

8-bit Microcontroller

8051 Family KEIL™ μ Vision® Driver User Manual

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

Table of Contents

1	Overview	3
2	System Requirements	4
3	Install Nuvoton 8051 KEIL™ μVision® Driver	5
4	Start to Work with μVision® Interface	6
4.1	Creating a New Project	6
4.2	Setting Debug Options	9
5	Use the Debug Features	11
5.1	Debugging a New Project	11
5.2	Updating CONFIG	17
5.3	Read or Write SFR	19
5.4	Read or Write Direct RAM	20
5.5	Read or Write Indirect RAM	20
5.6	Read or Write XRAM	22
5.7	Read Code (N76E003 / N76E885 / N76E616 Only)	23
6	Project Working Flow	24
6.1	Program Code Download Flow	24
6.2	Program Code Debug Flow	24
7	Check Version	25
7.1	Nu-Link-Me Bridge Firmware Version	25
7.2	Appendix Limitations of KEIL™ C51 Evaluation Edition	26
8	REVISION HISTORY	27

1 OVERVIEW

The Nuvoton 8051 KEIL™ μVision® Driver allows the KEIL™ μVision®2, μVision®3, μVision®4 or μVision®5 IDE to communicate with Nuvoton in-circuit-emulation (ICE) or on-chip-debug (OCD) device. The μVision IDE can start and stop program execution, set breakpoints, check variables, inspect and modify memory contents, and perform single-step through programs running on your actual target hardware. This document describes how to install and use Nuvoton 8051 KEIL™ μVision® Driver with programs written by the KEIL™ C51/A51 compiler.

After installing the Nuvoton 8051 KEIL™ μVision® Driver after Rev. 1.06, it additionally provides Nuvoton own CPU database “Nuvoton 8051 Devices”. It collects all Nuvoton 8051 MCU and benefits user easily building KEIL™ μVision® project. For N76E885 and N76E616, the N76E885.h and N76E616.h file is also installed in KEIL™ general library. User can include it directly by writing e.g. “#include < N76E885.h file >” without caring of its file location.

2 SYSTEM REQUIREMENTS

- Software: KEIL™ μVision®2, μVision®3, μVision®4 or μVision®5.
- Hardware: Support evaluation board with Nu-Link-Me interface.

The Nuvoton 8051 KEIL™ μVision® Driver can support chips as listed in *Table 2-1*.

LPC 1T 8051 Product Line	
N76E003 series	N76E003AT20, N76E003AQ20
N76E616 series	N76E616AL48
N76E885 series	N76E885AT28, N76E885AT20, N76E885AQ20
LPC 4T 8051 Product Line	
N79E85J	N79E715AT28, N79E715AT20, N79E715AS28, N79E715AS20, N79E715AS16, N79E855AWG, N79E854AWG, N79E845AWG, N79E844AWG, N79E8432ASG, N79E815AT28, N79E815AT20, N79E814AT28, N79E814AT20 , N79E8132AS16

Table 2-1 Supported Chips

3 INSTALL NUVOTON 8051 KEIL™ μVISION® DRIVER

A complete set of files needed to run the Nuvoton 8051 KEIL™ μVision® Driver inside the μVision®2, μVision®3, μVision®4 or μVision®5 interface is contained in the package.

All steps are performed automatically by setup program. Please follow the steps below:

- Install μVision® and the C51/A51 Compiler on your computer.
- Run [Nuvoton_8051_Keil_uVision_Driver_v2.00.6561.exe](#), and then select the <KEIL install path>. The new files will appear inside <KEIL install path>\C51\BIN\Nuvoton_8051_Keil_uVision_Driver.DLL.

4 START TO WORK WITH μ VISION® INTERFACE

4.1 Creating a New Project

To create a new project, please follow the steps below.

Step–1: Run the program and invoke “**Project – New μ Vision Project**” to open the *Create New Project* form.

Step–2: Input the new project name and click the “**Save**” button.

