

NUC501 Quick Start Guide for ADS

V1.01.001

Publication Release Date: Apr. 2009

The information in this document is subject to change without notice.

The Nuvoton Technology Corp. shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

This documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from the Nuvoton Technology Corp.

Nuvoton Technology Corp. All rights reserved.

Table of Contents

- 1. Introduction5**
 - 1.1. About the Quick Start Guide..... 5
 - 1.2. About NCU501 5
 - 1.3. About ARM Developer Suite..... 5
- 2. Quick Start.....6**
 - 2.1. Installing the ADS..... 6
 - 2.2. Connecting to your target..... 6
 - 2.3. ARM Developer Suite Overview..... 6
 - 2.3.1. Build Process 7
 - 2.3.2. Debugger 8
 - 2.4. Step-by-Step..... 10
 - 2.4.1. Open an existed project 10
 - 2.4.2. Creating a new project..... 10
 - 2.4.3. Modify the C Code 11
 - 2.4.4. Build the C Code 11
 - 2.4.5. Open the Debugger - AxD..... 12
 - 2.4.6. Debug with AxD..... 13
 - 2.4.7. Conclusion..... 13
- 3. Revision History15**

1. Introduction

1.1. About the Quick Start Guide

This Quick Start Guide will instruct you on how to use the ADS (ARM Developer Suite) with the NUC501 development board. It gives you the information necessary to use ADS for your own projects and provides an overview of the most commonly used ADS features.

1.2. About NCU501

The NCU501 is an ARM7TDMI-based MCU, specifically designed to offer low-cost and high performance for various applications, like interactive toys, edutainment robots, and home appliances. It integrates the 32-bit RISC CPU with 32KB high-speed SRAM, crypto engine with OTP key, boot ROM, LDO regulator, ADC, DAC, I2C, SPI, USB2.0 FS Device, & GPIO into a cost-affordable while feature-rich micro-controller.

With so many practical peripherals integrated around the high-performance ARM7 CPU, the NCU501 is suitable for such applications as Interactive toys, edutainment robots, and home appliances. Whenever MIPS-hungry task meets cost-effective demand, you'll find the NCU501 truly useful to satisfy the requirement.

1.3. About ARM Developer Suite

ARM[®] Developer Suite[™] (ADS) is ARM's software development tools. It includes a suite of applications, together with supporting documentation and examples for the ARM family of RISC processors.

You can use ADS to develop, build, and debug C, C++, or ARM assembly language programs.

ADS consists of the following major components:

- Command-line development tools
- GUI development tools
- Utilities
- Supporting software.

2. Quick Start

2.1. Installing the ADS

The ARM Developer Suite is now available as a standalone product, shipping as the ARM Developer Suite Archive CD by ARM Corp. This contains ADS 1.2.1, ADS 1.1 and ADS 1.0.1 (on Unix platforms, only the code generation tools are provided for versions 1.1 and 1.0.1).

Please contact with ARM for more information. <http://www.arm.com/>.

