

HanbiteBook

Realtime 32

TECH DIY

아두이노와 환경모니터링

직접 만든 아두이노 기기로 환경 지키기

Environmental Monitoring with Arduino

에밀리 거츠, 패트릭 디 저스토 지음 / 이영수 옮김

O'REILLY®  한빛미디어
Hanbit Media, Inc.

Project Book

WATCHING OUR
WORLD WITH
SENSORS

Environmental Monitoring with Arduino

Building Simple Devices to Collect
Data About the World Around Us

Emily Gertz & Patrick Di Justo



O'REILLY®

Maker
PRESS

이 도서는 O'REILLY의
Environmental Monitoring with Arduino의
번역서입니다.

Tech DIY

아두이노와 환경모니터링

직접 만든 아두이노 기기로
환경 지키기

Tech DIY **아두이노와 환경모니터링** 직접 만든 아두이노 기기로 환경 지키기

초판발행 2013년 6월 26일

지은이 에밀리 거츠, 패트릭 디 저스토 / **옮긴이** 이영수 / **펴낸이** 김태현
펴낸곳 한빛미디어(주) / **주소** 서울시 마포구 양화로 7길 83 한빛미디어(주) IT출판부
전화 02-325-5544 / **팩스** 02-336-7124
등록 1999년 6월 24일 제10-1779호
ISBN 978-89-6848-638-8 15000 / **정가** 9,900원

책임편집 배용석 / **기획** 김창수 / **편집** 안선화
디자인 표지 여동일, 내지 스튜디오 [임], 조판 북누리
마케팅 박상용, 박주훈

이 책에 대한 의견이나 오타자 및 잘못된 내용에 대한 수정 정보는 한빛미디어(주)의 홈페이지나 아래 이메일로 알려주세요.

한빛미디어 홈페이지 www.hanbit.co.kr / **이메일** ask@hanbit.co.kr

Published by HANBIT Media, Inc. Printed in Korea Copyright © 2013 HANBIT Media, Inc.

Authorized Korean translation of the English edition of *Environmental Monitoring with Arduino*,

ISBN 9781449310561 © 2012 Emily Gertz and Patrick Di Justo. This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

이 책의 저작권은 오라일리사와 한빛미디어(주)에 있습니다.

저작권법에 의해 보호를 받는 저작물이므로 무단 복제 및 무단 전재를 금합니다.

지금 하지 않으면 할 수 없는 일이 있습니다.

책으로 펴내고 싶은 아이디어나 원고를 메일(ebookwriter@hanbit.co.kr)로 보내주세요.

한빛미디어(주)는 여러분의 소중한 경험과 지식을 기다리고 있습니다.

저자 소개

지은이_ **에밀리 거츠** Emily Gertz

1994년부터 웹 프로듀서이자 커뮤니티 운영자, 콘텐츠 전략가로 인터넷에서 활동해 왔다. 환경 전문 계간지 《온어스 OnEarth》의 통신원으로, 월드체인지(Worldchanging.com)의 엔지니어이자 예술가인 나탈리 제레미엔코(Natalie Jeremijenko)를 인터뷰한 2004년 이후 《온어스》의 DIY 환경모니터링 분야를 담당하고 있다. 2011년 4월에는 일본 시민들이 자발적으로 방사선 모니터링에 나선 것과 관련한 기사를 《온어스》에 기고하기도 했다. 《그리스트 Grist》, 《드웰 Dwell》, 《사이언티픽 아메리칸 Scientific American》, 《파퓰러 메카닉스 Popular Mechanics》 등 많은 잡지를 통해 기고가로 활동 중이다.

지은이_ **패트릭 디 저스토** Patrick Di Justo

C, C++, Java, 프로그래밍의 전문가로, 미국 연방준비제도(Federal Reserve)를 위해 로봇 프로그래머로 일해 왔다. 아두이노는 2007년에 처음 구입했다. 미국의 IT 월간지 《와이어드 매거진 Wired Magazine》의 편집위원으로 매달 《와이어드 매거진》에 칼럼 'What's Inside'를 연재하는 것은 물론, 《드웰 Dwell》, 《사이언티픽 아메리칸》, 《파퓰러 사이언스 Popular Science》 《뉴욕타임즈 The New York Times》 등 많은 미디어에 기고하는 기고가로도 활동 중이다. 지은 책으로 『The Science of Battlestar Galactica』(Wiley, 2010)가 있다.

역자 소개

역자_ 이영수

초등학교 시절부터 부수고 만드는 게 취미였던 것이 대학으로까지 이어져, 대학에서 전자공학을 전공했다. IBM 국제구매사무소장, 삼성전자 디스플레이 개발팀장, STMicroelectronics 한국지사장 등 갑과 을을 넘나들며 직장 생활을 해왔다. 초등학교 5학년 때 처음으로 ‘트랜지스터’ 라디오를 만들었으며, 대학 실습시간엔 구시대의 ‘진공관’ 라디오를 만드는 아이러니를 겪기도 했다. 지금도 취미가 땀질인, 철이 덜 든 ‘애어른’이다.

저자 서문

이 책은 ‘보이지 않는 것을 보이게 하는(Making the Invisible Visible)’ 방법에 관한 책이다.

책에서 소개하는 각 프로젝트에서는 특정 환경조건⁰¹을 제시하고, 이를 모니터링할 수 있는 작고 값싼 전자기기의 제작법을 단계별로 알려준다. 나아가 이 전자기기가 모니터링한 결과치의 의미 또한 설명해줄 것이다.

환경모니터링⁰²을 시작하면 여러분은 새로운 방식으로 주변 세상을 이해하게 될 것이다. 가령, 수질검사기를 통해서 보면, 맑고 깨끗하게 흐르는 시냇물을 그냥 맑고 깨끗한 물이 아니라 엄청나게 많은 미립자를 포함하고 있는 물로 보게 된다(6장 참조).

온도와 습도를 측정하는 장치를 이용하면, 하루 중 가장 더운 때가 정오가 아니라 실제로는 오후 3시경이라는 것을 알게 될 것이다(8장 참조).

전자기파 측정기를 만들어 살펴보면, 아무리 조용한 실내라도 보이지 않고 들리지 않는 전기적 진동이 웅웅거리고 있다는 것을 발견할 것이다(4장 참조).

우리는 환경모니터링을 보통 정부기관이나 대학, 기업의 전문가에게 맡긴다. 전문 교육을 받았음은 물론, 다루기 힘들고 고가인 장비도 가진 이들은 종종 소속 기관의 아젠다^{agenda}로 환경 문제에 접근한다. 그 자체로도 복잡한 자연환경은 인간과 인간의 활동이 만들어낸 것들로 인해 더욱 복잡해진다. 때문에 환경에 관한 전문 지식은 우리의 생명과 지역사회에 매우 중요한 역할을 하고, 과학적 분석과 전문

01 **역자주** 환경 전체의 상황이나 그것을 구성하는 인자 각각의 양적·질적 상태를 말하는데, 여기서는 기상(날씨), 기후, 지형, 소리, 진동 등을 가리킨다.

02 **역자주** 환경의 상태와 그 변화를 기록하기 위해 과학적으로 계획된 연속적인 측정과 관측을 말한다.

지식은 ‘인간의 활동이 환경과 우리의 건강에 미치는 영향’을 관리할 실질적 규제를 만드는 데 핵심적인 역할을 한다.

우리가 직접 환경모니터링을 할 수 있게 되면, 우선 전문가들만 환경모니터링을 할 수 있다는 편견이 없어진다. 그에 따라 환경문제를 이해하기 시작하면 지식이 쌓이고, 지식이 쌓이면 우리의 삶을 보다 좋아지도록 만드는 힘(전기를 아끼는 일부터 오염을 유발하는 기업에 책임을 묻는 일, 기후변화를 연구하는 과학자를 돕는 일) 또한 가질 수 있다.

