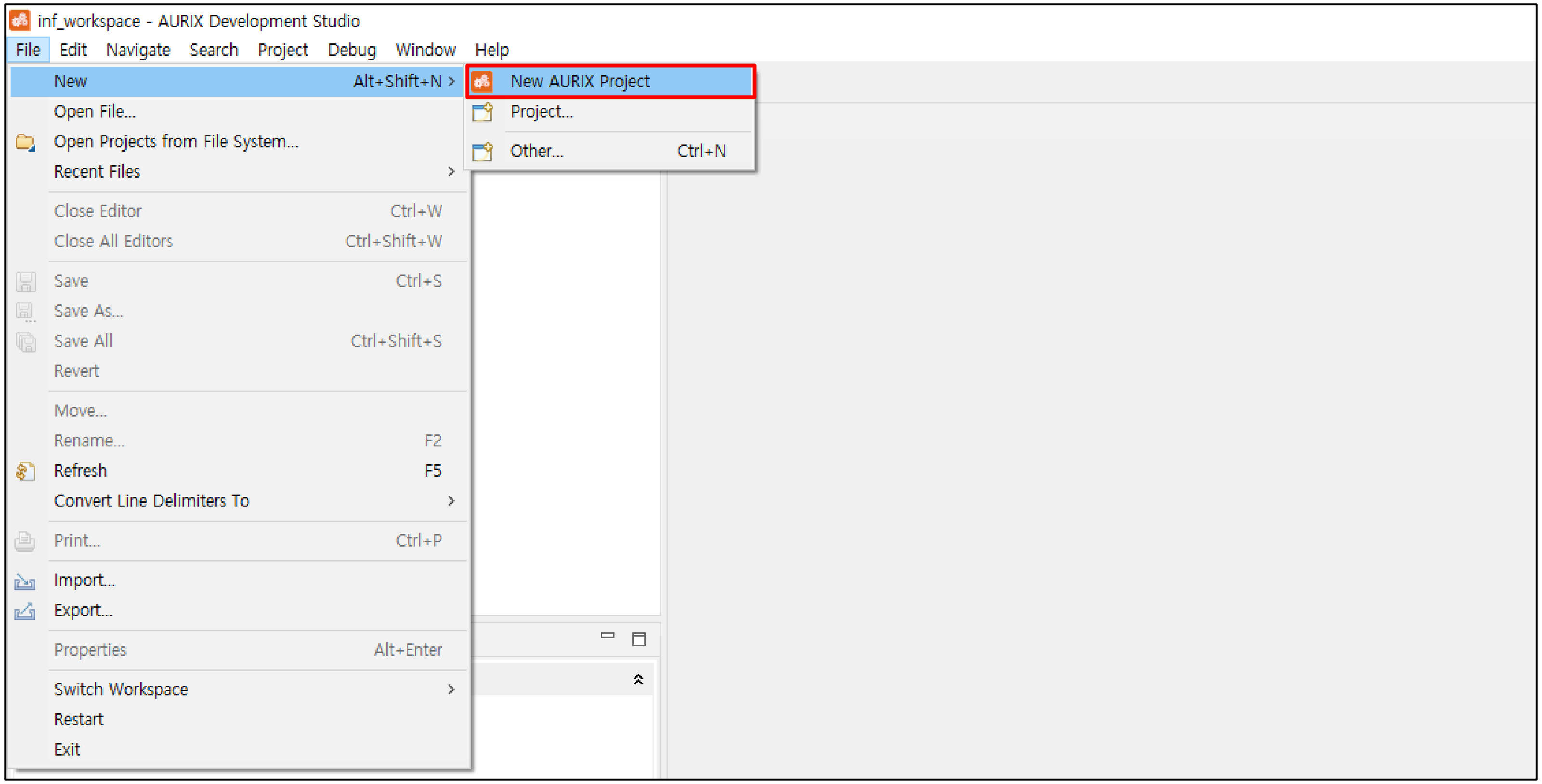
Infineon TC275 Getting Started

Architecture and Compiler for Embedded System LAB.
School of Electronics Engineering, KNU, KOREA
2021-12-23

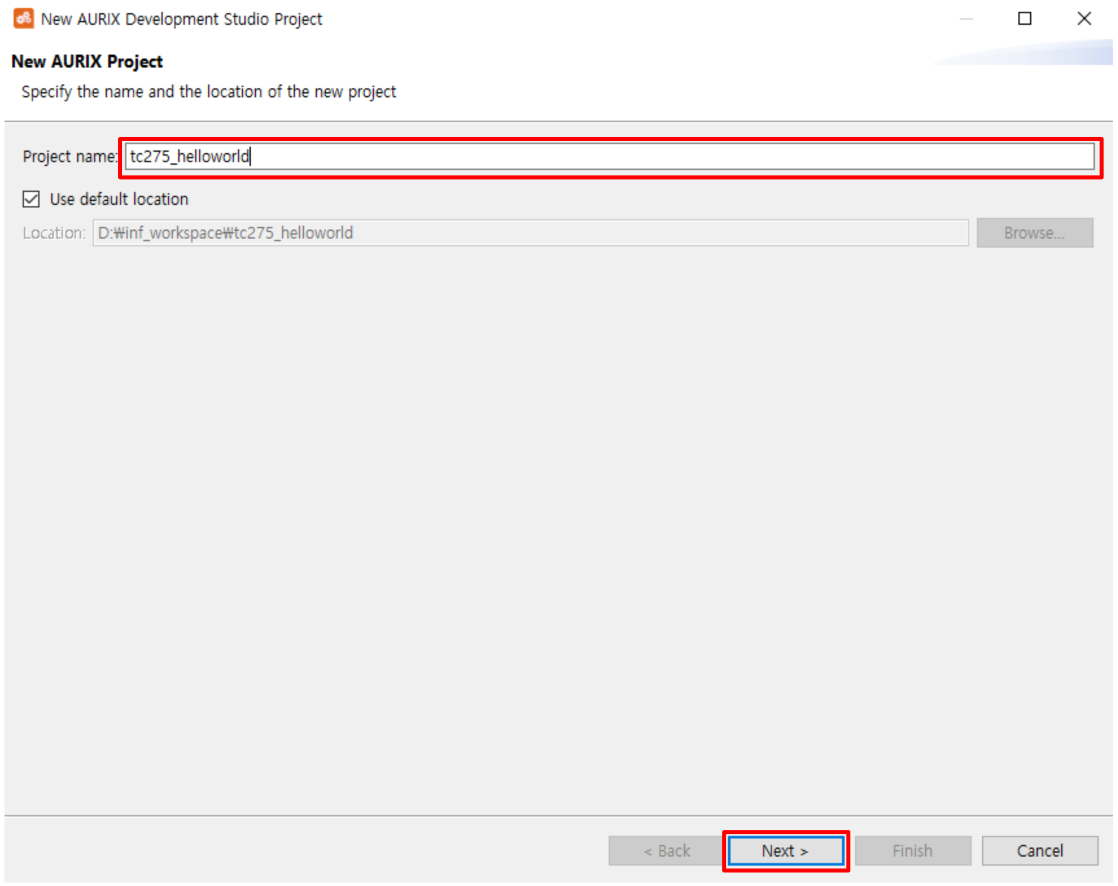
New Project

1. AURIX Development Studio를 실행하고 왼쪽 상단의 **'File – New –New AURIX Project'**를 클릭한다.



New Project

2. Project name을 입력하고, **‘Next’**를 클릭한다.


New AURIX Development Studio Project

New AURIX Project

Specify the name and the location of the new project

Project name: tc275_helloworld

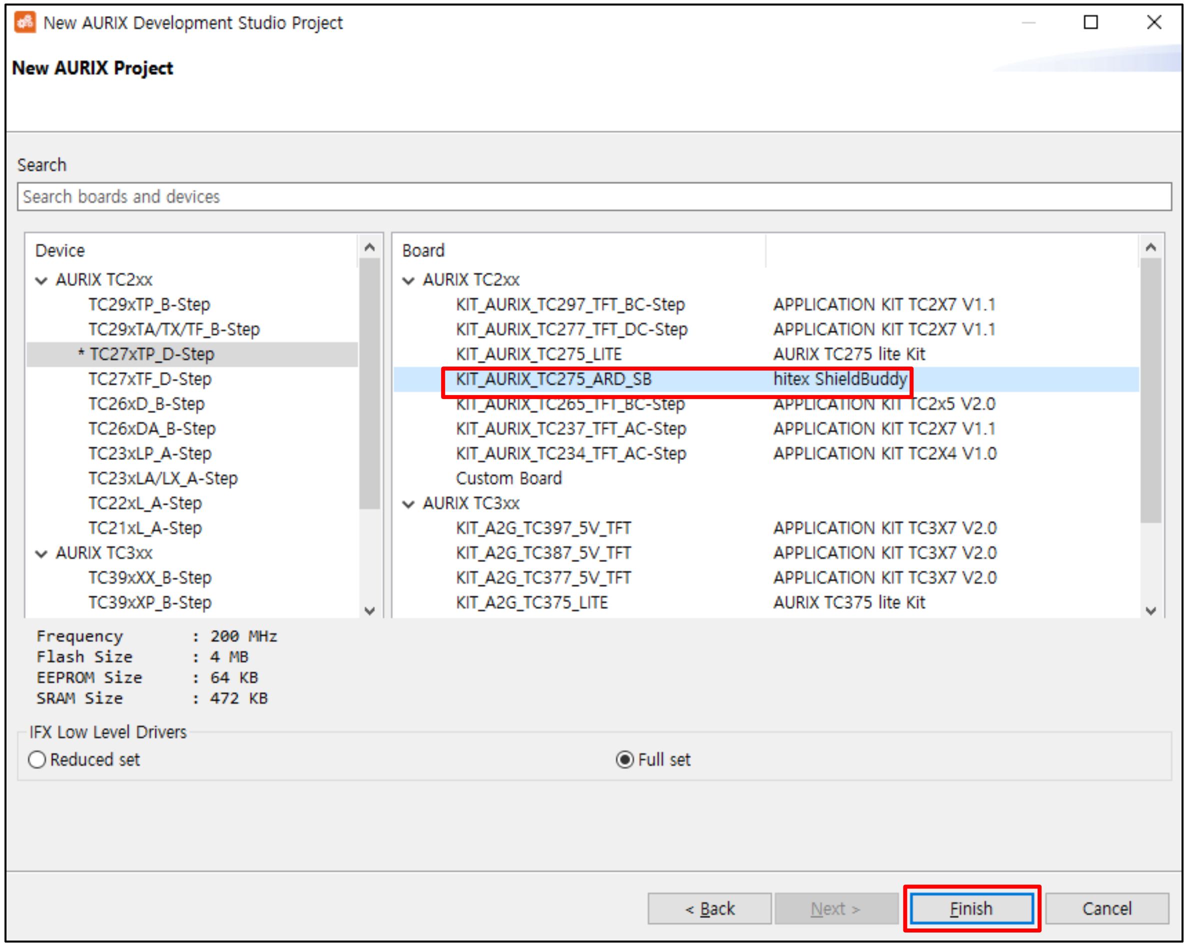
☑ Use default location

Location: D:₩inf_workspace₩tc275_helloworld

ACE Lab.

New Project

3. Device에서 **'AURIX TC2xx– TC27XTP_D-Step –KIT_AURIX_TC275_ARD_SB'**를 선택하고, 다른 설정은 그대로 유지한 채 **'Finish'**를 클릭한다.


New AURIX Development Studio Project

New AURIX Project

Search

Search boards and devices

Device
AURIX TC2xx
TC29xTP_B-Step
TC29xTA/TX/TF_B-Step
* TC27xTP_D-Step
TC27xTF_D-Step
TC26xD_B-Step
TC26xDA_B-Step
TC23xLP_A-Step
TC23xLA/LX_A-Step
TC22xL_A-Step
TC21xL_A-Step
AURIX TC3xx
TC39xXX_B-Step
TC39xXP_B-Step

Board	
AURIX TC2xx	
KIT_AURIX_TC297_TFT_BC-Step	APPLICATION KIT TC2X7 V1.1
KIT_AURIX_TC277_TFT_DC-Step	APPLICATION KIT TC2X7 V1.1
KIT_AURIX_TC275_LITE	AURIX TC275 lite Kit
KIT_AURIX_TC275_ARD_SB	hitex ShieldBuddy
KIT_AURIX_TC265_TFT_BC-Step	APPLICATION KIT TC2x5 V2.0
KIT_AURIX_TC237_TFT_AC-Step	APPLICATION KIT TC2X7 V1.1
KIT_AURIX_TC234_TFT_AC-Step	APPLICATION KIT TC2X4 V1.0
Custom Board	
AURIX TC3xx	
KIT_A2G_TC397_5V_TFT	APPLICATION KIT TC3X7 V2.0
KIT_A2G_TC387_5V_TFT	APPLICATION KIT TC3X7 V2.0
KIT_A2G_TC377_5V_TFT	APPLICATION KIT TC3X7 V2.0
KIT_A2G_TC375_LITE	AURIX TC375 lite Kit

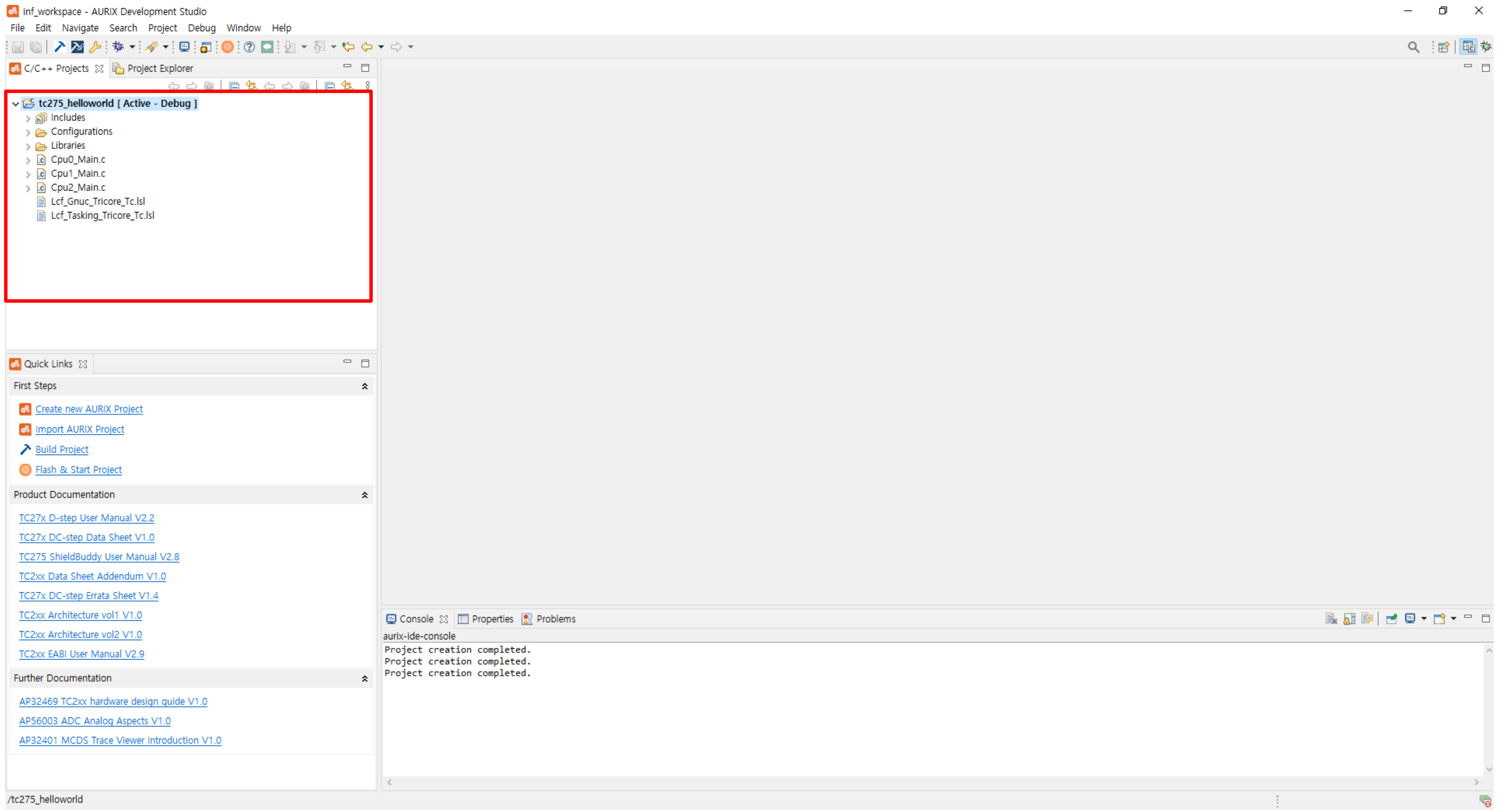
```
Frequency       : 200 MHz
Flash Size      : 4 MB
EEPROM Size     : 64 KB
SRAM Size       : 472 KB
```

IFX Low Level Drivers

Reduced set [✓ Full set]

New Project

4. 왼쪽의 Project Explorer 창에서 프로젝트가 생성된 것을 확인한다.

