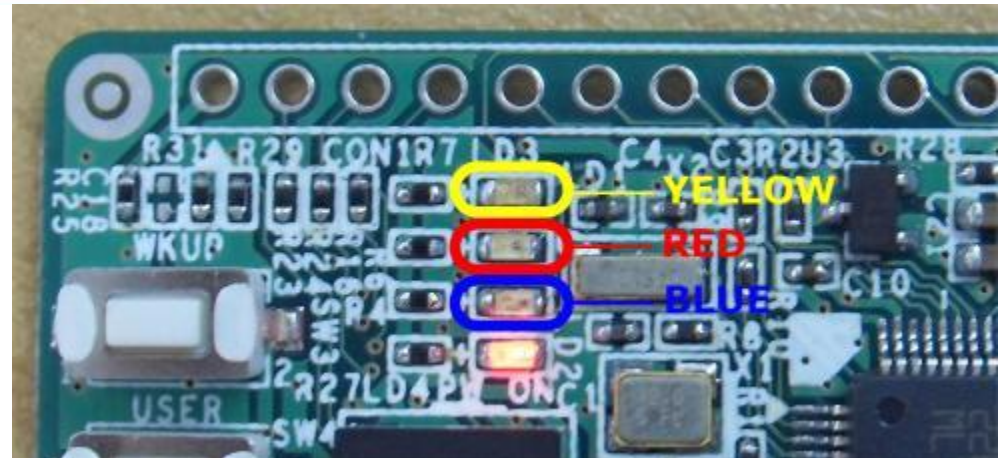
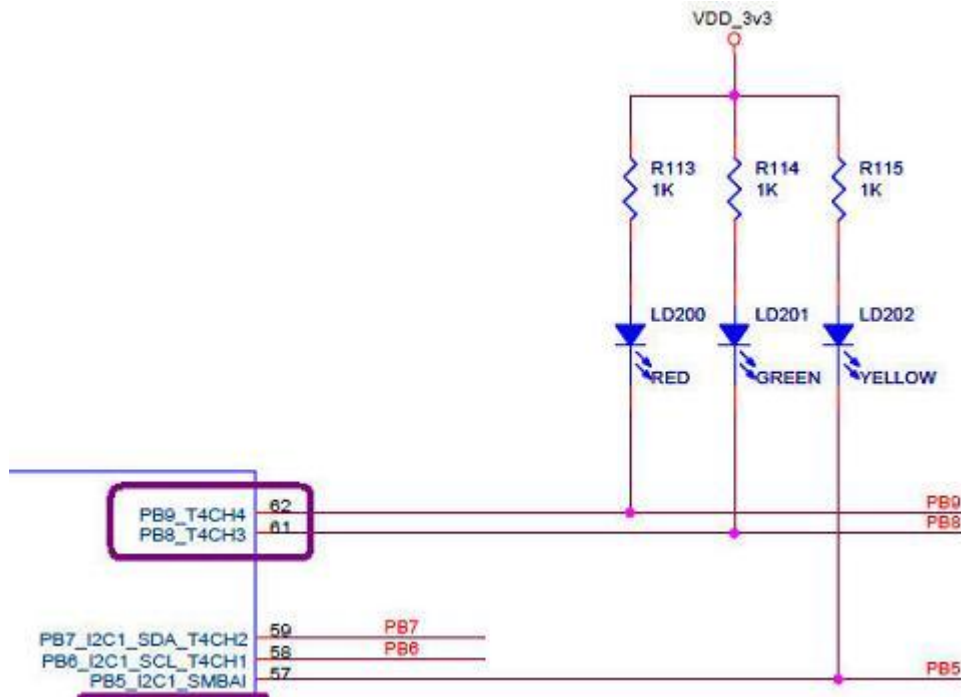


Led On Off 실습

2009.11.20

LED 회로 및 보드 위치

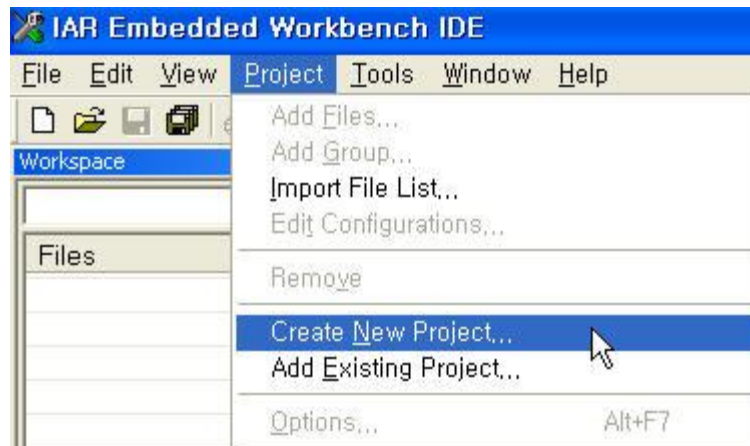
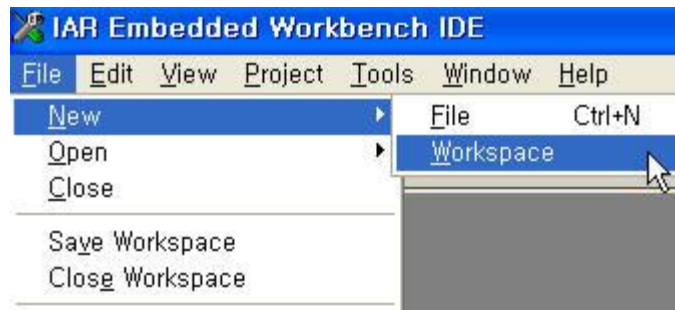


