아두이노 Arduino
상상을 스케치하다

아두이노 Arduino 상상을 스케치하다

초판 1쇄 발행 2014년 5월 30일

지은이 허경용
펴낸이 장성두
펴낸곳 제이펍

출판신고 2009년 11월 10일 제406-2009-000087호
주소 경기도 파주시 문발로 141 뮤즈빌딩 403호
전화 070-8201-9010 / **팩스** 02-6280-0405
홈페이지 www.jpub.kr / **이메일** jeipub@gmail.com

편집부 이민숙, 이 슬, 이주원 / **소통·기획팀** 현지환
본문디자인 성은경 / **표지디자인** 미디어픽스
용지 에스에이치페이퍼 / **인쇄** 한승인쇄 / **제본** 광우제책사

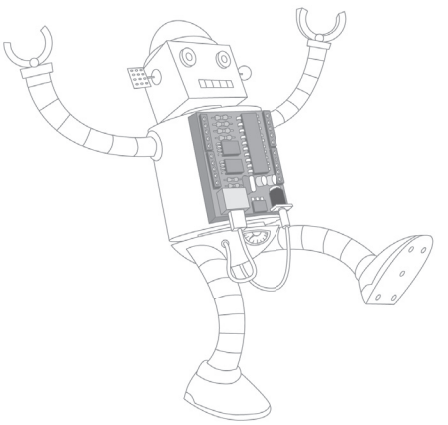
ISBN 978-89-94506-93-7 (93000)
값 30,000원

※ 이 책은 저작권법에 따라 보호를 받는 저작물이므로 무단 전재와 무단 복제를 금지하며,
이 책 내용의 전부 또는 일부를 이용하려면 반드시 저작권자와 제이펍의 서면동의를 받아야 합니다.
※ 잘못된 책은 구입하신 서점에서 바꾸어 드립니다.

제이펍은 독자 여러분의 아이디어와 원고 투고를 기다리고 있습니다. 책으로 펴내고자 하는 아이디어나 원고가 있으신 분께서는 책의 간단한 개요와 차례, 구성과 저(역)자 약력 등을 메일로 보내주세요. **jeipub@gmail.com**

아두이노 Arduino 상상을 스케치하다

센서, 디스플레이, 블루투스, DIY까지 아두이노 프로젝트의 거의 모든 것에 관하여



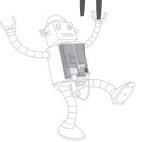
허경용 지음

Jpub 제이펍

※ 드리는 말씀

- 이 책에 기재된 내용을 기반으로 한 운용 결과에 대해 저자, 소프트웨어 개발자 및 제공자, 제이펍 출판사는 일체의 책임을 지지 않으므로 양해 바랍니다.
- 이 책에 등장하는 회사명, 제품명은 일반적으로 각 회사의 등록 상표(또는 상표)이며, 본문 중에는 ™, ©, ® 마크 등을 생략하고 있습니다.
- 이 책에서 사용하고 있는 제품 버전은 독자의 학습 시점에 따라 책의 내용과 다를 수 있습니다.
- 이 책에 사용한 그림은 대부분 저자가 직접 촬영하였으며, 그 외의 그림은 소유자의 허락하에 싣거나 출처를 표시하였습니다.
- 연습문제는 독자의 생각을 넓히는 데 도움을 주자는 취지로 만들어졌으며, ◎ 아이콘에 해당하는 문제에 한해서 해답을 제공합니다. 해답은 제이펍 홈페이지(www.jpub.kr)에서 확인할 수 있습니다.
- 이 책에는 (주)뉴티씨에서 구매할 수 있는 아두이노 호환 보드의 할인 쿠폰이 제공됩니다. 자세한 내용은 책의 맨 마지막 페이지를 참고하시기 바랍니다.
- 소스코드를 직접 타이핑해 보는 것을 권장하므로 본 도서의 소스코드는 별도로 제공되지 않습니다.
- 책의 내용과 관련된 문의사항은 저자(hgycap@deu.ac.kr) 혹은 제이펍 출판사(jeipub@gmail.com)로 연락주시기 바랍니다.

차례



머리말 x

베타리더 후기 xii

CHAPTER 01 아두이노란 무엇인가? 1

CHAPTER 02 마이크로컨트롤러 17

2.1 마이크로프로세서와 마이크로컨트롤러 17

2.2 교차 개발 환경 20

2.3 개발 프로그램 22

CHAPTER 03 아두이노 개발 환경 설정 28

CHAPTER 04 아두이노 프로그램의 구조 49

CHAPTER 05 아두이노의 기본 입출력 61

5.1 디지털 입출력 62

5.2 아날로그 입출력 70

CHAPTER 06 아두이노를 위한 C/C++ 언어 76

6.1 C/C++ 언어 테스트 환경 77

6.2 main 함수 79

6.3 데이터형 79

6.4 연산자 89

6.5 제어문 101

6.6 배열 107

6.7 함수 111

6.8 포인터 112

6.9 클래스 117

CHAPTER 07 아두이노 함수 128

7.1 디지털 입출력 함수 129
7.2 아날로그 입출력 함수 132
7.3 고급 입출력 함수 138
7.4 시간 함수 141
7.5 수학 함수 144
7.6 삼각 함수 147
7.7 난수 생성 함수 148
7.8 비트 조작 함수 150
7.9 외부 인터럽트 함수 152
7.10 인터럽트 함수 155

CHAPTER 08 기본 클래스 158

8.1 Serial 159
8.2 String 172

CHAPTER 09 라이브러리 189

9.1 기본 라이브러리 189
9.2 외부 라이브러리 195
9.3 라이브러리 만들기 200

CHAPTER 10 아두이노를 위한 전자 부품 209

10.1 저항 209
10.2 다이오드 211
10.3 커패시터 212
10.4 트랜지스터 214
10.5 집적회로 215
10.6 브레드보드 217

CHAPTER 11 시리얼 통신 사용하기 221

11.1 시리얼 통신을 통한 RGB LED 밝기 제어 222
11.2 시리얼 통신에서 숫자 형식 지정하기 225

11.3 Handshaking에 의한 시리얼 통신 연결 226
11.4 시리얼 통신을 통한 다수의 LED 제어 229

CHAPTER 12 소프트웨어 시리얼 234

CHAPTER 13 디지털 및 아날로그 입출력 248

13.1 아두이노 스케치를 위한 최소 코드 249
13.2 내장 LED 점멸 249
13.3 풀다운 저항이 연결된 버튼 입력 250
13.4 풀업 저항이 연결된 버튼 입력 252
13.5 아날로그 입력 254
13.6 PWM 형식의 아날로그 출력 257
13.7 아날로그 입력에 의한 LED 밝기 조절 259
13.8 delay 함수 없이 LED 점멸하기 261
13.9 버튼을 누른 횟수 세기 263
13.10 내부 풀업 저항을 이용한 버튼 입력 265
13.11 멜로디 재생 268
13.12 미니 키보드 276
13.13 음높이 조절하기 278
13.14 LED 미터 280