2.2. Connecting to your target

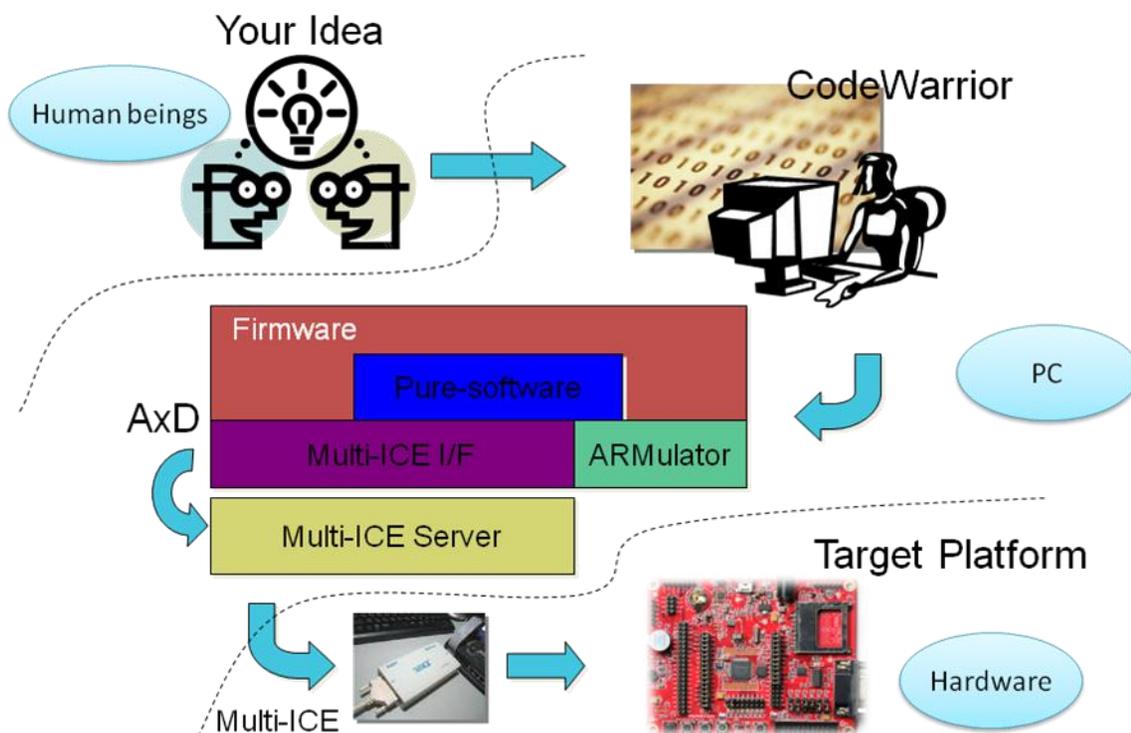
In order to do source level debug, a JTAG-based debug hardware, such as RealView Multi-ICE, is required. The cable of the ICE need to connect to the 20-pin ICE connector on NUC501 DEV Board as follows:

RealView Multi-ICE connects to the NUC501 using the 20-pin ribbon cable.



2.3. ARM Developer Suite Overview

The ADS includes text editor, compiler, assembler and debugger. The user could write the program and compile it by CodeWarrior IDE environment. Then the output code image could be downloaded to the target board through debug tool and debug the program in source level or assemble level. The following figure shows the code development flow of ADS and the relationship of each component.



The development flow of ADS.

The debug tool of ADS is called AxD. it supports a software simulator of ARM, which called ARMulator and supports to connect the target board through Multi-ICE.

2.3.1. Build Process

The CodeWarrior is the IDE of ARM. It includes compiler, assembler, and linker. The CodeWarrior provides four different project templates, including:

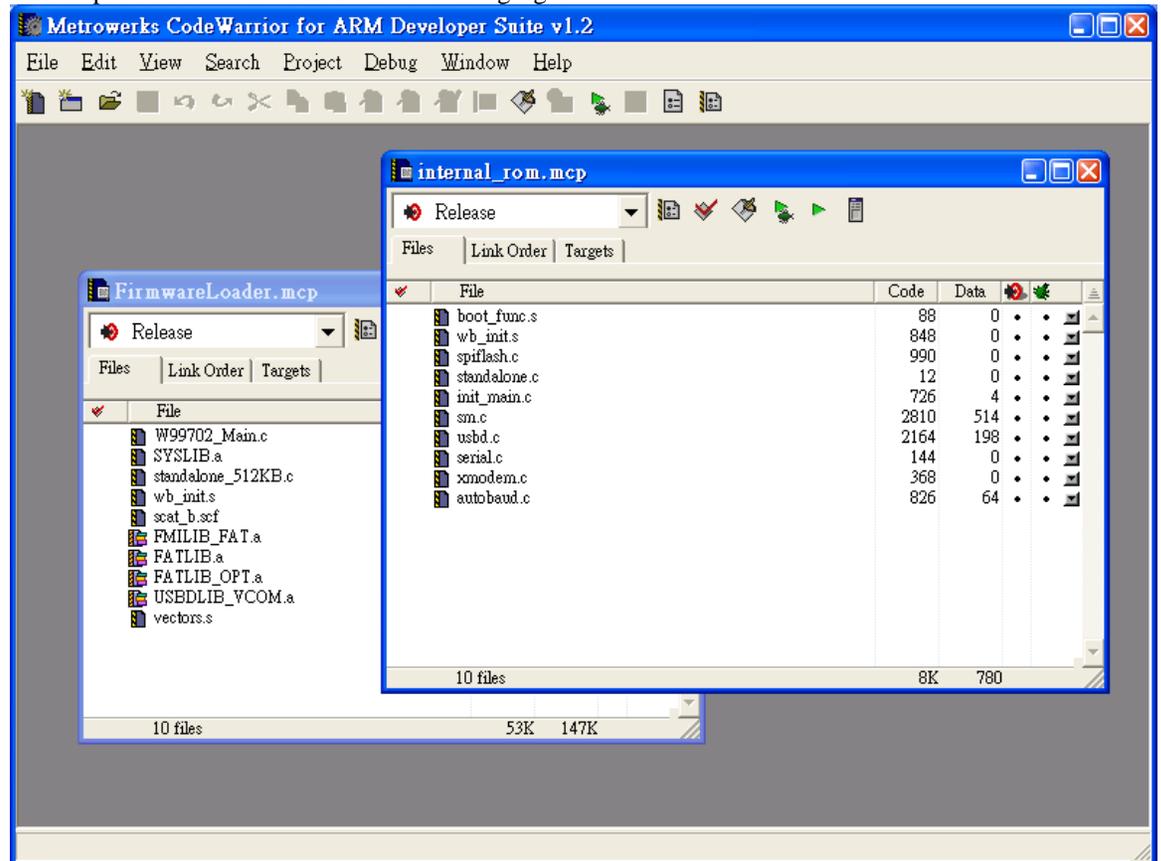
- ARM Executable Image
- ARM Object Library
- Thumb Executable Image
- Thumb Object Library
- Thumb/ARM Interworking Image

