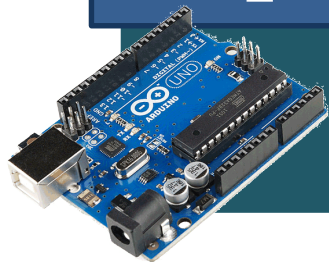


2022년 IoT기반 스마트 솔루션 개발자 양성과정



Firmware [펌웨어]

3-digitalWrite & digitalWrite

담당 교수 : 유근택

010-5486-5376

rqt3340@naver.com

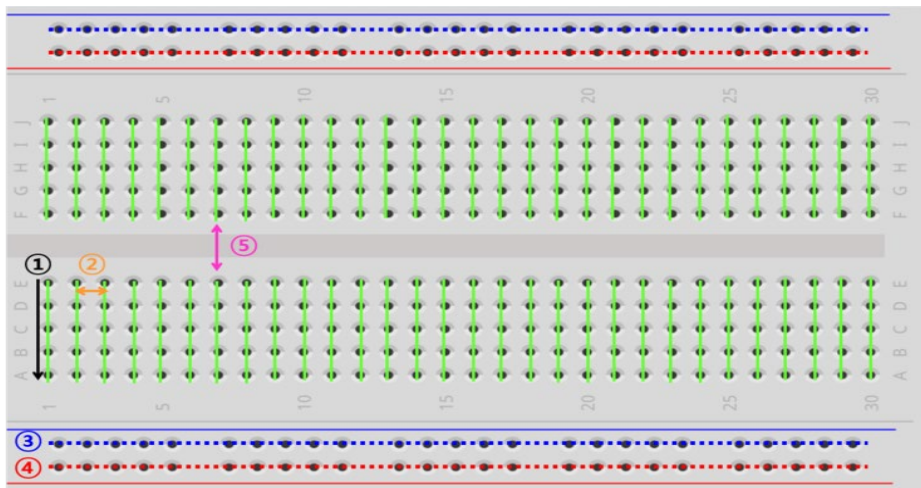
<http://cafe.naver.com/cbdsp>



충북대학교 공동훈련센터

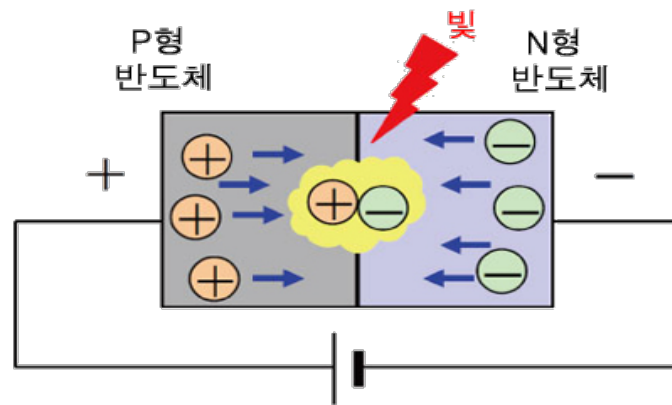
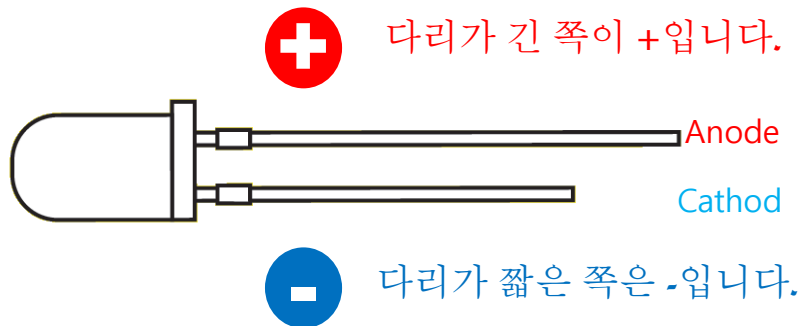
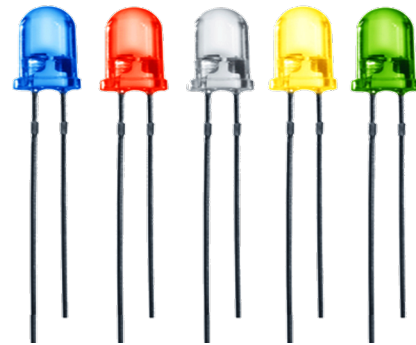
Bread Board

- 브레드 보드는 전자 회로의 프로토타입(시제품)을 만드는 데 사용되고 납땜이 필요 없으며, 재사용할 수 있는 장치



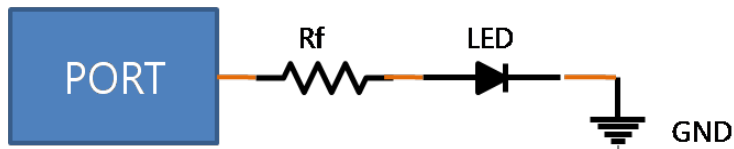
1. LED

- Light Emitting Diode : 발광 다이오드
- PN접합 반도체로 에너지 밴드 겹에 의한 발광현상을 이용
- 수명이 반 영구적, 절전 형 발광, 다양한 색상
- 표시장치나 조명장치로 사용함
- 극성이 있음(Anode, Cathode)

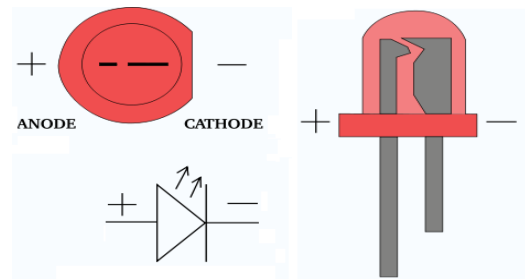
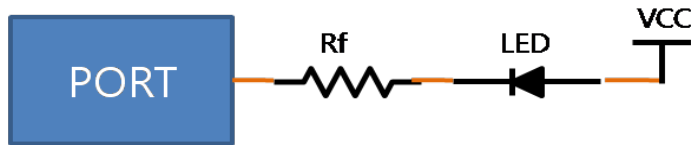


LED의 사용

- Anode에 + 전압이 연결되고 Cathode에 -전압이 연결됨
- 전류 제한 저항 : LED에 정격 전류가 흐르도록 함 (LED 전후에 관계 없음)
- 정방향 연결 : 논리적으로 포트가 1일 때 켜짐



- 역방향 연결 : 논리적으로 포트가 0일 때 켜짐



전류 제한 저항값 구하기

- 옴의 공식 : $R = V / I$
- LED에 흐르는 정격 전류 : $10\text{mA} = 0.01\text{A}$
- LED의 순방향전압 : Forward Voltage = 1.4V
- LED의 공급전압 : 5V



- $R_f = (5\text{V} - 1.4\text{V}) / 0.01\text{A}$
 $= 360 \, \Omega \Rightarrow 390 \, \Omega \, (< 1\text{K} \, \Omega)$

Digital / Analog

❖ 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)

$10\text{bit}(2^{10})$

0 (LOW)에서

1023 (HIGH)까지

`analogRead()`

아날로그 입력장치로부터 아날로그 신호를 읽어온다(0~1023)

`analogWrite()`

아날로그 출력장치에 아날로그 신호를 보낸다.(0~255)



Program Structure

❖ 순차 구조란?

```
void setup(){  
  구문 1  
  구문 2  
}
```

최초 한 번만 실행

```
void loop(){  
  구문 3  
  구문 4  
  구문 5  
  구문 6  
  구문 7  
}
```

무한 반복 실행

구문 1 → 구문 2 → 구문 3 → 구문 4 → 구문
5 → 구문 6 → 구문 7 → 구문 3 → 구문 4 →
구문 5 → 구문 6 → 구문 7 → 구문 3 → 구문
4 → 구문 5 → 구문 6 → 구문 7 ...



