2022년 IoT기반 스마트 솔루션 개발자 양성과정

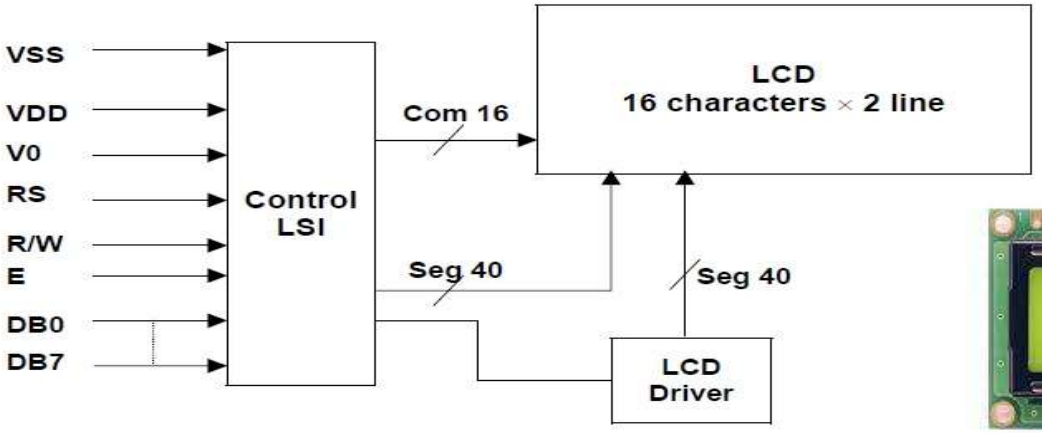
Embedded Application

6-Text LCD

담당 교수 : 윤 종 이
010-9577-1696
ojo1696@naver.com
https://cafe.naver.com/yoons2022

Text LCD

- Text LCD
  - Text LCD(Character LCD)는 액정을 이용하여 화소에 도달하는 빛을 선택적으로 투과시키거나 차단시켜 문자를 표시하는 시각적 전달장치
- Text LCD 기능 및 구조
  - TEXT LCD 구조는 LCD 패널(표시기)과 제어기가 함께 모듈 형태로 되어 있음
  - 제어기 내부에는 명령(Instruction) 레지스터, 데이터(Data) 레지스터, AC(Address Counter), BF(Busy Flag), 문자발생램(CGRAM), 문자발생롬(CGROM), 데이터표시램(DDRAM)이 있음



Text LCD Interface

- TEXT LCD의 인터페이스 핀 연결(PIN CONNECTIONS)

No.	Symbol	Function
1	VSS	Ground 0V
2	VDD	Logic power supply, +5V
3	V0	Voltage for LCD drive
4	RS	Data / Instruction register select
5	R/W	Read / Write
6	E	Enable signal, start data read/write
7	DB0	Data Bus Line
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	LED A	LED Anode, power supply +
16	LED K	LED Cathode, ground 0V

Text LCD 제어

- LCD 제어기
  - 명령 레지스터(IR) : DDRAM과 CGRAM에 관한 Clear Display, Cursor At Home, Function Set, Set Address 등의 제어명령을 가짐
  - 데이터 레지스터(DR) : DDRAM과 CGRAM에 쓰고 읽은 데이터를 일시적으로 저장함
    (RS(4번핀) 을 사용하여 데이터와 명령 레지스터를 선택)
  - BF : 1이면 LCD Controller가 동작 중으로 명령 수행 불능, 0이면 다음 명령 수행 가능
  - DDRAM(Data Display RAM) : 표시될 문자의 아스키(ASCII)코드가 저장되어 있는 메모리
  - CGRAM(Character Generator RAM) : 사용자가 원하는 문자를 만들기 위해 사용하는 메모리

Text LCD 제어

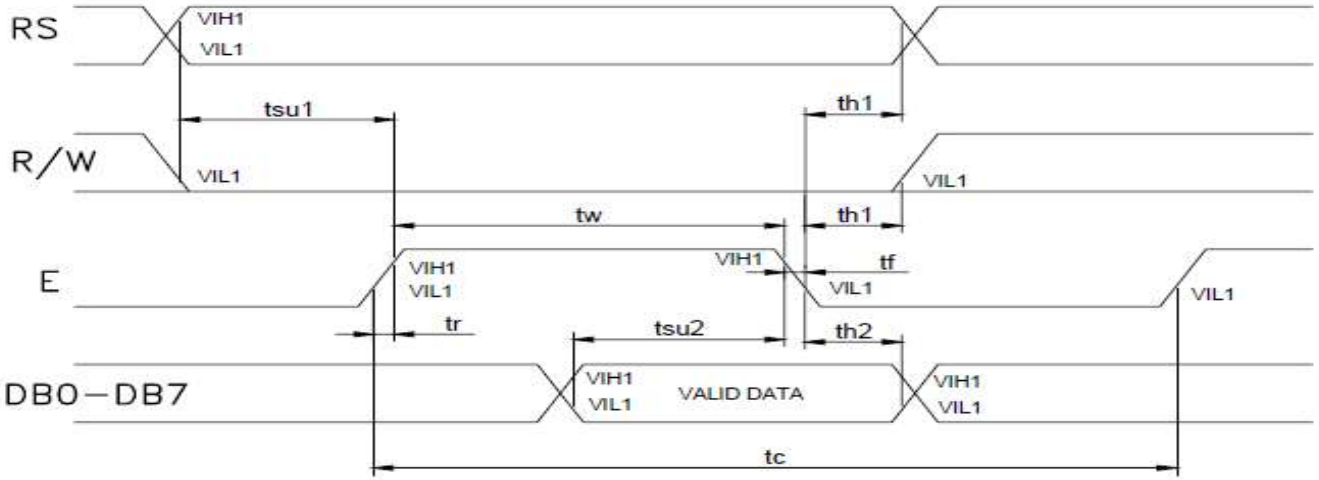
- 제어 방법
  - TEXT LCD 장치를 제어하기 위해서 제어기에 정해진 명령을 전달하고 명령에 따라 표시부에 문자 등을 표시.
  - LCD 모듈 제어를 위해 제어신호의 동작 타이밍은 매우 중요.

Text LCD Write Timing

Write cycle (Ta=25℃, VDD=5.0V)

Parameter	Symbol	Test pin	Min.	Typ.	Max.	Unit
Enable cycle time	t_c	E	500	-	-	ns
Enable pulse width	t_w		300	-	-	
Enable rise/fall time	t_r, t_f		-	-	25	
RS; R/W setup time	t_{su1}	RS; R/W	100	-	-	
RS; R/W address hold time	t_{h1}	RS; R/W	10	-	-	
Read data output delay	t_{su2}	DB0~DB7	60	-	-	
Read data hold time	t_{h2}		10	-	-	

Write mode timing diagram

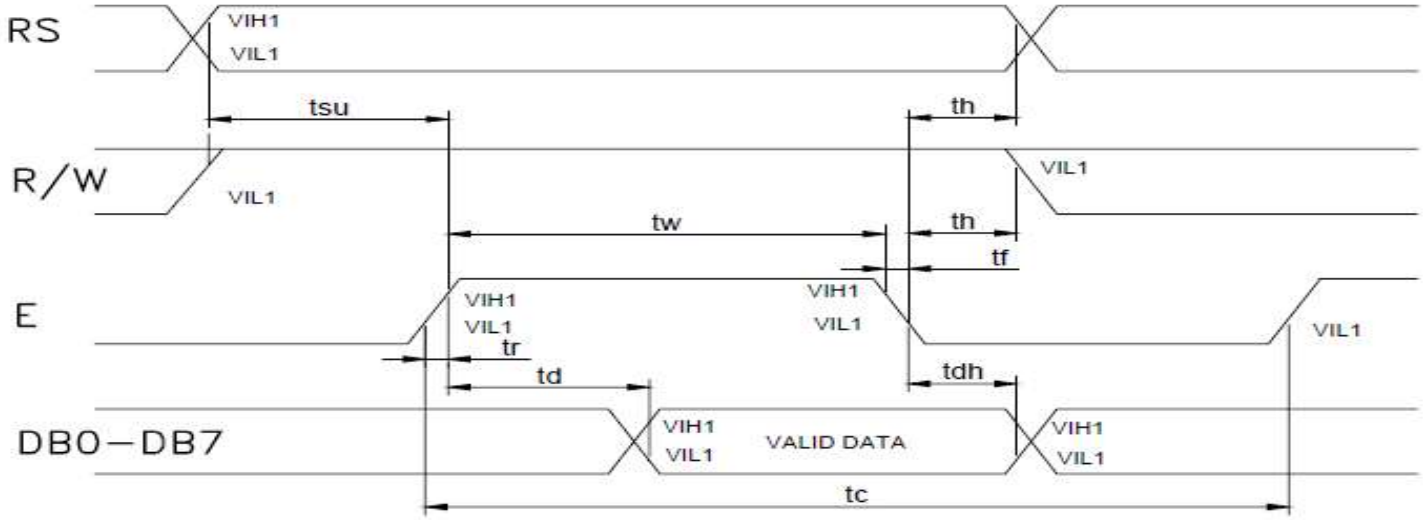


Text LCD Read Timing

Read cycle (Ta=25℃, VDD=5.0V)

Parameter	Symbol	Test pin	Min.	Typ.	Max.	Unit
Enable cycle time	t_c	E	500	-	-	ns
Enable pulse width	t_w		300	-	-	
Enable rise/fall time	t_r, t_f		-	-	25	
RS; R/W setup time	t_{su}	RS; R/W	100	-	-	
RS; R/W address hold time	t_h	RS; R/W	10	-	-	
Read data output delay	t_d	DB0~DB7	60	-	90	
Read data hold time	t_{dh}		20	-	-	

Read mode timing diagram



Text LCD 제어 명령

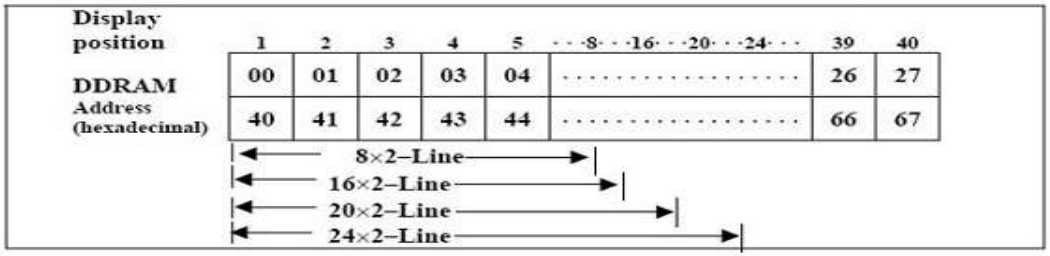
Instruction	Instruction Code										DESCRIPTION	Executed Time(fosc =270KHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write “20H” to DDRAM and set DDRAM address to “00H ” from AC	1.53mS
Cursor At Home	0	0	0	0	0	0	0	0	1	-	Set DDRAM address to “00H” from AC and return cursor to its original Position if shifted. The contents of DDRAM are not changed.	1.53mS
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and enable the shift of entire display.	39μS
Display On/Off Control	0	0	0	0	0	0	1	D	C	B	Set display (D), cursor(C), and Blinking of cursor(B) ON/OFF control bit.	39μS
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	-	-	Set cursor moving and display shifts cursor bit, and the direction, without changing of DDRAM data.	39μS
Function Set	0	0	0	0	1	DL	N	F	-	-	Sets interface data length (DL:8-BIT/4-BIT), number of display lines(N:2-line/1-line) and, display font type (F:5x11dots/5x8 dots).	39μS
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39μS
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39μS
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0μS
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM / CGRAM)	43μS
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Reads data from internal RAM (DDRAM / CGRAM).	43μS

*"-":don't care

NOTE : When an MPU program with checking the Busy Flag(DB7) is made, it must be necessary 1/2Fosc is necessary for executing the next instruction by the falling edge of the 'E' signal after the Busy Flag(DB7)goes to "LOW" .

