피에조 스피커

OneKick

목차

- 기초 학습
  - 제어구조
- Switch 읽기
  - 개 요
  - 예 제(1)
  - 예 제(2)
- 응용 실습

Basic learning

기초 학습

산술 연산자

OneKick

- = (assignment operator)
  - 대입 연산자
  - 왼쪽에 있는 변수에 오른쪽에 있는 값을 저장
  - ex) a에 13을 저장
    - a = 13;

- + (addition)
  - 두 피연산자의 값을 더함
  - ex) 4와 3을 더한 값을 a에 저장
    - a = 4 + 3;

- - (subtraction)
  - 왼쪽의 피연산자 값에서 오른쪽의 피연산자 값을 뺌
  - ex) 4에서 3을 뺀 값을 a에 저장
    - a = 4 - 3;

산술 연산자

OneKick

- * (multiplication)
  - 두 피연산자의 값을 곱함
  - ex) 4와 3을 곱한 값을 a에 저장
    - a = 4 * 3;

- / (division)
  - 왼쪽의 피연산자 값을 오른쪽의 피연산자 값으로 나눔
  - ex) 4를 3으로 나눈 값을 a에 저장
    - a = 4 / 3;

- % (modulo)
  - 왼쪽의 피연산자 값을 오른쪽의 피연산자 값으로 나누었을 때의 나머지 반환
  - ex) 4를 3으로 나눈 나머지 값을 a에 저장
    - a = 4 % 3;

비교 연산자

OneKick

- == (equal to)
  - 왼쪽에 있는 (변수) 값과 오른쪽에 있는 (변수) 값이 같은지 비교
  - a와 b가 같으면 if문을 실행

```
if(a == b)
{
  // action
}
```

- != (not equal to)
  - 왼쪽에 있는 (변수) 값과 오른쪽에 있는 (변수) 값이 다른지 비교
  - a와 b가 다르면 if문을 실행

```
if(a != b)
{
  // action
}
```

비교 연산자

OneKick

- < (less than)
  - 왼쪽에 있는 (변수) 값이 오른쪽에 있는 (변수) 값보다 작은지 비교
  - a가 b보다 작으면 if문을 실행

```
if(a < b)
{
  // action
}
```

- >(greater than)
  - 왼쪽에 있는 (변수) 값이 오른쪽에 있는 (변수) 값보다 큰지 비교
  - a가 b보다 크면 if문 실행

```
if(a < b)
{
  // action
}
```

비교 연산자

OneKick

- <= (less than or equal to)
  - 왼쪽에 있는 (변수) 값이 오른쪽에 있는 (변수) 값보다 작거나 같은가를 비교
  - a가 b보다 작거나 같으면 if문 실행

```
if(a <= b)
{
  // action
}
```

- >=(greater than or equal to)
  - 왼쪽에 있는 (변수) 값이 오른쪽에 있는 (변수) 값보다 크거나 같은가를 비교
  - a가 b보다 크거나 같으면 if문 실행

```
if(a >= b)
{
  // action
}
```

논리 연산자

OneKick

- && (and)
  - 왼쪽과 오른쪽이 모두 참일 경우에 1을 반환
  - a와 b가 참이면 if문을 실행

```
if(a && b)
{
  // action
}
```

- || (or)
  - 왼쪽이나 오른쪽이 참, 즉 하나이상이 참일 경우 1을 반환
  - a 또는 b가 참이면 if문을 실행

```
if(a || b)
{
  // action
}
```

논리 연산자

- ! (not)
  - 변수가 참이면 0, 거짓이면 1을 반환
  - a가 거짓이면 if문을 실행

```
if(!a)
{
  // action
}
```