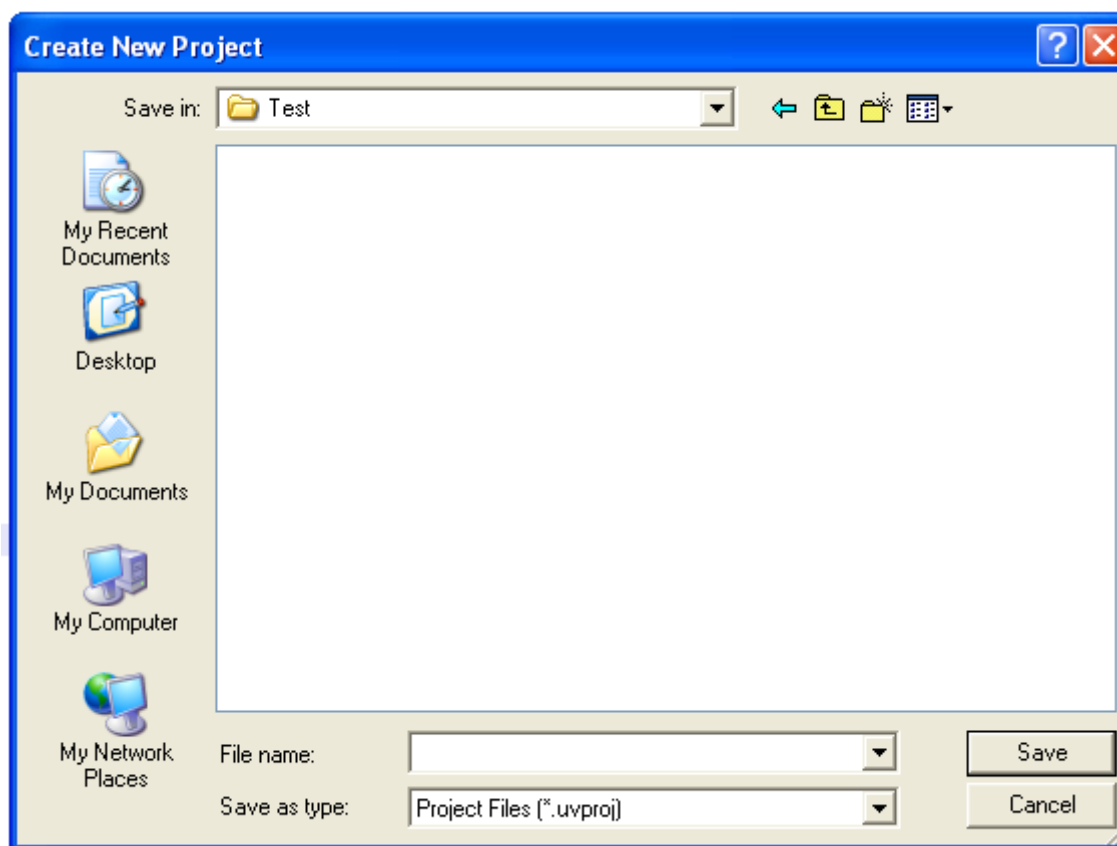


Figure 4-1 Create a New Project

Step-3: Select “Nuvoton 8051 Devices” and click the “OK” button.

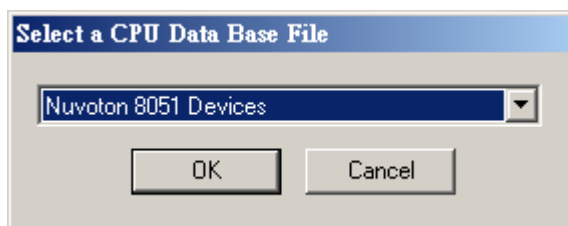


Figure 4-2 Select a CPU Database

Step-4: Select your device.

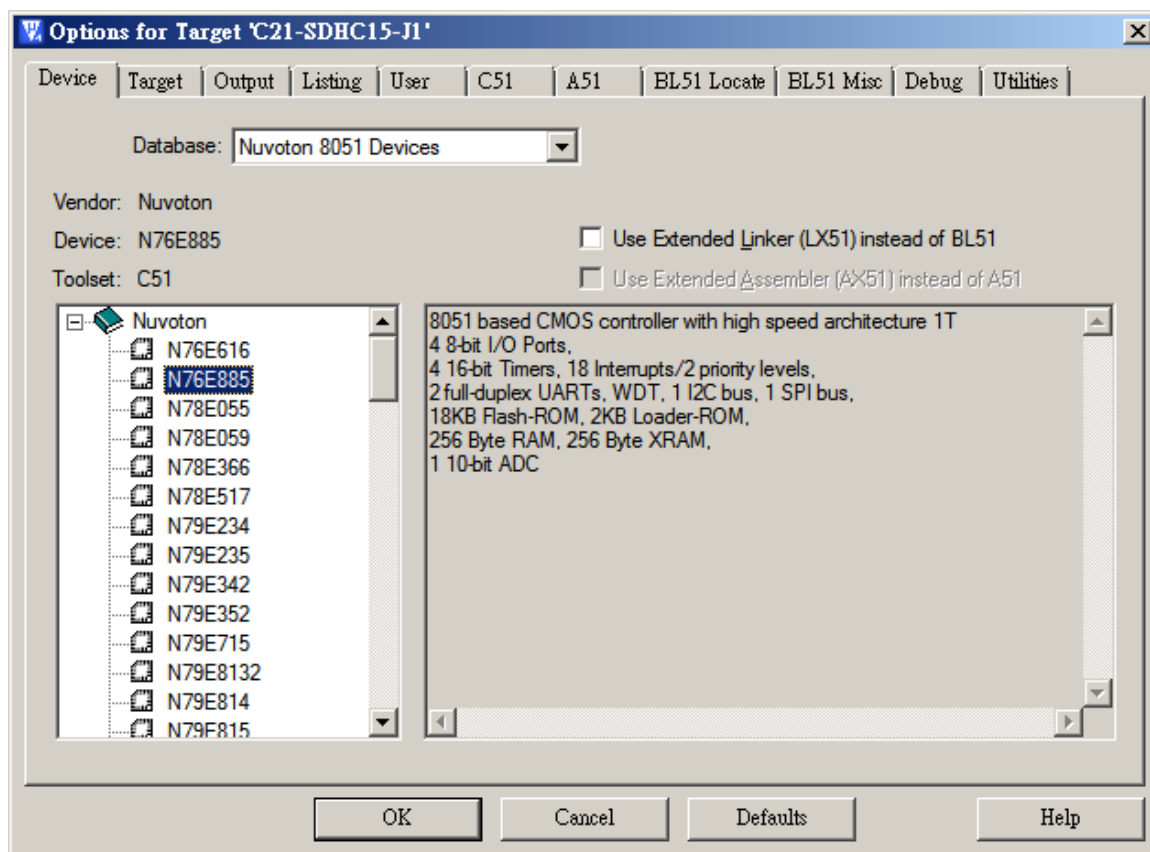


Figure 4-3 Select Your Device from the Database List

Step-5: Click “Yes” to copy “STARTUP.A51” to the project folder and complete the flow (as shown in *Figure 4-5*).

