

STM32

[STM32] IAP + USART (AN4657)



이지훈

2017. 4. 7. 13:21

이웃추가

AN4657은 AN2557과 거의 동일하고 다만 SPL이 아닌 STM32CubeMx+HAL 라이브러리로 되어 있다는 차이점과 라이브러리 변경에 따른 APPLICATION_ADDRESS 가 0x0800 3000 에서 0x0800 4000 으로 변경된 점이다.

IAP 동작 Flow는 AN2557과 차이점이 없다.

1. STM32CubeMX 설정

F103RB 보드를 선택한다.

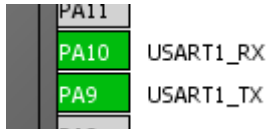
Boards List: 17 Items		
Type	Reference	MCU
Nucleo64	NUCLEO-F030R8	STM32F030R8Tx
Nucleo64	NUCLEO-F070RB	STM32F070RBTx
Nucleo64	NUCLEO-F072RB	STM32F072RBTx
Nucleo64	NUCLEO-F091RC	STM32F091RCTx
Nucleo64	NUCLEO-F103RB	STM32F103RBTx
Nucleo64	NUCLEO-F302R8	STM32F302R8Tx
Nucleo64	NUCLEO-F303RE	STM32F303RETx
Nucleo64	NUCLEO-F334R8	STM32F334R8Tx
Nucleo64	NUCLEO-F401RE	STM32F401RETx
Nucleo64	NUCLEO-F411RE	STM32F411RETx
Nucleo64	NUCLEO-L053R8	STM32L053R8Tx
Nucleo64	NUCLEO-L152RE	STM32L152RETx
Nucleo64	NUCLEO-F446RE	STM32F446RETx
Nucleo64	NUCLEO-L476RG	STM32L476RGTx
Nucleo64	NUCLEO-F410RB	STM32F410RBTx
Nucleo64	NUCLEO-L073RZ	STM32L073RZTx
Nucleo64	NUCLEO-L452RE	STM32L452RETx

RCC를 설정한다.

