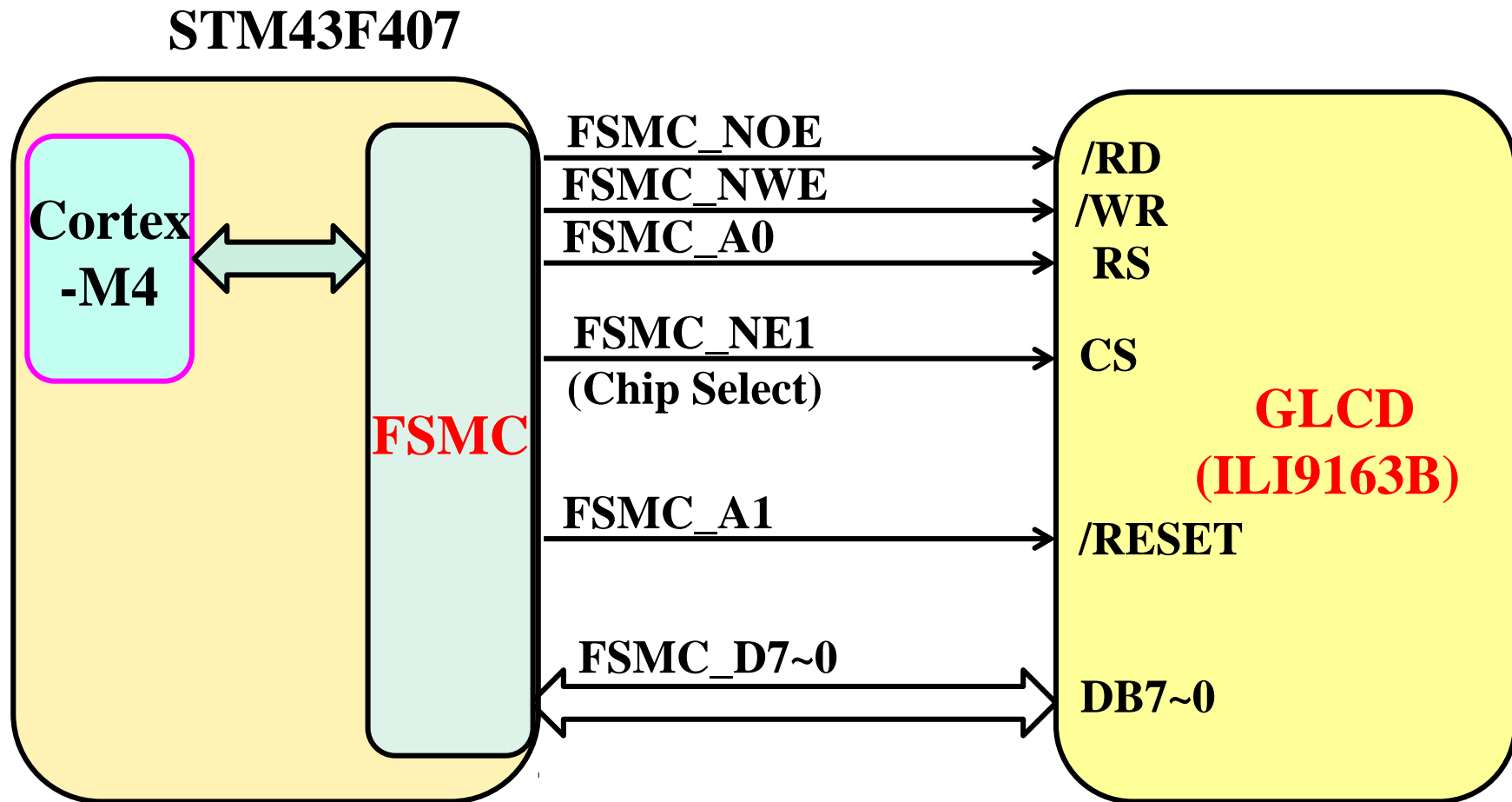


STM32F407과 GLCD의 인터페이스

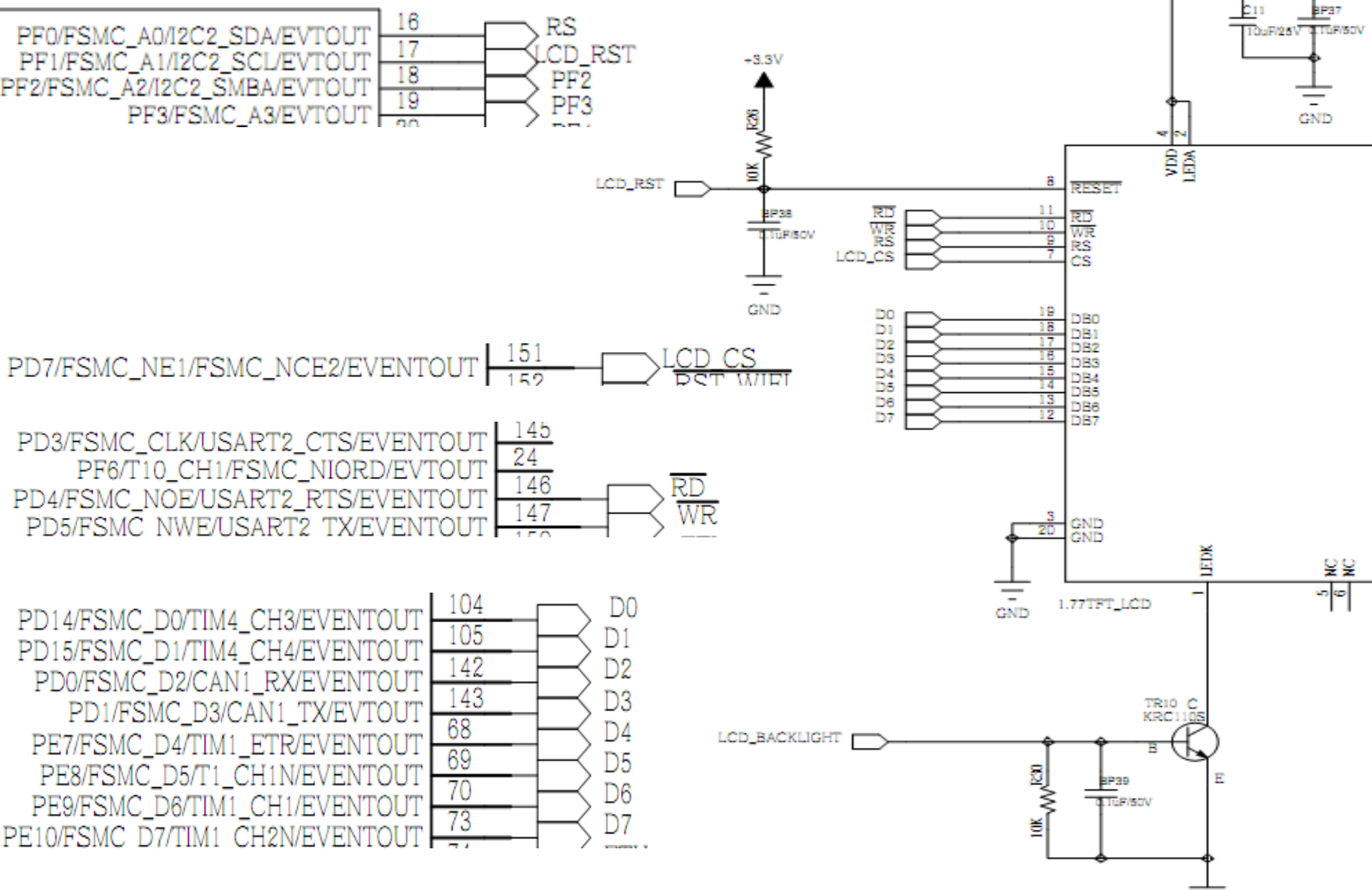