저자
에밀리 거츠

역자 서문

대학에 다닐 때 포트란으로 프로그램 몇 줄 짜본 것이 내가 해본 프로그램 경험의 전부였고, 직장에서는 주로 아날로그 회로만 만졌다. 몇 년 전, 뜻한 바 있어 마이크로컨트롤러MCU에 때늦게 입문해 C언어로 프로그램을 작성할 수 있는 단계에 이르렀지만, 여전히 어려운 부분이 많다. 각 포트의 초기화, 타이머 설정, 각종 인터럽트의 적용 등 실제 프로그램뿐만 아니라 초기화 작업에 있어서도 머리를 쥐어짤 때가 한두 번이 아니다.

MCU용 프로그램을 작성하려면 최소한 몇 개월간 전문교육을 받고 실무 경험 또한 몇 년간은 해야 하는 것이 필수다. 하지만 우리 같은 일반인은 이런 경험을 쌓을 기회가 거의 없다고 해도 과언이 아니다.

그런데! 1년 전, 우연히 아두이노를 만나면서 눈앞에 신천지가 펼쳐졌다! MCU용 C언어와는 비교도 되지 않을 만큼 프로그램을 작성하기가 쉬웠다. AD 변환^{analog to digital conversion}을 하여 전압과 온도를 측정하고, LCD 디스플레이를 붙여 그 값을 표시하고, 시리얼 통신을 이용해 데이터를 PC로 보내 저장하고, 릴레이로 각종 외부장치를 제어하고, I2C^{Inter-IC} 버스로 다른 디바이스와 연결하며, 심지어 인터넷에 연결하여 전 세계의 다른 사람들과 데이터를 공유하는 일 등이 정말로 누워서 떡 먹기였다. 심지어 일반 소형 MCU로는 계산하기 어려운 부동소수점 계산까지 가능했다! 이런 획기적인 오픈 플랫폼을 개발하고, 모든 사람에게 무료로 공개한 아두이노 개발자에게 축복 있으라!

아두이노를 이용하면 마이크로컨트롤러의 무궁무진하고 강력한 기능으로 아두이노를 알기 전이라면 꿈도 꾸지 못했을 온갖 독창적인 창조물을 직접 만들 수 있다. 초등학생부터 할아버지까지, 남녀를 불문하고, 전공자이든 비전공자든 관계없이 말이다! 많은 아티스트들이 아두이노를 이용한 모바일 예술작품을 만들고 있는 요즘의 추세는, 비

전공자들이 아두이노를 얼마나 쉽게 이용하고 있는지를 보여주는 좋은 예다. 수많은 예제 프로그램과 라이브러리가 맨땅에 부딪히며 겪어야 할 많은 수고를 덜어준다. 오픈된 각종 인터넷 사이트나 동호회가 여러분을 외로운 늑대로 두지 않을 것이다. 오직 여러분이 가진 꿈의 한계만이 여러분의 한계가 될 것이다.

영어에 눈을 뜨게 되는 초등학교 고학년부터 아두이노를 만나게 하여, 한창 자라나는 아이들이 창의력을 꽃피울 수 있는 기초를 만들어주었으면 하는 바람에서 번역에 참여했다. 스마트폰으로 조종하는 무선 쿼드콥터Quardcopter나 로봇을 만드는 중학생이 나타나는 시대를 기대하며…….

2013년 5월 이영수

일러두기

이 책의 프로젝트는 쉬운 것부터 시작하여 복잡한 것으로 진행된다. 그러니 여러분이 직접 책에 나와 있는 순서대로 만들어보기 바란다.

이미 아두이노를 경험한 사람이라면, 각 프로젝트의 끝 부분에 제시한 ‘해볼 만한 것들’을 참고하라. ‘해볼 만한 것들’에서는 장치를 개조하거나 프로그램을 변경해볼 것을 제안하는데, 이런 과정을 통해 아두이노에 좀 더 숙달될 것이다. 여러분 자신의 아이디어로 장치를 개조해보고, 그 경험을 우리에게 알려주기를 희망한다.

장치를 만들어 시험해본 후 프로젝트의 난이도를 높이고 싶다면, 부품을 납땜으로 연결하여 보다 영구적으로 다시 만들면 된다. 여러분이 만든 장치를 보관할 만한 유용하고 튼튼한 케이스에 관한 내용을 10장에 담았으니, 단순하거나 세련된 케이스를 선택하여 만들어보기 바란다.

장치를 만드는 방법을 최선을 다해 알기 쉽게 설명했지만, 아무리 간단한 프로젝트라 하더라도 가끔은 여러분이 느끼기에 곤혹스럽기도 할 것이다. 즐겁고 흥미롭게 만들 수 있도록 몇 가지 팁을 소개하겠다.

분해하기

맨 처음 만든 장치가 제대로 작동하는 일은 그리 흔치 않다. 하지만 실망은 금물이다. 필자들이 만든 장치들 대부분도 처음부터 제대로 작동하지는 않았다. 우리는 여러분에게 도움이 될 만한 해결책을 발견했다. 장치를 개별 부품 단위로, 입력과 출력 부품으로 구분하여 분해하는 것이다.

사전 준비 철저히 하기

다른 부품에 연결하기 전에, 각각의 부품이 제대로 작동하는지 반드시 확인하라. 부품이 제대로 작동한다면 장치에 연결했을 때도 제대로 작동할 것이다.

저장, 백업, 문서화

코딩할 때는 다음의 사항을 필히 지켜라.

1. 저장: 코딩 작업을 하면서 수시로 코드를 저장하라.
2. 백업: 작성한 코드를 하드디스크 이외의 장치(C 드라이브와는 물리적으로 분리된 드라이브, 메모리 카드, USB 드라이브)에 항상 백업하라.
3. 문서화: 코드를 작성할 때는 각각의 코드가 어떤 일을 하는지 주석을 달라(1장 '코드에 설명을 달는 이유' 참조). 며칠만 지나도 각각의 코드가 무슨 코드인지 가물가물해진다. 장치를 만들면서 알게 된 내용을 메모해 둔다면, 나중에 참고하기가 좋을 것이다.

이처럼 간단한 세 가지 사항을 지속적으로 지켜나간다면 컴퓨터가 망가지거나, 노트북이 콘크리트 바닥에 떨어지거나, 고양이가 키보드를 밟고 지나가더라도 걱정할 필요가 없다. 작업 내용의 복사본이 항상 안전하게 다른 곳에 저장되어 있을 것이기 때문이다.

변경은 한 번에 하나씩만

코드나 장치의 디자인을 바꾸고자 한다면(여러분이 그러기를 진심으로 바란다), 한 번에 한 가지씩만 변경하여 시험해본 다음에 다른 것을 변경하라. 변경에 따라 장치의 작동이 멈출 수도 있기 때문에, 이 과정은 아주 중요하다. 만약 하나만 변경한다면, 그것 하나만 취소함으로써 장치의 이전 작동 버전으로 쉽게 되돌릴 수 있다. 이렇게 하면 어떤 문제라도 쉽게 고칠 수 있어서 자신 있게 진행할 수 있을 것이다.

합치기

책에 나온 장치들은 모듈 방식으로 설계되고 만들어졌기 때문에, 약간의 수정만으로 입력과 출력 부품을 교환할 수 있다. 온도 장치의 출력을 디스플레이가 아닌 패치베이 Pachube⁰¹에 올리고 싶은가? 한번 시도해보라! 하드웨어는 쉽게 개조할 수 있고, 코드 변경은 한 장치의 코드를 잘라 다른 장치에 붙여 넣는(cut & paste) 정도의 간단한 작업이다.

사실 어떤 변경은 이해가 되지 않을 수도 있다. 온도계에 오디오 출력을 붙인다면 좀 괴상할 것이다. 그러나 하고 싶다면 시도해보라. 무엇이 만들어질지 누가 알겠는가? 여기, 좋은 매시업 mashup 예가 있다. 측정기의 오디오 출력을 가이저 계수기의 입력과 합친 다음 코드를 약간 수정하면, 영화에서나 볼 수 있는 옛날 방식의 ‘뚝, 뚝, 뚝’ 거리는 방사선 검출기를 만들 수 있다.

