

# 알람시계 타이머



## 아두이노 I2C



아두이노의 내부 타이머를  
직접 사용해 타이머 구현

아두이노의 내부 타이머를 직접 사용하지 않고  
millis() 함수를 이용해 타이머 구현

- 아두이노에 TimerOne 라이브러리를 설치하고 100ms마다 타이머 인터럽트를 발생시키고 1초마다 D13 에 연결된 LED를 Blink하시오.

## 복습

 오픈플랫폼 입문의 인터럽트 기능 사용하기, 타이머기능 사용하기

 인터럽트 기능 사용하기

...> <https://cafe.naver.com/ctcemb/2541>

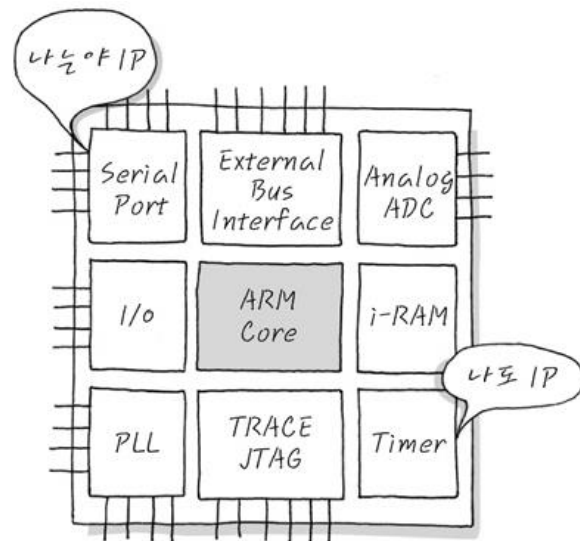
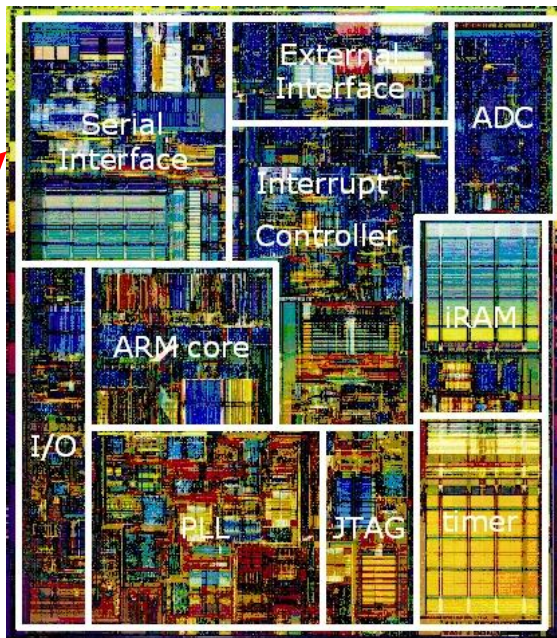
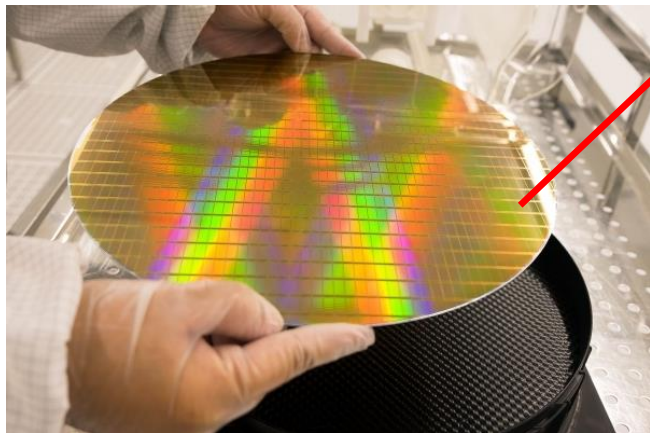
 타이머 기능 사용하기

