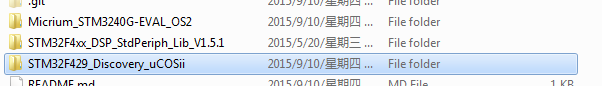
STM32F429 DISCOVERY board uCOSii transplantation Guide

1. Download STM3240G-EVAL\_OS2.zip from Micrium or from my github <https://github.com/lanniaoershi/STM32F429_Discovery_UCOSII_Transplantation/tree/master/Micrium_STM3240G-EVAL_OS2>

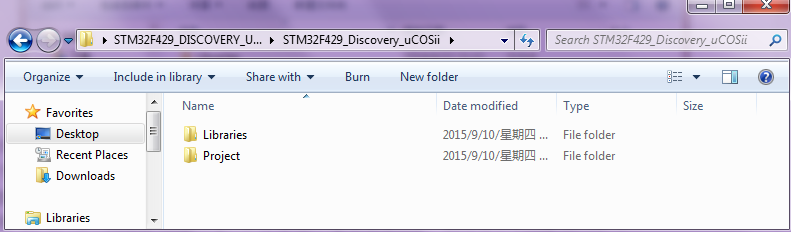
Download stm32f4\_dsp\_stdperiph\_lib.zip from ST or from my github

<https://github.com/lanniaoershi/STM32F429_Discovery_UCOSII_Transplantation/tree/master/STM32F4xx_DSP_StdPeriph_Lib_V1.5.1>

1. Create a folder for transplantation, my folder named “STM32F429\_Discovery\_uCOSii”



Create two folders in “STM32F429\_Discovery\_uCOSii”



Create folder “CMSIS” in Libraries folder,

Create folders “ARM-MDK” and “User” in Project folder.

1. Add file in to folder, put STM32F4xx\_DSP\_StdPeriph\_Lib\_V1.5.1\Libraries\STM32F4xx\_StdPeriph\_Driver whole folder in to Libraries folder create before.

Put STM32F4xx\_DSP\_StdPeriph\_Lib\_V1.5.1\Libraries\CMSIS\Device\ST\STM32F4xx\Source\Templates\arm\startup\_stm32f429\_439xx.s in to STM32F429\_Discovery\_uCOSii\Libraries\CMSIS

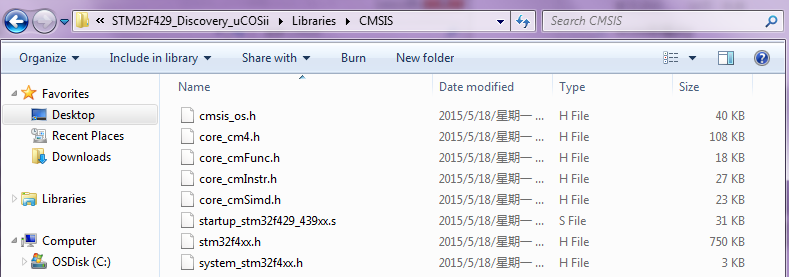
Put

STM32F4xx\_DSP\_StdPeriph\_Lib\_V1.5.1\Libraries\CMSIS\Device\ST\STM32F4xx\Include stm32f4xx.h and system\_stm32f4xx.h into STM32F429\_Discovery\_uCOSii\Libraries\CMSIS

Put STM32F4xx\_DSP\_StdPeriph\_Lib\_V1.5.1\Libraries\CMSIS\Include\core\_cm4.h and core\_cmFunc.h and core\_cmInstr.h and core\_cmSimd.h in to STM32F429\_Discovery\_uCOSii\Libraries\CMSIS

Put STM32F4xx\_DSP\_StdPeriph\_Lib\_V1.5.1\Libraries\CMSIS\RTOS\Template\cmsis\_os.h in to STM32F429\_Discovery\_uCOSii\Libraries\CMSIS

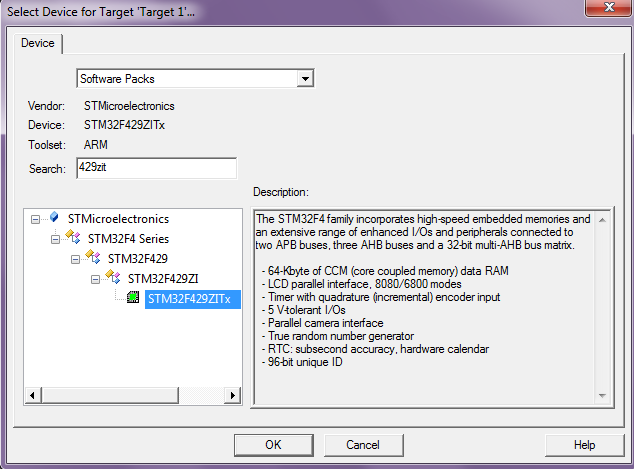
And CMSIS folder will look like



Put STM32F4xx\_DSP\_StdPeriph\_Lib\_V1.5.1\Project\STM32F4xx\_StdPeriph\_Templates

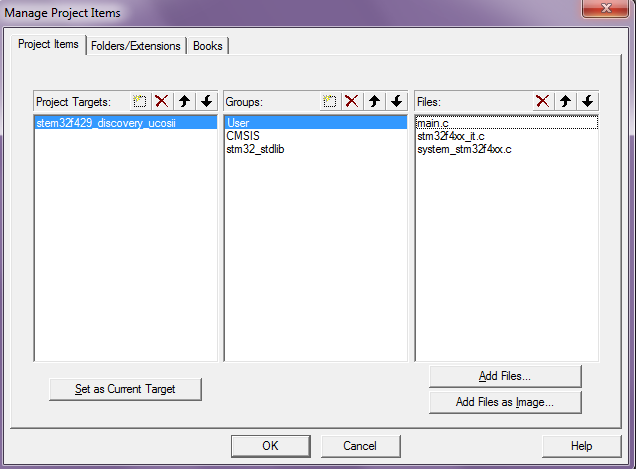
main.c main.h stm32f4xx\_conf.h stm32f4xx\_it.c stm32f4xx\_it.h system\_stm32f4xx.c in to STM32F429\_Discovery\_uCOSii\Project/User

1. Create new Keil project, and chose 429ZITx

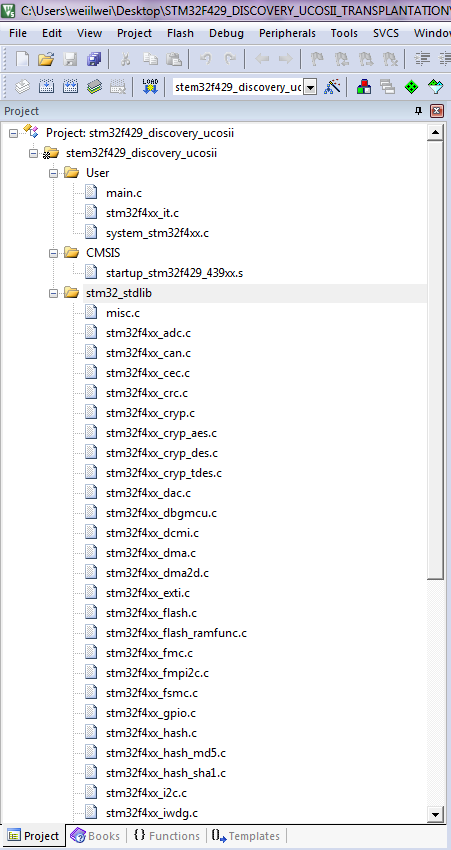


Then “OK”, if asking put startup file in to project, chose no, and now have a clean 429ZITx project.

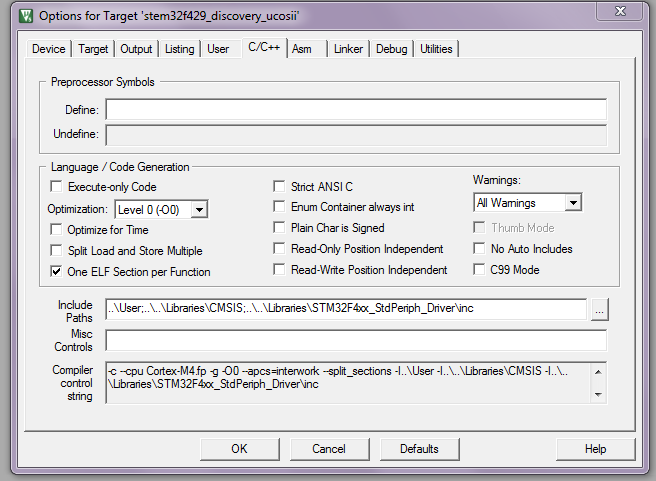
Project-> manage-> Project Items… create group and add files



Finally project structure look like



Add build include path

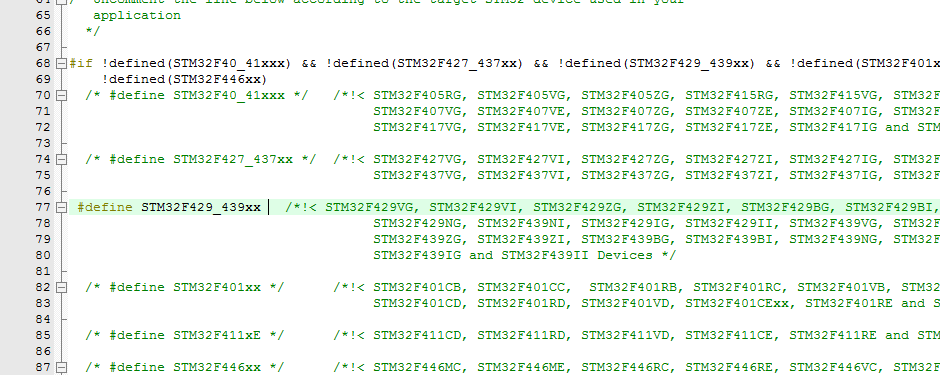


Then build, it should be have some error, here listed all error I have met

1. Not specific product

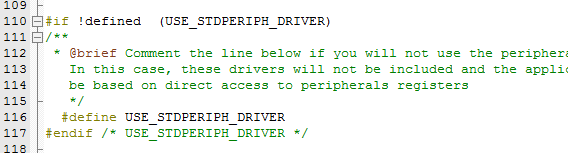


Solution: open stm32f4xx.h in STM32F429\_Discovery\_uCOSii\Libraries\CMSIS uncomment “#define STM32F429\_439xx”



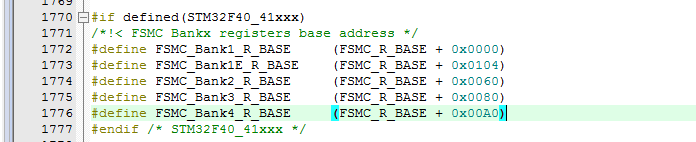
1. identifier "RCC\_ClocksTypeDef" is undefined

Cause by stm32f4xx\_rcc.h not include, and because in stm32f4xx.h “USE\_STDPERIPH\_DRIVER” is commented, uncomment it.



1. identifier "FSMC\_XXXXXXXXX" is undefined

In file stm32f4xx.h FSMC\_XXXX was define for F40\_41xx



Delete stm32f4xx\_fsmc.c from Libraries or un-include from build

After solved error, build should be done.

1. Option for debugger, stm32f429 discovery use on board st-link download with SWD

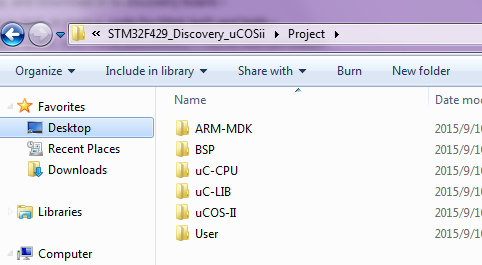
Set it up, and download in to discovery board.

In order to test, in main.c, code for blink led3 and led4.

Tag v0.1 release this test code.

1. Add UCOSii file to project

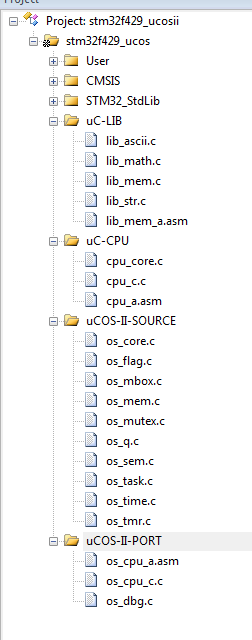
Copy Micrium\_STM3240G-EVAL\_OS2/software/ uC-CPU uC-LIB uCOS-II to STM32F429\_Discovery\_uCOSii\Project



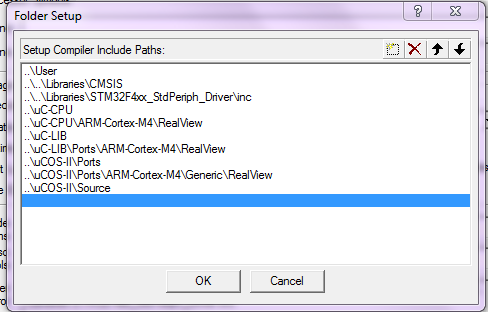
Copy Micrium\_STM3240G-EVAL\_OS2\Software\EvalBoards\ST\STM3240G-EVAL\uCOS-II\KeilMDK app\_cfg.h, lib\_cfg.h in to STM32F429\_Discovery\_uCOSii\Project\User

Copy Micrium\_STM3240G-EVAL\_OS2\Software\EvalBoards\ST\STM3240G-EVAL\uCOS-II\ cpu\_cfg.h, os\_cfg.h in to STM32F429\_Discovery\_uCOSii\Project\User

Add file to project like

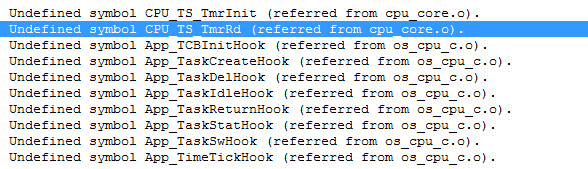


Then remember add include path



Then build, here may have some error

I met follow error



Solution: Change file User/cpu\_cfg.h line 99 “#define CPU\_CFG\_TS\_32\_EN DEF\_ENABLED” to “#define CPU\_CFG\_TS\_32\_EN DEF\_DISABLED”

Change file User/os\_cfg.h line 30 “#define OS\_APP\_HOOKS\_EN 1u” to “#define OS\_APP\_HOOKS\_EN 0u”

1. Using ucos create task, here will meet some exceptions, follow to see how.

Create simple task by:

led\_init();

OSInit();

OSTaskCreate(led\_blink\_task, 0, &led\_blink\_task\_stk[LED\_BLINK\_STK\_SIZE-1],10);

OSStart();

Refer to project to check detail.

Exceptions:

1. After OSStart() led not blink, tracking to “OSStartHang” in os\_cpu\_a.asm, means system hanging infinitely.

Solution:

Change all “PendSV\_Handler”(line: 83, 232, 233) in file startup\_stm32f429\_439xx.s to “OS\_CPU\_PendSVHandler” in order to match with os\_cpu\_a.asm and os\_cpu.h, this is only one simple solution. But it can work.

1. In (1), we are still using Delay(); to delay time, ucos has provide OSTimeDly() and OSTimeDlyHMSM(); if direct use this function replace with Delay(); led will not blink;

Solution: