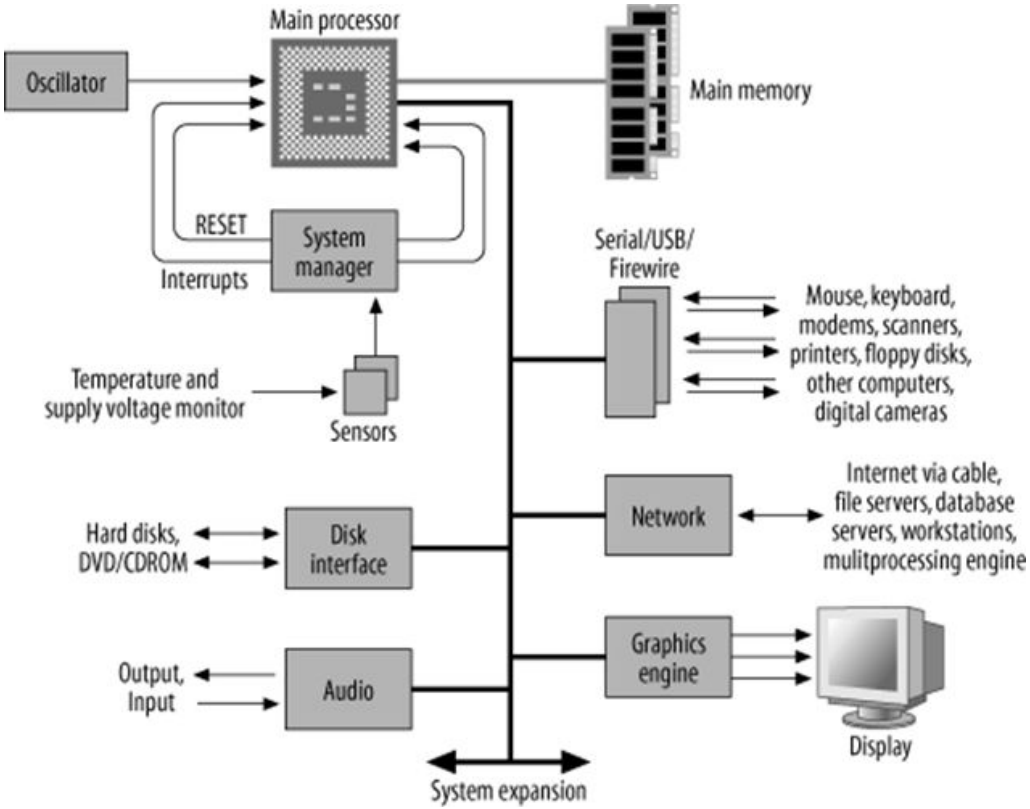
STM32를 활용한 FreeRTOS입문

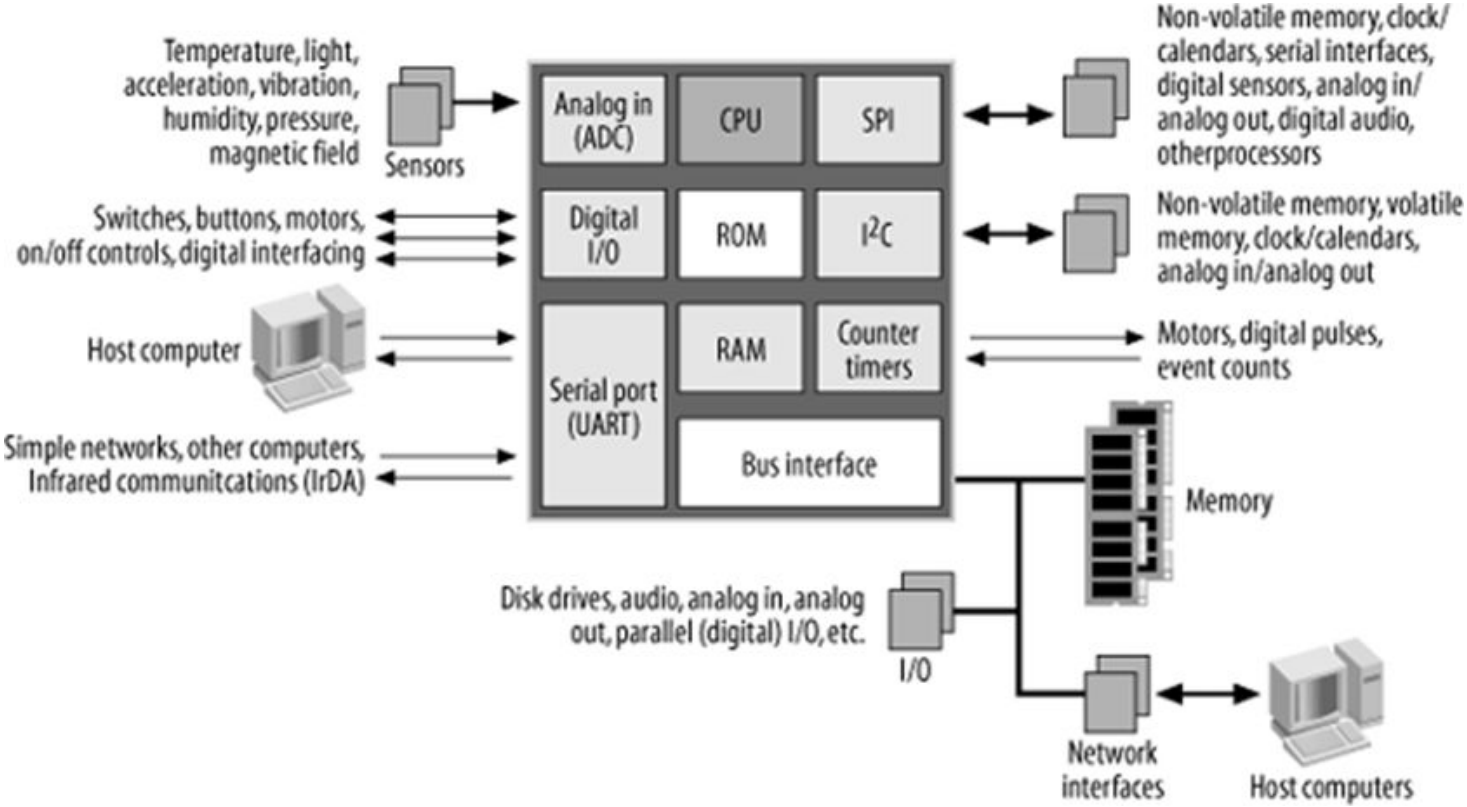
회사소개

http://www.codezoo.co.kr/

PC 하드웨어


Main processor
Oscillator
Main memory
RESET
System manager
Interrupts
Serial/USB/ Firewire
Mouse, keyboard, modems, scanners, printers, floppy disks, other computers, digital cameras
Temperature and supply voltage monitor
Sensors
Internet via cable, file servers, database servers, workstations, mulitprocessing engine
Network
Hard disks, DVD/CDROM
Disk interface
Graphics engine
Output, Input
Audio
Display
System expansion

임베디드 시스템 하드웨어



ARM

ARM Holdings

영국의 반도체 회사

- 프로세서를 설계하고 라이센싱
- 소프트웨어 개발 도구 설계 및 판매

Joint venture between Acorn Computers and Apple

1990

Designed into first mobile phones and then smartphones

1993 onwards

Now all electronic devices can use smart Arm technology

Today

ARM Core

Cortex-A

Highest performance

Optimized for high-level operating systems

Cortex-R

Fast response

Optimized for high performance, hard real-time applications

Cortex-M

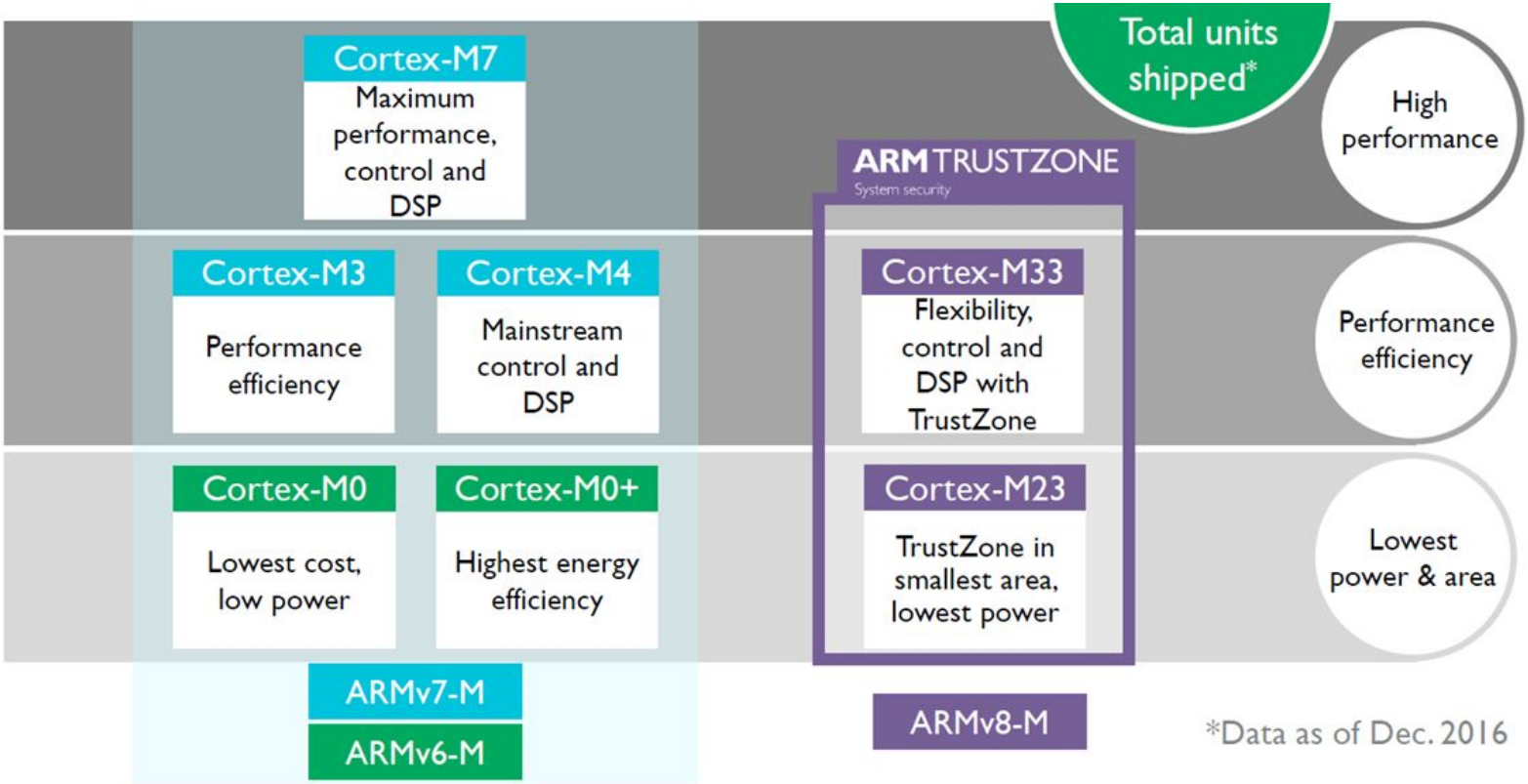
Smallest/lowest power

Optimized for discrete processing and microcontrollers

ARM Core

	Cortex-A	Cortex-R	Cortex-M
설계	고주파 클럭, 긴 파이프 라인, 고성능, 멀티미디어 지원 (NEON 명령어 세트 확장)	고주파 클럭, 중간에서 긴 길이의 파이프 라인, 확정적 (저지연 인터럽트)	보통 더 짧은 파이프라인, 초저전력, 확정적 (저지연 인터럽트)
시스템 기능	메모리 관리 장치(MMU), 캐시 메모리, ARM TrustZone 보안 확장	메모리 보호 장치(MPU), 캐시 메모리, 밀착 결합 메모리(Tightly Coupled Memory)	메모리 보호 장치(MPU), 중첩 벡터형 인터럽트 컨트롤러 (NVIC), 웨이크업 인터럽트 컨트롤러(WIC), 최신 ARM TrustZone 보안 기능 확장
용도	모바일 컴퓨팅, 스마트폰, 에너지 효율 서버, 하이엔드 마이크로프로세서	산업용 마이크로컨트롤러, 자동차, 하드디스크 컨트롤러, 베이스밴드 모뎀	마이크로컨트롤러, 심층 임베디드(Deeply embedded) 시스템 (예: 센서, MEMS, 혼합 신호 IC), 사물인터넷 (IoT)

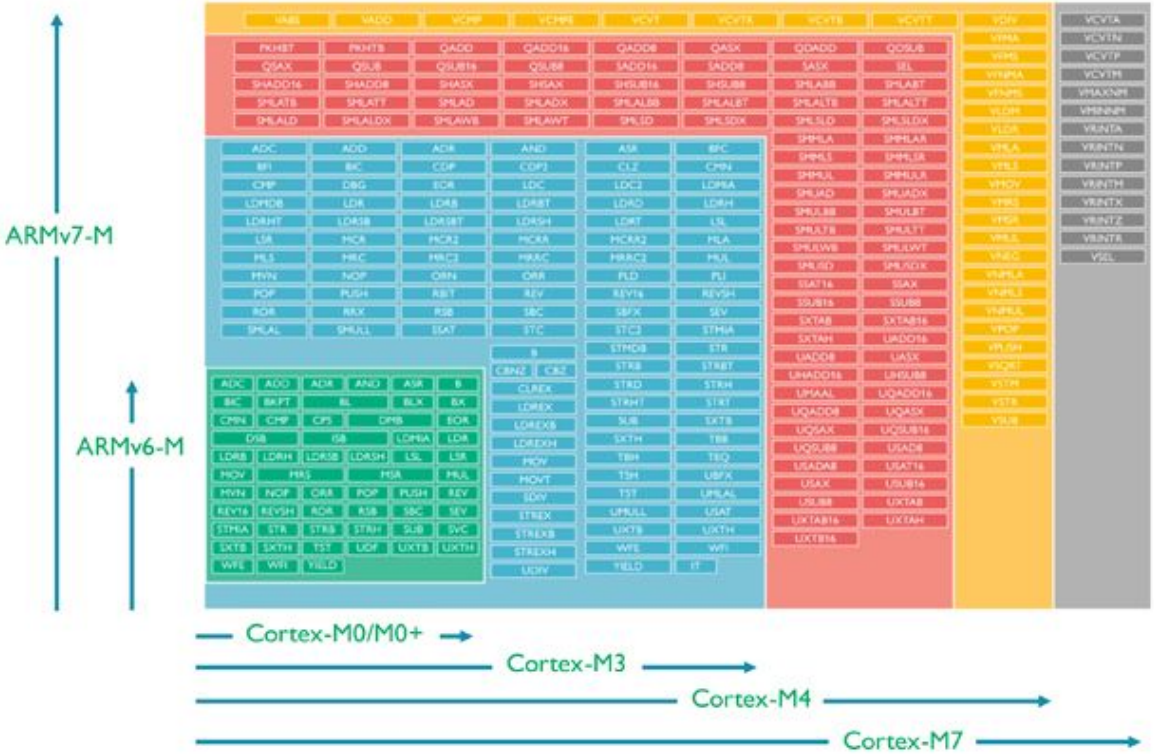
Cortex-M


Total units shipped*
High performance
Performance efficiency
Lowest power & area
Cortex-M7
Maximum performance, control and DSP
ARM TRUSTZONE
System security
Cortex-M3
Performance efficiency
Cortex-M4
Mainstream control and DSP
Cortex-M33
Flexibility, control and DSP with TrustZone
Cortex-M0
Lowest cost, low power
Cortex-M0+
Highest energy efficiency
Cortex-M23
TrustZone in smallest area, lowest power
ARMv7-M
ARMv6-M
ARMv8-M
*Data as of Dec. 2016