main program

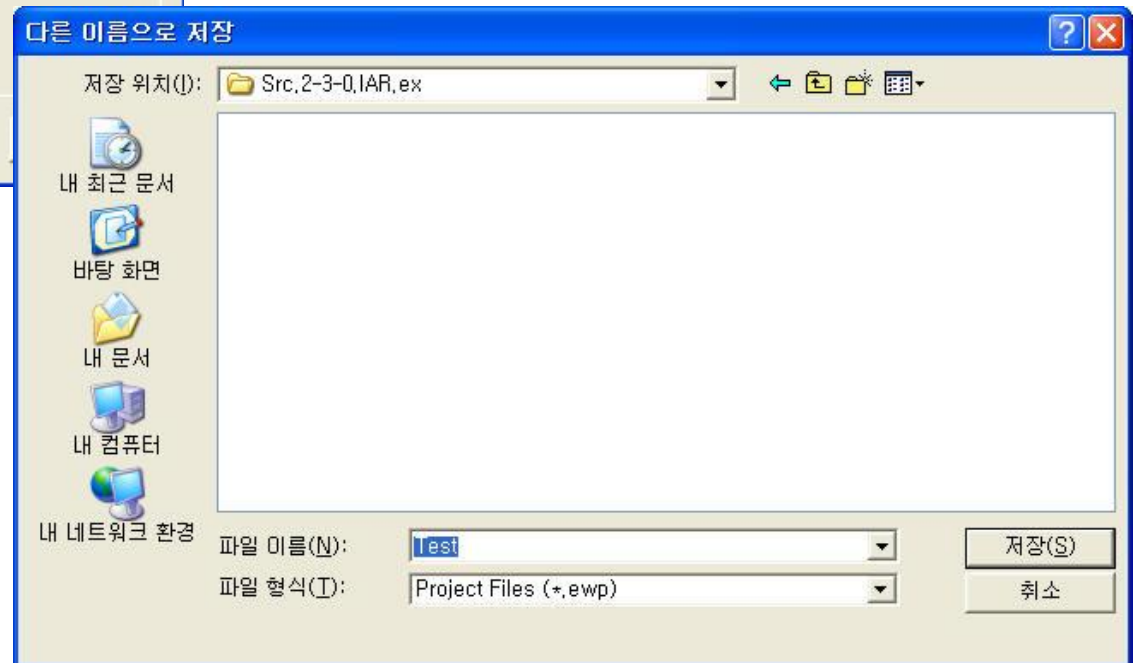
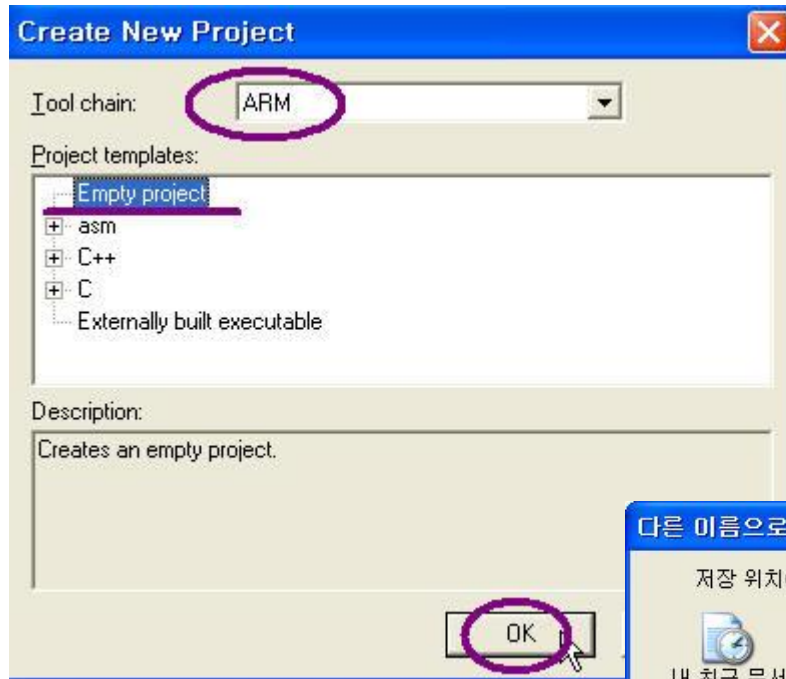
```
int main(void)
{
    (*(volatile unsigned *)0x40021018) |= 0x8;
    (*(volatile unsigned *)0x40010C04) |= 0x10;

    while(1)
    {
        ;
    }
}
```

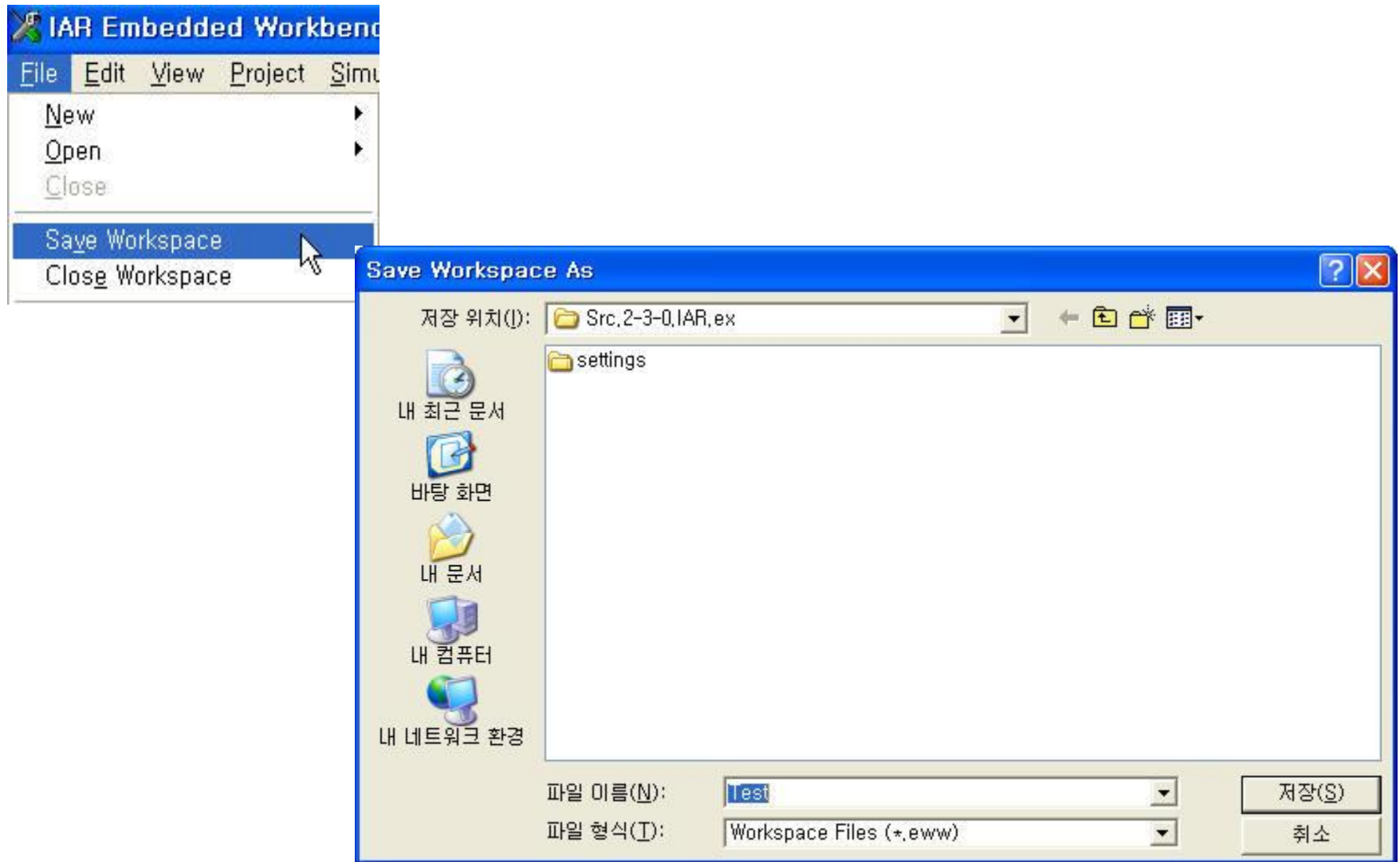
IAR Workbench Project 만들기 (1)



