

---

# 아두이노 시뮬레이션 프로그래밍

---

v1.0

김영준 저

공학박사, 목원대학교 겸임교수  
前 Microsoft 수석연구원

헬로앱스

<http://www.helloapps.co.kr>

## 03 절전기능 LED 후레쉬

### 학습 목표

- 아두이노의 디지털 버튼 작동 방법을 이해하고 다양한 용도로 활용할 수 있다.
- 문자와 문자라인 출력문을 이용하여 아두이노 내부의 값을 컴퓨터 화면에 출력할 수 있다.

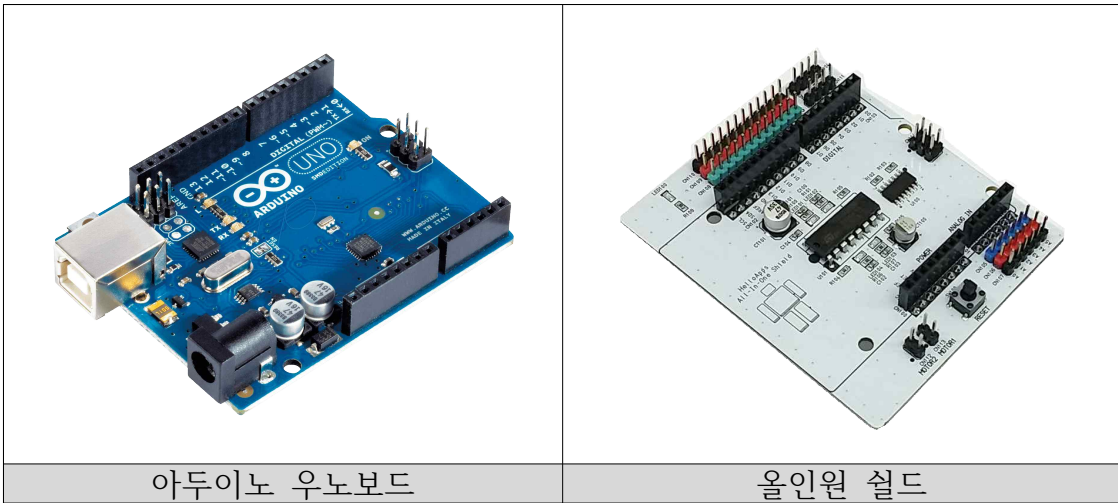
### 실습 개요

- 문자라인 출력문을 이용하여 값을 출력해 본다.
- 문자출력문과 문자라인 출력문을 조합하여 구구단을 출력해 본다.
- 디지털 읽기 (DigitalRead) 명령어로 버튼값을 인식해 본다.
- 버튼 눌림 상태를 이용하여 LED가 켜져 있는 시간을 지정할 수 있다.

