RISC-V CON

ONLINE WEBINAR

Andes Technology Andes Software Solutions for RISC-V

Solution Architecture Division

Niraj Dengale

Andes Corporate Overview

Silicon Valley Ties	• Core R&D from AMD, DEC, Intel, MIPS, nVidia, and Sun
15-Year CPU IP Company	• IPO in 2017; HQ in Taiwan • AndeStar™ V1-V3, V5 (RISC-V)
>1 Bn Annual Run Rate of Andes-Embedded SoC	• ~300 customers in TW, CN, US, EU, JP, KR
Founding Platinum Member and Major Contributor	• Chairing Task Groups • Contributing to GNU, LLVM, uBoot, glibc, Linux, etc.

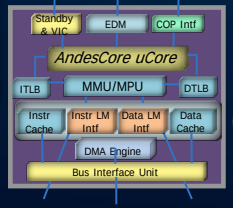
Andes Product Overview

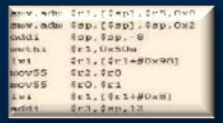
Best extensions to RISC-V

AndeStar™ V5 Architecture

Highly optimized design with leading PPA

AndesCore™ Processors





AndeSight™ Tools

Professional IDE with high code quality

Handy peripheral IPs to speed up SoC construction

AndeShape™ Platforms

AndeSoft™ Stacks

Extensive SW stacks from bare metal, RTOS to Linux

Agenda

- Overview of Tools and Runtime Support
- AndeSight™ IDE: Simulator, Compilation and debugging
- AndeSoft™ BSP: Bare Metal, RTOS, Linux and DSP ISA
- Summary

SoC SW Development Environment

AndeSight™ IDE

- Eclipse-Based IDE
- Streamlined GUI
- Feature-rich Editor
- Managed Build System
- Optimized Toolchains
- Source Level Debugger
- Profiling Analysis
- Extensive Demo Projects
- Flash ISP

build executables

debug interactively

AndeSoft™ SW Stack

- Application Layers
- Middleware
- Drivers | App Drivers
- OS/Kernel | Libraries

Andes/Partners' Solutions

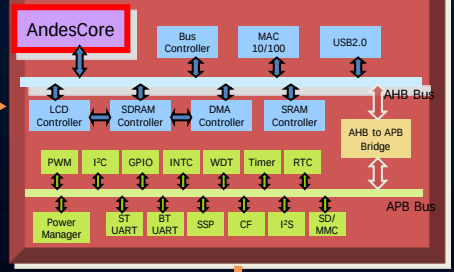
Customers' Designs

AICE

Virtual SoC Configuration

SoC based on AndesCore™

Virtual SoC



AndeShape™

Profiling/Tracing/Debugging data

Overview of Tools and Runtime Support

AndeSight™ IDE

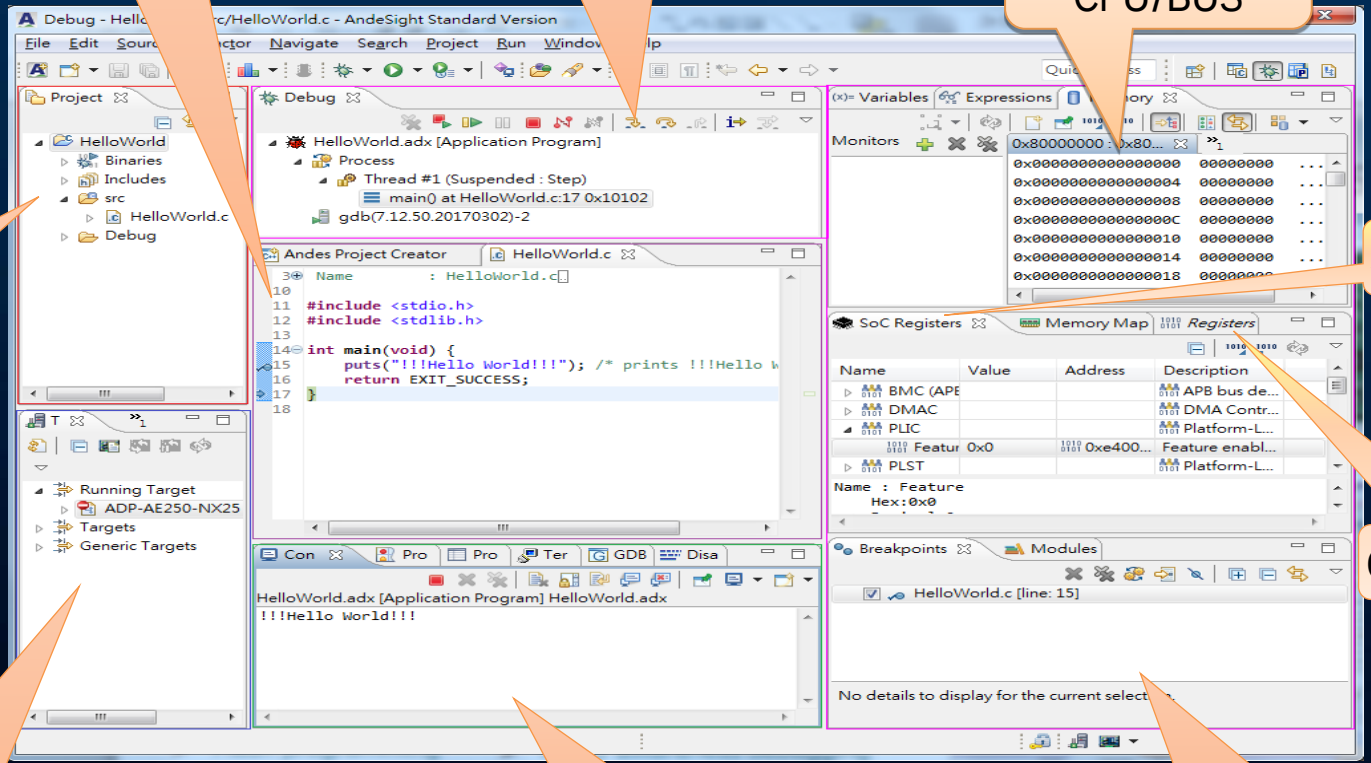
- **Simulator:** AndeSim (near-cycle accuracy), Qemu
- **Compilation:** GNU toolchain, LLVM compiler/linker, optimized MCU library, DSP library
- **Debugging:** GDB, speed-optimized OpenOCD, USB-to-JTAG ICE cable

AndeSoft™ BSP

- Several Bare metal sample projects for Andes-specific features
- **RTOS:** FreeRTOS, Zephyr, LiteOS, RT-Thread
- **Linux:** MMU/TLB support, LTP tested
- Arduino support for Andes Corvette-F1 FPGA board

