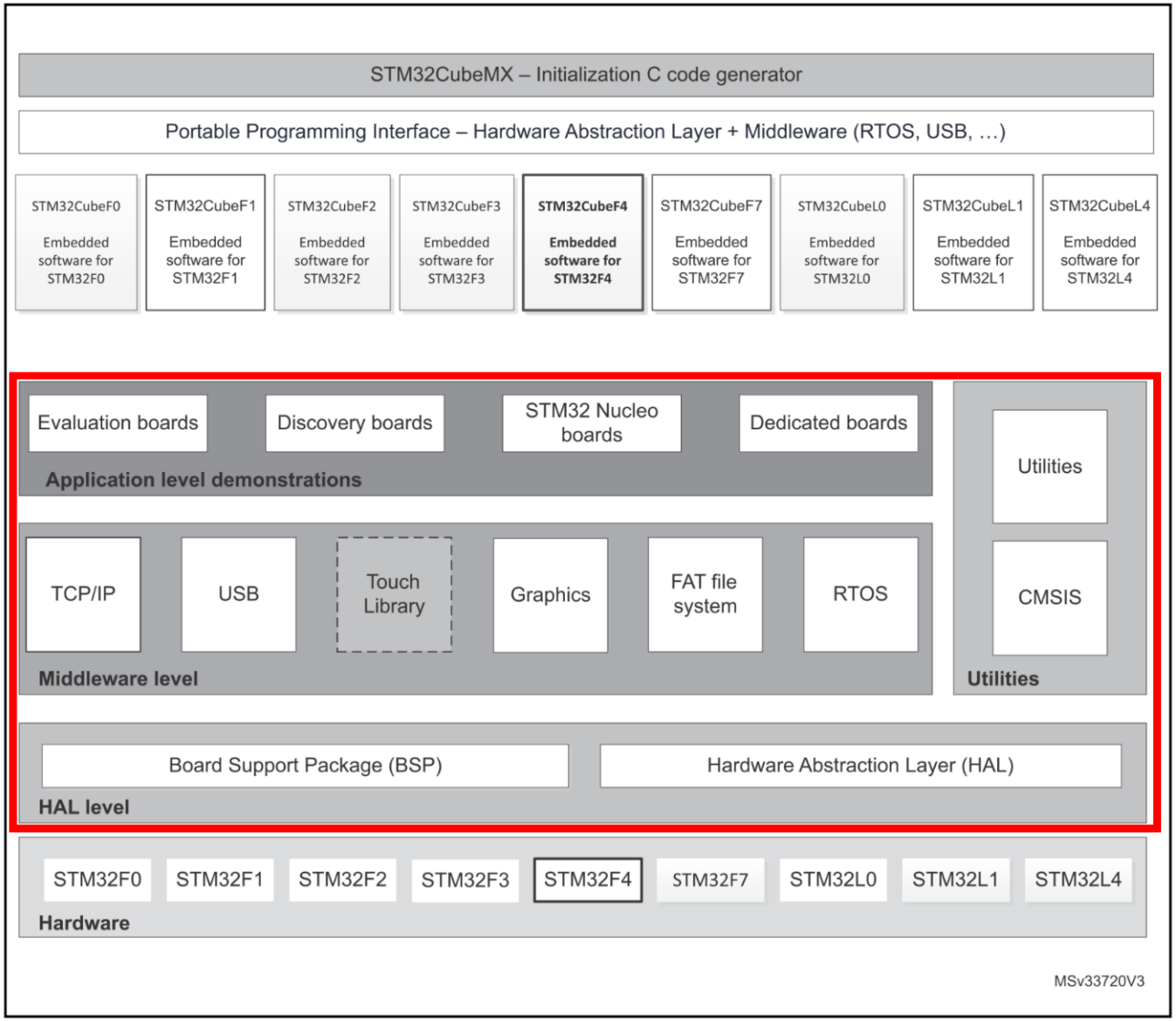
Lesson 5
Standard Peripheral Library

Lecturer: Harvard Tseng

TIM2->CR1 |= 0x1<<4;
What does this mean?

- R/W register coding style is hard to read for human.
- We prefer using functions rather than R/W register.

Separate hardware & software.



What does StdPeriph Library contain?

