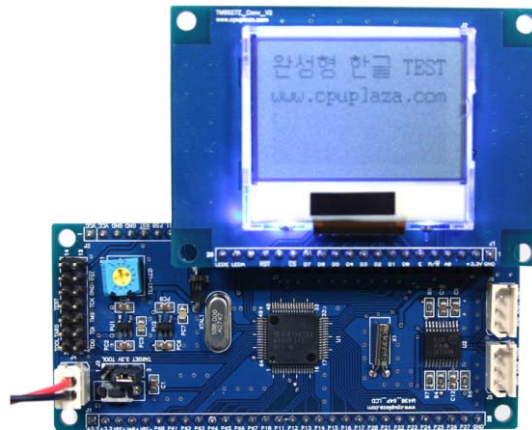


MSP430_GrapLcd

(Graphic-LCD 메뉴얼)



CPUPLAZA

제품 소개

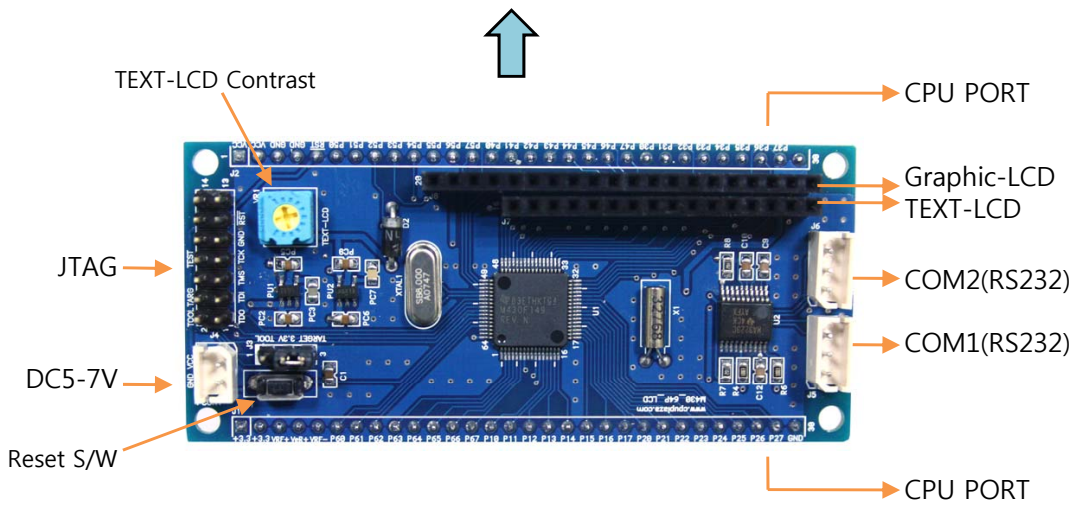
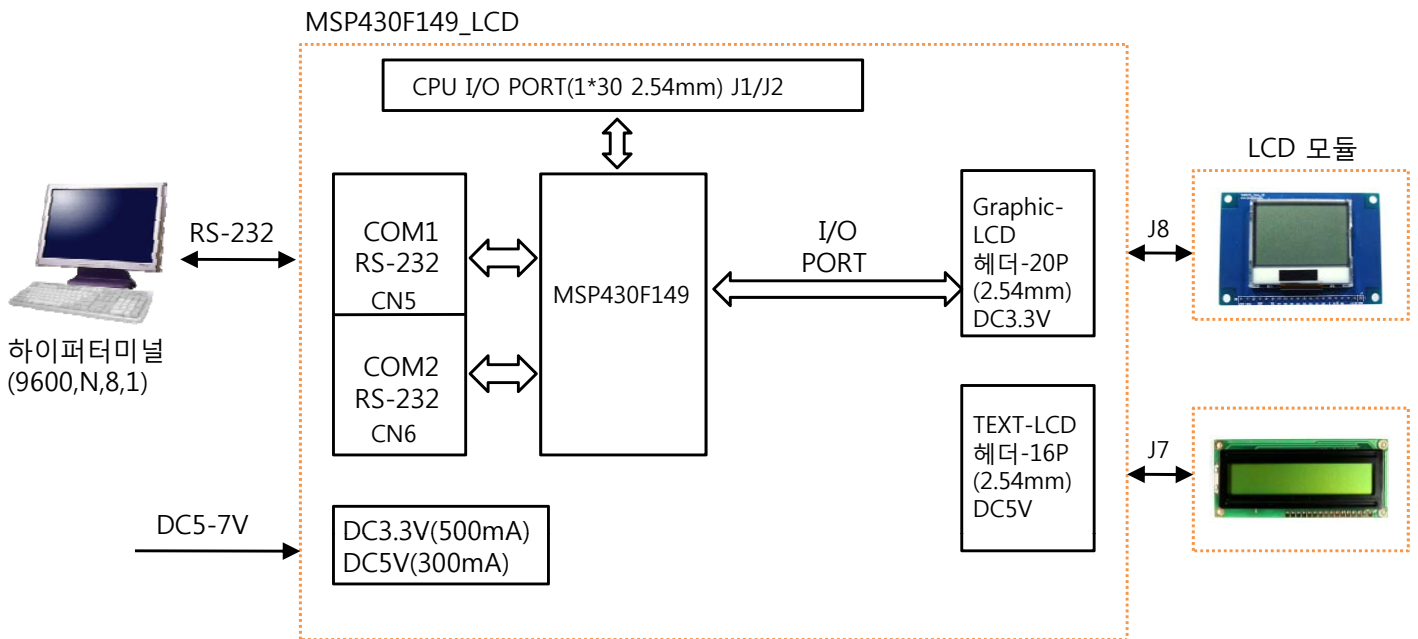
MEGA128_CLCD는 TEXT-LCD, Graphic-LCD(흑백), Color TFT-LCD를 연결하여 테스트 할 수 있으며, 압전식 터치 컨트롤러(TSC2003)가 내장되어 있어 LCD 모델에 상관없이 터치를 별도 헤더 핀으로 지원합니다.

