2023년 IoT기반 스마트 솔루션 개발자 양성과정

Embedded Application

17-Dot Matrix LED

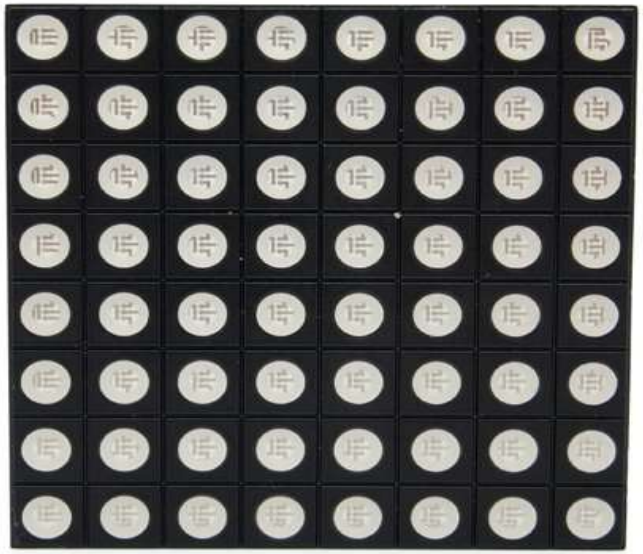
담당 교수 : 윤 종 이
010-9577-1696
ojo1696@naver.com
https://cafe.naver.com/yoons2023

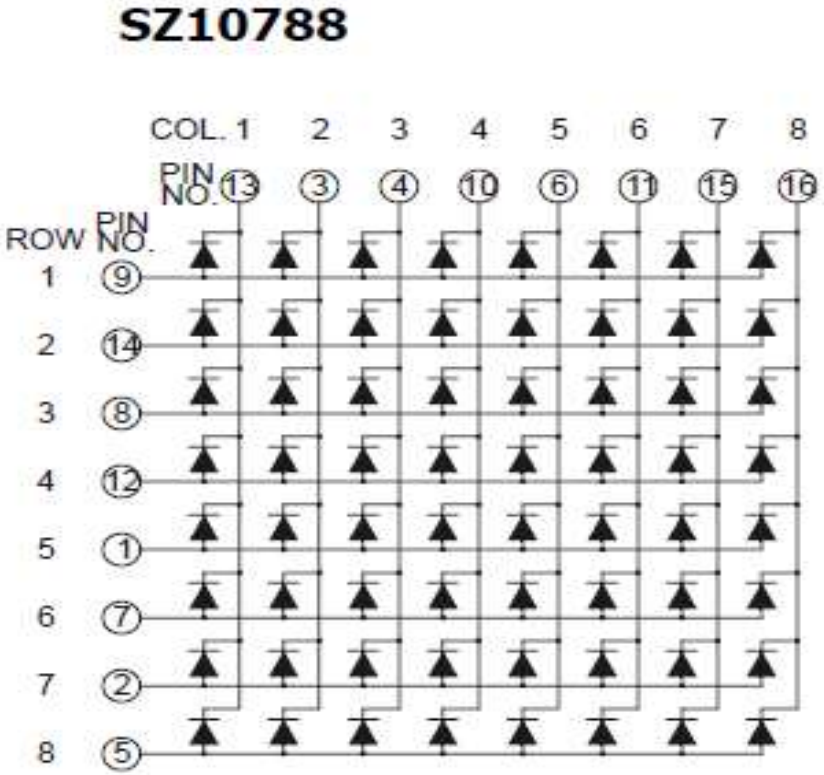
Matrix LED

- LED 매트릭스(행렬)의 형태로 나열하여 글씨나 간단한 도형을 출력할 수 있는 장치
- 행렬의 수와 색상에 따라 다양한 형태로 제작
- 5 x 7, 8 x 8, 16 x 16, 단색 또는 다색

