

## 1. CCS3.3 DSP/BIOS PRD(periodic fuction manager) 생성

### 1. 디렉토리 구성

..Wcmd	: Linker 커맨드 파일
..DSP2833x_headers	: Chip관련 헤더 파일 및 헤더용 Linker 커맨드 파일
..Winclude	: 사용자 인클루드 파일
..Wtestprj_2	: 사용자 프로젝트 파일 및 실행 파일(.HEX)
..Wtestsrc_2	: 사용자 소스 파일

### 2. 디렉토리 설명

다른 디렉토리는 기존 Task 생성 예제 파일을 그대로 사용하고 testprj\_2, testsrc\_2 디렉토리만 복사해서 사용 하거나 기존의 testprj\_1, testsrs\_1 의 소스만 아래 진행에 따라 추가 수정 하여 사용 하시면 됩니다.

### 3. Setup CCStudio v3.3을 실행 합니다.

프로젝트 Open 사용법은 Task 생성 예제를 참조 하시기 바랍니다.

#### 4. 소스코드 설명(Main\_Bios.c)

```
#include "DSP2833x_Device.h" <- DSP 초기화 및 설정 관련
#include "F28335_example.h" <- 사용자 외부 함수, 변수, 정의 관리

void main(void)
{
    InitSysCtrl();           <- CPU 클럭 설정((30*10) / 2 = 150M)
    InitPieCtrl();          <- 인터럽트 관련 초기화
    InitWatchdog();         <- watch-dog 설정 및 초기화
    InitGpio();             <- CPU I/O 설정(IN,OUT,기본기능..) _EX_BUS_ON정의에 따라 외부 버스 ON
    InitXintf();           <- 내부 주변 디바이스 클럭 설정 및 외부 버스 타이밍 설정

    ** DSP/BIOS 관련 설정 **
    #ifdef EXAMPLE_FLASH
        memcpy(&secureRamFuncs_runstart,
               &secureRamFuncs_loadstart,&secureRamFuncs_loadend - &secureRamFuncs_loadstart);
        InitFlash();
    #endif
    asm(" EALLOW");         <- Enable EALLOW protected register access
    GpioCtrlRegs.GPBMUX1.bit.GPIO32 = 0; <- GPIO032 GPIO
    GpioCtrlRegs.GPBDIR.bit.GPIO32 = 1; <- GPIO32 output
    GpioDataRegs.GPBSET.bit.GPIO32 = 1; <- GPIO32 pin is set to 1
    asm(" EDIS");           // Disable EALLOW protected register access

    ** DSP/BIOS에서 TINT2,DLOGINT를 사용 하므로 BIOS사용 인터럽트 허가 **
    SetDBGIER(IER | 0x6000); <- Enable everything in IER, plus TINT2 and DLOGINT
    *(volatile unsigned int *)0x00000C14 |= 0x0C00;<- Set TIMER2 FREE=SOFT=1

    ** 아래 main()를 종료 하면 DSP/BIOS가 동작.. **

}

void UserInit(void){       <- 이 함수는 리셋시 DSP/BIOS 초기화 부분에서 한번 수행 후
                           DSP/BIOS관련 및 사용자 초기화 함수 추가
}

void task1_proc(void){     <- 이 함수는 스케줄러에 관리 되는 TASK

}
```

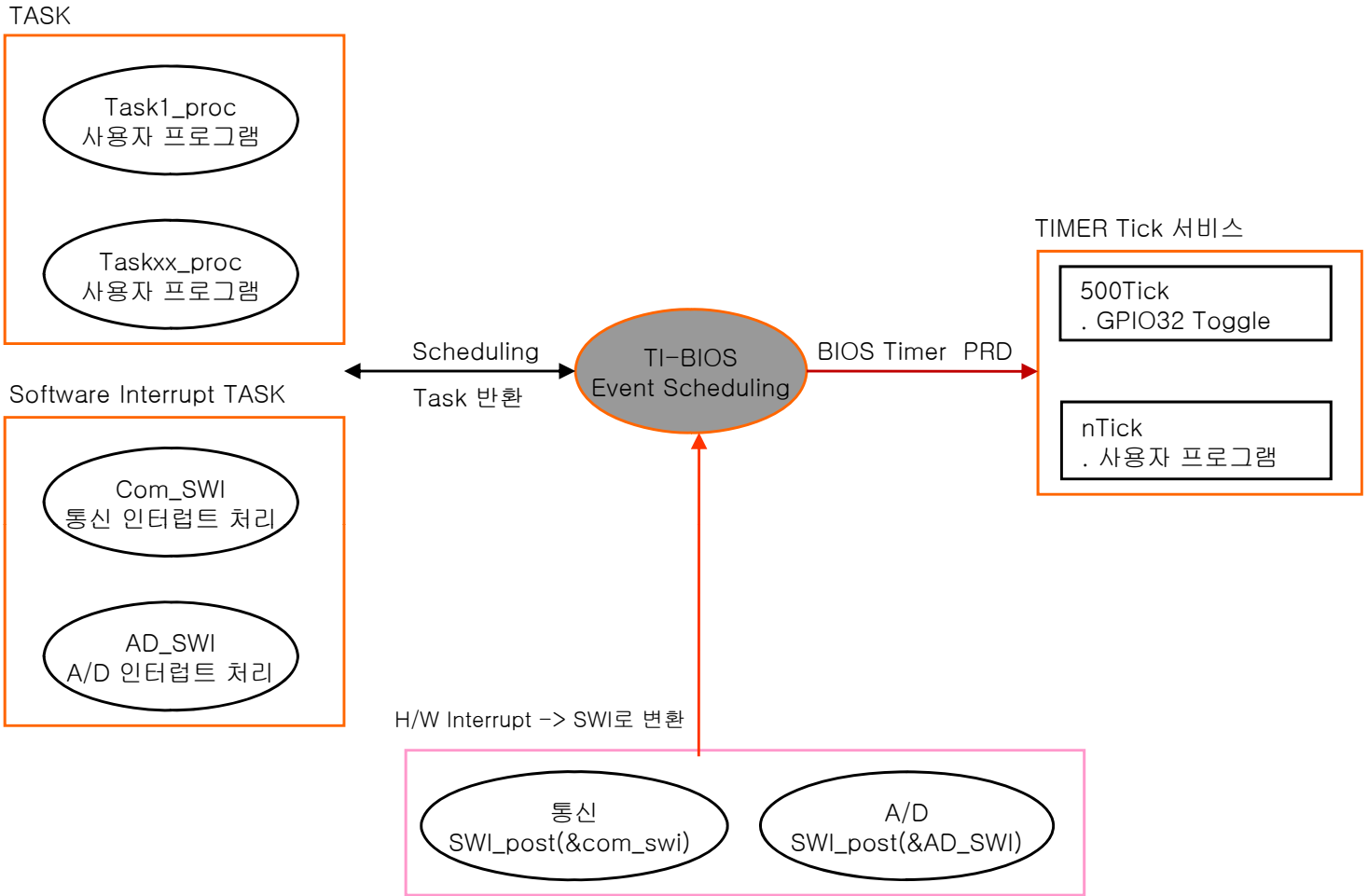
#### \* GPIO 설명

DSP는 특수 기능 레지스터에 접근 하기 위해서는 EALLOW Bit를 Set후 접근후 사용 한다.  
사용 후 EALLOW Bit를 Clear한다.

