|  |
| --- |
| 임베디드 설계 및 실험 |
| 실험 결과 보고서 |
| 7주차 3조 |

|  |
| --- |
|  |

**실험 내용**

목차

[7주차 실험 Schematic 1](#_Toc54001636)

[Clock 설정을 파악하고, 지정된 Clock으로 설정하기 2](#_Toc54001637)

[1. void SetSysClock(void) 2](#_Toc54001638)

[UART를 설정하고, 지정된 Baud rate로 설정하기 4](#_Toc54001639)

[1. void RCC\_Enable(void) 4](#_Toc54001640)

[2. void PortConfiguration(void) 4](#_Toc54001641)

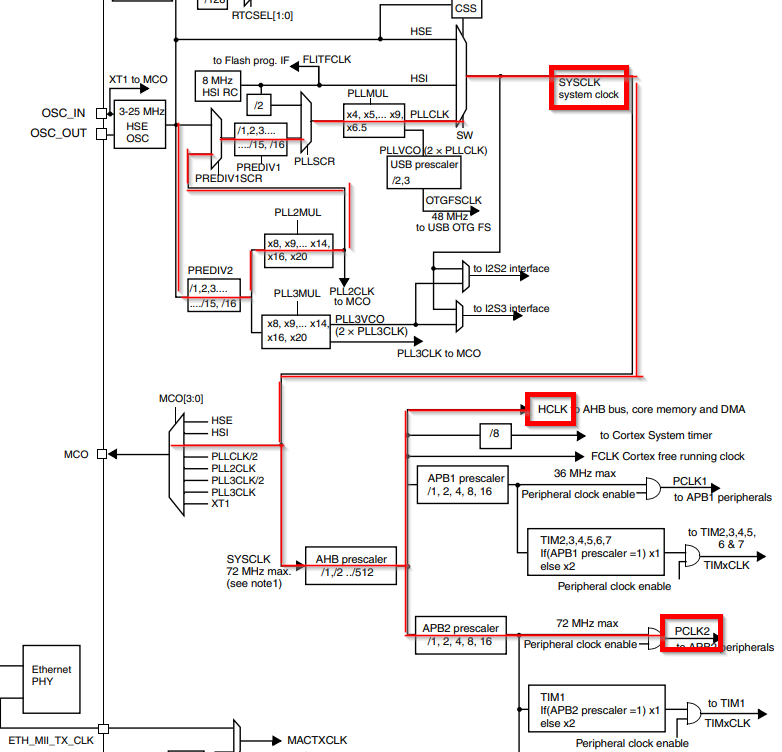
[3. void UartInit(void) 5](#_Toc54001642)

[Putty를 통해 데이터 출력 5](#_Toc54001643)

[실행 결과 5](#_Toc54001644)

## 7주차 실험 Schematic

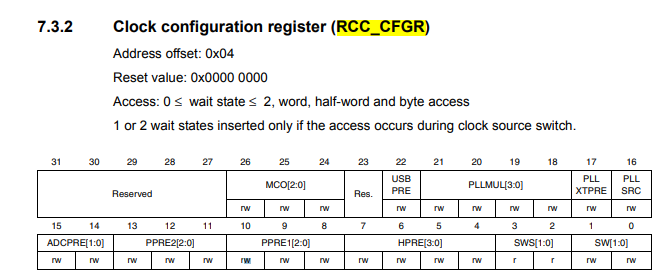
* System Clock(SYSCLK) 20MHz로 설정
* HCLK을 20MHz로 설정
* PCLK2을 20MHz로 설정



## Clock 설정을 파악하고, 지정된 Clock으로 설정하기

### void SetSysClock(void)

* PREDIV2에서 10으로 나누고 PLL2MUL에서 8을 곱한다. 그 뒤에 PREDIV1에서 6으로 나누고 PLLMUL에서 다시 6을 곱해 SYSCLK을 20MHz로 설정한다.
* MCO에서 SYSCLK 값을 선택하게 하고 HCLK와 PCLK2로 동일하게 20MHz를 사용하므로 prescaler는 div 1로 설정한다.