### 3.1 준비하기

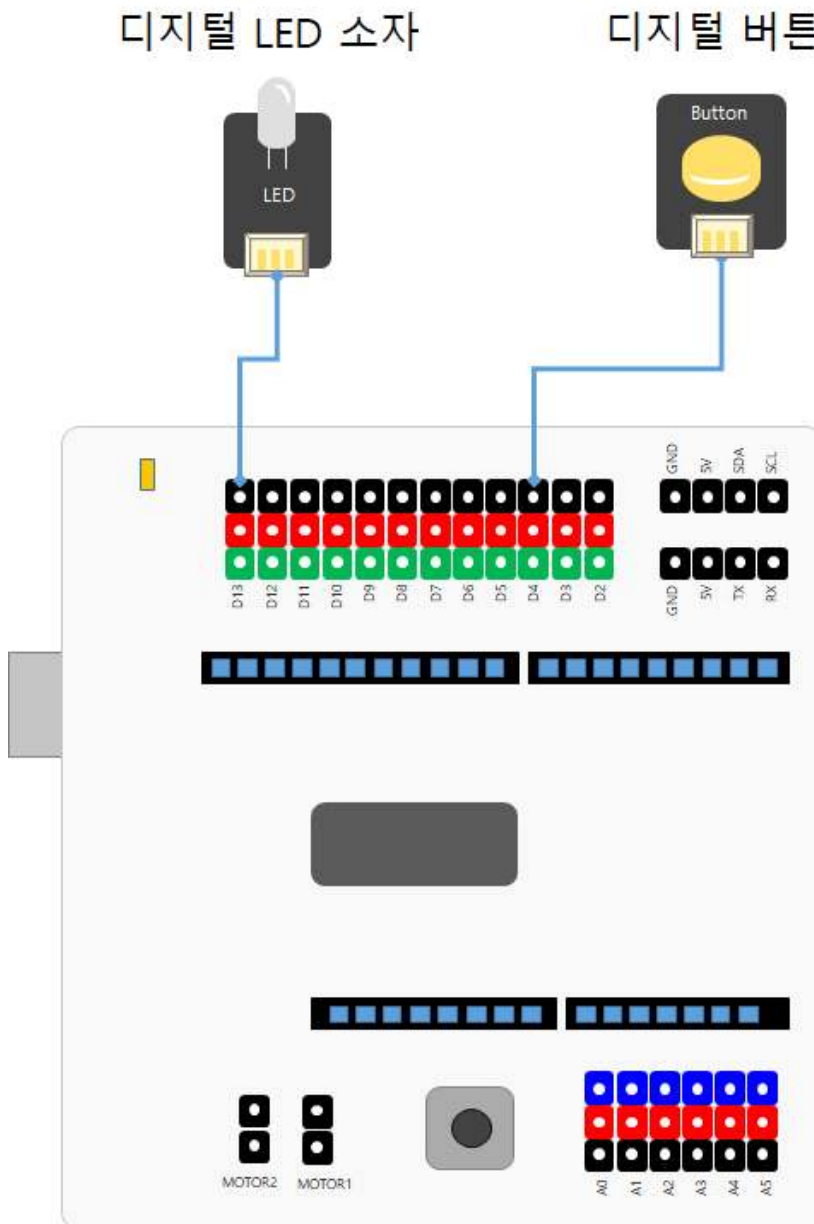
#### 준비물

- 아래 준비물은 실제 아두이노 키트가 있는 경우를 예로 든 것이며, 아래 하드웨어 구성과 동일한 내용을 시뮬레이션으로 진행하게 된다.
- 아두이노 보드, 올인원 쉴드, 디지털 LED 모듈 1개, 디지털 버튼 모듈 1개



H/W 연결하기

- 디지털 버튼을 디지털 4번 핀에 연결한다.
- 디지털 LED 소자를 디지털 13번에 연결한다.



시뮬레이션 상에서의 연결 정보

- 시뮬레이션 상에서는 디지털 LED 소자가 각각 디지털 11번, 12번, 13번에 연결되어 있으며, 버튼은 디지털 4번 ~ 8번에 연결되어 있다.



- 디지털 핀에 연결된 부품
  - 디지털 2번: 서보 모터
  - 디지털 4번 ~ 8번: 버튼 센서
  - 디지털 9번: 스피커
  - 디지털 11번: 초록색 LED
  - 디지털 12번: 노란색 LED
  - 디지털 13번: 빨간색 LED

### 3.2 문자라인 출력하기

문자라인 출력하기

- 아두이노에서 콘솔 화면에 문자열을 한 줄씩 출력하는 명령어는 다음과 같이 PrintLine 또는 Serial.println 명령어를 사용한다.



SPL 스크립트	스케치 코드
PrintLine("Hello")	Serial.println("Hello")
PrintLine(12345)	Serial.println(12345)

화면에 "Hello World" 출력하기

- 시뮬레이션 실행 화면에 Hello World 라는 단어가 한 번만 출력되도록 프로그램을 작성해 본다.



1씩 증가하는 숫자 출력하기

- 화면에 1씩 증가하는 숫자를 출력해 보자.