한국산업기술대학교 메카트로닉스공학과
마이크로컴퓨터구조
담당교수: 남윤석

● GLCD 인터페이스 구성: 외부 SRAM 연결방법 사용

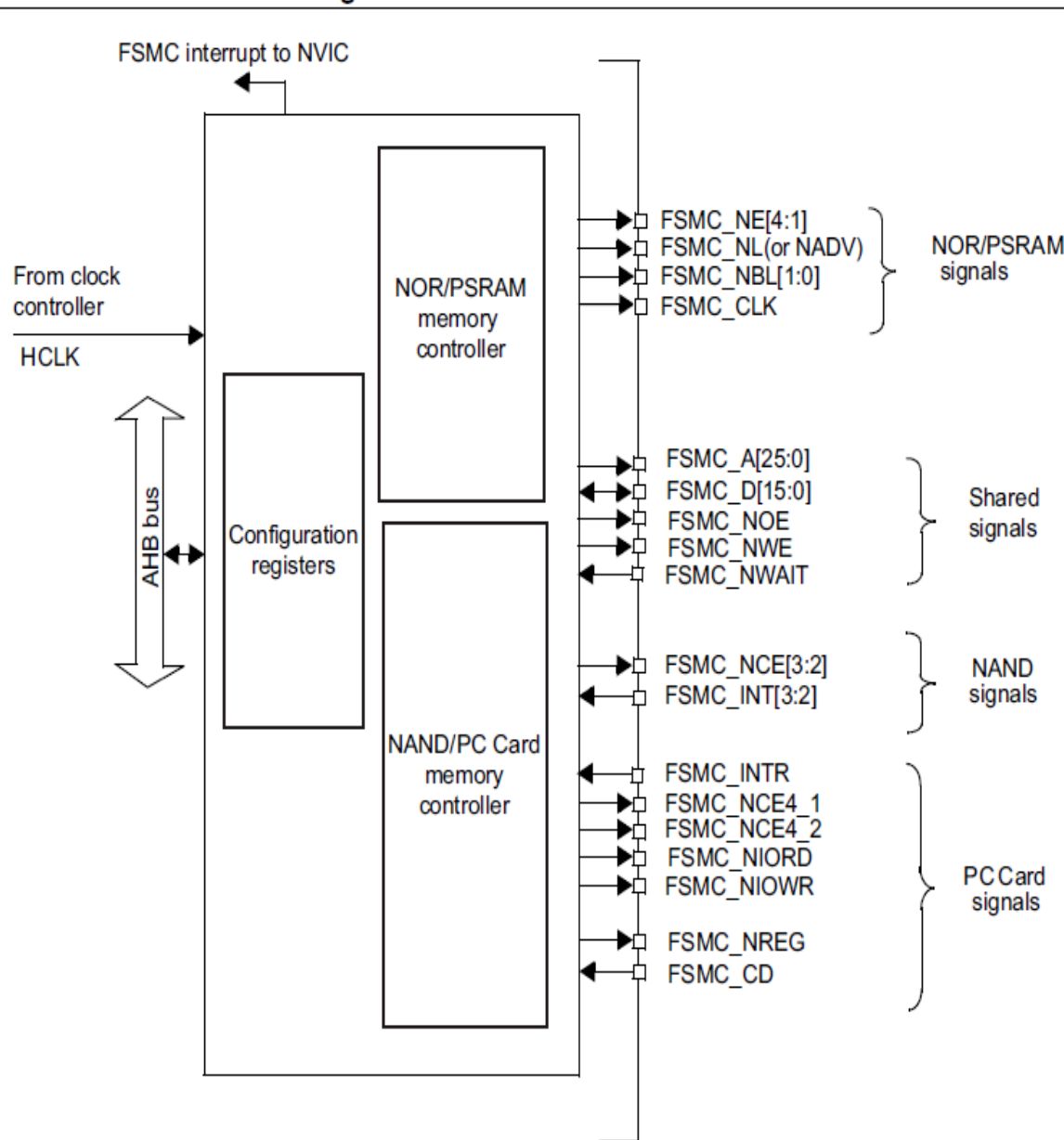


● 실습키트 GLCD 부분 회로도



● STM32F407: FSMC(Flexible static Memory Controller)

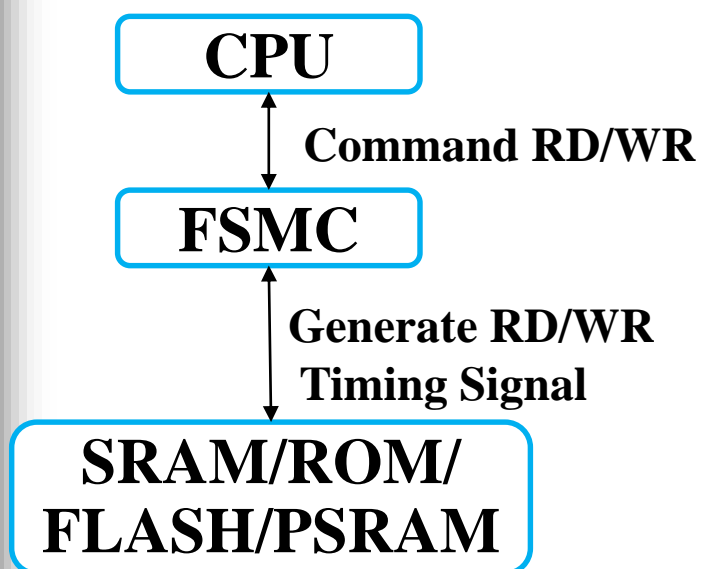
FSMC block diagram



Supported Memory Type

- SRAM
- ROM
- NOR/NAND Flash
- PSRAM(Pseudo SRAM)
- LCD Parallel Interface

Memory Access Structure



*PSRAM : SRAM의 접근 방식과 DRAM의 메모리 셀 구조를 채용한 메모리

● 프로그램에서의 GLCD set-up 과정

ILI
driver
초기화

- ILI_Init
: FSMC(외부 SRAM)이용 GLCD 초기화

GLCD
초기화

- 배경색, 글자색, 폰트 지정

GLCD
초기화면

- 초기 화면 표시

● GLCD 제어용 색 및 함수 선언 (GLCD.h, GLCD.c)

```
#define RGB_BLACK    GET_RGB(0, 0, 0)
#define RGB_WHITE    GET_RGB(0xFF, 0xFF, 0xFF)
#define RGB_RED      GET_RGB(255, 0, 0)
#define RGB_GREEN    GET_RGB(0, 255, 0)
#define RGB_YELLOW   GET_RGB(255, 255, 0)
#define RGB_BLUE     GET_RGB(0, 0, 255)
#define RGB_GRAY     GET_RGB(240, 240, 240)

void    LCD_Init(void);
void    RestoreOldColor();
void    SaveOldColor();

void    LCD_SetBackColor(UINT32 _Color); //화면배경색 지정
UINT16  LCD_GetBackColor(void); //현재의 화면배경색 리턴
void    LCD_SetTextColor(UINT32 _Color); //글자색 지정
UINT16  LCD_GetTextColor(void); //현재의 글자색 리턴
void    LCD_SetPenColor(UINT32 Color); //펜색 지정
UINT16  LCD_GetPenColor(void); //현재의 펜색 리턴
void    LCD_SetBrushColor(UINT32 Color); //브러쉬색 지정
UINT16  LCD_GetBrushColor(void); //현재의 브러쉬색 리턴
```

● GLCD 제어용 색 및 함수 선언 (GLCD.h, GLCD.c)

```
void LCD_SetFont(gl_sFONT *pFont); //폰트 지정  
gl_sFONT* LCD_GetFont(void); //현재의 폰트 리턴
```

