

# 아두이노 프로그래밍

## 2일차 - Part1 아날로그 명령어

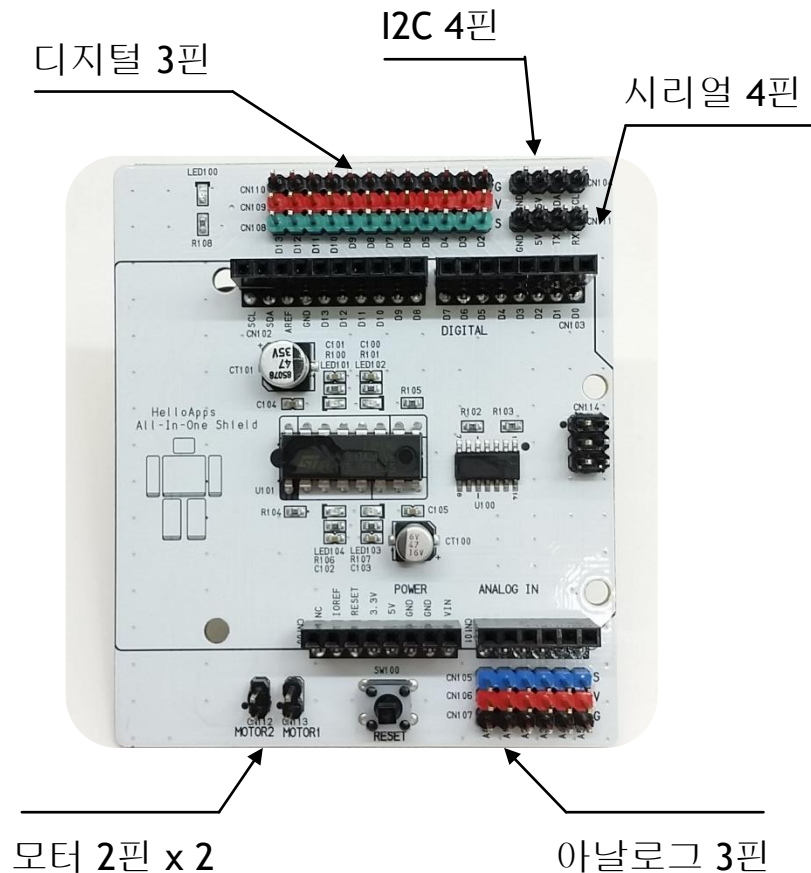
강사: 김영준 헬로앱스 대표  
헬로앱스 ([www.helloapps.co.kr](http://www.helloapps.co.kr))

