

MEGA128_CLCD

(TEXT-LCD 메뉴얼(16*2))



CPUPLAZA

제품 소개

MEGA128_CLCD는 TEXT-LCD, Graphic-LCD(흑백), Color TFT-LCD를 연결하여 테스트 할 수 있으며, 압전식 터치 컨트롤러(TSC2003)가 내장되어 있어 LCD 모델에 상관없이 터치를 별도 헤더 핀으로 지원합니다.

제품 특징

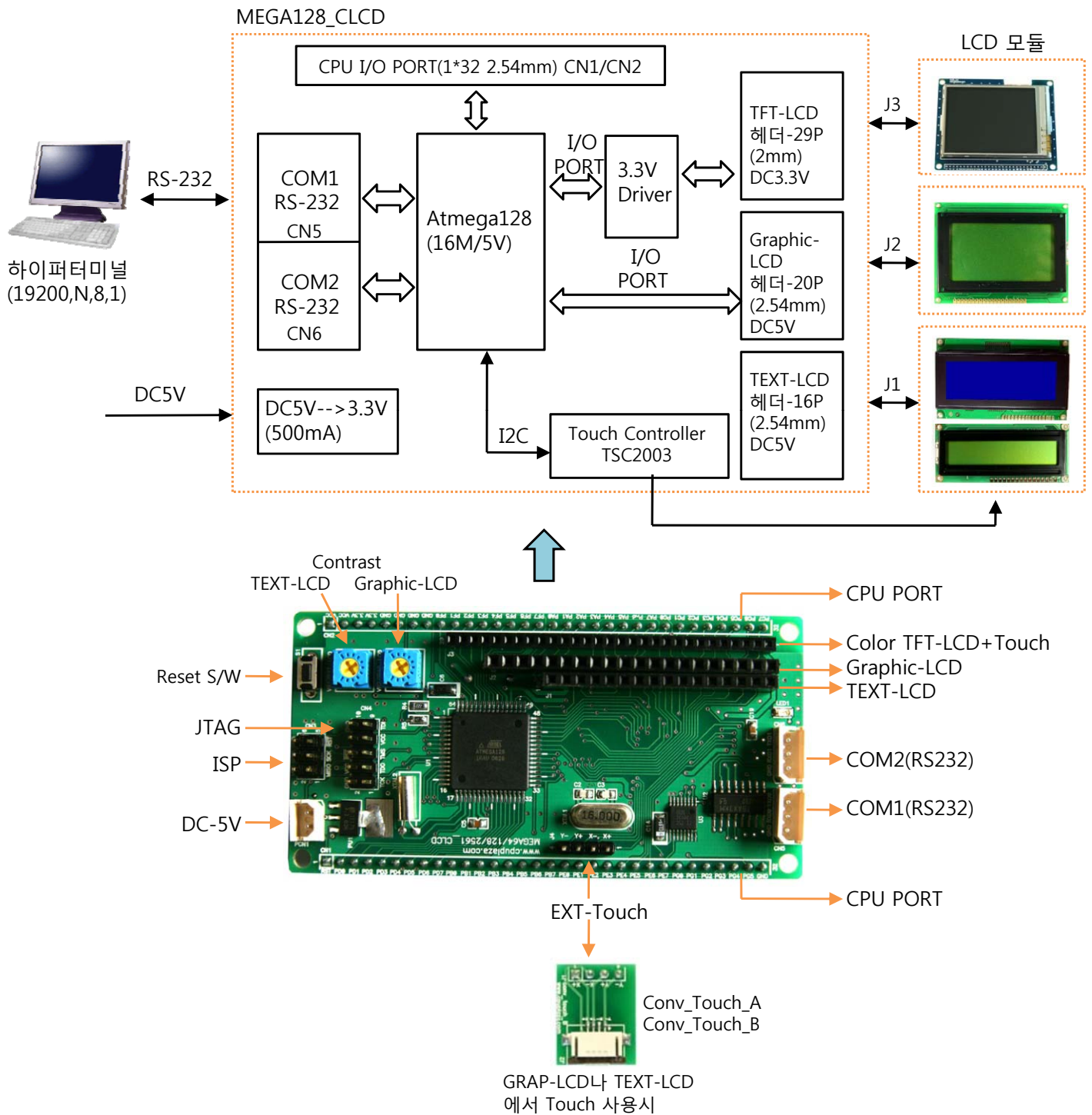
- 전원 : 외부 DC 5.0V
- TEXT LCD : 16*2 / 16*4 / 20*4 지원
 - I/O 방식 제어
- Graphic-LCD(흑백) : 128*64 지원
 - I/O 방식 제어
- Color(64K) TFT-LCD 지원 : 320*240 지원
 - 80-System 16Bit I/O 방식 제어
- 압전식 터치 지원 – TSC2003 Controller
- Atmega128 CPU 모든 I/O 1열 헤더 지원
- RS232 2PORT 포트 별도 지원
- 완성형 한글 지원(완성형 -> 조합형 코드 변환 사용)
- TEXT-LCD/Graphic-LCD/ Color TFT-LCD 라이브러리(소스 제공)