(0,0)

```
void LCD_Clear(UINT32 Color); // 화면 클리어
```

```
void LCD_DrawChar(UINT16 x, UINT16 y, char ch);
```

```
void LCD_DrawText(UINT16 x, UINT16 y, char *str);
```

```
void LCD_DrawHorLine(UINT16 x, UINT16 y, UINT16 width);
```

```
void LCD_DrawVerLine(UINT16 x, UINT16 y, UINT16 height);
```

```
void LCD_DrawRectangle(UINT16 x, UINT16 y, UINT16 width, UINT16  
height);
```

```
void LCD_DrawFillRect(UINT16 x, UINT16 y, UINT16 width, UINT16 height);
```

```
//void LCD_BitmapDisplay(uint16_t x, uint16_t y, Bitmap *bitmap);
```

```
void LCD_DrawLine(UINT16 x1, UINT16 y1, UINT16 x2, UINT16 y2);
```

```
void LCD_DrawPixel(UINT16 x, UINT16 y);
```

```
void LCD_ClearLine(UINT16 Row);
```

```
void LCD_DisplayChar(UINT16 Row, UINT16 Col, char ch);
```

```
void LCD_DisplayText(UINT16 Row, UINT16 Col, char *str);
```

● GLCD 제어용 주요 functions 설명 (GLCD.h, GLCD.c)

```
void LCD_Init(void) // GLCD 초기화 루틴
{
    LCD_BkColor = RGB_BLACK; // 바탕색: 검정
    LCD_BrushColor = LCD_BkColor; // 브러시색: 검정
    LCD_TextColor = RGB_WHITE; // 글색: 흰
    LCD_PenColor = LCD_TextColor; // 펜색: 흰
    LCD_pFont = &Gulim7; // 초기폰트: 굴림체(크기 7)

    ILI_Init(); // GLCD driver 초기화
    LCD_Clear(RGB_WHITE); // 초기에 흰색으로 클리어
}
```

```
void LCD_Clear(UINT32 Color) // 주어진 색으로 LCD 화면 클리어
void LCD_SetFont(gl_sFONT *pFont) // 문자표시 주어진 폰트 사용
void LCD_SetBackColor(UINT32 _Color) //화면배경색 지정
void LCD_SetTextColor(UINT32 _Color) // 글자색 지정
```

```
void LCD_DisplayChar(UINT16 Row, UINT16 Col, char ch) //지정
된 위치(Col:x, Row:y)에 문자(ch) 표시
void LCD_DisplayText(UINT16 Row, UINT16 Col, char *str) // 지정
된 위치(Col:x, Row:y)에 문자열(*str) 표시
```


● GLCD 제어용 주요 functions 설명 (GLCD.h, GLCD.c)

void LCD_DrawChar(UINT16 x, UINT16 y, char ch)

// (x,y)에 문자(ch) 그리기

void LCD_DrawText(UINT16 x, UINT16 y, char *str)

// (x,y)에 문자열(*str) 그리기

void LCD_DrawHorLine(UINT16 x, UINT16 y, UINT16 width)

// (x,y) 점을 기점으로 width의 폭을 가진 수평선 그리기

void LCD_DrawVerLine(UINT16 x, UINT16 y, UINT16 height)

// (x,y) 점을 기점으로 height의 높이를 가진 수직선 그리기

void LCD_DrawRectangle(UINT16 x, UINT16 y, UINT16 width, UINT16 height)

// (x,y)점을 기점으로 폭(width), 높이(height)의 직사각형 그리기

void LCD_DrawFillRect(UINT16 x, UINT16 y, UINT16 width, UINT16 height)

// (x,y)점을 기점으로 폭(width), 높이(height)의 직사각형의 내부를 채우기

void LCD_DrawLine(UINT16 x1, UINT16 y1, UINT16 x2, UINT16 y2)

