# NUC501 Quick Start Guide for Keil

V1.01.002

*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.

# Table of Contents

# 1. Introduction

## 1.1. About the Quick Start Guide

This Quick Start Guide will instruct you on how to use the Keil™ Microsoft Windows based software development tools with the NUC501 development board. It gives you the information necessary to use µVision3 for your own projects and provides an overview of the most commonly used µVision3 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 Keil™ µVision3 IDE

The µVision3 IDE is a Windows-based software development platform that combines a robust editor, project manager, and makes facility. The µVision3 integrates all tools including the C compiler, macro assembler, linker/locator, and HEX file generator. The µVision3 helps expedite the development process of your embedded applications by providing the following:

- Full-featured source code editor,
- Device database for configuring the development tool setting,
- Project manager for creating and maintaining your projects,
- Integrated make facility for assembling, compiling, and linking your embedded applications,
- Dialogs for all development tool settings,
- True integrated source-level Debugger with high-speed CPU and peripheral simulator,
- Advanced GDI interface for software debugging in the target hardware and for connection to Keil™ ULINK,
- Flash programming utility for downloading the application program into Flash ROM,
- Links to development tools manuals, device datasheets & user's guides.

# 2. Quick Start

## 2.1. Installing the Keil™ Software

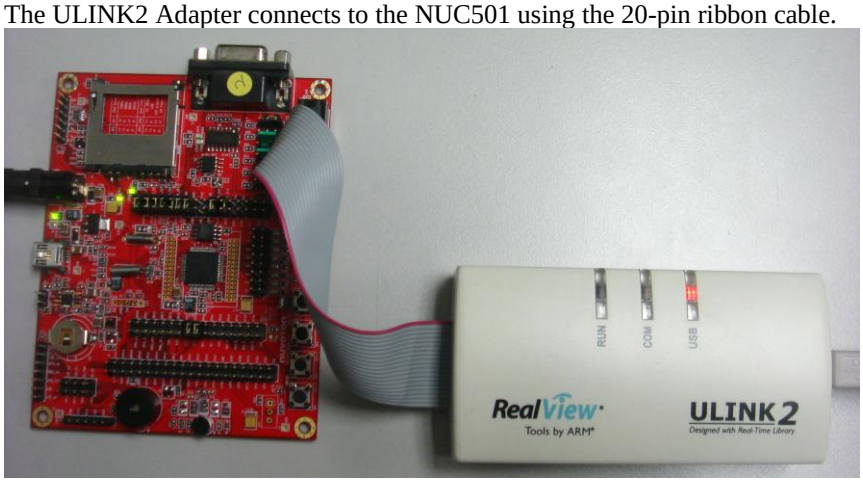
You can download the Keil™ RealView® Microcontroller Development Kit Evaluation software from http://www.keil.com/. It contains the Keil™ µVision3 IDE. The evaluation version of the tools has a 16K limit on images, but comes in a license-free version.

More information please reference Read Me First document from Keil™ about how to install Keil™ µVision3 software.

## 2.2. Connecting to your target

The target is powered via your PC, through its USB port. The Keil™ ULINK family of adapters connect the USB port of your PC to the JTAG port of your target board allowing you to download and debug embedded programs running on your target hardware.ULINK2 support standard JTAG and Real-Time Agent for on-the-fly target debugging.

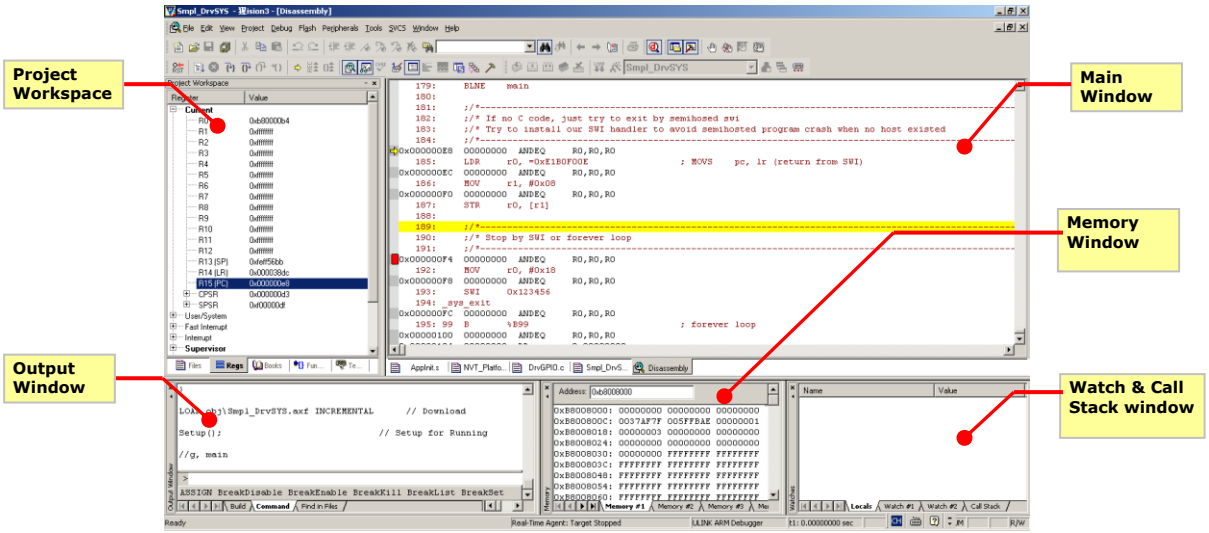
The ULINK2 Adapter connects to the NUC501 using the 20-pin ribbon cable.




## 2.3. µVision3 Overview

µVision3 has two operating modes:

- **Build Mode**: Allows you to translate all the application files and to generate executable programs. The features of the Build Mode are described under Creating Applications.
- **Debug Mode**: Provides you with a powerful debugger for testing your application. The Debug Mode is described in Testing Programs.



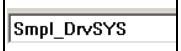

## 2.3.1. Build Process

The Build Target command runs the Compiler and Assembler. The tools automatically generate file dependencies so only those files that have changed are retranslated. You may enable additional Global Code Optimizations which are performed by incremental re-compilations of C modules and other utilities. The Project menu provides access to project files and dialogs for project management.

