

# Nu Eclipse SDCC Quick Start

## CONTENTS

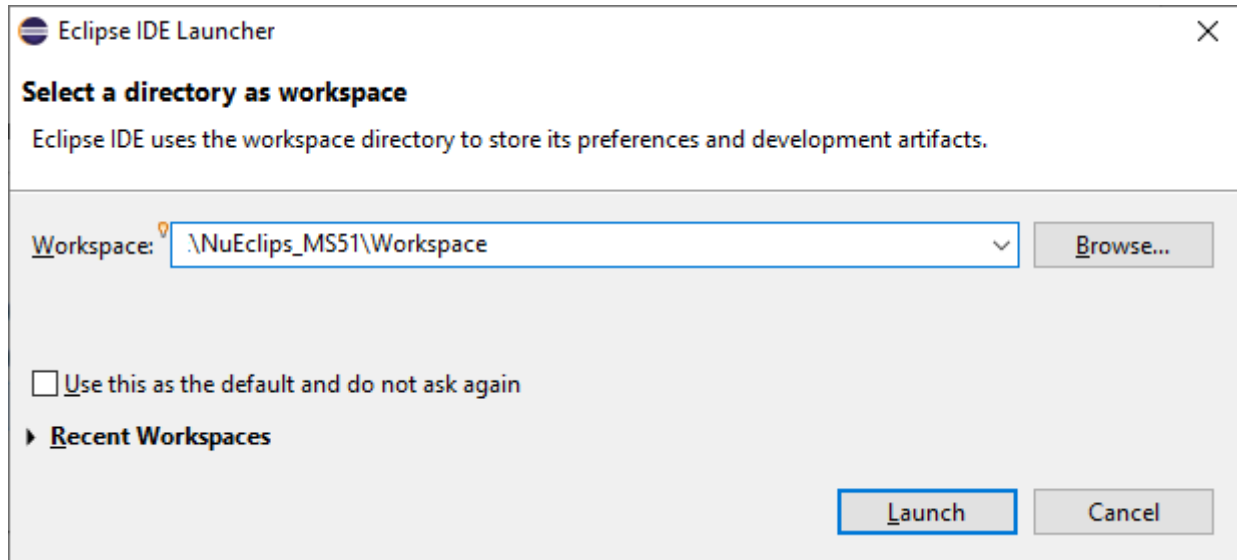
<b>1</b>	<b>PROCESS TO SETUP A PROJECT.....</b>	<b>2</b>
1.1	Start an Existing BSP project .....	2
1.1.1	Eclipse Workspace Always Import Project.....	2
1.1.2	Import Project.....	3
1.1.3	Compiler / Build.....	4
1.1.4	Debug.....	5
1.1.5	Add or Remove Library .c file .....	7
1.1.6	Add User Code.....	8
1.1.7	Close Project .....	8
1.2	1.2 Create a new Project from an existing BSP project .....	9
1.2.1	Copy the existing SDCC project folder to the new Project folder .....	9
1.2.2	1.2.2 Import this project in Workspace .....	9
1.2.3	Rename Project .....	9
1.3	Create A New Project (Not Recommended) .....	11
1.3.1	New Project Define.....	11
1.3.2	New Create Library And User Source Code.....	11
<b>2</b>	<b>PROJECT PROPERTIES .....</b>	<b>14</b>
2.1	Setting Project Properties .....	14
2.1.1	Build / Environment / PATH.....	14
2.1.2	Build setting .....	14
<b>3</b>	<b>DEBUG ENVIRONMENT .....</b>	<b>17</b>
3.1.1	Set the location of different regions to .....	17
3.1.2	Use The Expression Window To View Variables.....	17
3.1.3	Set Break Point .....	18
<b>4</b>	<b>APPLICATION TIPS .....</b>	<b>19</b>

# 1 PROCESS TO SETUP A PROJECT

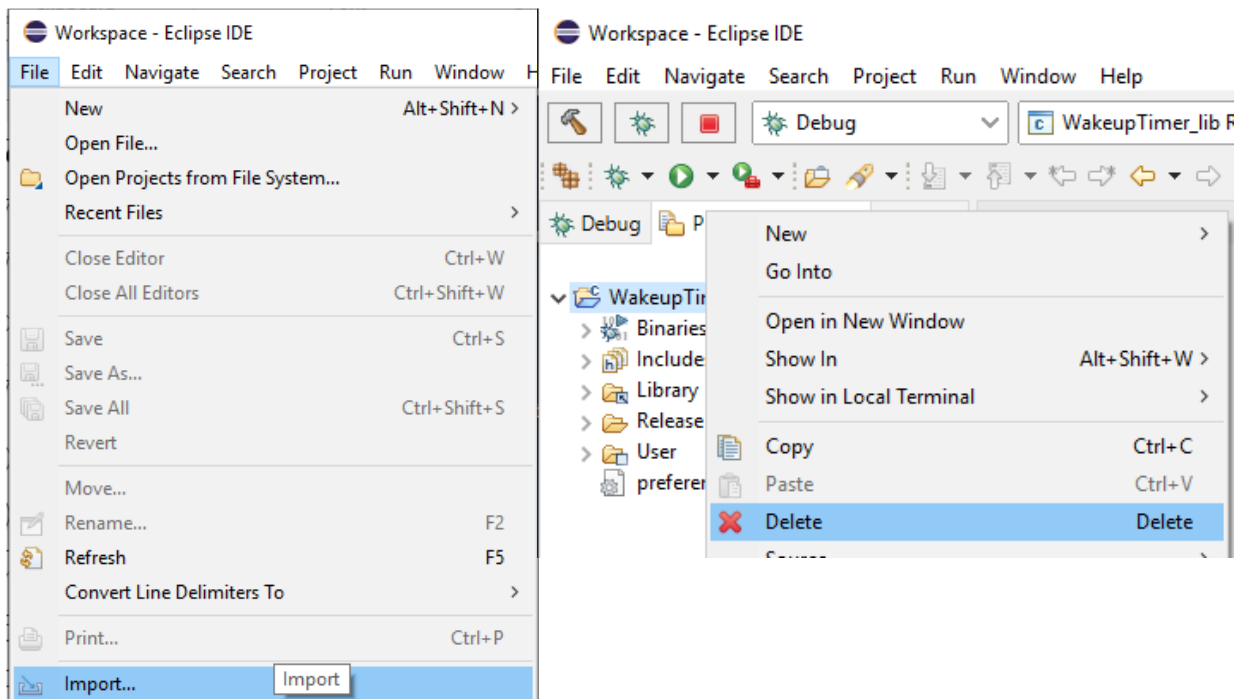
## 1.1 Start an Existing BSP project

### 1.1.1 Eclipse Workspace Always Import Project

BSP has created a workspace folder that can be opened directly. Directly click the corresponding path in the workspace of the launcher

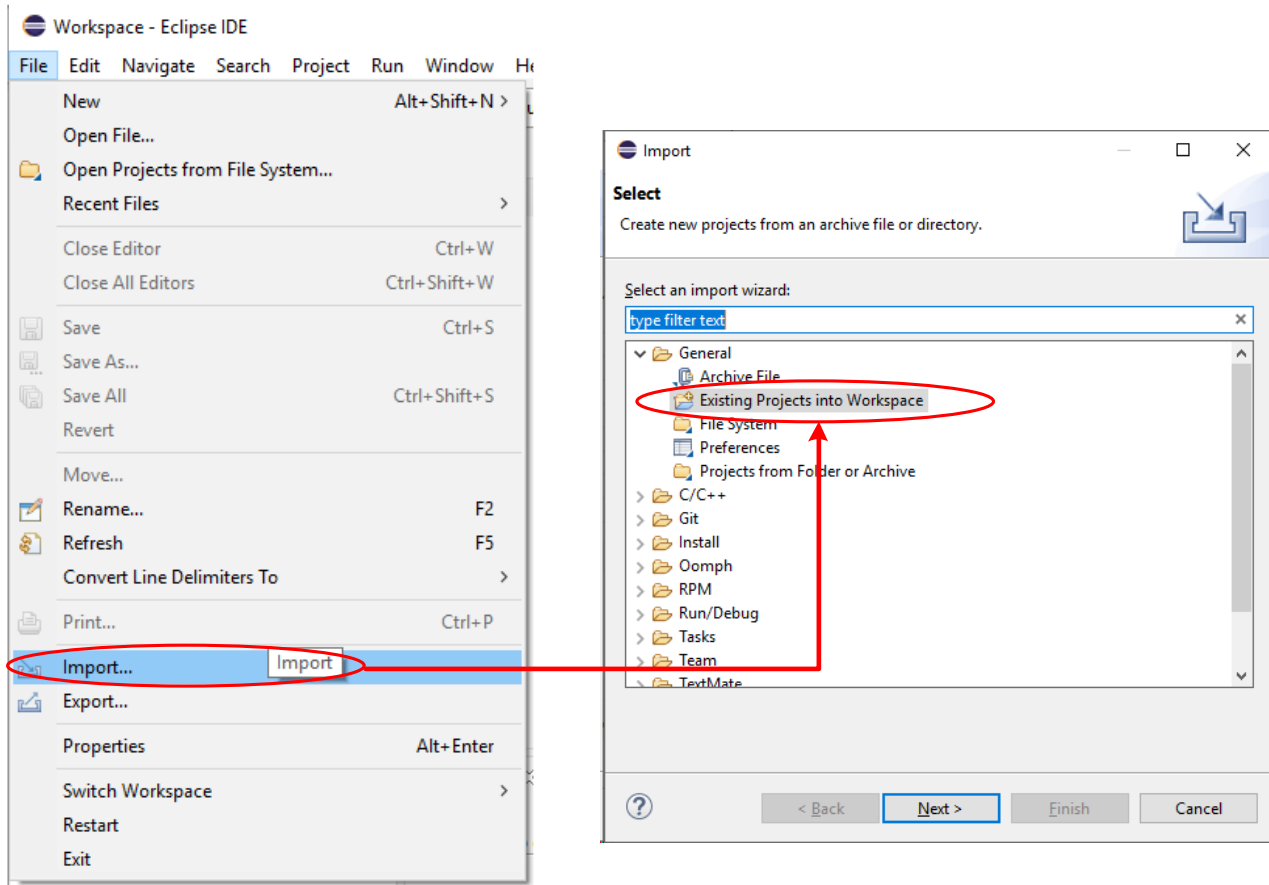


When opening a project, use “import”, and to closing it, use “delete project” in workspace

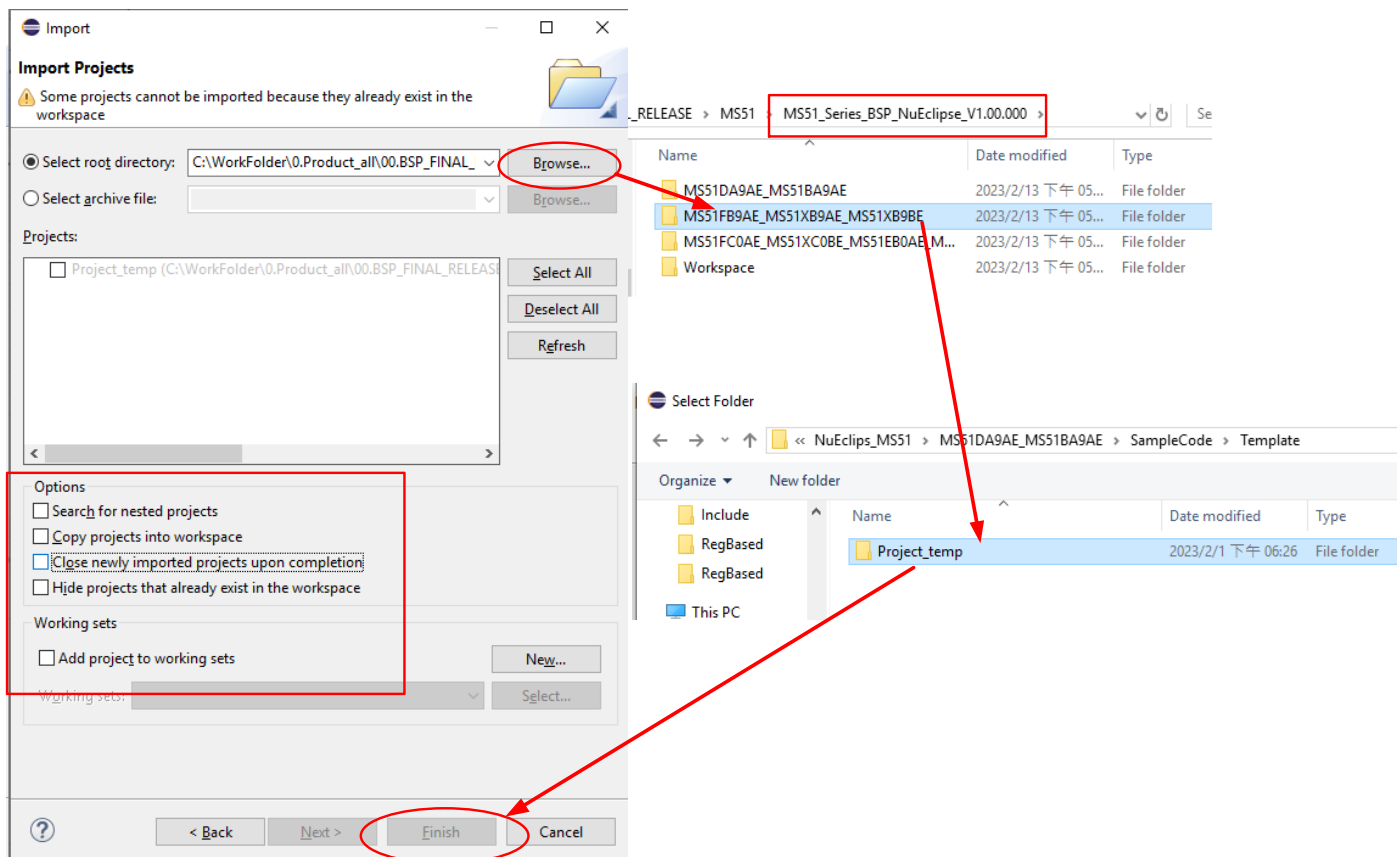


## 1.1.2 Import Project

Click “file / Import “Existing Projects into Workspace” item.