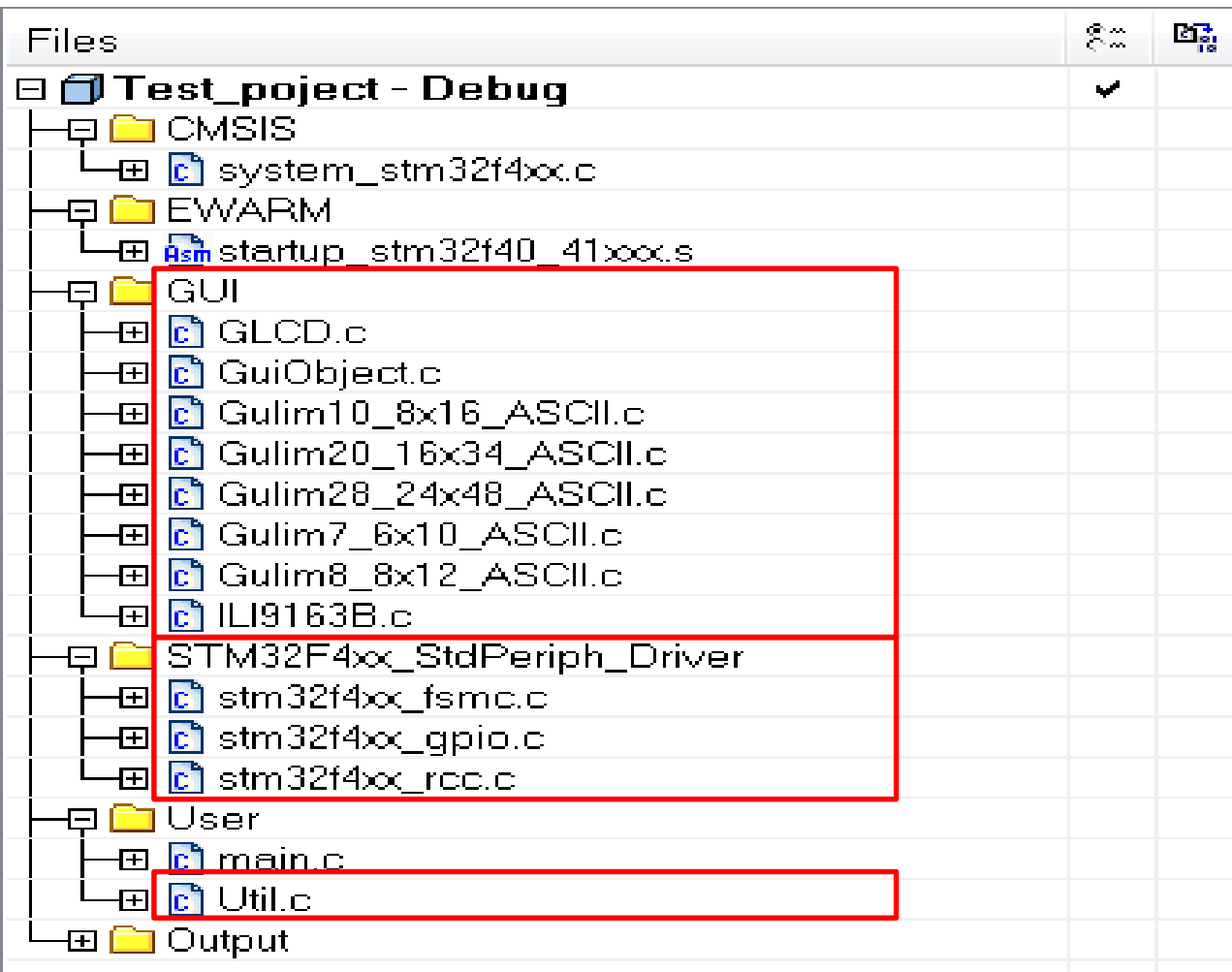
// (x1,y1)점에서 (x2,y2)점까지의 선 그리기

void LCD_DrawPixel(UINT16 x, UINT16 y)

