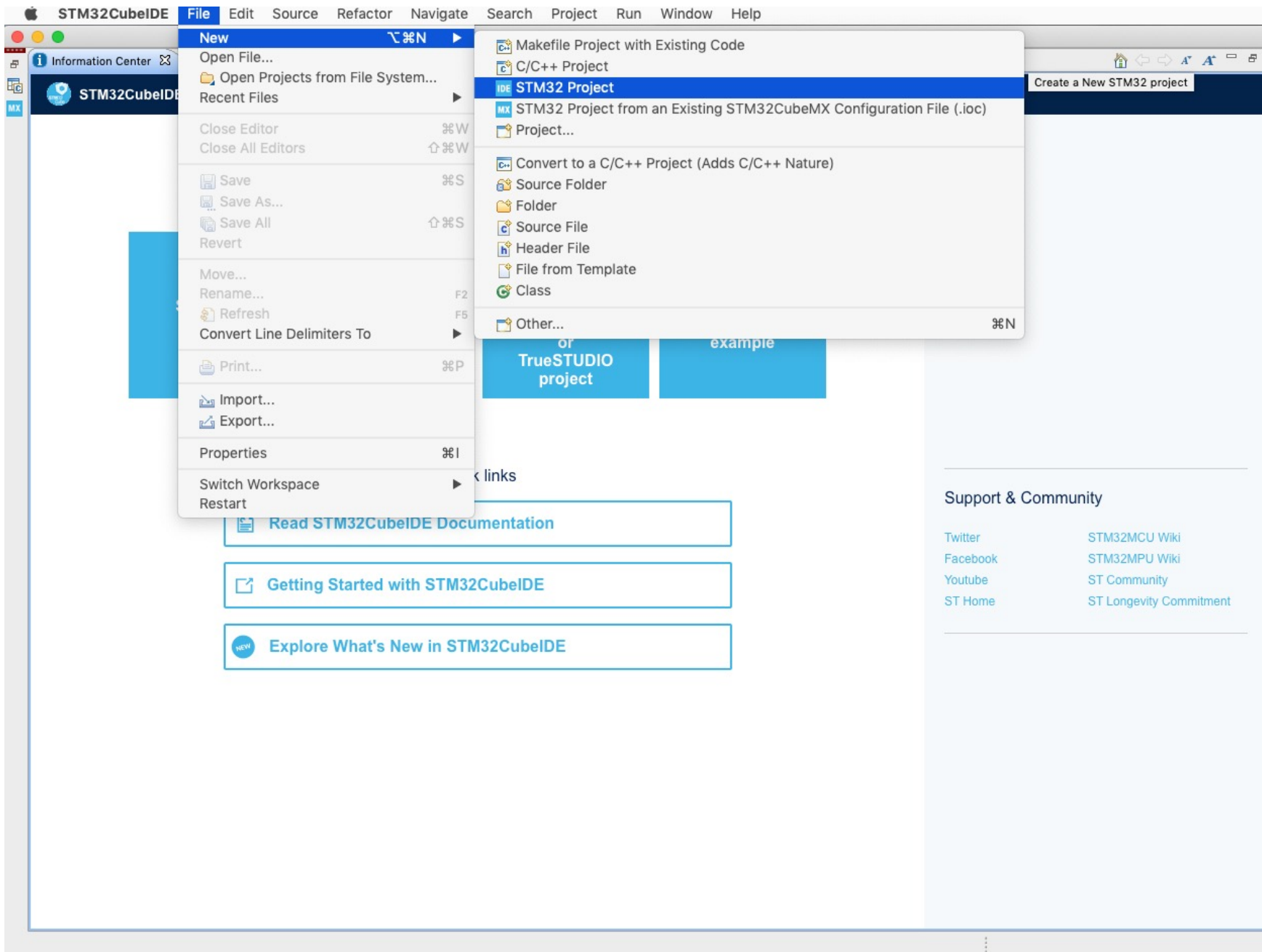




STM32CubeIDE 프로젝트 시작하기



- [File]→[New]→[STM32 Project]

* Note: STM32CubeIDE는 STM32CubeMX로 생성된 ioc 파일로 프로젝트를 생성할 수 있으며, 기존의 SW4STM32 및 TrueSTUDIO 컴파일러로 생성된 프로젝트들도 Import가 가능하다.

Target Selection

Select STM32 target or STM32Cube example

IDE

- MCU/MPU Selector 또는 Board Selector를 선택한다.

MCU/MPU Selector

Board Selector

Example Selector

Cross Selector

MCU/MPU Filters

★

📄

🔍

🔄

Part Number

Core

Series

Line

Check/Uncheck All

☐ STM32F0x0 Value line

☐ STM32F0x1

☐ STM32F0x2

☐ STM32F0x8

☐ STM32F100 Value line

☐ STM32F101

☐ STM32F102

☐ STM32F103

☐ STM32F105/107

☐ STM32F2x5

☐ STM32F2x7

☐ STM32F301

☐ STM32F302

☐ STM32F303

☐ STM32F334

☐ STM32F373

☐ STM32F3x8

☐ STM32F401

☐ STM32F405/415

☐ STM32F407/417

☐ STM32F410

☒ STM32F411

☐ STM32F412

★

STM32F4 Series

★

STM32F411RE

ACTIVE

Active

Product is in mass production

Unit Price for 10kU (US\$) : 2.902

Board: [NUCLEO-F411RE](#)

LQFP64

The STM32F411xC/xE devices are based on the high-performance Arm® Cortex®-M4 32-bit RISC core operating at a frequency of up to 100 MHz. The Cortex®-M4 core features a Floating point unit (FPU) single precision which supports all Arm single-precision data-processing instructions and data types. It also implements a full set of DSP instructions and a memory protection unit (MPU) which enhances application security.

The STM32F411xC/xE belongs to the STM32 Dynamic Efficiency product line (with products combining power efficiency, performance and integration) while adding a new innovative feature called Batch Acquisition Mode (BAM) allowing to save even more power consumption during data batching.

The STM32F411xC/xE incorporate high-speed embedded memories (up to 512 Kbytes of Flash memory, 128 Kbytes of SRAM), and an extensive range of enhanced I/Os and peripherals connected to two APB buses, two AHB bus and a 32-bit multi-AHB bus matrix.

All devices offer one 12-bit ADC, a low-power RTC, six general-purpose 16-bit timers including one PWM timer for motor control, two

MCUs/MPUs List: 9 items

Display similar items

Export

*	Part No	Reference	Marketing S...	Unit Price f...	Board	Package	Flash	RAM	IO	Freq.
☆	STM32F411...	STM32F411...	Active	2.285		UFQFP...	256 kBytes	128 kBytes	36	100 MHz
☆	STM32F411...	STM32F411...	Active	2.285		WLCSP49	256 kBytes	128 kBytes	36	100 MHz
☆	STM32F411...	STM32F411...	Active	2.795		UFQFP...	512 kBytes	128 kBytes	36	100 MHz
☆	STM32F411...	STM32F411...	Active	2.795		WLCSP49	512 kBytes	128 kBytes	36	100 MHz
☆	STM32F411...	STM32F411...	Active	2.391		LQFP64	256 kBytes	128 kBytes	50	100 MHz
★	STM32F411...	STM32F411...	Active	2.902	NUCLEO-F411RE	LQFP64	512 kBytes	128 kBytes	50	100 MHz
☆	STM32F411...	STM32F411...	Active	2.732		LQFP100	256 kBytes	128 kBytes	81	100 MHz
☆	STM32F411...	STM32F411...	Active	3.242		UFBGA...	512 kBytes	128 kBytes	81	100 MHz
☆	STM32F411...	STM32F411...	Active	3.242	STM32F411E-DISCO	LQFP100	512 kBytes	128 kBytes	81	100 MHz

?