제품 특징

- 전원 : 외부 DC 5.0-7V
- TEXT LCD : 16*2 지원
 - I/O 방식 제어
- Graphic-LCD(흑백) : 128*64 지원
 - I/O 방식 제어
- MSP430F149 CPU 모든 I/O 1열 헤더 지원
- RS232 2PORT(절전모드) 포트 별도 지원
- 완성형 한글 지원(완성형 -> 조합형 코드 변환 사용)
- TEXT-LCD/Graphic-LCD 라이브러리(소스 제공)
 - * Graphic-LCD 라이브러리 함수
 - . Clscr() / Cleardevice() // 화면 소거
 - . Gotoxy() / Gotoxyt() // 커저 좌표 지정
 - . Setcolor() // 컬러 지정
 - . Backlight_Ctrl() // 백라이트 ON/OFF
 - . 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
 - . GLCD_Dread() // 바이트 단위 정보 취득

제품 사양

1. 제품 구성 및 명칭



2. 커넥터 기능 및 사양

PCN1 : DC 전원 입력

NUM	NAME
1	DC5V-7V
2	GND

J3 : 전원 입력 방법 설정

NUM	NAME
1-2	TOOL(보드<-JTAG)
2-3	TARGET(보드->JTAG)

J4 : JTAG(4Wire)

NUM	NAME	NUM	NAME
1	TDO	8	TEST
2	TOOL 전원(보드 <- JTAG)	9	GND
3	TDI	10	N.C
4	TARGET 전원(보드 -> JTAG)	11	RSTW
5	TMS	12	N.C
6	N.C	13	N.C
7	TCK	14	N.C

J6: UART2(RS-232)

NUM	NAME
1	RXD
2	TXD
3	+3.3V
4	GND

J5: UART1(RS-232)

NUM	NAME
1	RXD
2	TXD
3	+3.3V
4	GND

PORT P3.3으로 파워다운 제어
* 자동 파워다운 기능

J7 : TEXT-LCD

NUM	NAME	PORT
1	GND	GND
2	VCC	+5V
3	VO	가변 저항
4	RS	P5.0
5	R/W	P5.1
6	E	P5.2
7-14	DB0-DB7	P4.0-P4.7
15	LEDA	+4.3V
16	LEDK	GND

J8 : Graphic-LCD

NUM	NAME	PORT
1	GND	GND
2	VCC	+3.3V
3	N.C	
4	A0	P5.0
5	R/W	P5.1
6	E	P5.2
7-14	DB0-DB7	P4.0-P4.7
15	CS	P5.3
16	N.C	
17	RSTW	P5.5
18	N.C	
19	LEDA	+5V
20	LEDE	P5.4

J1 : I/O 확장

NUM	NAME
1 2	3.3V(내부 LDO 출력) 500mA
3	VREF+
4	VeREF+
5	VREF-/VeREF-
6	P6.0
7	P6.1
8	P6.2
9	P6.3
10	P6.4
11	P6.5
12	P6.6
13	P6.7
14	P1.0
15	P1.1

NUM	NAME
16	P1.2
17	P1.3
18	P1.4
19	P1.5
20	P1.6
21	P1.7
22	P2.0
23	P2.1
24	P2.2
25	P2.3
26	P2.4
27	P2.5
28	P2.6
29	P2.7
30	GND

J2 : I/O 확장

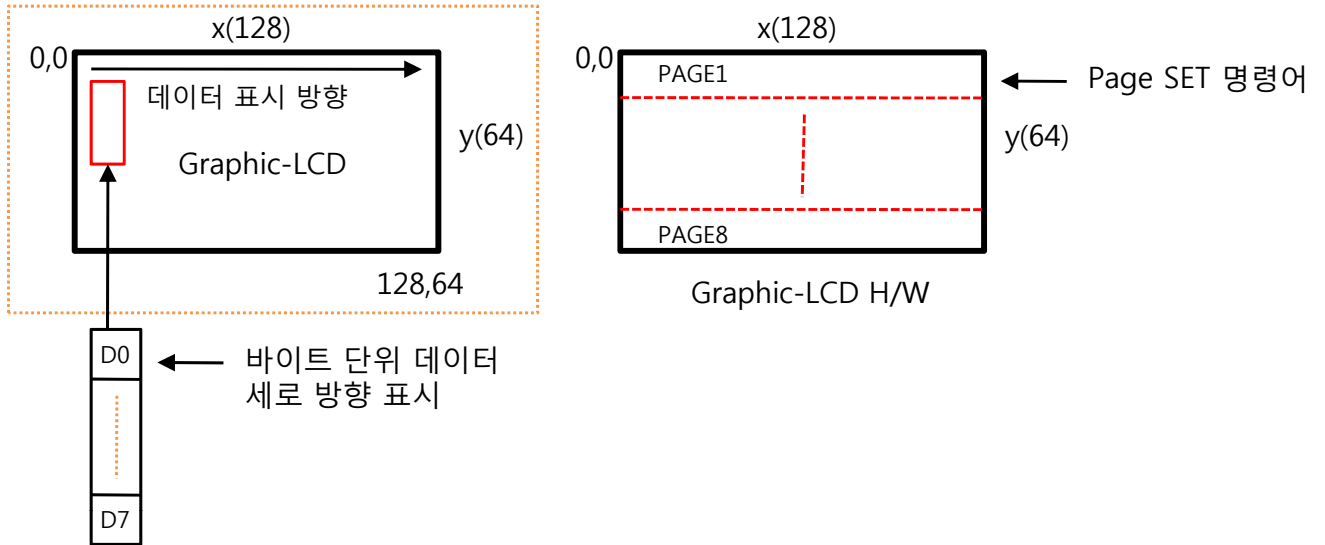
NUM	NAME
1 2	VCC(외부 전원 입력) DC5-7V
3 4	GND
5	RSTW
6	P5.0
7	P5.1
8	P5.2
9	P5.3
10	P5.4
11	P5.5
12	P5.6
13	P5.7
14	P4.0
15	P4.1

NUM	NAME
16	P4.2
17	P4.3
18	P4.4
19	P4.5
20	P4.6
21	P4.7
22	P3.0
23	P3.1
24	P3.2
25	P3.3
26	P3.4
27	P3.5
28	P3.6
29	P3.7
30	N.C

3. LCD 설명

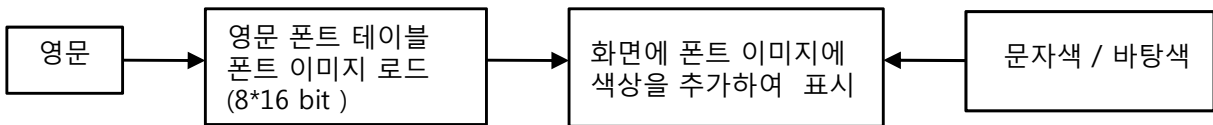
1. Graphic-LCD는 가로(128) 세로(64)의 영역에 한글 및 여러가지 그래픽 함수를 지원 합니다.

