

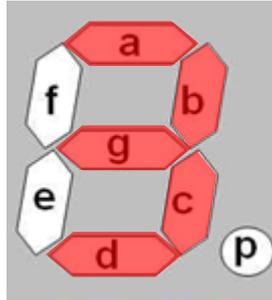
7-segment 구동

Department of Embedded IT
Busan University of Foreign Studies

Contents

1. 7-Segment
2. 7-Segment Array
3. 한 개의 7-Segment 실습
4. 7-Segment Array 실습
5. 0000 ~ 9999 숫자 디스플레이 실습

1. 7-Segment

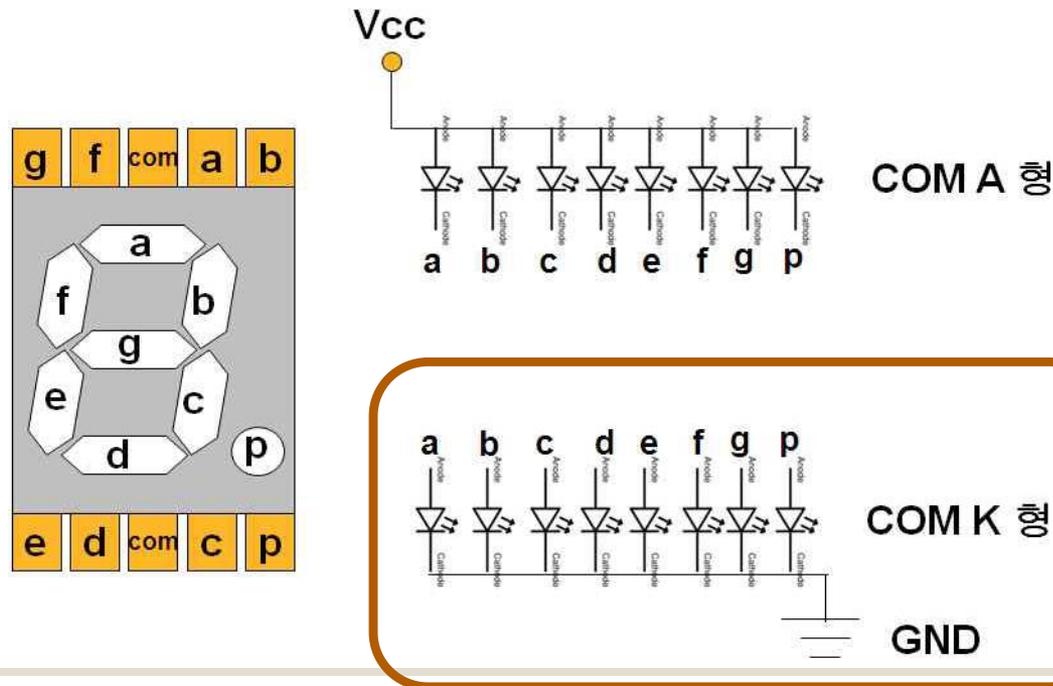


p	g	f	e	d	c	b	a
0	1	0	0	1	1	1	1
0x4F							

- 7-Segment는 일정 개수 이상의 LED를 사용하여 숫자 및 영문자를 표시할 수 있는 표시장치
- 보통 7-Segment LED 라고 하기도 하는데, 이는 FND의 한 종류임

