

Infineon TC275 Getting Started

Hyeongrae Kim

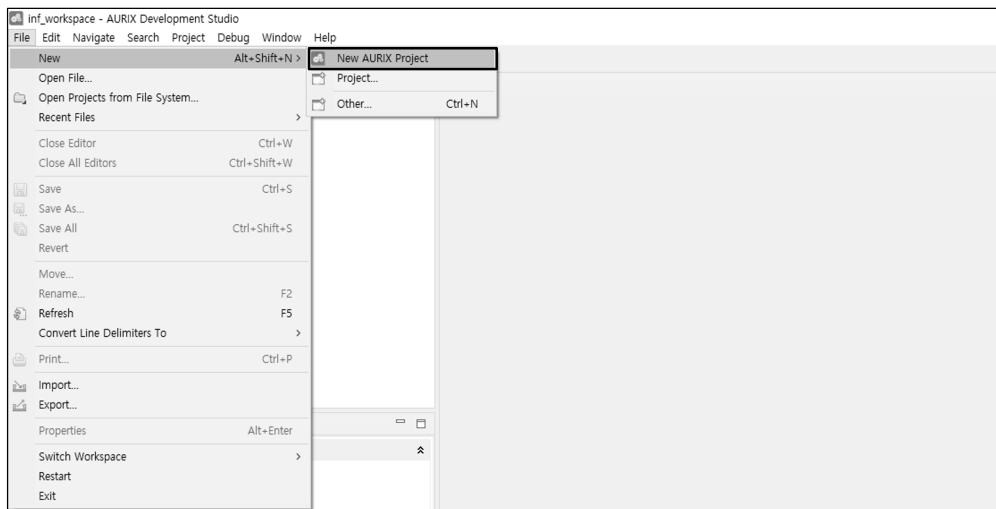
Architecture and Compiler for Embedded system LAB.

