

# NuEclipse SDCC Quick Start V1.02

## CONTENTS

<b>1</b>	<b>PROCESS TO SETUP A PROJECT.....</b>	<b>3</b>
1.1	Start an Existing BSP project .....	3
1.1.1	Eclipse Workspace Always Import Project.....	3
1.1.2	Import Project.....	4
1.1.3	Compiler / Build.....	5
1.1.4	Debug.....	6
1.1.5	Add or Remove Library .c file .....	8
1.1.6	Add User Code.....	9
1.1.7	Close Project.....	10
1.2	Create a new Project from an existing BSP project .....	11
1.2.1	Copy the existing SDCC project folder to the new Project folder .....	11
1.2.2	Import this project in Workspace .....	11
1.2.3	Rename Project .....	11
1.3	Create A New Project (Not Recommended) .....	13
1.3.1	New Project Define.....	13
1.3.2	New Create Library And User Source Code.....	13
<b>2</b>	<b>PROJECT PROPERTIES .....</b>	<b>16</b>
2.1	Setting Project Properties .....	16
2.1.1	Build / Environment / PATH.....	16
2.1.2	Build setting .....	16
<b>3</b>	<b>DEBUG ENVIRONMENT .....</b>	<b>19</b>
3.1	Entry Debug Mode .....	19
3.1.1	Always using GDB Nuvoton Nu-link Debugging options window into debug mode.....	19
3.2	Check Item before into Debugging.....	20
3.3	Variables Display .....	22
3.3.1	Set The Location Of Different Regions To Display .....	22
3.3.2	Use The Expression Window To View Variables.....	23
3.4	Set Break Point.....	25
<b>4</b>	<b>APPLICATION TIPS &amp; FAQ .....</b>	<b>26</b>
4.1	File Name .....	26
4.2	Coding Rule .....	26

4.3 Save All.....26

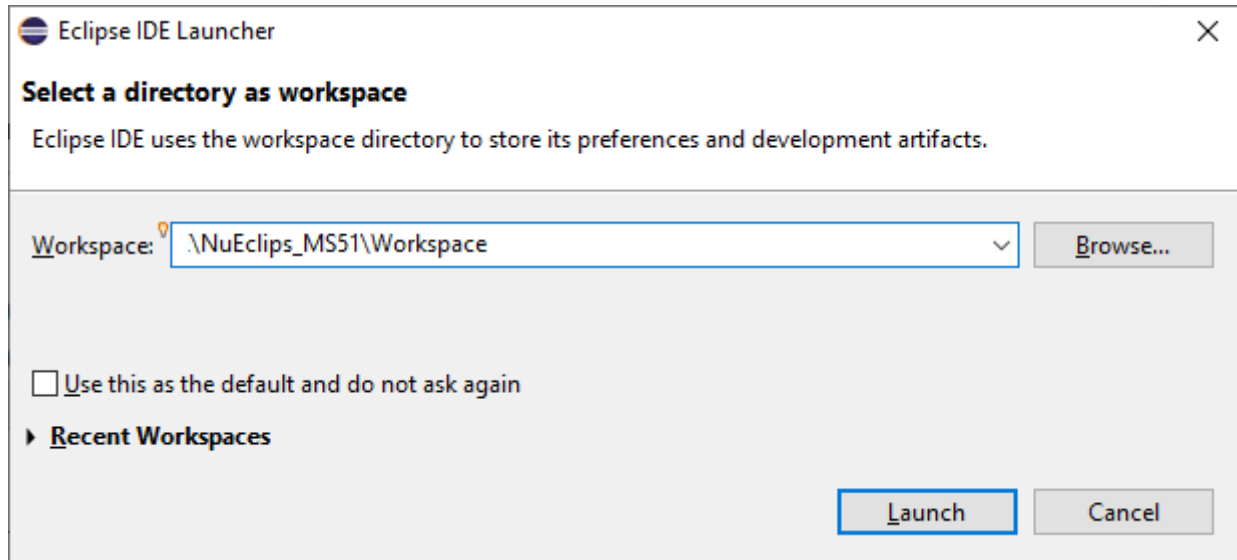
4.4 Check Item Before into Debugging .....27

# 1 PROCESS TO SETUP A PROJECT

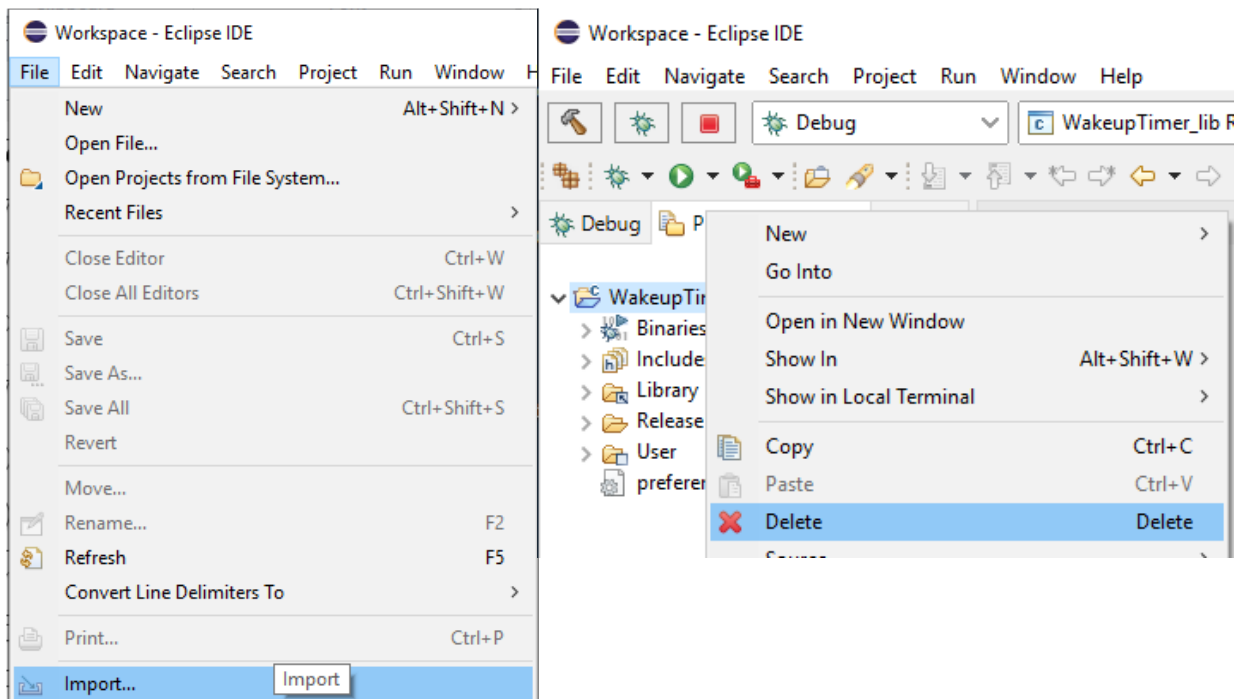
## 1.1 Start an Existing BSP project

### 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

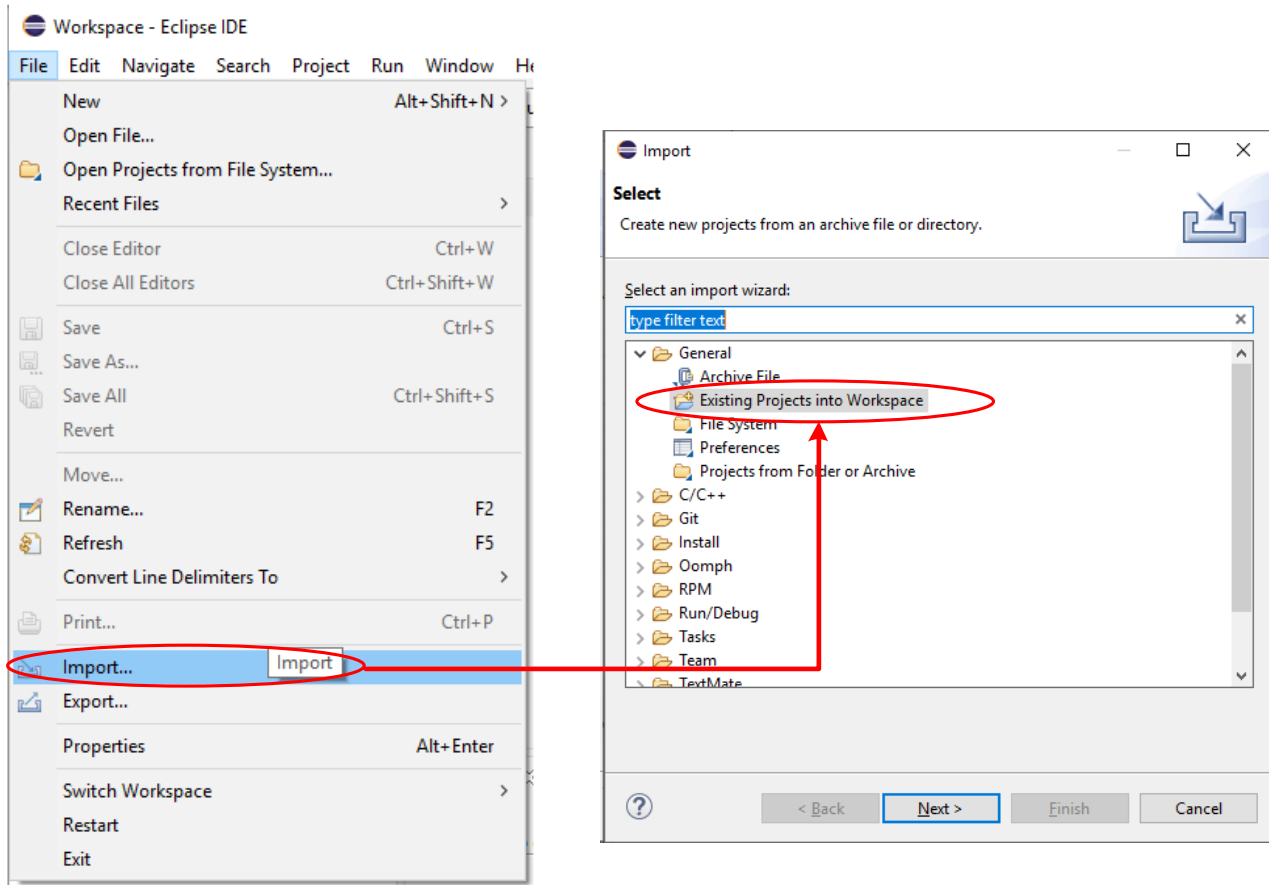


When opening a project, use “import”, and to closing it, use “delete project” in workspace

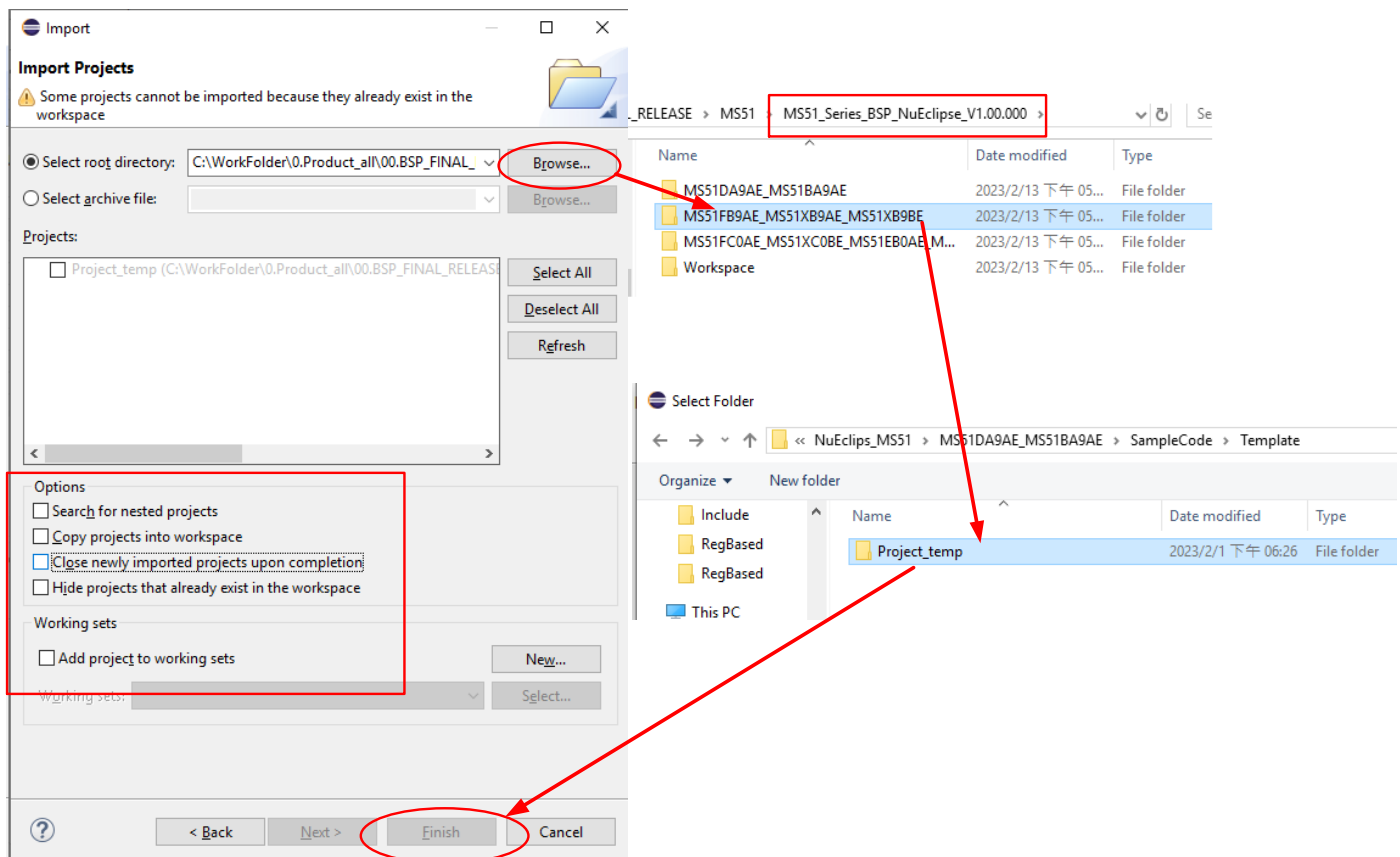


## 1.1.2 Import Project

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 Eclipse IDE interface during a build process. The top-left pane shows the 'Project Explorer' with 'Project\_temp (in SDCC)' expanded, listing 'Includes', 'Library', 'User', and 'preferences.ini'. A red arrow points from this project to the 'Build Project' option in the context menu. The bottom-left pane shows the 'Project Explorer' again, but with 'Binaries', 'Release', and 'main.c' added to the 'User' folder. The bottom-right pane shows the 'CDT Build Console' output, which includes the command 'make --no-print-directory post-build' and the resulting checksums for 'Project\_temp.bin'. The final line of the console output, '19:02:05 Build Finished. 0 errors, 0 warnings. (took 3s.845ms)', is circled in red.

```
Workspace - Eclipse IDE
File Edit Navigate Search Project Run W
Debug Project Explorer
Project_temp (in SDCC)
  Includes
  Library
  User
  preferences.ini

Project Explorer
Project_temp (in SDCC)
  Binaries
  Includes
  Library
  Release
  User
  main.c
  preferences.ini

main.c
7
8 /* File Function: MS51 DEMO project
9
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

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)
```

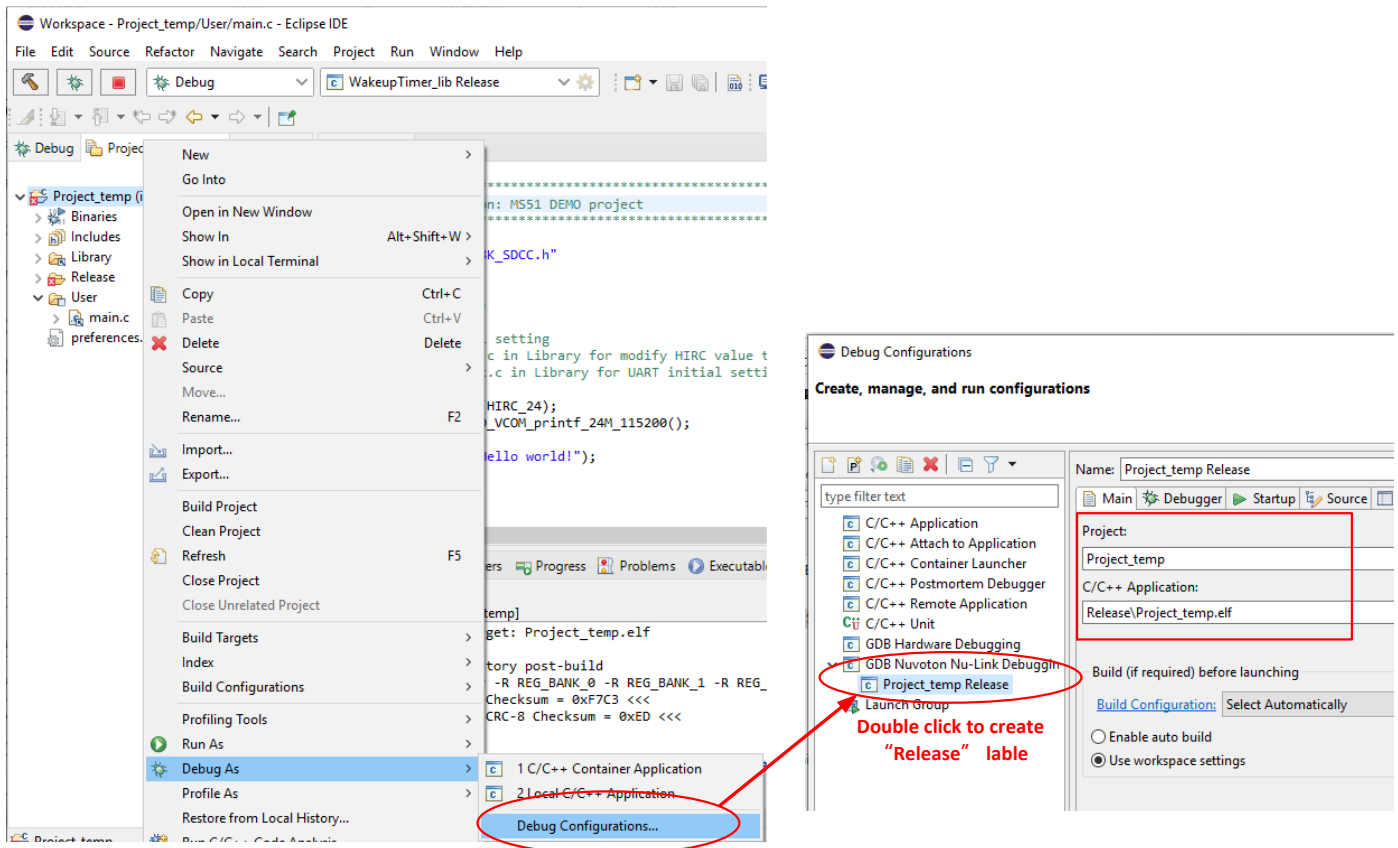
### 1.1.4 Debug

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.

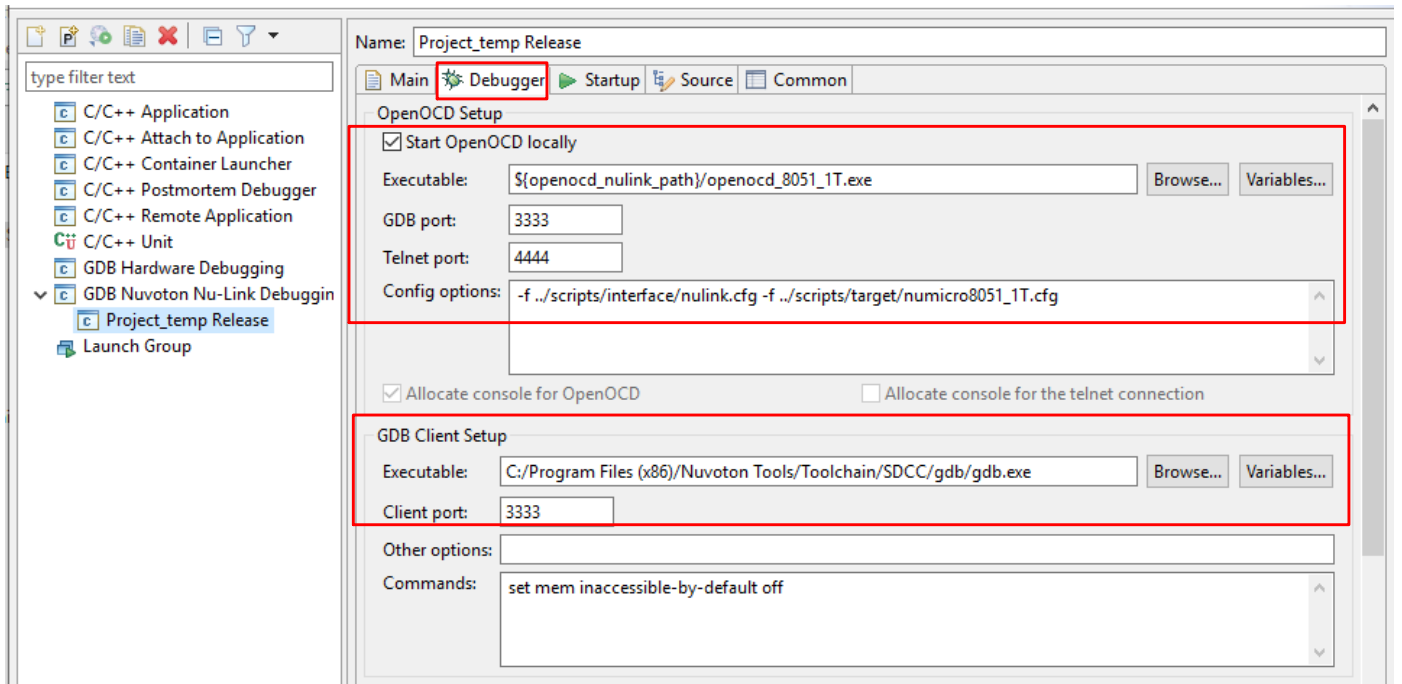
**Note:** You can only connect **ONE** Nu-Link debug tool to your computer at a time, regardless of whether the other Nu-Link is connected to the target IC or not.



Confirm the parameters in the "Debugger" page

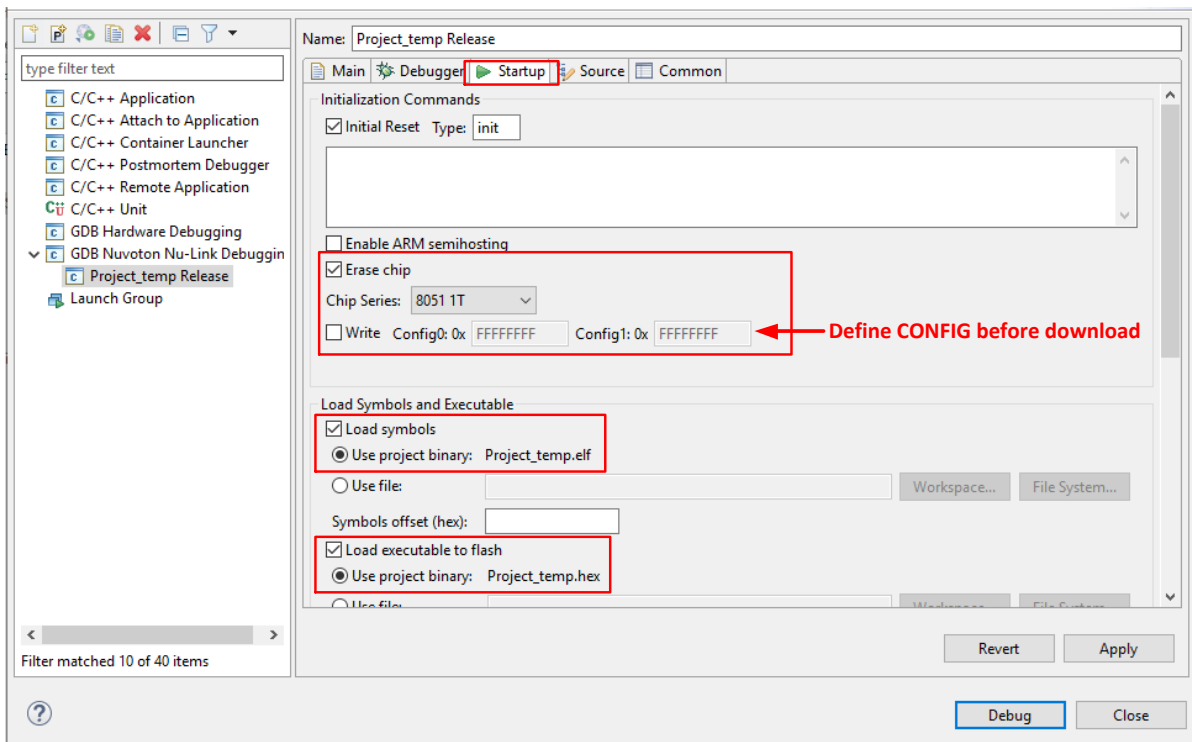
GDB Client Setup may need to manually fill in the parameters for the first time, and find the gdb.exe installation path under the corresponding Toolchain. The initial value is

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

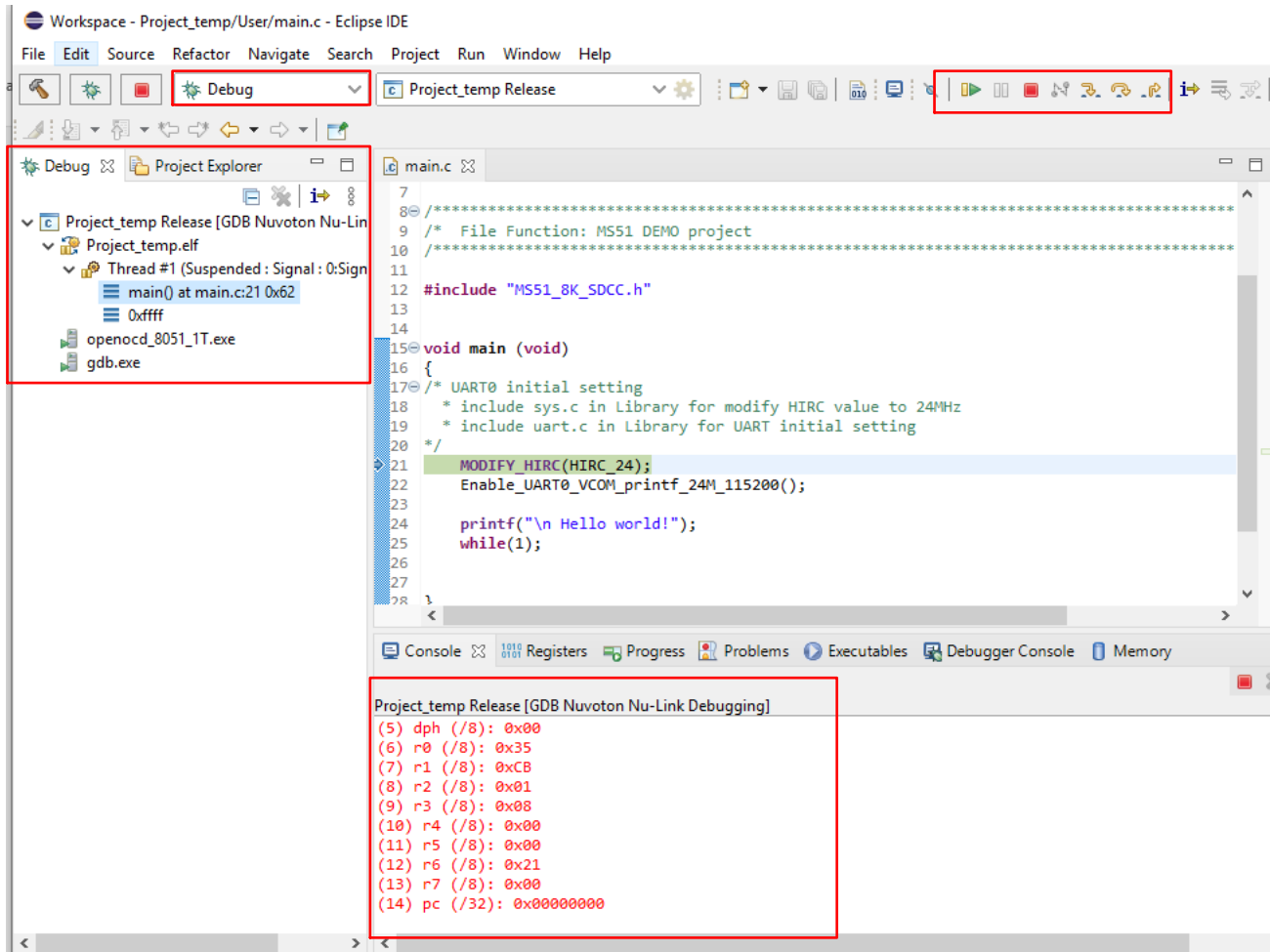


Confirm the parameters in the Startup page, if you need to change CONFIG, must **fill in manually**.

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

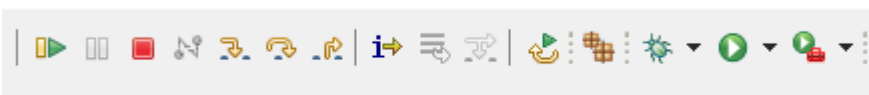


Following show an complete debugging download information with keywords:

Device Name , Erase complete, write bytes from file, target halt from 0x00

```
UART4_Printf Release [GDB Nuvoton Nu-Link Debugging]
Info : Device Name: MS51PC0AE
Info : IDCODE: 0x00DA5332
Info : IDCODE: 0x00000100
Info : Device Name: MS51PC0AE
undefined debug reason 7 - target needs reset
NuMicro.cpu: target state: halted
target halted due to debug-request, pc: 0x00000000
numicro chip_erase complete
Info : IDCODE: 0x00DA5332
Info : IDCODE: 0x00000100
Info : Device Name: MS51PC0AE
Info : Nuvoton NuMicro: Flash Write ...
wrote 6003 bytes from file
NuMicro.cpu: target state: halted
target halted due to debug-request, pc: 0x00000000
```

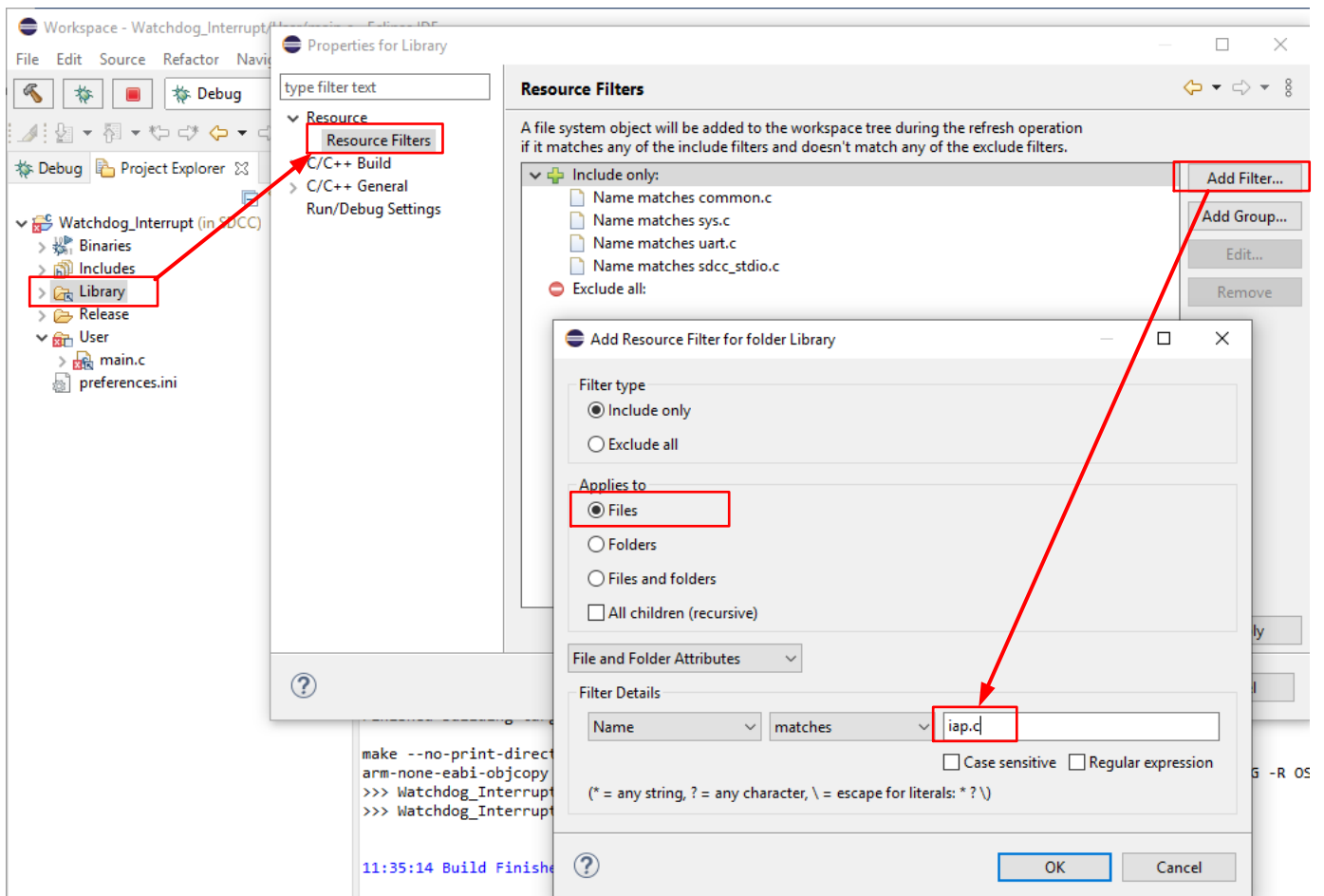
Following bottom means the entry debug mode.



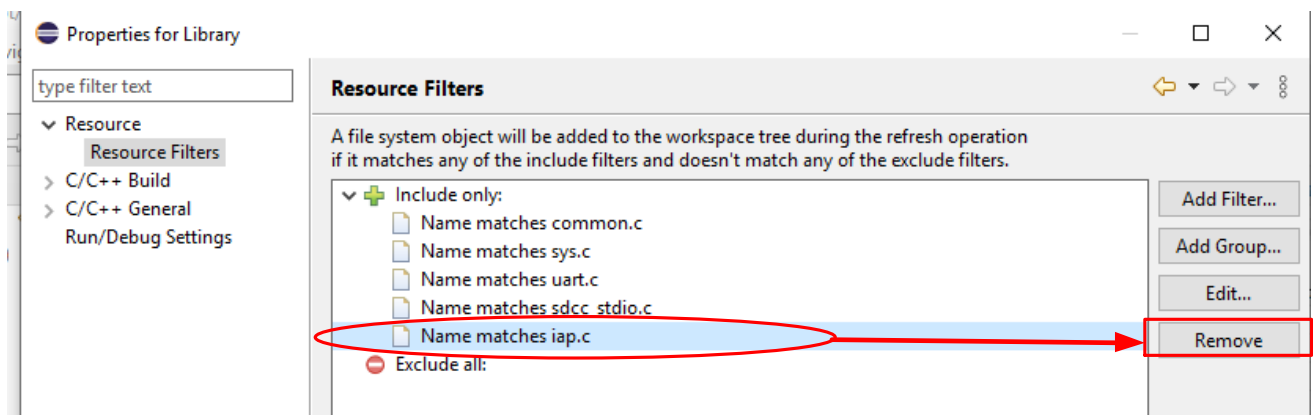
### 1.1.5 Add or Remove Library .c file

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

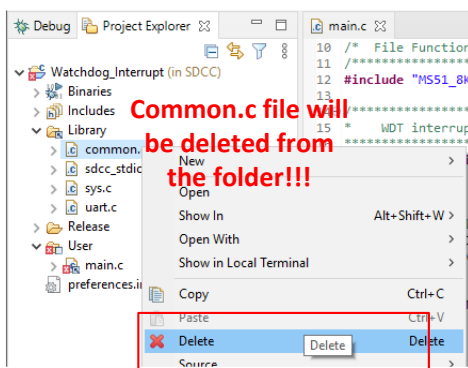




Remove lib, is Remove Filter



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



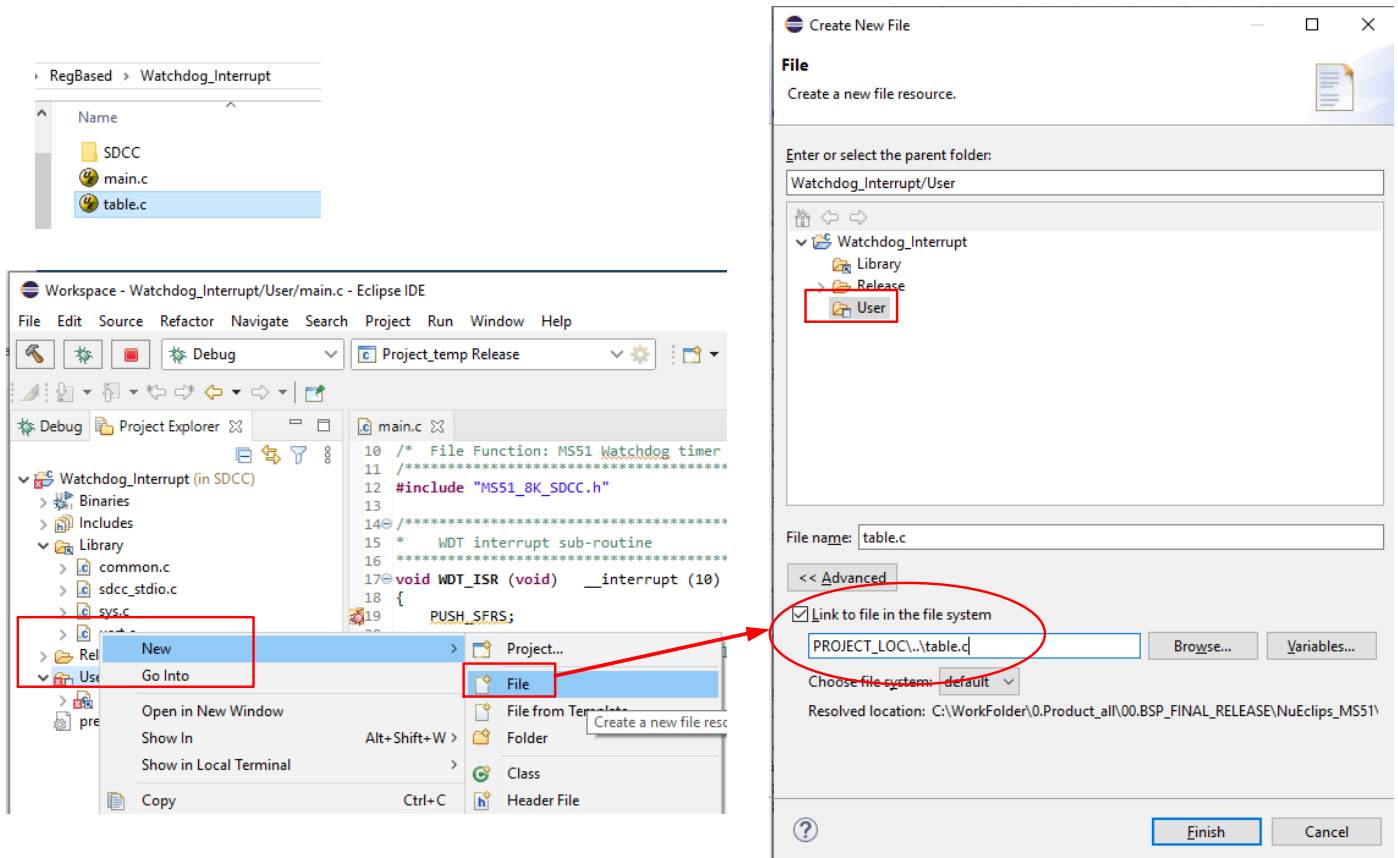
Common.c file will be deleted from the folder!!!