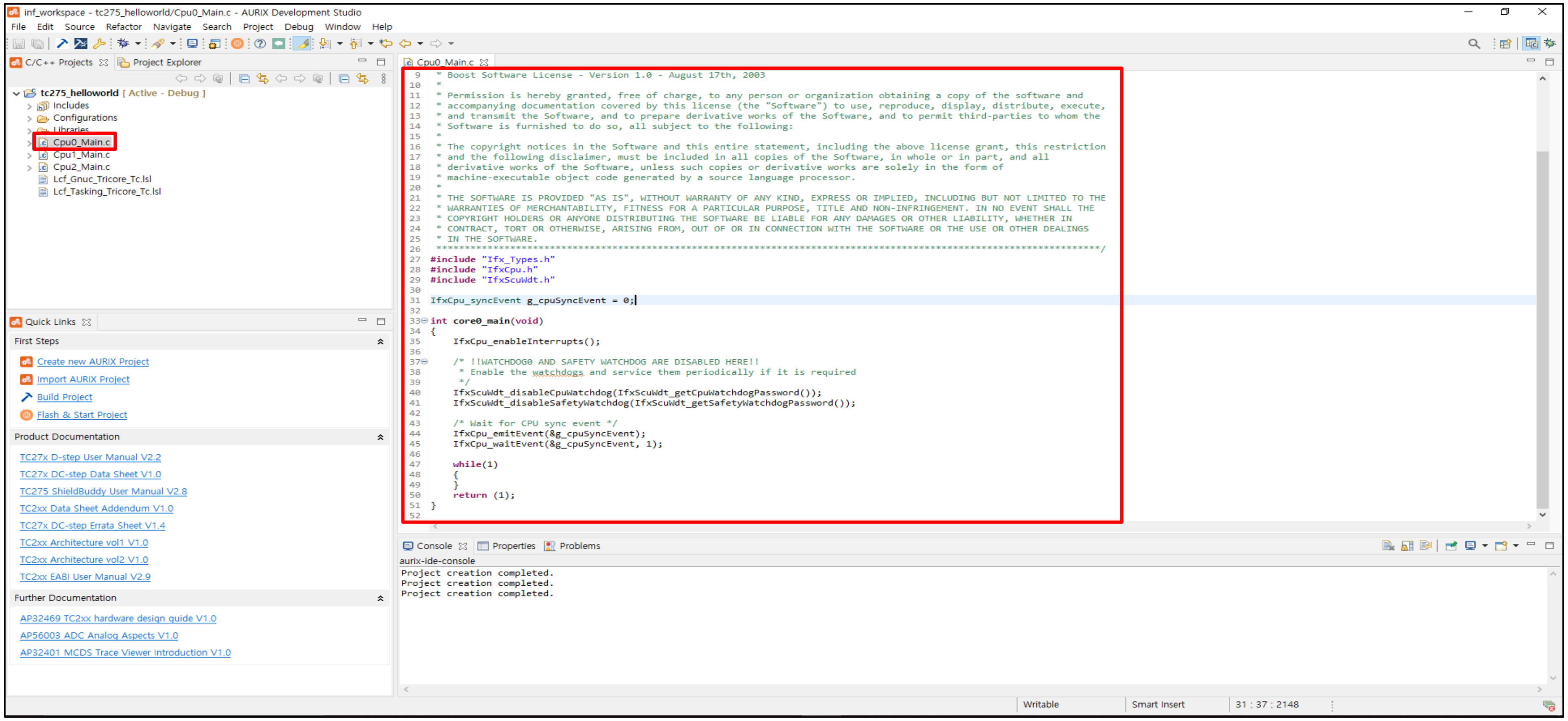

inf_workspace - AURIX Development Studio
File Edit Navigate Search Project Debug Window Help
C/C++ Projects
Project Explorer
tc275_helloworld [Active - Debug]
Includes
Configurations
Libraries
Cpu0_Main.c
Cpu1_Main.c
Cpu2_Main.c
Lcf_Gnuc_Tricore_Tc.lsl
Lcf_Tasking_Tricore_Tc.lsl
Quick Links
First Steps
Create new AURIX Project
Import AURIX Project
Build Project
Flash & Start Project
Product Documentation
TC27x D-step User Manual V2.2
TC27x DC-step Data Sheet V1.0
TC275 ShieldBuddy User Manual V2.8
TC2xx Data Sheet Addendum V1.0
TC27x DC-step Errata Sheet V1.4
TC2xx Architecture vol1 V1.0
TC2xx Architecture vol2 V1.0
TC2xx EABI User Manual V2.9
Further Documentation
AP32469 TC2xx hardware design guide V1.0
AP56003 ADC Analog Aspects V1.0
AP32401 MCDS Trace Viewer Introduction V1.0
Console
Properties
Problems
aurix-ide-console
/tc275_helloworld

```
Project creation completed.
Project creation completed.
Project creation completed.
```

ACE Lab.

Edit Project

5. 왼쪽의 Project Explorer 창에서 **Project name**으로 생성된 파일인 **'Cpu0_Main.c'** 파일을 더블 클릭하여 활성화한다.


inf_workspace - tc275_helloworld/Cpu0_Main.c - AURIX Development Studio
File Edit Source Refactor Navigate Search Project Debug Window Help
C/C++ Projects
Project Explorer
tc275_helloworld [Active - Debug]
Includes
Configurations
Libraries
Cpu0_Main.c
Cpu1_Main.c
Cpu2_Main.c
Lcf_Gnuc_Tricore_Tc.lsl
Lcf_Tasking_Tricore_Tc.lsl
Quick Links
First Steps
Create new AURIX Project
Import AURIX Project
Build Project
Flash & Start Project
Product Documentation
TC27x D-step User Manual V2.2
TC27x DC-step Data Sheet V1.0
TC275 ShieldBuddy User Manual V2.8
TC2xx Data Sheet Addendum V1.0
TC27x DC-step Errata Sheet V1.4
TC2xx Architecture vol1 V1.0
TC2xx Architecture vol2 V1.0
TC2xx EABI User Manual V2.9
Further Documentation
AP32469 TC2xx hardware design guide V1.0
AP56003 ADC Analog Aspects V1.0
AP32401 MCDS Trace Viewer Introduction V1.0
Cpu0_Main.c
Console
Properties
Problems
aurix-ide-console
Writable
Smart Insert
31 : 37 : 2148

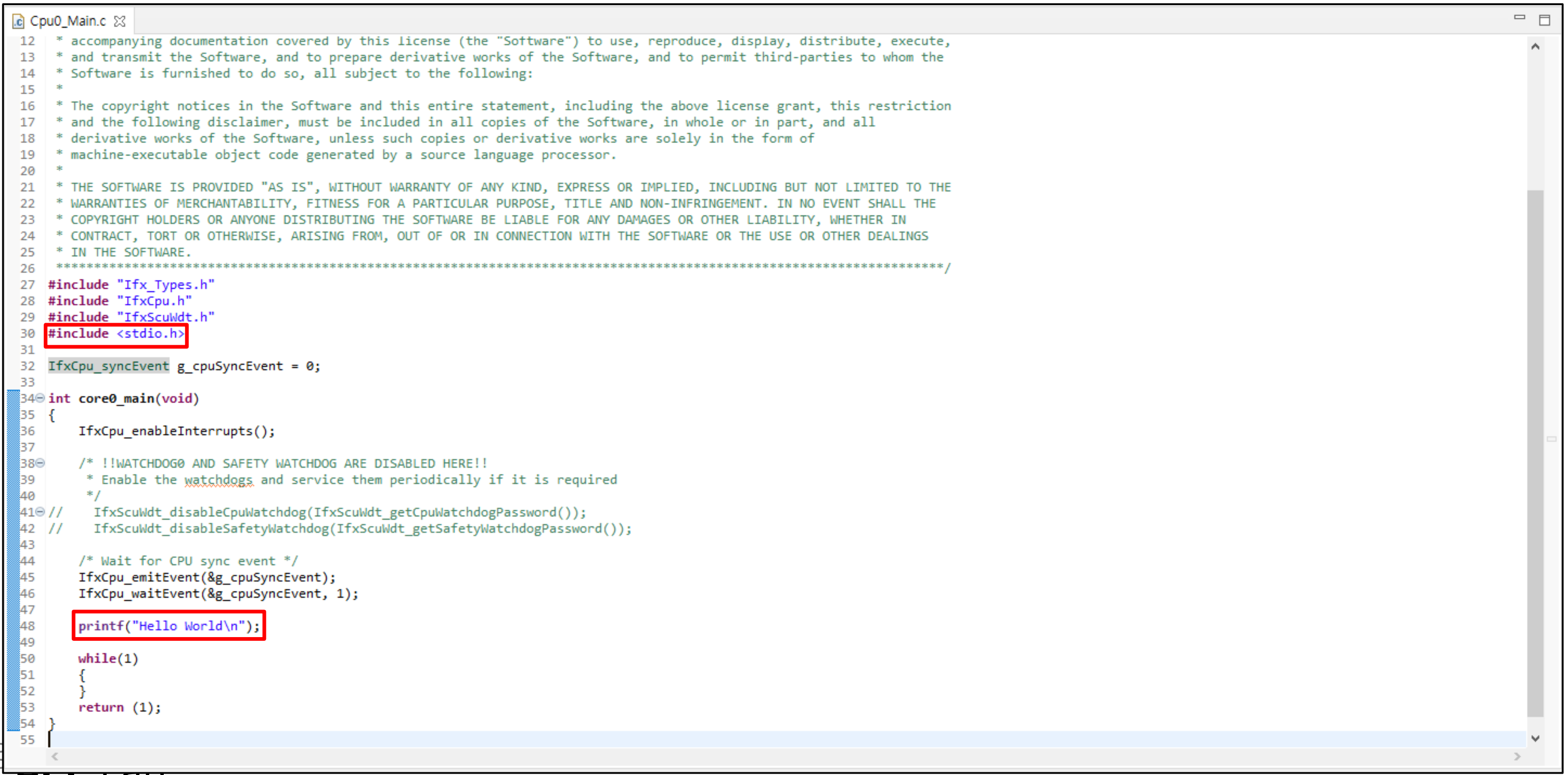
```c
 * Boost Software License - Version 1.0 - August 17th, 2003
 *
 * Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and
 * accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute,
 * and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the
 * Software is furnished to do so, all subject to the following:
 *
 * The copyright notices in the Software and this entire statement, including the above license grant, this restriction
 * and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all
 * derivative works of the Software, unless such copies or derivative works are solely in the form of
 * machine-executable object code generated by a source language processor.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
 * WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE
 * COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN
 * CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
 * IN THE SOFTWARE.
 *********************************************************************************************************************/
#include "Ifx_Types.h"
#include "IfxCpu.h"
#include "IfxScuWdt.h"

IfxCpu_syncEvent g_cpuSyncEvent = 0;

int core0_main(void)
{
    IfxCpu_enableInterrupts();

    /* !!WATCHDOG0 AND SAFETY WATCHDOG ARE DISABLED HERE!!
     * Enable the watchdogs and service them periodically if it is required
     */
    IfxScuWdt_disableCpuWatchdog(IfxScuWdt_getCpuWatchdogPassword());
    IfxScuWdt_disableSafetyWatchdog(IfxScuWdt_getSafetyWatchdogPassword());

    /* Wait for CPU sync event */
    IfxCpu_emitEvent(&g_cpuSyncEvent);
    IfxCpu_waitEvent(&g_cpuSyncEvent, 1);

    while(1)
    {
    }
    return (1);
}

```

```
Project creation completed.
Project creation completed.
Project creation completed.
```

ACE Lab.

Edit Project

6. 'Cpu0_main.c' 파일은 **core0_main 함수를 포함**하고 있으며 이를 수정하여 동작을 설계한다. (본 실습에서는 멀티코어를 사용하지 않으므로 core1_main과 core2_main 함수는 사용하지 않는다.)

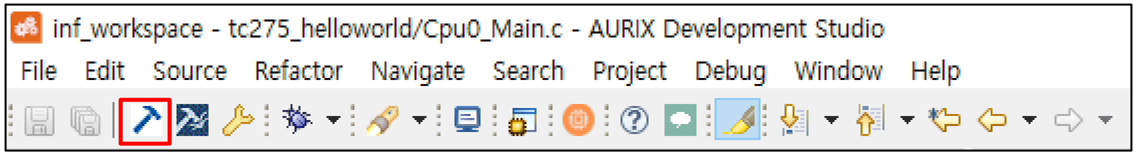


Cpu0_Main.c

```c
 * accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute,
 * and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the
 * Software is furnished to do so, all subject to the following:
 *
 * The copyright notices in the Software and this entire statement, including the above license grant, this restriction
 * and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all
 * derivative works of the Software, unless such copies or derivative works are solely in the form of
 * machine-executable object code generated by a source language processor.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
 * WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE
 * COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN
 * CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
 * IN THE SOFTWARE.
 *********************************************************************************************************************/
#include "Ifx_Types.h"
#include "IfxCpu.h"
#include "IfxScuWdt.h"
#include <stdio.h>

IfxCpu_syncEvent g_cpuSyncEvent = 0;

int core0_main(void)
{
    IfxCpu_enableInterrupts();

    /* !!WATCHDOG0 AND SAFETY WATCHDOG ARE DISABLED HERE!!
     * Enable the watchdogs and service them periodically if it is required
     */
//    IfxScuWdt_disableCpuWatchdog(IfxScuWdt_getCpuWatchdogPassword());
//    IfxScuWdt_disableSafetyWatchdog(IfxScuWdt_getSafetyWatchdogPassword());

    /* Wait for CPU sync event */
    IfxCpu_emitEvent(&g_cpuSyncEvent);
    IfxCpu_waitEvent(&g_cpuSyncEvent, 1);

    printf("Hello World\n");

    while(1)
    {
    }
    return (1);
}
```

Build

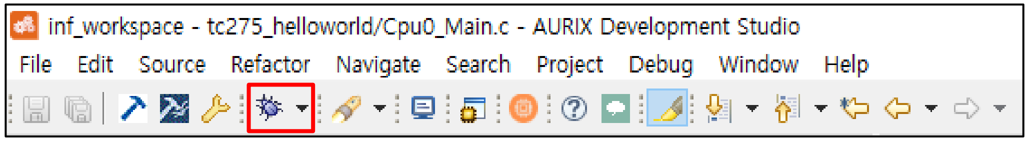
7. 상단의 메뉴에서 **'Build'** 버튼을 클릭하여 Build를 실행한다.
   (Build/Debug는 Active Project에 대해 수행되기 때문에 Build를 수행할 Project를 Active Project로 미리 설정해야 한다. 'Project Explorer – 대상 Project'에서 우클릭 한 뒤, 'Set Active Project'를 클릭하여 Active Project로 설정할 수 있다.)