- Hundreds of examples
- Middleware level driver
- Peripheral driver
- CMSIS

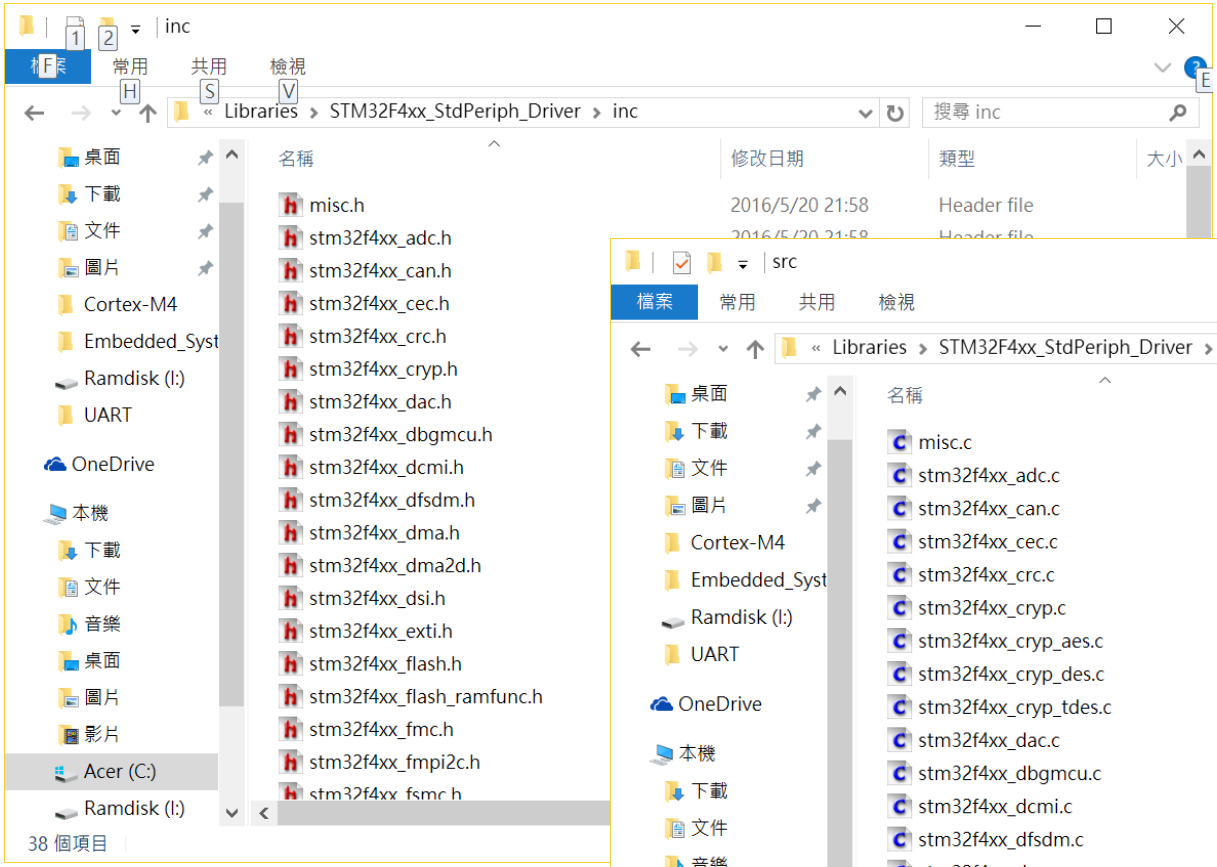
Middleware level

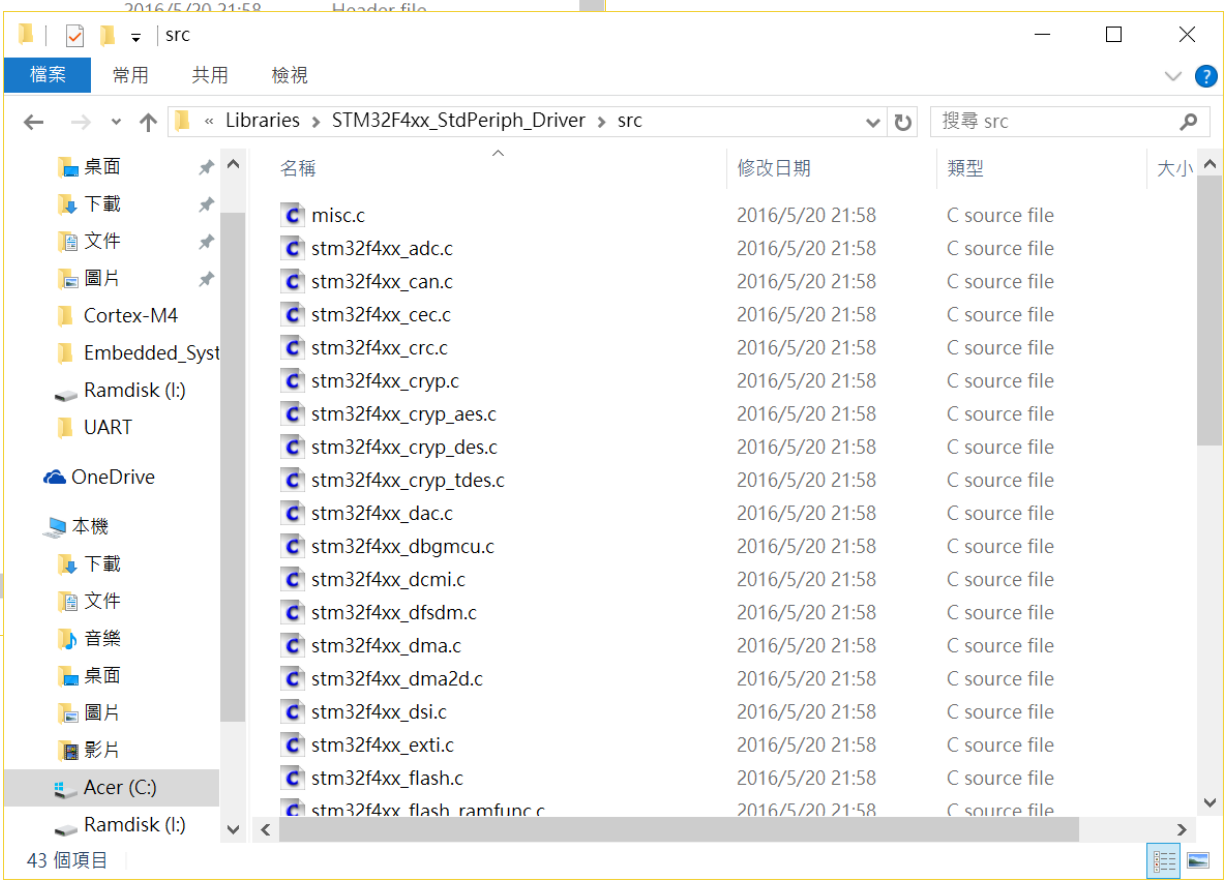
- TCP/IP
- USB
- Graphics
- …

HAL level

- HAL = Hardware Abstraction Layer
- Contains lots of peripheral driver.

Files


inc
檔案
常用
共用
檢視
Libraries › STM32F4xx_StdPeriph_Driver › inc
搜尋 inc
名稱
修改日期
類型
大小
misc.h 2016/5/20 21:58 Header file
stm32f4xx_adc.h
stm32f4xx_can.h
stm32f4xx_cec.h
stm32f4xx_crc.h
stm32f4xx_cryp.h
stm32f4xx_dac.h
stm32f4xx_dbgmcu.h
stm32f4xx_dcmi.h
stm32f4xx_dfsdm.h
stm32f4xx_dma.h
stm32f4xx_dma2d.h
stm32f4xx_dsi.h
stm32f4xx_exti.h
stm32f4xx_flash.h
stm32f4xx_flash_ramfunc.h
stm32f4xx_fmc.h
stm32f4xx_fmpi2c.h
38 個項目


src
檔案
常用
共用
檢視
Libraries › STM32F4xx_StdPeriph_Driver › src
搜尋 src
名稱
修改日期
類型
大小
misc.c 2016/5/20 21:58 C source file
stm32f4xx_adc.c
stm32f4xx_can.c
stm32f4xx_cec.c
stm32f4xx_crc.c
stm32f4xx_cryp.c
stm32f4xx_cryp_aes.c
stm32f4xx_cryp_des.c
stm32f4xx_cryp_tdes.c
stm32f4xx_dac.c
stm32f4xx_dbgmcu.c
stm32f4xx_dcmi.c
stm32f4xx_dfsdm.c
stm32f4xx_dma.c
stm32f4xx_dma2d.c
stm32f4xx_dsi.c
stm32f4xx_exti.c
stm32f4xx_flash.c
43 個項目

StdPeriph Library Setup

Step1. Download StdPeriph Library

- Download from ST.