## 1.1.6 Add User Code

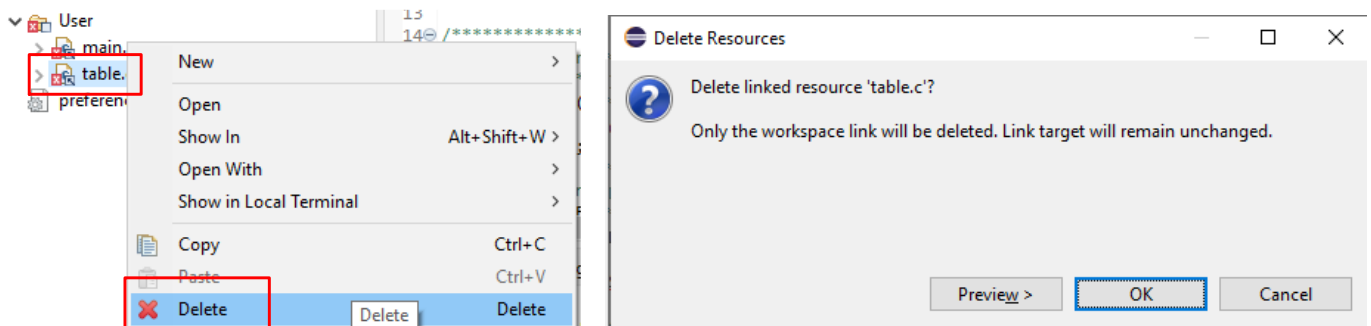
Add a new table.c, right-click user / New / File, and click Link to file in the file system on the Create New File page. It is

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>**

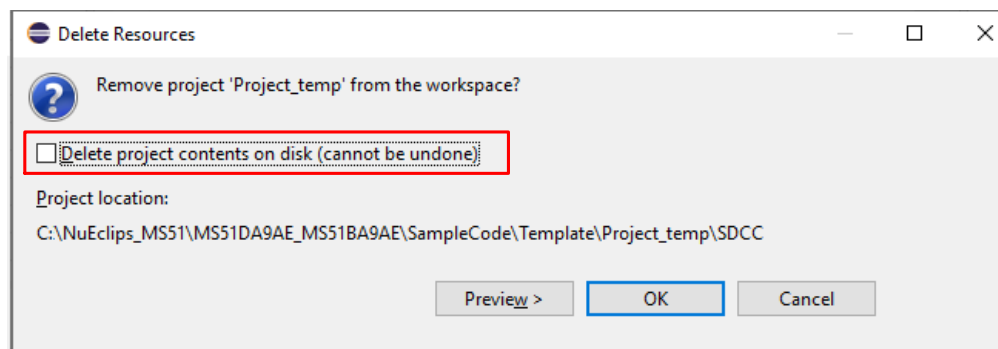


Delete user .c file



## 1.1.7 Close Project

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



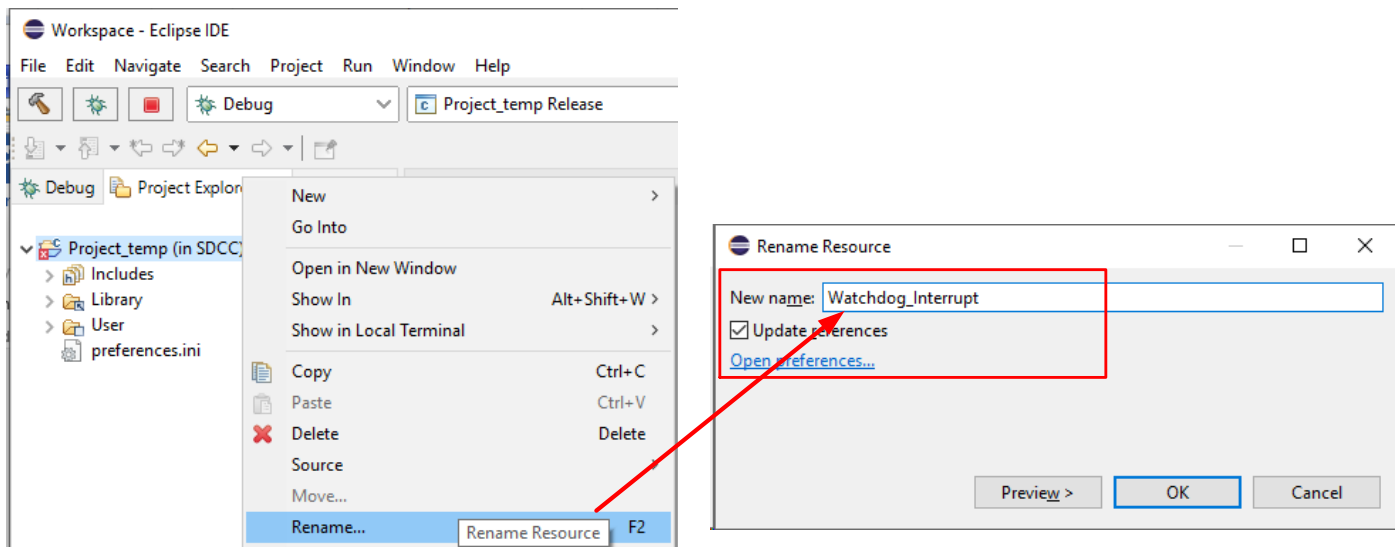
## 1.2 1.2 Create a new Project from an existing BSP project

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

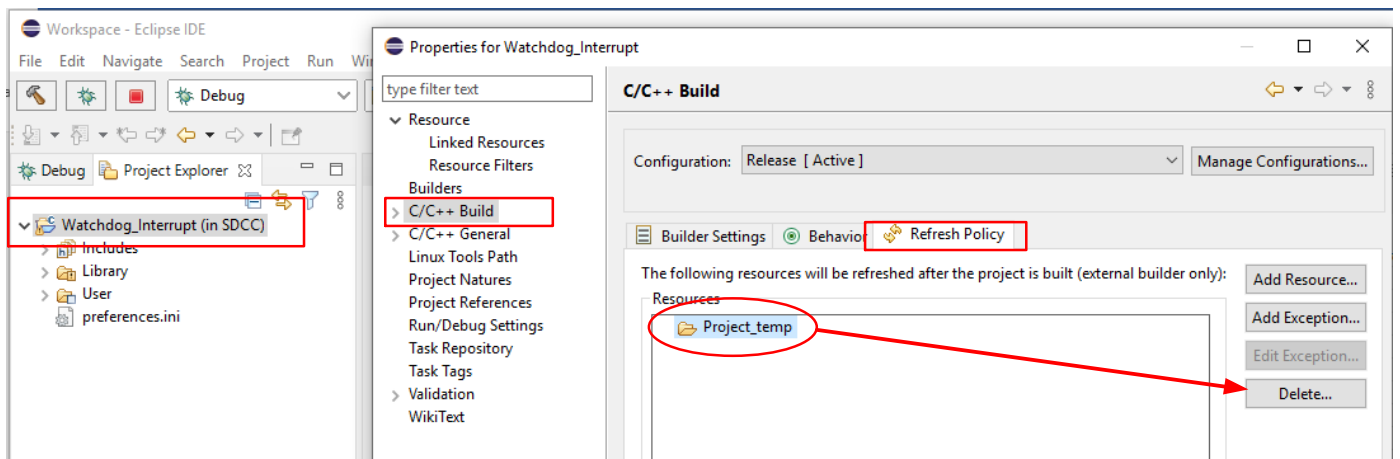
NuEclips_MS51 > MS51DA9AE_MS51BA9AE > SampleCode > RegBased > Watchdog_Interrupt			
Name	Date modified	Type	Size
SDCC	2023/2/1 下午 06:04	File folder	
main.c	2023/2/1 下午 06:06	C File	3 KB

### 1.2.2 1.2.2 Import this project in Workspace

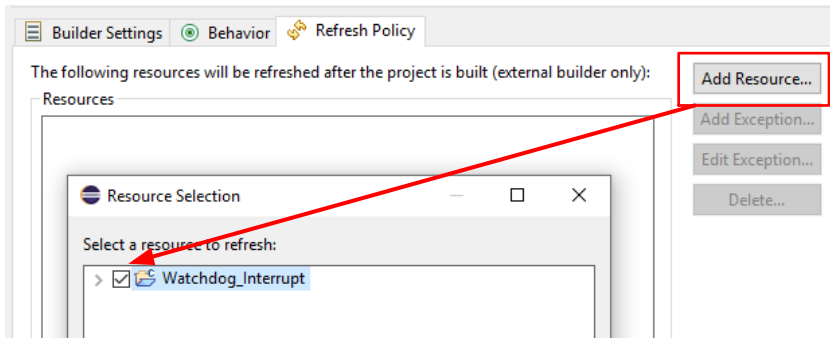
### 1.2.3 Rename Project



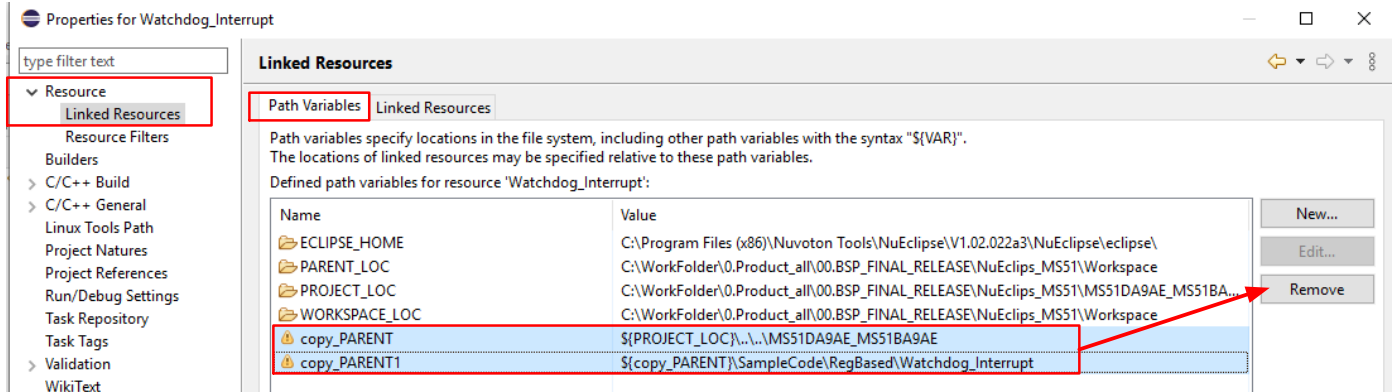
After confirming that the Project name has been changed, right-click Properties, modify Refresh Policy, and delete the old Resource



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



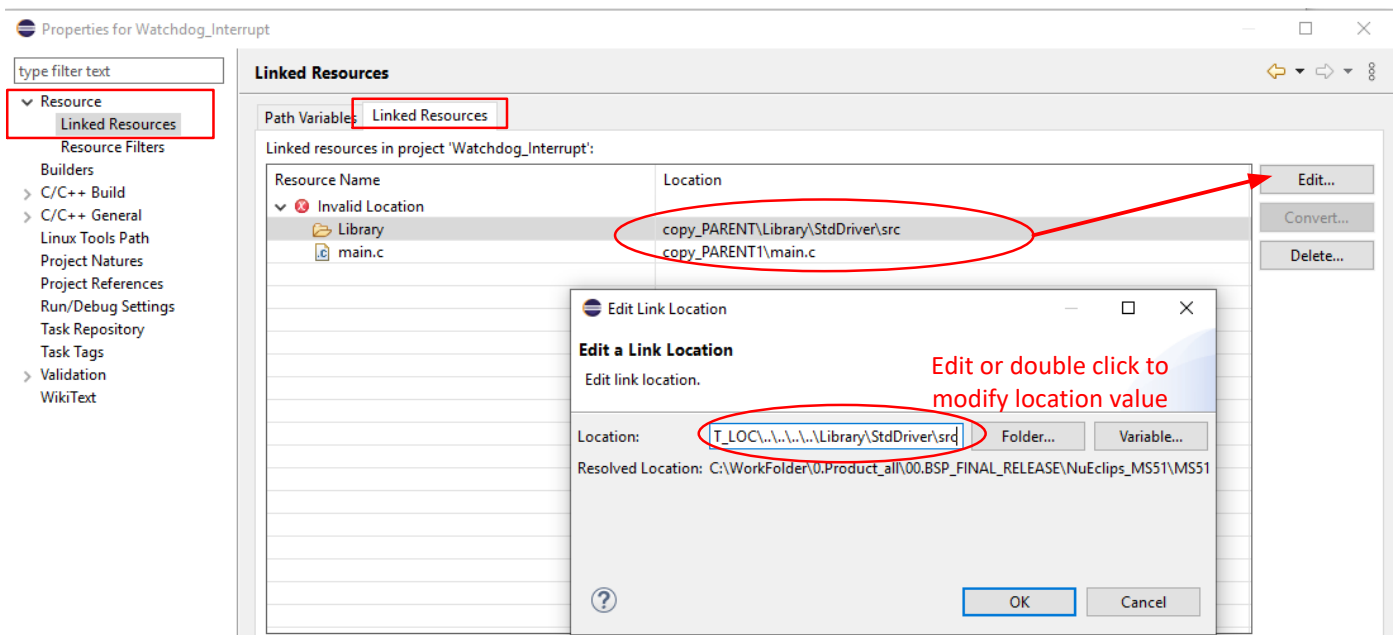
Remove the error Resource caused after Rename by eclipse system.



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

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

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

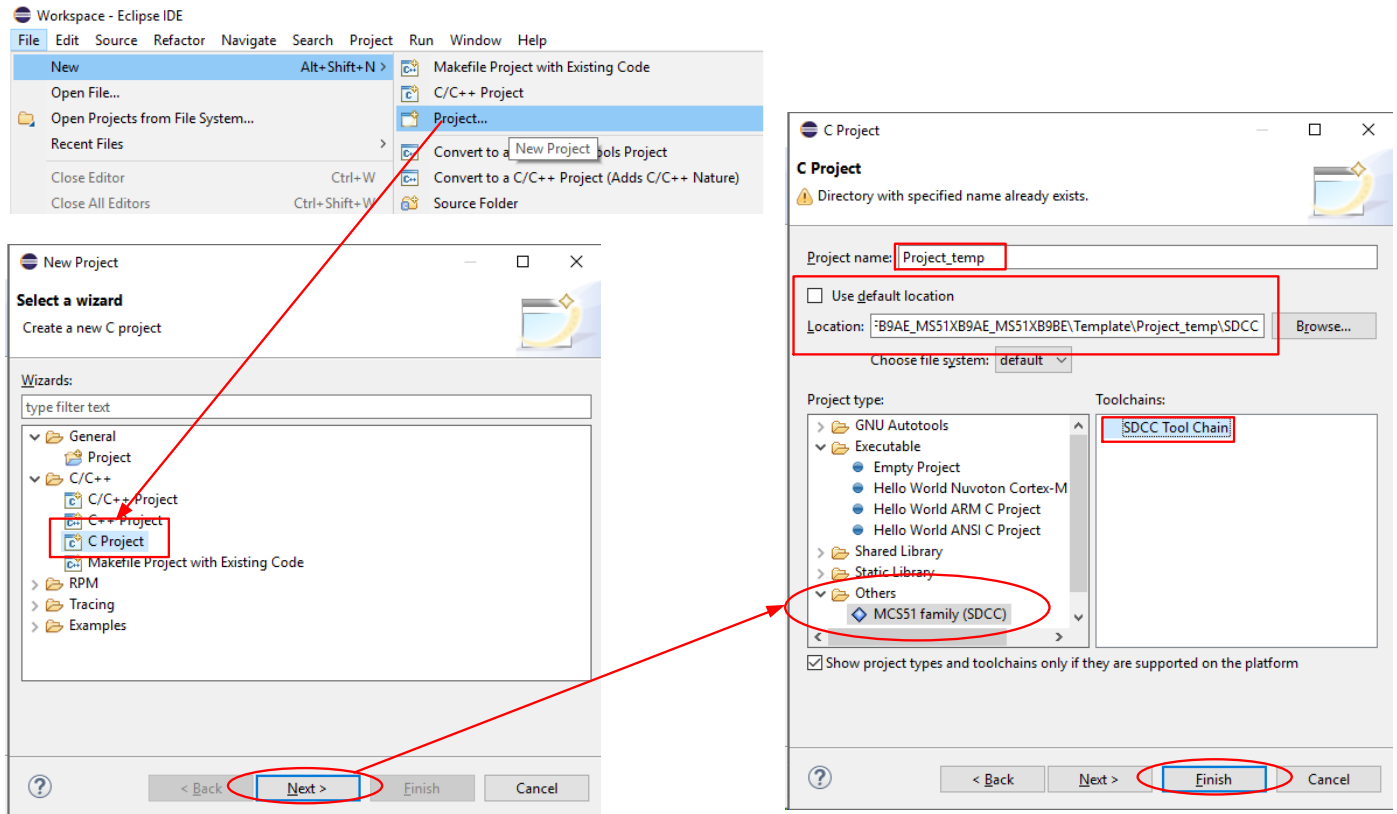


Build the Project after all modify same as section 1.1.3 and Debug project same a section 1.1.4

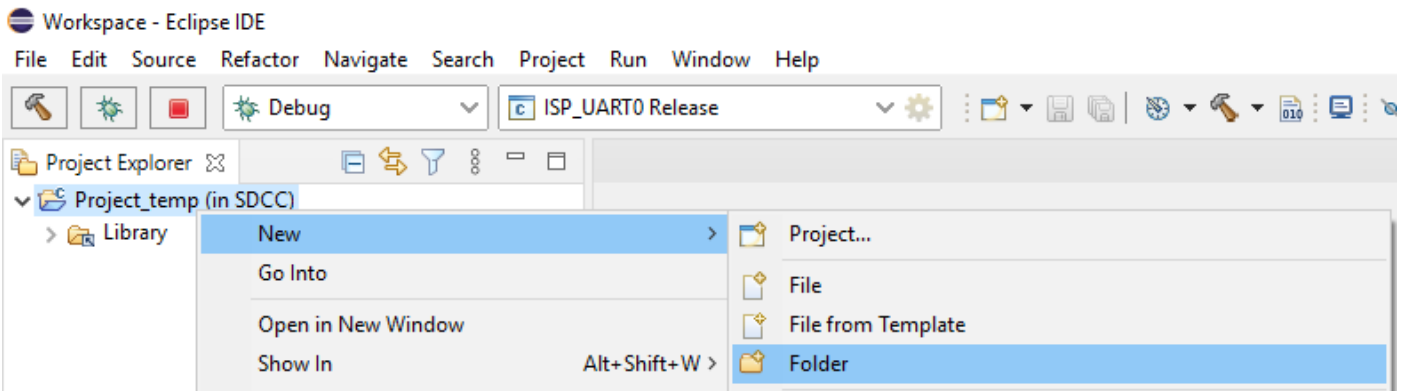
## 1.3 Create A New Project (Not Recommended)

### 1.3.1 New Project Define

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

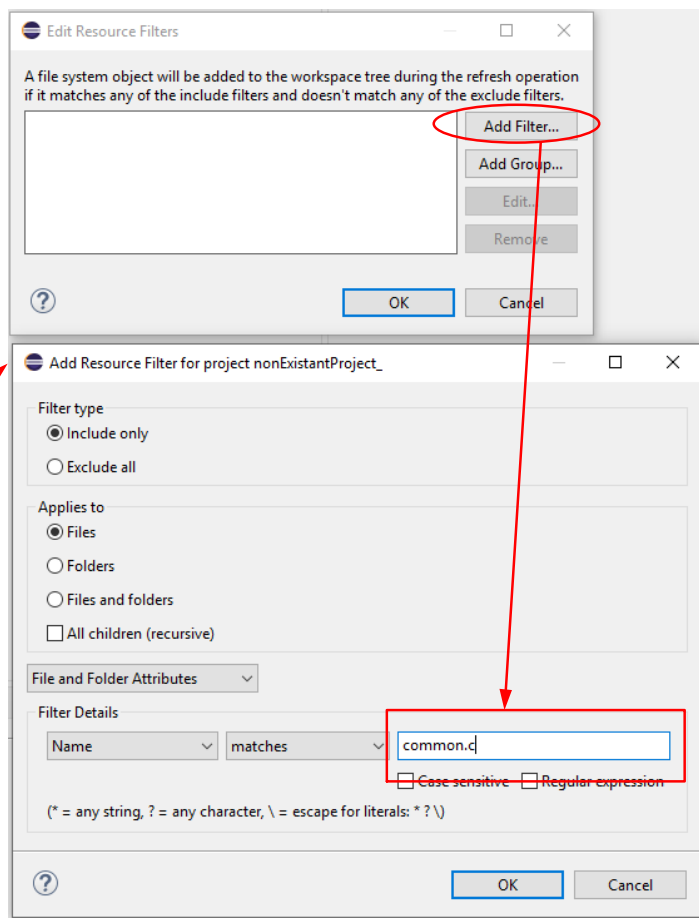
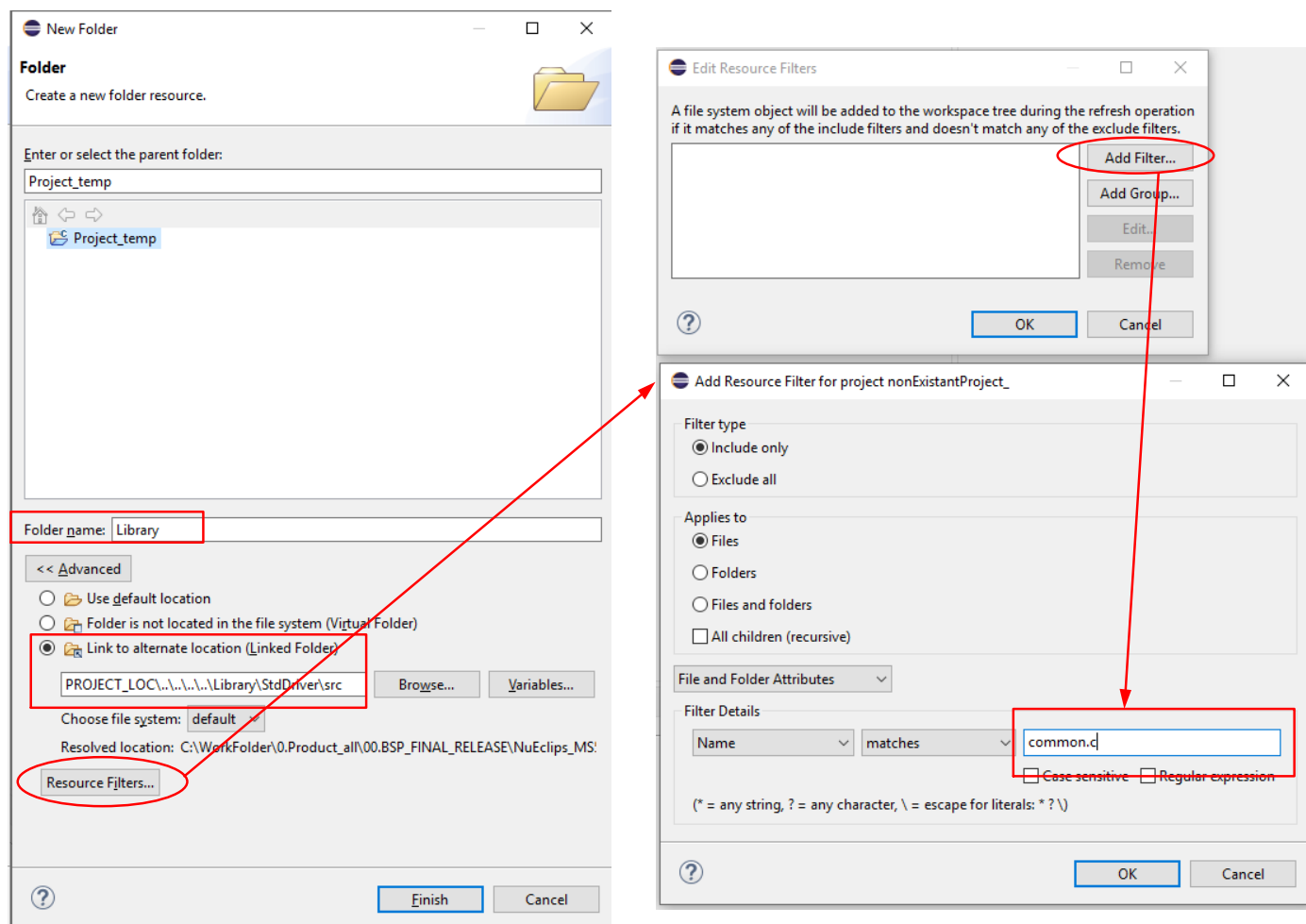


### 1.3.2 New Create Library And User Source Code

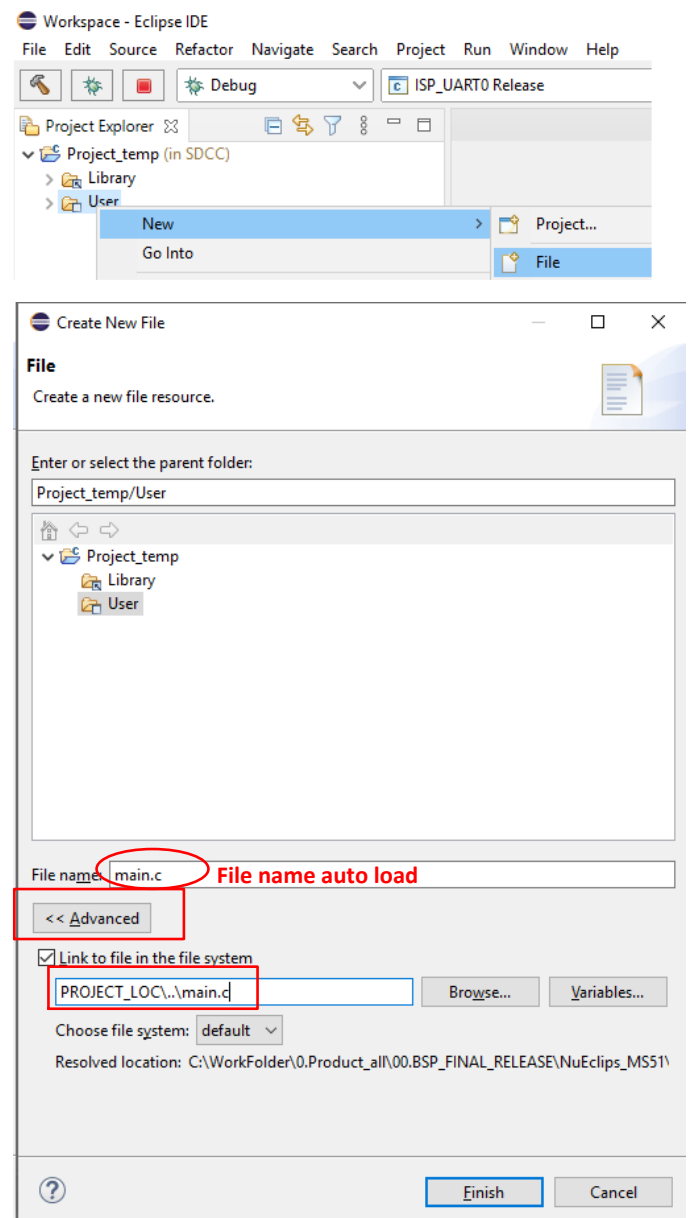
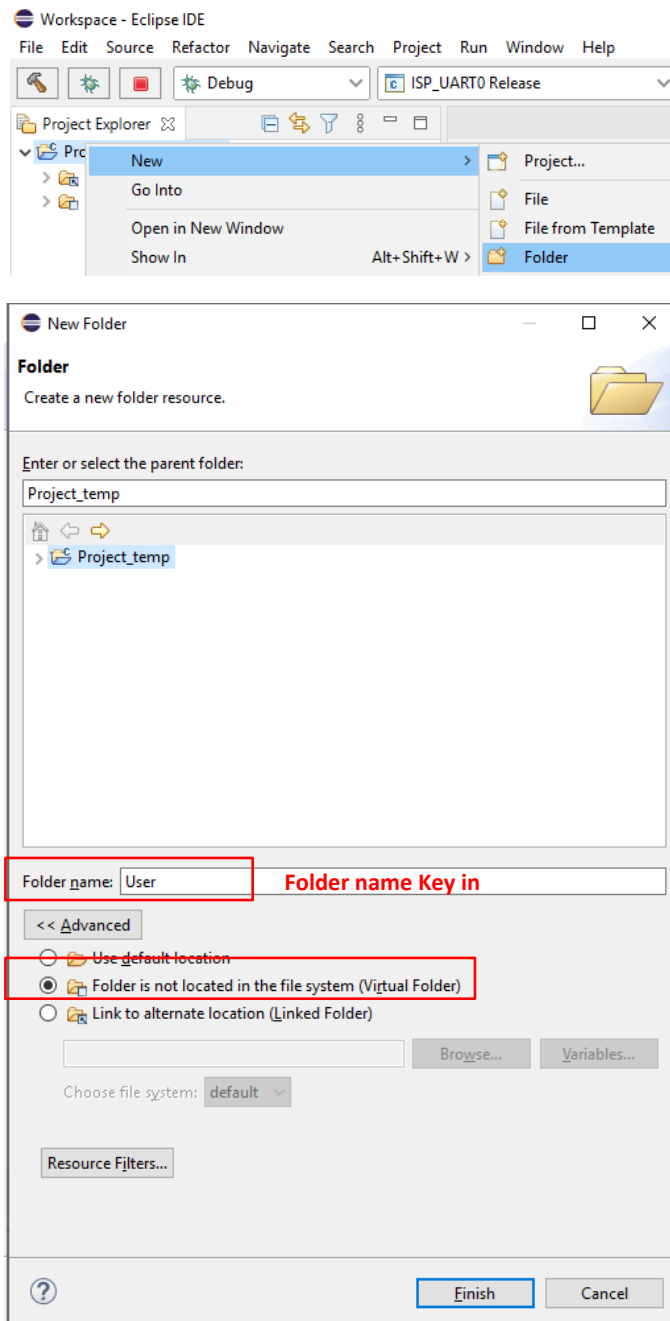


Link folder method. You can directly link the entire existing folder path, and automatically add all files in the folder without adding a filter.

