

# 코딩키트

블록 코딩에서 사물인터넷 코딩까지  
쉽고 재미있게 배울 수 있어요.

블록 코딩 / 아두이노 코딩 / 라즈베리파이 코딩 / 사물인터넷 코딩

코딩북1 - 코딩키트 소개



Introduction of  
Codingkit

**SISODREAM**  
Silicon to Software

(주)시소드림 SISODREAM Inc.



- 본 문서의 저작권 -

본 문서에 대한 모든 저작권은 (주)시소드림에 있습니다. 본 문서는 자유롭게 배포할 수 있습니다. 단, 상업적인 목적으로 이용을 하거나 판매를 할 수는 없습니다. 또한 문서를 수정하여 배포할 수 없으며 인용할 경우 출처를 밝혀주시고, 인용한 부분이 1 페이지 이상 혹은 5 곳 이상일 경우에는 본 문서의 저작권자인 (주)시소드림에 허가를 득해야 합니다. 본 문서를 인쇄하여 누적수량 5 부 이상 배포할 경우에도 (주)시소드림의 허가를 득해야 합니다. 본 문서에 대한 이러한 사항이 지켜지지 않을 경우 민형사상의 책임을 물을 수 있습니다.

발행일 : 2015 년 8 월 28 일 (초판), 2017 년 3 월 8 일 (개정판)

발행처 : (주)시소드림

연락처 : [ck@sisodream.com](mailto:ck@sisodream.com) / 031-757-7755

## [ 코딩? 컴퓨터에게 쓰는 작업 지시서! ]

코딩(Coding)이란? 컴퓨터에 일을 시키기 위한 작업 지시서를 쓰는 것과 같습니다. 사람과 사람 사이에 의사 소통을 하기 위해서는 한국어, 영어, 중국어와 같은 언어라는 것이 필요합니다. 사람과 컴퓨터가 의사 소통을 하기 위해서도 이와 비슷한 언어가 필요합니다. 이러한 언어들로는 C-언어, 자바(JAVA), 파이썬(Python) 등이 있지요. 이러한 언어들로 컴퓨터에 어떤 일을 시키기 위해서는 각 언어의 문법에 맞게 작업 지시서 같은 것을 작성하여 컴퓨터에게 전달해 주어야 합니다. 이렇게 컴퓨터의 언어로 작업 지시서를 쓰는 것을 코딩이라고 합니다. 다른 말로 프로그래밍이라고도 하지요. 그리고 이 작업 지시서를 코드라고 부릅니다.

사람과 다르게 컴퓨터는 문법적으로 한치의 오차도 없는 완벽하게 정확한 코드를 원합니다. 그래서 코드를 컴퓨터에게 전달하기 전에 문법 검사 등을 합니다. 이 과정에서 잘못된 부분이 생기면 다시 검토하여 수정하고 다시 문법 검사를 합니다. 문법 검사에서 이상이 없으면 정말로 컴퓨터만 알 수 있고, 사람은 알 수 없는 숫자들로 코드를 바꾸어 줍니다. 이런 숫자 코드들을 컴퓨터에게 전달하여 컴퓨터가 어떠한 일을 하게 합니다. 코드가 문법적으로 완벽하다고 해도 정확한 작업 지시서가 될 수는 없습니다. 그 작업 지시서를 받은 컴퓨터가 엉뚱한 일을 한다면 그 작업 지시서는 잘못된 것이기 때문입니다. 이렇게 되면 코드를 다시 수정하는 코딩 작업이 필요합니다. 컴퓨터가 일을 제대로 할 때까지 계속해서 일련의 과정들을 반복하는 것이지요. 이러한 반복 작업을 디버깅이라고 합니다.

위에서 말한 컴퓨터만이 알 수 있는 숫자 코드들이 여러분에게 친숙한 소프트웨어 또는 프로그램이라고 부르는 것들입니다. 그리고 사람이 코딩한 코드를 문법 검사하고, 소프트웨어로 바꾸는 과정을 컴파일(Compile)이라고 부릅니다. 이와 같이 코딩, 컴파일 등의 과정들을 거쳐서 문서 작성 소프트웨어, 인터넷 브라우저 소프트웨어, 영화 보는 소프트웨어와 같은 모든 소프트웨어들이 다 만들어지는 것입니다.

이상의 설명에서 알 수 있듯이 소프트웨어가 만들어 지는 작업은 크게 코딩과 컴파일로 나눌 수 있습니다. 그런데, 코딩은 사람이 하는 일이고 컴파일은 컴퓨터가 하는 일입니다. 그래서 여러분이 어떤 소프트웨어를 만든다고 하시면 코딩만 하면 됩니다. 이 코딩이라는 작업을 매우 어렵게 생각하는데, 절대로 그렇게 어려운 작업이 아닙니다. 단지 컴퓨터 언어로 작업 지시서를 작성하면 되는 것입니다. 그런데, 컴퓨터에게 매우 복잡한 일을 시킨다면 작업 지시서도 매우 복잡해 지겠지요. 하지만 컴퓨터에게 매우 간단한 일을 시킨다면 정말 간단한 작업 지시서가 작성되는 것입니다.

예를 들어 다음과 같이 컴퓨터에 붙어 있는 전등을 3초 동안 켜다가 끈다고 하면 다음과 같은 간단한 작업 지시서를 작성하면 됩니다.

```
(LAMP, ON);  
Delay(3);  
(LAMP, OFF);
```

(LAMP, ON) 은 전등을 켜라고 지시하는 것입니다. Delay(3) 은 현재 상태를 3초 동안 유지하라는 것입니다. 즉, 전등을 3초동안 켜 채로 유지하라는 것이지요. (LAMP, OFF) 는 전등을 끄라는 것입니다.

여기서 조금만 더 복잡하고 실생활에 쓰이는 예를 들어 보겠습니다. 여러분이 아파트 복도나 계단에서 쉽게 볼 수 있는 사람이 다가오면 불이 켜지는 전등에도 간단한 컴퓨터가 붙어 있습니다. 그 컴퓨터에는 다음과 같이 조금 더 복잡한 작업 지시서가 들어 있을 것입니다.

```
if (DetectPerson)
(LAMP, ON);
```

if (DetectPerson) 는 만약 무엇인가가 다가오는 것을 발견하면 그 다음 줄에 오는 지시 내용을 해라 하는 뜻입니다. 그 다음 줄의 지시 내용은 (LAMP, ON) 입니다. 즉, 전등을 켜라는 것이지요.

위의 코드에서 무엇인가가 다가오는 것이 발견되어 전등을 켜는데, 계속 켜고 있을 수는 없지요. 어느 정도 시간이 지나면 자동으로 꺼야겠지요. 그래서 다음과 같은 코드를 추가 합니다.

```
if (DetectPerson) {
(LAMP, ON);
Delay(3);
(LAMP, OFF);
}
```

위와 같이 하면 "발견, 전등 켜기, 3초 후에 전등 자동 끄기" 라는 작업 지시서가 됩니다. 그런데, 이 일을 한번만 하면 안 되니깐 계속할 수 있도록 다음과 같이 해 줍니다.

```
while (1) {
if (DetectPerson) {
(LAMP, ON);
Delay(3);
(LAMP, OFF);
}
}
```

