(ARM,Cortex-M3, STM32F207) Mango-M32F2, Project 구성 및 빌드 (최종) 및 소스

http://www.mangoboard.com/
http://cafe.naver.com/embeddedcrazyboys
Crazy Embedded Laboratory

Document History

Revision	Date	Change note

1. (ARM,Cortex-M3, STM32F207) Mango-M32F2, Project 구성 및 빌드 (최종) 및 소스

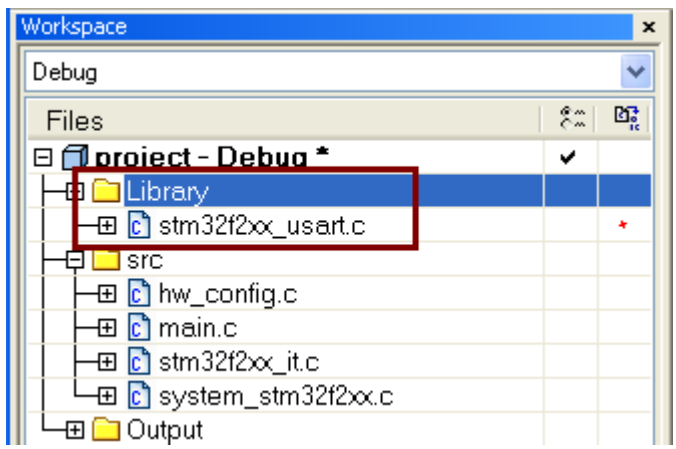
전체공개 2012.04.12 20:22 | 삭제

푸우(yhoh) 카페스탭

http://cafe.naver.com/embeddedcrazyboys/17650 주소복사

첨부파일(1)

이제 링크에러를 해결합니다.



uart와 관련한 소스 파일을 포함하고 빌드합니다.

```
Warning[Pe223]: function "assert_param" declared implicitly D:\Wk.Src\Mango-M32F2\Libraries\STM32F2xx_StdPeriph_Driver\src\stm32f2xx_usart.c 181
```

이것은 물론 에러는 아니지만 포함은 시키는 것이 맞을 것입니다.

stm32f2xx_conf.h (src)

```
/* Uncomment the line below to expanse the "assert_param" macro in the
  Standard Peripheral Library drivers code */
#define USE_FULL_ASSERT    1
```

USE_FULL_ASSERT가 정의되도록 변경합니다.

하지만 이를 변경하고 난 후에도 여전히 warning은 사라지지 않습니다.

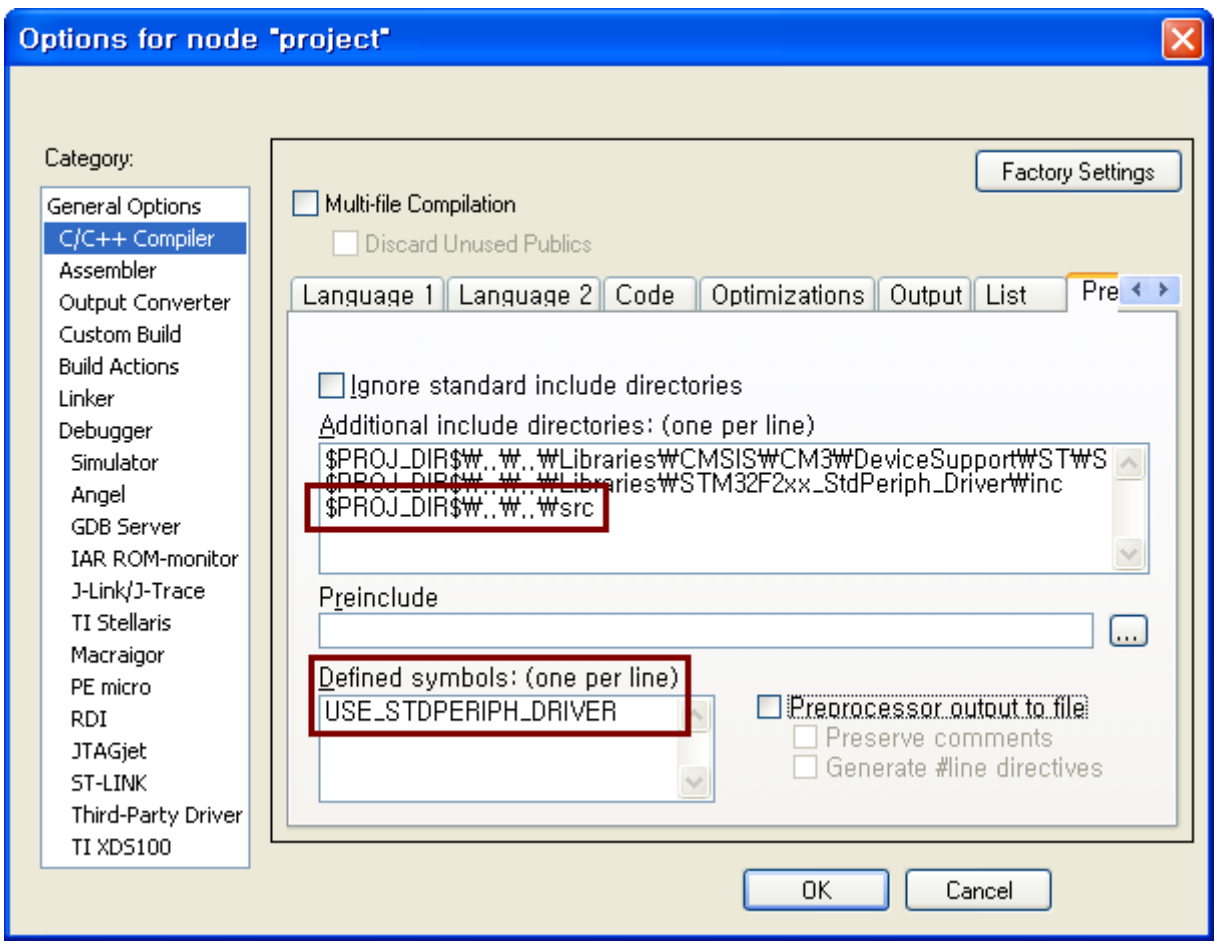
stm32f2xx.h (Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F2xx)

```
#ifdef USE_STDPERIPH_DRIVER
 #include "stm32f2xx_conf.h "
```

```
#endif /* USE_STDPERIPH_DRIVER */
```

위와 같이 정의가 되어 있습니다.

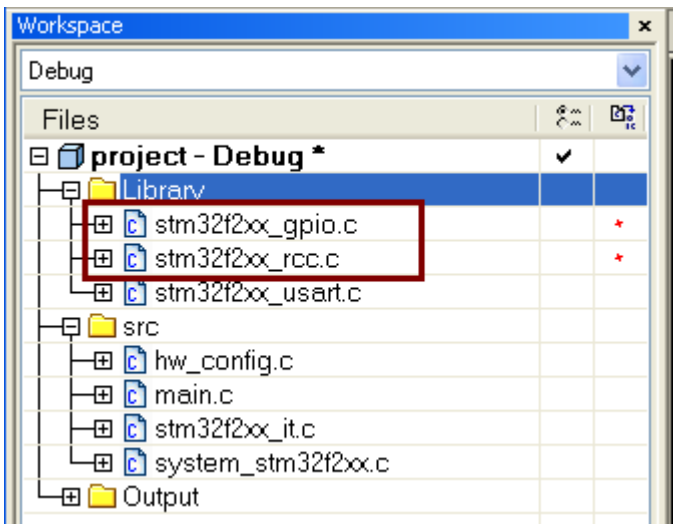
결국 USE_STDPERIPH_DRIVER가 정의되어야만 stm32f2xx_conf.h를 포함하게 됩니다.



USE_STDPERIPH_DRIVER를 위 그림처럼 포함시키고

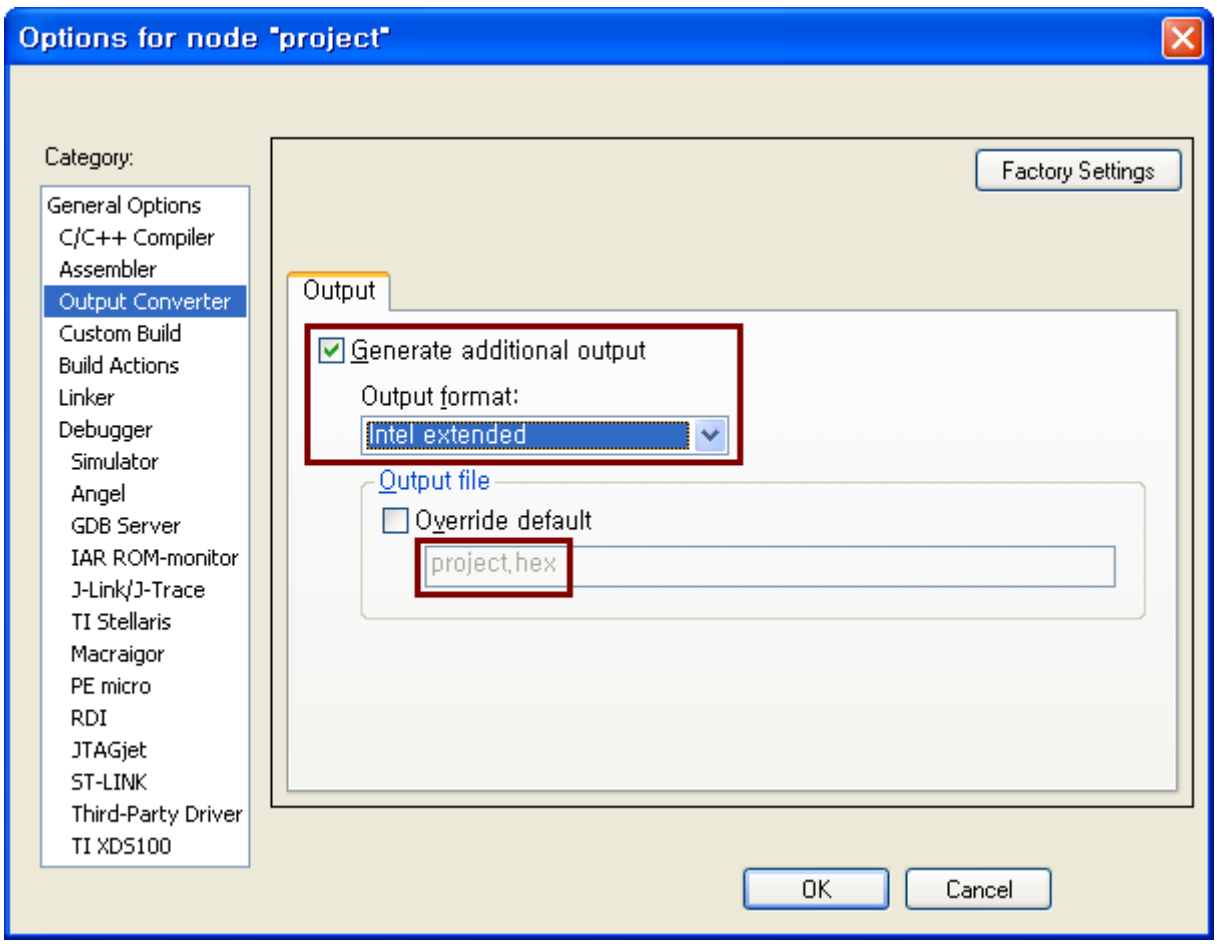
또한 stm32f2xx_conf.h가 존재하는 폴더도 함께 포함시킵니다.

```
Error[Li005]: no definition for "RCC_AHB1PeriphClockCmd" [referenced from D:\Wk.Src\Mango-M32F2\project\EWARM\Debug\Obj\hw_config.o]
Error[Li005]: no definition for "GPIO_PinAFConfig" [referenced from D:\Wk.Src\Mango-M32F2\project\EWARM\Debug\Obj\hw_config.o]
```



이제 빌드는 완료하였습니다.

하지만 다운로드할 바이너리가 생성되지 않았습니다.

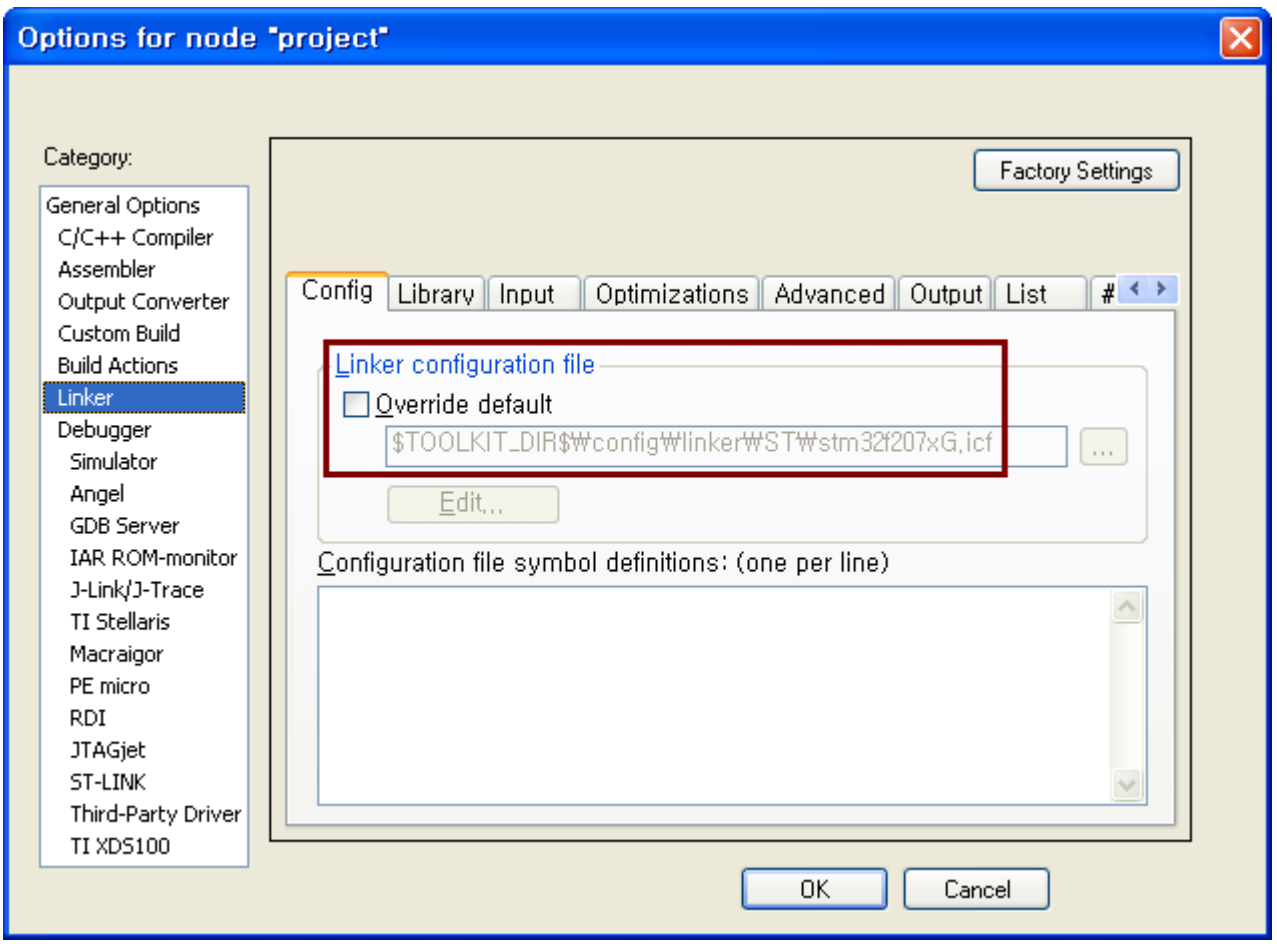


위와 같이 설정을 하여야 project.hex 파일이 생성됩니다.

```
SYSCLK_Frequency = 16000000
HCLK_Frequency = 16000000
PCLK1_Frequency = 16000000
PCLK2_Frequency = 16000000
```

이 상태에서 프로그램을 수행시켜 보면 위와 같이 클럭이 16MHz가 됩니다.
STM32F207의 경우 120MHz까지 수행이 되는데 16MHz로 돌고 있는 것은
내부 클럭을 이용해서 수행이 되는 상태라고 볼 수 있습니다.
이를 120MHz까지 수행되도록 변경할 필요가 있습니다.

그에 앞서서 icf 관련한 부분을 살펴보겠습니다.



현재 project\EWARM 폴더에 icf 파일이 3개 있습니다.
각각에 대해서는 좀더 분석이 필요하지만
일단은 예전부터 사용하던 stm32f2xx_flash.icf를 꼭 포함시켜야 할 듯 하지만
IAR 프로젝트에 디폴트로 들어있는 icf를 그대로 이용해도 무방합니다.
디폴트로 들어있는 것과 project\EWARM 폴더의 것이 조금 다르기는 하지만
동작 상에서는 아직까지는 특별히 차이가 나는 것은 없습니다.
만약 디폴트의 것을 바꿀 필요가 있으면 내용을 편집해야 할 것입니다.

```
/*!< At this stage the microcontroller clock setting is already configured,
     this is done through SystemInit() function which is called from startup
     file (startup_stm32f2xx.s) before to branch to application main.
     To reconfigure the default setting of SystemInit() function, refer to
     system_stm32f2xx.c file
   */
```

main.c에서 위의 주석을 발견할 수 있습니다.
자세히 읽어보면 startup_stm32f2xx.s에서 SystemInit()이 호출되고
여기서 모든 클럭 설정이 이루어진다는 것입니다.
그 이후에 비로소 main 함수에 진입한다는 것입니다.
결국 startup_stm32f2xx.s를 포함시켜야 하는 것을 알 수 있습니다.

Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F2xx\startup에 보면 4가지 종류가 있습니다.
지금 사용하는 툴이 IAR이니 iar 폴더의 것을 사용하면 될 것입니다.

SystemInit() 함수는 system_stm32f2xx.c에 들어 있습니다.
그런데 이 파일이 src 폴더에 있습니다.
하지만 src의 system_stm32f2xx.c 파일과
Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F2xx에 있는 system_stm32f2xx.c가 완전히 동일한 것입니다.
src의 system_stm32f2xx.c 파일은 삭제합니다.
그리고 라이브러리의 system_stm32f2xx.c 파일을 추가합니다.

```
SYSCLK_Frequency = 120000000
HCLK_Frequency = 120000000
PCLK1_Frequency = 30000000
PCLK2_Frequency = 60000000
```

이제 원하는 120 MHz를 얻게 되었습니다.
HCLK, PCLK1, PCLK2도 그에 따라서 변화된 값을 알 수 있습니다.