

# **TI DSP, MCU, Xilinx Zynq FPGA**

## **프로그래밍 전문가 과정**

**GPIO based on STM32F407**

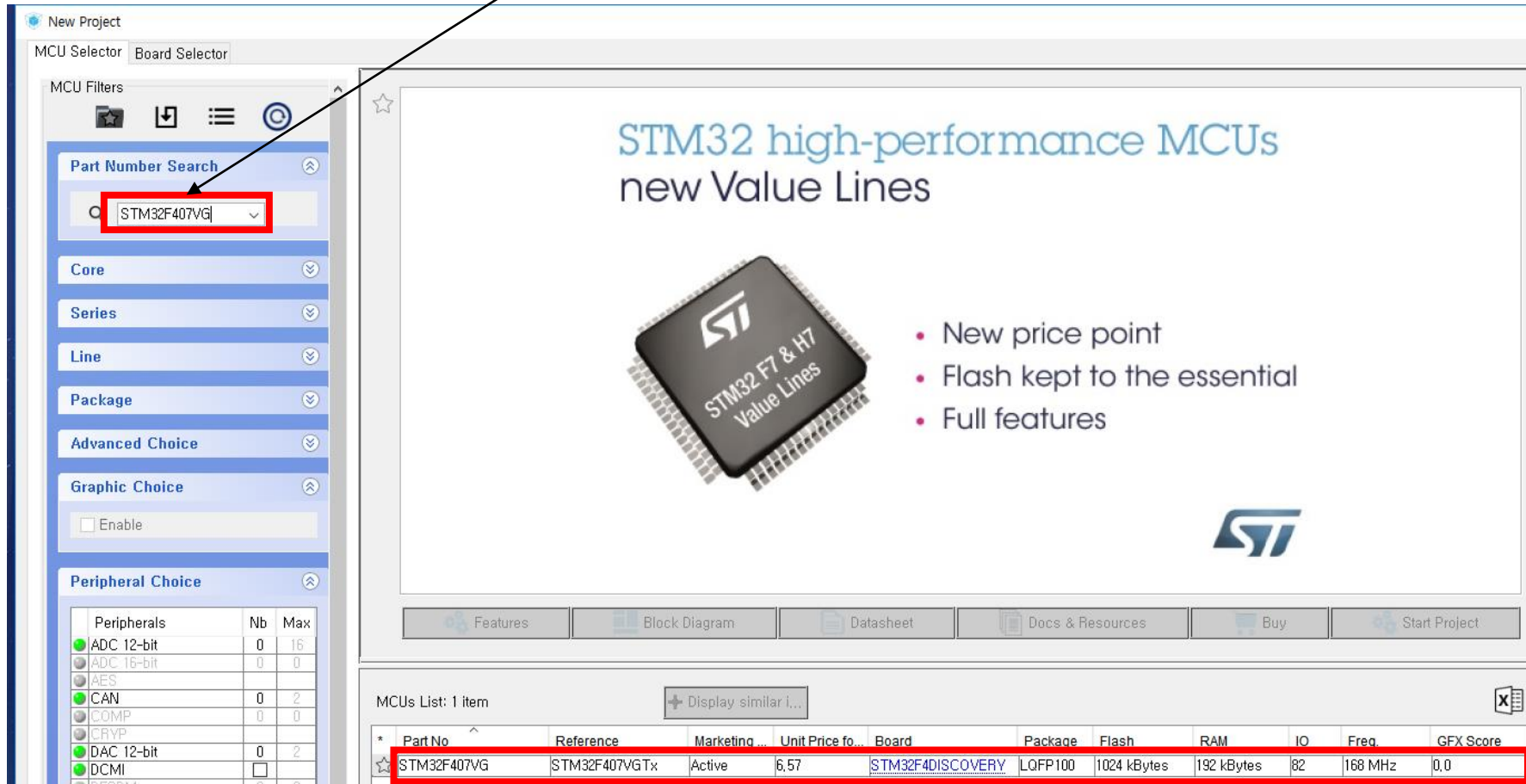
**강사 – Innova Lee(이상훈)**  
gcccompil3r@gmail.com