By default, the new ARM project will include three default targets, which are:

- Debug
- Release
- DebugRel

Each target is different in optimize level. The Debug target is fully debug at expense of optimization, the Release target is fully optimization at expense of debug, and the Debug target is debug/optimization trade-off option.

The snapshot CodeWarrior IDE is as following figure:

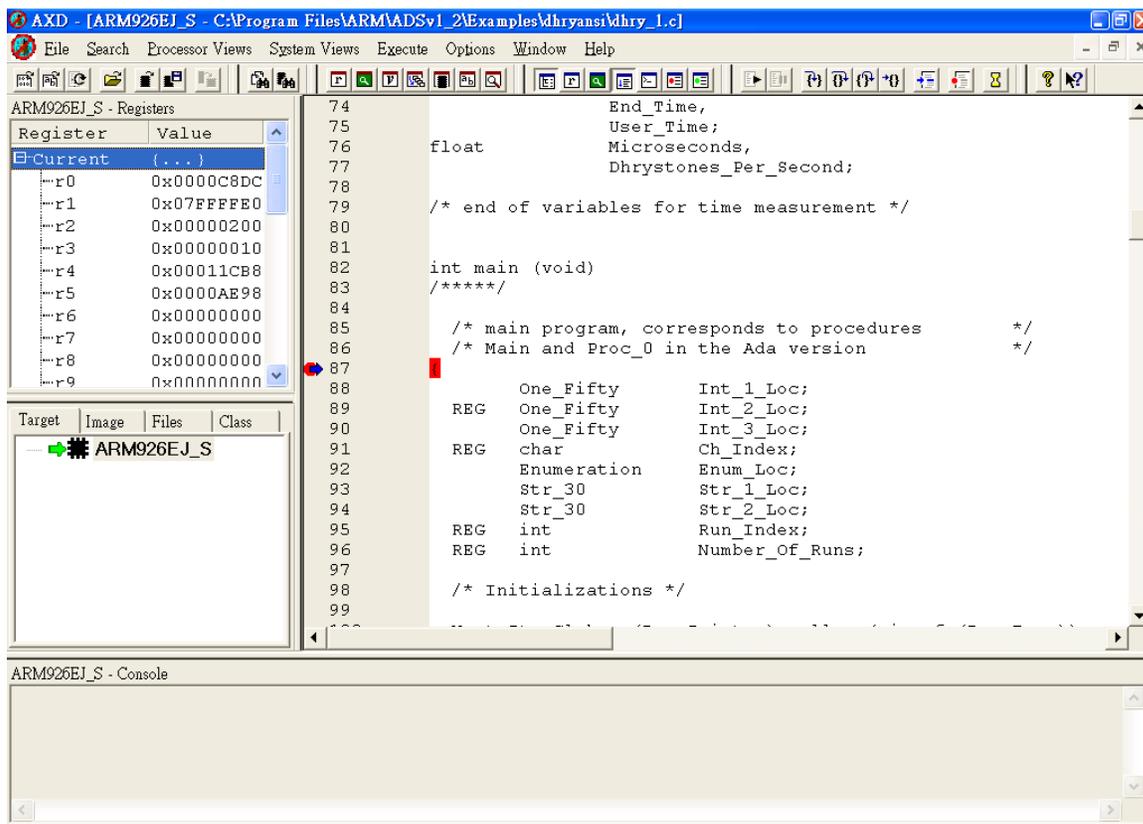


The file extend name of CodeWarrior project file is “.mcp”. After installed the ADS, it is able to open the project with CodeWarrior by double clicking the existed project file.

2.3.2. Debugger