피에조 스피커

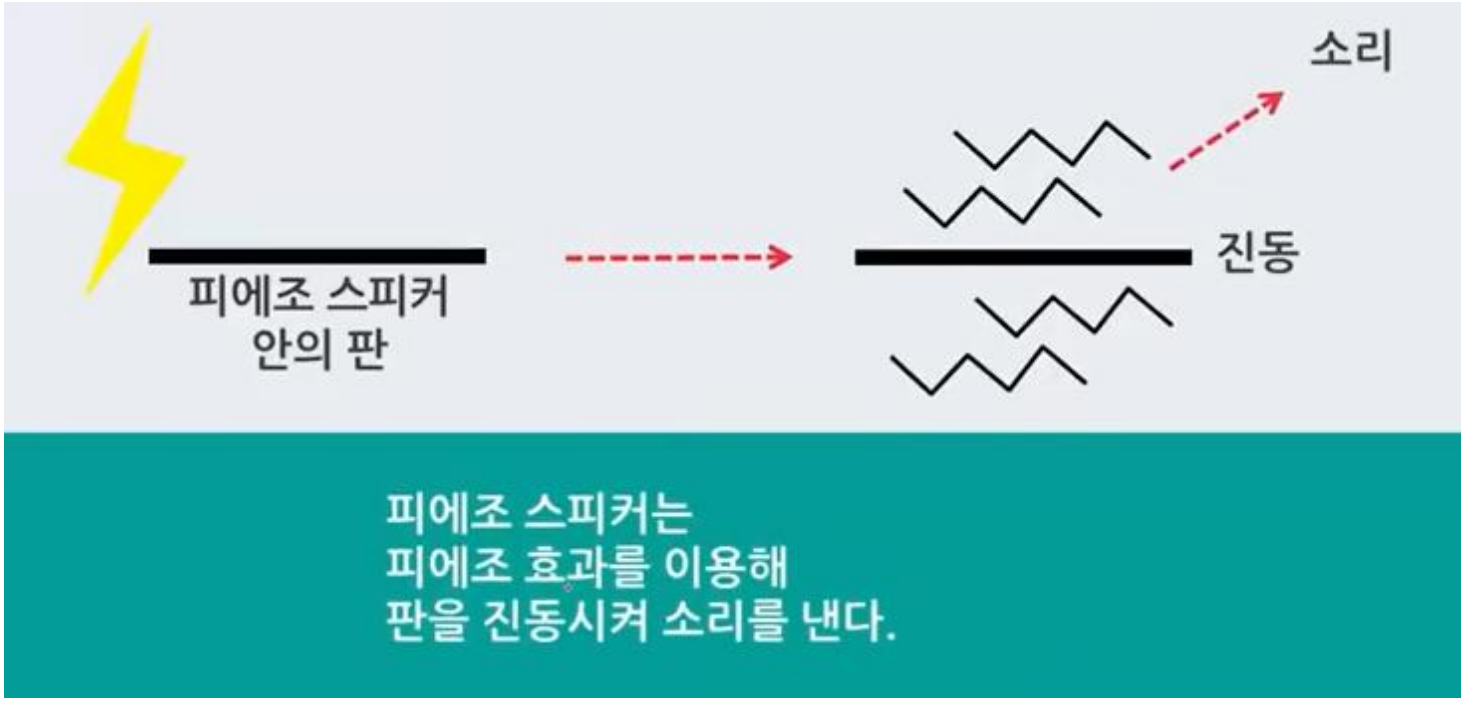
능동부저와 수동부저

- Active(능동)부저와
  - LED를 켜고 끄는 것처럼 Arduio 핀으로 간단히 켜고 끌수 있음
  - 예) 알람

- Passive(수동) 부저의 차이
  - 장치 또는 스피커의 경우 "사운드 신호"를 보냄
  - Arduino에서 "톤"을 생성하여 하드웨어 및 소프트웨어로 음악 재생
  - 단점은 사운드를 생성하는데 프로세서 시간이 필요하다는 것
  - 이처럼 프로그램(스케치)이 동시에 많은 일을 하는 경우 문제
  - 피에조 스피커는 Arduino에서 직접 구동 될 수 있음

좌 -<Active 능동 부저> 우- <Passive 수동 부저>

피에조 스피커 원리



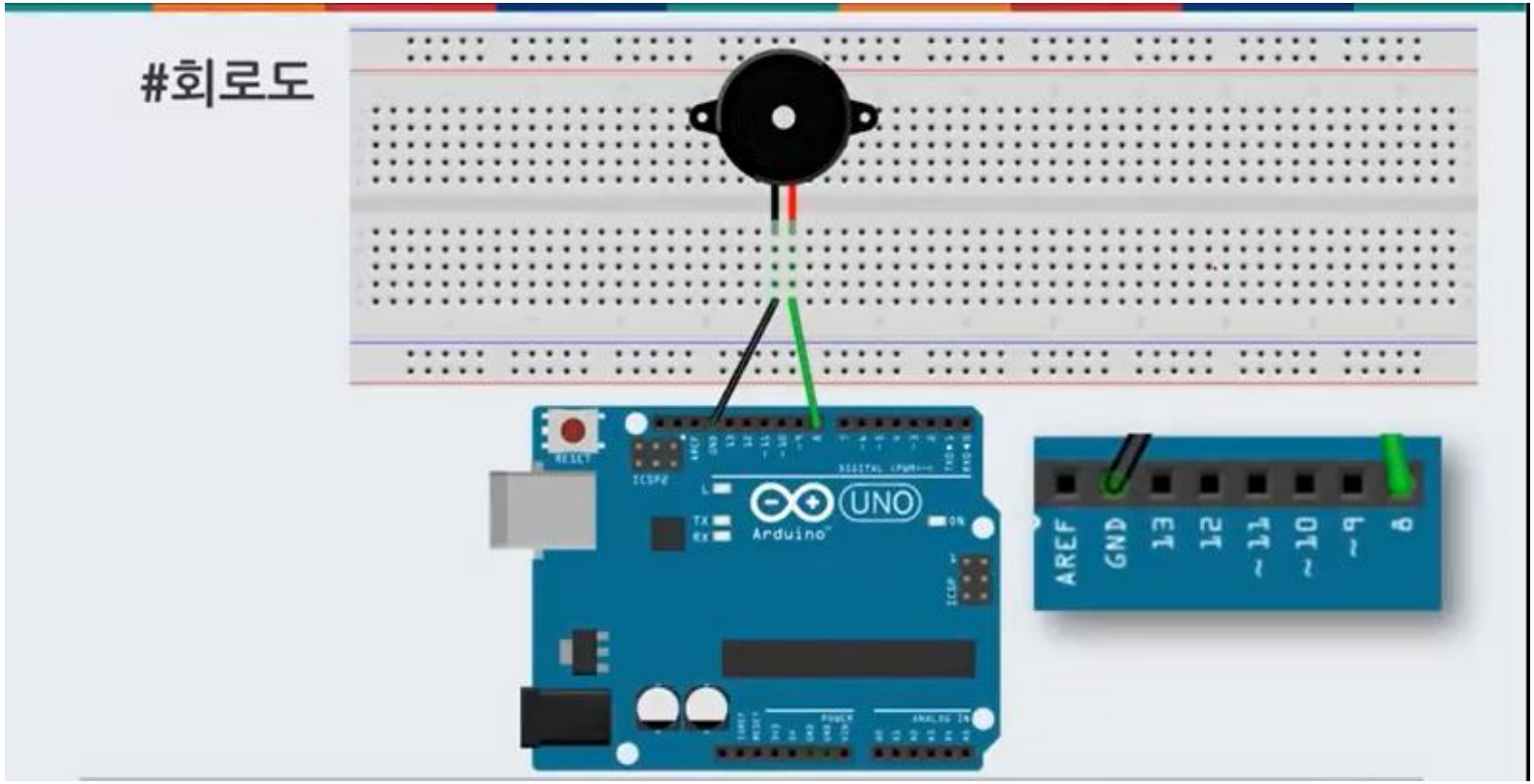
능동부저 소리내기

OneKick

```
const int buzzerPin= 8;


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


void loop() {
        digitalWrite(buzzerPin, HIGH);
        delay(1000);
        digitalWrite(buzzerPin, LOW);
        delay(1000);
        digitalWrite(buzzerPin, HIGH);
        delay(5000);
        digitalWrite(buzzerPin, LOW);
        delay(5000);
}
```

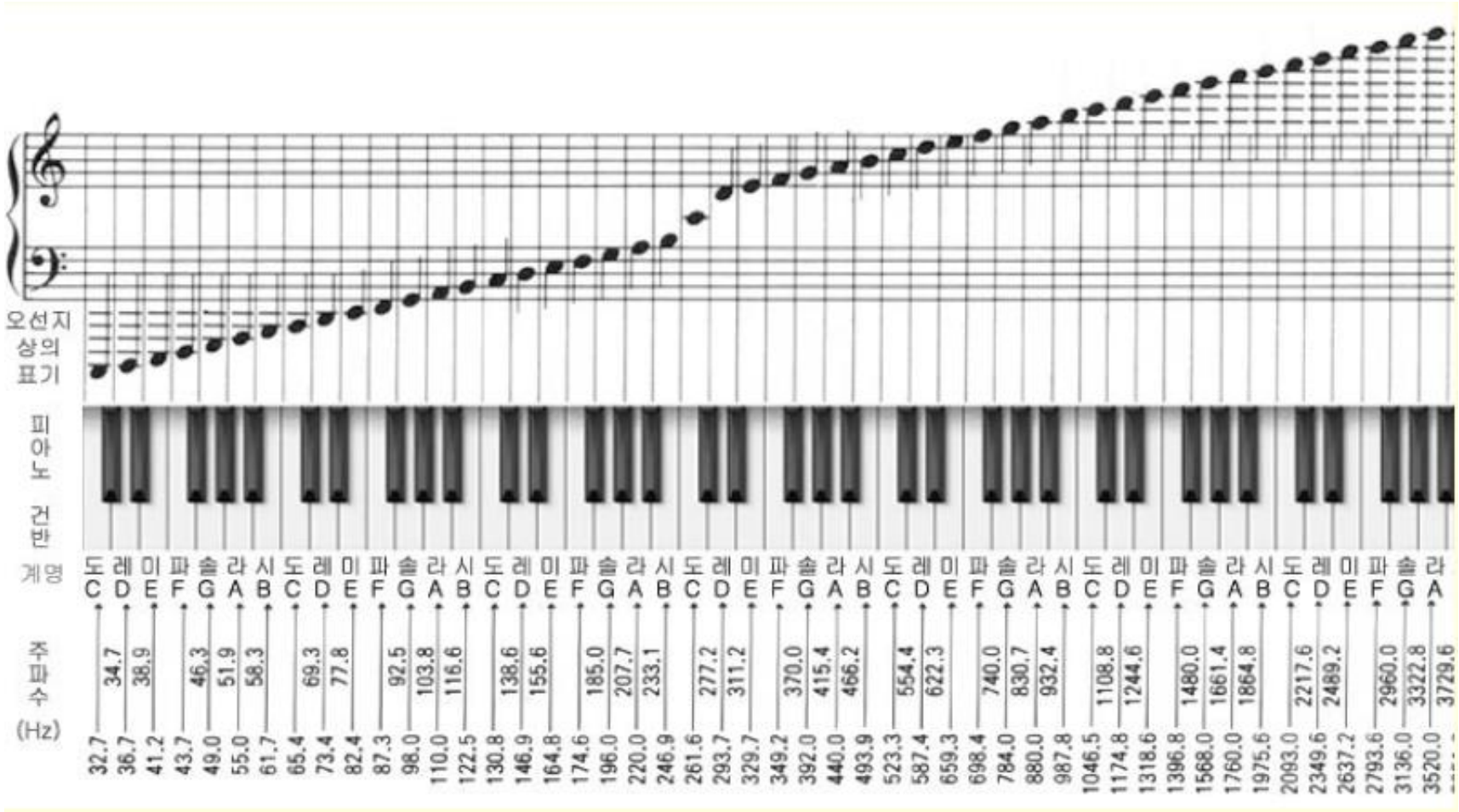
학교종이 땡땡땡 연주하기

- +는 8번
- 나머지는 GND
- 안에 얇은 금속판이 있어서 전기가 들어가면 떨려서 판의 떨림을 소리로 표현



수동부저 사용하기

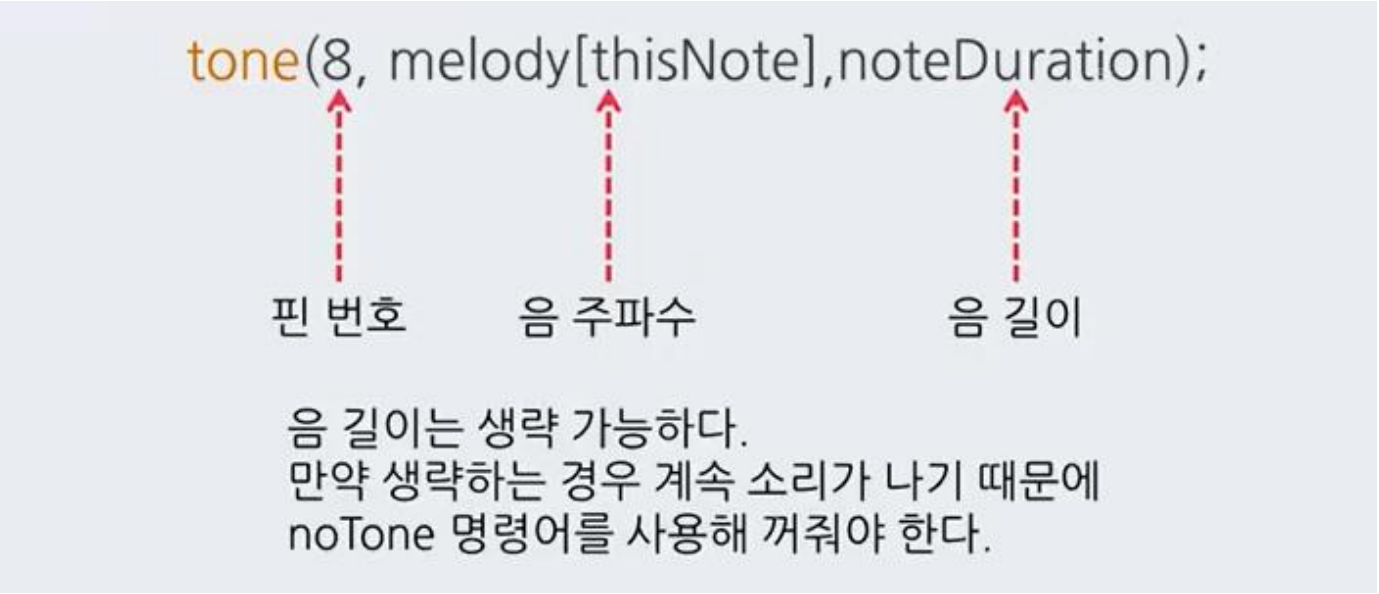
- 파일 → 예제 → 02.Degital-toneMelody 선택
- pitches.h 를 통해 음계표 확인



학교종이 땡땡땡 (코드)

