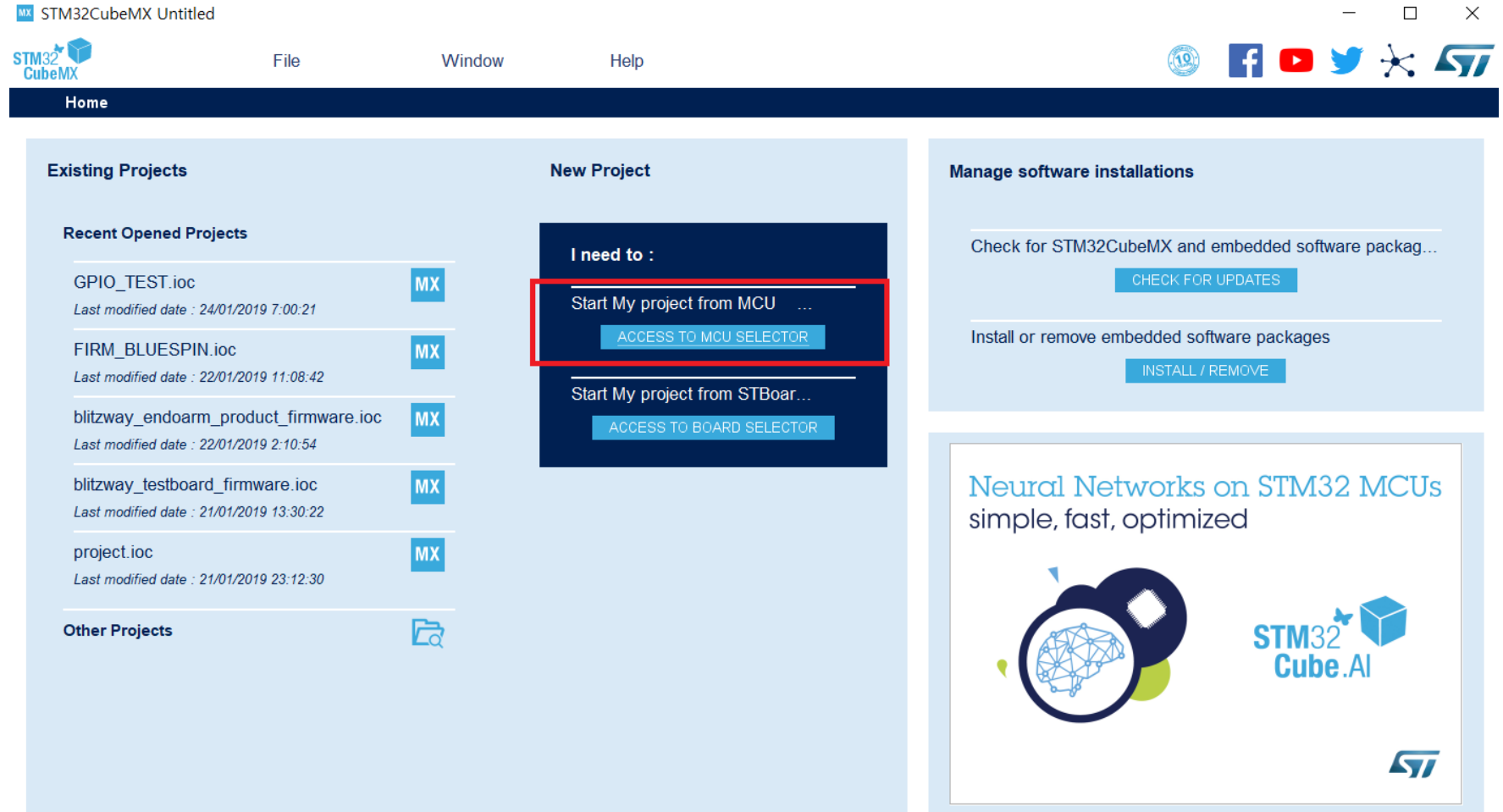


### 3. 개발환경 세팅

# 3. 개발환경 세팅

- STM32CubeMX

- STM32용 Base 코드 자동 생성 툴



### 3. 개발환경 세팅

- STM32CubeMX

- STM32용 Base 코드 자동 생성 툴

STM32F446RE

High-performance foundation line, ARM Cortex-M4 core with DSP and FPU, 512 Kbytes Flash, 180 MHz CPU, ART Accelerator, Dual QSPI

**ACTIVE** Active  
Product is in mass production

Unit Price for 10kU (US\$) : 4.374

Board: NUCLEO-F446RE

LQFP64

The STM32F446x/E devices are based on the high-performance ARM®Cortex®-M4 32-bit RISC core operating at a frequency of up to 180 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 STM32F446x/E devices incorporate high-speed embedded memories (Flash memory up to 512 Kbyte, up to 128 Kbyte of SRAM), up to 4 Kbytes of backup SRAM, and an extensive range of enhanced I/Os and peripherals connected to two APB buses, two AHB buses and a 32-bit multi-AHB bus matrix.

All devices offer three 12-bit ADCs, two DACs, a low-power RTC, twelve general-purpose 16-bit timers including two PWM timers for motor control, two general-purpose 32-bit timers.

They also feature standard and advanced communication interfaces.

