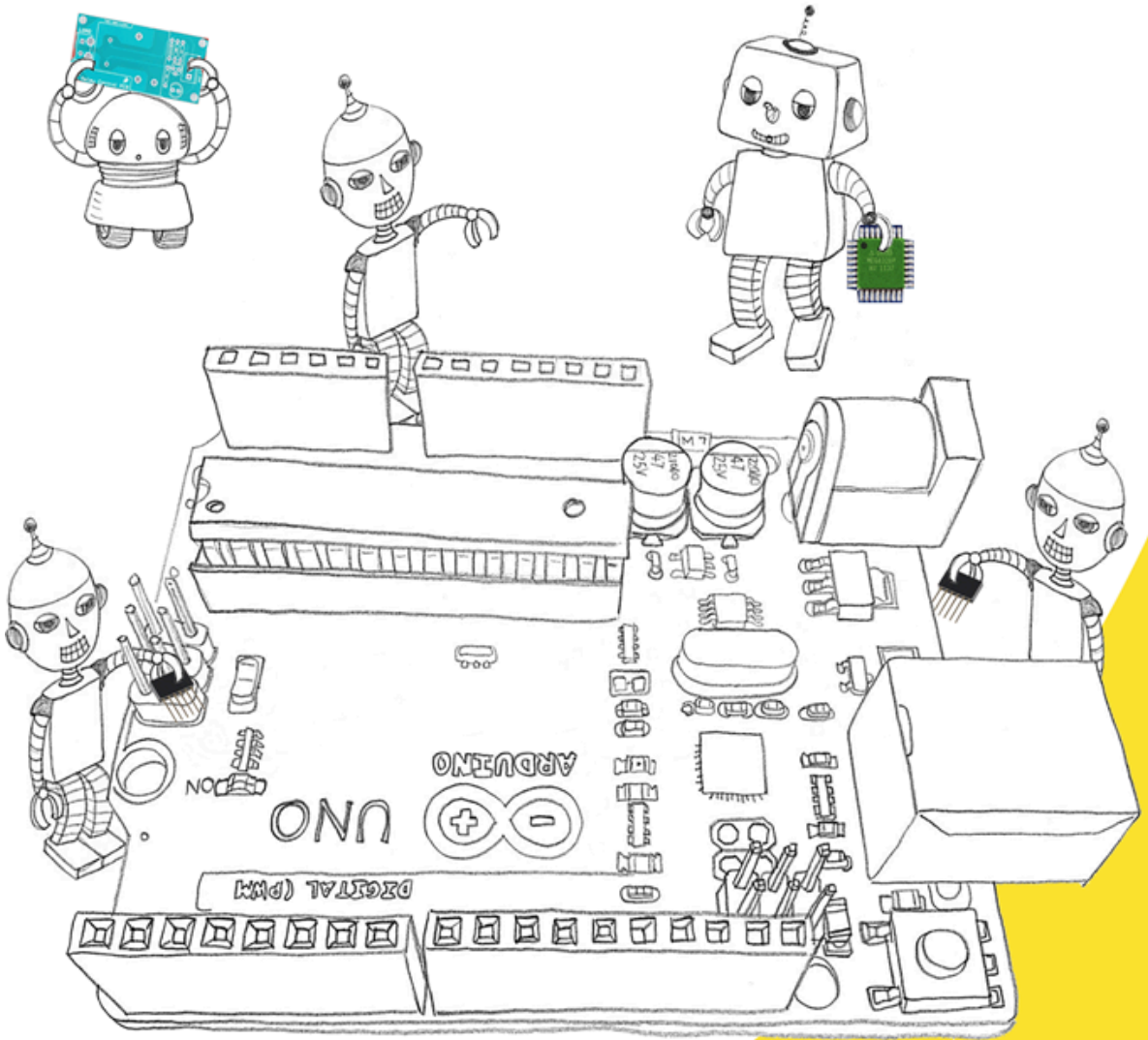


일단계 아두이노



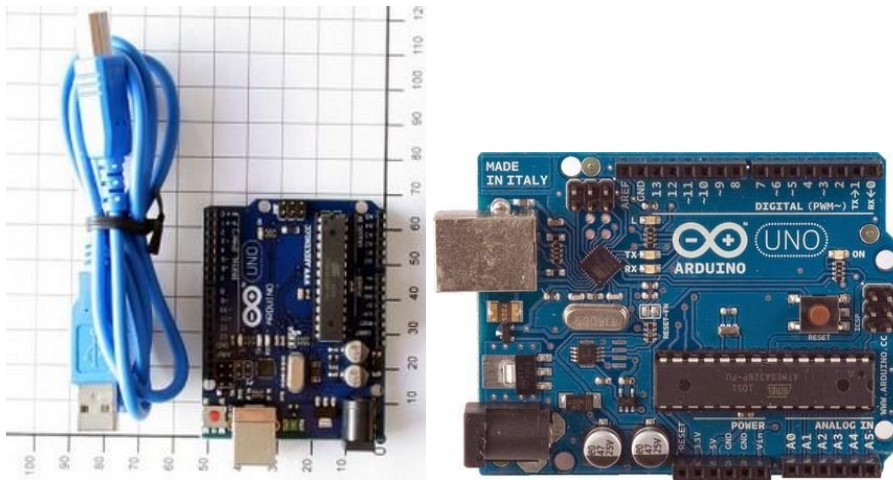
국립목포대학교 전기및제어공학과

박장현

1장 아두이노 개요

1.1 아두이노란?

아두이노(arduino)는 마이크로 컨트롤러를 기반으로 한 오픈소스 원보드 마이크로 컴퓨터(micro computer)이다. 예전에는 이런 시스템을 설계하고 주변 기기들을 연결시켜 동작시키기 위해서는 전기/전자와 관련된 전문적인 기술이나 지식이 필요했다. 이 보드의 특징은 그러한 어려움을 극복하고자 기술 숙련도가 낮은 디자이너 혹은 예술가들이 쉽게 사용할 수 있도록 설계된 것이다. 따라서 비전공자들도 손쉽게 익히고 사용할 수 있다는 큰 장점을 가지며 그로 인해 널리 사용되고 있다.