도움 청하기

도움을 청하는 것을 부끄러워하지 말기 바란다. 아두이노 프로젝트에 관한 도움이라면 더더욱 그렇다. 아두이노 생태계 전체는 ‘누구나 지식에 관해 무료로 접근할 수 있다’는 철학에 근거하여 만들어졌다.

회로를 만들거나 코드를 작성하는 것에 대해 어떤 사람이 여러분보다 더 많이 알고 있을 수 있지만, 여러분 또한 다른 사람보다 더 많이 알 수도 있다.

01 **역자주** Xively(Pachube(발음은 패치베이)에서 시작하여 Cosm로 변경되었다가 지금에 이르렀다)는 온라인 데이터베이스 서비스로, 개발자들이 센서로부터 얻은 데이터(예를 들면 물건이나 기기, 빌딩에서 얻은 에너지와 환경 데이터)를 웹에 연결하고, 그 데이터를 기반으로 한 애플리케이션을 만들도록 해준다. 2007년, 설계자인 유스만 헤이그 Usman Haque가 만들었다. 2011년 일본에서 발생한 원자로 사고 당시, 자원봉사자들이 일본 전국의 가이저 계수기를 연결하여 방사능 낙진을 모니터링하는 데 사용했다. Xively는 2011년 7월 로그인인 LogMeIn[®]에 흡수되었다.

그 누구도 아두이노의 모든 기능을 활용하지는 못한다. 때문에 어떤 의미에서는 아두이노에 관한 한 전문가가 없다. 온라인을 통해서든 얼굴을 맞대고서든, 여러분이 예의 바르고, 품위 있고, 남과 정보를 공유하는 걸 주저하지 않는다면 누구든 여러분을 즐겁게 도와줄 것이다. 마찬가지로 여러분이 몇 주 동안 질문만 하다가 처음으로 다른 사람의 문제에 도움을 준다면, 그때 맛보게 될 짜릿함은 무엇과도 비교할 수 없을 것이다.

아두이노 사용자들과 의견을 나눌 수 있는 공간을 소개한다.

- 아두이노의 기능 소개, 사용자 포럼 등의 온라인 홈: <http://www.arduino.cc>
- 오라일리(이 책의 원서 출판사)의 활발한 아두이노 사용자 커뮤니티: <http://forums.oreilly.com/>
- 직접 방문할 수 있는 메이커^{maker} 작업장이나 인근의 사용자 커뮤니티를 찾는 데 유용한 해커스페이스 위키: <http://hackerspaces.org/wiki/Hackerspaces>

실험을 두려워하지 마라

이 책에서 다루는 모든 기기는 한 가지 방법 이상으로 조립할 수 있다. 전자기파 측정기에 사용할 1메가옴 저항이 없는가? 470킬로옴과 560킬로옴 저항으로 대신할 수 있다. 두 저항을 합치면 1메가옴을 약간 넘지만 사용상 문제는 없다.

물론, 이 책에서 소개하는 방법과는 다른(아마 더 좋은) 방법으로 여기에 있는 모든 장치를 만들 수도 있다. 여러분이 이러한 방법을 알아내서 우리에게 알려주기 바란다. 예제 코드들은 이 책의 공식 코드 보관소인 기트허브^{GitHub}(<https://github.com/ejgertz/EMWA/>)에서 다운받을 수 있다. 필자들이나 기트허브의 이용자들이 버그를 수정한 코드나 기능이 확장된 코드를 업데이트하면 다운받을 수 있도록, 이 보관소를 지속적으로 지켜보기 바란다.

유념할 사항

이 책에서 논의되는 기술, 콘텐츠와 기술의 소유자가 도입을 모색하는 기술의 한계, 그리고 이 기술의 사용을 실제적으로 제한하는 법률은 지속적으로 변하고 있다. 따라서 책에서 소개하는 일부 프로젝트는 작동하지 않을 수 있고, 사용 중인 시스템에 의도하지 않은 피해를 끼칠 수도 있으며, 현재의 법이나 사용자 계약과 일치하지 않을 수 있다.

적절한 장비와 안전장치를 사용하고, 본인에게 적절한 수준의 기술과 경험이 있는지 판단하는 것 등 여러분의 안전은 모두 여러분 자신의 책임이다. 책에서 진행하는 프로젝트를 만들기 위해 사용하는 전원과 다른 재료는, 사전에 안전장치를 갖추고 충분히 주의하여 적절하게 사용하지 않으면 매우 위험할 수 있다. 프로젝트는 또한 어린이들이 따라해서는 안 된다.

최대한 주의를 기울여 이 책을 준비했지만, 오라일리 미디어와 저자, 한빛미디어와 역자는 착오나 누락에 대해 책임지지 않는다. 다시 한 번 말하지만 『아두이노와 환경모니터링』에서 기술한 설명이나 제안을 이용하는 것은 여러분의 책임하에 있다. 오라일리 미디어와 저자, 한빛미디어와 역자는 혹시 발생할지도 모르는 손상이나 부상, 비용을 책임지지 않는다. 여러분의 활동이 지적재산권을 포함한 법률을 준수하는 것 역시 여러분의 책임이다.

예제 파일

책에서 사용된 예제 파일은 이 책의 공식 코드 보관소인 기트허브(<https://github.com/ejgertz/EMWA>)에서 받을 수 있다.

한빛 eBook 리얼타임

한빛 eBook 리얼타임은 IT 개발자를 위한 eBook입니다.

요즘 IT 업계에는 하루가 멀다 하고 수많은 기술이 나타나고 사라져 갑니다. 인터넷을 아무리 뒤져도 조금이나마 정리된 정보를 찾는 것도 쉽지 않습니다. 또한 잘 정리되어 책으로 나오기까지는 오랜 시간이 걸립니다. 어떻게 하면 조금이라도 더 유용한 정보를 빠르게 얻을 수 있을까요? 어떻게 하면 남보다 조금 더 빨리 경험하고 습득한 지식을 공유하고 발전시켜 나갈 수 있을까요? 세상에는 수많은 종이책이 있습니다. 그리고 그 종이책을 그대로 옮긴 전자책도 많습니다. 전자책에는 전자책에 적합한 콘텐츠와 전자책의 특성을 살린 형식이 있다고 생각합니다.

한빛이 지금 생각하고 추구하는, 개발자를 위한 리얼타임 전자책은 이렇습니다.

1. eBook Only: 빠르게 변화하는 IT 기술에 대해 핵심적인 정보를 신속하게 제공합니다.

500페이지 가까운 분량의 잘 정리된 도서(종이책)가 아니라, 핵심적인 내용을 빠르게 전달하기 위해 조금은 거칠지만 100페이지 내외의 전자책 전용으로 개발한 서비스입니다. 독자에게는 새로운 정보를 빨리 얻을 수 있는 기회가 되고, 자신이 먼저 경험한 지식과 정보를 책으로 펴내고 싶지만 너무 바빠서 엄두를 못 내는 선배, 전문가, 교수분에게는 보다 쉽게 집필하실 기회가 되리라 생각합니다. 또한 새로운 정보와 지식을 빠르게 전달하기 위해 O'Reilly의 전자책 번역 서비스도 하고 있습니다.

2. 무료로 업데이트되는, 전자책 전용 서비스입니다.

종이책으로는 기술의 변화 속도를 따라잡기가 쉽지 않습니다. 책이 일정한 분량 이상으로 집필되고 정리되어 나오는 동안 기술은 이미 변해 있습니다. 전자책으로 출간된 이후에도 버전 업을 통해 중요한 기술적 변화가 있거나, 저자(역자)와 독자가 소통하면서 보완되고 발전된 노하우가 정리되면 구매하신 분께 무료로 업데이트해 드립니다.

3. 독자의 편의를 위하여, DRM-Free로 제공합니다.

구매한 전자책을 다양한 IT기기에서 자유롭게 활용하실 수 있도록 DRM-Free PDF 포맷으로 제공합니다. 이는 독자 여러분과 한빛이 생각하고 추구하는 전자책을 만들어 나가기 위해, 독자 여러분이 언제 어디서 어떤 기기를 사용하더라도 편리하게 전자책을 볼 수 있도록 하기 위함입니다.

4. 전자책 환경을 고려한 최적의 형태와 디자인에 담고자 노력했습니다.

종이책을 그대로 옮겨 놓아 가독성이 떨어지고 읽기 힘든 전자책이 아니라, 전자책의 환경에 가능한 최적화하여 쾌적한 경험을 드리고자 합니다. 링크 등의 기능을 적극적으로 이용할 수 있음은 물론이고 글자 크기나 행간, 여백 등을 전자책에 가장 최적화된 형태로 새롭게 디자인하였습니다.

앞으로도 독자 여러분의 충고에 귀 기울이며 지속해서 발전시켜 나가도록 하겠습니다.

지금 보시는 전자책에 소유권한을 표시한 문구가 없거나 타인의 소유권한을 표시한 문구가 있다면 위법하게 사용하고 계실 가능성이 높습니다. 이 경우 저작권법에 의해 불이익을 받으실 수 있습니다.

다양한 기기에 사용할 수 있습니다. 또한 한빛미디어 사이트에서 구입하신 후에는 횡수에 관계없이 다운받으실 수 있습니다.

한빛미디어 전자책은 인쇄, 검색, 복사하여 붙이기가 가능합니다.

전자책은 오타자 교정이나 내용의 수정보완이 이뤄지면 업데이트 관련 공지를 이메일로 알려드리며, 구매하신 전자책의 수정본은 무료로 내려받으실 수 있습니다.

이런 특별한 권한은 한빛미디어 사이트에서 구입하신 독자에게만 제공되며, 다른 사람에게 양도나 이전되지 않습니다.

차례

01	세상에서 가장 짧은 전자공학 입문서	1
	1.1 아두이노.....	1
	1.2 전자회로와 전자 부품.....	3
	1.3 아두이노 프로그래밍.....	7
	1.4 첫 번째 스케치: LED 깜빡이기.....	8
	1.4.1 부품.....	8
	1.4.2 IDE 설치하기.....	8
	1.4.3 브레드보드에 회로 구성하기.....	9
	1.4.4 코드 작성하기.....	10
	1.4.5 해볼 만한 것들.....	12
02	프로젝트: 소음 측정기 / LED 바 출력	13
	2.1 소음 측정: 마이크로폰.....	13
	2.1.1 LED 바.....	15
	2.2 장치 만들기.....	16
	2.2.1 부품.....	16
	2.2.2 브레드보드에 회로 구성하기.....	16
	2.2.3 코드 작성하기.....	18
	2.2.4 해볼 만한 것들.....	22
03	새로운 부품: 4문자 디스플레이	23
	3.1 시험 프로젝트.....	24
	3.1.1 부품.....	24
	3.1.2 브레드보드에 회로 만들기.....	25

3.1.3 코드 작성하기.....	26
3.1.4 해볼 만한 것들.....	30

04 전자기파 간섭 측정하기(그리고 나쁜 음악 만들기) 31

4.1 EMI 발생원 찾기.....	32
4.2 장치 만들기.....	33
4.2.1 부품.....	33
4.2.2 8옴짜리 스피커.....	34
4.2.3 EMI 검출기 만들기.....	35
4.2.4 코드 작성하기.....	36
4.2.5 스케치 돌리기.....	38
4.2.6 휴대 모드 장치에 전원 공급하기.....	39
4.3 이 장치로 무엇을 측정하는가?.....	39
4.3.1 해볼 만한 것들.....	41

05 프로젝트: 물의 전도도 / 수치 출력 42

5.1 전도도는 무엇이고, 왜 관심을 가져야 하는가?.....	42
5.2 장치 만들기.....	42
5.2.1 부품.....	43
5.2.2 프로브 만들기.....	44
5.2.3 브레드보드로 회로 만들기.....	45
5.2.4 코드 작성하기.....	46
5.2.5 측정값을 읽는 법.....	48
5.2.6 해볼 만한 것들.....	49

6.1 이더넷 실드 사용하기.....	51
6.1.1 이더넷 포트.....	51
6.1.2 MAC 주소.....	52
6.1.3 IP 주소.....	53
6.1.4 SD 카드 슬롯.....	55
6.2 이더넷 실드 시험하기.....	56
6.2.1 부품.....	56
6.2.2 조립하기.....	57
6.3 SD 카드 슬롯 시험하기.....	57
6.3.1 부품.....	57
6.3.2 조립하기.....	58
6.4 해볼 만한 것들.....	58

7.1 날씨를 측정하기 위해 일기 예보관이 될 필요는 없다.....	59
7.2 가용한 측정 결과 얻기.....	60
7.3 첫 번째 전자 센서: DHT-22.....	62
7.4 코드 라이브러리 사용하기.....	63
7.5 장치 만들기.....	64
7.5.1 부품.....	65
7.5.2 브레드보드로 회로 만들기.....	65
7.5.3 코드 작성하기.....	66
7.5.4 해볼 만한 것들.....	72

08 **위치 정보가 포함된 데이터를 패치베이에서 실시간 공유하기** 73

8.1 시험 프로젝트: 패치베이에 연결하여 데이터 올리기.....74

 8.1.1 부품.....74

 8.1.2 패치베이 계정 개설하기.....74

 8.1.3 코드 작성하기.....76

8.2 해볼 만한 것들.....77

09 **프로젝트: 방사선 계수기 / 인터넷에 데이터 공유하기** 78

9.1 가이거 계수기.....80

9.2 장치 만들기.....82

 9.2.1 부품.....82

 9.2.2 브레드보드에 회로 만들기.....83

 9.2.3 코드 작성하기.....84

9.3 이 장치로 무엇을 측정하는가?.....90

9.4 고장 모드 분석.....91

9.5 해볼 만한 것들.....93

10 **장치 보관용 케이스 만들기** 94

1 | 세상에서 가장 짧은 전자공학 입문서

전자장치를 직접 만드는 사람^{DIYer, Do-It-Yourselfer}이거나 아두이노 초보자라면, 책 전반에서 다룰 장치 대부분을 프로그래밍하고 만드는 데 1장의 내용이 많은 도움이 될 것이다.

이미 자신만의 전자장치를 만들고 있는 독자라면, 필요에 따라 기억을 되살리는 기회로 1장을 읽어보기 바란다.

1.1 아두이노

아두이노는 간단히 말해, 전자공학이나 프로그래밍에 익숙하지 않은 사람들을 위해 특별히 개발한 보드 하나짜리 컴퓨터라고 보면 된다. 아두이노의 보드는 값싼 뿐만 아니라, 플랫폼(윈도와 맥 OS X, 리눅스 등)에 관계없이 작동하며, 아주 쉽게 프로그래밍할 수 있다. 아두이노의 하드웨어와 소프트웨어는 오픈소스화 되어 있으며 확장할 수도 있다.

또한, 아두이노는 강력하다. 크기가 아주 작음에도 불구하고, 아두이노는 아폴로 우주 계획^{Apollo Space Program}⁰¹에 사용한 항법 컴퓨터^{Navigation Computer}와 비슷한 컴퓨팅 파워를 갖고 있다. 거기다 가격은 항법 컴퓨터의 1/35,000 수준이다.

프로그래머, 디자이너, DIY족, 아티스트들이 아두이노의 컴퓨팅 파워와 간단함을 이용해서 양방향 센서, 공예품, 그리고 장난감을 포함한 온갖 혁신적인 장치를 만들고 있다.

이 책에 실린 모든 장치는 아두이노 우노^{Arduino Uno}(그림 1-1과 그림 1-2)를 사용하여 만들었다. 이 책을 쓸 즈음(2011년 말)에는 가장 최신 모델이었지만 여러분이 읽을 즈음이면 새로운 버전의 제품이 나왔을지도 모르겠다.