**학생 – 안상재**  
sangjae2015@naver.com

# 1. LED 깜빡이기 (GPIO OUTPUT)

- CubeMX 설정

필자는 **STM32F407VGT6** 이라는 MCU가 탑재된 보드(STM32F4-DISC1)를 사용함



더블클릭으로 프로젝트 생성

STM32CubeMX GPIO\_TEST.ioc: STM32F407VGTx

File Project Pinout Window Help

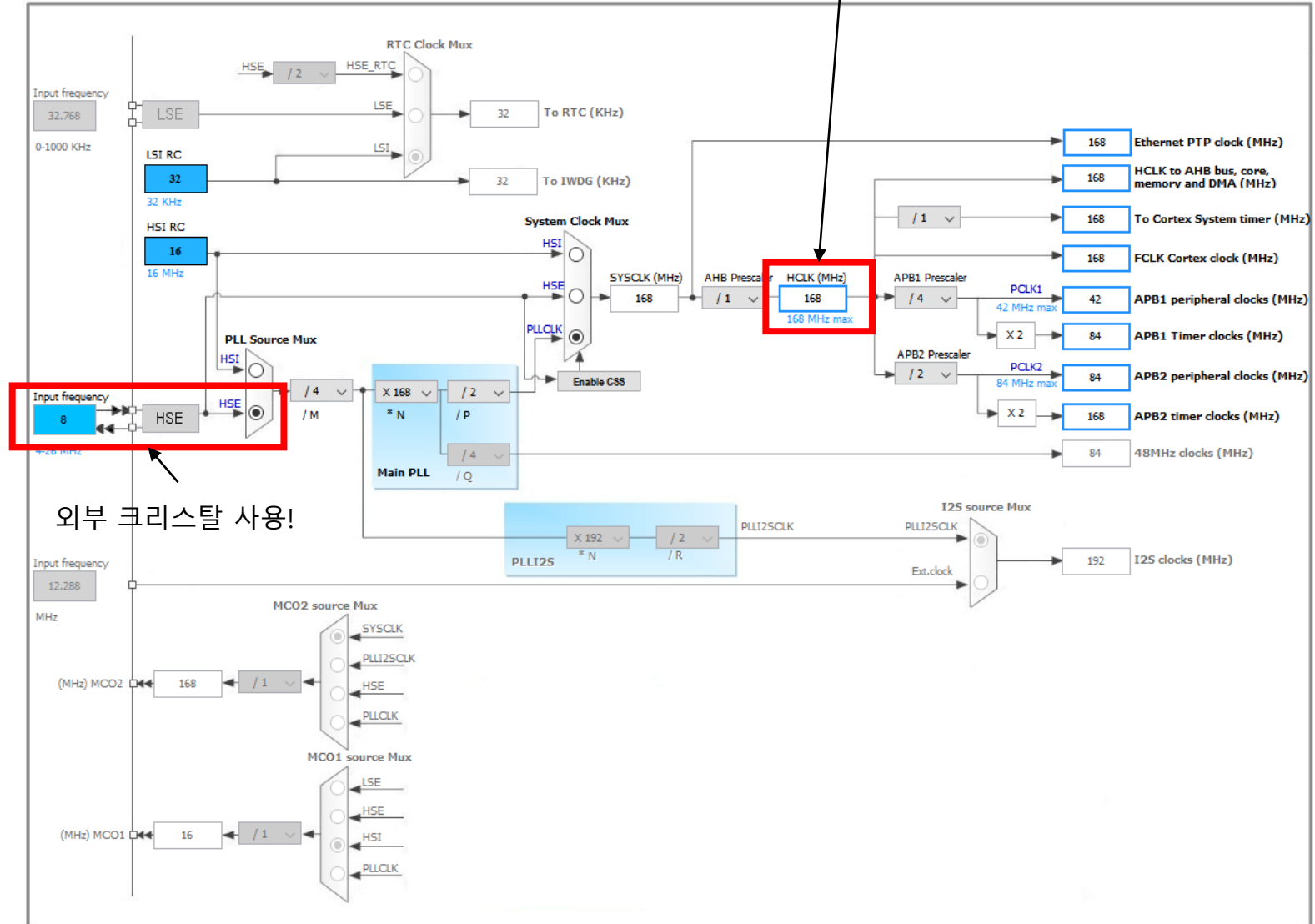
Keep Current Signals Placement

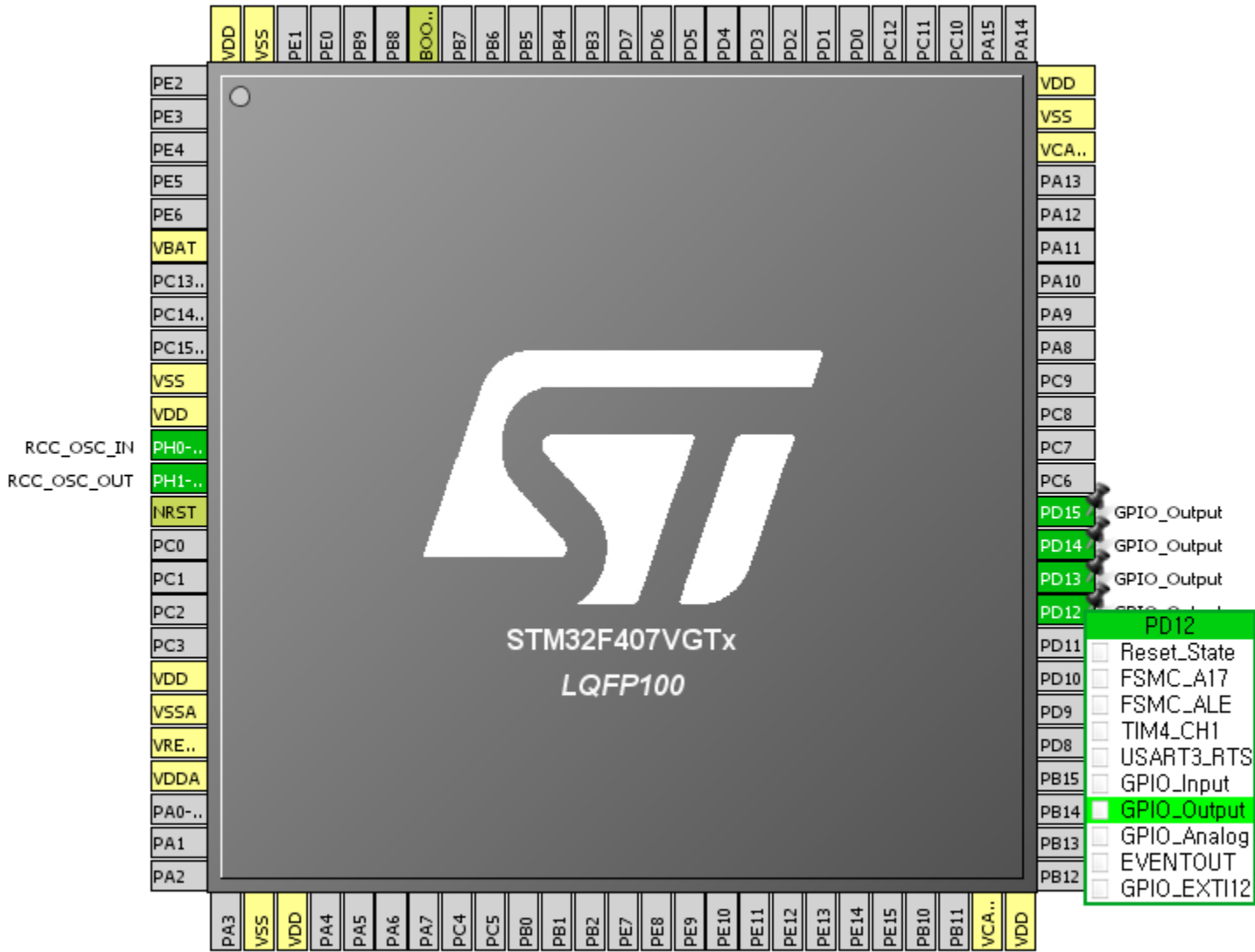
**Pinout** Clock Configuration Configuration Power Consumption Calculator