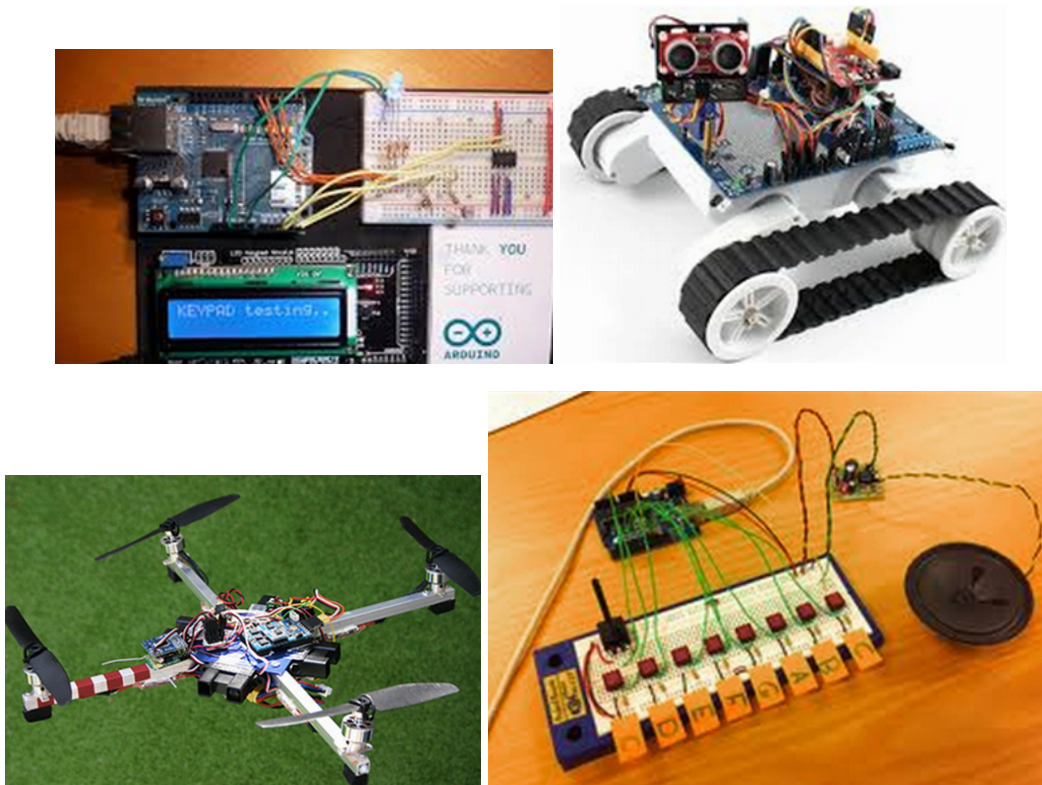


[그림 1.1.1] 가장 기본적이고 널리 사용되는 아두이노 우노 R3 보드의 외형과 크기

아두이노 보드를 이용하여 어떤 일들을 할 수 있는지 간단한 몇 가지 예를 들어보면 다음과 같다.

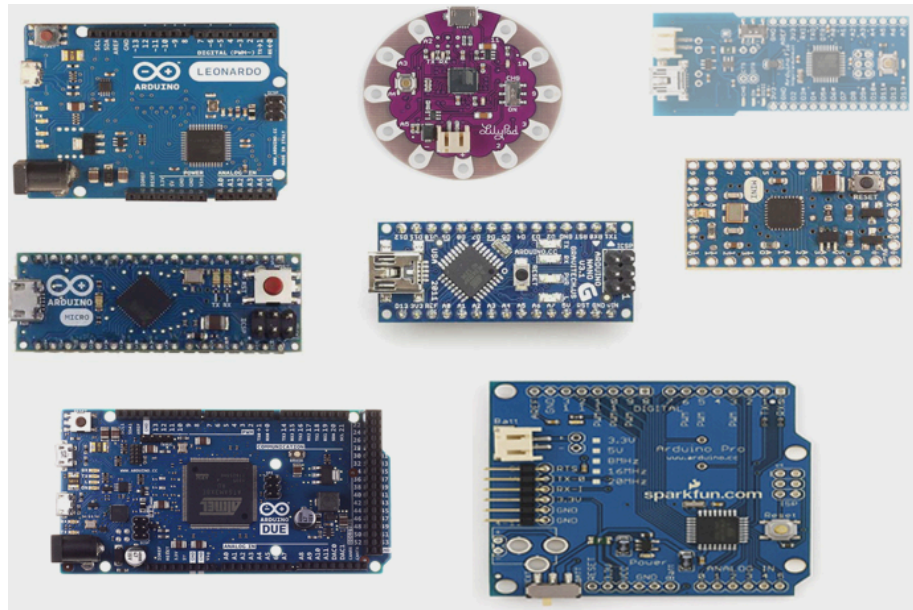
- 사람이 접근하면 자동으로 조명이 켜지는 장치
- 애완동물에게 정해진 시간에 먹이를 공급하는 장치.
- 화분의 습도가 낮아지면 자동으로 물을 주라는 트윗을 보내는 장치.
- 여러 개의 LED를 이용하여 귀걸이나 목걸이의 장식으로 이용.
- 드론(drone)의 제어 장치
- 좀 더 전문적인 응용으로 로봇의 제어 장치.

위의 예는 단순한 몇 개의 예를 든 것이고 인터넷 검색을 해보면 정말 많은 응용 제품들이 만들어지고 있으며 일반인 뿐만 아니라 기업에서도 이것을 이용하여 상용 제품을 출시하며 비즈니스 모델로 이용하고 있는 등 널리 사용되고 있다.



[그림 1.1.2] 다양한 아두이노 응용 작품들

초기(2010년 전후)의 아두이노 보드들은 AVR 이라는 8비트 마이크로컨트롤러를 기반으로 하고 있으나 최근에는 Cortex-M3를 이용한 제품(Arduino Due)을 비롯한 여러 고성능 제품군이 개발되고 블루투스 통신이나 와이파이 등 부가 기능을 더한 제품들도 많이 출시되었다. 또한 소프트웨어 개발을 위한 스케치(sketch)라는 사용하기 쉬운 통합 개발 환경(IDE)이 제공되며 이것 또한 오픈소스로 공개되어 있다. 또한 C++언어로 개발 가능하나 관련 라이브러리가 잘 갖추어져 있어서 개발 난이도는 낮은 편이다.



[그림 1.1.3] 다양한 아두이노 보드들

앞에서도 언급했듯이 아두이노의 가장 큰 장점은 전기/전자 관련 깊은 지식이 없이도 마이크로컨트롤러를 쉽게 동작시킬 수 있다는 점이다. 일반적으로 AVR 프로그래밍이 WinAVR로 컴파일하여, ISP장치를 통해 업로드를 하는 등 일반인들에게는 어렵고 번거로운 과정을 거치는데 비해서, 아두이노는 전용 IDE를 이용하여 컴파일된 펌웨어를 USB를 통해 업로드를 쉽게 할 수 있다. 또한 다른 모듈에 비해 비교적 저렴하고, 윈도우를 비롯해 맥의 OS X, 리눅스와 같은 여러 OS를 모두 지원한다. 그리고 아두이노 보드의 회로도가 CCL에 따라 공개되어 있으므로, 누구나 직접 보드를 직접 만들고 수정할 수 있으며 실제로 수많은 유사(호환)제품이 존재한다. 특히 정품과 동일한 기능의 저가형 중국산 제품들이 애용된다.

아두이노가 인기를 끌면서 이를 비즈니스에 활용하는 기업들도 늘어나고 있다. 장난감 회사 레고는 자사의 로봇 장난감과 아두이노를 활용한 로봇 교육 프로그램을 학생과 성인을 대상으로 북미 지역에서 운영하고 있다. 자동차회사 포드는 아두이노를 이용해 차량용 하드웨어와 소프트웨어를 만들어 차량과 상호작용을 할 수 있는 오픈XC라는 프로그램을 선보이기도 했다 그외 수많은 IT기업들과 코딩 교육 회사들이 아두이노 플랫폼을 이용한 제품을 내놓고 있다.