Syntax :

- Setup구문

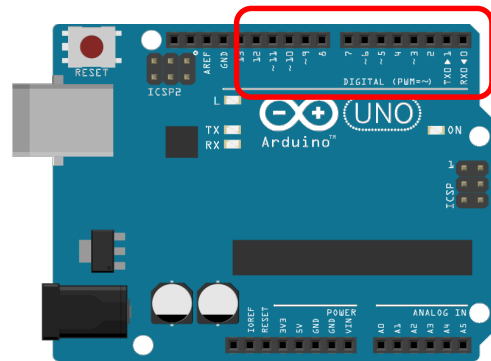
- `pinMode(pin, mode)`
- `pin` : 디지털 출력핀 번호
- `mode` : `INPUT` , `OUTPUT`

- Loop구문

- `digitalWrite(pin, value)`
- `Pin` : 디지털 출력핀 번호
- `value` : `HIGH` 또는 `LOW`

```
void setup( ){  
    pinMode(8,OUTPUT);  
}
```

```
void loop( ){  
    digitalWrite(8,HIGH);  
    delay(100);  
    digitalWrite(8,LOW);  
    delay(100);  
}
```



FUNCTIONS

❖ 함수 설명

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

핀의 상태를 입력 또는 출력으로 설정

* 디지털 입/출력시 사용



FUNCTIONS

❖ 함수 설명

`delay(대기할 시간(밀리초 / ms 단위));`

입력한 시간만큼 프로그램 일시 중지(대기, 유지)

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



FUNCTIONS

❖ 함수 설명

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

pinMode에서 출력으로 설정한 핀의 상태를 제어(ON, OFF)



Ex A3-1 : 정논리

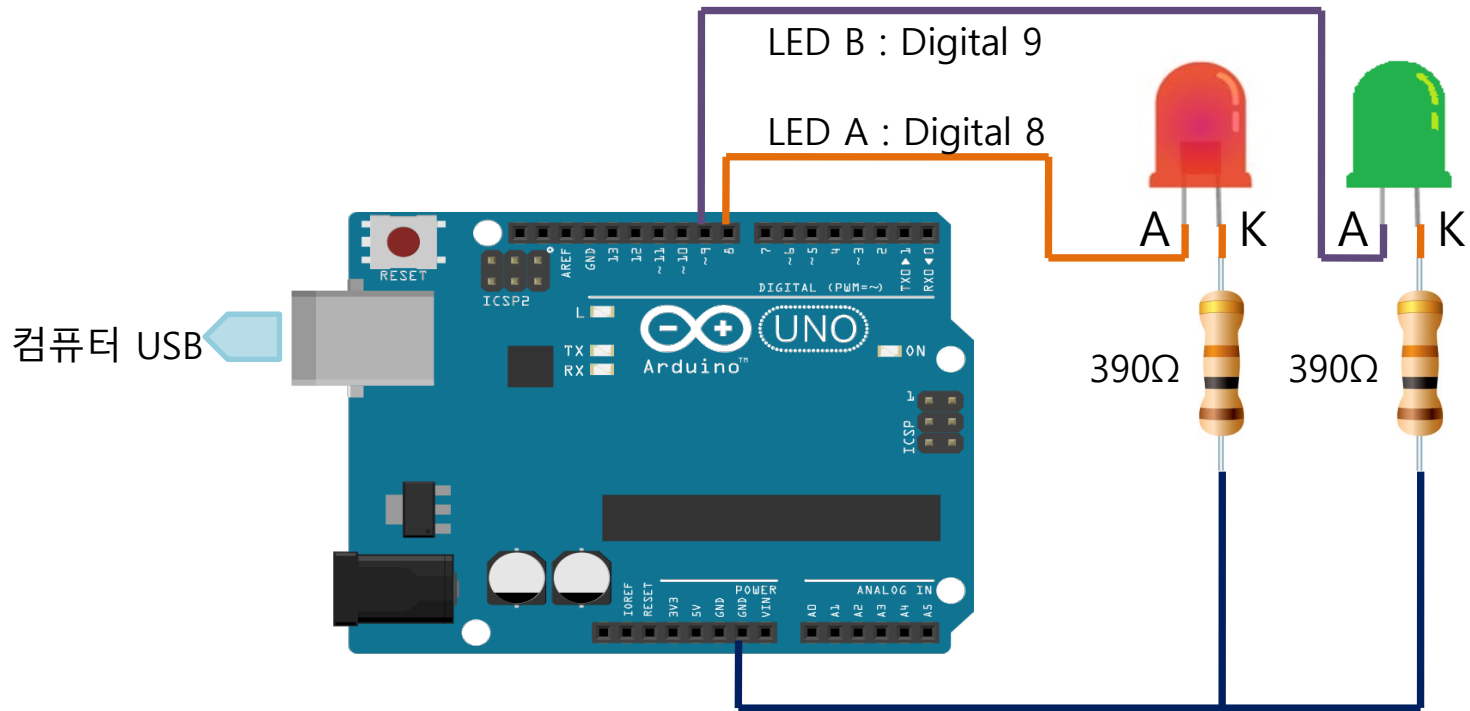
- 경찰차 사인등 만들기
 - 적색, 녹색 LED를 순차적으로 켜다
- 사용 부품

No	Parts	Qty
1	Arduino Uno	1
2	Bread Board	1
3	LED 적색	1
4	LED 녹색	1
5	저항 390Ω	2

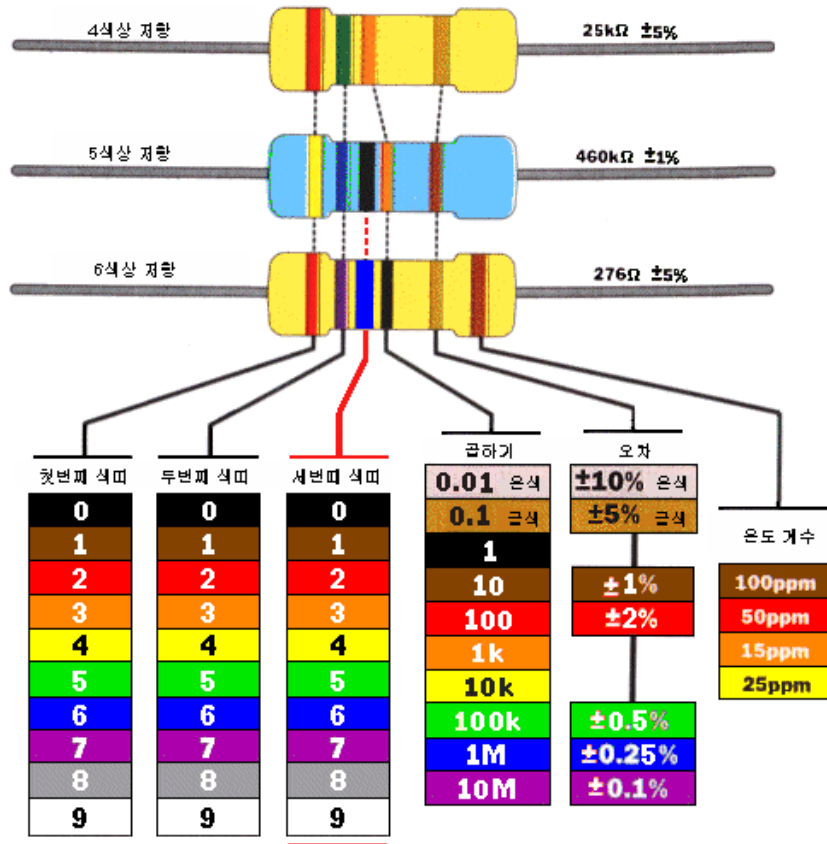
- Delay함수 값을 변경하면서 상태를 확인한다
- 정방향/역방향 LED 연결을 구성한다



