REPORT

**제목 : 8조 5주차 실험결과 보고서**

**수강과목 : 임베디드 시스템 설계 및 실험**

**조 원 : 201524455 문성욱**

**201524615 한광재**

**201524426 김민준**

**201524577 정주홍**

**제출일자 : 2019.10.10**

1. **실험목표**
   1. Clock Tree의 이해 및 사용자 Clock 설정
   2. UART 통신의 원리를 배우고, 실제 설정 방법 파악
2. **배경지식**
3. Clock Tree

STM32의 내부 Clock의 흐름

HSI, HSE, PPL의 출력중 하나를 사용하여 System Clock을 출력

PLL은 HSI와 HSE를 곱하거나 나누어서 원하는 주파수를 출력

1. UART( Universal Asynchronous Receiver/Transmitter)  
   비동기 통신 프로토콜

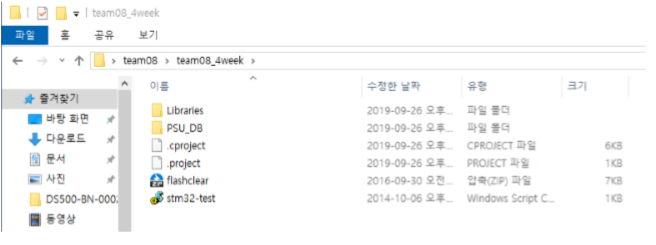
Rx(데이터 수신)와 Tx(데이터 송신)의 교차 연결

서로의 Ground를 연결

baud rate를 일치시켜야 통신 가능

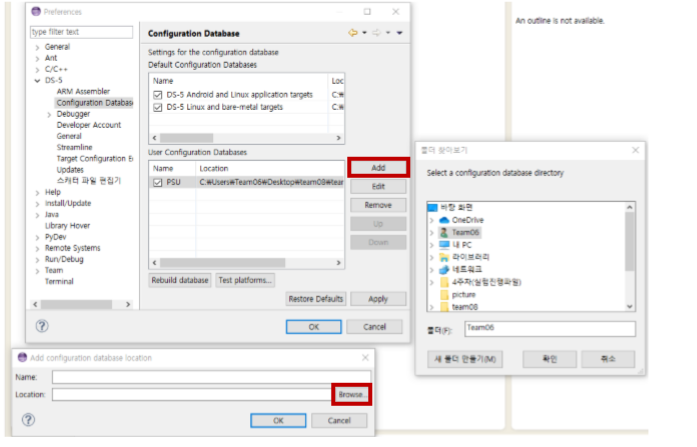
1. **실험방법**
2. 프로젝트 생성

Eclipse for DS-5 프로젝트 생성 후 DB파일, 라이브러리, 스캐터 파일 등 프로젝트 폴더로 복사



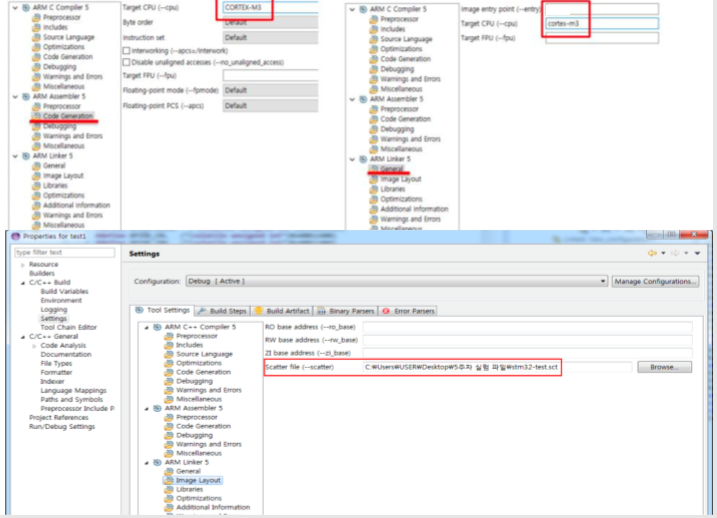
1. 데이터베이스 추가

Window -> preference -> Configuration Database -> Add에서 PSU\_DB를 선택 -> Rebuild database -> OK



