

# IAR EWARM Compiler 프로젝트 만들기

## I. 개 요

CORTEX-M3의 컴파일러 중 하나인 IAR 사의 IAR EWARM 컴파일러를 이용해서 프로젝트를 만들어 본다.

## II. 사용환경

- 1) OS : Windows XP SP3
- 2) Compiler : IAR EWARM 5.4
- 3) Chip : STM32F103RET6
- 4) Board : Rovitek사 의 Cortex-M3 S.T.D v3.0  
(구매관련 : <http://devicemart.co.kr/goods/view.php?seq=15591>)

## III. 프로젝트 만들기

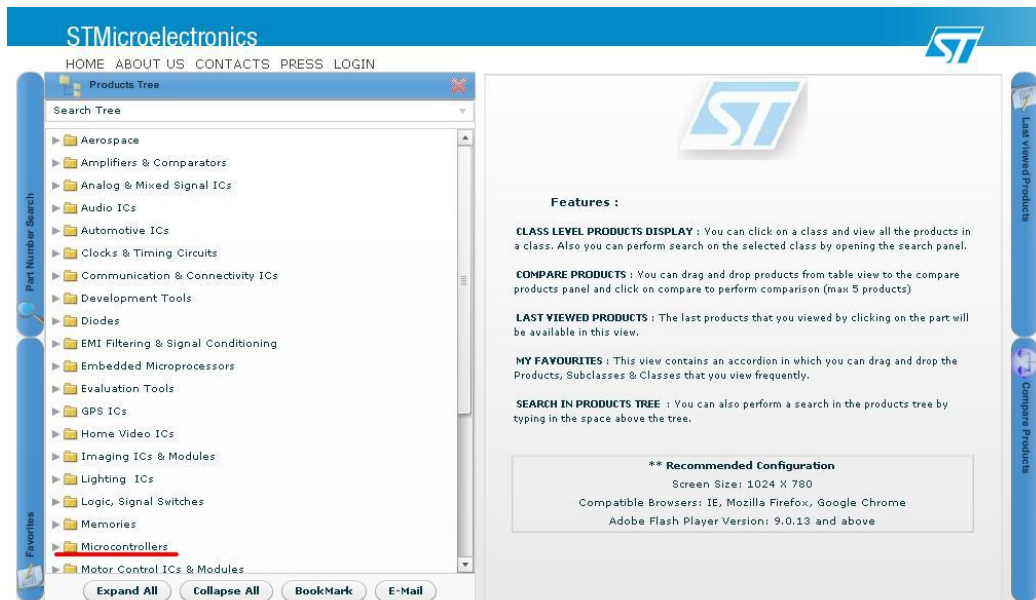
### 1) 준비과정

프로젝트를 만들기 전에 먼저 ST사의 홈페이지에서 라이브러리 파일을 다운받아야 한다.

먼저 <http://www.stm.com> 홈페이지에 접속한 다음 Products 부분에서 Microcontrollers 부분을 클릭한다.



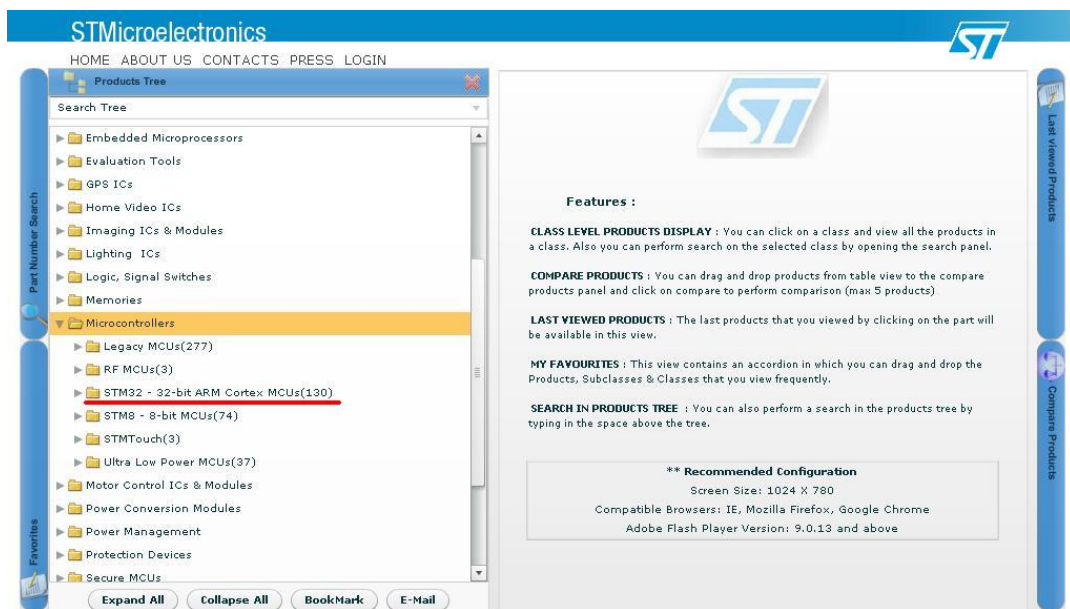
[그림 1] STM 홈페이지에서 Products 부분



All rights reserved ©2011 STMicroelectronics | Terms of use | Privacy Policy | Careers at ST

[그림 2] ST사의 Products Tree

처음 ST사의 홈페이지에서 Product를 클릭하면 [그림 2]와 같은 화면이 나온다. 여기서 Microcontrollers를 클릭하면 아래 [그림 3]과 같이 대표적인 MCU 종류들이 나타난다.