- Graphic-LCD



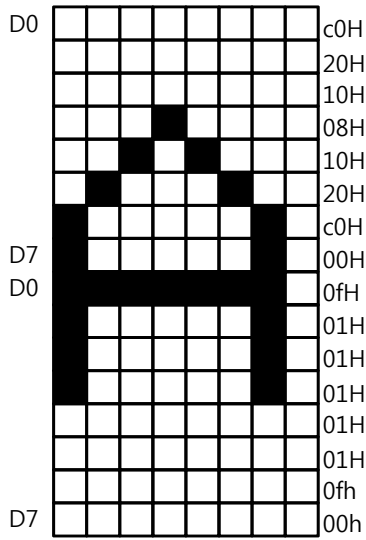
4. LCD 문자 표시

1. Graphic-LCD 화면에 영문자 'A'를 표현 할 경우 예제 입니다.



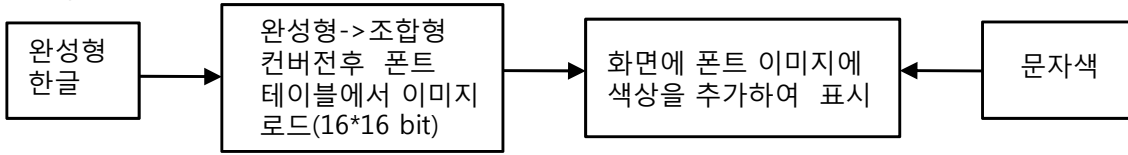
'A'
41H

NUM	DATA
0	c0H
1	20H
2	10H
3	08H
4	10H
5	20H
6	c0H
7	00H
8	0fH
9	01H
10	01H
11	01H
12	01H
13	01H
14	0fH
15	00H



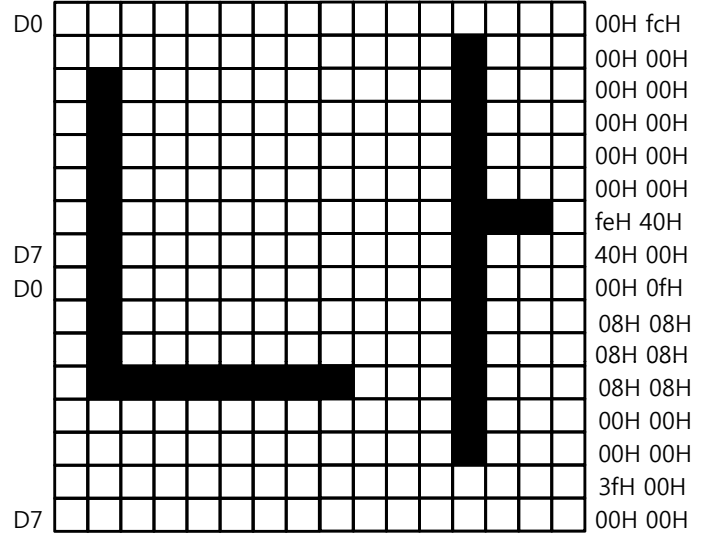
폰트 이미지 16Byte
상위 8Byte,하위 8Byte 이미지 표시

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



완성형
"나"
B3AAH
↓
조합형
"나"
9061H

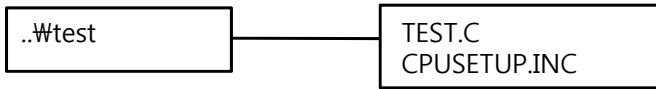
NUM	DATA	NUM	DATA
0	00H	16	00H
1	fcH	17	0fH
2	00H	18	08H
3	00H	19	08H
4	00H	20	08H
5	00H	21	08H
6	00H	22	08H
7	00H	23	08H
8	00H	24	00H
9	00H	25	00H
10	00H	26	00H
11	00H	27	00H
12	feH	28	3fH
13	40H	29	00H
14	40H	30	00H
15	00H	31	00H



폰트 이미지 32Byte
상위 16Byte,하위 16Byte 이미지 표시

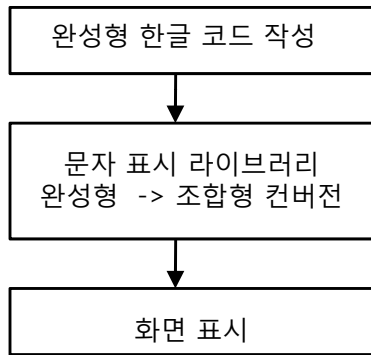
S/W 사양

1. 프로그램 구성



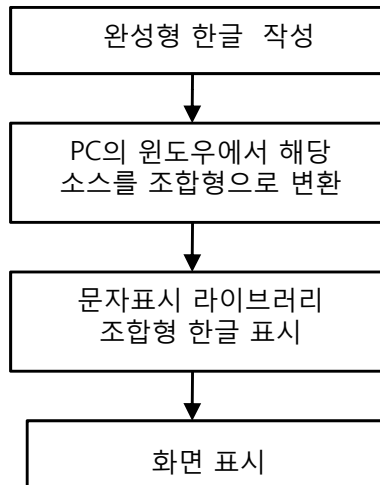
1. TEST.C : 메인 프로그램
통신(19200,N8,1)으로 LCD 테스트 프로그램 관리
2. CPUSETUP.INC :
인터럽트 관리 및 CPU 초기화 루틴

* 완성형 코드 지원시



(당사 제품)

* 조합형 코드 지원시



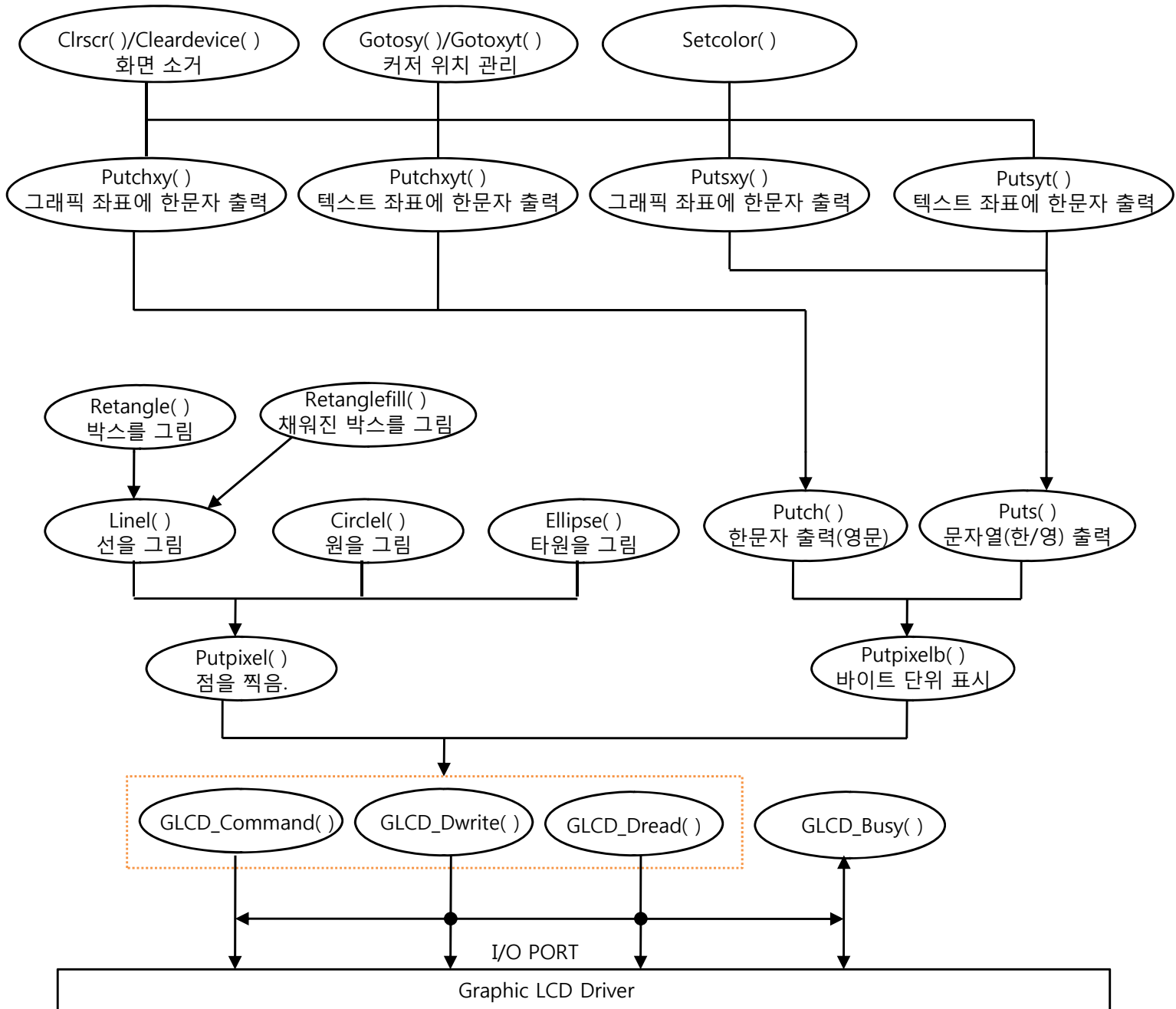
** Windows Program **

. HanConv.exe

. 산(DOS용)

* 소스 코드명이 달라야 함

2. Graphic-LCD 라이브러리 구성

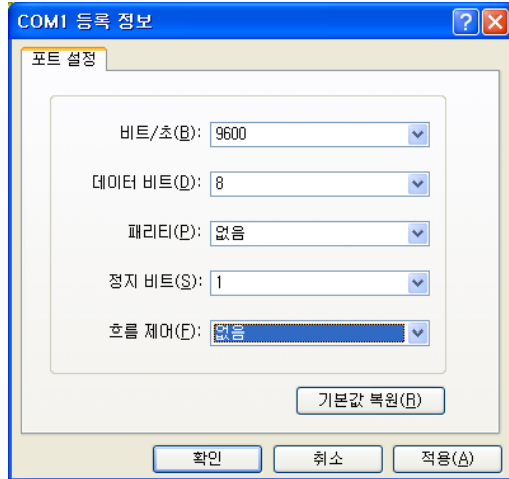
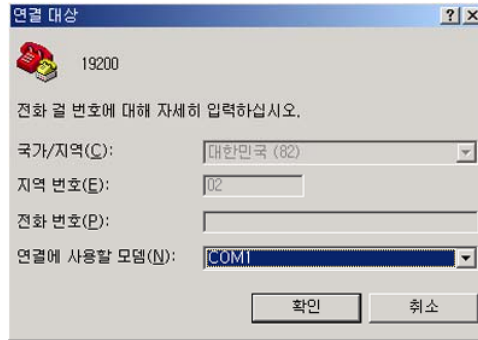
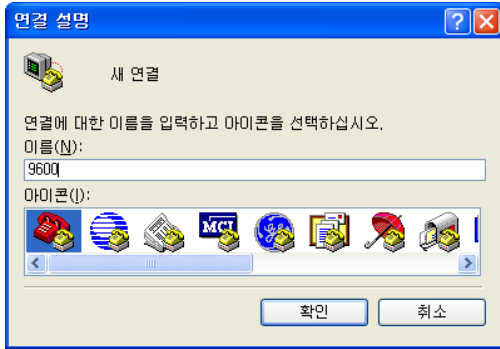


