

KF8 Series

ChipON IDE User Manual

2016-12

Contents

1. OVERVIEW.....	5
2. INSTALLATION.....	6
3. START CHIPON IDE SOFTWARE.....	9
4. NEW ASSEMBLY PROJECT.....	11
5. NEW ASSEMBLY TYPE SOURCE FILE CODE AND COMPILE.....	17
6. NEW C PROJECT.....	21
7. NEW C TYPE SOURCE FILE CODE AND COMPILE.....	21
8. CONFIGURE CHIP CONFIGURATION BIT.....	21
9. DOWNLOAD HEX FILE.....	24
10. RESOURCE UTILIZATION.....	25
11. WRITE BACK THE CALIBRATION VALUES.....	26
12. PROJECT DEVELOPMENT AND DEBUGGING.....	26
13. FIRMWARE UPGRADE.....	35
14. HARDWARE EMULATION(DEBUGGER) COMMON ERROR RESOLUTION.....	36
15. IDE USAGE SKILLS.....	38
15.1 TOGGLE WORKSPACE.....	38
15.2 EXPORT PROJECT.....	38
15.3 IMPORT PROJECT.....	39
15.4 KEYWORD HIGHLIGHTING SETTINGS.....	40
15.5 CODE COMPLETION.....	41
15.6 REFACTOR.....	41
15.7 SHOW LINE NUMBERS.....	43
15.8 RELATED SHORTCUT KEYS.....	43
15.9 HISTORY FILE COMPARSION.....	44
15.10 CUSTOM PERSPECTIVE.....	46
15.11 FIND/REPLACE.....	46
15.12 RETURN TO THE PREVIOUS OPERATION POSITION.....	48
15.13 CLOSE PROJECT.....	48
15.14 OPEN PROJECT.....	49
15.15 MAXIMIZE, MINIMIZE AND CLOSE THE CURRENT WINDOW.....	49

15.16 QUICK COMMENT CODE.....	49
15.17 CHANGE THE POSITION OF THE VIEW WINDOW.....	50
15.18 CHANGE EDITOR FONT.....	50
15.19 RESTORE INITIAL VIEW LAYOUT.....	51
15.20 CLEAN PROJECT.....	51
15.21 QUICKLY LOCATE ERROR INFORMATION.....	51
15.22 RESTORE DELETED FILES.....	52
15.23 DEBUG SETTINGS.....	54
15.24 CODE JUMP.....	54
15.25 MANAGER CONFIGURATION.....	55
15.26 VIEW COMPILE RESULTS.....	56
15.27 MAKE CODE COMPILATION EFFICIENCY HIGH.....	56

Before using this manual, please read the following license agreement carefully. The products described in this manual can only be used if the following license agreement is agreed.

Copyright Notice

Without the written permission of Shanghai ChipON Microelectronics Technology Co., Ltd., no company or individual may copy all or part of this user manual in any form.

Important Statement

Shanghai ChipON Microelectronics Technology Co., Ltd. strives to make the information provided in this manual accurate and applicable. However, the products and manuals may include technical or printing errors. Shanghai ChipON Microelectronics Co., Ltd. reserves the right to change all or part of this User Manual without prior notice.

1. Overview

ChipON IDE is an integrated development environment for KungFu chip development and use, which is produced by Shanghai ChipON Microelectronics Technology Co., Ltd.

The ChipON IDE mainly supports the following functions:

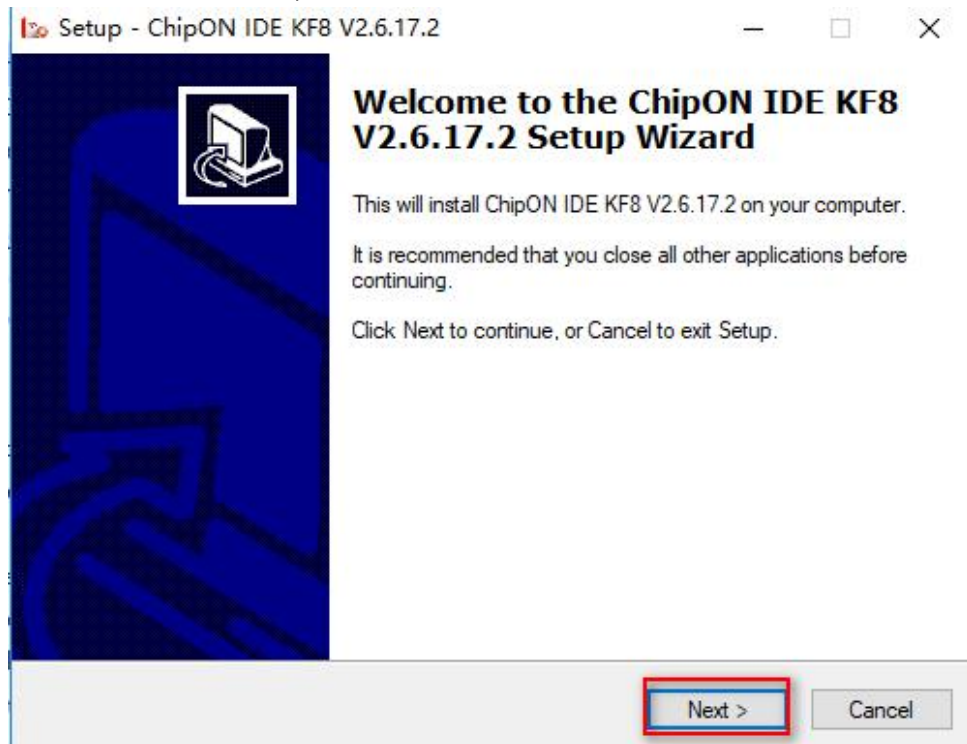
- ◆ Support compilation projects
- ◆ Support C language projects
- ◆ Support ICSP online download
- ◆ Support multi-project management
- ◆ Support assembly and intelligent input of C language
- ◆ Support assembly and floating prompt of C language code
- ◆ Support real-time viewing of chip resource utilization
- ◆ Support to view information about related chips
- ◆ Support viewing and editing of HEX files
- ◆ Supports highlighting assembly keywords in C language

ChipON IDE supports the following chip types, but more types are supported in the tool.

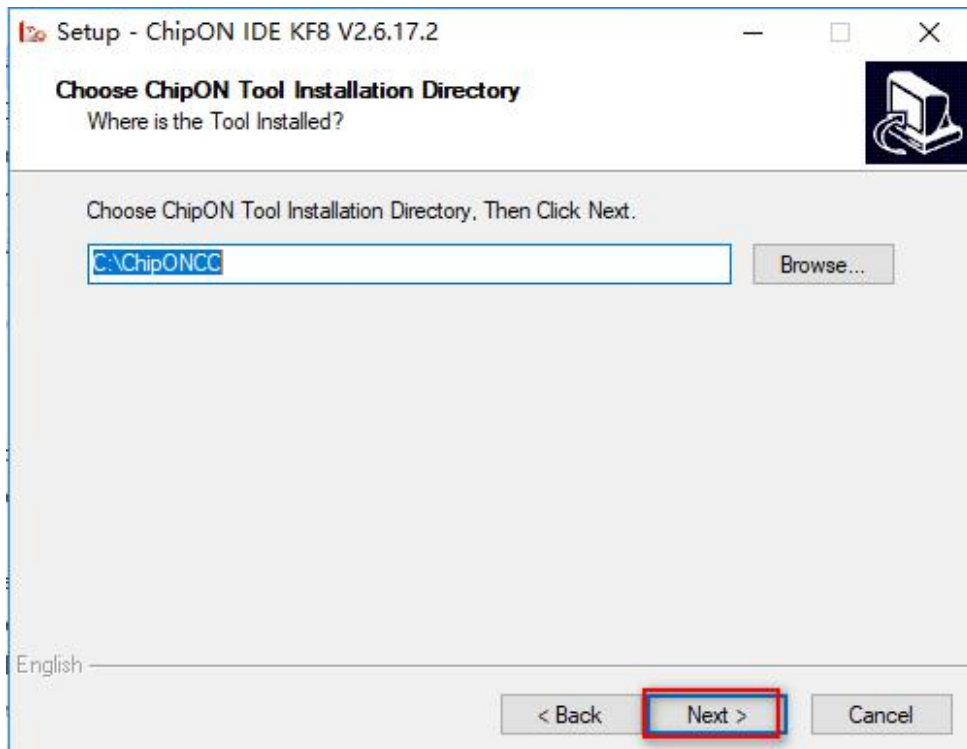
KF8F200	KF8F201	KF8F202	KF8F203
KF8F204	KF8F210	KF8F211	KF8F212
KF8F213	KF8F214	KF8F221	KF8F222
KF8F232	KF8F233	KF8F300	KF8F303
KF8F304	KF8F310	KF8F312	KF8F313
KF8F321	KF8F323	KF8F324	KF8F333
KF8F334	KF8F335	KF8F336	KF8S1005
KF8S1006	KF8S1010	KF8S1011	KF8S1022
KF8S1023	KF8S1024	KF8S1025	KF8S1100
KF8S1101	KF8S1001	KF8S1020	KF8V111
KF8V112	KF8V120	KF8V200	KF8V211
KF8V212	KF8V216	KF8V218	KF8V220
KF8TS2402	KF8TS2408	KF8TS2410	KF8TS2414
KF8TS2302	KF8TS2308	KF8TS2310	KF8TS2314
KF8F4156	KF8F4158	KF8V204	KF8V427
KF8TS2716	KF8S210	KF8S212	KF8S360
KF8F4155	KF8F3155	KF8F2156	KF8F3156
KF8TS2516	KF8V327	KF8V429	KF8S1007
KF8F1020	KF8TS2702	KF8TS2708	KF8TS2710
KF8TS2714	KF8V304	KF8V404	KF8V325

2. Installation

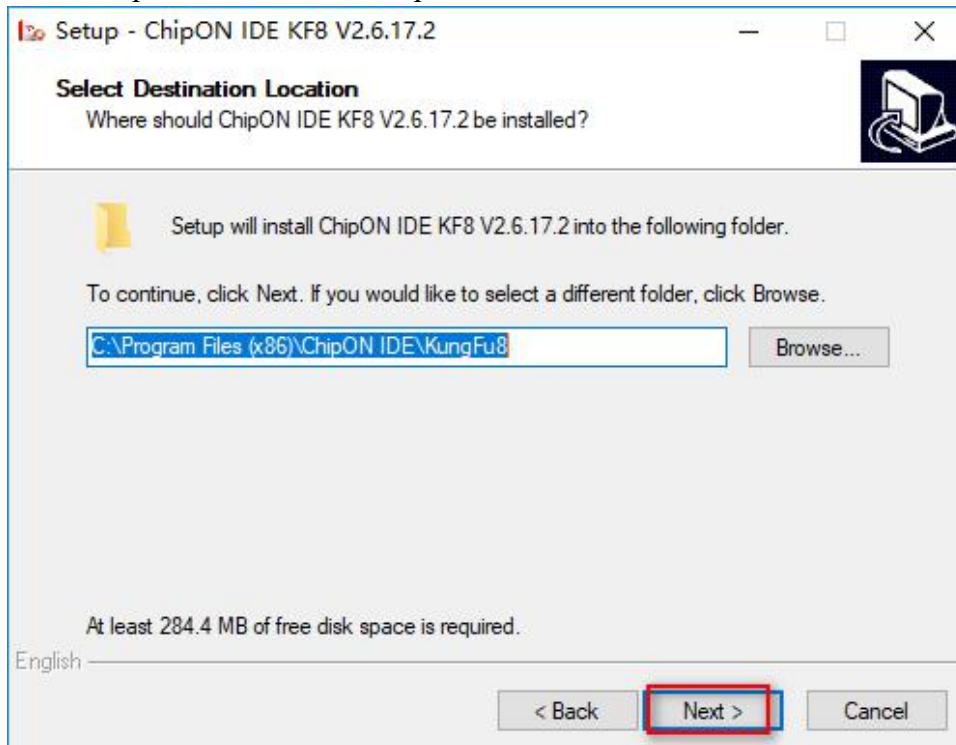
Double-click the ChipON IDE VX.X.X.exe installation file to enter the installation wizard interface, select "Next".



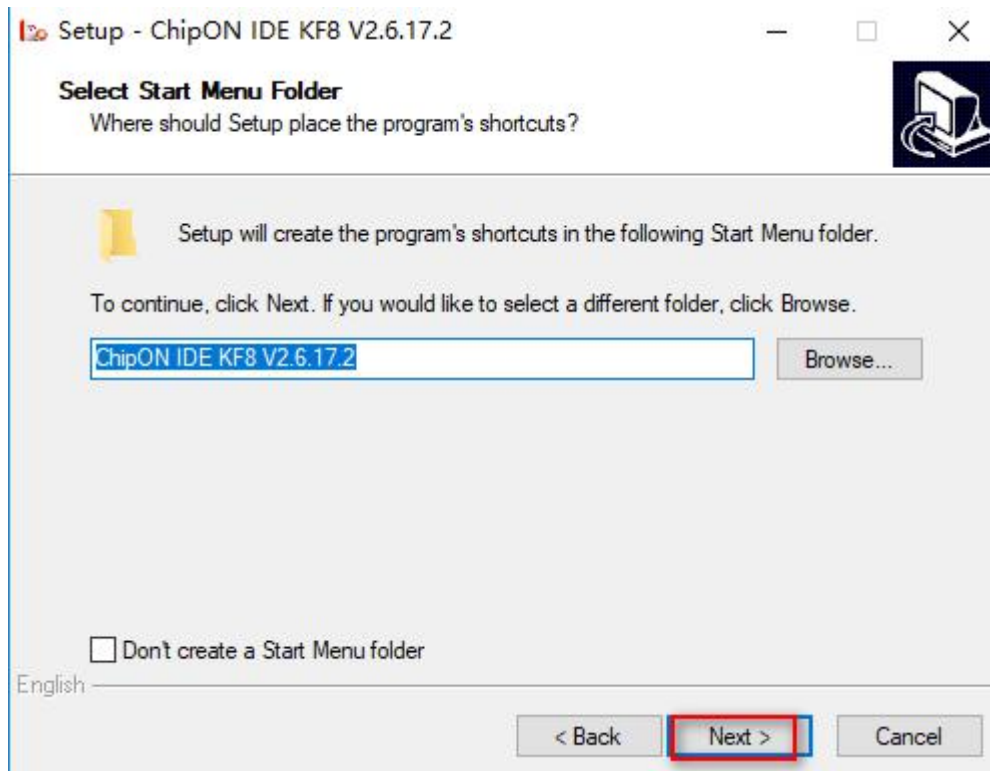
Set the directory to install the ChipON toolkit and then click "Next".



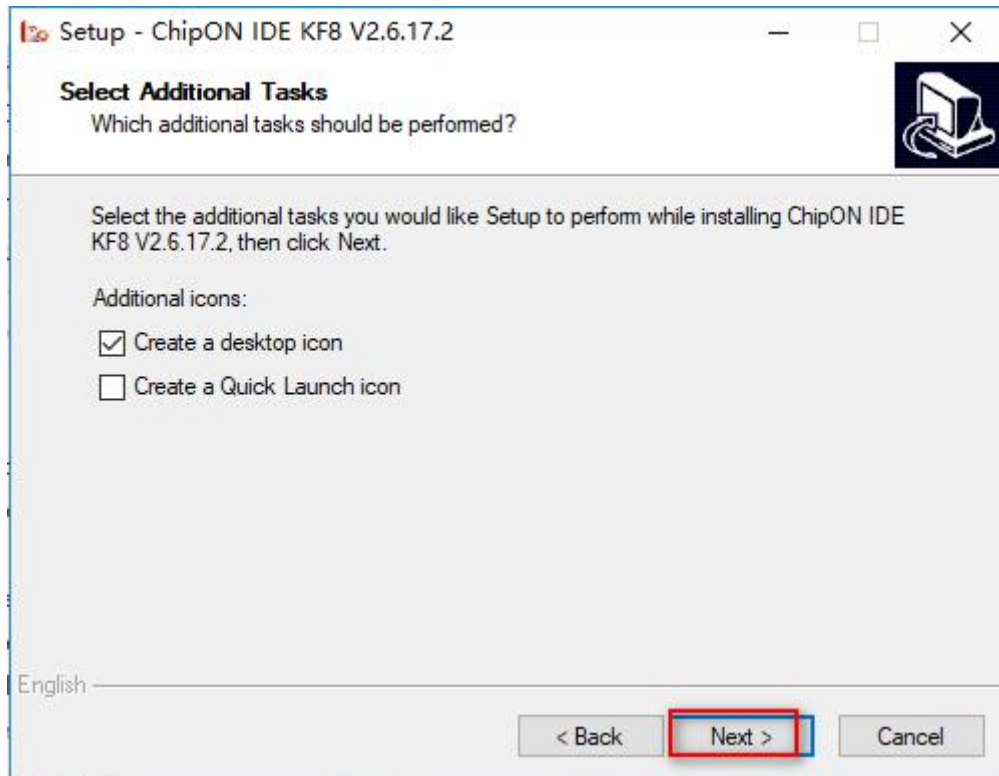
Set the ChipON IDE installation path, and then click "Next".



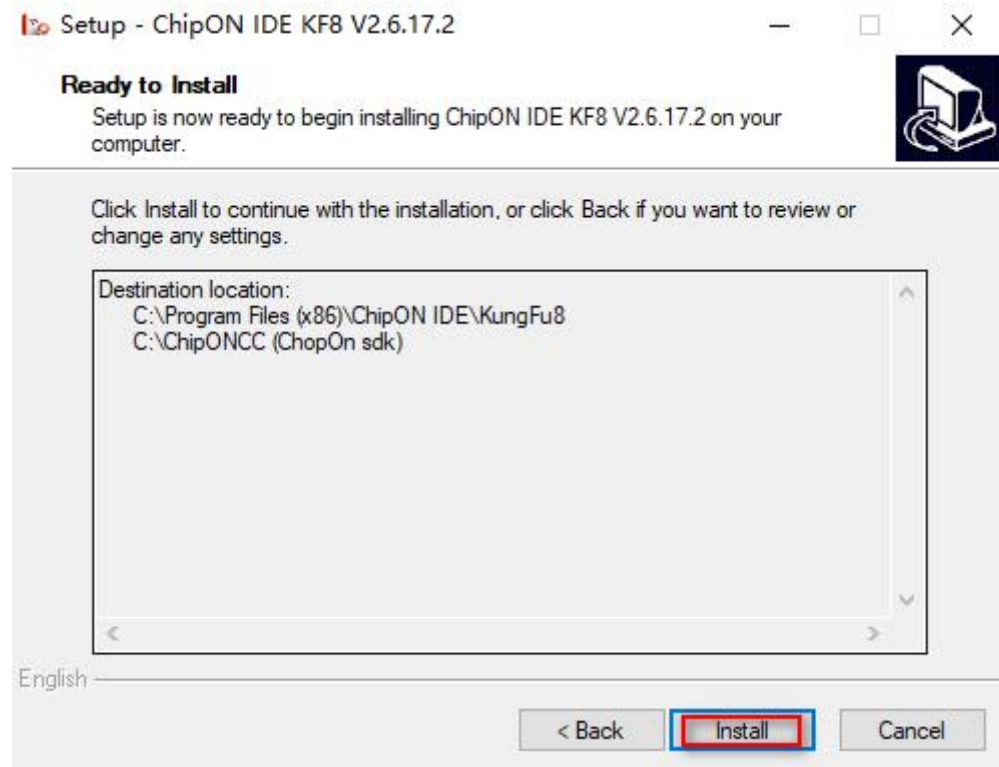
Set the location where the program shortcut is stored, and then click "Next".



Choose whether to create desktop and Quick Launch bar icons, and then click "Next":



Confirm the selected installation information and click the "Install" button to install ChipON IDE software.



After the installation is complete, click "Finish" to complete the installation process and then you can start running the ChipOn IDE.

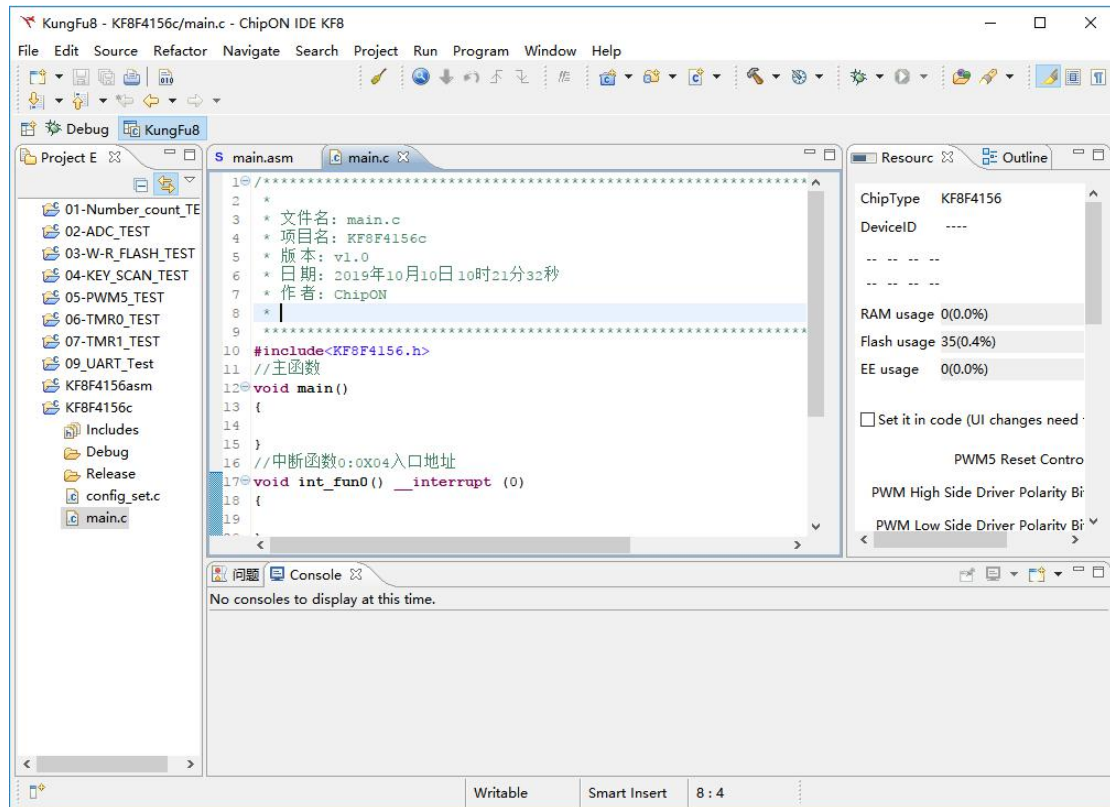
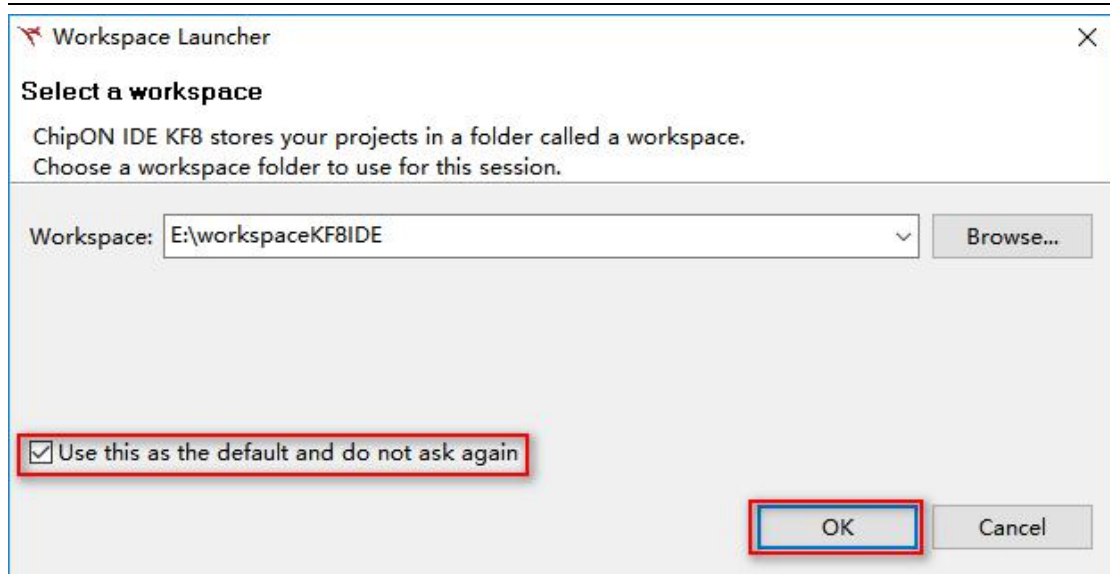


3. Start ChipON IDE Software

There are three ways to start ChipON IDE Program:

1. Start ChipON IDE by double-clicking the desktop shortcuts of ChipON IDE.
2. Click "ChipON IDE VX.X.X" in start menu of this computer to start ChipON IDE.
3. Find "chaponide.exe" in installation directory, double-click "chaponide.exe" to start ChipON IDE.