01 1961년 케네디 대통령이 추진한 미국의 우주 개발 계획으로, 인간을 달에 착륙시킨 후에 무사히 지구로 귀환시키는 것이 목적이었다. 1969년 7월 21일, 인류 최초로 우주선 아폴로 11호가 달에 착륙했다. 이후 6회에 걸쳐 달에 착륙하여 탐사 활동을 진행했고, 1972년 12월 아폴로 17호를 마지막으로 이 계획은 종료되었다.

그림 1-1 아두이노 우노(Rev. 2)의 앞면

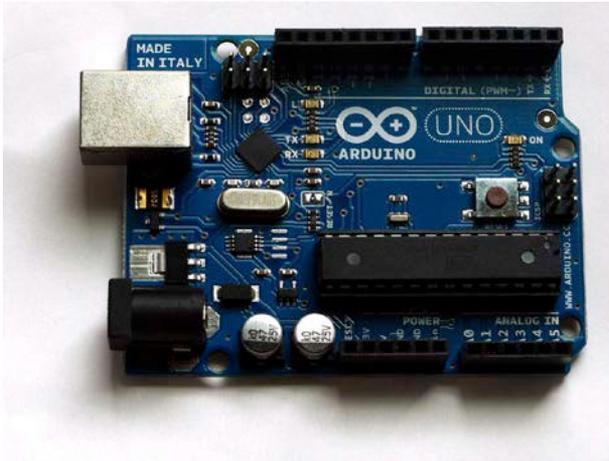


그림 1-2 아두이노 우노의 뒷면



이 책에서 다룰 장치를 만들고 프로그래밍하기 위해 아두이노 우노의 기술 규격을 꼭 알아야 필요는 없다. 하지만 아두이노의 공식 웹 사이트(<http://arduino.cc/en/Main/ArduinoBoardUno>)에 그에 관한 정보가 있으니 궁금한 독자는 찾아보기 바란다.

1.2 전자회로와 전자 부품

전자회로는 이름에서 알 수 있듯이 전기가 흐르는 원형과 매우 비슷한 길을 말한다. 각 회로는 시작, 중간, 끝이 있으며 일반적으로 끝점은 시작점에 아주 가깝다. 회로 중간에 있는 다양한 전자 부품을 통과하면서 전기가 흐르며, 이때 전류가 변하게 된다.

이 책에서 소개하는 각 장치는 아두이노와 다른 전자 부품을 연결한 회로다. 전자 부품의 일부는 전원과 전기의 흐름을 통제하고, 다른 전자 부품은 주변의 특정 조건을 감지하며, 또 다른 전자 부품은 이 조건들을 디스플레이로 출력한다.

이제 회로에 사용할 일부 부품을 살펴보자.

발광다이오드 LED, Light Emitting Diode

LED는 다양한 희토류⁰² 금속으로 만들어진 램프로, 아주 적은 전류에도 굉장히 밝은 빛을 낸다. 빛의 파장은 LED 내부 물질의 구성에 따라 결정된다(녹색, 청색, 황색, 적색 등의 가시광선과 적외선, 자외선을 발광한다).

우리가 만들어볼 장치에 사용되는 LED는 기술적으로 ‘미니어처^{miniature} LED’라고 하는데, 두 개의 다리^{wire lead}를 갖고 있는 아주 작은 램프다. 긴 다리는 양극^{Anode}이고 이보다 조금 짧은 다리는 음극^{Cathode}이다. 미니어처 LED는 직경이 2밀리미터에서 8밀리미터

02 **역자주** 란탄(Lanthan), 세륨(cerium), 디스프로슘(dysprosium) 등의 원소를 일컫는 말로 희귀 광물의 한 종류다. 희토류는 화학적으로 안정되면서도 열을 잘 전달하는 성질이 있어 삼파장 전구, LCD 연마광택제, 가전제품 모터자석, 광학렌즈, 전기차 배터리 합금 등의 제품을 생산할 때 쓰인다.

터 정도의 단일 램프, 바 형태의 표시기, 숫자/기호 표시기로 디스플레이나 발광 장치, 데이터 송신장치에 사용된다. 이 책에서 다루는 여러 종류의 환경 센서를 만들어보는 과정을 통해, 여러 형태의 LED를 다루는 법을 배울 수 있다.

저항^{Resistor}

저항은 전자공학 분야에서 부지런히 일하는 일벌 같은 존재다. 그렇다면 저항은 무슨 일을 할까? 전기가 잘 통하지 않는 물질로 만들어진 저항은 전기가 잘 흐르지 못하도록 막는 역할을 한다. 즉, 저항은 전류의 세기를 줄이는, 작고 단순한 안정기의 역할을 하는 것이다.

일부 전자 부품은 아주 섬세해서 너무 많은 전류가 가해질 경우 쉽게 타버린다. 그래서 저항의 역할이 매우 중요하다. 회로의 저항은 적정량의 전기가 부품에 흐르도록 해준다. 저항이 없는 회로는 생각할 수조차 없으며, LED를 사용하는 경우라면 필히 저항을 사용해야 한다.

이 책에서 소개하는 프로젝트를 만드는 과정을 통해 저항으로 전류를 조절하는 여러 독창적 방법을 배울 수 있다.

납땀

납땀은 도전성 금속인 납을 녹여서 다른 금속 조각을 이어 붙이는 일을 말한다. 작은 규모의 전자 공작에서는 ‘납땀 인두’라는 공구를 사용하는데, 인두의 팁에 실납^{wires of solder}을 녹인 다음, 녹은 납으로 부품을 회로에 연결한다.

납땀으로 아주 안정된 회로를 만들 수 있지만, 이 점이 되려 문제가 될 수도 있다. 납땀을 하면 부품을 재사용하거나 회로를 재구성하기 어려워지는 것이다. 납땀을 할 때는 또한 부품이 합선되지 않도록 특히 주의해야 한다. 이 책의 영역을 벗어나는 내용이라서 납땀

에 대해 더 이상 자세히 이야기할 수는 없지만, 납땀은 DIY 전자 공작에서 아주 유용한 기술이다. 납땀에 대해 좀 더 알고 싶으면 다음의 링크를 따라가도록 하라.

http://mightyohm.com/files/soldercomic/FullSolderComic_EN.pdf

납땀 대신 사용할 수 있는 방법이 있는데, 바로 브레드보드 breadboard, 뽕판을 이용하는 것이다.

납땀이 필요 없는 브레드보드

브레드보드는 선을 꽂을 수 있는 핀들이 배열된 작은 플라스틱 판이다(좀 더 자세한 내용은 아래에 있다). 선을 이용하여 아두이노를 포함한 다른 전자 부품을 핀에 연결하면 되기 때문에 납땀이 필요 없다.

브레드보드를 이용하면 이처럼 부품을 납땀하지 않고도 여러 부품과 조립품을 빠르게 시험해볼 수 있어서 회로 개발을 훨씬 쉽게 할 수 있다. 일반적으로 개발 단계에만 사용하도록 만들어졌지만 워낙 쉽고 빠르게 사용할 수 있어서, DIY족 중 많은 사람들이 최종 기기에도 브레드보드를 사용한다.

회로기판에 납땀을 하기 싫다면 브레드보드를 이용하기 바란다. 이 책에서 다루는 각 장치도 브레드보드를 사용하고 있다.

연결선 Wire

연결선은 가장 기본적인 부품으로, 회로를 통하여 전자가 이동하는 길을 만들어준다. 이 책의 프로젝트에서는 1밀리미터 직경의 점퍼 와이어 jumper wire를 사용하는데, 점퍼 와이어는 브레드보드 핀과 아두이노에 꼭 맞는 금속 팁을 갖고 있으며 여러 색상의 절연물로 피복되어 있다.



여러 색상의 점퍼 와이어를 가능한 한 많이 확보할 것! 아두이노로 회로를 만들 때는 점퍼 와이어를 아무리 많이 갖고 있어도 부족하다.

거의 대부분의 부품을 다음에 링크한 온라인 상점에서 구매할 수 있다.