STM32 Standard Peripheral Libraries

Last Viewed | Share | Bookmark | Download | Full screen

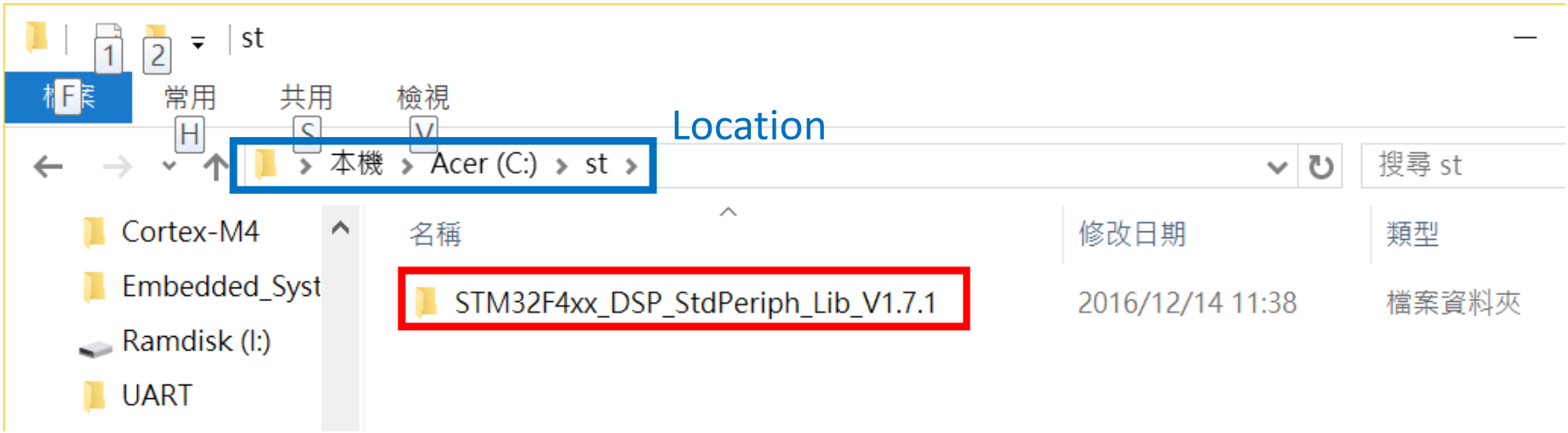
STM32 Standard Peripheral Libraries

Total Parts: (9) for STM32 Standard Peripheral Libraries | Matching Parts : (9) | Customize Table | Reset All

Part Number	Marketing Status	Supplier	Supported Devices	Software Type	Software Version
STSW-STM32054 STM32F10x standard peripheral library	Active	ST	STM32, STM32F1	Firmware	3.5.0
STSW-STM32062 STM32F2xx standard peripherals library (UM1061)	Active	ST	STM32, STM32F2	Firmware	1.1.0
STSW-STM32065 STM32F4 DSP and standard peripherals library	Active	ST	STM32, STM32F4	Firmware	1.8.0
STSW-STM32115 STM32F37x/38x DSP and standard peripherals library, including 73 ex...	Active	ST	STM32F, STM32F3	Firmware	1.0.0