***Build Toolbar***


Only describe functions in common use as shown as below:

| Command Option | Tool Button | Function Description | Hot key |
|---|---|---|---|
| **Translate…** | | Translate current file | None |
| **Build Target** | | Translate modified files and build application | F7 |
| **Rebuild Target** | | Re-translate all source files and build application | None |

| | | | |
|---|---|---|---|
| **Batch Build** |  | Execute build component on the selected project/targets of a Multi-project workspace | None |
| **Stop Build** |  | Stop current build process | None |
| **Flash Download** |  | Call Flash download utility as configure under options. | None |
| **Target Option** |  | Set project components, configure tool environment and manage books. | None |
| **Select Current Project Target** |   | Select current target | None |
| **Manage Project** |  | Set Project Component, Configure tool environment and manage books. | None |
| **Configure Editor** |  | Change Colors, Fonts, Shortcuts and Editor option | None |

## 2.3.2. Debugger

The µVision3 IDE/Simulator/Debugger accelerates your learning efforts by providing a single environment for editing, simulating, and testing target hardware. Most debugger and editor functions may be quickly accessed from the toolbar.

You may use the context menu or the Editor Toolbar to insert breakpoints. Breakpoints you set while editing are activated when you start your debugging session. µVision3 marks the status of each source line in the Attributes column of the editor window. This provides a quick overview of the current breakpoint and execution status.

***Debugger***

Only describe functions in common use as shown as below:

| Command Option | Tool Button | Function Description | Hot key |
|---|---|---|---|
| **Reset CPU** |  | Set CPU to reset state | Ctrl+F5 |

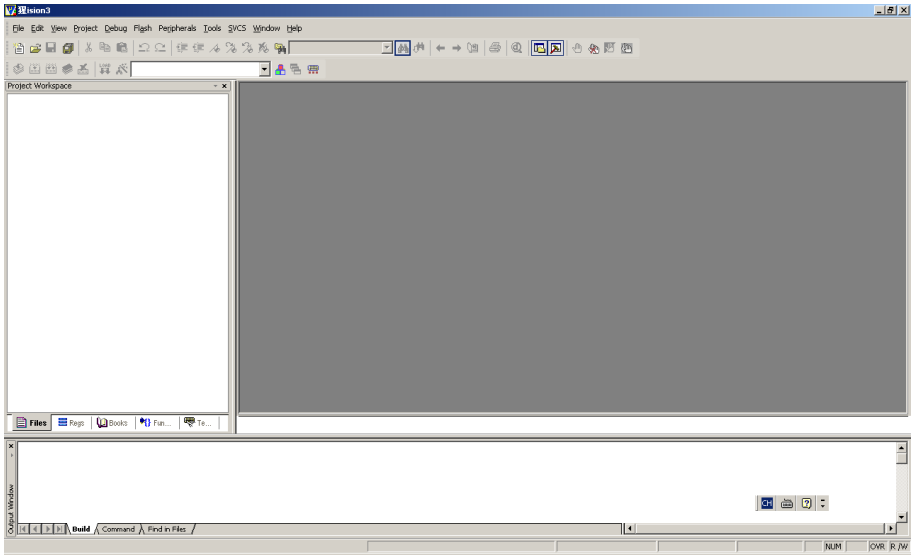
| | | | |
|---|---|---|---|
| **Go** |  | Run until the next active breakpoint | F5 |
| **Halt Execution** |  | Stop running | ESC |
| **Single step into** |  | Execute a single step into a function | F11 |
| **Step Over** |  | Execute a single step over a function | F10 |
| **Step Out** |  | Step out of current function | None |
| **Run till current line** |  | Execute until the current cursor line is reached | None |
| **Start/Stop debug Session** |  | Start or Stop uVision3 debug mode | None |
| **Insert/Remove Breakpoint** |  | Toggle breakpoint on current line | None |
| **Kill All Breakpoints** |  | Remove all breakpoints in the program | None |
| **Enable/Disable Breakpoint** |  | Enable/Disable breakpoint on current line | Alt+F7 |
| **Disable All Breakpoints** |  | Disable all breakpoints in the program | None |
| **Show next statement** |  | Show next executable statement /instruction | None |
| **Disassembly** |  | Show or hide Disassembly window | None |
| **Watch & Call Stack window** |  | Show or hide Watch & Call Stack window | None |
| **Memory window** |  | Show or hide Memory window | None |

## 2.4. Step-by-Step

This section details all of the materials necessary to download code to an ARM-based development board for debug in the Keil™ µVision3 IDE using the JTAG debug agent.

### 2.4.1. Starting the software

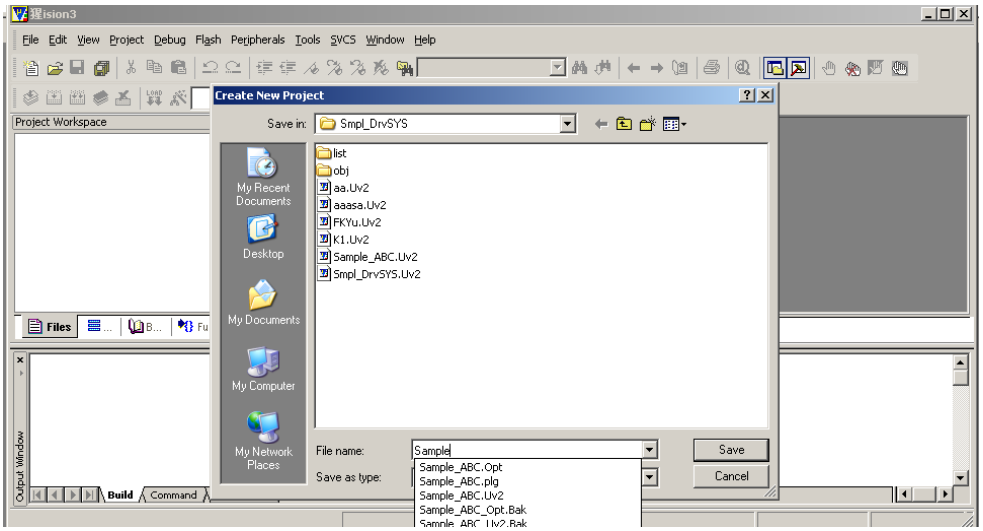
Double-click on the uVision icon to start the user interface. The compiler, assembler, linker and dScope will be called from within uVision in this tutorial. After you invoke mVision, the window shown as below appears. From this window, you can create projects, edit files, configure the tool, assemble, link, and invoke the debugger.




