



누구나 쉽게 개발할 수 있는
5W1H 아두이노 개발도구
육하원칙
실습 매뉴얼

김지성 지음

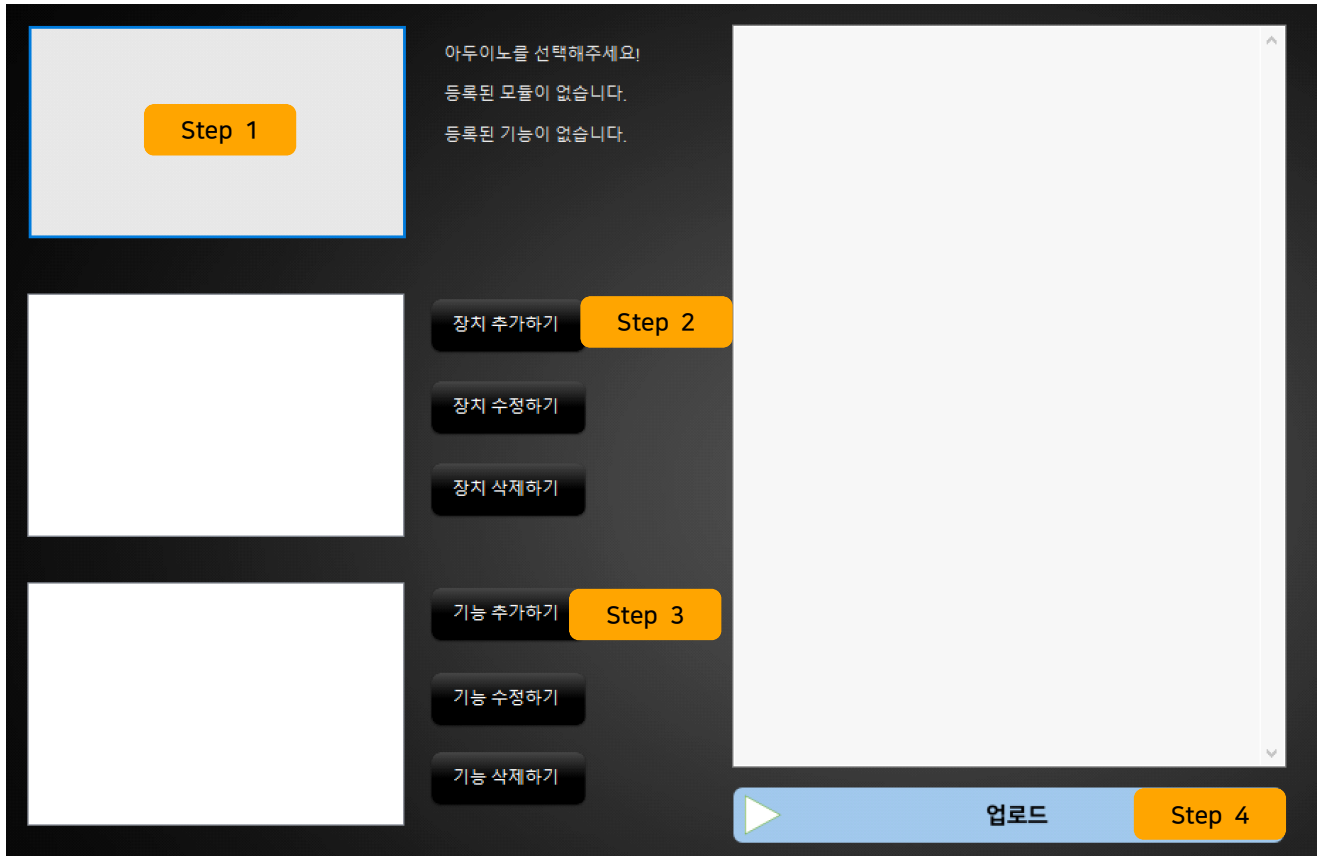
LED를 제어하는 기초부터 음성인식 스마트홈까지 내손으로 만들어보자!

Arduino IoT Studio 사용법	2
브레드보드 사용법	6
푸시버튼 사용법	7
LED 사용법	9
LED 켜보기	11
LED 깜빡이기	15
Button 3번 클릭 시 LED 켜기	20
버튼을 활용하여 LED 제어하기	27
2개의 버튼을 활용하여 2개의 LED 제어하기	34
Servo Motor 제어하기	42
서보모터 버튼으로 제어하기	47
블루투스 통신으로 LED켜기	52
블루투스 통신으로 서보모터 제어하기	61
현재 온/습도 출력하기	70
압력값 출력하기	74
FSR 센싱하기	78
FSR 센서로 앉은 자세 기울기 감지하기	83
조도센서를 이용하여 LED 제어하기	91
LCD “Hello World” 출력하기	96
7세그먼트 제어하기	98
스마트홈 개발툴을 이용해 블루투스 LED, Servo Motor 제어하기	104
음성인식 IoT 스마트홈 Studio 사용법	115
음성인식 스마트홈 만들기	116

Arduino IoT Studio 사용법

1

AI Studio 메인화면



위 그림은 AI Studio의 메인화면입니다.

AI Studio를 통한 개발은 크게 4단계의 과정으로 이루어져있습니다.

첫 번째 단계로 Step1의 버튼을 눌러 프로젝트에서 사용할 아두이노 보드를 선택합니다.

두 번째 단계로 Step2의 버튼을 눌러 프로젝트에서 사용할 모듈을 선택합니다. 여기서 모듈은 LED나 버튼 등의 다룰 수 있는 장치를 의미합니다.

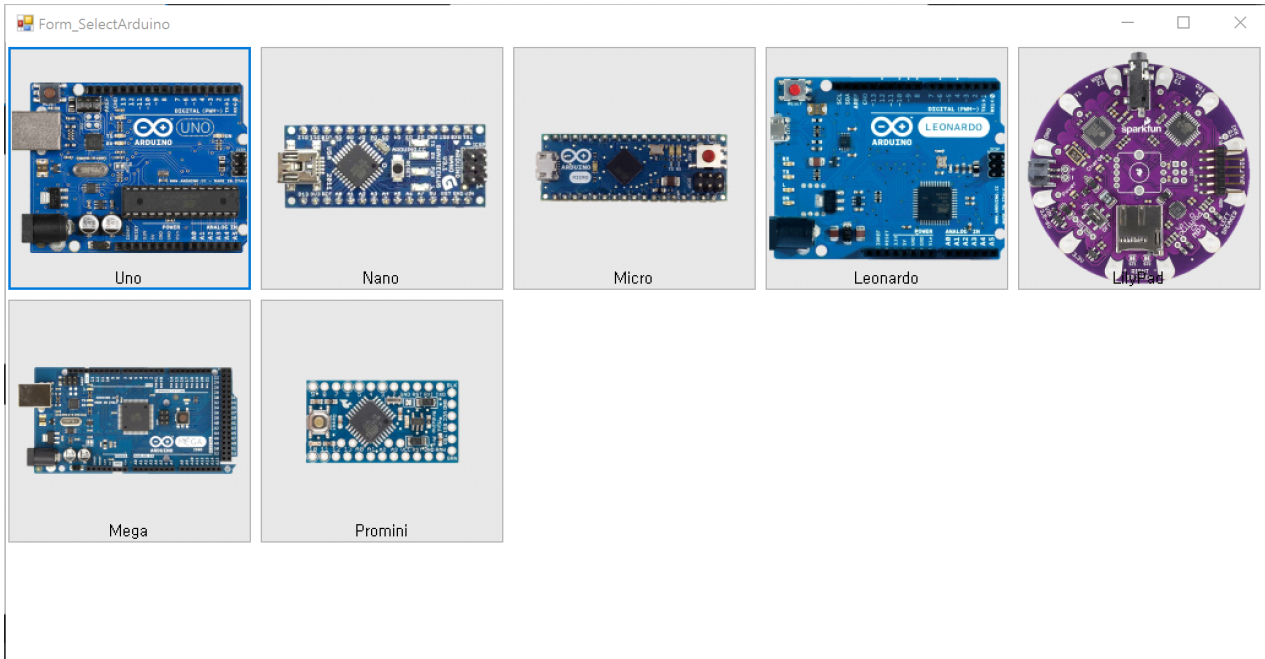
세 번째 단계로 추가한 모듈을 어떻게 다룰 것인지 설정합니다.

네 번째 단계로 만들어진 코드를 선택한 아두이노에 업로드 합니다.

네 번째 단계까지 마쳤다면 아두이노는 여러분이 설정한 방식으로 동작할 것 입니다.

2

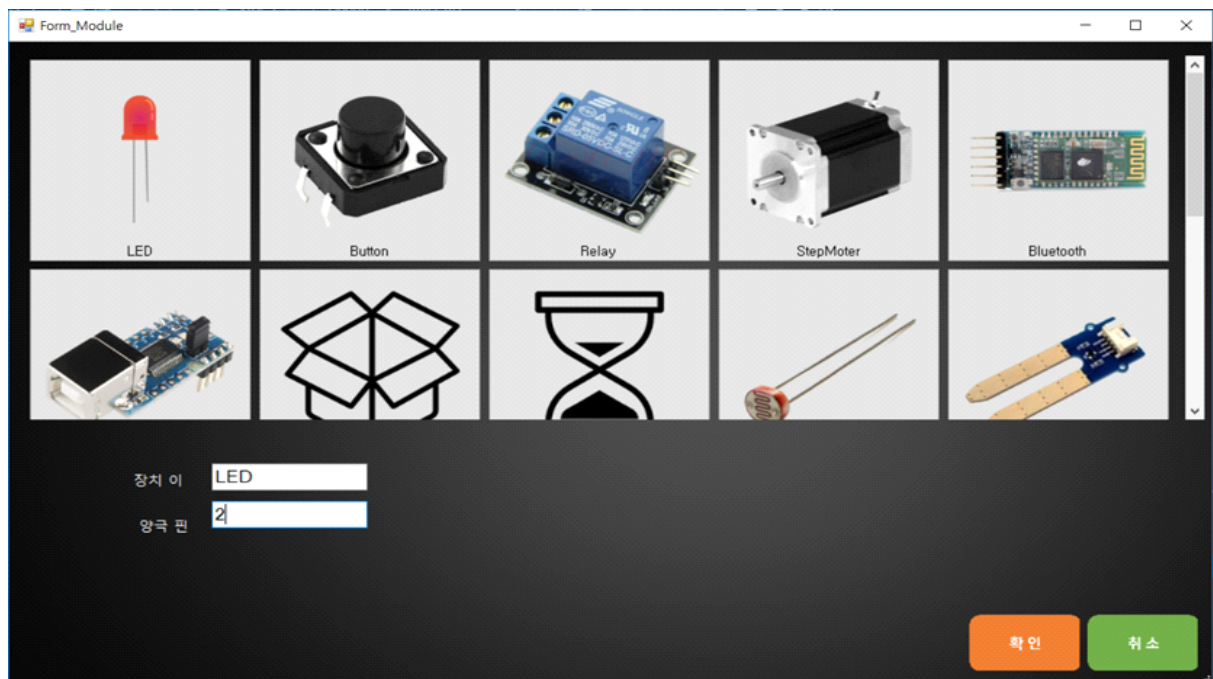
아두이노 선택하기



Step1의 버튼을 눌러 이 프로젝트에서 사용할 아두이노 보드를 선택합니다.
 학습용으로 가장 많이 사용되는 아두이노는 Uno와 Nano입니다.

3

사용할 모듈 등록하기



Step2의 버튼을 눌러 이 프로젝트에서 사용할 모듈을 등록합니다.
 여러 개의 모듈이 등록될 수 있으며, 등록 시 선택한 모듈의 이름과 핀 그리고 필요하다면 추가적인 속성을 입력하게 됩니다.
 LED를 사용하고 싶은 경우 모듈 선택 창에서 LED를 선택한 후 장치이름을 입력하고 아두이노에 모듈을 몇번 핀에 연결할 것인지 핀 번호를 입력한 후 확인 버튼을 누릅니다.

The screenshot shows a window titled 'Form_Function' with a dark background. At the top, there is a text input field containing 'LED켜기'. Below it is a dropdown menu labeled '모듈이' (Module is). Underneath that is another dropdown menu labeled '일 때' (When). At the bottom, there is a third dropdown menu labeled '모듈을' (Module) and a final dropdown menu labeled '한다.' (Does). At the very bottom are two buttons: an orange '확인' (Confirm) button and a green '취소' (Cancel) button. Five blue callout boxes with arrows point to the following elements: 1. The 'LED켜기' text box: '아두이노가 수행할 기능의 이름을 지정해 준다.' (Specify the name of the function to be performed by the Arduino). 2. The '모듈이' dropdown: ''누가' 모듈을 선택한다.' (Select the 'Who' module). 3. The '일 때' dropdown: ''언제'에 대한 조건을 정의한다.' (Define the condition for 'When'). 4. The '모듈을' dropdown: ''무엇을' 모듈을 선택한다.' (Select the 'What' module). 5. The '한다.' dropdown: ''어떻게' 에 대한 행동을 정의한다.' (Define the action for 'How').

첫 번째로 기능에 대한 이름을 정의해줍니다. 예를 들면 버튼 LED켜기 와 같이 정의합니다.

두 번째로 '누가' 에 대한 모듈을 선택합니다. 어떠한 모듈에 대해 반응하고 싶은 경우 조사할 모듈을 지정합니다. 예를 들면 '버튼'의 상태를 조사하고 싶은 경우 '버튼'을 선택합니다.

세 번째로 '언제'에 대한 조건을 선택합니다. 예를 들면 "버튼이 눌렸을때"를 조사하고 싶다면 '눌렸을때(HIGH)'를 선택합니다.

네 번째로 '언제'에 대한 부가적인 조건을 선택합니다. 부가적인 조건은 '같을 때', '이상일 때', '이하일 때', '초과일 때', '미만일 때' 중에서 선택할 수 있습니다.

다섯 번째로 '~일 때', 또는 '~이고', '~이거나'를 선택합니다. '~일 때'를 선택한다면 지금까지

설정된 조건이 만족할 때 명령을 실행하게 됩니다. 만약 '~이고', '~이거나'를 선택한다면 여러 개의 조건을 동시에 사용할 수 있는데, 이때 '~이고'를 선택했다면 모든 조건을 만족해야 명령을 실행하고, '~이거나'를 선택한다면 여러 개의 조건 중 하나이상 만족했을 때 명령을 실행합니다.

여섯 번째로 '무엇을'에 대한 모듈을 선택합니다. 앞서 설정한 조건을 만족했을 때 제어할 모듈을 선택합니다. 예를 들면 LED를 켜고싶은 경우 'LED'를 선택합니다.

일곱 번째로 '무엇을'에서 설정한 모듈을 '어떻게' 할 것인지를 선택합니다. 예를 들면 LED를 켜고 싶다면 '켜기(HIGH)'를 선택합니다.

여덟 번째로 '무엇을'에 대한 부가조건을 선택합니다. 부가적인 조건은 '설정한다', '더한다', '뺀다', '곱한다', '나눈다' 중에서 선택할 수 있습니다.

일곱 번째로 '~한다' 또는 '~하고'를 선택합니다. '~한다'를 선택했을 때 지금까지 선택한 명령을 마칩니다. '~하고'를 선택했을 때 여러 개의 명령을 동시에 실행할 수 있습니다. 예를 들면 LED1과 LED2를 '~하고'를 선택하여 HIGH로 설정한다면 두 LED를 동시에 불을 켤 수 있습니다.

5 번역 결과 확인하기 및 업로드

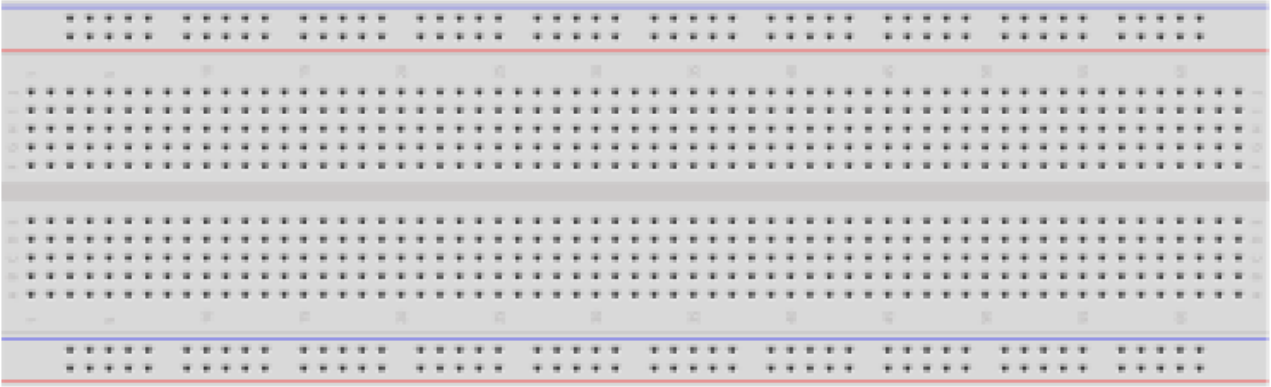


우측에서 볼 수 있는 최종 결과를 확인하고 업로드 버튼을 눌러 선택한 아두이노에 코드를 업로드 합니다. 이후 아두이노는 사용자가 설정한 행동대로 움직일 것 입니다.

브레드보드 사용법

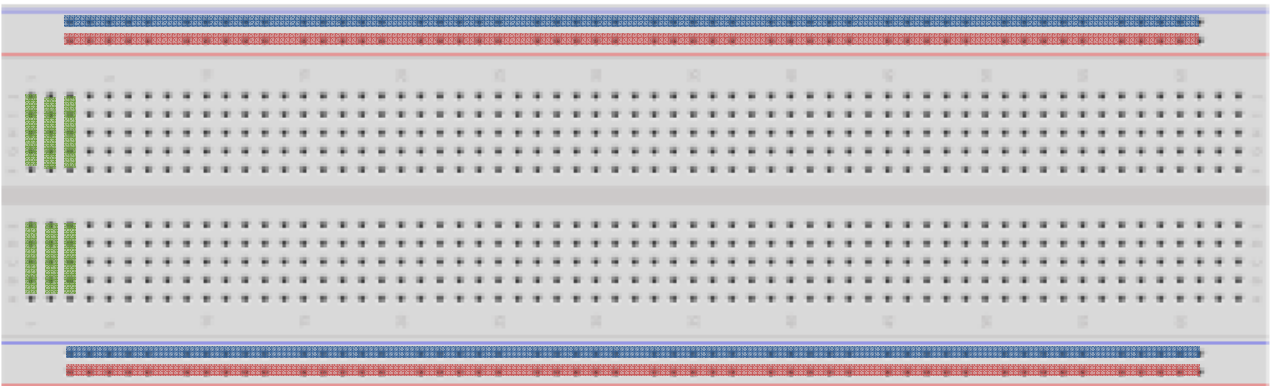
1

브레드보드의 외형



2

브레드보드의 사용법



브레드보드를 처음 사용하는 사람이라면 저 브레드보드의 구멍들이 모두 같아 보일 겁니다. 하지만 브레드보드에 내부에 금속이 있는데 각기 다르게 세로방향 가로방향으로 배치 되어있기 때문에 사용 방법을 따라야 합니다. 일단 브레드 보드의 내부 연결 구조부터 살펴봅시다.

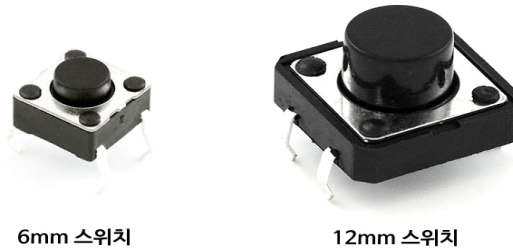
사진의 브레드보드 위쪽을 보시면 파란색 선과 빨간색 선이 지나가는 부분이 있습니다. 그 부분을 (버스)부분이라고 부르고 빨간색은 + 파란색은 - 전원 선을 연결합니다.

중간에 보면 세로로 5칸씩 구멍이 있는데 이 부분을 (IC영역/부품영역)이라고 하고 부품을 설치하는 영역 입니다. 사진을 보면 부품영역에 위 아래로 5칸씩 되어있습니다. 이것은 위아래 서로 5칸씩 격리 되어있고 위에 5칸과 아래 5칸은 서로 전기가 통하지 않습니다.

이미지에 빨간색, 파란색, 초록색으로 표시해 두었습니다. 이것은 브레드보드 내부에 금속으로 연결되어있는 선입니다.

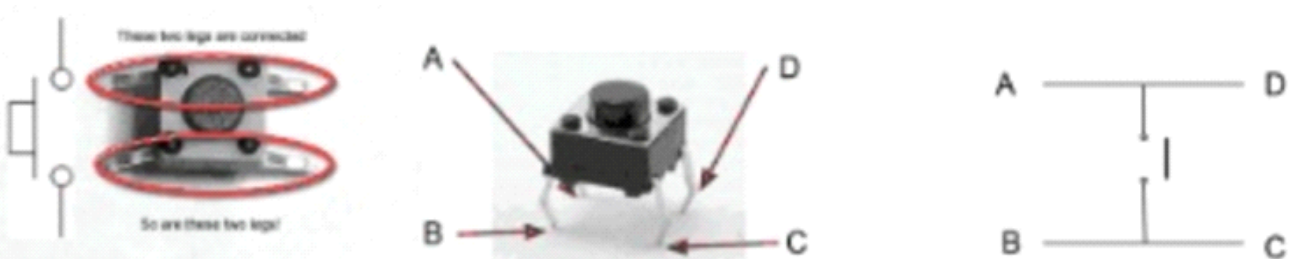
푸시버튼 사용법

1 푸시버튼의 외형



푸시버튼은 회로 구성에 있어서 기본이 되는 전자부품으로 모터와 같은 장치들을 온/오프하는 용도로 많이 사용됩니다. 푸시 버튼은 실생활에서도 계산기, 자동차, 게임기 등 어디서든 볼 수 있는데, 그 이유는 누르는 동작만으로 센서를 키거나, 특정 값을 전달 할 수 있기 때문입니다.

2 푸시버튼의 구조



푸시버튼의 구조는 그림과 같이 생겼습니다. A와 D, B와 C가 내부적으로 연결되어 있고, 버튼을 누르면 A와 B 또는 A와 C, D와 B 또는 D와 C가 연결됩니다.

3 디지털 회로의 두가지 방식

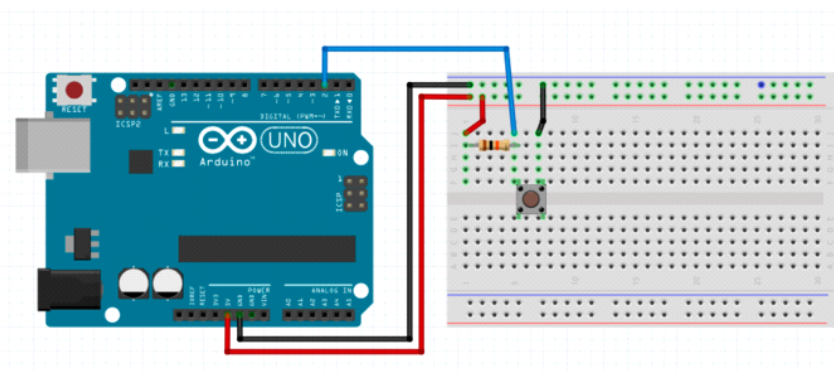


그림 25 풀업 방식 연결 회로

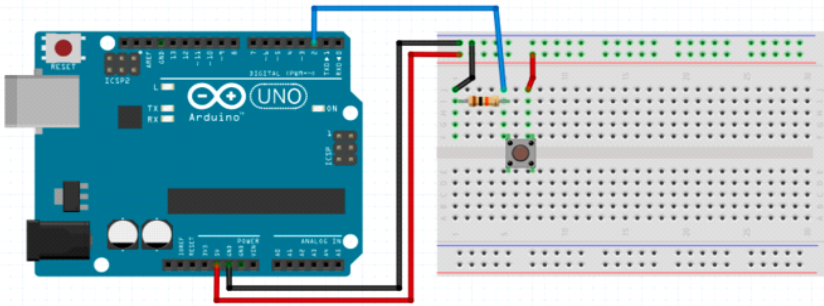


그림 26 풀다운 방식 연결 회로

푸시버튼을 연결하는 회로에는 풀업 방식과 풀다운 방식이 있습니다. 먼저, 풀업/풀다운 방식을 설명하기에 앞서 플로팅 상태에 대해 알아야 합니다.

컴퓨터는 0과 1로 전기적 신호를 구분합니다. 0은 LOW, 1은 HIGH라고도 합니다. 아두이노 보드에 연결된 버튼이 눌리지 않으면 5V나 0V가 아닌 어중간한 값이 전해지면서 1인지 0인지 판단할 수 없는 상태가 됩니다. 이를 플로팅 상태라고 하는데, 마치 0(LOW)와 1(HIGH) 사이를 부유하는 모습과 같아서 지어진 명칭입니다. 이 문제를 해결하는 방식이 바로 풀업, 풀다운입니다.

일단 풀업과 풀다운은 반대의 개념을 가지고 있습니다.

풀업(PULL-UP)은 플로팅 상태일때의 값을 끌어올린다는 의미를 가지고 있습니다. 쉽게 설명하자면 버튼을 누르지 않았을 때의 값을 1(HIGH)로 설정하고, 버튼을 눌렀을 때 0(LOW)값으로 변경하는 방식입니다.

반대로, 풀다운(PULL-DOWN)은 플로팅 상태일 때 값을 낮춘다는 의미를 가지고 있습니다. 버튼을 누르지 않았을 때 즉, 스위치가 열려있을 때의 값을 0(LOW)으로 설정합니다.

풀업 방식으로 버튼과 아두이노 보드를 연결할 때는 그림3과 같이 아두이노 보드의 5V핀에 저항, 버튼 순으로 연결하고, 버튼의 같은 내부 연결핀과 아두이노 보드의 입력핀에 연결합니다.

풀다운 방식은 그림 4와 같이 아두이노 보드의 5V핀이 아닌 GND핀에 저항과 버튼을 연결하고, 버튼의 같은 내부 연결핀과 아두이노 보드의 입력핀을 연결합니다.

만약, 버튼과 아두이노 보드가 풀업 방식으로 연결되어 있으면 버튼을 눌렀을 때 작동을 하지 않고, 버튼에서 손을 떼면 LED와 같은 장치가 작동을 합니다.

버튼을 눌렀을 때 작동을 해야 하므로, 풀다운 방식으로 연결하여 사용합니다.

LED 사용법

1

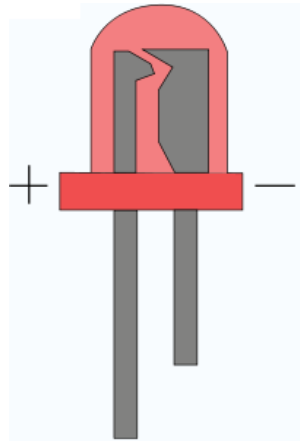
LED 외형



LED는 그림과 같이 생겼습니다. 신호등과 자동차 헤드라이트, LED TV 등 실생활에 자주 쓰입니다. LED는 아두이노 보드에 연결하여 켜고 끌 수 있습니다.

2

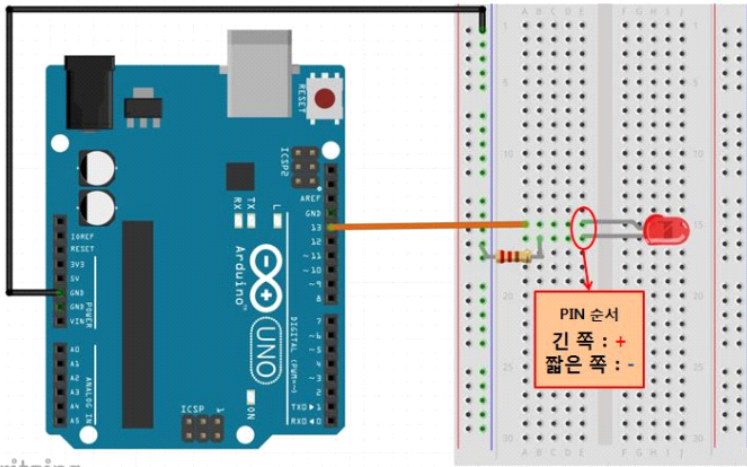
LED 극성



LED는 극성이 있습니다. 일반적으로 다리의 긴 쪽이 (+), 짧은 쪽이 (-) 극성을 띠지만 간혹 LED 다리 길이가 같은 경우는 LED단면이 넓은 쪽이 (-)극성입니다. 극성을 다르게 연결했을 때는 LED가 터지거나 반대로 작동하므로 극성에 주의해주세요.

3

LED 연결

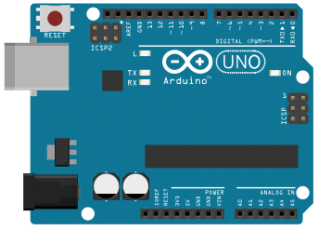


LED를 아두이노 보드에 연결할 때는 LED의 짧은 쪽에 저항을 연결하고 저항과 아두이노 보드의 GND(-극)를 연결합니다. LED의 긴 쪽은 아두이노 보드의 입력 핀에 연결합니다.

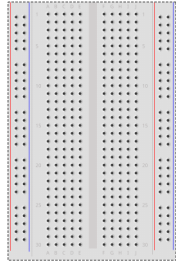
LED 켜보기

1

부품 준비하기



아두이노 보드 1개



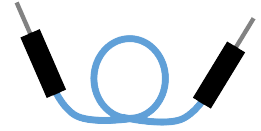
브레드보드 1개



LED 1개



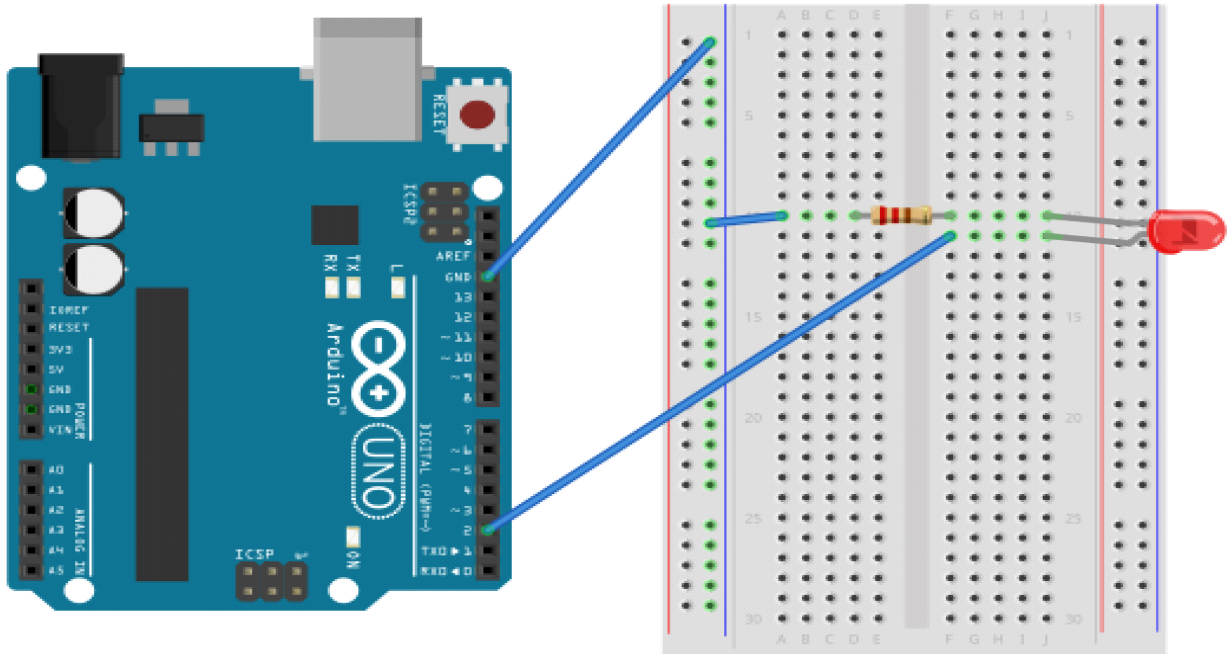
220Ω 저항 1개



수수 점퍼선 3개

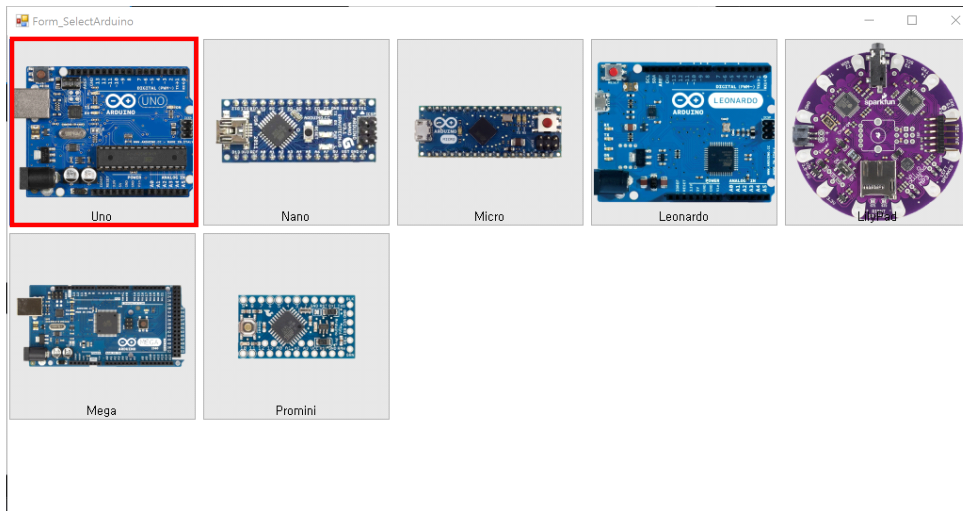
2

회로 구성하기



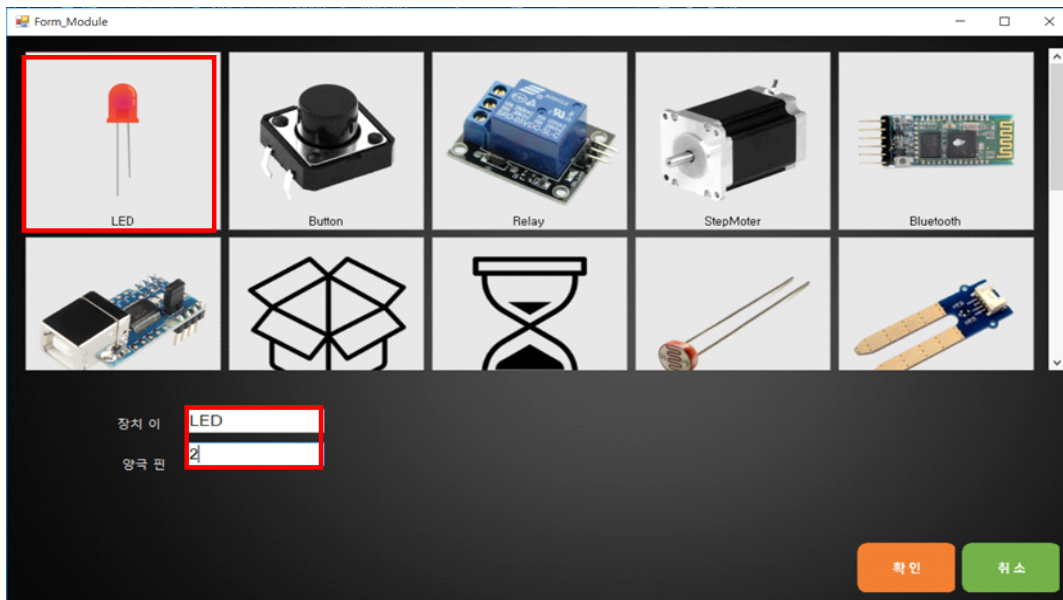
1. 브레드보드에 LED를 연결합니다.
2. LED의 음극(-) 다리에 저항을 연결합니다.
3. 아두이노 보드의 GND핀과 브레드보드의 GND라인을 연결합니다.
4. GND라인과 저항을 연결합니다.
5. 아두이노 보드의 2번 핀과 LED의 양극(+) 핀을 연결합니다.

