

# Chapter 6-2

## 표시장치 제어 이론 및 실습

마이크로프로세서

HRI 연구실

김동한



Human-Robot Interaction  
Laboratory



KYUNG HEE  
UNIVERSITY

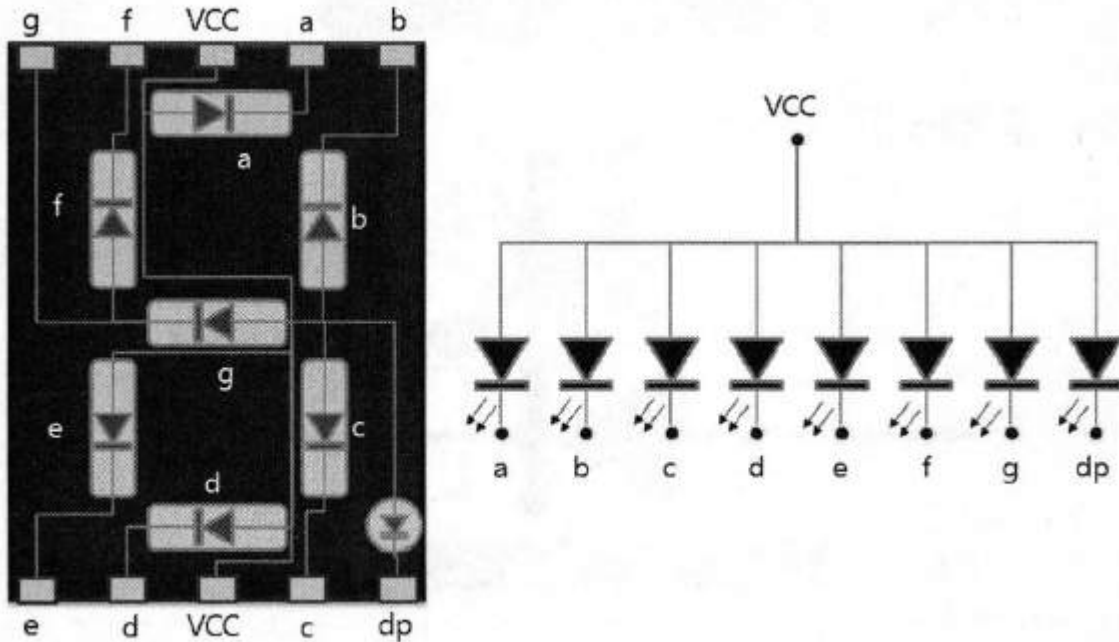
## 6.2 FND(7-Segment)제어

### 7-Segment 표시장치 (Seven-segment display)

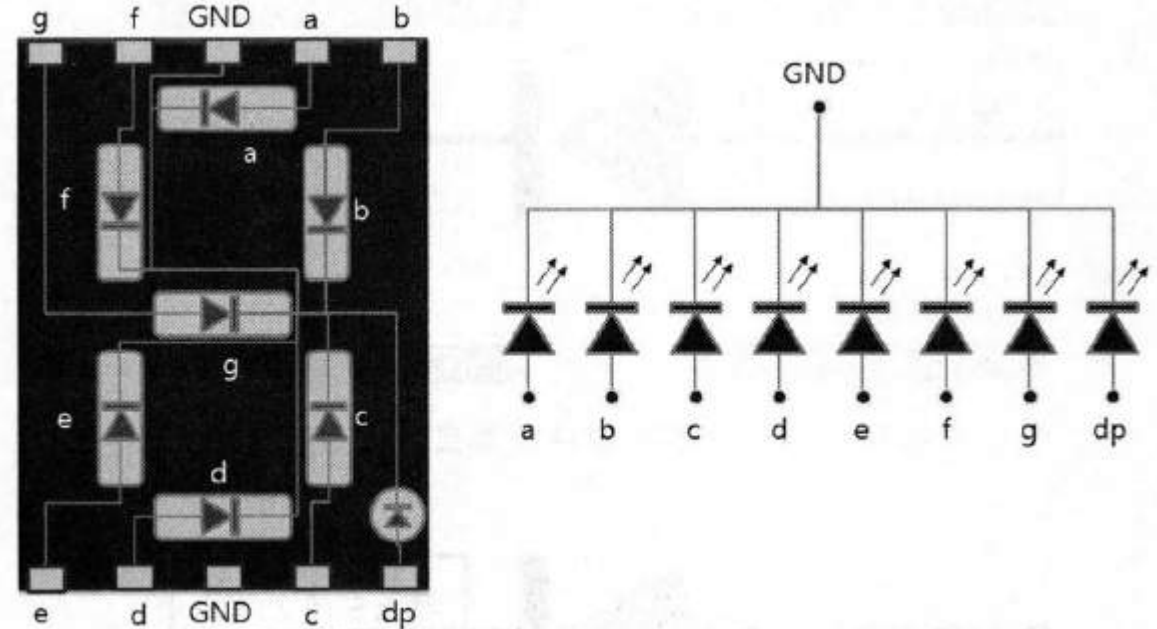
- FND(Flexible Numerical Diode) : 7-segment라고 하고 숫자나 간단한 기호 표현에 많이 이용되며, a에서 g까지 7개의 LED와 소수점 표시를 한 DP(Decimal Point)의 LED 1개로 구성되어 있다.
- 각 LED는 표시되는 문자의 일부분을 구성하므로 각각을 세그먼트(segment :부분, 구분)라고 부르고 조합하여 숫자를 표현할 수 있다.
- 표시장치로서 7개의 LED가 숫자를 표시할 수 있는 형태로 구성되어 있다.
- Common Anode Type : 공통 핀에 VCC(5V)를 인가하고 제어 신호는 0V로 제어한다.