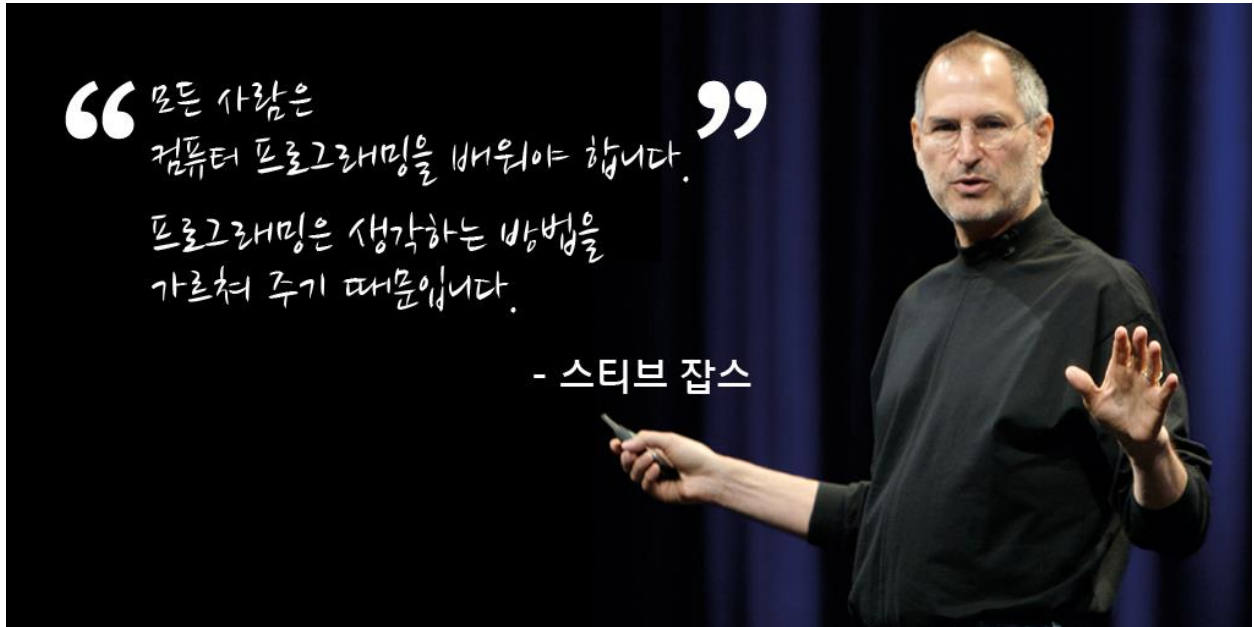
여기서 while (1) 이라고 쓴 부분은 중괄호({ })로 묶은 부분을 계속해서 반복해서 하라는 뜻입니다. 이렇게 하면 이 컴퓨터는 1년 365일 하루 종일 무엇인가가 다가오면 전등의 불을 3초 동안 켜는 일만 합니다. 3초 동안 켜는 시간이 너무 짧은 것 같으면 다음 코드만 고치면 얼마든지 시간을 조절할 수가 있습니다.

```
Delay(10)
```

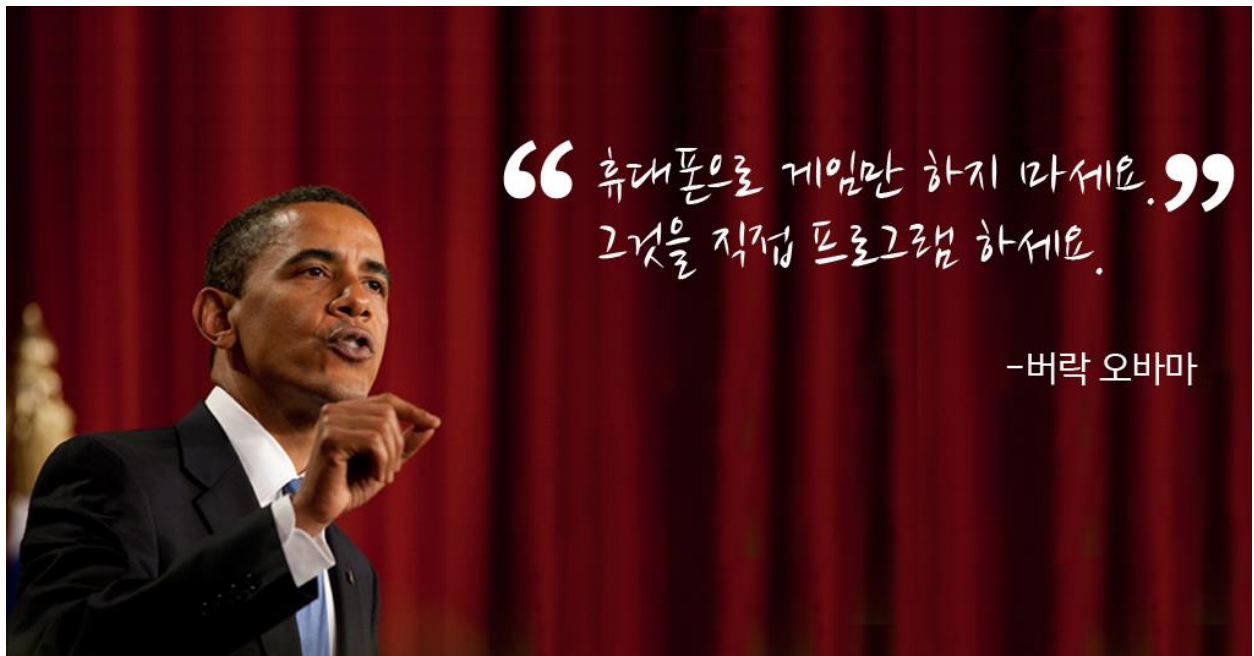
이것은 10초 동안 켜지도록 작업 지시서를 수정한 것입니다. 이렇게 코딩은 쉽고 재미있습니다. 또한 실생활에 보탬이 되지요.

[ 코딩교육? 이젠 선택이 아닌 필수! ]

애플사의 스티브 잡스는 다음과 같이 말했습니다.



버락 오바마 미국 대통령도 코딩 관련하여 다음과 같이 말했습니다.



이렇게 유명하신 분들이 코딩교육의 중요성에 대해서 강조하였습니다.



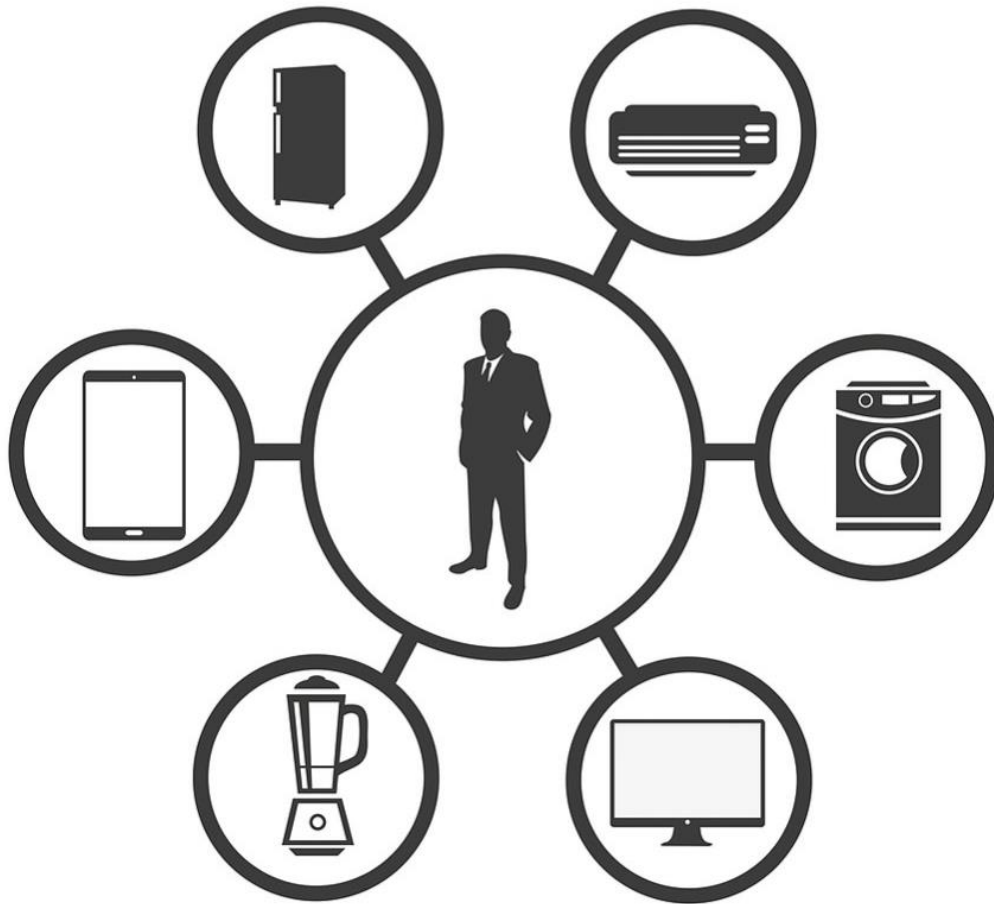
컴퓨터가 처음 만들어지던 시절에는 여러 사람이 컴퓨터 한 대를 공유하여 사용하였습니다.





PC(Personal Computre, 개인용 컴퓨터)가 보급되던 시절에는 한 사람이 컴퓨터 한 대를 사용하였습니다.



2017년 현재에는 한 사람에게 여러대의 컴퓨터가 있습니다. 한 사람이 스마트폰, PC, 네비게이션, 테블릿 등 여러 대의 컴퓨터를 사용합니다. 이 뿐만이 아니라 컴퓨터는 도처에 널려 있습니다. 집 안에는 TV, 냉장고, 도어락, 전자레인지 등 곳곳에 있습니다. 길을 걸을 때도 컴퓨터는 여기 저기에 보입니다. 신호등, CCTV, 전광판 등, 이제 여러분은 무수히 많은 컴퓨터 속에 살고 있습니다.



