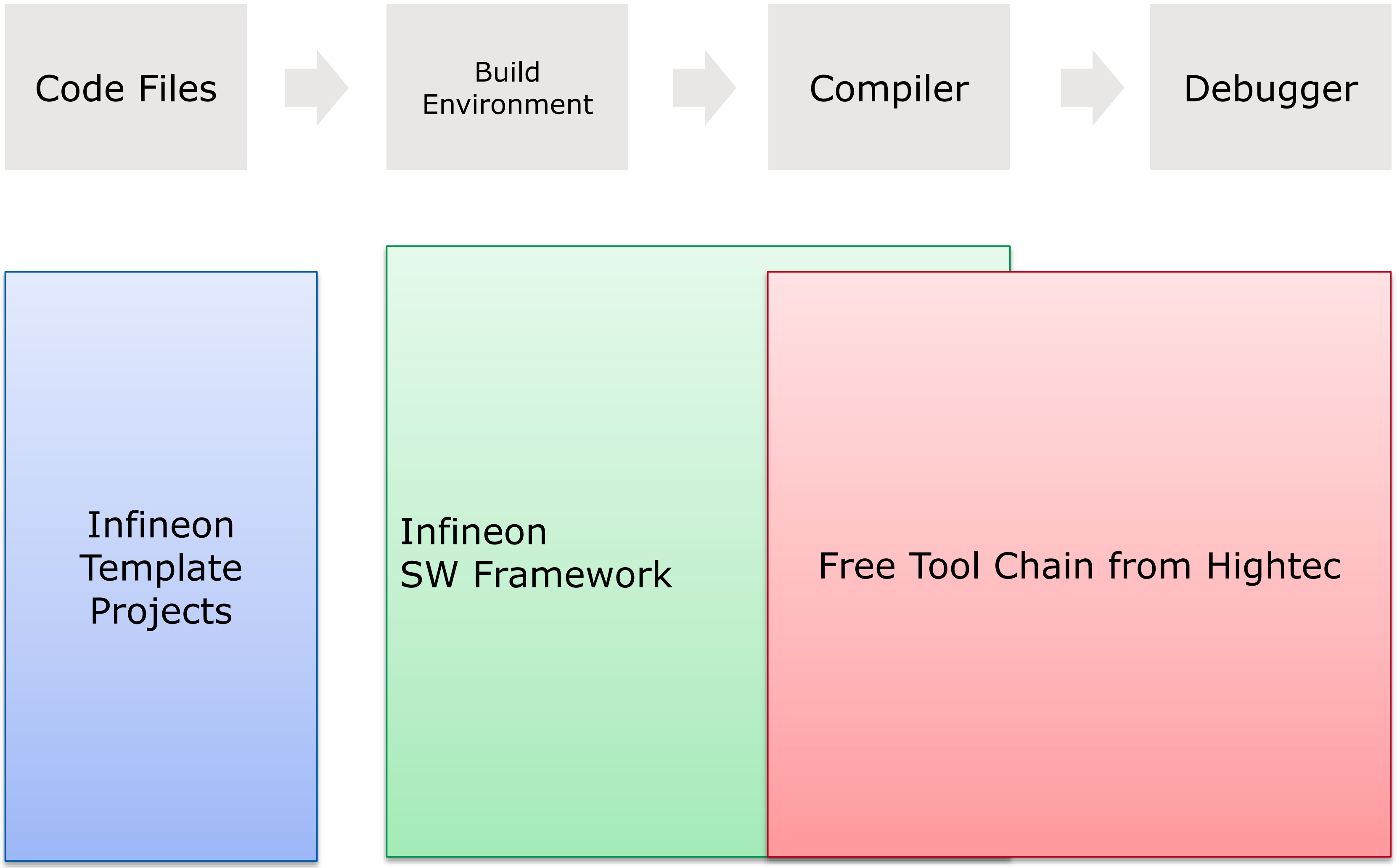
# AURIX 2G Hands-on – First Blinky

IFCN ATV SMD GC SAE MC

## Tool Overview for Handson

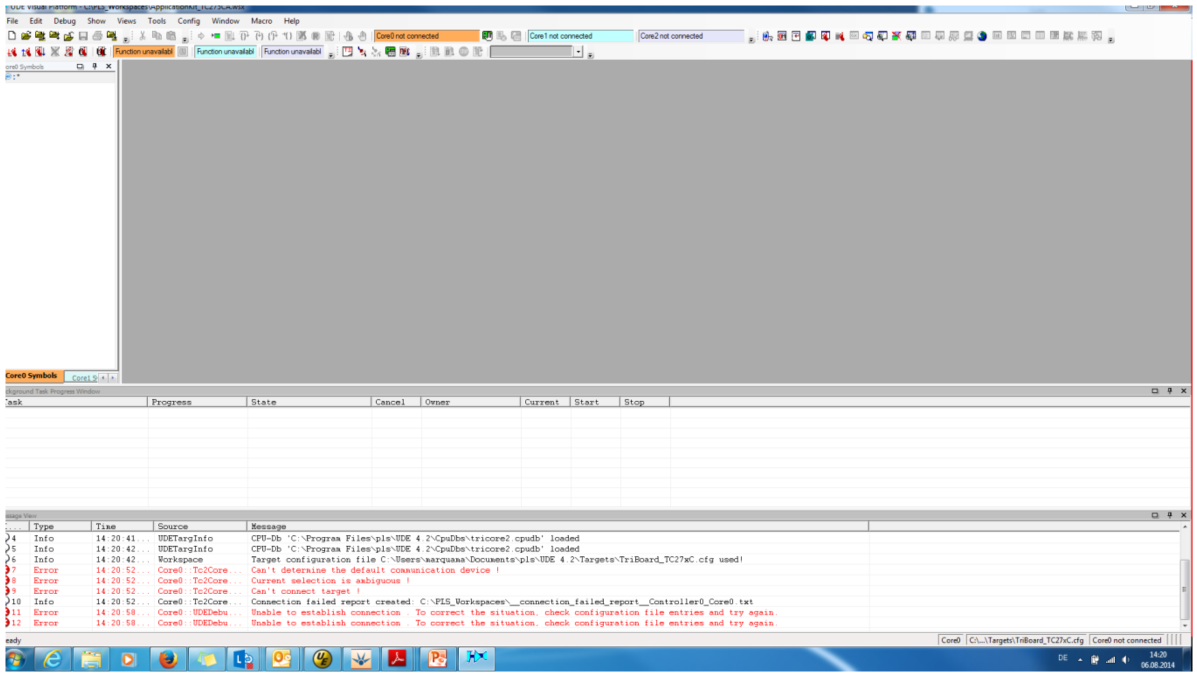


# Hightec Free Entry Toolchain

- GNU Compiler 4.9.1.0
- PLS Debugger (UDE Desktop)
- Hightec Eclipse IDE
- Example Project

Free TriCore™ Entry Tool Chain
The easiest way to start with a TriBoard.
Version 4.6.2.1 (with TC275C Support)

- We will use
  - Debugger from PLS
  - GNU compiler


File Edit Debug Show Views Tools Config Window Macro Help
Core0 not connected
Core1 not connected
Core2 not connected
Function unavailabl
Core0 Symbols
Task Progress State Cancel Owner Current Start Stop

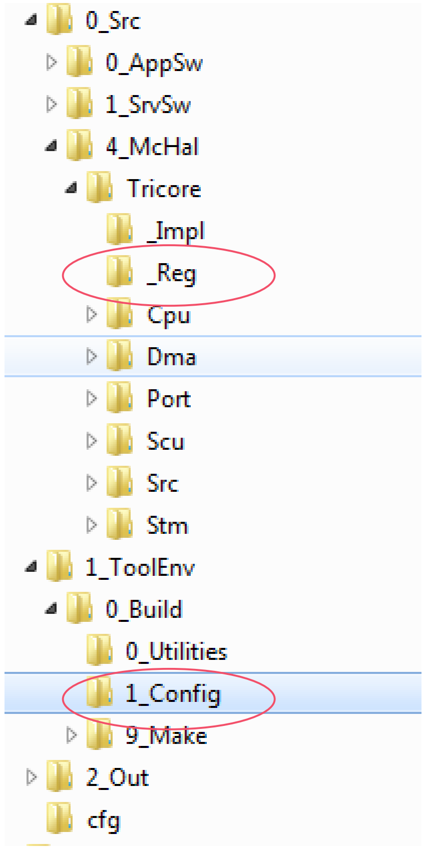
```
Type   Time         Source           Message
4  Info   14:20:41...  UDETargInfo      CPU-Db 'C:\Program Files\pls\UDE 4.2\CpuDbs\tricore2.cpudb' loaded
5  Info   14:20:42...  UDETargInfo      CPU-Db 'C:\Program Files\pls\UDE 4.2\CpuDbs\tricore2.cpudb' loaded
6  Info   14:20:42...  Workspace        Target configuration file C:\Users\marquama\Documents\pls\UDE 4.2\Targets\TriBoard_TC27xC.cfg used!
7  Error  14:20:52...  Core0::Tc2Core...  Can't determine the default communication device !
8  Error  14:20:52...  Core0::Tc2Core...  Current selection is ambiguous !
9  Error  14:20:52...  Core0::Tc2Core...  Can't connect target !
10 Info   14:20:52...  Core0::Tc2Core...  Connection failed report created: C:\PLS_Workspaces\__connection_failed_report__Controller0_Core0.txt
11 Error  14:20:58...  Core0::UDEDebu...  Unable to establish connection . To correct the situation, check configuration file entries and try again.
12 Error  14:20:58...  Core0::UDEDebu...  Unable to establish connection . To correct the situation, check configuration file entries and try again.
```

Core0 | C:\...\Targets\TriBoard_TC27xC.cfg | Core0 not connected
14:20
06.08.2014

# Infineon SW Framework

- Build Environment for quick start
- Supports three compilers (Hightec, Tasking, Windriver)
- **Eclipse** or Command Line based compiling

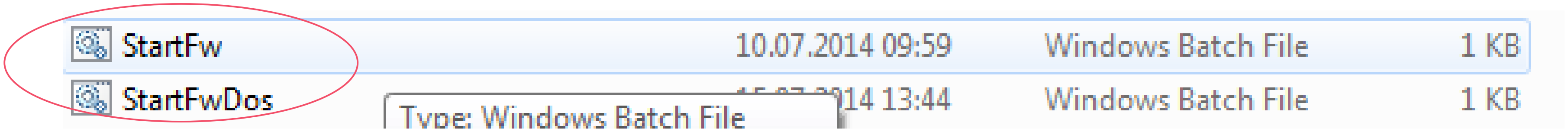
- Provides SW Templates for each derivative
  - Special Function Registers
  - Map Files
  - Startup
  - Basic Drivers
  - Main.c for every core



```
0_Src
  0_AppSw
  1_SrvSw
  4_McHal
    Tricore
      _Impl
      _Reg
      Cpu
      Dma
      Port
      Scu
      Src
      Stm
1_ToolEnv
  0_Build
    0_Utilities
    1_Config
    9_Make
2_Out
cfg
```

# Start Eclipse

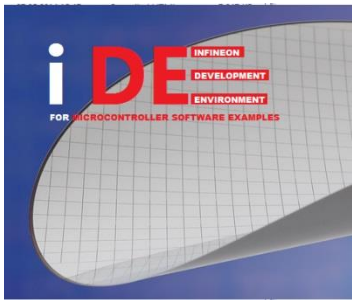
› Start Eclipse with StartFw.bat
  - Otherwise you will see errors while compiling because environment variables are not set correctly



| | | | |
|---|---|---|---|
| StartFw | 10.07.2014 09:59 | Windows Batch File | 1 KB |
| StartFwDos | 15.07.2014 13:44 | Windows Batch File | 1 KB |

Type: Windows Batch File

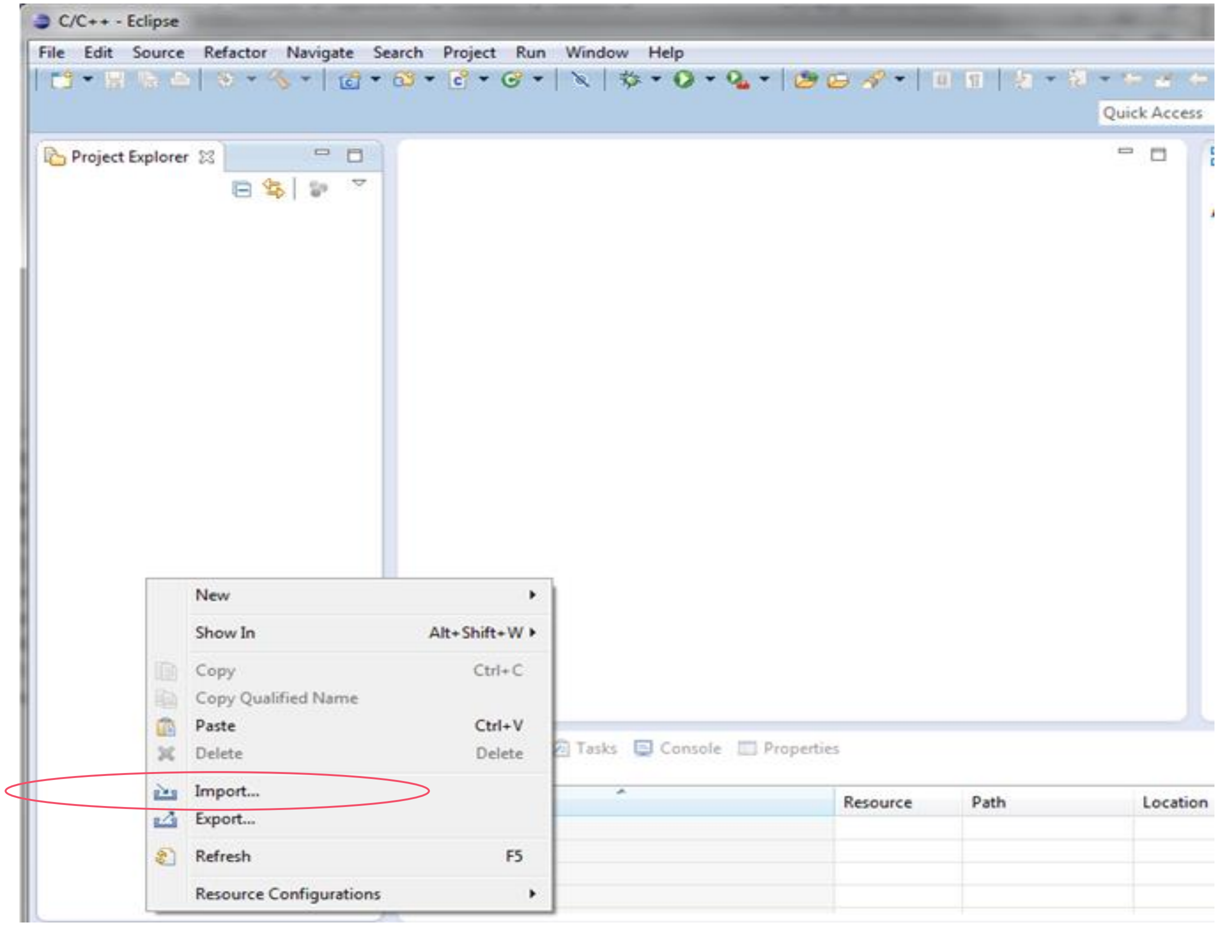
  - It should look like this:



  - If you want to use a different Editor you can also use StartFwDos.bat to open a terminal window
  - Type make and hit enter to compile in that case

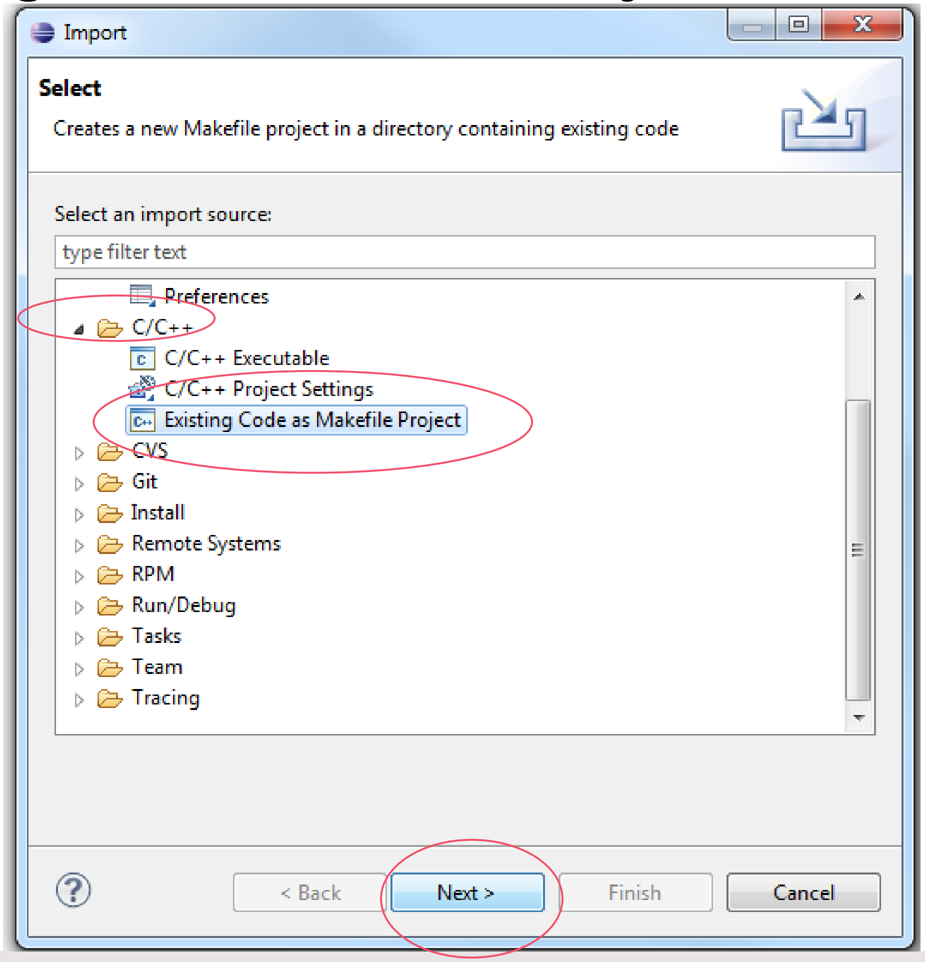
# Add a project

- Right click into the free white space on the left
- Select Import



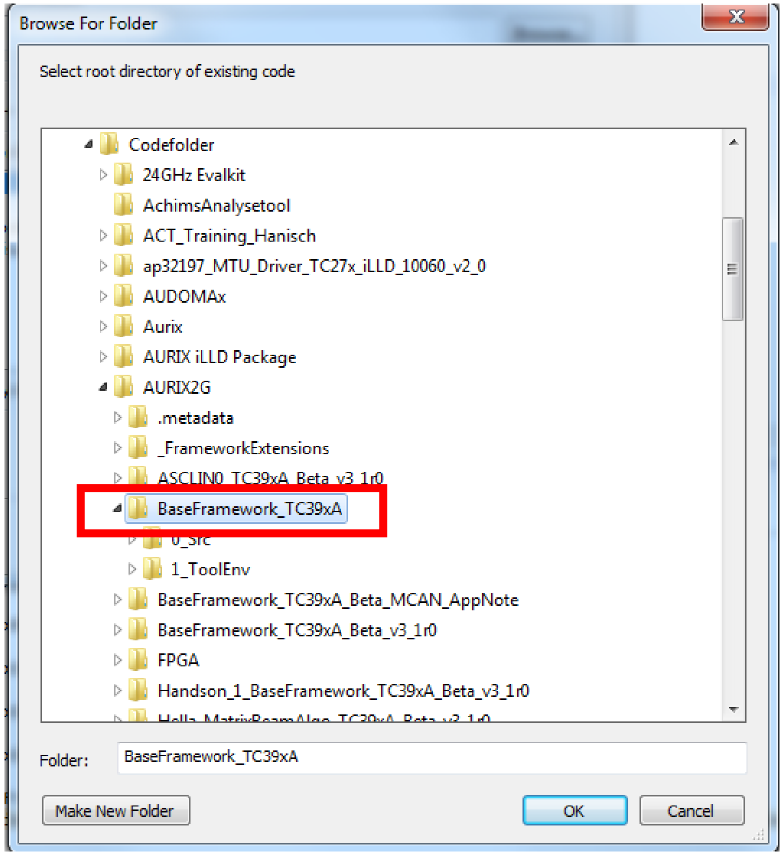
# Add a project

- Select C\C++
- Select Existing Code as Makefile Project
- Next…



# Add a project

- Click „Browse“
- Select the template project for the TC39x (you specified the path to the template projects during the installation).
- OK…