\

RCC->CFGR |= (uint32\_t)RCC\_CFGR\_HPRE\_DIV1;

: AHB prescaler를 나누지 않음. SYSCLK 그대로 사용

RCC->CFGR |= (uint32\_t)RCC\_CFGR\_PPRE2\_DIV1;

: AHB2 prescaler를 나누지 않음

RCC->CFGR |= (uint32\_t)RCC\_CFGR\_PPRE1\_DIV1;

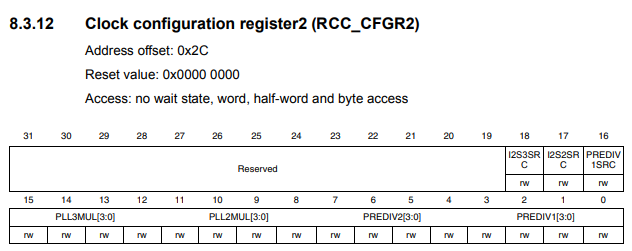
: AHB1 prescaler를 나누지 않음

RCC->CFGR &= (uint32\_t)~(RCC\_CFGR\_PLLXTPRE | RCC\_CFGR\_PLLSRC | RCC\_CFGR\_PLLMULL);

: 해당부분 값 0으로 초기화

RCC->CFGR |= (uint32\_t)(RCC\_CFGR\_PLLXTPRE\_PREDIV1 | RCC\_CFGR\_PLLSRC\_PREDIV1 | RCC\_CFGR\_PLLMULL6 | RCC\_CFGR\_MCO\_SYSCLK);

: PREDIV1 사용 설정 | PLLSRC에서 PREDIV1의 값 선택 | PLLMUL에서 6 곱하기 | MCO에서 SYSCLK 선택



RCC->CFGR2 &= (uint32\_t)~(RCC\_CFGR2\_PREDIV1 | RCC\_CFGR2\_PLL2MUL | RCC\_CFGR2\_PREDIV1 | RCC\_CFGR2\_PREDIV1SRC);

: 해당부분 값 0으로 초기화

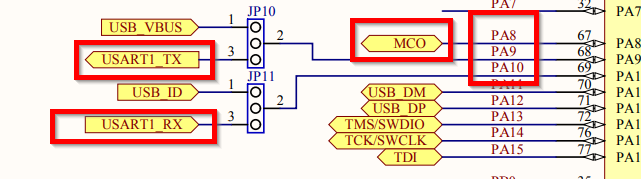
RCC->CFGR2 |= (uint32\_t)(RCC\_CFGR2\_PREDIV2\_DIV10 | RCC\_CFGR2\_PLL2MUL8 | RCC\_CFGR2\_PREDIV1SRC\_PLL2 | RCC\_CFGR2\_PREDIV1\_DIV6);

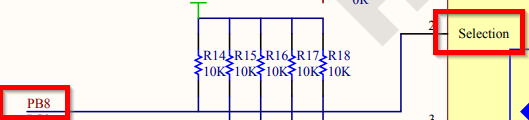
: PREDIV2 /10으로 설정 | PLLMUL2 \*8로 설정 | PREDIV1SRC가 PLL2 CLK을 선택 | PREDIV1을 /6으로 설정.

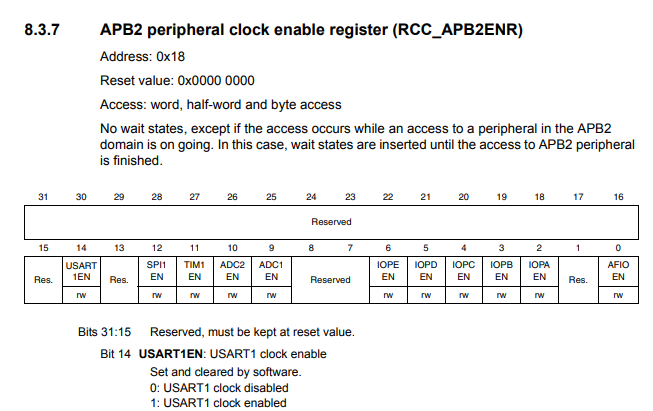
## UART를 설정하고, 지정된 Baud rate로 설정하기

### void RCC\_Enable(void)

UART 포트와 A,B clock enable







RCC->APB2ENR |= RCC\_APB2ENR\_IOPAEN;