Step 1) 아두이노 선택



- 이번 실습에서 사용할 아두이노보드를 선택합니다.

Step 2) 모듈 등록



'LED켜기' 실습 목표를 위해서 사용할 모듈에 'LED'를 등록합니다.

1. 장치 이름은 'LED'로 입력합니다.
2. 아두이노의 디지털 2번 핀에 연결하였기 때문에 양극 핀은 '2'를 입력합니다.

Step 3) 트리거 만들기

1. 트리거 명 지정하기

Form_Function

LED켜기

모듈이

일 때

모듈을

한다.

확인 취소

아두이노가 수행할 기능의 이름을 지정해 준다.

2. 조건부 : “누가”, “언제” 설정하기

: LED 켜기에선 조건부가 존재 하지 않으므로 생략

3. 명령부 : “무엇을”, “어떻게” 설정하기

Form_Function

LED켜기

모듈이

일 때

LED 모듈을

HIGH 설정 한다.

확인 취소

무엇을 : LED

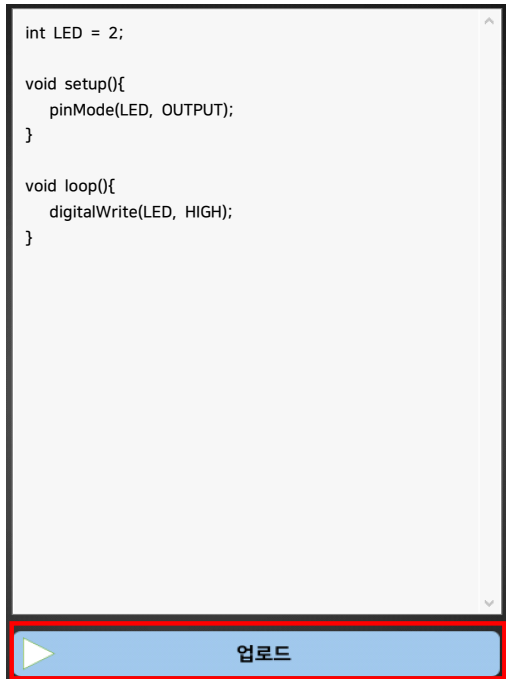
어떻게 : HIGH(켜기) 설정 합니다.

Step 4) 결과 및 업로드

```
int LED = 2;

void setup(){
  pinMode(LED, OUTPUT);
}

void loop(){
  digitalWrite(LED, HIGH);
}
```



1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.
2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

4

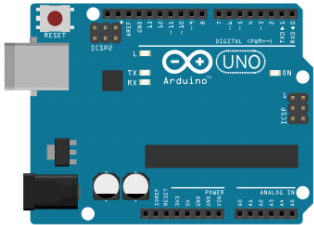
결과 소스코드

```
1   int LED = 2;           //디지털 2번 핀에 LED 연결
2
3   void setup(){
4     pinMode(LED, OUTPUT); //디지털 2번 핀을 출력 모드로 설정
5   }
6   void loop(){
7     digitalWrite(LED, HIGH); //디지털 2번 핀으로 HIGH(1 또는 ON)출력
8   }
```

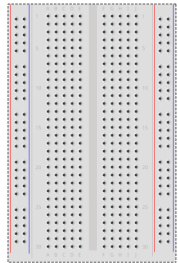
LED 깜빡이기

1

부품 준비하기



아두이노 보드 1개



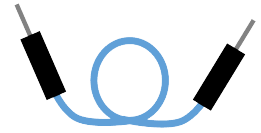
브레드보드 1개



LED 1개



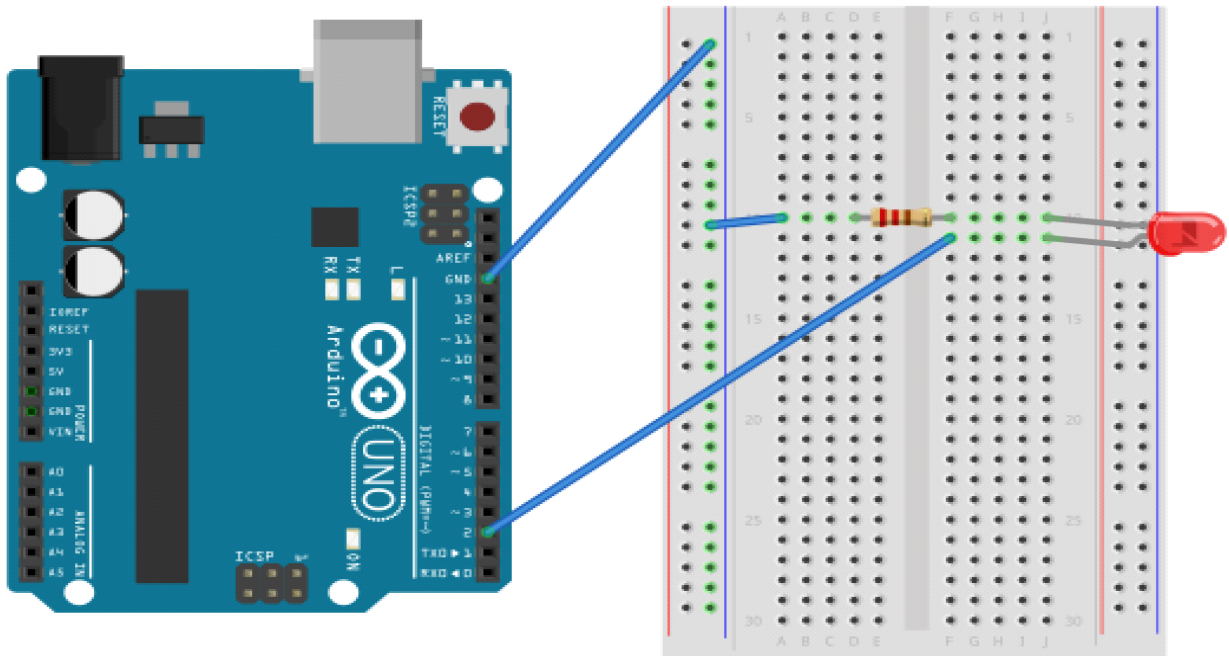
220Ω 저항 1개



수수 점퍼선 3개

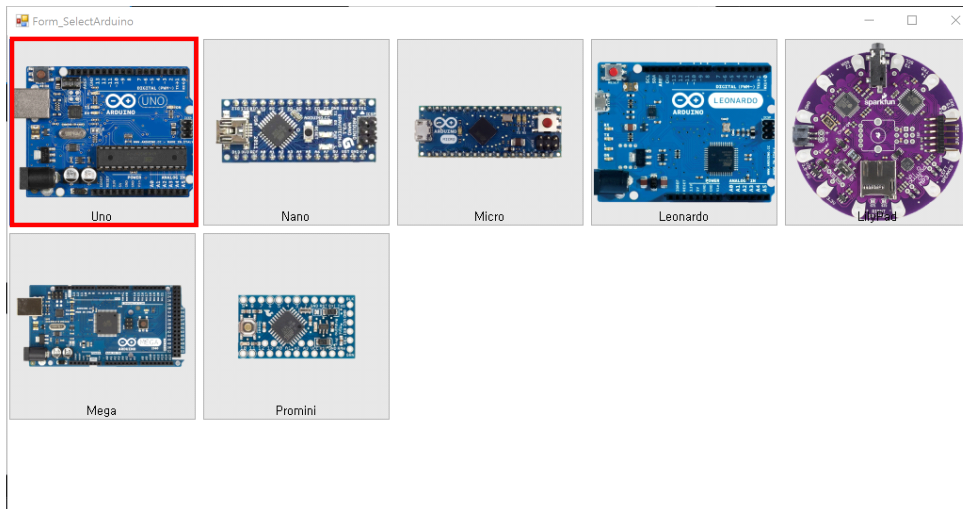
2

회로 구성하기



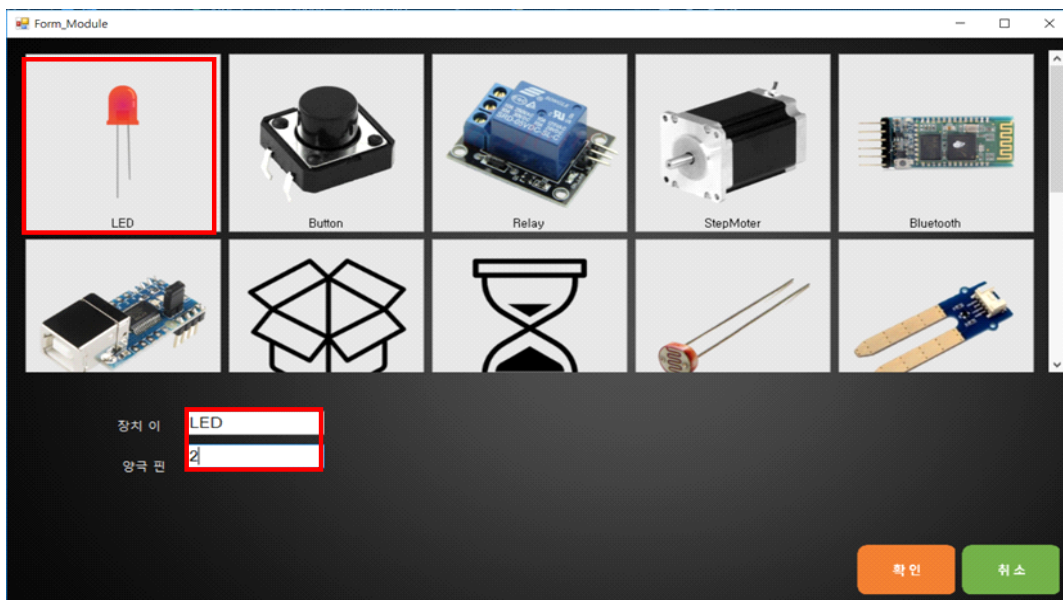
1. 브레드보드에 LED를 연결합니다.
2. LED의 음극(-) 다리에 저항을 연결합니다.
3. 아두이노 보드의 GND핀과 브레드보드의 GND라인을 연결합니다.
4. GND라인과 저항을 연결합니다.
5. 아두이노 보드의 2번 핀과 LED의 양극(+) 핀을 연결합니다.

Step 1) 아두이노 선택



- 이번 실습에서 사용할 아두이노보드를 선택합니다.

Step 2) 모듈 등록



'LED켜기' 실습 목표를 위해서 사용할 모듈 LED를 등록합니다.

1. 장치 이름은 'LED'로 입력합니다.
2. 아두이노의 디지털 2번 핀에 연결하였기 때문에 양극 핀은 '2'를 입력합니다.

Step 3) 트리거 만들기

1. 트리거 명 지정하기

The screenshot shows a window titled 'Form_Function' with a text input field containing 'LED 깜빡이기'. Below it are several dropdown menus for selecting modules and actions. At the bottom, there are '확인' (Confirm) and '취소' (Cancel) buttons.

아두이노가 수행할 기능의 이름을 지정해 준다.

2. 조건부 : "누가", "언제" 설정하기

: LED 켜기에서는 조건부가 존재하지 않으므로 생략한다.

3. 명령부 : "무엇을", "어떻게" 설정하기

The screenshot shows the 'Form_Function' window with the 'Command' section expanded. It includes dropdowns for 'LED' (set to 'HIGH') and 'Delay' (set to '1000'). There are also '확인' (Confirm) and '취소' (Cancel) buttons at the bottom.

무엇을 : LED / Delay

어떻게 : LED를 HIGH(켜기) 설정 하고 1초 동안 멈춘다.

Form_Function

LED 깜빡이기

모듈이

일 때

1000 설정 하고,

LED 모듈을

LOW 설정 하고,

Delay 모듈을

1000 설정 한다.

확인 취소

무엇을 : LED / Delay

어떻게 : LED를 LOW(끄기) 설정 하고 1초 동안 멈춘다.

Step 4) 결과 및 업로드

```
int LED = 2;

void setup(){
  pinMode(LED, OUTPUT);
}

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

업로드
```

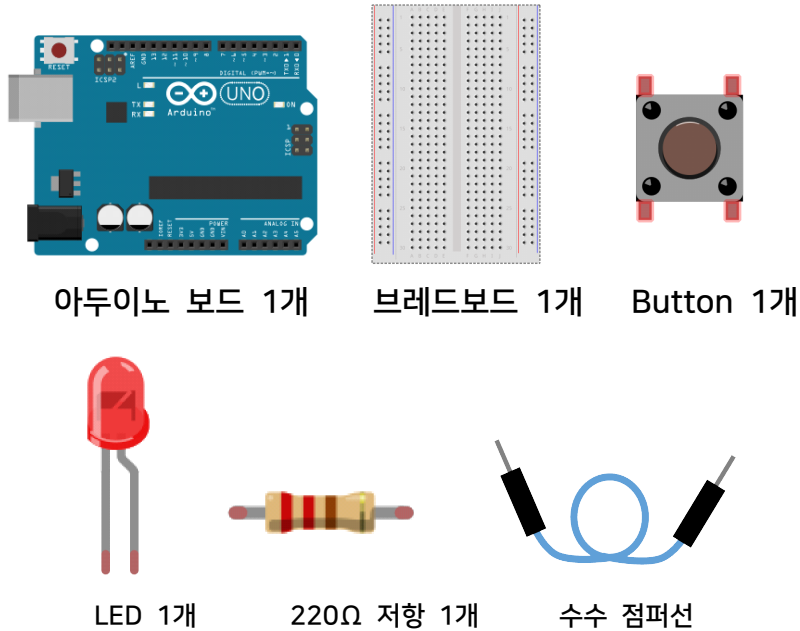
1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.
2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

```
1 int LED = 2; //디지털 2번 핀에 LED 연결
2
3 void setup(){
4     pinMode(LED, OUTPUT); //디지털 2번 핀을 출력 모드로 설정
5 }
6
7 void loop(){
8     digitalWrite(LED, HIGH); //디지털 2번 핀으로 HIGH(1 또는 ON)출력
9     delay(1000); //1초 동안 멈춤
10    digitalWrite(LED, LOW); //디지털 2번 핀으로 LOW(0 또는 OFF)출력
11    delay(1000); //1초 동안 멈춤
12 }
```

Button 3번 클릭 시 LED 켜기

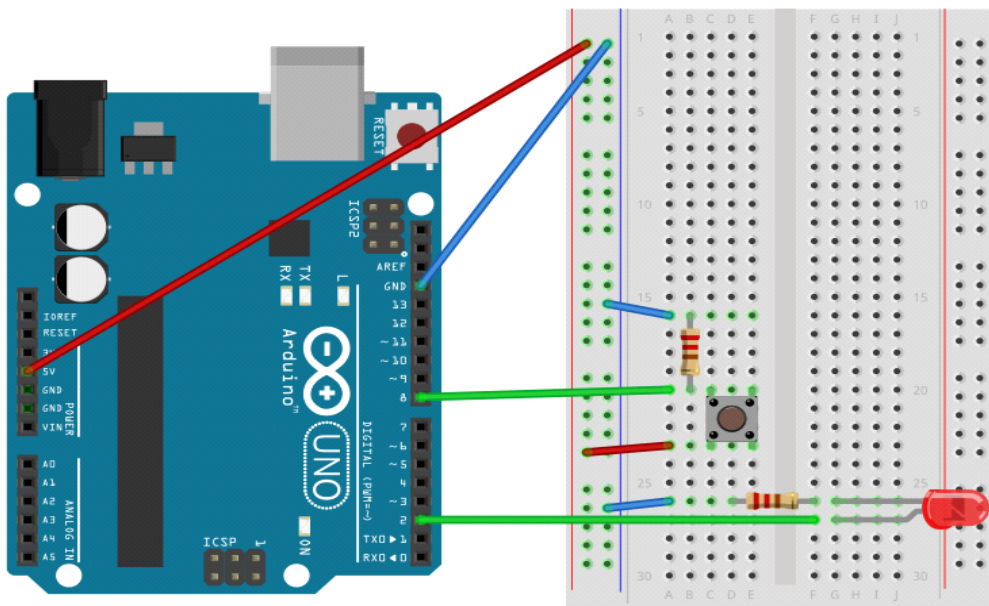
1

부품 준비하기



2

회로 구성하기

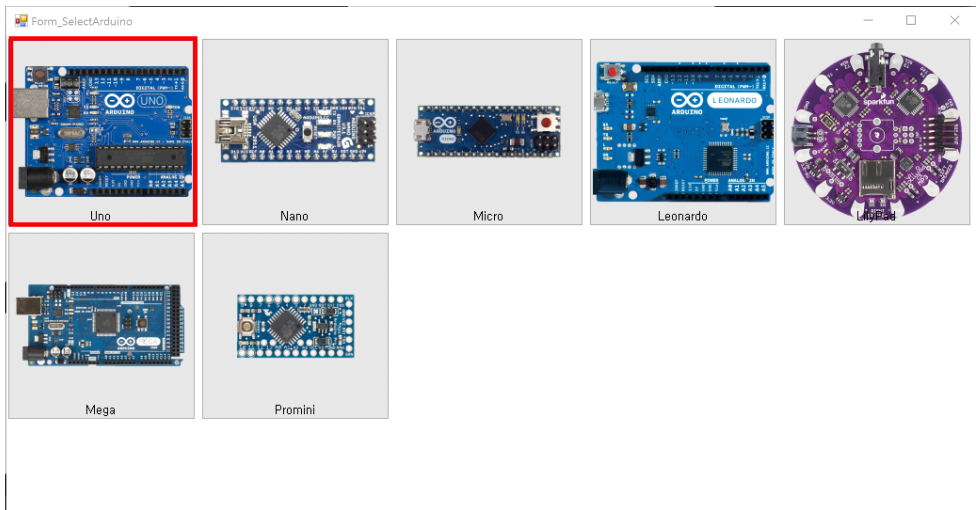


1. 브레드보드에 LED를 연결합니다.
2. LED의 음극(-) 다리에 저항을 연결합니다.
3. 아두이노 보드의 GND핀과 5V핀을 브레드보드의 GND라인과 Vcc라인에 연결합니다.
4. GND라인과 저항을 연결합니다.

5. 아두이노 보드의 2번 핀과 LED의 양극(+) 핀을 연결합니다.
6. 브레드보드에 BUTTON을 연결합니다.
7. BUTTON에 저항을 연결한 후 GND라인에 연결합니다.
8. 저항이 연결된 BUTTON 핀에 아두이노 8번 핀을 연결합니다.
9. 아두이노의 Vcc라인과 버튼을 연결합니다.

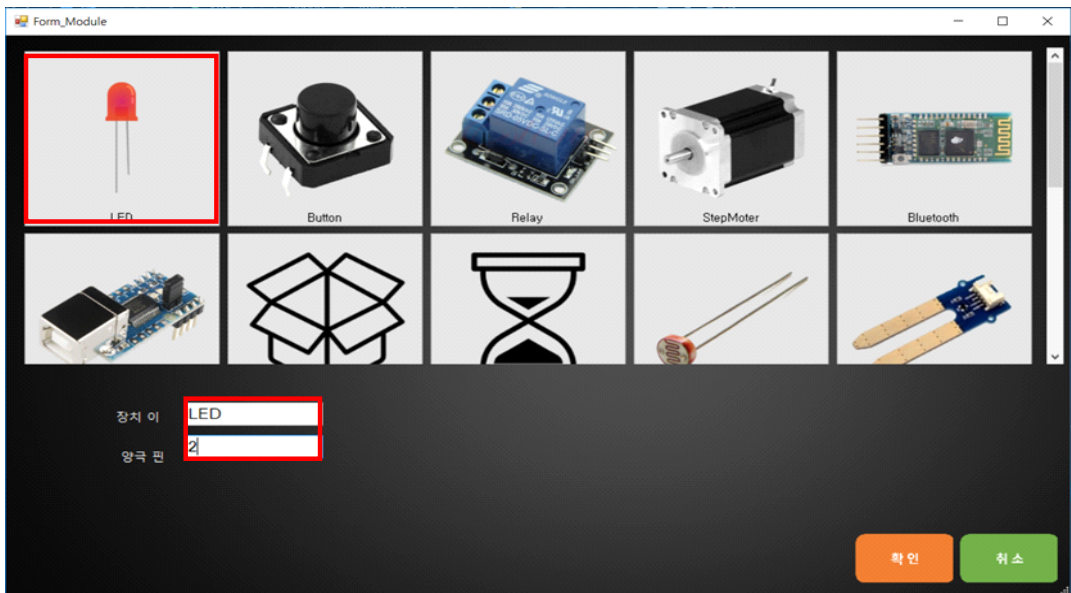
3 AI Studio

Step 1) 아두이노 선택



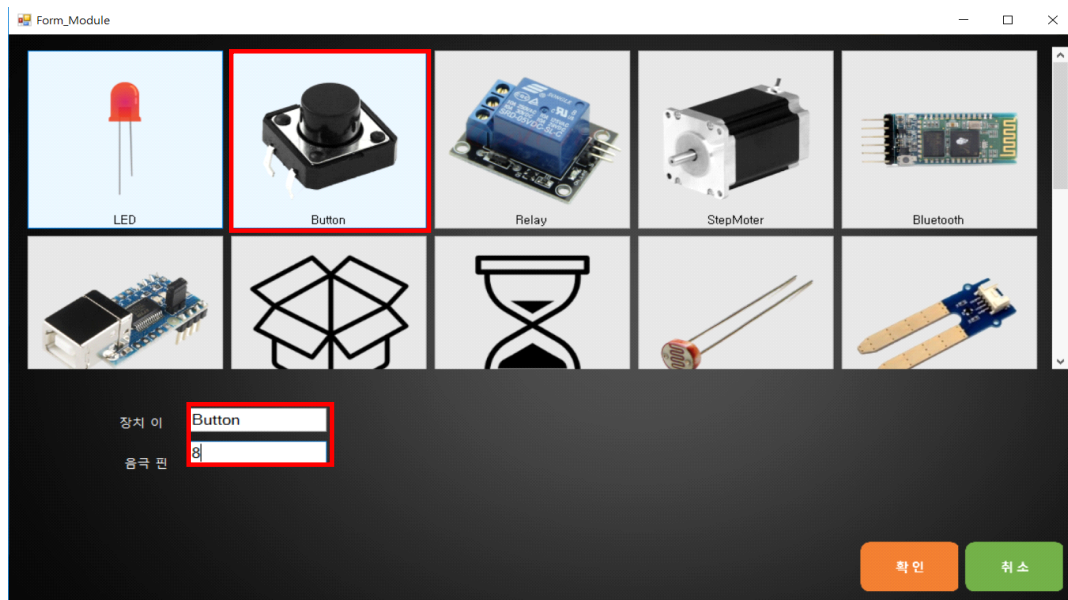
- 이번 실습에서 사용할 아두이노보드를 선택합니다.

Step 2) 모듈 등록



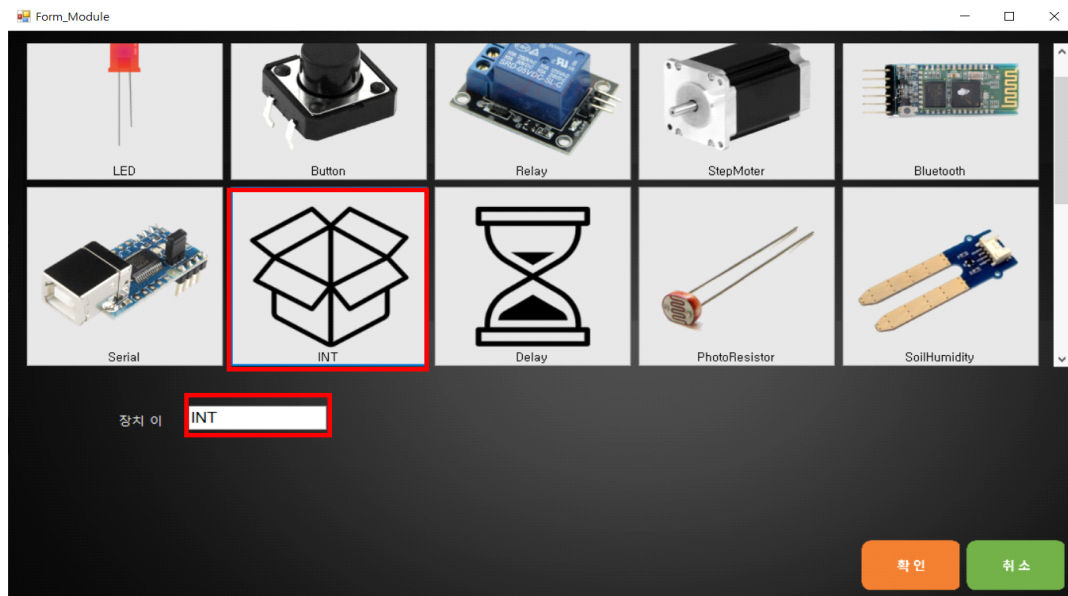
실습을 위해서 프로그램에 사용할 모듈에 LED를 등록합니다.

1. 장치 이름은 'LED'로 입력합니다.
2. 아두이노의 디지털 2번 핀에 연결하였기 때문에 양극 핀은 '2'를 입력합니다.



실습을 위해서 프로그램에 사용할 모듈에 BUTTON를 등록합니다.

1. 장치 이름은 'BUTTON'으로 입력합니다.
2. 아두이노의 디지털 8번 핀에 연결하였기 때문에 핀은 '8'를 입력합니다.



실습을 위해서 프로그램에 사용할 변수를 등록합니다.

1. 변수의 이름은 'INT'으로 입력합니다.

Step 3) 트리거 만들기

1. 트리거 명 지정하기

아두이노가 수행할 기능의 이름을 지정해 준다.

2_1 조건부 : "누가", "언제" 설정하기

누가 : Button 언제 : HIGH(눌렀을 때)

무엇을 : 변수 INT을 어떻게 1을 더한다.

2_2 조건부 : "누가", "언제" 설정하기

Form_Function

Button 3번 클릭 시 LED 켜기

INT 모듈이

3 같은 값 일 때

모듈을

한다.

확인 취소

누가 : 변수 INT 가

언제 : 3일 때

3. 명령부 : “무엇을”, “어떻게” 설정하기

Form_Function

Button 클릭 시 Count

INT 모듈이

3 같은 값 일 때

LED 모듈을

HIGH 설정 한다.

확인 취소

무엇을 : LED

어떻게 : LED를 HIGH(켜기) 설정 한다.

Step 4) 결과 및 업로드

```
int LED = 2;
int Button = 8;
int INT = 0;

void setup(){
  pinMode(LED, OUTPUT);
  pinMode(Button, INPUT);
}

void loop(){
  if(digitalRead(Button) == HIGH){
    INT = INT +1;
  }
  if( INT == 3){
    digitalWrite(LED, HIGH);
  }
}
```

업로드

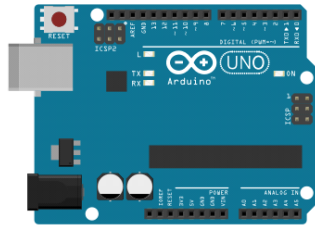
1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.
2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

```
1 int LED = 2; //디지털 2번 핀에 LED 연결
2 int Button = 8; //디지털 8번 핀에 Button 연결
3 int INT = 0; //변수 INT에 0 저장
4
5 void setup(){
6     pinMode(LED, OUTPUT); //디지털 2번 핀을 출력 모드로 설정
7     pinMode(Button, INPUT); //디지털 8번 핀을 입력 모드로 설정
8 }
9
10 void loop(){
11     if(digitalRead(Button) == HIGH){ //Button이 눌렸을 때
12         INT = INT +1; //변수 INT에 1을 더한다
13     }
14     if( INT == 3){ //변수 INT가 3이 되었다면
15         digitalWrite(LED, HIGH); //디지털 2번 핀으로 HIGH(1 또는 ON)출력
16     }
17 }
```

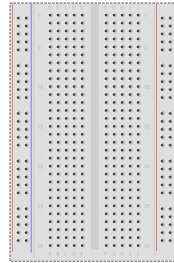
버튼을 활용하여 LED 제어하기

1

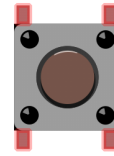
부품 준비하기



아두이노 보드 1개



브레드보드 1개



버튼 1개



LED 1개



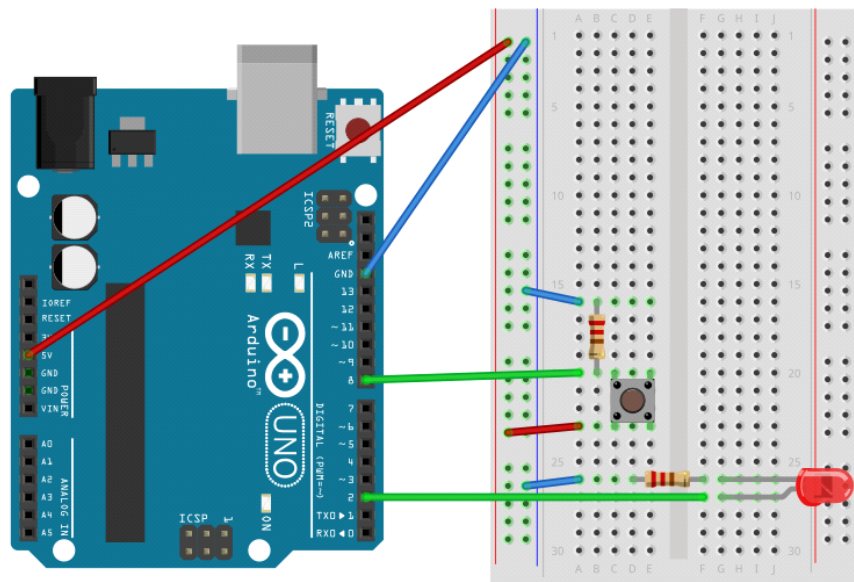
220Ω 저항 1개



수수 점퍼선 5개

2

회로 구성하기



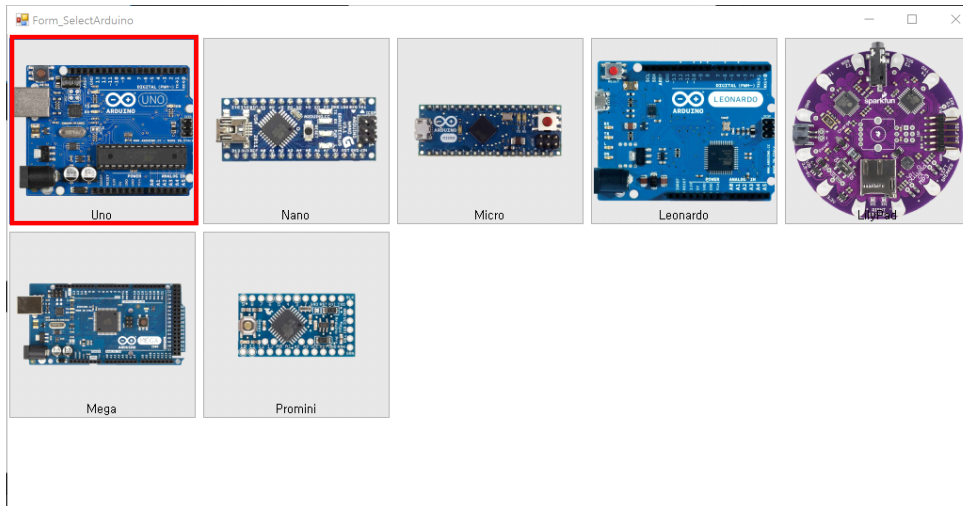
1. 브레드보드에 LED를 연결합니다.
2. LED의 음극(-) 다리에 저항을 연결합니다.
3. 아두이노 보드의 GND핀과 5V핀을 브레드보드의 GND라인과 Vcc라인에 연결합니다.
4. GND라인과 저항을 연결합니다.
5. 아두이노 보드의 2번 핀과 LED의 양극(+) 핀을 연결합니다.

6. 브레드보드에 BUTTON을 연결합니다.
7. BUTTON에 저항을 연결한 후 GND라인에 연결합니다.
8. 저항이 연결된 BUTTON 핀에 아두이노 8번 핀을 연결합니다.
9. 아두이노의 Vcc라인과 버튼을 연결합니다.

3

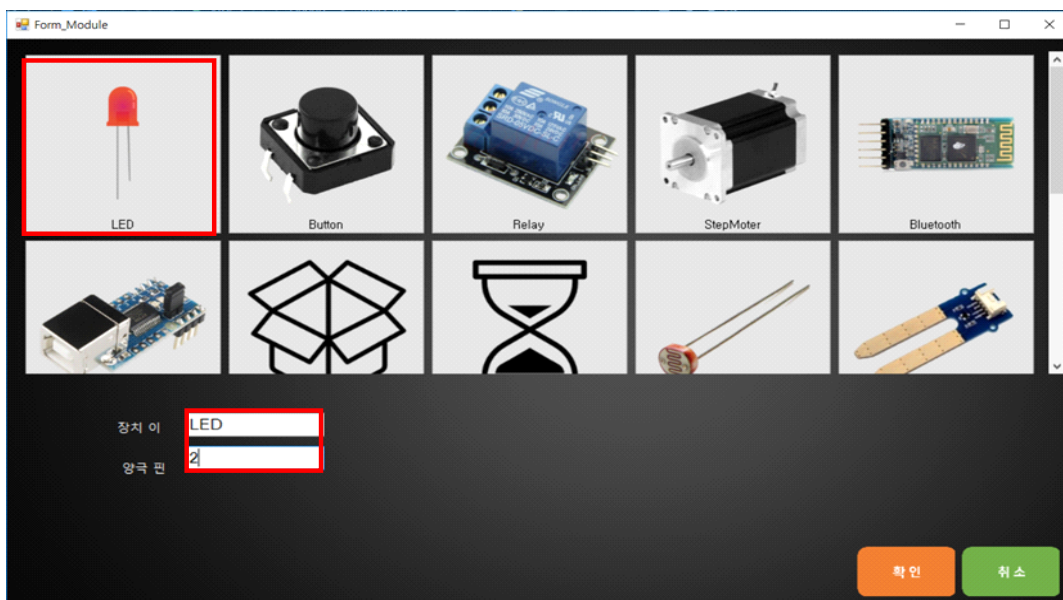
AI Studio

Step 1) 아두이노 선택



- 이번 실습에서 사용할 아두이노 보드를 선택합니다.