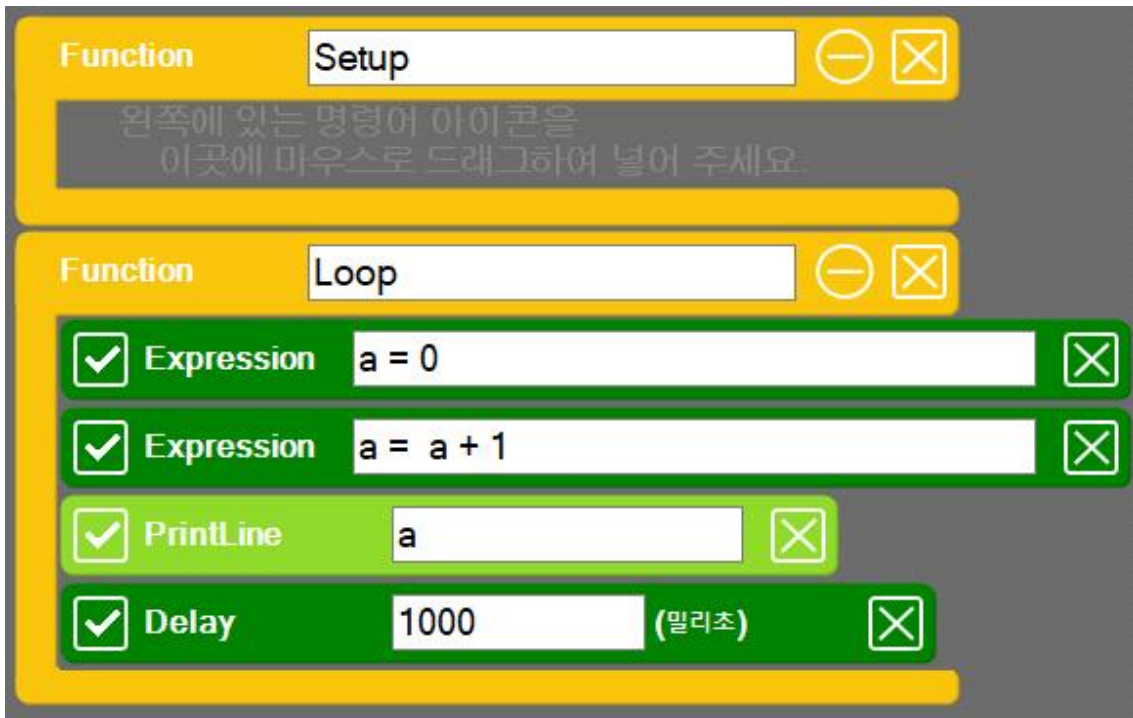
SPL 스크립트	스케치 코드
<pre> void setup() { }  a = 0  void loop() {   a = a + 1   PrintLine(a)   Delay(1000) }         </pre>	<pre> void setup() {   //시리얼 통신 속도를 설정해 준다.   Serial.begin(115200); }  int a = 0;  void loop() {   a = a + 1;   Serial.println(a);   delay(1000); }         </pre>

※ 주의 사항

※ Print 명령어는 같은 줄에 값을 이어서 출력하고 PrintLine 명령어는 값을 출력한 후, 줄을 바꾸게 된다. PrintLine 명령어를 사용해야할 곳에 Print 명령어를 사용하지 않도록 주의를 기울인다.

전역변수와 지역변수

- 위의 프로그램을 아래와 같이 수정하여 실행할 경우, 결과가 어떻게 달라지는 지 예상해 보고 그 이유에 대해 살펴본다.



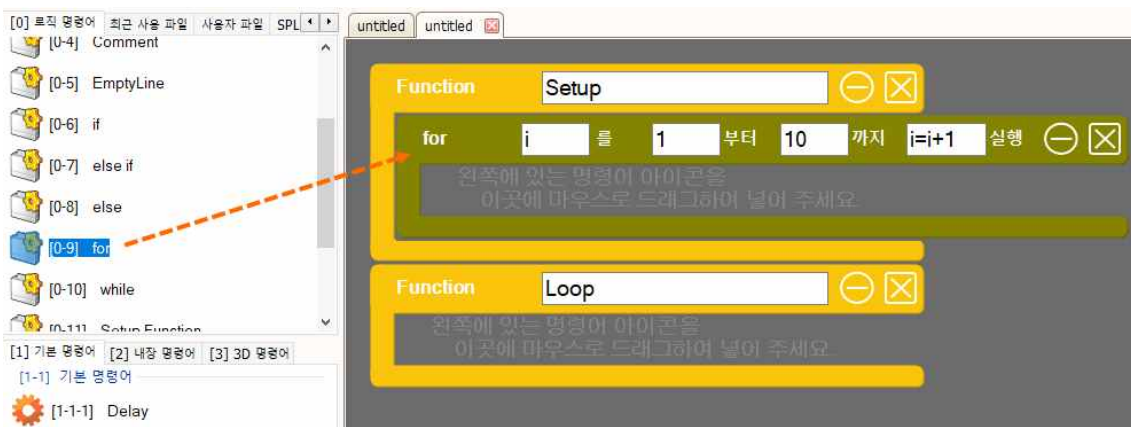
- 변수 a가 함수 바깥쪽에 선언된 경우는 전역변수 라고 부르며, 함수 안에 선언된 경우에는 지역변수라고 부른다.
- 전역변수일 경우에는 함수의 명령어 실행이 끝나도 계속 같이 저장되어 있다.