Click the “project folder” corresponding to BSP in import Projects, and click Finish.



### 1.1.3 Compiler / Build

Right-click on the Project name and select "Build Project". After normal completion, till find "0 errors, 0 warnings" and checksum related information will be displayed.

The screenshot illustrates the process of building a project in the Eclipse IDE. The top-left pane shows the 'Project Explorer' with 'Project\_temp (in SDCC)' selected. A red arrow points from this project to the 'Build Project' option in the context menu. The bottom-left pane shows the updated project structure, including 'Binaries', 'Includes', 'Library', 'Release', and 'User' folders. The bottom-right pane shows the 'CDT Build Console' output, which includes the build command, checksums, and the final status: '19:02:05 Build Finished. 0 errors, 0 warnings. (took 3s.845ms)'.

Workspace - Eclipse IDE

File Edit Navigate Search Project Run W

Debug Project Explorer

Project\_temp (in SDCC)

- Includes
- Library
- User
- preferences.ini

Build Project

Incremental Build of Selected Files

main.c

```
7
8
9 /* File Function: MS51 DEMO project
10
11
12 #include "MS51_8K_SDCC.h"
13
14
15 void main (void)
16 {
17 /* UART0 initial setting
18 * include sys.c in Library for modify HIRC value to 24MHz
19 * include uart.c in Library for UART initial setting
20 */
21 MODIFY_HIRC(HIRC_24);
22 Enable_UART0_VCOM_printf_24M_115200();
23
24 printf("\n Hello world!");
25 while(1);
26
27
28 }
```

Console

CDT Build Console [Project\_temp]

Finished building target: Project\_temp.elf

make --no-print-directory post-build  
arm-none-eabi-objcopy -R REG\_BANK\_0 -R REG\_BANK\_1 -R REG\_BANK\_2 -R  
>>> Project\_temp.bin Checksum = 0xF7C3 <<<  
>>> Project\_temp.bin CRC-8 Checksum = 0xED <<<

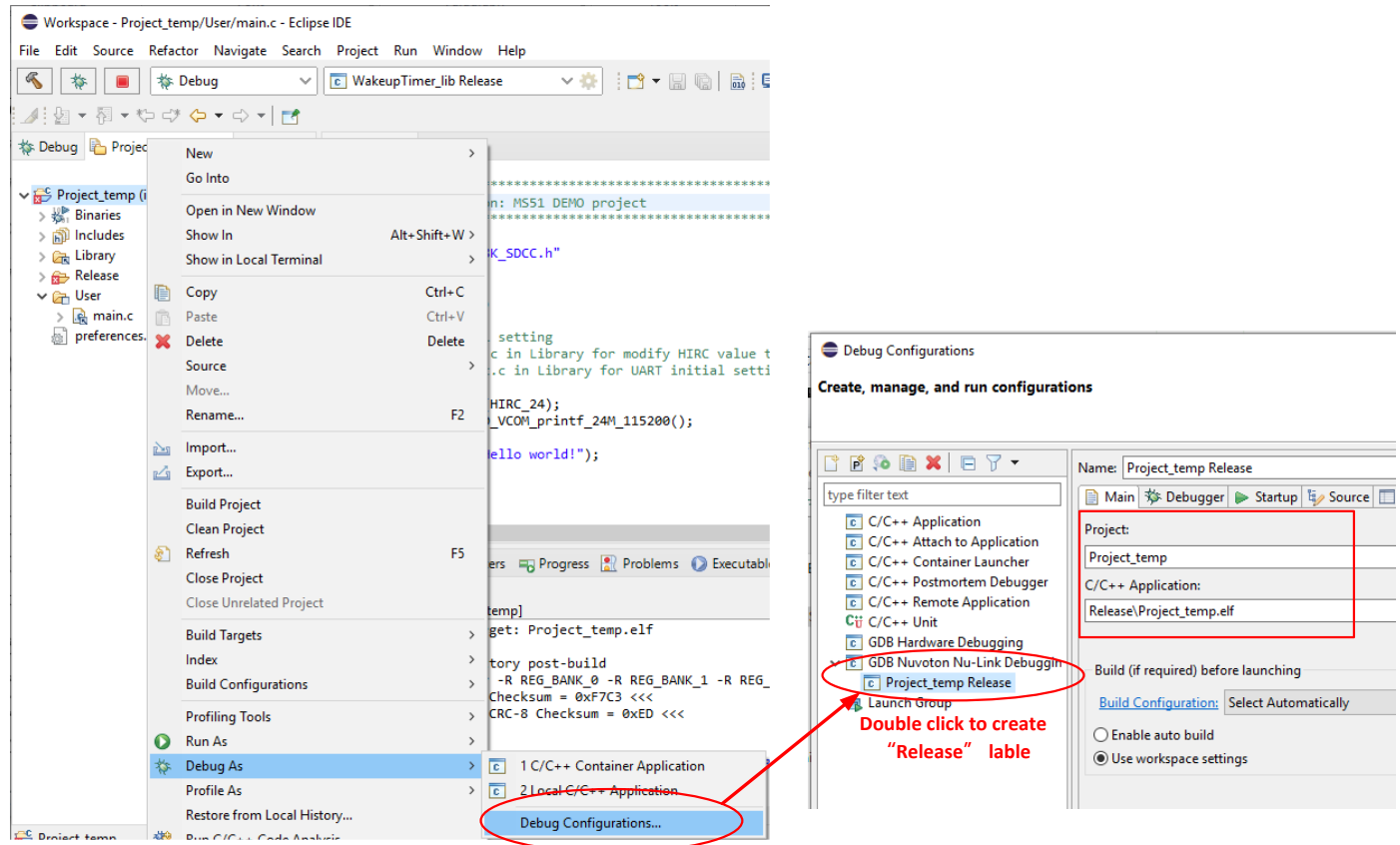
19:02:05 Build Finished. 0 errors, 0 warnings. (took 3s.845ms)

### 1.1.4 Debug

Right click on the Project name and select "Debug As" Debug Configurations.

In the Debug Configurations page, double-click GDB Nuvoton NuLink Debugging. The system will automatically generate the Project Release label. Check whether the .elf content of the Release project appears automatically in the C/C++ Application tag.

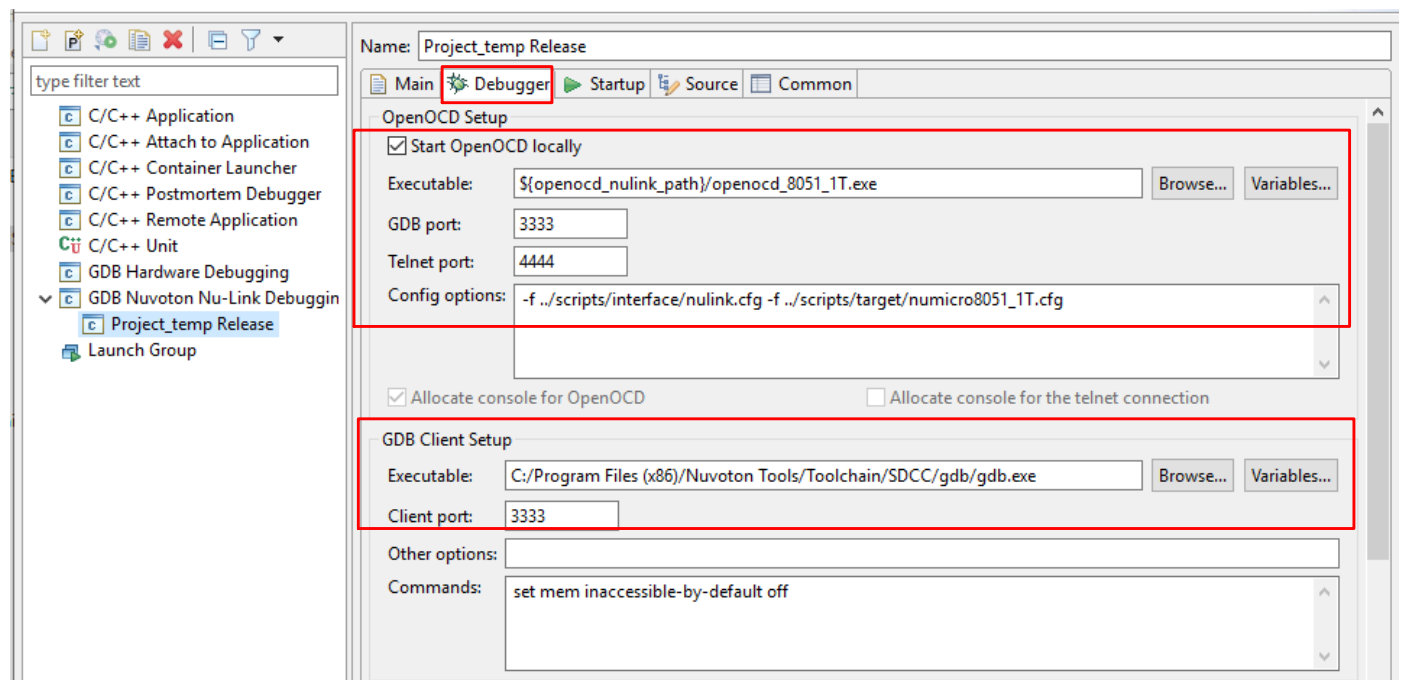
If there is no corresponding totally same name as the project, it means that there is an error in the build action, and you need to check the build/Compiler results.