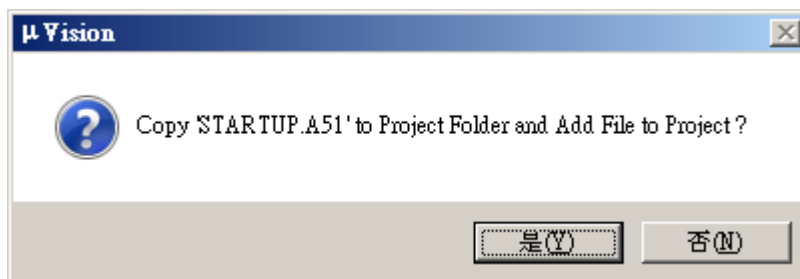


Figure 4-4 Add “STARTUP.A51” to Project

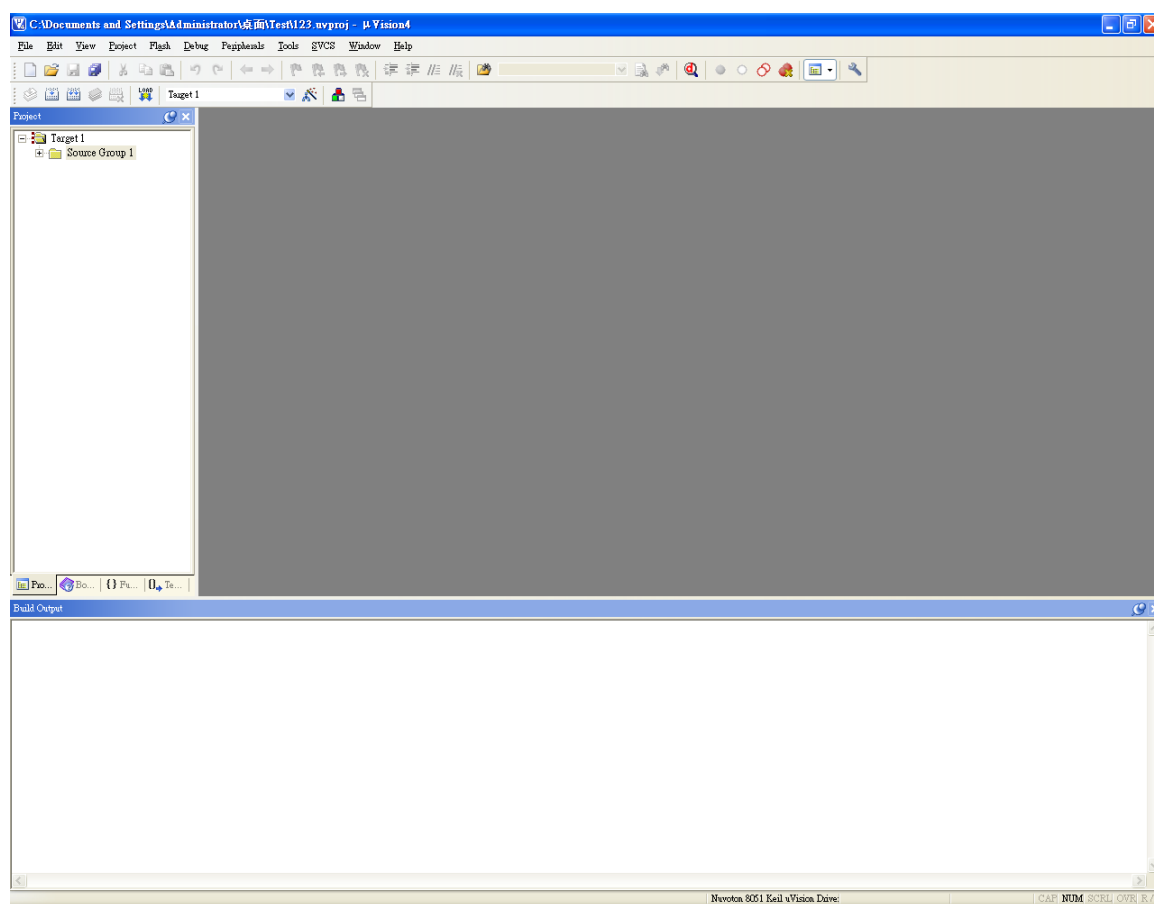


Figure 4-5 Complete the Process to Create a New Project

4.2 Setting Debug Options

To set the debug options, please follow the steps below.

Step-1: Select the **Debug** tab in the “Options for Target” form.

Step-2: Make sure that the “Use:” option is checked.

Step-3: Select “Nuvoton 8051 KEIL μ Vision Driver”.

Step-4: Set options as shown in the figure below.

