

\* 익스플로링 아두이노  
Exploring Arduino™

Jeremy Blum 지음, 김찬웅 옮김

## 지은이 Jeremy Blum 제레미 블럼

코넬대학교에서 전기컴퓨터 공학을 전공하고 현재 Google의 구글 글래스팅에서 하드웨어 엔지니어로 근무 중이다. 3D 프린터, 의수, 동작 인식 장갑과 같은 마이크로컨트롤러 기반 제품들을 직접 디자인하였다. 또한 유명한 아두이노 튜토리얼 시리즈를 비롯하여 다양한 엔지니어링 강의를 유튜브에 올려 전 세계 수백만 독자를 거느리고 있는 엔지니어링계의 유튜브 스타이기도 하다.

## 기술 감수 Scott Fitzgerald 스캇 피츠제럴드

2006년부터 교수 도우 및 개인 프로젝트에서 아두이노 플랫폼을 사용해 온 아티스트이자 뉴욕대학의 ITP에서 물리 컴퓨팅을 강의하는 교육자이다. 또한 아두이노 팀에서 아두이노에 관한 문서 작성 및 소개 튜토리얼을 제작하고 있다. 2011년 Making Things Talk 개정판의 기술 감수를 맡았고, 2012년 공식 아두이노 스타터 키트 교재를 집필했다.

## 옮긴이 김찬웅 chankim@snu.ac.kr

전직 소프트웨어 엔지니어이자 교육자로 몇 권의 프로그래밍 서적을 집필했다. 한국을 비롯하여 미국, 싱가포르, 말레이시아, 대만, 홍콩 등에서 학생, 엔지니어, 공무원 등을 가르쳤다. 현재는 그 경험을 바탕으로 개발도상국에서 피지컬 컴퓨팅을 이용한 IT 교육을 실현하려고 공부 중이다.

## 번역본 기술 감수 허경 khur@ginue.ac.kr

IEEE 시니어 멤버로 전자공학을 공부한 연구자이자 교육자이다. 삼성종합기술원 연구원을 거쳐 현재는 경인교육대학교 컴퓨터교육과에서 학생들을 가르치고 있다.

## IT CookBook, 익스플로링 아두이노 Exploring Arduino™

초판발행 2014년 05월 19일

지은이 제레미 블럼(Jeremy Blum) / 옮긴이 김찬웅 / 펴낸이 김태현  
펴낸곳 한빛아카데미(주) / 주소 서울시 마포구 잔다리로7길 16 한빛아카데미(주)  
전화 02-336-7195 / 팩스 02-336-7199  
등록 2013년 1월 14일 제2013-000013호 / ISBN 979-11-5664-107-0 93000

책임편집 김현용 / 기획 김이화 / 편집 김이화  
디자인 표지 더그라프, 내지 여동일  
영업 이윤형, 길진철, 유재욱, 김태진, 주희 / 마케팅 김호철

이 책에 대한 의견이나 오타자 및 잘못된 내용에 대한 수정 정보는 한빛아카데미(주)의 홈페이지나 아래 이메일로 알려주세요. 잘못된 책은 구입하신 서점에서 교환해 드립니다. 책값은 뒷표지에 표시되어 있습니다.

홈페이지 [www.hanbit.co.kr](http://www.hanbit.co.kr) / 이메일 [question@hanbit.co.kr](mailto:question@hanbit.co.kr)

Copyright © 2013 by John Wiley & Sons, Inc., Indianapolis, Indiana All Rights reserved.  
This translation published under license with the original publisher John Wiley & Sons, Inc.  
Korean edition copyright © 2014 by HanbitAcademy Inc.  
이 책의 한국어 판 저작권은 대니홍 에이전시를 통한 저작권사와의 독점 계약으로 한빛아카데미에 있습니다.  
신저작권법에 의해 국내에서 보호를 받는 저작물이므로 무단전재와 복제를 금합니다.

지금 하지 않으면 할 수 없는 일이 있습니다.  
책으로 펴내고 싶은 아이디어나 원고를 메일([writer@hanbit.co.kr](mailto:writer@hanbit.co.kr))로 보내주세요.  
한빛아카데미(주)는 여러분의 소중한 경험과 지식을 기다리고 있습니다.

한빛아카데미(주)는 한빛미디어(주)의 대학교재 출판 부문 자회사입니다.

# 1시간 강의를 위해 3시간을 준비하는 마음!

## 군더더기 없는 핵심 원리 + 말랑말랑 쉬운 콘텐츠

핵심 원리 하나만 제대로 알면 열 가지 상황도 해결할 수 있습니다.  
친절한 설명과 명확한 기승전결식 내용 전개로 학습 의욕을 배가시켜줍니다.

## 핵심 원리 → 풍부한 예제와 연습문제 → 프로젝트로 이어지는 계단 학습법

기본 원리를 다져주는 예제, 본문에서 배운 내용을 촘촘하게 점검해볼 수 있는 연습문제,  
현장에서 바로 응용할 수 있는 프로젝트를 단계별로 구성해 학습의 완성도를 높였습니다.

## 학습욕구를 높여주는 현장 이야기가 담긴 IT 교과서

필드 어드바이저의 인터뷰와 주옥 같은 현업 이야기를 담았습니다.  
강의실 밖 현장의 요구를 접하는 기회를 제공하고,  
학생들 스스로 필요한 공부를 할 수 있도록 방향을 제시합니다.



## 아두이노 세계로 항해를 시작한 여러분에게

여러분은 완벽한 타이밍을 만났다. 사람들이 자주 말하는 것처럼 “우리는 미래에 살고 있다.” 이 책에서는 여러 가지 내용을 다루는데, 여러분은 주위의 다양한 도구를 사용하여 주변 환경을 바꿀 수 있는 기회와 능력을 갖추게 될 것이다. 불과 몇 년 전만 해도 전공자가 아니면 적절한 마이크로컨트롤러를 선택하여 몇 분 안에 주변을 제어하는 일은 거의 불가능했다. 하지만 이제는 아두이노에 간단한 명령을 입력하면 복잡한 시스템도 제어할 수 있다. 여러분이 전자, 프로그래밍, HCI, 예술, 제어 시스템 세계로 탐험을 시작하는 데 아두이노가 사용하기 좋은 가장 친근한 도구가 될 것이라 믿는다. 이 책을 통해 여러분은 아두이노를 사용한 동작 감지부터 인터넷으로 통신하는 무선 제어 시스템 제작 방법 등을 배울 것이다. 아두이노를 사용하여 무엇을 만들든 그 가능성은 무한하다.

독자 여러분이 공학을 처음 접하는 사람이든 임베디드 시스템 디자인을 시작하려는 베테랑 개발자이든, 아두이노는 모두에게 적합한 플랫폼이다. 만약 여러분이 아두이노를 사용하여 무언가를 개발하는 데 참고서가 필요하다면 이 책이 도움이 될 것이다. 특정 프로젝트들을 함께 제작해보는 과정에서 쓸만한 코드 작성 방법, 시스템 구조 등을 익히기에도 충분하다. 이 책을 읽는 동안 독자 여러분이 배우게 될 전기공학, 시스템 디자인, 컴퓨터 프로그래밍 관련 지식들은 아두이노가 아닌 다른 플랫폼을 사용하는 프로젝트를 수행할 때에도 도움이 될 것이다.

### 감사의 글

우선 이 책을 출간할 수 있게 도와준 Wiley 출판사의 친구들에게 감사의 말을 전한다. Mary James는 이 책을 쓸 수 있게 용기를 주었고, Jennifer Lynn은 내가 이 책을 완성할 수 있도록 도와주었다. 또 날카로운 눈으로 기술 감수를 봐준 Scott Fitzgerald에게도 감사의 말을 전한다.

Element14의 멋진 친구들이 아니었다면, 이 책의 전초가 된 아두이노 튜토리얼 시리즈는 제작되지 못했을 것이다. 특히 Sabrina Deitch와 Saga Jethani와 함께 일할 수 있었던 것은 큰 영광이었다. 나는 이 책을 집필하는 동시에 석사학위를 마쳤고, 두 개의 회사를 운영했다. 나의 모든 책무들 사이에서 균형을 잡으려 노력하는 동안 참고 기다려 준 교수님들과 동료들에게 큰 감사를 드린다.

마지막으로 끊임없는 응원으로 내가 왜 이 일들을 하고 있는지 상기시켜 준 우리 가족, 특히 부모님, 동생 David에게 고맙다고 말하고 싶다.