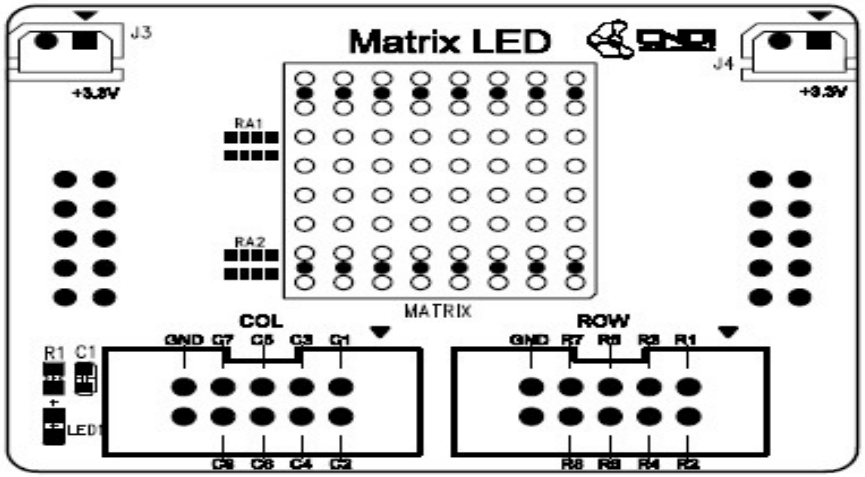
Matrix LED 제어

- ROW 선택
- COL Data 출력(부논리)

