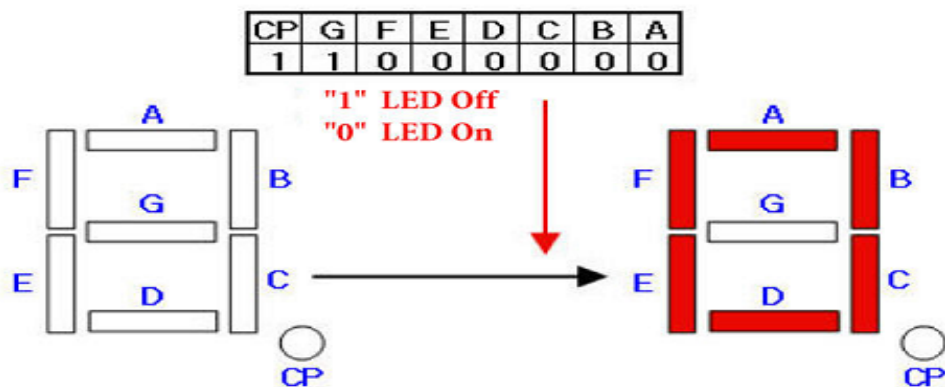
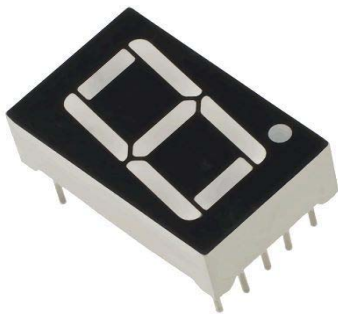


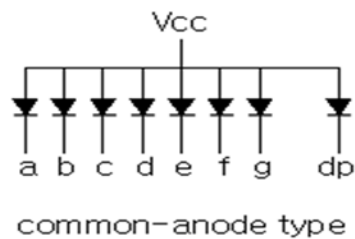
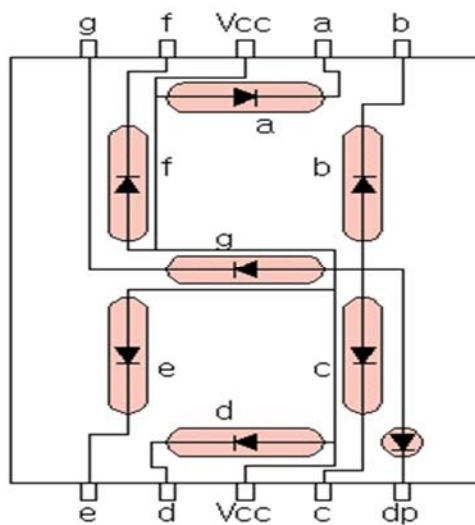
실험 4. 7-Segment 실험

4-1. 7-Segment 란?

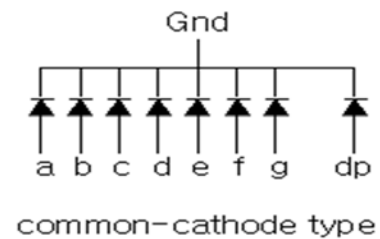
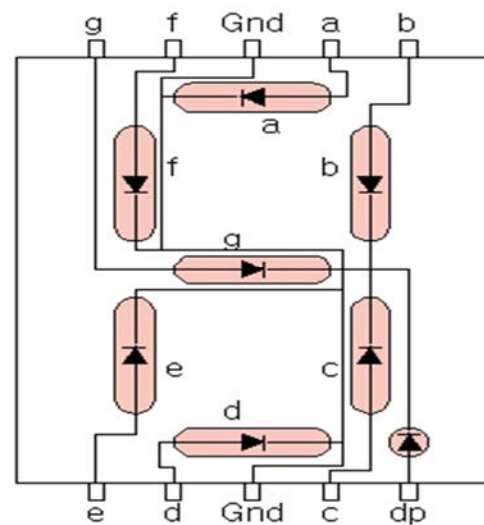
4-1-1. 7-Segment (FND: Flexible Numeric Display)



4-1-2. 7-Segment Technology



common-anode type



common-cathode type

1) Common Cathode: 출력을 위해 "1" 출력

▣ 예) 숫자 "2"를 출력하려면

a, b, d, e, g 는 "1", c, f, dp는 "0"을 출력

2) Common Anode: 출력을 위해 "0" 출력

▣ 예) 숫자 "2"를 출력하려면

a, b, d, e, g 는 "0", c, f, dp는 "1"을 출력

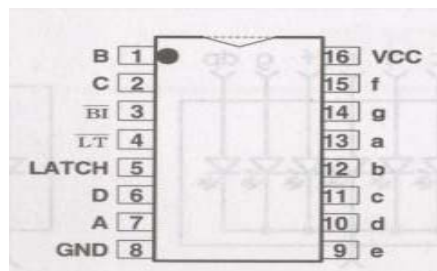
3) 7-Segment Operation

① 7-Segment 동작방법

- Static Method: Hardware 복잡, Software 간단
- Dynamic Method: Hardware 간단, TTL Driver 줄일 수 있음, Software 복잡

② 7-Segment Driver 예: 7-Segment를 구동하기 위하여 필요한 TTL IC

- 4511(Common Cathod 7-Segment Driver):
- BCD to Seven Segment Decoder
- 4511 Function Table: (1: +5[V], 0: 0[V], x : don't care)

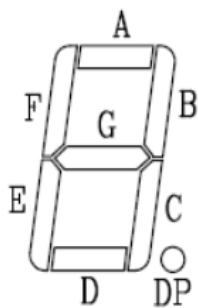


③ 숫자에 대응하는 LED 값 (Common Anode 예)

	DP	G	F	E	D	C	B	A	HEX
0	1	1	0	0	0	0	0	0	0xC0
1	1	1	1	1	1	0	0	1	0xF9
2	1	0	1	0	0	1	0	0	0xA4
3	1	0	1	1	0	0	0	0	0xB0
4	1	0	0	1	1	0	0	1	0x99
5	1	0	0	1	0	0	1	0	0x92

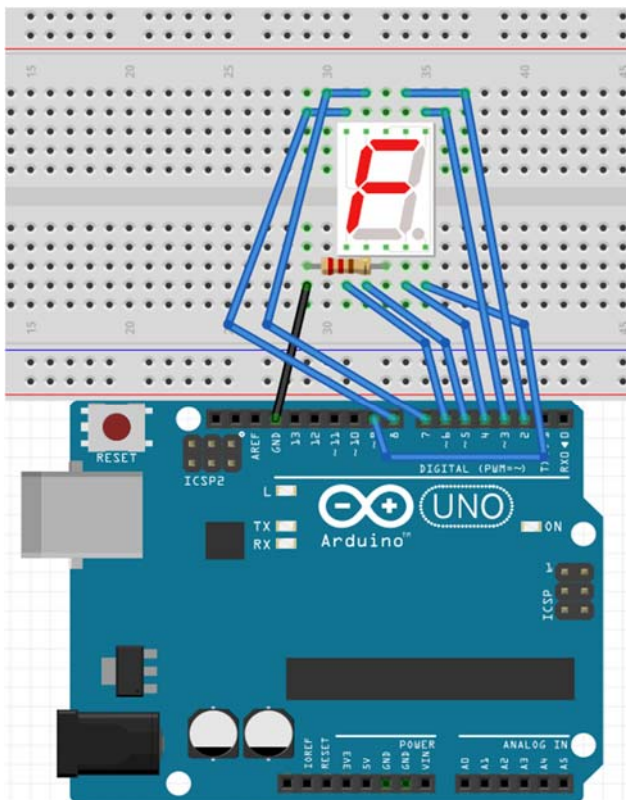
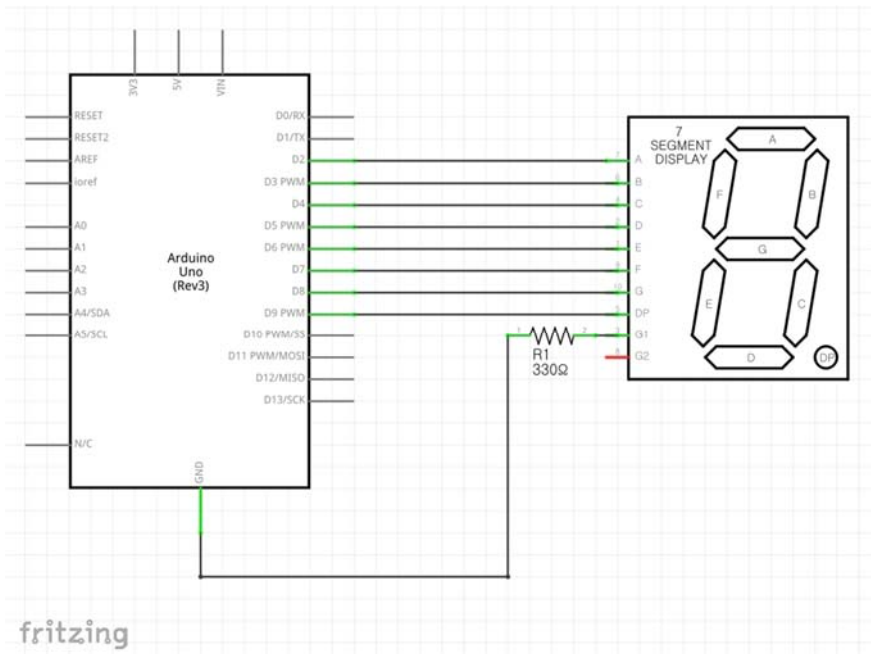
6	1	0	0	0	0	0	1	0	0x82
7	1	1	0	1	1	0	0	0	0xD8
8	1	0	0	0	0	0	0	0	0x80
9	1	0	0	1	0	0	0	0	0x90