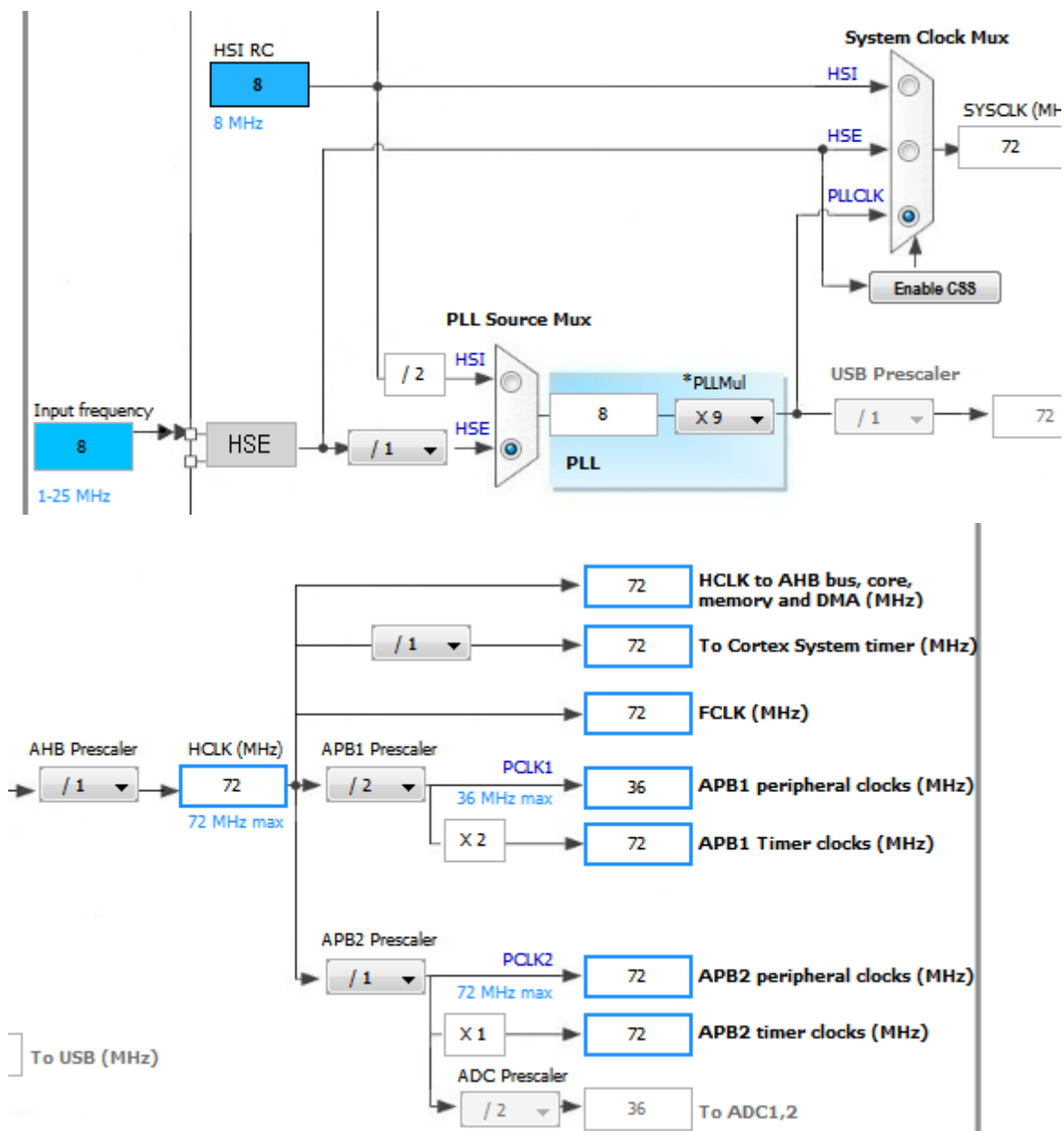
eziya's

☐ Master Clock Output

USART1을 활성화 한다.






시스템 Clock을 설정한다.



USART1 설정은 기본 설정을 그대로 이용한다.

eziya's

Configure the below parameters :

Search :   

<input type="checkbox"/> Basic Parameters	
Baud Rate	115200 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1
<input type="checkbox"/> Advanced Parameters	
Data Direction	Receive and Transmit
Over Sampling	16 Samples

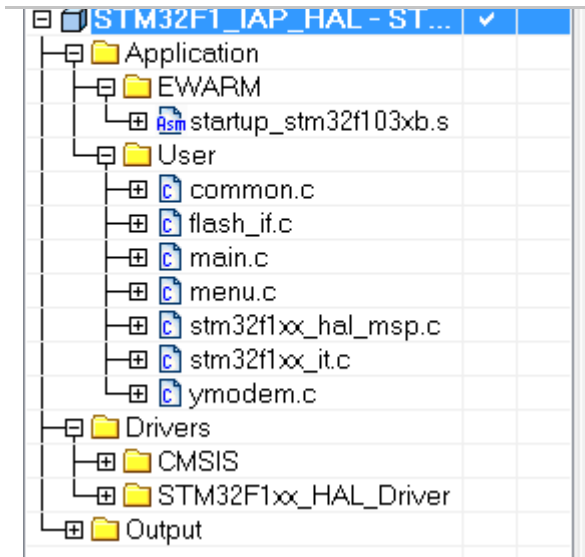
코드를 생성한다.

2. 프로젝트 구성

ST 사이트에서 STM32Cube_FW_F1_V1.4.0 예제를 다운로드 받아서 STM32Cube_FW_F1_V1.4.0 \Projects\STM3210C_EVAL\Applications\IAP\IAP_Main\Src 폴더에서 common.c, flash_if.c menu.c ymodem.c 파일을 복사한다.

Inc 폴더에서 common.h flash_if.h menu.h ymodem.h 파일을 복사한다.

eziya's



우선 컴파일을 해보면 아래와 같이 오류가 발생한다. 대강 확인해 보면 Flash Page 관련 정의와 보드 BSP 관련 오류로 판명된다.

Changed settings forces a full rebuild...

Building configuration: STM32F1_IAP_HAL_USART - STM32F1_IAP_HAL_USART

Updating build tree...

4 file(s) deleted.