하지만, 대부분의 사람들은 컴퓨터 언어인 코드를 읽거나 사용할 줄 모릅니다. 이런 코드를 배우는 것은 매우 어렵고 전문적인 학교를 다녀야 하는 것처럼 느껴집니다. 물론 한 때는 코딩을 배우고 익히는 것이 매우 어렵고 힘든 일이었습니다. 하지만 기술의 진보로 이제는 코딩을 배우는 것이 그렇게 어렵지 않습니다.

	<p><b>마크 주커버그</b> 페이스북 CEO</p> <p>간단하게 뭔가 만들고 싶은 것이 생겨 컴퓨터 프로그래밍을 시작하게 되었습니다. 저나 저의 형제들이 좋아할 만한 것을 만들고 싶었어요. 조그마한 프로그램을 작성하고 그것을 계속 수정해 나갔죠.</p>
	<p><b>드류 휴스턴</b> 드롭박스 창업자</p> <p>이건 악기를 연주하거나 스포츠 운동을 하는 것과 크게 다르지 않아요. 처음 시작할 때는 아주 어려워 보이죠. 하지만 점점 시간이 지나면 어떻게 하는지 감이 잡힐 거예요.</p>

\*출처 : code.org

코딩(컴퓨터 프로그래밍)은 배울만 해요.  
코딩이 처음에 사람들을 두렵게 한다는 것을 저도 알고 있어요.  
그런 것들이 많이 있죠. 그렇지만 어떤 것이든지 안 그렇겠어요.

크리스 보쉬  
NBA 올스타



사실 코딩하는 사람들이 하는 일들은 꽤 단순해요  
코딩이라는 것은 일반 사람들이 생각하는 것처럼 복잡하고 어려운  
일을 하는게 아니에요.  
문제들을 보다 작은 문제들로 나누어서 하나 하나 해결하는 나가는  
단순한 작업들입니다.

마틴디 아데 아그보  
초기 페이스북 엔지니어



코딩을 배우기 위해 천재가 될 필요는 없어요.  
판단력이 필요할 뿐이죠..

바네사 허스트  
Girl Develop 창립자



덧셈, 뺄셈.. 다 그런 것들이에요.

빌 게이츠  
Microsoft 창립자



토니 세이  
zappos CEO

아마 모두들 구구단 정도는 알고 있으니까..

코딩을 하기 위해 천재가 될 필요는 없어요.  
글을 읽기 위해 천재가 돼야 하는건 아니잖아요.

브론웬 그라임스  
Valve 기술 아티스트







\*출처 : code.org

이렇게 곳곳에 컴퓨터가 널리 있게 되면서 코딩교육의 필요성이 높아지고 여기에 코딩을 배우는 것이 그렇게 어렵지 않은 환경이 되다 보니 이젠 코딩교육이 선택이 아닌 필수가 되었습니다. 이러한 필요성 때문에 세계 여러나라에서 코딩교육을 정규 교육 과정에 포함시키기 시작하였습니다.

앞으로 기술 발전이 더 고도화되고 복잡해짐에 따라서 컴퓨터의 쓰임새는 더 많아지고 다양해질 것입니다. 이에 따라서 코딩교육은 더욱 더 중요해질 것입니다. 그리고 코딩을 잘 하는 사람이 미래 사회를 이끌어 가게 될 것입니다.



 <p><b>드류 휴스턴</b> 드롭박스 창업자</p>	<p>아이디어를 상상하고, 그것을 실제로 만드는 것이 가능해진 것 뿐만 아니라, 버튼 하나만 누르면 수백만명의 사람이 함께 할 수 있게 된다는 건... 제 생각엔 이런 걸 경험할 수 있는 세대는 우리가 처음이 아닌가 싶어요.</p>
 <p><b>마크 주커버그</b> 페이스북 CEO</p>	<p>대학 기숙사 방에서도 뭔가를 시작할 수 있다고 생각해 보세요. 큰 회사를 만들어 보지 않은 사람들이 한데 모여서, 함께 수억명 사람들의 삶을 바꿀 수 있는 무엇인가를 만들 수 있다고 생각해 보세요. 진짜 끝내주는 일이죠? 그렇지 않아요? 놀라운 일이에요.</p>
 <p><b>윌.아이.엠</b> 블랙 아이드피스 리더</p>	<p>우린 지금 2015년에 있어요. 우리 모두는 컴퓨터과학기술에 의존하고 있죠. 통신, 은행, 정보 ... 그런데, 우리 중 어느 누구도 코드를 읽거나 사용할 줄 몰라요.</p>
<p>앞으로 프로그래머는 미래의 마법사가 될 것입니다. 다른 사람과 비교하면 마치 마법의 힘을 가지고 있는 것처럼 보이는 거죠.</p>	 <p><b>게이브 뉴웰</b> 밸브 창립</p>

\*출처 : code.org

## [ 코딩키트 소개 ]

코딩키트(Coding Kit)는 여러분이 코딩 공부를 쉽고 재미있게 할 수 있도록 도와주는 키트입니다.

코딩키트는 다음 그림과 같은 로봇 제작에 많이 사용되는 부품들이 장착된 하드웨어 보드를 이용하여 코딩 공부를 하는 것입니다.



코딩키트를 활용하면 로봇 제작에 응용할 수 있는 LED, 모터, LCD, 센서 등을 직접 다루는 코딩을 함으로써 지루하지 않고 재미있게 코딩을 배울 수 있습니다.

또한 요즘 크게 부각되고 있는 사물인터넷(IoT : Internet of Things) 등으로 인하여 하드웨어를 이용한 코딩 공부의 중요성이 강조되고 있습니다. 전(前) 구글 부사장이자 안드로이드 OS를 처음 개발한 앤디 루빈도 이렇게 말했습니다.

**"..hardware will drive future ecosystems." (하드웨어가 미래의 IT 생태계를 이끌게 될 것이다.)**

이렇게 하드웨어를 이용한 코딩 학습은 매우 중요해지고 있습니다. 그래서 코딩키트를 이용하여 코딩을 배우게 되면 컴퓨터 언어를 익힐 수 있을 뿐만 아니라 LED, 모터, LCD, 센서 등 각종 하드웨어에 대한 지식도 높아지고, 관련된 코딩도 배우게 됩니다.

컴퓨터라고 하면 일반적으로 노트북, 데스크탑 등의 PC(Personal Computre) 를 생각하는 경우가 많은데, 이러한 PC 는 전 세계에 있는 모든 컴퓨터의 5% 도 되지 않는 숫자입니다. 그래서 프로그래머라고 하는 코딩을 직업으로 하는 사람들 중 극히 일부가 PC 용 소프트웨어를 다루는 코딩을 합니다. 대부분의 프로그래머는 PC가 아닌 우리의 일상 생활에 있는 스마트폰, TV, 냉장고 등의 소프트웨어, 또는 큰 건물의 자

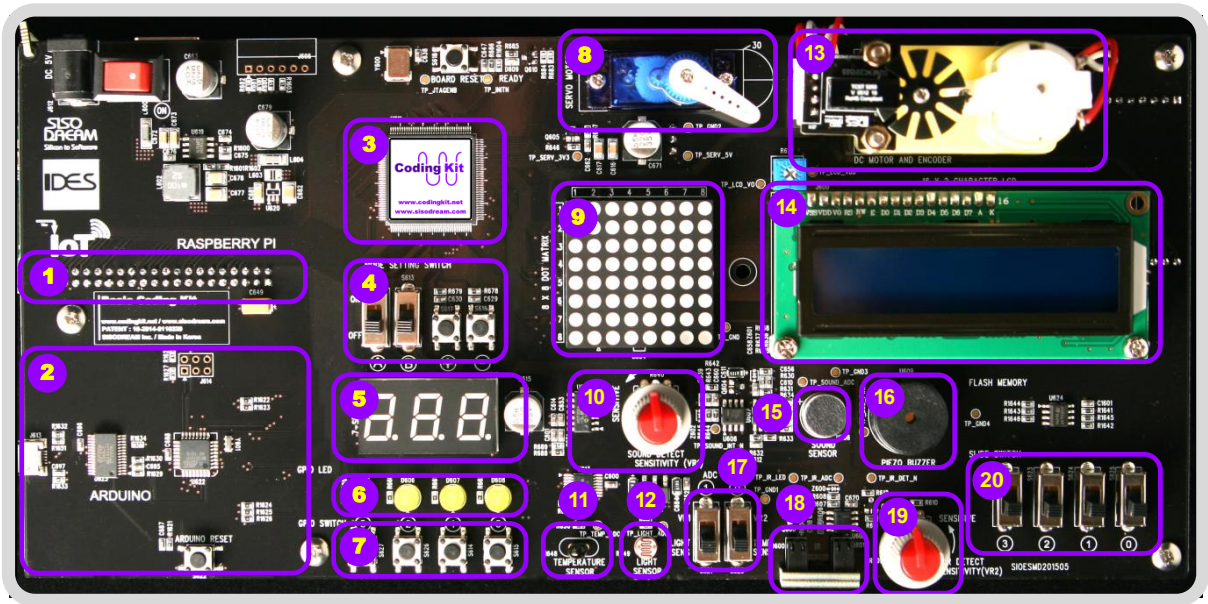
동화 장치들을 제어하는 소프트웨어, 또는 인공위성을 컨트롤 하고 통신하는 소프트웨어 등의 코딩을 합니다. 이러한 소프트웨어들은 대부분은 하드웨어를 직접적으로 다루는 소프트웨어 코딩입니다. 그래서 하드웨어를 이용한 코딩 학습이 중요한 것입니다.