// (x,y)점에 픽셀 표시

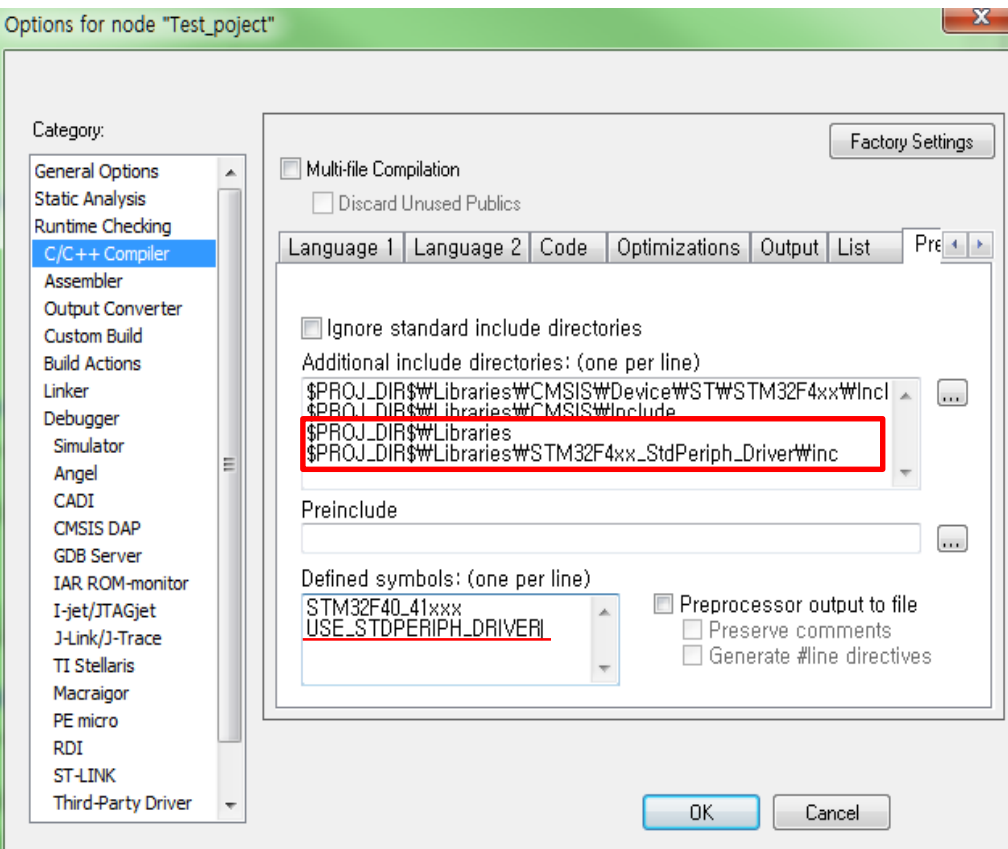
● GLCD 사용시 프로젝트 환경구성

- e-class에서 새로운 Libraries 디렉토리 다운로드!
- 다음화면과 같이 그룹 및 파일을 추가



● GLCD 사용시 프로젝트 환경구성

- 다음화면과 같이 **Option > C/C++ Compiler > Pre-processor** 에서 빨간 박스 추가



- 추가 경로

`$PROJ_DIR$\Libraries\CMSIS\Device\ST\STM32F4xx\Include`

`$PROJ_DIR$\Libraries\CMSIS\Include`

`$PROJ_DIR$\Libraries`

`$PROJ_DIR$\Libraries\STM32F4xx_StdPeriph_Driver\inc`

- 추가 심볼

`STM32F40_41xxx`

`USE_STDPERIPH_DRIVER`

(STM32F4xx_StdPeriph_Driver 에 존재하는 소스들을 사용하기 위해 추가된 심볼)