

# 임베디드 이해하기

임베디드란 무엇인가?

컴퓨터의 작동원리

소프트웨어 용어

마무리 학습

참조



## | 임베디드 이해하기

## | 임베디드란 무엇인가?

임베디드는 아주 작은 형태의 전자회로를 가지고 있는 컴퓨터입니다. 따라서 임베디드는 다양한 사물에 숨겨져 있습니다. 임베디드의 뜻이 우리말로 번역하면 “내장된”이라고 할 수 있습니다. 우리 주위에서도 임베디드 장비를 쉽게 찾아볼 수 있습니다. 대표적인 것이 신호등, 지하철 개찰구 등을 들 수 있습니다.

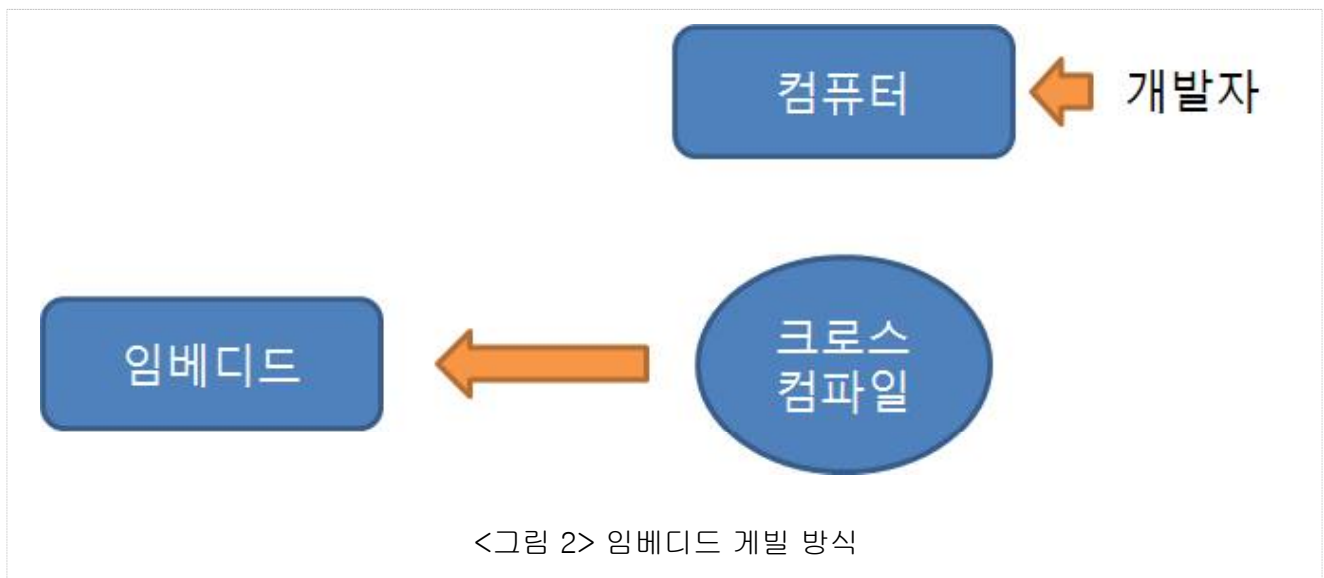


<그림 1> 임베디드 사용 용역

임베디드는 매우 다양한 영역에서 사용되고 있습니다. 집에서 사용하는 냉장고나 선풍기도 임베디드를 사용합니다. 스마트폰도 임베디드 장치라 할 수 있는데, 내부에 다양한 센서와 액추에이터 그리고 초소형 고성능 컴퓨터를 탑재하고 있습니다. 수백개에서 수천개의 전자회로를 가지고 있는 자동차도 대표적인 임베디드 장치입니다.