- CAN2
- CRC
- DAC
- DCMI
- ETH
- FSMC
- I2C1
- I2C2
- I2C3
- I2S2
- I2S3
- IWDG
- RCC**
  - High Speed Clock (HSE) Crystal/Ceramic Resonator
  - Low Speed Clock (LSE) Disable
  - Master Clock Output BYPASS Clock Source
  - Master Clock Output Crystal/Ceramic Resonator
  - Audio Clock Input (I2S\_CKIN)
- RNG
- RTC
- SDIO
- SPI1
- SPI2
- SPI3
- SYS**
- TIM1
- TIM2
- TIM3
- TIM4
- TIM5
- TIM6
- TIM7
- TIM8
- TIM9

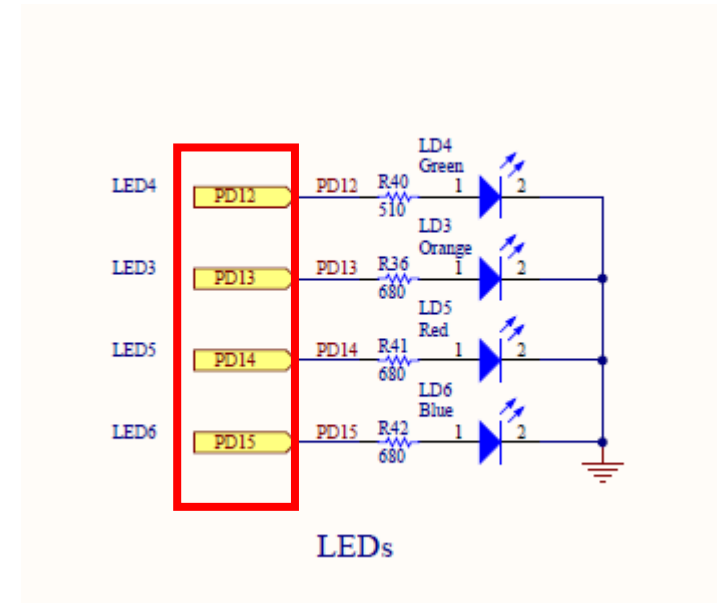
MCU가 허용하는 최대 주파수로 맞춰줌!