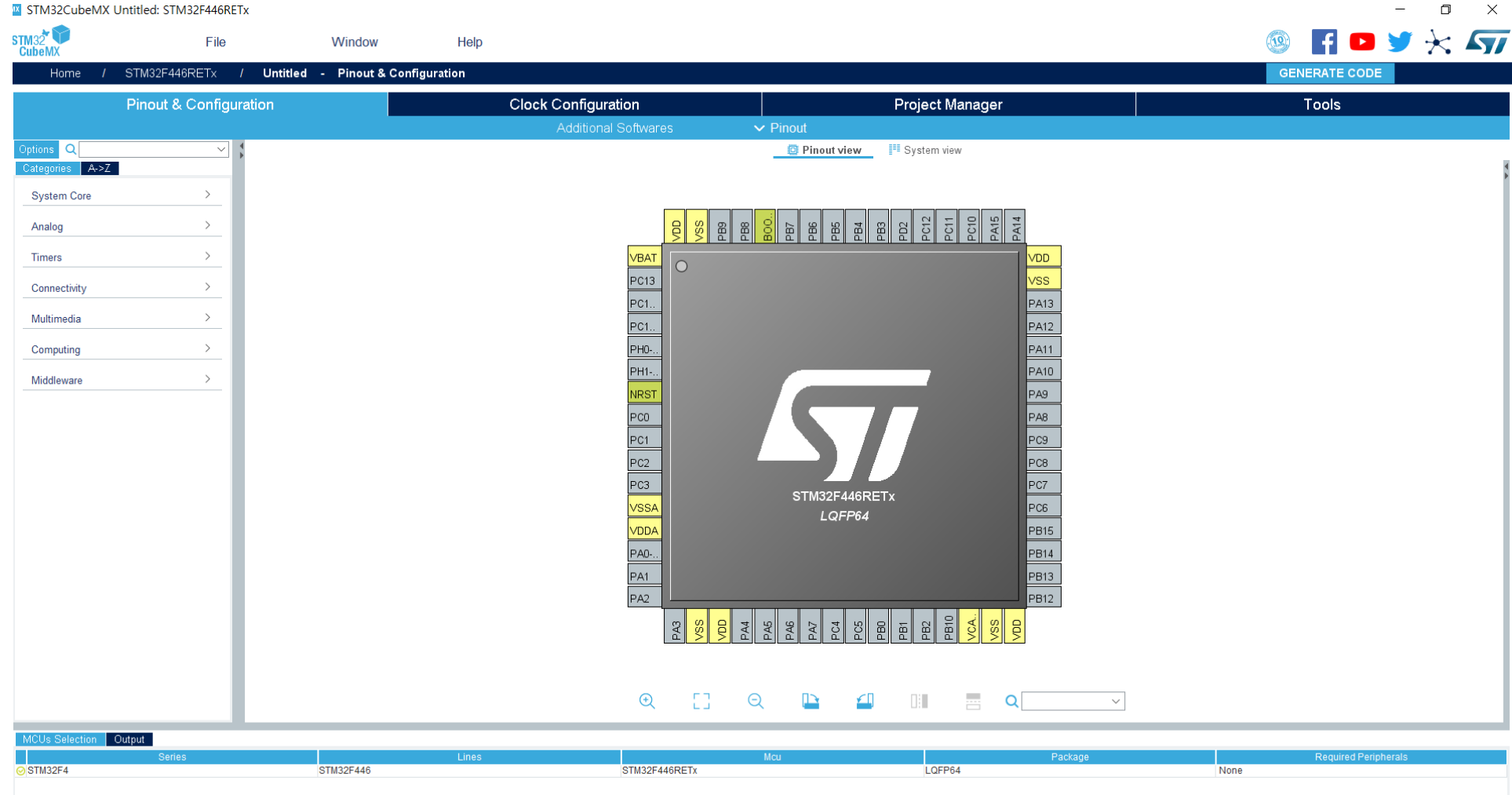
MCUs List: 1 item

* Part No	Reference	Marketing Stat.	Unit Price for 10kU (US\$)	Board	Package	Flash	RAM	IO	Freq.	GFX Score
STM32F446RE	STM32F446RETx	Active	4.374	NUCLEO-F446RE	LQFP64	512 kBytes	128 kBytes	50	180 MHz	0.0