Before the ChipON IDE runs, you should need to select a workspace. This workspace is the source file path that determines the storage of the project you create. You can also set the "Use this as the default and do not ask again". And then click the "OK" enter the main development interface of ChipON IDE.

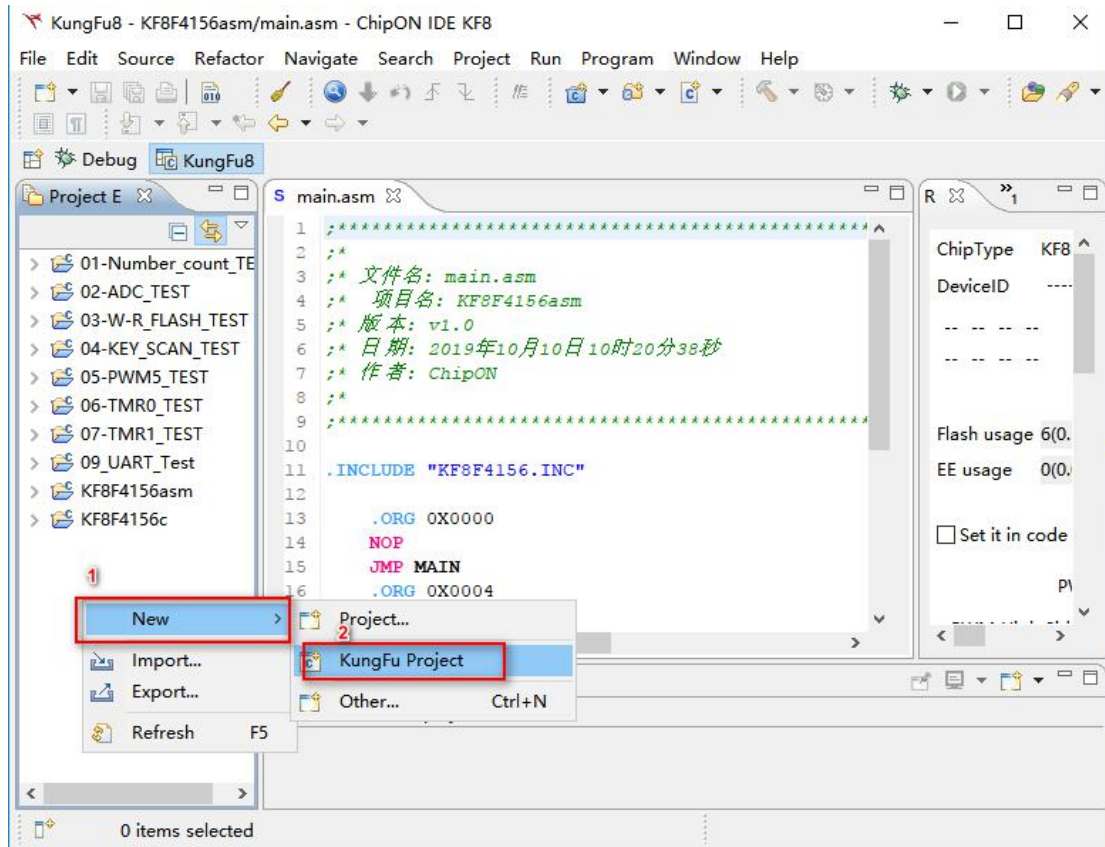


It should be noted that ChipON IDE adopts a spatial project management mode. There can be a single project or multiple projects in a space. Multiple workspaces can be established, but the history records only keep the last five spaces, and each workspace is required to be independent, i.e. no new workspace can be established under the space.

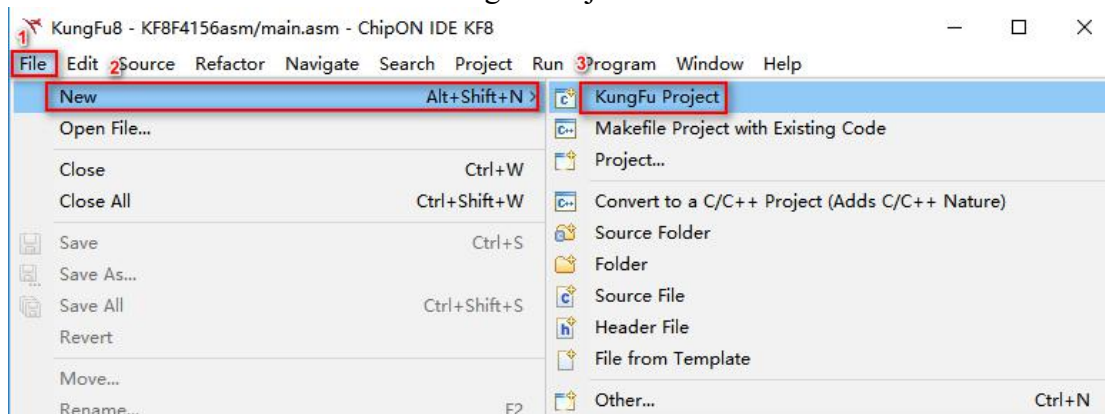
4. New Assembly Project

There are three ways to new project:

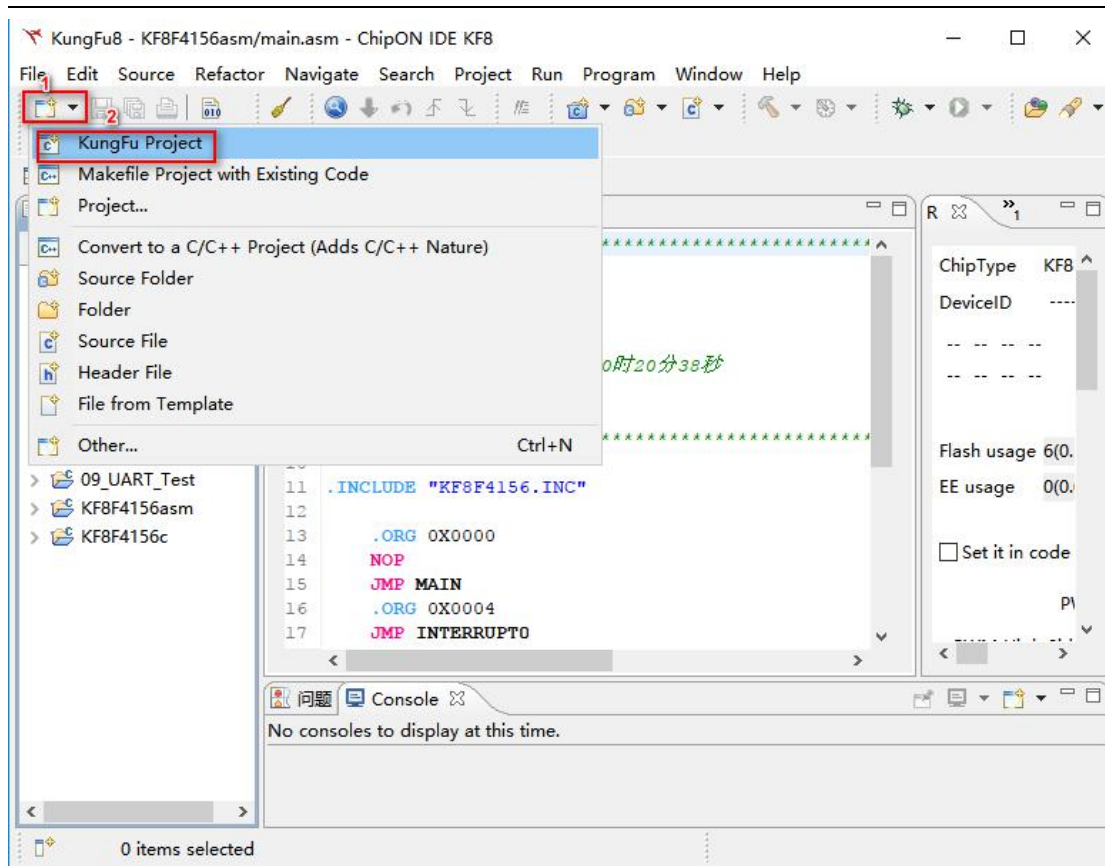
1. Right-click in the blank space of Project Explorer and select "New"→"KungFu Project".



2. Select "File"→"New"→"KungFu Project" in menu.

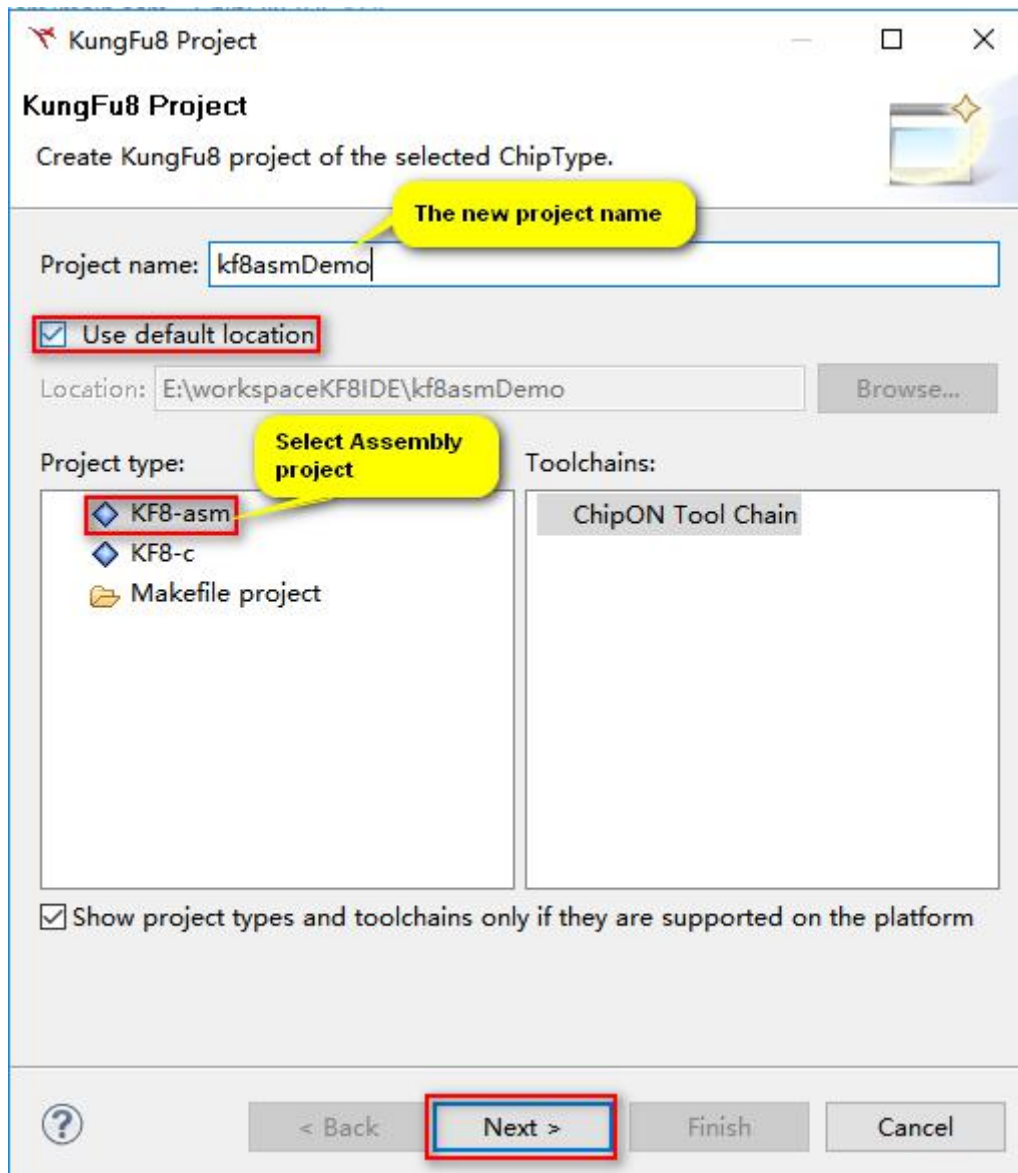


3. Click the "New" icon to new KungFu project.

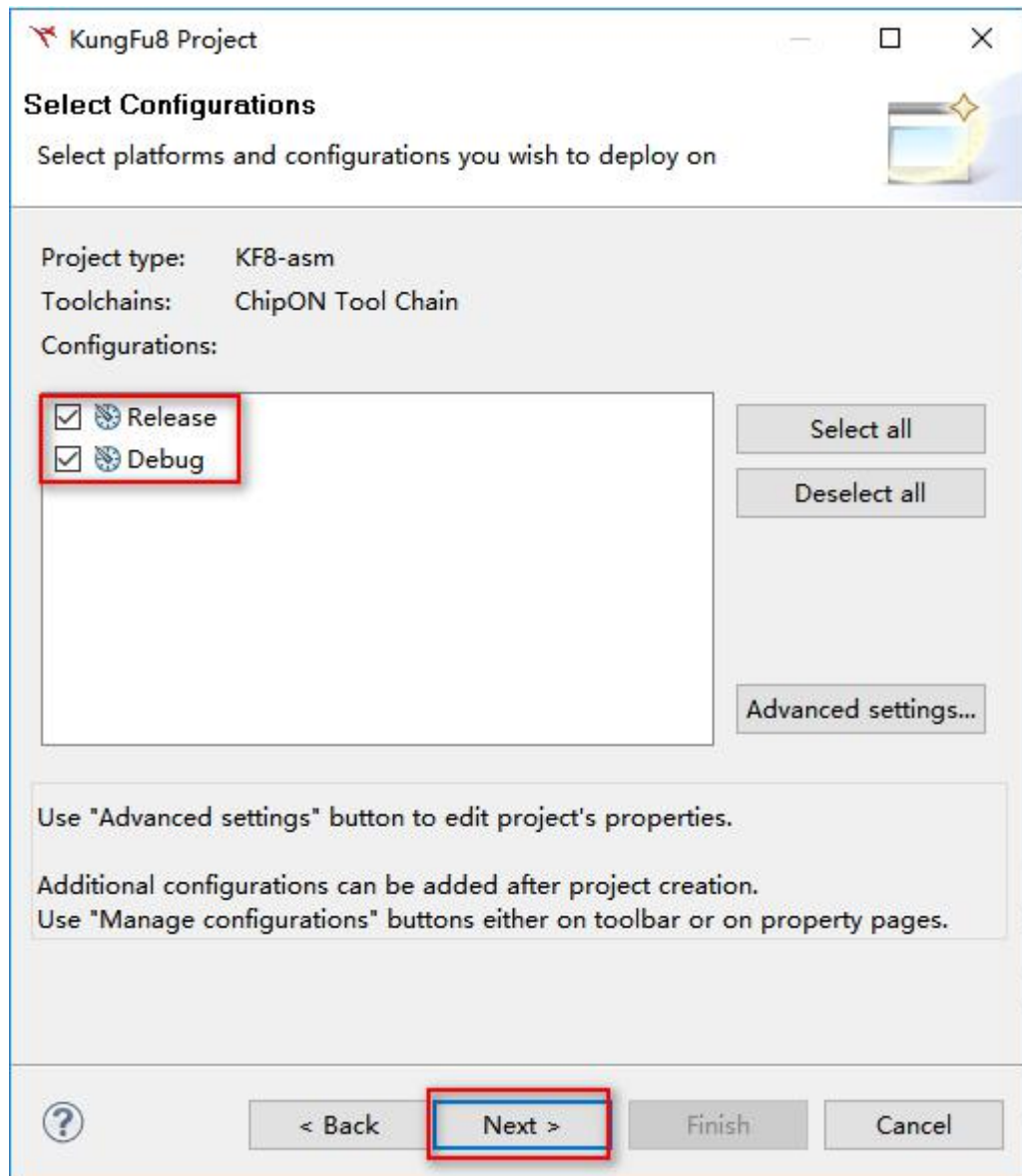


After Selecting "KungFu Project", Set the project name. The project location can use the default location or customize the location. It is recommended to use the default location, i.e. the project is created in the current workspace.

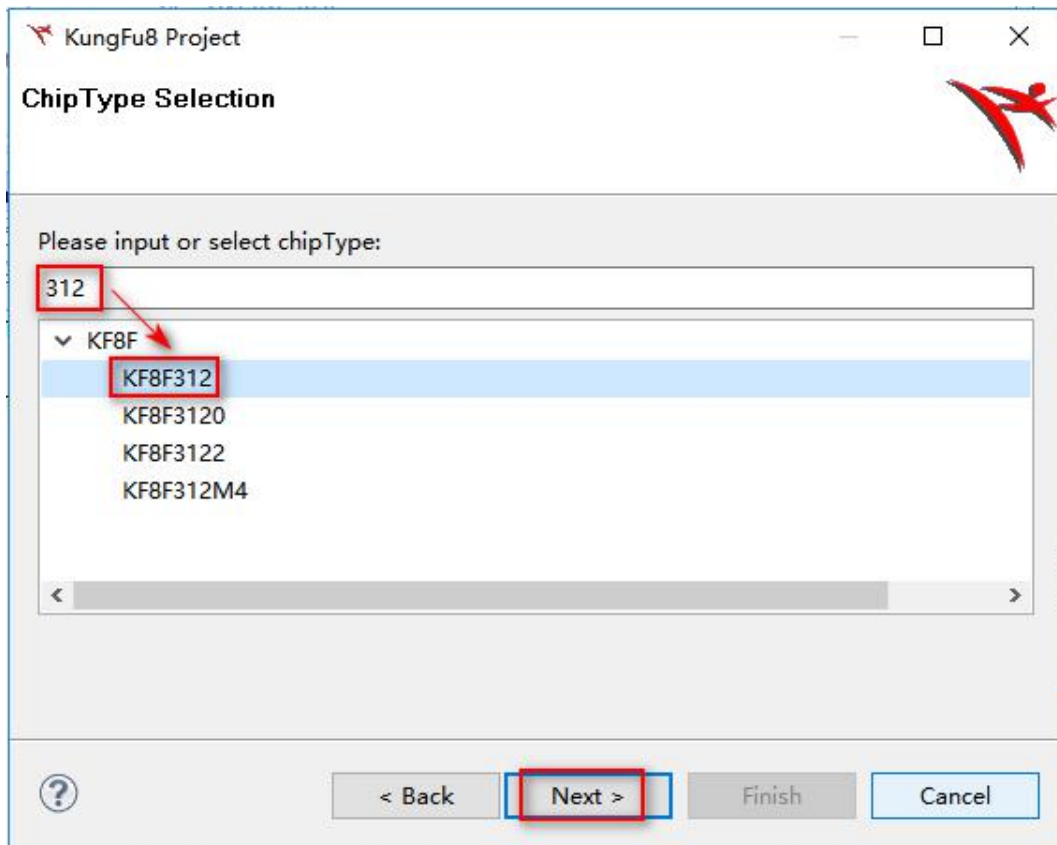
Select the type and tool chain of the new project, select "KF8-asm" in the project type, and then click "Next".



Next, select the platform and configuration to deploy, i.e. **Debug** in debug mode or **Release** in normal operation mode. Generally, by default, do not modify and click "Next".



Then, select the chip type to be used in the project. Here, you can select the chip from chip types list tree, or you can quickly filter through input all or part of the character of the chip type. For example, select "KF8F312", enter "312" in the input box, and then select the corresponding complete model from the filtered content in the following table.

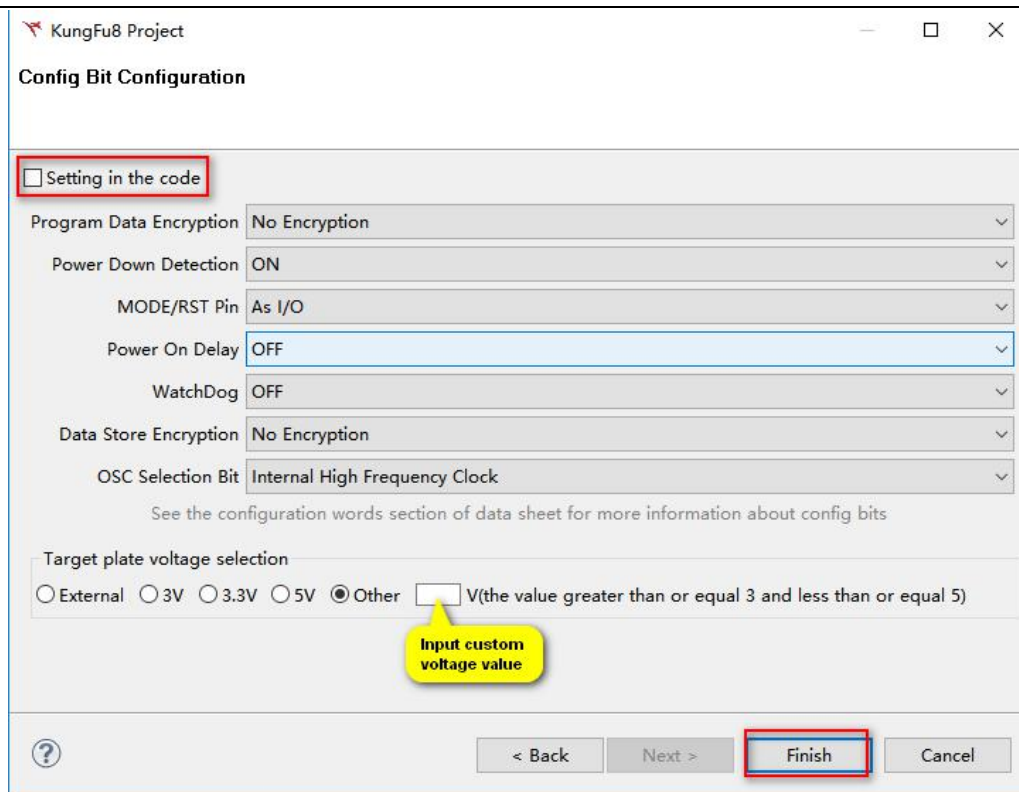


After selecting the chip type, the debugging voltage value selection and the configuration function of the auxiliary configuration word are provided. You can choose to set it in the code or use the special dialog box to configure it.

When you use the dialog box to create file "config_set.asm", the contents are as follows:

```
" __config 0x2007,0x06fd  
.end "
```

Where 0x2007 is the mapping address of the configuration word and 0x06FD is the configuration word result.