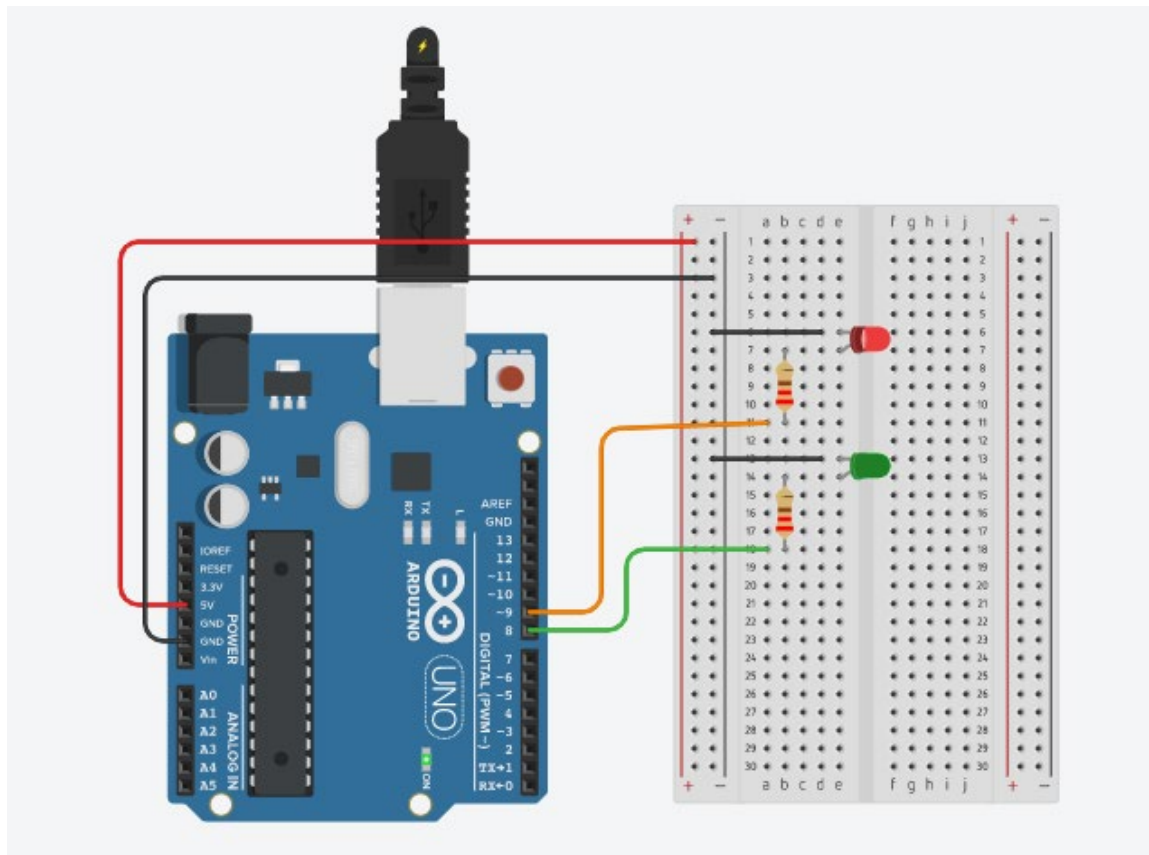
Ex A3-1 : 정논리 결선도



알아두기 : 저항 색상띠



Ex A3-1 : 정논리



Ex A3-1 : 정논리 Sketch

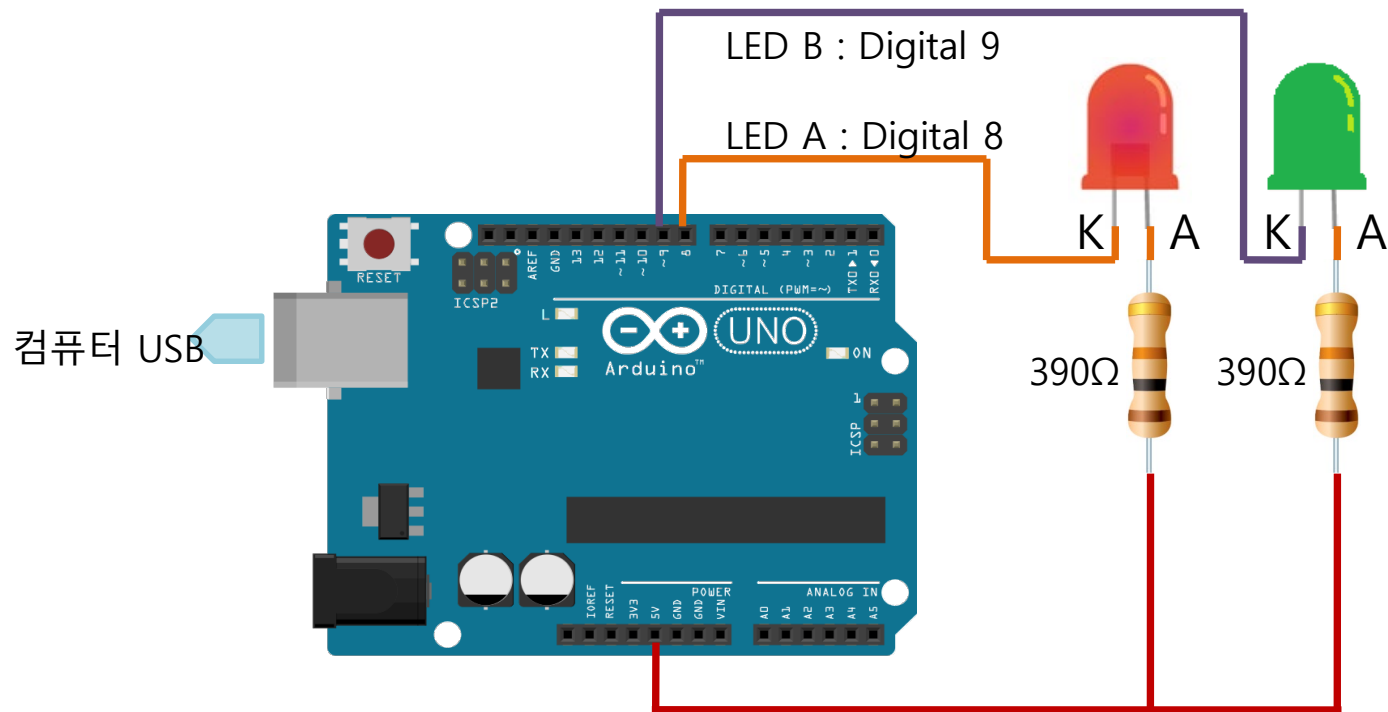
```
#define LED_1 8
#define LED_2 9

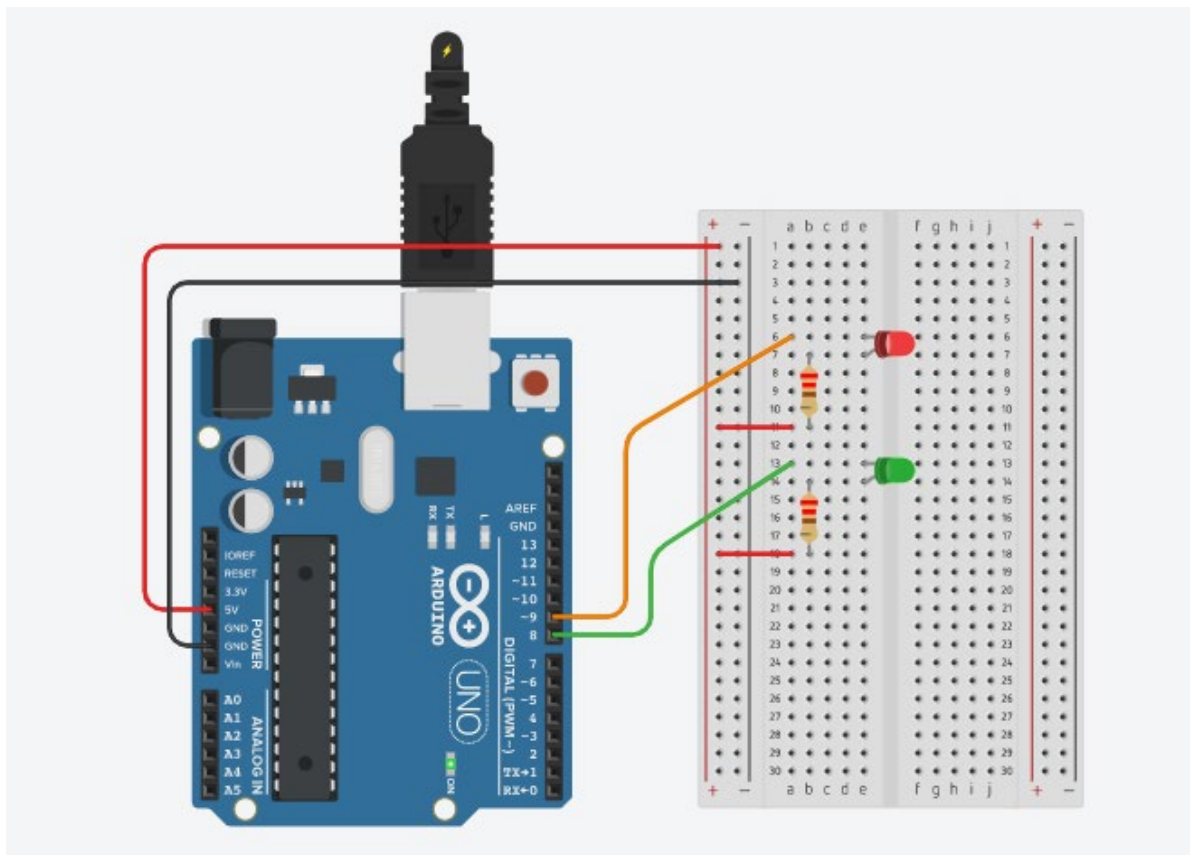
void setup( ) {
  pinMode(LED_1,OUTPUT);
  pinMode(LED_2,OUTPUT);
}

void loop( ) {
  digitalWrite(LED_1,HIGH);
  digitalWrite(LED_2,LOW);
  delay(500);
  digitalWrite(LED_1,LOW);
  digitalWrite(LED_2,HIGH);
  delay(500);
}
```