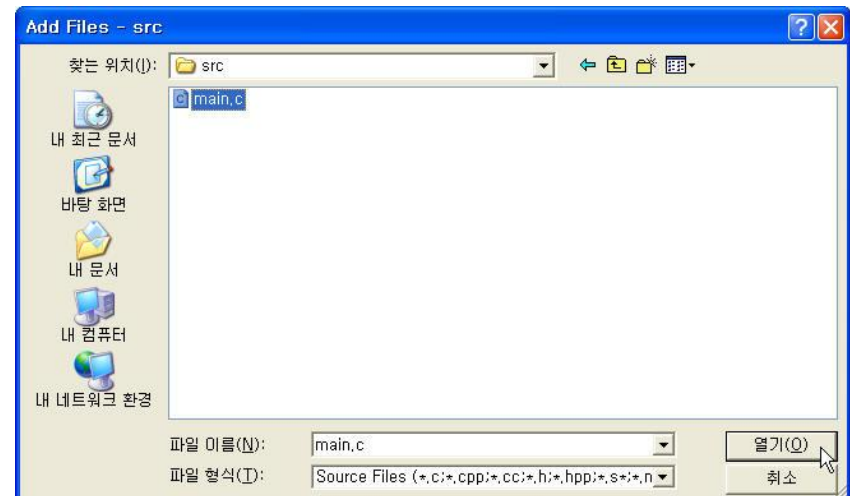
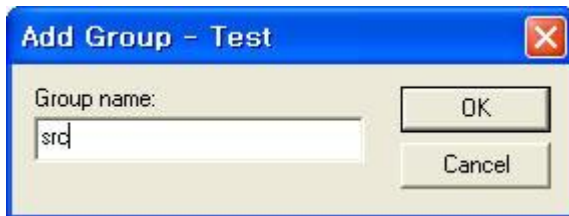
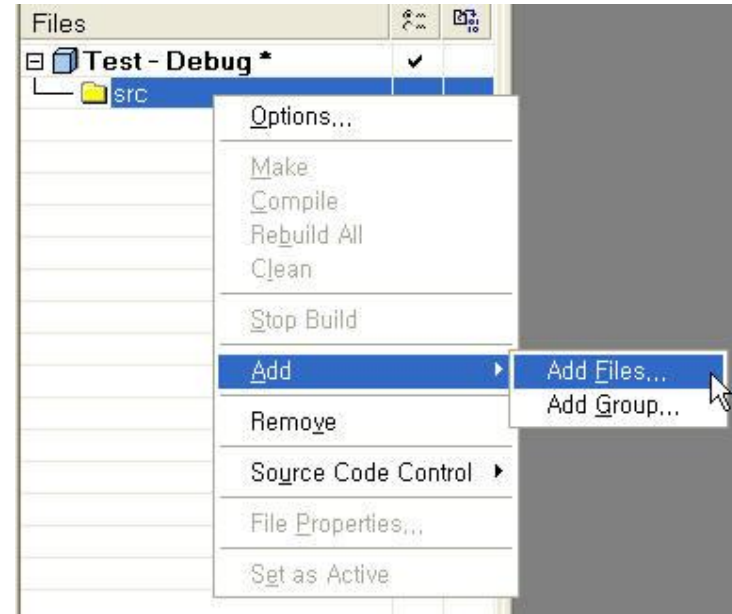
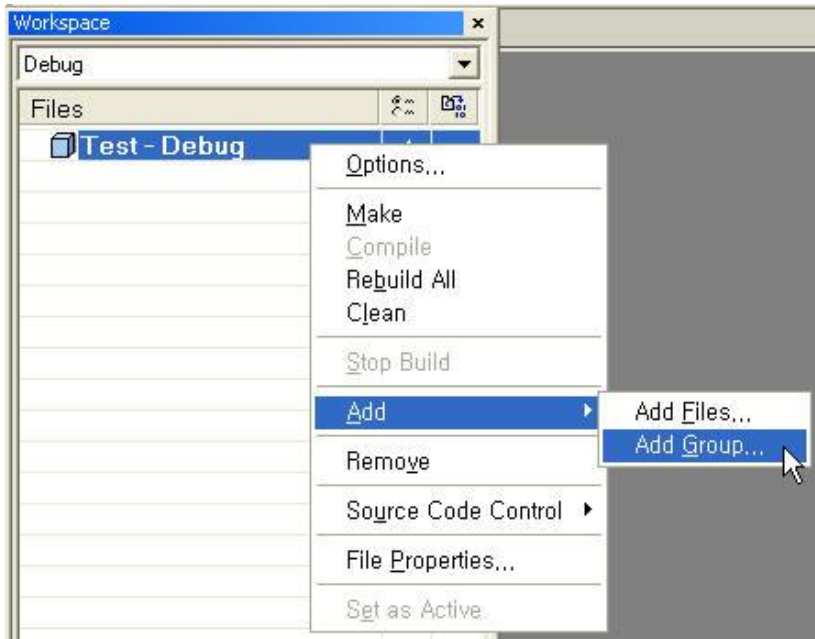
IAR Workbench Project 만들기 (2)