KungFu8 Project

Config Bit Configuration

☐ Setting in the code

Program Data Encryption: No Encryption

Power Down Detection: ON

MODE/RST Pin: As I/O

Power On Delay: OFF

WatchDog: OFF

Data Store Encryption: No Encryption

OSC Selection Bit: Internal High Frequency Clock

See the configuration words section of data sheet for more information about config bits

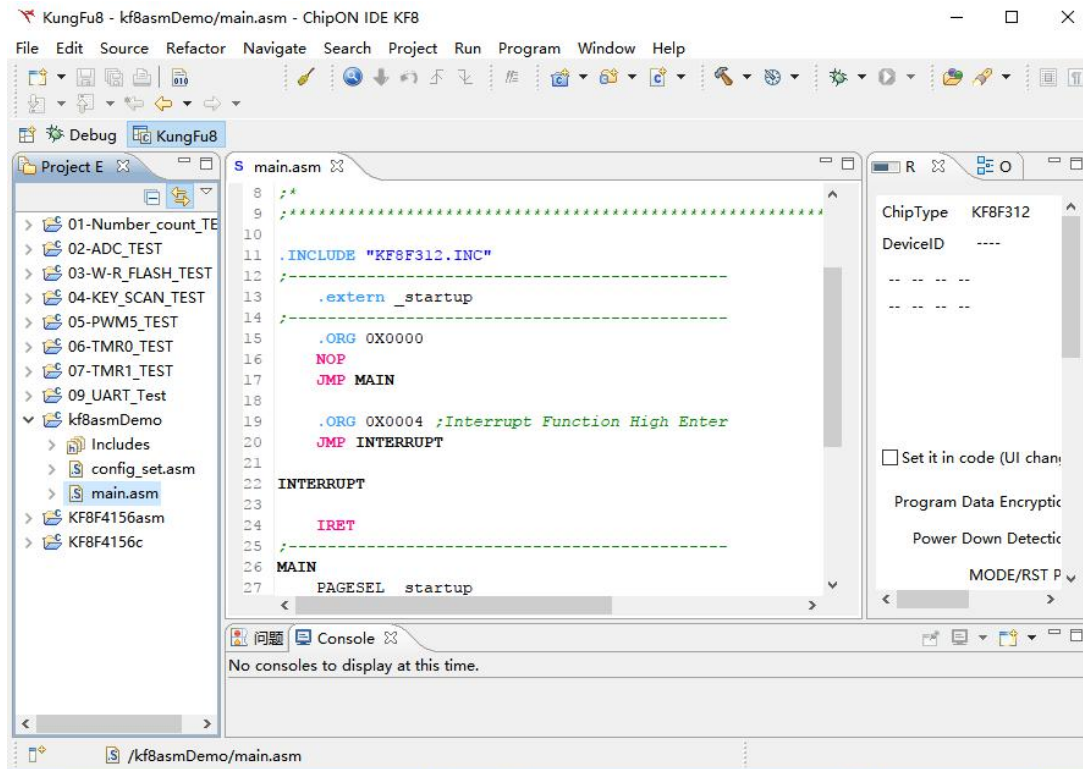
Target plate voltage selection

☐ External ☐ 3V ☐ 3.3V ☐ 5V ☒ Other V(the value greater than or equal 3 and less than or equal 5)

Input custom voltage value

< Back Next > **Finish** Cancel

Then click "Finsh", and a simplest test-asm project has been created. In new project, the basic frame code of the selected model is provided by default, including the entry expression of MAIN entry and interrupt.

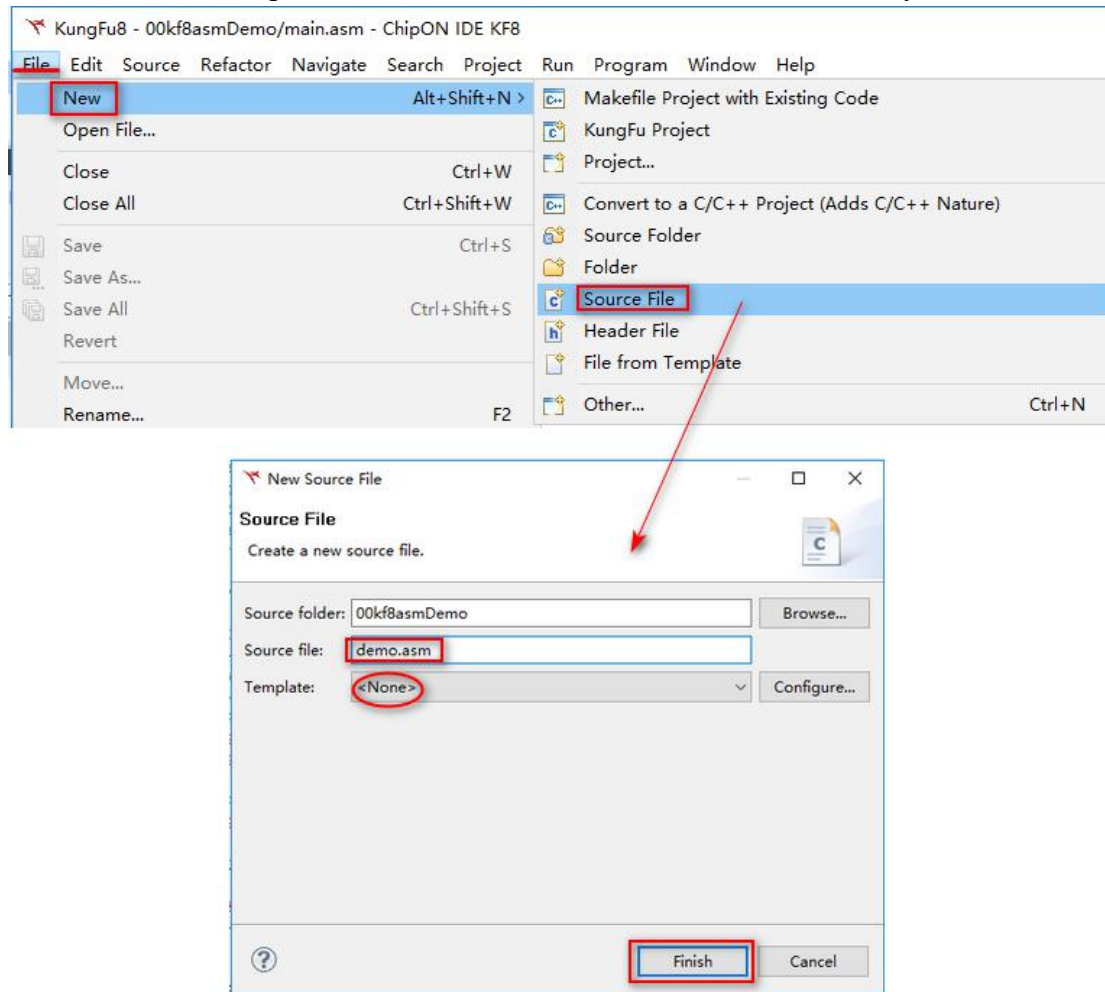


5. New Assembly Type Source File Code and Compile

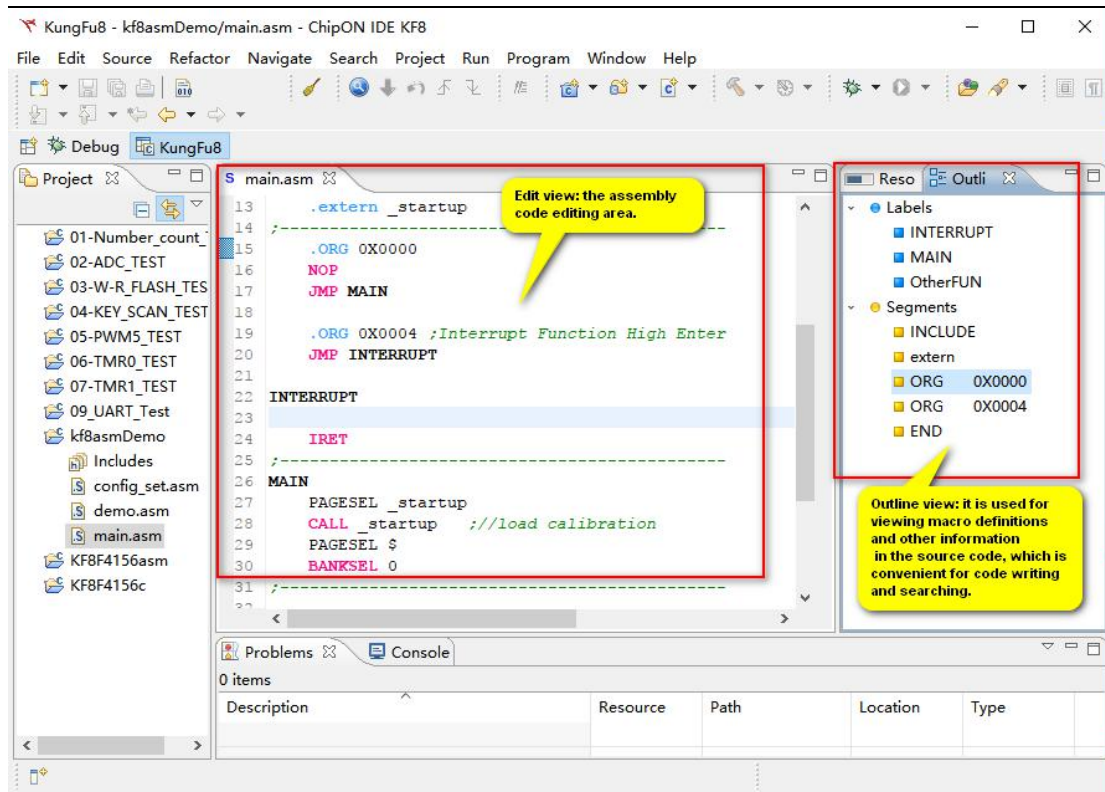
1. New File

Select project and right-click, "New"→"Source File", then set source file name on the page of the new source file.

Note: The "Template" item select "<None>" when create assembly source file.



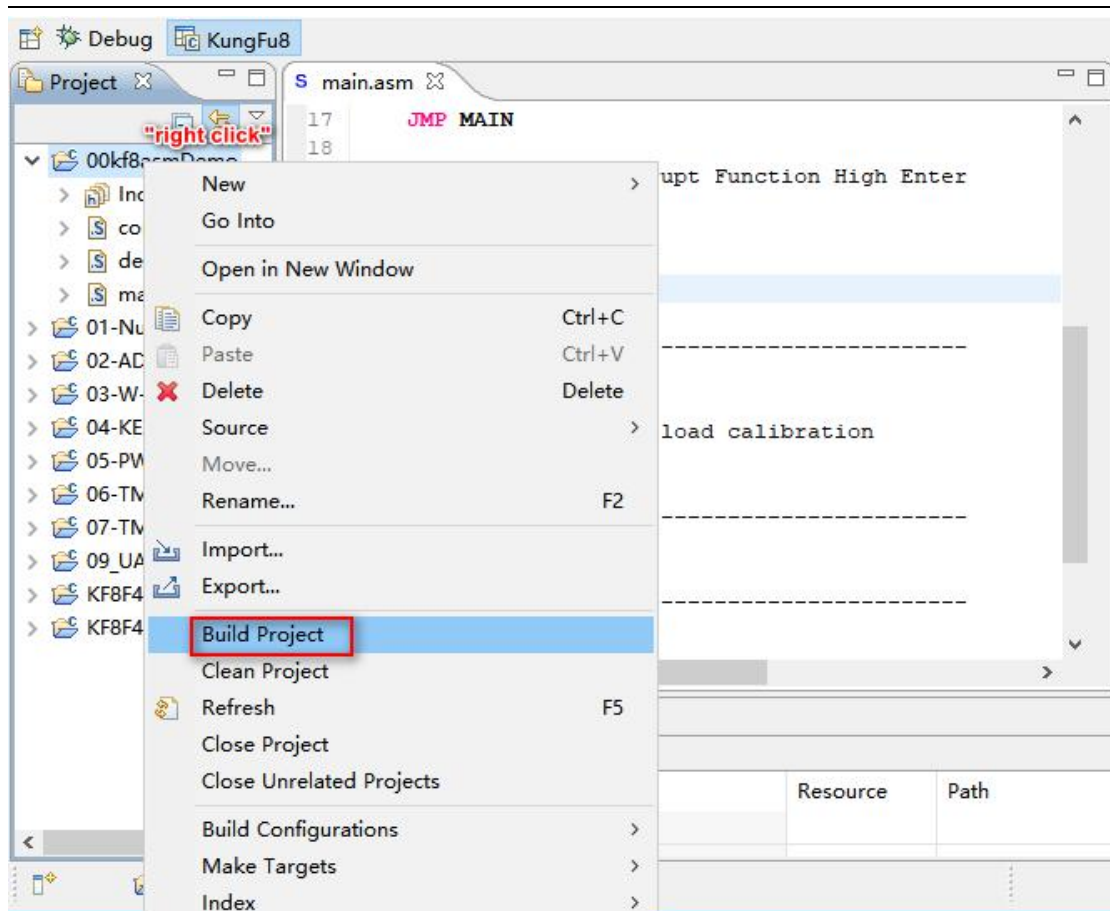
After creating the new assembly file, you can write code to this assembly source file in Editor view. As shown in the figure below. Moreover, it can also create a assembly header file, and the suffix of the assembly header file is ".inc".



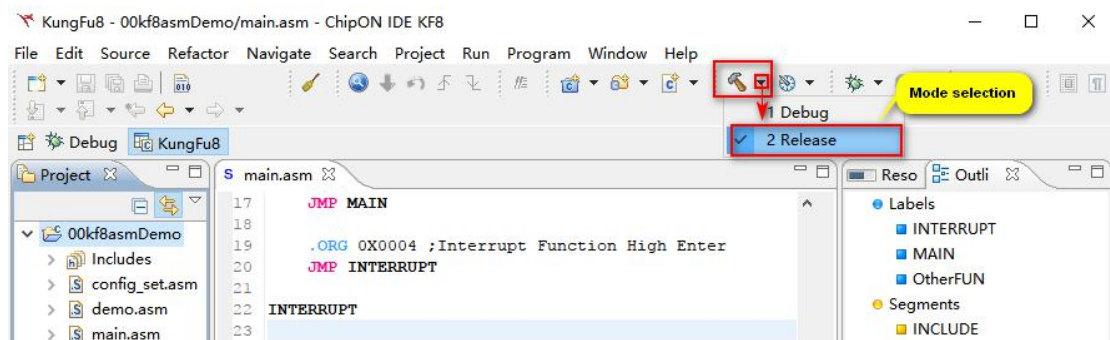
2. Compile source files

Note: The project should be selected first. The project can be selected in the Project Explorer or the corresponding code file.

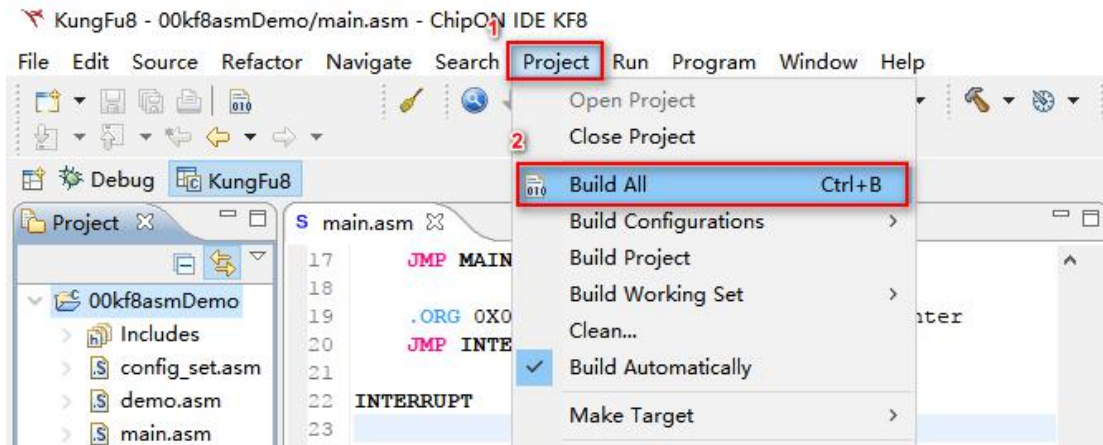
Method 1. Right-click the project, and then click "Build Project":




Method 2. Build project through the shortcut toolbar, where compilation will be performed according to the selected mode, and the modification mode can be selected through the drop-down triangle.



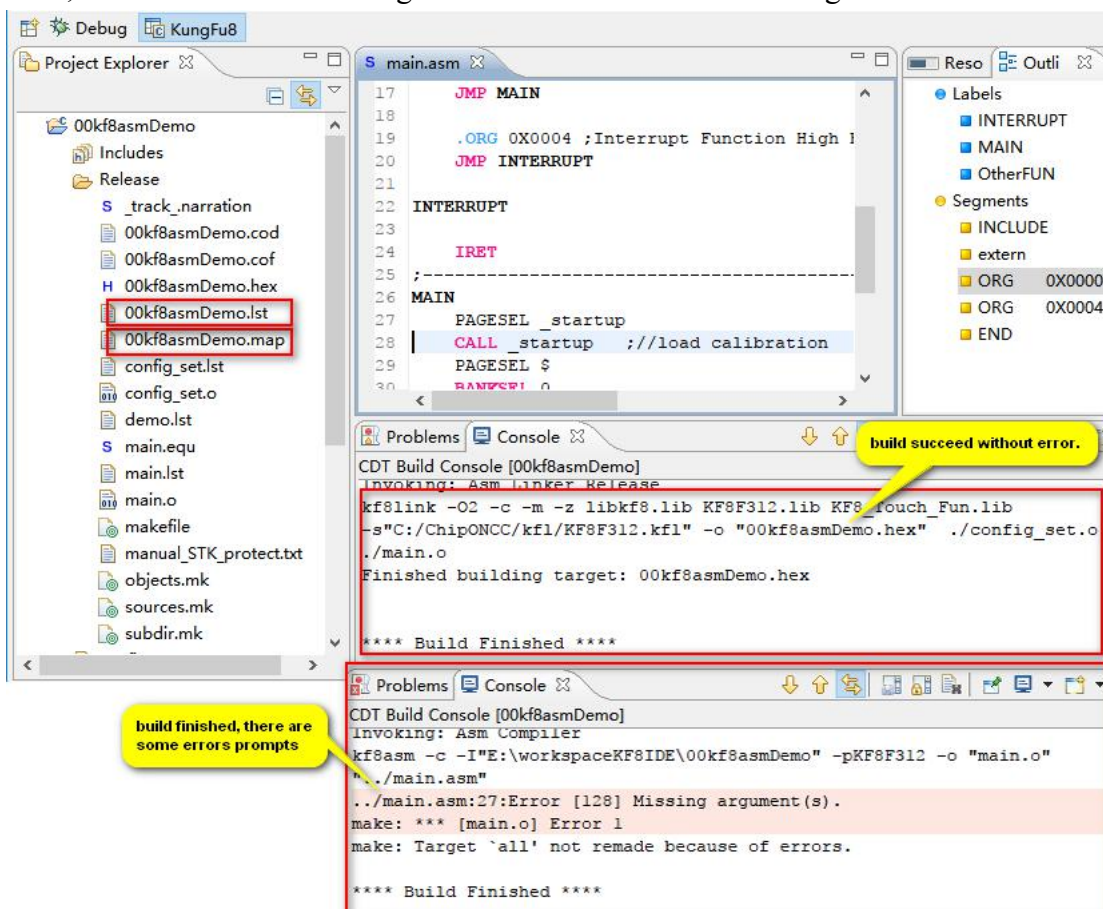
Method 3. Click "Project"→"Build All" menu, or use the shortcut key "CTRL + B". This moment all items in the workspace will be built.



Method 4. Click "Build All"  in the shortcut toolbar. At this time, all projects in the workspace will be built.

After the compilation is completed, in the "Release" or "Debug" directory, a corresponding hex file is generated, including an ".lst" file associated with hex and a ".map" file of the usage of chip program segments and data segments.

If there is an error in the code, an error message will be prompted in the console view, note that the error message is not limited to the red file flag.



6. New C Project

There are also three ways to new C project that is the as creating assembly project. You just need to make sure that select "KF8-c" during the new C project process, and all other process of new c project is the same the process of new assembly project. Please refer to [4. New Assembly Project](#) for detailed procedures.

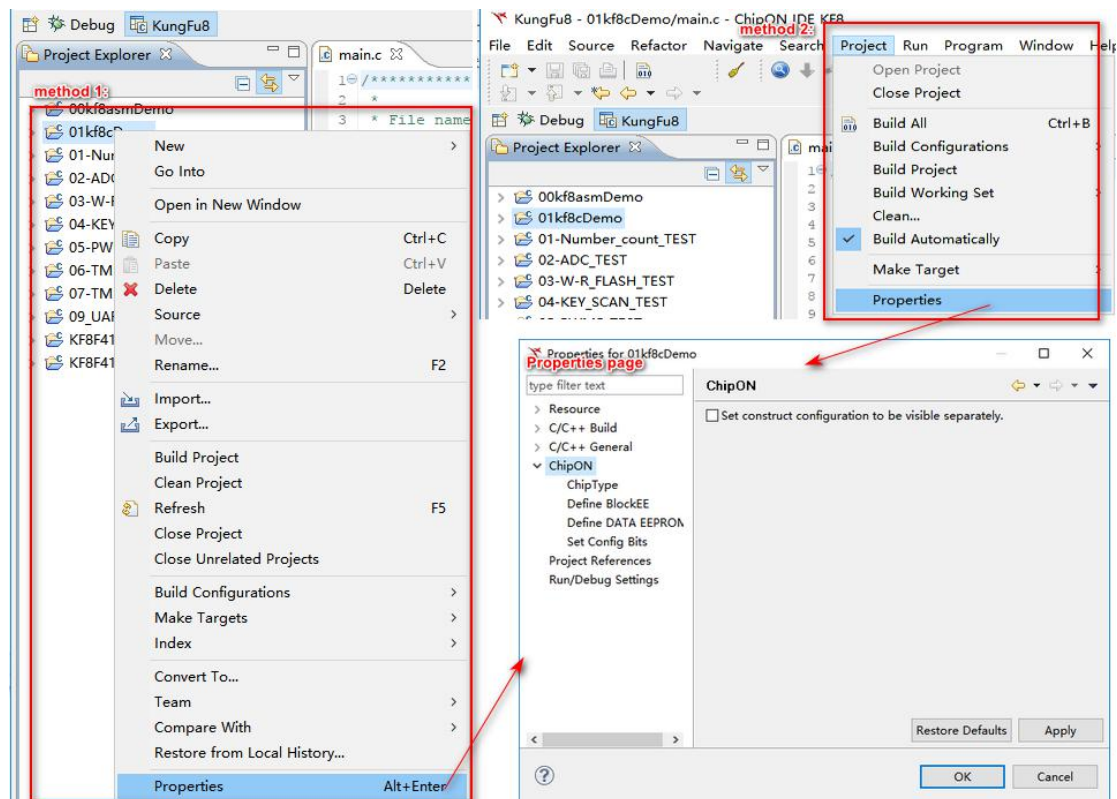
7. New C Type Source File Code and Compile

The process of creating C type source file code and compile refers to [5. New Assembly Type Source File Code and Compile](#).

