



Microprocessor

4th Week: Static+Dynamic FND

survey

- ◆ The lecture contents are..
- ◆ The speed and pronunciation of speaking is..
- ◆ Homeworks are..
- ◆ Other suggestions

Label from 'very easy' to 'very hard' for three questions and any suggestions you want in this class.

If you have no suggestions and feel easy for most of this class, no need to mail me.

Mail : wowow11111@naver.com

Previous Homework

- ◆ Write a program that turns on sequentially from LED0 to LED7
 - Only one LED should be on at a time
 - If you finish to turn on LED7, start from 0 again (repeat)

practice4.answer1

```
#include <avr/io.h>

void delay(unsigned long x)
{
    while(x--);
}

int main(void)
{
    DDRB = 0xFF;

    while(1)
    {
        PORTB = 0b00000001; // 1
        delay(100000);
        PORTB = 0b00000010; // 2
        delay(100000);
        PORTB = 0b00000100; // 4
        delay(100000);
        PORTB = 0b00001000; // 8
        delay(100000);
        PORTB = 0b00010000; // 16
        delay(100000);
        PORTB = 0b00100000; // 32
        delay(100000);
        PORTB = 0b01000000; // 64
        delay(100000);
        PORTB = 0b10000000; // 128
        delay(100000);
    }
}
```

practice4.answer2

```
#include <avr/io.h>

void delay(unsigned long x)
{
    while(x--);
}

int main(void)
{
    DDRB = 0xFF;
    PORTB = 1;

    while(1)
    {
        delay(100000);

        if(PORTB == 128)
            PORTB = 1;

        else
            PORTB *= 2;
    }
}
```

practice4.answer3

```
#include <avr/io.h>

void delay(unsigned long x)
{
    while(x--);
}

int main(void)
{
    DDRB = 0xFF;
    PORTB = 1;

    while(1)
    {
        delay(100000);

        if(PORTB == 128)
            PORTB = 1;

        else
            PORTB = PORTB << 1;
    }
}
```

Delay function

◆ We have created our own delay function.

→ ***Not exact***

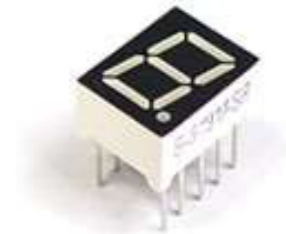
```
1 void delay(unsigned long x)
2 {
3     while(x--);
4 }
```

◆ Built-in function

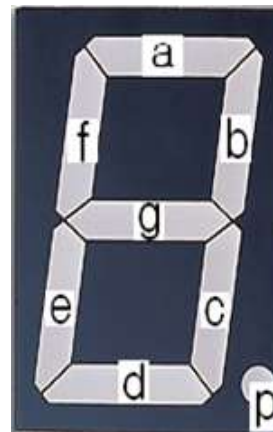
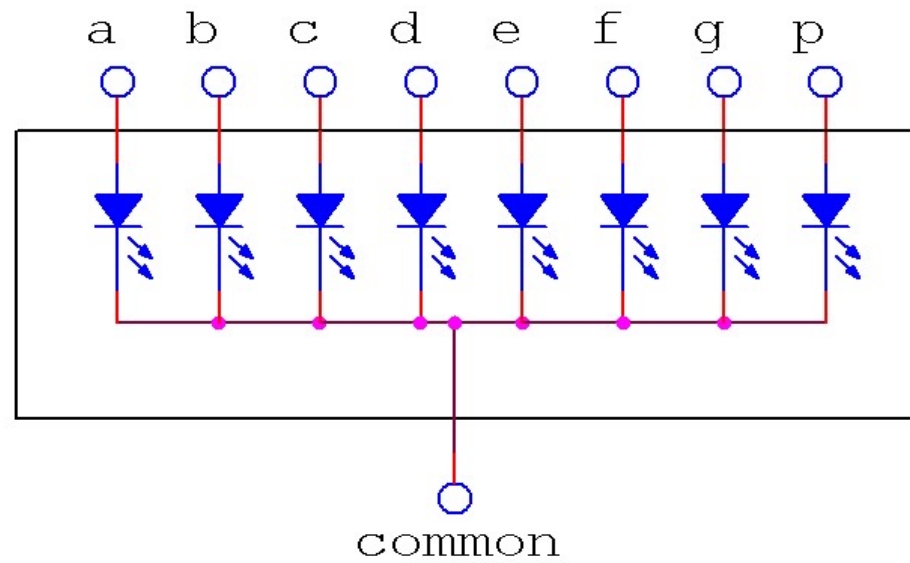
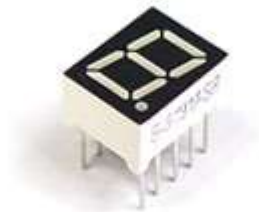
```
1 #define F_CPU 14745600
2 #include <util/delay.h>
3
4 void _delay_us(double __us)
5 void _delay_ms(double __ms)
```