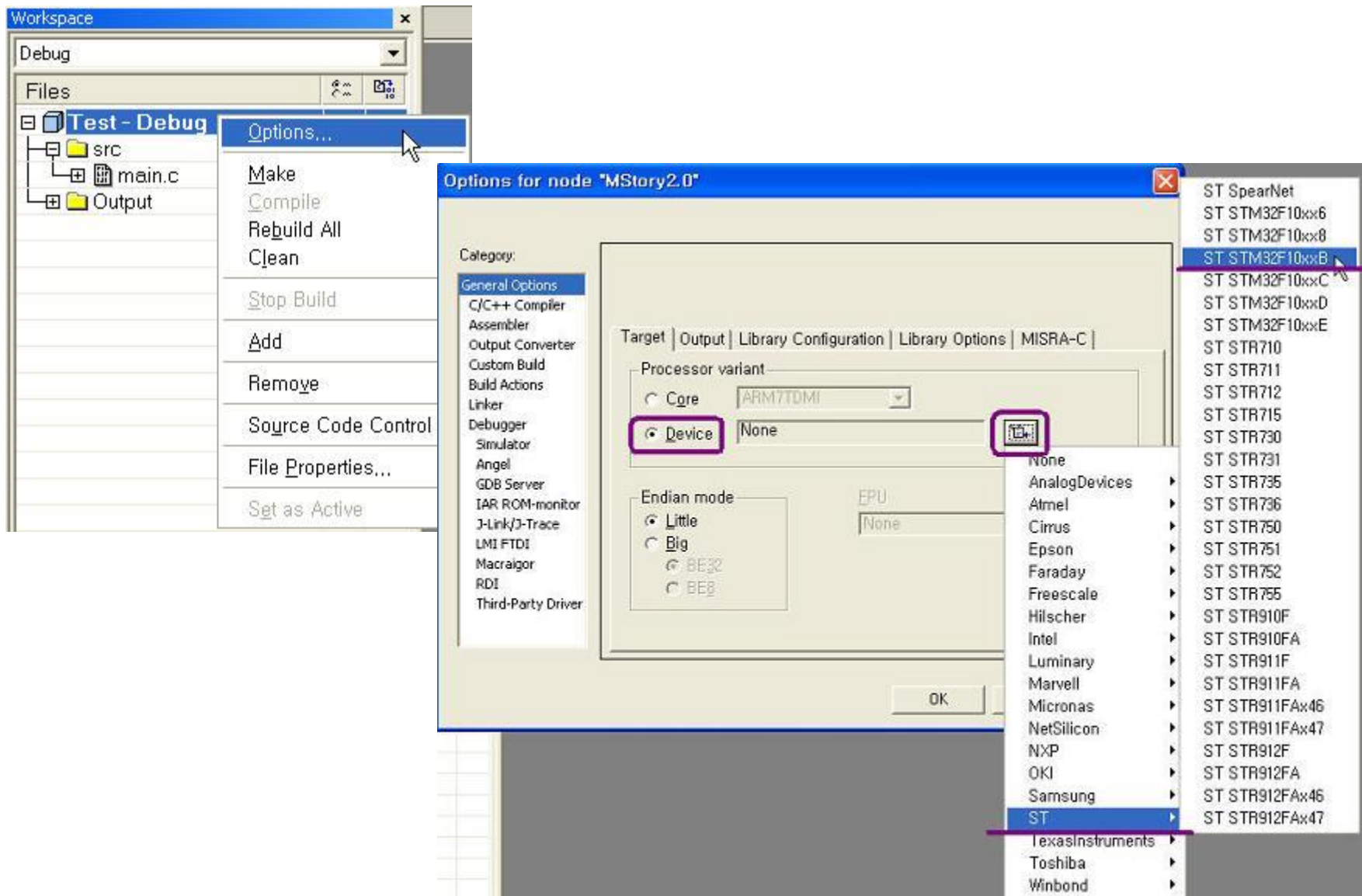
IAR Workbench Project 만들기 (3)