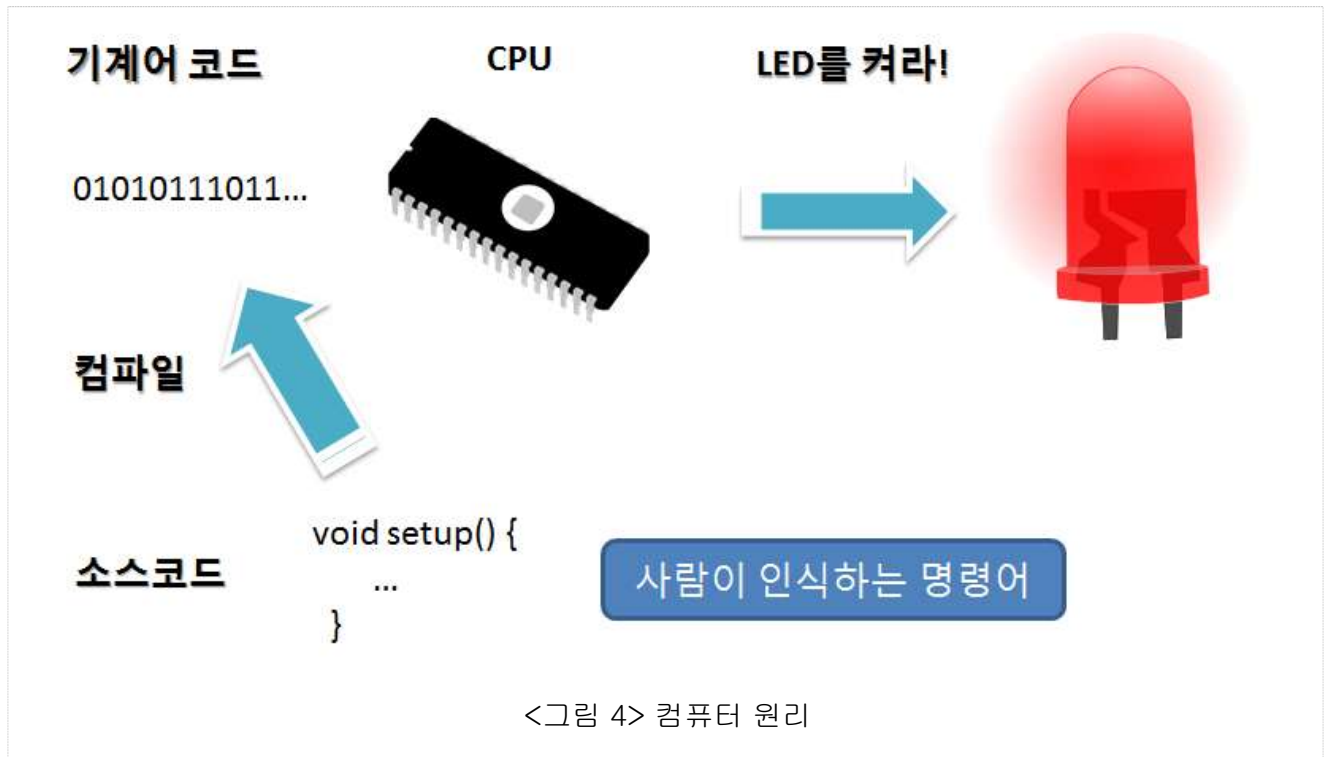
### ▪ 임베디드 개발 방식

아두이노를 이용해 다양한 프로젝트를 하는 것도 일종의 임베디드 개발이라 할 수 있습니다. 임베디드는 기존 컴퓨터와는 달리 키보드와 모니터 같은 입력이나 출력장치가 없는 것이 일반적입니다. 따라서 임베디드를 개발하기 위해서는 다른 컴퓨터에 연결해서 개발해야 합니다.



대개 컴퓨터 시스템과 임베디드가 서로 호환이 되지 않기 때문에 크로스 컴파일이라는 것을 이용합니다. 예로 인텔의 x86 윈도우 컴퓨터에서 아두이노가 칩을 사용하는 아두이노에서 사용할 프로그램을 컴파일하는 것도 이에 해당됩니다.