< Back

Next >

Cancel

Finish

의공종합설계및실습1 (8269)

3

Target Selection

Select STM32 target or STM32Cube example

- Board type: Nucleo-64
- MCU/MPU Series : STM32F1

* 이, 경우 오직 한개의 보드가 나타나며, 이를 선택하고 다음 버튼을 누른다.

MCU/MPU Selector

Board Selector

Example Selector

Cross Selector

Part Number

Vendor

Type

Check/Uncheck All

☐ Discovery Kit
 ☐ Evaluation Board
 ☐ Evaluation Kit
 ☐ Nucleo USB Dongle
 ☐ Nucleo-144
 ☐ Nucleo-32
 ☒ Nucleo-64
 ☐ Nucleo-RF Kit

MCU/MPU Series

Check/Uncheck All

☐ STM32F0
 ☒ STM32F1
 ☐ STM32F2
 ☐ STM32F3
 ☐ STM32F4
 ☐ STM32F7
 ☐ STM32G0
 ☐ STM32G4
 ☐ STM32H7
 ☐ STM32L0
 ☐ STM32L1
 ☐ STM32L4
 ☐ STM32L4+
 ☐ STM32L5
 ☐ STM32MP1
 ☐ STM32WB

Features

Large Picture

Docs & Resources

Datasheet

Buy

STM32F1 Series

★

NUCLEO-F103RB

ACTIVE

Active

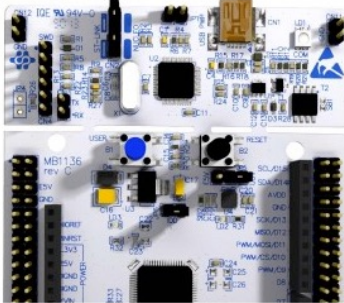
Product is in mass production

Part Number : NUCLEO-F103RB

Commercial Part Number : NUCLEO-F103RB

Unit Price (US\$) : 10.32

Mounted Device : STM32F103RBTx



The STM32 Nucleo-64 board provides an affordable and flexible way for users to try out new concepts and build prototypes by choosing from the various combinations of performance and power consumption features, provided by the STM32 microcontroller. For the compatible boards, the external SMPS significantly reduces power consumption in Run mode.

The ARDUINO® Uno V3 connectivity support and the ST morpho headers allow the easy expansion of the functionality of the STM32 Nucleo open development platform with a wide choice of specialized shields.


The STM32 Nucleo-64 board does not require any separate probe as it integrates the ST-LINK debugger/programmer.

The STM32 Nucleo-64 board comes with the STM32 comprehensive free software libraries and examples available with the STM32Cube MCU Package.

Features

Boards List: 1 item

Export

*	Overview	Commercial Part ...	Type	Marketing Status	Unit Price (US\$)	Mounted Device
★		NUCLEO-F103RB	Nucleo-64	Active	10.32	STM32F103RBTx

< Back

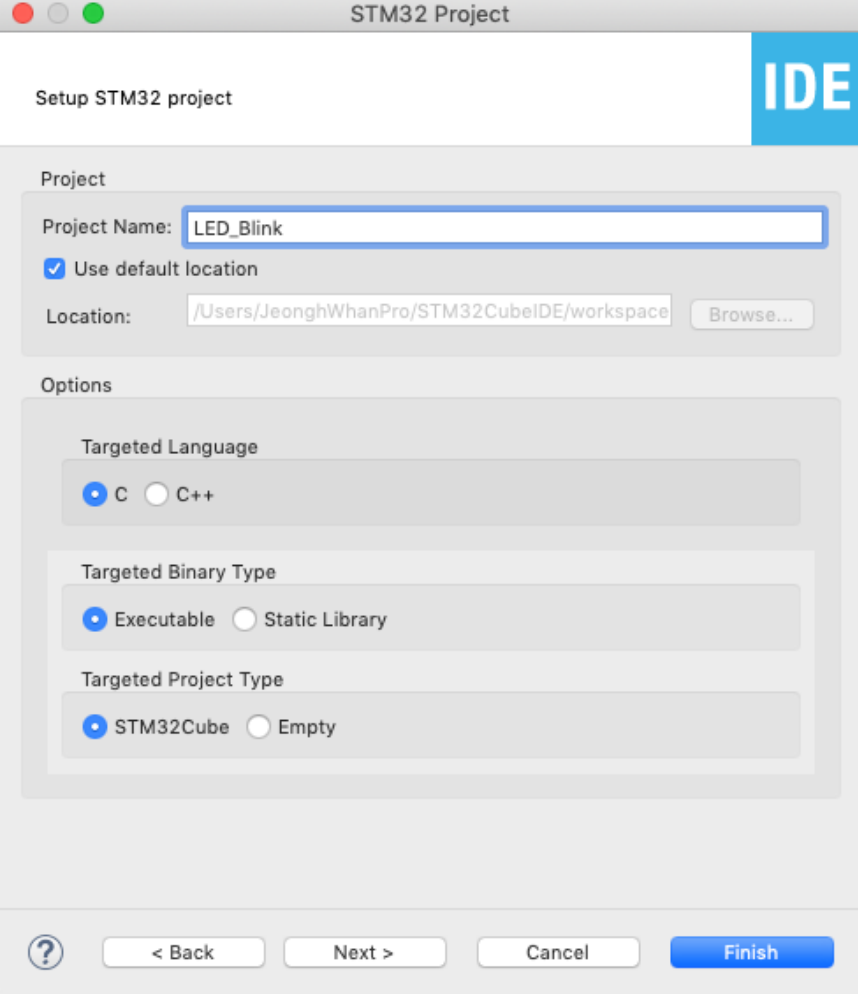
Next >

Cancel

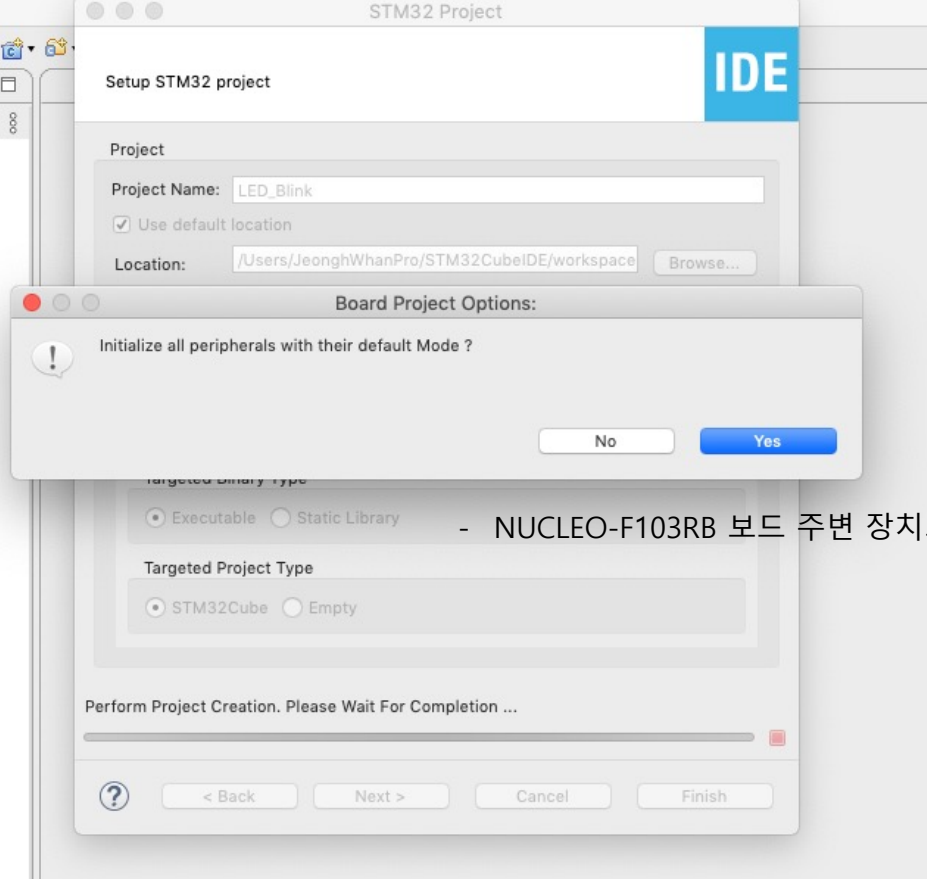
Finish

의공종합설계및실습1 (8269)