* TEXT-LCD(16*2)라이브러리 함수

. Clscr()	// 화면 소거
. Gotoxy()	// 커저 좌표 지정
. Putch() / Putchxyt()	// 한 문자 표시
. Puts() / Putsxyt()	// 문자열 표시
. Cursor_Set()	// Cursor ON/OFF
. LCD_Dread()	// LCD 데이터 취득

제품 사양

1. 제품 구성 및 명칭



2. 커넥터 기능 및 사양

PCN1 : DC 전원 입력

NUM	NAME
1	DC5V
2	GND

CN3 : ISP

NUM	NAME
1	MISO(TXD)
2	VCC
3	SCK
4	MOSI(RXD)
5	RST
6	GND

CN4 : JTAG

NUM	NAME
1	TCK
2	GND
3	TDO
4	VCC
5	TMS
6	RST
7	VCC
8	N.C
9	TDI
10	GND

J4 : 확장 Touch

NUM	NAME
1	X+
2	X-
3	Y+
4	Y-

J1 : TEXT-LCD

NUM	NAME	PORT
1	GND	GND
2	VCC	+5V
3	VO	가변 저항
4	RS	PE5
5	R/W	PG0
6	E	PE4
7-14	DB0-DB7	PA0-PA7
15	LEDA	+4.3V
16	LEDK	GND

J2 : Graphic-LCD

NUM	NAME	PORT
1	GND	GND
2	VCC	+5V
3	VO	가변 저항
4	RS	PE5
5	R/W	PG0
6	E	PE4
7-14	DB0-DB7	PA0-PA7
15	CS1	PC0
16	CS2	PC1
17	RSTW	PE6
18	VEE	
19	LEDA	+4.3
20	LEDK	GND

J3 : Color TFT-LCD(80-System)

NUM	NAME	PORT
1	VLCD	3.3V
2	VLCD	3.3V
3	GND	GND
4	GND	GND
5	RSTW	PE6
6	CSW	PG2
7	RS	PE5
8	RDW	PG1
9	WRW	PG0
10-17	DB0-DB7	PA0-PA7
18-25	DB8-DB15	PC0-PC7
26	X+	Touch X+
27	X-	Touch X-
28	Y+	Touch Y+
29	Y-	Touch Y-

CN1 : CPU PORT

NUM	NAME
1	RSTW
2	PD0
3	PD1
4	PD2
5	PD3
6	PD4
7	PD5
8	PD6
9	PD7
10	PB0
11	PB1
12	PB2
13	PB3
14	PB4
15	PB5
16	PB6
17	PB7
18	PE0
19	PE1
20	PE2
21	PE3
22	PE4
23	PE5
24	PE6
25	PE7
26	PG0
27	PG1
28	PG2
29	PG3
30	PG4
31	PENW
32	GND

CN2 : CPU PORT

NUM	NAME
1	VCC(+5V)
2	VCC(+5V)
3	3.3V
4	3.3V
5	GND
6	GND
7	GND
8	GND
9	PF0
10	PF1
11	PF2
12	PF3
13	PF4
14	PF5
15	PF6
16	PF7
17	PA0
18	PA1
19	PA2
20	PA3
21	PA4
22	PA5
23	PA6
24	PA7
25	PC0
26	PC1
27	PC2
28	PC3
29	PC4
30	PC5
31	PC6
32	PC7

CN5 : COM1-RS232

NUM	NAME
1	GND
2	ERXD0
3	ETXD0

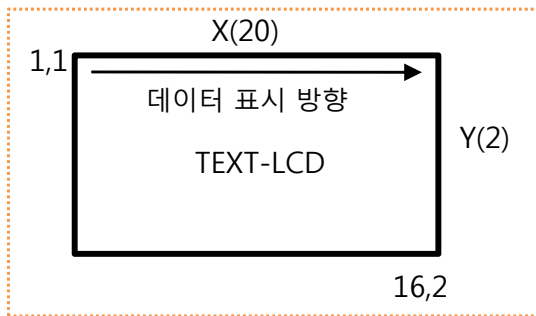
CN6 : COM2-RS232

NUM	NAME
1	GND
2	ERXD1
3	ETXD1

3. LCD 설명

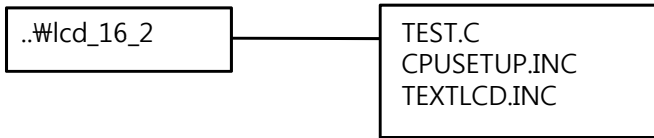
1. TEXT-LCD는 가로(20) 세로(2)의 영역에 영문 표시와 커저 제어 등을 지원하는 라이브러 함수를 제공 합니다.

- TEXT-LCD



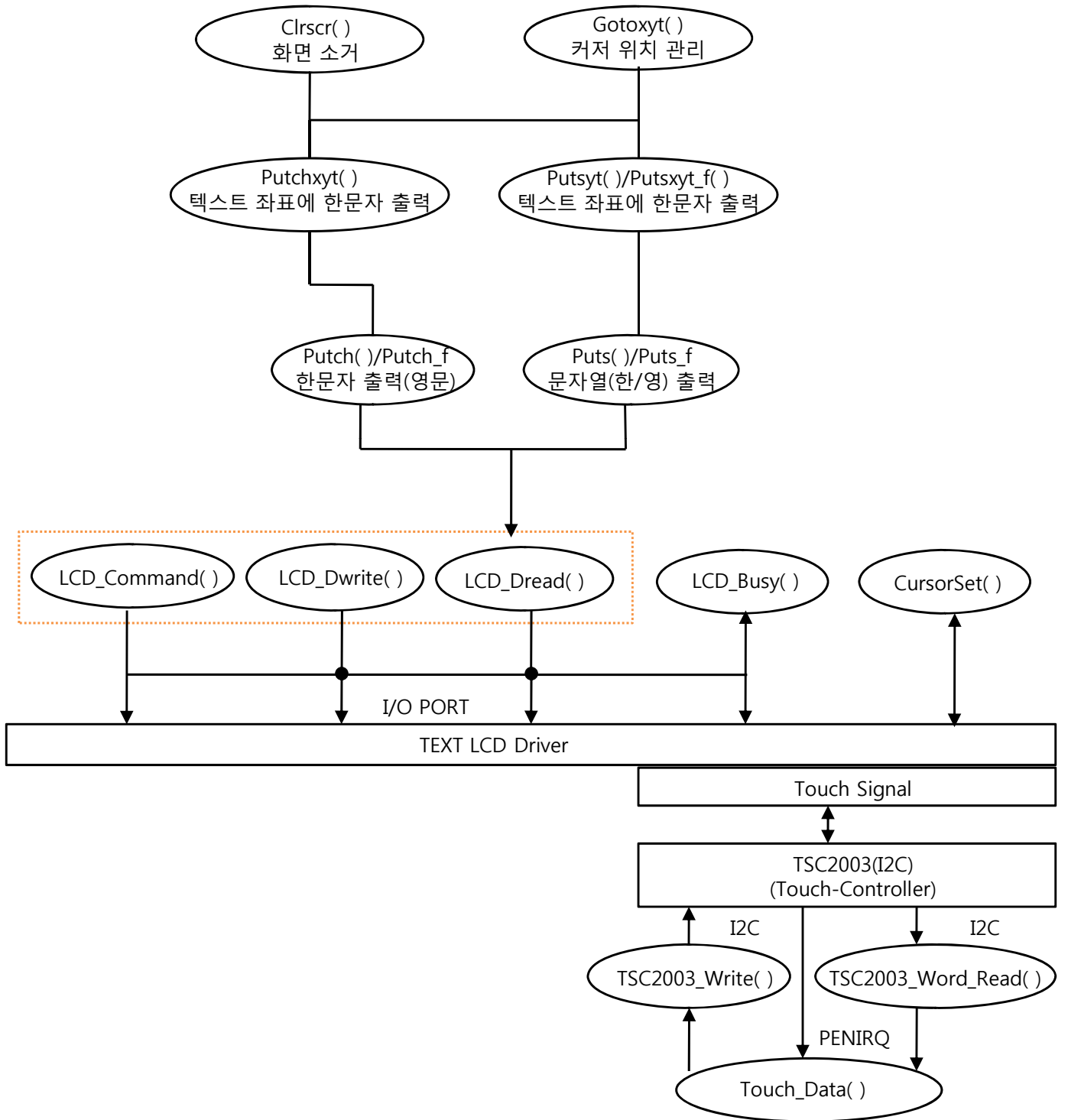
S/W 사양

1. 프로그램 구성



1. TEST.C : 메인 프로그램
통신(19200,N8,1)으로 LCD 테스트 프로그램 관리
2. CPUSETUP.INC :
인터럽트 관리 및 CPU 초기화 루틴
3. TEXTLCD.INC :
TEXT-LCD 관리 및 그래픽 라이브러리 소스

2. TEXT-LCD 라이브러리 구성

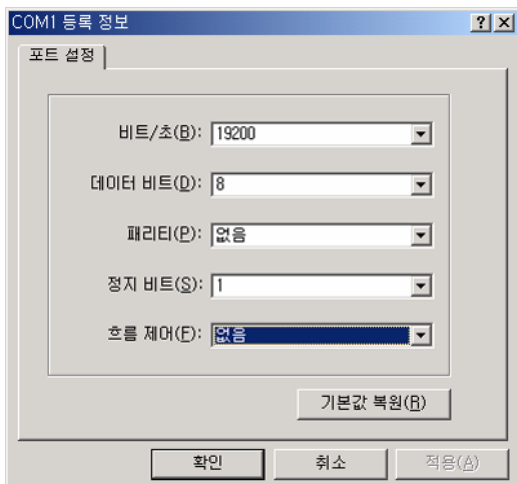
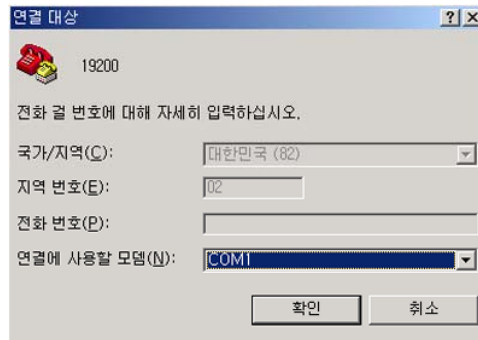
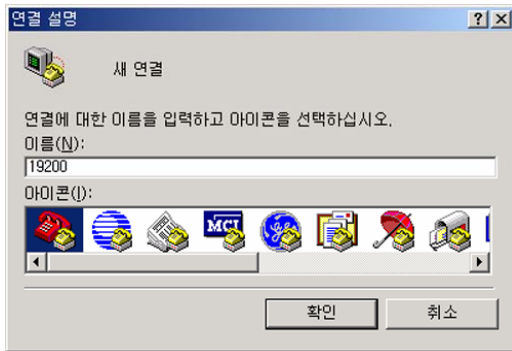


3. PC 통신 프로그램

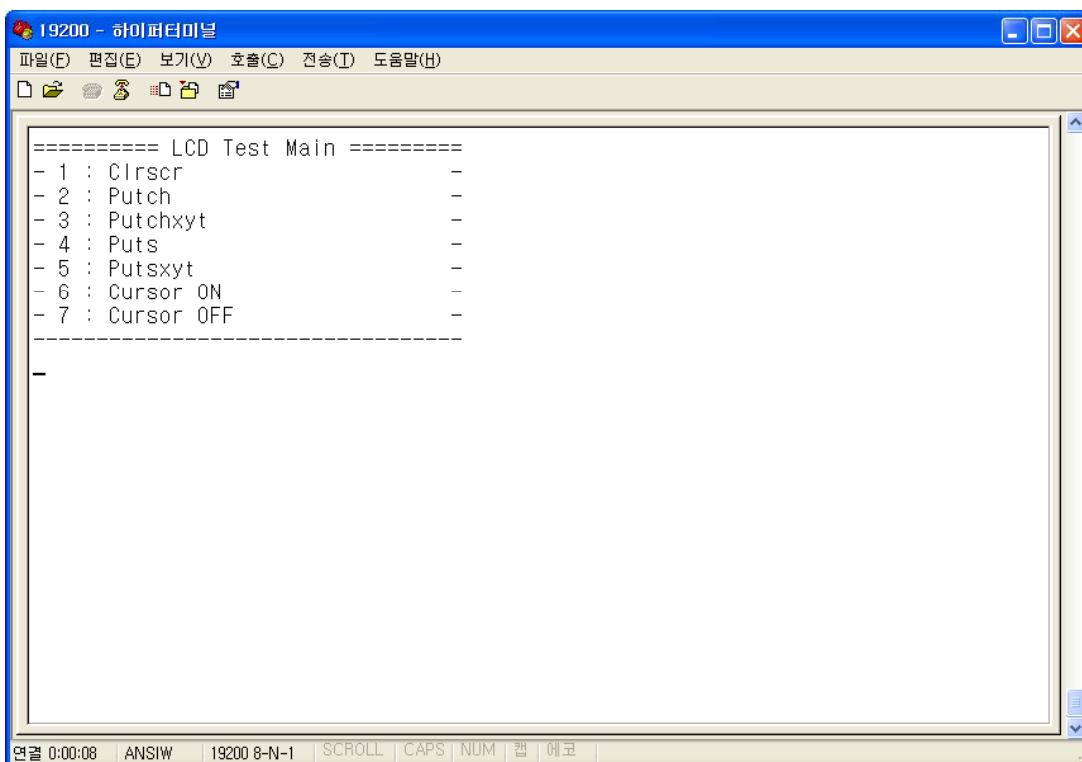
1. 하이퍼 터미널을 시작 합니다.



2. 하이퍼 터미널에서 통신 포트를 설정합니다.



3. 하이 터미널이 실행되면 보드를 Reset후 메뉴에 따라 키를 눌러 테스트 합니다.



LCD 라이브러리(textlcd.inc)

Clrscr

LCD 화면을 모두 지움.

함 수 : void Clrscr(void)

인 수 : 없음.

반 환 : 없음.

설 명 : 현재 화면을 모두 지운다.

[예제]

C기본 헤더 선언

```
.  
C기본 헤더 선언  
.br/>  
#include "cpusetup.inc"           // CPU 초기화 및 인터럽트 관리 프로그램  
#include "textlcd.inc"           // TEXT-LCD 라이브러리  
  
void main(void)  
{  
    Cpu_Setup();                 // CPU 초기화  
    Lcd_Init();                 // TEXT-LCD 초기화  
  
    // TEXT-LCD TEST Code..  
    Clrscr();                   // TEXT-LCD 화면 소거.  
    while(1)  
    {  
        사용자 프로그램  
    }  
}
```

* 실행 결과(LCD 화면)



Cursor_Set

커저 표시를 ON/OFF 한다.