Step2. Extract

- Extract to folder C:\st\



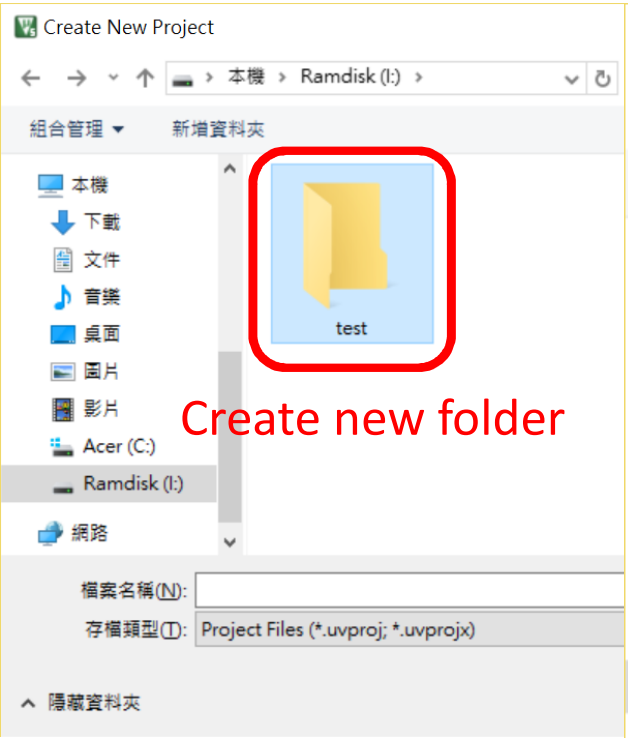
File tree

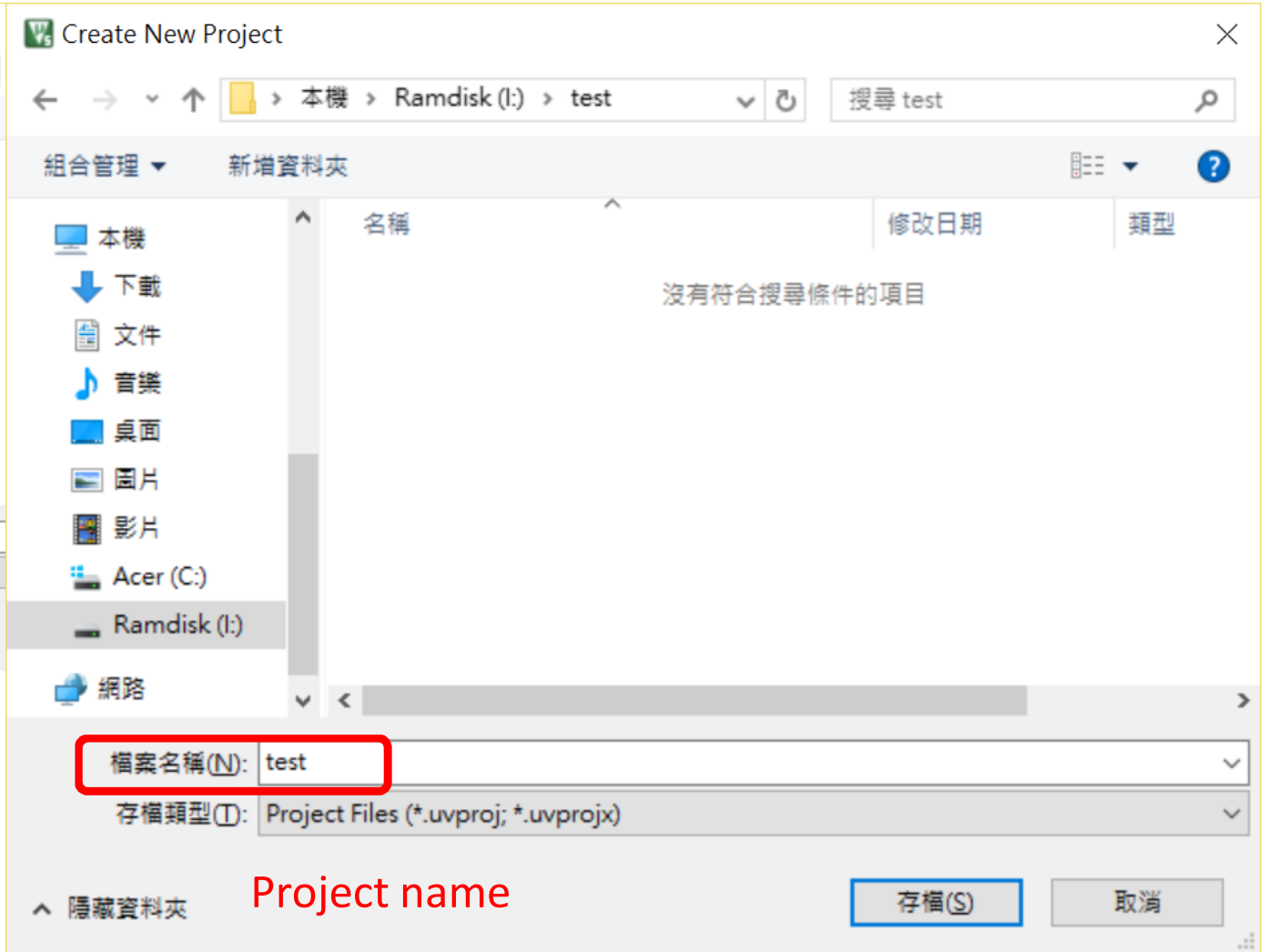
- st
  - STM32F4xx_DSP_StdPeriph_Lib_V1.7.1
    - _htmresc
    - Libraries
      - CMSIS