그럼 이제 여러분에게 쉽고 재미있게 코딩을 배울 수 있는 코딩킷을 본격적으로 소개해 보겠습니다. 다양한 디바이스(Device : 버튼, 스위치, LCD 등의 부품을 말함)가 장착된 코딩보드(Coding Board)가 있습니다. 그리고 코딩에 관한 쉽고 친절한 설명이 되어 있는 교재로 코딩북(Coding Book)이 있습니다. 또한 질문과 답변 및 새로운 예제를 만날수 있는 코딩사이트(<http://www.codingsite.net>)가 있습니다. 코딩보드, 코딩북, 코딩사이트로 구성된 코딩킷은 여러분에게 **세상에서 코딩을 배우는 가장 쉽고 재미있는 방법**을 알려 드립니다. 여러분의 코딩 공부를 위한 최고의 지원을 해 드립니다.

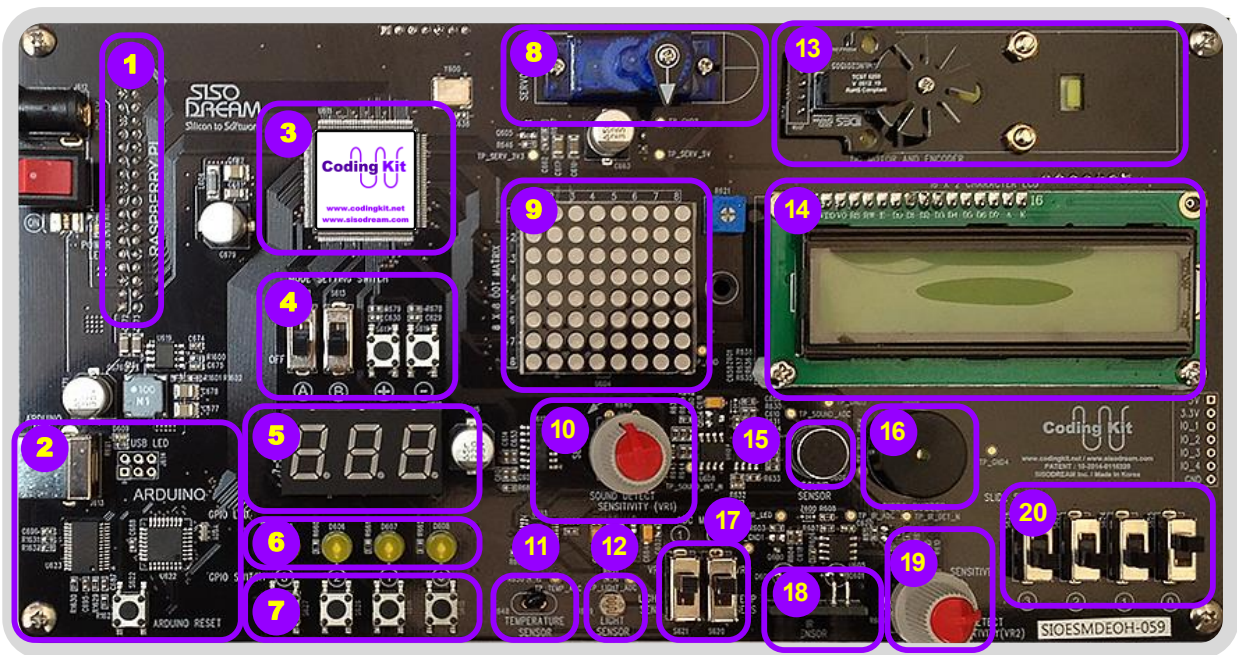
먼저 코딩보드에 대해서 알아보도록 하겠습니다. 몇 년 전만 해도 하드웨어와 접목된 코딩을 배우는 것은 매우 어려웠습니다. 그런데, 근래에는 이런 하드웨어 코딩을 매우 쉽게 배울 수 있도록 만들어 둔 하드웨어와 소프트웨어가 조합된 시스템들이 있습니다. 이러한 시스템들 중 대표적인 시스템이 아두이노(Arduino)와 라즈베리파이(Raspberry Pi)입니다. 이 두 시스템이 지금은 워낙 유명해져서 여러분이 한번씩은 들어 봤을 것입니다. 코딩킷은 이 대표적인 두 시스템을 모두 활용합니다. 그 중 아두이노는 코딩보드에 장착이 되어 있고 라즈베리파이는 장착할 수 있는 커넥터가 있습니다. 그래서 여러분이 코딩킷을 이용하여 매우 쉽게 하드웨어 융합 코딩을 배울 수 있습니다. 여러분이 아두이노, 라즈베리파이를 처음 들어 봤다고 해도 상관 없습니다. 굳이 그 두 시스템을 알고 시작할 필요는 없습니다. 코딩킷을 이용하여 코딩 공부를 해 나가다 보면 자연스럽게 알게 될 것입니다.

또한 코딩보드에는 다음 그림과 같이 간단한 입출력 장치인 LED, 버튼, 스위치들과 각종 모터 및 센서들이 장착되어 있습니다. 여기에 캐릭터 LCD, 도트매트릭스, 세븐세그먼트 같은 디스플레이 장치도 연결되어 있습니다. 현재 코딩보드는 다음과 같이 2 가지 형태로 출시되어 있습니다. 부품의 배치에 약간의 차이가 있을 뿐 기능과 사용 방법은 동일합니다.





< 코딩보드 Long Version >



< 코딩보드 Short Version >

1. 라즈베리파이 커넥터 (Raspberry Pi Connector)
2. 아두이노 (Arduino)
3. 스위칭 시스템 스위칭 칩 (Switching System, Switching Chip)



4. 스위칭 시스템 스위치와 버튼 (Switching System, Switch & Button)
5. 스위칭 시스템 세븐세그먼트 (Switching System, 7-Segment)
6. LED 4 개
7. 버튼 4 개 (Button)
8. 서보 모터 (Servo Motor)
9. 도트매트릭스 (Dotmatrix)
10. 가변 저항 1 (Variable Resistor 1)
11. 온도 센서 (Temperature Sensor)
12. 밝기 센서 (Light Sensor)
13. DC 모터 (DC Motor)와 인코더(Encoder)
14. 캐릭터 LCD (Character LCD)
15. 소리 센서 (Sound Sensor)
16. 부저 (Buzzer)
17. ADC 모드 스위치
18. 적외선 센서 (IR Sensor)
19. 가변 저항 2 (Variable Resistor 2)
20. 스위치 4 개

앞으로 코딩보드에 장착된 디바이스들에 대해서 많은 예제를 해 볼 것입니다. 매우 재미있는 시간이 될 것입니다.

이제 코딩북에 대해서 설명 드리겠습니다. 코딩북은 기본적으로 읽는 사람이 코딩을 전혀 모른다는 가정 하에서 시작합니다. 그래서 매우 기초적인 것도 모두 다 설명합니다. 그리고 최대한 이해하기 쉽게 설명을 합니다. 또한 소프트웨어적인 코딩만 할 수 있고 하드웨어를 잘 모르시는 분들에게는 하드웨어에 대한 이해를 돕는데 매우 많은 도움이 되도록 설명하였습니다. 하드웨어를 동작시키기 위해서는 코딩을 어떻게 해야 하는지에 대한 매우 쉽고 자세한 설명이 있습니다.

매우 쉬운 블록코딩부터 시작해서 난이도가 있는 아두이노와 라즈베리파이 연동 코딩까지 코딩에 관하여 매우 폭넓게 학습하실 수 있습니다. 그래서 집에 코딩키트만 하나 있으면 초등학교 아이들뿐만 아니라 가정주부인 엄마도 배울 수 있고, 직장인 아빠는 물론 대학에 다니는 삼촌, 이모, 고모까지 모두 배울 수 있습니다.

코딩북은 다음과 같이 구성되어 있습니다.

1. 코딩키트 소개  
본 교재로 코딩키트에 대한 소개를 합니다.

## 2. 블록코딩 – 아빠와 함께하는 곰곰이의 코딩 놀이

블록코딩이란 컴퓨터 언어를 문법적으로 타이핑하여 코딩하는 것이 아니고 레고블록을 조립하듯이 하는 코딩을 말합니다. 처음 코딩을 접하는 코딩 입문자를 대상으로 한 코딩을 매우 쉽게 배울 수 있는 코딩 방식입니다. 코딩북에는 이 블록코딩 교재가 있습니다.

## 3. 아두이노 코딩

아두이노를 이용한 코딩을 쉽고 재미있게 배울 수 있는 코딩 교재입니다.

## 4. 라즈베리파이 코딩

아두이노보다 상위 레벨의 컴퓨터인 라즈베리파이를 이용한 코딩을 배울 수 있는 교재입니다.

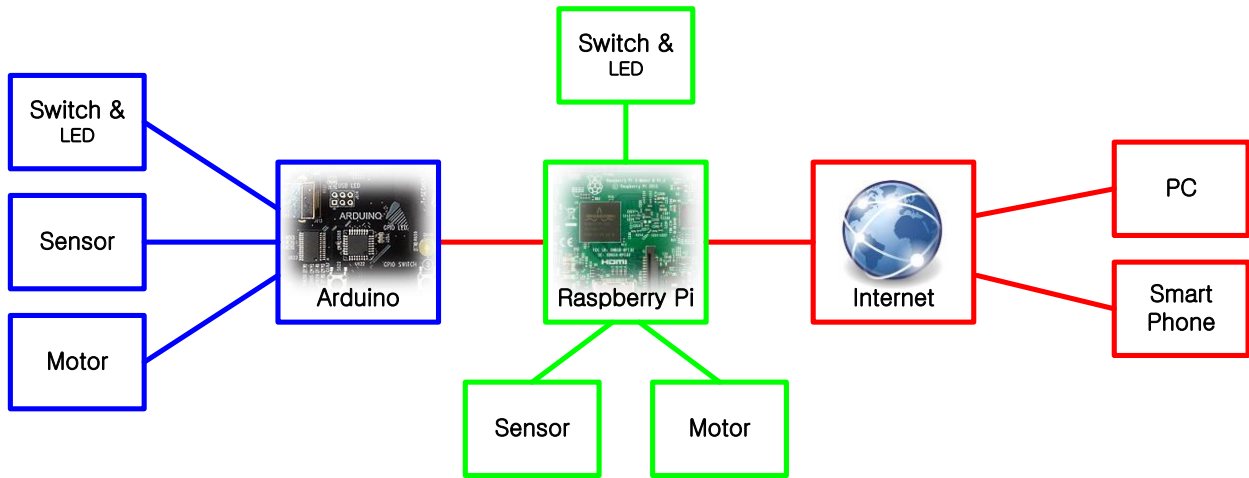
## 5. 코딩킷 확장판 - 아두이노와 라즈베리파이 연동 및 사물인터넷 코딩

아두이노와 라즈베리파이 두 소형 컴퓨터가 서로 연동하는 코드를 배울 수 있는 교재입니다. 또한 요즘 한참 이슈가 되고 있는 사물인터넷 코딩을 아두이노, 라즈베리파이, 인터넷을 이용하여 배웁니다.

코딩북의 순서는 디바이스들을 컨트롤해 나가면서 컴퓨터 언어에 대한 개념과 문법 등을 설명합니다. 여타의 교육용 하드웨어 보드의 설명서는 단순히 하드웨어 컨트롤에 대한 설명만 있습니다. 컴퓨터 언어를 이용한 코딩에 대한 설명등은 매우 부족한 형편입니다. 하지만 코딩킷에서는 컴퓨터 언어의 개념 및 문법 등의 설명을 하면서 하드웨어 디바이스들을 컨트롤합니다. 이렇게 함으로서 여러분은 자연스럽게 컴퓨터 언어와 코딩도 익히고 하드웨어도 배웁니다.

또한 코딩북에서는 수많은 예제를 이용하여 코딩보드에 있는 디바이스들을 하나 하나 컨트롤하는 것에 대한 설명을 합니다. 이 예제와 설명으로 디바이스를 컨트롤하는 방법을 배울 뿐 만 아니라 컴퓨터 언어에 대한 이해와 코딩 기술을 습득할 수 있습니다. 이렇게하여 컴퓨터 언어의 개념과 문법을 배울 수 있고 컴퓨터 언어를 활용한 코딩 감을 익힐 수 있는 것입니다. 여기서 말하는 코딩 감이라는 것은 매우 중요한 것입니다. 이 코딩 감이라는 것이 있어야 여러분 스스로 코딩을 할 수 있게 되는 것입니다. 우리가 영어 공부할 때도 영어에 대한 감이 늘어야 영어를 잘 할 수 있는 것과 같은 이치입니다.

코딩북의 전체적인 내용의 흐름은 다음 그림으로 표현할 수 있습니다.



블록코딩과 아두이노 코딩에서는 아두이노로 스위치 및 LED 같은 단순 디바이스를 컨트롤하는 것을 배우는 것으로 시작하여 센서의 신호를 받고 모터를 컨트롤하는 등의 다양한 코딩을 배웁니다. 그림의 파란색 부분입니다. 라즈베리파이 코딩에서는 라즈베리파이로 여러 디바이스를 컨트롤하는 것을 배우며 파이썬이라는 코딩 언어도 배웁니다. 그림의 연두색 부분입니다. 다섯번째로 배우는 코딩키트 확장판에서는 코딩키트로 코딩할 수 있는 코딩의 완성판으로 아두이노와 라즈베리파이를 연동하는 코딩을 배우고 이를 바탕으로 사물인터넷(Internet of Things : IoT) 코딩을 배웁니다. 그림의 붉은색 부분입니다.

이제 코딩사이트에 대해서 설명드리겠습니다. 코딩사이트 주소는 다음과 같습니다.

<http://www.codingsite.net>

코딩사이트에는 코딩키트 관련 자료가 많이 있습니다. 코딩북의 코드 이외의 많은 예제 코드들이 있습니다. 이 예제들에는 난이도가 매우 높은 예제들도 다수 포함되어 있어 여러분의 코딩 실력을 한층 더 높인데 많은 도움이 될 것입니다. 또한 질문과 답변 게시판을 이용하여 여러분이 궁금해 하시거나 코딩 중에 막히는 부분등에 대한 질문을 하시면 빠르고 친절하게 답변을 드립니다.

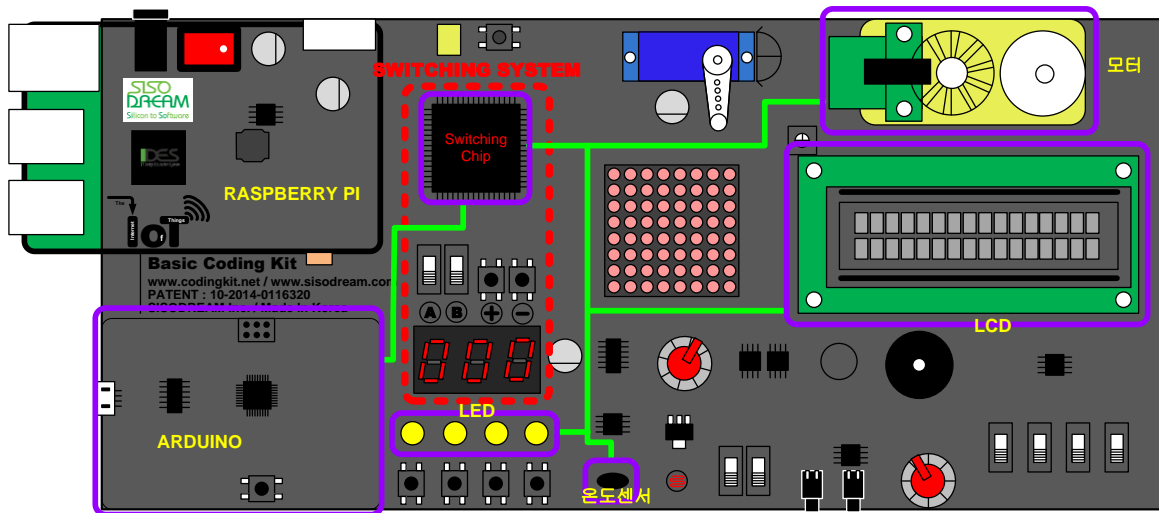
코딩사이트에는 디바이스 설명과 같은 하드웨어 자료도 많이 있습니다. 여기에 DIY 게시판을 만들어 코딩키트를 활용하여 직접 제작하시는 분들에게도 많은 도움을 드릴 예정입니다.

여러분이 코딩보드, 코딩북, 코딩사이트를 적절히 잘 활용하시면 여러분의 코딩 실력은 매우 빠르게 향상될 것입니다. 부디 여러분의 코딩 인생에 저희의 코딩키트가 든든한 밑거름이 되었으면 합니다.

여기서 또 한가지 매우 중요한 코딩키트의 특징을 말씀 드리고 코딩키트 소개를 마치도록 하겠습니다. 코딩키트에는 매우 많은 디바이스들이 장착되어 있습니다. 이 디바이스들은 아두이노 또는 라즈베리파이에게 매우 다양한 조합으로 연결되도록 설계되어 있습니다. 이러한 다양한 연결을 통하여 여러분은 여러 난이도

의 매우 많은 예제를 코딩키트로 해 볼 수 있습니다. 예를 들어 온도 센서로 입력되는 온도를 LCD 에 표시 하면서, 온도가 너무 높으면 선풍기를 모터로 돌려 온도를 낮춥니다. 온도가 너무 낮으면 난로를 LED 로 켜 온도를 높입니다. 이 예제를 하려면 아두이노 혹은 라즈베리파이에 온도 센서, LED, LCD, 모터를 연결 해야 합니다. 하지만 코딩키트에는 이런 디바이스들 이외에도 세븐세그먼트, 도트매트릭스, 밝기 센서, 소리 센서 등이 장착되어 있습니다. 이 디바이스들을 이용하려면 이전에 연결된 것을 끊고 새로운 것을 연결해야 합니다. 심지어 코딩키트에서는 똑같은 디바이스를 아두이노에 연결하였다가 라즈베리파이에 연결하였다가 합니다. 그렇게 연결을 바꾸는데 여러분은 버튼을 몇 번 누르기만 하면 됩니다. 만약 다른 아두이노나 라즈베리파이 학습 제품을 이용했다면, 이러한 디바이스들을 여러 번 뺐다 꽂았다는 하거나, 전선을 연결하는 등의 매우 번거로운 작업들을 거쳐야 하기 때문에 여러 난이도의 다양한 예제를 연습하기가 더 어렵습니다. 이러한 것들을 모두 가능하게 하는 것은 코딩키트의 강력한 **스위칭 시스템** 때문입니다.

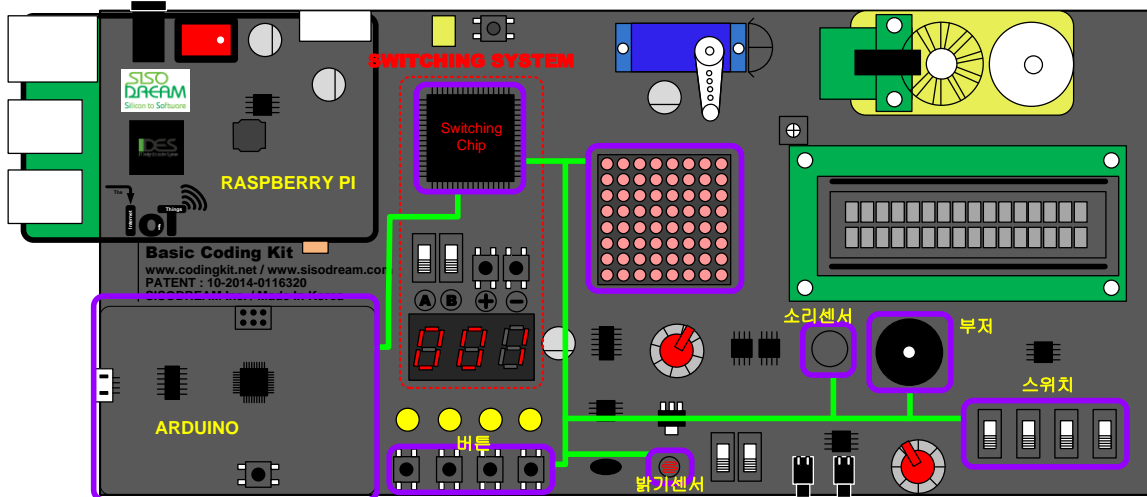
이 부분에 대해서 예를 들어 조금 더 설명드리겠습니다. 다음은 코딩키트가 아두이노에 모터, LCD, LED, 온도 센서를 연결한 구성을 보여 줍니다.



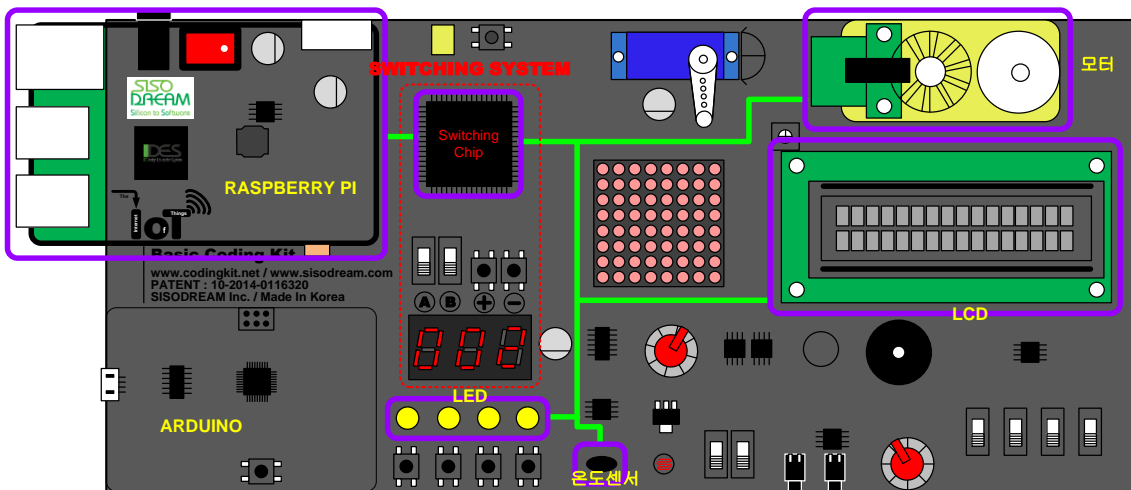
위 그림의 붉은색 점선 안이 스위칭 시스템입니다. 이 부분의 A, B 스위치와 +, - 버튼으로 스위칭 칩을 컨트롤합니다. 그러면 이 스위칭 칩에서 아두이노와 라즈베리파이에 디바이스의 연결을 만들어 냅니다. 현재의 스위칭 시스템에 의한 연결이 몇 번인지는 세븐 세그먼트에 표시됩니다. 현재 아두이노에 모터, LCD, LED, 온도 센서가 연결된 것은 000 번( ) 연결이라고 세븐 세그먼트가 표시하는 것입니다. (이 000 번은 실제 번호와 조금 다른 설명을 위한 예입니다.) 위 그림에서 보라색 박스는 디바이스들을 표시하는 것이고 연두색 선은 각각의 연결을 표시하는 것입니다.

여기서 + 버튼을 누르면 다음 그림과 같이 세븐세그먼트가 001 번을 가리키고 또 다른 연결을 만들어 냅니다.

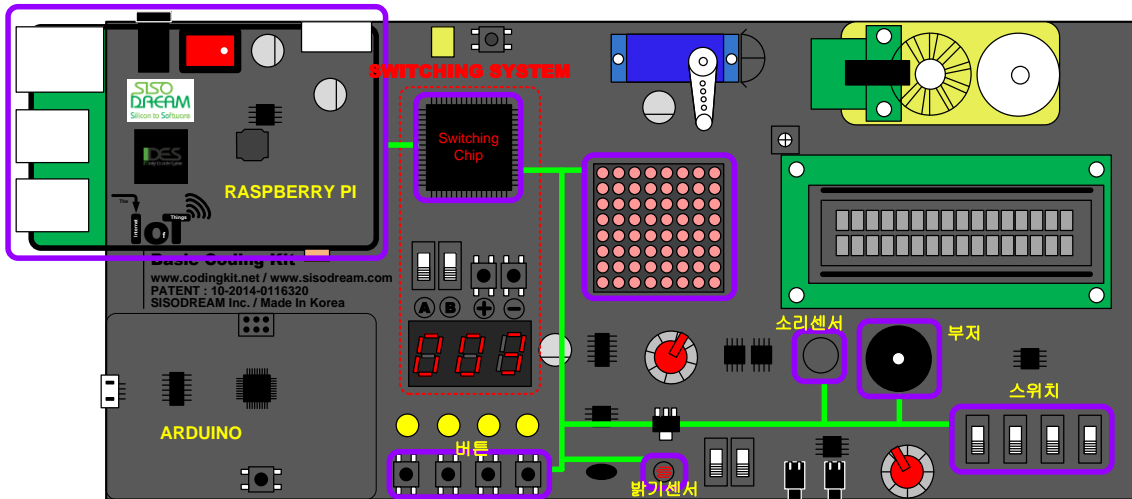




이 001 번 연결에는 버튼, 밝기 센서, 소리 센서, 부저, 스위치, 도트매트릭스가 연결됩니다. + 버튼을 또 누르면 다음과 그림과 같이 세븐세그먼트는 002 번을 표시하면서 스위칭 칩은 002 번 연결을 만들어 냅니다.