CHAPTER 14 센서 사용하기 1 285

14.1 광 센서 286
14.2 압력 센서 290
14.3 온도 센서 295

CHAPTER 15 센서 사용하기 2: 초음파 거리 센서 300

15.1 초음파 센서 모듈 – SRF05 301
15.2 초음파 거리 센서 모듈 – Grove 306

CHAPTER 16 7 세그먼트 표시 장치 312

16.1 7 세그먼트 표시 장치 314
16.2 7 세그먼트 표시 장치 + 7447 317

16.3 7 세그먼트 표시 장치 + 74164 320
16.4 7 세그먼트 표시 장치 + 74595 323

CHAPTER 17 네 자리 7 세그먼트 표시 장치 329
CHAPTER 18 LED 매트릭스 338
CHAPTER 19 텍스트 LCD 359
CHAPTER 20 그래픽 LCD 373
CHAPTER 21 TFT LCD 386
CHAPTER 22 블루투스 401
22.1 블루투스 모듈 HC-06 402
22.2 아두이노와 컴퓨터 간 블루투스 통신 406
22.3 아두이노와 스마트폰 간 블루투스 통신 413
22.4 마스터와 슬레이브 간 블루투스 통신 416

CHAPTER 23 서보 & DC 모터 423
23.1 서보 모터 424
23.2 DC 모터 430
23.3 모터 쉴드 434

CHAPTER 24 날짜와 시간 1 439
24.1 소프트웨어 RTC 440
24.2 하드웨어 RTC 443

CHAPTER 25 날짜와 시간 2: Wire 라이브러리 450
25.1 Wire 라이브러리 451
25.2 Wire 라이브러리를 이용한 RTC 모듈 제어 457

CHAPTER 26 I2C와 SPI 통신을 이용한 아두이노 연결 467
26.1 I2C 468
26.2 I2C를 이용한 아두이노 간 통신 469
26.3 SPI 471

26.4 SPI 라이브러리 472
26.5 SPI를 이용한 아두이노 간 통신 476

CHAPTER 27 디지털 입출력 확장 482
27.1 키 매트릭스 483
27.2 아날로그 입력 핀을 이용한 버튼 입력 확장 489
27.3 MCP23017 칩 492

CHAPTER 28 EEPROM 라이브러리 502
CHAPTER 29 타이머 라이브러리 509
29.1 Metro 라이브러리 513
29.2 MsTimer2 라이브러리 515

CHAPTER 30 SD 라이브러리 519
30.1 SD 클래스 521
30.2 File 클래스 524

CHAPTER 31 부트로더와 메모리 537
31.1 부트로더 537
31.2 ATmega328의 메모리 544

CHAPTER 32 DIY 아두이노 549
32.1 DEUino 회로 550
32.2 아두이노 보드를 이용한 업로드 561
32.3 프로그래머를 이용한 업로드 570
32.4 UART 시리얼 변환기를 이용한 업로드 573
32.5 8MHz 내부 클록 이용 576

CHAPTER 33 아두이노와 호환 가능한 마이크로컨트롤러 1: ATmega128 584
CHAPTER 34 아두이노와 호환 가능한 마이크로컨트롤러 2: ATtiny85 602

찾아보기 613

머리말

이탈리아 변방에서 온 아두이노는 비전공자들도 주변 환경과 상호작용하는 하드웨어를 쉽게 설계하고 제작할 수 있도록 만들어진 플랫폼이다. 마이크로컨트롤러를 조금이라도 다루어보았다면 아두이노의 기본 예제인 LED 깜빡이기를 테스트하는 순간 "쉽다"는 것을 눈치 챌 수 있을 것이다. 기억하기 쉬운 이름으로 추상화된 아두이노의 함수들은 레지스터의 이름과 각 비트의 의미를 따로 찾아보지 않아도 되도록 직관적이고 쉬운 소프트웨어 개발 환경을 제공한다. 또한, 표준화된 하드웨어 플랫폼은 다양한 종류의 확장 보드들을 쉽게 구할 수 있도록 해준다. 이처럼 쉬운 마이크로컨트롤러 개발 환경이 가능한 이유는 아두이노의 오픈 소스 정신에 있을 것이다. 그러나 이렇게 표준화되고 간편한 하드웨어/소프트웨어 환경은 다른 마이크로컨트롤러에서는 찾아보기 어려운 것이 사실이다.

아이폰이 스마트폰 시대를 이끌어 가는 선두주자가 될 수 있었던 것은 앱스토어를 통해 구축한 '생태계' 때문이었음을 우리 모두가 잘 알고 있다. 안드로이드 진영 역시 플레이스토어를 통해 이에 대응하고 있다. 아두이노 역시 마찬가지다. 만일, 초음파 센서에 대한 정보가 필요하다면 검색창에 '아두이노 초음파 센서'라고 입력해 보라. 여러 페이지에 걸쳐 나열되는 수많은 정보들은 오히려 당혹스러울 정도다. 지난 10년간 수많은 하드웨어/소프트웨어 개발자의 참여로 구축된 아두이노의 생태계는 여타 마이크로컨트롤러가 따라가지 못할 강점이며 이는 앞으로도 계속될 것이다.

하지만 아두이노를 다루기 위해서는 하드웨어는 물론이거니와 소프트웨어에 대한 지식이 선행되어야 한다. 아두이노가 다른 마이크로컨트롤러에 비해 알아야 할 하드웨어 및 소프트웨어(특히 프로그래밍에 관련된 지식) 관련 내용이 적기는 하지만 아예 없지는 않다. 만약 하드웨어와 소프트웨어에 대한 지식이 있다면, 아두이노는 생각한 바를 빠른 시간에 구현하고 확인할 수 있도록 해주는 훌륭한 프로토타이핑 도구가 되어줄 것이다. 그러나 관련 지식이 없다고 해서 실망할 필요는 없다. 관심과 흥미만 있다면 아두이노는 여느 마

이크로컨트롤러에 비해 배우기 쉽고 활용 가능성 또한 무한한 플랫폼으로서, 이를 통해 주변 환경과 상호작용할 수 있는 길을 열어줄 것이다.