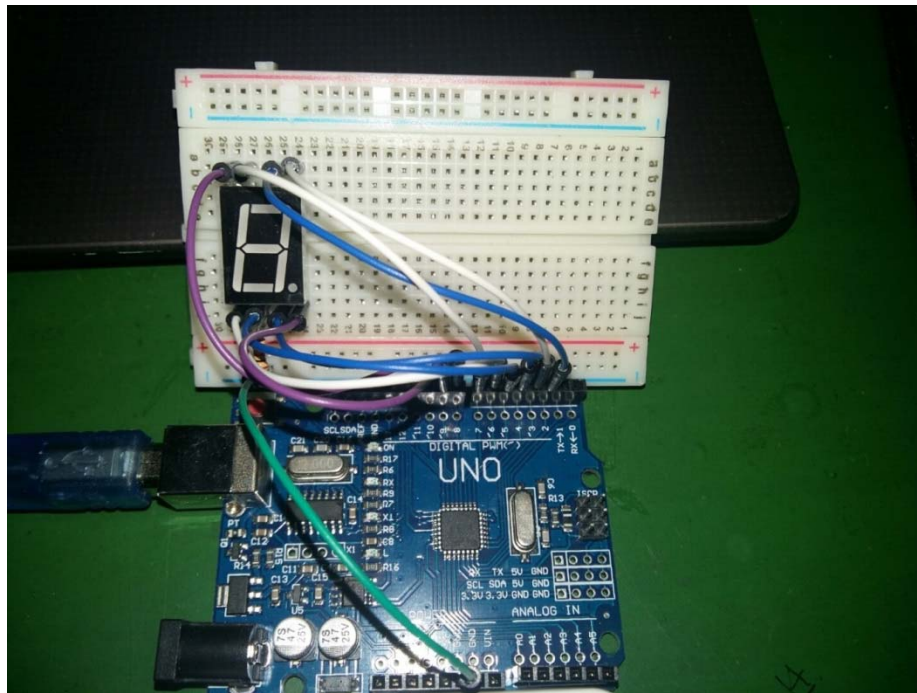
- 숫자에 대응하는 LED 값 (Common Cathod 예)



문자	a	b	c	d	e	f	g	dp
숫자	(PC5)	(PC4)	(PC2)	(PC1)	(PC0)	(PC6)	(PC7)	(PC3)
0	1	1	1	1	1	1	0	0
1	0	1	1	0	0	0	0	0
2	1	1	0	1	1	0	1	0
3	1	1	1	1	0	0	1	0
4	0	1	1	0	0	1	1	0
5	1	0	1	1	0	1	1	0
6	0	0	1	1	1	1	1	0
7	1	1	1	0	0	0	0	0
8	1	1	1	1	1	1	1	0
9	1	1	1	0	0	1	1	0
a	1	1	1	1	1	0	1	0
b	0	1	0	1	1	1	1	0
c	1	0	0	1	1	1	0	0
d	0	1	1	1	1	0	1	0
e	1	0	0	1	1	1	1	0
f	1	0	0	0	1	1	1	0

4.2 7-Segment 를 이용한 I/O 실험 회로도, 배선도 및 결선도





7-SEGMENT와 아두이노를 연결한다. 회로 구성 시 7-SEGMENT의 Vcc에 330 Ω 저항을 연결해 준다. 모든 세그먼트 포트에 330 Ω 저항을 각각 연결해도 무관하다. 저항을 연결해 주는 것은 단지 전류조절을 해주기 위함이다. 포트가 많으니 배선에 주의해서 연결하도록

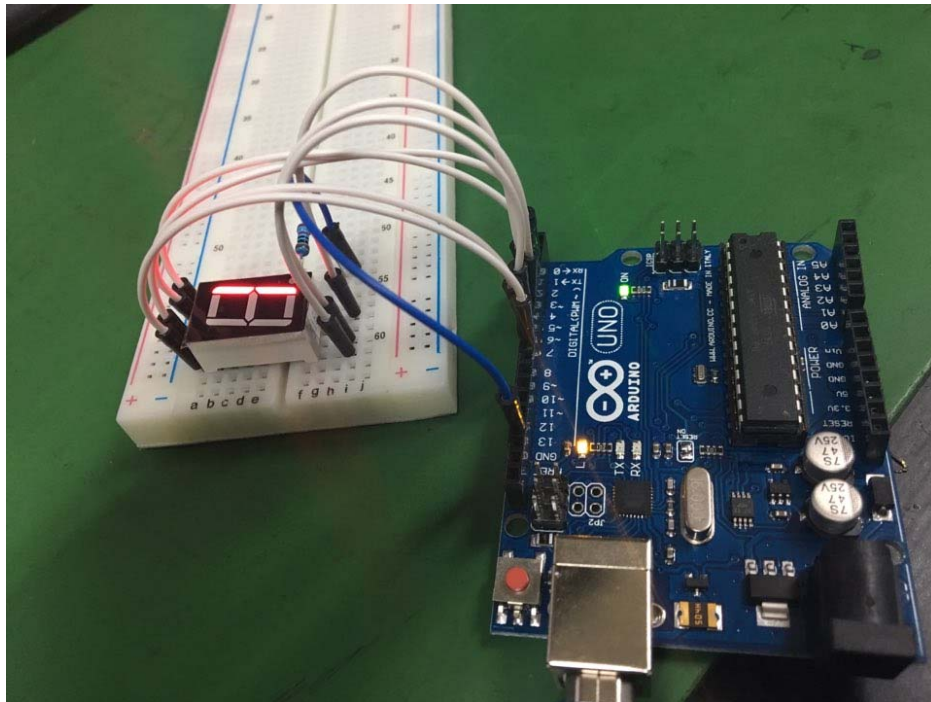
4.3 Program

예제 1) 7-segment 로 숫자 1 Display 하기

```
int ledA = 2; //각각의 세그먼트를 연결할 아두이노 포트 번호 지정
int ledB = 3;
int ledC = 4;
int ledD = 5;
int ledE = 6;
int ledF = 7;
int ledG = 8;
int DOT = 9;

void setup(){
    pinMode(ledA, OUTPUT); //아두이노 포트를 출력포트로 설정
    pinMode(ledB, OUTPUT);
    pinMode(ledC, OUTPUT);
    pinMode(ledD, OUTPUT);
    pinMode(ledE, OUTPUT);
    pinMode(ledF, OUTPUT);
    pinMode(ledG, OUTPUT);
    pinMode(DOT, OUTPUT);
}

void loop(){ //7-SEGMENT로 숫자 1 출력하기
    digitalWrite(ledA, LOW);
    digitalWrite(ledB, HIGH);
    digitalWrite(ledC, HIGH);
    digitalWrite(ledD, LOW);
    digitalWrite(ledE, LOW);
    digitalWrite(ledF, LOW);
    digitalWrite(ledG, LOW);
    digitalWrite(DOT, LOW);
}
```



[출력결과]

