 폰을 사용하여 마이크로 컨트롤러를 프로그래밍 해야 합니다. (부록 1 참조)
- 3) Unit 4.3에는 컴퓨터와 인터넷이 필요합니다
- 4) 학생 모듈과 함께 사용 가능

# 단원 4.1

## 조건부 프로그래밍 구조 소개

### 학습 목표

이 단원에서 학생은 수신 된 입력에 따라 다른 출력을 생성하는 프로그래밍을 작성할 수 있다.

### 성공 기준

푸시 버튼 (내장 또는 외부 연결)을 누르면 학생이 내장 LED를 켤 수 있다.

입력과 출력을 결합하여 환경과의 상호 작용에 따라 반응하는 프로젝트를 만들 수 있습니다.

마이크로 컨트롤러는 수신 된 입력 (정보)에 따라 다른 출력 (응답)을 공급하는 두 뇌 역할을 합니다.

입력 장치를 통해 정보를 받아 원하는 값과 비교할 수 있습니다. 읽은 값이 원하는 입력과 일치하면 의도 된 출력을 생성하도록 마이크로 컨트롤러를 프로그래밍 할 수 있습니다. 이를 조건부 프로그래밍 구조라고 합니다. 조건부 프로그래밍 구조는 다음과 같이 순서도에서 시각화 할 수 있습니다.

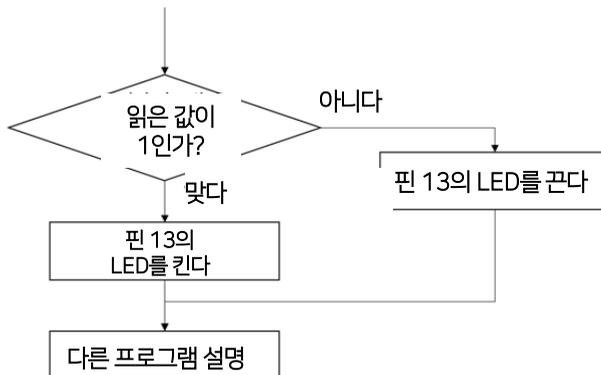


도표 4.1.(a) : 조건부 프로그래밍 구조에 대한 순서도.

# 단원 4.1

도표 4.1 (a)에 표시된 순서도 프로그램은 다음과 같습니다.

```

6 void loop() {
7   // put your main code here,
8   int x = digitalRead(2);
9   if (x == 1){
10    digitalWrite(13, HIGH);
11  } else {
12    digitalWrite(13, LOW);
13  }
14 }

```

비교 연산자는 서로 다른 두 값을 비교하는 데 사용되며 비교가 유효한 경우 부울(Boolean) 결과 True를 생성합니다. Arduino에는 6 개의 비교 연산자가 있습니다

비교 연산자	의미	예시
==	같음	<pre> x = digitalRead(5); if (x == 1){ digitalWrite(13, HIGH); }else{ digitalWrite(13, LOW); } </pre> <p>x 값을 읽으십시오(핀 5에 연결된 푸시 버튼). x가 1인 경우(누름 버튼을 누름) 높은 디지털 신호를 핀 13으로 전송하여 LED를 켜십시오. 또는 핀 13에 낮은 디지털 신호를 보내 LED를 끄십시오.</p>

# 단원 4.1

비교 연산자	의미	예시
!=	같지 않음	<pre>x = digitalRead(5); if (x != 1){ digitalWrite(13, HIGH); }else{ digitalWrite(13, LOW); }</pre> <p>값 x를 읽습니다 (핀 5에 연결된 푸시 버튼). x가 1이 아닌 경우 (푸시 버튼을 누르지 않음), 높은 디지털 신호를 핀 13으로 보내 LED를 켜줍니다. 또는 낮은 디지털 신호를 핀 13으로 보내 LED를 끕니다.</p>
>=	보다 크거나 같음	<pre>x = analogRead(5); if (x&gt;=100){ digitalWrite(13, HIGH); }else{ digitalWrite(13, LOW); }</pre> <p>x (핀 A0에 연결된 포텐서미터)를 읽습니다. 읽은 값이 100 이상인 경우 핀 13에서 LED를 켜려면 HIGH 디지털 신호를 보내십시오. 그렇지 않으면 핀 13에서 LED를 끄려면 LOW 디지털 신호를 보내십시오.</p>

# 단원 4.1

비교 연산자	의미	예시
>	보다 크다	<pre>x = analogRead(5); if (x &gt; 100){ digitalWrite(13, HIGH); }else{ digitalWrite(13, LOW); }</pre> <p>x (핀 A0에 연결된 포텐셔미터)를 읽습니다. 읽은 값이 100보다 크면 (100 제외) HIGH 디지털 신호를 보내 핀 13에서 LED를 켭니다. 또는 LOW 디지털 신호를 보내 핀 13에서 LED를 끕니다.</p>
<=	보다 작거나 같음	<pre>x = analogRead(5); if (x &lt;= 100){ digitalWrite(13, HIGH); }else{ digitalWrite(13, LOW); }</pre> <p>x (핀 A0에 연결된 포텐셔미터)를 읽습니다. 읽은 값이 100보다 작거나 100보다 작으면 HIGH 디지털 신호를 보내 핀 13에서 LED를 켭니다. 또는 LOW 디지털 신호를 보내 핀 13에서 LED를 끕니다.</p>

## 단원 4.1

비교 연산자	의미	예시
<	보다 작음	<pre>x = analogRead(A0); if (x &lt; 100){     digitalWrite(13, HIGH); } else {     digitalWrite(13, LOW); }</pre> <p>x (핀 A0에 연결된 포텐셔미터)를 읽습니다. 읽은 값이 100보다 작으면 (100 제외) 핀 13에서 LED를 켜려면 HIGH 디지털 신호를 보내십시오. 그렇지 않으면 핀 13에서 LED를 끄려면 LOW 디지털 신호를 보내십시오.</p>

두 한계 사이의 값을 확인하려면 논리 연산자 AND를 사용해야 합니다.

예를 들어 500 ~ 700 (501-> 699) 사이의 값은 다음과 같이 감지 할 수 있습니다.  
코드 :

```
6 void loop() {
7     // put your main code here,
8     int x = analogRead(A2);
9     if (x > 500 && x < 700) {
10        digitalWrite(13, HIGH);
11    } else {
12        digitalWrite(13, LOW);
13    }
14 }
```

(AND 기호는 &&)

## 단원 4.1

예 1 : (Maker UNO에만 해당)

다음 코드는 핀 2 (Maker UNO 용 내장)에 연결된 푸시 버튼을 읽고 해당 값을 표시한 다음 푸시 버튼을 누르면 핀 13에서 LED를 켭니다. 그렇지 않으면 LED를 끕니다.

(다른 Arduino의 경우 푸시 버튼을 핀 2에 연결하고 핀 2의 핀 모드를 INPUT\_PULLUP에서 INPUT으로 변경 해야합니다)

```
sketch_jan07a $  
1 void setup() {  
2  
3   pinMode(2, INPUT_PULLUP);  
4   pinMode(13, OUTPUT);  
5   Serial.begin(9600);  
6 }  
7  
8 void loop() {  
9  
10  int x = digitalRead(2);  
11  Serial.println(x);  
12  if (x == 1){  
13    digitalWrite(13, HIGH);  
14  } else {  
15    digitalWrite(13, LOW);  
16  }  
17 }
```

## 단원 4.1

예 2 : (핀 A0에 LDR을 연결해야 함)

이 프로그램은 핀 A0에 연결된 LDR (광센서)의 값을 읽습니다.

주변 광을 기준으로 2 가지 출력을 생성합니다. 주변 환경이 어두운 경우 (값이 400이거나 400보다 작으면) 핀 13의 LED가 켜집니다. 주변이라면 환경이 밝고 (값 읽기가 400 이상임) 13 번 핀의 LED가 꺼집니다.

```
sketch_jan07a $  
1 void setup() {  
2  
3   pinMode(A0, INPUT);  
4   pinMode(13, OUTPUT);  
5   Serial.begin(9600);  
6 }  
7  
8 void loop() {  
9  
10  int x = analogRead(A0);  
11  Serial.println(x);  
12  if(x <= 400){  
13    digitalWrite(13, HIGH);  
14  } else {  
15    digitalWrite(13, LOW);  
16  }  
17 }
```

## 단원 4.2

# 입력 및 출력 회로의 조합

### 학습 목표

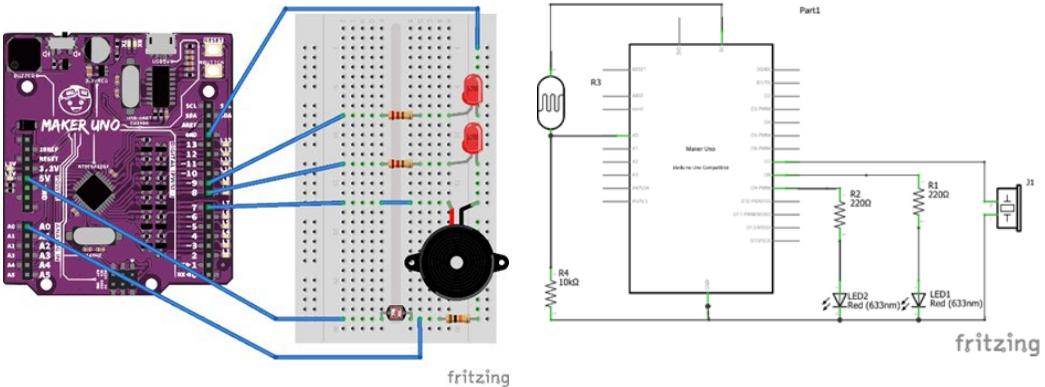
이 단원에서는 학생이 회로도를 읽고 회로도를 기반으로 회로를 연결할 수 있습니다.

### 성공 기준

학생들은 주어진 회로도에 따라 최소 1 개의 회로를 만들 수 있습니다

입력과 출력의 결합 회로는 별도로 연결하는 것과 크게 다르지 않습니다.  
다음은 입력 및 출력 장치의 조합을 포함하는 회로도의 몇 가지 예입니다.

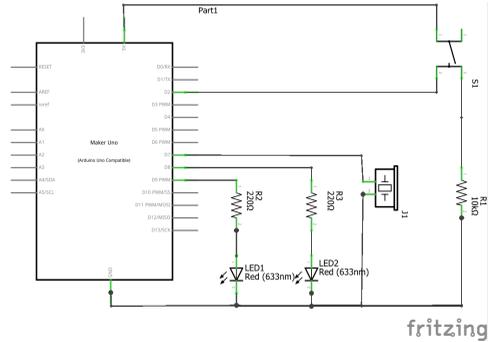
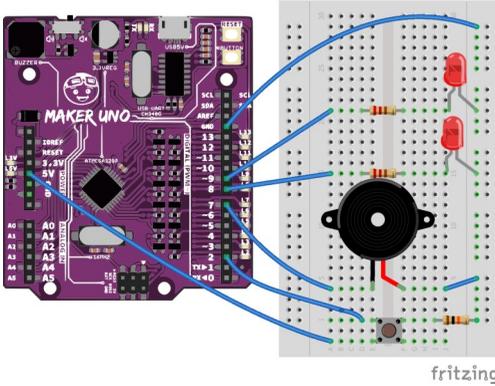
## 광센서, 부저, LED 회로



이 회로를 사용하면 간단한 뮤직 박스를 만들 수 있습니다.  
뮤직 박스가 열리면(밝기 증가) 마이크로 컨트롤러가 음악을 재생하고 LED를 켵니다.

