1. Introduction to Testing Platform

Development Board : CH32F103C8T6 and CH32F203C8T6 Board

MCU : CH32F103C8T6 \ CH32F203C8T6

Frequency : 72MHz(F103) \ 144MHz(F203)

2. Pin connection instructions

The SPI testing program (including software SPI and hardware SPI) or software IIC testing program of CH32F103C8T6 can be directly plugged into the CH32F103C8T6 development board (see figure below).

The hardware IIC test program for CH32F103C8T6 or all test programs for CH32F203C8T6 can only be connected using DuPont cables.

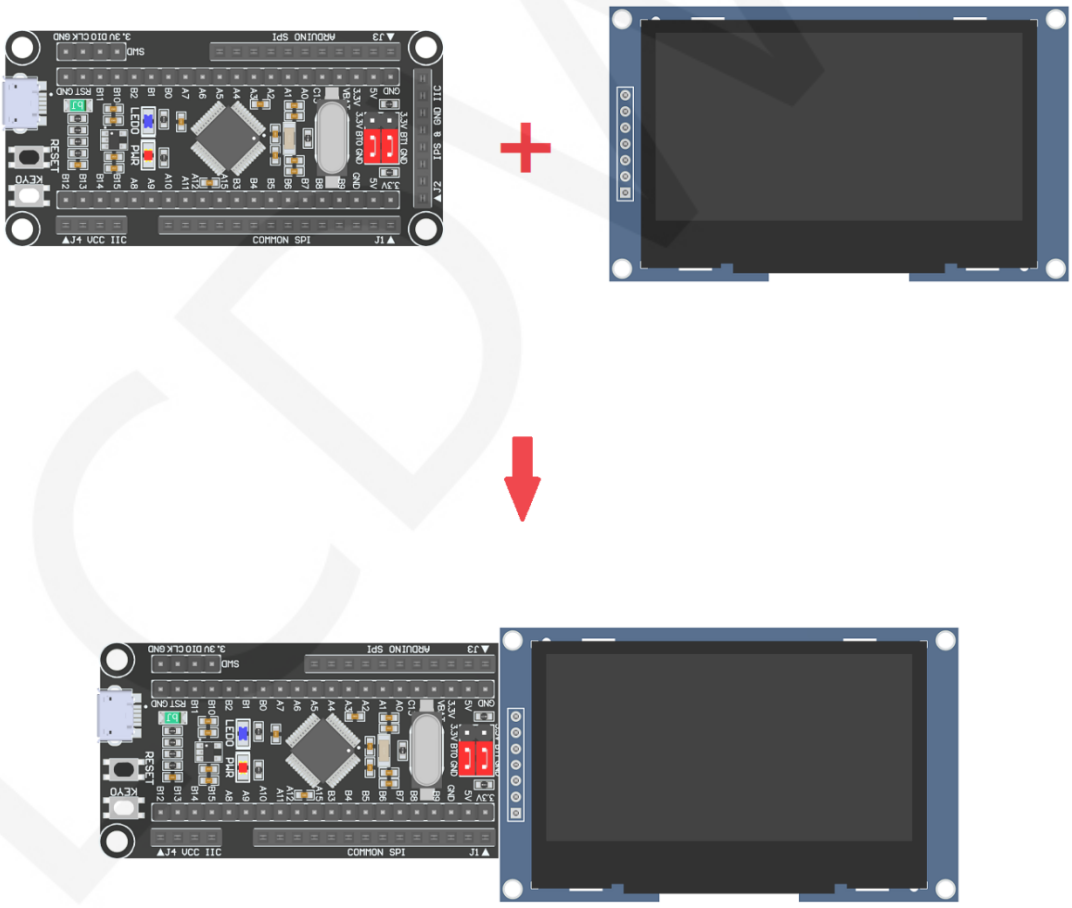


Figure 1: Module Inline CH32F103C8T6 Development Board

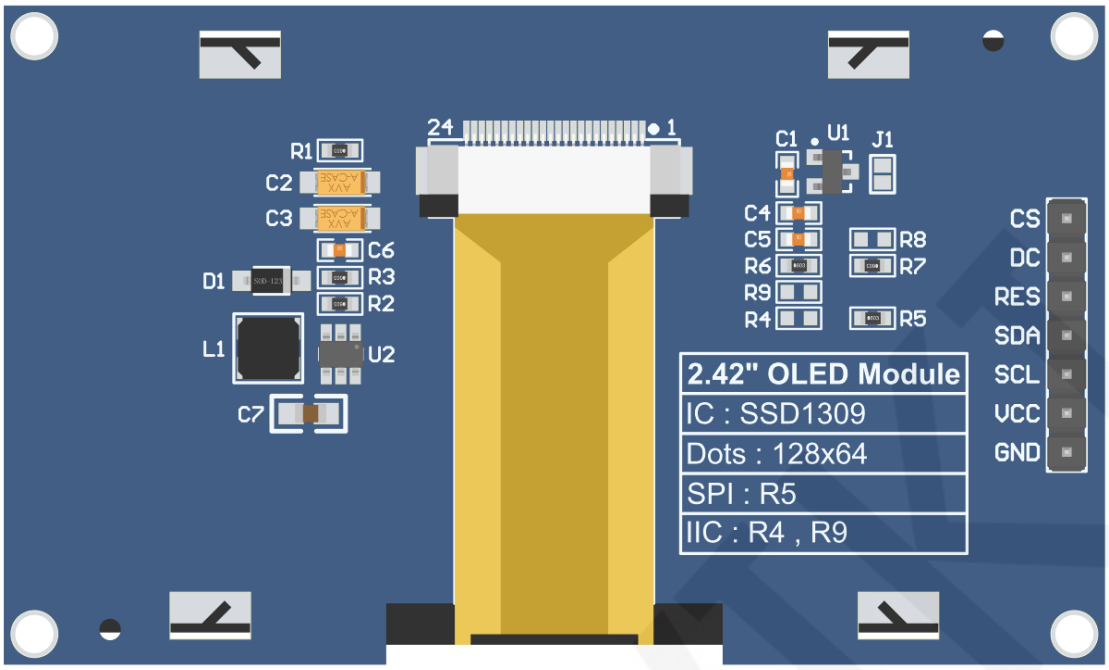


Figure 2 Module Back Pins

NOTE:

A. Connect to a 5V microcontroller, which can short circuit J1 to keep the IO voltage and IO high level consistent;

B. R8 is not soldered by default. If there is no need to control the CS pin, R8 solders the 0R resistor to keep the CS signal grounded;

C. If SPI communication mode is selected, R5 will weld 0R resistor, and R4 and R9 will be disconnected;

D. If IIC communication mode is selected, R4 and R9 will be welded with 0R resistor, and R5 will be disconnected;

CH32F103C8T6/CH32F203C8T6 SPI Test Program Pin Direct Insertion Instructions			
Number	**Module pins**	**Corresponding CH32F103/CH32F203 development board wiring pins**	**Remarks**
1	**GND**	GND	OLED screen power supply ground
2	**VCC**	5V/3.3V	OLED screen power supply positive
3	**SCL**	PA5	SPI bus clock signal

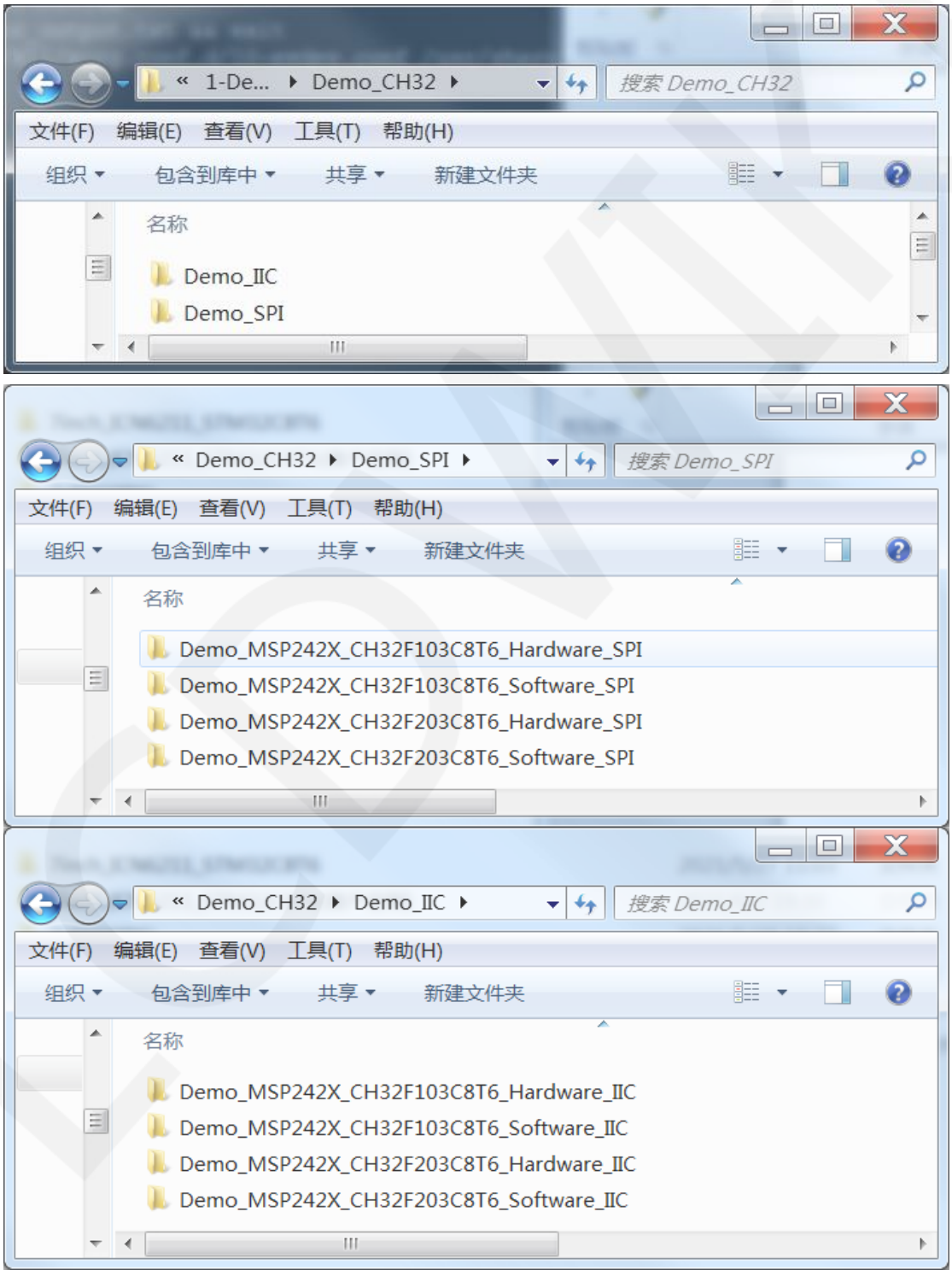
4	**SDA**	PA7	SPI bus write data signal
5	**RES**	PB8	OLED screen reset control signal, low-level reset
6	**DC**	PB7	OLED screen command/data selection control signal High level: data, low level: command
7	**CS**	PB9	OLED screen chip selection control signal, effective at low level (if welding R8, CS pin may not be connected)

CH32F103C8T6/CH32F203C8T6 IIC Test Program Pin Direct Insertion Instructions

Number	Module pins	Corresponding CH32F103/CH32F203 development board wiring pins		Remarks
		Hardware IIC	Software IIC	
1	**GND**	GND		OLED screen power supply ground
2	**VCC**	5V/3.3V		OLED screen power supply positive
3	**SCL**	PB6	PA5	IIC bus clock signal
4	**SDA**	PB7	PA7	IIC bus data signal
5	**RES**	PB8/3.3V		OLED screen reset control signal, low-level reset (if no control is required, the RES pin can be connected to a high-level (3.3V))
6	**DC**	PB5/GND/3.3V	PB7/GND/3.3V	IIC bus selects signal from device address When connecting to the PB5 or PB7 pin, set it to low level: 0x78, and set it to high level: 0x7A Low level (connected to GND): 0x78, high level (connected to 3.3V): 0x7A
7	**CS**	PB9/GND		OLED screen chip selection control signal, effective at low levels When using IIC communication, there is no need for control. When connecting to PB9, PB9 must be set to low level or GND can be connected (if welding R8, CS pin can not be connected)

3. Demo Function Description

This sample program includes two types of MCU programs, CH32F103C8T6 and CH32F203C8T6. Each MCU program also includes hardware SPI and IIC, as well as software SPI and IIC functions, which are located in Demo_ Under the CH32 directory, as shown in the following figure:



✧ Description of sample program content

The sample program includes the following content:

A. Home screen display;

B. Single color screen brushing

C. Rectangle drawing display;

D. Circular drawing display;

E. Triangle drawing display;

F. English display;

G. Display of numbers and symbols

H. Chinese display;

I. BMP monochrome image display;

J. Menu simulation display;

✧ Example program display direction switching instructions

Found macro definition **USE_HORIZONTAL** and **COLOR_STATE** in **HARDWARE\OLED\oled.h** file, as shown in the following:

```
#define USE_HORIZONTAL      0 // 设置显示方向: 0-正常, 1-旋转180度
#define COLOR_STATE         0 // 设置显示模式: 0-正常显示, 1-反色显示
```

Modify USE_ HORIZONTAL and COLOR_ STATE macro according to the following definition:

```
#define  USE_ HORIZONTAL  0     //0 ° rotation  (Default value)
#define  USE_ HORIZONTAL  1     //180 ° rotation
#define  COLOR_ STATE  0 //Black background, monochrome display content(Default value)
#define  COLOR_ STATE  1 //Monochrome background, black display content
```

✧ Example program IIC slave device address modification instructions (only for IIC test programs)