## 한국 독자들에게

아두이노 플랫폼을 사용하는 흥미진진한 세계를 한국의 독자 여러분께 소개할 수 있게 되어 매우 기쁩니다. 저는 아두이노를 활용한 임베디드 디자인, 컴퓨터 공학, HCI, 전기공학 등의 개념을 유튜브를 통해 전파해 왔으며, 전 세계 수백만 명이 제 동영상을 통해 아두이노를 접했습니다. 더 많은 독자와 나이를 불문하고 이제 막 아두이노 프로토타이핑 플랫폼을 공부하려는 분들에게 필요한 지혜와 기법을 좀 더 깊게 전달하고 싶다는 제 오랜 소망을 드디어 『Exploring Arduino』 출간과 함께 현실화하게 되었네요.

아두이노 플랫폼은 적절한 도구를 사용하고, 관련된 지식을 적용하면 누구나 놀라운 것들을 만들어 낼 수 있습니다. 그리고 이런 작품을 제작하는 데 『Exploring Arduino』 한국어판이 도움을 줄 수 있다니 정말 신납니다. 책에서 다루는 실습 예제와 학습 주제들은 흥미롭고 유용하면서도 예술적인 전자 프로젝트를 진행하는 데 필요한 기술을 가르쳐 줄 것입니다.

여러분도 아두이노와 시작하는 여정을 인터넷에서 저를 비롯한(트위터@sciguy14) 전 세계 아두이노 커뮤니티와 함께 공유해 보세요. 아두이노와 함께 하는 작업이 한층 즐거워지고 신선한 자극을 받을 수 있을 것입니다. 여러분의 아두이노 작품을 만날 수 있기를 기대합니다!

한국 독자들의 행운을 기원하며,  
미국 샌프란시스코에서 제레미 블럼

## 궁극의 아두이노 피지컬 컴퓨팅 교재

영국에서는 2014년 9월부터 컴퓨터 프로그래밍 과정이 초중고에서 의무교과로 지정된다. 우리나라에서도 미래창조과학부와 교육부가 프로그래밍 교육의 의무교과 지정을 놓고 시범연구를 진행하고 있다. 코딩 교육을 창의성 교육의 선도적 역할로 이해한다고는 하지만, 우리는 프로그래머를 어떤 직업이라고 생각할까?

스마트폰 화면과 컴퓨터 모니터에서만 구현되는 소프트웨어 프로그래밍 따라하기가 지겨워졌다면, 아두이노, 라즈베리파이 등을 이용해 생각하는 대로 무엇이든 만들 수 있는 피지컬 컴퓨팅의 세계로 눈을 돌려 보자. 영국 프로그래밍 교육의 핵심에는 영국에서 만든 라즈베리파이가 있다. 그리고 전 세계적으로는 메이커들에게 훨씬 인기가 많은 아두이노가 있다. 피지컬 컴퓨팅의 미덕은 단 한 가지이다. 남을 따라할 것인가? 아니면 내가 만들 것인가?

아두이노를 비롯한 피지컬 컴퓨팅은 이미 우리 생활 곳곳에 들어와 있다. 아두이노와 같은 종류의 마이크로컨트롤러는 여러분이 매일 만지는 냉장고를 비롯한 가전제품은 물론 지하철의 스크린도어, 리얼리티 쇼에 자주 등장하는 헬리캠에도 사용된다. 송호준 씨가 만든 세계 최초의 개인용 인공위성도 아두이노로 만든 것이다.

무한한 가능성을 발견하고 마이크로컨트롤러의 세계에 뛰어들었지만 무엇을 어떻게 해야 할지 모르겠는가? 그런 여러분에게 『Exploring Arduino』는 진지한 궁극의 해답을 줄 수 있을 것이다. 이 책은 이제까지 출판된 어떤 아두이노 서적보다도 방대한 분야를 공학적 관점에서 최대한 쉽게 설명한다. 그냥 따라 해보고 끝나는 게 아닌 여러분의 머릿속에 살아 있는 아이디어 창고를 만들어 줄 것이다.

원래 였긴이도 아두이노 교재 집필을 준비하고 있었지만, 유튜브의 튜토리얼을 통해 이미 알고 있던 제레미 블럼이 아두이노 책을 출간했다는 소식을 듣자마자 책을 구입하여 읽고 나서는 집필을 포기하고 번역을 결심했다. 집필을 포기하는 대신 번역을 선택할 만큼 이 책이 지닌 의미는 매우 크다.

책의 주요 대상 독자는 관련 전공자나 아두이노에 관심이 있는 대학생이나 전자·컴퓨터를 공부하는 고등학생이다. 비록 책이 나이 어린 학생들이 읽기에는 다소 어려운 내용과 용어들을 포함하고 있으나, 관련 정보를 조금만 검색해 보면 쉽게 이해할 수 있을 것이다. 청소년과학탐구대회 출전을 준비했던 초등학교 고학년부터 식물에 물을 주러 옥상까지 올라가기 귀찮아서 로봇 정원을 만들고 싶어 하는 50대 아저씨까지 누구나 볼 수 있는 책이다. 공부는 하고 싶을 때 하는 것이지 때가 따로 있는 것은 아니다.

였긴이 김찬용 

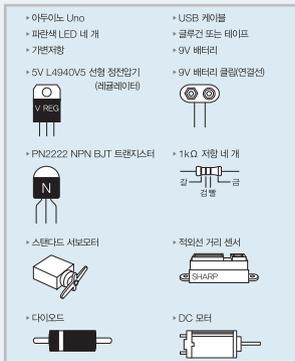
## 이 책을 읽기 전에

### 누구를 위한 책인가

이 책은 아두이노를 다루고 싶어하는 모든 수준의 독자를 대상으로 한다. 각 장들은 서로 연결되어 있고, 이전 장에서 배운 개념과 프로젝트 구성을 이용하여 더 깊이 있는 내용으로 발전된다. 걱정은 마시라. 새롭고 복잡한 개념을 만날 때마다 연관된 개념을 어디에서 처음 접했는지 알려줄 것이다.

이 책은 독자가 프로그래밍이나 전기공학에 대한 사전 경험이 거의 없다고 가정한다. 다양한 수준의 독자들이 더 자세하게 개념을 이해할 수 있도록 여러 구성 요소를 사용해 설명했다. 이 책에서 설명한 기술적인 사항들은 아두이노 사용법을 익히려면 반드시 알아야 하는 필수 내용은 아닐 수도 있지만 아두이노 기반 프로젝트를 이해하고 확장시키려면 알아두어야 한다.

## 이 책의 구성요소



### 사용 부품

이 장의 실습을 따라하는 데 필요한 부품들을 소개한다. 찾기 어려운 부품도 쉽게 찾을 수 있도록 특징도 함께 표기했다.

노의 PWM 출력 핀에 연결한 후 EN 핀을 빠른 속도로 켜고 고

- **3V, 4V(11번, 14번 핀)**: 이 핀들은 오른쪽 드라이버용 출력 핀이므로 이 핀들은 연결하지 않는다.
- **3A, 4A(10번, 15번 핀)**: 이 두 핀으로 오른쪽 드라이버의 출력 핀이므로 이 핀들은 연결하지 않는다.
- **3, 4EN(9번 핀)**: 이 핀은 오른쪽 H-브리지 드라이버를 제어하지 않으므로 이 핀을 GND에 직접 연결하여 작동하지 않도록 한다.

[그림 4-7]처럼 회로를 연결하고, 가변저항은 이전 장

**WARNING** H-브리지 내의 회로가 합선되는 것을 방지하려면

프로그래밍으로 모터를 회전시키기 전에 회로의 연결

서 1, 2EN(1번 핀)을 5V 전원에 연결하고 1A 핀은

로가 정상적으로 작동하는지 확인할 수 있다. 반대로

하면 모터가 반대 방향으로 회전할 것이다.

### Warning

책에서 주의 표시를 발견하면 특히 조심해야 한다. 프로젝트의 특정 단계에서 잘못 작동될 경우 전자회로가 손상될 가능성이 있다.

마지막으로 아두이노 보드가 컴퓨터의 어떤 포트의 [도구] 메뉴-[시리얼 포트]에서 연결된 포트를 찾

COM# 포트가 보일 것이다. #은 시리얼 포트 번호를

컴퓨터에 여러 시리얼 장치가 연결되었을 때는 아두이노 보드의 USB 커넥터

인한 후 다시 연결하여 선택하는 방법도 있다.

맥이나 리눅스에서는 연결된 시리얼 포트가 '/dev/serial#'로 표시된다. \*은 임의의 알파벳이나 숫자(들

**Tip** 마이크로컨트롤러(Microcontroller) 또는 MCU(Microcontroller Unit)는 프로세서와 메모리, 프로그래밍으로 제어

두 포함한 장치이다. 마이크로프로세서는 주로 개인용 컴퓨터

로컨트롤러는 무선 리모컨, 자동차의 엔진 제어(ECU), 가전제품 등이다.

### Tip과 Note

간단한 용어 설명과 유용한 힌트 및 사용 팁, 각장의 실습을 쉽게 따라할 수 있도록 관련 자료 소개 및 부가 정보를 제공해 준다.