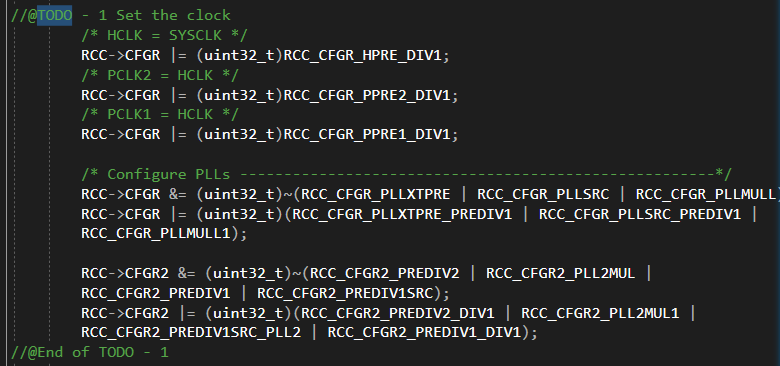
1. Scatter file 추가

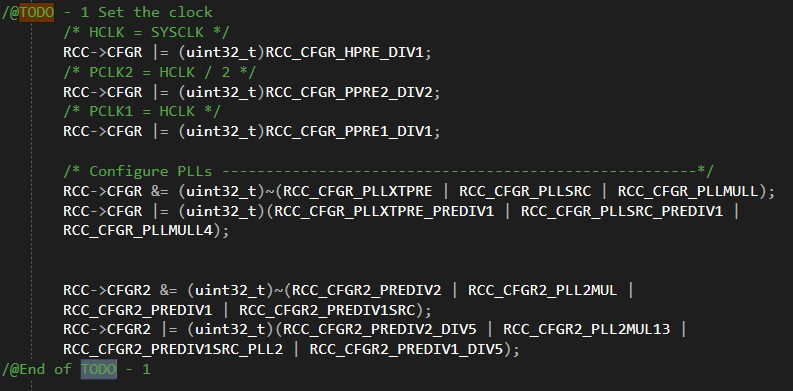
Properties -> C/C++ Build -> Settings 에서 스캐터 파일 추가