- Adafruit Industries <http://adafruit.com>
- Emartee <http://www.emartee.com>
- Electronic Goldmine <http://www.goldmine-elec.com>
- SparkFun <http://www.sprakfun.com>

이 책에서 다루는 프로젝트에 필요한 전자 부품을 비롯하여 책과 장비, 키트 등을 MAKE와 오라일리 미디어의 ‘메이커 셰드(www.makershed.com)’에서 구매할 수 있다. 이 사이트에서는 특히, 책에 소개한 프로젝트들의 부품 세트 역시 판매하고 있으니 참고하기 바란다.

여러분 대부분은 아마 ‘라디오 Shack RadioShack(www.radioshack.com)’⁰³에 더 익숙할 것이다. 이 책을 쓰는 동안 필자들도 마지막 부품을 구하기 위해 라디오 Shack으로 뛰어간 게 한두 번이 아니었다.

지난 몇 년 동안 라디오 Shack은 휴대전화나 가전제품의 전망이 더 밝다고 생각하여 전자 부품의 재고를 줄여 왔다. 그러나 최근 들어 메이커 운동 Maker Movement⁰⁴을 받아들이기 시작하면서부터는 미국 내의 일부 라디오 Shack 매장에서도 아두이노를 취급하고 있으니 참고하기 바란다.

03 **역자주** 미국의 전자 기기 소매 체인점으로 북아메리카, 유럽, 중앙아메리카, 남아메리카, 아프리카 지역에 걸쳐 운영하고 있다.

04 **역자주** 필요한 것을 스스로 만드는 사람들이 만드는 방법을 공유하고 발전시키는 흐름을 통칭하는 말로, 여기서 말하는 메이커(Maker)는 기술 마니아부터 공예가(crafter), 교육자, 텅커러(tinkerer), 취미 공학자, 엔지니어, 아티스트, 과학 클럽, 학생, 저자, 자신이 제조한 물건을 파는 사람까지의 다양한 사람들을 포괄한다. 더 자세한 내용을 알고 싶다면 웹 사이트(<http://www.make.co.kr>)를 방문해보라.

1.3 아두이노 프로그래밍

컴퓨터 프로그램이란 코드로 된 연속적인 명령을 내려 일정한 기능을 수행 하도록 컴퓨터에 지시하는 프로그램을 말한다. 아두이노에서 실행되는 프로그램을 ‘스케치(sketches)’라고 한다.

이 책에서 소개하는 대부분의 스케치에 따라, 아두이노는 센서가 연결된 핀을 통해 들어온 데이터를 읽고, LED나 디스플레이 장치가 연결된 핀을 통해 정보를 쓴다.

간혹 스케치의 명령을 받은 아두이노가 입수한 정보를 특정 방법으로 처리하는데, 데이터 스트림(흐름)을 결합하거나, 입력 값을 참고 값과 비교하거나, 데이터를 읽을 수 있는 형태로 변환하기도 한다.

아두이노 프로그램은 `setup()`과 `loop()`, 두 부분으로 되어 있다.

`setup()`

원하는 작업을 하기 위해서 사전에 무엇을 알아야 하는지, 아두이노에게 설명하는 부분이다. 예를 들어 `setup()`에서는 어떤 핀들을 입력과 출력으로 할 것인지 설정하고, 따로 지정하지 않은 핀은 (디폴트로) 아무것도 하지 않게 설정한다. 결과를 표시하기 위해 4문자 표시 장치를 사용한다면, 출력을 어떻게 표시할 것인지 `setup()` 안에서 지시해야 한다. 시리얼 포트나 이더넷(Ethernet) 연결을 통하여 외부와 정보를 나누고 싶을 때도 연결을 위한 모든 명령들은 여기에 있어야 한다.

`loop()`

입력과 출력으로 무엇을 할 것인지 아두이노에게 지시하는 부분이다. 일부 컴퓨터와는 달리 아두이노는 정지하지 않는다. 아두이노가 루프 내부의 명령어를 실행하면 곧바로 `loop()`의 맨 위로 돌아가, 처음부터 명령어를 다시 실행한다.

1.4 첫 번째 스케치: LED 깜빡이기

오래된 전통(2006년까지 거슬러 올라간다)에 따라, LED를 깜빡이게 하는 스케치를 제일 먼저 작성해볼 것이다.

아두이노의 핀은 입력과 출력으로 사용할 수 있다. 그래서 이 스케치에서는 아두이노의 13번 핀을 LED OUTPUT 핀으로 설정한 다음, 번갈아 13번 핀에 전기를 보내고(핀을 HIGH로 설정) 끄는(핀을 LOW로 설정)다. 그러면, 매 교대 때마다 LED가 꺼졌다 켜졌다 할 것이다.

책에서 다루는 모든 스케치는 ‘아두이노 통합개발환경(IDE^{Integrated Development Environment}, 이하 IDE)이라는 소프트웨어를 사용하여 코드를 작성하고, 아두이노에 코드를 업로드한다.

1.4.1 부품

1. 아두이노 우노
2. 브레드보드
3. LED

1.4.2 IDE 설치하기

아두이노의 온라인 홈(<http://arduino.cc/en/Main/Software>)에서 ‘아두이노 IDE’를 다운받은 후 지시에 따라 컴퓨터에 설치한다.

소프트웨어를 설치한 다음 IDE를 실행하면 그림 1-3과 같은 화면이 나타날 것이다.

그림 1-3 Mac 컴퓨터에서 실행한 IDE

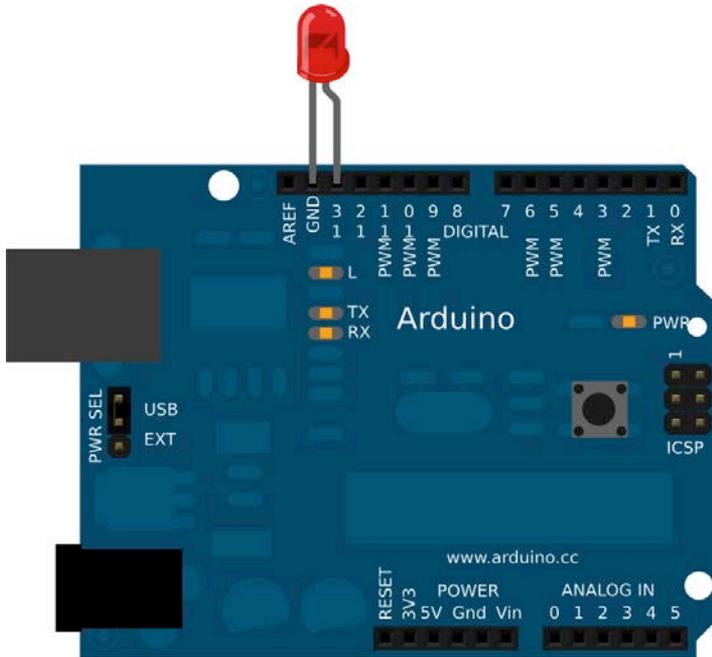


1.4.3 브레드보드에 회로 구성하기

이 프로젝트의 회로 부분은 아주 간단하다.

그림 1-4와 같이 LED의 긴 다리를 아두이노의 13번 핀에 꽂는다.

그림 1-4 LED의 긴 다리가 아두이노의 13번 핀에 꽂혀 있는 모습(그림은 Fritzing.org로 만들었다.)



1.4.4 코드 작성하기

이 코드는 아두이노 IDE를 실행한 후 ‘File > Examples’에서 볼 수 있거나, 이 책의 공식 코드 보관소인 기트허브(<https://github.com/ejgertz/EMWA/blob/master/chapter-1/blink>)에서 찾을 수 있다.

```
/*  
깜빡이  
LED를 반복적으로 1초 동안 켜다 1초 동안 끈다.  
이 예제는 공개 도메인에 등록되어 있는 예제 코드를 바탕으로 하고 있다.  
*/
```

```

void setup() {
    // 디지털 핀을 출력으로 초기화한다.
    // 대부분의 아두이노 보드는 13번 핀에 LED가 연결되어 있다.
    pinMode(13, OUTPUT);
}

void loop() {
    digitalWrite(13, HIGH);    // LED를 켜다.
    delay(1000);              // 1초 동안 유지한다.
    digitalWrite(13, LOW);    // LED를 끈다.
    delay(1000);              // 1초 동안 유지한다.
}

```

이 스케치의 loop() 안에 있는 코드는 13번 핀을 HIGH(전압을 5볼트로 높인다)로 1000밀리 세컨드(1초) 동안 유지하도록 설정하고, 그 후 LOW(전압을 0볼트로 낮춘다)로 1000밀리 세컨드 동안 유지하도록 설정한다.

예제에서 ‘/*...*/’ 부분이나 ‘//’ 라인을 알아차렸는가? 코드에 설명(주석)을 붙이는 부분으로, 다른 사람이나 자신이 알아볼 수 있도록, 각 코드가 무엇을 하는지 설명해준다. 컴퓨터에서 프로그램을 수행할 때는 ‘/*’ 와 ‘*/’ 마크 사이에 있는 모든 내용을 무시한다. 또한 이 코드 라인의 ‘//’ 이후는 단순한 설명이므로 무시하도록 지시한다.

코드에 설명을 다는 이유

코드에 설명을 다는 것은 즉, 여러분의 스케치에 코드가 어떻게 작동하는지를 평이하게 설명하는 것이다. 이것은 아주 좋은 생각이다. 그 이유는 다음과 같다.

아두이노가 어떤 작업을 수행하도록 몇 시간 동안 씨름한 끝에 해결책을 발견했다고 가정하자. 와우! 아두이노를 연결하고, 코드를 작성하고, 스케치를 로드하고…… 보라! 드디어 작동한다!

몇 달 후 여러분이 다른 프로젝트를 진행하면서 아두이노가 이전 프로젝트와 비슷한 일을 수행하게 하고 싶다면, 당연히 '문제없어. 이전에 작성했던 코드를 쓰면 되지'라고 생각할 것이다. 그러나 막상 스케치를 열어보면 아무것도 이해할 수 없다.

처음에는 대단히 창의적인 상태에서 코드를 작성했을 것이다. 하지만 뇌의 화학물질은 흐르는 강물 처럼 흘러서 지나가 버렸고, 당시의 아이디어는 한여름 번개같이 번쩍이다 사라져버렸다. 몇 달이 지나 정신 상태가 확연히 다른 지금은 이 코드가 도대체 무슨 일을 하는지 기억나지도 않고, 처음부터 다시 시작해야 한다면…… 어떻게 해야 할까?

이럴 때를 대비하여 만약 코드에 처음부터 설명을 달아둔다면, 각 변수들이 어디에 사용되었는지, 각 함수들이 어떤 일을 하는지, 각 핀이 어떤 것을 제어했는지 정확히 알 수 있다. 그러면 여러분의 인생이 훨씬 더 즐거워지지 않겠는가.

요약하면, 코드에 설명을 다는 데 몇 분 정도 투자하라.

1.4.5 해볼 만한 것들

스케치를 개조하여 LED가 조금 다른 작업을 수행하도록 만들어보자.

1. 빠르게 두 번 깜빡이기
2. 천천히 두 번 깜빡이기
3. LED를 0.5초 동안 켜 후 깜빡임 사이에 2초 정지하기

축하한다! 여러분은 이제 아두이노 프로그래머다! 지금부터는 진짜 재미있는 일을 해보자.

2 | 프로젝트: 소음 측정기 / LED 바 출력

그 누구도 소리를 냄새 맡거나 맛보거나 만질 수 없다. 그러나 소음(우리들 대부분이 싫어하는 소리)은 우리 주변에 가장 흔하게 존재하는 환경오염의 원인 중 하나다.

소음 공해는 우리가 살고, 놀고, 일하고, 공부하는 일상적인 활동을 방해하거나 우리가 원하지 않는 아주 크고 끊임없는 소리로 인해 발생하는 공해를 일컫는다. 도로의 자동차 소음, 머리 위의 비행기 소리, 건설 장비의 소음, 혹은 벽을 통하여 흘러드는 여러분 이웃의 큰 TV 소리 등이 모두 소음 공해가 되는 것이다. 소음 공해는 비단 격심한 짜증만 유발하는 것이 아니다. 미국의 환경보호 단체에 의하면 소음 공해는 스트레스와 스트레스로 인한 질병('모든 소음은 나를 병들게 한다'), 고혈압, 피로감, 난청(청력 상실) 등을 일으키는 데 직접적으로 악영향을 끼친다.

무신경한 뉴욕 시민도 유해한 소음에 관해서라면 냉정함을 잃어버리는 것 같다. 뉴욕 시 공공서비스 신고 전화에 접수되는 불만 가운데 월등히 높은 비율로 나타나는 것이 바로 원하지 않는 소음 문제다.

2.1 소음 측정: 마이크로폰

소리는 공기 분자의 움직임에 의해 발생한다. 어떤 물체가 앞뒤로 진동하면 기압파가 만들어지는데, 먼저 공기를 한쪽 방향으로 압축했다가 다음에 다른 방향으로 압축한다. 이렇게 압축된 파형은 진동의 진원지로부터 외부의 모든 방향으로 퍼져 나가서, 방해물에 부딪쳐 소멸되거나 반사되거나 점차 감쇠되어 없어진다.

우리가 사용하는 마이크로폰에 파동이 도달하면, 그 압력으로 마이크로폰의 떨림판이 진동한다. 떨림판이 진동하면 마이크로폰 뒤쪽에 부착되어 있는 자석의 자기장을 변화

시키고, 이렇게 변화된 자기장에 의해 마이크روف폰의 전선에 아주 낮은 전류가 흐른다. 우리가 만들 장치를 이용하여 바로 이 전류를 측정할 것이다.

일반적으로 마이크روف폰에 흐르는 전류는 아주 낮기 때문에, 아두이노 보드만으로는 신호의 변화를 검출하기 어렵다. 그래서 '소형 소리 센서 마이크(이마티 EmarTEE 부품 번호 part number 42021 www.emartee.com/product/41496/Mini%20Microphone)'를 선택했다. 이 마이크는 증폭기가 장착된 '쪽 보드 breakout board'에 부착되어 있는데, 이 특별한 증폭기는 아두이노 보드가 쉽게 검출할 수 있도록 미약한 신호를 증폭하며, 다른 용도로도 쉽게 사용할 수 있게 해준다.

이마티 소형 소리 센서를 구할 수 없으면, 자메코 Jameco(부품 번호 ECM-60PC-R www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10001_136574_-1)의 마이크를 사용해도 된다. 단, 이때는 아두이노 스케치를 조금 수정해야 할 수도 있다는 걸 유념하기 바란다.

소음 공해로부터 고래 구하기

압력파가 공기를 통해서 전달된다고 말했지만, 정확히 말해 압력파는 거의 모든 연속 매질을 통해 쉽게 전달된다(금속, 유리, 물까지도). 실제로, 주로 선박 엔진에 의해 발생하는 바닷속 소음이 고래나 돌고래 같은 포유동물에게 해가 된다는 사실이 점점 명확해지고 있다.

바닷속 포유동물은 수중 음파로 의사소통을 하는데, 소음기(消音器)가 장착되지 않은 선박의 엔진 때문에 의사소통하는 데 어려움을 겪는다. 수중 음파 기록 장치를 연구에 사용한 과학자들이 2010년 출간한 연구보고서⁰⁵에 의하면, 멸종 위기종인 북대서양참고래가 상대를 찾기 위해 바닷속 소음보다 큰 소리를 내고 있다는 사실이 밝혀졌다. 상대를 찾지 못하면 짹짹기를 못하고 결국 새끼를 낳을 수 없게 된다.

장치를 개조하여 바닷속의 소음을 들을 수 있게 된다면, 복잡하겠지만 아주 훌륭한 프로젝트가 될 것이다. 이를 위해 마이크روف폰은 방수가 되어야 하고 돌고래나 고래가 사용하는 주파수를 잡을 수 있도록 설계되어야 한다. 아두이노 자체도 방수가 되어야 하고 데이터를 저장하거나('6.1.4 SD 카드 슬롯' 참조), 다른 장치에 데이터를 출력하는 방법이 있어야 한다.