Cortex-M

프로세스	설명
Cortex-M0	저비용, 초저전력 마이크로컨트롤러 및 고도의 임베디드 애플리케이션용 초소형 프로세서(12K 게이트에서 시작)
Cortex-M0+	소형 임베디드 시스템을 위한 최고 수준의 에너지 효율 프로세서. Cortex-M0 프로세서와 비슷한 크기와 프로그래머 모델이지만 단일 사이클 I/O 인터페이스와 벡터 테이블 재배치등 추가 기능이 포함되어 있다.
Cortex-M1	FPGA 설계에 최적화된 소형 프로세서 설계이며 FPGA의 메모리 블록을 통해 밀착 결합 메모리(Tightly Coupled Memory)를 구현한다. Cortex-M0과 동일한 명령어 세트
Cortex-M3	복잡한 작업을 보다 신속하게 처리할 수 있도록 많은 명령어가 설정된 저전력 마이크로 컨트롤러를 위한 작지만 강력한 임베디드 프로세서. 하드웨어 디바이더(Hardware Divider)와 Multiply-Accumulate(MAC) 명령어를 탑재하고 있다. 이와 함께 소프트웨어 개발자가 애플리케이션을 더 빠르게 개발할 수 있도록 포괄적인 디버그 및 추적 기능을 지원한다.
Cortex-M4	이 프로세서는 Cortex-M3의 모든 기능을 제공하며, Single Instruction Multiple Data(SIMD)와 보다 빠른 단일 사이클 MAC 연산과 같은 디지털 신호 처리(Digital Signal Processing) 작업에서 추가 명령어 대상을 제공한다. 또한 IEEE 754 부동 소수점 표준을 지원하는 단정도(Single Precision) 부동 소수점 유닛 (옵션)도 제공한다.
Cortex-M7	하이엔드 마이크로컨트롤러 및 프로세싱 집약적인 애플리케이션용 고성능 프로세서. Cortex-M4에서 사용할 수 있는 모든 ISA 기능과 캐시 및 밀착 결합 메모리(Tightly Coupled Memory)와 같은 추가 메모리 기능뿐만 아니라 배정도(double precision) 부동 소수점을 위한 추가 지원을 제공한다.
Cortex-M23	초저전력과 저비용 설계용으로 설계된 소형 프로세서로 Cortex-M0+ 프로세서와 비슷하지만 명령어 세트 및 시스템 수준의 기능이 다양하게 향상됐다. 이와 함께 TrustZone 보안 기능 확장을 지원한다.
Cortex-M33	기존 Cortex-M3및 Cortex-M4 프로세서와 비슷하지만 더욱 향상된 유연성을 갖춘 시스템 설계와 보다 나은 에너지 효율성과 성능을 제공하는 주류 프로세서 설계. 또한 이 프로세서는 TrustZone 보안 확장을 지원한다.

Cortex-M Command Set



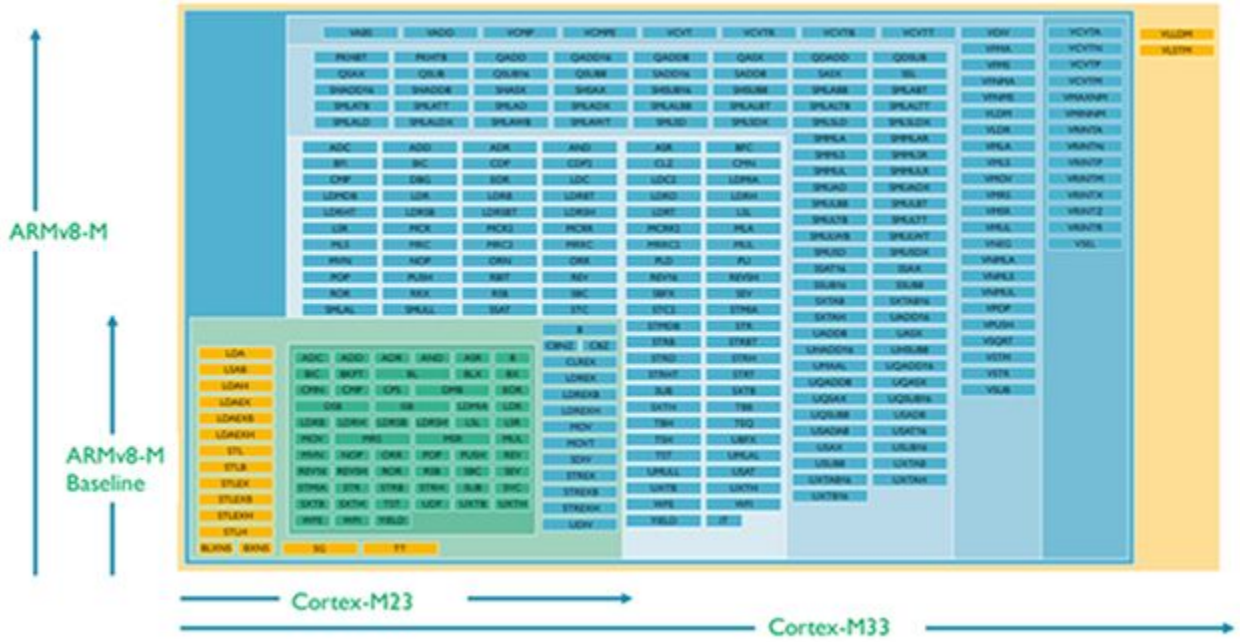
Floating Point

DSP (SIMD, fast MAC)

Advanced data processing
bit field manipulations

General data processing
I/O control tasks

Cortex-M Command Set



Cortex-M Command Set

Architecture	설 명
ARMv6-M	Cortex-M0, Cortex-M0+ 및 Cortex-M1 프로세서용.
ARMv7-M	Cortex-M3, Cortex-M4 및 Cortex-M7 프로세서용. DSP 유형의 명령어 (예: SIMD)를 지원하기 위한 ARMv7-M의 확장은 ARMv7E-M으로 명명된다.
ARMv8-M	이 아키텍처 릴리스는 다음과 같이 나뉘어진다. Cortex-M23 프로세서용 Baseline 서브 프로파일 Cortex-M33 프로세서용 Mainline 서브 프로파일

Cortex-M 생산 회사

STM32

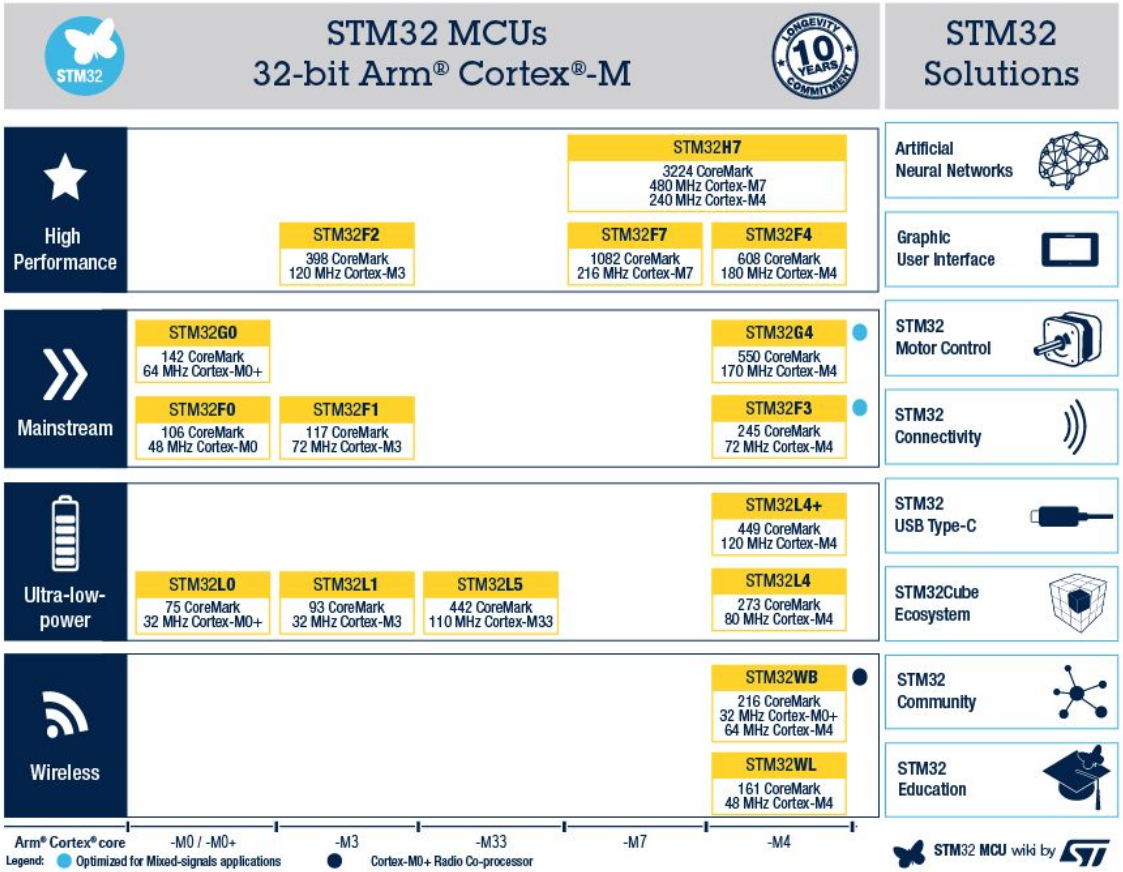
STMicroelectronics

스위스 제네바에 본사를 둔 전자제품과 반도체를 생산하는 기업

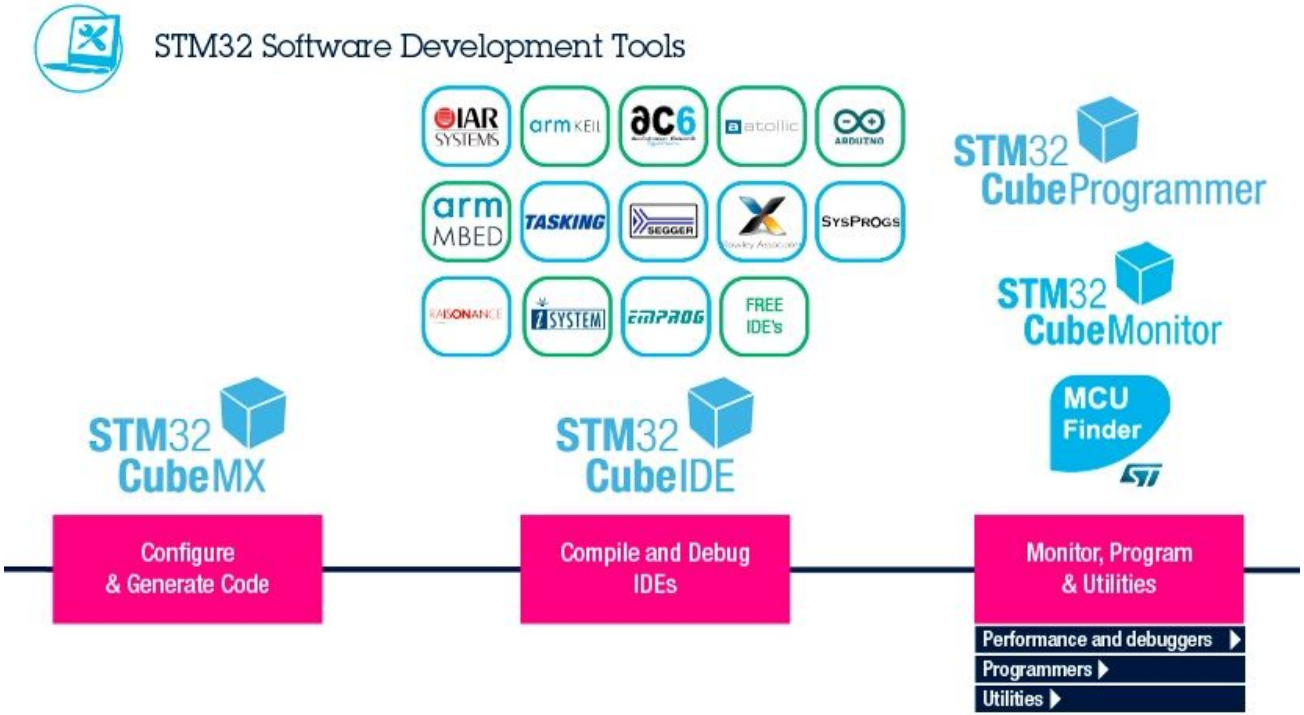
STM32

32-bit ARM Cortex-m processor core를 사용하는 STMicroelectronics에서 생산하는 Microcontroller 제품군

STM32


STM32 MCUs
32-bit Arm® Cortex®-M
LONGEVITY 10 YEARS COMMITMENT
STM32 Solutions
High Performance
STM32H7
3224 CoreMark
480 MHz Cortex-M7
240 MHz Cortex-M4
STM32F2
398 CoreMark
120 MHz Cortex-M3
STM32F7
1082 CoreMark
216 MHz Cortex-M7
STM32F4
608 CoreMark
180 MHz Cortex-M4
Mainstream
STM32G0
142 CoreMark
64 MHz Cortex-M0+
STM32G4
550 CoreMark
170 MHz Cortex-M4
STM32F0
106 CoreMark
48 MHz Cortex-M0
STM32F1
117 CoreMark
72 MHz Cortex-M3
STM32F3
245 CoreMark
72 MHz Cortex-M4
Ultra-low-power
STM32L4+
449 CoreMark
120 MHz Cortex-M4
STM32L0
75 CoreMark
32 MHz Cortex-M0+
STM32L1
93 CoreMark
32 MHz Cortex-M3
STM32L5
442 CoreMark
110 MHz Cortex-M33
STM32L4
273 CoreMark
80 MHz Cortex-M4
Wireless
STM32WB
216 CoreMark
32 MHz Cortex-M0+
64 MHz Cortex-M4
STM32WL
161 CoreMark
48 MHz Cortex-M4
Artificial Neural Networks
Graphic User Interface
STM32 Motor Control
STM32 Connectivity
STM32 USB Type-C
STM32Cube Ecosystem
STM32 Community
STM32 Education
Arm® Cortex® core
-M0 / -M0+
-M3
-M33
-M7
-M4
Legend:
Optimized for Mixed-signals applications
Cortex-M0+ Radio Co-processor
STM32 MCU wiki by

STM32 S/W Tools

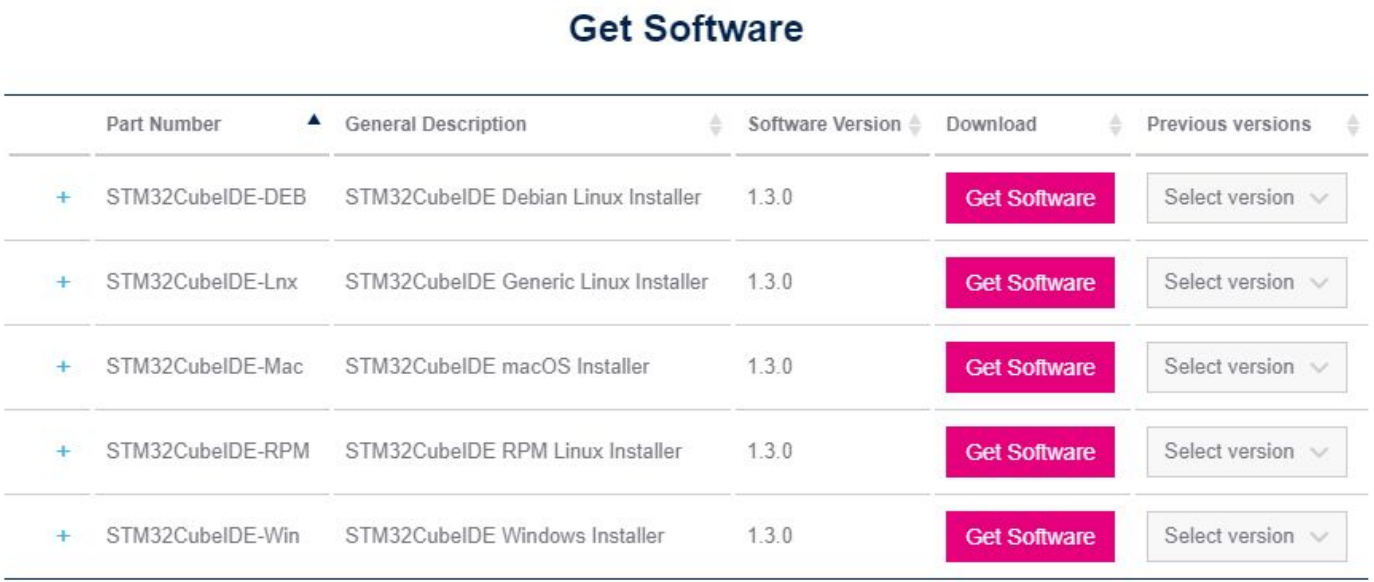


STM32CubeIDE

Eclipse 기반의 STM32 개발을 위한 통합 IDE

STM32CubeIDE 다운로드

https://www.st.com/en/development-tools/stm32cubeide.html



STM32CubeIDE 다운로드

License Agreement

ACCEPT

Please indicate your acceptance or NON-acceptance by selecting “I ACCEPT” or “I DO NOT ACCEPT” as indicated below in the media.

BY INSTALLING COPYING, DOWNLOADING, ACCESSING OR OTHERWISE USING THIS SOFTWARE PACKAGE OR ANY PART THEREOF (AND THE RELATED DOCUMENTATION) FROM STMICROELECTRONICS INTERNATIONAL N.V, SWISS BRANCH AND/OR ITS AFFILIATED COMPANIES (STMICROELECTRONICS), THE RECIPIENT, ON BEHALF OF HIMSELF OR HERSELF, OR ON BEHALF OF ANY ENTITY BY

Get Software

If you have an account on my.st.com, login and download the software without any further validation steps.

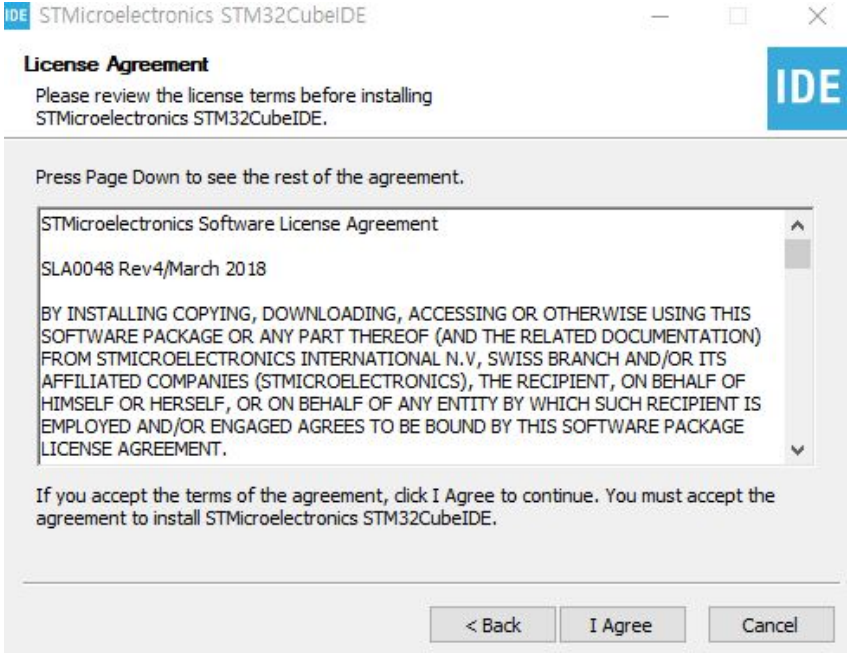
Login/Register

If you don't want to login now, you can download the software by simply providing your name and e-mail address in the form below and validating it.

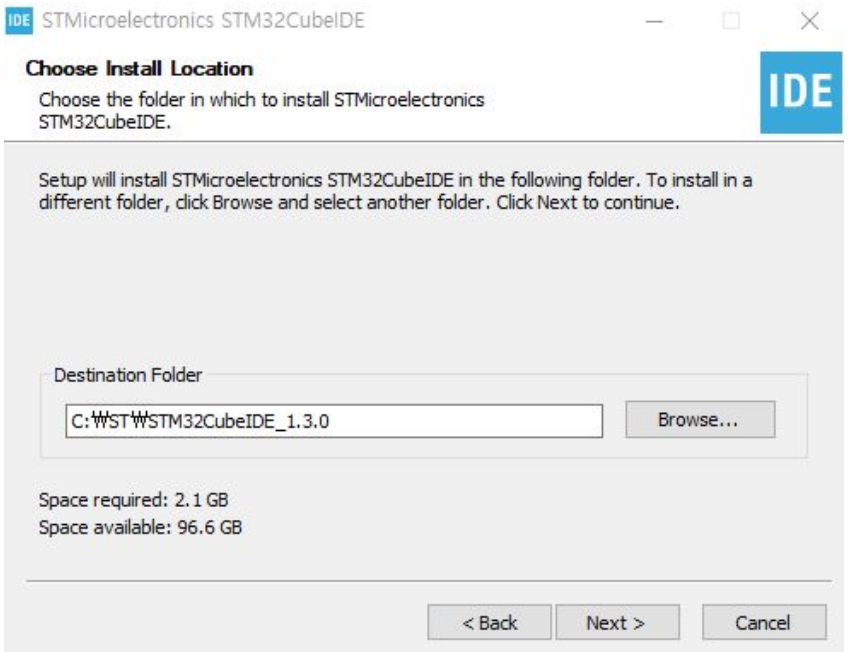
This allows us to stay in contact and inform you about updates of this software.

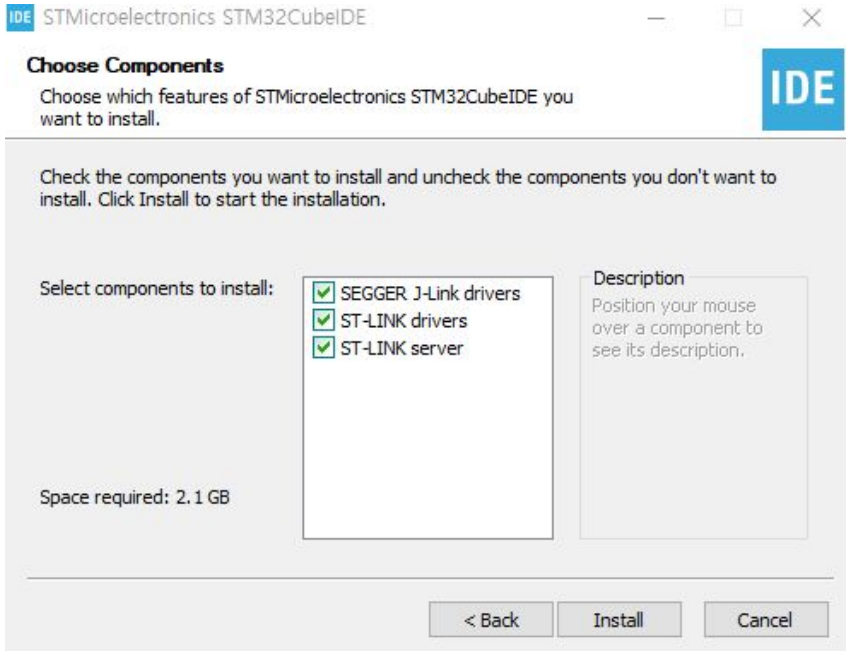
For subsequent downloads this step will not be required for most of our software.

STM32CubeIDE 설치

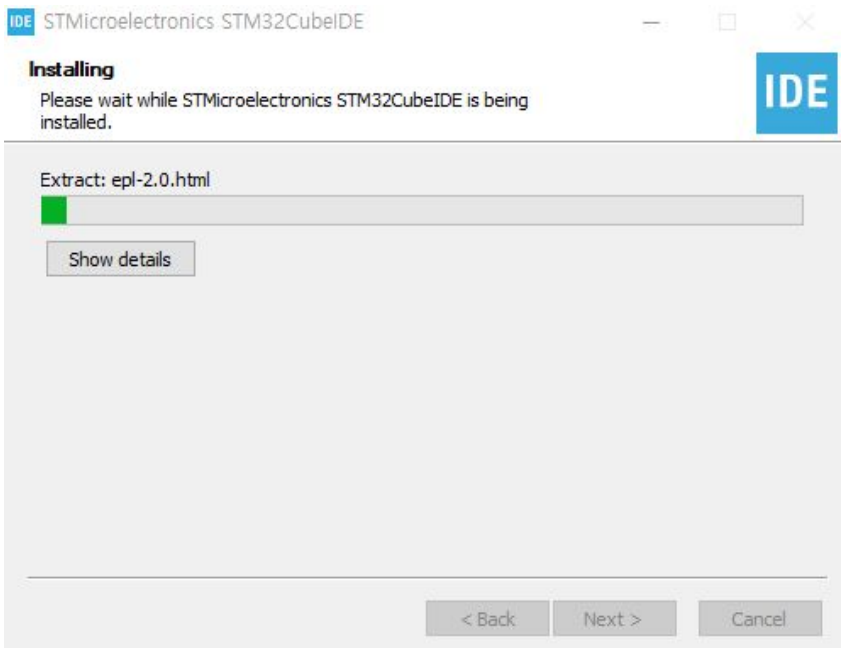


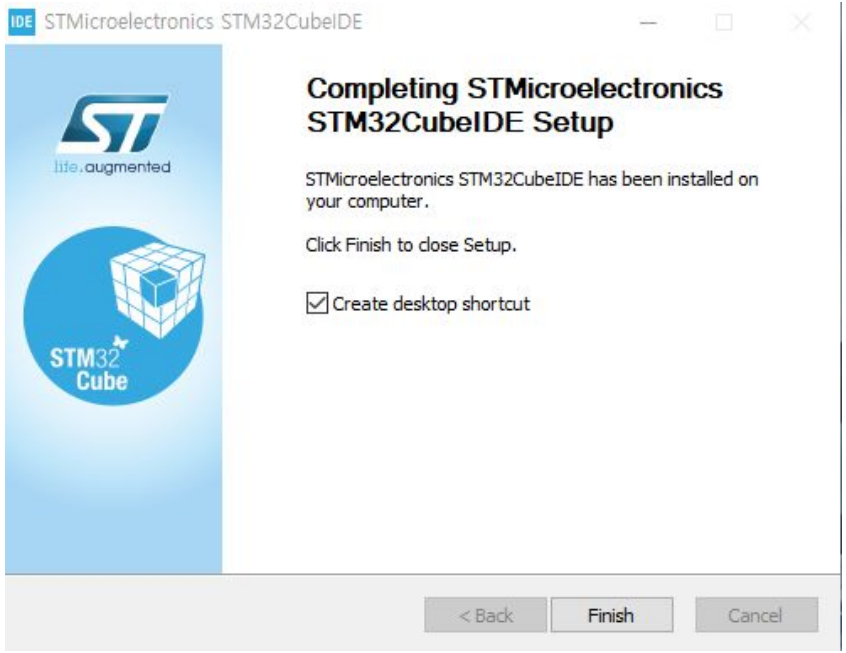
STM32CubeIDE 설치



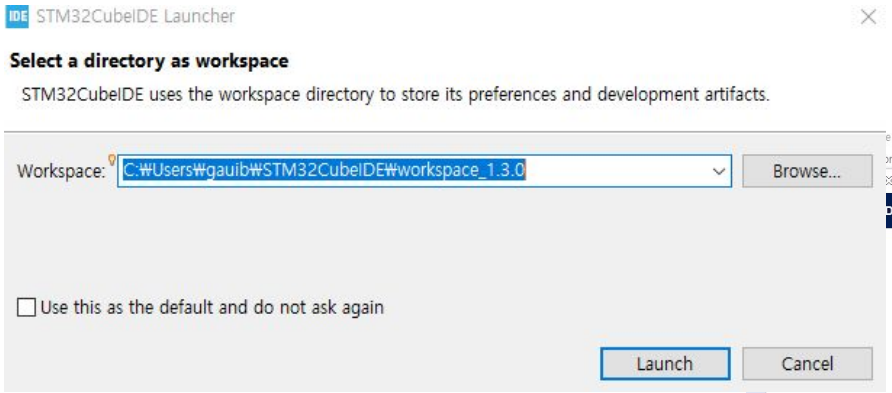


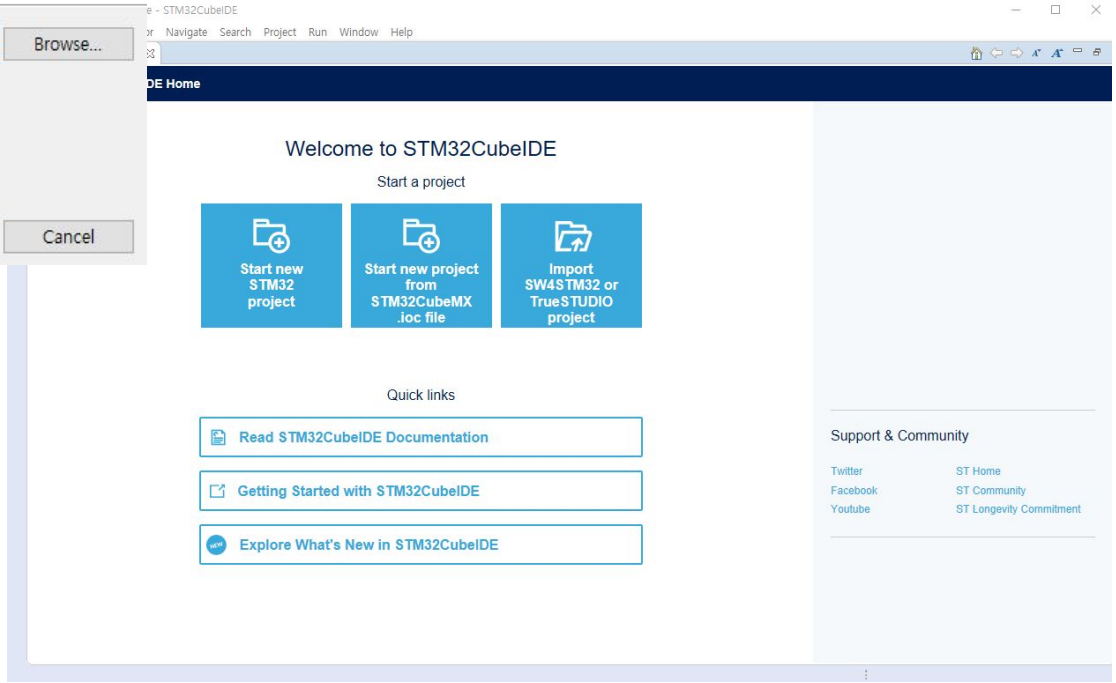
STM32CubeIDE 설치



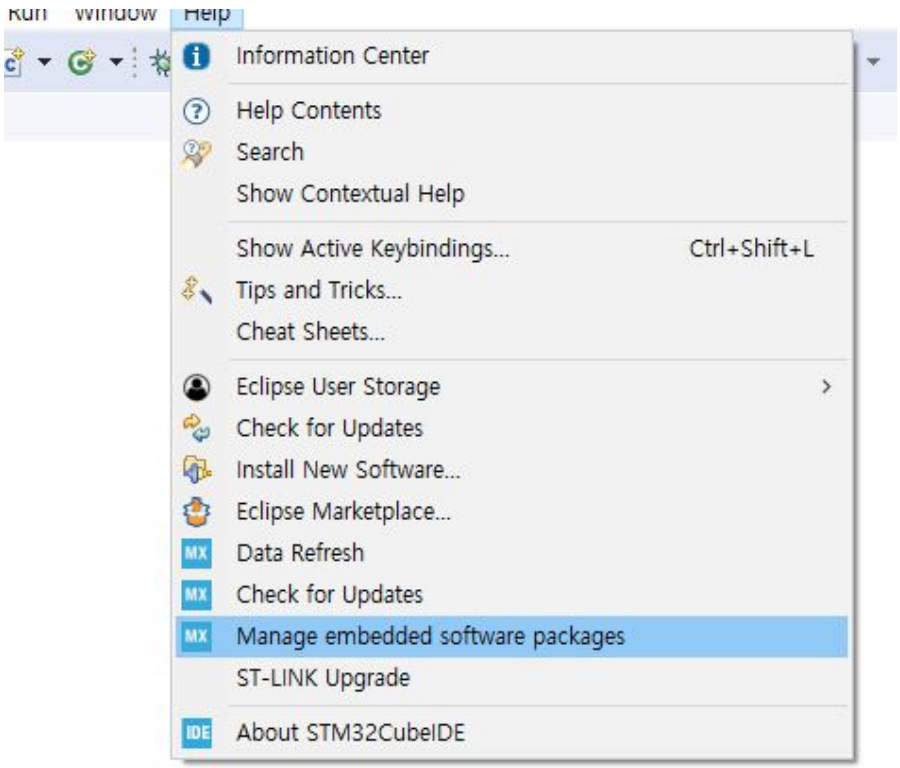


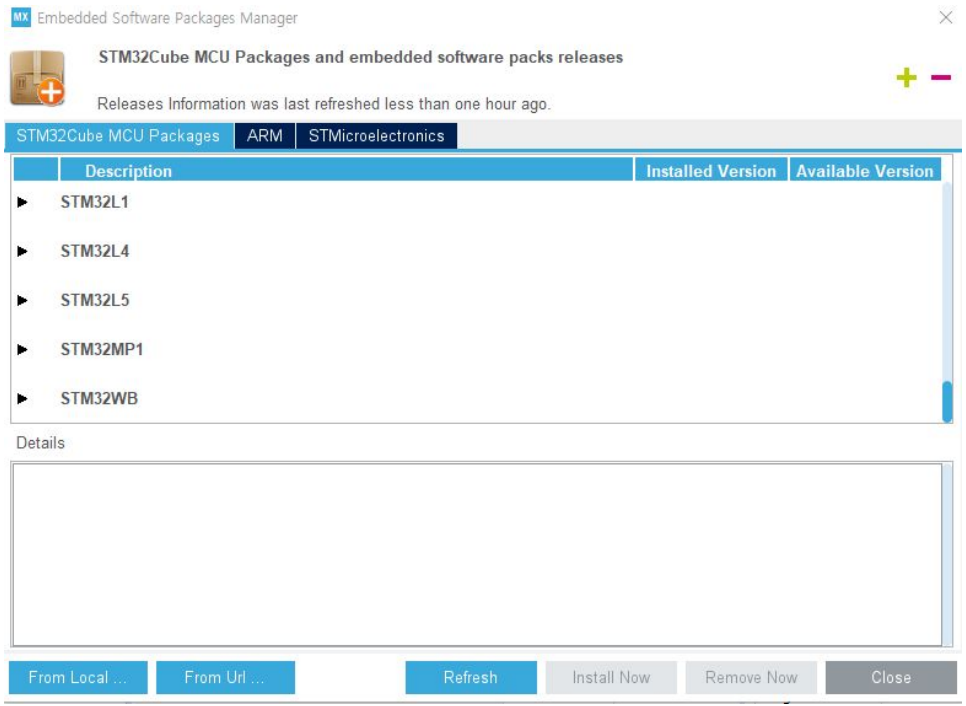
STM32CubeIDE 실행



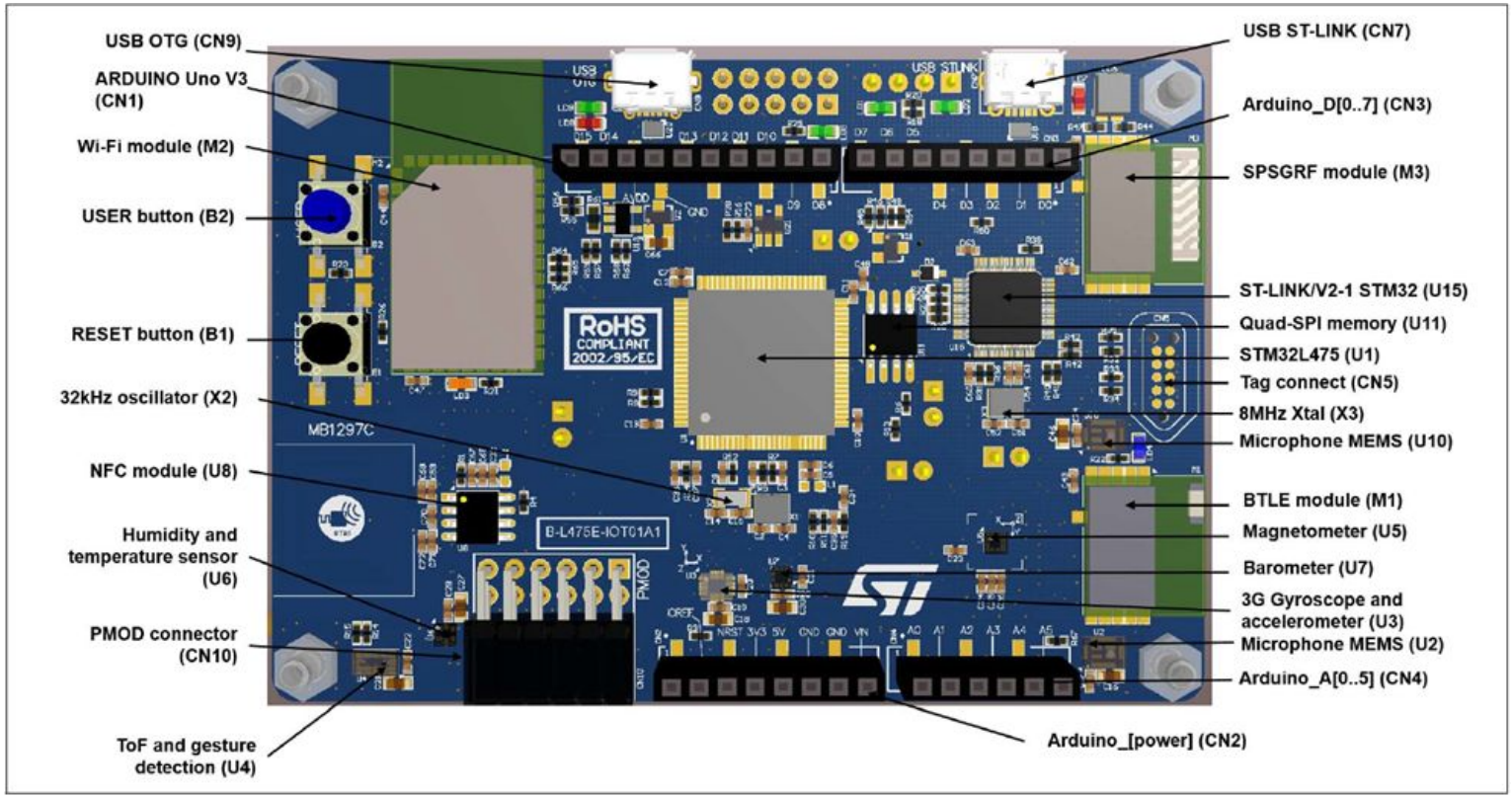


STM32CubeIDE S/W package 관리

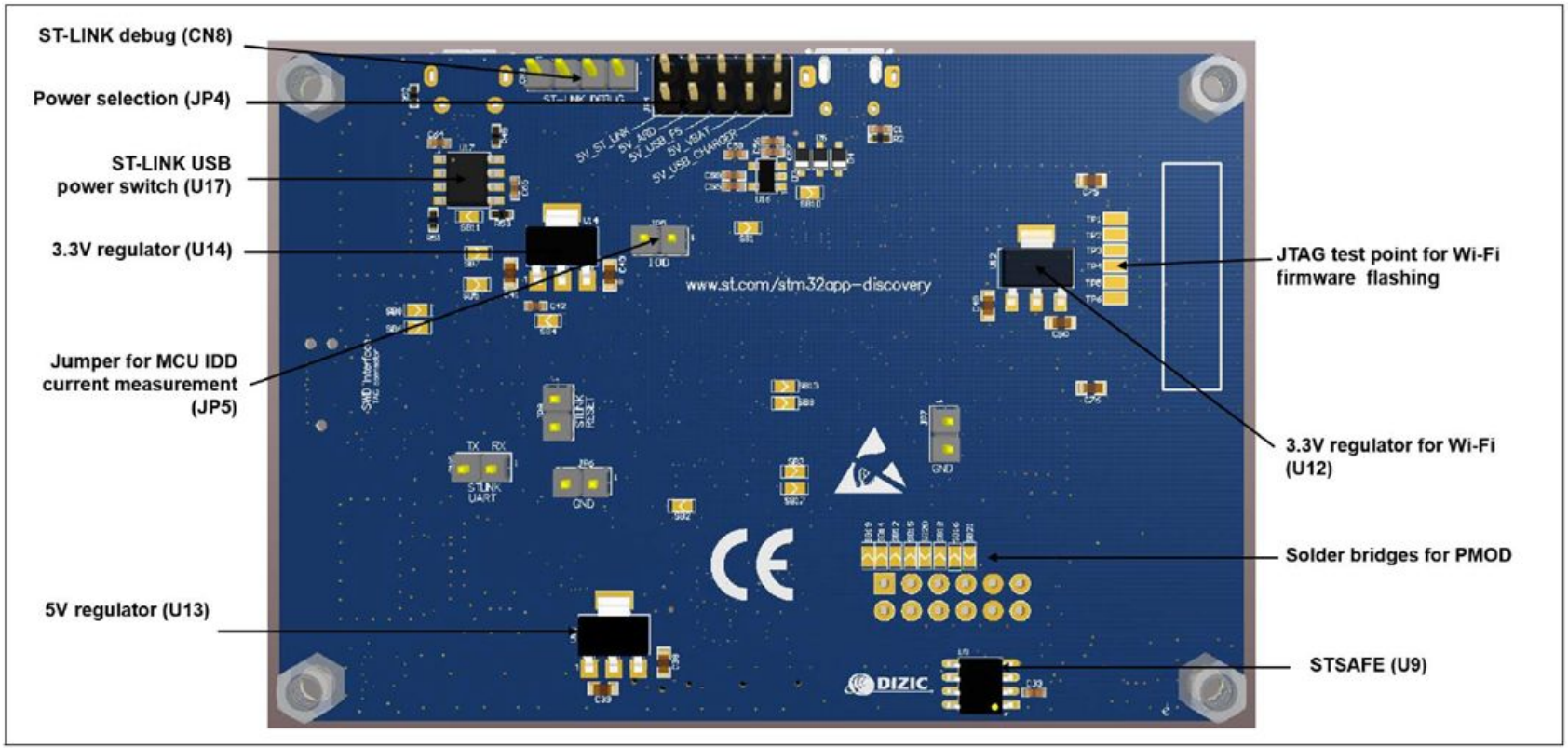




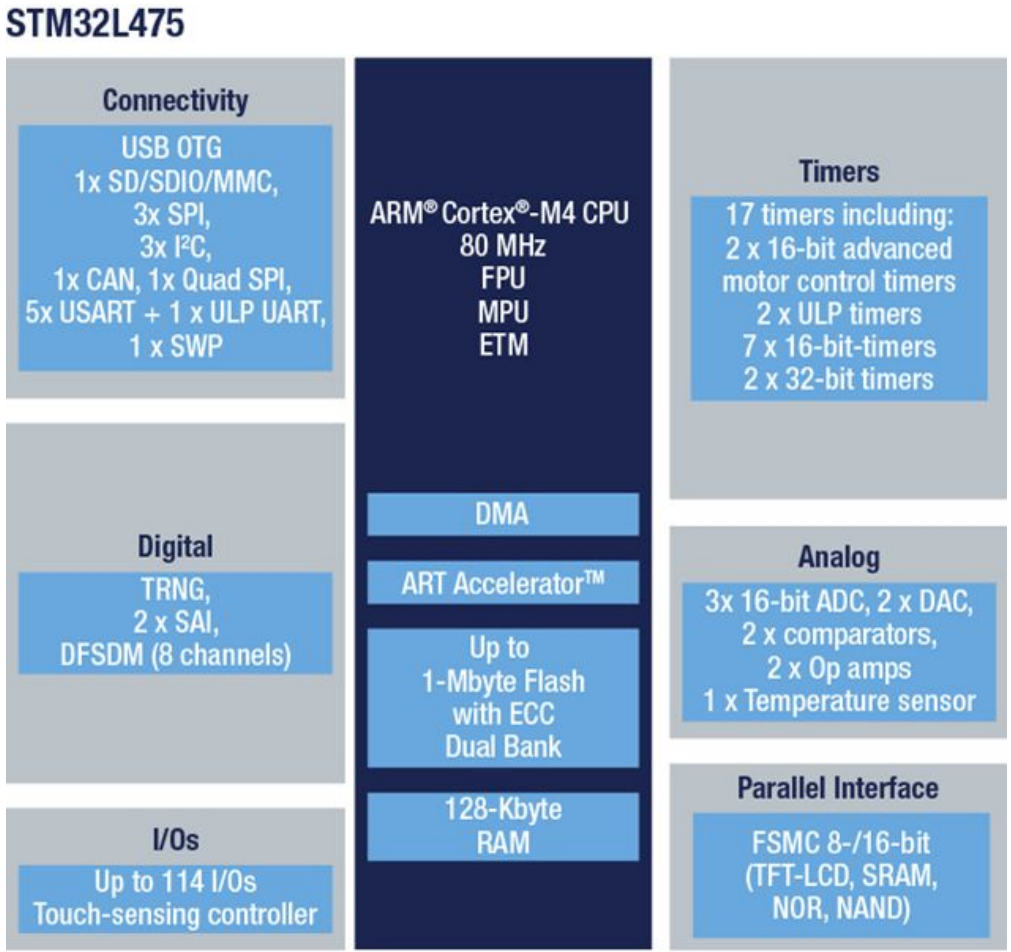
실습보드(B-L475E-IOT01A)



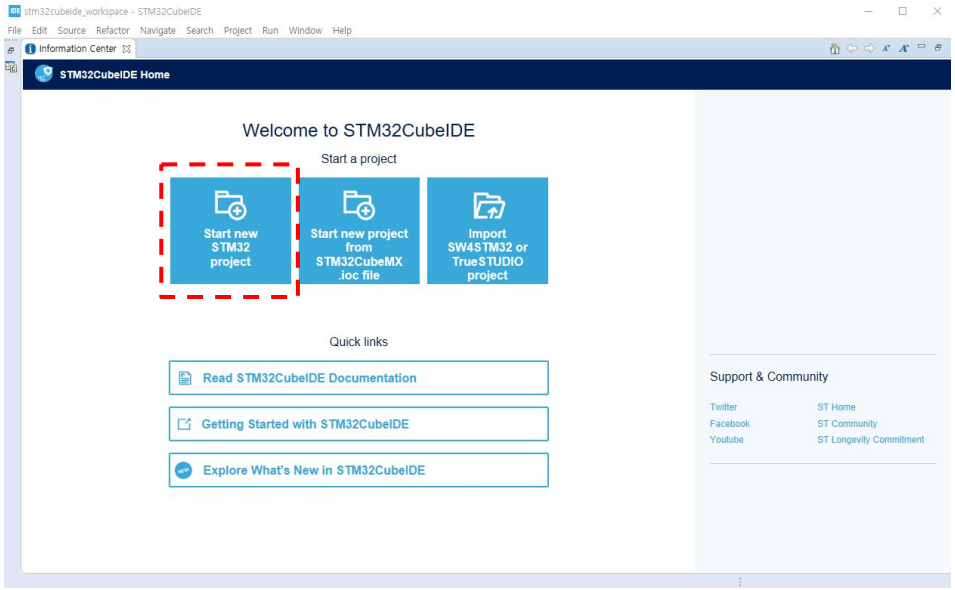
실습보드(B-L475E-IOT01A)

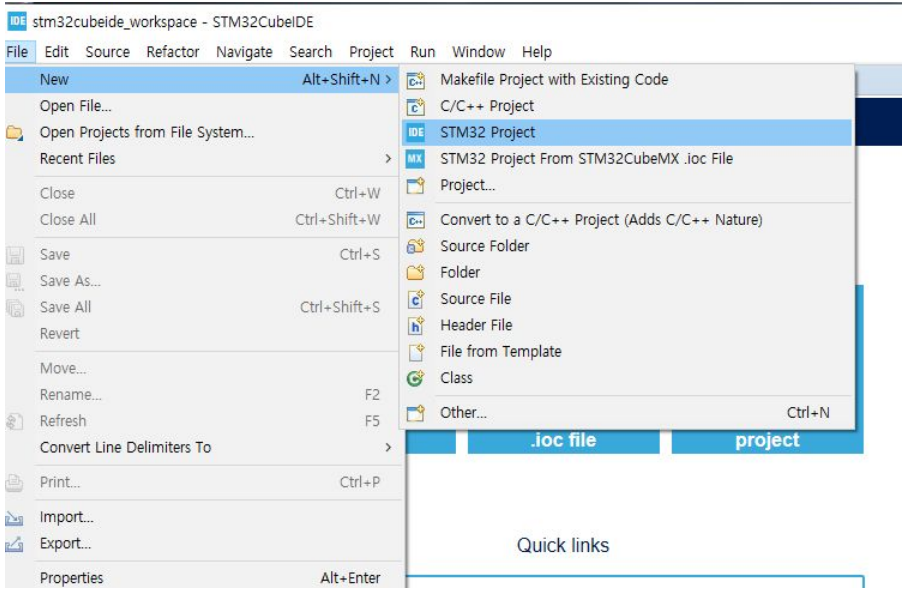


실습보드


STM32L475
Connectivity
USB OTG
1x SD/SDIO/MMC,
3x SPI,
3x I²C,
1x CAN, 1x Quad SPI,
5x USART + 1 x ULP UART,
1 x SWP
Digital
TRNG,
2 x SAI,
DFSDM (8 channels)
I/Os
Up to 114 I/Os
Touch-sensing controller
ARM® Cortex®-M4 CPU
80 MHz
FPU
MPU
ETM
DMA
ART Accelerator™
Up to
1-Mbyte Flash
with ECC
Dual Bank
128-Kbyte
RAM
Timers
17 timers including:
2 x 16-bit advanced
motor control timers
2 x ULP timers
7 x 16-bit-timers
2 x 32-bit timers
Analog
3x 16-bit ADC, 2 x DAC,
2 x comparators,
2 x Op amps
1 x Temperature sensor
Parallel Interface
FSMC 8-/16-bit
(TFT-LCD, SRAM,
NOR, NAND)

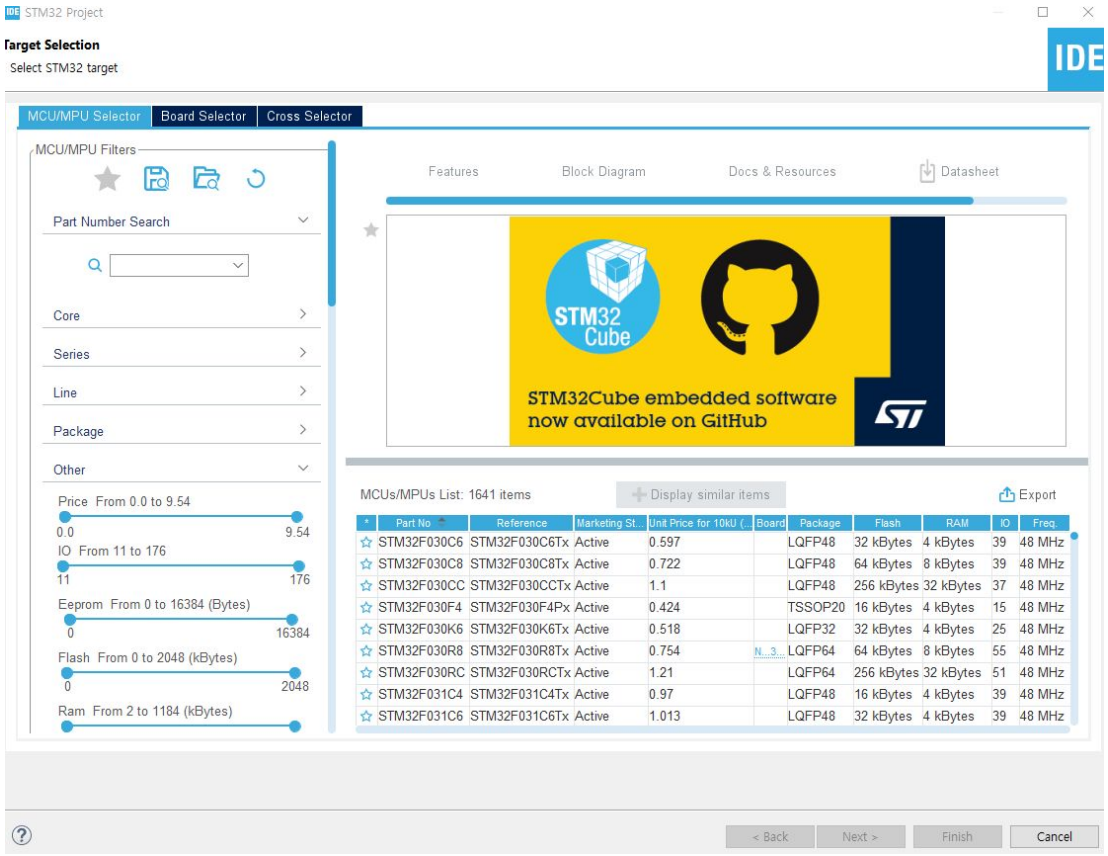
STM32CubeIDE Project 생성





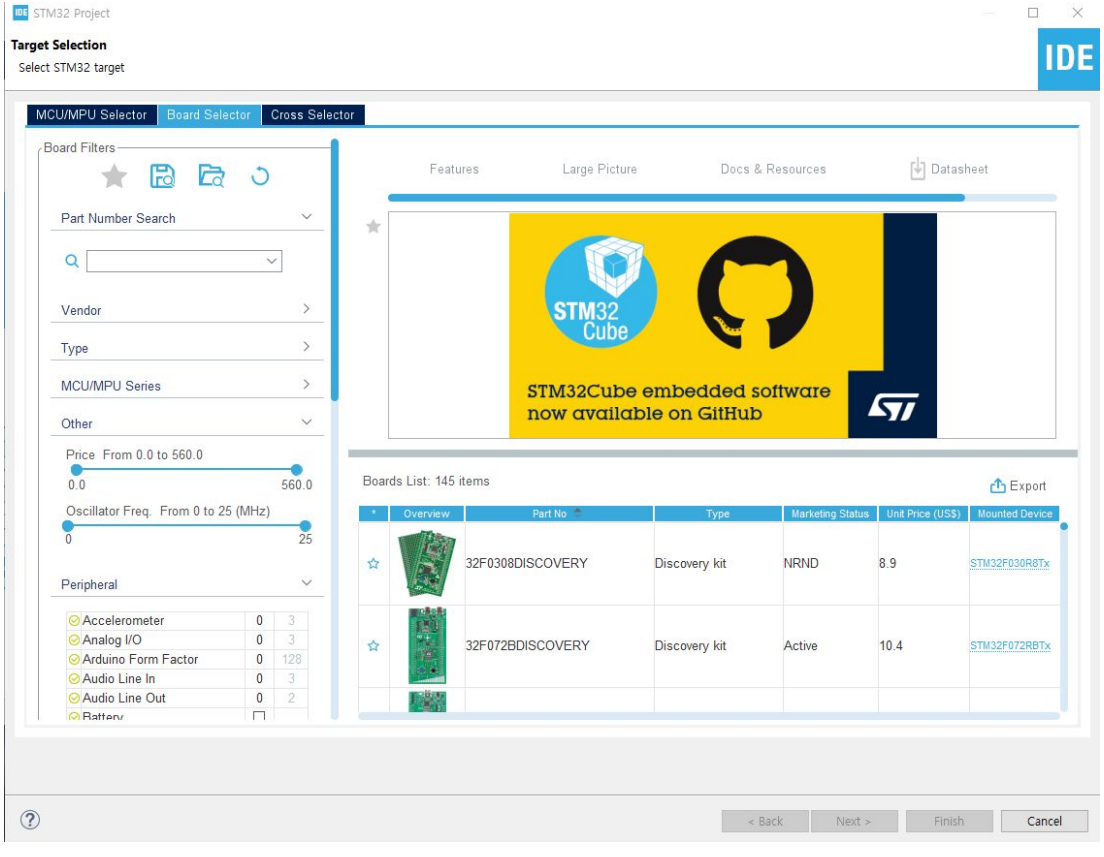
STM32CubeIDE Project 생성

MCU/MPU Selector



STM32CubeIDE Project 생성

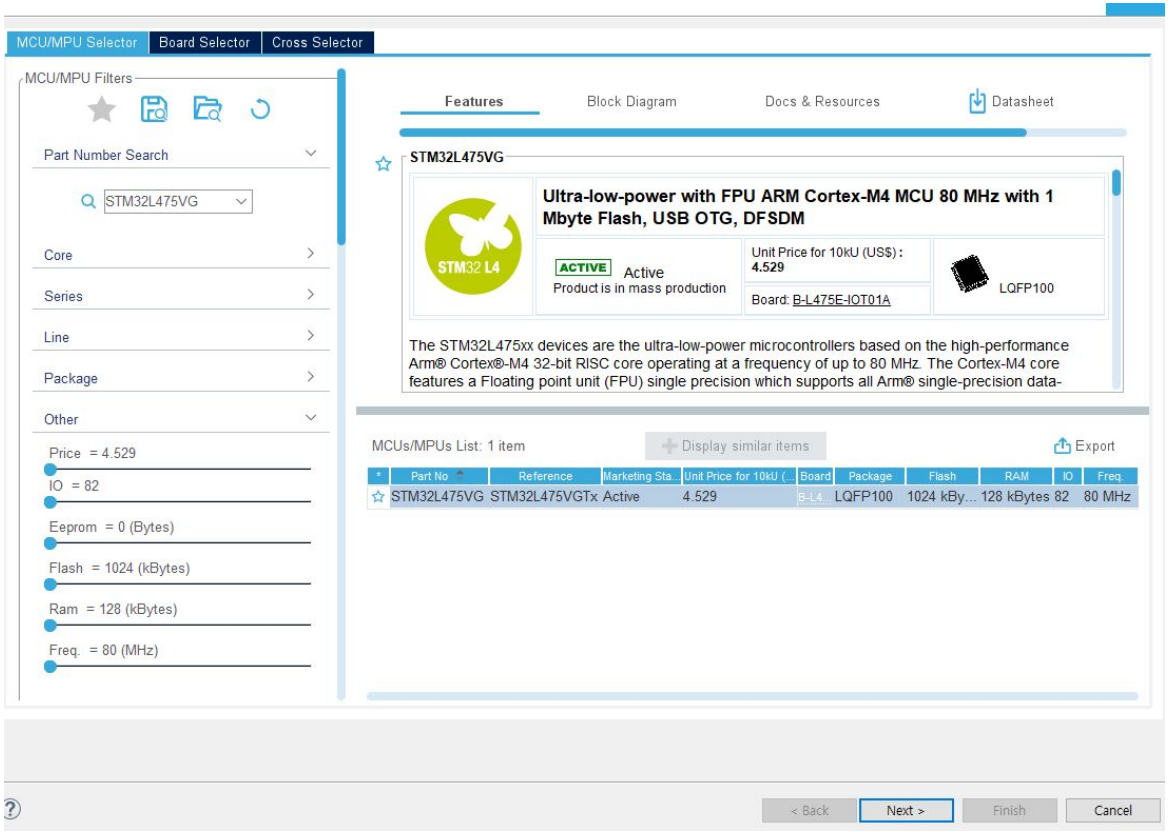
Board Selector



STM32CubeIDE Project 생성

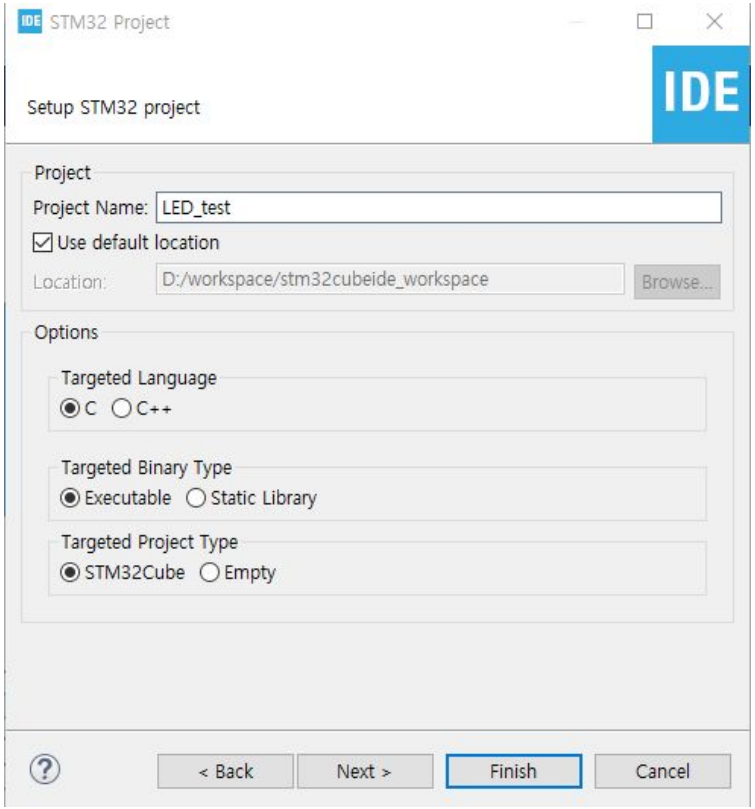
사용할 MCU의

Part Number 선택

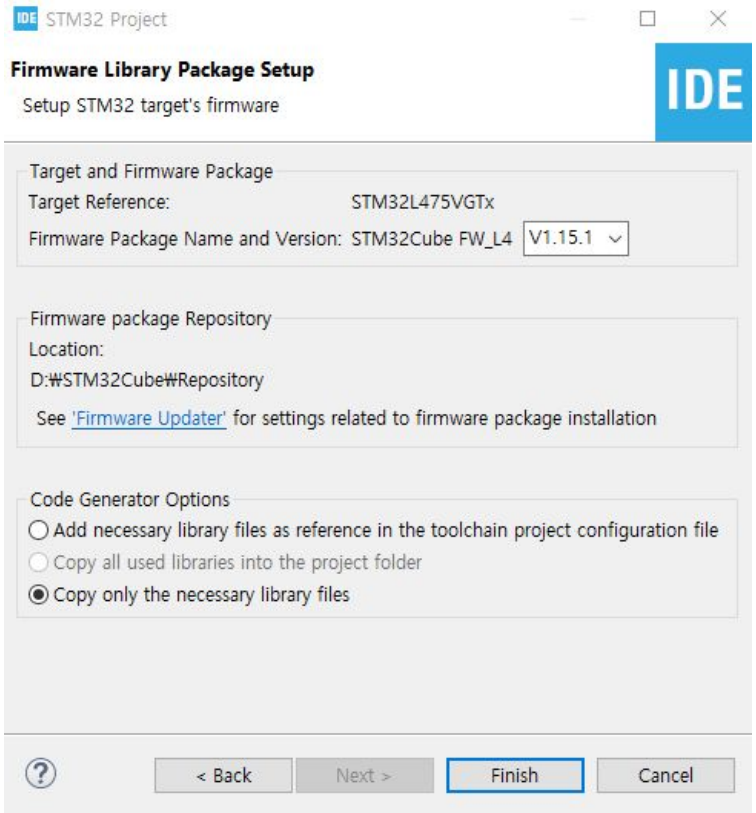


STM32CubeIDE Project 생성

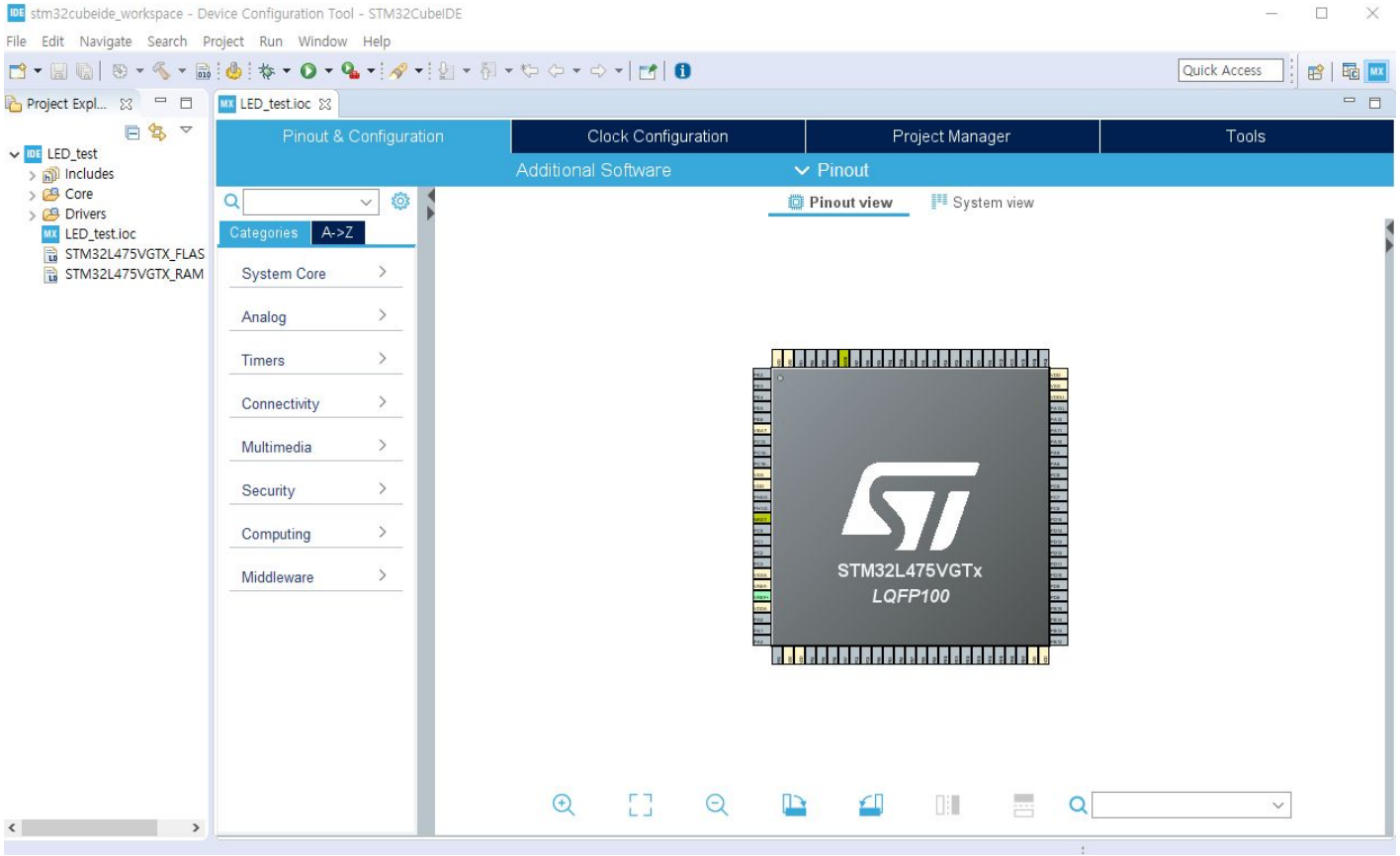
Project
이름설정



S/W
package
설정



STM32CubeIDE Project 생성



Firmware

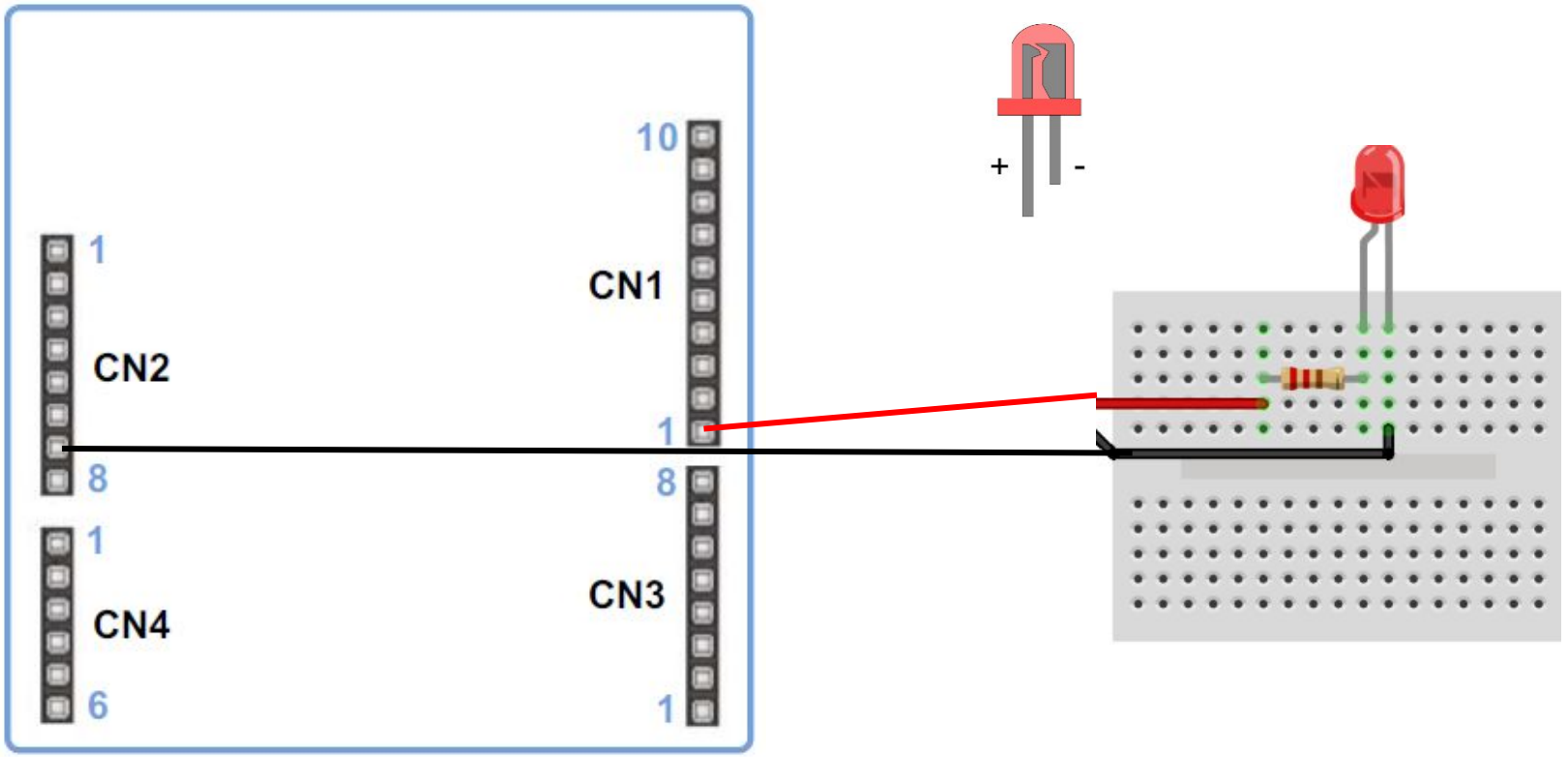
Firmware

컴퓨팅과 공학 분야에서 특정 하드웨어 장치에 포함된 소프트웨어, 소프트웨어를 읽어 실행하거나, 수정되는 것도 가능한 장치를 뜻한다. 펌웨어는 ROM이나 PROM에 저장되며, 하드웨어보다는 교환하기가 쉽지만, 소프웨어보다는 어렵다.

인용 : 위키백과 (https://ko.wikipedia.org/wiki/%ED%8E%8C%EC%9B%A8%EC%96%B4)

Firmware : 임베디드 하드웨어를 동작시키며 특정 기능의 수행을 목표로하는 프로그램

LED Blinky



LED Blinky

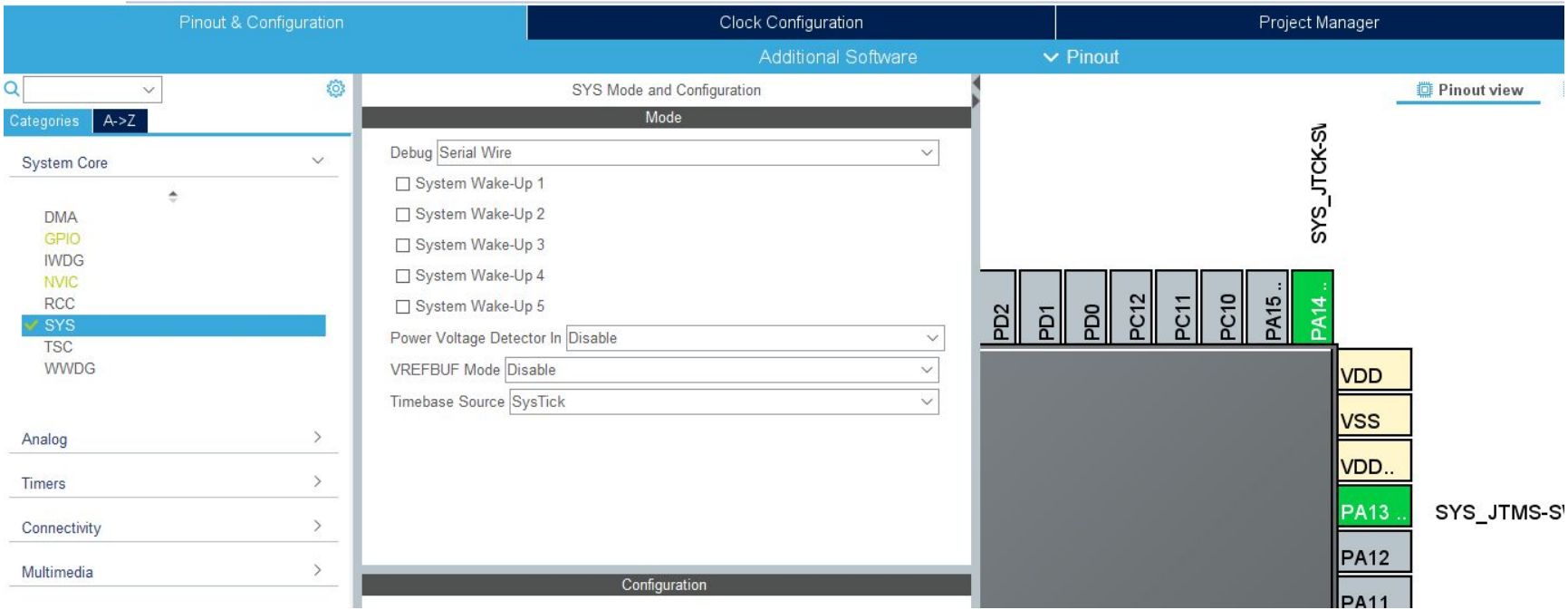
Connector	Pin number	Pin name	Signal name	STM32 pin	Function
CN2	1	NC	-	-	-
	2	IOREF	-	-	3.3 V reference
	3	NRST	STM_NRST	NRST	Reset
	4	3.3 V	-	-	3.3 V input/output
	5	5V	-	-	5V
	6	GND	-	-	GND
	7	GND	-	-	GND
	8	VIN	-	-	Power input
CN4	1	A0	ARD.A0-ADC	PC5	ADC
	2	A1	ARD.A1-ADC	PC4	ADC
	3	A2	ARD.A2-ADC	PC3	ADC
	4	A3	ARD.A3-ADC	PC2	ADC
	5	A4	ARD.A4-ADC	PC1	ADC / I2C3_SDA
	6	A5	ARD.A5-ADC	PC0	ADC / I2C3_SCL

LED Blinky

CN1	10	SCL/D15	ARD.D15-I2C1_SCL	PB8	I2C1_SCL
	9	SDA/D14	ARD.D14-I2C1_SDA	PB9	I2C1_SDA
	8	AVDD	VDDA	-	VDDA
	7	GND	GND	-	Ground
	6	SCK/D13	ARD.D13-SPI1_SCK/LED1	PA5	SPI1_SCK / LED1
	5	MISO/D12	ARD.D12-SPI1_MISO	PA6	SPI1_MISO
	4	PWM/MOSI/D11	ARD.D11-SPI1_MISO/PWM	PA7	SPI1_MOSI / TIMxx
	3	PWM/CS/D10	ARD.D10-SPI_SSN/PWM	PA2	TIM2_CH3
	2	PWM/D9	ARD.D9-PWM	PA15	TIM2_CH1
	1	D8	ARD.D8	PB2	GPIO
CN3	8	D7	ARD.D7	PA4	GPIO
	7	PWM/D6	ARD.D6-PWM	PB1	TIM3_CH4
	6	PWM/D5	ARD.D5-PWM	PB4	TIM3_CH1
	5	D4	ARD.D4	PA3	TIMxx
	4	PWM/D3	ARD.D3-PWM/INT1_EXTI0	PB0	TIM3_CH3 / EXTI0
	3	D2	ARD.D2-INT0_EXTI14	PD14	EXTI14
	2	TX/D1	ARD.D1-UART4_TX	PA0	UART4_TX
	1	RX/D0	ARD.D0-UART4_RX	PA1	UART4_RX

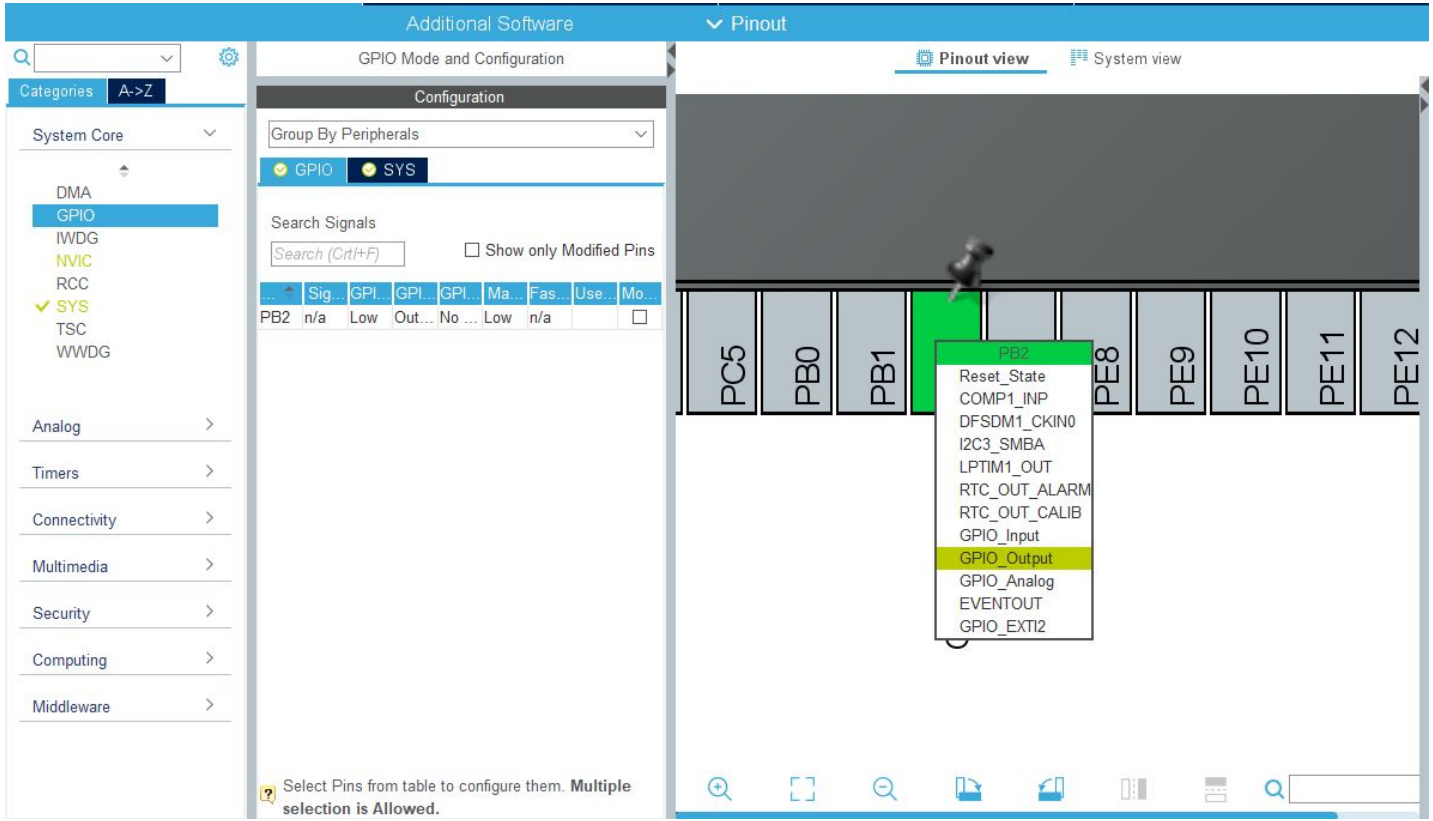
LED Blinky

Debug Pin 설정



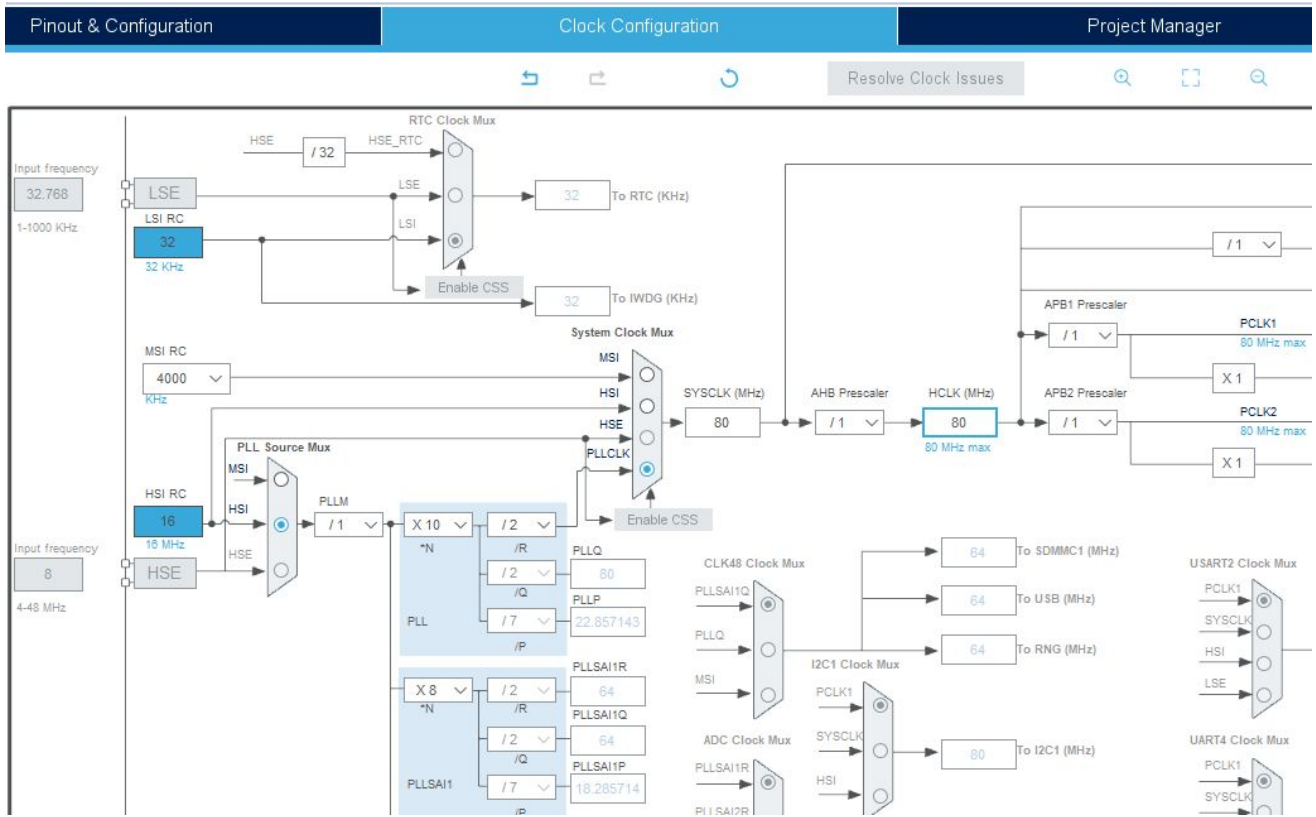
LED Blinky

LED pin 설정



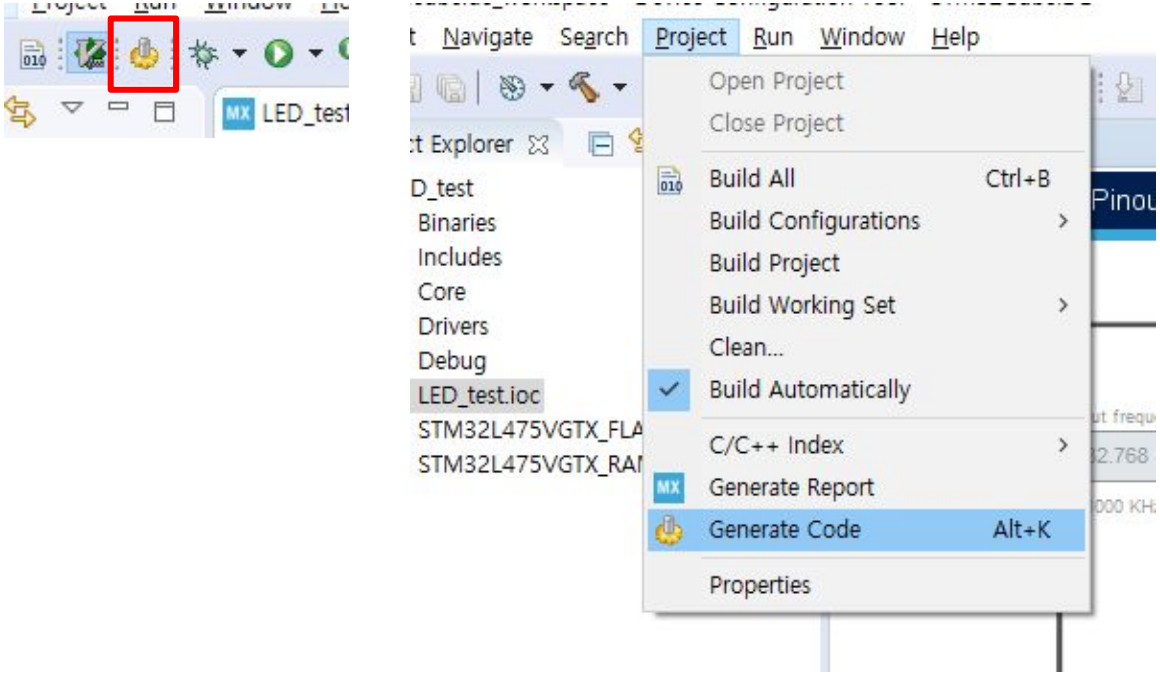
LED Blinky

Clock 설정



LED Blinky

코드 생성



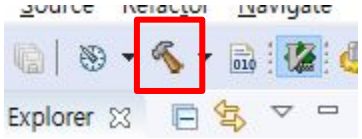
LED Blinky

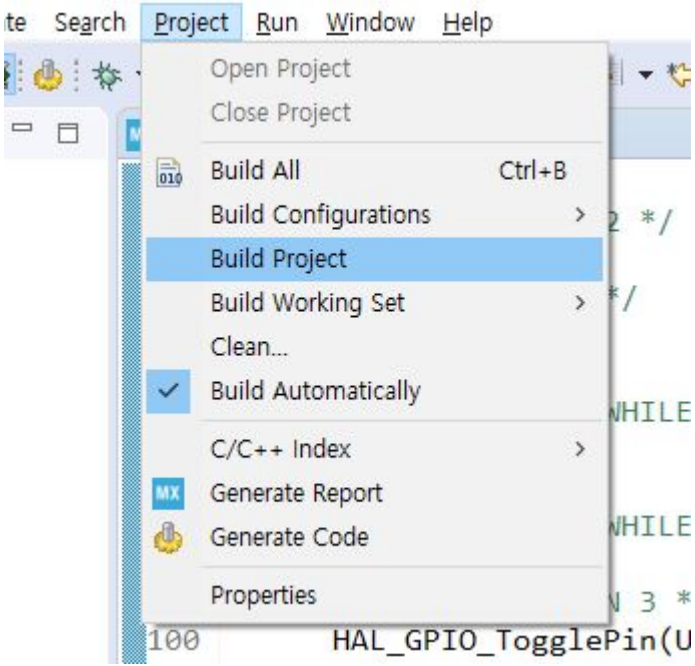
User code 작성

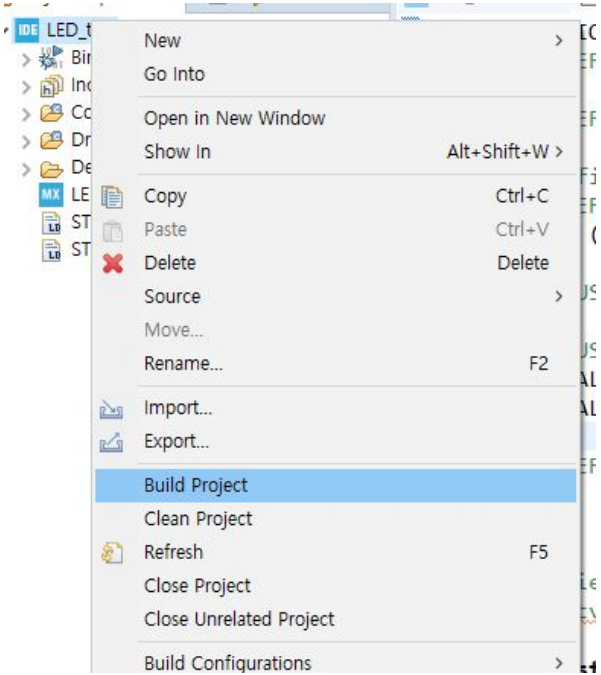
```
90
91    /* USER CODE END 2 */
92
93    /* Infinite loop */
94    /* USER CODE BEGIN WHILE */
95    while (1)
96    {
97      /* USER CODE END WHILE */
98
99      /* USER CODE BEGIN 3 */
00        HAL_GPIO_TogglePin(USER_LED_GPIO_Port, USER_LED_Pin);
01        HAL_Delay(500);
02    }
03    /* USER CODE END 3 */
04  }
05
```

LED Blinky

Build Project

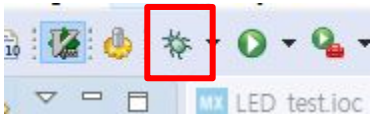


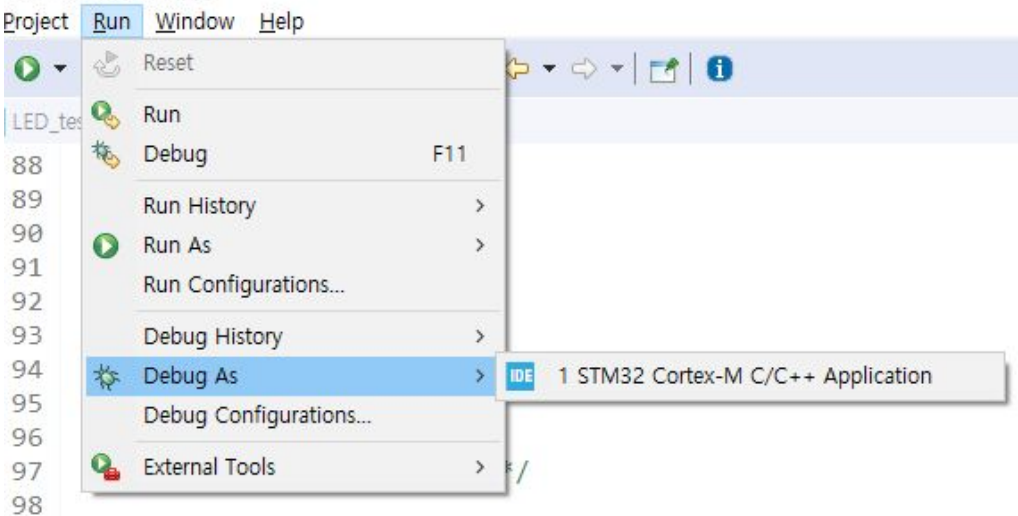


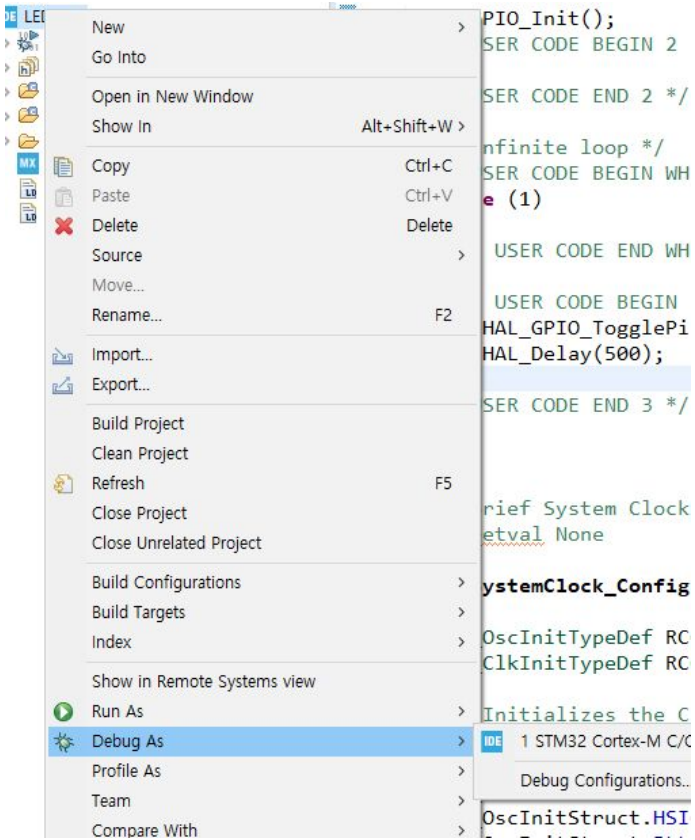


LED Blinky

Debug

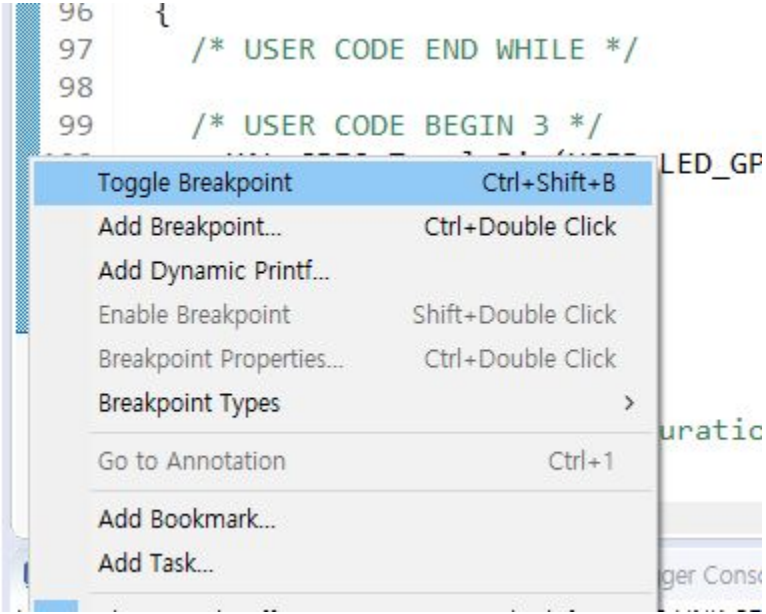


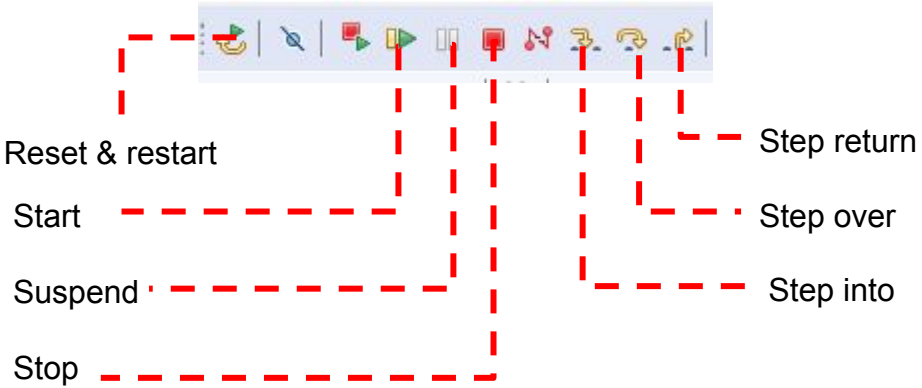




LED Blinky

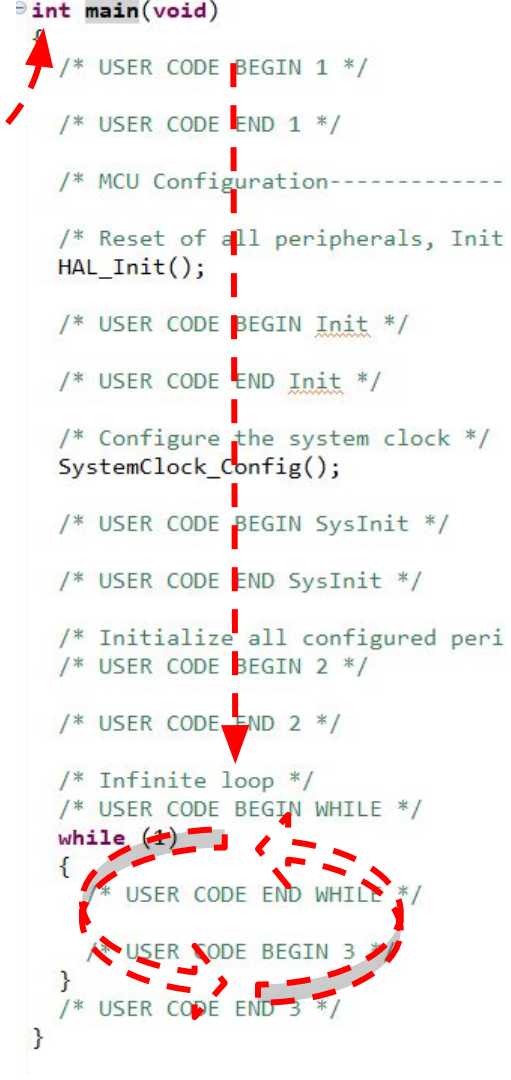
Debugging





Firmware 흐름(NO-RTOS)

START UP



```
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------

  /* Reset of all peripherals, Init
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peri
  /* USER CODE BEGIN 2 */

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */
}
```

실습

LED Blinky

- LED 2개를 연결하고 LED1 500ms, LED2 700ms로 설정

감사합니다.