## | 컴퓨터의 작동 원리

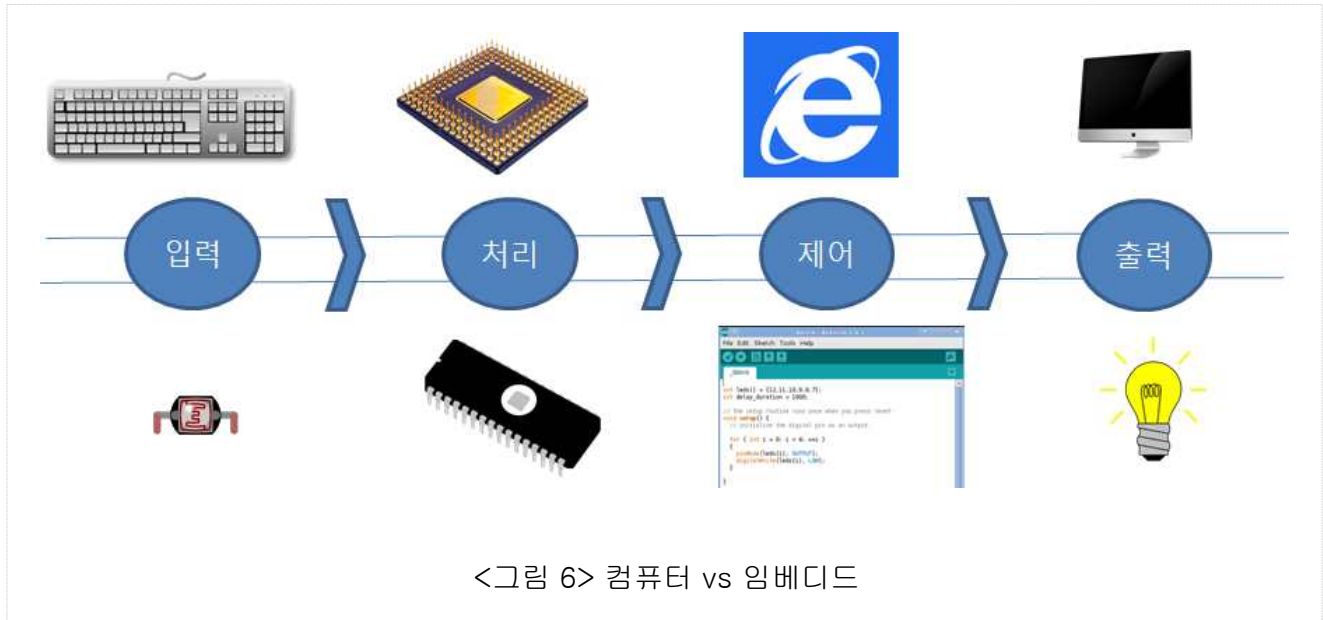


먼저 아두이노에서 `setup`, `loop`와 같이 사람이 인식할 수 있는 말로 작성한 것을 소스 코드라고 합니다. 이것을 다시 컴퓨터와 같은 기계가 알아들을 수 있는 말, 즉 기계어 코드로 변환하는 것을 컴파일이라고 말합니다. 기계는 실제로는 0과 1만 인식할 수 있기 때문에 기계어 코드는 0과 1이 조합되어서 만들어집니다. 이렇게 기계어 코드가 생성되면 다시 CPU가 해석해서 동작하게 됩니다.



앞의 컴퓨터 원리를 다시 살펴보면 소프트웨어와 하드웨어로 구분할 수 있습니다. 소스 코드를 작성하고 기계어 코드를 작성하는 것까지가 소프트웨어 영역이라 할 수 있습니다. 여기서 기계어는 사용되는 하드웨어의 종류에 따라 다르게 생성됩니다. 예로 인텔 CPU와 ARM CPU에서 사용하는 기계어가 다른 것을 들 수 있습니다. 컴파일러는 이와 같은 하드웨어의 차이에 따라 맞는 기계어를 생성해줍니다.

하드웨어 영역으로 넘어오면 CPU에서 기계어를 해석해서 반도체와 같은 부품을 이용해 물리적인 현상을 통해서 실제 결과물을 실행하게 됩니다. 여러분이 배운 아두이노의 경우 결과물은 센싱이나 액추에이터를 통한 제어라 할 수 있습니다.



컴퓨터와 임베디드는 컴퓨터 원리 측면에서는 동일하게 작동합니다. 하지만 각 과정을 자세히 비교하면 서로 다른 특징을 가지고 있습니다.