- GPIO 필터 설정 레지스터(입력시 몇 클럭 동안 유지시 입력핀을 유효 시킬 것 인지 사용할때)  
GPxCTRL : GPIO에 Qualification 를 사용할 때 CLK 종류 지정  
GPxQSELn : GPIO에 Qualification 를 사용할 때 값 설정
- GPIO 설정 레지스터  
GPxMUXn : 0-3값으로 GPIO 또는 특정 기능 핀으로 I/O핀을 정의  
GPxDIRn : GPIO핀을 입력(0) 또는 출력(1) 으로 설정  
GPxPAUn : GPIO핀에 PULL UP 저항을 설정
- GPIO 데이터 레지스터  
GPxDATn : GPIO 입출력 데이터 레지스터  
GPxSETn : 해당 비트가 1이면 GPIO 핀을 High로 유지 시킴  
GPxCLEARn : 해당 비트가 1이면 GPIO 핀을 Low로 유지 시킴  
GPxTOGGLEn: GPIO 해당 핀을 TOGGLE 함.

## - CCS3.3 DSP/BIOS PRD 생성

\* PRD란 스케줄링 에서 Clock Manager의 기본 클럭을 사용하여 주기적인 시간 마다 발생하는 Time Task로서 이 루틴은 타이머 인터럽트의 일종이므로 소스 코딩시 정제되는 코드나 너무 긴 시간이 소요 되는 코딩은 자제 해 주시기 바랍니다.

