

Chapter 10. 스피커와 센서

마이크로컨트롤러를 디지털 데이터만을 처리할 수 있지만 ADC를 통해 아날로그 신호를 디지털 신호로 변환하여 받아들일 수 있으며, PWM 신호를 사용하여 아날로그 신호를 출력하는 것과 유사한 효과를 얻을 수 있다. 이 장에서는 아날로그 신호를 출력하는 예로 스피커를 통해 음을 재생하는 방법과 아날로그 신호를 입력하는 예로 센서 신호를 읽어 들이는 방법에 대해 알아본다.

1. 스피커 출력

아두이노 메가2560에 사용된 ATmega2560 마이크로컨트롤러는 아날로그 신호 출력이 불가능하지만 PWM 신호를 사용하여 아날로그 신호와 유사한 효과를 얻을 수 있다. 아날로그 신호 출력한 비슷한 효과를 내는 예는 아두이노의 tone 함수에서 찾아볼 수 있으며 tone 함수는 PWM 신호를 사용하여 스피커로 단음을 재생할 수 있도록 해준다. tone 함수는 50% 듀티 사이클과 지정된 주파수를 가지는 구형파(square wave)를 사용하여 단음을 재생한다.

```
void tone(uint8_t pin, unsigned int frequency, unsigned long duration = 0)
```

- 매개변수

pin : 핀 번호

frequency : 출력 주파수 (unsigned int)

duration : 출력 지속시간, 밀리초 단위 (unsigned long)

- 반환값 : 없음

tone 함수를 통해 음을 재생할 때 특정 시간에는 하나의 음만을 재생할 수 있다. 다른 핀을 통해 이미 음이 재생 중이면 tone 함수를 호출하여도 원하는 소리를 들을 수 없으며 동일한 핀을 통해 tone 함수를 다시 호출하면 새롭게 지정된 주파수의 톤이 재생된다. tone 함수는 재생 시간을 지정할 수 있으며 재생 시간이 지정되지 않으면 디폴트값인 0이 사용되어 noTone 함수가 호출될 때까지 음의 재생이 계속된다. noTone 함수는 tone 함수에 의한 구형파 출력을 멈추어 음의 재생을 멈춘다.

```
void noTone(uint8_t pin)
```

매개변수

pin : 핀 번호

반환값 : 없음

tone 함수를 사용하기 위해서는 각 음높이에 해당하는 주파수를 알아야 한다. 각 음 높이에 대한 정의는 인터넷에서 다운 받을 수 있다.¹⁾ 다운받은 'pitches.h' 파일은 스케치북 디렉터리 아래 'libraries' 디렉터리에 'pitches' 이름의 디렉터리를 만들고 그 아래에 복사한다. 라이브러리가 설치되었다면 '스케치 → Include Library' 메뉴에 'pitches' 항목이 추가되어 있음을 확인할 수 있다. pitches 항목을 선택하면 pitches 라이브러리를 사용하기 위한 헤더 파일이 자동으로 포함된다.

```
#include <pitches.h>
```

학습보드에는 단음 재생을 위해 스피커가 아날로그 3번 A3 핀(디지털 57번)에 연결되어 있다.

1)

<http://code.google.com/p/arduino/source/browse/trunk/build/shared/examples/Digital/toneMultiple/pitches.h?r=943>