첫번째 컴퓨터는 키보드나 마우스와 같은 입력장치를 이용해 사용자의 명령을 받는다면, 임베디드는 다양한 센서를 통해 정보를 입력받습니다. 두번째 컴퓨터는 32비트, 64비트와 같은 고성능 CPU를 이용한다면, 임베디드는 저전력 저비용 CPU가 많이 사용됩니다. 예로 아두이노의 아트메가 칩이 8비트 CPU를 사용하는 것도 이와 같습니다. 세번째, 컴퓨터는 고성능 처리장치와 자원을 이용해 운영체제를 돌려 다양한 응용프로그램을 실행할 수 있습니다. 반면 임베디드는 자원이 매우 제한적이어서 대개 특정 기능만 담당하는 펌웨어 하나만 돌아가는 경우가 많습니다. 네번째 컴퓨터는 모니터, 스피커와 같은 정형화된 주변장치를 이용해 출력을 합니다. 이에 비해 임베디드는 다양한 종류의 액추에이터를 이용해 출력을 합니다.



<그림 7> 컴퓨터의 발전

컴퓨터는 지금도 아주 빠른 속도로 발전하고 있습니다. 초기 컴퓨터였던 에니악에 경우에는 0과 1에 대한 계산을 하기 위해 사람이 일일이 전선을 바꿔 연결해주어야 했습니다. 또한 방에 가득찰 정도로 크기도 컸습니다. 하지만 성능은 현재 컴퓨터에 비교하면 아주 나빴습니다. 주로 포탄의 방위각등을 재기위한 군사용으로 사용이 되었습니다. 이런 컴퓨터가 일반인들에게 보급될 수 있는 계기를 만들어 준 것이 애플 컴퓨터입니다. 그 이후 1990년대를 지나면서 급속도로 컴퓨터의 보급이 이루어졌습니다. 현재의 스마트폰도 컴퓨터가 발전되어 만들어진 결과라 할 수 있습니다. 이와 같은 컴퓨터는 사용된지 100년이 거의 안 되지만 이제는 사람들의 삶에 중요한 영향을 준다고 할 수 있습니다.

## | 소프트웨어 용어

### ▪ 오픈소스 소프트웨어

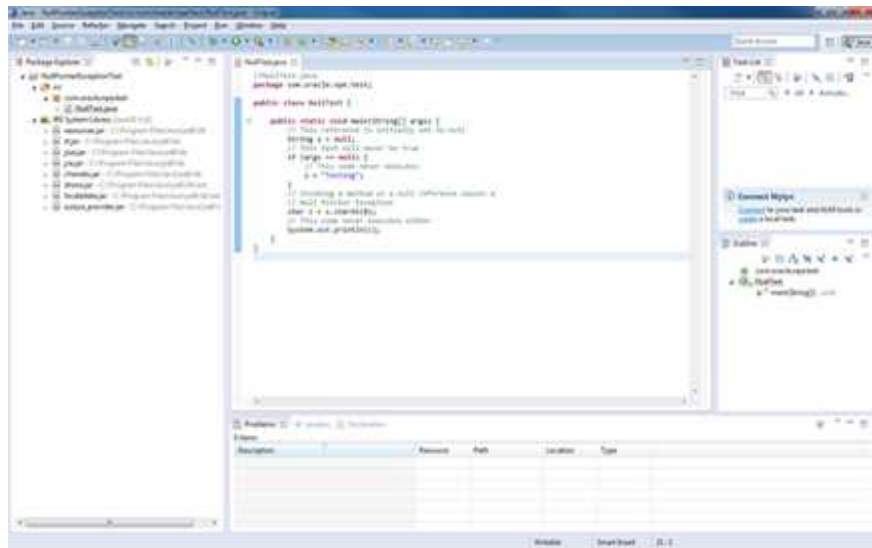
일반적으로 소프트웨어는 소스 코드를 공개하지 않고 실제 결과물만 판매했습니다. 대표적인 회사가 마이크로소프트웨어와 같은 회사를 들 수 있습니다. 1980년대 후반에 들면서 폐쇄적인 소프트웨어 문화가 소프트웨어 사업 발전을 저해한다는 말이 나오면서 오픈소스 소프트웨어 운동이 일어나게 됩니다. 이 운동의 일환은 모든 사람에게 소스 코드를 공개하고 공유함으로써 함께 더 좋은 소프트웨어를 만드는 것입니다. 그리고 이와 관련해 GNU와 같은 오픈소스 소프트웨어 재단이 출범하기도 했습니다. 오픈소스 운영체제로 유명한 리눅스가 GNU에서 만들어졌습니다. 또한 모질라 재단에서 만들어진 오픈소스 웹 브라우저인 파이어 폭스와 같은 것도 있습니다. 현재에 와서는 오픈소스 프로그램이 상용 프로그램보다 더 좋은 경우를 많이 볼 수 있습니다.

### ▪ 오픈소스 하드웨어

2002년 오픈소스 소프트웨어가 있지만 하드웨어 영역은 폐쇄적인 것을 들어 오픈소스 하드웨어라는 말이 나오기 시작합니다. 그 중 가장 대표적인 오픈소스 하드웨어가 여러분이 배우는 아두이노입니다. 아두이노는 소프트웨어 뿐만 아니라 하드웨어 회로도까지 모두 공개해서 누구나 쉽게 아두이노를 만들 수 있도록 했습니다.



### ■ 통합개발환경

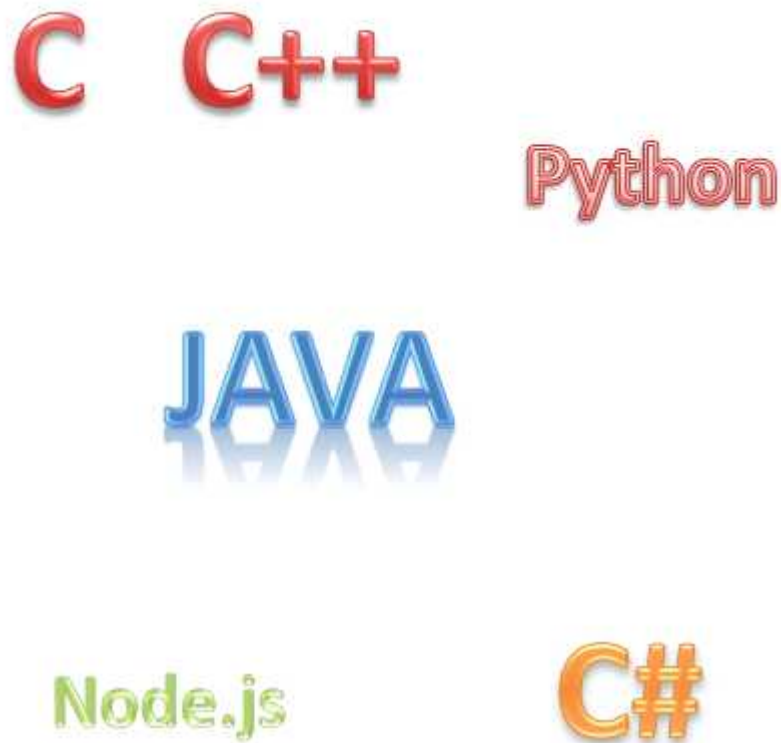


## 코드 편집 + 컴파일 + 디버깅

<그림 8> 통합개발환경

통합개발환경(IDE : Integrated Development Environment)은 코드 편집 기능, 컴파일 기능 그리고 디버깅 기능을 가지고 있습니다. 여러분이 아두이노 코드를 작성하는 프로그램도 통합개발환경입니다. 대표적인 통합개발환경으로 MS의 비주얼 스튜디오, 오픈소스 재단에서 주로 사용하는 이클립스등을 들 수 있습니다.