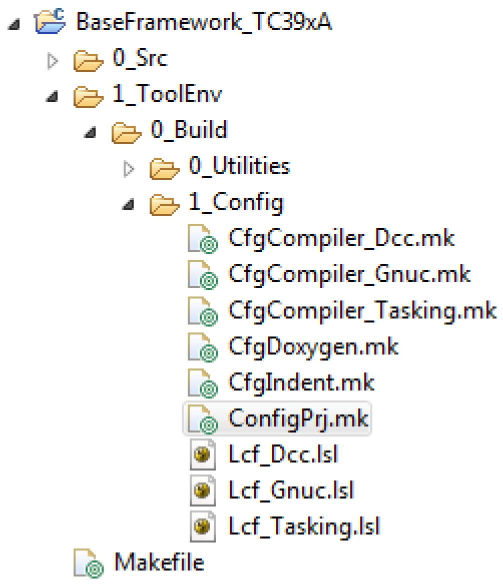


Codefolder
- 24GHz Evalkit
- AchimsAnalysetool
- ACT_Training_Hanisch
- ap32197_MTU_Driver_TC27x_iLLD_10060_v2_0
- AUDOMAx
- Aurix
- AURIX iLLD Package
- AURIX2G
  - .metadata
  - _FrameworkExtensions
  - ASCLIN0_TC39xA_Beta_v3_1r0
  - BaseFramework_TC39xA
    - 0_Src
    - 1_ToolEnv
  - BaseFramework_TC39xA_Beta_MCAN_AppNote
  - BaseFramework_TC39xA_Beta_v3_1r0
  - FPGA
  - Handson_1_BaseFramework_TC39xA_Beta_v3_1r0

Folder: BaseFramework_TC39xA

# Configure the Compile Environment of the Project

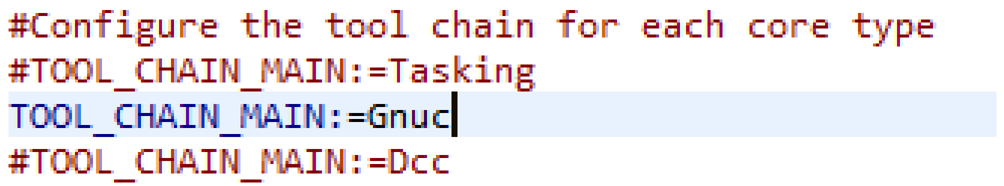
› Open the Configuration folder for the Tool Environment



```
BaseFramework_TC39xA
  0_Src
  1_ToolEnv
    0_Build
      0_Utilities
      1_Config
        CfgCompiler_Dcc.mk
        CfgCompiler_Gnuc.mk
        CfgCompiler_Tasking.mk
        CfgDoxygen.mk
        CfgIndent.mk
        ConfigPrj.mk
        Lcf_Dcc.lsl
        Lcf_Gnuc.lsl
        Lcf_Tasking.lsl
  Makefile
```

› Open ConfigPrj.mk

› Select GNU compiler by commenting/uncommenting the correct lines (GNU is default)

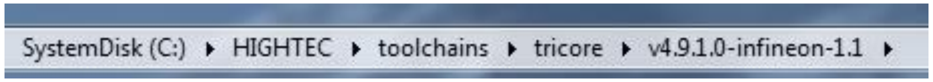


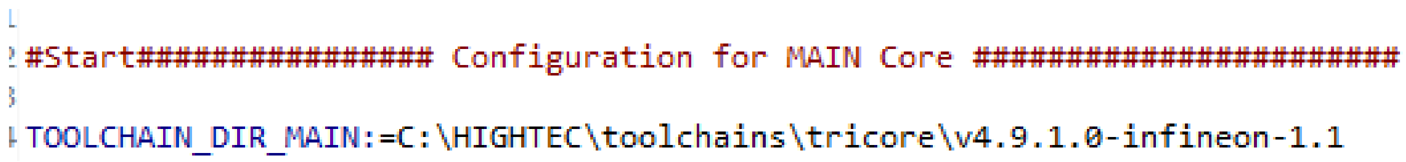
```
#Configure the tool chain for each core type
#TOOL_CHAIN_MAIN:=Tasking
TOOL_CHAIN_MAIN:=Gnuc
#TOOL_CHAIN_MAIN:=Dcc
```

# Configure the Compile Environment of the Project

› Open CfgCompilerGNUC.mk

› Set TOOLCHAIN_DIR_MAIN to the path where you installed the Hightec Free Entry Toolchain


SystemDisk (C:) ▸ HIGHTEC ▸ toolchains ▸ tricore ▸ v4.9.1.0-infineon-1.1 ▸

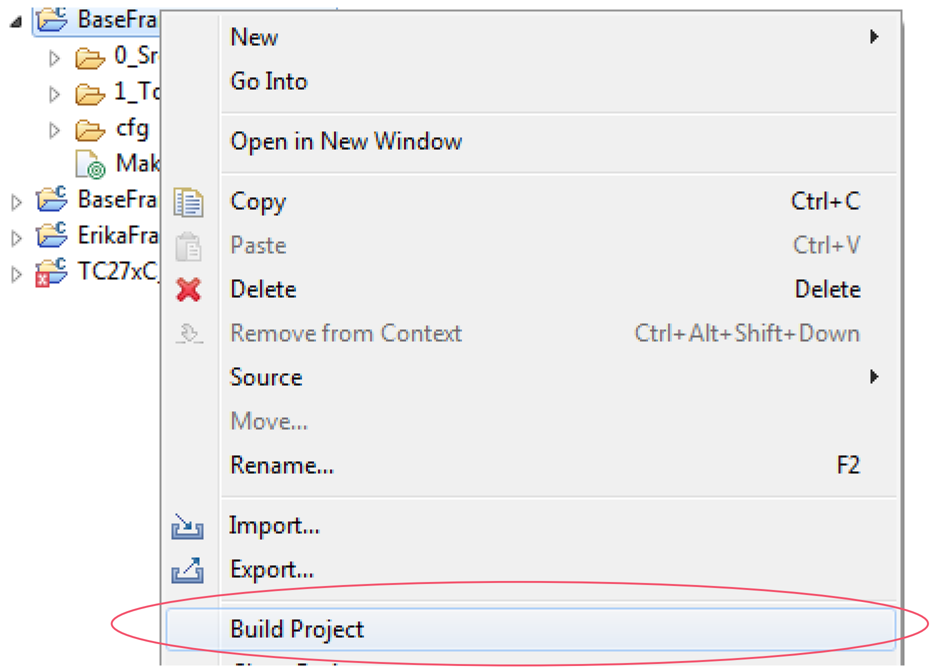


```
#Start################# Configuration for MAIN Core ########################

TOOLCHAIN_DIR_MAIN:=C:\HIGHTEC\toolchains\tricore\v4.9.1.0-infineon-1.1
```

› Save the file

# Compile the Project

- Right-click on the Project Name in the Project Explorer
- Select „Build Project“