Ex A3-1 : 부논리 결선도





충북대학교 공동훈련센터

Ex A3-1 : 부논리 Sketch

```
#define LED_1 8
#define LED_2 9

void setup( ) {
  pinMode(LED_1,OUTPUT);
  pinMode(LED_2,OUTPUT);
}

void loop( ) {
  digitalWrite(LED_1,LOW);
  digitalWrite(LED_2,HIGH);
  delay(500);
  digitalWrite(LED_1,HIGH);
  digitalWrite(LED_2,LOW);
  delay(500);
}
```



Ex A3-2

- 경찰차 사인등 만들기 2
 - 적색, 녹색 LED를 2번씩 순차적으로 켜다
- 사용 부품

No	Parts	Qty
1	Arduino Uno	1
2	Bread Board	1
3	LED 적색	1
4	LED 녹색	1
5	저항 390Ω	2

- Delay함수 값을 변경하면서 상태를 확인한다
- 정방향/역방향 LED 연결을 구성한다



Ex A3-3

- 신호등 만들기
 - 적색 LED를 켜 후, 10초 뒤 적색 LED는 끄고 녹색 LED를 켜다.
 - 다시 10초 후 녹색 LED 3회 점멸 후 끄고 동시에 적색 LED를 켜다.
- 사용 부품

No	Parts	Qty
1	Arduino Uno	1
2	Bread Board	1
3	LED 적색	1
4	LED 녹색	1
5	저항 390Ω	2

- Delay함수 값을 변경하면서 상태를 확인한다.



Ex A3-4

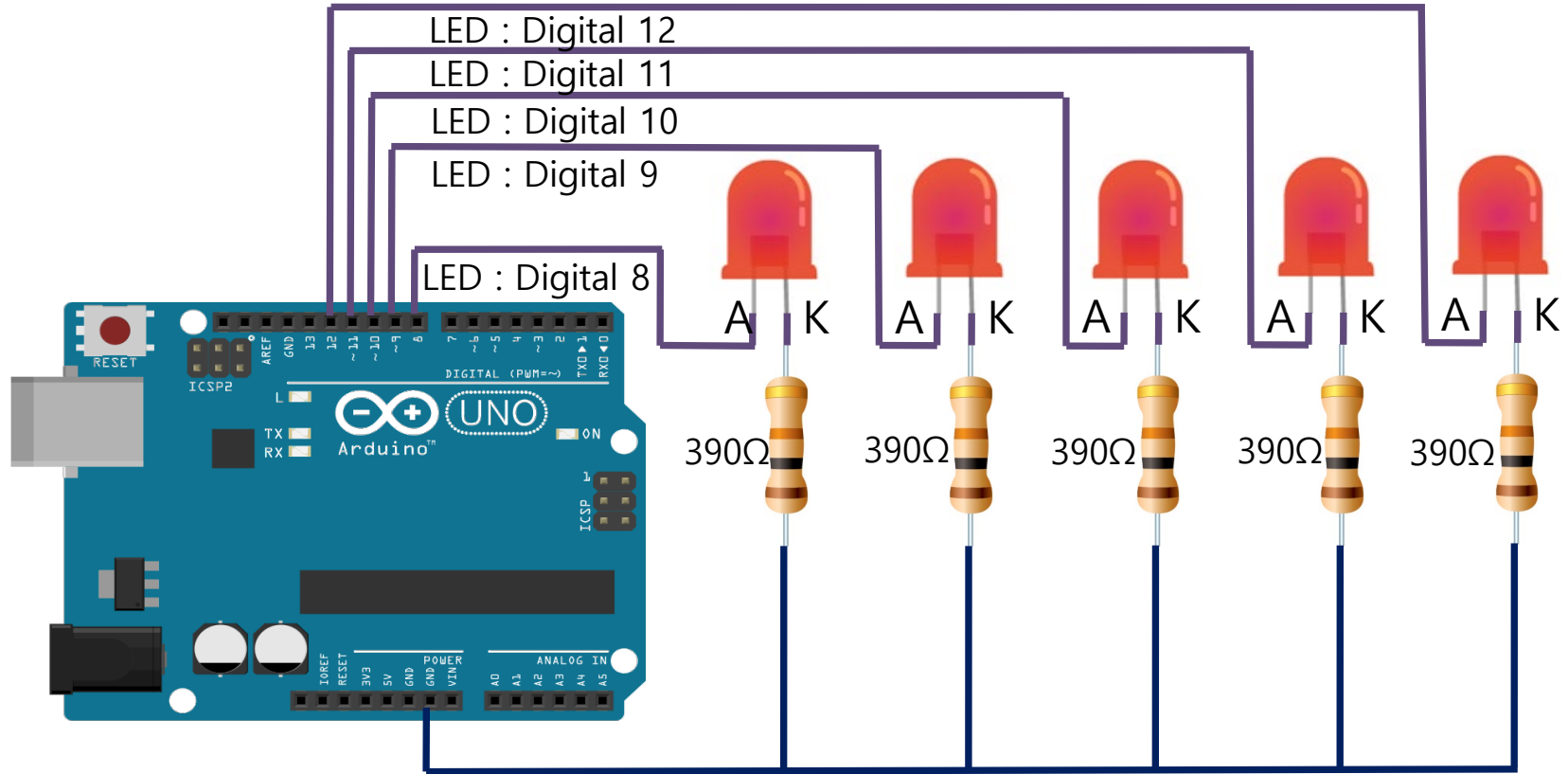
- 전격작전 키트 LED 구현하기
- 적색 LED 5개를 순차적으로 켜며, 좌우로 반복한다.
- 사용 부품

No	Parts	Qty
1	Arduino Uno	1
2	Bread Board	1
3	LED 적색	5
4	저항 390Ω	5

- Delay함수 값을 변경하면서 상태를 확인한다



Ex A3-4 : 정논리 결선도



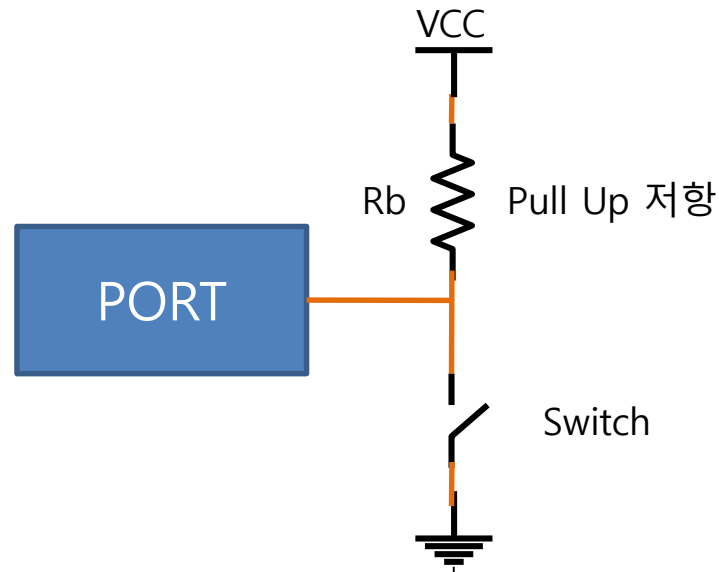
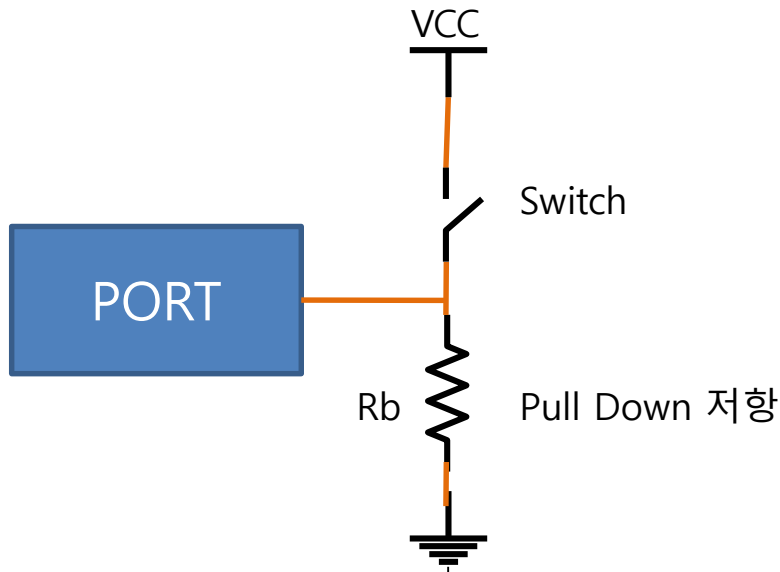
2. Switch

- 전류의 흐름을 개폐함(On/OFF)
- 신호의 입력용으로 사용(Tact Switch)
- Off시 입력 논리값을 명확히 함
 - Pull Up / Pull Down 저항을 함께 연결
- 채터링(Chattering) 방지
- 플레그(Flag) 처리



Switch의 사용

- 정논리 입력 : 스위치를 누르면 1(High Voltage)
- 역논리 입력 : 스위치를 누르면 0(Low Voltage)
- Pull Up/ Pull Down 저항 : 4.7K~10K Ω



Syntax : digitalWrite()

- Setup구문

- pinMode(pin, mode)

- pin : 디지털 입력핀 번호
 - mode : INPUT, OUTPUT

- Loop구문

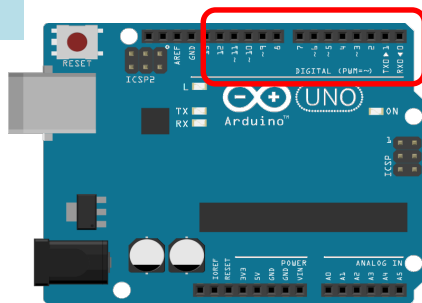
- digitalWrite(pin)

- pin : 디지털 입력핀 번호
 - Return : HIGH 또는 LOW

```
int val = 0;

void setup( ) {
  pinMode(8,INPUT);
}

void loop( ) {
  val=digitalRead(8);
}
```



Ex A3-5

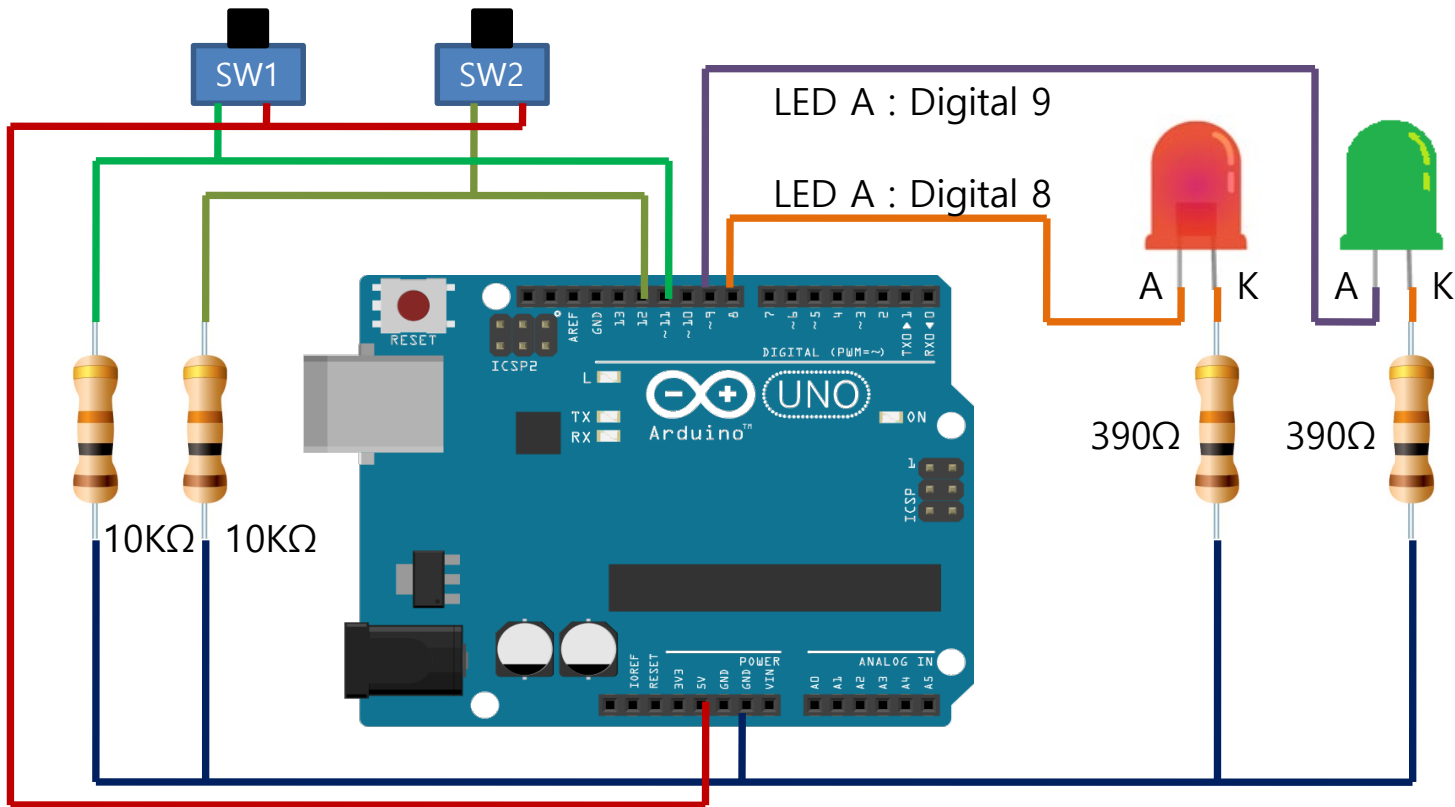
- 스위치 1->적색LED, 스위치 2->녹색LED
- 사용 부품

No	Parts	Qty
1	Arduino Uno	1
2	Bread Board	1
3	LED 적색	1
4	LED 녹색	1
5	저항 390Ω	2
6	Tact Switch	2
7	저항 10KΩ	2