8. Configure Chip Configuration Bit

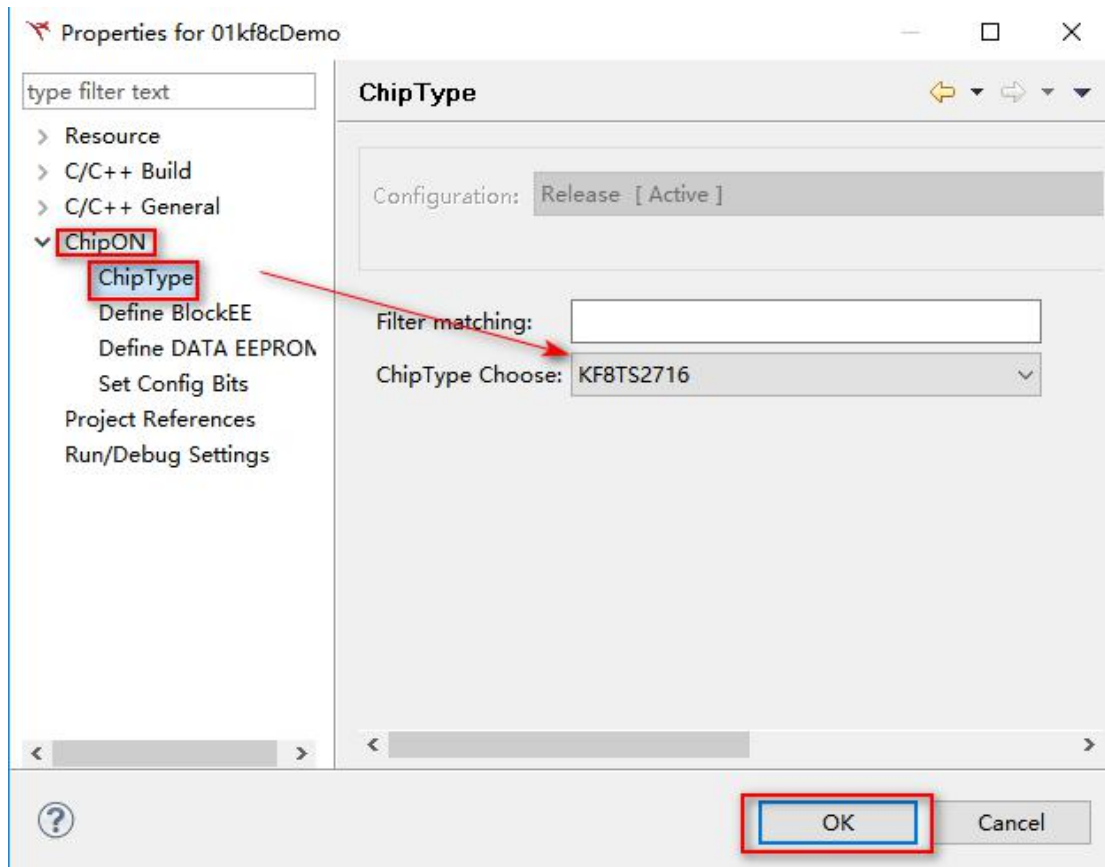
(1) Open Configuration Page

Right-click the selected project and select "Properties" to enter properties page, or select project and enter properties page by clicking "Project"→"Properties" in menu.



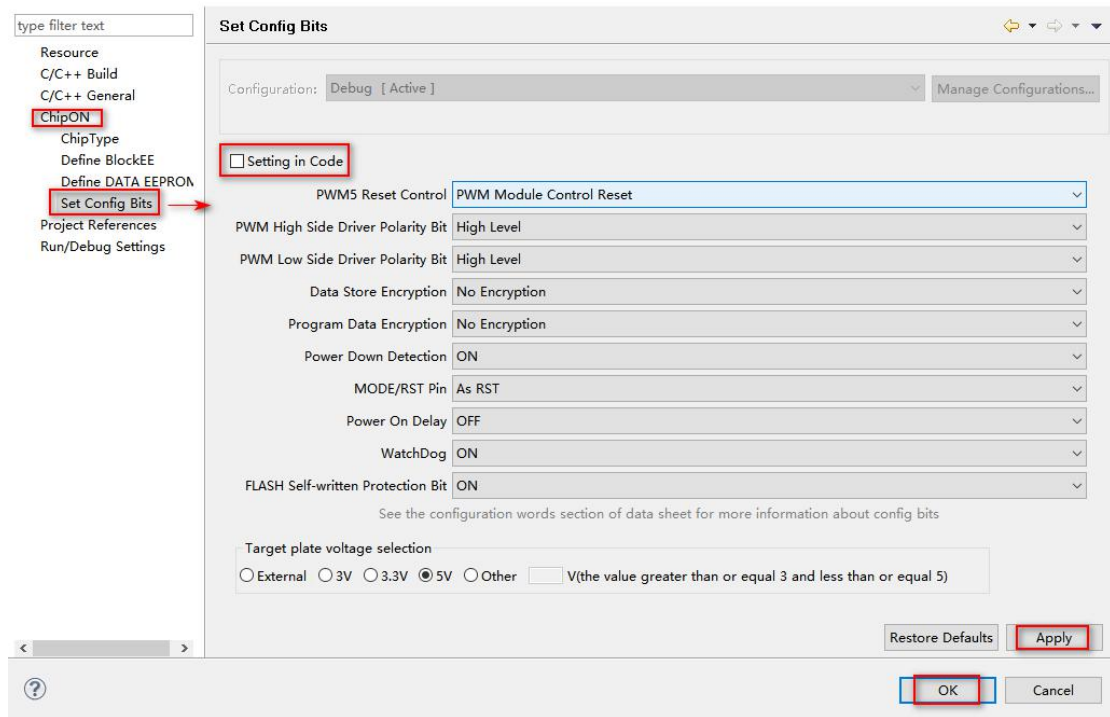
(2) Set the ChipType

In properties page, select "ChipON"→"ChipType", choose chipType in the right of the properties page. As shown below.



(3) Set Configuration Bit

In properties page, select "ChipON"→"Set Config Bits", unchecked "Setting in code" option and configure the parameters as needed, then click "Apply" or "OK". As shown below.



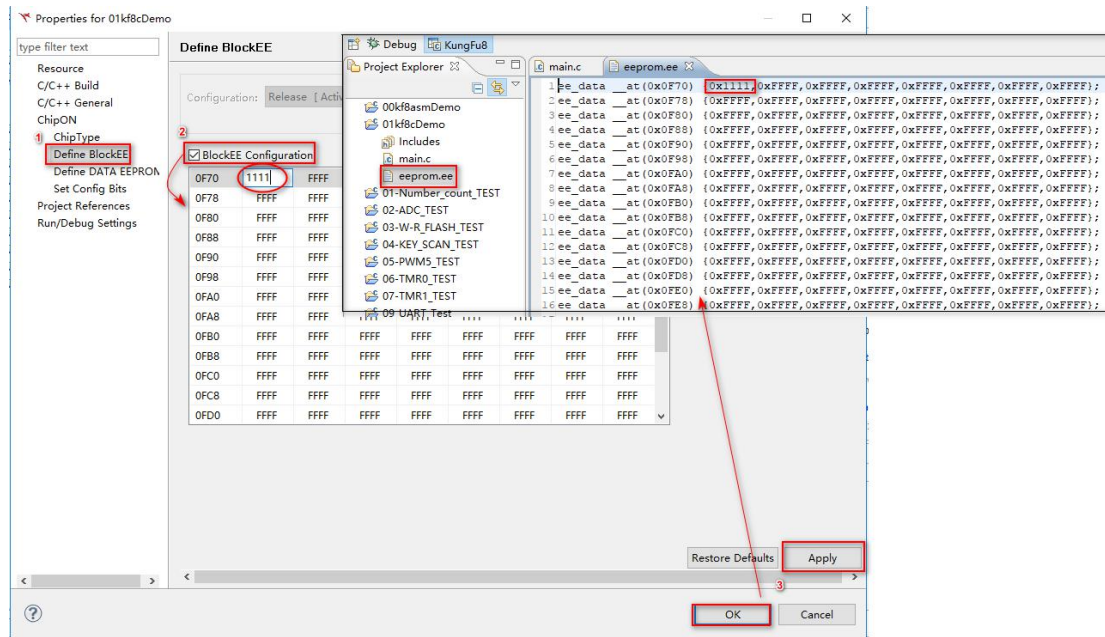
Finish configuration above, the file "config_set.c" or "config_set.asm" will be generated in the project root directory.

If you check "Setting in code" option, the file "config_set.c" or "config_set.asm" will be deleted. And you can use configuration items of configuration word to generate configuration word directly in "Resource Utilization" view.

(4) Configure BlockEE of Chip

In properties page, select "ChipON"→"Define BlockEE", check the "BlockEE Configuration" option and BlockEE view is displayed on the right side of the properties page.


Double-click the data in the BlockEE view to modify the selected data, then click "Apply" or "OK" to enable modified data to take effect. And the "eeprom.ee" file will be generate in the root directory of this project. As shown below. If uncheck the "BlockEE Configuration" option, the "eeprom.ee" file will be deleted.

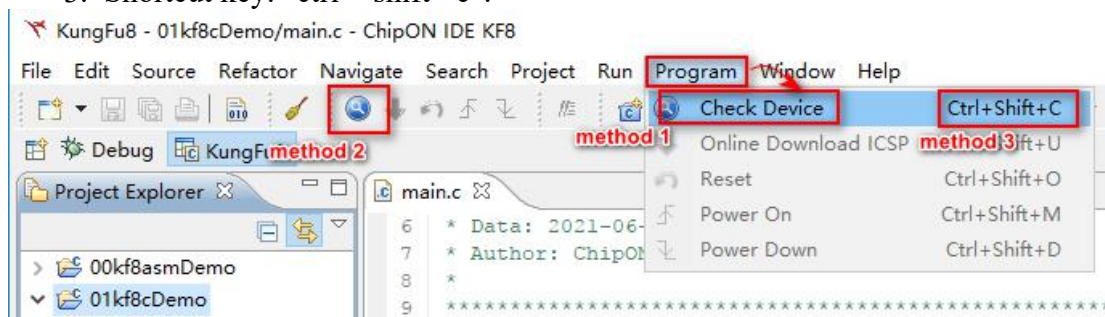


9. Download Hex File

(1) Check the equipment


Checking the equipment is required when the programmer is not recognized for the first time. It can check the connection of the hardware device. There three ways to execute this operation:

1. Click "Program"→ "Check Device" on the menu.
2. Click the icon  on the toolbar.
3. Shortcut key: "ctrl + shift +c".



(2) Online ICSP Download

After checking the device successfully, then start downloading program by using one of the following three ways.

1. Click "Program"→ "Online Download ICSP" on the menu.
2. Click the shortcut toolbar icon .
3. Shortcut key: "ctrl + shift + u".

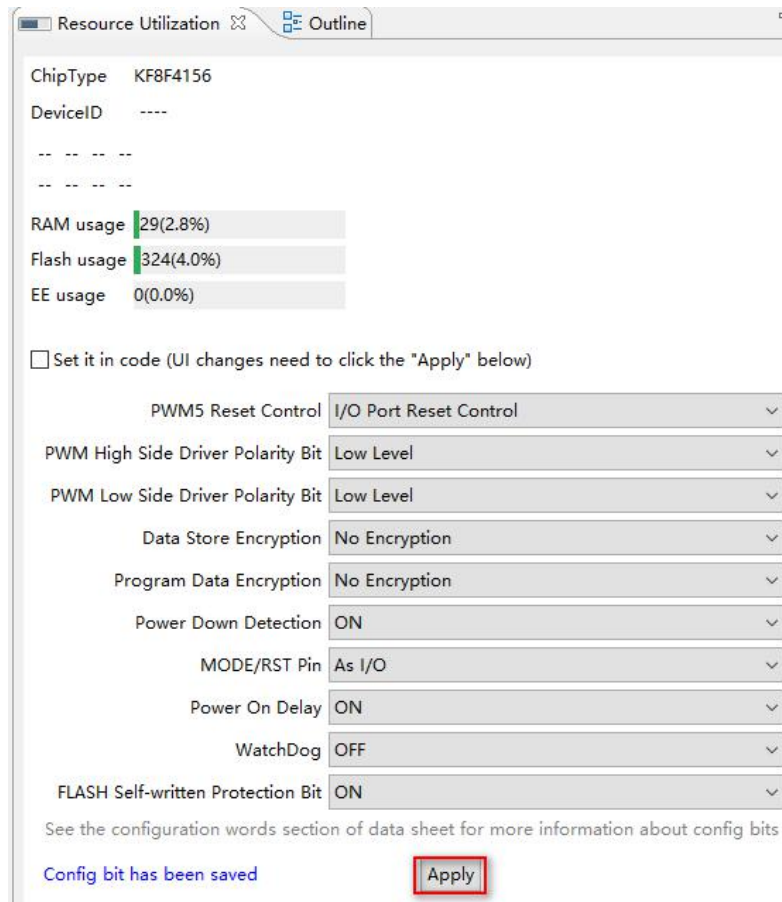


The downloading information is displayed in the "Console" view, including programming start, programming succeed or programming failed information.

10. Resource Utilization

When you need to view the resource usage of Flash or BlockEE in the current project during the development process, you can view it through the "Resource Utilization" view or set the configuration bit in the "Resource Utilization" view.

If the usage rate exceeds 95%, the usage value will be display in red font as prompt. When the configuration bit information is configured, it needs to click "Apply" for the configuration to take effect.



It should be noted that "Resource Utilization" view page is also responsible

for providing configuration bits. When this view does not open, the new project will be abnormal.

11. Write Back the Calibration Values

When the program is interrupted during downloading, if the prompt "LDO calibration value and clock correction value have been lost, please write back the calibration value or replace the chip", please click the "Write Back Calibration Value" button to write the calibration value saved in the program into the chip.

This function has been added to the program execution process, and no additional operation is required. At the same time, the write-back function is only valid after reading the valid information of the chip. When switching project, this information will be lost, i.e. the original calibration value is not maintained, that is, the write-back function cannot be supported.

12. Project Development and Debugging

1、Code development

In the newly established newest template file for code writing operations. Different folders can be created to distinguish the purpose of the code. However, for computer differences, this feature may directly report the assembler crash exception. At this time, the header file and source file need to be placed under the first-level path and delete the defined folder.

The code font setting can be viewed in Chapter 15 to change the editor font implementation.

It is not recommended to change the project name frequently. This operation includes information store, Once the process is abnormal, the information will be inconsistent. At this time, the compilation may be abnormal, or the compilation result cannot be refreshed in time, resulting in the existence of hex not being recognized when downloading the program.

You can perform a refresh operation on the directory, or the shortcut key F5. This exception suggests creating a new project, and then copying the original project file directly to the new project.

More support features can be found in Chapter 15 Tips.

2、Matters needing attention in debugging project

(1) Debug Configuration

P0.0 and P0.1 must be reserved for debugger use and need to be set as digital input ports. The MODE pin needs to be enabled, and P0.0 and P0.1 cannot be

used as program functions. They include port operation, port pwm output and port AD function.

(2) Embedded assembly

Assembly code can be embedded when debugging, but not all functions are assembly code. Embedded code cannot be viewed by the process, and will be run by the process as an additional existence of the previous C expression. As follows:

```
void init_chip()
{
    __asm
    CALL    0XFFF
    MOV     OSCCAL0,R0
    CALL    0XFFE
    MOV     OSCCAL1,R0
    CALL    0XFFD
    MOV     VRECAL,R0
    __endasm;
}
```

It needs to be modified to the following form:

```
void init_chip()
{
    volatile unsigned char i=0;
    __asm
    CALL    0XFFF
    MOV     OSCCAL0,R0
    CALL    0XFFE
    MOV     OSCCAL1,R0
    CALL    0XFFD
    MOV     VRECAL,R0
    __endasm;
    i=1;
}
```

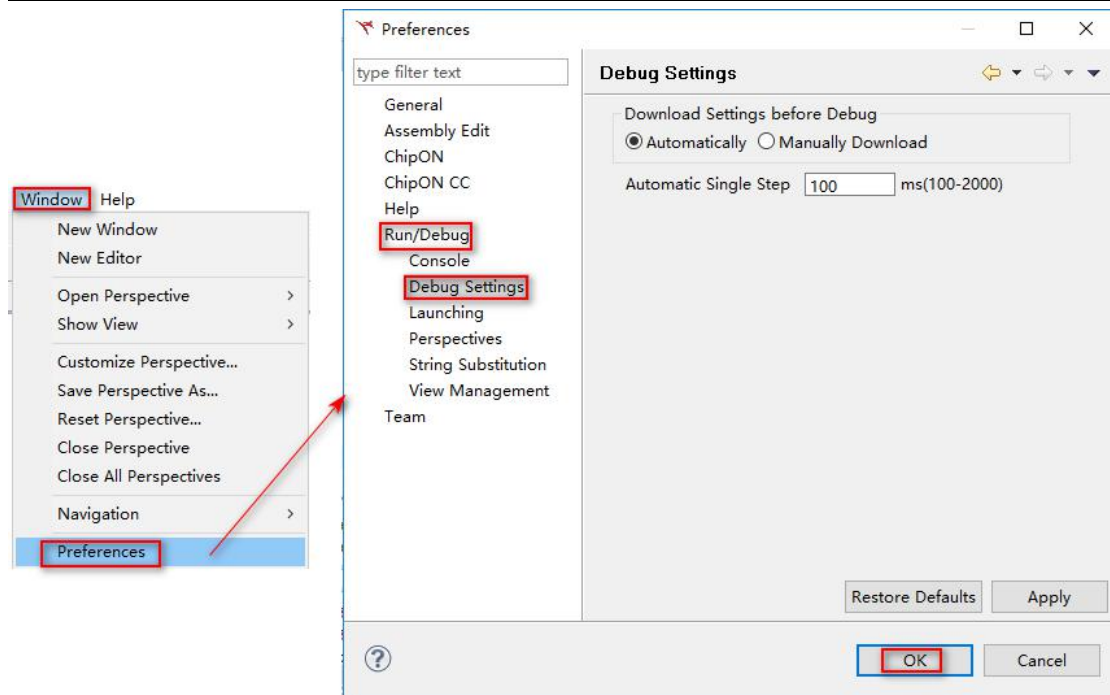
Reason: The coff file generated by the embedded assembly code compiler does not have a line number. When the whole function does not have a line number, the current line number cannot be found after stepping in.

- (3) Program return instructions such as CRET, IRET and RRET cannot be embedded in assembly code. Single-step skipping and single-step jumping will analyze the machine code, and a function can only have one return instruction. The reason is that the exit traversal code of a function finds CRET, RRET, IRET, and breaks the point here to run the function. The early return may not hold, and the exit function intention operation will fail.

- (4) When the chip is debugged, the overhead of monitoring code and some variables will be added. The RAM occupies less than 16 bytes and the program address occupies less than 250 words. The debug variable address is fixed and cannot conflict with the address defined by the program display.
- (5) This software supports bit structures and unions, but the recognition degree of structures and unions is not friendly enough. For example, the variable values in the debugging interface are checked the dislocation and cannot be monitored. At the same time, it is not recommended that structures and unions objects participate in complex operations, otherwise dislocation may occur.
- (6) It is not recommended to use pointers and floating points in design projects. These two types of code are inefficient to compile and take up more program space.
- (7) The IDLE directive cannot be used during debugging, that is, sleep conditions cannot be simulated.
- (8) Watchdogs cannot be used during debugging.
- (9) Functions cannot be written as static.
- (10) Special symbols, such as parentheses, cannot exist in the project name. The project space does not recommend a path that is too deep and a name that is too long.
- (11) The project you are debugging does not support defining static functions.
- (12) Debugging only supports hardware online simulation. When the programmer connection is missing, debugging and debugging view adjustment are not performed by default.

3、 Debug Settings

Select the "window"→"Preferences" on the menu, select "Run/Debug"→"Debug Settings" in Preferences, select "Download Settings before Debug" in the right page, and fill in the automatic single step interval value in the text box, as shown below.



Automatically Download: indicates that the project will be downloaded to the chip before entering debug mode.

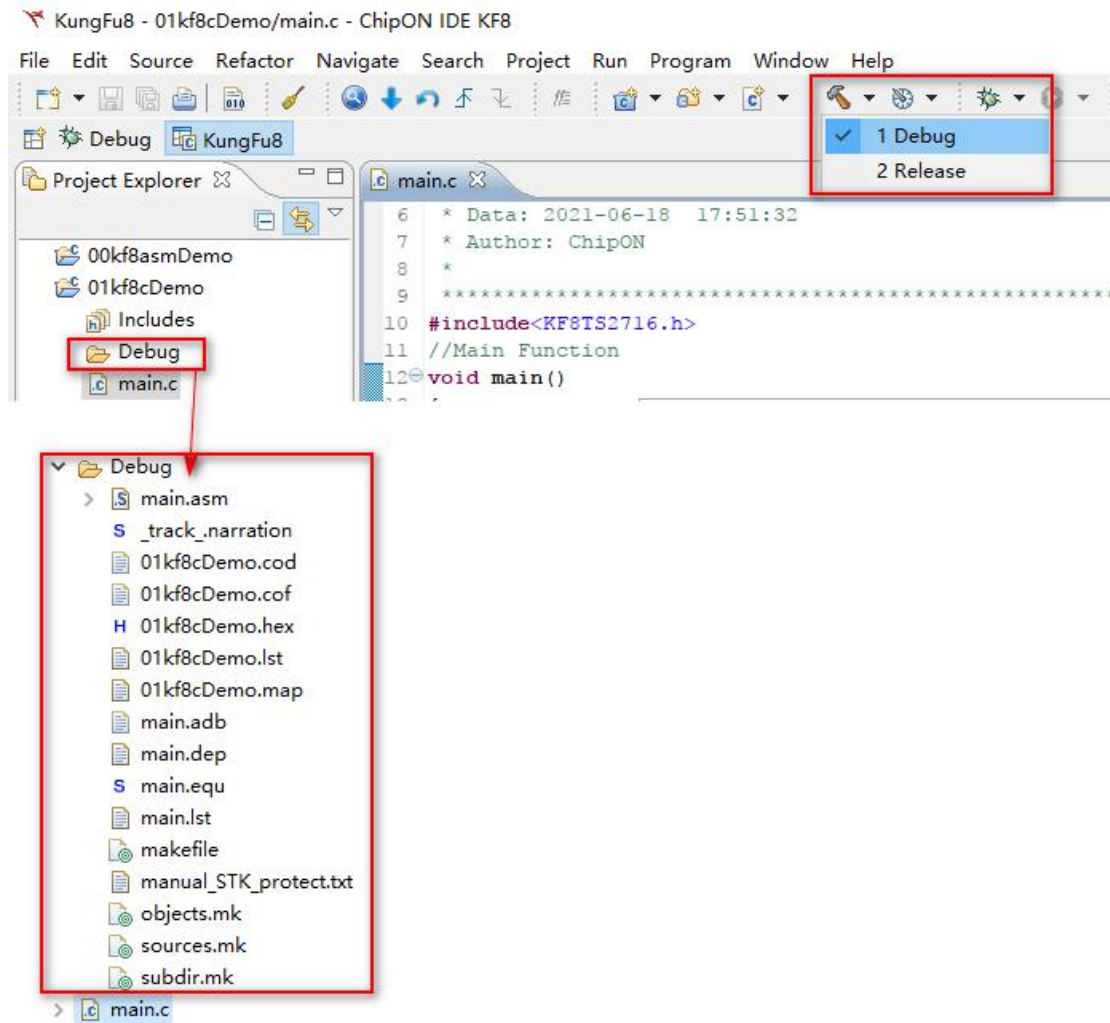
Manually Download: indicates that the project directly enters debugging mode.

Automatic Single step: indicates the time a project waits after completing a step.

4、Compile project

Note: The project should be selected first, and the compilation mode should be "Debug" mode.