Updating build tree...

startup_stm32f103xb.s

flash_if.c

Error[Pe020]: identifier "OB_WRP_PAGES8TO9" is undefined E:\MyWorkSpace\STM32\STM32F1_IAP_HAL_USART\Src\flash_if.c 177

Error[Pe020]: identifier "OB_WRP_PAGES10TO11" is undefined E:\MyWorkSpace\STM32\STM32F1_IAP_HAL_USART\Src\flash_if.c 177

Error[Pe020]: identifier "OB_WRP_PAGES12TO13" is undefined E:\MyWorkSpace\STM32\STM32F1_IAP_HAL_USART\Src\flash_if.c 177

Error[Pe020]: identifier "OB_WRP_PAGES14TO15" is undefined E:\MyWorkSpace\STM32\STM32F1_IAP_HAL_USART\Src\flash_if.c 177

Error[Pe020]: identifier "OB_WRP_PAGES16TO17" is undefined E:\MyWorkSpace\STM32\STM32F1_IAP_HAL_USART\Src\flash_if.c 177

Error[Pe020]: identifier "OB_WRP_PAGES18TO19" is undefined E:\MyWorkSpace\STM32\STM32F1_IAP_HAL_USART\Src\flash_if.c 177

Error[Pe020]: identifier "OB_WRP_PAGES20TO21" is undefined E:\MyWorkSpace\STM32

eziya's

```

Error[Pe020]: identifier "OB_WRP_PAGES24T025" is undefined E:\MyWorkspace\STM32
\STM32F1_IAP_HAL_USART\Src\flash_if.c 177
Error[Pe020]: identifier "OB_WRP_PAGES26T027" is undefined E:\MyWorkspace\STM32
\STM32F1_IAP_HAL_USART\Src\flash_if.c 177
Error[Pe020]: identifier "OB_WRP_PAGES28T029" is undefined E:\MyWorkspace\STM32
\STM32F1_IAP_HAL_USART\Src\flash_if.c 177
Error[Pe020]: identifier "OB_WRP_PAGES30T031" is undefined E:\MyWorkspace\STM32
\STM32F1_IAP_HAL_USART\Src\flash_if.c 177
Error[Pe020]: identifier "OB_WRP_PAGES32T033" is undefined E:\MyWorkspace\STM32
\STM32F1_IAP_HAL_USART\Src\flash_if.c 177
Error[Pe020]: identifier "OB_WRP_PAGES34T035" is undefined E:\MyWorkspace\STM32
\STM32F1_IAP_HAL_USART\Src\flash_if.c 177
Error[Pe020]: identifier "OB_WRP_PAGES36T037" is undefined E:\MyWorkspace\STM32
\STM32F1_IAP_HAL_USART\Src\flash_if.c 177
Error[Pe020]: identifier "OB_WRP_PAGES38T039" is undefined E:\MyWorkspace\STM32
\STM32F1_IAP_HAL_USART\Src\flash_if.c 177
...
common.c
Fatal Error[Pe1696]: cannot open source file "stm3210c_eval.h" E:\MyWorkspace\S
TM32\STM32F1_IAP_HAL_USART\Inc\common.h 44
    searched: "E:\MyWorkspace\STM32\STM32F1_IAP_HAL_USART\Inc\"
    searched: "E:\MyWorkspace\STM32\STM32F1_IAP_HAL_USART\Src\"
    searched: "E:\MyWorkspace\STM32\STM32F1_IAP_HAL_USART\EWARM/./Inc
\"
    searched: "E:\MyWorkspace\STM32\STM32F1_IAP_HAL_USART\EWARM/./Driv
ers/STM32F1xx_HAL_Driver/Inc\"
    searched: "E:\MyWorkspace\STM32\STM32F1_IAP_HAL_USART\EWARM/./Driv
ers/STM32F1xx_HAL_Driver/Inc/Legacy\"
    searched: "E:\MyWorkspace\STM32\STM32F1_IAP_HAL_USART\EWARM/./Driv
ers/CMSIS/Device/ST/STM32F1xx/Include\"
    searched: "E:\MyWorkspace\STM32\STM32F1_IAP_HAL_USART\EWARM/./Driv
ers/CMSIS/Include\"

Error[Pe020]: identifier "UartHandle" is undefined E:\MyWorkspace\STM32\STM32F1
_IAP_HAL_USART\Src\common.c 169
Error[Pe020]: identifier "UartHandle" is undefined E:\MyWorkspace\STM32\STM32F1
_IAP_HAL_USART\Src\common.c 180