- 정논리/역논리 Switch 연결을 구성한다



Ex A3-5 : 정논리 결선도



Ex A3-5 : 정논리 Sketch

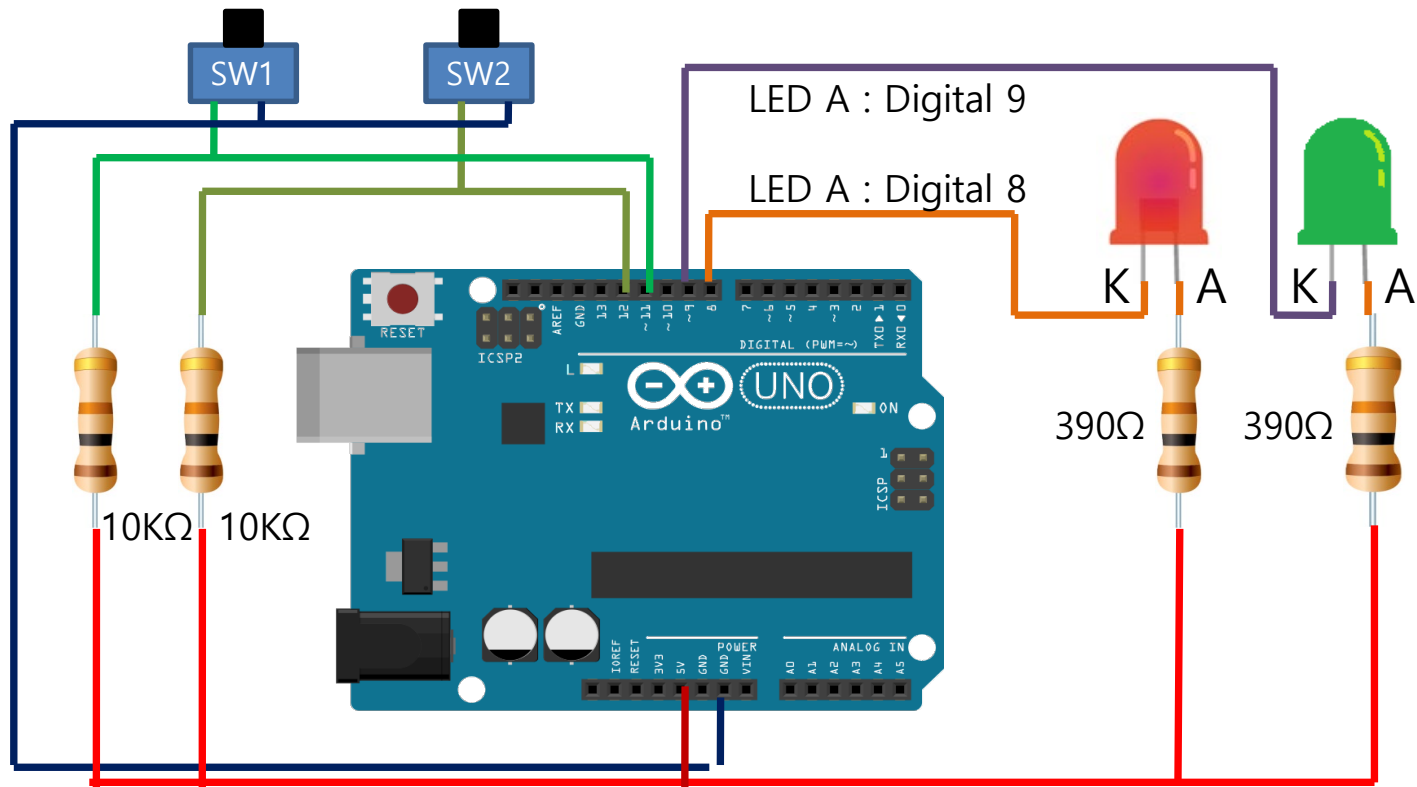
```
#define LED_1_Pin 8
#define LED_2_Pin 9
#define SW1_Pin 11
#define SW2_Pin 12
int SW1=0;
int SW2=0;
```

```
void setup( ) {
  pinMode(LED_1_Pin,OUTPUT);
  pinMode(LED_2_Pin,OUTPUT);
  pinMode(SW1_Pin,INPUT);
  pinMode(SW2_Pin,INPUT);
}

void loop( ) {
  SW1=digitalRead(SW1_Pin);
  digitalWrite(LED_1,SW1);
  SW2=digitalRead(SW2_Pin);
  digitalWrite(LED_2,SW2);
}
```



Ex A3-5 : 부논리 결선도



Ex A3-5 : 부논리 Sketch

```
#define LED_1_Pin 8
#define LED_2_Pin 9
#define SW1_Pin 11
#define SW2_Pin 12
int SW1=0;
int SW2=0;
```

```
void setup( ) {
  pinMode(LED_1_Pin,OUTPUT);
  pinMode(LED_2_Pin,OUTPUT);
  pinMode(SW1_Pin,INPUT);
  pinMode(SW2_Pin,INPUT);
}

void loop( ) {
  SW1=digitalRead(SW1_Pin);
  digitalWrite(LED_1, !SW1);
  SW2=digitalRead(SW2_Pin);
  digitalWrite(LED_2, !SW2);
}
```



Ex A3-6

- 스위치 1->LED 모두 ON, 스위치 2->LED 모두 OFF
- 사용 부품

No	Parts	Qty
1	Arduino Uno	1
2	Bread Board	1
3	LED 적색	1
4	LED 녹색	1
5	저항 390 Ω	2
6	Tact Switch	2
7	저항 10K Ω	2



Syntax : if ()

- 조건에 따라 다른 문장을 실행

```
if (조건식1) {  
    문장 1  
}  
else if (조건식2) {  
    문장 2  
}  
else if (조건식3) {  
    문장 3  
}  
else {  
    문장 4  
}
```



Ex A3-7

- 스위치 1->LED 토글
- 사용 부품

No	Parts	Qty
1	Arduino Uno	1
2	Bread Board	1
3	LED 적색	1
4	LED 녹색	1
5	저항 390 Ω	2
6	Tact Switch	1
7	저항 10K Ω	1



Ex A3-7 : Sketch

```
#define LED_1_Pin 8
#define LED_2_Pin 9
#define SW1_Pin 11
#define SW2_Pin 12
int LED_1=LOW;
int LED_2=LOW;
int SW1=1;
int SW2=1;

void setup( ) {
  pinMode(LED_1_Pin,OUTPUT);
  pinMode(LED_2_Pin,OUTPUT);
  pinMode(SW1_Pin,INPUT);
  pinMode(SW2_Pin,INPUT);
}
```

```
void loop( ) {
  SW1=digitalRead(SW1_Pin);
  if (SW1 == 0){
    if (LED_1==HIGH) LED_1=LOW;
    else LED_1=HIGH;
    digitalWrite(LED_1_Pin,LED_1);
  }

  SW2=digitalRead(SW2_Pin);
  if (SW2 == 0){
    if (LED_2==HIGH) LED_2=LOW;
    else LED_2=HIGH;
    digitalWrite(LED_2_Pin,LED_2);
  }
  delay(500);
}
```



Ex A3-8

- 아두이노를 시작하면 모든 LED가 꺼져 있다.
- 스위치를 누르면 첫번째 LED가 켜지고, 스위치를 계속 누르면 이전 LED는 꺼지고 다음 LED가 켜진다.
- 마지막 세번째 LED가 켜진후 스위치를 한번 더 누르면 전체 LED가 꺼진다.
- 이를 계속 반복한다.

- 사용 부품

No	Parts	Qty
1	Arduino Uno	1
2	Bread Board	1
3	LED 적색	1
4	LED 녹색	1
4	LED 황색	1
5	저항 390Ω	3
6	Tact Switch	1
7	저항 10KΩ	1



Ex A3-8 : Sketch Define

```
#define LED_1_Pin 5
#define LED_2_Pin 6
#define LED_3_Pin 7
#define SW1_Pin 11
#define SW2_Pin 12
int LED_Mode=0;
int SW1=1;
int SW2=1;

void setup( ) {
  pinMode(LED_1_Pin,OUTPUT);
  pinMode(LED_2_Pin,OUTPUT);
  pinMode(LED_3_Pin,OUTPUT);
  pinMode(SW1_Pin,INPUT);
  pinMode(SW2_Pin,INPUT);
}
```



Ex A3-8 : Sketch loop

```
void loop( ) {  
  SW1=digitalRead(SW1_Pin);  
  if (SW1 == 0){  
    LED_Mode = LED_Mode + 1;  
    if (LED_Mode>3) LED_Mode=1;  
  
    if (LED_Mode==1){  
      digitalWrite(LED_1_Pin,HIGH);  digitalWrite(LED_2_Pin,LOW);  digitalWrite(LED_3_Pin,LOW);  
    } else if (LED_Mode==2){  
      digitalWrite(LED_1_Pin,LOW);   digitalWrite(LED_2_Pin,HIGH);  digitalWrite(LED_3_Pin,LOW);  
    } else if (LED_Mode==3){  
      digitalWrite(LED_1_Pin,LOW);   digitalWrite(LED_2_Pin,LOW);  digitalWrite(LED_3_Pin,HIGH);  
    } else {  
      digitalWrite(LED_1_Pin,LOW);   digitalWrite(LED_2_Pin,LOW);  digitalWrite(LED_3_Pin,LOW);  
      LED_Mode=0;  
    }  
  }  
  delay(500);  
}
```



Ex A3-9

- 앞의 실습에서 스위치 한개를 더 추가하여,
- 두번째 스위치를 누를때 LED가 반대로 이동하게 한다.
- 사용 부품

No	Parts	Qty
1	Arduino Uno	1
2	Bread Board	1
3	LED 적색	1
4	LED 녹색	1
4	LED 황색	1
5	저항 390Ω	3
6	Tact Switch	2
7	저항 10KΩ	2



