



3. 아두이노 출력

아두이노 출력

1. LED 깜박임 예제를 통해 디지털 출력 이해하기
2. PWM을 이용한 LED 밝기 제어
3. 여러 개의 LED 상태 제어하기
4. 시프트레지스터를 사용하여 LED 제어하기
5. RGB LED 제어하기
6. 릴레이로 LED 제어하기

LED 깜박임 예제를 통해 디지털 출력 이해하기

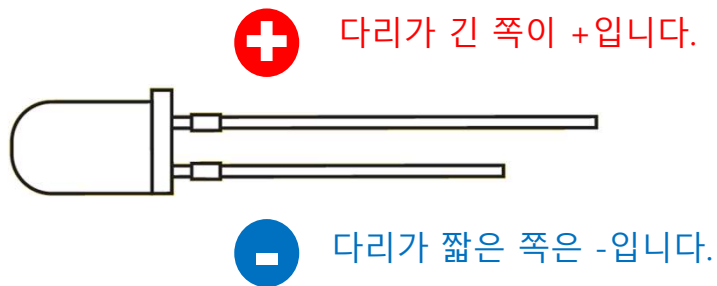
❖ LED란?



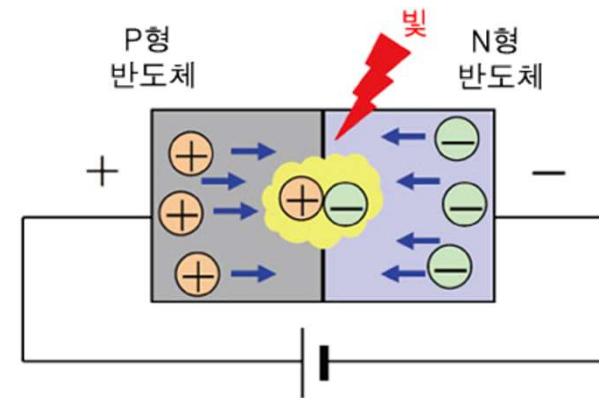
한쪽 방향으로만 전류가 흐르도록 제어하는 반도체 소자를 다이오드라고 하며, 다이오드 중 전기 에너지를 빛 에너지로 변환하는 것을 LED(Light Emitting Diode)라고 한다.
LED는 다양한 색깔의 빛을 낼 수 있으며, 에너지 효율이 높고 직진성이 좋아 멀리서도 잘 보인다는 장점이 있다.

LED 깜박임 예제를 통해 디지털 출력 이해하기

❖ LED 구조 및 원리



LED는 2개의 전극 단자로 구성되어 있으며, 긴 단자는 애노드(Anode), 짧은 단자는 캐소드(Cathode)라 부른다. 애노드에 (+)전극을 캐소드에 (-)전극을 연결하면 LED가 켜지게 된다.



LED는 발광 다이오드라고 하며, Ga(갈륨), P(인), As(비소)를 재료로 하여 반도체. 다이오드의 특성을 가지고 있으며, 전류를 흐르게 하면 각종 색으로 빛을 발한다. LED에 전류를 흐르면 각 반도체에 있는 전자와 정공이 결합하게 되는데 이 때 생기는 에너지가 빛이 된다.

LED 깜박임 예제를 통해 디지털 출력 이해하기

❖ 디지털과 아날로그 신호

디지털신호 

OFF
(0, LOW)



ON
(1, HIGH)

0_(LOW)

또는

1_(HIGH)

`digitalRead()`

디지털 입력장치로부터 디지털 신호를 읽어온다

`digitalWrite()`

디지털 출력장치에 디지털 신호를 보낸다

아날로그 신호



0
(0, LOW)



100
(1023, HIGH)

10bit(2^{10})

0_(LOW)에서

1023_(HIGH)까지

`analogRead()`

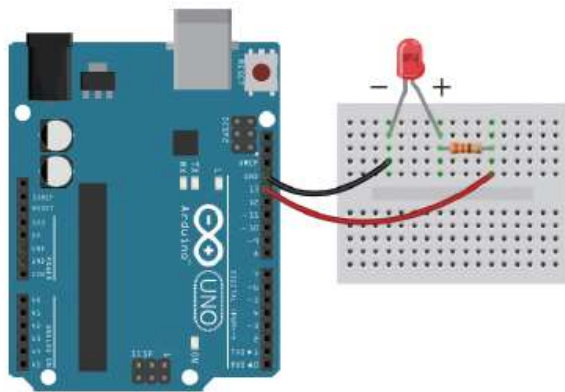
아날로그 입력장치로부터 아날로그 신호를 읽어온다(0~1023)