## 이 책에서 배울 내용들

아두이노로 직접 실습하면서 전기공학, 컴퓨터 공학, 제품 디자인(설계), 고수준의 사고를 함께 익힐 수 있다. 만약 독자 여러분이 각 부분의 원리를 구체적으로 살펴보지 않고 각 단계별로 지시문만 따라 하기 원한다면, 이 책은 적합하지 않다.

이 책에 제시된 아두이노 프로젝트의 하드웨어 제작을 통해 부품을 연결하는 방법뿐만 아니라 회로도를 읽는 방법, 데이터시트를 읽는 방법, 특정 기능을 구현하는 데 필요한 부품이 무엇인지 함께 배울 수 있다. 프로그램 작성도 마찬가지이다. 물론 전체 코드를 제공하겠지만, 먼저 몇 번의 중간 과정을 통해 프로그램을 완성한다. 자주 사용하는 프로그램 함수에 익숙해지고, 좋은 코드 작성법도 체득할 수 있을 것이다. 알고리즘에 대한 이해도 더 깊어질 것이고, 물리학의 개념과 디지털 디자인 원칙도 함께 익힐 수 있다.

이 책의 프로젝트를 수행하면서 아두이노 프로그래밍을 더 깊이 이해하며, 더 복잡한 전기 시스템 개발 및 여러 분야에서 다른 플랫폼을 활용할 때에도 필요한 기본 기술을 갖출 수 있게 되기 바란다.

여기서 잠깐

### H-브리지로 간단한 회로 만들기

H-브리지를 사용할 때 고려해야 할 중요한 한 가지가 있다. 왼쪽의 두 스위치가 일직선으로 연결되면서 9V 전원이 아무런 저항도 가지지 않은 채 그대로 전지 합선을 결합한 적이 있다면, 아바도 다시는 똑같은 경험을 하고 싶지 않 매우 빨리 과열될 것이고, 드물지만 폭발하거나 누액이 흐르기도 한다. 더 나쁜 경우를 생각해 보자. 이 실습에서 사용하는 H-브리지인 SN754410 Qd 합선으로 평가되기 전에 회로를 차단하는 과열감지 센더를 장착해 내장되어 있어 하드웨어의 일부만을 평가하려는 사례가 H-브리지에서는 일어나지 않도록 주의해야 한다.

### 8. H-브리지 회로 연결

만 H-브리지 두 개를 합치면 [그림 4-5]처럼 완전한 풀 H-브리지에서는 DC 모터 하나를 구동하려고 한 H-브리지 두 개를 사 H-브리지는 SN754410으로, 하나의 IC에 만 H-브리지 네 개가 나

```
int(temperature); //온도 출력
int("); //원표( ) 출력
int(brightness); //조도 출력 후 결바꿈
//기후 순서값 증가

D, LOW; //제어기 출력 중이 아니면 LED 끄

\n last);

digitalRead(BUTTON); //버튼값을 읽음
//현재 버튼값이 어떤 버튼값이 다음

동안 기다림
!Read(BUTTON); //버튼값을 다시 읽음

현재 버튼값을 반환
```



### 요약

이 장에서 배운 것을 요약하면 다음과 같다.

- 아두이노는 USB-시리얼 변환기를 사용하여 컴퓨터에 연결
- 아두이노 보드는 전용 IC를 사용하거나 내장 USB 기능을 사
- 아두이노는 USB 시리얼을 연결하여 컴퓨터로 데이터를 출력
- 플랫폼 문자(n)와 탭(\n) 등 특수 문자를 사용하면 시리얼 데

### 한 걸음 더

이 장에서 배운 내용을 바탕으로 다음 프로젝트들 스

- 아두이노 출력을 감지 시스템을 만들고 출력이 감지될 Processing 프로그램
- 비디오 게임에 연결하여 게임을 조종할 수 있는 아두이노 Le
- 아두이노에 연결된 LED 수십 개로 Processing GUI 프

## 여기서 잠깐

현재 주제나 관련 내용에 대한 보다 깊이 있는 내용을 알려준다.

## QR코드

실행 결과 및 관련 동영상은 QR코드로 확인할 수 있다. 예제 번호와 함께 실행 결과가 등장하는 시간도 함께 표시하였다.

## 요약과 한 걸음 더

각 장에서 다룬 내용을 요약하고, 배운 내용을 활용해 추가로 해볼 수 있는 프로젝트들을 제시한다.

## 부품 구입

이 책의 실습을 수행하는 데 필요한 부품 목록을 다음 웹사이트에서 다운로드할 수 있다. 국내 사정에 맞게 일부 부품을 대체했다.

[www.hanbit.co.kr/exam/4107](http://www.hanbit.co.kr/exam/4107)

전자/로봇/기계부품 쇼핑몰 디바이스마트에서 필요한 부품만 개별적으로 구매하거나 한 번에 구매할 수 있다.

[www.devicemart.co.kr](http://www.devicemart.co.kr)



상품 페이지 바로가기

[http://www.devicemart.co.kr/goods/exploring\\_arduino.php](http://www.devicemart.co.kr/goods/exploring_arduino.php)

오픈 소스 하드웨어의 장점은 무엇보다도 모든 사람에게 공개한다는 것이다. 아두이노를 비롯한 주변 기기의 회로도가 모두 공개되어 있어 전 세계 누구든 만들어 팔 수 있다. 그중 [www.seeedstudio.com](http://www.seeedstudio.com), [www.hobbyking.com](http://www.hobbyking.com)은 해외 사이트이지만 매우 저렴하게 구입할 수 있다. 그렇더라도 해외에서 물품을 구입하기 전에는 반드시 국내 가격과 비교해보고, 관세 여부를 조사하도록 하자.

## 더 필요한 도구들

전자부품 외에도 몇 가지 도구와 물품들이 더 필요하다. 가장 중요한 것은 아두이노 IDE와 호환되는 컴퓨터이다. 맥 OS X 10.4 이상, 윈도우 XP 이상, 리눅스 디스트로 버전 이상의 운영체제가 설치되어 있어야 한다.

이 책의 실습을 진행할 때 꼭 필요한 것은 아니지만 하드웨어 제작 및 회로 디버깅에 필요한 도구는 다음과 같다. 이 책의 실습뿐만 아니라 다른 프로젝트를 진행할 때도 유용하게 활용할 수 있을 것이다.

- **인두와 뿔납** : 실습을 진행할 때 반드시 필요한 도구는 아니다. 하지만 만능기판에 회로를 구성하거나 납땀이 필요한 아두이노 실드 및 부품을 사용할 경우에는 필요하다.
- **다기능 회로시험기(멀티미터)** : 반드시 필요한 것은 아니지만 회로에 전류나 전압이 제대로 인가되는지 디버깅할 때 매우 유용하다.
- **소형 스크류 드라이버 세트**
- **글루건**

## 예제소스 및 공식 웹사이트

실습에 필요한 예제 소스는 다음 주소에서 내려받을 수 있다.

<http://www.hanbit.co.kr/exam/4107>

저자가 관리하는 책의 웹사이트 주소는 다음과 같다. 관련 부품 정보 및 각 장의 핵심 주제를 다루는 튜토리얼 동영상, 결과 동영상, 회로도, 소스 코드 등이 잘 정리되어 있다. 웹사이트에 접속한 후 상단의 '번역' 링크를 클릭하면 웹페이지가 한국어로 자동 번역된다.

[www.exploringarduino.com](http://www.exploringarduino.com)

## 참고 웹사이트

아두이노 플랫폼으로 프로젝트를 수행하는 동안 여러 가지 질문이나 문제에 부딪칠 것이다. 아두이노를 사용할 때 가장 좋은 점 중 하나는 매우 뛰어난 인터넷 지원 커뮤니티가 있다는 점이다. 이미 왕성하게 활동하는 아두이노 유저들이 다른 사용자를 돕고 있다.

- 공식 아두이노 레퍼런스 홈페이지

[www.arduino.cc/en/Reference/HomePage](http://www.arduino.cc/en/Reference/HomePage)

- 제레미 블럼의 아두이노 튜토리얼

[www.jeremyblum.com/category/arduino-tutorials](http://www.jeremyblum.com/category/arduino-tutorials)

- ardufruit 아두이노 튜토리얼

[learn.adafruit.com/category/learn-arduino](http://learn.adafruit.com/category/learn-arduino)

- 공식 아두이노 포럼

[www.arduino.cc/forum](http://www.arduino.cc/forum)

- element14 아두이노 커뮤니티

[www.element14.com/community/groups/arduino](http://www.element14.com/community/groups/arduino)

- 네이버 아두이노 스토리 카페

[cafe.naver.com/arduinostory](http://cafe.naver.com/arduinostory)

위에서 소개한 웹사이트를 모두 살펴봐도 문제를 해결할 수 없다면 저자인 제레미 블럼에게 트위터로 연락해도 좋다(@sciguy14). 어쩌면 저자가 답을 줄 수도 있을 테니까. 제레미 블럼의 웹사이트([www.jeremyblum.com](http://www.jeremyblum.com))의 contact 페이지로도 직접 연락할 수 있다. 하지만 빠른 응답을 보장할 수는 없다.