Confirm the parameters in the "Debugger" page

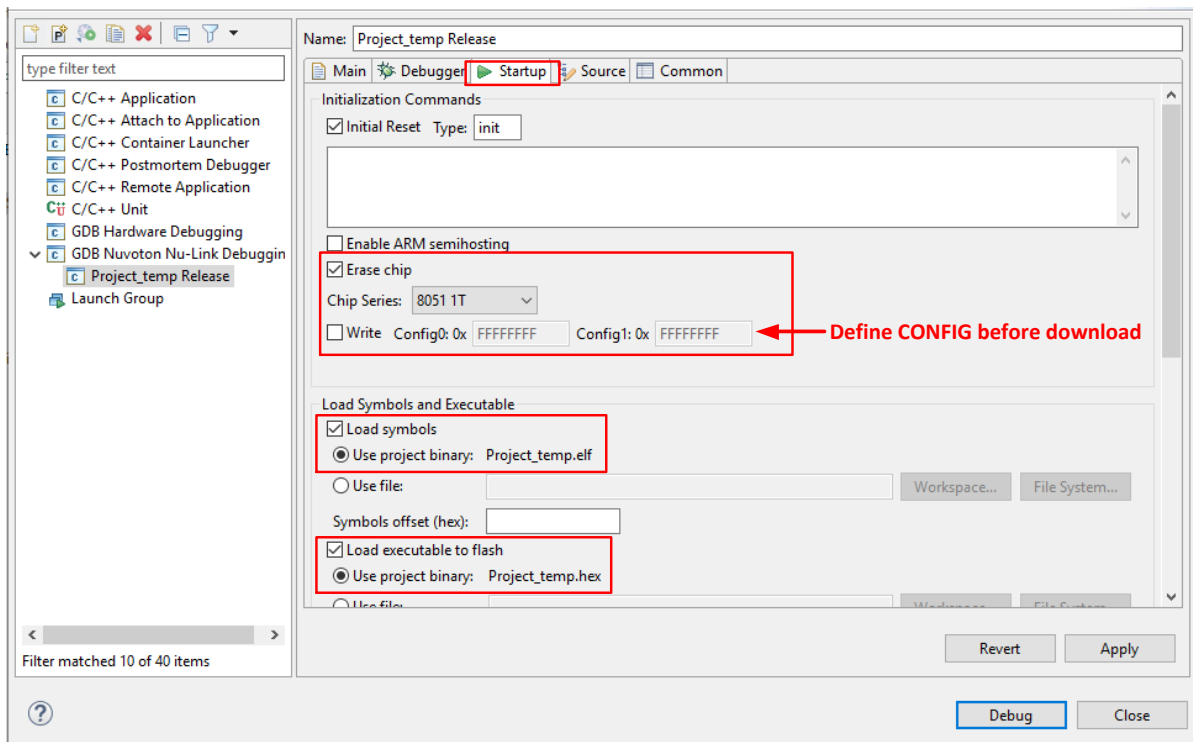
GDB Client Setup may need to manually fill in the parameters for the first time, and find the gdb.exe installation path under the corresponding Toolchain. The initial value is

**C:/Program Files (x86)/Nuvoton Tools/Toolchain/SDCC/gdb/gdb.exe**

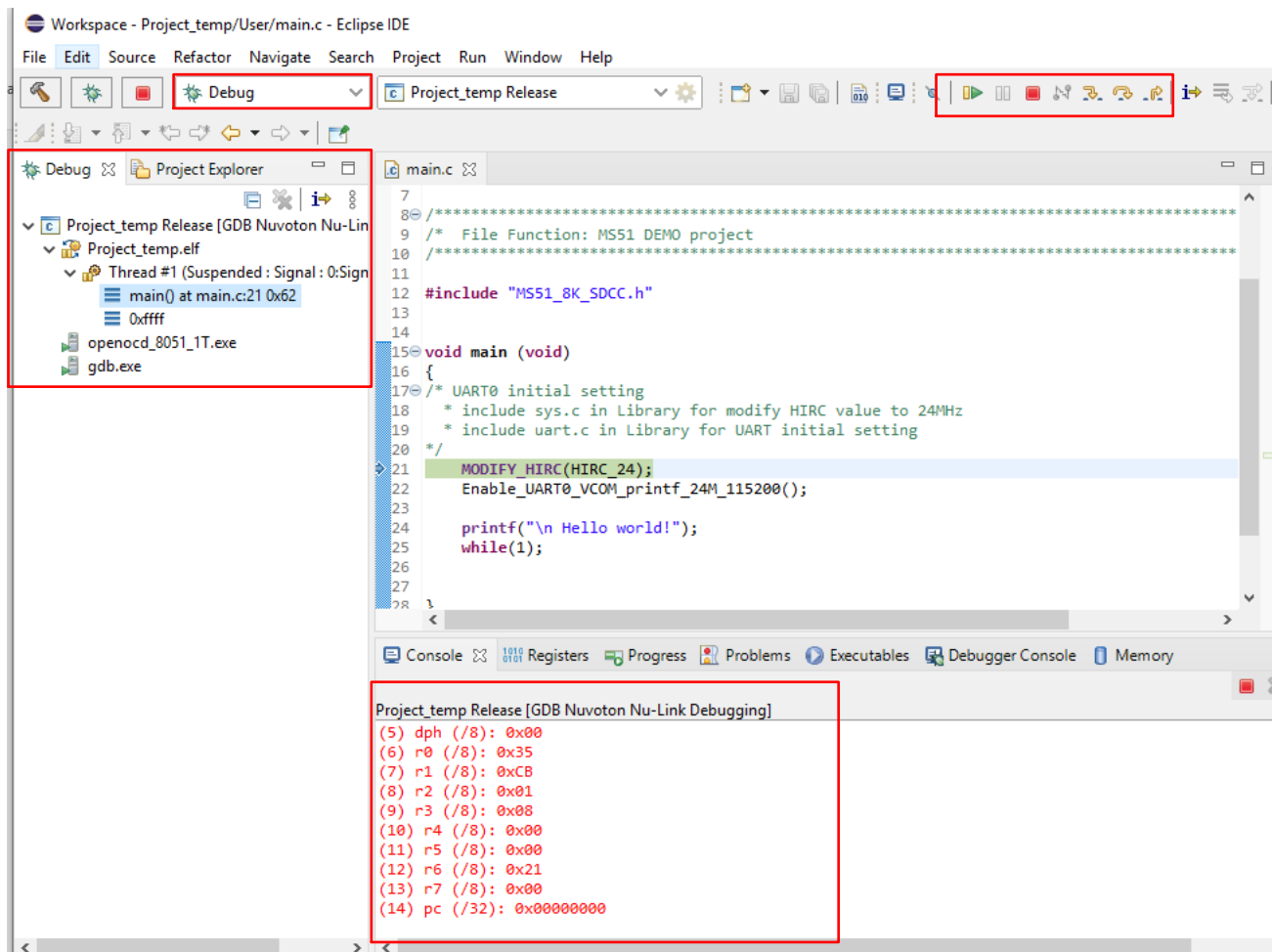


Confirm the parameters in the Startup page, if you need to change CONFIG, must **fill in manually**.

Confirm that the name of load .elf / .hex is consistent with Project, and the name is same as the project then click Debug

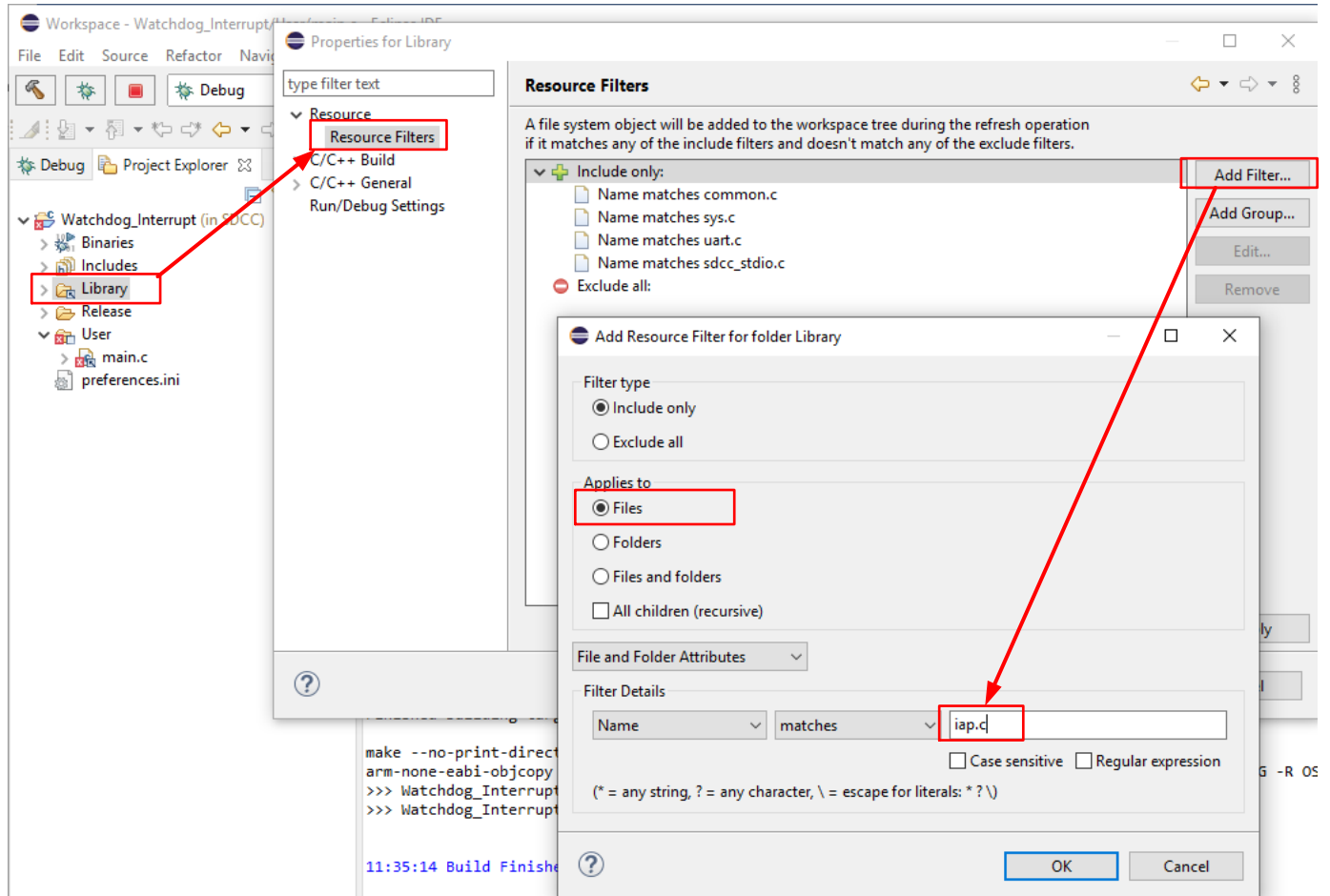


Display the following window and instructions to enter debug mode

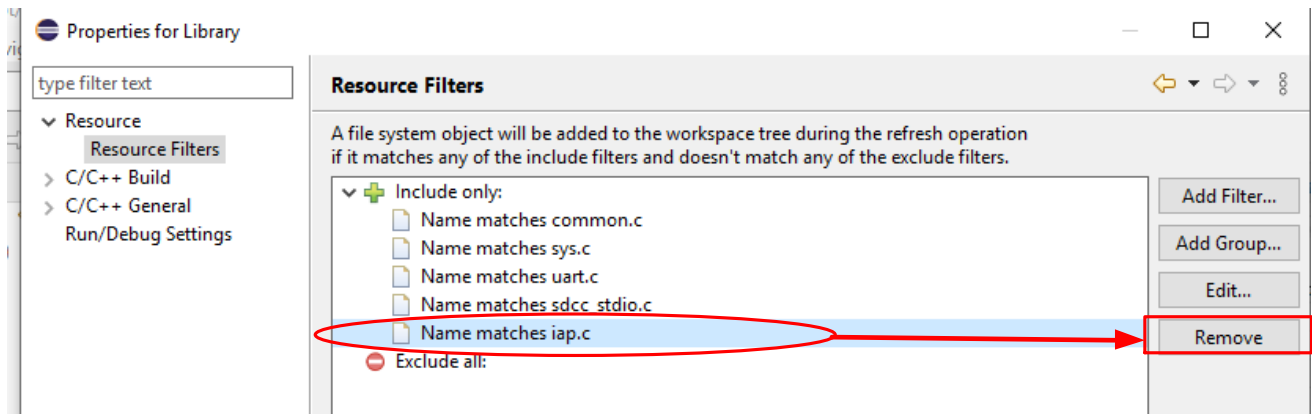


## 1.1.5 Add or Remove Library .c file

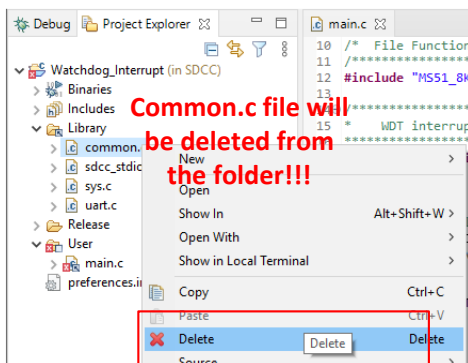
The lib .c file is added to the project for build, which is added by "Add Filter"



