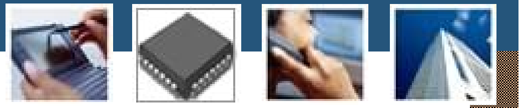
AndeSight^{TM} MCU version v2.0.0

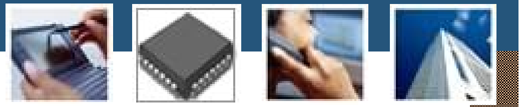
Andes Technology
Tel: +886-3-6668300
Business : sales@andestech.com
Technical : support@andestech.com

Agenda



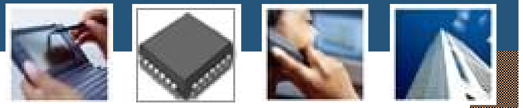
- ❖ Introduction to BSP and AndeSight
- ❖ Introduction of AndeSight installation
  - AndeSight installation
  - How to deploy license
  - 2-wire AICE
- ❖ AndeSight overview
  - Overview
  - target

Agenda



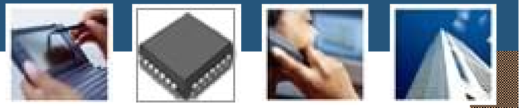
- ❖ Via chip profile create a new project (simulator)
  - Build
  - Run (Console View)
  - Debug
  - Profiling
- ❖ Stop the simulator, demo AICE
  - AICE plug-in detect, Target Monitor
  - Terminal View
  - Run and Debug on EVB (via AICE)
  - Target manipulation
  - How to change the toolchain

Agenda



- ❖ Debug Perspective – (using JPEG demo)
  - Debug assembly
  - Memory View
  - Memory Browser View
  - Register View
  - SOC register View
  - GDB command View

Agenda



- ❖ Compiler option setting
  - How to add compiler option
  - Optimization option for speed and space
  - GNU Utility setting
- ❖ Makefile project and C project
  - Generic project demo
  - The environment variable of Makefile project
- ❖ Flash burn and binary debugging

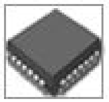
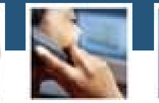
Agenda

- IntelJ3 burning introduction
  - The IntelJ3 program and IntelJ3 spec
- AndeSight200MCUbeta under Start menu
  - AICE
  - Documents
  - Toolchains

Agenda

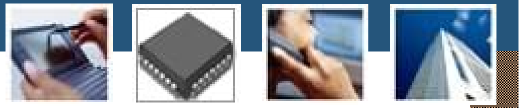
- Demo program
  - JPEG
  - demo-lm
  - demo-ls1
  - demo-ls2
  - demo-ls3
  - demo-int
  - demo-int-c-ext
  - demo-pfm
  - demo-cache

Agenda

- How to import a program
  - From file system
  - From existing project
- How to create Chip Profile
  - Chip Profile setting
  - How to use SOCgenerator
- Plug-in
  - ClientTCF demo

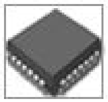
Agenda

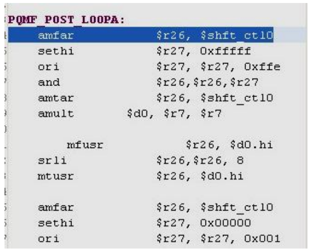


- Some tools:
  - file explorer
  - open element
  - trace symbol
- Resource on Internet

❖ Introduction to BSP and AndeSight

Andes' Main Lines of Business





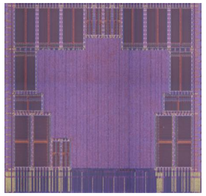
```
PQMF_POST_LOOPA:
     amfar             $r26, $shft_ctl0
     sethi             $r27, 0xfffff
     ori               $r27, $r27, 0xffe
     and               $r26,$r26,$r27
     amtar             $r26, $shft_ctl0
     amult         $d0, $r7, $r7

         mfusr             $r26, $d0.hi
     srli              $r26,$r26, 8
     mtusr             $r26, $d0.hi

     amfar             $r26, $shft_ctl0
     sethi             $r27, 0x00000
     ori               $r27, $r27, 0x001
```

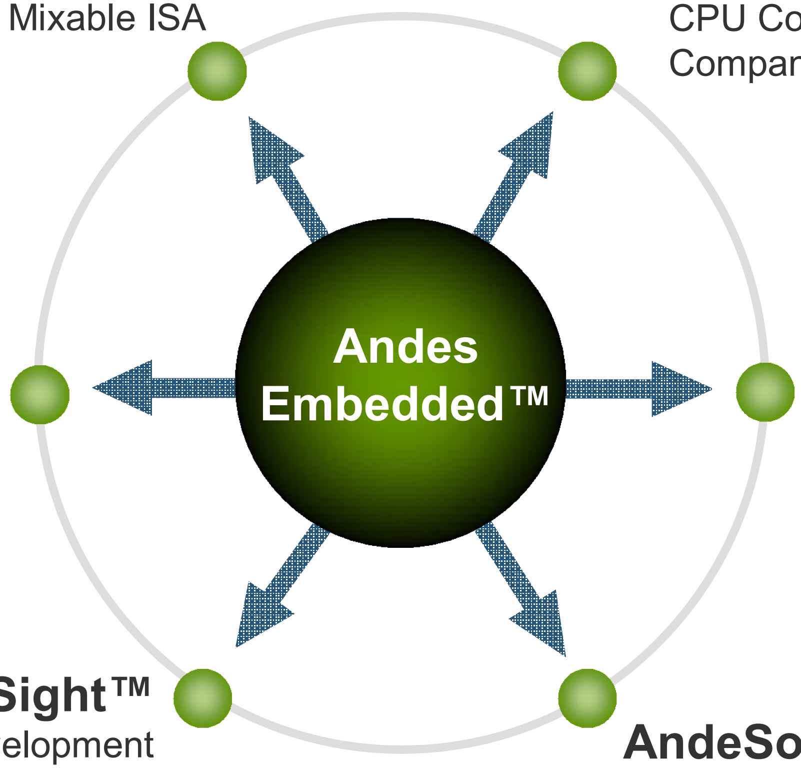
AndeStar™

Patented 16/32-bit
Mixable ISA

AndesCore™

CPU Core Families
Companion Engine

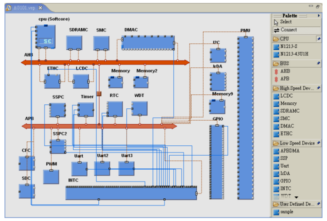




Andes
Embedded™

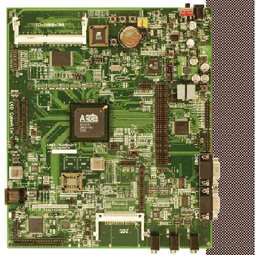
AndESLive™

ESL Integrated
Virtual Environment



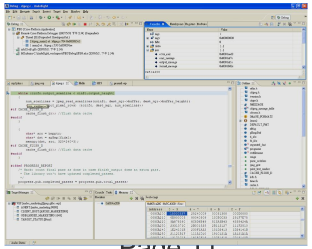
AndeShape™

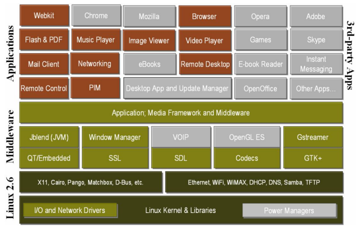
SoC + EVB + ICE



AndeSight™

Integrated Development
Environment





Applications
Webkit | Chrome | Mozilla | Browser | Opera | Adobe
Flash & PDF | Music Player | Image Viewer | Video Player | Games | Skype
Mail Client | Networking | eBooks | Remote Desktop | E-book Reader | Instant Messaging
Remote Control | PIM | Desktop App and Update Manager | OpenOffice | Other Apps…

3rd-party Apps

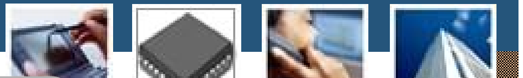
Middleware
Application; Media Framework and Middleware
Jblend (JVM) | Window Manager | VOIP | OpenGL ES | Gstreamer
QT/Embedded | SSL | SDL | Codecs | GTK+

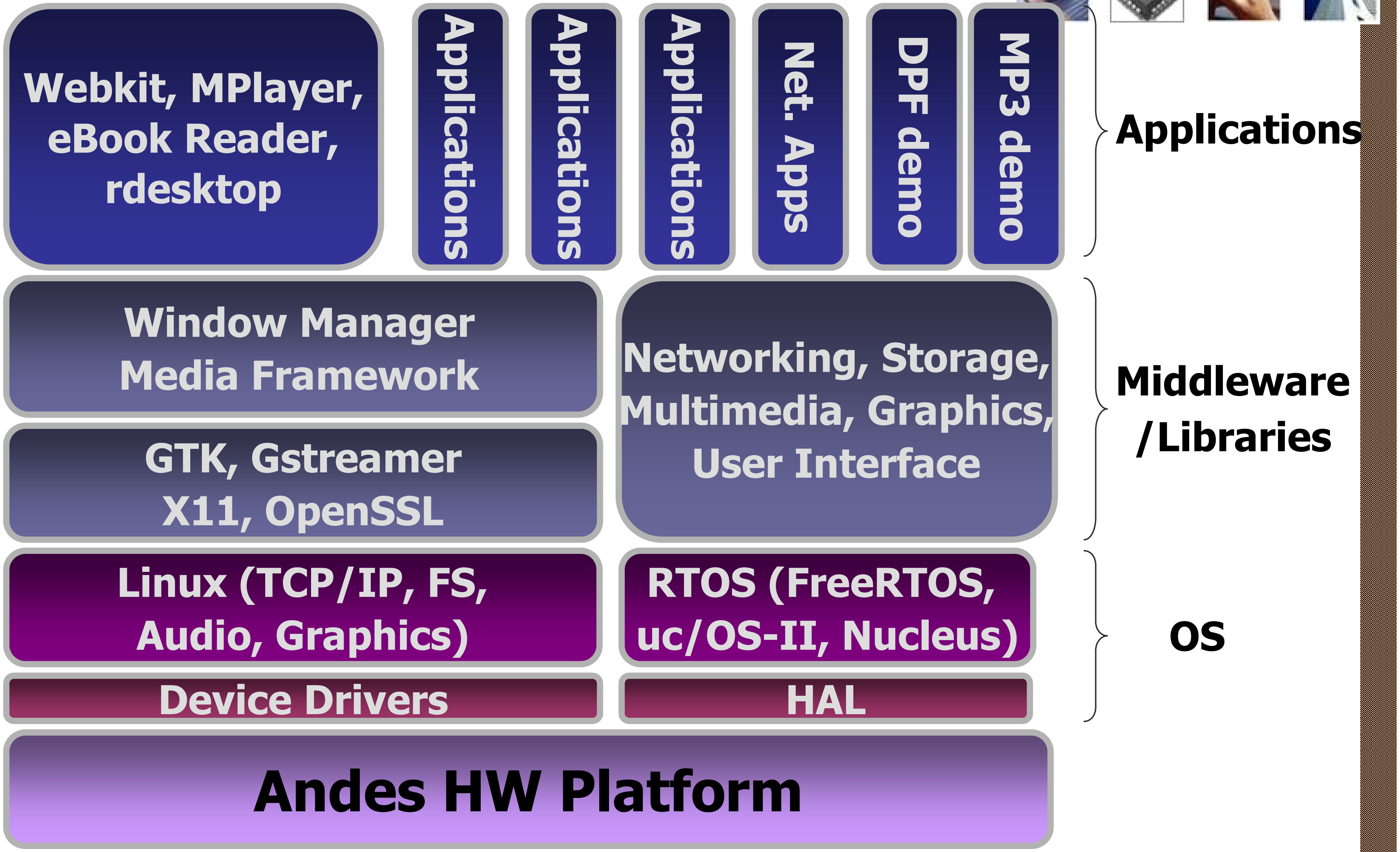
Linux 2.6
X11, Cairo, Pango, Matchbox, D-Bus, etc. | Ethernet, WiFi, WiMAX, DHCP, DNS, Samba, TFTP
I/O and Network Drivers | Linux Kernel & Libraries | Power Managers

AndeSoft™

Optimized Target SW Stack
Including Linux/RTOS, Drivers,
Middleware, and Applications

AndeSoft™ Stack – Linux and RTOS



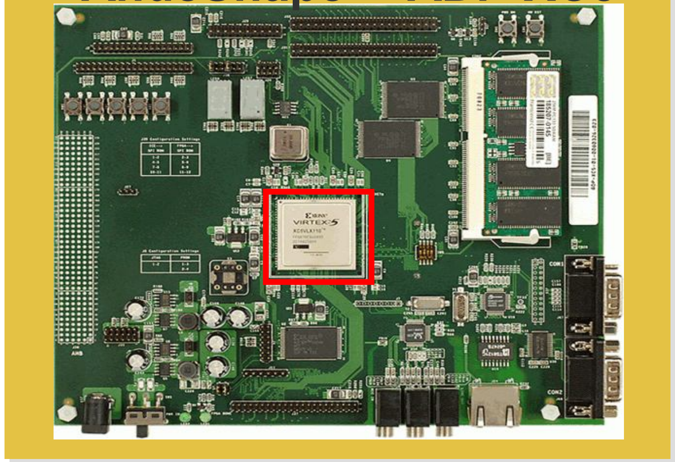


AndeSoft™ BSP

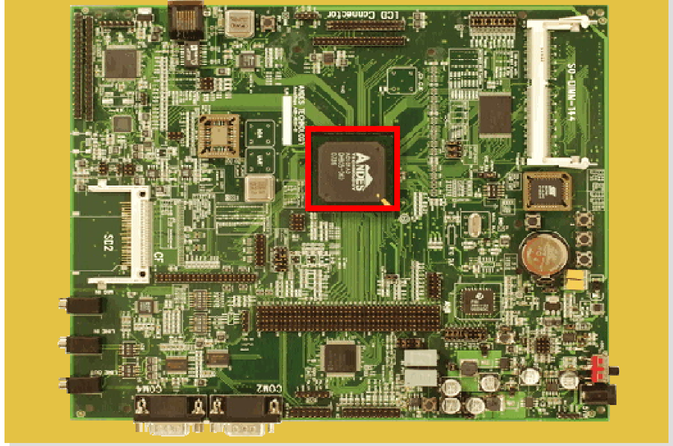
❖ Andes SoC dev. platform

- Toolchain
  - glibc, uClibc, newlib and mculib libraries
- U-boot
- Non-OS environment
  - Startup sample code
  - MP3, JPEG demo
- RTOS environment
  - uC/OS-II, FreeRTOS
  - UART, MAC, LCD, AC97 and SD drivers.
  - LWIP
  - DPF demo

AndeShape™ ADP-XC5



AndeShape™ ADP-AG101



AndeSoft™ BSP

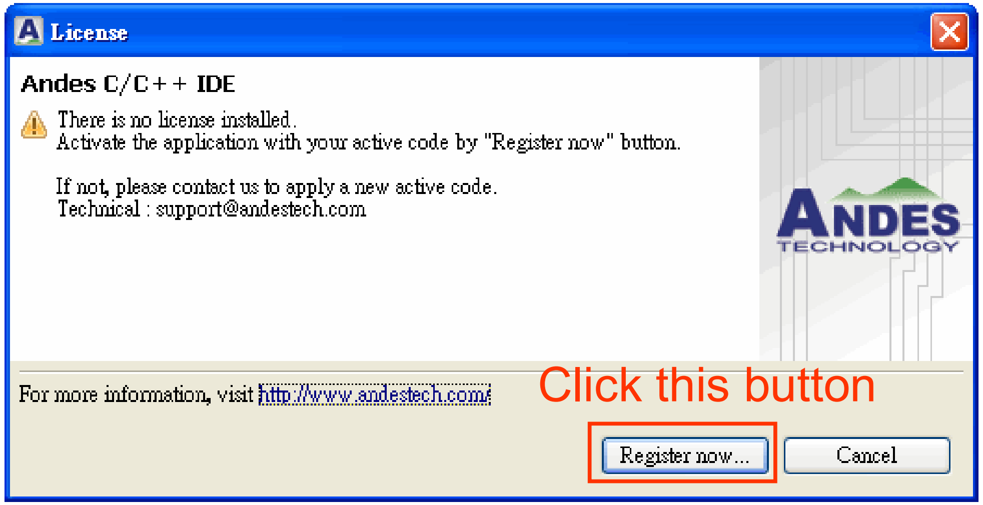
- Linux environment
  - Boot up from SD, Flash, TFTP
  - Kernel 2.6.32 with Andes architecture options
  - Utility
    - POSIX timer
    - Performance monitor
    - MTD (Memory Device)
    - OProfile
  - Applications
    - Busybox
    - MPlayer
    - Fbv

AndeShape™ ADP-AG102

- Introduction of AndeSight installation
  - AndeSight installation
  - How to deploy license
  - 2-wire AICE

License deploy(1)

- After installation, AndeSight^{TM} will prompt you to enter activation code.



License

Andes C/C++ IDE

There is no license installed.
Activate the application with your active code by "Register now" button.

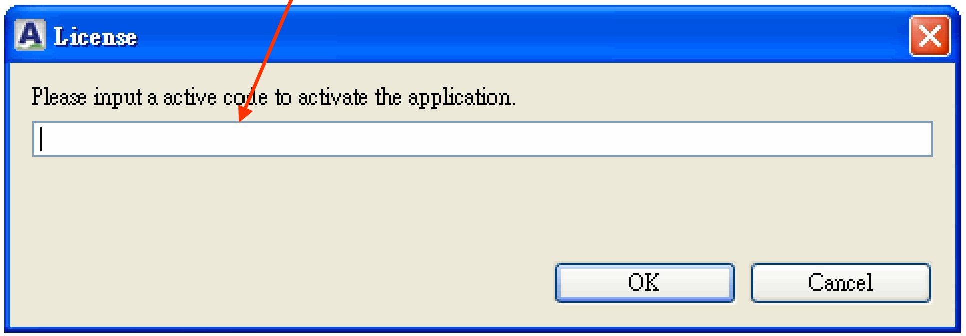
If not, please contact us to apply a new active code.
Technical : support@andestech.com

For more information, visit http://www.andestech.com/

Click this button

```
Andes,110930 : A70D3230422E4D6BE7AB5ACA9C93EAD842221C0[redacted]
```

Enter activation code here

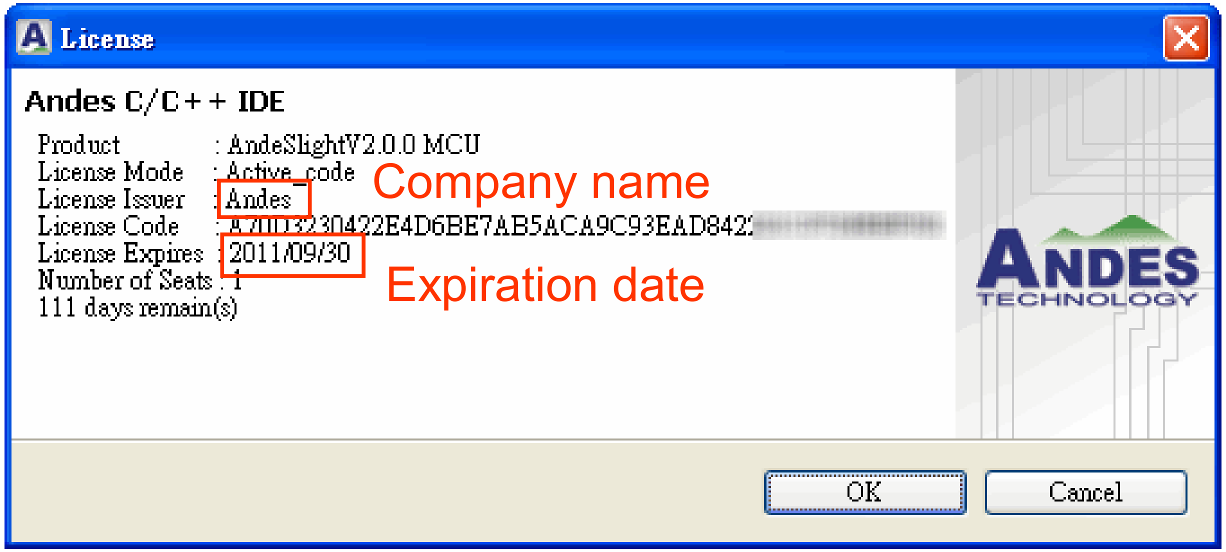


License

Please input a active code to activate the application.

License deploy(2)

❖ AndeSight^{TM} will show the license period

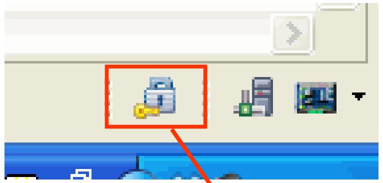


Andes C/C++ IDE

```
Product         : AndeSlightV2.0.0 MCU
License Mode    : Active_code
License Issuer  : Andes
License Code    : A70D3230422E4D6BE7AB5ACA9C93EAD842[redacted]
License Expires : 2011/09/30
Number of Seats : 1
111 days remain(s)
```

Company name

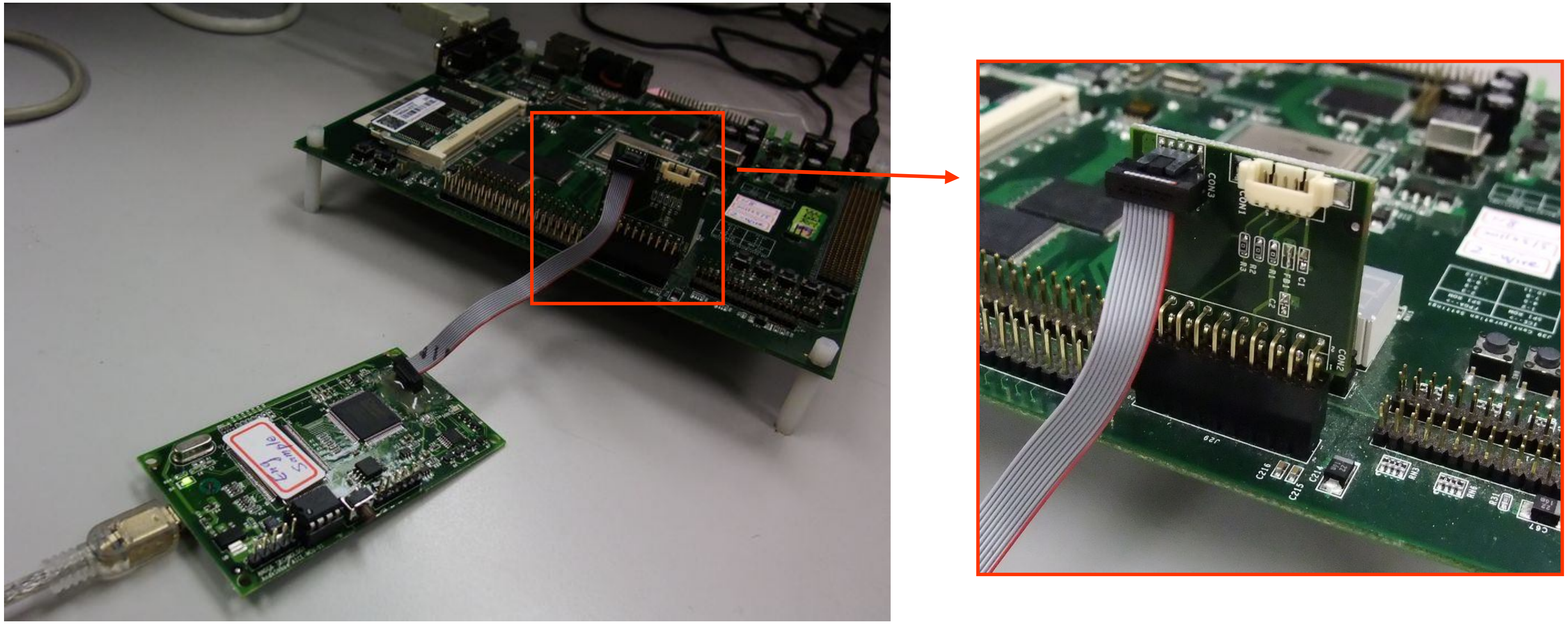
Expiration date

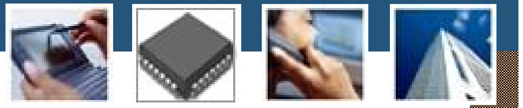


License control at the lower right corner of AndeSight has an undeploy function

How to connect 2-wire AICE™

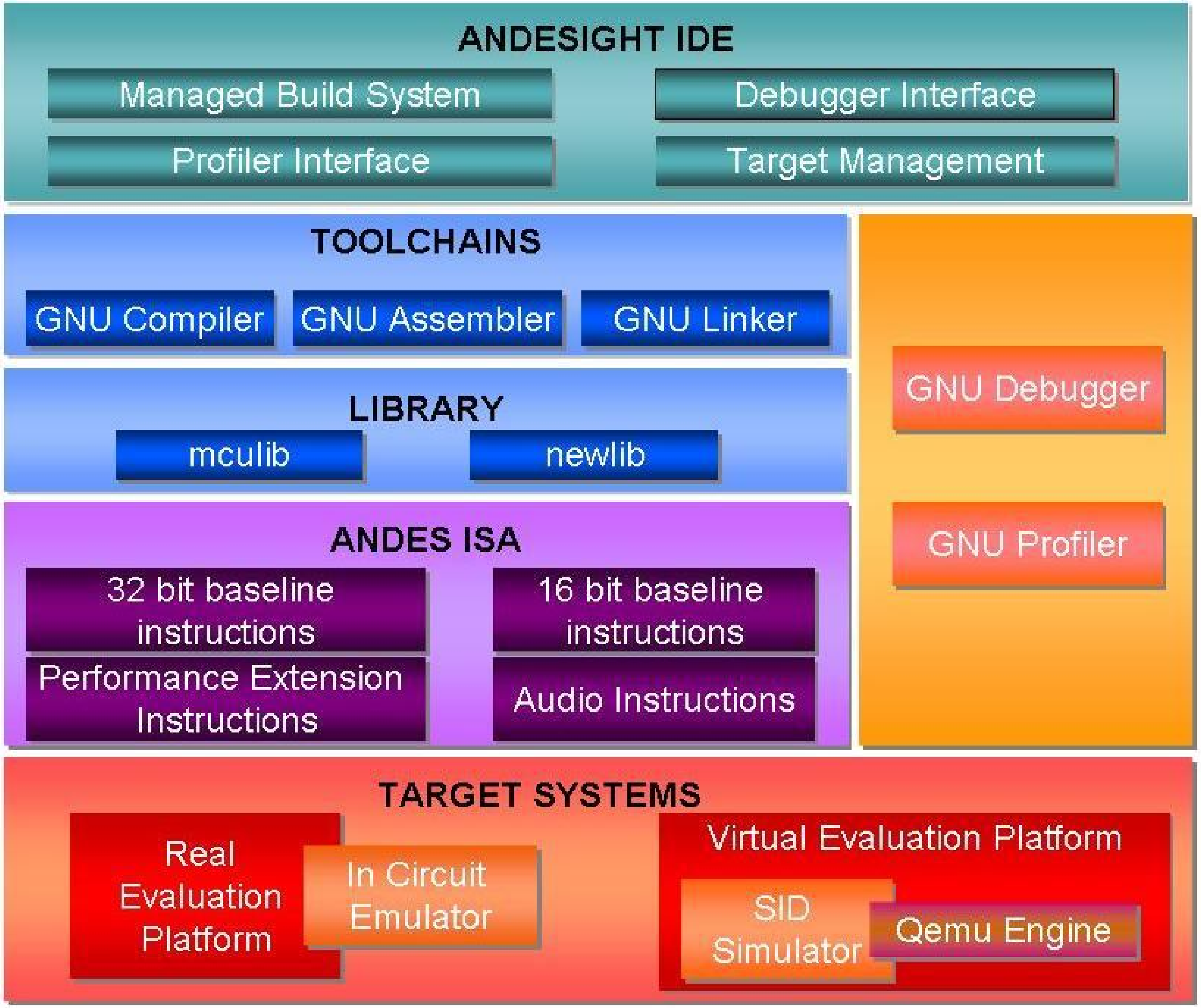
- ❖ Pin1 connect to Pin1



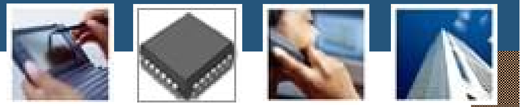


- ❖ AndeSight overview
  - Overview
  - target

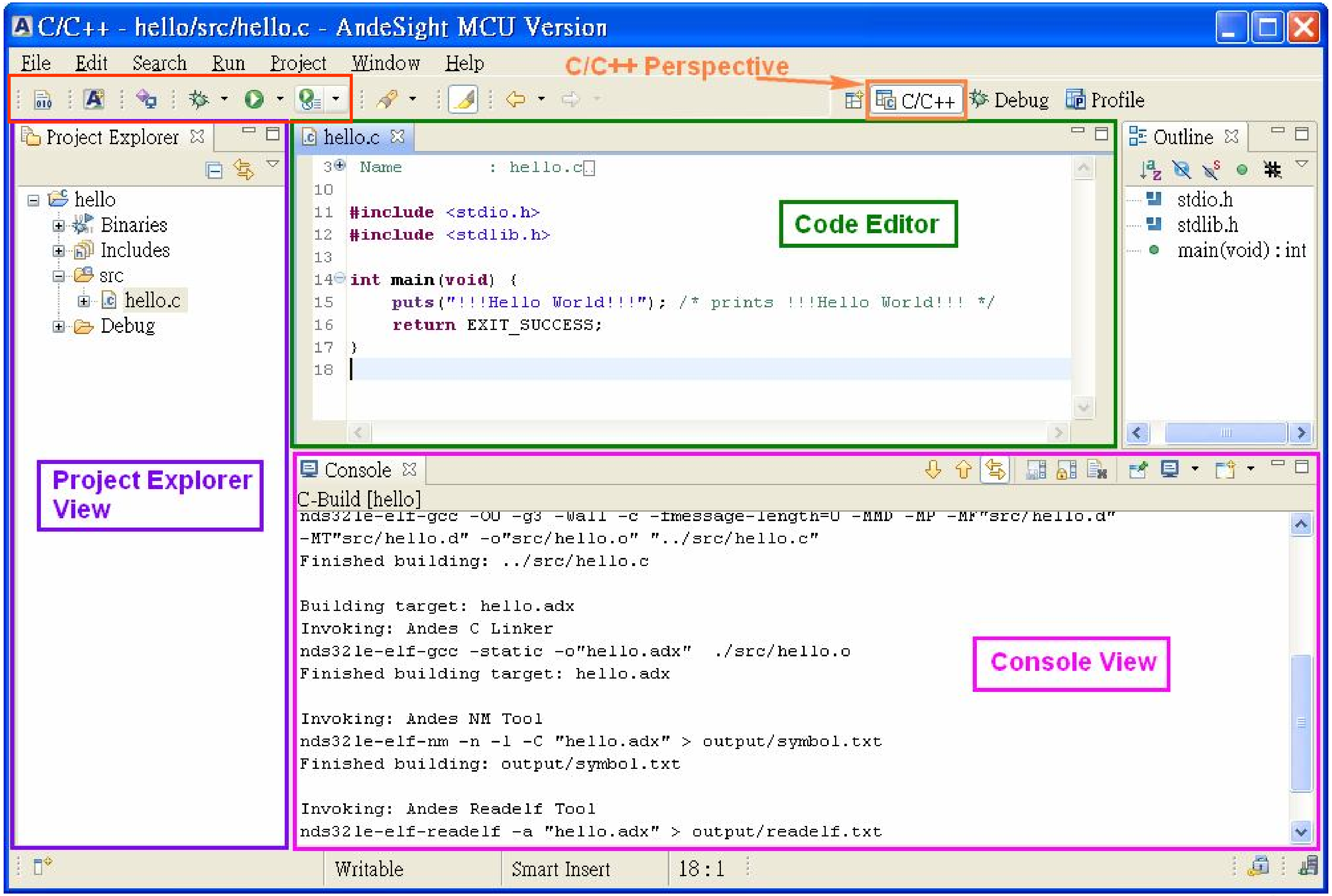
IDE Overview (1)



IDE Overview (2)



toolbar



C/C++ Perspective

Code Editor

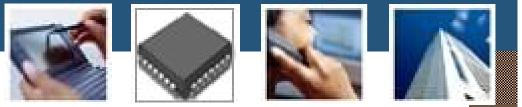
```
 Name          : hello.c

#include <stdio.h>
#include <stdlib.h>

int main(void) {
    puts("!!!Hello World!!!"); /* prints !!!Hello World!!! */
    return EXIT_SUCCESS;
}
```

Project Explorer View

```
C-Build [hello]
nds32le-elf-gcc -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"src/hello.d"
-MT"src/hello.d" -o"src/hello.o" "../src/hello.c"
Finished building: ../src/hello.c

Building target: hello.adx
Invoking: Andes C Linker
nds32le-elf-gcc -static -o"hello.adx"  ./src/hello.o
Finished building target: hello.adx

Invoking: Andes NM Tool
nds32le-elf-nm -n -l -C "hello.adx" > output/symbol.txt
Finished building: output/symbol.txt

Invoking: Andes Readelf Tool
nds32le-elf-readelf -a "hello.adx" > output/readelf.txt
```

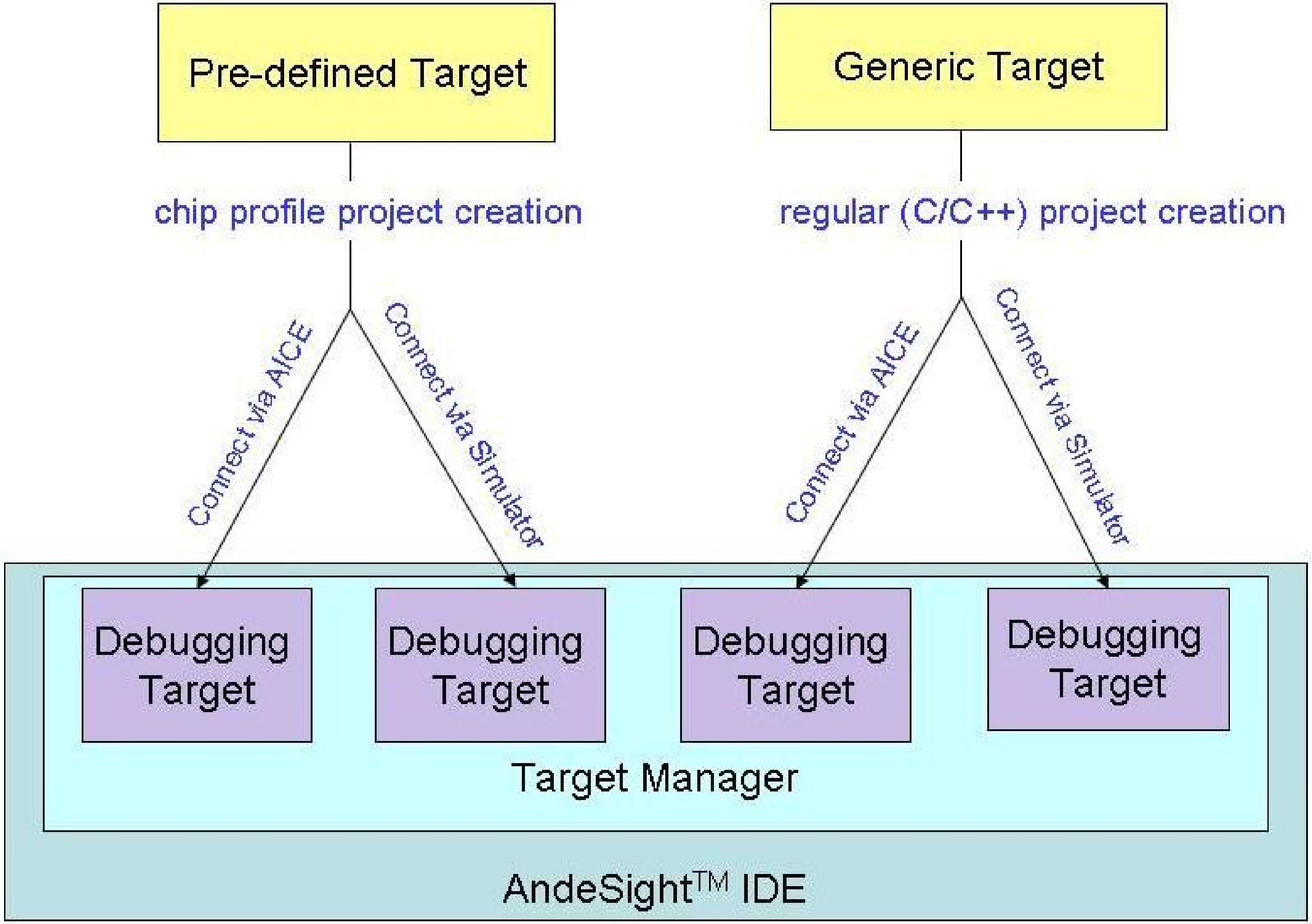
Console View

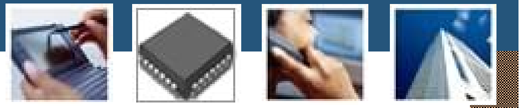
Andes Toolchains and Their Andes Cores



Toolchain Name	Andes Cores				
	N1033A-S	N1033-S	N903A-S	N903-S	N801-S
nds32le-elf-[newlib|mculib]-v2	▪	▪	▪	▪ (for 32 GPR)	
nds32[be|le]-elf-[newlib|mculib]-v2j				▪ (for 16 GPR)	
nds32le-elf-[newlib|mculib]-v3m					▪