이 책에는 전공자 및 비전공자들 모두가 아두이노를 사용하기 위해 필요한 하드웨어 및 소프트웨어 관련 내용을 모두 포함시키고자 노력하였다. 반복적으로 회로를 구성해 보고 코드를 입력하여 실행시켜 봄으로써 아두이노를 사용하기 위해 필요한 지식을 습득할 수 있으리라 생각한다. 책의 대부분에서 아두이노 우노를 이용하여 구상한 바를 실제로 구현하는 데 필요한 다양한 재료들을 소개하고, 이를 이용할 수 있는 방법을 보여주는 데 주안점을 두었다. 하지만 모든 내용을 다룰 수는 없었으며, 내용을 선별하는 데 개인적인 기호가 반영된 것은 당연하다. 되짚어 볼수록 아쉬운 점이 자꾸 늘어나지만 다루지 못한 내용들에 대해서는 다음 기회를 기약하고자 한다.

아두이노는 쉽다. 이제 남은 것은 상상력뿐이다. 아무런 지식 없이도 5분이면 LED에 불을 켤 수 있는 아두이노가 흥미롭기는 하겠지만, 단지 그뿐이라면 흥미로움이 오래 가지는 않을 것이다. 학습을 위한 학습이 아닌 생각한 바를 구현하기 위한 도구로서의 아두이노를 염두에 두고, 상상력이 필요한 곳을 찾아 주변을 둘러보다 보면 아두이노는 훌륭한 조력자로 자리를 지키고 있을 것이다.

지은이 **허경용**

베타리더 후기

최해성(티켓몬스터)

아두이노에 관해 상세하게, 잘 풀어 설명되어 있어 아두이노를 접해 보지 못한 사람들도 이 책으로 학습한다면 어느새 전문가에 준하는 자신을 발견할 수 있을 것입니다.

송종근(위시컴퍼니)

전문적인 지식이 없어도 하드웨어를 접해 보고 이를 통해 무엇인가를 만들어 볼 수 있는 길을 아두이노가 제공해 주고 있습니다. 이번 기회를 통해 아두이노를 통해 움직이고, 생각대로 어떤 일을 해내는 재미있는 경험을 해보았습니다. 공학 계열 전공자가 아니더라도 누구나 충분히 재미를 느낄 만하니 많은 사람들이 한 번씩 경험해 봤으면 합니다.

이아름

이 책을 보면서 대학 동안 공부했던 것이 새록새록 떠올랐네요. '그때 좀 더 열심히 공부했으면 베타리딩에 큰 도움이 되었을 텐데' 하는 아쉬움이 있습니다. 책을 보면서 공부를 더 해야겠단 생각이 드는 건 오랜만이네요. 감사합니다!

정연모(이지닉스)

전자 쪽을 전혀 모르더라도 쉽게 이해할 수 있을 만큼 자세한 설명과 개발에 대한 기초 내용이 포함되어 있습니다. 누구라도 이 책 한 권이면 아두이노를 이용하여 개발하는 데 전혀 문제가 없을 정도입니다.

최아연(숭실대학교)

베타리딩으로 읽어보는 내내 문장도 깔끔하고 설명도 섬세해서 굉장히 많은 노력이 들어간 책이구나 하는 생각이 들었습니다. 주변에 처음 아두이노를 시작하는 사람이 있다면 주저 없이 추천해 주고 싶은 책입니다.

제이펍은 책에 대한 애정과 기술에 대한 열정이 뜨거운 베타리더들로 하여금
출간되는 모든 서적에 사전 검증을 시행하고 있습니다.

CHAPTER 01

아두이노란 무엇인가?

학습 목표

아두이노는 오픈 소스 기반의 하드웨어 및 소프트웨어 개발 환경을 바탕으로 비전공자들도 손쉽게 컨트롤 장치를 만들 수 있도록 하기 위해 시작된 프로젝트이다. 이 장에서는 아두이노의 개발 배경 및 특징을 살펴보고 아두이노 사용 방법과 활용 가능성에 관해 알아본다.

아두이노(Arduino)를 인터넷에서 검색해 보면 아트멜(Atmel)사의 마이크로컨트롤러를 기반으로 만들어진 소형 보드를 가장 먼저 볼 수 있다. 그렇다면 아두이노는 마이크로컨트롤러 보드를 지칭하는 것일까? 어떤 점에서는 맞지만 정확하다고는 할 수 없다. 아두이노는 마이크로컨트롤러 보드와 더불어 보드를 이용하여 프로그램을 개발할 수 있는 소프트웨어 개발 환경까지 함께 이르는 말이기 때문이다.

아두이노 이전에도 마이크로컨트롤러는 물론이거니와 마이크로컨트롤러를 내장한 개발 보드들이 다수 존재하였다. 대표적인 마이크로컨트롤러로는 AVR*, ARM** 등이 있으며 현재도 많이 사용되는 마이크로컨트롤러들이다. AVR과 ARM이 성능 및 활용성이 이미 입증된 마이크로컨트롤러들임에도 아두이노가 필요한 이유는 무엇일까? 아두이노가 기존의 AVR이나 ARM과 다른 점은 빠른 시간에 쉽게 마이크로컨트롤러 기반의 하드웨어 제어 장치를 만들 수 있도록 해준다는 점에서 찾아야 한다. 아두이노의 기능은 기존의 마이크로컨트롤

* http://www.atmel.com

** http://www.arm.com

러 보드와 다르지 않으며, 아두이노 우노(UNO)의 경우 성능은 다른 마이크로컨트롤러 보드에 비해서도 낮은 것이 사실이다. 하지만 아두이노가 주목을 끄는 이유는 아두이노가 오픈 소스를 바탕으로 만들어졌기 때문이다. 아두이노의 하드웨어는 아트멜의 마이크로컨트롤러와 그 주변 회로들로 구성된다. 아두이노의 하드웨어 구성은 오픈 소스로 공개되어 있어 공개된 사양을 바탕으로 손쉽게 새로운 기능을 추가하거나 기존 기능을 변경할 수 있으며, 실제로 다양한 아두이노 호환 보드들이 판매되고 있다.