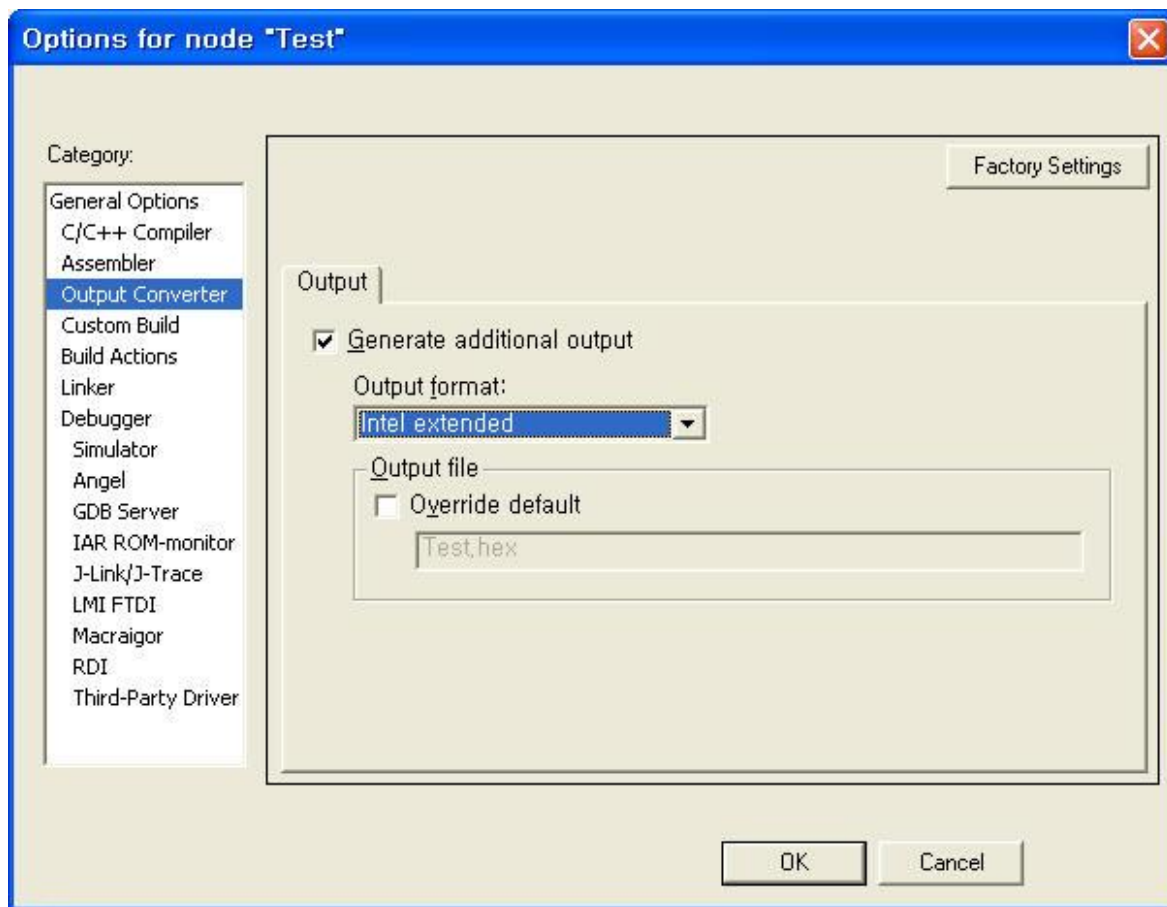
IAR Workbench Project 만들기 (4)



Project Option 설정 (1)

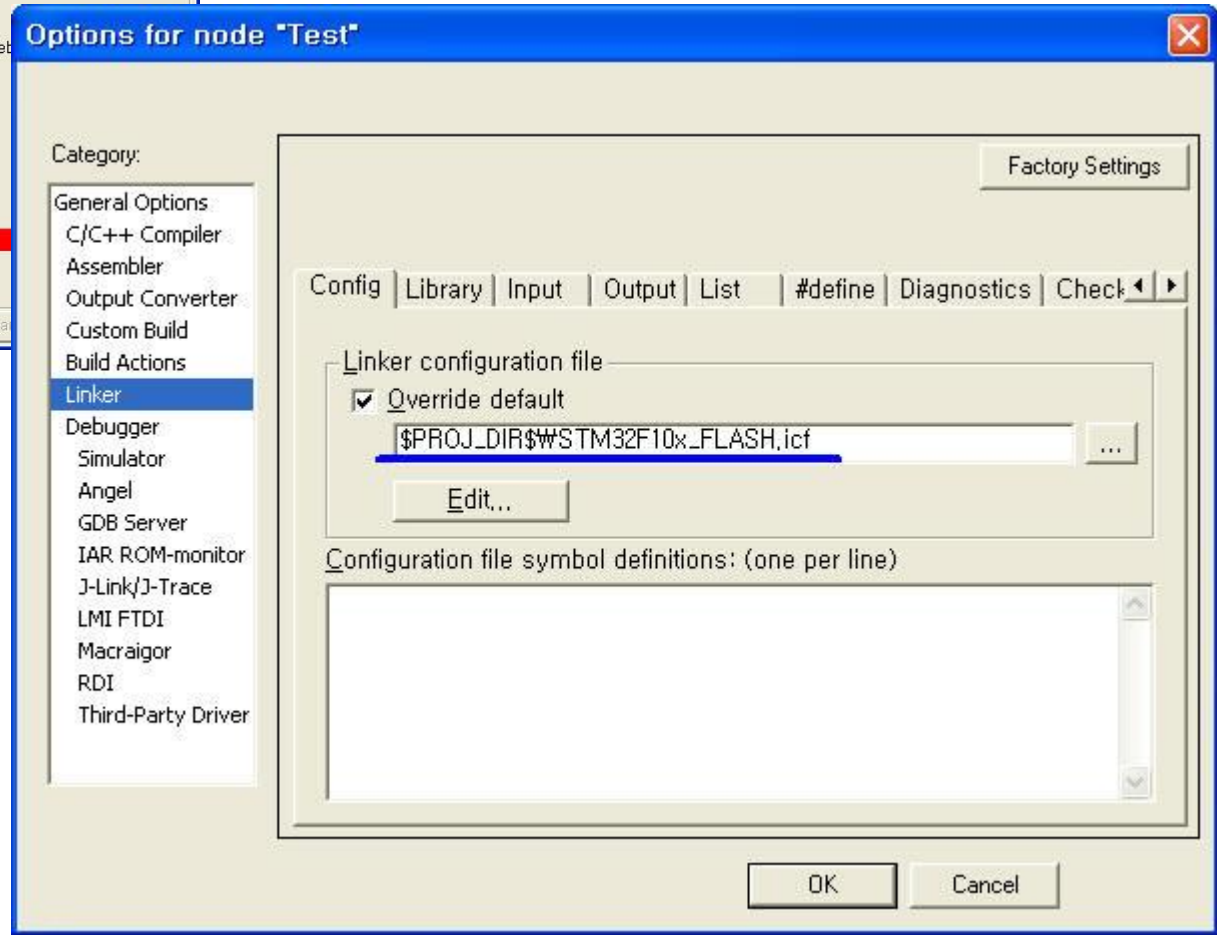
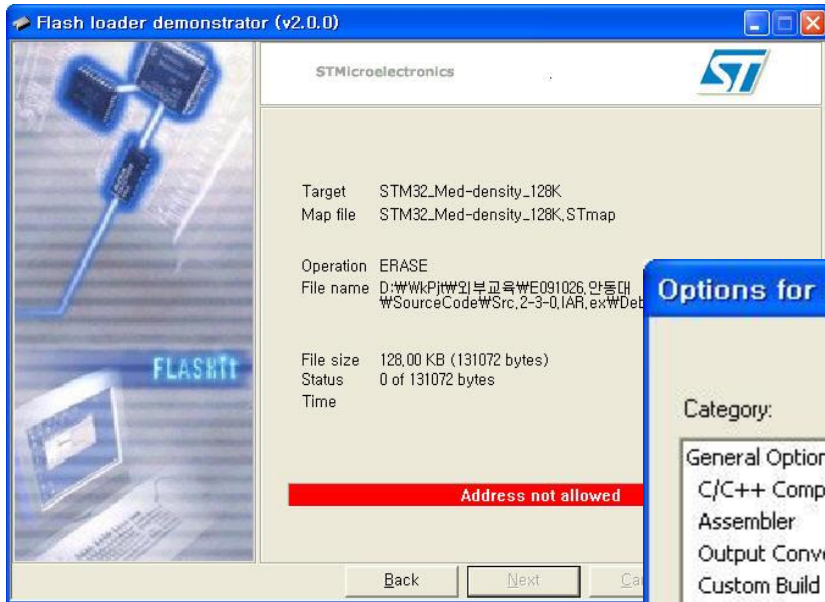