Ex A3-9 : Sketch Main

```
#define LED_1_Pin 5
#define LED_2_Pin 6
#define LED_3_Pin 7
#define SW1_Pin 11
#define SW2_Pin 12
int LED_Mode=0;
int SW1=1;
int SW2=1;

void setup() {
  pinMode(LED_1_Pin,OUTPUT);
  pinMode(LED_2_Pin,OUTPUT);
  pinMode(LED_3_Pin,OUTPUT);
  pinMode(SW1_Pin,INPUT);
  pinMode(SW2_Pin,INPUT);
}
```

```
void loop() {
  SW1=digitalRead(SW1_Pin);
  SW2=digitalRead(SW2_Pin);

  if (SW1 == 0){
    if (++LED_Mode>3) LED_Mode=1;
  }

  if (SW2 == 0){
    if (--LED_Mode<0) LED_Mode=3;
  }

  LED_Disply(LED_Mode );
  delay(500);
}
```



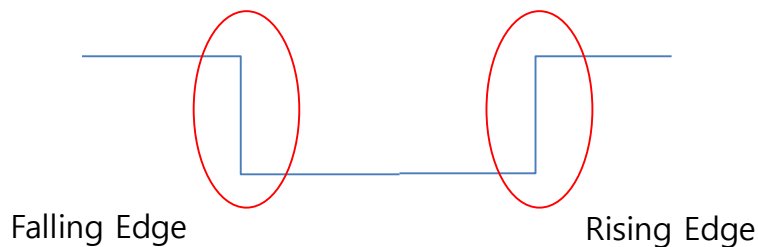
Ex A3-9 : Sketch Sub

```
void LED_Disply(int Mode ) {  
  switch (Mode){  
    case 1:  
      digitalWrite(LED_1_Pin,HIGH); digitalWrite(LED_2_Pin,LOW); digitalWrite(LED_3_Pin,LOW);  
      break;  
    case 2:  
      digitalWrite(LED_1_Pin,LOW); digitalWrite(LED_2_Pin,HIGH); digitalWrite(LED_3_Pin,LOW);  
      break;  
    case 3:  
      digitalWrite(LED_1_Pin,LOW); digitalWrite(LED_2_Pin,LOW); digitalWrite(LED_3_Pin,HIGH);  
      break;  
    default :  
      digitalWrite(LED_1_Pin,LOW); digitalWrite(LED_2_Pin,LOW); digitalWrite(LED_3_Pin,LOW);  
      LED_Mode=0;  
      break;  
  }  
}
```

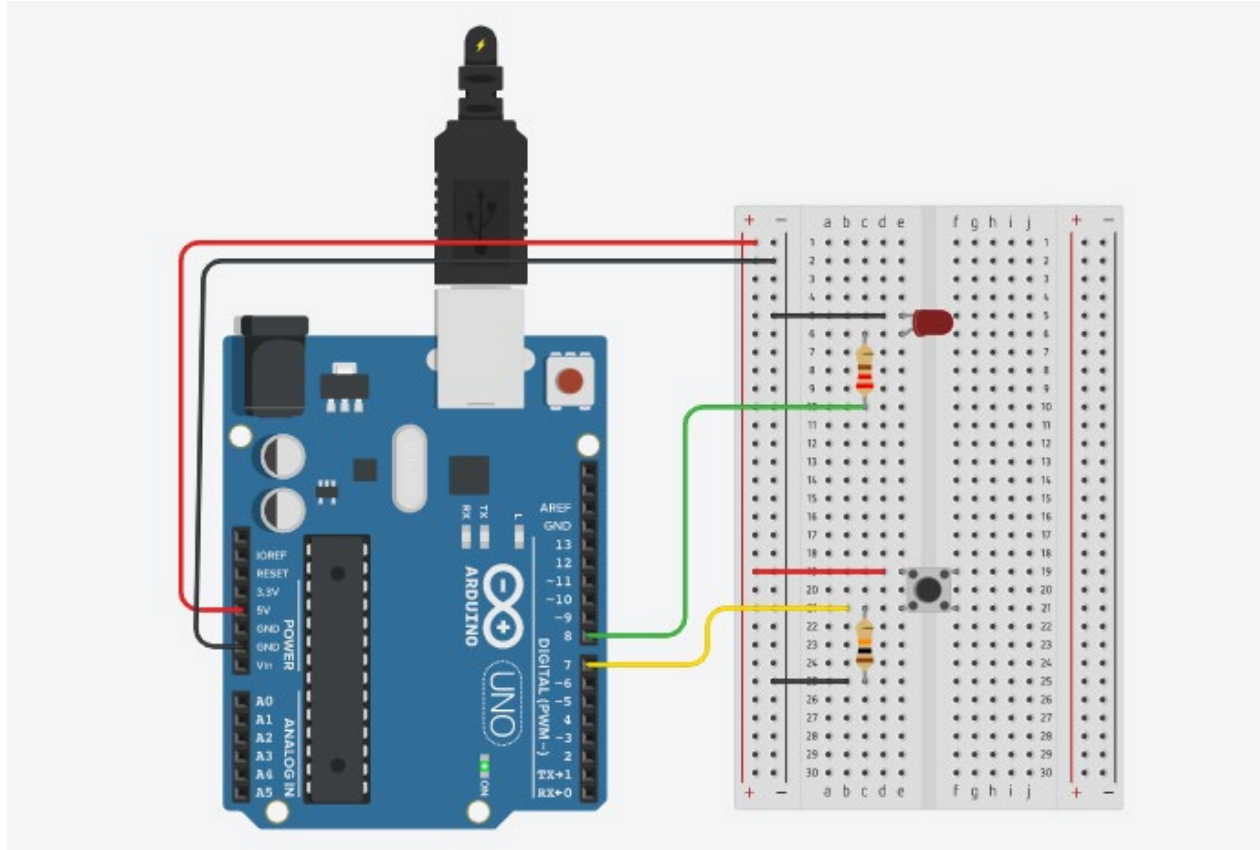


토글(Toggle)제어

- 2가지 상태를 번갈아 가며 제어하는 방식
 - 참이면 거짓으로 거짓이면 참으로 바꿈
- 버튼을 누를 때마다 LED ON/OFF 제어
- 에지(Edge) 입력
 - 버튼 누름 상태가 아닌 누르거나 떼는 것을 체크하는 것
 - Button Down : 누르는 순간
 - Button Up : 떼는 순간



토글(Toggle) 제어



Ex A4-1 : Toggle

```
#define ButtonPin 7
#define LED_PIN 11

int led_state = LOW;
int pre_button_state = HIGH;

void setup( )
{
    pinMode(LED_PIN, OUTPUT);
    pinMode(ButtonPin, INPUT);
}
```

```
void loop( )
{
    int button_state = digitalRead(ButtonPin);

    if (button_state != pre_button_state )
    {
        if(button_state == HIGH && pre_button_state == LOW )
        {
            if(led_state == LOW)
                led_state = HIGH;
            else
                led_state = LOW;
        }
        pre_button_state = button_state;
    }
    digitalWrite(LED_PIN,led_state);
}
```



3. 밝기 조절

- ✓ LED에 입력되는 전력은 PWM(Pulse Width Modulation)을 이용하여 조절.
- ✓ PWM : 고속의 스위칭으로 High와 Low 신호의 비율을 조절하여 LED의 밝기, 모터의 회전 등을 조절하는 방법
- ✓ Arduino에서는 analogWrite() 명령어로 구현
- ✓ Arduino UNO의 경우 3, 5, 6, 9, 10, 11 번 핀이 PWM을 지원한다.



PWM

❖ 펄스 폭 변조(PWM)

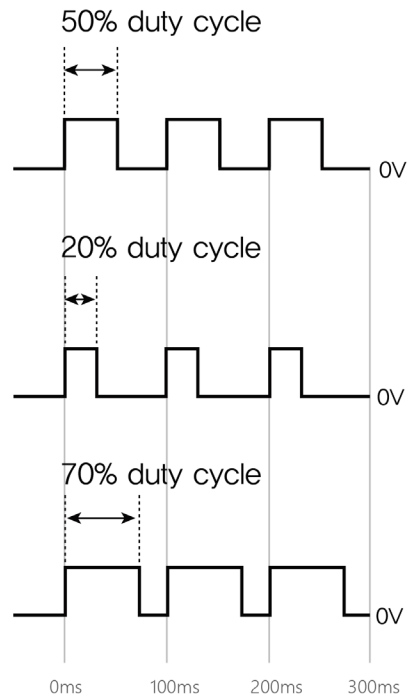
디지털 신호를 사용하여 아날로그와 유사한 신호를 만드는 방법으로, PWM을 사용하면 디지털 신호를 아날로그 신호와 유사하게 표현해 줄 수 있다.