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

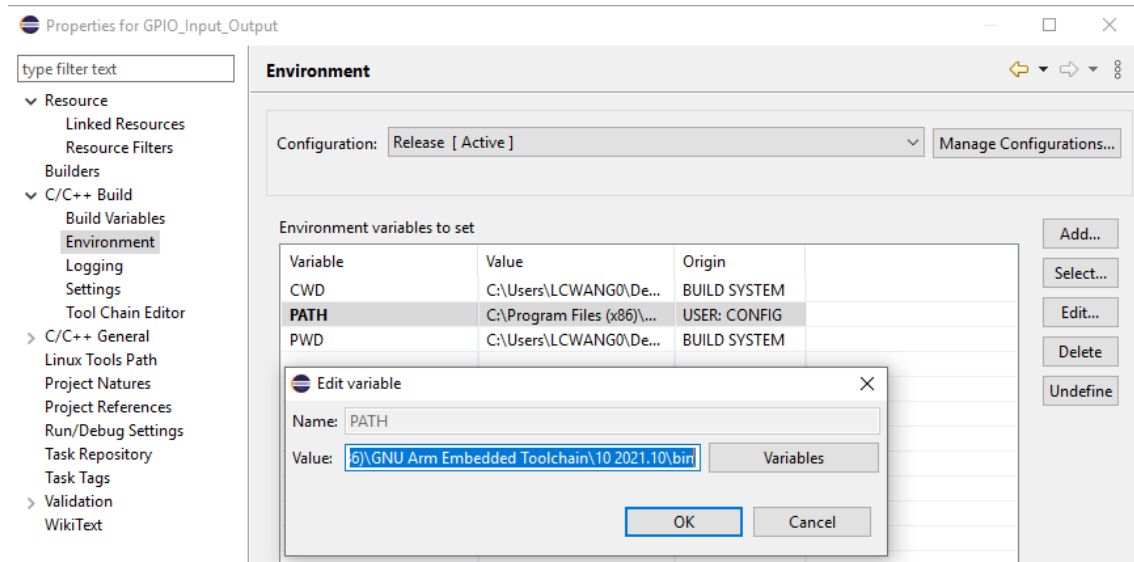


## 2 PROJECT PROPERTIES

### 2.1 Setting Project Properties

#### 2.1.1 Build / Environment / PATH

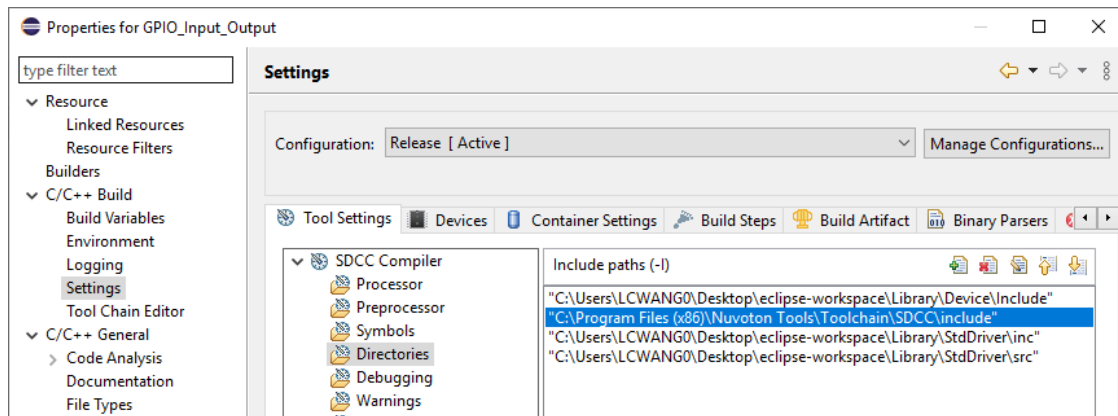
C:\Program Files (x86)\Nuvoton Tools\Toolchain\SDCC\bin;C:\Program Files (x86)\GNU ARM Eclipse\Build Tools\2.8-201611221915\bin;C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2021.10\bin



#### 2.1.2 Build setting

##### To added the include folder link

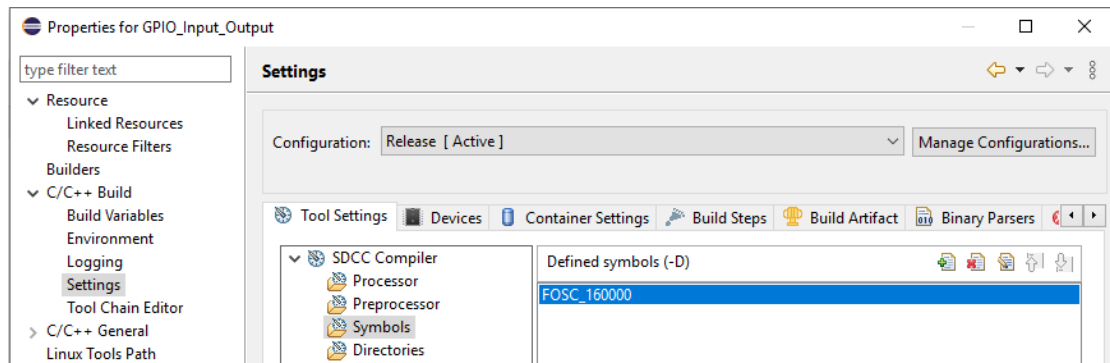
Settings / Tool Settings / SDCC Compiler / Directories Special added C:\Program Files (x86)\Nuvoton Tools\Toolchain\SDCC\include



##### To added the Project Define

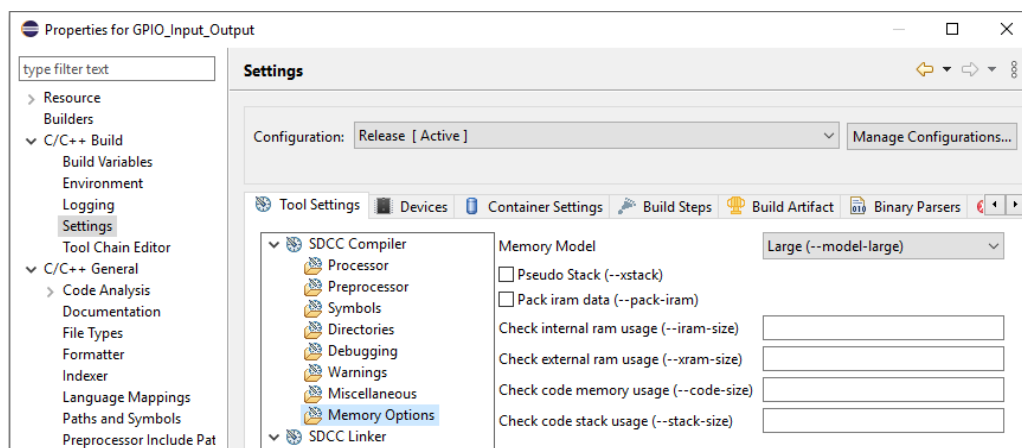
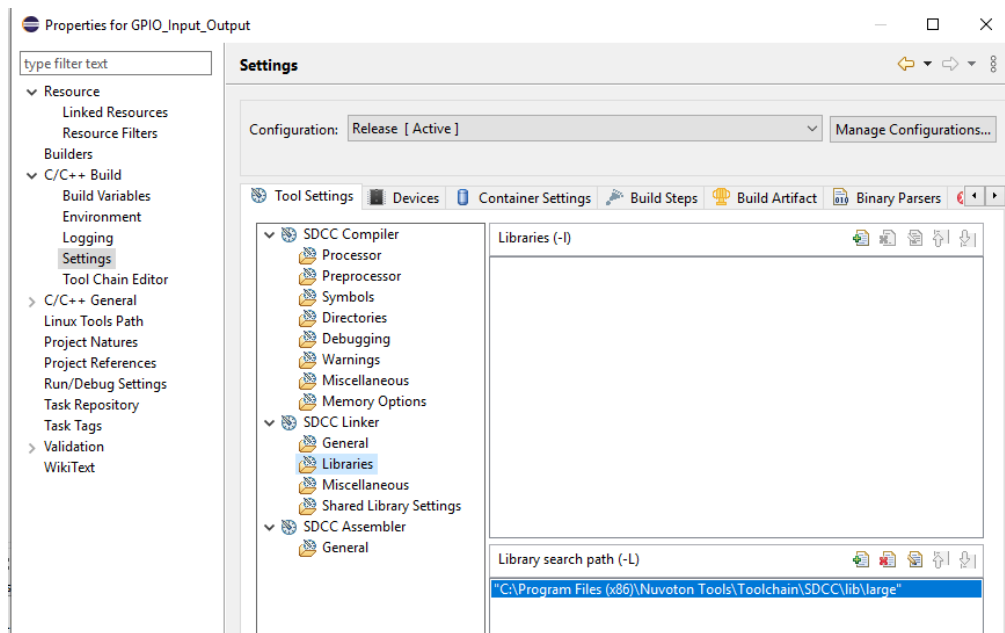
Directories/ Tool Settings / SDCC Compiler / Symbols 寫Define





### To setting default memory module

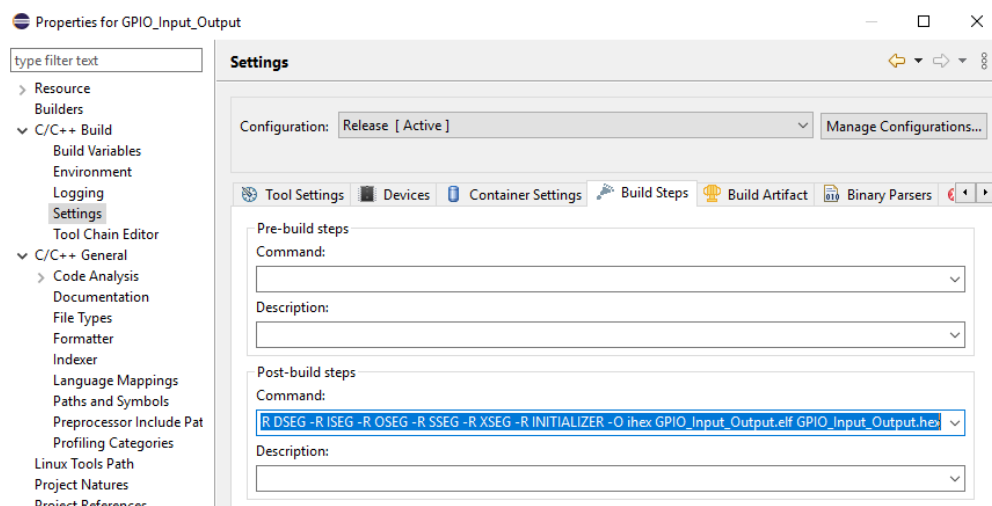
“Directories/ Tool Settings / SDCC Linker / Library” base on “Directories/ Tool Settings / SDCC Compiler / Memory Options” select SMALL or LARGE



