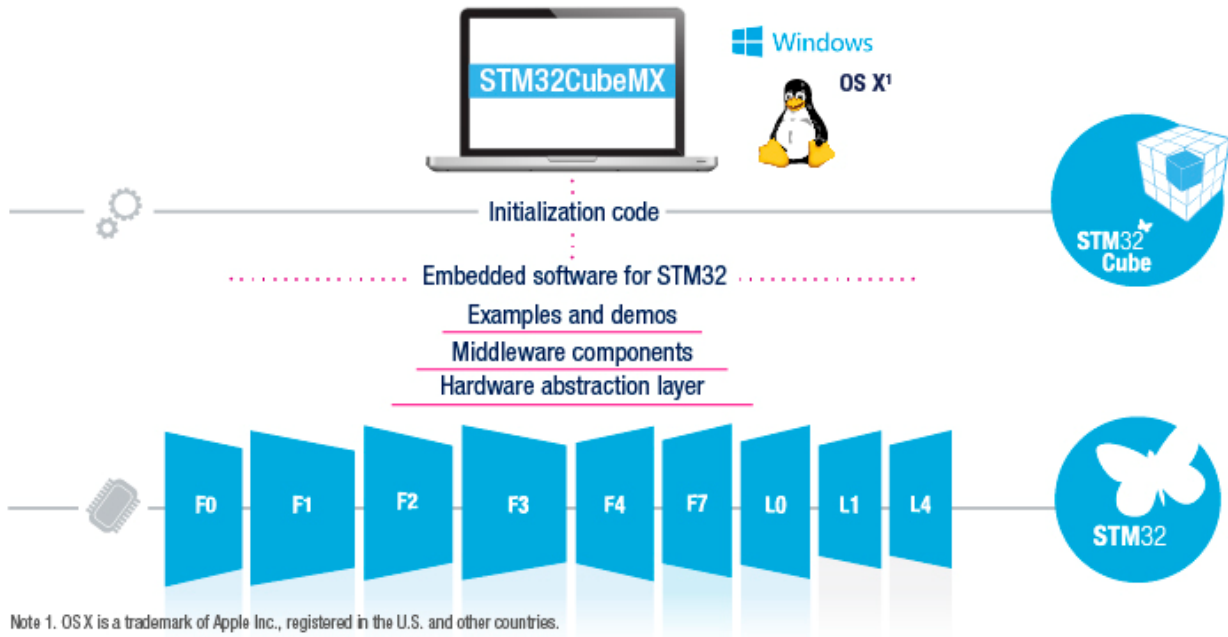


STM32CubeMX Quick Start Guide.



INTRODUCTION

이 문서는 STM32CubeMX 를 처음 사용하는 사람들을 위해 간략한 예제와 함께 STM32CubeMX 사용방법을 간략하게 설명하기 위해 작성되었습니다. 자세한 사항은 www.st.com 에서 User manual “STM32CubeMX for STM32 configuration and initialization C code generation” 문서를 참고하시기 바랍니다.

DESCRIPTION

STM32CubeMX 는 사용자가 STM32 Microcontroller 를 사용하는 프로젝트를 시작하는데 있어 그래픽한 인터페이스 환경에서 Clock, Peripheral, Middleware 를 설정하여 초기화 C 코드를 생성함으로써 노력, 시간 그리고 비용을 감소시켜 개발자들이 쉽게 개발 할 수 있도록 도와주는 툴입니다.

- 모든 STM32 포트폴리오를 포함하고 있어 Microcontroller 를 쉽게 선택.
- 그래픽한 인터페이스 환경에서 pin, clock, peripheral, middleware 를 쉽게 설정하고 일치하는 초기화 C 코드를 생성.
- Standalone 또는 Eclipse plug-in 으로 사용 가능.
- 설정 상태를 report 형태의 문서로 생성 가능.
- 소모전류에 대한 시뮬레이션 가능.
- 초기화 C 코드를 준비되어 있는 IDE tool chain(IAR, KEIL, GCC ...) 으로 선택하여 생성 가능.

Contents

1	STM32CubeMX 설치	4
2	Welcome page	6
3	New Project Window	7
4	Main window	8
4.1	Toolbar and menus	10
4.2	File menu	10
4.3	Project menu	10
4.4	Pinout menu	11
4.5	Window menu	13
4.6	Help menu	13
5	Clock tree configuration view	14
6	Configuration view	16
6.1	IP and Middleware Configuration window	17
6.2	User Cnstants configuration window	18
6.3	GPIO Configuration window	19
6.4	DMA Configuration window	19
6.5	NVIC Configuration window	21
6.6	FreeRTOS middleware configuration view	24
7	Power Consumption Calculator (PCC) view	25
8	STM32CubeMX Simple Example	31

Figure

FIGURE 1	STM32CUBEMX 다운로드 PAGE (WWW.ST.COM/STM32CUBEMX)	4
FIGURE 2	STM32CUBEMX 라이브러리 패키지 설치	4
FIGURE 3	REPOSITORY 폴더 설정	5
FIGURE 4	다운로드된 REPOSITORY 의 라이브러리들	5
FIGURE 5	WELCOME PAGE	6
FIGURE 6	MCU SELECTION	7

FIGURE 7 BOARD SELECTION	7
FIGURE 8 MAIN MENU	8
FIGURE 9 MAIN WINDOW (NO CHECK THE PERIPHERAL DEFAULT MODE)	9
FIGURE 10 MAIN WINDOW (CHECK THE PERIPHERAL DEFAULT MODE)	9
FIGURE 11 PINOUT MENU (PINOUT 탭이 선택된 상태)	11
FIGURE 12 PINOUT MENUS (PINOUT 탭이 선택되지 않은 상태).....	11
FIGURE 13 CLOCK TREE VIEW	14
FIGURE 14 HSI 클럭을 선택	15
FIGURE 15 비활성화 되어 있는 HSE CLOCK SOURCE	15
FIGURE 16 활성화된 HSE CLOCK SOURCE.....	15
FIGURE 17 PLL CLOCK SOURCE 선택	16
FIGURE 18 STM32CUBEMX CONFIGURATION VIEW	17
FIGURE 19 IP CONFIGURATION WINDOW (STM32F4 SERIES)	18
FIGURE 20 USER CONSTANTS WINDOW	18
FIGURE 21 생성된 MXCONSTANTS.H 파일	18
FIGURE 22 GPIO CONFIGURATION WINDOW - GPIO SELECTION	19
FIGURE 23 DMA CONFIGURATION	20
FIGURE 24 DMA CONFIGURATION	21
FIGURE 25 NVIC CONFIGURATION TAB - FREERTOS DISABLED	22
FIGURE 26 NVIC CONFIGURATION TAB - FREERTOS ENABLED	23
FIGURE 27 I2C NVIC CONFIGURATION WINDOW\	23
FIGURE 28 FREERTOS CONFIGURATION VIEW	25
FIGURE 29 POWER CONSUMPTION CALCULATOR DEFAULT VIEW	27
FIGURE 30 BATTERY SELECTION	28
FIGURE 31 STEP MANAGEMENT FUNCTIONS	28
FIGURE 32 POWER CONSUMPTION SEQUENCE: NEW STEP DEFAULT VIEW	29
FIGURE 33 EDIT STEP WINDOW	30
FIGURE 34 STM32F401C-DISCO BOARD	31
FIGURE 35 SELECT MCU ON STM32F401C-DISCO	32
FIGURE 36 STM32F401VCTx PINOUT VIEW	32
FIGURE 37 STM32F401C-DISCO : PART OF EXTERNAL OSCILLATOR	33
FIGURE 38 ENABLE HSE	33
FIGURE 39 ADJUST MCU SYSTEM CLOCK	33
FIGURE 40 SET PA0-WKUP.....	34
FIGURE 41 PA0-WKUP CONFIGURATION.....	34
FIGURE 42 ENABLE EXTI LINE0 (PA0-WAUP)	35
FIGURE 43 STM32F401C-DISCO CODE GENERATION	36
FIGURE 44 STM32F401C-DISCO INITIALIZATION CODE WITH WAKE-UP	36
FIGURE 45 GPIO INITIALIZATION CODE	37
FIGURE 46 WAKE-UP(PA0) INTERRUPT OPERATION	37

Table


TABLE 1 FILE MENU FUNCTIONS.....	10
TABLE 2 PROJECT MENU	10
TABLE 3 PINOUT MENU	12
TABLE 4 PIN MENU.....	13
TABLE 5 HELP MENU.....	13

1 STM32CubeMX 설치

▶ STM32CubeMX 설치파일 및 자료 : <http://www.st.com/stm32cube>

STM32CubeMX ACTIVE Save to My ST Share Print

STM32Cube initialization code generator

 Download Databrief

QUICK VIEW **DESIGN** **GET SOFTWARE**

STM32CubeMX is part of STMicroelectronics STMCube™ original initiative to ease developers life by reducing development efforts, time and cost. STM32Cube covers STM32 portfolio.

STM32Cube includes the STM32CubeMX which is a graphical software configuration tool that allows generating C initialization code using graphical wizards.

It also embeds a comprehensive software platform, delivered per series (such as STM32CubeF4 for STM32F4 series). This platform includes the STM32Cube HAL (an STM32 abstraction layer embedded software, ensuring maximized portability across STM32 portfolio), plus a consistent set of middleware components (RTOS, USB, TCP/IP and graphics). All embedded software utilities come with a full set of examples.

STM32CubeMX is a graphical tool that allows configuring STM32 microcontrollers very easily and generating the corresponding initialization C code through a step-by-step process.

Step one consists in selecting the STMicroelectronics STM32 microcontroller that matches the required set of peripherals.

The user must then configure each required embedded software thanks to a pinout-conflict solver, a clock-tree setting helper, a power-consumption calculator, and an utility performing MCU peripheral configuration (GPIO, USART, ...) and middleware stacks (USB, TCP/IP, ...).

Finally, the user launches the generation of the initialization C code based on the selected configuration. This code is ready to be used within several development environments. The user code is kept at the next code generation.

GET SOFTWARE

Part Number	Software Version	Marketing Status	Supplier	Order from ST
STM32CubeMX	4.15.0	Active	ST	<div>Click</div> <div>Get Software</div>

Figure 1 STM32CubeMX 다운로드 Page (www.st.com/stm32cubemx)

▶ 다운로드된 압축 파일을 해제 후 [SetupSTM32CubeMX-x.xx.x.exe] 파일을 설치합니다.

[Help] → [Install New Libraries] 선택 후 Library list 에서 device 및 version 을 고려하여 적절한 library 를 다운로드.

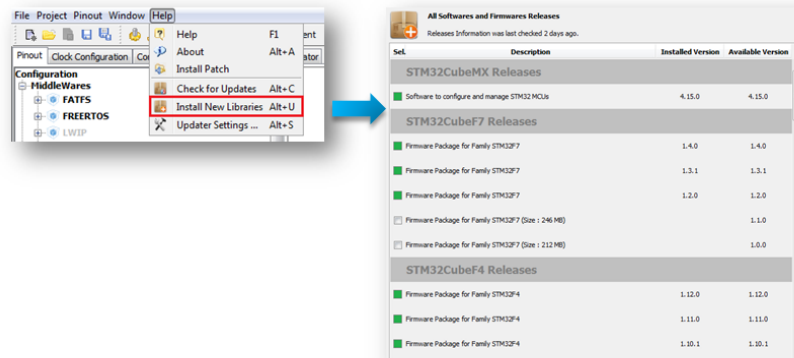


Figure 2 STM32CubeMX 라이브러리 패키지 설치

선택된 Library의 다운로드 경로는 [Help] → [Update Settings ...] → [Repository Folder]에서 설정.

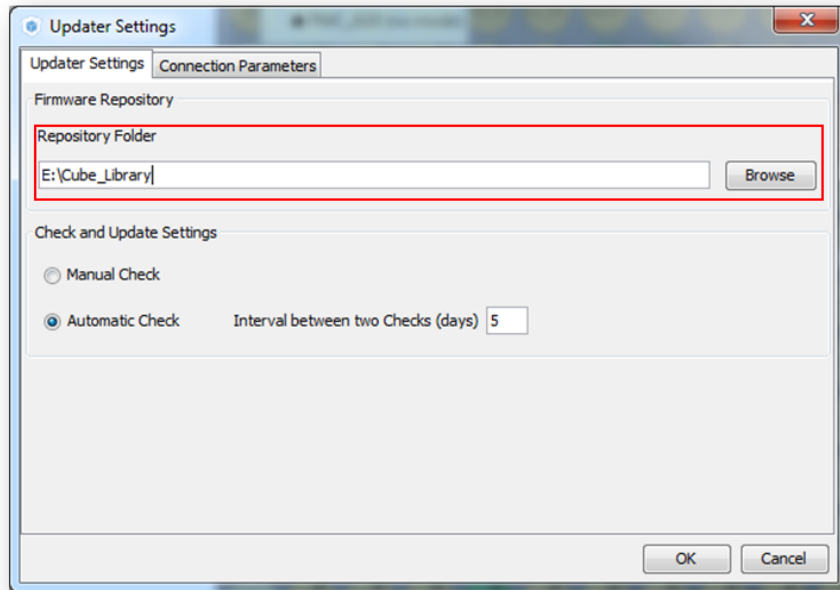


Figure 3 Repository 폴더 설정

STM32Cube_FW_F0_V1.6.0	2016-06-24 오후 1...	File folder	
STM32Cube_FW_F1_V1.4.0	2016-06-24 오후 1...	File folder	
STM32Cube_FW_F2_V1.3.0	2016-06-24 오후 1...	File folder	
STM32Cube_FW_F3_V1.5.0	2016-06-23 오후 6:...	File folder	
STM32Cube_FW_F4_V1.12.0	2016-06-24 오후 1...	File folder	
STM32Cube_FW_F7_V1.4.0	2016-06-23 오후 6:...	File folder	
STM32Cube_FW_L0_V1.7.0	2016-06-24 오후 1...	File folder	
STM32Cube_FW_L1_V1.5.0	2016-06-24 오후 1...	File folder	
STM32Cube_FW_L4_V1.5.0	2016-06-24 오후 1...	File folder	
stm32cube_fw_f0_v160.zip	2016-06-24 오후 1...	zip Archive	132,913 KB
stm32cube_fw_f1_v140.zip	2016-06-24 오후 1...	zip Archive	79,148 KB
stm32cube_fw_f2_v130.zip	2016-06-24 오후 1...	zip Archive	81,024 KB
stm32cube_fw_f2_v131.zip	2016-06-24 오후 1...	zip Archive	14,660 KB
stm32cube_fw_f3_v150.zip	2016-06-23 오후 6:...	zip Archive	128,915 KB
stm32cube_fw_f4_v1120.zip	2016-06-24 오후 1...	zip Archive	350,152 KB
stm32cube_fw_f7_v140.zip	2016-06-23 오후 6:...	zip Archive	517,618 KB
stm32cube_fw_l0_v170.zip	2016-06-24 오후 1...	zip Archive	71,093 KB
stm32cube_fw_l1_v150.zip	2016-06-24 오후 1...	zip Archive	65,482 KB
stm32cube_fw_l4_v150.zip	2016-06-24 오후 1...	zip Archive	229,839 KB
stm32cube_fw_l4_v151.zip	2016-06-24 오후 1...	zip Archive	33,433 KB

