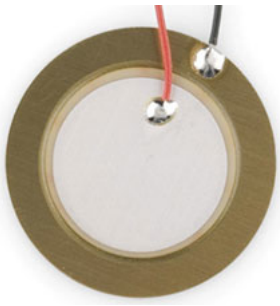
마. 진동 감지 및 소리 출력

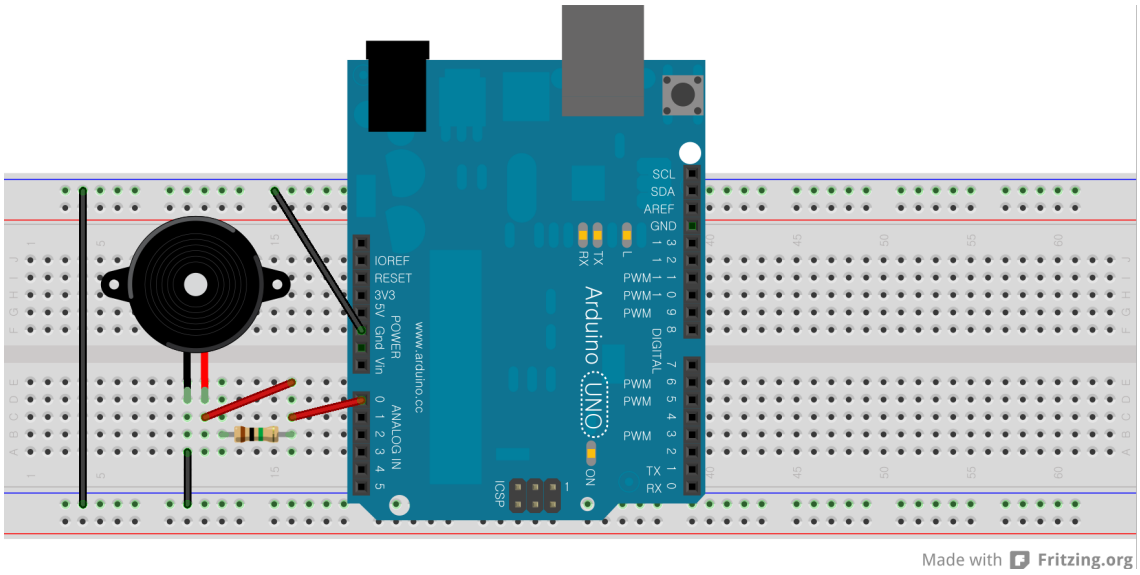
1) 진동 감지하기

피에조 센서는 소리를 내는 출력의 기능 이외에도 충격을 감지하는 센서로의 활용이 가능한다. 이번 프로젝트에서는 외부의 충격을 감지해서 LED를 점멸하는 코드를 작성해보고, 소리를 출력하는 기능은 추후에 작업하도록 한다. 피에조 센서는 압전소자로서, 2개의 면에 전압을 가하면 전압에 비례한 변형이 발생하여 소리가 나거나, 압력이나 비틀림이 주어지면 반대로 전압이 발생되는 소자이다.

