

2019.10.02

Embedded Network
Systems Lab.

Minjae Kim
min7ae@gmail.com

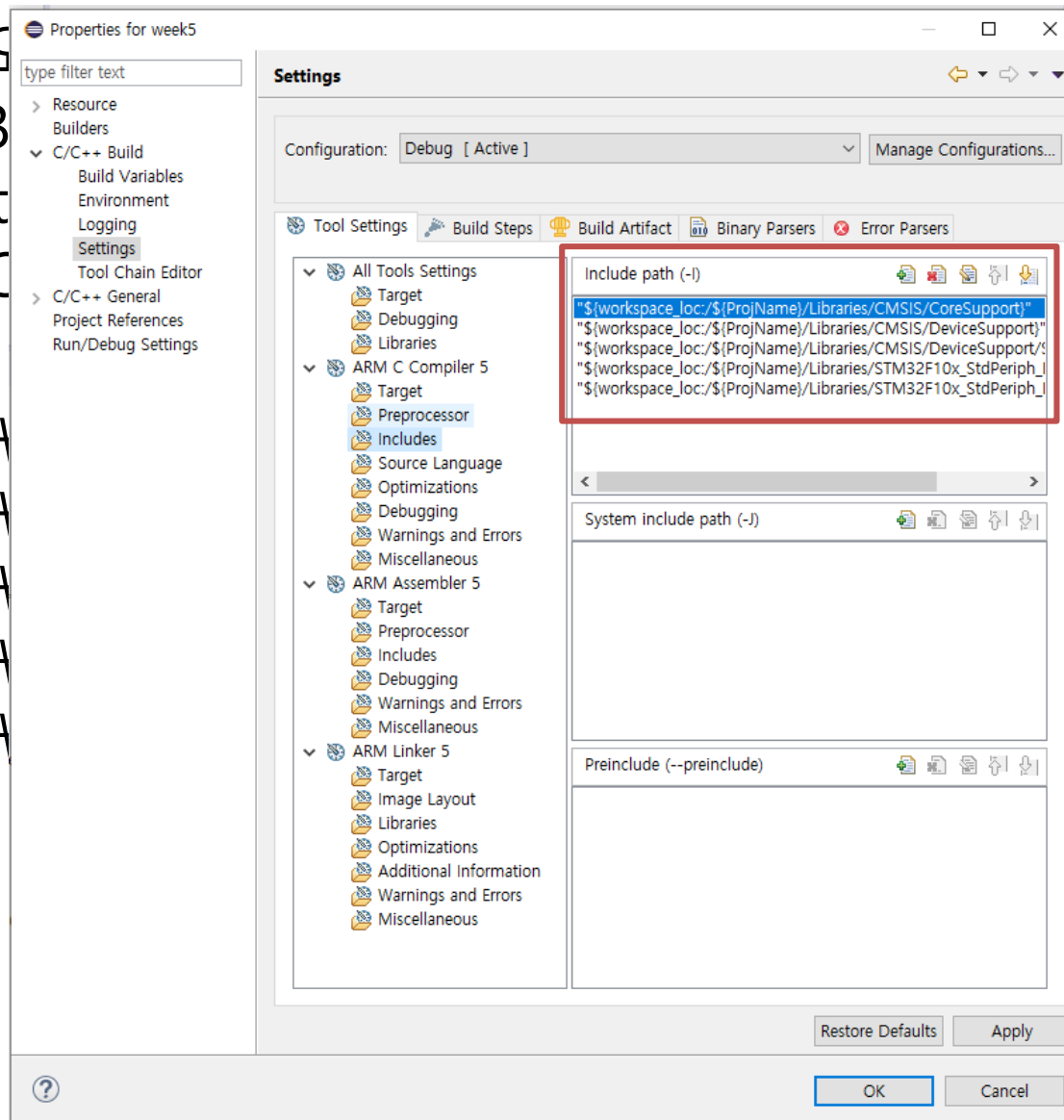
임베디드 시스템 설계 및 실험 5주차

- Project 오른쪽 클릭 > properties
- C/C++ Build > Settings
- Tool Settings
- ARM C Compiler 5 > Includes에서 library 파일 추가
- Libraries\CMSIS\CoreSupport
- Libraries\CMSIS\DeviceSupport
- Libraries\CMSIS\DeviceSupport\Startup
- Libraries\STM32F10x_StdPeriph_Driver_v3.5\inc
- Libraries\STM32F10x_StdPeriph_Driver_v3.5\src

