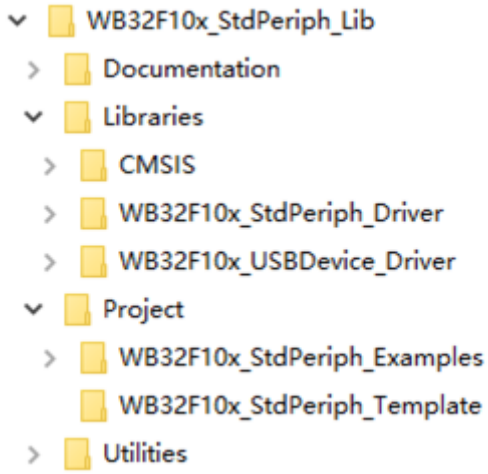
WB32F10x
Getting Started Development

Westberry Technology (ChangZhou) Corp., ltd

Contents

1 WB32F10x Firmware Library Introduction

The WB32F10x Standard Peripherals Library is structured as follows:



This library contains a collection of routines, data structures and macros covering the features of WB32F10x peripherals.

Documentation for the WB32F10X firmware library is stored in the Documentation folder.

- **Libraries** contains three subfolders, **CMSIS**, **WB32F10x_StdPeripher_Driver** and **WB32F10x_USBDevice_Driver**.
- The **CMSIS** stores the startup files, headers, etc. associated with the **WB32F10x** chip.
- The **WB32F10X_STDPeripher_Driver** contains the source code of the **WB32F10x** firmware library, which is related to the standard peripheral.
- The **WB32F10x_USBDevice_Driver** contains the WB32F10X USB device protocol stack code.

The Project folder contains two subfolders, **WB32F10X_StdPeripher_Examples** and **WB32F10X_StdPeripher_Template**.

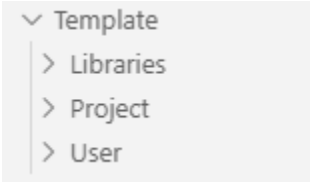
- The **WB32F10X_StdPeripher_Example** folder contains the official source code of the firmware provided by **WestberryTech** for reference.
- The **WB32F10X_StdPeripher_Template** folder contains the file templates needed to create the new project.

The **Utilities** folder holds common source code.

2 Use the keil mdk to create the project

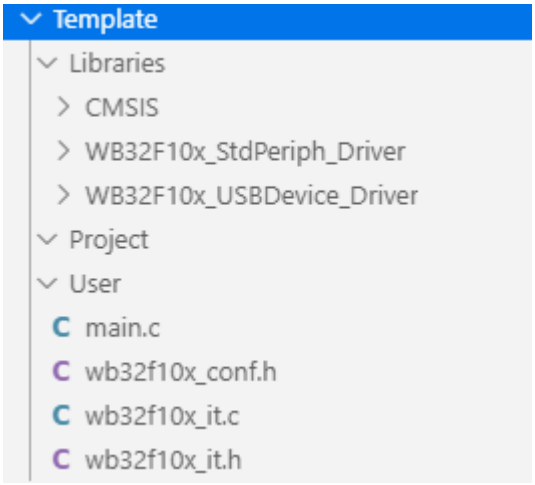
Step 01, Create a new folder named **Template** to hold the entire project.

Step 02, Create **Libraries**, **Project** and **User** subfolders in the Template folder(You can also make the project directory structure according to your own habits.).

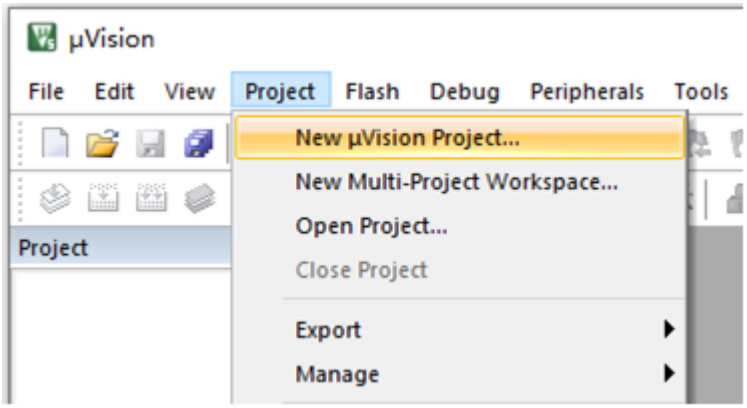


Step 03, Copy the contents of the **Libraries** floder in the **WB32F10x_StdPeriph_Lib** floder to the Template\Libraries folder.

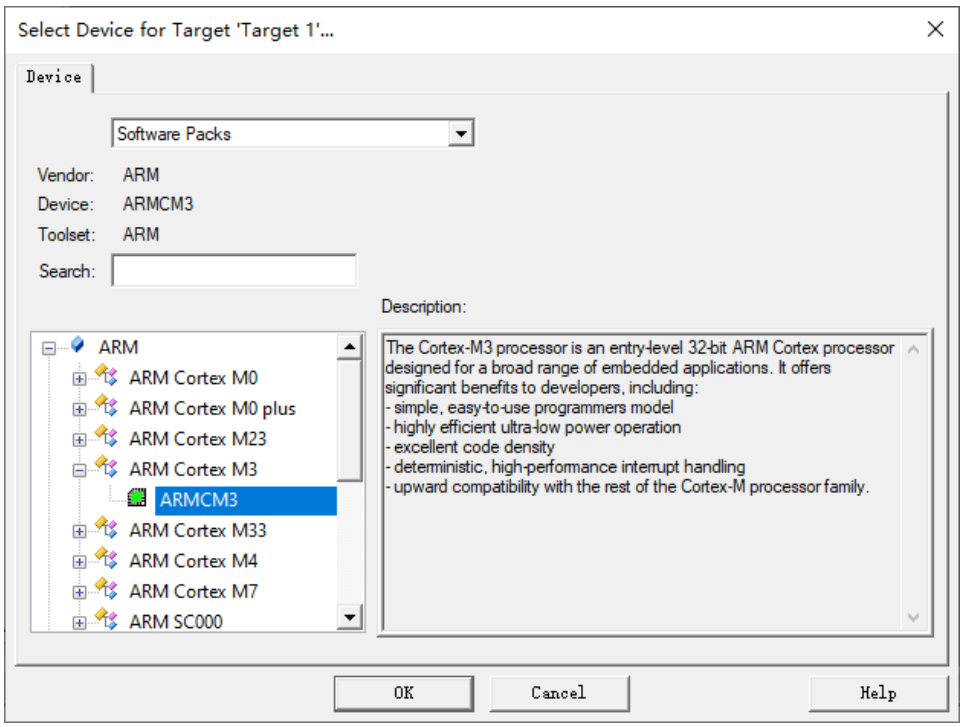
Step 04, Copy the contents of the **Project\ WB32F10X_STDPeripher_Template** folder from the **WB32F10x_StdPeriph_Lib** floder to the **Template\User** folder.