      - STM32F4xx_StdPeriph_Driver
        - inc
        - src
    - Project ← Examples in here
    - Utilities

StdPeriph_Driver

Step3. Create new project

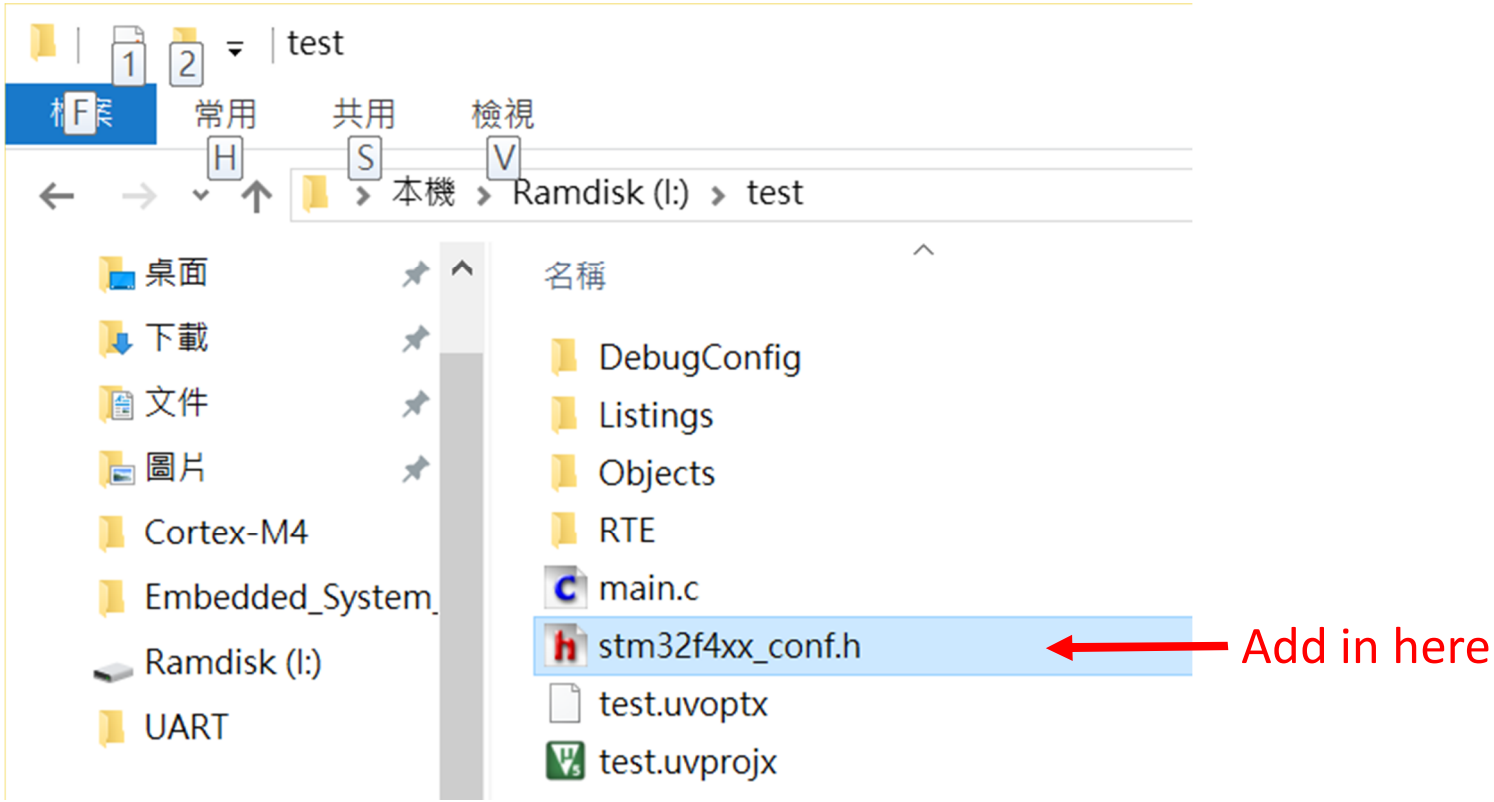
- Setup as usual.



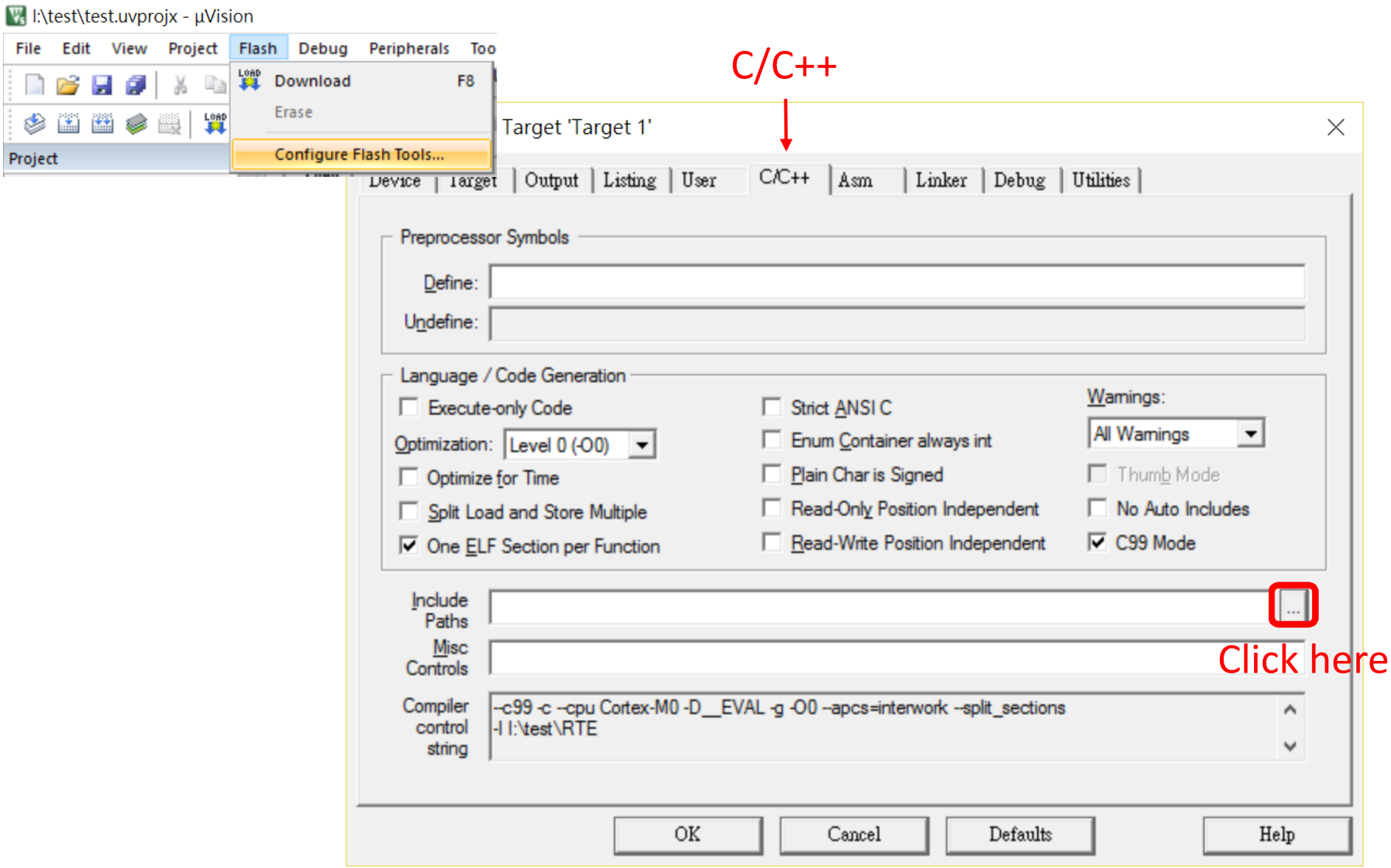


Step4. Add stm32f4xx_conf.h

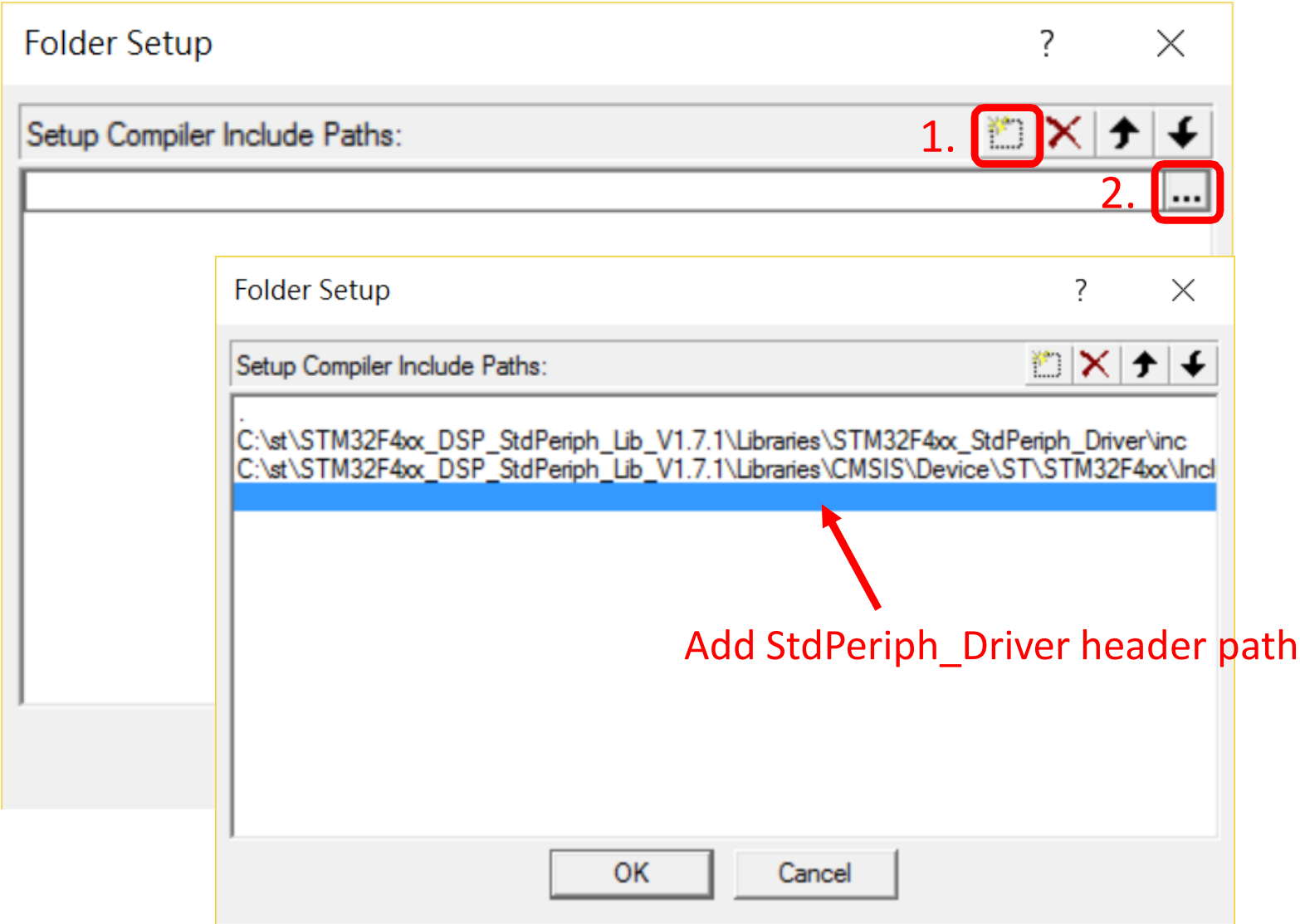
- This header can be found in any example.