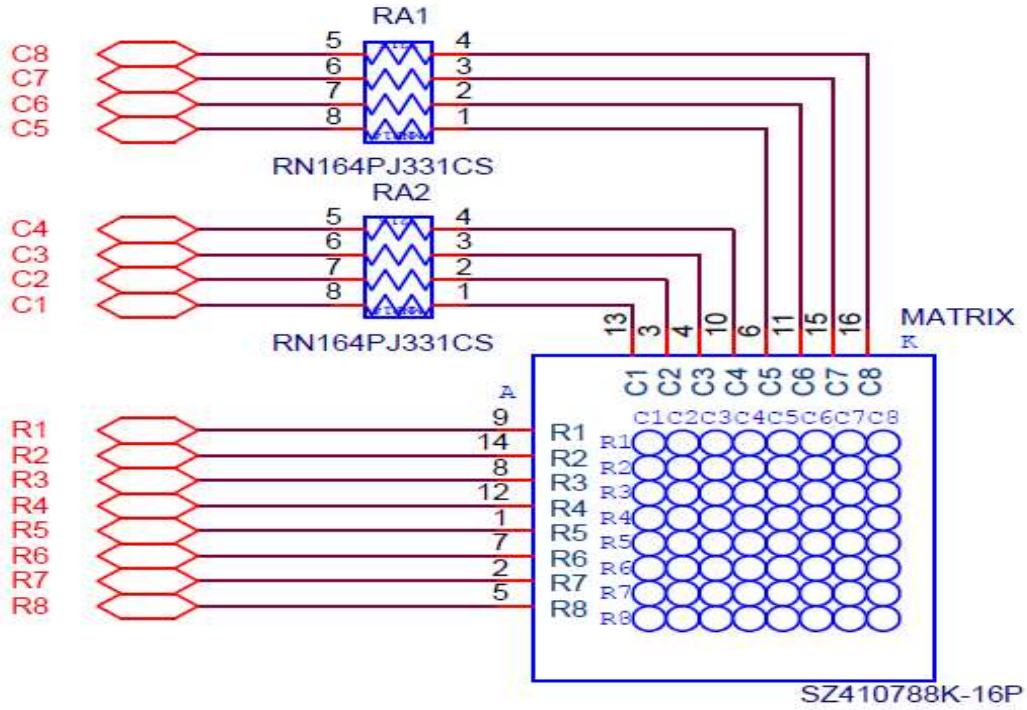


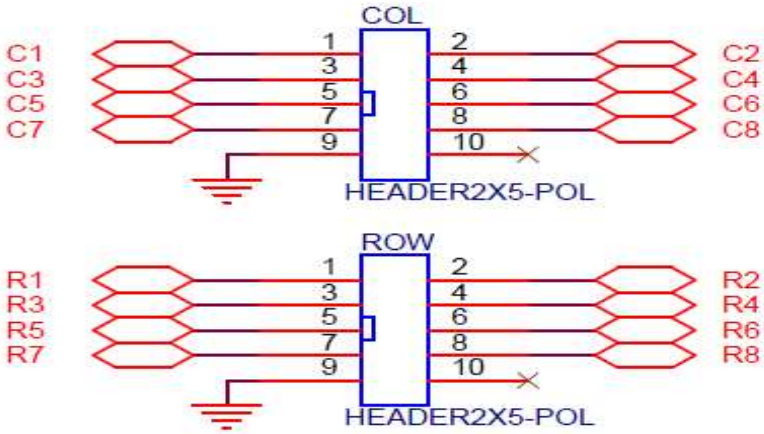


Matrix LED Module

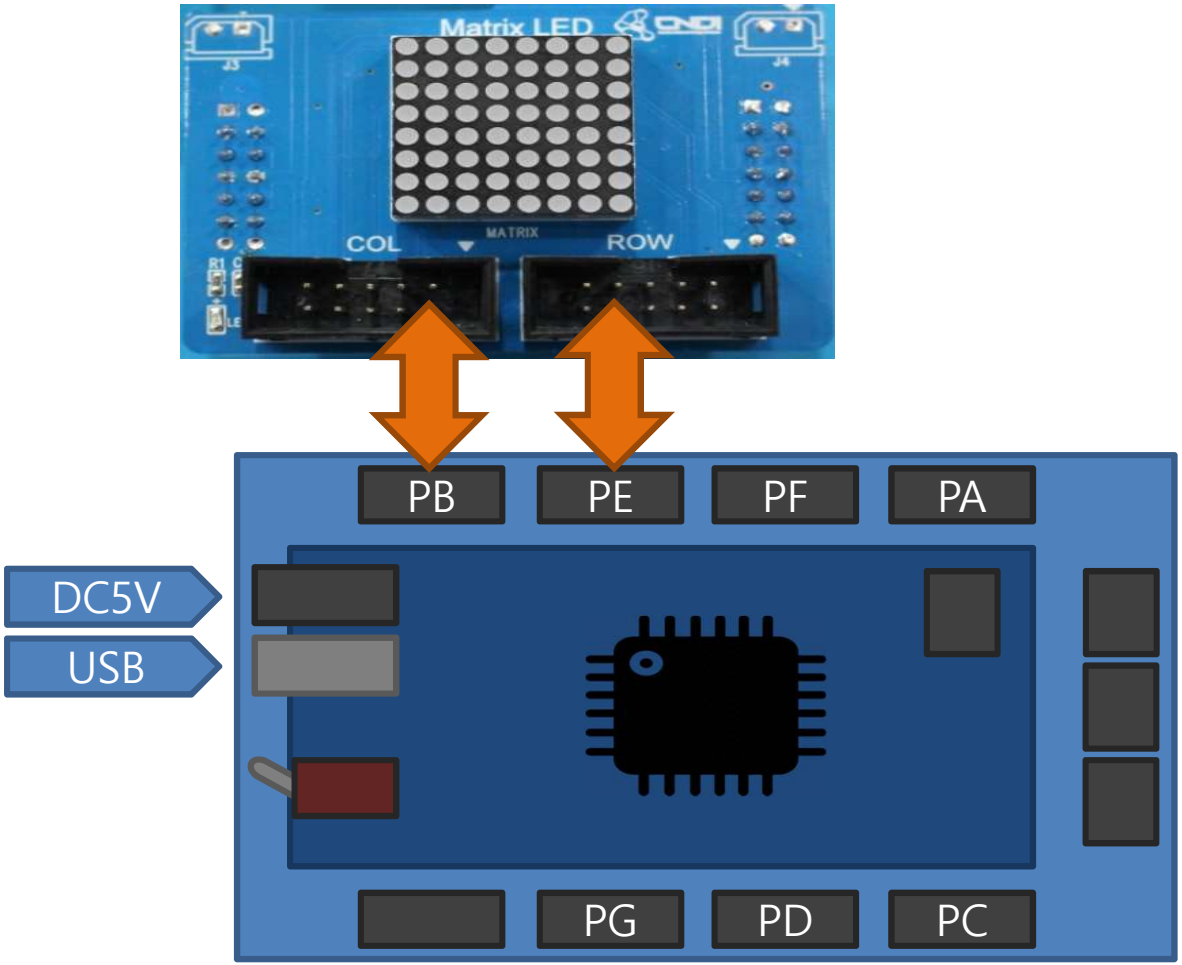


Matrix LED Circuit





Wiring



Ex-1 : Smile Display

	0	1	2	3	4	5	6	7	
01									3C
02									42
04									A5
08									81
10									A5
20									99
40									42
80									3C

Ex-1 : Define

```
#define F_CPU 14745600UL
#include <avr/io.h>
#include <util/delay.h>

#define DOT_COL PORTB
#define DOT_COL_DDR DDRB
#define DOT_ROW PORTE
#define DOT_ROW_DDR DDRE
#define dTime 3

unsigned char DOT_COL_TBL[8] ={0x3C,0x42,0xA5,0x81,0xA5,0x99,0x42,0x3C};
unsigned char DOT_ROW_TBL[8] ={0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
```

Ex-1 : main

```
void CPU_Setup( ) {
    DOT_COL_DDR=0xff;
    DOT_ROW_DDR=0xff;
}

int main(void) {
    CPU_Setup( );