FND(7-segment)

- ◆ Flexible Numeric Display
 - LED set for numeric display
- ◆ Two kinds of FND
 - ① Static
 - ② Dynamic



staticFND



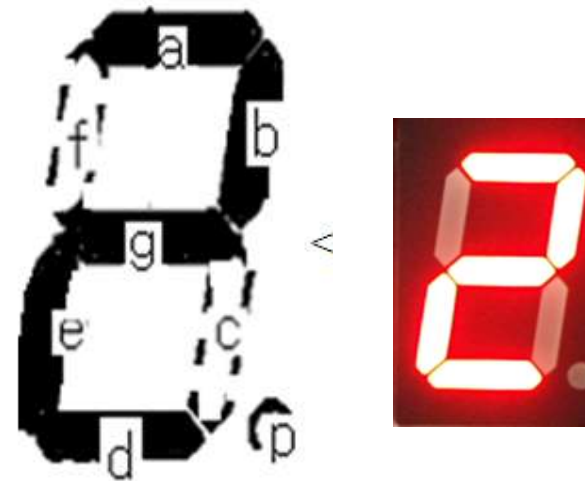
staticFND1

◆ Display 2 on FND

```
#include<avr/io.h>

int main(void)
{
    DDRB=0xFF;
    PORTB=0x5B; // 0b01011011

    return 0;
}
```



FND	p	g	f	e	d	c	b	a
MCU	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Out	0	1	0	1	1	0	1	1

staticFND2

◆ Display decimal 2 digits from 0 to 99

```
#include<avr/io.h>

const char Font[17] = {0x3F,0x06,0x5B,0x4F,0x66,0x6D, //0, 1, 2, 3, 4, 5
                       0x7D,0x07,0x7F,0x6F,0x77,0x7C, //6, 7, 8, 9, A, B
                       0x39,0x5E,0x79,0x71,0x00}; //C, D, E, F, NULL

void delay(unsigned long x)
{
    while(x--);
}

int main(void)
{
    unsigned char digit1, digit0;
    unsigned int count=0;

    DDRB = 0xFF;
    DDRC = 0xFF;

    while(1)
    {
        digit1=count / 10;
        digit0=count % 10;

        PORTB=Font[digit1];
        PORTC=Font[digit0];

        delay(30000);
        count++;

        if(count>99)
            count=0;
    }
}
```

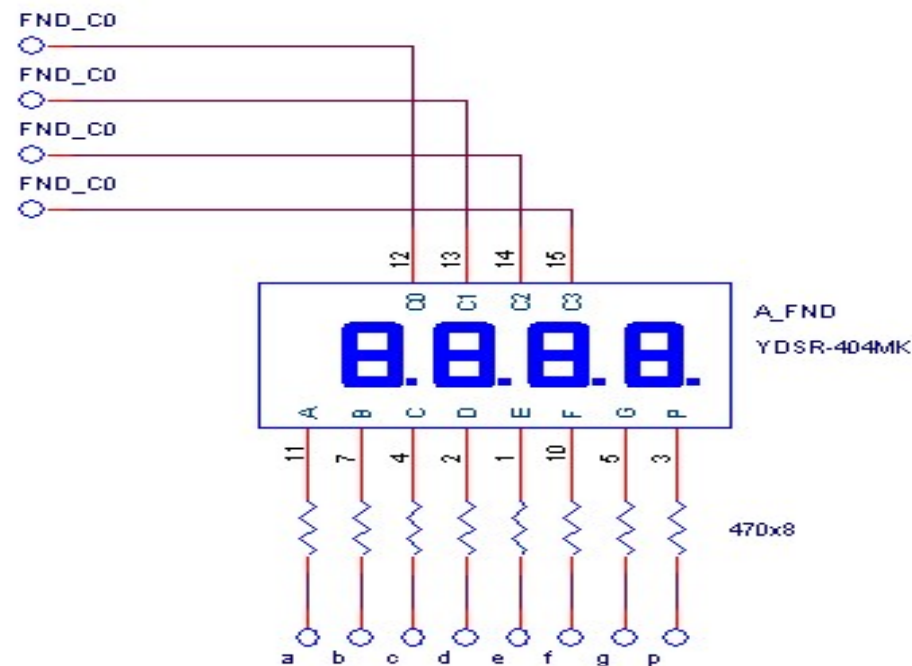


PB[7:0] -> FND1[A:H]

PC[7:0] -> FND2[A:H]

dynamicFND

- ◆ 4 cathode(-) common lines with one 7-segment selectable
- ◆ Connect the [C0:C3], [A:P] terminals to the microprocessor ports
- ◆ [A:P] are used in common, and switch [C0:C3]



dynamicFND1

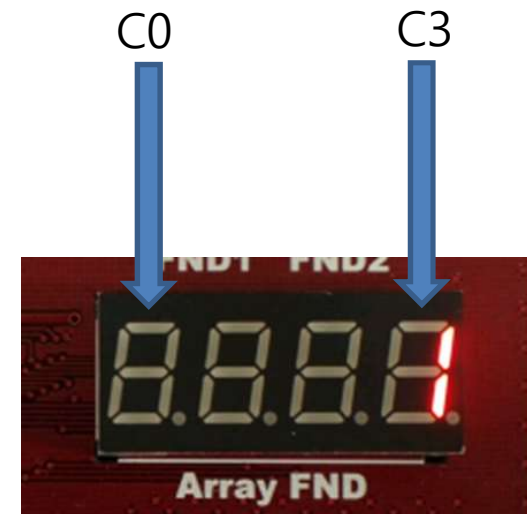
- ◆ Display 1 on the first Array FND (rightside)

```
#include<avr/io.h>

int main(void)
{
    DDRB=0xFF;
    DDRC=0xFF;

    PORTB=0x06; // 0000 0110
    PORTC=0xF7; // 1111 0111

    return 0;
}
```



FND	p	g	f	e	d	c	b	a	C3	C2	C1	C0
MCU	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PG3	PG2	PG1	PG0
Out	0	0	0	0	0	1	1	0	0	1	1	1

dynamicFND2

- ◆ Display the same number in 4 digits

```
#include <avr/io.h>

const char Font[17] = {0x3F,0x06,0x5B,0x4F,0x66,0x6D, //0, 1, 2, 3, 4, 5
                       0x7D,0x07,0x7F,0x6F,0x77,0x7C, //6, 7, 8, 9, A, B
                       0x39,0x5E,0x79,0x71,0x00}; //C, D, E, F, NULL

void delay(unsigned long x)
{
    while(x--);
}

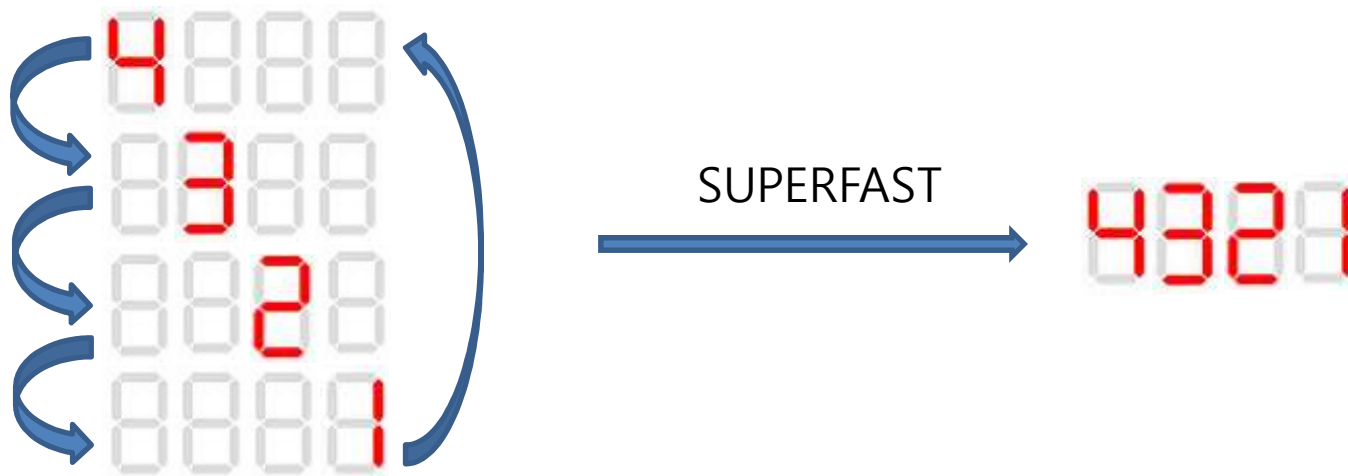
int main(void)
{
    int i;
    DDRB=0xFF;
    DDRC=0xFF;

    PORTB=0x00;
    PORTC=0xF0;

    while(1)
    {
        for(i=0;i<17;i++)
        {
            PORTB=Font[i];
            delay(100000);
        }
    }
}
```

dynamicFND3.1

- ◆ Display the independent number in 4 digits
- ◆ Since there is a common [A:P], there is no way to display each LEDs differently
- ◆ We **deceive** our 'eyes' by displaying the number of each LEDs very quickly and repeatedly.