예제 2) A~F, 0~9 까지 순서대로 switch ~ case 문을 이용하여 출력하는 프로그램 작성하기

```
int ledA = 2; //각각의 세그먼트를 연결할 아두이노 포트 번호 지정
int ledB = 3;
int ledC = 4;
int ledD = 5;
int ledE = 6;
int ledF = 7;
int ledG = 8;
int DOT = 9;

void setup(){
    pinMode(ledA, OUTPUT); //아두이노 포트를 출력포트로 설정
    pinMode(ledB, OUTPUT);
    pinMode(ledC, OUTPUT);
    pinMode(ledD, OUTPUT);
    pinMode(ledE, OUTPUT);
    pinMode(ledF, OUTPUT);
    pinMode(ledG, OUTPUT);
    pinMode(DOT, OUTPUT);
    clr(); //서브함수 호출
    set();
}

void loop(){
    int i;

    for(i=0; i<6; i++){ //알파벳 A~F 출력
        Alphabet(i); //서브함수 호출
        delay(1000);
    }

    for(i=0; i<10; i++){ //숫자 0~9 출력
        number(i); //서브함수 호출
        delay(1000);
    }
}
```

```

    }
}

void Alphabet(int n){          //알바벳 출력 서브함수
    switch(n){
        case 0:                //A 출력
            digitalWrite(ledA, HIGH);
            digitalWrite(ledB, HIGH);
            digitalWrite(ledC, HIGH);
            digitalWrite(ledD, LOW);
            digitalWrite(ledE, HIGH);
            digitalWrite(ledF, HIGH);
            digitalWrite(ledG, HIGH);
            digitalWrite(DOT, LOW);
            break;

        case 1:                //B 출력
            digitalWrite(ledA, HIGH);
            digitalWrite(ledB, HIGH);
            digitalWrite(ledC, HIGH);
            digitalWrite(ledD, HIGH);
            digitalWrite(ledE, HIGH);
            digitalWrite(ledF, HIGH);
            digitalWrite(ledG, HIGH);
            digitalWrite(DOT, LOW);
            break;

        case 2:                //C 출력
            digitalWrite(ledA, HIGH);
            digitalWrite(ledB, LOW);
            digitalWrite(ledC, LOW);
            digitalWrite(ledD, HIGH);
            digitalWrite(ledE, HIGH);

```

```
digitalWrite(ledF, HIGH);
digitalWrite(ledG, LOW);
digitalWrite(DOT, LOW);
break;

case 3:                                //D 출력
digitalWrite(ledA, HIGH);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(ledD, HIGH);
digitalWrite(ledE, HIGH);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, LOW);
digitalWrite(DOT, LOW);
break;

case 4:                                //E 출력
digitalWrite(ledA, HIGH);
digitalWrite(ledB, LOW);
digitalWrite(ledC, LOW);
digitalWrite(ledD, HIGH);
digitalWrite(ledE, HIGH);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(DOT, LOW);
break;

case 5:                                //F 출력
digitalWrite(ledA, HIGH);
digitalWrite(ledB, LOW);
digitalWrite(ledC, LOW);
digitalWrite(ledD, LOW);
digitalWrite(ledE, HIGH);
```

```

    digitalWrite(ledF, HIGH);
    digitalWrite(ledG, HIGH);
    digitalWrite(DOT, LOW);
    break;
}
}

void number(int n){                                //숫자 출력 서브함수
    switch(n){
        case 0:                                    //0 출력
            digitalWrite(ledA, HIGH);
            digitalWrite(ledB, HIGH);
            digitalWrite(ledC, HIGH);
            digitalWrite(ledD, HIGH);
            digitalWrite(ledE, HIGH);
            digitalWrite(ledF, HIGH);
            digitalWrite(ledG, LOW);
            digitalWrite(DOT, LOW);
            break;

            case 1:                                //1 출력
                digitalWrite(ledA, LOW);
                digitalWrite(ledB, HIGH);
                digitalWrite(ledC, HIGH);
                digitalWrite(ledD, LOW);
                digitalWrite(ledE, LOW);
                digitalWrite(ledF, LOW);
                digitalWrite(ledG, LOW);
                digitalWrite(DOT, LOW);
                break;

            case 2:                                //2 출력
                digitalWrite(ledA, HIGH);

```



```
digitalWrite(ledB, LOW);
digitalWrite(ledC, HIGH);
digitalWrite(ledD, HIGH);
digitalWrite(ledE, LOW);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(DOT, LOW);
break;

case 6: //6 출력
digitalWrite(ledA, HIGH);
digitalWrite(ledB, LOW);
digitalWrite(ledC, HIGH);
digitalWrite(ledD, HIGH);
digitalWrite(ledE, HIGH);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(DOT, LOW);
break;

case 7: //7 출력
digitalWrite(ledA, HIGH);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(ledD, LOW);
digitalWrite(ledE, LOW);
digitalWrite(ledF, LOW);
digitalWrite(ledG, LOW);
digitalWrite(DOT, LOW);
break;

case 8: //8 출력
digitalWrite(ledA, HIGH);
```

```
digitalWrite(ledB, HIGH);  
digitalWrite(ledC, HIGH);  
digitalWrite(ledD, HIGH);  
digitalWrite(ledE, HIGH);  
digitalWrite(ledF, HIGH);  
digitalWrite(ledG, HIGH);  
digitalWrite(DOT, LOW);  
break;
```

```
case 9:                                     //9 출력
```

```
digitalWrite(ledA, HIGH);  
digitalWrite(ledB, HIGH);  
digitalWrite(ledC, HIGH);  
digitalWrite(ledD, HIGH);  
digitalWrite(ledE, LOW);  
digitalWrite(ledF, HIGH);  
digitalWrite(ledG, HIGH);  
digitalWrite(DOT, LOW);  
break;
```

```
default:
```

```
digitalWrite(ledA, LOW);  
digitalWrite(ledB, LOW);  
digitalWrite(ledC, LOW);  
digitalWrite(ledD, LOW);  
digitalWrite(ledE, LOW);  
digitalWrite(ledF, LOW);  
digitalWrite(ledG, LOW);  
digitalWrite(DOT, LOW);  
break;
```

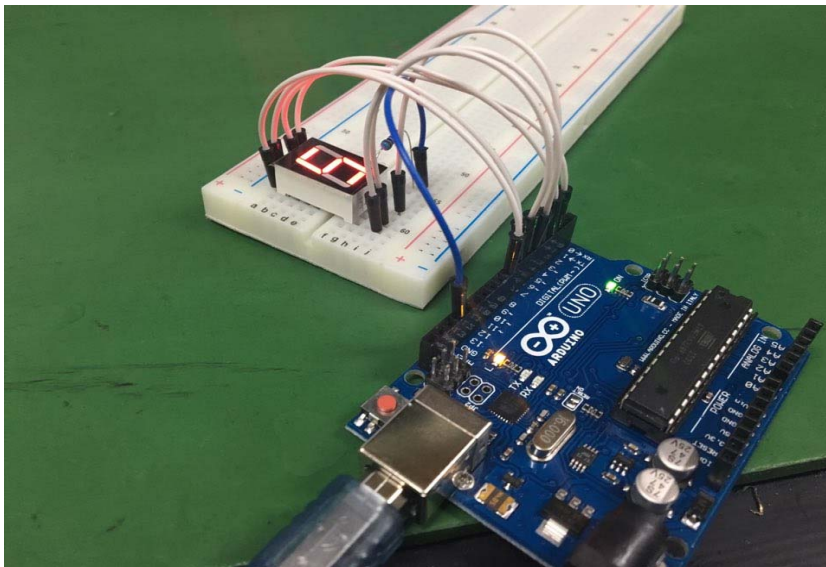
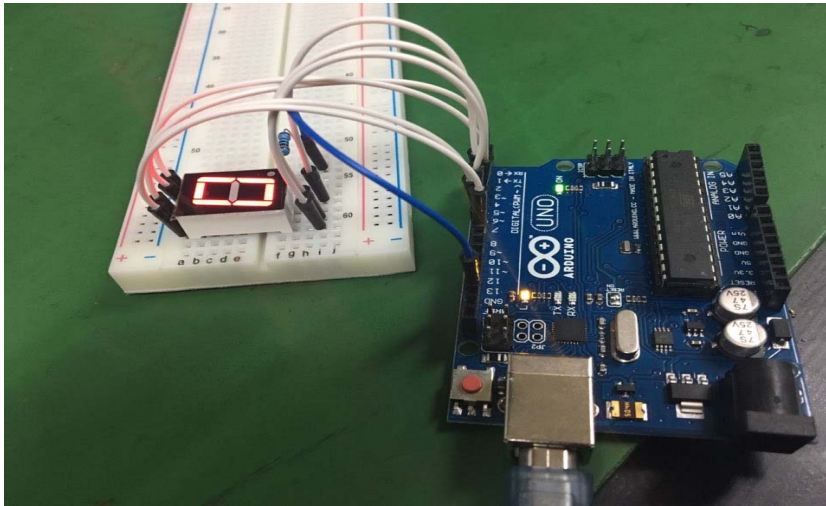
```
}
```

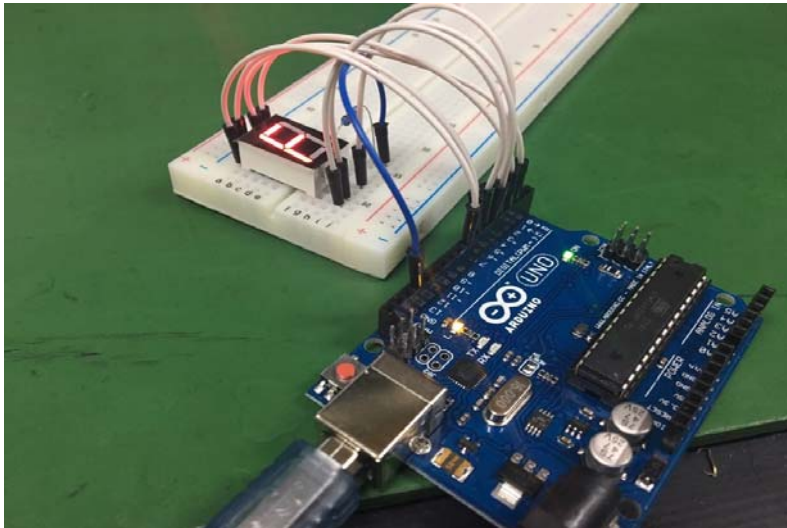
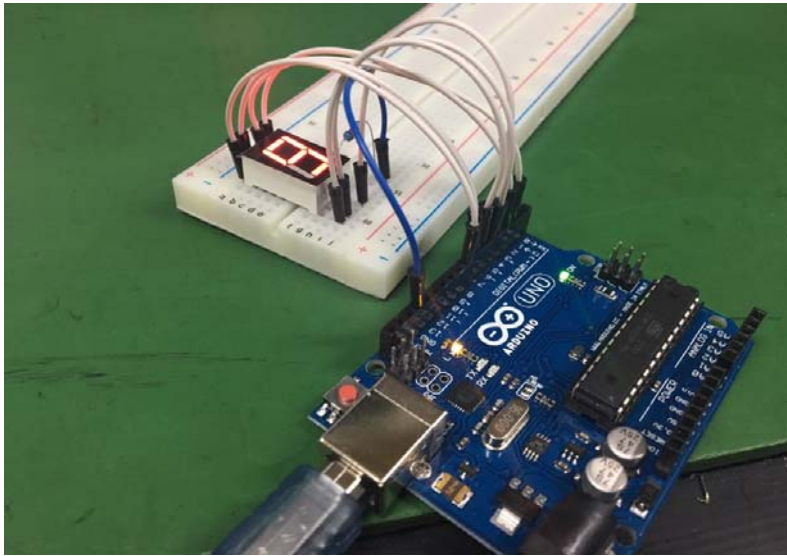
```
}
```

```
void clr(){ //Clear 서브함수
    digitalWrite(ledA, LOW);
    digitalWrite(ledB, LOW);
    digitalWrite(ledC, LOW);
    digitalWrite(ledD, LOW);
    digitalWrite(ledE, LOW);
    digitalWrite(ledF, LOW);
    digitalWrite(ledG, LOW);
    digitalWrite(DOT, LOW);
    delay(500);
}

void set(){ //setting 서브함수
    digitalWrite(ledA, HIGH);
    digitalWrite(ledB, HIGH);
    digitalWrite(ledC, HIGH);
    digitalWrite(ledD, HIGH);
    digitalWrite(ledE, HIGH);
    digitalWrite(ledF, HIGH);
    digitalWrite(ledG, HIGH);
    digitalWrite(DOT, HIGH);
    delay(2000);
}
```


- *clr();*
- *switch(n);* case문
- *break;*
- *default:* 스위치문에서 입력된 값과 대응되는 case문이 없을 때





예제 3) 7-segment 의 숫자를 0, 2, 4, 6, 8 짝수 숫자를 7-segment 에 차례대로 반복하여 표시하는 프로그램 작성하기

```
/*7-SEGMENT에 0,2,4,6,8의 짝수 숫자를 반복해서 display한다.*/

int ledA = 2; //각각의 세그먼트를 연결할 아두이노 포트 번호 지정
int ledB = 3;
int ledC = 4;
int ledD = 5;
int ledE = 6;
int ledF = 7;
int ledG = 8;
int DOT = 9;

void setup(){
    pinMode(ledA, OUTPUT); //아두이노 포트를 출력포트로 설정
    pinMode(ledB, OUTPUT);
    pinMode(ledC, OUTPUT);
    pinMode(ledD, OUTPUT);
    pinMode(ledE, OUTPUT);
    pinMode(ledF, OUTPUT);
    pinMode(ledG, OUTPUT);
    pinMode(DOT, OUTPUT);
    clr(); //동작확인을 위한 서브함수 호출
    set();
}

void loop(){
    int i;

    for(i=0; i<5; i++){ //숫자 0,2,4,6,8 출력
        number(i*2); //서브함수 호출
        delay(1000);
    }
}
```

```

void number(int n){                                //숫자 출력 서브함수
    switch(n){
        case 0:                                    //0 출력
            digitalWrite(ledA, HIGH);
            digitalWrite(ledB, HIGH);
            digitalWrite(ledC, HIGH);
            digitalWrite(ledD, HIGH);
            digitalWrite(ledE, HIGH);
            digitalWrite(ledF, HIGH);
            digitalWrite(ledG, LOW);
            digitalWrite(DOT, LOW);
            break;

        case 1:                                    //1 출력
            digitalWrite(ledA, LOW);
            digitalWrite(ledB, HIGH);
            digitalWrite(ledC, HIGH);
            digitalWrite(ledD, LOW);
            digitalWrite(ledE, LOW);
            digitalWrite(ledF, LOW);
            digitalWrite(ledG, LOW);
            digitalWrite(DOT, LOW);
            break;

        case 2:                                    //2 출력
            digitalWrite(ledA, HIGH);
            digitalWrite(ledB, HIGH);
            digitalWrite(ledC, LOW);
            digitalWrite(ledD, HIGH);
            digitalWrite(ledE, HIGH);
            digitalWrite(ledF, LOW);
            digitalWrite(ledG, HIGH);
            digitalWrite(DOT, LOW);
    }
}