Project Option 설정 (2)



- output 파일의 형태는 binary여도 되고, Intel extended의 hex 형태여도 상관은 없다.
- ST의 Flash loader 프로그램에서 두 가지 형태를 모두 지원한다

Address not allowed Error



stm32f10x_flash.icf

ILINK Configuration File

```
/*-Specials-*/
```

```
define symbol __ICFEDIT_intvec_start__ = 0x08000000;
```

```
/*-Memory Regions-*/
```

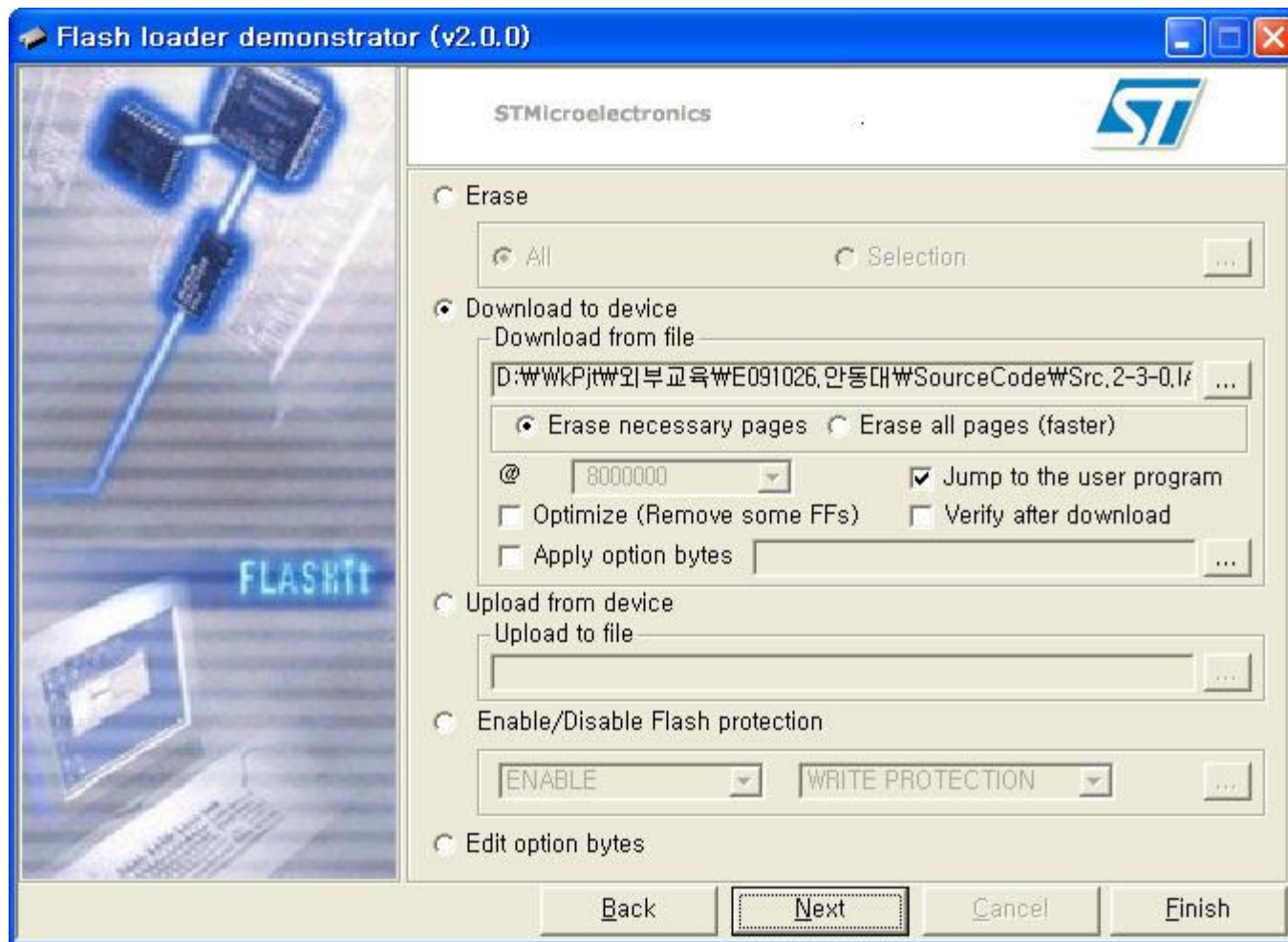
```
define symbol __ICFEDIT_region_ROM_start__ = 0x08000000 ;
```

```
define symbol __ICFEDIT_region_ROM_end__ = 0x0807FFFF;
```

```
define symbol __ICFEDIT_region_RAM_start__ = 0x20000000;
```

```
define symbol __ICFEDIT_region_RAM_end__ = 0x2000FFFF;
```