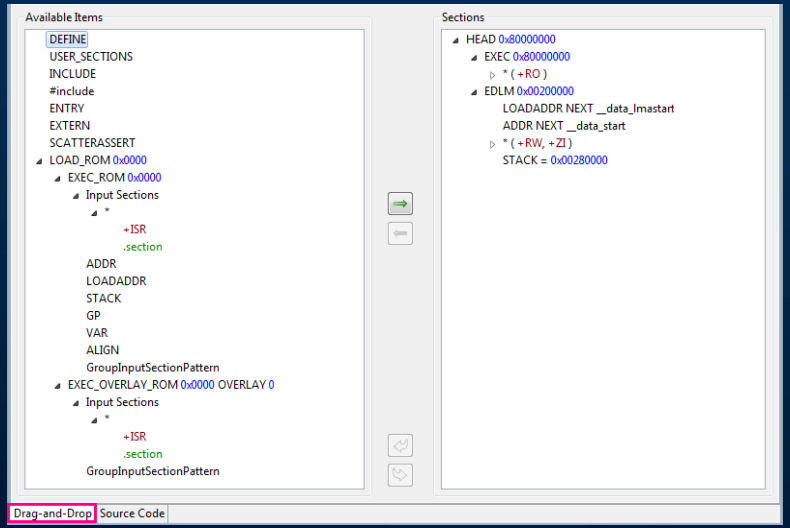
AndeSight™ IDE

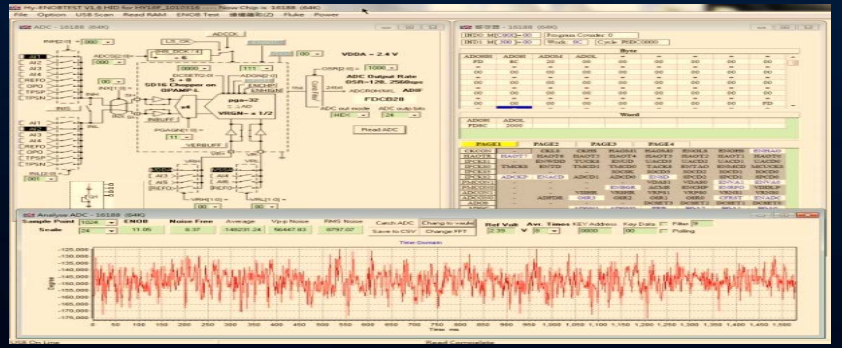
Comprehensive Development Environment


Editor View
Calling Stack
Memory from CPU/BUS
Project Explorer
SoC Registers
CPU Registers
Target Manager
Console
Breakpoint
Debug - Hello/HelloWorld.c - AndeSight Standard Version
File Edit Source Navigate Search Project Run Window Help
Project
HelloWorld
Binaries
Includes
src
HelloWorld.c
Debug
Debug
HelloWorld.adx [Application Program]
Process
Thread #1 (Suspended : Step)
main() at HelloWorld.c:17 0x10102
gdb(7.12.50.20170302)-2
Variables
Expressions
Monitors
0x80000000 : 0x80...
0x0000000000000000 00000000
0x0000000000000004 00000000
0x0000000000000008 00000000
0x000000000000000C 00000000
0x0000000000000010 00000000
0x0000000000000014 00000000
0x0000000000000018 00000000
Andes Project Creator
HelloWorld.c
Name : HelloWorld.c
#include <stdio.h>
#include <stdlib.h>
int main(void) {
puts("!!!Hello World!!!"); /* prints !!!Hello W
return EXIT_SUCCESS;
}
SoC Registers
Memory Map
Registers
Name Value Address Description
BMC (APE APB bus de...
DMAC DMA Contr...
PLIC Platform-L...
Featur 0x0 0xe400... Feature enabl...
PLST Platform-L...
Name : Feature
Hex:0x0
Running Target
ADP-AE250-NX25
Targets
Generic Targets
Con
Pro
Pro
Ter
GDB
Disa
HelloWorld.adx [Application Program] HelloWorld.adx
!!!Hello World!!!
Breakpoints
Modules
HelloWorld.c [line: 15]
No details to display for the current select

AndeSight™: Professional IDE

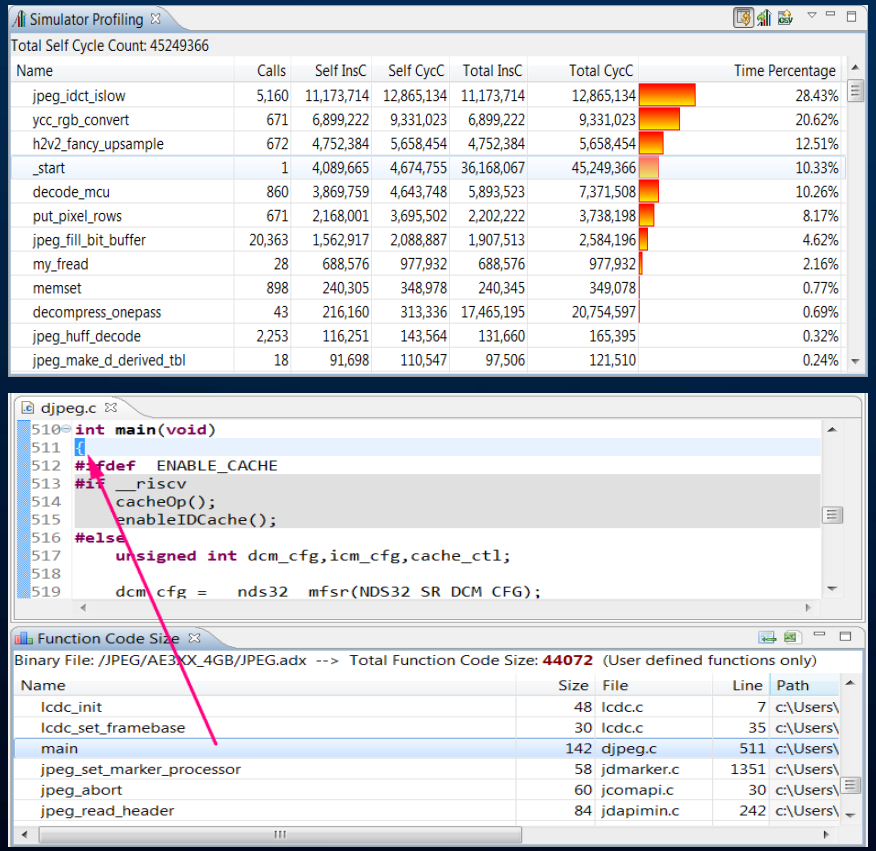
- **Project Setup:**
  - Meta linker script editor
  - Flash ISP configured through GUI
- **Debug Support:**
  - Virtual hosting
  - Register Bitfield display/update
  - Break-n-Display on exceptions
  - Script-Based RTOS awareness
  - Stack protection handling
- **Custom Plugin Interface**





AndeSight™: Profiling

- **Program Analysis**
  - Function Profiling
  - Code Coverage
  - Performance Meter
  - Function Code Size
  - (Static) Stack Size



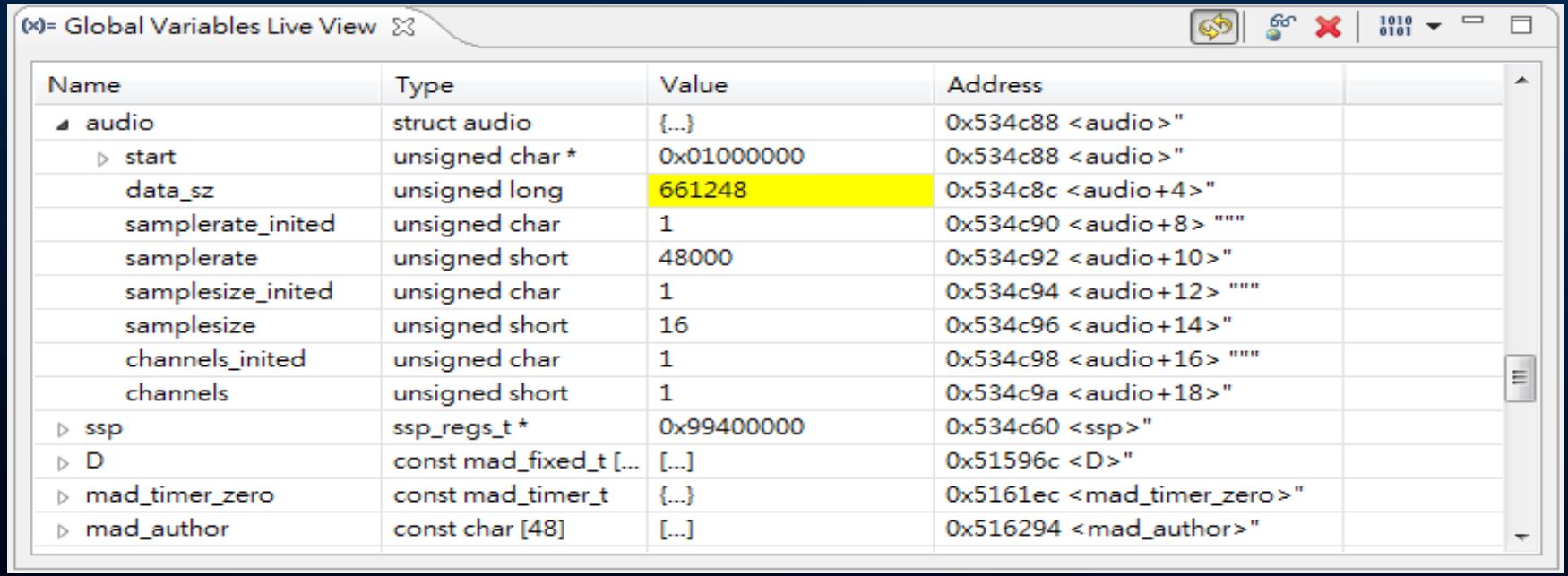
Name	Calls	Self InsC	Self CycC	Total InsC	Total CycC	Time Percentage
jpeg_idct_islow	5,160	11,173,714	12,865,134	11,173,714	12,865,134	28.43%
ycc_rgb_convert	671	6,899,222	9,331,023	6,899,222	9,331,023	20.62%
h2v2_fancy_upsample	672	4,752,384	5,658,454	4,752,384	5,658,454	12.51%
_start	1	4,089,665	4,674,755	36,168,067	45,249,366	10.33%
decode_mcu	860	3,869,759	4,643,748	5,893,523	7,371,508	10.26%
put_pixel_rows	671	2,168,001	3,695,502	2,202,222	3,738,198	8.17%
jpeg_fill_bit_buffer	20,363	1,562,917	2,088,887	1,907,513	2,584,196	4.62%
my_fread	28	688,576	977,932	688,576	977,932	2.16%
memset	898	240,305	348,978	240,345	349,078	0.77%
decompress_onepass	43	216,160	313,336	17,465,195	20,754,597	0.69%
jpeg_huff_decode	2,253	116,251	143,564	131,660	165,395	0.32%
jpeg_make_d_derived_tbl	18	91,698	110,547	97,506	121,510	0.24%

```
510 int main(void)
511 {
512 #ifdef  ENABLE_CACHE
513 #if __riscv
514     cacheOp();
515     enableIDCache();
516 #else
517     unsigned int dcm_cfg,icm_cfg,cache_ctl;
518
519     dcm_cfg =   nds32  mfsr(NDS32 SR DCM CFG);
```

Name	Size	File	Line	Path
lcdc_init	48	lcdc.c	7	c:\Users\
lcdc_set_framebase	30	lcdc.c	35	c:\Users\
main	142	djpeg.c	511	c:\Users\
jpeg_set_marker_processor	58	jdmarker.c	1351	c:\Users\
jpeg_abort	60	jcomapi.c	30	c:\Users\
jpeg_read_header	84	jdapimin.c	242	c:\Users\

Global Variables Live View

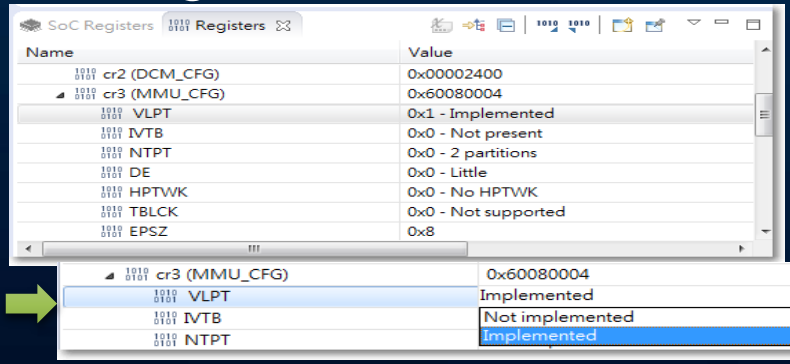
- Runtime updating global variables in a fixed interval
- Highlight the changes in yellow



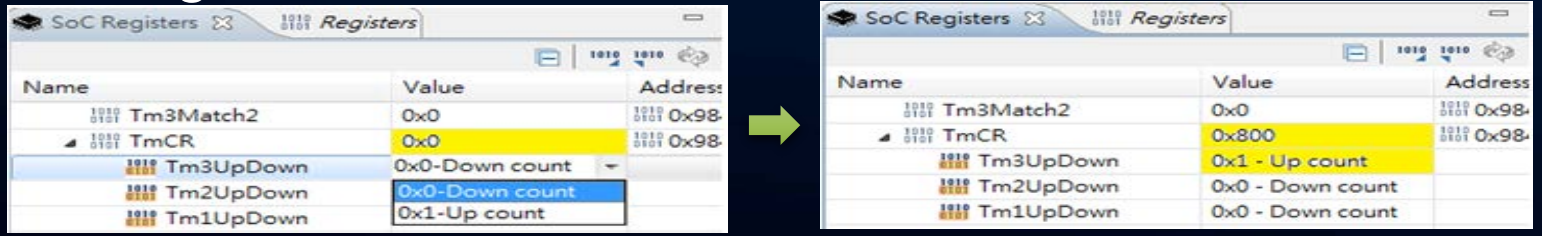
Name	Type	Value	Address
audio	struct audio	{...}	0x534c88 <audio>"
start	unsigned char *	0x01000000	0x534c88 <audio>"
data_sz	unsigned long	661248	0x534c8c <audio+4>"
samplerate_inited	unsigned char	1	0x534c90 <audio+8> """
samplerate	unsigned short	48000	0x534c92 <audio+10>"
samplesize_inited	unsigned char	1	0x534c94 <audio+12> """
samplesize	unsigned short	16	0x534c96 <audio+14>"
channels_inited	unsigned char	1	0x534c98 <audio+16> """
channels	unsigned short	1	0x534c9a <audio+18>"
ssp	ssp_regs_t *	0x99400000	0x534c60 <ssp>"
D	const mad_fixed_t [...	[...]	0x51596c <D>"
mad_timer_zero	const mad_timer_t	{...}	0x5161ec <mad_timer_zero>"
mad_author	const char [48]	[...]	0x516294 <mad_author>"

Register Bitfield Viewing and Update

- **Bit Fields Display and Update**
  - CPU registers
  - SoC registers
- **CPU registers**



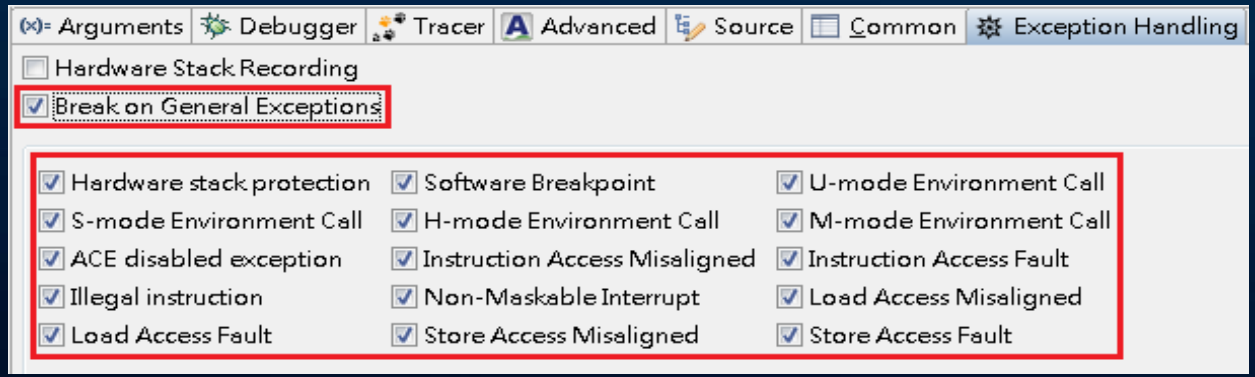
- **SoC registers**



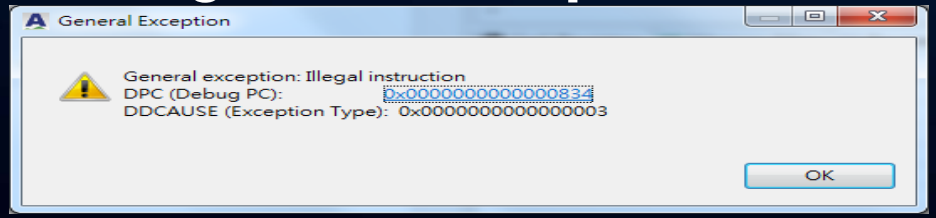
- **Benefits**
  - Bit fields can be modified at runtime
  - Description of Bit fields can be shown
  - Good for debugging and programming

General Exception Handling

- **It helps user to catch the root cause with ease**
  - No need to modify the source code
- **IDE debug configurations setting:**



- **When general exception is raised, it will pop up an error**



Script-Based RTOS Awareness

- Provide RTOS information to help debugging
- Display contents controlled by a Python script

Task List

task name	number	priority	start of stack	top of stack	status
"IDLE"	3	0	0x208438 <uxIdleTaskStack.2447>	0x208af0 <...	Running
"Task 2"	2	2	0x200cf8 <ucHeap+2272>	0x2013d0 ...	Delayed
"Task 1"	1			0x200b08 ...	Delayed
"Tmr Svc					

Click to show register list

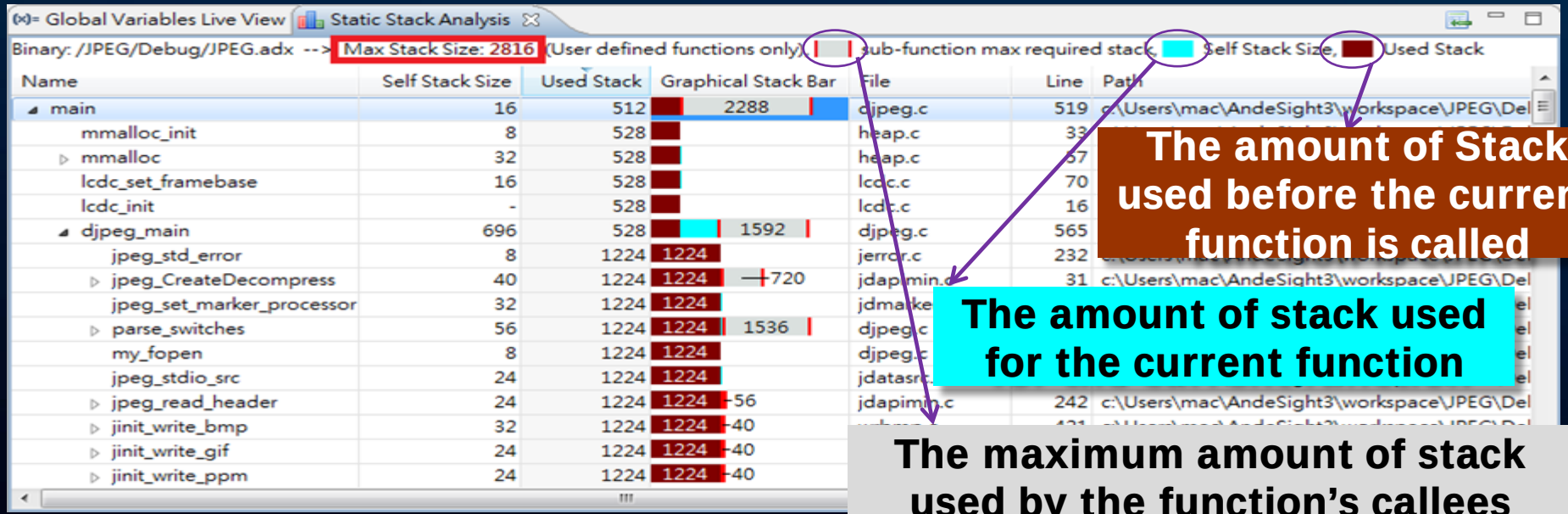
Registers			
$x1	$x2	$x3	$x4
0x00003d4a	0x002073b0	0x00200900	0x00000000
$x5	$x6	$x7	$x8
0x00000000	0x00000001	0x7ea0c49a	0x0000000a
$x9	$x10	$x11	$x12
0x0020bd08	0x00000000	0x40520000	0x00000000

Event List

queue name	handler address	max length	item size	messages waiting
"TmrQ"	0x200378	5	32	0
task name	number			
"Tmr Svc"	4			

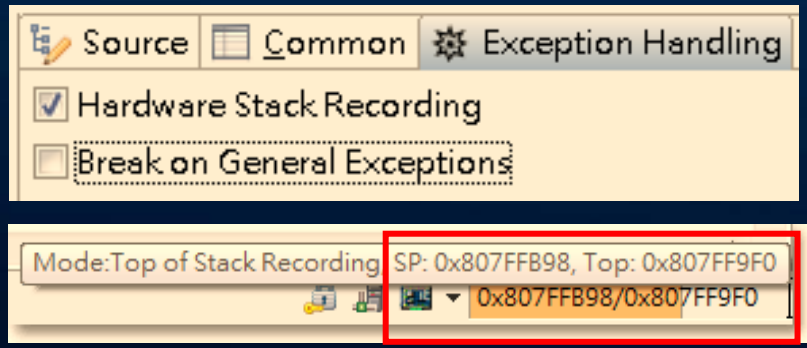
Static Stack Analysis View

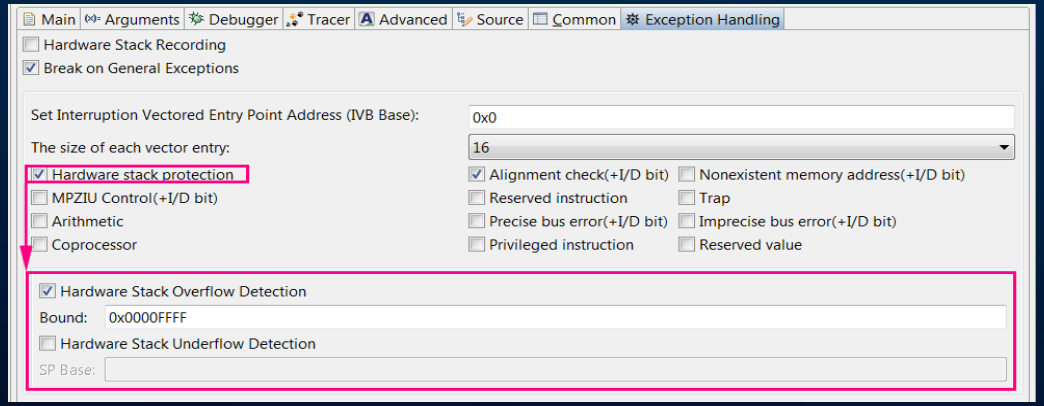
- Sizes estimated statically after project is built
- Report the maximum stack size of the whole program
- Display information of stack usage for each function



StackSafe™ Protection Handling

- **Record mode**
  - Track the maximum usage of stack pointer
- **Protection mode**
  - Raise an exception if over the allowed limit

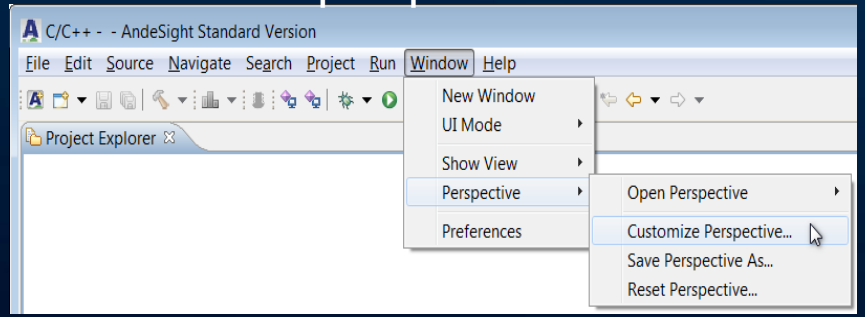




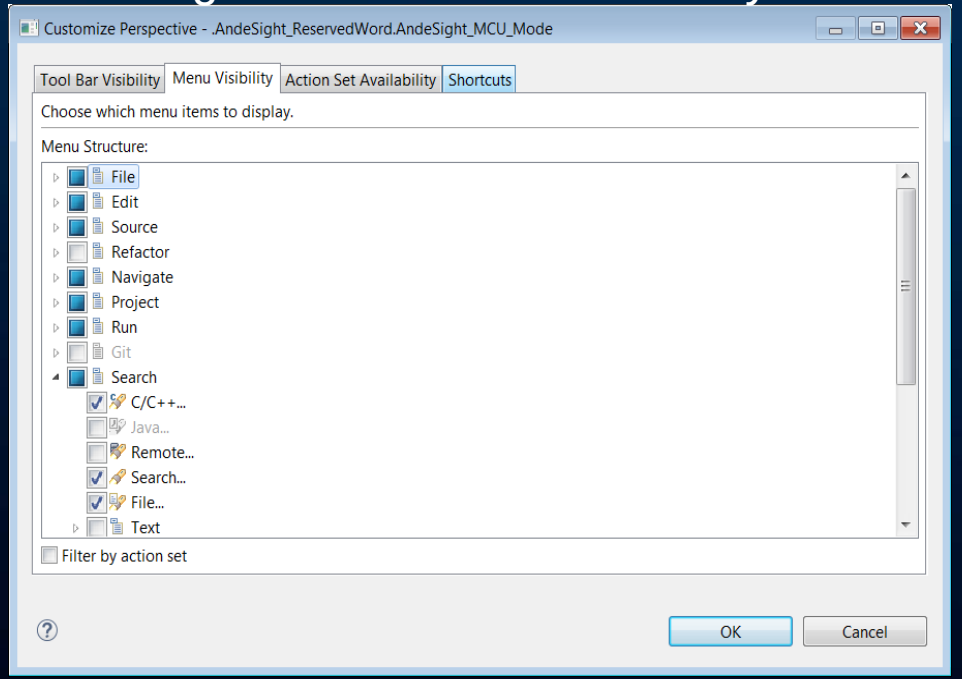
Custom UI

- **Customize the layout of the Menu and Toolbar items to meet your needs**

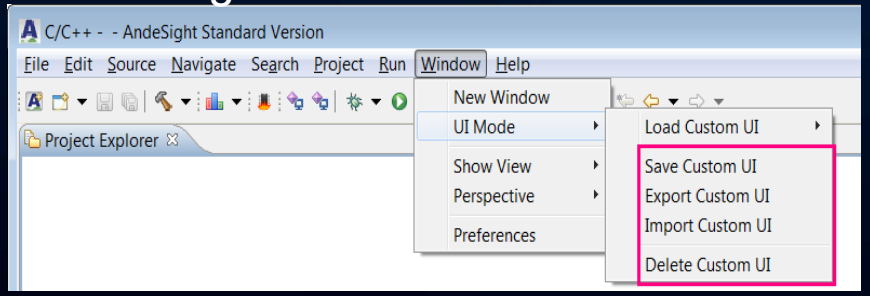
- Customize perspective



- Change menu & toolbar visibility



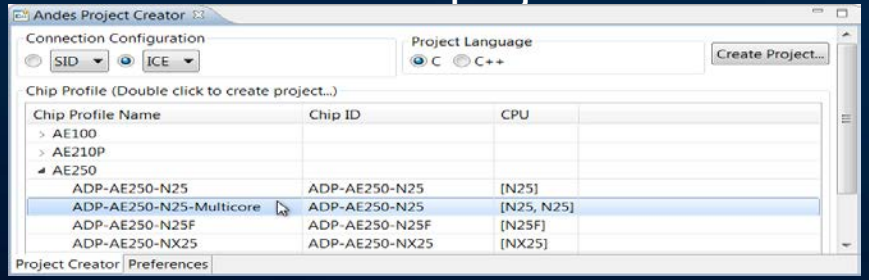
- Manage custom UI



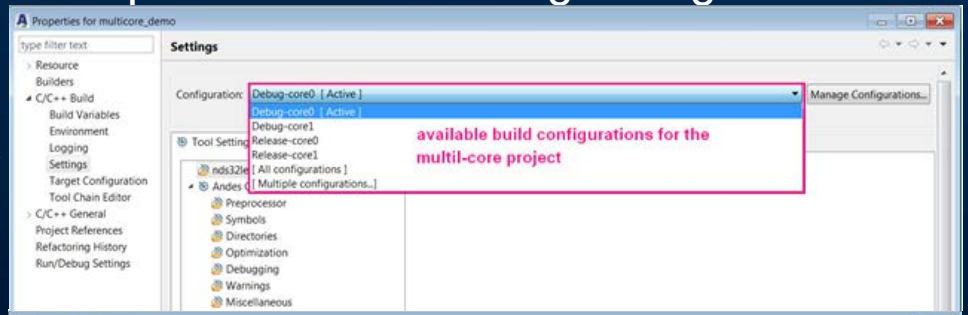
Multicore Development Support

- **Develop the multicore software by simply creating multicore projects with separate build and debug configurations**

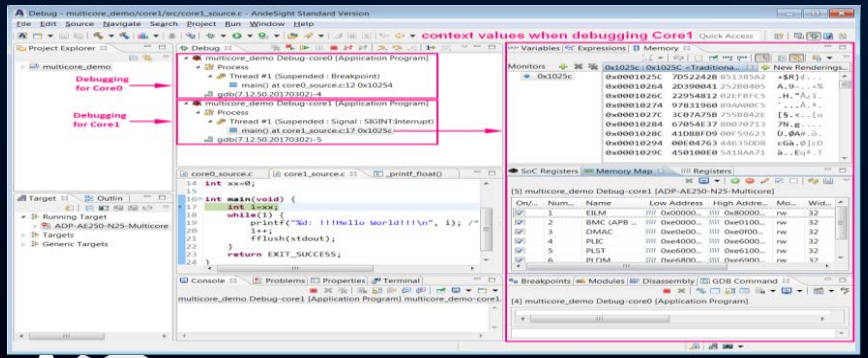
- Create multicore project

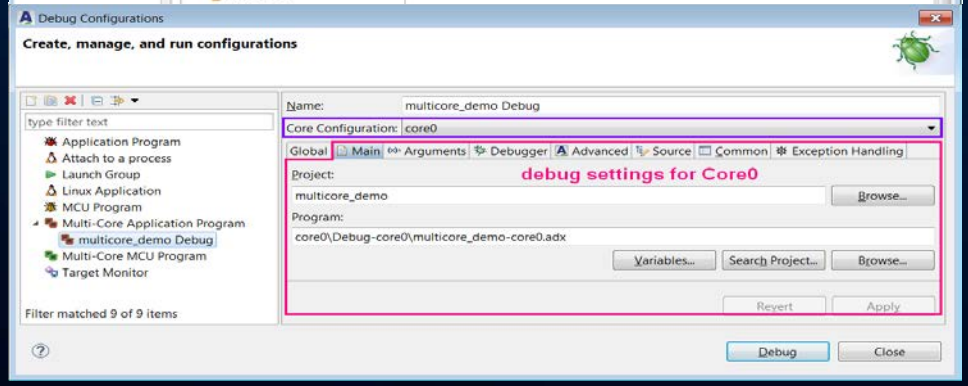


- Separate build & debug configurations

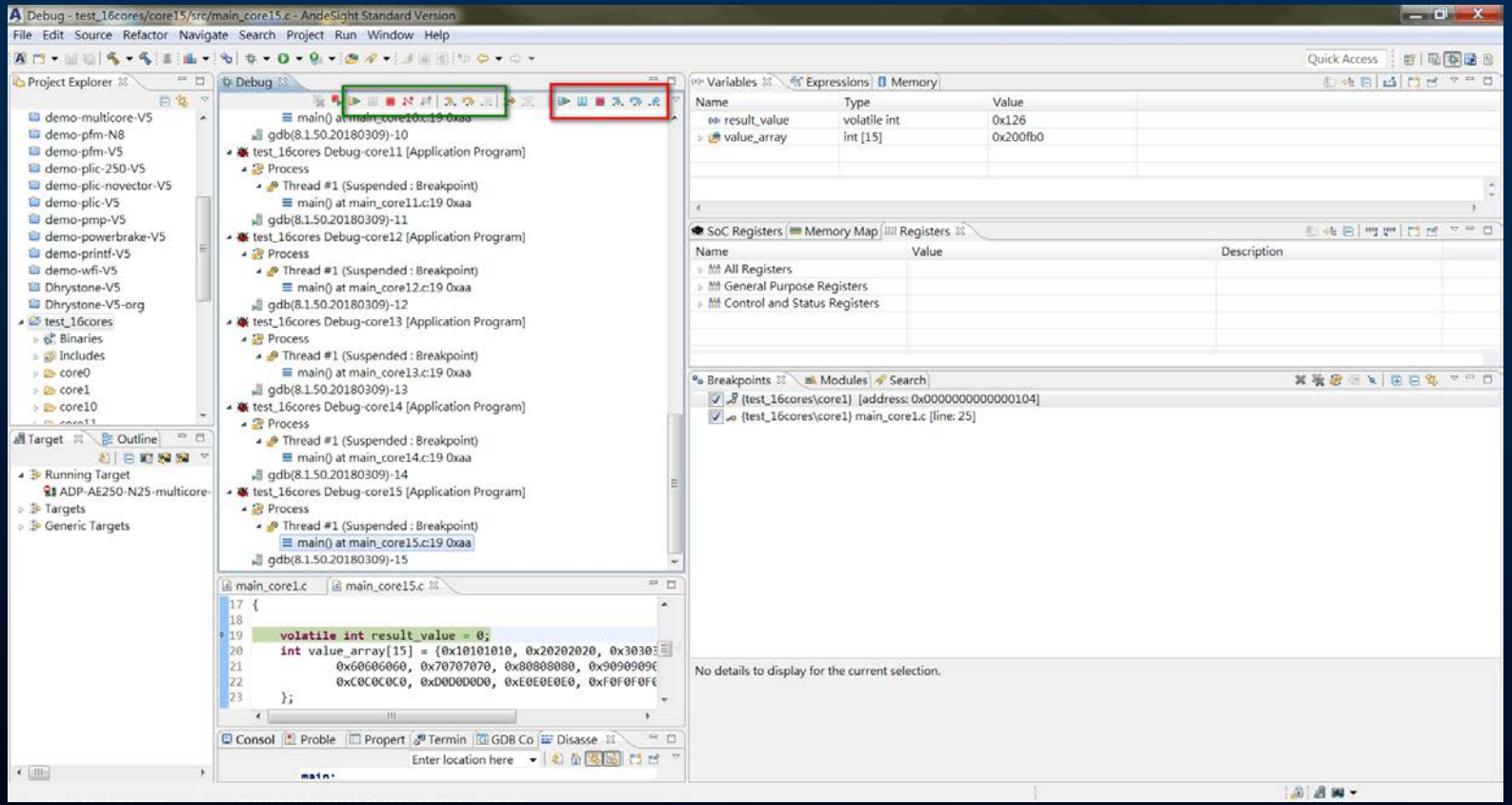


- Build & debug project





Multicore Debug



Cache Dump View

- Easily monitor CPU caches when program suspends

Cache Dump View

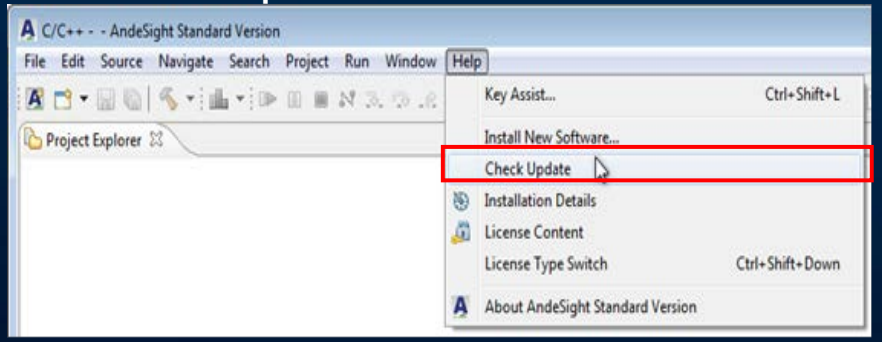
L1 Instruction Cache

Address	Set	Way	V	D	L	00	04	08	0C	10
000000039a000080	0004	0001	0	0	0	cbb807a2	7d002773	03900793	7e5b07ea	470d0017
000000039a0000a0	0005	0000	0	0	0	60050513	288000ef	47058c10	8c00b7e5	00001517
000000039a0000a0	0005	0001	0	0	0	60050513	288000ef	47058c10	8c00b7e5	00001517
000000039a0000c0	0006	0000	0	0	0	c78397c2	00230007	00a300fe	012300fe	01a300fe
000000039a0000c0	0006	0001	0	0	0	c78397c2	00230007	00a300fe	012300fe	01a300fe
000000039a0000e0	0007	0000	0	0	0	03a300fe	031300fe	bfbd0203	64966436	79d27972
000000039a0000e0	0007	0001	0	0	0	03a300fe	031300fe	bfbd0203	64966436	79d27972
000000039a000100	0008	0000	0	0	0	ebbe2355	5ffb564b	fb1ff76f	bebbe3fd	e7ff337b

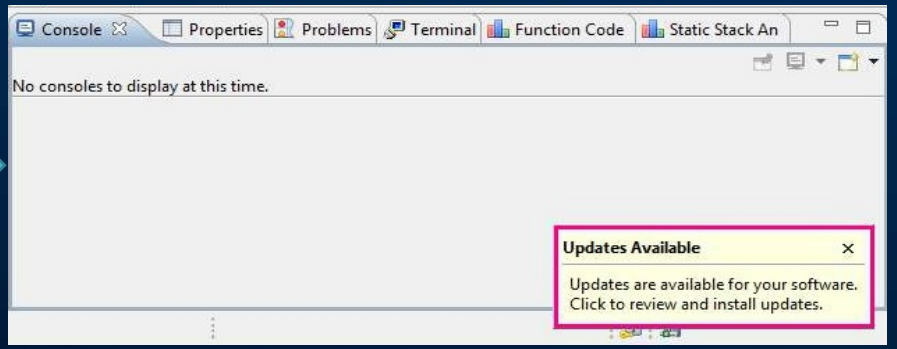
AndeSight Updater

■ Friendly and easy-to-use update interface

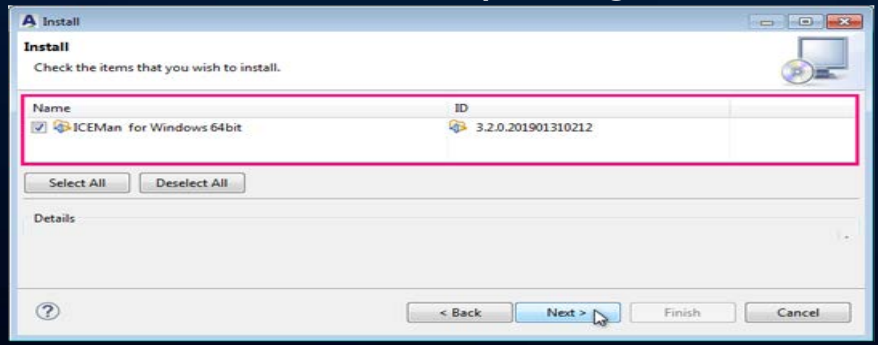
- Check update



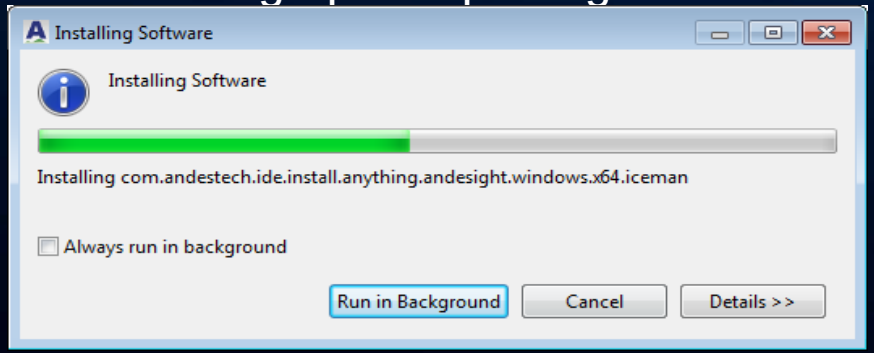
- Notification wizard



- Select the desired packages



- Installing update packages



AndeSoft™ BSP

Bare Metal, RTOS, Linux and DSP ISA

AndeSoft™: Application Building Blocks

Fundamental

- Compiler and toolchain are contributed to and supported officially by **GNU and LLVM** communities
- Optimized **MCUlib**, newlib, glibc and **DSPlib**
- Concise linker script and its tools, **Linker Scattering-and-Gathering (LdSaG)**
- **Sample programs** to demo AndesCore™ features

Real-Time Operating Systems

aws qualified device Amazon FreeRTOS

- **Open source:** Zephyr, FreeRTOS
- **Commercial:** ThreadX, LiteOS, RT-Thread, SylixOS
- **RISC-V ready:** VxWorks, µC/OS-[II/III], MyNewt, embOS, RTEMS, NuttX, seL4, uC3/Compact, AliOS Things

Linux, Middleware and SW Framework

- **Linux kernel** since 4.17, device drivers and advanced features: **strace**, **ftrace**, **Perf**, **SMU**, **power throttling**, **suspend to RAM** and **kernel module**
- **U-Boot** and **BBL**
- **Andes6**: connect LPWAN to IPv6 seamlessly

AndeSoft™: Bare Metal

- Rich **startup demo** projects for Andes-specific features

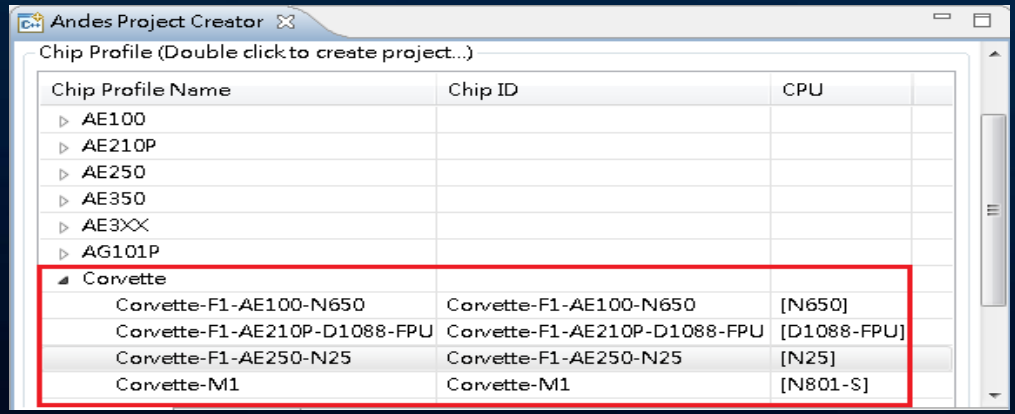
Categories	Startup demo
Interrupt	PLIC, CLIC
Memory	MMU, PMP, cache, cache lock, ECC, bus matrix slave port
Power Management	PowerBrake, hibernate, WFI CPU standby/resume
Programming	DSP, printf UART redirect, C++ programming
Misc	StackSafe™, performance monitor, SMP

- **AMSI (Andes MCU Software Interface) driver APIs**
  - UART, GPIO, RTC, PWM, SPI, I2C, WDT and DMA

Arduino Development

- Support Arduino language reference APIs
- Support Arduino standard libraries
- AndeSight™ IDE Arduino software development plugin

AndeShape™Corvette-F1



AndeSoft™: RTOS

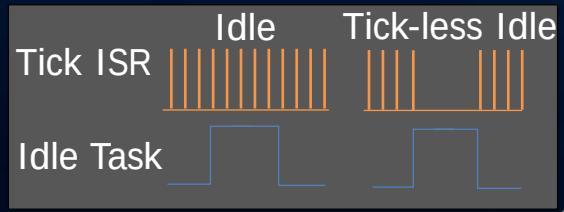
AliOS Things · embOS · FreeRTOS · Apache mynewt · RTEMS · RT-Thread · seL4 · SylixOS embedded · ThreadX · µC/OS RTOS and Stacks · µC3 micro c cube Compact · VxWorks · Zephyr™

- A market-leading real-time operating system (RTOS) for MCU
- Scalable size, with memory footprint as low as 9KB
- Tick-less idle
  - Power-saving by stopping periodic tick interrupt in the idle mode, supported by standard RISC-V architecture
- AWS FreeRTOS Qualified
- FreeRTOS test suite verified
- RTOS-awareness debugging

FreeRTOS

aws qualified device Amazon FreeRTOS

ANDES TECHNOLOGY · IAR SYSTEMS · LAUTERBACH DEVELOPMENT TOOLS · SEGGER



AndeSoft™: RTOS

- An OS that runs best on MCUs for wearable and IoT devices
- Very small memory footprint (will run in 8k)
- Highly configurable, highly modular
- Apache 2.0 license, hosted by Linux Foundation

Zephyr™

- Pre-certified by TUV and UL to many safety standards
  - IEC-61508 SIL 4, IEC-62304 SW Safety Class C, ISO 26262 ASIL D and EN 50128
- EAL4+ Common Criteria security certification
- Small footprint, as 2KB instruction area and 1KB of RAM

- Support 32-bit/64-bit version, multicore (AMP & SMP) and MMU
- Large-scale middleware such as Qt, CODESYS, Python, Java Script and POSIX API
- Functional safety certified for rail transport, medical, industrial automation, automotive, electric power and aerospace

SylixOS® embedded

AndeSoft™: RTOS

- A Lightweight IoT Operating System that Makes Everything Around Us Smart
- Ultra-small kernel, basic kernel size of less than 10 KB
- Low power consumption
- One-stop software platform, lowering development requirements and improving development efficiency
- Open Source under a BSD 3-Clause license

RT-Thread

- An open source IoT operating system under the Apache 2.0 license
- Abundant software components such as GUI, TCP/IP stack, file system and standard API support like POSIX, CMSIS, C++ runtime, Javascript
- Device and cloud integration design, easy to connect various IoT devices with cloud

AndeSoft™: Linux

- **Linux Kernel**
  - LTS RV32/RV64 port, since 4.17
  - SMP support
  - Cache coherence and cache non-coherence support
  - Linux Test Projects (LTP) verified
  - Device drivers for AE350 platform
- **U-Boot**
  - RV32/RV64 port, maintainer and contributor
  - SMP support
  - Supervisor mode support
  - Device drivers for AE350 platform
- **BBL**

AndeSoft™: Linux

- **Linux Distribution and Build System***

*: available upon request

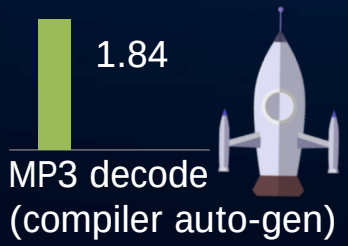
- **Linux Kernel Tools**
  - **strace / ftrace** for developers to debug
  - **Perf** to evaluate the bottleneck of the whole system
  - **System Management Unit (SMU)**
    - Suspend-to-RAM: suspended by sysfs and wakeup by RTC and UART interrupt
    - PowerBrake: power throttling mechanism controlled by sysfs
  - **Kernel module** support all relocation types for RV32 and RV64
- **Andes GitLab Service for Linux Development Packages**

Qemu

- **Support AndeStar™ RISC-V V5 32/64 bits CPU**
- **Support AE350 SoC platform**
- **Integrated with AndeSight™ IDE**
- **Early software development and verification**
  - Run U-Boot and Linux with LTP
  - Run FreeRTOS with software applications
  - Used by HPE and openSUSE project for UEFI EDKII
  - Used by Red Hat for Fedora RISC-V port regression farm

DSP Support

- **DSP ISA assembly programming**
  - Derived and evolved from real use cases over decades
    - Support 32 bits and 64 bits
    - Support saturation and rounding
    - Cover SIMD, partial SIMD, bit manipulation and etc.
- **DSP intrinsic functions**
  - As C-like functions without bothering to program in assembly
- **DSP library**
  - >200 functions in 8 categories (basic, complex, controller, filtering, matrix, statistics, transform and utils)
- **Some source patterns are recognized by compiler, then DSP instructions are auto-generated to facilitate development**
- **Compatible with CMSIS-DSP library API**
  - By including an API wrapper header file



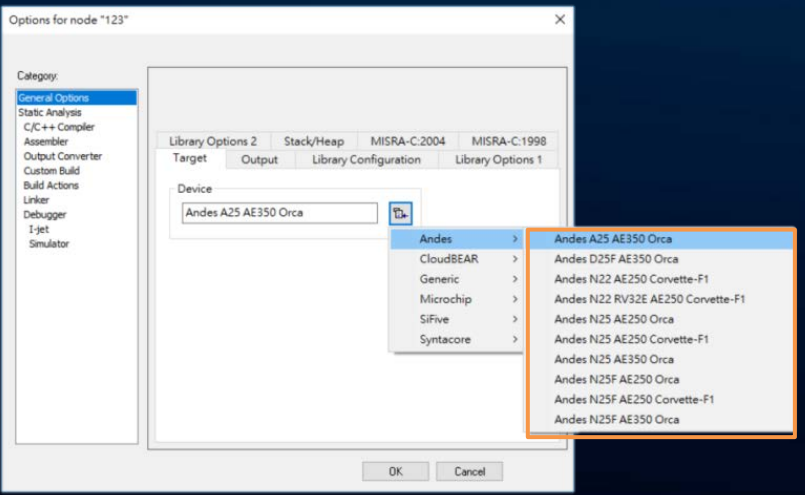
IAR Embedded Workbench for RISC-V

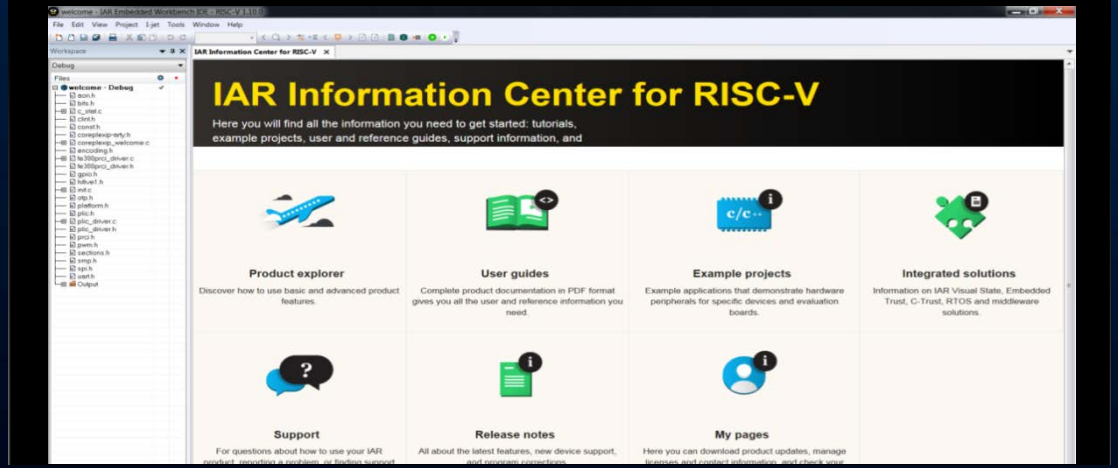
- Complete build and debug toolchain for RISC-V
- Support all series of Andes RISC-V CPU

IDE
- Embedded Workbench

Debugger
- I-Jet ICE
- C-SPY tool

Toolchain
- Compiler
- C-STAT
- Functional Safety





Summary

- **AndeSight™ IDE: Rich features to speed up SW development**
  - Ease of use to reduce development time
- **AndeSoft™ BSP: Well-integrated building blocks to help users to build SoC software quickly and easily**
  - Highly-optimized toolchains for better performance and smaller memory footprint
- **Andes Comprehensive RISC-V SW solutions to achieve fast time-to-market and high quality**
  - Supporting 5 Bn+ SoC

Thank you!
See you next Webinar