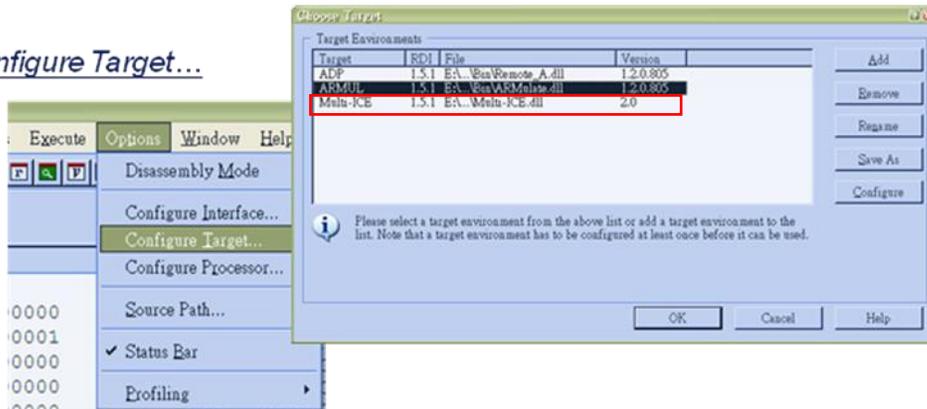
The debugger tool of ADS is called AxD. AxD includes a software simulator which called ARMulator. Furthermore, the AxD is able to link to the target board through Multi-ICE.

The snapshot of AxD is as following figure:



Because AxD is default to connect to ARMulator, it needs to set the target to Multi-ICE server manually.

Options → Configure Target...

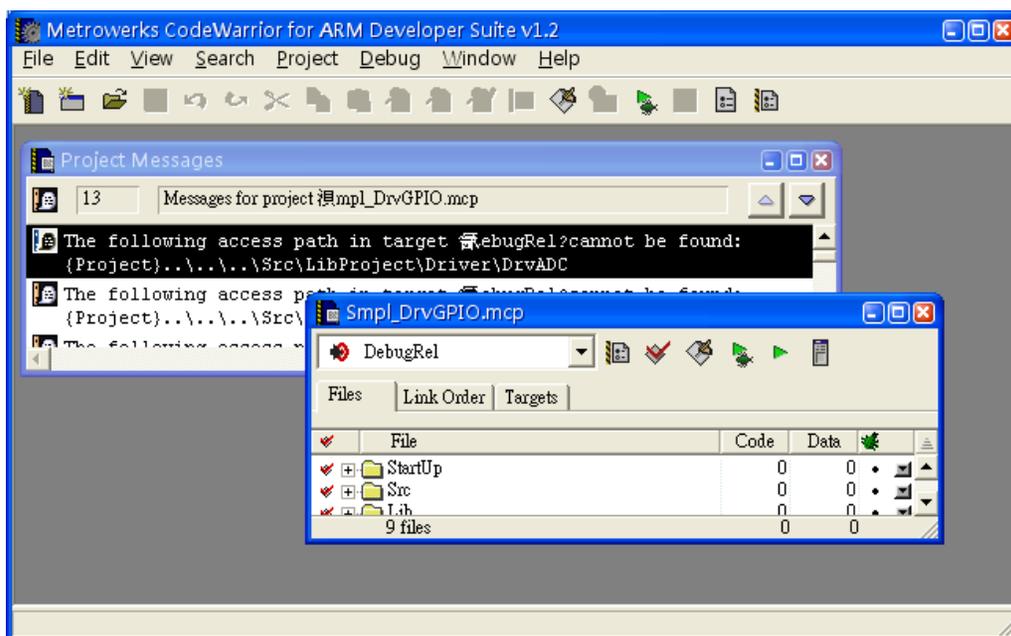


2.4. Step-by-Step

This section details all of the materials necessary to download code to an ARM-based development board for debug by ADS with using the JTAG debug agent.

