FTM8ForgeIDE

User Manual

Fortior Technology (Shenzhen) Co., Ltd.

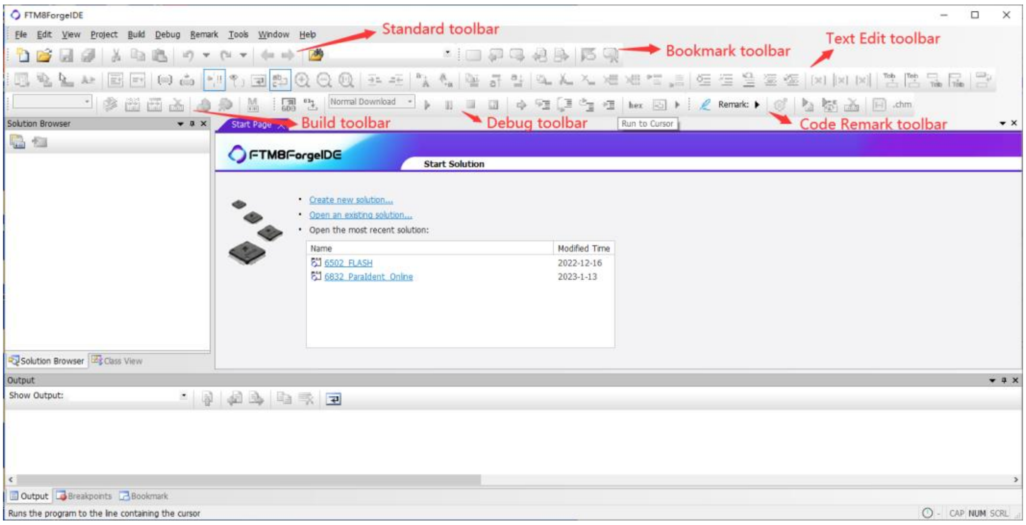
Contents

1 FTM8ForgeIDE Introduction

FTM8ForgeIDE, or an integrated development environment, is a computer program that includes tools for a variety of programming and software tasks. This chapter introduces the FTM8ForgeIDE tools and windows, which aims to help the programmers to develop embedded applications quickly.

Run the FTM8ForgeIDE as administrator. The default interface is shown as below:

Figure 1-1 FTM8ForgeIDE Startup Interface



- Create new solution...
- Open an existing solution...
- Open the most recent solution:

Name	Modified Time
6502_FLASH	2022-12-16
6832_Paraldent_Online	2023-1-13

Runs the program to the line containing the cursor

Figure 1-2 FTM8ForgeIDE Main Interface

1. Meau Bar: It integrates all tools and settings of the FTM8ForgeIDE.
2. Standard toolbar: It mainly has the following functions:
   - New, open, save and save as
   - Cut, copy, paste, undo and redo
   - Navigate backwards, navigate forwards, bookmark
3. Build toolbar: It includes compile, build, build all and stop build buttons.
4. Debug toolbar: It includes download code, go, break, stop debugging, restart, and step through.

5. Text Edit toolbar: It includes insert line comment, remove line comment, zoom in, zoom out and so on.
6. Bookmark toolbar: Use bookmarks for navigating between marked sections of the code.
7. Code Remark toolbar: View and generate help document for header files in the current project.
8. Solution Browser Window: Show all source and files in the current solutions.
9. Start Page: List the most recently used solutions.

1.1 Menu Bar

1.1.1 File Menu

The menu File includes commands for managing files.

File Menu	Icon	Shortcut	Description
New			Create a new project, file, and solution.
Open	📂	Ctrl+O	Open a file.
Close			Close the file.
Open Solution	📂	Ctrl+Shift+O	Open a project or solution.
Close Solution			Close a project or solution.
Save	💾	Ctrl+S	Save the file.
Save As			Save and rename the file.
Save All	💾	Ctrl+Shift+S	Save all open source and text files, including the project configuration settings.
Keil Project Convert			Convert a Keil project to FTM8ForgeIDE project. See 1.2 Keil Project Convert for details.
Recent Files			List all recently used source or text files.
Recent Solution			List all recently used solutions.
Exit			Exit FTM8ForgeIDE and prompt to save files.

1.1.2 Edit Menu

Edit Menu	Icon	Shortcut	Description
Undo	↶ ▾	Ctrl+Z	Cancel the last edit operation.
Redo	↷ ▾	Ctrl+Y	Restore the last undone operation.
Cut	✂	Ctrl+X	Cut the selected text to the clipboard.
Copy	📄	Ctrl+C	Copy the selected text to the clipboard.
Paste	📋	Ctrl+V	Paste the text from the clipboard.
Delete	✗	Delete	Delete the selected text from the file.

Edit Menu	Icon	Shortcut	Description
Navigate Backwards	←	Ctrl+ -	Move the cursor back to the position occupied before a 'find' or 'go to line' command was executed.
Navigate Forwards	→	Ctrl+Shift+-	Move the cursor to the position before a 'Navigate Backwards' command was executed.
Find	🔍	Ctrl+F	Search for text patterns in the file.
Find/replace in Files	📂	Ctrl+Shift+H	Searches for text patterns in several files. Or replace the specified text with another text in several files.
Go to line		Ctrl+G	Move the cursor to the specified line number in the current source file.
Replace	A→B	Ctrl+H	Replace the specified text with another text.
Format Code	≡	Ctrl+K Ctrl+D	Format source code in the file.
Format Selected Code	≡	Ctrl+K Ctrl+F	Format the selected source code.
Toggle Outlining	{…}	Ctrl+M Ctrl+L	Offer commands for source code outlining in the active file.
Toggle Outlining Current	{…}	Ctrl+M	Offer commands for code outlining on the current line.
View			See the following View Option table.
Indent			See the following Indent Option table.
Convert Case			See the following Convert Case Option table.
Line Operation			See the following Line Operation Option table.
Comment/Uncomment			See the following Comment / Uncomment Option table.
Space Character Operation			See the following Space Character Operation Option table.
Character Encoding			Include ANSI, UTF8, UFT8-BOM, UCS2 Big-Endian and UCS2 Little-Endian.
Breakpoint	●	Ctrl+Alt+B	Set the breakpoint.

View Option	Icon	Shortcut	Description
View Indent Guides		Ctrl+R, Ctrl+W	Show or hide the indent guides.
View Space			Show or hide.
Auto Wrap Line		Ctrl+E, Ctrl+W	.
View Newline Character			Show or hide line breaks.
Zoom In			Zoom in on the text.
Zoom Out			Zoom out on the text.
Zoom Default			Set the font to 1:1 to the window size.

Indent Option	Icon	Shortcut	Description
Increase Indent			Increase the line indentation of each selected line with one tab.
Decrease Indent			Decrease the line indentation of each selected line with one tab.

Convert Case Option	Icon	Shortcut	Description
To Uppercase		Ctrl+Shift+U	Convert the selected text to uppercase letters.
To Lowercase		Ctrl+U	Convert the selected text to lowercase letters.

Line Operation Option	Icon	Shortcut	Description
Duplicate Current Line		Ctrl+D	Copy the current line and paste the next line.
Move Up Selected Lines		Ctrl+Alt+D	Move the selected line up.
Move Down Selected Lines		Alt+Down	Move the selected line down.
Copy Selected (Current) Line		Ctrl+Shift+C	Copy the selected line to the clipboard.
Cut Selected (Current) Line		Ctrl+Shift+X	Cut the selected line to the clipboard.
Remove Selected (Current) Line		Ctrl+Shift+Delete	Delete the text of the current line.
Remove All Blank Line			Delete all blank lines from the file.
Remove All Blank Line(With Space			Replace blank lines with spaces in the file.

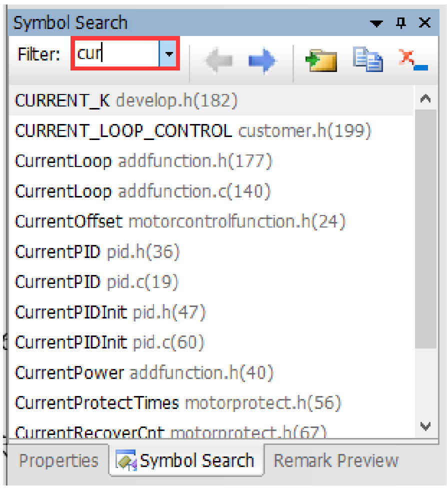
Line Operation Option	Icon	Shortcut	Description
Characters)			
Insert Blank Line Above			Insert a blank line above the line.
Insert Blank Line Below			Insert a blank line below the line.

Comment/Uncomment Option	Icon	Shortcut	Description
Add/Remove Line Comment		Ctrl+K Ctrl+/	Convert the selected lines to comments. Or convert commented lines to code text.
Add Line Comment			Convert the selected lines to comments.
Remove Line Comment			Convert commented lines to code text.
Add Block Comment		Ctrl+?	Convert the selected code block to comments.
Remove Block Comment			Convert commented code block to code text.

Space Character Operation Option	Icon	Description
Remove Line-Head and Tail Space	(×)	Delete each leading tab, leading space, trailing tab or trailing space in the selected text.
Remove Line-Head Space	|(×)	Delete each leading tab or leading space in the selected text.
Remove Line-Tail Space	(×)|	Delete each trailing tab or trailing space in the selected text.
EOL to Space		Converts the newline characters into spaces.
Tab to Space(All)	Tab	Replace line-head tabs with spaces in the file.
Tab to Space(Line-Head)	Tab	Replace line-head tabs with spaces in the selected text.
Space to Tab(All)	Tab	Replace line-head spaces with tabs in the file.
Space to Tab(Line-Head)	Tab	Replace line-head spaces with tabs in the selected text.

1.1.3 View Menu

View Menu	Icon	Shortcut	Description
Solution Browser		Ctrl+Alt+L	Show the Solution Browser window. See 1.4.1 Solution Browser for details.
Class View		Ctrl+Shift+C	Show the Class View window. See 1.4.2 Class View for details.
Resource View		Ctrl+ Shift +E	Show the Resource View window.
Properties View		Alt+Enter	Show the Properties View window. See

View Menu	Icon	Shortcut	Description
			1.4.3 Properties View for details.
Bookmark Window		Ctrl+K Ctrl+W	Show the Bookmark window. You can use bookmarks for navigating between marked sections of the code. Bookmarks are displayed as magenta squares in the editor margin and are set through the toolbar buttons.
Command Window		Ctrl+Alt+A	Show the Command window.
Symbol Window		Ctrl+Alt+S	Show the Symbol Search window. Enter what you want to search. The window displays the search result and the location in the file. Click ⇦ ⇨ to forward or backward, click to open the directory where the file is located, click to copy file name to clipboard, click to clear the list.  Symbol Search Filter: cur CURRENT_K develop.h(182) CURRENT_LOOP_CONTROL customer.h(199) CurrentLoop addfunction.h(177) CurrentLoop addfunction.c(140) CurrentOffset motorcontrolfunction.h(24) CurrentPID pid.h(36) CurrentPID pid.c(19) CurrentPIDInit pid.h(47) CurrentPIDInit pid.c(60) CurrentPower addfunction.h(40) CurrentProtectTimes motorprotect.h(56) CurrentRecoverCnt motorprotect.h(67) Properties Symbol Search Remark Preview Figure 1-3
Remark Preview	CHM	Remark Preview	Show the Remark window.
Find Result 1/ Find Result 2			Show the Result 1 / Result 2 window.
Output		Alt+2	Show the Output window.

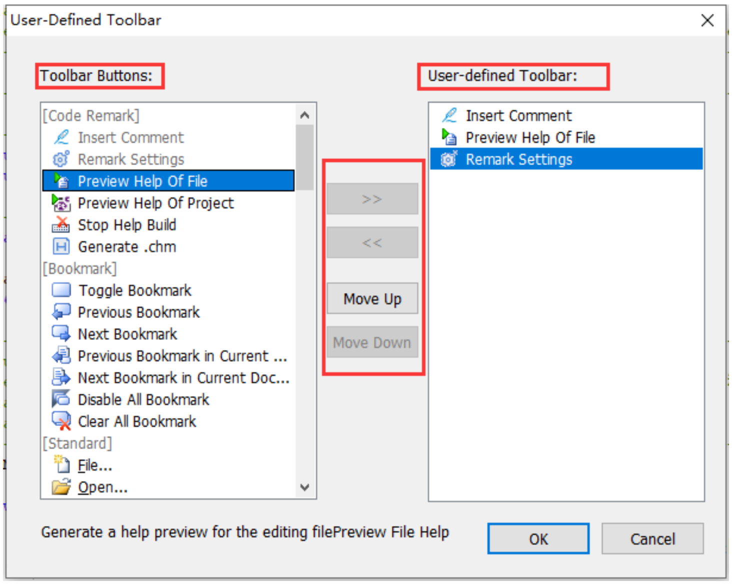
View Menu	Icon	Shortcut	Description
Breakpoints	●	Ctrl+B	Show the Breakpoints window. See 1.4.4 Breakpoints Window for details.
Debug Windows			Show the Debug windows.
Toolbars			Show or hide the toolbars.
Customize User-defined Toolbar…			See section Customize User-defined Toolbar for details.
Full Screen	▤	Shift + Alt +Enter	Maximize the window to full screen.

Debug Window Option	Shortcut	Description
Watch		Show the Watch window. See 1.4.8 Watch Window for details.
Auto Variables	Ctrl+Alt+V，A	Show the Auto Variables window.
Local Variables	Alt+4	Show the Local Variables window.
Memory		Show the Memory window. See 1.4.7 Memory Window for details.
Disassembly	Alt+8	Show the Disassembly window. See 1.4.5 Disassembler Window for details.
Register	Alt+5	Show the Register window. See 1.4.6 Registers Window for details.

1.1.3.1 Customize User-defined Toolbar

Click “View” → “Customize User-defined Toolbar…” to show the User-Defined Toolbar window. As shown in Figure 1-4, the Toolbar Buttons list displays all tools that you can use. The User-defined Toolbar list displays the tools that you selected.

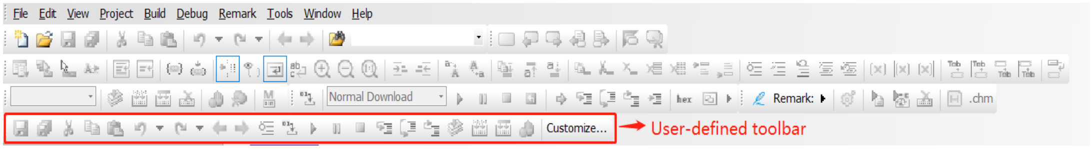
Click >> to add the selected tool to User-defined Toolbar list, click << to remove the selected tool from User-defined Toolbar list, click Move Up to move the selected tool up and click Move Down to move the selected tool down.



Generate a help preview for the editing filePreview File Help

Figure 1-4

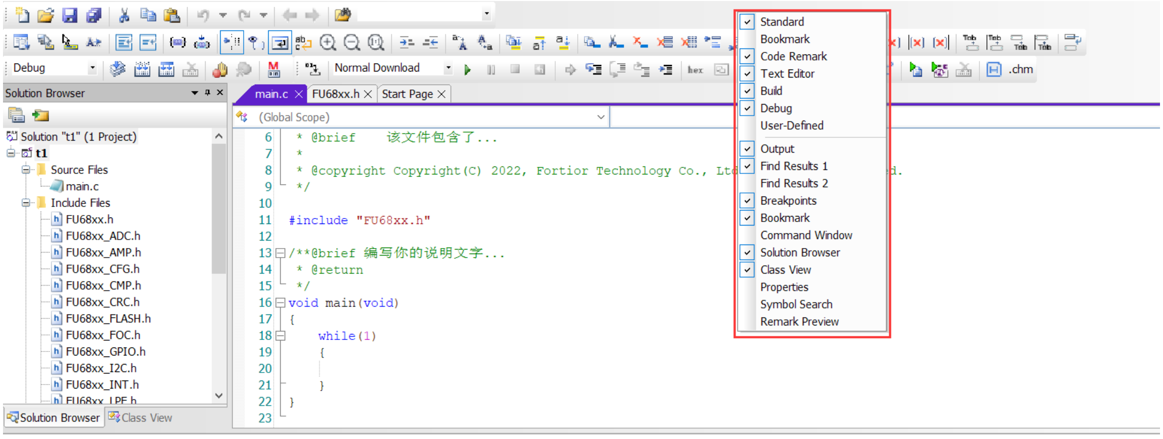
As shown in the following figure, you can move the User-Defined Toolbar as you want.



User-defined toolbar

Figure 1-5

You can right click the mouse at any position of the toolbar to select to show or hide some toolbars, as shown in Figure 1-6. If the toolbar is selected, it is displayed. If not, it is hidden. You can select them according to your needs.



```
 * @brief    该文件包含了...
 *
 * @copyright Copyright(C) 2022, Fortior Technology Co., Ltd.
 */

#include "FU68xx.h"

/**@brief 编写你的说明文字...
 * @return
 */
void main(void)
{
    while(1)
    {
    }
}
```

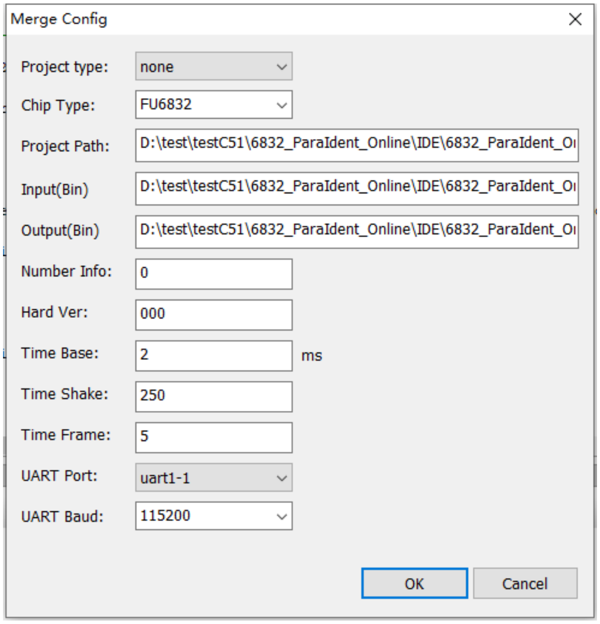
Figure 1-6

1.1.4 Project Menu

Project Menu	Description
Add to Project	Add a new file, folder or some existing files to the project.
Dependencies…	Show the Project Dependencies window.
Build Sequence…	Show the Compilation Generation Order window.
Settings…	Show the Project Settings window. See 1.3 Setting for details.
Exports Makefile…	Show the Exports Makefile(s) window.
Change Body	Show the Select Body window to change the chip type.
Insert Project into Solution…	Add an existing project to the solution.

1.1.5 Build Menu

Build Menu	Icon	Shortcut	Description
Compile		Ctrl+F7	Compile the file. Syntax check is performed during compilation and the compilation information is displayed in the Output window.
Build		F7	Build the target project, compile the modified file, and generate hex file.
Rebuild All		Ctrl+Alt+F7	Rebuild the target project, compile all sources files, and generate hex file.
Batch Build			Execute build-commands on the selected project.
Set Active Configuration			Show the Set Active Project Configuration window. There are two configurations: Debug and Release. You can debug program built under debug version. Program built under release version is more convenient for programmers without debugging function.
Merge Bin	M		Shows the Merge Config window. Add the bootloader bin file and the user-developed bin file. Set the device information. And then click “OK” to merge.

Build Menu	Icon	Shortcut	Description
			 Merge Config Project type: none Chip Type: FU6832 Project Path: D:\test\testC51\6832_ParaIdent_Online\IDE\6832_ParaIdent_O Input(Bin) D:\test\testC51\6832_ParaIdent_Online\IDE\6832_ParaIdent_O Output(Bin) D:\test\testC51\6832_ParaIdent_Online\IDE\6832_ParaIdent_O Number Info: 0 Hard Ver: 000 Time Base: 2 ms Time Shake: 250 Time Frame: 5 UART Port: uart1-1 UART Baud: 115200 OK Cancel Figure 1-7

Merge Bin Option	Description
Project type	There are two types: keil and none. The default is keil.
Chip type	It includes FU68XX, FU5821, FU6511, FU6512, FU6521, FU6522, FU6812, FU6812x2, FU6813, FU6815, FU6816, FU6825, FU6832, FU6866, FU6881, FUEBK1.
Project Path	The project path.
Input (Bin)	The input bin file name. If the project type is none, you shall enter the file path.
Output (Bin)	The output bin file name.
Number Info	Devices number information. The default is 0.
Hard Ver	Hardware information. The default is 000.
Time Base	The default is 2ms.
Time Shake	The default is 250.
Time Frame	The default is 5.
UART Port	The default is uart1-1.
UART Baud	The default is 115200.

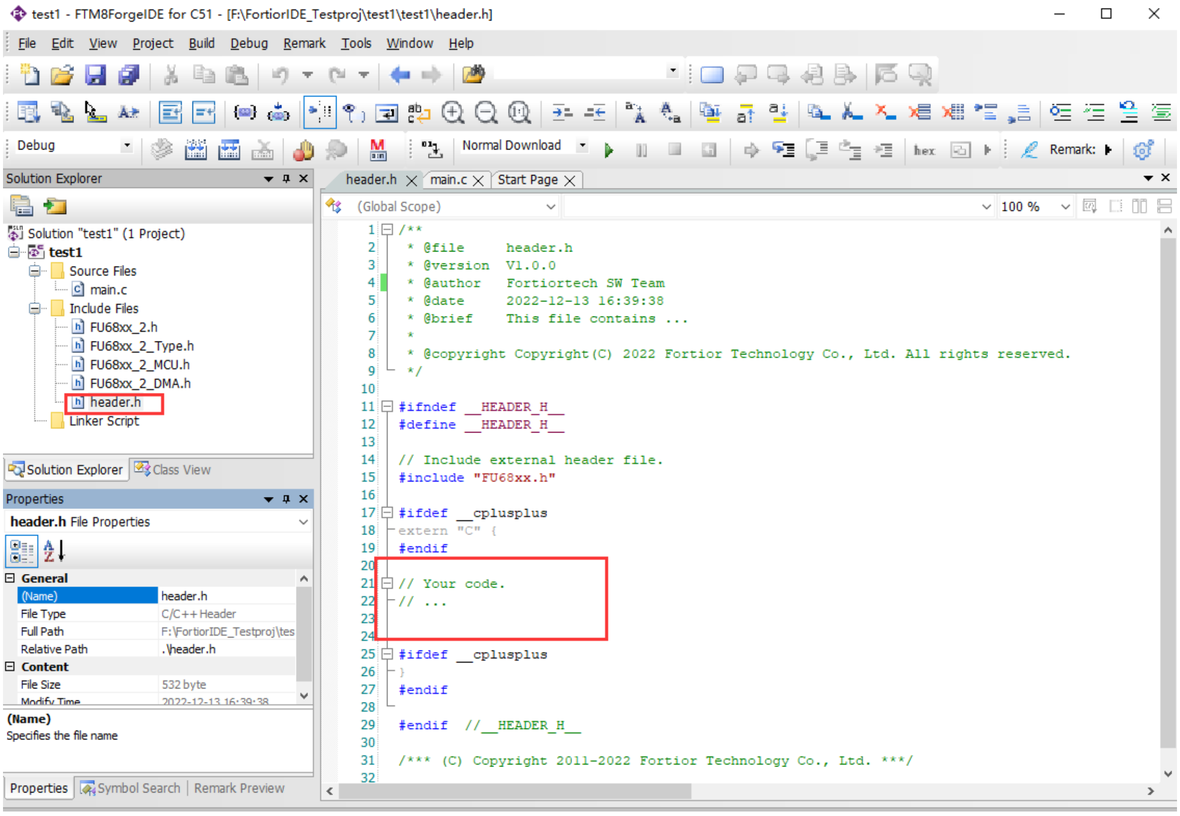
1.1.6 Debug Menu

Debug Menu	Icon	Shortcut	Description
Download Code	01	F8	Download the hex file to the Flash ROM device.

Debug Menu	Icon	Shortcut	Description
Download Mode			Program code download modes: Smart Download, Normal Download, Without Download
Go	▶	F5	Start a debugging session.
Break	⏸	Ctrl+Alt +Break	Pause a debugging session.
Stop Debugging	■	Shift +F5	Stop a debugging session.
Restart	↺	Ctrl+ Shift +F5	Restart a debugging session.
Step Into	↘	F11	Execute a single-step debugging into a function. It run the its internal code step-by-step.
Step Over	↷	F10	Execute a single-step debugging over a function. It executes all called functions without stepping into them.
Step Out	↖	Shift+F11	Step out of the function.
Run to Cursor	→	Ctrl+F	Execute debugging to the current cursor position.
Breakpoints	▣	Ctrl+B	Open the Breakpoints Window. See section 1.4.4 Breakpoints Window for details.
Toggle Breakpoint	●	F9	Set the breakpoint on the current line.
Remove All Breakpoints	●		Remove all breakpoints in the program.
Disable All Breakpoints	●		Disable all breakpoints in the program.

1.1.7 Remark Menu

Remark: Create declaration samples for functions, data interfaces, etc. in header files. A newly created header.h file is shown in Figure 1-8. See Create A New File for creating a file.



General

(Name)	header.h
File Type	C/C++ Header
Full Path	F:\FortiorIDE_Testproj\tes
Relative Path	.\header.h

Content

File Size	532 byte
Modify Time	2022-12-13 16:39:38

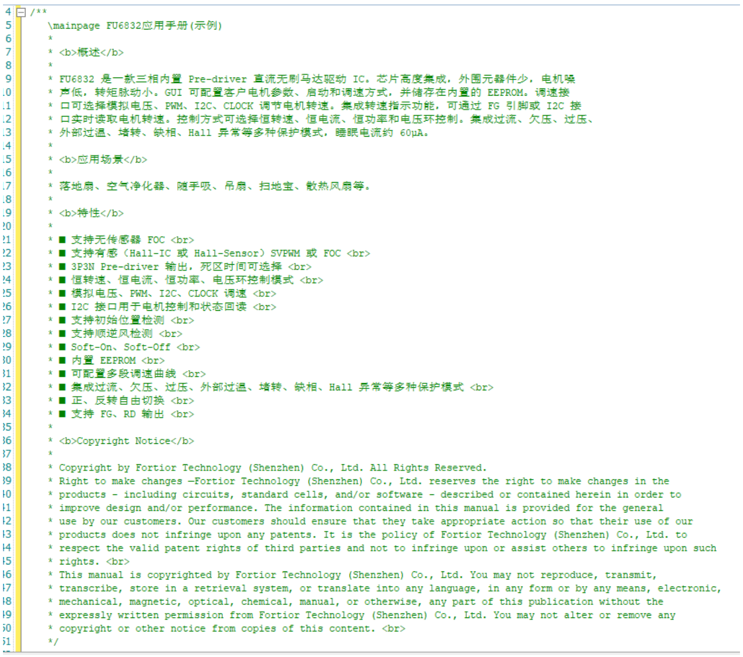
(Name)
Specifies the file name

```
/**
  * @file      header.h
  * @version   V1.0.0
  * @author    Fortiortech SW Team
  * @date      2022-12-13 16:39:38
  * @brief     This file contains ...
  *
  * @copyright Copyright(C) 2022 Fortior Technology Co., Ltd. All rights reserved.
  */

#ifndef __HEADER_H__
#define __HEADER_H__

// Include external header file.
#include "FU68xx.h"

#ifdef __cplusplus
extern "C" {
#endif

// Your code.
// ...


#ifdef __cplusplus
}
#endif

#endif  //__HEADER_H__

/*** (C) Copyright 2011-2022 Fortior Technology Co., Ltd. ***/
```

Figure 1-8

- Demo:

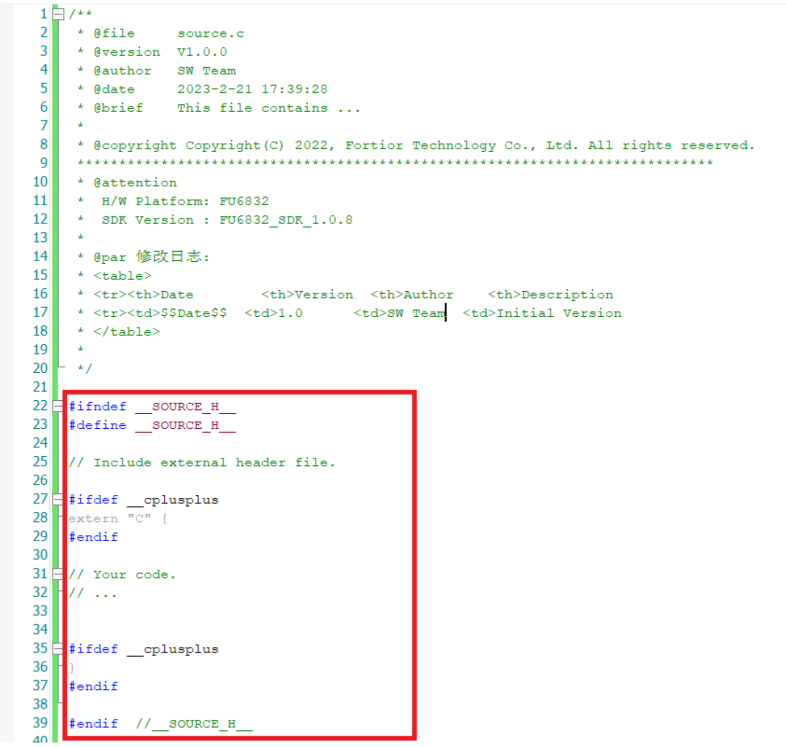
  - Project: Click “Remark”→ “Demo” → “Project” to add a project declaration sample in the header.h file, as the Figure 1-9.



```
/**
    \mainpage FU6832应用手册(示例)
    *
    * <b>概述</b>
    *
    * FU6832 是一款三相内置 Pre-driver 直流无刷马达驱动 IC。芯片高度集成，外围元器件少，电机噪
    * 声低，转矩脉动小。GUI 可配置客户电机参数、启动和调速方式，并储存在内置的 EEPROM。调速接
    * 口可选择模拟电压、PWM、I2C、CLOCK 调节电机转速。集成转速指示功能，可通过 FG 引脚或 I2C 接
    * 口实时读取电机转速。控制方式可选择恒转速、恒电流、恒功率和电压环控制。集成过流、欠压、过压、
    * 外部过温、堵转、缺相、Hall 异常等多种保护模式，睡眠电流约 60μA。
    *
    * <b>应用场景</b>
    *
    * 落地扇、空气净化器、随手吸、吊扇、扫地宝、散热风扇等。
    *
    * <b>特性</b>
    *
    * ■ 支持无传感器 FOC <br>
    * ■ 支持有感（Hall-IC 或 Hall-Sensor）SVPWM 或 FOC <br>
    * ■ 3P3N Pre-driver 输出，死区时间可选择 <br>
    * ■ 恒转速、恒电流、恒功率、电压环控制模式 <br>
    * ■ 模拟电压、PWM、I2C、CLOCK 调速 <br>
    * ■ I2C 接口用于电机控制和状态回读 <br>
    * ■ 支持初始位置检测 <br>
    * ■ 支持顺逆风检测 <br>
    * ■ Soft-On、Soft-Off <br>
    * ■ 内置 EEPROM <br>
    * ■ 可配置多段调速曲线 <br>
    * ■ 集成过流、欠压、过压、外部过温、堵转、缺相、Hall 异常等多种保护模式 <br>
    * ■ 正、反转自由切换 <br>
    * ■ 支持 FG、RD 输出 <br>
    *
    * <b>Copyright Notice</b>
    *
    * Copyright by Fortior Technology (Shenzhen) Co., Ltd. All Rights Reserved.
    * Right to make changes –Fortior Technology (Shenzhen) Co., Ltd. reserves the right to make changes in the
    * products - including circuits, standard cells, and/or software - described or contained herein in order to
    * improve design and/or performance. The information contained in this manual is provided for the general
    * use by our customers. Our customers should ensure that they take appropriate action so that their use of our
    * products does not infringe upon any patents. It is the policy of Fortior Technology (Shenzhen) Co., Ltd. to
    * respect the valid patent rights of third parties and not to infringe upon or assist others to infringe upon such
    * rights. <br>
    * This manual is copyrighted by Fortior Technology (Shenzhen) Co., Ltd. You may not reproduce, transmit,
    * transcribe, store in a retrieval system, or translate into any language, in any form or by any means, electronic,
    * mechanical, magnetic, optical, chemical, manual, or otherwise, any part of this publication without the
    * expressly written permission from Fortior Technology (Shenzhen) Co., Ltd. You may not alter or remove any
    * copyright or other notice from copies of this content. <br>
    */
```

Figure 1-9

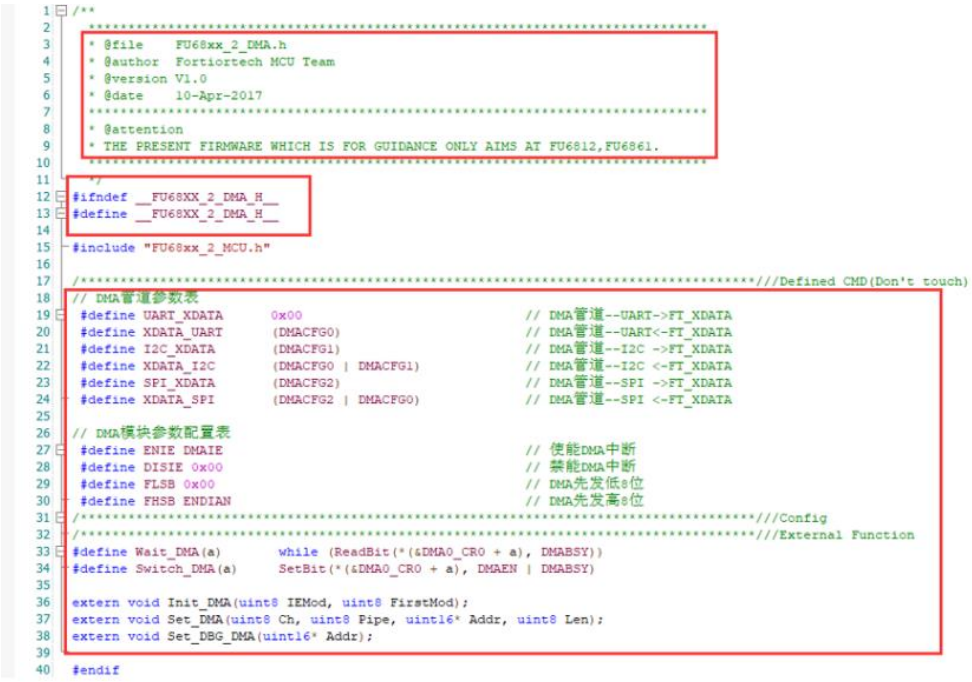
  - File: Click “Remark” → “Demo” → “File” to add a file declaration sample in the header.h file, such as function declaration, variable declaration, structure declaration, etc.



```
/**
 * @file     source.c
 * @version  V1.0.0
 * @author   SW Team
 * @date     2023-2-21 17:39:28
 * @brief    This file contains ...
 *
 * @copyright Copyright(C) 2022, Fortior Technology Co., Ltd. All rights reserved.
 *****************************************************************************
 * @attention
 *  H/W Platform: FU6832
 *  SDK Version : FU6832_SDK_1.0.8
 *
 * @par 修改日志:
 * <table>
 * <tr><th>Date        <th>Version  <th>Author    <th>Description
 * <tr><td>$$Date$$  <td>1.0      <td>SW Team  <td>Initial Version
 * </table>
 *
 */

#ifndef __SOURCE_H__
#define __SOURCE_H__

// Include external header file.

#ifdef __cplusplus
extern "C" {
#endif

// Your code.
// ...


#ifdef __cplusplus
}
#endif

#endif  //__SOURCE_H__
```

Figure 1-10

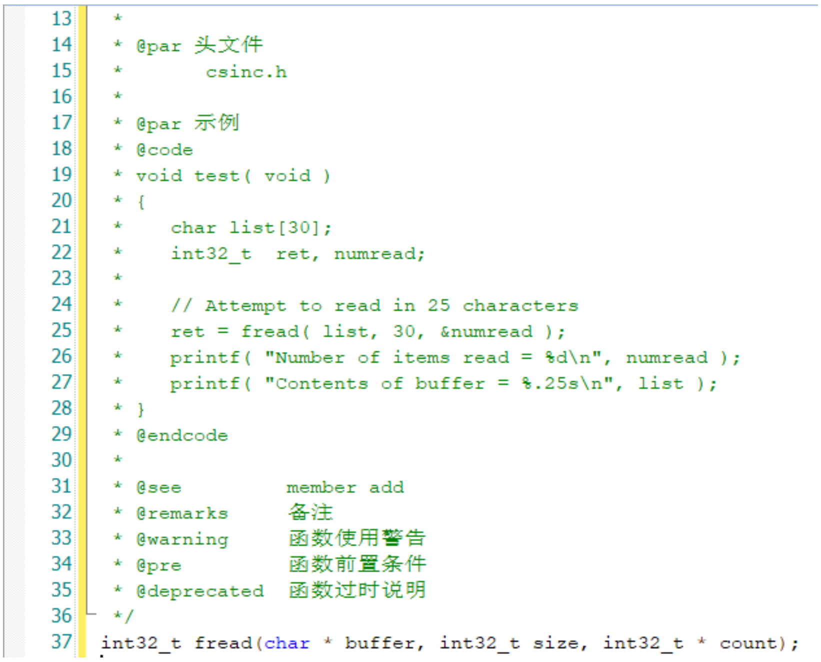
You can modify a file declaration based on the demo and program codes to declare the file as below.



```
/**
  ******************************************************************************
  * @file     FU68xx_2_DMA.h
  * @author   Fortiortech MCU Team
  * @version V1.0
  * @date     10-Apr-2017
  ******************************************************************************
  * @attention
  * THE PRESENT FIRMWARE WHICH IS FOR GUIDANCE ONLY AIMS AT FU6812,FU6861.
  ******************************************************************************
  */
#ifndef __FU68XX_2_DMA_H__
#define __FU68XX_2_DMA_H__

#include "FU68xx_2_MCU.h"

/******************************************************************************///Defined CMD(Don't touch)
// DMA管道参数表
 #define UART_XDATA       0x00                            // DMA管道--UART->FT_XDATA
 #define XDATA_UART       (DMACFG0)                       // DMA管道--UART<-FT_XDATA
 #define I2C_XDATA        (DMACFG1)                       // DMA管道--I2C ->FT_XDATA
 #define XDATA_I2C        (DMACFG0 | DMACFG1)             // DMA管道--I2C <-FT_XDATA
 #define SPI_XDATA        (DMACFG2)                       // DMA管道--SPI ->FT_XDATA
 #define XDATA_SPI        (DMACFG2 | DMACFG0)             // DMA管道--SPI <-FT_XDATA

// DMA模块参数配置表
 #define ENIE DMAIE                                       // 使能DMA中断
 #define DISIE 0x00                                       // 禁能DMA中断
 #define FLSB 0x00                                        // DMA先发低8位
 #define FHSB ENDIAN                                      // DMA先发高8位
/******************************************************************************///Config
/******************************************************************************///External Function
#define Wait_DMA(a)        while (ReadBit(*(&DMA0_CR0 + a), DMABSY))
#define Switch_DMA(a)      SetBit(*(&DMA0_CR0 + a), DMAEN | DMABSY)

extern void Init_DMA(uint8 IEMod, uint8 FirstMod);
extern void Set_DMA(uint8 Ch, uint8 Pipe, uint16* Addr, uint8 Len);
extern void Set_DBG_DMA(uint16* Addr);

#endif
```

Figure 1-11

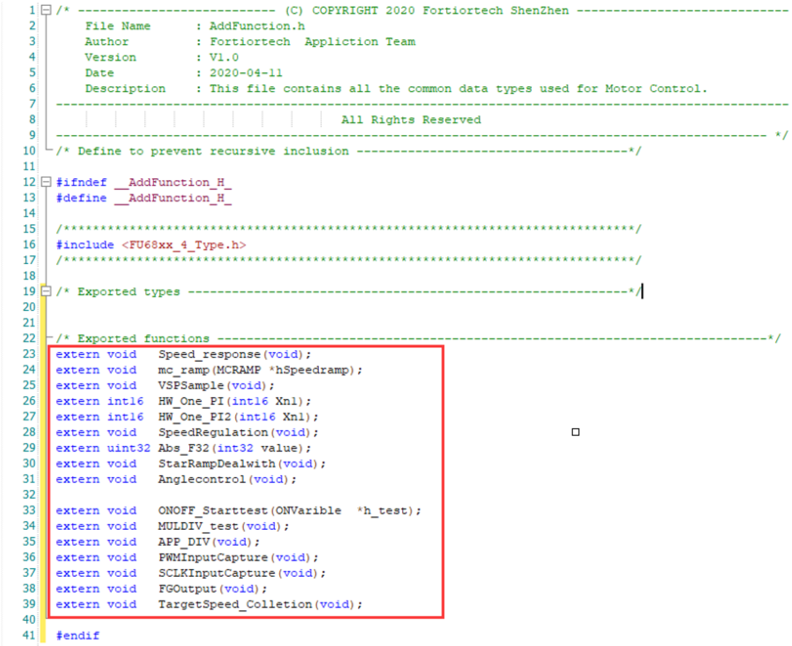
- Function: Click “Remark” → “Demo” → “Function” to add a function declaration sample in the header.h file.



```
  *
  * @par 头文件
  *        csinc.h
  *
  * @par 示例
  * @code
  * void test( void )
  * {
  *    char list[30];
  *    int32_t  ret, numread;
  *
  *    // Attempt to read in 25 characters
  *    ret = fread( list, 30, &numread );
  *    printf( "Number of items read = %d\n", numread );
  *    printf( "Contents of buffer = %.25s\n", list );
  * }
  * @endcode
  *
  * @see          member add
  * @remarks      备注
  * @warning      函数使用警告
  * @pre          函数前置条件
  * @deprecated   函数过时说明
  */
int32_t fread(char * buffer, int32_t size, int32_t * count);
```

Figure 1-12

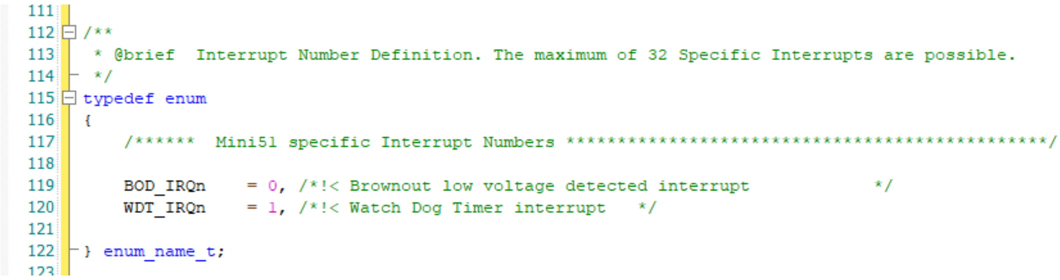
You can modify the function declaration based on the demo and program codes as below.



```
/* -------------------------- (C) COPYRIGHT 2020 Fortiortech ShenZhen ----------------------------
    File Name       : AddFunction.h
    Author          : Fortiortech  Appliction Team
    Version         : V1.0
    Date            : 2020-04-11
    Description     : This file contains all the common data types used for Motor Control.
--------------------------------------------------------------------------------------------------
                                        All Rights Reserved
------------------------------------------------------------------------------------------------- */
/* Define to prevent recursive inclusion -------------------------------------*/

#ifndef __AddFunction_H_
#define __AddFunction_H_

/******************************************************************************/
#include <FU68xx_4_Type.h>
/******************************************************************************/

/* Exported types -------------------------------------------------------------*/


/* Exported functions ----------------------------------------------------------------------*/
extern void    Speed_response(void);
extern void    mc_ramp(MCRAMP *hSpeedramp);
extern void    VSPSample(void);
extern int16   HW_One_PI(int16 Xn1);
extern int16   HW_One_PI2(int16 Xn1);
extern void    SpeedRegulation(void);
extern uint32 Abs_F32(int32 value);
extern void    StarRampDealwith(void);
extern void    Anglecontrol(void);

extern void    ONOFF_Starttest(ONVarible  *h_test);
extern void    MULDIV_test(void);
extern void    APP_DIV(void);
extern void    PWMInputCapture(void);
extern void    SCLKInputCapture(void);
extern void    FGOutput(void);
extern void    TargetSpeed_Colletion(void);

#endif
```

Figure 1-13

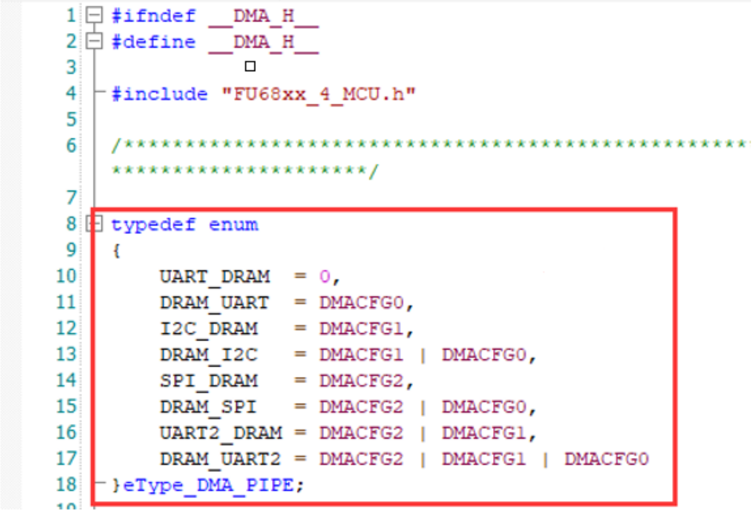
- ➢ Enum: Click “Remark” → “Demo” → “Enum” to add a enumeration type declaration sample in the header.h file.



```

/**
  * @brief  Interrupt Number Definition. The maximum of 32 Specific Interrupts are possible.
  */
typedef enum
{
    /******  Mini51 specific Interrupt Numbers ***********************************************/

    BOD_IRQn     = 0,  /*!< Brownout low voltage detected interrupt              */
    WDT_IRQn     = 1,  /*!< Watch Dog Timer interrupt   */

} enum_name_t;
```

Figure 1-14

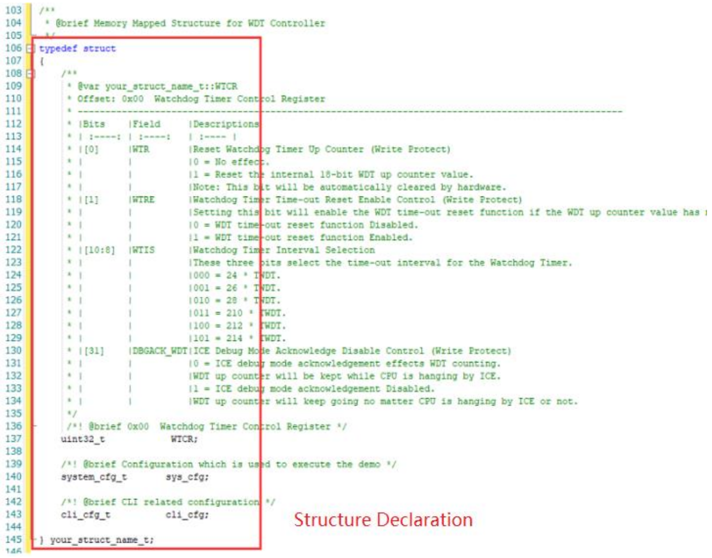
You can modify the enumeration type declaration based on the demo and program codes as below.



```
#ifndef __DMA_H__
#define __DMA_H__

#include "FU68xx_4_MCU.h"

/*****************************************************
*********************/

typedef enum
{
    UART_DRAM   = 0,
    DRAM_UART   = DMACFG0,
    I2C_DRAM    = DMACFG1,
    DRAM_I2C    = DMACFG1 | DMACFG0,
    SPI_DRAM    = DMACFG2,
    DRAM_SPI    = DMACFG2 | DMACFG0,
    UART2_DRAM = DMACFG2 | DMACFG1,
    DRAM_UART2 = DMACFG2 | DMACFG1 | DMACFG0
}eType_DMA_PIPE;
```

Figure 1-15

- Struct: Click “Remark” → “Demo” → “Struct” to add a structure declaration sample in the header.h file.

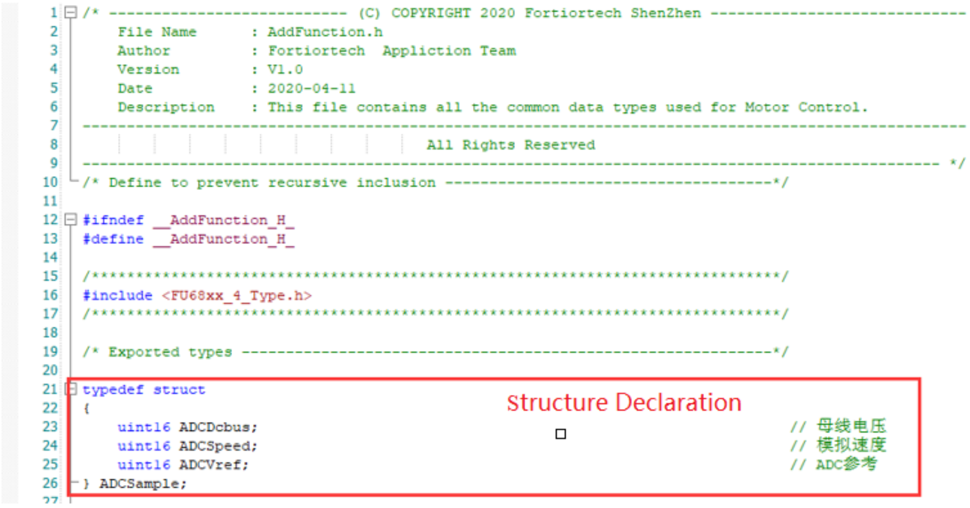


```
/**
 * @brief Memory Mapped Structure for WDT Controller
 */
typedef struct
{
    /**
     * @var your_struct_name_t::WTCR
     * Offset: 0x00  Watchdog Timer Control Register
     * ---------------------------------------------------------------------------------------------
     * |Bits    |Field     |Descriptions
     * | :----: | :----:   | :---- |
     * |[0]     |WTR       |Reset Watchdog Timer Up Counter (Write Protect)
     * |        |          |0 = No effect.
     * |        |          |1 = Reset the internal 18-bit WDT up counter value.
     * |        |          |Note: This bit will be automatically cleared by hardware.
     * |[1]     |WTRE      |Watchdog Timer Time-out Reset Enable Control (Write Protect)
     * |        |          |Setting this bit will enable the WDT time-out reset function if the WDT up counter value has
     * |        |          |0 = WDT time-out reset function Disabled.
     * |        |          |1 = WDT time-out reset function Enabled.
     * |[10:8]  |WTIS      |Watchdog Timer Interval Selection
     * |        |          |These three bits select the time-out interval for the Watchdog Timer.
     * |        |          |000 = 24 * TWDT.
     * |        |          |001 = 26 * TWDT.
     * |        |          |010 = 28 * TWDT.
     * |        |          |011 = 210 * TWDT.
     * |        |          |100 = 212 * TWDT.
     * |        |          |101 = 214 * TWDT.
     * |[31]    |DBGACK_WDT|ICE Debug Mode Acknowledge Disable Control (Write Protect)
     * |        |          |0 = ICE debug mode acknowledgement effects WDT counting.
     * |        |          |WDT up counter will be kept while CPU is hanging by ICE.
     * |        |          |1 = ICE debug mode acknowledgement Disabled.
     * |        |          |WDT up counter will keep going no matter CPU is hanging by ICE or not.
     */
    /*! @brief 0x00  Watchdog Timer Control Register */
    uint32_t            WTCR;

    /*! @brief Configuration which is used to execute the demo */
    system_cfg_t        sys_cfg;

    /*! @brief CLI related configuration */
    cli_cfg_t           cli_cfg;

} your_struct_name_t;
```

Structure Declaration

Figure 1-16

You can modify the Structure type declaration based on the demo and program codes as below.

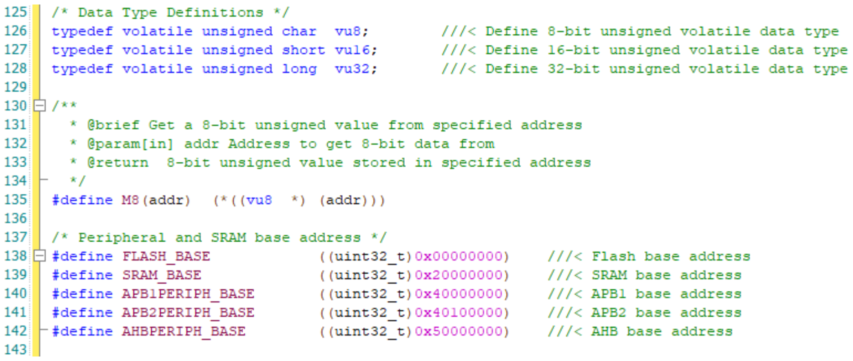


```
/* ------------------------- (C) COPYRIGHT 2020 Fortiortech ShenZhen ----------------------------
    File Name      : AddFunction.h
    Author         : Fortiortech  Appliction Team
    Version        : V1.0
    Date           : 2020-04-11
    Description    : This file contains all the common data types used for Motor Control.
--------------------------------------------------------------------------------------------------
                                          All Rights Reserved
------------------------------------------------------------------------------------------------ */
/* Define to prevent recursive inclusion ---------------------------------------*/
#ifndef __AddFunction_H_
#define __AddFunction_H_

/******************************************************************************/
#include <FU68xx_4_Type.h>
/******************************************************************************/

/* Exported types -------------------------------------------------------------*/
typedef struct
{
    uint16 ADCDcbus;                                                              // 母线电压
    uint16 ADCSpeed;                                                              // 模拟速度
    uint16 ADCVref;                                                               // ADC参考
} ADCSample;
```

Structure Declaration

Figure 1-17

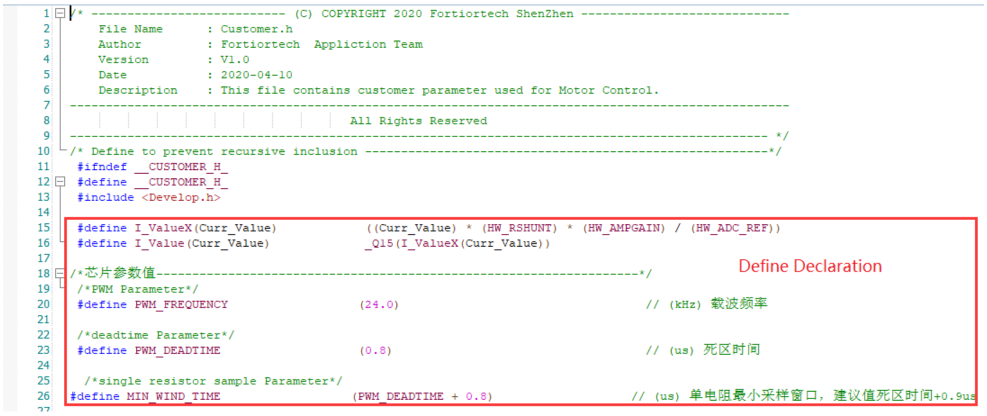
- Define: Click “Remark” → “Demo” → “Define” to add a define declaration sample in the header.h file..



```
/* Data Type Definitions */
typedef volatile unsigned char  vu8;        ///< Define 8-bit unsigned volatile data type
typedef volatile unsigned short vu16;       ///< Define 16-bit unsigned volatile data type
typedef volatile unsigned long  vu32;       ///< Define 32-bit unsigned volatile data type

/**
    * @brief Get a 8-bit unsigned value from specified address
    * @param[in] addr Address to get 8-bit data from
    * @return  8-bit unsigned value stored in specified address
    */
#define M8(addr)   (*((vu8  *) (addr)))

/* Peripheral and SRAM base address */
#define FLASH_BASE              ((uint32_t)0x00000000)    ///< Flash base address
#define SRAM_BASE               ((uint32_t)0x20000000)    ///< SRAM base address
#define APB1PERIPH_BASE         ((uint32_t)0x40000000)    ///< APB1 base address
#define APB2PERIPH_BASE         ((uint32_t)0x40100000)    ///< APB2 base address
#define AHBPERIPH_BASE          ((uint32_t)0x50000000)    ///< AHB base address
```

Figure 1-18

You can modify the define declaration based on the demo and program codes as below.

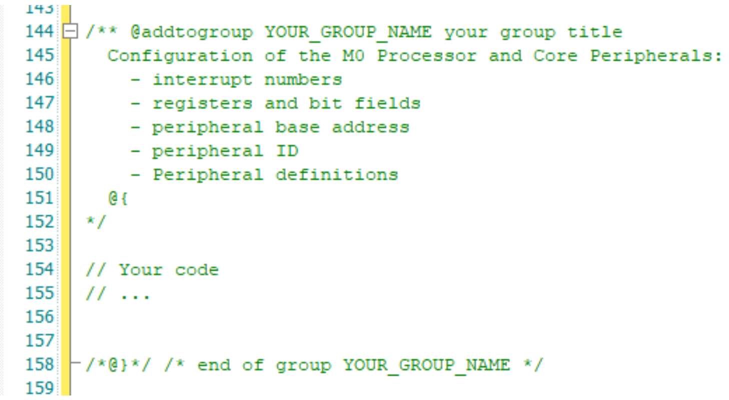


```
/* --------------------------- (C) COPYRIGHT 2020 Fortiortech ShenZhen -----------------------------
    File Name      : Customer.h
    Author         : Fortiortech  Appliction Team
    Version        : V1.0
    Date           : 2020-04-10
    Description    : This file contains customer parameter used for Motor Control.
----------------------------------------------------------------------------------------------------
                                         All Rights Reserved
------------------------------------------------------------------------------------------------- */
/* Define to prevent recursive inclusion -------------------------------------------------------*/
 #ifndef __CUSTOMER_H_
 #define __CUSTOMER_H_
 #include <Develop.h>

 #define I_ValueX(Curr_Value)            ((Curr_Value) * (HW_RSHUNT) * (HW_AMPGAIN) / (HW_ADC_REF))
 #define I_Value(Curr_Value)             _Q15(I_ValueX(Curr_Value))

/*芯片参数值------------------------------------------------------------------*/
 /*PWM Parameter*/
 #define PWM_FREQUENCY                  (24.0)                              // (kHz) 载波频率

 /*deadtime Parameter*/
 #define PWM_DEADTIME                   (0.8)                               // (us) 死区时间

  /*single resistor sample Parameter*/
#define MIN_WIND_TIME                  (PWM_DEADTIME + 0.8)               // (us) 单电阻最小采样窗口，建议值死区时间+0.9us
```

Define Declaration

Figure 1-19

- ➢ Group: Click “Remark” → “Demo” → “Group” to add a group declaration sample in the header.h file..

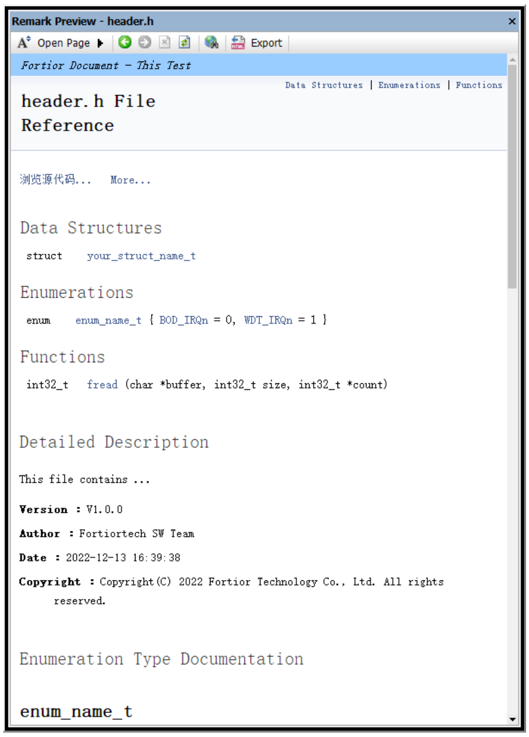


```
/** @addtogroup YOUR_GROUP_NAME your group title
  Configuration of the M0 Processor and Core Peripherals:
    - interrupt numbers
    - registers and bit fields
    - peripheral base address
    - peripheral ID
    - Peripheral definitions
  @{
*/

// Your code
// ...


/*@}*/ /* end of group YOUR_GROUP_NAME */
```

Figure 1-20

- ■ File Head Comment: Click “Remark” → “File Head Comment” to add a file header comment sample in the header.h file. The shortcut is Ctrl+Alt+K;
- ■ Symbol Comment: Click “Remark” → “Symbol Comment” to add a symbol comment sample in the header.h file. The shortcut is Ctrl+Shift+K;
- ■ brief: Click “Remark” → “brief” to add a brief declaration sample in the header.h file. The shortcut is Ctrl+L;
- ■ brief(After Line): Click “Remark” → “brief(After Line)” to add a brief declaration sample behind the code line in the header.h file. The shortcut is Ctrl+Shift +L;
- ■ Image(Non Chinese name): Click “Remark”→ “Image(Non Chinese name)” to insert a picture and add a picture declaration sample in the header.h file;
- ■ Attention: Click “Remark” → “Attention” to add an attention declaration sample in the header.h file;
- ■ Copyright: Click “Remark” → “Copyright” to add a copyright information declaration sample in the header.h file;

- Example: Click “Remark” → “Example” to add an example codes declaration in the header.h file;
- Include: Click “Remark” → “Include” to add a header file declaration sample in the header.h file;
- See: Click “Remark” → “See” to add a view member declaration sample in the header.h file;
- List: Click “Remark” → “List” to add a list declaration sample in the header.h file;
- Params: Click “Remark” → “Params” to add a parameters declaration sample in the header.h file;
- Ref: Click “Remark” → “Ref” to add a reference declaration sample in the header.h file;
- Mult Ref: Click “Remark” → “Mult Ref” to add a multiple references declaration sample in the header.h file;
- Table: Click “Remark” → “Table” to add an author information declaration sample in the header.h file;
- Table2: Click “Remark” → “Table2” to add an author information comment sample;
- Else: Click “Remark” → “Else” to add a declaration sample about remarks, warning, preconditions and deprecated information comment samples in the header.h file;
- User Define: Generate user-defined comments;
- Remark Settings: Set the path for adding pictures;
- Preview Help Of File : Click or “Remark” → “Preview Help Of File” to show the Remark Preview window;


Remark Preview - header.h
Open Page
Export
Data Structures | Enumerations | Functions
浏览源代码...
More...

Fortior Document – This Test

header.h File Reference

Data Structures

```
struct    your_struct_name_t
```

Enumerations

```
enum    enum_name_t { BOD_IRQn = 0, WDT_IRQn = 1 }
```

Functions

```
int32_t   fread (char *buffer, int32_t size, int32_t *count)
```

Detailed Description

This file contains ...

Version : V1.0.0

Author : Fortiortech SW Team

Date : 2022-12-13 16:39:38

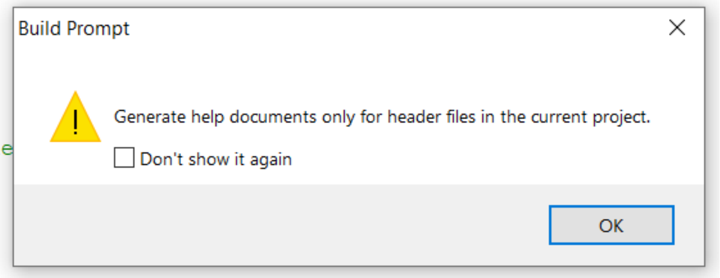
Copyright : Copyright(C) 2022 Fortior Technology Co., Ltd. All rights reserved.

Enumeration Type Documentation

enum_name_t

Figure 1-21

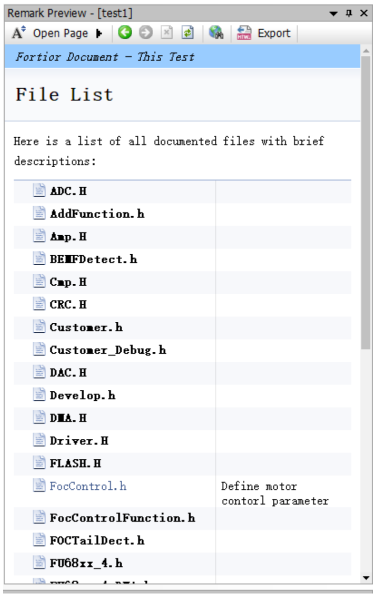
- Preview Help Of Project : Click or "Remark" → "Preview Help Of Project" to show Remark Preview window. All header.h files are listed in the window with brief descriptions, as shown in Figure 1-23;


Build Prompt

Generate help documents only for header files in the current project.

□ Don't show it again

Figure 1-22



Fortior Document – This Test

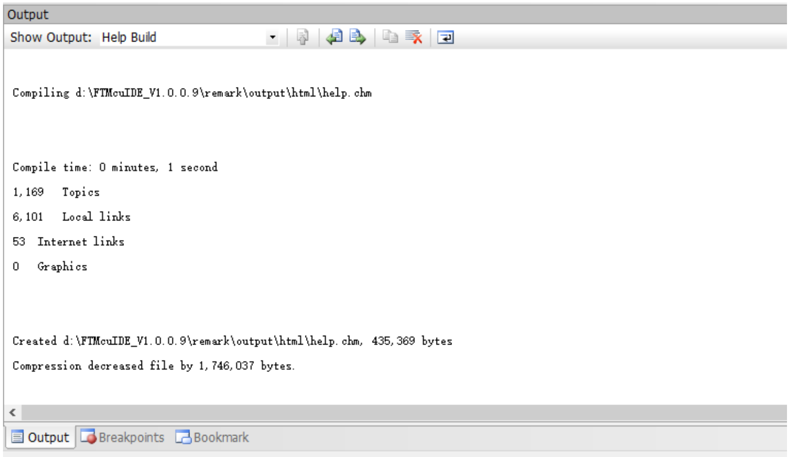
File List

Here is a list of all documented files with brief descriptions:

ADC.H	
AddFunction.h	
Amp.H	
BEMFDetect.h	
Cmp.H	
CRC.H	
Customer.h	
Customer_Debug.h	
DAC.H	
Develop.h	
DMA.H	
Driver.H	
FLASH.H	
FocControl.h	Define motor contorl parameter
FocControlFunction.h	
FOCTailDect.h	
FU68xx_4.h	

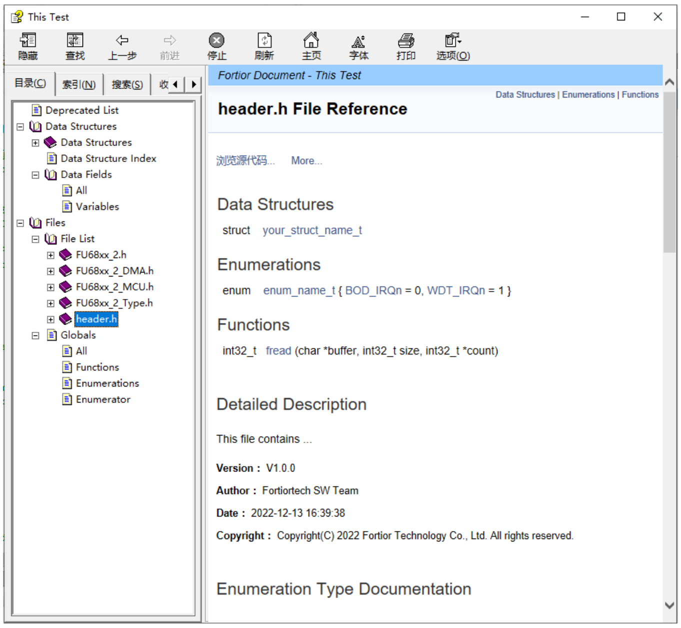
Figure 1-23

- Generate.chm : Click or "Remark" → "Generate.chm" to generate .chm help files. The Output window displays the compilation process and results of the generated help files, as shown in Figure 1-24.



```
Compiling d:\FTMcuIDE_V1.0.0.9\remark\output\html\help.chm



Compile time: 0 minutes, 1 second
1,169   Topics
6,101   Local links
53  Internet links
0   Graphics



Created d:\FTMcuIDE_V1.0.0.9\remark\output\html\help.chm, 435,369 bytes
Compression decreased file by 1,746,037 bytes.
```

Figure 1-24



Fortior Document - This Test

Data Structures | Enumerations | Functions

header.h File Reference

浏览源代码... More...

Data Structures

struct your_struct_name_t

Enumerations

enum enum_name_t { BOD_IRQn = 0, WDT_IRQn = 1 }

Functions

int32_t fread (char *buffer, int32_t size, int32_t *count)

Detailed Description

This file contains ...

Version : V1.0.0

Author : Fortiortech SW Team

Date : 2022-12-13 16:39:38

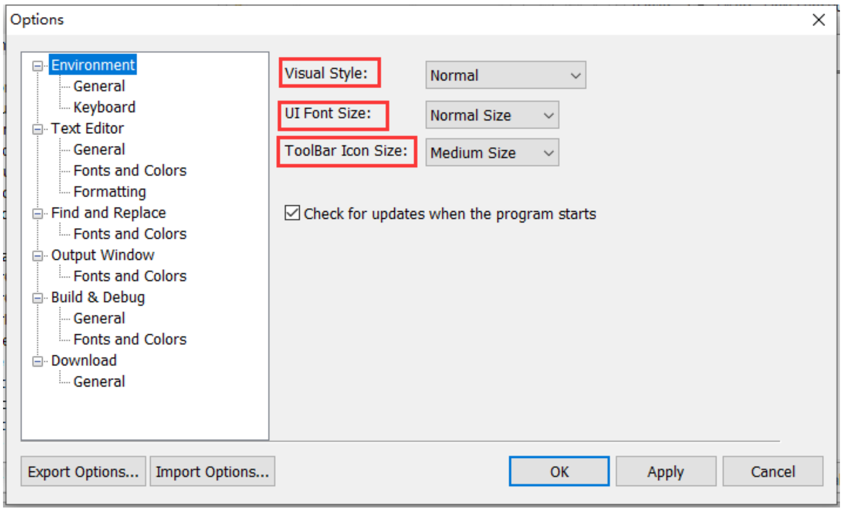
Copyright : Copyright(C) 2022 Fortior Technology Co., Ltd. All rights reserved.

Enumeration Type Documentation

Figure 1-25

1.1.8 Tools Menu

Tools: Click to show the Options window. You can set the interface style, text font size, colors and so on.



Visual Style: Normal

UI Font Size: Normal Size

ToolBar Icon Size: Medium Size

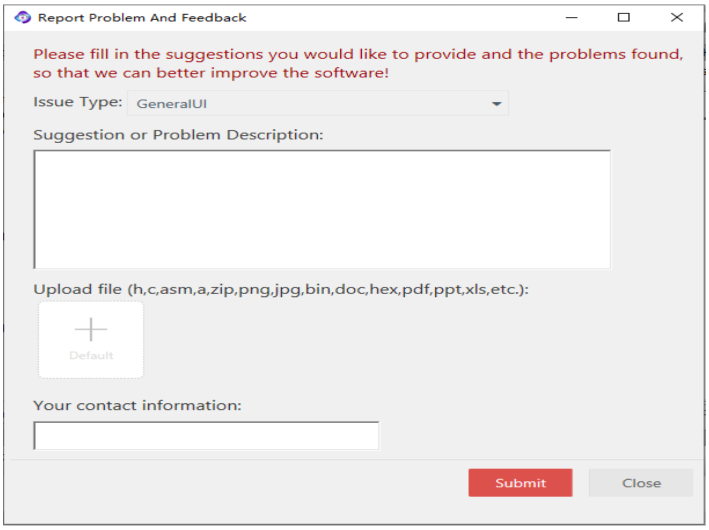
☑ Check for updates when the program starts

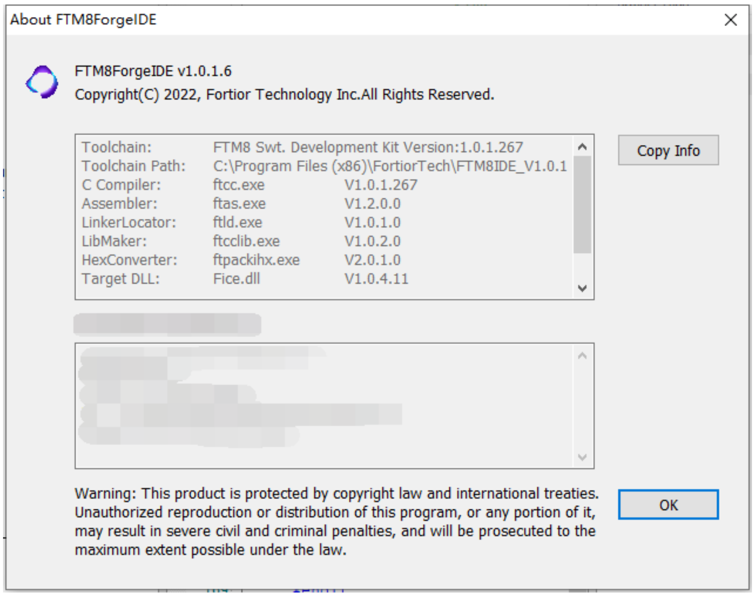
Figure 1-26

1.1.9 Window Menu

Window Menu	Description
New Window	Open a window to display the file.
Close All	Close all opening files.
Rearrange Windows	Rearrange the window.
Windows	Switch among the open files.

1.1.10 Help Menu

Help Menu	Description
Show Start Page	You create a new solution, open an existing solution and quickly open a solution in the recently used solution list.
Fortiortech on The Web	Visit Fortior Technology website.
User Manual (English)	Open the user manual in English
User Manual (Simplified Chinese)	Open the user manual in Chinese
Programming Guide	There're some manual: Porting Guide(Simplified Chinese), FTM8 C Programming Guide (Simplified Chinese) and FAQ.
Report Problem& Feedback Message…	Display the feedback information interface, where you can fill in the suggestions or the problems found during use, and attach a document for explanation.  Report Problem And Feedback Please fill in the suggestions you would like to provide and the problems found, so that we can better improve the software! Issue Type: GeneralUI Suggestion or Problem Description: Upload file (h,c,asm,a,zip,png,jpg,bin,doc,hex,pdf,ppt,xls,etc.): Your contact information: Figure 1-27

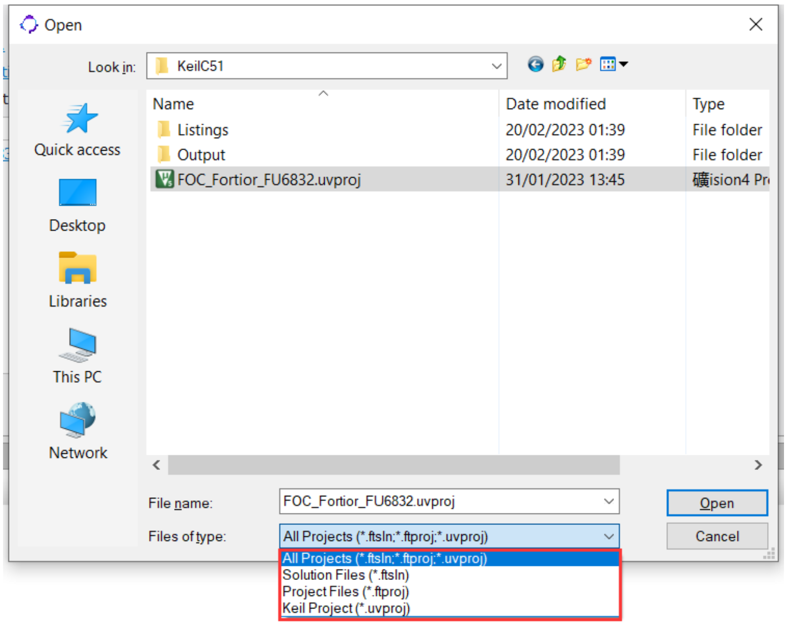
Help Menu	Description
About IDE	The information about FTM8ForgeIDE.  About FTM8ForgeIDE FTM8ForgeIDE v1.0.1.6 Copyright(C) 2022, Fortior Technology Inc.All Rights Reserved. Toolchain: FTM8 Swt. Development Kit Version:1.0.1.267 Toolchain Path: C:\Program Files (x86)\FortiorTech\FTM8IDE_V1.0.1 C Compiler: ftcc.exe V1.0.1.267 Assembler: ftas.exe V1.2.0.0 LinkerLocator: ftld.exe V1.0.1.0 LibMaker: ftcclib.exe V1.0.2.0 HexConverter: ftpackihx.exe V2.0.1.0 Target DLL: Fice.dll V1.0.4.11 [redacted] [redacted] Warning: This product is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law. Figure 1-28

1.2 Keil Project Convert

1.2.1 Convert the project directly and open it

First, click “File” → “Open Solution...” to select the keil project file to open. As shown in Figure 1-29, the software supports three types of files: *.ftsln, *.ftproj and *.uvproj. The default file type is “All Projects”.

Then, click the “Open”, and the software prompts “You open a keil project, which needs to be converted to a FTM8ForgeIDE project. Do you need to convert the project now?”, as shown in Figure 1-30;



Open

Look in: KeilC51

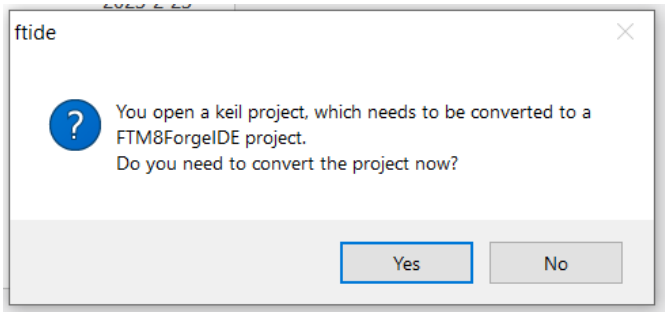
Name	Date modified	Type
Listings	20/02/2023 01:39	File folder
Output	20/02/2023 01:39	File folder
FOC_Fortior_FU6832.uvproj	31/01/2023 13:45	礦ision4 Pr

File name: FOC_Fortior_FU6832.uvproj

Files of type: All Projects (*.ftsln;*.ftproj;*.uvproj)

All Projects (*.ftsln;*.ftproj;*.uvproj)
Solution Files (*.ftsln)
Project Files (*.ftproj)
Keil Project (*.uvproj)

Figure 1-29



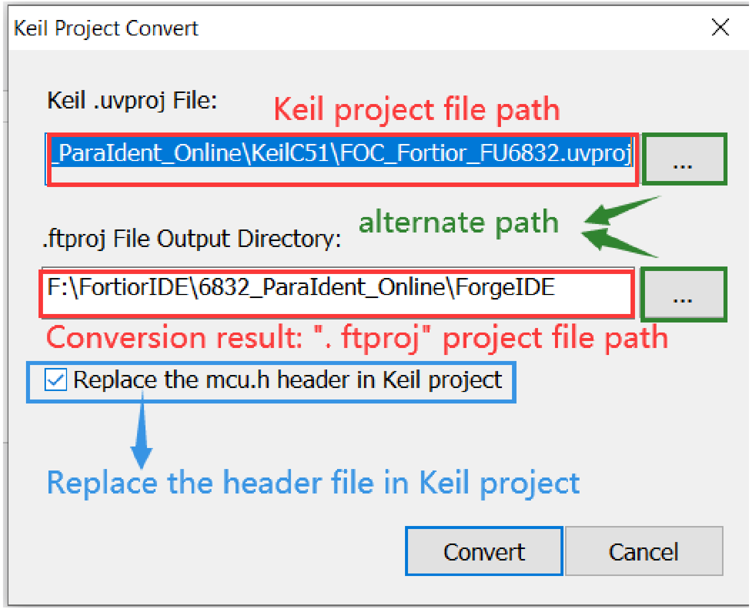
ftide

You open a keil project, which needs to be converted to a FTM8ForgeIDE project.
Do you need to convert the project now?

Figure 1-30

If you click "Yes", the software displays the file path of the Keil project and the converted ".ftproj" project, and you can click the ... to change the path address, as shown in Figure 1-31. If "Replace the mcu.h header in Keil project" is checked, the chip configuration standard header file of FTM8ForgeIDE overwrites the header file corresponding to Keil. If it is not checked, it is not overwritten.

If you click the "No", the software does not open the Keil project file without any operation, and the current page remains unchanged.



Keil Project Convert

Keil .uvproj File: _ParaIdent_Online\KeilC51\FOC_Fortior_FU6832.uvproj

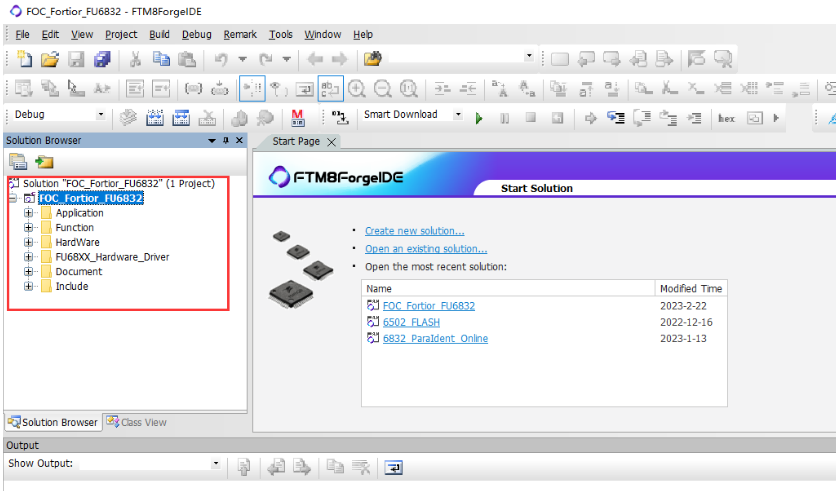
.ftproj File Output Directory: F:\FortiorIDE\6832_ParaIdent_Online\ForgeIDE

☑ Replace the mcu.h header in Keil project

Keil project file path
...
alternate path
...
Conversion result: ". ftproj" project file path
Replace the header file in Keil project
Convert
Cancel

Figure 1-31

Finally, click “Convert” to convert. If the conversion is successful, the software opens the project directly, as shown in Figure 1-32, and the path of the conversion result (i.e. the converted “.ftproj” project file) pops up, as shown in Figure 1-33.

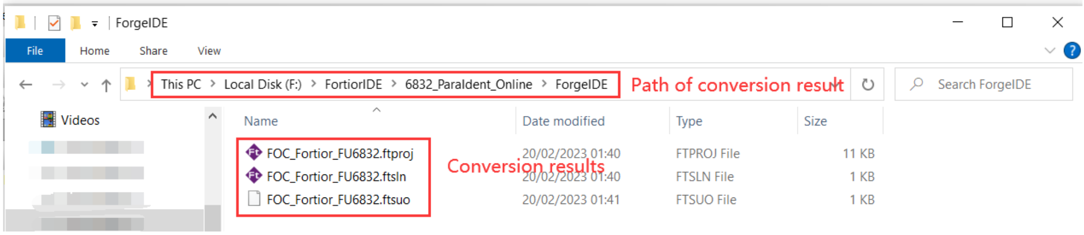


Solution "FOC_Fortior_FU6832" (1 Project)
- FOC_Fortior_FU6832
  - Application
  - Function
  - HardWare
  - FU68XX_Hardware_Driver
  - Document
  - Include

- Create new solution...
- Open an existing solution...
- Open the most recent solution:

Name	Modified Time
FOC_Fortior_FU6832	2023-2-22
6502_FLASH	2022-12-16
6832_Paraldent_Online	2023-1-13

Figure 1-32

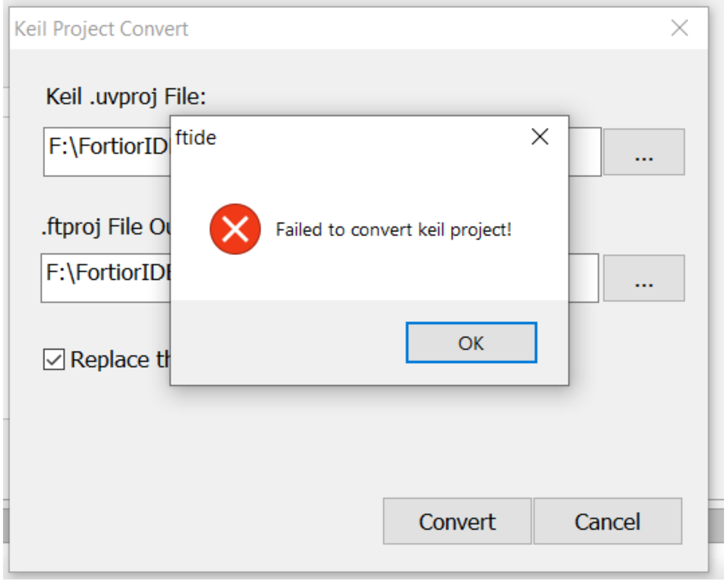


This PC › Local Disk (F:) › FortiorIDE › 6832_Paraldent_Online › ForgeIDE

Name	Date modified	Type	Size
FOC_Fortior_FU6832.ftproj	20/02/2023 01:40	FTPROJ File	11 KB
FOC_Fortior_FU6832.ftsln	20/02/2023 01:40	FTSLN File	1 KB
FOC_Fortior_FU6832.ftsuo	20/02/2023 01:41	FTSUO File	1 KB

Figure 1-33

If the conversion fails, the software prompts “Failed to convert keil project!”, as shown in Figure 1-34.



Keil .uvproj File:
F:\FortiorID

.ftproj File O
F:\FortiorIDI

☑ Replace tl

Failed to convert keil project!

Figure 1-34

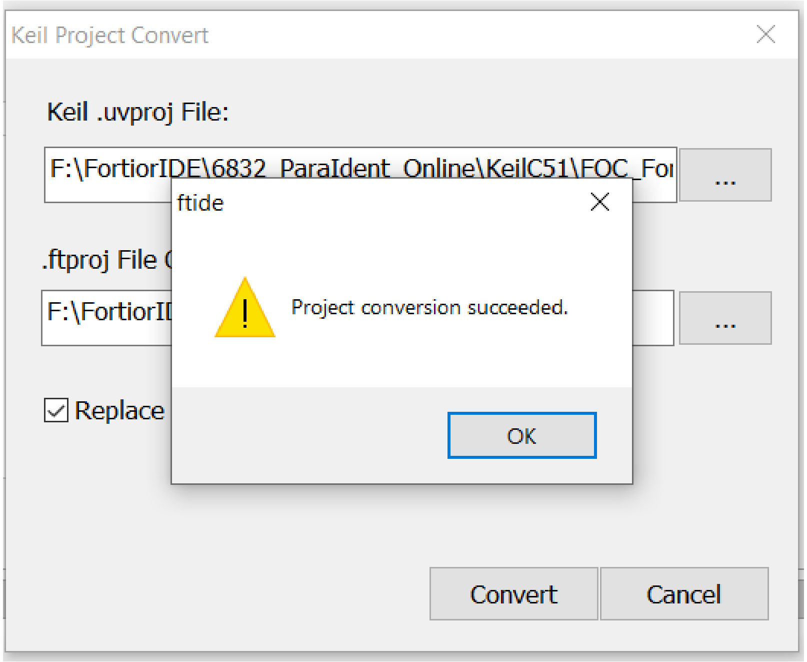
1.2.2 Convert project only

First, click “File” → “Keil Project Convert” to convert the project, as shown in Figure 1-31.

Then, select the Keil project file to be converted, the software generates the path of the output file ".ftproj", and you can click ... to change the output file path.

Finally, click "Convert" to convert. If the conversion fails, as shown in Figure 1-34; if the conversion succeeds, the software prompts "Project conversion succeeded.", as shown in Figure 1-35, and the software pops up the path of the conversion result (that is, the converted ".ftproj" project file), as shown in Figure 1-33.

⚠ Note: Click "Keil Project Convert" to only perform the project conversion operation. After the conversion is successful, the project is not opened. You shall manually open the project.



F:\FortiorIDE\6832_ParaIdent_Online\KeilC51\FOC_Foı

F:\FortiorIÍ

☑ Replace

Project conversion succeeded.

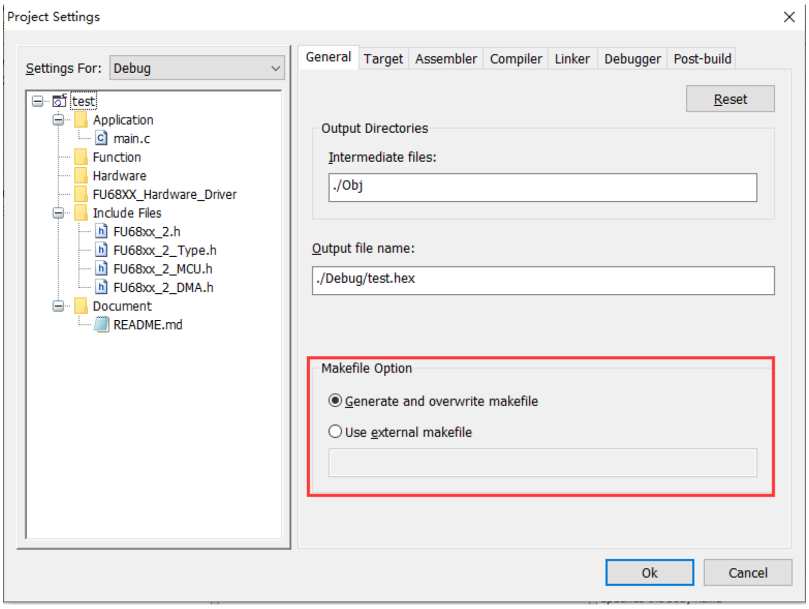
Figure 1-35

1.3 Setting

1.3.1 Setting

Click "Project" → "Settings" to set the project. The shortcut key is Alt+F.

1. General: contains the directory and name of the output file and the selection of the makefile file. Among them, "Generate and overwrite makefile" automatically generates the makefile for the selected software; "Use external makefile" selects the makefile set by the programmer, as shown in Figure 1-36.
2. Target: As shown in Figure 1-37, the choice of chip type, please refer to the "FTM C Programming Guide (Simplified Chinese)" for specific descriptions.



Settings For: Debug

Output Directories

Intermediate files: ./Obj

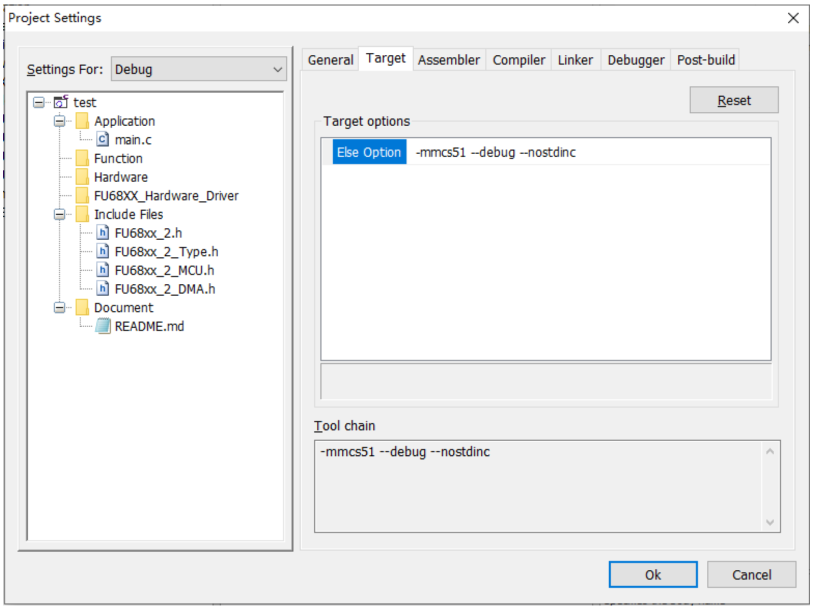
Output file name: ./Debug/test.hex

Makefile Option

● Generate and overwrite makefile

○ Use external makefile

Figure 1-36



Settings For: Debug

Target options

Else Option: -mmcs51 --debug --nostdinc

Tool chain

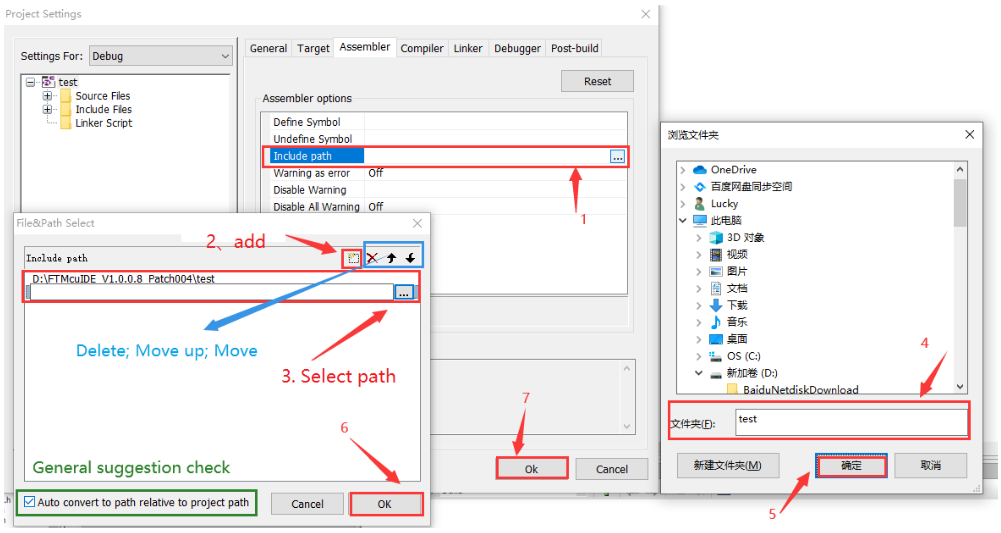
```
-mmcs51 --debug --nostdinc
```

Figure 1-37

3. Assembler:

- Define Symbol: Define macro in assembly code;
- Undefine Symbol: Cancel macro in assembly code;
- Include path: Add the address of the header file, as shown in Figure 1-38 below;
- Warning as error: The default is Off, which turns off the operation that treats warnings as errors;
- Disable Warning: Generally not set;
- Disable All Warning: The default is Off;

- Auto convert to path relative to project path: It is generally recommended to check this option.



Settings For: Debug

Include path: D:\FTMcuIDE_V1.0.0.8_Patch004\test

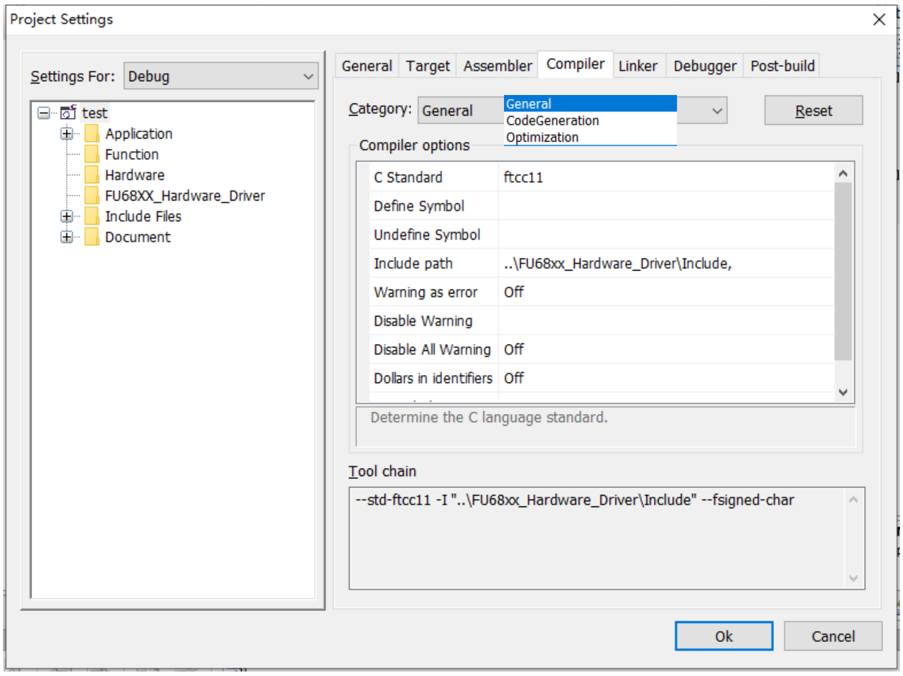
Define Symbol	
Undefine Symbol	
Include path	
Warning as error	Off
Disable Warning	
Disable All Warning	Off

文件夹(F): test

☑ Auto convert to path relative to project path

Figure 1-38

4. Compiler: Category has three categories: General, CodeGeneration and Optimization, as shown in Figure 1-39 below;



Settings For: Debug

Category: General

C Standard	ftcc11
Define Symbol	
Undefine Symbol	
Include path	..\FU68xx_Hardware_Driver\Include,
Warning as error	Off
Disable Warning	
Disable All Warning	Off
Dollars in identifiers	Off

Determine the C language standard.

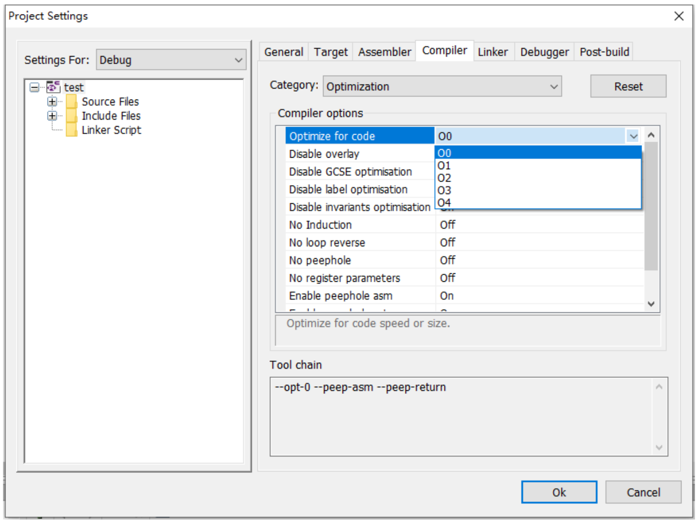
Tool chain

```
--std-ftcc11 -I "..\FU68xx_Hardware_Driver\Include" --fsigned-char
```

Figure 1-39

- C Standard: C language standard, select ftcc11 here, or other standards;
- Define Symbol: Define macros in C language;
- Undefine Symbol: Cancel macros in C language;
- Include path: Add the address of the header file;
- Warning as error: The default is Off;
- Disable Warning: Generally not set;
- Disable All Warning: The default is Off;

- Signed Char: The default value is On, indicating that the defined variable is a signed variable. For example, char char8 is a signed character;
- Reentrant intlong: The default is On;
- Reentrant float: The default is On;
- Optimize for code: code optimization level, with the following five levels available. Default is O4, as shown in Figure 1-40;



Settings For: Debug

test
Source Files
Include Files
Linker Script

Category: Optimization

Compiler options

Optimize for code	O0
Disable overlay	
Disable GCSE optimisation	
Disable label optimisation	
Disable invariants optimisation	
No Induction	Off
No loop reverse	Off
No peephole	Off
No register parameters	Off
Enable peephole asm	On

Optimize for code speed or size.

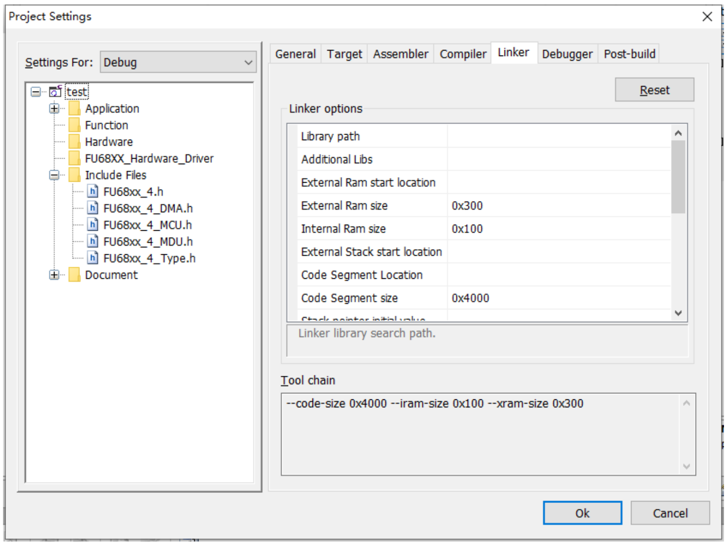
Tool chain

```
--opt-0 --peep-asm --peep-return
```

Figure 1-40

5. Linker:

- External Ram size: size of external memory, 0x300 here;
- Internal Ram size: memory size, 0x100 here;
- Code Segment size: code segment size, here 0x4000;
- Function Codeseg: Absolute function positioning, please see 1.3.2 Absolute function positioning absolute function positioning for details.



Settings For: Debug

test
Application
Function
Hardware
FU68XX_Hardware_Driver
Include Files
FU68xx_4.h
FU68xx_4_DMA.h
FU68xx_4_MCU.h
FU68xx_4_MDU.h
FU68xx_4_Type.h
Document

Linker options

Library path	
Additional Libs	
External Ram start location	
External Ram size	0x300
Internal Ram size	0x100
External Stack start location	
Code Segment Location	
Code Segment size	0x4000

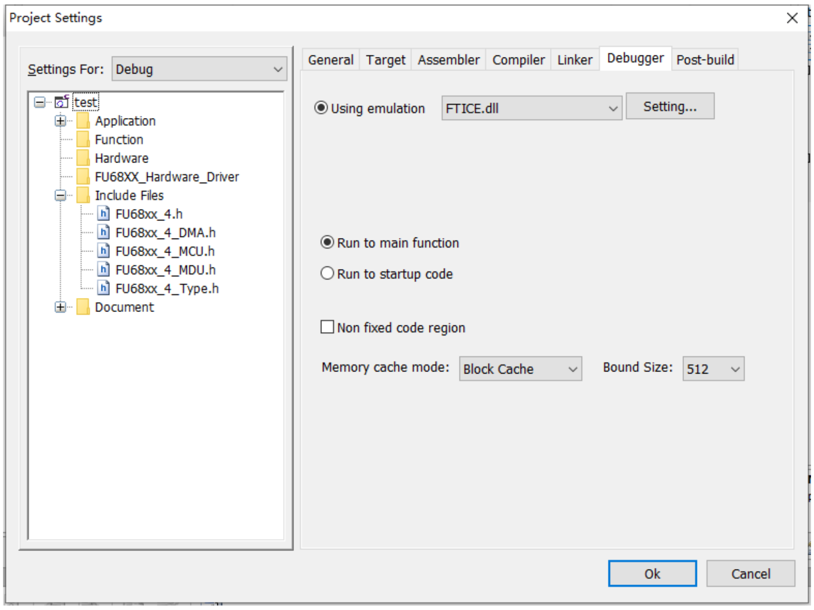
Linker library search path.

Tool chain

```
--code-size 0x4000 --iram-size 0x100 --xram-size 0x300
```

Figure 1-41

6. Debugger:
   - FTICE.dll library file is applied to FTM8ForgeIDE and Keil IDE;
   - Run to main function: Indicates that the register initialization operation is completed from the main function during debugging;
   - Run to startup code: Indicates that the debugging starts from the startup code. At this time, the register is not initialized and the last operation value is retained;



Project Settings

Settings For: Debug

- test
  - Application
  - Function
  - Hardware
  - FU68XX_Hardware_Driver
  - Include Files
    - FU68xx_4.h
    - FU68xx_4_DMA.h
    - FU68xx_4_MCU.h
    - FU68xx_4_MDU.h
    - FU68xx_4_Type.h
  - Document

General | Target | Assembler | Compiler | Linker | Debugger | Post-build

● Using emulation FTICE.dll Setting...

● Run to main function

○ Run to startup code

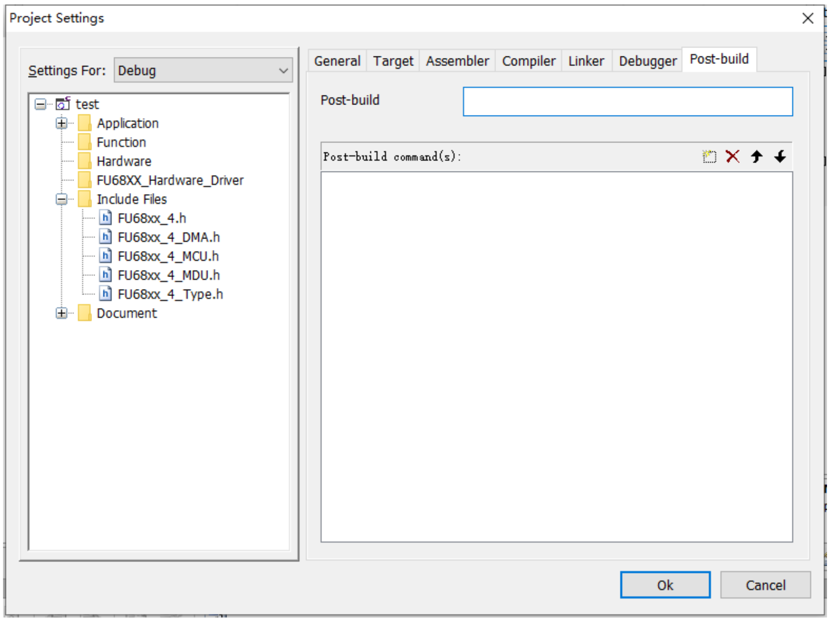
□ Non fixed code region

Memory cache mode: Block Cache Bound Size: 512

Ok Cancel

Figure 1-42

7. Post-build: Refer to the “FTM8 C Programming Guide (Simplified Chinese)”.



Project Settings

Settings For: Debug

- test
  - Application
  - Function
  - Hardware
  - FU68XX_Hardware_Driver
  - Include Files
    - FU68xx_4.h
    - FU68xx_4_DMA.h
    - FU68xx_4_MCU.h
    - FU68xx_4_MDU.h
    - FU68xx_4_Type.h
  - Document

General | Target | Assembler | Compiler | Linker | Debugger | Post-build

Post-build

Post-build command(s):

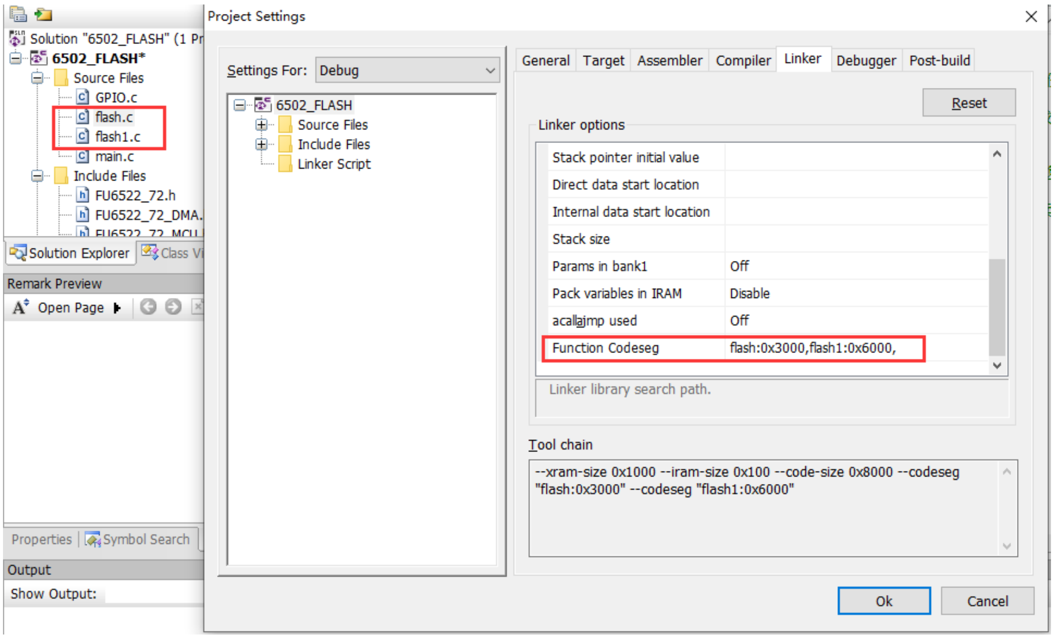
Ok Cancel

Figure 1-43

1.3.2 Absolute function positioning

Click “Project” → “Setting” → “Linker” → “Function Codeseg”, fill in the “.c” file name and corresponding address in the format shown in Figure 1-44. The two files are separated by commas, and click “OK” to compile the file. No error indicates that the function has successfully set the address;

⚠ Note: A “.c” file is only correspond to one absolute address, and multiple addresses cannot be set. However, multiple functions are written in the “.c” file, and their addresses are sorted in order.



Settings For: Debug

Linker options

Stack pointer initial value	
Direct data start location	
Internal data start location	
Stack size	
Params in bank1	Off
Pack variables in IRAM	Disable
acalljmp used	Off
Function Codeseg	flash:0x3000,flash1:0x6000,

Linker library search path.

Tool chain

```
--xram-size 0x1000 --iram-size 0x100 --code-size 0x8000 --codeseg
"flash:0x3000" --codeseg "flash1:0x6000"
```

Figure 1-44

1.4 Window Introduction

In the FTM8Forge IDE, you can dock windows at specific locations and manage them by using tag groups. You can make a window in the “suspended” state, that is, make it always in the upper layer of other windows. Changing the size and position of the “floating window”, does not affect other windows

1.4.1 Solution Browser

Solution Browser: Displays all project files in a tree structure to help users manage project files, as shown in Figure 1-45.

1.4.2 Class View

Class View: The tree structure shows the program-defined structure, macro definitions, global variables, etc., as shown in Figure 1-46.

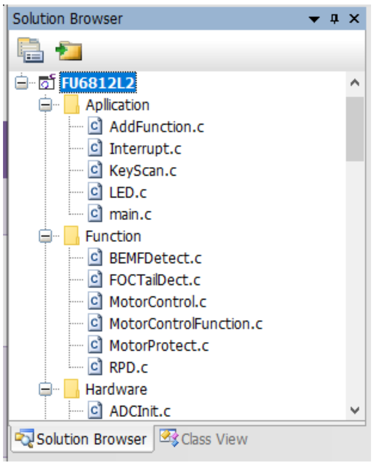


Figure 1-45

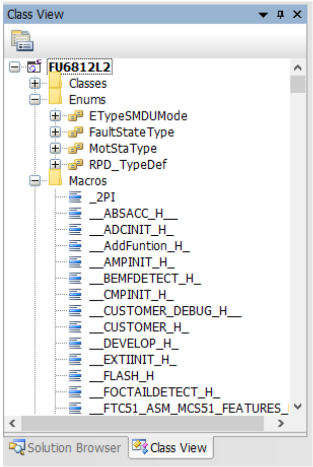
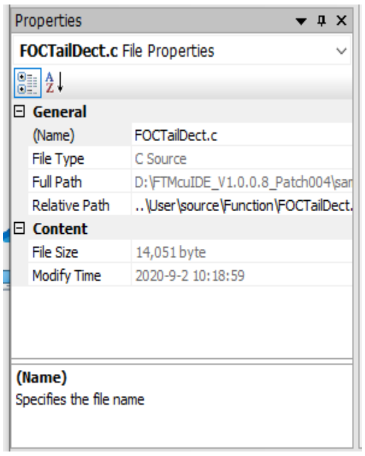


Figure 1-46


Properties

FOCTailDect.c File Properties

General

(Name)	FOCTailDect.c
File Type	C Source
Full Path	D:\FTMcuIDE_V1.0.0.8_Patch004\sar
Relative Path	..\User\source\Function\FOCTailDect.

Content

File Size	14,051 byte
Modify Time	2020-9-2 10:18:59

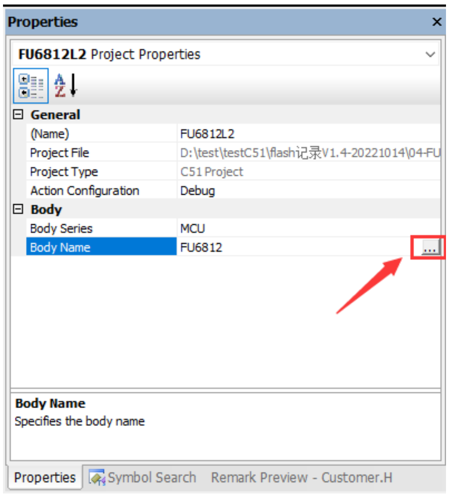
(Name)
Specifies the file name

Figure 1-47

1.4.3 Properties View

Properties View: Display the General, Body and Content of the selected project or file.,

- General: as shown in Figure 1-47
  - Name: Project Name, File Name;
  - Type: Project Type, File Type;
  - Absolute Path: Project File, Full Path
  - Relative Path;
  - Action Configuration: Debug;
- Content: as shown in Figure 1-47
  - File Size;
  - Modify Time;
- Body: as shown in Figure 1-48
  - Body Series: MCU;
  - Body Name: If you want to change the Body of the created project, you can click on the Body Name to change it, and the “SupportBody” displays the supported chip sub models, as shown in Figure 1-49.



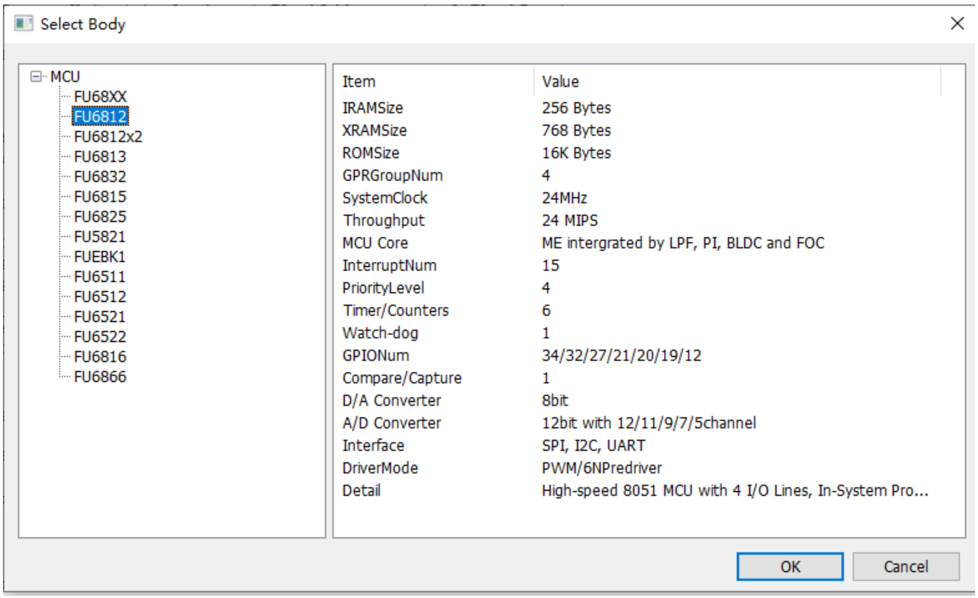
Properties

FU6812L2 Project Properties

General	
(Name)	FU6812L2
Project File	D:\test\testC51\flash记录V1.4-20221014\04-FU
Project Type	C51 Project
Action Configuration	Debug
Body	
Body Series	MCU
Body Name	FU6812

Body Name
Specifies the body name

Figure 1-48



Select Body

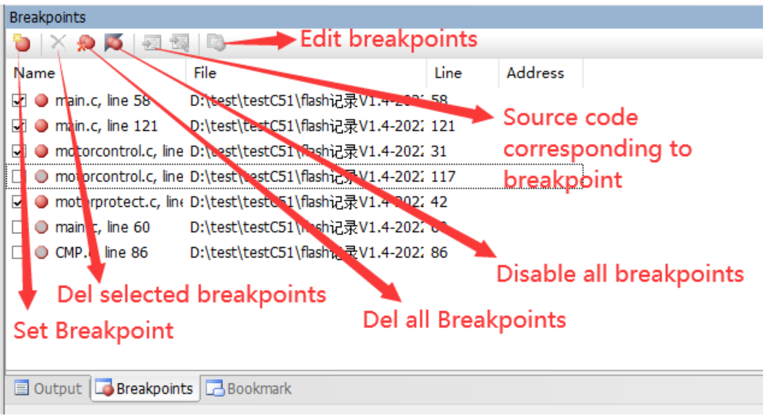
- MCU
  - FU68XX
  - FU6812
  - FU6812x2
  - FU6813
  - FU6832
  - FU6815
  - FU6825
  - FU5821
  - FUEBK1
  - FU6511
  - FU6512
  - FU6521
  - FU6522
  - FU6816
  - FU6866

Item	Value
IRAMSize	256 Bytes
XRAMSize	768 Bytes
ROMSize	16K Bytes
GPRGroupNum	4
SystemClock	24MHz
Throughput	24 MIPS
MCU Core	ME intergrated by LPF, PI, BLDC and FOC
InterruptNum	15
PriorityLevel	4
Timer/Counters	6
Watch-dog	1
GPIONum	34/32/27/21/20/19/12
Compare/Capture	1
D/A Converter	8bit
A/D Converter	12bit with 12/11/9/7/5channel
Interface	SPI, I2C, UART
DriverMode	PWM/6NPredriver
Detail	High-speed 8051 MCU with 4 I/O Lines, In-System Pro...

Figure 1-49

1.4.4 Breakpoints Window

Breakpoints Window: Displays the location of all breakpoints set by programmers in the program, including the file name, address, and number of lines, as shown in Figure 1-50 below. The shortcut key is Alt+F9.



Breakpoints

Name	File	Line	Address
☑ ● main.c, line 58	D:\test\testC51\flash记录V1.4-202:	58	
☑ ● main.c, line 121	D:\test\testC51\flash记录V1.4-202:	121	
☑ ● motorcontrol.c, line	D:\test\testC51\flash记录V1.4-202:	31	
☐ ● motorcontrol.c, line	D:\test\testC51\flash记录V1.4-202:	117	
☑ ● motorprotect.c, line	D:\test\testC51\flash记录V1.4-202:	42	
☐ ● main.c, line 60	D:\test\testC51\flash记录V1.4-202:	60	
☐ ● CMP.c, line 86	D:\test\testC51\flash记录V1.4-202:	86	

Edit breakpoints

Source code corresponding to breakpoint

Disable all breakpoints

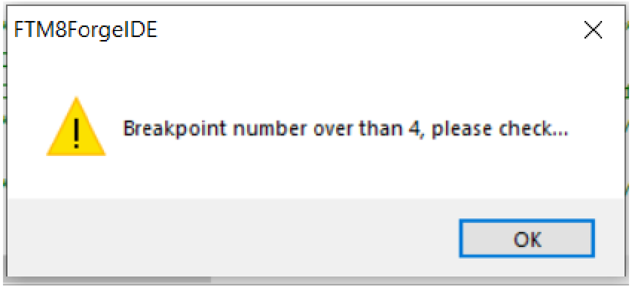
Del selected breakpoints

Del all Breakpoints

Set Breakpoint

Figure 1-50

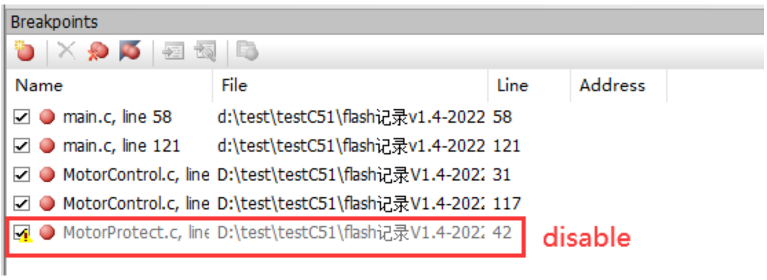
In static mode, You can more than 4 breakpoints; During dynamic debugging, the maximum number of valid breakpoints is 4, as shown in Figure1-51 and 1-52.



FTM8ForgeIDE

Breakpoint number over than 4, please check...

Figure 1-51



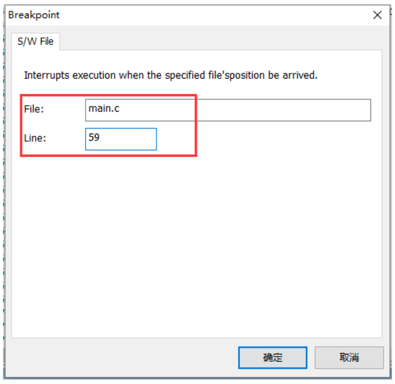
Breakpoints

Name	File	Line	Address
☑ ● main.c, line 58	d:\test\testC51\flash记录v1.4-2022	58	
☑ ● main.c, line 121	d:\test\testC51\flash记录v1.4-2022	121	
☑ ● MotorControl.c, line	D:\test\testC51\flash记录V1.4-202:	31	
☑ ● MotorControl.c, line	D:\test\testC51\flash记录V1.4-202:	117	
☑ ● MotorProtect.c, line	D:\test\testC51\flash记录V1.4-202:	42	

disable

Figure 1-52

: To set a breakpoint, enter the file name and number of lines to set the breakpoint, as shown in Figure 1-53. The setting result is shown in Figure 1-54, and the Breakpoints window shows the new breakpoint just set.



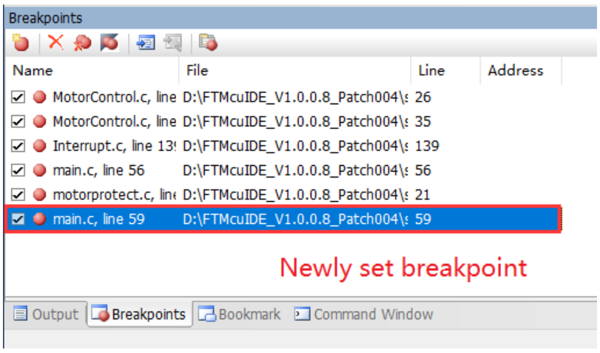
Breakpoint

S/W File

Interrupts execution when the specified file'sposition be arrived.

File: main.c

Line: 59



Breakpoints

Name	File	Line	Address
☑ ● MotorControl.c, line	D:\FTMcuIDE_V1.0.0.8_Patch004\	26	
☑ ● MotorControl.c, line	D:\FTMcuIDE_V1.0.0.8_Patch004\	35	
☑ ● Interrupt.c, line 13	D:\FTMcuIDE_V1.0.0.8_Patch004\	139	
☑ ● main.c, line 56	D:\FTMcuIDE_V1.0.0.8_Patch004\	56	
☑ ● motorprotect.c, line	D:\FTMcuIDE_V1.0.0.8_Patch004\	21	
☑ ● main.c, line 59	D:\FTMcuIDE_V1.0.0.8_Patch004\	59	

Newly set breakpoint

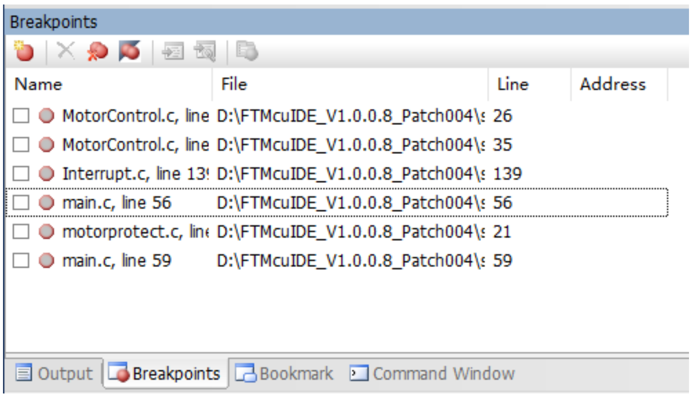
Figure 1-53

Figure 1-54

✕: Deletes the selected breakpoint.

: Delete all breakpoints.

: Make all the breakpoints set disable, that is, do not tick all breakpoints, as shown in Figure 1-55. If a breakpoint is checked, the breakpoint is enabled.

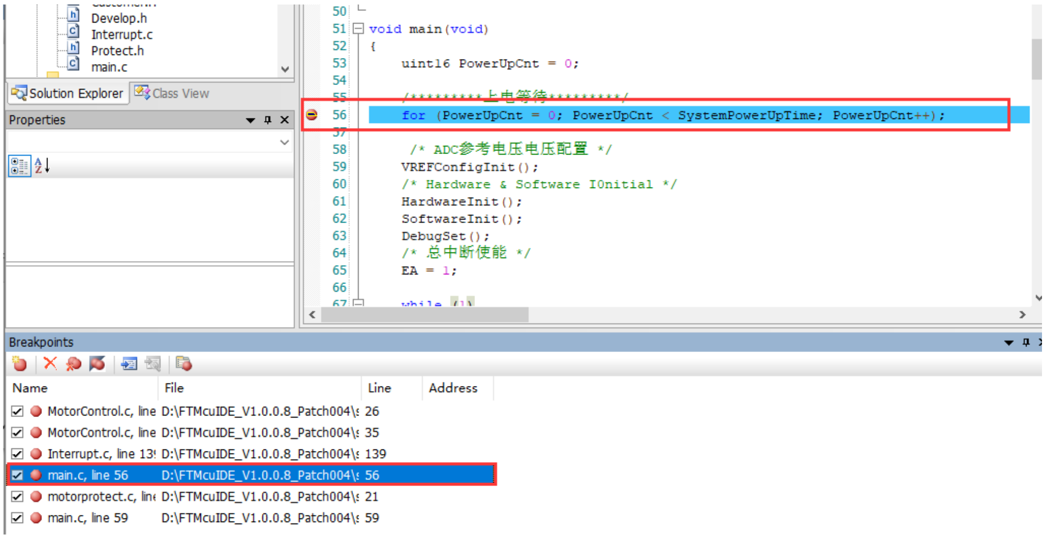


Breakpoints

Name	File	Line	Address
☐ ● MotorControl.c, line	D:\FTMcuIDE_V1.0.0.8_Patch004\	26	
☐ ● MotorControl.c, line	D:\FTMcuIDE_V1.0.0.8_Patch004\	35	
☐ ● Interrupt.c, line 13	D:\FTMcuIDE_V1.0.0.8_Patch004\	139	
☐ ● main.c, line 56	D:\FTMcuIDE_V1.0.0.8_Patch004\	56	
☐ ● motorprotect.c, line	D:\FTMcuIDE_V1.0.0.8_Patch004\	21	
☐ ● main.c, line 59	D:\FTMcuIDE_V1.0.0.8_Patch004\	59	

Figure 1-55

: The source code corresponding to the selected breakpoint, as shown in Figure 1-56.



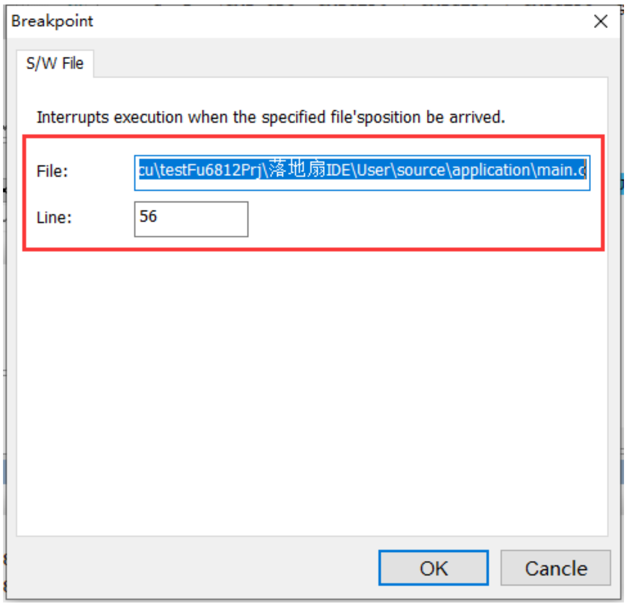
```
50  
51  void main(void)
52  {
53      uint16 PowerUpCnt = 0;
54  
55      /*********上电等待*********/
56      for (PowerUpCnt = 0; PowerUpCnt < SystemPowerUpTime; PowerUpCnt++);
57  
58       /* ADC参考电压电压配置 */
59      VREFConfigInit();
60      /* Hardware & Software IOnitial */
61      HardwareInit();
62      SoftwareInit();
63      DebugSet();
64      /* 总中断使能 */
65      EA = 1;
66  
67      while (1)
```

Breakpoints

Name	File	Line	Address
☑ ● MotorControl.c, line	D:\FTMcuIDE_V1.0.0.8_Patch004\	26	
☑ ● MotorControl.c, line	D:\FTMcuIDE_V1.0.0.8_Patch004\	35	
☑ ● Interrupt.c, line 13	D:\FTMcuIDE_V1.0.0.8_Patch004\	139	
☑ ● main.c, line 56	D:\FTMcuIDE_V1.0.0.8_Patch004\	56	
☑ ● motorprotect.c, line	D:\FTMcuIDE_V1.0.0.8_Patch004\	21	
☑ ● main.c, line 59	D:\FTMcuIDE_V1.0.0.8_Patch004\	59	

Figure 1-56

: Edit the selected breakpoint, for example, select the breakpoint main.c line56, click this option to display the settings page shown in Figure 1-57, you can modify or delete the breakpoint.



Interrupts execution when the specified file'sposition be arrived.

File: cu\testFu6812Prj\落地扇IDE\User\source\application\main.c

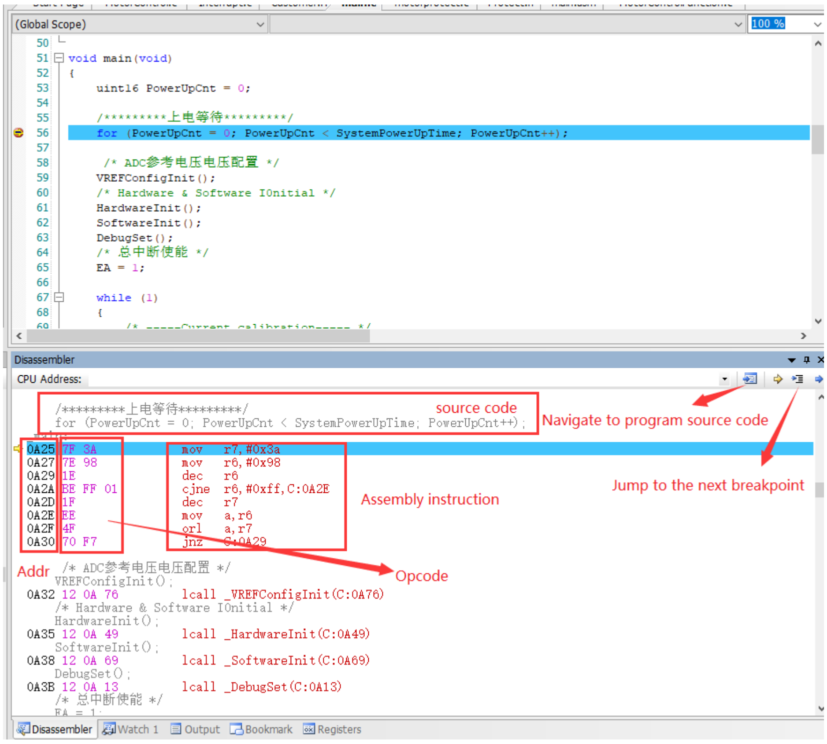
Line: 56

Figure 1-57

1.4.5 Disassembler Window

Disassembler Window: Displays the address, opcode, operand, assembly instructions, and corresponding source code, as shown in Figure 1-58. An assembly instruction is composed of an instruction mnemonic (abbreviation of the name of the instruction) and symbols representing variables, registers, and constants, that is, each assembly instruction is represented by an assembly instruction mnemonic followed by one or more symbols.

The address is the memory address where each assembly instruction is located, the opcode is the hexadecimal code corresponding to the mnemonic of the assembly instruction, such as 7F, and the operand is the operand of the assembly instruction, such as 3A corresponding to #0x3a.



```
(Global Scope)                                                100 %

void main(void)
{
    uint16 PowerUpCnt = 0;

    /*********上电等待*********/
    for (PowerUpCnt = 0; PowerUpCnt < SystemPowerUpTime; PowerUpCnt++);

     /* ADC参考电压电压配置 */
    VREFConfigInit();
    /* Hardware & Software IOnitial */
    HardwareInit();
    SoftwareInit();
    DebugSet();
    /* 总中断使能 */
    EA = 1;

    while (1)
    {
        /* -----Current calibration----- */
```

```
Disassembler
CPU Address:

    /*********上电等待*********/
    for (PowerUpCnt = 0; PowerUpCnt < SystemPowerUpTime; PowerUpCnt++);
0A25 7F 3A         mov    r7,#0x3a
0A27 7E 98         mov    r6,#0x98
0A29 1E            dec    r6
0A2A BE FF 01      cjne   r6,#0xff,C:0A2E
0A2D 1F            dec    r7
0A2E EE            mov    a,r6
0A2F 4F            orl    a,r7
0A30 70 F7         jnz    C:0A29
    /* ADC参考电压电压配置 */
    VREFConfigInit();
0A32 12 0A 76      lcall _VREFConfigInit(C:0A76)
    /* Hardware & Software IOnitial */
    HardwareInit();
0A35 12 0A 49      lcall _HardwareInit(C:0A49)
    SoftwareInit();
0A38 12 0A 69      lcall _SoftwareInit(C:0A69)
    DebugSet();
0A3B 12 0A 13      lcall _DebugSet(C:0A13)
    /* 总中断使能 */
    EA = 1;
```

Figure 1-58

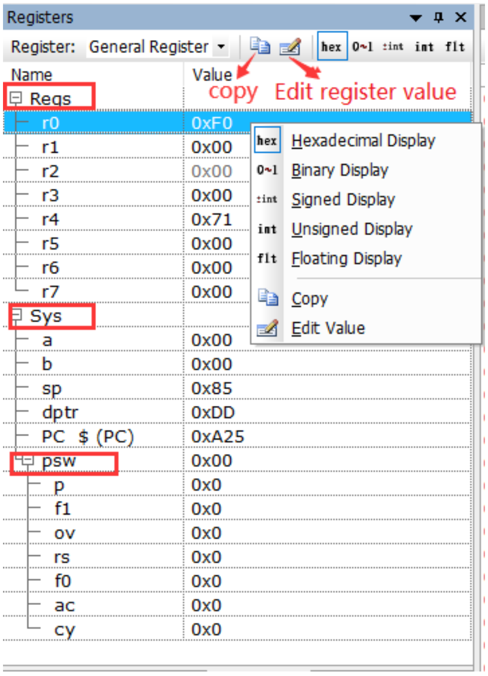
When running the program, click debug, and the assembly instructions run in sequence with the source code running line by line. The breakpoint set in the source code window also exists at the corresponding assembly instruction; On the contrary, if the breakpoint set in the source code window or disassembly window is canceled, the breakpoint corresponding to the disassembly window or source code window is also canceled.

: Click this button to locate the source code corresponding to the running assembly instruction.

: Click this button to run the program directly to the next breakpoint.

1.4.6 Registers Window

Registers Window: In the register window, three registers "All", "General Register" and "GPIO" are displayed. "All" contains all register types, and you can select according to your own needs, as shown in Figure 1-59.



Name	Value
Regs	
r0	0xF0
r1	0x00
r2	0x00
r3	0x00
r4	0x71
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x85
dptr	0xDD
PC $ (PC)	0xA25
psw	0x00
p	0x0
f1	0x0
ov	0x0
rs	0x0
f0	0x0
ac	0x0
cy	0x0

Figure 1-59

Regs: There is a set of current working registers r0~r7 that display the values in r0~r7 when the current assembly code runs.

Sys register: Includes: “a” accumulator; “b”; “sp” stack pointer, which is used to point to the top of the current stack; “dptr” data pointer, which is used as the address register of off-chip RAM addressing; “PC” program pointer, which indicates where the current program is running, pointing to the code area.

Psw program status word: including “p”, “f1”, “ov”, “rs”, “f0”, “ac”, “cy”.

“p” parity flag bit: every time an assembly instruction is executed, the chip sets or resets “p” according to the parity number of 1 in “a”. Odd is 1 and even is 0.

“Ov” is the overflow flag. When adding and subtracting signed numbers, the hardware sets or resets it; When OV=1, it means that a number has exceeded the range that the accumulator represents a signed number in the form of complement, that is, it exceeds the range of - 128~+127.

“rs” selects bits for the working register group, which is set or cleared during programming.

“f0” is a software flag bit, a user-defined status flag, which is set or reset by software.

“ac” is the auxiliary carry flag bit. When performing an 8-bit addition operation, if the highest bit (D3) of the lower half byte has a carry, then “ac”=1, otherwise “ac”=0; When performing 8-bit subtraction operation, if D3 has borrow, “ac”=1, otherwise “ac”=0.

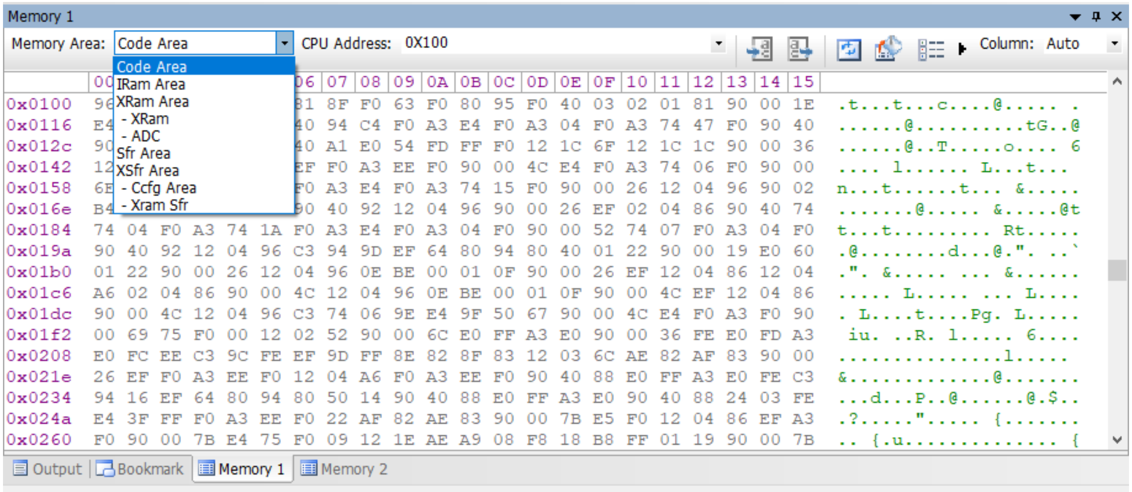
“cy” is the carry flag bit. When using operations such as addition, subtraction, multiplication and division, left shift, and right shift, “cy” is affected. When the highest bit (D7) of the data is added to generate carry, “cy”=1, otherwise “cy”=0; When performing such a subtraction operation, if the operation result has a debit, “cy”=1, otherwise “cy”=0.

As shown in Figure 1-59, click to copy the selected register and register value; Click the

“Edit Value” option to edit the selected register value; Click hex to display the register value in hexadecimal format; Click 0~1 to display the register value in binary form; Click ±int to display the register value in the form of signed integer; Click int to display the register value in the form of unsigned integer; Click flt to display the register value as a single-precision floating-point number.

1.4.7 Memory Window

Memory Window: In the memory window, you can view the data in different memory spaces such as Code, IRam, XRam, CCFG, and enter the queried address in the CPU Address to reach the address, as shown in Figure 1-60.

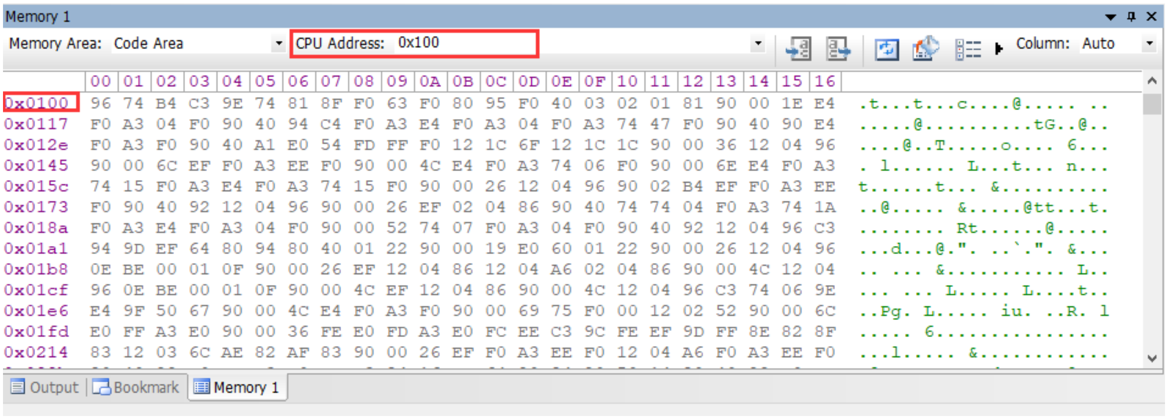


Memory Area: Code Area   CPU Address: 0X100   Column: Auto

```
        00                06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15
0x0100  96                81 8F F0 63 F0 80 95 F0 40 03 02 01 81 90 00 1E  .t...t...c....@..... .
0x0116  E4                40 94 C4 F0 A3 E4 F0 A3 04 F0 A3 74 47 F0 90 40  ......@..........tG..@
0x012c  90                40 A1 E0 54 FD FF F0 12 1C 6F 12 1C 1C 90 00 36  ......@..T.....o.... 6
0x0142  12                EF F0 A3 EE F0 90 00 4C E4 F0 A3 74 06 F0 90 00  .... l...... L...t...
0x0158  6E                F0 A3 E4 F0 A3 74 15 F0 90 00 26 12 04 96 90 02  n...t......t... &.....
0x016e  B4                90 40 92 12 04 96 90 00 26 EF 02 04 86 90 40 74  .......@..... &.....@t
0x0184  74 04 F0 A3 74 1A F0 A3 E4 F0 A3 04 F0 90 00 52 74 07 F0 A3 04 F0  t...t......... Rt.....
0x019a  90 40 92 12 04 96 C3 94 9D EF 64 80 94 80 40 01 22 90 00 19 E0 60  .@........d...@.". ..`
0x01b0  01 22 90 00 26 12 04 96 0E BE 00 01 0F 90 00 26 EF 12 04 86 12 04  .". &..... ... &......
0x01c6  A6 02 04 86 90 00 4C 12 04 96 0E BE 00 01 0F 90 00 4C EF 12 04 86  ..... L..... ... L....
0x01dc  90 00 4C 12 04 96 C3 74 06 9E E4 9F 50 67 90 00 4C E4 F0 A3 F0 90  . L....t....Pg. L.....
0x01f2  00 69 75 F0 00 12 02 52 90 00 6C E0 FF A3 E0 90 00 36 FE E0 FD A3   iu. ..R. l..... 6....
0x0208  E0 FC EE C3 9C FE EF 9D FF 8E 82 8F 83 12 03 6C AE 82 AF 83 90 00  ...............l.....
0x021e  26 EF F0 A3 EE F0 12 04 A6 F0 A3 EE F0 90 40 88 E0 FF A3 E0 FE C3  &.............@.......
0x0234  94 16 EF 64 80 94 80 50 14 90 40 88 E0 FF A3 E0 90 40 88 24 03 FE  ...d...P..@......@.$..
0x024a  E4 3F FF F0 A3 EE F0 22 AF 82 AE 83 90 00 7B E5 F0 12 04 86 EF A3  .?....."..... {.......
0x0260  F0 90 00 7B E4 75 F0 09 12 1E AE A9 08 F8 18 B8 FF 01 19 90 00 7B  .. {.u.............. {
```

Figure 1-60

As shown in Figure 1-61, enter 0x100 in the CPU Address and press Enter. The memory window reaches the address of 0x100 to facilitate the programmer to view the address data.

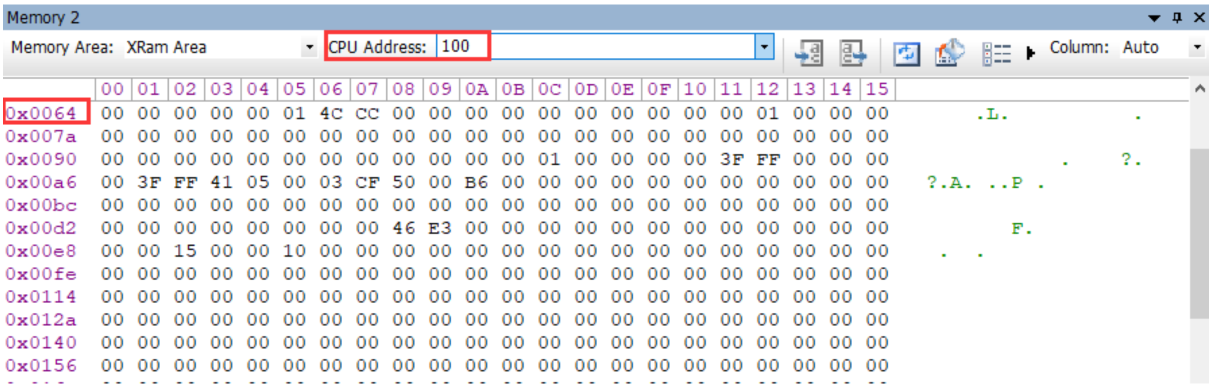


Memory Area: Code Area   CPU Address: 0x100   Column: Auto

```
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16
0x0100  96 74 B4 C3 9E 74 81 8F F0 63 F0 80 95 F0 40 03 02 01 81 90 00 1E E4  .t...t...c....@..... ..
0x0117  F0 A3 04 F0 90 40 94 C4 F0 A3 E4 F0 A3 04 F0 A3 74 47 F0 90 40 90 E4  .....@..........tG..@..
0x012e  F0 A3 F0 90 40 A1 E0 54 FD FF F0 12 1C 6F 12 1C 1C 90 00 36 12 04 96  ....@..T.....o.... 6...
0x0145  90 00 6C EF F0 A3 EE F0 90 00 4C E4 F0 A3 74 06 F0 90 00 6E E4 F0 A3  . l...... L...t... n...
0x015c  74 15 F0 A3 E4 F0 A3 74 15 F0 90 00 26 12 04 96 90 02 B4 EF F0 A3 EE  t......t... &..........
0x0173  F0 90 40 92 12 04 96 90 00 26 EF 02 04 86 90 40 74 74 04 F0 A3 74 1A  ..@..... &.....@tt...t.
0x018a  F0 A3 E4 F0 A3 04 F0 90 00 52 74 07 F0 A3 04 F0 90 40 92 12 04 96 C3  ......... Rt......@.....
0x01a1  94 9D EF 64 80 94 80 40 01 22 90 00 19 E0 60 01 22 90 00 26 12 04 96  ...d...@.". ..`.". &...
0x01b8  0E BE 00 01 0F 90 00 26 EF 12 04 86 12 04 A6 02 04 86 90 00 4C 12 04  .. ... &........... L..
0x01cf  96 0E BE 00 01 0F 90 00 4C EF 12 04 86 90 00 4C 12 04 96 C3 74 06 9E  ... ... L..... L....t..
0x01e6  E4 9F 50 67 90 00 4C E4 F0 A3 F0 90 00 69 75 F0 00 12 02 52 90 00 6C  ..Pg. L..... iu. ..R. l
0x01fd  E0 FF A3 E0 90 00 36 FE E0 FD A3 E0 FC EE C3 9C FE EF 9D FF 8E 82 8F  ..... 6.................
0x0214  83 12 03 6C AE 82 AF 83 90 00 26 EF F0 A3 EE F0 12 04 A6 F0 A3 EE F0  ...l..... &............
```

Figure 1-61

Similarly, entering the decimal number 100 (0x64) in CPU Address reaches its corresponding hexadecimal address, as shown in Figure 1-62.

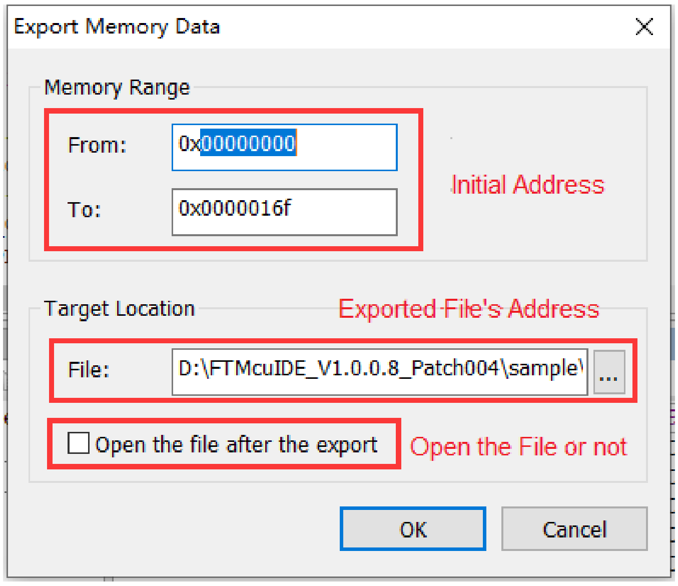


Memory Area: XRam Area
CPU Address: 100

```
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15
0x0064  00 00 00 00 00 01 4C CC 00 00 00 00 00 00 00 00 00 00 01 00 00 00          .L.         .
0x007a  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0090  00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 3F FF 00 00 00                 .    ?.
0x00a6  00 3F FF 41 05 00 03 CF 50 00 B6 00 00 00 00 00 00 00 00 00 00 00   ?.A. ..P .
0x00bc  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00d2  00 00 00 00 00 00 00 00 46 E3 00 00 00 00 00 00 00 00 00 00 00 00           F.
0x00e8  00 00 15 00 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     .  .
0x00fe  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0114  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x012a  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0156  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Figure 1-62

Export Memory Data: Export the data of the currently selected memory space, as shown in Figure 1-63. First, enter the start and end address of the exported memory space, then select the address of the exported file, and then select whether to open the file after export. If selected, open the file after the end of the exported file, and finally click OK to finish exporting the memory data.



Memory Range

From: 0x00000000

To: 0x0000016f

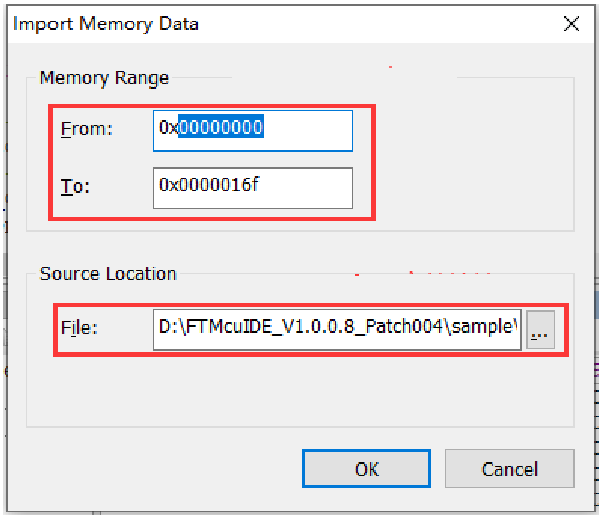
Target Location

File: D:\FTMcuIDE_V1.0.0.8_Patch004\sample\

☐ Open the file after the export

Figure 1-63

Import Memory Data: Import the memory data file, as shown in Figure 1-64. First, enter the start and end address of the imported memory space, then select the imported memory file, and click OK to import the memory file data into the set start and end address range.



Import Memory Data

Memory Range

From: 0x00000000

To: 0x0000016f

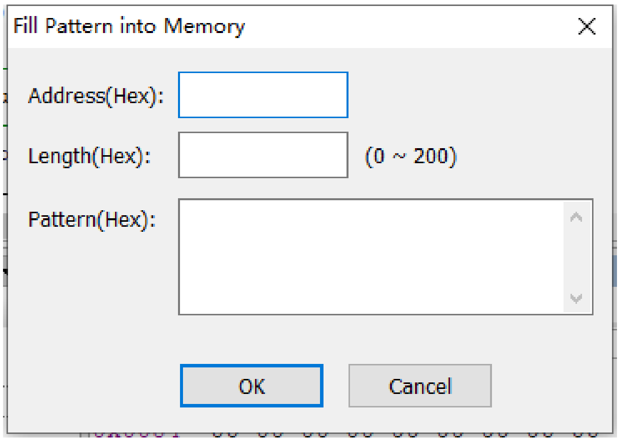
Source Location

File: D:\FTMcuIDE_V1.0.0.8_Patch004\sample\

Figure 1-64

: Refresh Memory.

: Fill Pattern, as shown in Figure 1-65.



Fill Pattern into Memory

Address(Hex):

Length(Hex): (0 ~ 200)

Pattern(Hex):

Figure 1-65

: Set the style of memory window, as shown in Figure 1-66, including a column composed of several bytes, data format, description and column layout format. The data format is displayed in five ways: hexadecimal, binary, signed integer, unsigned integer and single-precision floating-point type. For details, please see 1.4.6 Registers Window.

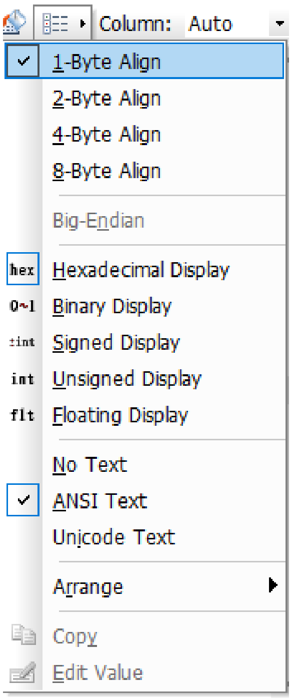
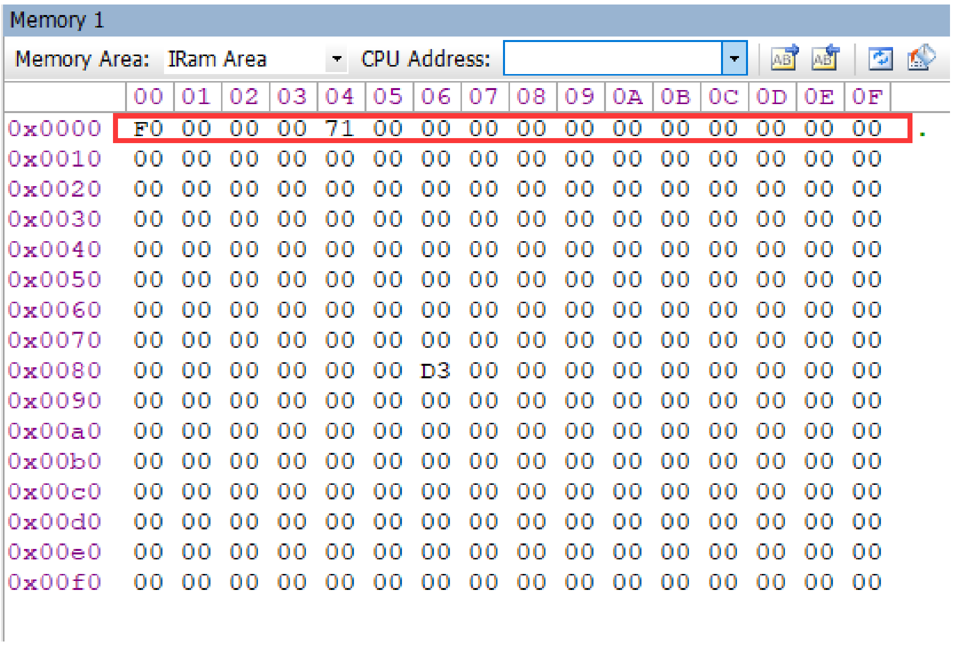


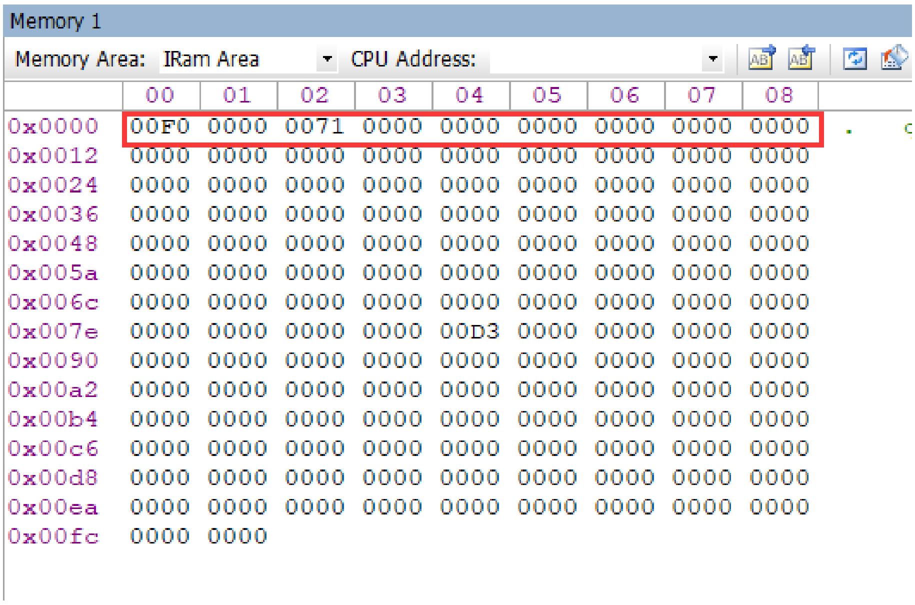
Figure 1-66

When 1-Byte Align is selected, the memory window is shown in Figure 1-67; When 2-Byte Align is selected, the memory window is shown in Figure 1-68. Similarly, you can know the style of the window when other selections are made.



```
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0x0000  F0 00 00 00 71 00 00 00 00 00 00 00 00 00 00 00  .
0x0010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0080  00 00 00 00 00 00 D3 00 00 00 00 00 00 00 00 00
0x0090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00a0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00b0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00c0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00d0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00e0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00f0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

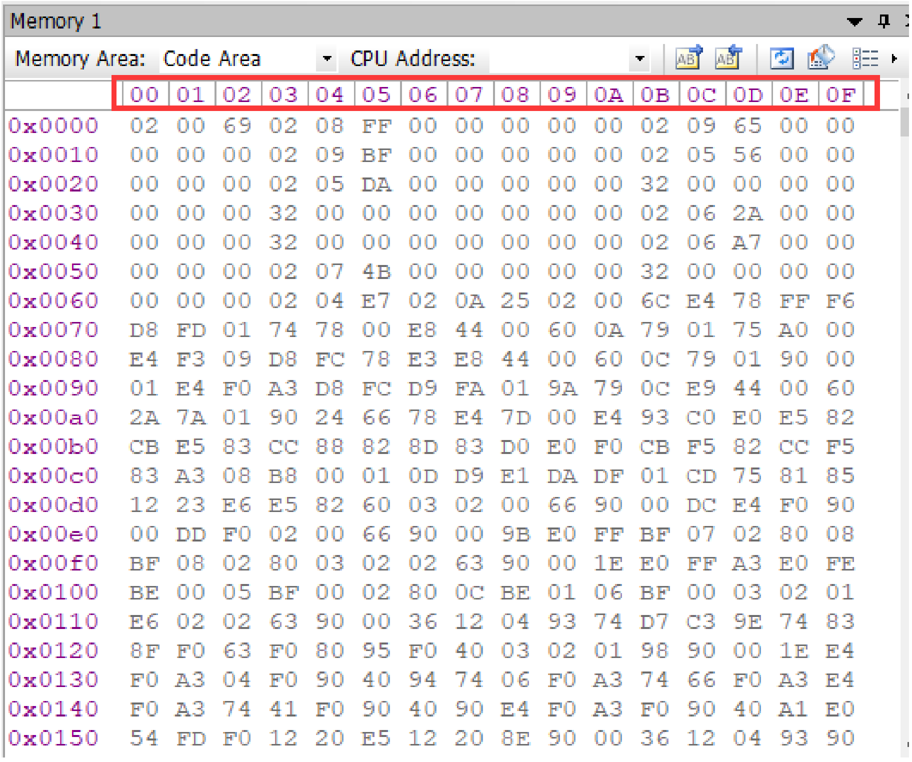
Figure 1-67



```
        00   01   02   03   04   05   06   07   08
0x0000  00F0 0000 0071 0000 0000 0000 0000 0000 0000
0x0012  0000 0000 0000 0000 0000 0000 0000 0000 0000
0x0024  0000 0000 0000 0000 0000 0000 0000 0000 0000
0x0036  0000 0000 0000 0000 0000 0000 0000 0000 0000
0x0048  0000 0000 0000 0000 0000 0000 0000 0000 0000
0x005a  0000 0000 0000 0000 0000 0000 0000 0000 0000
0x006c  0000 0000 0000 0000 0000 0000 0000 0000 0000
0x007e  0000 0000 0000 0000 00D3 0000 0000 0000 0000
0x0090  0000 0000 0000 0000 0000 0000 0000 0000 0000
0x00a2  0000 0000 0000 0000 0000 0000 0000 0000 0000
0x00b4  0000 0000 0000 0000 0000 0000 0000 0000 0000
0x00c6  0000 0000 0000 0000 0000 0000 0000 0000 0000
0x00d8  0000 0000 0000 0000 0000 0000 0000 0000 0000
0x00ea  0000 0000 0000 0000 0000 0000 0000 0000 0000
0x00fc  0000 0000
```

Figure 1-68

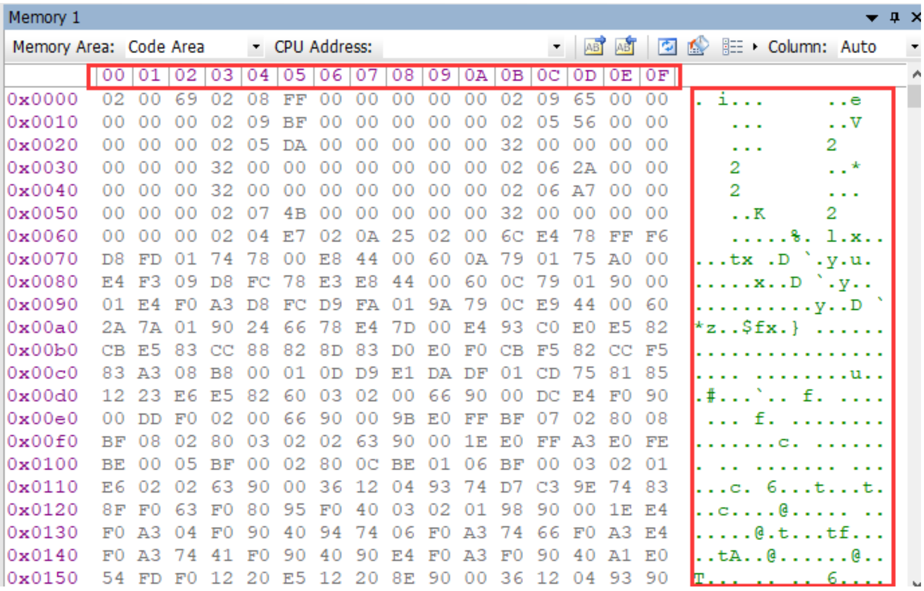
When No Text is selected, the memory window is displayed in hexadecimal 16-bit data format without description, as shown in Figure 1-69.



```
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0x0000  02 00 69 02 08 FF 00 00 00 00 00 02 09 65 00 00
0x0010  00 00 00 02 09 BF 00 00 00 00 00 02 05 56 00 00
0x0020  00 00 00 02 05 DA 00 00 00 00 00 32 00 00 00 00
0x0030  00 00 00 32 00 00 00 00 00 00 00 02 06 2A 00 00
0x0040  00 00 00 32 00 00 00 00 00 00 00 02 06 A7 00 00
0x0050  00 00 00 02 07 4B 00 00 00 00 00 32 00 00 00 00
0x0060  00 00 00 02 04 E7 02 0A 25 02 00 6C E4 78 FF F6
0x0070  D8 FD 01 74 78 00 E8 44 00 60 0A 79 01 75 A0 00
0x0080  E4 F3 09 D8 FC 78 E3 E8 44 00 60 0C 79 01 90 00
0x0090  01 E4 F0 A3 D8 FC D9 FA 01 9A 79 0C E9 44 00 60
0x00a0  2A 7A 01 90 24 66 78 E4 7D 00 E4 93 C0 E0 E5 82
0x00b0  CB E5 83 CC 88 82 8D 83 D0 E0 F0 CB F5 82 CC F5
0x00c0  83 A3 08 B8 00 01 0D D9 E1 DA DF 01 CD 75 81 85
0x00d0  12 23 E6 E5 82 60 03 02 00 66 90 00 DC E4 F0 90
0x00e0  00 DD F0 02 00 66 90 00 9B E0 FF BF 07 02 80 08
0x00f0  BF 08 02 80 03 02 02 63 90 00 1E E0 FF A3 E0 FE
0x0100  BE 00 05 BF 00 02 80 0C BE 01 06 BF 00 03 02 01
0x0110  E6 02 02 63 90 00 36 12 04 93 74 D7 C3 9E 74 83
0x0120  8F F0 63 F0 80 95 F0 40 03 02 01 98 90 00 1E E4
0x0130  F0 A3 04 F0 90 40 94 74 06 F0 A3 74 66 F0 A3 E4
0x0140  F0 A3 74 41 F0 90 40 90 E4 F0 A3 F0 90 40 A1 E0
0x0150  54 FD F0 12 20 E5 12 20 8E 90 00 36 12 04 93 90
```

Figure 1-69

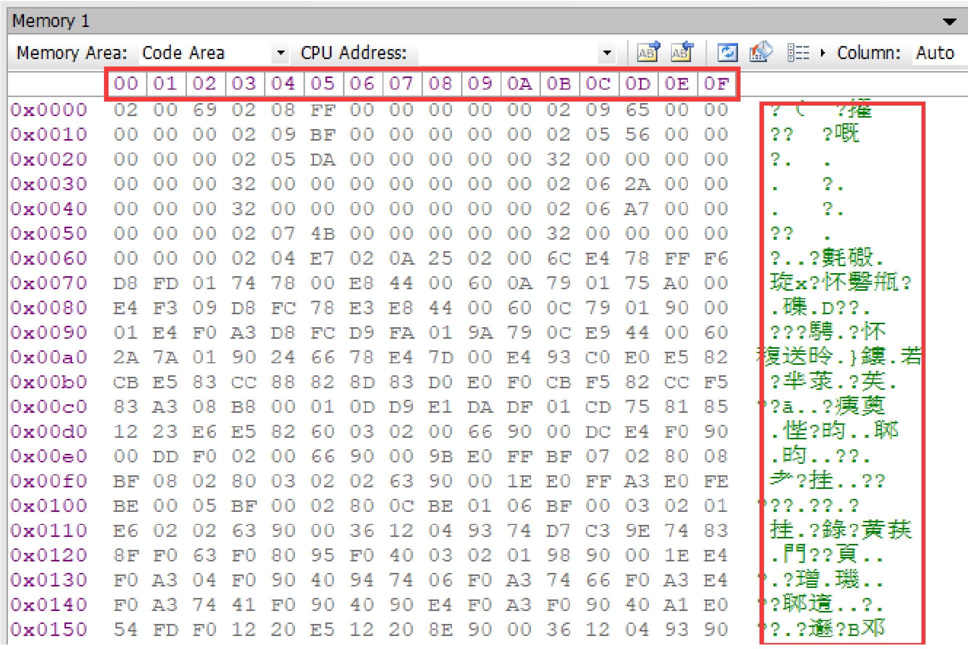
When ANSI Text is selected, the memory window is displayed in hexadecimal 16-bit data format with corresponding English description beside it, as shown in Figure 1-70;



```
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0x0000  02 00 69 02 08 FF 00 00 00 00 00 02 09 65 00 00   . i...       ..e
0x0010  00 00 00 02 09 BF 00 00 00 00 00 02 05 56 00 00     ...        ..V
0x0020  00 00 00 02 05 DA 00 00 00 00 00 32 00 00 00 00     ...        2
0x0030  00 00 00 32 00 00 00 00 00 00 00 02 06 2A 00 00     2          ..*
0x0040  00 00 00 32 00 00 00 00 00 00 00 02 06 A7 00 00     2          ...
0x0050  00 00 00 02 07 4B 00 00 00 00 00 32 00 00 00 00     ..K        2
0x0060  00 00 00 02 04 E7 02 0A 25 02 00 6C E4 78 FF F6     .....%. l.x..
0x0070  D8 FD 01 74 78 00 E8 44 00 60 0A 79 01 75 A0 00   ...tx .D `.y.u.
0x0080  E4 F3 09 D8 FC 78 E3 E8 44 00 60 0C 79 01 90 00   .....x..D `.y..
0x0090  01 E4 F0 A3 D8 FC D9 FA 01 9A 79 0C E9 44 00 60   ..........y..D `
0x00a0  2A 7A 01 90 24 66 78 E4 7D 00 E4 93 C0 E0 E5 82   *z..$fx.} ......
0x00b0  CB E5 83 CC 88 82 8D 83 D0 E0 F0 CB F5 82 CC F5   ................
0x00c0  83 A3 08 B8 00 01 0D D9 E1 DA DF 01 CD 75 81 85   .... ........u..
0x00d0  12 23 E6 E5 82 60 03 02 00 66 90 00 DC E4 F0 90   .#...`.. f. ....
0x00e0  00 DD F0 02 00 66 90 00 9B E0 FF BF 07 02 80 08    ... f. ........
0x00f0  BF 08 02 80 03 02 02 63 90 00 1E E0 FF A3 E0 FE   .......c. ......
0x0100  BE 00 05 BF 00 02 80 0C BE 01 06 BF 00 03 02 01   . .. ....... ...
0x0110  E6 02 02 63 90 00 36 12 04 93 74 D7 C3 9E 74 83   ...c. 6...t...t.
0x0120  8F F0 63 F0 80 95 F0 40 03 02 01 98 90 00 1E E4   ..c....@..... ..
0x0130  F0 A3 04 F0 90 40 94 74 06 F0 A3 74 66 F0 A3 E4   .....@.t...tf...
0x0140  F0 A3 74 41 F0 90 40 90 E4 F0 A3 F0 90 40 A1 E0   ..tA..@......@..
0x0150  54 FD F0 12 20 E5 12 20 8E 90 00 36 12 04 93 90   T... .. .. 6....
```

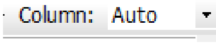
Figure 1-70

When Unicode Text is selected, the memory window is displayed in hexadecimal 16-bit data format with corresponding text description beside it, as shown in Figure 1-71.

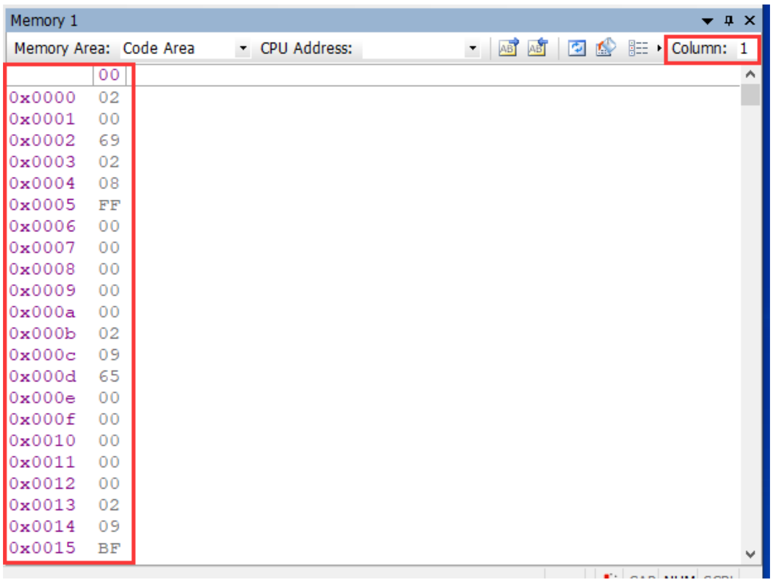


```
        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0x0000  02 00 69 02 08 FF 00 00 00 00 00 02 09 65 00 00   ? (   ?攉
0x0010  00 00 00 02 09 BF 00 00 00 00 00 02 05 56 00 00   ??   ?嘅
0x0020  00 00 00 02 05 DA 00 00 00 00 00 32 00 00 00 00   ?.   .
0x0030  00 00 00 32 00 00 00 00 00 00 00 02 06 2A 00 00   .    ?.
0x0040  00 00 00 32 00 00 00 00 00 00 00 02 06 A7 00 00   .    ?.
0x0050  00 00 00 02 07 4B 00 00 00 00 00 32 00 00 00 00   ??   .
0x0060  00 00 00 02 04 E7 02 0A 25 02 00 6C E4 78 FF F6   ?..?毴礈.
0x0070  D8 FD 01 74 78 00 E8 44 00 60 0A 79 01 75 A0 00   琁x?怀磬瓶?
0x0080  E4 F3 09 D8 FC 78 E3 E8 44 00 60 0C 79 01 90 00   .磦.D??.
0x0090  01 E4 F0 A3 D8 FC D9 FA 01 9A 79 0C E9 44 00 60   ???騁.?怀
0x00a0  2A 7A 01 90 24 66 78 E4 7D 00 E4 93 C0 E0 E5 82   複送昤.}鏤.若
0x00b0  CB E5 83 CC 88 82 8D 83 D0 E0 F0 CB F5 82 CC F5   ?芈菉.?芺.
0x00c0  83 A3 08 B8 00 01 0D D9 E1 DA DF 01 CD 75 81 85   ??a..?瘨蓃
0x00d0  12 23 E6 E5 82 60 03 02 00 66 90 00 DC E4 F0 90   .惿?昀..郕
0x00e0  00 DD F0 02 00 66 90 00 9B E0 FF BF 07 02 80 08   .昀..??.
0x00f0  BF 08 02 80 03 02 02 63 90 00 1E E0 FF A3 E0 FE   耂?挂..??
0x0100  BE 00 05 BF 00 02 80 0C BE 01 06 BF 00 03 02 01   ???.??.?
0x0110  E6 02 02 63 90 00 36 12 04 93 74 D7 C3 9E 74 83   挂.?錄?黃扶
0x0120  8F F0 63 F0 80 95 F0 40 03 02 01 98 90 00 1E E4   .門??頁..
0x0130  F0 A3 04 F0 90 40 94 74 06 F0 A3 74 66 F0 A3 E4   ?.?璔.璣..
0x0140  F0 A3 74 41 F0 90 40 90 E4 F0 A3 F0 90 40 A1 E0   ??郕遺..?.
0x0150  54 FD F0 12 20 E5 12 20 8E 90 00 36 12 04 93 90   ??.?遾?B邓
```

Figure 1-71



: Set the number of rows displayed in the memory window data. The options are Auto, 1, 2, 4, 8, 16, 32 and 64. When Auto is selected, the memory window adjusts the displayed data according to the size of the window; When 1 is selected, it indicates that the memory window data is displayed in 1 column, as shown in Figure 1-72; Similarly, other selections indicate that the memory window data is displayed in the selected number of columns.



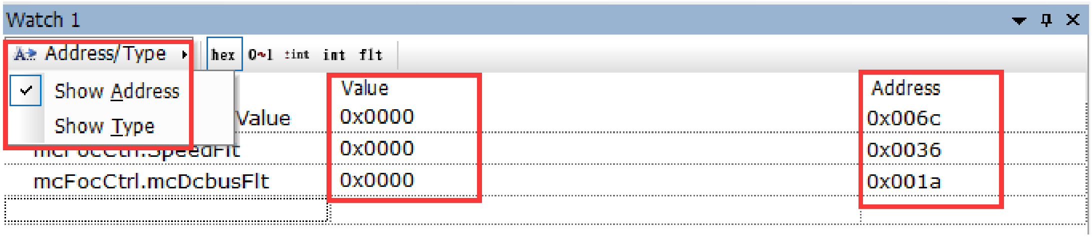
```
        00
0x0000  02
0x0001  00
0x0002  69
0x0003  02
0x0004  08
0x0005  FF
0x0006  00
0x0007  00
0x0008  00
0x0009  00
0x000a  00
0x000b  02
0x000c  09
0x000d  65
0x000e  00
0x000f  00
0x0010  00
0x0011  00
0x0012  00
0x0013  02
0x0014  09
0x0015  BF
```

Figure 1-72

1.4.8 Watch Window

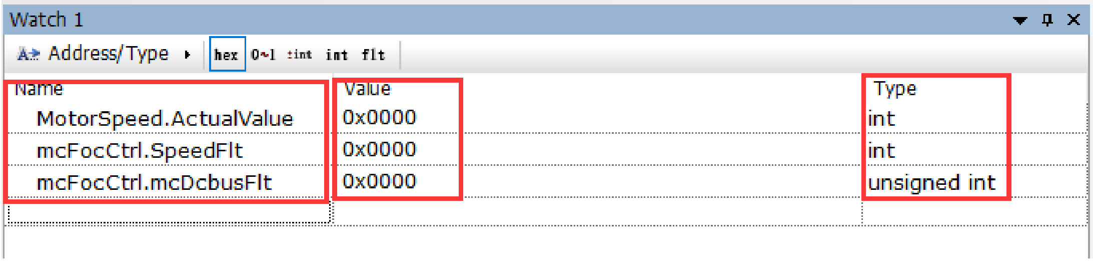
The FTM8ForgeIDE allows programmers to view the values of variables or expressions in source code, enabling you to track the changes of their values in real time during operation. You can view variables in many ways. For example, in the source code window, place the mouse over the variable you want to view, or view it by opening the Watch window.

Observe the change of variable value when the program is running, and you can enter the variable and query the variable value by yourself. The Watch window has the option of displaying variable address and variable type. When the variable address is selected, it is shown in Figure 1-73. When the display variable type is selected, as shown in Figure 1-74.



	Value	Address
	0x0000	0x006c
mcFocCtrl.SpeedFlt	0x0000	0x0036
mcFocCtrl.mcDcbusFlt	0x0000	0x001a

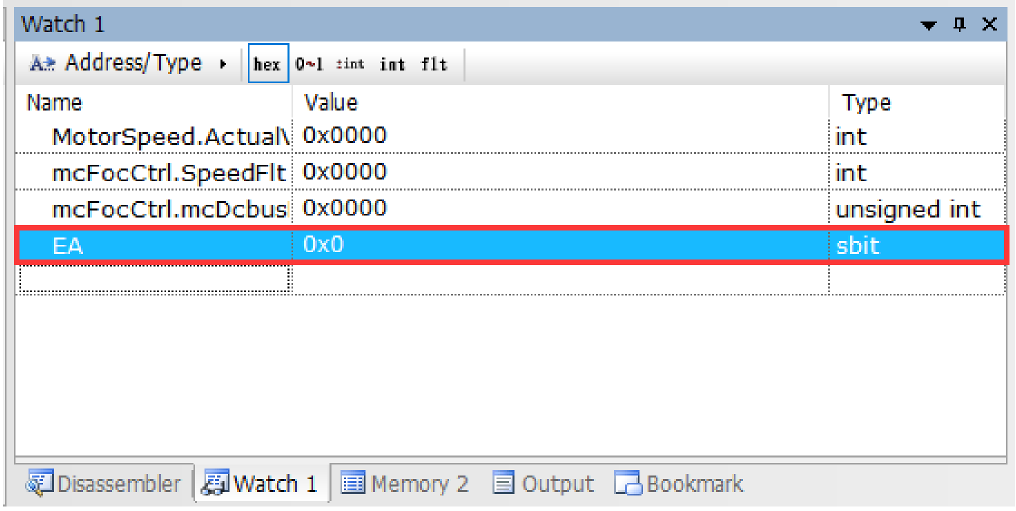
Figure 1-73



Name	Value	Type
MotorSpeed.ActualValue	0x0000	int
mcFocCtrl.SpeedFlt	0x0000	int
mcFocCtrl.mcDcbusFlt	0x0000	unsigned int

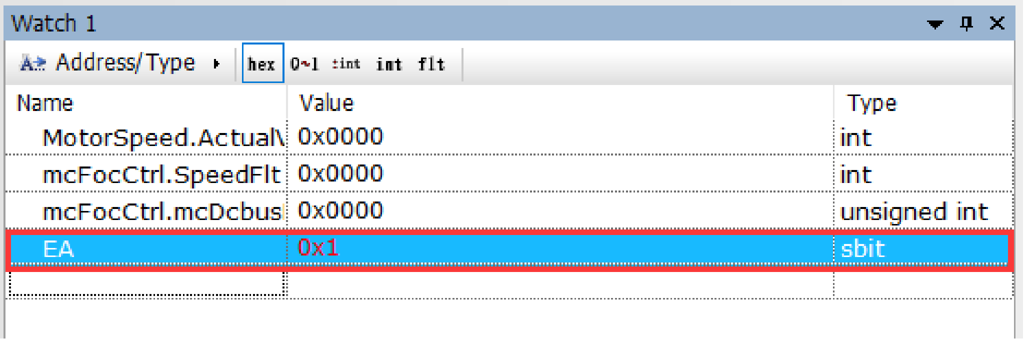
Figure 1-74

When running the program, set a monitoring point on the variable EA: click the dotted rectangle in the Watch window. When the input area appears, type EA and press Enter. The window immediately displays the current EA value 0x0 and type sbit, as shown in Figure 1-75. Continue to run the program. When the variable value monitored in the Watch window changes, the color of the variable value changes to red to indicate the change. As shown in Figure 1-76, the value of EA changes to 0x1. If you want to delete a variable in the Watch window, select it, and then right-click to select Delete.



Name	Value	Type
MotorSpeed.Actual\	0x0000	int
mcFocCtrl.SpeedFlt	0x0000	int
mcFocCtrl.mcDcbus	0x0000	unsigned int
EA	0x0	sbit

Figure 1-75



Name	Value	Type
MotorSpeed.Actual\	0x0000	int
mcFocCtrl.SpeedFlt	0x0000	int
mcFocCtrl.mcDcbus	0x0000	unsigned int
EA	0x1	sbit

Figure 1-76

Variable values are displayed in five ways: hexadecimal, binary, signed integer, unsigned integer and single-precision floating-point type. For details, see 1.4.6 Registers Window.

1.4.9 Command Window

Command Window: Support command line input.

2 Create New Application

2.1 Create A New Solution

Select "File" →"New" →"Blank Solution" in the upper left corner, as shown in Figure 2-1.

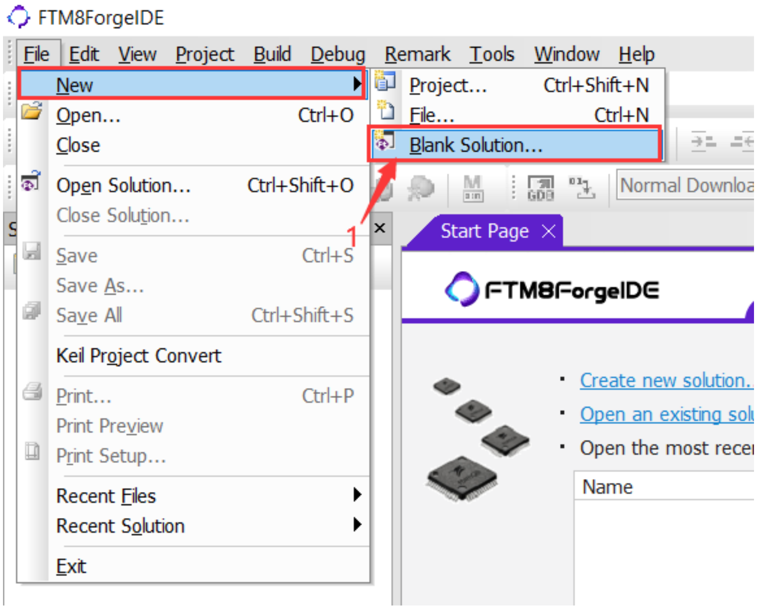
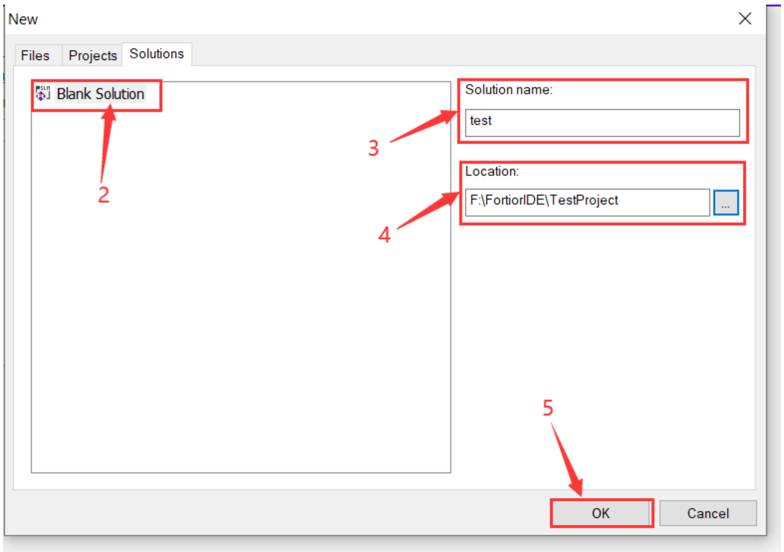


Figure 2-1

Select "Blank Solution", enter solution name, and select storage path of the solution from Location, as shown in Figure 2-2.



Solution name: test

Location: F:\FortiorIDE\TestProject

Figure 2-2

Click "OK". The creation result is shown in Figure 2-3. You can create or add projects and files in Solution "test".

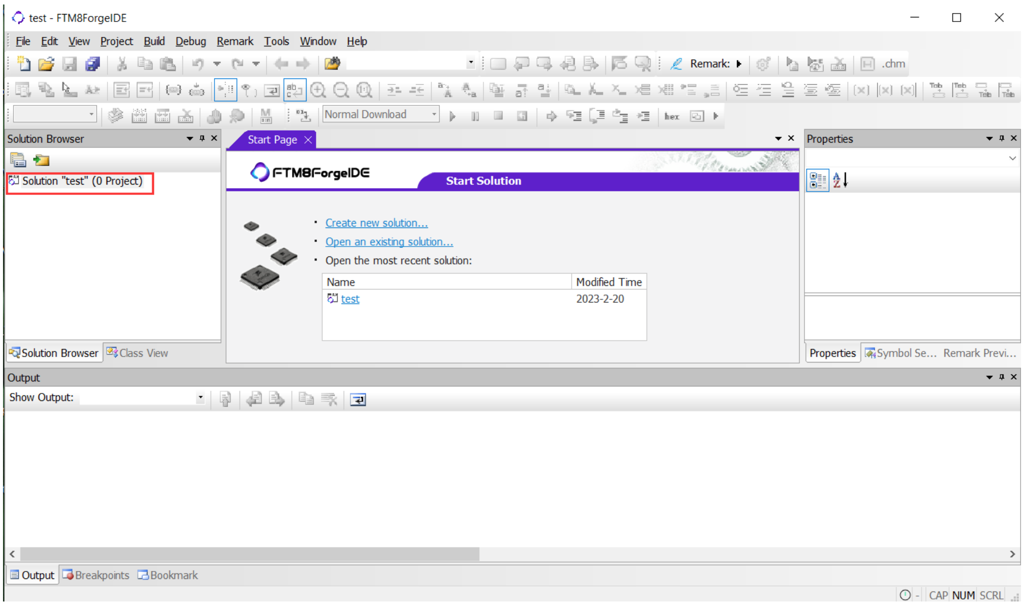


Figure 2-3

2.2 Create A New Project

Step 1: Select "File" →"New" →"Project" in the upper left corner, as shown in Figure 2-4.

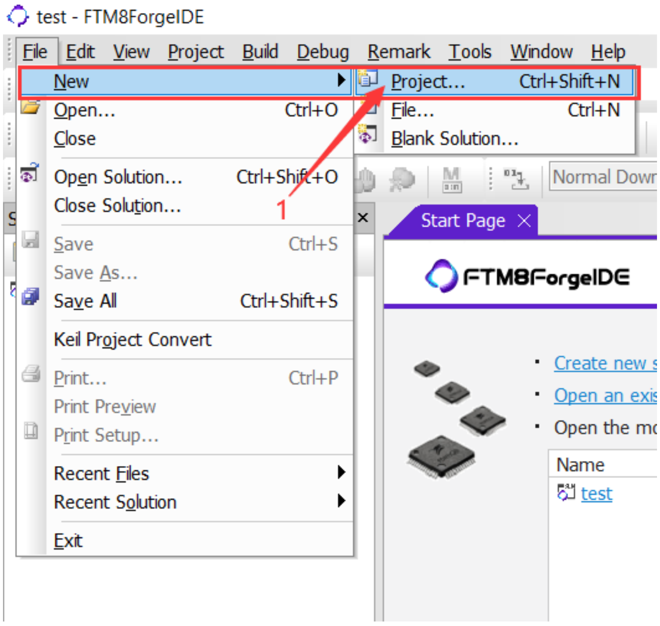


Figure 2-4

Step 2: Select the project type, including MCU8 Project, MCU8 Library Project, MCU8 Assembly Project, MCU8 Simple Project, and MCU8 Bootloader Project, as shown in Figure 2-5.

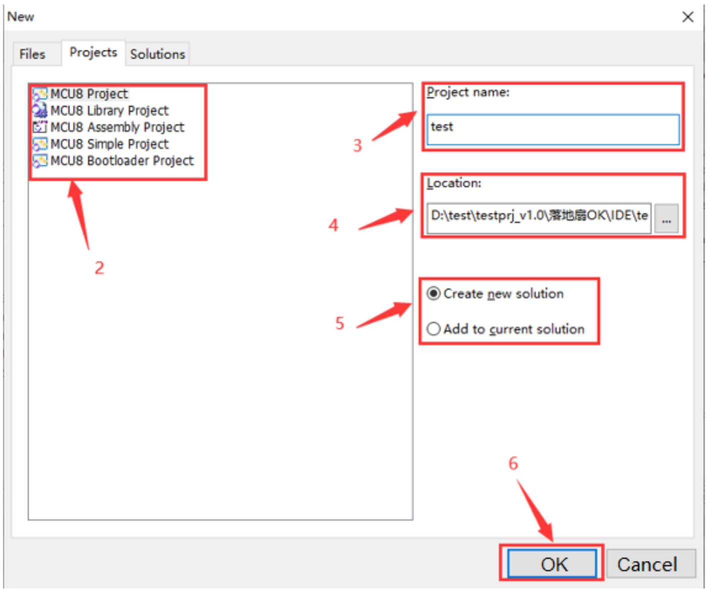
The MCU8 Library Project is a library project that generates a ". lib" library file. MCU8 Assembly Project is an assembly project that generates a ". hex" file. MCU8 Bootloader Project creates a Bootloader project. MCU8 Project is a complete C project. MCU8 Simple Project is a simplified version of C project.

Step 3: Enter the project name (for example, "test" or "test.c"), as shown in Figure 2-5. If you do not enter a suffix, the system defaults the new project to be a C source project with the suffix ".c";

Step 4: Select storage path of the new project, as shown in Figure 2-5;

Step 5: Select "Create a new solution" or "Add to the current solution", as shown in Figure 2-5. The former runs the project in a newly created solution, and the latter adds the project to an existing solution;

Step 6: Click "OK" to select the chip type, as shown in Figure 2-5



Project name: test

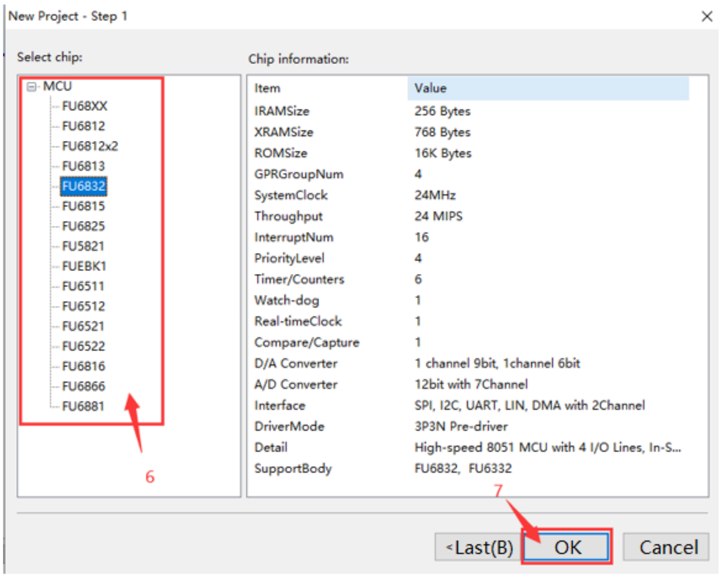
Location: D:\test\testprj_v1.0\落地扇OK\IDE\te

● Create new solution

○ Add to current solution

Figure 2-5

Select the chip model. As shown in Figure 2-6, the chip information is displayed on the right.



Chip information:

Item	Value
IRAMSize	256 Bytes
XRAMSize	768 Bytes
ROMSize	16K Bytes
GPRGroupNum	4
SystemClock	24MHz
Throughput	24 MIPS
InterruptNum	16
PriorityLevel	4
Timer/Counters	6
Watch-dog	1
Real-timeClock	1
Compare/Capture	1
D/A Converter	1 channel 9bit, 1channel 6bit
A/D Converter	12bit with 7Channel
Interface	SPI, I2C, UART, LIN, DMA with 2Channel
DriverMode	3P3N Pre-driver
Detail	High-speed 8051 MCU with 4 I/O Lines, In-S...
SupportBody	FU6832, FU6332

Figure 2-6

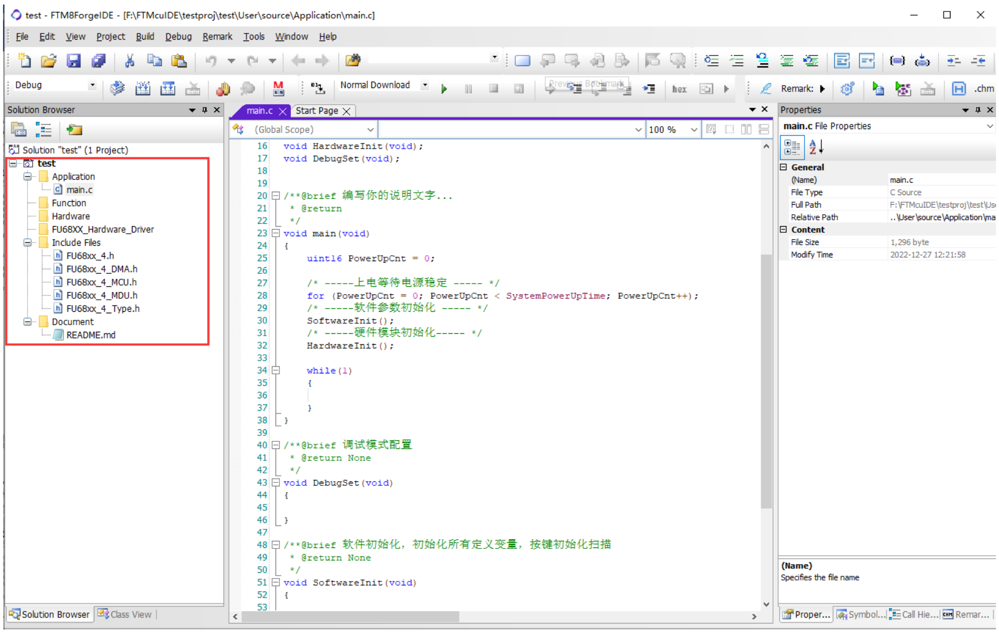
Step 7: Click “Finish” to complete the creation. The creation result is shown in following figure. It contains the source file “main.c” and the header file of the default chip type. The user can write program code in “main.c”.

Step 7: Click "Finish". The creation result is shown in following figure. It contains the source file "main.c" and header file of the selected chip. You can write program code into "main.c" file.

See Figure 2-7 if MCU8 Project and MCU8 Library Project are created;

See Figure 2-8 if MCU8 Assembly Project and MCU8 Simple Project are created;

See Figure 2-9 if MCU8 Bootloader Project is created;



Solution "test" (1 Project)
- test
  - Application
    - main.c
  - Function
  - Hardware
  - FU68XX_Hardware_Driver
  - Include Files
    - FU68xx_4.h
    - FU68xx_4_DMA.h
    - FU68xx_4_MCU.h
    - FU68xx_4_MDU.h
    - FU68xx_4_Type.h
  - Document
    - README.md

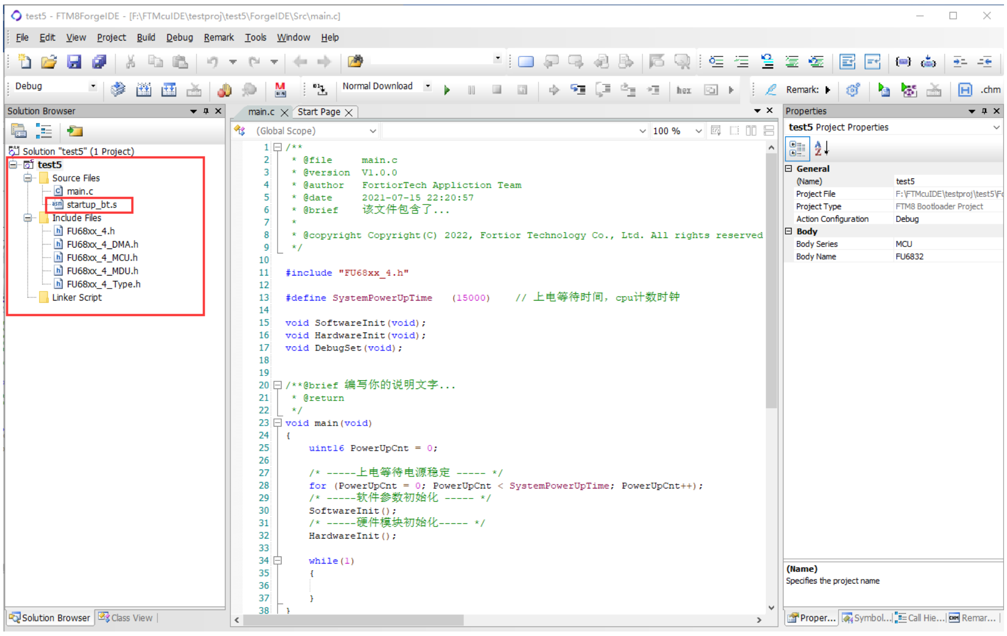
```
void HardwareInit(void);
void DebugSet(void);


/**@brief 编写你的说明文字...
 * @return
 */
void main(void)
{
    uint16 PowerUpCnt = 0;

    /* -----上电等待电源稳定 ----- */
    for (PowerUpCnt = 0; PowerUpCnt < SystemPowerUpTime; PowerUpCnt++);
    /* -----软件参数初始化 ----- */
    SoftwareInit();
    /* -----硬件模块初始化----- */
    HardwareInit();

    while(1)
    {

    }
}

/**@brief 调试模式配置
 * @return None
 */
void DebugSet(void)
{

}

/**@brief 软件初始化，初始化所有定义变量，按键初始化扫描
 * @return None
 */
void SoftwareInit(void)
{
```

main.c File Properties

General	
(Name)	main.c
File Type	C Source
Full Path	F:\FTMcuIDE\testproj\test\Us
Relative Path	..\User\source\Application\ma
Content	
File Size	1,296 byte
Modify Time	2022-12-27 12:21:58

(Name)
Specifies the file name

Figure 2-7



Solution "test5" (1 Project)
- test5
  - Source Files
    - main.c
    - startup_bt.s
  - Include Files
    - FU68xx_4.h
    - FU68xx_4_DMA.h
    - FU68xx_4_MCU.h
    - FU68xx_4_MDU.h
    - FU68xx_4_Type.h
  - Linker Script

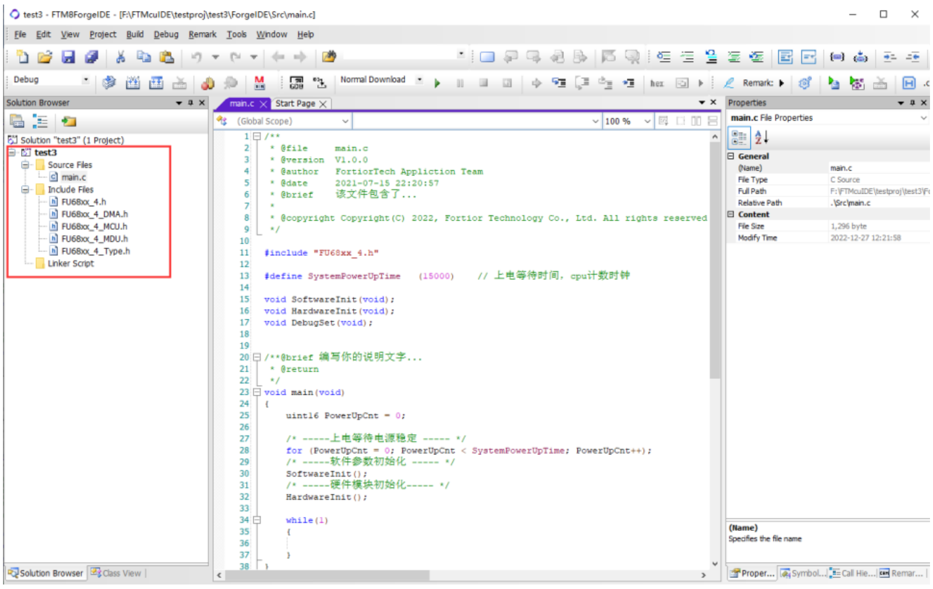
```
/**
 * @file     main.c
 * @version  V1.0.0
 * @author   FortiorTech Appliction Team
 * @date     2021-07-15 22:20:57
 * @brief    该文件包含了...
 *
 * @copyright Copyright(C) 2022, Fortior Technology Co., Ltd. All rights reserved
 */

#include "FU68xx_4.h"

#define SystemPowerUpTime   (15000)    // 上电等待时间，cpu计数时钟

void SoftwareInit(void);
void HardwareInit(void);
void DebugSet(void);


/**@brief 编写你的说明文字...
 * @return
 */
void main(void)
{
    uint16 PowerUpCnt = 0;

    /* -----上电等待电源稳定 ----- */
    for (PowerUpCnt = 0; PowerUpCnt < SystemPowerUpTime; PowerUpCnt++);
    /* -----软件参数初始化 ----- */
    SoftwareInit();
    /* -----硬件模块初始化----- */
    HardwareInit();

    while(1)
    {

    }
}
```

test5 Project Properties

General	
(Name)	test5
Project File	F:\FTMcuIDE\testproj\test5\Fo
Project Type	FTM8 Bootloader Project
Action Configuration	Debug
Body	
Body Series	MCU
Body Name	FU6832

(Name)
Specifies the project name

Figure 2-8



```
/**
 * @file     main.c
 * @version  V1.0.0
 * @author   FortiorTech Appliction Team
 * @date     2021-07-15 22:20:57
 * @brief    该文件包含了...
 *
 * @copyright Copyright(C) 2022, Fortior Technology Co., Ltd. All rights reserved
 */

#include "FU68xx_4.h"

#define SystemPowerUpTime   (15000)    // 上电等待时间，cpu计数时钟

void SoftwareInit(void);
void HardwareInit(void);
void DebugSet(void);


/**@brief 编写你的说明文字...
 * @return
 */
void main(void)
{
    uint16 PowerUpCnt = 0;

    /* -----上电等待电源稳定 ----- */
    for (PowerUpCnt = 0; PowerUpCnt < SystemPowerUpTime; PowerUpCnt++);
    /* -----软件参数初始化 ----- */
    SoftwareInit();
    /* -----硬件模块初始化----- */
    HardwareInit();

    while(1)
    {

    }
}
```

main.c File Properties

General
(Name) main.c
File Type C Source
Full Path F:\FTMcuIDE\testproj\test3\F
Relative Path .\Src\main.c

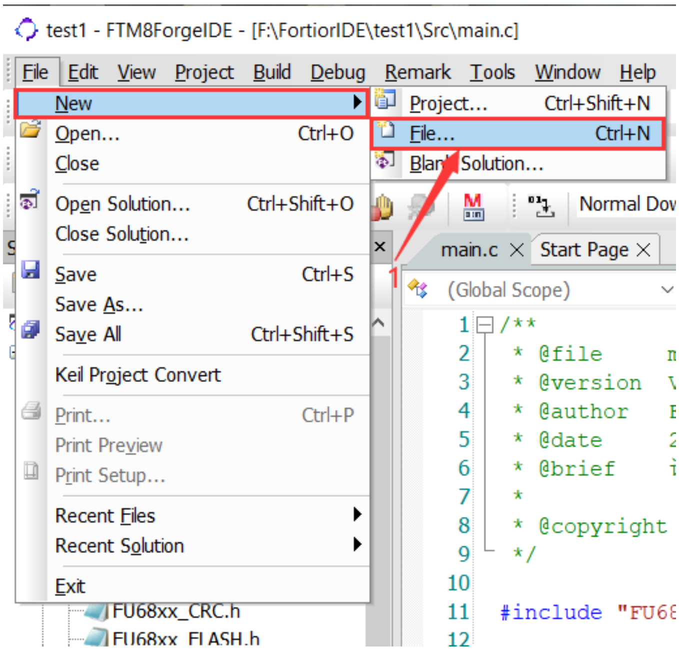
Content
File Size 1,296 byte
Modify Time 2022-12-27 12:21:58

(Name)
Specifies the file name

Figure 2-9

2.3 Create A New File

Step 1: Select "File" →"New" →"File" in the upper left corner, as shown in Figure 2-10.

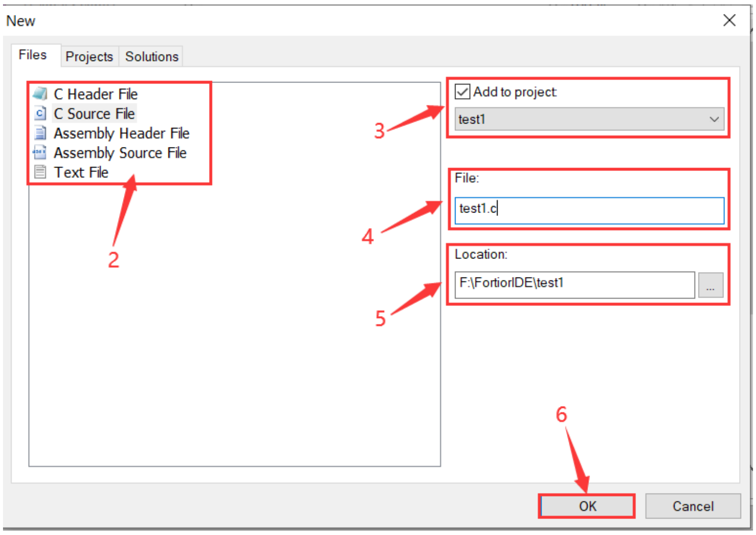


```
/**
 * @file     m
 * @version  V
 * @author   F
 * @date     2
 * @brief    i
 *
 * @copyright
 */

#include "FU68
```

Figure 2-10

Step 2: Select the file type, including "C Header File", "C Source File", "Assembly Header File", "Assembly Source File" and "Text File", as shown in Figure 2-11.

- C Header File: xxx.h;
- C Source File: xxx.c;
- Assembly Header File: xxx.h;
- Assembly Source File: xxx.c;



☑ Add to project: test1

File: test1.c

Location: F:\FortiorIDE\test1

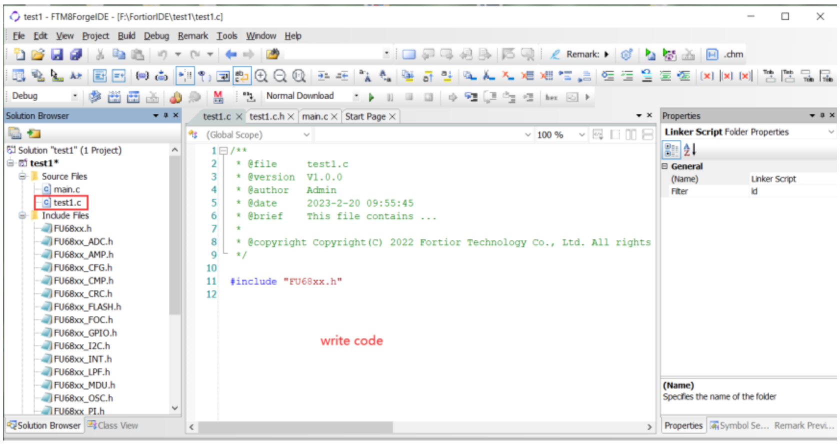
Figure 2-11

Step 3: Check or uncheck "Add to project". If it is selected, the created file is added to the selected project. If not, a separate file is created;

Step 4: Enter the file name, as shown in Figure 2-11. When "C Source File" is selected, you can enter the file name as "test1.c" or "test1". If you enter "test1", the system automatically completes the file suffix according to the selected file type;

Step 5: Select storage path of the new file;

Step 6: Click "OK". The creation result is shown in Figure 2-12. The newly created "test. c" file appears in "Source Files".

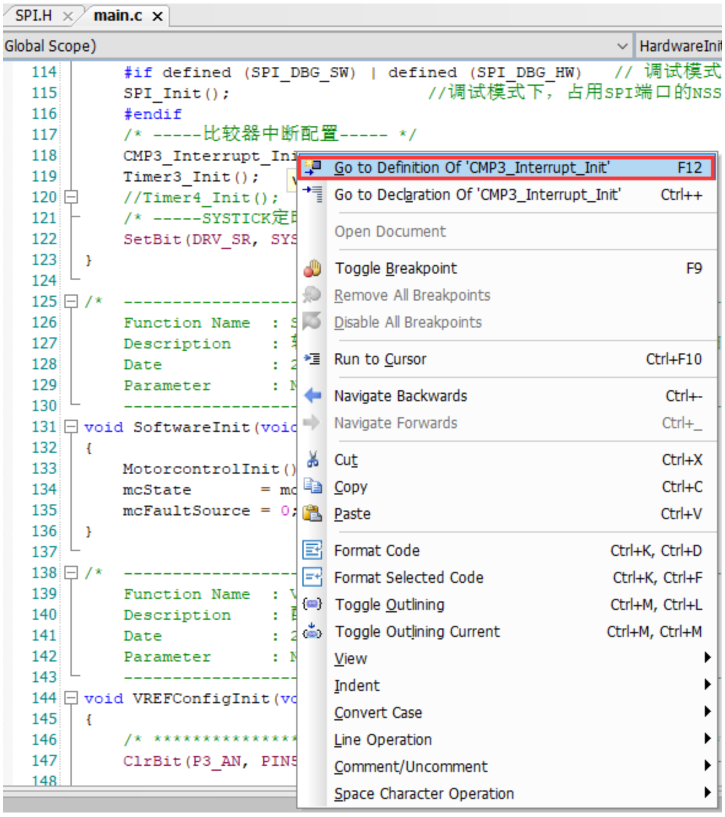


```
/**
 * @file     test1.c
 * @version  V1.0.0
 * @author   Admin
 * @date     2023-2-20 09:55:45
 * @brief    This file contains ...
 *
 * @copyright Copyright(C) 2022 Fortior Technology Co., Ltd. All rights
 */

#include "FU68xx.h"
```

Figure 2-12

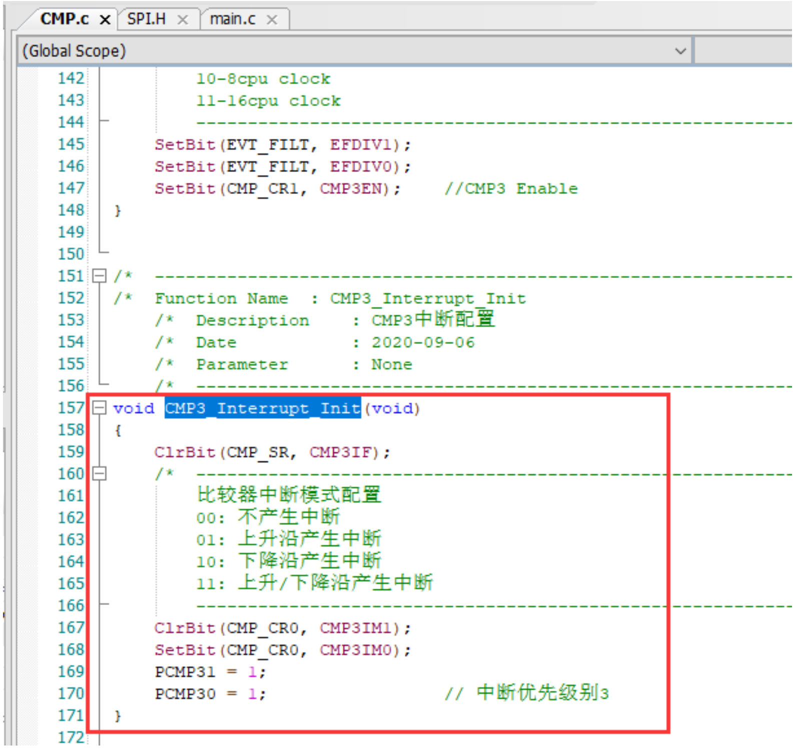
3 Code Editor

1. Support the association function. After defining a structure and structure members, and entering the structure name and English "." in the program, you can select the required structure members for the following operations；
2. Display the function at the mouse position, without sliding up and down to find the function entry;
3. Present the syntax of C and assembler by text style and color bar;
4. Directly reach the program line from the error list;
5. Support multi-byte characters and multiple encoding methods ;
6. Show/Hide editor edge ;
7. Parentheses match;
8. Auto Indent;
9. Indefinitely undo or redo an action in each window;
10. Click "Remark" to generate ".chm" help file to view the header file information;
11. Click to generate a “help.chm” help document to facilitate the user to view and understand the header file information;
12. Support the differential display of predefined failure codes, that is, graying out failure codes;
13. Support variable preview function. The variable definition is displayed when the mouse moves onto the variable. It is not necessary to right-click and select "Go to Definition Of 'XXX'" to view the variable;
14. Support absolute function positioning;
15. Jump to a function definition. You can move the mouse onto a function, right click and select "Go to Definition Of 'XXX'" to jump to the definition of the function. As shown in Figure 3-1, right click and select "Go to Definition Of 'CMP3_Interrupt_Init'" to jump to its corresponding function definition, as shown in Figure 3-2.


SPI.H
main.c
Global Scope)
HardwareInit
Go to Definition Of 'CMP3_Interrupt_Init' F12
Go to Declaration Of 'CMP3_Interrupt_Init' Ctrl++
Open Document
Toggle Breakpoint F9
Remove All Breakpoints
Disable All Breakpoints
Run to Cursor Ctrl+F10
Navigate Backwards Ctrl+-
Navigate Forwards Ctrl+_
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Format Code Ctrl+K, Ctrl+D
Format Selected Code Ctrl+K, Ctrl+F
Toggle Outlining Ctrl+M, Ctrl+L
Toggle Outlining Current Ctrl+M, Ctrl+M
View
Indent
Convert Case
Line Operation
Comment/Uncomment
Space Character Operation

```
        #if defined (SPI_DBG_SW) | defined (SPI_DBG_HW)    // 调试模式
        SPI_Init();                                  //调试模式下，占用SPI端口的NSS
        #endif
        /* -----比较器中断配置----- */
        CMP3_Interrupt_Ini
        Timer3_Init();
        //Timer4_Init();
        /* -----SYSTICK定E
        SetBit(DRV_SR, SYS
    }

/*  ------------------
        Function Name  : S
        Description    : 
        Date           : 2
        Parameter      : N
        ------------------
void SoftwareInit(voic
{
        MotorcontrolInit()
        mcState        = mc
        mcFaultSource = 0;
}

/*  ------------------
        Function Name  : V
        Description    : 
        Date           : 2
        Parameter      : N
        ------------------
void VREFConfigInit(vc
{
        /* **************
        ClrBit(P3_AN, PIN5
```

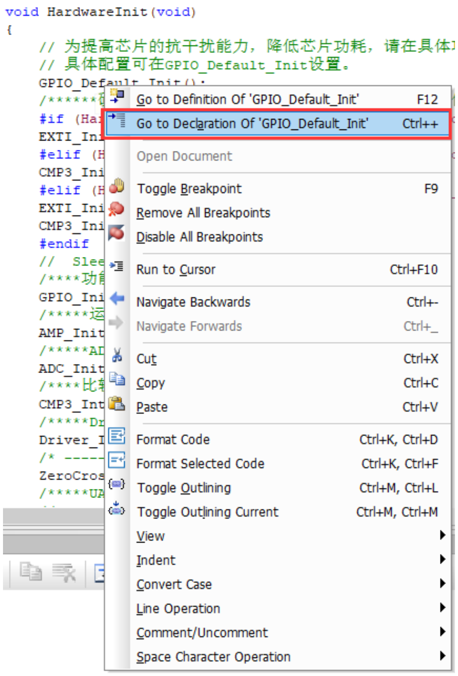
Figure 3-1



```
            10-8cpu clock
            11-16cpu clock
            ----------------------------------------------------------
        SetBit(EVT_FILT, EFDIV1);
        SetBit(EVT_FILT, EFDIV0);
        SetBit(CMP_CR1, CMP3EN);     //CMP3 Enable
}


/*  ---------------------------------------------------------------
/*  Function Name  : CMP3_Interrupt_Init
    /*  Description    : CMP3中断配置
    /*  Date           : 2020-09-06
    /*  Parameter      : None
    /*  ---------------------------------------------------------------
void CMP3_Interrupt_Init(void)
{
        ClrBit(CMP_SR, CMP3IF);
        /*  -----------------------------------------------------------
            比较器中断模式配置
            00: 不产生中断
            01: 上升沿产生中断
            10: 下降沿产生中断
            11: 上升/下降沿产生中断
            -----------------------------------------------------------
        ClrBit(CMP_CR0, CMP3IM1);
        SetBit(CMP_CR0, CMP3IM0);
        PCMP31 = 1;
        PCMP30 = 1;                  // 中断优先级别3
}
```

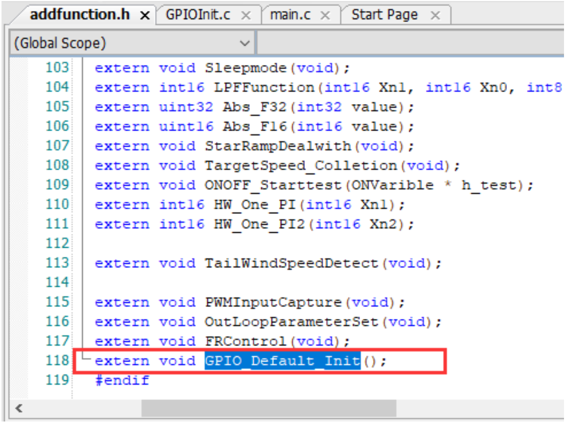
Figure 3-2

16. Jump to the declaration of a function in the header file. You can move the mouse onto a function, right click and select "Go to Declaration Of 'XXX'" to jump to the declaration of a function in the header file. As shown in Figure 3-3, right click and select "Go to Declaration Of 'GPIO_Default_Init'" to jump to the function declaration in the head file, as shown in Figure 3-4.



```
void HardwareInit(void)
{
    // 为提高芯片的抗干扰能力，降低芯片功耗，请在具体
    // 具体配置可在GPIO_Default_Init设置。
    GPIO_Default_Init();
    /******
    #if (Ha
    EXTI_Ini
    #elif (H
    CMP3_Ini
    #elif (H
    EXTI_Ini
    CMP3_Ini
    #endif
    //  Slee
    /****功
    GPIO_Ini
    /*****运
    AMP_Init
    /*****AD
    ADC_Init
    /****比
    CMP3_Int
    /*****Dr
    Driver_I
    /* -----
    ZeroCros
    /*****UA
```

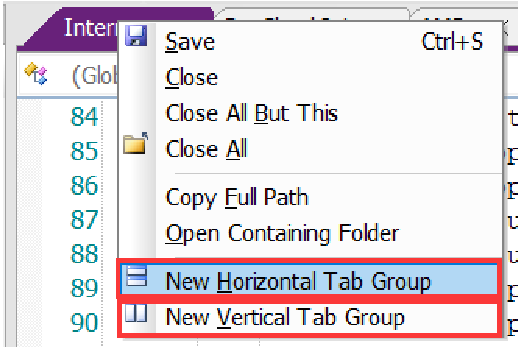
Go to Definition Of 'GPIO_Default_Init' F12
Go to Declaration Of 'GPIO_Default_Init' Ctrl++
Open Document
Toggle Breakpoint F9
Remove All Breakpoints
Disable All Breakpoints
Run to Cursor Ctrl+F10
Navigate Backwards Ctrl+-
Navigate Forwards Ctrl+_
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Format Code Ctrl+K, Ctrl+D
Format Selected Code Ctrl+K, Ctrl+F
Toggle Outlining Ctrl+M, Ctrl+L
Toggle Outlining Current Ctrl+M, Ctrl+M
View
Indent
Convert Case
Line Operation
Comment/Uncomment
Space Character Operation



```
extern void Sleepmode(void);
extern int16 LPFFunction(int16 Xn1, int16 Xn0, int8
extern uint32 Abs_F32(int32 value);
extern uint16 Abs_F16(int16 value);
extern void StarRampDealwith(void);
extern void TargetSpeed_Colletion(void);
extern void ONOFF_Starttest(ONVarible * h_test);
extern int16 HW_One_PI(int16 Xn1);
extern int16 HW_One_PI2(int16 Xn2);

extern void TailWindSpeedDetect(void);

extern void PWMInputCapture(void);
extern void OutLoopParameterSet(void);
extern void FRControl(void);
extern void GPIO_Default_Init();
#endif
```

Figure 3-3 Figure 3-4

17. Support double-screen file display function . If you use this feature to view two files at the same time, such as C file and H file. Right click to select the horizontal screen or click ; right click to select the vertical screen or click , as shown in Figure 3-5; If you want to return the single-screen mode, click the option shown in Figure 3-6 or click again.



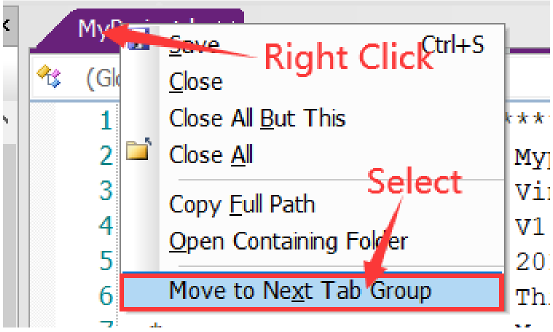
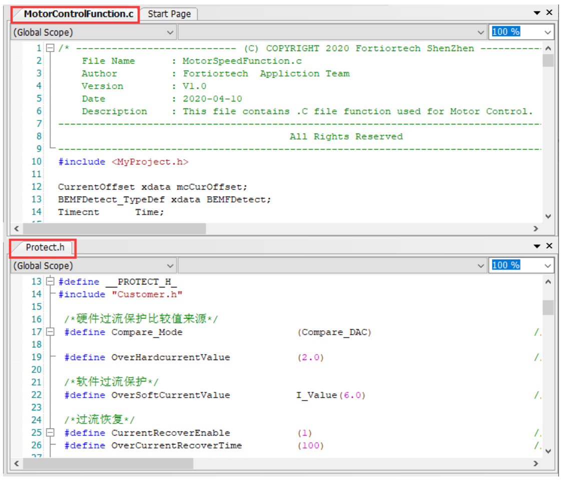


Figure 3-5 Figure 3-6

If horizontal screen is selected, the effect is shown in Figure 3-7 below;

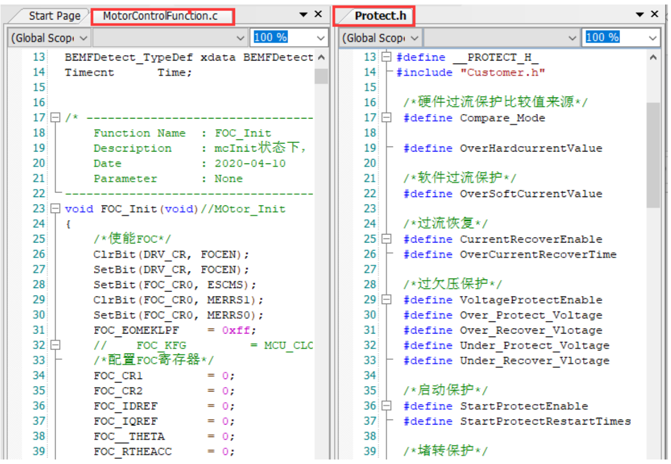


```
/* -------------------------- (C) COPYRIGHT 2020 Fortiortech ShenZhen ----------
     File Name      : MotorSpeedFunction.c
     Author         : Fortiortech  Appliction Team
     Version        : V1.0
     Date           : 2020-04-10
     Description    : This file contains .C file function used for Motor Control.
----------------------------------------------------------------------------------
                                       All Rights Reserved
----------------------------------------------------------------------------------
#include <MyProject.h>

CurrentOffset xdata mcCurOffset;
BEMFDetect_TypeDef xdata BEMFDetect;
Timecnt      Time;
```

```
#define __PROTECT_H_
#include "Customer.h"

 /*硬件过流保护比较值来源*/
 #define Compare_Mode                      (Compare_DAC)                    /.

 #define OverHardcurrentValue              (2.0)                            /.

 /*软件过流保护*/
 #define OverSoftCurrentValue              I_Value(6.0)                     /.

 /*过流恢复*/
 #define CurrentRecoverEnable              (1)                              /.
 #define OverCurrentRecoverTime            (100)                            /.
```

Figure 3-7

If vertical screen is selected, the effect is shown in Figure 3-8 below.

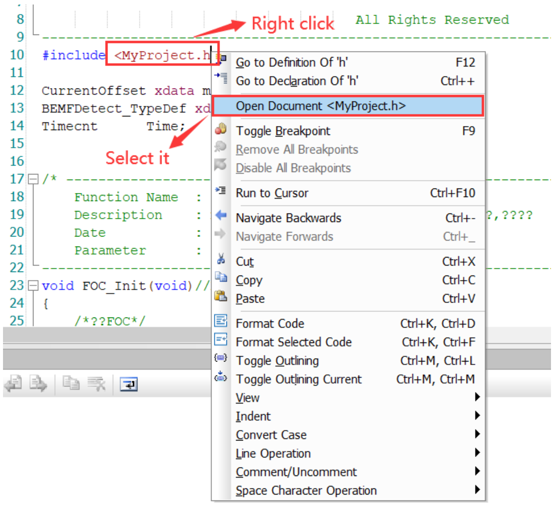


```
BEMFDetect_TypeDef xdata BEMFDetect
Timecnt      Time;


/* -----------------------------------
     Function Name   : FOC_Init
     Description     : mcInit状态下,
     Date            : 2020-04-10
     Parameter       : None
------------------------------------
void FOC_Init(void)//MOtor_Init
{
     /*使能FOC*/
     ClrBit(DRV_CR, FOCEN);
     SetBit(DRV_CR, FOCEN);
     SetBit(FOC_CR0, ESCMS);
     ClrBit(FOC_CR0, MERRS1);
     SetBit(FOC_CR0, MERRS0);
     FOC_EOMEKLPF      = 0xff;
     //    FOC_KFG          = MCU_CLC
     /*配置FOC寄存器*/
     FOC_CR1           = 0;
     FOC_CR2           = 0;
     FOC_IDREF         = 0;
     FOC_IQREF         = 0;
     FOC__THETA        = 0;
     FOC_RTHEACC       = 0;
```

```
#define __PROTECT_H_
#include "Customer.h"

 /*硬件过流保护比较值来源*/
 #define Compare_Mode

 #define OverHardcurrentValue

 /*软件过流保护*/
 #define OverSoftCurrentValue

 /*过流恢复*/
 #define CurrentRecoverEnable
 #define OverCurrentRecoverTime

 /*过欠压保护*/
 #define VoltageProtectEnable
 #define Over_Protect_Voltage
 #define Over_Recover_Vlotage
 #define Under_Protect_Voltage
 #define Under_Recover_Vlotage

 /*启动保护*/
 #define StartProtectEnable
 #define StartProtectRestartTimes

 /*堵转保护*/
```

Figure 3-8

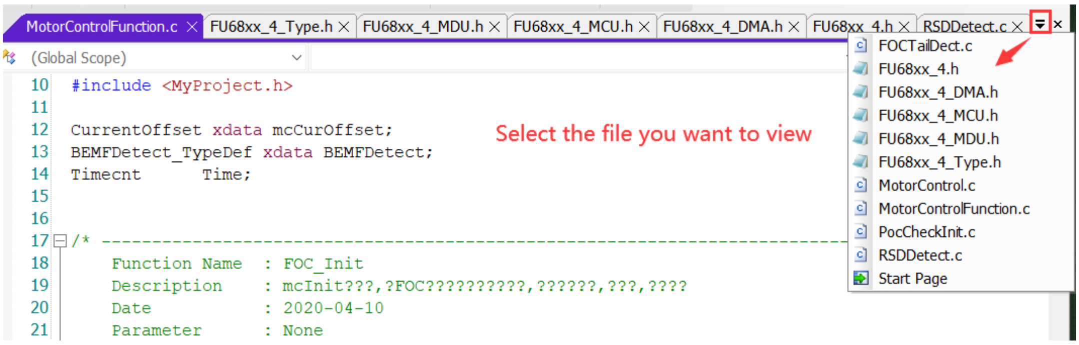
18. Quickly open the header file, such as the header file contained in the C file. You can quickly open the header file, as shown in Figure 3-9.


Right click
Select it
Go to Definition Of 'h' F12
Go to Declaration Of 'h' Ctrl++
Open Document <MyProject.h>
Toggle Breakpoint F9
Remove All Breakpoints
Disable All Breakpoints
Run to Cursor Ctrl+F10
Navigate Backwards Ctrl+-
Navigate Forwards Ctrl+_
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Format Code Ctrl+K, Ctrl+D
Format Selected Code Ctrl+K, Ctrl+F
Toggle Outlining Ctrl+M, Ctrl+L
Toggle Outlining Current Ctrl+M, Ctrl+M
View
Indent
Convert Case
Line Operation
Comment/Uncomment
Space Character Operation

```
                                                 All Rights Reserved
---------------------------------------------------------------------
#include <MyProject.h

CurrentOffset xdata m
BEMFDetect_TypeDef xd
Timecnt       Time;


/* ------------------                                        --------
     Function Name   :
     Description     :                                        ?,????
     Date            :
     Parameter       :
---------------------                                        --------
void FOC_Init(void)//
{
     /*??FOC*/
```

Figure 3-9

19. If there are too many open files and you don't want to close them, you can select them in the way shown in Figure 3-10.

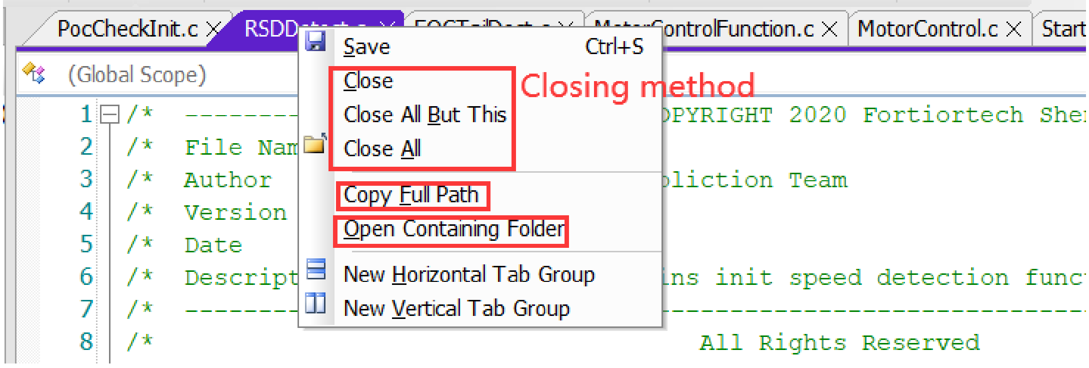


```
#include <MyProject.h>

CurrentOffset xdata mcCurOffset;
BEMFDetect_TypeDef xdata BEMFDetect;
Timecnt      Time;

/* ---------------------------------------------------------------------------
    Function Name  : FOC_Init
    Description    : mcInit???,?FOC??????????,??????,???,????
    Date           : 2020-04-10
    Parameter      : None
```

Select the file you want to view

Figure 3-10

You can close the files in the way shown in Figure 3-11. You can also open the corresponding folder or copy the file path. It is very convenient to open the library file, and you do not need to find the file one by one in the installation directory.



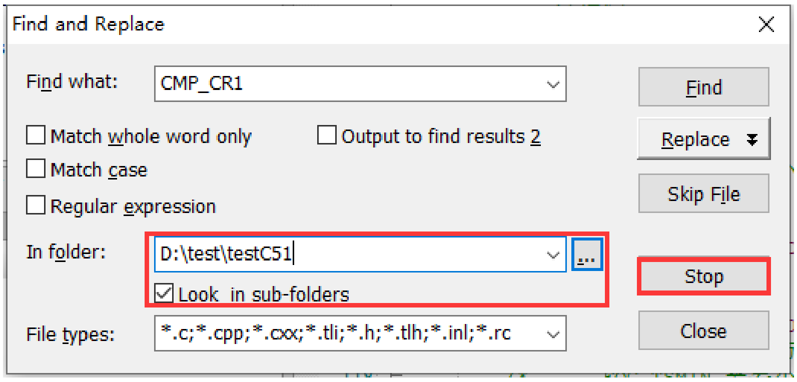
```
/* --------
/* File Nam            OPYRIGHT 2020 Fortiortech She
/* Author              plication Team
/* Version
/* Date
/* Descript            ins init speed detection func
/* --------            --------------------------------
/*                      All Rights Reserved
```

Closing method

Figure 3-11

20. Find and replace: click or press Ctrl+F or Ctrl+H to implement the feature, as shown in Figure 3-12. The search results are shown in Figure 3-13.

- In folder: Select the folder to search;
- Look in sub-folders: Search the subfolders under the folder;
- Stop: Stop searching;



Find what: CMP_CR1

□ Match whole word only □ Output to find results 2

□ Match case

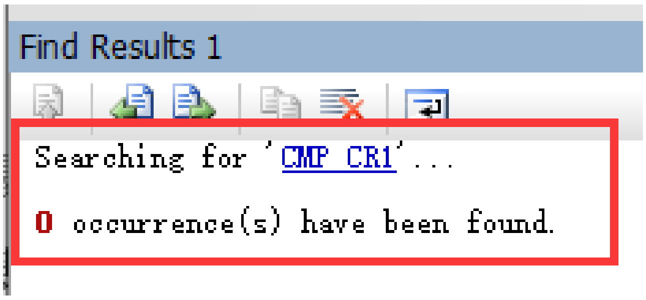
□ Regular expression

In folder: D:\test\testC51

☑ Look in sub-folders

File types: *.c;*.cpp;*.cxx;*.tli;*.h;*.tlh;*.inl;*.rc

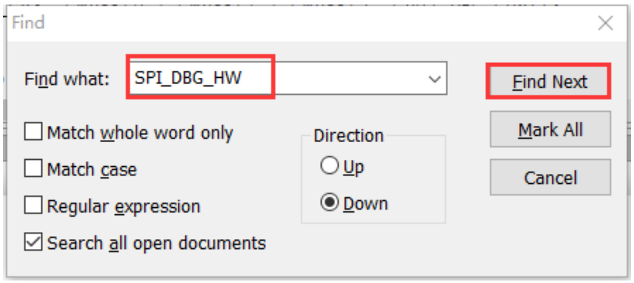
Figure 3-12



```
Searching for 'CMP_CR1'...
0 occurrence(s) have been found.
```

Figure 3-13

Separate search is operated as shown in Figure 3-14. Enter the characters you want to find, then select the search method and direction, and click "Find Next". There are two search directions: Up or Down, and four search methods: "Match whole word only", "Match case", "Regular expression" and "Search all open documents".



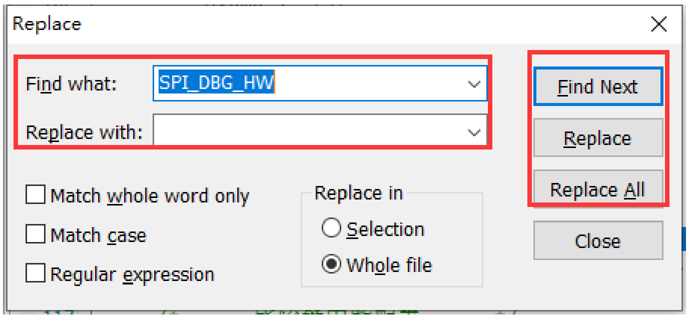
Find what: SPI_DBG_HW

□ Match whole word only
□ Match case
□ Regular expression
☑ Search all open documents

Direction: ○ Up ● Down

Figure 3-14

Individual replacement is operated as shown in Figure 3-15. Enter the characters that you want you find, then enter the replaced characters. Click "Find Next" to search the characters, and click "Replace" to replace them one by one. You can also click "Replace All" to replace all the characters at a time. There are three search methods: "Match whole word only", "Match case" and "Regular expression". The replacement range includes the current selected file and the entire file.



Find what: SPI_DBG_HW
Replace with:

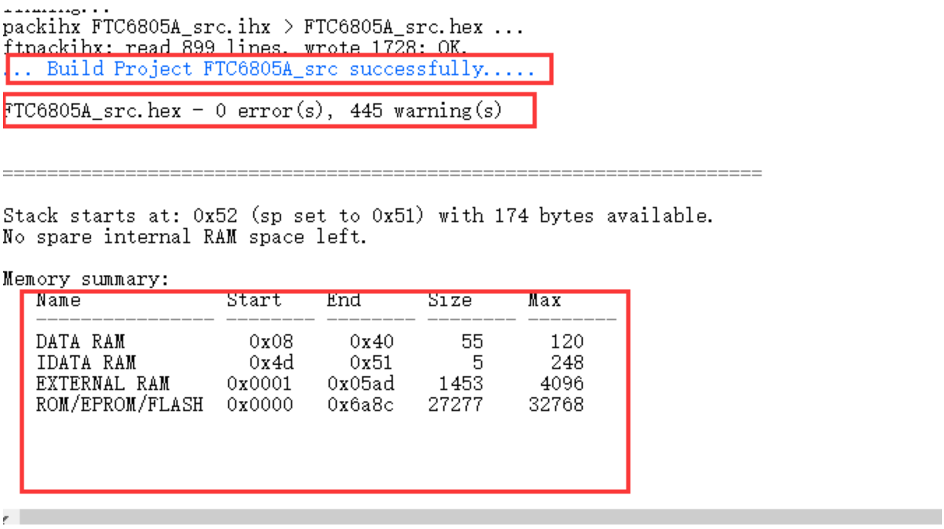
□ Match whole word only
□ Match case
□ Regular expression

Replace in: ○ Selection ● Whole file

Figure 3-15

4 Compiler

Click or to compile the code and generate the ".hex" file. If the code is compiled successfully, as shown in Figure 4-1 below, Output window displays the blue font "Build Project FTC6805A_src successfully", and generates "FTC6805A_src.hex" file with 0 error. If there is a warning, the number of warning is displayed as well.



```
packihx FTC6805A_src.ihx > FTC6805A_src.hex ...
ftpackihx: read 899 lines, wrote 1728: OK.
... Build Project FTC6805A_src successfully.....

FTC6805A_src.hex - 0 error(s), 445 warning(s)

==================================================================

Stack starts at: 0x52 (sp set to 0x51) with 174 bytes available.
No spare internal RAM space left.

Memory summary:
   Name                Start     End       Size      Max
   ----------------    --------  --------  --------  --------
   DATA RAM              0x08      0x40       55       120
   IDATA RAM             0x4d      0x51        5       248
   EXTERNAL RAM        0x0001    0x05ad     1453      4096
   ROM/EPROM/FLASH     0x0000    0x6a8c    27277     32768
```

Figure 4-1

As shown in Figure 4-1, after the code is compiled successfully, the start and end addresses, space size and maximum memory space of each RAM are displayed.

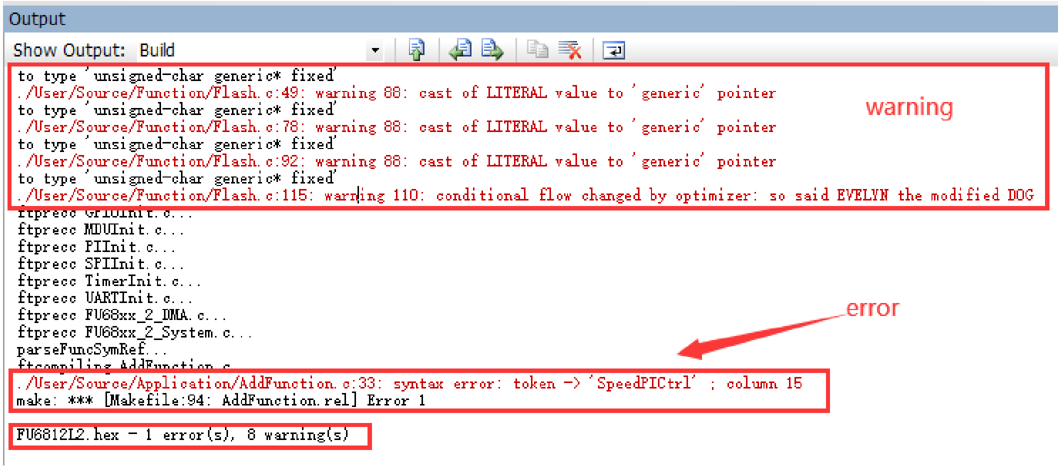
The start address of DATA RAM is 0x08, the end address is 0x40, the space size is 55 bytes, and the maximum memory is 120 bytes;

The start address of IDATA RAM is 0x4d, the end address is 0x51, the space size is 5 bytes, and the maximum memory is 248 bytes;

The start address of EXTERNAL RAM is 0x0001, the end address is 0x05ad, the space size is 1453 bytes, and the maximum memory is 4096 bytes;

The start address of ROM/EPROM/FLASH is 0x0000, the end address is 0x6a8c, the space size is 27277 bytes, and the maximum memory is 32768 bytes.

If the compilation fails, the number of errors and warnings are displayed in the Output window, and the location of errors and warnings are indicated in red font. As shown in Figure 4-2, one error occurs, and it is generated at line 33 of the "AddFunction.c" file. The error type is syntax error, and the token ";" is missing.

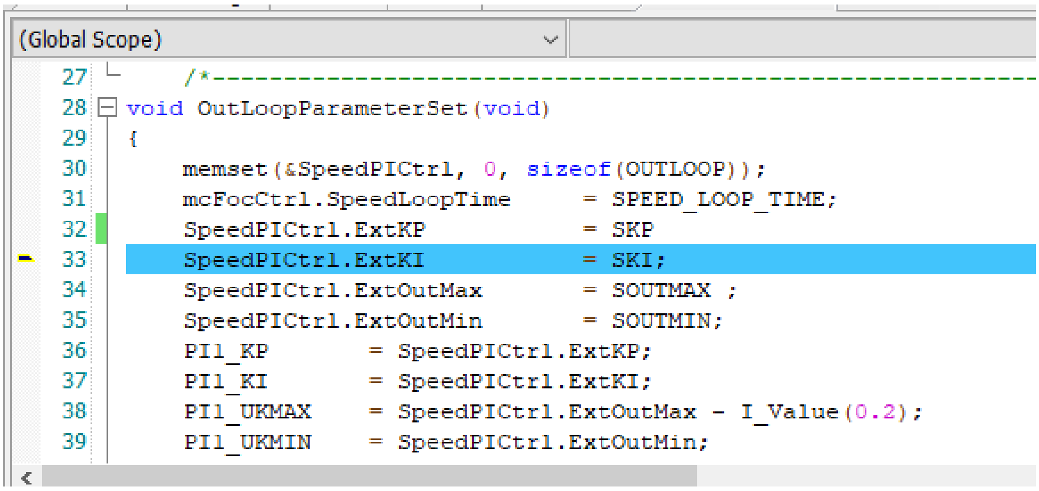


```
to type 'unsigned-char generic* fixed'
./User/Source/Function/Flash.c:49: warning 88: cast of LITERAL value to 'generic' pointer
to type 'unsigned-char generic* fixed'
./User/Source/Function/Flash.c:78: warning 88: cast of LITERAL value to 'generic' pointer
to type 'unsigned-char generic* fixed'
./User/Source/Function/Flash.c:92: warning 88: cast of LITERAL value to 'generic' pointer
to type 'unsigned-char generic* fixed'
./User/Source/Function/Flash.c:115: warning 110: conditional flow changed by optimizer: so said EVELYN the modified DOG
ftprecc GPIOInit.c...
ftprecc MDUInit.c...
ftprecc PIInit.c...
ftprecc SPIInit.c...
ftprecc TimerInit.c...
ftprecc UARTInit.c...
ftprecc FU68xx_2_DMA.c...
ftprecc FU68xx_2_System.c...
parseFuncSymRef...
ftcompiling AddFunction.c
./User/Source/Application/AddFunction.c:33: syntax error: token -> 'SpeedPICtrl' ; column 15
make: *** [Makefile:94: AddFunction.rel] Error 1

FU6812L2.hex - 1 error(s), 8 warning(s)
```

Figure 4-2

Double click the error to reach the location where the error occurs (similarly, double lick the warning to reach the location where the warning is generated), as shown in Figure 4-3, that is, the location where the blue cursor hits.

In particular, the compiler alerts that there is a syntax error in line 33, but in fact the error is often not in line 33. It is likely that the error is caused by a syntax error in front of line 33. In this case, line 33 and its preceding lines shall be checked as well. Through inspection, it is found that the error is in line 32. The token ";" is missing in the assignment statement. After the token is added, the compilation succeeds.



```
27      /*-----------------------------------------------------------
28  void OutLoopParameterSet(void)
29  {
30      memset(&SpeedPICtrl, 0, sizeof(OUTLOOP));
31      mcFocCtrl.SpeedLoopTime      = SPEED_LOOP_TIME;
32      SpeedPICtrl.ExtKP            = SKP
33      SpeedPICtrl.ExtKI            = SKI;
34      SpeedPICtrl.ExtOutMax        = SOUTMAX ;
35      SpeedPICtrl.ExtOutMin        = SOUTMIN;
36      PI1_KP       = SpeedPICtrl.ExtKP;
37      PI1_KI       = SpeedPICtrl.ExtKI;
38      PI1_UKMAX    = SpeedPICtrl.ExtOutMax - I_Value(0.2);
39      PI1_UKMIN    = SpeedPICtrl.ExtOutMin;
```

Figure 4-3

5 Download and Emulation

5.1 Download

FTM8ForgeIDE supports methods to download the program code: Smart Download, Normal Download and Without Download.

- Smart Download: After the chip is successfully programmed for the first time, if the program code is changed, programming operation is not performed during the next download so as to save time;
- Normal Download: The program code is downloaded each time when the chip is programmed;
- Without Download: No download operation is performed. IDE directly enters the simulation environment and considers the chip has been programmed.

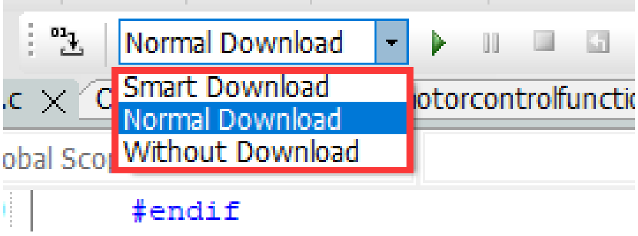
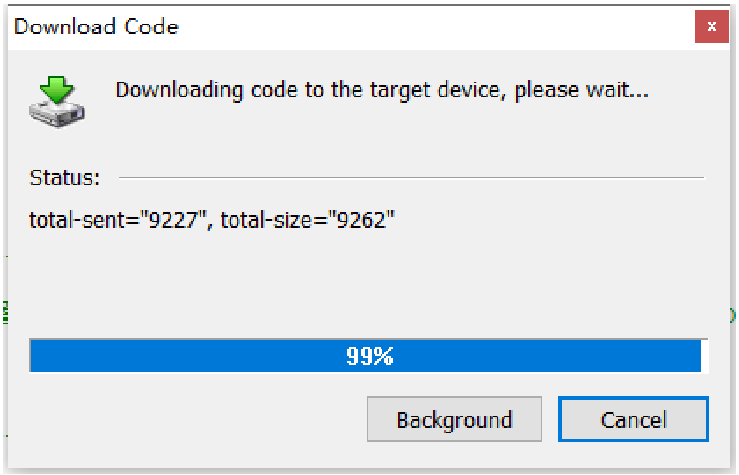


Figure 5-1

Connect the computer with the programming tool or development board, and click 01 to download the compiled hex file to the flash of the chip. The download process is shown in Figure 5-2.


Download Code

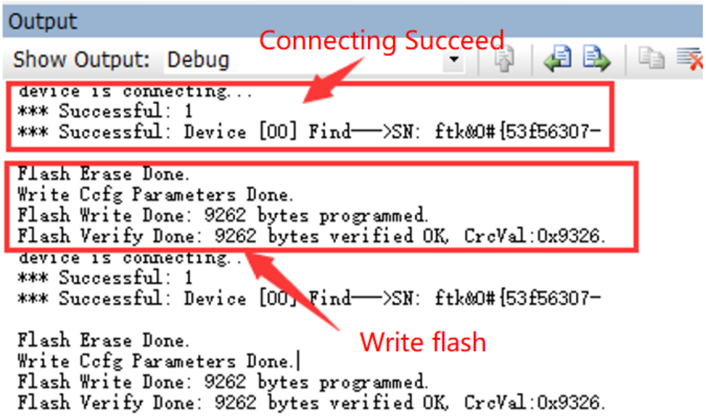
Downloading code to the target device, please wait...

Status:

total-sent="9227", total-size="9262"

Figure 5-2

The program code has been successfully downloaded to the chip when the download progress bar reaches 100%. The download results are as shown in Figure 5-3: the device has been successfully connected, the Flash erase has been completed, 9262 bytes of program code has been written to Flash, and the Flash verification is successful.



Connecting Succeed

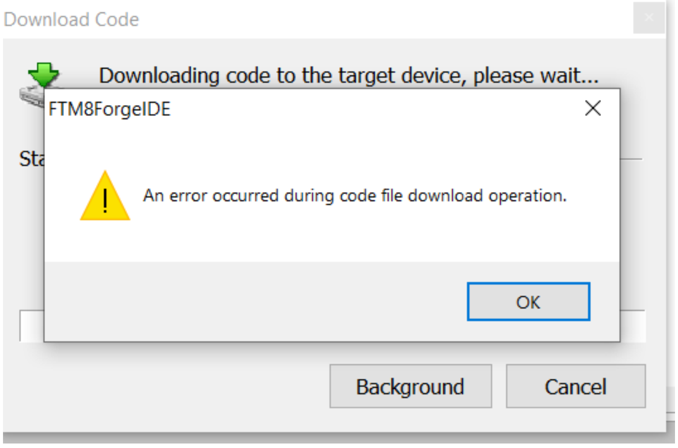
Write flash

```
device is connecting...
*** Successful: 1
*** Successful: Device [00] Find——>SN: ftk&0#{53f56307-

Flash Erase Done.
Write Cofg Parameters Done.
Flash Write Done: 9262 bytes programmed.
Flash Verify Done: 9262 bytes verified OK, CrcVal:0x9326.
device is connecting...
*** Successful: 1
*** Successful: Device [00] Find——>SN: ftk&0#{53f56307-

Flash Erase Done.
Write Cofg Parameters Done.|
Flash Write Done: 9262 bytes programmed.
Flash Verify Done: 9262 bytes verified OK, CrcVal:0x9326.
```

Figure 5-3

When you click the download button to download the program code and the computer fails to connect to the programming tool (such as the emulator) , a pop-up window appears to notify an error, as shown in Figure 5-4.

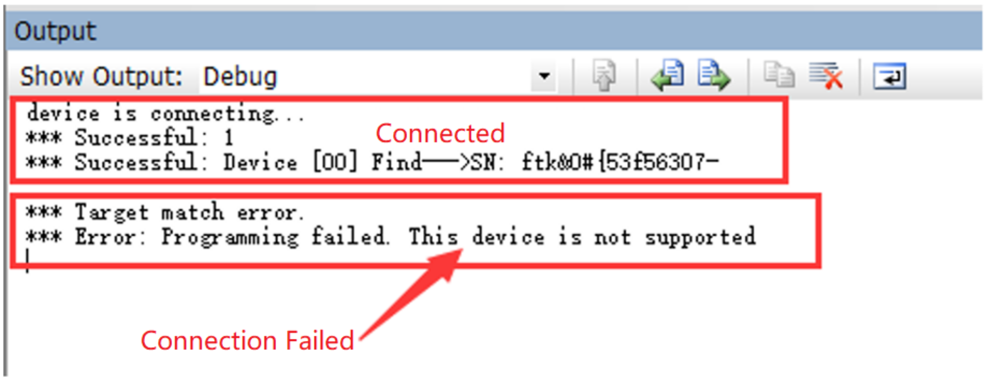
Similarly, you click the download button to download the program code and the computer fails to connect to development board, a pop-up window appears to notify an error, as shown in Figure 5-4. Also, Output window displays error message to alert the connection failure, as shown in Figure 5-5.



Downloading code to the target device, please wait...

An error occurred during code file download operation.

Figure 5-4



Connected

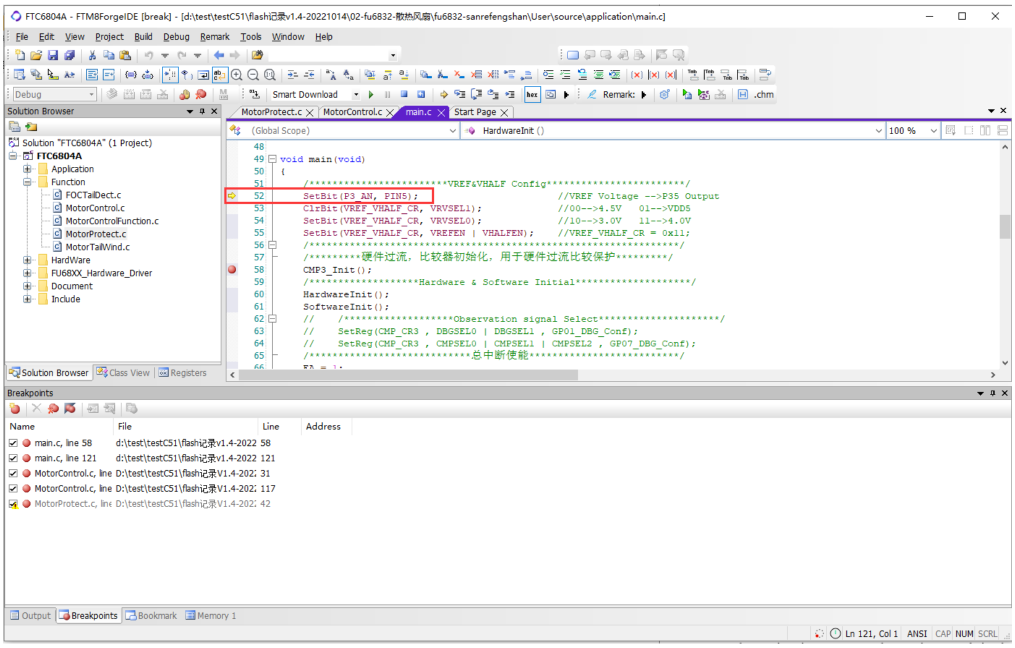
Connection Failed

```
device is connecting...
*** Successful: 1
*** Successful: Device [00] Find——>SN: ftk&0#{53f56307-

*** Target match error.
*** Error: Programming failed. This device is not supported
```

Figure 5-5

5.2 Emulation

Connect the computer with the programming tool or development board, click ▶ to conduct emulation debugging and download the compiled hex file to the flash of the chip. The download process is shown in Figure 5-2. When the download progress bar reaches 100%, the program code has been successfully downloaded to the chip, and can be used for emulation debugging. The debug interface is shown in Figure 5-6.

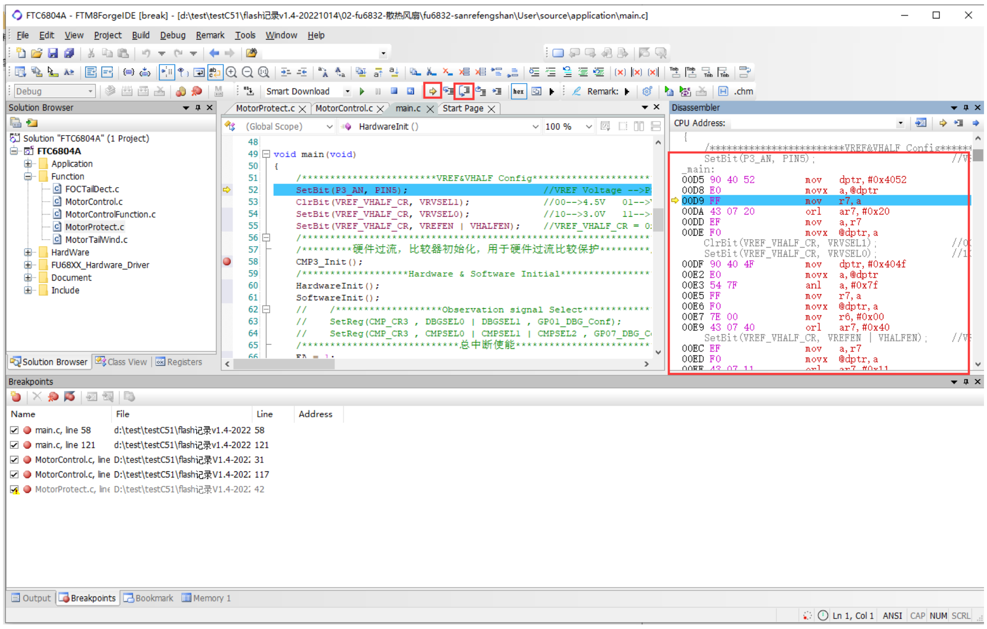


```
void main(void)
{
    /***********************VREF&VHALF Config***********************/
    SetBit(P3_AN, PIN5);                            //VREF Voltage -->P35 Output
    ClrBit(VREF_VHALF_CR, VRVSEL1);                 //00-->4.5V   01-->VDD5
    SetBit(VREF_VHALF_CR, VRVSEL0);                 //10-->3.0V   11-->4.0V
    SetBit(VREF_VHALF_CR, VREFEN | VHALFEN);        //VREF_VHALF_CR = 0x11;
    /****************************************************************/
    /*********硬件过流，比较器初始化，用于硬件过流比较保护*********/
    CMP3_Init();
    /*******************Hardware & Software Initial*******************/
    HardwareInit();
    SoftwareInit();
//    /*******************Observation signal Select*******************/
//    SetReg(CMP_CR3 , DBGSEL0 | DBGSEL1 , GP01_DBG_Conf);
//    SetReg(CMP_CR3 , CMPSEL0 | CMPSEL1 | CMPSEL2 , GP07_DBG_Conf);
    /***************************总中断使能***************************/
```

Name	File	Line	Address
☑ ● main.c, line 58	d:\test\testC51\flash记录v1.4-2022	58	
☑ ● main.c, line 121	d:\test\testC51\flash记录v1.4-2022	121	
☑ ● MotorControl.c, line	D:\test\testC51\flash记录V1.4-202:	31	
☑ ● MotorControl.c, line	D:\test\testC51\flash记录V1.4-202:	117	
☑ ● MotorProtect.c, line	D:\test\testC51\flash记录V1.4-202:	42	

Figure 5-6

Click ⇨ to perform disassembly instruction debugging, and click ⇲ or F10 to conduct single-step debugging, as shown in Figure 5-7.

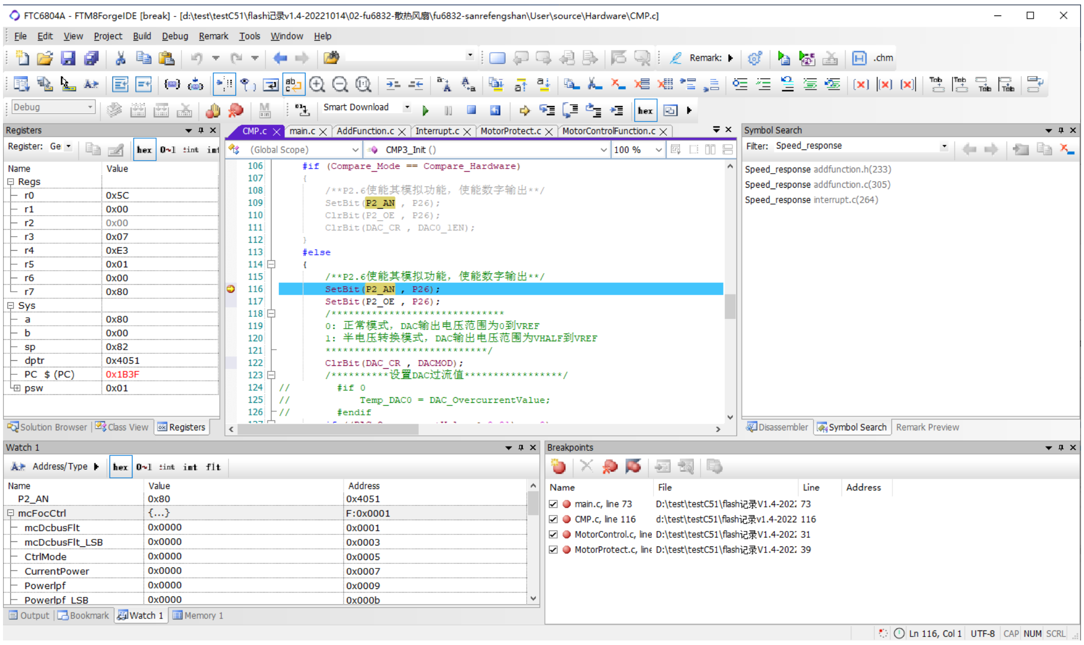


```
{
    /***********************VREF&VHALF Config******
    SetBit(P3_AN, PIN5);                                //V
_main:
00D5 90 40 52          mov    dptr,#0x4052
00D8 E0                movx   a,@dptr
00D9 FF                mov    r7,a
00DA 43 07 20          orl    ar7,#0x20
00DD EF                mov    a,r7
00DE F0                movx   @dptr,a
    ClrBit(VREF_VHALF_CR, VRVSEL1);                     //0
    SetBit(VREF_VHALF_CR, VRVSEL0);                     //1
00DF 90 40 4F          mov    dptr,#0x404f
00E2 E0                movx   a,@dptr
00E3 54 7F             anl    a,#0x7f
00E5 FF                mov    r7,a
00E6 F0                movx   @dptr,a
00E7 7E 00             mov    r6,#0x00
00E9 43 07 40          orl    ar7,#0x40
    SetBit(VREF_VHALF_CR, VREFEN | VHALFEN);            //V
00EC EF                mov    a,r7
00ED F0                movx   @dptr,a
```

Name	File	Line	Address
☑ ● main.c, line 58	d:\test\testC51\flash记录v1.4-2022	58	
☑ ● main.c, line 121	d:\test\testC51\flash记录v1.4-2022	121	
☑ ● MotorControl.c, line	D:\test\testC51\flash记录V1.4-202:	31	
☑ ● MotorControl.c, line	D:\test\testC51\flash记录V1.4-202:	117	
☑ ● MotorProtect.c, line	D:\test\testC51\flash记录V1.4-202:	42	

Figure 5-7

Click to set the breakpoint. Click or F5 to reach the next breakpoint from current code line, as shown in Figure 5-8.



FTC6804A - FTM8ForgeIDE [break] - [d:\test\testC51\flash记录v1.4-20221014\02-fu6832-散热风扇\fu6832-sanrefengshan\User\source\Hardware\CMP.c]

```
    #if (Compare_Mode == Compare_Hardware)
    {
        /**P2.6使能其模拟功能，使能数字输出**/
        SetBit(P2_AN , P26);
        ClrBit(P2_OE , P26);
        ClrBit(DAC_CR , DAC0_1EN);
    }
    #else
    {
        /**P2.6使能其模拟功能，使能数字输出**/
        SetBit(P2_AN , P26);
        SetBit(P2_OE , P26);
        /******************************
        0: 正常模式，DAC输出电压范围为0到VREF
        1: 半电压转换模式，DAC输出电压范围为VHALF到VREF
        ******************************/
        ClrBit(DAC_CR , DACMOD);
        /**********设置DAC过流值****************/
//        #if 0
//            Temp_DAC0 = DAC_OvercurrentValue;
//        #endif
```

Registers

Name	Value
Regs	
r0	0x5C
r1	0x00
r2	0x00
r3	0x07
r4	0xE3
r5	0x01
r6	0x00
r7	0x80
Sys	
a	0x80
b	0x00
sp	0x82
dptr	0x4051
PC $ (PC)	0x1B3F
psw	0x01

Watch 1

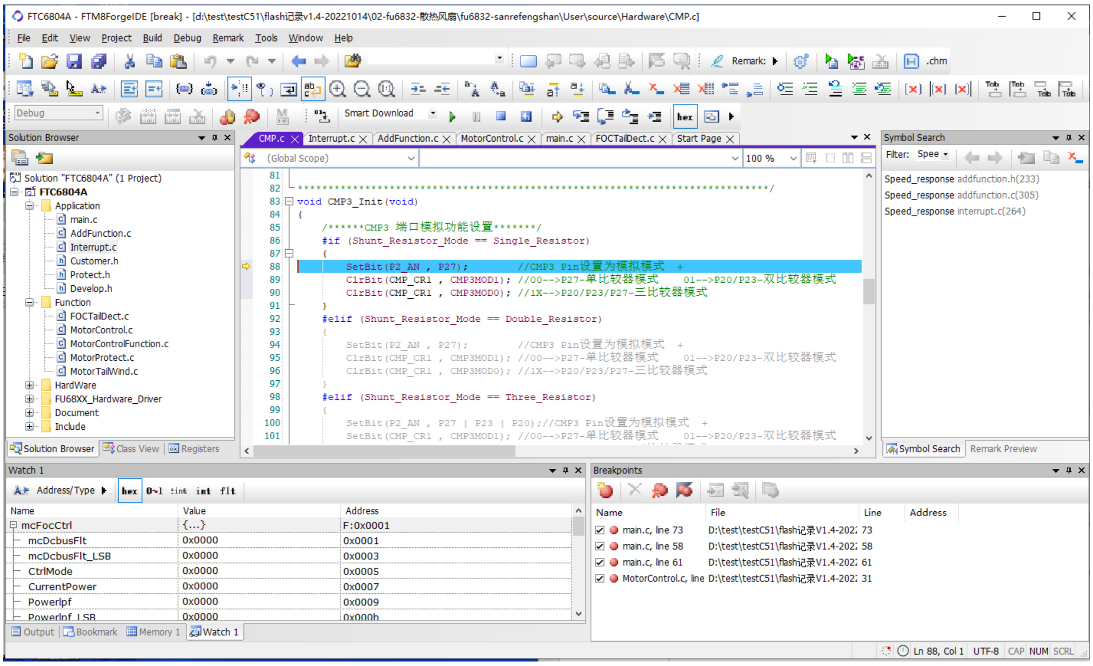
Name	Value	Address
P2_AN	0x80	0x4051
mcFocCtrl	{...}	F:0x0001
mcDcbusFlt	0x0000	0x0001
mcDcbusFlt_LSB	0x0000	0x0003
CtrlMode	0x0000	0x0005
CurrentPower	0x0000	0x0007
Powerlpf	0x0000	0x0009
Powerlpf_LSB	0x0000	0x000b

Breakpoints

Name	File	Line	Address
☑ main.c, line 73	D:\test\testC51\flash记录V1.4-202:	73	
☑ CMP.c, line 116	d:\test\testC51\flash记录v1.4-2022	116	
☑ MotorControl.c, line	D:\test\testC51\flash记录V1.4-202:	31	
☑ MotorProtect.c, line	D:\test\testC51\flash记录V1.4-202:	39	

Figure 5-8

Click to perform debugging in a function, and click to exit the function, as shown in Figure 5-9.



FTC6804A - FTM8ForgeIDE [break] - [d:\test\testC51\flash记录v1.4-20221014\02-fu6832-散热风扇\fu6832-sanrefengshan\User\source\Hardware\CMP.c]

```
******************************************************************************/
void CMP3_Init(void)
{
    /******CMP3 端口模拟功能设置*******/
    #if (Shunt_Resistor_Mode == Single_Resistor)
    {
        SetBit(P2_AN , P27);          //CMP3 Pin设置为模拟模式  +
        ClrBit(CMP_CR1 , CMP3MOD1); //00-->P27-单比较器模式     01-->P20/P23-双比较器模式
        ClrBit(CMP_CR1 , CMP3MOD0); //1X-->P20/P23/P27-三比较器模式
    }
    #elif (Shunt_Resistor_Mode == Double_Resistor)
    {
        SetBit(P2_AN , P27);          //CMP3 Pin设置为模拟模式  +
        ClrBit(CMP_CR1 , CMP3MOD1); //00-->P27-单比较器模式     01-->P20/P23-双比较器模式
        ClrBit(CMP_CR1 , CMP3MOD0); //1X-->P20/P23/P27-三比较器模式
    }
    #elif (Shunt_Resistor_Mode == Three_Resistor)
    {
        SetBit(P2_AN , P27 | P23 | P20);//CMP3 Pin设置为模拟模式  +
        SetBit(CMP_CR1 , CMP3MOD1); //00-->P27-单比较器模式     01-->P20/P23-双比较器模式
```

Watch 1

Name	Value	Address
mcFocCtrl	{...}	F:0x0001
mcDcbusFlt	0x0000	0x0001
mcDcbusFlt_LSB	0x0000	0x0003
CtrlMode	0x0000	0x0005
CurrentPower	0x0000	0x0007
Powerlpf	0x0000	0x0009
Powerlpf_LSB	0x0000	0x000b

Breakpoints

Name	File	Line	Address
☑ main.c, line 73	D:\test\testC51\flash记录V1.4-202:	73	
☑ main.c, line 58	D:\test\testC51\flash记录V1.4-202:	58	
☑ main.c, line 61	D:\test\testC51\flash记录V1.4-202:	61	
☑ MotorControl.c, line	D:\test\testC51\flash记录V1.4-202:	31	

Figure 5-9

Output window displays disassembly window, register window, watch window and memory window. See section 1.4 Window Introduction for details.

6 Revision History

Version	Description	Date	Prepared By
V1.0	First release. Translated from Chinese V1.0.	2023/08/11	Eric & Freya

Copyright Notice

If there are any differences between the Chinese and the English contents, please take the Chinese version as the standard.

Fortior Technology (Shenzhen) Co., Ltd.
Room 203, 2/F, Building No.11, Keji Central Road 2,
Software Park, High-Tech Industrial Park, Shenzhen, P.R. China 518057
Tel: 0755-26867710
Fax: 0755-26867715
URL: http://www.fortiortech.com

Contained herein