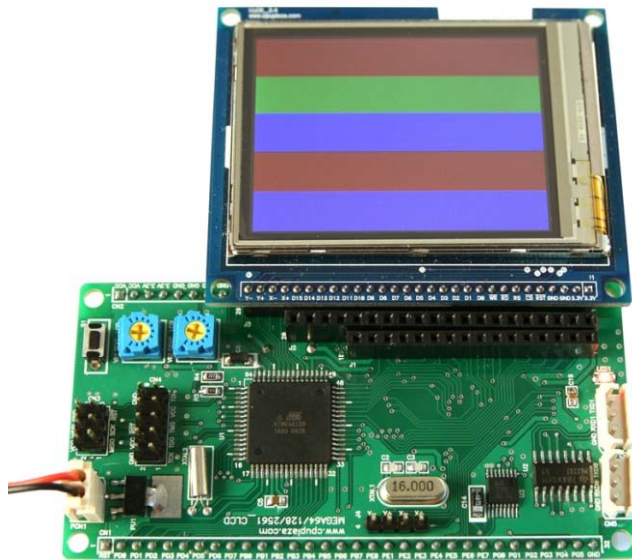


# MEGA128\_CLCD

(Color TFT-LCD 2.8" 메뉴얼)



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

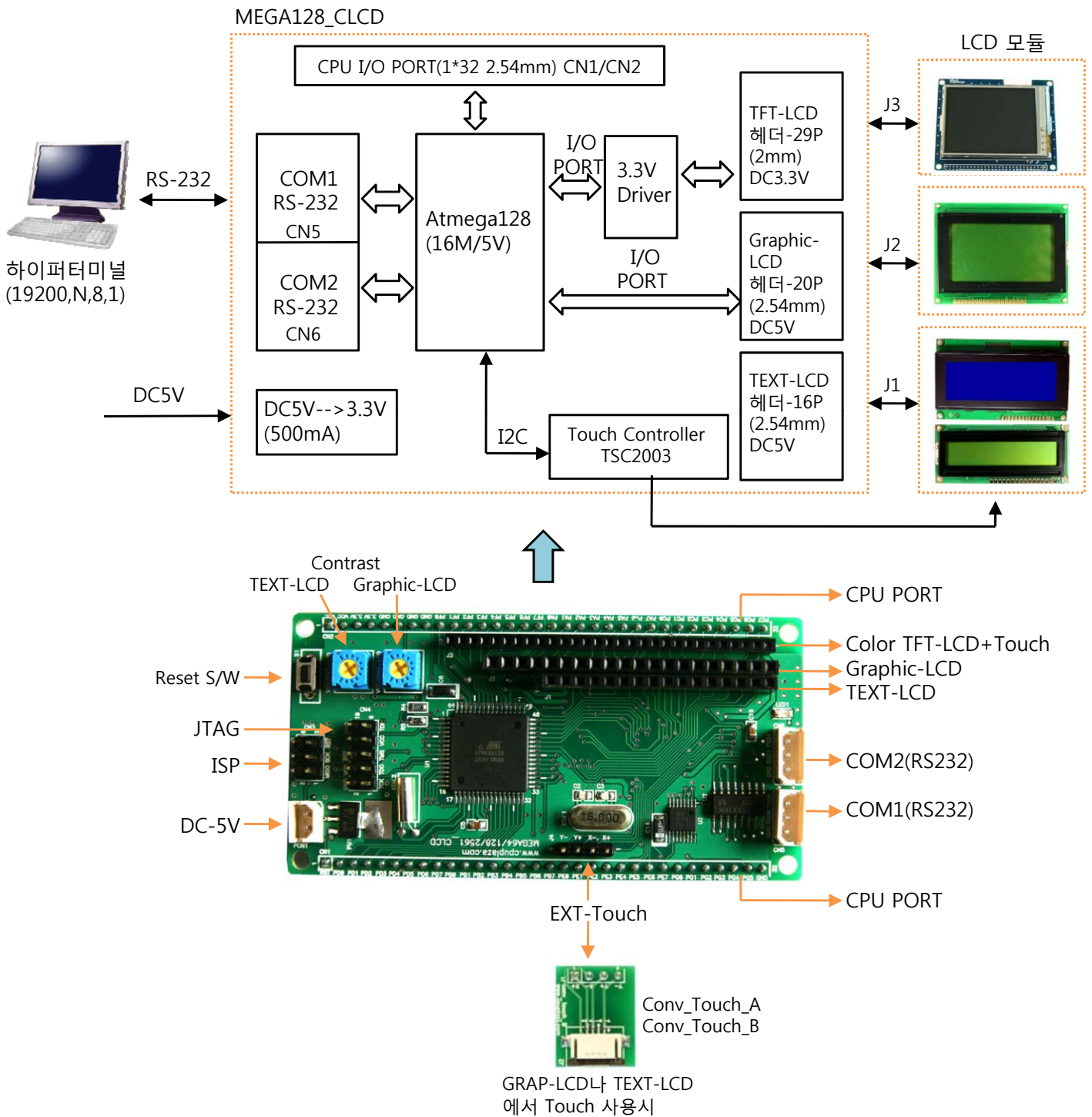
- 전원 : 외부 DC 5.0V
- TEXT LCD : 16\*2 / 16\*4 / 20\*4 지원
  - I/O 방식 제어
- Graphic-LCD(흑백) : 128\*64 지원
  - I/O 방식 제어
- Color(64K) TFT-LCD 지원 : 320\*240 지원
  - 80-System 16Bit I/O 방식 제어
- 압전식 터치 지원 – TSC2003 Controller
- Atmega128 CPU 모든 I/O 1열 헤더 지원
- RS232 2PORT 포트 별도 지원
- 완성형 한글 지원(완성형 -> 조합형 코드 변환 사용)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

\* Color TFT-LCD 라이브러리 함수

. clrscr ( ) / Cleardevice ( )	// 화면 소거
. Gotoxy ( ) / Gotoxyt ( )	// 커저 좌표 지정
. Setcolor ( ) / Setbkcolor ( )	// 컬러 지정
. ScreenSet ( );	// 화면 Windows 지정
. Putch ( ) / Putchxy ( ) / Putchxyt ( )	// 한 문자 표시
. Puts ( ) / Putsxy ( ) / Putsxyt ( )	// 문자열 표시
. Line ( ) / Rectangle ( ) / Rectanglefill ( )	// 라인 및 박스 그림
. Circle ( ) / Ellipse ( )	// 원, 타원 지원
. Putpixelb ( ) / PutImage ( ) / Putpixel ( )	// 픽셀 및 이미지 지원
. CLCD_GRAM_dread ( )	// 픽셀 정보 취득
. Han_Conv ( )	// 완성형 조합형 변환
. RGB_Disp ( )	// RGB 패턴 표시