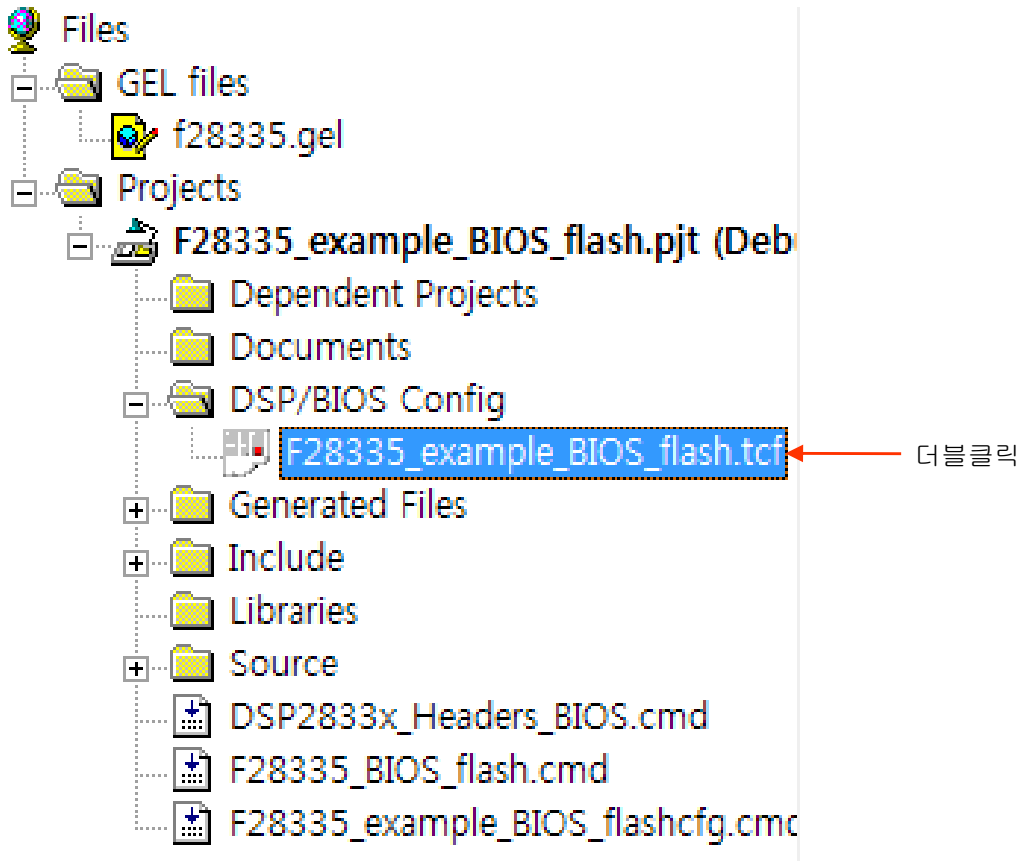


1. Main\_Bios.c 를 open후 아래 소스코드를 입력 후 저장한다.

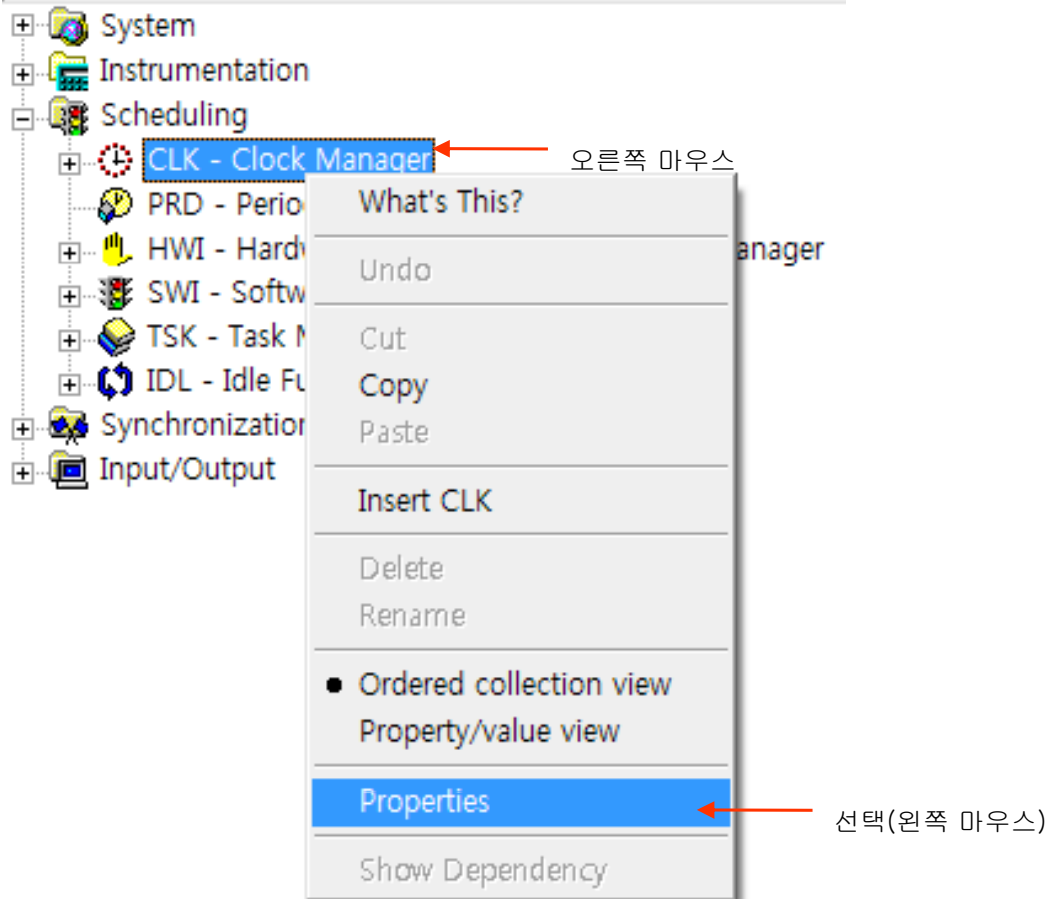
아래 함수에 있는 실행 코드는 C소스의 타이머 인터럽트와 유사 하므로 최대한 간단 명료 하게 코딩 하고 while() 문처럼 무한 루프에 빠지는 코드는 작성 하지 않는것이 원칙 이다 다른 스케줄러에 영향을 줌.

```
void Io_Blink(void)                <- PRD 스킴줄 함수 명
{
    GpioDataRegs.GPBTOGGLE.bit.GPIO32 = 1; <- GPIO32 토글
}
```

2. DSP/BIOS Config->\*.tcf 를 실행 한다.



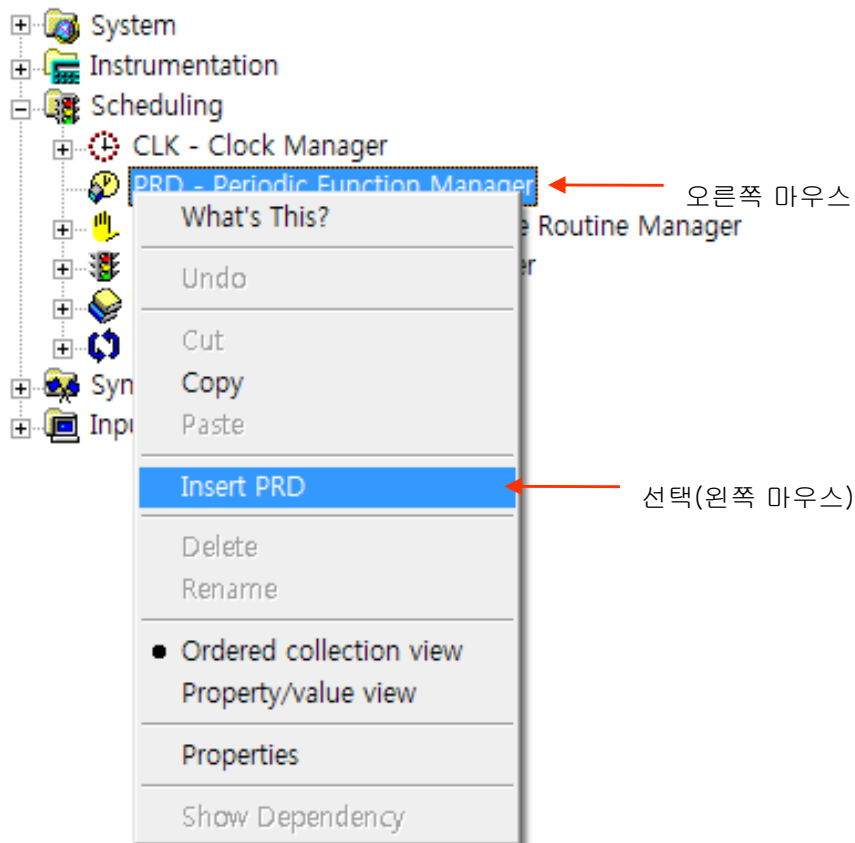
3. 스케줄링 기본 스케줄 클럭을 확인 한다.



4. DSP/BIOS에 설정된 기본 클럭은 1ms 이다.  
이 클럭은 아래 PRD(Periodic Function Manager)에 기본 클럭(1 Tick)으로 사용 된다.



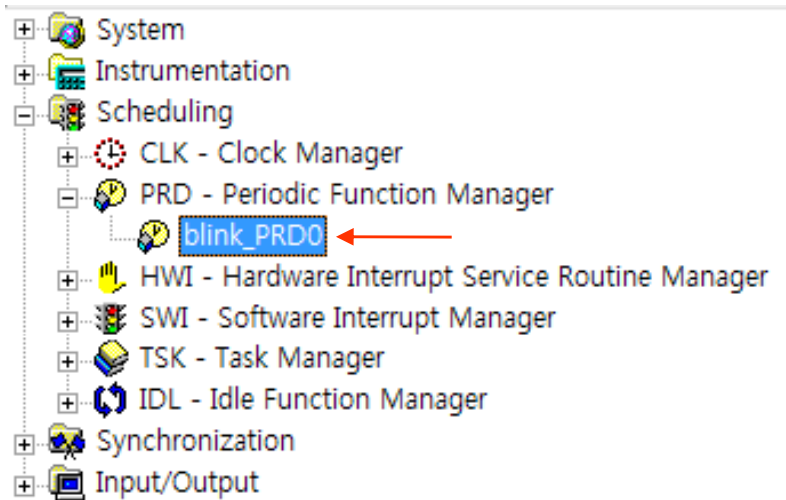
5. 스케줄링 PRD(Periodic Function Manager) 에서 아래와 같이 PRD를 생성 한다.



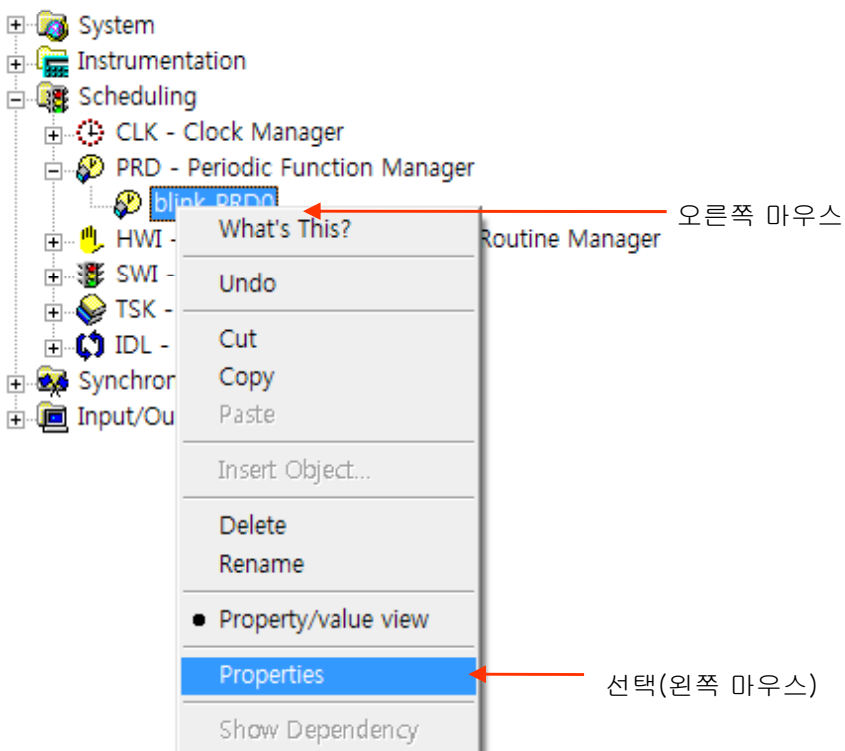
\* PRD TASK 관리 명을 입력 한다



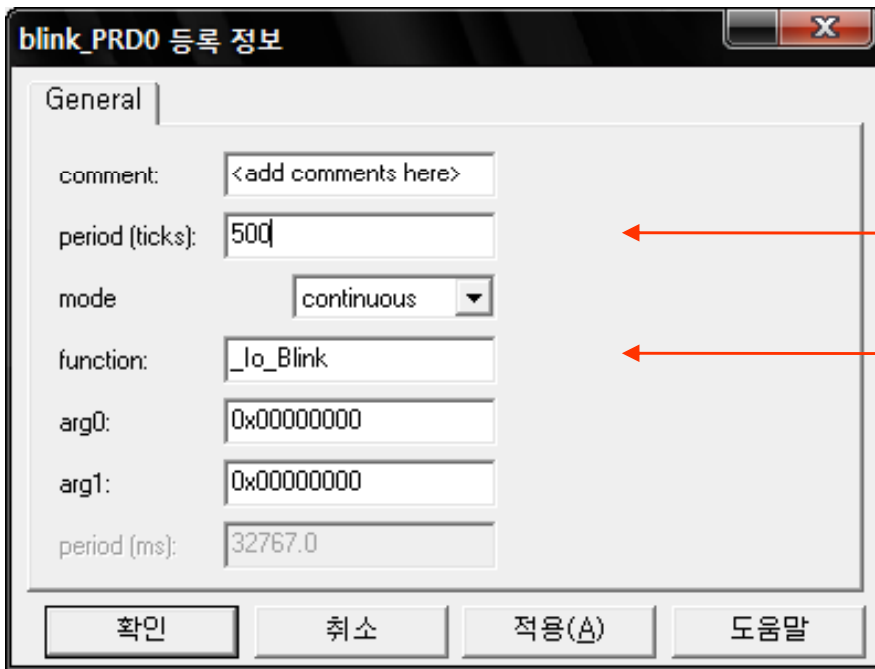
\* Task 관리자 생성 확인



\* 생성한 PRDTask에 사용자 환경을 설정 한다.(blink\_PRD0 선택후 오른쪽 버튼)



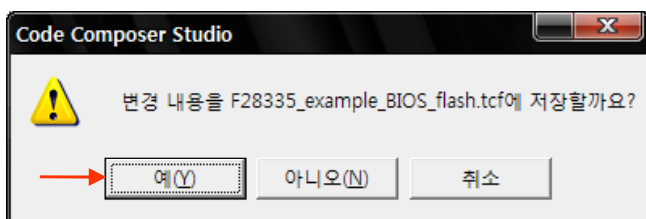
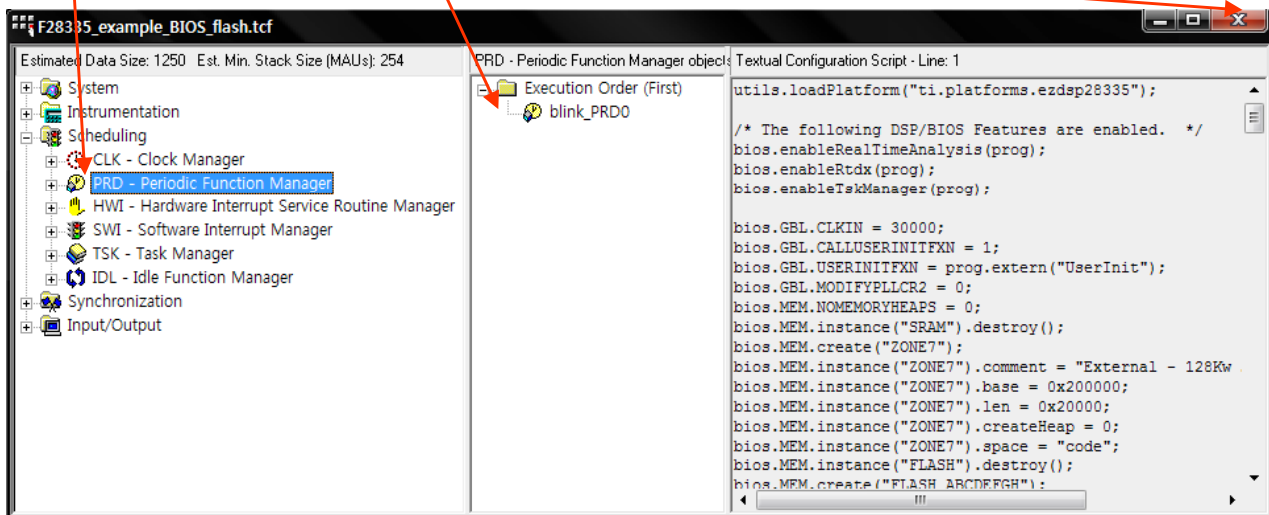
\* General 에서 기본 정보를 설정 한다.



← 실행할 Tick수(1ms\*500 = 500ms)

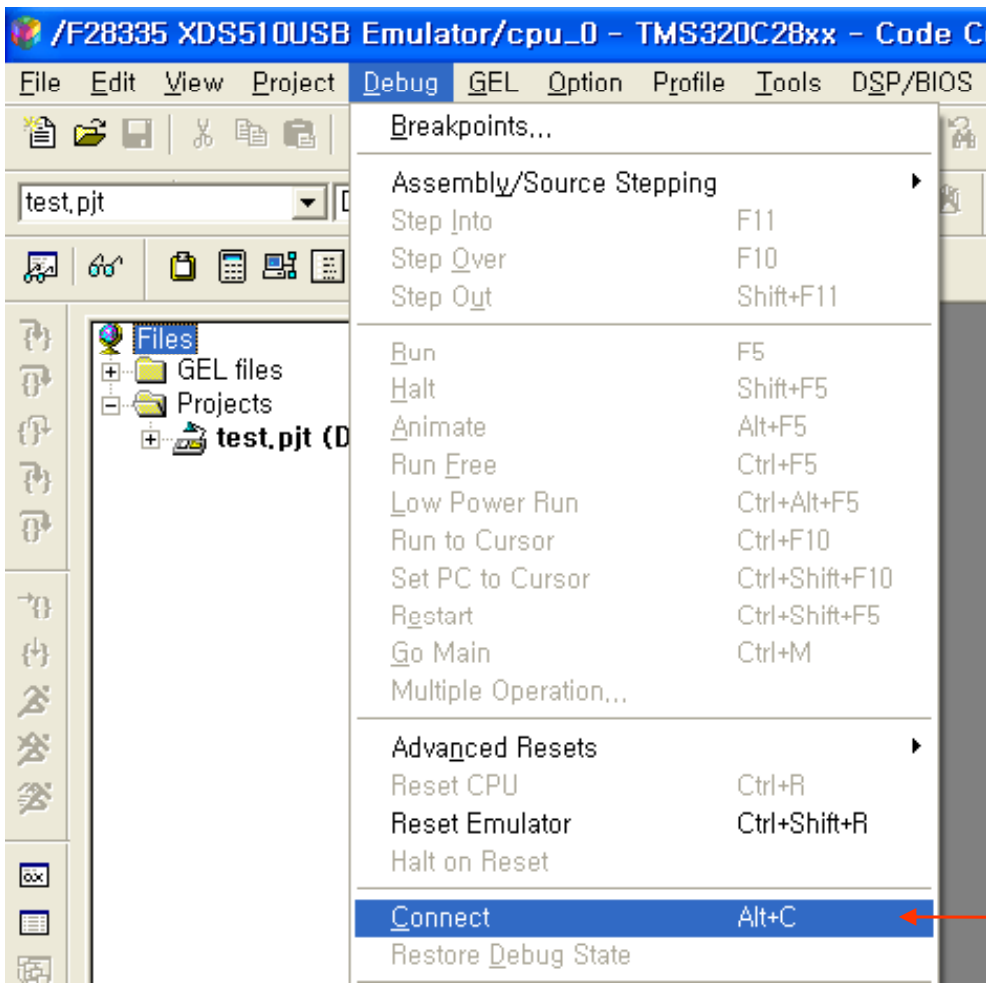
← 실제 실행할 함수명 앞에 \_를 붙임.

\* Task Manager에서 생성 된 PRDTask를 확인후 \*.tcf 파일을 종료 합니다.