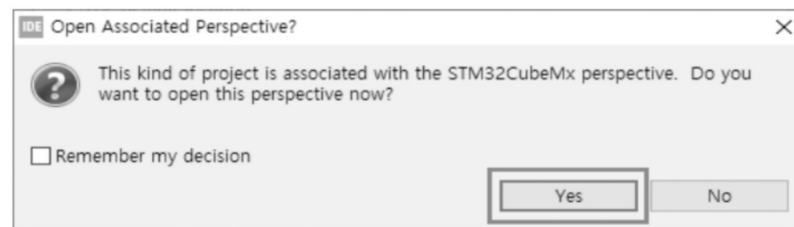
4



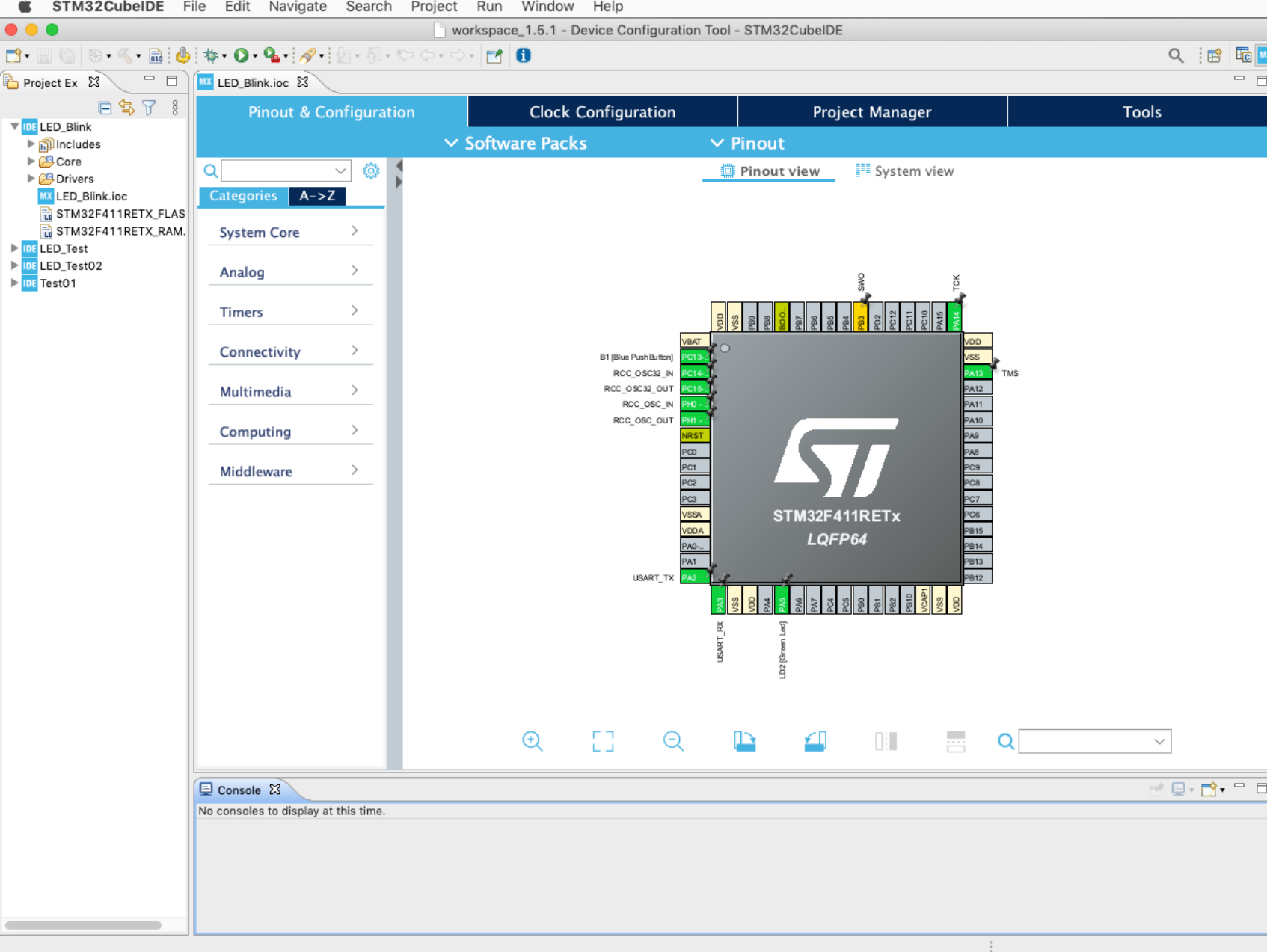
- "LED_Blink"로 새 프로젝트 이름을 설정한다.
- 기본 저장 위치는 아래 Location에 나와 있으면 사용자가 원하는 위치에 변경 가능하다.



- NUCLEO-F103RB 보드 주변 장치의 기본 설정을 불러 온다.



- Eclipse기반 IDE환경에 CubeMX layout view를 추가하기 위한 'Yes'를 선택한다.



- Reference board에서 default모드로 설정되어 있는 핀들이 표시된다.

- 기본적으로 설정되어 있는 Pin들은 다음과 같다.

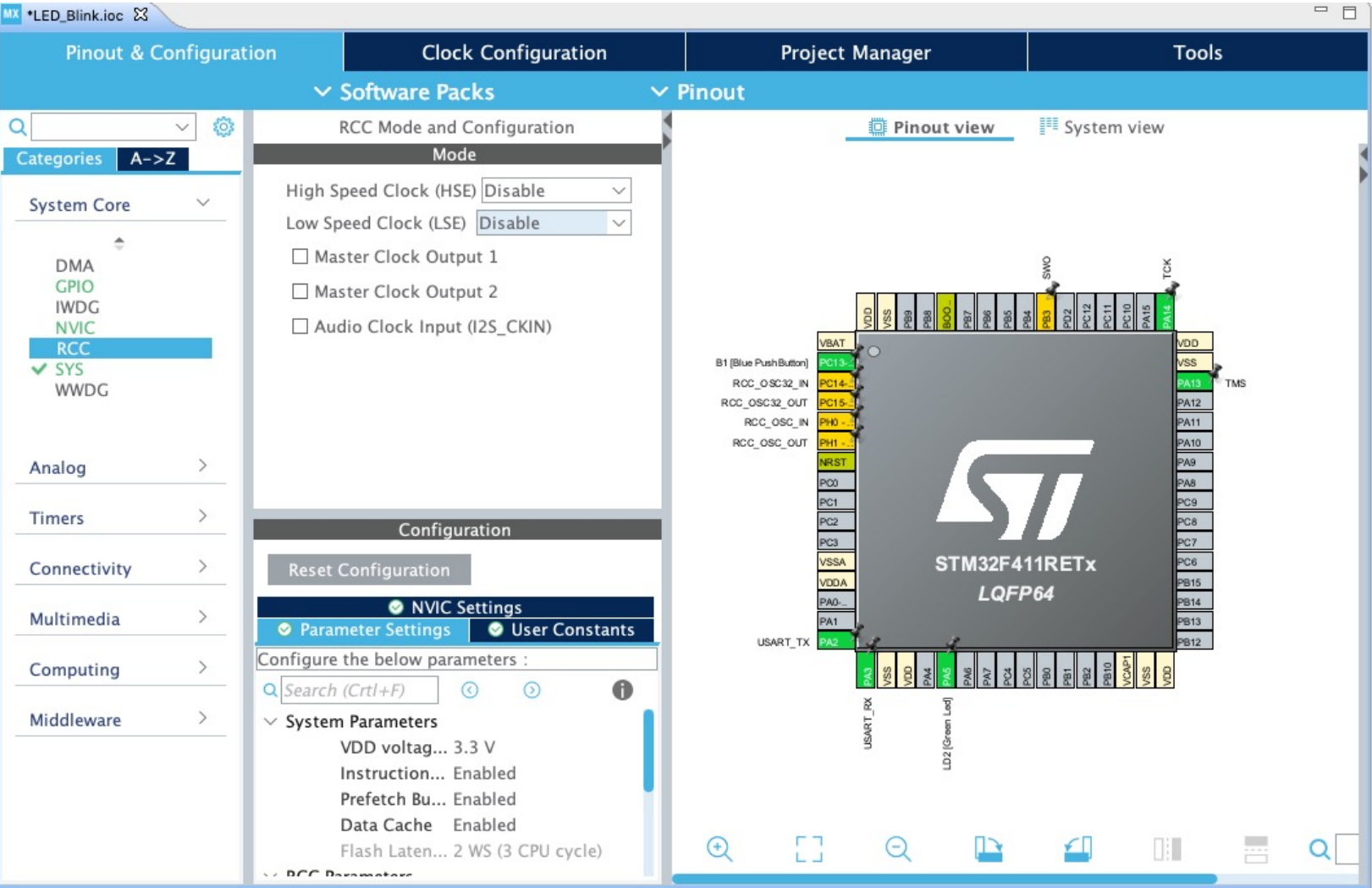
- * Serial Wire Debug 3핀(SWO, TCK, TMS)
- * UART2 2핀 (USART_TX, USART_RX)
- * Push Button (B1)
- * LED (LD2)

- HSE(RCC_OSC_IN / RCC_OSC_OUT) 과 LSE (RCC_OSC32_IN / RCC_OSC32_OUT) 부분은 뒤에 사용할때까지 비활성화(disable)로 설정한다.

* 참고: NUCLEO보드에는 HSE클럭용 크리스탈(X3)는 마운트되어 있지 않다.

- 이번 예제에서는 LD2 LED를 깜박이도록 (blinking) 하는 예제를 작성해 볼 예정이다.

Pinout & Configuration – RCC(Reset Clock Controller) 설정



- RCC의 설정
System Core에서 RCC를 선택하고, HSE와 LSE를 disable로 설정한다.

- RCC(Reset Clock Controller)
- HSE(High Speed External) : 외부에서 입력되는 높은 주파수의 clock으로 PLL을 거쳐 system clock으로 입력됨.
 - HSI(High Speed Internal) : 8/16MHz. 내장된 고정밀의 RC발진 회로에 의한 clock. 외부 crystal이 없어도 동작 가능한 clock.
 - SE(Low Speed External):32.768kHz. RTC에 사용되는 Clock용으로 주로 사용
 - LSI(Low Speed Internal): 내부 RC발진. Low Power Mode지원, Watchdog이나 AWU(Auto Wakeup)의 clock로 사용

Pinout & Configuration – GPIO(General Purpose Input/Output) 설정

MX *LED_Blink.ioc

Pinout & Configuration

Clock Configuration

Project Manager

Tools

Software Packs

Pinout

Search

Categories

A-Z

System Core

DMA

GPIO

IWDG

NVIC

RCC

SYS

WWDG

Analog

Timers

Connectivity

Multimedia

Computing

Middleware

GPIO Mode and Configuration

Configuration

Group By Peripherals

GPIO

Single Mapped Signals

SYS

USART

NVIC

Search Signals

Search (Ctrl+F)

Show only Modified Pins

Pin ...	Signal ...	GPIO o...	GPIO m...	GPIO Pu...	Maxim...	User La...	Modified
PA5	n/a	Low	Output ...	No pull...	Low	LD2 [Gr...	✓
PC13-...	n/a	n/a	Externa...	No pull...	n/a	B1 [Blu...	✓

GPIO output level

Low

GPIO mode

Output Push Pull

GPIO Pull-up/Pull-down

No pull-up and no pull-down

Maximum output speed

Low

User Label

LD2 [Green Led]

Pinout view

System view

VBAT

VDD

VSS

PB9

PB8

BOOT

PB7

PB6

PB5

PB4

PB3

SWO

PD2

PC12

PC11

PC10

PA15

PA14

TCK

VDD

VSS

TMS

PA13

PA12

PA11

PA10

PA9

PA8

PC9

PC8

PC7

PC6

PB15

PB14

PB13

PB12

VDD

VSS

VCAP1

PB10

PB1

PB2

PC5

PA6

PA7

PA4

VDD

VSS

PA3

USART_RX

LD2 [Green Led]

PA2

PA1

PA0...

VDDA

VSSA

PC3

PC2

PC1

PC0

NRST

PH0

RCC_OSC_IN

PC15

PC14

PC13

B1 [Blue PushButton]

USART_TX

LD2 [Green Led]

PA2

PA1

PA0...

VDDA

VSSA

PC3

PC2

PC1

PC0

NRST

PH0

RCC_OSC_IN

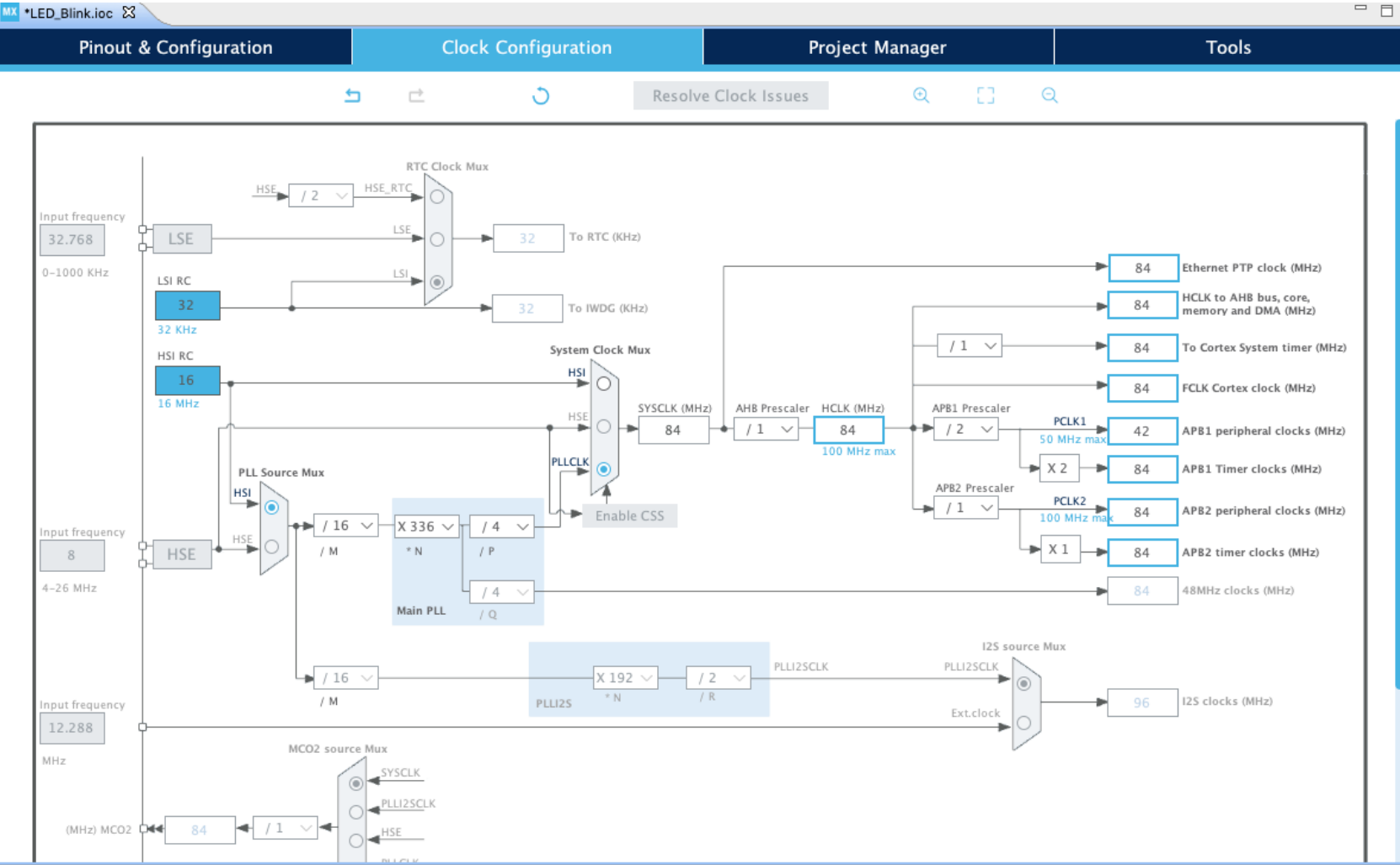
PC15

PC14

PC13

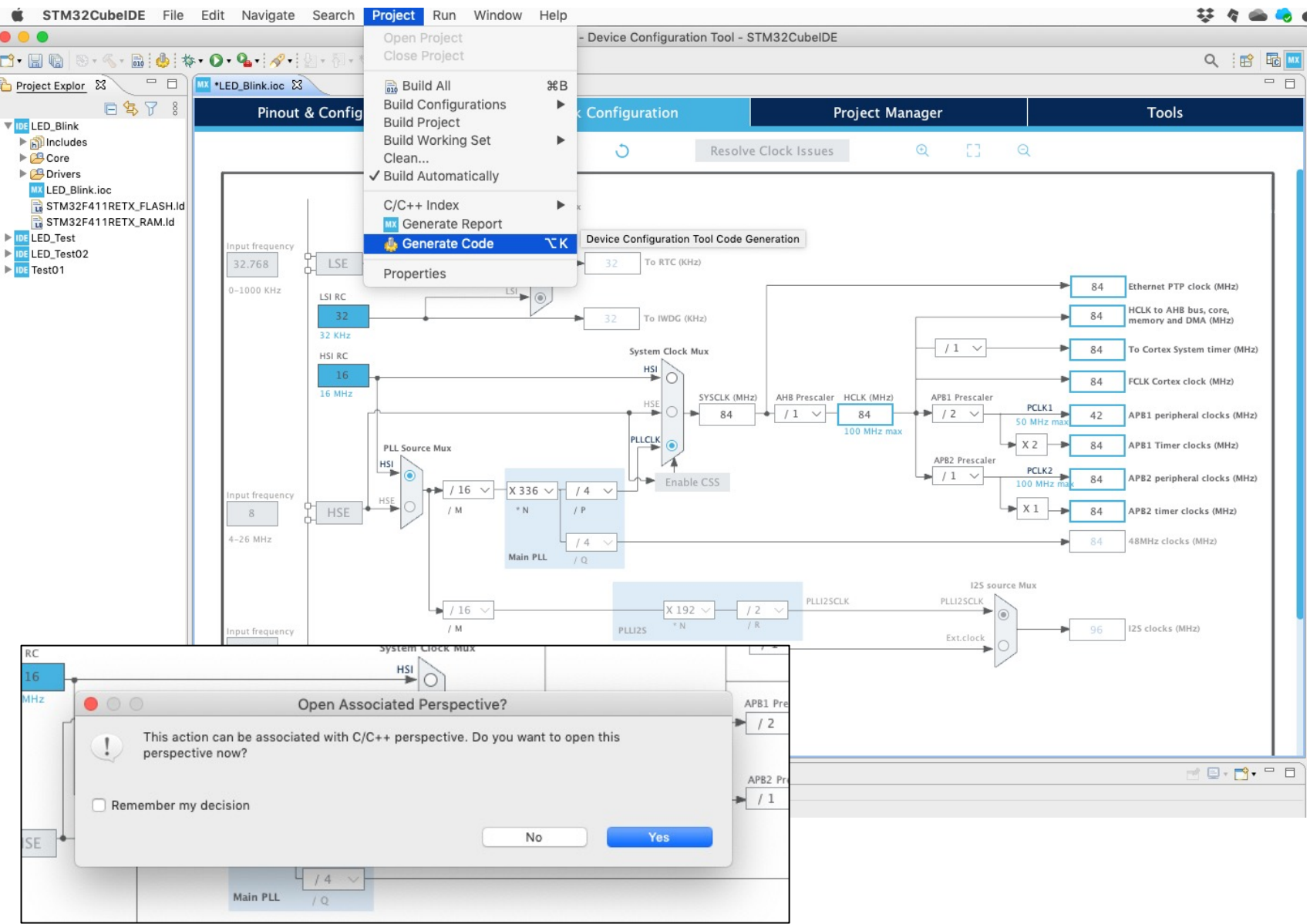
B1 [Blue PushButton]

Pinout & Configuration – Clock Configuration



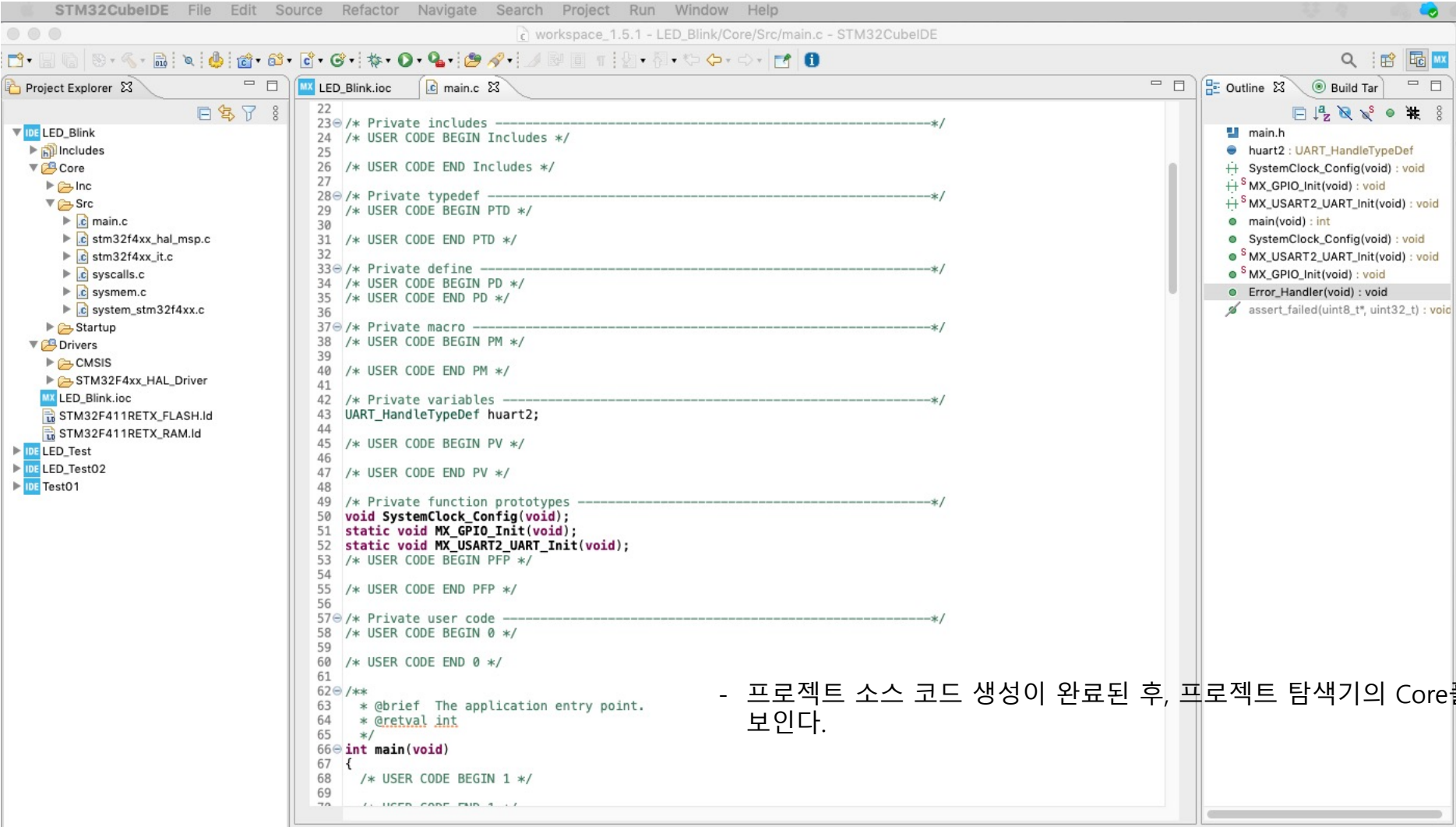
- System Clock에 대하여 설정한다.
- System Clock Mux에서 PLLCLK를 선택하여, HCLK가 84MHz가 되도록 설정한다.
(STM32F103의 경우, 64MHz / max. 72MHz)

Code Generation - 01



- 설정한 상태에 따라 코드를 생성하기 위해서 [Project] → [Generate Code] (Alt+K) 메뉴를 실행한다.

Code Generation - 02



- 프로젝트 소스 코드 생성이 완료된 후, 프로젝트 탐색기의 Core폴더안에 Src폴더에서 main.c가 보인다.

프로젝트의 빌드

STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

LED_Blink

Includes

Core

Inc

Src

main.c

stm32f4xx_hal_msp.c

stm32f4xx_it.c

syscalls.c

systemem.c

system_stm32f4xx.c

Startup

Drivers

CMSIS

STM32F4xx_HAL_Driver

LED_Blink.ioc

STM32F411RETX_FLASH.ld

STM32F411RETX_RAM.ld

LED_Test

LED_Test02

Test01

LED_Blink.ioc

main.c

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

Project

Open Project

Close Project

Build All

Build Configurations

Build Project

Build Working Set

Clean...

Build Automatically

Build Targets

C/C++ Index

Generate Report

Generate Code

Properties

main.c - STM32CubeIDE

Build Project

Problems Tasks Console Properties

CDT Build Console [LED_Blink]

arm-none-eabi-objdump -h -S LED_Blink.elf > "LED_Blink.list"

arm-none-eabi-objcopy -O binary LED_Blink.elf "LED_Blink.bin"

text data bss dec hex filename

7224 20 1636 8880 22b0 LED_Blink.elf

Finished building: default.size.stdout

Finished building: LED_Blink.bin

Finished building: LED_Blink.list

20:00:15 Build Finished. 0 errors, 0 warnings. (took 1s.91ms)

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

/* Private includes -----*/

/* USER CODE BEGIN Includes */

/* USER CODE END Includes */

/* Private typedef -----*/

/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define -----*/

/* USER CODE BEGIN PD */

/* USER CODE END PD */

/* Private macro -----*/

/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables -----*/

UART_HandleTypeDef huart2;

/* USER CODE BEGIN PV */

/* USER CODE END PV */

/* Private function prototypes -----*/

void SystemClock_Config(void);

static void MX_GPIO_Init(void);

static void MX_USART2_UART_Init(void);

/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code -----*/

/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/**

* @brief The application entry point.

* @retval int

*/

int main(void)

{

/* USER CODE BEGIN 1 */

- 기본적으로 생성된 프로젝트가 정상적으로 빌드 되는지 확인.

- 정상적으로 프로젝트가 진행되면 다음과 같은 화면이 나타난다.

소스 코드의 작성-01

- 자, 이제 앞에서 설정한 정보를 바탕으로 NUCLEO보드의 LED가 깜빡이는 코드를 작성.
- 이를 위하여, STM32CubeIDE에서 제공하고 있는 HAL_Driver에서 제공하는 함수를 사용함.

Project Explorer

LED_Blink

Binaries

Includes

Core

Inc

Src

Startup

Drivers

CMSIS

STM32F4xx_HAL_Driver

Inc

Src

stm32f4xx_hal_cortex.c

stm32f4xx_hal_dma_ex.c

stm32f4xx_hal_dma.c

stm32f4xx_hal_exti.c

stm32f4xx_hal_flash_ex.c

stm32f4xx_hal_flash_ramfunc.c

stm32f4xx_hal_flash.c

stm32f4xx_hal_gpio.c

stm32f4xx_hal.h

GPIO_NUMBER

HAL_GPIO_DeInit(GPIO_TypeDef*, uint32_t) : void

HAL_GPIO_EXTI_Callback(uint16_t) : void

HAL_GPIO_EXTI_IRQHandler(uint16_t) : void

HAL_GPIO_Init(GPIO_TypeDef*, GPIO_InitTypeDef*) : void

HAL_GPIO_LockPin(GPIO_TypeDef*, uint16_t) : HAL_StatusTypeDef

HAL_GPIO_ReadPin(GPIO_TypeDef*, uint16_t) : GPIO_PinState

HAL_GPIO_TogglePin(GPIO_TypeDef*, uint16_t) : void

HAL_GPIO_WritePin(GPIO_TypeDef*, uint16_t, GPIO_PinState) : void

stm32f4xx_hal_pwr_ex.c

stm32f4xx_hal_pwr_ex.c

stm32f4xx_hal_pwr.c

stm32f4xx_hal_rcc_ex.c

stm32f4xx_hal_rcc.c

stm32f4xx_hal_tim_ex.c

stm32f4xx_hal_tim.c

stm32f4xx_hal_uart.c

stm32f4xx_hal.c

stm32f4xx_hal.h

__STM32F4xx_HAL_VERSION

__STM32F4xx_HAL_VERSION_MAIN

__STM32F4xx_HAL_VERSION_RC

__STM32F4xx_HAL_VERSION_SUB1

__STM32F4xx_HAL_VERSION_SUB2

BSCKSEL_BIT_NUMBER

CMP_PD_BIT_NUMBER

CMPCR_CMP_PD_BB

CMPCR_OFFSET

IDCODE_DEVID_MASK

MCHDLYCR_BSCKSEL_BB

MCHDLYCR_OFFSET

MEMRMP_OFFSET

SYSCFG_OFFSET

UFB_MODE_BB

UFB_MODE_BIT_NUMBER

uwTick : volatile uint32_t

uwTickFreq : HAL_TickFreqTypeDef

uwTickPrio : uint32_t

HAL_DBGMCU_DisableDBGSleepMode(void) : void

HAL_DBGMCU_DisableDBGStandbyMode(void) : void

HAL_DBGMCU_DisableDBGStopMode(void) : void

HAL_DBGMCU_EnableDBGSleepMode(void) : void

HAL_DBGMCU_EnableDBGStandbyMode(void) : void

HAL_DBGMCU_EnableDBGStopMode(void) : void

HAL_DeInit(void) : HAL_StatusTypeDef

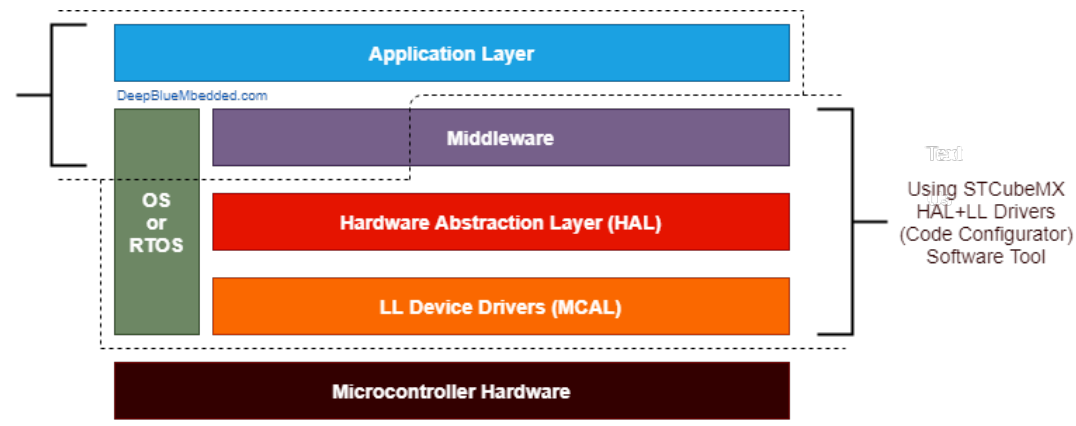
HAL_Delay(uint32_t) : void

HAL_DisableCompensationCell(void) : void

HAL_DisableMemorySwappingBank(void) : void

HAL_EnableCompensationCell(void) : void

HAL_EnableMemorySwappingBank(void) : void



Hardware Abstraction Layer (HAL) / low-layer APIs (LL)

`void HAL_GPIO_TogglePin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)`

`_weak void HAL_Delay(uint32_t Delay)`

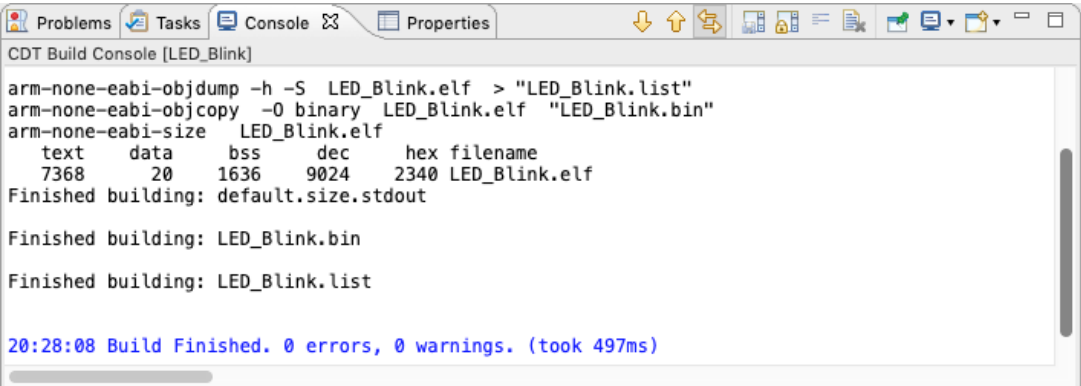
소스 코드의 작성-02

```
97 while (1)
98 {
99     /* USER CODE END WHILE */
100     HAL_GPIO_TogglePin(LD2_GPIO_Port,LD2_Pin);
101     HAL_Delay(500); // ms
102     /* USER CODE BEGIN 3 */
103 }
```

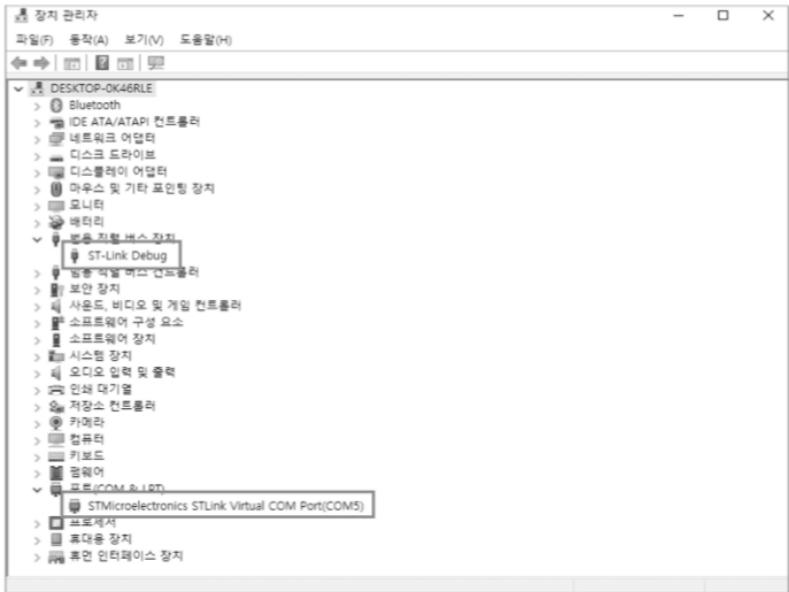
- main.c 소스 안에 위와 같이 추가하여 입력한다.

```
60 /* Private defines -----
61 #define B1_Pin GPIO_PIN_13
62 #define B1_GPIO_Port GPIOC
63 #define USART_TX_Pin GPIO_PIN_2
64 #define USART_TX_GPIO_Port GPIOA
65 #define USART_RX_Pin GPIO_PIN_3
66 #define USART_RX_GPIO_Port GPIOA
67 #define LD2_Pin GPIO_PIN_5
68 #define LD2_GPIO_Port GPIOA
69 #define TMS_Pin GPIO_PIN_13
70 #define TMS_GPIO_Port GPIOA
71 #define TCK_Pin GPIO_PIN_14
72 #define TCK_GPIO_Port GPIOA
73 #define SW0_Pin GPIO_PIN_3
74 #define SW0_GPIO_Port GPIOB
```

- LED 포트와 핀번호는 main.h안에서 다시 한번 확인할 수 있다.



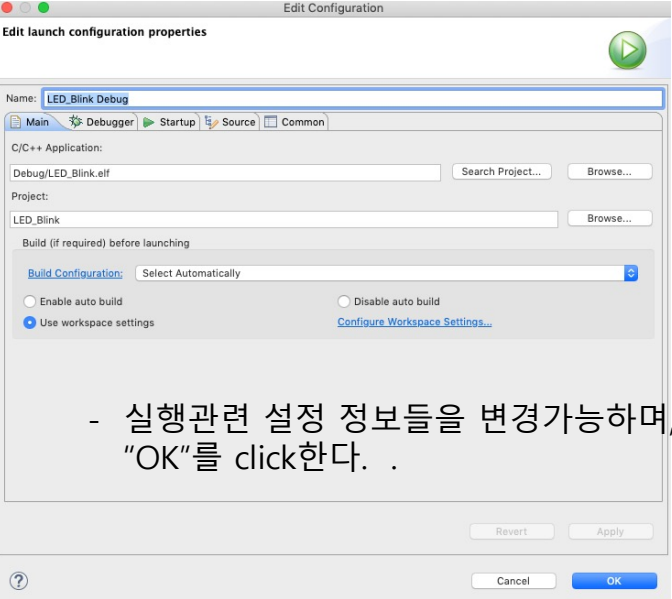
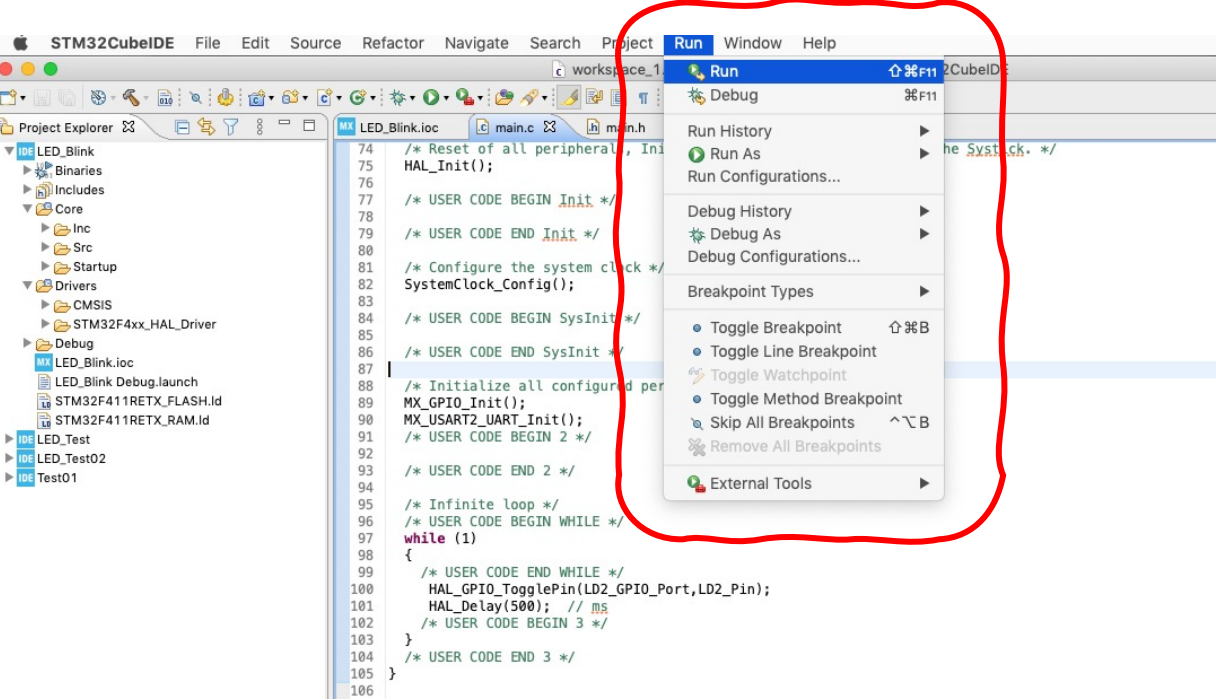
- 정상적으로 빌드되면, 위와 같은 메시지가 나온다. .



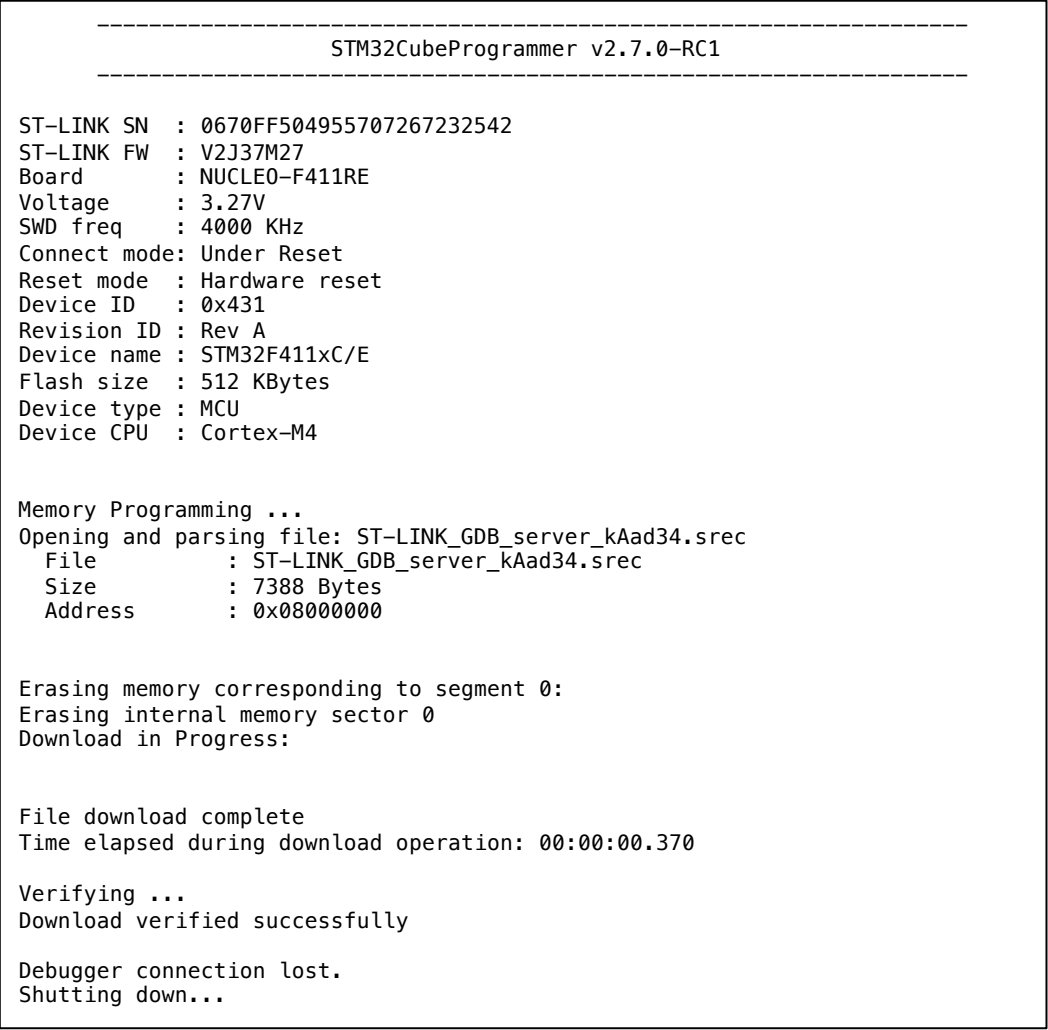
- NUCLEO보드에 다운하여 실행해 보기 전, 정상적으로 보드가 연결되어 있는 장치관리자를 통하여 확인한다.

참고] 사용자가 코드를 입력할 때는 반드시 CubeMX에서 생성한 “USER CODE BEGIN”과 “USER CODE END” 주석문 사이에 입력해야 한다. 그래야, Generate Code를 다시 수행하여도 사용자가 입력한 코드가 사라지지 않는다.

소스 코드의 실행



- 실행관련 설정 정보들을 변경가능하며, 기본값으로 실행하려면 "OK"를 click한다. .



- IDE는 변경된 화일을 자동으로 저장하며, 빌드가 실행된다.
- 빌드가 정상적으로 완료되면, target board에 자동으로 다운로드까지 한번에 처리되는 것을 확인할 수 있다.
- 보드의 Green LED가 깜빡거리는 것을 확인할 수 있다.