함 수 : void Cursor_Set(char sts)

인 수 : char sts

ON

OFF

반 환 : 없음.

// 커저 ON(1)/OFF(0)

// 커저를 켜

// 커저를 끄

설 명 : LCD상에 커저 표시를 제어 한다.

[예제]

.
C기본 헤더 선언

.
#include "cpusetup.inc"

// CPU 초기화 및 인터럽트 관리 프로그램

#include "textlcd.inc"

// TEXT-LCD 라이브러리

void main(void)

{

Cpu_Setup();

// CPU 초기화

Lcd_Init();

// TEXT-LCD 초기화

// TEXT-LCD TEST Code..

Cursor_Set(ON);

// 커저를 켜

while(1)

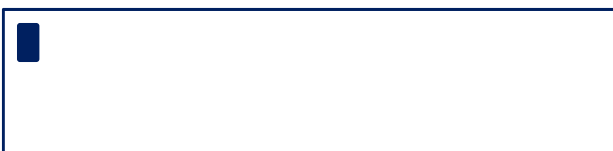
{

사용자 프로그램

}

}

* 실행 결과(LCD 화면)



Gotoxyt

커져 좌표 를 지정.

함 수 : void Gotoxyt(char x,char y)

인 수 : char x

char y

반 환 : 없음.

// 커져의 x(가로) 좌표(1 - 16)

// 커져의 y(세로) 좌표(1 - 2)

설 명 : 커져 위치 지정.

문자를 표현하는 함수에서 사용.

* 16 * 2 TEXT-LCD 일때

관련 함수 : Putch,Puts

[예제]

.

C기본 헤더 선언

.

```
#include "cpusetup.inc"
```

// CPU 초기화 및 인터럽트 관리 프로그램

```
#include "textlcd.inc"
```

// TEXT-LCD 라이브러리

```
void main(void)
```

```
{
```

```
    Cpu_Setup();
```

// CPU 초기화

```
    Lcd_Init();
```

// TEXT-LCD 초기화

```
// TEXT-LCD TEST Code..
```

```
    Gotoxyt(1,2);
```

// TEXT-LCD 커져 위치 지정.

```
    while(1)
```

```
    {
```

```
        사용자 프로그램
```

```
    }
```

```
}
```

Putch

현재 커저 위치에 문자를 표시한다.

함 수 : void Putch(char data)
인 수 : char data // 표시할 문자
반 환 : 없음.

설 명 : 현재 커저 위치에 문자를 표시 한다.
관련 함수 : Puchxyt

[예제]

.
C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램  
#include "textlcd.inc" // TEXT-LCD 라이브러리
```

```
void main(void)  
{  
    Cpu_Setup(); // CPU 초기화  
    Lcd_Init(); // TEXT-LCD 초기화  
  
    // TEXT-LCD TEST Code..  
    Putch('2'); // 화면에 '2' 표시  
    while(1)  
    {  
        사용자 프로그램  
    }  
}
```

* 실행 결과(LCD 화면)

2

Putchxyt

지정한 좌표에 문자를 표시한다.

함 수 : void Putchxyt(char x, char y, char data)
인 수 : char x // 커저의 x(가로) 좌표(1 - 16)
 char y // 커저의 y(세로) 좌표(1 - 2)
 char c // 표시할 문자
반 환 : 없음.

설 명 : 지정한 좌표에 문자를 표시 한다.
 * 16 * 2 TEXT-LCD 일때

[예제]

C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램
#include "textlcd.inc" // TEXT-LCD 라이브러리

void main(void)
{
    Cpu_Setup(); // CPU 초기화
    Lcd_Init(); // TEXT-LCD 초기화

    // TEXT-LCD TEST Code..
    Putchxyt(2,1,'2'); // 화면의 2(x),1(y)위치에 '2' 표시
    while(1)
    {
        사용자 프로그램
    }
}
```

* 실행 결과(LCD 화면)



2

Puts

현재 커저 위치에 문자열을 표시한다.

함 수 : void Puts(char *stbuf[])
인 수 : char *stbuf[] // 표시할 문자열
반 환 : 없음.

설 명 : 현재 커저 위치에 문자열을 표시 한다.
관련 함수 : Putsxyt

[예제]

.
C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램  
#include "textlcd.inc" // TEXT-LCD 라이브러리
```

```
void main(void)  
{  
    Cpu_Setup(); // CPU 초기화  
    Lcd_Init(); // TEXT-LCD 초기화  
  
    // TEXT-LCD TEST Code..  
    Puts("TEST LCD"); // 화면에 TEST LCD 표시  
    while(1)  
    {  
        사용자 프로그램  
    }  
}
```

* 실행 결과(LCD 화면)

TEST LCD

Putsxyt

지정한 좌표에 문자열을 표시한다.

함 수 : void Putsxyt(char x, char y, char stbuf[])
인 수 : char x // 커저의 x(가로) 좌표(1 - 16)
 char y // 커저의 y(세로) 좌표(1 - 2)
 char str[] // 표시할 문자열
반 환 : 없음.

설 명 : 지정한 좌표에 문자열을 표시 한다.
 * 16 * 2 TEXT-LCD 일때

[예제]

C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램
#include "textlcd.inc" // TEXT-LCD 라이브러리

void main(void)
{
    Cpu_Setup(); // CPU 초기화
    Lcd_Init(); // TEXT-LCD 초기화

    // TEXT-LCD TEST Code..
    Putsxyt(2,1,"TEST LCD"); // 화면의 2(x),1(y)위치에 TEST LCD 표시
    while(1)
    {
        사용자 프로그램
    }
}
```

* 실행 결과(LCD 화면)

TEST LCD

LCD_Dread

현재 커저 위치의 데이터를 취득

함 수 : nsigned char LCD_Dread(void)

인 수 : void

반 환 : 00 - ffh

설 명 : 현재 커저 위치의 데이터를 취득.

[예제]

.
C기본 헤더 선언

```
.  
#include "cpusetup.inc"           // CPU 초기화 및 인터럽트 관리 프로그램  
#include "textlcd.inc"           // TEXT-LCD 라이브러리  
  
void main(void)  
{  
    Cpu_Setup();                 // CPU 초기화  
    Lcd_Init();                  // TEXT-LCD 초기화  
  
    // TEXT-LCD TEST Code..  
    uch = LCD_Dread();           // 데이터 리드  
    printf("byte read : %02x\n",uch); // 데이터 표시  
    while(1)  
    {  
        사용자 프로그램  
    }  
}
```

LCD_Init

TEXT-LCD Driver IC를 초기화 한다.

함 수 : void Lcd_Init(void)

인 수 : void

반 환 : void

설 명 : TEXT-LCD Driver IC를 초기화 한다.

[예제]

```
.  
C기본 헤더 선언  
.  
  
#include "cpusetup.inc"           // CPU 초기화 및 인터럽트 관리 프로그램  
#include "textlcd.inc"           // TEXT-LCD 라이브러리  
  
void main(void)  
{  
    Cpu_Setup();                 // CPU 초기화  
    Lcd_Init();                 // TEXT-LCD 초기화  
    while(1)  
    {  
        사용자 프로그램  
    }  
}
```

LCD_Command

TEXT-LCD Driver IC에 command 데이터를 써넣는다.

함 수 : void LCD_Command(unsigned char uch)

인 수 : unsigned char uch // command 데이터

반 환 : void

설 TEXT-LCD Driver IC에 command 데이터를 써넣는다

LCD_Dwrite

TEXT-LCD에 데이터를 써넣는다.

함 수 : void LCD_Dwrite(unsigned char uch)

인 수 : unsigned char uch // command 데이터

반 환 : void

설 명 : TEXT-LCD에 데이터를 써넣는다.

GLCD_Busy

TEXT-LCD Driver IC의 Busy 상태를 읽음.

함 수 : void GLCD_Busy(void)

인 수 : void

설 명 : TEXT-LCD Driver IC의 Busy 상태를 읽음.