Target Systems in Relation to AndeSight™

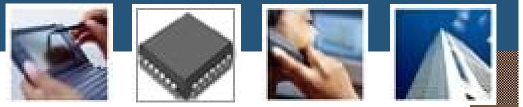


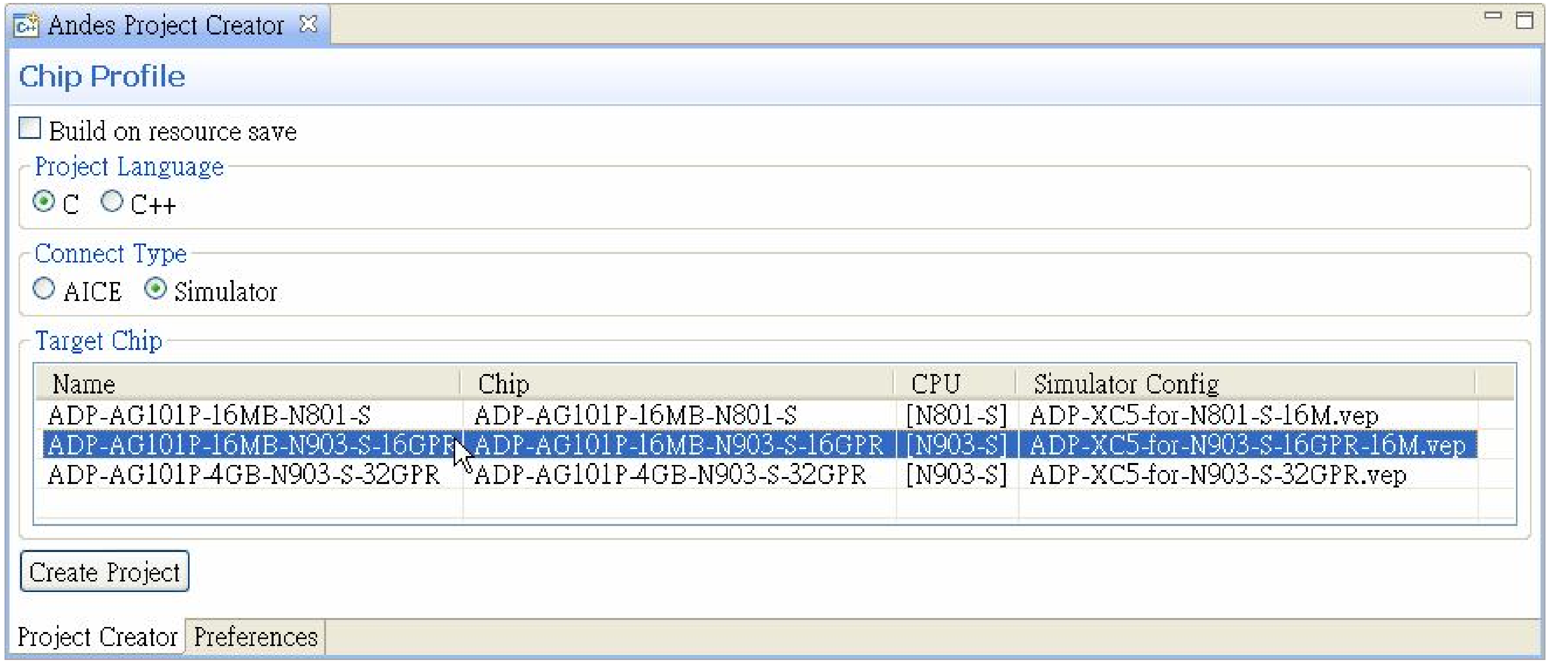


❖ Via chip profile create a new project (simulator)

- Build
- Run
- Console View
- Debug
- Profiling

Create a New Project for a Pre-defined Target (1)





Andes Project Creator

Chip Profile

□ Build on resource save

Project Language

● C ○ C++

Connect Type

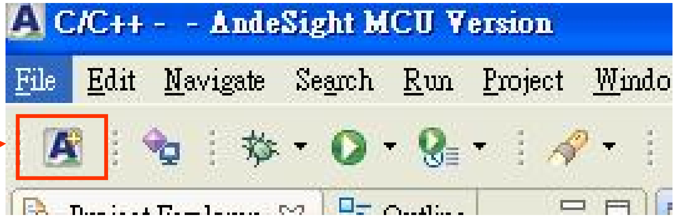
○ AICE ● Simulator

Target Chip

Name	Chip	CPU	Simulator Config
ADP-AG101P-16MB-N801-S	ADP-AG101P-16MB-N801-S	[N801-S]	ADP-XC5-for-N801-S-16M.vep
ADP-AG101P-16MB-N903-S-16GPR	ADP-AG101P-16MB-N903-S-16GPR	[N903-S]	ADP-XC5-for-N903-S-16GPR-16M.vep
ADP-AG101P-4GB-N903-S-32GPR	ADP-AG101P-4GB-N903-S-32GPR	[N903-S]	ADP-XC5-for-N903-S-32GPR.vep

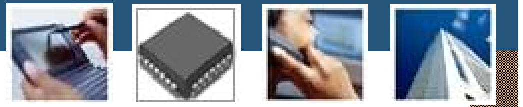
Create Project

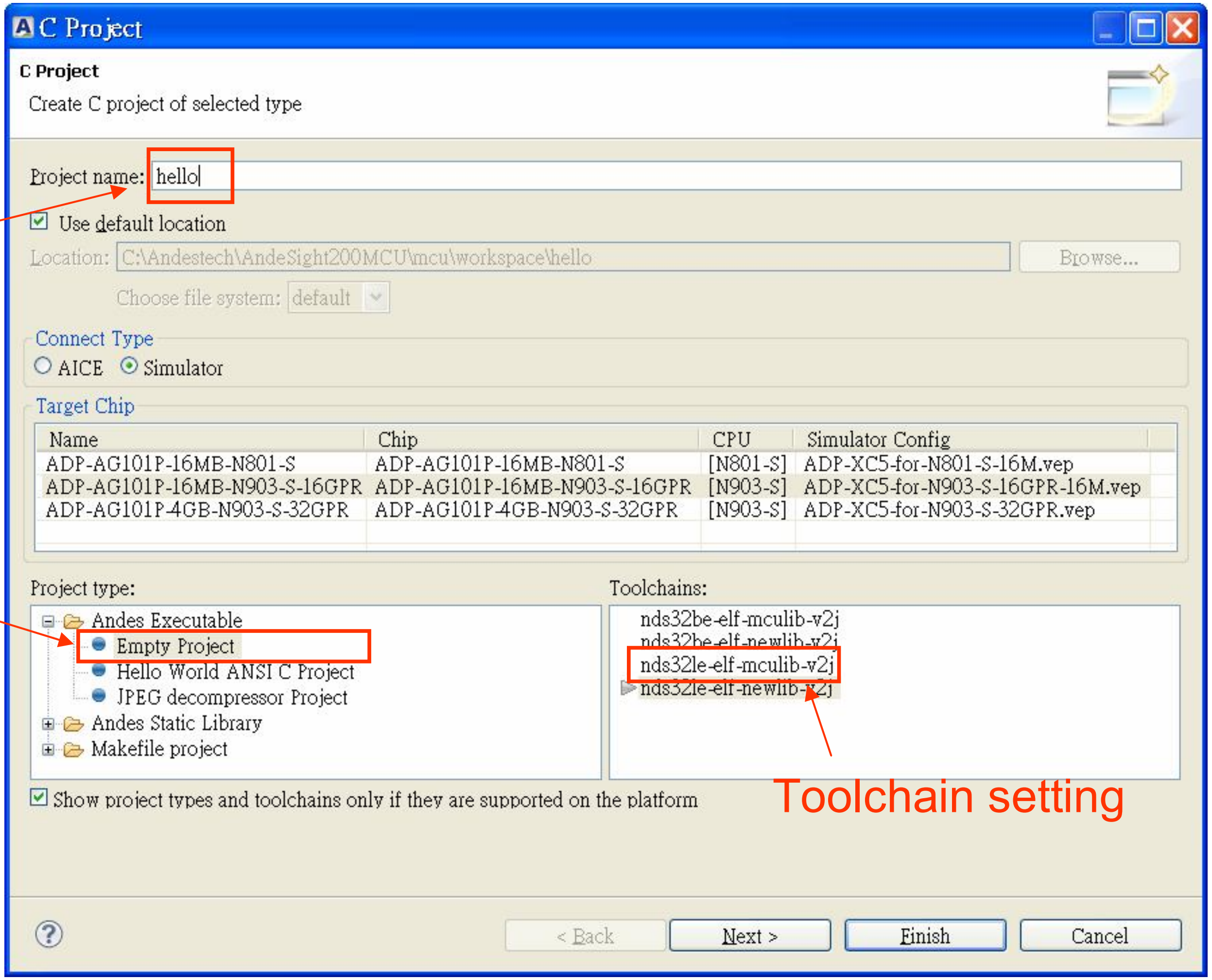
Project Creator | Preferences



Click here to call Andes Project Creator

Create a New Project for a Pre-defined Target (2)





C Project

Create C project of selected type

Project name: hello

☑ Use default location

Location: C:\Andestech\AndeSight200MCU\mcu\workspace\hello

Choose file system: default

Connect Type

○ AICE ● Simulator

Target Chip

Name	Chip	CPU	Simulator Config
ADP-AG101P-16MB-N801-S	ADP-AG101P-16MB-N801-S	[N801-S]	ADP-XC5-for-N801-S-16M.vep
ADP-AG101P-16MB-N903-S-16GPR	ADP-AG101P-16MB-N903-S-16GPR	[N903-S]	ADP-XC5-for-N903-S-16GPR-16M.vep
ADP-AG101P-4GB-N903-S-32GPR	ADP-AG101P-4GB-N903-S-32GPR	[N903-S]	ADP-XC5-for-N903-S-32GPR.vep

Project type:

- Andes Executable
  - Empty Project
  - Hello World ANSI C Project
  - JPEG decompressor Project
- Andes Static Library
- Makefile project

Toolchains:

- nds32be-elf-mculib-v2j
- nds32be-elf-newlib-v2j
- nds32le-elf-mculib-v2j
- nds32le-elf-newlib-v2j

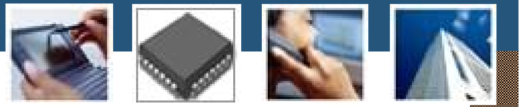
☑ Show project types and toolchains only if they are supported on the platform

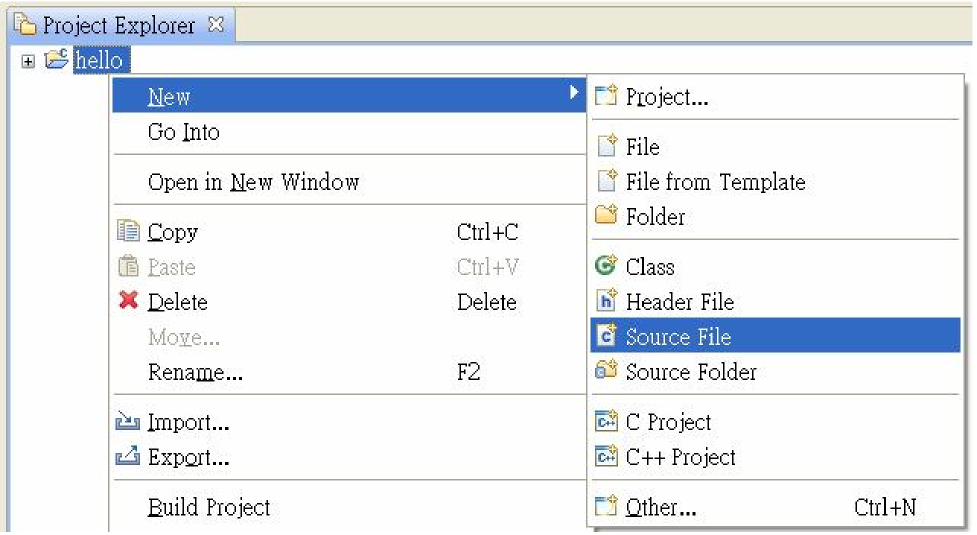
project name

Empty Project

Toolchain setting

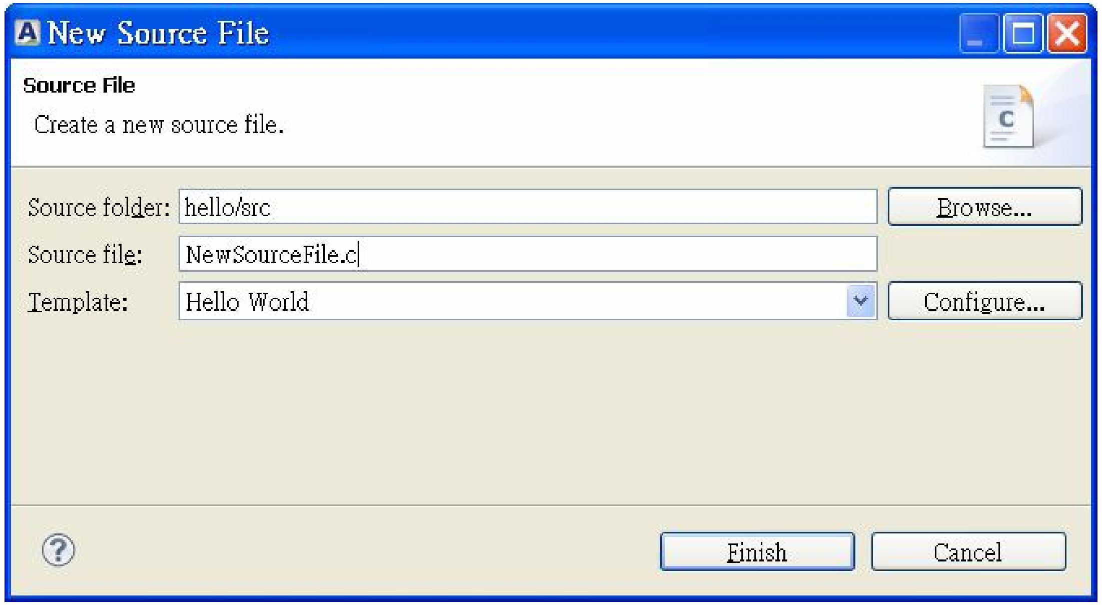
Create a New Source/Header File





Step1

Step2



New Source File

Source File

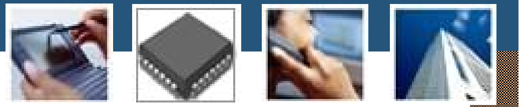
Create a new source file.

Source folder: hello/src

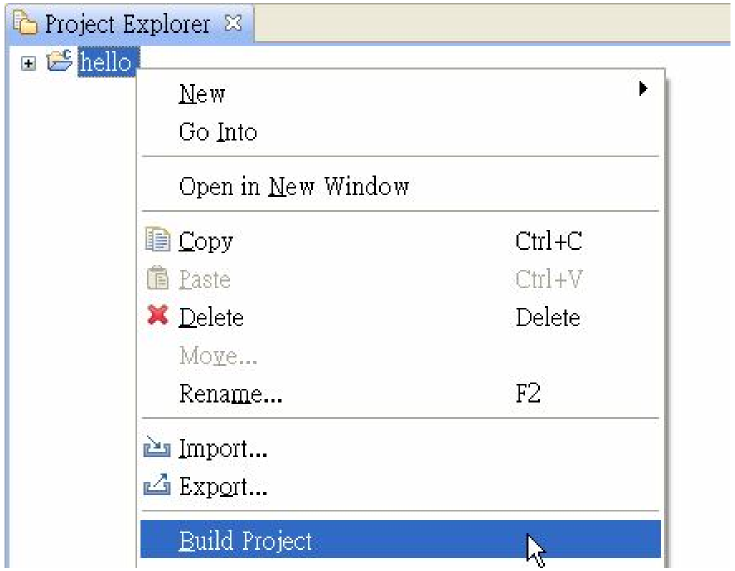
Source file: NewSourceFile.c

Template: Hello World

Build Project

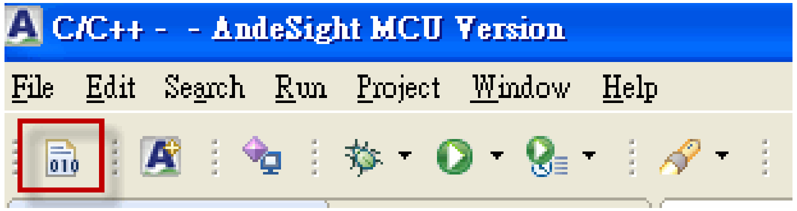


Right click the project folder
→ Build Project

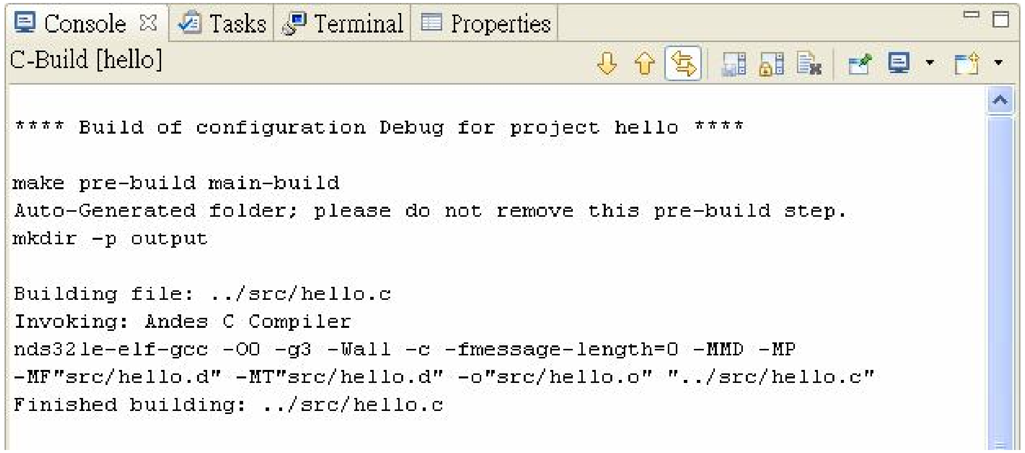


Or

click the build button on toolbar

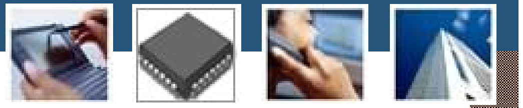


The building process in the "Console" view

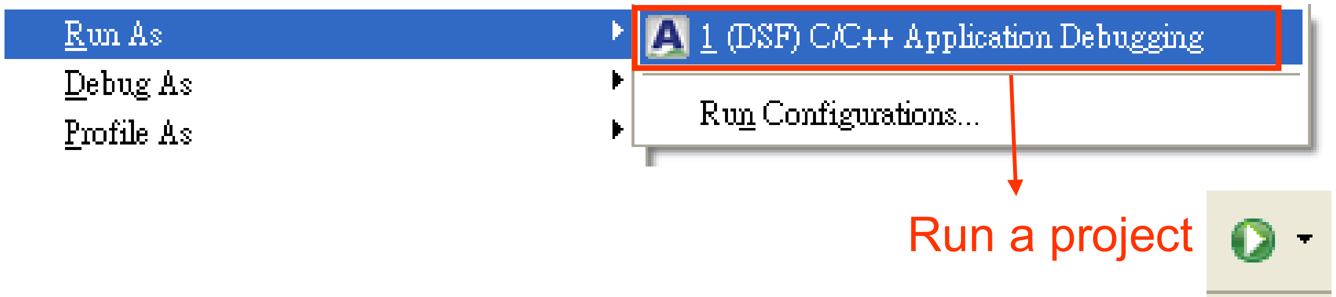


```
**** Build of configuration Debug for project hello ****

make pre-build main-build
Auto-Generated folder; please do not remove this pre-build step.
mkdir -p output

Building file: ../src/hello.c
Invoking: Andes C Compiler
nds32le-elf-gcc -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP
-MF"src/hello.d" -MT"src/hello.d" -o"src/hello.o" "../src/hello.c"
Finished building: ../src/hello.c
```

Run a Project

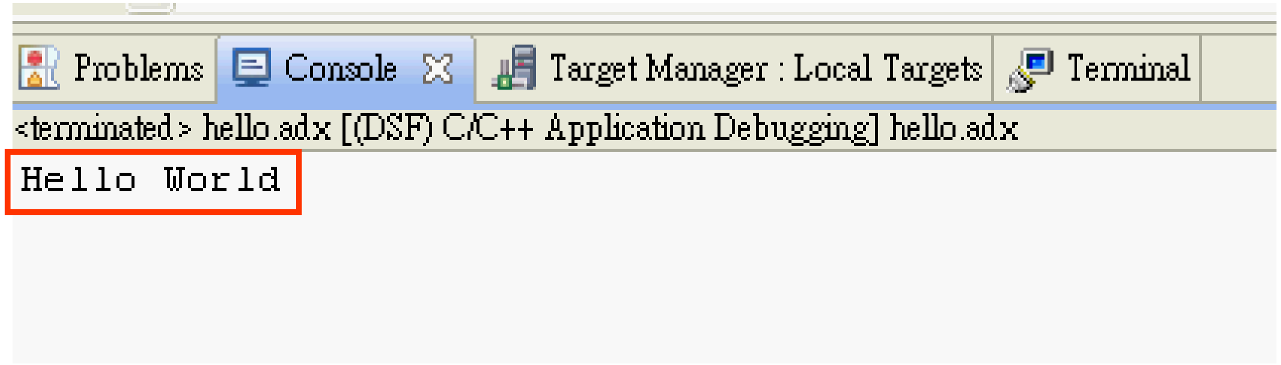


❖ Right click the project folder and select “Run As > (DSF) C/C++ Application Debugging”



Run a project

Result

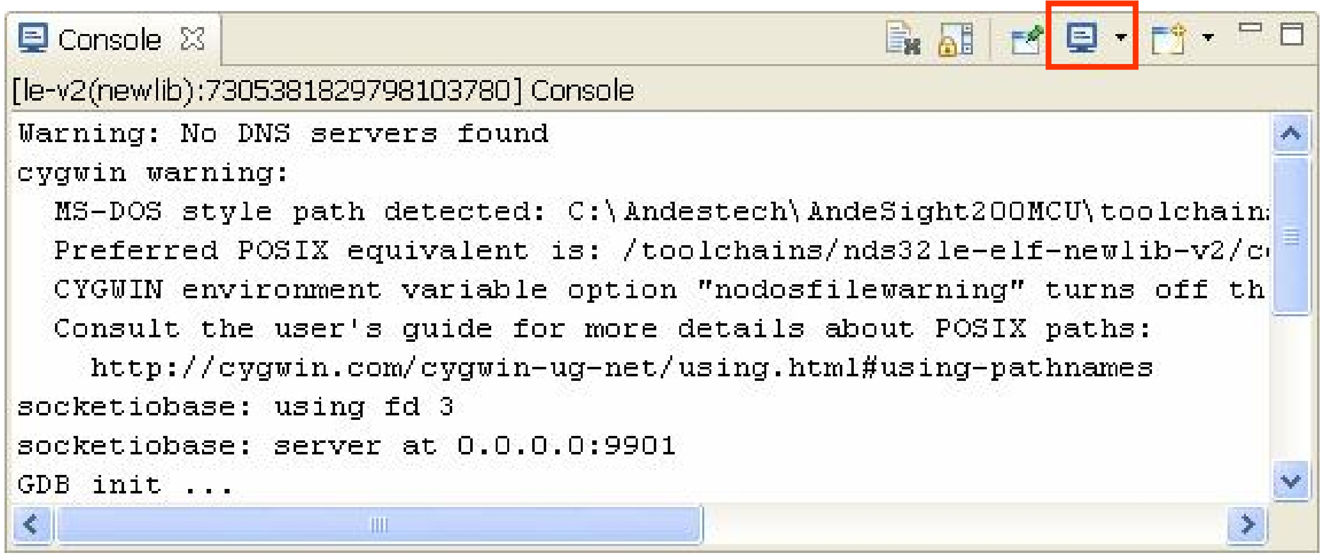


```
<terminated> hello.adx [(DSF) C/C++ Application Debugging] hello.adx
Hello World
```

Console View

❖ The Console View is a command line interface on AndeSight™.

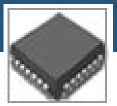
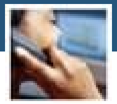
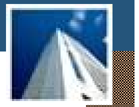
Users may switch between consoles

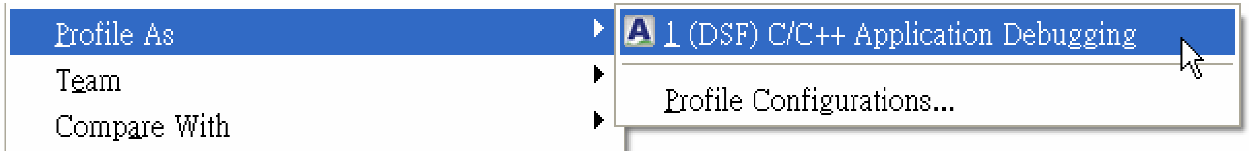


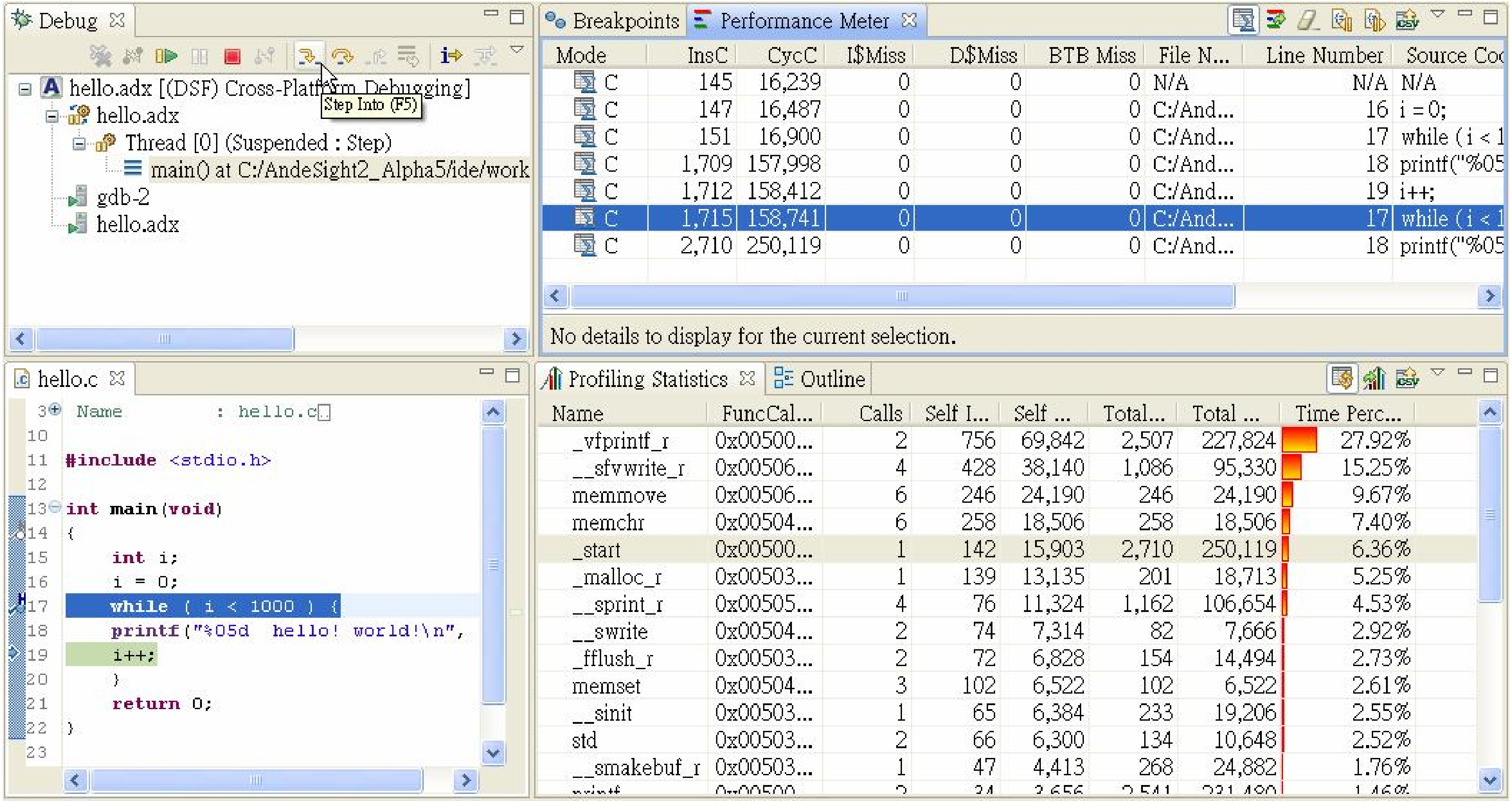
[le-v2(newlib):7305381829798103780] Console

```
Warning: No DNS servers found
cygwin warning:
  MS-DOS style path detected: C:\Andestech\AndeSight200MCU\toolchain
  Preferred POSIX equivalent is: /toolchains/nds32le-elf-newlib-v2/c
  CYGWIN environment variable option "nodosfilewarning" turns off th
  Consult the user's guide for more details about POSIX paths:
    http://cygwin.com/cygwin-ug-net/using.html#using-pathnames
socketiobase: using fd 3
socketiobase: server at 0.0.0.0:9901
GDB init ...
```

Profiling


Profile As
Team
Compare With
1 (DSF) C/C++ Application Debugging
Profile Configurations...



- hello.adx [(DSF) Cross-Platform Debugging]
  - hello.adx
    - Thread [0] (Suspended : Step)
      - main() at C:/AndeSight2_Alpha5/ide/work
  - gdb-2
  - hello.adx

Mode	InsC	CycC	I$Miss	D$Miss	BTB Miss	File N...	Line Number	Source Coc
C	145	16,239	0	0	0	N/A	N/A	N/A
C	147	16,487	0	0	0	C:/And...	16	i = 0;
C	151	16,900	0	0	0	C:/And...	17	while (i < 1
C	1,709	157,998	0	0	0	C:/And...	18	printf("%05
C	1,712	158,412	0	0	0	C:/And...	19	i++;
C	1,715	158,741	0	0	0	C:/And...	17	while (i < 1
C	2,710	250,119	0	0	0	C:/And...	18	printf("%05

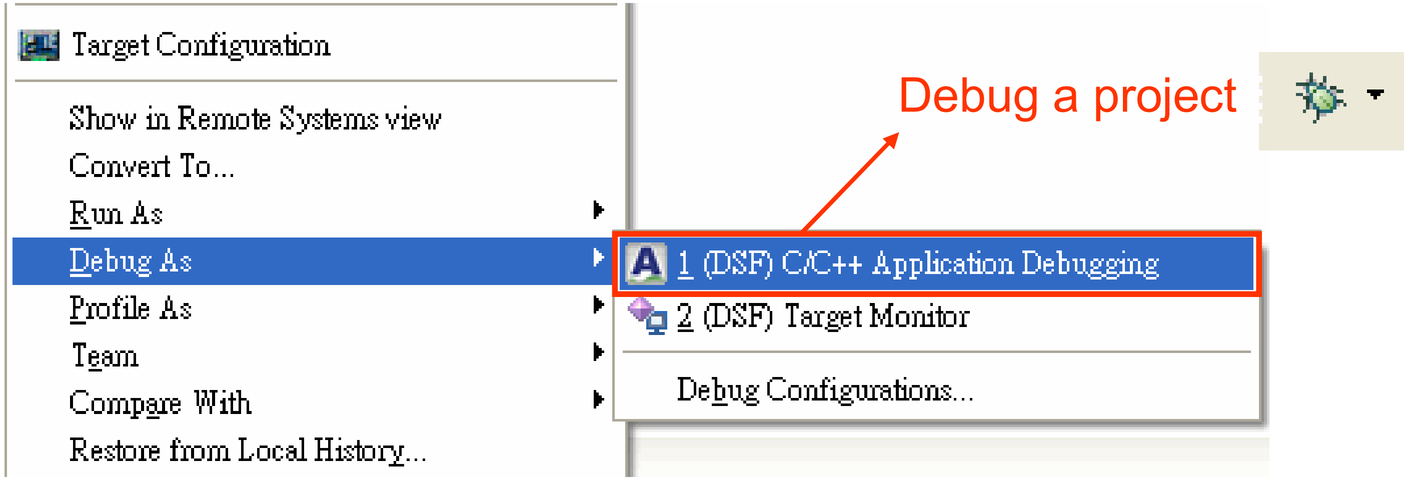
No details to display for the current selection.

```
 Name          : hello.c

#include <stdio.h>

int main(void)
{
    int i;
    i = 0;
    while ( i < 1000 ) {
    printf("%05d hello! world!\n",
    i++;
    }
    return 0;
}
```

Name	FuncCal...	Calls	Self I...	Self ...	Total...	Total ...	Time Perc...
_vfprintf_r	0x00500...	2	756	69,842	2,507	227,824	27.92%
__sfvwrite_r	0x00506...	4	428	38,140	1,086	95,330	15.25%
memmove	0x00506...	6	246	24,190	246	24,190	9.67%
memchr	0x00504...	6	258	18,506	258	18,506	7.40%
_start	0x00500...	1	142	15,903	2,710	250,119	6.36%
_malloc_r	0x00503...	1	139	13,135	201	18,713	5.25%
__sprint_r	0x00505...	4	76	11,324	1,162	106,654	4.53%
__swrite	0x00504...	2	74	7,314	82	7,666	2.92%
_fflush_r	0x00503...	2	72	6,828	154	14,494	2.73%
memset	0x00504...	3	102	6,522	102	6,522	2.61%
__sinit	0x00503...	1	65	6,384	233	19,206	2.55%
std	0x00503...	2	66	6,300	134	10,648	2.52%
__smakebuf_r	0x00503...	1	47	4,413	268	24,882	1.76%
printf	0x00500...	2	34	3,656	2,541	231,480	1.46%

Debug a Project

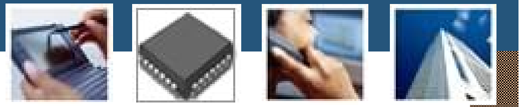
❖ Right click the project folder and select "Debug As > (DSF) C/C++ Application Debugging"



Debug a project

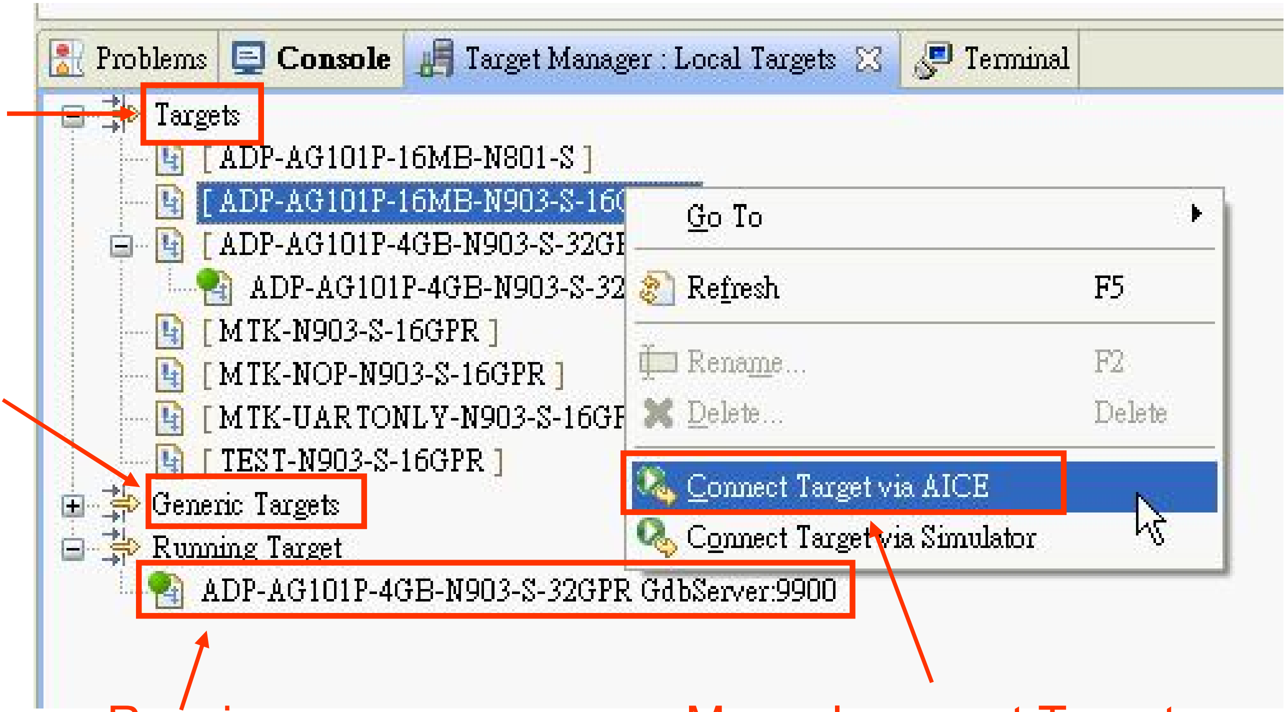
- Stop the simulator, demo AICE
  - AICE plug-in detect
  - Target Monitor
  - Terminal View
  - Run and Debug on EVB (via AICE)
  - Target manipulation
  - How to change the toolchain

Target Management



(1) (Pre-defined) Targets

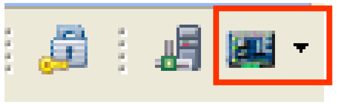
(2) Generic Targets



Running

Manual connect Target

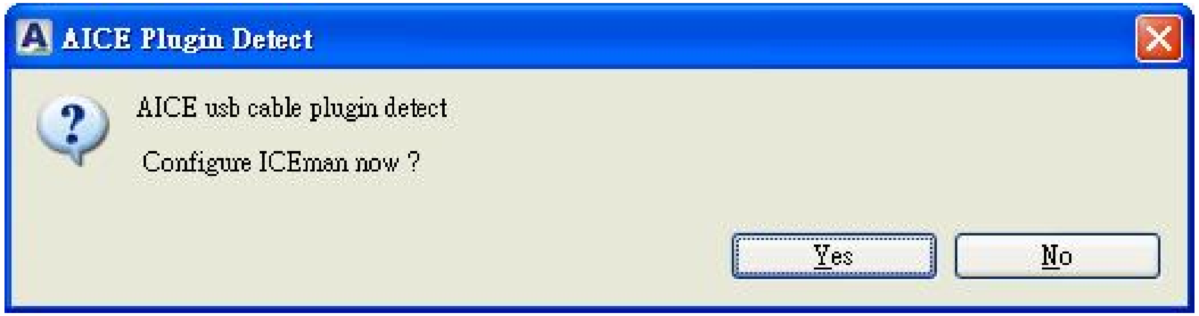
- Available
- Running
- Busy



Running-target icon at the lower right corner of Andesight

AICE Plugin detect

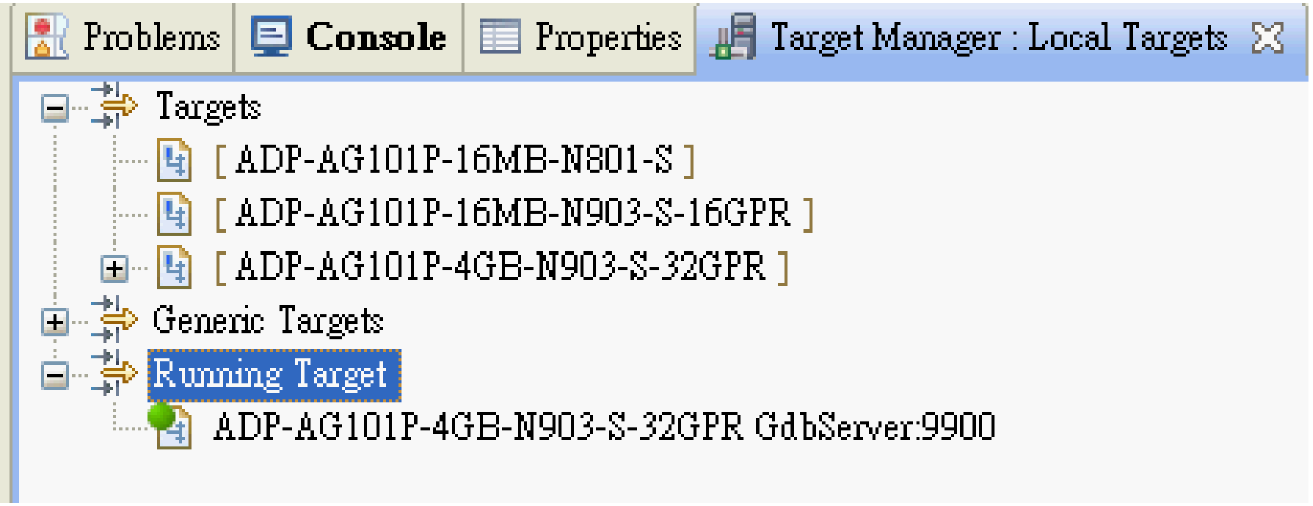
When we connect the AICE to PC, the target start automatically.