# 아날로그 센서의 연결

# 아두이노 올인원 센서 쉴드

아날로그 센서

## ▶ 초보자용 아두이노 올인원 센서 쉴드

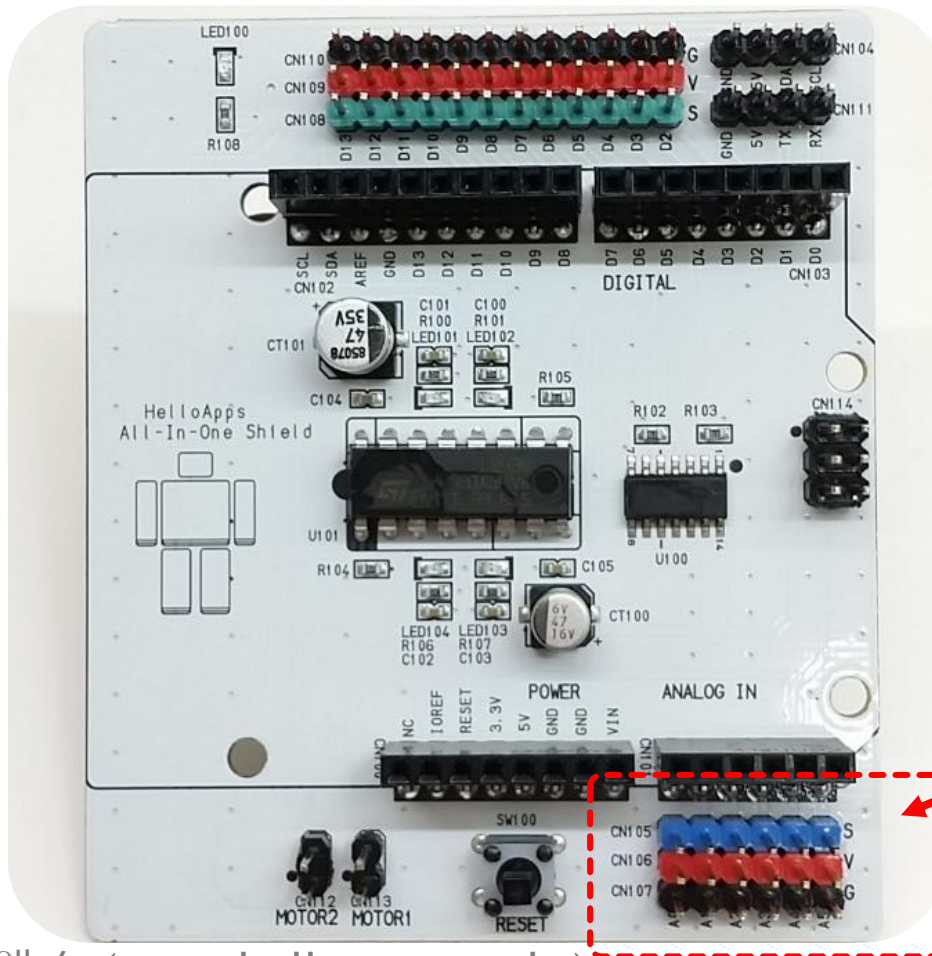


브레드 보드가 필요없는  
초보자용 아두이노 보드

# 아날로그 센서 연결하기

센서 연결하기

## ▶ 아날로그 핀



아날로그 핀에는  
0번 ~ 5번 까지  
번호가 표시되어 있습니다.  
(A0 ~ A5)

아날로그 핀에 부품을 연결할  
때에는 핀 번호를 확인해야  
합니다.

아날로그 부품은 아날로그  
핀에 연결합니다.

# 아날로그 센서 연결하기

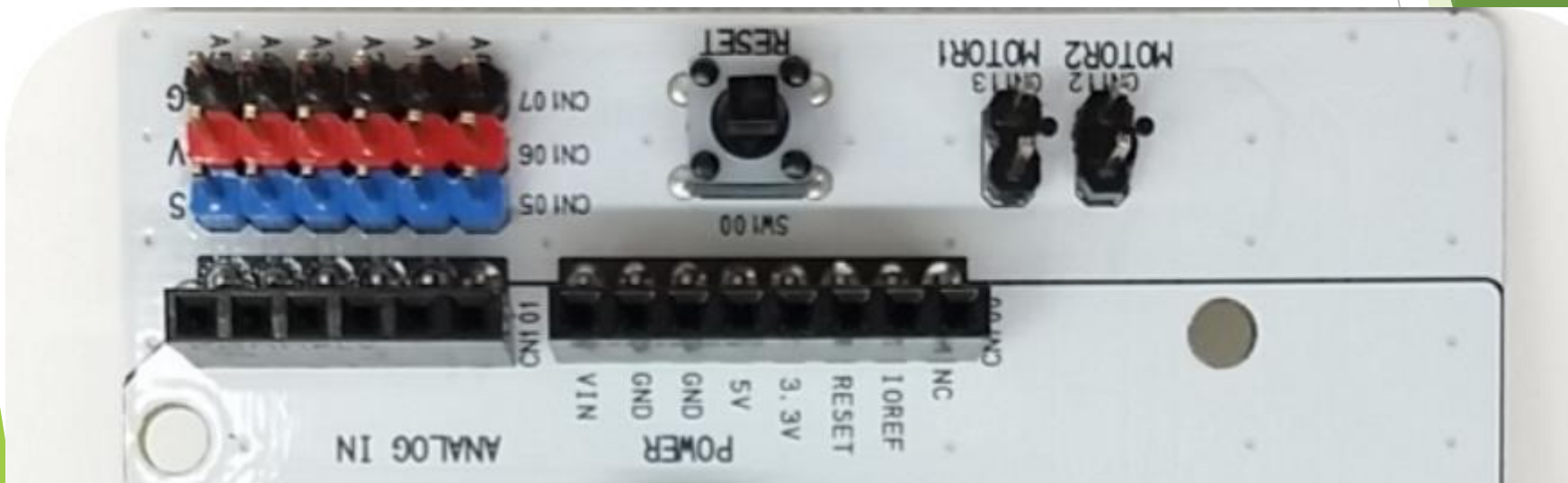
센서 연결하기

## ▶ 아날로그 핀

5번  
(A5)