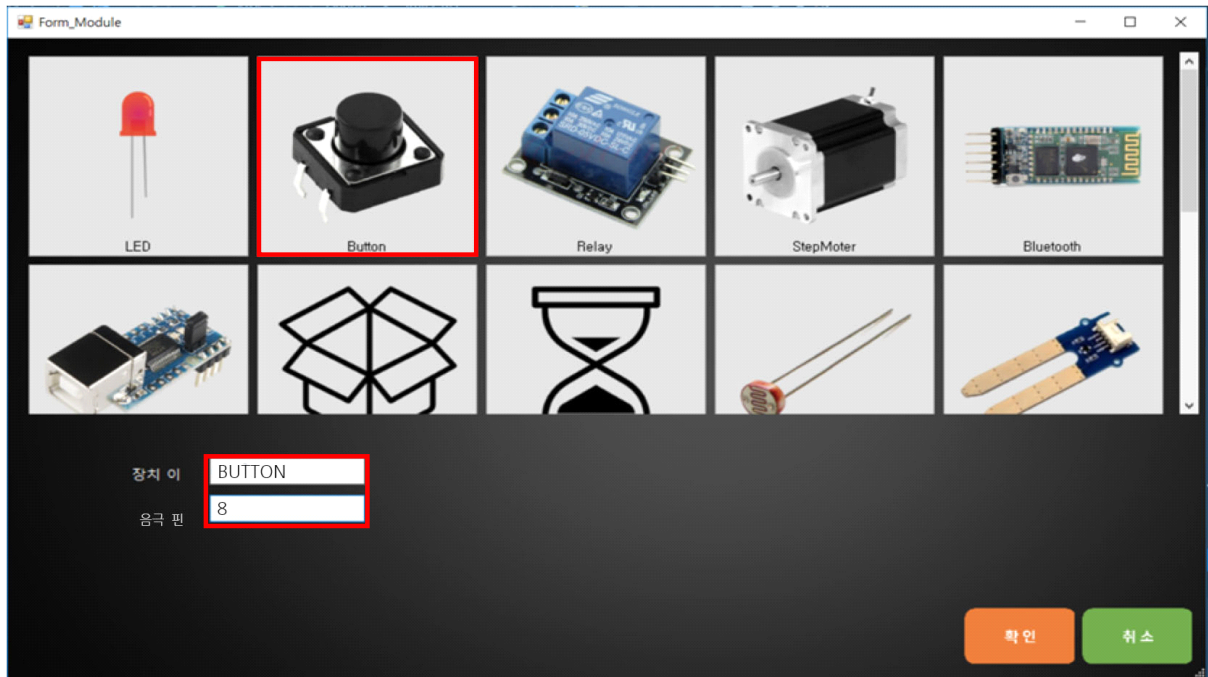
Step 2) 모듈 등록



실습을 위해서 프로그램에 사용할 모듈에 LED를 등록합니다.

1. 장치 이름은 'LED'로 입력합니다.

2. 아두이노의 디지털 2번 핀에 연결하였기 때문에 양극 핀은 '2'를 입력합니다.

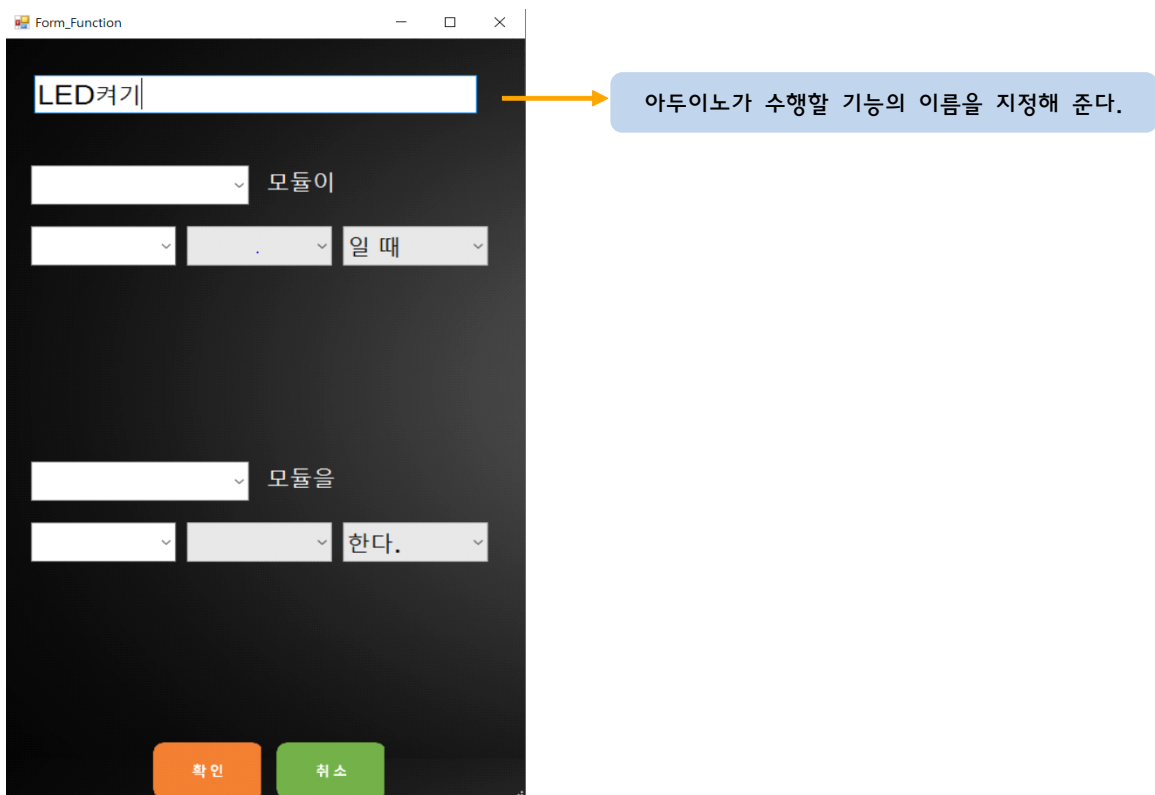


실습을 위해서 프로그램에 사용할 모듈에 버튼을 등록합니다.

1. 장치 이름은 'BUTTON'로 입력합니다.
2. 아두이노의 디지털 5번 핀에 연결하였기 때문에 음극 핀은 '5'를 입력합니다.

Step 3) 트리거 만들기

1. 트리거 명 지정하기



2. 조건부 : "누가", "언제" 설정하기

LED 켜기

BUTTON 모듈이

HIGH 같은 값 일 때

모듈을

한다.

확인 취소

누가 : 버튼

언제 : HIGH(눌림)일 때

3. 명령부 : "무엇을", "어떻게" 설정하기

LED켜기

BUTTON 모듈이

HIGH 같은 값 일 때

LED 모듈을

HIGH 설정 한다.

확인 취소

무엇을 : LED

어떻게 : HIGH(켜기) 설정 합니다.

1. 트리거 명 지정하기

아두이노가 수행할 기능의 이름을 지정해 준다.

2. 조건부 : "누가", "언제" 설정하기

누가 : 버튼

언제 : LOW(눌리지 않음)일 때

3. 명령부 : "무엇을", "어떻게" 설정하기

Form_Function

LED 끄기

BUTTON 모듈이

LOW 같은 값 일 때

LED 모듈을

LOW 설정 한다.

확인 취소

Step 4) 결과 및 업로드

```

int LED = 2;
int BUTTON = 8;

void setup(){
  pinMode(LED, OUTPUT);
  pinMode(BUTTON, INPUT);
}

void loop(){
  if(digitalRead(BUTTON)==HIGH)
  {
    digitalWrite(LED, HIGH);
  }
  if(digitalRead(BUTTON)==LOW)
  {
    digitalWrite(LED, LOW);
  }
}

```

업로드

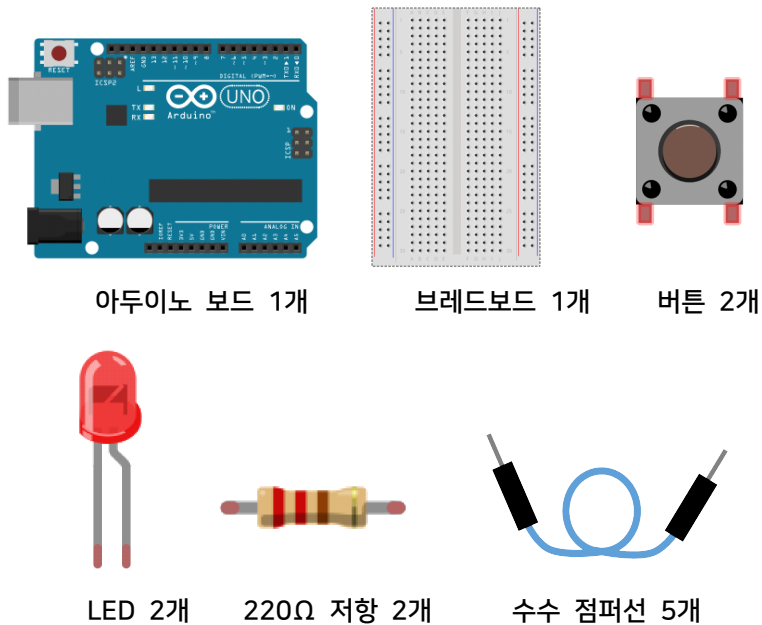
1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.
2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

```
1 int LED = 2; //디지털 2번 핀에 LED 연결
2 int BUTTON = 8; //디지털 5번 핀에 BUTTON 연결
3
4 void setup(){
5     pinMode(LED, OUTPUT); //디지털 2번 핀을 출력 모드로 설정
6     pinMode(BUTTON, INPUT); //디지털 5번 핀을 입력 모드로 설정
7 }
8
9 void loop(){
10     if(digitalRead(BUTTON)==HIGH) //디지털 5번 핀의 값이 HIGH와 같은지 판단
11     {
12         digitalWrite(LED, HIGH); //디지털 2번 핀으로 HIGH(1 또는 ON)출력
13     }
14     if(digitalRead(BUTTON)==LOW) //디지털 5번 핀의 값이 LOW와 같은지 판단
15     {
16         digitalWrite(LED, LOW); //디지털 2번 핀으로 LOW(0 또는 OFF)출력
17     }
18 }
```

2개의 버튼을 활용하여 2개의 LED 제어하기

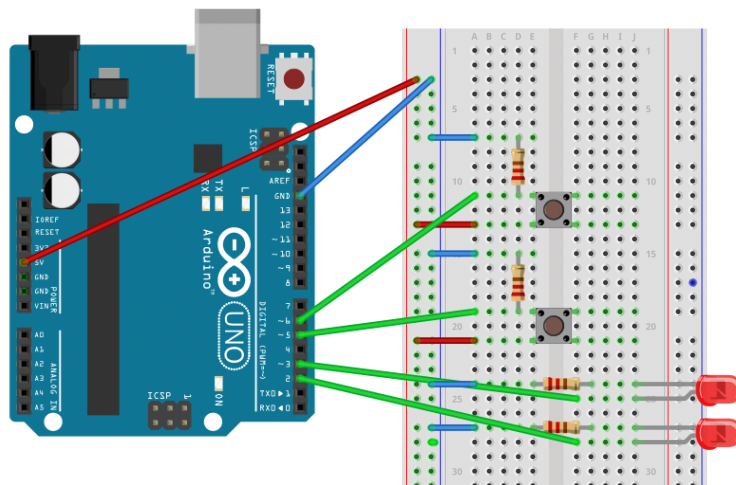
1

부품 준비하기



2

회로 구성하기

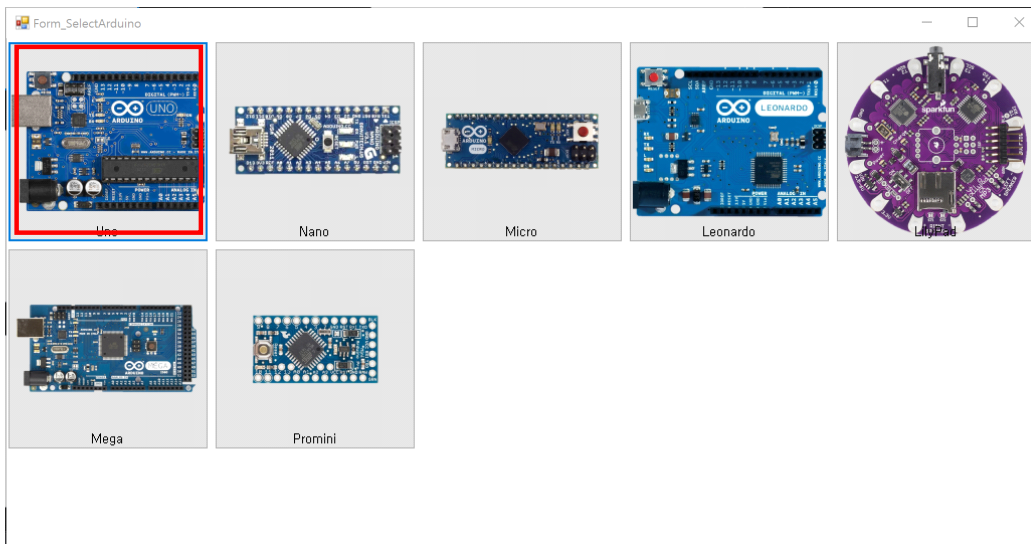


1. 브레드보드에 LED를 연결합니다.
2. LED1, LED2의 음극(-) 다리에 저항을 연결합니다.
3. 아두이노 보드의 GND핀과 5V핀을 브레드보드의 GND라인과 Vcc라인에 연결합니다.
4. GND라인과 저항을 연결합니다.
5. 아두이노 보드의 2번 핀과 LED의 양극(+) 핀을 연결합니다.
6. 아두이노 보드의 3번 핀과 LED의 양극(+) 핀을 연결합니다.
7. 브레드보드에 BUTTON1, BUTTON2을 연결합니다.

8. BUTTON1에 저항을 연결한 후 GND라인에 연결합니다.
9. BUTTON2에 저항을 연결한 후 GND라인에 연결합니다.
10. 저항이 연결된 BUTTON1 핀에 아두이노 5번 핀을 연결합니다.
11. 저항이 연결된 BUTTON2 핀에 아두이노 6번 핀을 연결합니다.
12. 아두이노의 Vcc라인과 버튼을 연결합니다.

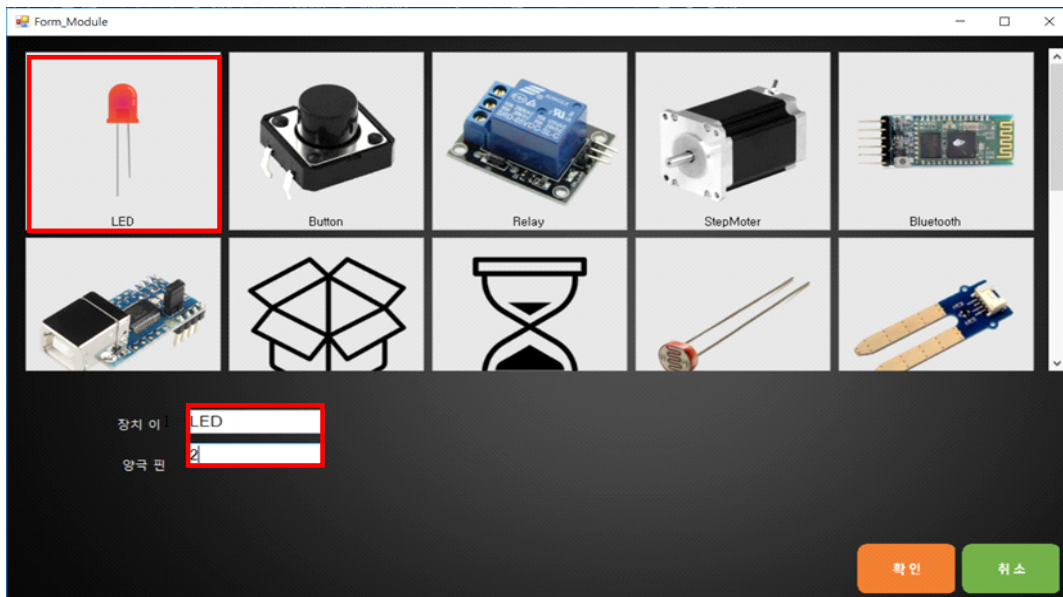
3 AI Studio

Step 1) 아두이노 선택



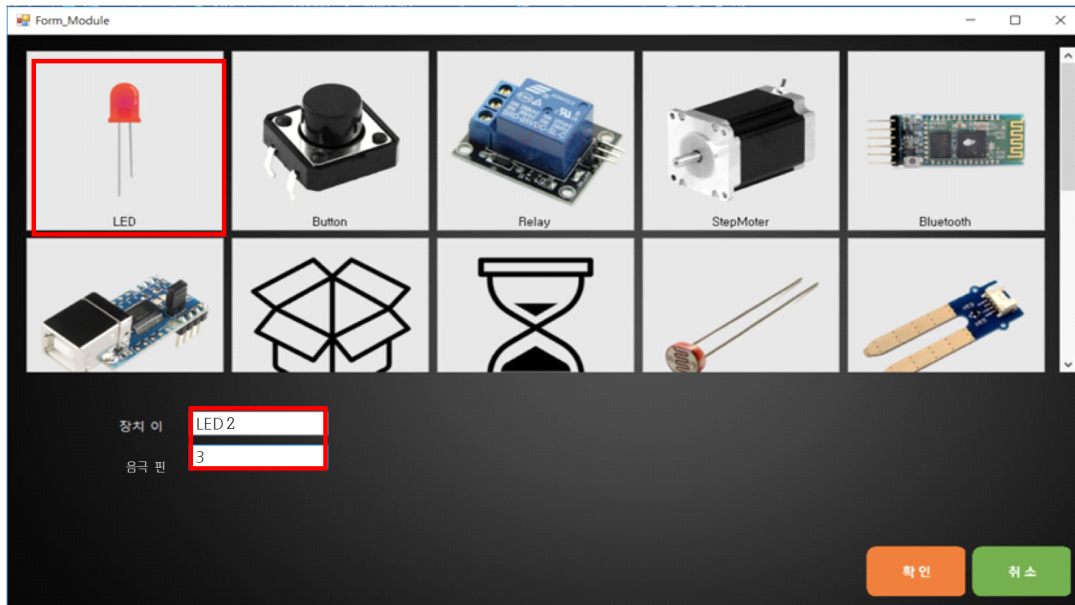
- 이번 실습에서 사용할 아두이노 보드를 선택합니다.

Step 2) 모듈 등록

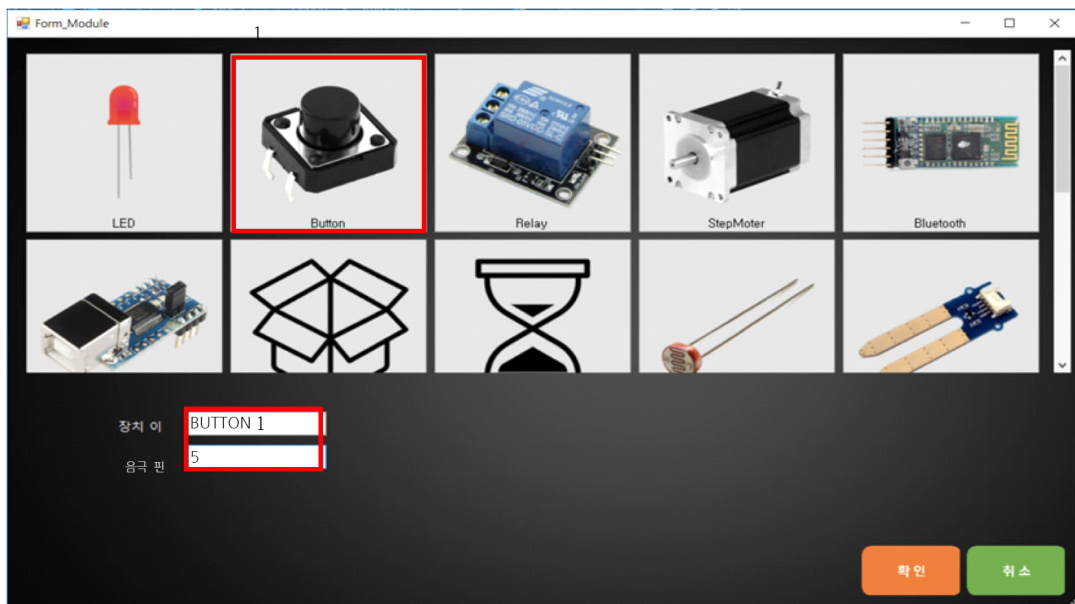


실습을 위해서 프로그램에 사용할 모듈에 LED를 2개 등록합니다.

1. 장치 이름은 'LED1'로 입력합니다.
2. 아두이노의 디지털 2번 핀에 연결하였기 때문에 양극 핀은 '2'를 입력합니다.

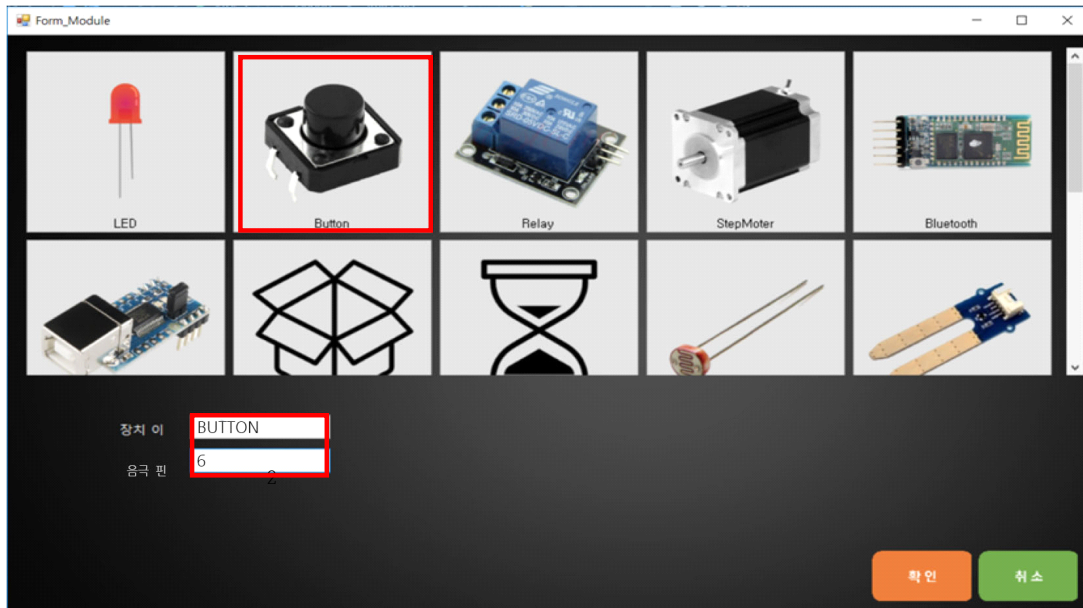


3. 장치 이름은 'LED2'로 입력합니다.
4. 아두이노의 디지털 3번 핀에 연결하였기 때문에 양극 핀은 '3'를 입력합니다.



실습을 위해서 프로그램에 사용할 모듈에 버튼을 등록합니다.

1. 장치 이름은 'BUTTON1'로 입력합니다.
2. 아두이노의 디지털 5번 핀에 연결하였기 때문에 음극 핀은 '5'를 입력합니다.

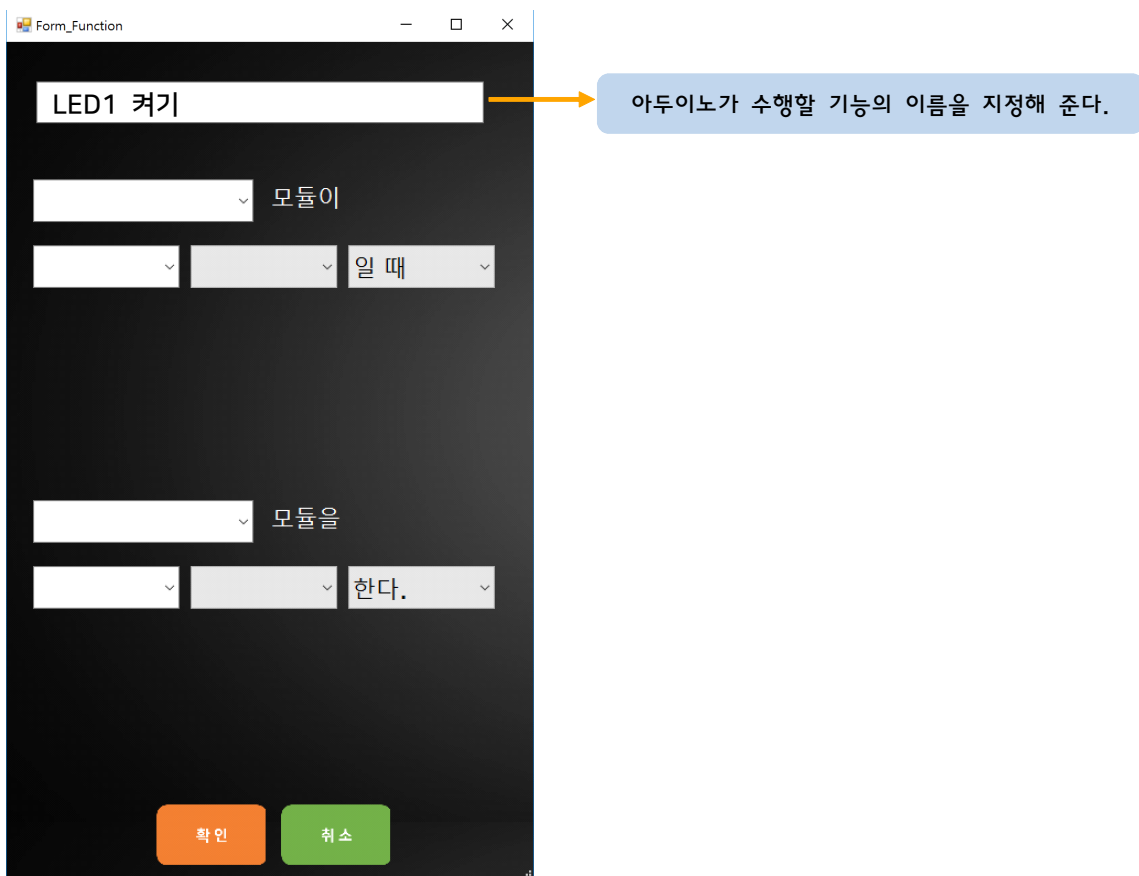


실습을 위해서 프로그램에 사용할 모듈에 버튼을 등록합니다.

1. 장치 이름은 'BUTTON2'로 입력합니다.
2. 아두이노의 디지털 6번 핀에 연결하였기 때문에 음극 핀은 '6'를 입력합니다.

Step 3) 트리거 만들기

1. 트리거 명 지정하기



2. 조건부 : "누가", "언제" 설정하기

Form_Function

LED1 켜기

BUTTON1 모듈이

HIGH 같은 값 일 때

모듈을

한다.

확인 취소

누가 : 버튼1

언제 : HIGH(눌림)일 때

3. 명령부 : "무엇을", "어떻게" 설정하기

Form_Function

LED1 켜기

BUTTON1 모듈이

HIGH 같은 값 일 때

LED1 모듈을

HIGH 설정 한다.

확인 취소

무엇을 : LED1

어떻게 : HIGH(켜기) 설정 합니다.

1. 트리거 명 지정하기

LED2 켜기

BUTTON2 모듈이

HIGH 같은 값 일 때

모듈을

한다.

확인 취소

아두이노가 수행할 기능의 이름을 지정해 준다.

2. 조건부 : "누가", "언제" 설정하기

LED2 켜기

BUTTON2 모듈이

HIGH 일 때

모듈을

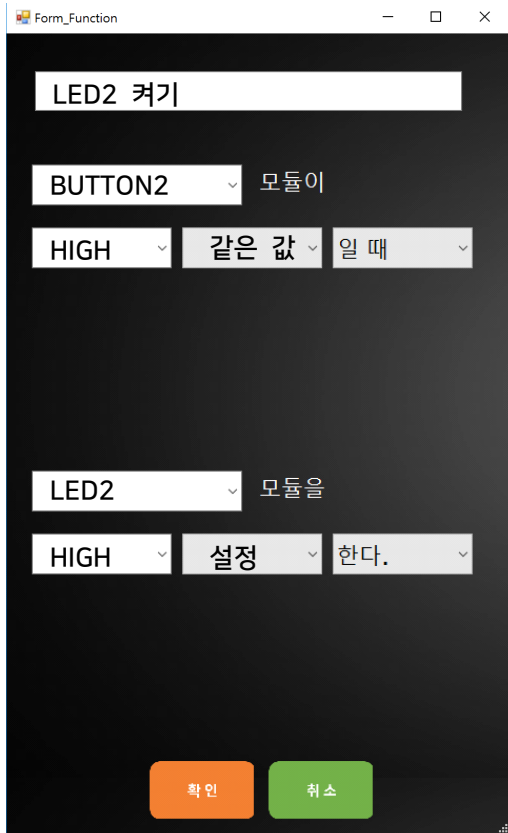
한다.

확인 취소

누가 : 버튼2

언제 : LOW(눌리지 않음)일 때

3. 명령부 : "무엇을", "어떻게" 설정하기



Step 4) 결과 및 업로드

```

int LED1 = 2;
int LED2 = 3;
int BUTTON1 = 5;
int BUTTON2 = 6;

void setup(){
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(BUTTON1, INPUT);
  pinMode(BUTTON2, INPUT);
}

void loop(){
  if(digitalRead(BUTTON1)==HIGH)
  {
    digitalWrite(LED1, HIGH);
  }
  if(digitalRead(BUTTON2)==HIGH)
  {
    digitalWrite(LED2, HIGH);
  }
}

```

업로드

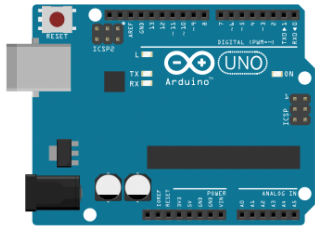
1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.
2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

```
1 int LED1 = 2; //디지털 2번 핀에 LED 연결
2 int LED2 = 3; //디지털 3번 핀에 LED 연결
3 int BUTTON = 5; //디지털 5번 핀에 BUTTON 연결
4 int BUTTON = 6; //디지털 6번 핀에 BUTTON 연결
5
6 void setup(){
7     pinMode(LED1, OUTPUT); //디지털 2번 핀을 출력 모드로 설정
8     pinMode(LED2, OUTPUT); //디지털 3번 핀을 출력 모드로 설정
9     pinMode(BUTTON1, INPUT); //디지털 5번 핀을 입력 모드로 설정
10    pinMode(BUTTON2, INPUT); //디지털 6번 핀을 입력 모드로 설정
11 }
12
13 void loop(){
14     if(digitalRead(BUTTON1)==HIGH)
15     { //디지털 5번 핀의 값이 HIGH와 같은지 판단
16         digitalWrite(LED1, HIGH); //디지털 2번 핀으로 HIGH(1 또는 ON)출력
17     }
18     if(digitalRead(BUTTON2)==LOW)
19     { //디지털 6번 핀의 값이 HIGH와 같은지 판단
20         digitalWrite(LED2, HIGH); //디지털 3번 핀으로 HIGH(1 또는 ON)출력
21     }
22 }
```

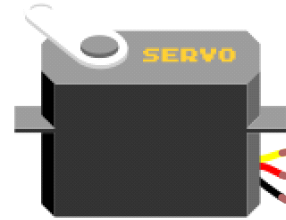
Servo Motor 제어하기

1

부품 준비하기



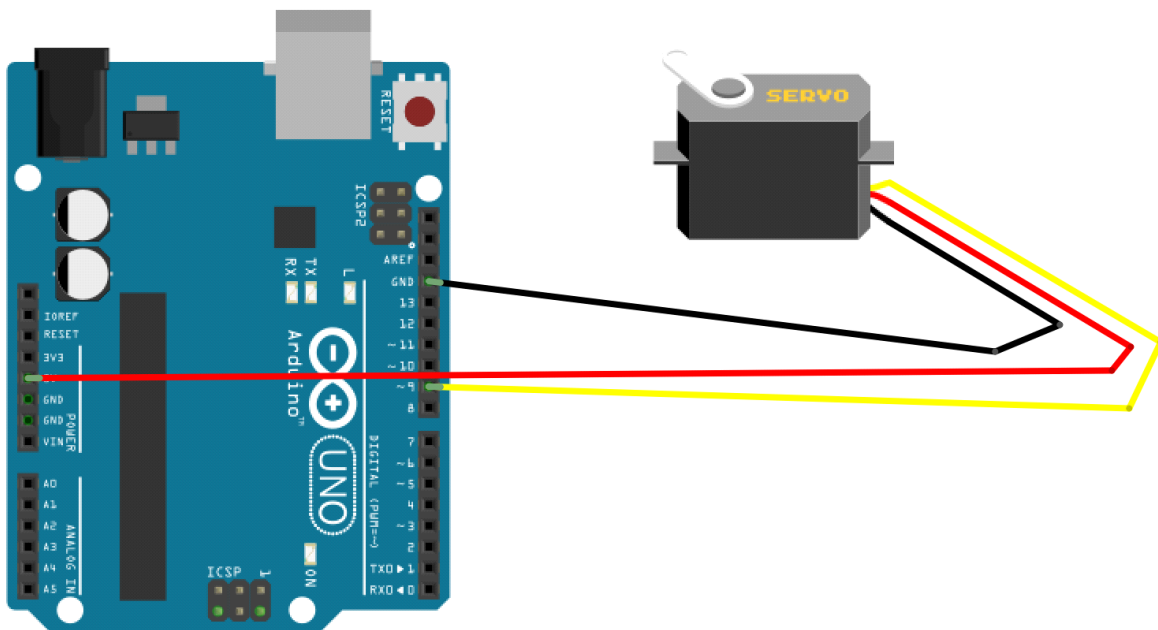
아두이노 보드 1개



서보모터 1개

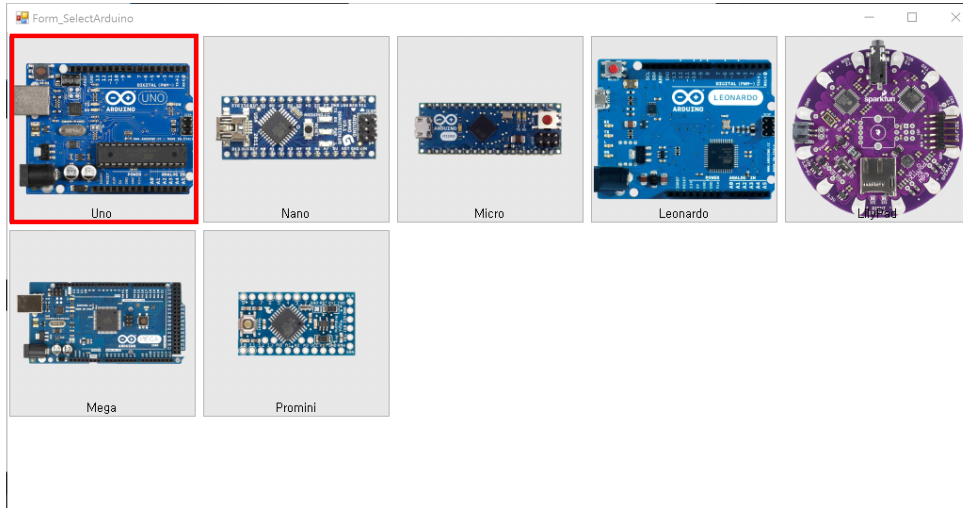
2

회로 구성하기



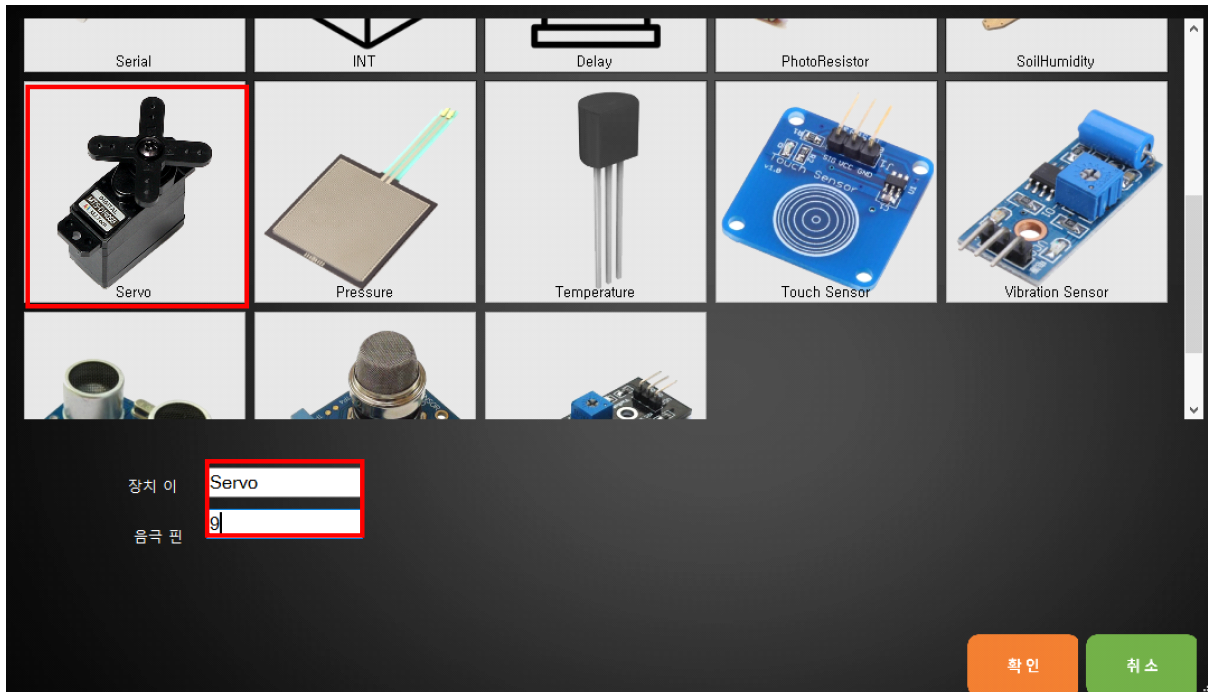
서보모터를 다음과 같이 연결합니다.