AICE Plugin Detect

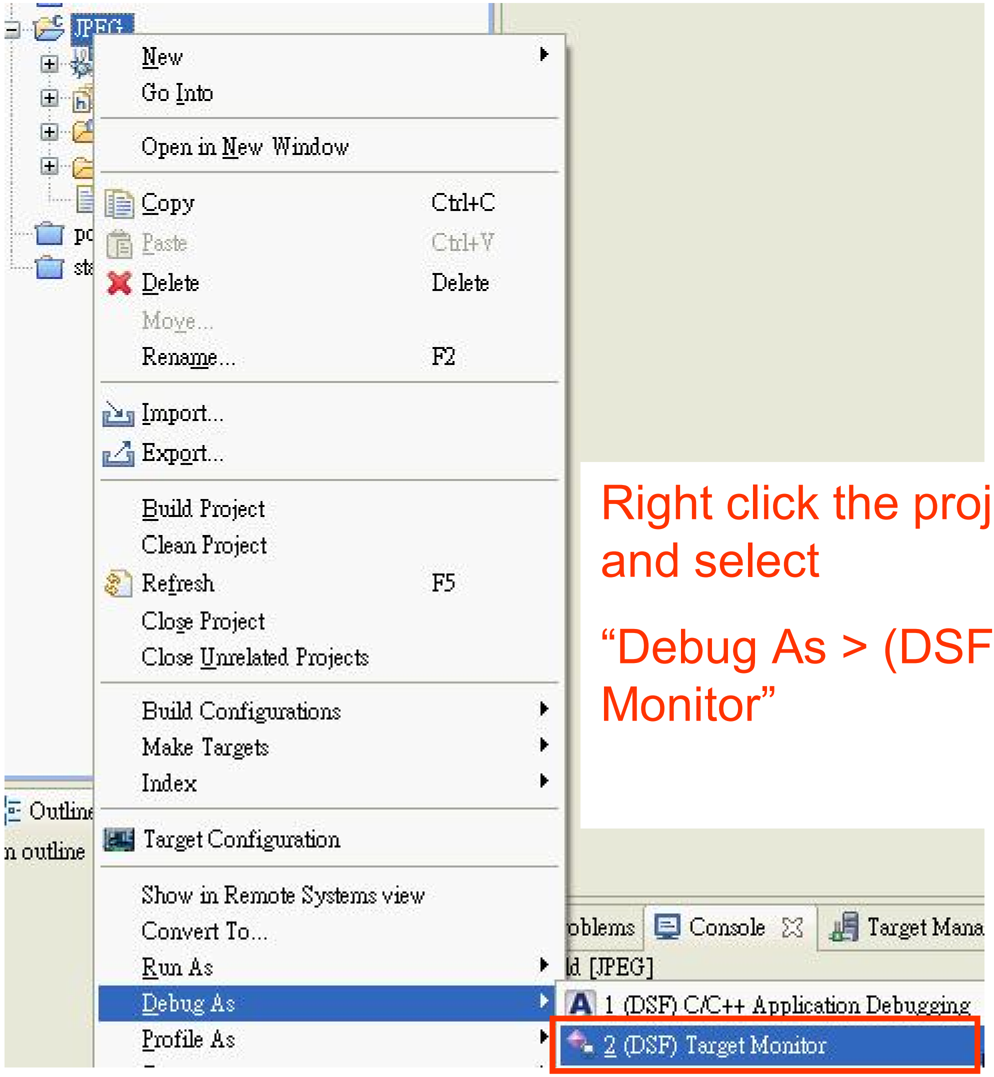
AICE usb cable plugin detect

Configure ICEman now ?



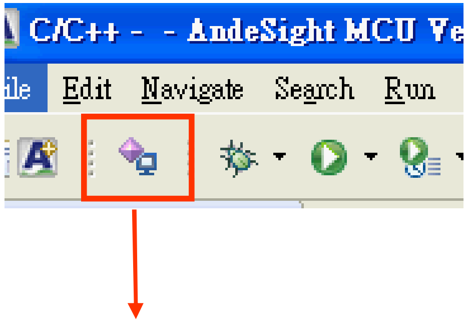
```
Targets
    [ ADP-AG101P-16MB-N801-S ]
    [ ADP-AG101P-16MB-N903-S-16GPR ]
  + [ ADP-AG101P-4GB-N903-S-32GPR ]
+ Generic Targets
- Running Target
    ADP-AG101P-4GB-N903-S-32GPR GdbServer:9900
```

Monitor Target (1)



Right click the project folder and select

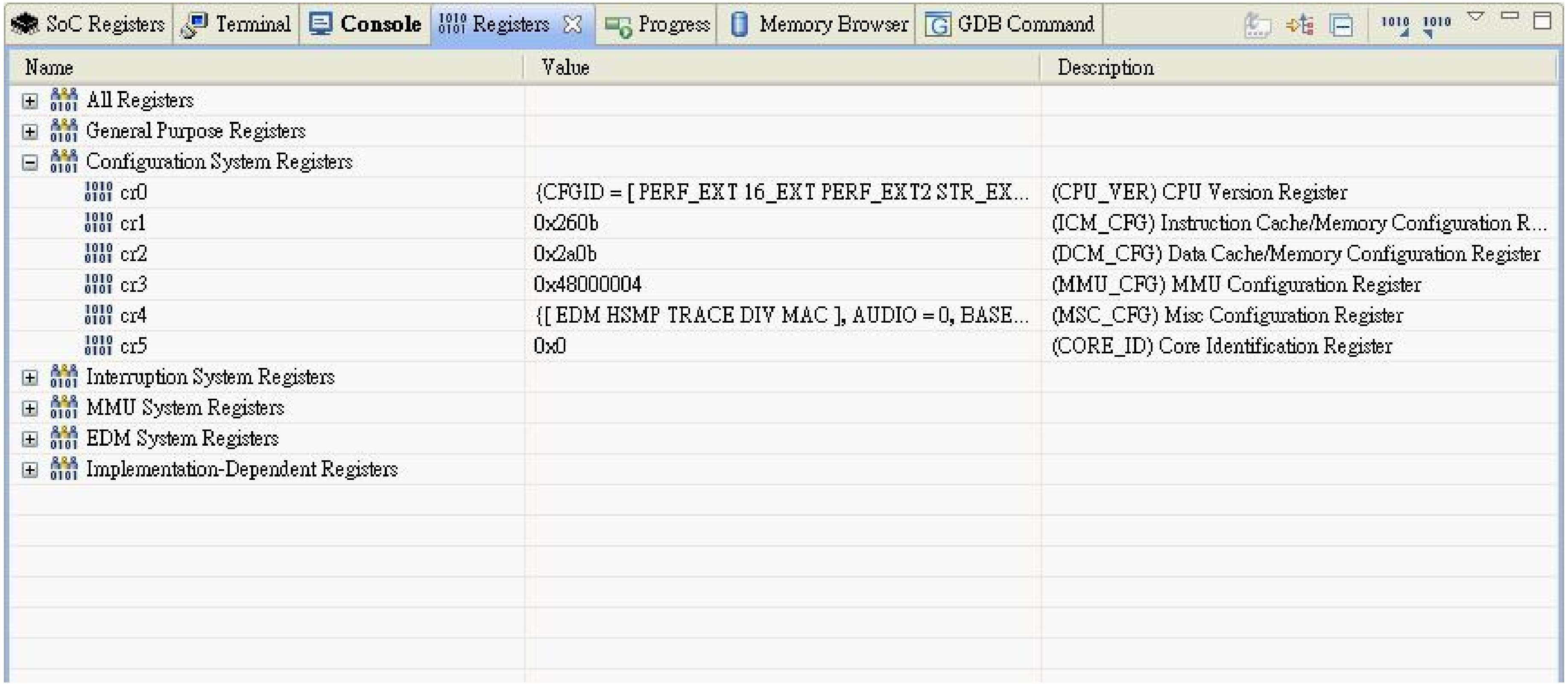
“Debug As > (DSF) Target Monitor”



Or click this button

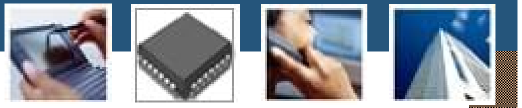
Monitor Target (2)

Monitor Target function for users to examine the default debug values before commencing a debug session

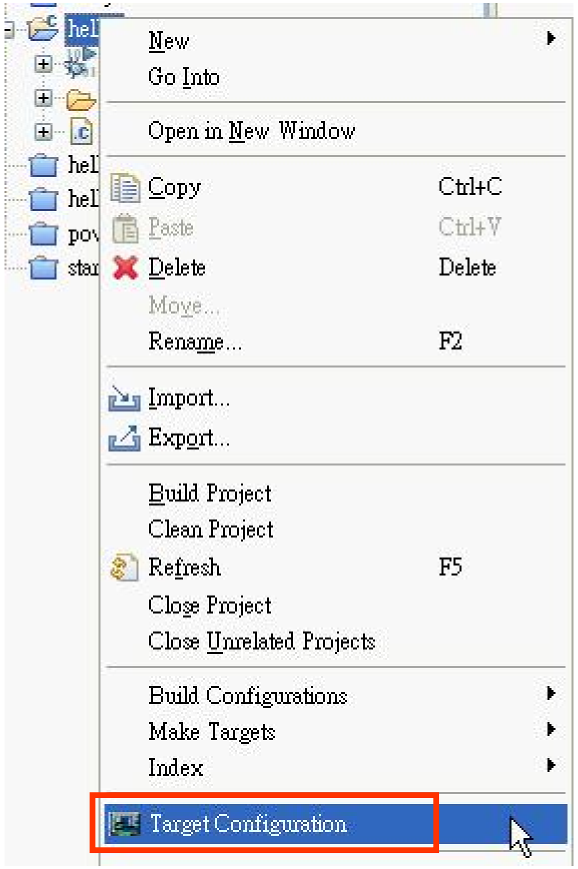


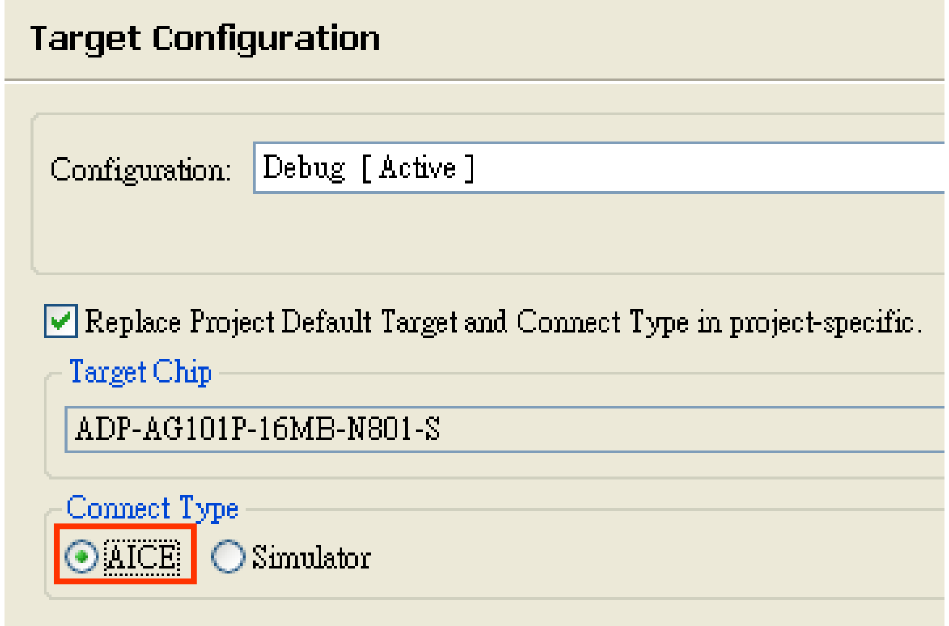
Name	Value	Description
⊞ All Registers		
⊞ General Purpose Registers		
⊟ Configuration System Registers		
cr0	{CFGID = [PERF_EXT 16_EXT PERF_EXT2 STR_EX...	(CPU_VER) CPU Version Register
cr1	0x260b	(ICM_CFG) Instruction Cache/Memory Configuration R...
cr2	0x2a0b	(DCM_CFG) Data Cache/Memory Configuration Register
cr3	0x48000004	(MMU_CFG) MMU Configuration Register
cr4	{[EDM HSMP TRACE DIV MAC], AUDIO = 0, BASE...	(MSC_CFG) Misc Configuration Register
cr5	0x0	(CORE_ID) Core Identification Register
⊞ Interruption System Registers		
⊞ MMU System Registers		
⊞ EDM System Registers		
⊞ Implementation-Dependent Registers		

Change Target



Set the Connect Type of Target Configuration as “AICE”.





Target Configuration

Configuration: Debug [Active]

☑ Replace Project Default Target and Connect Type in project-specific.

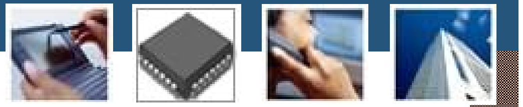
Target Chip

ADP-AG101P-16MB-N801-S

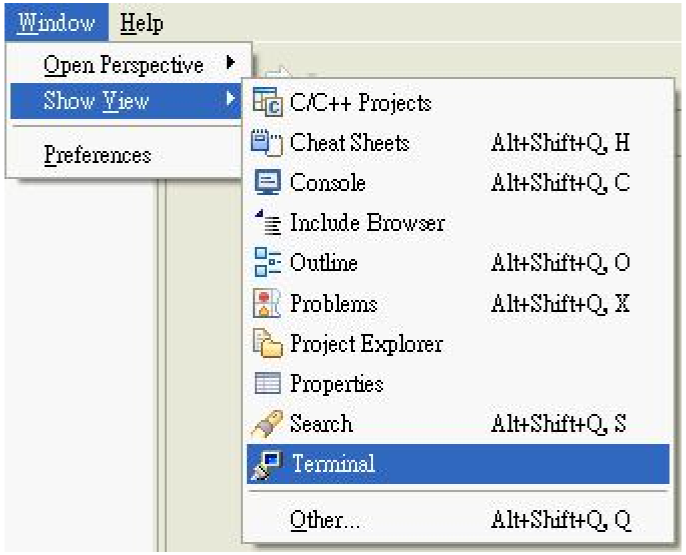
Connect Type

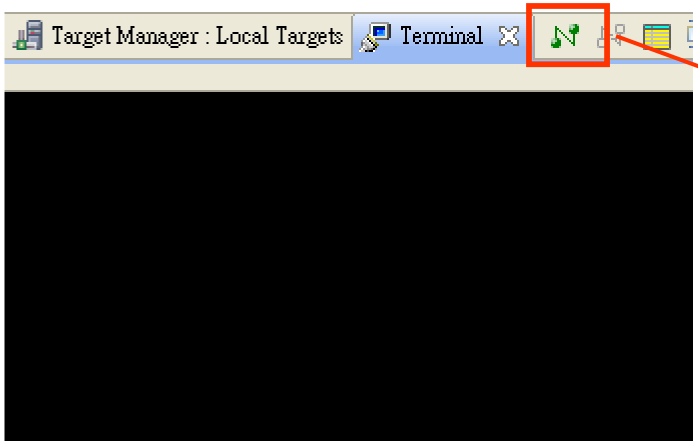
● AICE ○ Simulator

Terminal View



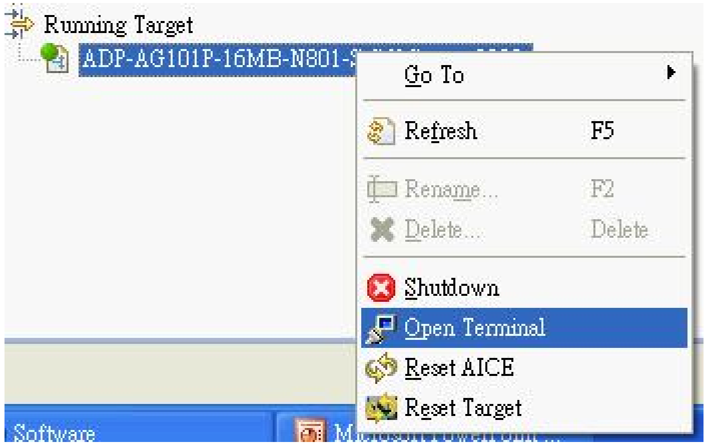
Method 1: from toolbar

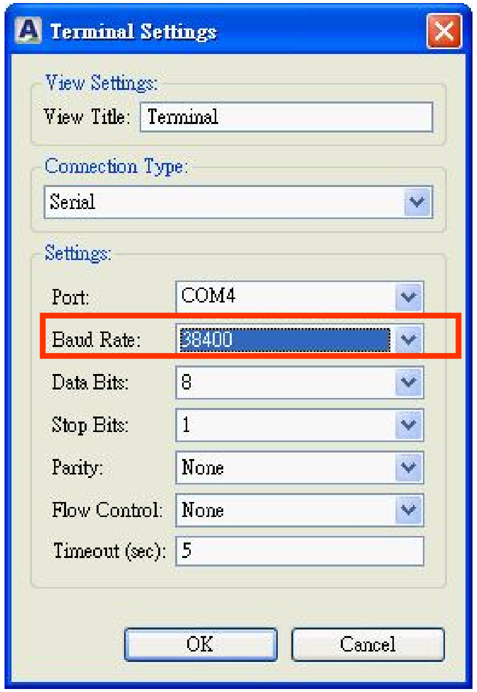




Connect

Method 2: from target




Terminal Settings

View Settings:

View Title: Terminal

Connection Type:

Serial

Settings:

Port: COM4

Baud Rate: 38400

Data Bits: 8

Stop Bits: 1

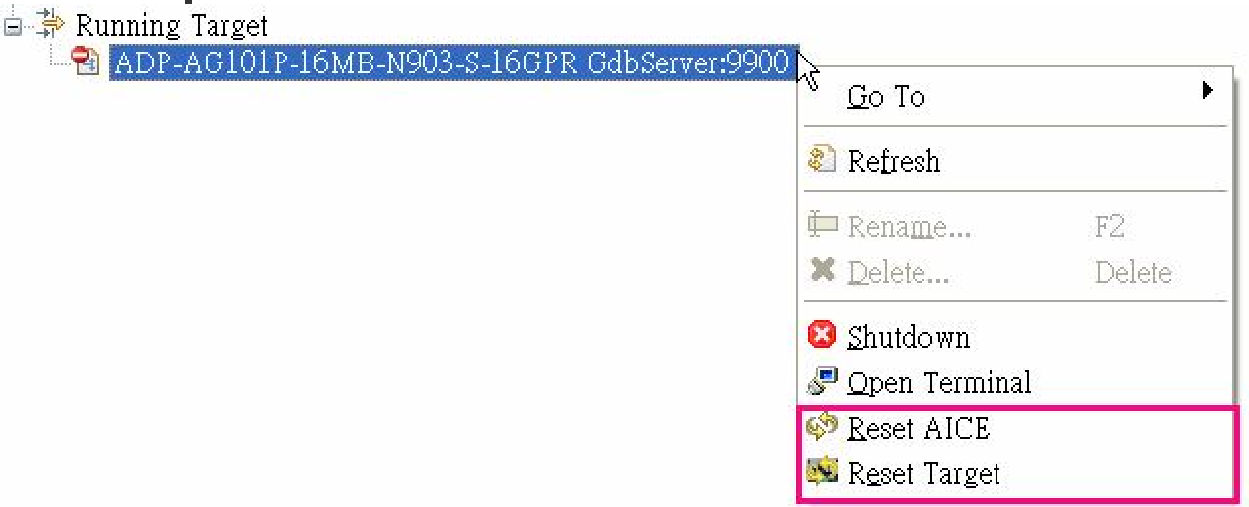
Parity: None

Flow Control: None

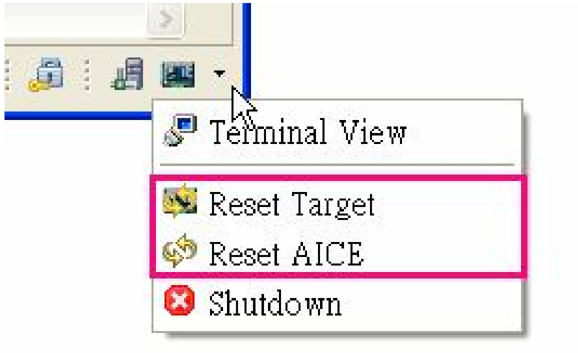
Timeout (sec): 5

How to Reset/Shutdown a Target/AICE™

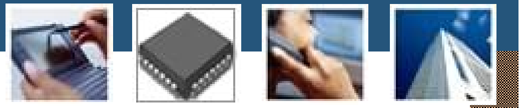
1. Select a running AICE™ target and right click to evoke a pull-down menu



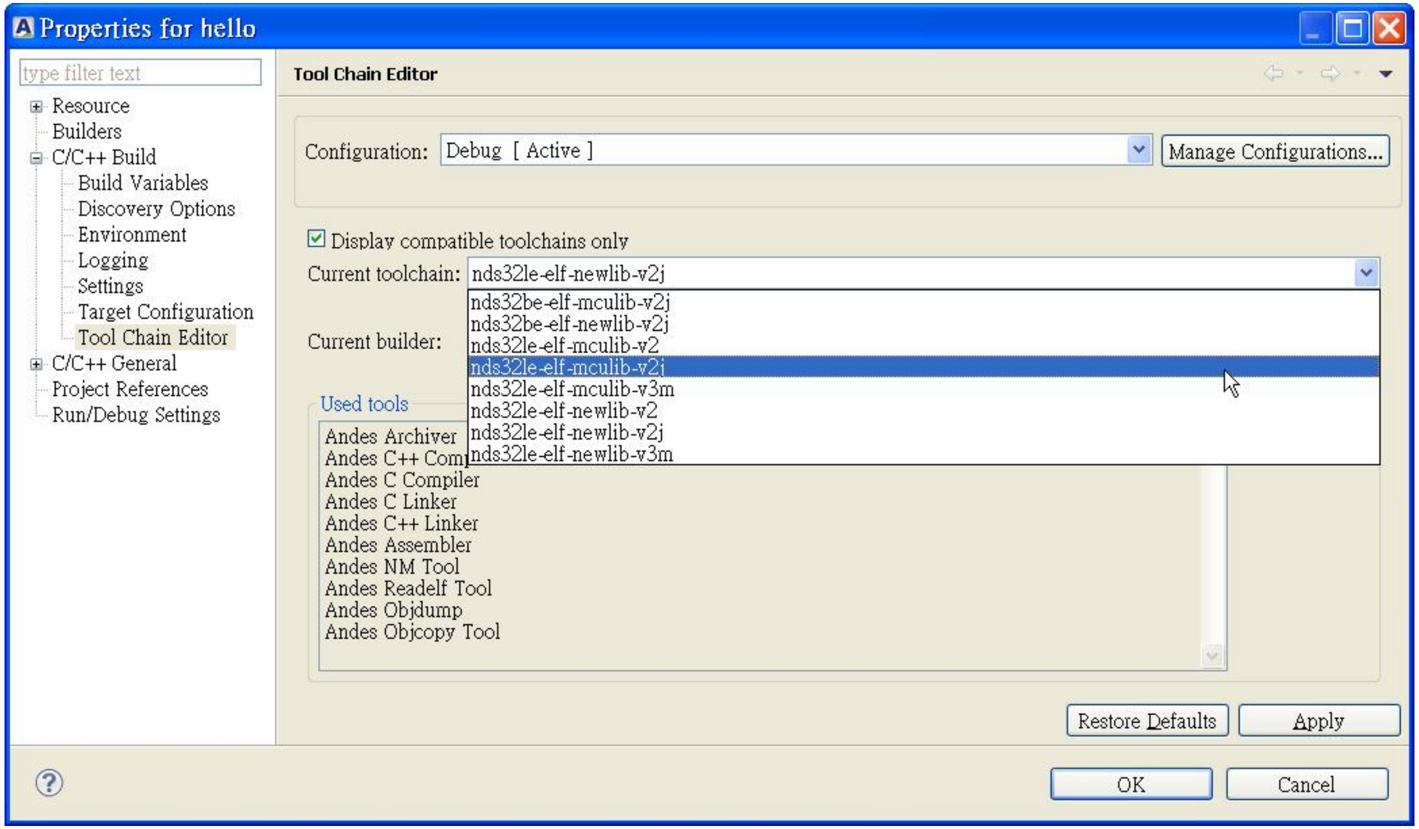
2. Shortcut at the lower right corner of AndeSight™



Select/Switch a Toolchain



- Right click the project folder and select "Properties" to invoke the project properties dialog. Select "C/C++ Build > Tool Chain Editor"



Configuration: Debug [Active]

☑ Display compatible toolchains only

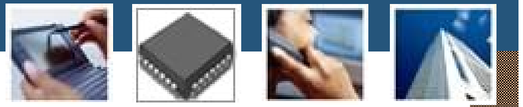
Current toolchain: nds32le-elf-newlib-v2j

```
nds32be-elf-mculib-v2j
nds32be-elf-newlib-v2j
nds32le-elf-mculib-v2
nds32le-elf-mculib-v2j
nds32le-elf-mculib-v3m
nds32le-elf-newlib-v2
nds32le-elf-newlib-v2j
nds32le-elf-newlib-v3m
```

Current builder:

Used tools

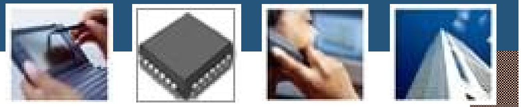
```
Andes Archiver
Andes C++ Com
Andes C Compiler
Andes C Linker
Andes C++ Linker
Andes Assembler
Andes NM Tool
Andes Readelf Tool
Andes Objdump
Andes Objcopy Tool
```

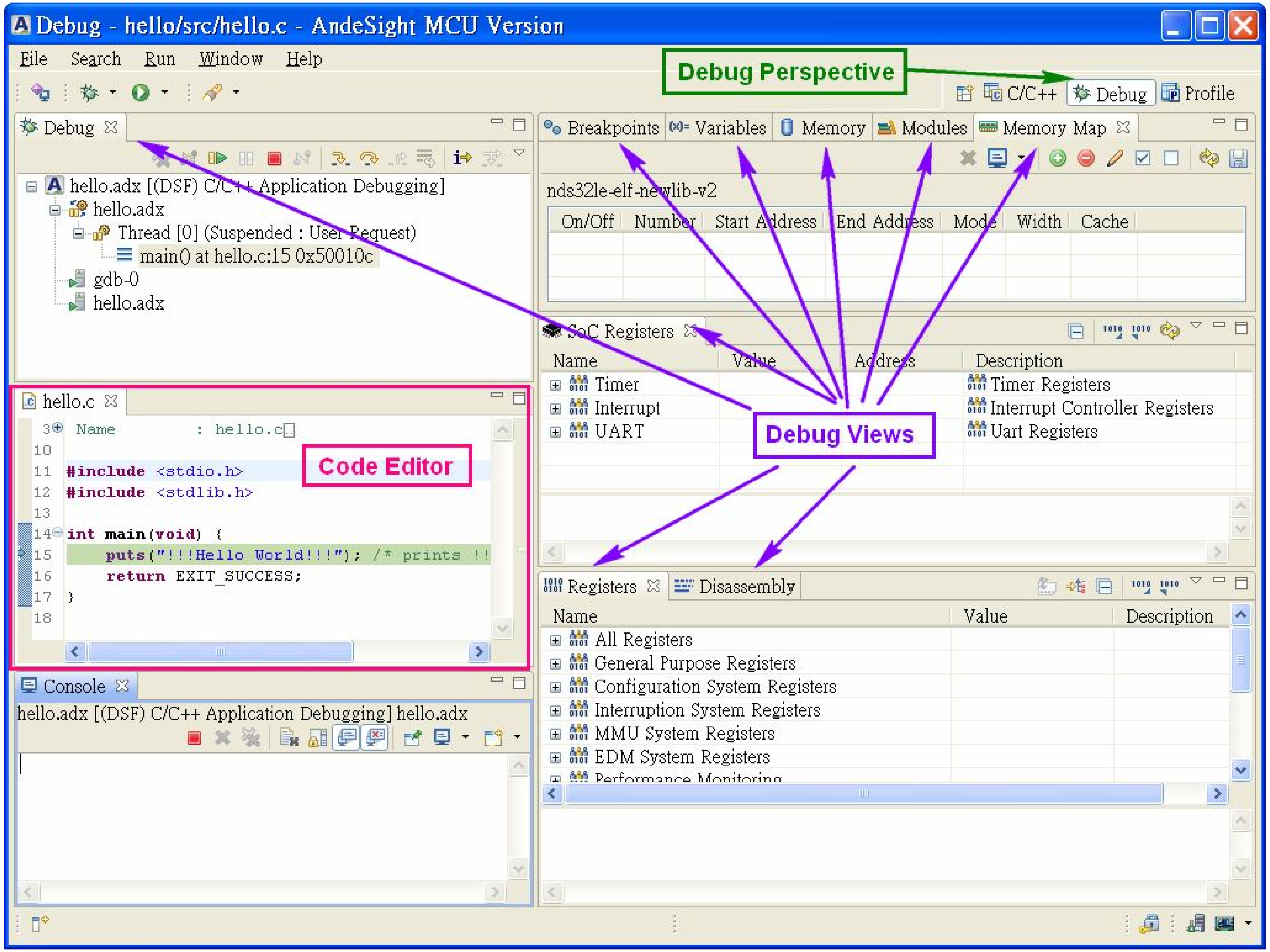


❖ Debug Perspective – (using JPEG demo)

- Debug assembly
- Memory View
- Memory Browser View
- Register View
- SOC register View
- GDB command View

The Debug Perspective





hello.adx [(DSF) C/C++ Application Debugging]
- hello.adx
  - Thread [0] (Suspended : User Request)
    - main() at hello.c:15 0x50010c
- gdb-0
- hello.adx

```
 Name          : hello.c

#include <stdio.h>
#include <stdlib.h>

int main(void) {
    puts("!!!Hello World!!!"); /* prints !!
    return EXIT_SUCCESS;
}
```

nds32le-elf-newlib-v2

On/Off	Number	Start Address	End Address	Mode	Width	Cache

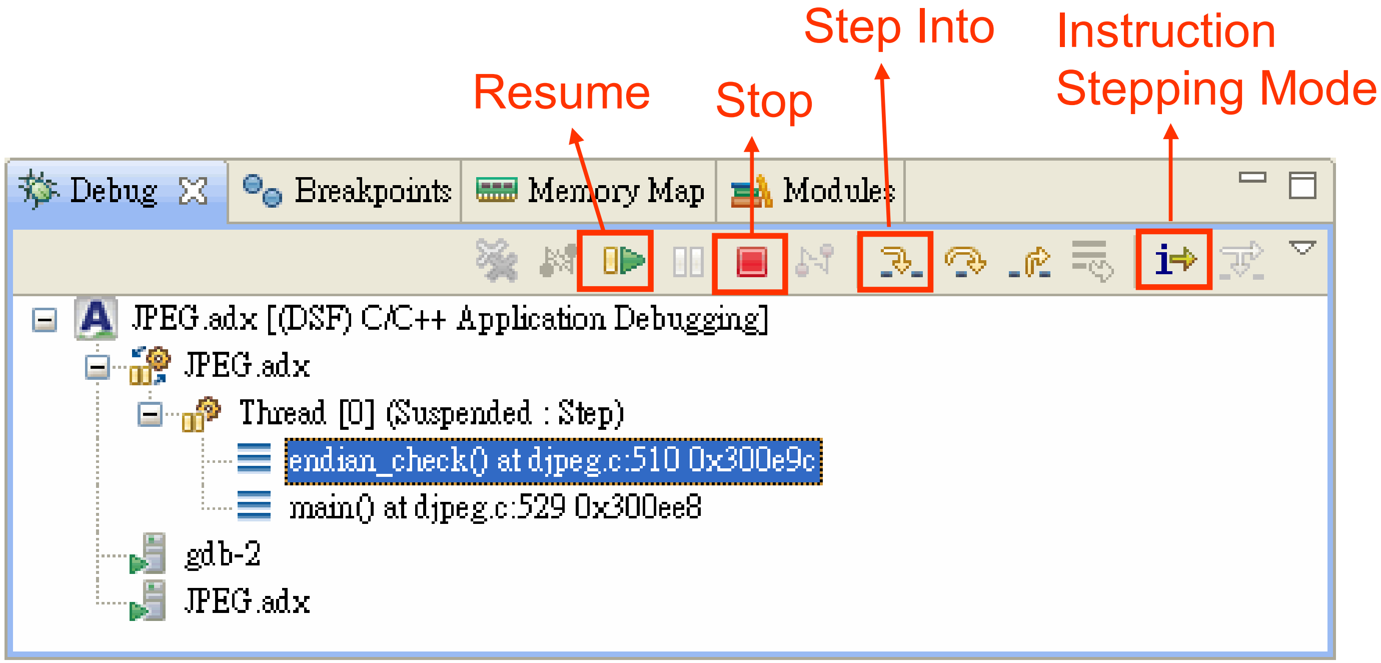
Name	Value	Address	Description
Timer			Timer Registers
Interrupt			Interrupt Controller Registers
UART			Uart Registers

hello.adx [(DSF) C/C++ Application Debugging] hello.adx

Name	Value	Description
All Registers		
General Purpose Registers		
Configuration System Registers		
Interruption System Registers		
MMU System Registers		
EDM System Registers		
Performance Monitoring		

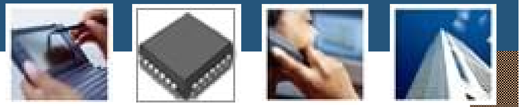
Debug View

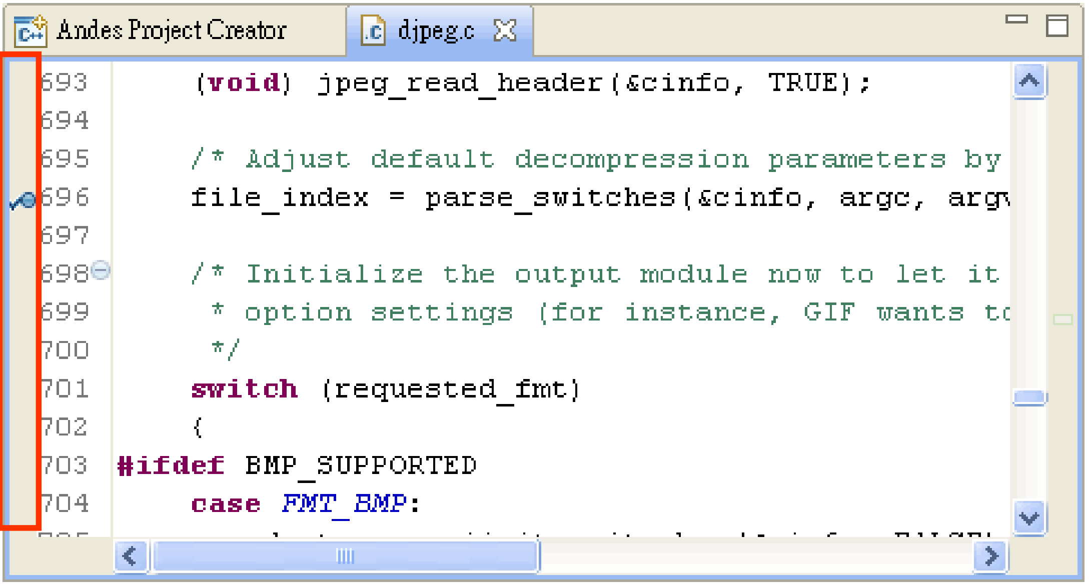
- This view displays the execution stack outlining suspended and active threads for each target being debugged



```
JPEG.adx [(DSF) C/C++ Application Debugging]
  JPEG.adx
    Thread [0] (Suspended : Step)
      endian_check() at djpeg.c:510 0x300e9c
      main() at djpeg.c:529 0x300ee8
  gdb-2
  JPEG.adx
```

How to Set Breakpoints

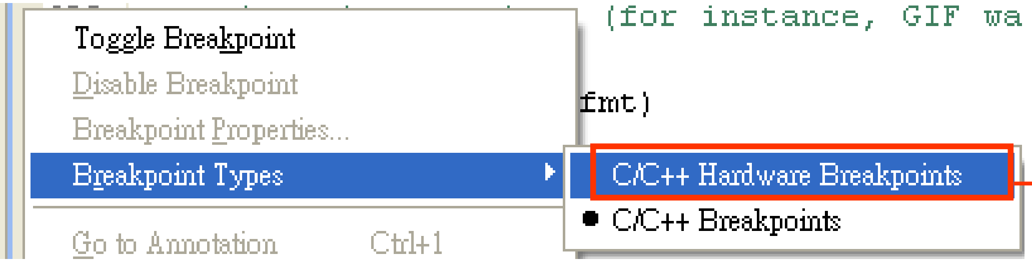




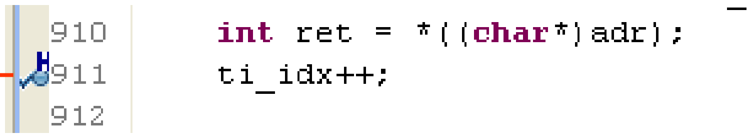
```
    (void) jpeg_read_header(&cinfo, TRUE);

    /* Adjust default decompression parameters by
    file_index = parse_switches(&cinfo, argc, argv

    /* Initialize the output module now to let it
     * option settings (for instance, GIF wants to
     */
    switch (requested_fmt)
    {
#ifdef BMP_SUPPORTED
    case FMT_BMP:
```

Double click the row header to set breakpoint

Right click the row header can change breakpoint type



Change to hardware Breakpoints

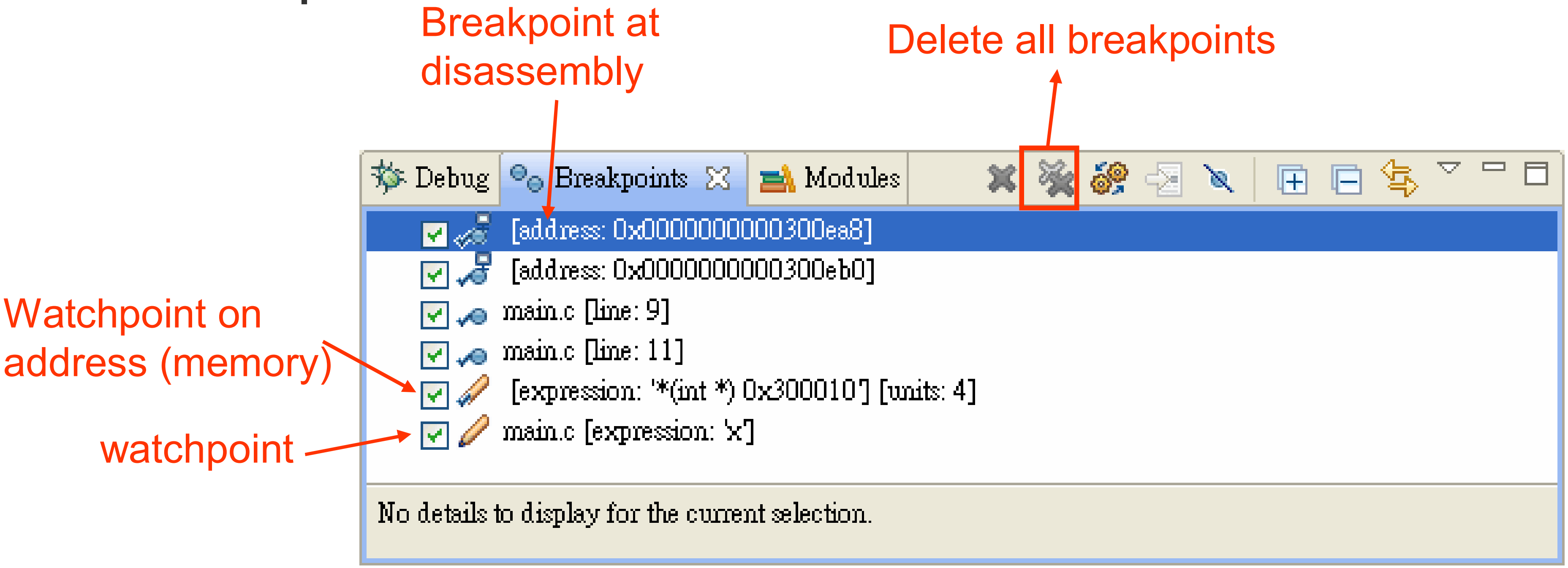


```
int ret = *((char*)adr);
ti_idx++;
```

hardware Breakpoint

Breakpoints View

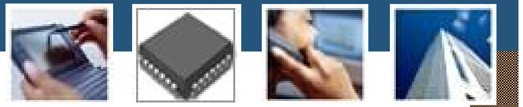
- The full path for the breakpoint set is displayed in Breakpoints View.