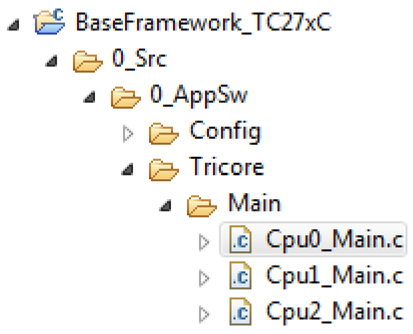


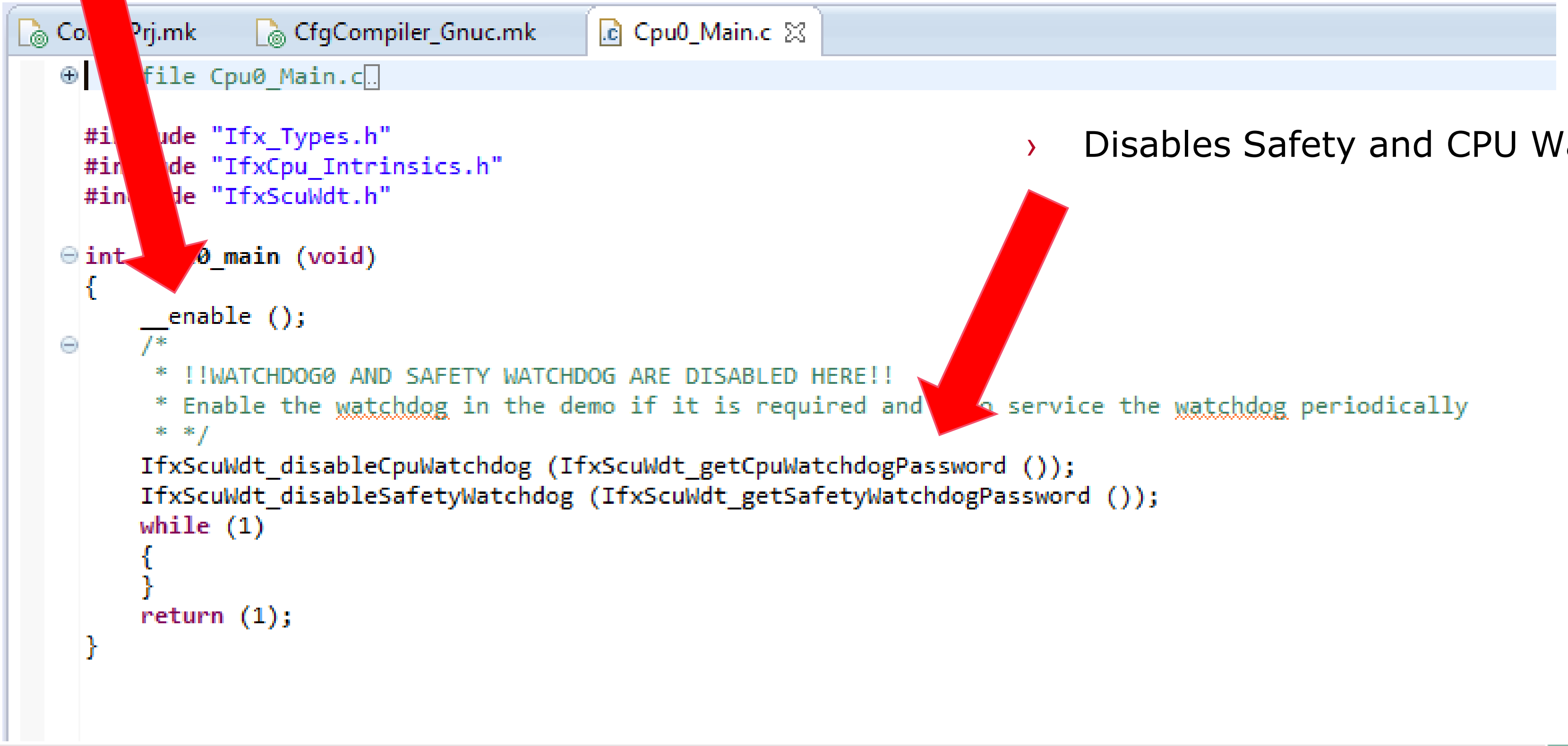
# Hello LED with AURIX 2G

› Open Source File Cpu0_Main.c

›

› Used to enable interrupts for this core





```
    file Cpu0_Main.c

#i    ude "Ifx_Types.h"
#in    de "IfxCpu_Intrinsics.h"
#in    de "IfxScuWdt.h"

int     0_main (void)
{
    __enable ();
    /*
     * !!WATCHDOG0 AND SAFETY WATCHDOG ARE DISABLED HERE!!
     * Enable the watchdog in the demo if it is required and     o service the watchdog periodically
     * */
    IfxScuWdt_disableCpuWatchdog (IfxScuWdt_getCpuWatchdogPassword ());
    IfxScuWdt_disableSafetyWatchdog (IfxScuWdt_getSafetyWatchdogPassword ());
    while (1)
    {
    }
    return (1);
}
```

› Disables Safety and CPU Watchdogs

## CPU and Safety Watchdogs

- Each CPU has one dedicated Watchdog timer
- Additionaly there is a Safety Watchdog
  - Timeout Watchdog
  - Protection of important registers
    - ENDINIT Protection
- Correct access to protected registers (also in case of disabled watchdogs)
  - Clear corresponding ENDINIT Bit
  - Access Register
  - Set corresponding ENDINIT Bit
- **EACH step is MANDATORY / Wrong access will end in Reset**

**Use routines provided by IFX in IfxScuWdt.h**

# Toggle LED – Port Operation

- Input/Output defined by
  - IOCR
- Output
  - Pin state can be modified
    - OMSR
    - OMCR
    - OMR
    - OUT
    - One Internal HW-unit at a time
- Input
  - Input signal can be used by multiple peripherals

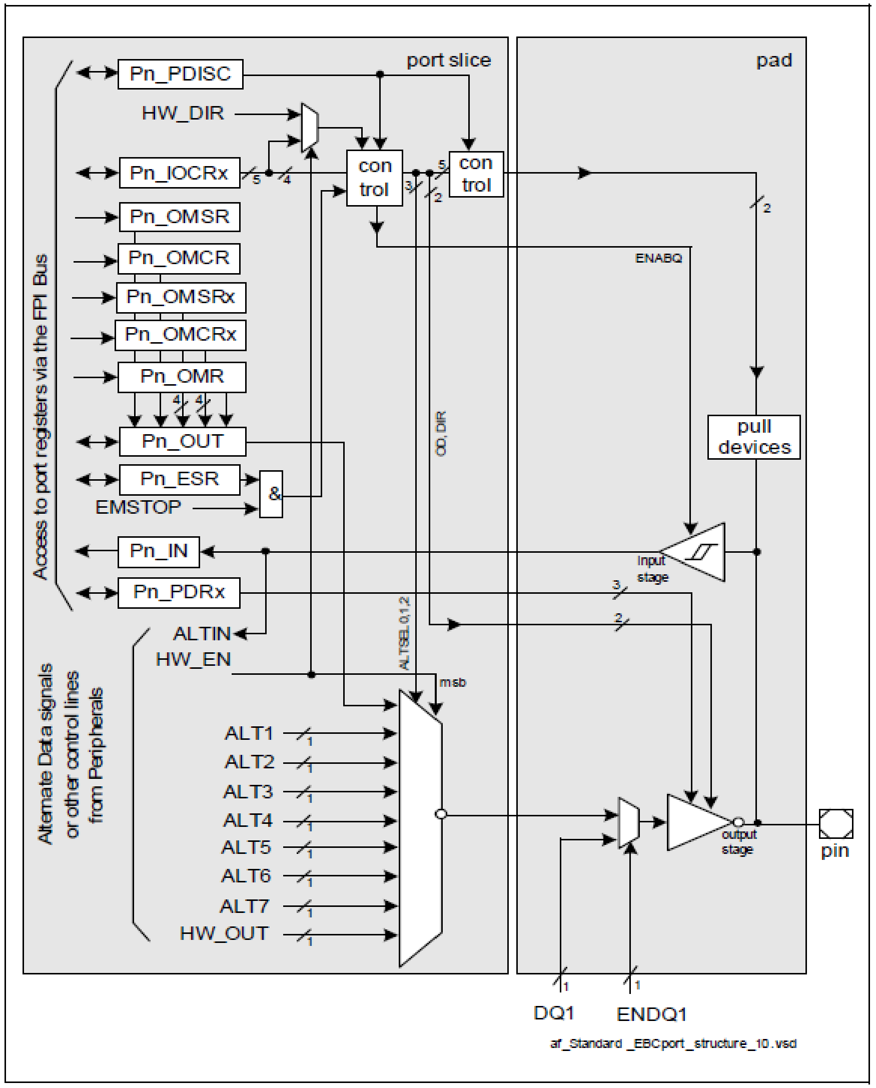


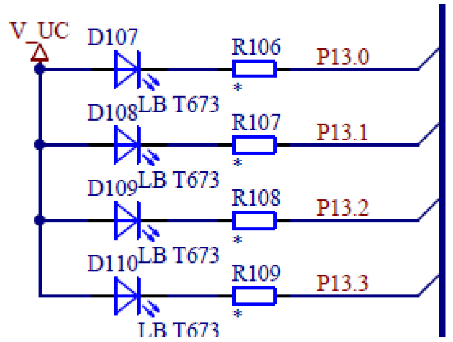
**Figure 14-1 General Structure of a Port Pin**

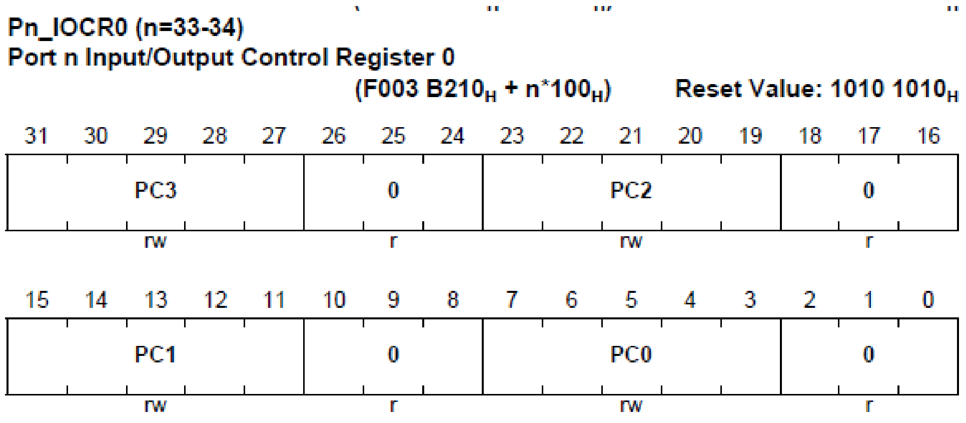
# Toggle LED – What to do

- We select P33.4
- Applicationboard see below
- Configure the pin as an output
  - configure the bit field P33_IOCR4.B.PC4 to 0x10

| | | |
|---|---|---|
| $0XX10_B$ | | Input pull-up device connected[1)] |

| | | | |
|---|---|---|---|
| $10000_B$ | Output | Push-pull | General-purpose output |
| $10001_B$ | | | Alternate output function 1 |
| $10010_B$ | | | Alternate output function 2 |
| $10011_B$ | | | Alternate output function 3 |




**Pn_IOCR0 (n=33-34)**
**Port n Input/Output Control Register 0**
**($F003\ B210_H + n*100_H$)** **Reset Value: $1010\ 1010_H$**

| 31–27 | 26–24 | 23–19 | 18–16 |
|---|---|---|---|
| PC3 | 0 | PC2 | 0 |
| rw | r | rw | r |

| 15–11 | 10–8 | 7–3 | 2–0 |
|---|---|---|---|
| PC1 | 0 | PC0 | 0 |
| rw | r | rw | r |

| Field | Bits | Type | Description |
|---|---|---|---|
| **PC0, PC1, PC2, PC3** | [7:3], [15:11], [23:19], [31:27] | rw | **Port Control for Port n Pin 0 to 3** This bit field determines the Port n line x functionality (x = 0-3) according to the coding table (see **Table 14-5**). |

# Accessing Registers and bit Fields

- Add #include „Ifx_reg.h“
  - Adds all the register definition headers of the Template project

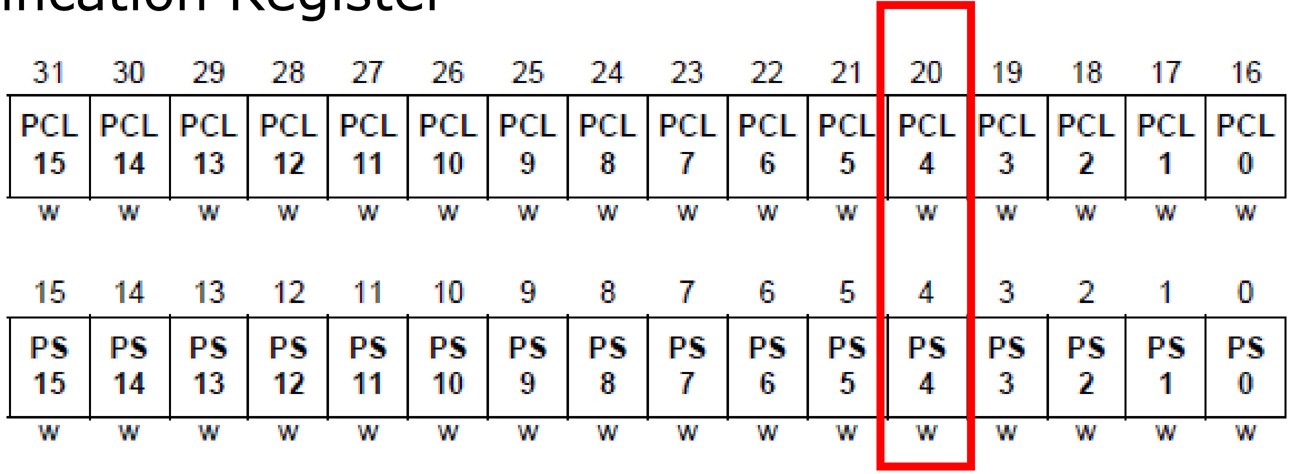
- Each Register is defined with <Modulname>_<Registername> as a Union with 32 bit access (.U) and bit field access (.B.)

- Two lines can have the same effect:

- P33_IOCR4.U = (0x10 << 0); //Configure Pin 4 of Port 33 as output
- P33_IOCR4.B.PC4 = 0x10; //Configure Pin 4 of Port 33 as output

# Toggle LED – What to do

- AURIX provides a simple way to toggle a pin
  - Output Modification Register
  - Reset
  - Set
  - Both -> Toggle

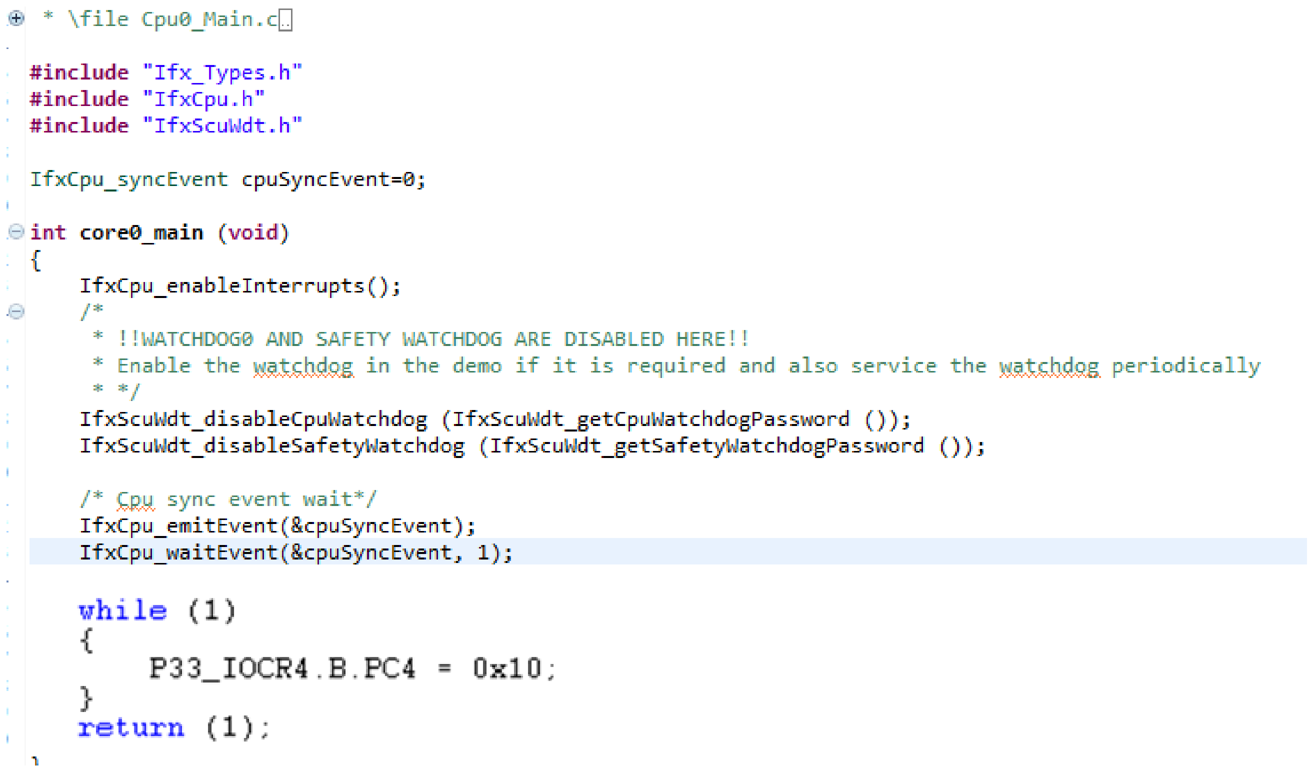


| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PCL 15 | PCL 14 | PCL 13 | PCL 12 | PCL 11 | PCL 10 | PCL 9 | PCL 8 | PCL 7 | PCL 6 | PCL 5 | PCL 4 | PCL 3 | PCL 2 | PCL 1 | PCL 0 |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PS 15 | PS 14 | PS 13 | PS 12 | PS 11 | PS 10 | PS 9 | PS 8 | PS 7 | PS 6 | PS 5 | PS 4 | PS 3 | PS 2 | PS 1 | PS 0 |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

- P33_OMR.U |= (0x1 << 4) | (0x1 << 20);

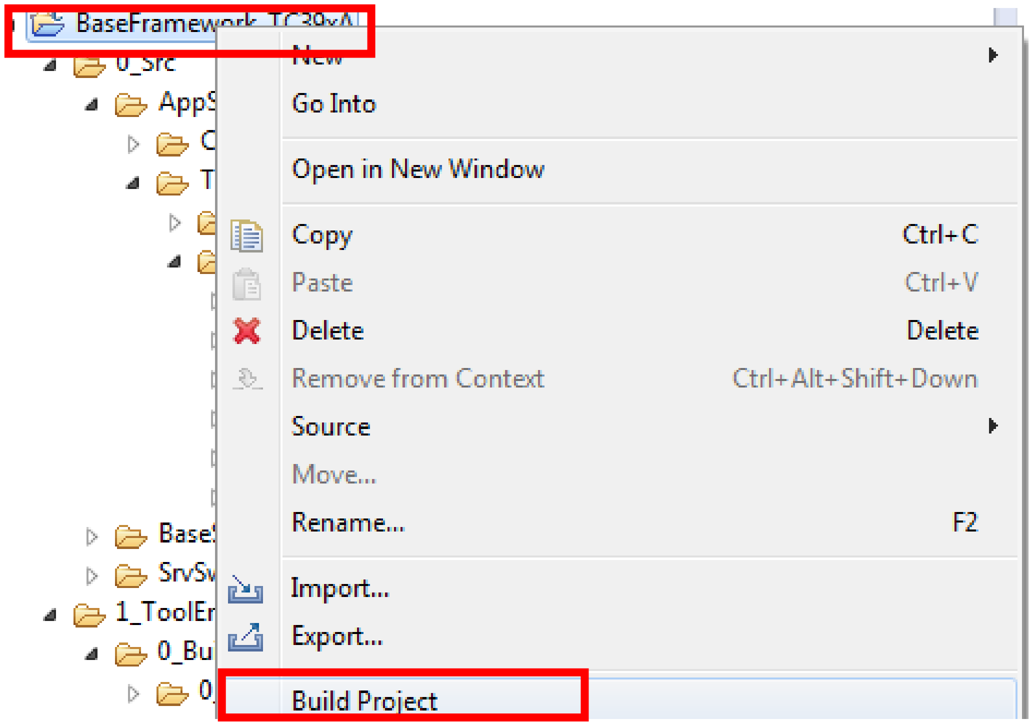
# Status Check



```
 * \file Cpu0_Main.c

#include "Ifx_Types.h"
#include "IfxCpu.h"
#include "IfxScuWdt.h"

IfxCpu_syncEvent cpuSyncEvent=0;

int core0_main (void)
{
    IfxCpu_enableInterrupts();
    /*
     * !!WATCHDOG0 AND SAFETY WATCHDOG ARE DISABLED HERE!!
     * Enable the watchdog in the demo if it is required and also service the watchdog periodically
     * */
    IfxScuWdt_disableCpuWatchdog (IfxScuWdt_getCpuWatchdogPassword ());
    IfxScuWdt_disableSafetyWatchdog (IfxScuWdt_getSafetyWatchdogPassword ());

    /* Cpu sync event wait*/
    IfxCpu_emitEvent(&cpuSyncEvent);
    IfxCpu_waitEvent(&cpuSyncEvent, 1);

    while (1)
    {
        P33_IOCR4.B.PC4 = 0x10;
    }
    return (1);
}
```

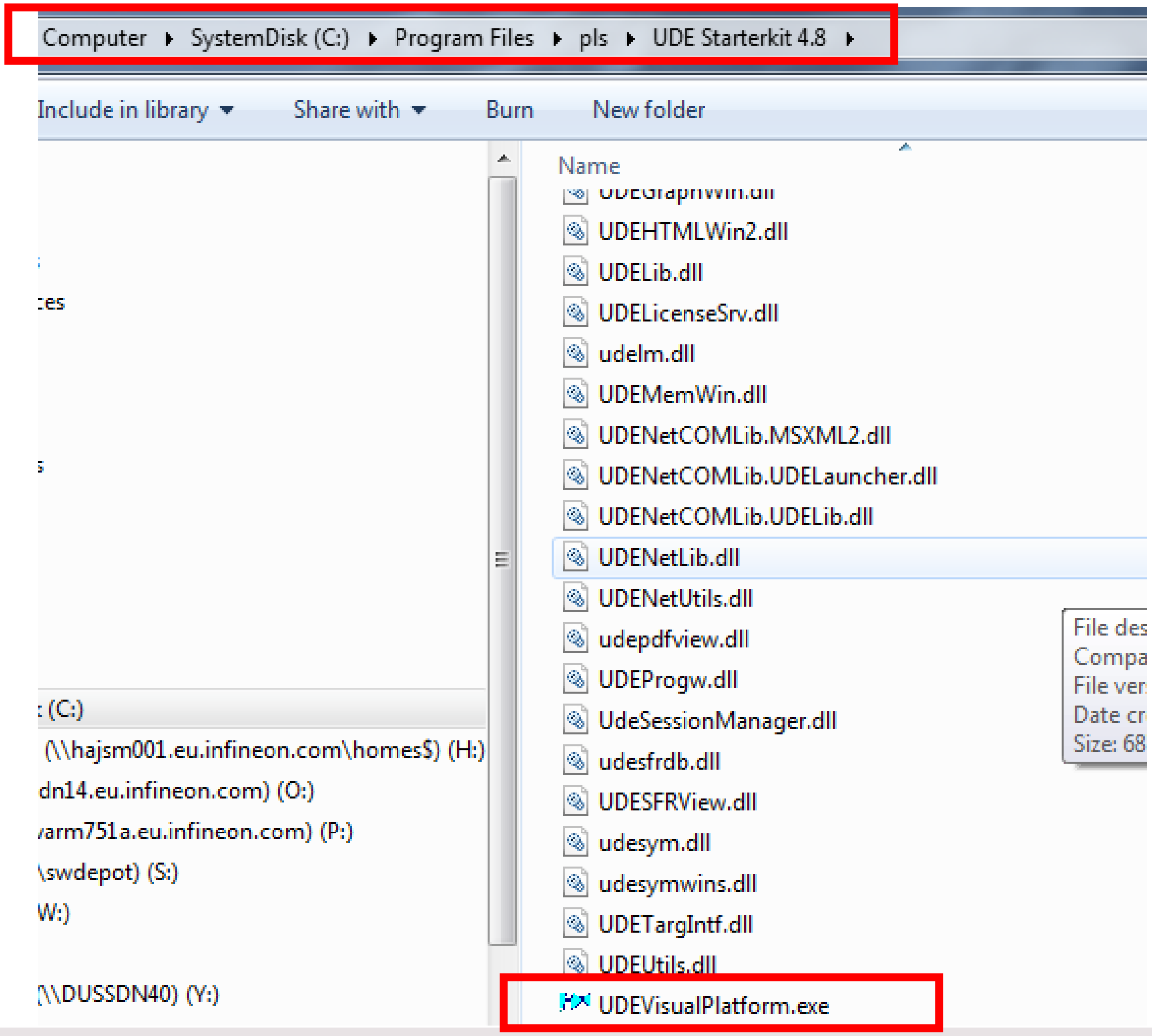
# Compile the Project

– Right click on project name and select „Build Project"



# Debug the Project

› Open UdeVisualPlatofrm.exe


Computer ▸ SystemDisk (C:) ▸ Program Files ▸ pls ▸ UDE Starterkit 4.8 ▸
Include in library ▾
Share with ▾
Burn
New folder
Name

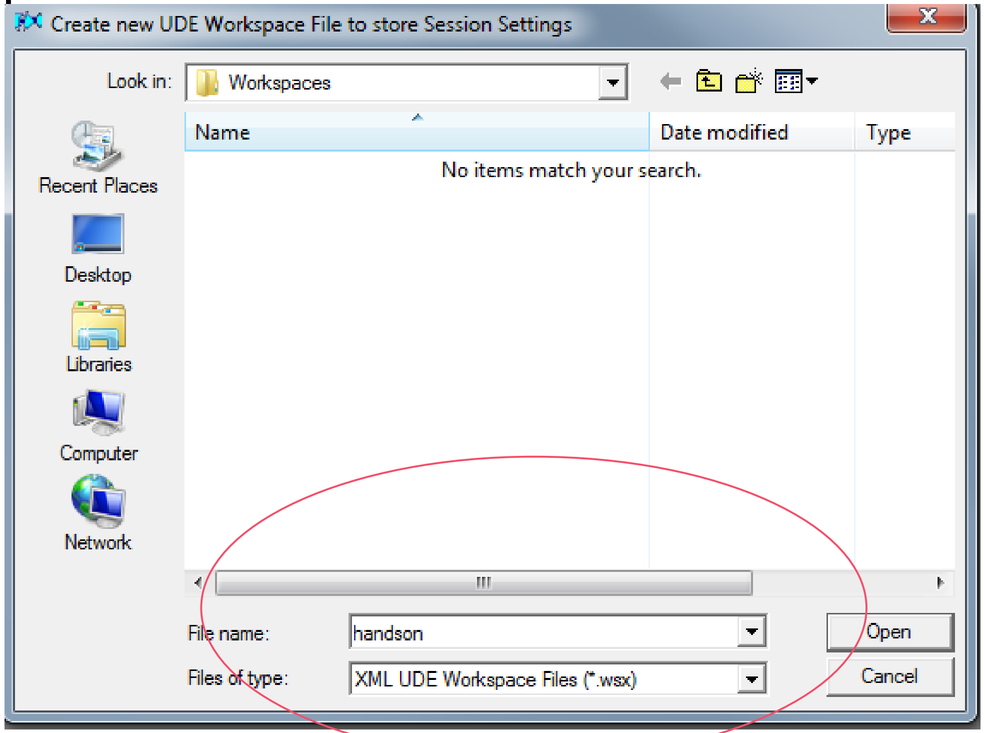
UDEGraphWin.dll
UDEHTMLWin2.dll
UDELib.dll
UDELicenseSrv.dll
udelm.dll
UDEMemWin.dll
UDENetCOMLib.MSXML2.dll
UDENetCOMLib.UDELauncher.dll
UDENetCOMLib.UDELib.dll
UDENetLib.dll
UDENetUtils.dll
udepdfview.dll
UDEProgw.dll
UdeSessionManager.dll
udesfrdb.dll
UDESFRView.dll
udesym.dll
udesymwins.dll
UDETargIntf.dll
UDEUtils.dll
UDEVisualPlatform.exe

File des
Compa
File ver
Date cr
Size: 68

ces
: (C:)
(\\hajsm001.eu.infineon.com\homes$) (H:)
dn14.eu.infineon.com) (O:)
varm751a.eu.infineon.com) (P:)
\swdepot) (S:)
W:)
(\\DUSSDN40) (Y:)

# Create new Debug Workspace

- Select „File“
- Select „New Workspace“
- Create a name
- Open…



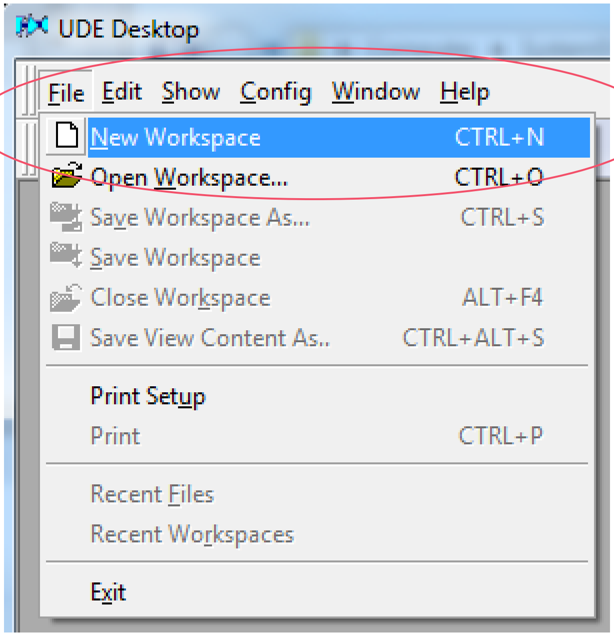
Create new UDE Workspace File to store Session Settings

Look in: Workspaces

| Name | Date modified | Type |
|---|---|---|
| No items match your search. | | |

File name: handson

Files of type: XML UDE Workspace Files (*.wsx)

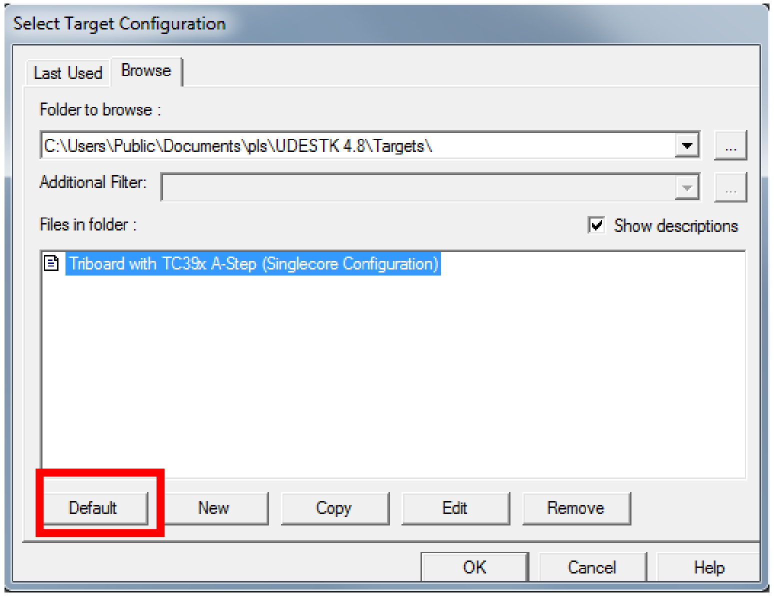


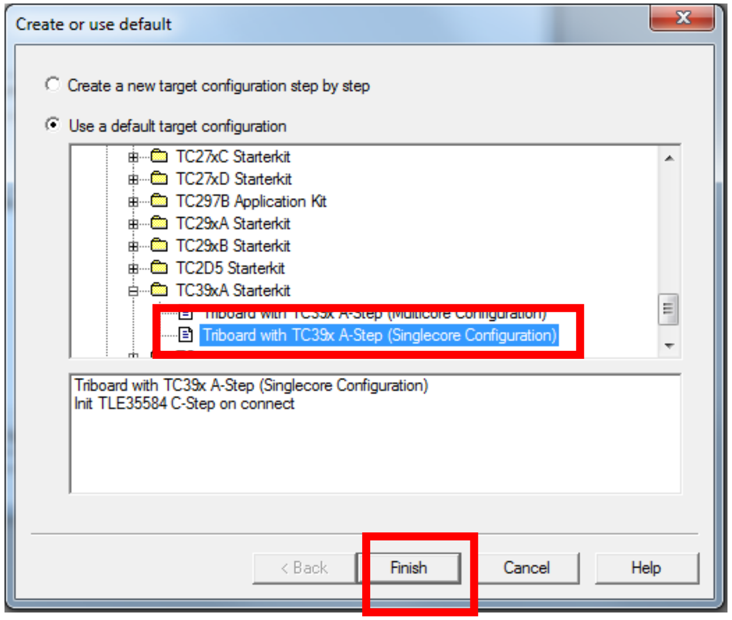
UDE Desktop

| | |
|---|---|
| New Workspace | CTRL+N |
| Open Workspace... | CTRL+O |
| Save Workspace As... | CTRL+S |
| Save Workspace | |
| Close Workspace | ALT+F4 |
| Save View Content As.. | CTRL+ALT+S |
| Print Setup | |
| Print | CTRL+P |
| Recent Files | |
| Recent Workspaces | |
| Exit | |

# Select correct Target

- Click **Default**
- Select **Triboard with TC39x A-step**
- Click **Ok**
- Save it anywhere

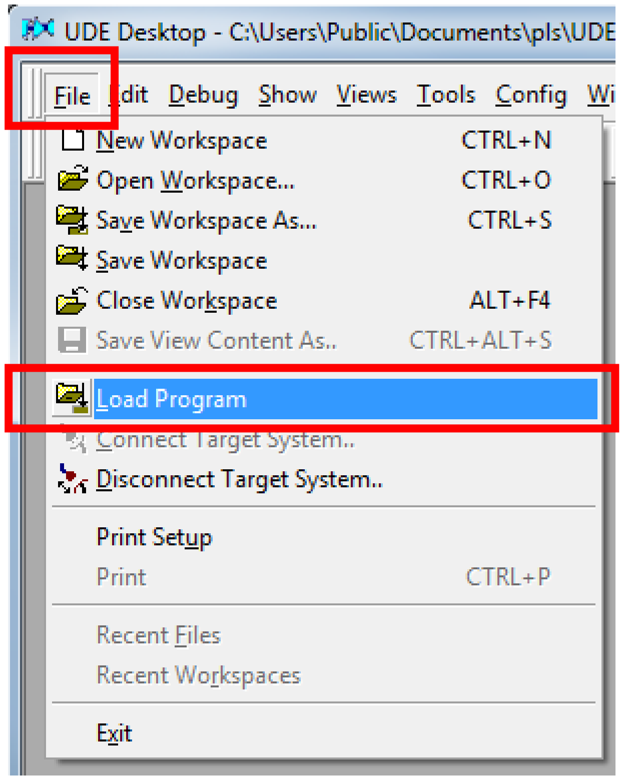

Select Target Configuration
Last Used
Browse
Folder to browse :
C:\Users\Public\Documents\pls\UDESTK 4.8\Targets\
Additional Filter:
Files in folder :
Show descriptions
Triboard with TC39x A-Step (Singlecore Configuration)
Default
New
Copy
Edit
Remove
OK
Cancel
Help

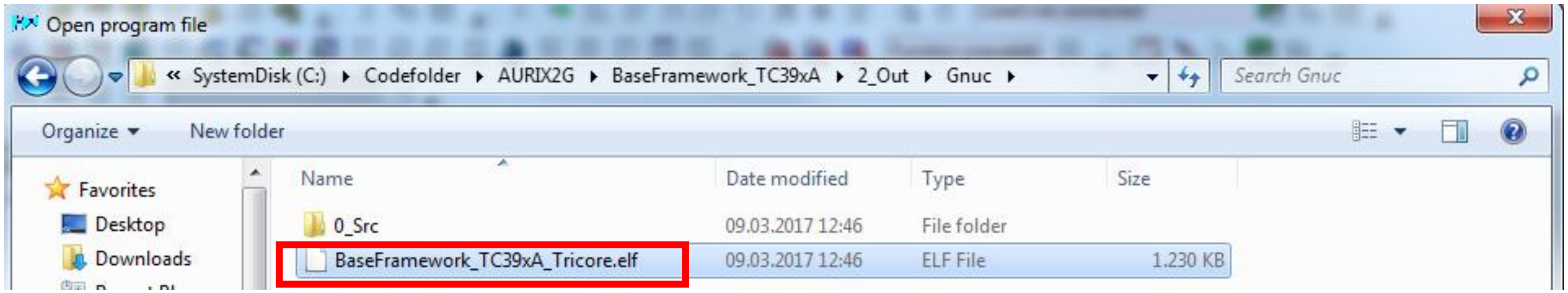

Create or use default
Create a new target configuration step by step
Use a default target configuration
TC27xC Starterkit
TC27xD Starterkit
TC297B Application Kit
TC29xA Starterkit
TC29xB Starterkit
TC2D5 Starterkit
TC39xA Starterkit
Triboard with TC39x A-Step (Multicore Configuration)
Triboard with TC39x A-Step (Singlecore Configuration)
< Back
Finish
Cancel
Help

Triboard with TC39x A-Step (Singlecore Configuration)
Init TLE35584 C-Step on connect

# Load the program

- Click File->Load Program
- Navigate to your project folder
  - 2_Out\Gnuc
- Select the .elf file
- Open…

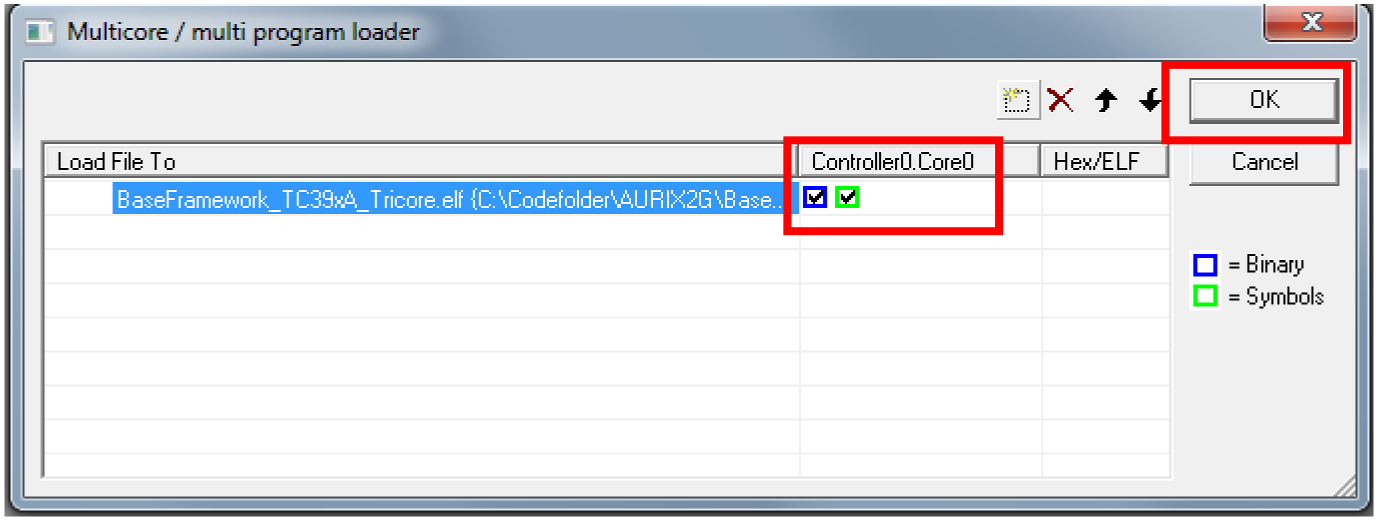

UDE Desktop - C:\Users\Public\Documents\pls\UDE


Open program file

« SystemDisk (C:) ▸ Codefolder ▸ AURIX2G ▸ BaseFramework_TC39xA ▸ 2_Out ▸ Gnuc ▸

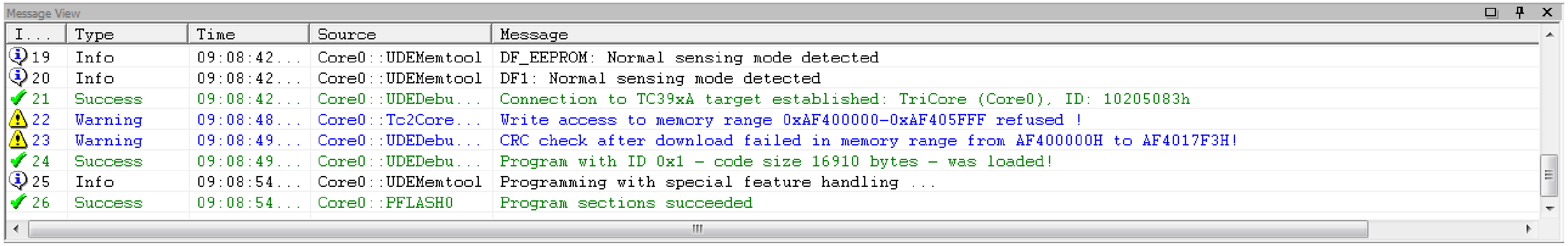
| Name | Date modified | Type | Size |
|---|---|---|---|
| 0_Src | 09.03.2017 12:46 | File folder | |
| BaseFramework_TC39xA_Tricore.elf | 09.03.2017 12:46 | ELF File | 1.230 KB |

# Program the device



Multicore / multi program loader
OK
Cancel
Load File To
Controller0.Core0
Hex/ELF
BaseFramework_TC39xA_Tricore.elf {C:\Codefolder\AURIX2G\Base...
= Binary
= Symbols

› Select „Program“
› You should see a success message on the bottom

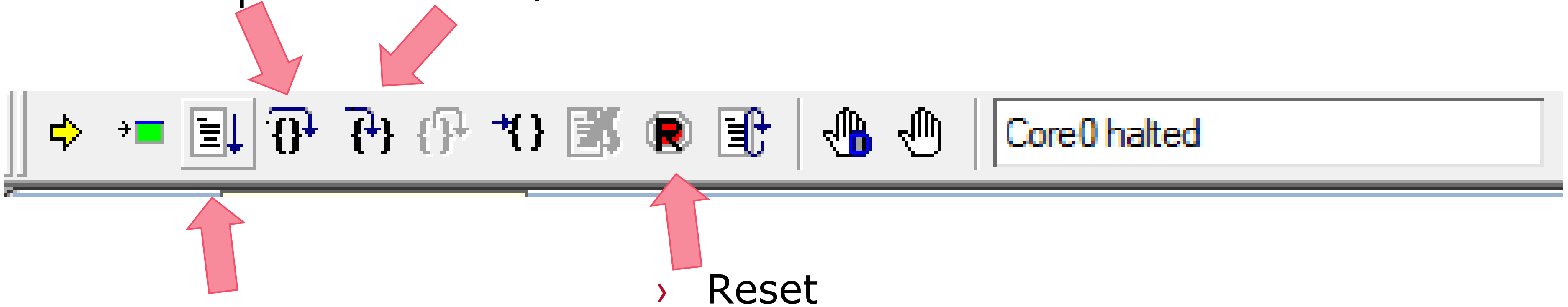


Message View

```
I...  Type     Time         Source            Message
19    Info     09:08:42...  Core0::UDEMemtool  DF_EEPROM: Normal sensing mode detected
20    Info     09:08:42...  Core0::UDEMemtool  DF1: Normal sensing mode detected
21    Success  09:08:42...  Core0::UDEDebu...  Connection to TC39xA target established: TriCore (Core0), ID: 10205083h
22    Warning  09:08:48...  Core0::Tc2Core...  Write access to memory range 0xAF400000-0xAF405FFF refused !
23    Warning  09:08:49...  Core0::UDEDebu...  CRC check after download failed in memory range from AF400000H to AF4017F3H!
24    Success  09:08:49...  Core0::UDEDebu...  Program with ID 0x1 - code size 16910 bytes - was loaded!
25    Info     09:08:54...  Core0::UDEMemtool  Programming with special feature handling ...
26    Success  09:08:54...  Core0::PFLASH0     Program sections succeeded
```

# Debug the Project

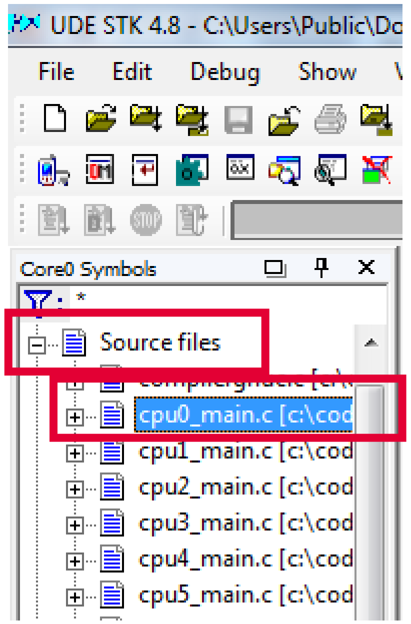
- Step Over
- Step Into

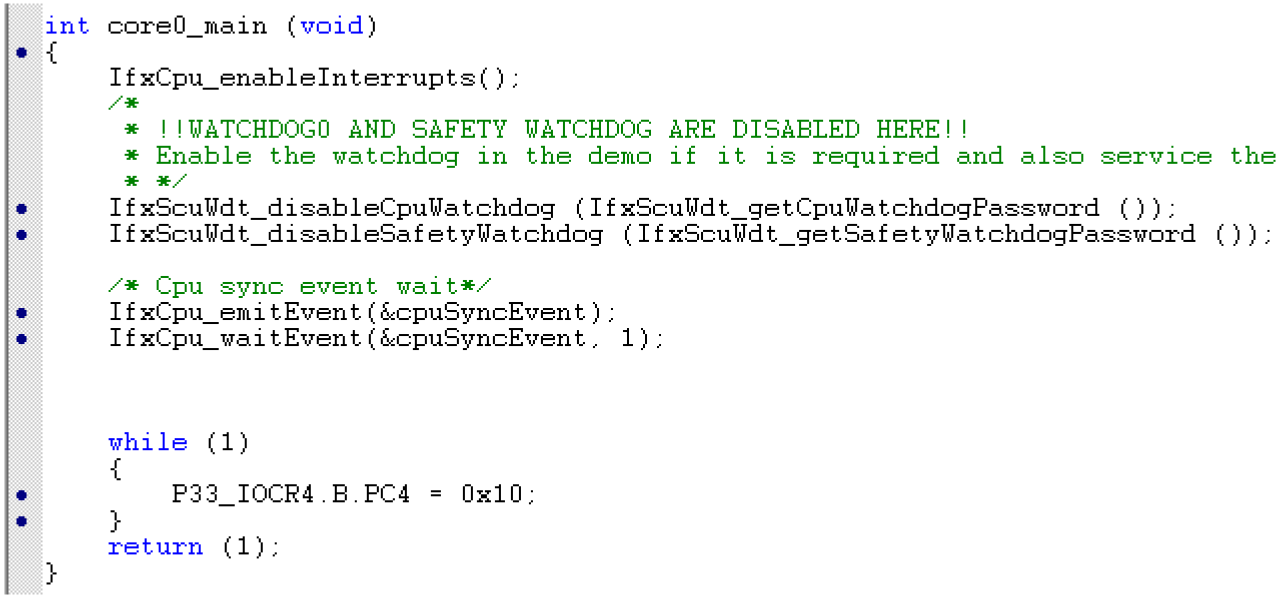

Core0 halted

- Reset
- Run Program

# Open Cpu0_Main.c

- On the left side chose „Source files“ and then cpu0_main.c
- Double-click on cpu0_main.c to open it

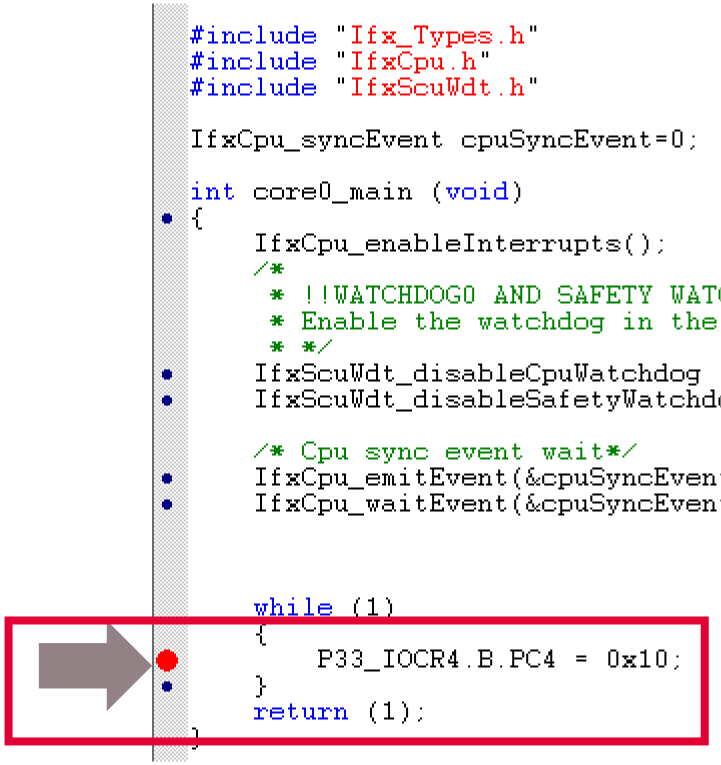




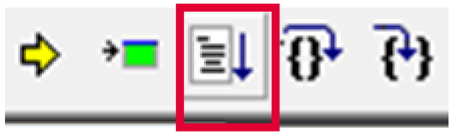
```
int core0_main (void)
{
    IfxCpu_enableInterrupts();
    /*
     * !!WATCHDOG0 AND SAFETY WATCHDOG ARE DISABLED HERE!!
     * Enable the watchdog in the demo if it is required and also service the
     * */
    IfxScuWdt_disableCpuWatchdog (IfxScuWdt_getCpuWatchdogPassword ());
    IfxScuWdt_disableSafetyWatchdog (IfxScuWdt_getSafetyWatchdogPassword ());

    /* Cpu sync event wait*/
    IfxCpu_emitEvent(&cpuSyncEvent);
    IfxCpu_waitEvent(&cpuSyncEvent, 1);


    while (1)
    {
        P33_IOCR4.B.PC4 = 0x10;
    }
    return (1);
}
```

# Place Breakpoint

- Place a breakpoint in the line toggling the LED pin
- Let the program run
- You see it stopping and the LED toggles each time the program is run



```
#include "Ifx_Types.h"
#include "IfxCpu.h"
#include "IfxScuWdt.h"

IfxCpu_syncEvent cpuSyncEvent=0;

int core0_main (void)
{
    IfxCpu_enableInterrupts();
    /*
     * !!WATCHDOG0 AND SAFETY WATCHDOG ARE DISABLED HERE!!
     * Enable the watchdog in the demo if it is required and also service the
     * */
    IfxScuWdt_disableCpuWatchdog (IfxScuWdt_getCpuWatchdogPassword ());
    IfxScuWdt_disableSafetyWatchdog (IfxScuWdt_getSafetyWatchdogPassword ());

    /* Cpu sync event wait*/
    IfxCpu_emitEvent(&cpuSyncEvent);
    IfxCpu_waitEvent(&cpuSyncEvent, 1);


    while (1)
    {
        P33_IOCR4.B.PC4 = 0x10;
    }
    return (1);
}
```



Part of your life. Part of tomorrow.