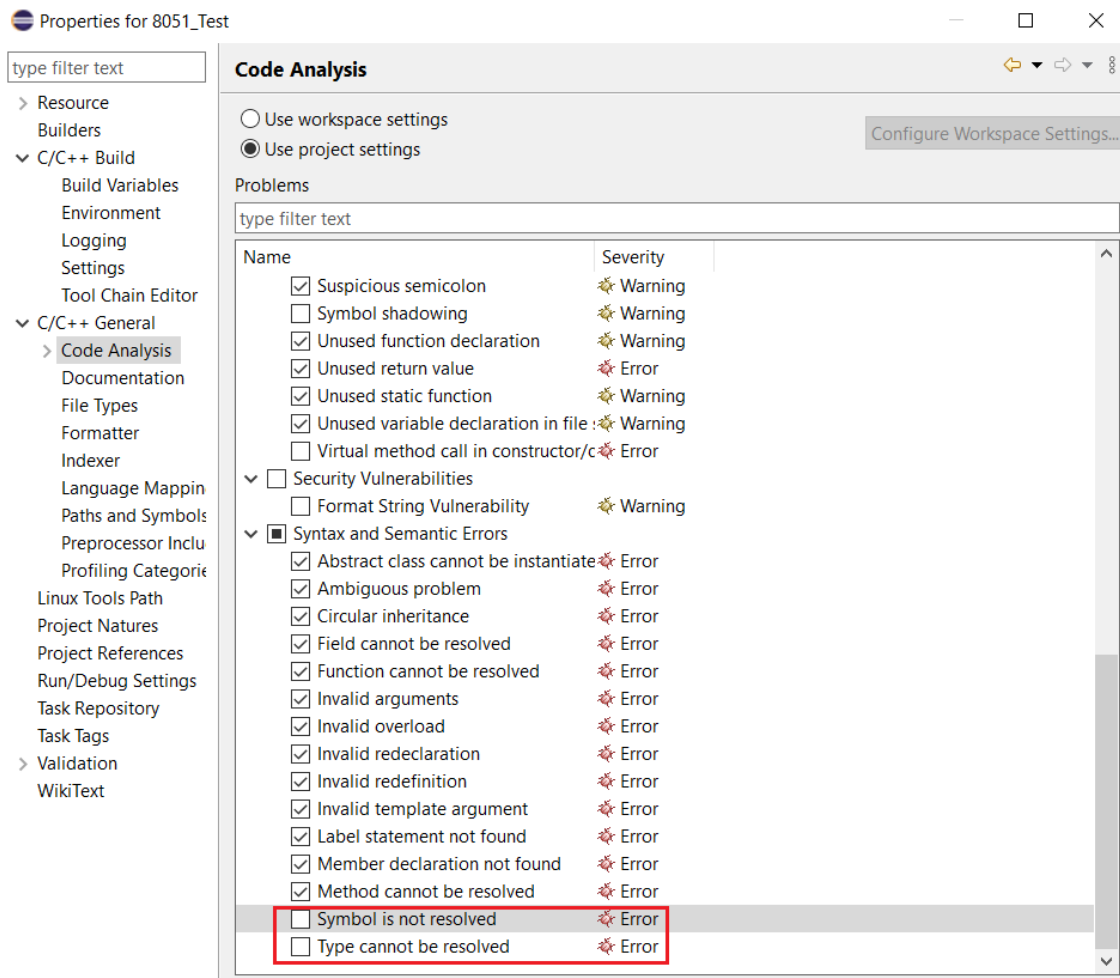
### Setting Post-build steps

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

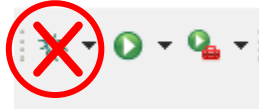


## 3 DEBUG ENVIRONMENT

### 3.1 Entry Debug Mode

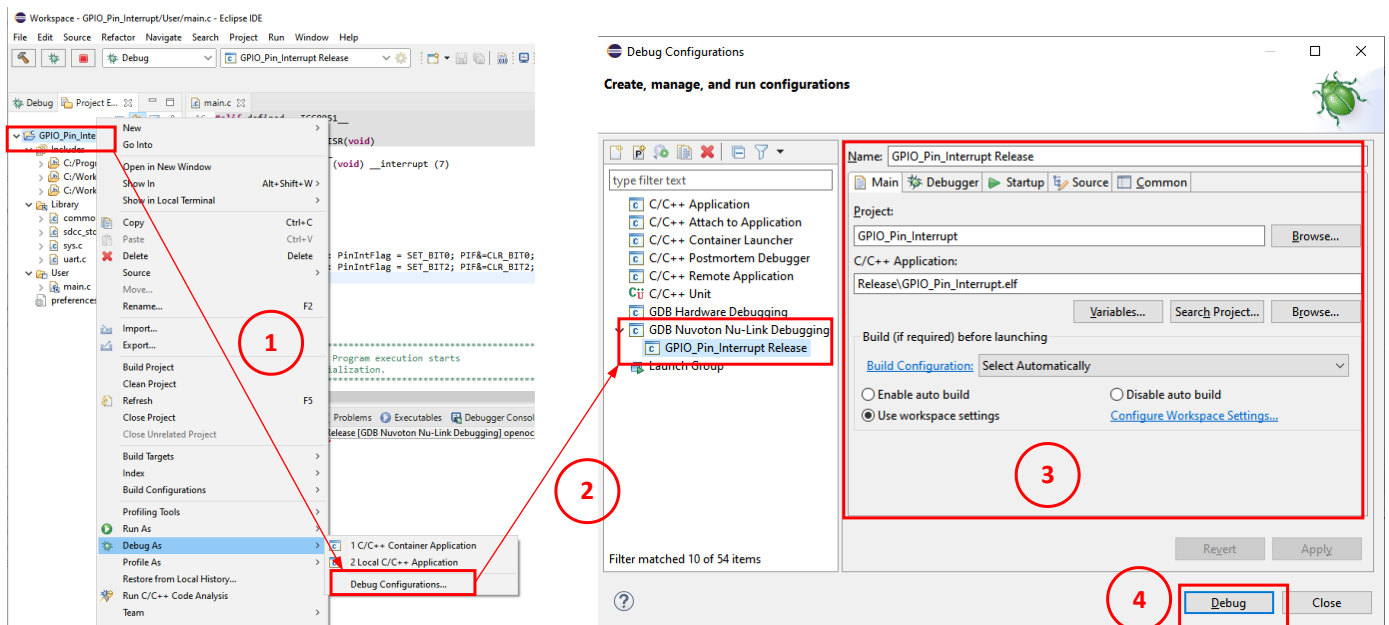
#### 3.1.1 Always using GDB Nuvoton Nu-link Debugging options window into debug mode

We suggest always using “Debug Configurations > GDB Nuvoton Nu-link Debugging” options to enable debug mode. Please do not use the debug button at first time !



#### Entry debug mode steps:

1. Right-click on the project name to bring up the following menu.
2. Select “Debug As” into “Debug Configurations” window.
3. Double click “GDB Nuvoton Nu-link Debugging” item to show “Project\_name Release” item
4. After check all debug option is correct, press “Debug” button to start debug mode.

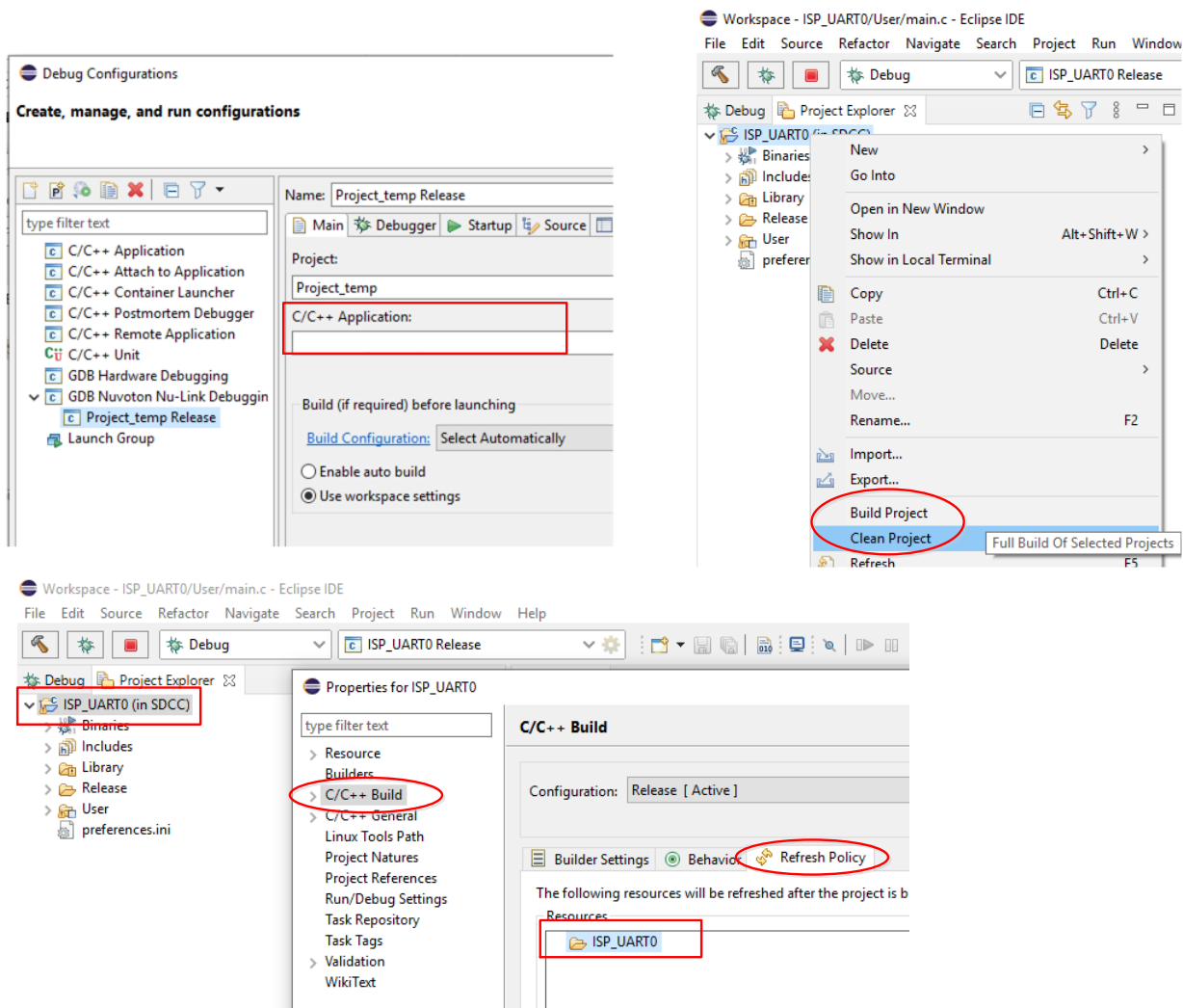


## 3.2 Check Item before into Debugging

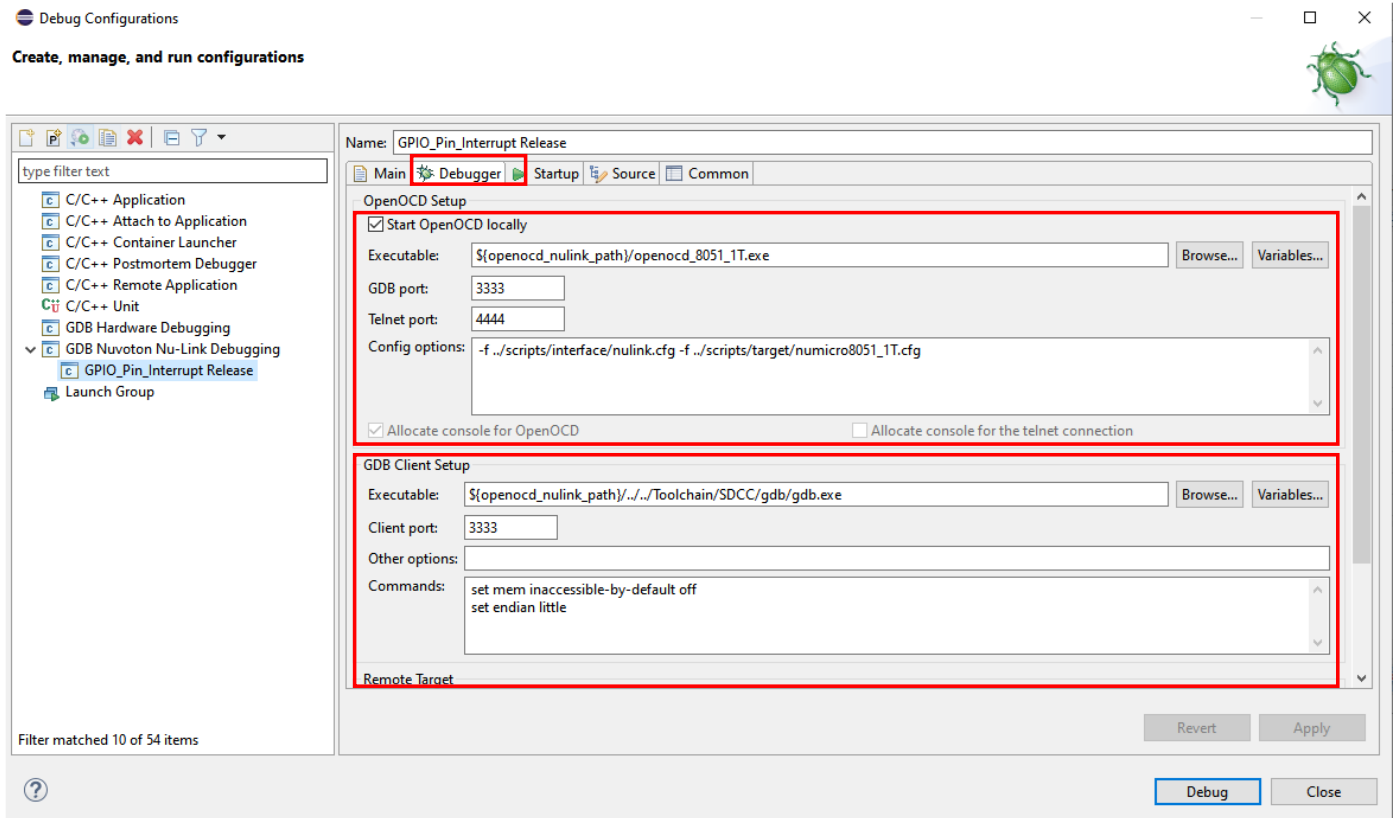
### 3.2.1.1 The Project.elf file

Ensure that, each time before entering debugging mode, you verify the "C/C++ Application" diagram with the corresponding ".elf" file. If without this file following steps is necessary.

- Delete "Project\_temp Release" label. Double click "GDB Nuvoton Nu-Link Debugging" to create a new one.
- If the "Project\_name.elf" still not showing. Begin by cleaning the project thoroughly, and then proceed to rebuild it until there are no errors. Reference [Chapter 4.4 Check Item Before into Debugging](#)
- Double-check that the project's refresh policy is correctly configured and that the project name matches consistently with the project settings.