# 3. 개발환경 세팅

- STM32CubeMX

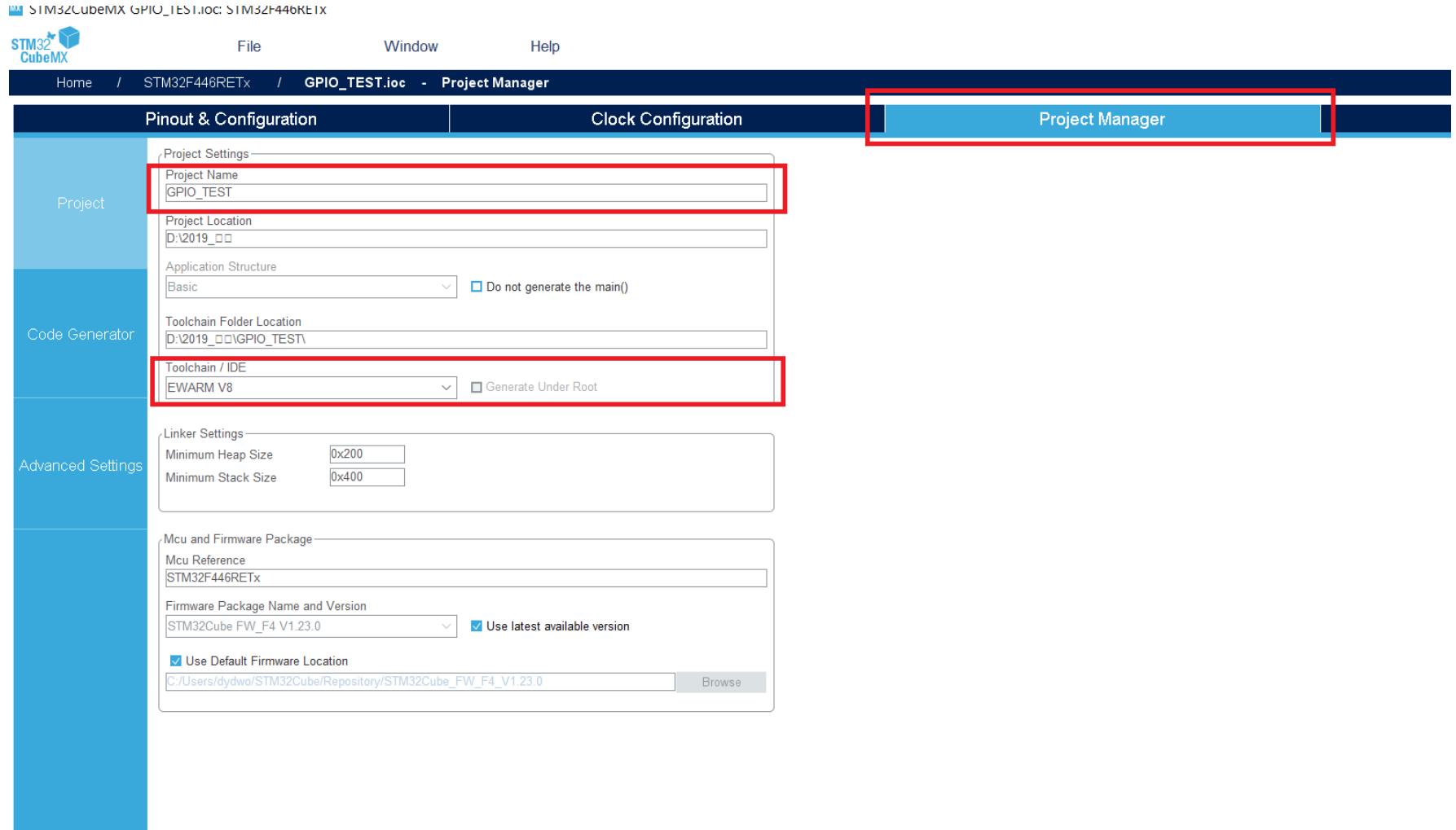
- STM32용 Base 코드 자동 생성 툴



# 3. 개발환경 세팅

- STM32CubeMX

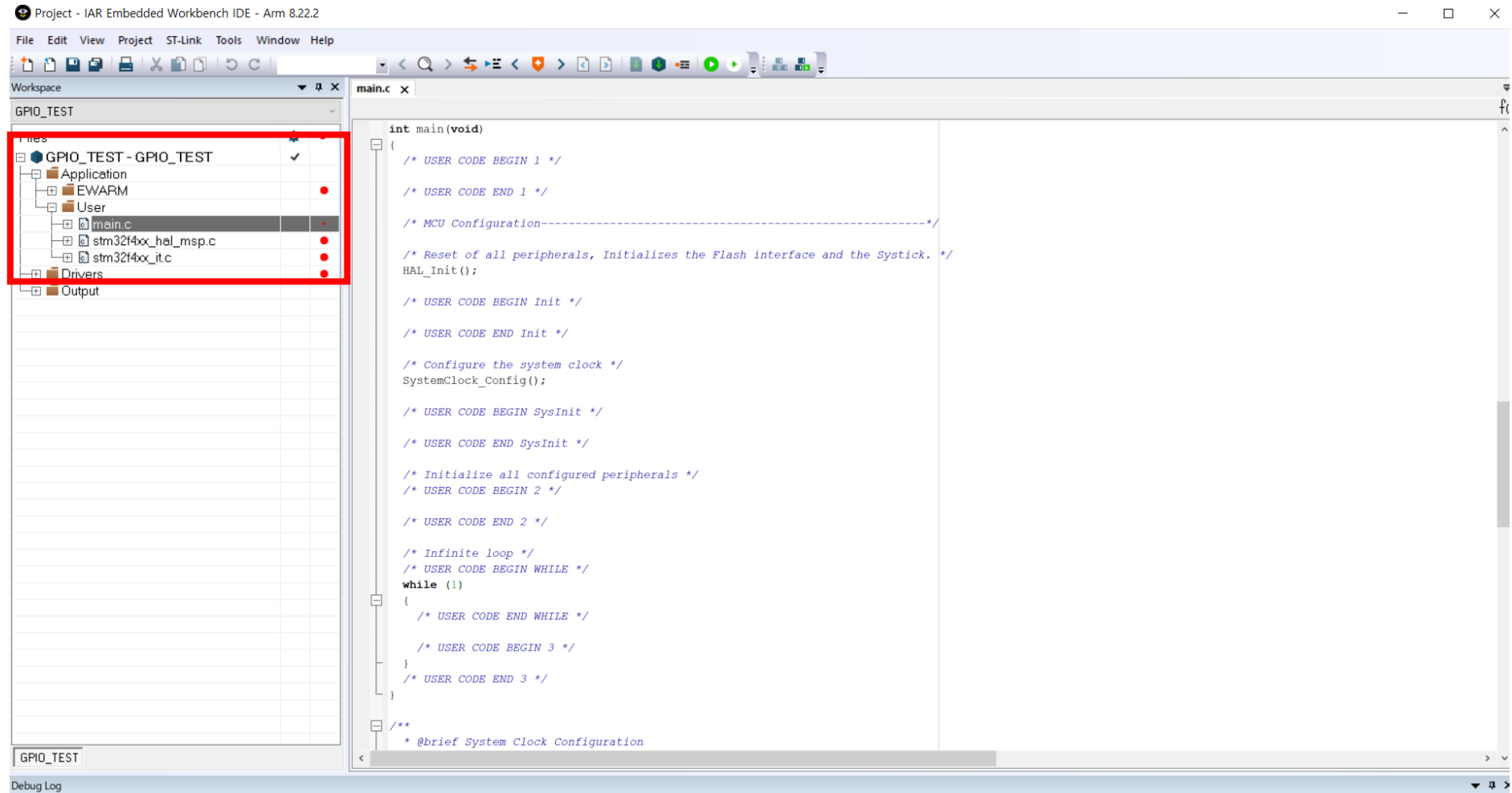
- STM32용 Base 코드 자동 생성 툴



# 3. 개발환경 세팅

- STM32CubeMX

- STM32용 Base 코드 자동 생성 툴



### 3. 개발환경 세팅

- STM32CubeMX

- 사용자의 코드는 **/\*USER**

**CODE\*/** 주석 사이에 작성해야 한다.

- 추후 설정 변경시에, CubeMX  
는 이러한 주석 사이의 코드들  
은 변경하지 않은 채 나머지를  
수정한다.

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

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    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 peripherals */
    /* 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 */