Step 1) 아두이노 선택



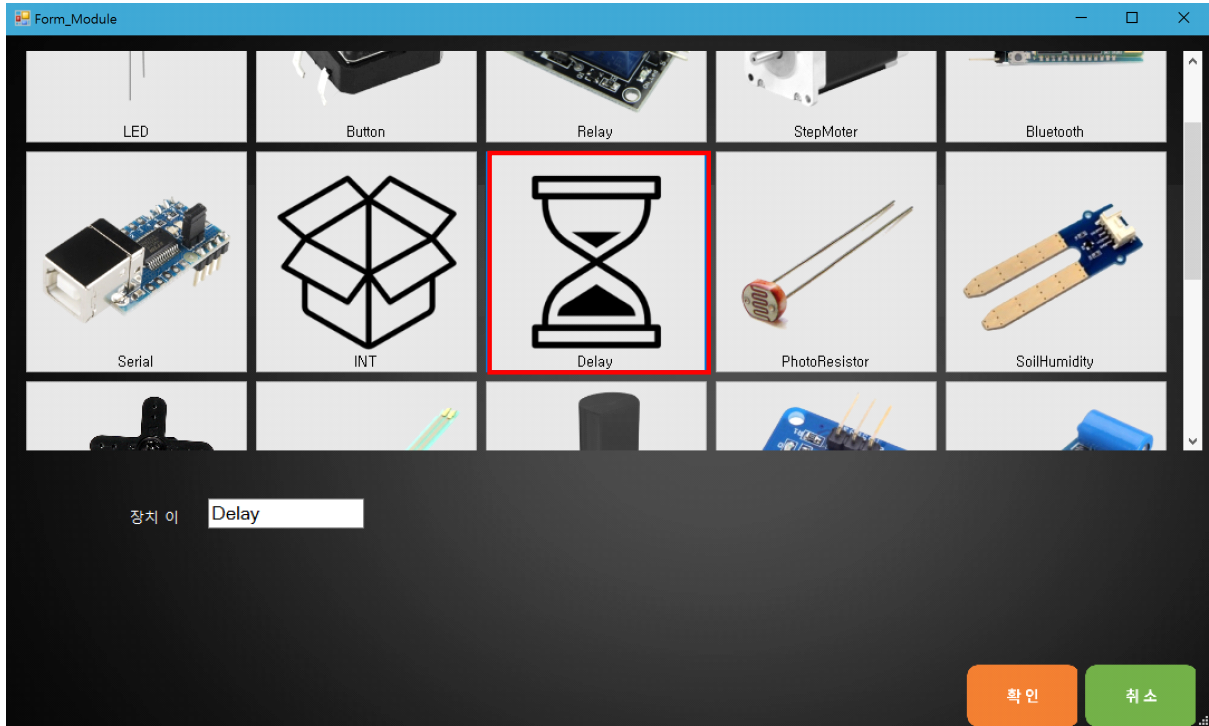
- 이번 실습에서 사용할 아두이노보드를 선택합니다.

Step 2) 모듈 등록



'Servo 모터 제어하기' 실습 목표를 위해서 사용할 모듈 Servo를 등록합니다.

1. 장치 이름은 'Servo'로 입력합니다.
2. 아두이노의 디지털 9번 핀에 연결하였기 때문에 양극 핀은 '9'를 입력합니다.



서보모터가 돌아가는 동안 프로그램을 잠시 멈추기 위해 Delay 모듈을 등록합니다.
 3. 장치 이름은 'Delay'로 입력합니다.

Step 3) 트리거 만들기

1. 트리거 명 지정하기
2. 조건부 : "누가", "언제" 설정하기
 : 서보모터 예선 조건부가 존재하지 않으므로 생략한다.
3. 명령부 : "무엇을", "어떻게" 설정하기

아두이노가 수행할 기능의 이름을 지정해 준다.

Form_Function

동작

모듈이

일 때

Servo 모듈을

180 설정 하고,

Delay 모듈을

1000 설정 하고,

Servo 모듈을

확인 취소

무엇을 : Servo / Delay

어떻게 : 서보모터를 180도로 회전하고 1초간 대기한다.

Form_Function

동작

모듈이

일 때

1000 설정 하고,

Servo 모듈을

0 설정 하고,

Delay 모듈을

1000 설정 한다.

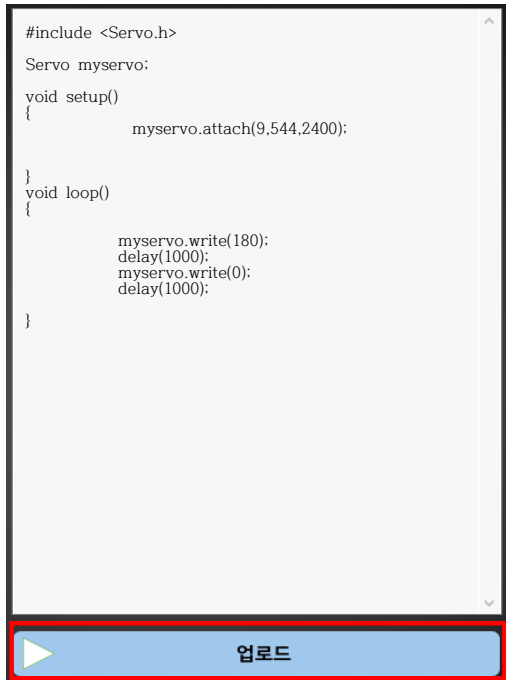
확인 취소

무엇을 : Servo / Delay

어떻게 : 서보모터를 0도로 회전하고 1초간 대기한다.

Step 4) 결과 및 업로드

```
#include <Servo.h>
Servo myservo;
void setup()
{
    myservo.attach(9,544,2400);
}
void loop()
{
    myservo.write(180);
    delay(1000);
    myservo.write(0);
    delay(1000);
}
```



1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.
2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

4

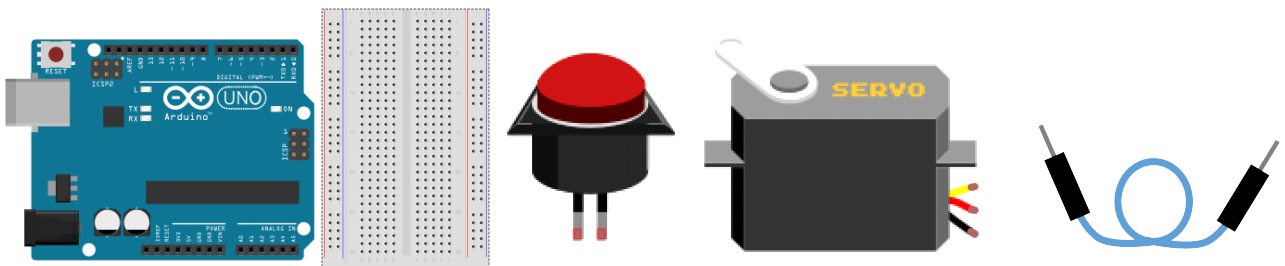
결과 소스코드

```
1  #include <Servo.h>
2
3  Servo myservo;
4
5  void setup()
6  {
7      myservo.attach(9,544,2400); //서보모터 초기화 값입니다
8
9
10 }
11 void loop()
12 {
13
14     myservo.write(180); //서보모터 각도를 180도로 조정합니다
15     delay(1000);       //서보모터가 회전하는 동안 프로그램을 1초간 정지
16     myservo.write(0); //서보모터 각도를 180도로 조정합니다
17     delay(1000);       //서보모터가 회전하는 동안 프로그램을 1초간 정지
18
19 }
```

서보모터 버튼을 제어하기

1

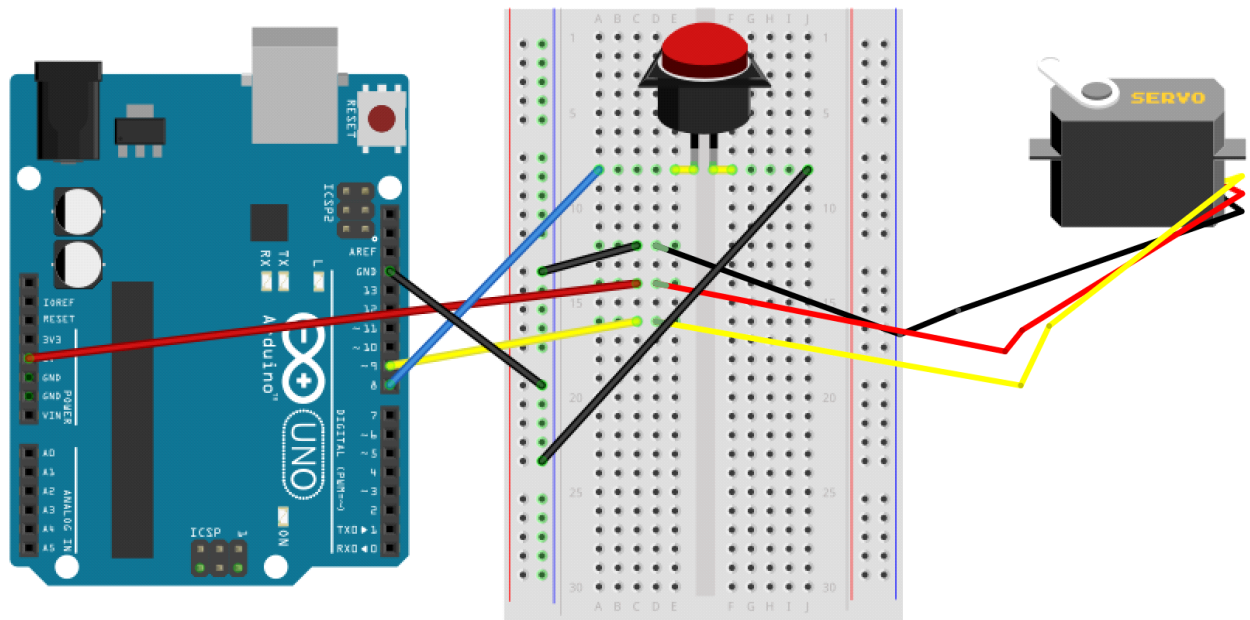
부품 준비하기



아두이노 보드 1개 브레드보드 1개 버튼 1개 서보모터 1개 수수 점퍼선 5개

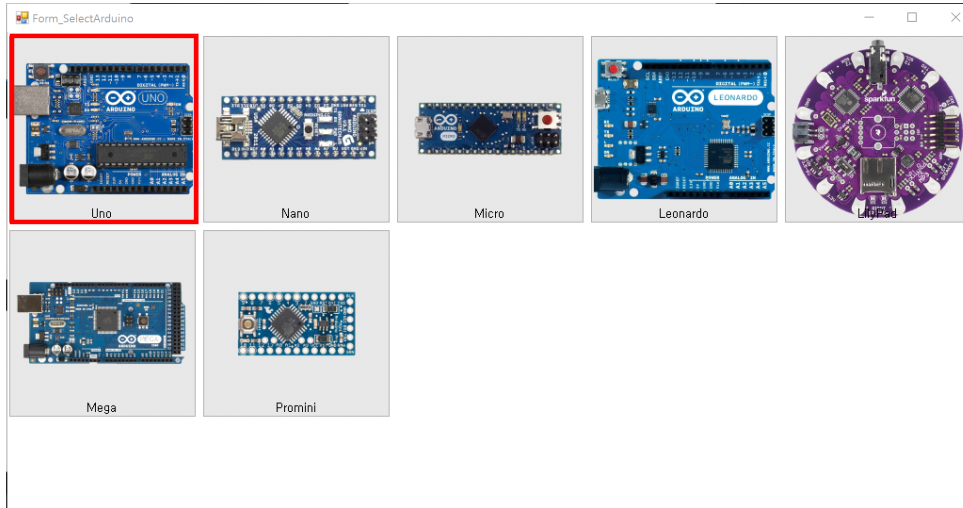
2

회로 구성하기



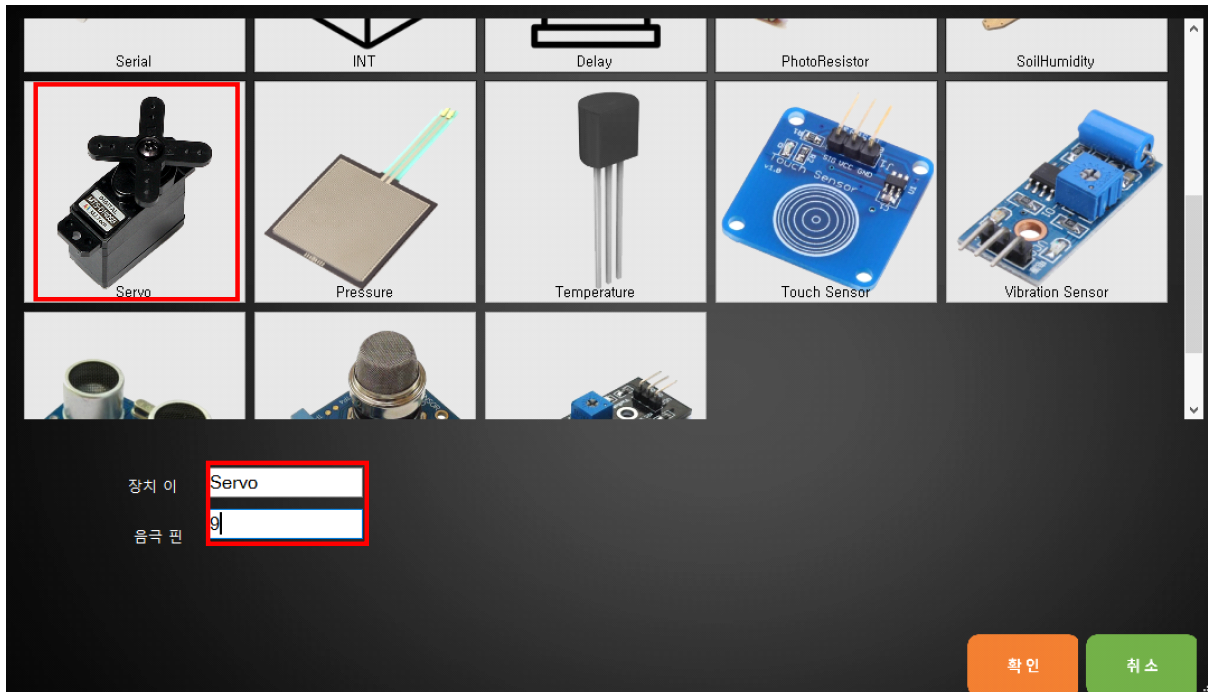
1. 브레드보드에 서보모터를 연결합니다.
2. 아두이노 보드의 GND핀과 브레드보드의 GND라인을 연결합니다.
3. 5V,GND라인과 서보모터 5V,GND라인을 연결합니다.
4. 아두이노 보드의 9번 핀과 서보모터 핀을 연결합니다.
5. 버튼을 8번 핀과 연결합니다.

Step 1) 아두이노 선택



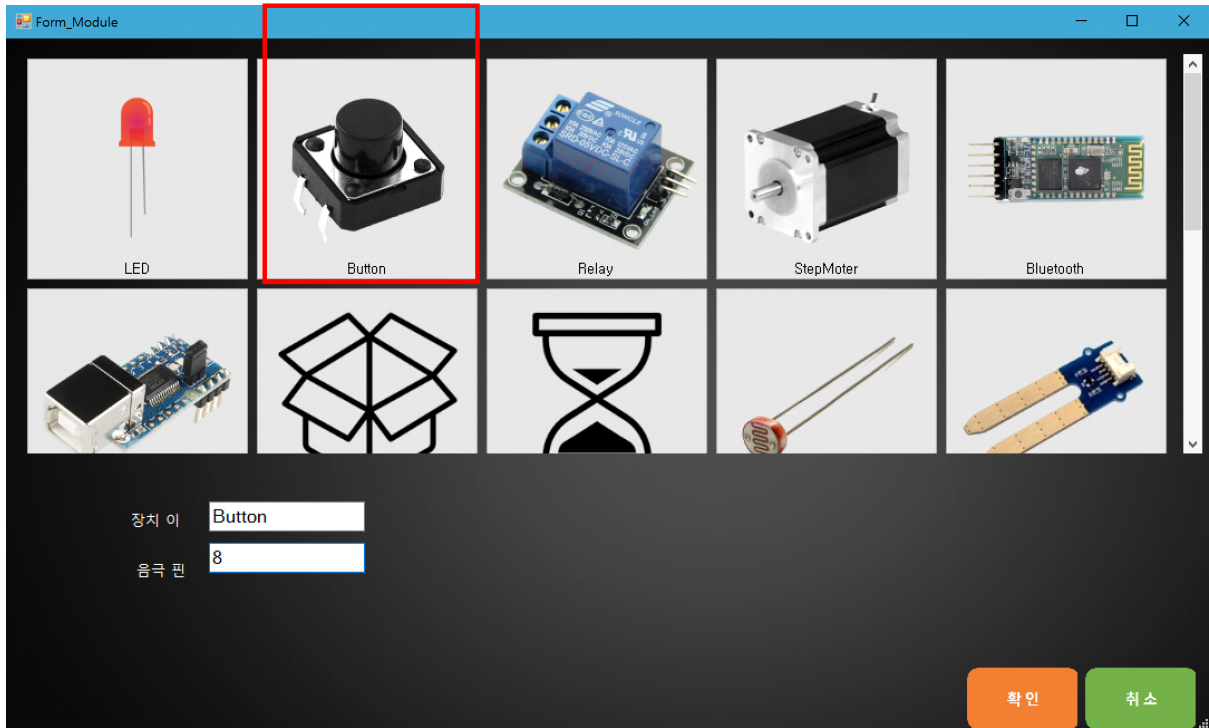
- 이번 실습에서 사용할 아두이노보드를 선택합니다.

Step 2) 모듈 등록



'서보모터 제어하기' 실습 목표를 위해서 사용할 모듈에 'LED'를 등록합니다.

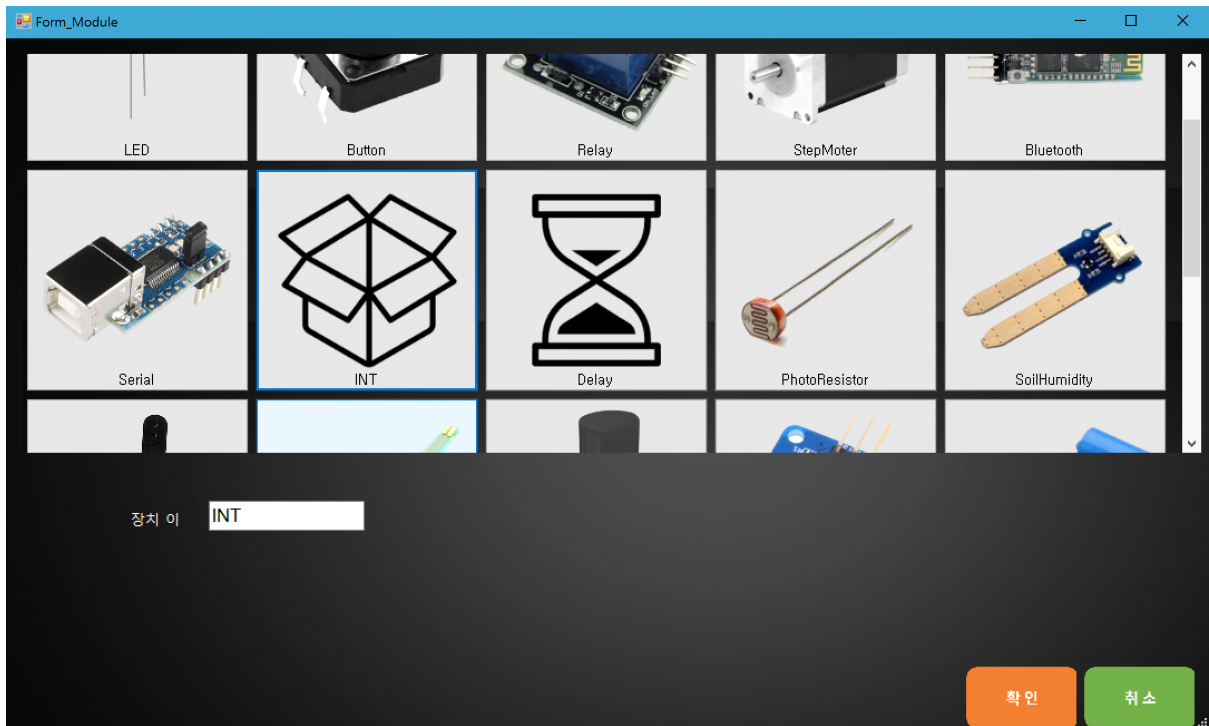
1. 장치 이름은 'Servo'로 입력합니다.
2. 아두이노의 디지털 9번 핀에 연결하였기 때문에 양극 핀은 '9'를 입력합니다.



서보모터를 버튼을 누를 때 회전시키기 위해 버튼 모듈을 추가합니다.

3. 장치 이름은 'Button'으로 합니다.

4. 아두이노의 디지털 8번 핀에 연결하였기 때문에 양극 핀은 '8'를 입력합니다.



각도를 조정하기 위해 그 값을 저장하는 변수 모듈이 필요합니다.

4. 장치 이름은 'INT'로 합니다.

Step 3) 트리거 만들기

동작

Button 모듈이

HIGH 같은 값 일 때

INT 모듈을

10 더하기 하고,

Servo 모듈을

INT 설정 한다.

확인 취소

무엇이 : Button

어떻게 : 눌렀을 때(HIGH)

무엇을 : INT / Servo

어떻게 : INT 값에 10을 더하고 그 값을 Servo의 각도로 설정합니다.

Step 4) 결과 및 업로드

```
#include <Servo.h>
Servo myservo;
int INT = 0;

void setup()
{
  myservo.attach(9,544,2400);
  pinMode(8,INPUT);
}

void loop()
{
  if(digitalRead(8) == HIGH)
  {
    INT += 10;
    myservo.write(INT);
  }
}
```

업로드

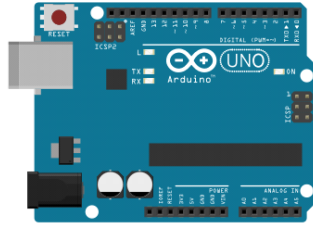
1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.
2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

```
1  #include <Servo.h>
2
3  Servo myservo;
4  int INT = 0;
5
6  void setup()
7  {
8      myservo.attach(9,544,2400);
9      pinMode(8,INPUT);
10
11
12 }
13 void loop()
14 {
15
16     if(digitalRead(8) == HIGH)        //버튼이 HIGH 일 때
17     {
18         INT += 10;                    //INT에 10을 더합니다
19         myservo.write(INT);          //그리고 서보모터를 INT값으로 돌립니다.
20     }
21 }
```

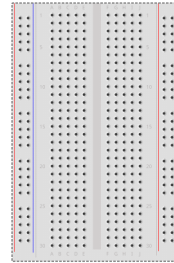
블루투스 통신으로 LED켜기

1

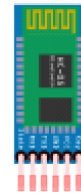
부품 준비하기



아두이노 보드 1개



브레드보드 1개



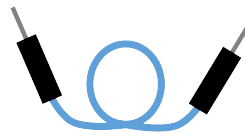
블루투스 모듈(HC-06)



LED 2개



220Ω 저항 2개



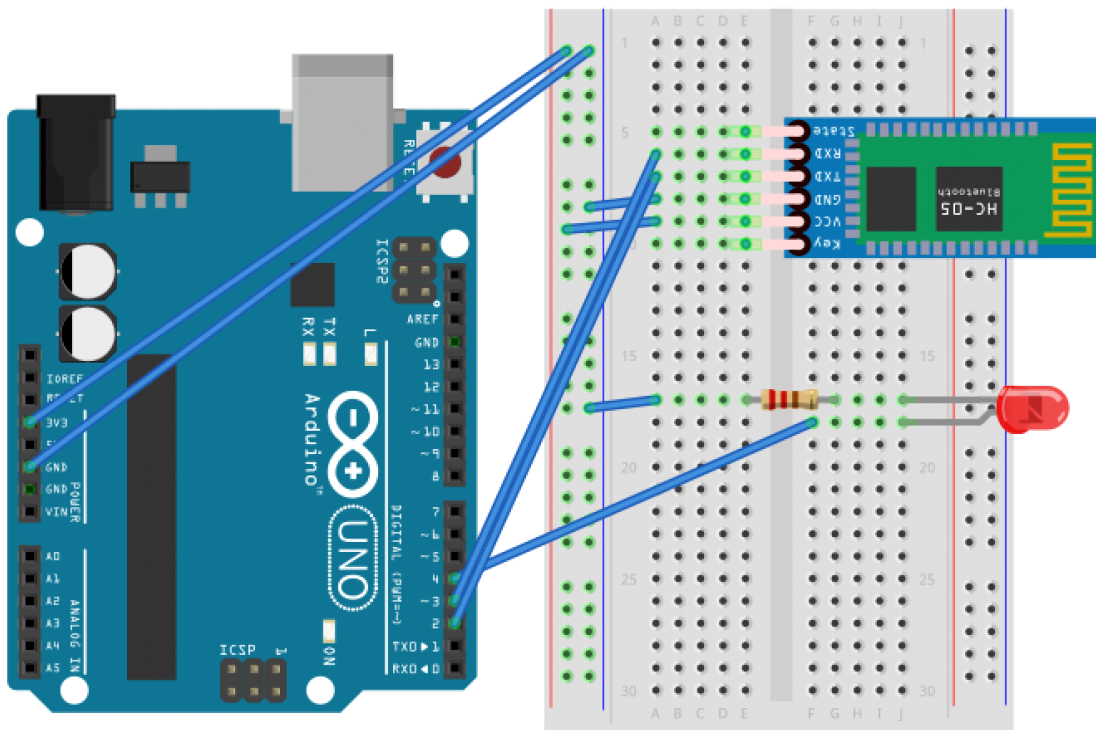
수수 점퍼선 5개



스마트폰

2

회로 구성하기



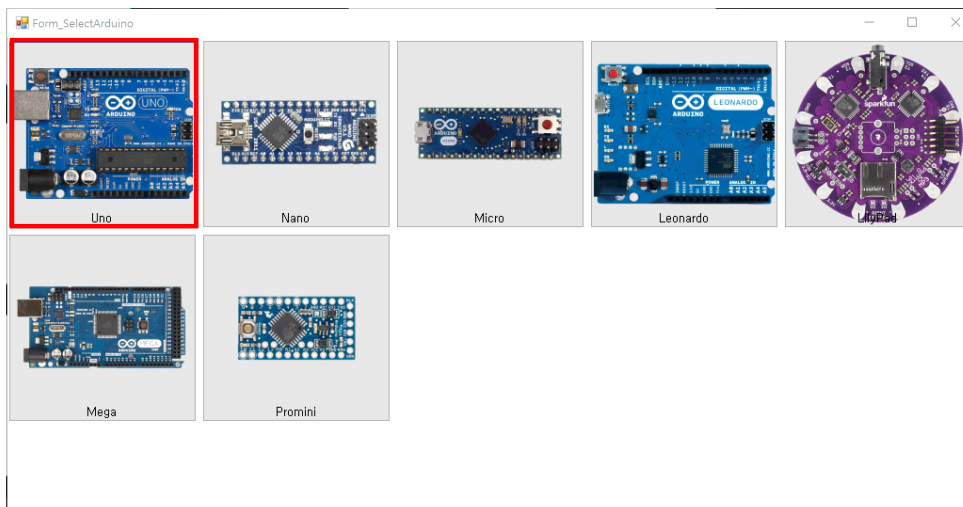
1. 브레드보드에 LED를 연결합니다.
2. LED의 음극(-) 다리에 저항을 연결합니다.

3. 아두이노 보드의 GND핀과 브레드보드의 GND라인을 연결합니다.
4. 아두이노 보드의 3.3V핀과 브레드보드의 VCC라인을 연결합니다.
5. GND라인과 저항을 연결합니다.
6. 아두이노 보드의 4번 핀과 LED의 양극(+) 핀을 연결합니다.
7. 브레드보드에 HC-06 블루투스 모듈을 연결합니다.
8. 블루투스 모듈의 GND핀과 브레드보드의 GND라인을 연결합니다.
9. 블루투스 모듈의 VCC핀과 브레드보드의 VCC라인을 연결합니다.
10. 아두이노의 2번 핀과 블루투스 모듈의 TXD 핀을 연결합니다.
11. 아두이노의 3번 핀과 블루투스 모듈의 RXD 핀을 연결합니다.

3

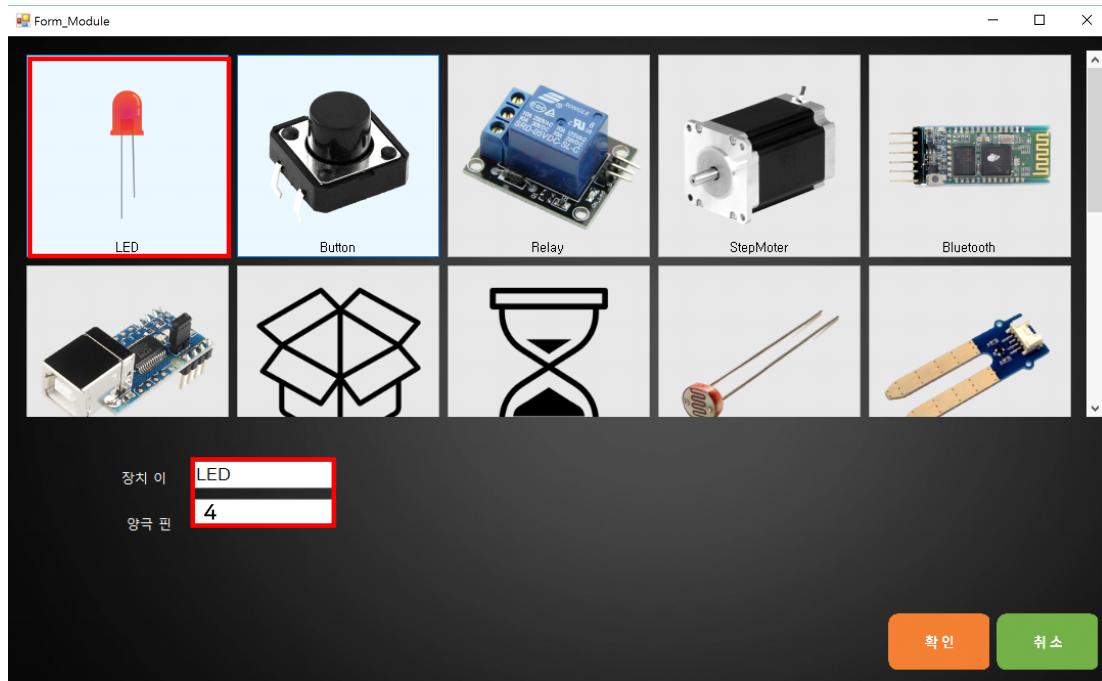
AI Studio

Step 1) 아두이노 선택



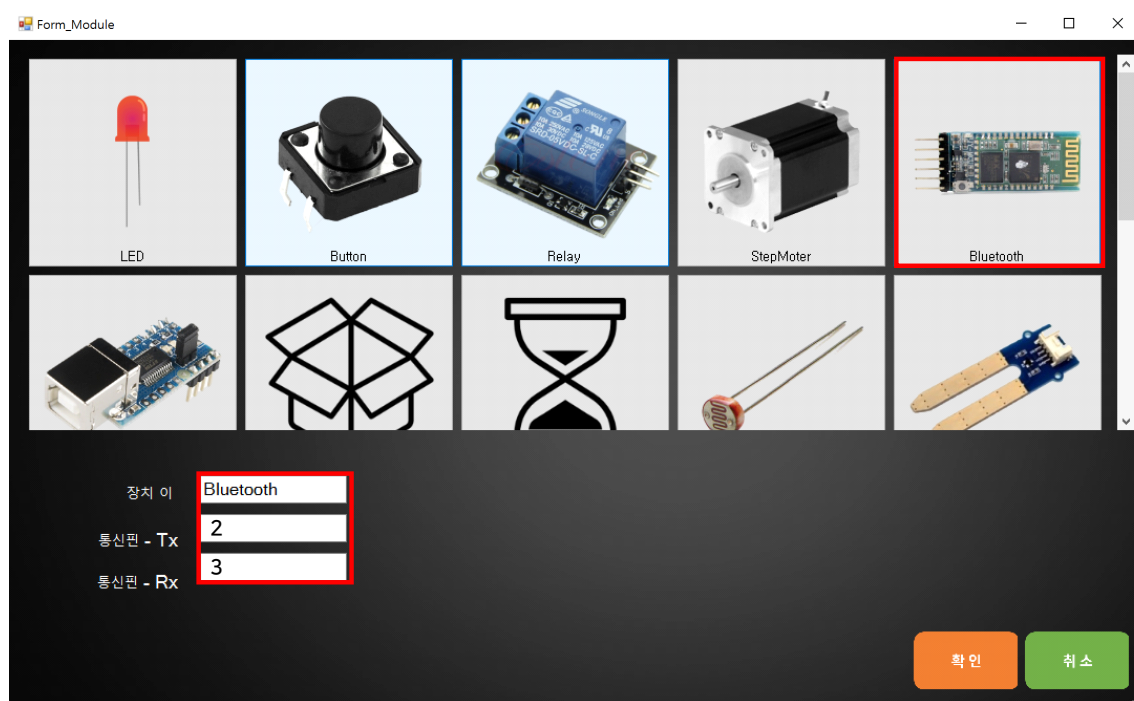
- 이번 실습에서 사용할 아두이노 보드를 선택합니다.

