

# Access CPU with AndeSight





- ❖ Create a project
- ❖ Access CPU with "Reset and Hold"
- ❖ Access CPU without "Reset and Hold"
- ❖ Access CPU with "Target Monitor"

# Create Empty Project



1. Click to create project

2. Select ICE

3. Select your chip profile

4. Create project

Andes Project Creator

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]

Project Creator Preferences

# 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

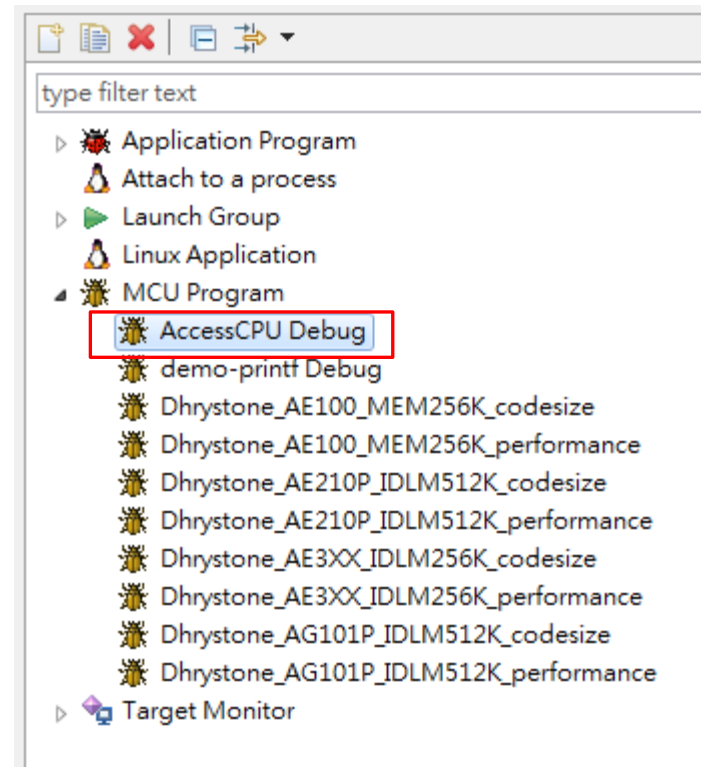
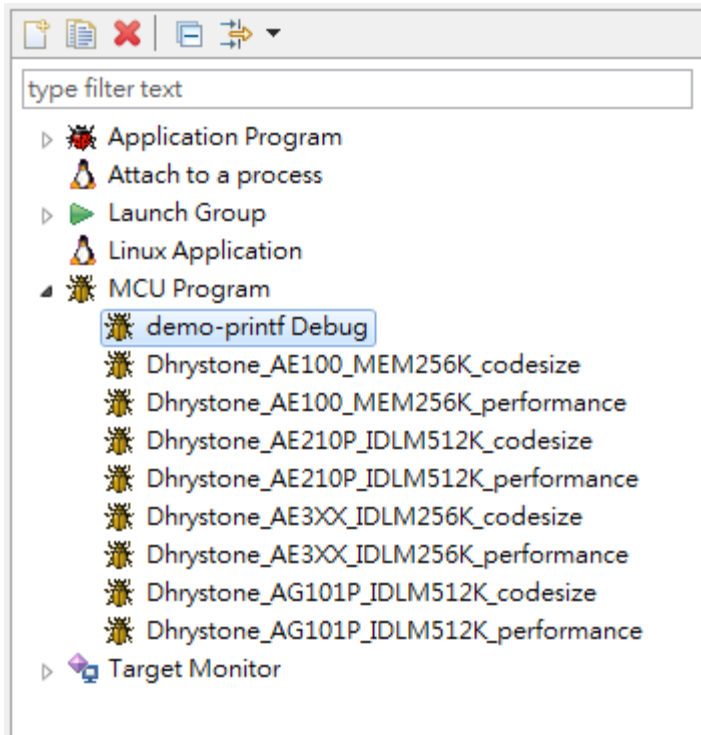
**1. Enter project name**

**2. Click Finish**

# MCU Program Settings



- ❖ Right click project > Debug As > Debug Configurations > Double click "MCU Program" if AccessCPU 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. 'AccessCPU Debug' is selected and highlighted with a red box and the number 1. The main panel on the right has tabs for 'Main', 'Debugger', 'Startup' (highlighted with a red box and the number 2), 'Tracer', 'Source', and 'Common'. Under the 'Startup' tab, there are four sections: '1. GDB Initialization Commands', '2. Binary File Options', '3. Runtime Options', and '4. GDB Run Commands'. In the '1. GDB Initialization Commands' section, the 'Reset and Hold' checkbox is checked and highlighted with a red box and the number 3. 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



The screenshot shows the Andes Studio IDE interface. On the left, the 'Project Explorer' shows a project named 'AccessCPU' with a target 'ADP-AE210P-N820 ICE:'. A red callout points to the 'Running Target' status. In the center, the 'Debug' pane shows the 'AccessCPU Debug [MCU Program]' with a 'Thread #1 (Suspended : User Request)' and a 'gdb(7.7.0.20140207)-78' process. On the right, the 'Registers' tab is selected, displaying a list of CPU registers and their values. A red callout points to this tab.

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	

Target  
Running

CPU  
registers



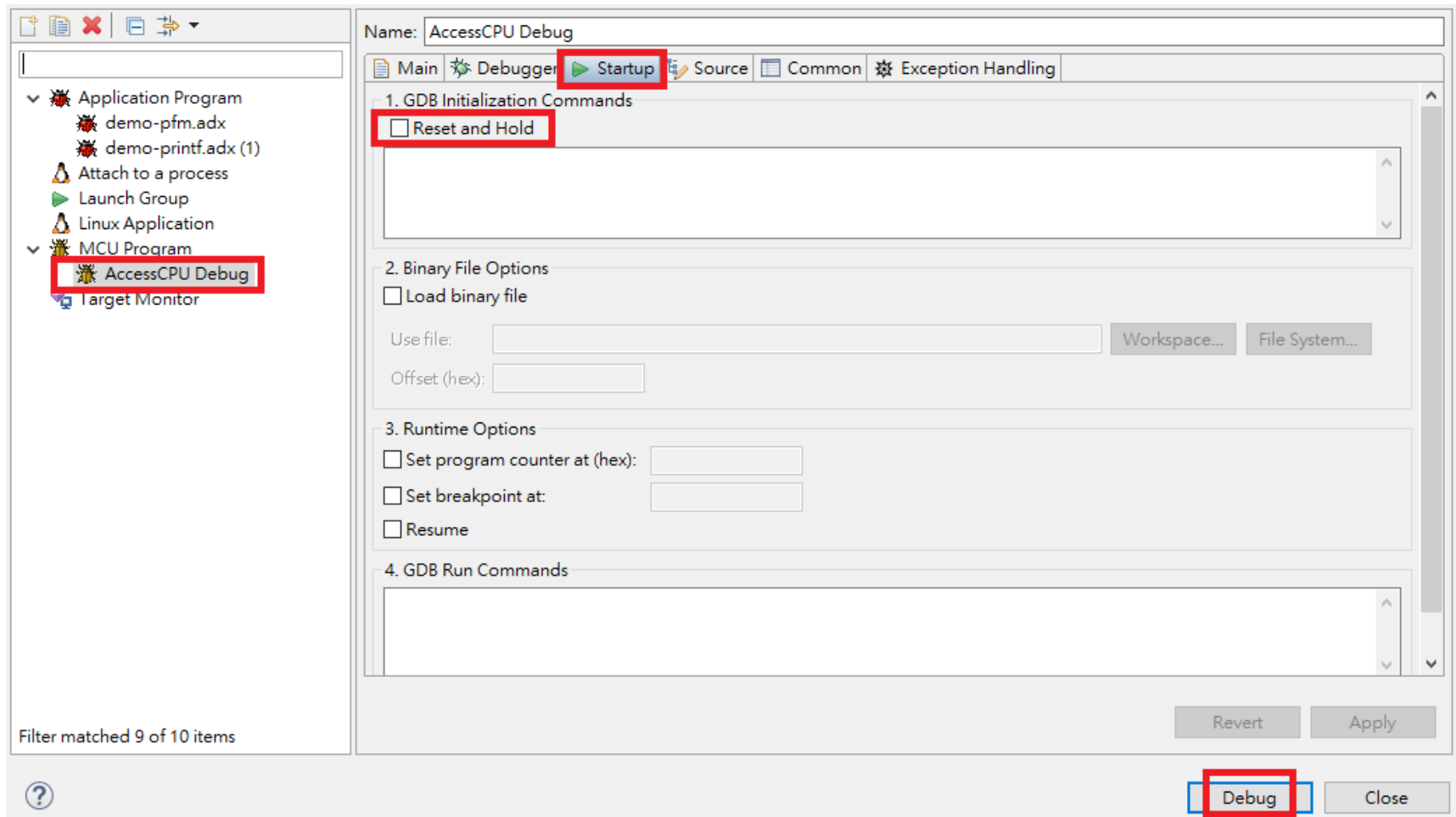
Monit   



# MCU Program without Reset and Hold



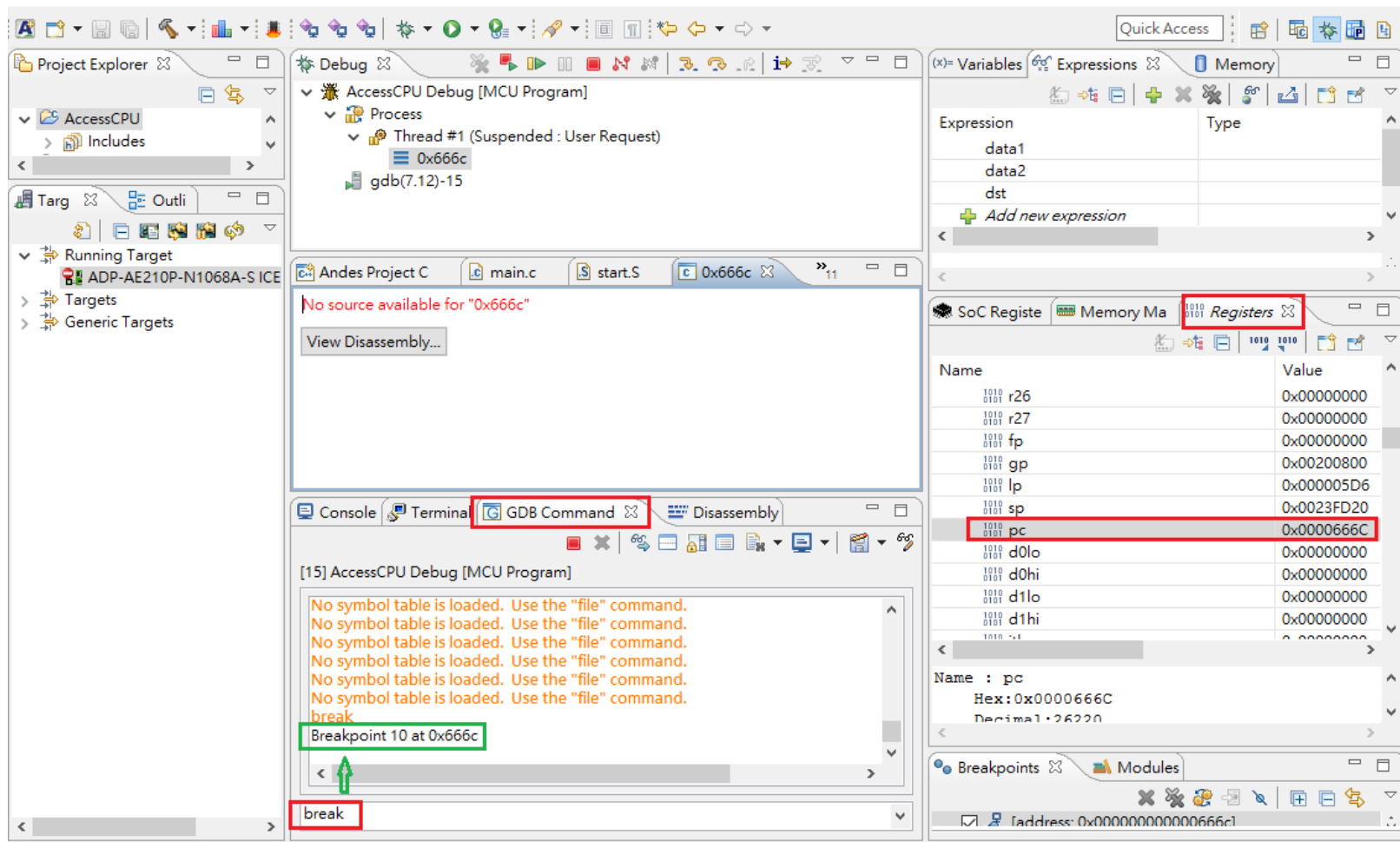
- ❖ Without “Reset and Hold” to reset the PC to 0



# MCU Program without Reset and Hold



- ❖ At "GDB command" view, type break to stop CPU
- ❖ The PC register at Registers view shows where the CPU stops



The screenshot displays the Andes Studio IDE interface during a debug session. The **Project Explorer** on the left shows the project structure. The **Debug** window in the center shows the **AccessCPU Debug [MCU Program]** process, with the **Thread #1 (Suspended : User Request)** at address **0x666c**. The **Console** window at the bottom shows the GDB command view, where the command **break** has been entered, resulting in **Breakpoint 10 at 0x666c**. The **Registers** window on the right shows the **pc** register (Program Counter) at address **0x0000666c**, which is highlighted with a red box. The **SoC Register** window also shows the **pc** register with its hexadecimal and decimal values.

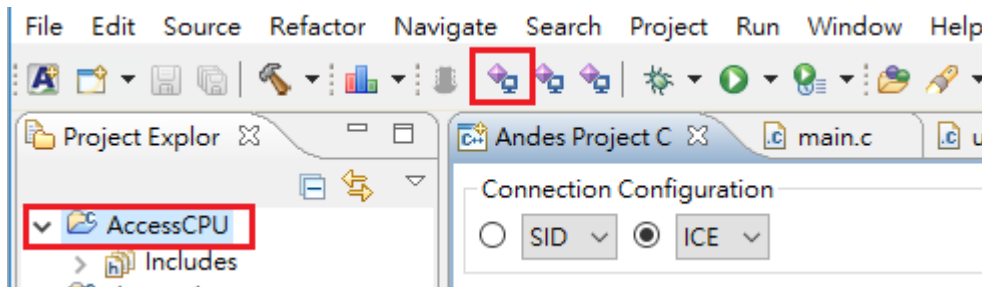
Name	Value
r26	0x00000000
r27	0x00000000
fp	0x00000000
gp	0x00200800
lp	0x000005D6
sp	0x0023FD20
pc	0x0000666C
d0lo	0x00000000
d0hi	0x00000000
d1lo	0x00000000
d1hi	0x00000000

Hex: 0x0000666C  
Decimal: 26220

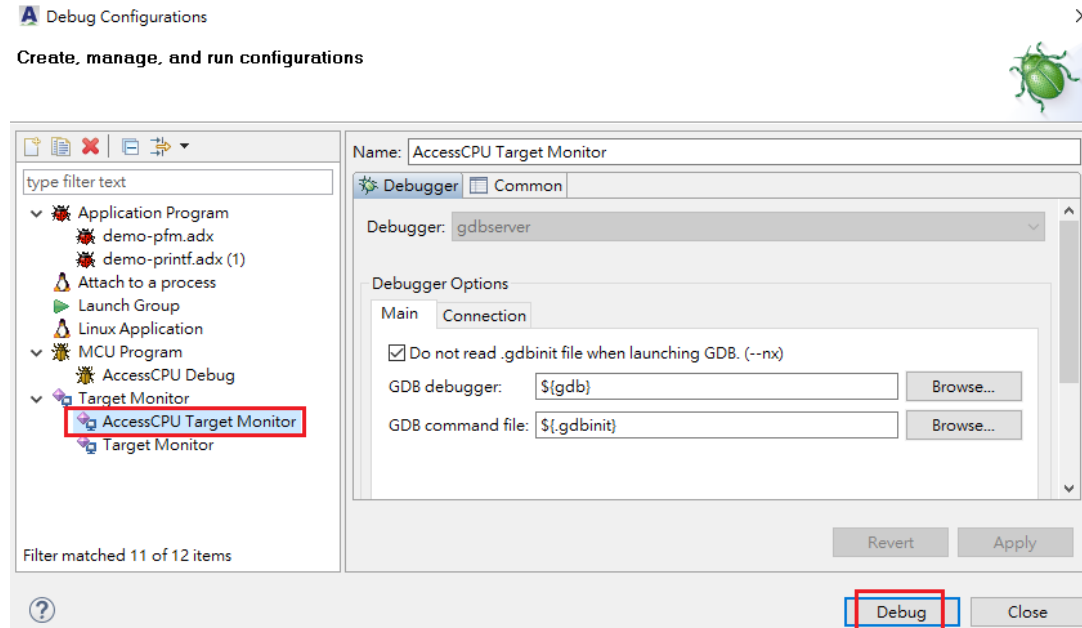
# Target Monitor



❖ Use “Target Monitor” icon



❖ Use “Debug as” > Debug Configurations > Target Monitor





File Edit Navigate Search Project Run Window Help

Project Explorer

- AccessCPU
  - Includes

Target

- Running Target
  - ADP-AE210P-N1068A-S ICE
    - Targets
    - Generic Targets

Debug

- AccessCPU Target Monitor (Target Monitor)
  - Process
    - Thread #1 (Suspended : User Request)
      - 0x666a
      - gdb(7.12)-22

Andes Project C | main.c | start.S | 0x666a

No source available for "0x666a"

View Disassembly...

Console

Enter location here

```
0000666a: bmski33 $r1, #0x0
0000666c: beqz38 $r1, 0x6666
0000666e: lwi $r0, [$r0+#0x20]
00006672: seb33 $r0, $r0
00006674: ret5 $lp
00006676: lwi $r2, [$r0+#0x34]
0000667a: bmski33 $r2, #0x5
0000667c: beqz38 $r2, 0x6676
0000667e: swi $r1, [$r0+#0x20]
00006682: ret5 $lp
00006684: push25 $r6, #0 ! {$r6, $fp, $gp, $lp}
00006686: mov55 $r6, $r0
```

Variables

Expression	Type	Value
data1		
data2		
dst		
Add new expr		

SoC Register

Name	Value
r23	0x00000000
r24	0x00000000
r25	0x00000000
r26	0x00000000
r27	0x00000000
fp	0x00000000
gp	0x00200800
lp	0x000005D6
sp	0x0023FD20
pc	0x0000666A
d0lo	0x00000000

Breakpoints

Modules

Address: 0x000000000000666c