### 3.2.1.2 Debugger Commands



### 3.2.1.3 Download completed

Ensure that the download is completed correctly. Following information means download fail, check item include:

- Only one Nu-Link connection is allowed when into debug.
- No other ICP or debug tool environment is open and trying connecting.

```
Info : IDCODE: 0x00FF0000
Warn : NuMicro flash driver: Failed to detect a known part
undefined debug reason 7 - target needs reset
NuMicro.cpu: target state: halted
target halted due to debug-request, pc: 0x000fffff
numicro chip_erase complete
NuMicro.cpu: target state: halted
target halted due to debug-request, pc: 0x000fffff
```

## 3.3 Variables Display

### 3.3.1 Set The Location Of Different Regions To Display

**Notice:**

With version **V1.02.028 and Later** the "Data Access Offset Definition" has changed. Please confirm your NuEclipse version first.

**Version V1.02.028 and later:**

memory space of SDCC declaration	Data access offset define
__code	(CODE address) + 0
__SFR page 0	(SFRs address) + 0x100000
__SFR page 1	(SFRs address) + 0x100100
__SFR page 2	(SFRs address) + 0x100200
__SFR page 3	(SFRs address) + 0x100300
__data	(DATA address) + 0x100000
__idata	(IDATA address) + 0x210000
__xdata	(XDATA address) + 0x310000

**Earlier Versions:**

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

### 3.3.2 Use The Expression Window To View Variables

#### Notice:

With version **V1.02.028 and Later** all variable need to display value in “Expression window” must defined as **global** variable.

#### 3.3.2.1 Expression

List the value Expression: **\*(&<Variable Name / Address> + Data access offset define)**

List the address Expression: **&<Variable Name>**

#### Version V1.02.028 and later:

The screenshot shows the IDE with source code on the left and the Expression window on the right. The source code includes headers and defines variables for Bandgap\_Voltage, VDD\_Voltage, and ADC\_BG\_Result. The Expression window displays a list of expressions and their corresponding types and values.

Expression	Type	Value
*(&TCON+0x100000)	volatile unsigned char	0xc0 (Hex)
*(&SFRS+0x100000)	volatile unsigned char	0x0 (Hex)
*(&ADC_BG_Result+0x310000)	unsigned int	0x67a (Hex)
*(&Bandgap_Voltage+0x310000)	float	1251
*(&ADCCONT1+0x100000)	volatile unsigned char	0x30 (Hex)
*(&ADCCON0+0x100000)	volatile unsigned char	0x88 (Hex)
&ADC_BG_Result	unsigned int *	9

#### Earlier Versions:

The screenshot shows the IDE with the Expression window. The window displays a list of expressions and their corresponding types and values.

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

#### 3.3.2.2 “Memory” window to check

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

#### Version V1.02.028 and later:

The screenshot shows the IDE with the Memory window. The window displays a table of memory addresses and their corresponding values.

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00310000	06	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00310010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00310020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00310030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00310040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00310050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

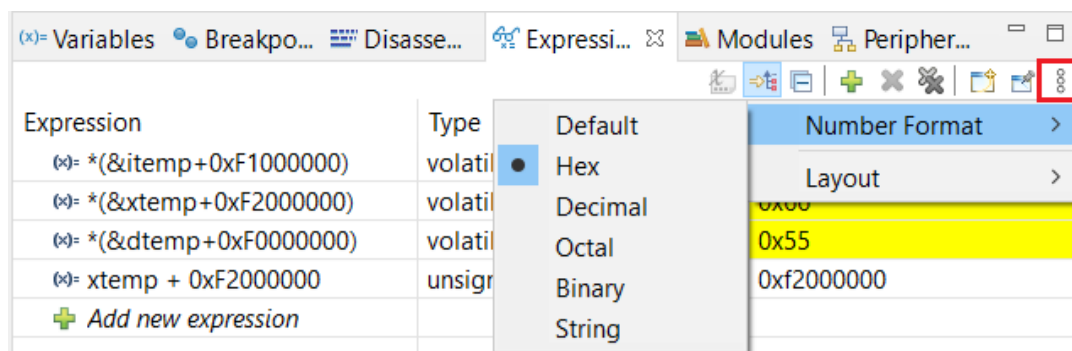
#### Earlier Versions:

The screenshot shows the IDE with the Memory window and a dialog box. The dialog box is titled "Monitor Memory" and contains a text field with the address "0x00 + 0xF2000000".

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

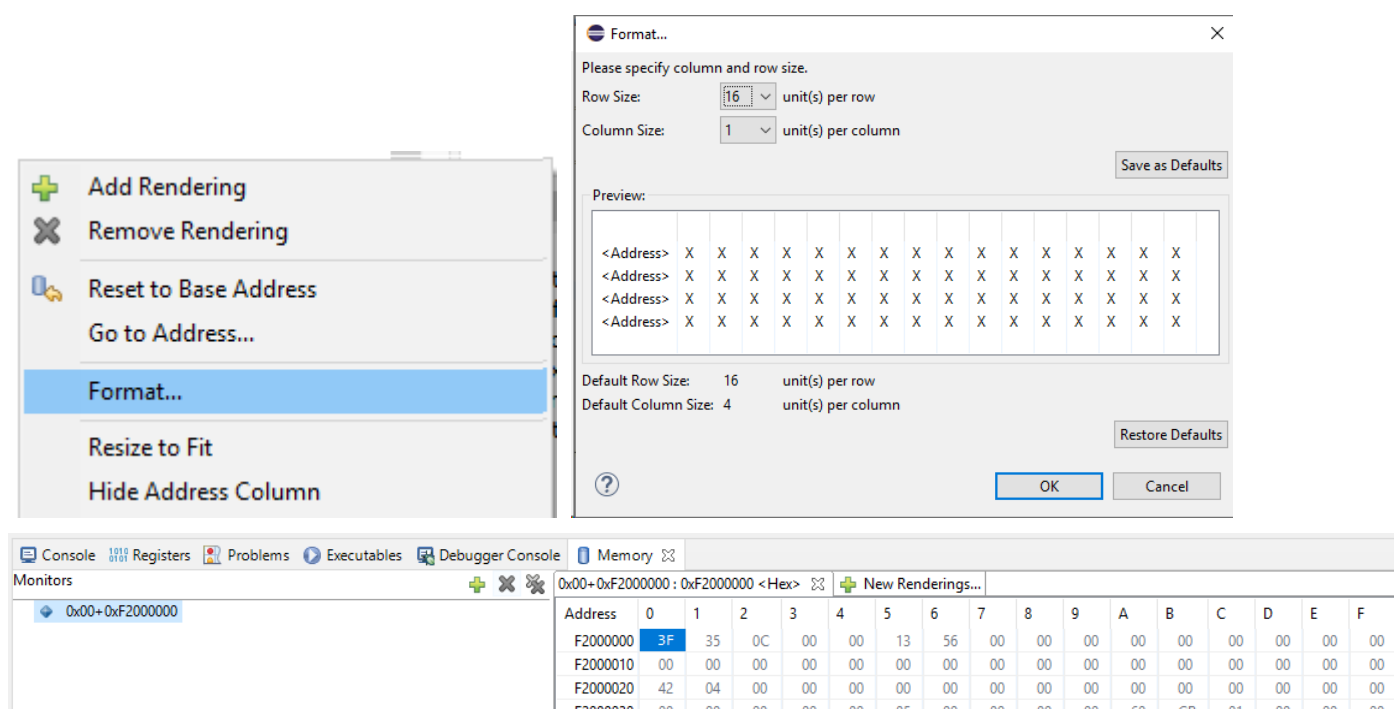
### 3.3.2.3 Display value format

Select different format as follows.



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

Right click on the Memory window area



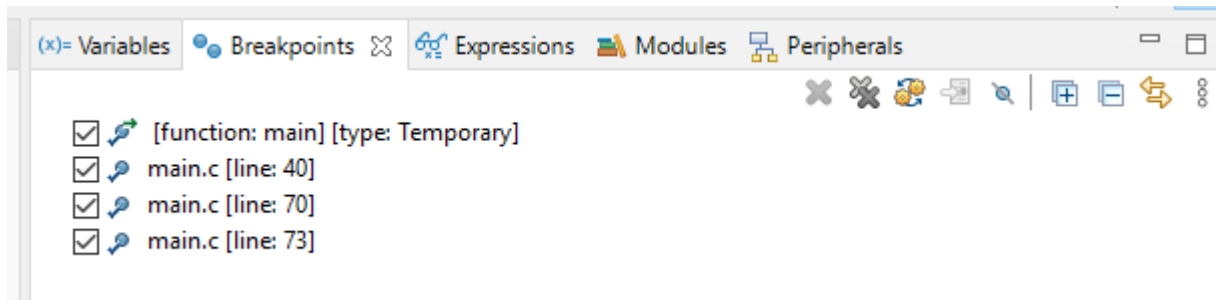


### 3.4 Set Break Point

```
38     Bandgap_Value = (BandgapHigh<<4)+BandgapLow;  
39     Bandgap_Voltage= Bandgap_Value*3/4;  
40     clr_CHPCON_IAPEN;  
41 }  
42
```

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 & FAQ

Each item is crucial to preventing project build errors! Especially since some errors may not be readily apparent during the build or when in debug mode!!

### 4.1 File Name

File extensions should be in lowercase. We recommend using all lowercase characters for the project's file names.

### 4.2 Coding Rule

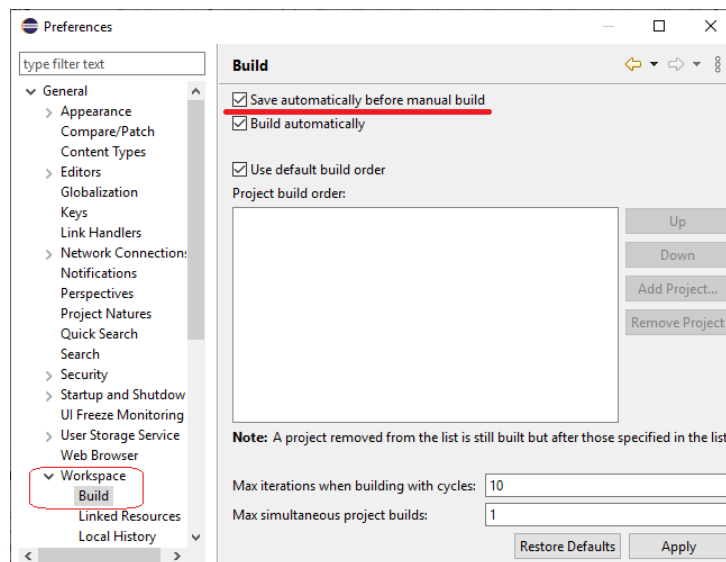
- 1) It is recommended to place all Interrupt Vectors in the main.c file.

### 4.3 Save All

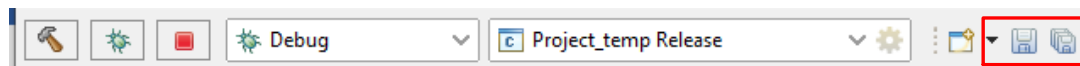
After modifying the code, make sure to "save all" code before initiating the build process.

- 1) By enabling the "Save automatically before manual build" setting in NuEclipse, your modified code will be automatically saved before building.

You can find this option in the following path: Window > Preferences > General > Workspace > Build



- 2) If necessary, save manually.



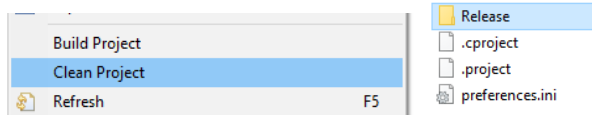
## 4.4 Check Item Before into Debugging

For instance, after importing the project, you may encounter various scenarios: no response when clicking "Project->Build Project," multiple clicks required for a response, or initial error reports that disappear upon clicking "Build Project" again.

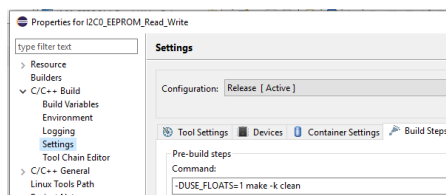
Please review the items and their respective solutions:

1) If you encounter incomplete information or unresponsiveness during any build process,

- Clicking "Clean Project" to reset the project records.
- Delet "Release" folder.
- Build Project again.



2) To ensure that the Clean Project step is automatically executed before every re-build, include the "make -k clean" command in the Pre-build steps.



3) Rebuild the project until you achieve a status of "0 errors" and "0 warnings."

Additionally, ensure that the "sdcc – debug out" and all corresponding ".c" files are successfully generating the corresponding ".rel" files

```
Building target: ADC_Bandgap_VDD.elf
Invoking: SDCC Linker
sdcc --debug --out-fmt-elf -L"C:\Program Files (x86)\Nuvoton 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 11s.588ms)
```

When you encounter build information such as "Nothing to be done," it is necessary to perform a clean operation followed by a build

```
Console  Registers  Progress  Problems  Executables  Memory Browser  Debugger Console
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)
```

4) Following information all means build error:

- The API library file is currently missing. Please add relation "lib.c" to the library filter.

```
Building target: ACPM_Wakeup_ML51(64KB_32KB)_ML54_ML56.elf
Invoking: SDCC Linker
sdcc --debug --out-fmt-elf -L"C:\Program Files (x86)\Nuvoton Tools\OpenOCD/bin/../../Tool

?ASlink-Warning-Undefined Global '_Timer0_Delay' referenced by module 'main'

?ASlink-Warning-Undefined Global '_Timer0_Delay_PARM_2' referenced by module 'main'

?ASlink-Warning-Undefined Global '_Timer0_Delay_PARM_3' referenced by module 'main'
make[1]: *** [ACPM_Wakeup_ML51(64KB_32KB)_ML54_ML56.elf] Error 1
makefile:47: recipe for target 'ACPM_Wakeup_ML51(64KB_32KB)_ML54_ML56.elf' failed
make[1]: Target 'main-build' not remade because of errors.
makefile:39: recipe for target 'all' failed
make: *** [all] Error 2
"make -k all" terminated with exit code 2. Build might be incomplete.

17:01:39 Build Failed. 4 errors, 0 warnings. (took 4s.30ms)
```