```

```
break;
```

```
case 3: //3 출력
```

```
digitalWrite(ledA, HIGH);  
digitalWrite(ledB, HIGH);  
digitalWrite(ledC, HIGH);  
digitalWrite(ledD, HIGH);  
digitalWrite(ledE, LOW);  
digitalWrite(ledF, LOW);  
digitalWrite(ledG, HIGH);  
digitalWrite(DOT, LOW);  
break;
```

```
case 4: //4 출력
```

```
digitalWrite(ledA, LOW);  
digitalWrite(ledB, HIGH);  
digitalWrite(ledC, HIGH);  
digitalWrite(ledD, LOW);  
digitalWrite(ledE, LOW);  
digitalWrite(ledF, HIGH);  
digitalWrite(ledG, HIGH);  
digitalWrite(DOT, LOW);  
break;
```

```
case 5: //5 출력
```

```
digitalWrite(ledA, HIGH);  
digitalWrite(ledB, LOW);  
digitalWrite(ledC, HIGH);  
digitalWrite(ledD, HIGH);  
digitalWrite(ledE, LOW);  
digitalWrite(ledF, HIGH);  
digitalWrite(ledG, HIGH);  
digitalWrite(DOT, LOW);
```



```
break;
```

```
case 6: //6 출력
```

```
digitalWrite(ledA, HIGH);  
digitalWrite(ledB, LOW);  
digitalWrite(ledC, HIGH);  
digitalWrite(ledD, HIGH);  
digitalWrite(ledE, HIGH);  
digitalWrite(ledF, HIGH);  
digitalWrite(ledG, HIGH);  
digitalWrite(DOT, LOW);  
break;
```

```
case 7: //7 출력
```

```
digitalWrite(ledA, HIGH);  
digitalWrite(ledB, HIGH);  
digitalWrite(ledC, HIGH);  
digitalWrite(ledD, LOW);  
digitalWrite(ledE, LOW);  
digitalWrite(ledF, LOW);  
digitalWrite(ledG, LOW);  
digitalWrite(DOT, LOW);  
break;
```

```
case 8: //8 출력
```

```
digitalWrite(ledA, HIGH);  
digitalWrite(ledB, HIGH);  
digitalWrite(ledC, HIGH);  
digitalWrite(ledD, HIGH);  
digitalWrite(ledE, HIGH);  
digitalWrite(ledF, HIGH);  
digitalWrite(ledG, HIGH);  
digitalWrite(DOT, LOW);
```

```

break;

case 9:                                //9 출력
digitalWrite(ledA, HIGH);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(ledD, HIGH);
digitalWrite(ledE, LOW);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(DOT, LOW);
break;

default:
digitalWrite(ledA, LOW);
digitalWrite(ledB, LOW);
digitalWrite(ledC, LOW);
digitalWrite(ledD, LOW);
digitalWrite(ledE, LOW);
digitalWrite(ledF, LOW);
digitalWrite(ledG, LOW);
digitalWrite(DOT, LOW);
break;
}
}

void clr(){                            //Clear 서브함수
digitalWrite(ledA, LOW);
digitalWrite(ledB, LOW);
digitalWrite(ledC, LOW);
digitalWrite(ledD, LOW);
digitalWrite(ledE, LOW);
digitalWrite(ledF, LOW);

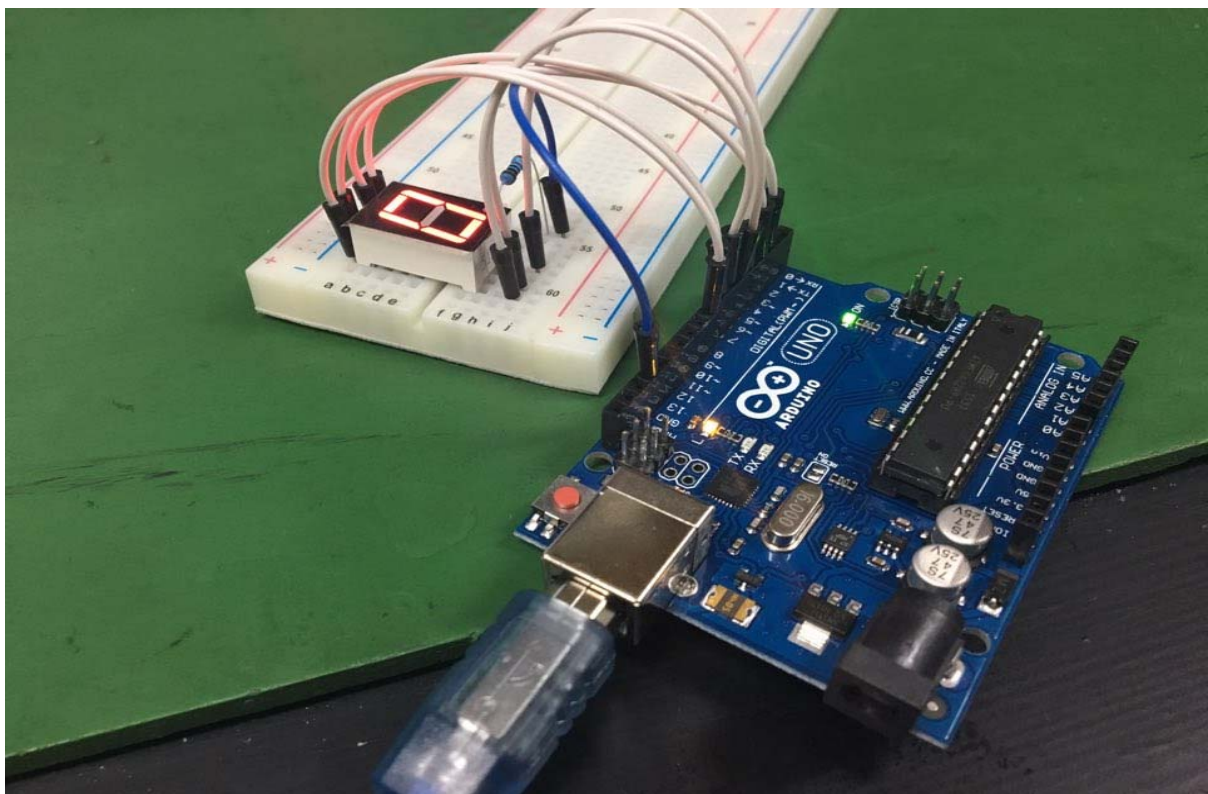
```

```

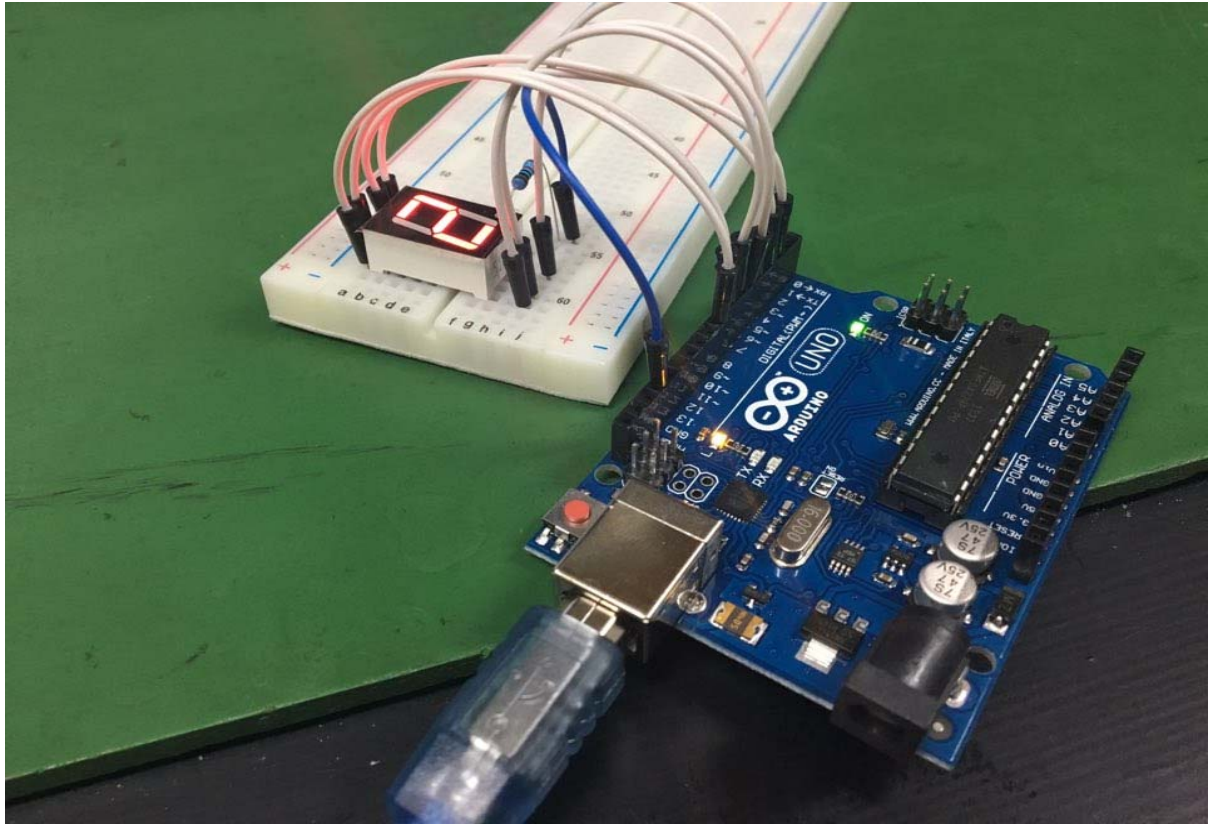
digitalWrite(ledG, LOW);
digitalWrite(DOT, LOW);
delay(500);
}

void set(){                                     //setting 서브함수
    digitalWrite(ledA, HIGH);
    digitalWrite(ledB, HIGH);
    digitalWrite(ledC, HIGH);
    digitalWrite(ledD, HIGH);
    digitalWrite(ledE, HIGH);
    digitalWrite(ledF, HIGH);
    digitalWrite(ledG, HIGH);
    digitalWrite(DOT, HIGH);
    delay(2000);
}

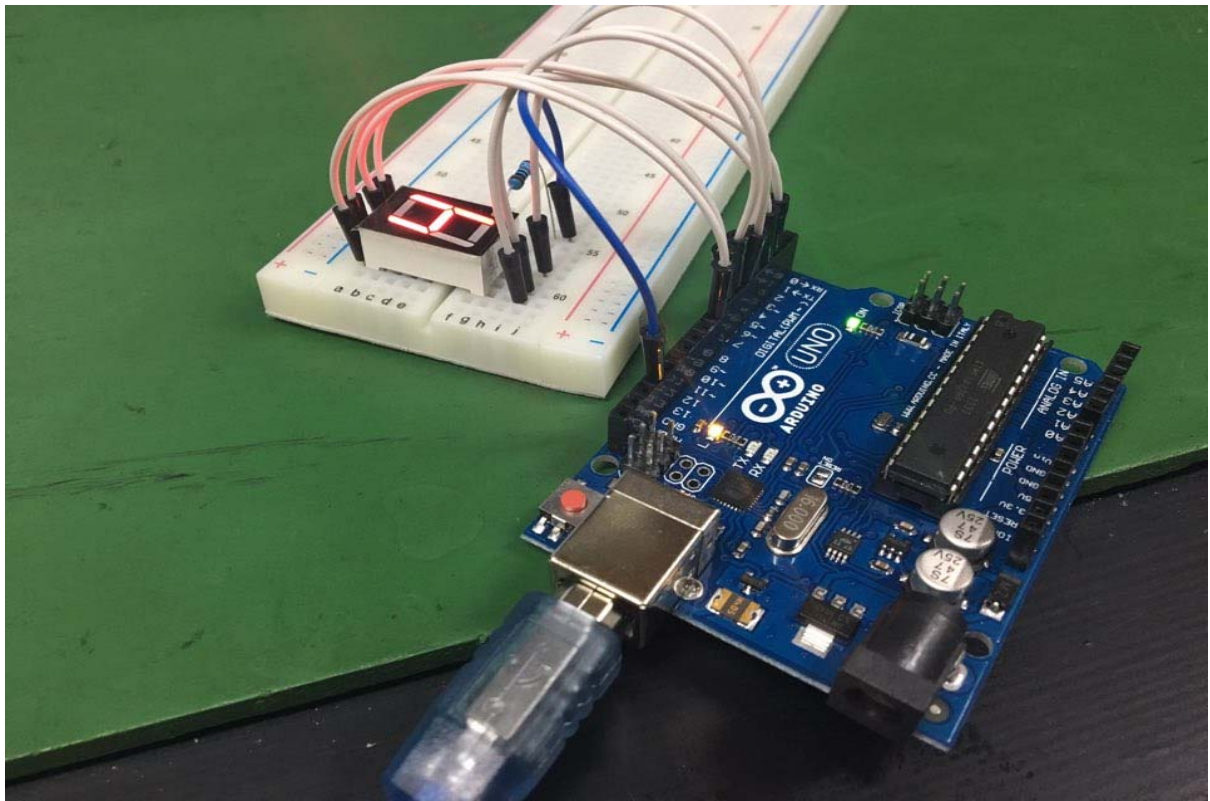
```



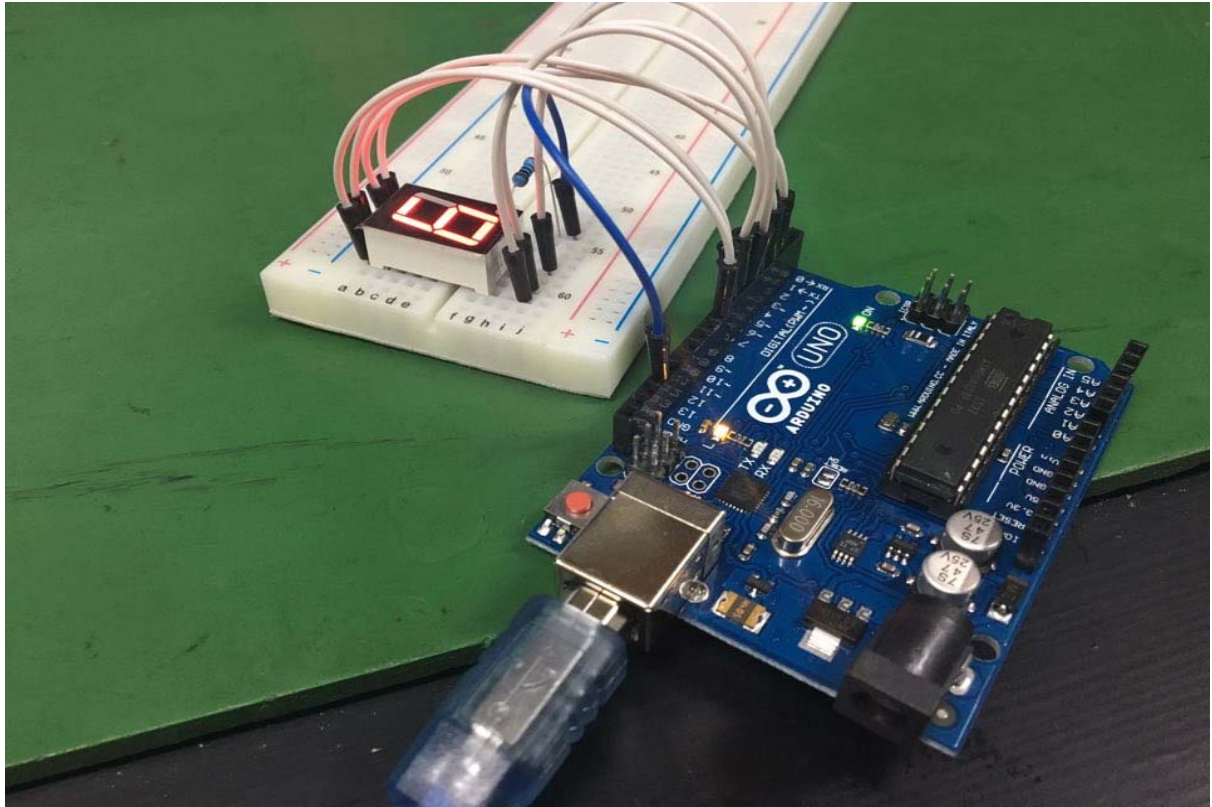
[프로그램 실행 결과1]