```

eziya's

우선 좀더 쉬운 include 관련 오류를 제거하기 위해서 "stm3210c_eval.h" include를 모두 제거한다. 두번 째로는 main.c 에 정의된 UART_HandleTypeDef huart1 의 이름을 UartHandle로 변경하고 common.h 파일에 extern 선언을 추가한다.

```
/* main.c */

/* Private variables -----
*/
UART_HandleTypeDef UartHandle;

/* common.h */
#ifndef __COMMON_H
#define __COMMON_H

/* Includes -----
*/
#include "stm32f1xx.h"

extern UART_HandleTypeDef UartHandle;
```

다시 빌드를 해본다. 이제 Flash 페이지 오류만 남아있고 나머지 오류는 모두 사라졌다. 오류가 나는 부분을 보면 아래 부분에서 FLASH_PAGE_TO_BE_PROTECTED 의 정의에 심볼들이 없다는 오류이다. 이유는 F103RB 보드와 예제 코드를 가져온 STM3210C_EVAL 보드의 Flash 메모리 스펙이 다르기 때문이다.

```
/* flash_if.c */

/* Get pages already write protected *****/
ProtectedPAGE = ~(OptionsBytesStruct.WRPPage) & FLASH_PAGE_TO_BE_PROTECTED;
```

eziya's

```

/* Define bitmap representing user flash area that could be write protected (check restricted to pages 8-39). */
#define FLASH_PAGE_TO_BE_PROTECTED (OB_WRP_PAGES8TO9 | OB_WRP_PAGES10TO11 | OB_WRP_PAGES12TO13 | OB_WRP_PAGES14TO15 | \
                                     OB_WRP_PAGES16TO17 | OB_WRP_PAGES18TO19 | OB_WRP_PAGES20TO21 | OB_WRP_PAGES22TO23 | \
                                     OB_WRP_PAGES24TO25 | OB_WRP_PAGES26TO27 | OB_WRP_PAGES28TO29 | OB_WRP_PAGES30TO31 | \
                                     OB_WRP_PAGES32TO33 | OB_WRP_PAGES34TO35 | OB_WRP_PAGES36TO37 | OB_WRP_PAGES38TO39 )

```

결론적으로 말해서, flash_if.h 파일에서 정의된 내용들을 STM32F103RB 에 맞게 수정해야 한다.

- STM32F103의 Flash page size는 **1KB**이다.
- User application 은 **16KB** 만큼 이동한 **0x08004000** 에 위치한다.
- STM32F103의 Flash 사이즈는 **128Kbytes** 이다. 따라서 USER_FLASH_END_ADDRESS 는 **0x08020000** 이 된다.
- USER_FLASH_SIZE는 **12KB**로 설정되어 있는데 이부분은 자신의 업데이트 할 프로그램의 크기에 맞게 수정하면 된다.
- FLASH_PAGE_TO_BE_PROTECTED 는 User application 이 올라가는 페이지 범위가 되는데 앞서 보았듯이 16KB offset에 프로그램의 크기가 12KB 이기 때문에 보호되는 페이지 범위는 16~27 이 된다.

```

/* Define to prevent recursive inclusion -----
*/
#ifndef __FLASH_IF_H
#define __FLASH_IF_H

/* Includes -----
*/
#include "stm32f1xx_hal.h"

/* Error code */
enum

```

eziya's

```

FLASHIF_WRITING_ERROR,
FLASHIF_PROTECTION_ERROR
};

/* protection type */
enum{
    FLASHIF_PROTECTION_NONE          = 0,
    FLASHIF_PROTECTION_PCROPENABLED  = 0x1,
    FLASHIF_PROTECTION_WRPENABLED    = 0x2,
    FLASHIF_PROTECTION_RDPENABLED    = 0x4,
};

/* protection update */
enum {
    FLASHIF_WRP_ENABLE,
    FLASHIF_WRP_DISABLE
};

/* Define the address from where user application will be loaded.
   Note: this area is reserved for the IAP code */
#define FLASH_PAGE_STEP          FLASH_PAGE_SIZE          /* Size of page : 1 K
bytes */
#define APPLICATION_ADDRESS      (uint32_t)0x08004000      /* Start user code ad
dress: ADDR_FLASH_PAGE_16, Offset 16Kbytes */

/* Notable Flash addresses */
#define USER_FLASH_END_ADDRESS   0x08020000              /* Size of flash : 128
Kbytes */

/* Define the user application size */
#define USER_FLASH_SIZE          ((uint32_t)0x00003000) /* 12Kbytes for us
er application */

/* Define bitmap representing user flash area that could be write protected (ch
eck restricted to pages 16-27). */
#define FLASH_PAGE_TO_BE_PROTECTED ( OB_WRP_PAGES16TO19 | OB_WRP_PAGES20TO23 |
OB_WRP_PAGES24TO27 )