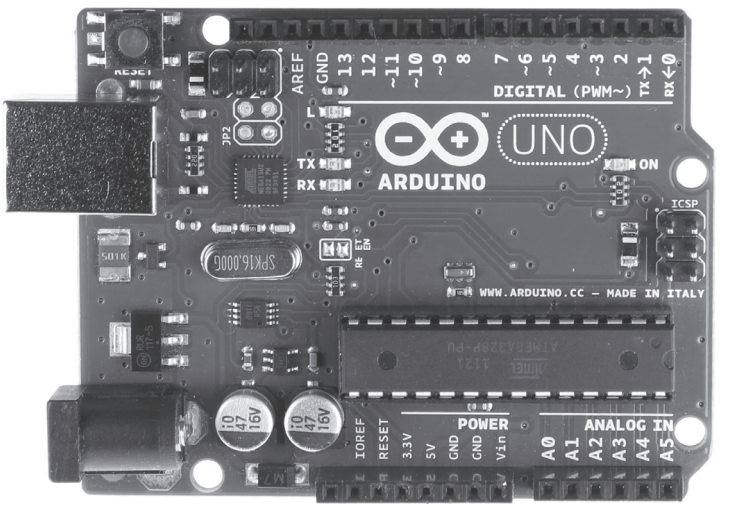


그림 1-1 아두이노 우노(UNO) 보드*

아두이노는 2005년 이탈리아 밀라노 옆에 위치한 이브레아(Ivrea)에서 예술가, 디자이너 및 학생들이 쉽게 사용할 수 있는 저렴한 컨트롤 장치를 만들 수 있도록 하기 위해 시작되었다. 프로젝트를 시작한 마시모 밴지(Massimo Banzi)와 데이비드 쿠아르티에예스(David Cuartielles)는 이 마을의 역사적 인물인 이태리의 왕 'Arduin of Ivrea'에서 그 이름을 따왔다고 한다. 2005년 발표 이후 아두이노 보드는 마이크로컨트롤러를 활용하여 제어 장치를 만들고자 하는 이들에게 주목을 받기 시작하여 그 판매량이 급증하고 있으며, 구글이 2011년 아두이노를 하드웨어 파트너로 선택한 것은 오픈 소스의 정신을 공유하고 있음은 물론 아두이노의 확장성과 편리함을 인정하였기 때문이라 하겠다.

* 출처: http://arduino.cc/en/Main/ArduinoBoardUno

아두이노 하드웨어는 이탈리아 회사인 스마트 프로젝트(Smart Projects)*에서 판매하고 있다. 하지만 아두이노는 하드웨어가 공개되어 있으므로 여러 회사에서 아두이노 호환 보드를 제작하여 판매하고 있다. 아두이노 하드웨어의 사양은 2005년 이후 몇 차례 개정 보완이 이루어져 여러 가지 공식 아두이노 보드가 판매되고 있으며** 현재 가장 많이 사용되는 보드는 아두이노 우노(UNO)이다. 아두이노 우노는 2011년 12월에 나온 R3 버전이 최신 버전이며, 이후 설명은 아두이노 우노 R3 버전을 기본으로 한다. 아두이노 우노 R3의 외형은 그림 1-2와 같다.

(a) 전면 (b) 후면

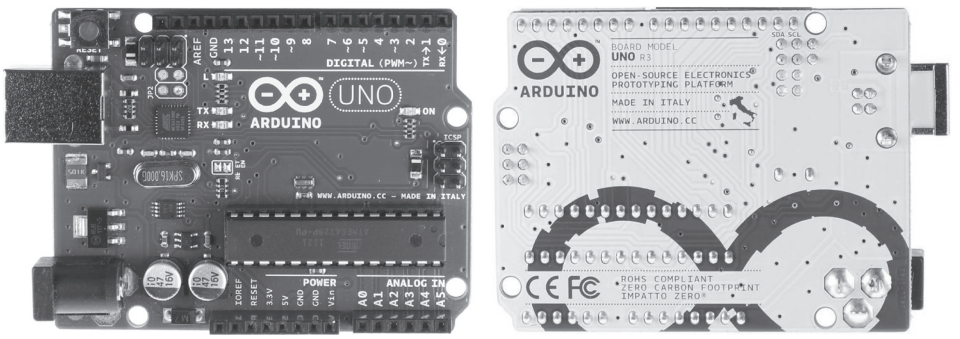


그림 1-2 아두이노 우노 R3***

아두이노 우노 보드는 ATmega328 마이크로컨트롤러를 중심으로 외부 장치 연결을 위한 핀 헤더, 전원 연결 잭, 프로그램 다운로드를 위한 USB 커넥터, 부트로더와 프로그램 다운로드를 위한 ISP 연결 커넥터 등으로 구성되어 있다.

* http://www.smartprj.com

** http://arduino.cc/en/Main/Products

*** 출처: http://arduino.cc/en/Main/ArduinoBoardUno

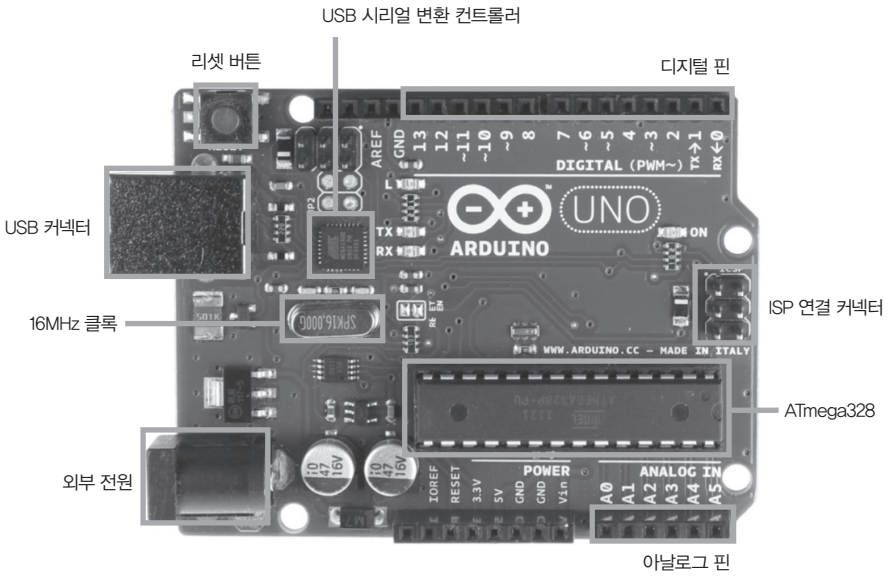


그림 1-3 **아두이노 우노 보드***

아두이노 우노는 프로그램 다운로드를 위해서 USB와 ISP 커넥터를 제공하고 있지만 USB를 통한 프로그램 다운로드를 기본으로 사용한다. 아두이노 보드와 USB 연결선만 있으면 별도의 장치 없이 아두이노로 프로그램을 작성하고 실행해 볼 수 있다. 아두이노 우노 R3의 주요 사양은 표 1-1과 같다.

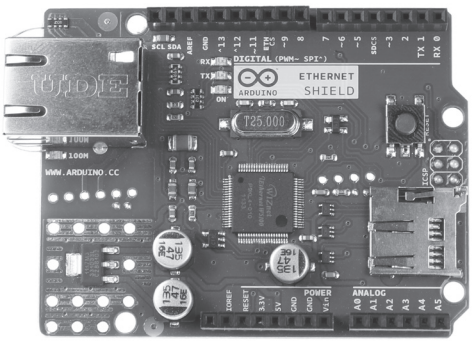
표 1-1 **아두이노 우노 R3 주요 사양**

항목	내용	비고
마이크로컨트롤러	ATmega328	
동작 전압	5V	
입력 전압	7V~12V	추천 입력 범위
디지털 입출력 핀	14개	6개 PWM 출력 핀
아날로그 입력 핀	6개	
플래시 메모리	32KB	ATmega328, 부트로더 0.5KB
SRAM	2KB	ATmega328
EEPROM	1KB	ATmega328
클록 주파수	16MHz	

* 출처: http://arduino.cc/en/Main/ArduinoBoardUno

아두이노의 또 다른 특징은 쉴드(shield)라고 불리는 다양한 확장 모듈이 존재한다는 점이다. 아두이노 보드의 핀 배치는 표준화되어 있으므로 이러한 표준을 준수한다면 누구든 손쉽게 원하는 기능을 수행하는 쉴드를 제작할 수 있다. 쉴드는 그림 1-4와 같이 아두이노 보드에 수직으로 장착하여 사용하며, 각 쉴드가 사용하는 핀이 중복되지 않는다면 여러 개의 쉴드를 적층하여 동시에 사용하는 것도 가능하다.

(a) 이더넷 쉴드*



(b) 모터 쉴드**

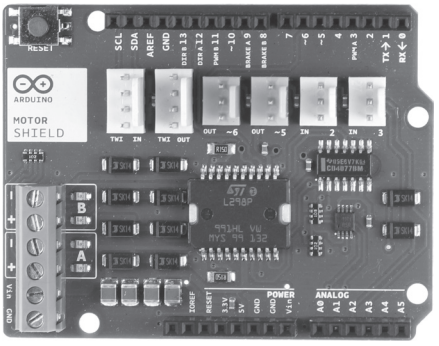


그림 1-4 아두이노 쉴드

아두이노의 장점은 간단하며, 사용하기 쉽고, 다양한 확장 쉴드가 존재한다는 하드웨어 측면에만 있는 것이 아니다. 아두이노는 하드웨어인 아두이노 보드와 소프트웨어인 통합 개발 환경(Integrated Development Environment, IDE)을 함께 일컫는 말이다. 아두이노 개발용 소프트웨어는 아두이노 보드와 마찬가지로 오픈 소스를 바탕으로 하고 있으며, 초보자들도 쉽게 프로그램을 작성할 수 있도록 직관적이고 간편한 인터페이스를 제공하고 있다.

아두이노의 통합 개발 환경은 다양한 운영체제에서 실행이 가능하도록 자바로 개발되었으며, IDE를 통해 한 번의 클릭으로 코드를 컴파일해서 아두이노 보드에 프로그램을 업로드하고 실행시킬 수 있다. 일반적으로 아두이노를 위해 만들어진 프로그램이나 코드를 스케치(sketch)라 부르며, 단어의 의미 그대로 그림을 그리듯이 프로그램을 쉽게 작성할 수 있음을 의미한다. 그림 1-5는 아두이노 프로그램을 '스케치'하는 개발 환경을 보여준다.

* 출처: http://arduino.cc/en/Main/ArduinoMotorShieldR3

** 출처: http://arduino.cc/en/Main/ArduinoEthernetShield

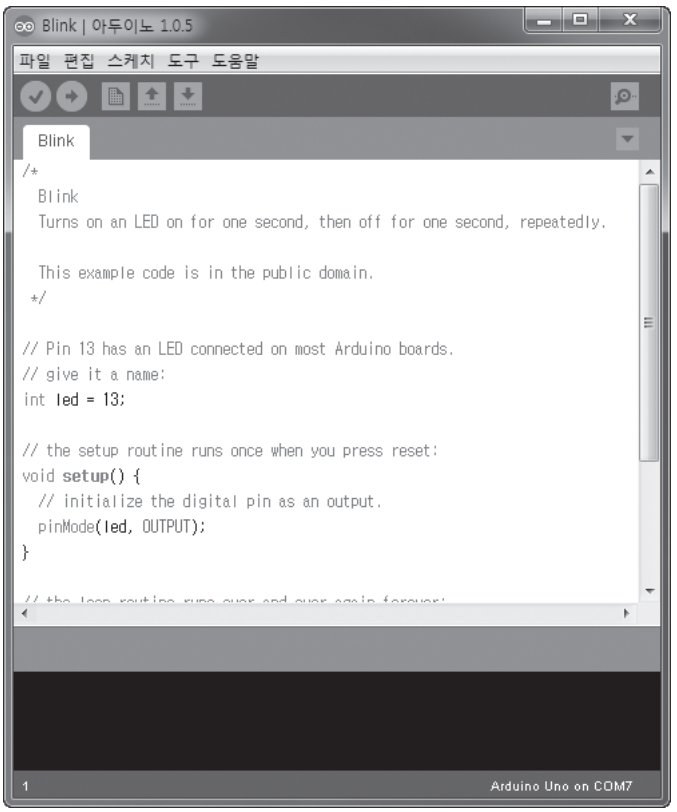


그림 1-5 **아두이노 통합 개발 환경**

아두이노 프로그램은 아두이노 공식 홈페이지*에서 다운받을 수 있으며, 이외에도 여러 가지 유용한 정보가 제공되고 있으므로 자주 들러 확인하기 바란다.

* http://arduino.cc

그림 1-6 **아두이노 공식 홈페이지**

아두이노 보드는 여러 인터넷 사이트를 통해 구입할 수 있으며, 표 1-2는 아두이노 공식 홈페이지에 나열된 국내 공식 판매 사이트들이다. 이외에도 오픈 하드웨어를 바탕으로 제작된 호환 보드를 온라인 사이트를 통해 쉽게 구할 수 있다.

표 1-2 국내 아두이노 공식 판매 사이트

사이트	주소
PlugHouse	http://www.plughouse.co.kr
MakeZone	http://www.makezone.co.kr
ArtRobot	http://artrobot.co.kr
Eleparts	http://www.eleparts.co.kr
JK DeviceShop	http://jkelec.cafe24.com
Hanjindata	http://www.smartkit.co.kr
INIPRO	http://www.inipro.net
Keytronics Lab Co.	http://www.keytronics.co.kr
Segyung Britestone	http://shop.britestone.com
SAMPLE Electronics	http://www.robot.co.kr
VCTEC Korea	http://vctec.co.kr
Robot Science Mall	http://www.robotscience.kr

아두이노가 여타 마이크로컨트롤러에 비해 가지는 장점을 요약하면 다음과 같다.

- **저렴한 가격:** 아두이노 보드는 다른 마이크로컨트롤러 플랫폼에 비해 상대적으로 가격이 싸다. 아두이노 보드는 저렴한 가격으로 직접 조립할 수도 있으며, 아두이노 우노 공식 보드의 경우 국내 판매 사이트에서 3만 5천 원 정도의 가격으로 구입 가능하다. 호환 보드의 경우 공식 보드 가격의 절반 이하로 구입할 수 있는 보드들도 다수 존재한다.
- **다양한 운영체제 지원:** 대부분의 마이크로컨트롤러 시스템이 윈도우만을 지원하는 데 반해 아두이노 개발 환경은 윈도우, 매킨토시 OS X, 리눅스를 지원한다.
- **쉽고 간단한 프로그래밍 환경:** 아두이노 프로그래밍 환경은 하드웨어나 프로그래밍에 경험이 적은 초보자들도 쉽게 접근할 수 있도록 쉽고 간단하게 만들어져 있다. 고급 기능을 원하는 고급 사용자들을 위한 유연성 역시 제공하고 있어 마이크로컨트롤러에 쉽게 입문해서 고성능 칩 활용을 위한 시작점으로서의 역할을 할 수 있다.
- **USB 지원:** 기존에 마이크로프로세서를 위해 많이 사용하던 직렬 또는 병렬 포트가 아닌 범용적인 USB를 기본으로 사용한다. 현재 AVR과 ARM 역시 직렬이나 병렬 포트 이외에 USB 연결을 지원하고 있다.
- **오픈 소스 소프트웨어:** 아두이노 개발 소프트웨어는 오픈 소스를 바탕으로 하고 있으므로 고급 사용자들이 기능을 확장할 수 있도록 하고 있다. 아두이노 개발 언어는 C++을 기반으로 하고 있으므로 라이브러리를 통해 그 기능을 확장할 수 있음은 물론이거니와 아두이노가 기반하고 있는 AVR용 C 언어를 아두이노 프로그램에 직접 사용할 수 있다.

- **오픈 소스 하드웨어:** 아두이노 보드는 아트멜의 마이크로컨트롤러를 기반으로 하고 있으며 하드웨어 사양은 CCL(Creative Commons License)로 공개되어 있다. 따라서 확장 보드를 쉽게 만들어낼 수 있으며 실제로 다양한 확장 보드들이 출시되어 있다. 또한, 아두이노 보드는 그 구조가 간단하여 어렵지 않게 브레드보드상에 직접 구현할 수 있다.

가장 많이 보급된 아두이노 우노 이외에도 여러 가지 공식 아두이노 보드들을 아두이노 사이트에서 확인할 수 있다. '아두이노 우노'는 ATmega328 마이크로컨트롤러를 사용하고 USB를 통한 프로그램 다운로드를 위해 별도의 전용 마이크로컨트롤러를 하나 더 사용한다. 반면 '아두이노 레오나르도(Leonardo)'를 비롯한 여러 보드는 USB 통신 기능이 내장된 ATmega32u4 마이크로컨트롤러를 사용하므로 USB 연결을 위한 별도의 마이크로컨트롤러를 필요로 하지 않는다. '아두이노 메가(Mega) 2560'은 ATmega 2560 마이크로컨트롤러를 사용하여 아두이노 우노에 비해 더 많은 디지털 및 아날로그 입출력을 제공하므로 많은 입출력이 필요한 경우 사용할 수 있다. '아두이노 나노(Nano)'는 아두이노 우노와 동일한 마이크로컨트롤러를 사용하면서 크기를 작게 만든 아두이도 보드로 소형 컨트롤러 제작을 원한다면 고려해 볼 수 있다. '아두이노 두에(Due)'는 ARM의 Cortex-M3를 바탕으로 한 32비트 마이크로컨트롤러를 사용하고 있어 고성능의 컨트롤러를 제작하기 위해 사용할 수 있다. 표 1-3은 아두이노 사이트의 공식 아두이노 보드들을 비교한 것으로 보다 자세한 내용은 홈페이지를 참고하면 된다.

아두이노로 할 수 있는 작업은 많지만 그 한계도 분명 존재한다. 아두이노 두에를 제외한 아두이노 보드들은 8비트 마이크로컨트롤러를 사용하고 있으며, 대부분의 보드가 16MHz에서 동작하고 32KB의 프로그램 메모리를 가지고 있다. 따라서 연산량이 많은 경우, 빠른 연산을 필요로 하는 경우, 고속의 데이터 전송을 요하는 경우에는 아두이노가 적합하지 않다. 한마디로 아두이노는 간단한 제어 장치를 만드는 데 적합하며, 아두이노가 '예술가, 디자이너 및 학생들이 쉽게 사용할 수 있는 저렴한 컨트롤 장치를 만들기 위해' 시작되었다는 점을 잊지 말아야 한다. 비전공자들을 위해서는 간단한 제어 장치를 각자의 전공 분야와 결합하기 위해, 전공자들을 위해서는 아이디어를 실제로 구현하기 이전에 가능성을 검토해 볼 수 있는 프로토타이핑 시스템으로서 그리고 본격적인 마이크로컨트롤러 학습을 위한 시발점으로 아두이노는 그 가치가 있다 하겠다. 아두이노를 이용한 다양한 프로젝트들은 인터넷 검색을 통해 쉽게 찾아볼 수 있으며 유튜브에서 다양한 작품의 실제 동작 모습들도 확인할 수 있으니 시간을 내어 검색해 보길 추천한다. 단, 다양하고 신기한 작품 감상에 시간이 지난 줄 몰라 약속 시간에 늦었더라도 필자를 원망하지는 말기 바란다.

표 1-3 아두이노 공식 보드

보드	마이크로 컨트롤러	디지털 핀 수	PWM 핀 수	아날로그 핀 수	클록 (MHz)	플래시 메모리 (KB)	기타
아두이노 우노	ATmega328	14	6	6	16	32	
아두이노 미니	ATmega328	14	6	8	16	32	• 소형 아두이노 보드 • USB 연결 커넥터 없음
아두이노 나노	ATmega328	14	6	8	16	32	• 아두이노 미니 보드에 브레드보드 장착용 핀과 USB 연결 커넥터 추가
릴리패드 아두이노	ATmega328V	14	6	6	8	16	• 입는 컴퓨터 개발을 위한 원형 아두이노 보드
아두이노 레오나르도	ATmega32u4	20	7	12	16	32	• USB 통신 기능이 내장된 ATmega32u4 사용
아두이노 마이크로	ATmega32u4	20	7	12	16	32	• USB 통신 기능이 내장된 ATmega32u4 사용 • 소형 아두이노 보드
아두이노 윤	ATmega32u4	20	7	12	16	32	• USB 통신 기능이 내장된 ATmega32u4 사용 • 이더넷 전용의 리눅스 기반 Atheros AR9331 프로세서를 함께 사용
아두이노 메가 2560	ATmega2560	54	15	16	16	256	• ATmega2560 사용으로 많은 수의 입출력 핀 제공
아두이노 메가 ADK	ATmega2560	54	15	16	16	256	• 아두이노 메가 2560에 구글 ADK 지원을 위한 USB 호스트 기능 추가
아두이노 두에	SAM3X8E (ARM Cortex-M3)	54	12	12	84	512	• 32비트 마이크로컨트롤러 사용

아두이노를 배우고 사용함에 있어 유용한 사이트 몇 군데를 소개하면서 이 장을 마무리하고자 한다. 무엇보다 가장 중요한 사이트는 아두이노 공식 사이트*일 것이다. 아두이노 공식 사이트에는 아두이노 공식 보드에 관한 정보와 학습에 필요한 참고 자료들을 제공하고

* http://arduino.cc

있다. 또한, 아두이노 놀이터(playground)*와 포럼**에서는 질문/답변 및 다양한 토론이 진행되고 있어 아두이노와 관련된 거의 모든 자료를 찾아볼 수 있다고 해도 과언이 아니다. 국내 사이트로는 아두이노 스토리 카페***가 있다. 아두이노 스토리 카페는 아두이노 관련 국내 최대의 커뮤니티로 아두이노를 비롯하여 마이크로컨트롤러와 관련된 다양한 정보들을 찾아볼 수 있다.

그림 1-7 **아두이노 스토리 카페**

아두이노 보드 구입은 표 1-2의 아두이노 공식 판매 사이트를 통해 가능하다. 표 1-2의 사이트에서는 아두이노뿐만 아니라 마이크로컨트롤러와 관련된 다양한 제품들을 판매하고 있다. 해외 판매 사이트 중에서는 스파크펀(SparkFun), 에이다프루트(Adafruit), 시드스튜디오(seeedstudio) 사이트를 방문해 보기를 추천한다. 스파크펀****은 공식 아두이노 보드뿐만 아니라 다양한 아두이노 확장 보드를 개발하여 판매하고 있으므로 유용하고 재미난 제품의 정보를 얻을 수 있다.

* http://playground.arduino.cc

** http://forum.arduino.cc

*** http://cafe.naver.com/arduinostory

**** https://www.sparkfun.com

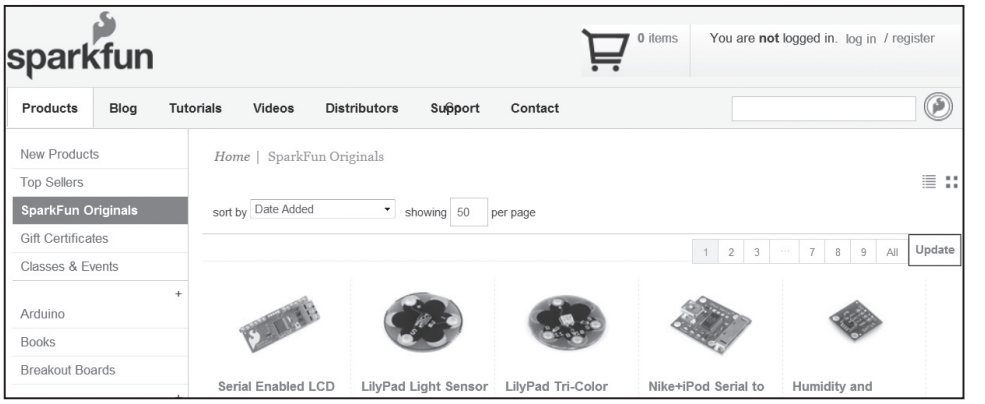


그림 1-8 **스파크펀 사이트**

에이다프루트*는 스파크펀과 유사하게 공식 아두이노 보드와 확장 보드들을 판매하고 있다. 특히 에이다프루트는 에이다프루트 러닝 시스템(AdaFruit Learning System)**을 통해 아두이노를 비롯한 다양한 주제에 관해 쉽고 친절하게 기술된 학습 자료를 제공하고 있으므로 읽어보기를 추천한다.

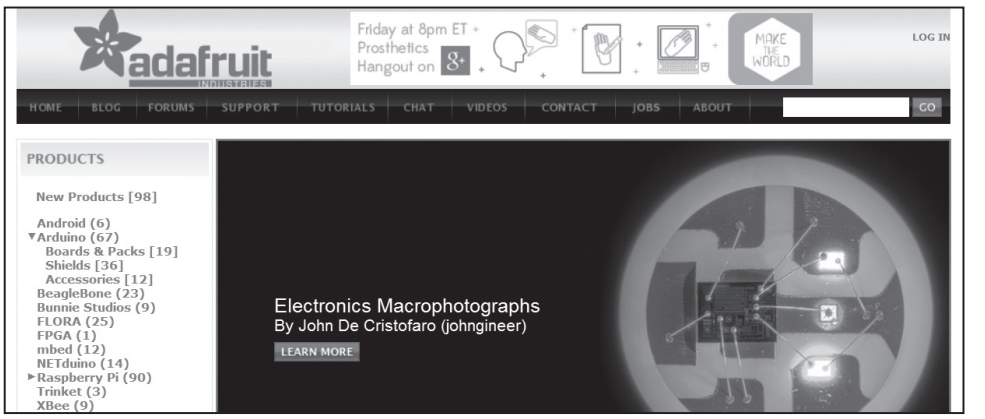


그림 1-9 **에이다프루트 사이트**

* http://www.adafruit.com/

** http://learn.adafruit.com/

시드스튜디오* 역시 아두이노 호환 보드 및 확장 모듈들을 판매하고 있다. 특히 시드스튜디오에서 제작하는 모듈들은 GROVE 시스템이라는 표준화된 입출력 인터페이스를 제공하므로 하드웨어에 익숙하지 않은 사용자들이나 모듈 개발자들은 참고할 만하다.

그림 1-10 **시드스튜디오 사이트**

아두이노 학습에 유용한 사이트들 역시 여러 곳이 있지만 체계적이고 방대한 내용을 다루고 있는 tronixstuff**를 추천한다. 국내에서 쉽게 구할 수 없는 부품을 사용하는 경우가 간혹 있어 구현에 어려움이 있는 경우가 있지만, 50개 이상의 장으로 구성된 튜토리얼은 한 권의 책에서 얻을 수 있는 정보 그 이상이다.

* http://www.seeedstudio.com/

** http://tronixstuff.com/tutorials/

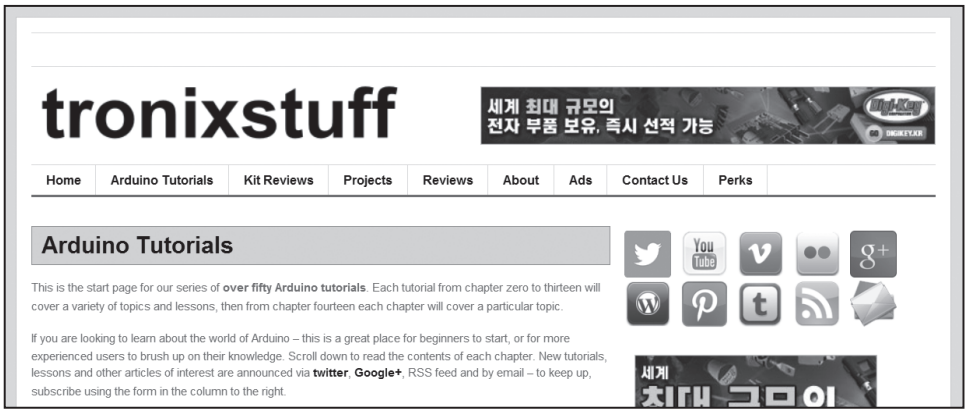


그림 1-11 tronixstuff 사이트

이외에도 open electronics*, instructables** 등에서 유용한 정보를 얻을 수 있으므로 시간을 내어 둘러보기 바란다.

* http://www.open-electronics.org/

** http://www.instructables.com/

정리

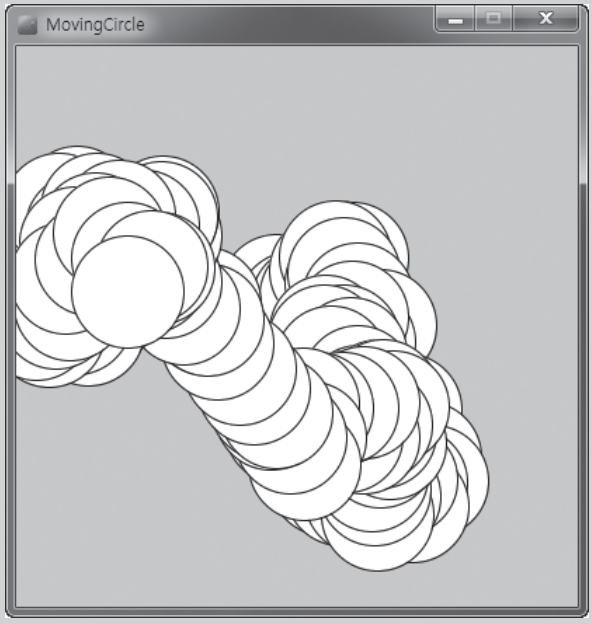
- 아두이노는 오픈 소스를 바탕으로 하는 하드웨어와 소프트웨어를 통칭하여 이르는 말로 비전공자들을 위한 플랫폼에서 출발하여 쉽고 빠르게 제어 장치를 구성할 수 있음을 가장 큰 특징으로 한다.
- 오픈 소스 하드웨어는 공개된 회로를 바탕으로 아두이노 보드 자체의 기능 개선 및 확장이 가능하도록 해주므로 다양한 호환 보드와 확장 보드들이 판매되고 있다. 원하는 기능의 보드가 없는 경우에는 아두이노의 표준을 준수하면서 직접 하드웨어를 설계하고 제작할 수도 있다.
- 소프트웨어 역시 오픈 소스를 바탕으로 하고 있으므로 무료로 다운받아 사용할 수 있으며, 비전공자들도 쉽게 사용할 수 있도록 간단하고 직관적인 사용자 인터페이스를 제공하고 있다.
- 공식 아두이노 보드에는 가장 널리 보급된 아두이노 우노를 포함하여 십여 종의 보드가 포함되어 있으며, 그 특성이 약간씩 달라 사용하고자 하는 용도에 맞게 선택하여 사용할 수 있다.
- 아두이노 관련 정보 대부분은 아두이노 공식 사이트와 놀이터, 포럼 등을 통해 얻을 수 있다.

연습문제

1. 아두이노는 다른 마이크로컨트롤러 보드에 비해 여러 가지 장점이 있다. 하지만 아두이노 역시 완벽한 마이크로컨트롤러 보드는 아니며 그 한계와 단점이 존재한다. 아두이노 보드의 한계와 단점은 무엇인지 생각해 보자.

2. 아두이노는 또 다른 오픈 소스 프로젝트인 프로세싱(Processing)을 바탕으로 만들어졌다. 프로세싱이 자바를 기반으로 비주얼 프로그래밍을 위해 만들어진 언어라면, 아두이노는 C++을 기반으로 하드웨어 제어를 위해 만들어진 언어라는 점에서 차이가 있다. 프로세싱 공식 홈페이지*에서 프로세싱 프로그램을 설치하고 다음 코드를 입력하여 무작위로 움직이는 원이 표시되는 결과를 확인해 보자. 프로세싱 프로그램은 아두이노와 마찬가지로 압축을 푸는 것만으로 실행이 가능하며 [Sketch]–[Run] 메뉴를 선택하여 프로그램을 실행시킬 수 있다. 그림 1–5의 아두이노 환경과 비교하여 프로세싱과의 차이점이 무엇인지 생각해 보자.

```
int X = 400;                    // 윈도우 크기, 넓이
int Y = 400;                    // 윈도우 크기, 높이
int max_step = 20;              // 한 번에 움직일 수 있는 최대 양
```

* http://www.processing.org/

```
int currentX = X / 2;              // 시작 위치, X
int currentY = Y / 2;              // 시작 위치, Y

void setup() {
  size(X, Y);             // 윈도우 크기 설정
}

void draw() {
  int moveX, moveY;
  int nextX, nextY;
  // 다음 원의 중심을 결정하고 윈도우 내에 중심이 있는지 검사
  do{
    moveX = (int)random(-max_step, max_step);
    moveY = (int)random(-max_step, max_step);
    nextX = currentX + moveX;
    nextY = currentY + moveY;
  }while(nextX < 0 || nextX >= X || nextY < 0 || nextY >= Y);

  currentX = nextX;                // 현재 위치 업데이트
  currentY = nextY;

  ellipse(currentX, currentY, 80, 80);      // 원 그리기
}
```