School of Electronics Engineering, KNU, KOREA



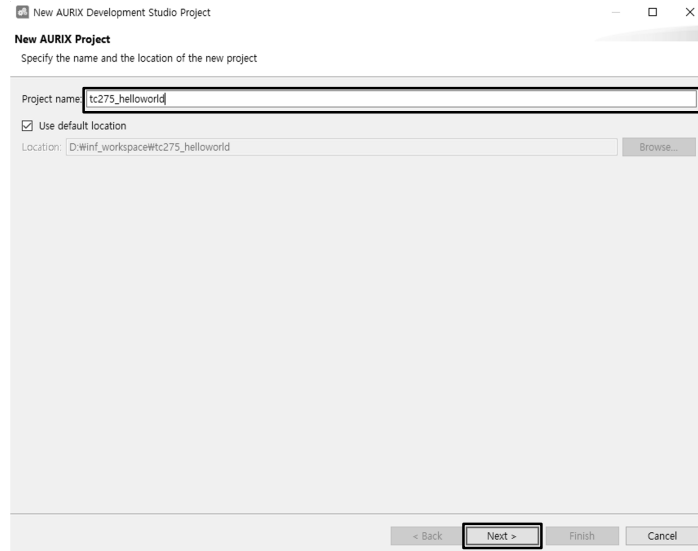
New Project

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



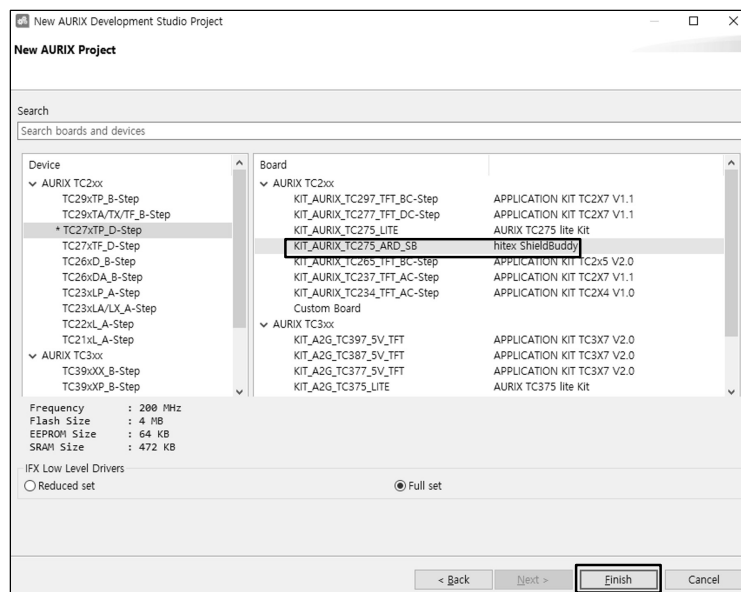
New Project

2. Project name을 입력하고, 'Next'를 클릭한다.



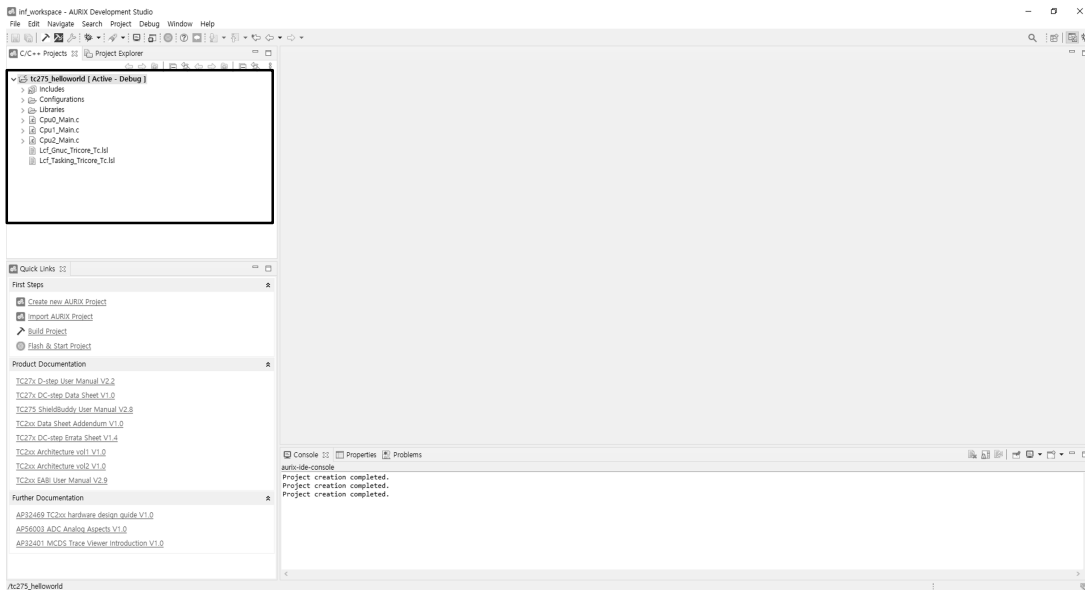
New Project

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



New Project

4. 왼쪽의 Project Explorer 창에서 프로젝트가 생성된 것을 확인한다.