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



FUNCTIONS

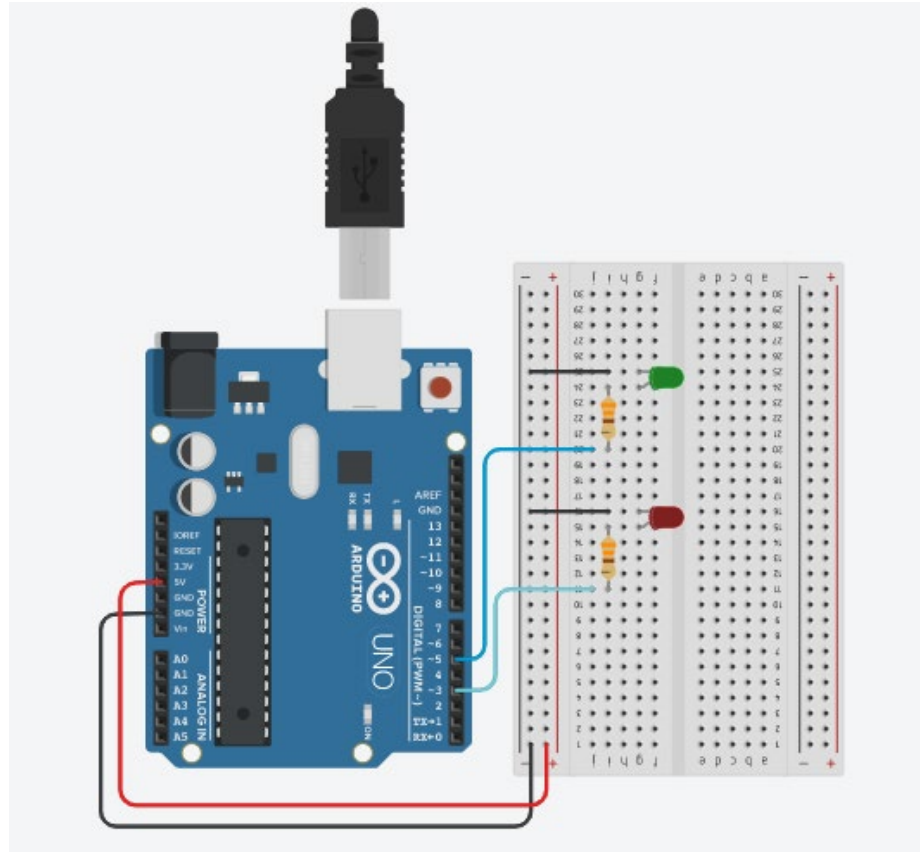
❖ 함수 설명

`analogWrite(` ^{사용할} _{PWM 핀의 번호} ^{제어할 값} _(0 ~ 255) `);`

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



Ex 5-1 : SerialWrite




```
int ledA  = 3;           //LED A를 3번핀에 연결
int ledB  = 5;           //LED B를 5번핀에 연결
int brightness = 0;      //밝기를 조절하기 위한 변수
int increment = 1;       //밝기 변수 증감을 위한 변수
```

```
void setup( )
```

```
{
// analogWrite 핀에는 별도의 설정이 불필요하다.
}
```

```
void loop( )
```

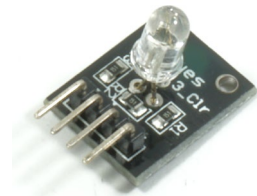
```
{
  analogWrite(ledA,brightness);           // LED A 밝기 조절
  analogWrite(ledB,255-brightness);       // LED B 밝기 조절

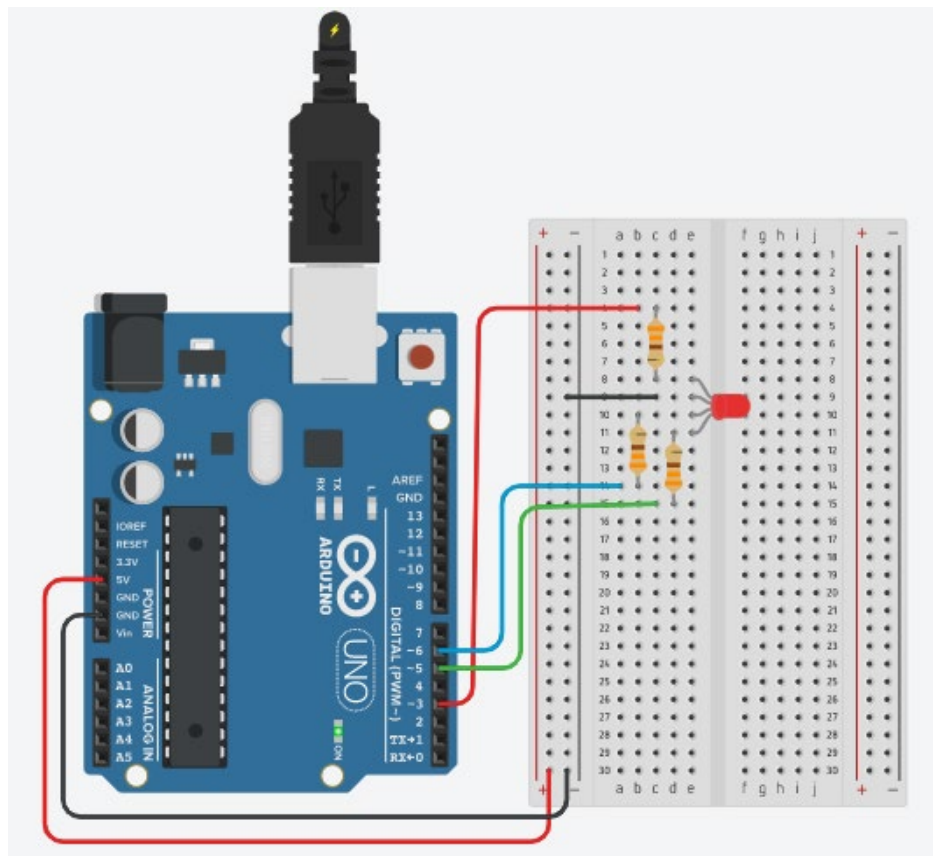
  brightness = brightness + increment;    // 밝기 조절
  if((brightness >= 255)|| (brightness <= 0)) increment = -increment; // 밝기 변수 증감 방향 변경
  delay(10); // 0.01 초간 지연
}
```



RGB LED

- ✓ 빛의 삼원색인 빨강(Red), 초록(Green), 파랑(Blue)빛을 조절하여 다양한 색을 표현하는 LED.
- ✓ 각각의 색이 0~255단계로 조절됨.
- ✓ 간판, 조명기구 등에 사용
- ✓ 모든 색이 출력될 때 백색 빛을 출력





RGB LED

```
int RedLed    = 3;    //LED R를 3번핀에 연결
int GreenLed  = 5;    //LED G를 5번핀에 연결
int BlueLed   = 6;    //LED B를 6번핀에 연결
```

void setup()

```
{
  ledOutput(255, 0, 0);
  delay(1000);
  ledOutput(0, 255, 0);
  delay(1000);
  ledOutput(0, 0, 255);
  delay(1000);
}
```

void loop() {

```
  for(int i=0;i<=255;++i) {
    ledOutput(255,i,0);
    delay(10); }
  for(int i=0;i<=255;++i) {
    ledOutput(0,255,i);
    delay(10); }
```

```
  for(int i=0;i<=255;++i) {
    ledOutput(i,0, 255);
    delay(10); }
  for(int i=0;i<=255;++i) {
    ledOutput(i, 255, 255);
    delay(10); }
  for(int i=0;i<=255;++i) {
    ledOutput(255, i, 255);
    delay(10); }
  for(int i=0;i<=255;++i) {
    ledOutput(255, 255, i);
    delay(10); }
}
```

void ledOutput(int Red, int Green, int Blue) {

```
  analogWrite(RedLed,Red);
  analogWrite(GreenLed,Green);
  analogWrite(BlueLed,Blue);
}
```

