

텍스트 LCD 제어

★ 텍스트 LCD 모듈 인터페이스 커넥터 핀의 기능

5. PIN DESCRIPTION

No.	Symbol	Function
1	VSS	GND(0V)
2	VDD	Power supply for Logic(+5.0V)
3	V0	Power supply for LCD drive
4	RS	Register selection (H: Data register , L :Instruction register)
5	R/W	Read/write selection (H: Read , L: Write)
6	E	Enable signal for LCM
7-14	DB0~DB7	Data Bus lines
15	LEDA	Power supply for Backlight(+5.0V)
16	LEDK	Power supply for Backlight(0V)

★ ARM7과 텍스트 LCD간의 핀 연결

- 1번(VSS) : GND
- 2번(VDD) : VCC5
- 3번(V0) : GND
- 4번(RS) : PA11
- 5번(R/W) : PA12
- 6번(E) : PA13
- 7~14번(DB0~DB7) : PA0~PA7
- 15번(LEDA) : VCC5
- 16번(LEDK) : PA14

★ 텍스트형 LCD 모듈의 동작 타이밍

2.AC Characteristics

(VDD = 4.5V ~ 5.5V, Ta = -30 °C ~ +85 °C)

Mode	Characteristic	Symbol	Min.	Typ.	Max.	Unit
Write Mode (Refer to Fig-6)	E Cycle Time	t _c	500	-	-	ns
	E Rise / Fall Time	t _p , t _f	-	-	20	
	E Pulse Width (High, Low)	t _w	230	-	-	
	R/W and RS Setup Time	t _{su1}	40	-	-	
	R/W and RS Hold Time	t _{h1}	10	-	-	
	Data Setup Time	t _{su2}	80	-	-	
	Data Hold Time	t _{h2}	10	-	-	
Read Mode (Refer to Fig-7)	E Cycle Time	t _c	500	-	-	ns
	E Rise / Fall Time	t _p , t _f	-	-	20	
	E Pulse Width (High, Low)	t _w	230	-	-	
	R/W and RS Setup Time	t _{su}	40	-	-	
	R/W and RS Hold Time	t _h	10	-	-	
	Data Output Delay Time	t _d	-	-	120	
	Data Hold Time	t _{DH}	5	-	-	

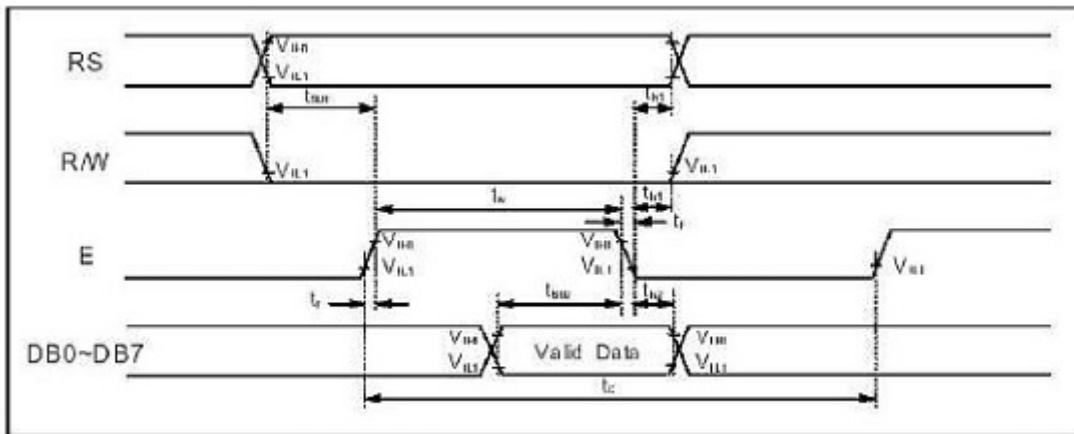


Figure 6 . Write Mode Timing Diagram

★ Write Mode 과정

1. R/W(PA12) : High, E : Low
2. RS(PA11) : High or Low
3. R/W(PA12) : High -> Low
----- 40 nSec이상 delay -----
4. E(PA13) : Low -> High
----- 150 nSec이상 delay -----
5. Data Bus(PA0~7) : 8비트 write
----- 80 nSec이상 delay -----
6. E(PA13) : High -> Low
----- 10 nSec이상 delay -----

★ 텍스트형 LCD 모듈의 제어 명령표

8. CONTROL AND DISPLAY INSTRUCTION

Instruction	Instruction Code										Description	Execution time (fosc=270KHz)
	RS	RW	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	1.52ms
Return Home	0	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.	1.52ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Assign cursor moving direction and enable the shift of entire display	38μs
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.	38μs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	-	-	Set cursor moving and display shift control bit, and the direction, without changing of DDRAM data.	38μs
Function Set	0	0	0	0	1	DL	N	F	-	-	Set interface data length (DL: 8-bit/4-bit), numbers of display line (N: 2-line/1-line) and, display font type (F: 5x10 dots/5x8 dots)	38μs
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	38μs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in counter	38μs
Read Busy Flag and Address Counter	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.	
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	38μs
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	38μs

- 명령 쓰기 루틴에서 자주 사용되는 명령

기능

제어 명령

동 작

Clear Display	0x01	LCD표시창을 클리어시키고 커서를 첫 줄의 첫 칸에 위치
Return Home	0x02	LCD의 표시내용을 그대로 두고 커서만은 홈으로 위치
Entry mode set	0x06	LCD표시창에 문자를 표시하고 커서를 오른쪽으로 이동
Display off	0x08	표시 off
Display on	0x0c	표시 on
control	0x0E 0x0F	표시 on 커서 on 표시 on 커서 on 블링크 on
Cursor or display shift	0x1C 0x18	현재 LCD에 표시되어 있는 내용을 오른쪽(0x1C), 왼쪽(0x18)으로 한 칸씩 이동
Function set	0x38 0x28	데이터 선 8비트, 2줄로 표시, 5x7도트사용 데이터 선 4비트, 2줄로 표시, 5x7도트사용

CG램 주소 0x40~0x7F

캐릭터 제너레이터 램의 주소값 설정

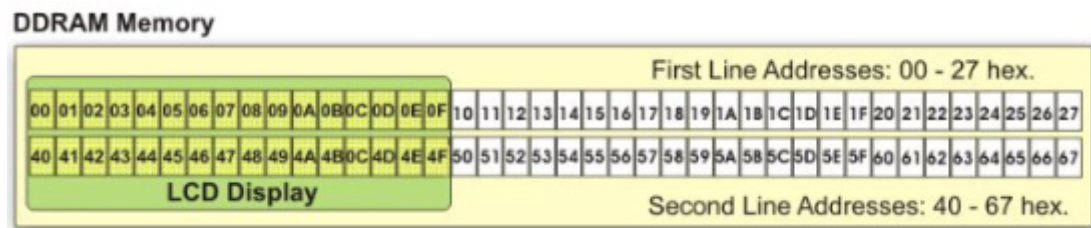
DD램 주소 0x80~0xff

데이터 디스플레이 램의 주소값 설정

★ 8비트 인터페이스의 경우 초기화 프로그래밍 순서

1. Function Set 명령.
2. Entry mode set 명령.
3. Display on/off control 명령.
4. DD RAM 어드레스 명령.
5. 표시할 문자 데이터를 보냄.
6. 필요한 경우 4~5번을 반복.

★ DDRAM Address



- 최상위 비트 1과 DDRAM Address를 OR연산하여 사용.

★ 캐릭터 LCD에 문자 출력 후 글자 시프팅하는 예제

```
/* -----*/
/* 2009 EHA Source - By Sung-bin Lee, 2009.06.03. */
/* -----*/

#include "project.h"

// CLCD 제어

#define D_BUS 0x000000FF // Data Bus (PA0~PA7)
#define RS 0x00000800 // Register Select(1=Data, 0=Instruction) (PA11)
#define RW 0x00001000 // Read/Write(1=Read, 0=Write) (PA12)
#define EN 0x00002000 // LCD Enable (PA13)
#define BL 0x00004000 // BackLight cathod (1=Off, 0=On) (PA14)

void LCD_Instruction(unsigned char ucData);
void LCD_Data(unsigned char ucData);
void LCD_Init();

int main()
{
```

```

volatile unsigned int iCnt = 0;
volatile unsigned int jCnt = 0;

unsigned char pStr[] = "Hello!!!";
unsigned char pStr1[] = "Welcome!";

*PIO_PER = D_BUS | RS | RW | EN | BL; // PIO Enable (PA0~PA7, PA11, PA12, PA13,
PA14)
*PIO_OER = D_BUS | RS | RW | EN | BL; // PIO Output Enable (PA0~PA7, PA11, PA12,
PA13, PA14)

LCD_Init(); // LCD 초기화

LCD_Instruction(0x06); // Entry Mode Set (커서 오른쪽증가, shift Off)
LCD_Instruction(0x80); // DD RAM Address (첫라인 첫위치)
for(iCnt=0; iCnt<8; ++iCnt){
    LCD_Data(pStr[iCnt]);
}
LCD_Instruction(0xC0); // DD RAM Address (두번째라인 첫위치)
for(iCnt=0; iCnt<8; ++iCnt){
    LCD_Data(pStr1[iCnt]);
}

for(iCnt=0; iCnt<8; ++iCnt){
    LCD_Instruction(0x1C); // Cursor or display shift (화면 변화지 않고 오른쪽으로 shift)
    for(jCnt=0; jCnt<1000000; ++jCnt);
}
for(iCnt=0; iCnt<8; ++iCnt){
    LCD_Instruction(0x18); // Cursor or display shift (화면 변화지 않고 왼쪽으로 shift)
    for(jCnt=0; jCnt<1000000; ++jCnt);
}

return 0;
}

void LCD_Init()
{
    LCD_Instruction(0x3c); // Function Set (8bit DataLength, 2Line, 5x11Dot)
    LCD_Instruction(0x1); // Display Clear
    LCD_Instruction(0x0c); // Display On/Off control (화면표시On, 커서Off, 커서깜빡임Off)
    LCD_Instruction(0x1); // Display Clear

```

```

}

void LCD_Instruction(unsigned char ucData)
{
    volatile unsigned int iCnt = 0;

    *PIO_CODR = EN;      // LCD 비활성화
    *PIO_CODR = RS;      // RS = 0 명령모드
    *PIO_CODR = RW;      // RW = 0 Write
    for(iCnt=0; iCnt<=50000;++iCnt);    // delay (최소 40nSec이상)
    *PIO_SODR = EN;      // LCD 활성화
    for(iCnt=0; iCnt<=160000;++iCnt);    // delay (최소 150nSec이상)
    *PIO_CODR = D_BUS;   // DataBus 초기화
    *PIO_SODR = ucData;
    for(iCnt=0; iCnt<=90000;++iCnt);    // delay (최소 80nSec이상)
    *PIO_CODR = EN;      // LCD 비활성화
    for(iCnt=0; iCnt<=15000;++iCnt);    // delay (최소 10nSec이상)

    return;
}

```

```

void LCD_Data(unsigned char ucData)
{
    volatile unsigned int iCnt = 0;

    *PIO_CODR = EN;      // LCD 비활성화
    *PIO_SODR = RS;      // RS = 1 Data모드
    *PIO_CODR = RW;      // RW = 0 Write
    for(iCnt=0; iCnt<=50000;++iCnt);    // delay (최소 40nSec이상)
    *PIO_SODR = EN;      // LCD 활성화
    for(iCnt=0; iCnt<=160000;++iCnt);    // delay (최소 150nSec이상)
    *PIO_CODR = D_BUS;   // DataBus 초기화
    *PIO_SODR = ucData;
    for(iCnt=0; iCnt<=90000;++iCnt);    // delay (최소 80nSec이상)
    *PIO_CODR = EN;      // LCD 비활성화
    for(iCnt=0; iCnt<=15000;++iCnt);    // delay (최소 10nSec이상)

    return;
}

```

'[ARM7](#)' 카테고리의 다른 글

<u>타이머카운터0(TC0)를 이용한 인터럽트</u> (1)	2009.06.08
<u>텍스트 LCD에서 사용자 정의문자 지정하기</u> (3)	2009.06.04
<u>텍스트 LCD 제어</u> (0)	2009.06.03
<u>디버그 유닛(DBGU)의 UART동작</u> (1)	2009.05.28
<u>PA0를 이용한 LED ON/OFF 제어</u> (0)	2009.05.28
<u>H-JTAG 설치 및 사용방법</u> (0)	2009.05.28