...> <https://cafe.naver.com/ctcemb/2542>

# 실제 CPU는 어떻게 생겼을까?

## ⚙️ ARM SoC(System On Chip)

🌈 웨이퍼 한 장에 똑같은 여러 개의 IC칩이 생산

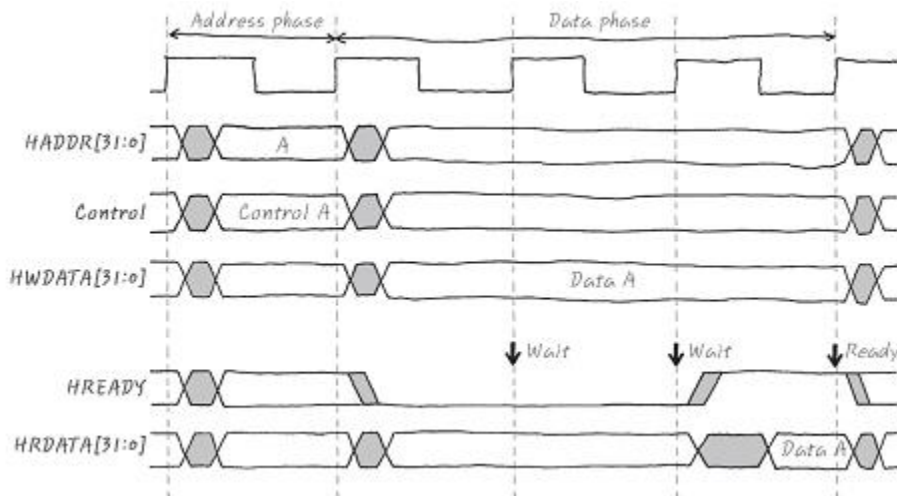
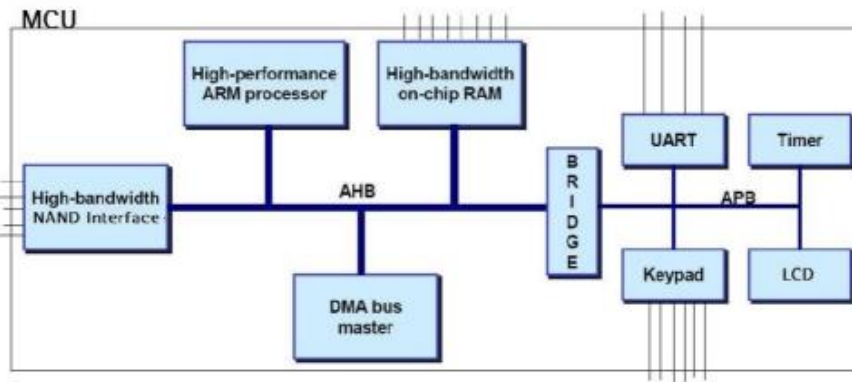


→ <http://recipes.egloos.com/4990377>

# ARM Core와 버스

⚙️ ARM core와 IP들은 AMBA라는 버스로 통신

🌈 AMBA - SoC안에서 IP끼리의 Bus 규격



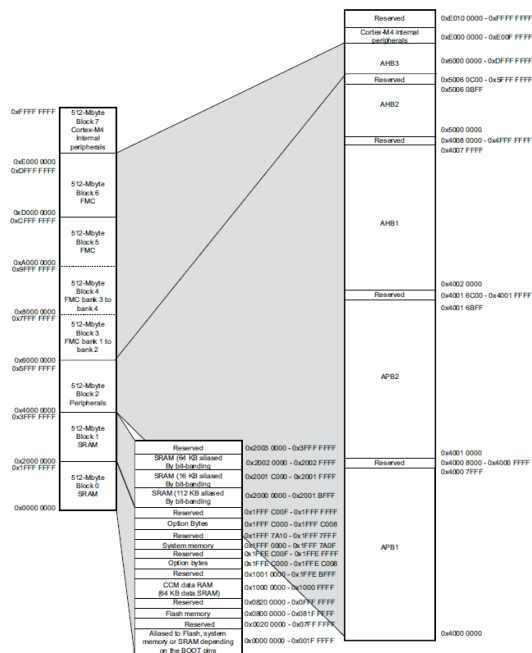
→ ARM사가 만든 AMBA버스는 크게 AHB버스와 APB버스가 있음

→ <http://recipes.egloos.com/4990377>

## CPU 내부의 Register들은 memory map의 주소로 접근

## STM32F429의 memory map

**Figure 19. Memory map**



- Memory map은 ARM core와 IP들을 AMBA버스로 연결할 때 정한 주소 모음
- STM32F429 datasheet P. 86
- APB1 : 0x4000 0000 ~ 0x4000 7FFF
- APB2 : 0x4001 0000 ~ 0x40016BFF
- AHB1 : 0x4002 0000 ~ 0x4007 FFFF
- AHB2 : 0x5000 0000 ~ 0x5006 0BFF
- AHB3 : 0x6000 0000 ~ 0xDFFF FFFF

# Memory map으로 레지스터 주소 찾기

## STM32F429의 GPIOB 레지스터 주소 찾기

### 메모리 맵의 레지스터에 관련 버스 표기

Table 1. STM32F4xx register boundary addresses (continued)

Boundary address	Peripheral	Bus	Register map
0x5006 0800 - 0x5006 0BFF	RNG	AHB2	Section 24.4.4: RNG register map on page 771
0x5006 0400 - 0x5006 07FF	HASH		Section 25.4.9: HASH register map on page 795
0x5006 0000 - 0x5006 03FF	CRYP		Section 23.6.13: CRYP register map on page 763
0x5005 0000 - 0x5005 03FF	DCMI		Section 15.8.12: DCMI register map on page 478
0x5000 0000 - 0x5003 FFFF	USB OTG FS		Section 34.16.6: OTG_FS register map on page 1326
0x4004 0000 - 0x4007 FFFF	USB OTG HS	AHB1	Section 35.12.6: OTG_HS register map on page 1472
0x4002 B000 - 0x4002 BBFF	DMA2D		Section 11.5: DMA2D registers on page 352
0x4002 8000 - 0x4002 93FF	ETHERNET MAC		Section 33.8.5: Ethernet register maps on page 1236
0x4002 6400 - 0x4002 67FF	DMA2		Section 10.5.11: DMA register map on page 335
0x4002 6000 - 0x4002 63FF	DMA1		
0x4002 4000 - 0x4002 4FFF	BKPSRAM		
0x4002 3C00 - 0x4002 3FFF	Flash interface register		Section 3.9: Flash interface registers
0x4002 3800 - 0x4002 3BFF	RCC		Section 7.3.24: RCC register map on page 265
0x4002 3000 - 0x4002 33FF	CRC		Section 4.4.4: CRC register map on page 115
0x4002 2800 - 0x4002 2BFF	GPIOK		Section 8.4.11: GPIO register map on page 287
0x4002 2400 - 0x4002 27FF	GPIOJ		
0x4002 2000 - 0x4002 23FF	GPIOI		Section 8.4.11: GPIO register map on page 287
0x4002 1C00 - 0x4002 1FFF	GPIOH		
0x4002 1800 - 0x4002 1BFF	GPIOG		
0x4002 1400 - 0x4002 17FF	GPIOF		
0x4002 1000 - 0x4002 13FF	GPIOE		
0x4002 0C00 - 0x4002 0FFF	GPIOD		
0x4002 0800 - 0x4002 0BFF	GPIOC		
0x4002 0400 - 0x4002 07FF	GPIOB		
0x4002 0000 - 0x4002 03FF	GPIOA		
0x4001 6800 - 0x4001 6BFF	LCD-TFT	APB2	Section 16.7.26: LTDC register map on page 512
0x4001 5800 - 0x4001 5BFF	SAI1	APB2	Section 29.17.9: SAI register map on page 963
0x4001 5400 - 0x4001 57FF	SPI6		Section 28.5.10: SPI register map on page 925
0x4001 5000 - 0x4001 53FF	SPI5		

→ STM32F429 Reference Manual P.65

→ GPIOB 레지스터는  
0x40020400~0x400207FF

→ TIM2는 관련 레지스터가  
0x40000000~0x400003FF 이고 해당 버스는  
APB1



# STM32F429의 타이머 구조

## STM32F429의 clock

### STM32F429의 clock 설정

STM32F429

최대 180MHz속도로 동작할 수 있는 CPU

- ...> 동작 속도가 빠르면 전력소모가 많음
- ...> 전력소모를 최소화해야 하는 응용분야에서는 최대 속도로 동작시킬 필요 없음



# STM32F429의 타이머 구조

## STM32F429의 clock

### STM32F429의 clock 설정

- ... STM32F429는 외부에 clock(크리스탈이나 오실레이터)을 장착하지 않아도 사용이 가능함
- ... CPU 내부에 16MHz의 CPU clock과 32KHz의 RTC(Real Time Clock)가 있음
- ... CPU clock은 HSE, HSI 중 하나를 선택할 수 있고 초단위 시간을 재는 RTC는 LSE, LSI중 하나를 선택가능

# STM32F429의 타이머 구조

## STM32F429의 clock

### STM32F429의 clock 설정

... 총 4개의 clock이 있으며 각각 External이나 Internal로 선택 가능

HSE

High Speed  
External

HSI

High Speed  
Internal

LSE

Low Speed  
External  
(32.768KHz)