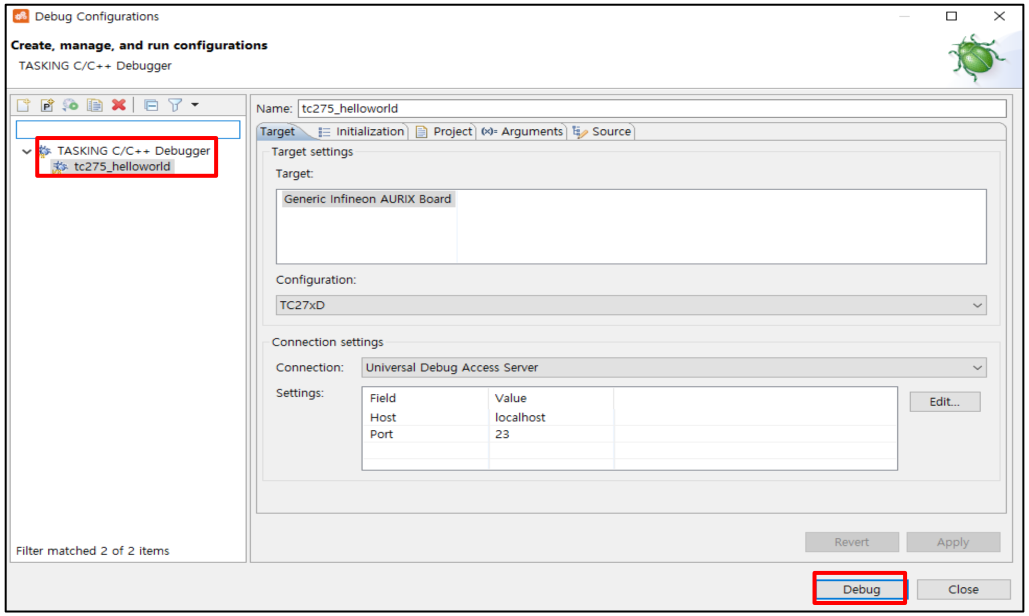

inf_workspace - tc275_helloworld/Cpu0_Main.c - AURIX Development Studio
File Edit Source Refactor Navigate Search Project Debug Window Help

Debug

8. 상단의 메뉴에서 **'Debug'** 버튼을 클릭하여 Debug를 실행한다.

✓ 'Debug' 버튼을 처음으로 클릭하면 'Debug Configurations' 창이 활성화된다. 왼쪽의 창에서 **'TASKING C/C++ Debugger – Project name'**을 확인하고 **'Debug'** 버튼을 클릭한다.
(이후에는 Debug가 바로 실행되며 'Debug Configurations' 창을 활성화하기 위해서는 'Debug' 버튼의 오른쪽 화살표를 클릭한 후 'Debug Configurations…' 버튼을 클릭한다.)


inf_workspace - tc275_helloworld/Cpu0_Main.c - AURIX Development Studio
File Edit Source Refactor Navigate Search Project Debug Window Help



Name: tc275_helloworld

Target settings

Target: Generic Infineon AURIX Board

Configuration: TC27xD

Connection settings

Connection: Universal Debug Access Server

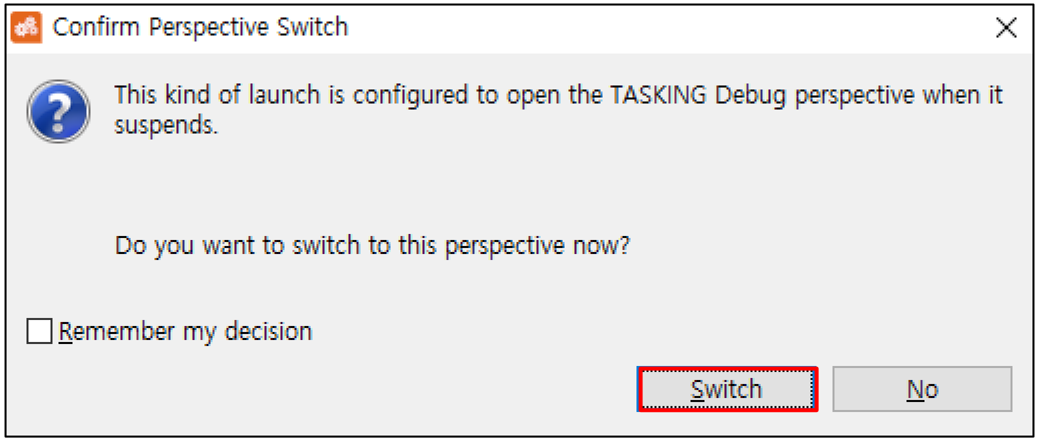
Settings:

Field	Value
Host	localhost
Port	23

ACE Lab.

Debug

8. 상단의 메뉴에서 **'Debug'** 버튼을 클릭하여 Debug를 실행한다.
   - ✓ 'Confirm Perspective Switch' 창이 뜨면 Switch를 눌러 디버그 창으로 전환한다.



Confirm Perspective Switch

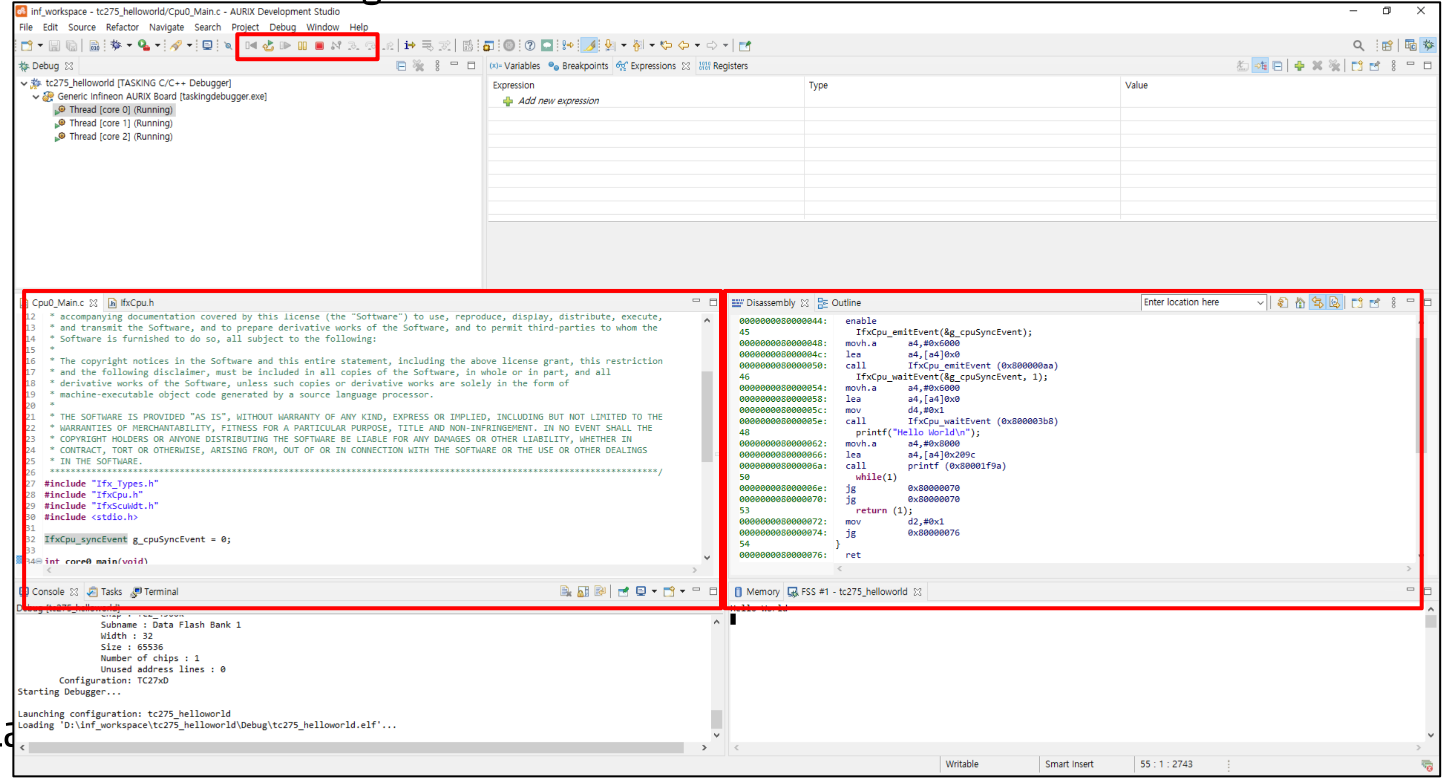
This kind of launch is configured to open the TASKING Debug perspective when it suspends.

Do you want to switch to this perspective now?

☐ Remember my decision

Debug

9. Debug 화면을 확인한다.
   - ✓ 상단의 Restart / Resume / Terminate / Step Into / Step Over 등을 클릭하여 실행을 제어가능
   - ✓ 가운데 창을 통해 소스 코드 및 어셈블리 코드를 확인할 수 있다.
   - ✓ 만약, 소스 코드 화면에 에러 메시지가 표시되면 'Terminate' 버튼을 클릭하여 Debug를 종료했다가 다시 Debug를 실행한다.


inf_workspace - tc275_helloworld/Cpu0_Main.c - AURIX Development Studio
File Edit Source Refactor Navigate Search Project Debug Window Help
Debug
tc275_helloworld [TASKING C/C++ Debugger]
Generic Infineon AURIX Board [taskingdebugger.exe]
Thread [core 0] (Running)
Thread [core 1] (Running)
Thread [core 2] (Running)
Variables Breakpoints Expressions Registers
Expression Type Value
Add new expression
Cpu0_Main.c IfxCpu.h
Disassembly Outline
Enter location here
Console Tasks Terminal
Memory FSS #1 - tc275_helloworld
Writable Smart Insert 55 : 1 : 2743

```
 * accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute,
 * and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the
 * Software is furnished to do so, all subject to the following:
 *
 * The copyright notices in the Software and this entire statement, including the above license grant, this restriction
 * and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all
 * derivative works of the Software, unless such copies or derivative works are solely in the form of
 * machine-executable object code generated by a source language processor.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
 * WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE
 * COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN
 * CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
 * IN THE SOFTWARE.
 *********************************************************************************************************************/
#include "Ifx_Types.h"
#include "IfxCpu.h"
#include "IfxScuWdt.h"
#include <stdio.h>

IfxCpu_syncEvent g_cpuSyncEvent = 0;

int core0_main(void)
```

```
0000000080000044:   enable
45                    IfxCpu_emitEvent(&g_cpuSyncEvent);
0000000080000048:   movh.a      a4,#0x6000
000000008000004c:   lea         a4,[a4]0x0
0000000080000050:   call        IfxCpu_emitEvent (0x800000aa)
46                    IfxCpu_waitEvent(&g_cpuSyncEvent, 1);
0000000080000054:   movh.a      a4,#0x6000
0000000080000058:   lea         a4,[a4]0x0
000000008000005c:   mov         d4,#0x1
000000008000005e:   call        IfxCpu_waitEvent (0x800003b8)
48                    printf("Hello World\n");
0000000080000062:   movh.a      a4,#0x8000
0000000080000066:   lea         a4,[a4]0x209c
000000008000006a:   call        printf (0x80001f9a)
50                    while(1)
000000008000006e:   jg          0x80000070
0000000080000070:   jg          0x80000070
53                    return (1);
0000000080000072:   mov         d2,#0x1
0000000080000074:   jg          0x80000076
54                  }
0000000080000076:   ret
```

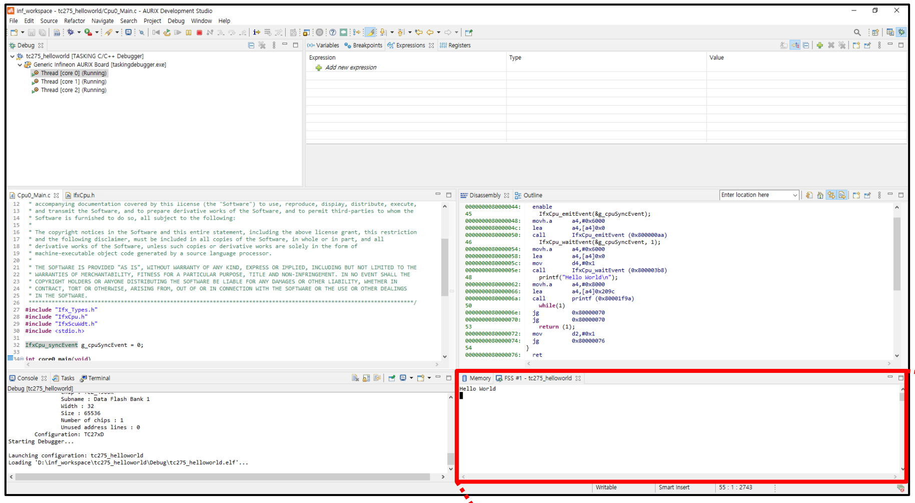
```
Debug [tc275_helloworld]
                Subname : Data Flash Bank 1
                Width : 32
                Size : 65536
                Number of chips : 1
                Unused address lines : 0
        Configuration: TC27xD
Starting Debugger...

Launching configuration: tc275_helloworld
Loading 'D:\inf_workspace\tc275_helloworld\Debug\tc275_helloworld.elf'...
```

ACE La

Debug

10. Debug를 통해 실행을 제어하고 결과를 확인한다.

- ✓ 상단의 **'Resume'** 버튼을 클릭하여 동작을 실행시킨다.
- ✓ 동작의 실행 결과로 우측 하단 창 (FSS)에 'Hello world'라는 메시지가 표시되는 것을 확인한다.



inf_workspace - tc275_helloworld/Cpu0_Main.c - AURIX Development Studio

```
tc275_helloworld [TASKING C/C++ Debugger]
  Generic Infineon AURIX Board [taskingdebugger.exe]
    Thread [core 0] (Running)
    Thread [core 1] (Running)
    Thread [core 2] (Running)
```