### 2.4.2. Creating a new project

Before writing any C-code, a project associated with our code needs to be created. This is done by first creating a new folder in the Keil directory in which your project will be saved. Next the Keil uV3 application can be launched and a new project is created. This is achieved by completing the following steps.

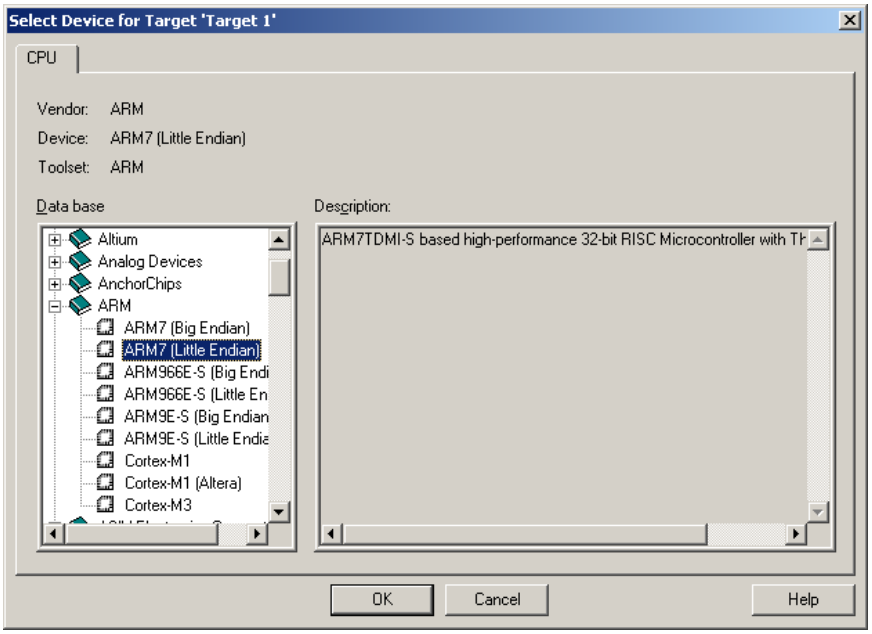
- Create a folder named Smpl_DrvSYS in your setting path
- Launch the uV3 application. Start -> Programs -> Keil µVision3
- Create a new project. From the main window, choose the 'Project' menu and select **New project** . And then a new window appears as shown below
- Select the folder that you've created previously ( Smpl_DrvSYS ) and on the bottom of the window type the name of your new project, eg. Smpl_DrvSYS and press SAVE.




## 2.4.3. Device Support

A new window appears as shown below and you are now required to configure your setup to target the specific ARM device you wish to use (in this example we will be using the ARM7) and the output file format generated after the compilation stage. This is achieved by completing the following steps.

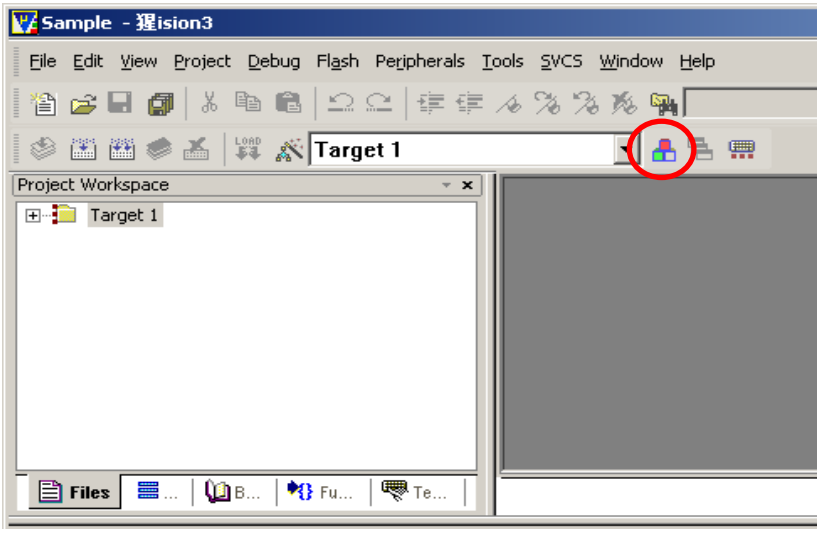
- Open the **ARM** folder.
- Select the item on which you will be developing, in this tech note we will be using the **ARM7(Little Endian)** as the target example.



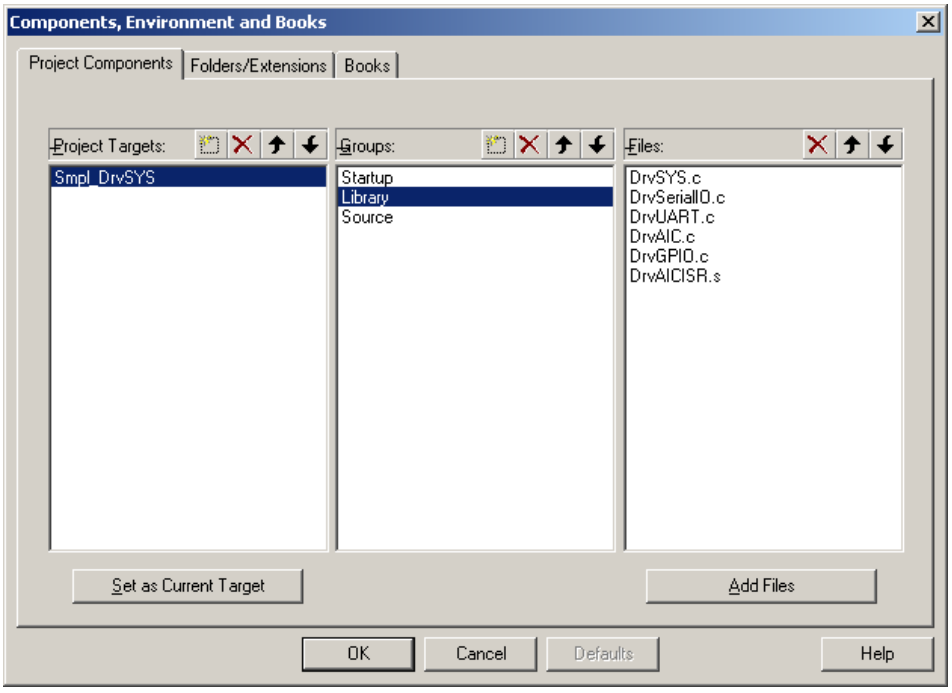
## 2.4.4. Project Management

µ Vision3 ensures easy and consistent Project Management. A single project file stores source file names and saves configuration information for Compiler, Assembler, Linker, Debugger, Flash Loader, and other utilities. The Project menu provides access to project files and dialogs for project management.

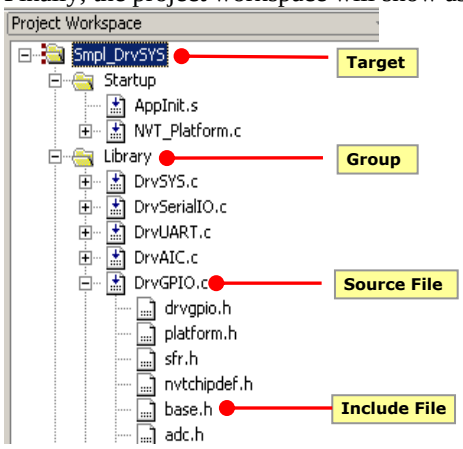
Select the icon to open **Management Project Component Setting**




A **Project Component Setting** window appears as shown below. According to your assignment, Create new group and link your source code.



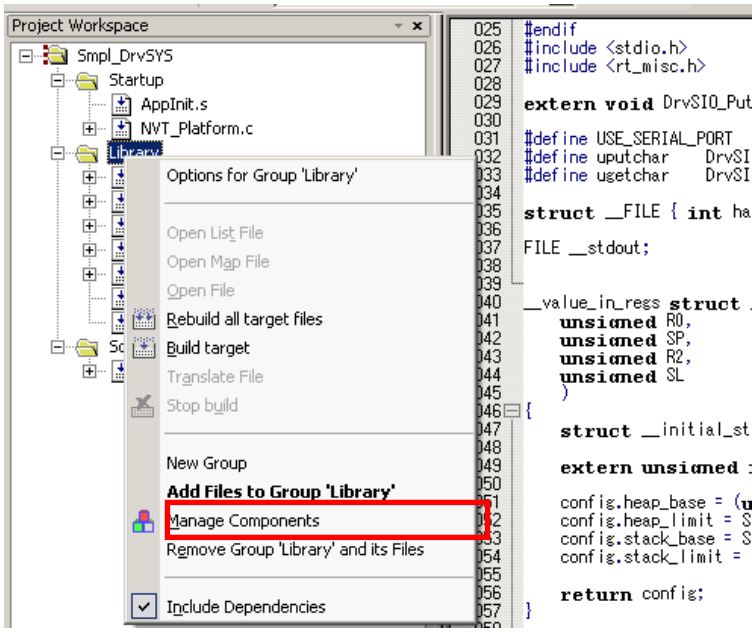
Finally, the project workspace will show as below:




## 2.4.5. Creating a C program

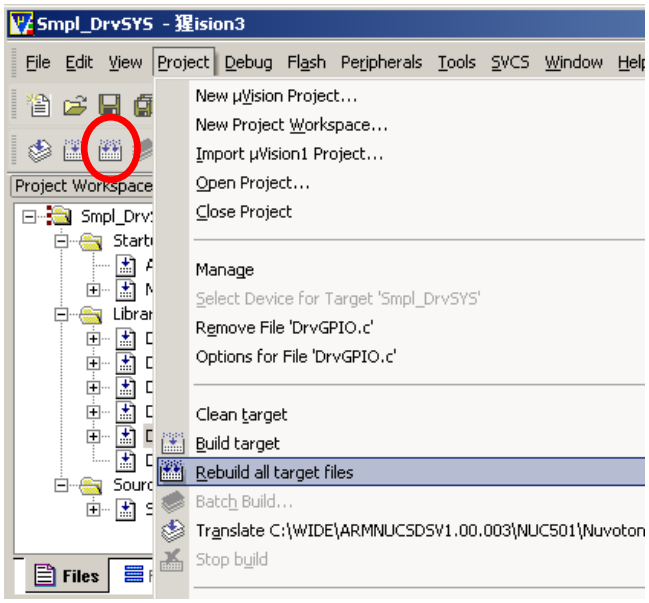
Now you also can write your C program. In the main window, choose the File pull-down menu and select New. A new window named <text1> will appear on the screen and you can write your code to <text1>.

Once you've typed all the code, again choose the File pull-down menu and select Save. A new Save dialog window appears. Save your new file as Smpl_DrvSYS.c in the Smpl_DrvSYS folder you had created earlier. At this stage, before compiling the C-program, we need to include it in our project. To do this you must click with the right mouse button on ‘Library ’and select Add Files to Group ‘Source Library’ as shown below.Select the Smpl_DrvSYS..c file that is in the Smpl_DrvSYS. folder and click on Add and then on Close.

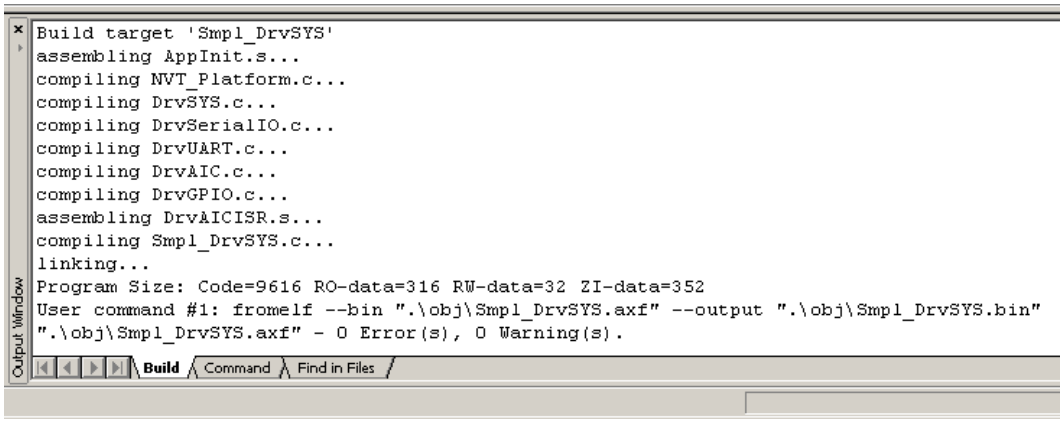


## 2.4.6. Compiling a C program

Select **Rebuild all target files** from the Project menu, or click on the **Rebuild all** button (icon).



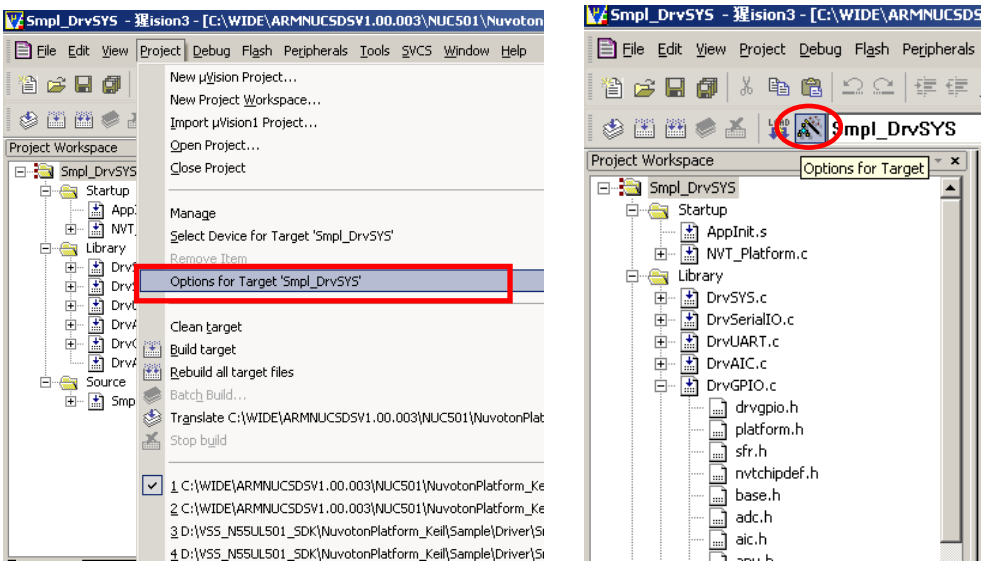

All of the source files are compiled and linked. The activity can be seen in the Build window at the bottom of the µVision IDE. (In this example, the process completes with an application named Saml_DrvSYS.axf and Saml_DrvSYS.bin built with no errors and no warnings.)



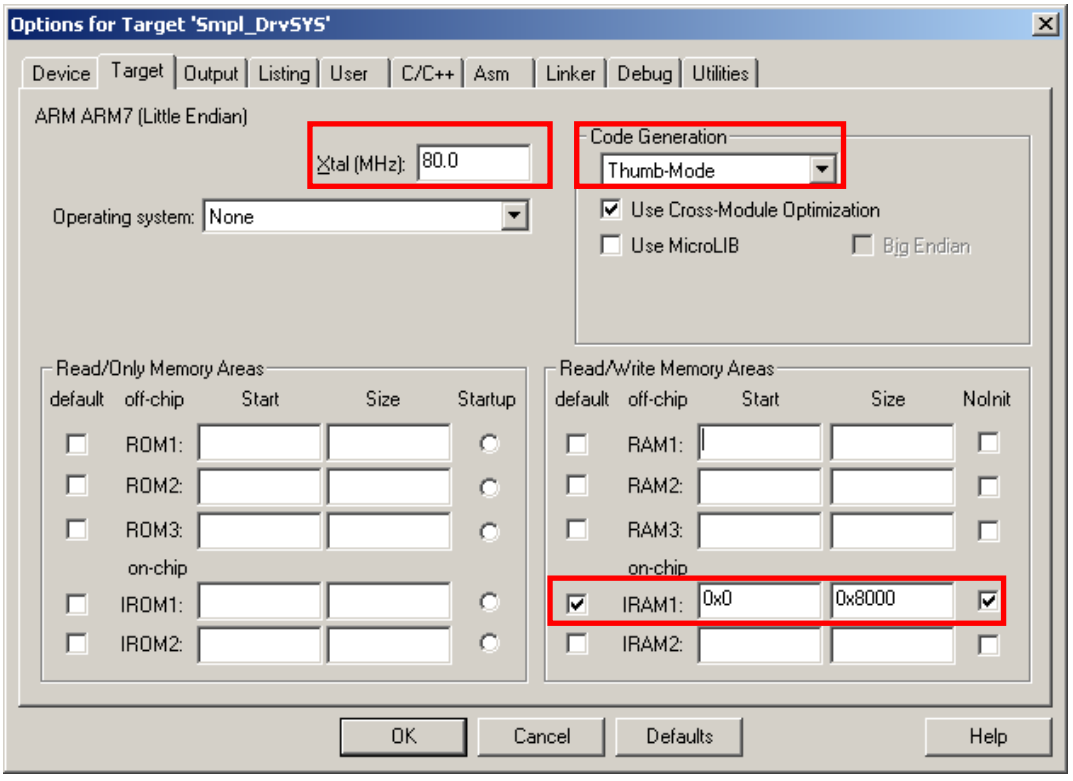

## 2.4.7. Connecting and Configuring the Hardware

Step-by-step to finished the section.

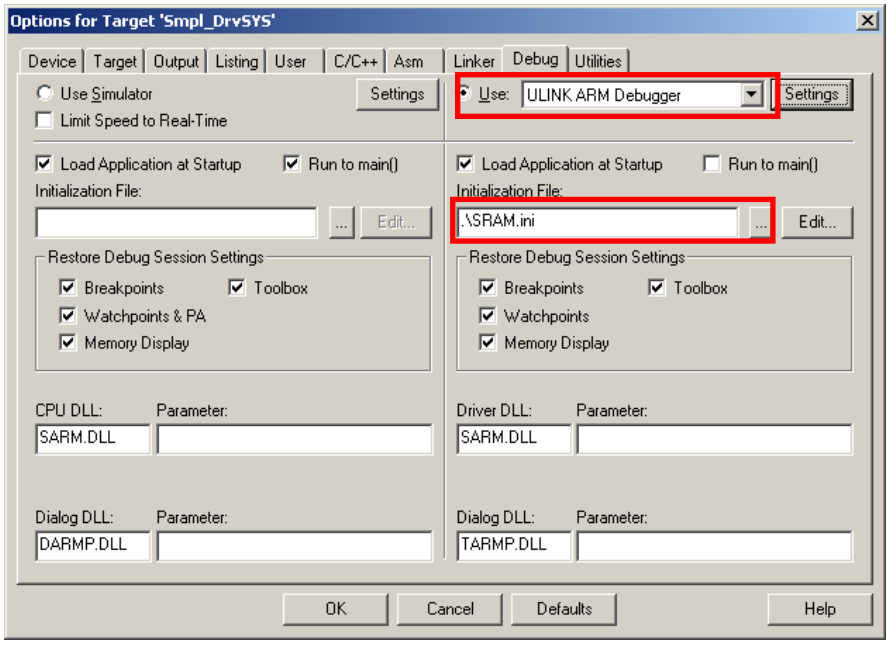
1. Click on **Project** => **Options for Target** => **your_target_name**. Or click on the **Options for Target** button (icon).




2. On Target tab, allows you to specify CPU and memory options. These are used to configure basic tool settings including those of the linker, debugger, and simulator. (In this example, select Xtal is 80MHz, code Generation to Thumb-Mode and IRAM1 size is 0x8000.)

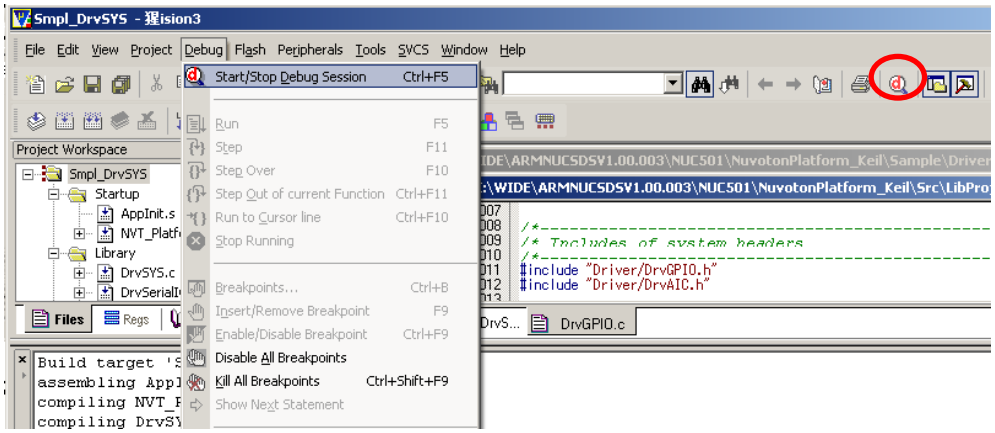



3. The setting dialog show up in a window on the center. Click **Debug** tab and Select **ULINK ARM Debugger**. You might need to select this from the drop-down menu if it is not already selected. (In NUC501, there is no flash to download. Please link the “SRAM.ini file in your project. ” )

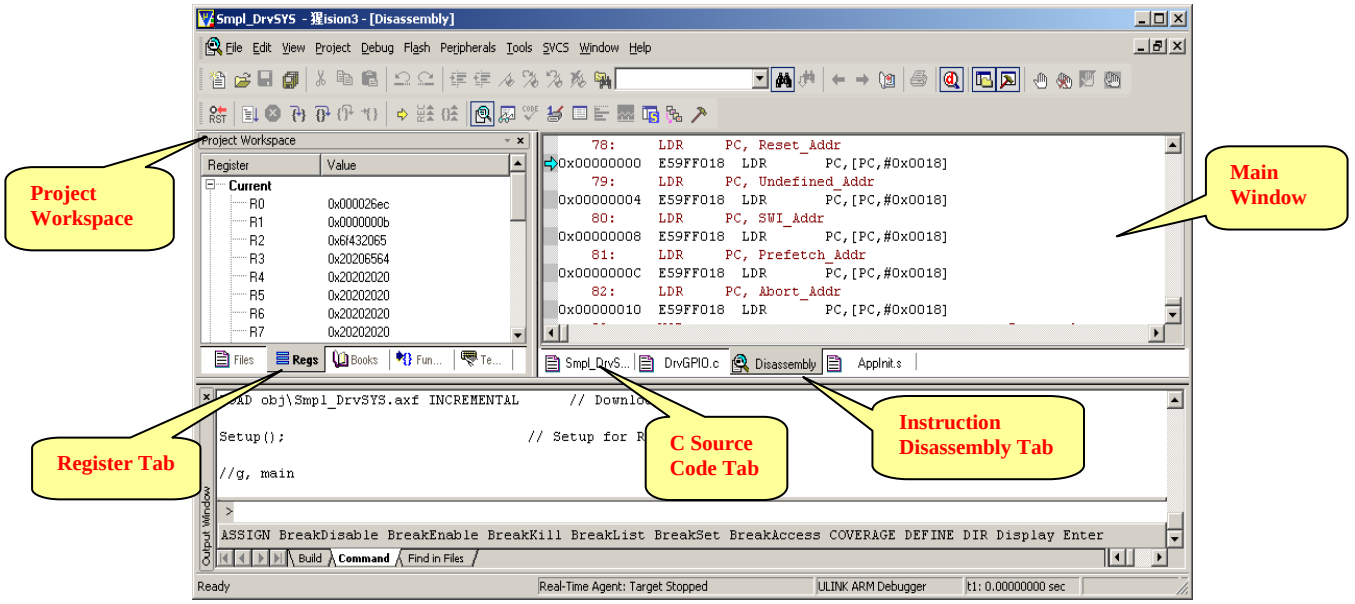



## 2.4.8. Simulating your source code

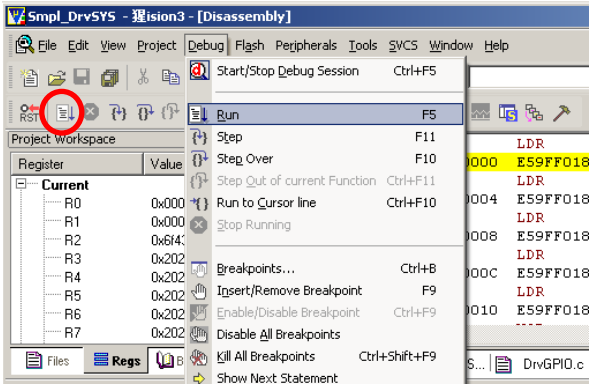
Another powerful feature of the uVision3 IDE is that it allows you to run your code. To start a simulation session you simply click on the on **Start/Stop Debug Session** option available from the **Debug** pull-down menu. Alternatively you can press <Cntrl+F7> or the 'Debug' icon available in the 'File' toolbar as shown as below.




The IDE switches to debugging mode. The processor registers show up in a window on the left, the debugger command window is visible at the bottom, and the main window shows the source code being debugged.



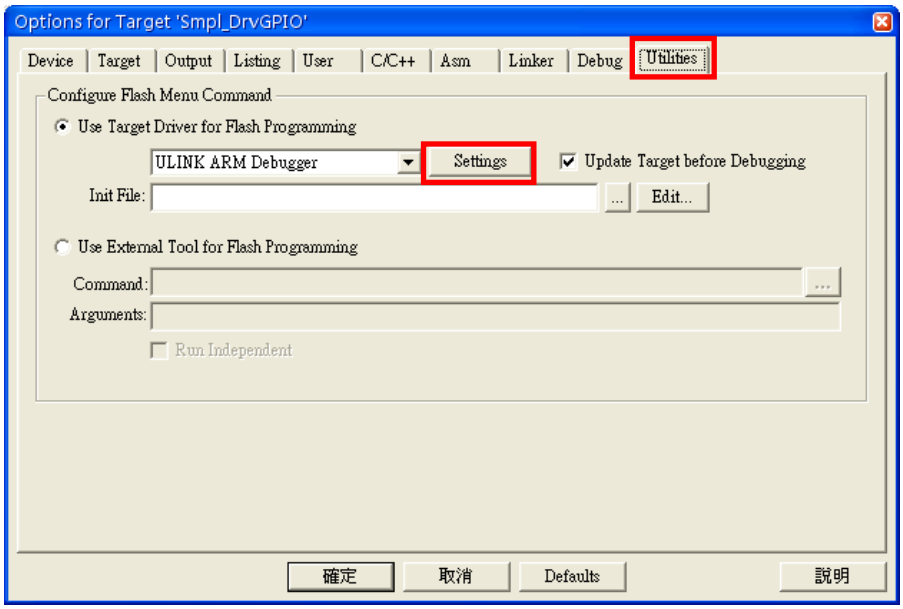

From here, you can examine and modify memory, program variables, and processor registers, set breakpoints, single step, and all other typical debugging activities. To run the program, select **Run** from the Debug menu, or click on the **Run** button (icon).

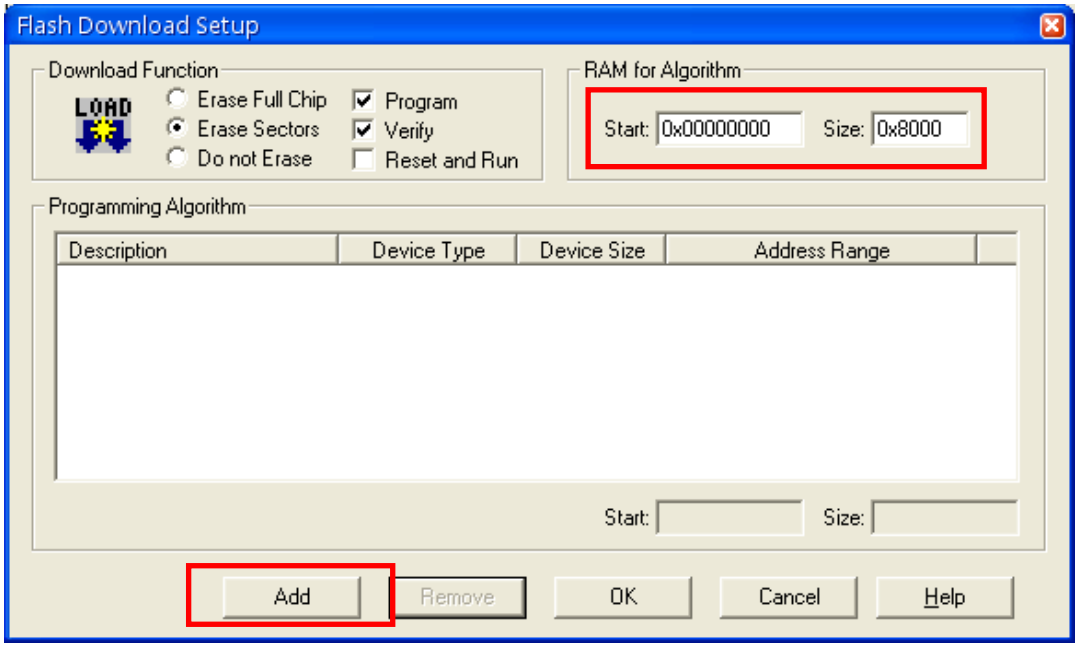


## 2.4.9. Flash Tool

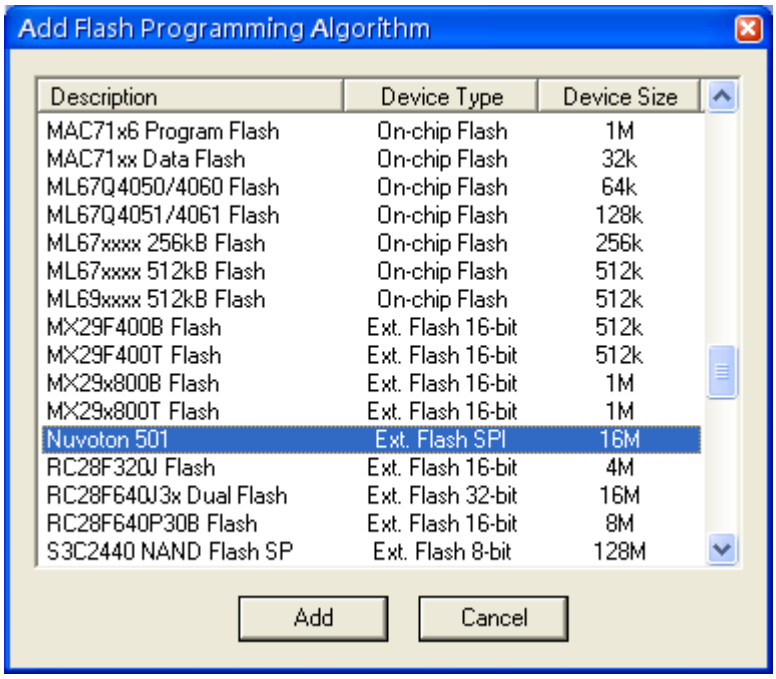
Keil tool chain supports to download image file to 501 DEV Board through ICE interface. A specified flash tool file should be provided to let Keil flash download function work. For 501, the flash tool is called Nuvoton501.FLX and this file could be found at \NuvotonPlatform_Keil\Sample\FlashTool.

To install the Nuvoton501.FLX, we need to copy it to the install directory of Keil, i.e. C:\ARM\Flash. After copy Nuvoton501.FLX to the specified directory, we can go back to the Keil IDE, open the Options for Targets dialog, select Utilities tab and open the Settings dialog.

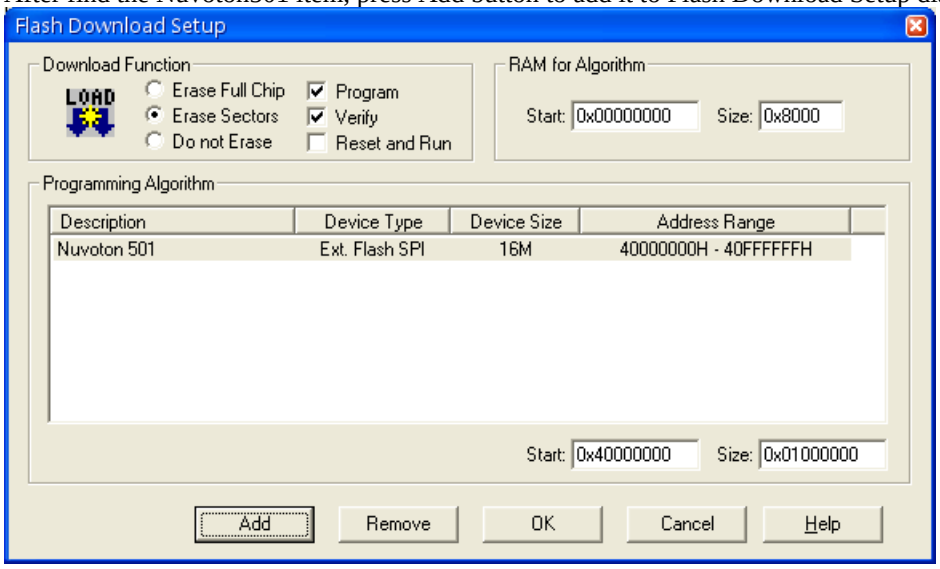



After the Settings button has been pressed, the Flash Download Setup dialog would be opened. Then we can set the start address to be 0x0, size to be 0x8000 in RAM for Algorithm setting. Finally, we press the Add button to add the flash tool. If the Nuvoton501.FLX was copy to \Keil\ARM\Flash, we should be able to find Nuvoton501 item in Add dialog:




After find the Nuvoton501 item, press Add button to add it to Flash Download Setup dialog.



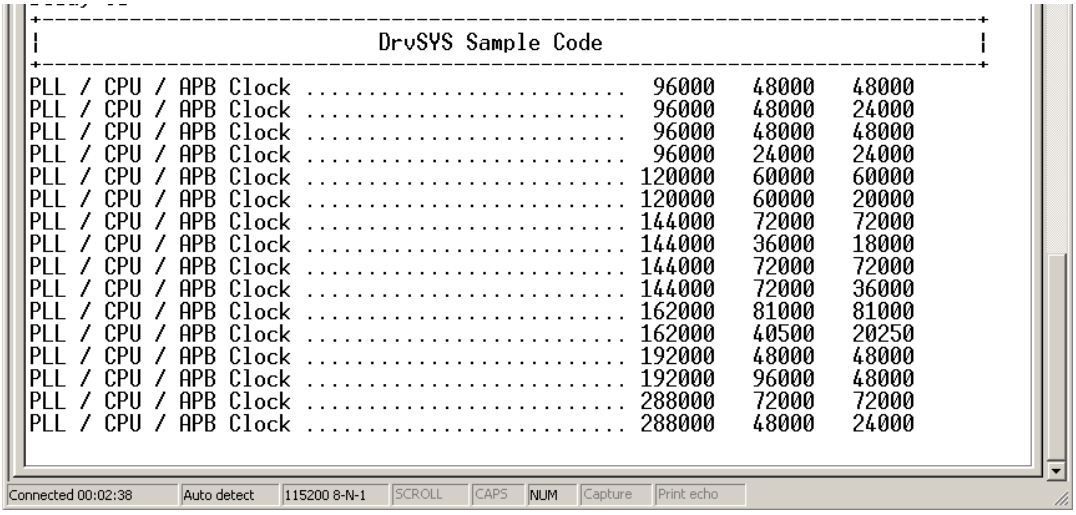

Now, we can press ok to finish the setting.

Note: Due to the flash base address is 0x40000000, we should set the RO base of linker according to the flash base address. Therefore, the Startup files, initial files(SRAM.ini, ROM.ini), via file and etc, should be modified according to this change.

All driver samples of 501 provide two project files to support execute in SRAM or execute in SPI Flash. These project files are named as Smpl_DrvXXX_ROM for execute in SPI Flash and named as Smpl_DrvXXX_SRAM for execute in SRAM. The user also can use them as a template to create their own project.

## 2.4.10. Conclusion

You have now installed the Keil™ RealView® Microcontroller Development Kit, and used it to build, load, and run a demonstration application on your Nuvoton® Development Board. From here, you can experiment with the debugger or start creating your own application using the Smpl_DrvSYS program as an example. If Smpl_DrvSYS project sample starts running, and you should see some text output to the hyper terminal display as shown as below:



# 3. Revision History

| Version | Date | Description |
|---|---|---|
| V1.01.001 | Apr. 14, 2009 | ● Created |
| V1.01.002 | Apr. 22, 2009 | ● Modified |

## 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.