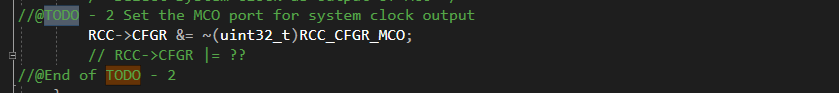


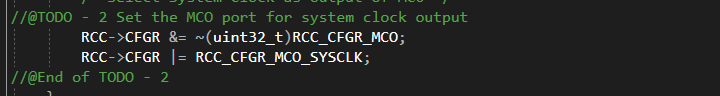
1. 주어진 Template.c 코드 작성

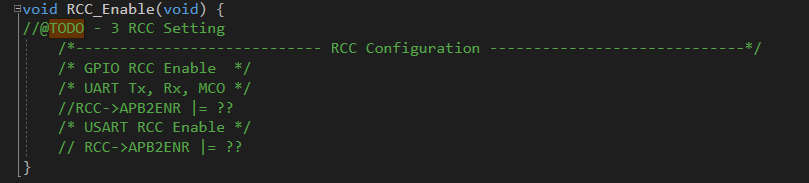
todo 부분의 코드를 수정

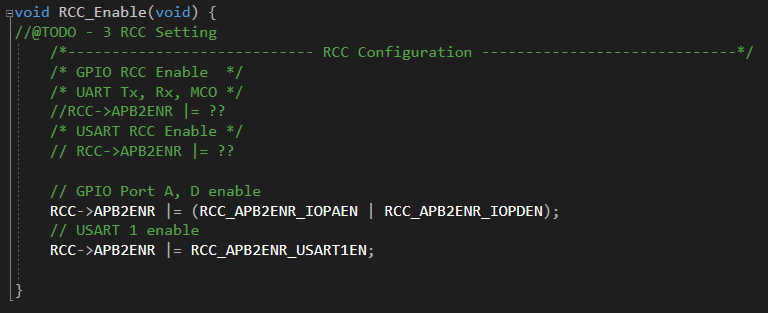


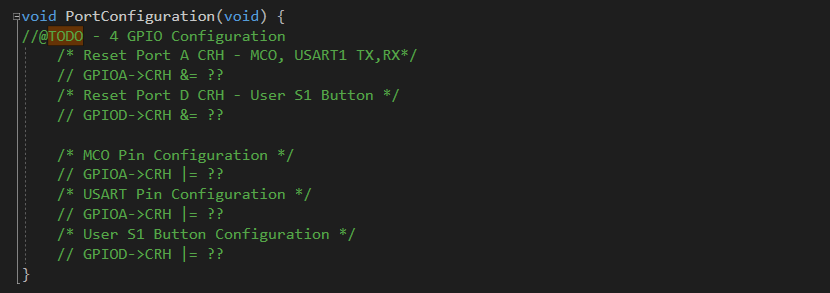


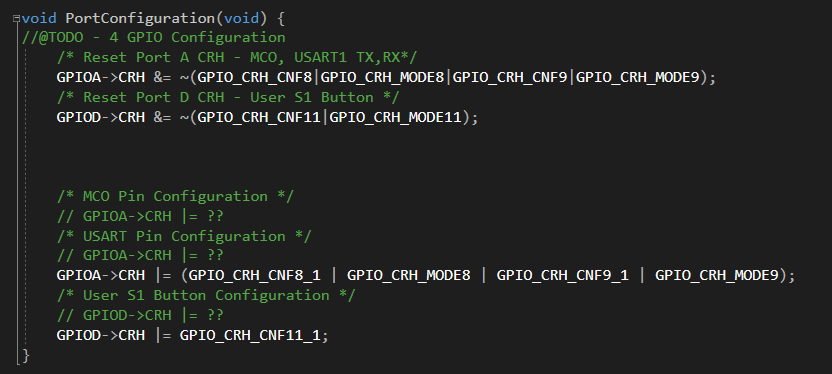


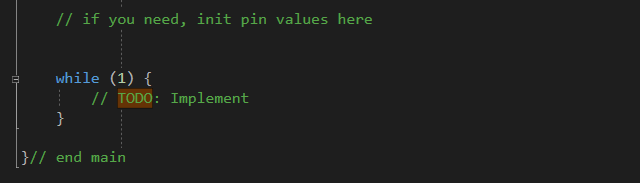


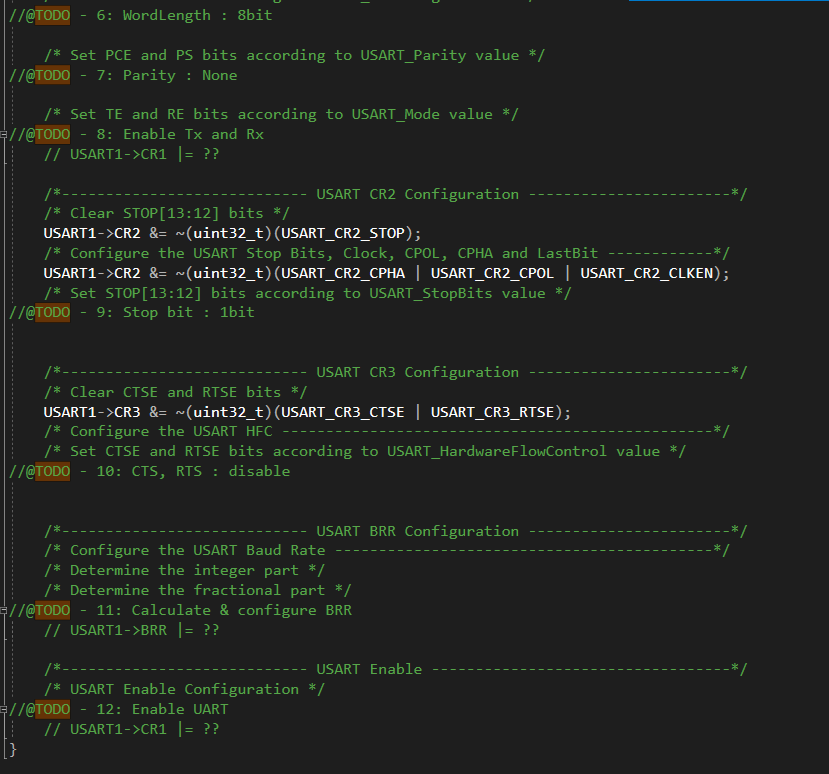
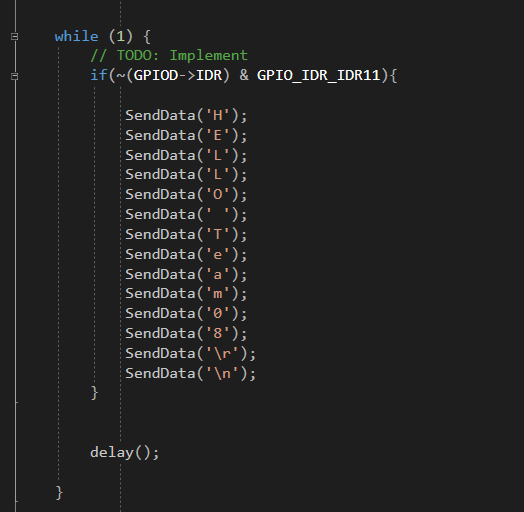


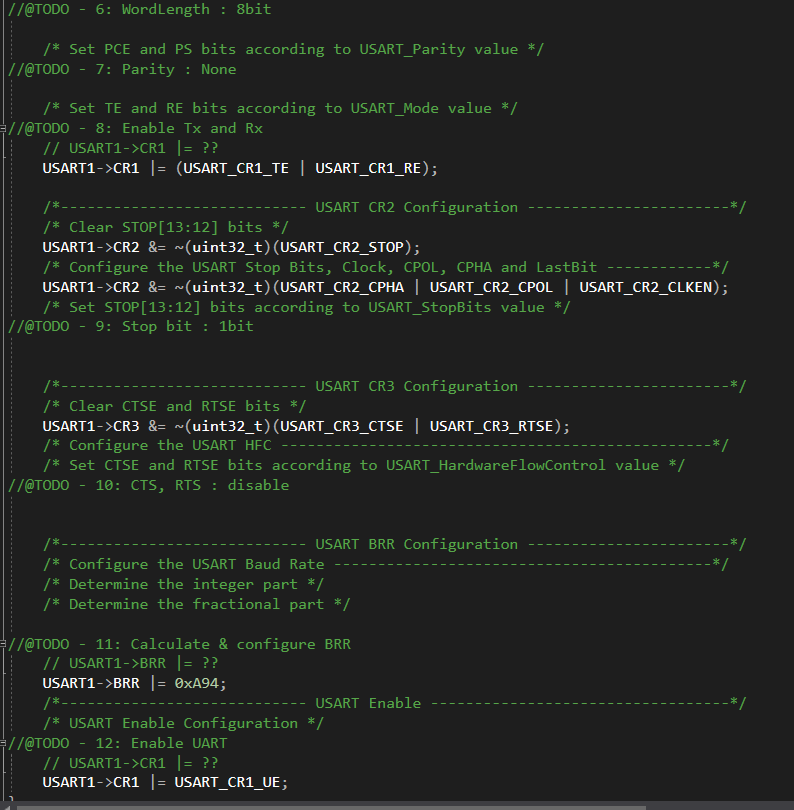




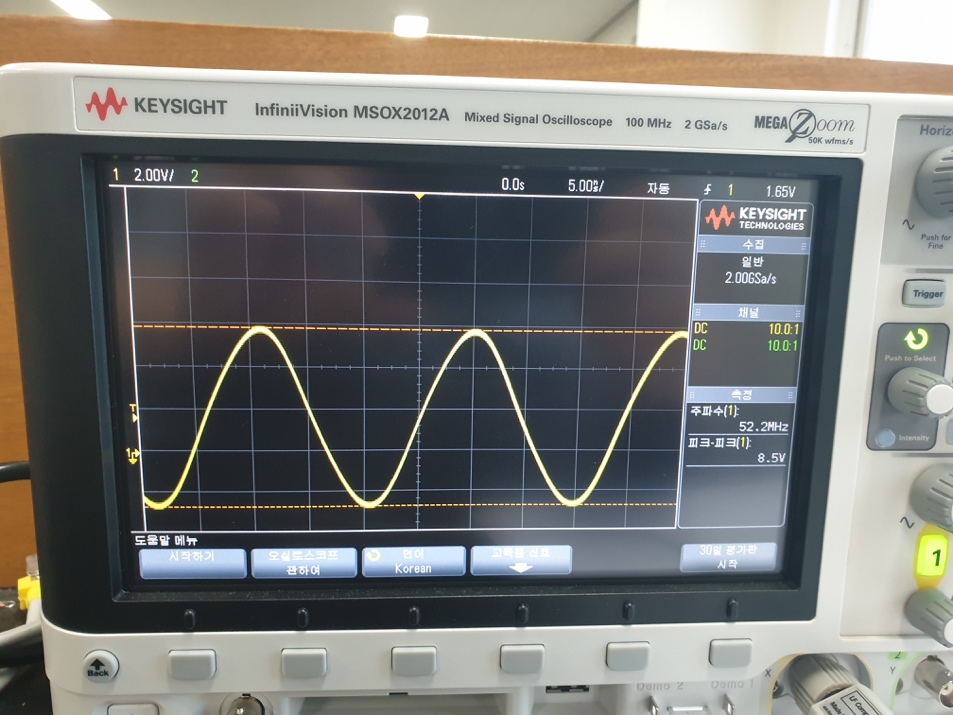






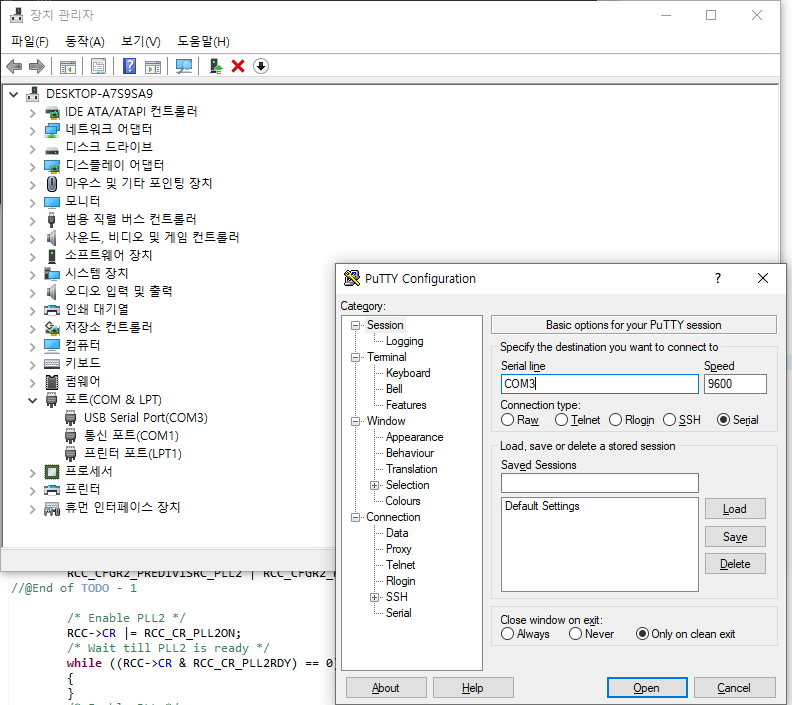


1. 오실로스코프에 PA08과 GND를 연결하여 주파수 출력



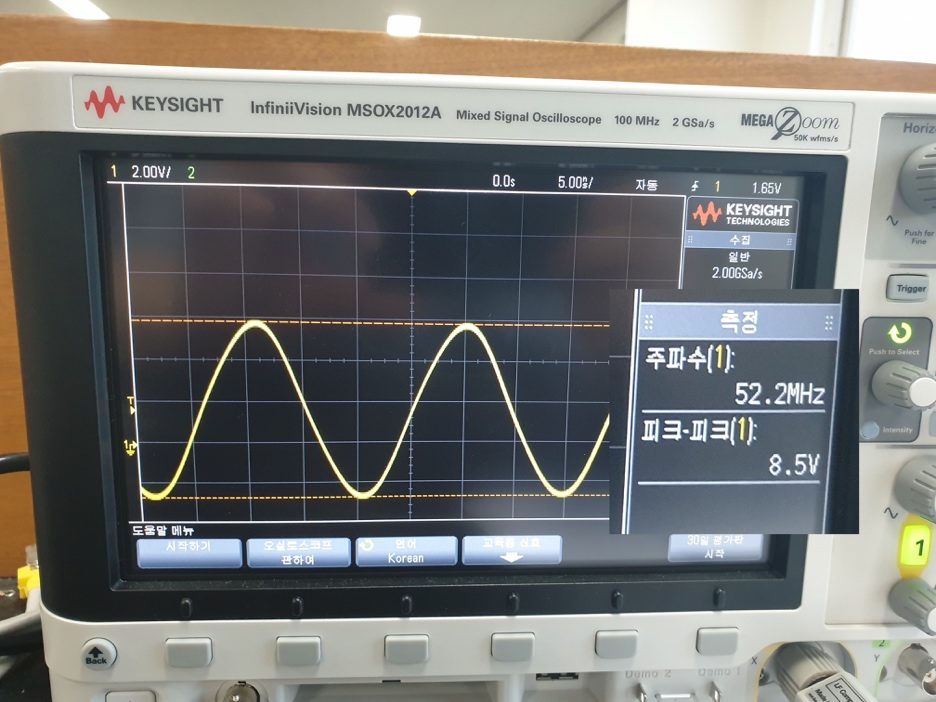
1. PuTTY 실행

pc 장치 관리자에 보드와 연결된 serial port를 확인하고 Connection Type을 serial로 선택하고 port와 Baud rate를 입력 후 Open을 클릭



1. **실험 결과**

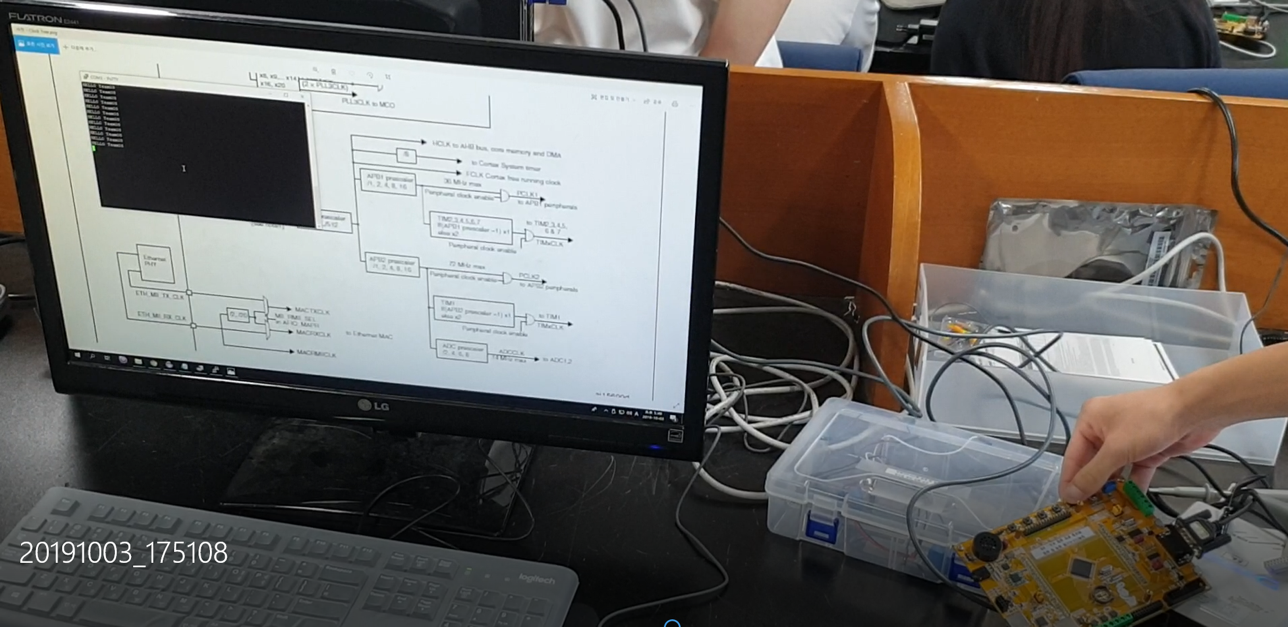
**A. 결과**



SYSCLK의 주파수를 측정한 오실로스코프 화면으로 52MHz가 출력됩니다.



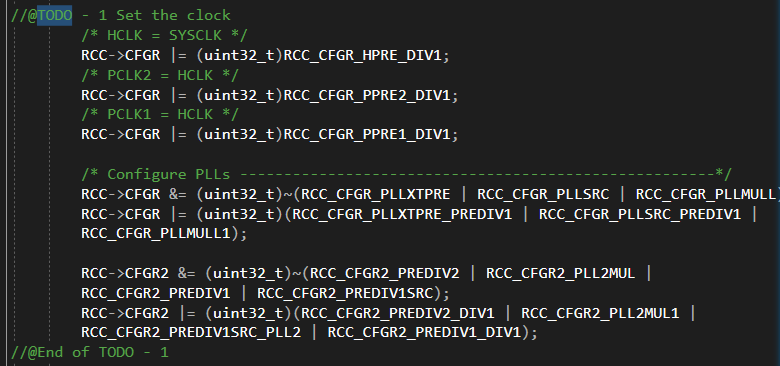
버튼을 누를 때마다 Putty를 통해 Hello Team08을 출력.



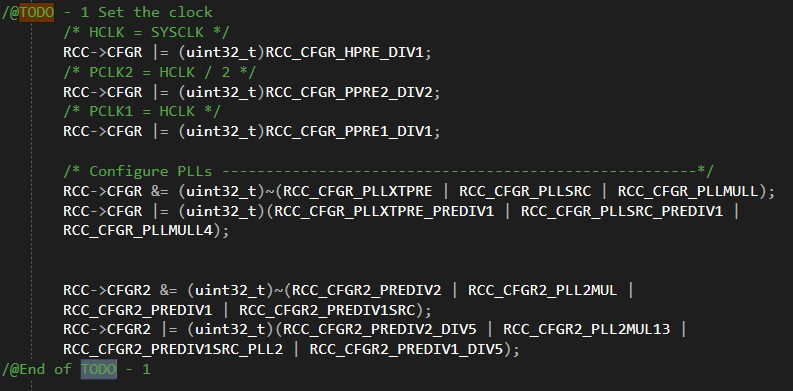
전체 화면

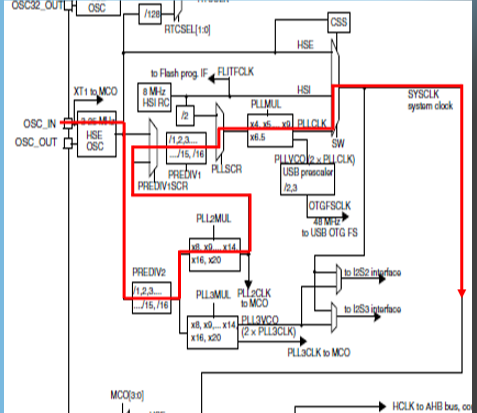
**B. 코드 분석**

수정 전



수정 후





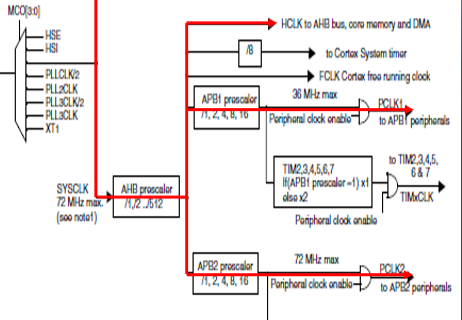
설정하고자 하는 SYSCLK이 52MHz이고 HSE OSC에서 25MHz가 나오므로 PREDIV2에서 /5를 PLL2MUL에서 \*13을 PREDIV1에서 /5를 PLLMUL에서 \*4를 계산해 주어야 하므로 주어진 template.c의

RCC->CFGR2 |= (uint32\_t)(RCC\_CFGR2\_PREDIV2\_DIV1 | RCC\_CFGR2\_PLL2MUL1 |

RCC\_CFGR2\_PREDIV1SRC\_PLL2 | RCC\_CFGR2\_PREDIV1\_DIV1);를

RCC->CFGR2 |= (uint32\_t)(RCC\_CFGR2\_PREDIV2\_DIV5 | RCC\_CFGR2\_PLL2MUL13 |

RCC\_CFGR2\_PREDIV1SRC\_PLL2 | RCC\_CFGR2\_PREDIV1\_DIV5);로 수정합니다.

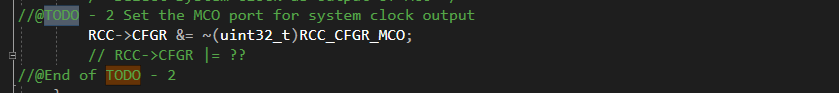


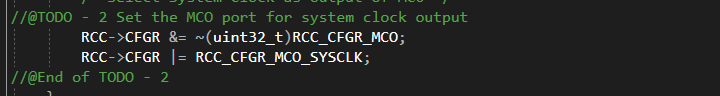
PCLK2로 26MHz를 내보내야 하므로

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

RCC->CFGR |= (uint32\_t)RCC\_CFGR\_PPRE2\_DIV2;로 수정합니다.

수정 전

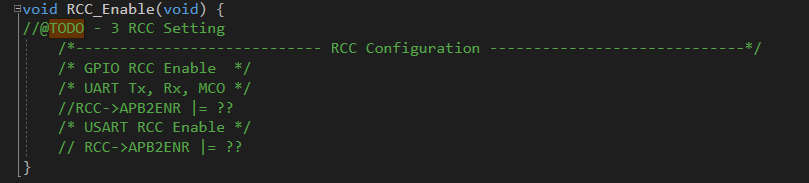
  
수정 후



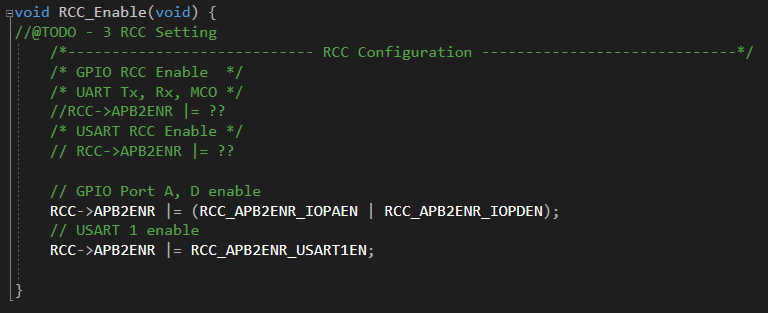
MCO port를 SYSCLK의 output으로 set하기 위해

RCC->CFGR |= RCC\_CFGR\_MCO\_SYSCLK; 를 추가합니다.