3. PC 통신 프로그램

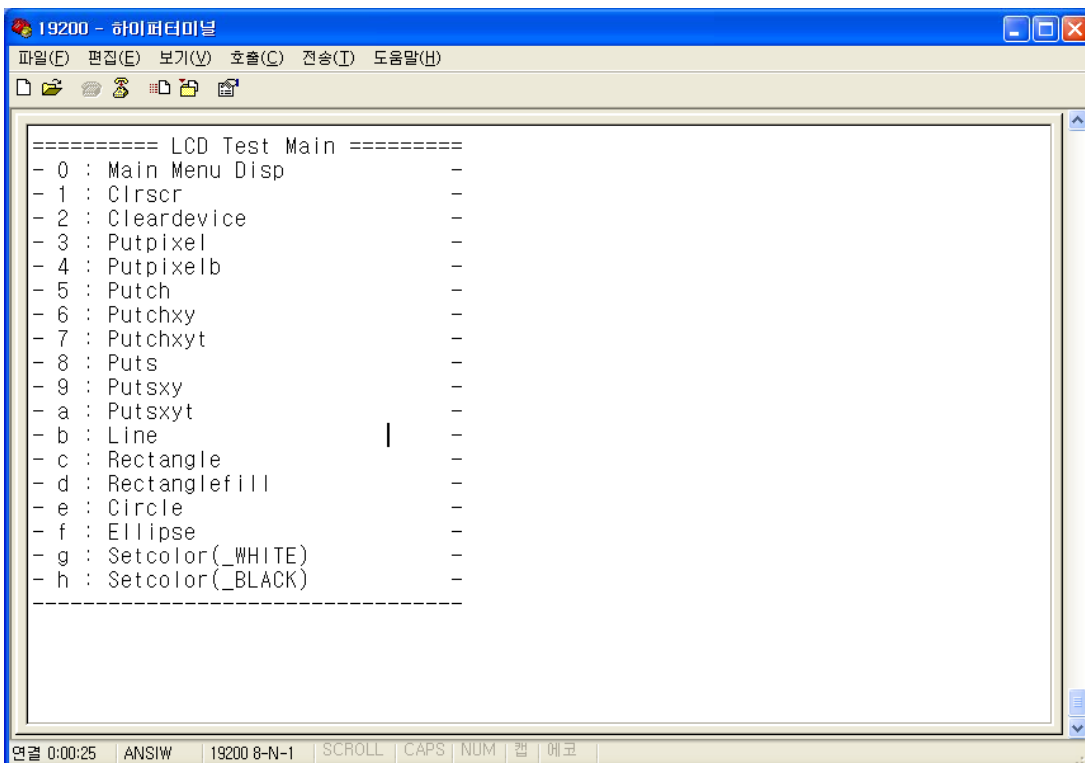
1. 하이퍼 터미널을 시작 합니다.



2. 하이퍼 터미널에서 통신 포트를 설정합니다.



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



LCD 라이브러리(glcd.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

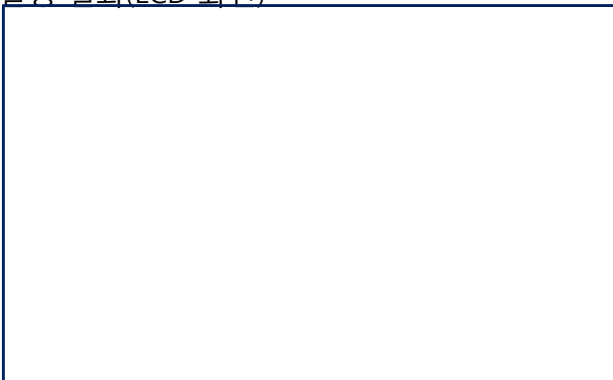
```
.
C기본 헤더 선언
.

#include "eng8x16.fnt"           // 영문 폰트
#include "han16x16.fnt"         // 한글 폰트
#include "hconvtbl.inc"         // 완성형 -> 조합형 변환 테이블

void main(void)
{
    cpu_setup();                // CPU 초기화
    GLCD_Init();                // 그래픽 LCD 초기화

    // LCD TEST Code..
    Clrscr();                    // 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.

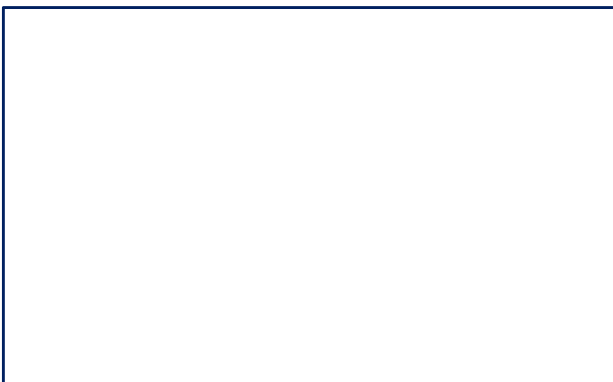
설 명 : 현재 화면을 모두 지운다.

[예제]

.
C기본 헤더 선언

```
.  
#include "cpusetup.inc"           // CPU 초기화 및 인터럽트 관리 프로그램  
#include "glcd.inc"              // 그래픽 LCD 라이브러리  
  
void main(void)  
{  
    Cpu_Setup();                 // CPU 초기화  
    Lcd_Init();                  // 그래픽 LCD 초기화  
  
    // LCD TEST Code..  
    Cleardevice();              // 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 - 127)
        int y                // 커저의 y(세로) 좌표(0 - 63)
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.

* 128 * 64 그래픽 LCD 일때

x = 128 - 1

y = 64 - 1

관련 함수 : Putch,Puts,Puts_f,Setcolor

[예제]

C기본 헤더 선언

```
.
#include "eng8x16.fnt"        // 영문 폰트
#include "han16x16.fnt"      // 한글 폰트
#include "hconvtbl.inc"      // 완성형 -> 조합형 변환 테이블

void main(void)
{
    cpu_setup();            // CPU 초기화
    GLCD_Init();           // 그래픽 LCD 초기화

    // LCD TEST Code..
    Gotoxy(1,2);           // 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 - 16)
        int y                // 커저의 y(세로) 좌표(1 - 4)
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.

* 128 * 64 그래픽 LCD 일때

x : $128 / 8 = 16$

y : $64 / 16 = 4$

관련 함수 : Putch,Puts,Puts_f,Setcolor

[예제]

C기본 헤더 선언

```
.
#include "eng8x16.fnt"        // 영문 폰트
#include "han16x16.fnt"      // 한글 폰트
#include "hconvtbl.inc"      // 완성형 -> 조합형 변환 테이블

void main(void)
{
    cpu_setup();            // CPU 초기화
    GLCD_Init();           // 그래픽 LCD 초기화

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

SetColor

문자색을 지정 한다.

함 수 : void Setcolor(int color)
인 수 : int color // 표시할 문자색
 _BLACK // 검정
 _WHITE // 흰색(평상시 문자색)
반 환 : 없음.

설 명 : LCD에 표시되는 문자색을 지정 한다.
변 수 : int _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 "eng8x16.fnt" // 영문 폰트
#include "han16x16.fnt" // 한글 폰트
#include "hconvtbl.inc" // 완성형 -> 조합형 변환 테이블
```

```
void main(void)
{
    cpu_setup(); // CPU 초기화
    GLCD_Init(); // 그래픽 LCD 초기화

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

* 실행 결과(LCD 화면)



1

Backlight_Ctrl

백라이트를 ON/OFF 한다.

```
함 수 : void Backlight_Ctrl (char sts)
인 수 : char sts                // 백라이트 ON/OFF
        0                        // OFF
        1                        // ON
```

반 환 : 없음.

설 명 : LCD의 백라이트 LED를 제어 한다.

[예제]

.
C기본 헤더 선언

```
.
#include "eng8x16.fnt"           // 영문 폰트
#include "han16x16.fnt"         // 한글 폰트
#include "hconvtbl.inc"         // 완성형 -> 조합형 변환 테이블
```

```
void main(void)
{
    cpu_setup();                // CPU 초기화
    GLCD_Init();                // 그래픽 LCD 초기화
```

```
// LCD TEST Code..
Backlight_Ctrl(1);             // 백라이트 ON
Putch('1');                    // 화면에 '1' 표시
while(1)
{
    사용자 프로그램
}
}
```


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,Gotoxy,Gotoxyt

[예제]

.
C기본 헤더 선언

```
#include "eng8x16.fnt" // 영문 폰트
#include "han16x16.fnt" // 한글 폰트
#include "hconvtbl.inc" // 완성형 -> 조합형 변환 테이블
```

```
void main(void)
{
    cpu_setup(); // CPU 초기화
    GLCD_Init(); // 그래픽 LCD 초기화

    // 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 - 127)
int y // 커저의 y(세로) 좌표(0 - 63)
unsigned c // 표시할 문자
반 환 : 없음.

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

* 128 * 64 그래픽 LCD 일때

x = 128 - 1

y = 64 - 1

관련 함수 : Setcolor

[예제]

```
.  
C기본 헤더 선언  
.  
  
#include "eng8x16.fnt" // 영문 폰트  
#include "han16x16.fnt" // 한글 폰트  
#include "hconvtbl.inc" // 완성형 -> 조합형 변환 테이블  
  
void main(void)  
{  
    cpu_setup(); // CPU 초기화  
    GLCD_Init(); // 그래픽 LCD 초기화  
  
    // 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 - 16)
int y // 커저의 y(세로) 좌표(1 - 4)
int unsigned c // 표시할 문자
반 환 : 없음.

설 명 : 지정한 텍스트 좌표에 문자를 표시 한다.

(한글 코드 제외)

* 128 * 64 그래픽 LCD 일때

x : 128 / 8 = 16

y : 64 / 16 = 4

관련 함수 : Setcolor

[예제]

```
.
C기본 헤더 선언
.

#include "eng8x16.fnt" // 영문 폰트
#include "han16x16.fnt" // 한글 폰트
#include "hconvtbl.inc" // 완성형 -> 조합형 변환 테이블

void main(void)
{
    cpu_setup(); // CPU 초기화
    GLCD_Init(); // 그래픽 LCD 초기화

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

* 실행 결과(LCD 화면)

A rectangular box representing the LCD screen. Inside the box, the number '2' is displayed in the top-left corner.

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, Gotoxy, Gotoxyt

[예제]

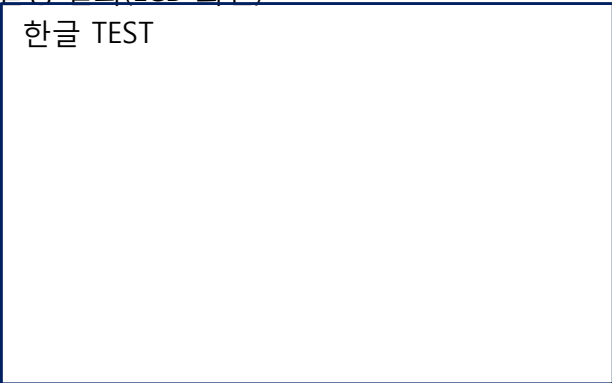
.
C기본 헤더 선언

```
#include "eng8x16.fnt" // 영문 폰트
#include "han16x16.fnt" // 한글 폰트
#include "hconvtbl.inc" // 완성형 -> 조합형 변환 테이블
```

```
void main(void)
{
    cpu_setup(); // CPU 초기화
    GLCD_Init(); // 그래픽 LCD 초기화

    // 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 - 127)
int y // 커저의 y(세로) 좌표(0 - 63)
const char *str // 표시할 문자열
반 환 : 없음.

설 명 : 지정한 그래픽 좌표에 문자열을 표시 한다.

(완성형 한글 코드 포함)
* 128 * 64 그래픽 LCD 일때

x = 128 - 1

y = 64 - 1

관련 함수 : Setcolor

[예제]

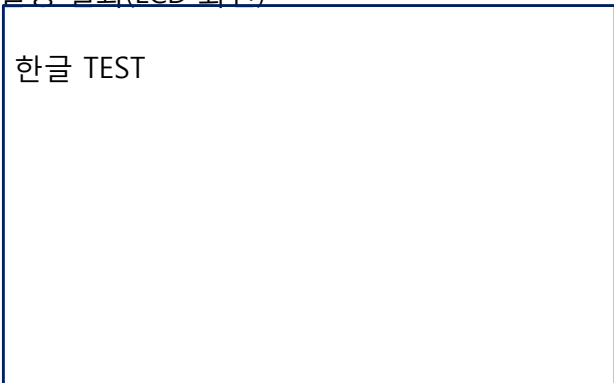
.
C기본 헤더 선언

```
#include "eng8x16.fnt" // 영문 폰트
#include "han16x16.fnt" // 한글 폰트
#include "hconvtbl.inc" // 완성형 -> 조합형 변환 테이블
```

```
void main(void)
{
    cpu_setup(); // CPU 초기화
    GLCD_Init(); // 그래픽 LCD 초기화

    // 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 - 16)
int y // 커저의 y(세로) 좌표(1 - 4)
const char *str // 표시할 문자
반 환 : 없음.

설 명 : 지정한 텍스트 좌표에 문자열을 표시 한다.

(완성형 한글 코드 포함)

* 128 * 64 그래픽 LCD 일때

x : 128 / 8 = 16

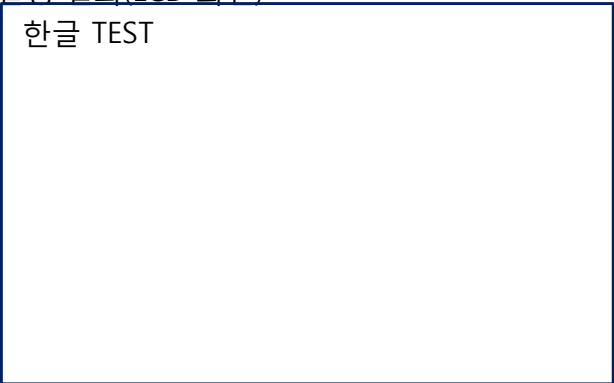
y : 64 / 16 = 4

관련 함수 : Setcolor

[예제]

```
.  
C기본 헤더 선언  
.  
  