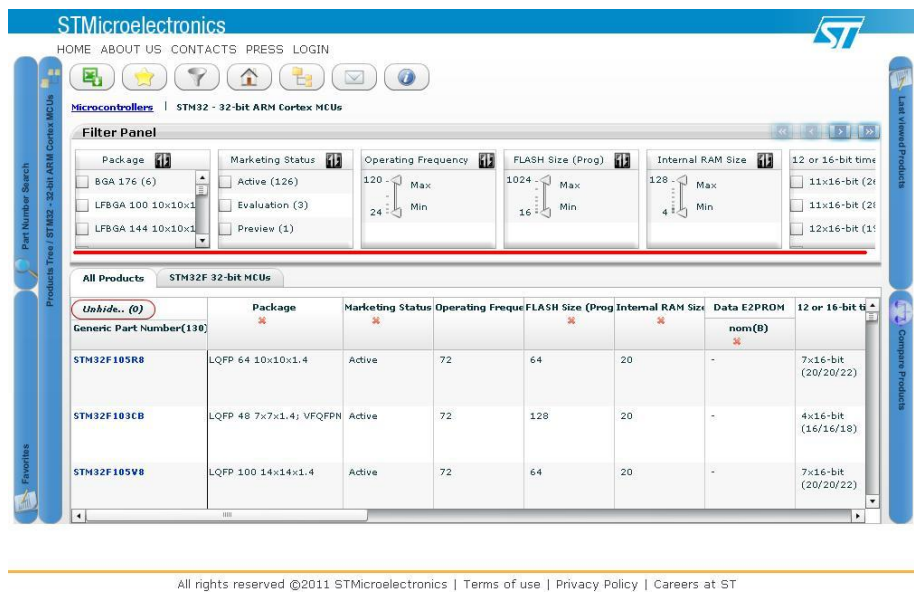


All rights reserved ©2011 STMicroelectronics | Terms of use | Privacy Policy | Careers at ST

[그림 3] ST사의 Cortex-M3 기반의 MCU 종류

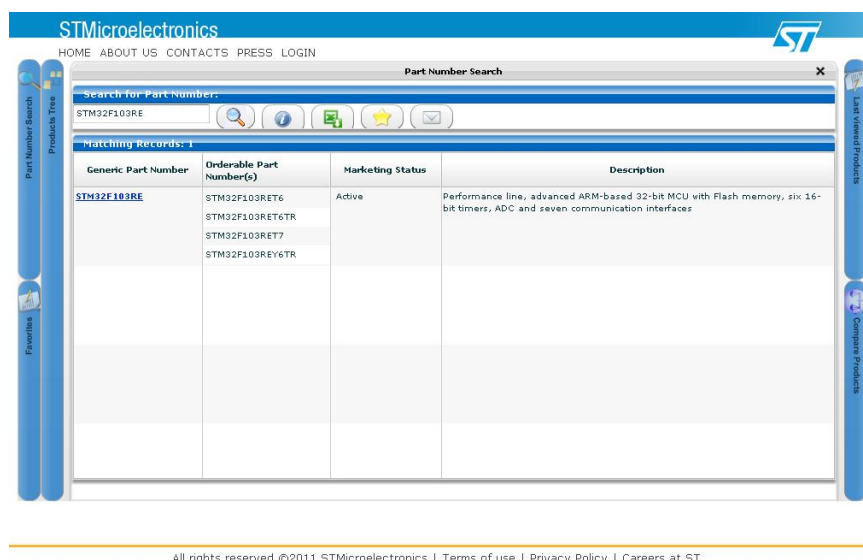
본 프로젝트에서 사용할 MCU는 STM32-32-bit ARM Cortex MCU이기 때문에 [그림 3]의 표시된 부분을 선택한다.

[그림 3]의 표시된 부분을 선택하면 아래 [그림 4]와 같이 여러 가지 사양을 선택할 수 있는 화면이 나오는데 여기서 자신이 적용하고자 하는 분야에 맞는 사양을 선택하면 해당되는 기능을 가진 MCU가 나타난다.



[그림 4] ST사의 MCU 선택 화면

[그림 4]에서 MCU를 사양을 선택 한 후 MCU를 찾아도 되지만 Part Number Search 탭을 클릭해서 직접 MCU의 모델명을 입력한 후 찾아도 된다.



[그림 5] Part Number Search 기능을 이용하여 검색한 화면

현재 본 예제에서 사용할 MCU는 STM32F103RE 이다. 따라서 [그림 5]의 화면에서 STM32F103RE를 클릭하면 아래 [그림 6]과 같은 화면이 나타난다.

**STM32F103RE**

Performance line, advanced ARM-based 32-bit MCU with Flash memory, six 16-bit timers, ADC and seven communication interfaces

Active

Quick view | Design support | Orderable products | ONLINE SUPPORT

**Description**

The STM32F103xC, STM32F103xD and STM32F103xE performance line family incorporates the high-performance ARM®Cortex™-M3 32-bit RISC core operating at a 72 MHz frequency, high-speed embedded memories (Flash memory up to 512 Kbytes and SRAM up to 64 Kbytes), and an extensive range of enhanced I/Os and peripherals connected to two APB buses. All devices offer three 12-bit ADCs, four general-purpose 16-bit timers plus two PWM timers, as well as standard and advanced communication interfaces: up to two I2Cs, three SPIs, two I2Ss, one SDIO, five USARTs, an USB and a CAN.

The STM32F103xx high-density performance line family operates in the -40 to +105 °C temperature range, from a 2.0 to 3.6 V power supply. A comprehensive set of power-saving mode allows the design of low-power applications.

These features make the STM32F103xx high-density performance line microcontroller family suitable for a wide range of applications such as motor drives, application control, medical and handheld equipment, PC and gaming peripherals, GPS platforms, industrial applications, PLCs, inverters, printers, scanners, alarm systems video intercom, and HVAC.