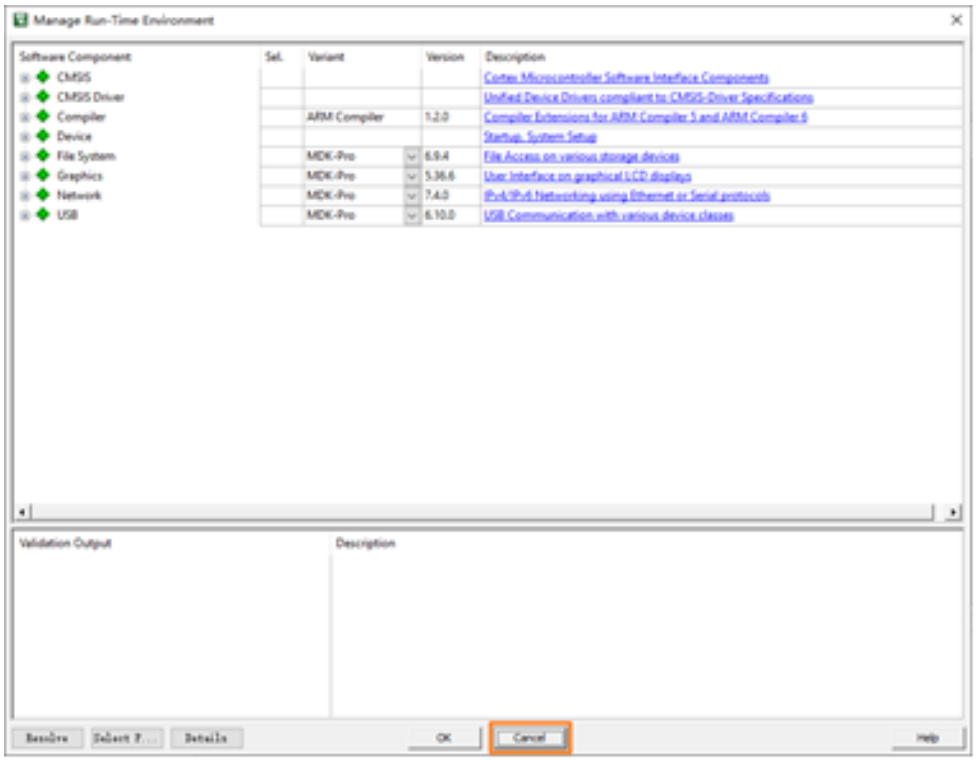
Step 05, Open the **Keil MDK** software， Click **Project-> New uVision Project...** as shown:



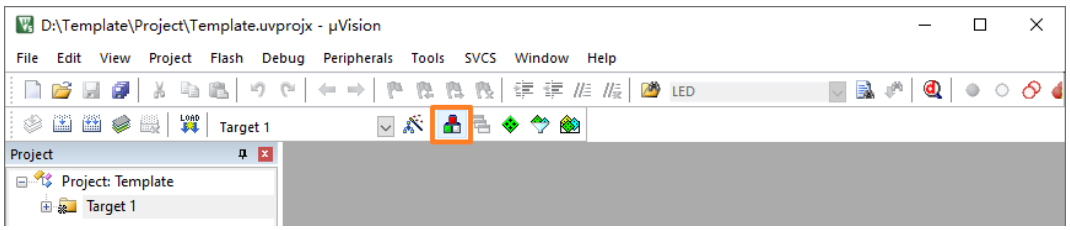
Step 06, Select the device to use for the project as **ARMCM3**, and then click **OK**.

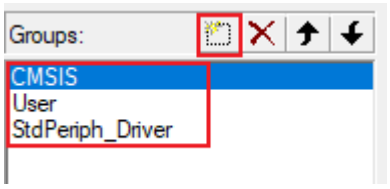


Step 07, You will see the **Manage Run-Time Environment** dialog pop up and you can click Cancel to close the dialog box.



Step 08, You can set up three Groups: **CMSIS**, **User**, **StdDriver** before import the Standard Peripherals Library file.





Add files to the Group

Add to the CMSIS Group:

Template\Libraries\CMSIS\Device\WB\WB32F10x\startup\arm\startup_wb32f10x.s

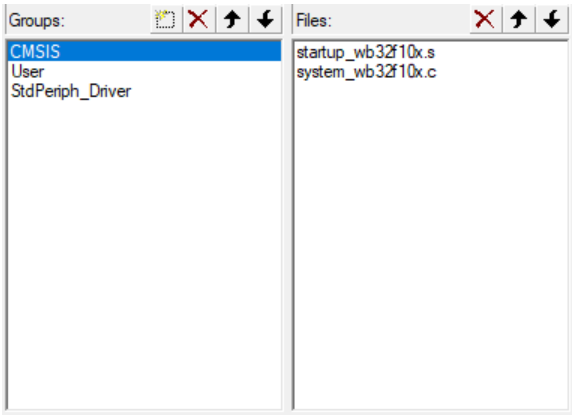
Template\Libraries\CMSIS\Device\WB\WB32F10x\system_wb32f10x.c