참고 동영상

- [아두이노 프로젝트들 #1](#)
- [아두이노 프로젝트들 #2](#)
- [아두이노 프로젝트들 #3](#)

1.2 아두이노의 종류

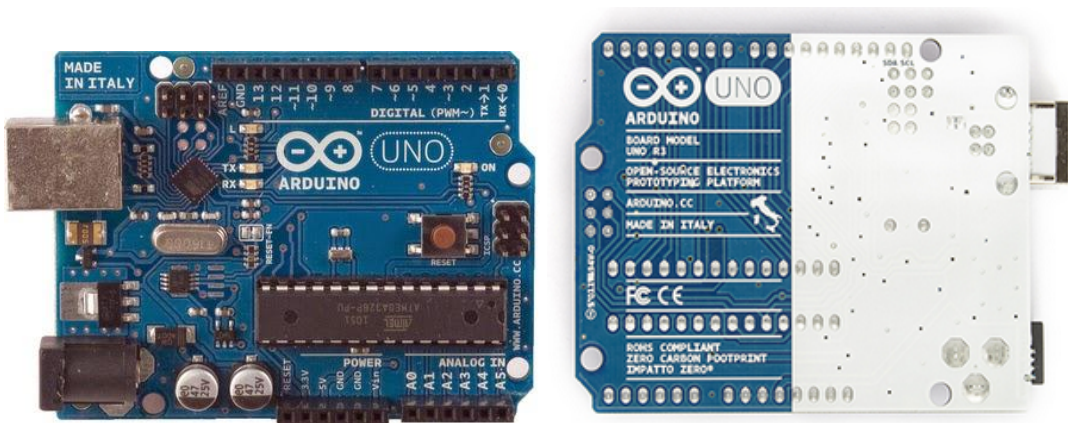
아두이노(arduino)보드는 하드웨어와 소프트웨어 구조가 모두 개방된 오픈소스 플랫폼이기 때문에 정품뿐만 아니라 수많은 변종 보드들이 존재한다. 여기에서는 동일한 Arduino IDE를 사용해서 개발할 수 있고 기본적으로 많이 사용되는 보드들 위주로 설명하도록 하겠다.

1.2.1 우노(uno), 나노(nano) / 프로(pro), 미니(mini), 프로미니(pro mini)

아두이노 우노, 나노, 프로, 미니, 프로미니 다섯 가지 보드는 모두 **atmega328p** 라는 프로세서를 기반으로 한 보드들이므로 코어 부분은 모두 동일하다. 다만 크기를 작게 만들거나 또는 양산을 위해서 **USB통신부가 제거되었거나** 하는 부분이 다를 뿐이다.

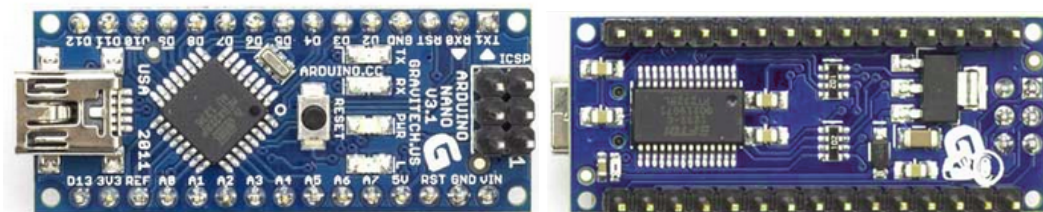
(a) 아두이노 우노와 나노

아두이노 우노는 가장 많이 사용되는 기본적인 아두이노 보드이다. 이 보드의 핀 배열이 거의 표준과 같이 사용된다. 2014년 현재 세 번째 버전인 우노 **R3** 가 유통되고 있다.



[그림 1.2.1] 아두이노 우노

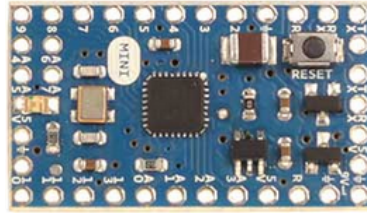
아두이노 나노(nano)는 우노와 거의 동일한 구성을 가지고 있다. 빵판에서 실험할 수 있도록 작은 크기와 핀배열을 가진다.



[그림 1.2.2] 아두이노 나노

(b) 아두이노 프로와 프로 미니

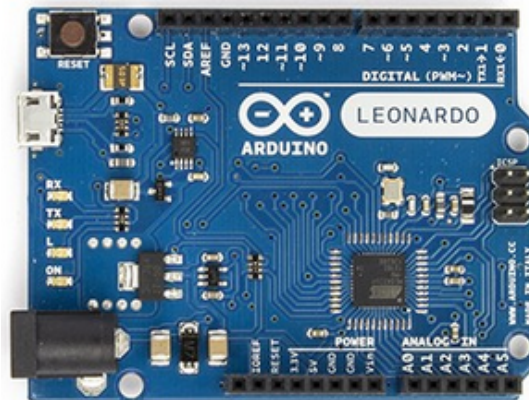
다량의 완성품에 장착하기 용이하도록 소형화시키고 usb-시리얼 변환 칩을 제거한 제품이다. 따라서 프로그래밍을 위한 별도의 usb시리얼 변환기가 필요하다. Atmega328 (혹은 Atmega168) 기반으로 아두이노 우노와 거의 동일한 스펙을 가진다.



[그림 1.2.3] 아두이노 미니 - 프로미니와 핀 배열이 거의 동일함

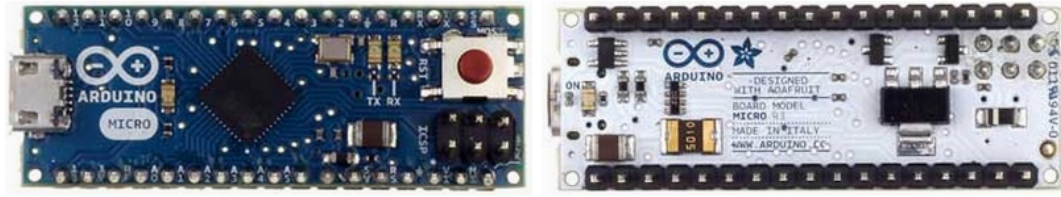
1.2.2 아두이노 레오나르도 (arduino leonardo)와 마이크로(micro)

우노 보드는 usb 통신을 위해서 메인 프로세서와 별도의 칩을 사용하는데 반해서 레오나르도 보드는 USB기능이 내장된 atmega32u4 를 메인 프로세서로 사용한다. 따라서 프로그램 다운로드와 시리얼통신 포트가 독립적으로 동작된다. 우노에 비교해서 장점은 키보드/마우스/조이스틱과 같은 주변기기로 인식시킬 수 있어서 PC와의 인터페이스를 응용한 제품에 대한 활용도가 높다는 것이다.



[그림 1.2.4] 아두이노 레오나르도

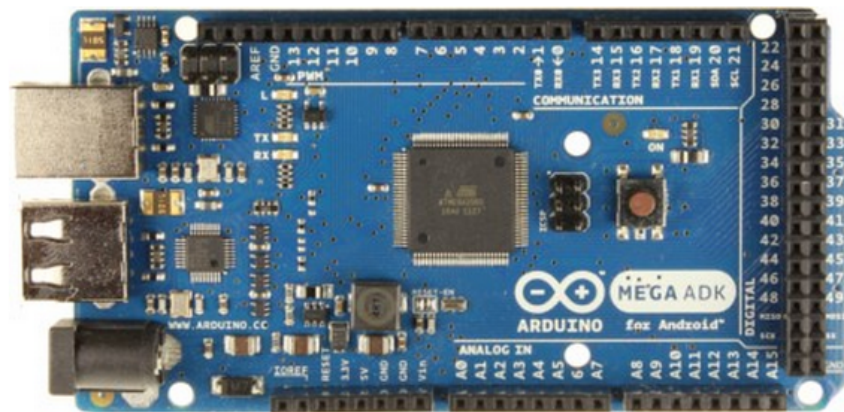
아두이노 마이크로는 레오나르도 보드와 동일한 프로세서를 사용한 소형 보드이다. 빵판이나 기타 완제품을 제작하는데 용이하도록 같은 기능에 크기를 줄여놓은 것이다.