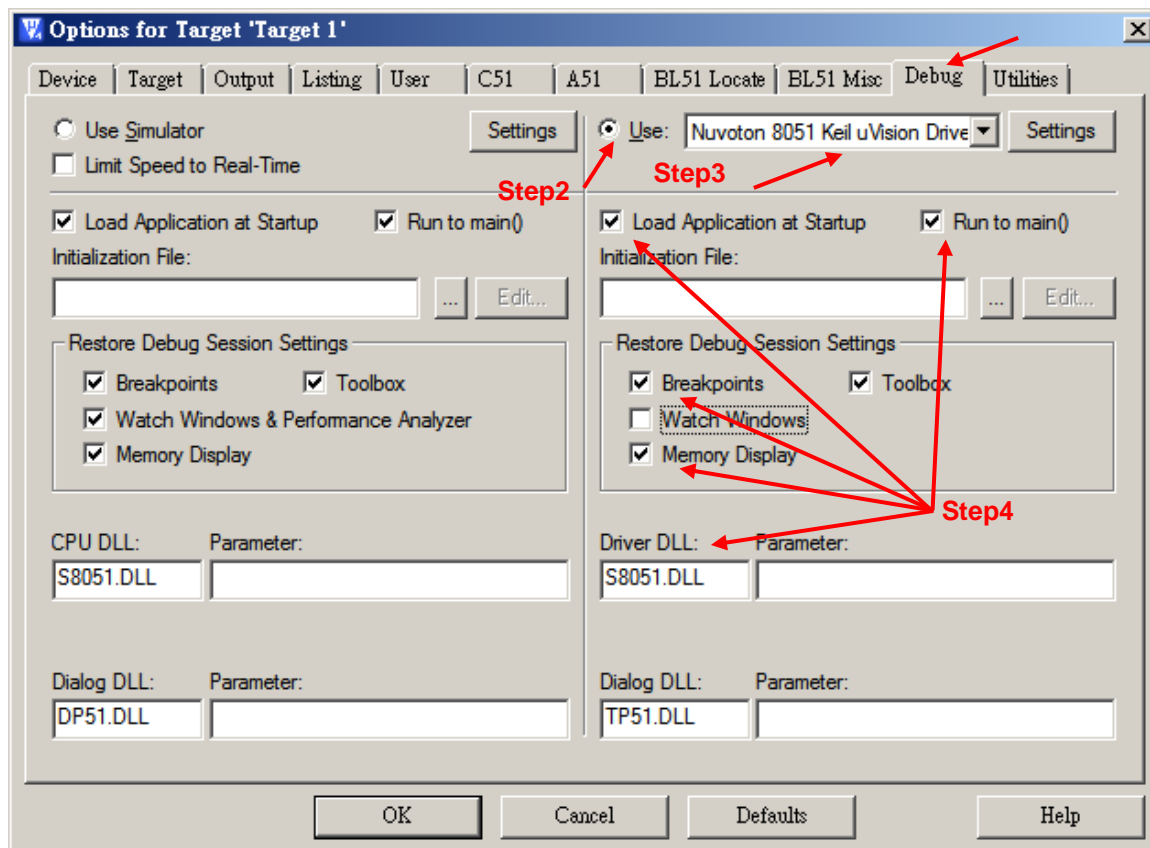


Figure 4-6 Set Debug Options

Step-5: Select the **Utilities** tab in the “Options for Target” form.

Step-6: Make sure that the “Use Target Driver for Flash Programming:” option is checked.

Step-7: Select “Nuvoton 8051 KEIL μVision Driver”.

Step-8: Click the “Settings” button.

Step-9: Check all options in the *Flash Download Setup* form.

Step-10: Click “OK” and complete the settings.

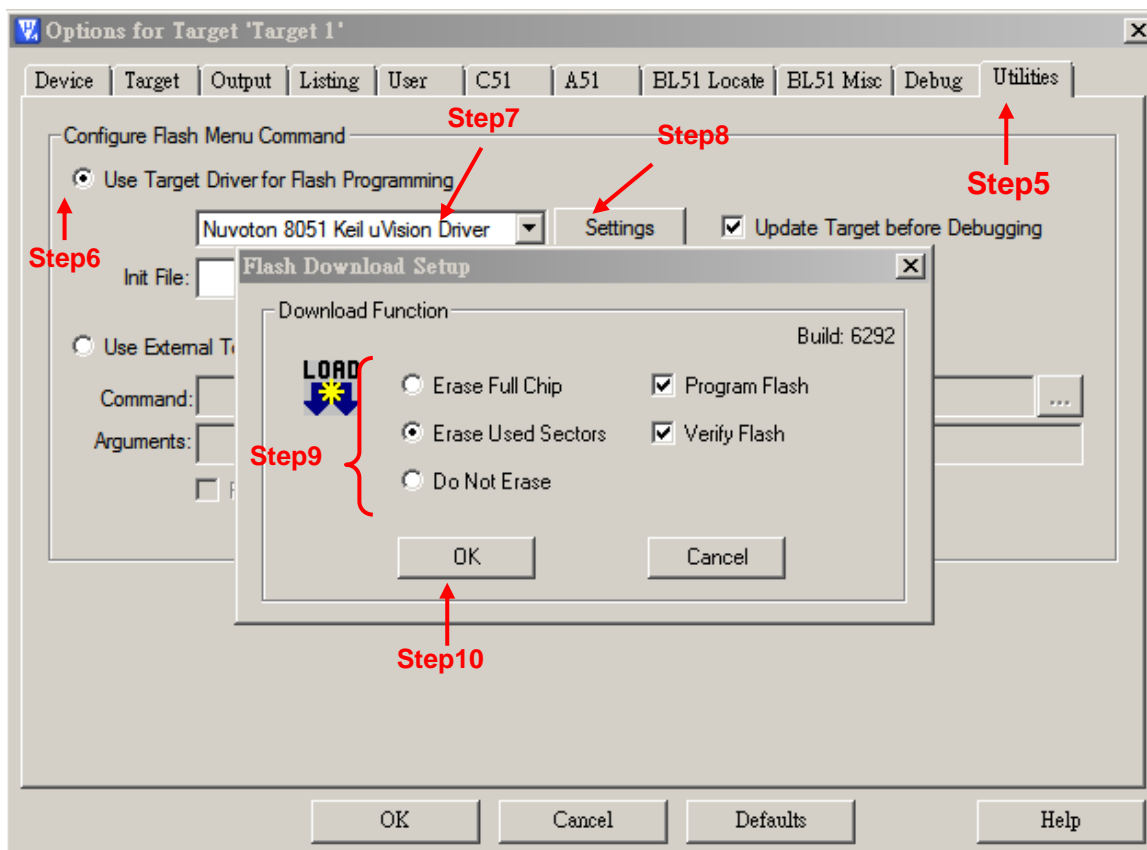


Figure 4-7 Set Download Options

5 USE THE DEBUG FEATURES

This chapter describes how to use Nuvoton 8051 KEIL™ μ Vision® Driver with 8051 C51/A51 application based on the example project.

5.1 Debugging a New Project

Please follow the steps below after the installation had been performed.

Step–1: Invoke “File – New” to create a new file and save into C language.

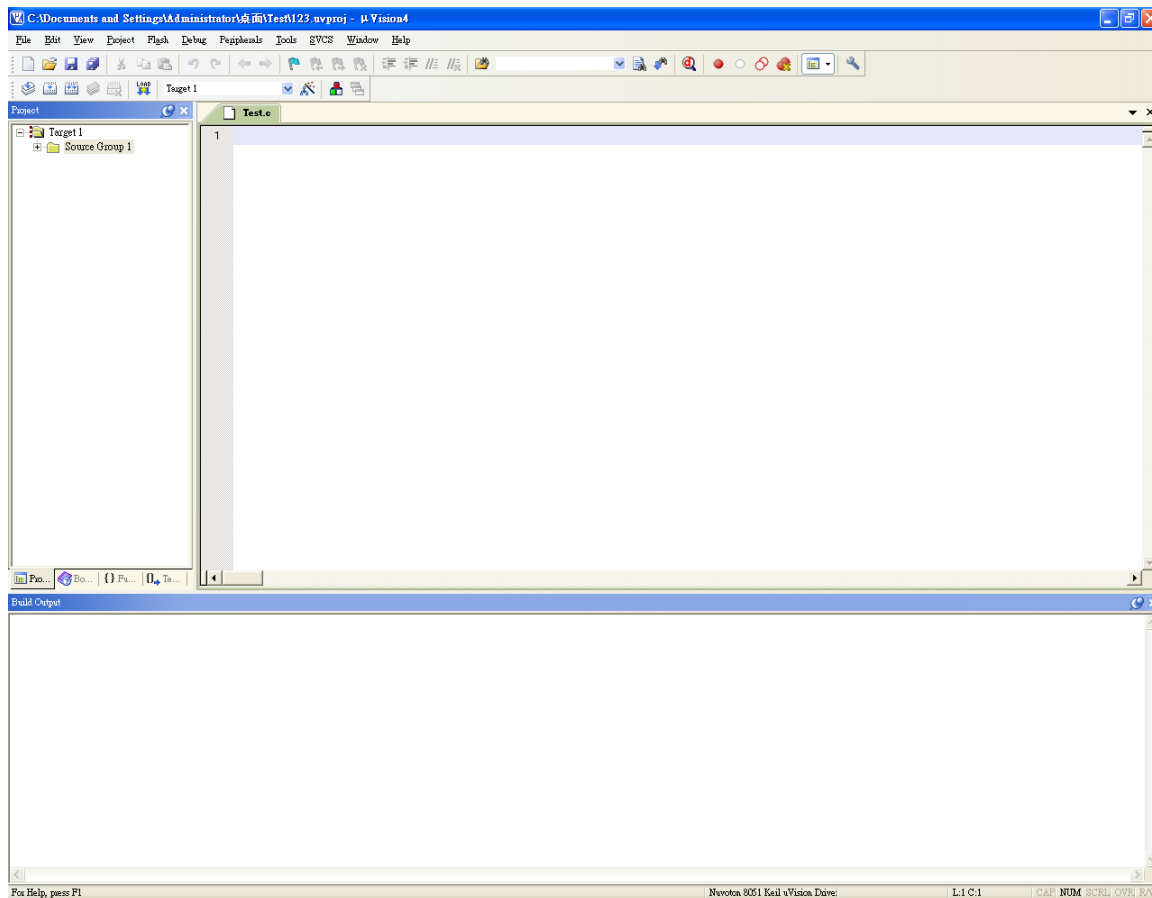


Figure 5-1 Create a New C File

Step-2: Write sample code as shown in *Figure 5-2* and save it.

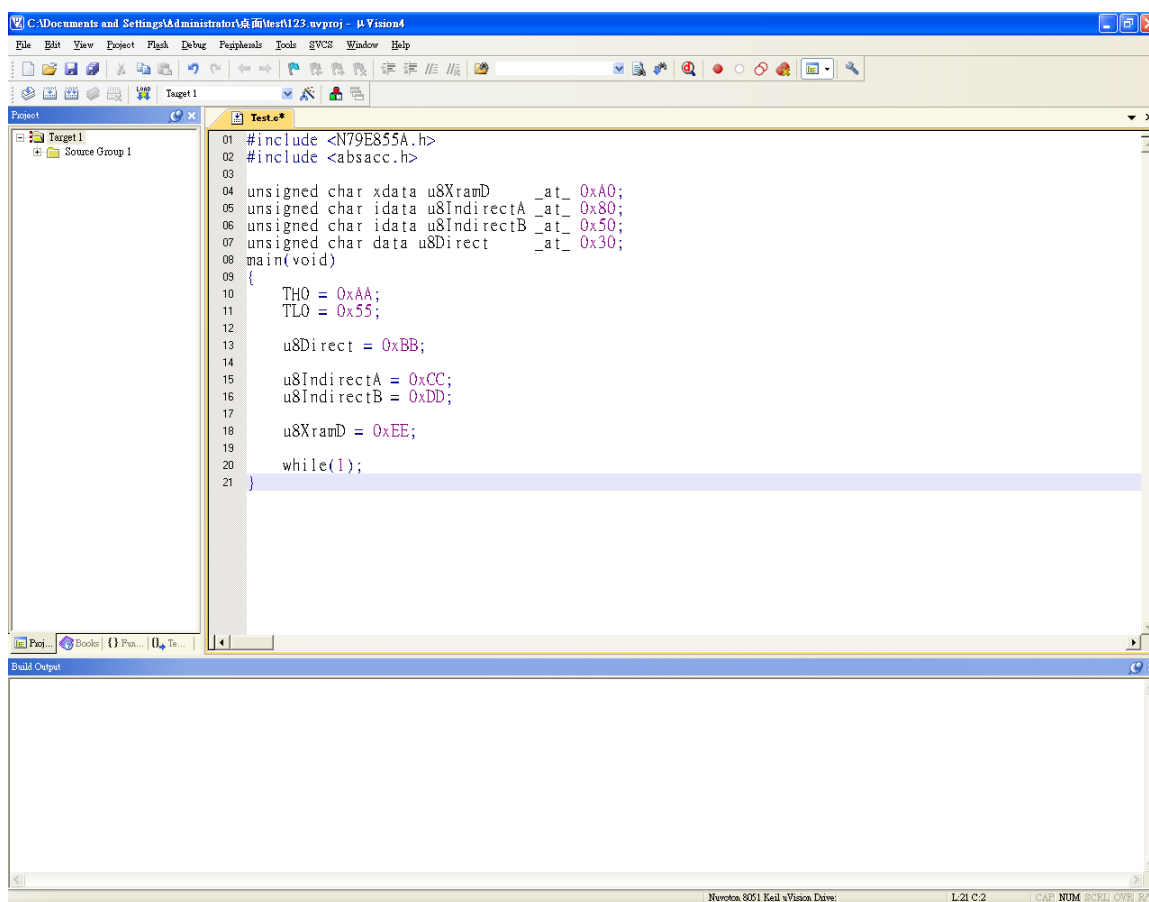


Figure 5-2 Write Sample Code

Step-3: Invoke “**Project – Manage – Components, Environment, Books**”. Add the C file that had been created through *Step-2*.