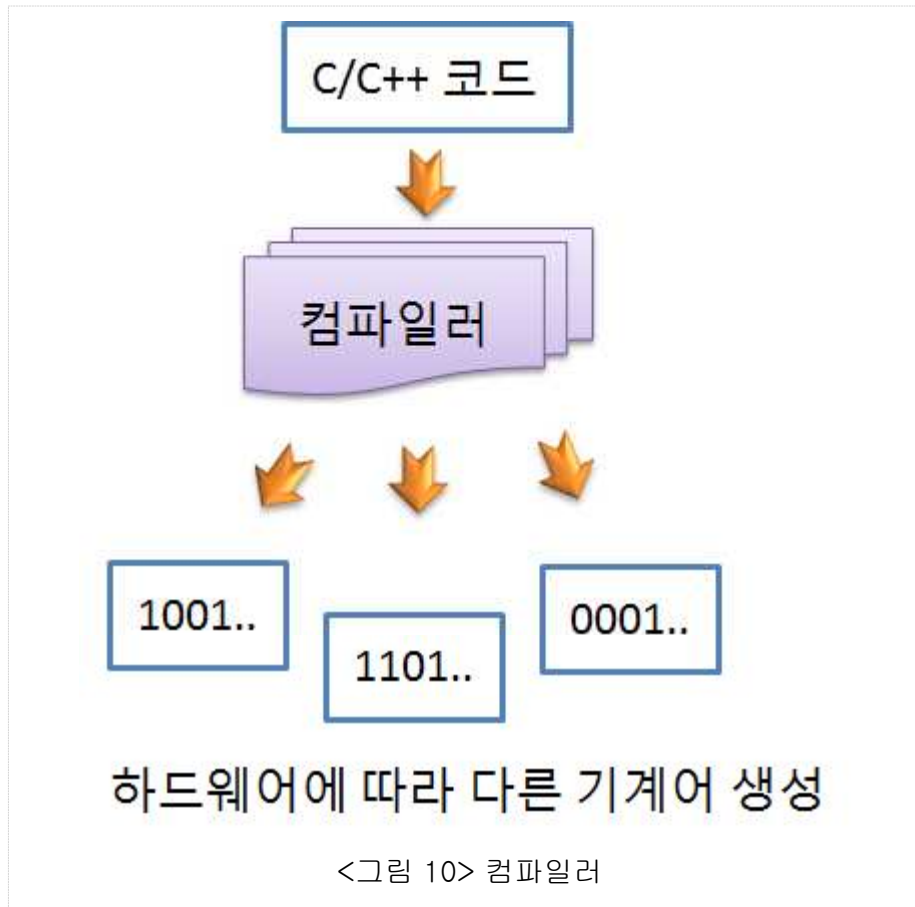
## ▪ 프로그램 언어



&lt;그림 9&gt; 프로그램 언어

사람의 언어가 다양하듯이 프로그램 언어도 다양한 종류를 가지고 있습니다. C, C++의 경우 임베디드 시스템 개발에 많이 사용됩니다. JAVA는 웹 프로그램, 안드로이드등에서 사용되고 있습니다. 파이썬은 웹 서비스에서 많이 사용되며, 아두이노와 같은 오픈소스 하드웨어인 라즈베리 파이에서 주로 사용되는 언어이기도 합니다. 노드JS는 웹 서비스와 관련된 언어로 최근 많은 인기를 얻고 있습니다. 이 노드JS도 비글본 블랙이라는 오픈소스 하드웨어에서 개발 언어로 사용하고 있습니다.

### ▪ 컴파일러

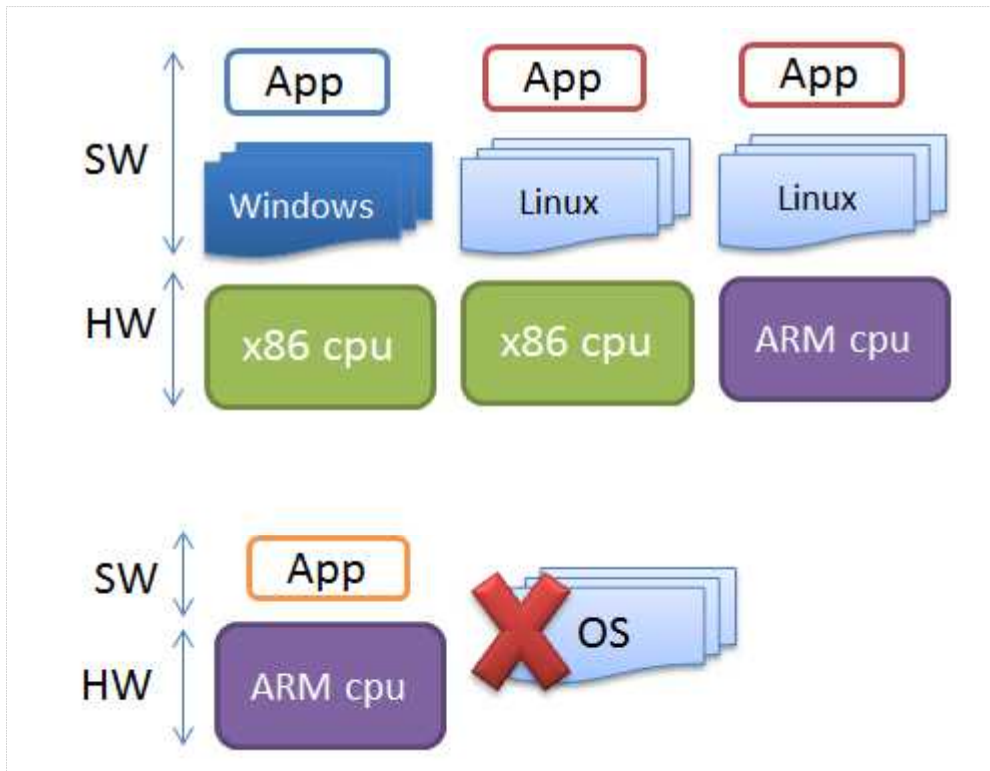


앞서 다양한 프로그램 언어가 있는 것을 보았는데, 이 프로그램 언어는 결과적으로 동일한 구조의 기계어로 만들어집니다. 그리고 이렇게 기계어로 변환하는 것을 바로 컴파일러라고 부릅니다. 컴파일러는 하드웨어의 종류에 따라 다양한 종류의 기계어를 생성합니다.



컴파일러와는 달리 특정 하드웨어를 똑같이 사용하면서 웹 브라우저에서 가볍게 실행되면서 기계어로 변환되는 것을 인터프리터라고 합니다. 그리고 컴파일과 인터프리터가 혼합된 버추얼 머신 같은 것이 있습니다. 닷넷 VM, JAVA등이 VM에서 돌아가는 대표적인 예입니다.