# 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

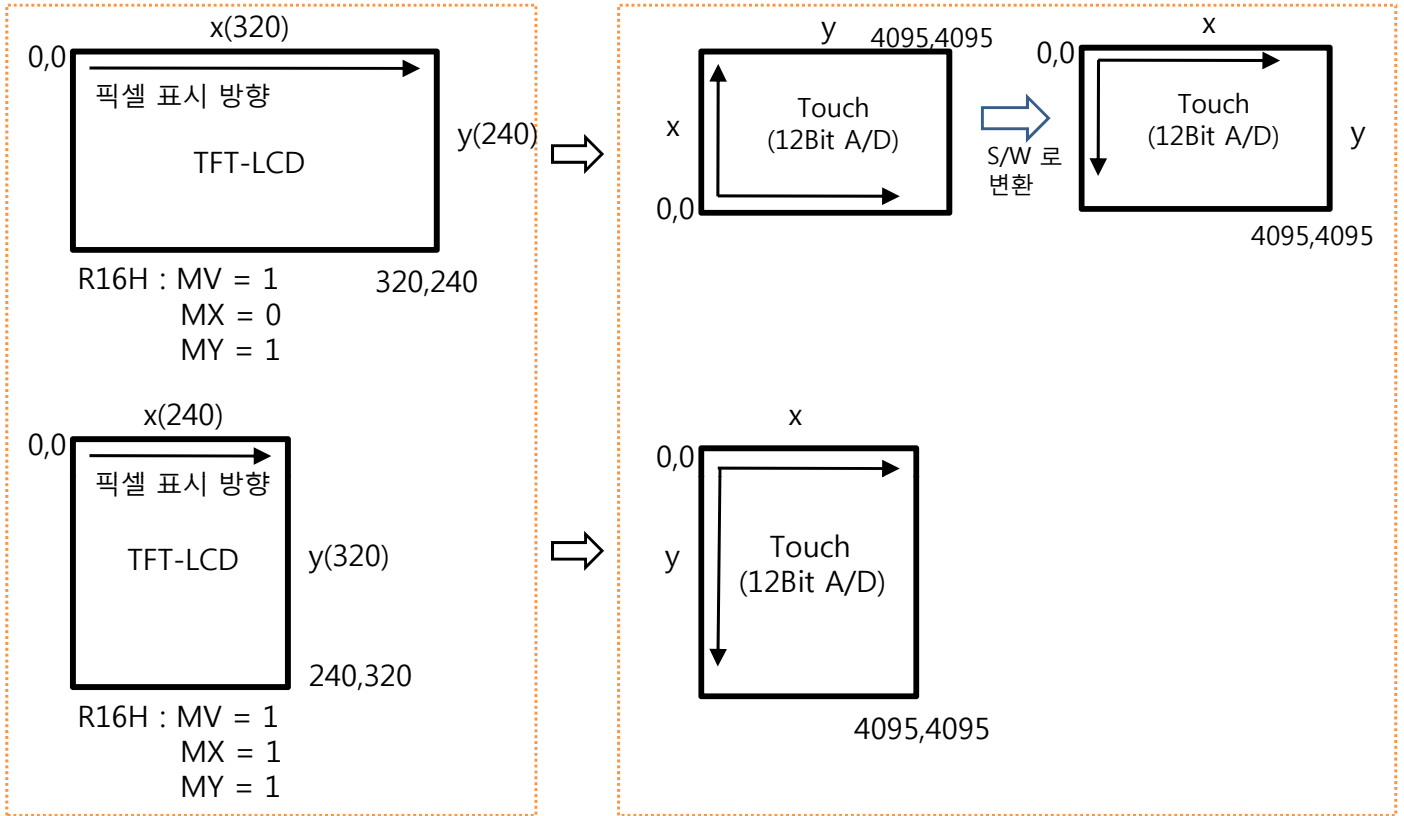
NUM	NAME
1	GND
2	ERXD1
3	ETXD1

### 3. LCD 설명

- TFT-LCD는 화면의 표시방법(가로/세로 방향)을 Driver IC의 설정에 따라 변경 할수 있습니다. 또한 화면의 모드에 따라 Touch Data 취득 방법도 변경 되어야 합니다.  
 - Screen\_Set() : 그래픽 지원 라이브러리 함수 지원

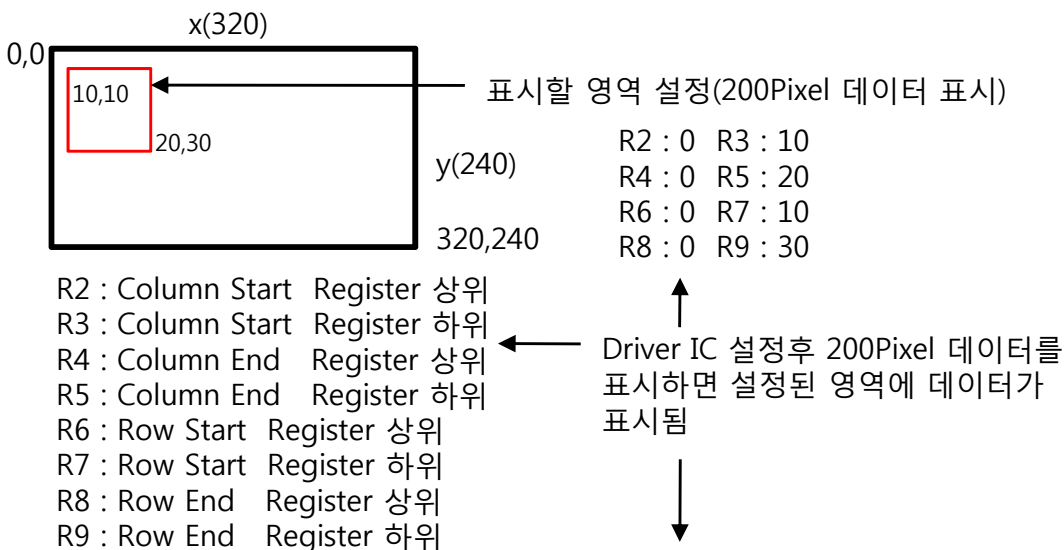
- TFT-LCD

- Touch



- TFT-LCD 화면에 데이터를 표시할때 Driver IC에 Windows를 설정하고 설정된 영역에 설정한 갯수만큼 픽셀 데이터(16Bit RGB)를 전송 하면 됩니다.

- TFT-LCD(10,10,20,30) 설정시

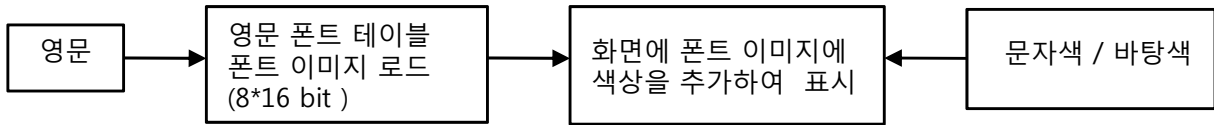


- TFT-LCD 화면에 표시되는 데이터 픽셀의 색상의 아래와 같습니다.



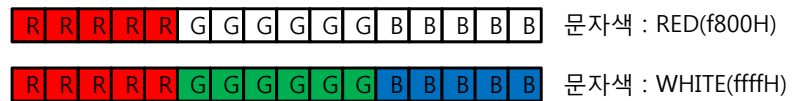
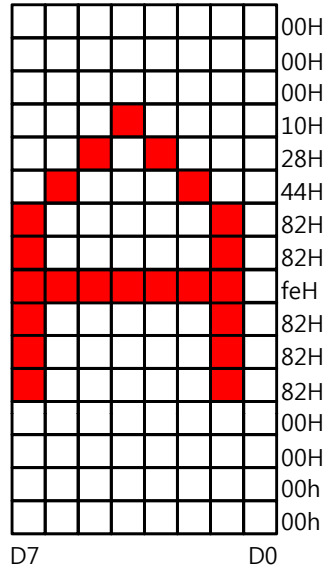
## 4. LCD 문자 표시

1. TFT-LCD 화면에 영문자 'A'를 문자색 적색 바탕색 흰색으로 표현 할 경우 예제 입니다.

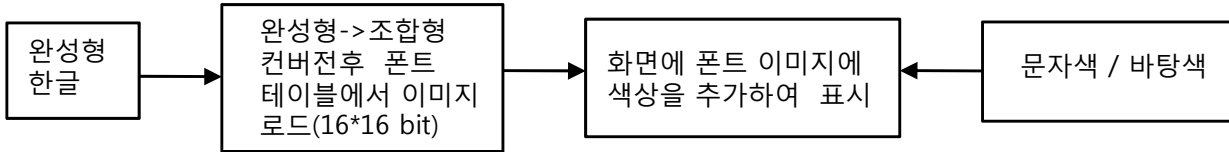


'A'  
41H

NUM	DATA
0	00H
1	00H
2	00H
3	10H
4	28H
5	44H
6	82H
7	82H
8	feH
9	82H
10	82H
11	82H
12	00H
13	00H
14	00H
15	00H

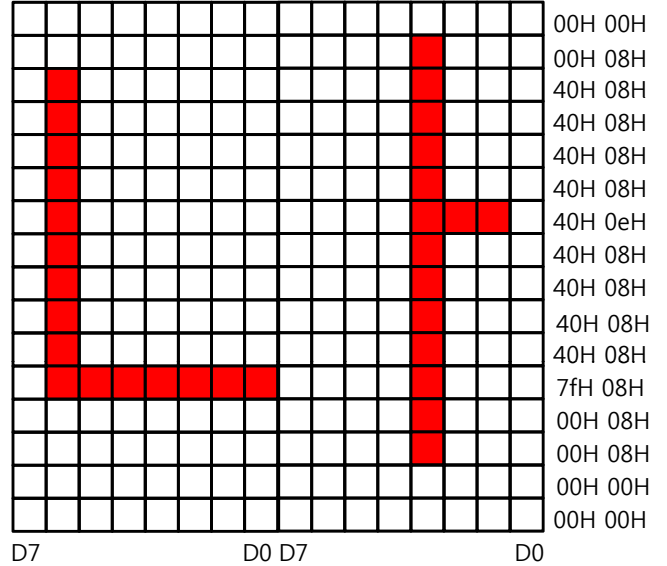


2. TFT-LCD 화면에 한글 "나"를 문자색 적색 바탕색 흰색으로 표현 할 경우 예제 입니다.



완성형  
"나"  
B3AAH  
↓  
조합형  
"나"  
9061H

NUM	DATA	NUM	DATA
0	00H	16	40H
1	00H	17	08H
2	00H	18	40H
3	08H	19	08H
4	40H	20	40H
5	08H	21	08H
6	40H	22	7fH
7	08H	23	08H
8	40H	24	00H
9	08H	25	08H
10	40H	26	00H
11	08H	27	08H
12	40H	28	00H
13	0eH	29	00H
14	40H	30	00H
15	08H	31	00H



폰트 이미지 32Byte

문자색 : RED(f800H)



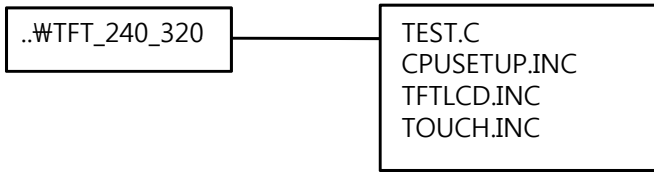
문자색 : WHITE(ffffH)