## 6.2 FND(7-Segment)제어

### 7-Segment 표시장치 (Seven-segment display)



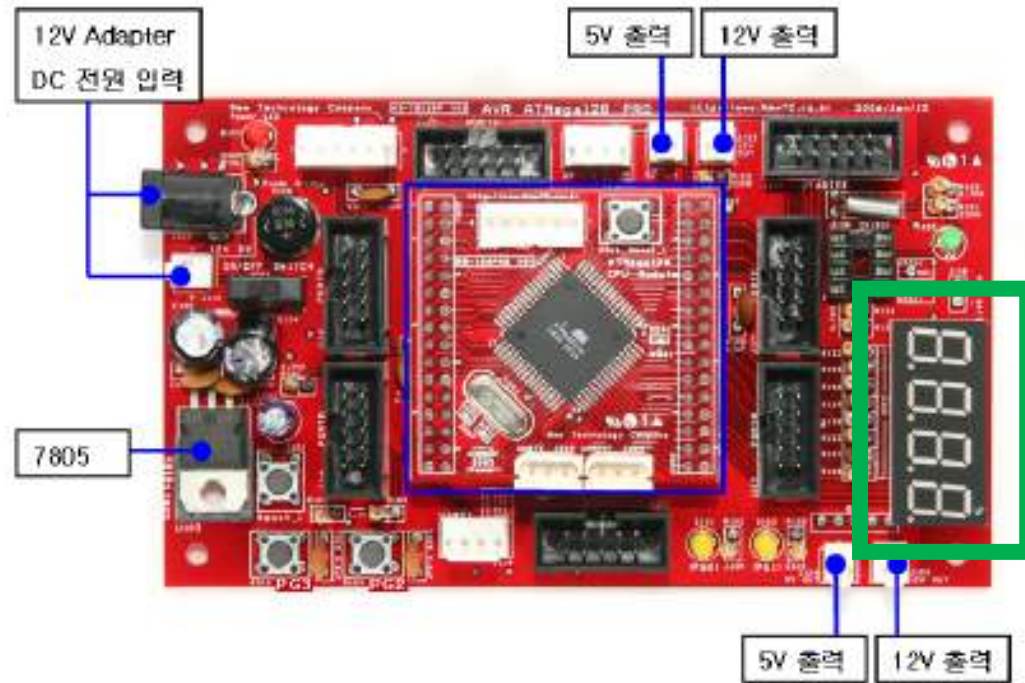
- Common Anode Type : 공통 핀에 Vcc(5V)를 인가하고 제어신호는 0V로 제어한다.



- Common Cathode Type: 공통 핀에 GND(0V)를 인가하고 신호는 High(5V)로 제어.