2.4.1. Open an existed project

After the ADS installed, we can open the CodeWarrior by double click the existed .mcp file. For example, we can open the GPIO driver sample code by double click the project file in `C:\WIDE\ARMNUCSDSV1.00.003\NUC501\NuvotonPlatform_ADS\Sample\Driver\Smpl_DrvGPIO\Smpl_DrvGPIO.mcp` and the snapshot is as follows:



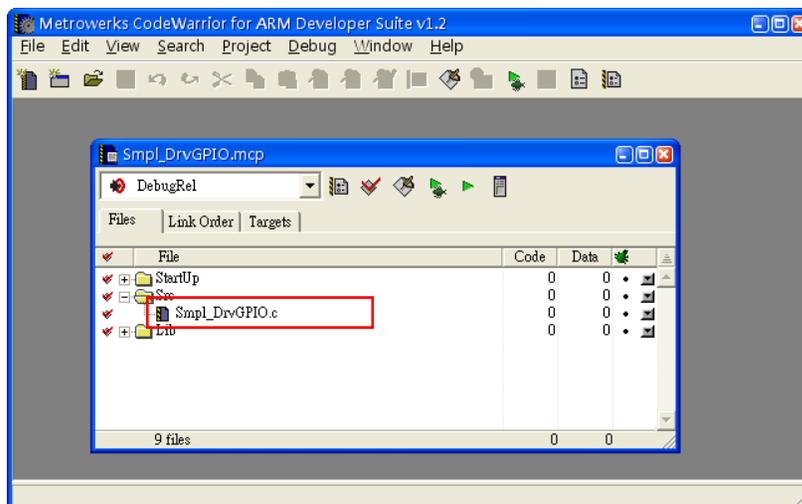
2.4.2. Creating a new project

It is recommended using the existed project as a template to create a new project. You may find the existed sample code which is nearest to your target application and copy it to new directory to be your first step. All driver sample code of NUC501 is target to be executed in SRAM and you should refer to the “APP_221_0001_Boot From SPI ROM” application note for booting from SPI ROM/Flash.

2.4.3. Modify the C Code

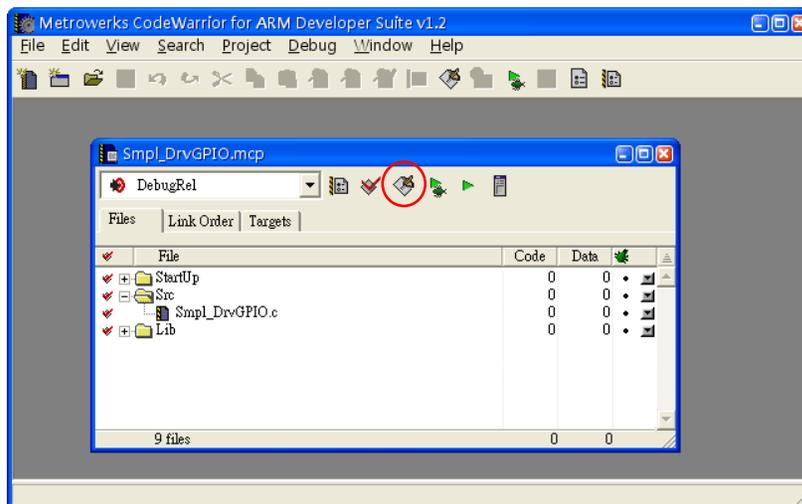
The source code of NUC501 could be divided into three groups, one is the startup code, one is the drivers and the other is main functions.

For example, the main function of the driver sample code of GPIO is “Smpl_DrvGPIO.c”.



Double click “Smpl_DrvGPIO.c” could open a text editor to modify the source code.

2.4.4. Build the C Code



To build the project, the user can click the icon , select Project->Make or press the hot key “F7”.

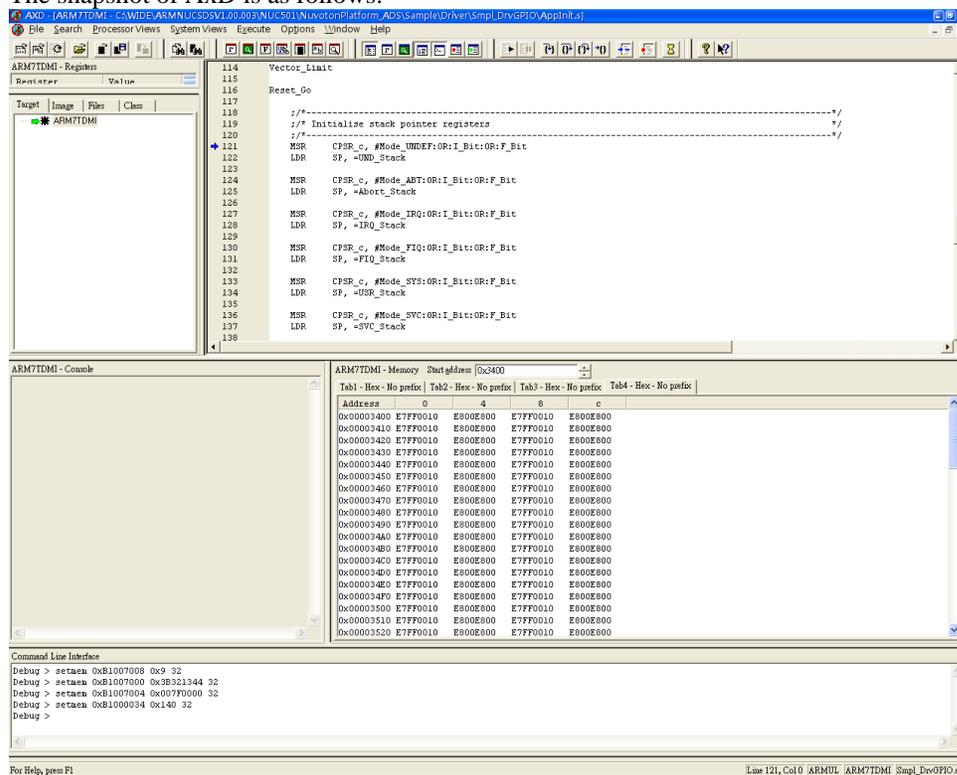
2.4.5. Open the Debugger - AxD

You can open the AxD manually or by CodeWarrior. To open the AxD in CodeWarrior, just click the

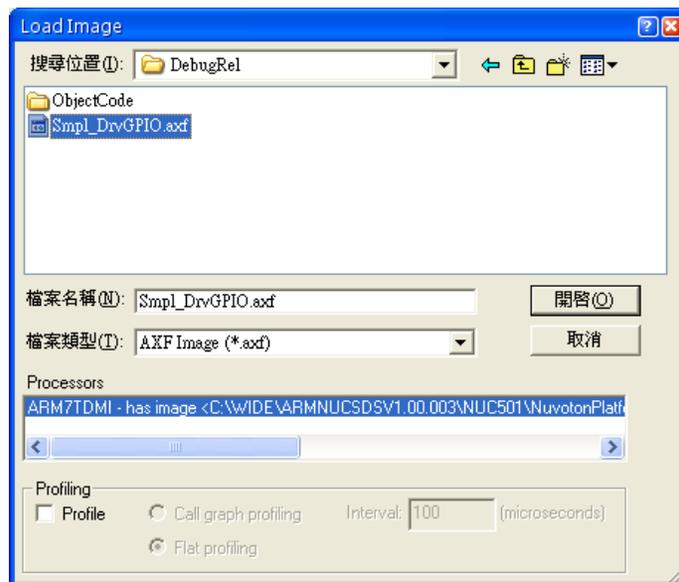
icon  by mouse. Then the AxD would be opened and the image would be loaded automatically, too.

Otherwise, you can double click the AxD icon  to start the AxD debugger.