#include "eng8x16.fnt" // 영문 폰트  
#include "han16x16.fnt" // 한글 폰트  
#include "hconvtbl.inc" // 완성형 -> 조합형 변환 테이블  
  
void main(void)  
{  
    cpu_setup(); // CPU 초기화  
    GLCD_Init(); // 그래픽 LCD 초기화  
  
    // 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

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 - 127)
        int y1           // 시작점 y(세로) 좌표(0 - 63)
        int x2           // 끝점 x(가로) 좌표(0 - 127)
        int y2           // 끝점 y(세로) 좌표(0 - 63)
```

반 환 : 없음.

설 명 : 지정한 좌표에 선을 그림.
* 128 * 64 그래픽 LCD 일때
x = 128 - 1
y = 64 - 1

관련 함수 : Setcolor

[예제]

C기본 헤더 선언

```
#include "eng8x16.fnt" // 영문 폰트
#include "han16x16.fnt" // 한글 폰트
#include "hconvtbl.inc" // 완성형 -> 조합형 변환 테이블
```

```
void main(void)
{
    cpu_setup(); // CPU 초기화
    GLCD_Init(); // 그래픽 LCD 초기화

    // 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 - 127)
        int y1           // 시작점 y(세로) 좌표(0 - 63)
        int x2           // 끝점 x(가로) 좌표(0 - 127)
        int y2           // 끝점 y(세로) 좌표(0 - 63)
```

반 환 : 없음.

설 명 : 지정한 좌표에 박스를 그림.
* 128 * 64 그래픽 LCD 일때
x = 128 - 1
y = 64 - 1

관련 함수 : Line,Setcolor

[예제]

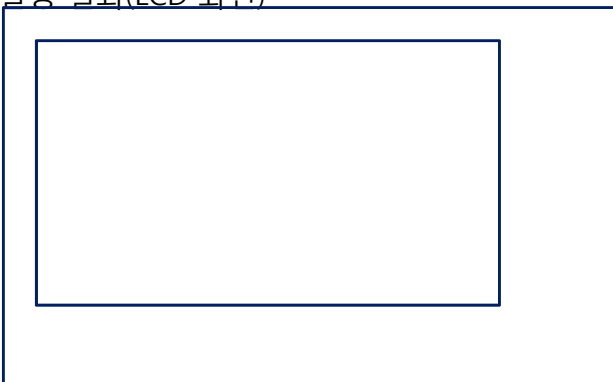
C기본 헤더 선언

```
#include "eng8x16.fnt"           // 영문 폰트
#include "han16x16.fnt"         // 한글 폰트
#include "hconvtbl.inc"         // 완성형 -> 조합형 변환 테이블
```

```
void main(void)
{
    cpu_setup();                // CPU 초기화
    GLCD_Init();                // 그래픽 LCD 초기화

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

* 실행 결과(LCD 화면)



Retanglefill

지정한 좌표에 채워진 박스를 그림.

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

```
인 수 : int x1           // 시작점 x(가로) 좌표(0 - 127)
        int y1           // 시작점 y(세로) 좌표(0 - 63)
        int x2           // 끝점 x(가로) 좌표(0 - 127)
        int y2           // 끝점 y(세로) 좌표(0 - 63)
```

반 환 : 없음.

설 명 : 지정한 좌표에 채워진 박스를 그림.

* 128 * 64 그래픽 LCD 일때

x = 128 - 1

y = 64 - 1

관련 함수 : Line,Setcolor

[예제]

C기본 헤더 선언

```
#include "eng8x16.fnt"           // 영문 폰트
#include "han16x16.fnt"         // 한글 폰트
#include "hconvtbl.inc"         // 완성형 -> 조합형 변환 테이블
```

```
void main(void)
{
    cpu_setup();                // CPU 초기화
    GLCD_Init();                // 그래픽 LCD 초기화

    // LCD TEST Code..
    Retanglefill(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           // 중심점 x(가로) 좌표(0 - 127)
        int yy           // 중심점 y(세로) 좌표(0 - 63)
        int radius       // 반지름 좌표(0 - 127)
```

반 환 : 없음.

설 명 : 지정한 좌표에 원을 그림.
* 128 * 64 그래픽 LCD 일때
x = 128 - 1
y = 64 - 1

관련 함수 : Setcolor

[예제]

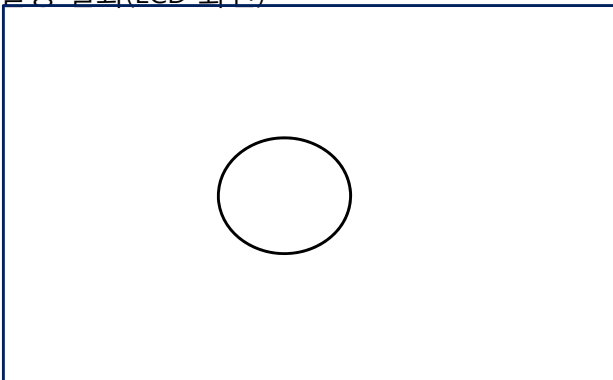
```
.
C기본 헤더 선언
.

#include "eng8x16.fnt"           // 영문 폰트
#include "han16x16.fnt"        // 한글 폰트
#include "hconvtbl.inc"        // 완성형 -> 조합형 변환 테이블

void main(void)
{
    cpu_setup();               // CPU 초기화
    GLCD_Init();              // 그래픽 LCD 초기화

// 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           // 중심점 x(가로) 좌표(0 - 127)
        int yy           // 중심점 y(세로) 좌표(0 - 63)
        int a0           // 가로 반경(0 - 127)
        int a0           // 세로 반경(0 - 127)
반 환 : 없음.
```

설 명 : 지정한 좌표에 타원을 그림.

* 128 * 64 그래픽 LCD 일때

x = 128 - 1

y = 64 - 1

관련 함수 : Setcolor

[예제]

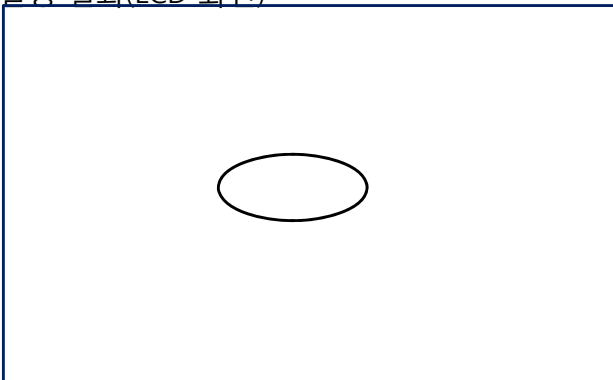
```
.
C기본 헤더 선언
.

#include "eng8x16.fnt"           // 영문 폰트
#include "han16x16.fnt"        // 한글 폰트
#include "hconvtbl.inc"        // 완성형 -> 조합형 변환 테이블

void main(void)
{
    cpu_setup();               // CPU 초기화
    GLCD_Init();              // 그래픽 LCD 초기화

// 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, char color)
인 수 : int x // x(가로) 좌표(0 - 127)
int y // y(세로) 좌표(0 - 63)
char color // 표시할 색상
_BLACK // 검정
_WHITE // 흰색(평상시 문자색)
반 환 : 없음.

설 명 : 지정한 좌표에 점을 찍음.
* 128 * 64 그래픽 LCD 일때
x = 128 - 1
y = 64 - 1

[예제]

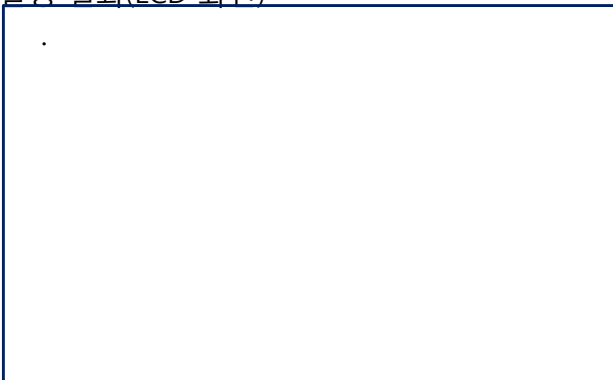
C기본 헤더 선언

```
#include "eng8x16.fnt" // 영문 폰트
#include "han16x16.fnt" // 한글 폰트
#include "hconvtbl.inc" // 완성형 -> 조합형 변환 테이블
```

```
void main(void)
{
    cpu_setup(); // CPU 초기화
    GLCD_Init(); // 그래픽 LCD 초기화

    // 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, char color)
인 수 : int x // x(가로) 좌표(0 - 127)
int y // y(세로) 좌표(0 - 63)
unsigned char datac // 표시할 픽셀 바이트
char color // 표시할 색상
_BLACK // 검정
_WHITE // 흰색(평상시 문자색)
반 환 : 없음.

설 명 : 지정한 좌표에 세로방향으로 바이트단위 픽셀 표시.
세로로 바이트 단위 표시
* 128 * 64 그래픽 LCD 일때
x = 128 - 1
y = 64 - 1

[예제]

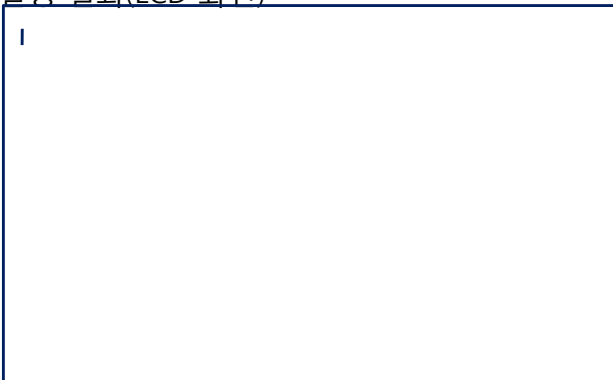
C기본 헤더 선언

```
#include "eng8x16.fnt" // 영문 폰트
#include "han16x16.fnt" // 한글 폰트
#include "hconvtbl.inc" // 완성형 -> 조합형 변환 테이블

void main(void)
{
    cpu_setup(); // CPU 초기화
    GLCD_Init(); // 그래픽 LCD 초기화

    // 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)



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.

GLCD_Dread

현재 커저 위치의 1바이트 데이터를 취득

함 수 : unsigned char GLCD_Dread(int port)
인 수 : int port // LCD_CS1/CS2중 선택
반 환 : 00 - ffh

설 명 : 현재 커저 위치의 1바이트 데이터를 취득.

[예제]

.
C기본 헤더 선언

.
#include "eng8x16.fnt" // 영문 폰트
#include "han16x16.fnt" // 한글 폰트
#include "hconvtbl.inc" // 완성형 -> 조합형 변환 테이블

void main(void)

{
unsigned char uch;

cpu_setup(); // CPU 초기화
GLCD_Init(); // 그래픽 LCD 초기화

// LCD TEST Code..
uch = GLCD_Dread(_LCD_CS1); // 데이터 리드(왼쪽 판넬)
printf("byte read : %02xWn",uch); // 데이터 표시
while(1)
{
사용자 프로그램
}
}

LCD_Init

Graphic Driver IC를 초기화 한다.

함 수 : void Lcd_Init(void)

인 수 : void

반 환 : void

설 명 : Graphic Driver IC를 초기화 한다.

[예제]

.
C기본 헤더 선언

```
.  
  
#include "eng8x16.fnt"           // 영문 폰트  
#include "han16x16.fnt"        // 한글 폰트  
#include "hconvtbl.inc"        // 완성형 -> 조합형 변환 테이블  
  
void main(void)  
{  
    cpu_setup();                // CPU 초기화  
    GLCD_Init();                // 그래픽 LCD 초기화  
  
    while(1)  
    {  
        사용자 프로그램  
    }  
}
```


GLCD_Command

Graphic Driver IC에 command 데이터를 써넣는다.

함 수 void GLCD_Command(int port,unsigned char uch)

인 수 : int port // LCD_CS1/CS2중 선택

unsigned char uch // command 데이터

반 환 : void

설 명 Graphic Driver IC에 command 데이터를 써넣는다

GLCD_Dwrite

Graphic LCD에 데이터를 써넣는다.

함 수 : void GLCD_Dwrite(int port,unsigned char uch)

인 수 : int port // LCD_CS1/CS2중 선택

unsigned char uch // command 데이터

반 환 : void

설 명 : Graphic LCD에 데이터를 써넣는다.

GLCD_Busy

Graphic LCD의 Driver IC의 Busy 상태를 읽음.

함 수 : void GLCD_Busy(int port)

인 수 : int port

// LCD_CS1/CS2중 선택

반 환 : void

설 명 : Graphic LCD의 Driver IC의 Busy 상태를 읽음.