LSI

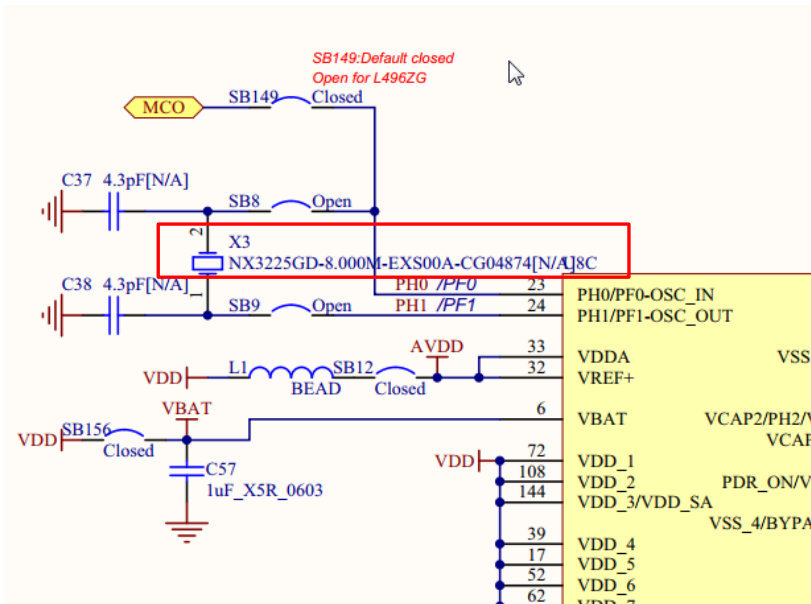
Low Speed Internal  
(32KHz)

# STM32F429의 타이머 구조

## STM32F429의 clock

### Nucleo-F429 보드의 clock 관련 상태

#### Nucleo-F429보드의 CPU clock 크리스탈



X3 8MHz

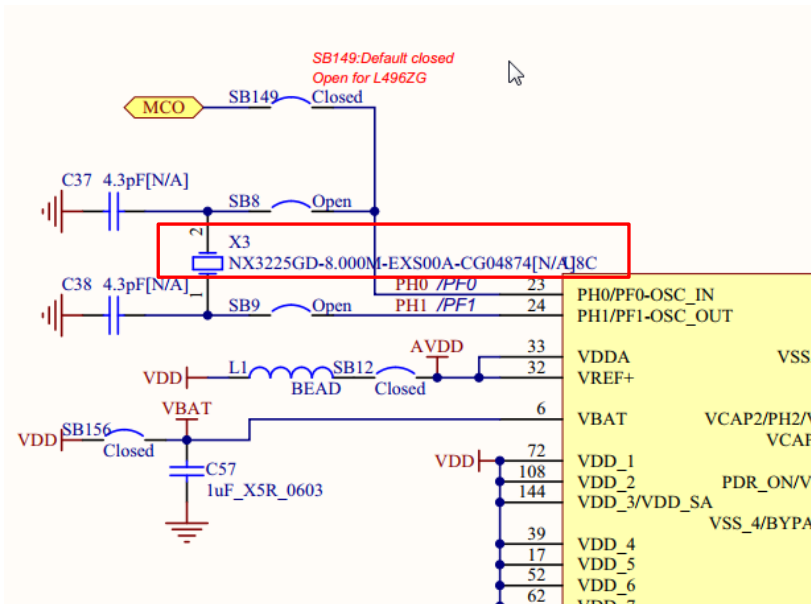
... 크리스탈이며, 미장착되어  
있음

# STM32F429의 타이머 구조

## STM32F429의 clock

### Nucleo-F429 보드의 clock 관련 상태

#### Nucleo-F429보드의 CPU clock 크리스탈



[N/A]

... Not Assembled 라는  
의미로 미조립 상태임

# STM32F429의 타이머 구조

## STM32F429의 clock

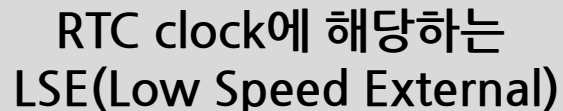
### Nucleo-F429 보드의 clock 관련 상태



HSE(High Speed External)를 선택할 수 없고 HSI(High Speed Internal) 를 선택해야 함

## STM32F429의 clock

## Nucleo-F429보드의 RTC clock 크리스탈



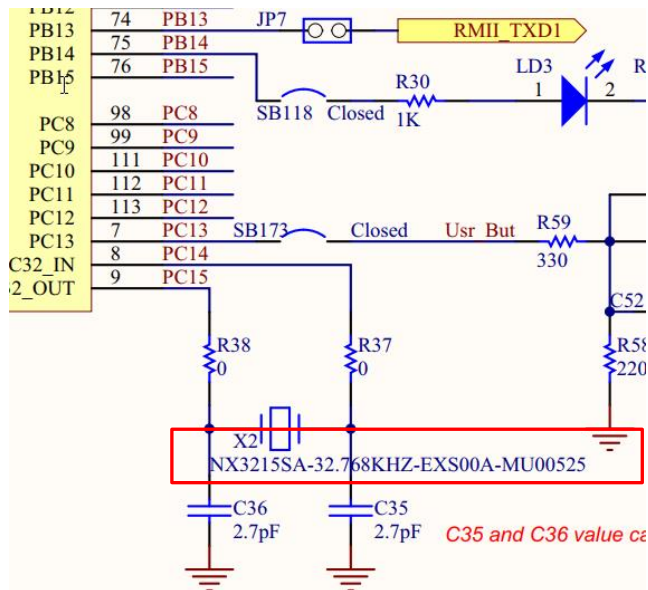
... X2 32.768KHz 크리스탈로  
장착되어 있음

# STM32F429의 타이머 구조

## STM32F429의 clock

### Nucleo-F429 보드의 clock 관련 상태

#### Nucleo-F429보드의 RTC clock 크리스탈



초단위의 시간을 재는  
LS(Low Speed)



... LSE, LSI 선택 가능, 외부 크리스탈인 32.768KHz가 더 정확하므로 LSE를 선택하는 것이 바람직함



# STM32F429의 타이머 구조

## STM32F429의 clock

### CubeMX로 Nucleo-F429 보드의 clock 설정

- ...> STM32F429의 Clock 설정은 복잡하지만 CubeMX툴을 사용하여 비교적 쉽게 설정이 가능
- ...> 다음은 Nucleo-F429보드에 맞게 최대 CPU속도로 설정하는 과정을 설명함
  - Nucleo-F429보드를 기준으로 한 새로운 프로젝트 생성
  - Low Speed Clock: LSE로 선택

# STM32F429의 타이머 구조

## STM32F429의 clock

 CubeMX로 Nucleo-F429 보드의 clock 설정

... CubeMX로 LSE를 선택하는 과정

1

System Core

RCC

LSE

Crystal/Ceramic Resonator 선택

# STM32F429의 타이머 구조

## STM32F429의 clock

### CubeMX로 Nucleo-F429 보드의 clock 설정

... CubeMX로 LSE를 선택하는 과정

2

PC14번과 PC15번이 RCC\_OSC32\_IN, RCC\_OSC32\_OUT으로 수정됨을 확인

3

Timers -> RTC 에 Activate Clock Source를 활성화

4

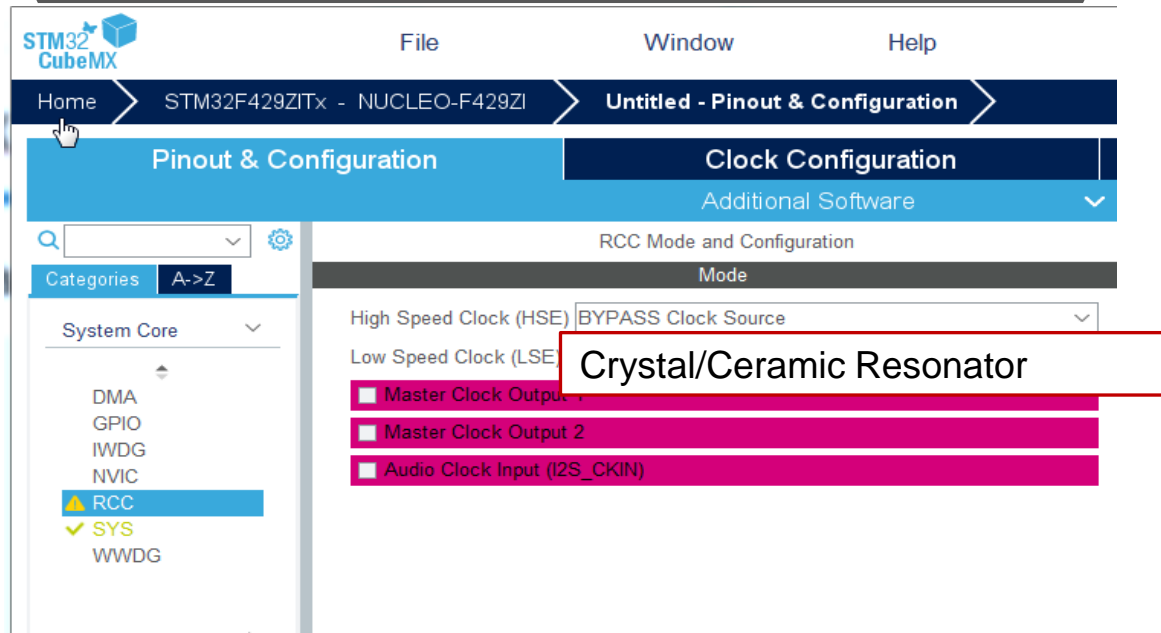
Clock Configuration에서 RTC Clock Mux를 LSE로 선택

# STM32F429의 타이머 구조

## STM32F429의 clock

### CubeMX로 Nucleo-F429 보드의 clock 설정

#### CubexMX를 통한 LSE 설정



# STM32F429의 타이머 구조

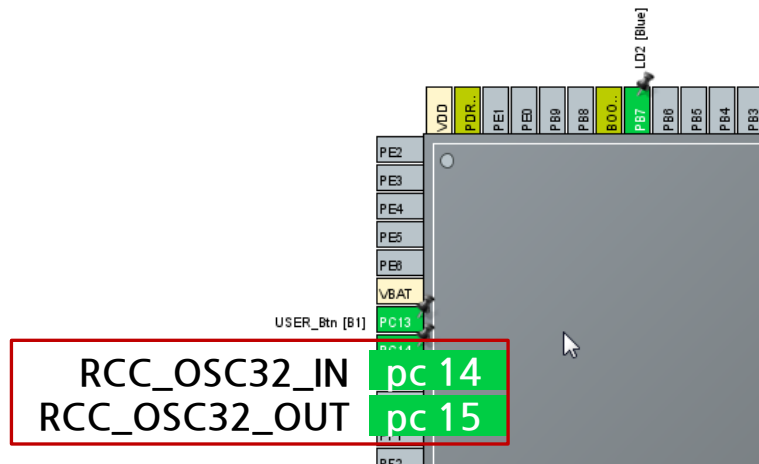
## STM32F429의 clock

### CubeMX로 Nucleo-F429 보드의 clock 설정

#### clock 관련 핀 설정

Pinout view

System view

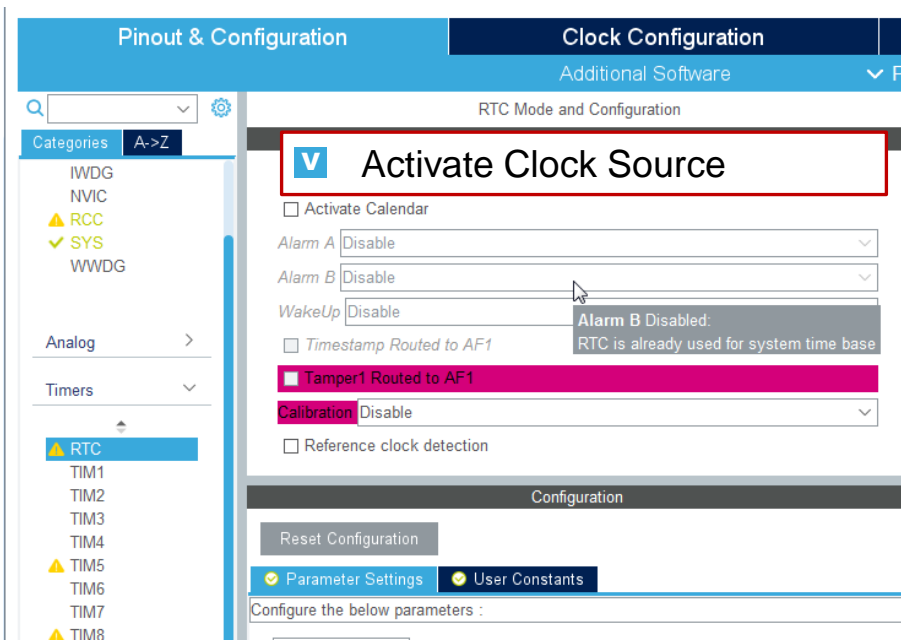


# STM32F429의 타이머 구조

## STM32F429의 clock

### CubeMX로 Nucleo-F429 보드의 clock 설정

#### RTC 설정을 Activate Clock Source



# STM32F429의 타이머 구조

## STM32F429의 clock

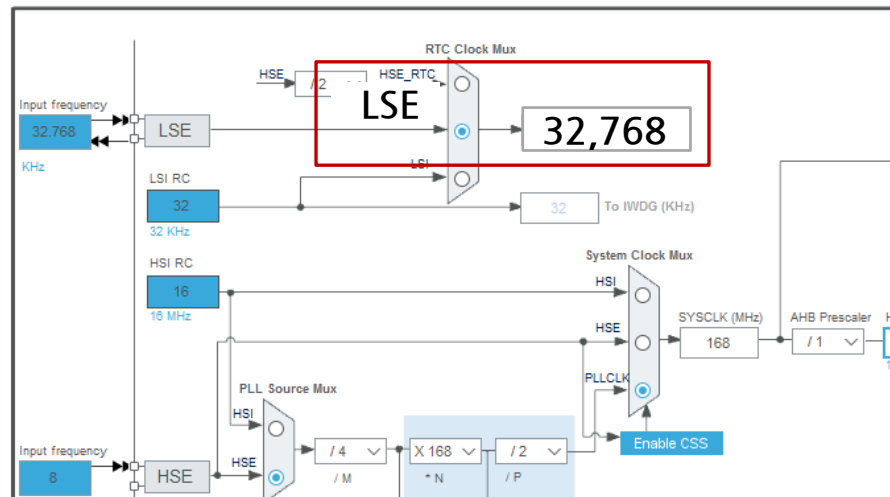
### CubeMX로 Nucleo-F429 보드의 clock 설정

#### LSE 선택

Pinout & Configuration

Clock Configuration

Resolve Clock Issues





# STM32F429의 타이머 구조



## STM32F429의 타이머



### 타이머의 종류

Systick

WatchDog  
Timer  
(IWDG, WWDG)

Basic Timer

General  
purpose Timer

Advanced-  
control Timer

# STM32F429의 타이머 구조

## STM32F429의 타이머

### 타이머의 종류

SysTick

항상 동작하는 시스템 타이머, HAL\_Delay()함수등에 사용

# STM32F429의 타이머 구조

## STM32F429의 타이머

### 타이머의 종류

WatchDog  
Timer(IWDG, WWDG)

CPU의 오동작을 탐지하여 문제가 발생하면 재부팅시켜주는 타이머

# STM32F429의 타이머 구조

## STM32F429의 타이머

### 타이머의 종류

Basic Timer

입출력 기능은 없고 시간기반 타이머용도 , TIMx (x:6,7)

# STM32F429의 타이머 구조

## STM32F429의 타이머

### 타이머의 종류

#### General purpose Timer

범용 타이머, 출력 비교, 원펄스, 입력캡처등으로 사용하는 타이머,  
TIMx (x: 2~5, 9~14)

# STM32F429의 타이머 구조

## STM32F429의 타이머

### 타이머의 종류

#### Advanced-control Timer

범용 타이머보다 많은 기능을 가지는 타이머,  
주로 모터 제어와 디지털 파워변환 용도로 사용, TIMx (x:1,8)

# STM32F429의 타이머 구조

## STM32F429의 타이머 레지스터

### General purpose Timer를 사용한 타이머 설정

- ... General purpose timer인 TIM2를 이용하여 10ms 주기의 타이머 인터럽트를 발생시키는 설정 예
- ... 원하는 주기를 발생시키려면 기본적으로 3가지를 설정할 수 있어야 함
- ... Timer에 공급되는 버스 Clock 속도 , Prescaler 값 , Peroid 값



# STM32F429의 타이머 구조

## STM32F429의 타이머 레지스터

### General purpose Timer를 사용한 타이머 설정

#### TIM2의 버스 clock

0x4000 3C00 - 0x4000 3FFF	SPI3 / I2S3	APB1	Section 28.5.10: SPI register map on page 925
0x4000 3800 - 0x4000 3BFF	SPI2 / I2S2		
0x4000 3400 - 0x4000 37FF	I2S2ext		
0x4000 3000 - 0x4000 33FF	IWDG	APB1	Section 21.4.5: IWDG register map on page 712
0x4000 2C00 - 0x4000 2FFF	WWDG		Section 22.6.4: WWDG register map on page 719
0x4000 2800 - 0x4000 2BFF	RTC & BKP Registers		Section 26.6.21: RTC register map on page 836
0x4000 2000 - 0x4000 23FF	TIM14		Section 19.5.12: TIM10/11/13/14 register map on page 694
0x4000 1C00 - 0x4000 1FFF	TIM13		
0x4000 1800 - 0x4000 1BFF	TIM12		Section 19.4.13: TIM9/12 register map on page 684
0x4000 1400 - 0x4000 17FF	TIM7		Section 20.4.9: TIM6 and TIM7 register map on page 707
0x4000 1000 - 0x4000 13FF	TIM6		
0x4000 0C00 - 0x4000 0FFF	TIM5		
0x4000 0800 - 0x4000 0BFF	TIM4		
0x4000 0400 - 0x4000 07FF	TIM3		Section 18.4.21: TIMx register map on page 648
0x4000 0000 - 0x4000 03FF	TIM2		