## Clock Configuration 탭

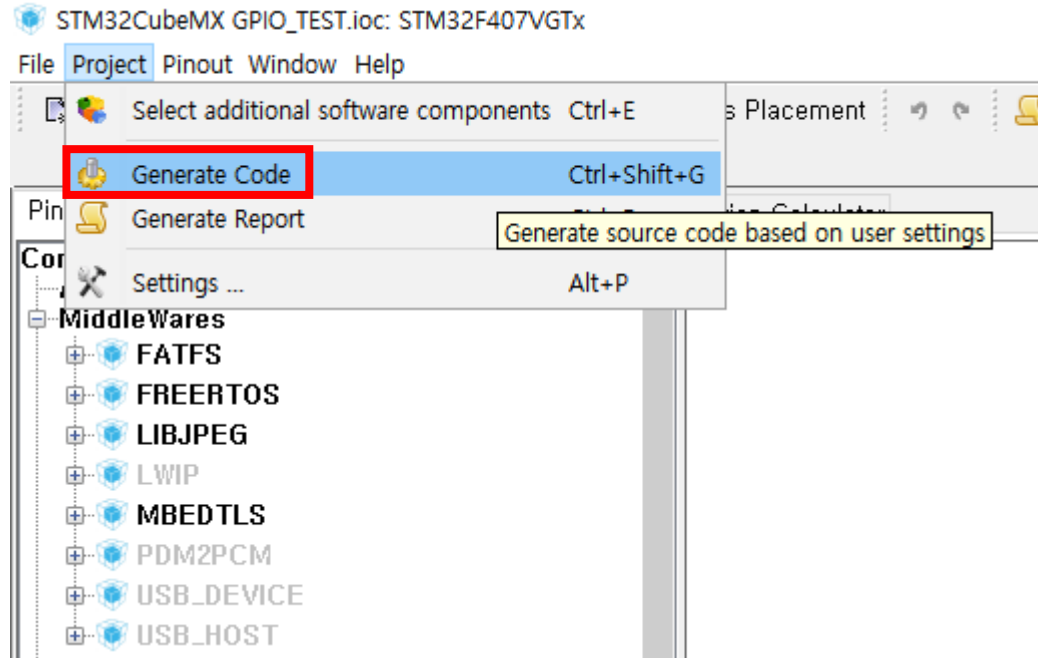




CubeMX Pinout



STM32F407-DISC1 보드 회로도



Generate Code 클릭 후, 프로젝트 생성

## - 소스 코드

**while (1)**

```
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_SET);

    HAL_Delay(1000); // milisecond 단위

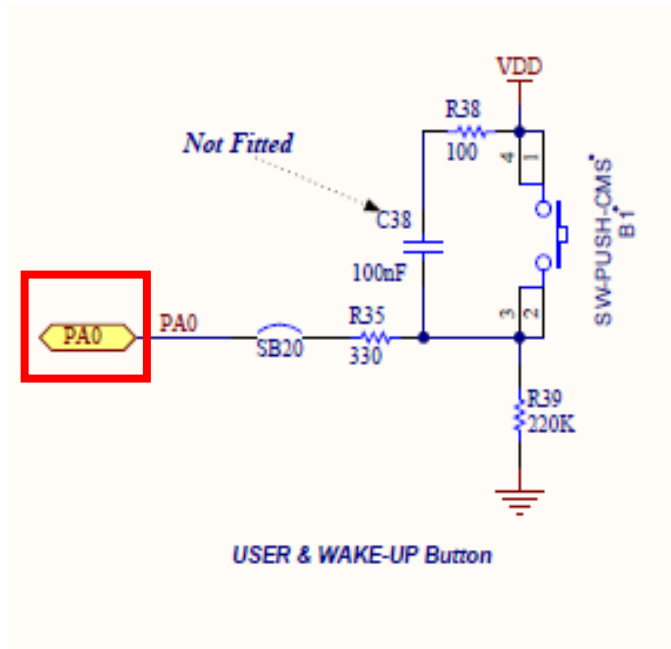
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET);

    HAL_Delay(1000); // milisecond 단위
    /* USER CODE BEGIN 3 */
}
```