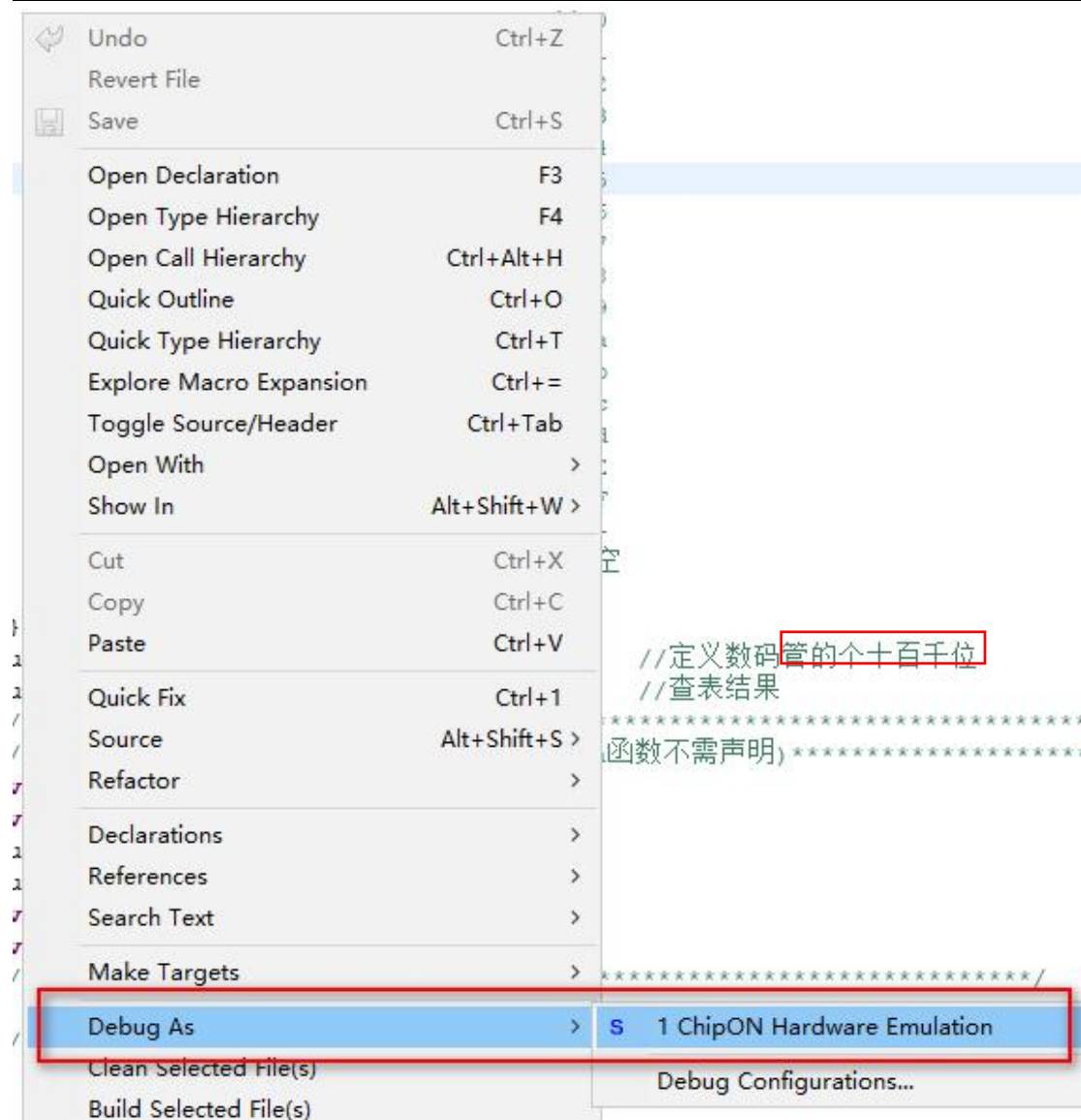
Build project through the shortcut toolbar. After the compilation is completed, add the "Debug" directory and the files under it. As following figure.



5、Download hex file

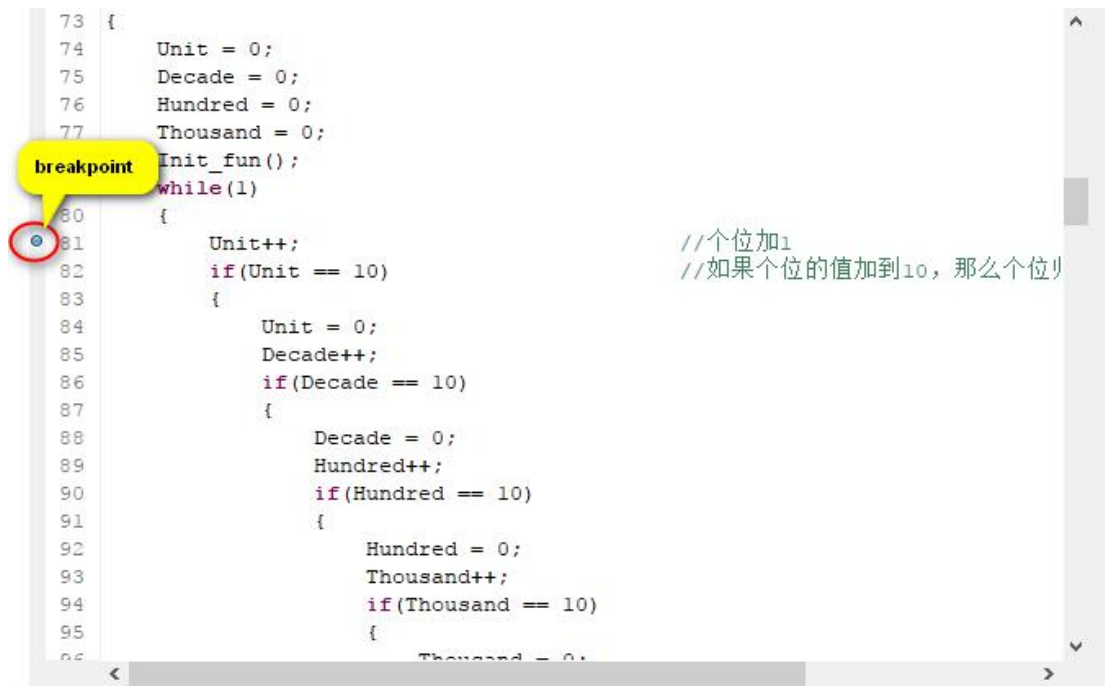
When manual download mode is selected, program download needs to be performed separately. When automatic mode is selected, the download of the program will be completed automatically during startup debugging. Enter Debugging Method: Right-click the file with the main function entry, select "Debugging Method (D)" and select "ChipON Hardware Emulation". Then the tool executes the download of the program (automatic download mode) and the debugging function starts.

The console view displays information about the procedure, including startup information and exception information.



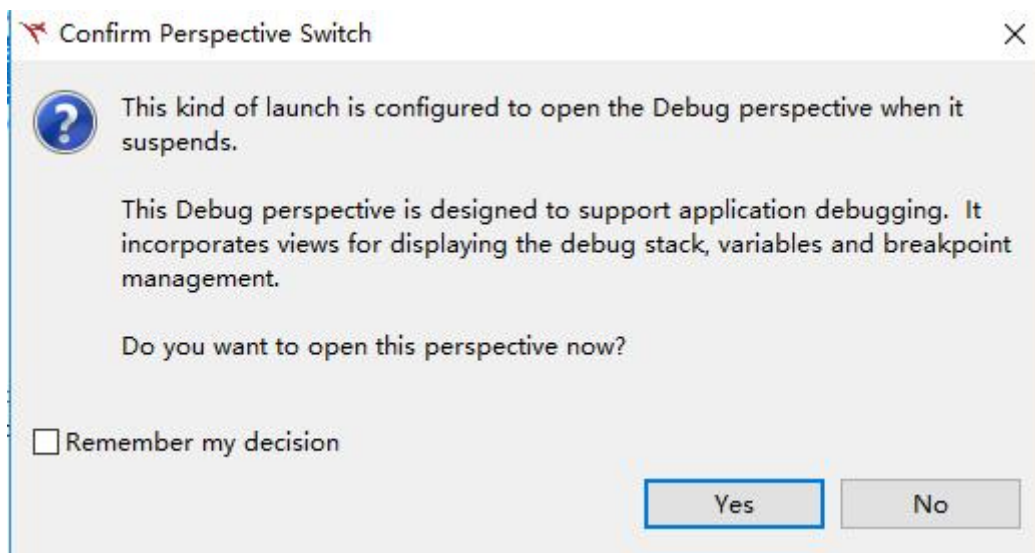
6、Set Breakpoint

Double-click the column on the left side of the editor to add or delete breakpoints. Currently, only one breakpoint can be set. At the same time, breakpoint information is stored as item information, but there may be inconsistencies. At this time, breakpoint reset is required.



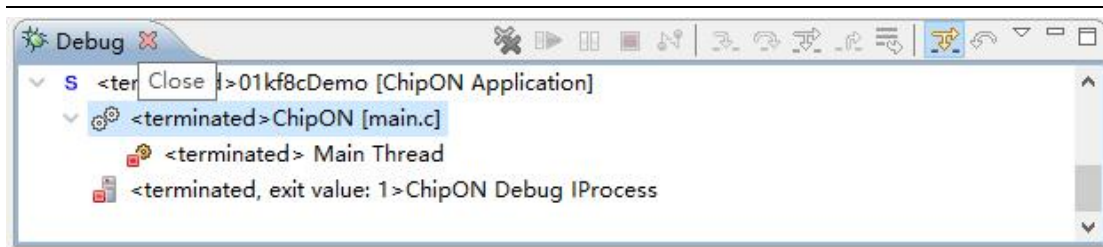
7、Enter Debugging


After starting to run the program, a prompt box will pop up before entering the debugging mode. Select whether to enter the debugging perspective, as shown in the figure. You can decide by default or click "Yes" to enter the debugging view.



8、Debug View


1) Debug View: Displays thread information when the project runs.



: Remove all terminating threads

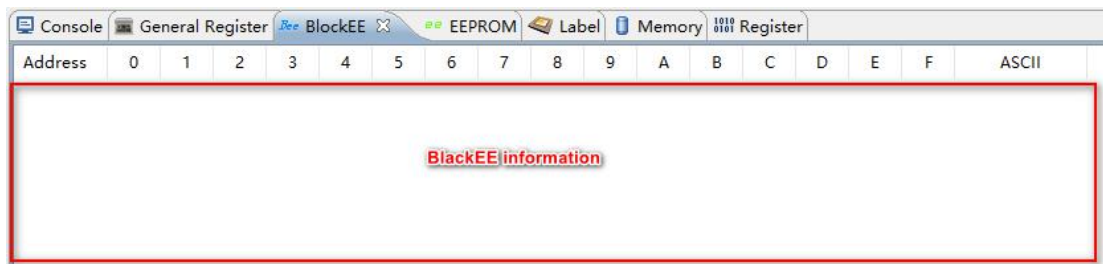
: Continue

: Pending

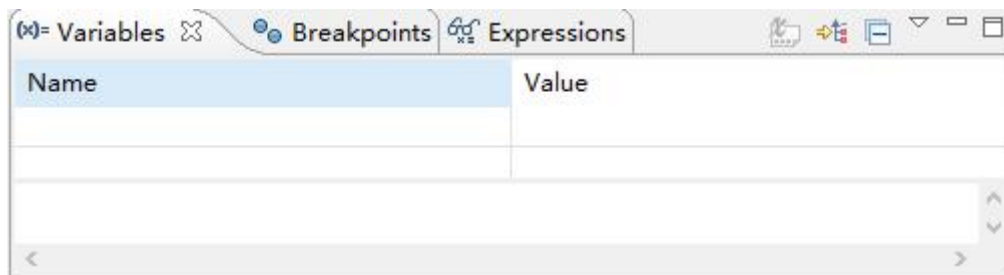
: Single Step in, Single Step Skip, Auto Single Step, Single Step Return

: Reset

2) BlockEE View: display the BlockEE information in the chip, which cannot be modified and needs to be opened.

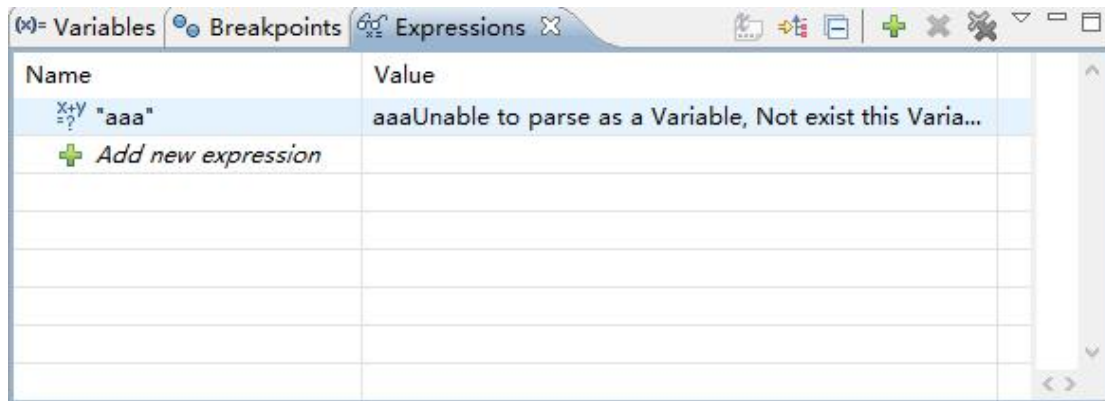


3) Variable View: Display local variables in the project, and the value can be modified. For variables that cannot be added through operation, automatically identify variables in the current code field and display the results. This function has certain limitations, and the recognition situation is not good enough.



4) Expression View: Display the value of any variable or expression that the user wants to see. It is an important monitoring view, which is not opened by default, needs to be opened manually, or after selecting variables in the code, select Add to

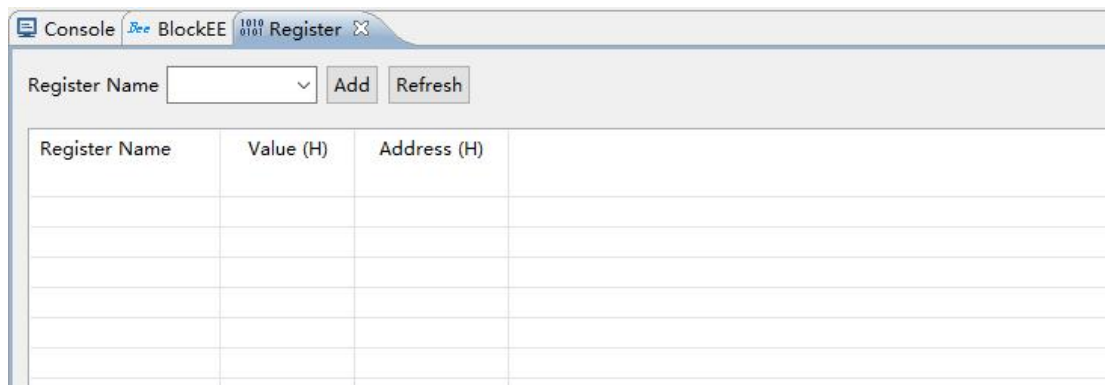
expression through the right-click menu.



5) Breakpoint View : Manage all breakpoints in the project. Double-click the breakpoint content to quickly locate the code.



6) Register View: Display the information of registers in the chip, which can be modified except for special addresses.



7) Memory View: Display all the information in the chip, which can be modified except for special addresses.

Console

BlockEE

Register

Memory

First Address(H)

0

Address Length(H)

40

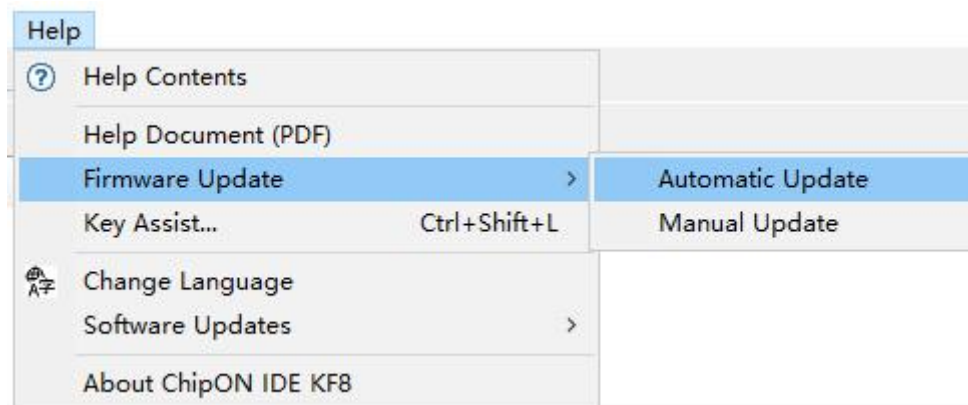
Search

Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
000	9E	FF	49	1A	04	38	FD	FF	8F	9E	1F	00	00	00	00	00	..f.8.....
010	00	00	00	EF	EF	00	FF	00	00	00	00	80	7F	00	30	000.
020	80	FF	00	00	00	FC	FF	FF	60	00	70	00	00	00	10	20`p....
030	50	00	FF	CF	CF	FB	00	01	00	00	00	00	08	9E	E9	00	P.....

Note: ChipON IDE provides default debug views, but not all views are opened when entering, but views can be opened and closed through operations. The common view is opened directly under the display view under the menu bar window. Some need to select others and perform the selection according to the category. How to operate: Click "window"→"Show view"→"Other". Such as the "Chip Info" and "Resource Utilization" view under the ChipON category. Debug view of chip information, including "Expression", "Register", "EEProm", "General register", etc..

13. Firmware Upgrade

Click "Help"→"Firmware Upgrade" on the menu. Select firmware upgrade mode to upgrade firmware.



There are two ways to upgrade firmware:

1. Automatic Upgrade

Download the firmware version program that comes with the release to the programmer.

2. Manual Upgrade

Select the file in the ". skp" format in the pop-up dialog box, and complete the firmware upgrade.

14. Hardware Emulation(Debugger) Common Error Resolution

1) Socket failed to establish connection!

Setting up a socket connection timed out. Please close some programs and try again, or check firewall settings.

2) Debugger not connected

The debugger is not connected or the serial port driver is not installed.

3) deviceICDPowerON () error!

The debugger set voltage operation failed, please confirm that debugging is normal or try again.

4) deviceReset () error!

Please check whether the configuration bit debug function is turned on, whether the MODE pin circuit is normal, and whether the ICSPCLK and ICSPDAT pin circuits are normal.

5) godomain () error!

Generally, this problem does not occur. If this problem occurs, it may be what the compiler line number go wrong. No more debugging. You need to start debugging in a file with main function. You can try to change the line number of main function and recompile it before starting debugging.

6) Flush ram (0x00-0x7F) failed!

Flush ram (0x80-0xFF) failed!

Flush ram (0x180-0x1FF) failed!

The communication is abnormal, and the general register value in the chip failed to be obtained. Please check the connection and try again.

7) The program has run away, please confirm that the program is correct and try again!

There are many reasons for this error:

1. Hardware problem-based program runs away.

2. Debug information may be abnormal.

3. The communication is abnormal. Please check the connection and try again.

8) The debug port is occupied, please reset P0.0, P0.1, and MODE!

P0.0 and P0.1 need to communicate with the debugger during, please do not set it as an output port or analog port. MODE pin needs to be used as a reset, please pay attention to external circuit.

9) The debug target is disconnected, please check the connection and try again!

Please check the connection and try again.

10) Setting breakpoint failed!

Please check the connection and try again. If it doesn't work after many attempts, there should be an error in the compiler information, or a breakpoint cannot be set here.

11) Running failed!

Please check the connection and try again.

12) Pause failed!

Please check the connection and try again.

13) Reset failed!

Please check the connection and try again.

14) RAM setup failed!

Please check the connection and try again.

15) Failed to print BEE!

Please check the connection and try again.

16) Serial port connection has been disconnected! Please check the connection and try again.

Abnormal data received by serial port.

Abnormal data sent by serial port.

Serial port receiving data communication is abnormal! Please check the connection and try again.

The serial port connection is interrupted. Please check whether the USB connection is normal.

17) Serial port receiving data communication is abnormal! Please check the connection and try again!

Serial communication throws an exception. Please check whether the USB connection is normal. Then try again.

18) Memory request failed!

Memory allocation failed. Please close some software and try again.

19) Abnormal PC communication reading! Please check the connection and try again.

Please check the connection and try again.

20) Exception reading register!

Failed to read all special function registers!

Please check the connection and try again.

21) Step failed!

Please check the connection and try again.

22) Running failed!

Check the connection and try again.

23) Failed to pause debugging target!

Check the connection and try again.

24) Failed to reset debug target!

Check whether the connection and configuration words are normal.

25) Debugging equipment is busy!

Debugger business!

Check the connection and try again.

26) Handling of Drive Recognition Class Problems

1) The driver is missing, perform the installation of the driver.

2) that port class, change the port number, such as use a smaller port for the

programmer port number, such as COM1.

3) The driver is not compatible, the driver is outdated than the chip, delete the historical driver, and install the latest driver.

4) Device exception: The device cannot be found in the computer device manager, and the non-driver is not installed.

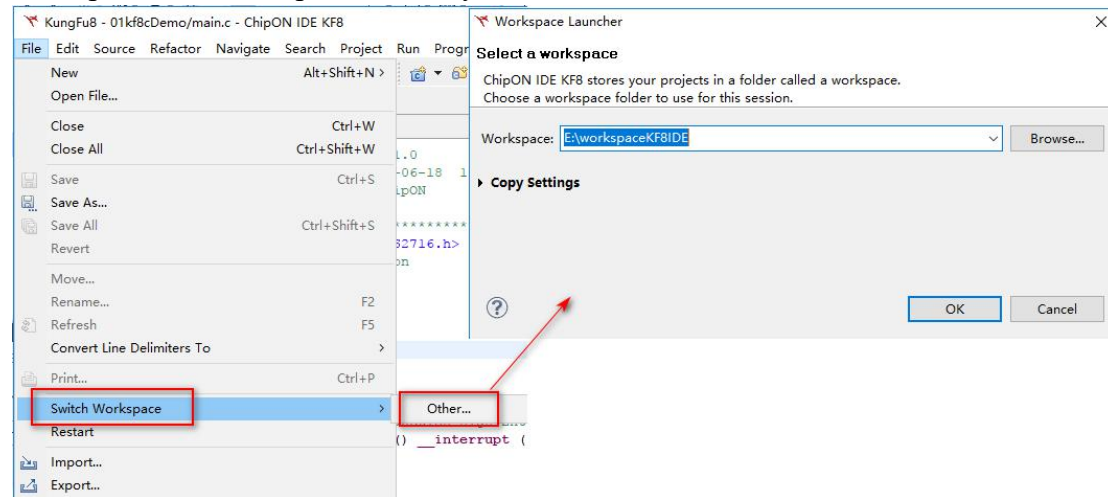
Note: See "Independent Download Package Programmer Driver" for detailed documentation on driver usage.

15. IDE Usage Skills

15.1 Toggle Workspace

Users can create multiple workspaces and switch between each workspace. Select "File"→"Switch Workspace"→"Other", as follows, you can select another workspace or create a new workspace. The switched workspace can be directly copied and pasted with the path, or confirmed by browsing the button.

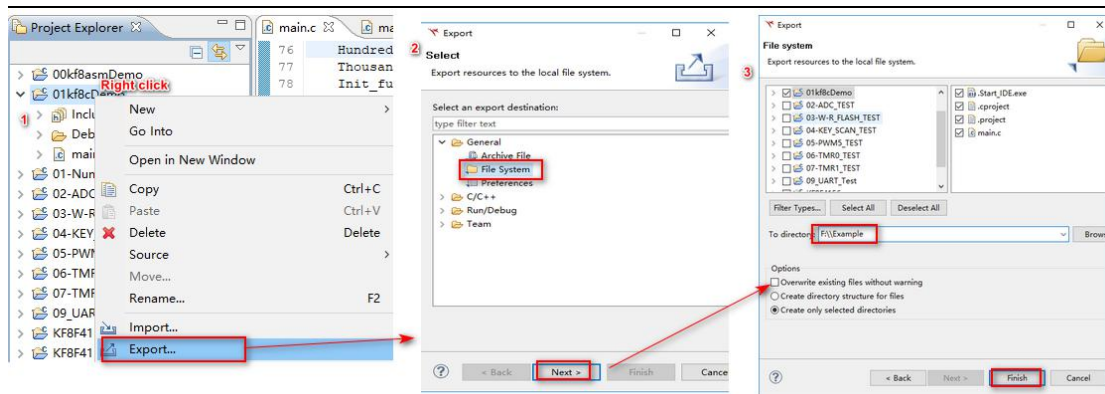
It should note that if the path is space, reference it. If the path is not a space, create a space at this location, but the space cannot be nested, that is, create a new workspace under the space directory.



15.2 Export Project

Export function can export the project files in the current workspace to the local specified path, which can be used as backup, etc.

Operation: Select the project file, right-click to open the menu, and select "Export" to enter the export interface. Select "General"→"File System", then click "Next", set the target path, click the "Finish".

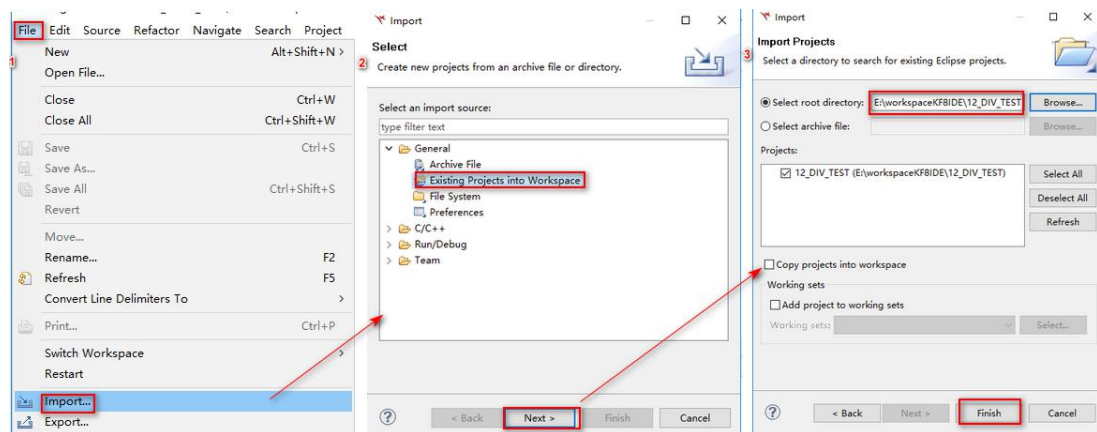


Note: The projects under the ChipON IDE direct project space are copied directly. Shortcut keys that support copy and paste. You can append date information as a backup of the project when pasting. It is not recommended to update the project name directly frequently. Once there is an abnormality in the process of changing the name, it may be inconsistent with the project information, which will cause inconvenience in use. By re-establishing the project, you can directly copy the project code file to the currently newly established project.

15.3 Import Project

Import function can import existing engineering files to the specified path of the current workspace.

Operation: Select "File"→"Import" to enter the import interface. Select "General"→"Existing Projects into Workspace" and click "Next". Then Select the project path and click Finish to successfully import the existing project into the current workspace.

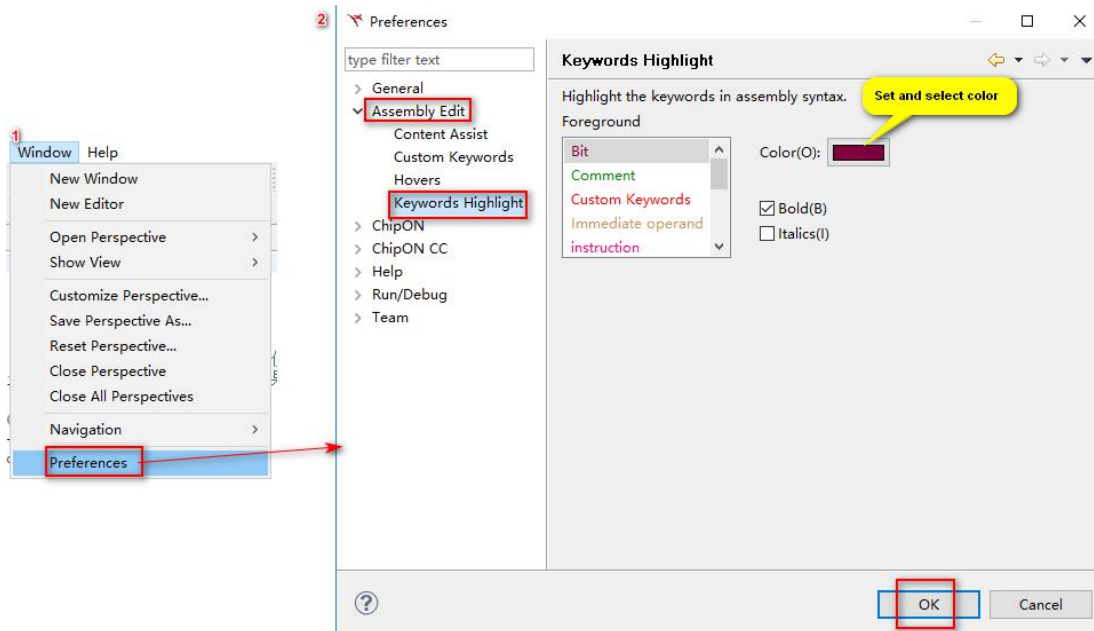


Note: Importing project with the same name is not supported in the space, and importing project in the current project space is not supported. It is recommended that this operation check "Copy project into workspace" to realize workspace-based project management.

15.4 Keyword Highlighting Settings

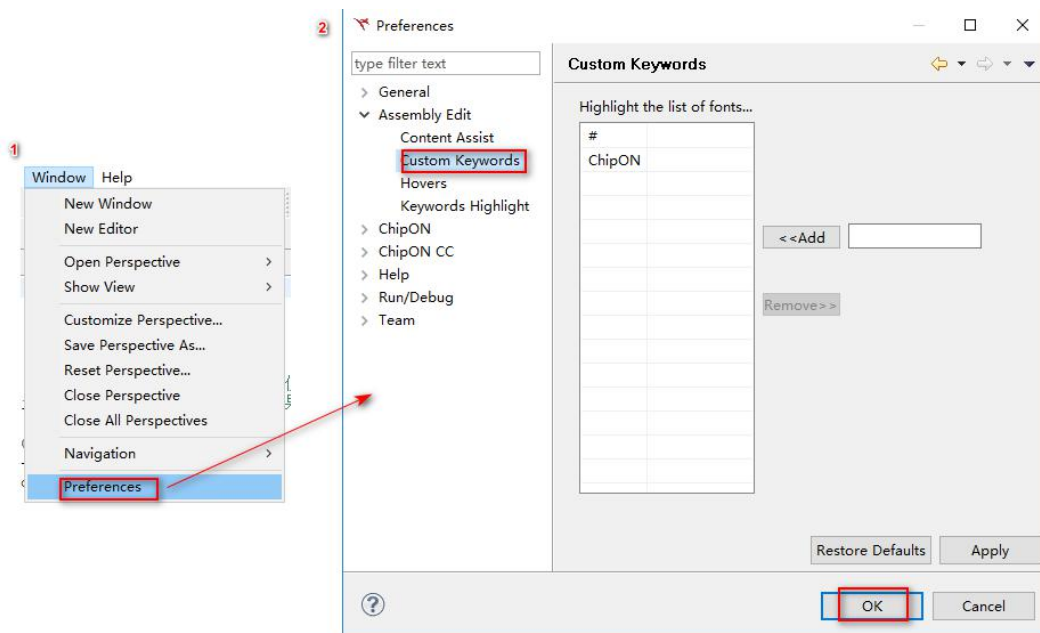
(1) Users can customize the color of keywords in assembly syntax displayed in the editor.

Select "Window"→"Preferences" to open preference settings. Then select "Assembly Edit"→"Keyword Highlight", you can set font color and set whether the font is bold or italic in the right page, click "OK" to complete the setting at last.



(2) Users can add keywords that need to be highlighted by themselves.

In the Preferences page, select "Assembly Edit"→"Custom Keywords", and add and delete keywords you need to define in the right page.



15.5 Code Completion

In the process of editing the code, if the instruction is relatively long, the input can be simplified through the code completion function. As shown below, if you need to enter a field with the label "INIT_ALL", enter "I" first, and IDE will automatically prompt the code. If the code prompt box disappears, it can be displayed through the shortcut key "ALT + /". Then choose "INIT_ALL" and click "Enter" key to input all fields.



For the development of C project, the structure and union will directly trigger the suspension box to prompt the content contained in the object. If you need to implement the prompt symbol in the above example, you need to manually execute the pop-up window indication by pressing the key. The shortcut key is "ALT+ /".

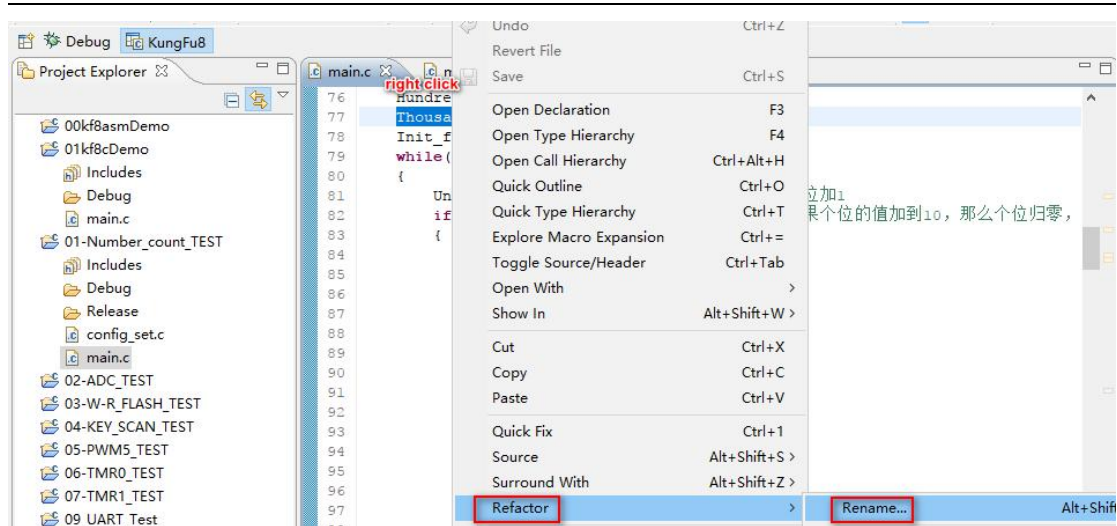
15.6 Refactor

Refactor refers to reconstructing existing code, uniformly modifying variable names or extracting reusable code into functions.

(1) Rename

If the meaning represented of variables in the existing code changes, the variable name needs to be modified. When variables are used too much, a variable name can be modified uniformly.

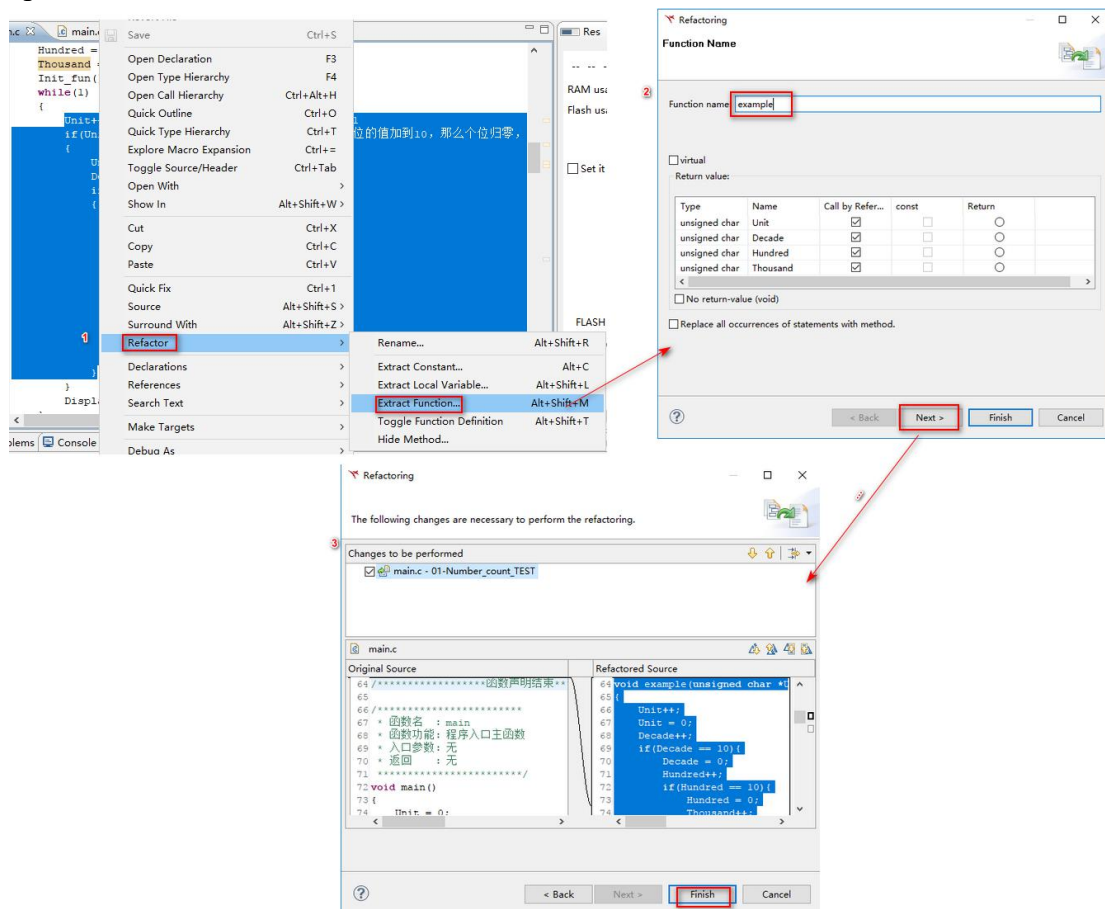
Select the name of the variable to be changed, right-click, and select "Refactor"→"Rename" from the right-click menu. Then input new name of variable and "Enter" to finish renaming.



(2) Extraction Function

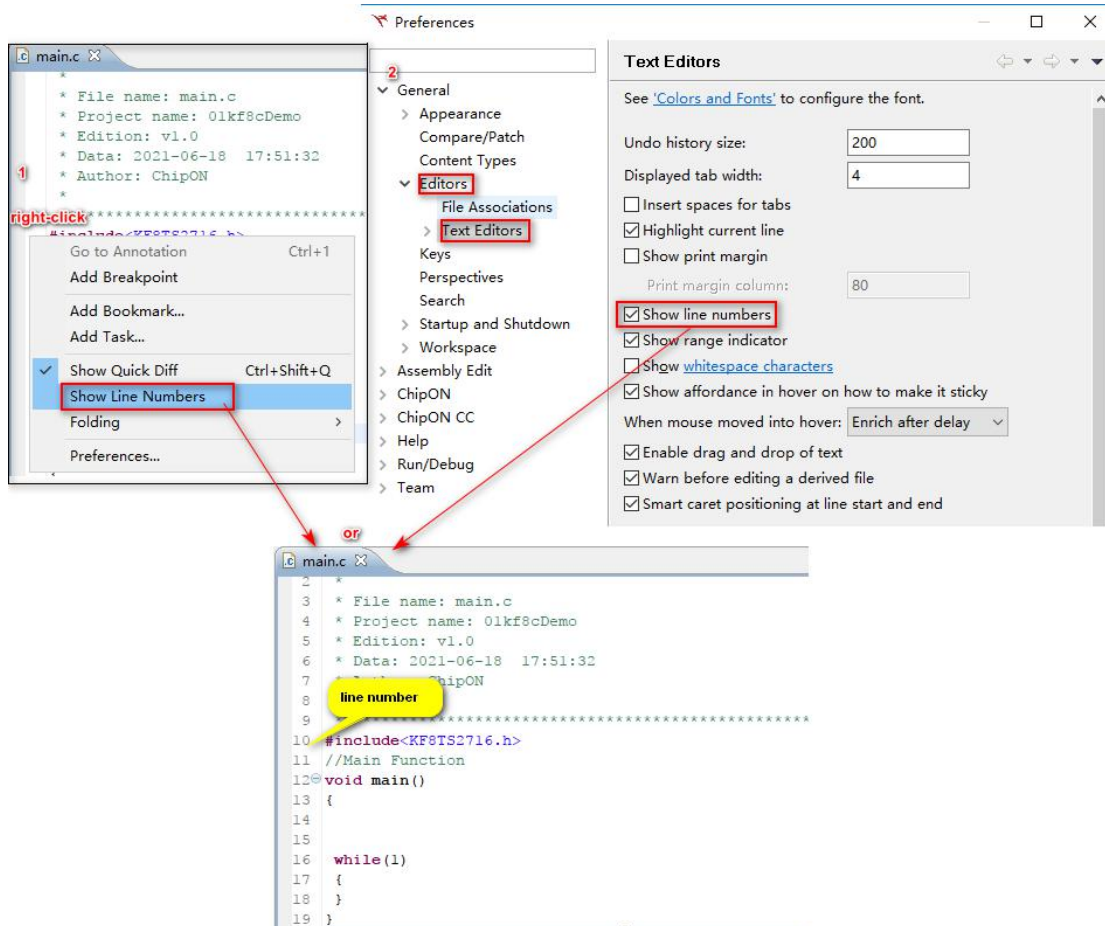
If you need to extract a segment of code from an existing function to facilitate code reuse, you can use extract function function.

Select the code block to be extracted, right-click, and select "Refactor"→"Extract Function" from the right-click menu. Then enter the required function name in the open refactor window, and select the parameter type according to the development needs, click "Next" to preview the refactored code. Click "Finish" to complete this operation.



15.7 Show Line Numbers

Right-click the left column of the editor and select the "Show Line Numbers". Or select "Window"→"Preferences"→"General"→"Editors"→"Text Editors", and select "Show line numbers" on the right page. Then you can see the line numbers of code.



15.8 Related Shortcut Keys

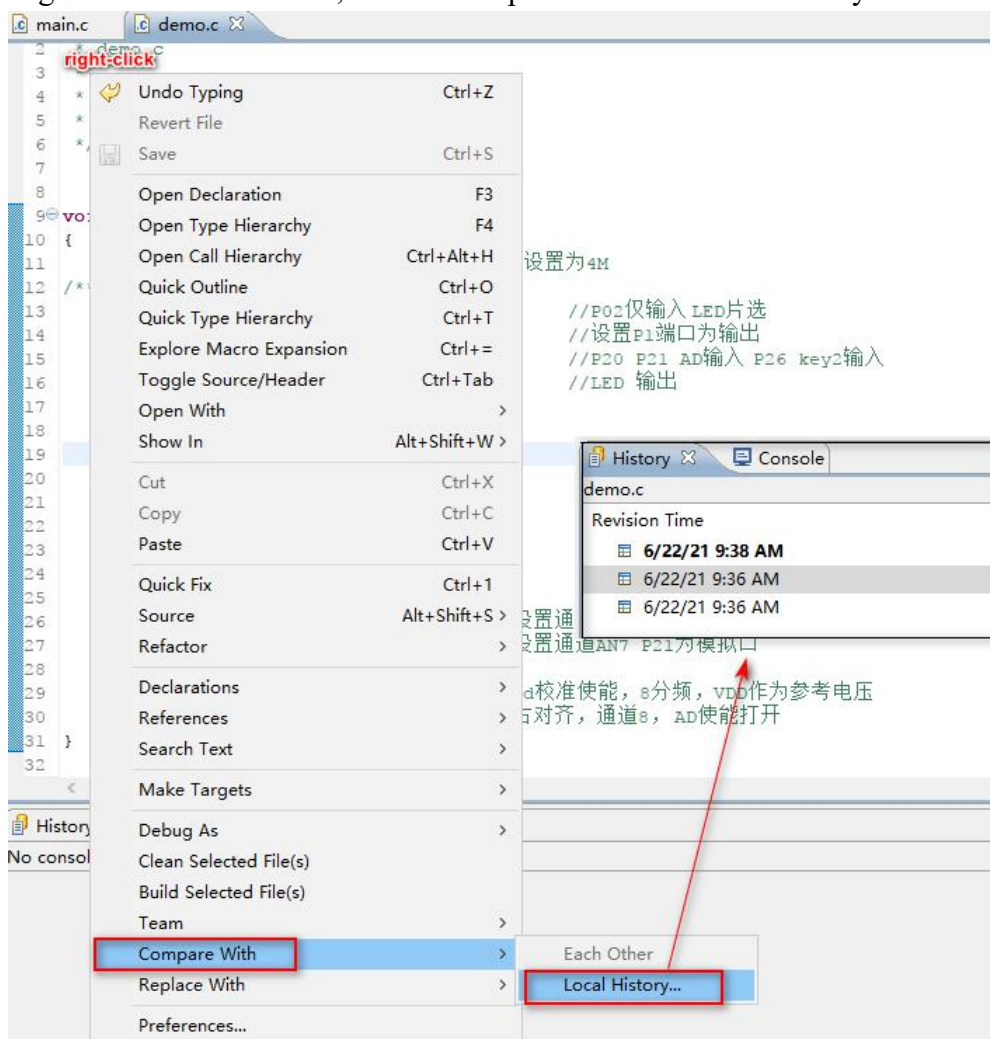
Shortcut	Function
Ctrl+/ or Ctrl+;	Comment or uncomment the line where the editor cursor is located.
Ctrl+B	Build all the projects in the workspace.
Ctrl+Shift+C	Check the downloader device before downloading the Hex file.
Ctrl+Shift+U	Online ICSP download
Ctrl+Shift+O	Reset operation
Ctrl+S	Save
Ctrl+C	Copy
Ctrl+V	Paste
Ctrl+X	Cut
Ctrl+Z	Undo Previous Action

Ctrl+D	Delete editor cursor line.
Ctrl+W	Close the current edited text.
Ctrl+F	Find Replace
Ctrl+H	Search
Ctrl+Alt+G	Finds the selected text in the workspace.
F5	Refresh
F2	Rename
Ctrl+Shift+X	Converts the selected string to uppercase.
Ctrl+Shift+Y	Converts the selected string to lowercase.
Alt+/	Code completion function

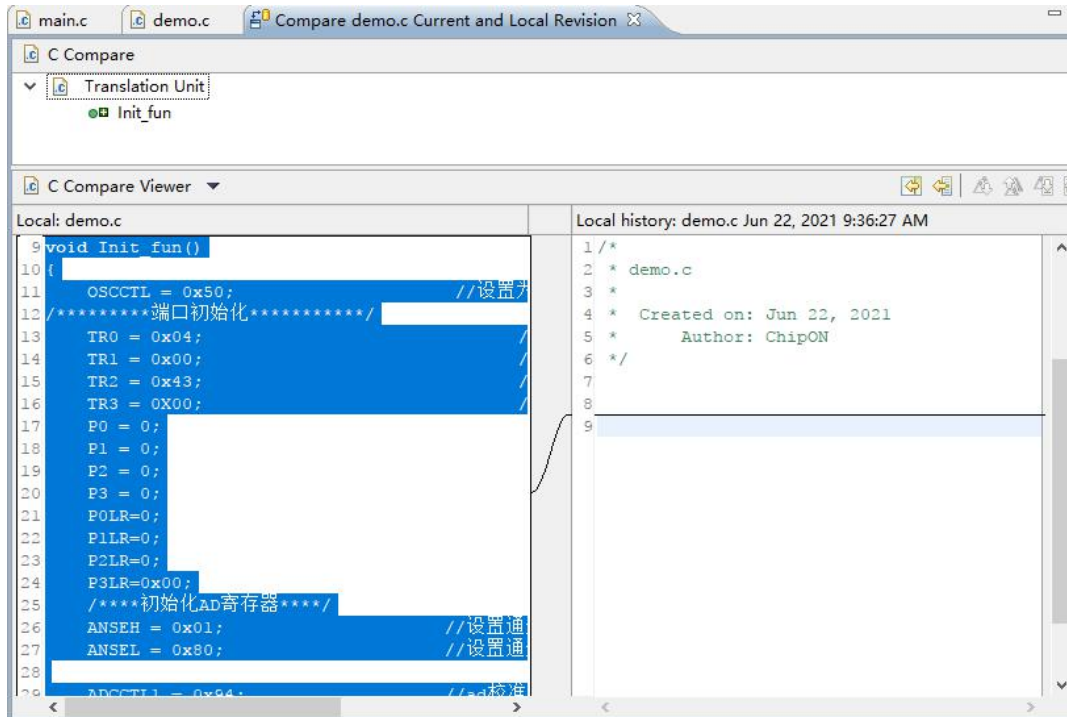
15.9 History File Comparison

In the process of editing a file, if you need to view a version in the history, you can do this by comparing objects.

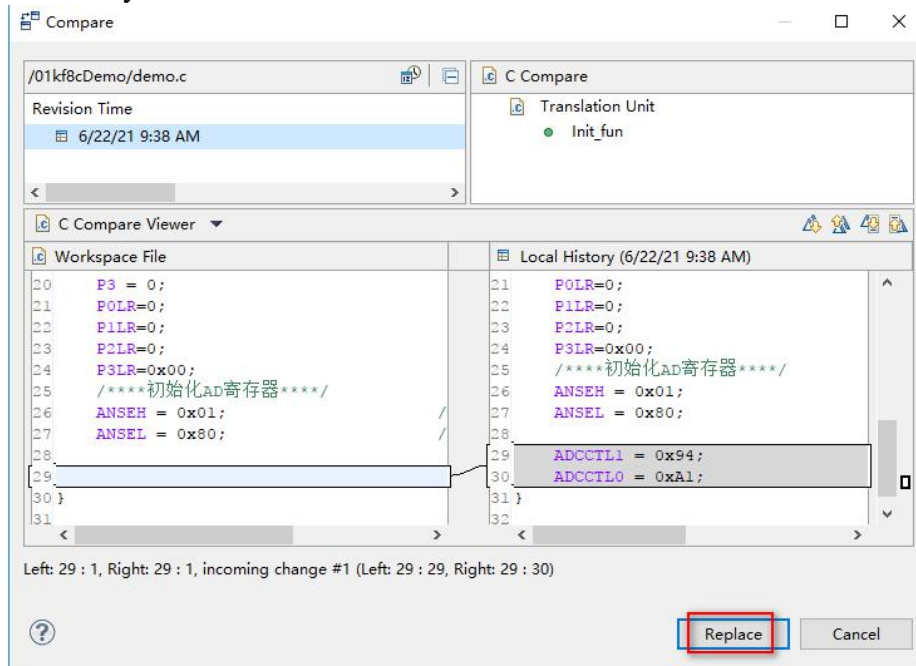
Right-click in editor view, select "Compare With"→"Local History...".



In the "History" view, double-click the file version to be compared to compare the current version with the historical version. You can see the difference between the two texts in comparison view.



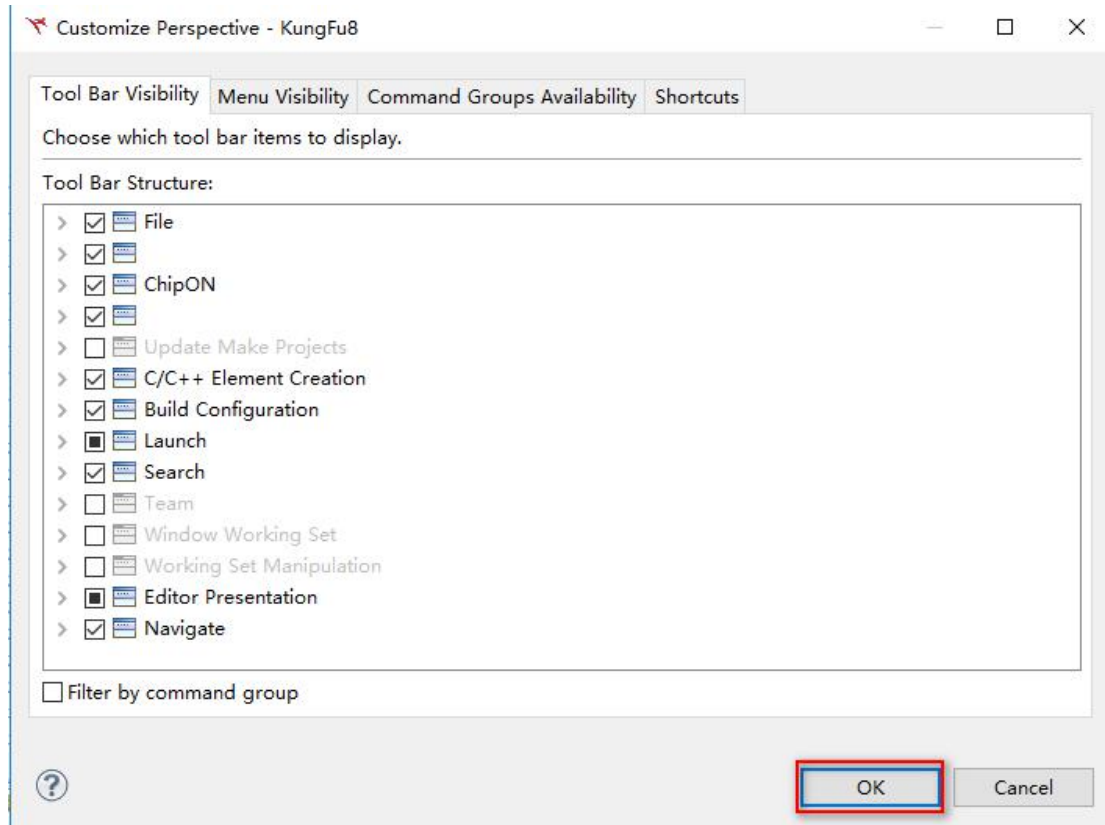
If you need to revert the current code to the history version, please right-click in editor view and select "Replaces With"→"Local History" to open the comparison window. Then double-click to select the historical version to be restored, and you can see the difference between the two files at the bottom of the page, click "Replace" to determine history code recover.



15.10 Custom Perspective

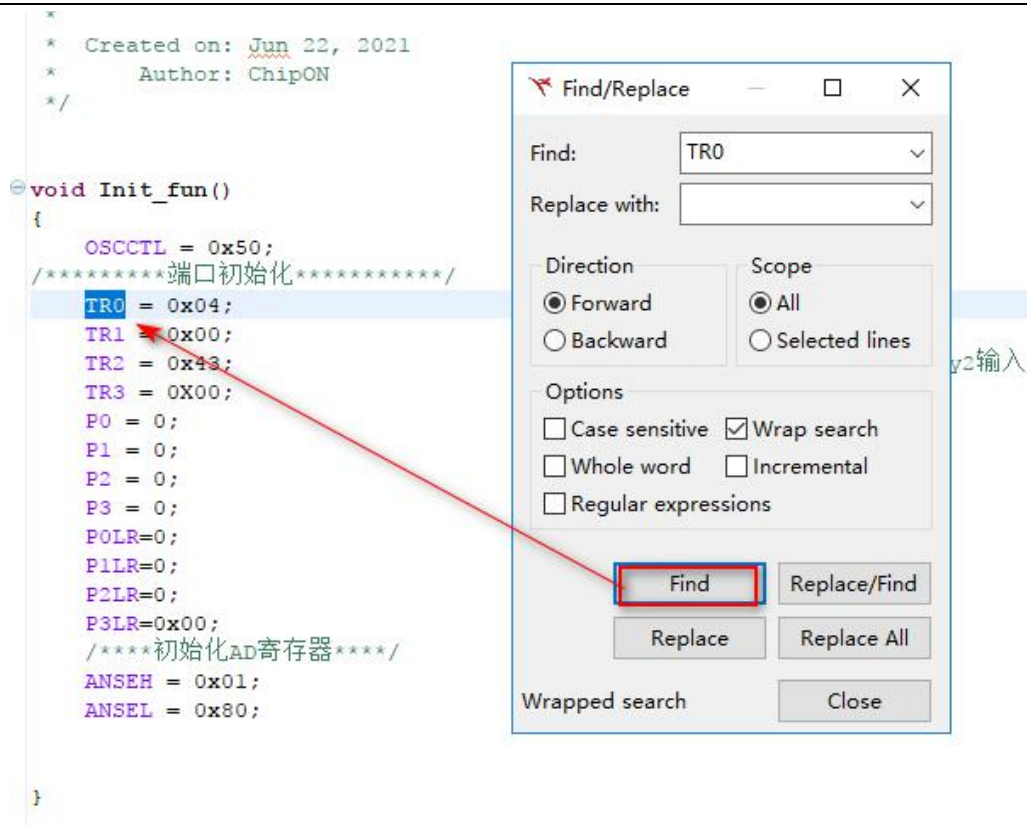
Users can add and delete the tools displayed in the development environment according to own needs. Users customize their own personalized development environment, which is convenient to improve efficiency.

Select "Window"→"Customize Perspective" to open the custom perspective page. And users can customize toolbar icons and menu bar menu option according to own habits and needs.

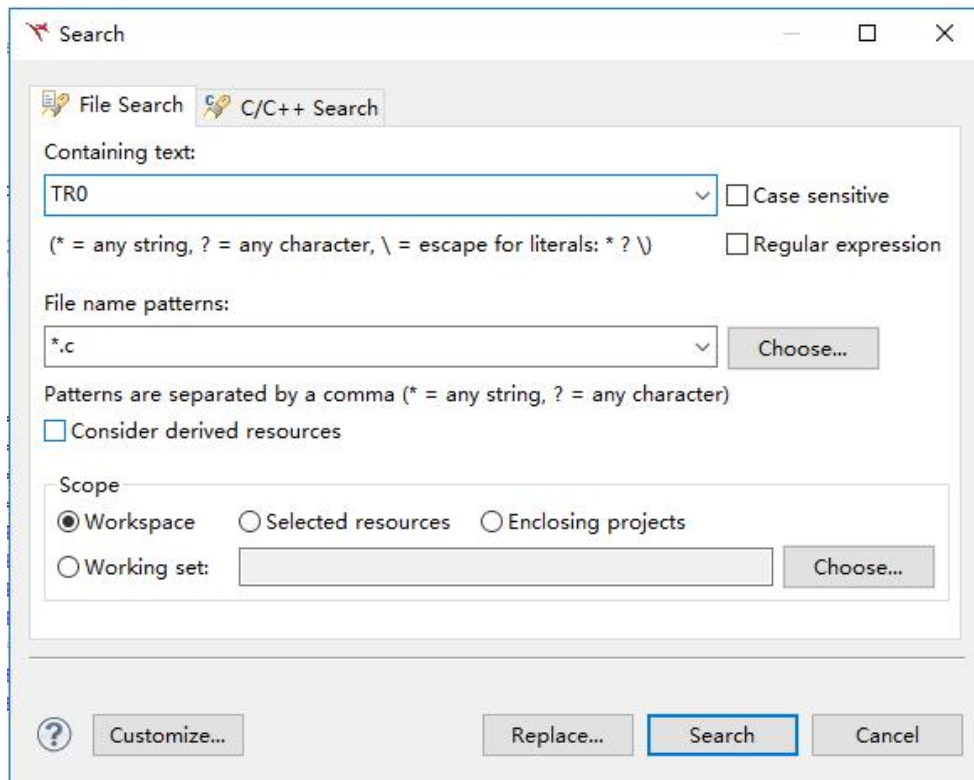


15.11 Find/Replace

The keyword search in the file by selecting "Edit"→"Find/Replace..." on the menu or the shortcut key "Ctrl + F". Enter the keyword to be searched in the open page and then click "Find" to find the keyword that you needed.




If you need search keyword during whole project or workspace, you can select "Search"→"Search..." on the menu or shortcut key "Ctrl + H" to open searching page. As shown below.



You can set the string you want to search and the scope of the search in above search window. The search results are displayed in the "Search" view.

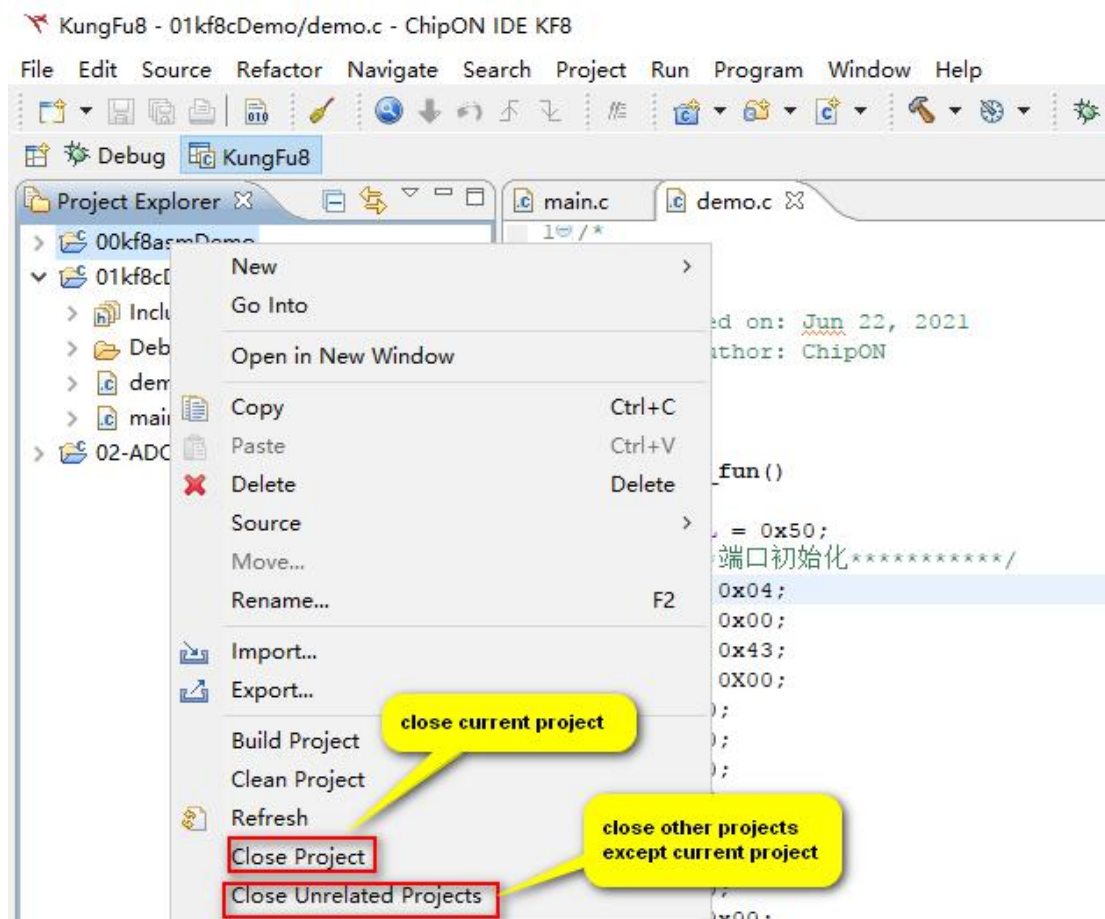
15.12 Return to the Previous Operation Position

During editing, a wide range of jumps will be made in the source file at times. If you want to return to the position of the previous operation cursor, you can realize this operation through the icon  in the toolbar.

15.13 Close Project

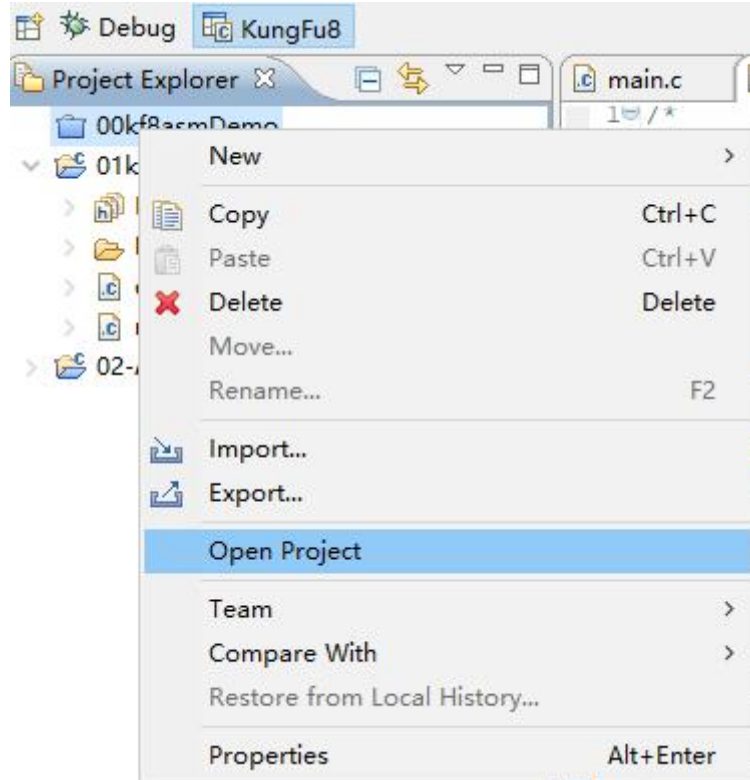
In the "Project Explorer" view, select the project, right-click and then select "Close Project" to close project.

If you need to close other projects besides this project, please select "Close Unrelated Projects".




15.14 Open Project


In the "Project Explorer" view, select the closed project that you want to open, then right-click, select "Open Project".



15.15 Maximize, Minimize and Close the Current Window

: It indicates minimize window.


: It indicates maximize window



: It indicates the minimized view window for restoration.

: It indicates close current window.

15.16 Quick Comment Code

➤ Single-line comment

For the line to be commented in the cursor point, the shortcut key "Ctrl +;" or "Ctrl +/" or click icon  on the toolbar.

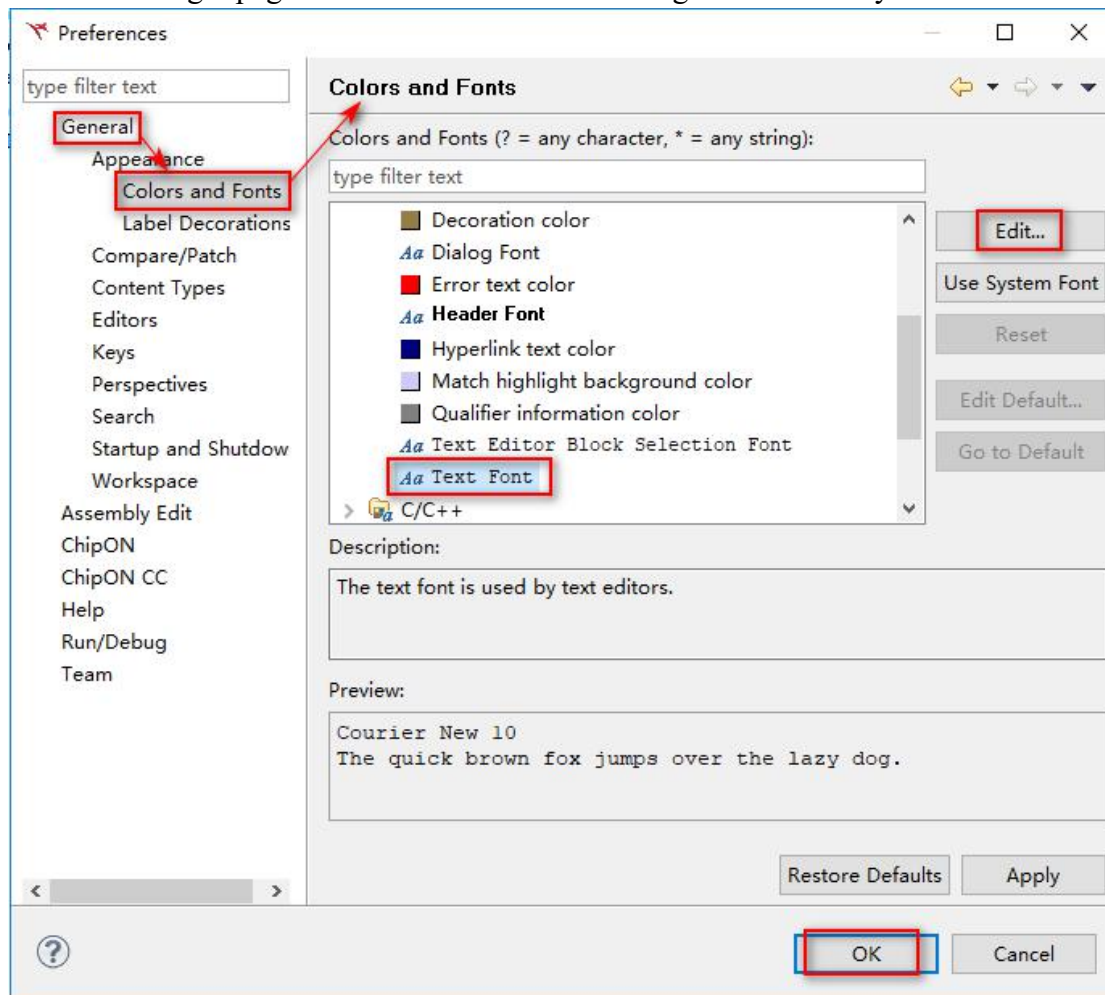
- Multi-line comment
Select the text to be annotated, and the shortcut key "Ctrl +;" or "Ctrl +/" or click icon  on the toolbar.
- Uncomment
Select the annotated text and the shortcut key "Ctrl +;" or "Ctrl +/" or click icon  on the toolbar.

15.17 Change the Position of the View Window

If you need to change the position of the view window in ChipON IDE, you can drag the window directly and place it in the position you want.

15.18 Change Editor Font

Select "Window"→"Properties" on the menu to open properties page. Select "General"→"Appearance"→"Colors and Fonts", and then select "Basic"→"Text Font" in the right page. Click "Edit..." to set or change editor font style or font size.

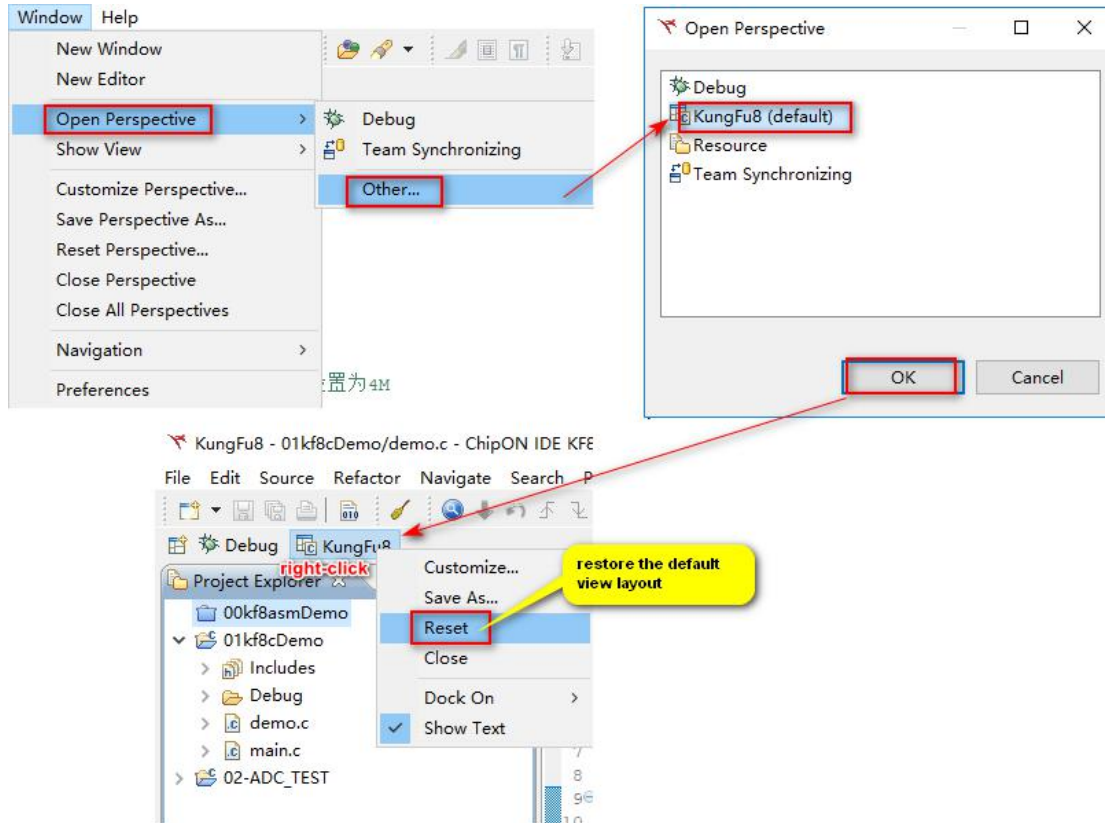


15.19 Restore Initial View Layout


Select the "KungFu8 Perspective" and right-click, select "Reset" to restore initial view layout of ChipON IDE.

If "KungFu8 Perspective" does not be found, please open "KungFu8 Perspective" through Selecting "Window"→"Open Perspective"→"Other"→"KungFu8" .

The other way realizes this target which is selecting "Window"→"Reset Perspective..." on the menu.

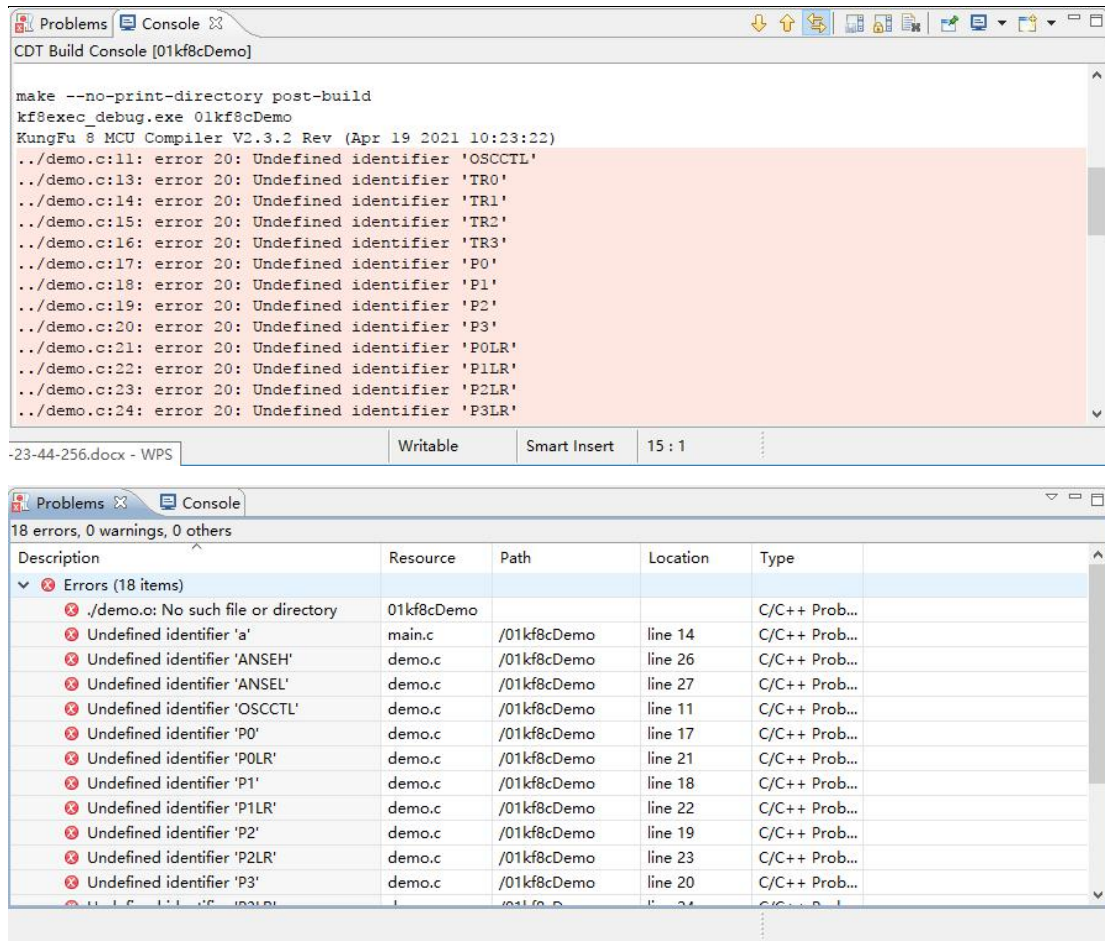


15.20 Clean Project

- Select the project and click the icon  on the toolbar.
- Select the project and right-click, then click "Clean Project".

15.21 Quickly Locate Error Information

After the build is completed, the build information will be displayed in the "Console" view. If there is an error or warning, you can double-click directly to jump to where the error occurred. And in the "Problem" view, the specific information about errors or warnings is also displayed.



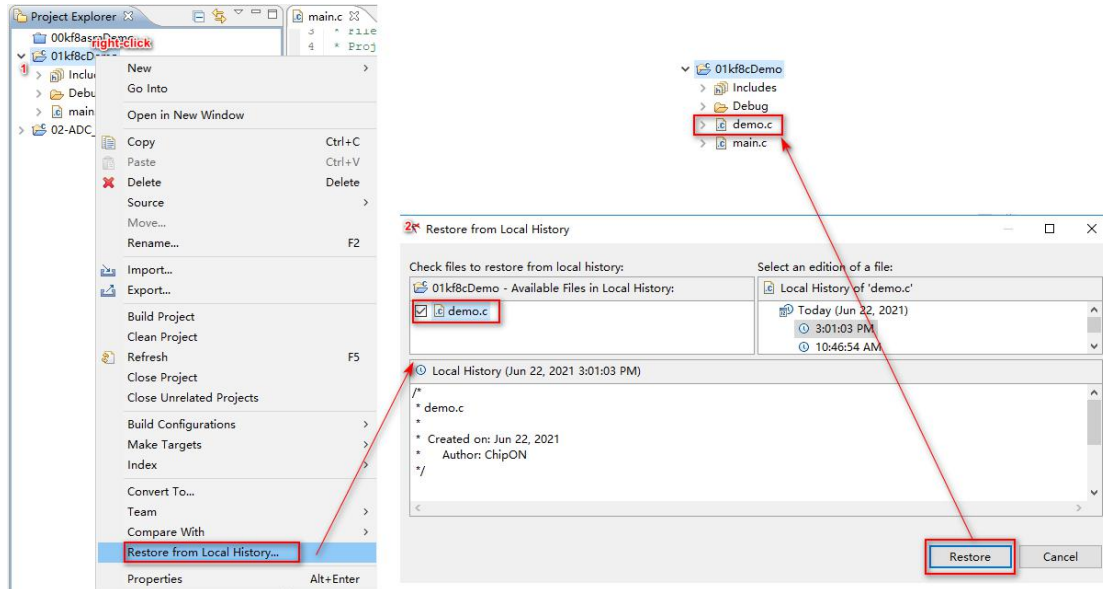
When it can not locate error quickly, it is also necessary to stipulate the content of the error report and judge the error source according to the content.

15.22 Restore Deleted Files

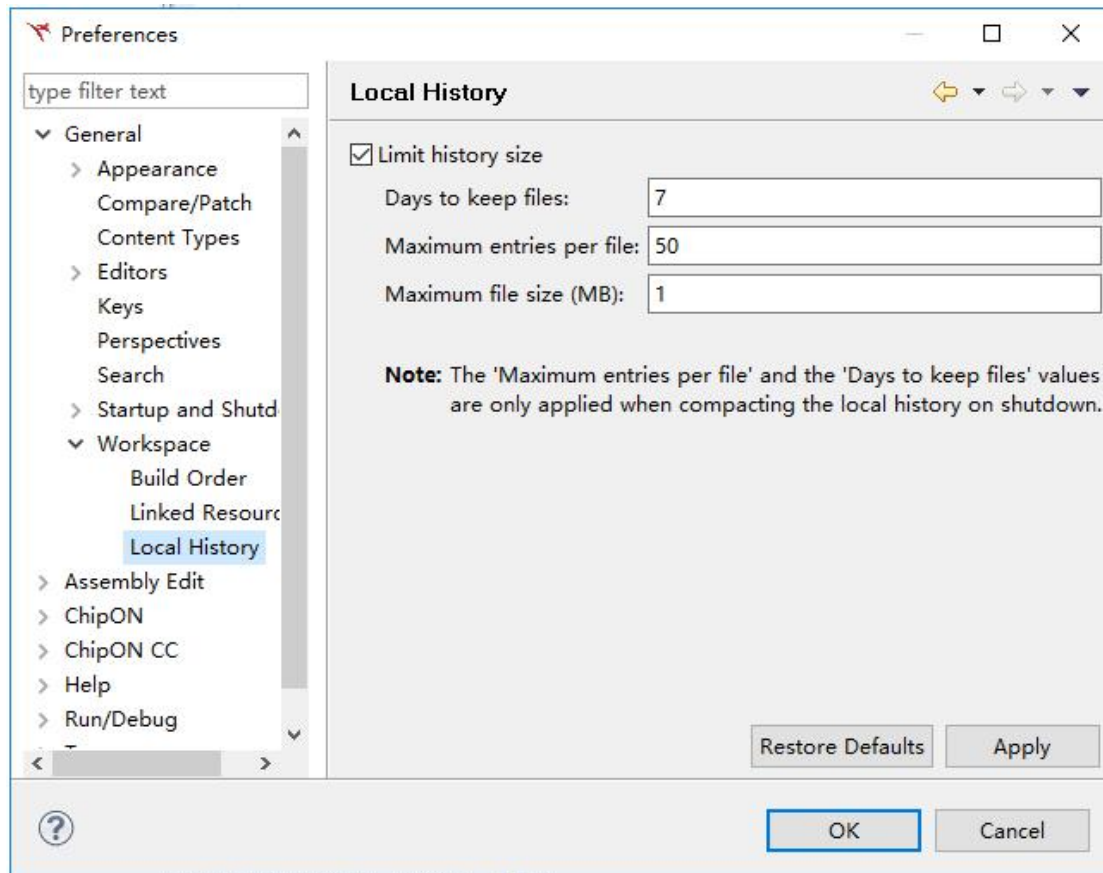
During development, if you want to restore previously deleted files, you can restore them through functions provided in "Project Explorer" view.

Select the project and right-click, then select "Restore from Local History" to open the "" dialog box. Next to select the file to be restored and click the "Restore" to take effect this operation.

For example, if you delete "demo.c" file, and you want to restore it, you can execute this operation. As shown below.

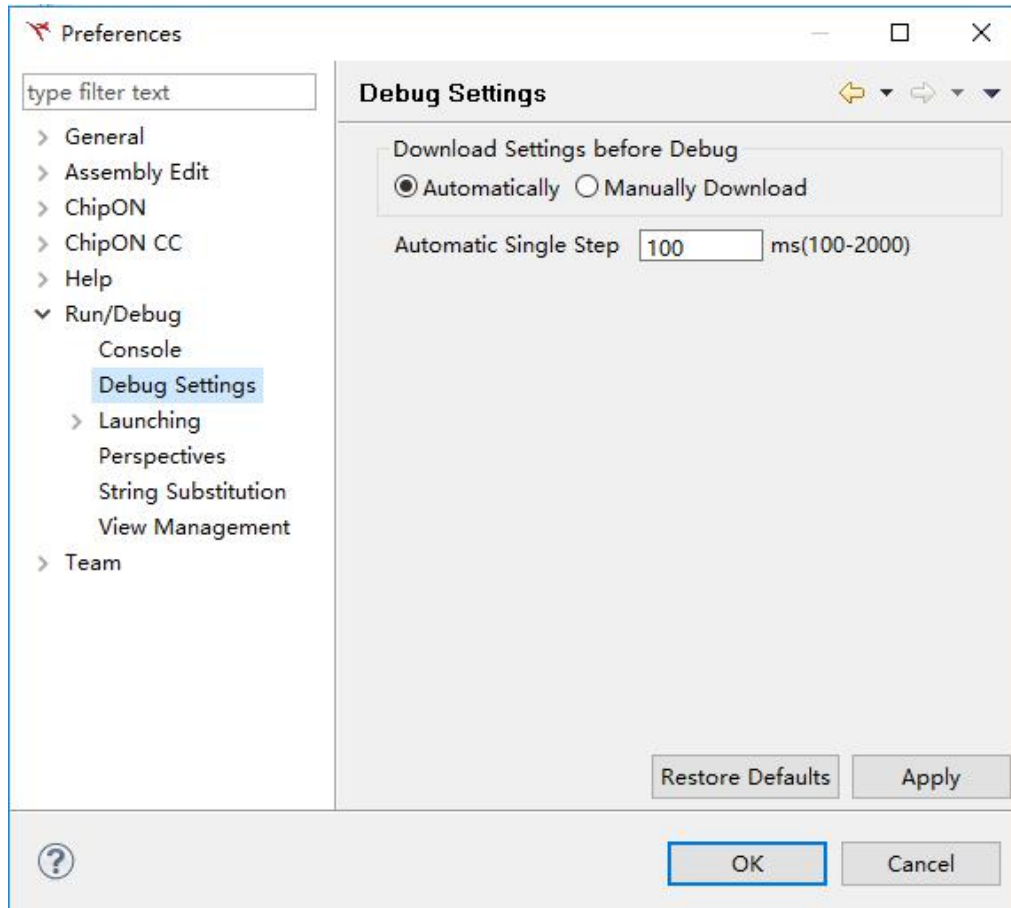


In addition, it should note that there is a certain time limit and number limit for the recovered files. Users can set this limit by themselves. Select "Window"→"Preference"→"General"→"Workspace"→"Local History". Then users can customize their own storage methods according to their own needs.



15.23 Debug Settings

Select the "Window"→"Preferences" on the menu, then select "Run/Debug"→"Debug Settings" in preferences page. Setting download mode in "Download Settings before Debug", and fill in the automatic step interval value in the text box, as shown in the following figure.



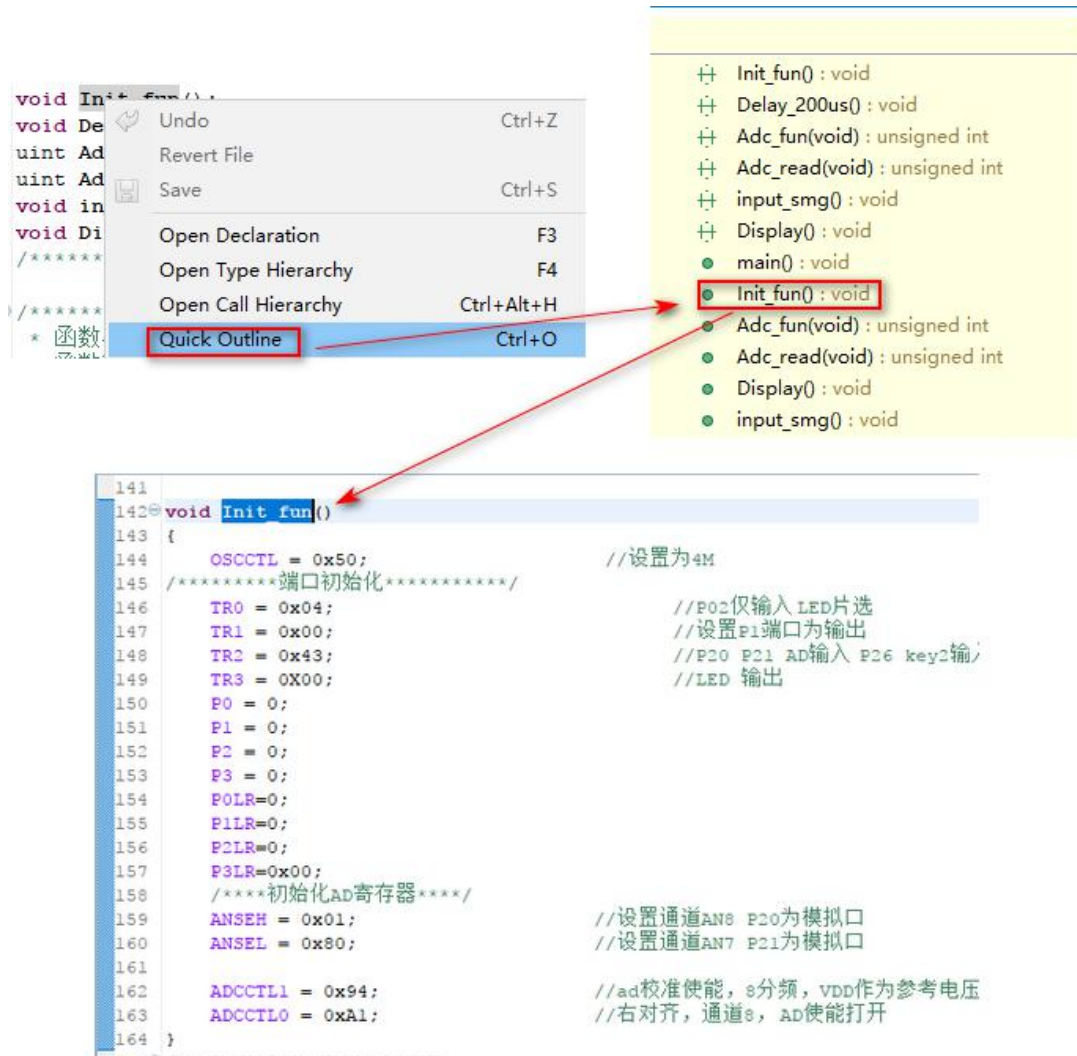
Click "OK". When entering the debugging mode, if "Automatically" is selected, download to the chip first, and then enter the debugging mode. If "Manually Download" is selected, enter the debugging mode directly, and click "Continuous Single Step". The direct interval between single steps is the value in the text box.

15.24 Code Jump

There are two method to jump to the special code.


Method 1: Hold down the "Ctrl" key and move the mouse over the code to jump. Then click mouse to jump to the target code.

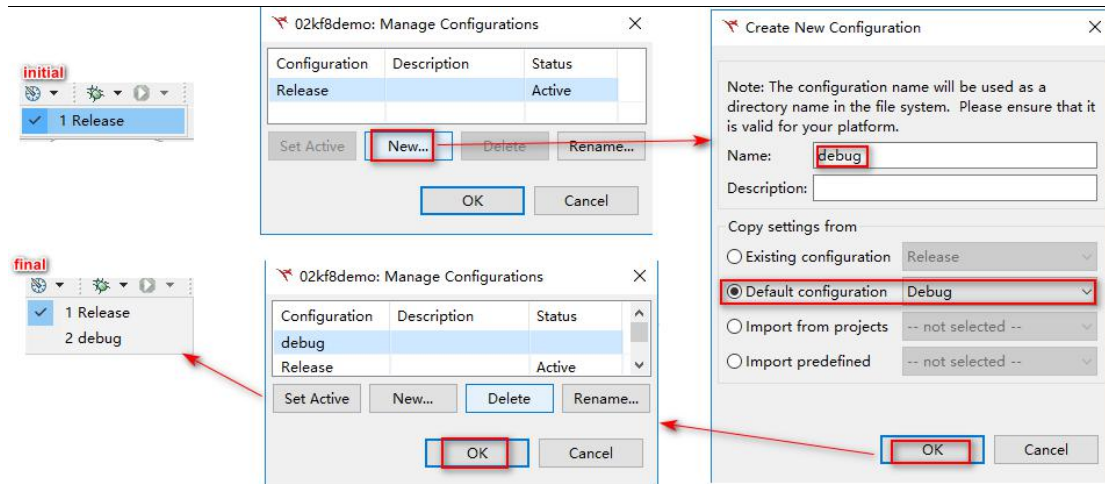
Method 2: Select the code to jump, right-click, and select "Quick Outline" in the menu, as shown in the figure.



15.25 Manager Configuration

When you create a new project, if you forget to select debug or release, you can retrieve it in the following way.

Firstly, click icon  on the toolbar, and the interface shown in the following figure will appear (the following figure is forgetting to select debug). Then click "New..." and click "OK" after setting "Debug" option. The detail process as shown below figure.



Note: The name of the new configuration is not limited. But it is suggested to implement the two configurations (Release and Debug) in above figure, and ensure that the name and meaning are consistent.

15.26 View Compile Results

Project compilation not only generates hex machine files, but also two very useful files. LST files and map files with the name of the project, which are output directory of compilation result.

Among them, LST contains the most information, including machine code under each program address, assembly instructions and source code correspondence. This is also the starting point to see if the compilation results are correct. The purpose of the map file is to view the address allocation of variables defined in the project or intermediate variables used in the compilation process. According to the address of the variable, you can view the actual result of the variable by opening the "Memory" view during debugging, which is not limited to the variable or expression window.

15.27 Make Code Compilation Efficiency High

a). Bit inversion

```
Flagbit0 = ~Flagbit0 ;//low efficiency
Flagbit0 = ! Flagbit0 ;//high efficiency
```

```
if(Flagbit0 ==0) //highest efficiency
{
    Flagbit0 = 1;}
else
{
    Flagbit0 = 0;}
```

b). Bit judgment

```
if(DCIN!=1)    //low efficiency
```

```
if(DCIN==0)    //high efficiency
if(DCIN==1)    // low efficiency
if(DCIN)       // high efficiency
```

c). Define

Under the condition of sufficient RAM space, bit variables defined by structures are not defined as byte type, which is efficient. It is inefficient to use pointers to express, so it is used as little as possible. You can define simple union structures. Such as

```
AD_sampl=ADCDATAH<<256|ADCDATAL
is less then
AD_sampl.lb= ADCDATAH; AD_sampl.lb= ADCDATAL;  efficiency high.
```

d). Expression

```
if(vot_value<(vot_temp-1))    //low efficiency
vot_temp=vot_temp-1;
if(vot_value<vot_temp)        //high efficiency
```

```
-----
a[i*j+2]=a[i*j];              // low efficiency
m=i*j;
a[m+2]=a[m];                  // high efficiency
```

e). Logical operation

The function with return value.

```
dc_value=adc(2)/2;           //low efficiency
dc_value=adc(2);
dc_value=dc_value/2;         // high efficiency
```

f). Array parameters

```
fun(a[],x,y)                  // low efficiency
fun(x,y,z)                    // Where Z control conditional branches directly use
                                global or local data to operate directly.
```

g). Embedding principles

The available R registers limit the use of R0 and R1, like others registers are used, it should be determined that the interrupt is turned off at this time, because R2-R5 is used as the protection memory of the interrupt field. For registers plus BANSEL to enable the compiler to automatically process partition positions. For function calls that need to consider page cutting supporting MOVP instruction series chips, PAGESEL modification can be used to enable the compiler to automatically process. Pre-assembly variables can only use special registers or global variables, C variables to assembly conversion reasons, embedded assembly operational variables need to add "_" to modify. Example:

```
unsigned int a;
void fun1()
{
    __asm
        MOV R0,#0x33
```

```
BANKSEL _a
MOV _a,R0
__endasm;
Flag=1;
}
void fun2()
{
    Fun3();
    __asm
        PAGESEL _fun1
        CALL _fun1
        PAGESEL $
    __endasm;
}
```