First, locate the macro definition **IIC_SLAVE_ADDR** in the **HARDWARE\IIC\iic.h** file, as shown in the following figure:

```
//定义IIC从设备地址
#define IIC_SLAVE_ADDR 0x78  //0x7A
```

Modify IIC_ SLAVE_ ADDR macro definition according to the following definition is sufficient to:

#define IIC_ SLAVE_ ADDR 0x78 //Slave device address is 0x78 (default value)

#define IIC_ SLAVE_ ADDR 0x7A //Slave device address is 0x7A

Next, find **OLED_Init_GPIO** function in the **HARDWARE\OLED\oled. c** file. If using the 0x7A slave device address, there is no need to annotate the code **GPIO_SetBits(GPIOB,GPIO_Pin_5)** (to make them effective). If using the 0x78 slave device address, the code **GPIO_SetBits(GPIOB,GPIO_Pin_5)** need to be annotated (to make them ineffective), as shown in the following figure(Taking the CH32F103C8T6 example program as an example):

```
void OLED_Init_GPIO(void)
{
  GPIO_InitTypeDef  GPIO_InitStructure;
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);  //使能B端口时钟
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5|GPIO_Pin_8|GPIO_Pin_9; //GPIOB10,11,12
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;     //推挽输出
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;//速度50MHz
  GPIO_Init(GPIOB, &GPIO_InitStructure);     //初始化GPIOB10、11、12
  GPIO_SetBits(GPIOB,GPIO_Pin_8);
  GPIO_ResetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_9);
  //slave address is 0x7A, select the folllow define:
  //GPIO_SetBits(GPIOB,GPIO_Pin_5);
}
```

4. Demo Usage Instructions

✧ Installing development tool software

Firstly, you need to install the development tool software, which uses Keil5. Please refer to the online download and installation methods for yourself.

✧ Installing Device Library

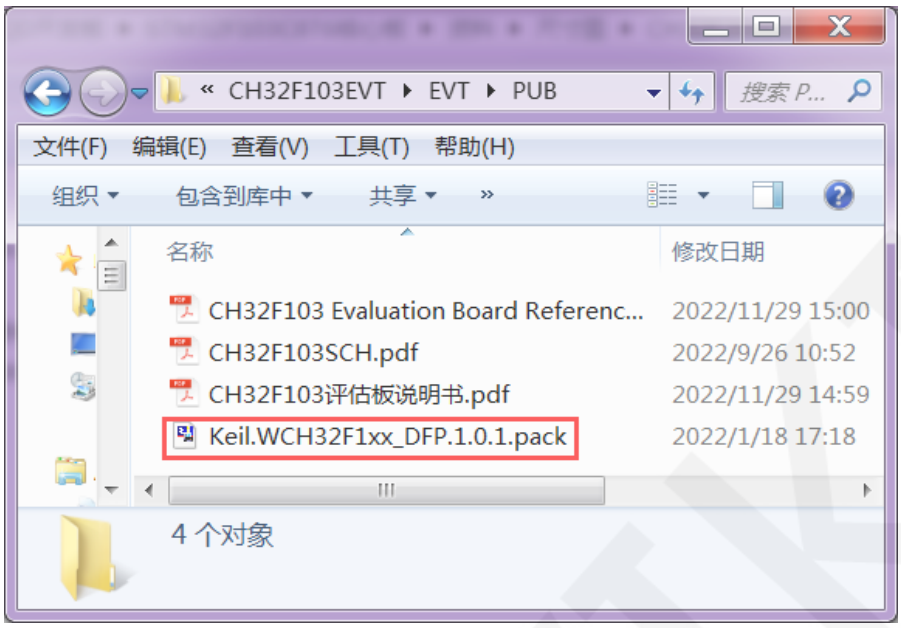
After installing the keil5 software, it is necessary to install the CH32 device library (omitted if already installed), and the download address is as follows:

CH32F103C8T6: https://www.wch.cn/downloads/CH32F103EVT_ZIP.html

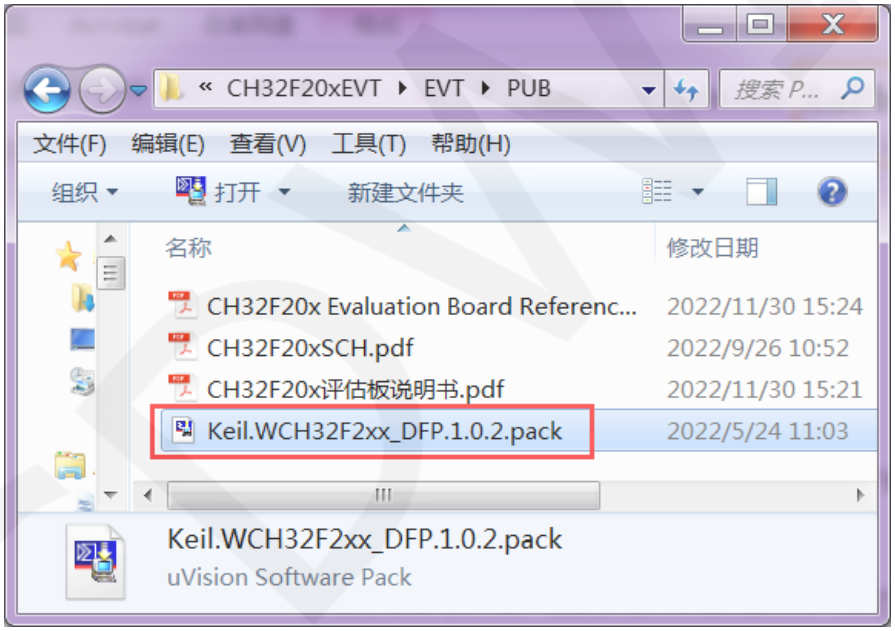
CH32F203C8T6: https://www.wch.cn/downloads/CH32F20xEVT_ZIP.html

After downloading the official information package, unzip it and find the pack file in the **EVT\PUB** directory, as shown in the following figure:

CH32F103C8T6's pack:



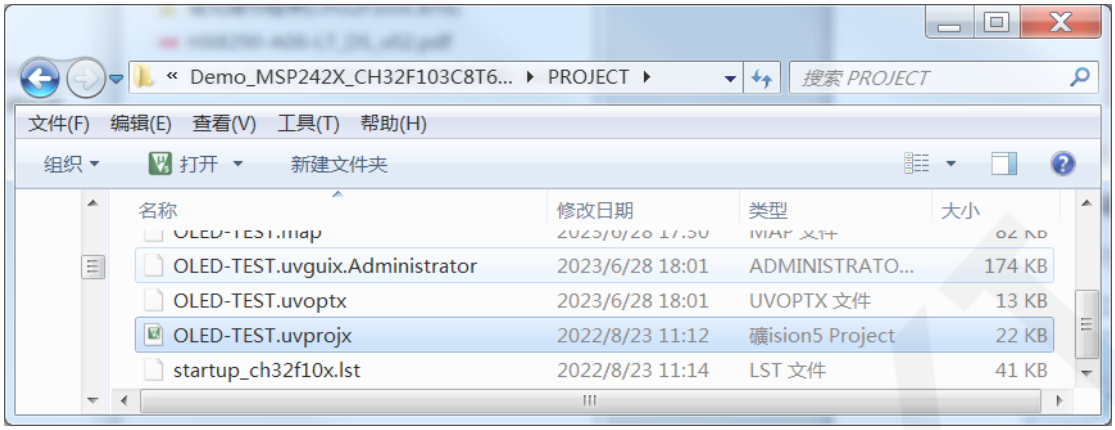
CH32F203C8T6's pack：



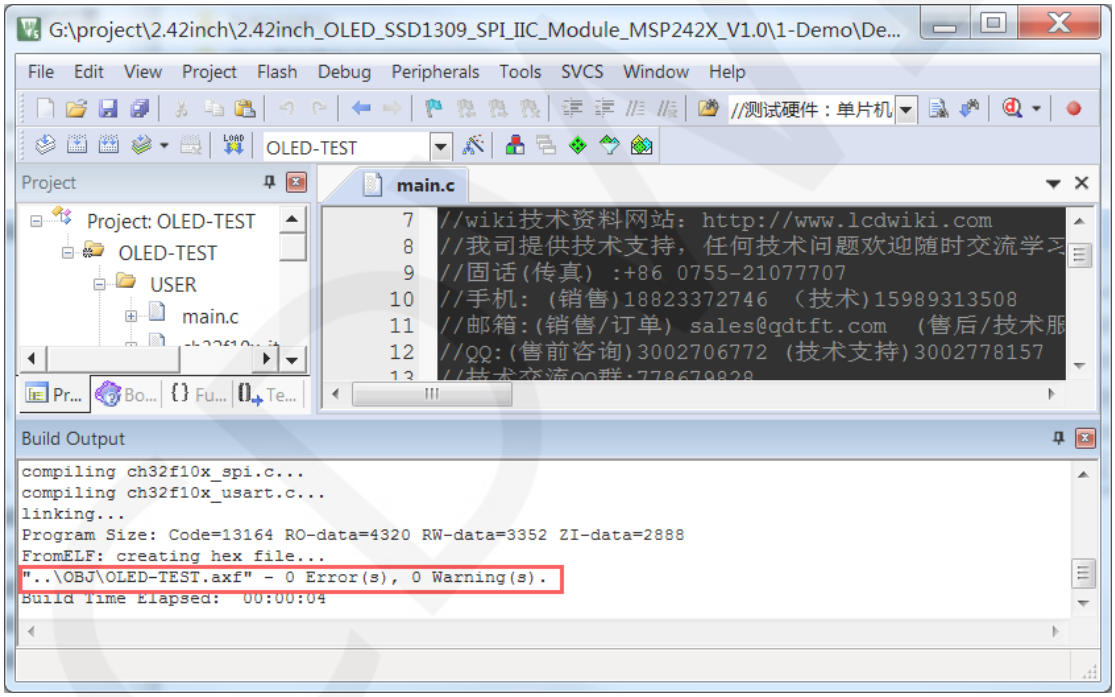
Double click on the pack file and follow the prompts to install.

✧ Compiling Programs

After the library installation is completed, open the **PROJECT** directory under the sample program, locate the **uvprojx** file, double-click to open the sample project, as shown in the following figure:



After opening the sample project, you can make modifications to the project code (or not). After the modifications are completed, click the compile button to compile the code. The following prompt appears, indicating successful compilation, as shown in the following figure:



✧ Download and Run Programs

The development board supports SWD download, USB download, and serial port download

When downloading SWD, ST Link or WCH Link downloaders can be used.

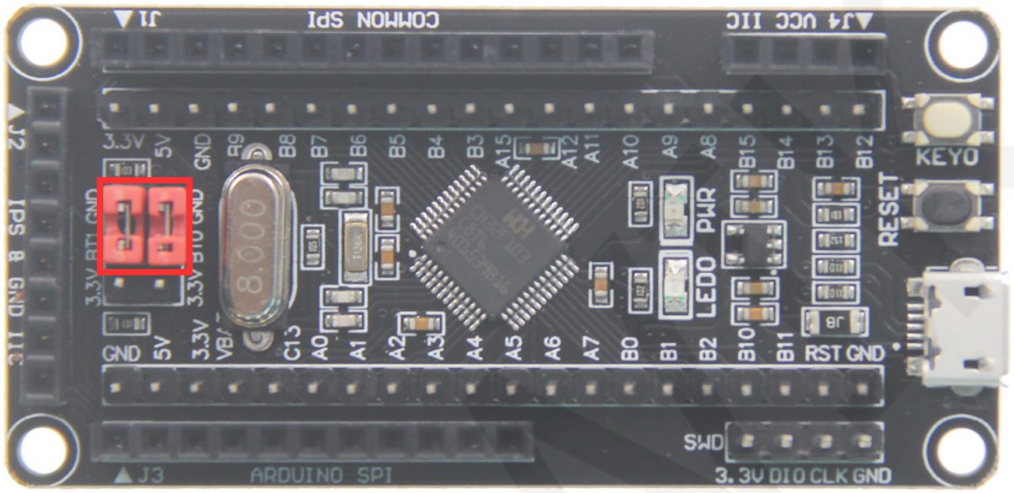
Here is an introduction to SWD download. For other download methods, please refer to the documentation in the development board documentation package or consult the

internet.

The steps for downloading SWD are as follows (using the CH32F103C8T6 development board as an example):

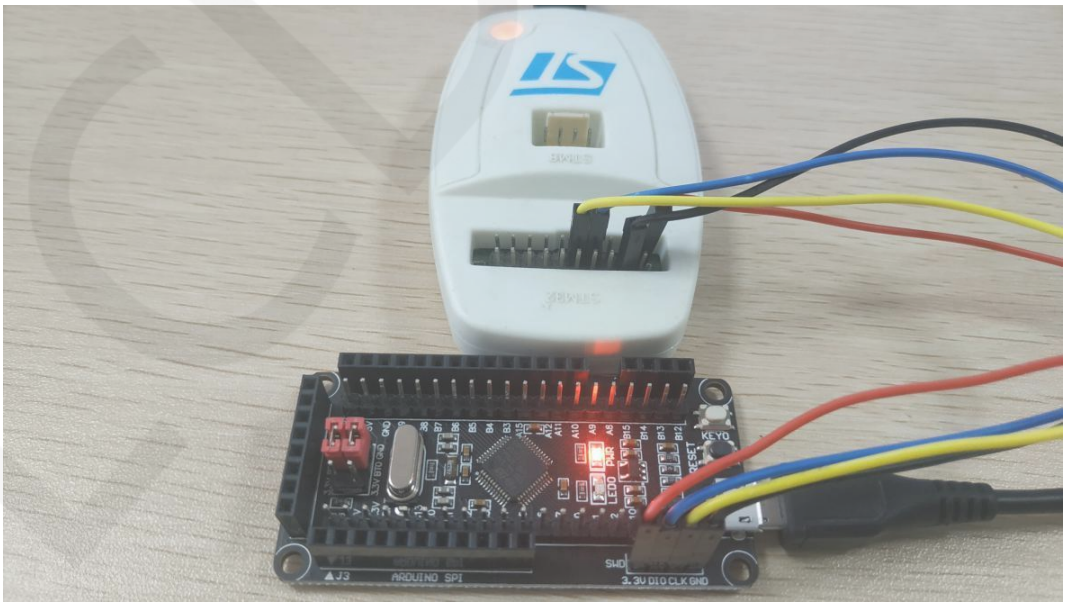
A. Firstly, ensure that the BT0 and BT1 pins of the MCU remain low, as shown in the following figure:

The BT0 and BT1 pins of CH32F103C8T6 are connected to GND using jump caps.