dynamicFND3.2

```
#include<avr/io.h>

#define FND_CO 0x01
#define FND_C1 0x02
#define FND_C2 0x04
#define FND_C3 0x08

const char Font[17] = {0x3F,0x06,0x5B,0x4F,0x66,0x6D, //0, 1, 2, 3, 4, 5
                      0x7D,0x07,0x7F,0x6F,0x77,0x7C, //6, 7, 8, 9, A, B
                      0x39,0x5E,0x79,0x71,0x00}; //C, D, E, F, NULL

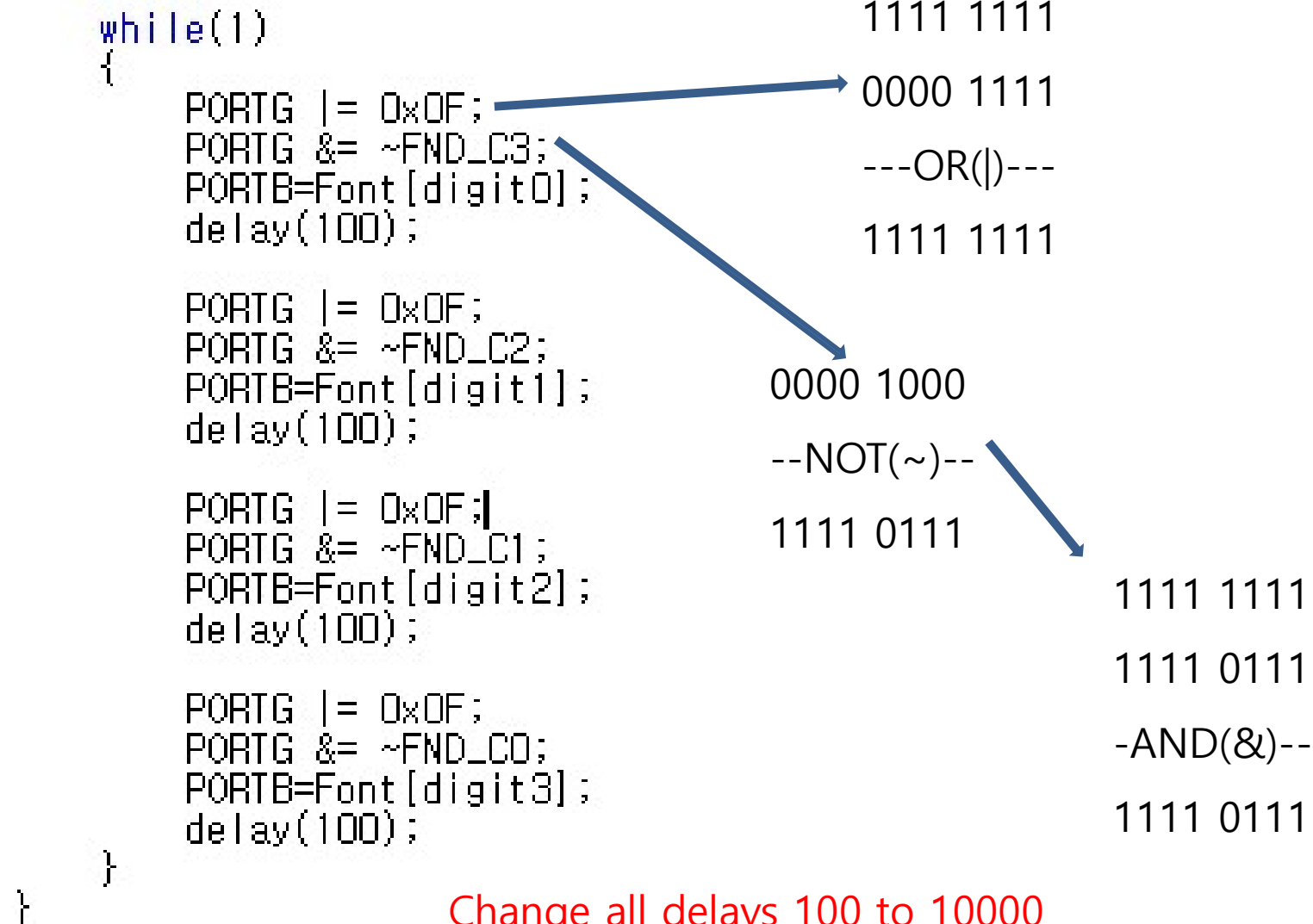
void delay(unsigned long x)
{
    while(x--);
}

int main(void)
{
    unsigned char digit0, digit1, digit2, digit3;
    DDRB=0xFF;
    DDRC=0xFF;

    PORTB=0x00;
    PORTC=0xFF;

    digit0 = 1;
    digit1 = 2;
    digit2 = 3;
    digit3 = 4;
```


dynamicFND3.3



Static & Dynamic FND

◆ Static

- High power consumption
- High data line usage

◆ Dynamic

- Lower power consumption than static FND
- Low data line usage
- Have eyestrain
- Difficult programming

Homework

- ◆ Write a program that counts sequentially from 0 to 999,
- ◆ 1 up every second, delay(10000) is 1 second - (assumption)
- ◆ 4-arrays of Dynamic FND is used (same as previous slide)
- ◆ No need to start again when reached 999. Simply end the program.