**Key Features**

- Core: ARM 32-bit Cortex™-M3 CPU
  - 72 MHz maximum frequency, 1.25 DMIPS/MHz (Dhrystone 2.1) performance at 0 wait state memory access
  - Single-cycle multiplication and hardware division
- Memories
  - 256 to 512 Kbytes of Flash memory
  - up to 64 Kbytes of SRAM
  - Flexible static memory controller with 4 Chip Select. Supports Compact Flash, SRAM, PSRAM, NOR and NAND memories
  - LCD parallel interface, 8080/6800 modes
- Clock, reset and supply management
  - 2.0 to 3.6 V application supply and I/Os
  - POR, PDR, and programmable voltage detector (PVD)
  - 4-to-16 MHz crystal oscillator
  - Internal 8 MHz factory-trimmed RC
  - Internal 40 kHz RC with calibration
  - 32 kHz oscillator for RTC with calibration
- Low power
  - Sleep, Stop and Standby modes

[그림 6] STM32F103RE 관련 정보 페이지

Part Search Number을 이용하여 모델명을 입력하면 [그림 6]과 같은 페이지가 나타나고 여기에 3개의 탭이 있다.

Quick View는 간략한 모델의 정보, 개요가 나타나있고, Design Support 부분에서는 설계 관련 자료가 있으며, Orderable products 탭에는 주문관련 정보가 있다.

관련 자료를 다운로드 받기 위해서 Design Support 탭을 클릭한 후 아래 [그림 7]의 밑줄 친 부분의 파일을 찾아서 다운받는다.

FIRMWARE			
	Description	Version	Size
	STM32 Embedded GUI library (AN3128)	1.0.1	18274KB
	STM32 I2C Communication Peripheral Application Library (UM1029)	1.0.0	1418KB
	STM32F101xx and STM32F103xx medium- and high-density devices: advanced I²C examples	4.0	1506KB
	STM32F101xx and STM32F103xx low-power modes	2.0.0	1394KB
	Implementing the ADPCM algorithm in high-density STM32F103xx microcontrollers	2.0.	1585KB
	Getting started with uClinux™ for STM32F10x high-density devices	1.0	2270KB
	How to migrate from the STM32F10xxx firmware library V2.0.3 to the STM32F10xxx standard peripheral library V3.0.0	0.0.6	871KB
	EEPROM emulation in STM32F101xx and STM32F103xx microcontrollers	3.1.0	1026KB
	Using the STM32F101xx and STM32F103xx DMA controller	2.0.0	1118KB
	Smartcard interface with the STM32F101xx and STM32F103xx	1.0	1098KB
	Improving STM32F101xx and STM32F103xx ADC resolution by oversampling	1.0	1129KB
	How to use the high-density STM32F103xx microcontroller to play audio files with an external I²S audio codec	2.0.0	1473KB
	STM3210E-EVAL demonstration firmware	2.0.0	5939KB
	3rd party firmware solutions	1.0.0	31KB
	Archive for legacy STM32F10xxx Firmware Library V2.0.3 and all related Firmware packages	2.0.3	23428KB
	CEC (consumer electronic control) C library using the STM32F101xx, STM32F102xx and STM32F103xx microcontrollers	2.0.0	1692KB
	STM32F10xxx Speex library firmware STM32, StdPeriph Lib, speex, audio	2.0.0	1786KB
	STM32F10xxx DSP library firmware	2.0.0	1392KB
	ST-LINK firmware upgrade	1.12.3	362KB
	DfuSe USB Device Firmware Upgrade STMicronics Extension: Contains the Demo GUI, Debugging GUI, all sources files and the protocol layer	3.0.1	6687KB
	How to achieve 32-bit timer resolution using the link system in STM32F101xx and STM32F103xx microcontrollers	3.0.0	1108KB
	<u>STM32F10x standard peripheral library</u>	3.5.0	21617KB
	STM32F10x and STM32L1xx USB full-speed device library	3.3.0	4646KB

[그림 7] ST 표준 라이브러리 다운로드

※ 현재 본 예제에서 사용될 라이브러리는 버전이 3.3.0을 사용하는데 이 버전은 현재 ST 홈페이지에서 다운 받을 수 없기 때문에 따로 파일로 첨부한다.

표준 라이브러리 파일을 다운 받았으면 압축을 풀어서 다음 경로에 있는 파일들을 따로 복사 해 두자.

₩Libraries₩CMSIS  
 ₩Libraries₩STM32F10x\_StdPeriph\_Driver  
 ₩Project₩STM32F10x\_StdPeriph\_Template₩EWARMv5 내 있는 \*.icf 파일  
 ₩Project₩STM32F10x\_StdPeriph\_Examples₩GPIO₩IOToggle 내 있는  
 stm32f10x\_conf.h stm32f10x\_it.h stm32f10x\_it.c 파일

※ 만약 지금까지의 과정을 생략하고 바로 파일을 다운 받고 싶으면 아래 주소에서 받으면 된다.