    while (1) {
        for (int k=0; k<8;k++){
            DOT_ROW=DOT_ROW_TBL[k];
            DOT_COL=~DOT_COL_TBL[k];
            _delay_ms(dTime);
        }
    }
}
```

Ex-2 : 화살표 표시

	0	1	2	3	4	5	6	7	
01									18
02									3C
04									7E
08									FF
10									18
20									18
40									18
80									18

Ex-2 : Define

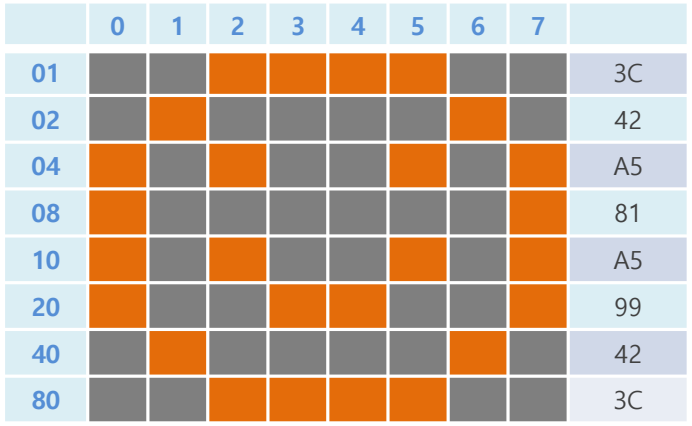
```
#define F_CPU 14745600UL
#include <avr/io.h>
#include <util/delay.h>

#define DOT_COL PORTB
#define DOT_COL_DDR DDRB
#define DOT_ROW PORTE
#define DOT_ROW_DDR DDRE
#define dTime 3
#define Speed 15

unsigned char DOT_COL_TBL[8] ={0x18,0x3C,0x7E,0xFF,0x18,0x18,0x18,0x00};
unsigned char DOT_ROW_TBL[8] ={0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
unsigned char ROW=0;

void CPU_Setup( ) {
    DOT_COL_DDR=0xff;
    DOT_ROW_DDR=0xff;
}
```

Ex-2 : main

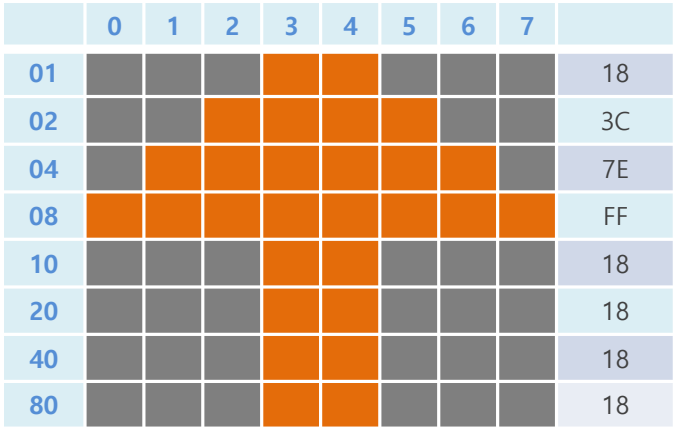
```
void Dot_Display( ){
    unsigned char tRow=ROW;
    for (int k=0; k<8;k++){
        DOT_ROW=DOT_ROW_TBL[k];
        DOT_COL=~DOT_COL_TBL[tRow];
        _delay_ms(dTime);
        if (++tRow>=8) tRow=0;
    }
}

int main(void) {
    CPU_Setup( );

    while (1) {
        for (int k=0;k<Speed;k++){
            Dot_Display();
        }
        if (++ROW>=8) ROW=0;
    }
}
```

Ex-3 : 방향 전환

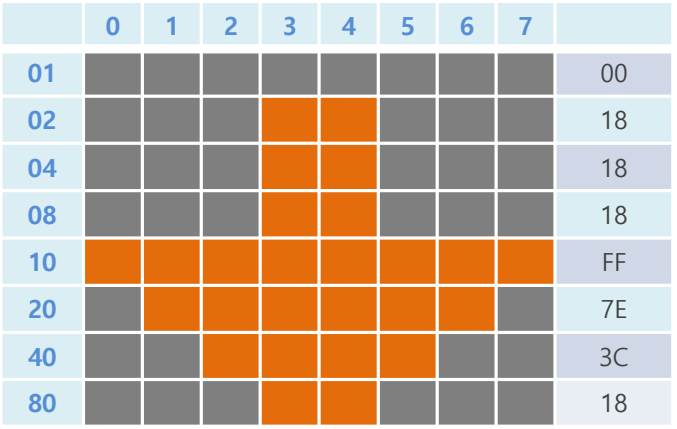
- 스위치 입력 받기
- 1번 : 정지 : 스마일 표시
- 2번 : UP : 화살표 UP
- 3번 : DN : 화살표 DN



	0	1	2	3	4	5	6	7	
01			■	■	■	■			3C
02		■					■		42
04	■		■			■		■	A5
08	■							■	81
10	■		■			■		■	A5
20	■			■	■			■	99
40		■					■		42
80			■	■	■	■			3C