수정 전



수정 후

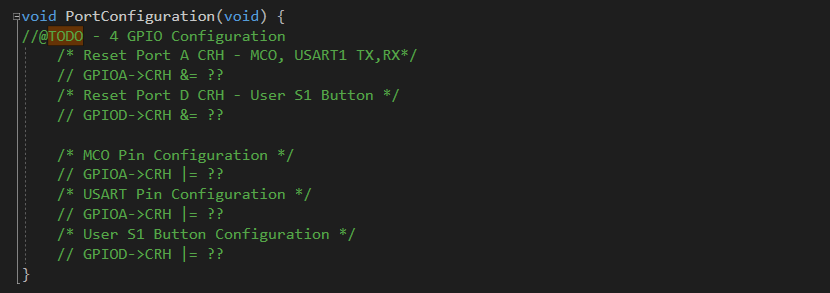


SYSCLK를 출력할 Port A와 버튼 입력을 받을 Port D 를 통신을 위해 USART 1을 enable 시키기 위해서

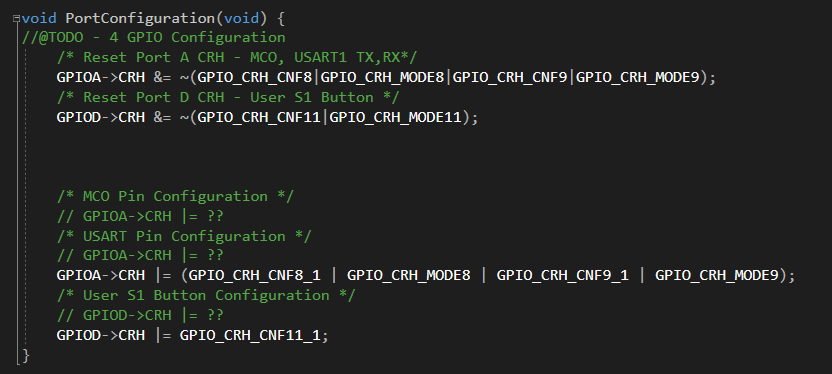
RCC->APB2ENR |= (RCC\_APB2ENR\_IOPAEN | RCC\_APB2ENR\_IOPDEN);

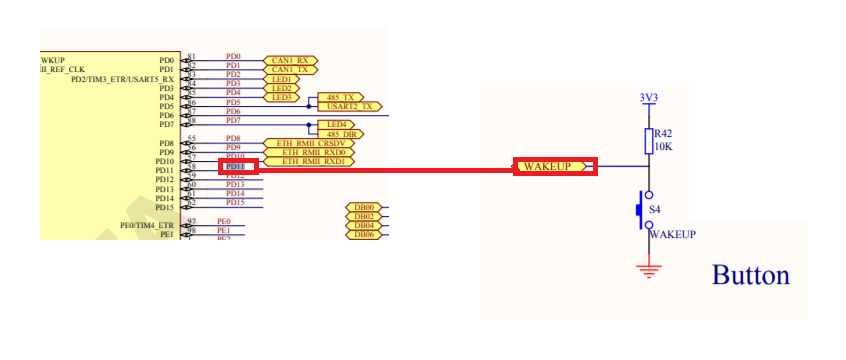
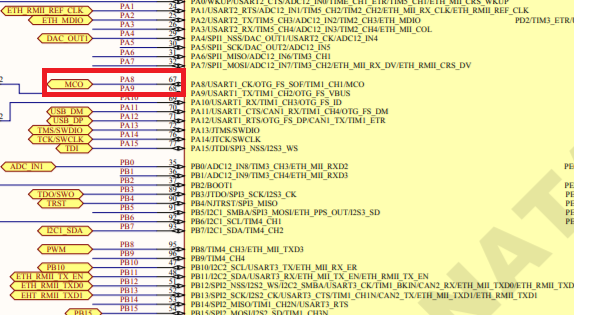
RCC->APB2ENR |= RCC\_APB2ENR\_USART1EN; 를 추가합니다.

수정 전



수정 후





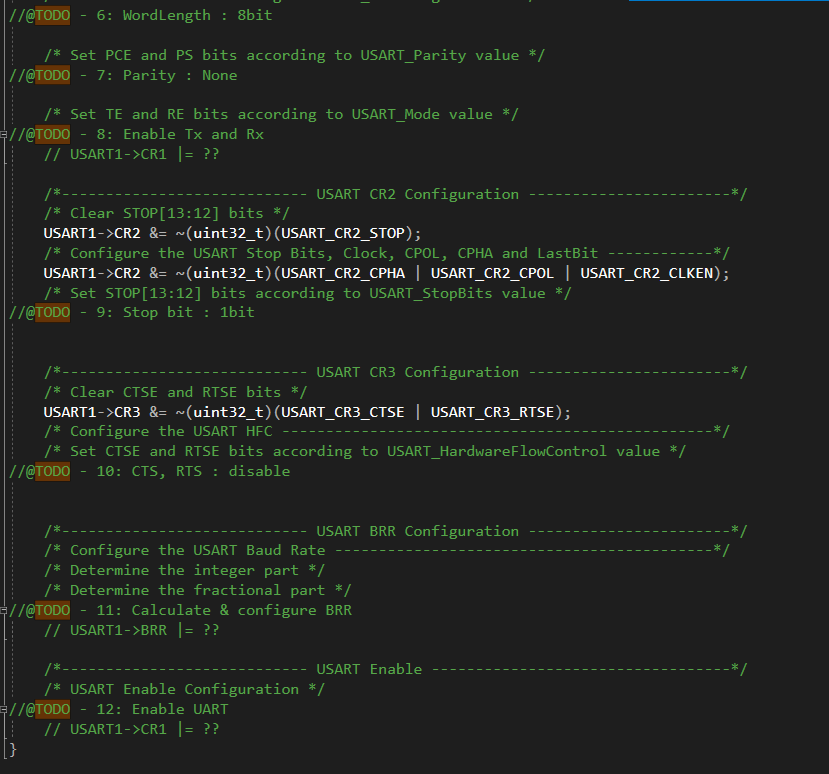
S1버튼의 input 값을 받아드리고 MCO의 output 값을 내보내기 위해서