Step5. Add include path

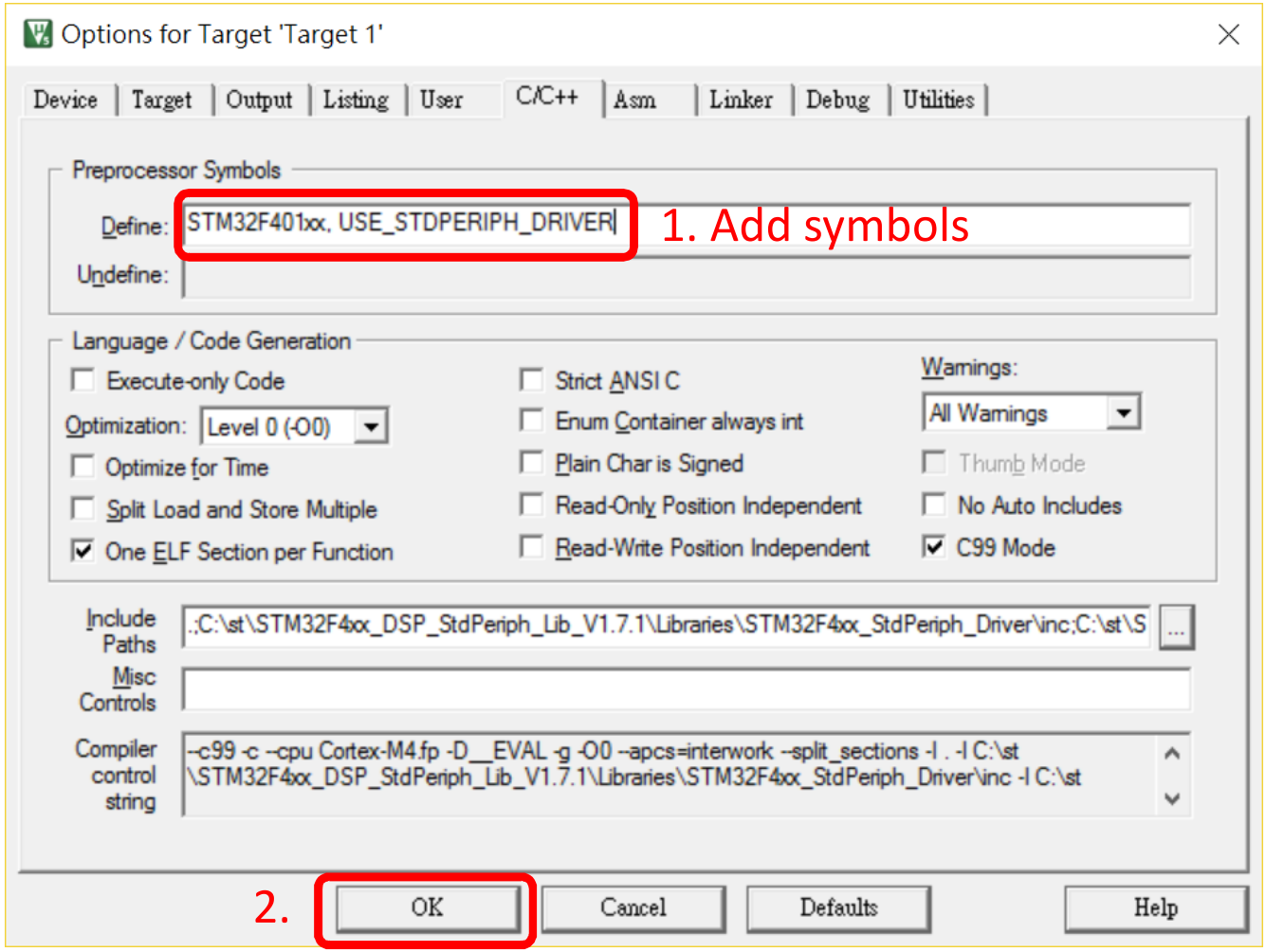


Step5. Add include path



Step6. Add Preprocessor Symbol

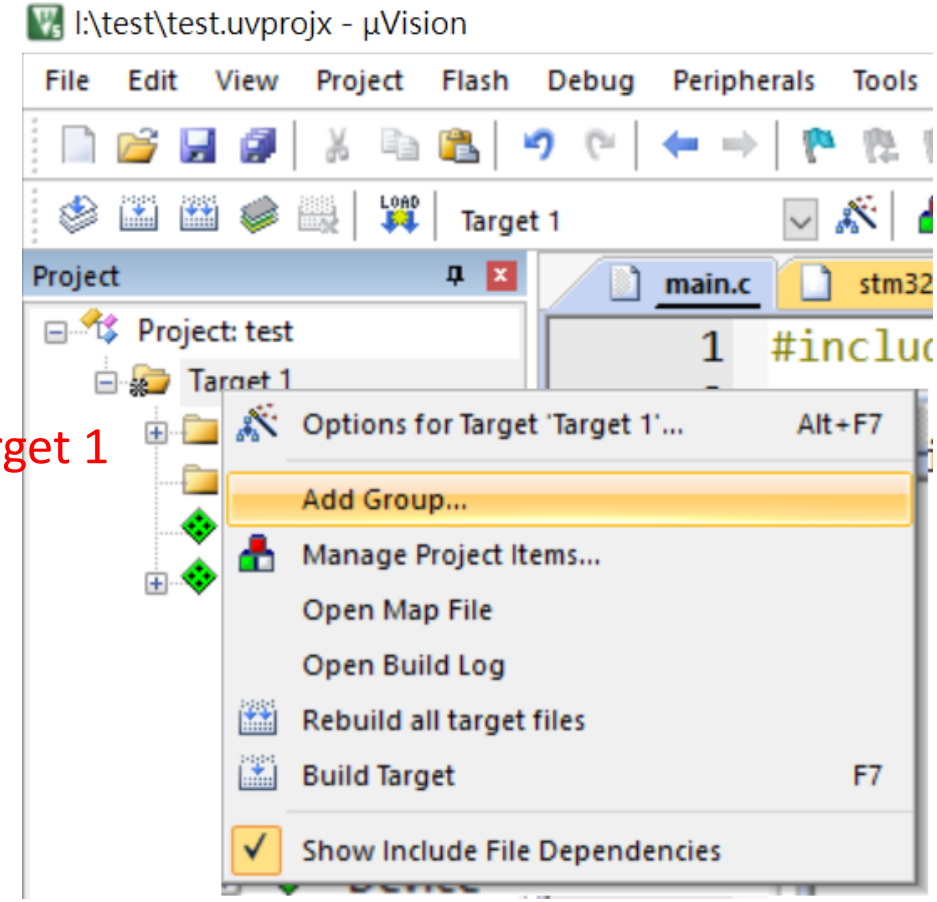
- Type STM32F401xx, USE_STDPERIPH_DRIVER.



Step8. Add source file

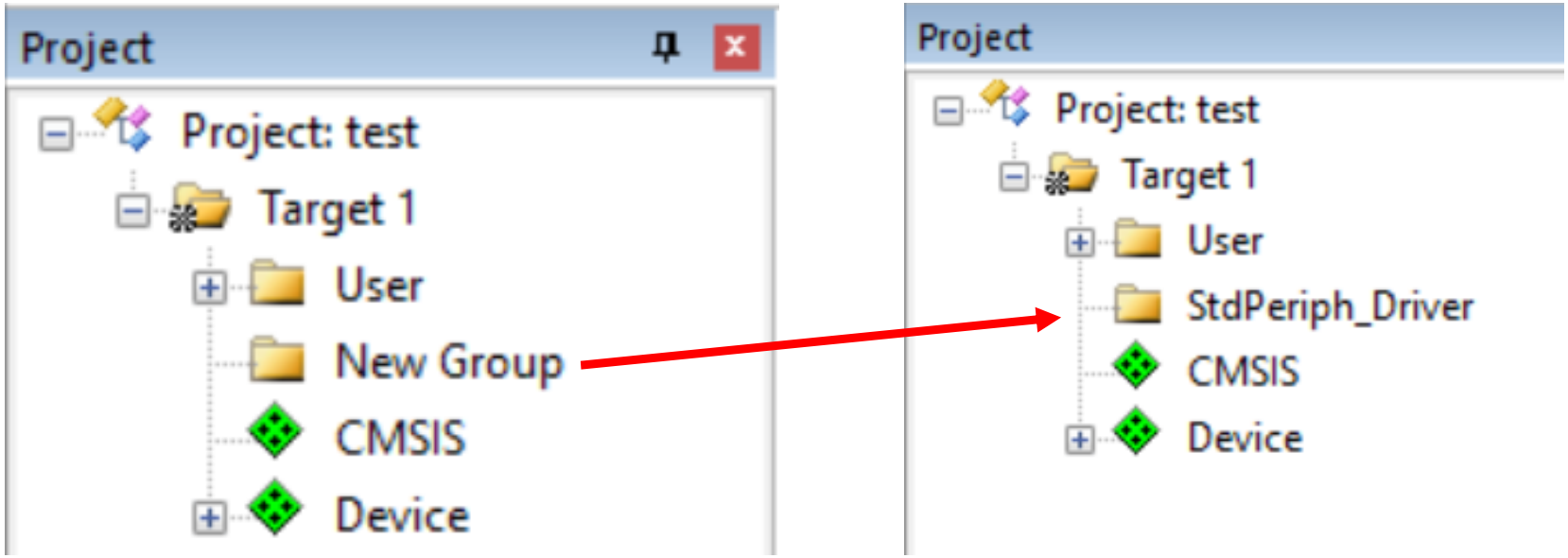
- Add group.

Right click on Target 1

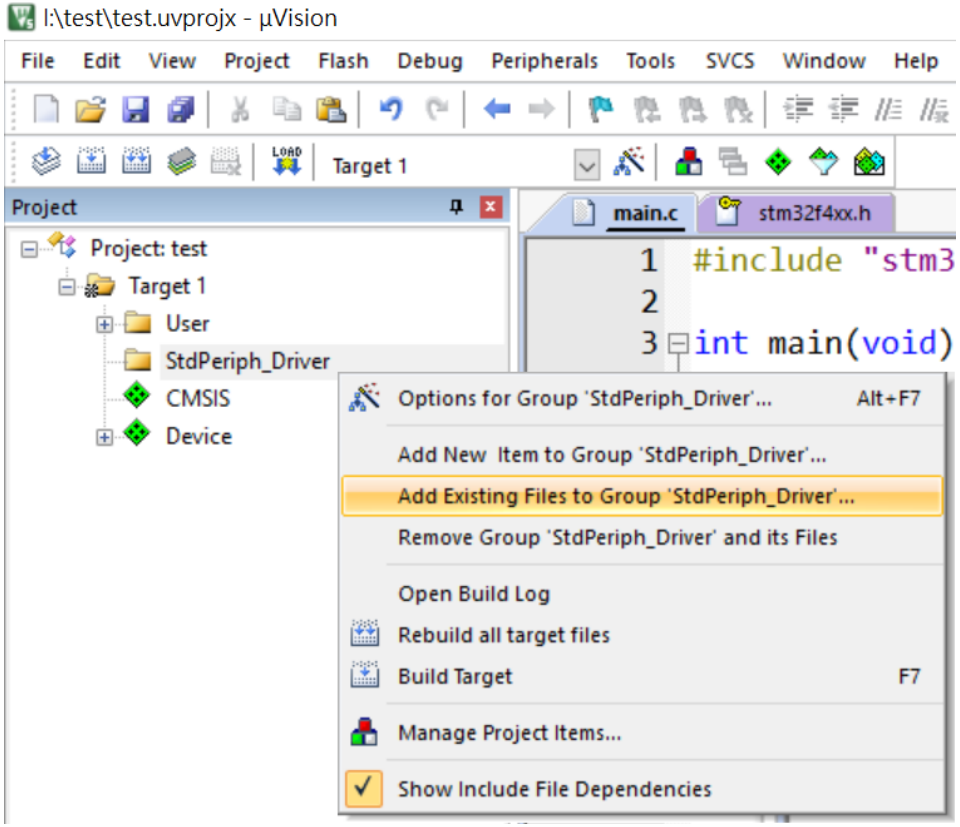


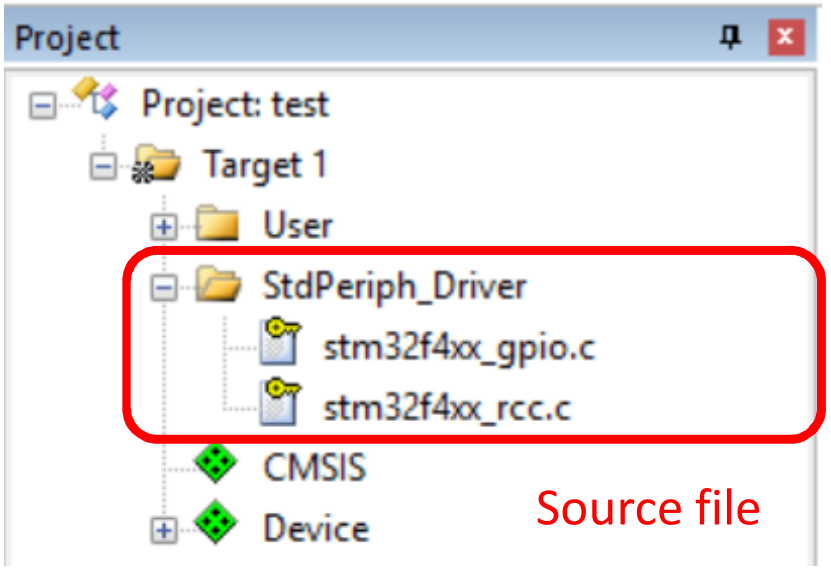
Step8. Add source file

- Rename New Group to StdPeriph_Driver.



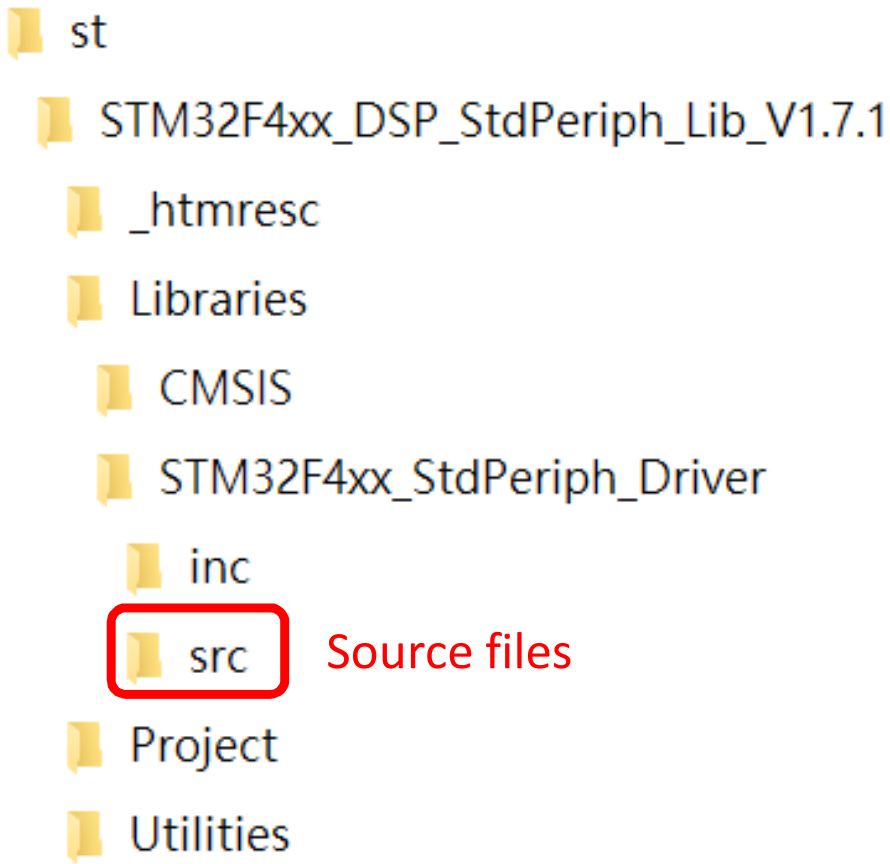
Step8. Add source file





Step8. Add source file

- Source files will be in here.



Step9. Test

- Blinky LED.

main.c

```c
#include "stm32f4xx.h"                    // Device header

int main(void){
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);

    GPIO_InitTypeDef GPIO_InitStruct;
    GPIO_InitStruct.GPIO_Pin = GPIO_Pin_5;
    GPIO_InitStruct.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStruct.GPIO_OType = GPIO_OType_PP;
    GPIO_Init(GPIOA, &GPIO_InitStruct);

    while(1){
        GPIO_ToggleBits(GPIOA, GPIO_Pin_5);
        for(int i=0; i<1600000; i++);
    }
    return 0;
}
```

Exercise

Transfer previous code using StdPeriph Library.

main.c | stm32f4xx.h

```c
#include "stm32f4xx.h"                    // Device header

int main(void){
    RCC->AHB1ENR |= RCC_AHB1ENR_GPIOCEN;

    GPIOC->MODER |= (0x1 << 2*5);

    while(1){
        GPIOC->BSRRH = (0x1 << 5);
        for(int i=0; i<1600000; i++);
        GPIOC->BSRRL = (0x1 << 5);
        for(int i=0; i<1600000; i++);
    }
    return 0;
}
```

main.c

```c
#include "stm32f4xx.h"                    // Device header

int main(void){
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);

    GPIO_InitTypeDef GPIO_InitStruct;
    GPIO_InitStruct.GPIO_Pin = GPIO_Pin_5;
    GPIO_InitStruct.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStruct.GPIO_OType = GPIO_OType_PP;
    GPIO_Init(GPIOA, &GPIO_InitStruct);

    while(1){
        GPIO_ToggleBits(GPIOA, GPIO_Pin_5);
        for(int i=0; i<1600000; i++);
    }
    return 0;
}
```

Performance comparison

	Code Size(Byte)	Diff.	Execute Cycle	Diff.
Low Level	4453		1337	
HAL	5427	21.9%	1912	43.0%
StdPeriph Library	6028	35.4%	2100	57.1%