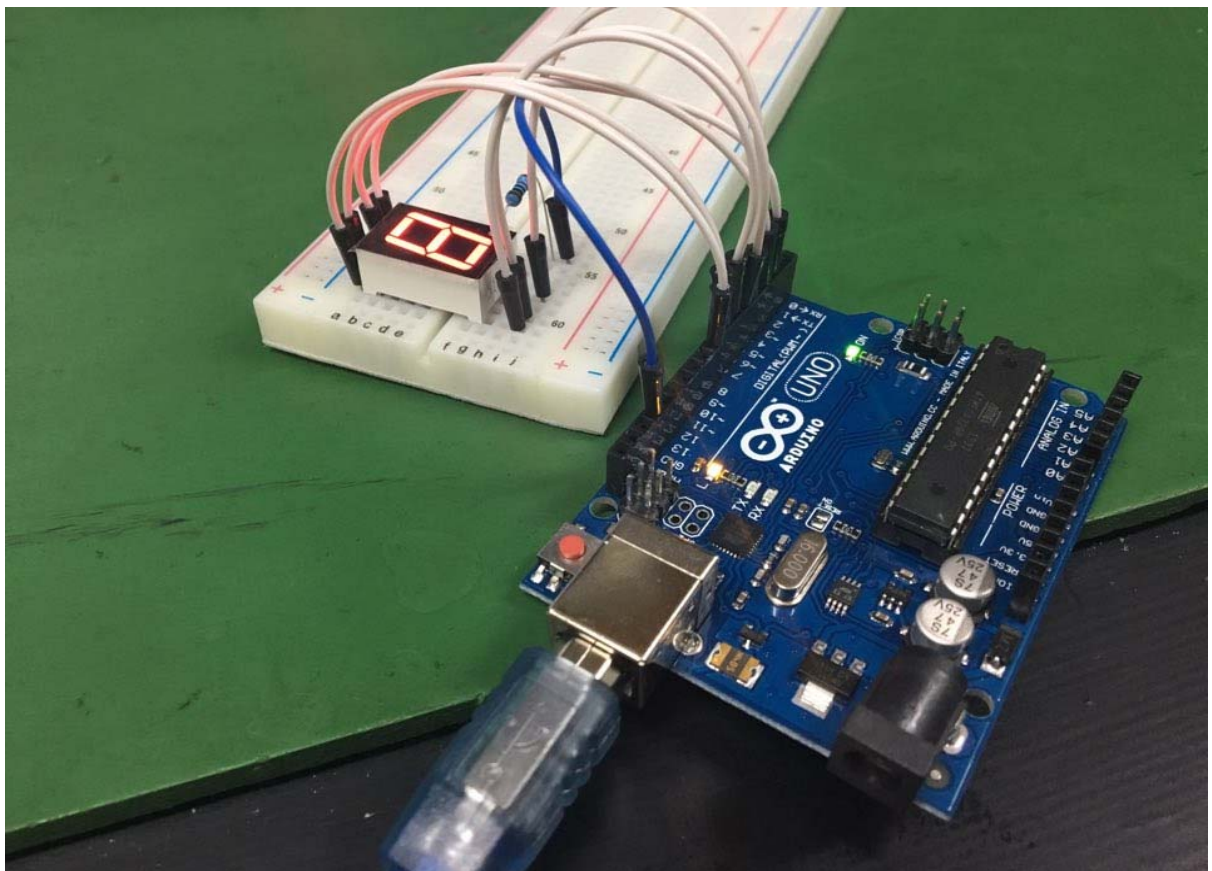
[프로그램 실행 결과2]



[프로그램 실행 결과3]



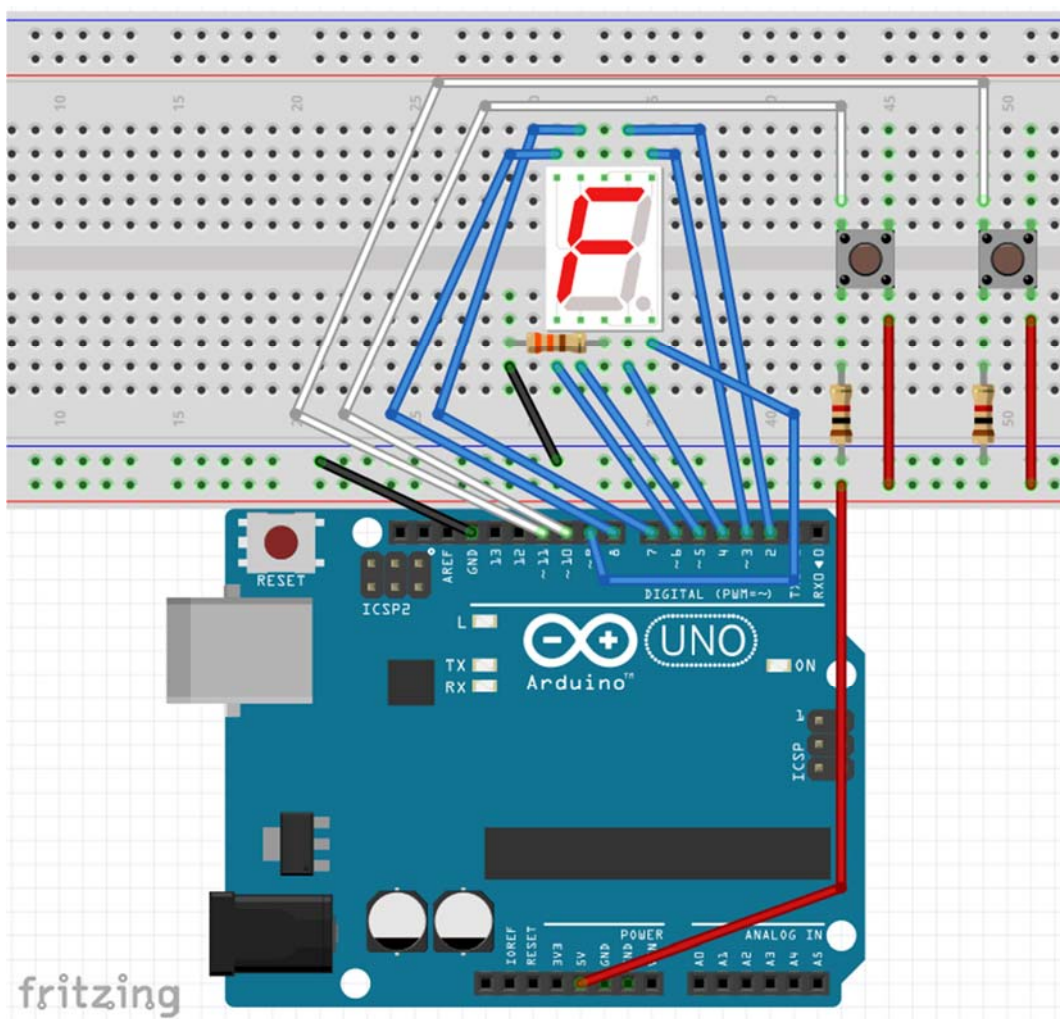
[프로그램 실행 결과4]



[프로그램 실행 결과5]

예제 4) 기존 회로도에 스위치 2개를 추가한 후 아래 조건에 맞도록 프로그램을 작성한다.

- 1) 스위치1을 누르면 7-SEGMENT에서 숫자 1,3,5,7,9를 순서대로 한 번 출력 후 5 에서 대기
- 2) 스위치2를 누르면 7-SEGMENT에서 숫자 0,2,4,6,8을 순서대로 한 번 출력 후 5 에서 대기



[배선도]



```
/* 1.디지털 출력 10번, 11번핀에 스위치1, 스위치2를 연결한다.  
2.스위치 1을 누르면 7-SEGMENT가 1,3,5,7,9의  
   홀수 숫자를 순서대로 한 번 출력한 후 5로 대기  
3.스위치 2를 누르면 7-SEGMENT가 0,2,4,6,8의  
   짝수 숫자를 순서대로 한 번 출력한 후 5로 대기*/
```

```
int ledA = 2; //각각의 세그먼트를 연결할 아두이노 포트 번호 지정  
int ledB = 3;  
int ledC = 4;  
int ledD = 5;  
int ledE = 6;  
int ledF = 7;  
int ledG = 8;  
int DOT = 9;  
int sw1 = 10;  
int sw2 = 11;  
  
void setup(){  
    pinMode(ledA, OUTPUT); //아두이노 포트를 출력포트로 설정  
    pinMode(ledB, OUTPUT);  
    pinMode(ledC, OUTPUT);  
    pinMode(ledD, OUTPUT);  
    pinMode(ledE, OUTPUT);  
    pinMode(ledF, OUTPUT);  
    pinMode(ledG, OUTPUT);  
    pinMode(DOT, OUTPUT);  
    clr(); //서브함수 호출  
    set();  
}  
  
void loop(){  
    int i;  
    int sw1_state;
```

```

sw1_state = digitalRead(sw1);    //스위치 신호 읽어오기
sw2_state = digitalRead(sw2);

if(sw1_state==1){
    for(i=0; i<5; i++){          //숫자 1,3,5,7,9 출력
        number(i*2+1);           //서브함수 호출
        delay(1000);
    }
    delay(1000);
    number(5);
}

if(sw2_state==1){
    for(i=0; i<5; i++){          //숫자 0,2,4,6,8 출력
        number(i*2);             //서브함수 호출
        delay(1000);
    }
    delay(1000);
    number(5);
}

}

void number(int n){              //숫자 출력 서브함수
    switch(n){
        case 0:                  //0 출력
            digitalWrite(ledA, HIGH);
            digitalWrite(ledB, HIGH);
            digitalWrite(ledC, HIGH);
            digitalWrite(ledD, HIGH);
            digitalWrite(ledE, HIGH);
            digitalWrite(ledF, HIGH);