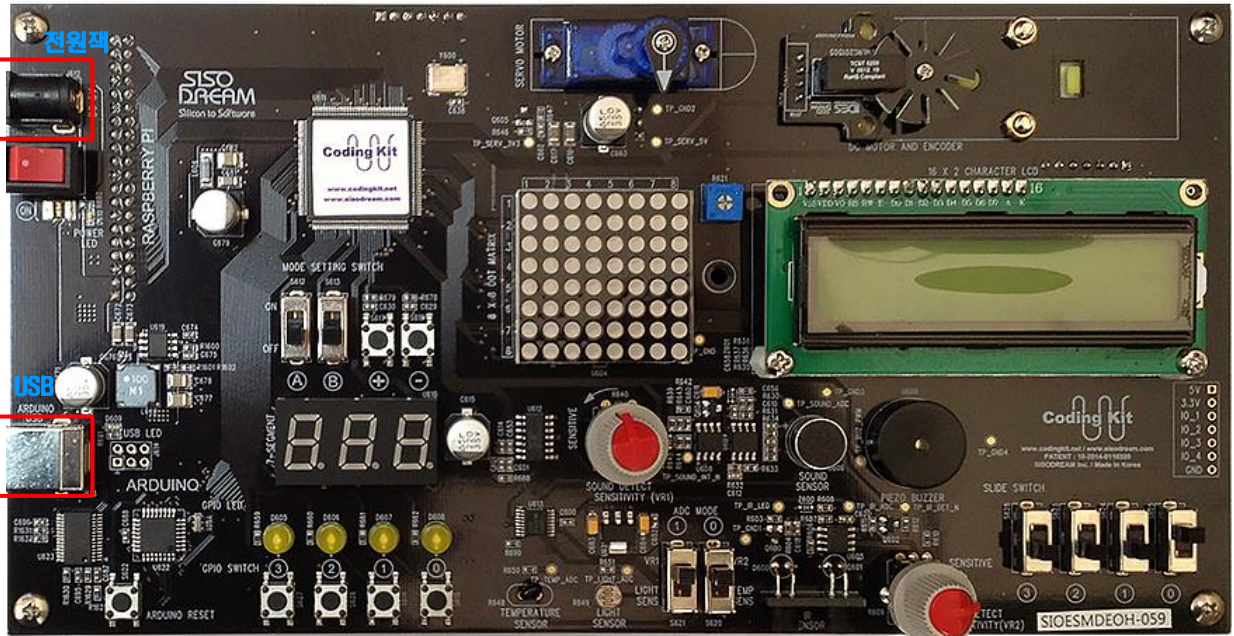
위의 그림은 아두이노가 아닌 라즈베리파이에 모터, LCD, LED, 온도 센서가 연결된 것입니다. 또 + 버튼을 누르면 다음과 같이 003 번 연결이 형성됩니다.



이와 같이 코딩키트는 스위칭 시스템을 이용하여 버튼을 누르는 동작만으로 여러가지 연결을 만들어 낼 수 있습니다. 이렇게 다양한 연결을 통하여 다양한 난이도의 예제를 손쉽게 할 수 있습니다. 이러한 예제 들은 코딩북에서 실제 동작 코드와 쉬운 설명으로 쓰여져 있습니다. 또한 쉽고 친절한 프로그램 언어의 문법 설명까지 포함되어 있어 코딩 공부에 최적의 환경을 제공합니다.

[ 코딩킷 준비 및 스위칭 시스템 ]

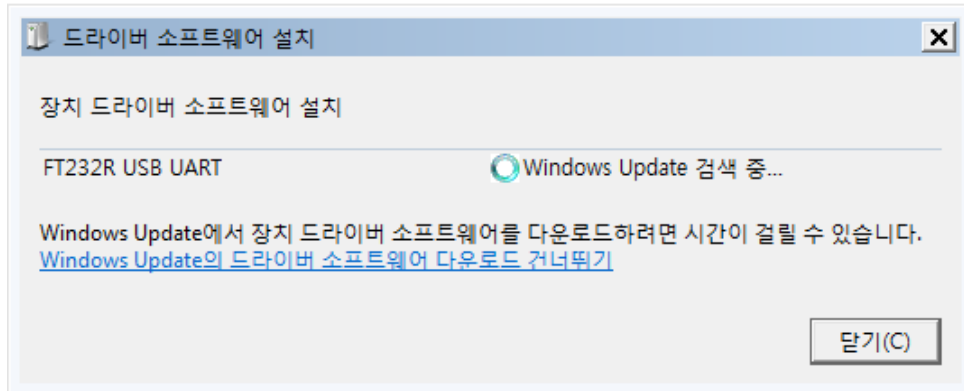
먼저 아답터를 전원 잭에 연결합니다. 그리고 전원 스위치를 눌러 코딩킷에 전기가 들어가도록 합니다. 그리고 USB 케이블을 PC 에 연결합니다.



그러면 세븐세그먼트가 깜박일 것입니다. 그리고 PC 에서는 다음과 같이 아두이노 드라이버 소프트웨어를 설치하기 위한 준비를 할 것입니다.

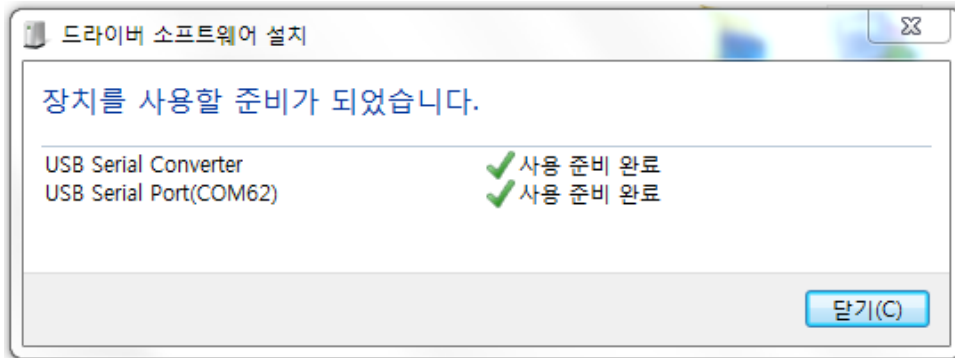


위의 그림의 붉은 박스 안을 클릭하면 다음과 같이 드라이버 소프트웨어를 자동으로 설치하는 화면을 보실 수 있습니다.

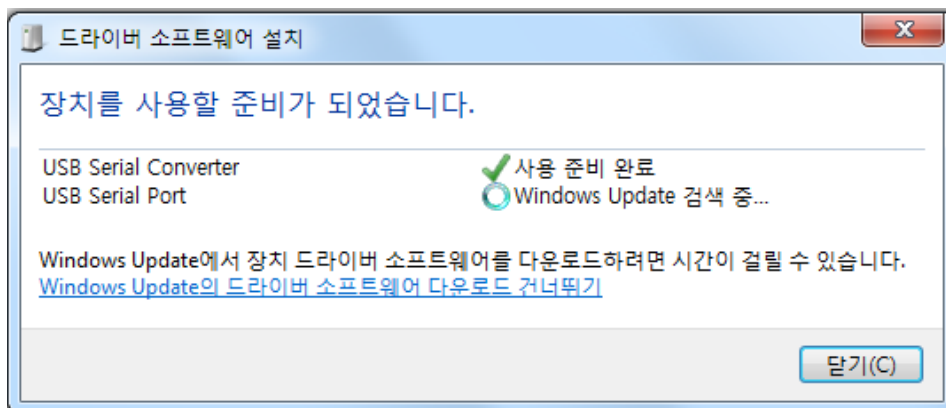


PC 의 설정은 이렇게 자동 이루어질 것입니다. 하지만 시간이 좀 오래 걸립니다.

다음과 같이 "드라이버 소프트웨어 설치"에서 모두 사용 준비 완료가 표시되면 드라이버 설치가 완료된 것입니다.