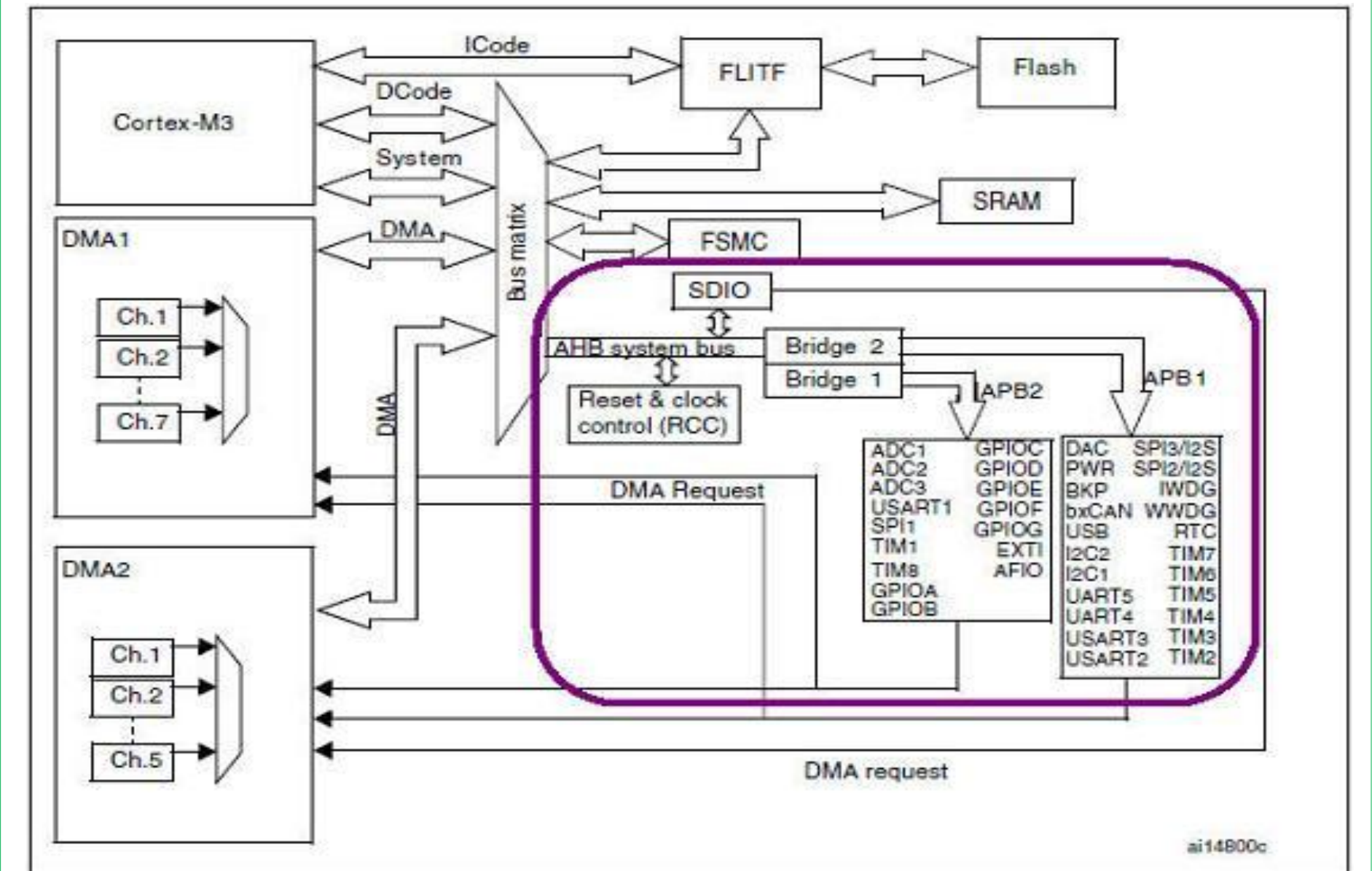
Flash loader Option



Peripheral Memory Map

APB memory space					
0xFFFF FFFF	reserved	0x4002 0000	reserved	0x4001 0000	AFIO
0xE010 0000	reserved	0x4001 3C00	USART1	reserved	reserved
0x6000 0000	reserved	0x4001 3800	reserved	0x4000 7400	PWR
0x4002 3400	CRC	0x4001 3400	SPI1	0x4000 7000	BKP
0x4002 3000	reserved	0x4001 3000	TIM1	0x4000 6C00	reserved
0x4002 2400	Flash Interface	0x4001 2C00	ADC2	0x4000 6800	bxCAN
0x4002 2000	reserved	0x4001 2800	ADC1	0x4000 6400	shared 512 byte USB/CAN SRAM
0x4002 1400	reserved	0x4001 2400	reserved	0x4000 6000	USB Registers
0x4002 1000	RCC	0x4001 1C00	Port E	0x4000 5C00	I2C2
0x4002 0400	DMA	0x4001 1800	Port D	0x4000 5800	I2C1
0x4002 0000	reserved	0x4001 1400	Port C	0x4000 5400	reserved
0x4001 3C00	reserved	0x4001 1000	Port B	0x4000 4C00	USART3
		0x4001 0C00	Port A	0x4000 4800	USART2
		0x4001 0800	EXTI	0x4000 4400	reserved
		0x4001 0400	AFIO	0x4000 3C00	SPI2
		0x4001 0000	reserved	0x4000 3800	reserved
		0x4000 7400	PWR	0x4000 3400	IWDG
				0x4000 3000	WWDG
				0x4000 2C00	RTC
				0x4000 2800	reserved
				0x4000 0C00	TIM4
				0x4000 0800	TIM3
				0x4000 0400	TIM2
				0x4000 0000	

APB2 - GPIOB



Reset and clock control (RCC) 레지스터

- APB2 peripheral clock enable register (RCC_APB2ENR)

➤ Address: 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC3 EN	USAR T1EN	TIM8 EN	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	IOPG EN	IOPF EN	IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	Res.	AFIO EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

- IO Port B를 나타내는 것은 3번 비트
- Bit 3 IOPBEN: I/O port B clock enable
 - 0: I/O port B clock disabled
 - 1: I/O port B clock enabled
- GPIO B를 사용하기 위해서 이것에 Clock을 공급할 수 있도록 enable 시키려고 하는 것이고 이것이 RCC_APB2ENR 레지스터의 3번 비트를 1로 설정하는 것이다. **3번 비트를 1로 만들기 위해서 0x8을 OR 연산** 하는 것이다.