```

eziya's

```

*/
/* ABSoulute value */
#define ABS_RETURN(x,y)                (((x) < (y)) ? (y) : (x))

/* Get the number of sectors from where the user program will be loaded */
#define FLASH_SECTOR_NUMBER            ((uint32_t)(ABS_RETURN(APPLICATION_ADDRES
S,FLASH_START_BANK1))>>12)

/* Compute the mask to test if the Flash memory, where the user program will be
loaded, is write protected */
#define FLASH_PROTECTED_SECTORS        (~(uint32_t)((1 << FLASH_SECTOR_NUMBER) -
1))

/* Exported functions -----
*/
void FLASH>If_Init(void);
uint32_t FLASH>If_Erase(uint32_t StartSector);
uint32_t FLASH>If_GetWriteProtectionStatus(void);
uint32_t FLASH>If_Write(uint32_t destination, uint32_t *p_source, uint32_t leng
th);
uint32_t FLASH>If_WriteProtectionConfig(uint32_t protectionstate);

#endif /* __FLASH_IF_H */

/***** (C) COPYRIGHT STMicroelectronics *****/END OF FILE***
*/

```

코드를 수정하고 다시 컴파일해 본다. 이제 문제없이 컴파일이 완료되는 것을 확인할 수 있다.

```

Building configuration: STM32F1_IAP_HAL_USART - STM32F1_IAP_HAL_USART
Updating build tree...
menu.c
flash_if.c
ymodem.c
Linking

```

eziya's

이제 main.c 파일을 수정해야 한다. 수정 내용은 간단하다 앞서 샘플 코드에서 시작 부분을 복사해서 붙여 놓고 KEY가 눌리는 부분만 NUCLEO-F103RB 보드의 PC13번 핀에 맞게 수정해 주면 된다.

```

/* Includes -----
*/
#include "main.h"
#include "stm32f1xx_hal.h"

/* USER CODE BEGIN Includes */
#include "menu.h"
/* USER CODE END Includes */

/* Private variables -----
*/
UART_HandleTypeDef UartHandle;

/* USER CODE BEGIN PV */
/* Private variables -----
*/
extern pFunction JumpToApplication;
extern uint32_t JumpAddress;

/* USER CODE END PV */

/* Private function prototypes -----
*/
void SystemClock_Config(void);
void Error_Handler(void);
static void MX_GPIO_Init(void);
static void MX_USART1_UART_Init(void);

/* USER CODE BEGIN PFP */
/* Private function prototypes -----

```

eziya's

```
/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

int main(void)
{

    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----
    */

    /* Reset of all peripherals, Initializes the Flash interface and the Systick.
    */
    HAL_Init();

    /* Configure the system clock */
    SystemClock_Config();

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_USART1_UART_Init();

    /* USER CODE BEGIN 2 */
    if(HAL_GPIO_ReadPin(B1_GPIO_Port, B1_Pin) == GPIO_PIN_RESET)
    {
        FLASH_If_Init();
        Main_Menu();
    }
    else
    {
        if (((*(__IO uint32_t*)APPLICATION_ADDRESS) & 0x2FFE0000 ) == 0x20000000)
        {
            /* Jump to user application */

```

eziya's

```
    __set_MSP((__IO uint32_t*) APPLICATION_ADDRESS);  
    JumpToApplication();  
}  
}  
  
/* USER CODE END 2 */  
  
/* Infinite loop */  
/* USER CODE BEGIN WHILE */  
while (1)  
{  
    /* USER CODE END WHILE */  
  
    /* USER CODE BEGIN 3 */  
  
}  
/* USER CODE END 3 */  
  
}
```

이제 IAP 프로그램을 ST-LINK를 이용해서 보드에 올리고 터미널 프로그램을 실행해서 User Button 을 누르고 부팅하는 경우에 정상적으로 메시지가 올라오는지 확인한다.

eziya's

```

=====
=                (C) COPYRIGHT 2016 STMicroelectronics                =
=                                                                =
=  STM32F1xx In-Application Programming Application  (Version 1.0.0)  =
=                                                                =
=                        By MCD Application Team                        =
=====

===== Main Menu =====

Download image to the internal Flash ----- 1
Upload image from the internal Flash ----- 2
Execute the loaded application ----- 3
Enable the write protection ----- 4
=====

```

3. 테스트

이제 테스트를 진행해 보자. 여기서 확인해야 할 것은 앞서 **User Application의 위치를 0x08004000**로 잡아놓았다는 점이다.

따라서 테스트로 업로드할 User Application 도 이에 맞춰서 시작 위치를 0x08004000으로 설정해야 한다. 우선 테스트 Application은 최대한 간단한게 Loop 안에서 LED를 Toggle하면서 USART1으로 Hello World 메시지를 전송하도록 코딩한다.

```

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
    HAL_Delay(500);
    HAL_UART_Transmit(&huart1, "Hello World!\r\n", 14, 100);
}
/* USER CODE END 3 */

```

eziya's

ET 값을 수정한다.

```
/* #define VECT_TAB_SRAM */
#define VECT_TAB_OFFSET 0x4000 /*!< Vector Table base offset field.
This value must be a multiple of 0x200. */
```

둘째로, icf 파일을 수정해서 링커가 메모리 Offset을 적용하도록 합니다.

```
/*###ICF### Section handled by ICF editor, don't touch! ****/
/*-Editor annotation file-*/
/* IcfEditorFile="$T00LKIT_DIR$\config\ide\IcfEditor\cortex_v1_0.xml" */
/*-Specials-*/
define symbol __ICFEDIT_intvec_start__ = 0x08004000;
/*-Memory Regions-*/
define symbol __ICFEDIT_region_ROM_start__ = 0x08004000 ;
define symbol __ICFEDIT_region_ROM_end__ = 0x0801FFFF;
define symbol __ICFEDIT_region_RAM_start__ = 0x20000000;
define symbol __ICFEDIT_region_RAM_end__ = 0x20004FFF;
/*-Sizes-*/
define symbol __ICFEDIT_size_cstack__ = 0x500;
define symbol __ICFEDIT_size_heap__ = 0x200;
/**** End of ICF editor section. ###ICF###*/

define memory mem with size = 4G;
define region ROM_region = mem:[from __ICFEDIT_region_ROM_start__ to __ICFEDIT_region_ROM_end__];
define region RAM_region = mem:[from __ICFEDIT_region_RAM_start__ to __ICFEDIT_region_RAM_end__];

define block CSTACK with alignment = 8, size = __ICFEDIT_size_cstack__ { }
;
define block HEAP with alignment = 8, size = __ICFEDIT_size_heap__ { }
;
```

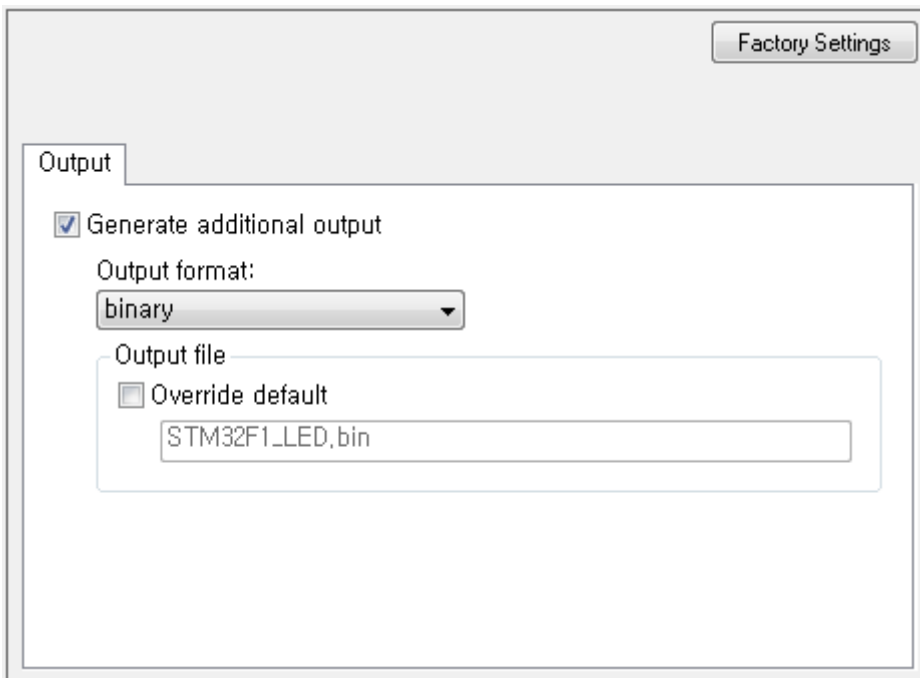
eziya's

```
place at address mem:__ICFEDIT_intvec_start__ { readonly section .intvec };
```

```
place in ROM_region { readonly };
```

```
place in RAM_region { readwrite,  
                      block CSTACK, block HEAP };
```