Library 사용(2)

- Project S
- C/C++ B
- Tool Sett
- ARM C C

- Libraries\
- Libraries\
- Libraries\
- Libraries\
- Libraries\



추가

Library 사용(3)

임베디드시스템설계및실험_수요일_5주차_템플릿.c 일부

```
20
21 void RCC_Enable(void) {
22     //@TODO
23     /*----- RCC Configuration -----*/
24     /* GPIO RCC Enable */
25     /* UART Tx, Rx, MCO */
26     RCC->APB2ENR |= (uint32_t)(RCC_APB2ENR_IOPAEN | RCC_APB2ENR_IOPBEN);
27     Macro Expansion
28     ((RCC_TypeDef *) (((uint32_t)0x40000000) + 0x20000) + 0x1000)
29 }
30 Press F2 for macro expansion steps
```

stm32_ReferenceManual.pdf p51

| | | | |
|-------------|-------------|-----------------------------|----------------------------|
| 0x4002 1000 | 0x4002 13FF | Reset and clock control RCC | Section 7.3.11 on page 121 |
|-------------|-------------|-----------------------------|----------------------------|

Library 사용(4)

임베디드시스템설계및실험_수요일_5주차_템플릿.c 일부

```

20
21 void RCC_Enable(void) {
22     // @TODO
23     /*----- RCC Configuration -----*/
24     /* GPIO RCC Enable */
25     /* UART Tx, Rx, MCO */
26     RCC->APB2ENR |= (uint32_t)(RCC_APB2ENR_IOPAEN | RCC_APB2ENR_IOPBEN);
27     Macro Expansion
28     ((RCC_TypeDef *) (((uint32_t)0x40000000) + 0x20000) + 0x1000)
29 }
30

```

Press "F2" for macro expansion steps

stm32f10x.h 일부

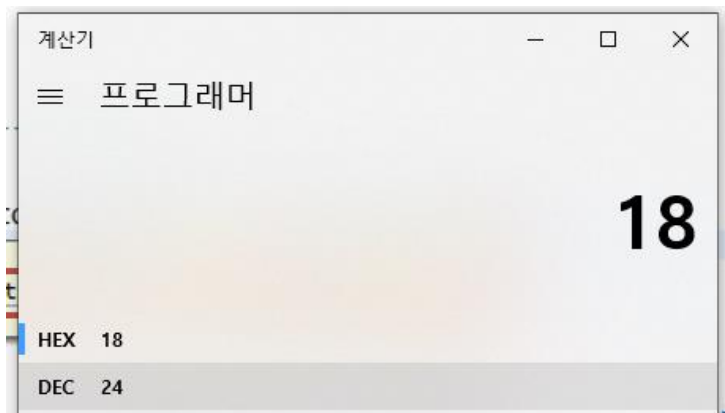
stm32_ReferenceManual.pdf p121, p156 참고

| Offset | Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 |
|--------|-------------|----------|----|----|----|----|----|----|----|----|----|---------|---------|--------|----------|----|----|--------|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 0x18 | RCC_APB2ENR | Reserved | | | | | | | | | | TIM11EN | TIM10EN | TIM9EN | Reserved | | | ADC3EN | USART1EN | TIM8EN | SPH1EN | TIM7EN | ADC2EN | ADC1EN | IOPGEN | IOPFEN | IOPEEN | IOPDEN | IOPCEN |
| | Reset value | | | | | | | | | | | 0 | 0 | 0 | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

```

16 #define __IO volatile
1076 typedef struct
1077 {
1078     __IO uint32_t CR;
1079     __IO uint32_t CFGR; ← 한칸 +0x04
1080     __IO uint32_t CIR;
1081     __IO uint32_t APB2RSTR;
1082     __IO uint32_t APB1RSTR;
1083     __IO uint32_t AHBENR;
1084     __IO uint32_t APB2ENR; // +0x18
1085     __IO uint32_t APB1ENR; (+24)
1086     __IO uint32_t BDCR;
1087     __IO uint32_t CSR;
1088
1089 #ifndef STM32F10X_CL
1090     __IO uint32_t AHBSTR;
1091     __IO uint32_t CFGR2;
1092 #endif /* STM32F10X_CL */
1093
1094 #if defined (STM32F10X_LD_V
1095     uint32_t RESERVED0;
1096     __IO uint32_t CFGR2;
1097 #endif /* STM32F10X_LD_VL
1098 } RCC_TypeDef;