```



```
digitalWrite(ledG, LOW);
digitalWrite(DOT, LOW);
break;

case 1:                                //1 출력
digitalWrite(ledA, LOW);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(ledD, LOW);
digitalWrite(ledE, LOW);
digitalWrite(ledF, LOW);
digitalWrite(ledG, LOW);
digitalWrite(DOT, LOW);
break;

case 2:                                //2 출력
digitalWrite(ledA, HIGH);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, LOW);
digitalWrite(ledD, HIGH);
digitalWrite(ledE, HIGH);
digitalWrite(ledF, LOW);
digitalWrite(ledG, HIGH);
digitalWrite(DOT, LOW);
break;

case 3:                                //3 출력
digitalWrite(ledA, HIGH);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(ledD, HIGH);
digitalWrite(ledE, LOW);
digitalWrite(ledF, LOW);
```

```
digitalWrite(ledG, HIGH);
digitalWrite(DOT, LOW);
break;

case 4:                                //4 출력
digitalWrite(ledA, LOW);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(ledD, LOW);
digitalWrite(ledE, LOW);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(DOT, LOW);
break;

case 5:                                //5 출력
digitalWrite(ledA, HIGH);
digitalWrite(ledB, LOW);
digitalWrite(ledC, HIGH);
digitalWrite(ledD, HIGH);
digitalWrite(ledE, LOW);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(DOT, LOW);
break;

case 6:                                //6 출력
digitalWrite(ledA, HIGH);
digitalWrite(ledB, LOW);
digitalWrite(ledC, HIGH);
digitalWrite(ledD, HIGH);
digitalWrite(ledE, HIGH);
digitalWrite(ledF, HIGH);
```



```
digitalWrite(ledG, HIGH);
digitalWrite(DOT, LOW);
break;

case 7:                                //7 출력
digitalWrite(ledA, HIGH);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(ledD, LOW);
digitalWrite(ledE, LOW);
digitalWrite(ledF, LOW);
digitalWrite(ledG, LOW);
digitalWrite(DOT, LOW);
break;

case 8:                                //8 출력
digitalWrite(ledA, HIGH);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(ledD, HIGH);
digitalWrite(ledE, HIGH);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(DOT, LOW);
break;

case 9:                                //9 출력
digitalWrite(ledA, HIGH);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(ledD, HIGH);
digitalWrite(ledE, LOW);
digitalWrite(ledF, HIGH);
```

```

    digitalWrite(ledG, HIGH);
    digitalWrite(DOT, LOW);
    break;

    default:
    digitalWrite(ledA, LOW);
    digitalWrite(ledB, LOW);
    digitalWrite(ledC, LOW);
    digitalWrite(ledD, LOW);
    digitalWrite(ledE, LOW);
    digitalWrite(ledF, LOW);
    digitalWrite(ledG, LOW);
    digitalWrite(DOT, LOW);
    break;
}
}

void clr(){                                     //Clear 서브함수
    digitalWrite(ledA, LOW);
    digitalWrite(ledB, LOW);
    digitalWrite(ledC, LOW);
    digitalWrite(ledD, LOW);
    digitalWrite(ledE, LOW);
    digitalWrite(ledF, LOW);
    digitalWrite(ledG, LOW);
    digitalWrite(DOT, LOW);
    delay(500);
}

void set(){                                     //setting 서브함수
    digitalWrite(ledA, HIGH);
    digitalWrite(ledB, HIGH);
    digitalWrite(ledC, HIGH);

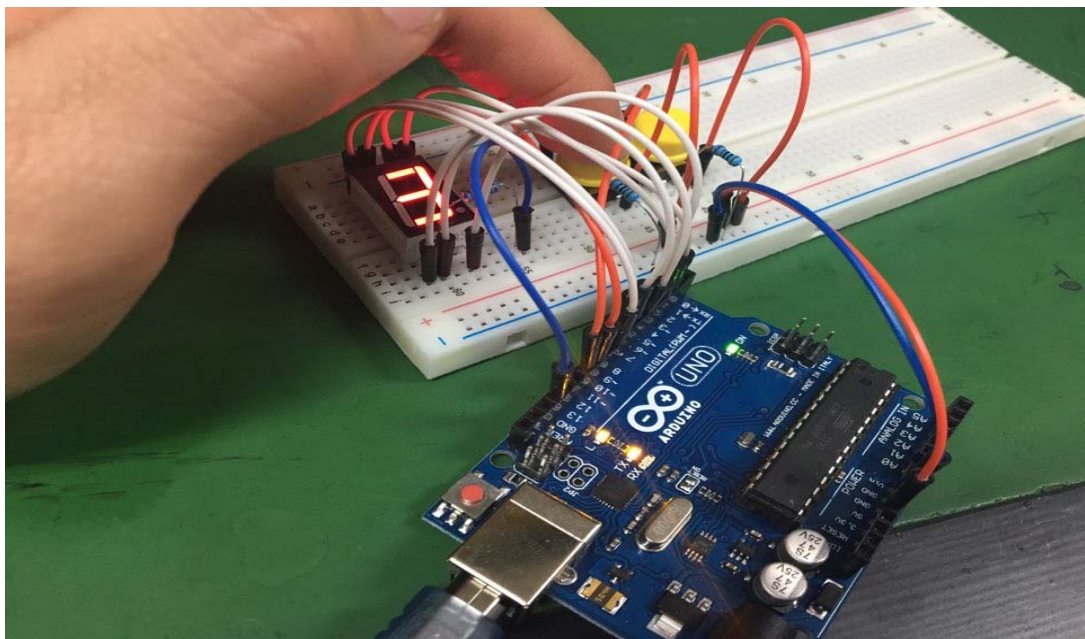
```

```

digitalWrite(ledD, LOW);
digitalWrite(ledE, LOW);
digitalWrite(ledF, LOW);
digitalWrite(ledG, LOW);
digitalWrite(DOT, LOW);
delay(500);
}

void set(){                                     //setting 서브함수
    digitalWrite(ledA, HIGH);
    digitalWrite(ledB, HIGH);
    digitalWrite(ledC, HIGH);
    digitalWrite(ledD, HIGH);
    digitalWrite(ledE, HIGH);
    digitalWrite(ledF, HIGH);
    digitalWrite(ledG, HIGH);
    digitalWrite(DOT, HIGH);
    delay(2000);
}

```



[프로그램 실행 결과]

Challenge 4-1)

위의 예제 3)에서 7-segment 의 숫자를 1, 3, 5, 7, 9 홀수로 숫자를 7-segment 에 차례대로 반복하여 표시하는 프로그램 작성하시오.

Challenge 4-2)

위의 예제 3)에서 7-segment 의 숫자를 data base 를 건드리지 말고 8, 6, 4, 2, 0 짝수 숫자를 역으로 7-segment 에 차례대로 반복하여 표시하는 프로그램 작성하시오.

Challenge 4-3)

위의 예제 4)에서 기존의 회로도를 이용하여 아래 조건에 맞도록 프로그램을 작성한다.

- 스위치1을 누르면 7-SEGMENT에서 숫자 1, 3, 5, 7, 9 홀수 순서대로 반복하여 출력한다.

GoldenBell 4-1)

위의 예제 4)에서 기존의 회로도를 이용하여 아래 조건에 맞도록 프로그램을 작성한다.

- 1) 스위치1을 누르면 7-SEGMENT에서 숫자 9, 7, 5, 3, 1 를 순서대로 반복하여 출력한다.
- 2) 스위치2를 누르면 7-SEGMENT에서 숫자 0, 2, 4, 6, 8을

순서대로 반복하여 출력한다.

GoldenBell 4-2)

위의 예제 4)에서 기존의 회로도를 이용하여 아래 조건에 맞도록 프로그램을 작성한다.

스위치1을 누르면 7-SEGMENT에서 숫자 0, 1, 2, 3, 4, 5, 6, 7, 8, 9를 순방향 순서대로 반복하여 출력하던 중 스위치2를 누르면 현재 출력 값에서 역방향 순서로 숫자를 7-SEGMENT에 반복하여 출력하는 프로그램을 작성한다.

역방향 순서로 반복하여 출력하던 중 스위치 1을 누르면 현재의 출력하던 값에서 다시 순방향으로 순서로 반복하여 출력하는 프로그램을 작성한다.

어떠한 경우 든 출력 중 스위치 1과 2를 동시에 누르면 0, 2, 4, 6, 8을 계속 출력 한다.