### 3.3 for 반복문

for 반복문

- 일정한 횟수 만큼 명령어를 반복적으로 실행시키기 위해서는 for 반복문을 사용한다.

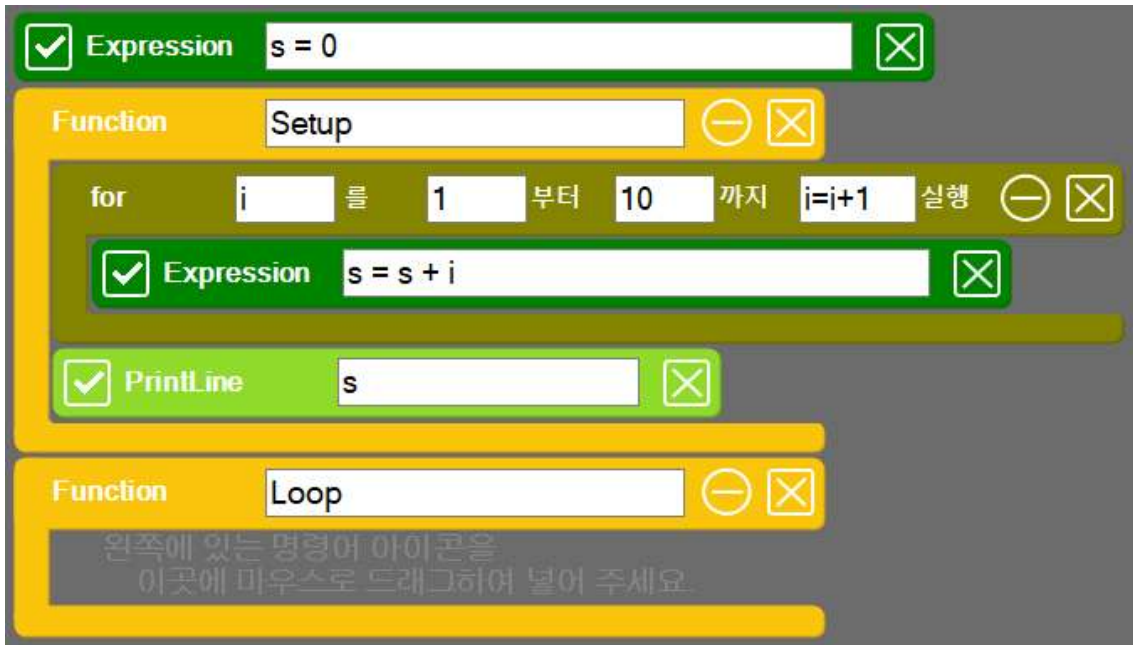


- 위의 예에서는 for 반복문 안에 값이 변수 i 가 정의되어 있다.
- 변수 i는 10이 될 때 까지 1씩 증가( $i = i + 1$ ) 한다.
- 아래의 코드는 증가된 변수 값을 화면에 출력해 보는 예제이다.

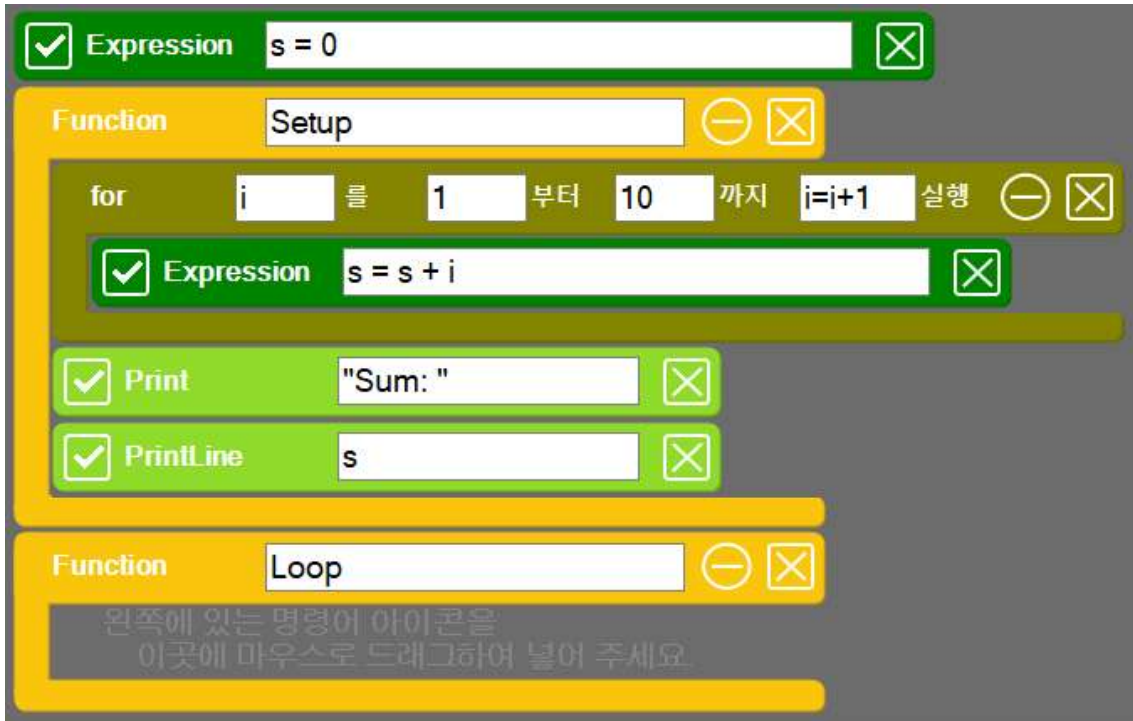


1부터 10까지의 합 구하기

- for 반복문을 이용하여 1부터 10까지의 합을 구하여 화면에 출력하는 기능을 구현해 보기 바란다.



- Print 명령어를 이용하여 결과가 “Sum: 55” 와 같이 표시되도록 명령어를 추가해 본다.



- 위의 예제에서 Print 명령어가 한 역할에 대해 설명해 본다.

### 3.4 구구단 출력하기

구구단 출력하기

- Print 명령어와 PrintLine 명령어를 조합하여 사용하면 화면에 필요한 정보를 보기 좋게 생성하여 표시할 수 있다. Print 명령어는 값을 옆으로 이어서 출력하는 기능이 있으므로, 구구단의 문자열을 프로그램으로 생성하여 표시할 수 있다.



```

7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
    
```

SPL 스크립트	스케치 코드
<pre> void setup() {   //7단을 출력해 본다.   for (i = 1; i &lt; 10; i++)   {     Print("7 x ")     Print(i)     Print(" = ")     PrintLine(i * 7)   } }  void loop() { } </pre>	<pre> void setup() {   Serial.begin(115200);    //7단을 출력해 본다.   for (int i = 1; i &lt; 10; i++)   {     Serial.print("7 x ");     Serial.print(i);     Serial.print(" = ");     Serial.println(i * 7);   } }  void loop() { } </pre>

실습

▶ 다중 for 반복문을 이용하여 구구단 전체를 출력해 본다.

SPL 스크립트	스케치 코드
<pre> void setup() {   for (i = 1; i &lt; 10; i++)   {     for (j = 1; i &lt; 10; j++)     {       Print(i)       Print(" x ")       Print(j)       Print(" = ")       PrintLine(i * j)     }   } }  void loop() { } </pre>	<pre> void setup() {   Serial.begin(115200);    for (int i = 1; i &lt; 10; i++)   {     for (int j = 1; i &lt; 10; j++)     {       Serial.print(i);       Serial.print(" x ");       Serial.print(j);       Serial.print(" = ");       Serial.println(i * j);     }   } }  void loop() { } </pre>

### 3.5 DigitalRead 명령어로 버튼값 읽어오기

DigitalRead

---

- 아두이노의 디지털 핀에서 값을 읽어 오는 명령어는 다음과 같다.

변수 = DigitalRead(핀번호)

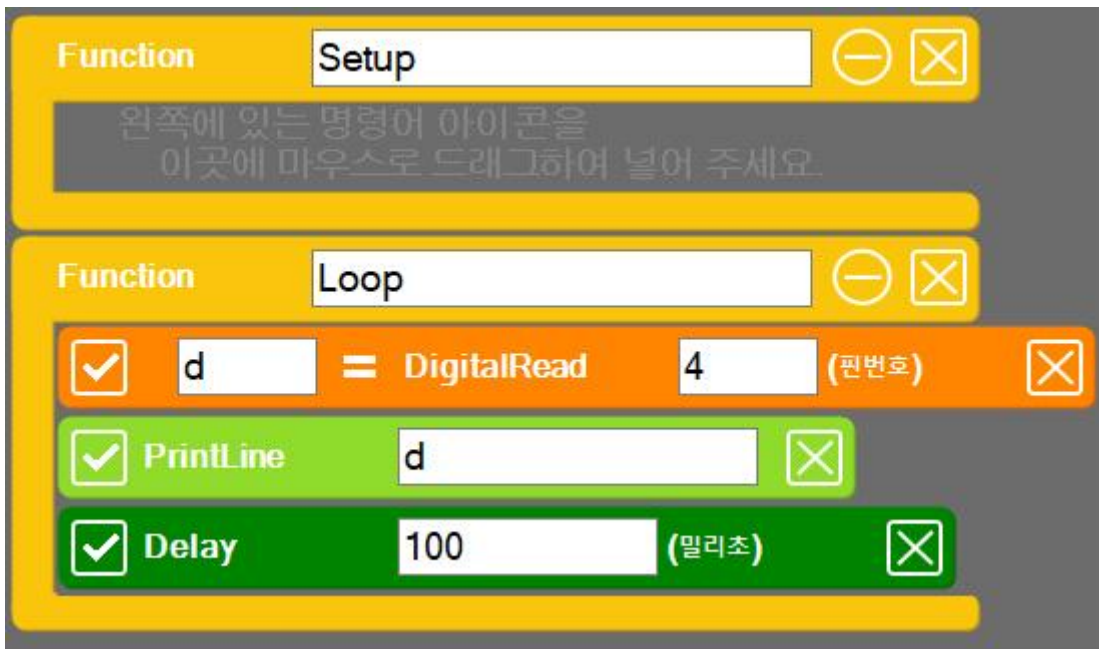


SPL 스크립트	스케치 코드
d2 = DigitalRead(2)	int d2 = digitalRead(2);

- DigitalRead() 함수로 읽은 값은 정수형 변수에 저장되며, 값은 0 (LOW) 또는 1 (HIGH) 중에 하나가 된다.

디지털 버튼값 출력하기

- 디지털 4번 핀에 연결되어 있는 디지털 버튼의 값을 읽어 와서 화면에 출력하는 코드이다. 핀 번호와 변수 이름을 확인한 후 아래의 코드를 작성해 본다.



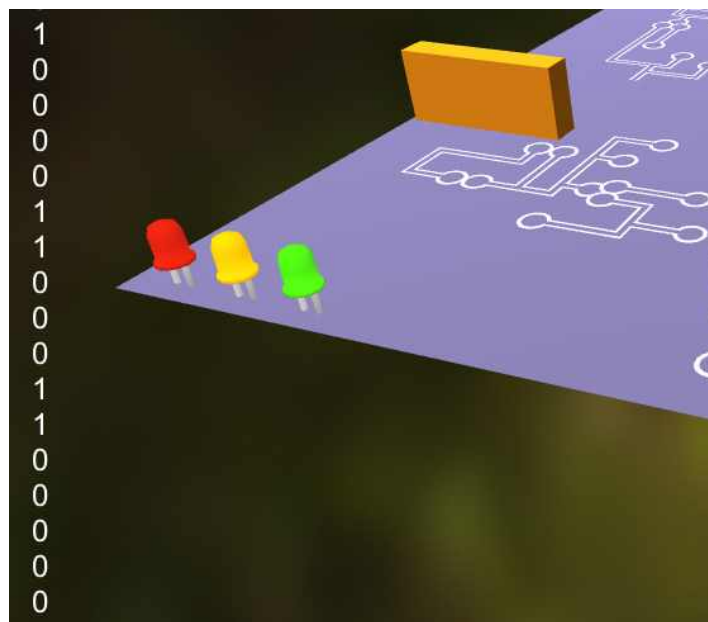
SPL 스크립트	스케치 코드
<pre>void setup() { }  void loop() {   d = DigitalRead( 4 )   PrintLine(d)   Delay(100) }</pre>	<pre>void setup() {   pinMode(2, INPUT);   Serial.begin(115200); }  void loop() {   int d = digitalRead( 4 );   Serial.println(d);   delay(100); }</pre>



- 실행 창에서 버튼1을 클릭하여 결과 값이 어떻게 달라지는 지 확인해 본다.



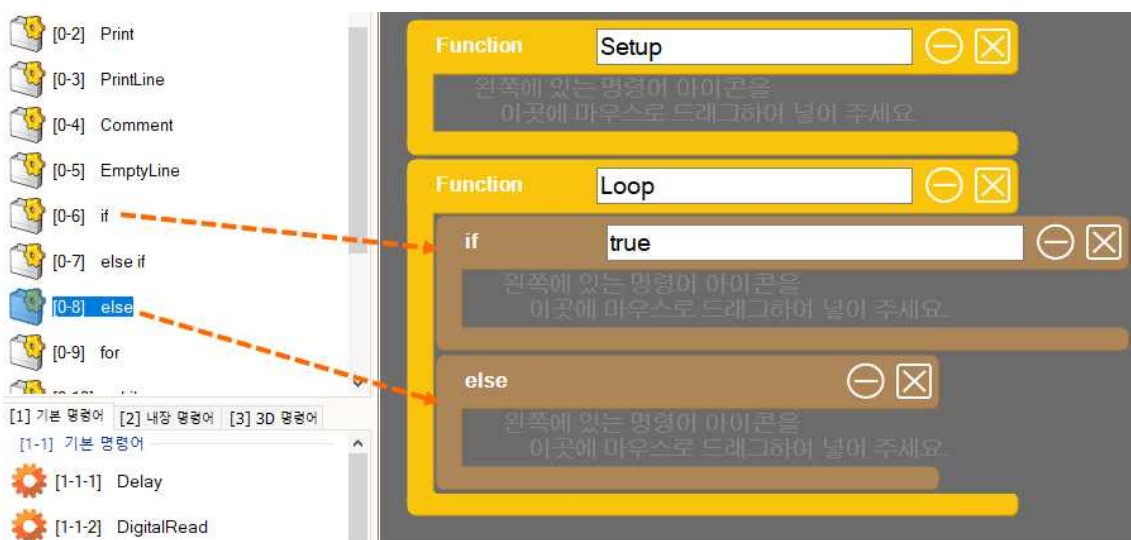
- 콘솔창에 표시되는 값을 확인해 보면, HIGH나 LOW가 아닌 0 또는 1 숫자로 표시되는 것에 유의한다.



### 3.6 버튼으로 LED 켜기

#### if 조건문

- C언어에서 if 조건비교 명령어는 괄호 안에 있는 조건이 참이면 if 블록 안에 있는 명령어를 실행하게 된다.



```

if (조건이 true 이면)
{
    //이곳에 있는 명령어들이 실행된다.
}
    
```

예)

```

if ( d2 == HIGH )
{
    PrintLine("Button Pressed")
}
    
```

- 조건이 만족하지 않는 경우에 명령어를 실행하려면 else 블록을 추가한다.

버튼의 상태 출력하기

- 버튼의 눌림 상태를 if 및 else 명령어로 체크하여 상태를 화면에 출력해 본다.



```

if ( d2 == HIGH )
{
    PrintLine("Button Pressed")
}
else
{
    PrintLine("Button Up")
}
    
```

if 조건문으로 LED 켜기

- 버튼의 상태를 0.1초 간격으로 읽어와서 버튼의 상태가 HIGH이면 13번 LED를 켜고 그렇지 않으면 13번 LED를 끄는 코드를 작성해 보자.

The screenshot shows the HelloApps IDE interface. It features two function blocks: 'Setup' and 'Loop'. The 'Loop' function contains the following logic:

- DigitalRead**: Reads the state of pin 4 into variable 'd'.
- if**: Checks if 'd == HIGH'.
  - if true**: Executes **DigitalWrite** on pin 13 with value HIGH.
  - else**: Executes **DigitalWrite** on pin 13 with value LOW.
- Delay**: Pauses the execution for 100 milliseconds.

- 아래의 예제에서 if 조건문에 유의하여 코드를 작성해 본다.

SPL 스크립트	스케치 코드
<pre> void setup() { }  void loop() {     d2 = DigitalRead(2)      if ( d2 == HIGH )     {         DigitalWrite(13, HIGH)     }     else     {         DigitalWrite(13, LOW)     }      Delay(100) }         </pre>	<pre> void setup() {     pinMode(2, INPUT);     pinMode(13, OUTPUT); }  void loop() {     int d2 = digitalRead(2);      if ( d2 == HIGH )     {         digitalWrite(13, HIGH);     }     else     {         digitalWrite(13, LOW);     }      delay(100); }         </pre>

### 3.7 버튼을 누르고 있는 횟수 측정하기

전역변수

- 버튼이 눌러져 있는 횟수를 저장하기 위해 전역변수를 정의한다. 함수 내부에 정의된 변수는 로컬변수 또는 지역변수라 하고, 함수 외부에 정의된 변수는 글로벌변수 또는 전역변수라고 한다.
- 전역변수는 프로그램 전체에 영향을 미치지만 지역변수는 함수 안에서만 사용할 수 있다.