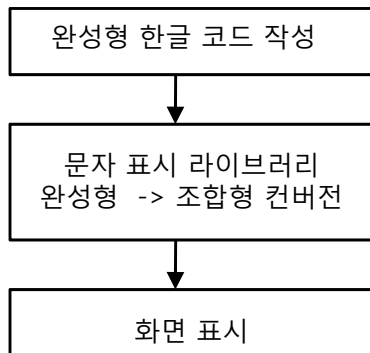
# S/W 사양

## 1. 프로그램 구성



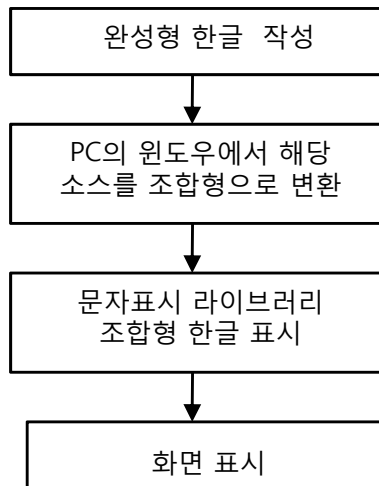
1. TEST.C : 메인 프로그램  
통신(19200,N8,1)으로 LCD 테스트 프로그램 관리
2. CPUSETUP.INC :  
인터럽트 관리 및 CPU 초기화 루틴
3. TFTLCD.INC :  
TFT-LCD 관리 및 그래픽 라이브러리 소스
3. TOUCH.INC :  
TOUCH 라이브러리 소스

### \* 완성형 코드 지원시



(당사 제품)

### \* 조합형 코드 지원시



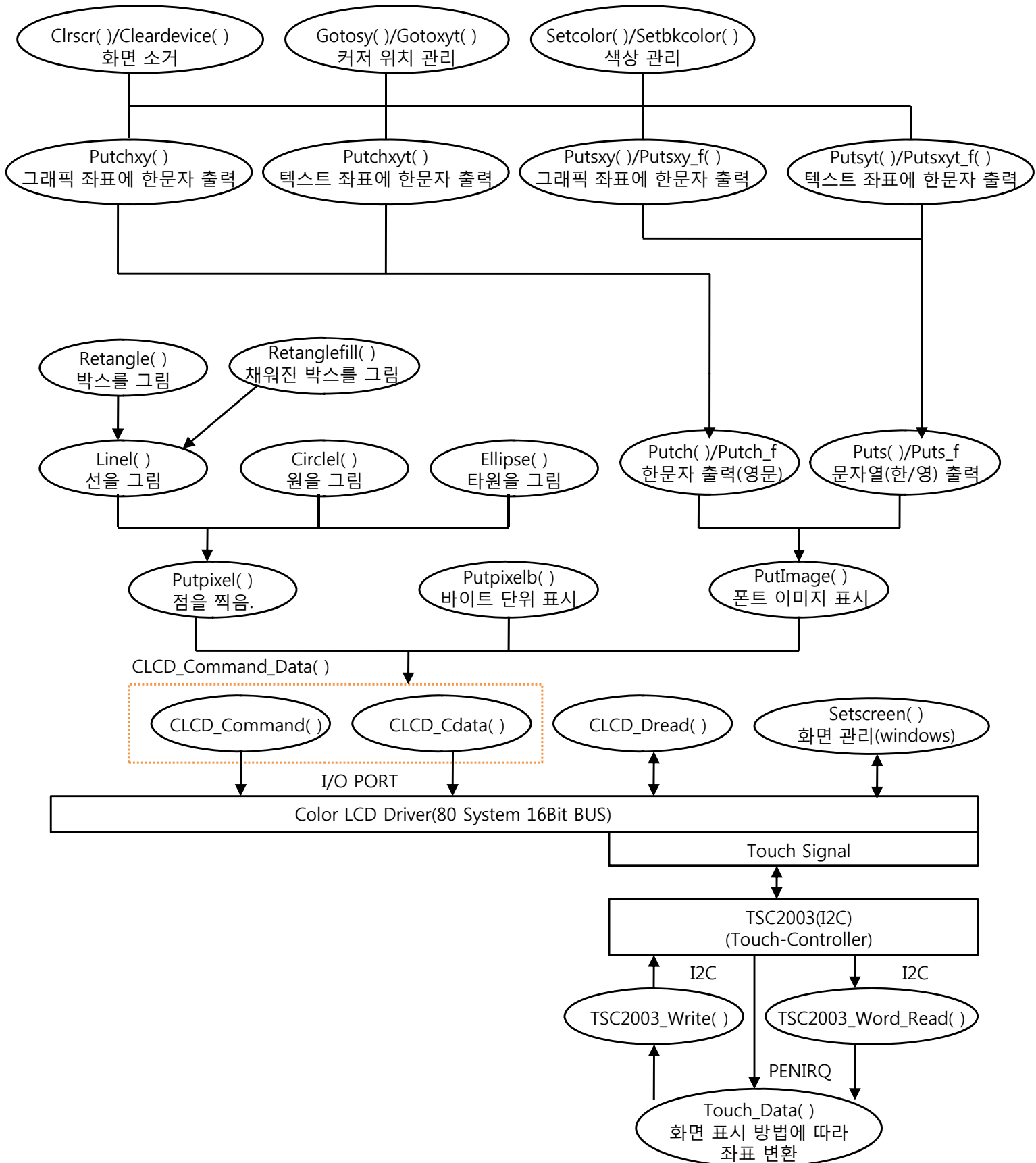
\*\* Windows Program \*\*

. HanConv.exe

. 산(DOS용)

\* 소스 코드명이 달라야 함

## 2. TFT-LCD 라이브러리 구성

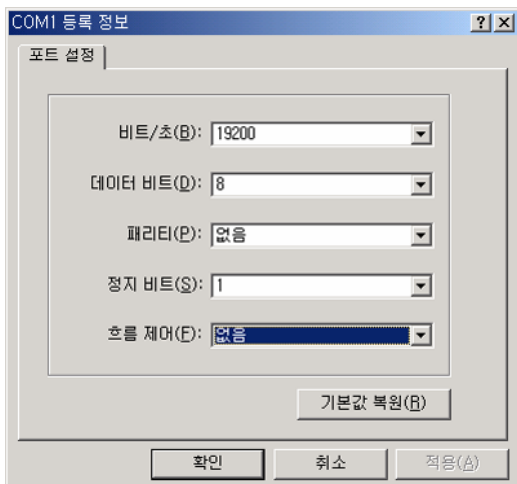
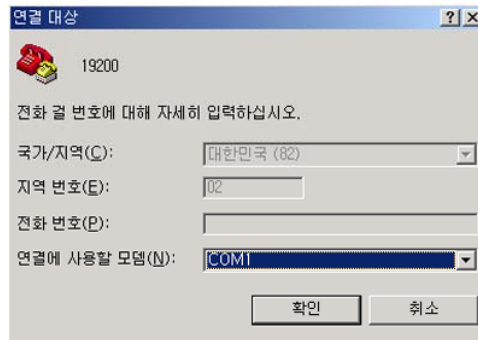
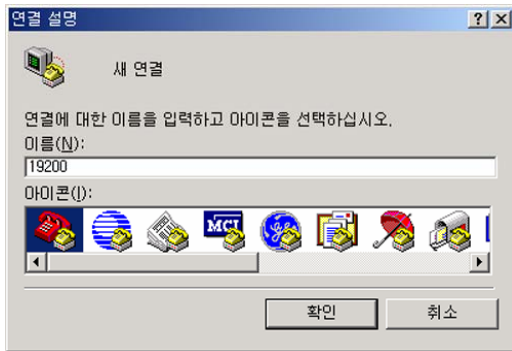


### 3. PC 통신 프로그램

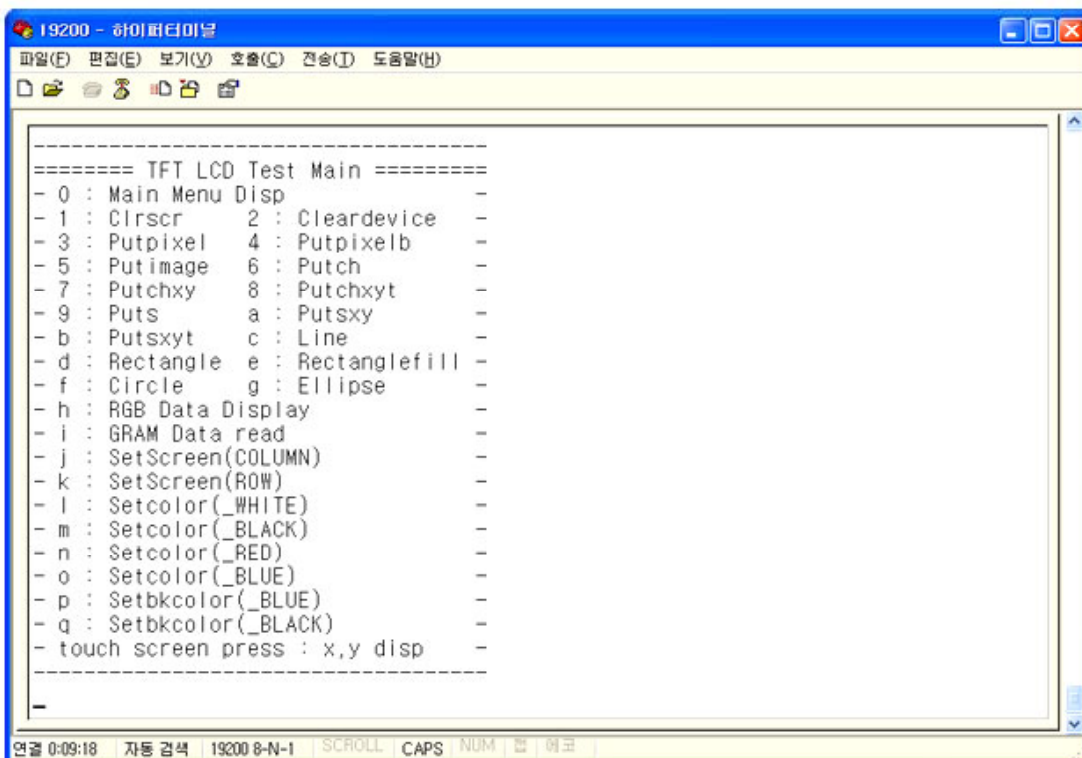
1. 하이퍼 터미널을 시작 합니다.



2. 하이퍼 터미널에서 통신 포트를 설정합니다.



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



# LCD 라이브러리(tftlcd.inc)

## Clrscr

LCD 화면을 모두 지움.

함 수 : void Clrscr(void)

인 수 : 없음.

반 환 : 없음.

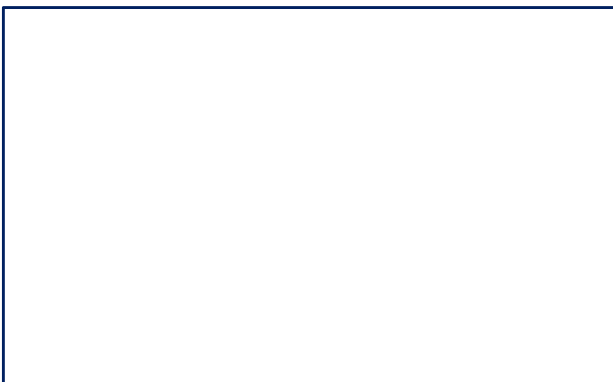
설 명 : 현재 화면을 모두 지운다.

[예제]

.  
C기본 헤더 선언

```
.  
#include "cpusetup.inc"           // CPU 초기화 및 인터럽트 관리 프로그램  
#include "tftlcd.inc"           // 그래픽 TFT-LCD 라이브러리  
  
void main(void)  
{  
    Cpu_Setup();                 // CPU 초기화  
    Clcd_Init();                 // 그래픽 TFT-LCD 초기화  
  
    // TFT-LCD TEST Code..  
    Clrscr();                     // TFT-LCD 화면 소거.  
    while(1)  
    {  
        사용자 프로그램  
    }  
}
```

\* 실행 결과(LCD 화면)



## Cleardevice

LCD 화면을 모두 지움.

함 수 : void Cleardevice(void)

인 수 : 없음.

반 환 : 없음.

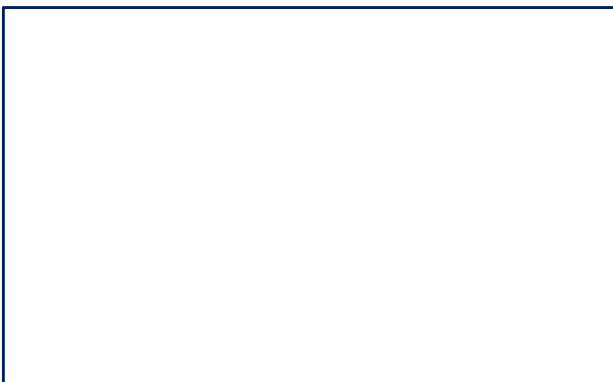
설 명 : 현재 화면을 모두 지운다.

[예제]

.  
C기본 헤더 선언

```
.  
#include "cpusetup.inc"           // CPU 초기화 및 인터럽트 관리 프로그램  
#include "tftlcd.inc"           // 그래픽 TFT-LCD 라이브러리  
  
void main(void)  
{  
    Cpu_Setup();                // CPU 초기화  
    Clcd_Init();                // 그래픽 TFT-LCD 초기화  
  
    // TFT-LCD TEST Code..  
    Cleardevice();              // TFT-LCD 화면 소거.  
    while(1)  
    {  
        사용자 프로그램  
    }  
}
```

\* 실행 결과(LCD 화면)



## Gotoxy

그래픽 모드 형태로 커저 좌표 를 지정.

```
함 수 : void Gotoxy(int x,int y)
인 수 : int x                // 커저의 x(가로) 좌표(0 - 319)
      int y                // 커저의 y(세로) 좌표(0 - 239)
반 환 : 없음.
변 수 : int _screen_curx   // X좌표 저장
      int _screen_cury    // y좌표 저장
```

설 명 : 그래픽 모드 형태로 커저 위치 지정.

문자를 표현하는 함수에서 사용.

\* 320 \* 240 그래픽 TFT-LCD 일때

x = 320 - 1

y = 240 - 1

관련 함수 : Putch,Puts,Puts\_f,Setcolor

[예제]

C기본 헤더 선언

```
.
#include "cpusetup.inc"      // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc"       // 그래픽 TFT-LCD 라이브러리

void main(void)
{
    Cpu_Setup();           // CPU 초기화
    Clcd_Init();          // 그래픽 TFT-LCD 초기화

// TFT-LCD TEST Code..
    Gotoxy(1,2);          // TFT-LCD 커저 위치 지정.
    while(1)
    {
        사용자 프로그램
    }
}
```

## Gotoxyt

텍스트 모드 형태로 커저 좌표 를 지정.

```
함 수 : void Gotoxyt(int x,int y)
인 수 : int x                // 커저의 x(가로) 좌표(1 - 40)
      int y                // 커저의 y(세로) 좌표(1 - 15)
반 환 : 없음.
변 수 : int _screen_curx    // X좌표 저장 변수
      int _screen_cury    // y좌표 저장 변수
```

설 명 : 텍스트 모드 형태로 커저 위치 지정.

문자를 표현하는 함수에서 사용.

\* 320 \* 240 그래픽 TFT-LCD 일때

x :  $320 / 8 = 40$

y :  $240 / 16 = 15$

관련 함수 : Putch,Puts,Puts\_f,Setcolor

[예제]

C기본 헤더 선언

```
.
.
#include "cpusetup.inc"      // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc"      // 그래픽 TFT-LCD 라이브러리

void main(void)
{
    Cpu_Setup();           // CPU 초기화
    Clcd_Init();          // 그래픽 TFT-LCD 초기화

// TFT-LCD TEST Code..
    Gotoxyt(1,2);         // TFT-LCD 커저 위치 지정.
    while(1)
    {
        사용자 프로그램
    }
}
```

## Setcolor

문자색을 지정 한다.

함 수 : void Setcolor(unsigned color)

인 수 : unsigned color

```
_BLACK // 표시할 문자색
        // 검정
    _RED // 적색
    _BLUE // 파랑
    _GREEN // 녹색
    _MAGENTA // 자주색(붉은)
    _GRAY // 회색
    _CYAN // 청녹색
    _YELLOW // 노랑
```

\* 사용자가 지정후 사용 가능.

반 환 : 없음.

설 명 : TFT-LCD에 표시되는 문자색을 지정 한다.

변 수 : unsigned \_screen\_color // 컬러값 저장 변수

관련 함수 : Putch,Puchxy,Puchxyt,Puts,Putsxy,Putsxyt,Puts\_f,Putsxy\_f,Putsxyt\_f  
Line,Rectangle,Retanglefill,Cicle,Ellipse

[예제]

.  
C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc" // 그래픽 TFT-LCD 라이브러리
```

```
void main(void)
{
    Cpu_Setup(); // CPU 초기화
    Clcd_Init(); // 그래픽 TFT-LCD 초기화
```

```
// TFT-LCD TEST Code..
Setcolor(_WHITE); // 흰색 지정
Putch('1'); // 화면에 '1' 표시
while(1)
{
    사용자 프로그램
}
}
```

\* 실행 결과(LCD 화면)



1



## Setbcolor

바탕색을 지정 한다.

함 수 : void Setbcolor(unsigned color)

인 수 : unsigned color // 표시할 바탕색  
\_BLACK // 검정  
\_RED // 적색  
\_BLUE // 파랑  
\_GREEN // 녹색  
\_MAGENTA // 자주색(붉은)  
\_GRAY // 회색  
\_CYAN // 청녹색  
\_YELLOW // 노랑

\* 사용자가 지정후 사용 가능.

반 환 : 없음.

설 명 : TFT-LCD에 표시되는 바탕색을 지정 한다.

변 수 : unsigned \_screen\_bcolor // 바탕색 저장 변수

관련 함수 : Putch,Puchxy,Puchxyt,Puts,Putsxy,Putsxyt,Puts\_f,Putsxy\_f,Putsxyt\_f  
Line,Rectangle,Retanglefill,Cicle,Ellipse

[예제]

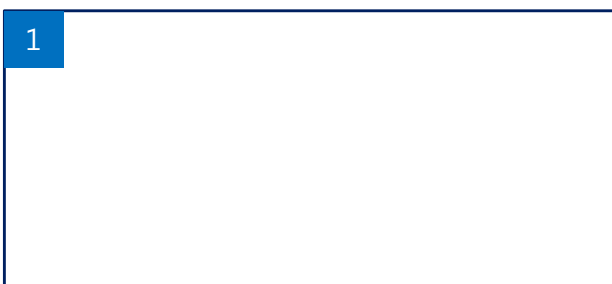
.  
C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램  
#include "tftlcd.inc" // 그래픽 TFT-LCD 라이브러리
```

```
void main(void)  
{  
    Cpu_Setup(); // CPU 초기화  
    Clcd_Init(); // 그래픽 TFT-LCD 초기화
```

```
// TFT-LCD TEST Code..  
Setbcolor(_BLUE); // 파랑색 지정  
Putch('1'); // 화면에 '1' 표시  
while(1)  
{  
    사용자 프로그램  
}
```

\* 실행 결과(LCD 화면)



# Setscreen

화면 표시 방법 및 Window를 설정 한다.

함 수 : void Setscreen(int mode,int x1,int y1,int x2,int y2)

```
인 수 : int mode           // 화면 표시 방법 설정
        _SCREEN_COLUMN_SET // 가로 화면 모드 설정(320 * 240)
        _SCREEN_ROW_SET    // 세로 화면 모드 설정(240 * 320)

        int x1             // 시작점의 x좌표 : 가로면 표시시(0-319) / 세로면 표시시(0-239)
        int y1             // 시작점의 y좌표 : 가로면 표시시(0-239) / 세로면 표시시(0-319)
        int x2             // 끝점의 x좌표 : 가로면 표시시(0-319) / 세로면 표시시(0-239)
        int y2             // 끝점의 y좌표 : 가로면 표시시(0-319) / 세로면 표시시(0-319)
```

설 명 : TFT-LCD 화면 모드 및 화면에 표시될 영역을 설정 한다.

```
변 수 : int _screen_mode // 화면 설정 모드값
        int _screen_xmax // x축 최대 설정값(x2)
        int _screen_ymax // y축 최대 설정값(y2)
```

관련 함수 : 전체 그래픽 관련 함수

[예제]

C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc"   // 그래픽 TFT-LCD 라이브러리
```

void main(void)

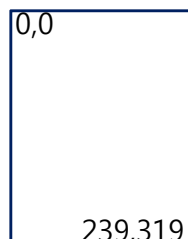
```
{
    Cpu_Setup();           // CPU 초기화
    Clcd_Init();          // 그래픽 TFT-LCD 초기화

    // TFT-LCD TEST Code..
    // _SCREEN_COLUMN_X 319,_SCREEN_COLUMN_Y 239
    // _SCREEN_ROW_X 239,_SCREEN_ROW_Y 319
    // 아래 소스 둘중 하나만 코딩.
    // 가로 화면 최대모드(0,0,319,239)
    ScreenSet(_SCREEN_COLUMN_SET,0,0,_SCREEN_COLUMN_X,_SCREEN_COLUMN_Y);
    // 세로 화면 최대모드(0,0,239,319)
    ScreenSet(_SCREEN_ROW_SET,0,0,_SCREEN_ROW_X,_SCREEN_ROW_Y)
    while(1)
    {
        사용자 프로그램
    }
}
```

\* 실행 결과(LCD 화면)



가로 화면 최대 표시



세로 화면 최대 표시

## Putch

현재 커저 위치에 문자를 표시한다.

함 수 : void Putch(unsigned c)  
인 수 : unsigned c // 표시할 문자  
반 환 : 없음.

설 명 : 현재 커저 위치에 문자를 표시 한다.  
( 한글 코드 제외)

관련 함수 : Puchxy,Puchxyt,Setcolor,Setbcolor

[예제]

.  
C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램  
#include "tftlcd.inc" // 그래픽 TFT-LCD 라이브러리
```

```
void main(void)  
{  
    Cpu_Setup(); // CPU 초기화  
    Clcd_Init(); // 그래픽 TFT-LCD 초기화
```

```
// TFT-LCD TEST Code..  
Putch('2'); // 화면에 '2' 표시  
while(1)  
{  
    사용자 프로그램  
}  
}
```

\* 실행 결과(LCD 화면)



2

## Putchxy

지정한 그래픽 좌표에 문자를 표시한다.

함 수 : void Putchxy(int x, int y,unsigned c)  
인 수 : int x // 커저의 x(가로) 좌표(0 - 319 / 0 - 239)  
int y // 커저의 y(세로) 좌표(0 - 239 / 0 - 319)  
unsigned c // 표시할 문자  
반 환 : 없음.

설 명 : 지정한 그래픽 좌표에 문자를 표시 한다.  
(한글 코드 제외)

가로 화면 모드시 : 320 \* 240 픽셀

- x = 320 - 1, y = 240 - 1

세로 화면 모드시 : 240 \* 320 픽셀

- x = 320 - 1, y = 240 - 1

관련 함수 : Setcolor,Setbcolor,Gotoxy,Gotoxyt

[예제]

C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc" // 그래픽 TFT-LCD 라이브러리

void main(void)
{
    Cpu_Setup(); // CPU 초기화
    Clcd_Init(); // 그래픽 TFT-LCD 초기화

    // TFT-LCD TEST Code..
    Putchxy(1,10,'2'); // 화면의 1(x),10(y)위치에 '2' 표시
    while(1)
    {
        사용자 프로그램
    }
}
```

\* 실행 결과(LCD 화면)



2

## Putchxyt

지정한 텍스트 좌표에 문자를 표시한다.

함 수 : void Putchxyt(int x, int y,unsigned c)  
인 수 : int x // 커저의 x(가로) 좌표(1 - 40 / 1 - 15)  
int y // 커저의 y(세로) 좌표(1 - 15 / 1 - 40)  
int unsigned c // 표시할 문자  
반 환 : 없음.

설 명 : 지정한 텍스트 좌표에 문자를 표시 한다.  
(한글 코드 제외)

가로 화면 모드시 : 320 \* 240 픽셀

- x : 320 / 8 = 40, y : 240 / 16 = 15

세로 화면 모드시 : 240 \* 320 픽셀

- x : 320 / 8 = 40, y : 240 / 16 = 15

관련 함수 : Setcolor,Setbcolor

[예제]

C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc" // 그래픽 TFT-LCD 라이브러리

void main(void)
{
    Cpu_Setup(); // CPU 초기화
    Clcd_Init(); // 그래픽 TFT-LCD 초기화

    // TFT-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(const char \*str)  
인 수 : const char \*str // 표시할 문자열  
반 환 : 없음.

설 명 : 현재 커저 위치에 문자열을 표시 한다.  
(완성형 한글 코드 포함)

관련 함수 : Putsxy, Putsxyt, Setcolor, Setbcolor, Gotoxy, Gotoxyt

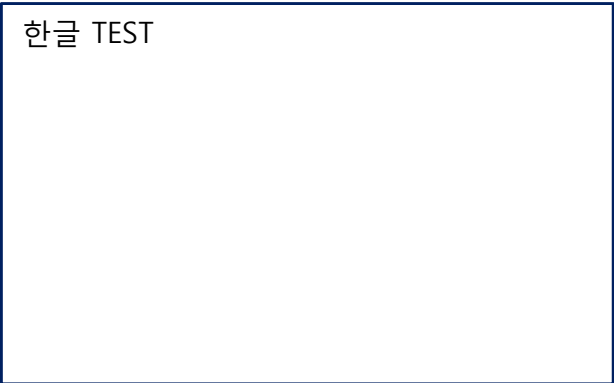
[예제]

.  
C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램  
#include "tftlcd.inc" // 그래픽 TFT-LCD 라이브러리
```

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

\* 실행 결과(LCD 화면)



한글 TEST

## Putsxy

지정한 그래픽 좌표에 문자열을 표시한다.

함 수 : void Putsxy(int x, int y,const char \*str)  
인 수 : int x // 커저의 x(가로) 좌표(0 - 319 / 0 - 239)  
int y // 커저의 y(세로) 좌표(0 - 239 / 0 - 319)  
const char \*str // 표시할 문자열  
반 환 : 없음.

설 명 : 지정한 그래픽 좌표에 문자열을 표시 한다.

(완성형 한글 코드 포함)

가로 화면 모드시 : 320 \* 240 픽셀

- x = 320 - 1, y = 240 - 1

세로 화면 모드시 : 240 \* 320 픽셀

- x = 320 - 1, y = 240 - 1

관련 함수 : Setcolor,Setbcolor

[예제]

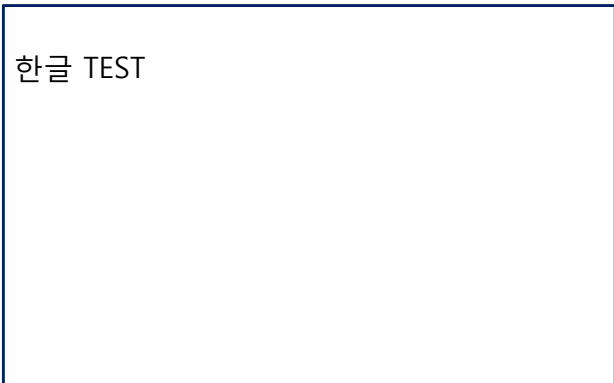
C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc" // 그래픽 TFT-LCD 라이브러리

void main(void)
{
    Cpu_Setup(); // CPU 초기화
    Clcd_Init(); // 그래픽 TFT-LCD 초기화

    // TFT-LCD TEST Code..
    Putsxy(1,10,"한글 TEST"); // 화면의 1(x),10(y)위치에 한글 TEST 표시
    while(1)
    {
        사용자 프로그램
    }
}
```

\* 실행 결과(LCD 화면)



한글 TEST

## Putsxyt

지정한 텍스트 좌표에 문자열을 표시한다.

함 수 : void Putsxyt(int x, int y, const char \*str)  
인 수 : int x // 커저의 x(가로) 좌표(1 - 40 / 1 - 15)  
int y // 커저의 y(세로) 좌표(1 - 15 / 1 - 40)  
const char \*str // 표시할 문자  
반 환 : 없음.

설 명 : 지정한 텍스트 좌표에 문자열을 표시 한다.  
(완성형 한글 코드 포함)

가로 화면 모드시 : 320 \* 240 픽셀

- x : 320 / 8 = 40, y : 240 / 16 = 15

세로 화면 모드시 : 240 \* 320 픽셀

- x : 320 / 8 = 40, y : 240 / 16 = 15

관련 함수 : Setcolor, Setbcolor

[예제]

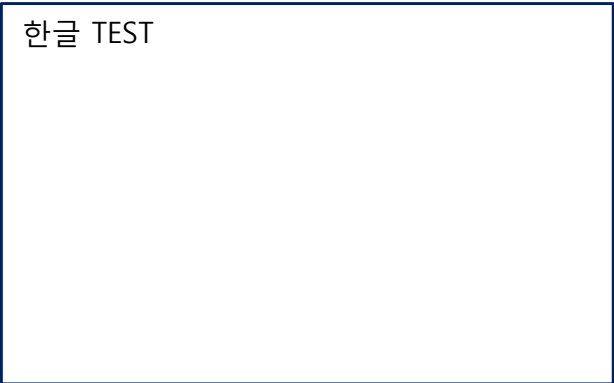
C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc" // 그래픽 TFT-LCD 라이브러리

void main(void)
{
    Cpu_Setup(); // CPU 초기화
    Clcd_Init(); // 그래픽 TFT-LCD 초기화

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

\* 실행 결과(LCD 화면)



한글 TEST



## Puts\_f

현재 커저 위치에 문자열(FLASH)을 표시한다. - CVAVR용

함 수 : void Puts\_f(const char flash \*str)  
인 수 : const char flash \*str // 표시할 문자열  
반 환 : 없음.

설 명 : 현재 커저 위치에 문자열(FLASH MEMORY 데이터) 을 표시 한다.  
(완성형 한글 코드 포함)

관련 함수 : Putsxy\_f, Putsxyt\_f, Setcolor, Setbcolor, Gotoxy, Gotoxyt

[예제]

.  
C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램  
#include "tftlcd.inc" // 그래픽 TFT-LCD 라이브러리
```

```
void main(void)  
{  
    Cpu_Setup(); // CPU 초기화  
    Clcd_Init(); // 그래픽 TFT-LCD 초기화  
  
    // TFT-LCD TEST Code..  
    Puts_f("한글 TEST"); // 화면에 한글 TEST 표시  
    while(1)  
    {  
        사용자 프로그램  
    }  
}
```

\* 실행 결과(LCD 화면)



한글 TEST

## Putsxy\_f

지정한 그래픽 좌표에 문자열(FLASH)을 표시한다. - CVAVR용

함 수 : void Putsxy\_f(int x, int y,const char flash \*str)  
인 수 : int x // 커저의 x(가로) 좌표(0 - 319 / 0 - 239)  
int y // 커저의 y(세로) 좌표(0 - 239 / 0 - 319)  
const char flash \*str // 표시할 문자열  
반 환 : 없음.

설 명 : 지정한 그래픽 좌표에 문자열(FLASH MEMORY 데이터) 을 표시 한다.

(완성형 한글 코드 포함)

가로 화면 모드시 : 320 \* 240 픽셀

- x = 320 - 1, y = 240 - 1

세로 화면 모드시 : 240 \* 320 픽셀

- x = 320 - 1, y = 240 - 1

관련 함수 : Setcolor,Setbcolor

[예제]

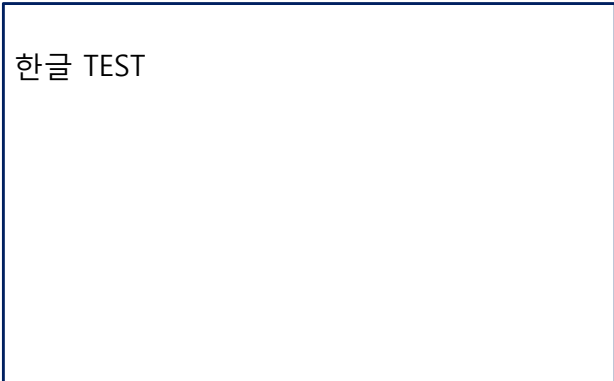
C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc" // 그래픽 TFT-LCD 라이브러리

void main(void)
{
    Cpu_Setup(); // CPU 초기화
    Clcd_Init(); // 그래픽 TFT-LCD 초기화

    // TFT-LCD TEST Code..
    Putsxy_f(1,10,"한글 TEST"); // 화면의 1(x),10(y)위치에 한글 TEST 표시
    while(1)
    {
        사용자 프로그램
    }
}
```

\* 실행 결과(LCD 화면)



한글 TEST

## Putsxyt\_f

지정한 텍스트 좌표에 문자열(FLASH)을 표시한다. - CVAVR용

함 수 : void Putsxyt\_f(int x, int y,const char flash \*str)  
인 수 : int x // 커저의 x(가로) 좌표(1 - 40 / 1 - 15)  
int y // 커저의 y(세로) 좌표(1 - 15 / 1 - 40)  
const char \*str // 표시할 문자  
반 환 : 없음.

설 명 : 지정한 텍스트 좌표에 문자열(FLASH MEMORY 데이터) 을 표시 한다.  
(완성형 한글 코드 포함)

가로 화면 모드시 : 320 \* 240 픽셀

- x : 320 / 8 = 40, y : 240 / 16 = 15

세로 화면 모드시 : 240 \* 320 픽셀

- x : 320 / 8 = 40, y : 240 / 16 = 15

관련 함수 : Setcolor,Setbcolor

[예제]

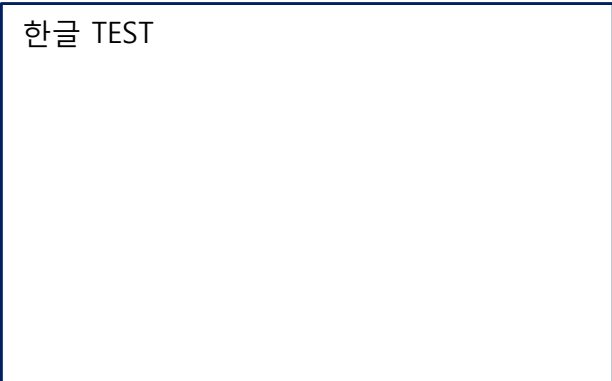
C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc" // 그래픽 TFT-LCD 라이브러리

void main(void)
{
    Cpu_Setup(); // CPU 초기화
    Clcd_Init(); // 그래픽 TFT-LCD 초기화

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

\* 실행 결과(LCD 화면)



한글 TEST

## Line

지정한 좌표에 선을 그림.

```
함 수 : void Line(int x1, int y1,int x2,int y2)
인 수 : int x1           // 시작점 x(가로) 좌표(0 - 319 / 0 - 239)
        int y1           // 시작점 y(세로) 좌표(0 - 239 / 0 - 319)
        int x2           // 끝점 x(가로) 좌표(0 - 319 / 0 - 239)
        int y2           // 끝점 y(세로) 좌표(0 - 239 / 0 - 319)
반 환 : 없음.
```

설 명 : 지정한 좌표에 선을 그림.  
가로 화면 모드시 : 320 \* 240 픽셀  
- x = 320 - 1, y = 240 - 1  
세로 화면 모드시 : 240 \* 320 픽셀  
- x = 320 - 1, y = 240 - 1  
관련 함수 : Setcolor

[예제]

C기본 헤더 선언

```
.
#include "cpusetup.inc"           // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc"            // 그래픽 TFT-LCD 라이브러리

void main(void)
{
    Cpu_Setup();                 // CPU 초기화
    Clcd_Init();                 // 그래픽 TFT-LCD 초기화

// TFT-LCD TEST Code..
    Line(0,0,127,0);            // 선을 그림
    while(1)
    {
        사용자 프로그램
    }
}
```

\* 실행 결과(LCD 화면)



## Retangle

지정한 좌표에 박스를 그림.

함 수 : void Retangle(int x1, int y1, int x2, int y2)  
인 수 : int x1 // 시작점 x(가로) 좌표(0 - 319 / 0 - 239)  
int y1 // 시작점 y(세로) 좌표(0 - 239 / 0 - 319)  
int x2 // 끝점 x(가로) 좌표(0 - 319 / 0 - 239)  
int y2 // 끝점 y(세로) 좌표(0 - 239 / 0 - 319)  
반 환 : 없음.

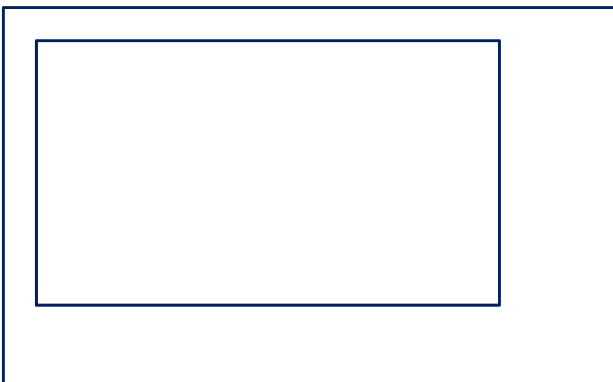
설 명 : 지정한 좌표에 박스를 그림.  
가로 화면 모드시 : 320 \* 240 픽셀  
- x = 320 - 1, y = 240 - 1  
세로 화면 모드시 : 240 \* 320 픽셀  
- x = 320 - 1, y = 240 - 1  
관련 함수 : Line,Setcolor

[예제]

C기본 헤더 선언

```
.  
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램  
#include "ftlcd.inc" // 그래픽 TFT-LCD 라이브러리  
  
void main(void)  
{  
    Cpu_Setup(); // CPU 초기화  
    Clcd_Init(); // 그래픽 TFT-LCD 초기화  
  
    // TFT-LCD TEST Code..  
    Retangle(10,10,100,50); // 박스를 그림  
    while(1)  
    {  
        사용자 프로그램  
    }  
}
```

\* 실행 결과(LCD 화면)



## Rectanglefill

지정한 좌표에 채워진 박스를 그림.

함 수 : void Rectangle(int x1, int y1, int x2, int y2)  
인 수 : int x1 // 시작점 x(가로) 좌표(0 - 319 / 0 - 239)  
 int y1 // 시작점 y(세로) 좌표(0 - 239 / 0 - 319)  
 int x2 // 끝점 x(가로) 좌표(0 - 319 / 0 - 239)  
 int y2 // 끝점 y(세로) 좌표(0 - 239 / 0 - 319)  
반 환 : 없음.

설 명 : 지정한 좌표에 채워진 박스를 그림.  
가로 화면 모드시 : 320 \* 240 픽셀  
- x = 320 - 1, y = 240 - 1  
세로 화면 모드시 : 240 \* 320 픽셀  
- x = 320 - 1, y = 240 - 1

관련 함수 : Line, Setcolor

[예제]

C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc" // 그래픽 TFT-LCD 라이브러리

void main(void)
{
    Cpu_Setup(); // CPU 초기화
    Clcd_Init(); // 그래픽 TFT-LCD 초기화

    // TFT-LCD TEST Code..
    Rectanglefill(10,10,100,50); // 채워진 박스를 그림
    while(1)
    {
        사용자 프로그램
    }
}
```

\* 실행 결과(LCD 화면)



## Circle

원을 그림.

함 수 : void Circle(int xx, int yy, radius)

인 수 : int xx

int yy

int radius

// 중심점 x(가로) 좌표(0 - 319 / 0 - 239)

// 중심점 y(세로) 좌표(0 - 239 / 0 - 319)

// 반지름 좌표(0 - 319 / 0 - 239)

반 환 : 없음.

설 명 : 지정한 좌표에 원을 그림.

가로 화면 모드시 : 320 \* 240 픽셀

- x = 320 - 1, y = 240 - 1

세로 화면 모드시 : 240 \* 320 픽셀

- x = 320 - 1, y = 240 - 1

관련 함수 : Setcolor

[예제]

.  
C기본 헤더 선언

#include "cpusetup.inc"

// CPU 초기화 및 인터럽트 관리 프로그램

#include "tftlcd.inc"

// 그래픽 TFT-LCD 라이브러리

void main(void)

{

Cpu\_Setup();

// CPU 초기화

Clcd\_Init();

// 그래픽 TFT-LCD 초기화

// TFT-LCD TEST Code..

Circle(64,32,10);

// 원을 그림

while(1)

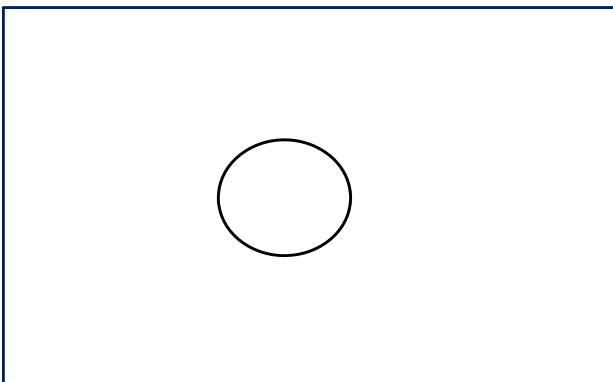
{

사용자 프로그램

}

}

\* 실행 결과(LCD 화면)



## Ellipse

타원을 그림.

함 수 : void Ellipse(int xx, int yy, int a0,int b0)

인 수 : int xx

int yy

int a0

int b0

// 중심점 x(가로) 좌표(0 - 319 / 0 - 239)

// 중심점 y(세로) 좌표(0 - 239 / 0 - 319)

// 가로 반경(0 - 319 / 0 - 239)

// 세로 반경(0 - 239 / 0 - 319)

반 환 : 없음.

설 명 : 지정한 좌표에 타원을 그림.

가로 화면 모드시 : 320 \* 240 픽셀

- x = 320 - 1, y = 240 - 1

세로 화면 모드시 : 240 \* 320 픽셀

- x = 320 - 1, y = 240 - 1

관련 함수 : Setcolor

[예제]

C기본 헤더 선언

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

// CPU 초기화 및 인터럽트 관리 프로그램

```
#include "tftlcd.inc"
```

// 그래픽 TFT-LCD 라이브러리

```
void main(void)
```

```
{
```

```
    Cpu_Setup();
```

// CPU 초기화

```
    Clcd_Init();
```

// 그래픽 TFT-LCD 초기화

```
// TFT-LCD TEST Code..
```

```
Ellipse(64,32,20,10);
```

// 타원을 그림

```
while(1)
```

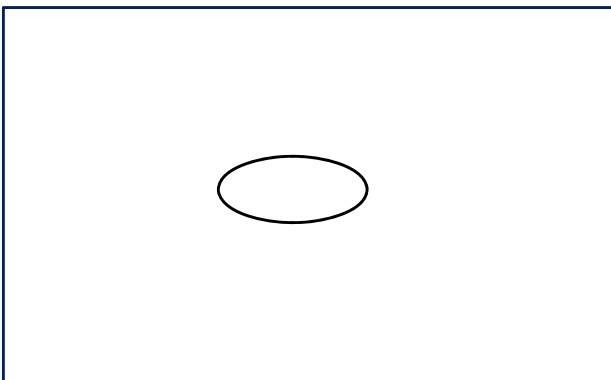
```
{
```

```
    사용자 프로그램
```

```
}
```

```
}
```

\* 실행 결과(LCD 화면)





## Putpixel

지정한 색으로 점을 찍음.

함 수 : void Putpixel(int x, int y, unsigned color)  
인 수 : int x // x(가로) 좌표(0 - 319 / 0 - 239)  
int y // y(세로) 좌표(0 - 239 / 0 - 319)  
unsigned color // Setcolor 색상 참조  
반 환 : 없음.

설 명 : 지정한 좌표에 점을 찍음.  
가로 화면 모드시 : 320 \* 240 픽셀  
- x = 320 - 1, y = 240 - 1  
세로 화면 모드시 : 240 \* 320 픽셀  
- x = 320 - 1, y = 240 - 1

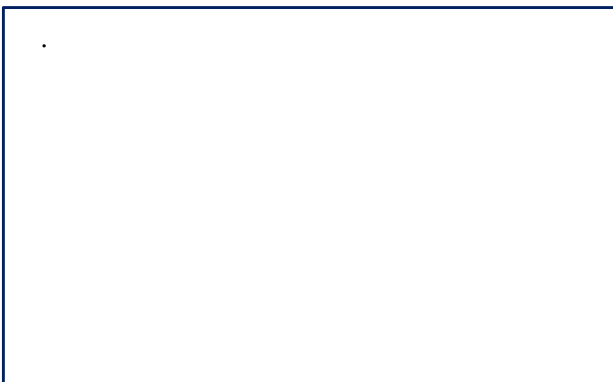
[예제]

.  
C기본 헤더 선언  
.

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램  
#include "tftlcd.inc" // 그래픽 TFT-LCD 라이브러리
```

```
void main(void)  
{  
    Cpu_Setup(); // CPU 초기화  
    Clcd_Init(); // 그래픽 TFT-LCD 초기화  
  
    // TFT-LCD TEST Code..  
    Putpixel(10,10,_WHITE); // 화면의 10(x),10(y)위치에 점을 찍음.  
    while(1)  
    {  
        사용자 프로그램  
    }  
}
```

\* 실행 결과(LCD 화면)



## Putpixelb

지정한 색으로 바이트단위 픽셀 표시.

함 수 : void Putpixelb(int x, int y, unsigned char datac, unsigned color)  
인 수 : int x // x(가로) 좌표(0 - 319 / 0 - 239)  
 int y // y(세로) 좌표(0 - 239 / 0 - 319)  
 unsigned color // Setcolor 색상 참조  
반 환 : 없음.

설 명 : 지정한 좌표에 세로방향으로 바이트단위 픽셀 표시.

가로 화면 모드시 : 320 \* 240 픽셀

- x = 320 - 1, y = 240 - 1

세로 화면 모드시 : 240 \* 320 픽셀

- x = 320 - 1, y = 240 - 1

[예제]

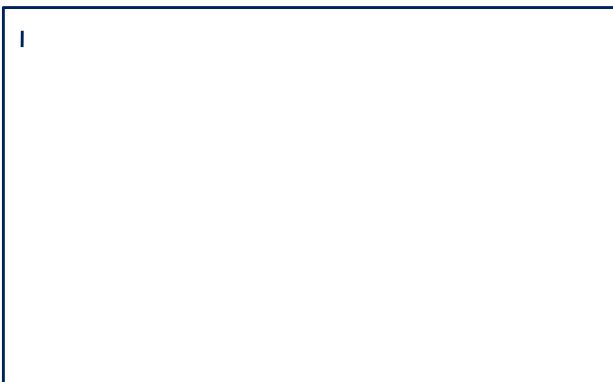
C기본 헤더 선언

```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc" // 그래픽 TFT-LCD 라이브러리

void main(void)
{
    Cpu_Setup(); // CPU 초기화
    Clcd_Init(); // 그래픽 TFT-LCD 초기화

    // TFT-LCD TEST Code..
    Putpixelb(10,10,0xff,_WHITE); // 화면의 10(x),10(y)위치에 바이트단위 픽셀 표시.
    while(1)
    {
        사용자 프로그램
    }
}
```

\* 실행 결과(LCD 화면)



## PutImage

커져 위치에 영문/한글 이미지 표시

함 수 : PutImage(unsigned char dbuf[],int xsize,int ysize)  
인 수 : unsigned char dbuf[] // 표시할 폰트 데이터  
int xsize // 표시할 x축 데이터 크기  
int ysize // 표시할 y축 데이터 크기  
반 환 : 없음.

설 명 : 현재 커져 위치에 지정한 색으로 문자 폰트를 표시한다.

가로 화면 모드시 : 320 \* 240 픽셀

- x = 320 - 1, y = 240 - 1

세로 화면 모드시 : 240 \* 320 픽셀

- x = 320 - 1, y = 240 - 1

참 고 : \_screen\_color : 문자색(Setcolor())  
\_screen\_bkcolor : 바탕색(Setbkcolor())  
\_screen\_curx / \_screen\_cury : X,Y좌표(Gotoxy())

[예제]

C기본 헤더 선언

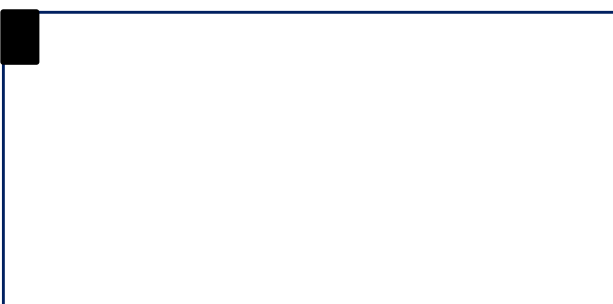
```
#include "cpusetup.inc" // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc" // 그래픽 TFT-LCD 라이브러리

void main(void)
{
    unsigned char buf[32]
    int i;

    Cpu_Setup(); // CPU 초기화
    Clcd_Init(); // 그래픽 TFT-LCD 초기화

    // TFT-LCD TEST Code..
    for(i = 0; i < 32; i++) buf[i] = 0xff;
    Setcolor(_WHITE); // 문자색 지정
    Gotoxy(0,0); // 커져 위치 지정
    PutImage(buf,16,16); // 화면에 Image 데이터 표시(x크기16 / y크기16)
    while(1)
    {
        사용자 프로그램
    }
}
```

\* 실행 결과(LCD 화면)



## RGB\_Disp

현재 화면에 RGB 패턴 데이터를 표시 한다.

함 수 void RGB\_Disp(void)

인 수 : void

반 환 : void

설 명 : 화면에 RGB 패턴 데이터를 표시 한다.

가로 화면 모드시 : 320 \* 240 픽셀

- x = 320 - 1, y = 240 - 1

세로 화면 모드시 : 240 \* 320 픽셀

- x = 320 - 1, y = 240 - 1

[예제]

.  
C기본 헤더 선언

```
#include "cpusetup.inc"           // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc"             // 그래픽 TFT-LCD 라이브러리
```

```
void main(void)
{
    Cpu_Setup();                 // CPU 초기화
    Clcd_Init();                 // 그래픽 TFT-LCD 초기화
```

```
// TFT-LCD TEST Code..
    RGB_Disp( );                 // 패턴 표시
    while(1)
    {
        사용자 프로그램
    }
}
```

\* 실행 결과(LCD 화면)



## Han\_Conv

조합형 코드를 완성형 코드로 변환한다.

함 수 : void Han\_Conv(char \*buf)

인 수 : char \*buf // 변환할 문자열 버퍼

반 환 : 없음.

변 수 : unsigned char han\_conv\_buf[32] // 한글 완성형 코드 저장 버퍼(32byte)

설 명 : 조합형 코드를 완성형 코드로 변환 한다.

## CLCD\_GRAM\_dread

설정된 영역의 픽셀 데이터를 취득(16Bit RGB)

함 수 : unsigned CLCD\_GRAM\_dread(void)

인 수 : void

반 환 : 16Bit RGB 데이터(RRRRRR GGGGGG BBBB)

설 명 : 현재 커저 위치에 표시된 픽셀의 데이터 값을 취득 한다.

가로 화면 모드시 : 320 \* 240 픽셀

- x = 320 - 1, y = 240 - 1

세로 화면 모드시 : 240 \* 320 픽셀

- x = 320 - 1, y = 240 - 1

[예제]

C기본 헤더 선언

```
.
#include "cpusetup.inc"           // CPU 초기화 및 인터럽트 관리 프로그램
#include "tftlcd.inc"           // 그래픽 TFT-LCD 라이브러리

void main(void)
{
    unsigned datau;

    Cpu_Setup();                // CPU 초기화
    Clcd_Init();                // 그래픽 TFT-LCD 초기화

    // TFT-LCD TEST Code..
    if(_screen_mode == _SCREEN_COLUMN_SET)
        ScreenSet(_SCREEN_COLUMN_SET,0,0,_SCREEN_COLUMN_X,_SCREEN_COLUMN_Y);// 가로 지정
    else ScreenSet(_SCREEN_ROW_SET,0,0,_SCREEN_ROW_X,_SCREEN_ROW_Y);// 세로 지정
    datau = CLCD_GRAM_dread();    // 픽셀 리드
    printf("pixel data : %04x\n",datau); // 데이터 표시
    while(1)
    {
        사용자 프로그램
    }
}
```

## CLCD\_Init

Color TFT-LCD Driver IC를 초기화 한다.

함 수 : void Clcd\_Init(void)

인 수 : void

반 환 : void

설 명 : Color TFT-LCD Driver-IC를 초기화 한다.  
(가로 화면 모드(320\*240)로 설정)

[예제]

.  
C기본 헤더 선언

```
.  
#include "cpusetup.inc"           // CPU 초기화 및 인터럽트 관리 프로그램  
#include "ftlcd.inc"             // 그래픽 TFT-LCD 라이브러리  
  
void main(void)  
{  
    unsigned datau;  
  
    Cpu_Setup();                 // CPU 초기화  
    Clcd_Init();                 // 그래픽 TFT-LCD 초기화  
  
    while(1)  
    {  
        사용자 프로그램  
    }  
}
```

## CLCD\_Command\_Data

Color TFT-LCD Driver IC의 Register NUM와 데이터를 써넣는다.

```
함 수 : void CLCD_Command_Data(int regnum,unsigned datau)
인 수 : int regnum                // Register Number
        unsigned datau           // Register에 써넣을 데이터
반 환 : void
```

설 명 : Color TFT-LCD Driver IC의 Register NUM와 데이터를 써넣는다.



## CLCD\_Command

Color TFT-LCD Driver IC의 Register NUM을 써넣는다.

함 수 : void CLCD\_Command(int regnum)

인 수 : int regnum // Register Number

반 환 : void

설 명 : Color TFT-LCD Driver IC의 Register NUM을 써넣는다.

## CLCD\_Cdata

Color TFT-LCD Driver IC의 Register NUM에 데이터를 써 넣는다.

함 수 : void CLCD\_Cdata(unsigned datau)

인 수 : unsigned datau

// Register에 써 넣을 데이터

반 환 : void

설 명 : Color TFT-LCD Driver IC의 Register NUM에 데이터를 써 넣는다.

# Touch 라이브러리(touch.inc)

## 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.  
취득한 좌표 데이터를 현재 표시 화면 모드에 맞게 변환 하여 저장 한다.

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  
#include "touch.inc" // 터치 관리 라이브러리  
  
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  
    }  
}  
}
```

## 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  
#include TSC2003_Setup(); // I2c 초기화및 TSC2003초기화  
void main(void)  
{  
    unsigned datau;  
  
    Cpu_Setup(); // CPU 초기화  
    TSC2003_Setup(); // I2C 초기화및 TSC2003초기화  
    while(1)  
    {  
        사용자 프로그램  
    }  
}
```

## TSC2003\_Word\_Read

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.

## 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.