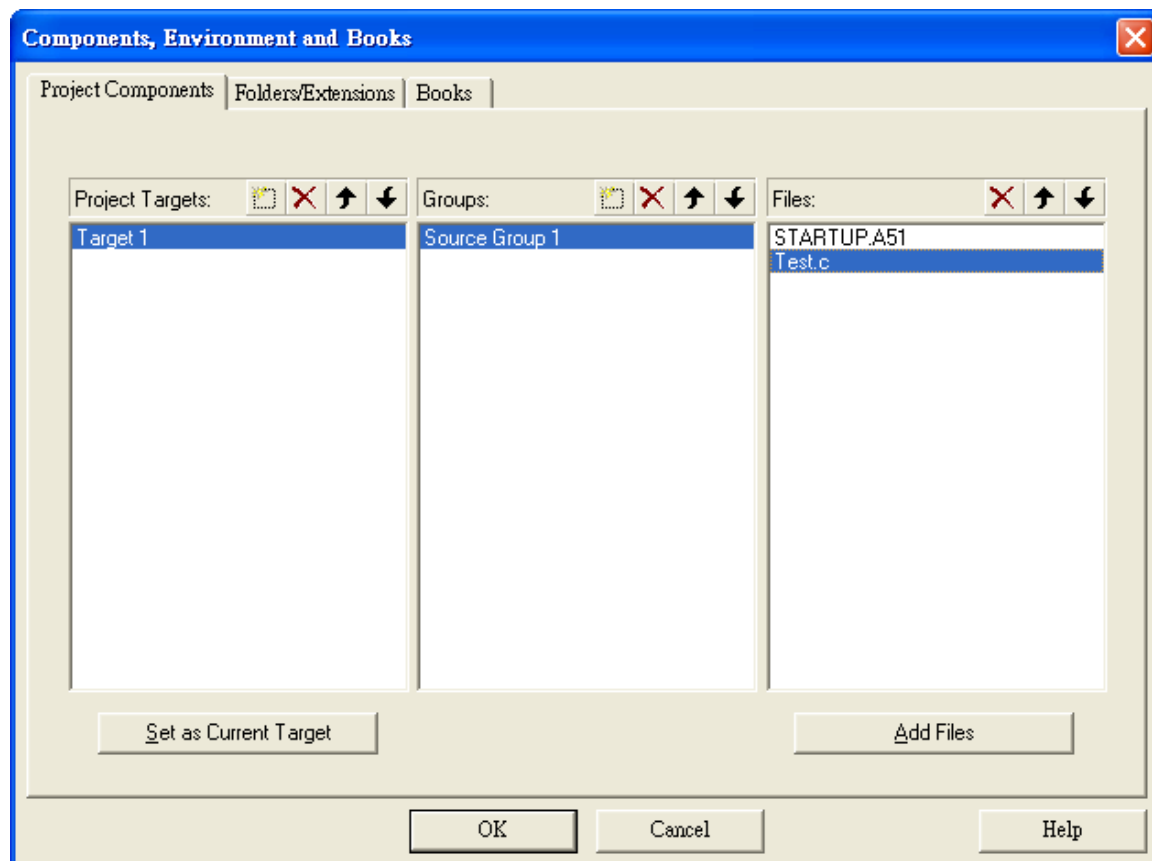


Figure 5-3 Add the C File in Your Project

Step-4: Invoke “Rebuild all target files” to build the project. Check if the *Build Output* window shows no warnings and errors.