`analogWrite()`

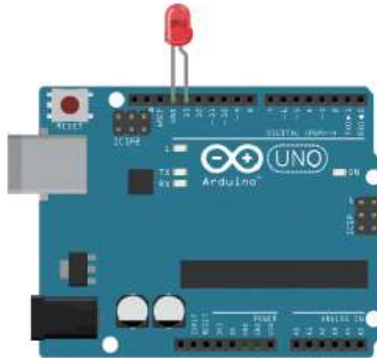
아날로그 출력장치에 아날로그 신호를 보낸다.(0~255)

LED 깜박임 예제를 통해 디지털 출력 이해하기

❖ 아두이노 보드 13번 핀에 LED와 저항 연결



(a) 브레드보드를 이용한 LED 연결



(b) 브레드보드 없이 LED 연결

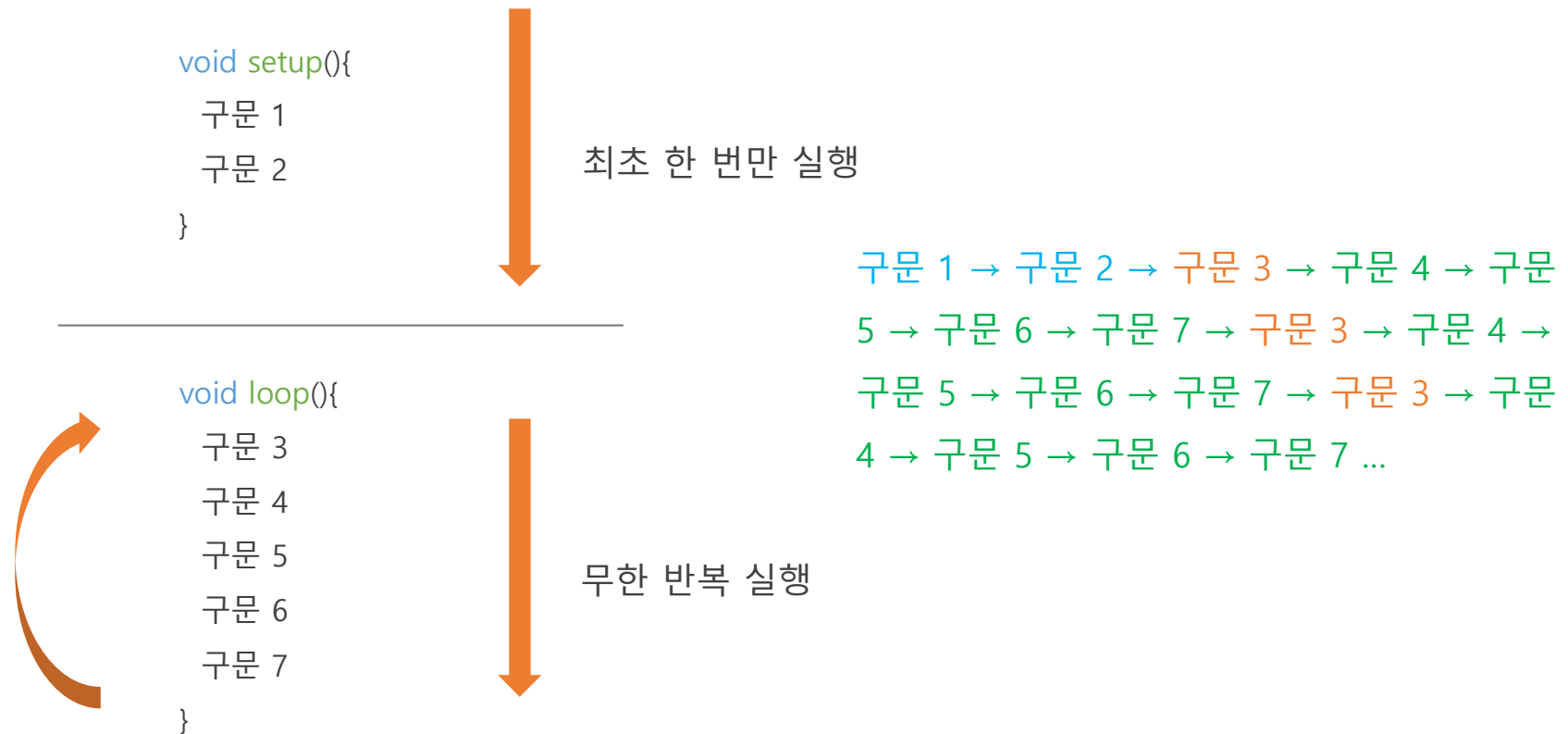
```
// setup 함수는 보드에 전원 공급 또는 리셋 버튼을 눌렀을 때 한 번 실행된다.
void setup()
{
    pinMode (13, OUTPUT);    // 13번 디지털 핀을 출력으로 설정
}

// loop 함수는 전원이 켜져 있는 동안 무한 반복해서 실행된다.
void loop()
{
    digitalWrite (13, HIGH);  // 13번 핀에 HIGH 설정. LED 켜짐
    delay (1000);             // 1000 ms(1초)동안 대기(시간 지연)
    digitalWrite (13, LOW);   // 13번 핀에 LOW 설정. LED 꺼짐
    delay (1000);             // 1000 ms(1초)동안 대기(시간 지연)
}
```