```



Library 사용(5)

임베디드시스템설계및실험_수요일_5주차_템플릿.c 일부

```
20
21 void RCC_Enable(void) {
22     //@TODO
23     /*----- RCC Configuration -----*/
24     /* GPIO RCC Enable */
25     /* UART Tx, Rx, MCO */
26     RCC->APB2ENR |= (uint32_t)(RCC_APB2ENR_IOPAEN | RCC_APB2ENR_IOPBEN);
27     Macro Expansion
28     ((RCC_TypeDef *) (((uint32_t)0x40000000) + 0x20000) + 0x1000)
29 }
30
```

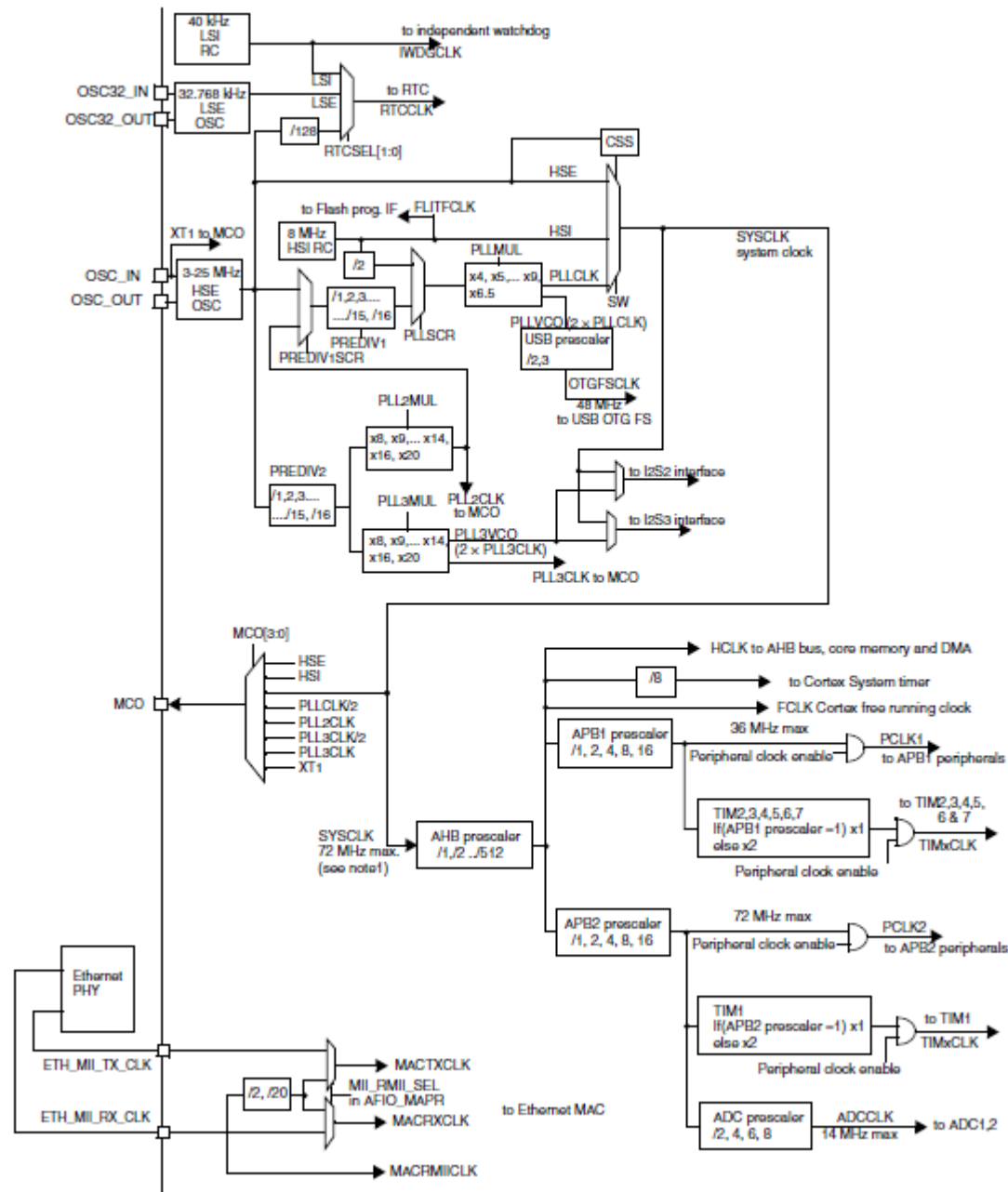
HEX 8
DEC 8
OCT 10
BIN 1000

stm32_ReferenceManual.pdf p121

| Offset | Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------------|----------|----|----|----|----|----|----|----|----|----|----|---------|---------|--------|----------|--------|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|----------|--------|---|
| 0x18 | RCC_APB2ENR | Reserved | | | | | | | | | | | TIM11EN | TIM10EN | TIM9EN | Reserved | ADC3EN | USART1EN | TIM8EN | SP11EN | TIM1EN | ADC2EN | ADC1EN | IOPGEN | IOPFEN | IOPEEN | IOPDEN | IOPCEN | IOPBEN | IOPAEN | Reserved | AFIOEN | |
| | Reset value | | | | | | | | | | | | 0 | 0 | 0 | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

```
2044 /***** Bit definition for RCC_APB2ENR register *****/
2045 #define RCC_APB2ENR_AFIOEN ((uint32_t)0x00000001) /*!< Function I/O clock enable */
2046 #define RCC_APB2ENR_IOPAEN ((uint32_t)0x00000004) /*!< I/O port A clock enable */
2047 #define RCC_APB2ENR_IOPBEN ((uint32_t)0x00000008) /*!< I/O port B clock enable */
2048 #define RCC_APB2ENR_IOPCEN ((uint32_t)0x00000010) /*!< I/O port C clock enable */