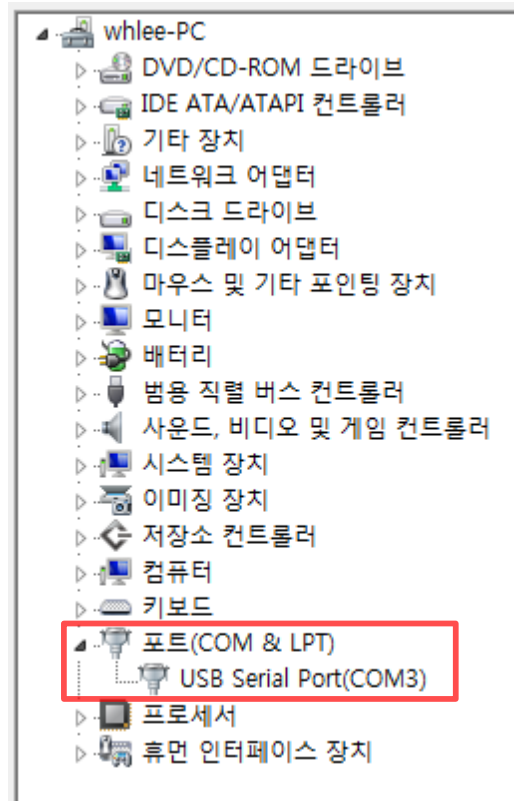


위의 그림과 같지 않고 다음과 같이 아직도 설치 중이라면 조금 더 기다립니다.



설치가 완료 되면 장치관리자에서 다음 그림과 같이 "USB Serial Port(COM3)" 을 확인할 수 있습니다. 그림에서는 COM3 이지만 다른 번호의 포트일 수 있습니다.

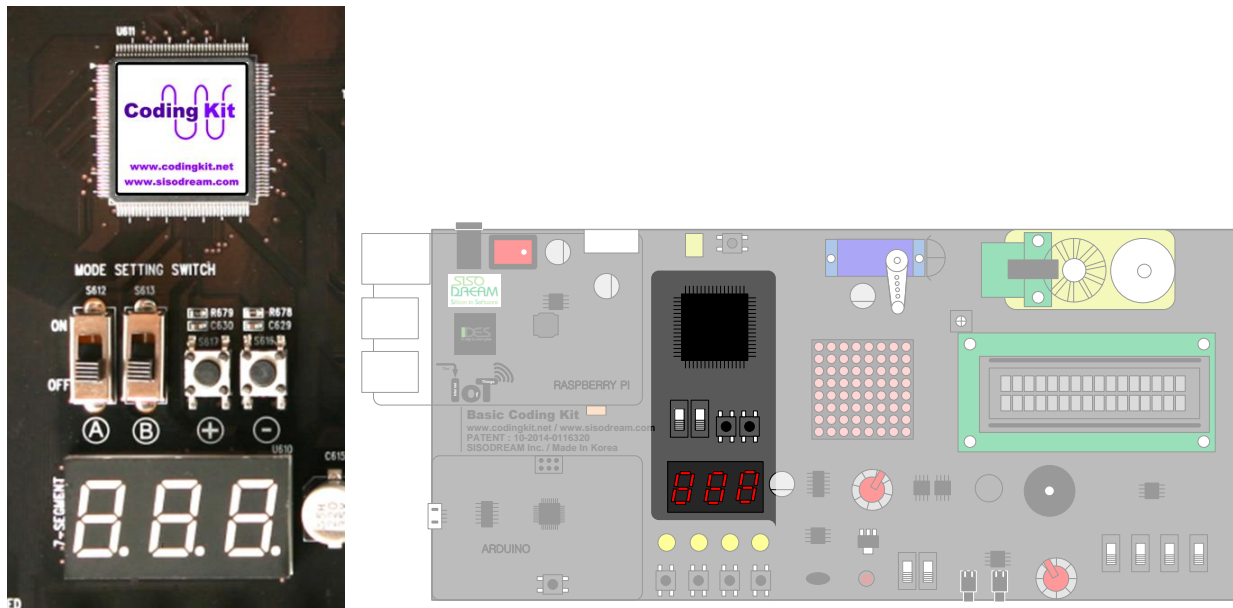




**Note :** 위의 설명은 Windows7 을 기준으로 설명하였습니다. Windows8, Windows10 에서의 사용은 코딩사이트의 관련 게시물을 참고해 주십시오. 코딩키트는 Windows7, Windows8, Windows10 만을 지원합니다.

**Note :** 만약 한참을 기다려도 드라이버 소프트웨어가 자동으로 설치되지 않을 경우 코딩사이트에 문의해 주십시오.

위에서 잠깐 언급하였던 코딩키트의 스위칭 시스템에 대해서 자세히 설명하겠습니다. 코딩키트의 스위칭 시스템은 아래 그림과 같이 스위칭 칩과 2 개의 스위치, 2 개의 버튼, 3 개의 세븐세그먼트로 구성됩니다.



이 중 스위치 2 개와 버튼 2 개는 스위칭 시스템을 컨트롤하는데 사용합니다. 코딩키트는 아두이노와 라즈베리파이 두 소형 컴퓨터에 어떤 디바이스들을 연결할지를 결정함으로써 여러분이 어떤 코딩을 할지를 결정합니다.

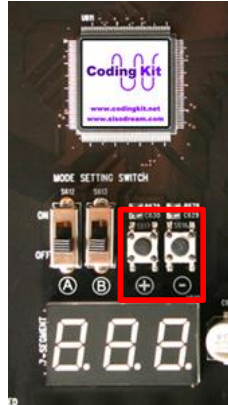
2 개의 스위치 A, B 를 다음과 같이 조정하여 어떤 코딩을 할지를 결정합니다.

A	B	코딩 종류	세븐세그먼트 표시
아래	아래	아두이노 코딩	
아래	위	블록 코딩	
위	아래	라즈베리파이 코딩	
위	위	아두이노 + 라즈베리파이 코딩	

위의 표와 같이 A, B 스위치의 조정에 따라서 코딩의 종류가 변경됩니다. 만약 아두이노 코딩을 하기를 원한다면 스위치 A, B 모두 아래로 내리면 됩니다. 블록코딩을 원한다면 A 스위치는 내리고, B 스위치는 올리면 됩니다. 라즈베리파이 코딩은 A 는 올리고, B 는 내립니다. 아두이노와 라즈베리파이 연동 코딩은 A,

B 모두 올리는 것입니다. 이 때마다 세븐세그먼트의 표시는 위의 표에 나온 것처럼 변하면서 깜박일 것입니다.

다음 그림의 붉은색 박스 안의 + 버튼을 누르면 두번째, 세번째 세븐세그먼트의 숫자가 올라갑니다. - 버튼을 누르면 숫자가 내려갑니다. 이렇게 하여 연결 번호를 설정합니다.



이렇게 세븐세그먼트에 표시된 영문과 숫자를 코딩키트에서는 스위칭 ID 라고 합니다. 이 스위칭 ID는 아두이노 일때는 A00, A01, A02, ... 로 표시가 되고, 블록코딩일 때는 b00, b01, b02, ... 로 표시가 됩니다. 라즈베리파이는 r00, r01, r02, ..., 아두이노와 라즈베리파이 연동일 때는 E00, E01, E02, ... 로 표시됩니다. 여기서 E 는 Extension 의 앞글자로 확장 모드를 의미합니다.

이렇게 스위치와 버튼으로 스위칭 ID 를 선택합니다. 이렇게 선택을 하여도 세븐세그먼트 계속 깜박이고 있습니다. 이것은 연결 설정을 선택만 했을 뿐이지 아직 연결은 이루어지지 않았다는 표시입니다. 이제 여러분이 + 혹은 - 버튼을 1~2 초간 계속 누르고 있으면 세븐세그먼트의 깜박임이 멈추고 연결이 이루어집니다. 스위칭 시스템 설정이 완료된 것입니다. 이렇게 세븐세그먼트의 깜박임이 멈춘 후에 코딩키트는 여러분이 코드에 맞는 수행을 합니다.

또 다른 스위칭 ID 로 바꾸려면 어떻게 해야 할까요? 그냥 가볍게 버튼을 다시 누르거나 스위치를 움직이면 세븐세그먼트는 다시 깜박이면서 스위치 ID 선택 모드가 됩니다. 이 때 연결은 다시 끊깁니다. 이렇게 해서 다시 설정하고 싶은 ID 를 설정합니다. 모든 설정은 세븐세그먼트가 깜박이면서 표시해 줍니다. 다 완료되면 다시 + 나 - 버튼을 1~2 초간 계속 누르고 있으면 됩니다. 이제 세븐세그먼트의 깜박임이 멈추고 사용하시면 됩니다.

각각의 스위칭 ID에 어떤 연결이 되어 있는지에 대해서는 각각의 교재의 "부록"에 정리해 두었습니다.