타이머에 따라 공급되는 버스 clock이 다름



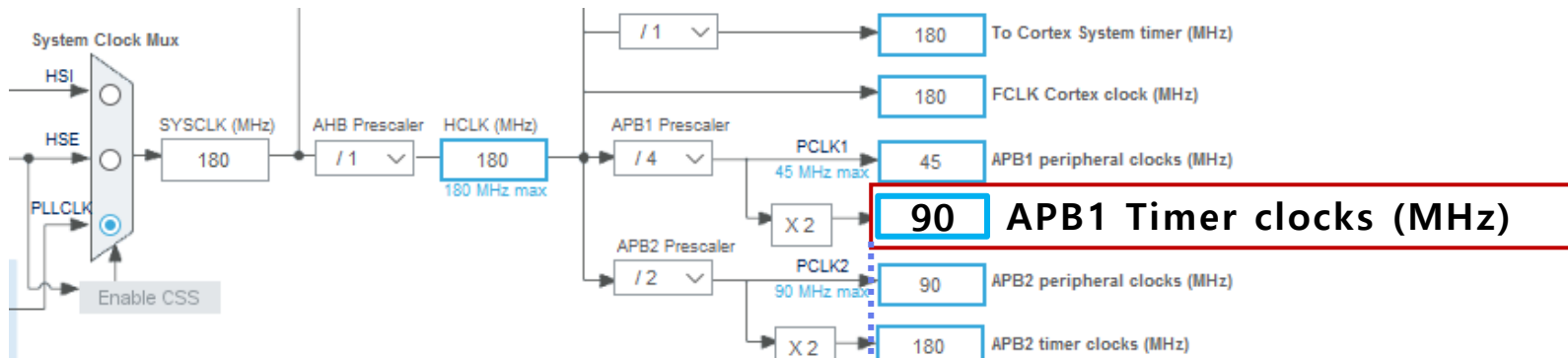
TIM2는 APB1을 사용

# STM32F429의 타이머 구조

## STM32F429의 타이머 레지스터

### General purpose Timer를 사용한 타이머 설정

#### APB1의 속도



APB1은 90MHz로 동작함

# STM32F429의 타이머 구조

## STM32F429의 타이머 레지스터

General purpose Timer를 사용한 타이머 설정

→ TIM2의 clock source 를 Internal clock으로 설정

### TIM2의 Clock Source

The screenshot shows the 'TIM2 Mode and Configuration' window in STM32CubeMX. On the left, a sidebar lists categories: System Core, Analog, and Timers. Under Timers, RTC, TIM1, and TIM2 are listed, with TIM2 selected and highlighted in blue. The main panel displays the configuration for TIM2. The 'Clock Source' dropdown menu is highlighted with a red box and contains the text 'Internal clock'. Other settings include Slave Mode (Disable), Trigger Source (Disable), Channel1 (Disable), Channel2 (Disable), Channel3 (Disable), Channel4 (Disable), and Combined Channels (Disable).

TIM2 Mode and Configuration	
Mode	
Slave Mode	Disable
Trigger Source	Disable
Clock Source	Internal clock
Channel1	Disable
Channel2	Disable
Channel3	Disable
Channel4	Disable
Combined Channels	Disable

# STM32F429의 타이머 구조

## STM32F429의 타이머 레지스터

### General purpose Timer를 사용한 타이머 설정

원하는 주기

Period  $\times$  (1/APB1)  $\times$  Prescaler

0.01 초

900  $\times$  (1/90MHz)  $\times$  1000

# STM32F429의 타이머 구조

## STM32F429의 타이머 레지스터

### General purpose Timer를 사용한 타이머 설정

#### TIM2의 Prescaler와 Period

Configuration

Reset Configuration

☒ Parameter Settings ☒ User Constants ☒ NVIC Settings ☒ DMA Settings

Configure the below parameters :

Count

Prescaler (PSC - 16 bits value)

1000

Counter Mode

Up

Counter Period (AutoReload Register - 32 bits va...

900

Internal Clock Division (CKD)

No Division

auto-reload preload

Disable

Trigger Output (TRGO) Parameters

Master/Slave Mode (MSM bit)

Disable (Trigger input effect not delayed)

Trigger Event Selection

Reset (UG bit from TIMx\_EGR)

# STM32F429의 타이머 제어 SW 설계하기

## SysTick 타이머 제어 초기화 SW 생성

### 동영상

CubeMX 를 사용하여 보드 선택



최대 clock으로 설정

▶ NVIC 설정에  
SysTick 타이머 설정 확인



코드 생성

▶ 1초단위로 UART 메시지를  
출력하는 코드 설계



UART 메시지 확인

...▶ 1ms속도의 인터럽트가 발생함을 확인하기 위함

# STM32F429의 타이머 제어 SW 설계하기

## SysTick 타이머 제어 초기화 SW 생성

### 동영상

...> 앞에 소개된 코드를 작성하고 동작시켜 테스트 진행

앞에 작성된 코드에 추가하여  
Timer2 clock source 설정



Prescaler와 Period설정



TIM2의 NVIC설정에서  
TIM2 break interrupt 활성화



TIM2의 PeriodElapsed  
Callback 함수 작성



10ms단위의 timer로 1초  
단위의 counter 를 UART로 출력하기



# 알람시계 실습

## 소스코드 관리

### 실습

... 타이머 관련 코드를 분리하여 Src 폴더 아래에 timer.c로 관리하시오.

Q & A  
Thank you