Step 2) 모듈 등록



실습을 위해서 프로그램에 사용할 모듈에 LED를 등록합니다.

1. 장치 이름은 'LED'로 입력합니다.
2. 아두이노의 디지털 4번 핀에 연결하였기 때문에 양극 핀은 '4'를 입력합니다.



실습을 위해서 프로그램에 사용할 모듈에 블루투스 모듈을 등록합니다.

1. 장치 이름은 'Bluetooth'로 입력합니다.
2. 블루투스 핀에 연결한 TXD, RXD 핀 번호에 맞추어 핀 번호를 입력합니다.

Step 3) 트리거 만들기

1. 트리거 명 지정하기

아두이노가 수행할 기능의 이름을 지정해 준다.

2. 조건부 : "누가", "언제" 설정하기

누가 : 블루투스모듈

언제 : 값이 On일때

3. 명령부 : "무엇을", "어떻게" 설정하기

Form_Function

LED켜기

Bluetooth 모듈이

On 같은 값 일 때

LED 모듈을

HIGH 설정 한다.

확인 취소

무엇을 : LED

어떻게 : HIGH(켜기) 설정 합니다.

1. 트리거 명 지정하기

Form_Function

LED 끄기

모듈이

일 때

모듈을

한다.

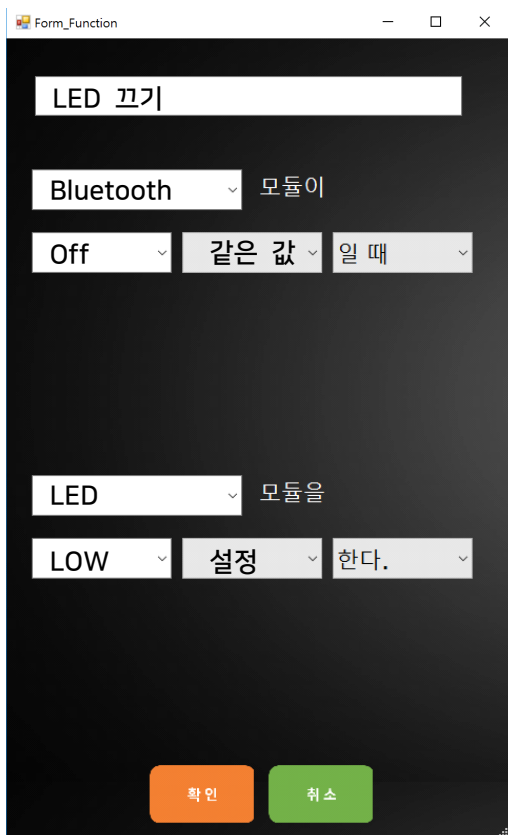
확인 취소

아두이노가 수행할 기능의 이름을 지정해 준다.

2. 조건부 : "누가", "언제" 설정하기



3. 명령부 : “무엇을”, “어떻게” 설정하기



Step 4) 결과 및 업로드

```
#include <SoftwareSerial.h>
SoftwareSerial blueSerial(2, 3);
String blueInputStr;
void setup()
{
    blueSerial.begin(9600);
    pinMode(4,OUTPUT);
}
void loop()
{
    while(blueSerial.available())
    {
        char blueChar =
(char)blueSerial.read();
        if(blueChar == '#')
        {
            blueInputStr = "";
            return;
        }
        blueInputStr+=blueChar;
    }
    //LED켜기
    if(blueInputStr.equals("On" )
```

```
{
    digitalWrite(4,HIGH);
}
//LED끄기
if(blueInputStr.equals("Off" )
{
    digitalWrite(4,LOW);
}
}
```

1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.
2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

4

결과 소스코드

```
1 #include <SoftwareSerial.h>
2
3 SoftwareSerial blueSerial(2, 3);
4 String blueInputStr;
5
6 void setup()
7 {
8     blueSerial.begin(9600);
9     pinMode(4,OUTPUT);
10 }
11 void loop()
12 {
13     while(blueSerial.available())
14     {
15         char blueChar = (char)blueSerial.read();
16         if(blueChar == '#')
17         {
18             blueInputStr = "";
19             return;
20         }
21         blueInputStr+=blueChar;
22     }
23     //LED켜기
24     if(blueInputStr.equals("On" )
```

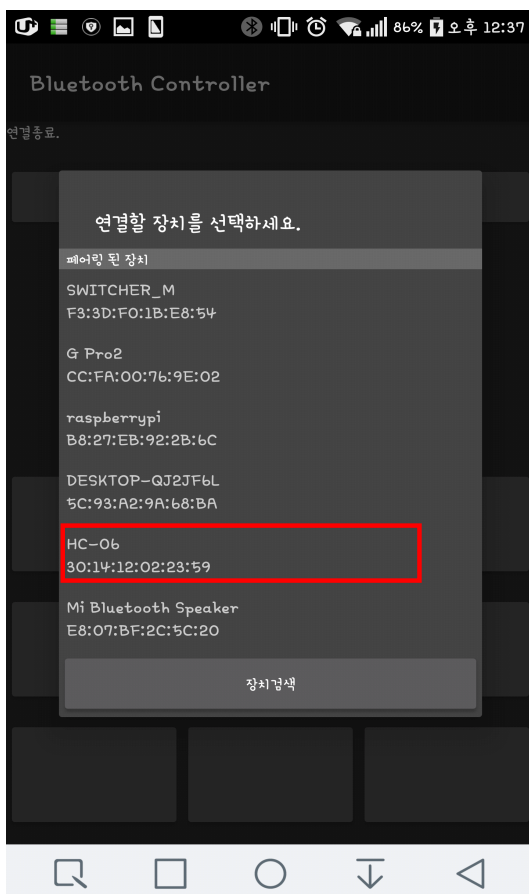
```

17         }
18         blueInputStr+=blueChar;
19     }
20     //LED켜기
21     if(blueInputStr.equals("On") )
22     {
23         digitalWrite(4,HIGH);
24     }
25     //LED끄기
26     if(blueInputStr.equals("Off") )
27     {
28         digitalWrite(4,LOW);
29     }

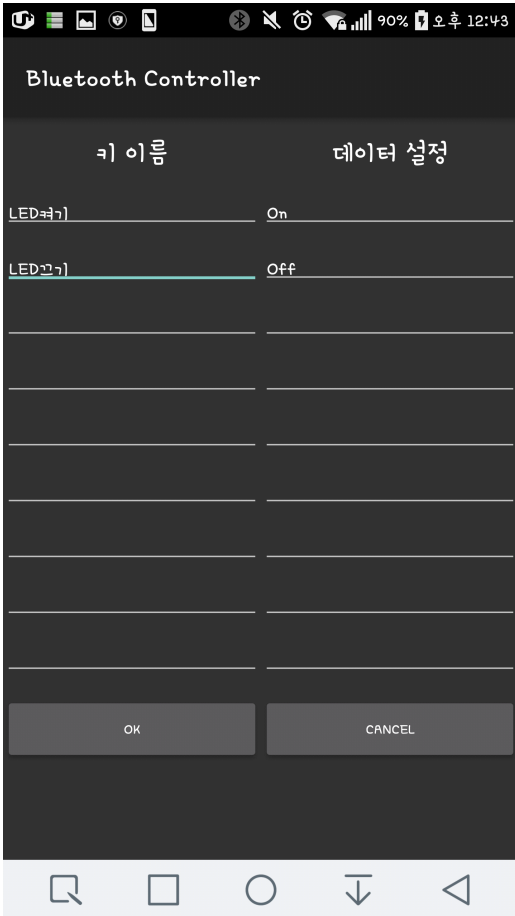
```

5

어플리케이션 설정하기



설치한 블루투스 어플리케이션을 실행하고 아두이노에 연결한 블루투스 모듈(HC-06)과 페어링합니다.



페어링 한 후 설정 버튼을 눌러 모듈에 보낼 데이터를 설정합니다.

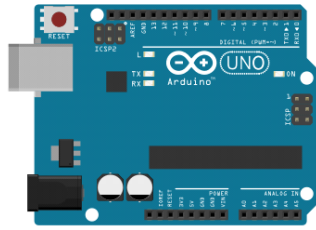
여기서 키 이름은 버튼의 이름이고 데이터 설정은 보낼 데이터입니다.



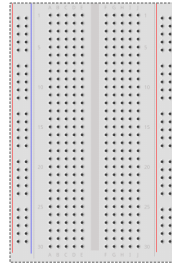
설정이 완료되었다면 버튼을 누른 후 아두이노의 동작을 관찰합니다.

블루투스 통신으로 서보모터 제어하기

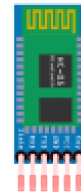
1 부품 준비하기



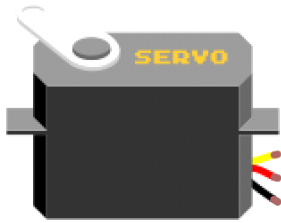
아두이노 보드 1개



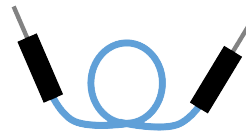
브레드보드 1개



블루투스 모듈(HC-06)



서보모터

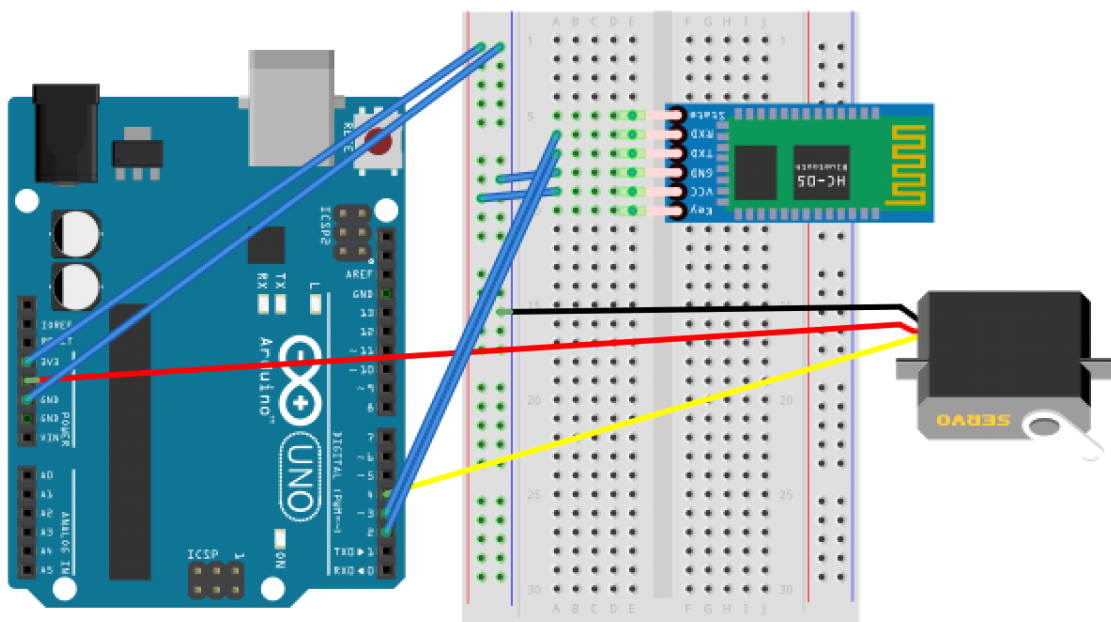


수수 점퍼선 5개



스마트폰

2 회로 구성하기

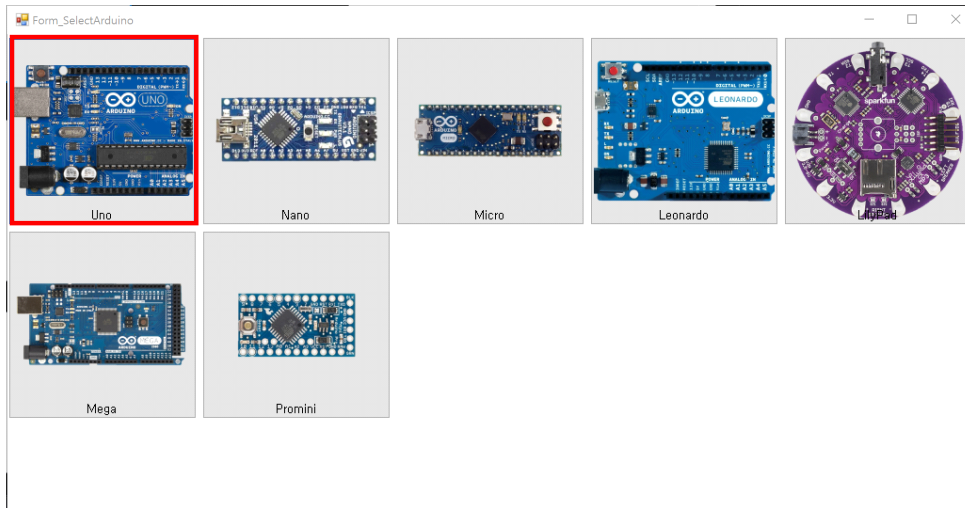


1. 브레드보드에 서보모터를 연결합니다.

2. 아두이노 보드의 GND핀과 브레드보드의 GND라인을 연결합니다.
3. 아두이노 보드의 3.3V핀과 브레드보드의 VCC라인을 연결합니다.
4. 서보모터의 GND선과 브레드보드의 GND라인을 연결합니다.
5. 서보모터의 데이터 선과 아두이노 보드의 4번 핀을 연결합니다.
6. 서보모터의 VCC선과 아두이노의 5V 핀을 연결합니다.
7. 브레드보드에 HC-06 블루투스 모듈을 연결합니다.
8. 블루투스 모듈의 GND핀과 브레드보드의 GND라인을 연결합니다.
9. 블루투스 모듈의 VCC핀과 브레드보드의 VCC라인을 연결합니다.
10. 아두이노의 2번 핀과 블루투스 모듈의 TXD 핀을 연결합니다.
11. 아두이노의 3번 핀과 블루투스 모듈의 RXD 핀을 연결합니다.

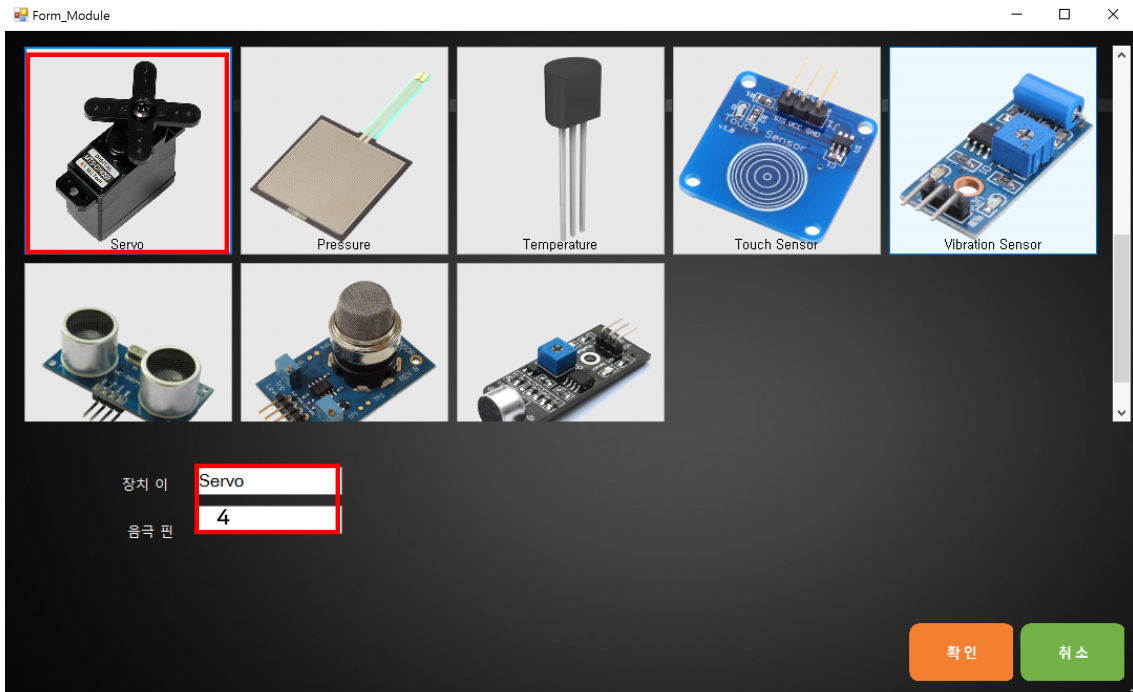
3 AI Studio

Step 1) 아두이노 선택



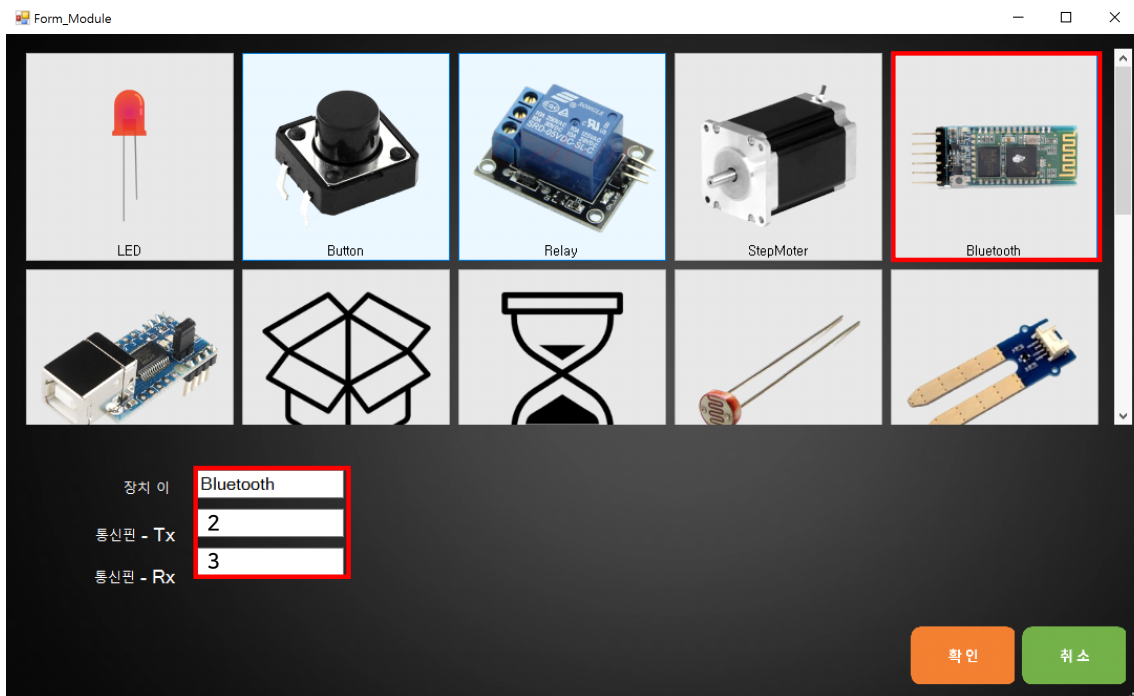
- 이번 실습에서 사용할 아두이노 보드를 선택합니다.

Step 2) 모듈 등록



실습을 위해서 프로그램에 사용할 모듈에 서보모터를 등록합니다.

1. 장치 이름은 'Servo'로 입력합니다.
2. 아두이노의 디지털 4번 핀에 연결하였기 때문에 데이터 핀은 '4'를 입력합니다.



실습을 위해서 프로그램에 사용할 모듈에 블루투스 모듈을 등록합니다.

1. 장치 이름은 'Bluetooth'로 입력합니다.
2. 블루투스 핀에 연결한 TXD, RXD 핀 번호에 맞추어 핀 번호를 입력합니다.

Step 3) 트리거 만들기

1. 트리거 명 지정하기

아두이노가 수행할 기능의 이름을 지정해 준다.

2. 조건부 : "누가", "언제" 설정하기

누가 : 블루투스모듈

언제 : 값이 On일때

3. 명령부 : "무엇을", "어떻게" 설정하기

무엇을 : LED

어떻게 : HIGH(켜기) 설정 합니다.

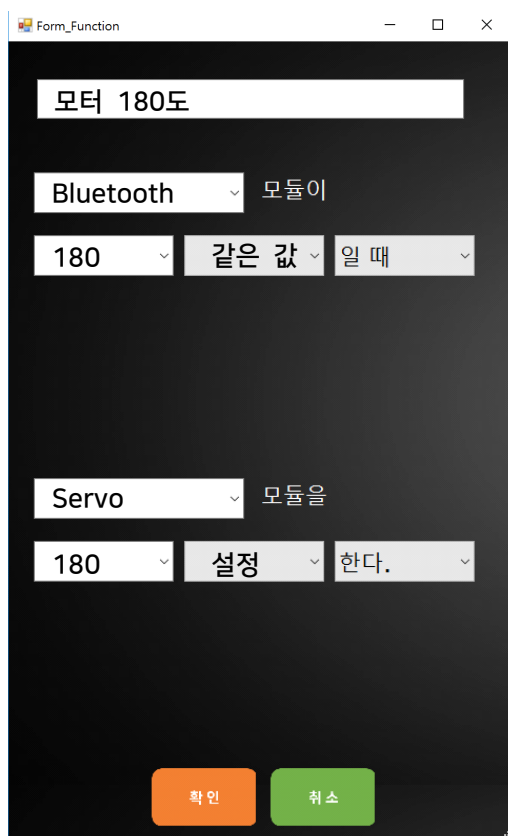
1. 트리거 명 지정하기

아두이노가 수행할 기능의 이름을 지정해 준다.

2. 조건부 : "누가", "언제" 설정하기



3. 명령부 : “무엇을”, “어떻게” 설정하기



Step 4) 결과 및 업로드

```

#include <SoftwareSerial.h>
#include <Servo.h>

SoftwareSerial blueSerial(2, 3);
String blueInputStr;
Servo myservo;

void setup()
{
    blueSerial.begin(9600);
    myservo.attach(4,544,2400);
}
void loop()
{
    while(blueSerial.available())
    {
        char blueChar =
(char)blueSerial.read();
        if(blueChar == '#')
        {
            blueInputStr = "";
            return;
        }
    }
}

```

업로드

```

}
        blueInputStr+=blueChar;
    }
    //모터 90도
    if(blueInputStr.equals("90") )
    {
        myservo.write(90);
    }
    //모터 180도
    if(blueInputStr.equals("180") )
    {
        myservo.write(180);
    }
}

```

업로드

1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.
2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

4

결과 소스코드

```

1  #include <SoftwareSerial.h>
2  #include <Servo.h>
3
4  SoftwareSerial blueSerial(2, 3);
5  String blueInputStr;
6  Servo myservo;
7
8  void setup()
9  {
10     blueSerial.begin(9600);
11     myservo.attach(4,544,2400);
12
13 }
14 void loop()
15 {
16     while(blueSerial.available())
17     {
18         char blueChar = (char)blueSerial.read();

```

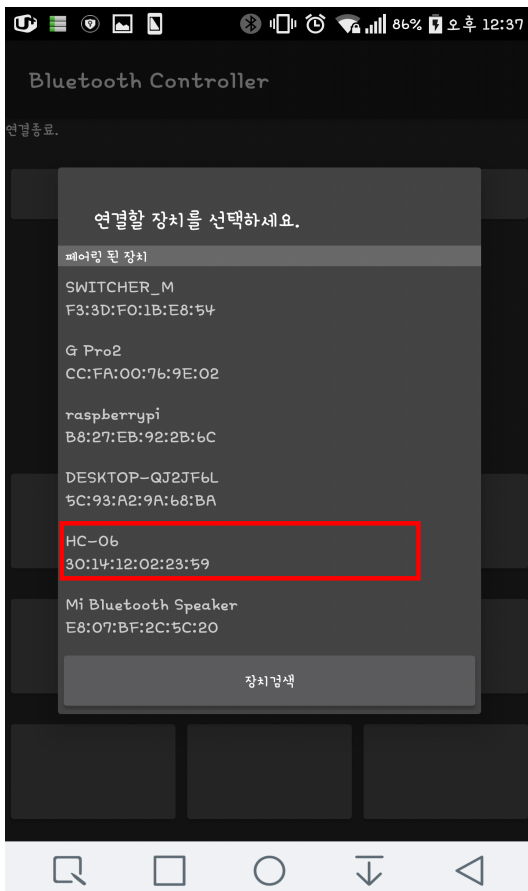
```

19         if(blueChar == '#')
20         {
21             blueInputStr = "";
22             return;
23         }
24         blueInputStr+=blueChar;
25     }
26     //모터 90도
27     if(blueInputStr.equals("90") )
28     {
29         myservo.write(90);
30     }
31     //모터 180도
32     if(blueInputStr.equals("180") )
33     {
34         myservo.write(180);
35     }
36 }

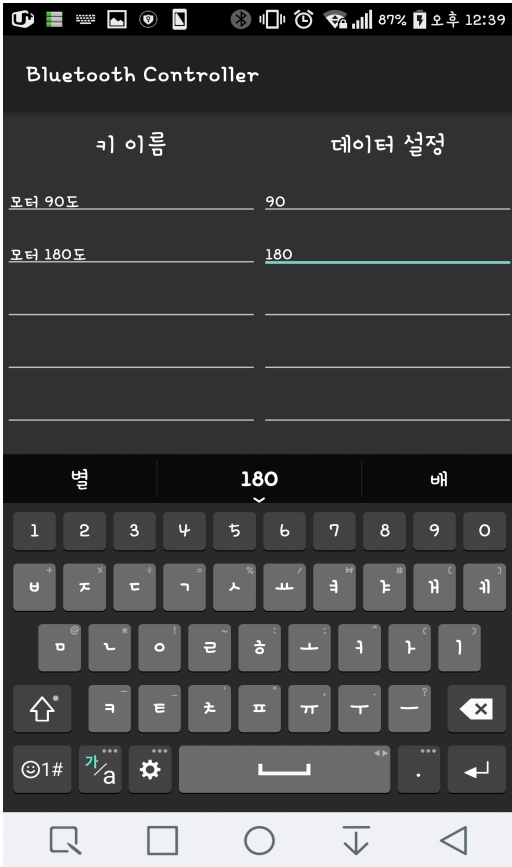
```

5

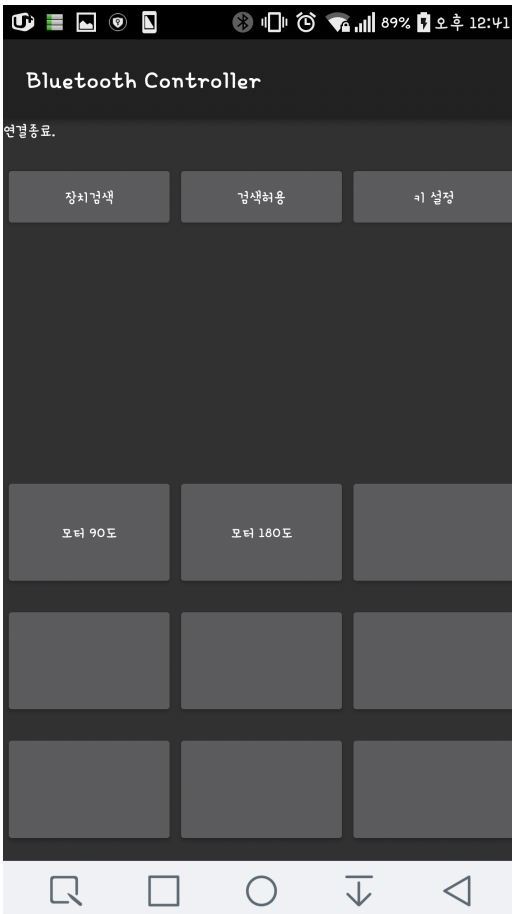
어플리케이션 설정하기



설치한 블루투스 어플리케이션을 실행하고 아두이노에 연결한 블루투스 모듈(HC-06)과 페어링합니다.



페어링 한 후 설정 버튼을 눌러 모듈에 보낼 데이터를 설정합니다.
 여기서 키 이름은 버튼의 이름이고 데이터 설정은 보낼 데이터입니다.

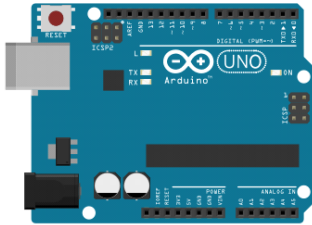


설정이 완료되었다면 버튼을 누른 후 아두이노의 동작을 관찰합니다.

현재 온/습도 출력하기

1

부품 준비하기



아두이노 보드 1개



DHT11(온습도센서)1개



5kΩ 저항 1개

2

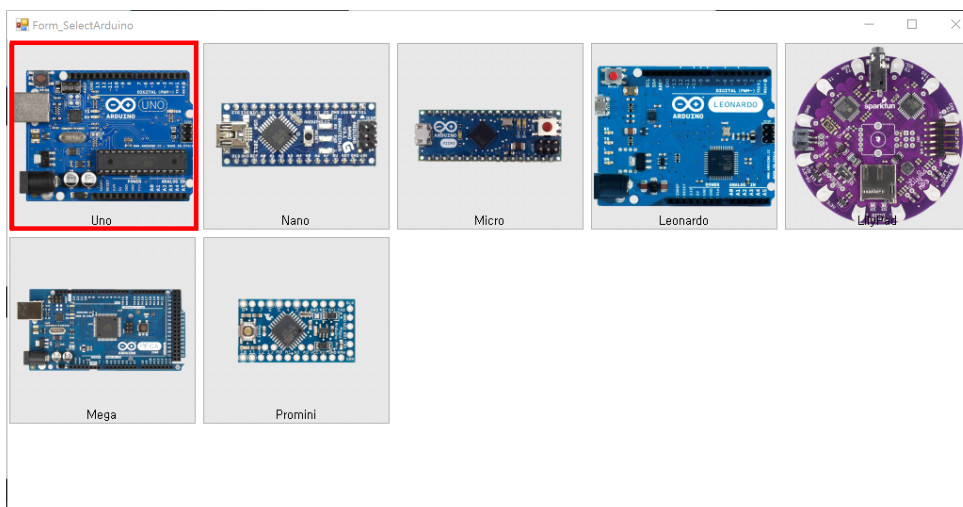
회로 구성하기

1. 온/습도센서의 4번째핀(GND)을 아두이노보드의 GND와연결합니다.
2. 온/습도센서의 2번째핀을 아두이노 보드의 2번째핀과 연결합니다.
3. 온/습도센서의 1번째핀(Vcc)을 아두이노 보드의 5V와 연결합니다.
- 4.Vcc로 연결된 선과 데이터로 연결된 선을 5kΩ저항으로연결합니다.

3

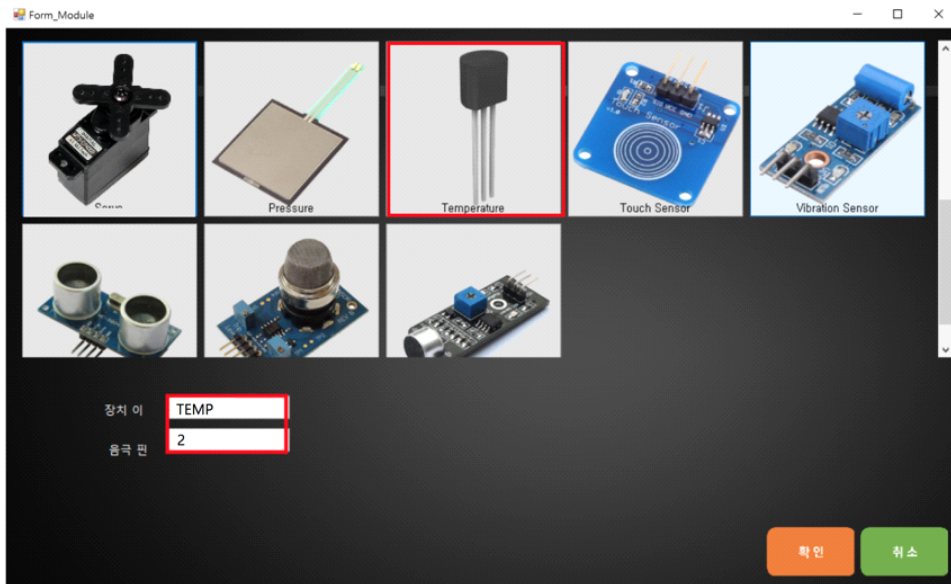
AI Studio

Step 1) 아두이노 선택



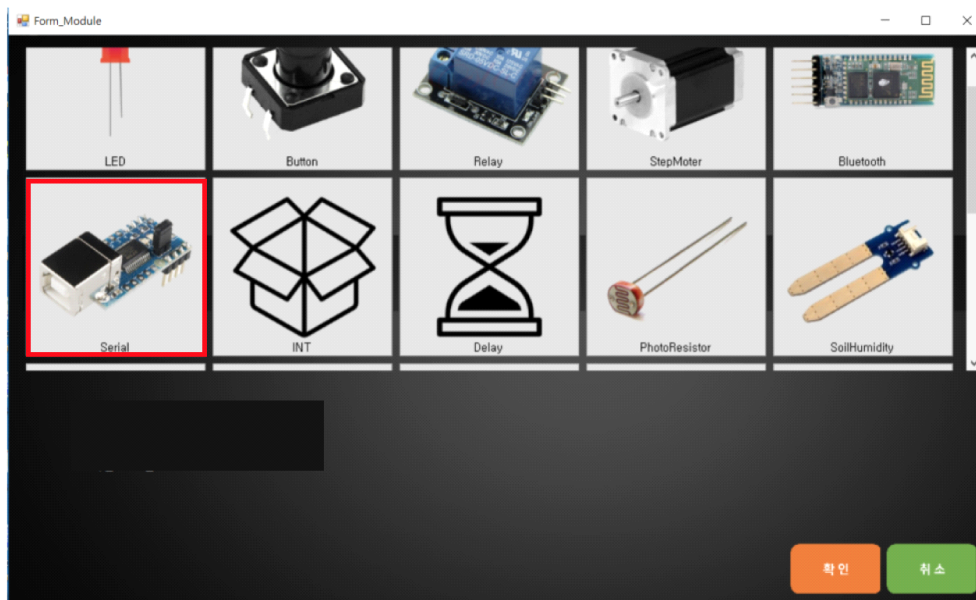
- 이번 실습에서 사용할 아두이노보드를 선택합니다.