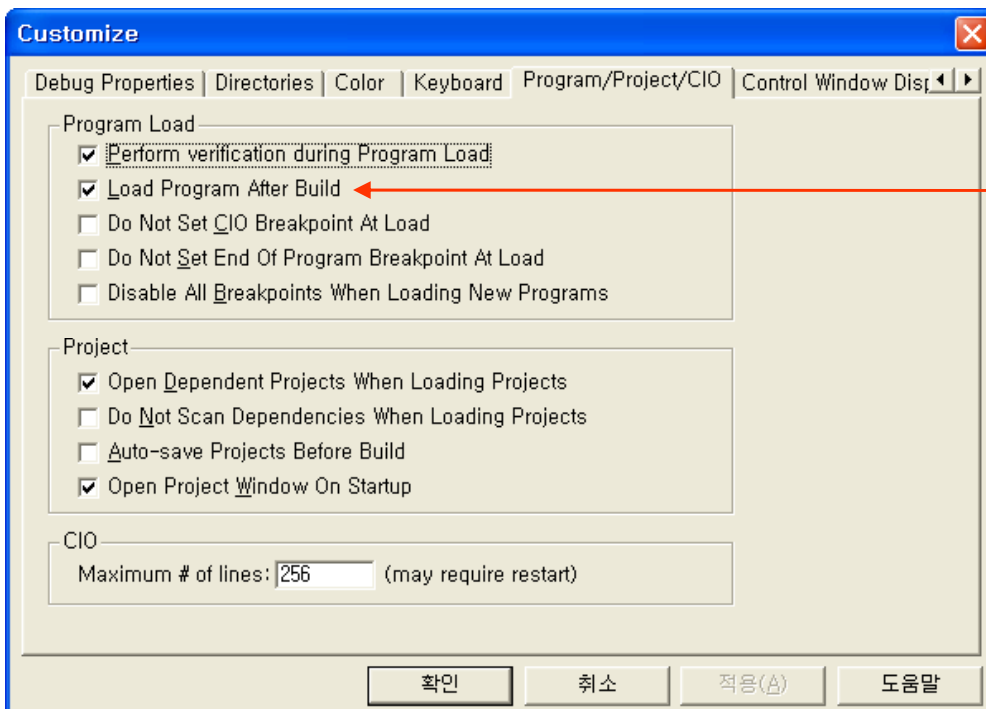
## - CCS3.3 DSP/BIOS PRD TASK 실행

1. JTAG 및 에뮬레이터를 연결 합니다.



선택후  
마우스 왼쪽 버튼 클릭

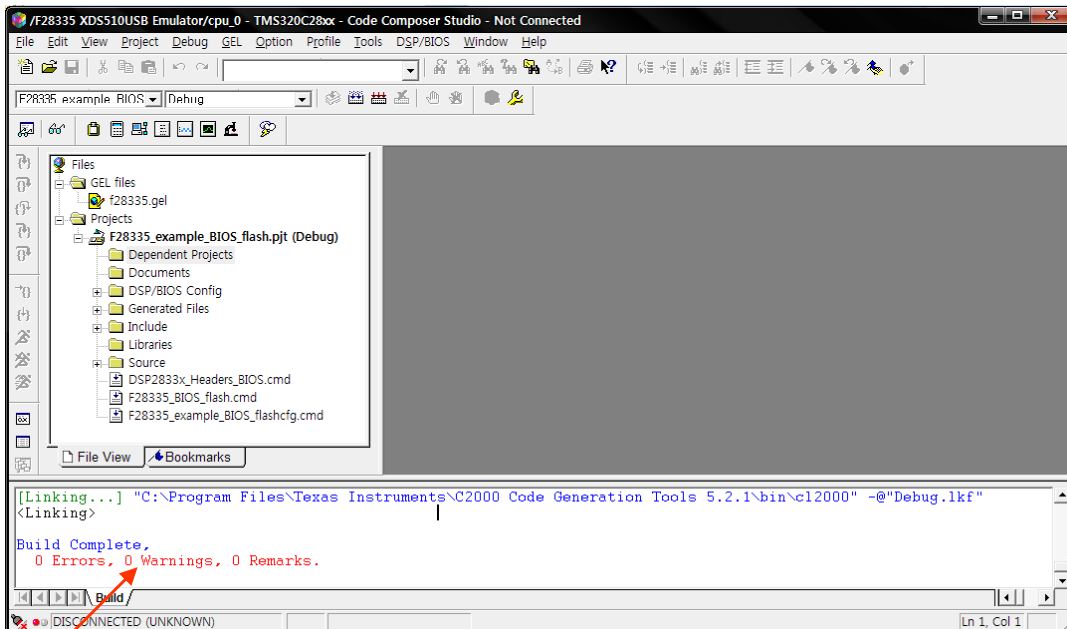
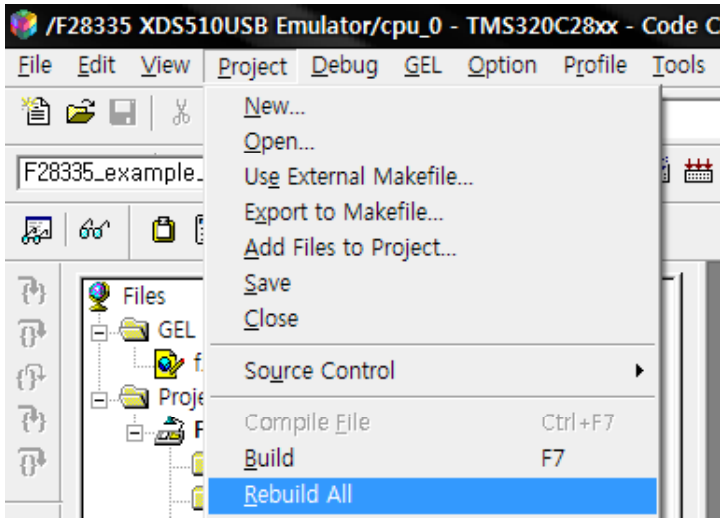
2. 내부럼 으로 프로그램을 실행할 경우 아래와 같이 설정 합니다.(Option->Customize)



체크

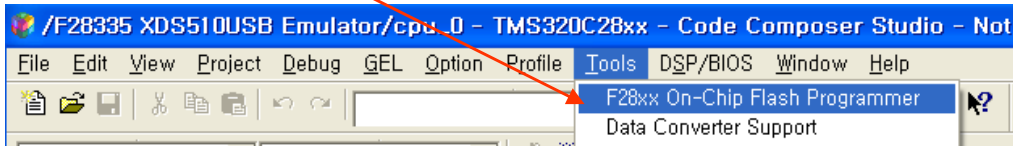


### 3. 컴파일 하기(Project->Rebuild All)

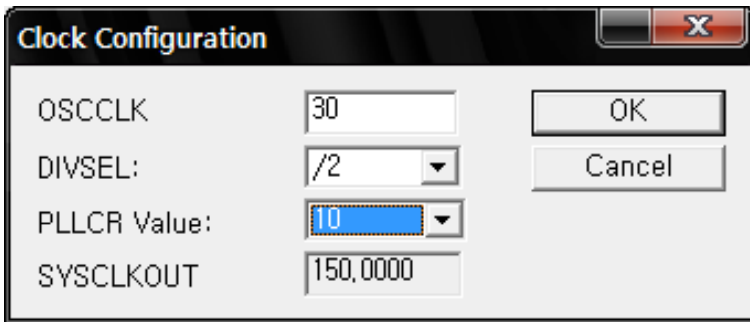


에러 확인

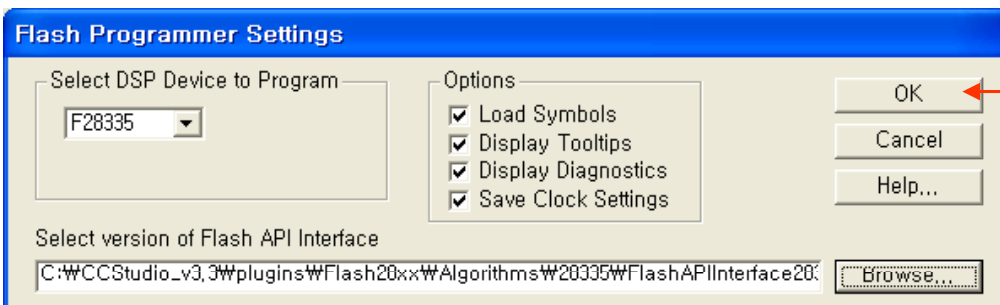
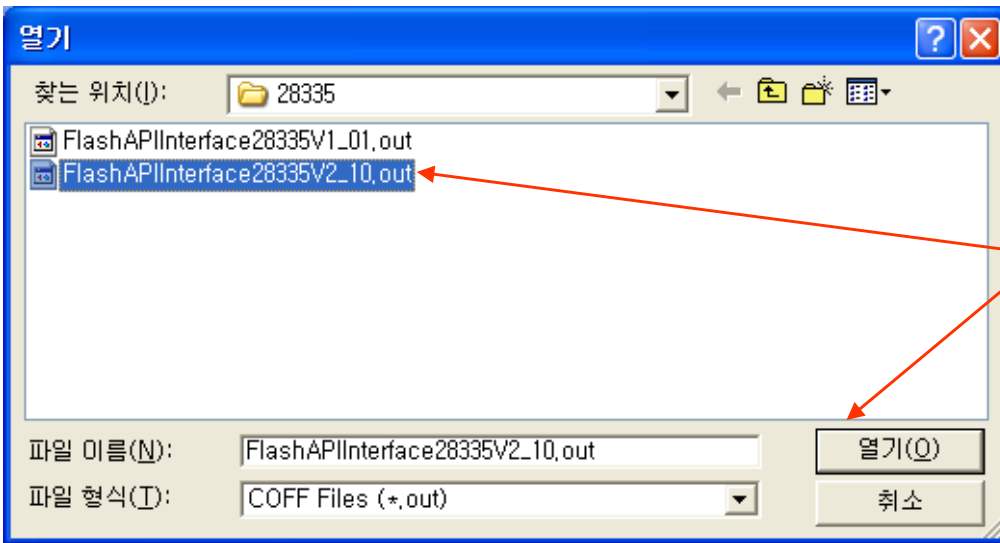
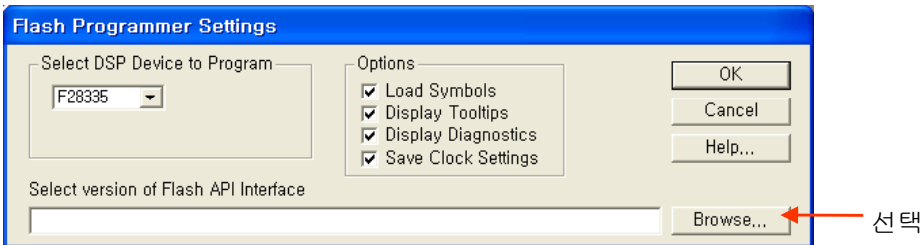
#### 4. FLASH에 프로그램 하기

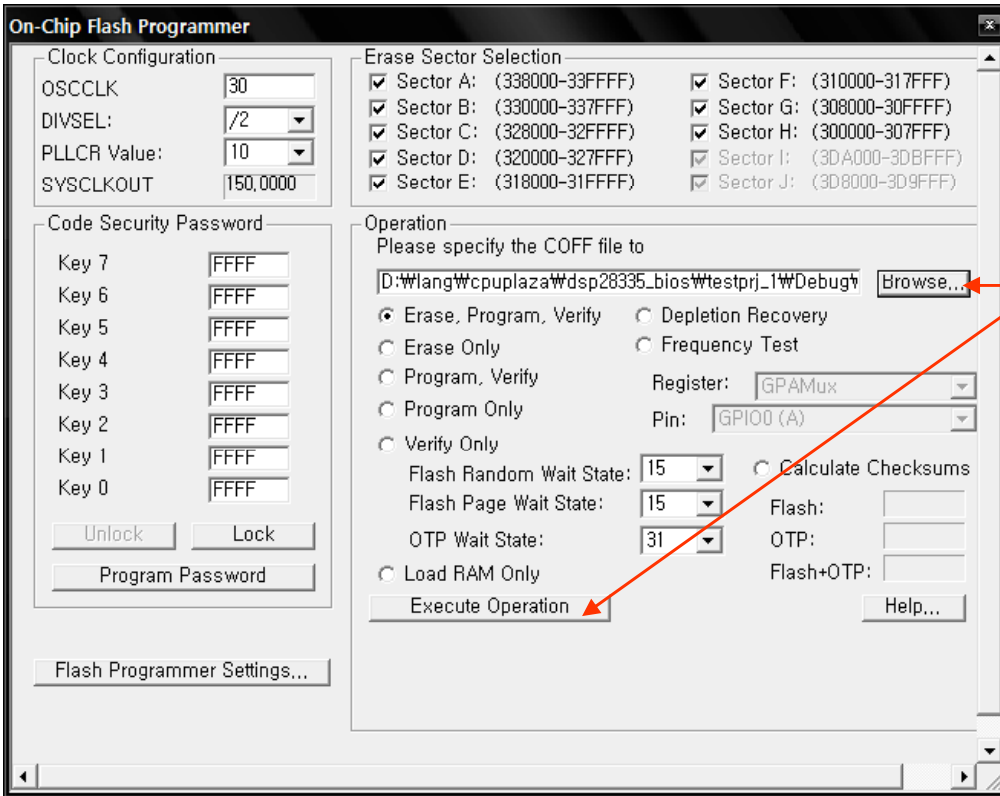


\* 아래 CLOCK 설정 메뉴를 사용자에게 맞게 설정 합니다.



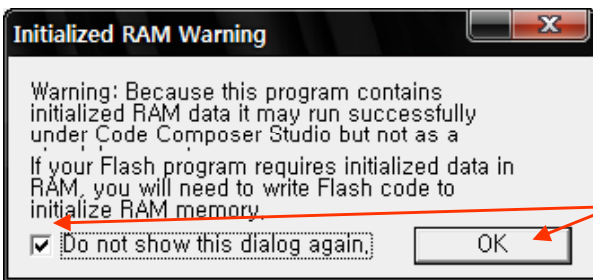
\* API Interface 파일을 등록 합니다.



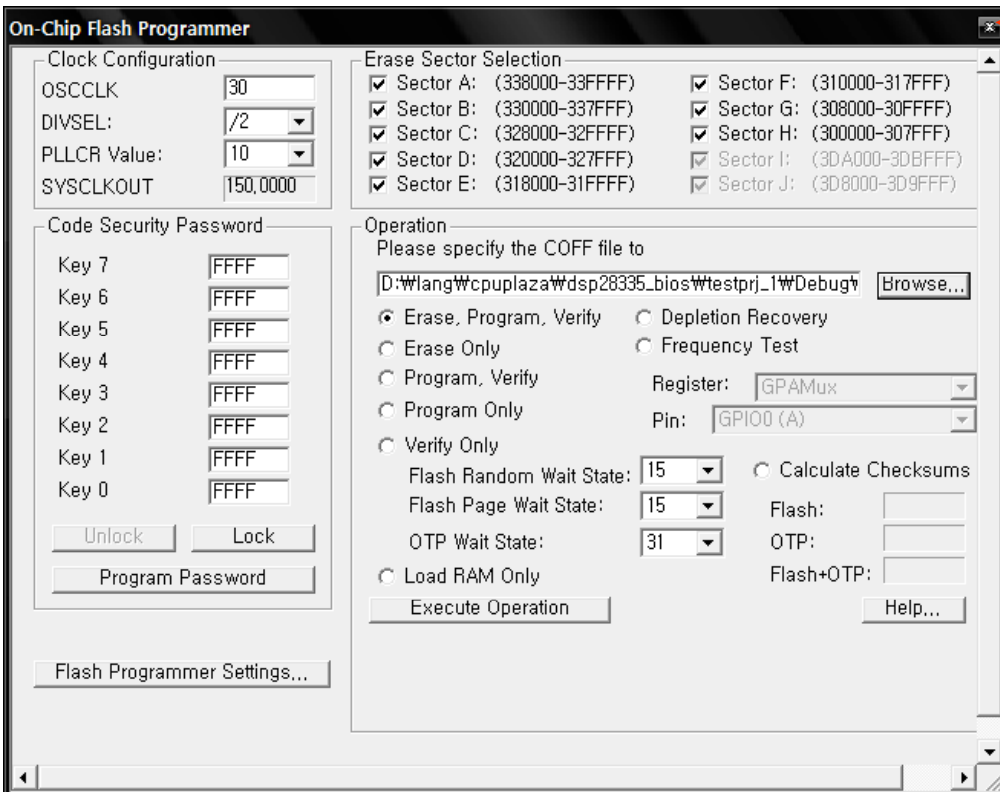


Browse.. 창에서 파일을 선택후 Execute Operation 탭을 실행합니다.

\* TI 실행 파일은 \*.OUT로 현재 작업 디렉토리 ..\wdebug\ 에 있습니다.

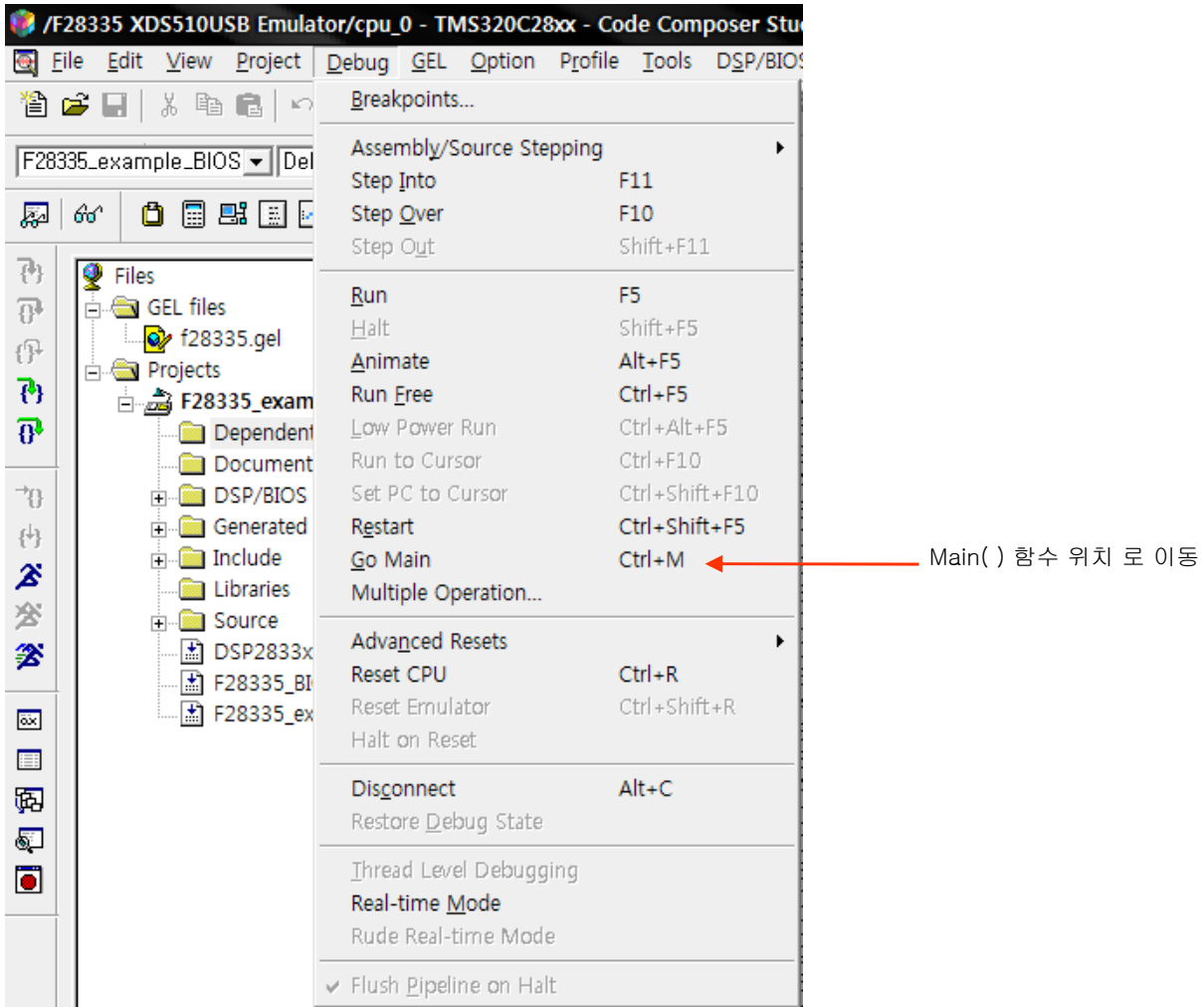


체크후 확인



닫음

5. 프로그램을 로딩후 Debug 탭에서 Go Main 기능을 실행 합니다.



6. Main\_Bios.c의 Io\_Blink() 에 break mode를 설정 합니다.

```
// ----- PRD0 스케줄러 -----  
// [인수] void  
// [참고] DSP/BIOS 500ms 스케줄 타임  
// -----  
void Io_Blink(void)  
{  
    GpioDataRegs.GPBToggle.bit.GPIO32 = 1;    // Toggle the pin  
}
```

1. 커서를 위치 시킨후 F9키를 누른다.(한번더 누르면 삭제)
2. F5를 누르면 노란 화살표가 나오며 정지 한다.  
그러면 현재 PRD Task 스케줄링 정상.
3. RUN 시킨 후 GPIO32 핀을 스코프로 관찰 한다.

- F5 : Debug->Run
- F9 : Debug->Break Point Toggle