## 6.2 FND(7-Segment)제어

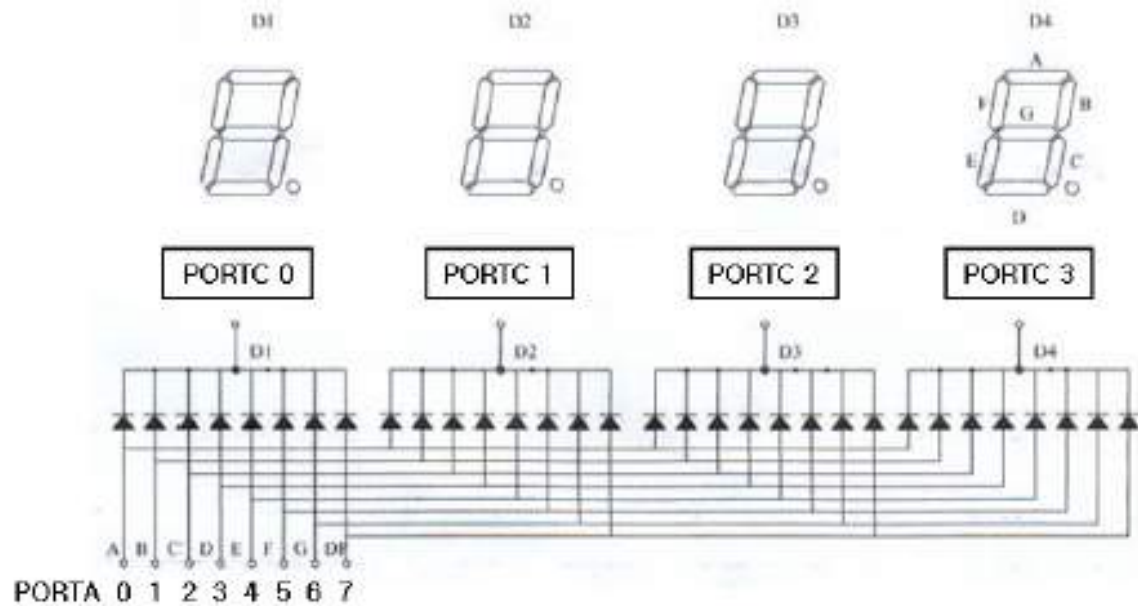
### 7-Segment 사용하기



교재는 외장 7-세그먼트 모듈을 다루고있지만, 이 강의에서는 내장 7-세그먼트 사용법을 간단히 소개한다.

## 6.2 FND(7-Segment)제어

### 7-Segment 사용하기



- 개발보드상의 7-Segment 는 PORTA와 PORTC의 하위 4bit 에 연결되어 있다.
- 7-Segment의 데이터 선을 공유하고 Select 신호가 있기 때문에 시간을 분할하여 한 자리씩 켜야 한다.
- 동시에 켜는 것 불가. 소프트웨어 적으로 Delay를 매우 짧게 주어 동시에 켜지는 것처럼 보이게 만 할 수 있음.)



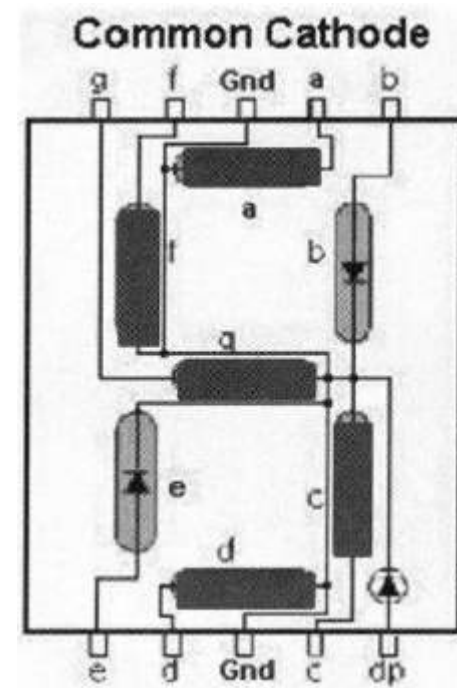
## 6.2 FND(7-Segment)제어

### 7-Segment 사용하기

숫자	PORTA 출력 데이터
0	0x3F
1	0x06
2	0x5B
3	0x4F
4	0x66
5	0x6D
6	0x7D
7	0x27
8	0x7F
9	0x67

Ex) 숫자 5 : 0x6 D : 0b0110 1110

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



- 각 7 세그먼트의 숫자는 7-세그먼트 포트에 다음과 같이 출력이 가능
- 공통(Common) 캐소드 7 -세그먼트 기준으로 작성된 데이터

## 6.2 FND(7-Segment)제어

### 실습. 1 내장 7-Segment 작동법 예제

```
#include <megal28.h>
#include <delay.h>

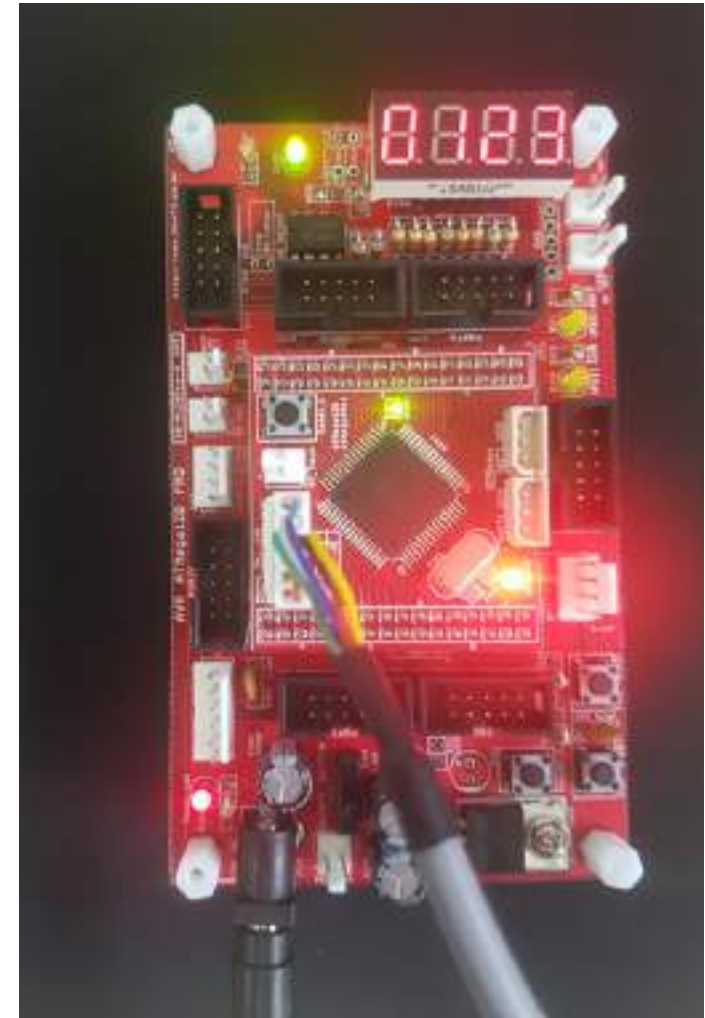
char ss_data[10] = {0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x27, 0x7f, 0x67};

void main(void)
{
    unsigned int i=0, digit=0;
    PORTA = 0x00;
    DDRA = 0xff;
    PORTC = 0x00;
    DDRC = 0xff;

    //insert your functional code here...

    while(1){

        PORTG = i; // LED 출력
        PORTC = 0x0f; // 7-Segment 출력 시작
        PORTA = ss_data[digit]; // 데이터 출력
        PORTC = ~(1<<digit); // Digit 선택
        delay_ms(500); // 500을 작은 값으로 변경하면 숫자가
        // 항상 켜져 있는 것처럼 보입니다.
        i++;
        digit=i%4;
    }
}
```



- Delay\_ms를 0으로 준 결과.
- 항상 켜져있는 것처럼 보인다.

## 6.2 FND(7-Segment)제어

실습. 2 딜레이 함수를 사용하여 1초에 한번 숫자를 +1 시키는 타이머 프로그램

- 내장 7-세그먼트를 사용
- 4가지의 셀렉트 비트중 하나를 골라 한 개의 자리수에만 디스플레이 (PORTC = 0x0e, 0x0d, 0x0b, 0x07 중 택 1)
- 0~9까지의 숫자를 1초에 한번씩 카운트하고 다시 0으로 돌아와 반복

```
#include <megal28.h>
#include <delay.h>
```

```
char ss_data[10] = {0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x27, 0x7f, 0x67};
```

```
void main(void)
```

```
{
```

```
    unsigned int i=0; //, digit=0;
```

```
    PORTA = 0x00;
```

```
    DDRA = 0xff;
```

```
    PORTC = 0x00;
```

```
    DDRC = 0xff;
```

```
    //insert your functional code here...
```

```
while(1){
```

```
    PORTC = 0x0e; // 7-Segment 출력 시작
```

```
    PORTA = ss_data[i]; // 데이터 출력
```

```
    delay_ms(1000); // 500을 작은 값으로 변경하면 숫자가  
    // 항상 켜져 있는 것처럼 보입니다.
```

```
    i++;
```

```
    if (i > 9 ) {
```

```
        i = 0;
```

```
    }
```

```
}
```

```
}
```



## 6.2 FND(7-Segment)제어

[과제5] FND를 활용하여 순차적으로 HELP를 FND에 나타내도록 하는 프로그램을 작성.

- 내장 7-세그먼트를 사용
- `char help_data[4] = { , , , };`