▪ 운영체제, 멀티플랫폼, 펌웨어

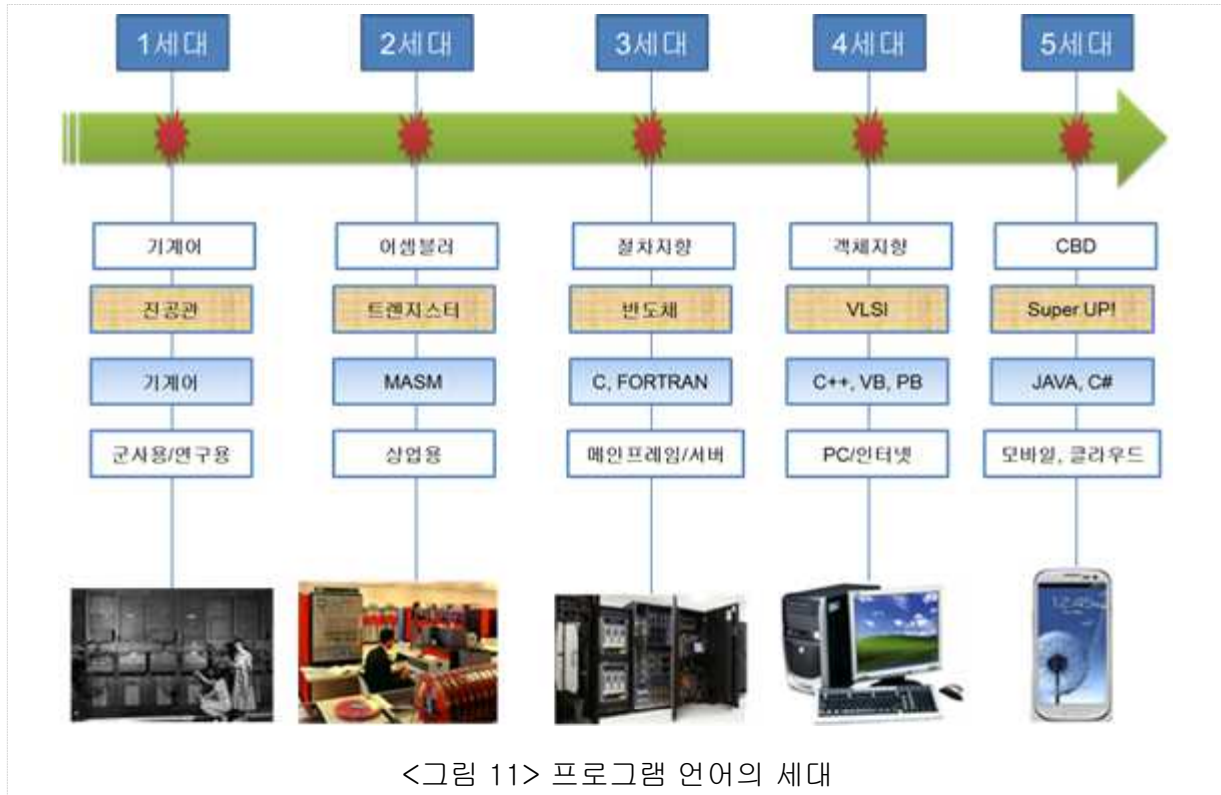


운영체제는 하드웨어와 소프트웨어에 사이에 있는 층과 같습니다. 운영체제의 대표적인 예가 윈도우, 리눅스, 맥 OS 등이 있습니다. 운영체제는 컴퓨터에 연결된 다양한 하드웨어를 제어할 수 있습니다. 이를 통해 운영체제 위에서 돌아가는 응용프로그램에서 쉽게 하드웨어 자원을 사용할 수 있도록 해줍니다.

멀티플랫폼은 하드웨어가 달라도 똑같은 프로그램이 돌아가는 것을 뜻합니다. 윈도우는 일반적으로 x86 CPU에서만 사용 가능하지만 리눅스는 x86, ARM 모두에서 실행할 수 있습니다.

펌웨어란 한 가지 프로그램만이 돌아가는 것을 말합니다. 여러분이 사용하는 아두이노가 바로 펌웨어 기반으로 되어있는 대표적인 기기라고 할 수 있습니다. 운영체제가 돌아가지 않고 개발자가 작성한 프로그램 한 가지만 실행됩니다.

### ■ 프로그램 언어의 세대



1세대 언어는 0과 1로 이루어진 기계어입니다. 기계어를 사용할 당시는 컴퓨터 기술이 발전하지 않았습니다. 진공관이라는 기계장치를 이용해 이진수를 처리하는 형태로 되어있었습니다. 주로 군사용이나 학술 연구용에만 사용되었습니다.

1960년대에 들어서면서 컴퓨터가 상업용으로 인정을 받으면서 큰 은행이나 정부등에서 사용되기 시작했습니다. 이 당시 트랜지스터가 개발이 되면서 컴퓨터 발전에 큰 영향을 주게 됩니다. 이때는 2세대라 할 수 있는 어셈블리어가 주로 사용되었습니다.

1970년대 3세대 절차지향 언어로 넘어오면서 본격적으로 컴퓨터 소프트웨어라는 말이 나오기 시작했습니다. 이때부터 메인 프레임, 서버등이 나오기 시작하면서 큰 기업을 중심으로 컴퓨터가 보급되기 시작합니다. C, 포트란 같은 언어가 이 당시 만들어졌습니다.

1980년대에 들어 소프트웨어의 사용이 급진적으로 늘어나기 시작했습니다. 컴퓨터가 일반인들에게 보급이 된 것이 큰 이유라 할 수 있습니다. 또한 인터넷의 사용도 큰 영향을 주었습니다. 따라서 소프트웨어 개발 생산성이 중요한 이슈로 부각이 되었고, 그로 인해 4세대 언어라 할 수 있는 객체지향 언어가 나타나기 시작합니다. 대표적인 것이 C++, VB라 할 수 있습니다.

2000년대 넘어오면서 5세대라 할 수 있는 CBD(Component-based development)라는 개념이 등장했습니다. JAVA나 C#이 CBD의 예라 할 수 있는데, 플랫폼을 가리지 않고 실행할 수 있다는 것이 기존 객체지향 언어와 다른 특징이라 할 수 있습니다. 이에 맞춰 컴퓨터의 사용영역도 클라우드, 모바일로 확장이 되기 시작합니다.

### | 도전해보기

우리 생활에 사용되는 임베디드 제품을 찾아봅시다.

소프트웨어와 하드웨어 특성을 정리해봅시다.

컴퓨터, 스마트폰, 아두이노를 비교해봅시다.

## | 참조

위키피디아 컴퓨터

<http://ko.wikipedia.org/wiki/컴퓨터>

위키피디아 소프트웨어

<http://ko.wikipedia.org/wiki/소프트웨어>

위키피디아 임베디드 시스템

[http://ko.wikipedia.org/wiki/임베디드 시스템](http://ko.wikipedia.org/wiki/임베디드_시스템)