Figure 4 다운로드된 Repository의 라이브러리들

2 Welcome page

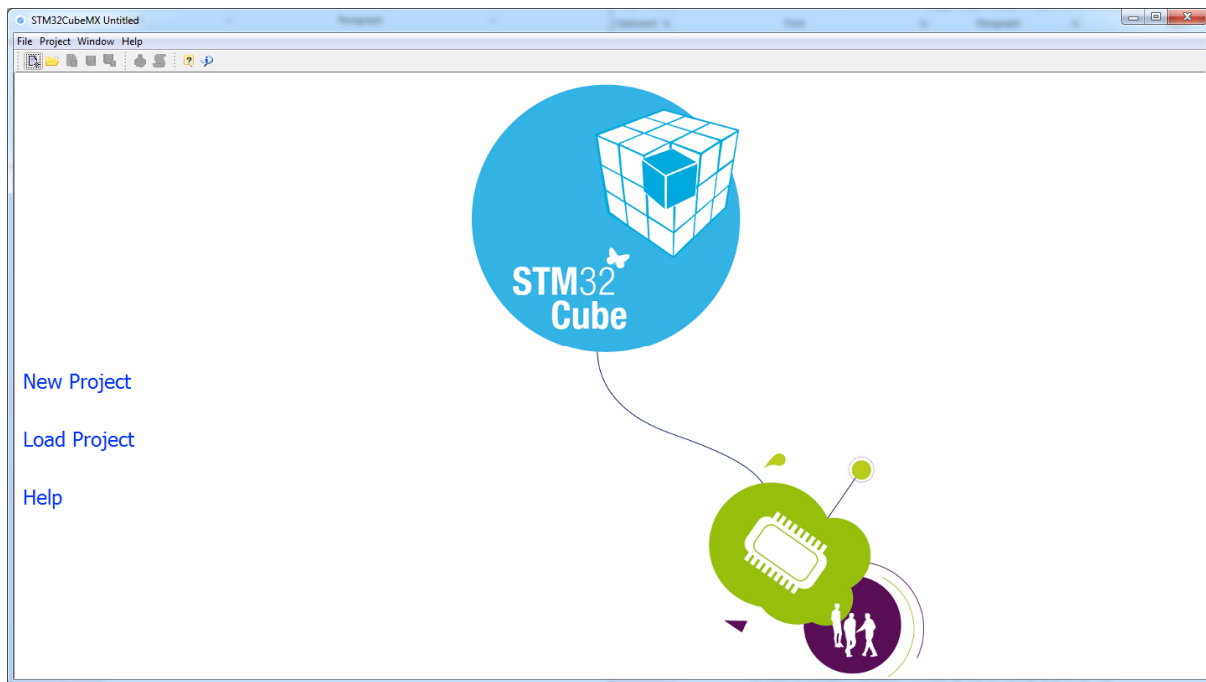


Figure 5 Welcome Page

[New Project] : New project window 창을 활성화하여 STM32CubeMX 의 새로운 프로젝트 생성을 시작합니다.

[Load Project] : 기존에 저장된 설정(.ios 파일)을 선택하기 위한 브라우저 창을 활성화 합니다.

* (주의) STM32CubeMX 가 새로운 버전으로 업그레이드 되었을 때, 새 프로젝트를 loading 하기 전에 반드시 프로젝트들을 항상 backup 해야합니다. (특히, 프로젝트가 user code 를 포함하고 있을 때)

[Help] : User manual 을 활성화 합니다.

3 New Project Window

New project window 선택할 수 있는 2 가지 Tap 을 가지고 있습니다.

- MCU 리스트를 제공하는 [MCU selector] tap
- STMicroelectronics 보드의 리스트를 보여주는 [Board Selector] tap

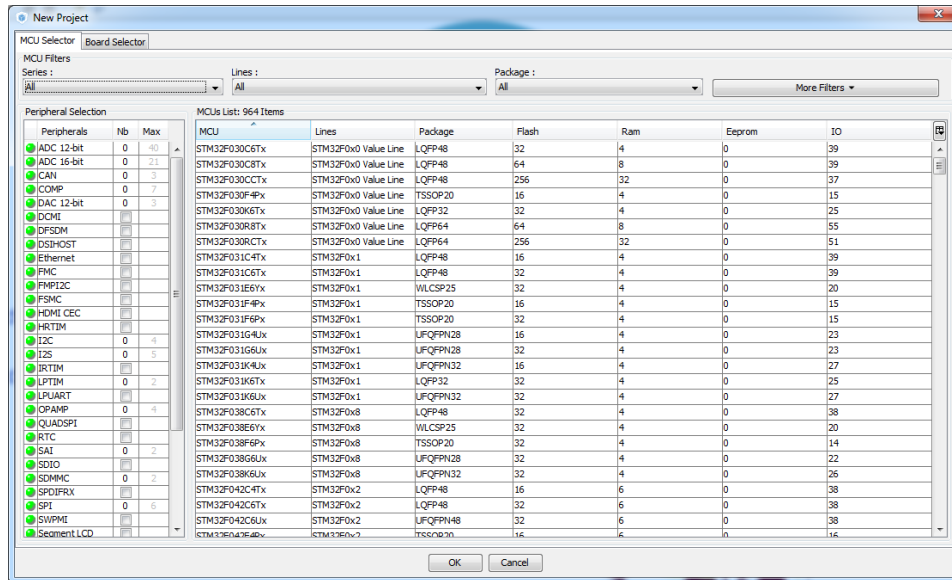


Figure 6 MCU Selection

[MCU Selector] 는 다양한 기준으로 필터링 가능 : series, lines, packages, peripherals 와 추가적으로 memory size 또는 I/O 갯수와 같은 MCU 특징으로 필터링이 가능합니다.

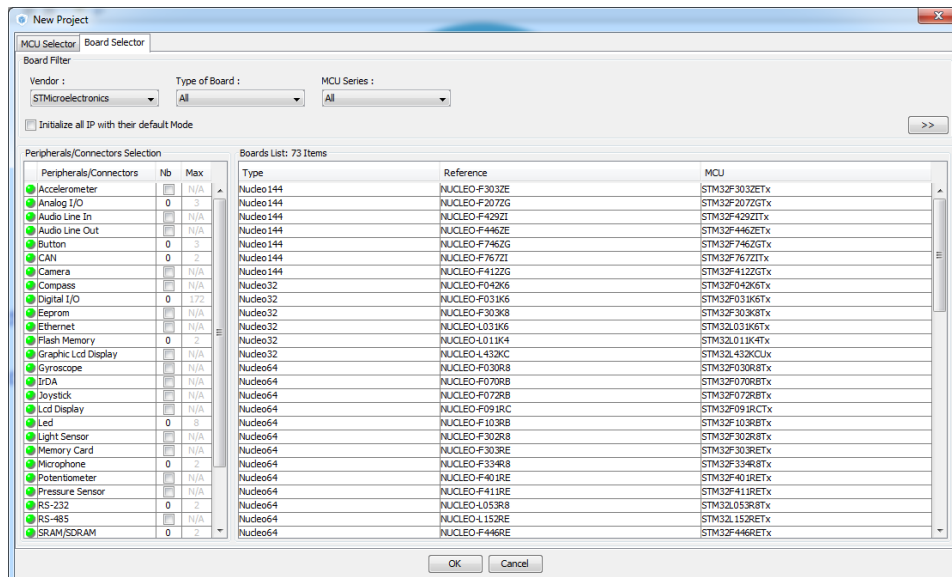


Figure 7 Board Selection

[Board Selector] 는 STM32 board type, series, peripheral 로 필터링이 가능합니다.

[Board Selector]의 경우 보드 상에 jumper 또는 solder bridge 에 의해 재설정되어진 상태의 보드 설정은 지원되지 않고 오직 기본 보드 설정만 제공됩니다.

보드가 선택되었을 때, [Pinout] view 는 선택된 보드의 LCD, button, communication interface, LED 등에 대한 Pin 할당에 따라 관련된 MCU part number 에 맞게 초기화 됩니다.

4 Main window

STM32 part number 또는 보드를 선택했거나 기존의 저장된 프로젝트를 불러왔을 때, 모든 STM32CubeMX 구성요소와 메뉴들이 나타납니다.

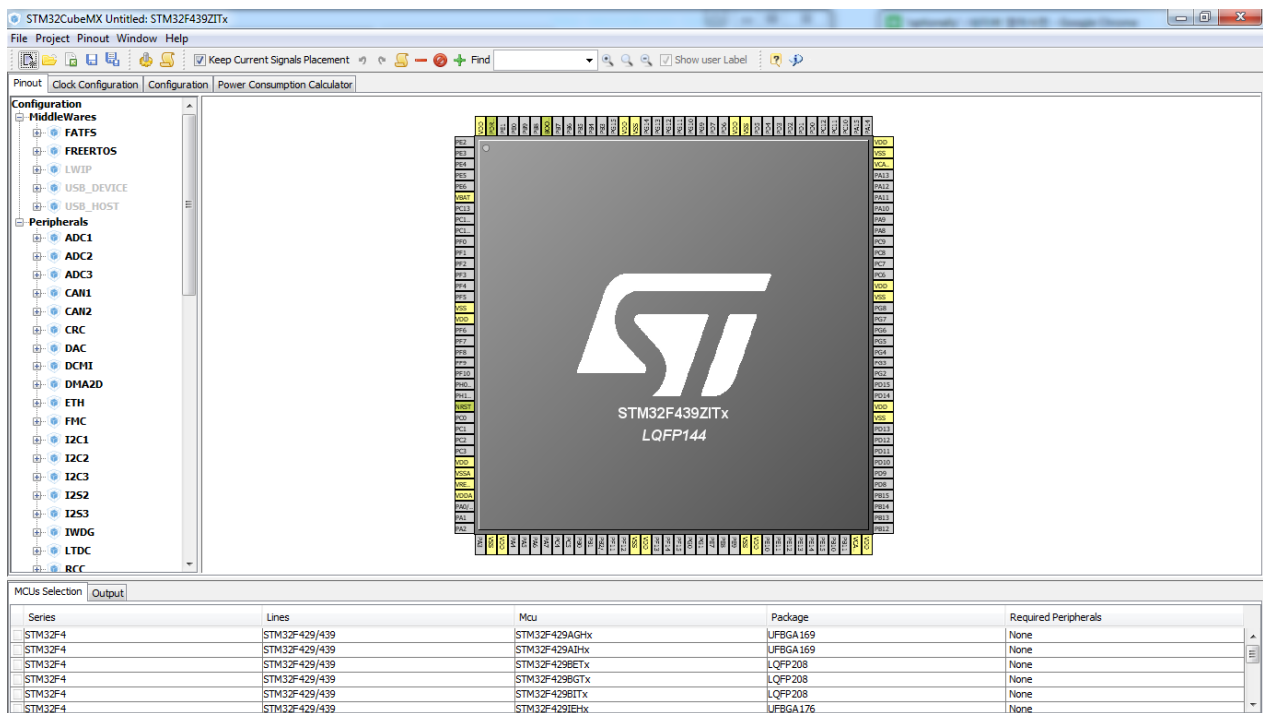


Figure 8 Main menu

peripheral default mode(Initialize all IP with their default mode)가 체크되지 않은 상태로 보드를 선택하면, 자동적으로 선택된 보드에 대한 pinout 을 설정합니다. 다만 peripheral mode 는 설정되지 않는 반면 GPIO 로 설정한 pin 들만 설정된 것으로 표시된다 (녹색으로 표시) . 그 다음에 사용자는 사용자의 어플리케이션에 요구되는 peripheral 을 peripheral menu tree 에서 수동으로선택하여 사용할 수 있습니다.

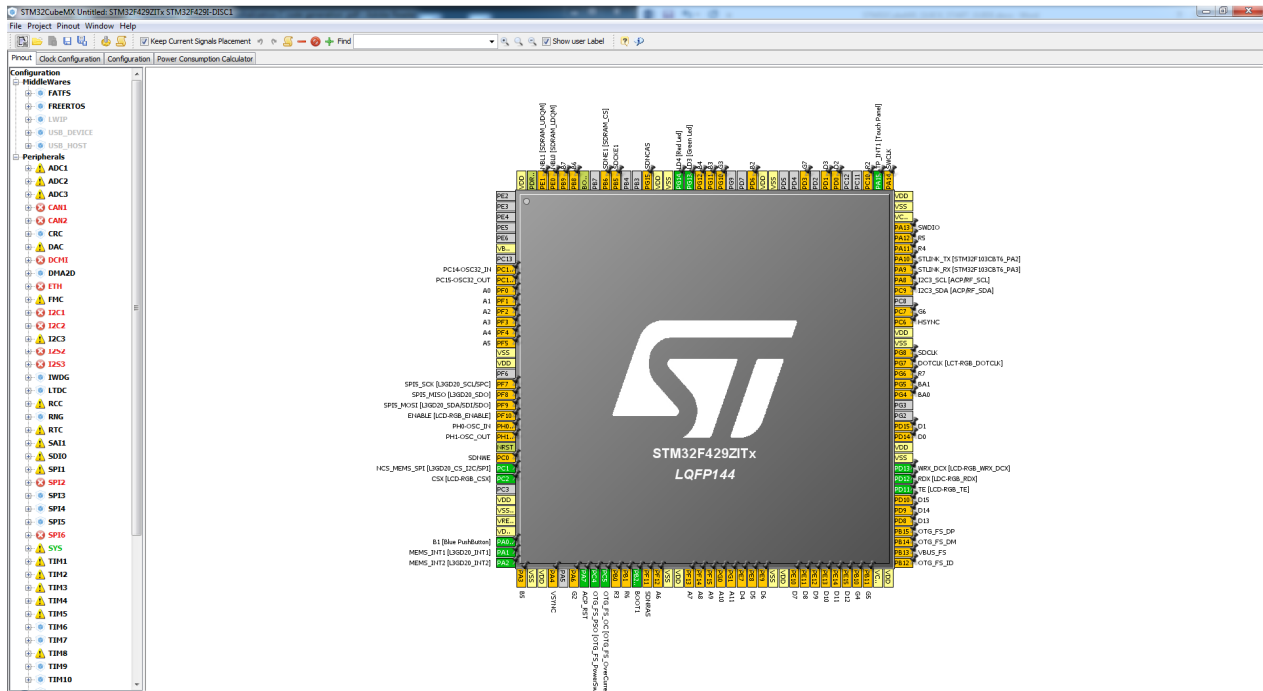


Figure 9 Main Window (No check the peripheral default mode)

peripheral default mode(Initialize all IP with their default mode)가 체크된 상태로 보드를 선택하면 자동적으로 선택된 보드 상에서 사용가능한 peripheral 에 대한 pinout 과 default mode 모두 설정합니다. 이것은 STM32CubeMX 는 사용자 어플리케이션과 관련된 것들 뿐만아니라 선택된 보드상에서 사용가능한 모든 peripheral 들에 대해 C 초기화 코드를 생성할 것입니다.

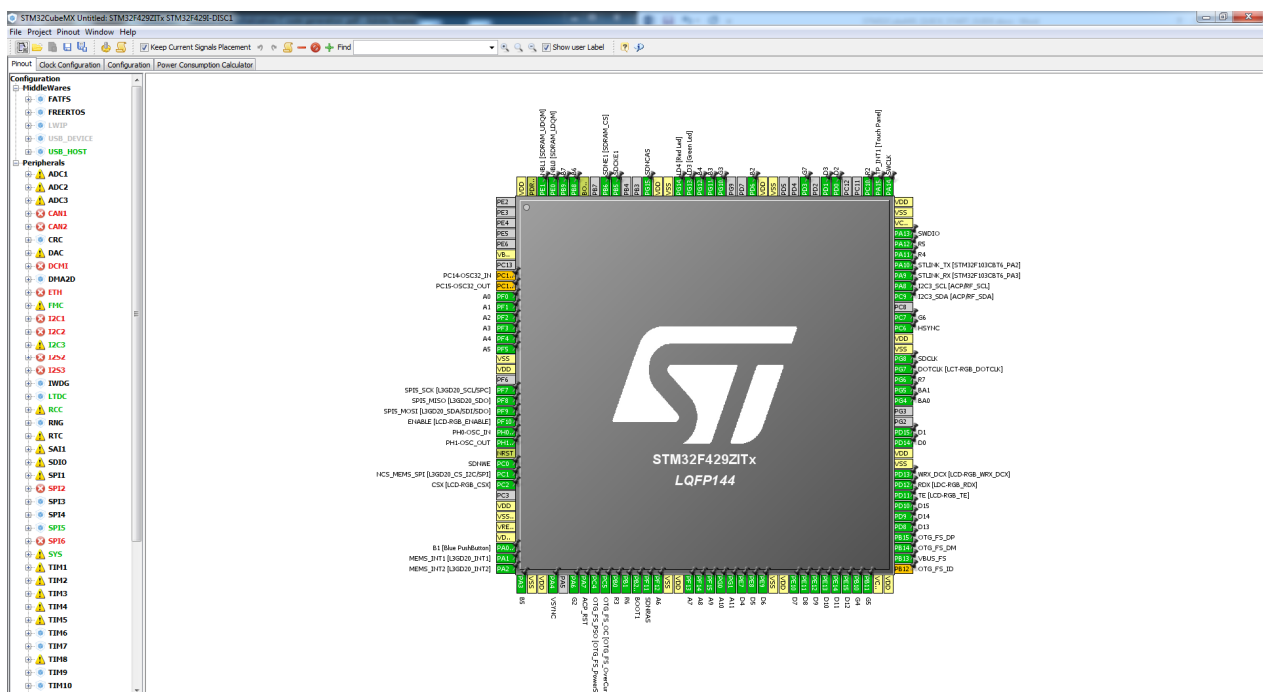


Figure 10 Main Window (Check the peripheral default mode)

4.1 Toolbar and menus






STM32CubeMX menu bar 에서는 다음과 같은 메뉴들을 사용가능합니다.

- File menu
- Project menu
- Pinout menu (Pinout view 가 선택되었을 때만 표시)
- Window menu
- Help menu

4.2 File menu

File menu 와 아이콘에 대한 설명은 아래 테이블(Table 1)을 참고하세요.




Table 1 File menu functions

Icon	Name	Description
	New project	지원되는 모든 MCU 들과 STMicroelectronics 보드들을 보여주는 새로운 프로젝트 창을 엽니다.
	Load Project	STM32CubeMX .ioc 설정 파일을 선택함으로써 기존의 STM32CubeMX 프로젝트 설정을 불러옵니다. *(주의) STM32CubeMX 버전이 업그레이드 되었을 때, 새 프로젝트를 읽어오기전 반드시 기존 프로젝트들을 반드시 backup 을 해야합니다.(특히, user code 를 포함하는 프로젝트의 경우)
	Import Project ...	설정들을 import 하고 import 된 설정파일을 선택하여 새로운 창을 엽니다.
	Save Project as ...	현재 프로젝트 설정(pinout, clock tree, IP, PCC)을 새로운 프로젝트로 저장합니다. 이 기능은 지정된 폴더에 사용자가 정의하는 파일 이름으로 .ioc 파일을 생성합니다.
	Save Project	현재의 프로젝트를 저장합니다.
No icon	Close Project	현재의 프로젝트를 닫고 welcome 페이지로 돌아갑니다.
No icon	Recent Projects >	가장 최근에 저장된 프로젝트 5 개를 보여줍니다.
No icon	Exit	필요에 따라 프로젝트를 저장할 지를 물어본 다음 STM32CubeMX 를 닫습니다.

4.3 Project menu

Table 2 Project menu

Icon	Name	Description
------	------	-------------

	Generate code	이 메뉴는 현재 설정(pinout, clocks, peripherals, middleware)에 대한 C 초기화 C 코드를 생성합니다. 이전에 정의된 것이 없다면 프로젝트 설정을 위한 창이 열립니다. - Note : STM32CubeMX 의 새로운 버전으로 업그레이드 할 때, 현재 프로젝트를 백업하는 것을 권장합니다. 사용가능한 새로운 펌웨어 라이브러리 버전이 있다면 새로운 펌웨어 라이브러리로 이전하라는 메시지가 표시됩니다.
	Generate Report ⁽¹⁾	이 메뉴는 현재의 프로젝트 설정을 pdf 파일 또는 text 파일로 생성합니다.
	Settings	이 메뉴는 프로젝트명, 폴더, toolchain 설정, C code 생성 옵션을 설정하기 위한 설정 창을 엽니다.

1. 프로젝트가 저장되어 있다면 Report 들은 프로젝트 .ioc 설정 파일과 같은 위치에 생성됩니다. 그렇지 않으면 사용자가 저장될 폴더와 .ioc 파일로 프로젝트 설정을 저장할지 여부를 선택할 수 있습니다.

4.4 Pinout menu

Pinout menu 와 Sub-menu 들은 pinout 탭이 선택되었을 때만 사용가능합니다 (Figure 11 참조). 그렇지 않으면 pinout menu 와 sub-menu 는 hidden 상태가 되어 있습니다 (Figure 12 참조).

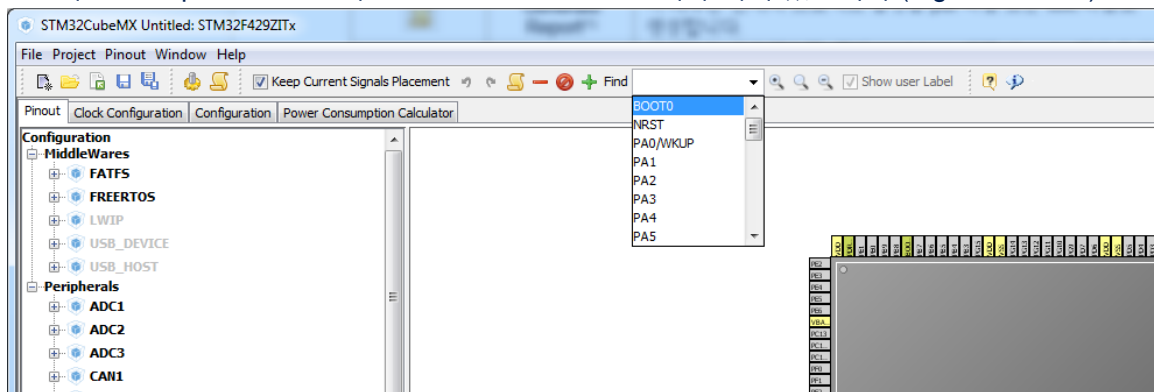


Figure 11 Pinout menu (Pinout 탭이 선택된 상태)

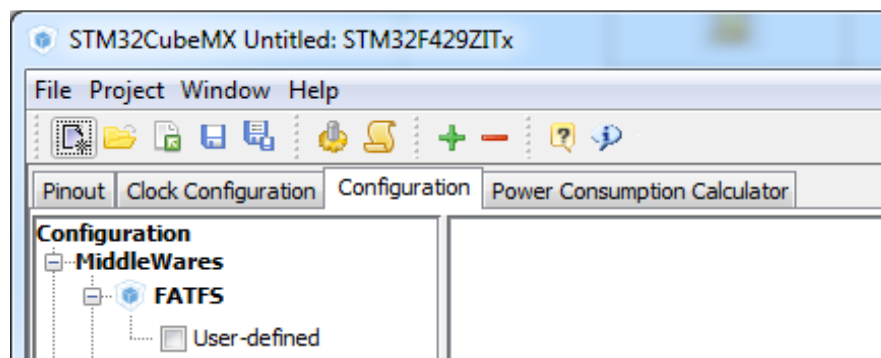
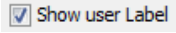







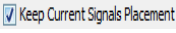


Figure 12 Pinout menus (Pinout 탭이 선택되지 않은 상태)

Pinout menu 와 icon 들의 설명은 Table 3 에서 참조하세요.

Table 3 Pinout menu

Icon	Name	Description
	Undo	이전 설정 단계로 하나씩 되돌립니다.
	Redo	Undo 했던 설정을 다시 되돌립니다.
No icon	Pins/Signals Options	Pin 의 Signal name 과 함께 구성된 모든 Pin 의 목록과 사용자가 각 Pin 에 대해 Label name 을 지정한 Label filed 의 목록을 보여주는 창을 엽니다. 이 메뉴를 활성화 하기 위해서 적어도 하나의 Pin 은 설정되어있어야 합니다.
	Pinout search field	사용자가 Pinout view 에서 signal label, signal name, pin name 으로 찾을 수 있도록 합니다. 검색 조건과 일치하는 Pin 또는 Pin 설정은 Chip view 에서 깜빡입니다. 깜빡임을 멈추기 위해서는 Chip view 를 클릭하세요.
	Show user labels	Pin 에 할당된 signal name 대신에 사용자가 정의한 Label 로 Chip view 에 보여줍니다.
No icon	Clear Pinouts	Pinout 창에 사용자가 지정한 pinout 설정들을 초기화 합니다.. 이 기능은 설정 창에서 pinout 에 영향이 있는 IP 들을 초기화 합니다.
No icon	Clear Single Mapped Signals	관련된 모드를 가지지 않는 Signal 들을 초기화 합니다.(오렌지색으로 표시되고 Pin 으로 고정되지 않은 상태)
No icon	Set unused GPIOs	사용하지 않는 전체 GPIO pin 들 중에서 설정 할 GPIO 들의 수를 지정하는 창을 열어줍니다. 지정할 수 있는 모드는 Input, Output 또는 Analog 입니다.(소모전류 최적화를 위해 이 설정을 권장) *(주의) 이 메뉴를 사용하기 전에 반드시 Debug pin 들은 MCU debug 기능에 접근하도록 설정되었는지 확인해야합니다.
No icon	Reset used GPIOs	설정된 전체 GPIO pin 들 중에 설정을 리셋하기 위한 GPIO 의 수를 지정하는 창을 열어줍니다.
	Generate csv text pinout file	.csv text 파일로 Pin 설정을 생성.
	Collapse All	IP / Middleware tree view 에 열려있는 tree 를 모두 접습니다.
	Disable Modes	Enable 된 모든 Peripheral 과 Middleware 들은 “Disabled”로 리셋합니다. Enable 상태(녹색)의 Peripheral 과 Middleware 들은

		“Unused”(회색)으로 리셋된다. IP 들과 Middleware label 들은 녹색에서 검정(unused) 또는 회색(not available)으로 변경된다.
	Expand All	기능상의 모든 모드들을 보여주기 위해 IP / Middleware tree view 를 모두 열어줍니다.
	Zooming in	Chip pinout diagram 을 확대.
	Best Fit	적절한 크기로 chip pinout diagram 을 조절.
	Zooming out	Chip pinout diagram 을 축소
	Keep current signals Placement	Toolbar 에서만 사용가능. 새로운 IP 동작 모드를 적용하기 위해 Pin 할당이 변경되는 것을 보호합니다. 체크박스를 “unchecked”상태로 두고 각 Pin 할당을 차단할 수 있는 새로운 pin 고정 기능을 사용하는 것이 권장됩니다.

4.5 Window menu

Window menu 는 output 기능을 사용할 수 있도록 합니다 (Table 4 참조).





Table 4 Pin menu


Name	Description
Outputs	<p>STM32CubeMX main 창 하단에 MCU selection 창이 열립니다.</p> <p>STM32CubeMX main 창 하단에 두가지 탭이 열립니다 :</p> <ul style="list-style-type: none"> - MCUs selection Tab : MCU selector 를 통해 선택된 사용자 조건과 일치하는 MCU 들을 나열. - Outputs Tab : 사용자가 사용 중 발생하는 STM32CubeMX 경고, 에러 메시지들을 표시.

4.6 Help menu

Help menu 와 아이콘 의 설명은 Table 5 를 참조하세요.

Table 5 Help menu

Icon	Name	Description
	Help Content	STM32CubeMX user manual 을 보여줍니다.
	About...	버전 정보를 보여줍니다.
	Check for Updates	다운로드 가능한 software 와 firmware 의 release 업데이트들을 보여줍니다.
	Install New Libraries	설치 가능한 모든 STM32CubeMX 와 firmware 의 release 를 보여줍니다. 녹색 체크박스는 PC 에 이미 설치된 최신상태의 것들을 나타냅니다.

	Updater Settings ...	Software 와 firmware release 들을 저장하는 Repository folder, internet 연결을 위한 proxy 설정, 자동 업데이트등을 설정할 수 있는 updater setting 창을 열어줍니다.
---	-----------------------------	---

5 Clock tree configuration view

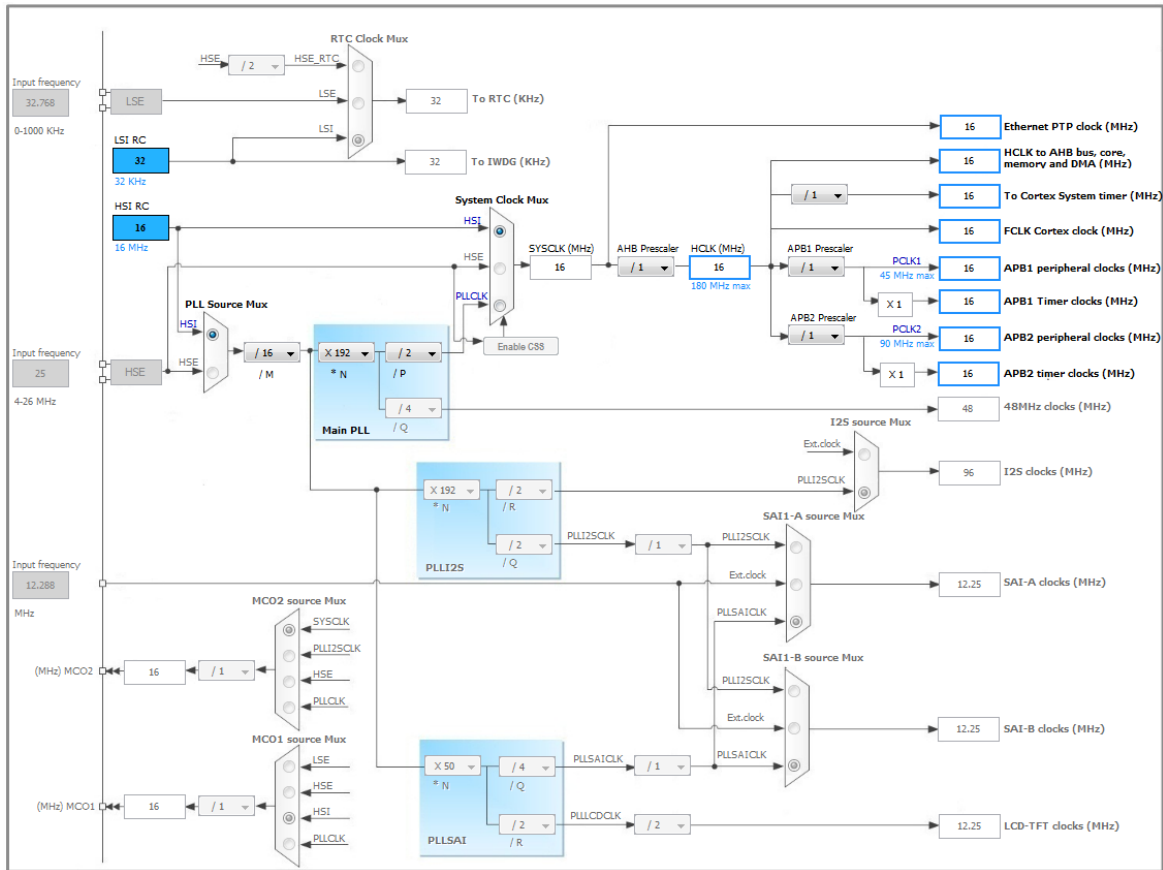


Figure 13 Clock tree view

STM32CubeMX 는 사용자에게 의해 선택된 prescaler 들과 clock source 에서 system, CPU 그리고 AHB/APB 버스 주파수를 자동적으로 생성합니다. 또한 잘못된 설정들의 최대 및 최소 조건의 동적 검증을 통해 검출되고 붉은색으로 표시됩니다. 사용자의 주파수 선택은 몇몇 parameter(예, UART baudrate limitation)에 영향을 줄 수 있습니다.

