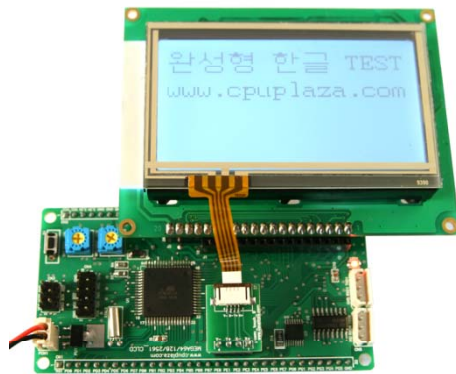


MEGA128_CLCD

(압전식 터치 메뉴얼)



CPUPLAZA

www.cpublaza.com

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.

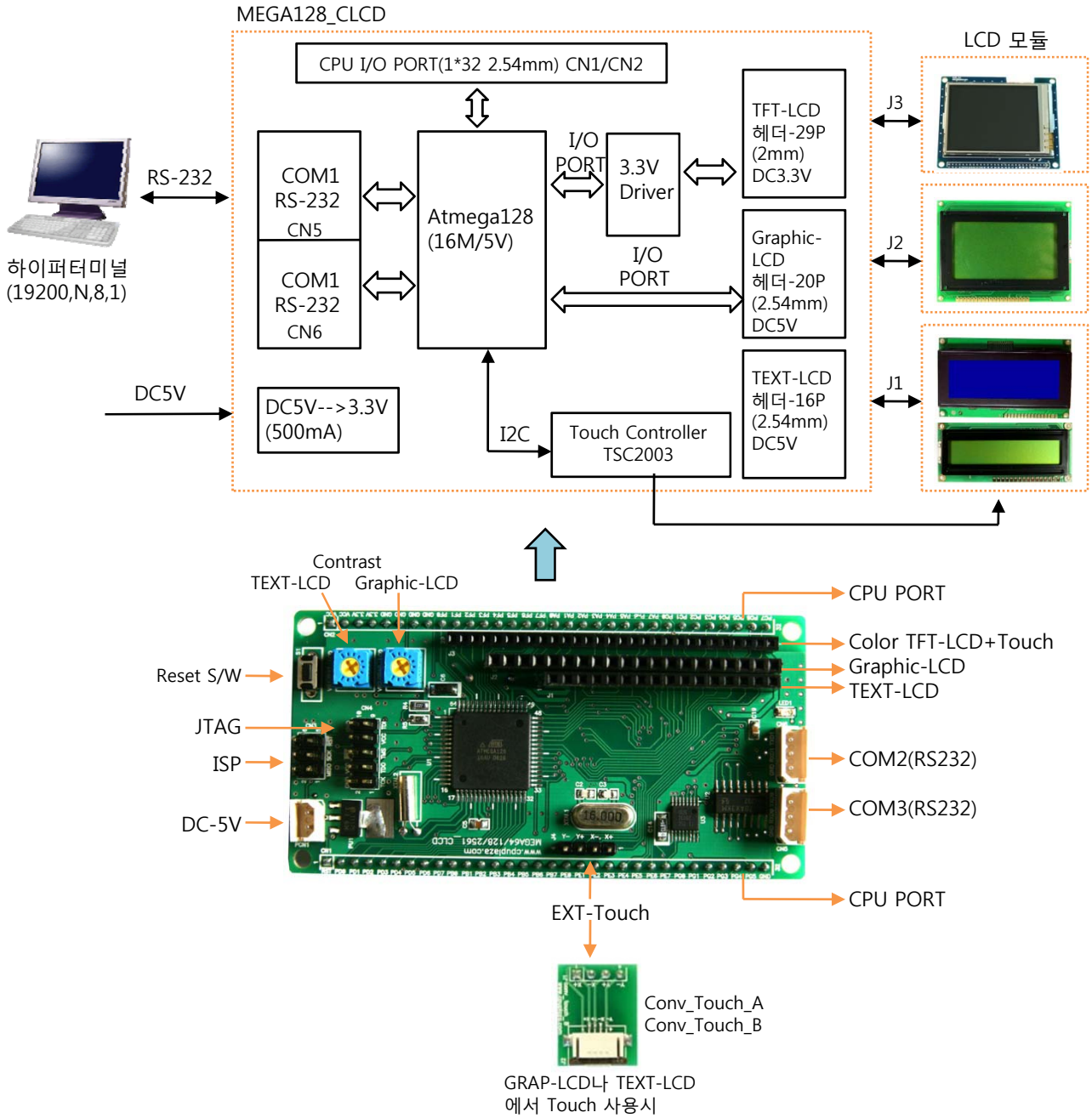
제품 특징

- 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)
- Touch 라이브러리(소스 제공)

```
. Touch_Data( )           // 터치 좌표 데이터 취득
. TSC2003_Write( )        // 터치 IC 데이터 라이트
. TSC2003_Data( )        // 터치 IC 데이터 리드
. TSC2003_Word_Read( )   // 터치 IC 데이터 전송후 리드
```

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 |

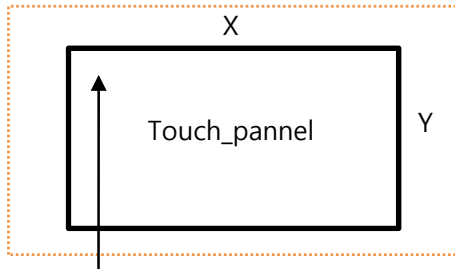
CN2 : CPU PORT

| 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 |

3. Touch 설명

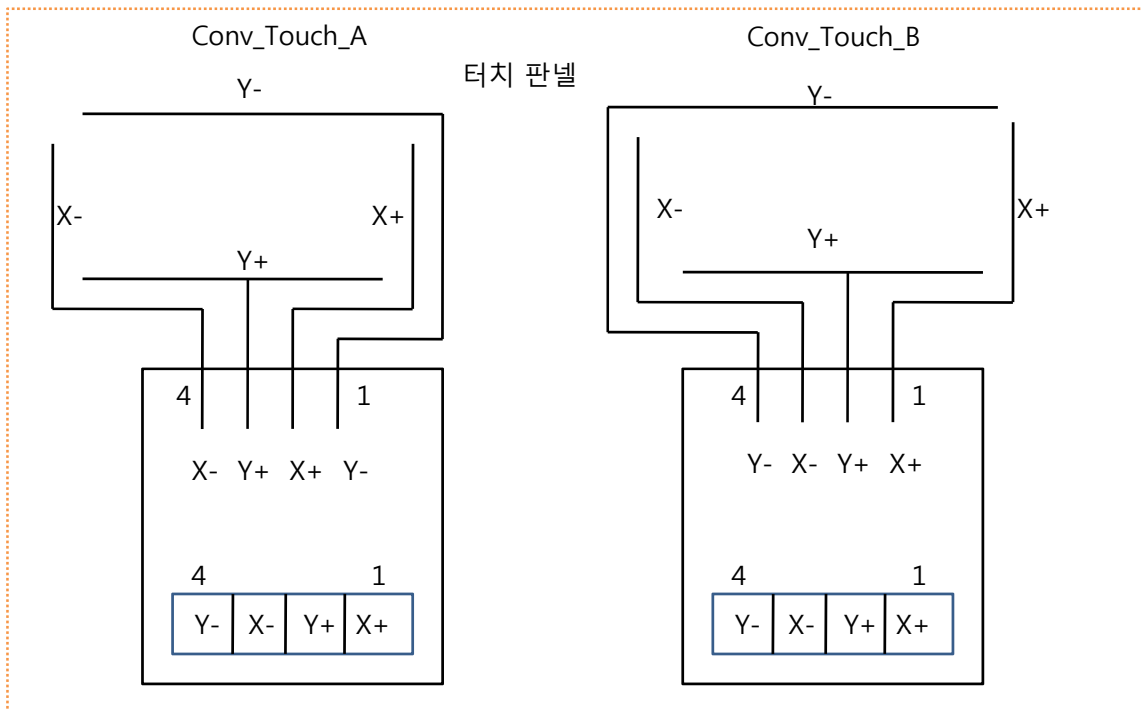
1. Touch Driver IC는 CPU와 I2C로 연결되며 터치를 누를때 A/D 12Bit 좌표 데이터를 읽어서 처리합니다.
터치와의 결선은 모델마다 차이가 있으므로 터치 매뉴얼을 보시고 변환보드를 사용해 제품에 연결 하시면 됩니다.

- Touch



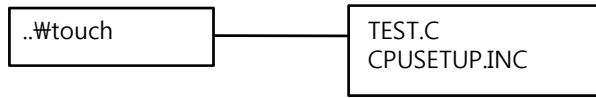
원점(0,0) : S/W로 옵셋 조정

- Touch모델별 변환보드



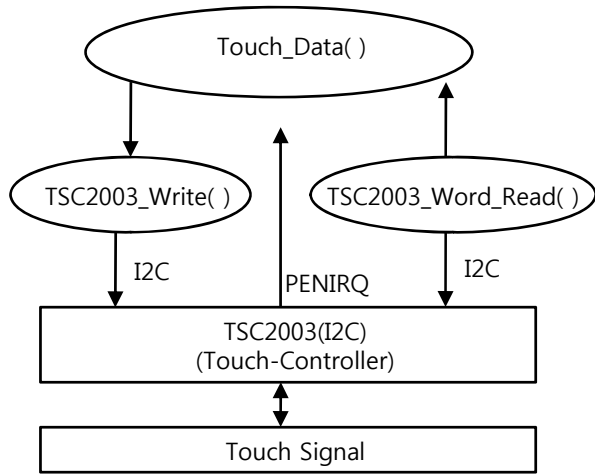
S/W 사양

1. 프로그램 구성



1. TEST.C : 메인 프로그램 및 터치 관리 프로그램
통신(19200,N8,1)으로 LCD 테스트 프로그램 관리
2. CPUSETUP.INC :
인터럽트 관리 및 CPU 초기화 루틴

2. TEXT-LCD 라이브러리 구성

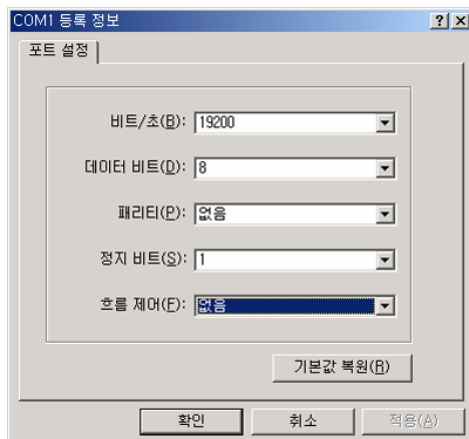
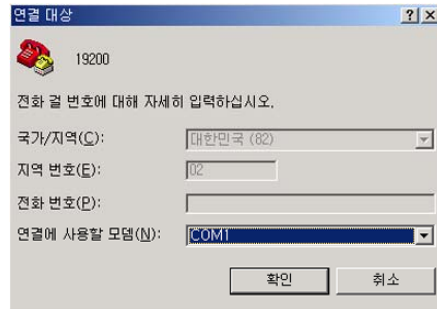
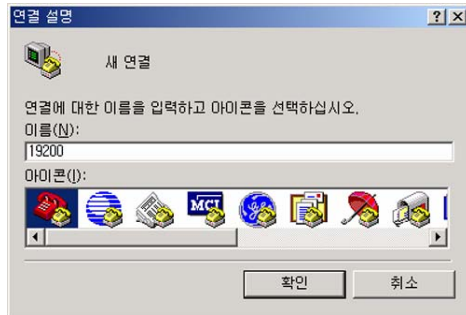


3. PC 통신 프로그램

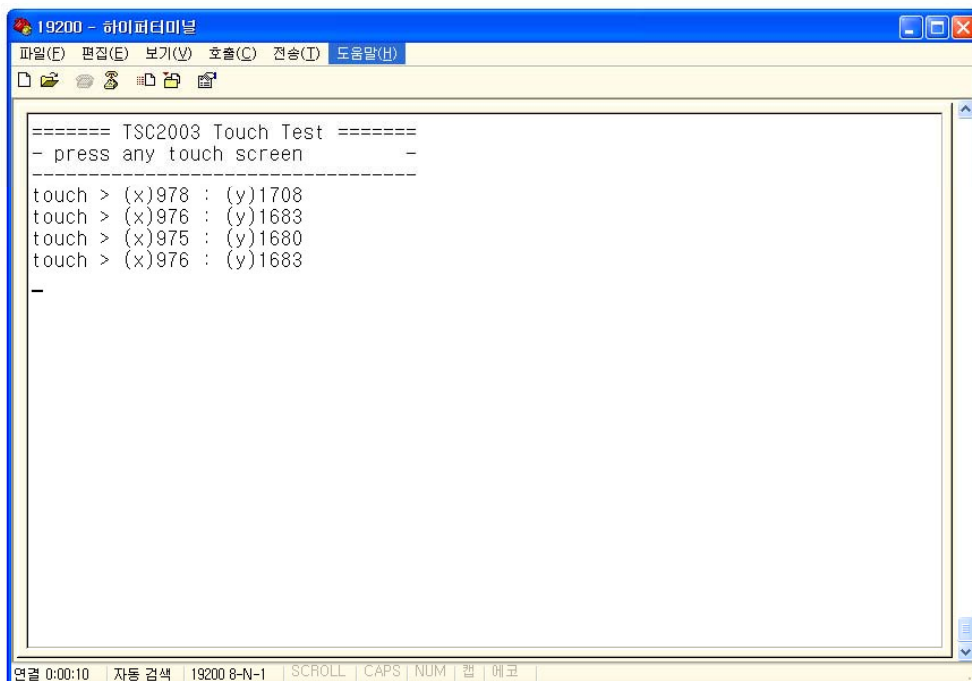
1. 하이퍼 터미널을 시작 합니다.



2. 하이퍼 터미널에서 통신 포트를 설정합니다.



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



Touch 라이브러리(test.c)

Touch_Data

터치 입력 정보를 취득(x,y)

함 수 : int Touch_Data(int scan_time)
인 수 : int scan_time // 터치를 누를때부터 정보 취득 타임(0 - 1000)
scan_time * 10ms
반 환 : 1(데이터 있음),0(데이터 없음)

설 명 : 터치 컨트롤러(TSC2003)으로 부터 유효 정보를 정해진 시간마다 취득한다.
변 수 : Touch_X // 터치 X축 정보가 저장되는 변수(0-4095)
Touch_Y // 터치 Y축 정보가 저장되는 변수(0-4095)

[예제]

.
C기본 헤더 선언
.

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램
```

```
// 함수 선언  
void main_menu_disp(void); // 메인 메뉴 표시  
int Touch_Data(int scan_time); // 터치 데이터 관리  
void TSC2003_Setup(void); // I2c 초기화및 TSC2003초기화  
void TSC2003_Write(unsigned char byte); // TSC2003 터치 Write  
unsigned TSC2003_Word_Read(unsigned char cmd); // TSC2003 터치 Word Read
```

```
int Touch_X,Touch_Y; // 터치 사용 변수
```

```
void main(void)
```

```
{  
    Cpu_Setup(); // CPU 초기화  
    TSC2003_Setup(); // I2c 초기화및 TSC2003초기화  
  
    // Touch TEST Code..  
    while(1)  
    {  
        if(Touch_Data(2)){ // 터치 데이터 검사 20ms마다  
            printf("touch > (x)%d : (y)%d\r\n",Touch_X,Touch_Y); // COM1으로 표시  
        }  
    }  
}
```

.
터치 라이브러리 소스 파일
.

TSC2003_Setup

터치 컨트롤러와 CPU의 I2C를 초기화 한다.

함 수 : void TSC2003_Setup(void)

인 수 : void

반 환 : void

설 명 : 터치 컨트롤러와 CPU의 I2C를 초기화 한다.

[예제]

C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램
```

```
// 함수 선언
void main_menu_disp(void); // 메인 메뉴 표시
int Touch_Data(int scan_time); // 터치 데이터 관리
void TSC2003_Setup(void); // I2c 초기화및 TSC2003초기화
void TSC2003_Write(unsigned char byte); // TSC2003 터치 Write
unsigned TSC2003_Word_Read(unsigned char cmd); // TSC2003 터치 Word Read
```

```
int Touch_X,Touch_Y; // 터치 사용 변수
```

```
void main(void)
{
    Cpu_Setup(); // CPU 초기화
    TSC2003_Setup(); // I2C 초기화및 TSC2003초기화
    while(1)
    {
        사용자 프로그램
    }
}
```

터치 라이브러리 소스 파일

TSC2003_Write

터치 컨트롤러로 데이터를 전송 한다.

함 수 : TSC2003_Write(unsigned char datau)

인 수 : unsigned char datau

// TSC2003 으로 전송할 데이터

반 환 : void

설 명 터치 컨트롤러로 I2C를 통해 데이터를 전송 한다.

TSC2003_Write

터치 컨트롤러로 데이터를 전송 한다.

함 수 unsigned TSC2003_Word_Read(unsigned char cmd))

인 수 : unsigned char cmd // TSC2003 으로 전송할 Command 데이터

반 환 : 터치 좌표 데이터(12bit)

설 명 터치 컨트롤러로 I2C를 통해 x,y 데이터 요구를 보낸뒤 좌표값(12bit)을 리턴 한다.