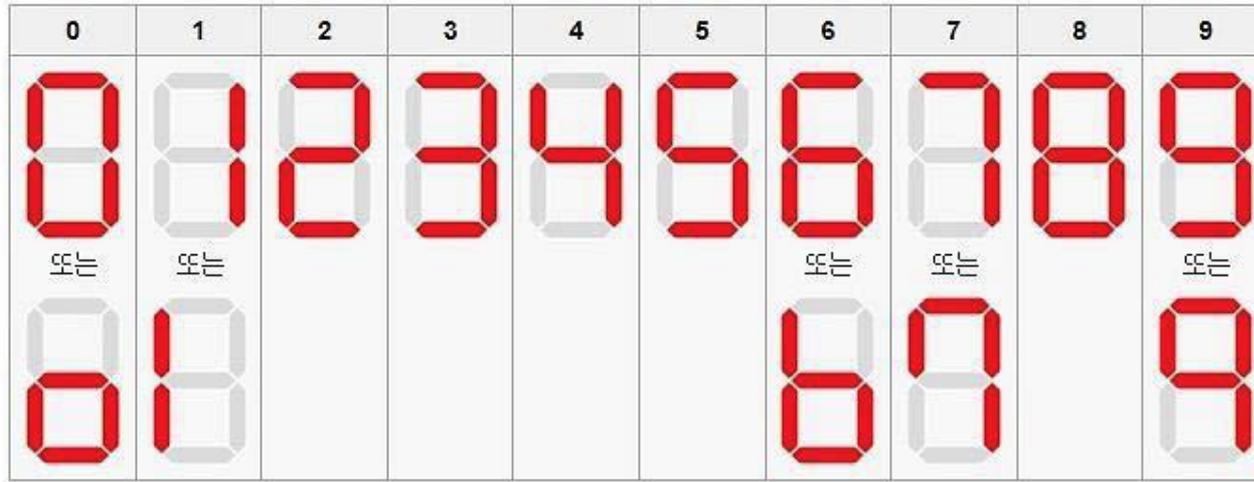
1. 7-Segment

- 7-Segment의 종류는 공통(Common) 단자에 인가되는 전원에 따라서 **Common Anode(+공통)**과 **Common Cathode(-공통)**으로 분류
- Anode(+공통)형 : 공통 단자에 VCC(+5V)를 연결하고 입력 단자에 0V를 인가하였을 때 해당하는 LED에 램프가 들어오는 것
- Cathode(-공통)형 : 공통단자에 접지(0V)를 연결하고 입력단자에 +5V를 인가하였을 때, 해당하는 LED에 램프가 들어오는 것



1. 7-Segment

❖ 각 숫자에 해당하는 7세그먼트 표시 장치



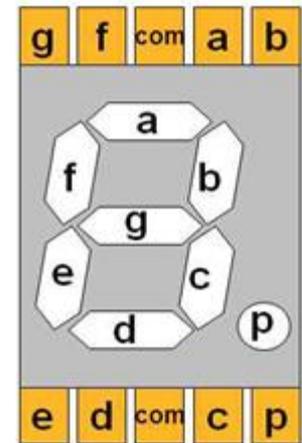
❖ array 형태의 7-Segment

- 숫자를 표시하는 방법은 단일 7-Segment와 동일하나 array형태의 7-Segment는 숫자를 표시할 7-Segment의 위치 지정이 필요
- 원하는 숫자를 표시하기 위해서는 우선 **7-Segment의 위치를 지정하고 원하는 숫자의 data를 각 핀에 입력**

1. 7-Segment

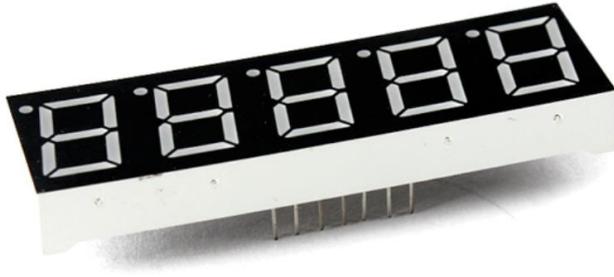
- 7-Segment LED에 16진수를 표현하기 위한 데이터 값

16진수	7-Segment의 비트값								데이터 값 (HEX)
	H	G	F	E	D	C	B	A	
0	0	0	1	1	1	1	1	1	0X3F
1	0	0	0	0	0	1	1	0	0X06
2	0	1	0	1	1	0	1	1	0X5B
3	0	1	0	0	1	1	1	1	0X4F
4	0	1	1	0	0	1	1	0	0X66
5	0	1	1	0	1	1	0	1	0X6D
6	0	1	1	1	1	1	0	1	0X7D
7	0	0	1	0	0	1	1	1	0X27
8	0	1	1	1	1	1	1	1	0X7F
9	0	1	1	0	0	1	1	1	0X67
A	0	1	1	1	0	1	1	1	0X77
B	0	1	1	1	1	1	0	0	0X7C
C	0	0	1	1	1	0	0	1	0X39
D	0	1	0	1	1	1	1	0	0X5E
E	0	1	1	1	1	0	0	1	0X79
F	0	1	1	1	0	0	0	1	0X71