The snapshot of AxD is as follows:



To load the executed image manually, the File->Load Image should be select.



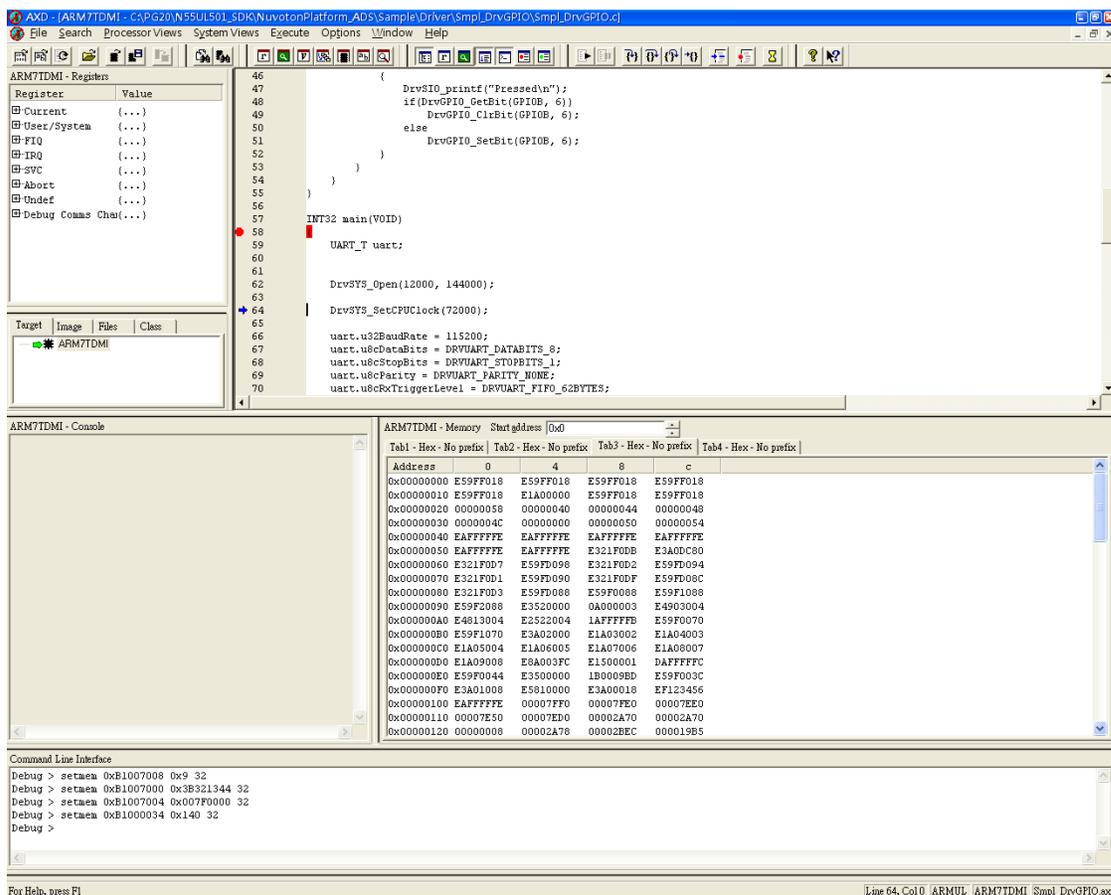
Due to the GPIO driver sample code is built to be executed in SRAM, the image will be load to SRAM (0x0~0x7FFF) directly after load the image.

2.4.6. Debug with AxD

The AxD support source level debugging, including step, go, breakpoint, watchpoint and etc. It is also able to modify the variables or memory content through ICE.

2.4.7. Conclusion

We can click the Go icon  of AxD to execute the program or just press the hot key “F5”. For example, the execute result of Smpl_DrvGPIO is as follows:



For more detail information of ADS, please refer to the documents of ARM Developer Suite documents.

3. Revision History

Version	Date	Description
V1.01.001	Apr. 30, 2009	• Created

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in equipment or systems intended for surgical implantation, atomic energy control instruments, aircraft or spacecraft instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for any other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications whereby failure could result or lead to personal injury, death or severe property or environmental damage.

Nuvoton customers using or selling these products for such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from their improper use or sales.