RCC->APB2ENR |= RCC\_APB2ENR\_IOPBEN;

RCC->APB2ENR |= RCC\_APB2ENR\_USART1EN;

### void PortConfiguration(void)

MCO 및 UART TX: Alternate function output Push-pull 으로 설정

UART RX: Input with pull-up / pulldown 으로 설정

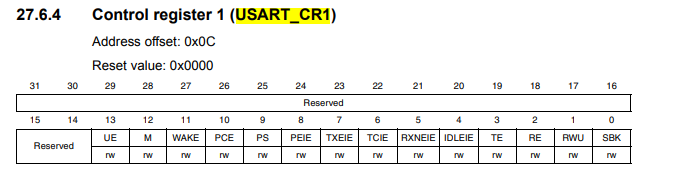
GPIOA->CRH |= 0xB;

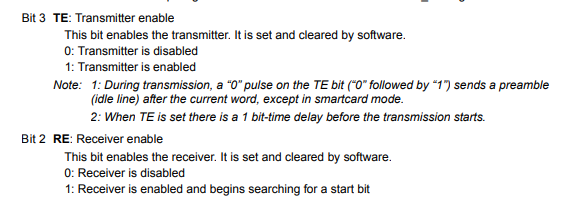
GPIOA->CRH |= 0x8B0;

GPIOB->CRH &= ~(GPIO\_CRH\_CNF8 );

GPIOB->CRH |= 0x8;

### void UartInit(void)





USART1->CR1 |= USART\_CR1\_TE | USART\_CR1\_RE;

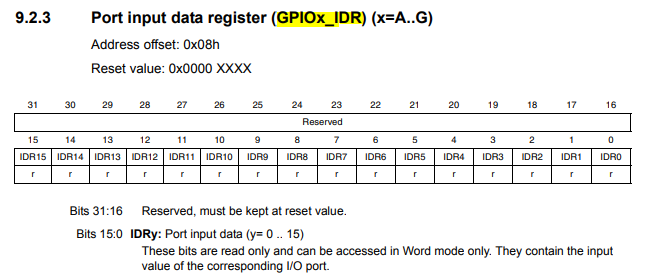
: Rx, Tx Enable

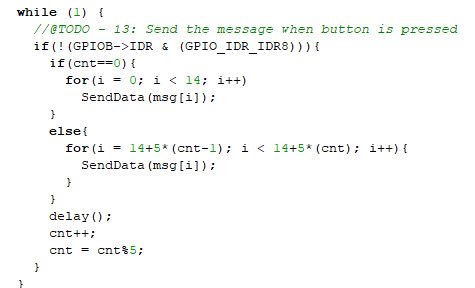
USART1->BRR |= 0xAE; // 강의 자료 27p 참고

USART1->CR1 |= USART\_CR1\_UE;

: enable USART

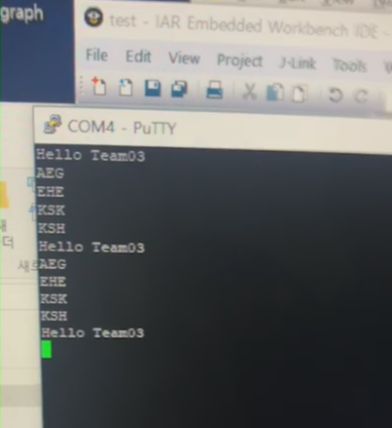
## Putty를 통해 데이터 출력





* Port B8 누를 때 값 전송.

## 실행 결과



### 전체 코드