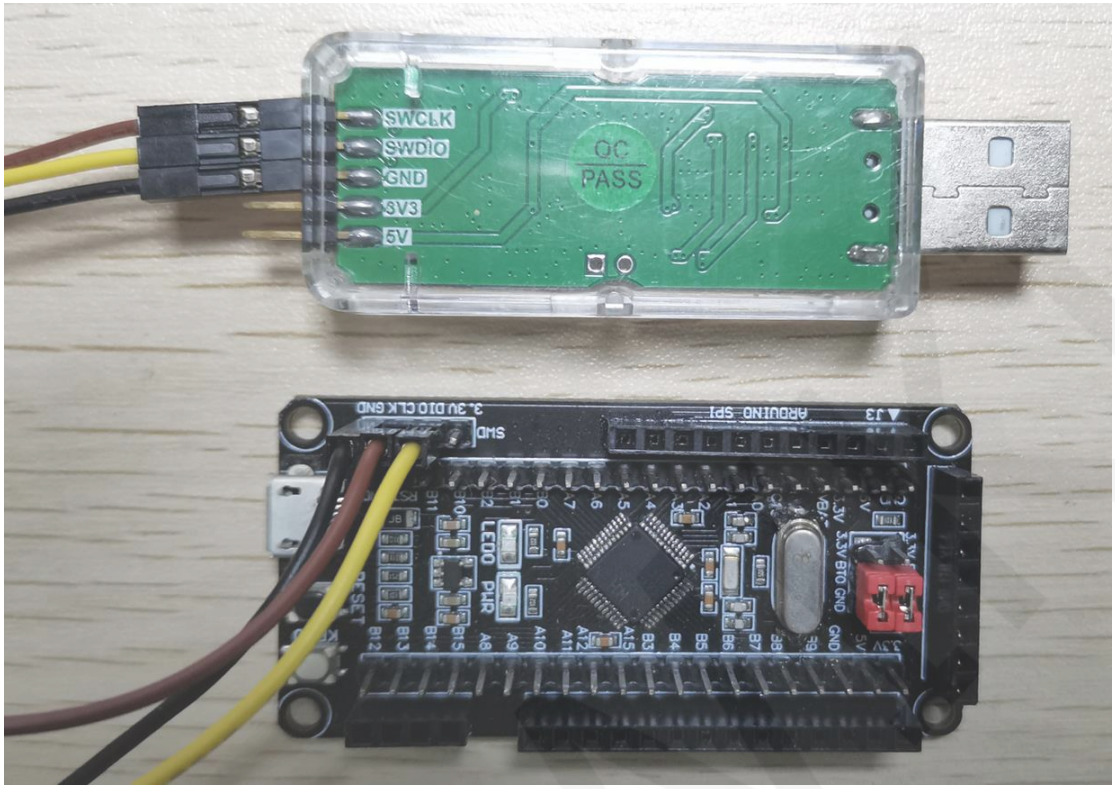


B. Find the **SWD** interface of the development board and connect it one by one with the pins of the emulator (theoretically, any emulator that supports the SWD protocol supports it), as shown in the following figure:

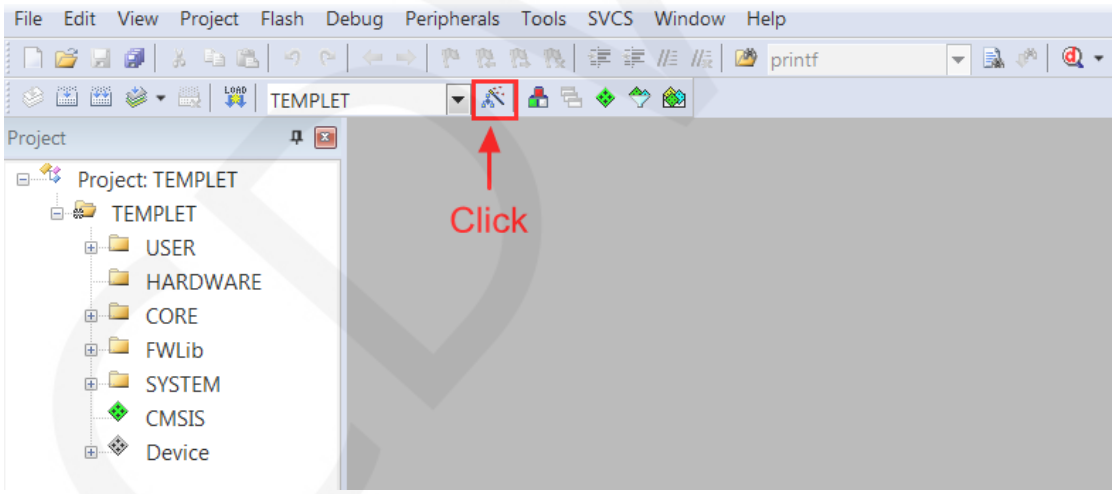
Connect to ST-Link:



Connect to WCH-Link:



C. Open the KEIL tool software and click the button shown in the following figure:



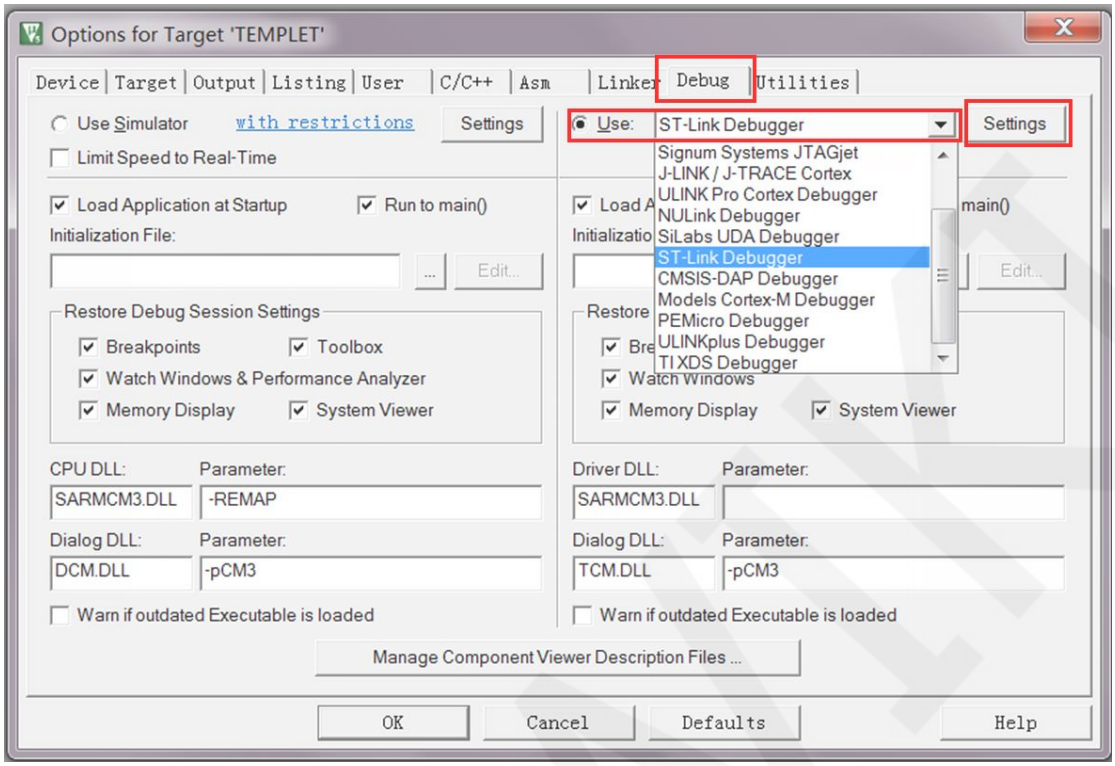
D. Click the **Debug** button in the pop-up interface, and then select the emulator in Use.

If using **ST-Link**, please select **ST-LINK Debugger**

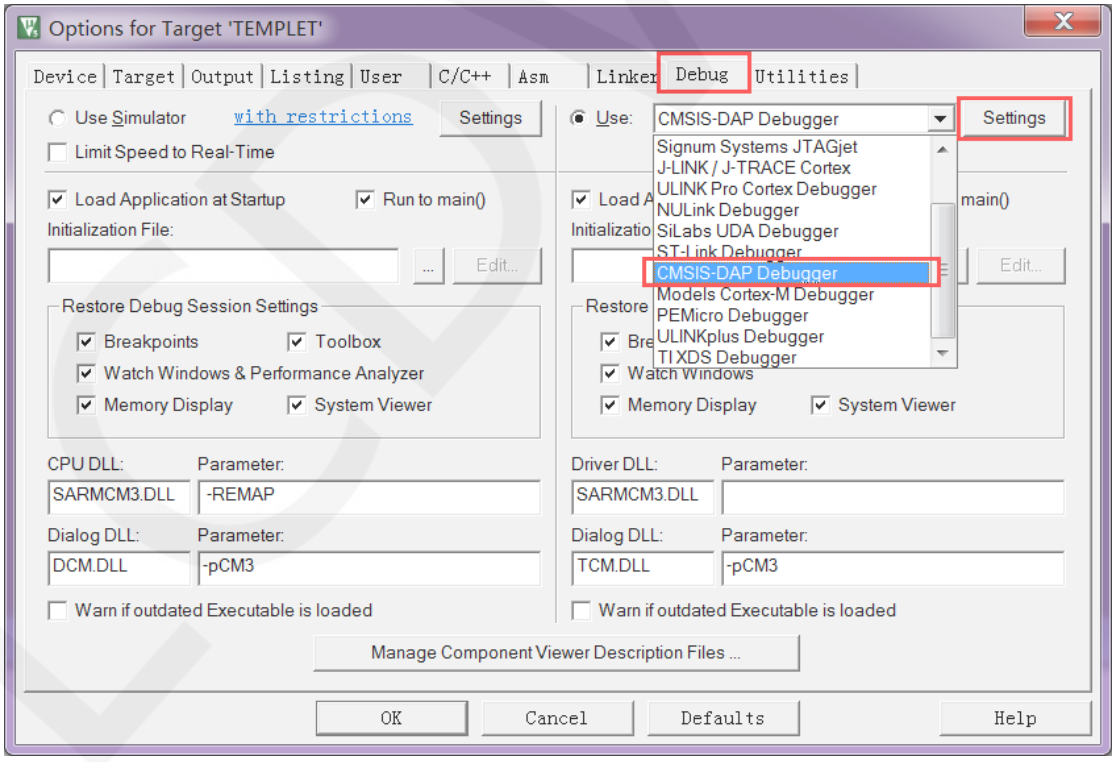
If using **WCH-Link**, please select **CMSIS-DAP Debugger**

As shown in the following figure:

Using ST-Link:

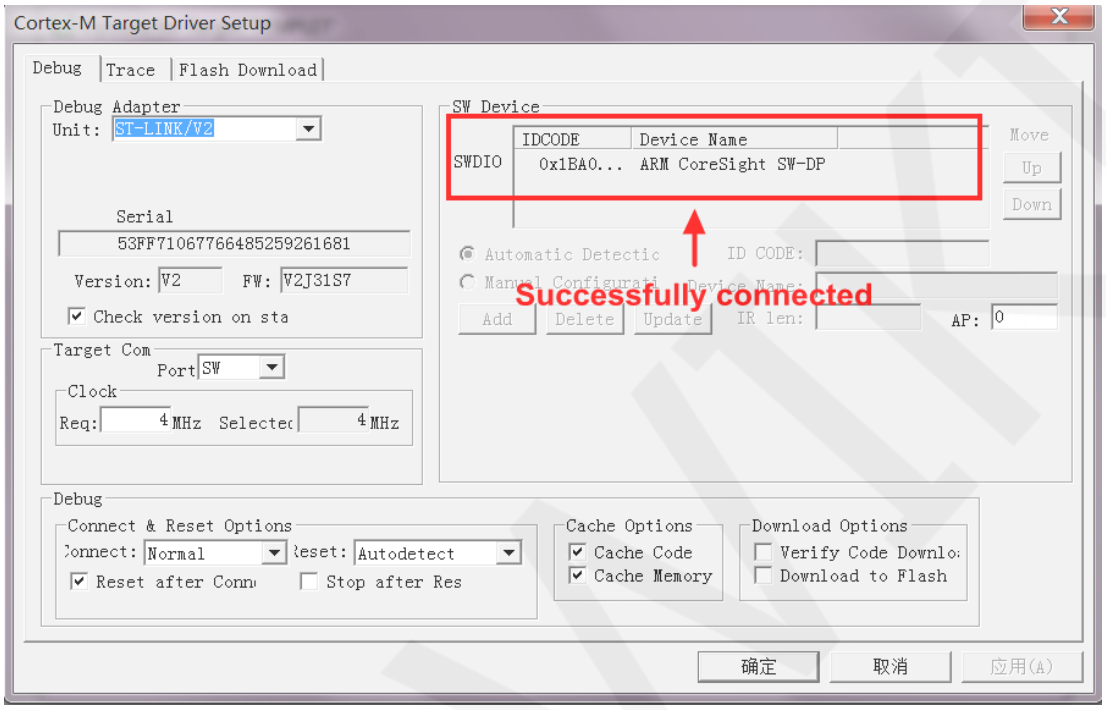


Using WCH-Link:

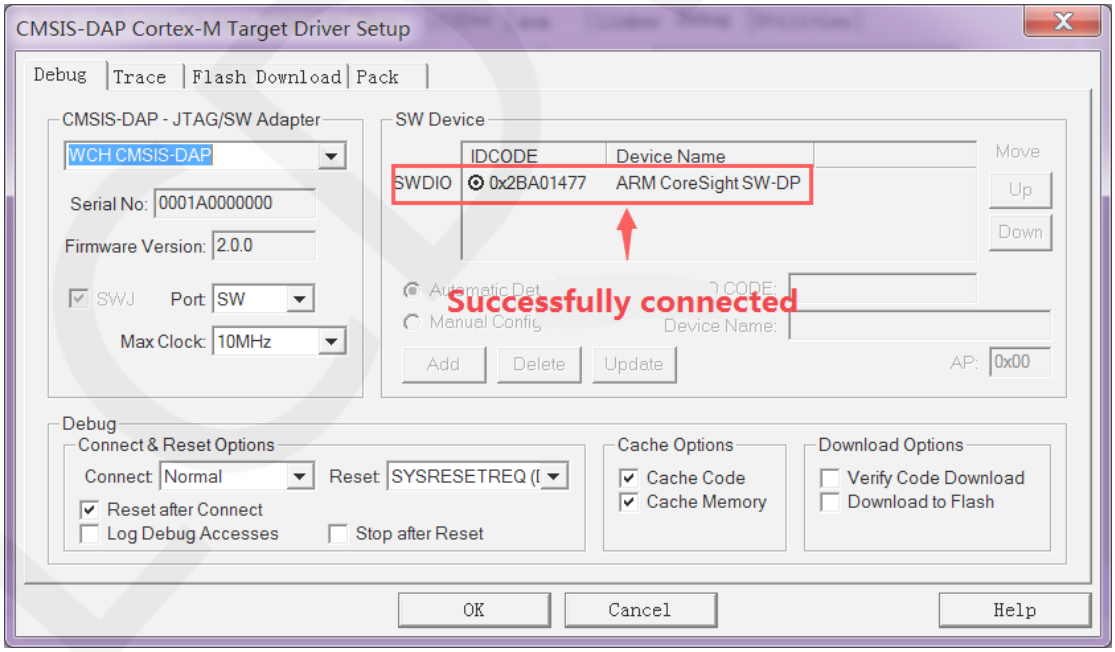


E. Power on the development board and click the **Settings** button next to Use (as shown in the previous operation). The following interface will pop up, indicating that the emulator has successfully connected:

ST-Link connection successful:

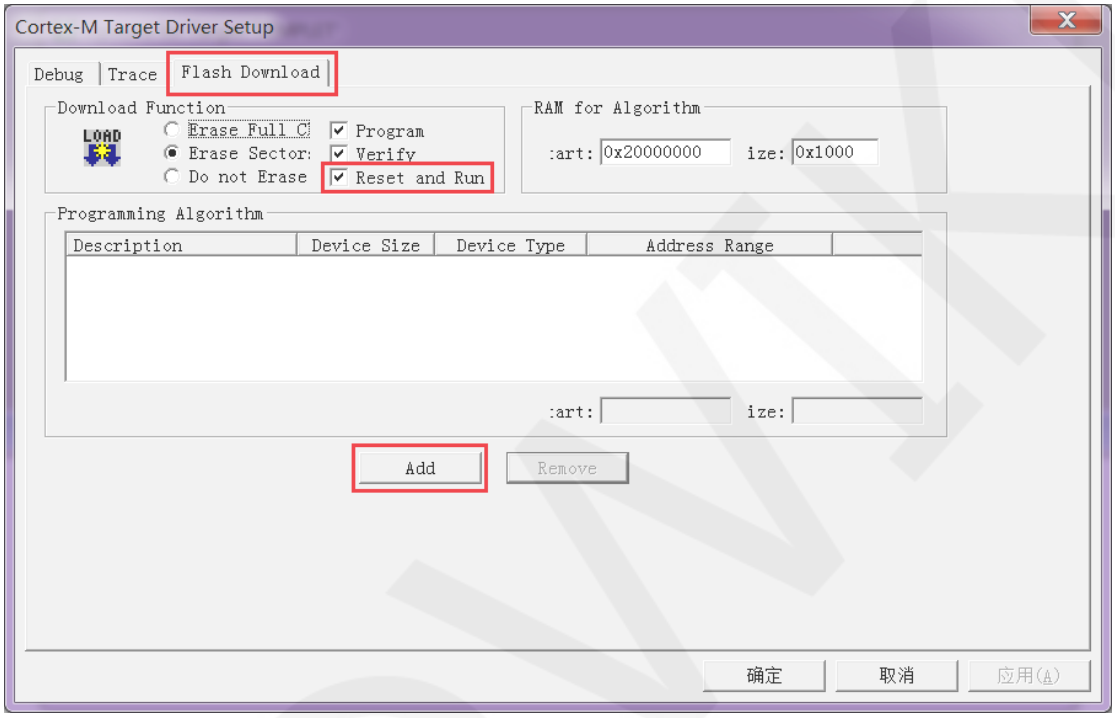


WCH-Link connection successful:

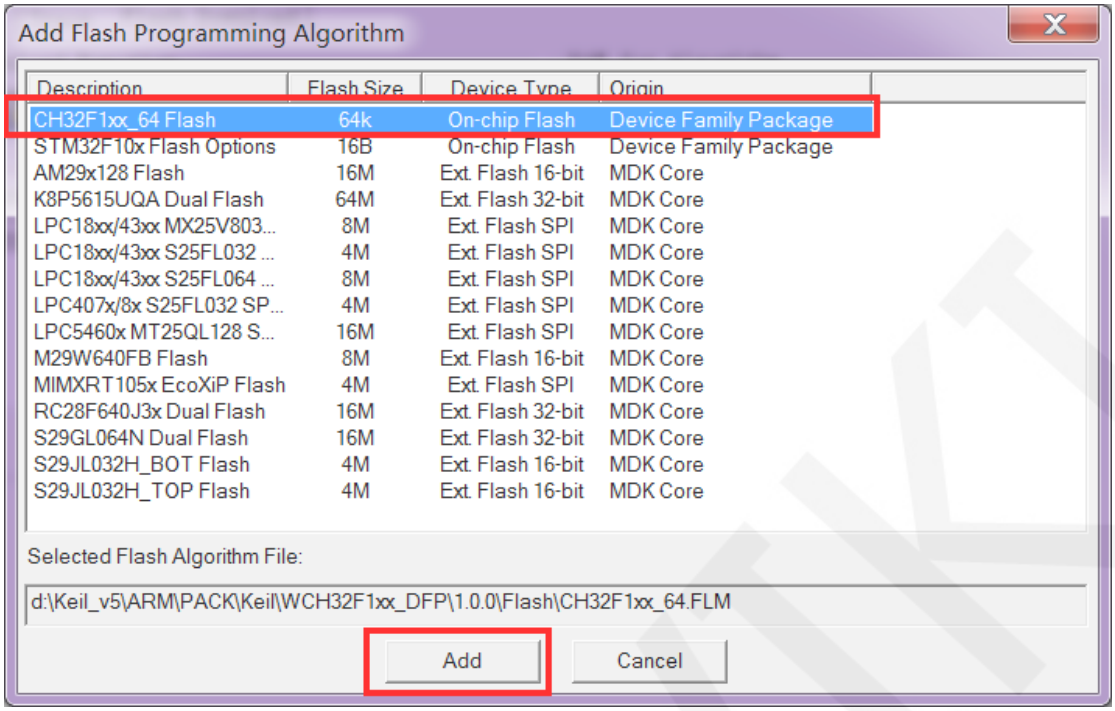


F. Click the **Flash Download** button to enter the flash settings interface, as shown in the following figure (if flash has been selected, this step can be omitted):

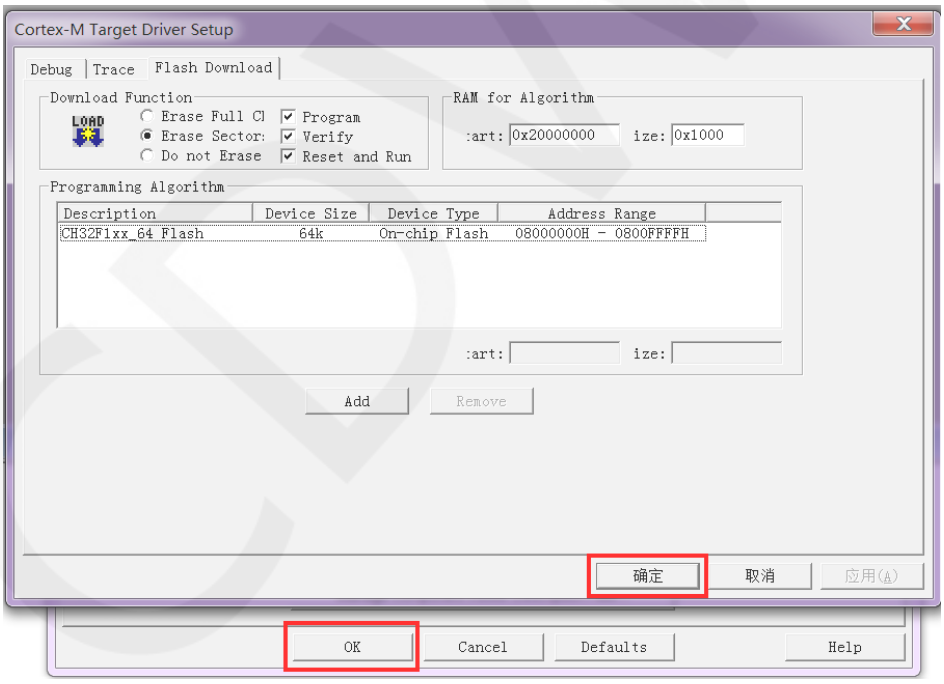
If you want the program to automatically run after successful download, you need to check **Reset and Run**. Otherwise, after successful download, you need to press the reset button or power off to restart before running the program.



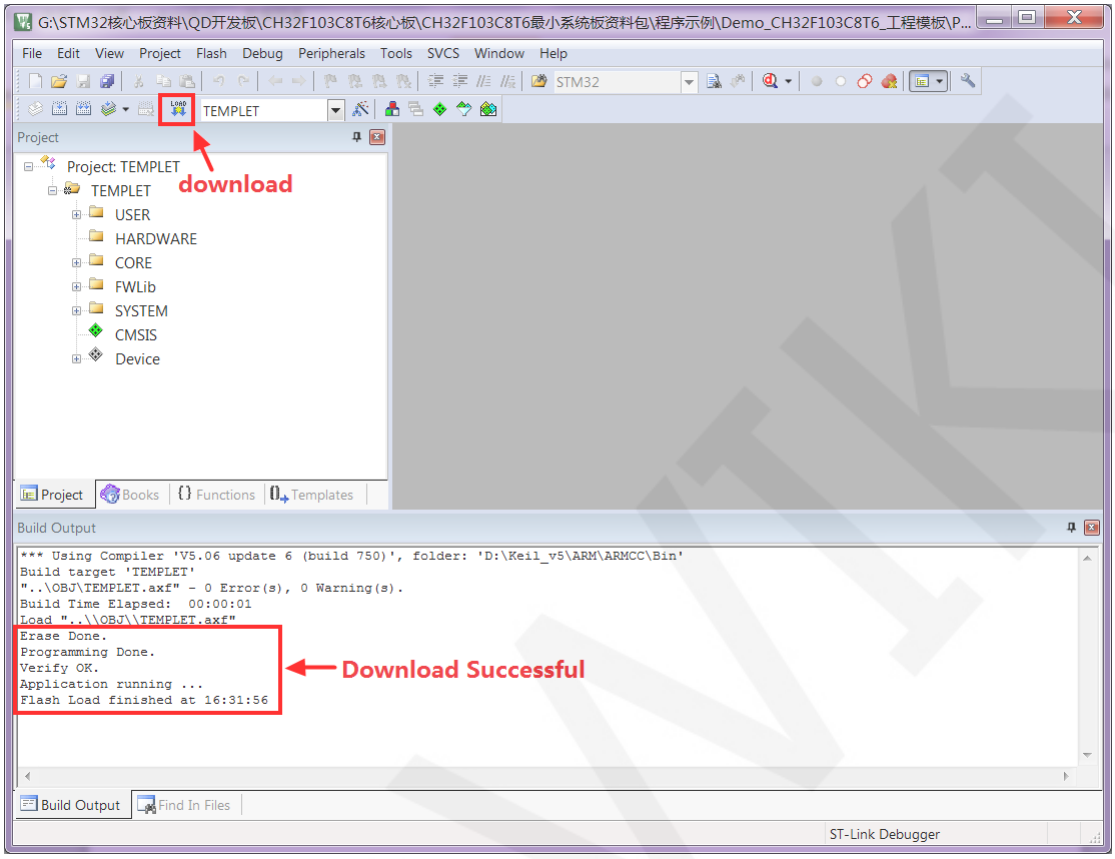
G. Click on the **Add** button (as shown in the above figure) to select flash. Generally, the first one is selected (the algorithm has already been processed), and once it is selected, click on the **Add** button below to exit, as shown below:



H. Click the **OK** button and the **OK** button to exit the settings interface, as shown in the following figure:



I. Click the **download** button to download the program, and the following prompt will appear, indicating successful download, as shown in the following figure:



J. If the display module displays characters and graphics normally, it indicates that the program has run successfully.