# 단원 4.2

## 광센서, 부저, LED 회로



이 회로를 사용하면 푸시 버튼으로 활성화되는 음악을 만들 수 있습니다. 상자가 열리면 (푸시 버튼을 더 이상 누르지 않음) 마이크로 컨트롤러가 음악을 재생하고 LED를 켭니다. 또는 버튼을 눌러 뮤직 박스를 켤 프로그램을 작성할 수도 있습니다.

## 단원 4.3

# 입력 및 출력 시뮬레이션

### 학습 목표

이 단원에서 학생은 전용 소프트웨어에서 입력 및 출력 회로를 시뮬레이션 할 수 있다.

### 성공 기준

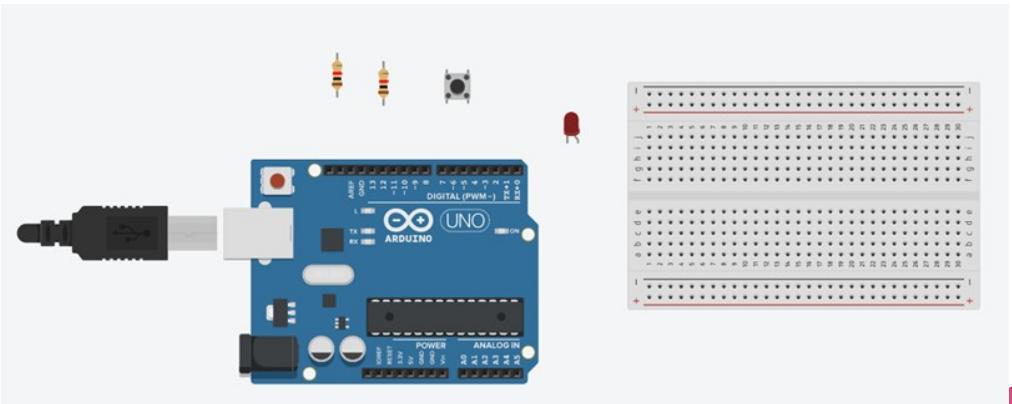
학생들은 프로그래밍으로 최소 1 개의 회로를 시뮬레이션 할 수 있다.

우리는 프리웨어인 Tinkerlab을 사용하여 출력 회로를 시뮬레이션할 것입니다.

웹링크 : <https://www.tinkercad.com/>입니다.

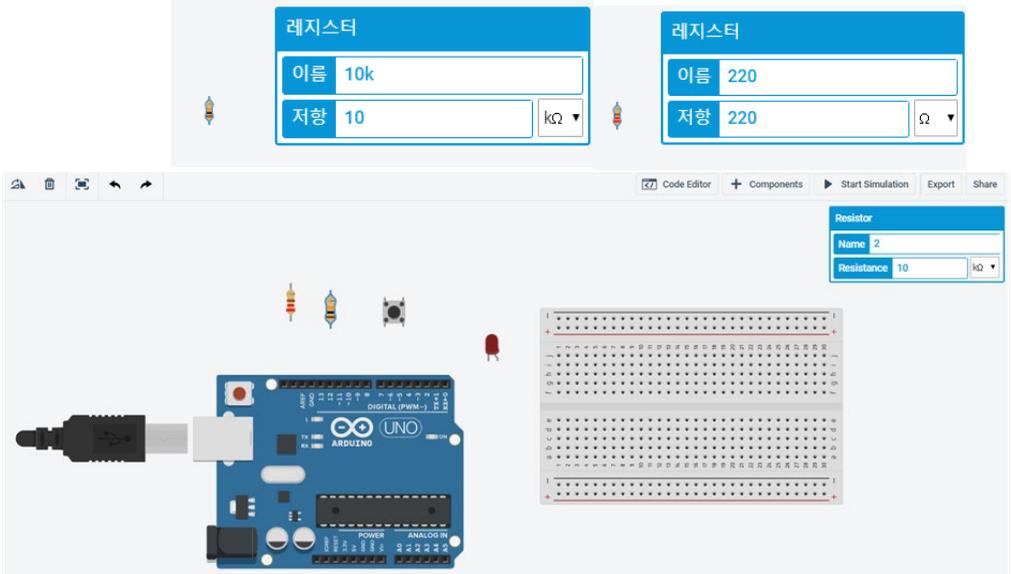
**푸시 버튼과 LED를 포함하는 회로. 푸시 버튼을 누르면 LED가 켜집니다.**

아두이노와 , 푸시버튼 , LED , 저항을 추가 합니다.

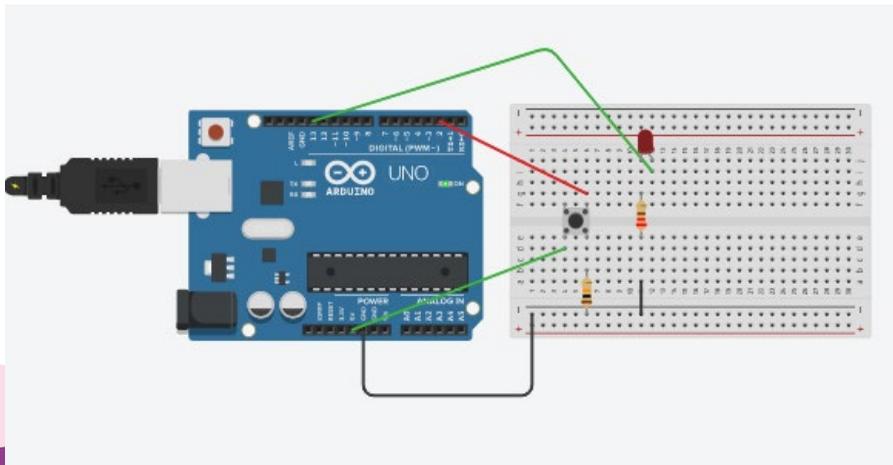


# 단원 4.3

저항을 클릭하고 값을 220 Ohm 및 10k Ohm으로 설정하여 저항의 저항 값이 올바른지 확인하십시오

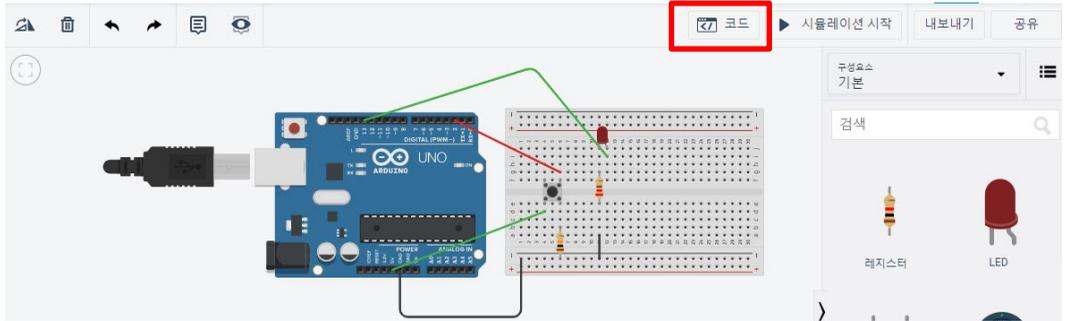


해당 요소를 브레드보드와 아누이노에 연결 합니다.



# 단원 4.3

화면 상단의 코드 버튼을 눌러 코딩을 진행 해봅시다.



새 변수 이름:

변수 버튼을 눌러 새로운 변수 값을 만듭니다.

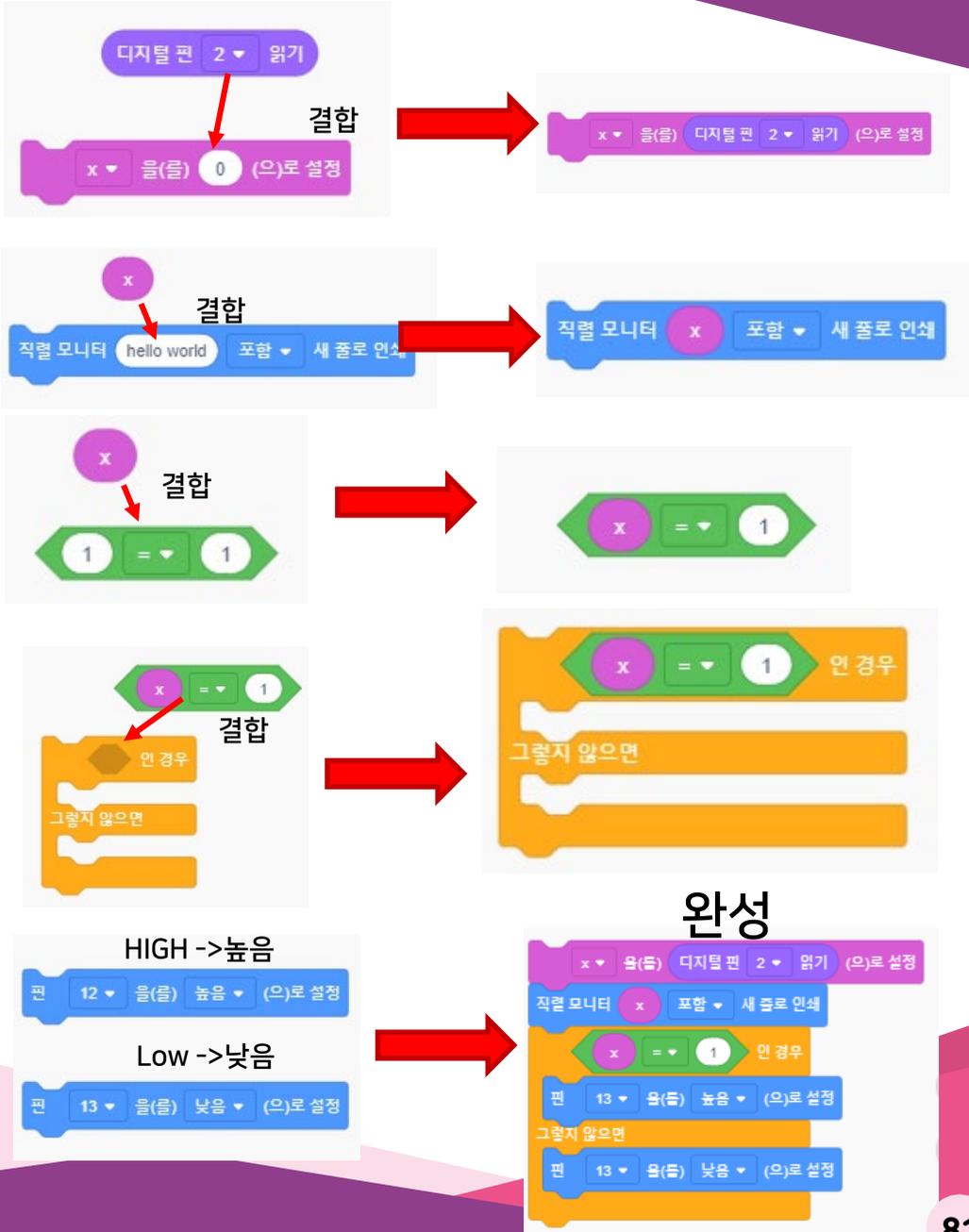
변수 이름은 꼭 영문이나 숫자만 가능합니다.

확인

취소

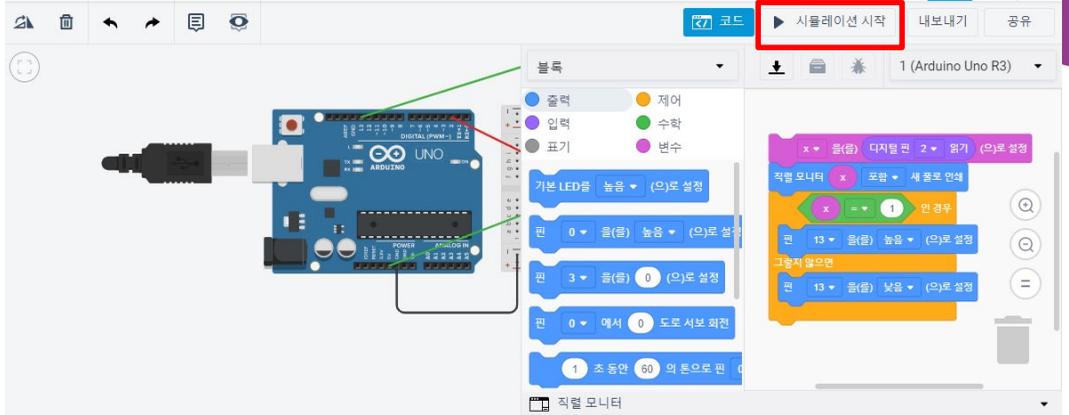


# 단원 4.3



# 단원 4.3

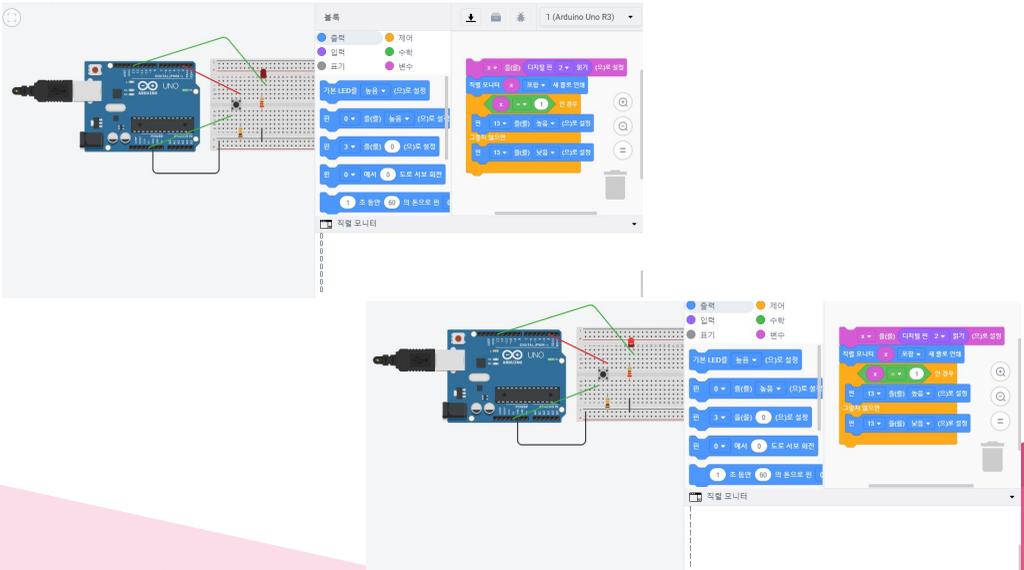
시뮬레이션 시작 버튼을 누릅니다



창의 제일 하단에 직렬 모니터 버튼을 누릅니다



스위치를 눌러 LED가 켜질 때 직렬모니터의 값이 1이 출력 되는지 확인 해봅시다



# 단원 4.3

## 프로젝트 창작

### 학습 목표

이 단원에서 학생은 입력 및 출력을 사용하는 간단한 프로젝트를 만들 수 있습니다.

### 성공 기준

학생은 입력 및 출력을 사용하는 프로젝트를 작성하고 프로젝트 작성을 시뮬레이션 및 문서화 할 수 있습니다.