## 아두이노란?

아두이노의 가장 좋은 점은 원하는 어떤 것이든 만들 수 있다는 것이다. 아두이노로 자동 식물 재배 시스템을 비롯하여 웹 서버, 자동 비행 멀티콥터 제작도 가능하다.

아두이노는 마이크로컨트롤러 개발 플랫폼으로, 통합 개발 환경을 사용하여 직관적으로 프로그래밍 할 수 있다. 아두이노를 센서, 액추에이터, 전구, 스피커, 실드 등 다른 통합 회로들과 연결하여 제어 시스템을 디자인한다. 아두이노는 이 시스템의 두뇌에 해당하며, 프로그램으로 각각을 제어한다.

아두이노는 활용 가능성이 무한하기 때문에 책에서 아두이노의 모든 분야를 다 다루기란 불가능하다. 그러므로 여러분의 프로젝트를 수행하는 데 필요한 기술들을 배울 수 있도록, 아두이노를 어떻게 활용하는지 안내해 주는 역할에 충실하겠다.

이 책의 1장에서는 아두이노의 여러 다양한 보드를 다룬다. 만약 여러분이 아두이노의 내부를 더 자세히 알고 싶다면 아두이노의 웹사이트를 방문하길 권한다. 아두이노는 오픈 소스 하드웨어로 웹사이트에서 모든 내부 구조도와 문서들을 무료로 제공한다. 부록에서도 아두이노의 기술적 사양 일부를 다룬다.

## 오픈 소스 플랫폼

여러분이 오픈 소스의 세계에 처음 입문했다면 환영한다. 책에서 오픈 소스 하드웨어 운동을 자세하게 다룰 수는 없지만 한 번쯤 알아보는 것도 좋다. 오픈 소스 하드웨어의 전체 개요를 살펴보고 싶다면 오픈 소스 하드웨어 협회의 웹사이트 ([www.oshwa.org/definition](http://www.oshwa.org/definition))에 접속해 보자. 저자의 TEDx 발표([www.jeremyblum.com/portfolio/tedx-cornell-university-2011/](http://www.jeremyblum.com/portfolio/tedx-cornell-university-2011/))에서도 오픈 소스 운동에 관해 들을 수 있다. 이 동영상은 Wiley 출판사 웹사이트에서도 제공된다.

아두이노는 오픈 소스 하드웨어이기 때문에 모든 디자인 파일, 구조도, 소스 코드가 모두에게 무료로 공개된다. 사용자가 아두이노를 특별한 목적에 맞게 쉽게 개조할 수 있을 뿐만 아니라, 기존 디자인에 아두이노 플랫폼을 통합할 수도 있다. 이는 아두이노 복제품 제작 및 판매가 가능하고 다른 프로젝트에도 아두이노 소프트웨어를 사용할 수 있다는 것을 뜻한다. 책에서는 아두이노 공식 하드웨어만 다루지만, 책의 실습 예제를 수행할 때 여러 가지 부가 기능을 갖춘 수백 종류의 아두이노 기반 호환 보드를 사용할 수도 있다.

아두이노의 오픈 소스 라이선스(저작권)는 아두이노 마크를 사용하지 않는 한 아두이노 디자인의 상업적 재활용도 허용한다. 여러분이 기존 프로젝트의 프로토타이핑에 아두이노를 사용하고, 그 프로젝트를 상업화하더라도 아무런 제약이 없다. 한 가지 예로, 저자인 제레미 블럼이 메인보드를 디자인한 3D 프린터인 MakerBot Replicator는 아두이노 Mega 플랫폼을 기반으로 한 전자회로를 사용한다([www.thingiverse.com/thing:16058](http://www.thingiverse.com/thing:16058)).

하지만 여러분은 책에 제시된 소스 코드나 하드웨어의 라이선스를 준수해야 함을 기억하자. 일부 라이선스는 이전 저작권자의 결과물에 새로운 디자인을 더해 발표할 때 원저작자의 저작물임을 밝힐 것을 요청하기도 한다. 더불어 여러분이 만든 완성품이나 코드도 동일한 라이선스로 공유할 것을 요구한다. 이런 공유 문화는 아두이노를 사용하면서 반드시 참조하게 될 온라인 문서와 지원 체계를 발전시켜 준다. 책의 모든 코드도 GNU GPL 라이선스로 보호하면, 여러분이 원하는 목적에 맞게 사용할 수 있다.

## Beyond this book

어쩌면 여러분은 제레미 블럼이 유튜브에 올린 아두이노 튜토리얼을 이미 알고 있을지도 모르겠다([www.youtube.com/sciguy14](http://www.youtube.com/sciguy14)). 저자가 운영하는 이 책의 공식 웹사이트에서도 볼 수 있다. 각 장을 학습하면서 연관된 이 동영상들도 함께 확인해 보자. 더불어 여러분이 전자회로, 마이크로컨트롤러, 컴퓨터 과학과 창의성으로 엄청난 아이디어를 구현하는 것에 관심이 있다면 저자의 웹사이트 포트폴리오 탭에서 그의 창작물을 한 번 살펴보고길 바란다. 제레미 블럼의 모든 프로젝트는 아두이노처럼 오픈 소스 라이선스를 준수하기 때문에 필요하다면 언제든지 가져다 쓸 수 있다. 여러분도 책에서 얻은 기술로 무언가를 만들었다면 여러 사람과 공유하여 재미를 느낄 수 있을 것이다. 아두이노 세계를 탐험하는 모험을 성공적으로 마치길 바란다. Good Luck!

# Chapter 05

## 소리 출력

01 스피커의 작동 원리 이해

02 `tone()` 함수를 사용한 음악 출력

03 마이크로 피아노 키보드 제작

요약

한 걸음 더

인간에게는 오감이 있지만 미각이나 후각은 전자 기기와는 다소 거리가 먼 감각이다. 보통은 촉각, 시각, 청각을 활용하여 컴퓨터와 상호작용한다. 이전 장에서 버튼, 가변저항과 통신할 때는 촉각을 사용했고, LED와 통신할 때는 시각을 사용했다. 이 장에서는 소리를 만들어 내는 아두이노의 기능을 활용하여 시스템이 어떤 상황에 놓여 있는지 피드백을 쉽게 얻는 방법을 배울 것이다.

아두이노를 사용하여 소리를 만드는 방법은 여러 가지가 있다. 그중 가장 간단한 방법은 책에서 중점적으로 다루는 `tone()` 함수를 사용하는 것이다. 나아가 음악을 매끄럽게 재생하는 것처럼 조금 더 복잡한 기능을 추가하려면 MP3 실드 등 외부 프로세싱을 아두이노와 연결해야 한다. 실드는 아두이노에 특정한 기능을 추가할 목적으로 아두이노 위에 부착하는 애드온 보드이다.

이 장에서는 실드를 다루지 않지만, 책의 후반부에서는 몇 가지 실드를 사용할 것이다. 예를 들어, 아두이노 Due에는 소리를 만들 수 있는 디지털-아날로그 변환기(DAC)가 기본으로 내장되어 있다.

## 필요한 부품

- ▶ 아두이노 Uno
- ▶ USB 케이블
- ▶ 푸시 버튼 다섯 개
- ▶ 접퍼선
- ▶ 브레드보드
- ▶ 10k $\Omega$  가변저항
- ▶ 저항
- ▶ 8 $\Omega$  스피커

- 150 $\Omega$  저항 한 개



- 10k $\Omega$  저항 다섯 개



아두이노로 소리를 출력하기 전에 먼저 우리가 소리를 인지하는 과정과 원리부터 이해하자. 음파의 생성 원리와 특징을 살펴보고, 소리와 음악을 출력하려면 그 특성을 어떻게 제어해야 하는지 알아보자.

## 1. 음파의 속성

소리는 공기로 전달되는 압력파(Pressure Wave)이다. 스피커, 드럼, 종 등 물체가 진동하면 그에 따라 주위의 공기도 함께 진동한다. 공기 분자는 진동하면서 주변 분자에 에너지를 전달하고, 주변의 공기 분자도 함께 진동한다. 압력파는 이 진동 입자의 연쇄반응으로 소리의 근원에서 인간의 고막으로 전달된다.