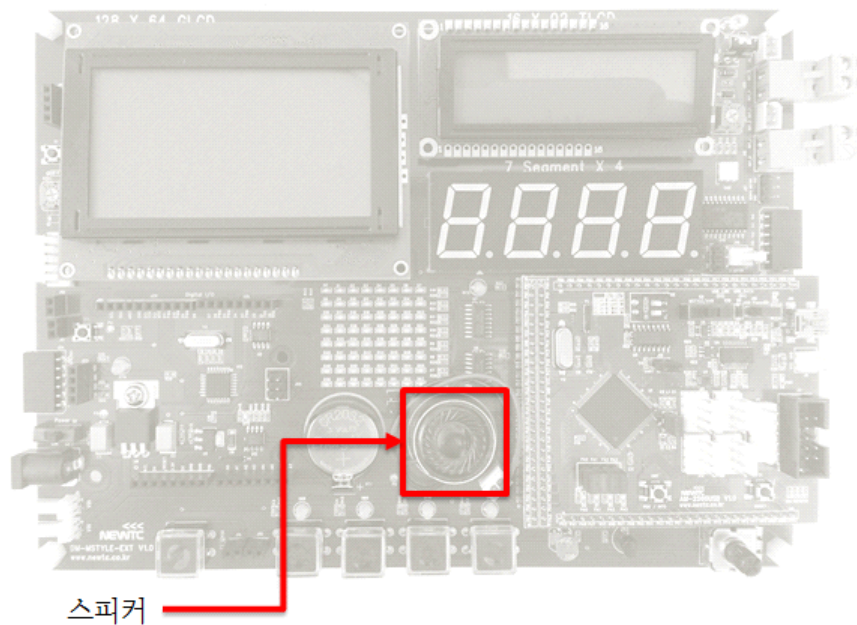


그림 1. 학습보드에 포함된 스피커

스케치 1은 학교종 멜로디를 재생하는 스케치의 예이다. 스케치를 간단하게 하기 위해 음의 길이는 모두 동일하게 설정하였다.

스케치 1. 멜로디 재생

```
#include <pitcher.h>

int speakerPin = 57;           // 스피커 연결 핀

int melody[] = {               // 학교 종 멜로디. 0은 침표
    NOTE_G4, NOTE_G4, NOTE_A4, NOTE_A4, NOTE_G4, NOTE_G4, NOTE_E4, 0,
    NOTE_G4, NOTE_G4, NOTE_E4, NOTE_E4, NOTE_D4, 0,
    NOTE_G4, NOTE_G4, NOTE_A4, NOTE_A4, NOTE_G4, NOTE_G4, NOTE_E4, 0,
    NOTE_G4, NOTE_E4, NOTE_D4, NOTE_E4, NOTE_C4, 0
};

int noteDuration = 4;          // 4분 음표

void setup() {
```

```
}

void loop() {
  for (int thisNote = 0; thisNote < sizeof(melody) / sizeof(int); thisNote++) {
    int noteLength = 1000 / noteDuration;      // 음표 길이를 시간으로 변환
    tone(speakerPin, melody[thisNote], noteLength);  // 멜로디 재생

    delay(noteLength);

    noTone(speakerPin);                        // 현재 음 재생 중지
  }

  while(true);                                // 1회 재생 후 대기
}
```

2. 온도 센서

스피커를 통한 음의 재생이 PWM 신호를 통해 아날로그 출력을 얻는 예라면 센서를 통한 데이터 입력은 아날로그 신호를 디지털로 변환하는 예이다. 학습보드에는 온도 센서와 조도 센서가 아날로그 1번 (디지털 55번) 핀과 아날로그 2번 (디지털 56번) 핀에 연결되어 있다.

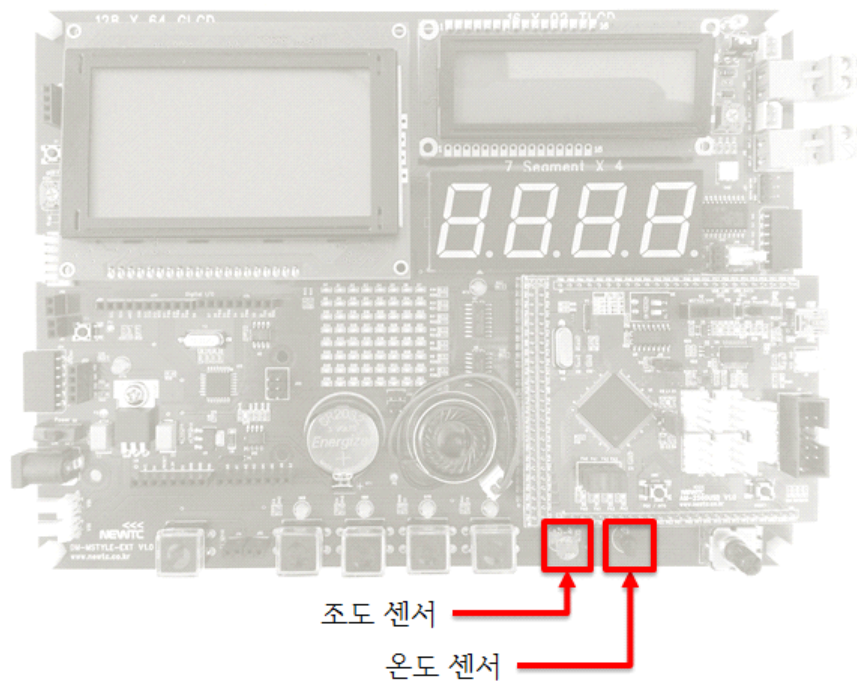


그림 2. 학습보드에 포함된 온도 센서와 조도 센서

먼저 온도 센서의 사용 방법을 살펴보자. 흔히 사용되는 LM35 온도센서는 트랜지스터와 동일한 형태를 가지고 있다.

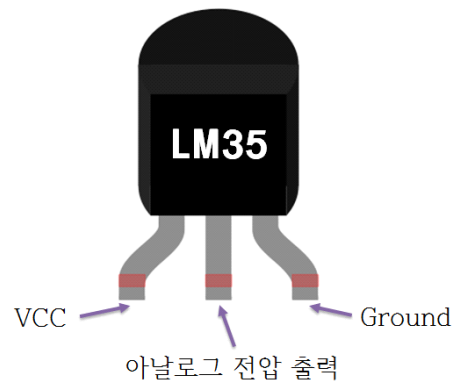


그림 3. LM35 온도센서

LM35 온도 센서는 온도가 1°C 증가할 때마다 10mV 출력 전압이 증가한다. 0°C에서 출력 전압은 0V이므로 ADC 값을 전압으로 변환한 후 100을 곱하면 현재 온도를 얻을 수 있다. 스케치 2는 온도 센서의 값을 읽어 시리얼 모니터로 출력하는 스케치의 예이다.

스케치 2. 온도 센서

```
void setup(){
  Serial.begin(9600);          // 직렬포트 초기화
}

void loop(){
  int reading = analogRead(55);    // 온도 센서 읽기

  // ADC 반환 값을 전압으로 변환
  float voltage = reading * 5.0 / 1024.0;
  Serial.print(voltage);
  Serial.print(" V : ");

  // 전압에 100을 곱하면 섭씨 온도를 얻을 수 있다.
  float temperatureC = voltage * 100;
  Serial.print(temperatureC);
  Serial.print(" C, ");

  // 화씨 온도로 변환
  float temperatureF = (temperatureC * 9.0 / 5.0) + 32.0;
  Serial.print(temperatureF);
  Serial.println(" F");

  delay(1000);
}
```

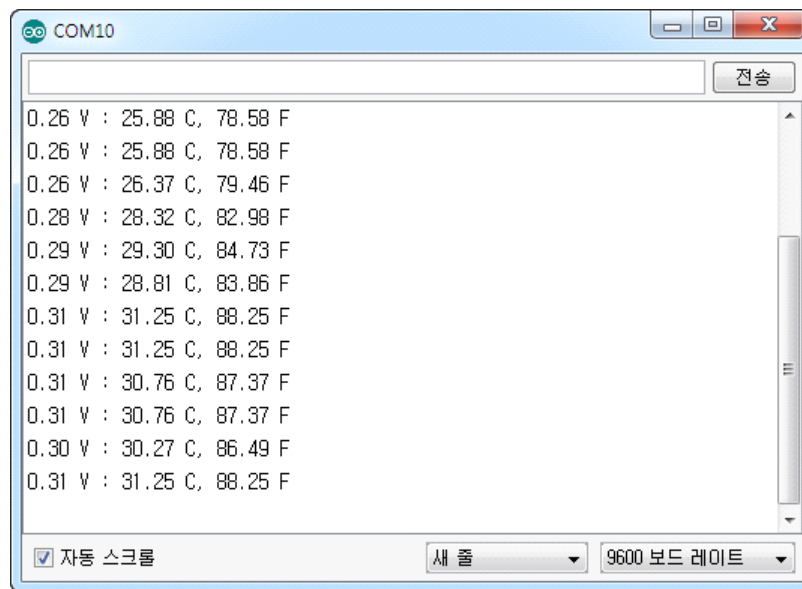


그림 4. 스케치 2 실행 결과

3. 조도 센서

빛의 양에 따라 물리적인 특성이 변하는 소자에는 포토레지스터(photo resistor), 포토다이오드(photo diode), 포토트랜지스터(photo transistor) 등이 있다. 이 중 흔히 사용되는 소자는 빛의 양에 따라 저항값이 변하는 포토레지스터로 흔히 광센서라고 불린다. 광센서는 CdS 광센서가 주로 사용되며, 카드뮴(Cd)과 황(S)을 화합하여 만들어지는 황화카드뮴 결정에 금속 다리를 결합하여 만들어진단.



그림 5. CdS 광센서

광센서는 광량에 반비례하는 저항값을 가진다. 즉, 가시광선이 없는 어두운 곳에서는 큰 저항

값을 가지다가 가시광선이 닿으면 저항값이 작아진다. 광센서는 가격이 저렴하고 사용법이 간단하지만, 반응속도가 느리고 광량에 따른 출력 특성이 정밀하지 않으므로 광량을 정밀하게 측정하고자 하는 경우에는 포토다이오드나 포토트랜지스터를 사용하여야 한다. 광센서는 저항값이 변하므로 전압 분배에 의해 광량에 따라 전압이 달라지도록 연결한다.

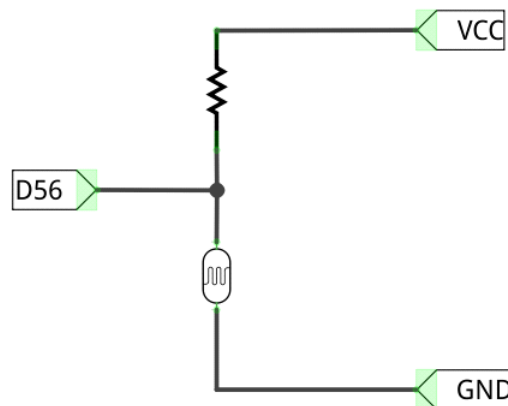


그림 6. 광센서 연결 회로도

스케치 3은 광센서에 인가되는 전압을 읽어 출력하는 예로 손으로 광센서를 가리면서 출력되는 값의 변화를 관찰해 보자.

스케치 3. 조도 센서

```
void setup(){
    Serial.begin(9600);           // 직렬 포트 초기화
}

void loop(){
    int reading = analogRead(56); // 조도 센서 읽기

    Serial.print("ADC : ");
    Serial.print(reading);

    // ADC 반환 값을 전압으로 변환
    float voltage = reading * 5.0 / 1024.0;
    Serial.print(", Voltage : ");
```

```
Serial.println(voltage);  
  
delay(1000);  
}
```

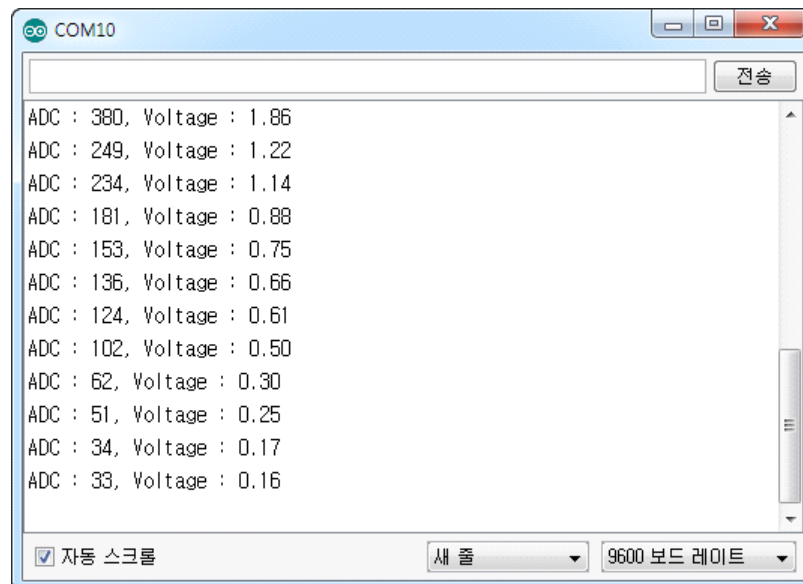


그림 7. 스케치 3 실행 결과