0번  
(A0)



# 아날로그 센서 연결하기

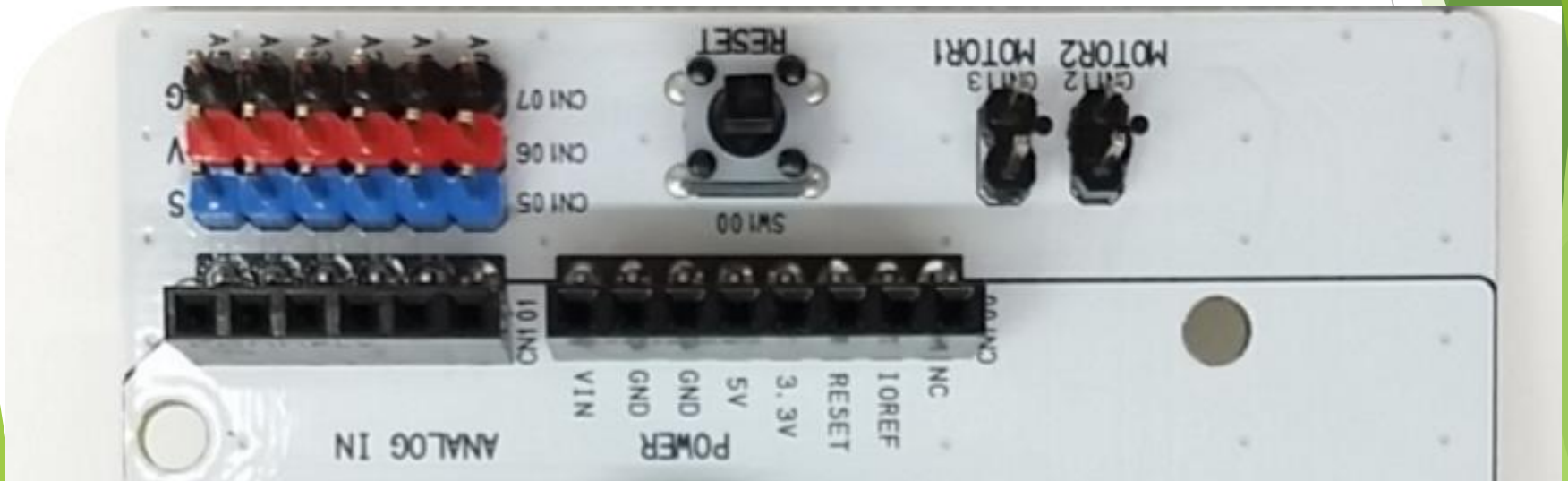
센서 연결하기

## ▶ 아날로그 핀

0번  
(A0)



아날로그 핀에 부품들을 연결할 때에는  
0번 핀부터 연결합니다.



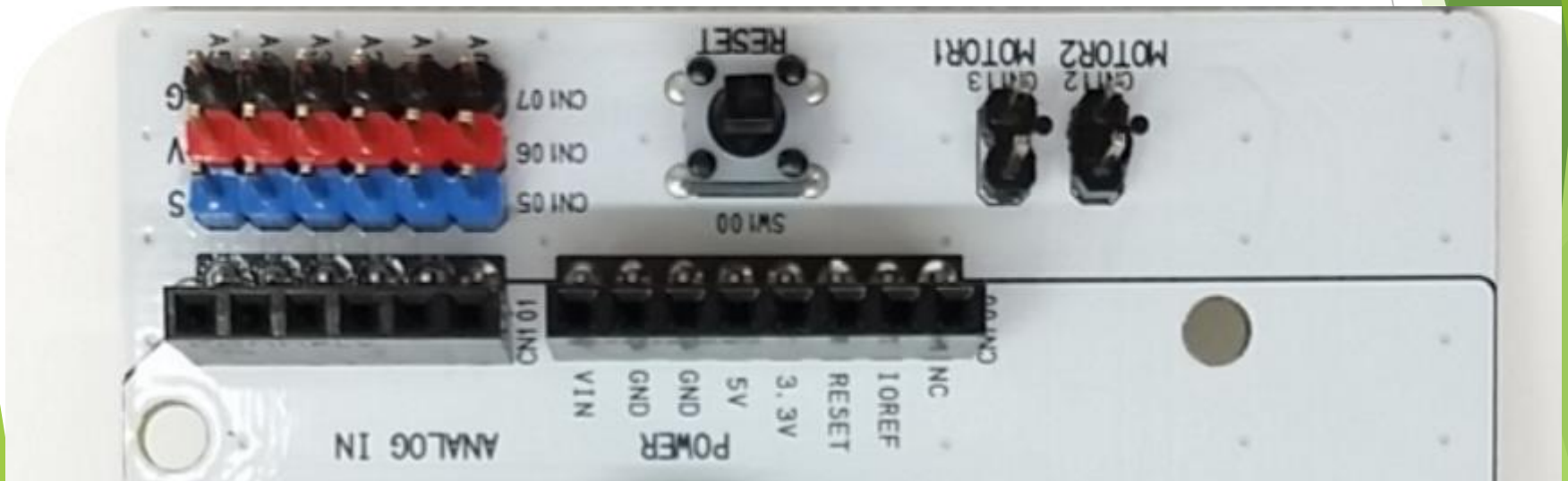
# 아날로그 센서 연결하기

센서 연결하기

## ▶ 아날로그 핀

아날로그 4번과 5번 핀은 다른 부품(I2C 핀 사용)과 충돌되므로, LCD나 LED 체인등 I2C 4핀에 부품이 연결되어 있는 경우에는 사용할 수 없습니다.

5번  
4번  
↓ ↓



# 조도 센서 연결 실습



# 조도센서 연결 실습

실습

- ▶ 실습) 아래의 조도센서를 아날로그 0번 핀에 연결해 봅니다.