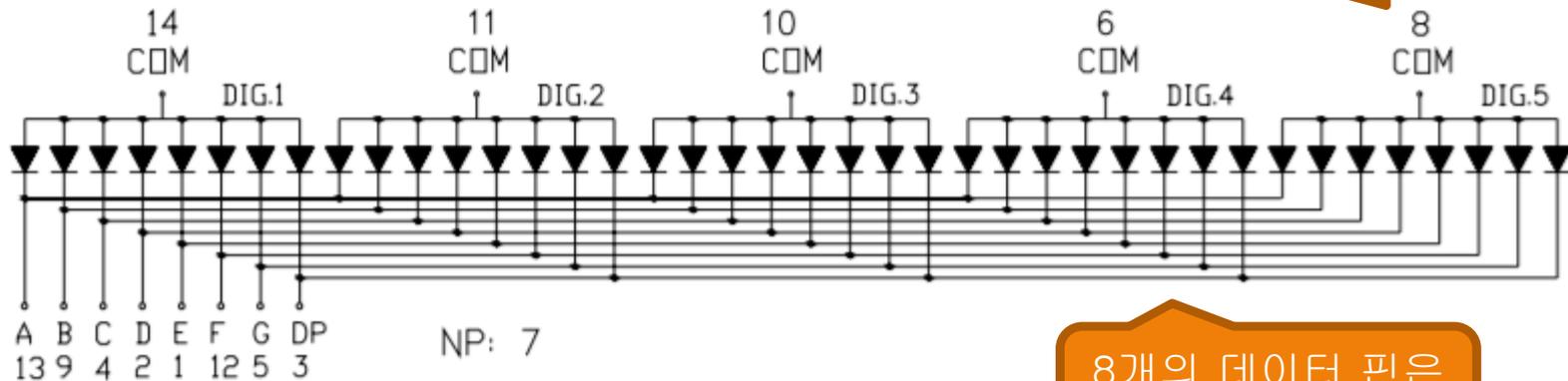


2. 7-Segment Array

- ❖ Common-Cathode 타입의 array 7-Segment 모듈



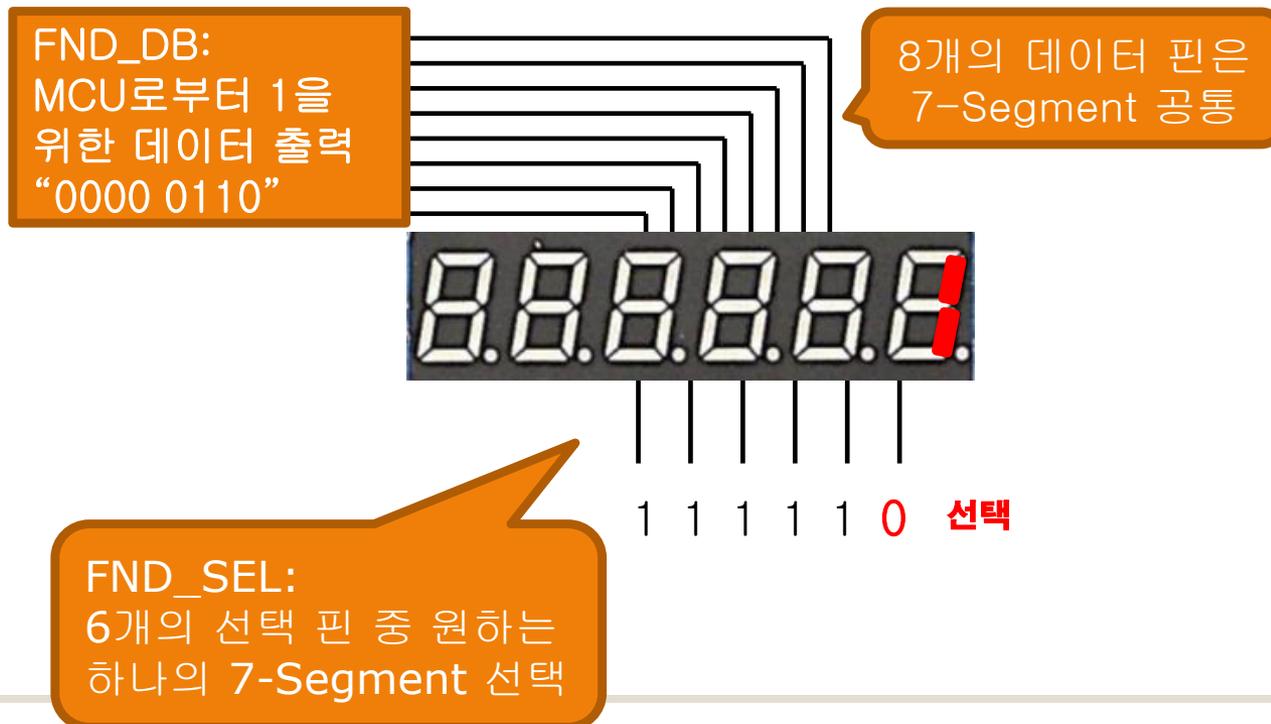
5개의 선택 핀 중 원하는 하나의 7-Segment 선택



8개의 데이터 핀은 7-Segment 공통

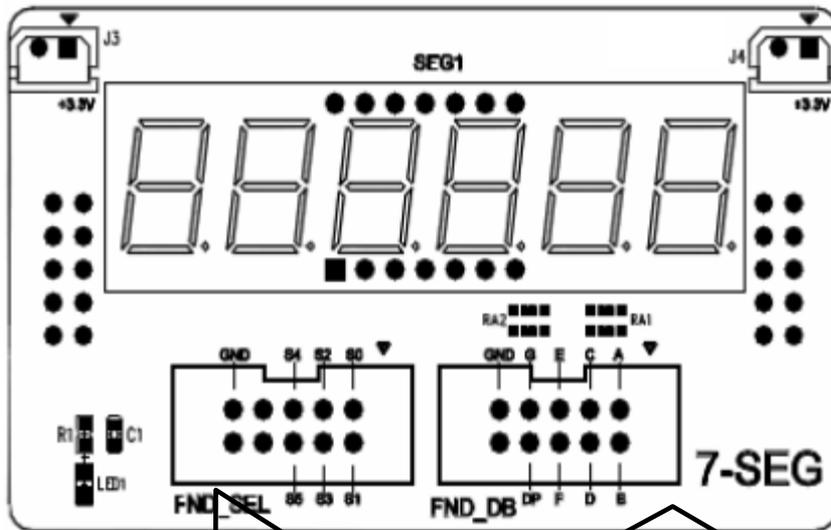
2. 7-Segment Array

- ❖ 7-Segment 모듈의 제어 방법은 Common Cathode 타입의 Array 형태의 모듈이므로 우선 **FND_SEL 포트에 하나의 FND를 선택하여 LOW를 인가**
- ❖ 숫자나 문자를 표시하기 위한 **FND_DB 포트에 HIGH/LOW를 인가하면** 선택된 FND에 원하는 숫자나 문자 표시



2. 7-Segment Array

- ❖ Common-Cathode 타입의 array 7-Segment 모듈



FND_SEL : 6개의
7-Segment 중 한
개 선택

FND_DB : 디스플레이 문자 정보 전송
선택된 **FND**에 문자 디스플레이

3. 한 개의 7-Segment 실습

- ❖ 한 개의 7-Segment(FND) 모듈에 데이터를 나타내기 위한 핀 연결

Device	Mega ADK	FND Module
PIN NO	22	FND_SEL S0
PIN NO	30	FND_DB A
PIN NO	31	FND_DB B
PIN NO	32	FND_DB C
PIN NO	33	FND_DB D
PIN NO	34	FND_DB E
PIN NO	35	FND_DB F
PIN NO	36	FND_DB G
PIN NO	37	FND_DB DP