<http://www.st.com/internet/mcu/product/164485.jsp>

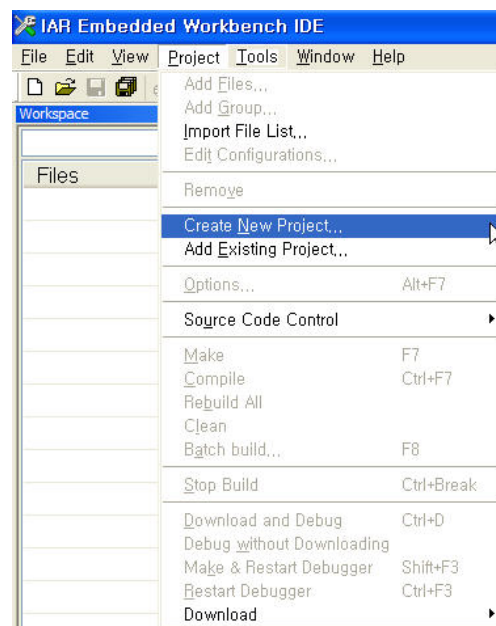


따로 복사해 둔 파일은 프로젝트를 만들 폴더 내에 다음과 같이 폴더를 해당

ST 표준 라이브러리		프로젝트 폴더
₩Libraries₩CMSIS	→	₩CMSIS
₩Libraries₩STM32F10x_StdPeriph_Driver	→	₩StdPeriphDrive
₩Project₩STM32F10x_StdPeriph_Template₩ EWARMv5 내 있는 *.icf 파일	→	프로젝트 폴더에 복사
stm32f10x_conf.h		
stm32f10x_it.h	→	₩USER
stm32f10x_it.c		

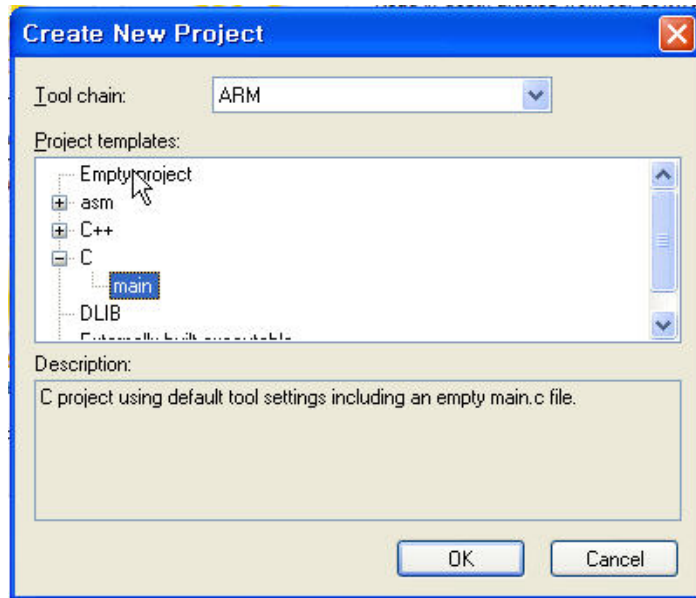
## 2) 프로젝트 만들기

프로젝트를 만들기 위해서 IAR EWARM을 실행 시킨다.



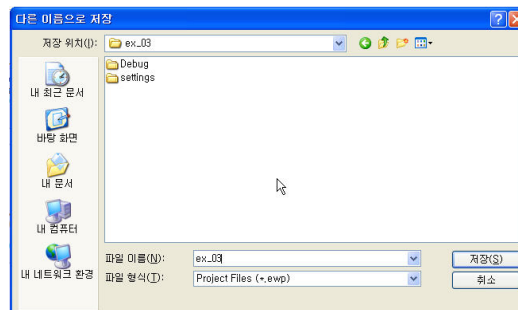
[그림 8] 프로젝트 만들기

Create New Project를 클릭하면 아래 그림과 같이 작성할 언어를 선택할 수 있는 창이 나온다. 이 때 사용하고자 하는 언어를 선택한다.



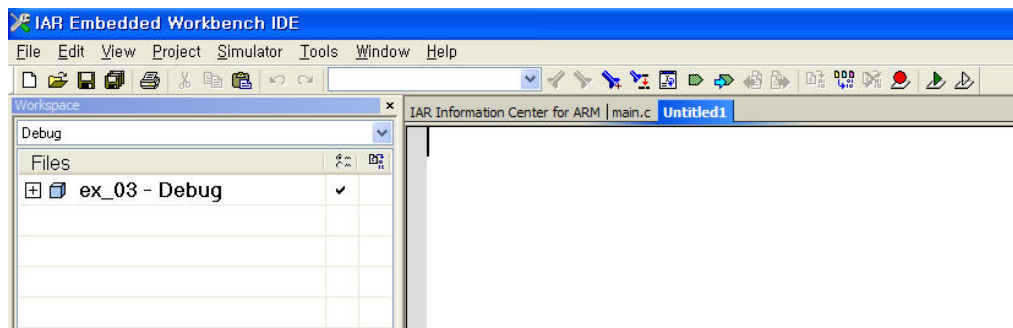
[그림 9] 사용 언어 선택

사용 언어를 선택 한 후 OK를 클릭하면 프로젝트를 저장하는 창이 나온다.  
이 때 원하는 위치에 저장을 하면 된다.



[그림 10] 프로젝트 저장

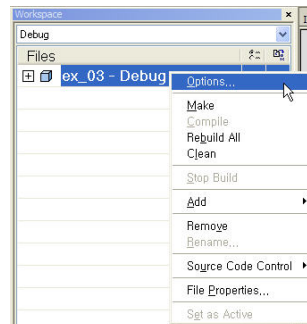
프로젝트를 저장 하고 나면 기본적으로 다음과 같은 화면이 나온다.