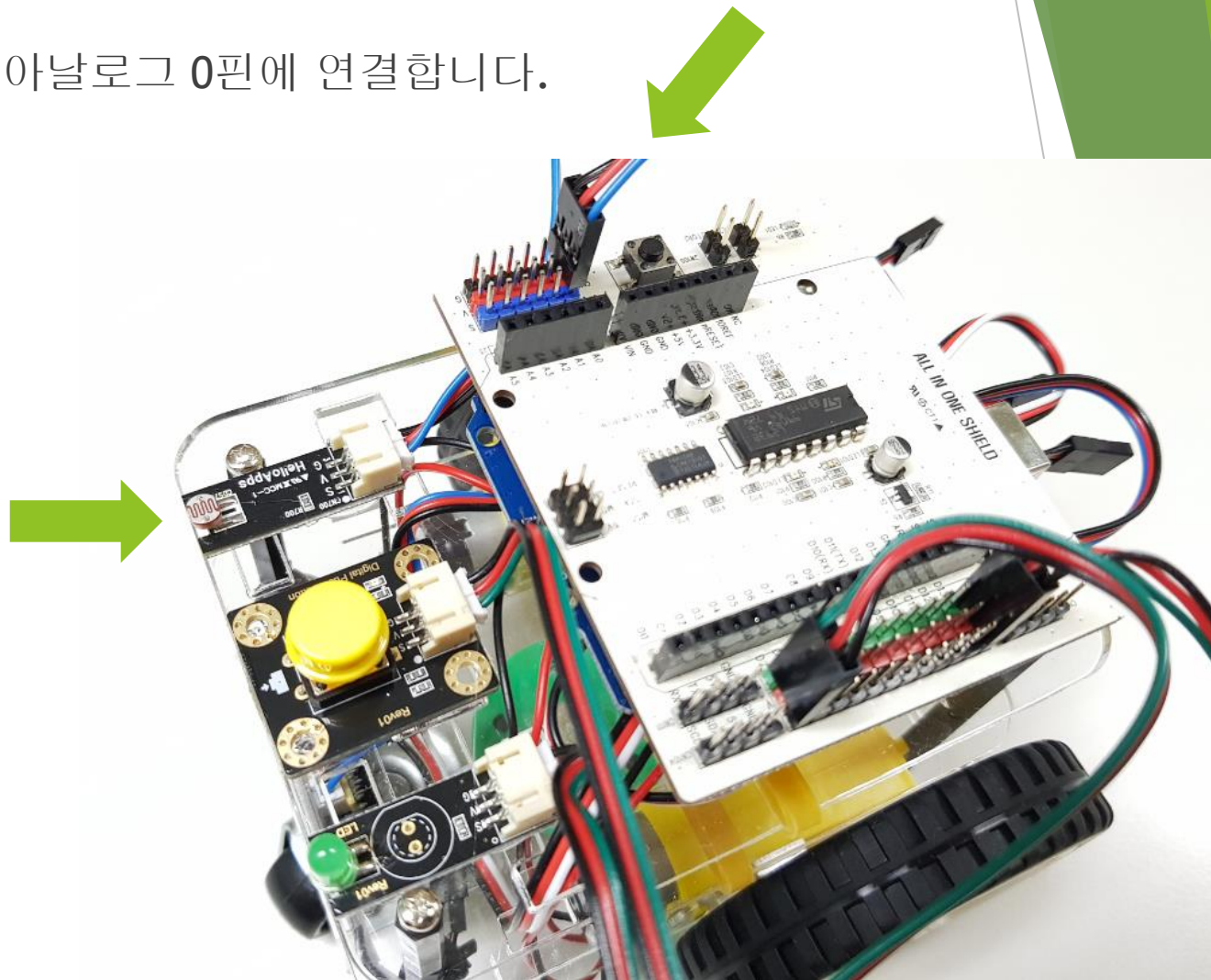


조도 센서

# 조도센서 연결 실습

센서 연결하기

- ▶ 조도센서를 아날로그 0핀에 연결합니다.



# 조도센서 연결 실습

- ▶ 조도센서를 아날로그 0핀에 연결합니다.



0번



# 아날로그 명령어

- ▶ 아날로그 핀에서 값을 읽어 올 때 사용하는 명령어

## AnalogRead

- ▶ 아날로그 핀에서 값을 읽어 올 때에는 핀번호와 값을 저장할 변수가 필요함

**a = AnalogRead (핀번호)**

# 조도 센서값 출력하기

# 조도 센서값 출력하기

- ▶ 0.1초 간격으로 조도 센서의 값을 읽어와서 콘솔창에 출력해 봅니다.

```
void setup()
{

}

void loop()
{
    a0 = AnalogRead(0)

}
```

나머지 기능을 완성해 봅니다.



# 조도 센서값 출력하기

- ▶ 0.1초 간격으로 조도 센서의 값을 읽어와서 콘솔창에 출력해 봅니다.

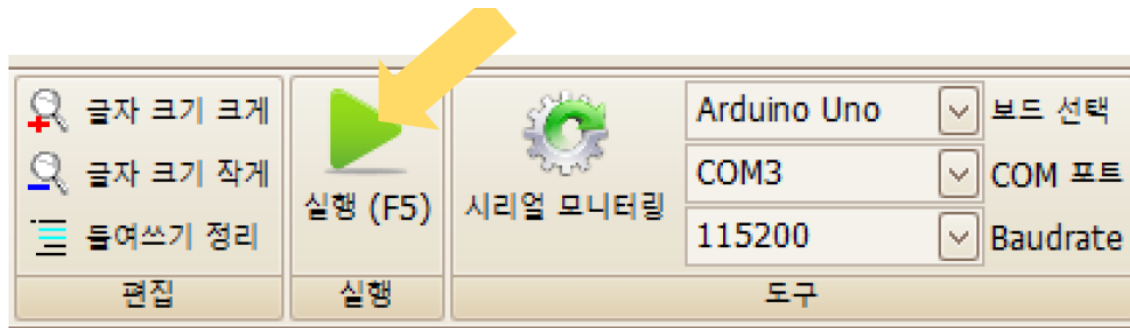
```
void setup()
{

}

void loop()
{
    a0 = AnalogRead(0)
    PrintLine(a0)
    Delay(100)
}
```

# 조도 센서값 출력하기

- ▶ 상단 메뉴의 가운데 있는 실행 버튼을 클릭하여 프로그램을 아두이노에 업로드 시킵니다.



실행 버튼을 클릭하여 프로그램을 아두이노 보드에 업로드 시킨다

프로그램을 실행한 후, 콘솔창에 표시되는 결과를 확인해 봅니다.

# 조도 센서값 출력하기

- ▶ 아래 조건의 값을 기록합니다.

가장 밝았을 때의 값	
가장 어두웠을 때의 값	
평균 값	

# 어두워지면 저절로 켜지는 장치 만들기

# 어두워지면 저절로 켜지는 장치

## ▶ 부품 연결

- 디지털 13번 핀에 **LED**를 연결합니다.
- 아날로그 0번 핀에 조도센서를 연결합니다.

# 어두워지면 저절로 켜지는 장치

- ▶ 아래의 코드에 조건 비교문을 추가해 봅시다.

```
void setup()
{
}

void loop()
{
    a0 = AnalogRead(0)
    PrintLine(a0)

    Delay(100)
}
```

# 어두워지면 저절로 켜지는 장치

- ▶ 아래의 코드에 if 비교 로직을 추가해 봅시다.

```
void loop()
{
    a0 = AnalogRead(0)
    PrintLine(a0)

    if ()
    {
    }
    else
    {
    }

    Delay(100)
}
```

# 어두워지면 저절로 켜지는 장치

- ▶ 아래의 코드에 **LED** 명령어를 추가해 봅시다.

```
void loop()
{
    a0 = AnalogRead(0)
    PrintLine(a0)

    if (a0 < 600)
    {
    }
    else
    {
    }

    Delay(100)
}
```



# 어두워지면 저절로 켜지는 장치

## ▶ 완성된 코드

```
void loop()
{
    a0 = AnalogRead(0)
    PrintLine(a0)

    if (a0 < 600)
    {
        DigitalWite(13, HIGH)
    }
    else
    {
        DigitalWite(13, LOW)
    }

    Delay(100)
}
```

# AnalogWrite

# AnalogWrite

- **AnalogWrite** 명령어는 아날로그 이름이 붙어있지만 디지털 핀에 적용합니다.
- 모든 디지털 핀에 사용할 수 있는 것은 아니고, **PWM** 핀에만 사용할 수 있습니다.
- 디지털 핀으로 가상으로 아날로그 처럼 값을 출력시킵니다.
- **0 ~ 255** 사이의 값을 입력하면 **0V ~ 5V** 사이의 가상 전압이 만들어 집니다.
- 디지털 핀 중에서 **3, 5, 6, 9, 10, 11**핀에만 사용 가능합니다.

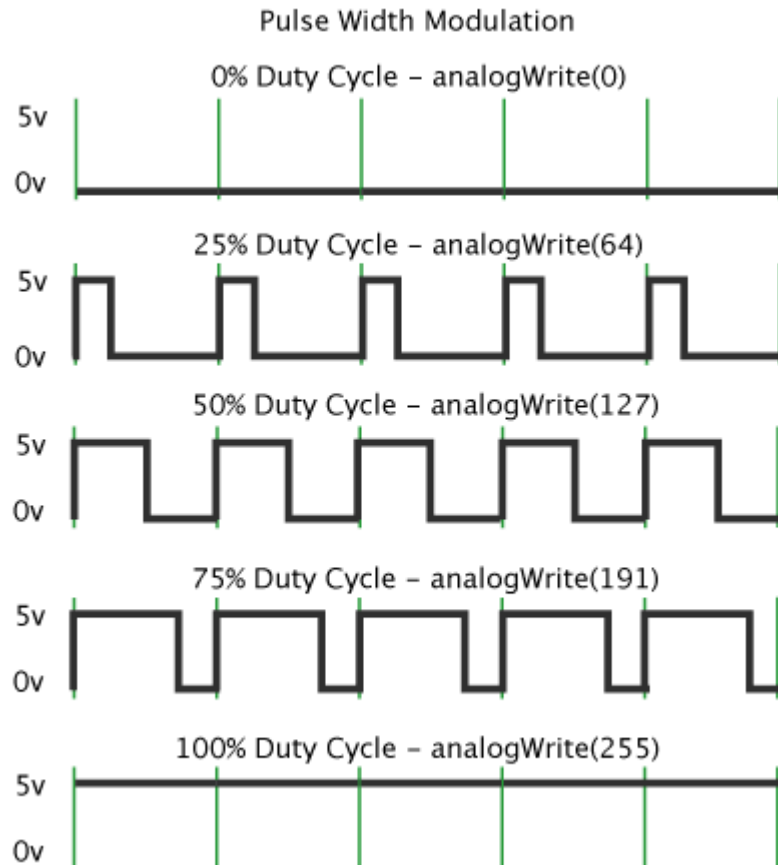
# AnalogWrite

- ▶ AnalogWrite 명령어를 사용할 수 있는 핀에는 ~ 표시가 있음



# AnalogWrite

- PWM은 필요한 전압을 0과 1의 간격을 조절하여 생성합니다.
- 0 ~ 255 사이의 값을 입력하면 0V ~ 5V 사이의 가상 전압이 만들어 집니다.



# AnalogWrite로 밝기 조절하기

# AnalogWrite로 밝기 조절하기

- LED를 PWM이 지원되는 디지털 11번 핀에 연결합니다.
- [0-9] for 반복문 명령어를 추가합니다.
- for 반복문 안에 그림과 같이 **PrintLine** 명령어와 **Delay**를 추가해 봅니다.
- LED 제어에 앞서 for 반복문을 테스트해 봅시다.
- 아래 예에서는 for 반복문에 있는 변수 **i**는 조건이 만족될 때 까지 1씩 증가합니다.

# AnalogWrite로 밝기 조절하기

```
void setup()
{
}

void loop()
{
  for (i = 1; i <= 10; i=i+1)
  {
    PrintLine(i)
    Delay(100)
  }
}
```



# AnalogWrite로 밝기 조절하기

- LED를 PWM이 지원되는 디지털 11번 핀에 연결합니다.
- 반복문을 이용하여 LED의 밝기를 256단계로 증가시켜 봅니다.
- for 반복문의 블록 안에 [1-1-7] AnalogWrite 명령어를 추가합니다.
- AnalogWrite의 핀번호를 11로 수정하고 입력창에 i를 입력해 줍니다.
- for 반복문과 AnalogWrite 입력창의 값을 그림과 같이 변경해 줍니다.

# AnalogWrite로 밝기 조절하기

```
void setup()
{
}

void loop()
{
    for (i = 1; i <= 10; i=i+1)
    {
        AnalogWrite(11, i)
        Delay(10)
    }
}
```

# 조도 센서값에 따라 밝기가 조절되는 LED 만들기

# 밝기가 조절되는 LED

- LED를 PWM이 지원되는 디지털 11번 핀에 연결합니다.
- 밝으면 LED를 어둡게 합니다.
- 어두우면 LED를 밝게 합니다.
- LED의 밝기는 조도 센서값과 반대가 되는 것에 유의합니다.

# 밝기가 조절되는 LED

```
void setup()
{
}

void loop()
{
    a0 = AnalogRead(0)

    a0 = 1023 - a0

    AnalogWrite(11, a0)

    Delay(100)
}
```

# 값을 변환시키는 매핑 함수 (map)

# Map 함수

- Map 함수는 값을 다른 범위로 매핑 변환하는 기능을 수행합니다.
- 아래 예에서 **a1**은 0 ~ 10 범위에서 0~100의 범위로 변환됩니다.
- 따라서 **a1**에는 90이 저장됩니다.
- **a2**는 0 ~ 10 범위에서 10 ~ 0의 범위로 변환됩니다.
- 따라서 **a2**에는 1이 저장됩니다.

# Map 함수

```
void setup()
{
    a = 9
    a1 = map(a, 0, 10, 0, 100)
    PrintLine(a1)

    a2 = map(a, 0, 10, 10, 0)
    PrintLine(a2)
}

void loop()
{
}
```



# Map 함수로 LED 밝기 조절하기

- LED를 PWM이 지원되는 디지털 11번 핀에 연결합니다.
- 밝으면 LED를 어둡게 합니다.
- 어두우면 LED를 밝게 합니다.
- LED의 밝기는 조도 센서값과 반대가 되는 것에 유의합니다.

# Map 함수로 LED 밝기 조절하기

```
void setup()
{
}

void loop()
{
    a0 = AnalogRead(0)

    a0 = Map(a0, 0, 1023, 1023, 0)

    AnalogWrite(11, a0)

    Delay(100)
}
```