Cpu0_Main.c

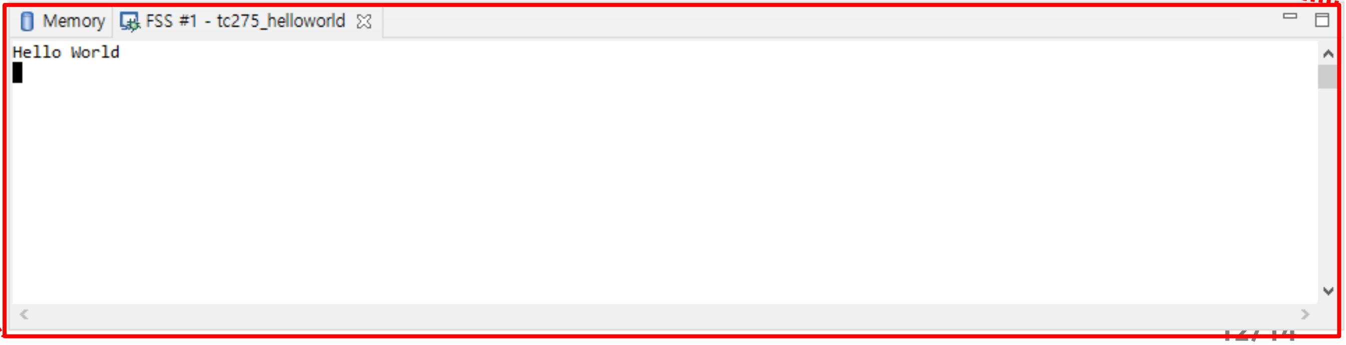
```
 * accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute,
 * and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the
 * Software is furnished to do so, all subject to the following:
 *
 * The copyright notices in the Software and this entire statement, including the above license grant, this restriction
 * and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all
 * derivative works of the Software, unless such copies or derivative works are solely in the form of
 * machine-executable object code generated by a source language processor.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
 * WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE
 * COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN
 * CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
 * IN THE SOFTWARE.
 *********************************************************************************************************************/
#include "Ifx_Types.h"
#include "IfxCpu.h"
#include "IfxScuWdt.h"
#include <stdio.h>

IfxCpu_syncEvent g_cpuSyncEvent = 0;

int core0_main(void)
```

Disassembly

```
0000000080000044:   enable
45                    IfxCpu_emitEvent(&g_cpuSyncEvent);
0000000080000048:   movh.a      a4,#0x6000
000000008000004c:   lea         a4,[a4]0x0
0000000080000050:   call        IfxCpu_emitEvent (0x800000aa)
46                    IfxCpu_waitEvent(&g_cpuSyncEvent, 1);
0000000080000054:   movh.a      a4,#0x6000
0000000080000058:   lea         a4,[a4]0x0
000000008000005c:   mov         d4,#0x1
000000008000005e:   call        IfxCpu_waitEvent (0x800003b8)
48                    printf("Hello World\n");
0000000080000062:   movh.a      a4,#0x8000
0000000080000066:   lea         a4,[a4]0x209c
000000008000006a:   call        printf (0x80001f9a)
50                    while(1)
000000008000006e:   jg          0x80000070
0000000080000070:   jg          0x80000070
53                    return (1);
0000000080000072:   mov         d2,#0x1
0000000080000074:   jg          0x80000076
54                  }
0000000080000076:   ret
```

Console — Debug [tc275_helloworld]

```
                Subname : Data Flash Bank 1
                Width : 32
                Size : 65536
                Number of chips : 1
                Unused address lines : 0
        Configuration: TC27xD
Starting Debugger...

Launching configuration: tc275_helloworld
Loading 'D:\inf_workspace\tc275_helloworld\Debug\tc275_helloworld.elf'...
```



Memory | FSS #1 - tc275_helloworld

```
Hello World
```

ACE Lab.

System Timer Test

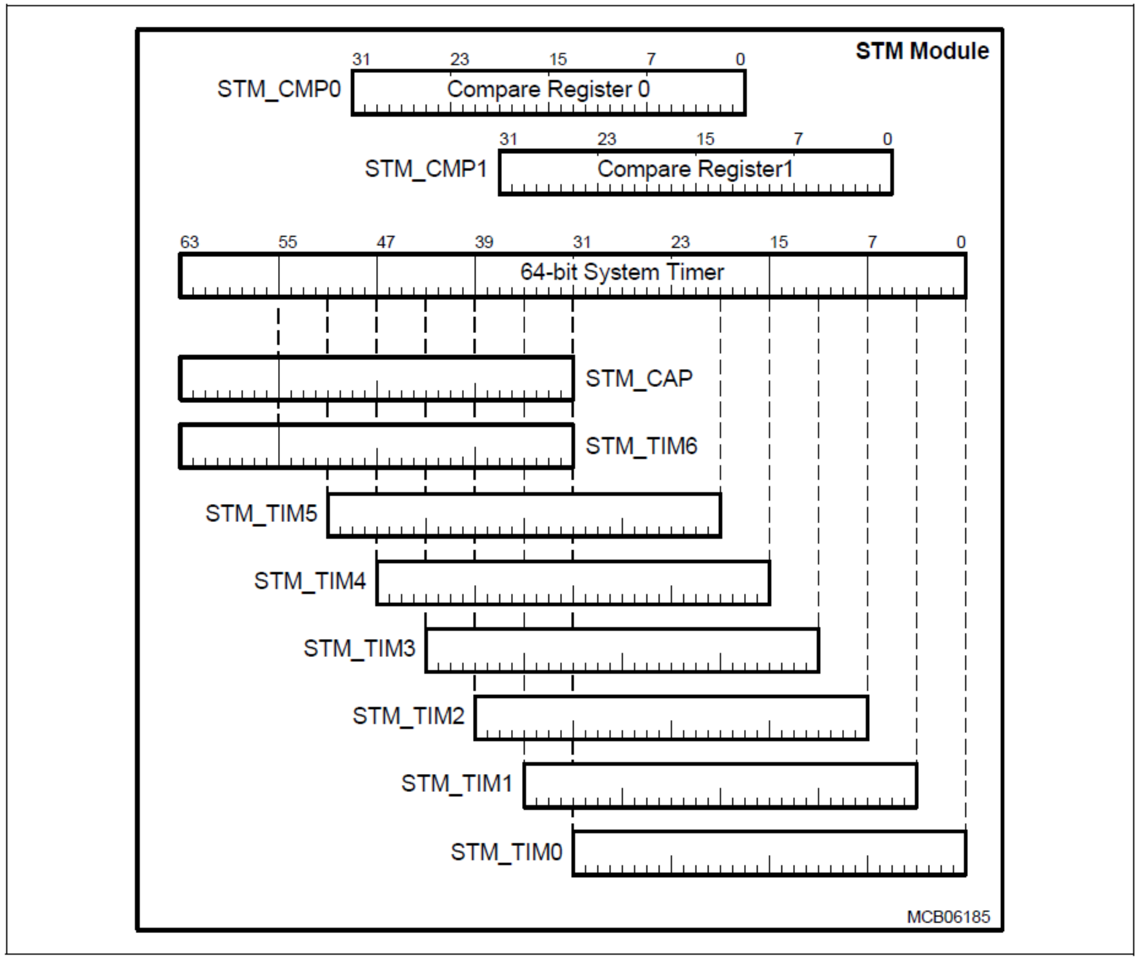


Figure 17-1 General Block Diagram of the STM Module

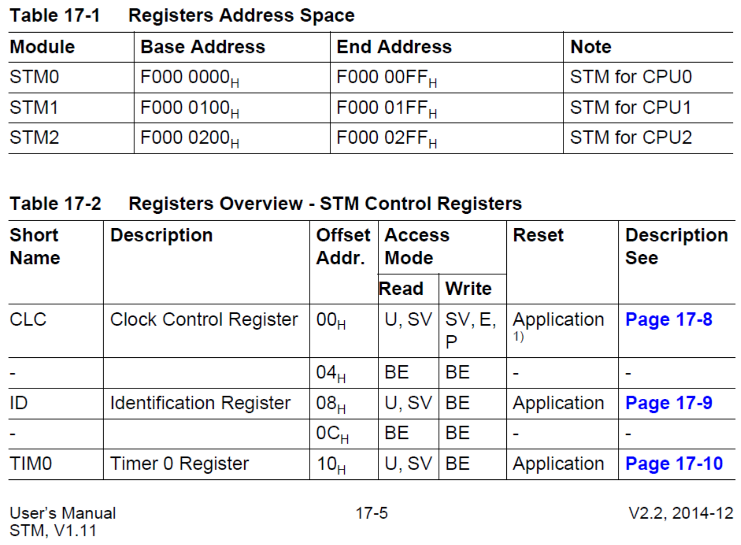


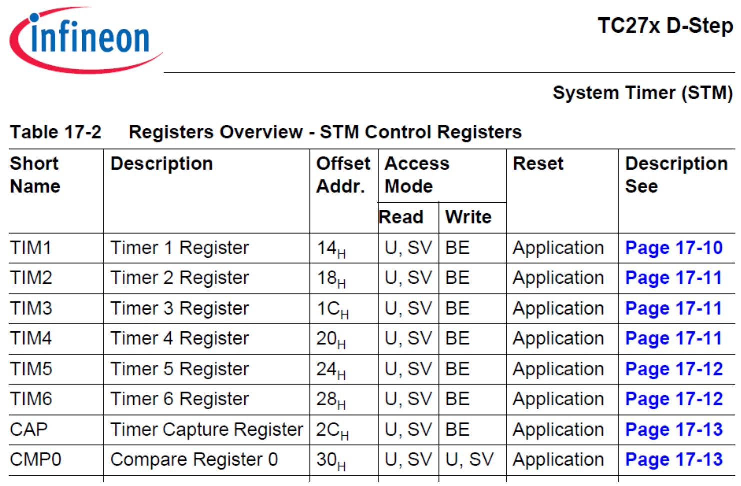
Table 17-1 Registers Address Space

Module	Base Address	End Address	Note
STM0	F000 0000$_H$	F000 00FF$_H$	STM for CPU0
STM1	F000 0100$_H$	F000 01FF$_H$	STM for CPU1
STM2	F000 0200$_H$	F000 02FF$_H$	STM for CPU2

Table 17-2 Registers Overview - STM Control Registers

Short Name	Description	Offset Addr.	Access Mode		Reset	Description See
			Read	Write		
CLC	Clock Control Register	00$_H$	U, SV	SV, E, P	Application [1]	Page 17-8
-		04$_H$	BE	BE	-	-
ID	Identification Register	08$_H$	U, SV	BE	Application	Page 17-9
-		0C$_H$	BE	BE	-	-
TIM0	Timer 0 Register	10$_H$	U, SV	BE	Application	Page 17-10

User's Manual 17-5 V2.2, 2014-12
STM, V1.11



TC27x D-Step

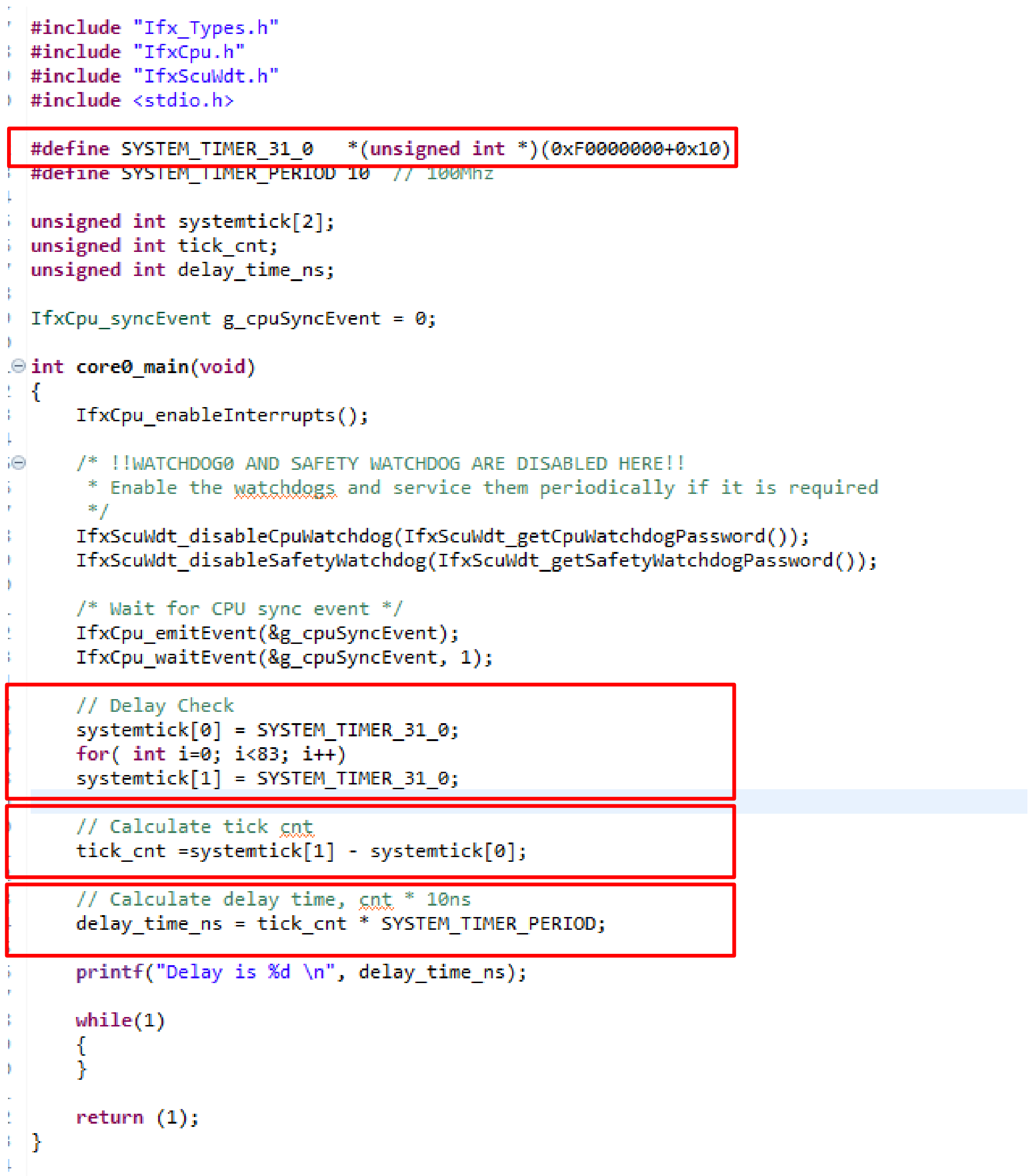
System Timer (STM)

Table 17-2 Registers Overview - STM Control Registers

Short Name	Description	Offset Addr.	Access Mode		Reset	Description See
			Read	Write		
TIM1	Timer 1 Register	14$_H$	U, SV	BE	Application	Page 17-10
TIM2	Timer 2 Register	18$_H$	U, SV	BE	Application	Page 17-11
TIM3	Timer 3 Register	1C$_H$	U, SV	BE	Application	Page 17-11
TIM4	Timer 4 Register	20$_H$	U, SV	BE	Application	Page 17-11
TIM5	Timer 5 Register	24$_H$	U, SV	BE	Application	Page 17-12
TIM6	Timer 6 Register	28$_H$	U, SV	BE	Application	Page 17-12
CAP	Timer Capture Register	2C$_H$	U, SV	BE	Application	Page 17-13
CMP0	Compare Register 0	30$_H$	U, SV	U, SV	Application	Page 17-13

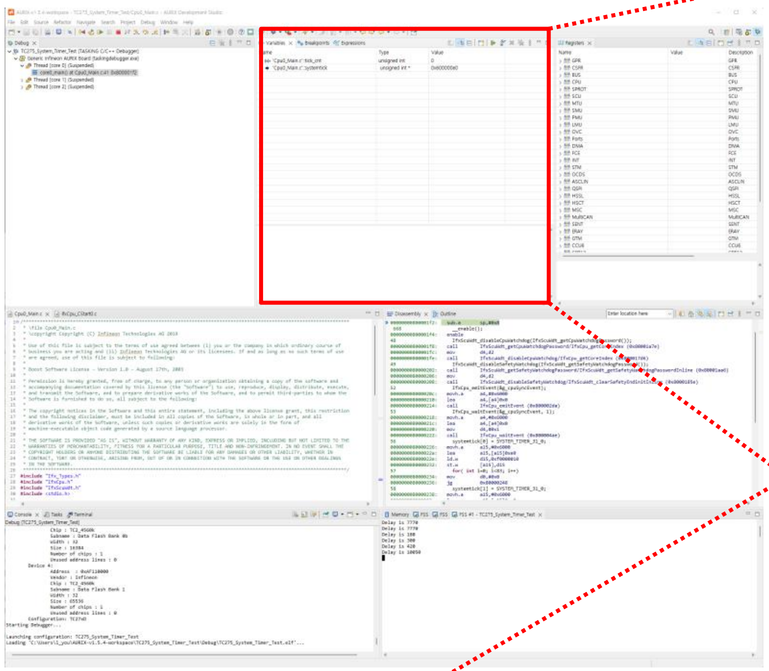
ACE Lab.