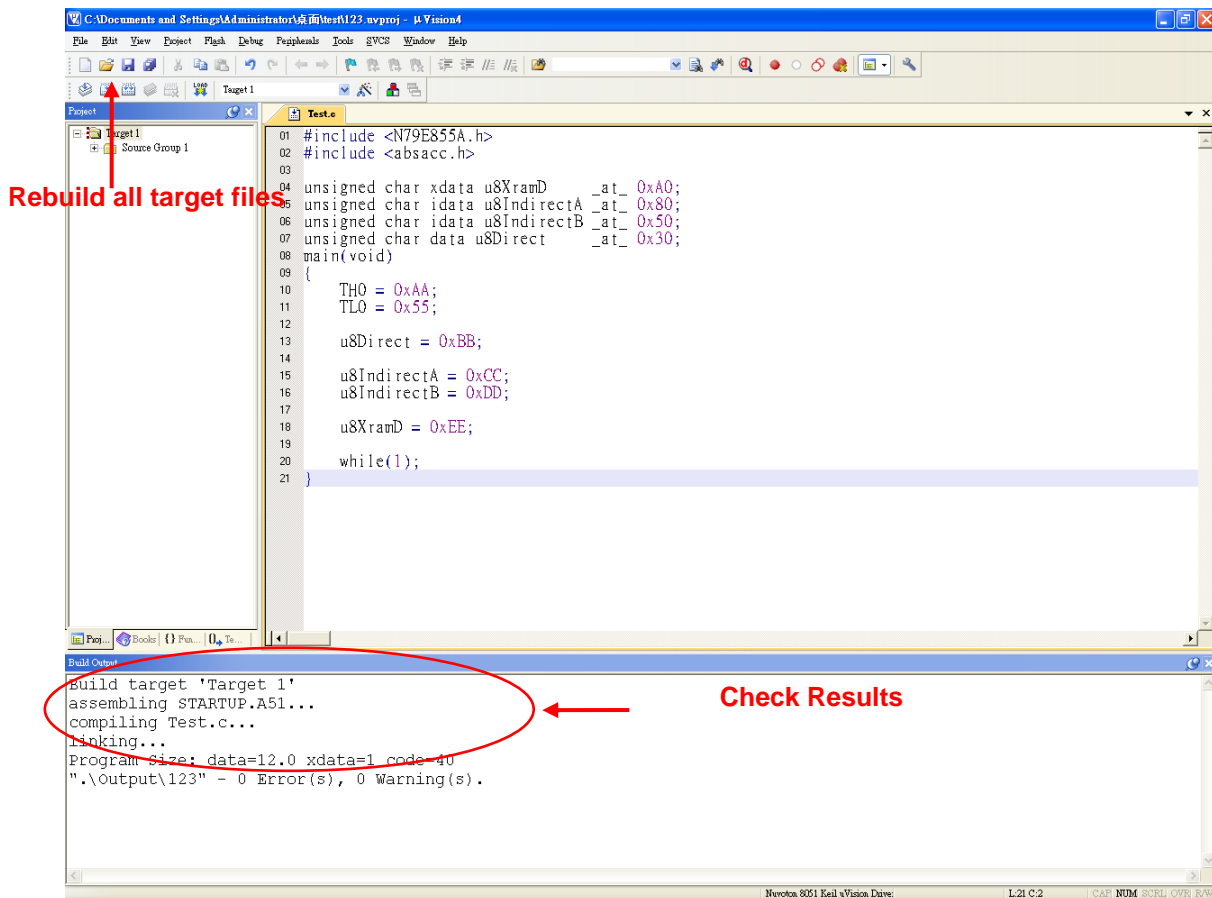


Figure 5-4 Build Target and Display Results

Step-5: Click “Download” to download the code to flash memory.

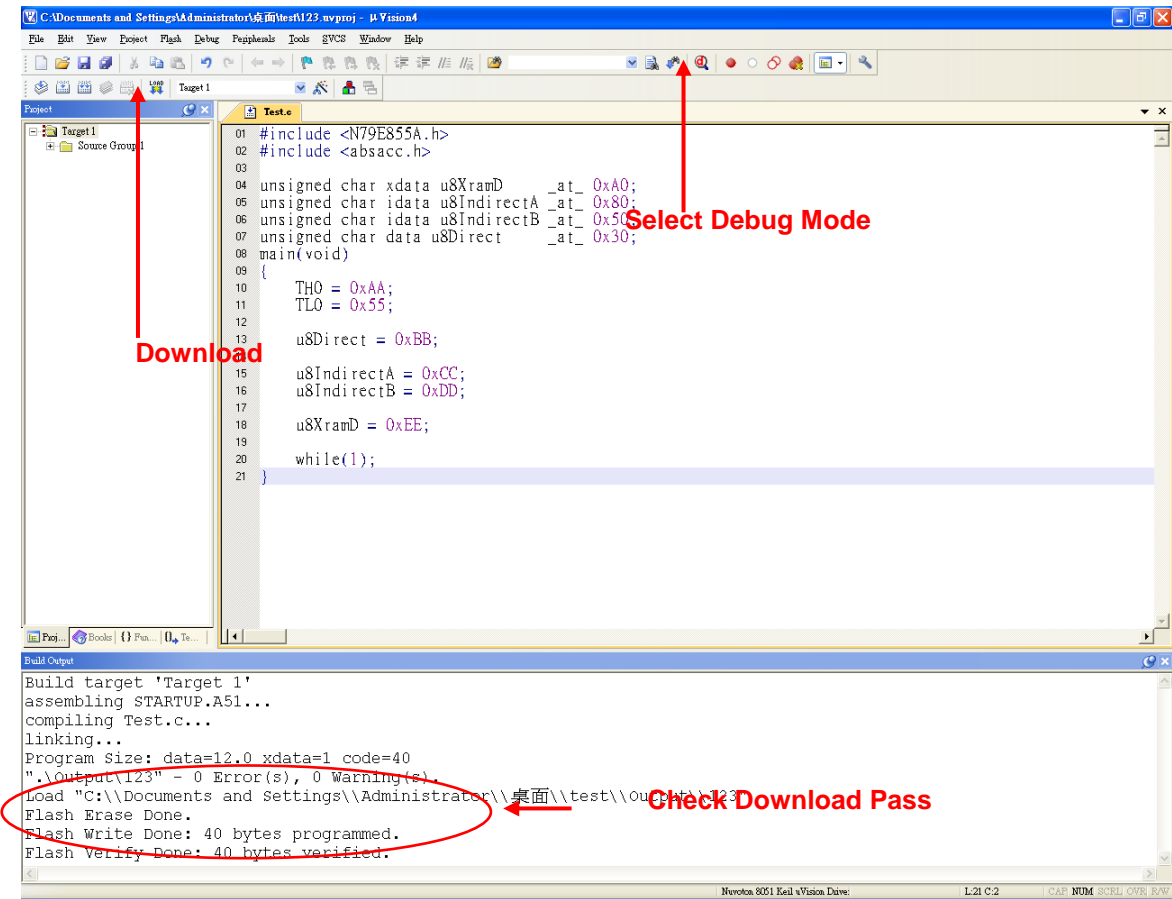


Figure 5-5 Download and Display Results

Step-6: Start debugging by clicking “Start/Stop Debug Session”.

Step-7: Select to free run, halt or run step by step. Breakpoints can be set or cleared, variables can be displayed and memory areas can be read, written or modified (refer to *Figure 5-6*).