2049 #define RCC_APB2ENR_IOPDEN ((uint32_t)0x00000020) /*!< I/O port D clock enable */
2050 #define RCC_APB2ENR_ADC1EN ((uint32_t)0x00000200) /*!< ADC 1 interface clock enable */
2051
```

HEX 4
DEC 4
OCT 4
BIN 0100



System Clock Generate

```
//@TODO - 1 Set the clock
/* HCLK = SYSCLK */
RCC->CFGR |= (uint32_t)RCC_CFGR_HPRE_DIV1;
/* PCLK2 = HCLK */
RCC->CFGR |= (uint32_t)RCC_CFGR_PPRE2_DIV1;
/* PCLK1 = HCLK */
RCC->CFGR |= (uint32_t)RCC_CFGR_PPRE1_DIV1;

/* Configure PLLs -----*/
RCC->CFGR &= (uint32_t)~(RCC_CFGR_PLLXTPRE | RCC_CFGR_PLLSRC | RCC_CFGR_PLLMULL);
RCC->CFGR |= (uint32_t)(RCC_CFGR_PLLXTPRE_PREDIV1 | RCC_CFGR_PLLSRC_PREDIV1 |
RCC_CFGR_PLLMULL1);

RCC->CFGR2 &= (uint32_t)~(RCC_CFGR2_PREDIV2 | RCC_CFGR2_PLL2MUL |
RCC_CFGR2_PREDIV1 | RCC_CFGR2_PREDIV1SRC);
RCC->CFGR2 |= (uint32_t)(RCC_CFGR2_PREDIV2_DIV1 | RCC_CFGR2_PLL2MUL1 |
RCC_CFGR2_PREDIV1SRC_PLL2 | RCC_CFGR2_PREDIV1_DIV1);

//@End of TODO - 1

/* Enable PLL2 */
RCC->CR |= RCC_CR_PLL2ON;
/* Wait till PLL2 is ready */
while ((RCC->CR & RCC_CR_PLL2RDY) == 0)
{
}
/* Enable PLL */
RCC->CR |= RCC_CR_PLLON;
/* Wait till PLL is ready */
while ((RCC->CR & RCC_CR_PLLRDY) == 0)
{
}
/* Select PLL as system clock source */
RCC->CFGR &= (uint32_t)((uint32_t)~(RCC_CFGR_SW));
RCC->CFGR |= (uint32_t)RCC_CFGR_SW_PLL;
/* Wait till PLL is used as system clock source */
while ((RCC->CFGR & (uint32_t)RCC_CFGR_SWS) != (uint32_t)0x08)
{
}

/* Select System Clock as output of MCO */
//@TODO - 2 Set the MCO port for system clock output
RCC->CFGR &= ~(uint32_t)RCC_CFGR_MCO;
// RCC->CFGR |= ??

//@End of TODO - 2
```