[그림 1.2.5] 아두이노 마이크로 (레오나르도 보드의 소형판)

1.2.3 아두이노 메가 2560 / 메가 ADK

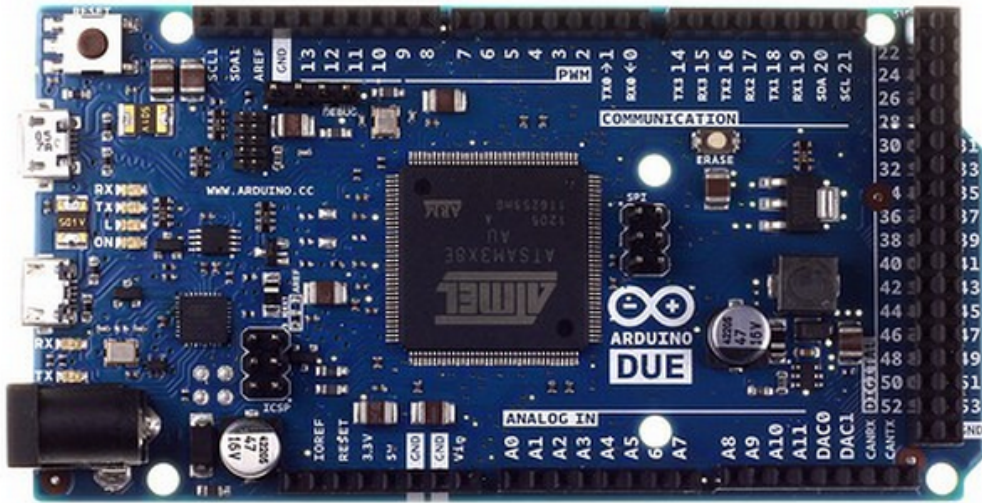
앞서 소개한 보드들의 부족한 핀수와 입출력 기능을 보강한 것이 아두이노 메가 계열의 보드들이며 프로세서로 ATmega2560을 사용한다.



[그림 1.2.6] 아두이노 메가 ADK

1.2.4 아두이노 두에 (arduino due)

위에서 소개한 아두이노 제품들이 8-bit 마이크로컨트롤러인 AVR 기반인 것과 달리 아두이노 두에는 32-bit ARM core 프로세서를 사용한다. 성능이 AVR기반의 아두이노들 보다 높고 핀 수가 매우 많아서 보다 전문적인 제품 개발과 연구 목적으로 사용할 수 있다. 동작 전압이 5V가 아니라 3.3V라는 점도 AVR기반의 아두이노와 다른 점이다.

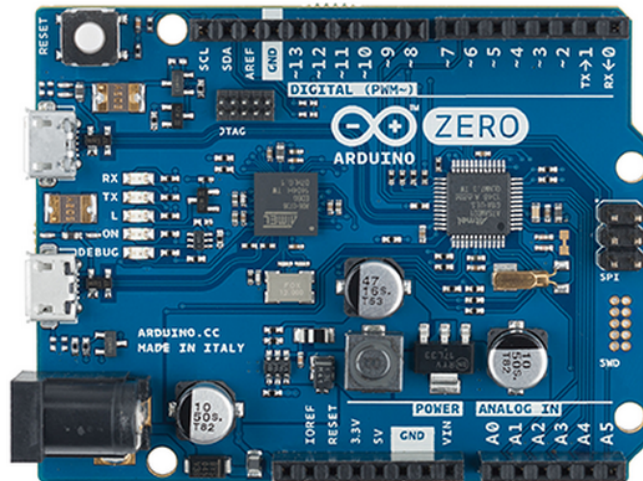


[그림 1.2.7] 아두이노 두에(Due) - ARM프로세서 기반의 아두이노

AVR기반의 아두이노가 세계적으로 호응을 얻으면서 같은 개발 환경을 사용하면서 좀 더 높은 성능을 낼 수 있는 호환 보드들이 다수 출시되고 있으며 아두이노 두에도 그 중 하나이다.

1.2.5 아두이노 제로 (arduino zero)

Atmel사와 아두이노가 동등 개발한 것으로 프로세서는 AVR 이 아니라 ATSAM21 이라는 ARM 계열의 32비트 프로세서를 채용하였다. 48MHz 의 클럭 주파수로 동작하는데 이것만 놓고 보면 아두이노 우노 성능의 3배이다.



[그림 1.2.8] 아두이노 제로

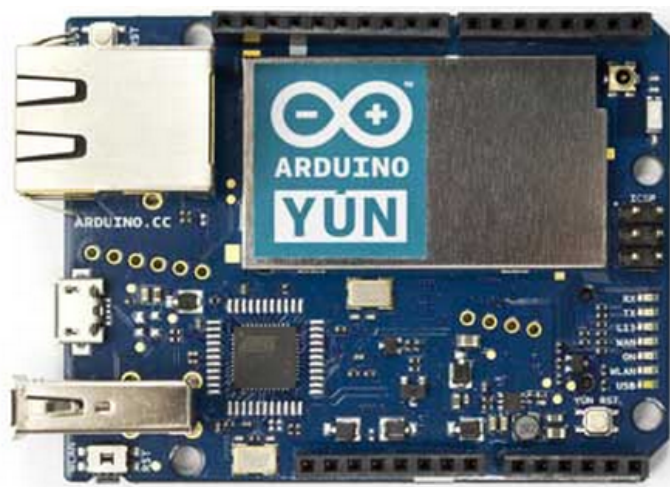
클럭 속도 외에 우노와 비교하여 특이한 점은 0번과 1번을 제외한 모든 디지털 핀에서 PWM 기능을 사용할 수 있다는 점과 플래시메모리가 256KB 로서 우노의 8배 정도로 늘었다는 것이다. 개발 환경에서도 별도의 usb 통신 단자를 통해서 atmel 의 Embedded Debugger

(EDBG) 기능을 사용할 수 있어서 디버깅에 별도의 장치가 필요없다는 점도 눈에 띈다. 우노와의 차이점을 정리하면 다음과 같다.

- 32비트 프로세서 채용
- 3.3V로 동작
- 48MHz의 클럭 주파수 (우노대비 4배)
- 12개의 PWM핀 (우노의 PWM핀은 6개)
- 256KB의 플래시메모리 (우노대비 8배)
- 디버깅(EDBG)을 위한 별도의 usb 단자 내장
- 12비트 ADC (우노는 10비트 ADC임)

1.2.6 아두이노 융 (arduino yun)

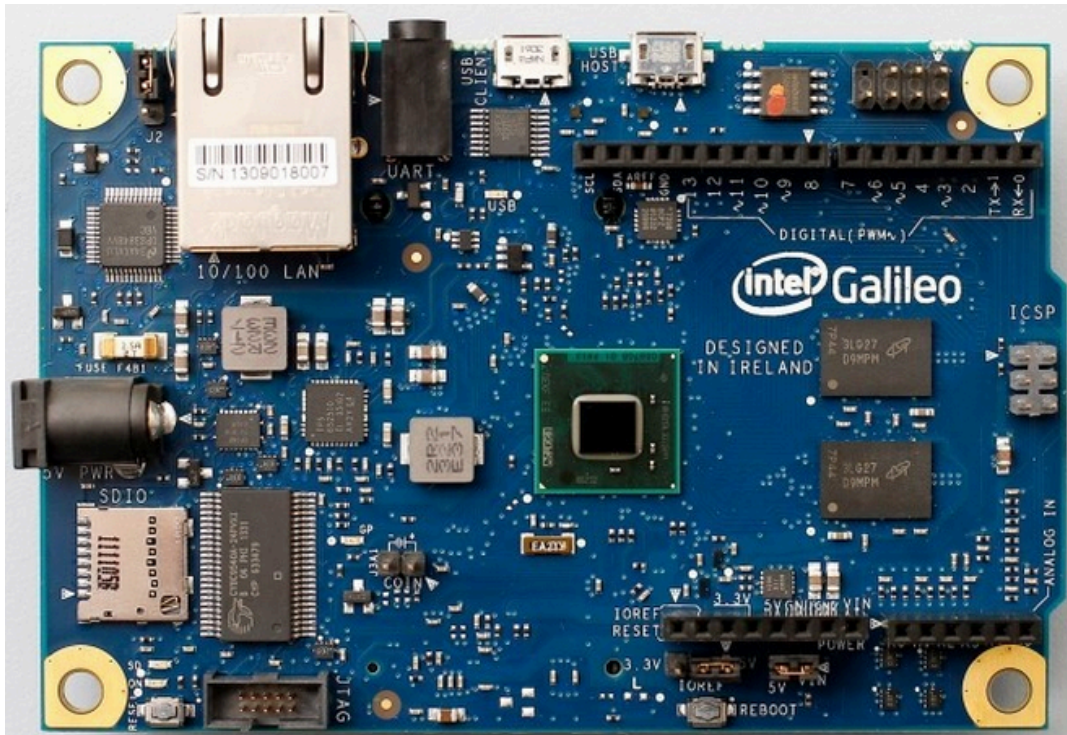
레오나르도 기판과 HTTP 통신과 같은 온갖 텍스트 기반의 작업을 처리하는 Linino(리눅스 변형 OS)를 구동하는 WI-FI 리눅스 기판을 내장하고 있다. 사물인터넷을 구현하기 위한 플랫폼으로서 USB통신뿐만 아니라 와이파이를 통해서도 프로그램이 가능하다.