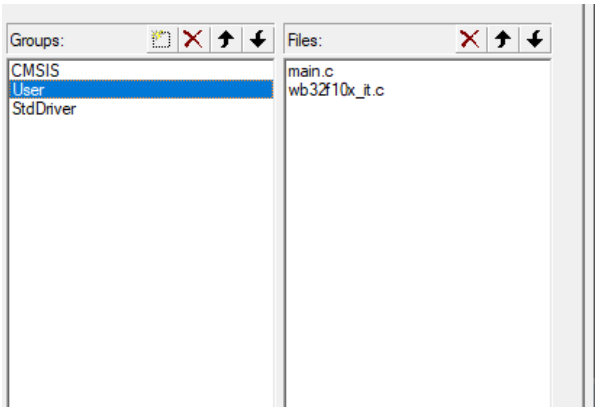
Add to the User Group:

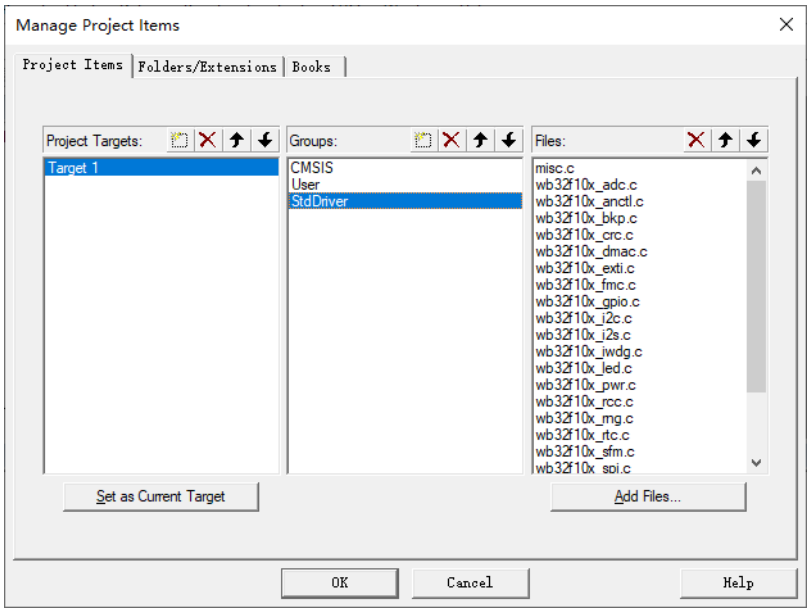
Template\User\main.c

Template\User\wb32f10x_it.c

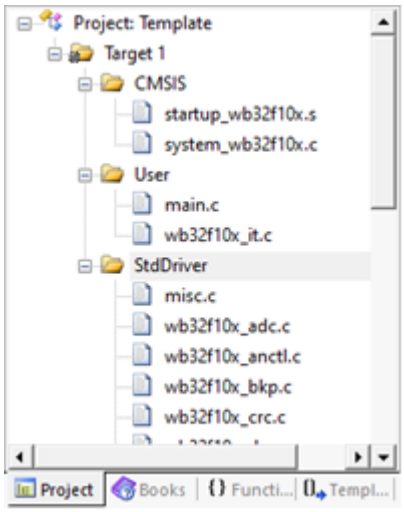
Add all the **.c**(source code)files in the Template\Libraries\WB32F10x_StdPeriph_Driver\ SRC folder to the **STDDrive**r Group



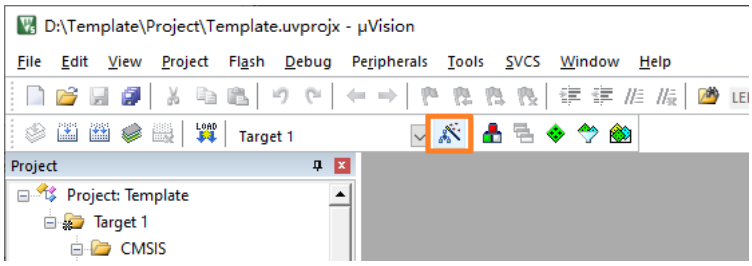




Finally,the whole project is structured as follows:



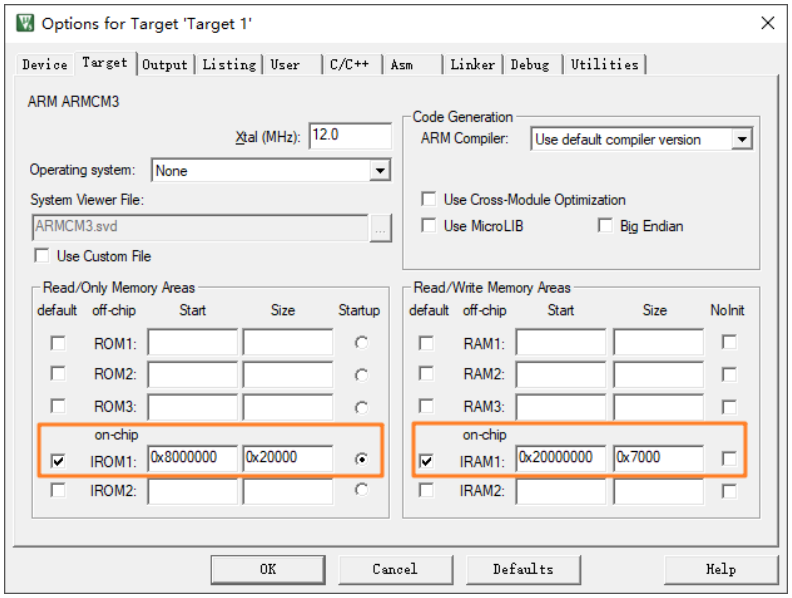
Step 09, Click the icon below to open the **Options for Target** dialog Box.