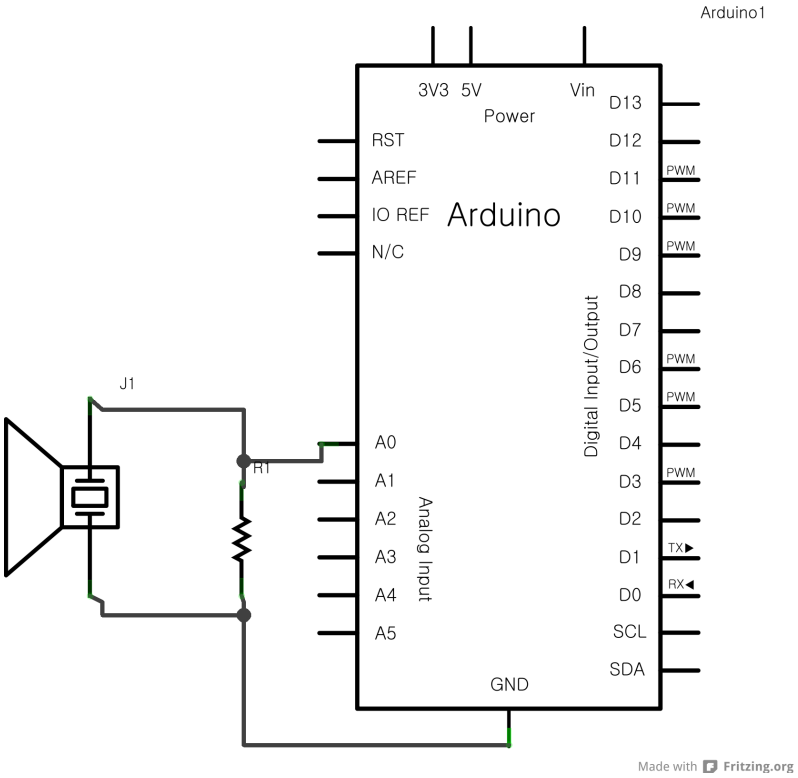


브레드보드

- 피에조는 극성을 띄고 있으므로, 극성에 주의해서 연결한다. 일반적으로 빨간색이 +임을 명시한다. 센서에는 1M옴의 높은 저항을 병렬로 연결해주고 아날로그 입력에 연결을 해준다. 아날로그 입력 역시 전압의 변화 값으로 읽혀지게 되며 0~5V 사이의 변화 값은 10비트 해상도에 따라 0~1023의 정수 값으로 변환되어 입력이 된다. 기준치를 넘어서는 압력이 가해지면 아두이노에 연결되어 있는 13번 LED를 On 시킨다. 13번 포트는 기본적으로 아두이노의 보드에 연결된 소형 LED에 연결이 되어 있다.



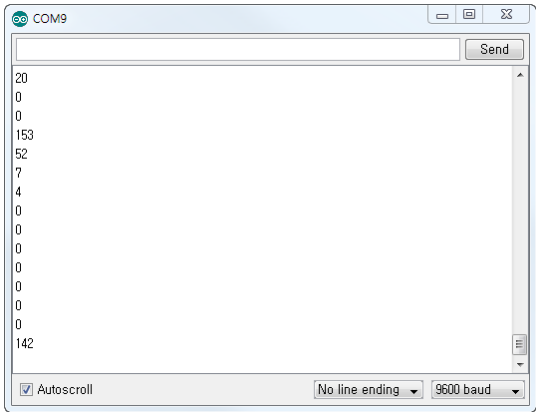
스케메틱



프로그램

```
int sensorPin = 0;
void setup()
{
    Serial.begin(9600);
}

void loop()
{
   int val = analogRead(sensorPin);
   Serial.println(val);
}
```



2) 도전과제

특정 수치를 넘어서면 LED를 연결하여 가시화 시켜보자.

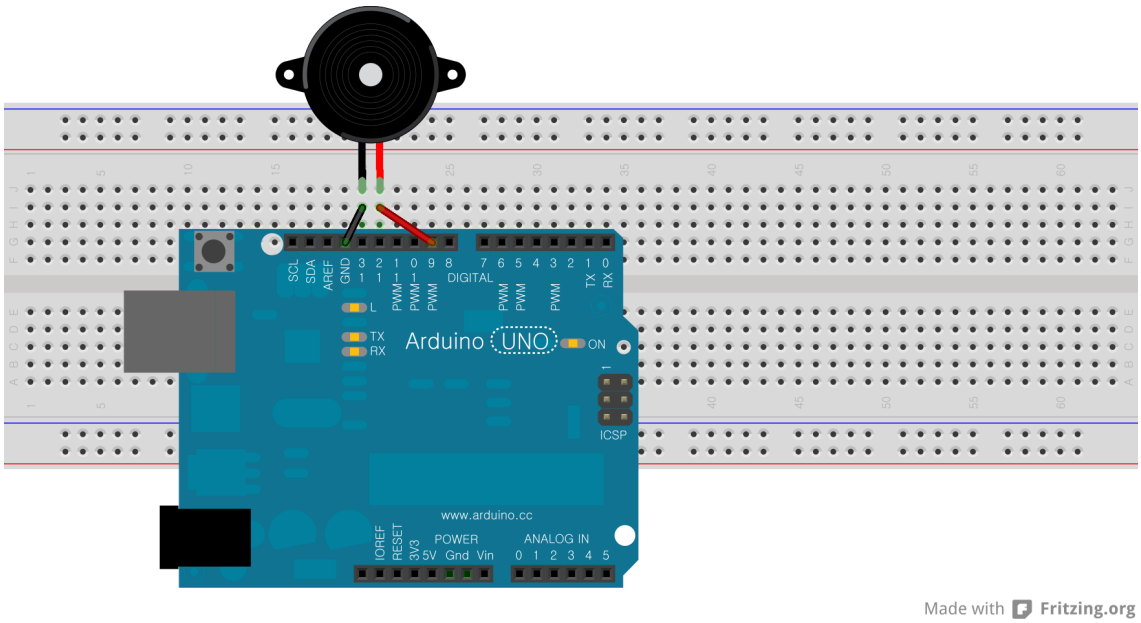
옆 사람과 2개 이상의 진동 감지 센서를 이용하여 다중의 LED를 가시화 시켜보자.

3) 소리 출력하기

이전 프로젝트에서는 피에조 센서를 입력으로 사용했다. 피에조 센서는 출력으로 소리를 낼 수 있으며 아두이노는 스피커와 같은 출력 장치를 통해 사운드를 출력할 수 있다. 소리는 공기의 진동을 통해서 만들어지게 되며, 아두이노는 펄스를 입력 받아 소리를 발생시키는 소형 세라믹 변환기로 이루어진 피에조 장치를 통해서 소리를 만들어 낼 수 있다. 소리의 주파수는 스피커에서 한 차례 진동이 발생하는 시간에 따라 결정되며, 시간이 짧을수록 주파수는 높아진다. 동일한 하드웨어를 변경하여 A0에 들어오는 임의의 값을 스피커로 출력해 보자. 아날로그 출력에 관한 것은 아날로그 출력 프로젝트에서 더 세부적으로 다룬다.

브레드보드

- 피에조는 극성을 띄고 있으므로, 극성에 주의해서 연결한다. 일반적으로 빨간색이 +임을 명시한다. 아두이노의 9번 포트에 피에조의 +극을 연결해준다. 아두이노의 라이브러리가 제공해주는 함수를 사용하여 소리를 출력한다. 보다 안정된 동작을 위하여 피에조의 + 측에 100옴 정도의 저항을 연결해 준다. 아래 그림에서는 표시가 되어 있지 않다.



```
const int speakerPin = 9;
const int sensorPin = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(sensorPin);
  int frequency = map(sensorValue, 0, 1023, 100, 5000);
  int duration = 250;
  tone(speakerPin, frequency, duration);//스피커핀, 주파수, 재생시간(milli second)
  Serial.println(sensorValue);
  delay(100);

}
```

위 코드는 아날로그 입력에서 들어오는 의미 없는 값을 통해서 소리를 재생시킬 수 있는 주파수 성분을 선정하는 코드이다. 물론 듣기 좋은 음악은 아니다.

가) 간단한 멜로디 재생하기

```
const int speakerPin = 9; // connect speaker to pin 9

char noteNames[] =       {'C','D','E','F','G','a','b'};
unsigned int frequencies[] = {262,294,330,349,392,440,494};
const byte noteCount = sizeof(noteNames); // the number of notes
                                          // (7 in this example)
//notes, a space represents a rest
char score[] = "CCGGaaGFFEEDDC GGFFEEDGGFFEED CCGGaaGFFEEDDC ";
const byte scoreLen = sizeof(score); // the number of notes in the score

void setup()
{
}

void loop()
{
  for (int i = 0; i < scoreLen; i++)
  {
    int duration = 333;  // each note lasts for a third of a second
    playNote(score[i], duration); // play the note
  }

  delay(4000); // wait four seconds before repeating the song
}

void playNote(char note, int duration)
{
  // play the tone corresponding to the note name
  for (int i = 0; i < noteCount; i++)
  {
    // try and find a match for the noteName to get the index to the note
    if (noteNames[i] == note) // find a matching note name in the array
      //  play the note using the frequency:
      tone(speakerPin, frequencies[i], duration);
  }
  // if there is no match then the note is a rest, so just do the delay
  delay(duration);
}
```