SPL 스크립트	스케치 코드
<pre> cnt = 0  void setup() { }  void loop() {     d2 = DigitalRead(2)      if ( d2 == HIGH )     {         cnt = cnt + 1         PrintLine(cnt)     }      Delay(100) } </pre>	<pre> int cnt = 0;  void setup() {     pinMode(2, INPUT);     Serial.begin(115200); }  void loop() {     int d2 = digitalRead(2);      if ( d2 == HIGH )     {         cnt = cnt + 1;         Serial.println(cnt);     }      delay(100); } </pre>

- 위의 코드에서 cnt = 0 선언 명령어를 loop() 함수 안으로 옮길 경우 결과가 어떻게 달라지는 지 확인해 본다.

### 3.8 절전기능 후레쉬 만들기

#### 설계하기

- 버튼이 눌러져 있는 횟수는 cnt라는 이름의 전역변수에 저장되어 있다.
- LED를 버튼을 누르고 있는 시간에 비례하여 켜지도록 기능을 구현하려고 한다. 예를 들어 버튼을 1초간 누르고 있으면 LED는 5초간 켜지고, 버튼을 10초간 누르고 있으면 LED가 50초 동안 켜져 있다가 꺼지도록 하려고 한다.
- 필요한 기능을 설계해 본다.
- 버튼이 눌릴 때 마다 cnt 변수의 값을 5씩 증가시켜 준다.
- cnt 변수의 값이 0보다 크면 cnt 변수의 값을 1씩 감소시켜 주고 LED를 켜준다.
- cnt 변수의 값이 0과 같으면 LED를 꺼준다.



구현하기

- 원하는 기능을 구현하기 위해 프로그램을 설계한 후 아래의 코드를 작성해 본다.

The screenshot shows the HelloApps IDE interface with the following code blocks:

- Function Setup**:
  - Expression: `cnt = 0`
- Function Loop**:
  - Expression: `d = DigitalRead 4 (핀번호)`
  - if `d == HIGH`:
    - Expression: `cnt = cnt + 5`
  - if `cnt > 0`:
    - Expression: `cnt = cnt - 1`
    - DigitalWrite: `13 (핀번호) HIGH (값)`
  - else:
    - DigitalWrite: `13 (핀번호) LOW (값)`
  - Delay: `100 (밀리초)`

SPL 스크립트	스케치 코드
<pre> cnt = 0  void setup() { }  void loop() {     d2 = DigitalRead(2)      if ( d2 == HIGH )         cnt = cnt + 5      if (cnt &gt; 0)     {         cnt = cnt - 1          DigitalWrite(13, HIGH)     }     else         DigitalWrite(13, LOW)      Delay(100) } </pre>	<pre> int cnt = 0;  void setup() {     pinMode(2, INPUT);     pinMode(13, OUTPUT); }  void loop() {     int d2 = digitalRead(2);      if ( d2 == HIGH )         cnt = cnt + 5;      if (cnt &gt; 0)     {         cnt = cnt - 1;          digitalWrite(13, HIGH);     }     else         digitalWrite(13, LOW);      delay(100); } </pre>

실습

- ▶ 버튼을 1초간 누르고 있으면 LED가 10초간 켜지고, 버튼을 10초간 누르고 있으면 LED가 100초 동안 켜져 있다가 꺼지도록 기능을 수정해 본다.
- ▶ LED가 꺼지기 10초 전에 LED를 점멸하여 10초 후에 LED가 꺼진다는 정보를 사용자에게 알리고자 한다. 어떠한 기능을 추가해야 할지 생각해 보고 기능을 완성해 본다.

기능 개선 사항
<ul style="list-style-type: none"> <li>■ LED가 꺼지기 10초전에 점멸로 정보를 주어 사용자가 다시 버튼을 눌러 LED가 켜져 있는 시간을 늘릴 수 있도록 기능을 개선시켜 본다.</li> </ul>