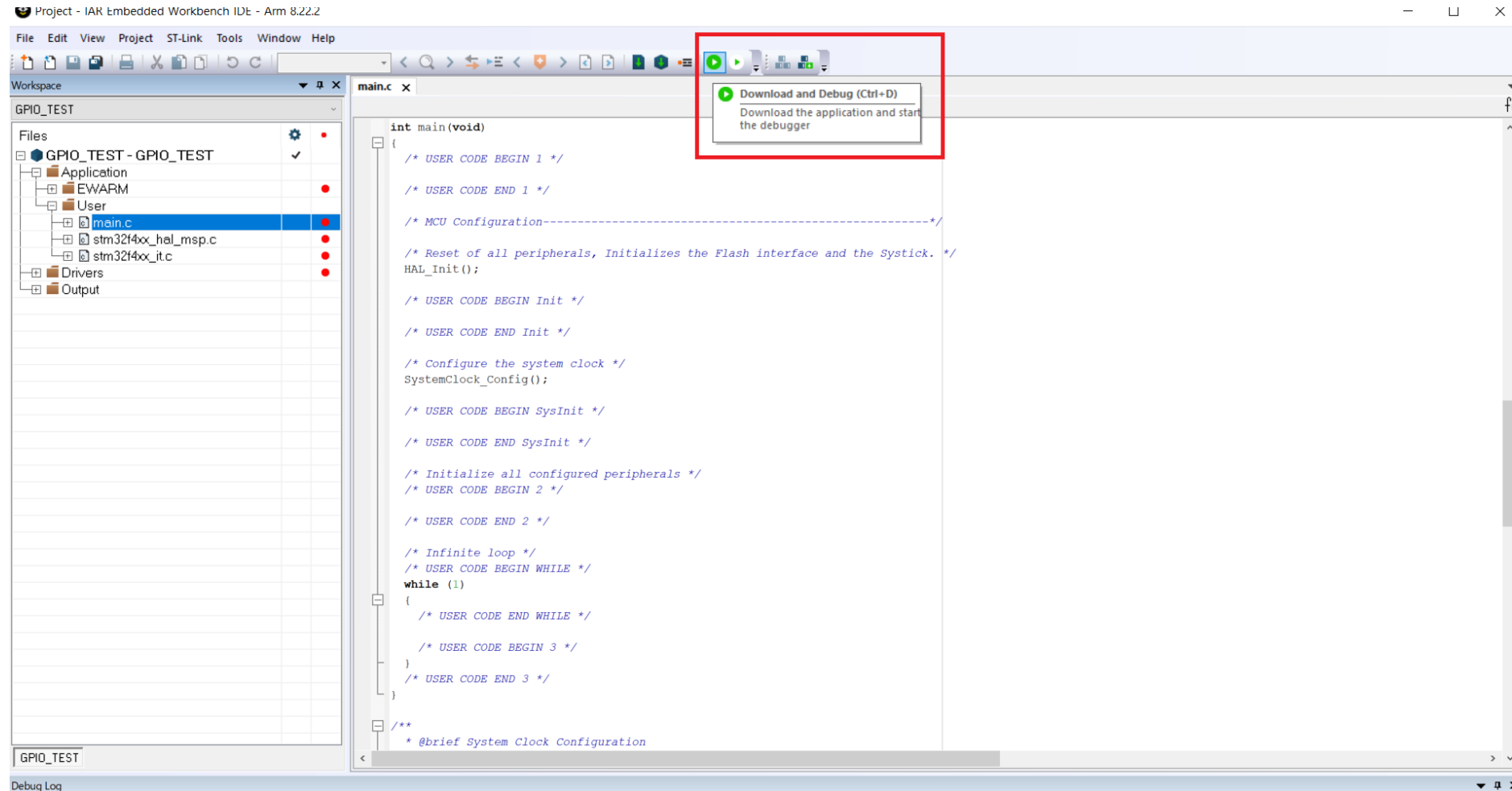
    /**
     * @brief System Clock Configuration

```

### 3. 개발환경 세팅

- STM32CubeMX

- Nucleo를 연결하고 Download and Debug를 눌러 업로드 확인





## 4. GPIO

## 4. GPIO

- GPIO(General Purpose Input Output)

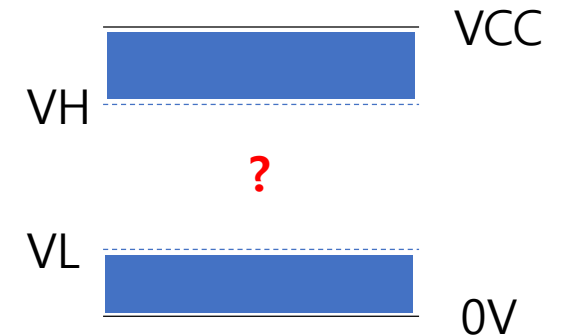
- 가장 기본적인 범용 입/출력 기능
- 아두이노의 **digitalWrite**, **digitalRead**에 대응되는 기능

### [Output Mode]

- 특정 핀으로 High Voltage(3.3V) 또는 Low Voltage(0V)를 출력한다.

### [Input Mode]

- 특정 핀으로 들어오는 전압이 High Voltage인지, 또는 Low Voltage인지 알아낸다.
- High Voltage와 Low Voltage는 특정 Threshold를 기준으로 한다.



## 4. GPIO

- GPIO(General Purpose Input Output)

Table 56. I/O static characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IL}$	FT, FTf, TTa and NRST I/O input low level voltage	$1.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	-	-	$0.35V_{DD}-0.04^{(1)}$ $0.3V_{DD}^{(2)}$	V
	BOOT0 I/O input low level voltage	$1.75\text{ V} \leq V_{DD} \leq 3.6\text{ V}$ , $-40\text{ }^{\circ}\text{C} \leq T_A \leq 105\text{ }^{\circ}\text{C}$	-	-	$0.1V_{DD}+0.1^{(1)}$	
		$1.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$ , $0\text{ }^{\circ}\text{C} \leq T_A \leq 105\text{ }^{\circ}\text{C}$	-	-		
$V_{IH}$	FT, FTf, TTa and NRST I/O input high level voltage <sup>(4)</sup>	$1.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	$0.45V_{DD}+0.3^{(1)}$ $0.7V_{DD}^{(2)}$	-	-	V
