
아두이노 시뮬레이션 프로그래밍

v1.0

김영준 저

공학박사, 목원대학교 겸임교수
前 Microsoft 수석연구원

헬로앱스

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

06 스마트 헤드라이트

학습 목표

- 아두이노의 아날로그 명령어를 이해하고 활용할 수 있다.
- 조도센서의 작동원리를 이해하고 활용할 수 있다.
- 어두워 지면 저절로 켜지는 장치를 구현하여 자동차의 헤드라이트와 같은 분야에 응용할 수 있다.
- 아두이노의 PWM 핀을 활용하여 LED의 밝기를 조절할 수 있다.

실습 개요

- 아두이노의 아날로그 센서 값을 읽어와서 화면에 표시한다.
- 임계치를 결정하여 어두워 지면 저절로 켜지는 장치를 구현하다.
- LED의 밝기를 조절해 본다.

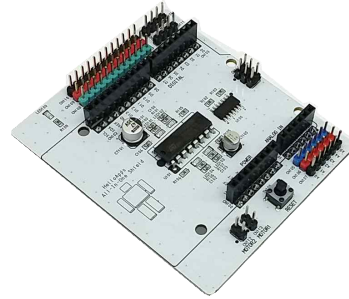
6.1 준비하기

준비물

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



아두이노 우노보드



올인원 쉴드



디지털 LED 모듈



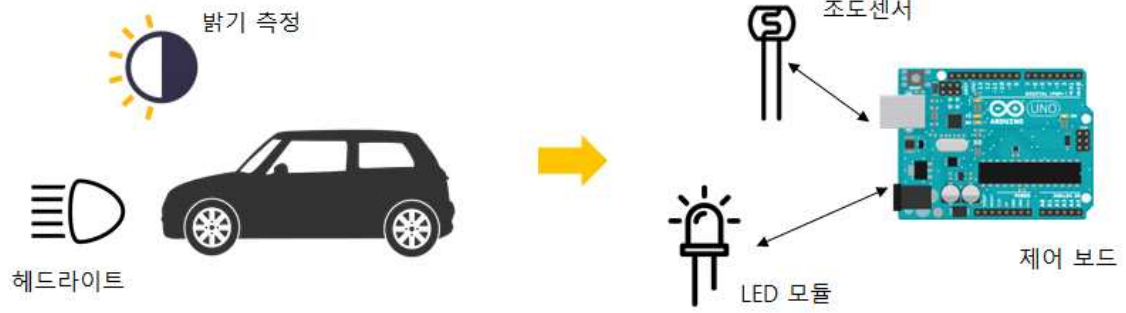
아날로그 조도 센서

시뮬레이션 상에서의 연결 정보

- 시뮬레이션 상에서는 디지털 LED 소자가 각각 디지털 11번, 12번, 13번에 연결되어 있으며, 조도센서는 아날로그 5번에 연결되어 있다.



- 디지털 핀에 연결된 부품
 - 디지털 11번: 초록색 LED
 - 디지털 12번: 노란색 LED
 - 디지털 13번: 빨간색 LED
 - 아날로그 5번: 조도센서



6.2 조도 센서값 확인하기

아날로그 센서값 읽기

- 아날로그 조도 센서를 활용하기 위해서는 첫 번째 절차로서 아날로그 센서인 조도 센서의 값을 확인해 보는 단계이다. 아날로그 센서는 다음과 같이 1024 크기의 값의 범위를 가진다.

아날로그 센서 값의 범위
0 ~ 1023

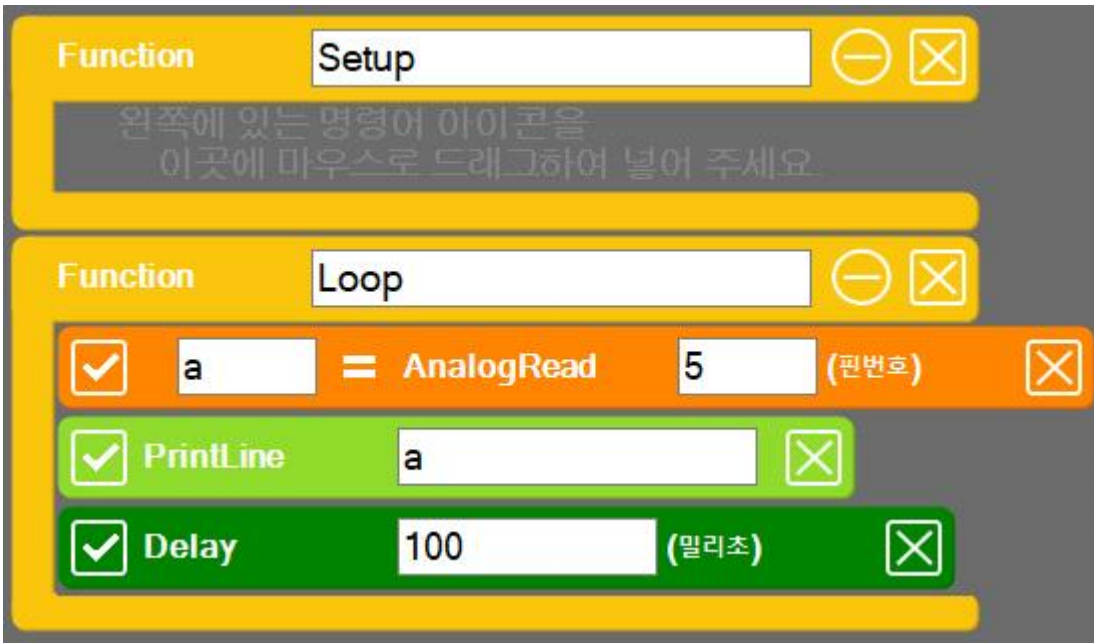
- 아날로그 센서의 값을 읽기 위해서는 다음과 같이 AnalogRead 명령어를 이용한다.



SPL 스크립트	스케치 코드
a = AnalogRead(핀번호)	int a = analogRead(핀번호);

조도 센서값 읽기

- 조도센서가 어떠한 특성으로 작동하는 지 확인하기 위해 다음의 코드를 실행하여 결과를 기록해 본다.



SPL 스크립트	스케치 코드
<pre>void setup() { } void loop() { //아날로그 5번 핀에서 값을 읽어서 //변수에 저장한다. a = AnalogRead(5) //한 라인씩 값을 콘솔에 출력한다. PrintLine(a) }</pre>	<pre>void setup() { Serial.begin(115200); } void loop() { //아날로그 0번 핀에서 값을 읽어서 //변수에 저장한다. int a = analogRead(5); //한 라인씩 값을 콘솔에 출력한다. Serial.println(a); }</pre>

<pre>//0.1초씩 기다린다. Delay(100) }</pre>	<pre>//0.1초씩 기다린다. delay(100); }</pre>
---------------------------------------	--

- 위의 코드를 실행한 후, 오른쪽 화면 아래에 있는 슬라이드 바를 움직여 밝기를 조절해 본다.



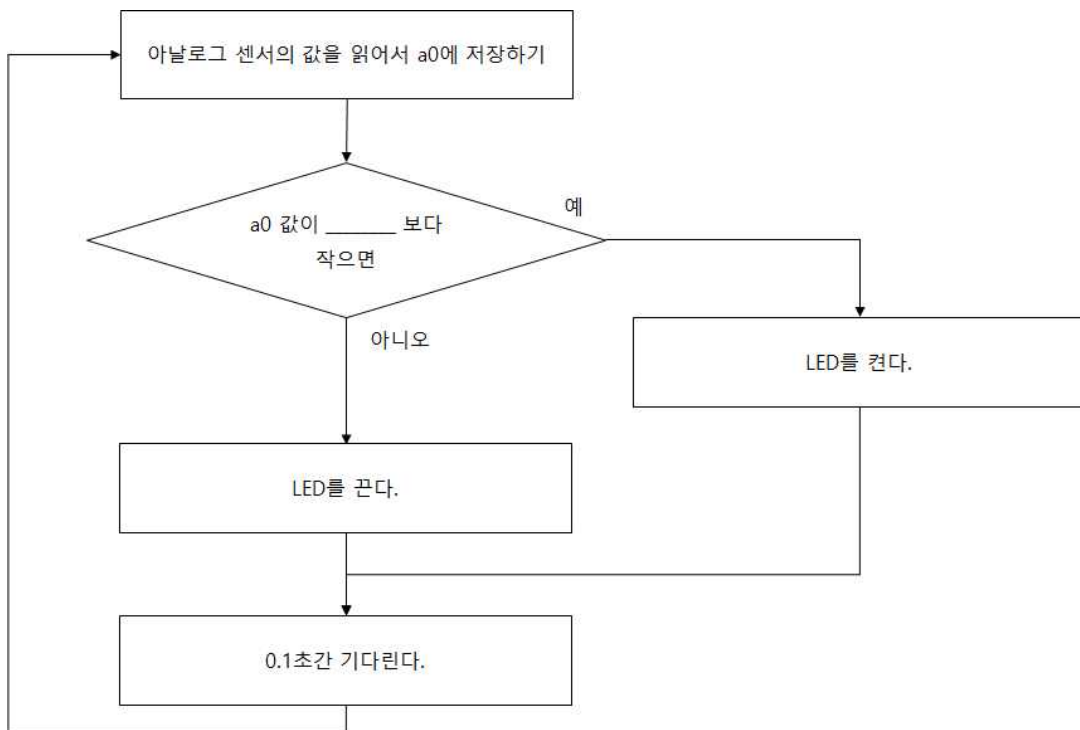
슬라이드 바를 좌우로 움직여 밝기를 변화시켜 본다.