**void SetSysClock(void) 내부 일부분
=> 고쳐서 목적 Clock을 생성**

Rcc & GPIO Configure

```
void RCC_Enable(void) {
//@TODO - 3 RCC Setting
/*----- RCC Configurati
/* GPIO RCC Enable */
/* UART Tx, Rx, MCO */
//RCC->APB2ENR |= ??
/* USART RCC Enable */
// RCC->APB2ENR |= ??
}

void GPIO_Configuration(void) {
//@TODO - 4 GPIO Configuration
/* Reset Port A CRH */
// GPIOA->CRH &= ??
/* Reset Port B CRH */
// GPIOB->CRH &= ??

/* MCO Pin Configuration */
// GPIOA->CRH |= ??
/* USART Pin Configuration */
// GPIOA->CRH |= ??
/* Joy-stick SELECT Button Configuration */
// GPIOB->CRH |= ??
}

void GPIO_Pin_InitSetting(void) {
//@TODO - 5 GPIO Set & Reset
/* Enable MCO */
// GPIOA->BSRR |= ??
/* Joy-stick Button Reset */
// GPIOB->BRR |= ??
}
```

RCC 설정
GPIO 설정 (AFIO!)
GPIO Pin 값 설정

UART Configure

```
void UartInit(void) {
    /*----- USART CR1 Configuration -----*/
    /* Clear M, PCE, PS, TE and RE bits */
    USART1->CR1 &= ~(uint32_t)(USART_CR1_M | USART_CR1_PCE | USART_CR1_PS | USART_CR1_TE | USART_CR1_RE);
    /* Configure the USART Word Length, Parity and mode -----*/
    /* Set the M bits according to USART_WordLength value */
    //@TODO - 6: WordLength : 8bit

    /* Set PCE and PS bits according to USART_Parity value */
    //@TODO - 7: Parity : None

    /* Set TE and RE bits according to USART_Mode value */
    //@TODO - 8: Enable Tx and Rx
    // USART1->CR1 |= ??

    /*----- USART CR2 Configuration -----*/
    /* Clear STOP[13:12] bits */
    USART1->CR2 &= ~(uint32_t)(USART_CR2_STOP);
    /* Configure the USART Stop Bits, Clock, CPOL, CPHA and LastBit -----*/
    USART1->CR2 &= ~(uint32_t)(USART_CR2_CPHA | USART_CR2_CPOL | USART_CR2_CLKEN);
    /* Set STOP[13:12] bits according to USART_StopBits value */
    //@TODO - 9: Stop bit : 1bit

    /*----- USART CR3 Configuration -----*/
    /* Clear CTSE and RTSE bits */
    USART1->CR3 &= ~(uint32_t)(USART_CR3_CTSE | USART_CR3_RTSE);
    /* Configure the USART HFC -----*/
    /* Set CTSE and RTSE bits according to USART_HardwareFlowControl value */
    //@TODO - 10: CTS, RTS : disable

    /*----- USART BRR Configuration -----*/
    /* Configure the USART Baud Rate -----*/
    /* Determine the integer part */
    /* Determine the fractional part */
    //@TODO - 11: Calculate & configure BRR
    // USART1->BRR |= ??

    /*----- USART Enable -----*/
    /* USART Enable Configuration */
    //@TODO - 12: Enable UART
    // USART1->CR1 |= ??
}
```

//@TODO
부분을 완성하여서
UART 코드 작성

Reference PDF를 보면서
적절한 값 세팅

1. Eclipse for DS-5 실행 후 프로젝트 생성
2. DB파일, 라이브러리, 스캐터 파일, flashclear 파일
프로젝트 폴더 안으로 복사
3. 프로젝트 및 DS-5 설정 진행

**1. 각 조별 목적 Clock MCO로 출력 -> 오실로스코프로 Hz 검사
(코드를 완성하고 손을 들어 검사를 한 후 진행)**

2. UART 통신을 이용하여 Putty로 출력

ex) 처음 Select Button Click시, "Hello TeamX"

두번째 Select Button Click시, "AAA" (조원 이니셜1)

세번째 Select Button Click시, "BBB" (조원 이니셜2)

네번째 Select Button Click시, "CCC" (조원 이니셜3)

: