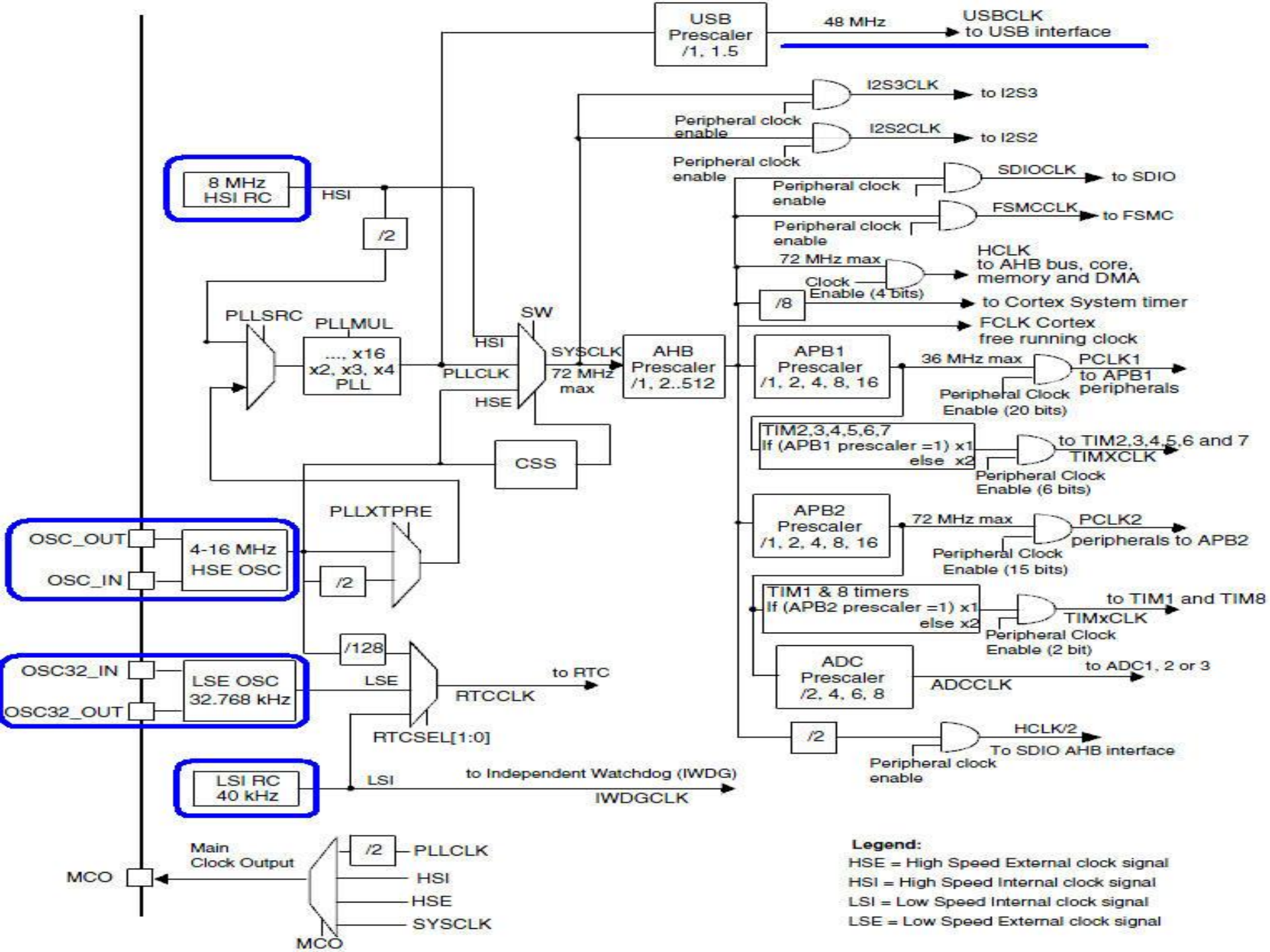
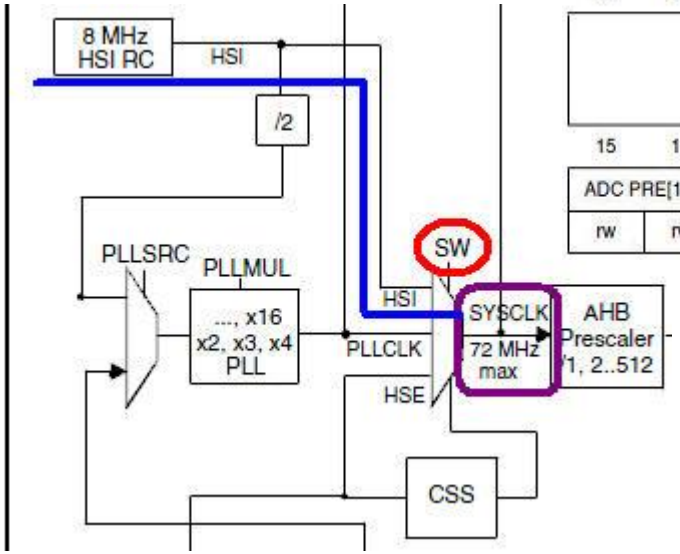
Clock Control 실습

2009.11.20


USB Prescaler /1, 1.5
48 MHz
USBCLK to USB interface
I2S3CLK
to I2S3
Peripheral clock enable
I2S2CLK
to I2S2
Peripheral clock enable
SDIOCLK
to SDIO
Peripheral clock enable
FSMCCLK
to FSMC
Peripheral clock enable
72 MHz max
Clock Enable (4 bits)
HCLK to AHB bus, core, memory and DMA
/8
to Cortex System timer
FCLK Cortex free running clock
8 MHz HSI RC
HSI
/2
PLLSRC
PLLMUL
..., x16 x2, x3, x4 PLL
HSI
PLLCLK
HSE
SW
SYSCLK
72 MHz max
AHB Prescaler /1, 2..512
APB1 Prescaler /1, 2, 4, 8, 16
36 MHz max
PCLK1 to APB1 peripherals
Peripheral Clock Enable (20 bits)
TIM2,3,4,5,6,7 If (APB1 prescaler =1) x1 else x2
to TIM2,3,4,5,6 and 7 TIMXCLK
Peripheral Clock Enable (6 bits)
CSS
PLLXTPRE
OSC_OUT
OSC_IN
4-16 MHz HSE OSC
/2
APB2 Prescaler /1, 2, 4, 8, 16
72 MHz max
PCLK2 peripherals to APB2
Peripheral Clock Enable (15 bits)
TIM1 & 8 timers If (APB2 prescaler =1) x1 else x2
to TIM1 and TIM8 TIMxCLK
Peripheral Clock Enable (2 bit)
/128
OSC32_IN
OSC32_OUT
LSE OSC 32.768 kHz
LSE
to RTC
RTCCLK
RTCSEL[1:0]
ADC Prescaler /2, 4, 6, 8
to ADC1, 2 or 3
ADCCLK
/2
HCLK/2
To SDIO AHB interface
Peripheral clock enable
LSI RC 40 kHz
LSI
to Independent Watchdog (IWDG)
IWDGCLK
Main Clock Output
MCO
/2
PLLCLK
HSI
HSE
SYSCLK
MCO
Legend:
HSE = High Speed External clock signal
HSI = High Speed Internal clock signal
LSI = Low Speed Internal clock signal
LSE = Low Speed External clock signal

Clock configuration register (RCC_CFGR)

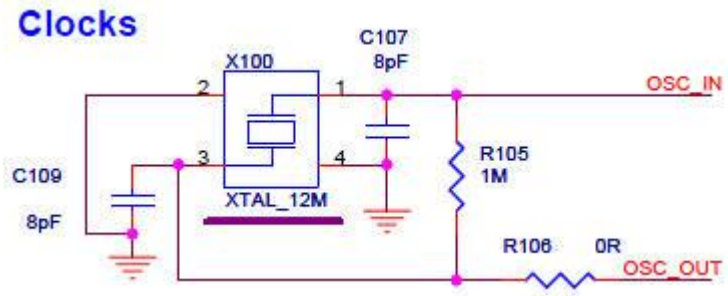


31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					MCO[2:0]			Res.	USB PRE	PLLMUL[3:0]				PLL XTPRE	PLL SRC
					rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC PRE[1:0]		PPRE2[2:0]			PPRE1[2:0]			HPRE[3:0]				SWS[1:0]		SW[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	rw	rw