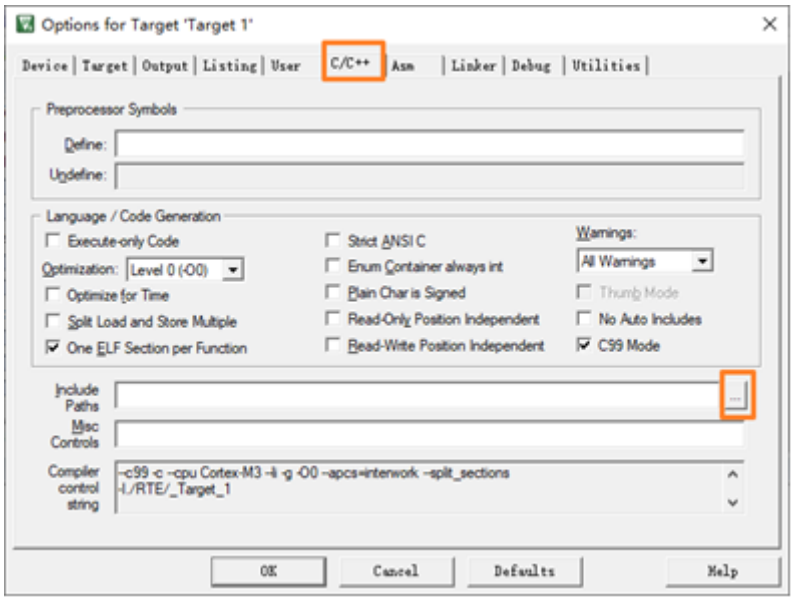
Configure Read/Only Memeory Areas and Read/Write Memory Areas (Configure the starting address and size of Flash and SRAM.).

Note： *Configure the **Flash** and **SRAM** sizes based on your IC type. 128KB Flash and 28KB SRAM are used as an example (see the following table for capacity configurations of other **Product Code**).*

Product Code	Flash Size	SRAM Size
WB32F10xx6	0x8000 (32KB)	0x3000 (12KB)
WB32F10xx8	0x10000 (64KB)	0x5000 (20KB)
WB32F10xx9	0x18000 (96KB)	0x7000 (28KB)
WB32F10xxB	0x20000 (128KB)	0x7000 (28KB)
WB32F10xxC	0x40000 (256KB)	0x9000 (36KB)



Step 10, Configure the header file include path for the project in the **C/C++** TAB.



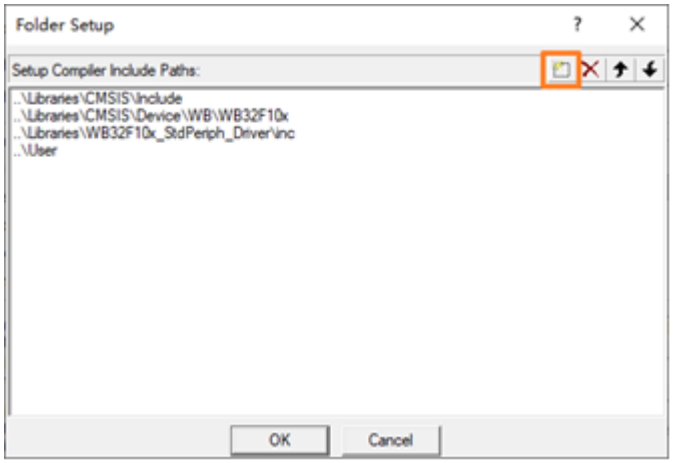
Add the paths as follows:

..\Libraries\CMSIS\Include

..\Libraries\CMSIS\Device\WB\WB32F10x

..\Libraries\WB32F10x_StdPeriph_Driver\inc

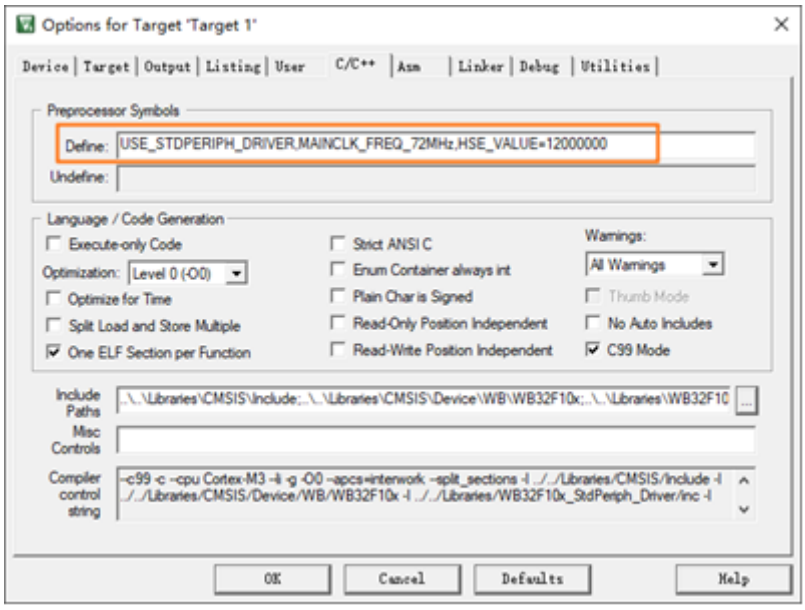
..\User



Step 11, Include **USE_STDALTER_DRIVER** and **MAINCLK_FREQ_72MHZ** definitions in the Preprocessor Symbols (see below for details on these two definitions).

USE_STDPERIPH_DRIVER definition indicates using Standard Peripherals Library
MAINCLK_FREQ_72MHz definition indicates using the 72MHz Main Clock configuration function predefined in system_wb32f10x.c to configure the Main Clock.
HSE_VALUE=12000000 definition indicates the external crystal frequency used is 12MHz.