3. 한 개의 7-Segment 실습

```
int segpin[8] = {30,31,32,33,34,35,36,37}; //데이터 핀을 배열로 정의
int FND_SEL=22; // S0 선택신호
int data[8] = {0,1,1,0,0,0,0,0}; // ' 1 ' 을 표시하기 위한 비트 데이터
```

```
void setup(){
    for(int i = 0; i<8; i++) {
        pinMode(segpin[i], OUTPUT); }
    pinMode(FND_SEL, OUTPUT);
    digitalWrite(FND_SEL,LOW);
}
```

```
void loop(){
    for(int i = 0; i<8; i++) {
        digitalWrite(segpin[i], data[i]);
    }
}
```

```
          A B C D E F G H
int segpin[8] = {30,31,32,33,34,35,36,37};
int data[8] = { 0, 1, 1, 0, 0, 0, 0, 0};
```

3. 한 개의 7-Segment 실습

- ❖ 실행결과 7-Segment에는 1이라는 숫자 표시
 - 7-Segment와 연결된 아두이노 보드의 데이터 핀을 segpin 배열에 저장
 - 7-Segment 표시할 숫자 ' 1 ' 을 위한 데이터를 data 배열에 저장
 - setup() 함수에서 7-Segment 연결된 보드의 핀들을 출력으로 설정
 - loop() 함수에서는 숫자 '1'을 표시하기 위한 data를 7-Segment와 연결된 데이터 핀에 출력

4. 7-Segment Array 실습

- ❖ 7-Segment 모듈의 제어 방법
 - 한 개의 FND를 선택하여 FND_SEL 포트에 LOW를 인가
 - 숫자나 문자를 표시하기 위한 FND_DB 포트에 HIGH/LOW 인가
- ❖ array 7-Segment 모듈에 0~5까지 숫자를 표시하고 1초에 한번씩 옆으로 숫자가 이동하는 스케치 작성

4. 7-Segment Array 실습

- ❖ 7-Segment 모듈에 0~5을 나타내는 프로그램을 위한 핀 연결

Device	Mega ADK	FND Module
PIN NO	22	FND_SEL S0
PIN NO	23	FND_SEL S1
PIN NO	24	FND_SEL S2
PIN NO	25	FND_SEL S3
PIN NO	26	FND_SEL S4
PIN NO	27	FND_SEL S5
PIN NO	30	FND_DB A
PIN NO	31	FND_DB B
PIN NO	32	FND_DB C
PIN NO	33	FND_DB D
PIN NO	34	FND_DB E
PIN NO	35	FND_DB F
PIN NO	36	FND_DB G
PIN NO	37	FND_DB DP

4. 7-Segment Array 실습

```
int FND_SEL[6] = {22,23,24,25,26,27}; //FND_SEL 포트와 연결된 핀 배열로 정의
int FND_DB[8] = {30,31,32,33,34,35,36,37}; //FND_DB 포트와 연결된 핀 배열로 정의
int FND_DATA[] = {0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x27,0x7F,0x67,0x77,0x7C,0x39,
                 0x5E,0x79,0x71}; //FND에 표시할 숫자 및 문자 0~9,A~F 의 data 값
```

```
void setup(){
    for(int i = 0; i<8; i++){
        pinMode(FND_DB[i],OUTPUT);    }
    for(int i = 0; i<6; i++){
        pinMode(FND_SEL[i],OUTPUT);
        digitalWrite(FND_SEL[i],HIGH);    }
}
```

4. 7-Segment Array 실습

```
void FndDisplay(byte Position, byte Data){  
    for(int i = 0; i<6; i++){  
        if( i == Position){  
            digitalWrite(FND_SEL[i],LOW); }  
        else{  
            digitalWrite(FND_SEL[i],HIGH); }  
        for(int j = 0; j<8; j++){  
            digitalWrite(FND_DB[j],bitRead(Data,j));  
        }  
    }  
}
```

1의 경우

FND_DATA[1]=0x06

0x06=0000 0110

bitRead(0000 0110,j)

j=0; FND_DB[0]=0

j=1; FND_DB[1]=1

j=2; FND_DB[2]=1

j=3; FND_DB[3]=0

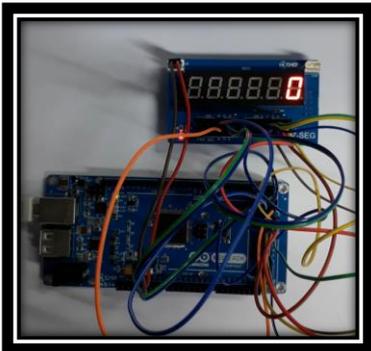
.

.

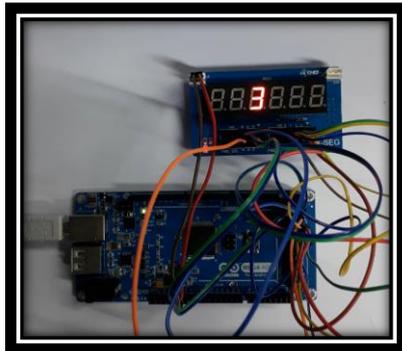
4. 7-Segment Array 실습

```
void loop(){  
    for(int i = 0; i<6; i++){  
        FndDisplay(i,FND_DATA[i]);  
        delay(1000);  
    }  
}
```

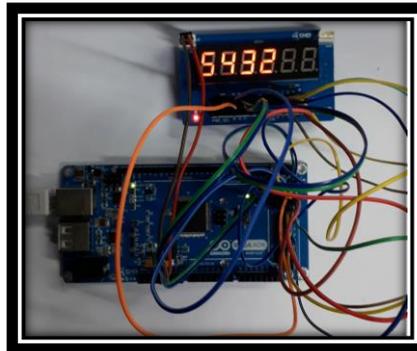
*** **delay(1000)**을 100, 10, 1로
변화시키면서 결과 확인



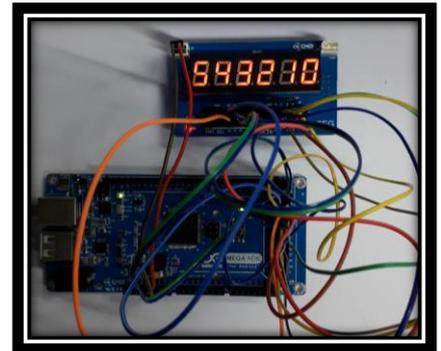
delay(1000)



delay(100)



delay(10)



delay(1)

4. 7-Segment Array 실습

- FND_SEL 포트와 FND_DB 포트에 연결된 핀을 각각 배열로 정의, 그리고 FND에 표시할 숫자 및 문자 0~9, A~F까지의 Data값을 배열로 정의
- **setup()함수에서 FND_SEL과 FND_DB 핀들을 출력으로 설정**
- **FND_SEL에 연결된 핀들을 HIGH로 출력 (모든 7-Segment는 미선택 상태)**

- loop() 함수에서는 FndDisplay() 함수로 표시할 위치 및 Data를 넘겨줌
- FndDisplay() 함수에서는 표시할 위치를 확인 후 해당 FND_SEL 핀에 LOW 출력
- 문자 및 숫자의 Data를 bitRead() 함수를 통해 bit화 시켜 FND_DB의 각 핀에 digitalWrite() 함수를 통해 출력

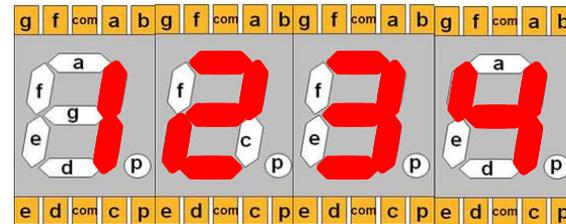
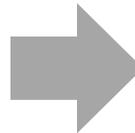
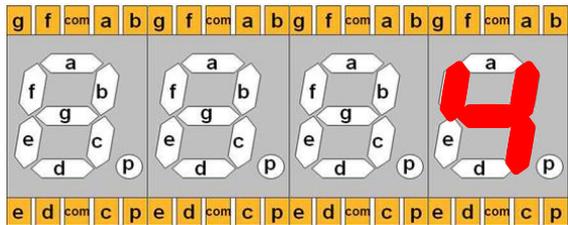
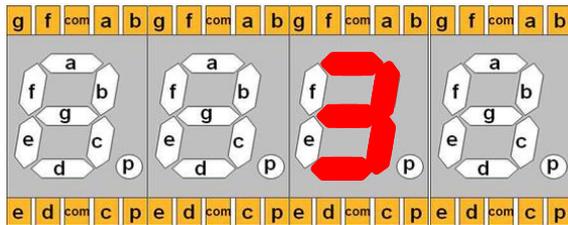
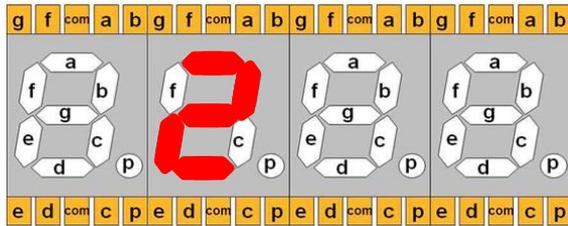
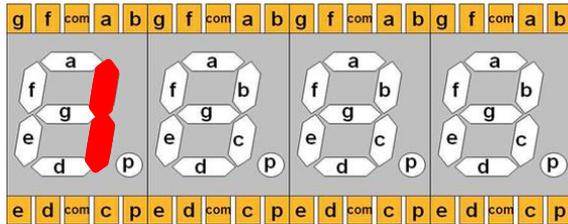
4. 7-Segment Array 실습

❖ bitRead()

- 매개 변수로 받은 값을 bit 단위로 변환하여 0 인지 1인지를 반환 하는 함수
- 매개 변수는 2가지로 **bit단위로 변환할 변수**와 **읽을 비트의 번호**
- 비트의 번호는 맨 오른쪽 비트가 0번째 비트

4. 7-Segment Array 실습

동적 디스플레이



5. 0000~9999 숫자 디스플레이 실습

- ❖ 4개의 7-Segment만 이용하여 0.5초 간격으로 숫자를 1씩 증가시켜 0000~9999까지 디스플레이 구현

Device	Mega ADK	FND Module
PIN NO	22	FND_SEL S0
PIN NO	23	FND_SEL S1
PIN NO	24	FND_SEL S2
PIN NO	25	FND_SEL S3
PIN NO	26	FND_SEL S4
PIN NO	27	FND_SEL S5
PIN NO	30	FND_DB A
PIN NO	31	ND_DB B
PIN NO	32	FND_DB C
PIN NO	33	FND_DB D
PIN NO	34	FND_DB E
PIN NO	35	FND_DB F
PIN NO	36	FND_DB G
PIN NO	37	FND_DB DP

5. 0000~9999 숫자 디스플레이 실습

```
int FND_SEL[6] = {22,23,24,25,26,27}; //FND_SEL 포트와 연결된 핀 배열로 정의
int FND_DB[8] = {30,31,32,33,34,35,36,37}; //FND_DB 포트와 연결된 핀 배열로 정의
int FND_DATA[] = {0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x27,0x7F,0x67,0x77,0x7C,0x39,
                  0x5E,0x79,0x71};

int temp;

void setup(){
    for(int i = 0; i<8; i++){
        pinMode(FND_DB[i],OUTPUT); }
    for(int i = 0; i<6; i++){
        pinMode(FND_SEL[i],OUTPUT);
        digitalWrite(FND_SEL[i],HIGH); }
    }
}
```

5. 0000~9999 숫자 디스플레이 실습

```
void FndDisplay(byte Position, byte Data){
    for(int i = 0; i<6; i++){
        if( i == Position){
            digitalWrite(FND_SEL[i],LOW);  }
        else{
            digitalWrite(FND_SEL[i],HIGH);  }

        for(int j = 0; j<8; j++){
            digitalWrite(FND_DB[j],bitRead(Data,j));  }
        }
    }
}
```

5. 0000~9999 숫자 디스플레이 실습

```
void loop(){  
    temp++;
```

```
    FndDisplay(3, );  
        delay(5);  
    FndDisplay(2, );  
        delay(5);  
    FndDisplay(1, );  
        delay(5);  
    FndDisplay(0, );  
        delay(5);  
}
```