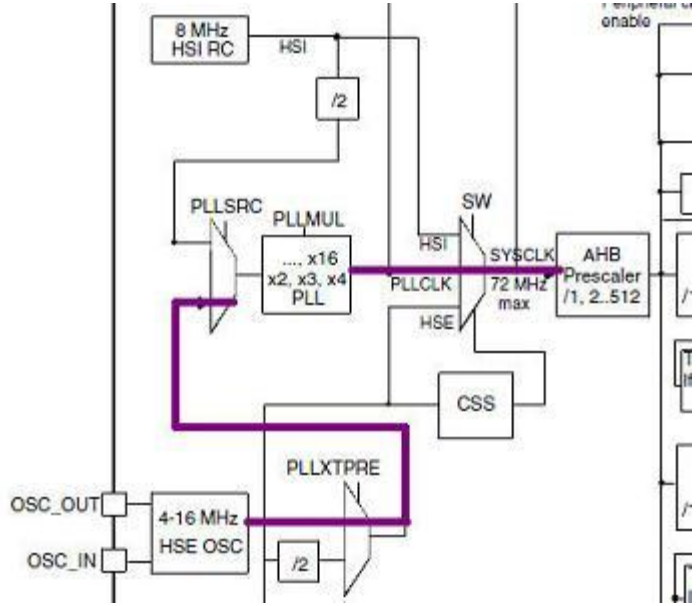
- Clock configuration register (RCC_CFGR)
  - Address offset: 0x04, Reset value: 0x0000 0000
- Bits 1:0 SW: System clock switch
  - **00: HSI selected as system clock**
  - 01: HSE selected as system clock
  - 10: PLL selected as system clock
  - 11: not allowed

12M crystal



- 망고보드의 외부에는 12M의 crystal이 달려있다.
- 외부에 이러한 clock을 달아 놓은 것은 **USB** 때문이다.
- USB를 위한 48 MHz Clock은 main PLL에서 생성되는데 이를 위한 clock source는 반드시 HSE crystal oscillator를 사용해야만 한다
- 내부에 기본적으로 들어있는 HSI oscillator의 정확도는 USB 규격을 만족시킬 수 있을 만큼 정확하지 않다. 그래서 USB를 사용하기 위해서는 내부 HSI를 사용할 수 없고 외부에 보다 정확한 crystal을 달아서 이용할 수밖에 없는 것이다.

72 MHz로 설정



- 72 MHz를 사용 - **PLL을 거쳐서 생성된 클럭**을 사용할 수밖에 없다.
- PLL 소스는 8 MHz의 내부 HSI를 사용하거나 외부 HSE를 사용
- 우리는 12 MHz의 crystal을 달아 놓았고 이것을 이용한다. 이것을 PLL을 구동 시키는 소스로 활용해야 한다.
- PLLMUL을 통해서 입력으로 들어온 클럭의 수치를 올려주어야 한다. 우리는 72 MHz를 사용할 것이기 때문에 **12 X 6 = 72가 되고, 곱할 값을 6으로 설정**해야 한다.

HSEON & HSERDY

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						PLL RDY	PLLON	Reserved				CSS ON	HSE BYP	HSE RDY	HSE ON
						r	rw					rw	rw	r	rw

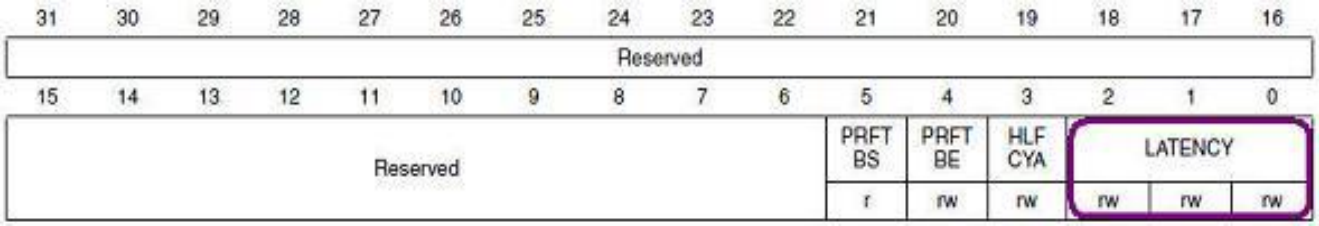
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]								HSITRIM[4:0]					Res.	HSI RDY	HSION
r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw		r	rw

```
RCC->CR |= ((uint32_t)RCC_CR_HSEON); /* Enable HSE */

/* Wait till HSE is ready and if Time out is reached exit */
do {
    HSEStatus = RCC->CR & RCC_CR_HSERDY;
    StartUpCounter++;
} while((HSEStatus == 0)
          && (StartUpCounter != HSEStartUp_TimeOut));
```

- HSE를 On 시켰을 때 바로 사용 가능하게 되는 것이 아니다.
- HSERDY - HSE oscillator가 사용 가능해졌는지 알수 있는 비트

Flash access Latency 설정

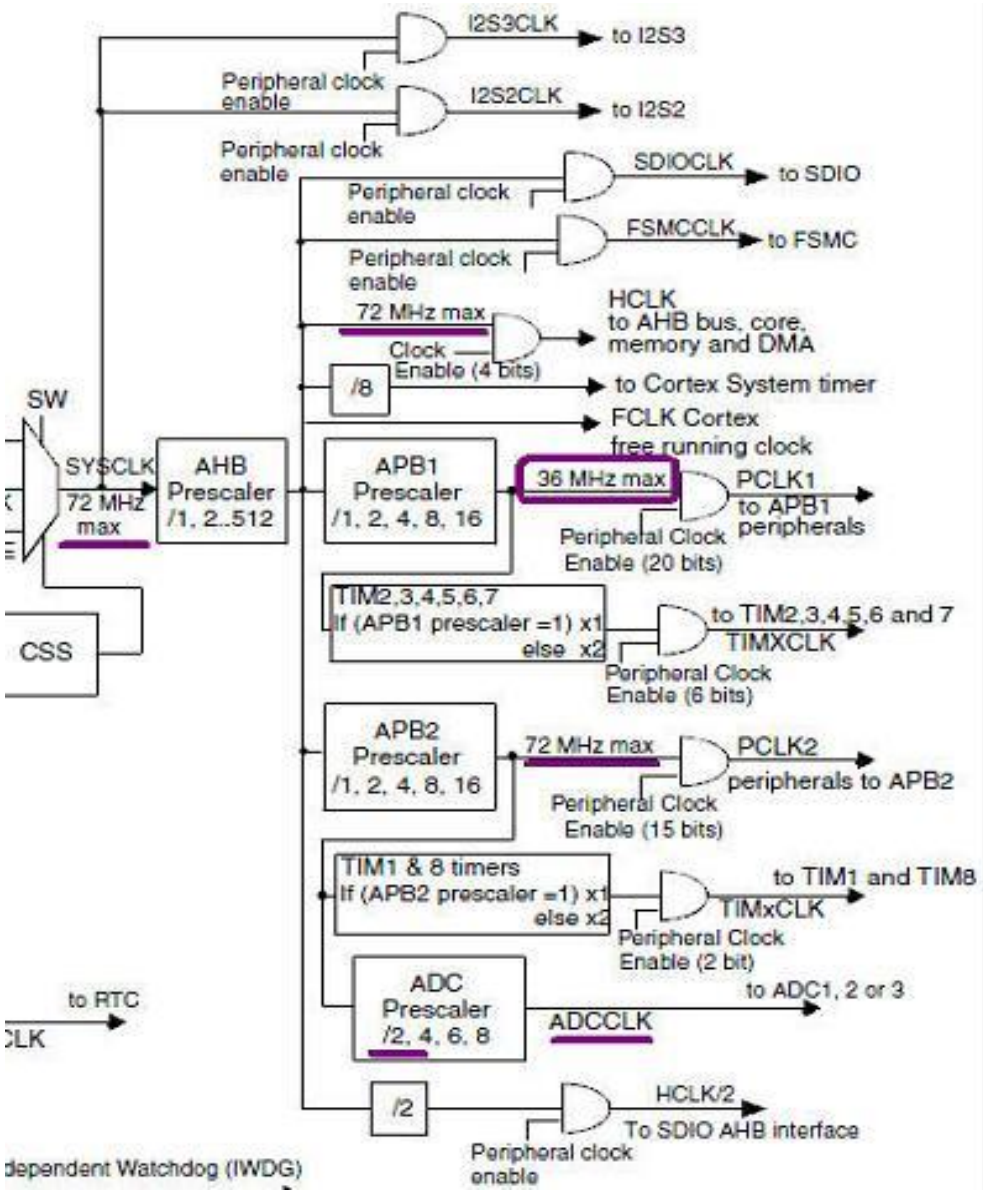



```
#define  FLASH_ACR_LATENCY_2   ((uint8_t)0x02)
FLASH->ACR |= (uint32_t)FLASH_ACR_LATENCY_2;
```

- 지금까지는 내부 **HSI 8 MHz**를 사용했다. 이 경우 클럭이 느리기 때문에 Flash를 access하기 위해서 기다릴 필요가 없다.
- 클럭을 72 MHz로 올리면 클럭이 빠르기 때문에 정상적으로 Flash access를 할 수가 없다. 적절하게 기다리는 시간이 필요하게 된다.

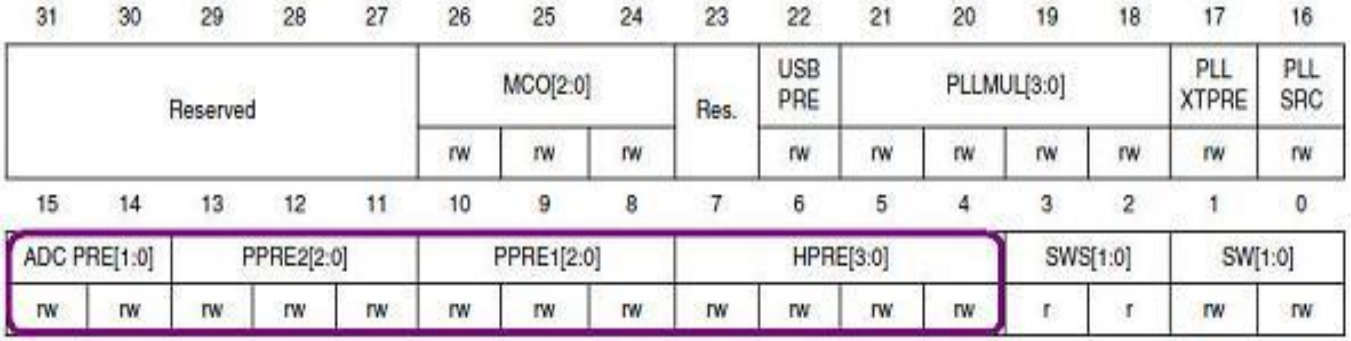
- Bits 2:0 LATENCY: Latency
  - 000 Zero wait state, if 0 < SYSCLK≤ 24 MHz
  - 001 One wait state, if 24 MHz < SYSCLK ≤ 48 MHz
  - **010 Two wait states, if 48 MHz < SYSCLK ≤ 72 MHz**

HCLK, PCLK1, PCLK2 설정



- SYSCLK는 72 MHz로 max로 설정
- HCLK는 역시 72 MHz로 max로 설정
- PCLK1은 36 MHz가 max
- PCLK2는 72 MHz로 max로 설정
- ADCCLK는 /2가 가장 작은 값이고 결국 36 MHz가 max

Clock configuration register (RCC_CFGR) 설정 (1)




```
#define  RCC_CFGR_HPRE_DIV1      ((uint32_t)0x00000000)
#define  RCC_CFGR_PPRE2_DIV1     ((uint32_t)0x00000000)
#define  RCC_CFGR_PPRE1_DIV2     ((uint32_t)0x00000400)
```

```
RCC->CFGR |= (uint32_t)RCC_CFGR_HPRE_DIV1;
RCC->CFGR |= (uint32_t)RCC_CFGR_PPRE2_DIV1;
RCC->CFGR |= (uint32_t)RCC_CFGR_PPRE1_DIV2;
```

Clock configuration register (RCC_CFGR) 설정 (2)

- Bits 14:14 ADCPRE: ADC prescaler - (ADCCLK)
  - 00: PLCK2 divided by 2
  - ADC prescaler 작업을 하지 않았다. 0으로 설정. PLCK2 / 2 (36 MHz)
- Bits 13:11 PPRE2: APB high-speed prescaler (APB2) - (PCLK2).
  - 0xx: HCLK not divided
  - PCLK2도 0으로 설정되고 우리의 경우 72 MHz가 설정된다.
- Bits 10:8 PPRE1: APB low-speed prescaler (APB1) - (PCLK1).
  - 0xx: HCLK not divided
  - 100: HCLK divided by 2
  - PCLK1의 경우 0으로 설정하면 HCLK를 그대로 사용할 수 있게 되지만 만약 **HCLK가 36 MHz보다 크게 되면 여기의 비트를 0으로 만들 수는 없다.**
  - 우리는 100을 선택하였고, 결국 36 MHz가 설정되게 하였다.
- Bits 7:4 HPRE: AHB prescaler (HCLK)
  - 0xxx: SYSCLK not divided
  - 1000: SYSCLK divided by 2
  - HCLK는 0으로 설정해서 System clock을 그대로 사용하여 결국 72 MHz가 되도록 설정한다.

PLL 설정


```
#define  RCC_CFGR_PLLSRC_HSE     ((uint32_t)0x00010000)
#define  RCC_CFGR_PLLMULL6       ((uint32_t)0x00100000)
RCC->CFGR |= (uint32_t)(RCC_CFGR_PLLSRC_HSE | RCC_CFGR_PLLMULL6);
```

- Bits 21:18 PLLMUL: PLL multiplication factor
  - **0100: PLL input clock x 6 = 72 MHz**
  - 이 값은 반드시 **PLL**이 **disable** 되어 있는 상태에서 설정해야 한다
- Bit 16 PLLSRC: PLL entry clock source
  - 0: HSI oscillator clock / 2 selected as PLL input clock
  - **1: HSE oscillator clock selected as PLL input clock**
  - 이 비트 역시 **PLL**이 **disable** 되어 있는 상태에서 설정해야 한다.
  - 외부 클럭 **HSE oscillator**를 **PLL**의 입력으로 사용할 것이기 때문에 이 비트를 1로 설정한다.

PLL ON - PLL enable 작업

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						PLL RDY	PLLON	Reserved				CSS ON	HSE BYP	HSE RDY	HSE ON
						r	rw					rw	rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]								HSITRIM[4:0]					Res.	HSI RDY	HSION
r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw		r	rw

```
#define  RCC_CR_PLLON        ((uint32_t)0x01000000)
#define  RCC_CR_PLLRDY       ((uint32_t)0x02000000)
RCC->CR |= RCC_CR_PLLON;
while((RCC->CR & RCC_CR_PLLRDY) == 0){;}
```

- HSE On & Wait 작업과 비슷한 작업
- Bit 24 PLLON: PLL enable
  - 0: PLL OFF, **1: PLL ON**
- Bit 25 PLLRDY: PLL clock ready flag
  - 0: PLL unlocked
  - **1: PLL locked**
  - PLL이 Lock 되었는지 검사. 1로 바뀌면 PLL이 Lock되었다는 것을 의미

System clock switch 설정 - RCC_CFGR 레지스터


```
#define  RCC_CFGR_SW_PLL    ((uint32_t)0x00000002)
#define  RCC_CFGR_SWS       ((uint32_t)0x0000000C)
RCC->CFGR |= (uint32_t)RCC_CFGR_SW_PLL;
while ((RCC->CFGR & (uint32_t)RCC_CFGR_SWS) != (uint32_t)0x08){;}
```

- Bits 1:0 SW: System clock switch
  - **10: PLL selected as system clock**
  - PLL을 시스템 클럭 소스로 사용하게 설정
  - HSE ON, PLL ON과 마찬가지로 **switch 설정 후 SWS 비트 Wait 필요**
- Bits 3:2 SWS: System clock switch status
  - 10: PLL used as system clock
  - 10 설정이 시스템 클럭 소스가 PLL 사용. 이때까지 값을 읽어서 기다린다.

SetSysClockTo72

```
int main(void)
{
    uint8_t ch;

    /* System Clocks Configuration */
    RCC_Configuration();

    /* Configure the GPIO ports */
    GPIO_Configuration();
```

```
void SystemInit (void)
{
  /* Reset the RCC clock configuration
  /* Set HSION bit */
  RCC->CR |= (uint32_t)0x00000001;

  /* Reset SW, HPRE, PPRE1, PPRE2, ADCP
#ifndef STM32F10X_CL
  RCC->CFGR &= (uint32_t)0xF8FF0000;
#else
  RCC->CFGR &= (uint32_t)0xF0FF0000;
#endif /* STM32F10X_CL */

  /* Reset HSEON, CSSON and PLLON bits
  RCC->CR &= (uint32_t)0xFEF6FFFF;

  /* Reset HSEBYP bit */
  RCC->CR &= (uint32_t)0xFFFBFFFF;

  /* Reset PLLSRC, PLLXTPRE, PLLMUL and
  RCC->CFGR &= (uint32_t)0xFF80FFFF;

#ifndef STM32F10X_CL
  /* Disable all interrupts and clear p
  RCC->CIR = 0x009F0000;
#else
  /* Reset PLL2ON and PLL3ON bits */
  RCC->CR &= (uint32_t)0xEBFFFFFF;

  /* Disable all interrupts and clear p
  RCC->CIR = 0x00FF0000;

  /* Reset CFGR2 register */
  RCC->CFGR2 = 0x00000000;
#endif /* STM32F10X_CL */

  /* Configure the System clock frequer
  /* Configure the Flash Latency cycles
  SetSysClock();
} ? end SystemInit ?
```

```
void RCC_Configuration(void)
{
  SystemInit();

  /* Enable GPIOA clock */
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIO_USART, ENABLE);

  /* Enable GPIOB clock */
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIO_LED, ENABLE);

  /* Enable USART1 clocks */
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1, ENABLE);
}
```

```
static void SetSysClock(void)
{
#ifdef SYSCLK_FREQ_HSE
  SetSysClockToHSE();
#elif defined SYSCLK_FREQ_24MHz
  SetSysClockTo24();
#elif defined SYSCLK_FREQ_36MHz
  SetSysClockTo36();
#elif defined SYSCLK_FREQ_48MHz
  SetSysClockTo48();
#elif defined SYSCLK_FREQ_56MHz
  SetSysClockTo56();
#elif defined SYSCLK_FREQ_72MHz
  SetSysClockTo72();
#endif

 /* If none of the define above is enable
    source (default after reset) */
}
```

실행 결과

```
----------------------
Mango test start ^)^
----------------------
Press menu key
----------------------
0> System Information
----------------------
1> LED Test
2> KEY Test
----------------------
x> quit

0 is selected

System_Information() S
RCC->CFGR : 0x11040A
SYSCLK_Frequency = 72000000
HCLK_Frequency = 72000000
PCLK1_Frequency = 36000000
PCLK2_Frequency = 72000000
ADCCLK_Frequency = 36000000
```