[그림 11] 프로젝트 생성 성공시 화면

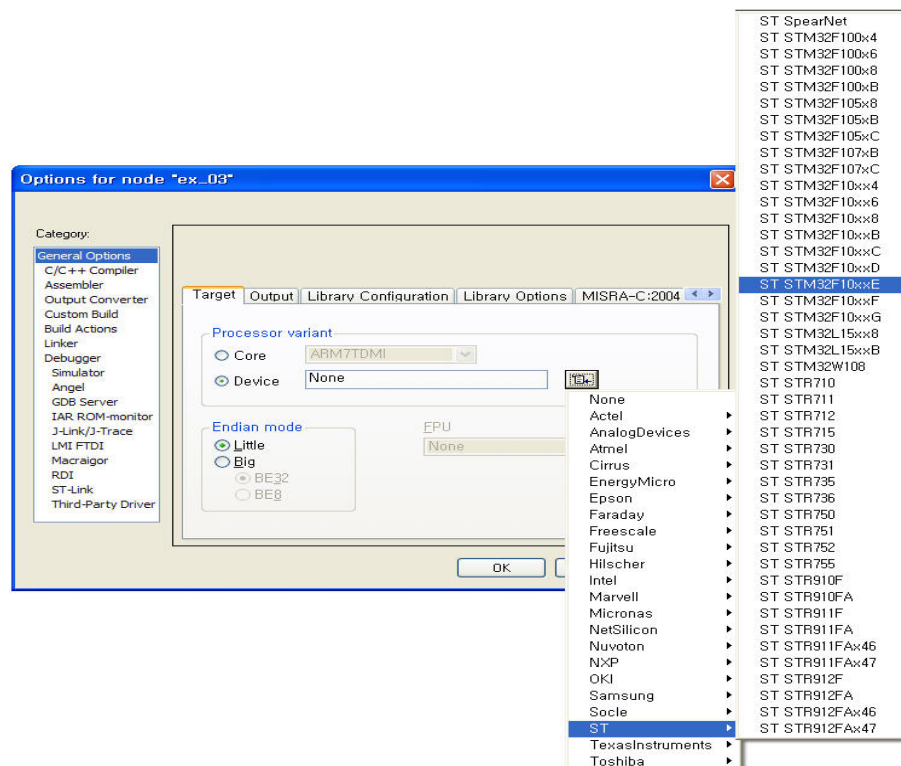
※ 만약 좌측면에 있는 Workspace 화면이 나오지 않는다면 메뉴바의 View → Workspace를 선택하면 나타난다.

이제 프로젝트에서 사용할 Cortex-M3 디바이스 및 각종 컴파일 설정을 해야 한다.  
먼저 [그림 12]과 같이 프로젝트 명에서 마우스 우클릭을 하면 팝업창이 나오는데 여기서 Option을 클릭한다.



[그림 12] 프로젝트 옵션

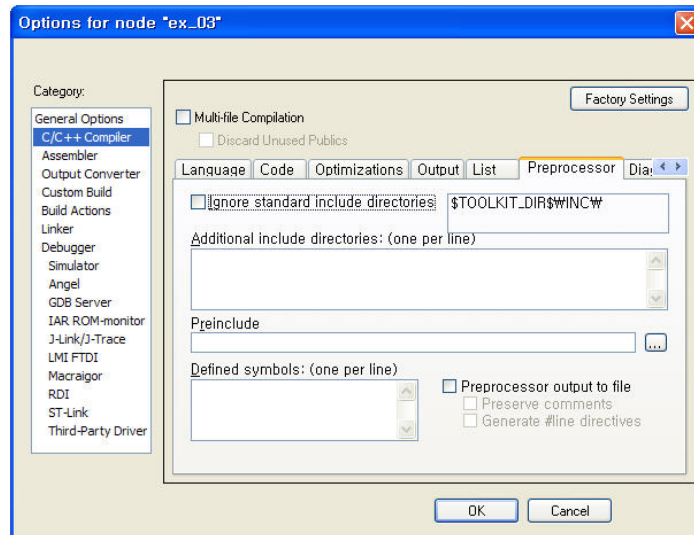
Option을 클릭하면 다음과 같은 화면이 나온다.



[그림 13] Option 설정 창



Option을 클릭하면 제일 먼저 [그림 13] 창이 나온다. 여기서 Device 부분을 선택 한 후 우측 네모 버튼을 클릭하여 사용할 디바이스를 선택한다.



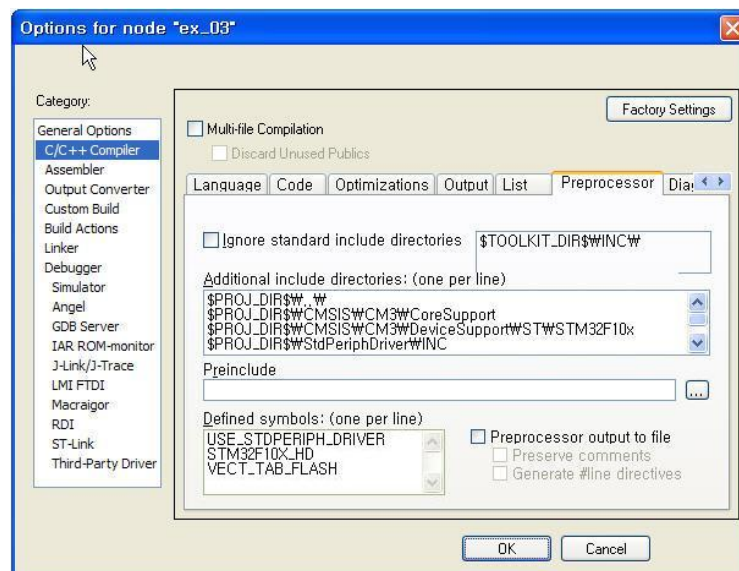
[그림 14] 헤더파일 위치 및 기타 설정

[그림 14]에서 Additional include directories :(one per line) 부분은 헤더파일의 위치를 지정하는 부분이다.

이 때 \$TOOLKIT\$로 시작하는 경로는 IAR EWARM이 설치된 폴더내부의 경로 이며 \$PROJ\_DIR\$은 프로젝트 폴더 내의 경로(상대경로)를 뜻한다.

Define Symbols는 컴파일시 옵션을 적는 곳이다.