Text LCD 주소/코드

- DDRAM 주소



- ASCII문자의 종류 및 코드 값

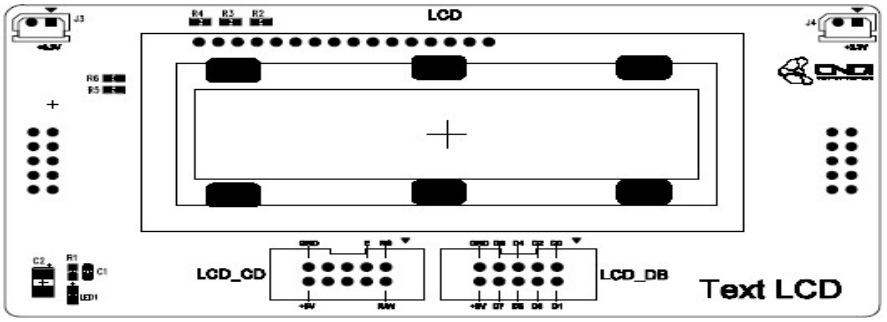
구분	00H	10H	20H	30H	40H	50H	60H	70H	80H	90H
0	사용자 정의 영역	미사용 영역		0	@	P	`	p	미사용 영역	
1			!	1	A	Q	a	q		
2			"	2	B	R	b	r		
3			#	3	C	S	c	s		
4			$	4	D	T	d	t		
5			%	5	E	U	e	u		
6			&	6	F	V	f	v		
7			'	7	G	W	g	w		
8			(	8	H	X	h	x		
9			)	9	I	Y	i	y		
A			*	:	J	Z	j	z		
B			+	;	K	[	k	{		
C			,	<	L	¥	l	|		
D			-	=	M	]	m	}		
E			.	>	N	^	n	→		
F			/	?	O	_	o	←		

8-Digit x 2-Line의 화면 구현을 위한 예제

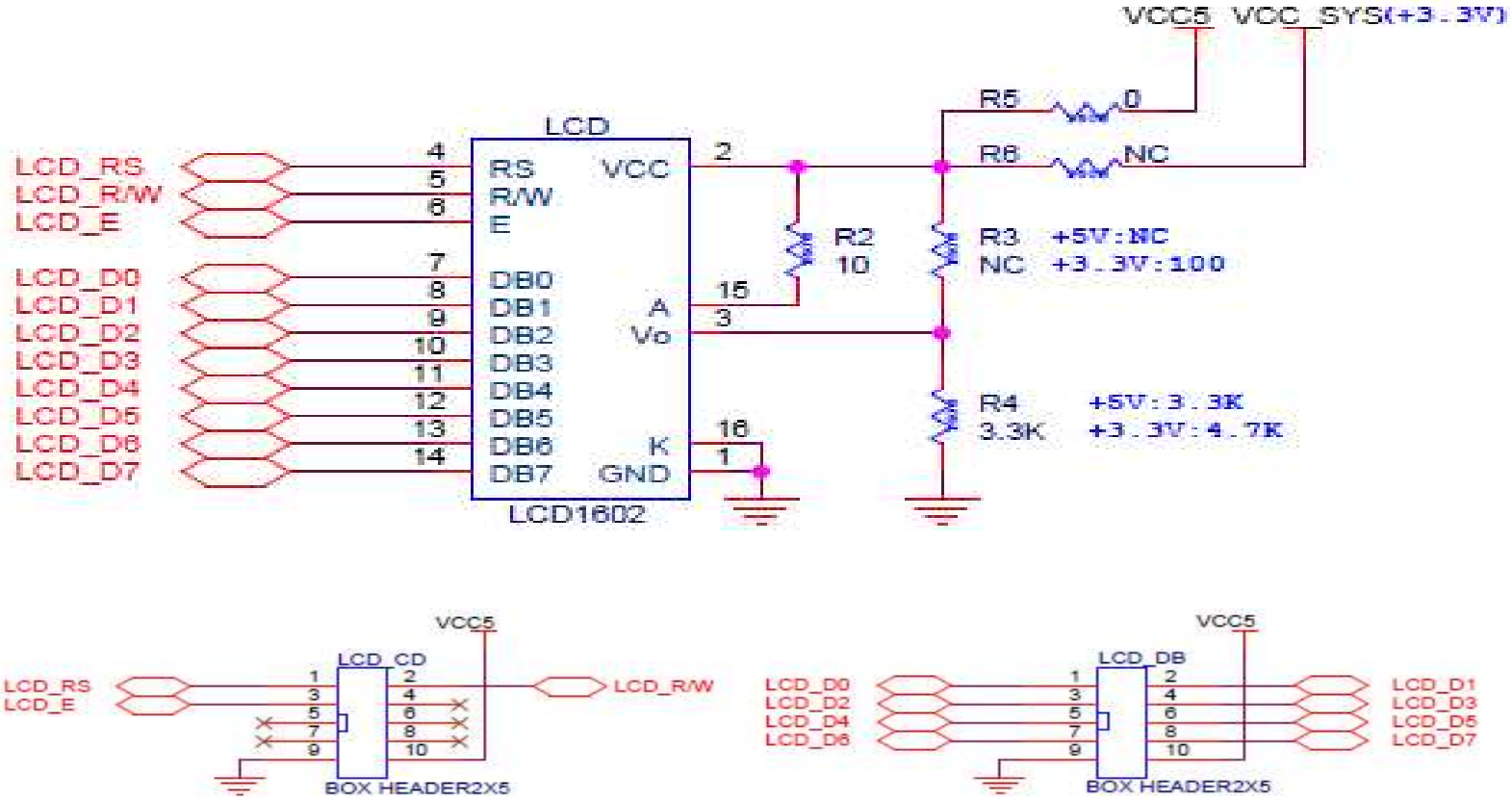
Step No	Instruction RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	Operation
1	Power supply on (the IC is initialized by the Internal reset circuit)											Initialized. No display.
2	Function set	0	0	0	0	1	1	1	0	*	*	Sets to 8-bit operation and selects 2-line display and 5×8 dot character font.
3	Display on/off control	0	0	0	0	0	0	1	1	1	0	Turns on display and cursor. All display is in space mode because of initialization.
4	Entry mode set	0	0	0	0	0	0	0	1	1	0	Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
5	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	0	0	0	Writes H. DDRAM has already been selected by initialization when the power was turned on. The cursor is incremented by one and shifted to the right
6	⋮										⋮	
7	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	0	0	1	Writes I.
8	Set DDRAM address	0	0	1	1	0	0	0	0	0	0	Sets DDRAM address so that t The cursor is positioned at the Head of the second lime.
9	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	1	0	1	Writes M.
10	⋮										⋮	
11	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	1	1	1	Writes O.
12	Entry mode set	0	0	0	0	0	0	0	1	1	1	Sets mode to shift display at the time of write.
13	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	1	0	1	Writes M. Display is shifted to the left. The first and second lines both shift at the same time.
14	⋮										⋮	
15	Return home	0	0	0	0	0	0	0	0	1	0	Returns both display and cursor to the original position (address 0).

Text LCD 제어

- 사용 모듈
  - AVR Module
  - TEXT LCD Module

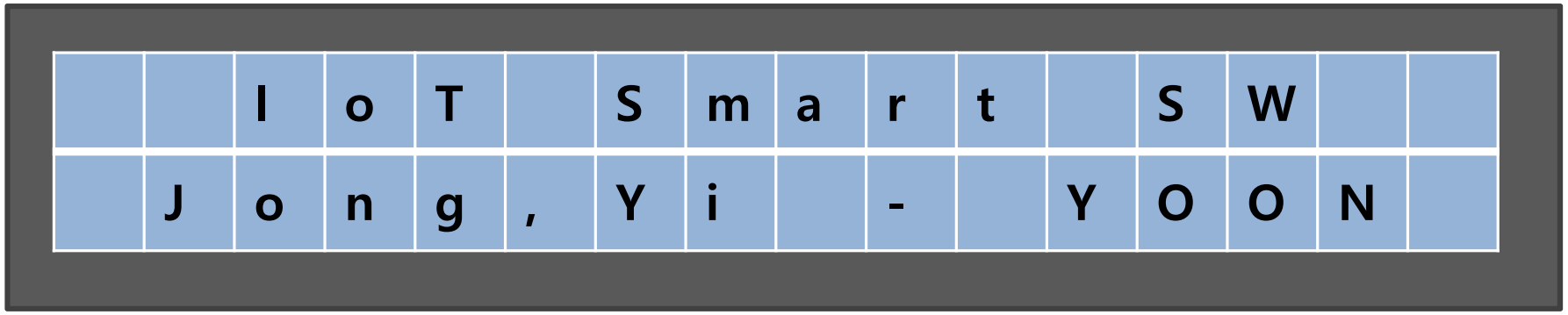


Text LCD Module Circuit

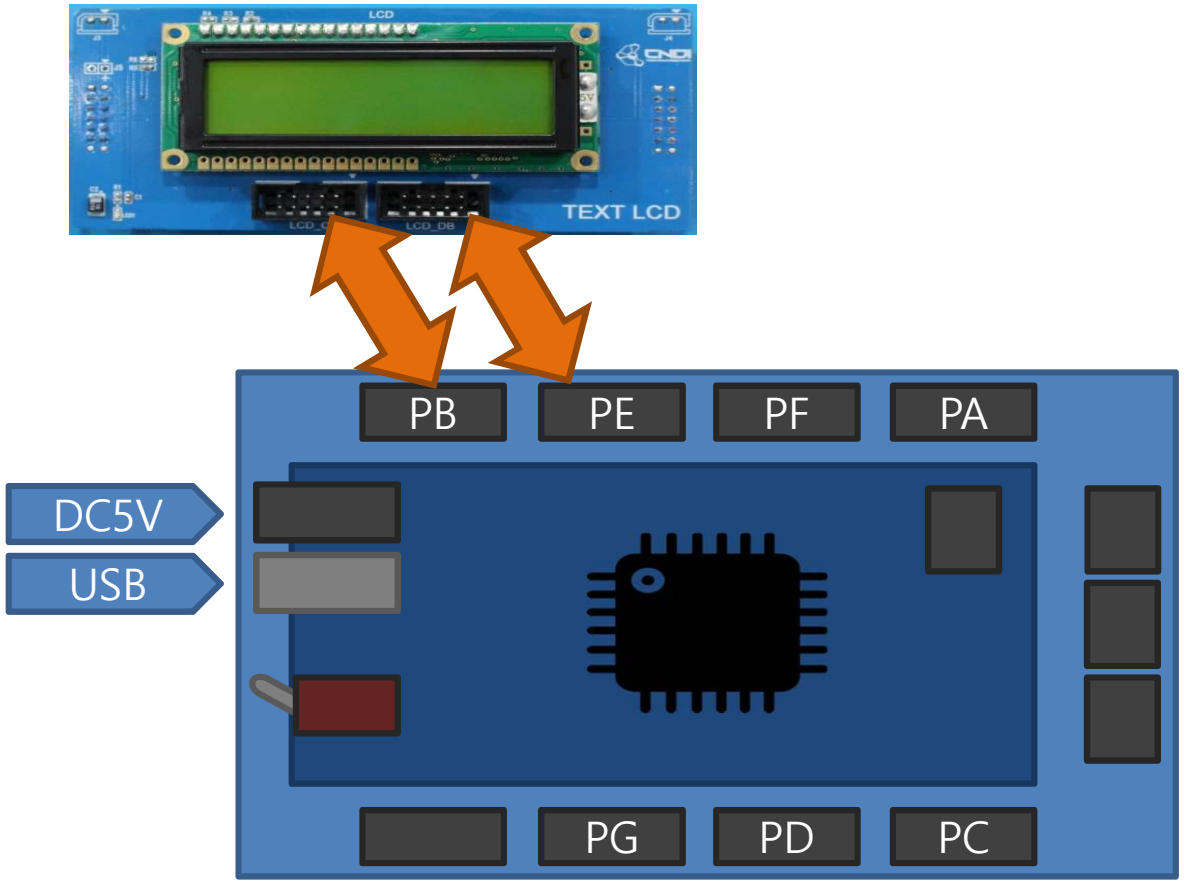


Ex-1 : Text LCD Display

Text LCD 에 다음과 같이 표시하여 보자



Ex-1 : Wiring



Ex-1 : TextLCD.c Write Sub

```
#define F_CPU 14745600UL
#define LCD_CD_PORT PORTB
#define LCD_CD_DDR  DDRB
#define LCD_DB_PORT PORTE
#define LCD_DB_DDR  DDRE

#include <avr/io.h>
#include <util/delay.h>

void write_Command ( unsigned char command ) {
   LCD_CD_PORT = 0x00;                    /* E = 0, R/W = 0, RS = 0 */
   LCD_DB_PORT = command;                 /* Command */

   LCD_CD_PORT |= 0x04;          _delay_us(110);          /* E = 1  Essential Delay for Simulator */
   LCD_CD_PORT &= ~(0x04);       _delay_us(110);          /* E = 0 */
}

void write_Data ( unsigned char data ) {
    LCD_CD_PORT = 0x00;                   /* RS = 0, R/W = 0, E = 0 */
    LCD_CD_PORT |= 0x01;                  /* RS = 1, DR->DDRAM */
    LCD_DB_PORT = data;

    LCD_CD_PORT |= 0x04;                  /* E = 1 */
    _delay_us(110);                       /* Essential Delay for Simulator */
    LCD_CD_PORT &= ~(0x04);               /* E = 0 */
    _delay_us(110);
}
```

Ex-1 : TextLCD.c Init/Print

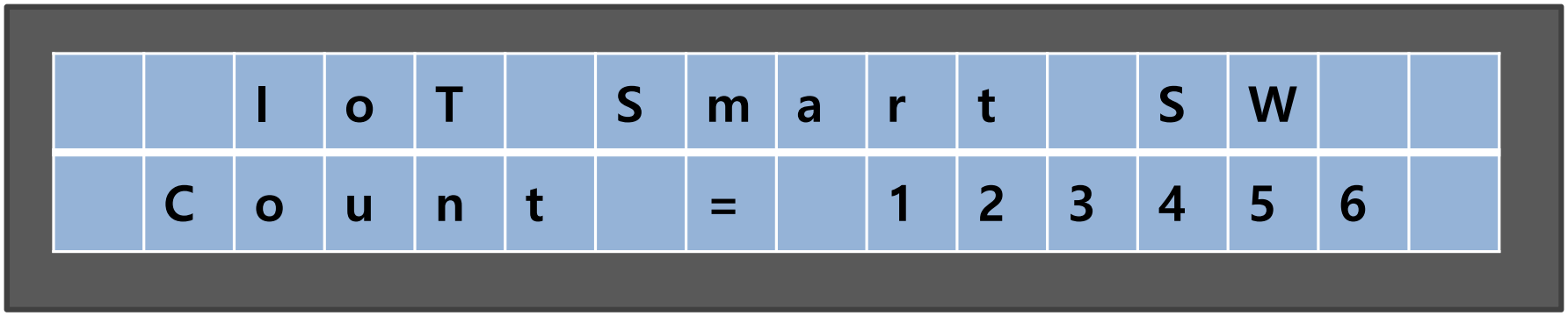
```
void printString ( char *string ) {
    while ( *string != '₩0' ) {
        write_Data ( *string );
        string ++;
    }
}

void LCD_Init (void) {
   LCD_DB_DDR=0xFF;
   LCD_DB_PORT=0x00;
   LCD_CD_DDR=0x07;              // 신호선 3 PIN  출력설정
   LCD_CD_PORT=0x00;

   _delay_us(110);
  LCD_CD_PORT &= ~(0x04);              _delay_us(110);          /* E = 0 */
  write_Command(0x38);                 _delay_us(220);          /* Function set */
  write_Command(0x0F);                 _delay_us(220);          /* Display ON */
  write_Command(0x02);                 _delay_ms(9);            /* Cursor At Home */
  write_Command(0x01);                 _delay_ms(9);            /* Clear Display */
  write_Command(0x06);                 _delay_us(220);          /* Entry mode set */
}
```

Ex-1 : main.c

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

int main(void){
    LCD_Init();

    while (1)  {
        write_Command(0x01);                /* Clear Display */
        _delay_ms(9);
        write_Command(0x80);                /* 1 Line Address */
        _delay_us(220);
        printString ( "   IoT Smart SW   " );
        write_Command(0xC0);                /* 2 line Address */
        _delay_us(220);
        printString ( " Jong,Yi - YOON " );
        _delay_ms(500);
    }
}
```

Ex-2 : Count Display

- Text LCD 에 다음과 같이 표시하여 보자
- Count 값은 0~999999



Ex-2 : Define

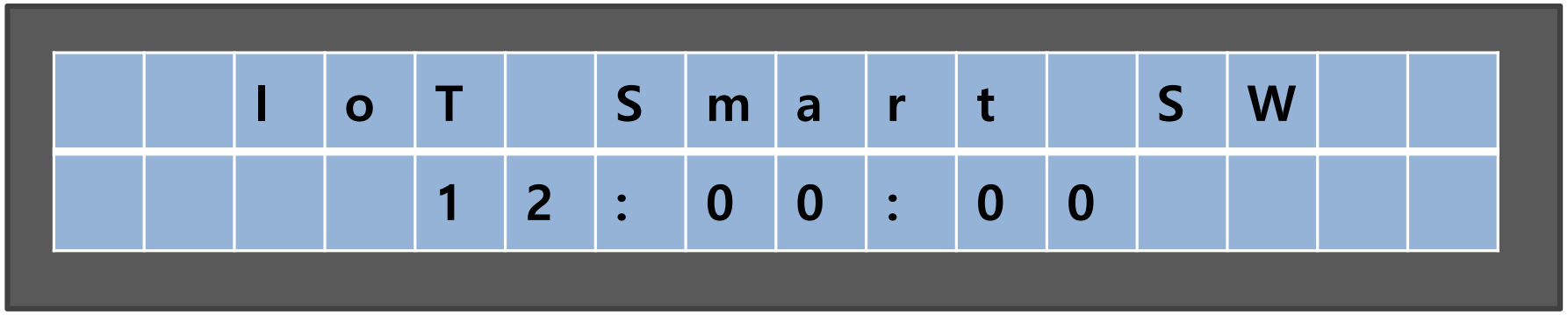
```
#define F_CPU 14745600UL
#include <avr/io.h>
#include <util/delay.h>

long Count=0;

unsigned char ASCII[17]={0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39, 0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x20};
unsigned char DISP[7];

void Hex2ASC(long No){
    long tmpNo=No;
    DISP[0]=ASCII[tmpNo/100000];
    tmpNo %=100000;
    DISP[1]=ASCII[tmpNo/10000];
    tmpNo %=10000;
    DISP[2]=ASCII[tmpNo/1000];
    tmpNo %=1000;
    DISP[3]=ASCII[tmpNo/100];
    tmpNo %=100;
    DISP[4]=ASCII[tmpNo/10];
    DISP[5]=ASCII[tmpNo%10];
}
```

Ex-2 : main

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

    write_Command(0x01);                _delay_ms(9);      /* Clear Display */
    write_Command(0x80);                _delay_us(220);    /* 1 Line Address */
    printString ( "  IoT Smart SW  " );
    write_Command(0xC0);                _delay_us(220);    /* 2 line Address */
    printString ( " Count = 000000 " );
    _delay_ms(500);

    while (1)   {
        Hex2ASC(Count);
        write_Command(0xC9);       _delay_us(220);
        printString ( DISP );          _delay_ms(500);

        if (++Count>999999) Count=0;
    }
}
```

Ex-3 : Clock Display

Text LCD 에 전자 시계를 구현 하여 보자



Ex-4 : 전화번호 표시

Text LCD 에 전화 번호 알림을 구현 하여 보자
전화번호는 좌측으로 부드럽게 흐르게 하자