- 밝기가 밝을 수록 출력되는 숫자 값이 크고, 어두워 질수록 값이 작아지는 것을 볼 수 있다.

6.3 헤드라이트가 켜지는 조건

LED가 켜지기 위한 조건

- 자동차의 헤드라이트 역할은 아두이노에서 LED가 대신한다. 본 실습에서는 어두워지는 조건을 만족하면 LED가 자동으로 켜지도록 기능을 구현하려고 한다.
- 아래의 순서도는 어두워지는 조건을 정의하는 것으로서, 아날로그 조도 센서의 값이 일정한 값 이하이면 LED가 켜지도록 하는 과정을 정의한 것이다.
- 빈 칸에 어떠한 숫자가 들어가야 할지 생각해 보고 값을 입력해 본다.



6.4 헤드라이트를 자동으로 켜는 기능 구현하기

LED를 자동으로 켜기

- 자동차의 헤드라이트를 외부 밝기에 따라 어두우면 자동으로 켜기 위한 코드를 다음과 같이 구현해 보자.

```
Function Setup
  왼쪽에 있는 명령어 아이콘을
  이곳에 마우스로 드래그하여 넣어 주세요.

Function Loop
  a = AnalogRead 5 (핀번호)
  PrintLine a
  if a < 600
    DigitalWrite 13 (핀번호) HIGH (값)
  else
    DigitalWrite 13 (핀번호) LOW (값)
  Delay 100 (밀리초)
```

SPL 스크립트	스케치 코드
<pre> void setup() { } void loop() { a = AnalogRead(5) PrintLine(a) if (a0 < 600) { //이곳에 LED를 켜는 명령어를 추가한다. DigitalWrite(13, HIGH) } else { //이곳에 LED를 끄는 명령어를 추가한다. DigitalWrite(13, LOW) } Delay(100) } </pre>	<pre> void setup() { Serial.begin(115200); } void loop() { int a = analogRead(5); Serial.println(a); if (a0 < 600) { //이곳에 LED를 켜는 명령어를 추가한다. digitalWrite(13, HIGH); } else { //이곳에 LED를 끄는 명령어를 추가한다. digitalWrite(13, LOW); } delay(100); } </pre>

※ 주의 사항

※ 조도 센서는 센서마다 약간 씩 차이가 날 수 있다. LED를 켜기 위한 기준 값은 센서 값을 확인해 보고, 최대 값과 최소 값의 중간 정도 값으로 설정해 보도록 한다.

6.5 밝기가 조절되는 헤드라이트

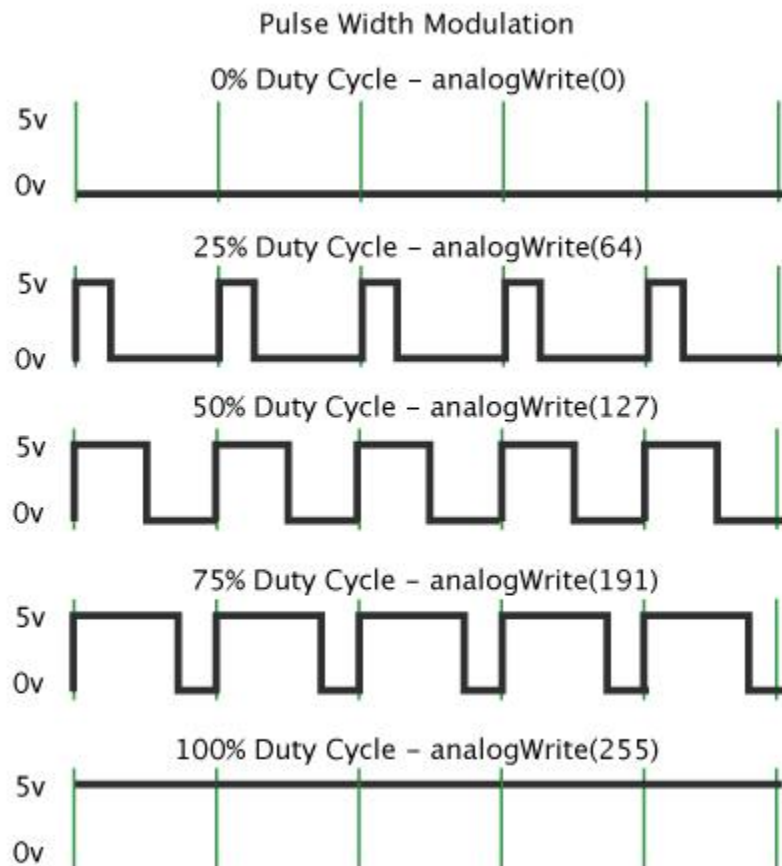
디지털 PWM 핀

- 이번에는 헤드라이트의 밝기를 조절하는 기능을 구현해 본다. 아두이노의 디지털 핀 중에서 3, 5, 6, 9, 10, 11핀의 경우에는 다른 핀들과 달리 256 단계로 출력값을 설정하는 기능이 가능하다. 이 핀들은 PWM (Pulse Width Modulation) 핀이라고 부른다.
- 아두이노 보드의 디지털 핀 중에서 PWM 핀에는 ~ 표시가 붙어 있다.



PWM의 작동 원리

- 아두이노 보드의 PWM은 필요한 전압을 0과 1의 간격을 조절하여 생성한다.
- 0 ~ 255 사이의 값을 PWM 핀에 입력하면 0V ~ 5V 사이의 가상 전압이 만들어 진다.



AnalogWrite

- PWM 핀에 값을 쓸 때에는 AnalogWrite 명령어를 사용한다.



SPL 스크립트	스케치 코드
AnalogWrite(핀번호, 값)	analogWrite(핀번호, 값);

- 기존의 DigitalWrite 명령어는 0과 1 또는 HIGH와 LOW 상태로만 값을 출력시킬 수 있지만, AnalogWrite 명령어를 사용하면 0 ~ 255 사이의 값을 디지털 핀에 출력시킬 수 있다.

AnalogWrite 명령어 사용법
<input type="checkbox"/> AnalogWrite(11, 0) : LED를 끄
<input type="checkbox"/> AnalogWrite(11, 255) : LED를 가장 밝게 켜
<input type="checkbox"/> AnalogWrite(11, 120) : 중간 밝기로 LED를 켜

※ 주의 사항

- ※ AnalogWrite 명령어는 비록 Analog라는 단어가 붙어 있긴 하지만 실질적으로는 디지털 명령어이다. 기존 AnalogRead 명령어에서 읽은 값은 0 ~ 1023 사이의 값을 가지는 반면에 AnalogWrite 명령어에서는 256 단계의 값만 사용할 수 있다.

구현하기

- 디지털 11번이 PWM 핀에 해당하므로 11번 핀에 연결된 LED를 제어해 본다.
- 아래의 코드는 LED의 밝기를 256단계로 변화를 주는 과정을 보여준다.



SPL 스크립트	스케치 코드
<pre>void setup() { } void loop() { for (i = 0; i < 256; i++) { //이곳에 LED의 밝기를 256단계로 //변화시키는 명령어를 추가한다. AnalogWrite(11, i) } }</pre>	<pre>void setup() { } void loop() { for (int i = 0; i < 256; i++) { //이곳에 LED의 밝기를 256단계로 //변화시키는 명령어를 추가한다. analogWrite(11, i); } }</pre>

<pre> Delay(10) } } </pre>	<pre> delay(10); } } </pre>
------------------------------------	-------------------------------------

- for 반복문은 반복되는 횟수를 지정할 수 있는 반복문으로서, 반복 조건문에 사용되는 변수값이 증가 및 감소 조건에 따라 값이 변화된다.

실습

- ▶ LED의 밝기가 서서히 밝아졌다가 서서히 어두워지도록 기능을 수정해 본다.