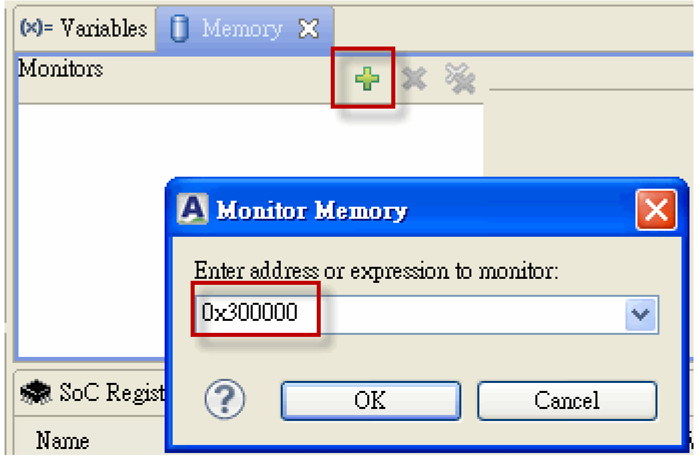


☑ [address: 0x0000000000300ea8]
☑ [address: 0x0000000000300eb0]
☑ main.c [line: 9]
☑ main.c [line: 11]
☑ [expression: '*(int *) 0x300010'] [units: 4]
☑ main.c [expression: 'x']

No details to display for the current selection.

Memory View (1)

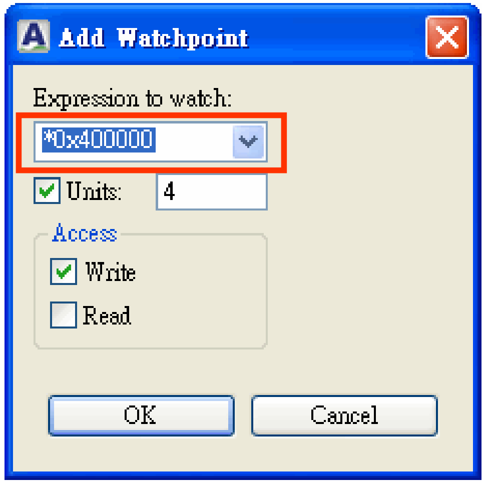




Monitor Memory

Enter address or expression to monitor:

0x300000



Add Watchpoint

Expression to watch: *0x400000

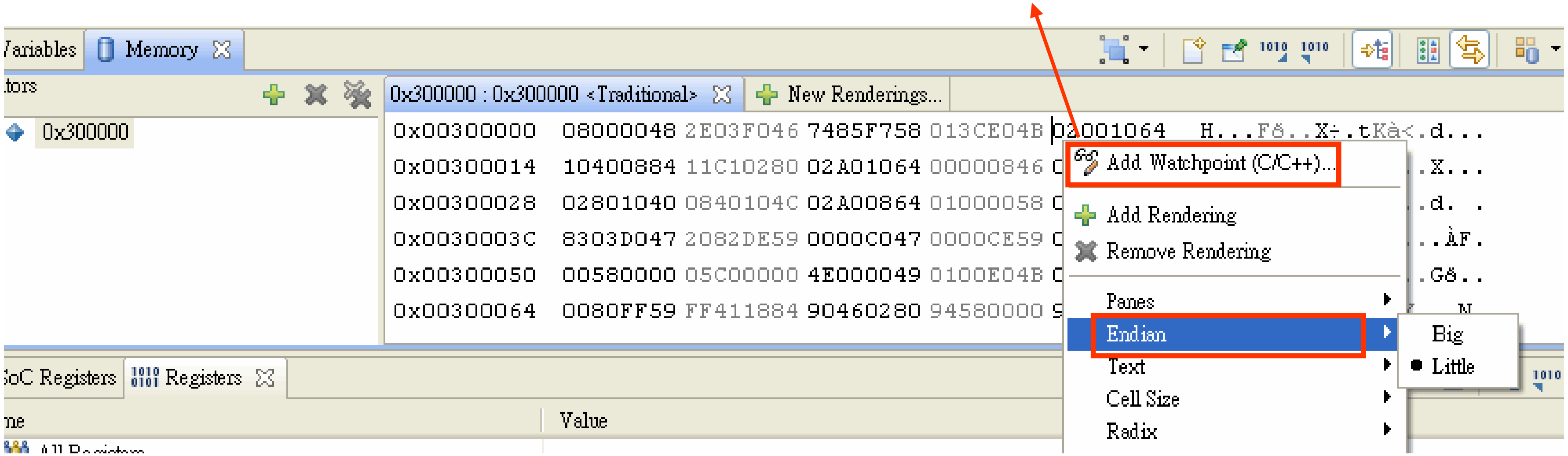
☑ Units: 4

Access

☑ Write

□ Read

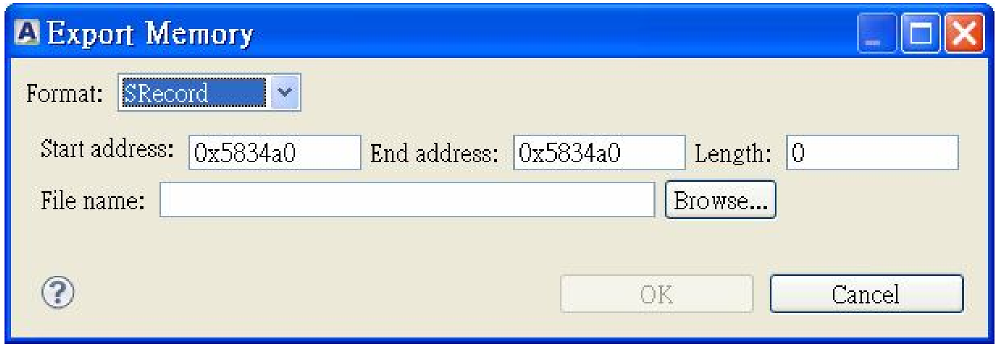
This is not correct. We fill in *0x502d60 or *(int*)0x502d60 manually



```
0x00300000  08000048 2E03F046 7485F758 013CE04B 02001064  H...Fő..X÷.tKà<.d...
0x00300014  10400884 11C10280 02A01064 00000846 C         .X...
0x00300028  02801040 0840104C 02A00864 01000058 C         .d. .
0x0030003C  8303D047 2082DE59 0000C047 0000CE59 C         ..ÀF.
0x00300050  00580000 05C00000 4E000049 0100E04B C         .Gő..
0x00300064  0080FF59 FF411884 90460280 94580000 9         N
```

Memory View (2)

Export Memory Values

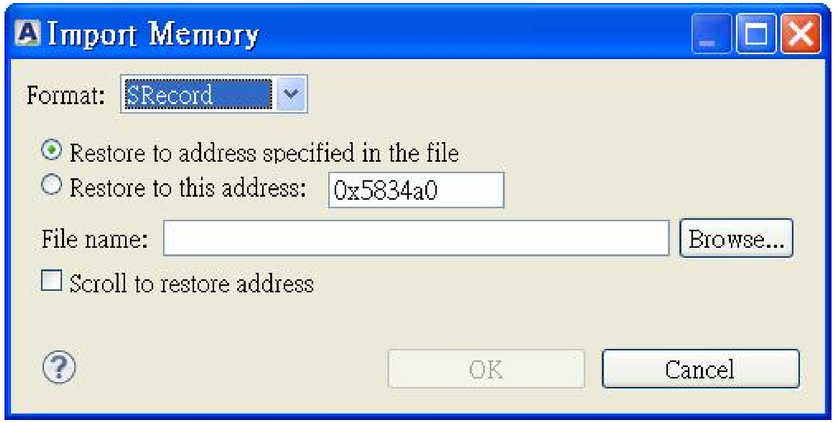


Format: SRecord

Start address: 0x5834a0 End address: 0x5834a0 Length: 0

File name:

Import Memory Values



Format: SRecord

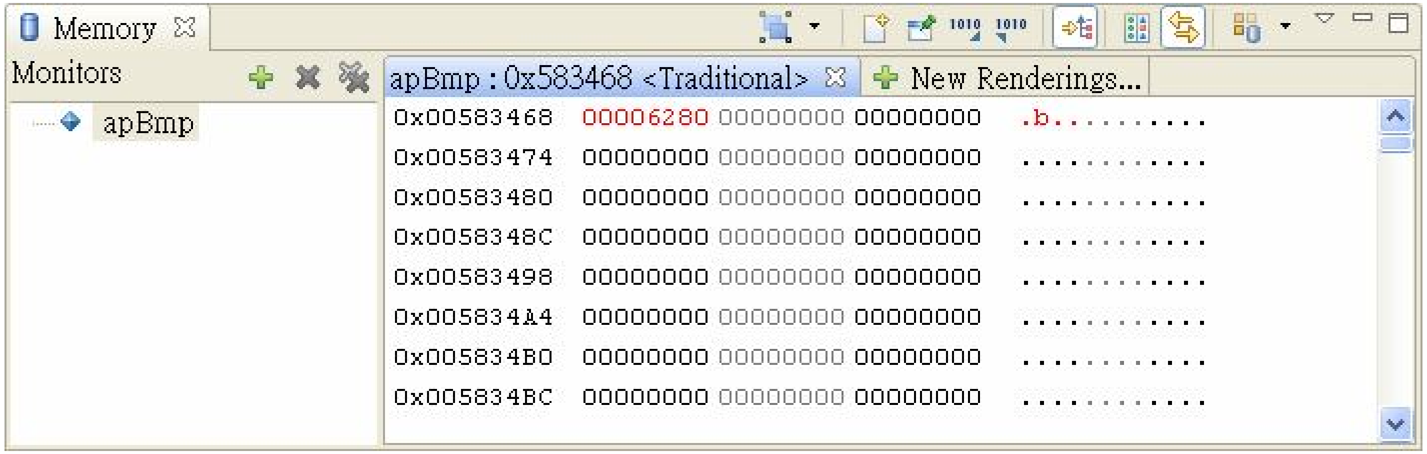
● Restore to address specified in the file

○ Restore to this address: 0x5834a0

File name:

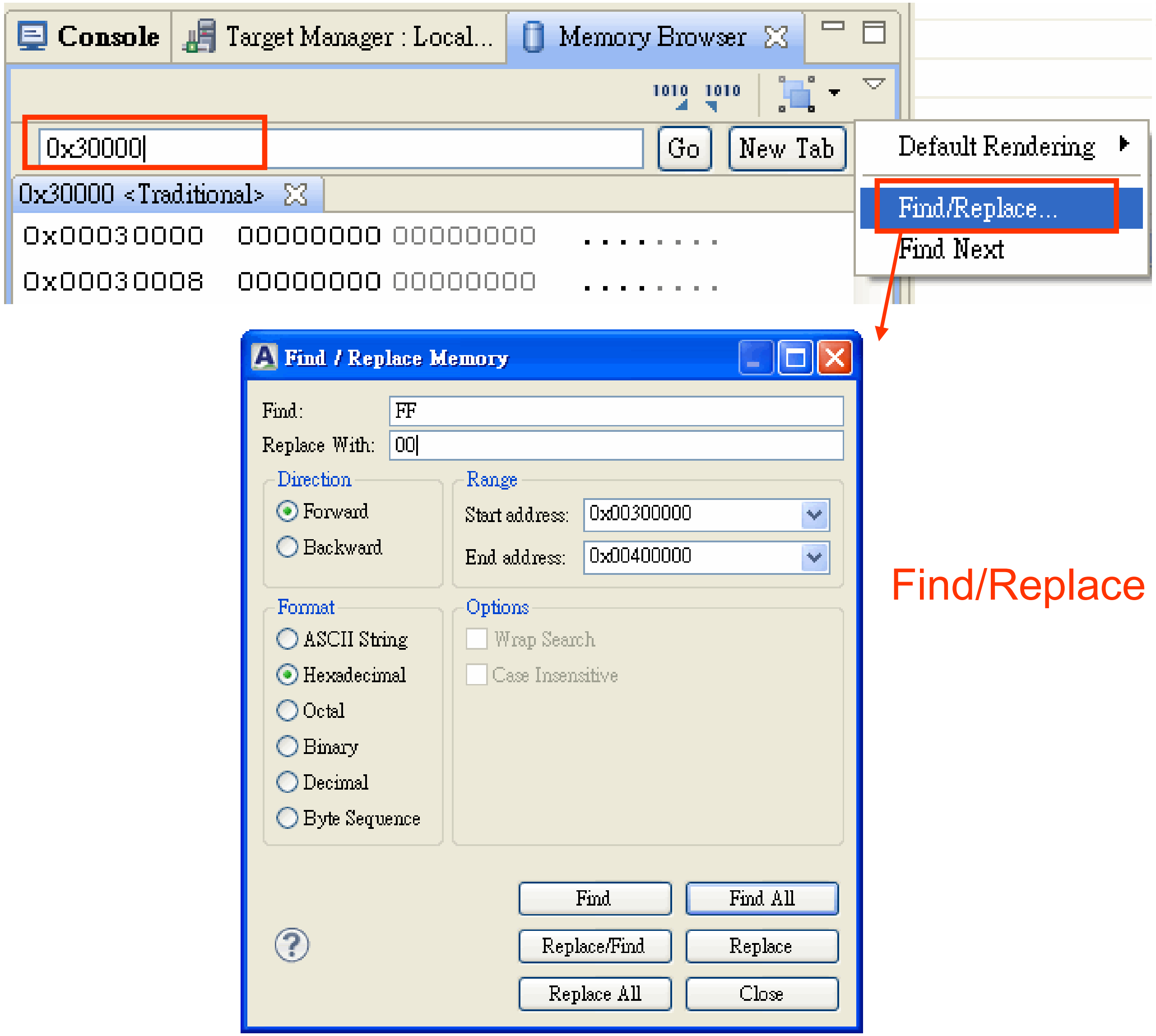
□ Scroll to restore address

Users may choose memory source either from CPU or BUS with this utility.
The deviation is marked in red.



```
0x00583468   00006280 00000000 00000000   .b..........
0x00583474   00000000 00000000 00000000   ............
0x00583480   00000000 00000000 00000000   ............
0x0058348C   00000000 00000000 00000000   ............
0x00583498   00000000 00000000 00000000   ............
0x005834A4   00000000 00000000 00000000   ............
0x005834B0   00000000 00000000 00000000   ............
0x005834BC   00000000 00000000 00000000   ............
```

Memory Browser View



0x30000

0x30000 <Traditional>

```
0x00030000   00000000 00000000   ........
0x00030008   00000000 00000000   ........
```

Find / Replace Memory

Find: FF

Replace With: 00

Direction
● Forward
○ Backward

Range
Start address: 0x00300000
End address: 0x00400000

Format
○ ASCII String
● Hexadecimal
○ Octal
○ Binary
○ Decimal
○ Byte Sequence

Options
□ Wrap Search
□ Case Insensitive

Find/Replace Memory

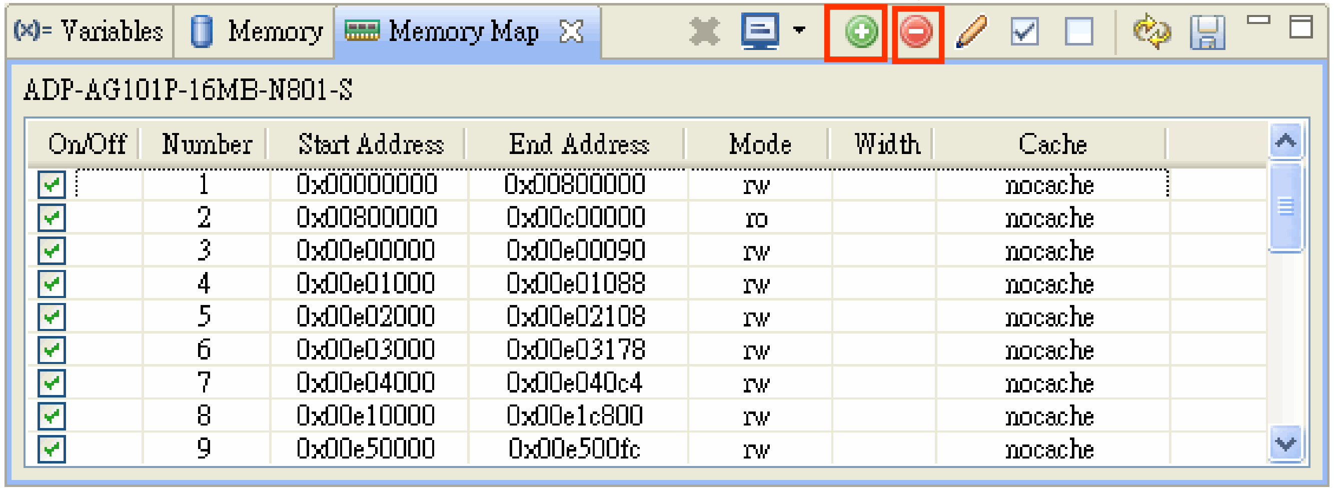
Confidential

Memory Map View

- Memory Map View is an interface for users to set or examine the memory regions and memory attributes (read-only, write-only, read/write) of the targets.

Add Memory Regions

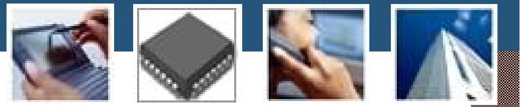
Delete Memory regions

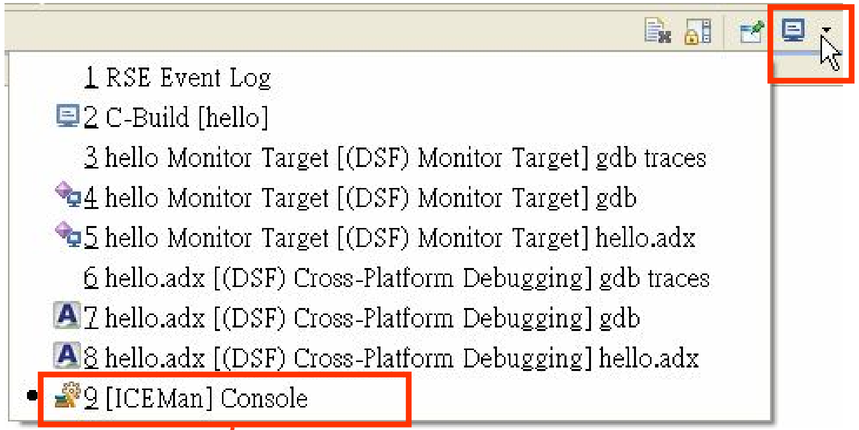


ADP-AG101P-16MB-N801-S

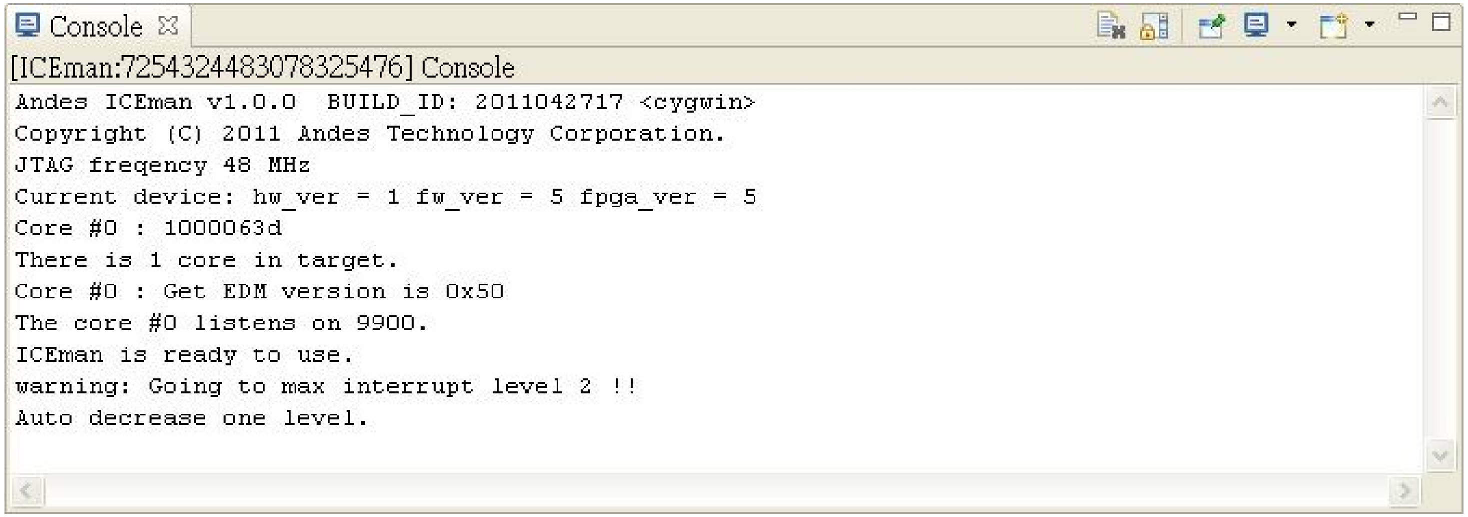
On/Off	Number	Start Address	End Address	Mode	Width	Cache
☑	1	0x00000000	0x00800000	rw		nocache
☑	2	0x00800000	0x00c00000	ro		nocache
☑	3	0x00e00000	0x00e00090	rw		nocache
☑	4	0x00e01000	0x00e01088	rw		nocache
☑	5	0x00e02000	0x00e02108	rw		nocache
☑	6	0x00e03000	0x00e03178	rw		nocache
☑	7	0x00e04000	0x00e040c4	rw		nocache
☑	8	0x00e10000	0x00e1c800	rw		nocache
☑	9	0x00e50000	0x00e500fc	rw		nocache

ICEman Console View





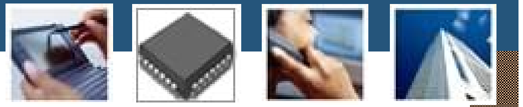
1 RSE Event Log
2 C-Build [hello]
3 hello Monitor Target [(DSF) Monitor Target] gdb traces
4 hello Monitor Target [(DSF) Monitor Target] gdb
5 hello Monitor Target [(DSF) Monitor Target] hello.adx
6 hello.adx [(DSF) Cross-Platform Debugging] gdb traces
7 hello.adx [(DSF) Cross-Platform Debugging] gdb
8 hello.adx [(DSF) Cross-Platform Debugging] hello.adx
9 [ICEMan] Console

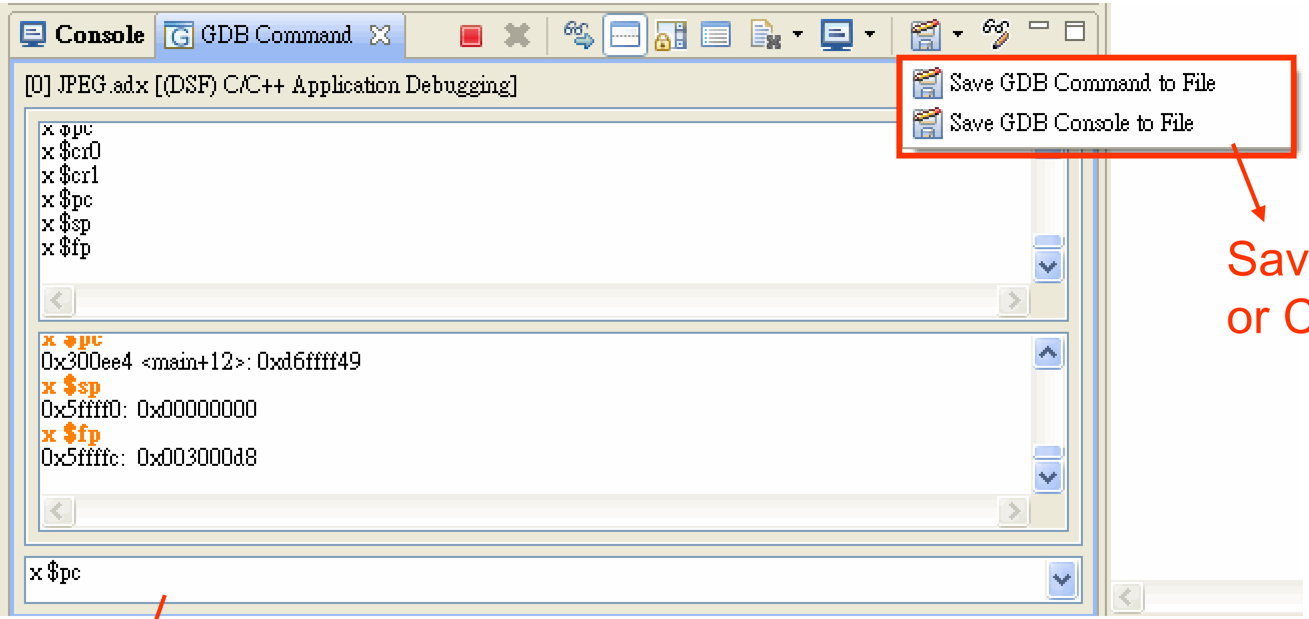


[ICEman:7254324483078325476] Console

```
Andes ICEman v1.0.0  BUILD_ID: 2011042717 <cygwin>
Copyright (C) 2011 Andes Technology Corporation.
JTAG freqency 48 MHz
Current device: hw_ver = 1 fw_ver = 5 fpga_ver = 5
Core #0 : 1000063d
There is 1 core in target.
Core #0 : Get EDM version is 0x50
The core #0 listens on 9900.
ICEman is ready to use.
warning: Going to max interrupt level 2 !!
Auto decrease one level.
```

GDB Command View





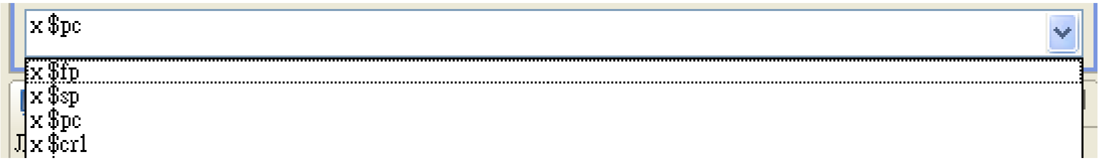
```
[0] JPEG.adx [(DSF) C/C++ Application Debugging]
```

```
x $pc
x $cr0
x $cr1
x $pc
x $sp
x $fp
```

```
x $pc
0x300ee4 <main+12>: 0xd6ffff49
x $sp
0x5ffff0:  0x00000000
x $fp
0x5ffffc:  0x003000d8
```

```
x $pc
```

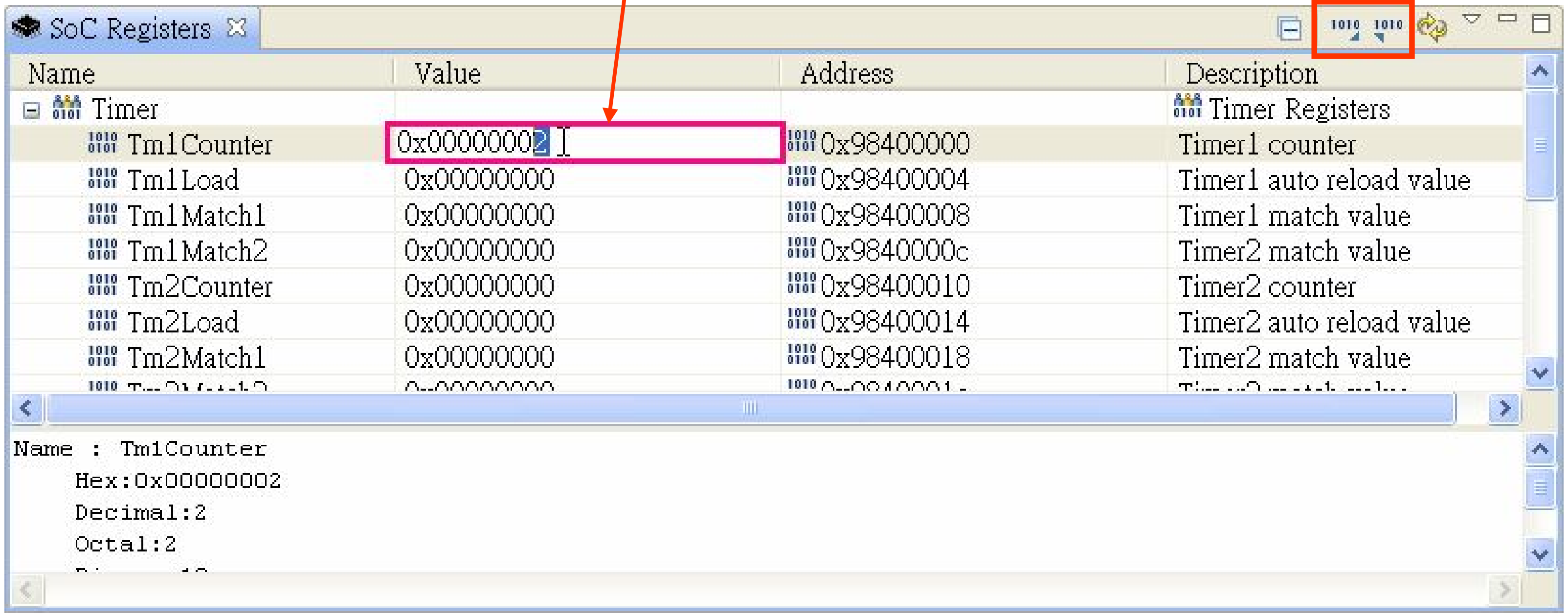
Save GDB Command or Console



```
x $pc
x $fp
x $sp
x $pc
x $cr1
```

This field keeps up to 20 records of the entered GDB commands

SoC Register View



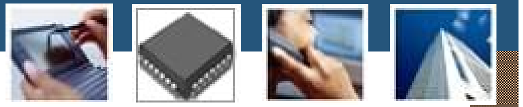
Modify Register Value

Import/Export Register Values

Name	Value	Address	Description
Timer			Timer Registers
Tm1Counter	0x00000002	0x98400000	Timer1 counter
Tm1Load	0x00000000	0x98400004	Timer1 auto reload value
Tm1Match1	0x00000000	0x98400008	Timer1 match value
Tm1Match2	0x00000000	0x9840000c	Timer2 match value
Tm2Counter	0x00000000	0x98400010	Timer2 counter
Tm2Load	0x00000000	0x98400014	Timer2 auto reload value
Tm2Match1	0x00000000	0x98400018	Timer2 match value
Tm2Match2	0x00000000	0x9840001c	Timer2 match value

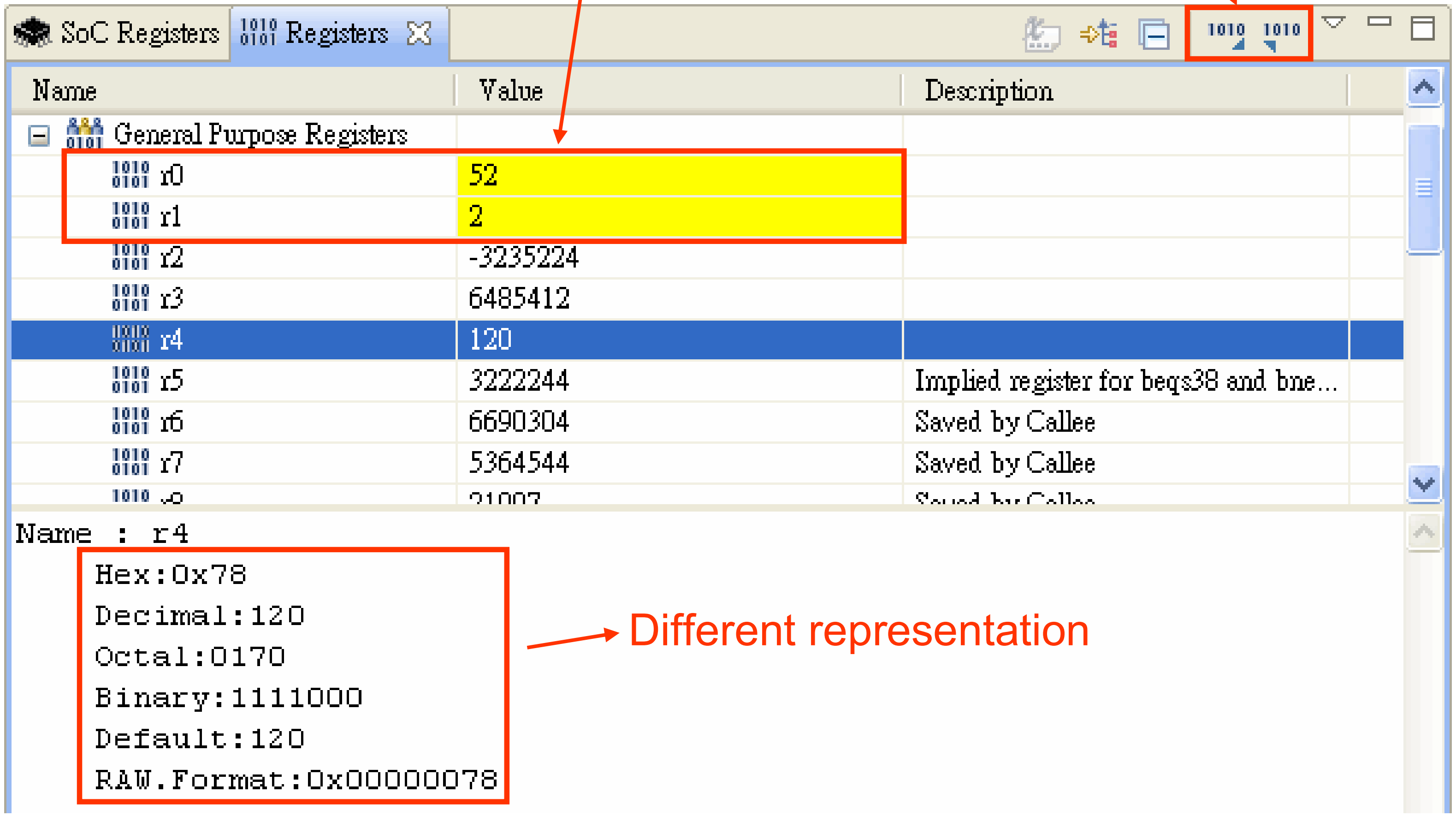
```
Name : Tm1Counter
     Hex:0x00000002
     Decimal:2
     Octal:2
     Binary:10
```

Register View



The changed values are highlighted

Import/Export Register Values

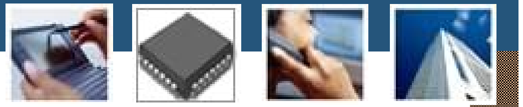


Name	Value	Description
General Purpose Registers		
r0	52	
r1	2	
r2	-3235224	
r3	6485412	
r4	120	
r5	3222244	Implied register for beqs38 and bne...
r6	6690304	Saved by Callee
r7	5364544	Saved by Callee
r8	21007	Saved by Callee

```
Name : r4
    Hex:0x78
    Decimal:120
    Octal:0170
    Binary:1111000
    Default:120
    RAW.Format:0x00000078
```

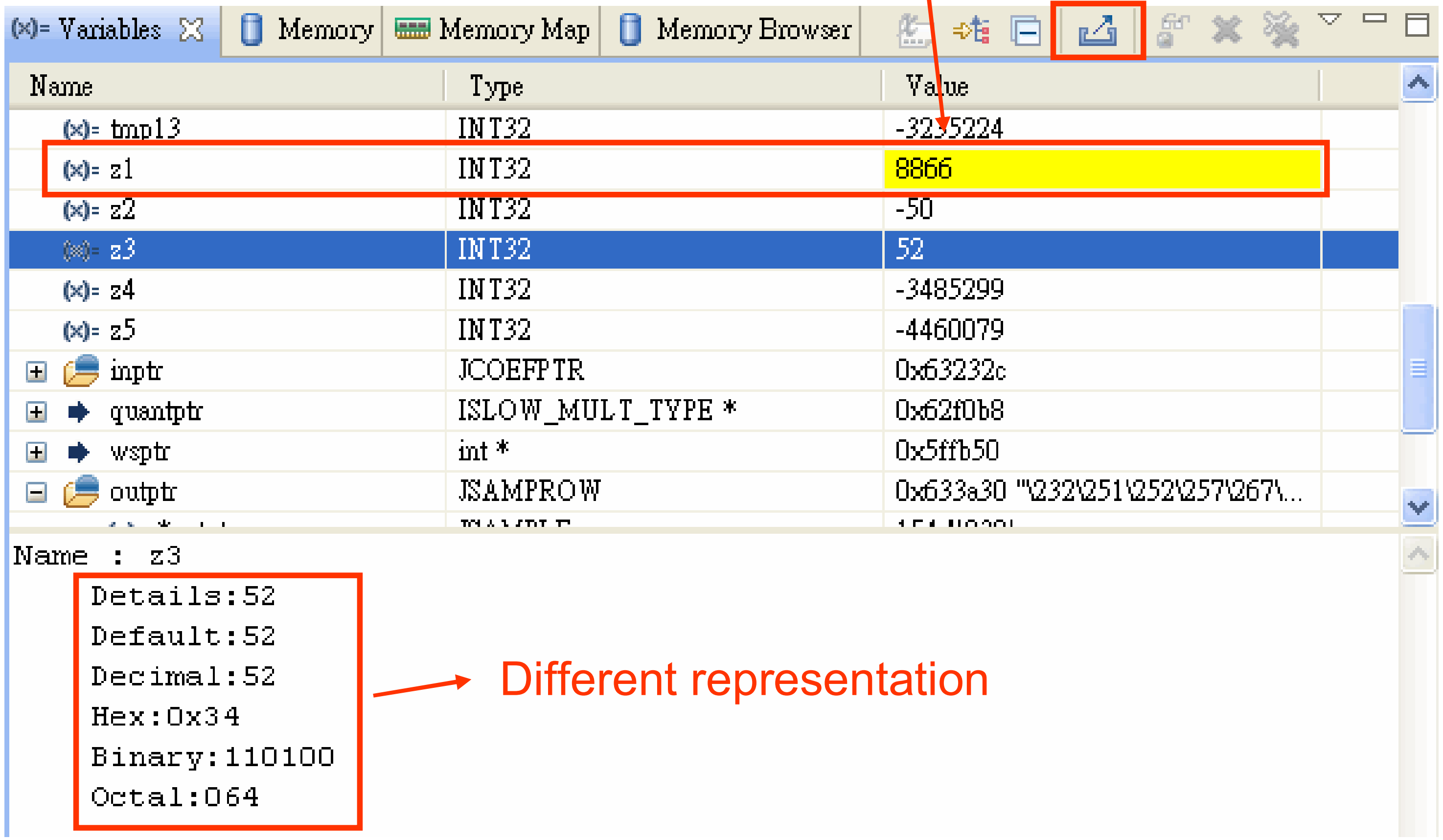
Different representation

Variables View (local variable)



The changed values are highlighted

Export Values

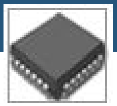
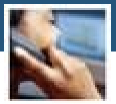
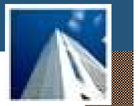


Name	Type	Value
(x)= tmp13	INT32	-3235224
(x)= z1	INT32	8866
(x)= z2	INT32	-50
(x)= z3	INT32	52
(x)= z4	INT32	-3485299
(x)= z5	INT32	-4460079
⊞ inptr	JCOEFPTR	0x63232c
⊞ quantptr	ISLOW_MULT_TYPE *	0x62f0b8
⊞ wsptr	int *	0x5ffb50
⊟ outptr	JSAMPROW	0x633a30 "\232\251\252\257\267\...

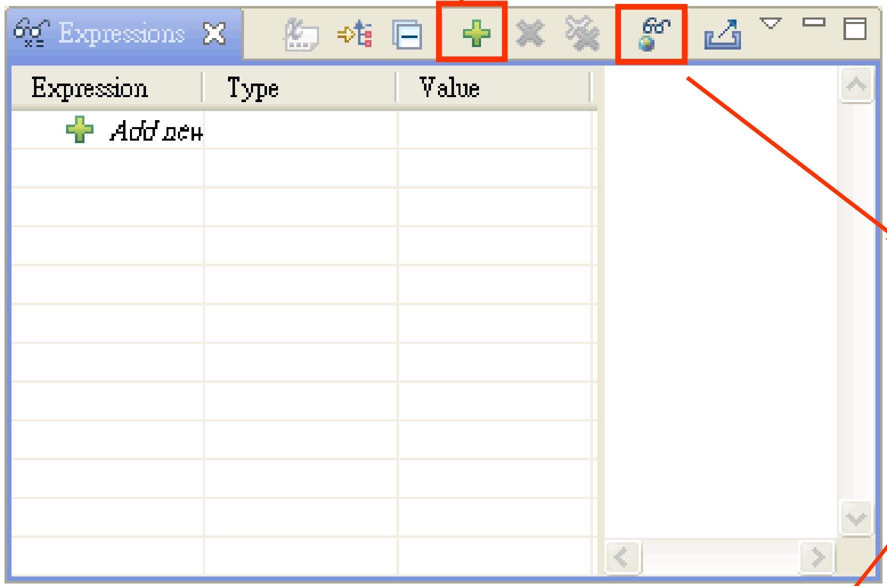
```
Name : z3
    Details:52
    Default:52
    Decimal:52
    Hex:0x34
    Binary:110100
    Octal:064
```

Different representation

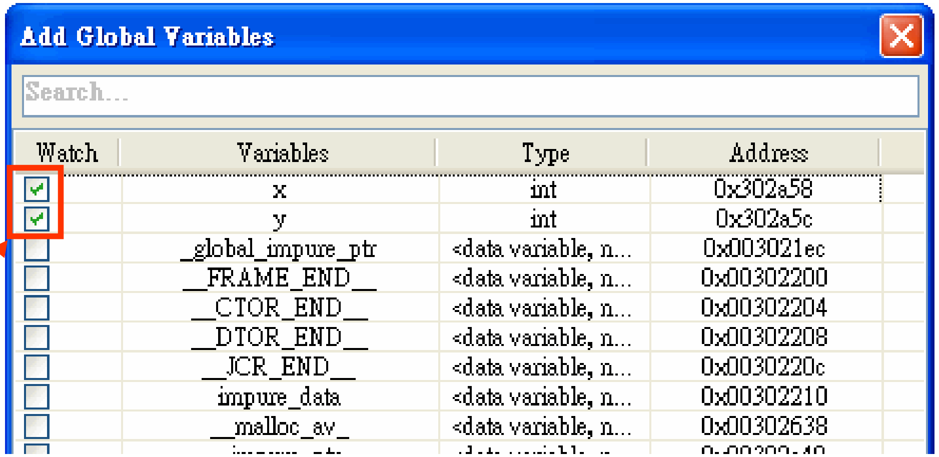
Expressions View (global variable)

Add Watch Expressions



List all global variables



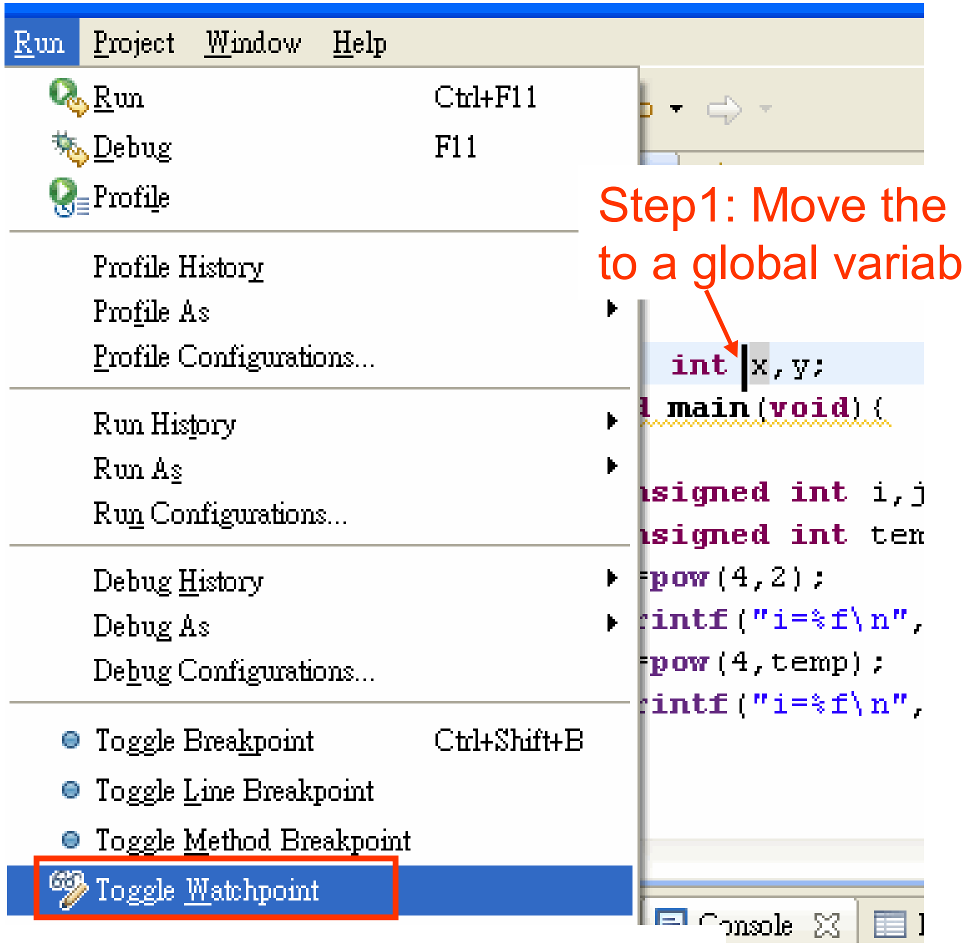
Add Global Variables

Search...

Watch	Variables	Type	Address
☑	x	int	0x302a58
☑	y	int	0x302a5c
□	_global_impure_ptr	<data variable, n...	0x003021ec
□	__FRAME_END__	<data variable, n...	0x00302200
□	__CTOR_END__	<data variable, n...	0x00302204
□	__DTOR_END__	<data variable, n...	0x00302208
□	__JCR_END__	<data variable, n...	0x0030220c
□	impure_data	<data variable, n...	0x00302210
□	__malloc_av_	<data variable, n...	0x00302638

Add global variables

Watchpoints

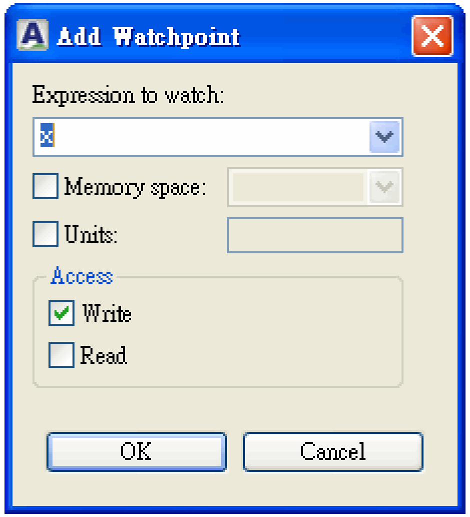


```
 int x,y;
l main(void){

nsigned int i,j
nsigned int tem
=pow(4,2);
rintf("i=%f\n",
=pow(4,temp);
rintf("i=%f\n",
```

Step1: Move the cursor to a global variable

Step2: Click “Toggle Watchpoint”

Step3: Select “Write” or “Read”



Add Watchpoint

Expression to watch: x

☐ Memory space:

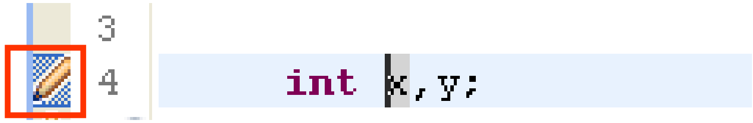
☐ Units:

Access

☑ Write

☐ Read

Step4: Complete

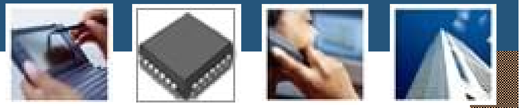


```
3
4      int x,y;
```

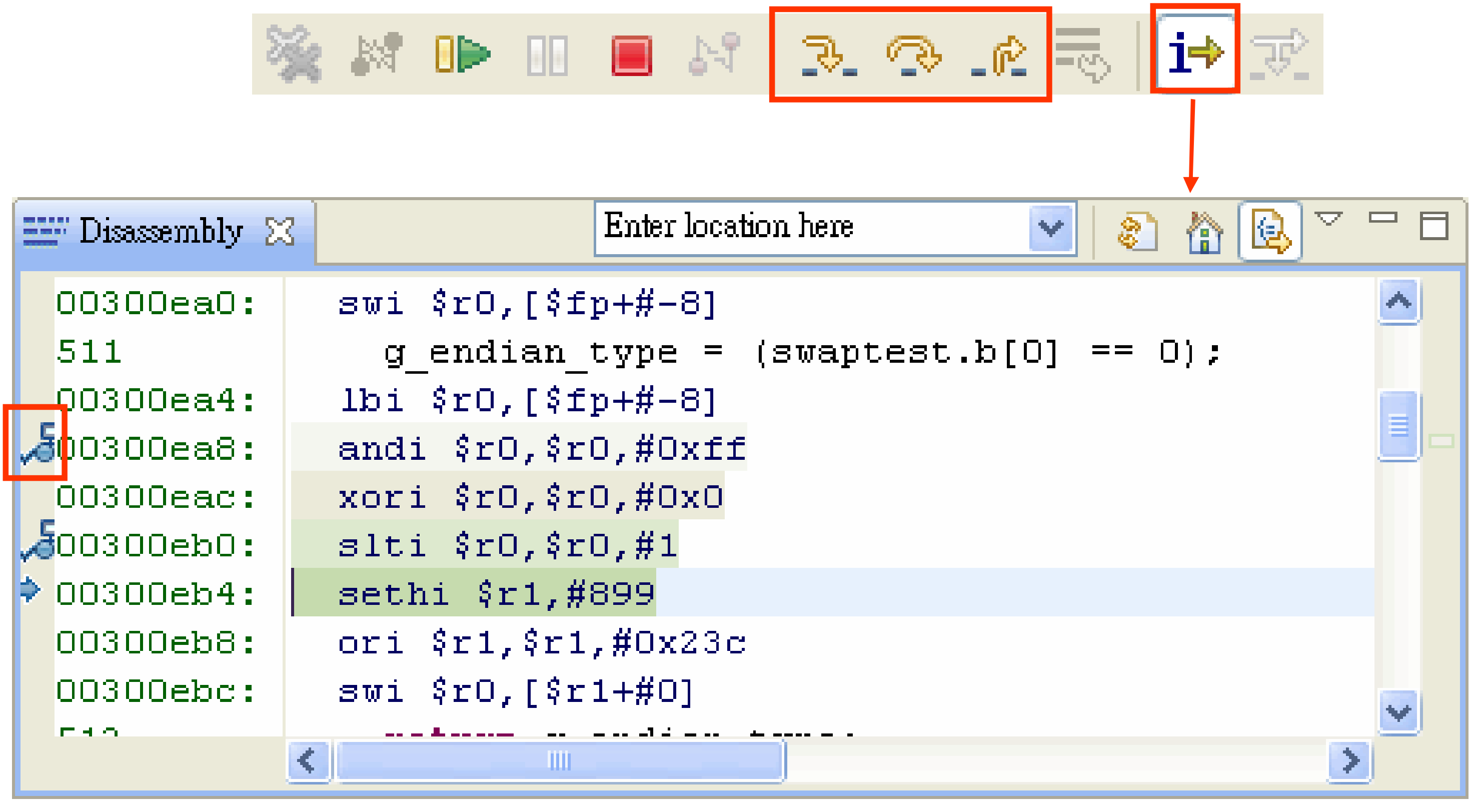
Page 57

Confidential

Disassembly View



Step Into, Step Over, Step Return

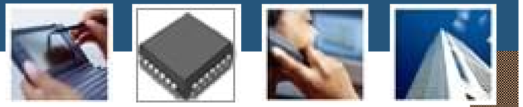


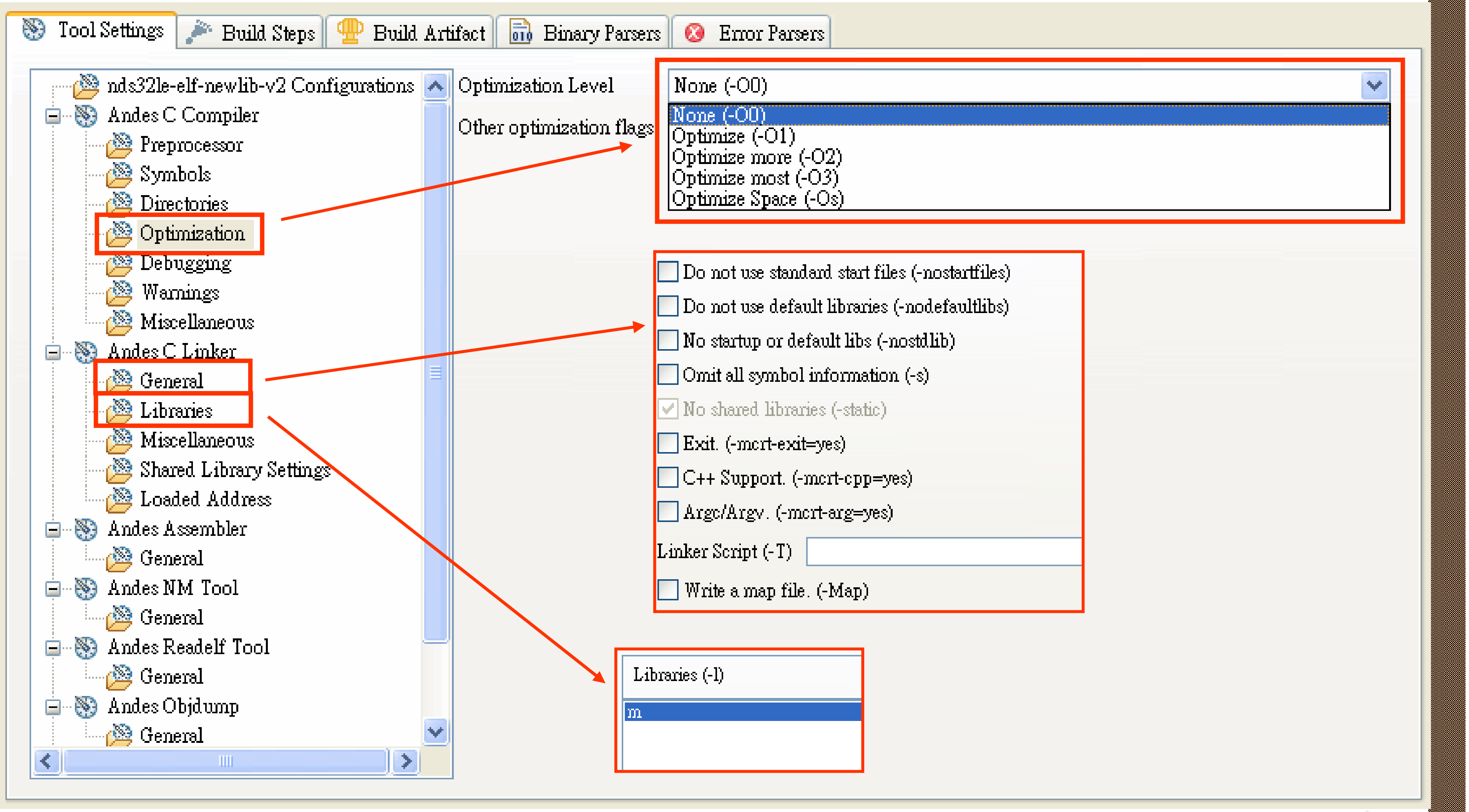
```
00300ea0:     swi $r0,[$fp+#-8]
511             g_endian_type = (swaptest.b[0] == 0);
00300ea4:     lbi $r0,[$fp+#-8]
00300ea8:     andi $r0,$r0,#0xff
00300eac:     xori $r0,$r0,#0x0
00300eb0:     slti $r0,$r0,#1
00300eb4:     sethi $r1,#899
00300eb8:     ori $r1,$r1,#0x23c
00300ebc:     swi $r0,[$r1+#0]
512             return g_endian_type;
```

Set breakpoints

- Compiler option setting
  - How to add compiler option
  - Optimization option for speed and space
  - GNU Utility setting

Managed Build System (1)





- nds32le-elf-newlib-v2 Configurations
  - Andes C Compiler
    - Preprocessor
    - Symbols
    - Directories
    - Optimization
    - Debugging
    - Warnings
    - Miscellaneous
  - Andes C Linker
    - General
    - Libraries
    - Miscellaneous
    - Shared Library Settings
    - Loaded Address
  - Andes Assembler
    - General
  - Andes NM Tool
    - General
  - Andes Readelf Tool
    - General
  - Andes Objdump
    - General

Optimization Level: None (-O0)

- None (-O0)
- Optimize (-O1)
- Optimize more (-O2)
- Optimize most (-O3)
- Optimize Space (-Os)

Other optimization flags

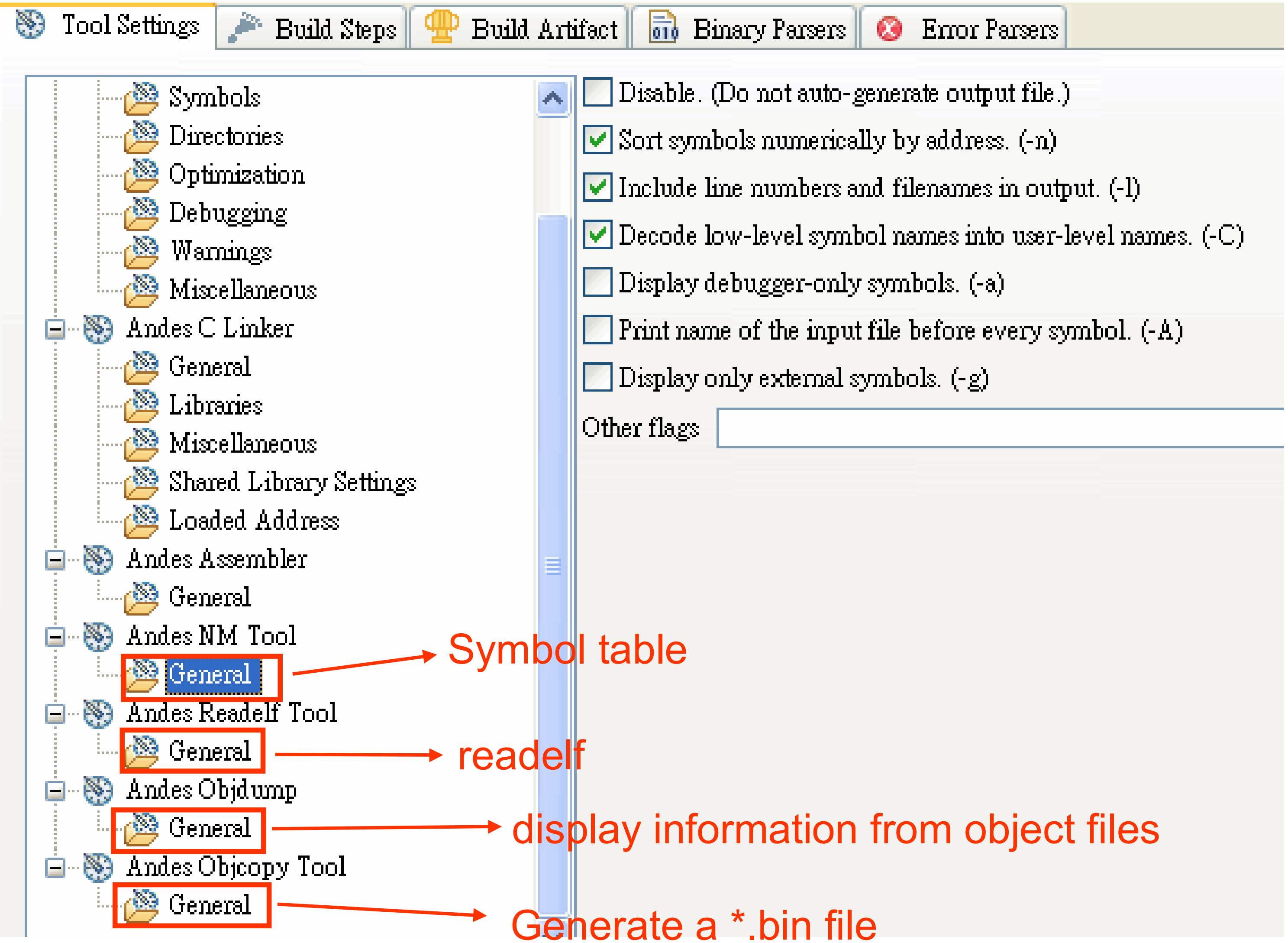
□ Do not use standard start files (-nostartfiles)
□ Do not use default libraries (-nodefaultlibs)
□ No startup or default libs (-nostdlib)
□ Omit all symbol information (-s)
☑ No shared libraries (-static)
□ Exit. (-mcrt-exit=yes)
□ C++ Support. (-mcrt-cpp=yes)
□ Argc/Argv. (-mcrt-arg=yes)
Linker Script (-T)
□ Write a map file. (-Map)

Libraries (-l)

m

Confidential

Managed Build System (2)



□ Disable. (Do not auto-generate output file.)

☑ Sort symbols numerically by address. (-n)

☑ Include line numbers and filenames in output. (-l)

☑ Decode low-level symbol names into user-level names. (-C)

□ Display debugger-only symbols. (-a)

□ Print name of the input file before every symbol. (-A)

□ Display only external symbols. (-g)

Other flags

Symbol table

readelf

display information from object files

Generate a *.bin file

Gcc options to maximize code size optimization

- -Os
- -fno-function-cse
- -funit-at-a-time
- -falign-jumps
- -fdata-sections
- -ffunction-sections   -Wl,--gc-sections

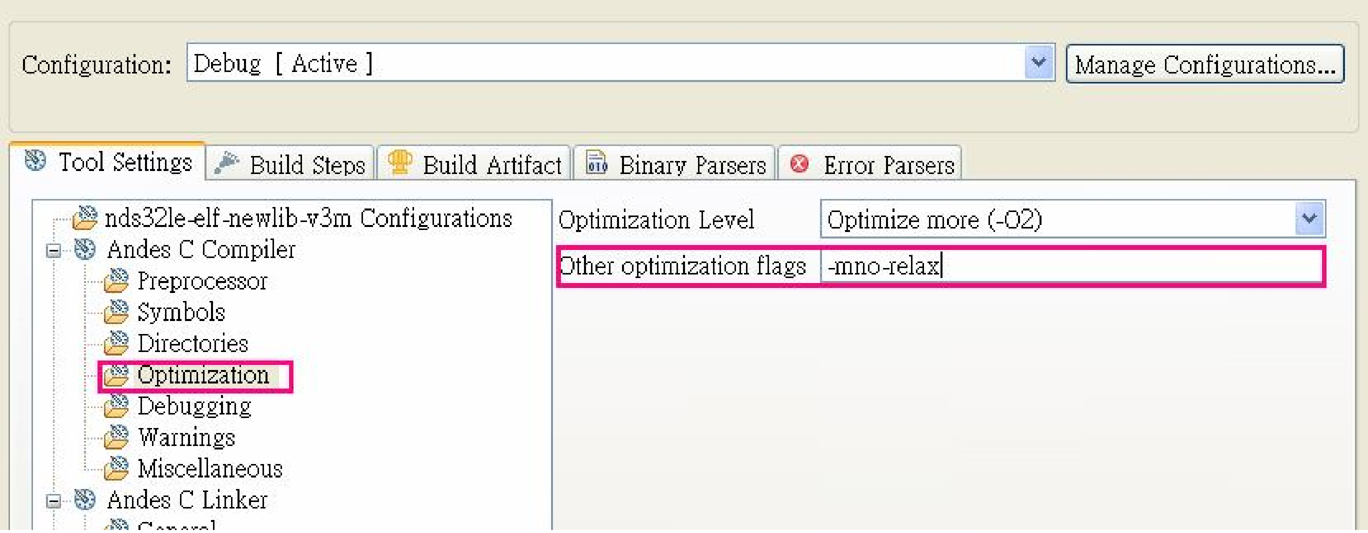
Gcc options to maximize code speed optimization

- -O3
- -fno-function-cse
- -funit-at-a-time
- -funroll-all-loops
- -fno-gcse

Default Applied Scenario of GCC Options at Each Optimization Level

Mnemonic	O0	O1	O2	O3	Os
-fomit-frame-pointer	N/A	Applied	Applied	Applied	Applied
-mrelax	N/A	N/A	Applied	Applied	Applied

- users can enter options -fno-omit-frame-pointer and -mno-relax in "Other optimization flags" field of "Andes C compiler > Optimization" page to avoid -fomit-frame-pointer and -mrelax



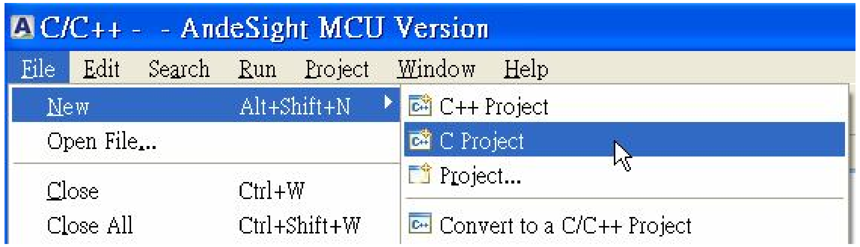
Configuration: Debug [Active]

Optimization Level: Optimize more (-O2)

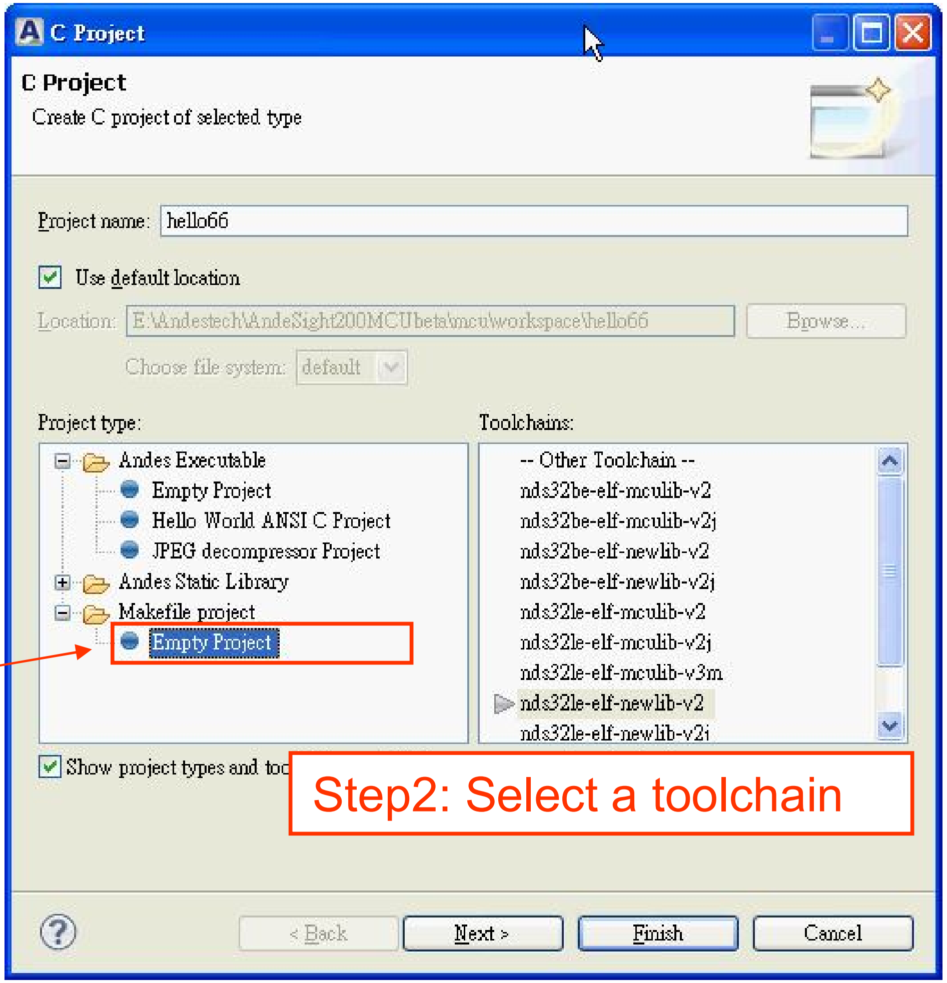
Other optimization flags: -mno-relax

- Makefile project and C project
  - Generic project demo
  - The environment variable of Makefile project

Create a New Project for a Generic Target


C/C++ - - AndeSight MCU Version
File Edit Search Run Project Window Help
New Alt+Shift+N
Open File...
Close Ctrl+W
Close All Ctrl+Shift+W
C++ Project
C Project
Project...
Convert to a C/C++ Project

Step1: File->New-> C Project



Project name: hello66

☑ Use default location

Location: E:\Andestech\AndeSight200MCUbeta\mcu\workspace\hello66

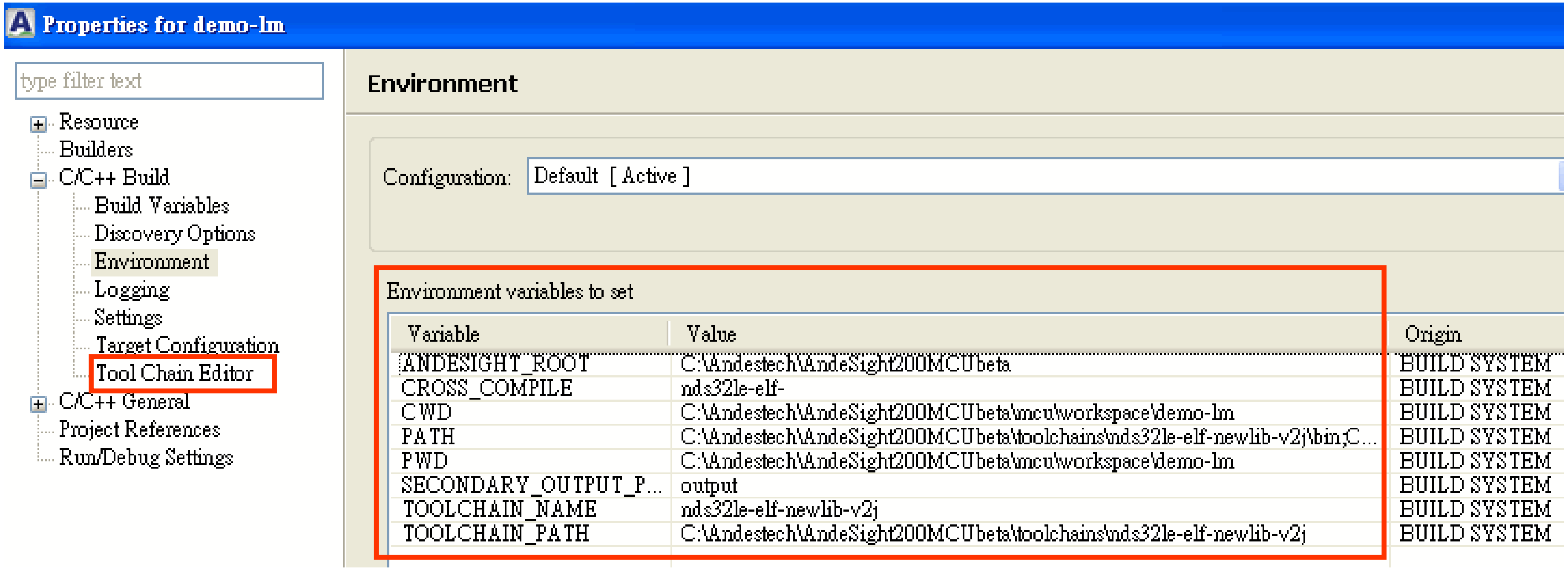
Project type:

Toolchains:

Makefile Project

Step2: Select a toolchain

Makefile project



Environment

Configuration: Default [Active]

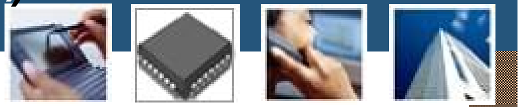
Environment variables to set

Variable	Value	Origin
ANDESIGHT_ROOT	C:\Andestech\AndeSight200MCUbeta	BUILD SYSTEM
CROSS_COMPILE	nds32le-elf-	BUILD SYSTEM
CWD	C:\Andestech\AndeSight200MCUbeta\mcu\workspace\demo-lm	BUILD SYSTEM
PATH	C:\Andestech\AndeSight200MCUbeta\toolchains\nds32le-elf-newlib-v2j\bin;C...	BUILD SYSTEM
PWD	C:\Andestech\AndeSight200MCUbeta\mcu\workspace\demo-lm	BUILD SYSTEM
SECONDARY_OUTPUT_P...	output	BUILD SYSTEM
TOOLCHAIN_NAME	nds32le-elf-newlib-v2j	BUILD SYSTEM
TOOLCHAIN_PATH	C:\Andestech\AndeSight200MCUbeta\toolchains\nds32le-elf-newlib-v2j	BUILD SYSTEM

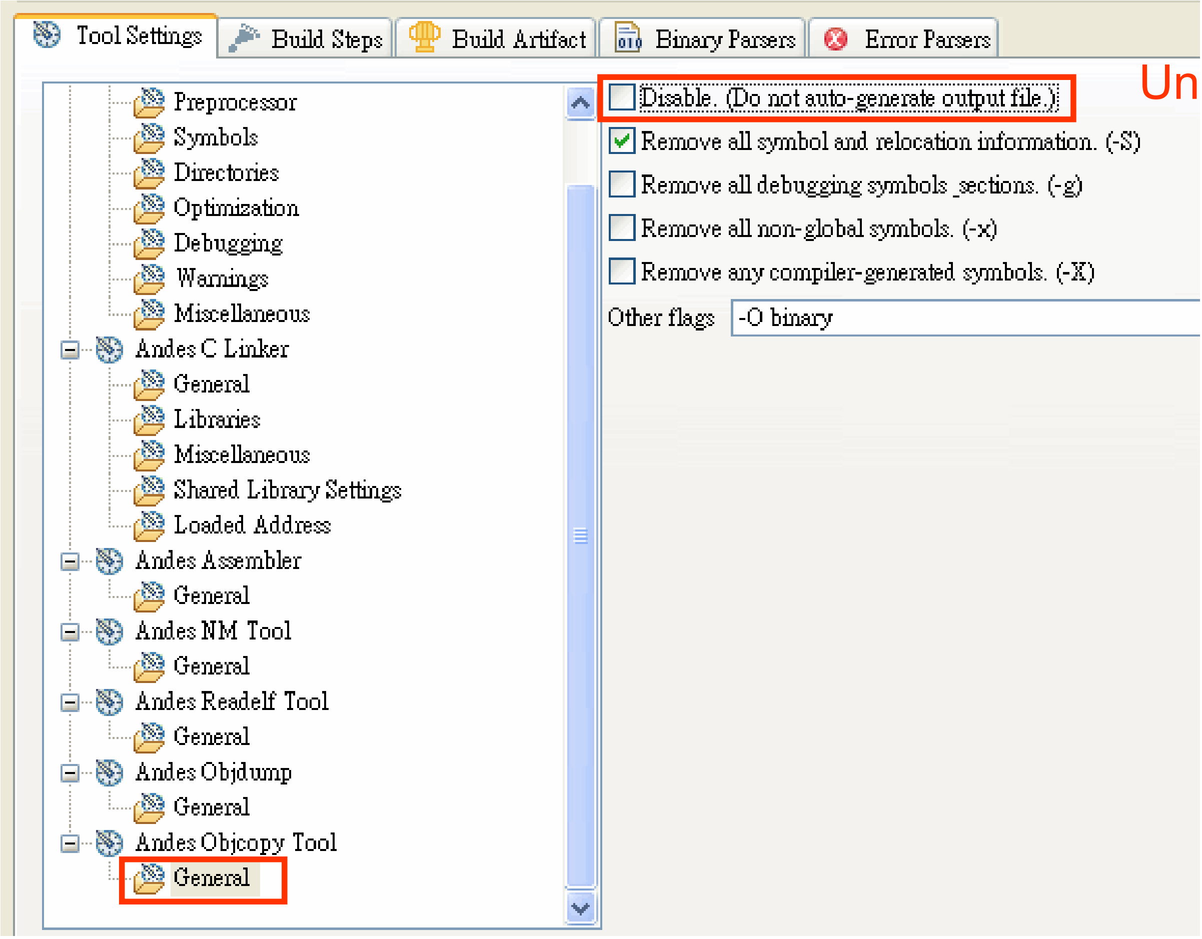
AndeSight manages the environment variables. When we change the toolchain setting here will change the environment variables.

- ❖ Flash burn and binary debugging

Flash Programming on a Real Board (1)



Step1: enable the Andes Objcopy Tool for generating an image file (binary type)



Tool Settings | Build Steps | Build Artifact | Binary Parsers | Error Parsers

- Preprocessor
- Symbols
- Directories
- Optimization
- Debugging
- Warnings
- Miscellaneous
- Andes C Linker
  - General
  - Libraries
  - Miscellaneous
  - Shared Library Settings
  - Loaded Address
- Andes Assembler
  - General
- Andes NM Tool
  - General
- Andes Readelf Tool
  - General
- Andes Objdump
  - General
- Andes Objcopy Tool
  - General

□ Disable. (Do not auto-generate output file.)

Uncheck this option

☑ Remove all symbol and relocation information. (-S)

□ Remove all debugging symbols _sections. (-g)

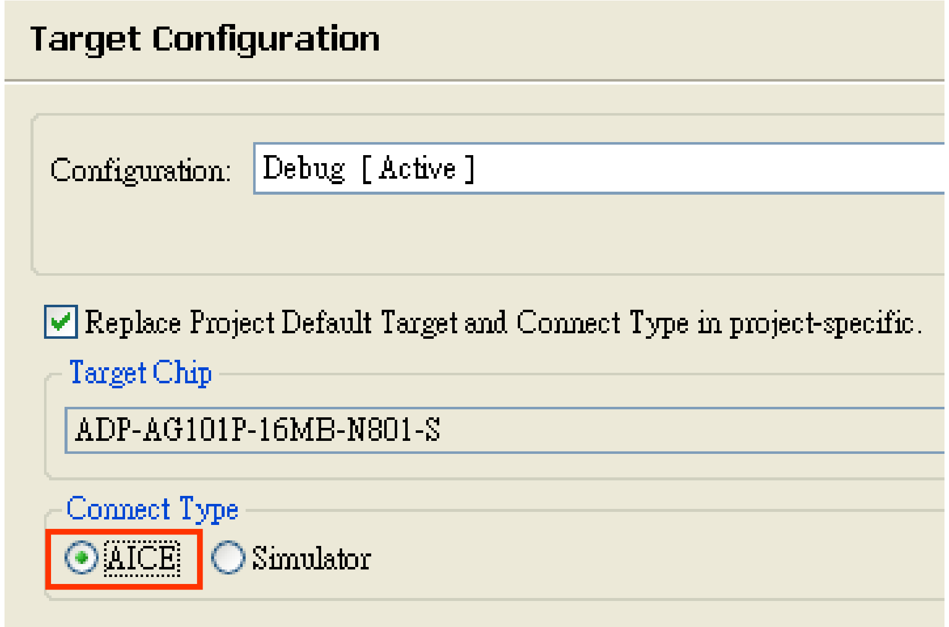
□ Remove all non-global symbols. (-x)

□ Remove any compiler-generated symbols. (-X)

Other flags: -O binary

Flash Programming on a Real Board (2)

Step2: Set the Connect Type of Target Configuration as "AICE".



Target Configuration

Configuration: Debug [Active]

☑ Replace Project Default Target and Connect Type in project-specific.

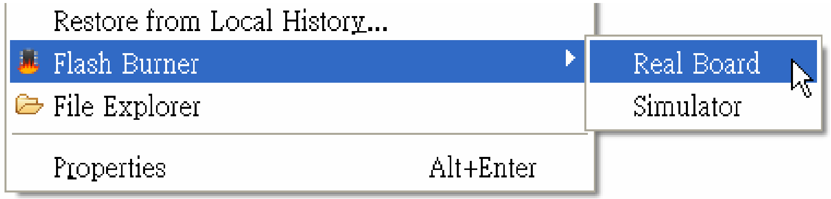
Target Chip

ADP-AG101P-16MB-N801-S

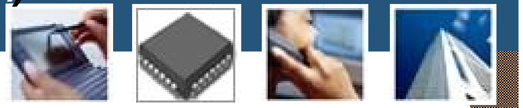
Connect Type

● AICE ○ Simulator

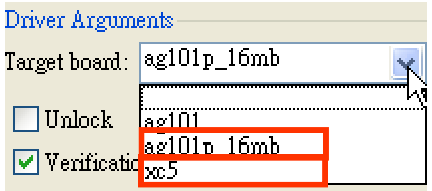
Step3: Right click the project folder and choose "Flash Burner > Real Board"



Flash Programming on a Real Board (3)



Step4: Click on "Burn"



Driver Arguments

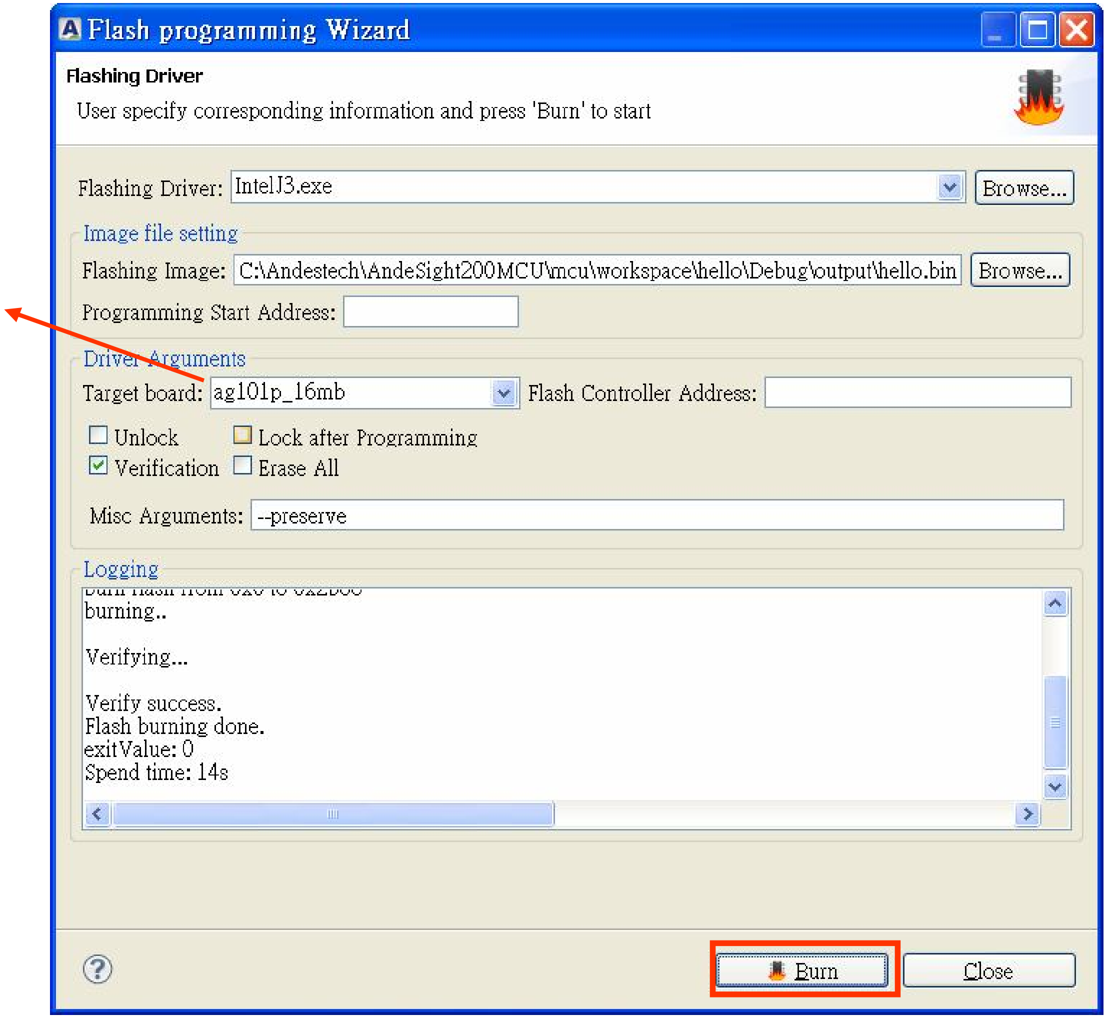
Target board: ag101p_16mb

☐ Unlock

☑ Verificatio

ag101p_16mb: N8 netlist

xc5: other 32bit netlist



Flashing Driver

User specify corresponding information and press 'Burn' to start

Flashing Driver: IntelJ3.exe

Image file setting

Flashing Image: C:\Andestech\AndeSight200MCU\mcu\workspace\hello\Debug\output\hello.bin

Programming Start Address:

Driver Arguments

Target board: ag101p_16mb

Flash Controller Address:

☐ Unlock ☐ Lock after Programming

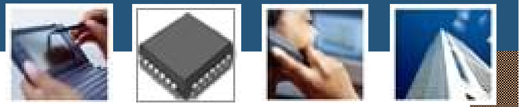
☑ Verification ☐ Erase All

Misc Arguments: --preserve

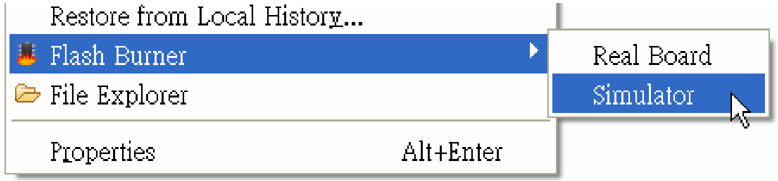
Logging

```
burn flash from 0x0 to 0x2b00
burning..

Verifying...

Verify success.
Flash burning done.
exitValue: 0
Spend time: 14s
```

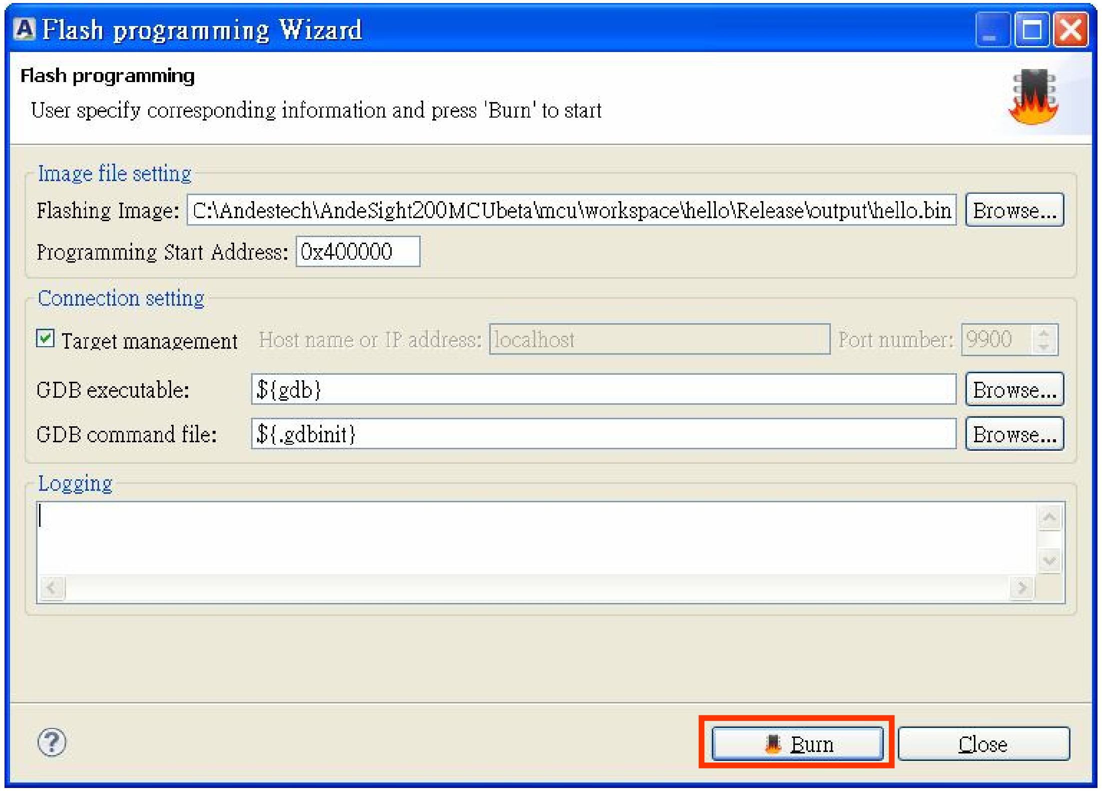
Flash Programming on a Simulator



Step1: Right click the project folder and choose "Flash Burner > Simulator"



Step2: Click on "Burn"



Flash programming

User specify corresponding information and press 'Burn' to start

Image file setting

Flashing Image: C:\Andestech\AndeSight200MCUbeta\mcu\workspace\hello\Release\output\hello.bin

Programming Start Address: 0x400000

Connection setting

☑ Target management Host name or IP address: localhost Port number: 9900

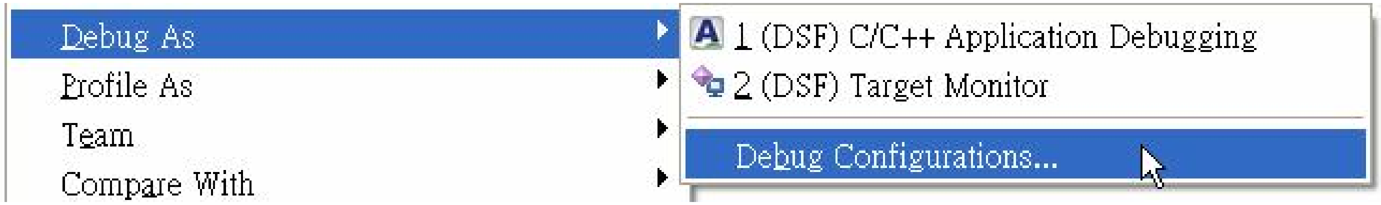
GDB executable: ${gdb}

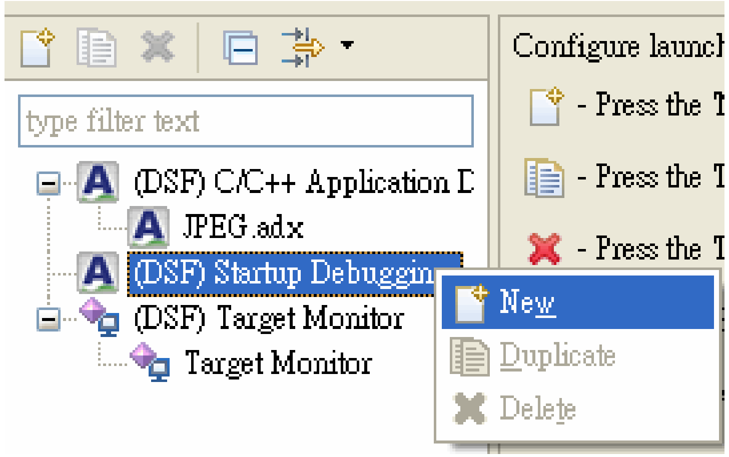
GDB command file: ${.gdbinit}

Logging

Startup Debugging

- ❖ Users may either burn the binary file onto a target or load image/symbol table onto ram prior to binary debugging.

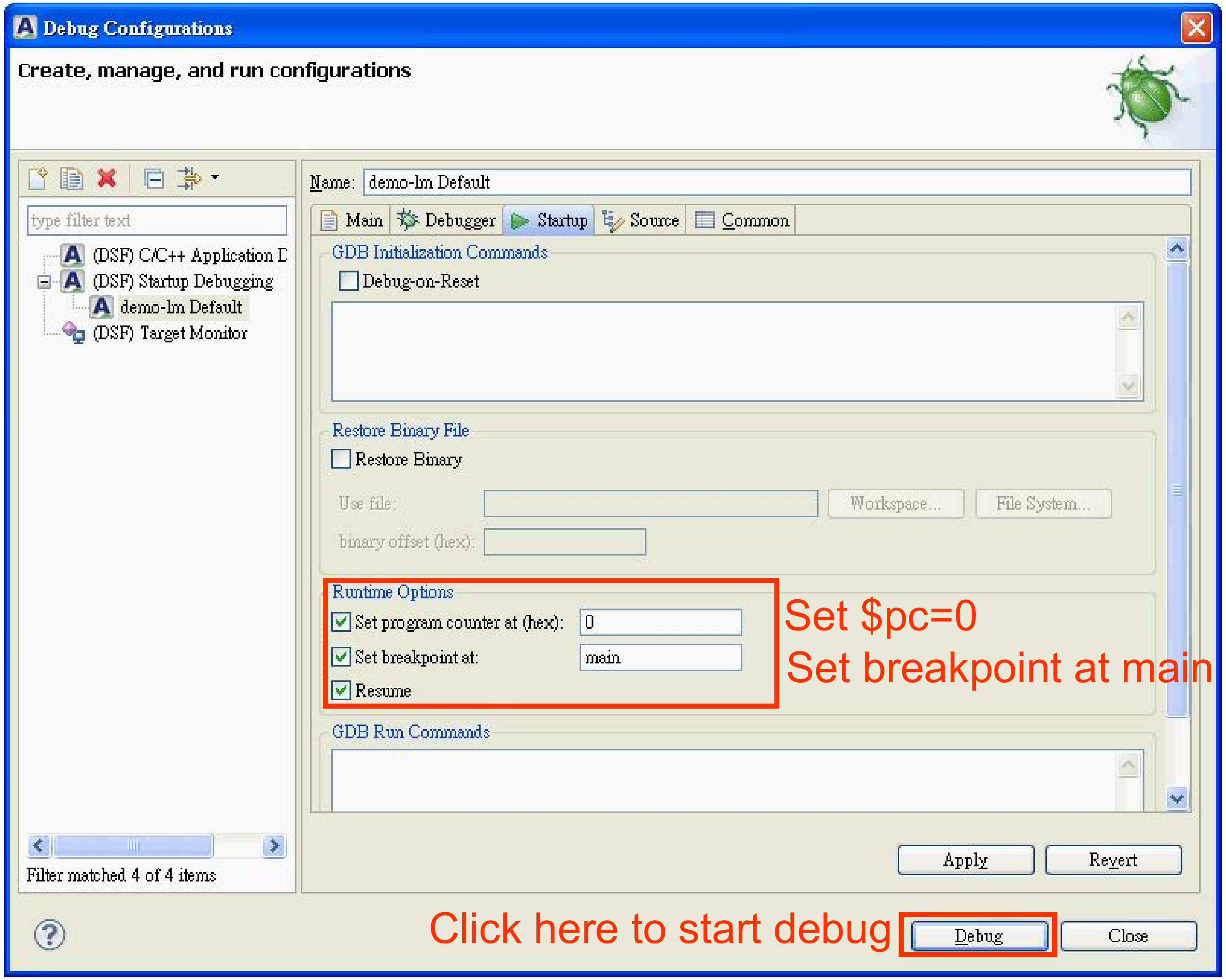




Page 73

Confidential

Startup debug



Debug Configurations

Create, manage, and run configurations

Name: demo-lm Default

GDB Initialization Commands

□ Debug-on-Reset

Restore Binary File

□ Restore Binary

Use file:

binary offset (hex):

Runtime Options

☑ Set program counter at (hex): 0

☑ Set breakpoint at: main

☑ Resume

GDB Run Commands

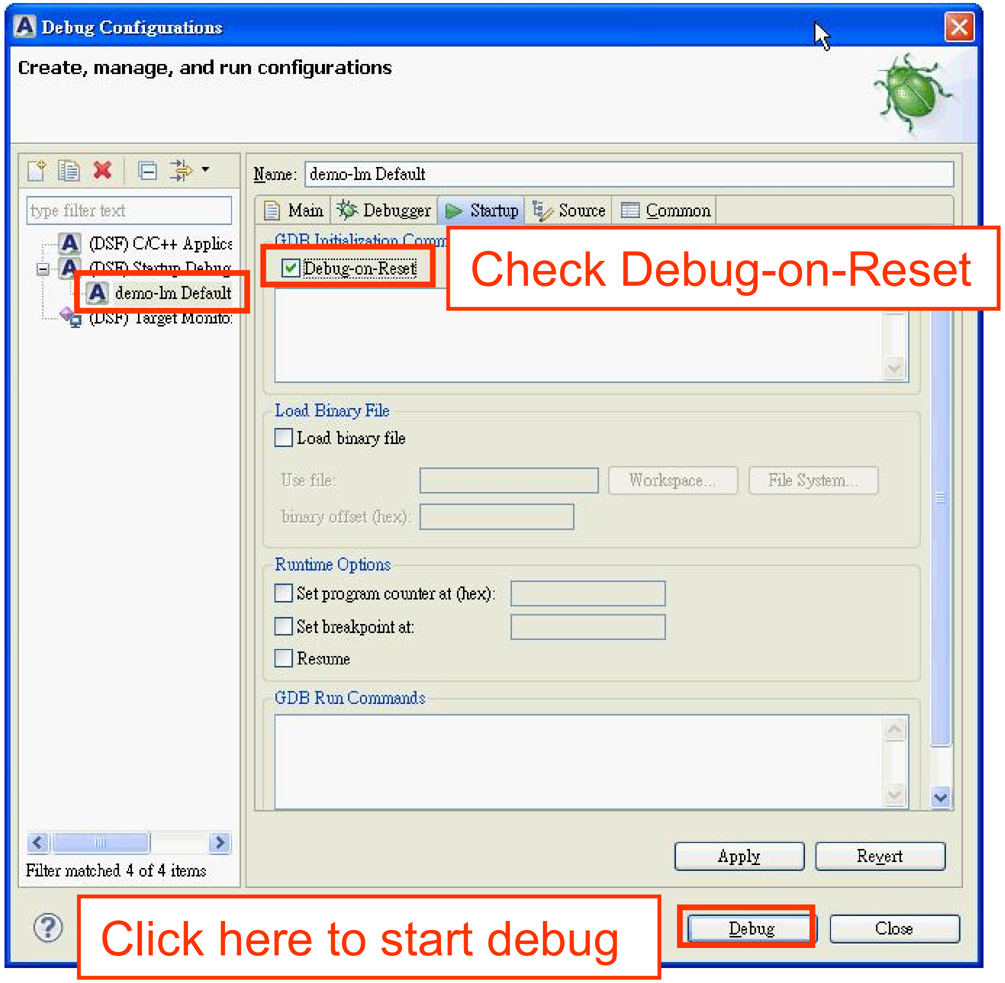
Set $pc=0

Set breakpoint at main

Click here to start debug

Debug-On-Reset Configuration (1)

- The debug-on-reset feature will make the debugger hold CPU right after debugging target reset.



Debug Configurations

Create, manage, and run configurations

Name: demo-lm Default

GDB Initialization Comm

☑ Debug-on-Reset

Load Binary File

□ Load binary file

Use file:

binary offset (hex):

Runtime Options

□ Set program counter at (hex):

□ Set breakpoint at:

□ Resume

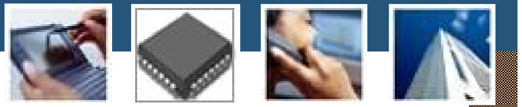
GDB Run Commands

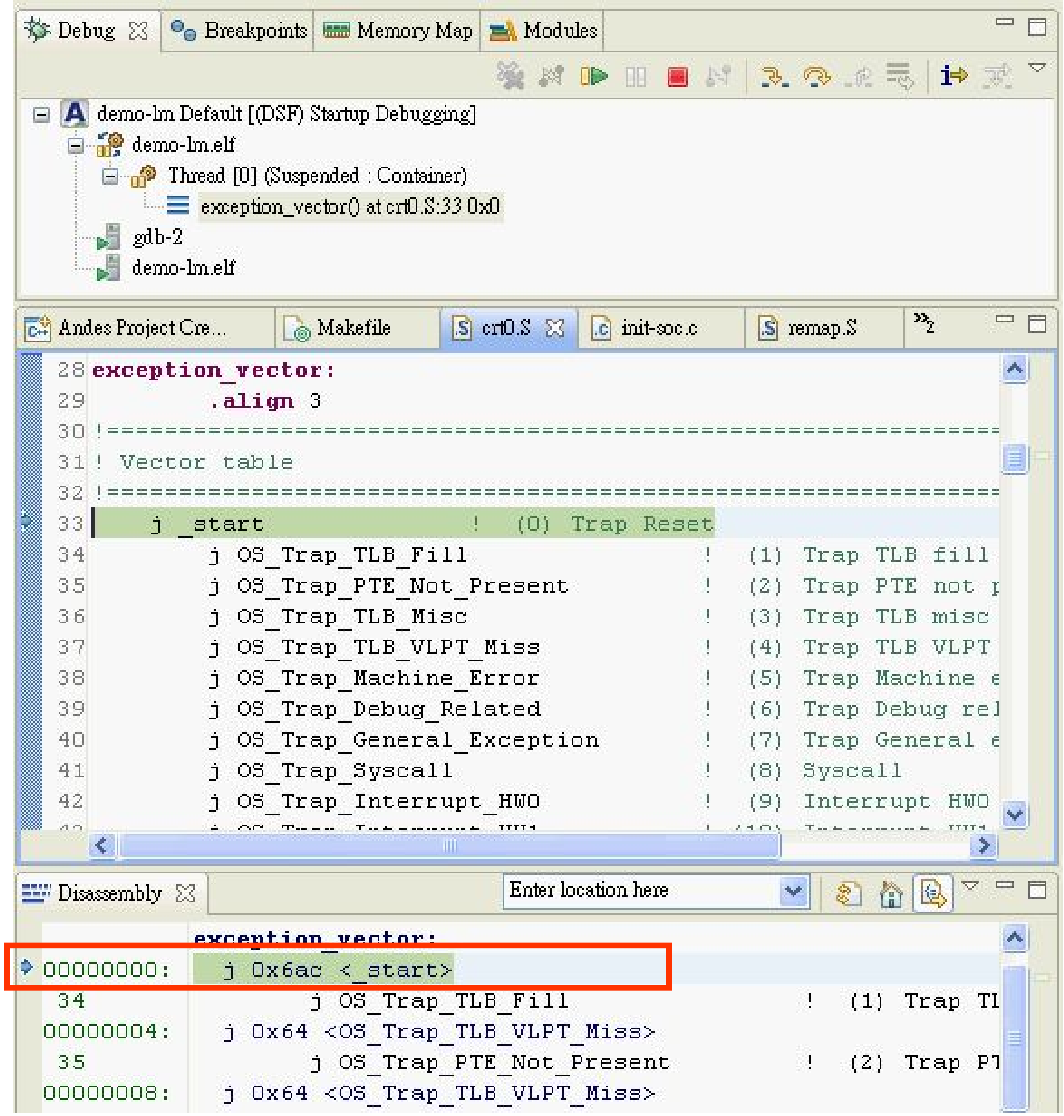
Filter matched 4 of 4 items

Check Debug-on-Reset

Click here to start debug

Debug-On-Reset Configuration (2)





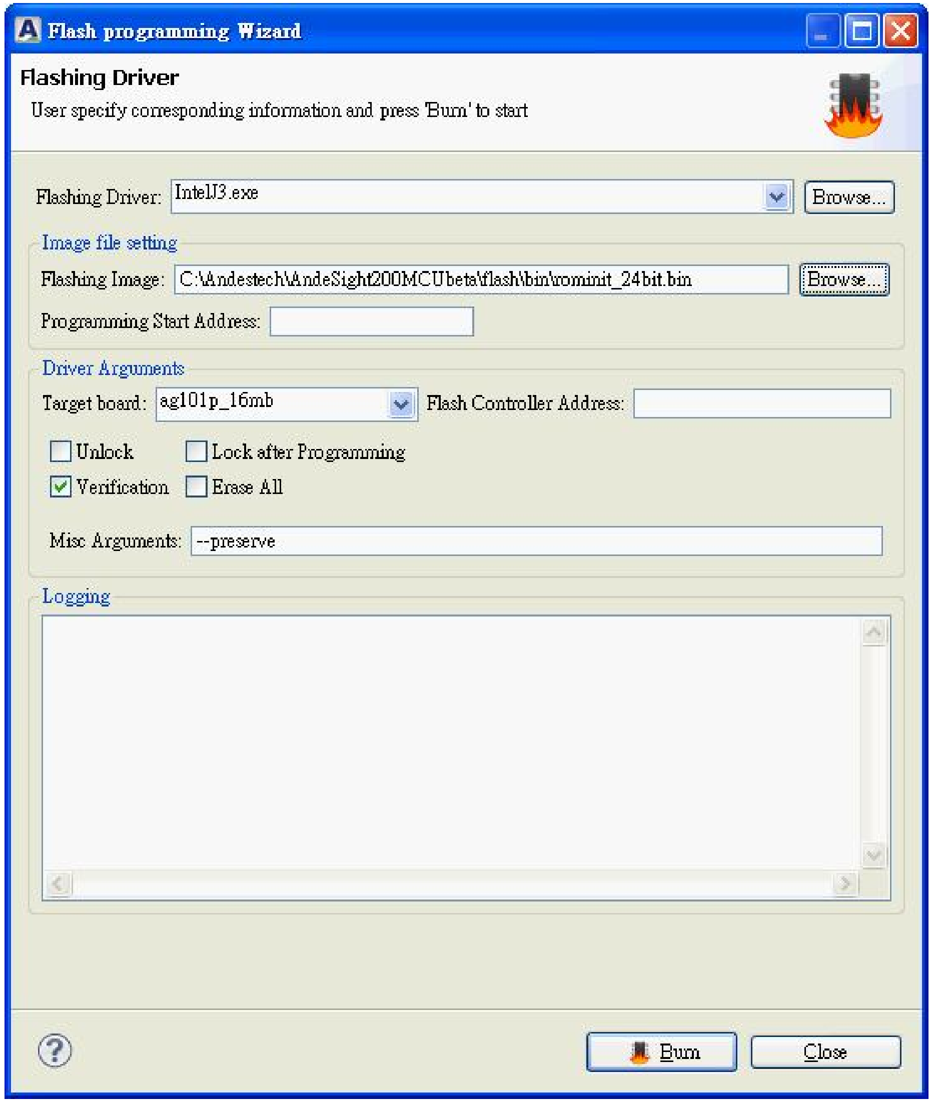
```
demo-lm Default [(DSF) Startup Debugging]
  demo-lm.elf
    Thread [0] (Suspended : Container)
      exception_vector() at crt0.S:33 0x0
  gdb-2
  demo-lm.elf
```

```
exception_vector:
        .align 3
!=================================================================
! Vector table
!=================================================================
    j _start                  !  (0) Trap Reset
        j OS_Trap_TLB_Fill                !  (1) Trap TLB fill
        j OS_Trap_PTE_Not_Present         !  (2) Trap PTE not p
        j OS_Trap_TLB_Misc                !  (3) Trap TLB misc
        j OS_Trap_TLB_VLPT_Miss           !  (4) Trap TLB VLPT
        j OS_Trap_Machine_Error           !  (5) Trap Machine e
        j OS_Trap_Debug_Related           !  (6) Trap Debug rel
        j OS_Trap_General_Exception       !  (7) Trap General e
        j OS_Trap_Syscall                 !  (8) Syscall
        j OS_Trap_Interrupt_HW0           !  (9) Interrupt HW0
```

```
            exception_vector:
00000000:   j 0x6ac <_start>
 34                j OS_Trap_TLB_Fill                !  (1) Trap TL
00000004:   j 0x64 <OS_Trap_TLB_VLPT_Miss>
 35                j OS_Trap_PTE_Not_Present         !  (2) Trap PT
00000008:   j 0x64 <OS_Trap_TLB_VLPT_Miss>
```

Hold CPU right after debugging target reset

Burn original boot code



Flashing Driver

User specify corresponding information and press 'Burn' to start

Flashing Driver: IntelJ3.exe

Image file setting

Flashing Image: C:\Andestech\AndeSight200MCUbeta\flash\bin\rominit_24bit.bin

Programming Start Address:

Driver Arguments

Target board: ag101p_16mb

Flash Controller Address:

☐ Unlock ☐ Lock after Programming

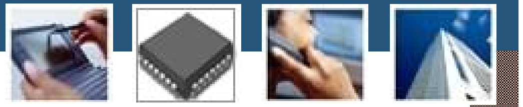
☑ Verification ☐ Erase All

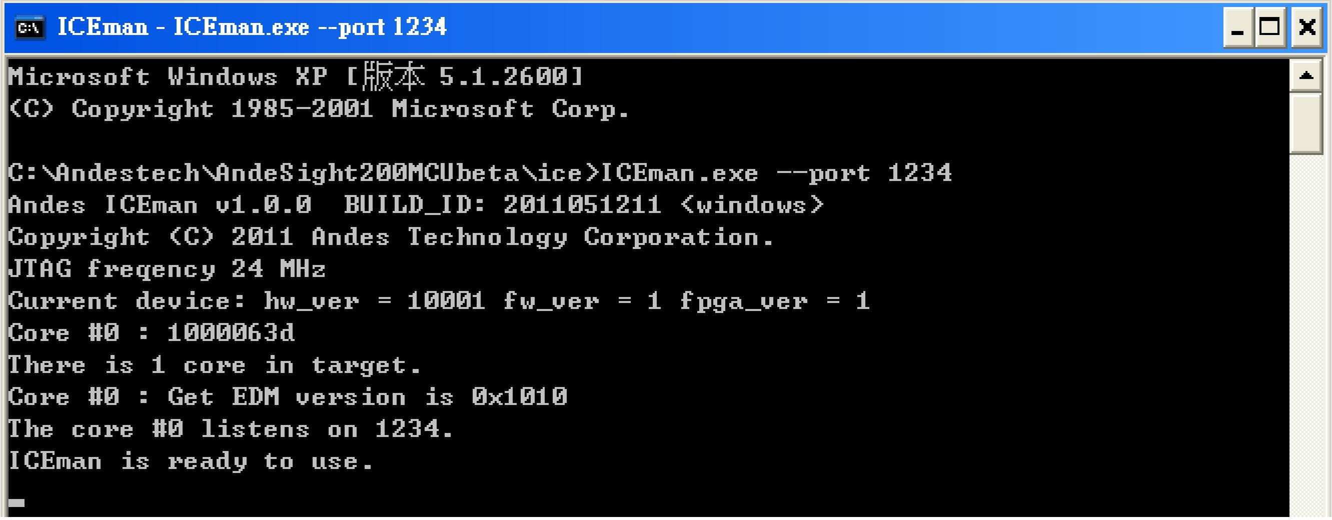
Misc Arguments: --preserve

Logging

Confidential

burn flash in command line

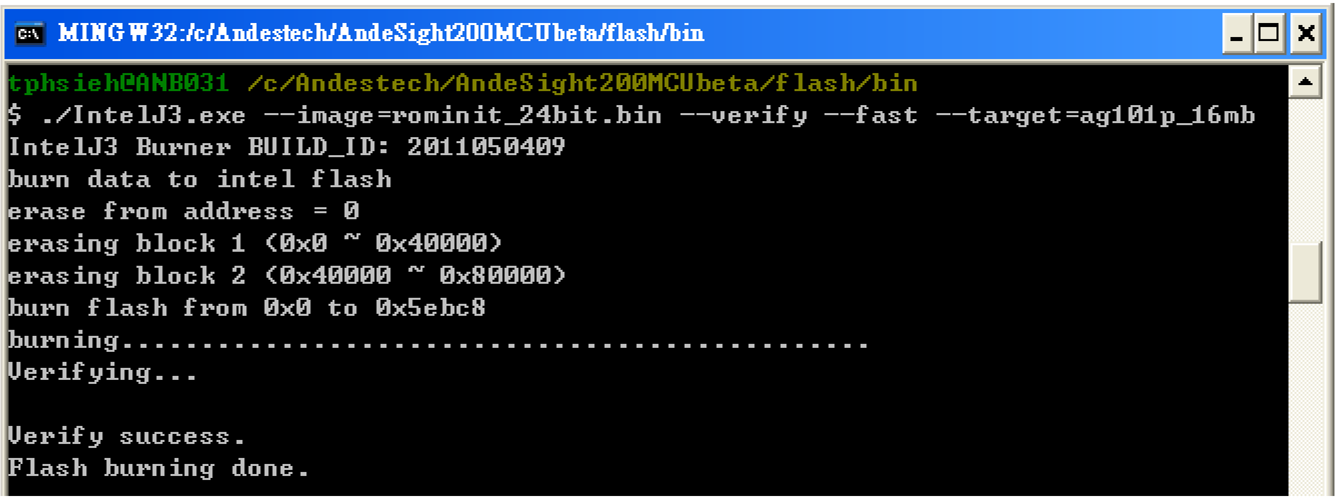




ICEman - ICEman.exe --port 1234

```
Microsoft Windows XP [版本 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Andestech\AndeSight200MCUbeta\ice>ICEman.exe --port 1234
Andes ICEman v1.0.0  BUILD_ID: 2011051211 <windows>
Copyright (C) 2011 Andes Technology Corporation.
JTAG freqency 24 MHz
Current device: hw_ver = 10001 fw_ver = 1 fpga_ver = 1
Core #0 : 1000063d
There is 1 core in target.
Core #0 : Get EDM version is 0x1010
The core #0 listens on 1234.
ICEman is ready to use.
```

./IntelJ3.exe --image=rominit_24bit.bin --verify --fast --arget=ag101p_16mb



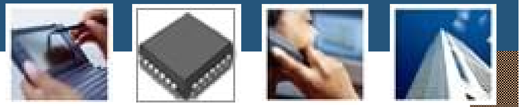
MINGW32:/c/Andestech/AndeSight200MCUbeta/flash/bin

```
tphsieh@ANB031 /c/Andestech/AndeSight200MCUbeta/flash/bin
$ ./IntelJ3.exe --image=rominit_24bit.bin --verify --fast --target=ag101p_16mb
IntelJ3 Burner BUILD_ID: 2011050409
burn data to intel flash
erase from address = 0
erasing block 1 (0x0 ~ 0x40000)
erasing block 2 (0x40000 ~ 0x80000)
burn flash from 0x0 to 0x5ebc8
burning.............................................
Verifying...

Verify success.
Flash burning done.
```

- IntelJ3 burning introduction
  - The IntelJ3 program and IntelJ3 spec

❖ AndeSight200MCUbeta under Start menu

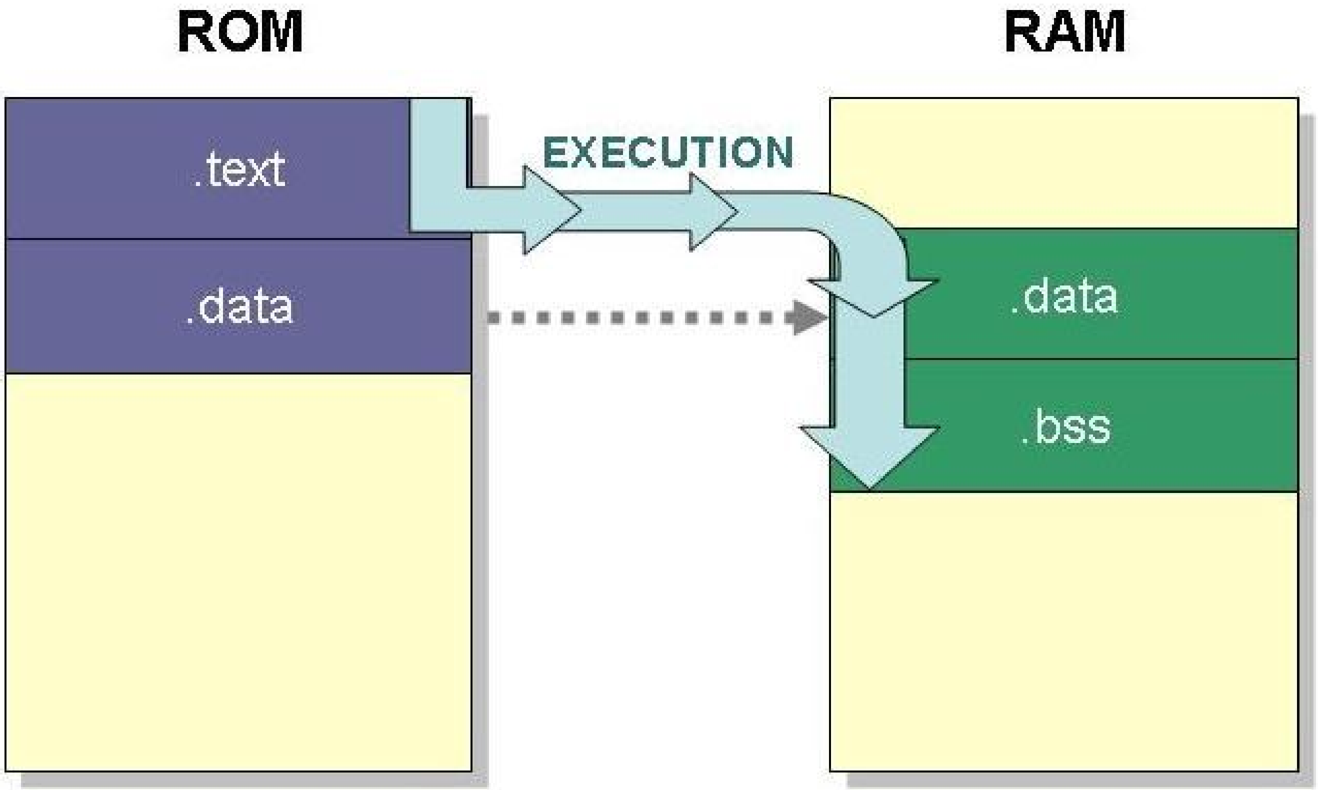
- AICE
- Documents
- Toolchains



- Demo program
  - JPEG
  - demo-lm
  - demo-ls1
  - demo-ls2
  - demo-ls3
  - demo-int
  - demo-int-c-ext
  - demo-pfm
  - demo-cache

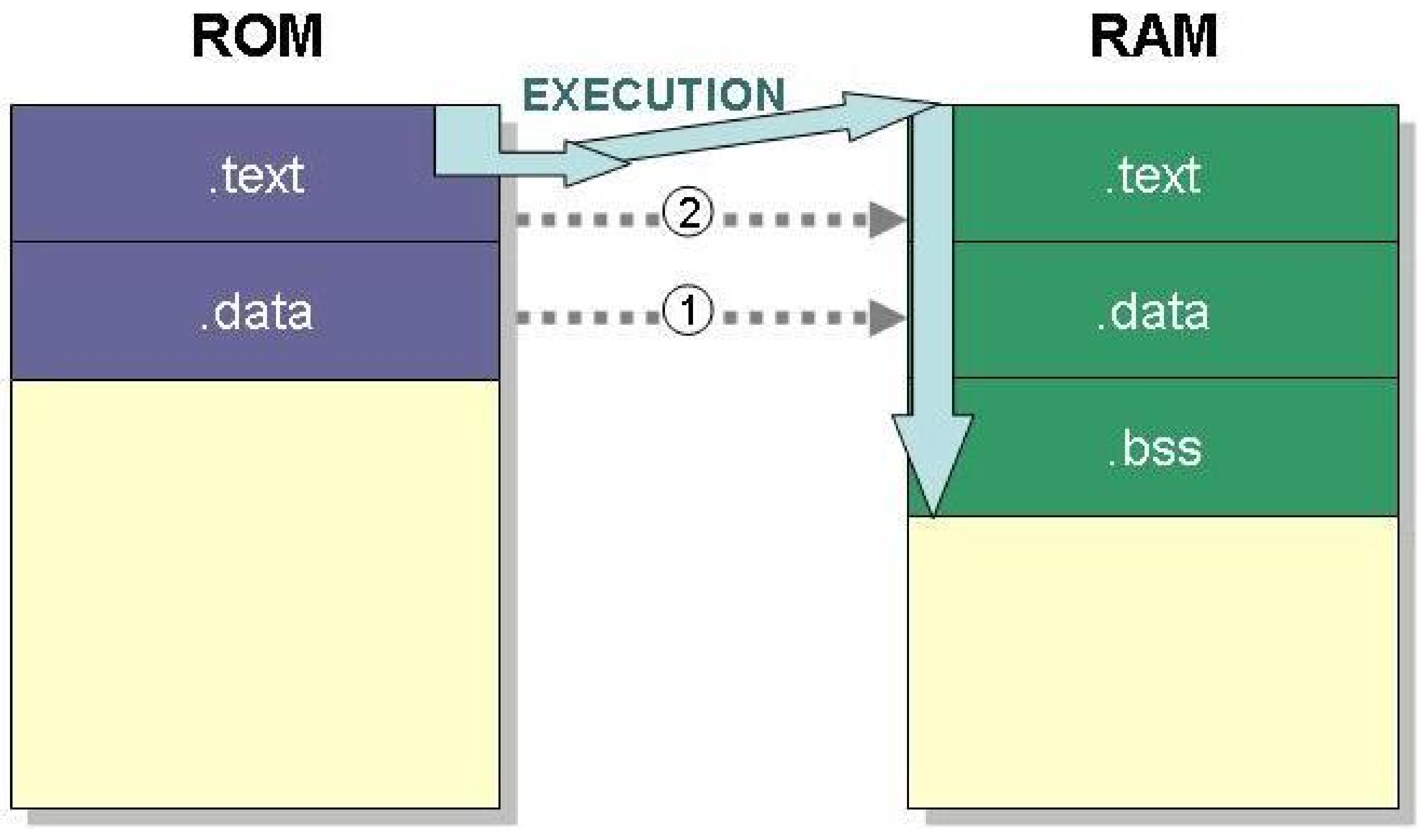
demo-ls1

- Boot-n-run from ROM



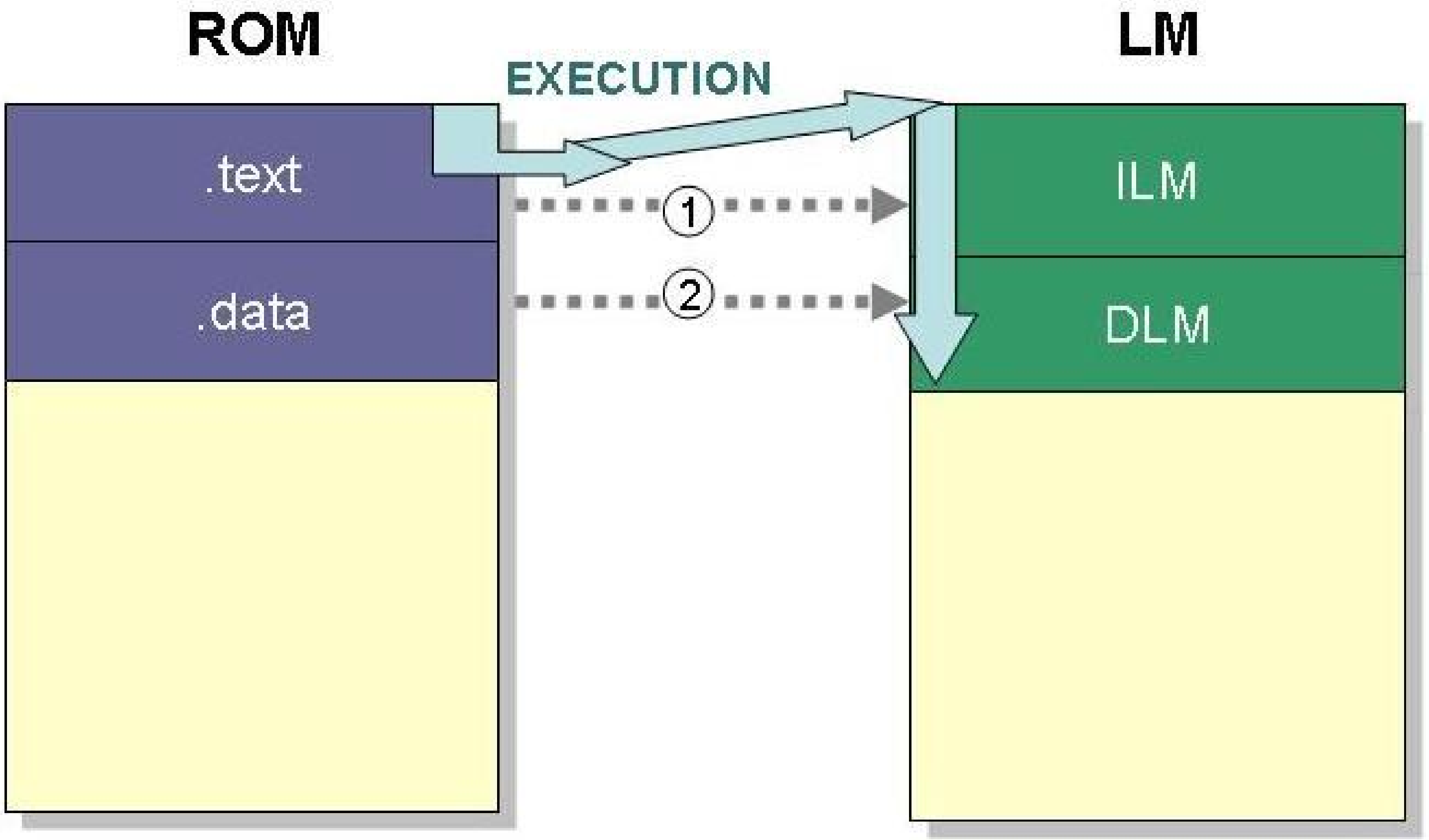
demo-ls2

- ❖ Boot from ROM and Copy-n-run from RAM



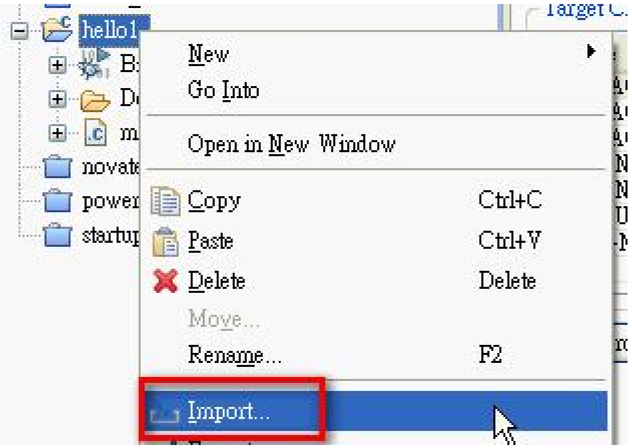
demo-ls3

- Boot from ROM and Copy-n-run from LM



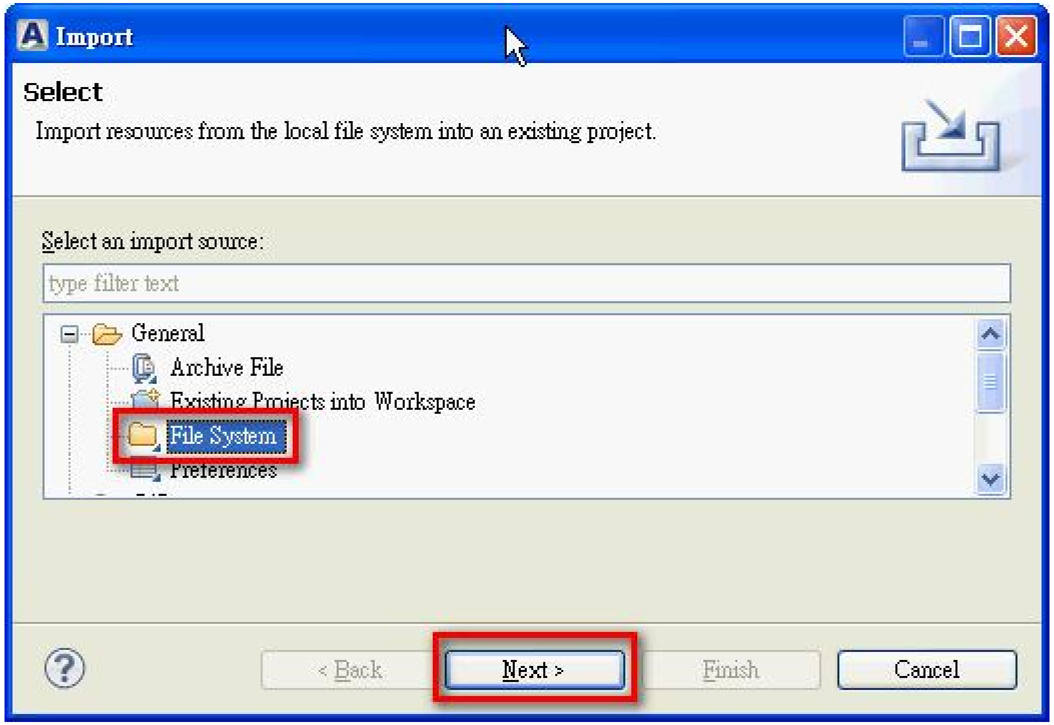
- How to import a program
  - From file system
  - From existing project

Import a program from file system (1)



Step1: Right click on project name and click "Import"

Step2: Select "File System"



Import

Select

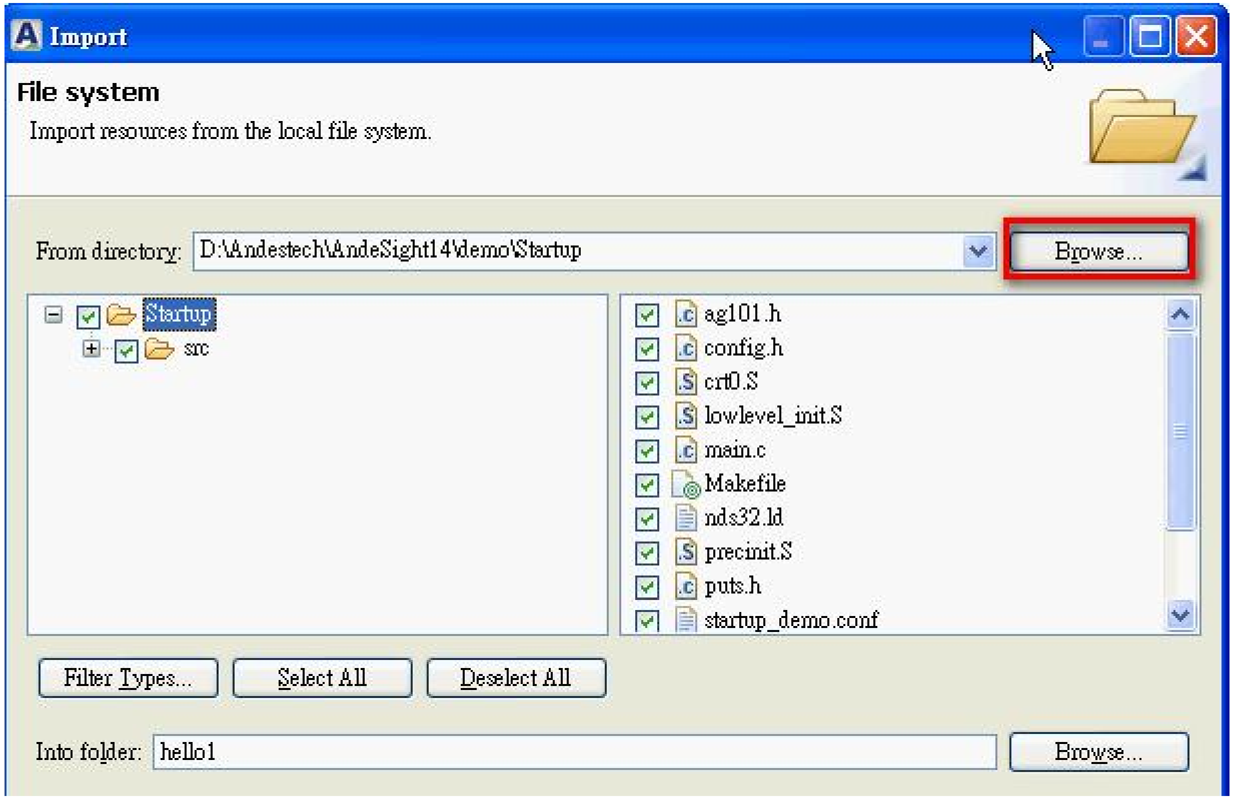
Import resources from the local file system into an existing project.

Select an import source:

type filter text

Import a program from file system (2)

Step3: Click Browse to select the desired files



Import

File system

Import resources from the local file system.

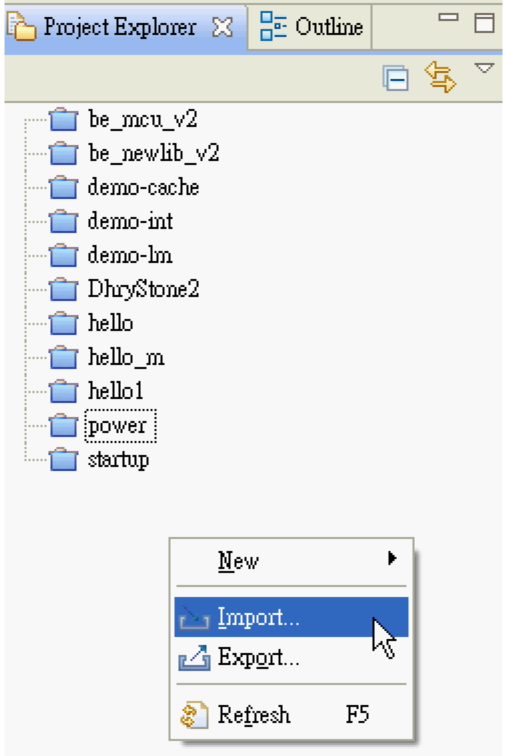
From directory: D:\Andestech\AndeSight14\demo\Startup

☑ Startup
☑ src

☑ ag101.h
☑ config.h
☑ crt0.S
☑ lowlevel_init.S
☑ main.c
☑ Makefile
☑ nds32.ld
☑ precinit.S
☑ puts.h
☑ startup_demo.conf

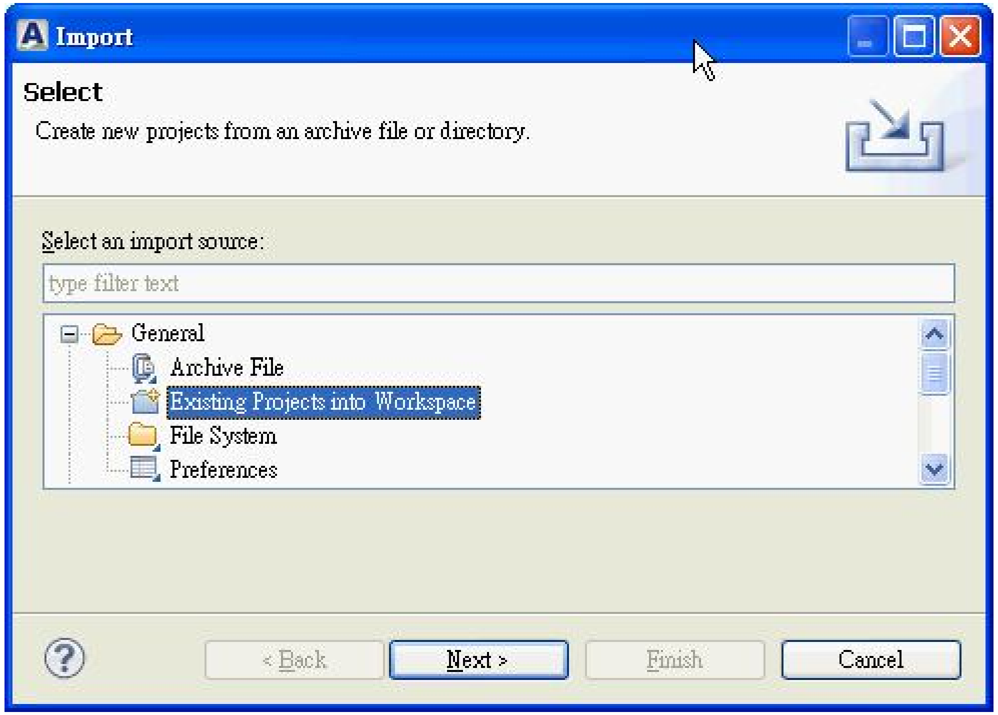
Into folder: hello1

Import a program from existing project (1)



Step1: Right click on Project Explorer and click “Import”

Step2: Select “Existing Projects into Workspace”


Import

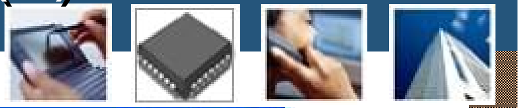
Select

Create new projects from an archive file or directory.

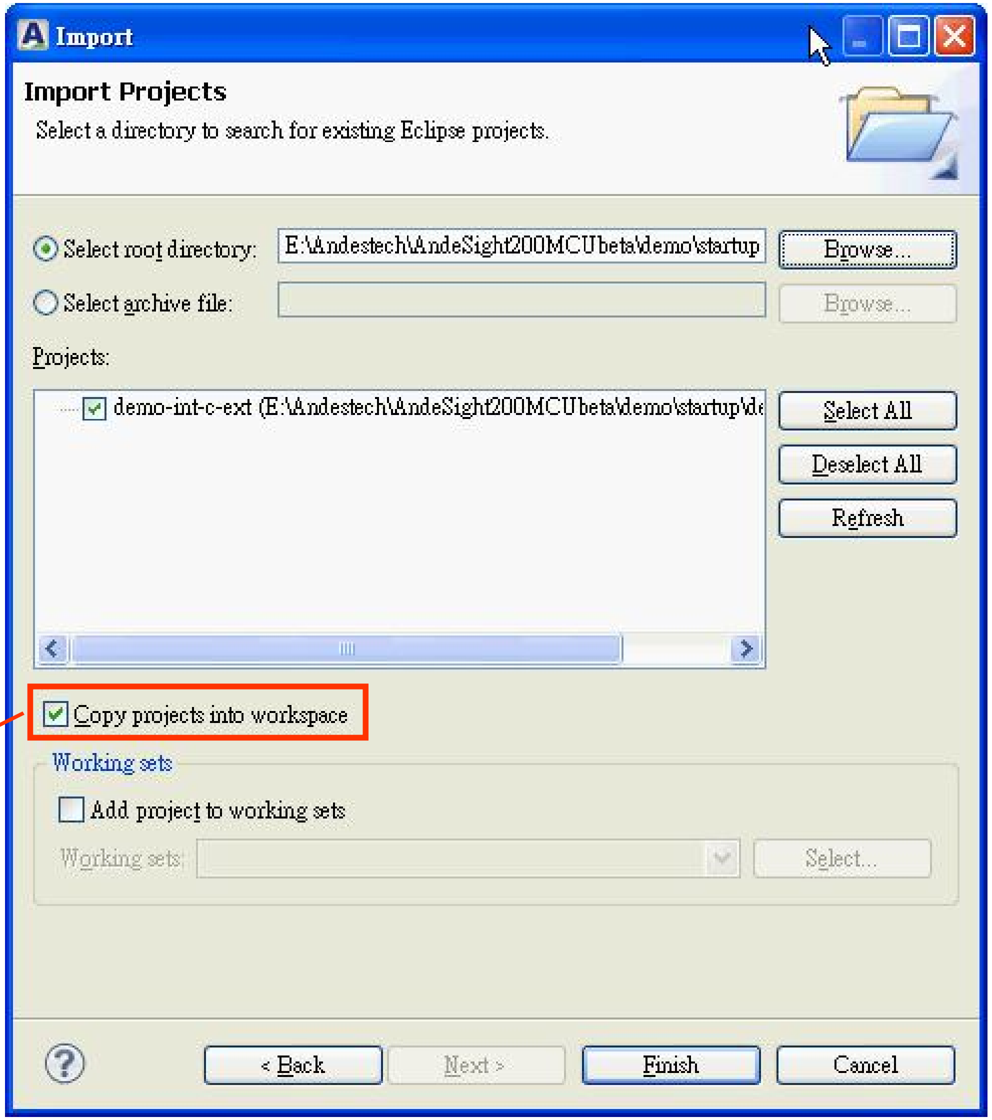
Select an import source:

Confidential

Import a program from existing project (2)



Step3: Click Browse to select the desired project



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: E:\Andestech\AndeSight200MCUbeta\demo\startup [Browse...]

○ Select archive file: [Browse...]

Projects:

☑ demo-int-c-ext (E:\Andestech\AndeSight200MCUbeta\demo\startup\de

☑ Copy projects into workspace

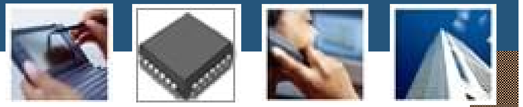
Working sets

☐ Add project to working sets

Working sets: [Select...]

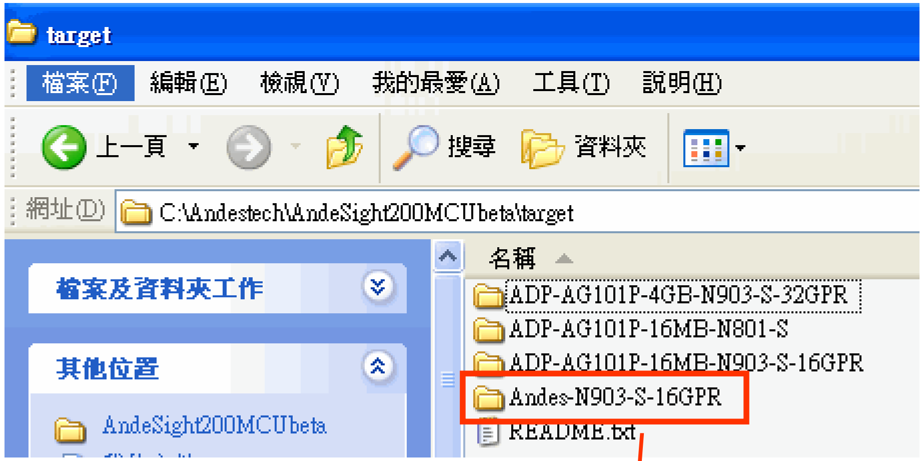
Copy projects into workspace

Confidential



- ❖ How to create Chip Profile
  - Chip Profile setting
  - How to use SOCgenerator

How to Create a Chip Profile



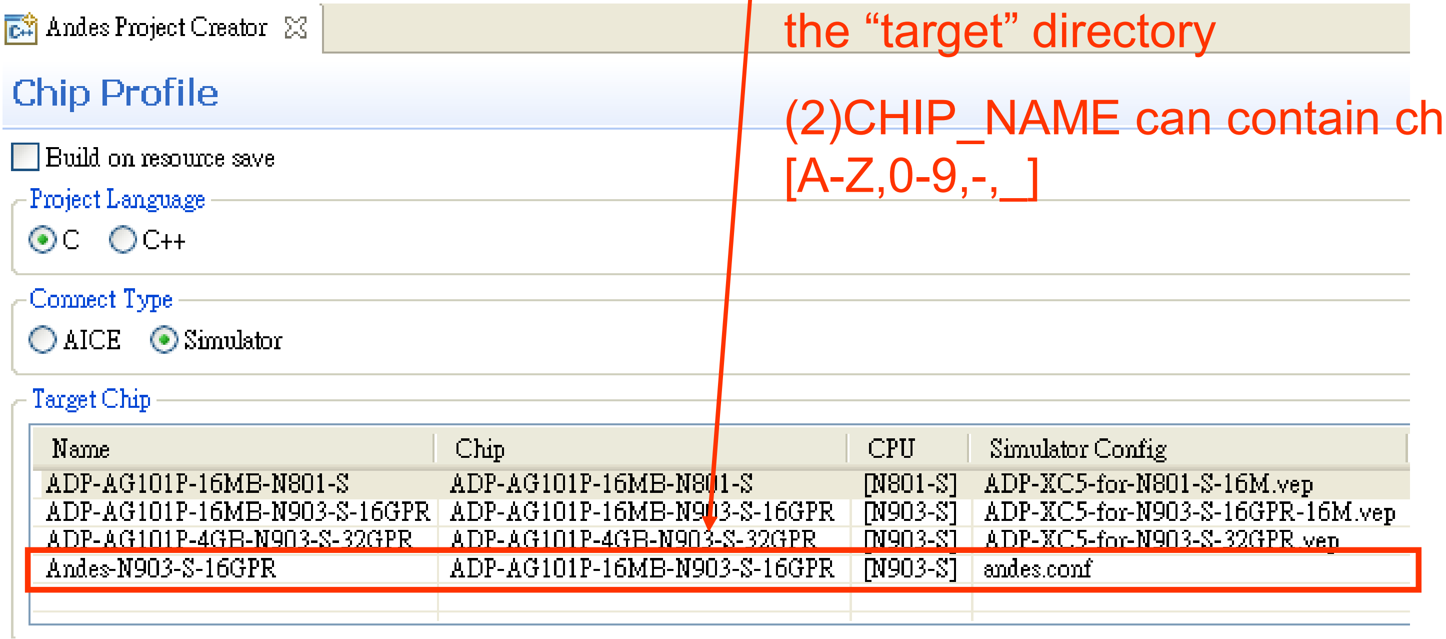
C:\Andestech\AndeSight200MCUbeta\target

名稱

- ADP-AG101P-4GB-N903-S-32GPR
- ADP-AG101P-16MB-N801-S
- ADP-AG101P-16MB-N903-S-16GPR
- Andes-N903-S-16GPR
- README.txt

(1) Create a CHIP_name directory under the “target” directory

(2)CHIP_NAME can contain characters [A-Z,0-9,-,_]



Chip Profile

□ Build on resource save

Project Language

● C ○ C++

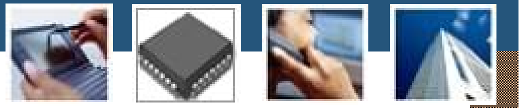
Connect Type

○ AICE ● Simulator

Target Chip

Name	Chip	CPU	Simulator Config
ADP-AG101P-16MB-N801-S	ADP-AG101P-16MB-N801-S	[N801-S]	ADP-XC5-for-N801-S-16M.vep
ADP-AG101P-16MB-N903-S-16GPR	ADP-AG101P-16MB-N903-S-16GPR	[N903-S]	ADP-XC5-for-N903-S-16GPR-16M.vep
ADP-AG101P-4GB-N903-S-32GPR	ADP-AG101P-4GB-N903-S-32GPR	[N903-S]	ADP-XC5-for-N903-S-32GPR.vep
Andes-N903-S-16GPR	ADP-AG101P-16MB-N903-S-16GPR	[N903-S]	andes.conf

Chip Profile for Pre-defined Targets



❖ Chip Profile contains all the necessary software settings

- Chip name
- Toolchain
- Flash driver
- Register file
- Memory map
- Simulator
- Linker script

Configure SoC Register View

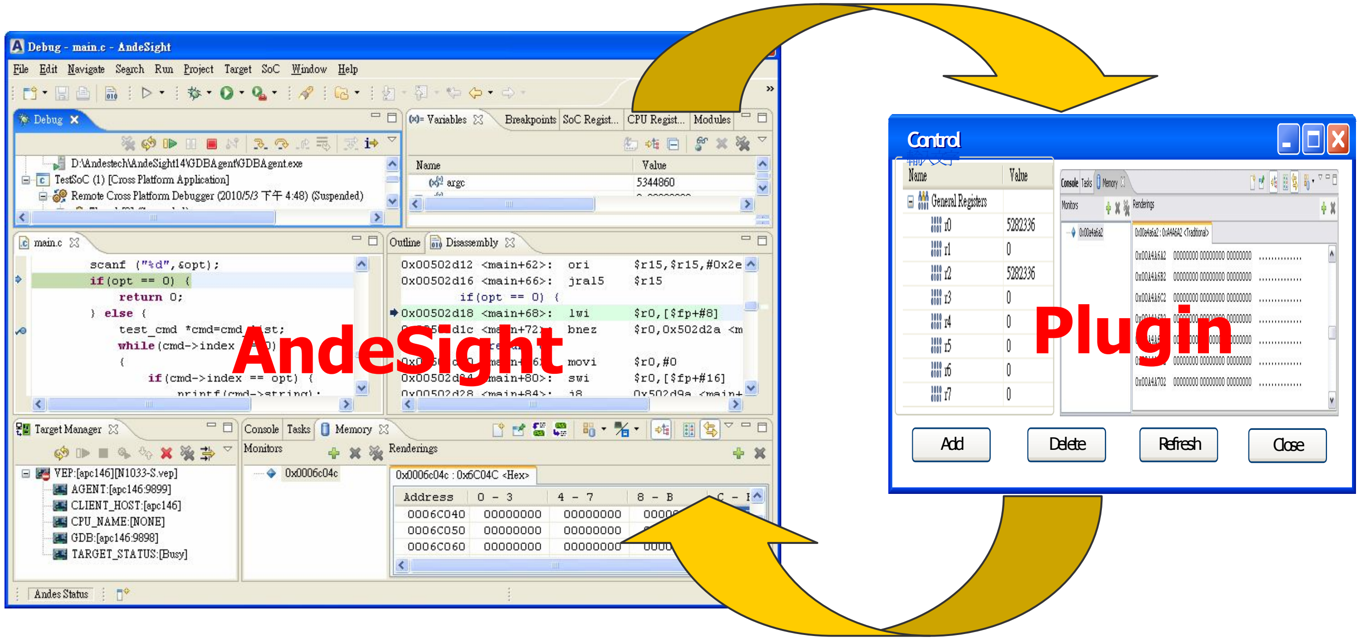
- AndeSight v2.0.0 allows users to configure the registers in SoC Register View. It involves the steps below:
  1. Install Python 2.7.x
  2. Create a description file for your SoC registers (ex. ADP-XC5.csv)
  3. Edit the description file
  4. Double-click the “generator.py” under ANDESIGHT_ROOT\SoC\SoCGenerator
  5. Rename the .regs file as “default.regs”