이 소리의 전달 원리를 이해하면 아두이노로 진동 입자의 두 가지 특성인 주파수(Frequency)와 진폭(Amplitude)을 제어할 수 있다. 주파수는 공기 입자가 얼마나 빨리 앞뒤로 진동하는지 나타낸다. 주파수가 높아질수록 소프라노처럼 높은 음을 뜻하고, 반대로 주파수가 낮아질수록 베이스처럼 낮은 음을 뜻한다. 진폭은 진동의 크기를 뜻하며, 진폭이 클수록 소리가 크고 진폭이 작을수록 소리가 작다. [그림 5-1]은 진폭과 주파수가 다양한 음파를 보여 주는 사인파(Sine Wave)의 곡선 그래프들이다.

[그림 5-1]은 낮은 도, 중간 도(가온 다), 높은 도 세 가지 피아노 음의 파형이다. 각 그래프는 주어진 주파수에서 낮은 진폭과 높은 진폭의 파형을 동시에 보여 준다. 중간 도 음의 파형을 중심으로 주파수와 진폭을 이해해 보자. 중간 도 음의 주파수는 261.63Hz(헤르츠)이다. 다시 말해, 스피커, 기타 줄, 피아노는 1초 동안 정확히 261.63번 진동하면서 중간 도 음을 만든다.

이 값의 역수를 사용하여 [그림 5-1]에서 보이는 파형의 주기를 찾을 수 있다. 중간 도 음의 주기인  $1/261.63$ 은 3.822ms로, 주파수가 완전히 한 번 진동할 때까지 걸리는 시간을 뜻한다. 아두이노를 사용하여 음을 만들 때도 네모파의 주기를 설정하고, 이것으로 음정을 조절할 수 있다.

그림 5-1 다양한 주파수와 진폭의 음파

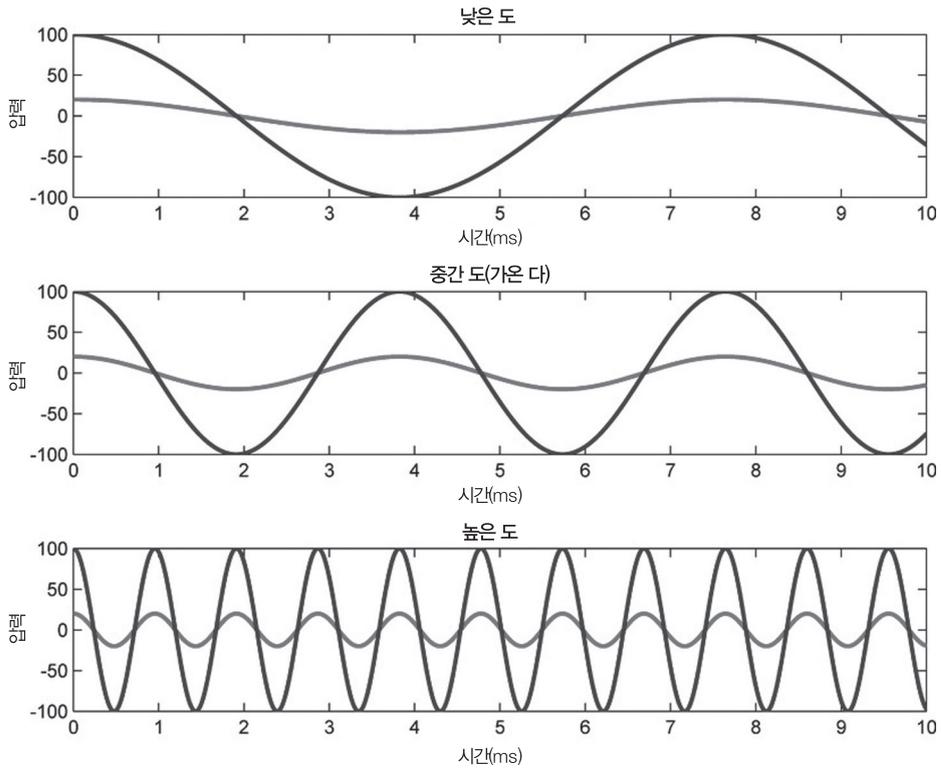
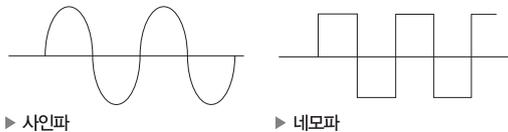


Image created with MATLAB

진폭은 스피커로 흐르는 전류의 양으로 조절할 수 있다. 다시 말해, 스피커와 가변저항을 함께 사용하면 스피커의 진폭(소리의 크기)을 조절할 수 있다.

디지털-아날로그 변환기를 내장한 아두이노 Due를 제외하고 아두이노는 실제 소리와 비슷한 사인파를 만들지 못한다. 네모파는 디지털 주기 파형으로, 사인 파형처럼 완만하지 않고 HIGH값과 LOW값 사이를 즉시 바꾸는 형태이다. 소리를 만드는 압력파를 생성할 수는 있으나 아날로그 형태의 사인파처럼 소리가 예쁘지는 않다.

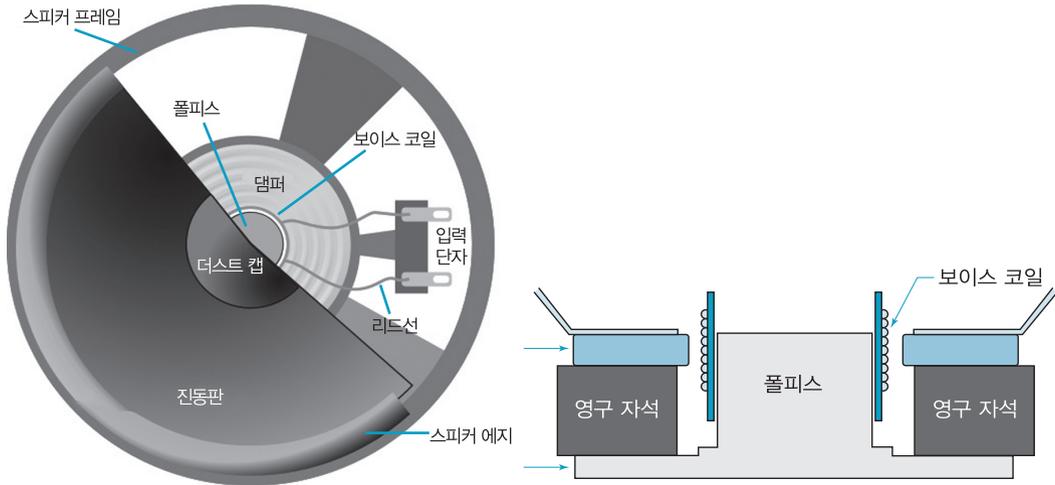
**TIP** 사인파와 네모파의 파형을 비교하면 다음과 같다. MP3 등 음악이나 영화(파일 형태)도 고해상도의 네모파로 만든 사인파이다. 아두이노 Due는 DAC를 내장하고 있기 때문에 고해상도의 사인파를 만들 수 있다.



## 2. 스피커가 소리를 출력하는 방법

앞 장에서 배운 모터와 마찬가지로 스피커도 전기를 운동으로 바꾸는 데 전자기력을 사용한다. 스피커의 뒷부분에 금속 조각을 가져다 대면 아마도 금속 조각이 스피커에 달라붙을 것이다. 모든 스피커의 뒷면에는 영구 자석이 부착되어 있기 때문이다. [그림 5-2]는 일반적인 스피커의 내부 구조를 나타낸 것으로, 영구 자석은 보이스 코일과 폴피스를 감싸고 있다.

그림 5-2 스피커의 내부 구조



© GNU Free Documentation License

사인파 전압 신호(아두이노는 네모파)가 보이스 코일로 보내지면 전류의 변화가 시작되어 자기장을 유도한다. 이 자기장의 영향으로 영구 자석이 보이스 코일을 끌어당겼다 밀어냈다 하면서 폴피스와 보이스 코일에 연결된 진동판을 앞뒤로 떨리게 한다. 앞뒤로 움직이는 이 떨림이 차례로 스피커 앞의 공기를 진동시켜 음파가 생성되어 우리 귀까지 전달되는 것이다.

아두이노 IDE는 내장 함수를 제공하여 임의의 주파수를 지닌 소리를 쉽게 만들 수 있게 한다. `tone()` 함수는 특정 주파수의 네모파를 지정된 출력 핀으로 출력할 수 있게 도와주며, 인수는 총 세 개이다(마지막 인수는 선택 사항).

- 첫 번째 인수는 소리를 출력할 핀 번호를 지정한다.
- 두 번째 인수는 출력할 소리의 주파수를 지정한다.
- 세 번째 인수는 출력할 소리의 지속 시간을 지정한다. 인수를 생략해도 실행에는 문제가 없지만, `noTone()` 함수를 호출하기 전까지는 소리가 계속 날 것이다.