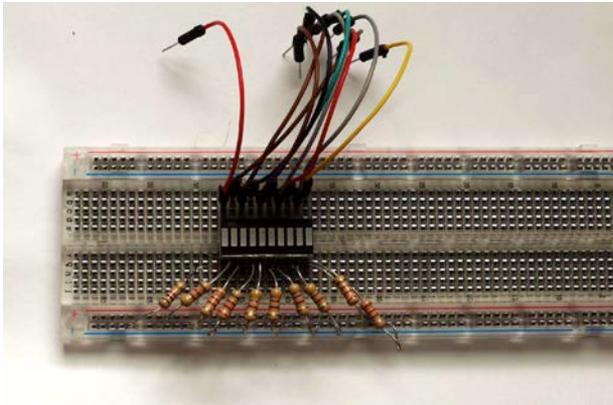
만약 이미 시도해보았다면, 그 결과를 우리에게 알려주기 바란다.

05 <http://www.onearth.org/blog/right-whales-yell-over-noise-pollution> 참고.

2.1.1 LED 바

LED 바 디스플레이는 여러 개의 LED를 멋진 플라스틱 케이스에 모아둔 것이다. 다른 회로는 포함되어 있지 않고(그림 2-1 참조), 심지어 전류를 제어하기 위한 저항조차 내장되어 있지 않다. 때문에 LED 바를 사용하고 싶지 않다면 굳이 사용하지 않아도 된다. 스파크펀 SparkFun([sku COM-09935www.sparkfun.com/products/9935](http://sku.COM-09935www.sparkfun.com/products/9935))을 비롯한 전자 부품 업체에서 구할 수 있다.

그림 2-1 아두이노에 연결할 점퍼 와이어와 함께, 브레드보드에 LED 바 디스플레이가 꽂혀 있는 모습. 개별 LED로도 LED 바 디스플레이를 만들 수 있다.



LED 바 대신 표준형 LED 몇 개를 사용해도 좋지만, 이때는 여러분이 사용하는 LED의 개수를 스케치의 'number_of_LEDs' 변수에 반영해야 한다는 걸 잊어서는 안 된다.

개별 LED를 쓰면, 색깔로 소음의 강도를 구분할 수 있다는 장점이 있다. 녹색 LED 5개, 황색 LED 3개, 적색 LED 2개를 위험 정도에 따라 표시되도록 해보자.

2.2 장치 만들기

2.2.1 부품

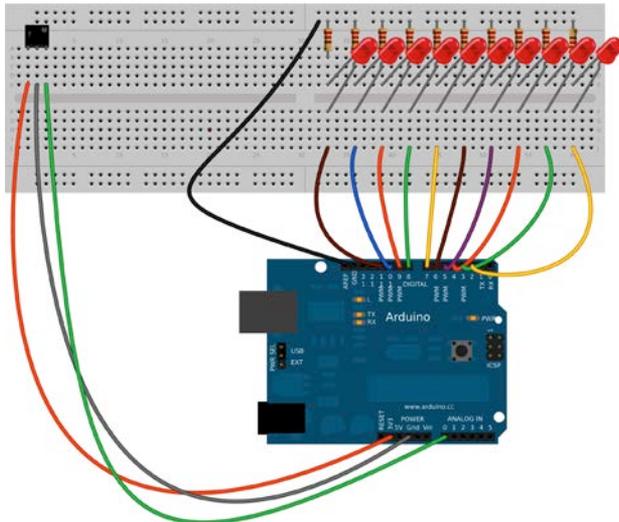
1. 아두이노
2. 브레드보드
3. 소형 소리 감지 마이크로폰(Emartee part number 42021)
4. 한 가지 색 혹은 여러 가지 색의 LED 5~10개, 또는 LED 바 디스플레이
5. 220옴 저항
6. 여러 색깔의 점퍼 와이어 10~15개

2.2.2 브레드보드에 회로 구성하기

회로를 만드는 방법은 다음과 같다.

1 단계 브레드보드에 마이크로폰을 꽂는다(그림 2-2).

그림 2-2 소음 센서가 브레드보드에 꽂혀 있고, 점퍼 와이어가 그라운드(GND), 전원, 데이터 핀에 각각 연결되어 있는 모습



2단계 마이크로폰의 그라운드(GND) 핀과 아두이노의 그라운드(GND) 핀을 선으로 연결한다.

3단계 마이크로폰의 전원(Power) 핀과 아두이노의 전원(power) 핀을 선으로 연결한다.

4단계 마이크로폰의 데이터(Data) 핀을 아두이노의 0번 아날로그(Analog) 핀에 연결한다.

5단계 아두이노의 2번 디지털(Digital) 핀을 브레드보드의 한 곳에 꽂는다.

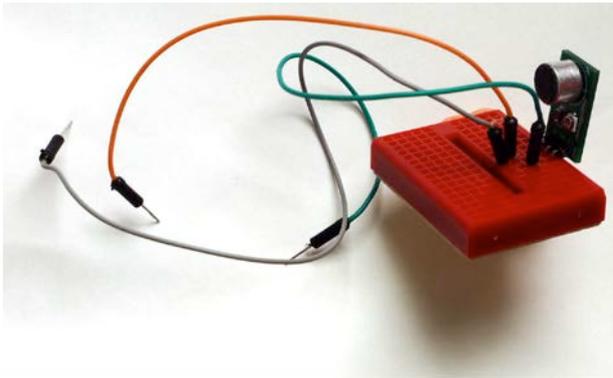
6단계 LED의 긴 다리 혹은 애노드(또는, LED 바의 애노드)를 D2로부터 연결된 점퍼 선이 꽂힌 곳(5단계)의 줄에 꽂는다. LED의 짧은 다리 혹은 캐소드(또는, LED 바의 캐소드)를 벌러 브레드보드 중앙의 홈을 건너 브레드보드의 다른 쪽 대응하는 줄에 꽂는다.

7단계 220옴 저항을 캐소드가 꽂힌 줄과 그라운드 레일에 꽂는다.

8단계 그라운드 레일에서 아두이노의 그라운드 핀으로 선을 연결한다.

완성된 최종 회로는 그림 2-3에서 확인할 수 있다.

그림 2-3 완성된 소음 측정기의 회로



5단계에서 7단계까지 아홉 번 반복하거나 사용하고자 하는 모든 LED에 대해 한 번씩 반복한다. 빛이 보기 좋게 줄지어 표시되도록 아두이노의 디지털 핀 번호와 브레드보드의 줄을 하나씩 증가시킨다.

머리를 쥐어뜯고 싶지 않다면, 각각의 LED에 같은 색깔의 연결선을 사용하지 말기 바란다. LED를 아두이노의 핀에 잘못 꽂으면 믿을 수 없을 정도로 실수를 찾아내기 어렵기 때문이다. 색깔을 교대로 쓰거나 무지개 색깔 전체를 사용하도록 한다.

2.2.3 코드 작성하기

아래의 코드는 깃허브(<https://github.com/ejgertz/EMWA/blob/master/chapter-2/NoiseMonitor>)에서 다운로드 받을 수 있다.

```
/*
소음 측정기
장치가 소음을 감지하도록 하는 스케치다.
이 예제는 공개 도메인에 등록되어 있는 예제 코드를 바탕으로 하고 있다.
*/

int sensorPin = A0; // 입력 장치를 연결할 입력 핀을 설정한다.

const int numberOfLEDs = 10;

const int numberOfSamples = 16;
int sample;
long signal[numberOfSamples];
long runningAverage;
long sumOfSamples = 0;
int counter =0;

int threshold[] = { 0, 47, 99, 159, 227, 308, 407, 535, 715, 800, 900 };

// 위의 임계 값을 지우고 아래의 임계 값을 사용하면
// LED의 민감도를 변경할 수 있다.
```