[그림 1.2.9] 아두이노 융(YUN)

1.2.7 갈릴레오 보드 (galilero board)

인텔의 쿼크(Quark) 프로세서 X1000 를 이용한 고성능의 아두이노 호환 보드이다. 최대 400MHz의 클럭속도를 가지는 초전력 x86 기반의 프로세서이다. 기존 아두이노 우노의 핀배열을 가지며 유사한 개발 환경을 이용할 수 있다. 별도의 갈릴레오보드 전용 IDE가 제공되지만 사용법은 아두이노 IDE 와 동일하다.



[그림 1.2.10] 인텔의 갈릴레오 보드

1.2.8 기타 변종 아두이노들

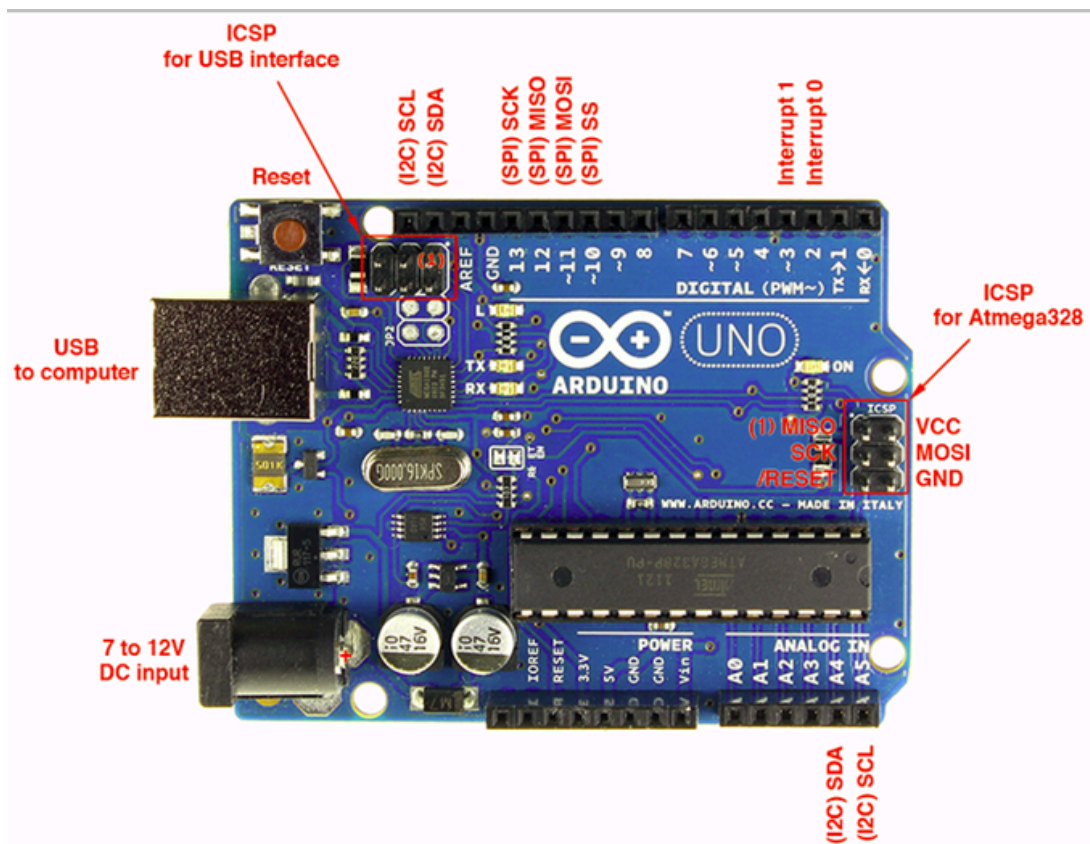
‘아두이노’는 등록된 상표이므로 정품 보드만 이 이름을 사용할 수 있다. 이런 이유로 보통 파생 제품들은 두이노(duino)라는 접미어로 새로운 이름을 만들어서 사용한다. 개중에는 공식 아두이노와 완벽하게 호환되면서 가격은 더 저렴한 것들이 많으므로 이것을 이용해도 같은 개발 도구와 라이브러리로 작업을 할 수 있다. 몇몇 예를 들어보면 다음과 같다.

- Freeduino
- Funduino
- Seeeduino
- Korduino
- Meduino (make block)
- Paperduino
- Boraduino
- Roboduino
- [Femtoduino](#) (가장 작은 아두이노 호환 복제품)
- Ruggeduino (I/O 보호 기능이 내장된 아두이노 보드)
- [pcDuino](#) (갈릴레오보드와 비슷한 개념)
- [Teensyduino](#)

이 외에도 수많은 변종 보드들이 개발되고 있으며 아두이노 코어를 사용하여 편의성을 증대시킨 제품들도 속속 개발되고 있다.

1.3 아두이노 우노 R3

아두이노 우노 R3는 ATmega328P 라는 AVR 8-bit 마이크로컨트롤러를 사용한다. 우노는 이탈리아어로 하나(one)의 의미이며 아두이노 보드들 중 가장 널리 사용되는 사실상 표준 보드라고 할 수 있다. (이탈리아어로 uno, due, tre .. 는 일, 이, 삼 ... 이다.) 이 보드는 PC와 USB로 연결할 수 있으며 이것만으로도 전원(5V) 공급과 프로그램 업로드 그리고 PC와의 시리얼 통신을 수행한다.



[그림 1.3.1] 아두이노 우노 R3 보드

1.3.1 전원 연결

아두이노 우노 보드는 내부적으로 5V 전압으로 동작한다. 아두이노를 동작시키기 위한 전원은 다음과 같이 두 가지 방법 중 한 가지로 인가하면 된다.

- USB로부터 5V 전원을 공급받아서 동작할 수 있다. 따라서 이 보드를 PC와 USB로 연결하면 개발할 수 있는 세팅은 끝난 것이다.
- 외부 전원을 연결하는 단자가 있는데 이것으로 7~12V 사이의 전원을 인가할 수 있고 권장 전압은 9V이다. 일반적인 AA 혹은 AAA 사이즈의 1.5V 건전지를 6개를

직렬 연결하거나 9V 전압의 건전지 혹은, 혹은 1.2V 충전지를 연결해서 사용해도 된다. 물론 AC어댑터도 전압 범위가 맞으면 사용 가능하다.

만약 USB와 전원 소켓에 둘 다 연결되어 있다면 소켓에서 공급되는 전원을 자동으로 사용하게 된다.

1.3.2 디지털 입출력 핀 14개 (0번~13번 핀)

디지털(digital) 입출력 핀들을 이용해서 외부의 디지털 신호를 읽어들이거나 또는 디지털 신호를 내보낼 수 있다.

- 디지털 신호(digital signal) - 0/1 과 같이 상태값이 두 가지만을 가지는 신호

라는 의미이다. 이 디지털 핀을 이용해서 LED를 켜고/끄거나 외부의 스위치가 눌러져 있는지 아닌지 등을 검출할 수 있다. 구체적으로 0V 와 5V 두 전압중 하나의 값을 가지며 이것은 프로그램으로 제어할 수 있다.

디지털 입출력으로 사용되면서 또한 부가적인 기능을 가지는 핀들은 다음과 같다.

- 0번과 1번 핀은 시리얼 통신에 사용된다. USB로 PC와 통신을 할 수 있다.
- 2번과 3번 핀은 인터럽트 기능을 갖는다.
- 3, 5, 6, 9, 10, 11번 핀은 PWM 기능을 가지며 아날로그 출력을 흉내낼 수 있다.

1.3.3 아날로그 입력 핀 6개 (A0~A5)

아날로그 입력 핀이란 외부의 아날로그 입력값을 읽어들이는 핀으로서 주로 센서(sensor)와 연결하여 사용된다. 아날로그(analog)신호는 디지털 신호와는 달리 연속값을 의미하면 예를 들어서 온도, 빛의 세기 등이 있다. 이러한 물리량을 센서가 전기 신호로 변환하며 이것을 이 아날로그 핀으로 읽어들이 수 있다. 센서를 통해 읽은 전압값은 0에서 1023 사이의 숫자로 변환된다. 기준 전압은 5V 이지만 1.1V의 내부 전압이 사용될 수 있으며 AREF핀으로 기준 전압을 직접 인가할 수도 있다. 그리고 아날로그 핀은 디지털 입/출력 핀으로도 사용할 수 있다.

1.3.4 아날로그 출력핀 6개 (3,5,6,9,10,11번핀)

디지털 출력핀이 0V/5V 두 가지 값만을 가질수 있는데 비해서 아날로그 출력핀은 0V~5V사이의 전압 값(256단계)을 가질 수 있다. 엄밀히 얘기하면 PWM방식으로 동작하므로 순수 아날로그 방식은 아니다.

1.3.5 인터럽트 (2, 3번 핀)

2번 핀과 3번 핀은 인터럽트(interrupt) 기능을 가진다. 인터럽트 처리(interrupt handling)라는 것은 이벤트를 처리하는 데 사용되는 기능으로서 특정한 신호가 발생했을 때 정해진 동작을 수행하여야 하는 경우 사용되는 방식이다. 예를 들어서 버튼이 눌러진 시점에서 (또는 떼어진 시점에서) 어떤 작업을 수행해야 하는 경우다.

1.3.6 그 외의 기능들

아두이노 우노 보드는 3.3V의 전압도 공급할 수 있다. 이는 USB만 연결한 경우도 마찬가지이다. AREF 핀은 아날로그 핀의 기준 전압을 설정하는 용도로 사용된다.

다른 기기와의 통신 기능은 다음과 같다.

- 시리얼 통신 : 0번, 1번 핀
- SPI 통신 : ICSP 헤더핀
- TWI (I2C) 통신 : A0, A1 핀

시리얼 통신 방식 외에는 일반적으로 사용 빈도가 낮지만 기기간 통신이나 일대다 통신을 하는 경우 SPI, TWI 통신이 널리 사용된다.

1.4 개발 환경

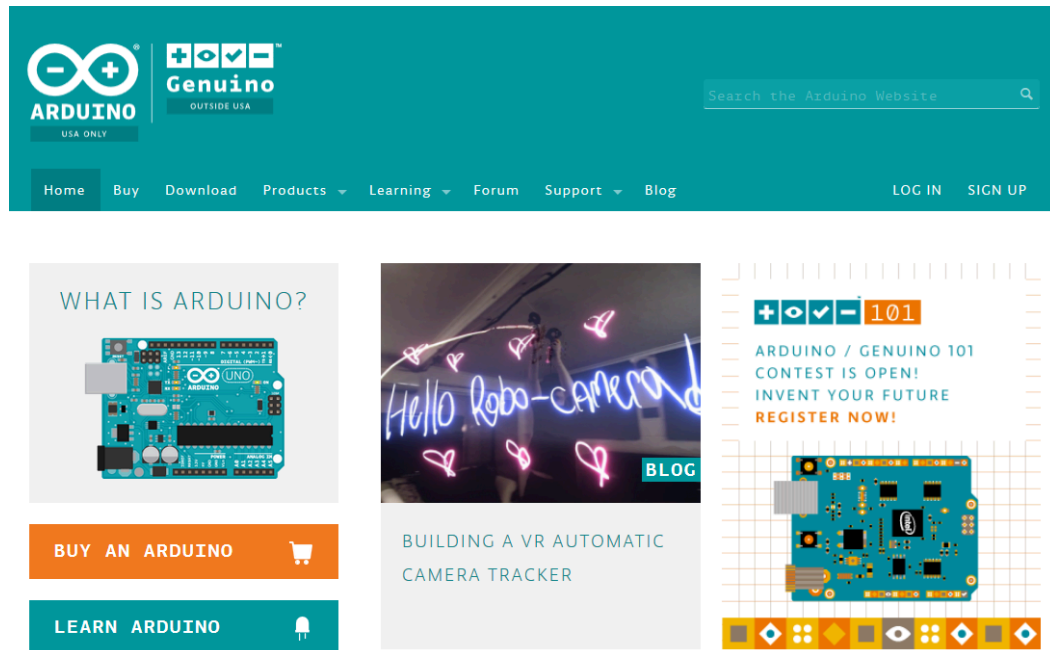
아두이노를 프로그래밍 하는데 [C++언어](#)가 사용된다. 만약 사용자가 C++언어에 익숙하다면 물론 좋겠지만 그렇지 않더라도 라이브러리가 사용하기 편하게 잘 갖추어져 있으므로 익히는데 큰 어려움은 없을 것이라 생각된다. 사실 사용하는 언어는 C++ 이지만 이것에 대해서 잘 모른다고 미리 겁먹을 필요는 없다. 어차피 아두이노라는 플랫폼 자체가 비전공자(디자이너, 예술가 등)들이 깊은 전공 지식 없이 개발을 할 수 있도록 설계가 된 것이기 때문이고 대중적인 인기를 얻은 이유 중 하나가 학습의 난이도가 낮다는 것이기 때문이다.

1.4.1 아두이노 IDE

프로그램을 작성/편집 하고 컴파일/디버깅 할 수 있는 통합 환경을 IDE(Integrated Development Environment)라고 한다.

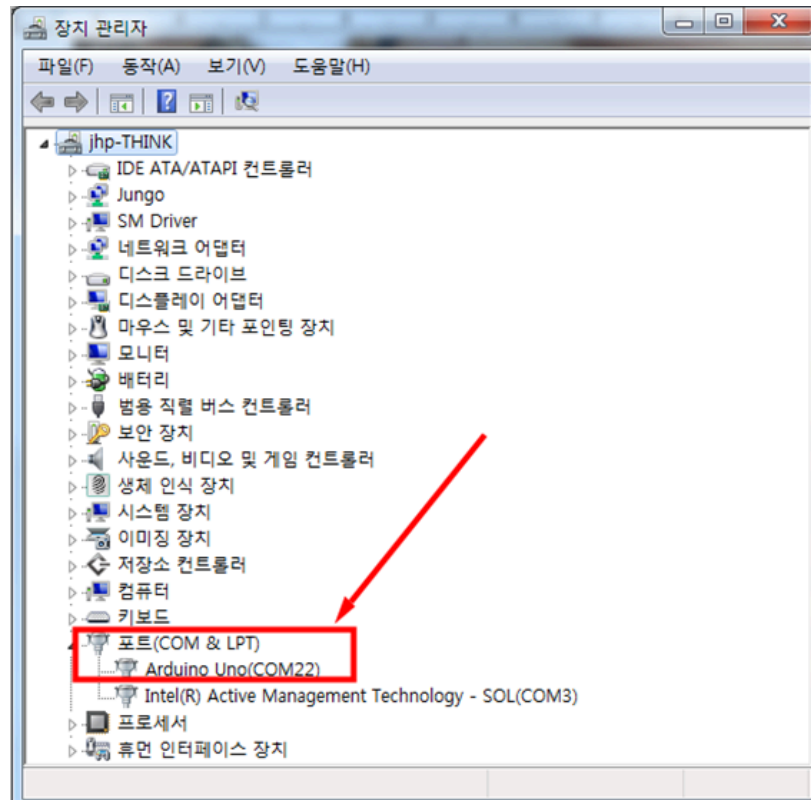
- IDE (Integrated Development Environment): 통합 개발 환경

아두이노는 **processing** 이라는 오픈소스 언어 환경(이것도 굉장히 흥미로운 프로젝트이다. processing.org 홈페이지에 자세한 내용이 있다.)을 이용하여 개발된 전용 **IDE**를 무료로 제공하고 있어서 편리하게 사용할 수 있다. 아래의 공식 다운로드 링크에서 **OS**에 맞는 프로그램을 다운받아서 설치한다. 프로그램 설치 과정은 간단하므로 설명은 생략한다. (로컬 PC에 설치가 번거롭다면 온라인 **IDE**를 사용하는 것도 고려해 볼 수 있다.)



[그림 1.4.1] 아두이노의 공식 홈페이지 arduino.cc

아두이노 **IDE**를 실행하기 전에 아두이노를 **USB**로 **PC**와 연결하는 것이 좋다. 아두이노 장치는 가상 시리얼기기로 인식이 되기 때문에 연결을 한 후에 시리얼포트 번호를 확인해야 한다. 윈도우즈의 경우 제어판>시스템 및 보안>시스템>장치관리자 로 찾아들어간 후 포트(COM&LPT) 항목을 보면 확인할 수 있다.



[그림 1.4.2] 윈도우즈 장치관리자에서 아두이노 포트 확인

이제 아두이노IDE를 실행시키면 <그림 1.4.3>와 같은 조금은 단순해 보이는 화면이 뜰 것이다. 이 프로그램을 이용하여 아두이노 프로그램을 입력하고 컴파일한 후 (USB로 아두이노 보드와 연결되어 있다면) 다운로드까지 수행할 수 있다. 이 IDE는 사실 **Processing**이라는 오픈소스 프로젝트에 기반한 것이다. 아두이노에서 취득한 데이터를 시각화하는데 많이 사용되는 플랫폼이니 관심이 있다면 홈페이지(processing.org)를 방문해서 관련 정보를 취득해 보는 것도 좋을 것이다.

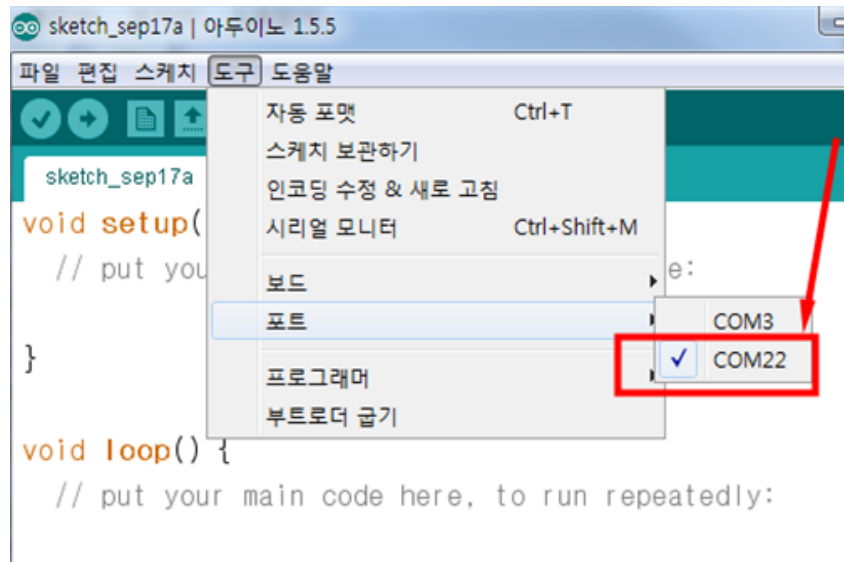


[그림 1.4.3] 아두이노 IDE의 외형

프로그램을 입력하기 앞서서 먼저 해야 할 것은 보드와 포트를 설정하는 것이다. 어떤 아두이노 보드를 사용하는지 그리고 연결된 포트의 번호는 무엇인지를 정확히 설정해야 정상적으로 프로그램을 다운로드하거나 시리얼통신을 수행할 수 있다.

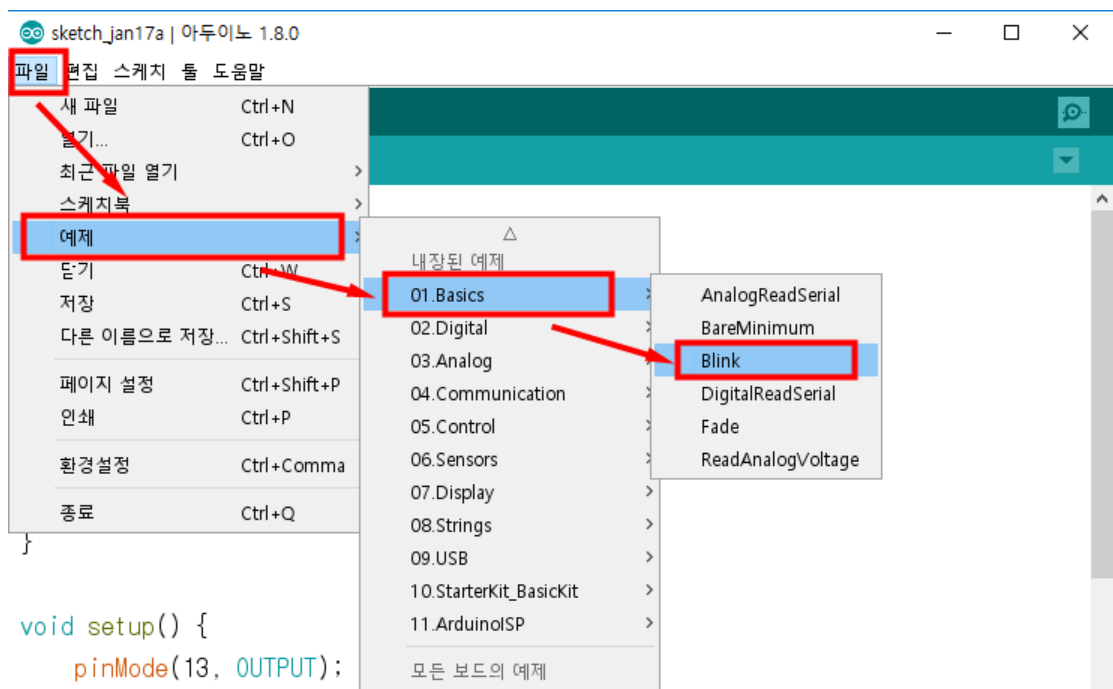
- (1) 도구>보드 항목 : 연결된 보드의 종류를 선택
- (2) 도구>포트 항목 : 가상 시리얼 포트의 번호를 선택

위의 두 항목만 올바르게 선택했다면 일단 개발 환경은 다 갖춘 셈이다. 만약 아두이노가 PC에 연결되어 있다면 포트는 자동으로 설정되어 있을 것이다.



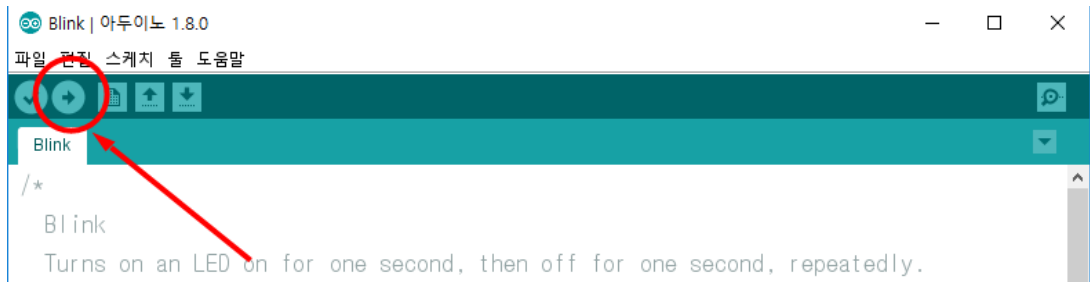
[그림 1.4.4] 포트 번호 설정하는 메뉴

이제 다음과 같이 내장된 예제를 열어보도록 하자.



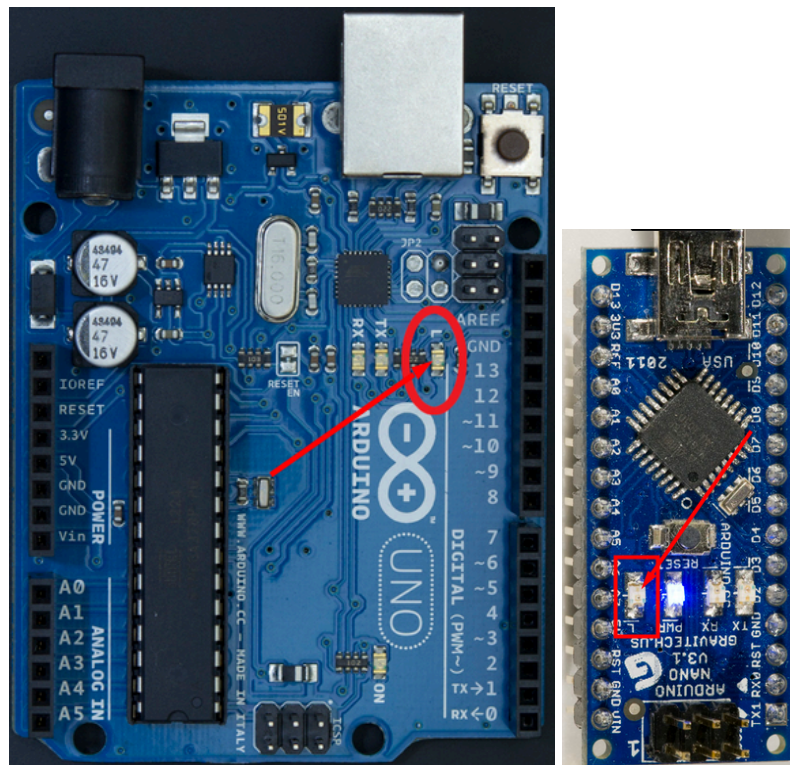
[그림 1.4.5] 내장 LED를 깜박이는 blink 예제의 선택

그러면 새로운 창에 C++ 언어로 작성된 blink 예제가 열릴 것이다. 프로그램에 대한 세부적인 사항들은 다음에 설명하도록 하고 일단 이것을 다음 그림의 버튼을 클릭하여 연결된 보드에 업로드(upload) 해보자.



[그림 1.4.6] 업로드(upload) 버튼

그러면 보드에 내장된 LED가 깜박이는 것을 확인할 수 있다.

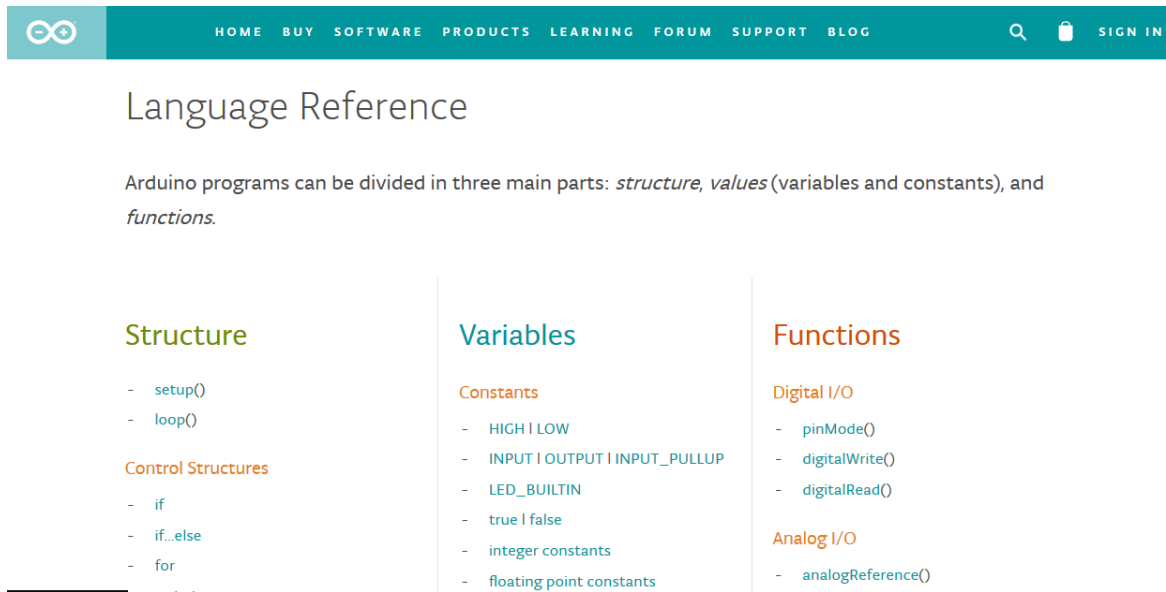


[그림 1.4.7] 아두이노 우노(좌측)와 나노(우측)의 내장 LED

이것이 잘 동작된다면 실습 환경과 테스트가 완료되었다고 볼 수 있다.

1.5 아두이노 라이브러리

아두이노 라이브러리(library)는 다양한 함수와 클래스(class)를 제공하여 사용자가 원하는 기능을 구현하기 위해서 편리하게 이용할 수 있다. arduino.cc 의 **reference** 메뉴에 들어가면 기본적인 함수들이 자세하게 설명되어 있다.



[그림 1.5.1] arduino.cc > LEARNING > REFERENCE

이러한 기본적인 함수들 외에도 전세계의 수많은 개발자들이 만들어서 배포하는 라이브러리들도 풍부하기 때문에 웬만한 기능은 사용자가 직접 작성할 필요가 없고 원하는 기능을 갖는 라이브러리를 자유롭게 다운로드 받아서 활용할 수 있다.