330 Ω 의 저항과 LED의 긴 쪽을 디지털 핀 13번 핀에 연결하고 짧은 쪽을 그라운드 즉 GND(Ground) 핀에 연결한다. 아두이노 보드에 내장된 LED에 이미 저항이 연결되어 있어서 이것과 같은 선으로 연결된 13번 핀의 경우 저항 없이 회로를 구성해도 문제 되지 않는다.

LED 깜박임 예제를 통해 디지털 출력 이해하기

❖ 순차 구조란?



LED 깜박임 예제를 통해 디지털 출력 이해하기

❖ 함수 설명

`pinMode(사용할 핀의 번호 , 입력(INPUT) 또는 출력(OUTPUT));`

핀의 상태를 입력 또는 출력으로 설정

* 디지털 입/출력시 사용

LED 깜박임 예제를 통해 디지털 출력 이해하기

❖ 함수 설명

`delay(대기할 시간(밀리초 / ms 단위));`

입력한 시간만큼 프로그램 일시 중지(대기, 유지)

$* 1\text{ms} = 1/1000\text{s}$

LED 깜박임 예제를 통해 디지털 출력 이해하기

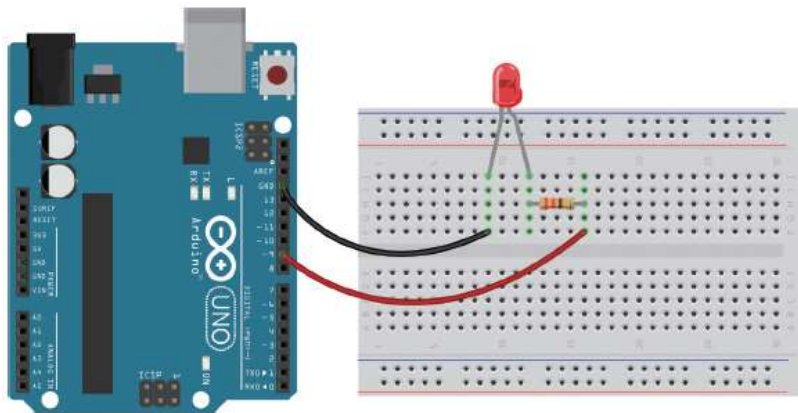
❖ 함수 설명

`digitalWrite(사용할 핀의 번호 , ON(HIGH, 1) 또는 OFF(LOW, 0));`

pinMode에서 출력으로 설정한 핀의 상태를 제어(ON, OFF)

PWM을 이용한 LED 밝기 제어

❖ 아두이노 보드 9번 핀에 LED와 저항 연결

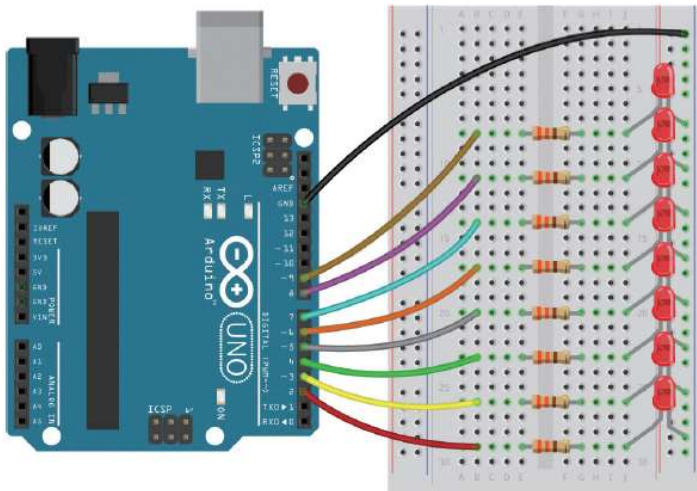


그림과 같이 ~9번 핀에 LED를 연결하고 파일 → 예제
→ 03.Analog → Fading 스케치 예제를 업로드 한다.

```
int ledPin = 9;           // 9번 핀에 LED 연결
void setup()
{
    // 아무것도 하지 않는다.
}
void loop()
{
    // fadeValue 변수를 선언하고 최솟값에서 최댓값까지 5씩 늘린다.
    for (int fadeValue = 0; fadeValue <= 255; fadeValue += 5)
    {
        analogWrite (ledPin, fadeValue);    // fadeValue 값 설정
        delay (30); // 30 ms 대기
    }
    // fadeValue 변수를 선언하고 최솟값에서 최댓값까지 5씩 줄인다.
    for (int fadeValue = 255; fadeValue >= 0; fadeValue -= 5)
    {
        analogWrite (ledPin, fadeValue);    // fadeValue 값 설정
        delay (30); // 30 ms 대기
    }
}
```

여러 개의 LED 상태 제어하기

❖ 아두이노 보드 2~9번 핀에 LED와 저항 연결



브레드보드의 가로선 한쪽을 아두이노 보드의 GND와 연결하고 8개 LED의 짧은 다리 쪽을 브레드보드의 그라운드 부분에 연결하면 배선이 훨씬 간단해진다.

```
int Led_pin[8] = { 2, 3, 4, 5, 6, 7, 8, 9 };    // 핀 2번부터 9번까지 연결

void setup()
{
    for (int i = 0; i < 8; i++)                  // 배열 인덱스 0에서 7까지 증가
    {
        pinMode (Led_pin[ i ], OUTPUT);        // 각 핀을 출력으로 설정
    }
}

void loop()
{
    for (int i = 0; i < 8; i++)                  // 배열 인덱스 0에서 7까지 증가
    {
        digitalWrite (Led_pin[ i ], HIGH);     // 2번 핀부터 차례대로 LED On
        delay (500); // 500 ms 지연
    }

    for (int i = 7; i >= 0; i--)                 // 배열 인덱스 7에서 0까지 감소
    {
        digitalWrite (Led_pin[ i ], LOW);      // 9번 핀부터 차례대로 LED Off
        delay (500); // 500 ms 지연
    }
}
```

PWM을 이용한 LED 밝기 제어

❖ 펄스 폭 변조(PWM)

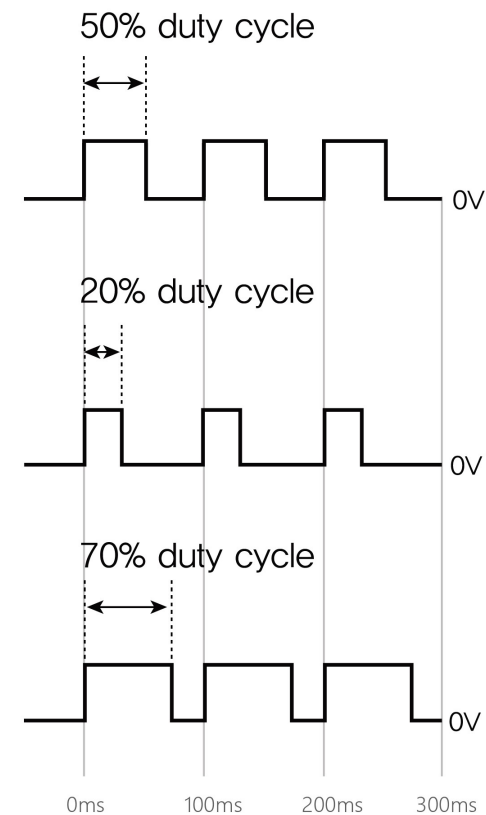
디지털 신호를 사용하여 아날로그와 유사한 신호를 만드는 방법으로, PWM을 사용하면 디지털 신호를 아날로그 신호와 유사하게 표현해 줄 수 있다.

디지털 신호를 일정 주기로 ON과 OFF를 반복하여 구형파를 만들고, ON으로 설정된 시간의 비율을 조정함으로써 출력 전압에 변화를 주는 것이다.

8bit(2^8)에서는 범위가 아래와 같다

0_(LOW) 에서

255_(HIGH) 까지



PWM을 이용한 LED 밝기 제어

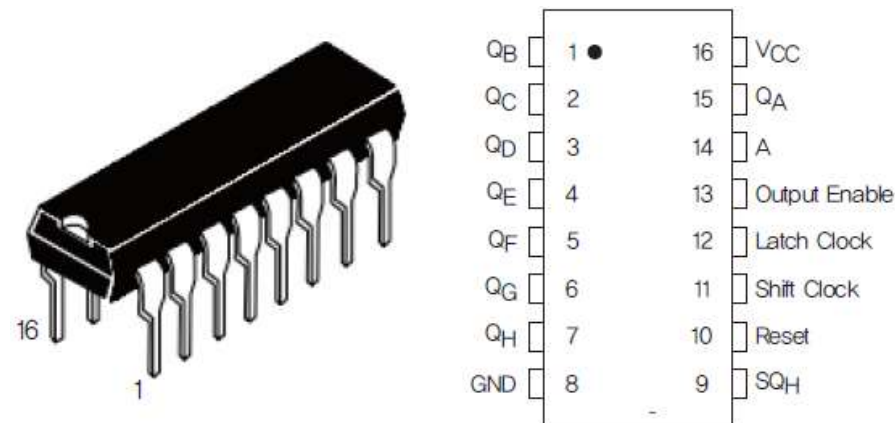
❖ 함수 설명

`analogWrite(사용할 PWM 핀의 번호 , 제어할 값 (0 ~ 255));`

함수에 입력한 핀의 상태를 제어(0~255)
PWM을 통해 제어할 경우 pinMode 설정 불필요

시프트레지스터를 사용하여 LED 제어하기

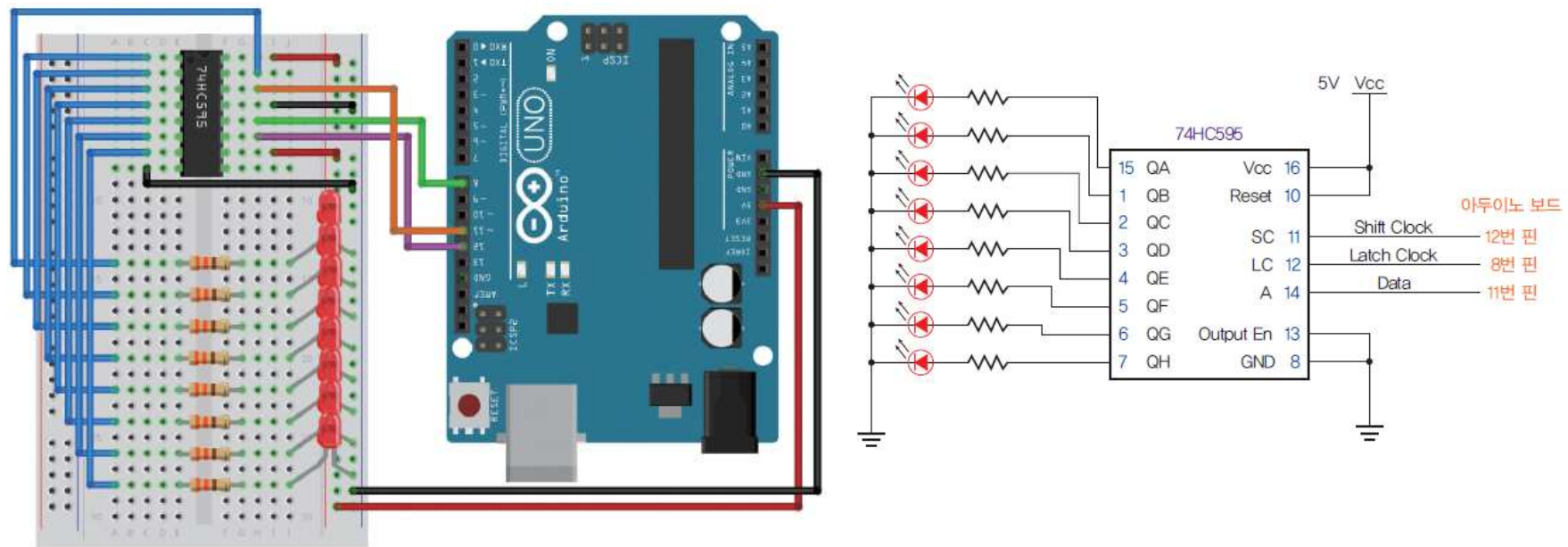
❖ IN74HC595AN IC와 핀 배열 및 설명



심볼	핀	설명
QB	1	병렬 데이터 출력 B
QC	2	병렬 데이터 출력 C
QD	3	병렬 데이터 출력 D
QE	4	병렬 데이터 출력 E
QF	5	병렬 데이터 출력 F
QG	6	병렬 데이터 출력 G
QH	7	병렬 데이터 출력 H
GND	8	그라운드(0 V)
SQH	9	직렬 데이터 출력
Reset	10	리셋(액티브 LOW)
Shift Clock	11	시프트레지스터 클럭 입력
Latch Clock	12	저장 래치 클럭 입력
Output Enable	13	출력 사용(액티브 LOW)
A	14	직렬 데이터 입력
QA	15	병렬 데이터 출력 A
VCC	16	전원 공급

시프트레지스터를 사용하여 LED 제어하기

❖ 시프트레지스터와 LED를 연결한 회로도와 배선 모습



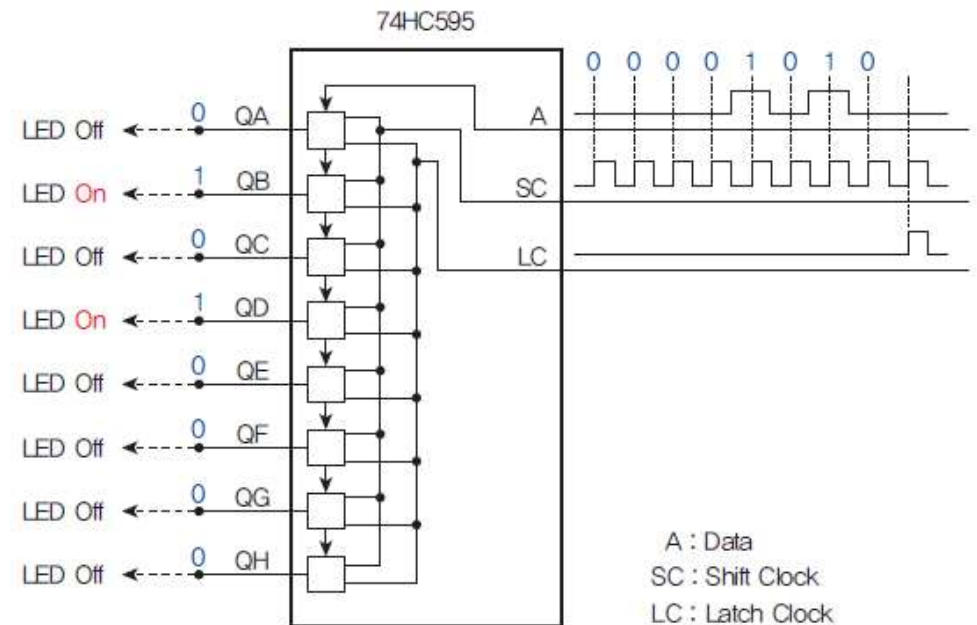
시프트레지스터를 사용하여 LED 제어하기

❖ 숫자 데이터 10의 출력 과정

```
int latchPin = 8;           // 74HC595의 12번 핀 연결
int clockPin = 12;          // 74HC595의 11번 핀 연결
int dataPin = 11;           // 74HC595의 14번 핀 연결

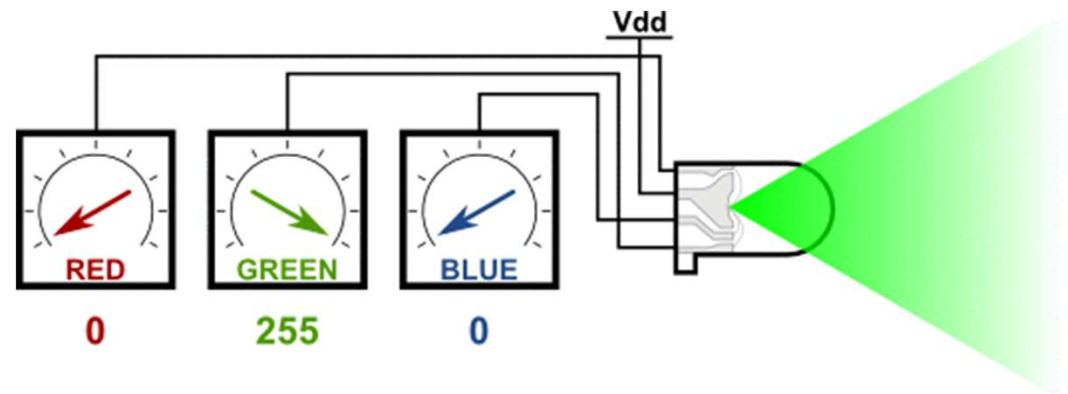
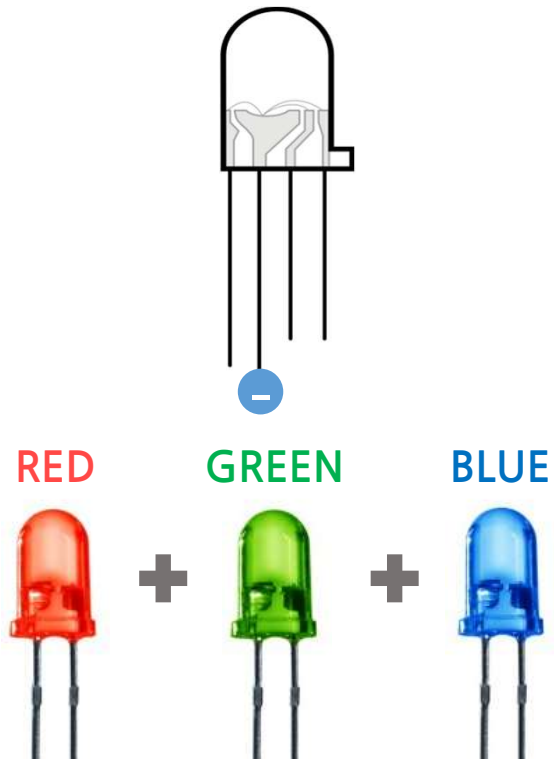
void setup()
{
  pinMode (latchPin, OUTPUT); // 아두이노 보드 8번 핀 출력 설정
  pinMode (clockPin, OUTPUT); // 아두이노 보드 12번 핀 출력 설정
  pinMode (dataPin, OUTPUT);  // 아두이노 보드 11번 핀 출력 설정
}

void loop()
{
  for (int number = 0; number < 256; number++)
  {
    digitalWrite (latchPin, LOW);
    shiftOut (dataPin, clockPin, MSBFIRST, number);
    digitalWrite (latchPin, HIGH);
    delay (100);
  }
}
```



RGB LED 제어하기

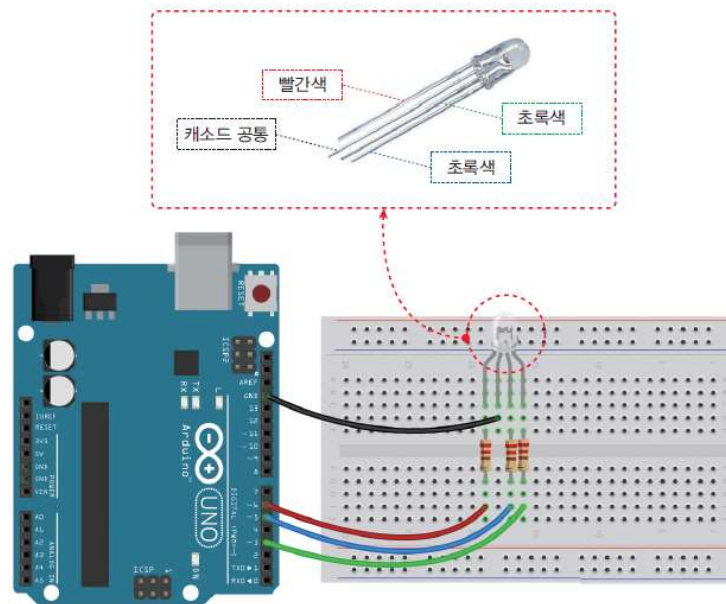
❖ RGB LED 동작 원리



< RGB LED 동작 원리(Cathode Common Type) >

RGB LED 제어하기

❖ 캐소드 공통 형 RGB LED와 아두이노 보드 연결



```
int redPin = 6;           // 빨간색 핀. 아두이노 보드 6번 핀에 연결
int greenPin = 5;         // 초록색 핀. 아두이노 보드 5번 핀에 연결
int bluePin = 3;          // 파란색 핀. 아두이노 보드 3번 핀에 연결
int redOff, greenOff, blueOff; // 각 LED를 끄기 위한 전역 변수 선언
void setup()              // analogwrite 함수 사용 시(PWM)
{                          // pinMode 설정 필요 없음
}

void loop()
{
  analogWrite (redPin, 255); // redPin으로 255 값 출력
  delay (1000);             // 1000 ms 지연
  analogWrite (redPin, redOff); // 빨간색 LED 끄

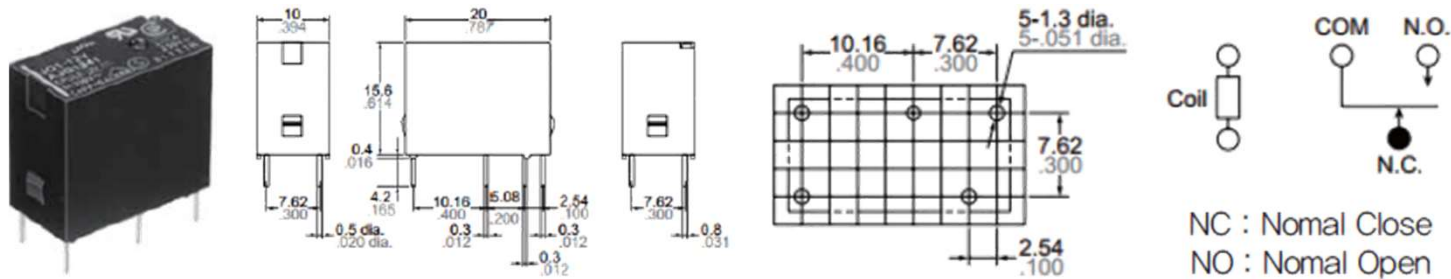
  analogWrite (greenPin, 255); // greenPin으로 255 값 출력
  delay (1000);             // 1000 ms 지연
  analogWrite (greenPin, greenOff); // 초록색 LED 끄

  analogWrite (bluePin, 255); // bluePin으로 255 값 출력
  delay (1000);             // 1000 ms 지연
  analogWrite (bluePin, blueOff); // 파란색 LED 끄
}
```

빨간색, 초록색, 파란색 세 종류의 색을 이용하며, 0~255 범위의 analogWrite 함수의 인자 값을 조절하여 다양한 색을 표현할 수 있다.

릴레이로 LED 제어하기

❖ JQ1-5V-F 릴레이와 핀 구성

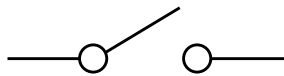
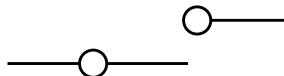
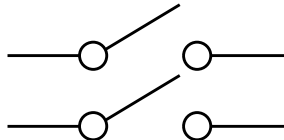
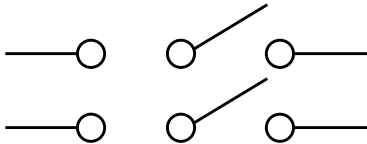


[주요 사양]

- ◆ 코일 유형: 비래칭(non-latching)
- ◆ 코일 전류: 80 mA
- ◆ 코일 전압: 5 VDC
- ◆ 접점 정격(전류): 5 A
- ◆ 스위칭 전압: 250 VAC, 110 VDC-최대
- ◆ 작동 시간: 20 ms
- ◆ 해제 시간: 10 ms
- ◆ 작동 온도: -40~70°C

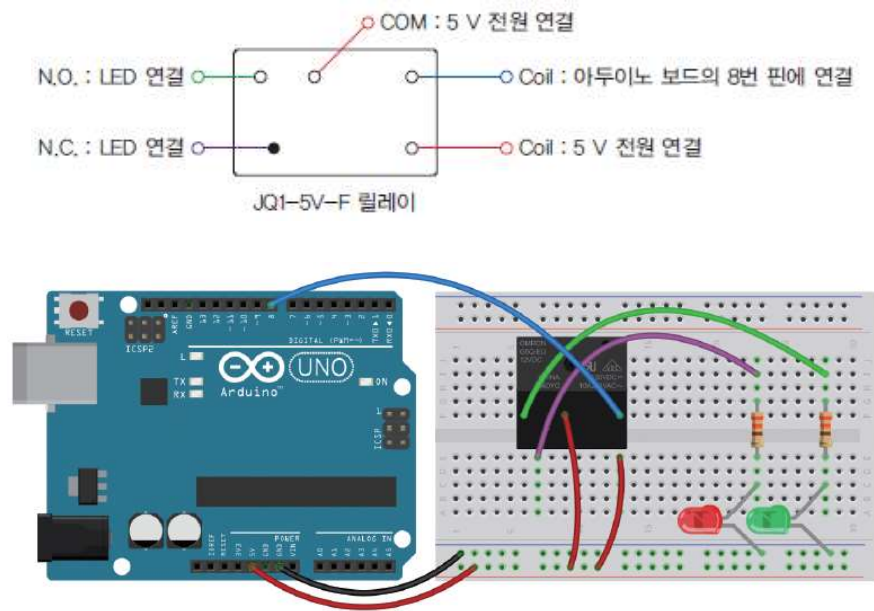
릴레이로 LED 제어하기

❖ 접점 구성에 따른 릴레이 분류

구분	접점 방식	동작 원리
SPST		단극 단투 스위치라 하며 가장 기본적인 릴레이이다. 하나의 선로를 On/Off 한다.
SPDT		단극 쌍투 릴레이라 하며 두 개의 선로 중에 하나를 선택한다.
DPST		쌍극 단투 릴레이라 하며 동시에 두 개의 선로를 On/Off 제어할 수 있다.
DPDT		쌍극 쌍투 릴레이라 하며 SPDT 릴레이 두 개를 한 번에 제어하는 원리이다.

릴레이로 LED 제어하기

❖ 릴레이와 LED 연결



```
int ledPin = 8;           // 코일 한쪽을 아두이노 보드의 8번 핀에 연결

void setup()
{
  pinMode (ledPin, OUTPUT);
}

void loop()
{
  digitalWrite (ledPin, LOW);  // 초록색 LED On
  delay (1000);

  digitalWrite (ledPin, HIGH); // 빨간색 LED On
  delay (1000);
}
```