

Access CPU with AndeSight



Create Empty Project



1. Click to create project

2. Select ICE

3. Select your chip profile

4. Create project

Connection Type: Simulator ☐ ICE ☒

Project Language: C ☒ C++ ☐

Chip Profile (Double click to create project...)

Chip Profile Name	Chip ID	CPU
AE100		
AE210P		
ADP-AE210P-D1088	ADP-AE210P-D1088	[D1088]
ADP-AE210P-E830	ADP-AE210P-E830	[E830]
ADP-AE210P-N1068A-S	ADP-AE210P-N1068A-S	[N1068A-S]
ADP-AE210P-N705-S	ADP-AE210P-N705-S	[N705-S]
ADP-AE210P-N801-S	ADP-AE210P-N801-S	[N801-S]
ADP-AE210P-N820	ADP-AE210P-N820	[N820]
ADP-AE210P-N968A-S	ADP-AE210P-N968A-S	[N968A-S]
AE3XX		
ADP-AE3XX-D15F	ADP-AE3XX-D15F	[D15F]
ADP-AE3XX-N15F	ADP-AE3XX-N15F	[N15F]
AG101P		
ADP-AG101P-16MB-E830-32I	ADP-AG101P-16MB-E830-32I	[E830]
ADP-AG101P-16MB-N705-S-32I	ADP-AG101P-16MB-N705-S-32I	[N705-S]
ADP-AG101P-16MB-N801-S-32I	ADP-AG101P-16MB-N801-S-32I	[N801-S]
ADP-AG101P-16MB-N820-32I	ADP-AG101P-16MB-N820-32I	[N820]
ADP-AG101P-4GB-D1088-SPU-32I	ADP-AG101P-4GB-D1088-SPU-32I	[D1088-SPU]
ADP-AG101P-4GB-N1068A-S-32I	ADP-AG101P-4GB-N1068A-S-32I	[N1068A-S]
ADP-AG101P-4GB-N1068A-SPU-32I	ADP-AG101P-4GB-N1068A-SPU-32I	[N1068A-SPU]

Enter Project Name



C Project

Create C project of selected type
Chip Profile: ADP-AE210P-N820

Project name:

☒ Use default location

Location:

Choose file system:

Project type:

- Andes Executable
 - ☒ Empty Project
 - ☐ Hello World ANSI C Project
- Andes Static Library
- Makefile project

Toolchains:

- nds32le-elf-mculib-v3m
- nds32le-elf-newlib-v3m

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

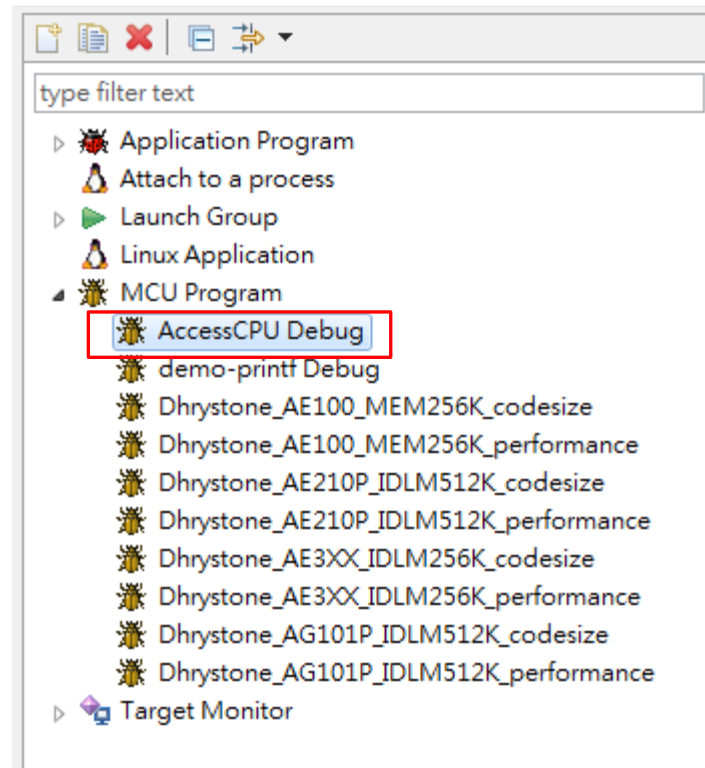
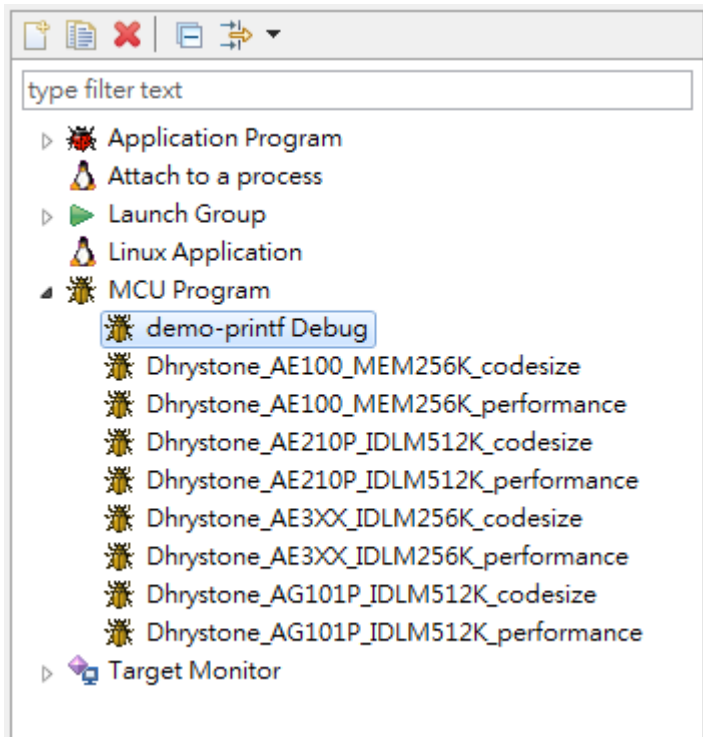
1. Enter project name

2. Click Finish

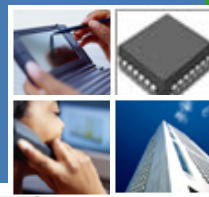
MCU Program Settings



- ❖ Right click project > Debug As > Debug Configurations > Double click "MCU Program" if your project is not in the list



MCU Program Settings



The screenshot shows the 'MCU Program Settings' dialog box. On the left, a tree view lists various debug configurations, with 'AccessCPU Debug' selected and marked with a red box and the number 1. The main panel on the right has tabs for 'Main', 'Debugger', 'Startup' (marked with a red box and the number 2), 'Tracer', 'Source', and 'Common'. Under the 'Startup' tab, the '1. GDB Initialization Commands' section contains a list box with 'Reset and Hold' checked, marked with a red box and the number 3. Below this are sections for '2. Binary File Options' (with 'Load binary file' unchecked and file selection buttons), '3. Runtime Options' (with 'Set program counter at (hex):', 'Set breakpoint at:', and 'Resume' all unchecked), and '4. GDB Run Commands'. At the bottom right, the 'Debug' button is highlighted with a red box and the number 4. Other buttons like 'Revert', 'Apply', and 'Close' are also visible.

CPU Registers



Target Running

CPU registers

The screenshot shows the Andes Studio IDE interface. The 'Project Explorer' on the left shows a project named 'AccessCPU' with a target 'ADP-AE210P-N820 ICE:'. The 'Debug' console in the center shows the target is running. The 'Registers' window on the right displays a list of CPU registers and their values.

Name	Value
All Registers	
General Purpose Registers	
Configuration System Registers	
Interrupt System Registers	
MMU System Registers	
mr0 (MMU_CTL)	{0x0, NTC0 = 0x0, NTC1 = 0x0}
mr6 (ILMB)	{IEN = 0x1, ILSMZ = 0x8, (raw
IBPA	0x0
ILSMZ	0x8 - 1024 kB
IEN	0x1 - Enabled
mr7 (DLMB)	{DEN = 0x0, DLMSZ = 0x7, Df
EDM System Registers	
Performance Monitoring	
Implementation-Dependent Regis	

Monit