이제 [그림 14]에서의 빈칸을 다음과 같이 하자



[그림 15] 각종 설정 완료 후

Additional include directories :(one per line) 부분은 다음과 같이 입력하면 된다.

```
$PROJ_DIR$W..W
$PROJ_DIR$WCMSISWCM3WCoreSupport
$PROJ_DIR$WCMSISWCM3WDeviceSupportWSTWSTM32F10x
$PROJ_DIR$WStdPeriphDriverWINC
$PROJ_DIR$WStdPeriphDriverWSRC
$PROJ_DIR$WUserW
```

Define symbols : (one per line) 부분은 다음과 같이 입력하면 된다.

```
USE_STDPERIPH_DRIVER
STM32F10X_HD
VECT_TAB_FLASH
```

Define symbols에 USE\_STDPERIPH\_DRIVER를 입력 하면

```
#define USE_STDPERIPH_DRIVER 1
```

따라서 컴파일하는 프로그램 내에서

```
#if USE_STDPERIPH_DRIVER
```

```
#endif
```

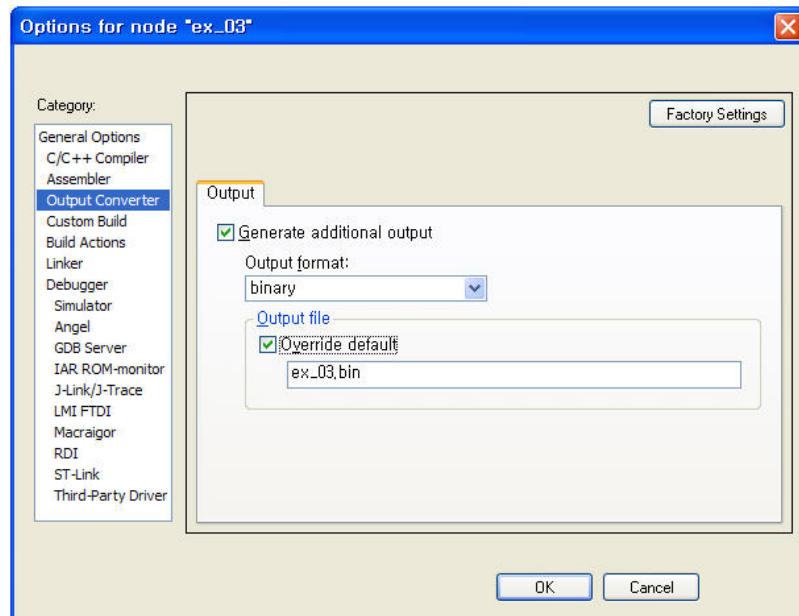
로 되어 있는 문장은 컴파일이 되어서 실행이 된다.

여기서 USE\_STDPERIPH\_DRIVER 는 ST사의 Cortex-M3 표준 라이브러리를 사용 한다는 것이고 STM32F10X\_HD는 사용하는 M3 칩의 종류에 해당하는 것이다.

만약 STM32F103RET6를 사용하지 않고 다른 것을 사용한다면 데이터 시트를 보고 사용하는 칩이 LD, MD 등 어느 부분에 속하는지 확인을 한 다음 적어야 한다.

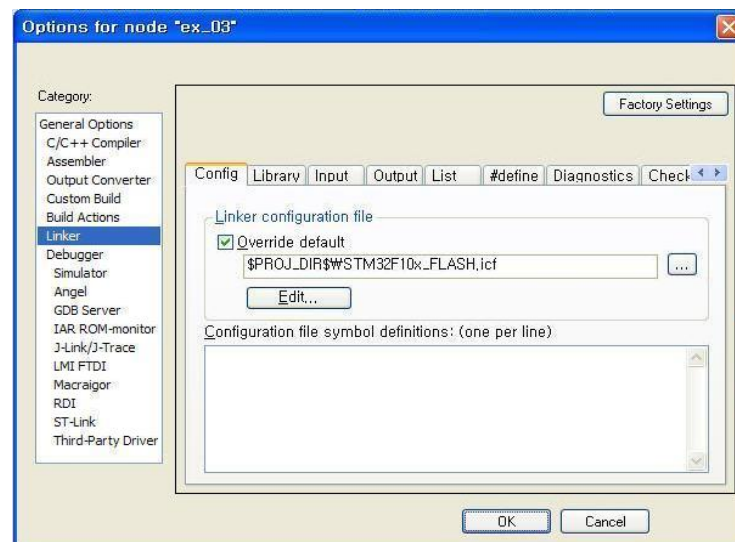
마지막으로 VECT\_TAB\_FLASH는 프로그램 메모리를 Flash memory에 저장한다는 것이다.

[그림 15] 와 같이 입력이 다 되었으면 이제 출력 파일을 컴파일 후 생성할 파일을 선택해야 한다.



[그림 16] 출력 파일 설정

[그림 16]와 같이 출력 파일을 설정하였으면 이제 Linker를 설정해야 한다.



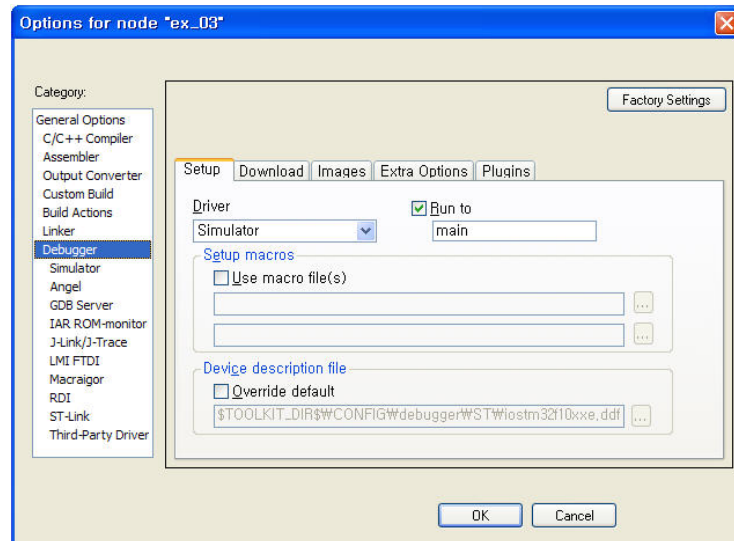
[그림 17] Linker 설정

[그림 17]에서 Override default를 체크한 후 우측 버튼을 클릭 한 후 위 [그림 16]과 같이 입력한다.

STM32F103RET6은 프로그램 메모리가 flash memory이기 때문에 stm32f10x\_flash.icf를 선택한다. 만약 외부에 메모리를 확장했다면 stm32f10x\_flash\_extsram.icf를 선택하면 된다.

이 외에 nor, ram은 칩에 사용된 메모리 및 확장한 메모리의 종류에 따라서 선택하여서 사용하면 된다.

[그림 18]과 같이 icf 설정을 모두 마쳤다면 마지막으로 Debugger를 설정하자.

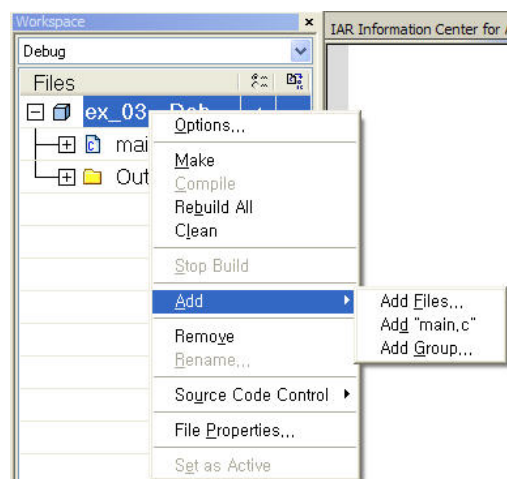


[그림 18] 디버거 선택

만약 JTAG나 디버거를 사용하지 않는다면 별다른 설정을 하지 말고 기본 설정을 사용하면 된다.

지금까지 프로젝트 설정을 마쳤다면 이제 사용할 각종 헤더파일을 추가해야 한다. 헤더파일은 1) 준비과정 에서 복사 해 둔 파일을 붙여와야 한다.

먼저 파일을 그룹별로 정리를 하기 위해서 Workspace에 다음과 같이 그룹을 만든다. 그룹을 만들 때 먼저 Workspace에 있는 프로젝트 명에서 마우스 오른쪽 버튼을 눌러서 팝업창이 뜬다.

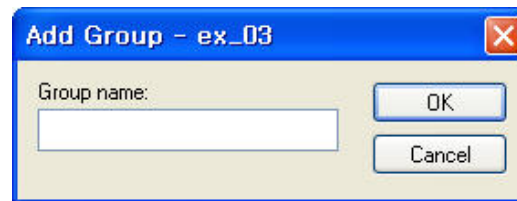


[그림 19] 추가 메뉴

[그림 19]에서 파일을 추가하고 싶으면 Add Files, 그룹을 추가하고 싶으면 Add

Group 그리고 Files와 Group의 중간에는 항상 현재 편집중인 파일을 바로 추가할 수 있게 되어 있다.

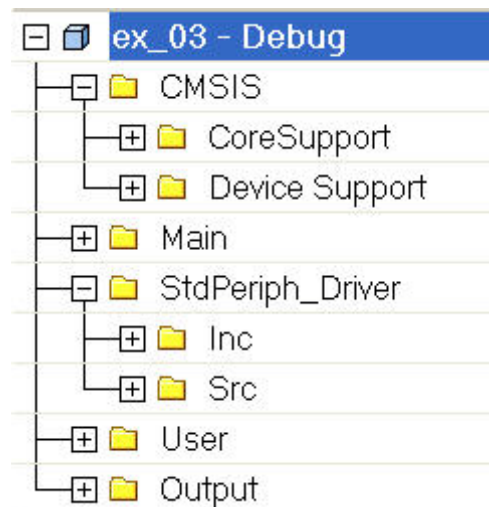
여기서 Group를 누르면 다음과 같은 화면이 나온다.



[그림 20] 그룹 추가하기

여기에 빈칸에 그룹명을 입력한 후 OK를 누르면 그룹이 생성된다.

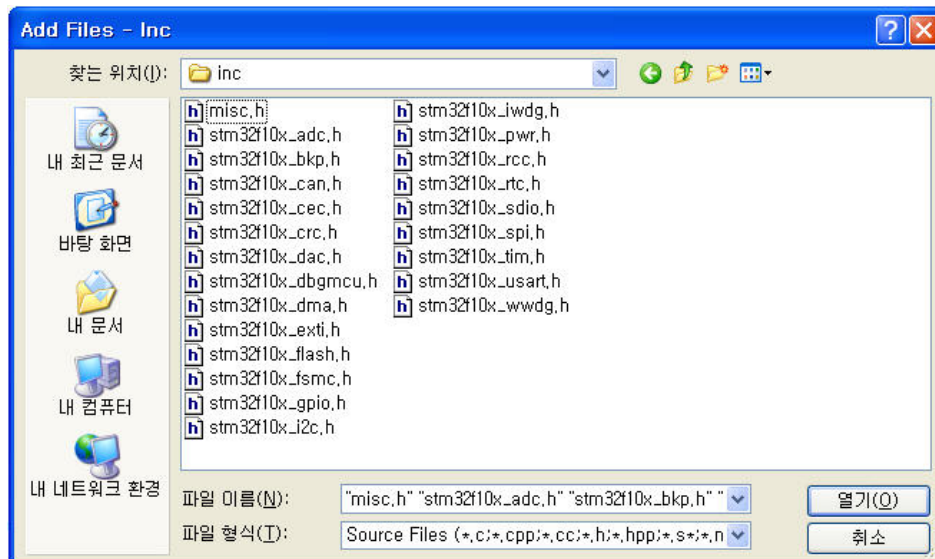
이제 다음과 같이 그룹을 생성하자.



[그림 21] 생성된 그룹

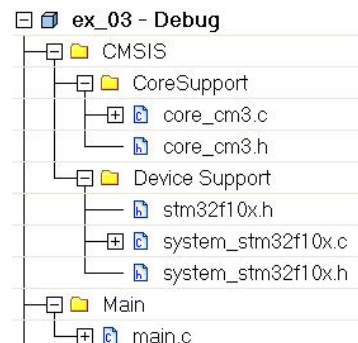
이 때 Output 그룹은 자동으로 프로젝트를 만들 때 생성이 되기 때문에 따로 생성하지 않아도 된다.

이제 생성된 그룹에 파일을 추가해야한다. 파일 추가는 [그림 19]에서 Add Files를 선택하면 파일을 추가할 수 있다.

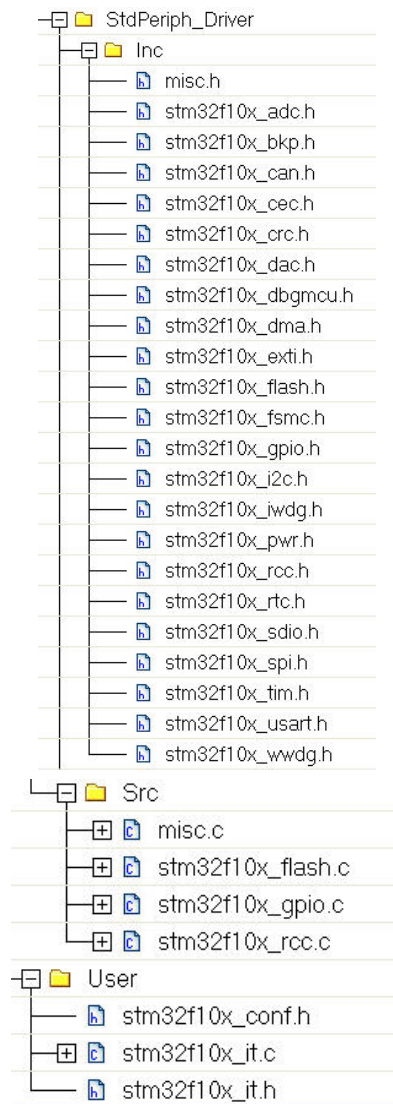


[그림 22] 파일 추가

이제 다음 그림과 같이 파일들을 찾아서 추가를 하자







[그림 23] 파일 추가

#### IV. 프로그램 소스 작성 및 컴파일

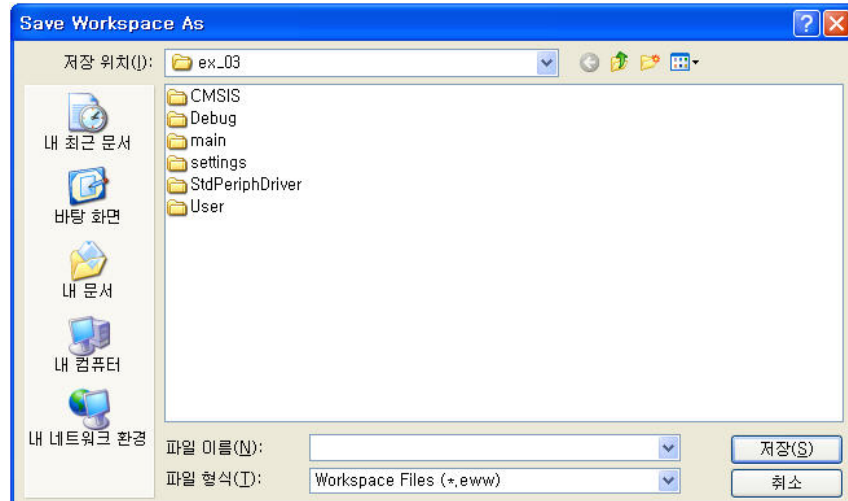
프로젝트 생성 및 설정이 다되었으면 이제 프로그램 소스를 입력 한 후 컴파일을 하자.  
예제소는 LED를 점멸하는 예제이다.



[그림 24] 컴파일

제일 왼쪽부터 컴파일, MAKE, 컴파일 중지, 중단점 설정, RUN(디버깅 모드 및 다운로드), STOP(디버깅 중단)

컴파일을 처음 하면 아래 그림과 같이 한번 더 프로젝트를 저장하는 창이 나온다.



[그림 25] 프로젝트 재 저장

컴파일 후 MAKE 버튼을 눌러서 다운로드 할 파일이 생성되도록 하자. 정상적으로 컴파일, MAKE가 되었다면 프로젝트 폴더 내 Debug\EXE\EXE 폴더 내에 \*.bin 파일이 생성된다.

## V. 작성 정보

작성자 : 최 중해([cjh0619@naver.com](mailto:cjh0619@naver.com))

작성일 : 2010. 8. 22.(일)

이 력 : 최초 작성

수정자 : 최 중해([cjh0619@naver.com](mailto:cjh0619@naver.com))

수정일 : 2010. 12. 6.(월)

이 력 : 프로젝트 설정 부분 수정

수정자 : 최중해([cjh0619@naver.com](mailto:cjh0619@naver.com))

수정일 : 2012. 8. 19.(일)

이 력 : 오타 수정, 그림 수정