`tone()` 함수는 아두이노의 하드웨어 타이머 세 개 중 하나를 사용하기 때문에 소리를 뒷단(백그라운드)에서 출력하는 동안에는 아두이노가 해당 타이머를 사용하여 다른 작업을 할 수 없다.

아두이노에서 소리를 출력하는 방법을 배우고 나면, 다양한 입력(버튼, 거리 센서, 가속도계 등)에 반응하는 소리를 재생할 수 있을 것이다. 이 장 마지막에서는 버튼 다섯 개로 작동하는 아주 간단한 피아노 키보드를 만들어 볼 것이다.

## 1. 게이름 설정 파일 추가

음악을 재생하는 프로그램을 만들 때 각 주파수와 게이름을 미리 매핑해 놓은 설정 파일을 사용하면 편리하다. 음을 261.63Hz처럼 주파수로 표현하는 것보다 도, 레, 미 등 문자로 표현하는 것이 더 일반적이고, 훨씬 직관적으로 음악을 재생할 수 있기 때문이다. 아두이노 IDE는 각 주파수에 따른 음계를 설정해 놓은 헤더 파일인 `pitches.h`를 제공한다. 이 파일은 아두이노 설치 폴더(`..\Arduino\examples\02.Digital\toneKeyboard`)에서 찾거나 책 공식 웹사이트인 [www.exploringarduino.com/content/ch5](http://www.exploringarduino.com/content/ch5)에서 다운로드할 수 있다. 이 헤더 파일을 실습 폴더에 넣어 두면 실습 프로그램에서 쉽게 사용할 수 있다.

**TIP** 아두이노에서 생성한 실습은 라이브러리\문서\Arduino\에 생성된다.

아두이노 IDE는 파일 이름과 똑같은 폴더를 생성한 후 그 안에 .ino 형식으로 스케치 파일을 저장한다. 필요한 파일을 생성된 폴더에 같이 넣어 두면 프로그램 내에서 파일을 불러들일 수 있으므로 코드가 깔끔해진다. 아두이노 IDE를 실행하여 빈 스케치 파일을 'music'으로 저장해 보자. 그리고 나서 새로 생성된 music 폴더에 pitches.h 파일을 복사하고, 아두이노 IDE를 종료했다 다시 music.ino 파일을 열면 [그림 5-3]처럼 탭이 두 개 나타난다.

**그림 5-3** 헤더 파일과 함께 나타난 아두이노 IDE



[pitches.h] 탭을 클릭하여 파일 내용을 살펴보자. pitches.h 파일은 각 주파수를 사람이 인식할 수 있는 게이름으로 매핑한 선언문이다. 하지만 .ino 파일처럼 폴더에 저장하는 것만으로는 헤더 파일을 사용할 수 없다. music에 다음 코드를 입력해야 컴파일 과정에서 헤더 파일이 프로그램에 포함되어 pitches.h를 사용할 수 있다.

```
#include "pitches.h" //게이름이 정의된 헤더 파일 삽입
```

컴파일러 입장에서는 헤더 파일의 내용을 복사하여 .ino 파일에 붙여 넣는 것과 #include 명령으로 헤더 파일을 불러오는 것이 기본적으로 똑같다. 하지만 #include를 사용했을 때의 코드가 훨씬 간결하고 가독성도 높아진다. 이제 음악을 재생하는 프로그램을 작성하면서 게이름이 정의된 pitches.h 파일을 사용해 보자.

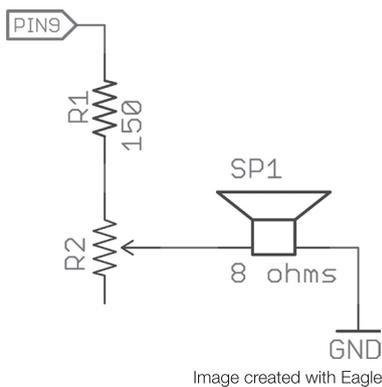
## 2. 스피커 연결

게이름이 정의된 헤더 파일을 추가했으므로 이제 테스트 회로를 만들고 간단한 음악을 재생할 수 있는 프로그램 준비가 끝났다. 테스트 회로는 아두이노의 디지털 입출력 핀에 스피커만 연결하면 되기에 매우 간단하다. 그러나 이전 장에서 배운 전류를 제한하는 저항에 유의해야 한다.

아두이노 입출력 핀이 너무 많은 전류를 소모하면 안 되므로 LED와 마찬가지로 스피커에도 전류를 제한하는 저항을 연결해야 한다. 앞에서 아두이노의 입출력 핀은 각각 최대 40mA 전류만 사용할 수 있다고 했다. 그러므로 스피커가 그 이상을 사용하지 못하도록 저항을 연결해야 한다. 시중에 판매되는 대부분의 스피커와 마찬가지로 실습에서 사용하는 스피커의 내부 저항은 8Ω이다. 이 저항은 전자석을 구성하는 보이스 코일에서 나온다.

앞에서 배운 옴의 법칙( $V=IR$ )을 상기해 보자. 아두이노의 디지털 입출력 핀에서는 5V 전원이 나오고, 사용할 수 있는 최대 전류는 40mA이다. 옴의 법칙으로 스피커에 붙일 저항값을 계산해 보면  $R=5V/40mA=125\Omega$ 이 된다. 스피커의 내부저항이 8Ω이므로, 스피커에 연결할 저항은 최소한  $125\Omega-8\Omega=117\Omega$ 은 되어야 한다. 시중에서 구할 수 있는 저항 중 이 값에 가장 가까운 것은 150Ω이다. 또 저항값을 바꿔 스피커로 출력하는 소리 크기를 조절할 수 있으므로 [그림 5-4]처럼 가변저항도 함께 사용한다. 다음 회로에서 R1은 150Ω이고, R2는 가변저항이다.

그림 5-4 음량 조절 장치를 추가한 스피커 회로도



이전 장에서 가변저항을 사용했던 방법과 달리 이 회로에서는 가변저항의 핀 세 개 중 두 개만 사용한다. 가운데 핀은 스피커와 연결하고, 양 끝에 위치한 핀 중 하나에는 150Ω 저항을 연결한다. 가변저항의 손잡이를 최대 저항값 쪽으로 끝까지 돌리면, 기존 회로에 저항값이 추

가되어 총 저항값은 '150Ω+가변저항'이 될 것이고, 소리는 작게 들릴 것이다. 반면, 가변저항을 반대 방향으로 끝까지 돌리면 가변저항의 값은 0Ω이 되어 회로에 추가적인 저항을 가하지 않는다. 이때 소리는 가장 크게 들린다. [그림 5-4]의 회로도를 참고하여 직접 회로를 연결해 보자. 스피커는 극성이 따로 없으므로 어떤 방향으로 연결해도 된다. 그리고 나서 [그림 5-5]를 참고하여 회로를 제대로 연결했는지 확인해 보자.

그림 5-5 스피커 회로도

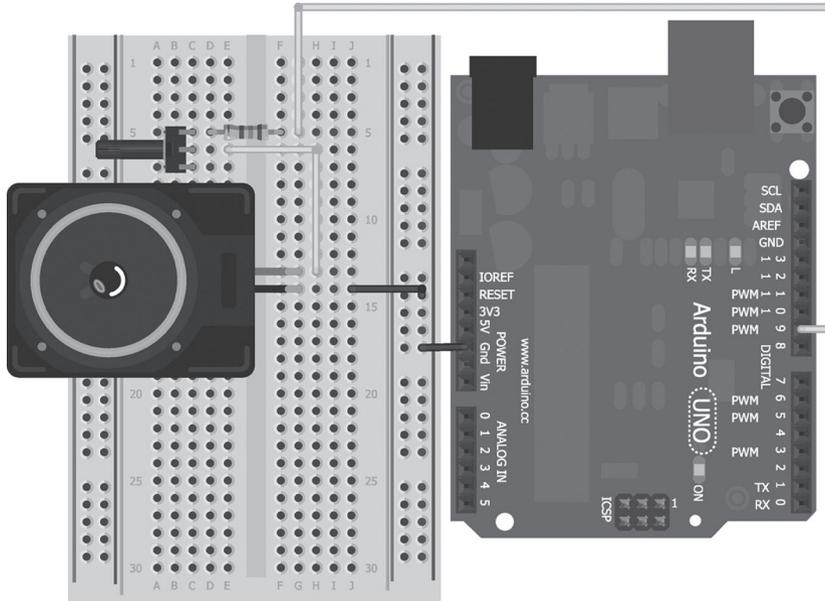


Image created with Fritzing

회로를 완성했으면 이제 음악을 재생하는 프로그램을 작성해 보자.

### 3. 배열을 사용한 음악 재생 프로그램 작성

아두이노의 tone() 함수를 사용하여 음악을 재생하려면, 여러 개의 값을 동시에 저장할 수 있는 배열 사용 방법을 알아야 한다. 배열을 사용하면 배열에 저장한 음들을 꺼내서 스피커로 출력하는 반복문을 활용한 간단한 프로그램을 완성할 수 있다.

## 배열 사용 방법

여기에서는 재생하려는 음의 목록을 저장하고 프로그램에서 이를 사용하려고 배열을 이용한다. 배열은 형식이 똑같은 연속된 값으로, 숫자를 매긴 목록이라고 할 수 있다. 이 값들을 함께 그룹으로 묶으면 반복 실행하기에 매우 이상적인 형태가 된다. 배열 내의 각 저장 공간은 목록에서 위치를 나타내는 인덱스로 구분되며, 각 인덱스에는 사용자가 입력한 값이 저장된다.

아두이노의 메모리를 적절하게 사용하려고 프로그램 내에서 배열을 선언할 때는 배열의 길이도 지정해야 한다. 배열을 선언할 때는 배열의 길이를 숫자로 표시하거나 입력할 값을 모두 넣어 선언할 수 있다. 예를 들어, 다음처럼 코드를 작성하여 정수값 네 개를 저장할 배열을 생성할 수 있다.

```
int numbers[4];
```

배열을 선언할 때 저장할 값을 모두 입력하면 [] 안에 배열의 길이를 지정하지 않아도 된다. 배열의 길이를 지정하지 않으면 입력되는 값의 개수와 동일한 길이로 배열이 선언된다.

```
//다음 두 가지 배열 선언 방법은 모두 유효
int numbers[4]={-7, 0, 6, 234};
int numbers[]={-7, 0, 6, 234};
```

배열은 0 기반 넘버링(Zero Indexed)을 사용한다. 배열의 첫 번째 자리 인덱스값은 0이고, 두 번째 자리 인덱스값은 1이다. 배열 이름 뒤의 대괄호([])에 인덱스값을 입력하여 배열 내 특정 위치에 저장된 값에 접근할 수 있다. 예를 들어, 9번 핀에 연결된 LED 밝기를 지정하는 값으로 numbers[] 배열의 세 번째 자리에 저장된 값을 사용하고 싶다면 다음처럼 코드를 작성한다.

```
analogWrite(9, numbers[2]);
```

numbers[] 배열의 인덱스값은 0부터 시작하므로, numbers[2]에서 2는 numbers[] 배열의 세 번째 자리에 저장된 값을 뜻한다. 이 세 번째 자리에 다른 값을 저장하고 싶다면 다음처럼 입력한다.

```
numbers[2]=10;
```

자, 이제 재생하고 싶은 음악의 계이름을 순서대로 저장하는 배열을 선언해 보자.

## 음계와 재생 지속 시간을 저장하는 배열 작성

연주하려는 음악 정보를 저장하려고 크기가 똑같은 배열 두 개를 사용할 것이다. 첫 번째 배열에는 출력할 음을 저장하고, 두 번째 배열에는 각 음을 몇 ms 동안 재생할 것인지 지속 시간을 저장한다. 이 배열들의 인덱스를 차례로 반복 실행하여 음악을 재생할 것이다.

이 배열들을 사용하여 다음처럼 짧고 간단한 노래를 만들어 보자.

```
//연주할 음이 저장된 배열 notes 선언
int notes[]={
    NOTE_A4, NOTE_E3, NOTE_A4, 0,
    NOTE_A4, NOTE_E3, NOTE_A4, 0,
    NOTE_E4, NOTE_D4, NOTE_C4, NOTE_B4, NOTE_A4, NOTE_B4, NOTE_C4, NOTE_D4,
    NOTE_E4, NOTE_E3, NOTE_A4, 0
};

//음의 지속 시간이 저장된 배열 times 선언. 단위는 ms
int times[]={
    250, 250, 250, 250,
    250, 250, 250, 250,
    125, 125, 125, 125, 125, 125, 125, 125,
    250, 250, 250, 250
};
```



이 코드에서 두 배열은 각 값이 20개로, 길이가 똑같다. 이 중 몇 개의 음은 0으로 지정되어 있는데, 0은 소리를 재생하지 않는 쉼표 구간이다. 첫 번째 배열에 저장된 각 음은 두 번째 배열의 재생 지속 시간과 짝지어져 있다. 250ms는 사분음표, 125ms는 팔분음표에 해당하며, 이 음악은 4분의 4박자 곡이다.

다음 단계에서 이 음악을 실행하여 자신만의 음악을 만들어 보자.

## 음악 재생 프로그램 완성

마지막 단계는 이 배열들을 음악으로 재생하는 것이다. 두 배열의 각 인덱스를 차례로 호출하는 for문을 사용하여 주어진 시간 동안 음을 연주하는 프로그램을 완성한다. 반복 재생을 피하려면 재생 기능을 setup() 함수에 넣어 단 한 번만 연주되도록 처리해야 한다. 다시 들으려면 간단하게 아두이노의 리셋 버튼을 눌러 시스템을 재시작하면 된다. [실습 5-1]은 지금까지 다룬 음악 재생 프로그램의 전체 코드이다.

```

//스피커로 음악 재생하기

#include "pitches.h" //게이름이 정의된 헤더 파일 삽입

const int SPEAKER=9; //9번 핀을 사용하는 SPEAKER 상수 정의

//연주할 음이 저장된 배열 notes 선언
int notes[]={
    NOTE_A4, NOTE_E3, NOTE_A4, 0,
    NOTE_A4, NOTE_E3, NOTE_A4, 0,
    NOTE_E4, NOTE_D4, NOTE_C4, NOTE_B4, NOTE_A4, NOTE_B4, NOTE_C4, NOTE_D4,
    NOTE_E4, NOTE_E3, NOTE_A4, 0
};

//음의 지속 시간이 저장된 배열 times 선언. 단위는 ms
int times[]={
    250, 250, 250, 250,
    250, 250, 250, 250,
    125, 125, 125, 125, 125, 125, 125, 125,
    250, 250, 250, 250
};

void setup()
{
    //게이름, 지속 시간 배열 순서대로 음악 재생
    for(int i=0; i<20; i++)
    {
        tone(SPEAKER, notes[i], times[i]);
        delay(times[i]);
    }
}

void loop()
{
    //음악을 다시 들으려면 리셋 버튼을 누름
}

```

다른 음악을 재생하려면 다음 사항에 유의하여 프로그램을 수정해야 한다.

- 1 연주할 음과 그 음의 지속 시간이 저장된 두 배열의 길이가 똑같아야 한다.
- 2 for문에서 변수 *i*의 최댓값을 배열의 길이와 똑같이 바꿔야 한다.
- 3 `tone()` 함수는 백그라운드로 실행될 수 있기 때문에 `tone()` 함수와 함께 `delay()` 함수를 사용하여 음의 재생 시간만큼 프로그램의 실행을 정지시켜야 한다. 그래야 앞의 음이 끝나기 전에 다음 음을 재생하지 않을 것이다.

## 4. `tone()` 함수의 한계

`tone()` 함수에는 주의해야 할 몇 가지 한계점이 있다. `servo` 라이브러리와 마찬가지로 `tone()` 함수도 아두이노 보드의 하드웨어 타이머를 사용한다. 이 때문에 `tone()` 함수를 사용하면 Mega를 제외한 아두이노 보드의 3번과 11번 핀에서는 PWM이 동작하지 않는다.

또 아두이노의 디지털 입출력 핀들은 디지털-아날로그 컨버터(DAC)가 아니므로 사인파 대신 지정된 주파수의 네모파만을 출력한다. 이것은 스피커에서 음을 만들기에는 충분하나, 음을 재생하기에는 적합하지 않다. 따라서 아두이노로 웨이브 파일을 재생하려면 음악 재생 실드(예 adafruit의 Wave Shield나 SparkFun의 MP3 shield)를 사용한다. 그리고 DAC 컨버터 추가나 DAC가 내장된 아두이노 Due와 Due에서만 사용할 수 있는 오디오 라이브러리를 활용하는 방법도 있다.

`tone()` 함수의 마지막 한계는 한 번에 하나의 핀에서만 사용할 수 있다는 것이다. 이 때문에 `tone()` 함수를 사용하여 여러 개의 스피커를 구동하는 것은 적합하지 않다. 아두이노 보드에서 동시에 여러 개의 스피커를 사용하려면 매뉴얼 타이머 인터럽트 제어를 사용해야 한다. 이것과 관련된 내용은 12장에서 살펴볼 것이다.

---

**Note** 조금 복잡하기는 하지만 아두이노로 다중 스피커를 제어하는 방법을 알고 싶다면 제레미 블럼의 블로그에서 추가적인 정보를 살펴보자.

[www.jeremyblum.com/2010/09/05/driving-5-speakers-simultaneously-with-an-arduino/](http://www.jeremyblum.com/2010/09/05/driving-5-speakers-simultaneously-with-an-arduino/)



음들을 연속적으로 재생하는 것은 완성되어 있는 다른 프로그램에 오디오 피드백을 추가할 수 있다는 점에서 괜찮은 옵션이다. 예를 들어, 앞 장에서 작성한 회전하는 거리 센서 회로에서 확인용 LED를 소리로 바꾸거나 LED와 함께 사용하는 방법을 생각해 볼 수 있다. 하지만 소리를 동적으로 제어하고 싶다면 간단한 마이크로 피아노 키보드를 만들어 보는 것도 좋다.

이번 실습에서는 간단한 오음음계(Pentatonic) 피아노를 만든다. 보통 한 옥타브는 음(도, 레, 미, 파, 솔, 라, 시) 일곱 개로 구성되지만, 오음음계는 다섯 개로 구성된다. 음 사이의 불협화음이 거의 없어 함께 울렸을 때 좋은 소리를 낼 수 있다.

---

**Note** 제레미 블럼의 프로젝트 중에 SudoGlove가 있다. 이것은 오음음계를 사용하여 음악을 합성할 수 있는 (무선 자동차도 조종할 수 있는) 제어용 장갑으로 [www.sudoglove.com](http://www.sudoglove.com)에서 관련 동영상과 자료를 확인할 수 있다.

---

아두이노 피아노를 만드는 데 사용하는 오음음계는 도, 레, 미, 솔, 라이다. 어떤 옥타브의 음계를 사용해도 상관없으나, 여기에서는 헤더 파일에서 4옥타브의 음계 선언문을 가져다 4옥타브를 만들 것이다.

먼저, 버튼 다섯 개를 아두이노에 연결해 보자. 2장에서는 10kΩ으로 풀다운 저항을 만들어 버튼을 사용했는데, 이 실습에서는 버튼을 누를 때만 음을 재생하기 때문에 반드시 풀다운 저항을 연결할 필요가 없다. [그림 5-6]처럼 버튼을 연결하고, 이전 실습에서 사용한 스피커 연결 회로는 그대로 둔다.

그림 5-6 마이크로 피아노 연결 회로도

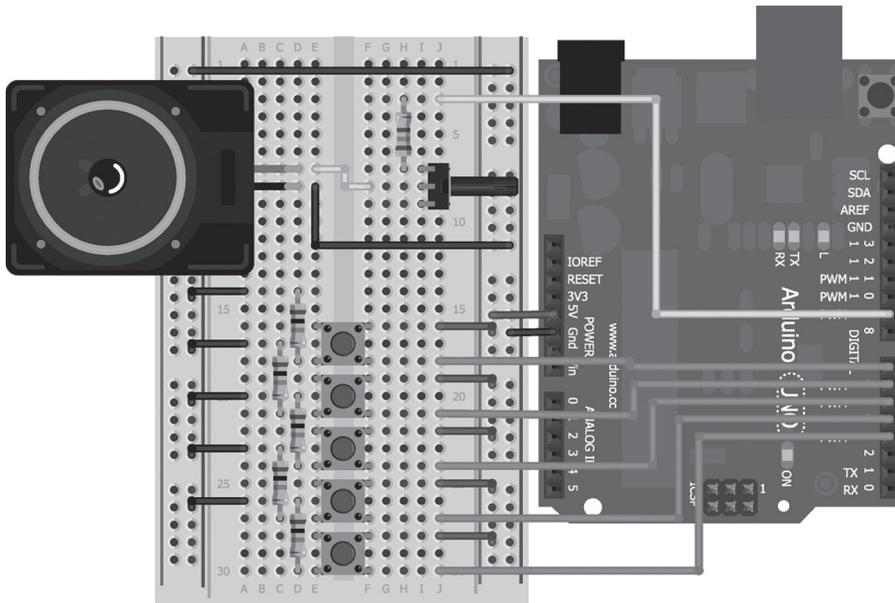


Image created with Fritzing

피아노 만들기 프로그램의 코드는 매우 단순하다. 메인 loop() 함수를 반복할 때마다 각 버튼을 눌렀는지 체크한다. 버튼을 눌렀다면 해당하는 음이 재생된다. 이 실습에서는 tone() 함수에 재생 지속 시간이 빠져 있는데, 버튼을 누르는 동안에는 해당 음을 계속 재생할 것이기 때문이다. 그 대신 버튼을 아예 누르지 않을 때는 소리의 재생을 멈추도록 메인 loop() 함수의 끝에서 noTone() 함수를 호출한다. 적은 수의 음만 사용하기 때문에 이전 실습처럼 pitches.h 헤더 파일을 가져오지 않고 필요한 음이 정의된 부분만 복사하여 프로그램에 직접 붙여 넣는다. [실습 5-2]를 참고하여 프로그램을 작성한 후 아두이노에 업로드해 보자. 버튼을 눌러 소리가 잘 들리는지 확인한다.

**실습 5-2** 펜타토닉 마이크로 피아노 만들기

piano.ino

```
//오음음계 피아노
//음계 : 도 레 미 솔 라

#define NOTE_C 262 //'도'음 정의
#define NOTE_D 294 //'레'음 정의
#define NOTE_E 330 //'미'음 정의
#define NOTE_G 392 //'솔'음 정의
```

```

#define NOTE_A 440 //'라'음 정의

const int SPEAKER=9; //9번 핀을 사용하는 SPEAKER 상수 정의
const int BUTTON_C=7; //7번 핀을 사용하는 도 버튼 BUTTON_C 상수 정의
const int BUTTON_D=6; //6번 핀을 사용하는 레 버튼 BUTTON_D 상수 정의
const int BUTTON_E=5; //5번 핀을 사용하는 미 버튼 BUTTON_E 상수 정의
const int BUTTON_G=4; //4번 핀을 사용하는 솔 버튼 BUTTON_G 상수 정의
const int BUTTON_A=3; //3번 핀을 사용하는 라 버튼 BUTTON_A 상수 정의

void setup()
{
    //모든 입출력 핀의 기본값은 입력이므로 setup() 함수에서 처리할 것이 없음
    //tone() 함수는 출력 제어
}

void loop()
{
    while(digitalRead(BUTTON_C))
        tone(SPEAKER, NOTE_C);
    while(digitalRead(BUTTON_D))
        tone(SPEAKER, NOTE_D);
    while(digitalRead(BUTTON_E))
        tone(SPEAKER, NOTE_E);
    while(digitalRead(BUTTON_G))
        tone(SPEAKER, NOTE_G);
    while(digitalRead(BUTTON_A))
        tone(SPEAKER, NOTE_A);

    //버튼을 누르지 않으면 음 재생을 멈춤
    noTone(SPEAKER);
}

```



while문은 버튼을 누르는 동안에는 해당하는 주파수로 tone() 함수를 끊임없이 호출한다. while문은 digitalRead() 함수로 버튼의 상태를 바로 읽어 들인다. 이것은 프로그램 코드가 단순하여 버튼의 누름 상태를 읽어 변수에 저장하는 방법을 사용할 필요가 없기 때문이다. digitalRead() 함수는 버튼이 HIGH값을 입력할 때마다 불리언의 값 'true'를 반환한다. 실행문이 단 한 줄뿐이므로, 코드를 간결하게 만들려고 while문에 {}를 사용하지 않았다. 하지만 while문의 실행문이 두 줄 이상이라면 이전 장과 마찬가지로 {}를 사용해야 한다.

## 요약

이 장에서 배운 것을 요약하면 다음과 같다.

- 스피커는 공기로 전달되는 압력파를 생성하고, 전달된 파형은 우리 귀를 통해 소리로 인식된다.
- 보이스 코일에서 전류 변화는 자기장을 유도하여 스피커에서 소리를 발생시킨다.
- `tone()` 함수는 임의의 주파수를 가진 소리를 일정 시간 동안 생성하는 데 사용한다.
- 아두이노 프로그래밍 언어는 일련의 데이터를 반복 실행할 수 있도록 데이터를 연속된 형태로 저장해 주는 배열을 지원한다.
- 스피커와 연결된 가변저항으로 음량을 조절할 수 있다.

## 한 걸음 더

앞에서 수행한 실습과 이 장에서 배운 내용을 바탕으로 다음 프로젝트를 스스로 진행해 보자.

- 거리 센서로 출입을 감지하여 알람을 울리는 출입문 감지 시스템
- 빛이나 온도 등 주변 환경 조건에 따라 달라지는 음악 재생 시스템
- 조명이나 소리로 생각이나 분위기를 전달하는 시청각 예술 설치물  
(☞ 자전거 감빱이 가방 [goo.gl/HVMYz8](http://goo.gl/HVMYz8))