- Plug-in
  - ClientTCF demo

AndeSight™: Customization

❖Interface for customer plugin integration

- Allow customer’s plugins to manipulate/control the target status
- Leverage Eclipse protocol for future proof



D:\Andestech\AndeSight14\GDBAgent\GDBAgent.exe
TestSoC (1) [Cross Platform Application]
Remote Cross Platform Debugger (2010/5/3 下午 4:48) (Suspended)

Name	Value
argc	5344860

```
scanf ("%d",&opt);
if(opt == 0) {
    return 0;
} else {
    test_cmd *cmd=cmd_list;
    while(cmd->index != 0)
    {
        if(cmd->index == opt) {
            printf(cmd->string);
```

```
0x00502d12 <main+62>:  ori     $r15,$r15,#0x2e
0x00502d16 <main+66>:  jral5   $r15
        if(opt == 0) {
0x00502d18 <main+68>:  lwi     $r0,[$fp+#8]
0x00502d1c <main+72>:  bnez    $r0,0x502d2a <m
        return 0;
0x00502d20 <main+76>:  movi    $r0,#0
0x00502d24 <main+80>:  swi     $r0,[$fp+#16]
0x00502d28 <main+84>:  j8      0x502d9a <main+
```

VEP:[apc146][N1033-S.vep]
- AGENT:[apc146:9899]
- CLIENT_HOST:[apc146]
- CPU_NAME:[NONE]
- GDB:[apc146:9898]
- TARGET_STATUS:[Busy]

0x0006c04c

0x0006c04c : 0x6C04C <Hex>

Address	0 - 3	4 - 7	8 - B	C - F
0006C040	00000000	00000000	0000	
0006C050	00000000	00000000		
0006C060	00000000	00000000	0000	

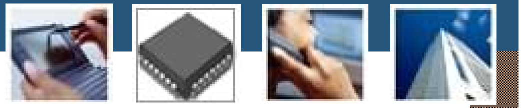
Name	Value
General Registers	
r0	5282336
r1	0
r2	5282336
r3	0
r4	0
r5	0
r6	0
r7	0

0x00a4a6a2

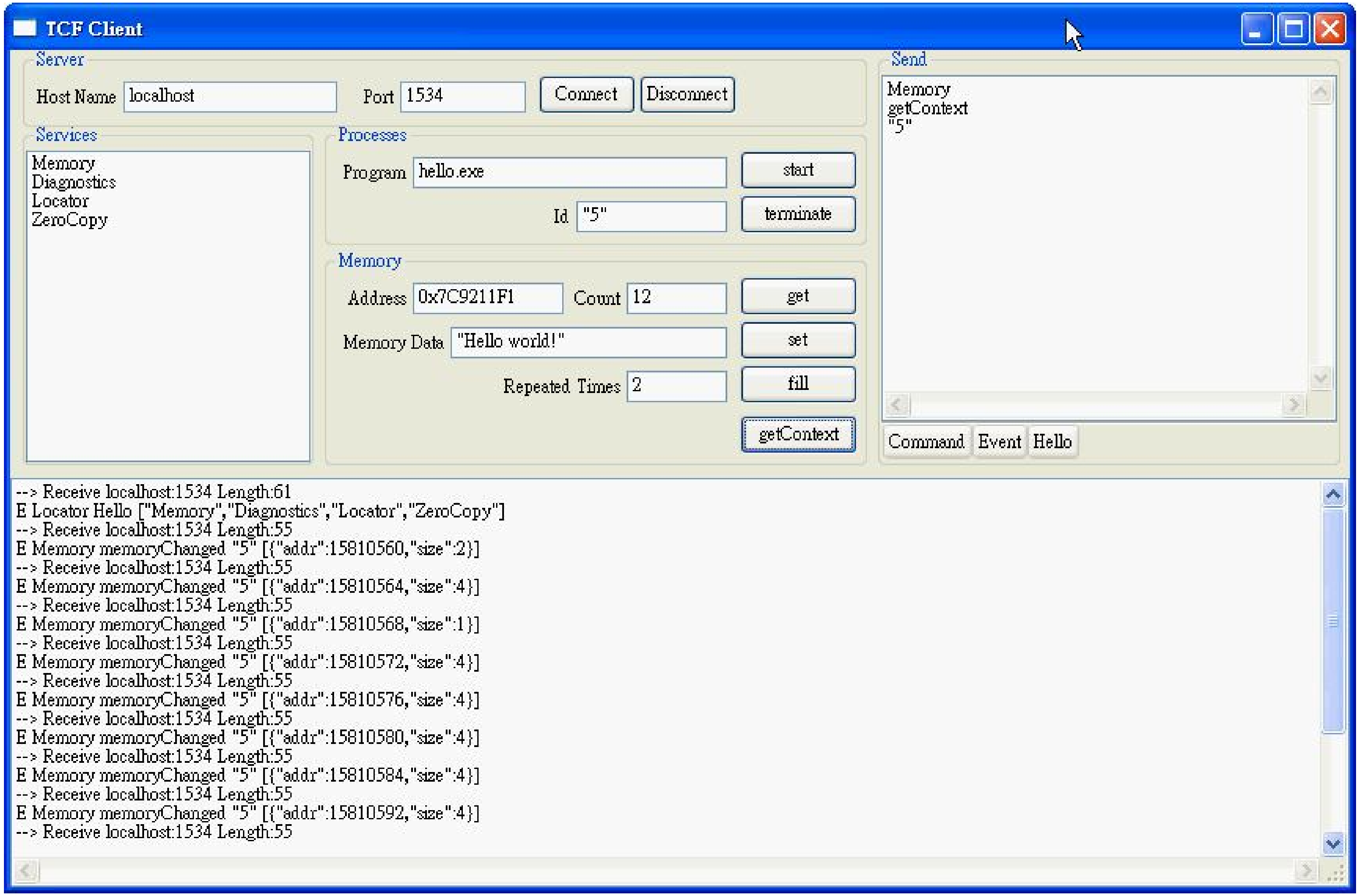
0x00a4a6a2 : 0xA4A6A2 <Traditional>

```
0x00A4A6A2  00000000 00000000 00000000  ..............
0x00A4A6B2  00000000 00000000 00000000  ..............
0x00A4A6C2  00000000 00000000 00000000  ..............
0x00A4A6D2  00000000 00000000 00000000  ..............
0x00A4A6E2  00000000 00000000 00000000  ..............
0x00A4A6F2  00000000 00000000 00000000  ..............
0x00A4A702  00000000 00000000 00000000  ..............
```

TCF Client



Path: AndeSight200MCUbeta\agent\clientTCF.exe.
It can communicate with AndeSight



Host Name: localhost
Port: 1534

Services:
Memory
Diagnostics
Locator
ZeroCopy

Program: hello.exe
Id: "5"

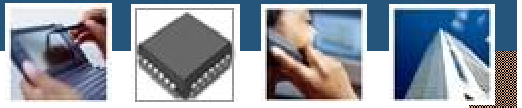
Address: 0x7C9211F1
Count: 12
Memory Data: "Hello world!"
Repeated Times: 2

Send:
```
Memory
getContext
"5"
```

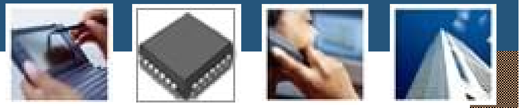
```
--> Receive localhost:1534 Length:61
E Locator Hello ["Memory","Diagnostics","Locator","ZeroCopy"]
--> Receive localhost:1534 Length:55
E Memory memoryChanged "5" [{"addr":15810560,"size":2}]
--> Receive localhost:1534 Length:55
E Memory memoryChanged "5" [{"addr":15810564,"size":4}]
--> Receive localhost:1534 Length:55
E Memory memoryChanged "5" [{"addr":15810568,"size":1}]
--> Receive localhost:1534 Length:55
E Memory memoryChanged "5" [{"addr":15810572,"size":4}]
--> Receive localhost:1534 Length:55
E Memory memoryChanged "5" [{"addr":15810576,"size":4}]
--> Receive localhost:1534 Length:55
E Memory memoryChanged "5" [{"addr":15810580,"size":4}]
--> Receive localhost:1534 Length:55
E Memory memoryChanged "5" [{"addr":15810584,"size":4}]
--> Receive localhost:1534 Length:55
E Memory memoryChanged "5" [{"addr":15810592,"size":4}]
--> Receive localhost:1534 Length:55
```

Confidential

Plug-in

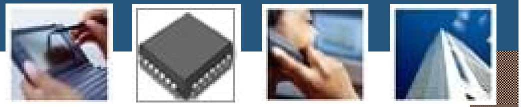


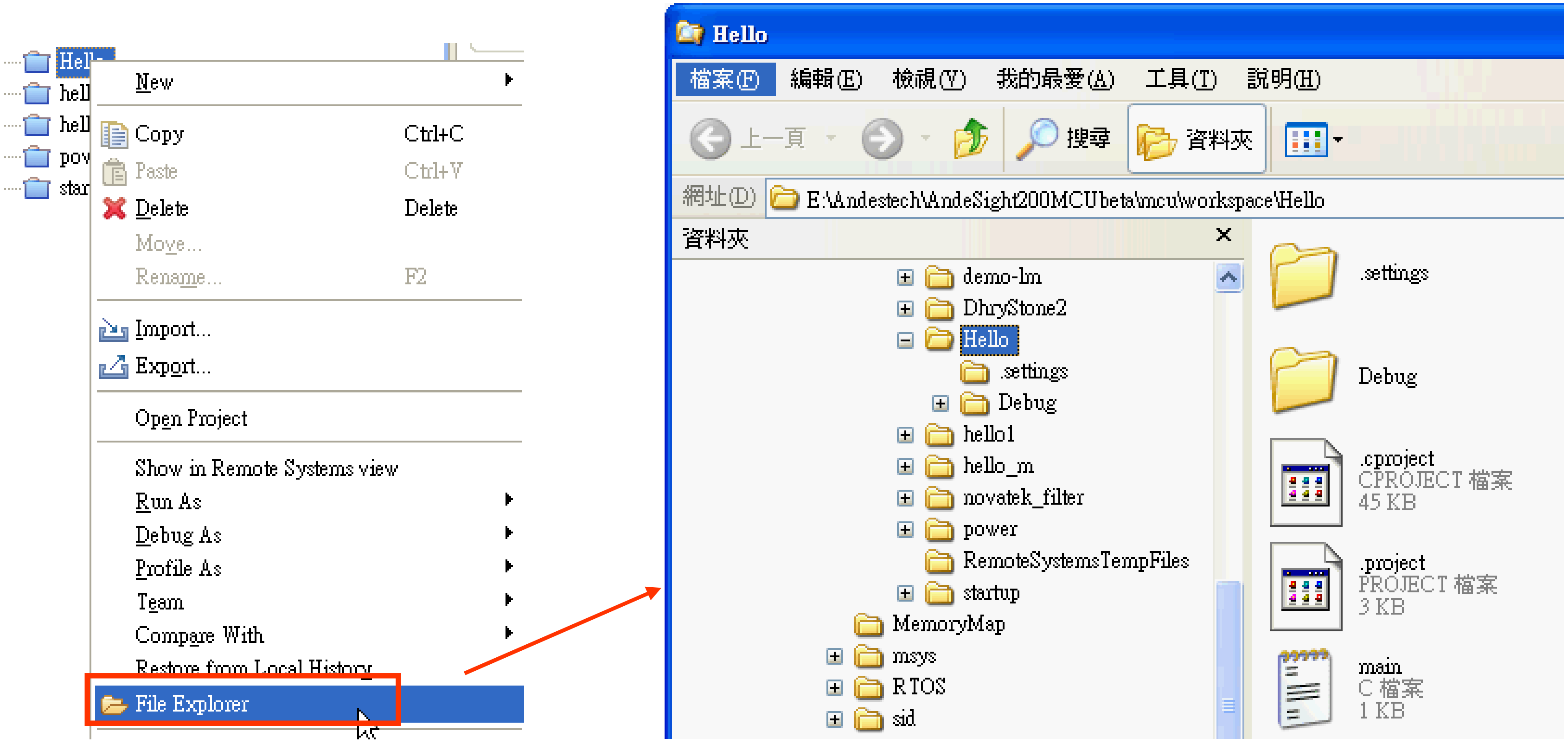
- Document:
  - AndeSight200MCUbeta\agent\Plugin_Integration_Guide_doc_v0.1.pdf
- Slide:
  - AndeSight200MCUbeta\agent\Plugin_Integration_Guide_ppt_v0.1.pdf



- Some tools:
  - file explorer
  - open element
  - trace symbol
- Resource on Internet

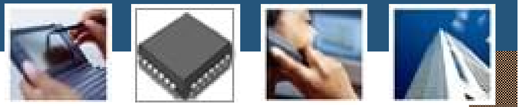
File Explorer

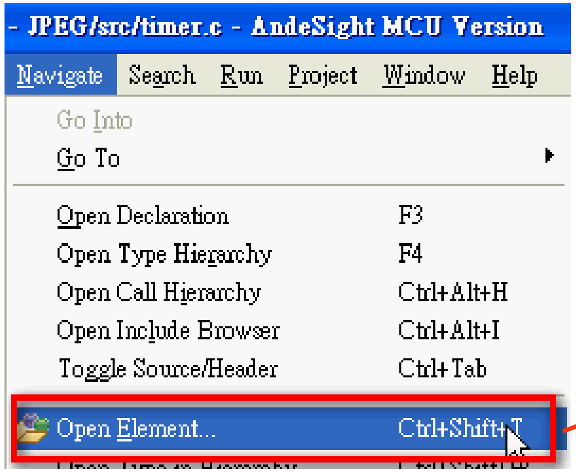




Quick open File Explorer

Open Element

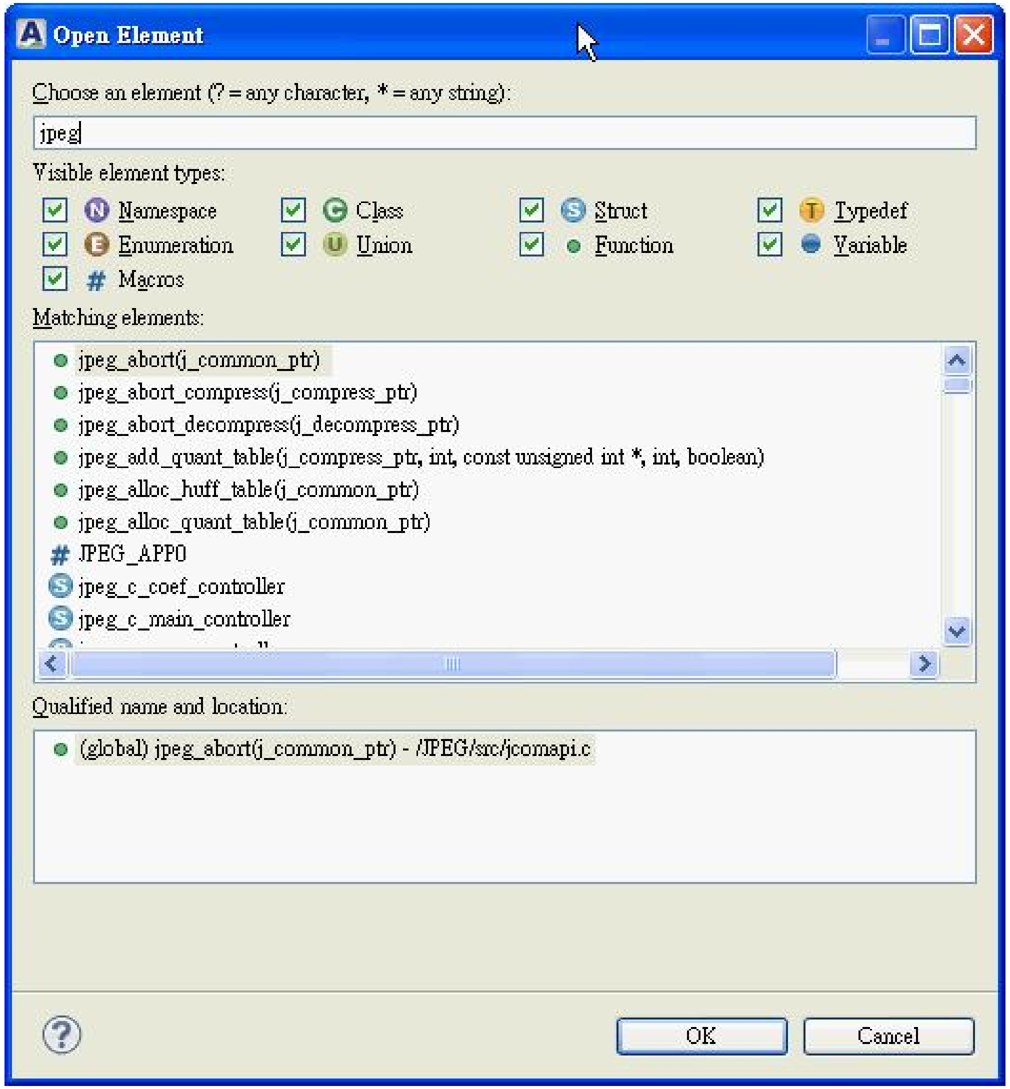




- JPEG/src/timer.c - AndeSight MCU Version
Navigate
Search
Run
Project
Window
Help
Go Into
Go To
Open Declaration F3
Open Type Hierarchy F4
Open Call Hierarchy Ctrl+Alt+H
Open Include Browser Ctrl+Alt+I
Toggle Source/Header Ctrl+Tab
Open Element... Ctrl+Shift+T

A convenient tool to search function

Note: This function is hidden in beta version



Choose an element (? = any character, * = any string):

jpeg

Visible element types:

☑ Namespace ☑ Class ☑ Struct ☑ Typedef
☑ Enumeration ☑ Union ☑ Function ☑ Variable
☑ Macros

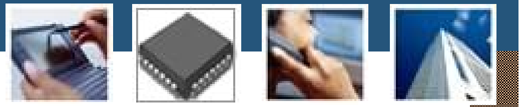
Matching elements:

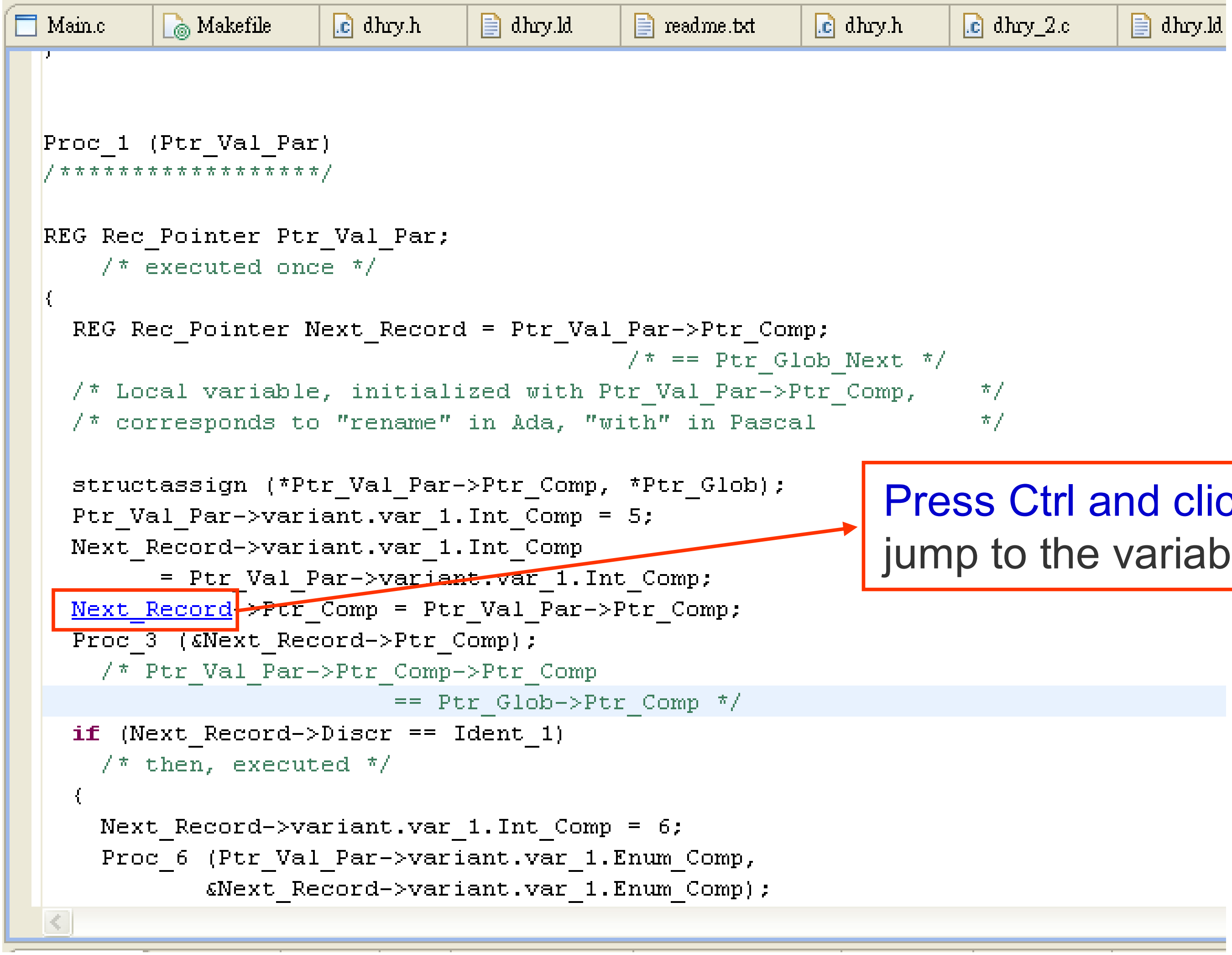
- jpeg_abort(j_common_ptr)
- jpeg_abort_compress(j_compress_ptr)
- jpeg_abort_decompress(j_decompress_ptr)
- jpeg_add_quant_table(j_compress_ptr, int, const unsigned int *, int, boolean)
- jpeg_alloc_huff_table(j_common_ptr)
- jpeg_alloc_quant_table(j_common_ptr)
- JPEG_APP0
- jpeg_c_coef_controller
- jpeg_c_main_controller

Qualified name and location:

- (global) jpeg_abort(j_common_ptr) - /JPEG/src/jcomapi.c

Trace symbol

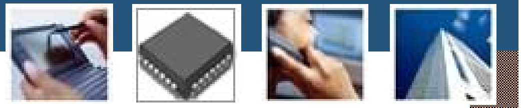




```
}


Proc_1 (Ptr_Val_Par)
/*******************/

REG Rec_Pointer Ptr_Val_Par;
    /* executed once */
{
  REG Rec_Pointer Next_Record = Ptr_Val_Par->Ptr_Comp;
                                      /* == Ptr_Glob_Next */
  /* Local variable, initialized with Ptr_Val_Par->Ptr_Comp,    */
  /* corresponds to "rename" in Ada, "with" in Pascal           */

  structassign (*Ptr_Val_Par->Ptr_Comp, *Ptr_Glob);
  Ptr_Val_Par->variant.var_1.Int_Comp = 5;
  Next_Record->variant.var_1.Int_Comp
        = Ptr_Val_Par->variant.var_1.Int_Comp;
  Next_Record->Ptr_Comp = Ptr_Val_Par->Ptr_Comp;
  Proc_3 (&Next_Record->Ptr_Comp);
    /* Ptr_Val_Par->Ptr_Comp->Ptr_Comp
                        == Ptr_Glob->Ptr_Comp */
  if (Next_Record->Discr == Ident_1)
    /* then, executed */
  {
    Next_Record->variant.var_1.Int_Comp = 6;
    Proc_6 (Ptr_Val_Par->variant.var_1.Enum_Comp,
           &Next_Record->variant.var_1.Enum_Comp);
```

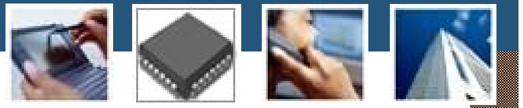
Press Ctrl and click variable can jump to the variable definition

Resources on internet

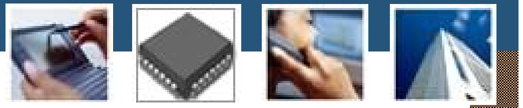


- ❖ Andes Workshop
  - http://forum.andestech.com/
- ❖ Andes Core 32bit RISC CPU
  - http://andescore.blogspot.com/
- ❖ Andes Core 台灣心 AndesCore的兩三事
  - http://nckuhuahua.pixnet.net/blog

Andes OSDK



- **Toolchain:**
  - binutils 2.15.19, gcc 3.4.4, glibc 2.3.5 and gdb 6.8
- **OS:** Linux 2.6.27
- **Emulator:** Qemu 0.9.1
  - Support AndeStar ISA/SPA V2
  - Peripherals: Interrupt Controller, Timer, UART, LCD, MAC, SD, Touch Screen, and SSP
- **Demo Apps:**
  - Frame buffer viewer, MPlayer, GDBserver
- **Documents:**
  - Programming Guide, ISA spec., OSDK Developer's Guide
- **Free download available http://osdk.andestech.com**

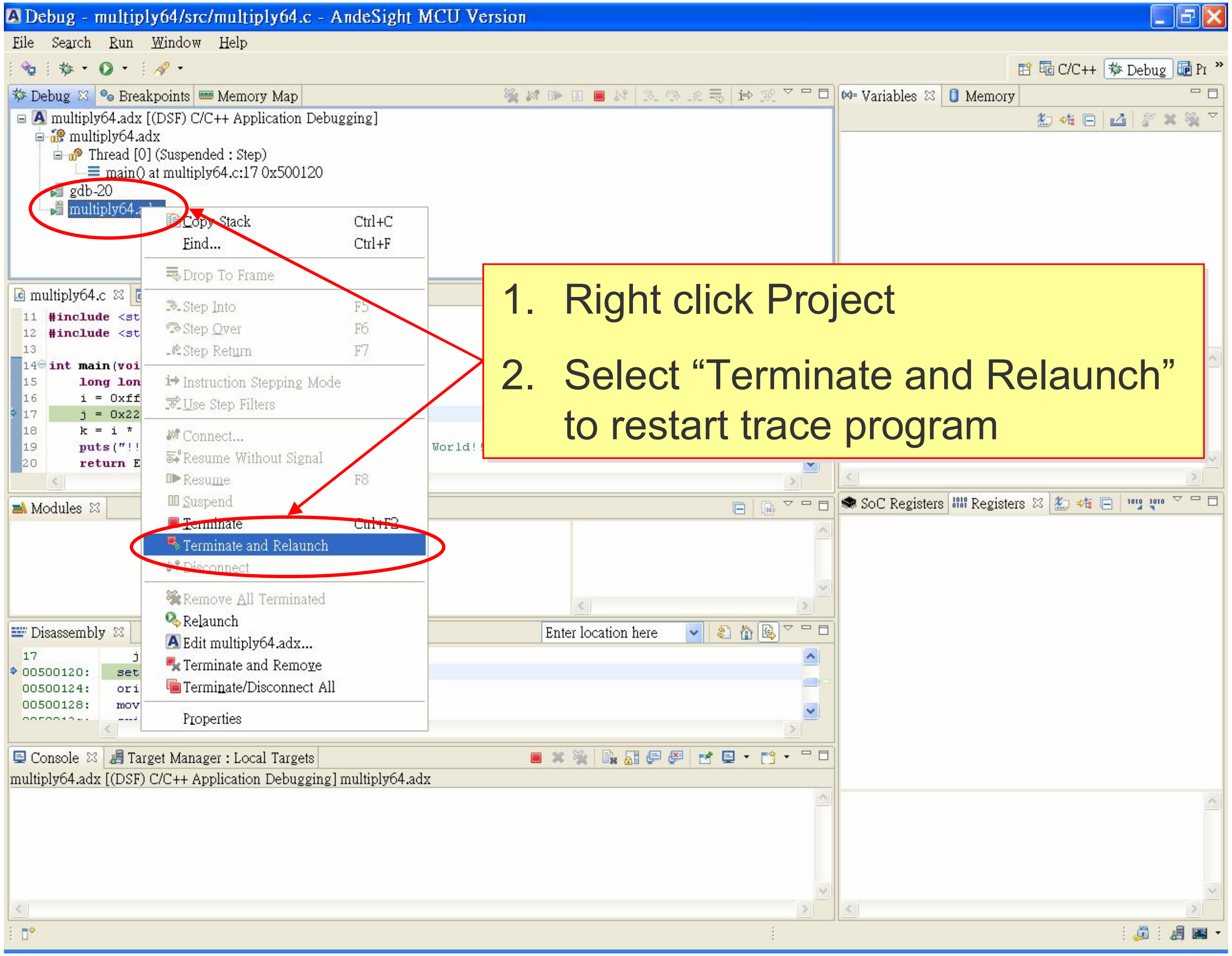


Thank You !

Andes Technology
Tel: +886-3-6668300
Business : sales@andestech.com
Technical : support@andestech.com

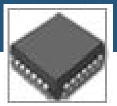
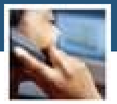
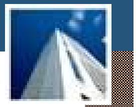
AndeSight Debug Operation

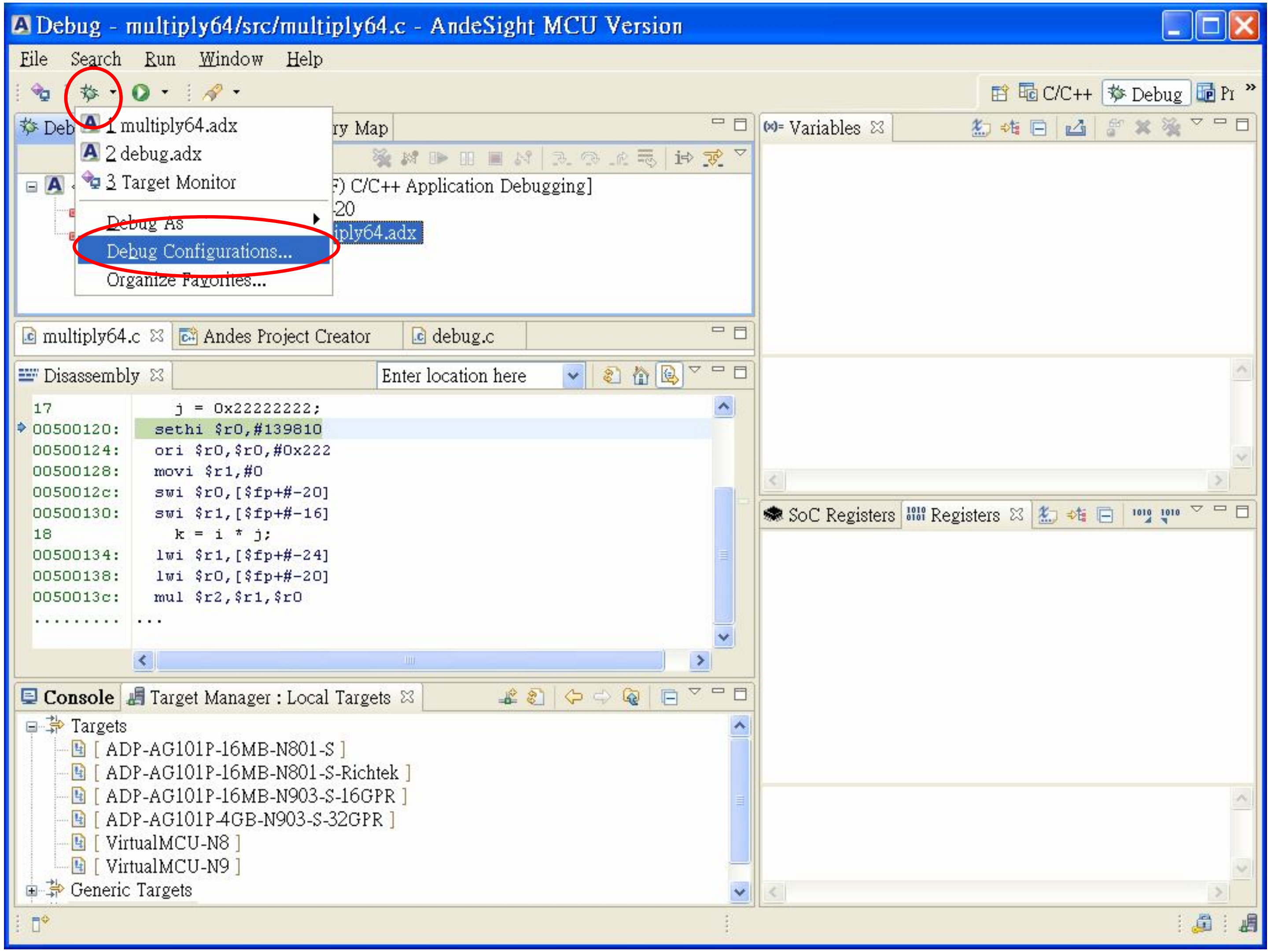
Terminate and Relaunch



1. Right click Project
2. Select “Terminate and Relaunch” to restart trace program

Build all + debug (1)

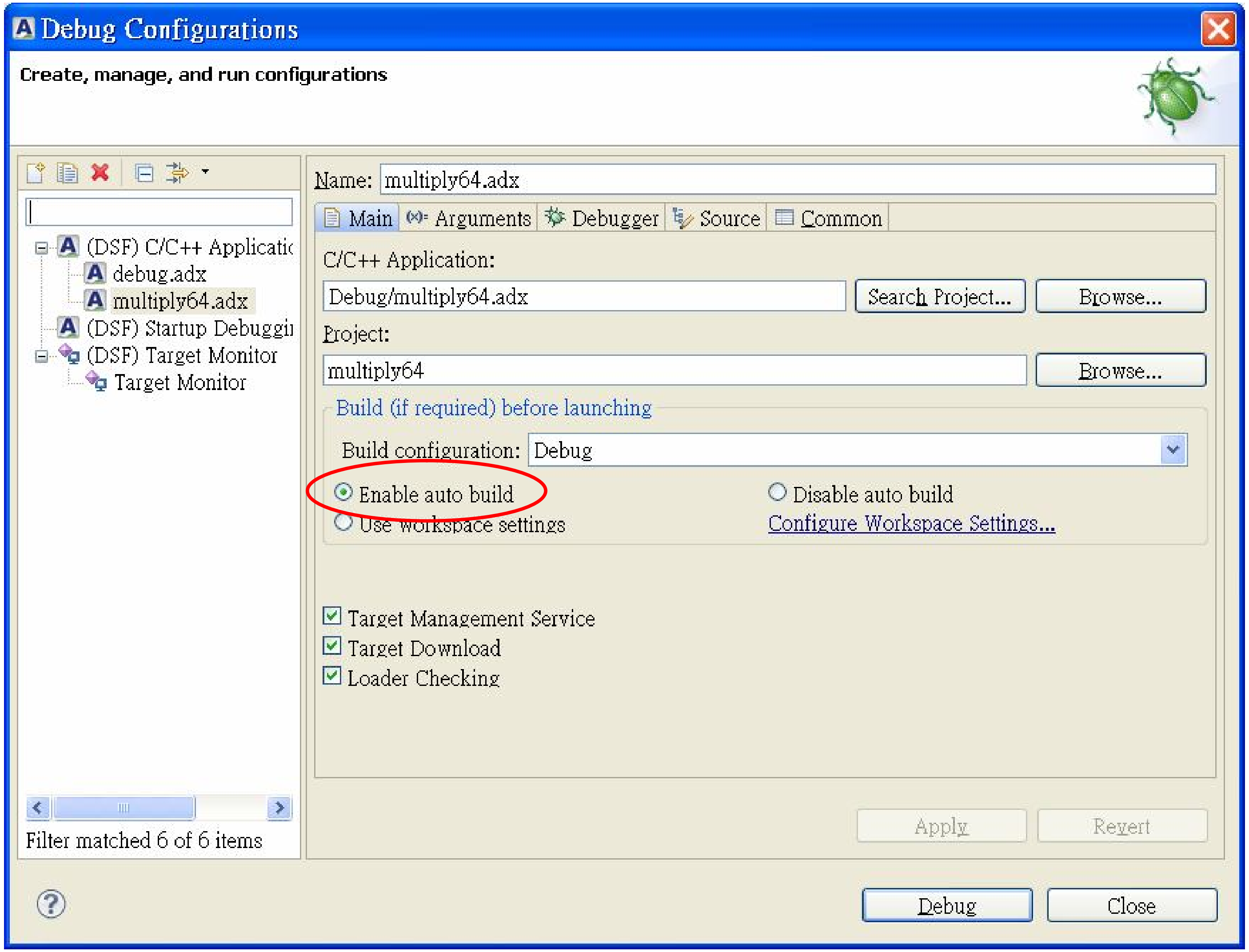






```
17                 j = 0x22222222;
00500120:   sethi $r0,#139810
00500124:   ori $r0,$r0,#0x222
00500128:   movi $r1,#0
0050012c:   swi $r0,[$fp+#-20]
00500130:   swi $r1,[$fp+#-16]
18                 k = i * j;
00500134:   lwi $r1,[$fp+#-24]
00500138:   lwi $r0,[$fp+#-20]
0050013c:   mul $r2,$r1,$r0
.........   ...
```

- Targets
  - [ADP-AG101P-16MB-N801-S]
  - [ADP-AG101P-16MB-N801-S-Richtek]
  - [ADP-AG101P-16MB-N903-S-16GPR]
  - [ADP-AG101P-4GB-N903-S-32GPR]
  - [VirtualMCU-N8]
  - [VirtualMCU-N9]
- Generic Targets

Build all + debug (2)



Debug Configurations

Create, manage, and run configurations

- (DSF) C/C++ Applicatic
  - debug.adx
  - multiply64.adx
- (DSF) Startup Debuggi
- (DSF) Target Monitor
  - Target Monitor

Filter matched 6 of 6 items

Name: multiply64.adx

C/C++ Application: Debug/multiply64.adx

Project: multiply64

Build (if required) before launching

Build configuration: Debug

● Enable auto build ○ Disable auto build

○ Use workspace settings Configure Workspace Settings...

☑ Target Management Service

☑ Target Download

☑ Loader Checking