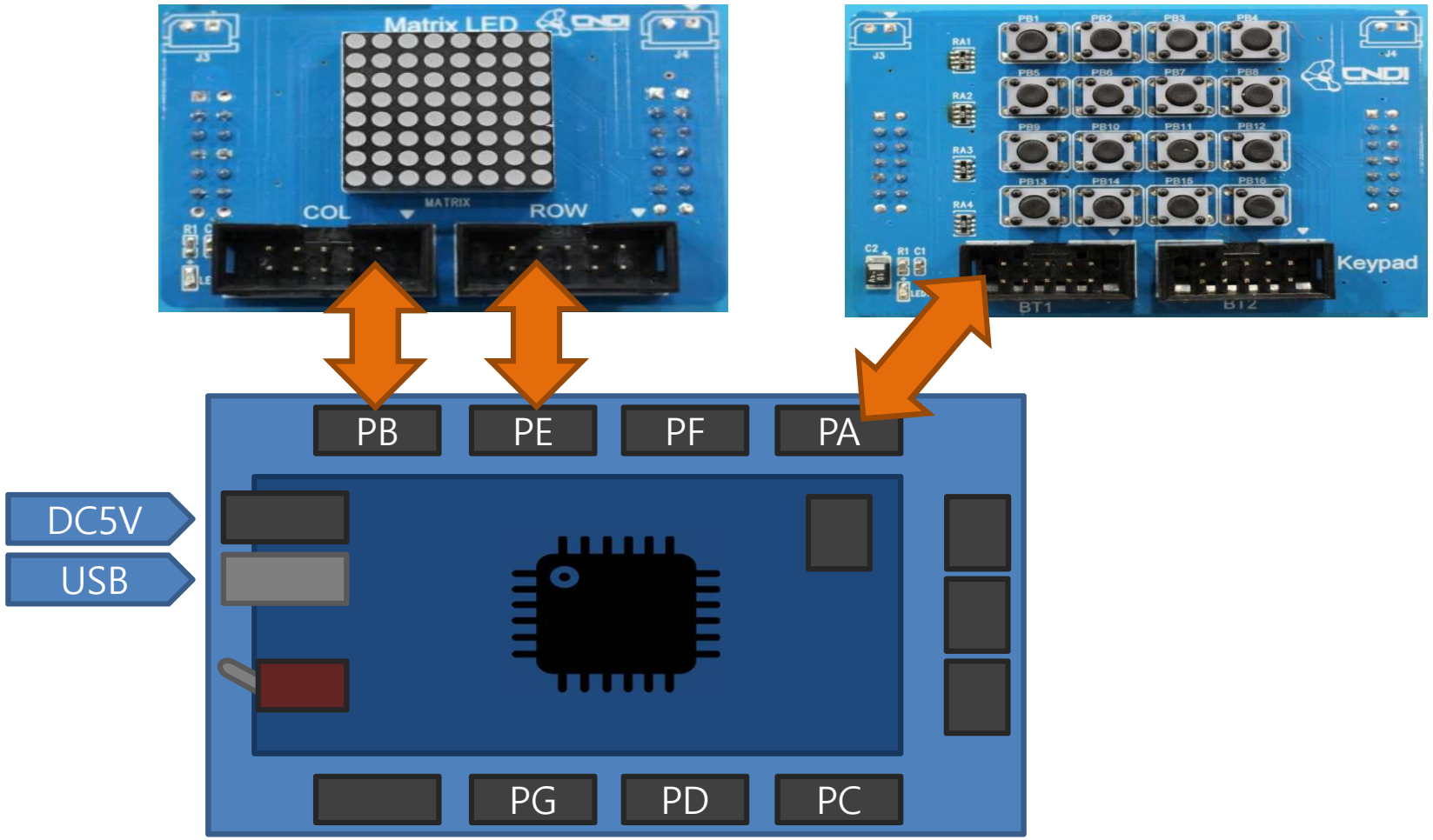


	0	1	2	3	4	5	6	7	
01				■	■				18
02			■	■	■	■			3C
04		■	■	■	■	■	■		7E
08	■	■	■	■	■	■	■	■	FF
10				■	■				18
20				■	■				18
40				■	■				18
80				■	■				18



	0	1	2	3	4	5	6	7	
01									00
02				■	■				18
04				■	■				18
08				■	■				18
10	■	■	■	■	■	■	■	■	FF
20		■	■	■	■	■	■		7E
40			■	■	■	■			3C
80				■	■				18

Ex-3 : Wiring


Matrix LED
COL
MATRIX
ROW
Keypad
BT1
BT2
PB
PE
PF
PA
DC5V
USB
PG
PD
PC

Ex-3 : Define

```
#define F_CPU 14745600UL
#include <avr/io.h>
#include <util/delay.h>

#define BTN PINA
#define BTN_DDR DDRA
#define DOT_COL PORTB
#define DOT_COL_DDR DDRB
#define DOT_ROW PORTE
#define DOT_ROW_DDR DDRE
#define dTime 3
#define Speed 15

unsigned char DOT_COL_TBL[3][8] ={ {0x3C,0x42,0xA5,0x81,0xA5,0x99,0x42,0x3C},
                                   {0x18,0x3C,0x7E,0xFF,0x18,0x18,0x18,0x00},
                                   {0x00,0x18,0x18,0x18,0xFF,0x7E,0x3C,0x18} };
unsigned char DOT_ROW_TBL[8] ={0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
signed char StartROW=0;
unsigned char DIR=0;
unsigned char KeyIn=0;
```