	BOOT0 I/O input high level voltage	$1.75\text{ V} \leq V_{DD} \leq 3.6\text{ V}$ , $-40\text{ }^{\circ}\text{C} \leq T_A \leq 105\text{ }^{\circ}\text{C}$	$0.17V_{DD}+0.7^{(1)}$	-	-	
		$1.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$ , $0\text{ }^{\circ}\text{C} \leq T_A \leq 105\text{ }^{\circ}\text{C}$				
$V_{HYS}$	FT, FTf, TTa and NRST I/O input hysteresis	$1.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	-	$10\%V_{DD}$	-	V
	BOOT0 I/O input hysteresis	$1.75\text{ V} \leq V_{DD} \leq 3.6\text{ V}$ , $-40\text{ }^{\circ}\text{C} \leq T_A \leq 105\text{ }^{\circ}\text{C}$	-	100m	-	
		$1.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$ , $0\text{ }^{\circ}\text{C} \leq T_A \leq 105\text{ }^{\circ}\text{C}$	-		-	
$I_{lkg}$	I/O input leakage current <sup>(3)</sup>	$V_{SS} \leq V_{IN} \leq V_{DD}$	-	-	$\pm 1$	$\mu\text{A}$
	I/O FT input leakage current <sup>(4)</sup>	$V_{IN} = 5\text{ V}$	-	-	3	

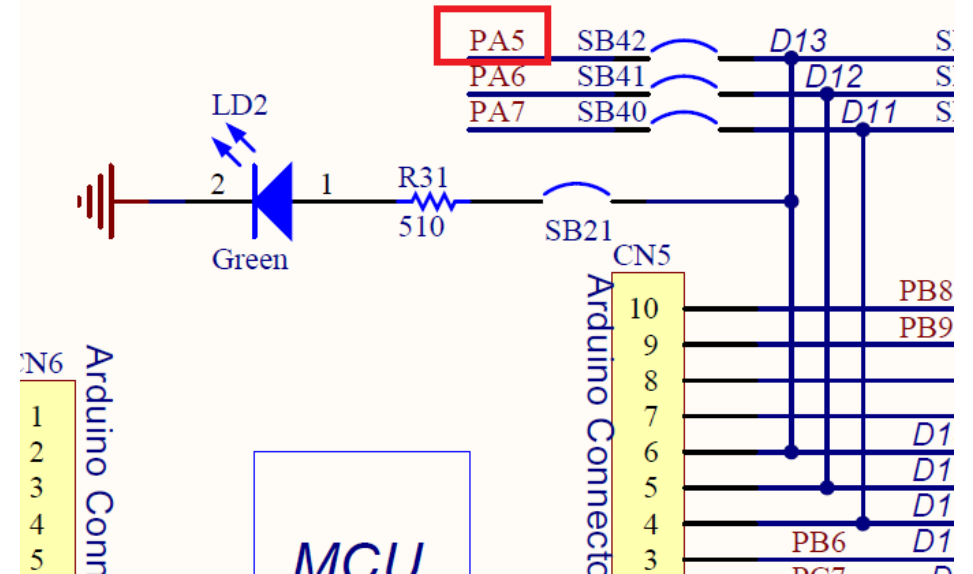
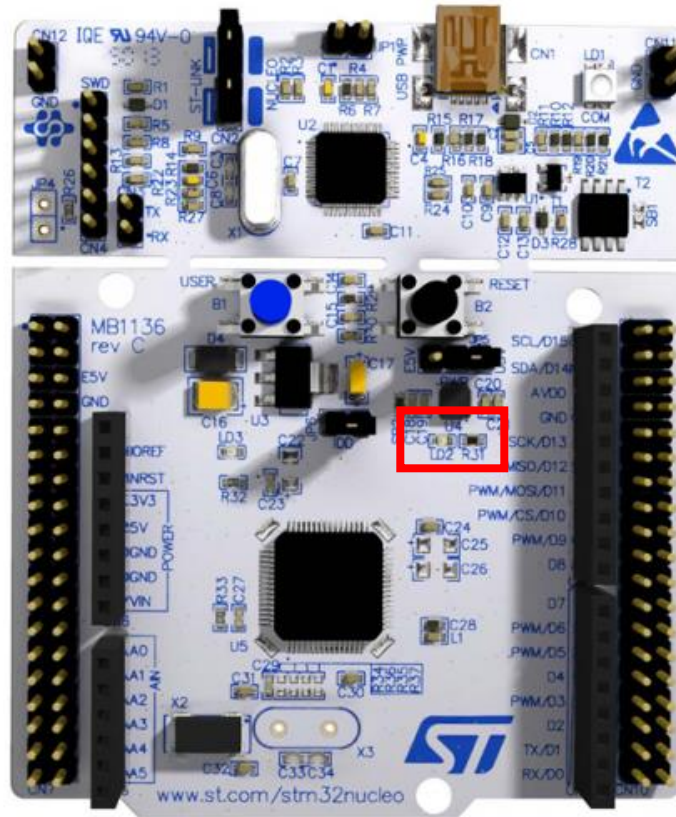
1. Guaranteed by design.

2. Tested in production.

## 4. GPIO

- GPIO Output

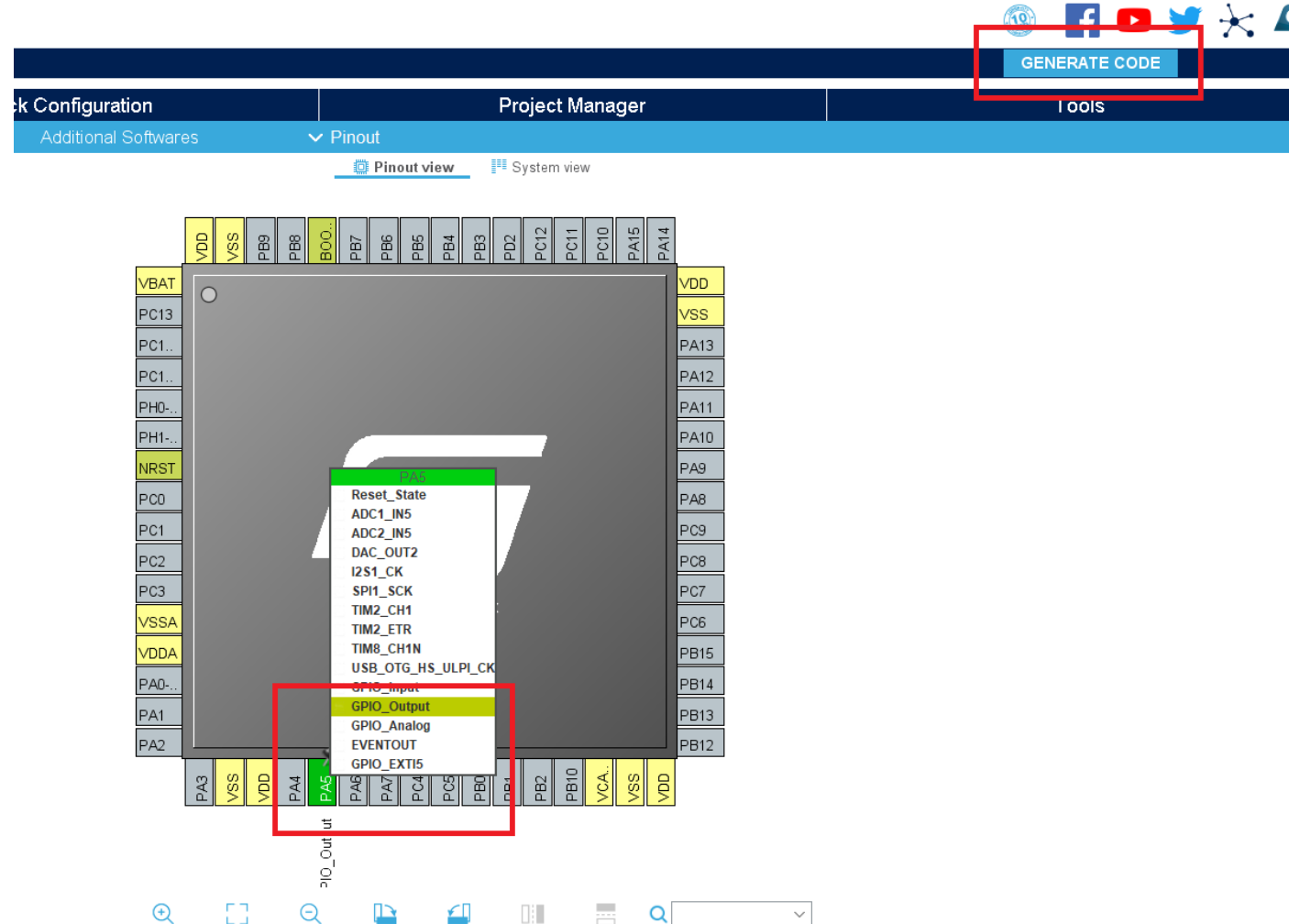
- 누클레오 보드 중앙에 있는 LD2 Led는 내부적으로 칩의 PA5 핀에 연결되어 있다.



## 4. GPIO

- GPIO Output

- CubeMX에서 PA5에 GPIO\_Output을 지정해 주고 "GENERATE CODE"를 통해 코드를 재생성 한다.



## 4. GPIO

- GPIO Output

- **HAL\_GPIO\_WritePin**(GPIOx, PIO\_Pin, PinState)

[LED 점등]

```
/* USER CODE BEGIN SysInit */  
  
/* USER CODE END SysInit */  
  
/* Initialize all configured peripherals */  
MX_GPIO_Init();  
/* USER CODE BEGIN 2 */  
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);  
/* USER CODE END 2 */  
  
/* Infinite loop */  
/* USER CODE BEGIN WHILE */  
while (1)
```

## 4. GPIO

- GPIO Output

- HAL\_GPIO\_TogglePin(GPIOx, PIO\_Pin)
- HAL\_Delay(ms)

[LED 점멸]

```
/* USER CODE END 2 */

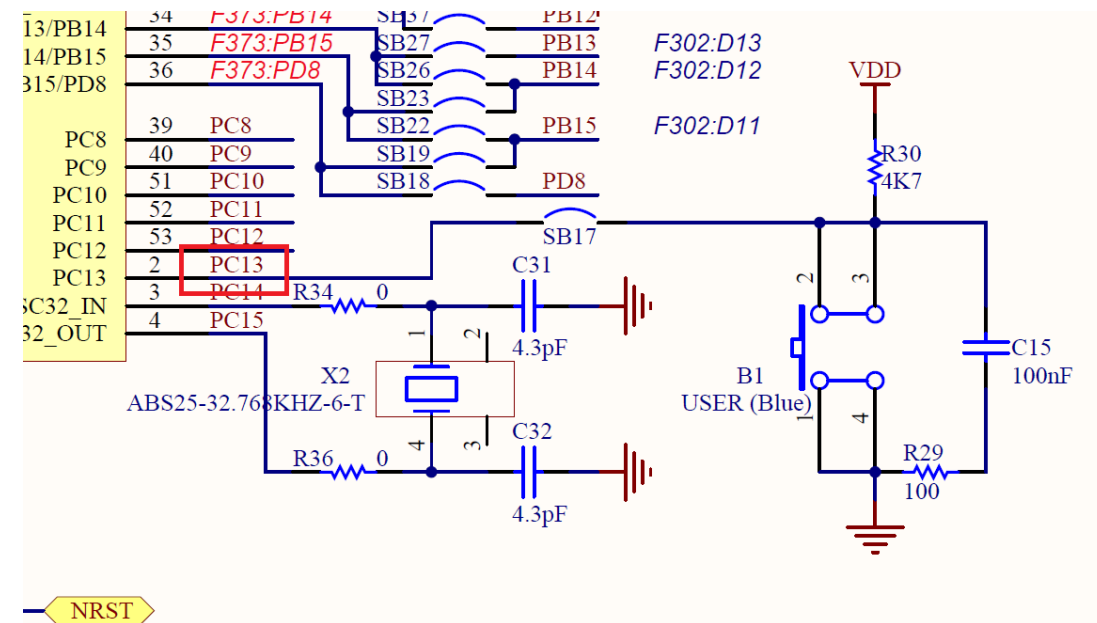
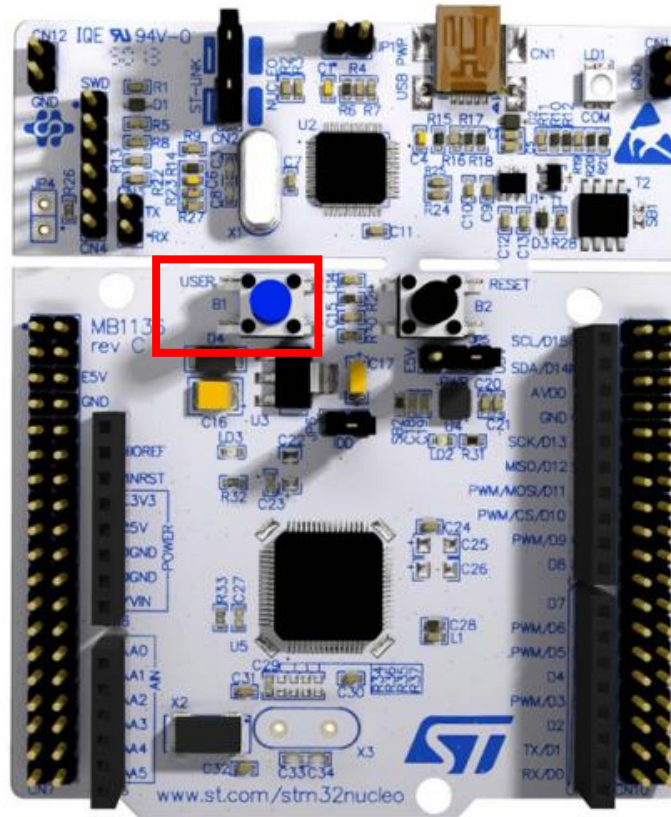
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_4);
    HAL_Delay(500);
    /* USER CODE END WHILE */

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

## 4. GPIO

- GPIO Input

- 누클레오 보드 중앙에 있는 USER 버튼(파란색)은 내부적으로 칩의 PC13 핀에 연결되어 있다.

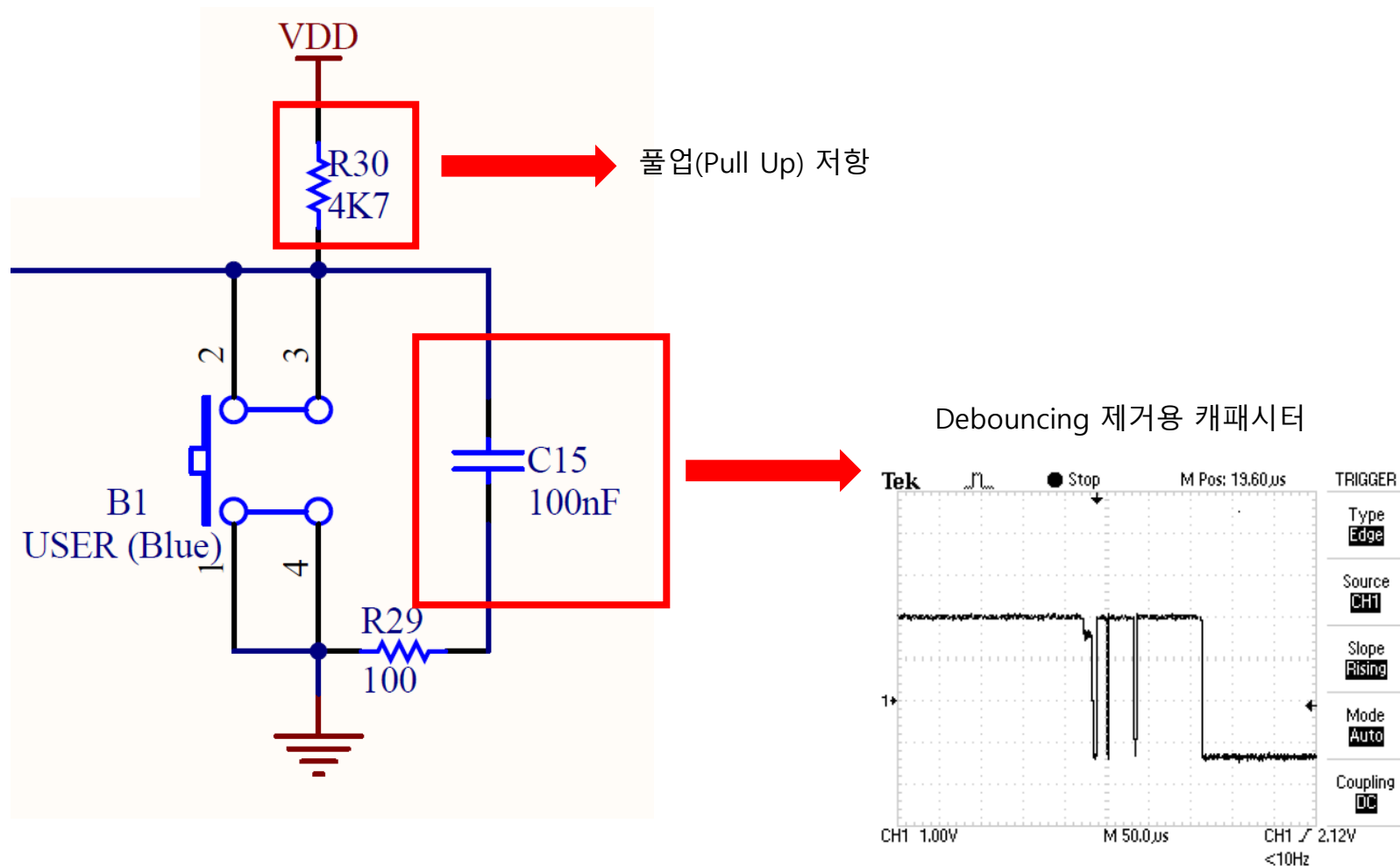




## 4. GPIO

- GPIO Input

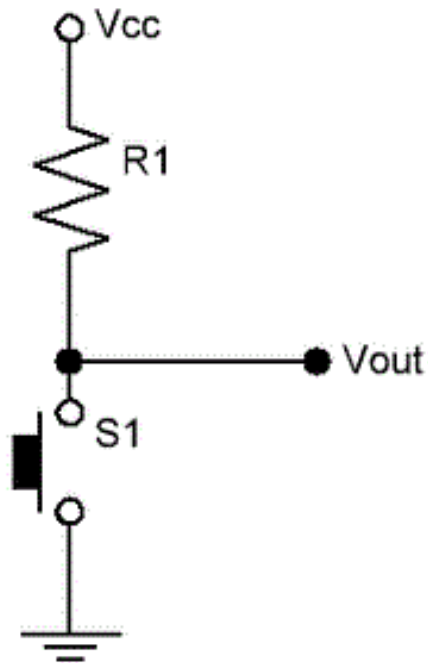
- 누클레오 보드 중앙에 있는 USER 버튼(파란색)은 내부적으로 칩의 PC13 핀에 연결되어 있다.



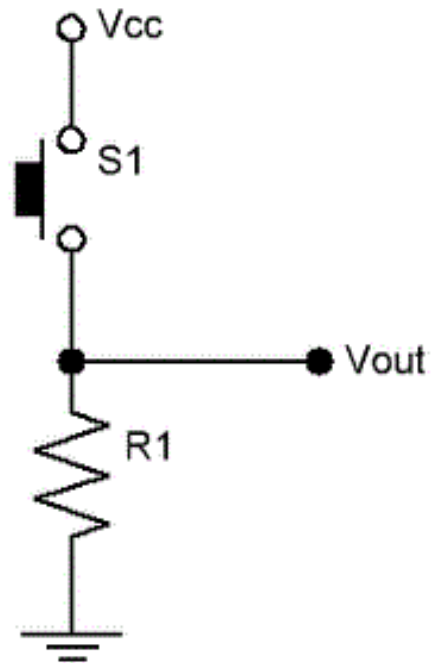
## 4. GPIO

- GPIO Input

- Pull Up과 Pull Down은 핀이 Floating 상태에 있는 것을 막아준다.



Pull Up

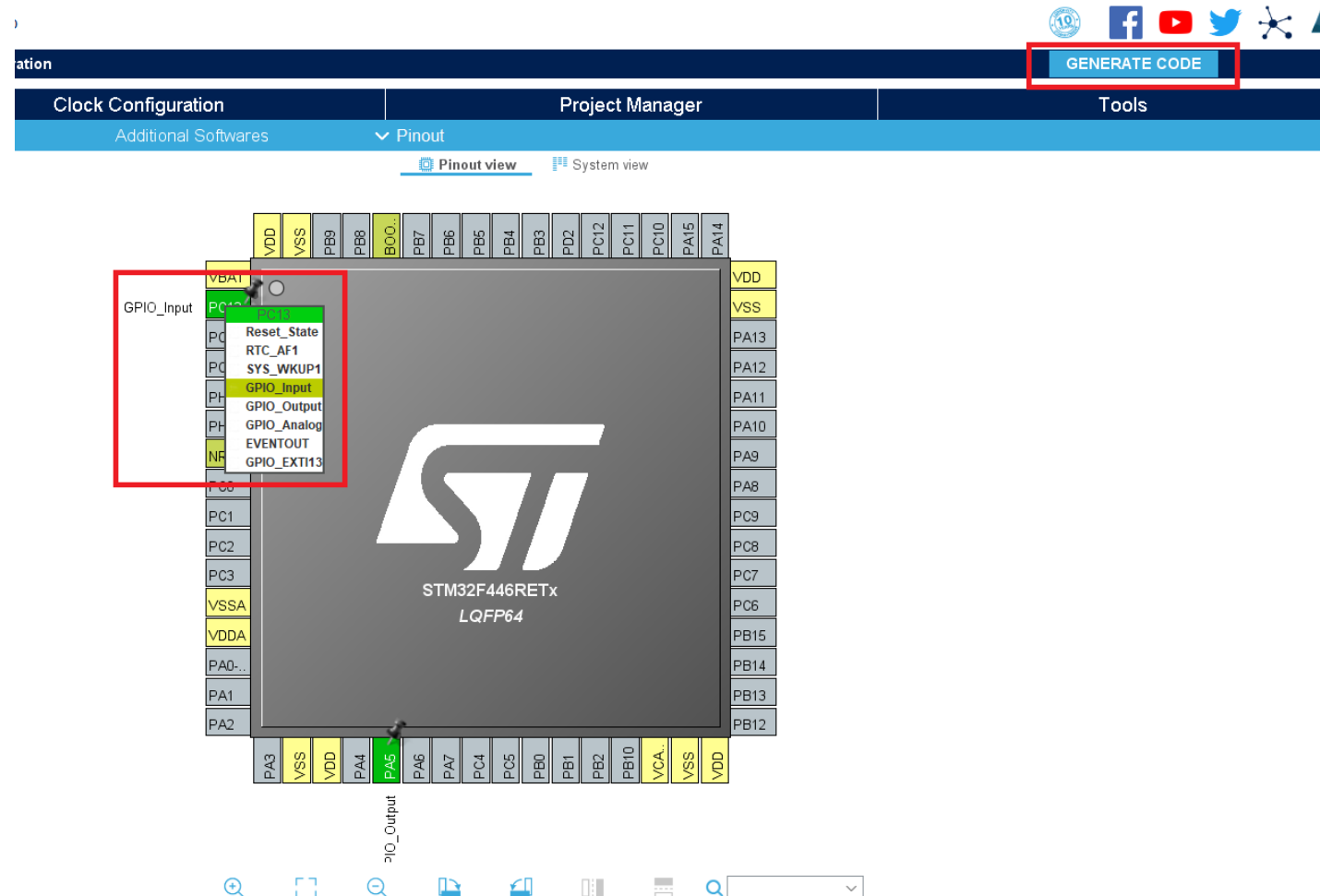


Pull Down

## 4. GPIO

- GPIO Input

- CubeMX에서 PC13을 GPIO\_Input으로 지정해 주고 "GENERATE CODE"를 통해 코드를 재생성 한다.



## 4. GPIO

- GPIO Input

- **HAL\_GPIO\_ReadPin**(GPIOx, PIO\_Pin)

[버튼을 이용한LED 점멸]

```
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    if(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET){
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
    }else{
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
    }

    HAL_Delay(10);
/* USER CODE END WHILE */

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

## 5. Interrupt

## 5. Interrupt



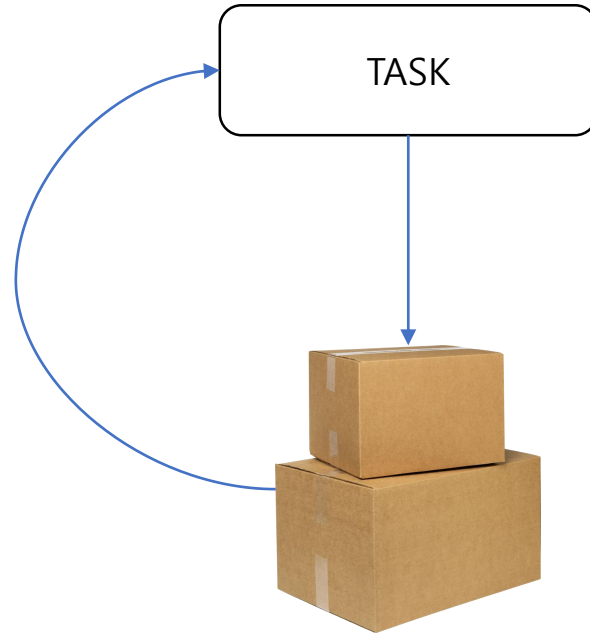
- 나는 지금 너무 바쁘는데 지난주에 시킨 택배가 언제 올지 모른다.
- 택배가 왔는지 아닌지 확인하기 위하여 어떻게 해야 할까?

## 5. Interrupt



- 나는 지금 너무 바쁘데 지난주에 시킨 택배가 언제 올지 모른다.
- 택배가 왔는지 아닌지 확인하기 위하여 어떻게 해야 할까?

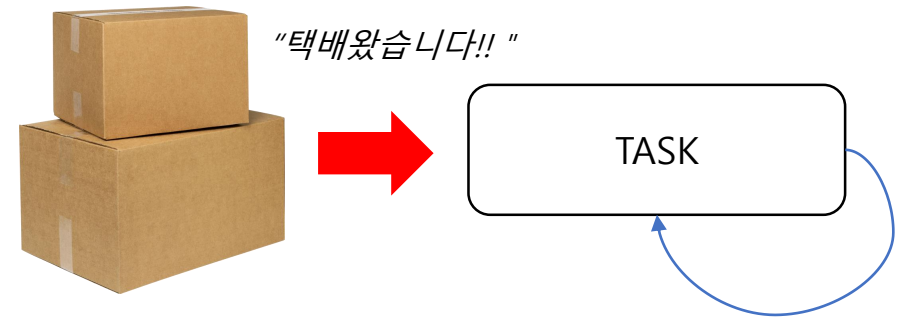
## 5. Interrupt



[Polling]

MCU에선 누구에게 이 사실을 알릴까?

"아저씨 택배 도착하면 010-XXXX-XXXX로 문자남겨주세요 "



[Interrupt]



# 5. Interrupt

- NVIC(Nested Vecter Interrupt Controller)

- 여러 종류의 인터럽트를 NVIC에 등록하면 NVIC가 인터럽트를 관리하기 시작한다.
- 인터럽트의 우선순위를 통해 여러 인터럽트의 순서를 정의한다.