Edit Project

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



Edit Project

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

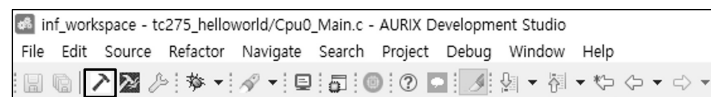


```
@ Cpu0_Main.c 33
12 * accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute,
13 * and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the
14 * Software is furnished to do so, all subject to the following:
15 *
16 * The copyright notices in the Software and this entire statement, including the above license grant, this restriction
17 * and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all
18 * derivative works of the Software, unless such copies or derivative works are solely in the form of
19 * machine-executable object code generated by a source language processor.
20 *
21 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
22 * WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE
23 * COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN
24 * CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
25 * IN THE SOFTWARE.
26 *****/
27 #include "IfxTypes.h"
28 #include "IfxCpu.h"
29 #include "IfxScuidt.h"
30 #include <stdio.h>
31
32 IfxCpu_syncEvent g_cpuSyncEvent = 0;
33
34 int core0_main(void)
35 {
36     IfxCpu_enableInterrupts();
37
38     /* !!WATCHDOG0 AND SAFETY WATCHDOG ARE DISABLED HERE!!
39      * Enable the watchdogs and service them periodically if it is required
40      */
41     IfxScuidt_disableCpuWatchdog(IfxScuidt_getCpuWatchdogPassword());
42     IfxScuidt_disableSafetyWatchdog(IfxScuidt_getSafetyWatchdogPassword());
43
44     /* Wait for CPU sync event */
45     IfxCpu_emitEvent(&g_cpuSyncEvent);
46     IfxCpu_waitEvent(&g_cpuSyncEvent, 1);
47     printf("Hello World!\n");
48
49     while(1)
50     {
51     }
52     return (1);
53 }
54
55
```

7/14

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.)

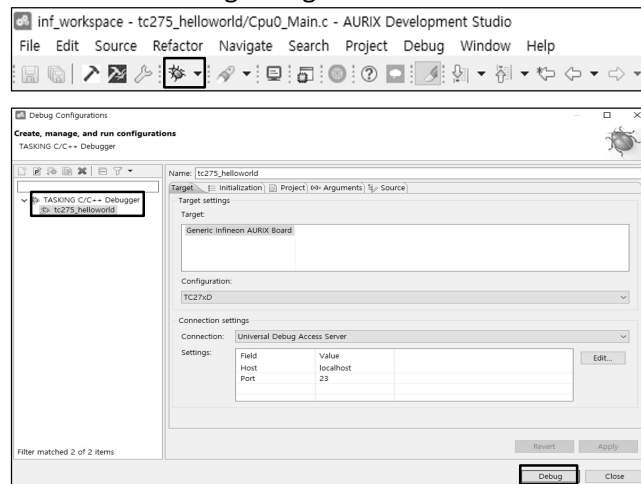


Debug

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.

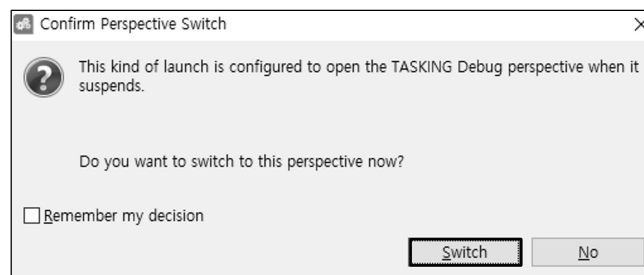
(이후에는 Debug가 바로 실행되며 'Debug Configurations' 창을 활성화하기 위해서는 'Debug' 버튼의 오른쪽 화살표를 클릭한 후 'Debug Configurations...' 버튼을 클릭한다.)



Debug

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

- ✓ 'Confirm Perspective Switch' 창이 뜨면 Switch를 눌러 디버그 창으로 전환한다.

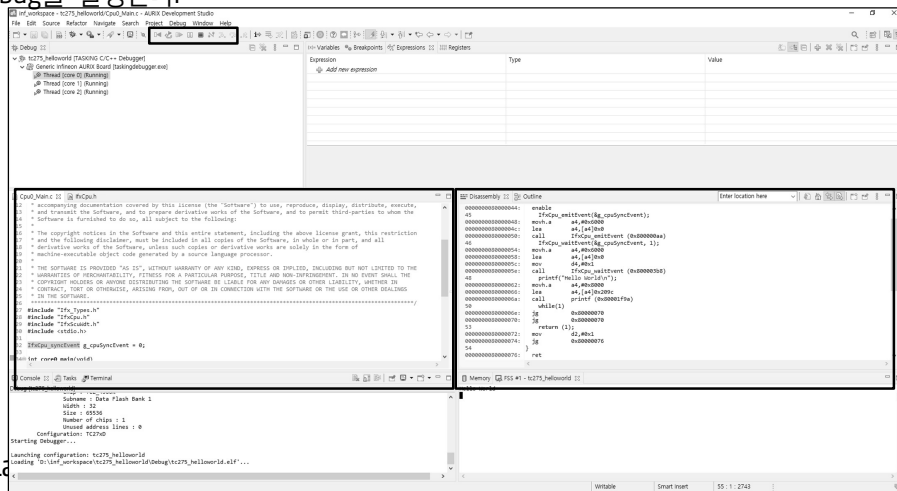


Debug

9. Debug 화면을 확인한다.

- ✓ 상단의 Restart / Resume / Terminate / Step Into / Step Over 등을 클릭하여 실행을 제어할 수 있다.
- ✓ 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.

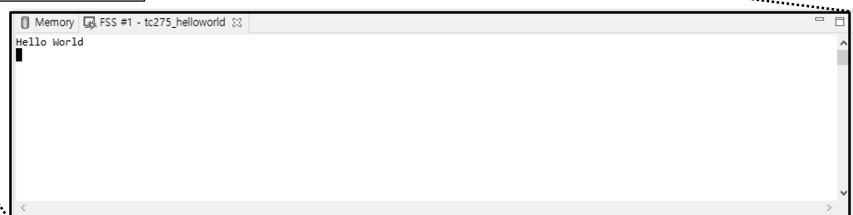
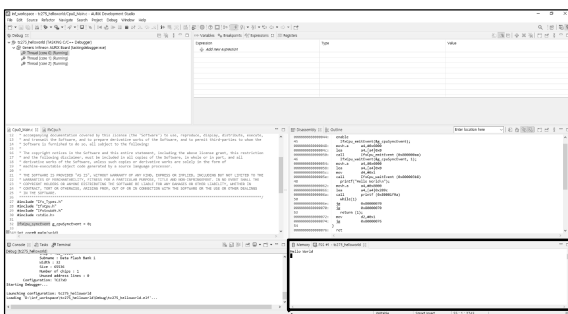


11/14

Debug

10. Debug를 통해 실행을 제어하고 결과를 확인한다.

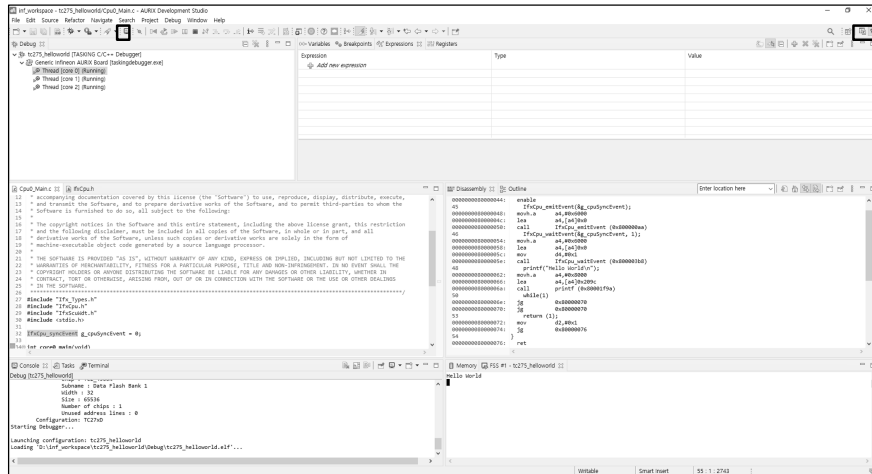
- ✓ 상단의 'Resume' 버튼을 클릭하여 동작을 실행시킨다.
- ✓ 동작의 실행 결과로 우측 하단 창 (FSS)에 'Hello world'라는 메시지가 표시되는 것을 확인한다.



Debug

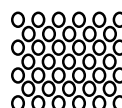
11. Debug를 종료한다.

- ✓ 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.



Q & A

Thank you for your attention

 Architecture and
Compiler
for Embedded Systems Lab.

School of Electronics Engineering, KNU
ACE Lab (hn02301@gmail.com)