A와 D PORT를 reset하는

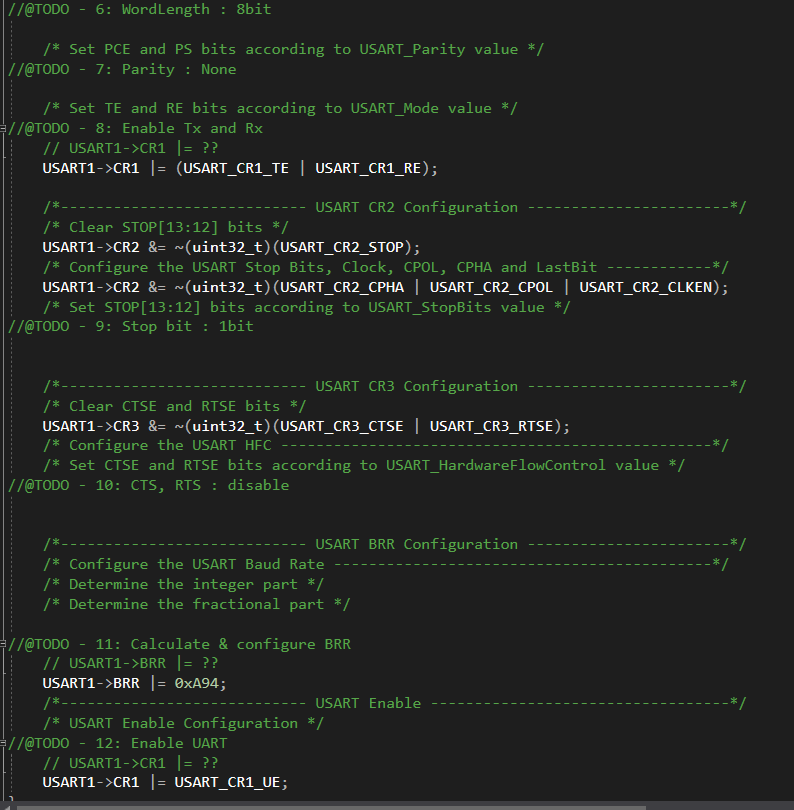
GPIOA->CRH &= ~(GPIO\_CRH\_CNF8|GPIO\_CRH\_MODE8|GPIO\_CRH\_CNF9|GPIO\_CRH\_MODE9);

GPIOD->CRH &= ~(GPIO\_CRH\_CNF11|GPIO\_CRH\_MODE11);를 추가합니다.

수정 전



수정 후



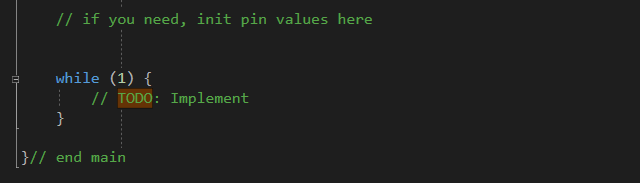
Todo 11에서

이므로 USART1->BRR |= 0xA94;를 추가합니다.

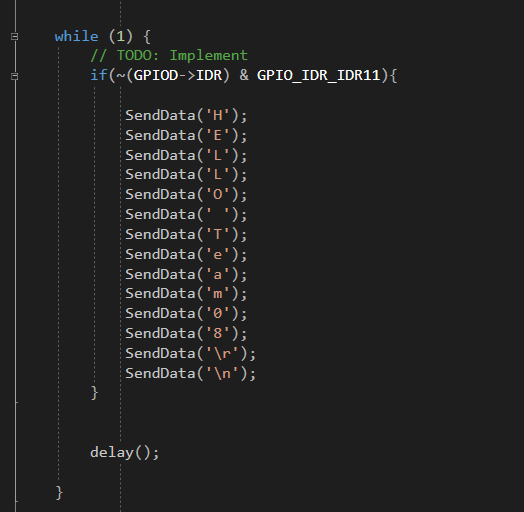
Todo 12에서 UART를 enable하기 위해서

USART1->CR1 |= USART\_CR1\_UE;를 추가합니다.

수정 전



수정 후



버튼을 눌러주는 동안 “HELLO Team08”을 보내주어야 하므로 위와 같이 while문을 채웠습니다.

1. **결론**

Clock tree의 그림을 보고 이를 이용하여 SYSCLK과 PCLK2를 직접 설정하여 주고 Baud Rate 식을 이용하여 UART통신을 하기 위한 코드를 직접 짜봄으로써 Clock과 UART에 대한 이해와 지식이 더욱 늘어났습니다.