Step 2) 모듈 등록



실습을 위해서 프로그램에 사용할 모듈에 온/습도센서를 등록합니다.

1. 장치 이름은 '온습도'로 입력합니다.
2. 아두이노의 디지털 2번 핀에 연결하였기 때문에 데이터 핀은 '2'를 입력합니다.



실행시, 시리얼창에 온/습도값을 출력하기 위해, 프로그램에 사용할 모듈에 시리얼을 등록합니다.

1. 장치 이름은 '시리얼'로 입력합니다.

Step 3) 트리거 만들기

- 조건부 : “누가”, “언제” 설정하기
:온습도센서에선 조건부가 존재하지 않으므로 생략한다.
- 명령부 : “무엇을”, “어떻게” 설정하기

Form_Function

온습도

모듈이

일 때

시리얼 모듈을

온습도 설정 한다.

확인 취소

아두이노가 수행할 기능의 이름을 지정해 준다.

무엇을 : 시리얼

어떻게 : 온습도 값으로 설정 한다.

Step 4) 결과 및 업로드

```
#include <DHT11.h>
int pin=2;
DHT11 dht11(pin);
void setup()
{
  Serial.begin(9600); //통신속도 설정
}
void loop()
{
  int err;
  float temp, humi;
  if((err=dht11.read(humi, temp))==0)
  {
    Serial.print(temp);
    Serial.print(humi);
  }
}
```



1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.
2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

4

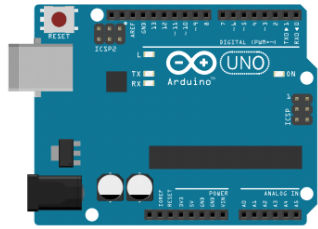
결과 소스코드

```
1  #include <DHT11.h>
2  int pin=2;
3  DHT11 dht11(pin);
4  void setup()
5  {
6      Serial.begin(9600); //통신속도 설정
7  }
8  void loop()
9  {
10     int err;
11     float temp, humi;
12     if((err=dht11.read(humi, temp))==0)
13     {
14         Serial.print(temp);
15         Serial.print(humi);
16     }
```

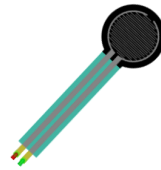
압력감출력하기

1

부품 준비하기



아두이노 보드 1개



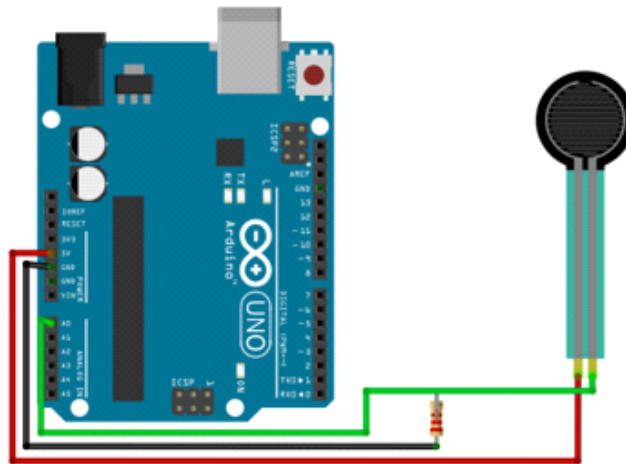
FSR 센서 1개



3.3kΩ 저항 1개

2

회로 구성하기

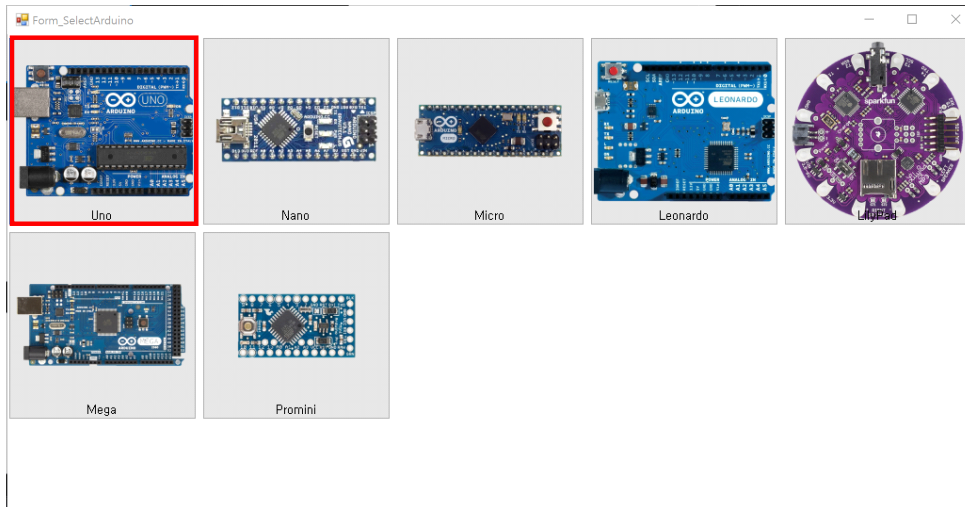


1. FSR센서의 한개의 핀을 아두이노 보드의 5V와연결합니다.
2. FSR센서의 나머지 핀은 아두이노 보드의 A0와연결합니다.
3. A0로 연결된 선과 아두이노 보드의 GND를 3.3kΩ저항으로 연결합니다.

3

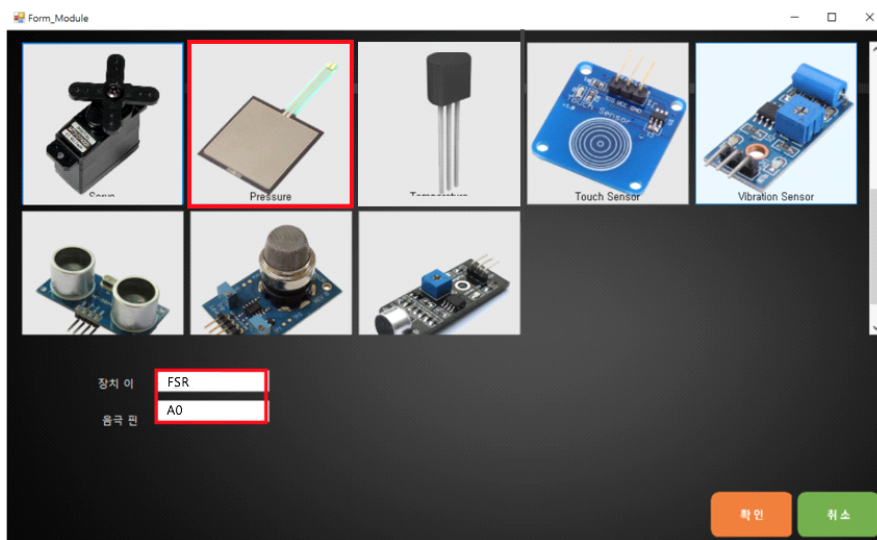
AI Studio

Step 1) 아두이노 선택



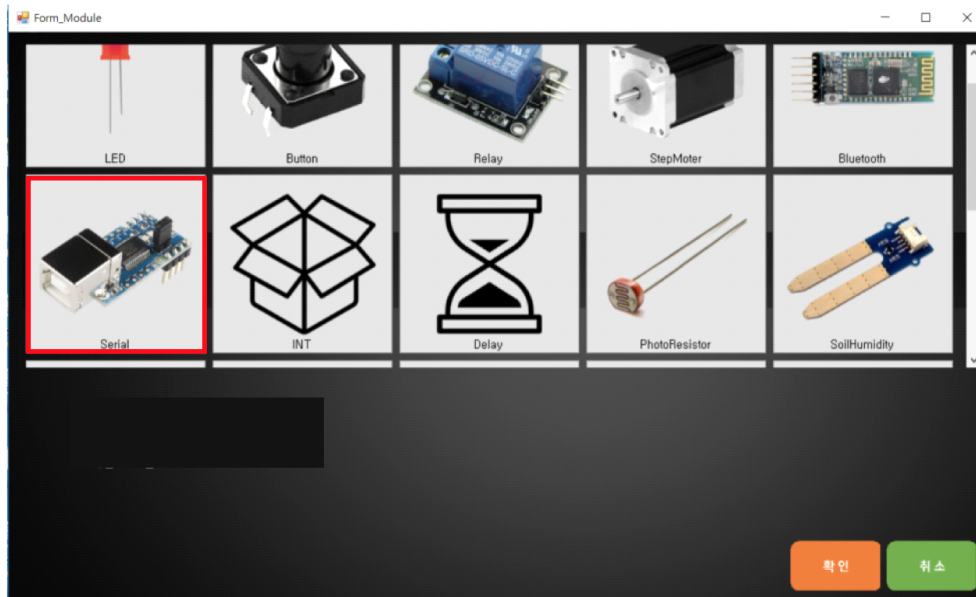
- 이번 실습에서 사용할 아두이노보드를 선택합니다.

Step 2) 모듈 등록



실습을 위해서 프로그램에 사용할 모듈에 FSR센서를 등록합니다.

1. 장치 이름은 'FSR'으로 입력합니다.
2. 아두이노의 아날로그 0번 핀에 연결하였기 때문에 데이터 핀은 'A0'를 입력합니다.



실행시, 시리얼창에 FSR값을 출력하기 위해, 프로그램에 사용할 모듈에 시리얼을 등록합니다.

1. 장치 이름은 '시리얼'로 입력합니다.

Step 3) 트리거 만들기

2. 조건부 : “누가”, “언제” 설정하기

:압력센서에선 조건부가 존재하지 않으므로 생략한다.

3. 명령부 : “무엇을”, “어떻게” 설정하기



Step 4) 결과 및 업로드

```
int LED = 3;
void setup() {
    Serial.begin(9600);
}
void loop() {
    int sensorval = analogRead(A0);
    int brightness = map(sensorval, 0
, 1023, 0, 255);
    analogWrite(LED,brightness);
    Serial .println("Sensorvalue : "
+String(sensorval) + ", LED Output : " +
String(brightness));
}
```



업로드

1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.
2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

3

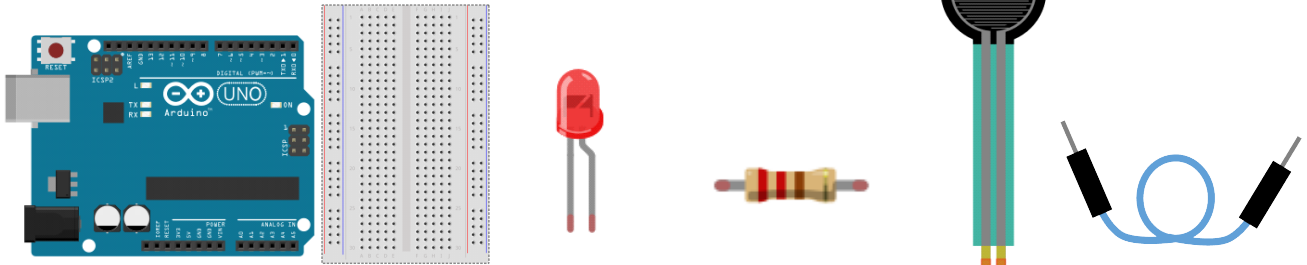
결과 소스코드

```
1 int LED = 3;
2 void setup() {
3     Serial.begin(9600);
4 }
5 void loop() {
6     int sensorval = analogRead(A0);
7     int brightness = map(sensorval, 0 , 1023, 0, 255);
8     analogWrite(LED,brightness);
9     Serial .println("Sensorvalue : " +String(sensorval) + ", LED Output :
10 " + String(brightness));
11 }
```

FSR 센싱하기

1

부품 준비하기



아두이노 보드 1개

브레드보드 1개

LED 1개

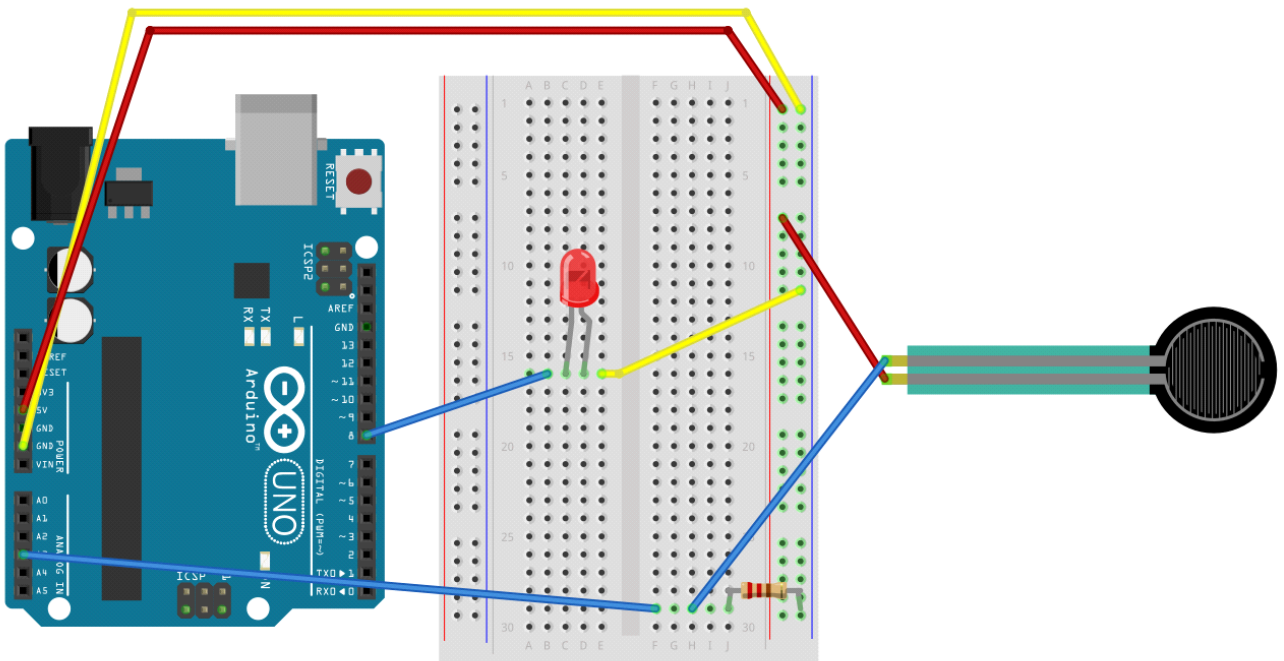
220Ω 1개

FSR 1개

수수 점퍼선 5개

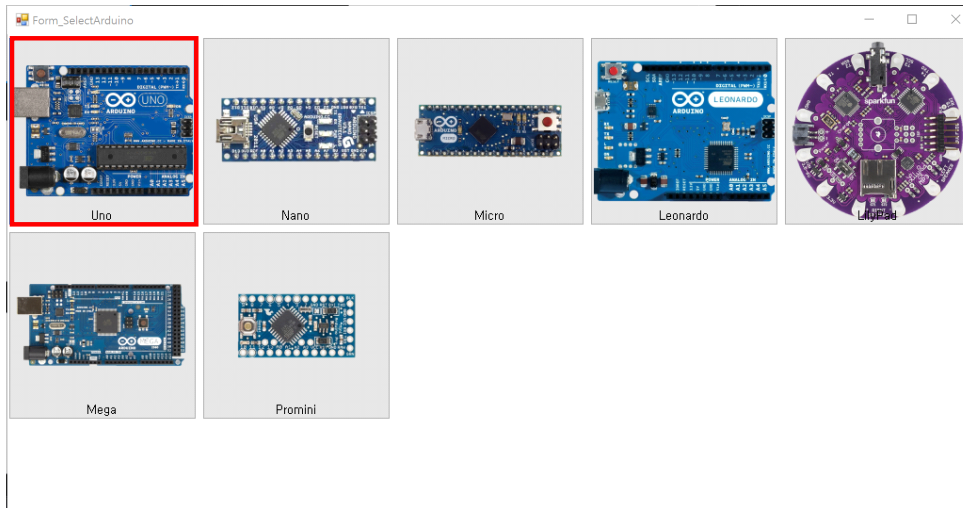
2

회로 구성하기



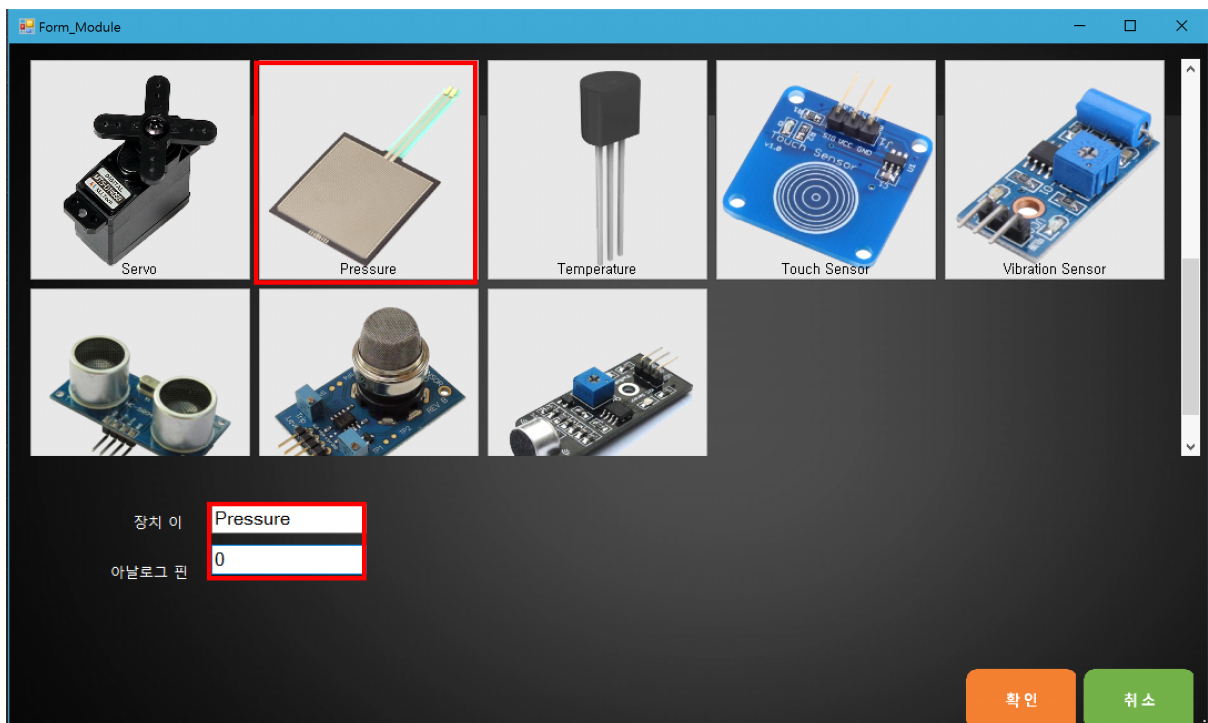
다음과 같이 연결합니다.

Step 1) 아두이노 선택



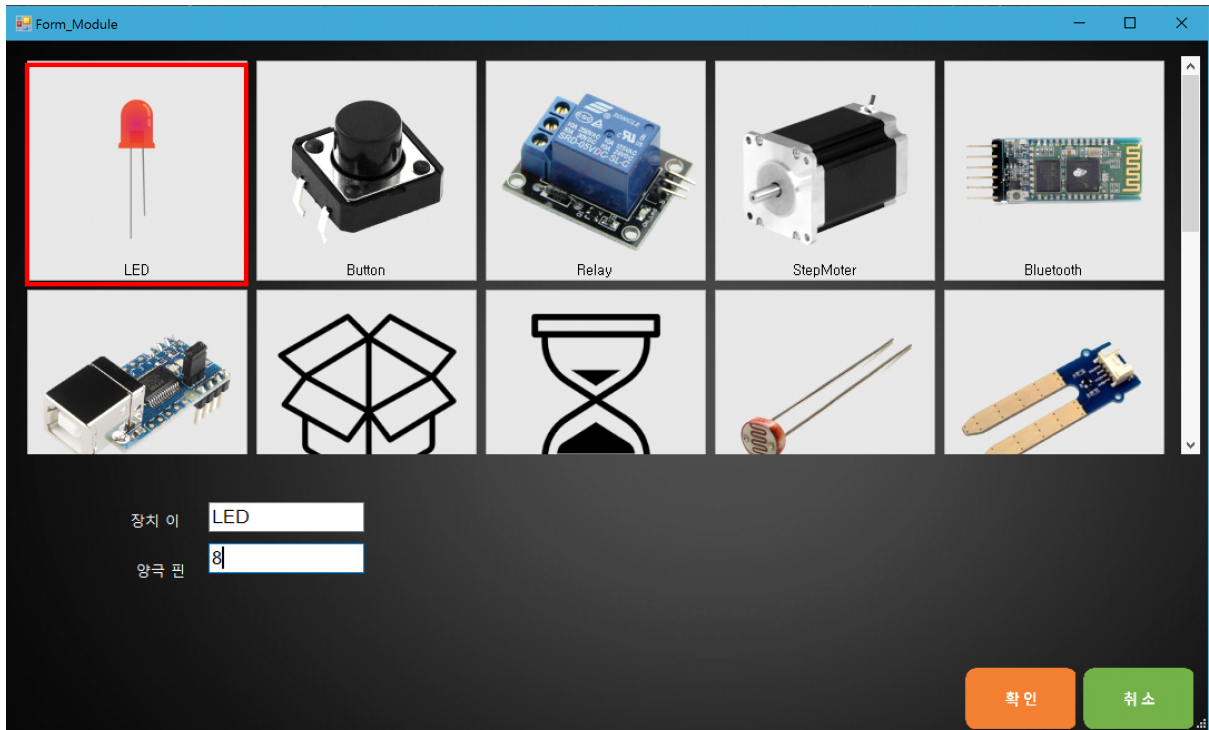
- 이번 실습에서 사용할 아두이노보드를 선택합니다.

Step 2) 모듈 등록



'FSR 센싱하기' 실습 목표를 위해서 사용할 모듈 Pressure를 등록합니다.

1. 장치 이름은 'Pressure'로 입력합니다.
2. 아두이노의 아날로그 0번 핀에 연결하였기 때문에 핀은 '0'를 입력합니다.



압력 감지가 되면 LED를 켜기 위해 사용할 모듈 LED를 등록합니다.

3. 장치 이름은 'LED'로 입력합니다.

4. 아두이노의 8번 핀에 연결하였기 때문에 양극 핀은 '8'를 입력합니다.

Step 3) 트리거 만들기

1. 트리거 명 지정하기
2. 조건부 : "누가", "언제" 설정하기
: Pressure가 50 이상이면
3. 명령부 : "무엇을", "어떻게" 설정하기
: LED를 HIGH 한다

아두이노가 수행할 기능의 이름을 지정해 준다.

동작

Pressure 모듈이

50 이상인 값 일 때

LED 모듈을

HIGH 설정 한다.

확인 취소



Step 4) 결과 및 업로드

```

void setup()
{
    int Pressure = A0;
    int LED = 8;
}
void loop()
{
    if(analogRead(Pressure)>=50){
        digitalWrite(LED, HIGH);
    }
}

```

업로드

1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.

2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

4

결과 소스코드

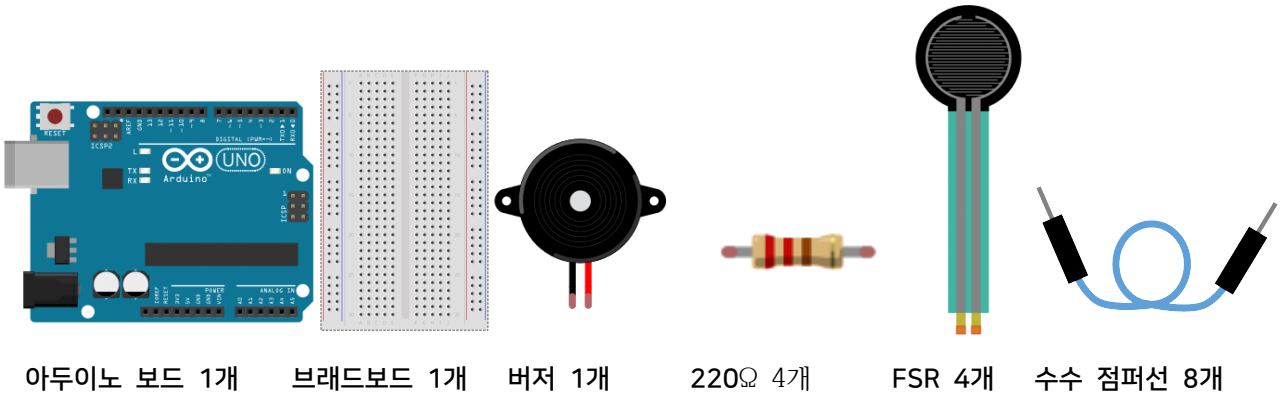
```
1 void setup()
2 {
3     int Pressure = A0; //압력 센서의 핀 번호입니다
4     int LED = 8;
5
6 }
7 void loop()
8 {
9
10     if(analogRead(Pressure)>=50){
11         digitalWrite(LED, HIGH);
12     }
13 }
```

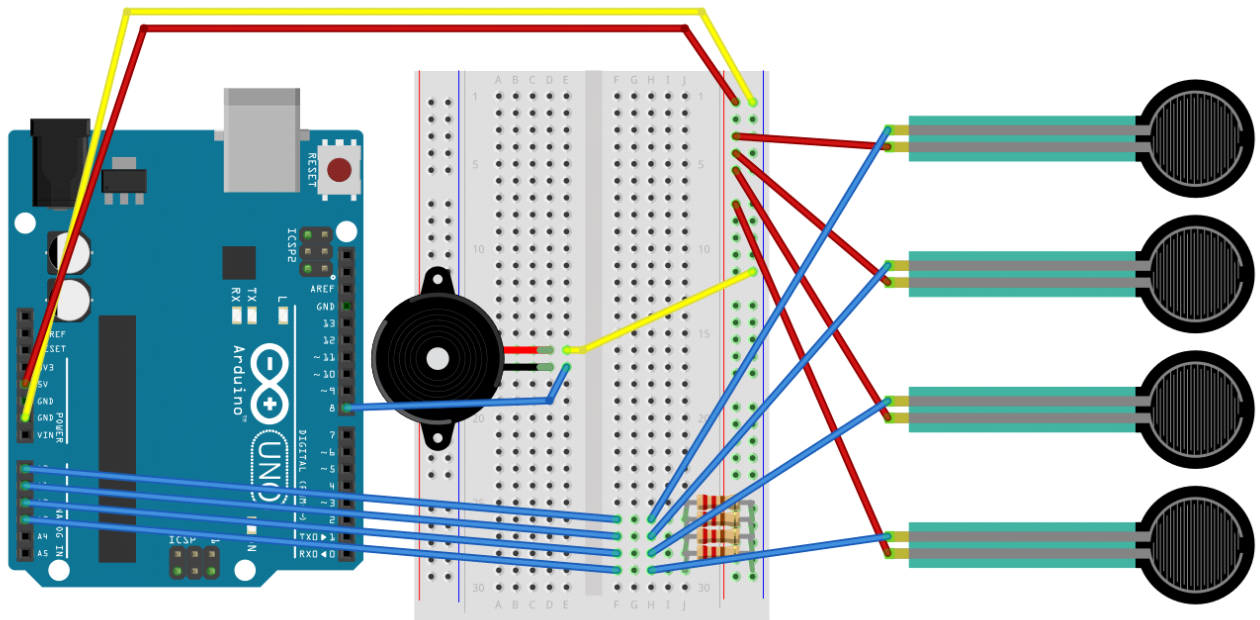
FSR 센서로 앉은 자세 기울기 감지하기

1 개요

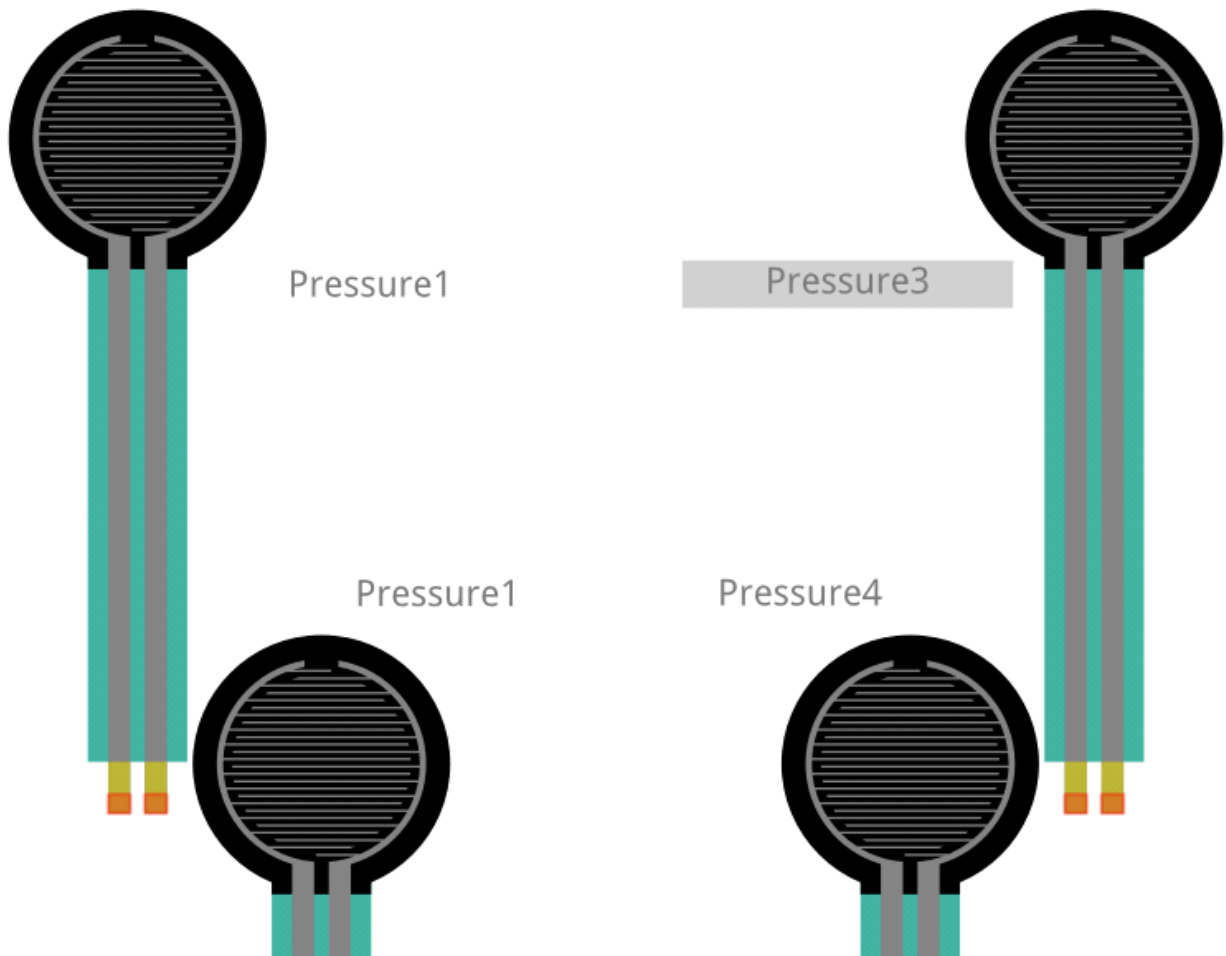
4개의 압력센서로 간단하게 자세감지 도구를 만들 수 있습니다. 이번 실습에서는 앉은 자세의 기울기가 일정 값 이상 차이가 날 때 경보를 울리는 장치를 실습해 봅시다.

2 부품 준비하기



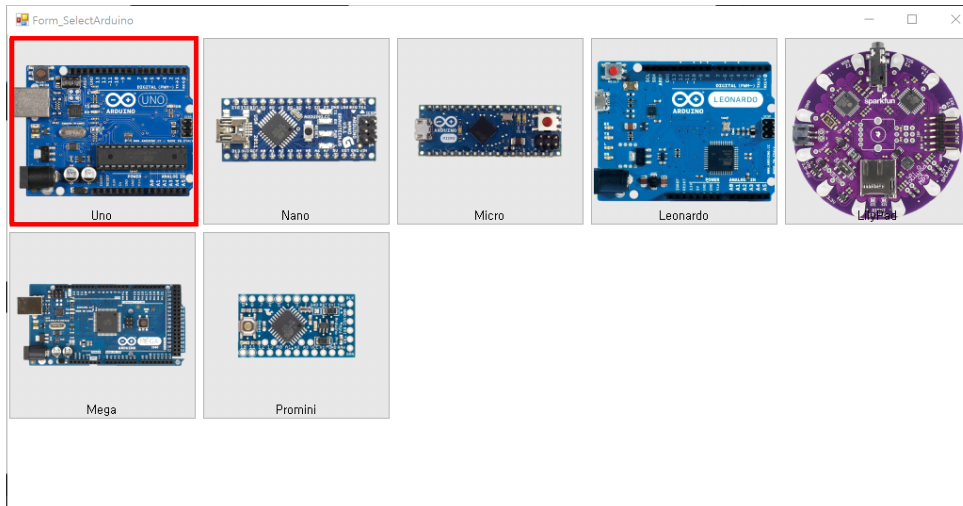


다음과 같이 연결합니다.



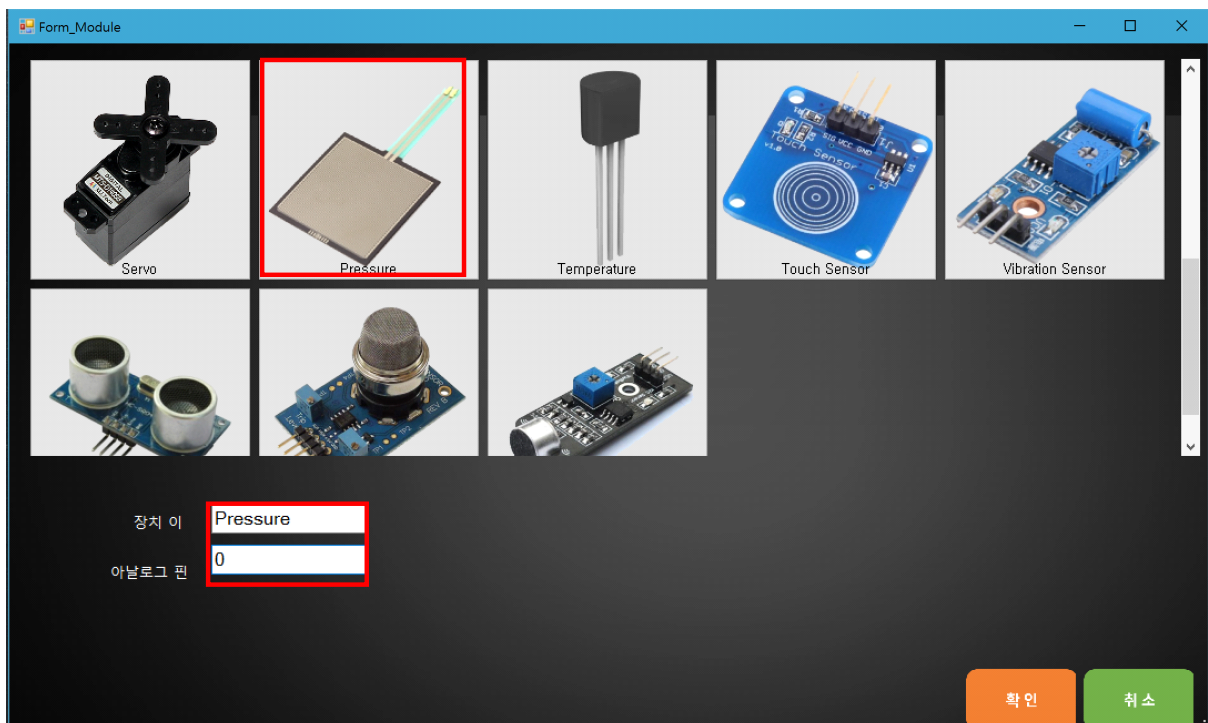
센서의 위치를 다음과 같이 배치합니다.

Step 1) 아두이노 선택



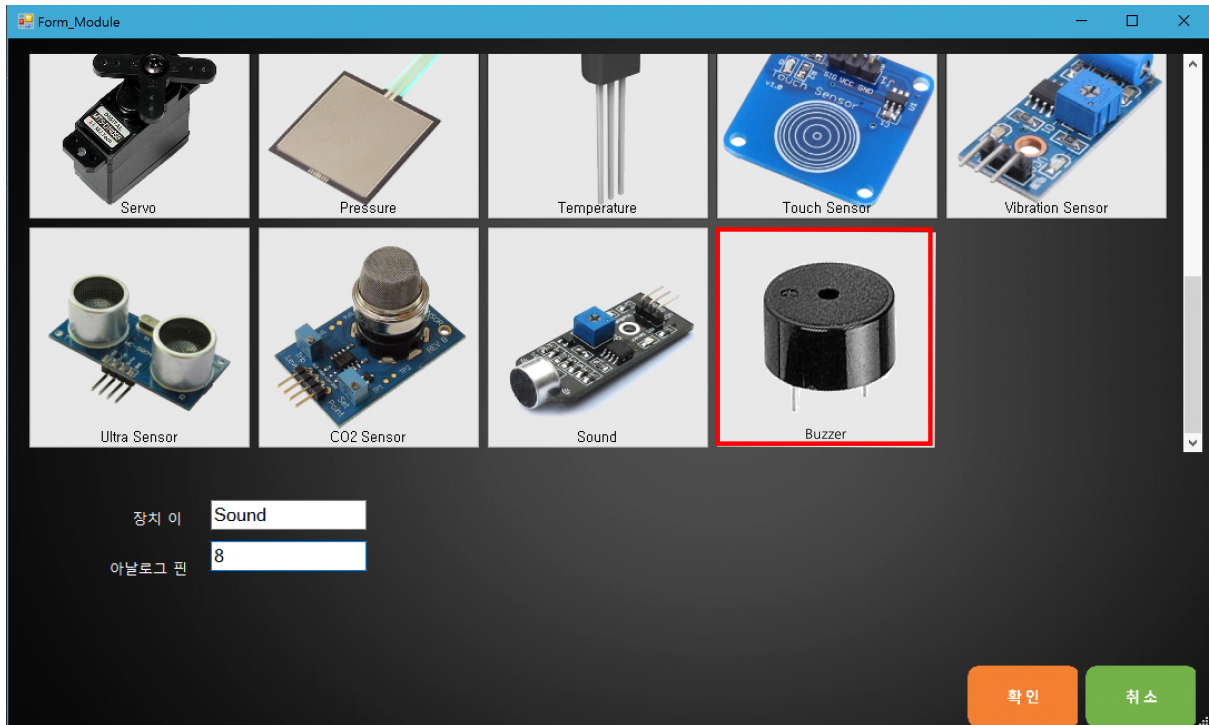
- 이번 실습에서 사용할 아두이노보드를 선택합니다.

Step 2) 모듈 등록



FSR 센서 앓은 자세의 기울기를 검출하려면 좌우 엉덩이, 허벅지의 압력을 센싱해야 합니다. 이를 위해서 사용할 모듈 Pressure를 4개 등록합니다.

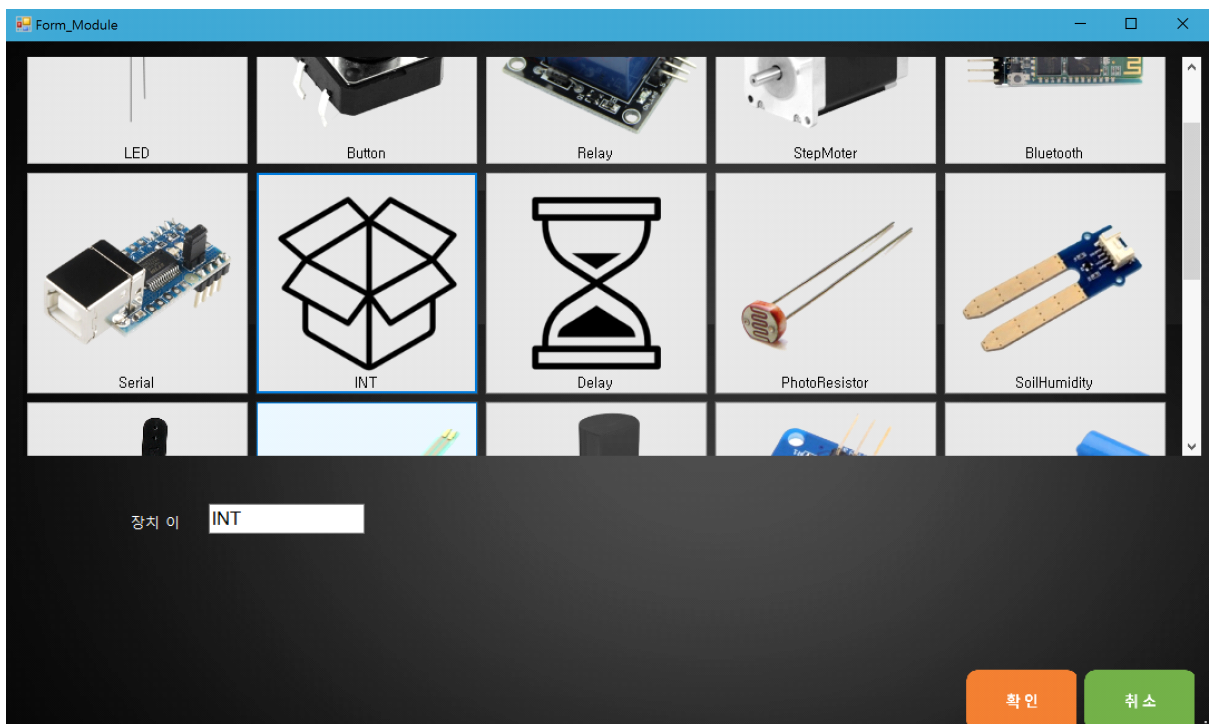
1. 장치 이름은 'Pressure1, Pressure2, Pressure3, Pressure4'로 입력합니다.
2. 아두이노의 아날로그 핀은 각각 '0, 1, 2, 3'을 입력합니다.



압력 감지가 되면 버저를 울리기 위해 사용할 모듈 Buzzer를 등록합니다.

3. 장치 이름은 'Buzzer'로 입력합니다.

4. 아두이노의 8번 핀에 연결하였기 때문에 양극 핀은 '8'를 입력합니다.



값을 임시로 저장하고 시리얼 모니터에 출력하기 위해 변수가 필요합니다

5. 6개의 INT 모듈의 이름을 'FSR1, FSR2, FSR3, FSR4, ISum, rSum'로 추가합니다.

Step 3) 트리거 만들기

무엇을 : FSR1, FSR2, FSR3, FSR4, ISum, rSum

어떻게 : Pressure1, Pressure2, Pressure3, Pressure4, Pressure1+Pressure3, Pressure2+Pressure4 로 설정한다

누가 : ISum

언제 : rSum + 50 이상이면

무엇을 : Buzzer

어떻게 : HIGH 로 설정한다.

오른쪽으로 기울어짐

rSum 모듈이

|Sum+50 이상인 값 일 때

Buzzer 모듈을

HIGH 설정 한다.

확인 취소

누가 : rSum

언제 : ISum + 50 이상이면

무엇을 : Buzzer

어떻게 : HIGH 로 설정한다.

왼쪽 다리를 꼬아 앉음

FSR1 모듈이

FSR2+50 이하인 값 일 때

Buzzer 모듈을

HIGH 설정 한다.

확인 취소

누가 : FSR1

언제 : FSR2 + 50 이하이면

무엇을 : Buzzer

어떻게 : HIGH 로 설정한다.

Form_Function

오른쪽 다리를 꼬아 앓음

FSR2 모듈이

FSR1+50 이하인 값 일 때

Buzzer 모듈을

HIGH 설정 한다.

확인 취소

누가 : FSR2

언제 : FSR1 + 50 이하이면

무엇을 : Buzzer

어떻게 : HIGH 로 설정한다.

Step 4) 결과 및 업로드

```

int FSR1 = 0;
int FSR2 = 0;
int FSR3 = 0;
int FSR4 = 0;
int lSum = 0;
int rSum = 0;

void setup() {
  Pressure1 = A0;
  Pressure2 = A1;
  Pressure3 = A2;
  Pressure4 = A3;
  Buzzer = 8;
}

void main() {
  FSR1 = analogRead(Pressure1);
  FSR2 = analogRead(Pressure2);
  FSR3 = analogRead(Pressure3);
  FSR4 = analogRead(Pressure4);
  lSum = FSR1+FSR3;
  rSum = FSR2+FSR4;

  if(lSum>=rSum+50){
    tone(Buzzer,500);
  }
  if(rSum>=lSum+50){
    tone(Buzzer,500);
  }
  if(FSR1<=FSR2+50){
    tone(Buzzer,500);
  }
  if(FSR2<=FSR1+50){
    tone(Buzzer,500);
  }
}

```

업로드

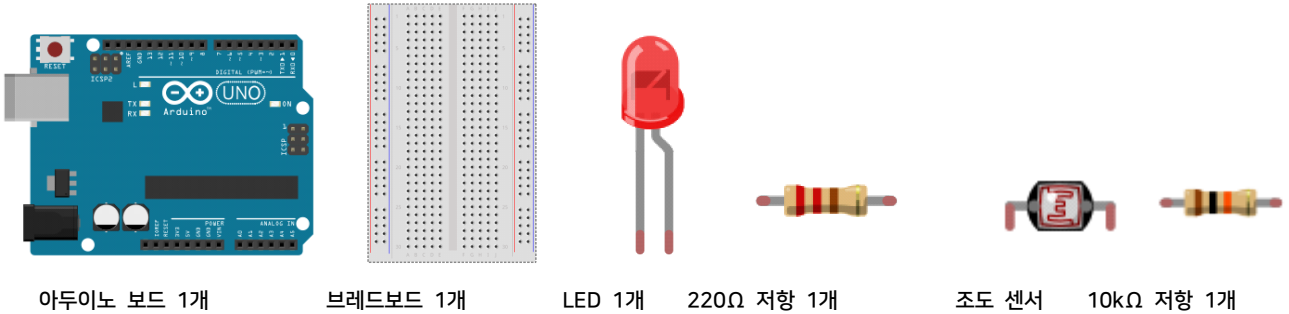
1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.
2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

```
1 int FSR1 = 0;
2 int FSR2 = 0;
3 int FSR3 = 0;
4 int FSR4 = 0;
5 int lSum = 0;
6 int rSum = 0;
7
8 void setup() {
9     Pressure1 = A0;
10    Pressure2 = A1;
11    Pressure3 = A2;
12    Pressure4 = A3;
13    Buzzer = 8;
14 }
15
16 void main() {
17     FSR1 = analogRead(Pressure1);
18     FSR2 = analogRead(Pressure2);
19     FSR3 = analogRead(Pressure3);
20     FSR4 = analogRead(Pressure4);
21     lSum = FSR1+FSR3;
22     rSum = FSR2+FSR4;
23
24     if(lSum>=rSum+50){
25         tone(Buzzer,500);
26     }
27     if(rSum>=lSum+50){
28         tone(Buzzer,500);
29     }
30     if(FSR1<=FSR2+50){
31         tone(Buzzer,500);
32     }
33     if(FSR2<=FSR1+50){
34         tone(Buzzer,500);
35     }
36 }
```

조도센서를 이용하여 LED 제어하기

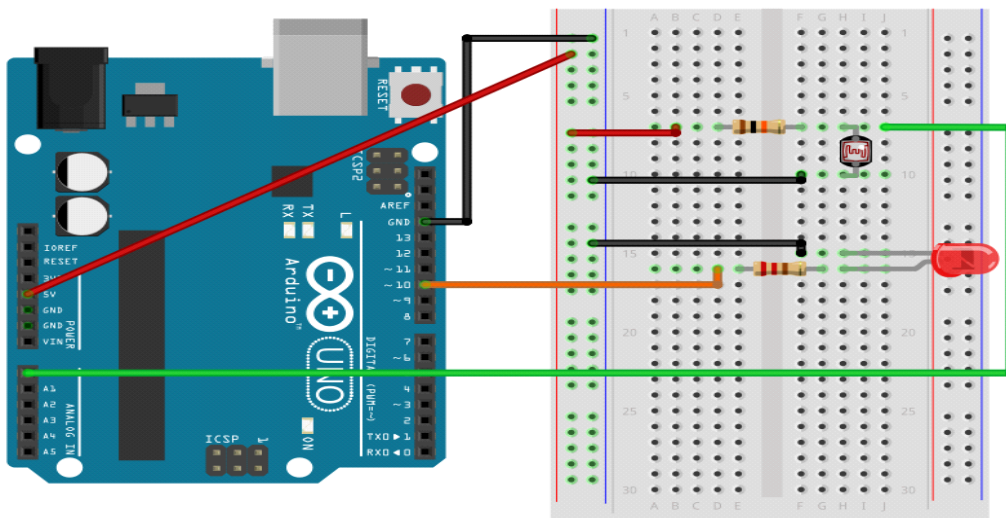
1

부품 준비하기



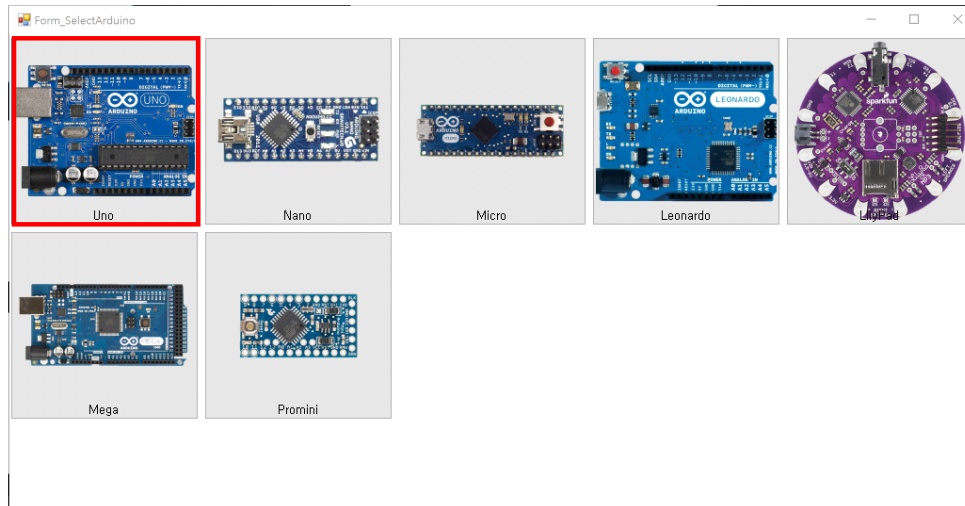
2

회로 구성하기



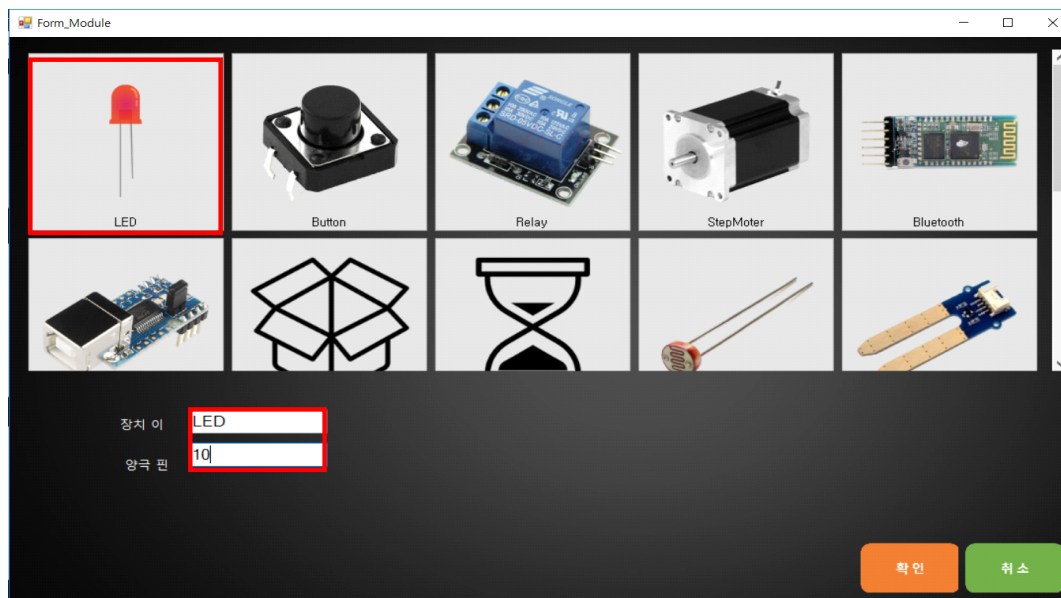
1. 브레드보드에 LED를 연결합니다.
2. LED의 음극(-) 다리에 저항을 연결합니다.
3. 아두이노 보드의 GND핀과 브레드보드의 GND라인을 연결합니다.
4. GND라인과 200Ω저항, 10kΩ저항을 연결합니다.
5. 아두이노 보드의 10번 핀과 LED의 양극(+) 핀을 연결합니다.
6. 10kΩ저항과 VCC, A0핀을 조도센서를 연결합니다.
7. GND라인과 조도센서를 연결합니다.

Step 1) 아두이노 선택



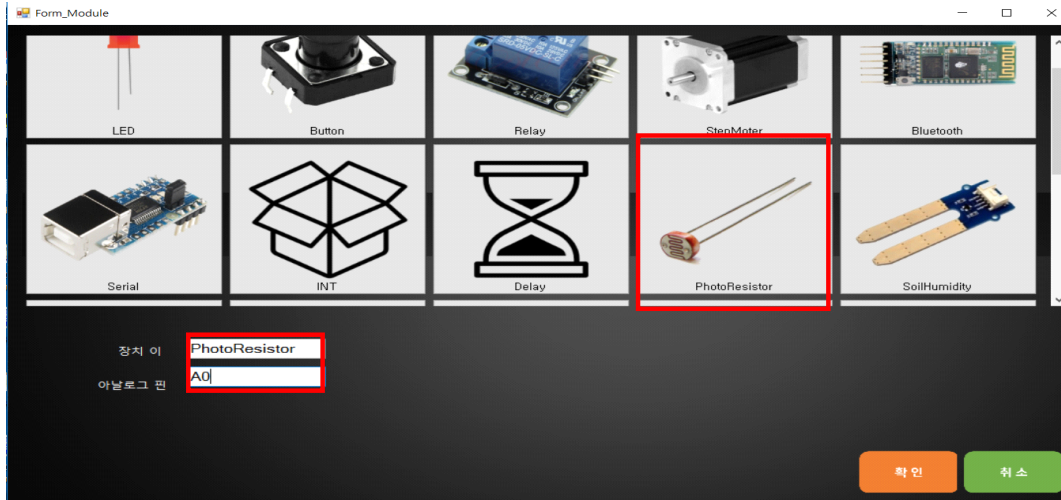
- 이번 실습에서 사용할 아두이노보드를 선택합니다.

Step 2) 모듈 등록



'조도센서를 이용 하여 LED켜기' 실습 목표를 위해서 사용할 모듈에 'LED'를 등록합니다.

1. 장치 이름은 'LED'로 입력합니다.
2. 아두이노의 디지털 10번 핀에 연결하였기 때문에 양극 핀은 '10'를 입력합니다.



'조도센서를 이용 하여 LED켜기' 실습 목표를 위해서 사용할 모듈에 '조도센서'를 등록합니다.

1. 장치 이름은 'PhotoResistor'로 입력합니다.
2. 아두이노의 아날로그 A0번 핀에 연결하였기 때문에 양극 핀은 'A0'를 입력합니다.

Step 3) 트리거 만들기

1. 기능1 트리거명, 조건부, 명령부 설정하기

어두워지면 LED 켜기 → 아두이노가 수행할 기능의 이름을 지정해 준다.

PhotoResistor 모듈이 → 누가 : 조도센서

200 이하인 값 일 때 → 언제 : 조도센서 값이 200 이하일 때

LED 모듈을 → 무엇을 : LED

HIGH 설정 한다. → 어떻게 : HIGH(켜기) 설정 한다.

2. 기능2 트리거명, 조건부, 명령부 설정하기

아두이노가 수행할 기능의 이름을 지정해 준다.

누가 : 조도센서

언제 : 조도센서 값이 200 초과일 때

무엇을 : LED

어떻게 : LOW(끄기) 설정 한다.

Step 4) 결과 및 업로드

```
void setup(){
  pinMode(10, OUTPUT);
  pinMode(A0, INPUT);
}

void loop(){
  //어두워지면 LED 켜기
  if(digitalRead(A0) <= 200){
    digitalWrite(10, HIGH);
  }
  //어둡지 않으면 LED 끄기
  if(digitalRead(A0) > 200){
    digitalWrite(10, LOW);
  }
}
```

업로드

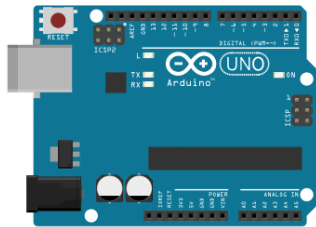
1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.
2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

```
void setup(){
1   pinMode(10, OUTPUT);    //디지털 10번 핀(LED) 출력 모드로 설정
2   pinMode(A0, INPUT);    //아날로그 A0번 핀(조도센서)에 입력 모드로
3   설정
4   }
5
6   void loop(){
7       //어두워지면 LED 켜기
8       if(digitalRead(A0) <= 200){    //조도센서에 값의 200 이하일 때
9           digitalWrite(10, HIGH);    //디지털 10번 핀으로 HIGH(1 또는 ON)출력
10      }
11      //어둡지 않으면 LED 끄기
12      if(digitalRead(A0) > 200){    //조도센서에 값이 200 초과일 때
13          digitalWrite(10, LOW);    //디지털 10번 핀으로 LOW(0 또는 OFF)출력
14      }
    }
```

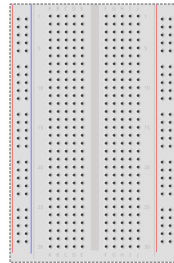

LCD "Hello world" 출력하기

1

부품 준비하기



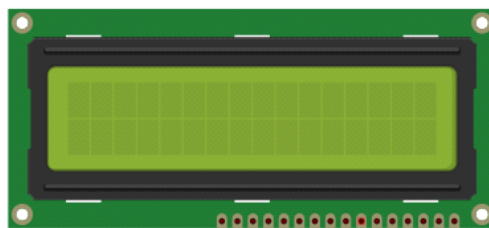
아두이노 보드 1개



브레드보드 1개



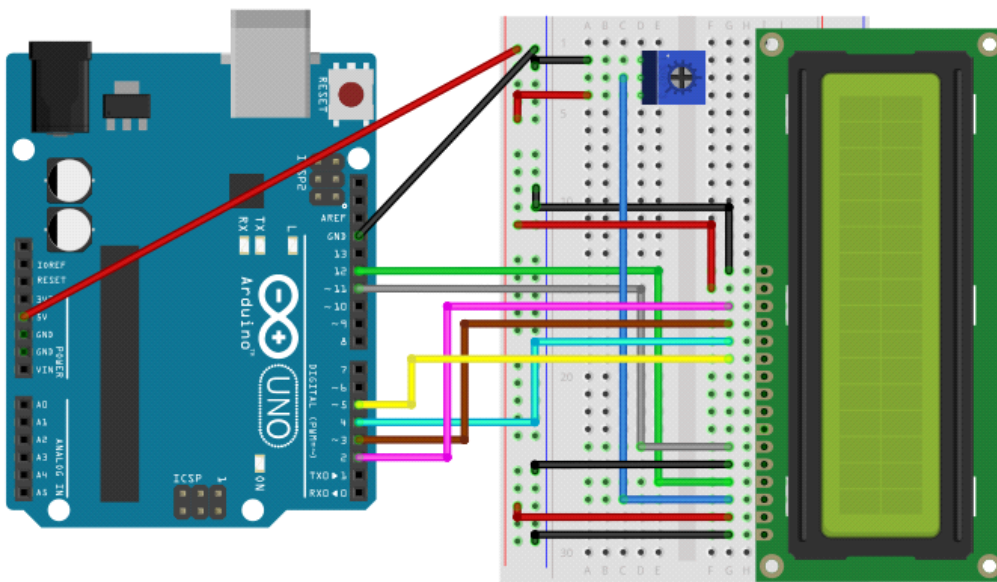
가변저항 1개



LCD 1개

2

회로 구성하기



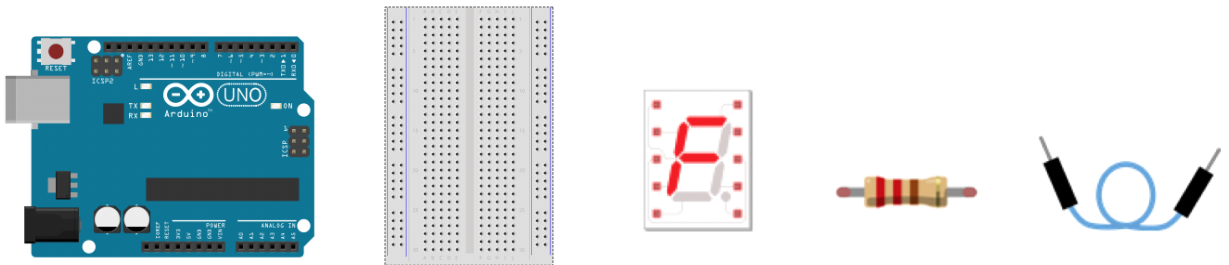
1. 브레드보드에 LCD를 연결합니다.
2. 가변저항의 GND와 VCC를 연결해줍니다.
3. LCD의 제어핀인 RS핀, E핀, 가변저항을 연결합니다.
4. 4개의 데이터 핀을 선택하여 연결해 줍니다.

```
1 #include <LiquidCrystal.h>           //LiquidCrystal.h 라이브러리를 포함
2
3 LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //LCD가 연결된 핀을 설정한다.
4
5 void setup(){
6     lcd.begin(16, 2);                 //LCD의 크기를 설정한다.
7     lcd.print("hello World!");       //LCD에 출력할 내용을 설정한다.
8 }
9
10 void loop(){
11     lcd.noDisplay();                 //화면을 끈다.
12     delay(500);                      //0.5초 지연
13     lcd.display();                  //화면을 켜다.
14     delay(500);                      //0.5초 지연
```

7세그먼트 제어하기

1

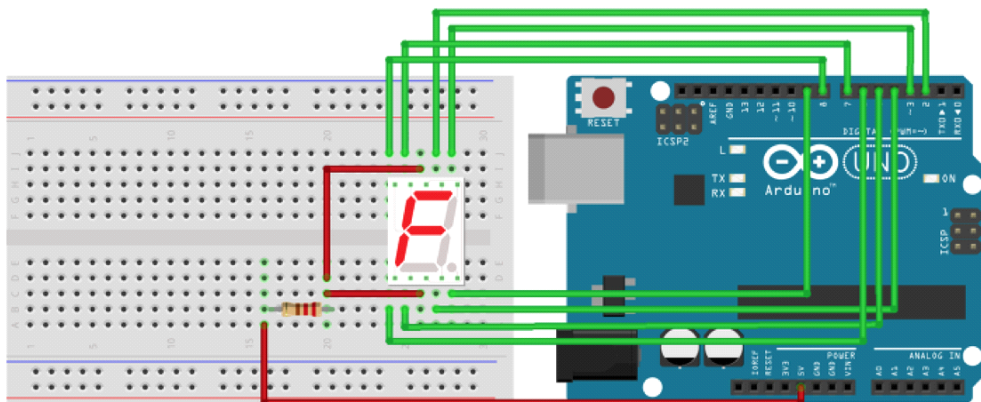
부품 준비하기



아두이노 보드 1개 브레드보드 1개 7세그먼트1개 220Ω 저항 수수점퍼선

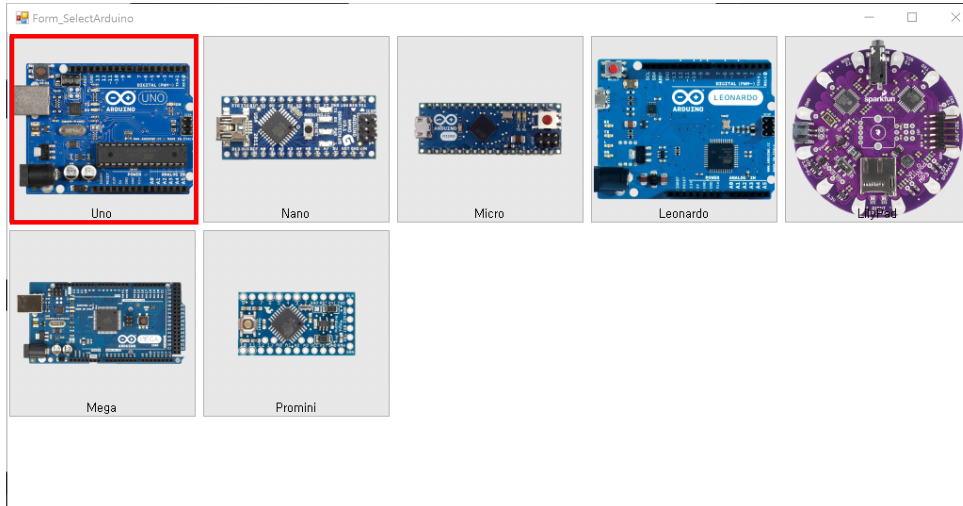
2

회로 구성하기



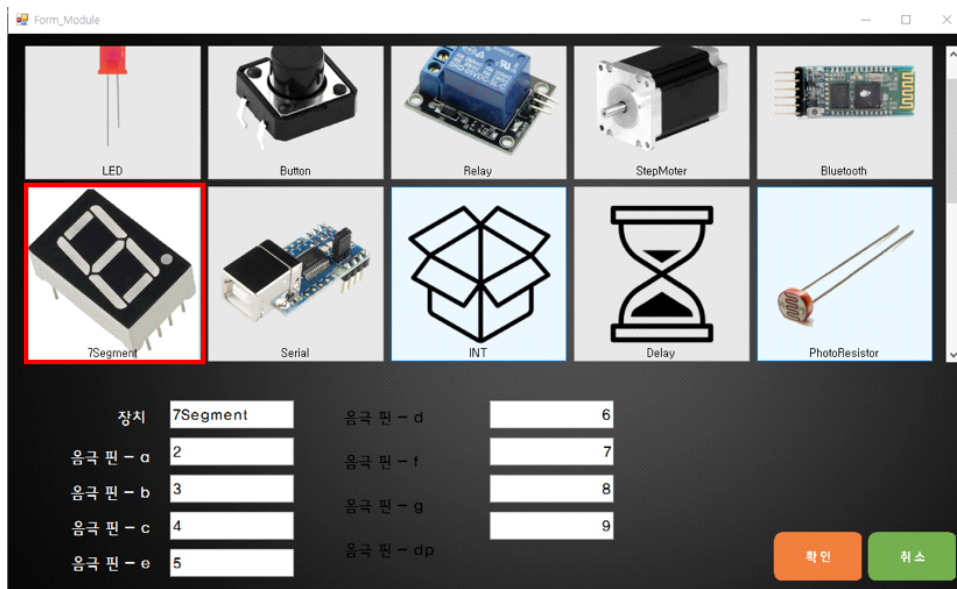
1. 브레드보드에 7세그먼트를 연결합니다.
2. 브레드보드에 220옴저항을 연결합니다.
3. 아두이노 보드의 VCC핀과 220옴저항을 서로 연결합니다.
4. 7세그먼트의 VCC핀 2개를 저항과 연결합니다.
5. 아두이노 보드의 2번 핀과 7세그먼트의 a핀을 서로 연결합니다.
6. 아두이노 보드의 3번 핀과 7세그먼트의 b핀을 서로 연결합니다.
7. 아두이노 보드의 4번 핀과 7세그먼트의 c핀을 서로 연결합니다.
8. 아두이노 보드의 5번 핀과 7세그먼트의 d핀을 서로 연결합니다.
9. 아두이노 보드의 6번 핀과 7세그먼트의 e핀을 서로 연결합니다.
10. 아두이노 보드의 7번 핀과 7세그먼트의 f핀을 서로 연결합니다.
11. 아두이노 보드의 8번 핀과 7세그먼트의 g핀을 서로 연결합니다.
12. 아두이노 보드의 9번 핀과 7세그먼트의 dp핀(점)을 서로 연결합니다.

Step 1) 아두이노 선택



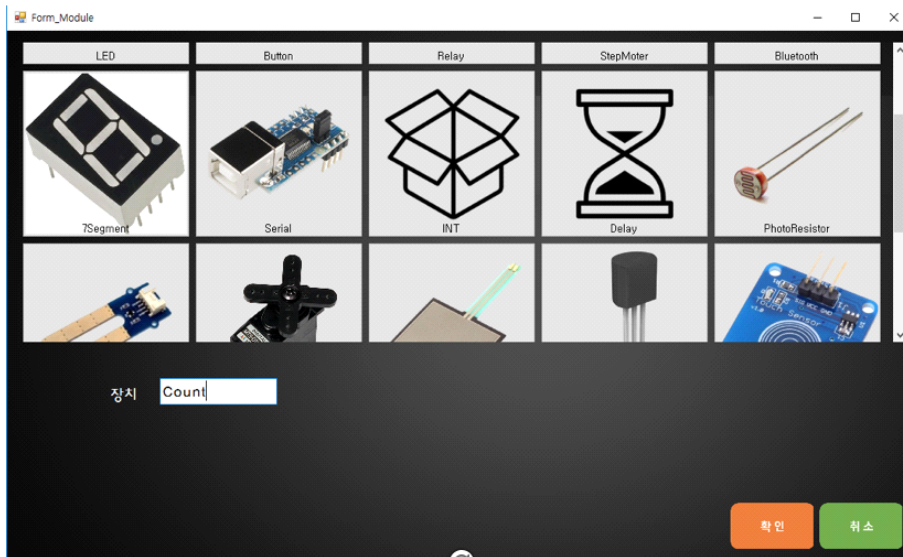
- 이번 실습에서 사용할 아두이노보드를 선택합니다.

Step 2) 모듈 등록



‘7세그먼트 제어하기’ 실습 목표를 위해서 사용할 모듈에 ‘7세그먼트’를 등록합니다.

1. 장치 이름은 ‘7Segment’로 입력합니다.
2. 회로에 연결한 것처럼 아두이노와 7세그먼트간 연결된 핀을 입력해줍니다.

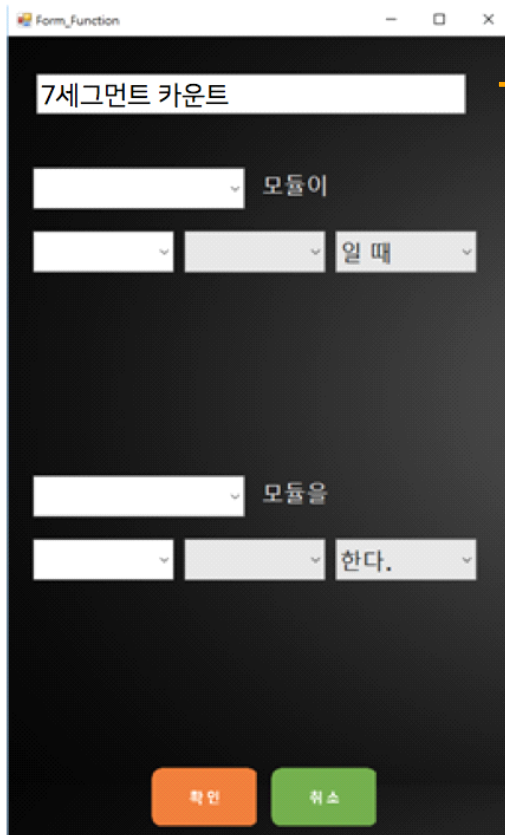


7세그먼트에 표시될 숫자를 기억할 변수를 생성해줍니다.

- INT모듈을 선택합니다.
- 모듈 이름은 Count로 해줍니다.

Step 3) 트리거 만들기

1. 트리거명 지정하기



아두이노가 수행할 기능의 이름을 지정해 준다.

7세그먼트 카운트

count 모듈이

9 이하 일 때

7Segmen 모듈을

count 설정 하고,

delay 모듈을

1000 설정 하고,

모듈을

확인 취소

7세그먼트를 Count로 설정합니다

매번 1초씩 쉬어준다

7세그먼트 카운트

count 모듈이

9 이하 일 때

count 설정 하고,

delay 모듈을

1000 설정 하고,

count 모듈을

1 더하기 한다.

확인 취소

매번 표시될 숫자를 1씩 더합니다.

Step 4) 결과 및 업로드

```
byte seven_seg_digits[10][7] = {
  { 0,0,0,0,0,0,1 }, // = 0
  { 1,0,0,1,1,1,1 }, // = 1
  { 0,0,1,0,0,1,0 }, // = 2
  { 0,0,0,0,1,1,0 }, // = 3
  { 1,0,0,1,1,0,0 }, // = 4
  { 0,1,0,0,1,0,0 }, // = 5
  { 0,1,0,0,0,0,0 }, // = 6
  { 0,0,0,1,1,1,1 }, // = 7
  { 0,0,0,0,0,0,0 }, // = 8
  { 0,0,0,1,1,0,0 }, // = 9
};

int Count = 0;

void sevenSeqWrite(byte digit)
{
  byte pin = 2;
  for (byte seqCount = 0; seqCount < 7; ++seqCount)
  {
    digitalWrite(pin, seven_seg_digits[digit][seqCount]);
    ++pin;
  }
}

void setup(){
  if(Count <= 9)
  {
    sevenSeqWrite(count - 1);
  }
}
```

업로드

1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.
2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

3

결과 소스코드

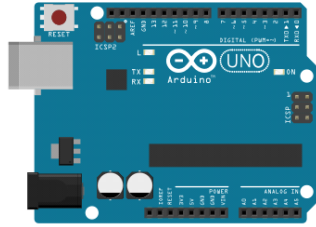
```
1   byte seven_seg_digits[10][7] = {
2     { 0,0,0,0,0,0,1 }, // = 0
3     { 1,0,0,1,1,1,1 }, // = 1
4     { 0,0,1,0,0,1,0 }, // = 2
5     { 0,0,0,0,1,1,0 }, // = 3
6     { 1,0,0,1,1,0,0 }, // = 4
7     { 0,1,0,0,1,0,0 }, // = 5
8     { 0,1,0,0,0,0,0 }, // = 6
9     { 0,0,0,1,1,1,1 }, // = 7
10    { 0,0,0,0,0,0,0 }, // = 8
11    { 0,0,0,1,1,0,0 } // = 9
12  };
13  int Count = 0;
14  void sevenSegWrite(byte digit)
15  {
16    byte pin = 2;
    for (byte seqCount = 0; seqCount < 7; ++seqCount)
    {
```

```
17     digitalWrite(pin, seven_seg_digits[digit][segCount]);
18     ++pin;
19     }
20 }
21 void setup(){
22     if(Count <= 9)
23     {
24         sevenSegWrite(count - 1);
25         delay(1000);
26         Count += 1;
27     }
28 }
```

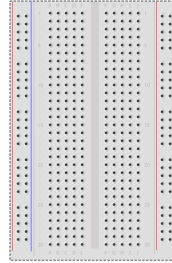

스마트홈 개발자들을 이용해 블루투스 LED, Servo Motor 제어하기

1

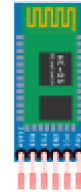
부품 준비하기



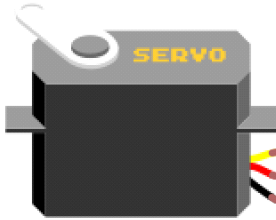
아두이노 보드 1개



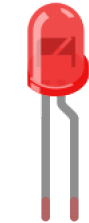
브레드보드 1개



블루투스 모듈(HC-06)



서보모터



LED 1개



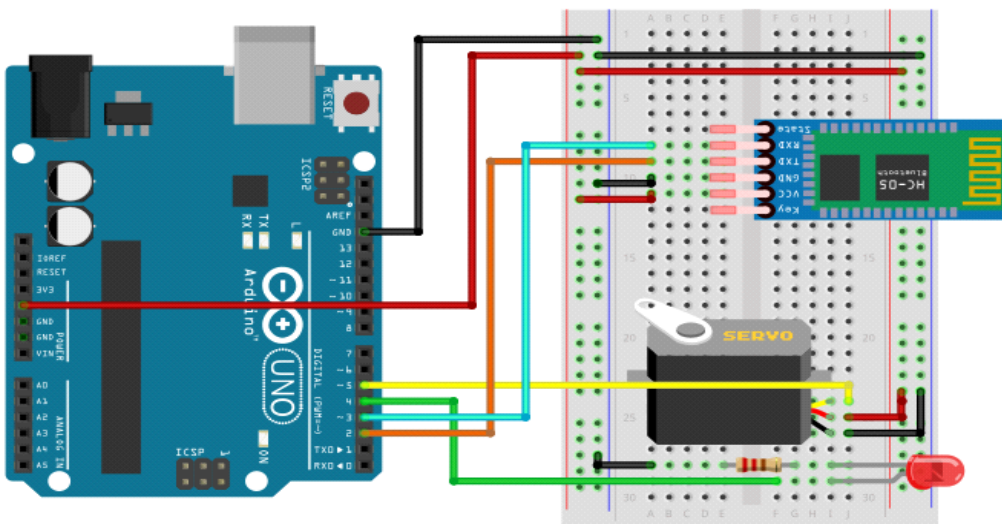
수수 점퍼선 5개



스마트폰

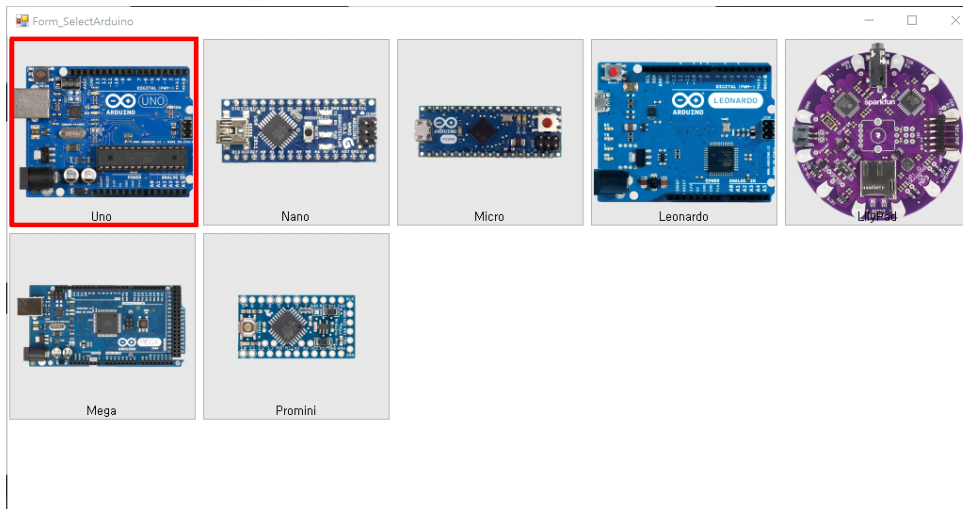
2

회로 구성하기



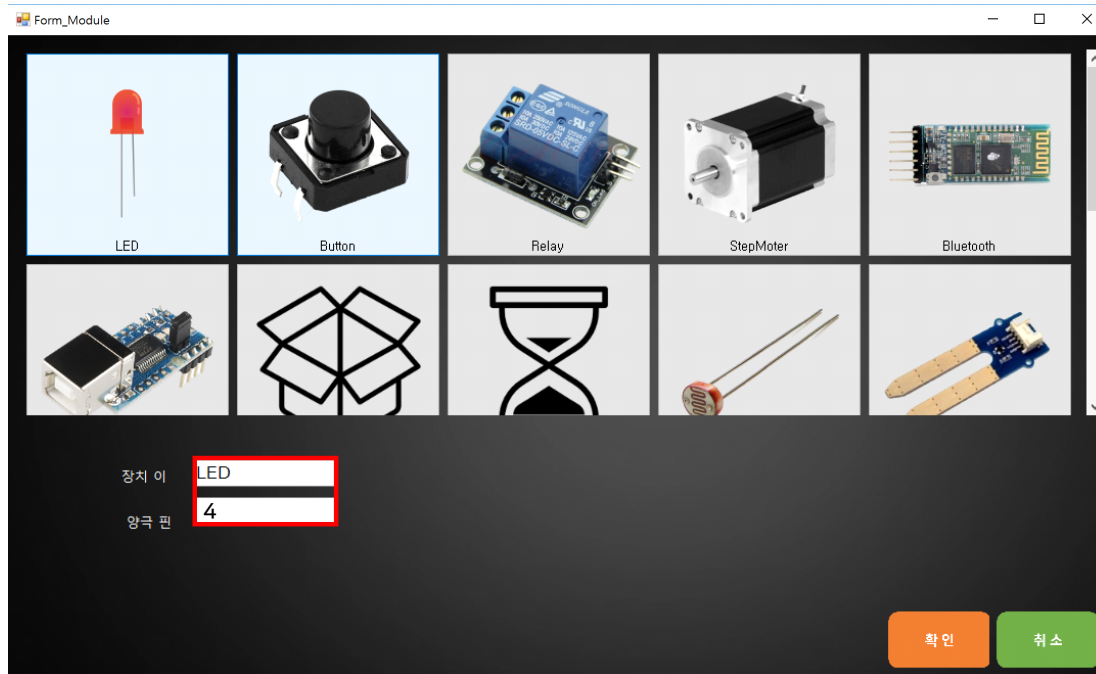
1. 브레드레드보드에 서보모터와 LED를 연결합니다.
2. 아두이노 보드의 GND핀과 브레드보드의 GND라인을 연결합니다.
3. 아두이노 보드의 5V핀과 브레드보드의 VCC라인을 연결합니다.
4. 서보모터의 GND선과 브레드보드의 GND라인을 연결합니다.
5. 서보모터의 데이터 선을 아두이노 보드의 5번 핀을 연결합니다.
6. 서보모터의 VCC선과 아두이노의 5V 핀을 연결합니다.
7. LED의 GND선과 200 Ω 를 브레드보드의 GND라인의 연결합니다.
8. LED의 데이터 선을 아두이노 보드의 4번 핀에 연결합니다.
9. 브레드보드에 HC-06 블루투스 모듈을 연결합니다.
10. 블루투스 모듈의 GND핀과 브레드보드의 GND라인을 연결합니다.
11. 블루투스 모듈의 VCC핀과 브레드보드의 VCC라인을 연결합니다.
12. 아두이노의 2번 핀과 블루투스 모듈의 TXD 핀을 연결합니다.
13. 아두이노의 3번 핀과 블루투스 모듈의 RXD 핀을 연결합니다.

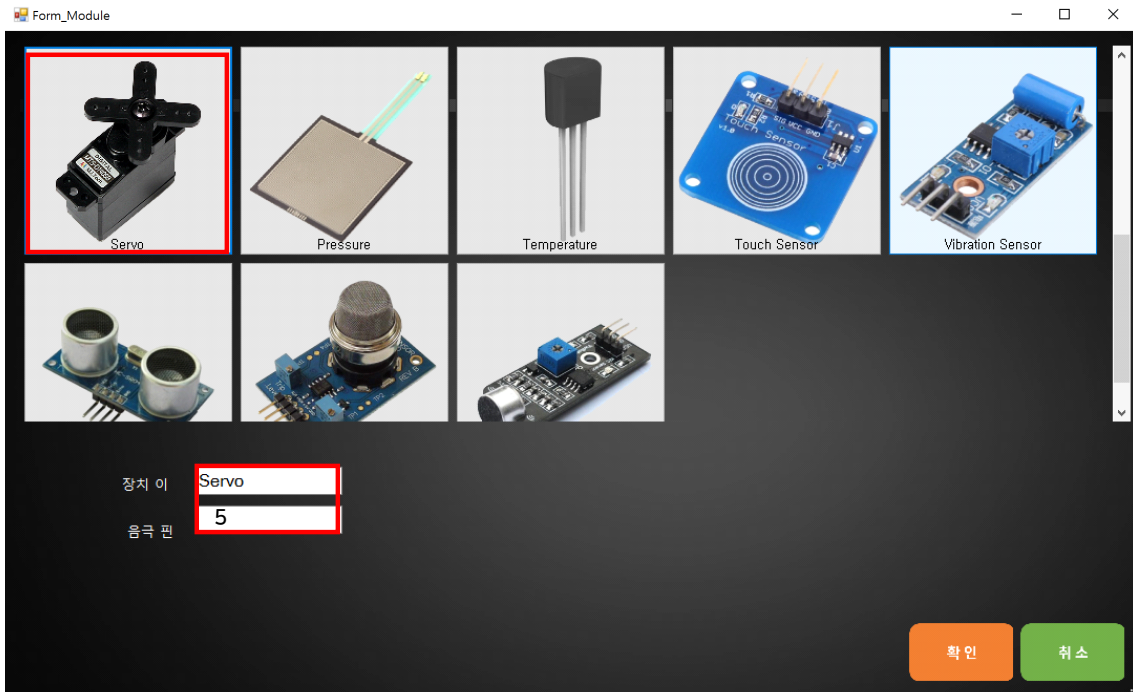
Step 1) 아두이노 선택



- 이번 실습에서 사용할 아두이노 보드를 선택합니다.

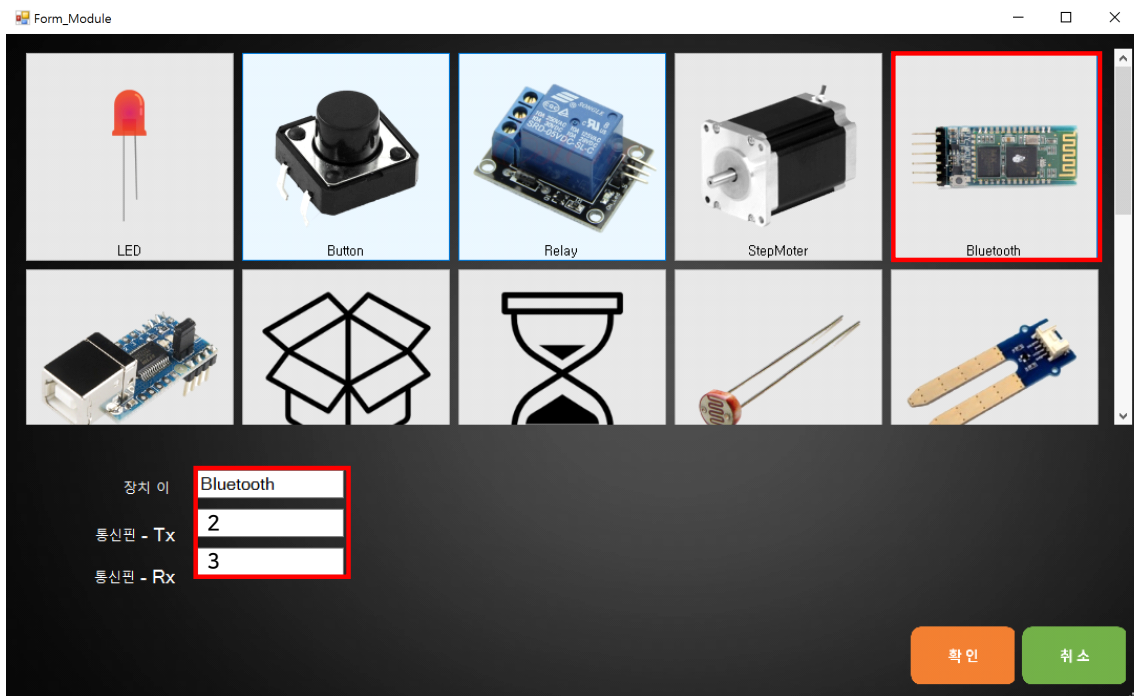
Step 2) 모듈 등록





실습을 위해서 프로그램에 사용할 모듈에 서보모터를 등록합니다.

1. 장치 이름은 'Servo'로 입력합니다.
2. 아두이노의 디지털 4번 핀에 연결하였기 때문에 데이터 핀은 '5'를 입력합니다.



실습을 위해서 프로그램에 사용할 모듈에 블루투스 모듈을 등록합니다.

1. 장치 이름은 'Bluetooth'로 입력합니다.
2. 블루투스 핀에 연결한 TXD, RXD 핀 번호에 맞추어 핀 번호를 입력합니다.

Step 3) 트리거 만들기

1. 트리거 명 지정하기

Form_Function

블루투스 - LED ON

모듈이

일 때

모듈을

한다.

확인 취소

아두이노가 수행할 기능의 이름을 지정해 준다.

2. 조건부 : "누가", "언제" 설정하기

Form_Function

블루투스 - LED ON

Bluetooth 모듈이

ON 같은 값 일 때

모듈을

한다.

확인 취소

누가 : 블루투스모듈

언제 : 값이 ON일때

3. 명령부 : "무엇을", "어떻게" 설정하기

블루투스 - LED ON

Bluetooth 모듈이

ON 같은 값 일 때

LED 모듈을

HIGH 설정 한다.

확인 취소

무엇을 : LED

어떻게 : HIGH(켜기) 설정 합니다.

1. 트리거 명 지정하기

블루투스 - LED OFF

모듈이

일 때

모듈을

한다.

확인 취소

아두이노가 수행할 기능의 이름을 지정해 준다.

2. 조건부 : "누가", "언제" 설정하기



블루투스 - LED OFF

Bluetooth 모듈이

OFF 같은 값 일 때

누가 : 블루투스모듈

언제 : 값이 OFF일때

확인 취소

3. 명령부 : "무엇을", "어떻게" 설정하기



블루투스 - LED OFF

Bluetooth 모듈이

OFF 같은 값 일 때

LED 모듈을

LOW 설정 한다.

무엇을 : LED

어떻게 : LOW(끄기) 설정 합니다.

확인 취소

1. 트리거 명 지정하기

Form_Function

블루투스 - Servo UP

모듈이

일 때

모듈을

한다.

확인 취소

아두이노가 수행할 기능의 이름을 지정해 준다.

2. 조건부 : "누가", "언제" 설정하기

Form_Function

블루투스 - Servo UP

Bluetooth 모듈이

UP 같은 값 일 때

모듈을

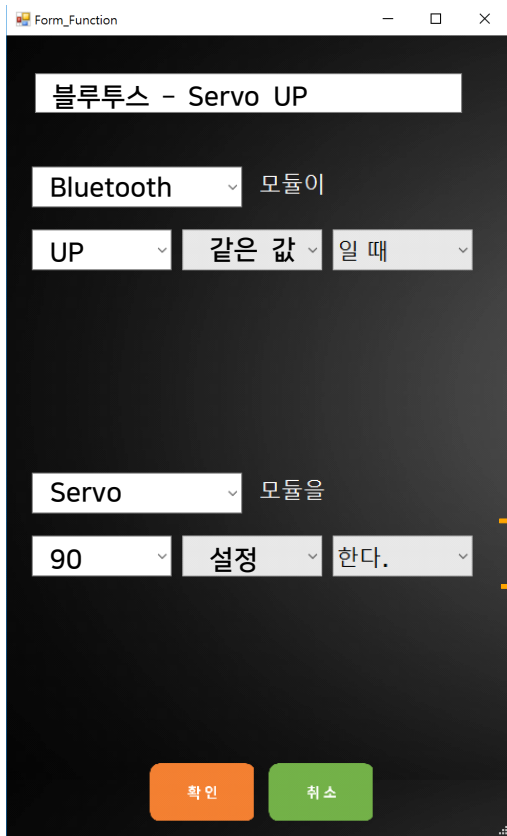
한다.

확인 취소

누가 : 블루투스모듈

언제 : 값이 UP일때

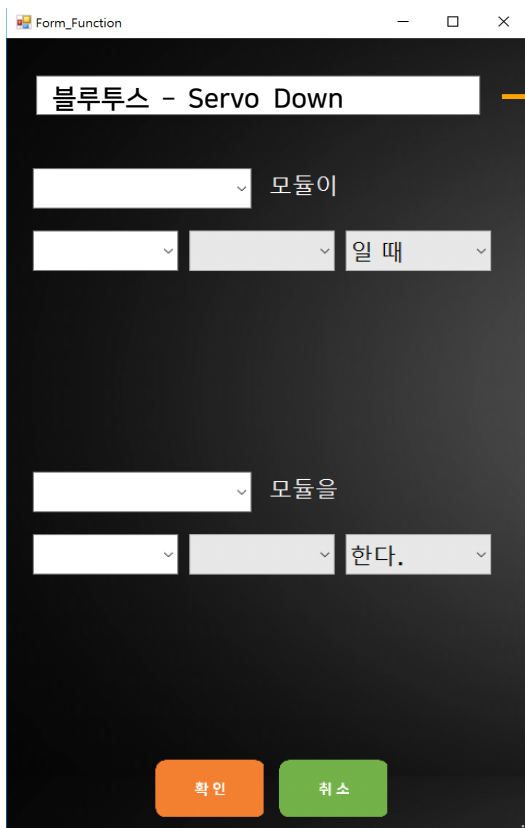
3. 명령부 : "무엇을", "어떻게" 설정하기



무엇을 : Servo

어떻게 : 90도로 회전 설정 합니다.

1. 트리거 명 지정하기



아두이노가 수행할 기능의 이름을 지정해 준다.

2. 조건부 : "누가", "언제" 설정하기

The screenshot shows a web form titled "블루투스 - Servo Down". It contains several dropdown menus and buttons. Two callout boxes with arrows point to specific settings:

- Callout 1: "누가 : 블루투스모듈" (Who: Bluetooth module) points to the "Bluetooth" dropdown menu.
- Callout 2: "언제 : 값이 DOWN일때" (When: When value is DOWN) points to the "DOWN" dropdown menu.

Other visible elements include a "확인" (Confirm) button in orange and a "취소" (Cancel) button in green.

3. 명령부 : "무엇을", "어떻게" 설정하기

The screenshot shows the same web form, but with different settings. Two callout boxes with arrows point to specific settings:

- Callout 1: "무엇을 : Servo" (What: Servo) points to the "Servo" dropdown menu.
- Callout 2: "어떻게 : 0도로 회전 설정 합니다." (How: Set to rotate 0 degrees) points to the "0" dropdown menu.

Other visible elements include a "확인" (Confirm) button in orange and a "취소" (Cancel) button in green.

Step 4) 결과 및 업로드

```
#include <SoftwareSerial.h>
#include <Servo.h>

SoftwareSerial blueSerial(2, 3);
String blueInputStr;
Servo myservo;

void setup()
{
    pinMode(4, OUTPUT);
    blueSerial.begin(9600);
    myservo.attach(5,544,2400);
}

void loop()
{
    while(blueSerial.available())
    {
        char blueChar =
            (char)blueSerial.read();
```



업로드

```
if(blueChar == '#')
{
    blueInputStr = "";
    return;
}
blueInputStr+=blueChar;
}
//LED ON
if(blueInputStr.equals("ON") )
{
    digitalWrite(4, HIGH);
}
//LED OFF
if(blueInputStr.equals("OFF") )
{
    digitalWrite(4, LOW);
}
```



업로드

```
//모터 90도
if(blueInputStr.equals("UP") )
{
    myservo.write(90);
}
//모터 0도
if(blueInputStr.equals("DOWN") )
{
    myservo.write(0);
}
```



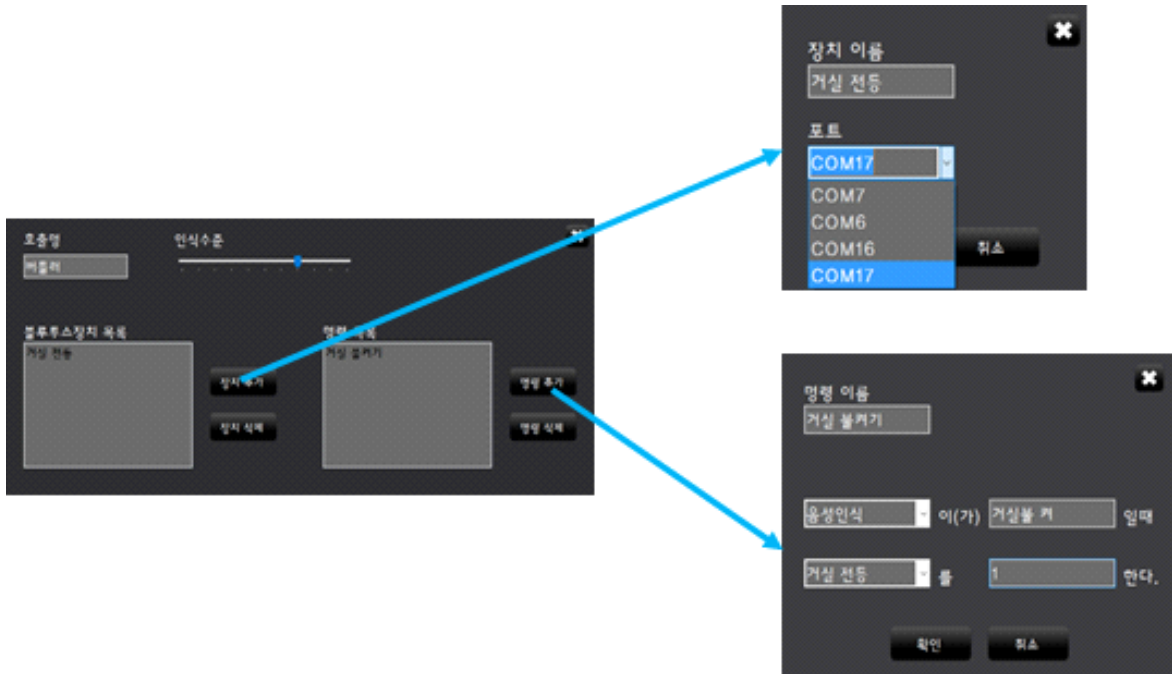
업로드

1. 만들어진 코드를 확인해 생각한 로직과 일치하는지 비교해봅시다.
2. 생각과 일치한다면 '업로드' 버튼을 눌러 아두이노에 코드를 업로드 합니다.

음성인식 IoT 스마트홈 Studio 사용법

1

음성인식 IoT 스마트홈 Studio UI



위 그림은 음성인식 IoT 스마트홈 Studio의 메인화면입니다.

이 서버를 통한 스마트홈 개발은 크게 3단계의 과정으로 이루어져있습니다.

첫 번째 단계로 트레이 아이콘을 눌러 메인화면을 호출합니다.

두 번째 단계로 장치 추가 버튼을 눌러 프로젝트에서 사용할 블루투스 모듈을 선택합니다.

세 번째 단계로 명령 추가 버튼을 눌러 두 번째 단계에서 추가한 블루투스 모듈을 선택하고 어떠한 데이터를 보낼 지 정의합니다.

세 번째 단계까지 마쳤다면 스마트홈 서버는 여러분이 설정한 방식으로 동작할 것 입니다.

음성인식 스마트홈 만들기

1

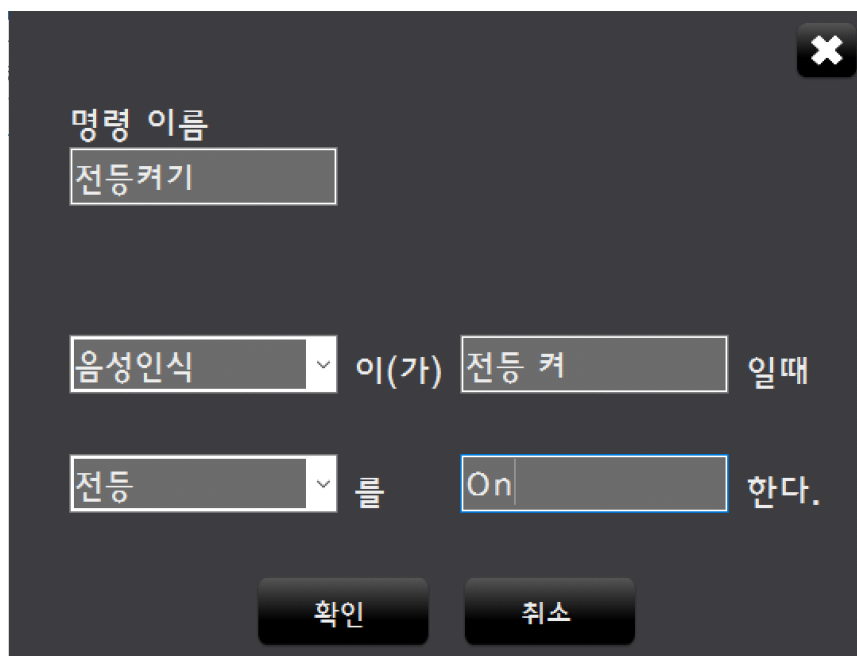
음성인식 IoT 스마트홈 Studio - 장치 추가



1. 블루투스 장치 2개를 페어링 합니다.(페어링한 장치는 COM 포트로 표현될것입니다.)
2. 블루투스 장치(COM)을 선택하고 장치 이름을 설정합니다.

2

음성인식 IoT 스마트홈 Studio - 음성명령 등록



1. 어떤 음성명령이 인식되었을 때 반응할 것인지 지정합니다.
2. 음성명령이 인식된 후 전등을 켜기 위해 추가한 모듈을 선택하고 보낼 명령을 입력합니다.

A settings dialog box with a dark background and a close button (X) in the top right corner. The title is '명령 이름' (Command Name) with the value '전등끄기' (Turn off lights) in a text input field. Below this, there are two rows of settings. The first row has a dropdown menu set to '음성인식' (Voice Recognition), followed by the text '이(가)' (this), a text input field containing '전등 꺼' (Turn off lights), and the text '일때' (when). The second row has a dropdown menu set to '전등' (Lights), followed by the text '를' (the), a text input field containing 'Off', and the text '한다.' (do).

명령 이름
전등끄기

음성인식 이(가) 전등 꺼 일때

전등 를 Off 한다.

확인 취소

1. 어떤 음성명령이 인식되었을 때 반응할 것인지 지정합니다.
2. 음성명령이 인식된 후 전등을 끄기 위해 추가한 모듈을 선택하고 보낼 명령을 입력합니다.

A settings dialog box with a dark background and a close button (X) in the top right corner. The title is '명령 이름' (Command Name) with the value '창문열기' (Open window) in a text input field. Below this, there are two rows of settings. The first row has a dropdown menu set to '음성인식' (Voice Recognition), followed by the text '이(가)' (this), a text input field containing '창문 열어' (Open window), and the text '일때' (when). The second row has a dropdown menu set to '창문' (Window), followed by the text '를' (the), a text input field containing 'Up', and the text '한다.' (do).

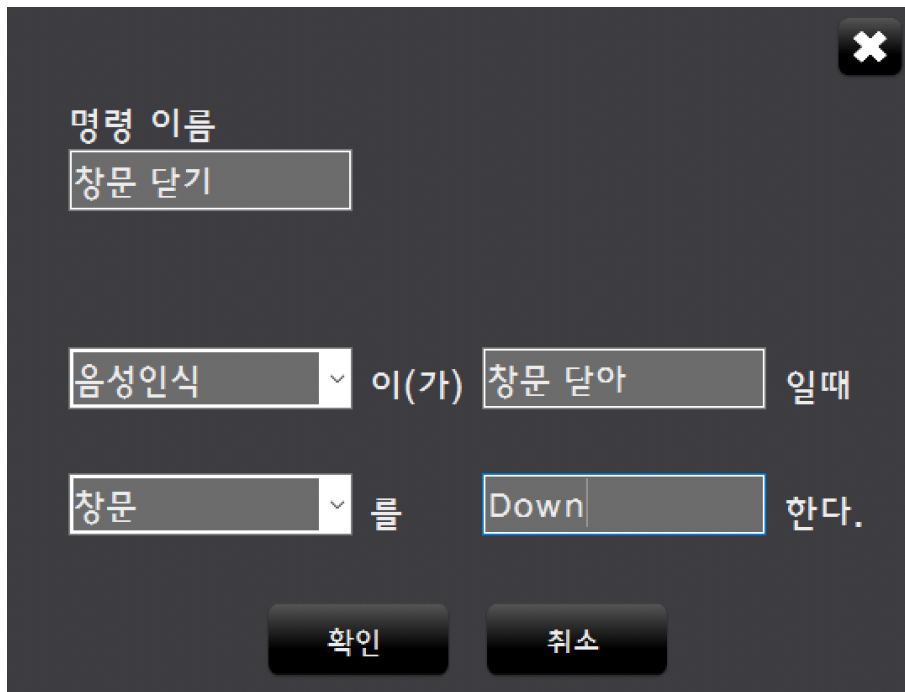
명령 이름
창문열기

음성인식 이(가) 창문 열어 일때

창문 를 Up 한다.

확인 취소

1. 어떤 음성명령이 인식되었을 때 반응할 것인지 지정합니다.
2. 음성명령이 인식된 후 창문을 열기 위해 추가한 모듈을 선택하고 보낼 명령을 입력합니다.



- 어떤 음성명령이 인식되었을 때 반응할 것인지 지정합니다.
- 음성명령이 인식된 후 창문을 닫기 위해 추가한 모듈을 선택하고 보낼 명령을 입력합니다.

3

음성인식 IoT 스마트홈 Studio - 결과 화면