STM32CubeMX 는 각 Peripheral 클럭에 대한 초기화 C 코드를 생성하기 위해 Clock tree view 에서 정의된 clock 설정을 사용합니다. 클럭 설정들은 project 에 main.c 와 stm32f4xx_hal_conf.h 내에 RCC 초기화 부분과 같이 생성된 C 코드에서 수행됩니다.(HSE, HSI 그리고 external clock 의 상수 값은 코드 상에서 Hertz 로 표기됩니다.)

다음의 sequence 는 STM32F4 기반의 어플리케이션에서 요구되는 클럭설정을 어떻게 설정할 것인가에 대한 설명합니다.

1. Clock tree view 를 볼수 있도록 Clock configuration 탭을 클릭합니다.(Figure 13 참조)
Internal(HSI, LSI), system(SYSCLK) 그리고 peripheral 클럭 주파수 필드는 편집할 수 없습니다.
System 과 peripheral 클럭은 clock source 을 선택하고 PLL, prescaler 그리고 multiplier 들을
추가적으로 사용함으로써 조절할 수 있습니다.
2. 먼저 microcontroller 의 system clock 를 공급할 clock source(HSE, HSI 또는 PLL CLK)를
선택합니다.
예를들어 Figure 14 은 내부 16MHz 클럭을 사용하기 위해 HSI 를 선택한 것을 보여줍니다.

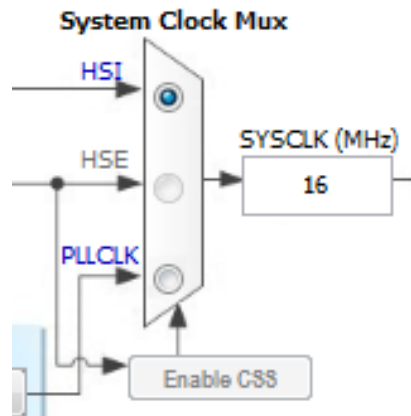


Figure 14 HSI 클럭을 선택

외부 클럭(HSE 또는 LSE)를 사용하기 위해 외부 클럭과 관련된 pin 들이 외부 클럭 crystal 을
연결하는데 사용되도록 RCC peripheral 이 pin out view 에서 구성되어야 한다.

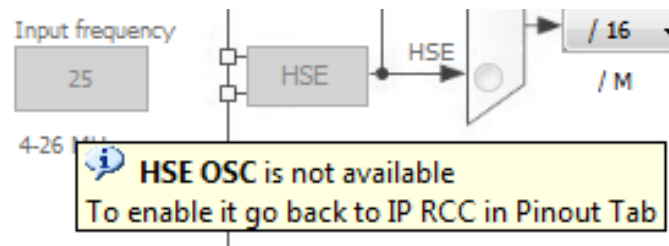


Figure 15 비활성화 되어 있는 HSE clock source

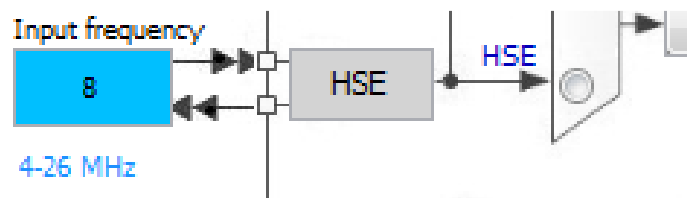


Figure 16 활성화된 HSE clock source

System clock source 로 PLL 을 사용할 경우, PLL 입력 clock source 로 HSI 또는 HSE 를 선택할 수 있습니다.

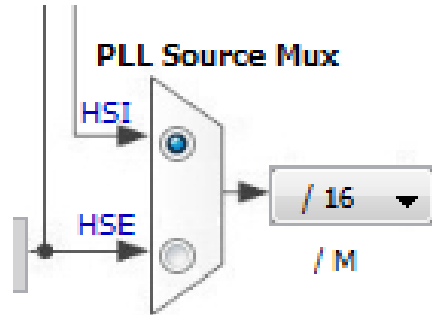


Figure 17 PLL clock source 선택

3. PLL 과 prescaler 를 사용하지 않고 HSI 를 사용한 16MHz 는 core 와 peripheral 클럭들에 그대로 설정됩니다.

Note : 경우에 따라 PLL 과 prescaler, Multiplier 들을 사용하여 system 과 peripheral 클럭들을 추가로 조절합니다. system 클럭으로부터 독립된 다른 clock source 들은 다음과 같이 설정 될 수 있습니다 :

- USB OTG FS, RNG 그리고 SDIO 클럭은 독립된 PLL 출력으로 공급됩니다.
- I2S peripheral 은 내부 클럭 PLLI2S 에 속해있거나 독립된 외부 clock source 로 공급됩니다.
- USB OTG HS 와 Ethernet 클럭은 외부 clock source 로부터 공급됩니다.

4. 경우에 따라 외부 회로에 두개의 클럭을 출력할 수 있는 Microcontroller Clock Output(MCO) pin 을 위해 prescaler 를 설정합니다.
5. Project 를 저장합니다.
6. 프로젝트 환경설정을 진행하기 위해 Configuration 탭으로 이동합니다.

6 Configuration view

STM32CubeMX Configuration 창은 (Figure 18 참조) 소프트웨어 설정이 가능한 모든 요소들(GPIO, peripheral, middleware)을 개략적으로 나타냅니다. 클릭가능한 버튼들은 생성할 코드에 포함되는 초기화 parameter 요소들의 설정 옵션들을 선택할 수 있도록 합니다.

- 녹색 체크마크(Green Checkmark) : 올바른 설정 상태
- 경고표시(Warning sign) : 불완전하지만 설정 가능한 상태
- 붉은색 'X' 표시(Red cross) : 잘못된 설정 상태

Note : Pinout 에 영향이 있는 GPIO 와 peripheral 모드는 오직 pinout view 에서만 설정 가능.

Configuration view 에서 MCU 는 왼쪽 패널에서 IP tree 를 보여주고 오른쪽 패널에서 Middleware, Multimedia, Connectivity, Analog, system, control 카테고리 구성된 IP 의 리스트를 보여준다.

각 peripheral 은 설정을 편집하기 위한 전용 버튼을 가진다(Figure 18 참조).

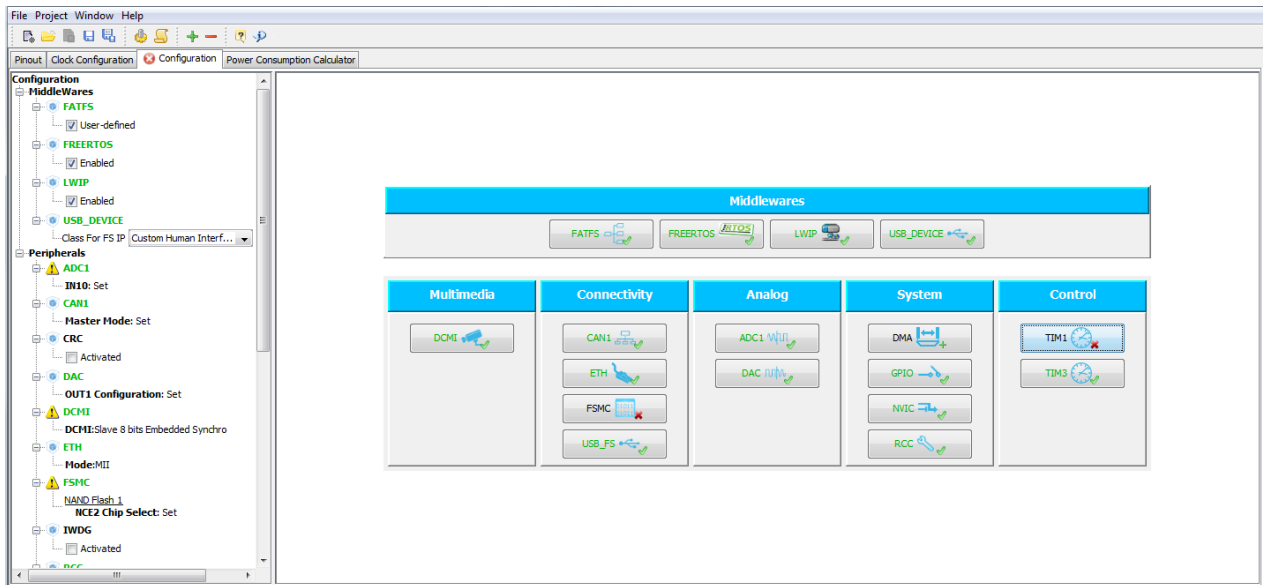


Figure 18 STM32CubeMX Configuration view

6.1 IP and Middleware Configuration window

이 창은 configuration 패널에서 IP instance 또는 Middleware 이름을 클릭하면 열립니다 그리고 선택된 동작모드에서 IP 또는 Middleware 를 초기화 하기 위해 요구되는 parameter 들을 설정할 수 있습니다. 이 설정은 설정과 일치하는 초기화 C 코드를 생성하기 위해 사용됩니다. (Figure 19 참조. IP configuration 창에 대한 예시)

configuration 창은 다음과 같은 몇가지 탭을 포함합니다 :

- **Parameter settings** : 선택된 peripheral 또는 Middleware 를 위한 라이브러리 전용 parameter 들을 설정.
- **NVIC, GPIO and DMA setting** : 선택된 peripheral 을 위한 parameter 들을 설정(4.3.5 참조).
- **User constants** : 전체 프로젝트를 위해 하나또는 몇 개의 사용자 정의 상수(constant)를 생성 (4.3.2 참조) .

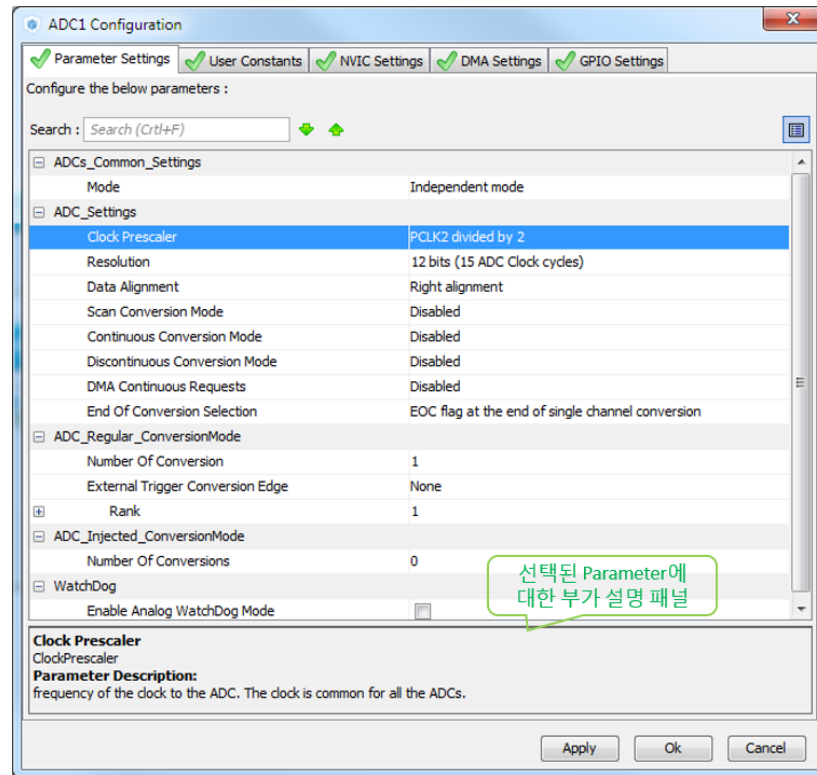


Figure 19 IP Configuration window (STM32F4 series)

6.2 User Constants configuration window

User Constants 창은 사용자 지정 상수를 정의하는 것이 가능합니다. 설정한 상수들은 mxconstants.h 파일 내 STM32CubeMX 사용자 프로젝트에 자동으로 생성됩니다. 정의된 상수는 peripheral 과 middleware parameter 들을 설정하기 위해 사용될 수 있습니다.

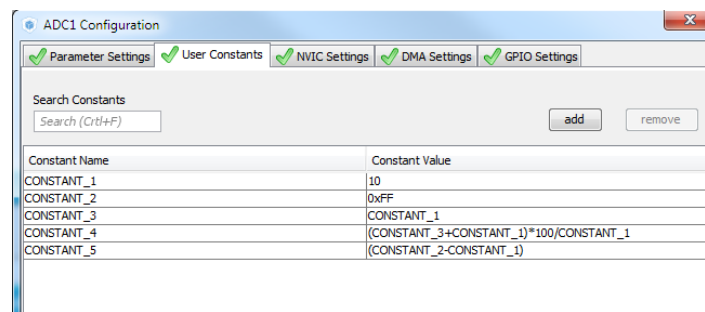


Figure 20 User Constants window

```

/* Includes -----*/
/* USER CODE BEGIN Includes */
/* USER CODE END Includes */

/* Private define -----*/
#define CONSTANT_1 10
#define CONSTANT_2 0xFF
#define CONSTANT_3 CONSTANT_1
#define CONSTANT_4 (CONSTANT_3+CONSTANT_1)*100/CONSTANT_1
#define CONSTANT_5 (CONSTANT_2-CONSTANT_1)
  
```

Figure 21 생성된 mxconstants.h 파일

6.3 GPIO Configuration window

GPIO pin 설정을 하기 위한 GPIO configuration 창을 열기 위해 Configuration 창에서 GPIO 탭을 클릭합니다(Figure 22 참조). 이 설정에는 일부 peripheral 환경설정 에 대해 적합하지 않을 수 있는 기본값들로 채워져 있습니다. 특히 GPIO speed 가 peripheral 의 통신 speed 에 적합한지, 필요한 곳에 내부 풀업이 선택되어 있는지 확인해야 합니다.

Note : GPIO 설정은 IP instance 환경설정 창에서 전용 GPIO 창을 통해 특정 IP instance 에 접근 될 수 있습니다. 게다가, GPIO 들은 output mode 로 설정될 수 있습니다(기본적으로 output level). 생성된 코드는 그에 따라 업데이트 될 수 있습니다.

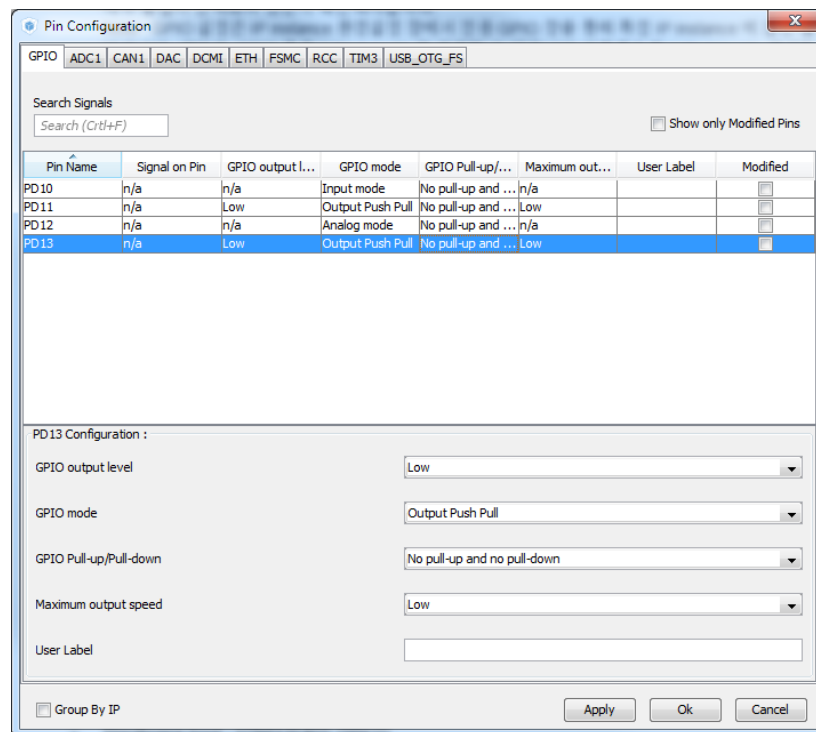


Figure 22 GPIO Configuration window - GPIO selection

6.4 DMA Configuration window

DMA 환경설정 창을 열기 위해 configuration 창에서 DMA 를 클릭합니다.

이 창은 MCU 상에서 사용가능한 전반적인 DMA 를 설정할 수 있습니다. DMA 인터페이스는 CPU 가 동작하는 동안 memory and peripheral 사이에 데이터 전송을 수행합니다. 또한 지원여부에 따라 memory to memory 전송을 설정 할 수 있습니다.

Note : USB 또는 Ethernet 과 같은 일부 IP 들은 IP 환경설정 창 또는 기본 설정에 의해 활성화 된 자신의 DMA controller 를 가집니다.

DMA 환경설정 창에서 “Add”를 클릭하면 peripheral 신호에 매핑될 수 있는 설정가능한 DMA

request 들 사이에서 선택하도록 제안하는 콤보박스과 함께 DMA 환경설정 창의 끝에 새로운라인을 추가합니다(Figure 23 참조).

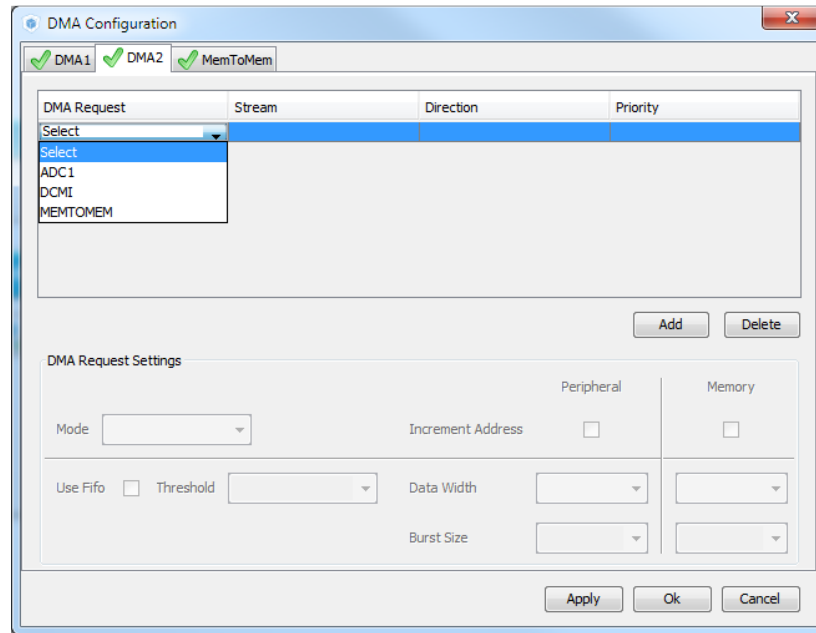


Figure 23 DMA Configuration

DMA request 를 선택하면 자동적으로 모든 stream 중 사용가능한 stream, direction(데이터 전송방향) 그리고 priority(우선순위)를 할당합니다. DMA 채널이 선택되었을 때, start address 와 같은 DMA 전송 run-time parameter 들을 충분히 기술하는 것은 application code 에 따릅니다. DMA request 는 peripheral 들 과 memory 들 사이에 데이터를 전송하기 위한 stream 을 받기위해 사용됩니다(Figure 24 참조). Stream 우선순위는 다음 DMA 전송을 선택하기 위한 stream 을 결정하기 위해 사용 될 것입니다.

DMA controller 들은 소프트웨어 우선순위 먼저, 그리고 소프트웨어 우선순위가 동일한 경우, stream number 에 의해 주어진 하드웨어 우선순위를 사용하여 dual priority system 을 지원합니다.

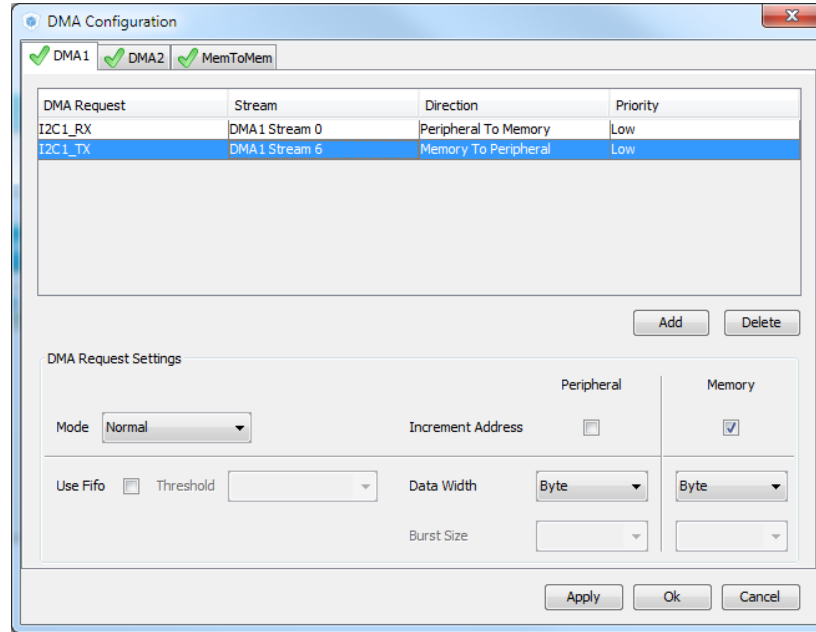


Figure 24 DMA Configuration

6.5 NVIC Configuration window

NVIC(Nested Vector Interrupt Controller) 창을 열기 위해 Configuration 창에서 NVIC 를 클릭합니다(Figure 25 참조).

Interrupt unmasking 과 interrupt handler 는 아래 두개의 탭에서 관리됩니다:

- **NVIC 탭**은 NVIC controller 에서 peripheral interrupt 를 활성화 하고 우선순위를 설정할 수 있도록 합니다.
- **Code generation 탭**은 코드생성과 관련된 interrupt 에 대해 옵션을 선택할 수 있도록 합니다.

▶ NVIC 탭을 사용한 interrupt 활성화

NVIC 창(Figure 25 참조) 사용하는 모든 interrupt 들을 보여주지 않습니다. 다만 pinout 과 configuration 창에서 선택된 IP 에 대한 사용가능한 interrupt 들만 보여줍니다. System interrupt 들은 나타나지만 비활성화 시킬 수는 없습니다.

활성화된 interrupt 를 필터링하거나 하지 않도록 [Show only enabled interrupts] 체크박스를 체크하거나 하지 않을 수 있습니다.

검색어에 따라 Interrupt vector table 을 검색할 수 있도록 [Search] 필드를 사용할 수 있습니다. 예를들어 pinout 창에서 UART IP 를 활성화하고 NVIC search 필드에서 UART 를 입력 후 녹색 화살표를 클릭하면 UART 와 관련된 모든 NVIC 를 표시해줍니다.

peripheral interrupt 가 활성화되면 해당 peripheral 에 대한 HAL_NVIC_SetPriority 와 HAL_NVIC_EnableIRQ function call 을 생성합니다.

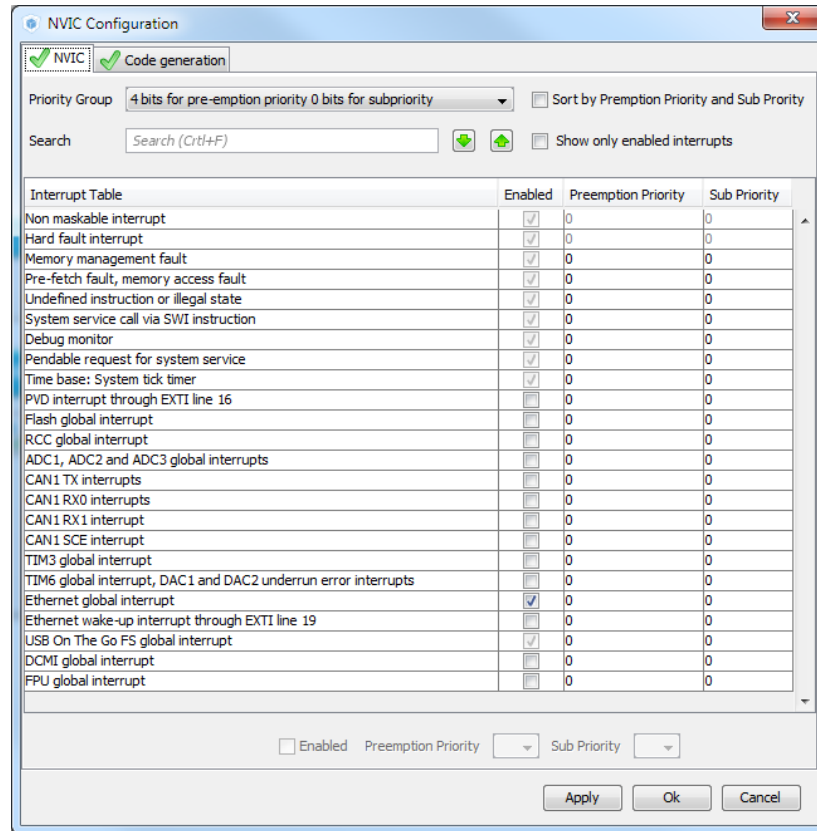


Figure 25 NVIC Configuration tab - FreeRTOS disabled

FreeRTOS 가 활성화 되었을 때, Figure 26 과 같이 추가 열이 나타납니다. 이 경우에 FreeRTOS API 들을 호출하는 모든 interrupt service routine(ISR)들은 LIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY parameter 에 정의된 우선순위 보다 낮은 우선순위를 가져야합니다.

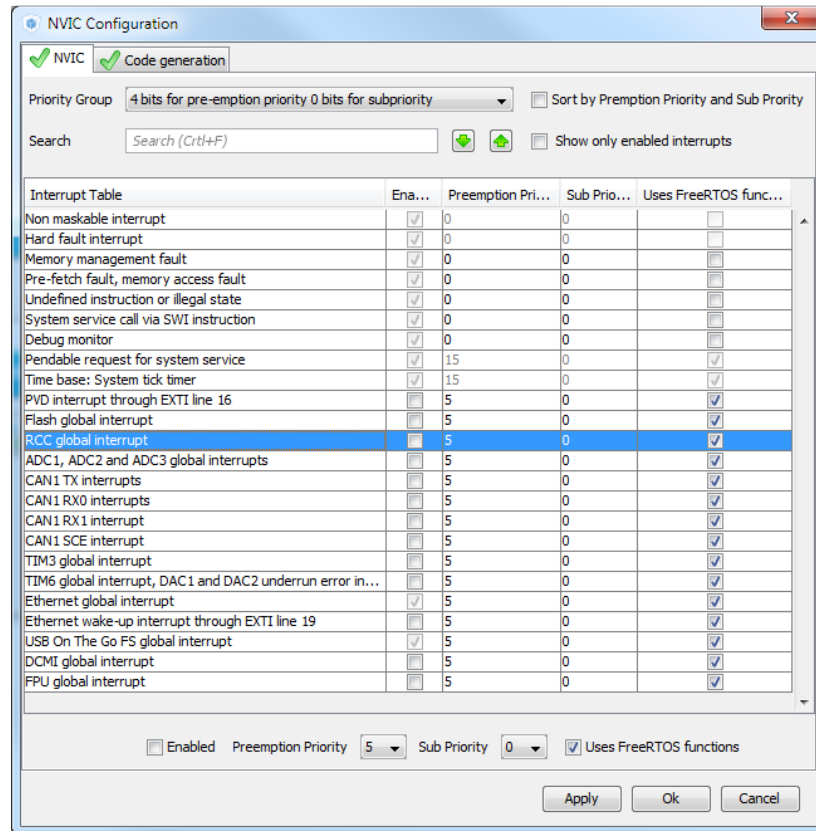


Figure 26 NVIC Configuration tab - FreeRTOS enabled

IP 의 전용 interrupt 는 IP configuration 창에서 NVIC 창을 통해 설정할 수 있습니다.

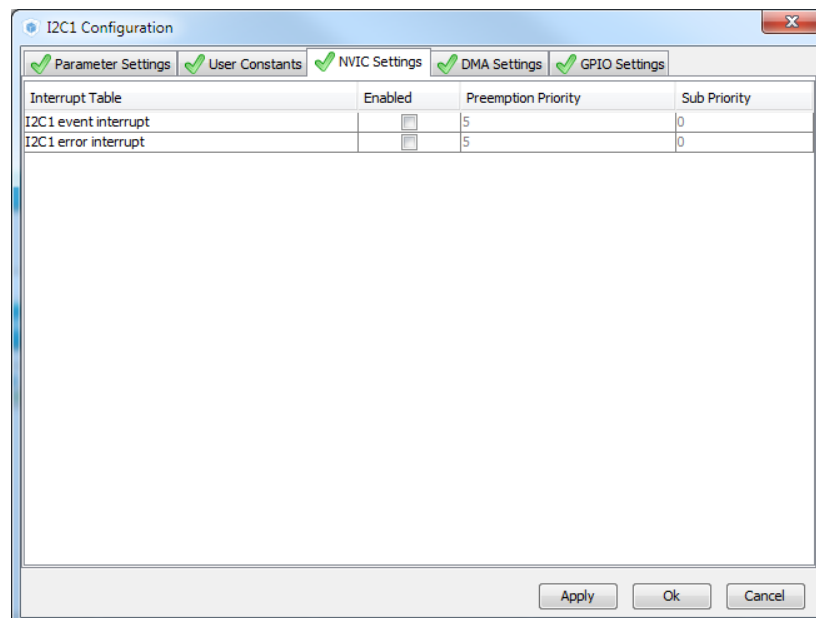


Figure 27 I2C NVIC Configuration window

STM32CubeMX NVIC configuration 은 우선순위 그룹(priority group) 선택, interrupt 활성화/비활성화, interrupt 우선순위 레벨(priority levels : preemption and sub-priority level) 설정으로 구성되어 있습니다.

1. Priority group 선택

몇몇 비트는 NVIC priority level 을 정의 할 수 있도록 합니다. 이 비트들은 두가지 priority type(preemption priority 와 sub-priority)과 일치하는 두 priority group 으로 나누어져있습니다. 예를들어, STM32F4 의 경우 NVIC priority group 0 는 0-bit preemption 과 4-bit sub-priority 와 일치합니다.

2. Interrupt 테이블에서 하나 또는 둘 이상의 interrupt vector 들을 선택하기 위해 하나 또는 둘 이상의 열을 선택합니다. 한번에 하나씩 또는 여러 개를 동시에 vector 들을 설정하기위해 아래의 interrupt 테이블 위젯을 사용합니다.

- **Enable checkbox** : interrupt 를 활성화/비활성화 시키기 위해 체크박스를 체크하거나 해제합니다.
- **Preemption priority** : priority level 을 선택합니다. Preemption priority 는 다른 interrupt 들을 차단할 수 있는 하나의 interrupt 능력 즉, preemption priority 의 level 이 높을 수록 interrupt 우선 실행하는 높은 선점 순위를 정의합니다.
- **Sub-priority** : priority level 을 선택합니다. Sub-priority 는 interrupt 의 우선순위를 정의하는데 동일한 preemption priority 를 가진 interrupt 가 동시에 발생한 경우 sub-priority 의 level 이 높은 interrupt 를 먼저 실행합니다.
- 변경 사항을 저장하기 위해 **Apply** 버튼을 클릭 후 **OK** 버튼을 클릭합니다.

6.6 FreeRTOS middleware configuration view

STM32CubeMX FreeRTOS configuration 창을 통해 사용자는 real-time OS 어플리케이션에 요구되는 모든 자원(resource)들 설정할수 있고 과 해당하는 heap 을 지정할 수 있습니다. FreeRTOS 요소들은 CMSIS-RTOS API 함수를 사용한 코드에서 정의되고 생성됩니다. FreeRTOS 생성은 다음과 같은 절차를 가집니다.

1. Configuration 탭에서 Tree view 에 있는 FreeRTOS 를 활성화합니다.
2. FreeRTOS 설정 창을 열기 위해 Configuration 창에 FreeRTOS 를 클릭합니다(Figure 28 참조).

User Constants 탭을 제외한 모든 탭은 task, timer, queue, semaphore 와 같은 환경설정 parameter 들과 object 들을 설정할 수 있도록 합니다.

Config parameters 에 값들은 kernel 과 software 설정들을 정의할 수 있도록 합니다.

Include parameters 탭은 사용자의 어플리케이션에 따라 요구되는 API 함수들만 선택하여 코드 사이즈를 최적화 시킬 수 있도록 합니다.

Config 와 Include parameter 들은 FreeRTOSConfig.h 파일에 포함됩니다.

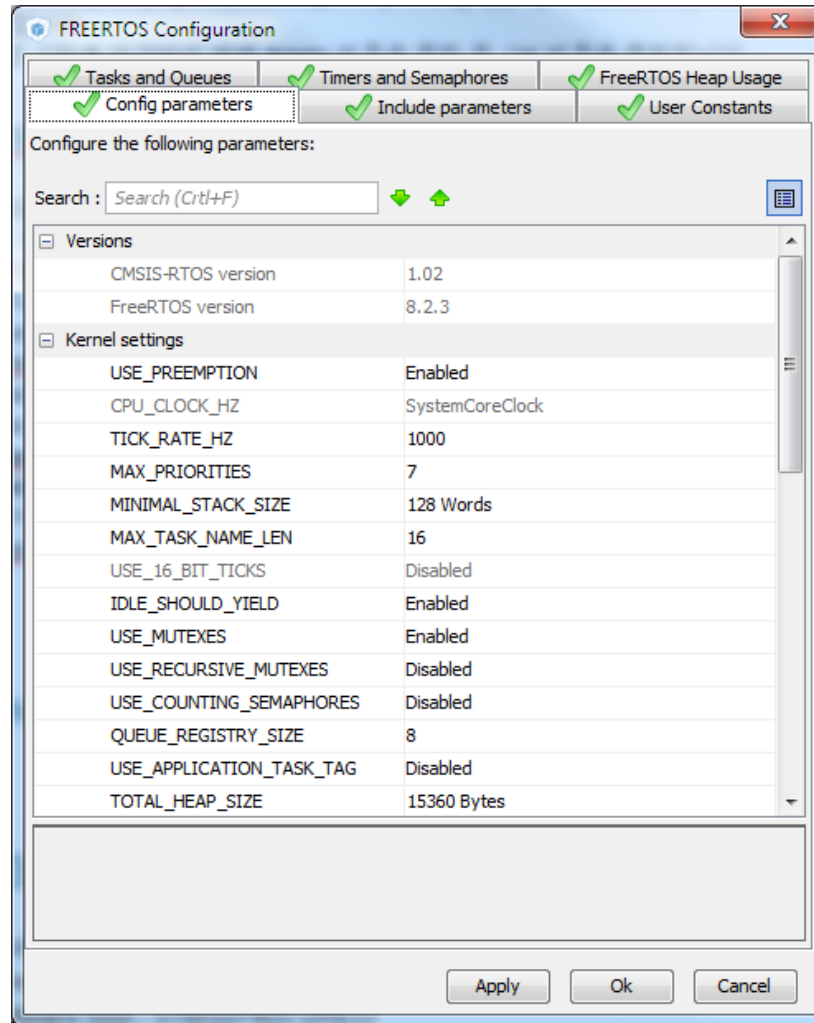


Figure 28 FreeRTOS Configuration view

7 Power Consumption Calculator (PCC) view

임베디드 시스템 어플리케이션의 지속적인 증가에 따라 소비전력(power consumption)은 중요한 관심사가 되었습니다. 소비전력을 초소화 시키기 위해 STM32CubeMX 는 **Power Consumption Calculator(PCC)** 탭 (Figure 29 참조)를 제공합니다. 이것은 주어진 microcontroller, battery 모델 그리고 사용자가 정의한 power sequence 에 따라 다음과 같은 결과를 제공합니다.

- **평균 소모전류(Average current consumption)**

: 소비전력 값은 데이터시트로부터 가져오거나 사용자 지정 버스 또는 core 주파수로부터 추가될 수 있습니다.

- **배터리 수명(battery life)**

- **평균 DMIPs**

: DMIPs 값은 MCU 의 데이터시트로부터 직접 가져오는 값으로 추가되거나 추정되는 값이 아닙니다.

- **최대 주변온도(Maximum ambient temperature : T_{AMAX})**

: 칩 내부 소비전력, 패키지 종류, 105°C 의 최대 접합온도(maximum junction temperature)에 따라 동작하기 좋은 상태를 보증하기 위해 최대 주변온도를 계산합니다.

현재 T_{AMAX} 구현은 I/O 소비를 고려하지 않습니다.정확한 T_{AMAX} 추정을 위해 I/O 소모전류는 Additional Consumption 필드를 사용하여 명시되어야 합니다. I/O 의 동적 소모전류에 대한 공식은 microcontroller 의 데이터시트에 명시되어 있습니다.

PCC view 는 개발자가 임베디드 어플리케이션 소모전류의 추정치를 예상할 수 있도록 하고 각각의 PCC power sequence 단계에서 아래와 같은 방법으로 추정된 값보다 더 낮출 수 있도록 도와줍니다 :

- 가능할 때에 low power mode 를 사용하도록 합니다.
- 각 단계 별 요구사항을 기반으로 clock source 와 주파수를 조절합니다.
- 각 단계에 필요한 peripheral 만 활성화합니다.

각 단계에서 사용자는 배터리 대신에 사용가능한 전원으로 VBUS 를 선택할 수 있습니다. 이것은 배터리 수명에 영향을 미칠 것 입니다. 서로 다른 전압레벨에서 소비전력 측정이 가능하다면, STM32CubeMX 역시 전압 값의 선택을 제안할 것입니다(Figure 29 참조).

Power consumption sequence 만들기

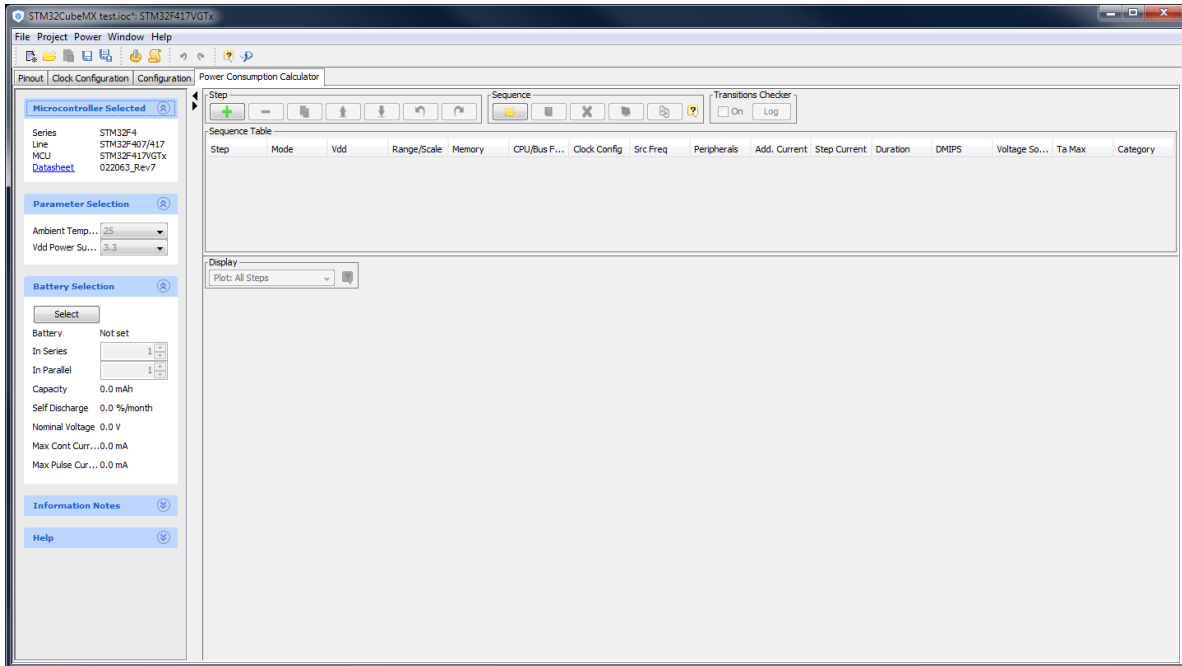


Figure 29 Power Consumption Calculator default view

- VDD 값 선택하기

복수 선택이 가능한 경우 사용자는 VDD 값을 선택해야합니다.

- 배터리 모델 선택하기(선택사항)

선택적으로 사용자는 배터리 모델을 선택할 수 있습니다. 이것 역시 소비전력 시퀀스가 구성될 때 할 수 있습니다.

사용자는 이미 정의되어 있는 배터리를 선택하거나 사용자의 어플리케이션에 가장 적합한 새로운 배터리를 추가하여 선택할 수 있습니다.

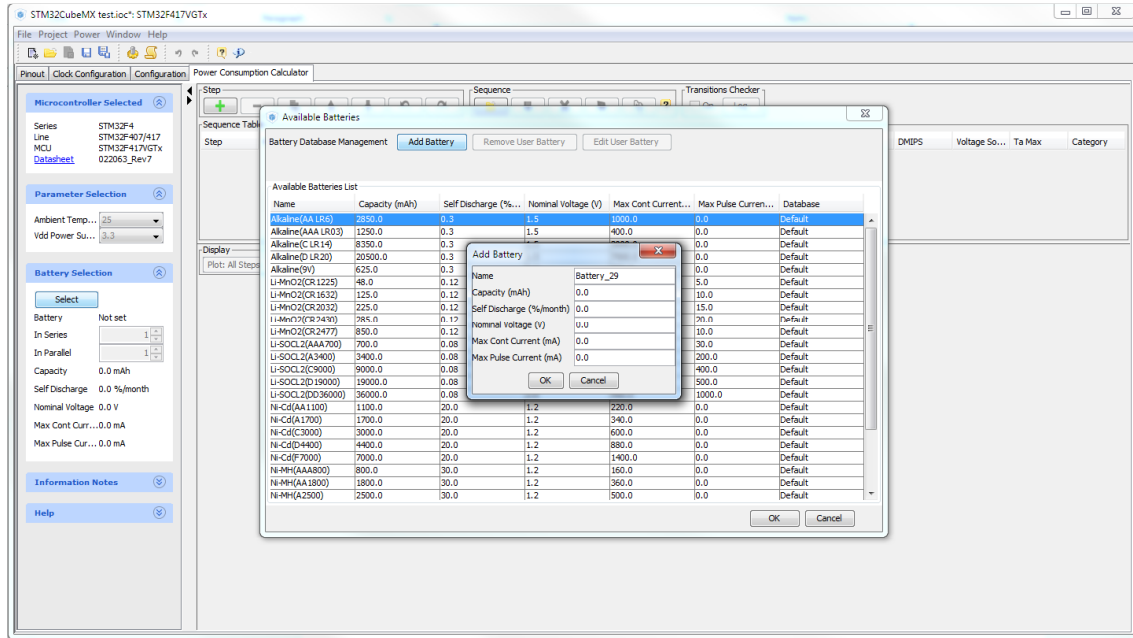


Figure 30 Battery selection

- Sequence Step 관리하기

각 Step 들은 step 버튼들(step 추가, 삭제, 복사, 이동)을 사용하여 sequence 내에서 재구성할 수 있습니다(Figure 31 참조).



Figure 31 Step management functions

- Step 추가

새로운 Step 을 추가하기 위해 아래와 같이 두가지 방법이 있습니다.

- power Consumption 패널에서 Add 를 클릭.빈 step 설정 상태로 **New Step** 창이 열립니다.
- Sequence 테이블에서 하나의 step 을 선택 후 **Duplicate** 를 클릭합니다. 복사된 step 설정상태로 **New Step** 창이 열립니다(Figure 32 참조).

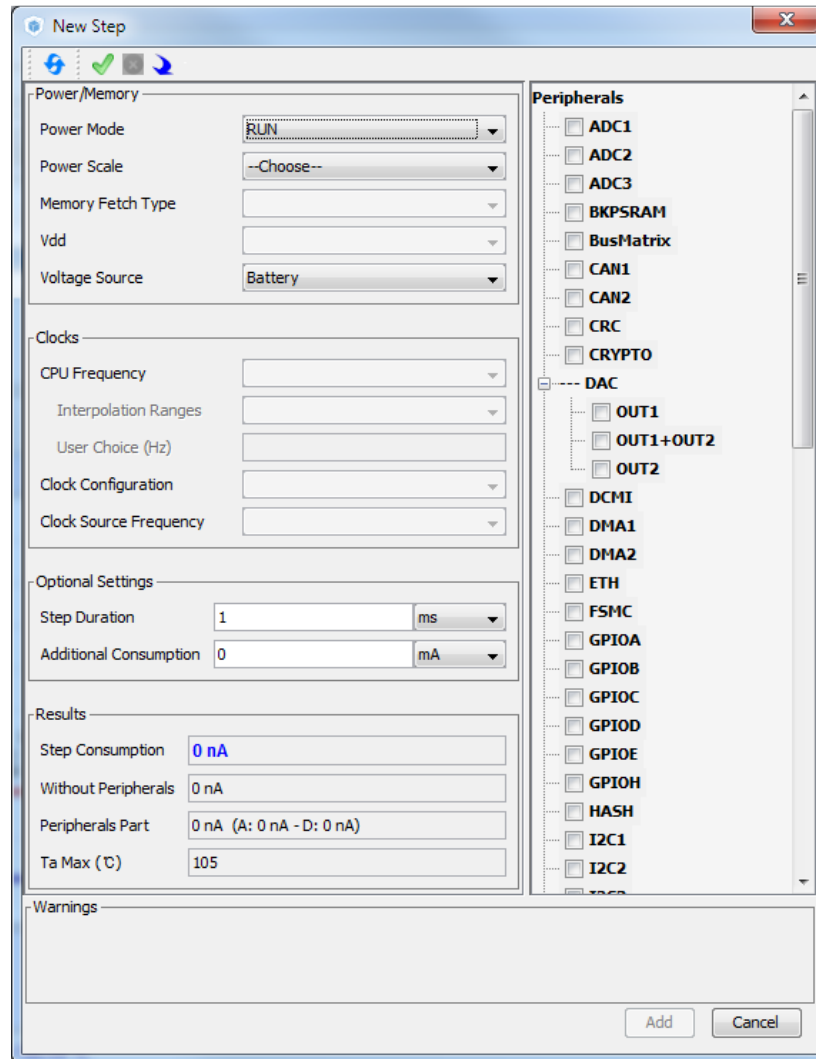


Figure 32 Power consumption sequence: new step default view

일단 step 이 설정되면, 소모전류와 T_{AMAX} 값의 결과는 power sequence 창에 나타납니다.

- Step 편집

Step 을 편집하기 위해 sequence 테이블을 더블클릭 합니다. 그러면 Edit Step 창이 열립니다(Figure 33 참조).

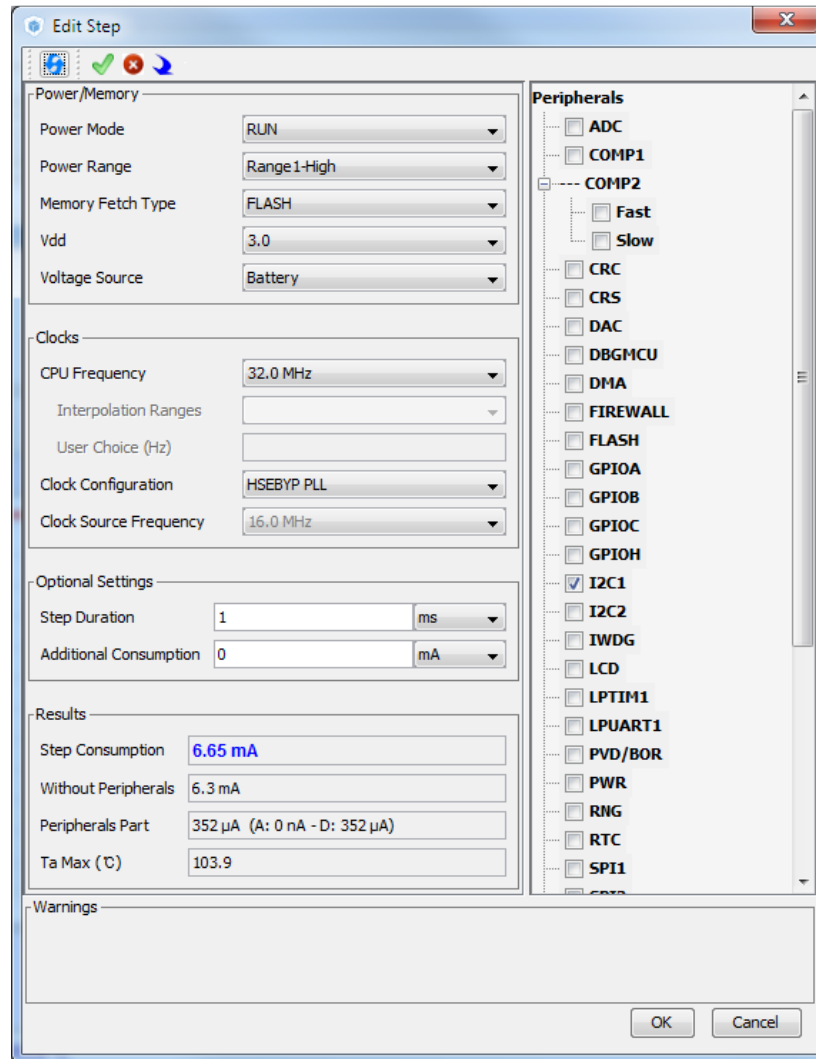


Figure 33 Edit Step window

- Step 이동

기본적으로 새로운 step 은 Sequence 의 마지막에 추가됩니다.

Sequence 테이블 상에 순서를 이동시키고자하는 step 을 클릭하고 Up/Down 버튼을 사용하여 이동합니다.

- Step 삭제

삭제하고자하는 step 을 선택하여 Delete 버튼을 클릭합니다.

8 STM32CubeMX Simple Example

STM32CubeMX 를 사용하여 Wake-up Key 를 구현합니다.

- Software tool : STM32CubeMX
- Hardware tool : STM32F401C-DISCO (Figure 34 참조)

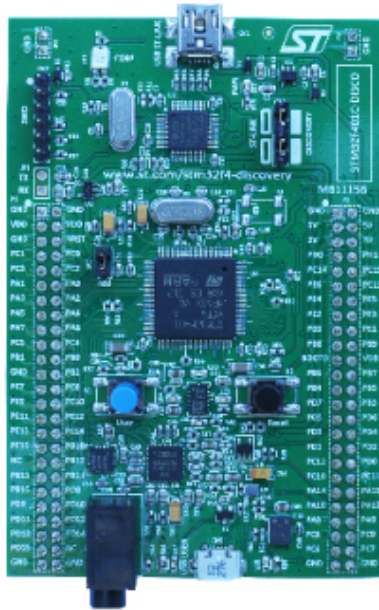


Figure 34 STM32F401C-DISCO board

우선 wake-up Key 를 구현하기 위해 STM32F401C-DISCO board 에 대한 정보를 User manual 에서 확인합니다.

User manual : http://www.st.com/resource/en/user_manual/dm00093902.pdf

1. MCU 선택

STM32CubeMX 를 실행하여 **New Project** 를 클릭 후 **MCU Select** 탭에서 STM32F401C-DISCO board 의 MCU 를 선택합니다(Figure 35 참조). STM32F401C-DISCO board 는 STM32F401VCTx 를 기반으로 되어있습니다.

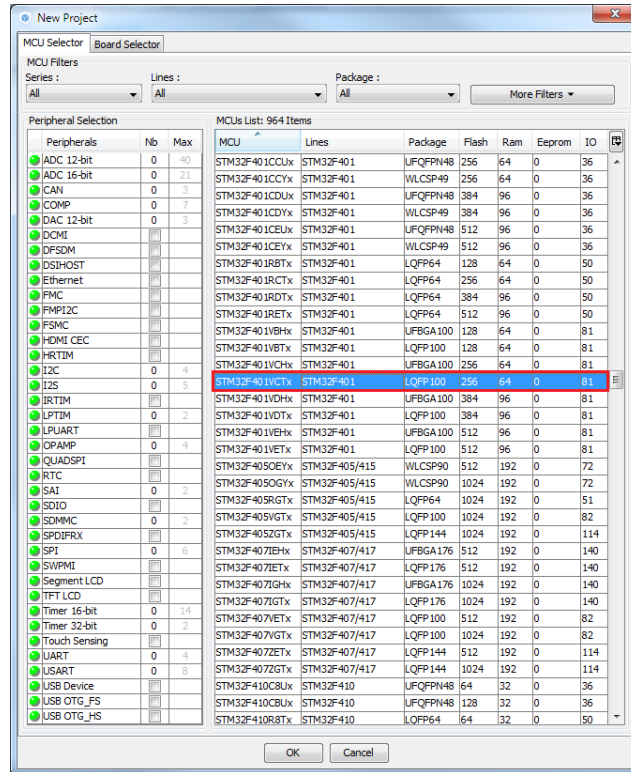


Figure 35 Select MCU on STM32F401C-DISCO

MCU 를 선택 후 아래 Figure 36 와 같이 **pinout view** 에 나타납니다.

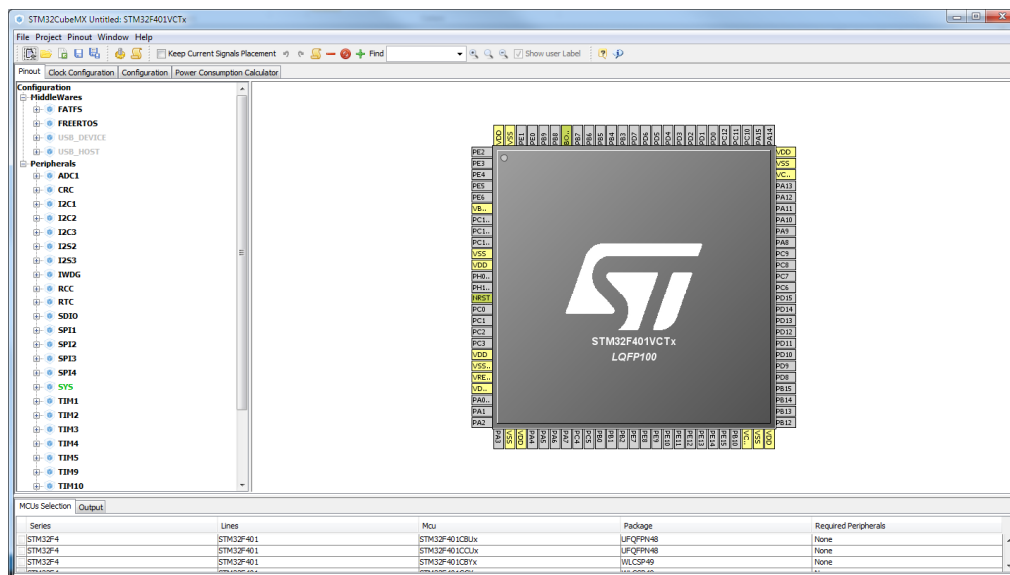
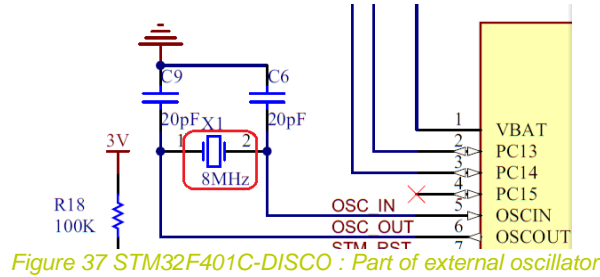


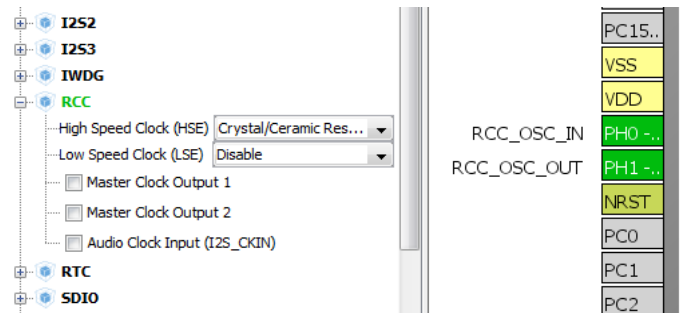
Figure 36 STM32F401VCTx Pinout view

2. RCC 설정

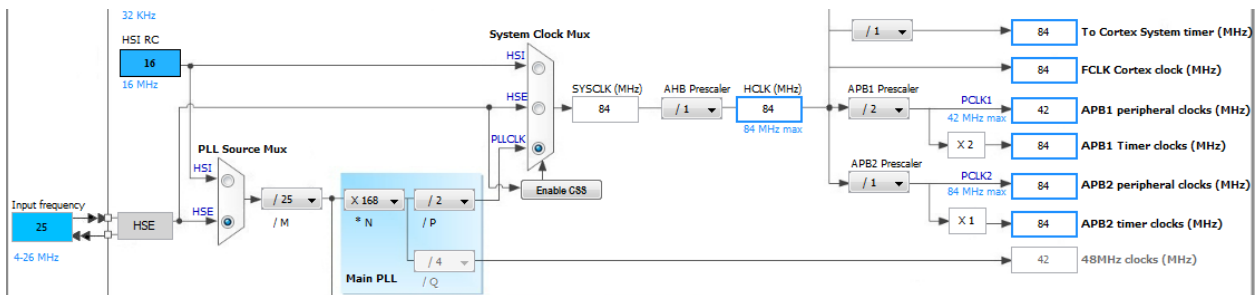
STM32F401C-DISCO 보드에는 아래 회로도(Figure 37 참조)에서 확인 할 수 있듯이 8MHz 외부 오실레이터가 HSE pin 에 연결되어 있습니다.



이 8MHz 외부 오실레이터를 사용하기 위해 좌측 pinout tree 상에 RCC 에서 HSE 를 Crystal/Ceramic Resonator 로 아래와 같이 설정합니다(Figure 38 참조).



다음 클럭을 설정하기 위해 **Clock Configuration** 탭으로 이동합니다. PLL Source Mux 상에서 활성화되어있는 HSE 를 선택하고 PLL 의 divider 와 multiplier 의 값(/M, *N, /P)을 조정한 후 System clock Mux 를 PLLCLK 로 선택하여 HCLK 가 84MHz 가 되도록 설정합니다(Figure 39 참조).



3. GPIO 설정

Pinout 탭을 클릭하여 다시 **Pinout view** 로 돌아갑니다. STM32F401C-DISCO 보드에서 wake-up 으로 사용할 수 있는 푸른색 버튼은 PA0 에 연결되어 있으므로 **GPIOA Pin0** 를 **external interrupt** 로 설정합니다(Figure 40 참조)

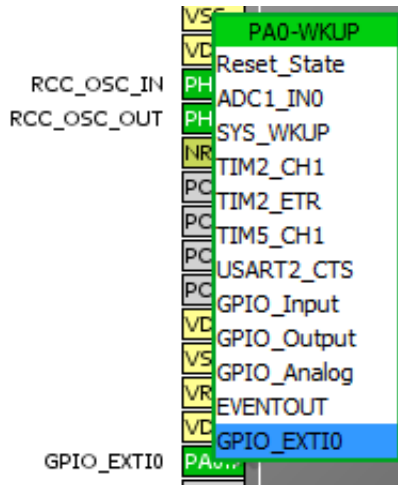


Figure 40 Set PA0-WKUP

Configuration 탭으로 이동하여 GPIO 설정을 확인합니다(Figure 41 참조).

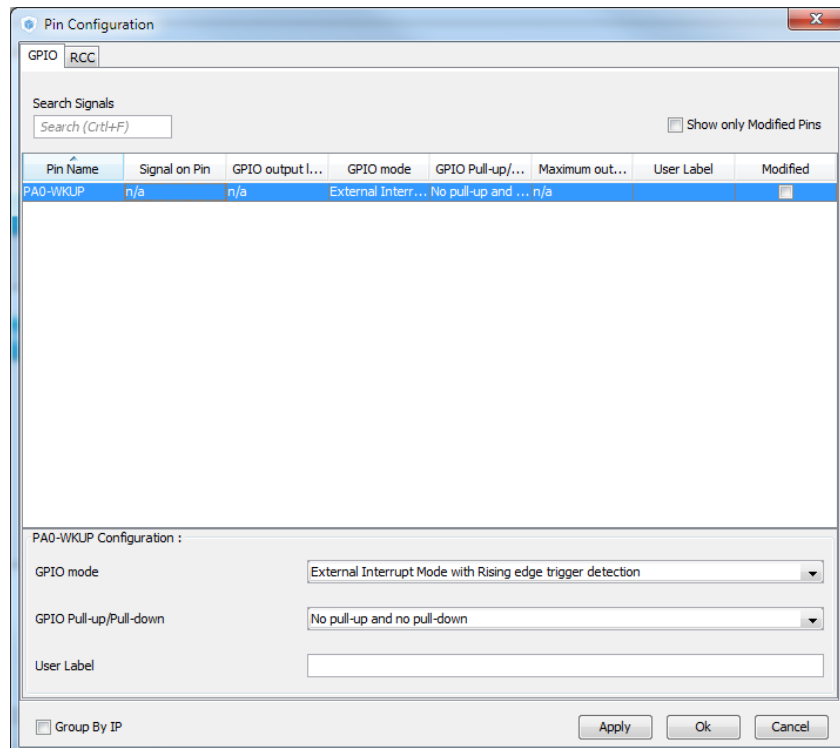


Figure 41 PA0-WKUP Configuration

다음 **NVIC Configuration** 의 interrupt 테이블에서 PA0-WKUP 과 매칭되는 interrupt(EXTI line0 interrupt)를 활성화합니다(참조 Figure 42).

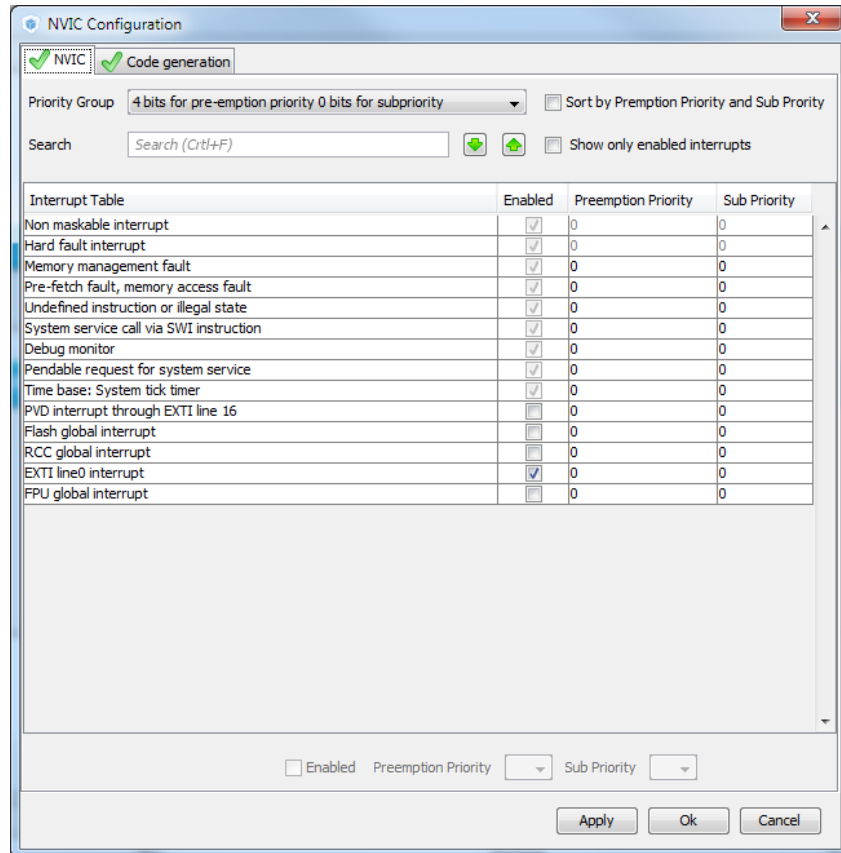


Figure 42 Enable EXTI line0 (PA0-WAUP)

4. 코드 생성

상단 menu bar 에서 [Project] -> [Generate Code]를 수행하여 설정된 MCU 의 초기화 C 코드를 생성합니다(Figure 43 참조).

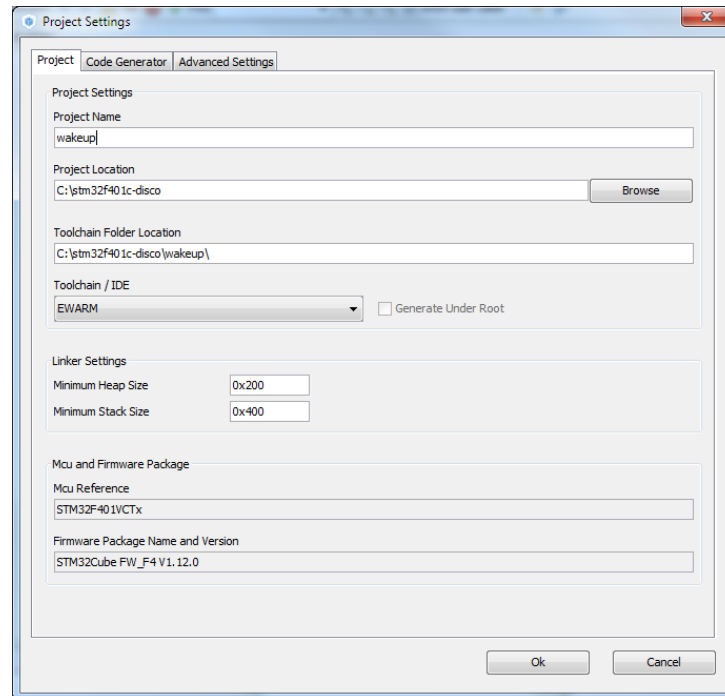


Figure 43 STM32F401C-DISCO code generation

코드 생성이 완료되면 Figure44, Figure45 과 같이 생성된 코드를 확인 할 수 있습니다.

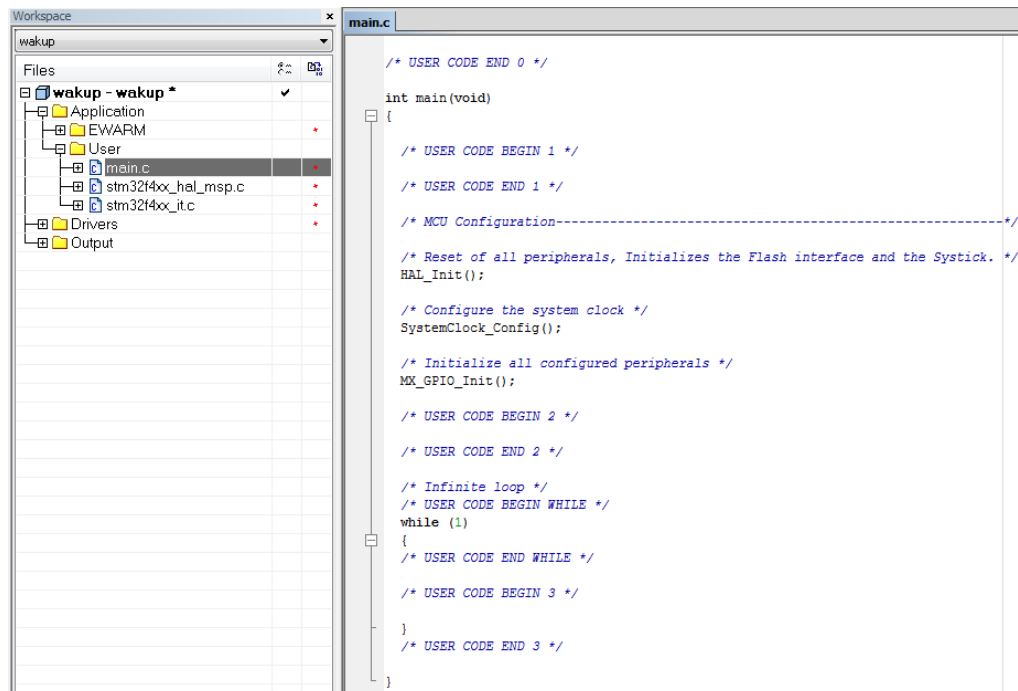


Figure 44 STM32F401C-DISCO initialization code with wake-up

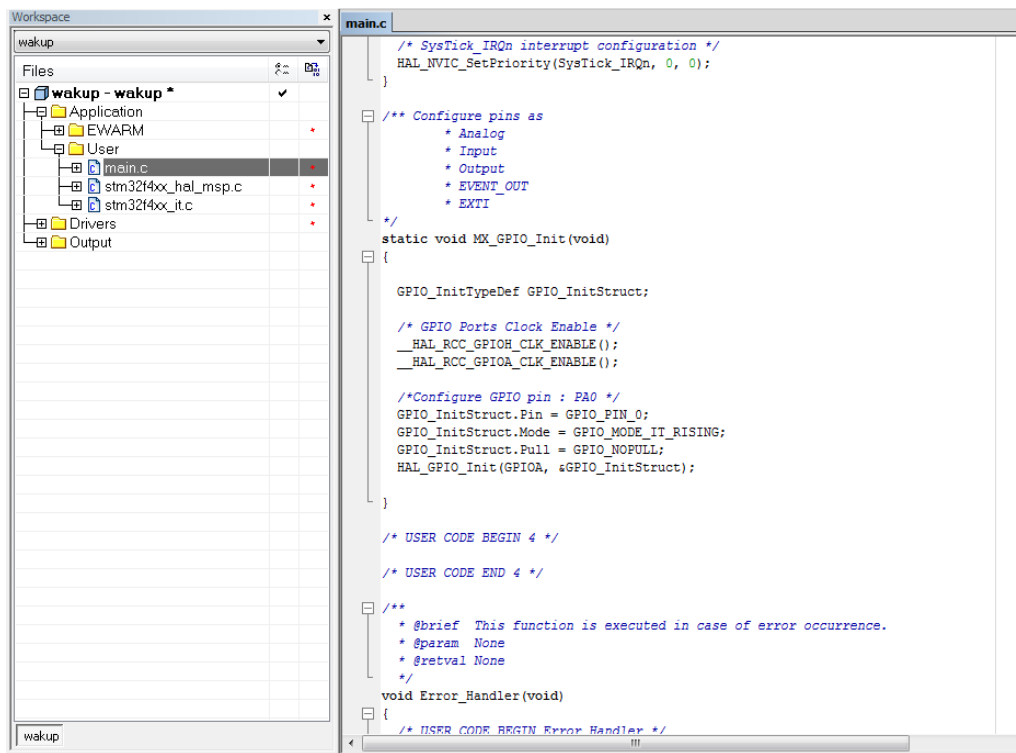


Figure 45 GPIO initialization code

5. 다운로드 및 동작 확인

생성된 코드를 빌드 후 STM32F401C-DISCO 보드에 생성된 바이너리를 다운로드합니다. 다운로드된 STM32F401C-DISCO 보드 상에서 푸른색 버튼을 누르면 PA0의 wake-up 인터럽트가 발생하여 stm32f4xx_it.c에 Figure 46와 같이 인터럽트 서비스 루틴(ISR)으로 진입하는 것을 확인할 수 있습니다.

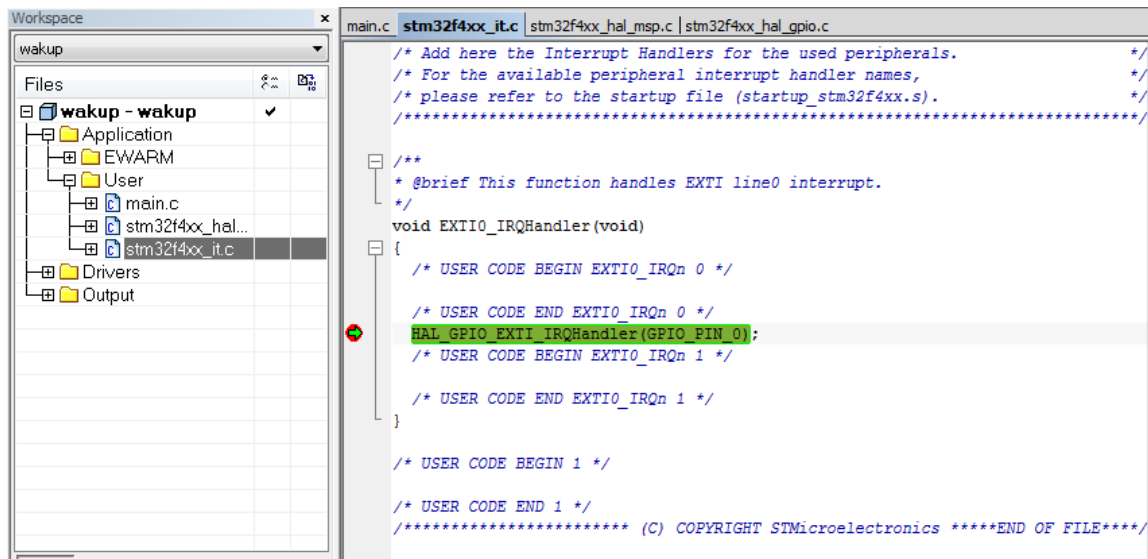


Figure 46 Wake-up(PA0) interrupt operation

2016 – STMicroelectronics all right reserved

저작권자의 사전 서면 승인이 없을 경우 전체 또는 일부 내용을 복제 및 가공할 수 없습니다. 본 문서를 비롯해 문서에 담긴 정보 및 관리 원칙은 ST의 계약 조건 이행 성립을 위해 작성된 내용이 아니며, 사전 고지 없이 변경될 수 있습니다.