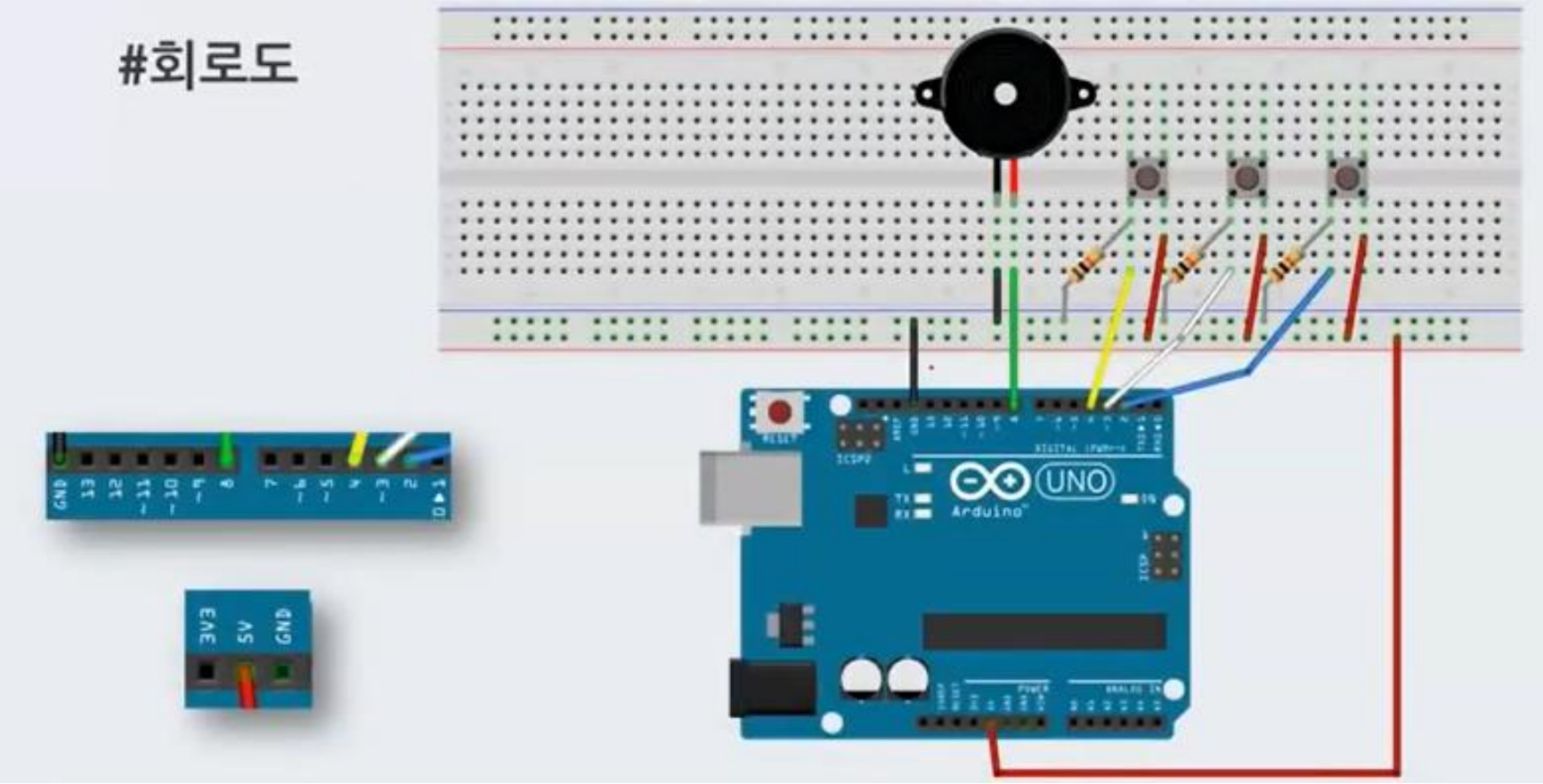
OneKick

```
#include “pitches.h”

//학교종이 땡땡땡의 음계
int melody[] = { NOTE_G4, NOTE_G4, NOTE_A5, NOTE_A5, NOTE_G4, NOTE_G4,
        NOTE_E4, NOTE_G4, NOTE_G4, NOTE_E4, NOTE_E4,NOTE_D4, 0 ,
        NOTE_G4, NOTE_G4, NOTE_A5, NOTE_A5, NOTE_G4, NOTE_G4, NOTE_E4,
        NOTE_G4, NOTE_E4, NOTE_D4, NOTE_E4, NOTE_C4, 0 };

        // 음 길이를 나타내는 배열
        int noteDurations[] = { 1,1,1,1,1,1,2,1,1,1,1,3,1,1,1,1,1,1,1,2,1,1,1,1,3,1};

        void setup() {
        for(int thisNote = 0; thisNote < 26; thisNote++){
                int noteDuration = 250 * noteDuration[thisNote];
                tone( 8, melody[thisNote], noteDuration);

                int pauseBetweenNotes = noteDurations * 1.30;
                delay ( pauseBetweenNotes);
                noTone(8);
                }
        }

        void loop() {
        }
```



음 길이는 생략 가능하다.
만약 생략하는 경우 계속 소리가 나기 때문에
noTone 명령어를 사용해 꺼줘야 한다.

피에조 스피커와 버튼을 이용하여 디지털 피아노 만들기



코드

OneKick

```
#define NOTE_C4  262          //pitches.h를 사용하지 않고 바로 매크로 상수 선언
#define NOTE_D4  294
#define NOTE_E4  330

int pins[] = { 2, 3, 4 };
int notes[] = { NOTE_E4, NOTE_D4, NOTE_C4};

void setup(){
        for( int i = 0; i < 3; i++){
                pinMode(pins[ i ], INPUT);
        }
}
void  loop() {
        for( int i = 0; i < 3; i++){
                if( digitalRead( pins[i] == HIGH){
                        tone(8, notes[i], 20);
                }
        }
}
```

응용 실습

응용실습

OneKick

- 반음을 포함한 노래를 재생시켜보기
- 피아노 건반의 개수 늘리기
- 버튼대신 다른 것으로 제어해 보기