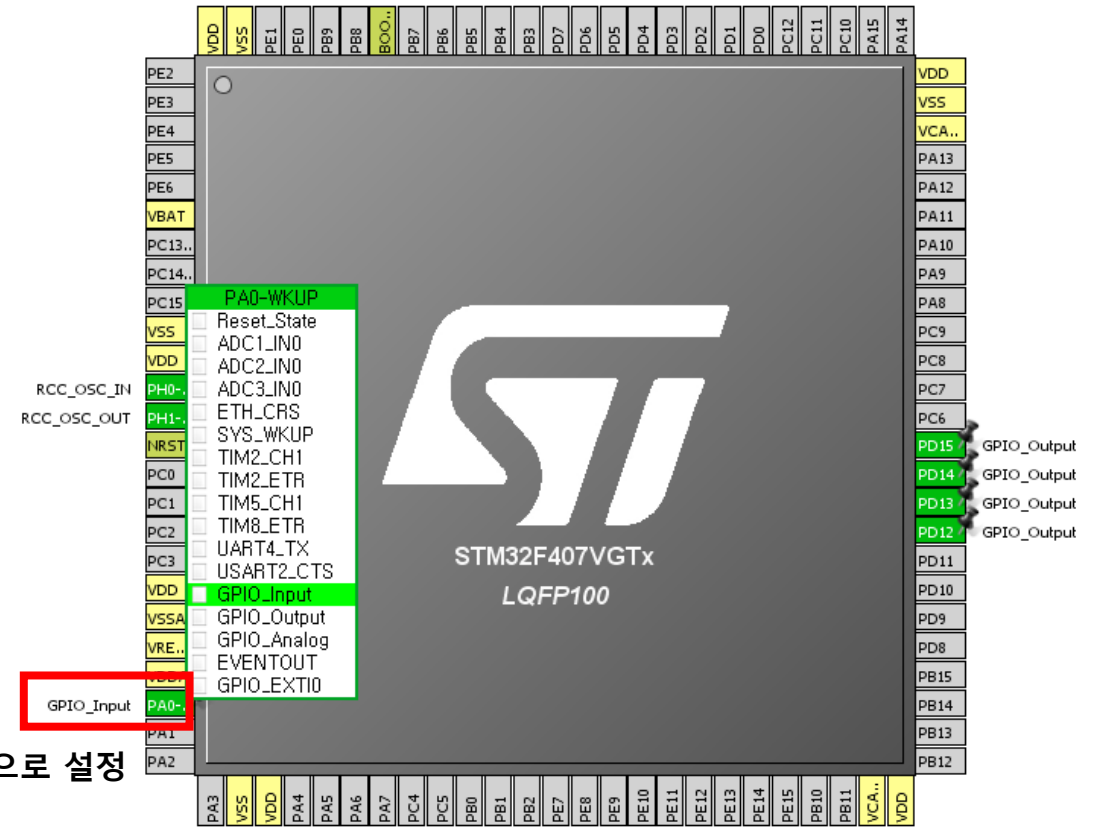
```
void HAL_GPIO_WritePin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin,
GPIO_PinState PinState)
{
    /* Check the parameters */
    assert_param(IS_GPIO_PIN(GPIO_Pin));
    assert_param(IS_GPIO_PIN_ACTION(PinState));

    if(PinState != GPIO_PIN_RESET)
    {
        GPIOx->BSRR = GPIO_Pin;
    }
    else
    {
        GPIOx->BSRR = (uint32_t)GPIO_Pin << 16U;
    }
}
```

## 2. 스위치 입력 (GPIO INPUT)



## 스위치 주변 회로



## CubeMX Pinout

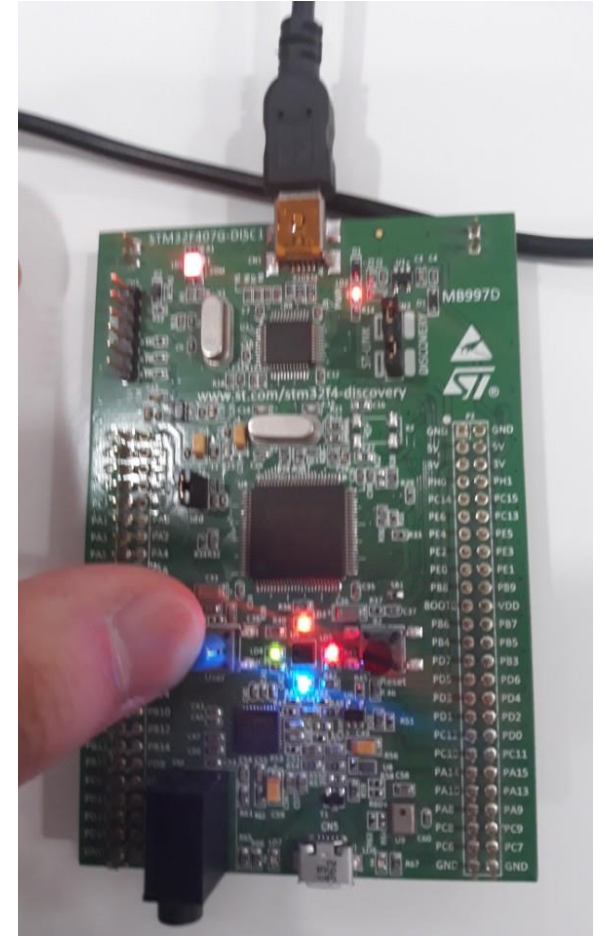
## GPIOA0 번 핀을 Input으로 설정

```

while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) == 1)  ← 스위치 입력 확인!
    {
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, SET);
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, SET);
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, SET);
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, SET);
    }
    else
    {
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, RESET);
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, RESET);
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, RESET);
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, RESET);
    }
}

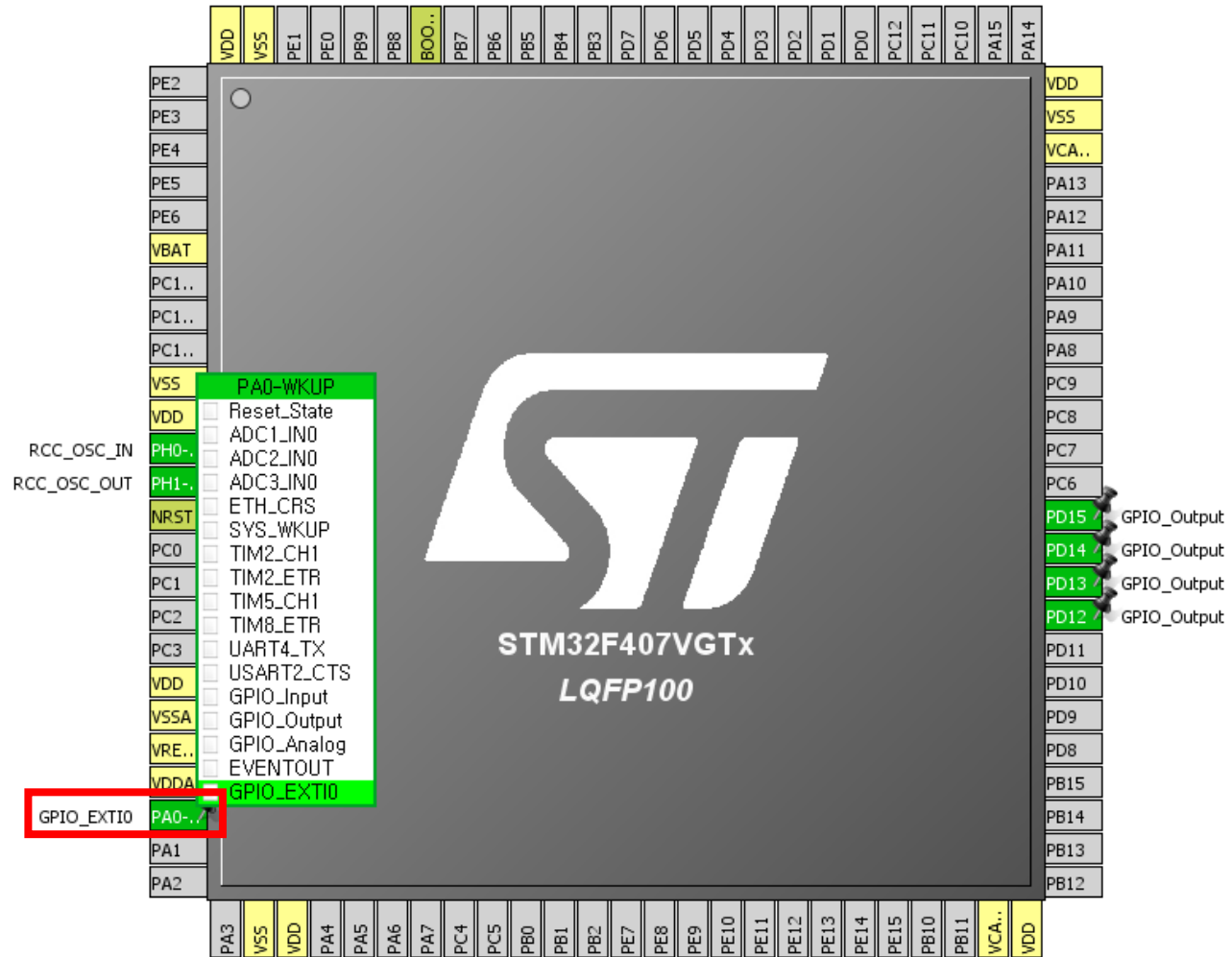
```



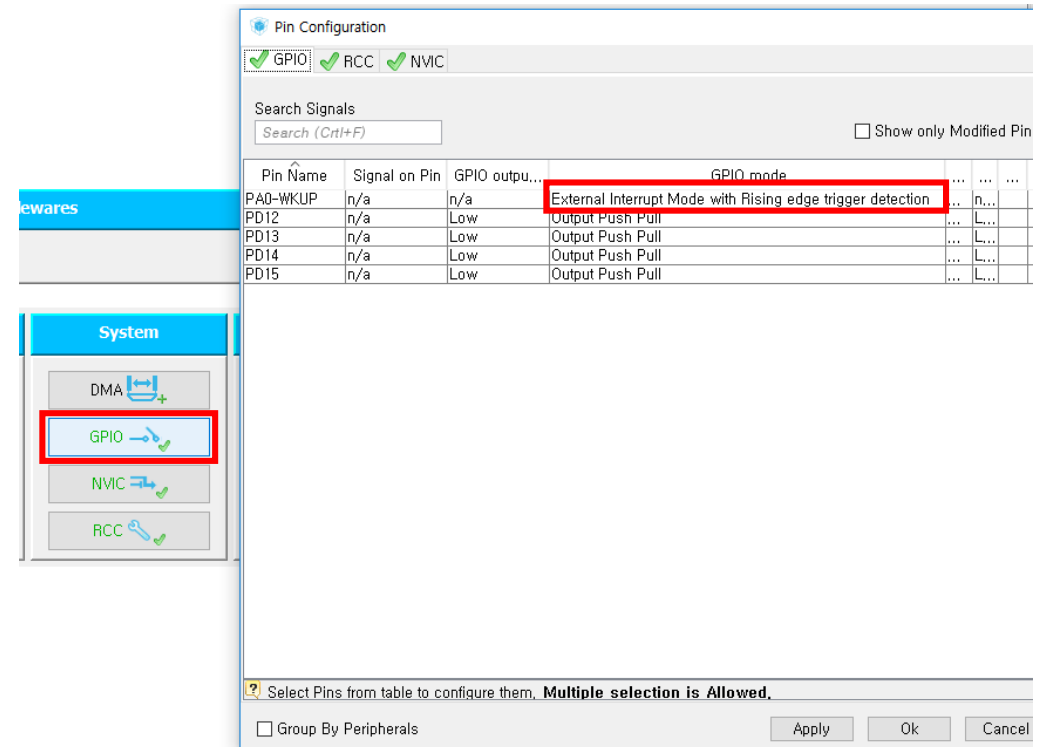
스위치 눌렀을 때 LED가 켜짐

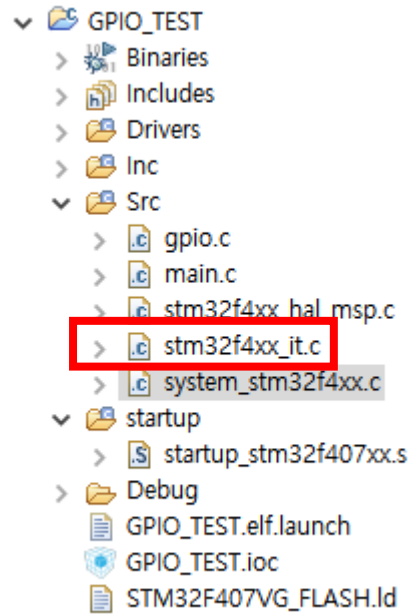


### 3. 외부 인터럽트 (External Interrupt)



인터럽트가 설정된 것을 확인!





stm32f4xx\_it.c 파일에 EXTI0에 대한 인터럽트 서비스 루틴이 생성됨!

```
void EXTI0_IRQHandler(void)
{
    /* USER CODE BEGIN EXTI0_IRQn 0 */

    /* USER CODE END EXTI0_IRQn 0 */

    HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_12);
    HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_13);
    HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_14);
    HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_15);

    HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);

    /* USER CODE BEGIN EXTI0_IRQn 1 */

    /* USER CODE END EXTI0_IRQn 1 */
}
```