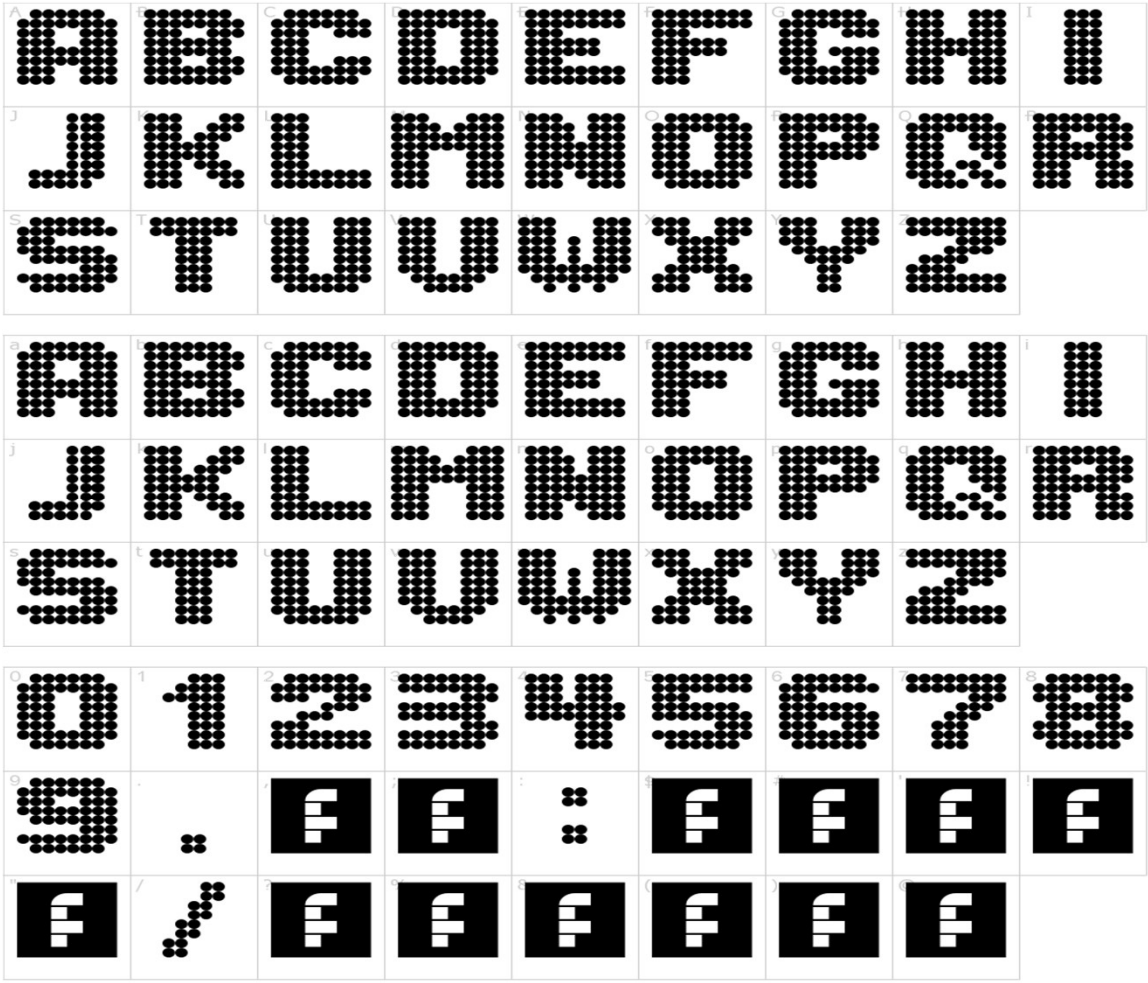
Ex-3 : Setup &Display

```
void CPU_Setup( ) {
     DOT_COL_DDR=0xff;
     DOT_ROW_DDR=0xff;
     BTN_DDR=0x00;
}

void Dot_Display(unsigned char Mode){
    unsigned char tRow=StartROW;

    for (char k=0; k<8;k++){
         DOT_ROW=DOT_ROW_TBL[k];
         DOT_COL=~DOT_COL_TBL[Mode][tRow];
         _delay_ms(dTime);

        if (++tRow>=8) tRow=0;
    }
}
```

Ex-3 : Define

```
int main(void) {
    CPU_Setup( );

    while (1) {
        for (int k=0;k<Speed;k++){
            Dot_Display(DIR);
        }

        KeyIn=~BTN;
        switch (KeyIn){
            case 0x00: break;
            case 0x01: DIR=0;   break;
            case 0x02: DIR=1;   break;
            case 0x04: DIR=2;   break;
            default:   DIR=0;     break;
        }

       switch (DIR){
            case 1:  if (++StartROW>=8) StartROW=0;   break;
            case 2:  if (--StartROW<0) StartROW=7;       break;
            default: StartROW=0;   break;
       }
    }
}
```

Ex-4 : 숫자 표시하기



Ex-4 : Number 1

	0	1	2	3	4	5	6	7	
01									38
02									3C
04									3E
08									38
10									38
20									38
40									38
80									38

Ex-4 : DotNumber.h

```
unsigned char DOT_SEL[8] ={0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80};

unsigned char DOT_NUM[10][8]={ {0x7e, 0xff, 0xe7, 0xe7, 0xe7, 0xe7, 0xff, 0x7e},    //0
                               {0x38, 0x3c, 0x3e, 0x38, 0x38, 0x38, 0x38, 0x38}, //1
                               {0x7e, 0xff, 0xe7, 0x70, 0x1c, 0x06, 0xff, 0xff},    //2
                               {0x7f, 0xff, 0xe0, 0x7f, 0x7f, 0xe0, 0xff, 0x7f},    //3
                               {0x77, 0x77, 0x77, 0xff, 0xff, 0x70, 0x70, 0x70},    //4
                               {0xff, 0xff, 0x07, 0x7f, 0xff, 0xe0, 0xff, 0x7e},    //5
                               {0x7e, 0xff, 0x07, 0x7f, 0xff, 0xe7, 0xff, 0x7e},    //6
                               {0xff, 0xff, 0xe0, 0x70, 0x38, 0x1c, 0x1c, 0x1c},    //7
                               {0x7e, 0xff, 0xe7, 0x7e, 0x7e, 0xe7, 0xff, 0x7e},    //8
                               {0x7e, 0xff, 0xe7, 0xff, 0xfe, 0xe0, 0xff, 0x7e} };  //9
```

Ex-4 : Define

```
#define F_CPU 14745600UL
#include <avr/io.h>
#include <util/delay.h>
#include "DotNumber.h"

#define DOT_COL PORTB
#define DOT_COL_DDR DDRB
#define DOT_ROW PORTE
#define DOT_ROW_DDR DDRE
#define dTime 3
#define Speed 50

unsigned char Count=0;

void CPU_Setup( ) {
    DOT_COL_DDR=0xff;
    DOT_ROW_DDR=0xff;
}
```

Ex-4 : main

```
void Dot_Display(unsigned char num){

    for (char k=0; k<8;k++){
        DOT_ROW=DOT_SEL[k];
        DOT_COL=~DOT_NUM[num][k];
        _delay_ms(dTime);
    }
}

int main(void) {
    CPU_Setup( );

    while (1) {
        for (int k=0;k<Speed;k++){
            Dot_Display(Count);
        }

        if (++Count>9) Count=0;
    }
}
```