Note: *The main frequency cannot exceed the maximum frequency supported by your IC type.*

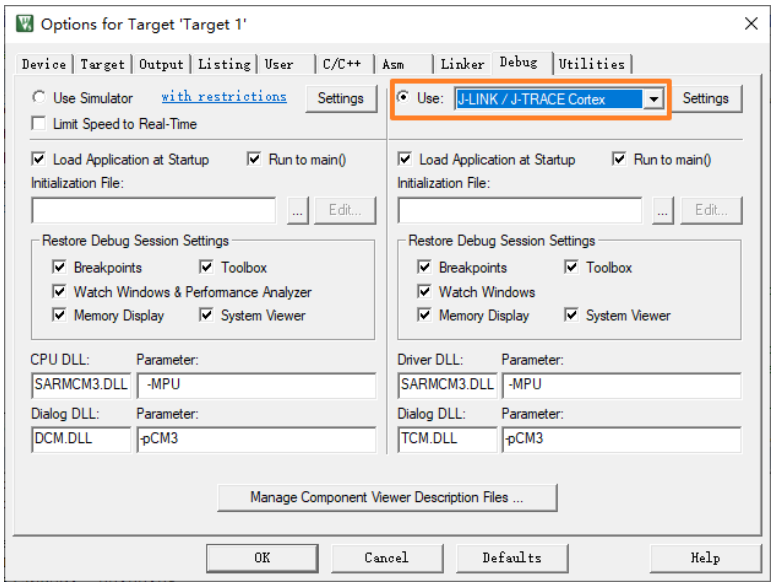


Step 12, Click **OK**. At this point, the project setup configuration is complete. Next we will configure debugging.

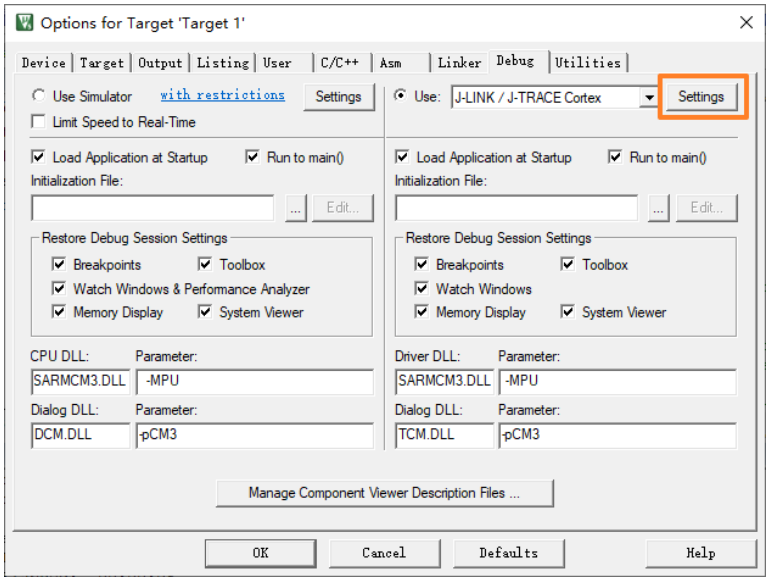
Step 13, Because the WB32F10x is an ARM Cortex-M3 chip, you can use a Cortex-M3-enabled debugger (e.g., JLink, Ulink, CMSIS-DAP, etc.) to debug your applications. Let's use JLink as an example to demonstrate the configuration debugging of WB32F10x

Step 14, Connect A JLink tool to your computer, use the JLink SWD interface to connect to the WB32F10X chip, and power the chip.

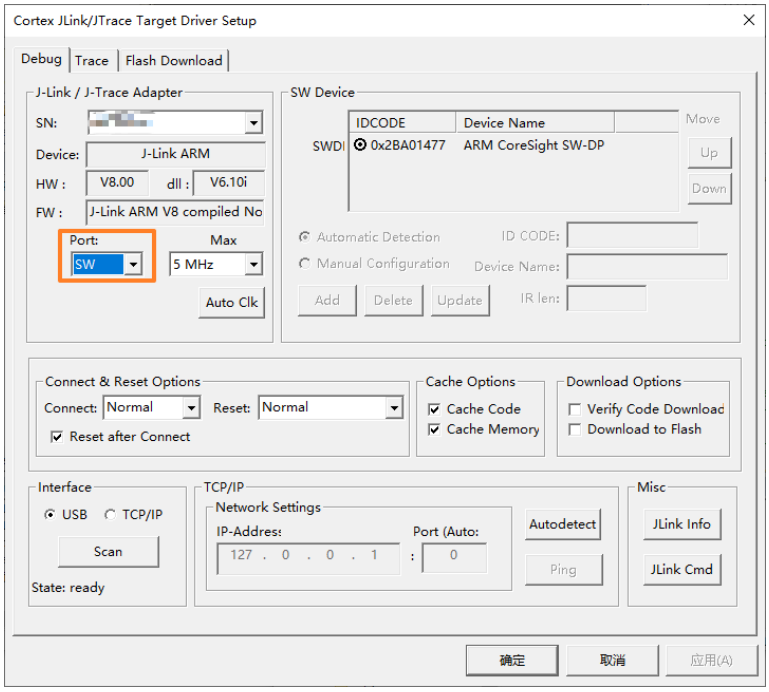
Step 15, Open the **Options for Target** dialog box, switch to the **Debug** TAB, and choose to **J-Link/J-TRACE Cortex**.



Step 16, Click Settings to configure the debugger.

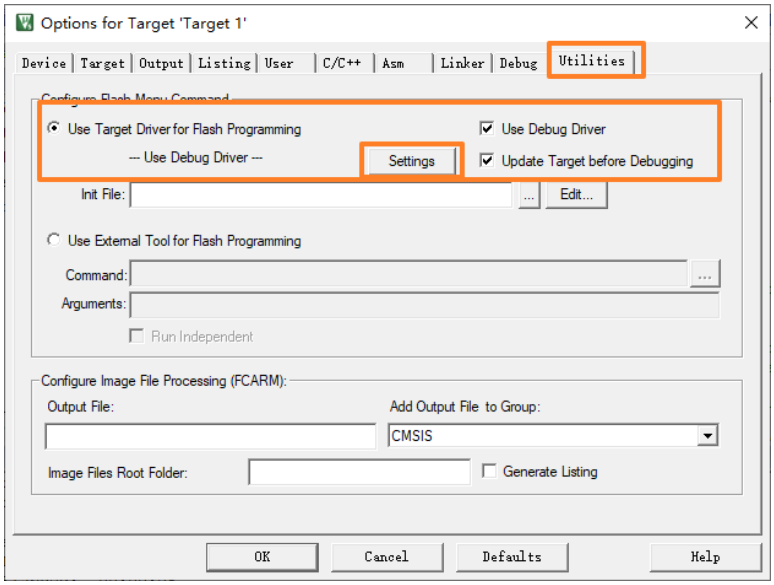


You can see on the right that JLink has detected the WB32F10X chip after select the SW interface.

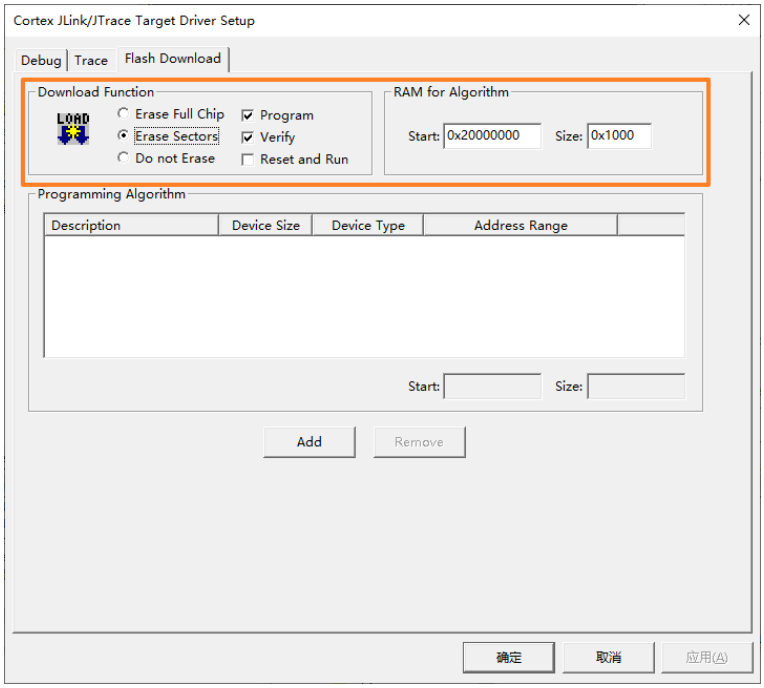


Step 17, Then click 确定**(OK)** to exit.In the Utilities TAB,

Step 18, You need to do the setup shown in the following figure:

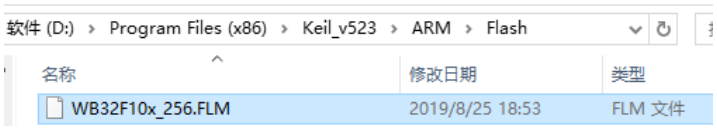


Then click the **Settings** button to open the Flash Programming configuration dialog box. Do the configuration as shown.

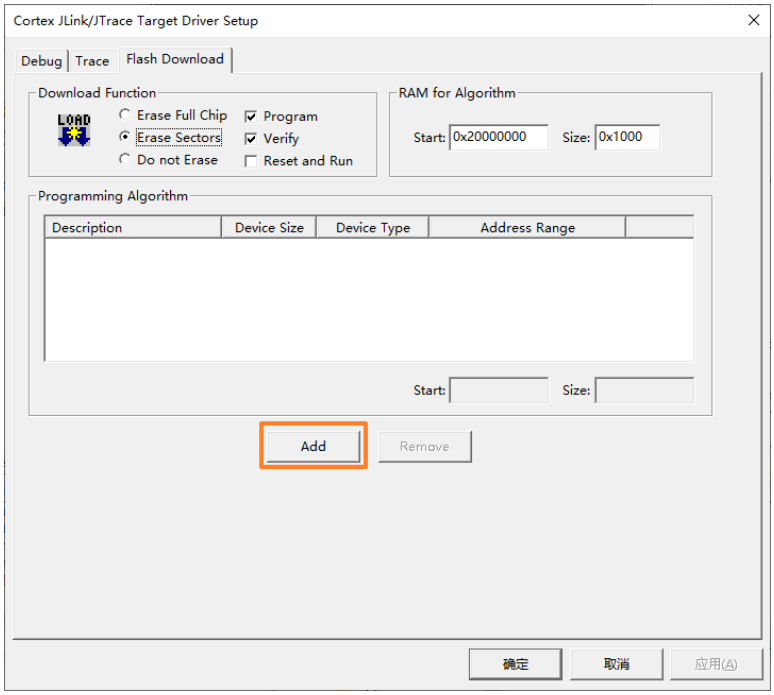


Step 19, Programming Algorithm configuration

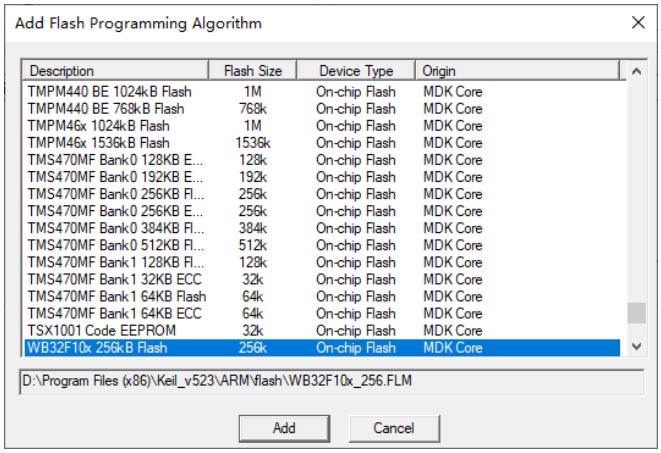
Step 20, Copy the **WB32F10X_256. FLM** file provided by **WestberryTech** to the corresponding folder in the installation directory of Keil MDK
(on my computer the path is D:\Program Files (x86)\Keil_v523\ARM\Flash)



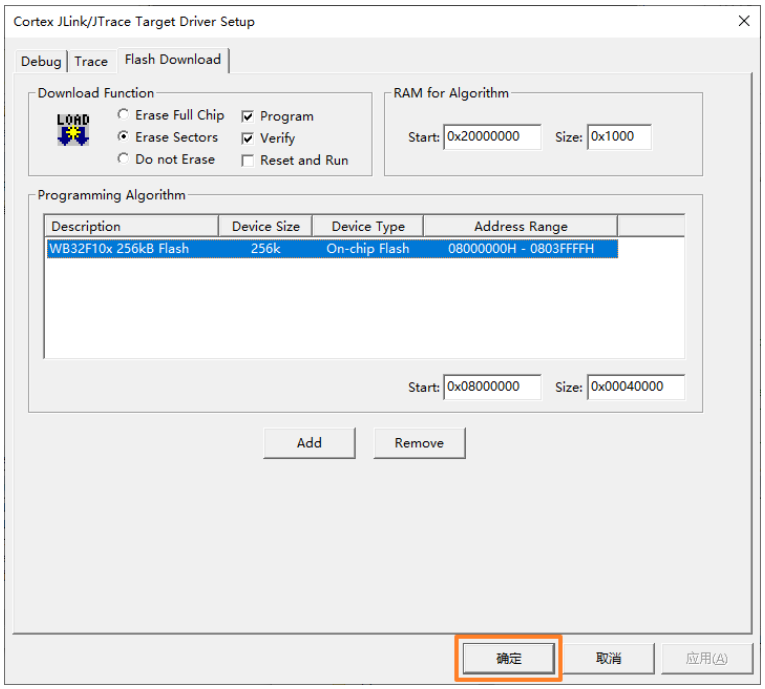
Then click the **Add** button in the **Programming Algorithm** dialog box



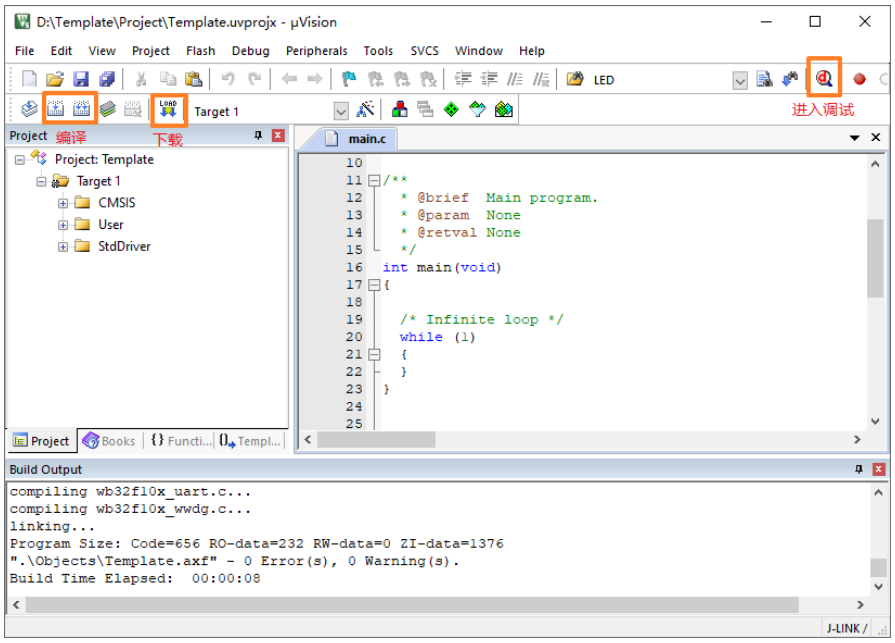
Step 21, Locate the Programming Algorithm named **WB32F10X 256KB Flash** and click **Add**.



Then click **确定(OK)**.

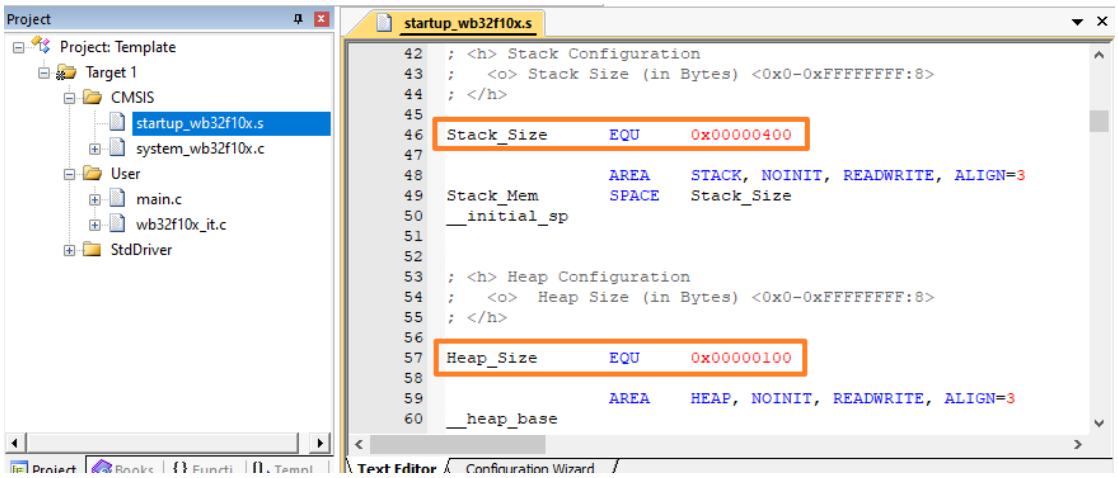


At this point, you are ready to **compile, download**, and **debug** the program. The configuration of the firmware library is described in the following section.

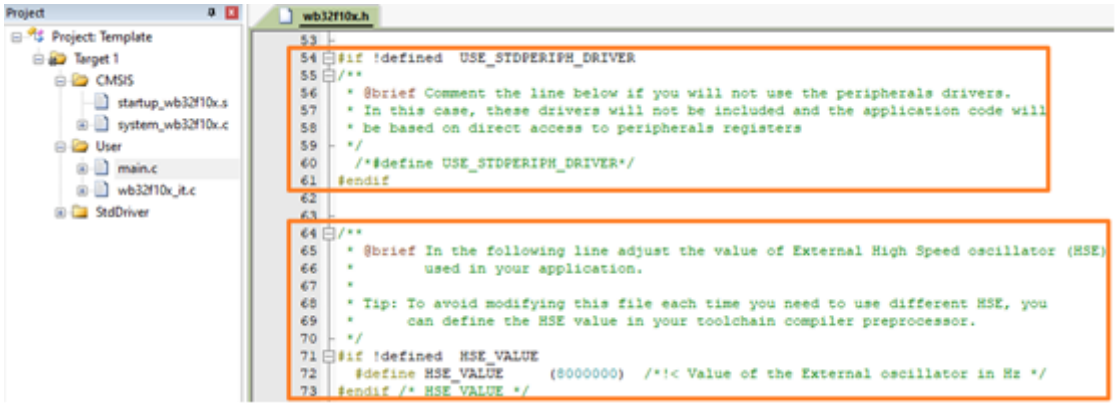


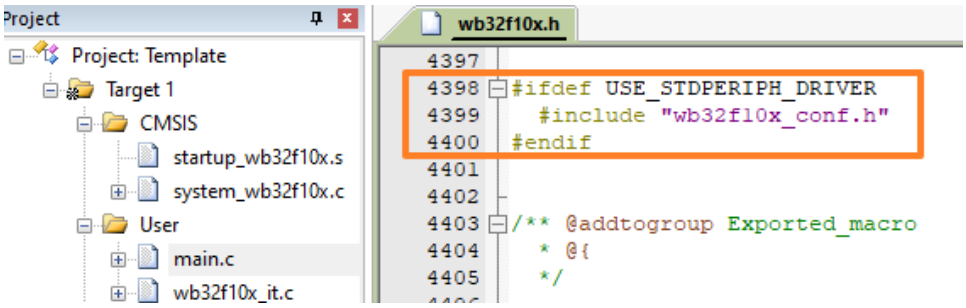
3 Details of WB32F10x Standard Peripherals Library

The size of the application stack and heap can be configured in startup_wb32f10x.s as follows:



You may want to note that there are two macro definitions in **wb32f10x.h**.



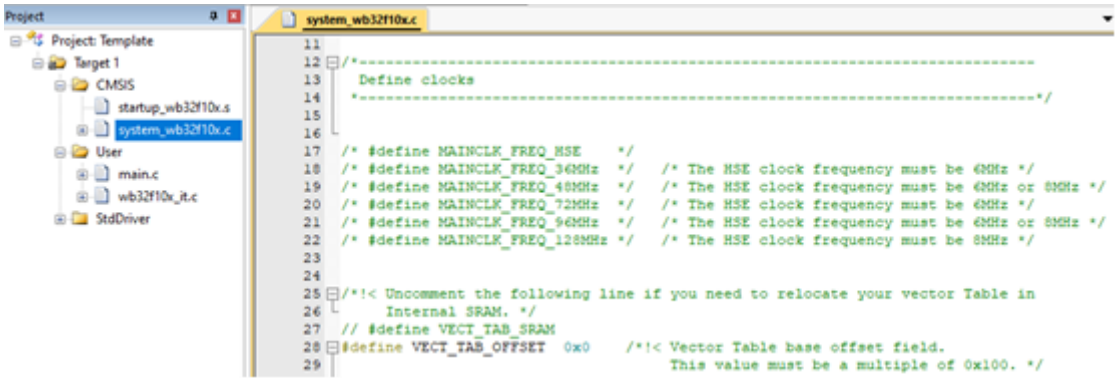


USE_STDPERIPH_DRIVER means that the application needs to use a peripheral driver from the firmware library and will include the WB32F10x_conf.h header file in the project.

HSE_VALUE is used to specify the frequency of the external crystal on the WB32F10x chip. By default, the external crystal HSE frequency of the Peripherals library is 8MH.

If you are using an external crystal oscillator other than 8MHz, be sure to modify or overwrite the definition in the compiler's global predefined!!

You may want to focus on a few definitions in **system_wb32f10x.c**.



MAINCLK_FREQ_* ;These macros define the frequency of the chip master clock after the program is started. You can only choose to define one of them (if none is defined, the chip master clock is MHSI). You can define it at the compiler global predefined. These macro definitions are required for the external crystal oscillator of the chip.

If you are defining MAINCLK_FREQ_72MHz, the external crystal frequency of the chip must be 6MHz or 12MHz. (**Note:** *the definition of HSE_VALUE must be overwrote as well*).

VECT_TAB_SRAM:This macro maps the interrupt vector to the SRAM(This macro needs to be defined for projects running in SRAM).

VECT_TAB_OFFSET: This macro is used to set the offset of the starting address for the interrupt vector (Relative to the starting address of Flash or SRAM.).

Revision History

Revision	Date	Description
1.2	2022/7/5	Initial Release

IMPORTANT NOTICE

Information in this document is provided solely in connection with WB products. This document, including any product of WB described in this document (the "Product"), is owned by WB under the intellectualproperty laws and treaties of the People's Republic of China and other jurisdictions worldwide. Westberry Technology (ChangZhou) Corp., ltd and its subsidiaries("WB") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.WB does not assume any liability arising out of the application or use of any Product described in this document. Purchasers are solely responsible for the choice, selection and use of the WB products and services described herein, and WB assumes no liability whatsoever relating to the choice, selection or use of the WB products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by WB for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed or manufactured for ordinary business, industrial, personal, or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage.

Resale of WB products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by WB for the WB product or service described herein and shall not create or extend in any manner whatsoever, anyliability of WB.