Remove lib, is Remove Filter



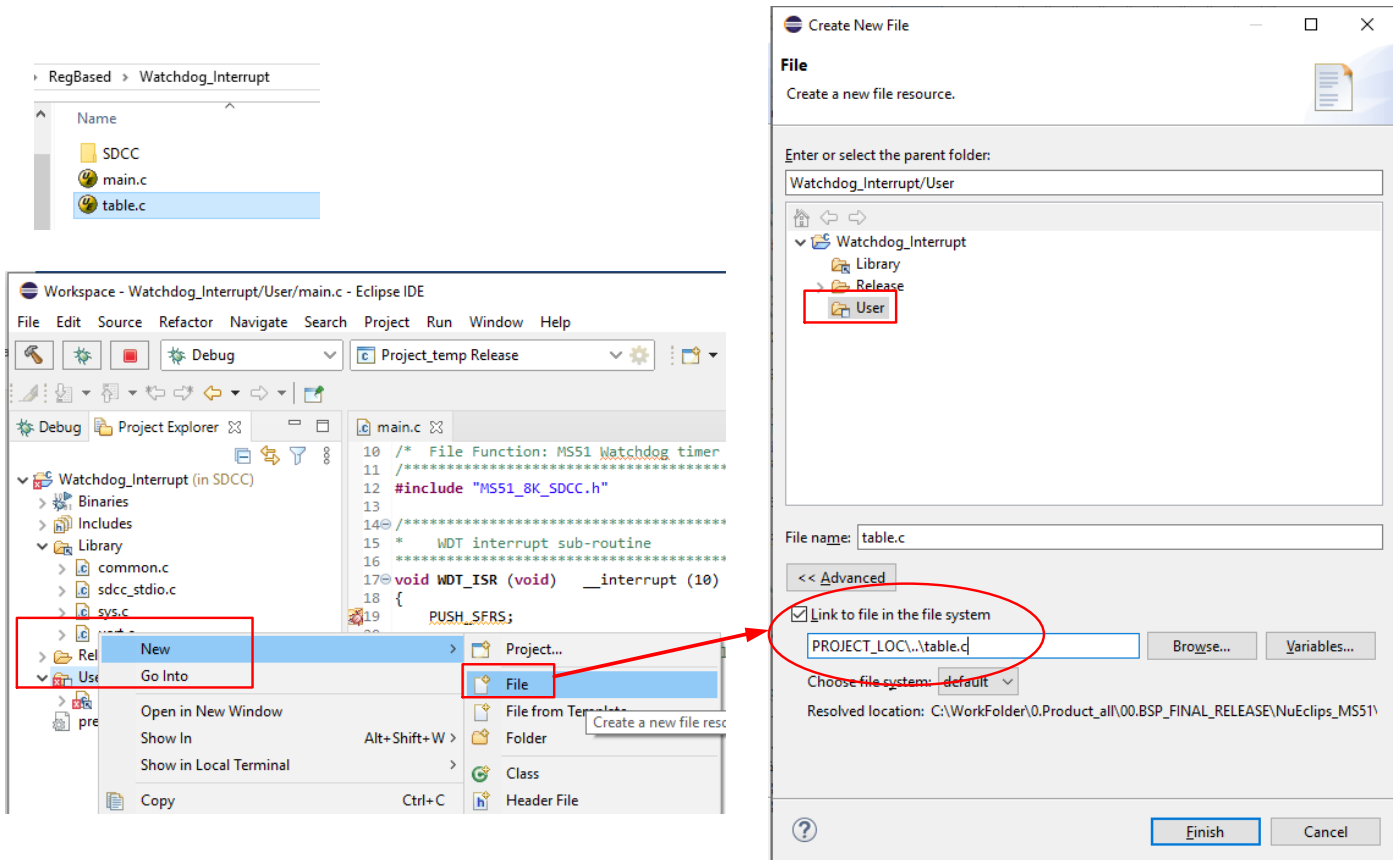
The remove action cannot select delete in Library, it will directly delete the source .c file in the folder



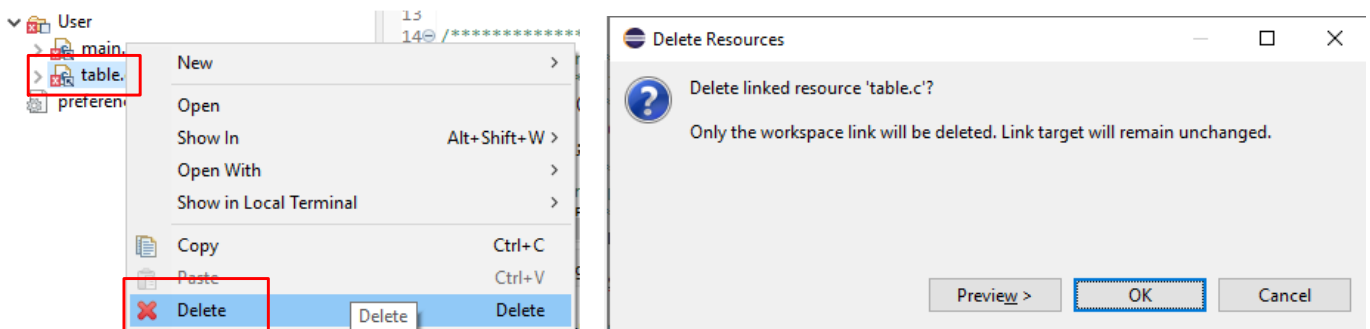
### 1.1.6 Add User Code

Add a new table.c, right-click user / New / File, and click Link to file in the file system on the Create New File page. It is recommended to fill in the relative path, the absolute path will cause the project location to move and cannot find the corresponding file issue.

**PROJECT\_LOC\.\<file name>**

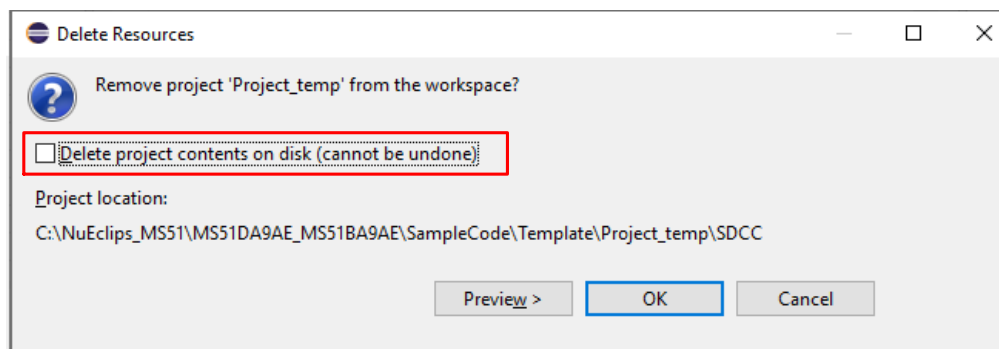


Delete user .c file



### 1.1.7 Close Project

Please **do not** check delete project contents on disk, SDCC project folder will be deleted.





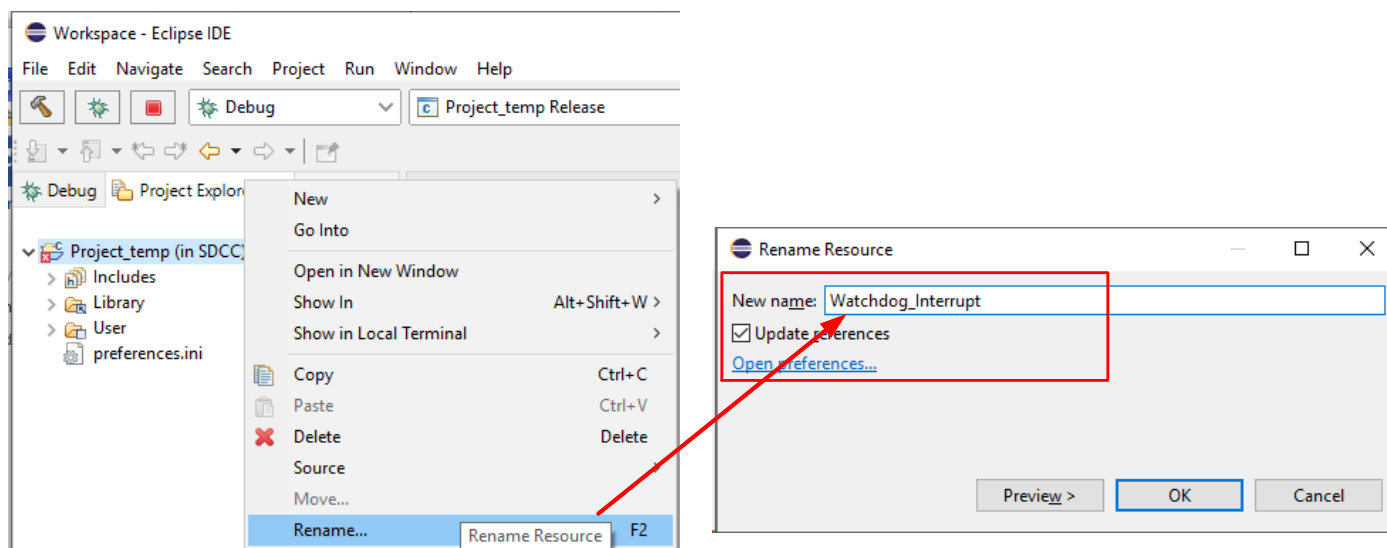
## 1.2 1.2 Create a new Project from an existing BSP project

### 1.2.1 Copy the existing SDCC project folder to the new Project folder

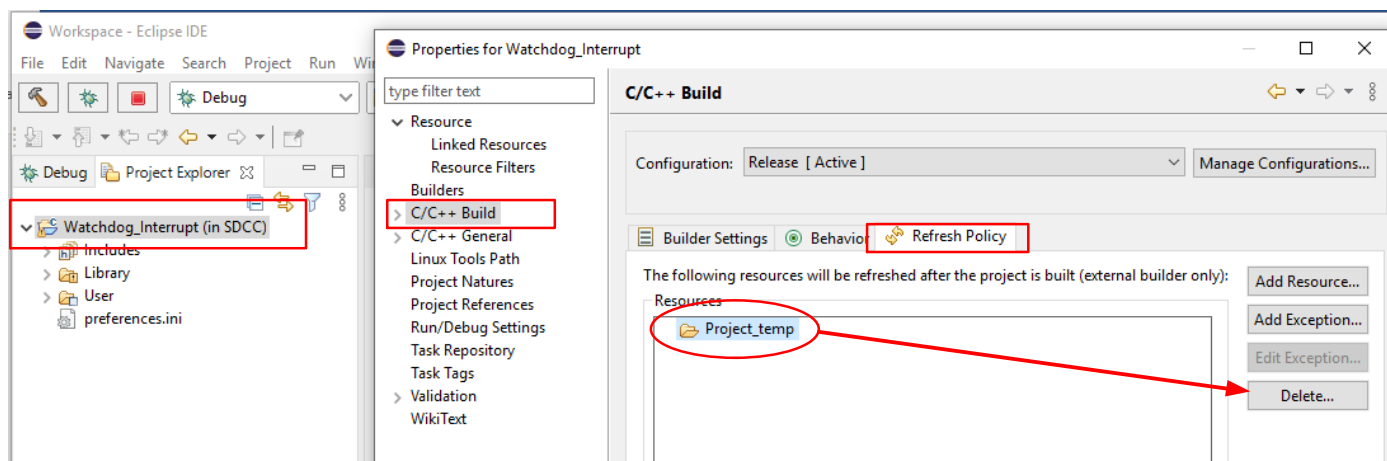
NuEclipse_MS51 > MS51DA9AE_MS51BA9AE > SampleCode > RegBased > Watchdog_Interrupt			
Name	Date modified	Type	Size
SDCC	2023/2/1 下午 06:04	File folder	
main.c	2023/2/1 下午 06:06	C File	3 KB

### 1.2.2 1.2.2 Import this project in Workspace

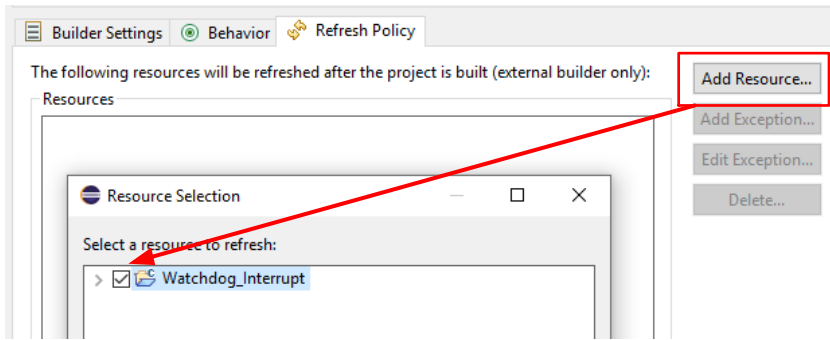
### 1.2.3 Rename Project



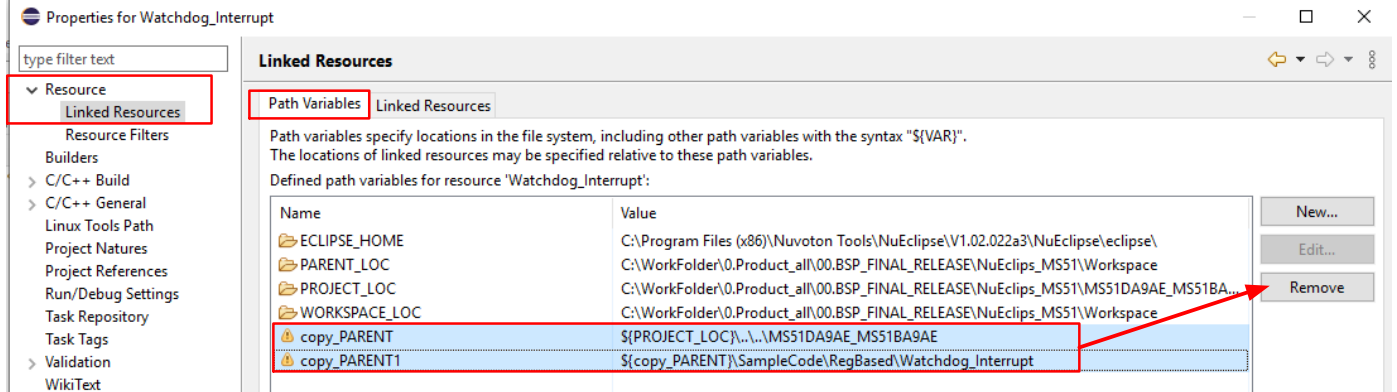
After confirming that the Project name has been changed, right-click Properties, modify Refresh Policy, and delete the old Resource



After deleting and adding, the system will automatically bring in the Project Resource after Rename, manually select it.



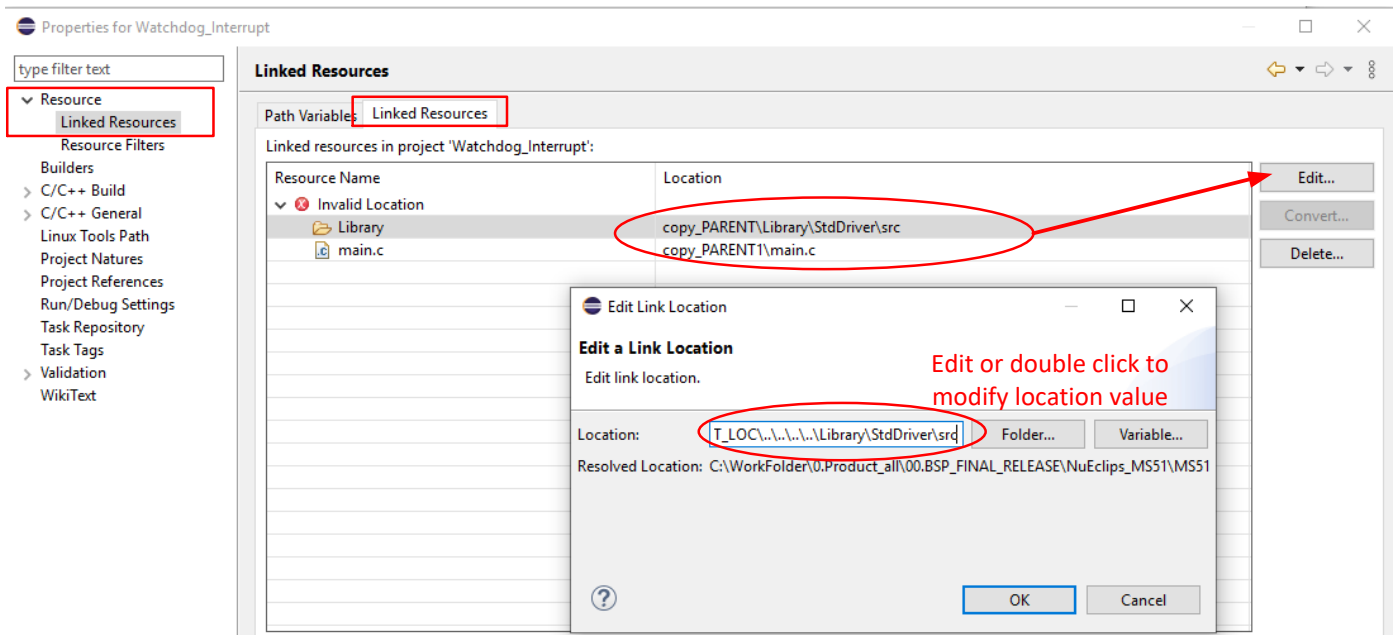
Remove the error Resource caused after Rename by eclipse system.



Correct the Resource linker settings, it is recommended to directly **paste** the following content

**PROJECT\_LOC\..\..\..\Library\StdDriver\src**

**PROJECT\_LOC\..\main.c**



Build the Project after all modify same as section 1.1.3 and Debug project same a section 1.1.4

Create a new Project (not recommended)

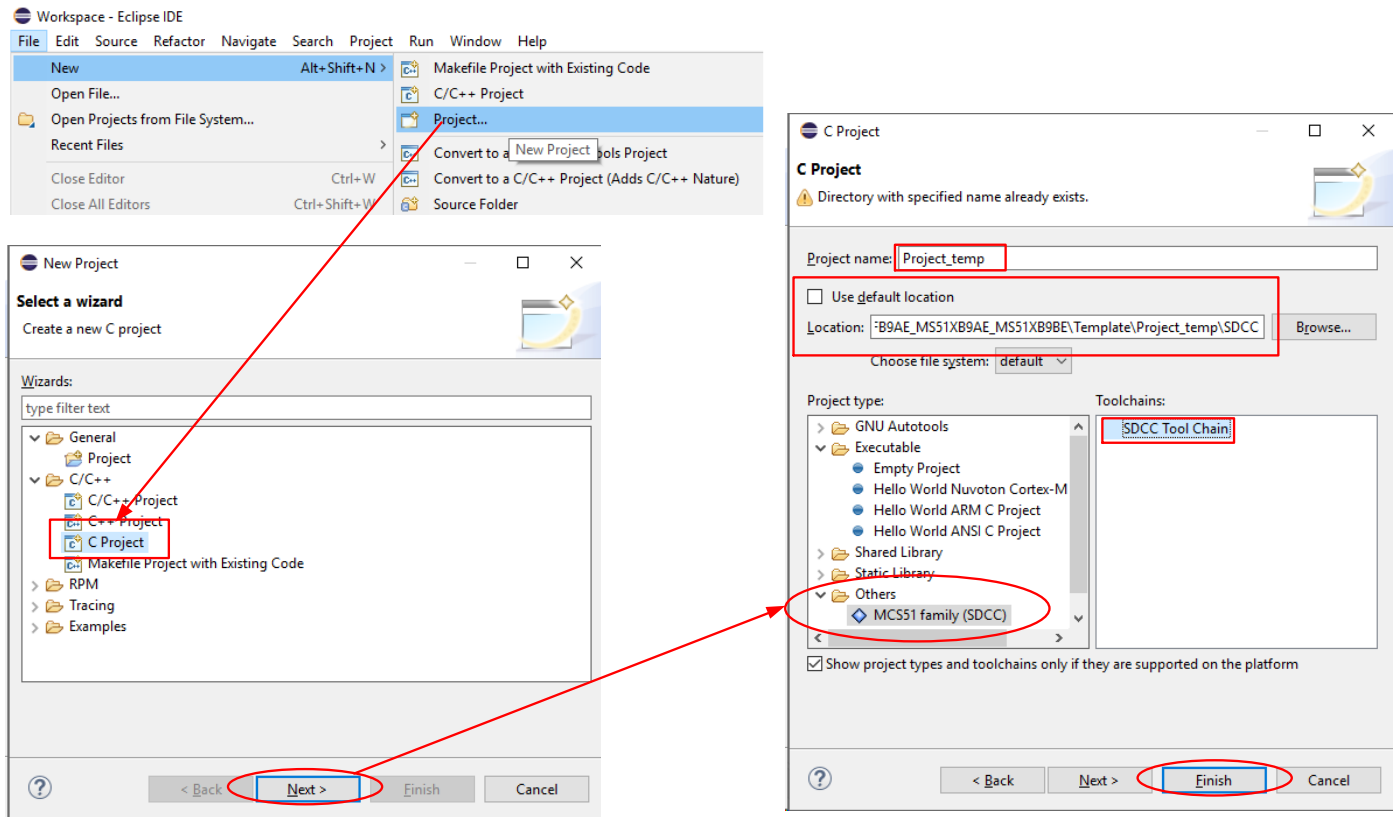
Create a new Project (not recommended)

Create a new Project (not recommended)

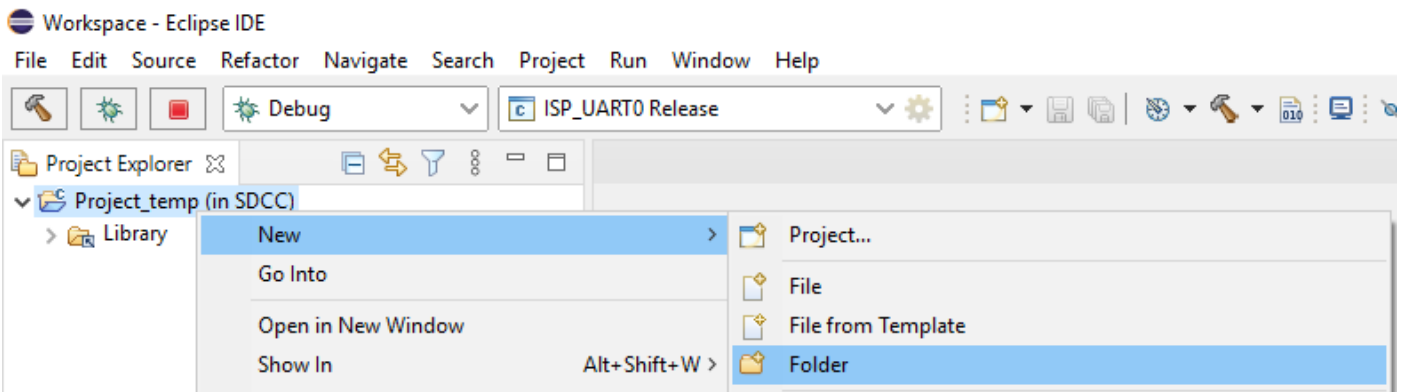
## 1.3 Create A New Project (Not Recommended)

### 1.3.1 New Project Define

Select “Others / SDCC Tool Chain”, Browse the location you want to put this project.

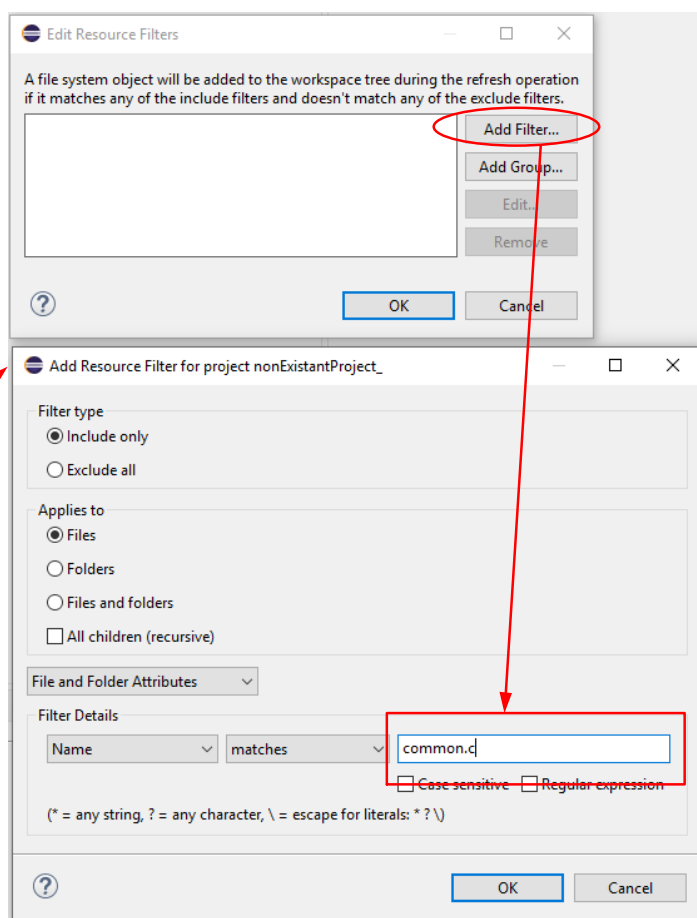
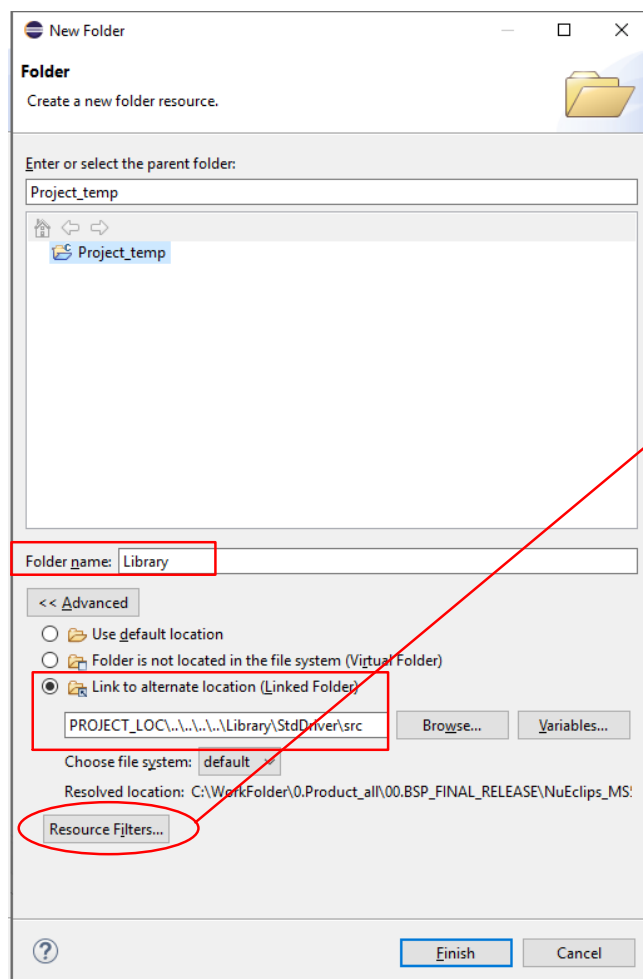


### 1.3.2 New Create Library And User Source Code

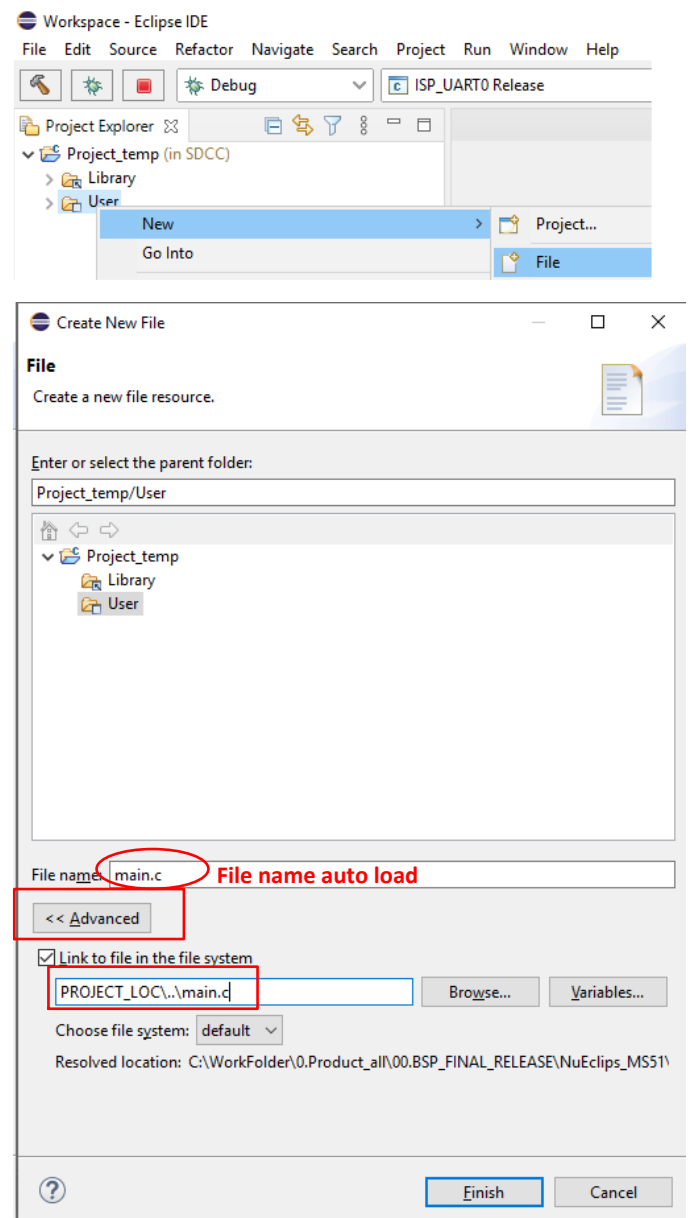
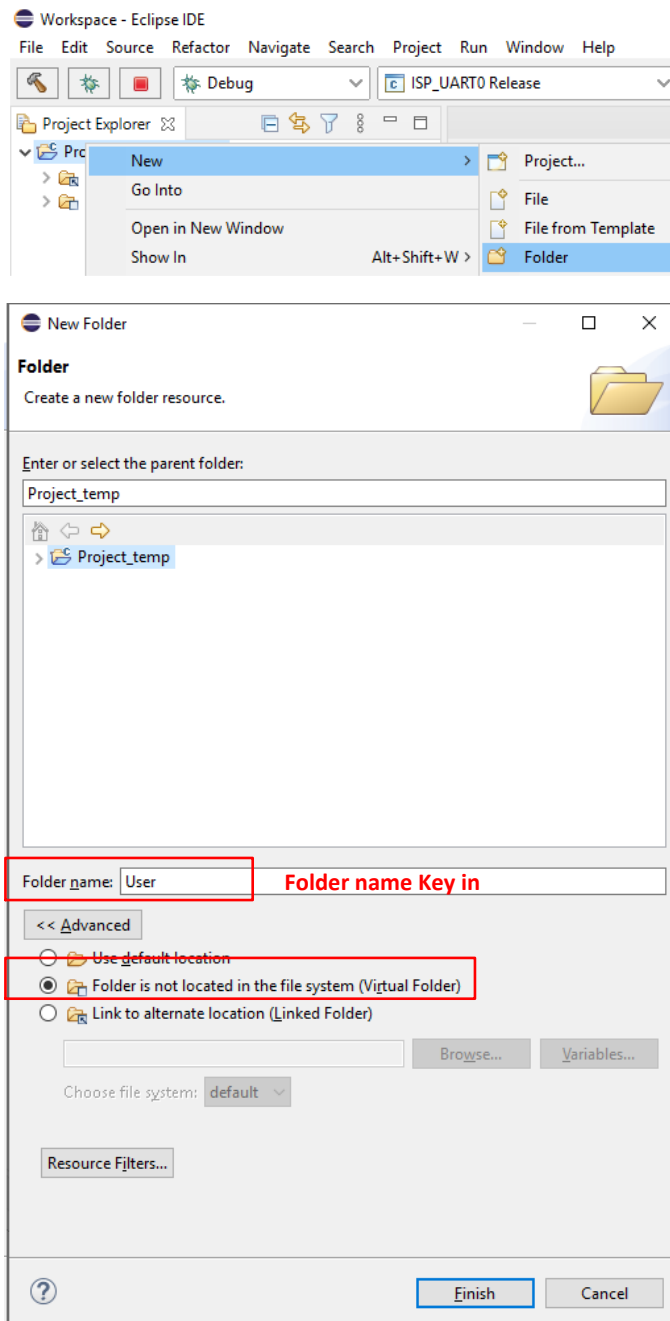


Link folder method. You can directly link the entire existing folder path, and automatically add all files in the folder without adding a filter.

Adding or reducing files is done in the way of Filter, not delete.



Link file method, first create a Virtual Folder, and then add a file link, each file is a single link, you can use delete to delete the link

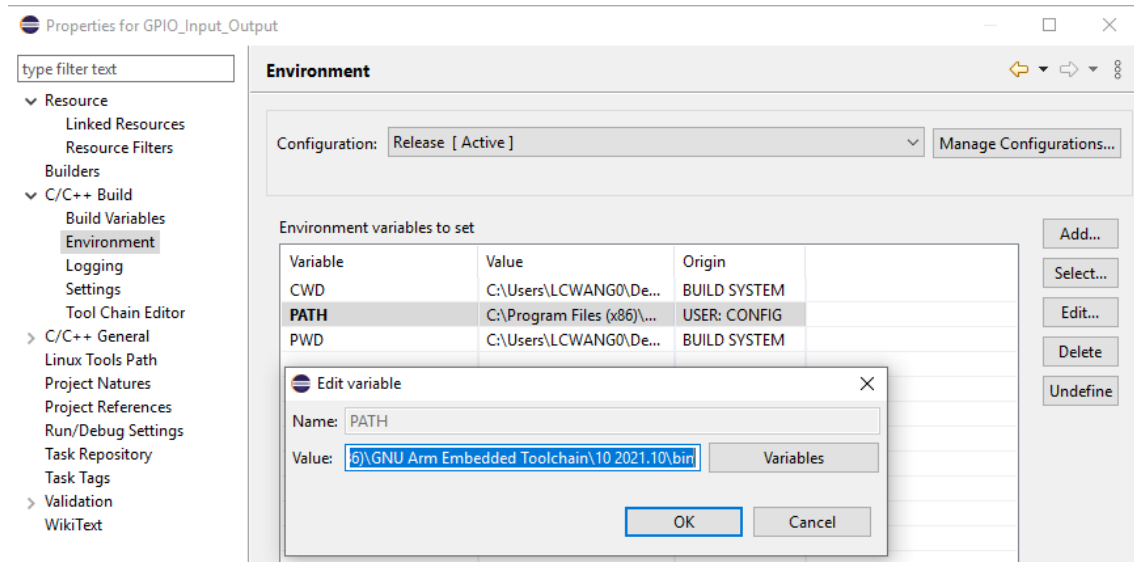


## 2 PROJECT PROPERTIES

### 2.1 Setting Project Properties

#### 2.1.1 Build / Environment / PATH

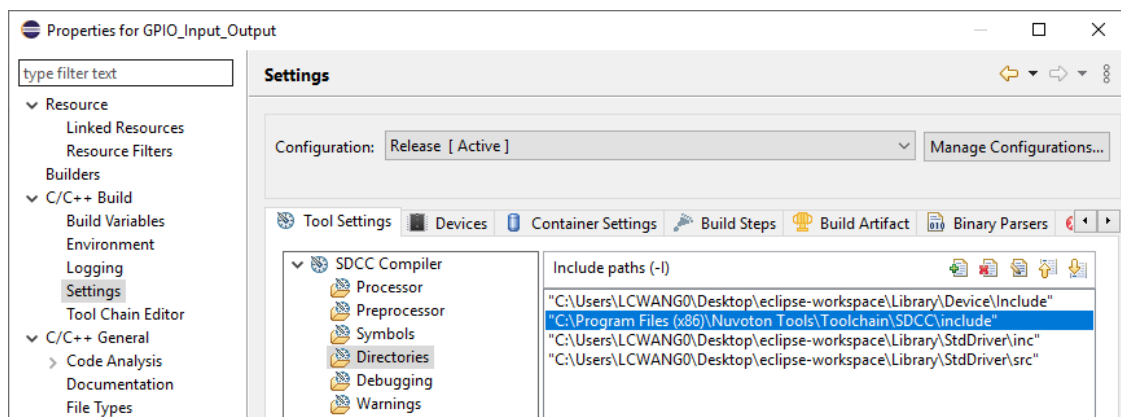
C:\Program Files (x86)\Nuvoton Tools\Toolchain\SDCC\bin;C:\Program Files (x86)\GNU ARM Eclipse\Build Tools\2.8-201611221915\bin;C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2021.10\bin



#### 2.1.2 Build setting

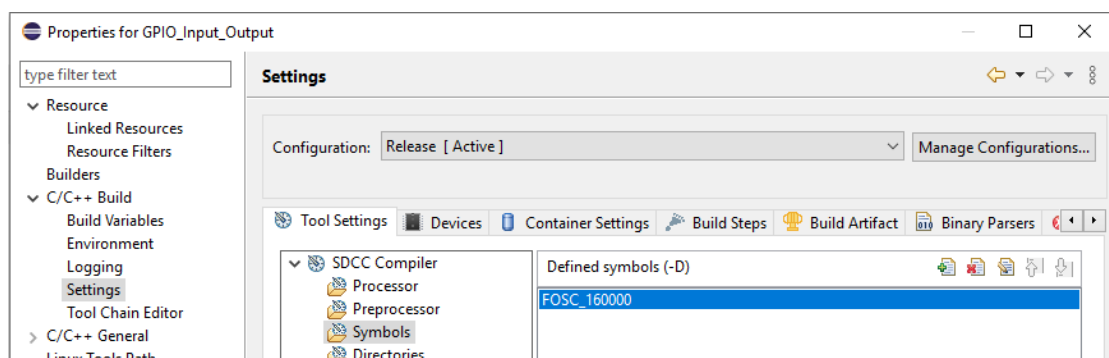
##### To added the include folder link

Settings / Tool Settings / SDCC Compiler / Directories Special added C:\Program Files (x86)\Nuvoton Tools\Toolchain\SDCC\include



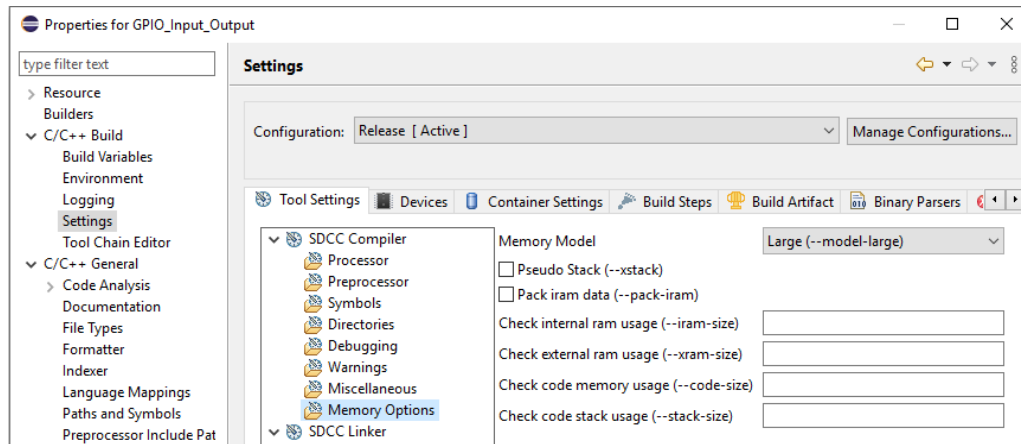
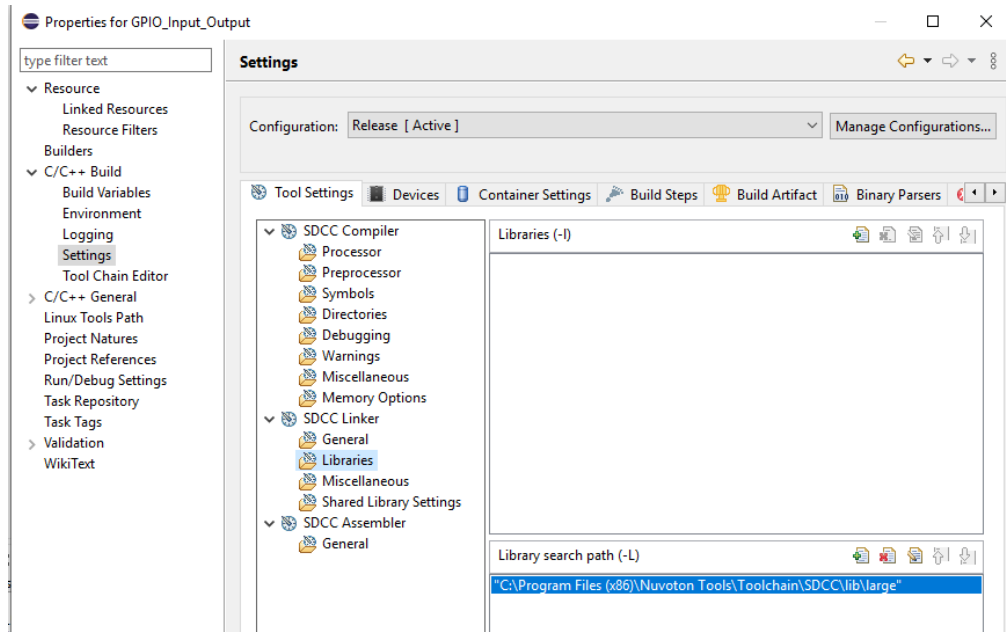
##### To added the Project Define

Directories/ Tool Settings / SDCC Compiler / Symbols 寫Define



## To setting default memory module

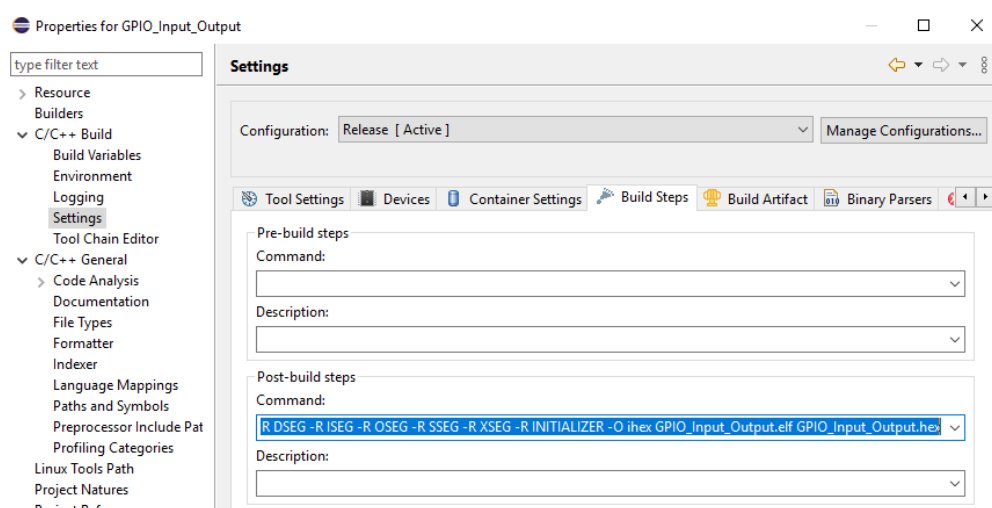
“Directories/ Tool Settings / SDCC Linker / Library” base on “Directories/ Tool Settings / SDCC Compiler / Memory Options” select SMALL or LARGE



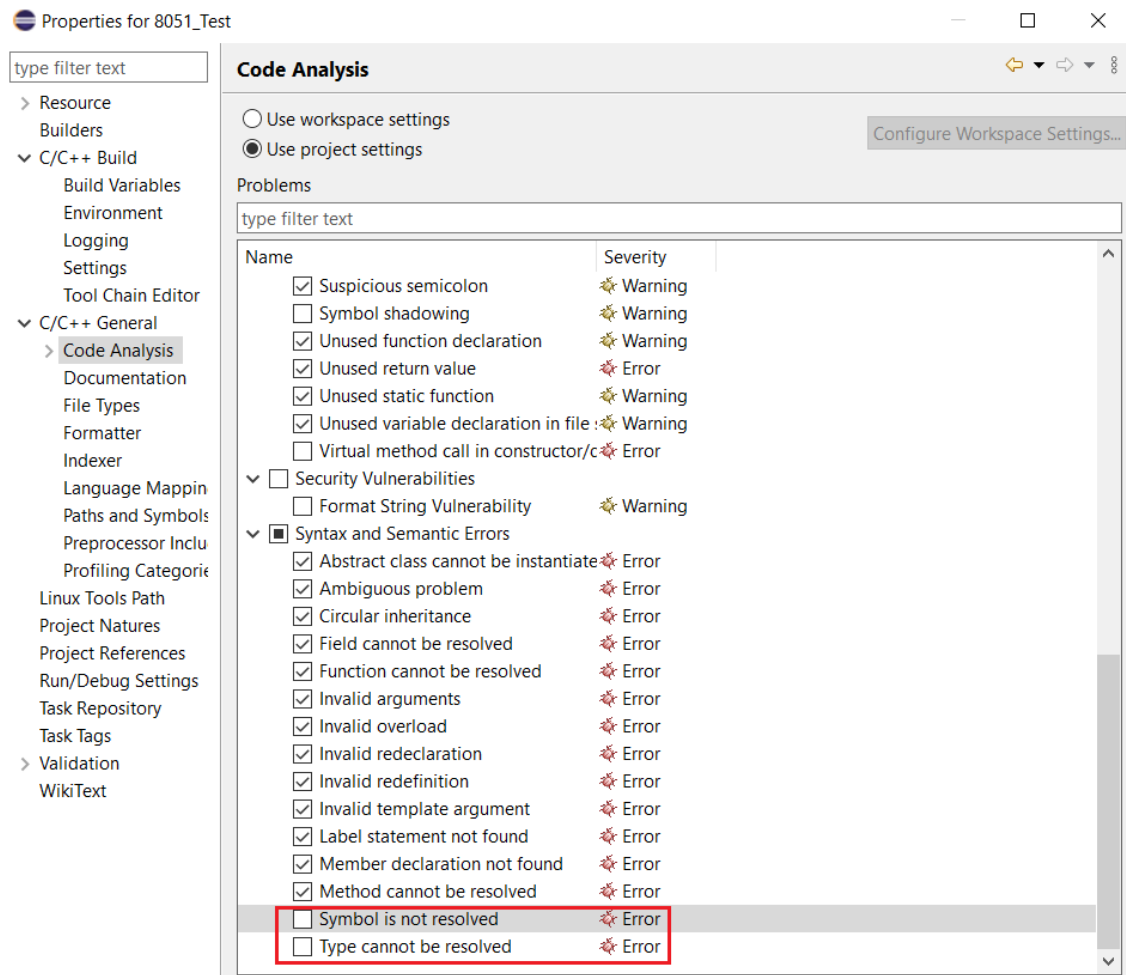
## Setting Post-build steps

Build Steps should add the command to generate hex files, elf and hex file names will be changed according to the project name of the folder, command as following:

```
arm-none-eabi-objcopy -R REG_BANK_0 -R REG_BANK_1 -R REG_BANK_2 -R REG_BANK_3 -R BSEG -R DSEG -R ISEG -R OSEG -R SSEG -R XSEG -R XISEG -R INITIALIZER -O ihex ${ProjName}.elf  
${ProjName}.hex;Hex2bin ${ProjName}.hex
```



Code Analysis cancels the two options of the red box





### 3 DEBUG ENVIRONMENT

#### 3.1.1 Set the location of different regions to

memory space of SDCC declaration	Data access offset define
code	(CODE address) + 0
data	(DATA address) + 0xF0000000
idata	(IDATA address) + 0xF1000000
xdata	(XDATA address) + 0xF2000000

#### 3.1.2 Use The Expression Window To View Variables

Fill in **\*(&<Variable Name / Address> + Data access offset define)**

For example: **\*(&SBUF+0xF0000000)**

Expression	Type	Value
(*)= *(&SBUF+0xF0000000)	volatile unsigned char	0 '\0'
(*)= *(&trimvalue16bit+0xF2000000)	unsigned int	12562

變更顯示格式

Expression	Type	Value
(*)= *(&itemp+0xF1000000)	volatile unsigned char	0x00
(*)= *(&xtemp+0xF2000000)	volatile unsigned int	0x55
(*)= *(&dtemp+0xF0000000)	volatile unsigned int	0x00000000
(*)= xtemp + 0xF2000000	unsigned int	0xf2000000

a. Use “memory window” to check memory area

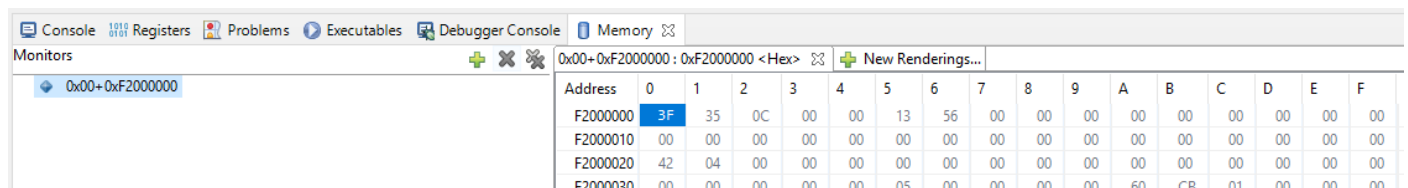
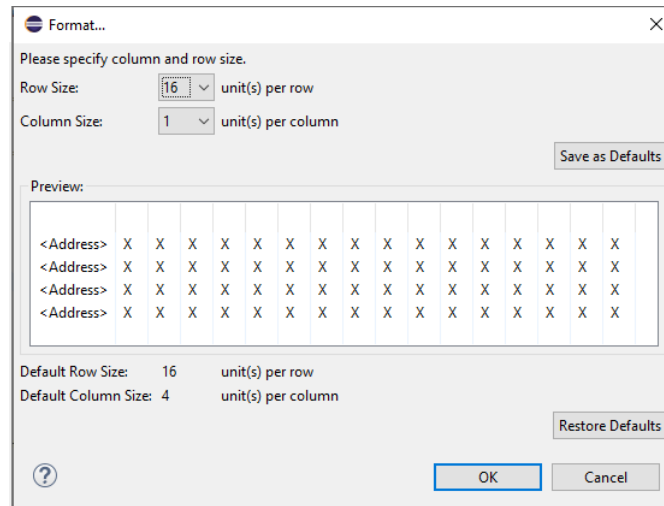
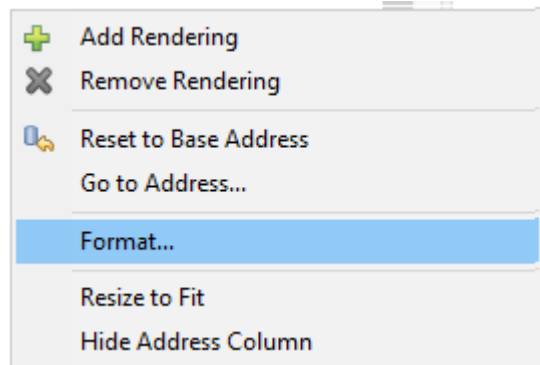
Use “Memory” or “Memory Browse” window to fill in <address> + < Data access offset define>

For example 0x00+0xF2000000 to check XDATA

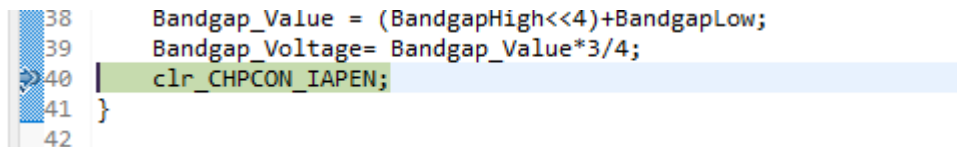
Address	0 - 3	4 - 7	8 - B	C - F
F2000000	34CB0100	00790E80	00000000	00000000
F2000010	00000000	00000000	00000000	00000000
F2000020	08000000	00000000	00000000	000060CB
F2000030	01006B13	807A0000	2E09510A	0000004A
F2000040	0937BD08	00000000	00000000	00000000
F2000050	00000000	00000000	00000000	00000000
F2000060	00000000	00000000	00000000	00000000
F2000070	00000000	00000000	00000000	00000000
F2000080	00000000	00000000	00000000	00000000
F2000090	00000000	00000000	00000000	00000000
F20000A0	00000000	00000000	00000000	00000000

The Memory window changes the display mode through the Format option.

Right click on the Memory window area

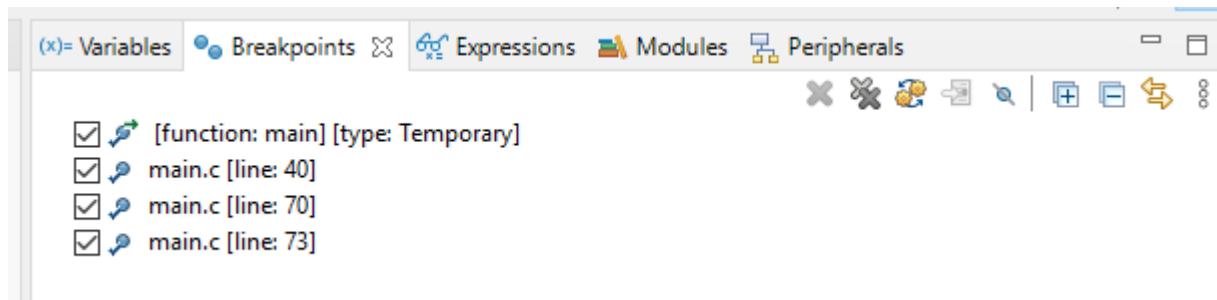


### 3.1.3 Set Break Point



The BreakPoints window lists all the settings, and all breakpoints can be deleted through XX.

8051 MCU only can set up to **8 break points**! Too many settings will **NOT cause alerts**, but will cause errors in the debug environment.



## 4 APPLICATION TIPS

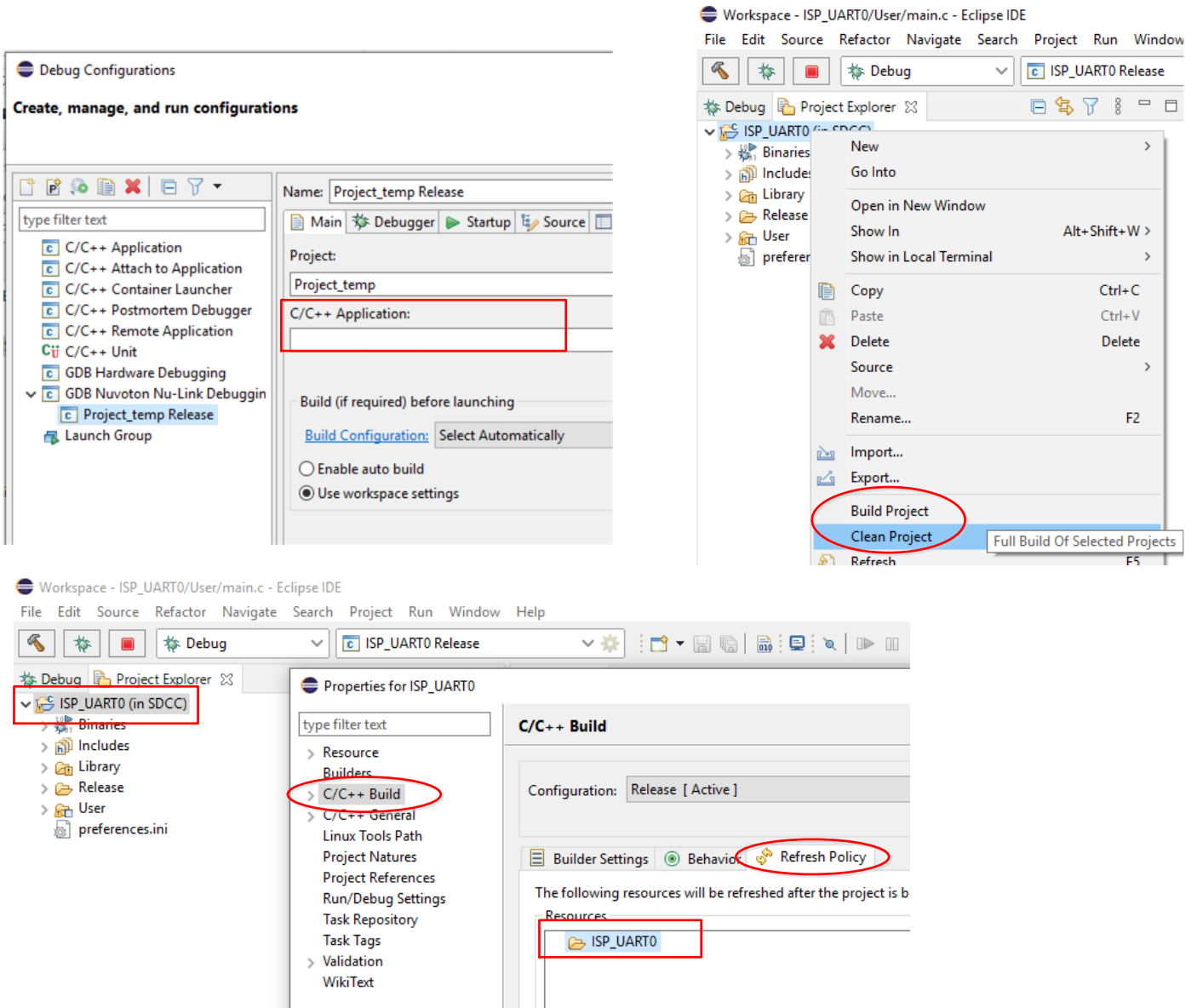
1. File extensions Name must be in lowercase.
2. Interrupt Vector should be placed in main.c
3. After changing the code content, be sure to re-archive before building.



4. If debug mode does not display the correct .elf

Clean Project and Build Project again until there is no error.

Confirm that the project Refresh Policy is established correctly, and the name is consistent with the Project.

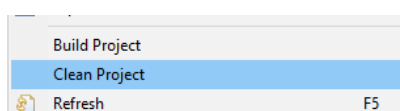


5. After Build check item

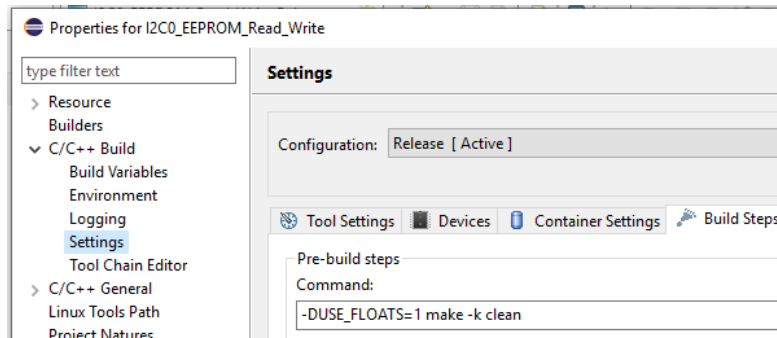
For example, after importing the project, there is no response after clicking "Project->Build Project" / The phenomenon of responding after clicking "Project->Build Project" several times / The project that reports errors first time, but then show no error if click Build Project again.

Solution:

- 1) If there is no complete information or no response during any build, click "Clean Project" to clear the project record



- 2) Since The Clean Project step must be performed every time before re-Build Project. Add “make –k clean” command in Pre-build steps.



- 3) Delete the .setting / Release folder in the project folder, and then re-build the Project until the words 0 error 0 warning appear, and check sdcc – debug out and all associated .c files have .rel generated.

```
Building target: ADC_Bandgap_VDD.elf
Invoking: SDCC Linker
sdcc --debug --out-fmt-elf -L"C:\Program Files (x86)\Wuvoton Tools\Toolchain\SDCC\lib\large" -o "ADC_Bandgap_VDD.elf" ./User/main.rel ./Library/IAP.rel ./Library/bod.rel ./Library/common.rel ./Library/delay
Finished building target: ADC_Bandgap_VDD.elf

make --no-print-directory post-build
arm-none-eabi-objcopy -R REG_BANK_0 -R REG_BANK_1 -R REG_BANK_2 -R REG_BANK_3 -R BSEG -R DSEG -R ISEG -R OSEG -R SSEG -R XSEG -R XISEG -R INITIALIZER -O ihex ADC_Bandgap_VDD.elf ADC_Bandgap_VDD.hex;Hex2bin ADC
>>> ADC_Bandgap_VDD.bin Checksum = 0x9A29 <<<
>>> ADC_Bandgap_VDD.bin CRC-8 Checksum = 0xA6 <<<

10:46:23 Build Finished. 0 errors, 0 warnings. (took 115.588ms)
```

- 4) When the following build information appears, you also need to clean and then build. For example “Nothing be done”

```
CDT Build Console [ADC_Bandgap_VDD]
11:42:47 **** Incremental Build of configuration Release for project ADC_Bandgap_VDD ****
make -k all
make --no-print-directory pre-build
DUSE_FLOATS=1

make --no-print-directory main-build
make[1]: Nothing to be done for 'main-build'.

11:42:48 Build Finished. 0 errors, 0 warnings. (took 937ms)
```