6.6 외부 밝기에 따라 자동 반응하는 헤드라이트

자동 감응 헤드라이트

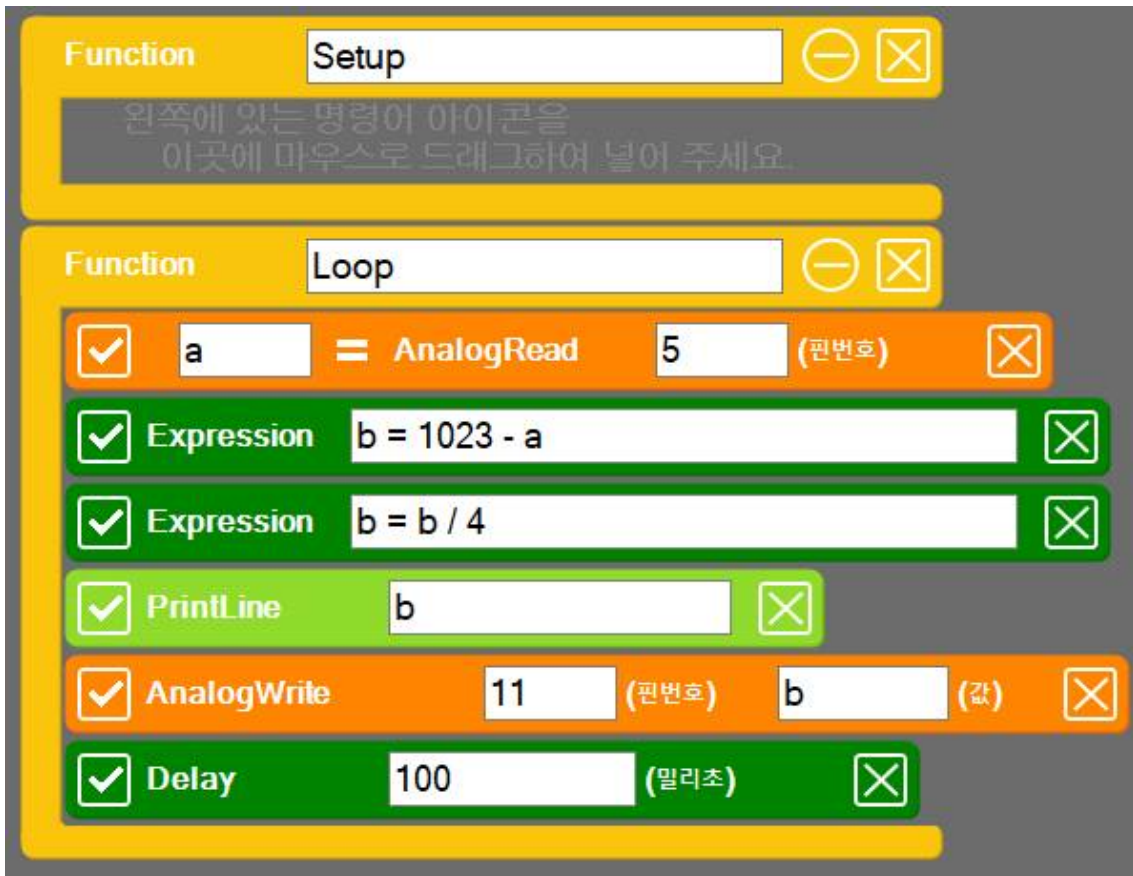
- 이전 활동에서 실습한 내용을 기반으로 외부의 밝기에 따라 차량의 헤드라이트 밝기가 자동으로 조절되는 기능을 작성해 보자.
- 다음과 같은 기능에 대해 생각해 보고, 필요한 로직을 완성해 본다.

외부 조도에 반응하는 헤드라이트

□외부의 밝기가 밝으면 헤드라이트의 밝기가 약해지고, 외부가 어두워 질수록 헤드라이트의 밝기는 세진다.

구현하기

- 아래의 코드는 조도 센서의 값을 LED의 밝기 값으로 변환하여 조도 센서 값에 따라 상대적으로 LED의 출력 값이 조절되는 기능을 추가한 과정을 보여준다.



SPL 스크립트	스케치 코드
<pre> void setup() { } void loop() { a = AnalogRead(5) //조도 센서값을 LED 출력값으로 변환한다. b = 1023 - a //값을 4로 나누어 0 ~ 255 사이의 //값으로 변환한다. b = b / 4 PrintLine(b) AnalogWrite(11, b) Delay(100) } </pre>	<pre> void setup() { Serial.begin(115200); } void loop() { int a = analogRead(5); //조도 센서값을 LED 출력값으로 변환한다. int b = 1023 - a; //값을 4로 나누어 0 ~ 255 사이의 //값으로 변환한다. b = b / 4; Serial.println(b); analogWrite(11, b); delay(100); } </pre>

6.7 매핑 함수

map() 함수

- 이전 예제에서는 읽은 센서의 값을 PWM에 전달하기 위한 용도로 변환하기 위해 2번 정도 계산 과정을 거쳤다. 그리 어려운 과정은 아니지만, 수식의 내용을 한 번에 이해하기 어려운 문제점이 발생할 수 있다.
- 아두이노에서는 값을 다른 범위로 한번에 변환시킬 수 있는 유용한 함수를 제공한다. 함수 이름은 map() 으로서 다음과 같이 사용할 수 있다.

변환값 = map(원본 값, 원본 시작값, 원본 끝 값, 변환 대상 시작값, 변환 대상 끝값)

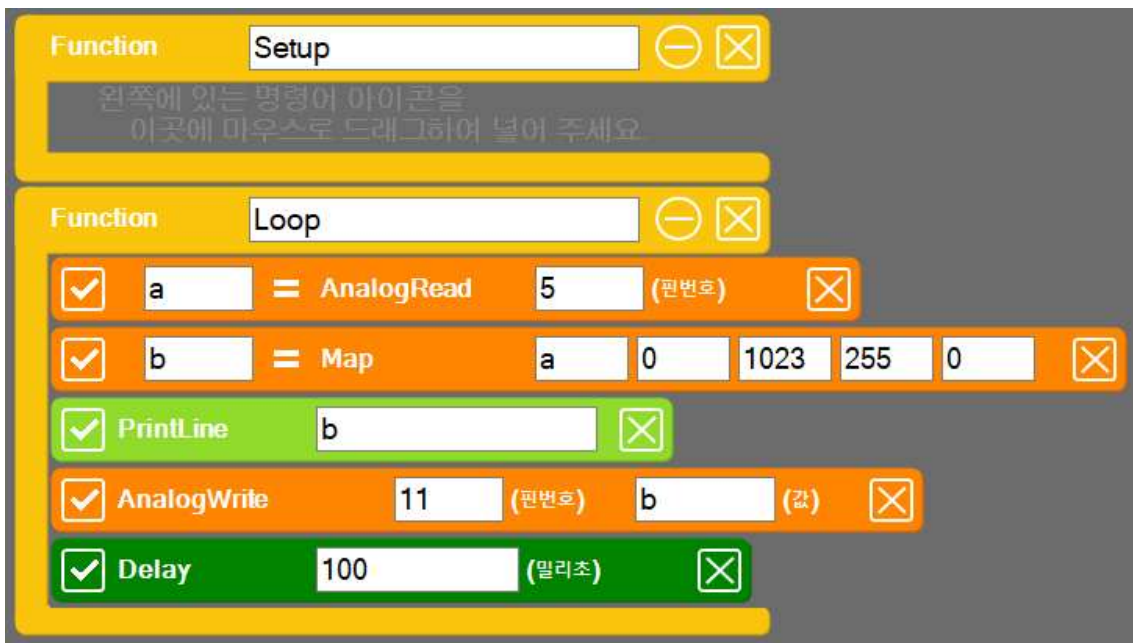
- 이전 예제에서 변환되는 과정을 map 함수를 이용하면 다음과 같이 적용할 수 있다.

<p>map 함수를 사용하지 않을 경우</p>	<pre>//조도 센서값을 LED 출력값으로 변환한다. b = 1023 - a //값을 4로 나누어 0 ~ 255 사이의 //값으로 변환한다. b = b / 4</pre>
<p>map 함수를 사용할 경우</p>	<pre>b = map(a, 0, 1023, 255, 0)</pre>

- a 값은 0 ~ 1023 범위를 가지는 값인데, 이를 255 ~ 0 범위로 값을 매핑 시키라는 의미입니다. 값의 범위가 작아지면서 반대로 뒤집히는 효과를 한 번에 얻을 수 있습니다.

실습

▶ map() 함수를 이용하여, 이전 활동의 예제를 수정해 본다.



```

void setup()
{
}

void loop()
{
    a = AnalogRead(5)
    b = map(a, 0, 1023, 255, 0)
    PrintLine(b)
    AnalogWrite(11, b)
    Delay(100)
}
    
```

▶ map() 함수를 활용하는 다양한 사례를 제시해 본다.