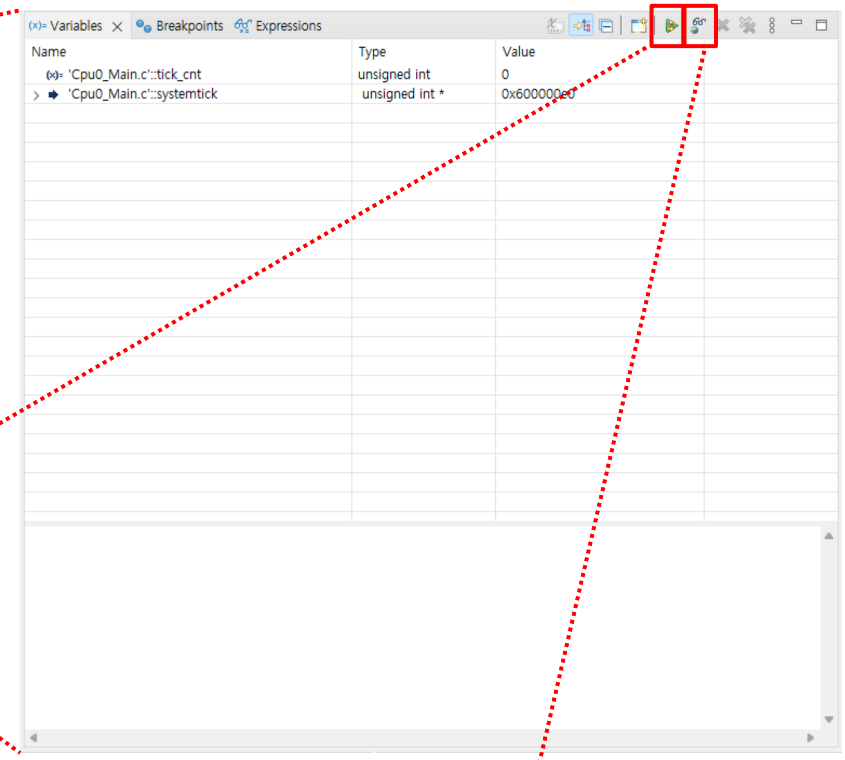
System Timer Test



```c
#include "Ifx_Types.h"
#include "IfxCpu.h"
#include "IfxScuWdt.h"
#include <stdio.h>

#define SYSTEM_TIMER_31_0   *(unsigned int *)(0xF0000000+0x10)
#define SYSTEM_TIMER_PERIOD 10  // 100Mhz

unsigned int systemtick[2];
unsigned int tick_cnt;
unsigned int delay_time_ns;

IfxCpu_syncEvent g_cpuSyncEvent = 0;

int core0_main(void)
{
    IfxCpu_enableInterrupts();

    /* !!WATCHDOG0 AND SAFETY WATCHDOG ARE DISABLED HERE!!
     * Enable the watchdogs and service them periodically if it is required
     */
    IfxScuWdt_disableCpuWatchdog(IfxScuWdt_getCpuWatchdogPassword());
    IfxScuWdt_disableSafetyWatchdog(IfxScuWdt_getSafetyWatchdogPassword());

    /* Wait for CPU sync event */
    IfxCpu_emitEvent(&g_cpuSyncEvent);
    IfxCpu_waitEvent(&g_cpuSyncEvent, 1);

    // Delay Check
    systemtick[0] = SYSTEM_TIMER_31_0;
    for( int i=0; i<83; i++)
    systemtick[1] = SYSTEM_TIMER_31_0;

    // Calculate tick cnt
    tick_cnt =systemtick[1] - systemtick[0];

    // Calculate delay time, cnt * 10ns
    delay_time_ns = tick_cnt * SYSTEM_TIMER_PERIOD;

    printf("Delay is %d \n", delay_time_ns);

    while(1)
    {
    }

    return (1);
}
```

Variable



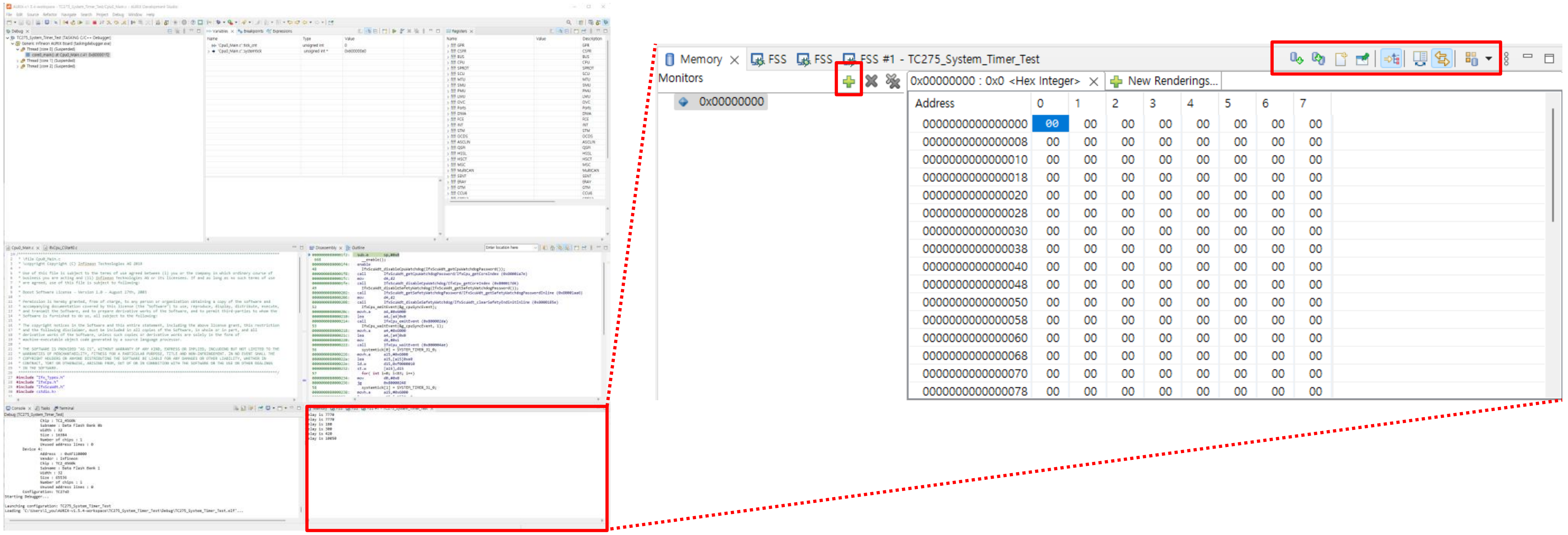


Name	Type	Value
(x)= 'Cpu0_Main.c'::tick_cnt	unsigned int	0
'Cpu0_Main.c'::systemtick	unsigned int *	0x600000e0

- RTM Mode 동작
- Varialbe 창에 추가된 변수를 실시간 값을 표현한다.

- Variable 창에 Add Variable을 통해서 전역 변수 추가 가능
- 지역변수는 자동으로 표시됨.
- 변수값은 Suspend 상태에서만 표시됨.

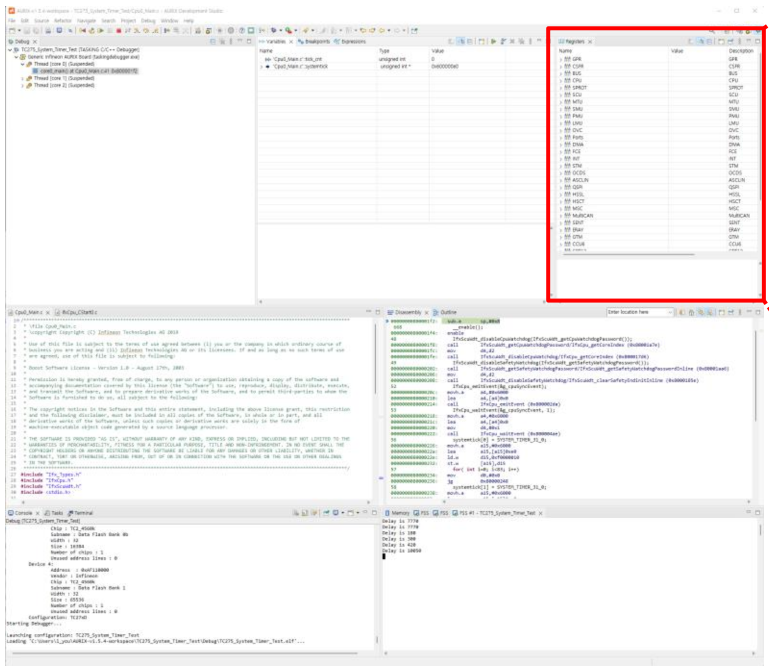
Memory

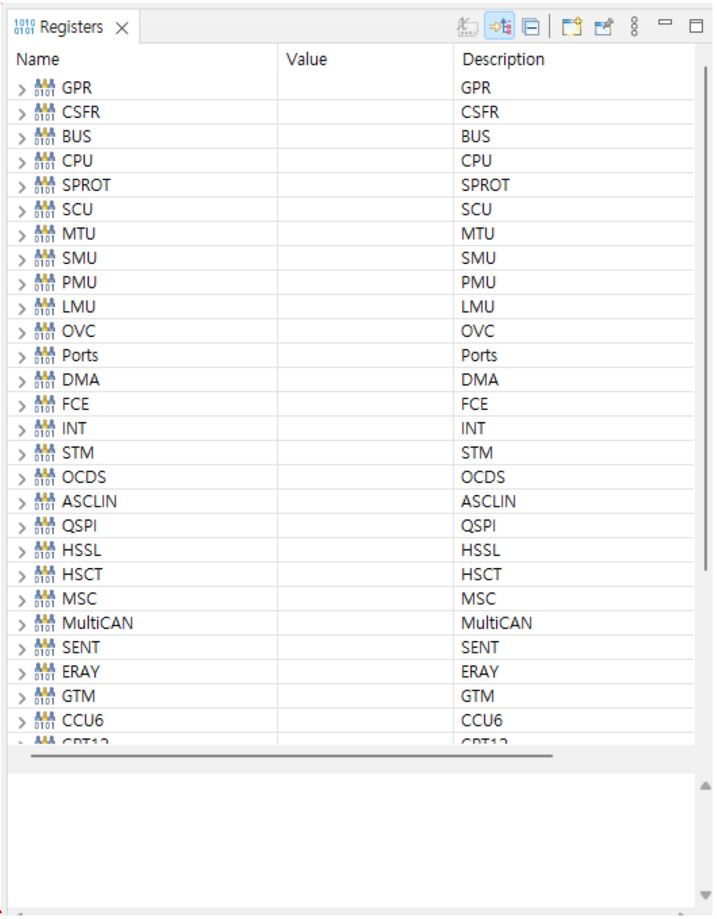

Memory
FSS
FSS
FSS #1 - TC275_System_Timer_Test
Monitors
0x00000000
0x00000000 : 0x0 <Hex Integer>
New Renderings...

```
Address            0   1   2   3   4   5   6   7
0000000000000000   00  00  00  00  00  00  00  00
0000000000000008   00  00  00  00  00  00  00  00
0000000000000010   00  00  00  00  00  00  00  00
0000000000000018   00  00  00  00  00  00  00  00
0000000000000020   00  00  00  00  00  00  00  00
0000000000000028   00  00  00  00  00  00  00  00
0000000000000030   00  00  00  00  00  00  00  00
0000000000000038   00  00  00  00  00  00  00  00
0000000000000040   00  00  00  00  00  00  00  00
0000000000000048   00  00  00  00  00  00  00  00
0000000000000050   00  00  00  00  00  00  00  00
0000000000000058   00  00  00  00  00  00  00  00
0000000000000060   00  00  00  00  00  00  00  00
0000000000000068   00  00  00  00  00  00  00  00
0000000000000070   00  00  00  00  00  00  00  00
0000000000000078   00  00  00  00  00  00  00  00
```

- Memory Tab을 클릭하여 사용가능
- +를 클릭해서 확인하고자 하는 Address를 입력해서 Dump 할 수 있다.
- 8-bit 기준으로 설정한 주소를 기준으로 Display 된다.

Registers



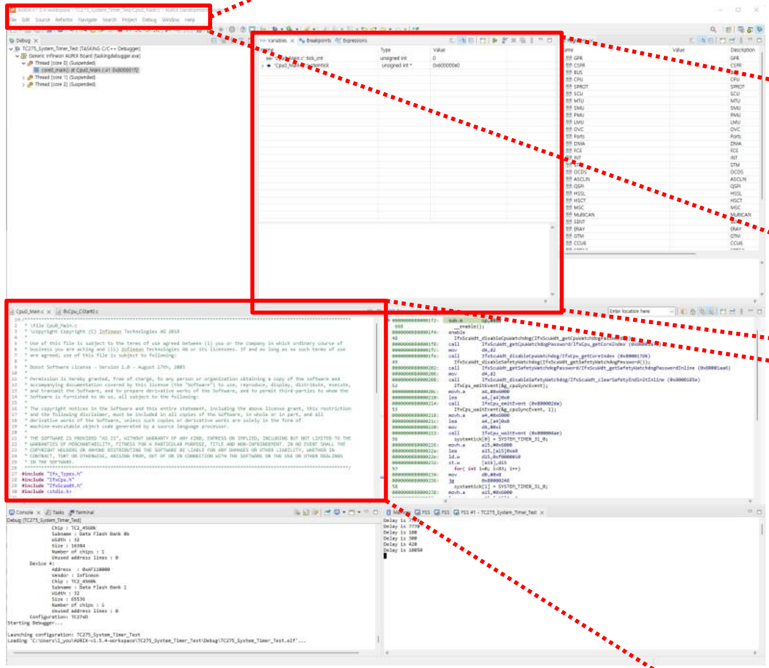


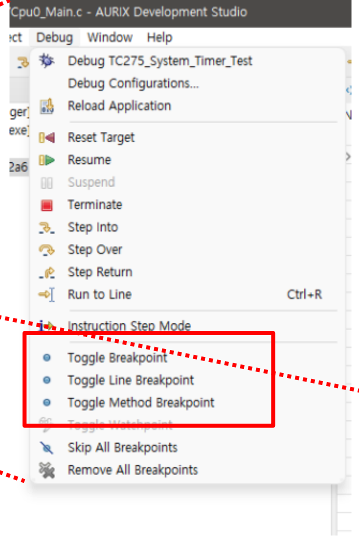
Name	Value	Description
GPR		GPR
CSFR		CSFR
BUS		BUS
CPU		CPU
SPROT		SPROT
SCU		SCU
MTU		MTU
SMU		SMU
PMU		PMU
LMU		LMU
OVC		OVC
Ports		Ports
DMA		DMA
FCE		FCE
INT		INT
STM		STM
OCDS		OCDS
ASCLIN		ASCLIN
QSPI		QSPI
HSSL		HSSL
HSCT		HSCT
MSC		MSC
MultiCAN		MultiCAN
SENT		SENT
ERAY		ERAY
GTM		GTM
CCU6		CCU6

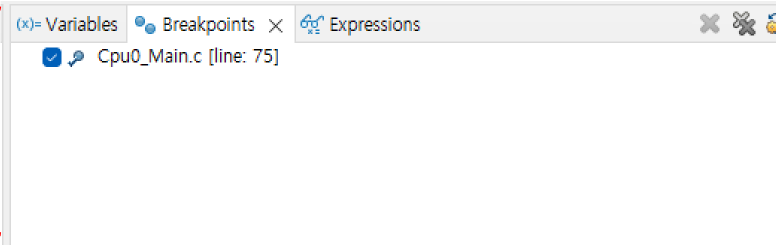
- Registers Tab을 이용하여 사용가능
- 각 Peripheral Register 상태를 확인이 가능하다.
- Read only / Wright Only를 고려해서 확인을 해야 한다.

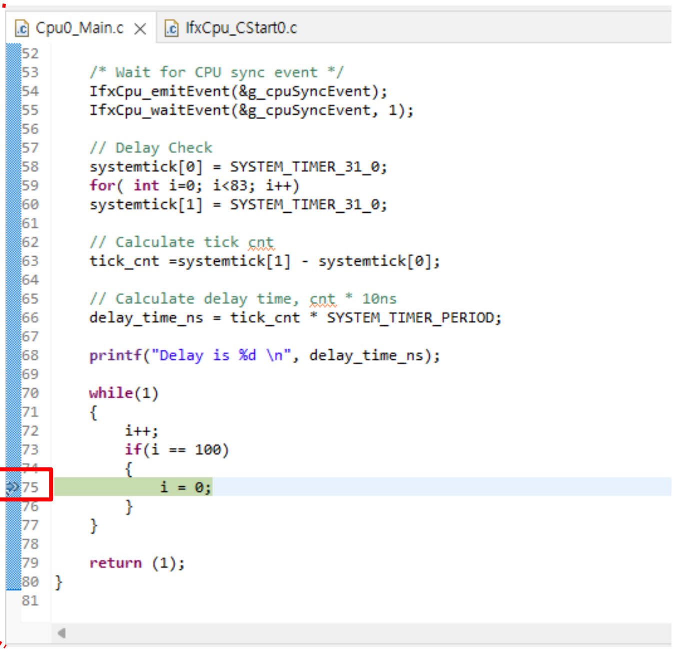
ACE Lab.

Break









```
    /* Wait for CPU sync event */
    IfxCpu_emitEvent(&g_cpuSyncEvent);
    IfxCpu_waitEvent(&g_cpuSyncEvent, 1);

    // Delay Check
    systemtick[0] = SYSTEM_TIMER_31_0;
    for( int i=0; i<83; i++)
    systemtick[1] = SYSTEM_TIMER_31_0;

    // Calculate tick cnt
    tick_cnt =systemtick[1] - systemtick[0];

    // Calculate delay time, cnt * 10ns
    delay_time_ns = tick_cnt * SYSTEM_TIMER_PERIOD;

    printf("Delay is %d \n", delay_time_ns);

    while(1)
    {
        i++;
        if(i == 100)
        {
            i = 0;
        }
    }

    return (1);
}
```

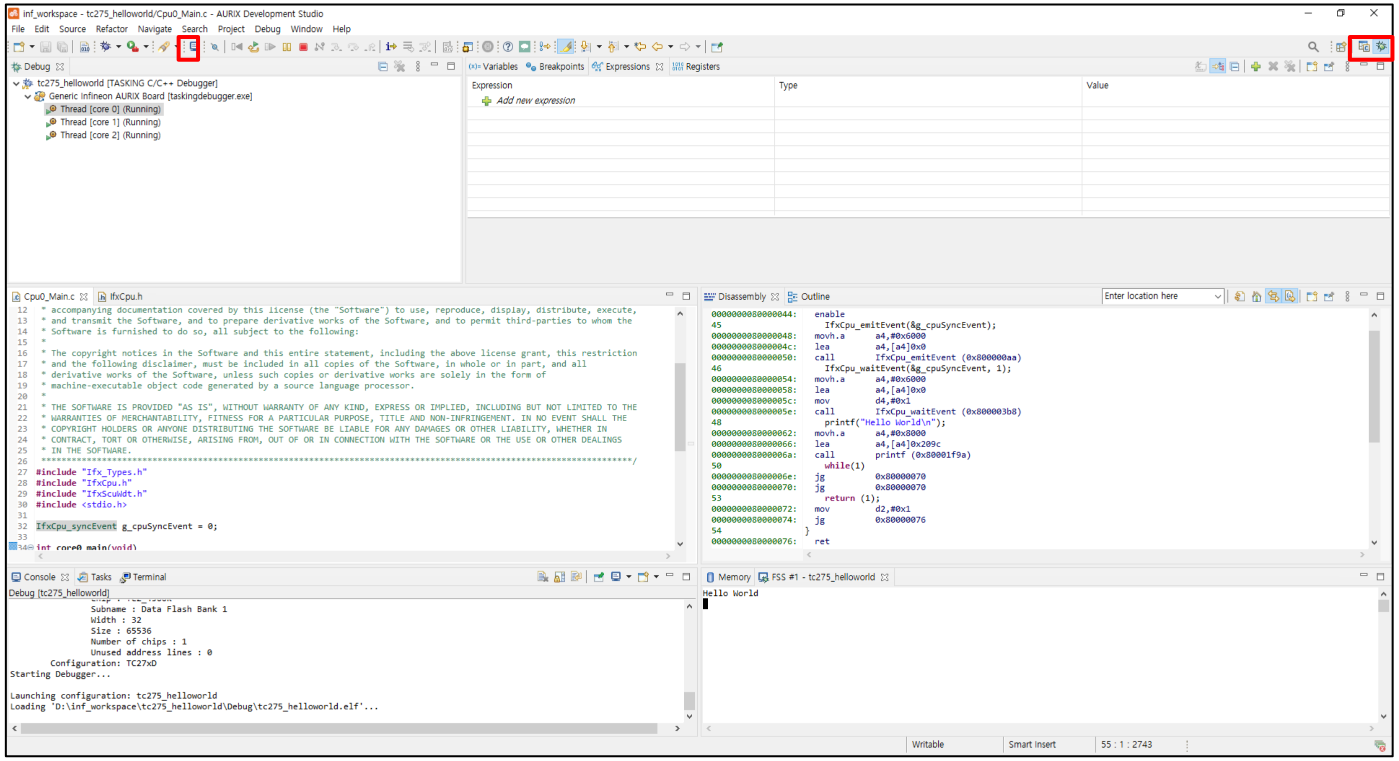
- Code tab에서 원하는 line을 클릭해서 Debugger menu를 클릭해서 break를 선택
- 또는 Code line을 Click하면 Break가 가능한 영역은 파란 띠가 나타나고 파란 라인에 Double Click 하면 Break를 설정 가능하다.
- Breakpoints tab을 통해서 설정된 break를 확인가능

Debug

11. Debug를 종료한다.

- ✓ 상단의 **'Terminate'** 버튼을 클릭하여 Debug를 종료한다.

  (Debug 종료 시, 꼭 'Terminate' 버튼을 클릭하여 정상적으로 종료한다.)
- ✓ Debug 종료 후, 소스코드 편집 화면으로 돌아가기 위해서는 우측 상단의 'C/C++'을 클릭한다.



inf_workspace - tc275_helloworld/Cpu0_Main.c - AURIX Development Studio

```
tc275_helloworld [TASKING C/C++ Debugger]
  Generic Infineon AURIX Board [taskingdebugger.exe]
    Thread [core 0] (Running)
    Thread [core 1] (Running)
    Thread [core 2] (Running)
```

```c
 * accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute,
 * and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the
 * Software is furnished to do so, all subject to the following:
 *
 * The copyright notices in the Software and this entire statement, including the above license grant, this restriction
 * and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all
 * derivative works of the Software, unless such copies or derivative works are solely in the form of
 * machine-executable object code generated by a source language processor.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
 * WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE
 * COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN
 * CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
 * IN THE SOFTWARE.
 *********************************************************************************************************************/
#include "Ifx_Types.h"
#include "IfxCpu.h"
#include "IfxScuWdt.h"
#include <stdio.h>

IfxCpu_syncEvent g_cpuSyncEvent = 0;

int core0_main(void)
```

```
0000000080000044:   enable
45                    IfxCpu_emitEvent(&g_cpuSyncEvent);
0000000080000048:   movh.a      a4,#0x6000
000000008000004c:   lea         a4,[a4]0x0
0000000080000050:   call        IfxCpu_emitEvent (0x800000aa)
46                    IfxCpu_waitEvent(&g_cpuSyncEvent, 1);
0000000080000054:   movh.a      a4,#0x6000
0000000080000058:   lea         a4,[a4]0x0
000000008000005c:   mov         d4,#0x1
000000008000005e:   call        IfxCpu_waitEvent (0x800003b8)
48                    printf("Hello World\n");
0000000080000062:   movh.a      a4,#0x8000
0000000080000066:   lea         a4,[a4]0x209c
000000008000006a:   call        printf (0x80001f9a)
50                    while(1)
000000008000006e:   jg          0x80000070
0000000080000070:   jg          0x80000070
53                    return (1);
0000000080000072:   mov         d2,#0x1
0000000080000074:   jg          0x80000076
54                }
0000000080000076:   ret
```

Debug [tc275_helloworld]

```
                Subname : Data Flash Bank 1
                Width : 32
                Size : 65536
                Number of chips : 1
                Unused address lines : 0
        Configuration: TC27xD
Starting Debugger...

Launching configuration: tc275_helloworld
Loading 'D:\inf_workspace\tc275_helloworld\Debug\tc275_helloworld.elf'...
```

FSS #1 - tc275_helloworld

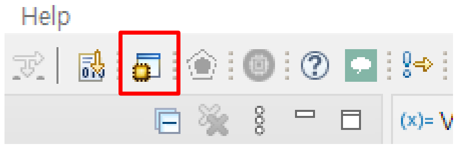
```
Hello World
```

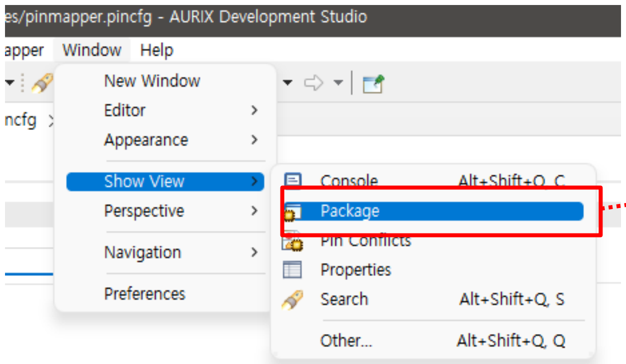
ACE Lab.

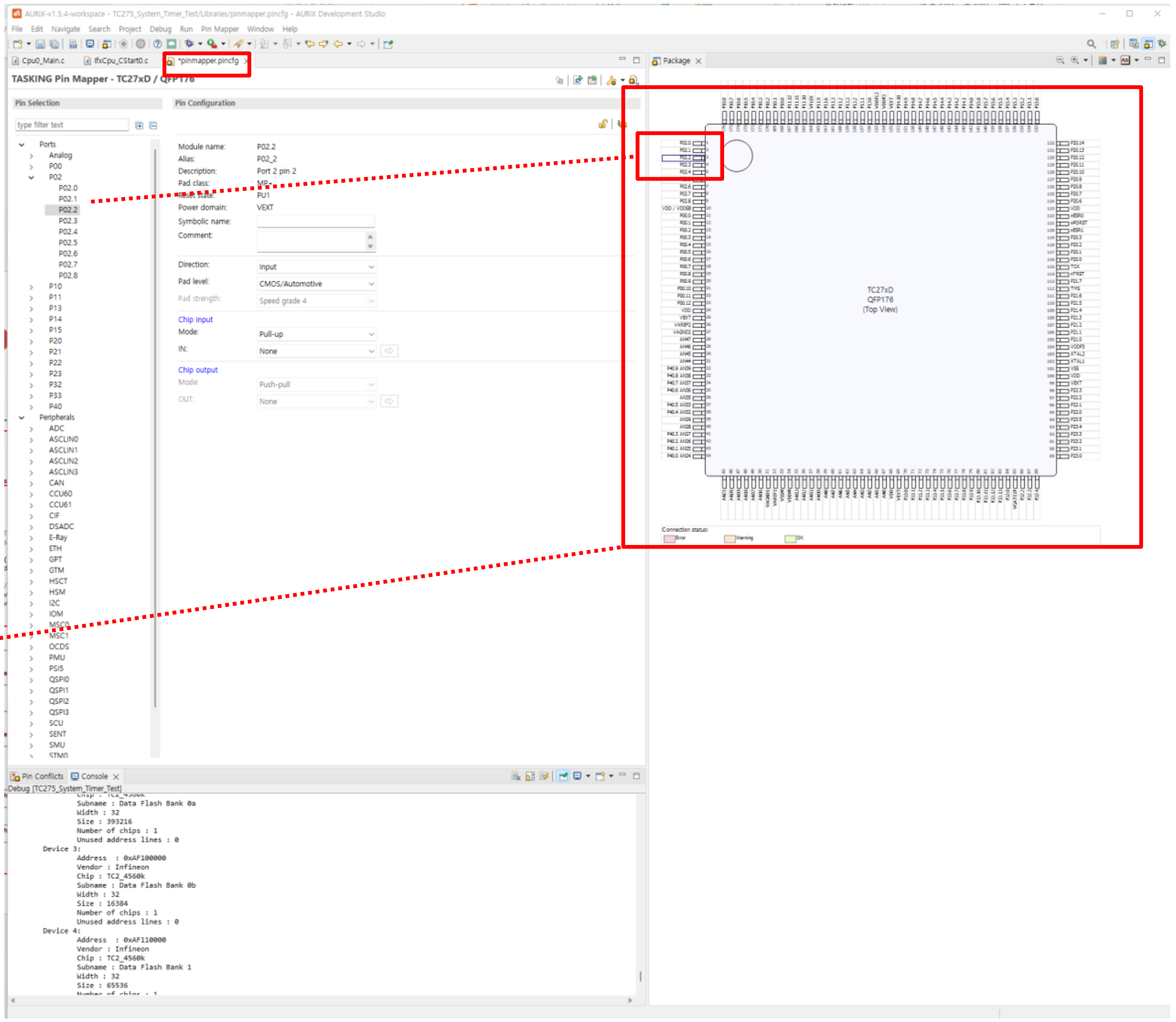
단축키

✓ Ctrl-B	Build	
✓ Ctrl-F2	Exit Debug	
✓ F5	Step Into	중단점 다음 라인, 다음 라인이 함수라면 함수 내부로 들어간다.
✓ F6	Step Over	중단점 다음 라인, 다음 라인이 함수라면 실행하되 내부로 들어가지는 않는다.
✓ F7	Step Return	현재 함수의 리턴으로 이동한다. 함수를 빠져 나온다.
✓ F8	Run	
✓ F3	Open Declation	

Pin Map







Pin Configuration

Module name: P02.2
Alias: P02_2
Description: Port 2 pin 2
Pad class: MP+
Reset state: PU1
Power domain: VEXT
Symbolic name:
Comment:

Direction: Input
Pad level: CMOS/Automotive
Pad strength: Speed grade 4

Chip input
Mode: Pull-up
IN: None

Chip output
Mode: Push-pull
OUT: None

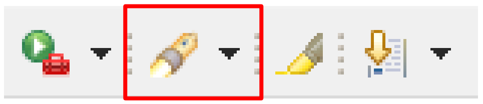
Debug [TC275_System_Timer_Test]

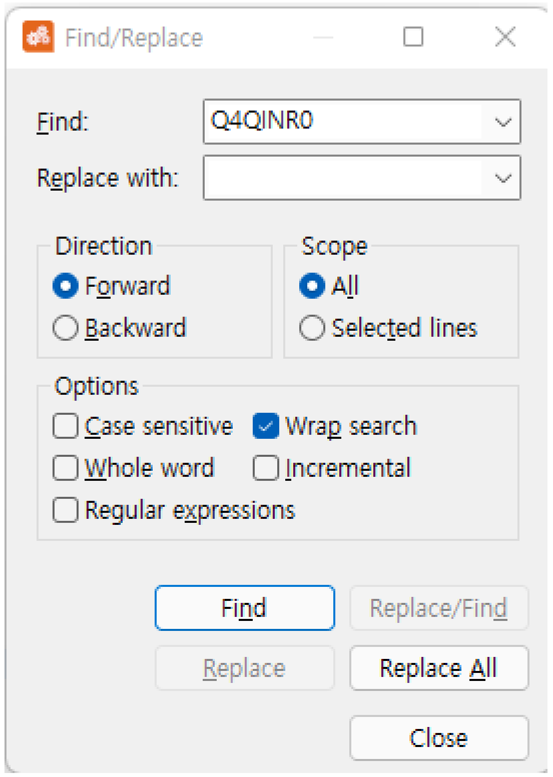
```
                Subname : Data Flash Bank 0a
                Width : 32
                Size : 393216
                Number of chips : 1
                Unused address lines : 0
        Device 3:
                Address  : 0xAF100000
                Vendor : Infineon
                Chip : TC2_4560k
                Subname : Data Flash Bank 0b
                Width : 32
                Size : 16384
                Number of chips : 1
                Unused address lines : 0
        Device 4:
                Address  : 0xAF110000
                Vendor : Infineon
                Chip : TC2_4560k
                Subname : Data Flash Bank 1
                Width : 32
                Size : 65536
```

- Pin Map Icon을 클릭하면 Pinmapper Tab이 생기면서 Pin Map 확인이 가능하다.
- Pin을 클릭하면 PIN 설정을 control 할 수 있다.

ACE Lab.

찾기





Find/Replace

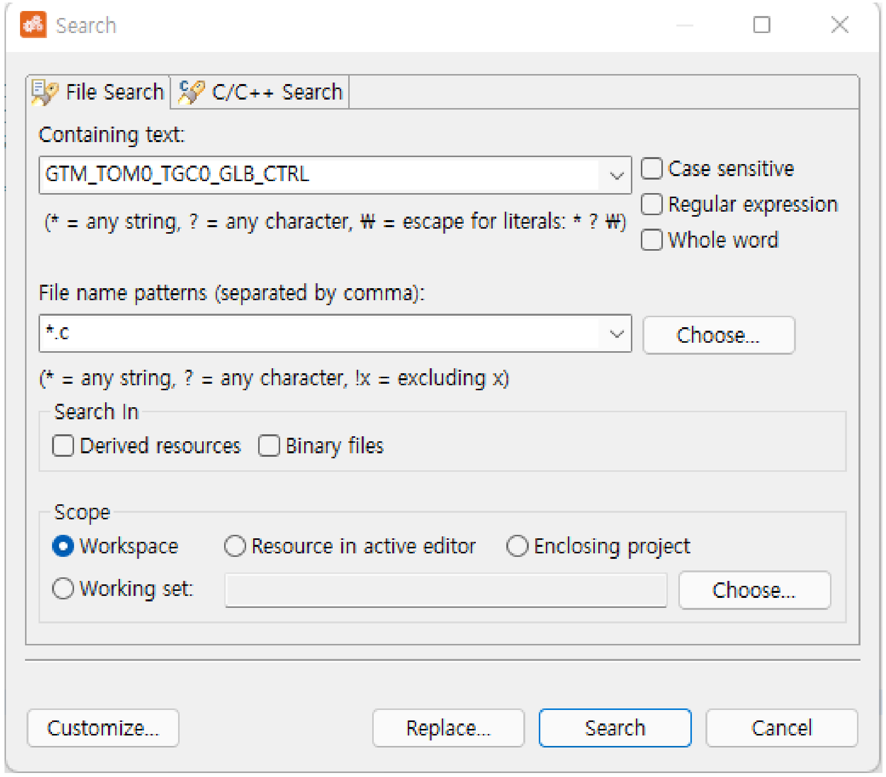
Find: Q4QINR0

Replace with:

Direction: [✓ Forward] Backward

Scope: [✓ All] Selected lines

Options: Case sensitive [✓ Wrap search] Whole word Incremental Regular expressions



Search

File Search | C/C++ Search

Containing text: GTM_TOM0_TGC0_GLB_CTRL

(* = any string, ? = any character, ₩ = escape for literals: * ? ₩)

Case sensitive Regular expression Whole word

File name patterns (separated by comma): *.c

(* = any string, ? = any character, !x = excluding x)

Search In: Derived resources Binary files

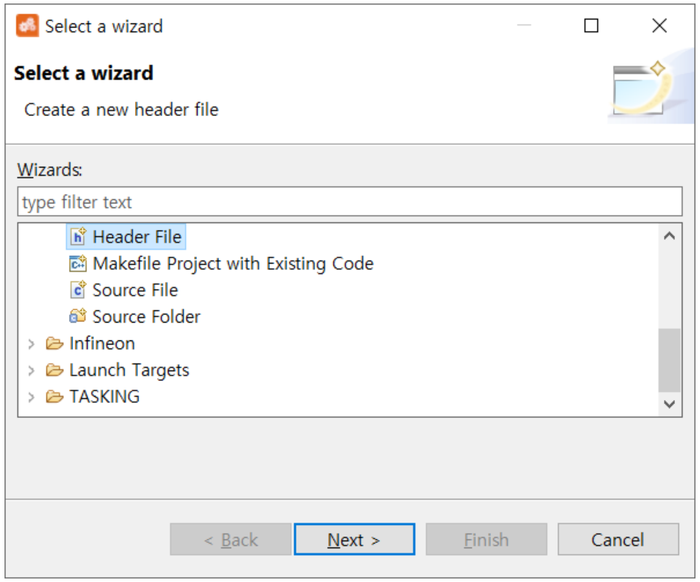
Scope: [✓ Workspace] Resource in active editor Enclosing project Working set:

- Find(Ctrl + F)를 통한 검색 : 현재 편집중인 파일내 검색
- Search를 통한 검색 : file들 간의 검색

Header File, C File 추가

File → New → Other

my_lib.h, my_lib.c 추가


Select a wizard

Select a wizard

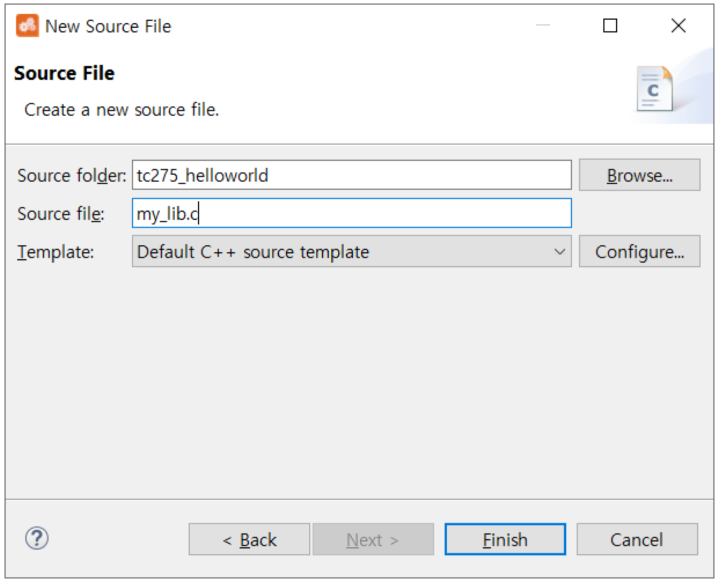
Create a new header file

Wizards:

type filter text

- Header File
- Makefile Project with Existing Code
- Source File
- Source Folder
- Infineon
- Launch Targets
- TASKING

< Back | Next > | Finish | Cancel


New Source File

Source File

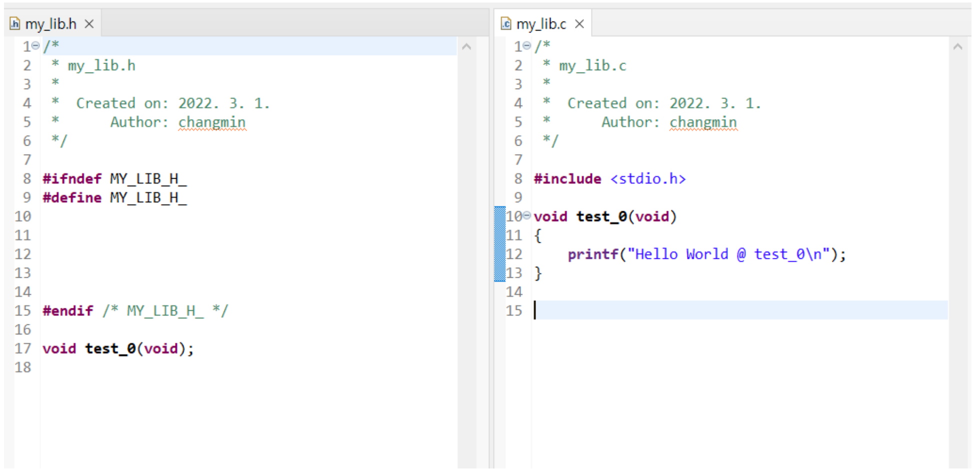
Create a new source file.

Source folder: tc275_helloworld [Browse...]

Source file: my_lib.c

Template: Default C++ source template [Configure...]

< Back | Next > | Finish | Cancel

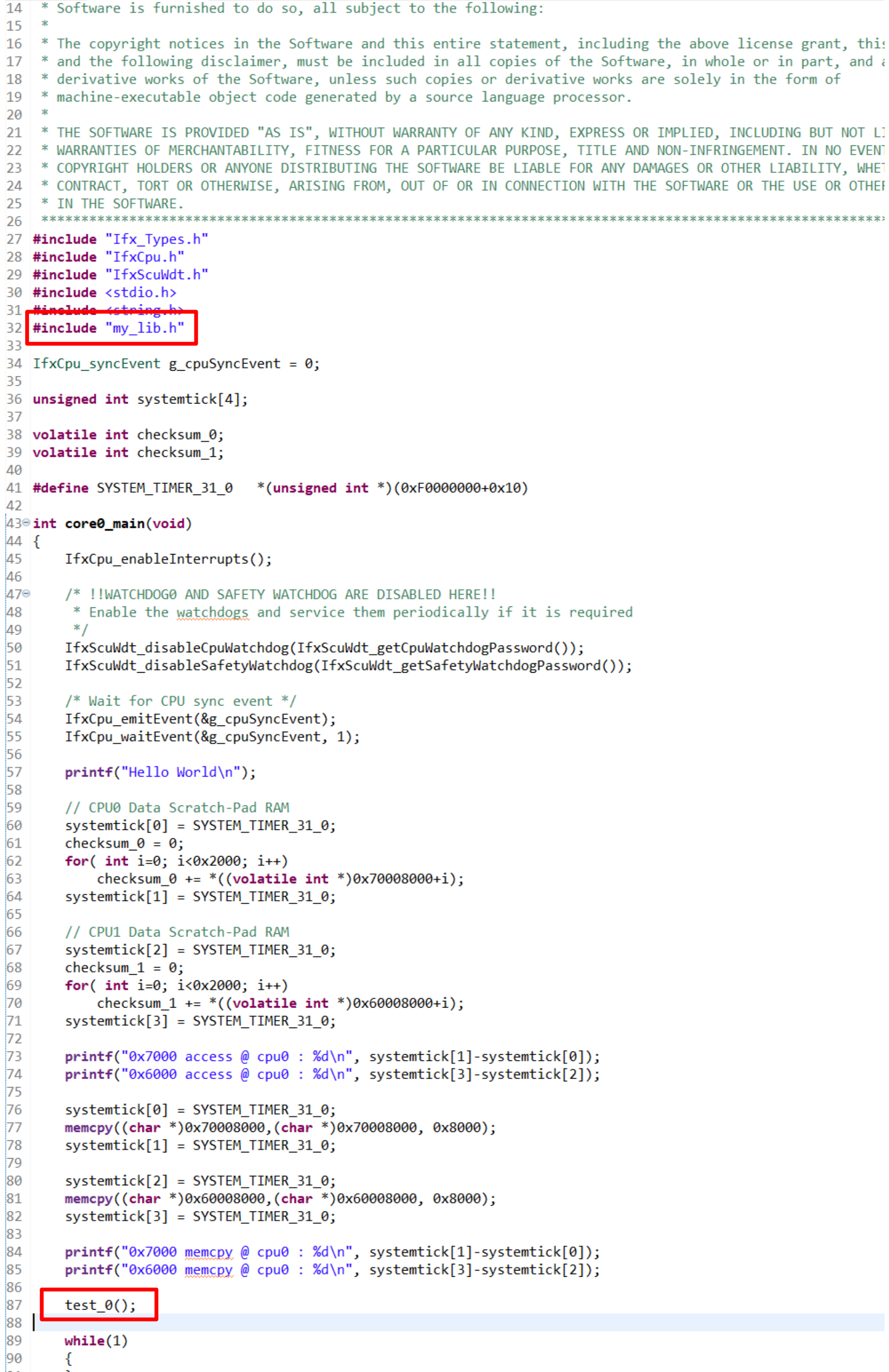

my_lib.h

```
/*
 * my_lib.h
 *
 *  Created on: 2022. 3. 1.
 *      Author: changmin
 */

#ifndef MY_LIB_H_
#define MY_LIB_H_




#endif /* MY_LIB_H_ */

void test_0(void);

```

my_lib.c

```
/*
 * my_lib.c
 *
 *  Created on: 2022. 3. 1.
 *      Author: changmin
 */

#include <stdio.h>

void test_0(void)
{
    printf("Hello World @ test_0\n");
}

```

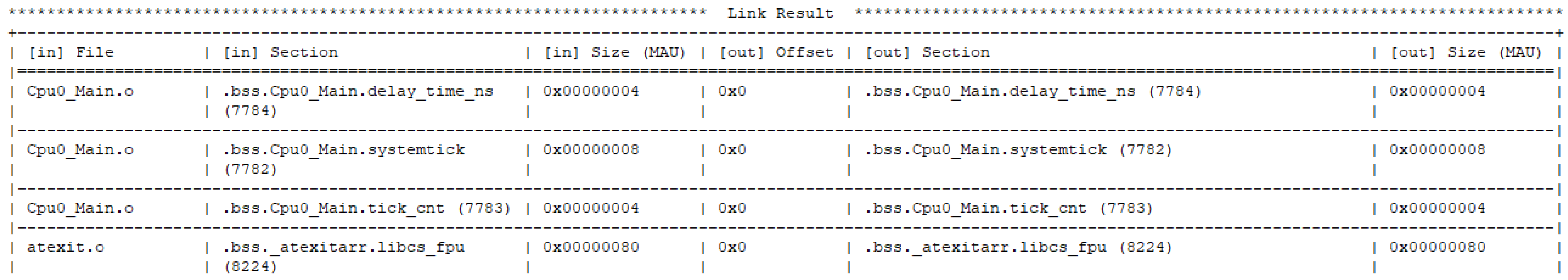
ACE Lab.



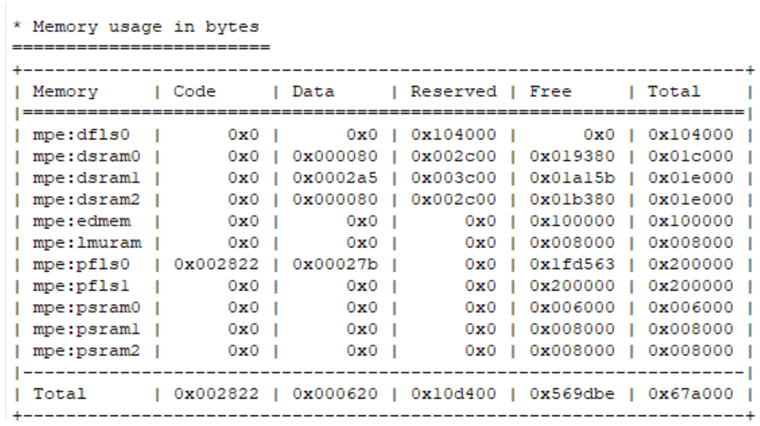
```c
 * Software is furnished to do so, all subject to the following:
 *
 * The copyright notices in the Software and this entire statement, including the above license grant, thi
 * and the following disclaimer, must be included in all copies of the Software, in whole or in part, and
 * derivative works of the Software, unless such copies or derivative works are solely in the form of
 * machine-executable object code generated by a source language processor.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT L
 * WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVEN
 * COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHE
 * CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHE
 * IN THE SOFTWARE.
 *********************************************************************************************************
#include "Ifx_Types.h"
#include "IfxCpu.h"
#include "IfxScuWdt.h"
#include <stdio.h>
#include <string.h>
#include "my_lib.h"

IfxCpu_syncEvent g_cpuSyncEvent = 0;

unsigned int systemtick[4];

volatile int checksum_0;
volatile int checksum_1;

#define SYSTEM_TIMER_31_0   *(unsigned int *)(0xF0000000+0x10)

int core0_main(void)
{
    IfxCpu_enableInterrupts();

    /* !!WATCHDOG0 AND SAFETY WATCHDOG ARE DISABLED HERE!!
     * Enable the watchdogs and service them periodically if it is required
     */
    IfxScuWdt_disableCpuWatchdog(IfxScuWdt_getCpuWatchdogPassword());
    IfxScuWdt_disableSafetyWatchdog(IfxScuWdt_getSafetyWatchdogPassword());

    /* Wait for CPU sync event */
    IfxCpu_emitEvent(&g_cpuSyncEvent);
    IfxCpu_waitEvent(&g_cpuSyncEvent, 1);

    printf("Hello World\n");

    // CPU0 Data Scratch-Pad RAM
    systemtick[0] = SYSTEM_TIMER_31_0;
    checksum_0 = 0;
    for( int i=0; i<0x2000; i++)
        checksum_0 += *((volatile int *)0x70008000+i);
    systemtick[1] = SYSTEM_TIMER_31_0;

    // CPU1 Data Scratch-Pad RAM
    systemtick[2] = SYSTEM_TIMER_31_0;
    checksum_1 = 0;
    for( int i=0; i<0x2000; i++)
        checksum_1 += *((volatile int *)0x60008000+i);
    systemtick[3] = SYSTEM_TIMER_31_0;

    printf("0x7000 access @ cpu0 : %d\n", systemtick[1]-systemtick[0]);
    printf("0x6000 access @ cpu0 : %d\n", systemtick[3]-systemtick[2]);

    systemtick[0] = SYSTEM_TIMER_31_0;
    memcpy((char *)0x70008000,(char *)0x70008000, 0x8000);
    systemtick[1] = SYSTEM_TIMER_31_0;

    systemtick[2] = SYSTEM_TIMER_31_0;
    memcpy((char *)0x60008000,(char *)0x60008000, 0x8000);
    systemtick[3] = SYSTEM_TIMER_31_0;

    printf("0x7000 memcpy @ cpu0 : %d\n", systemtick[1]-systemtick[0]);
    printf("0x6000 memcpy @ cpu0 : %d\n", systemtick[3]-systemtick[2]);

    test_0();

    while(1)
    {
```

MAP 파일

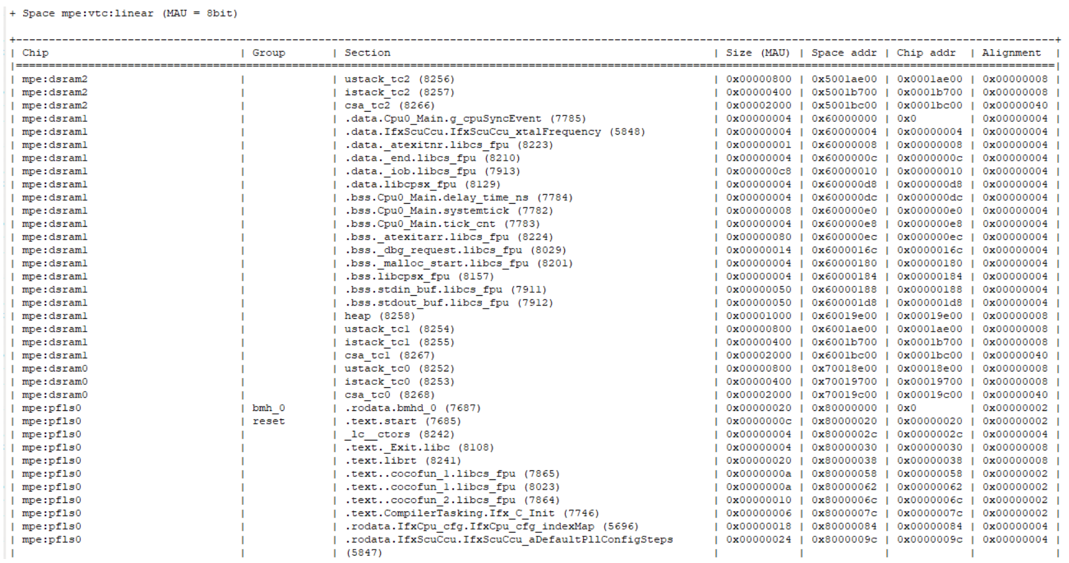
- Map 파일 경로 : Debugger 폴더 아래에 *.map로 존재



```
*******************************************************************  Link Result  *******************************************************************
+-------------------------------------------------------------------------------------------------------------------------------------------------+
| [in] File         | [in] Section                 | [in] Size (MAU) | [out] Offset | [out] Section                                      | [out] Size (MAU) |
|=================================================================================================================================================|
| Cpu0_Main.o       | .bss.Cpu0_Main.delay_time_ns | 0x00000004      | 0x0          | .bss.Cpu0_Main.delay_time_ns (7784)                | 0x00000004       |
|                   | (7784)                       |                 |              |                                                    |                  |
|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Cpu0_Main.o       | .bss.Cpu0_Main.systemtick    | 0x00000008      | 0x0          | .bss.Cpu0_Main.systemtick (7782)                   | 0x00000008       |
|                   | (7782)                       |                 |              |                                                    |                  |
|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Cpu0_Main.o       | .bss.Cpu0_Main.tick_cnt (7783) | 0x00000004    | 0x0          | .bss.Cpu0_Main.tick_cnt (7783)                     | 0x00000004       |
|-------------------------------------------------------------------------------------------------------------------------------------------------|
| atexit.o          | .bss._atexitarr.libcs_fpu    | 0x00000080      | 0x0          | .bss._atexitarr.libcs_fpu (8224)                   | 0x00000080       |
|                   | (8224)                       |                 |              |                                                    |                  |
```



```
* Memory usage in bytes
========================
+--------------------------------------------------------------------+
| Memory      | Code     | Data     | Reserved | Free     | Total    |
|====================================================================|
| mpe:dfls0   |      0x0 |      0x0 | 0x104000 |      0x0 | 0x104000 |
| mpe:dsram0  |      0x0 | 0x000080 | 0x002c00 | 0x019380 | 0x01c000 |
| mpe:dsram1  |      0x0 | 0x0002a5 | 0x003c00 | 0x01a15b | 0x01e000 |
| mpe:dsram2  |      0x0 | 0x000080 | 0x002c00 | 0x01b380 | 0x01e000 |
| mpe:edmem   |      0x0 |      0x0 |      0x0 | 0x100000 | 0x100000 |
| mpe:lmuram  |      0x0 |      0x0 |      0x0 | 0x008000 | 0x008000 |
| mpe:pfls0   | 0x002822 | 0x00027b |      0x0 | 0x1fd563 | 0x200000 |
| mpe:pfls1   |      0x0 |      0x0 |      0x0 | 0x200000 | 0x200000 |
| mpe:psram0  |      0x0 |      0x0 |      0x0 | 0x006000 | 0x006000 |
| mpe:psram1  |      0x0 |      0x0 |      0x0 | 0x008000 | 0x008000 |
| mpe:psram2  |      0x0 |      0x0 |      0x0 | 0x008000 | 0x008000 |
|--------------------------------------------------------------------|
| Total       | 0x002822 | 0x000620 | 0x10d400 | 0x569dbe | 0x67a000 |
+--------------------------------------------------------------------+
```



```
+ Space mpe:vtc:linear (MAU = 8bit)

+--------------------------------------------------------------------------------------------------------------------------------------------------+
| Chip            | Group   | Section                                              | Size (MAU) | Space addr | Chip addr  | Alignment  |
|==================================================================================================================================================|
| mpe:dsram2      |         | ustack_tc2 (8256)                                    | 0x00000800 | 0x5001ae00 | 0x0001ae00 | 0x00000008 |
| mpe:dsram2      |         | istack_tc2 (8257)                                    | 0x00000400 | 0x5001b700 | 0x0001b700 | 0x00000008 |
| mpe:dsram2      |         | csa_tc2 (8266)                                       | 0x00002000 | 0x5001bc00 | 0x0001bc00 | 0x00000040 |
| mpe:dsram1      |         | .data.Cpu0_Main.g_cpuSyncEvent (7785)                | 0x00000004 | 0x60000000 | 0x0        | 0x00000004 |
| mpe:dsram1      |         | .data.IfxScuCcu.IfxScuCcu_xtalFrequency (5848)       | 0x00000004 | 0x60000004 | 0x00000004 | 0x00000004 |
| mpe:dsram1      |         | .data._atexitnr.libcs_fpu (8223)                     | 0x00000001 | 0x60000008 | 0x00000008 | 0x00000004 |
| mpe:dsram1      |         | .data._end.libcs_fpu (8210)                          | 0x00000004 | 0x6000000c | 0x0000000c | 0x00000004 |
| mpe:dsram1      |         | .data._iob.libcs_fpu (7913)                          | 0x000000c8 | 0x60000010 | 0x00000010 | 0x00000004 |
| mpe:dsram1      |         | .data.libcpsx_fpu (8129)                             | 0x00000004 | 0x600000d8 | 0x000000d8 | 0x00000004 |
| mpe:dsram1      |         | .bss.Cpu0_Main.delay_time_ns (7784)                  | 0x00000004 | 0x600000dc | 0x000000dc | 0x00000004 |
| mpe:dsram1      |         | .bss.Cpu0_Main.systemtick (7782)                     | 0x00000008 | 0x600000e0 | 0x000000e0 | 0x00000004 |
| mpe:dsram1      |         | .bss.Cpu0_Main.tick_cnt (7783)                       | 0x00000004 | 0x600000e8 | 0x000000e8 | 0x00000004 |
| mpe:dsram1      |         | .bss._atexitarr.libcs_fpu (8224)                     | 0x00000080 | 0x600000ec | 0x000000ec | 0x00000004 |
| mpe:dsram1      |         | .bss._dbg_request.libcs_fpu (8029)                   | 0x00000014 | 0x6000016c | 0x0000016c | 0x00000004 |
| mpe:dsram1      |         | .bss._malloc_start.libcs_fpu (8201)                  | 0x00000004 | 0x60000180 | 0x00000180 | 0x00000004 |
| mpe:dsram1      |         | .bss.libcpsx_fpu (8157)                              | 0x00000004 | 0x60000184 | 0x00000184 | 0x00000004 |
| mpe:dsram1      |         | .bss.stdin_buf.libcs_fpu (7911)                      | 0x00000050 | 0x60000188 | 0x00000188 | 0x00000004 |
| mpe:dsram1      |         | .bss.stdout_buf.libcs_fpu (7912)                     | 0x00000050 | 0x600001d8 | 0x000001d8 | 0x00000004 |
| mpe:dsram1      |         | heap (8258)                                          | 0x00001000 | 0x60019e00 | 0x00019e00 | 0x00000008 |
| mpe:dsram1      |         | ustack_tc1 (8254)                                    | 0x00000800 | 0x6001ae00 | 0x0001ae00 | 0x00000008 |
| mpe:dsram1      |         | istack_tc1 (8255)                                    | 0x00000400 | 0x6001b700 | 0x0001b700 | 0x00000008 |
| mpe:dsram1      |         | csa_tc1 (8267)                                       | 0x00002000 | 0x6001bc00 | 0x0001bc00 | 0x00000040 |
| mpe:dsram0      |         | ustack_tc0 (8252)                                    | 0x00000800 | 0x70018e00 | 0x00018e00 | 0x00000008 |
| mpe:dsram0      |         | istack_tc0 (8253)                                    | 0x00000400 | 0x70019700 | 0x00019700 | 0x00000008 |
| mpe:dsram0      |         | csa_tc0 (8268)                                       | 0x00002000 | 0x70019c00 | 0x00019c00 | 0x00000040 |
| mpe:pfls0       | bmh_0   | .rodata.bmhd_0 (7687)                                | 0x00000020 | 0x80000000 | 0x0        | 0x00000002 |
| mpe:pfls0       | reset   | .text.start (7685)                                   | 0x0000000c | 0x80000020 | 0x00000020 | 0x00000002 |
| mpe:pfls0       |         | _lc__ctors (8242)                                    | 0x00000004 | 0x8000002c | 0x0000002c | 0x00000004 |
| mpe:pfls0       |         | .text._Exit.libc (8108)                              | 0x00000004 | 0x80000030 | 0x00000030 | 0x00000008 |
| mpe:pfls0       |         | .text.librt (8241)                                   | 0x00000020 | 0x80000038 | 0x00000038 | 0x00000008 |
| mpe:pfls0       |         | .text..cocofun_1.libcs_fpu (7865)                    | 0x0000000a | 0x80000058 | 0x00000058 | 0x00000002 |
| mpe:pfls0       |         | .text..cocofun_1.libcs_fpu (8023)                    | 0x0000000a | 0x80000062 | 0x00000062 | 0x00000002 |
| mpe:pfls0       |         | .text..cocofun_2.libcs_fpu (7864)                    | 0x00000010 | 0x8000006c | 0x0000006c | 0x00000002 |
| mpe:pfls0       |         | .text.CompilerTasking.Ifx_C_Init (7746)              | 0x00000006 | 0x8000007c | 0x0000007c | 0x00000002 |
| mpe:pfls0       |         | .rodata.IfxCpu_cfg.IfxCpu_cfg_indexMap (5696)        | 0x00000018 | 0x80000084 | 0x00000084 | 0x00000004 |
| mpe:pfls0       |         | .rodata.IfxScuCcu.IfxScuCcu_aDefaultPllConfigSteps   | 0x00000024 | 0x8000009c | 0x0000009c | 0x00000004 |
|                 |         | (5847)                                               |            |            |            |            |
```

Q & A

Thank you for your attention

Architecture and
Compiler
for Embedded Systems Lab.

School of Electronics Engineering, KNU

ACE Lab (hn02301@gmail.com)