마지막으로 테스트 프로그램이 YMODEM을 통해서 전송될 수 있도록 Binary 형태로 output이 나올 수 있도록 Project Option에서 Output Converter 를 **binary**로 선택합니다.



빌드를 하면 결과물로 .bin 파일이 생성되고 이 파일을 업로드하면 됩니다.

이제 보드에 IAP를 올려놓은 상태에서 User Button을 누르고 부팅을 해서 메뉴가 나오도록 합니다. 메뉴에서 1번을 선택하면 IAP는 수신상태로 들어갑니다.

eziya's

```

=====
=                (C) COPYRIGHT 2016 STMicroelectronics                =
=                                                                 =
=  STM32F1xx In-Application Programming Application  (Version 1.0.0)  =
=                                                                 =
=                                By MCD Application Team                                =
=====

===== Main Menu =====

Download image to the internal Flash ----- 1
Upload image from the internal Flash ----- 2
Execute the loaded application ----- 3
Enable the write protection ----- 4

=====

Waiting for the file to be sent ... (press 'a' to abort)
CCCCCCCCCCCCCCCC

```

파일 전송 메뉴에서 YMODEM을 선택하고 파일을 전송완료 하면 아래와 같이 완료 메시지가 표시 됩니다.

```

Waiting for the file to be sent ... (press 'a' to abort)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

Programming Completed Successfully!
-----
Name: STM32F1_LED.bin
Size: 4236 Bytes
-----

```

메뉴에서 3번을 눌러서 User Application 주소로 PC 값을 변경하면 User Application이 실행되면서 터미널에 Hello World!가 출력됩니다.

eziya's

```
Hello World!  
Hello World!  
Hello World!  
Hello World!  
Hello World!  
Hello World!
```

#stm32 #nucleo #iap #hal #an4657

1

15



이지훈

달릴 준비만 하는거 아냐...달려야 하는데...^^; <https://github.com/eziya>

이웃추가

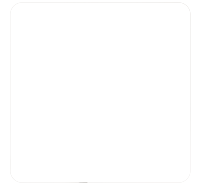
1

15

eziya's

HAL YOOX

HAL 폭 넓은 셀렉션, 신상입고, 최대 50%할인, \$190이상 무료배송!



마르지엘라 | 토즈 | 질샌더 | 프라다

IAP 광수분이온화장치

031)945-4286 공조기 탈취 및 살균 IAP 광수분이온화장치 전문기업.

(주)한국와이파이 특허,벤처

와이파이 설계구축 전문, 공공와이파이, 행사, 이벤트, IOT, 건축시설

회사소개 | 래퍼런스 | 견적문의 | 오시는길

이 블로그 STM32 카테고리 글



이 블로그의 #stm32 다른 글

[STM32 HAL] RTOS + LwIP + MBEDTLS#2

2020. 5. 14.

0 0

[STM32 HAL] RTOS + LwIP + MBEDTLS#1

2020. 5. 12.

2 4

[STM32 HAL] RTOS + LwIP TCP Client

2020. 3. 24.

3 13

[STM32 HAL] RTOS + LwIP TCP Echo Server

1

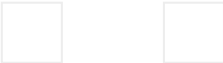
15

eziya's

[STM32 HAL] LwIP TCP Client

2020. 3. 19.

3 0



맨 위로

모먼트로 온라인집들이 하면
네이버페이 최대 1백만원

온라인집들이
가득!

아이와 함께
집도살아요

PC버전으로 보기