GPIO 초기화

- 0x40010C00부터 GPIO Port B
- 빨간색 LED는 Port B의 9번 pin에 연결

(*(volatile unsigned *)0x40010C04) |= 0x10

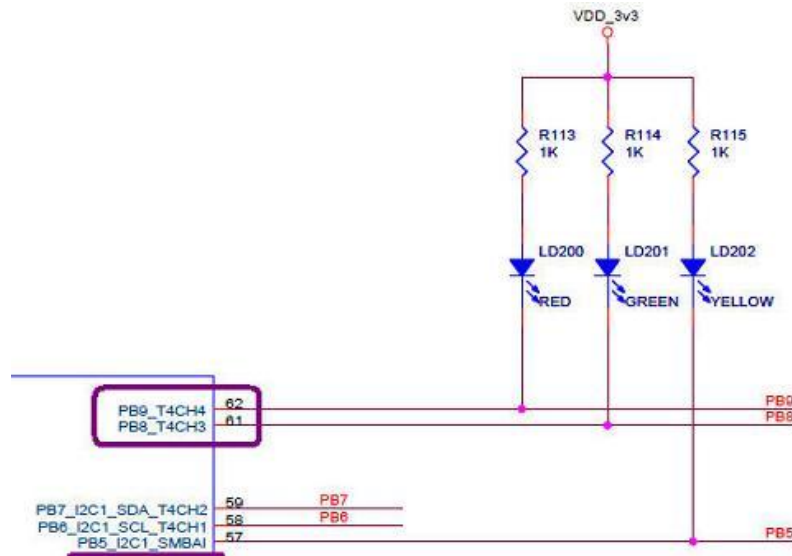
- Port configuration register low (GPIOx_CRL) (x=A..G)
 - Address offset: 0x00, Reset value: 0x4444 4444
- Port configuration register high (GPIOx_CRH) (x=A..G)
 - Address offset: 0x04, Reset value: 0x4444 4444
- low: GPIO Pin 번호가 0부터 7까지 낮은 숫자
- high: GPIO Pin 번호가 8부터 15까지 높은 숫자

GPIOx_CRH 설정

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF15[1:0]		MODE15[1:0]		CNF14[1:0]		MODE14[1:0]		CNF13[1:0]		MODE13[1:0]		CNF12[1:0]		MODE12[1:0]	
rw	rw	rw	rw	rw	rw	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
CNF11[1:0]		MODE11[1:0]		CNF10[1:0]		MODE10[1:0]		CNF9[1:0]		MODE9[1:0]		CNF8[1:0]		MODE8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- MODEy[1:0]: Port x mode bits (y= 8 .. 15)
 - 00: Input mode (reset state)
 - **01: Output mode, max speed 10 MHz.**
 - 10: Output mode, max speed 2 MHz.
 - 11: Output mode, max speed 50 MHz.
- CNFy[1:0]: Port x configuration bits (y= 8 .. 15)
 - In output mode (MODE[1:0] > 00):
 - **00: General purpose output push-pull**
 - 01: General purpose output Open-drain
 - 10: Alternate function output Push-pull
 - 11: Alternate function output Open-drain

LED Off ...



- LED의 한쪽은 VDD 3.3 V에 연결, 다른 한쪽은 STM32 CPU에 연결
- 핀에 0이 인가되어야 VDD 3.3 V에서 시작된 전류가 LED를 타고 흐르게 되고 그것이 LED의 불을 들어오게 만드는 것이다.
- 반대로 전기가 흐르지 않게 만들어야 LED를 끌 수 있는 것이다. 그러려면 1을 인가해야 한다.
- GPIO의 특정 핀에 1을 넣는 것을 Set이라 부르고,
- 0을 넣는 것을 Reset이라고 부른다.

LED Off Test Code

```
static void delay_int_count(volatile unsigned int nTime)
{
    for(; nTime > 0; nTime--);
}
```

```
int main(void)
{
    (*(volatile unsigned *)0x40021018) |= 0x8;
    (*(volatile unsigned *)0x40010C04) |= 0x10;

    delay_int_count(806596); // about 1 second
    (*(volatile unsigned *)0x40010C10) |= 0x200; // LED Off, GPIO Set

    delay_int_count(806596);
    (*(volatile unsigned *)0x40010C14) |= 0x200; // LED On, GPIO Reset

    .....
}
```

Port bit Set Reset register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

- Port bit set/reset register (GPIOx_BSRR) (x=A..G)
 - Address offset: 0x10, Reset value: 0x0000 0000
- 0x40010C00 GPIO Port B base + address offset 0x10
- 왜 Set Reset 레지스터일까? 이 레지스터 하나로 **Set**도 처리하고 **Reset**도 처리할 수 있다.
- $(*(volatile unsigned *)0x40010C10) |= 0x200$; 처리는 결국 9번 비트를 1로 만들어서 Set 작업을 수행하는 것이다.
- 이 작업으로 인해서 LED가 꺼지게 되는 것이다.

Port bit Reset register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

- Port bit reset register (GPIOx_BRR) (x=A..G)
 - Address offset: 0x14, Reset value: 0x0000 0000
- 0x40010C00 GPIO Port B base address + address offset 0x14
- BRR이라는 것이 Bit Reset Register를 의미하는 것이다.
- 하위 16 비트 각각이 하나의 핀을 Reset 시킬 수 있는 것이다.
- 우리는 9번 핀을 작업하고 있기 때문에 9번 비트를 1로 설정하여야 한다. 0x200을 OR 연산하는 것이 바로 이 작업을 수행하는 것이다.