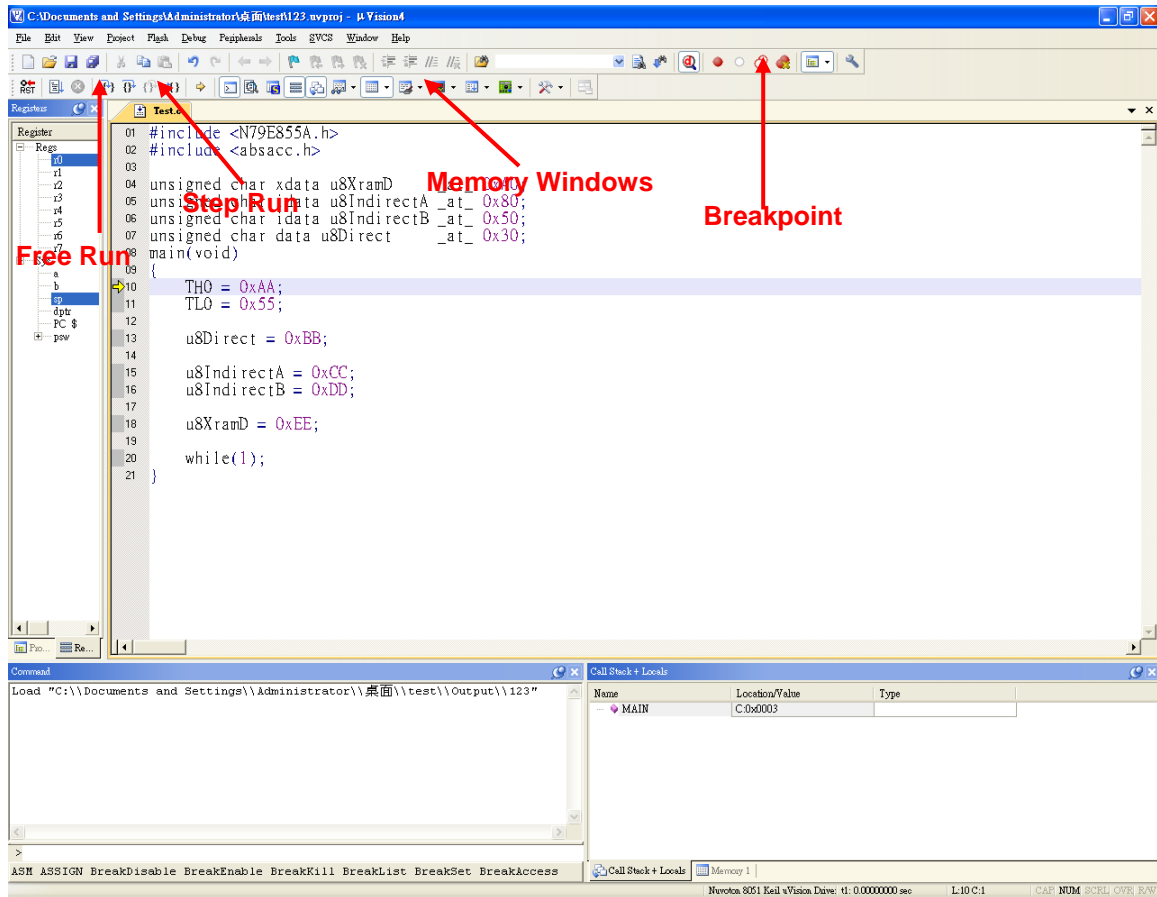


Figure 5-6 Execute Debug Mode

NOTE: Nuvoton ICE/OCD supports eight breakpoints. One breakpoint is reserved for the debug command, and the other seven breakpoints can be used.

5.2 Updating CONFIG

Step-1: Select the Debug tab in the “Options for Target” form.

Step-2: Click the “Settings” button.

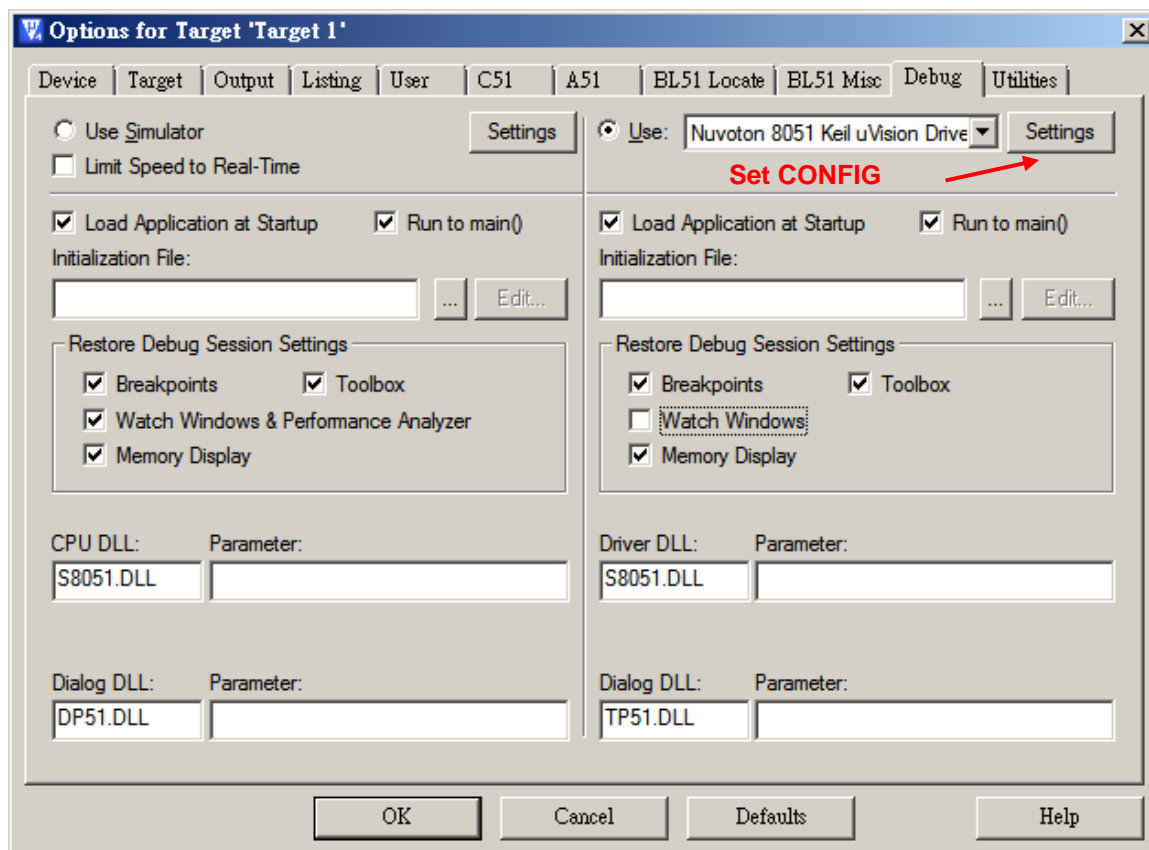


Figure 5-7 Set CONFIG Button

Step_3: Set CONFIG and click “Write Config”.

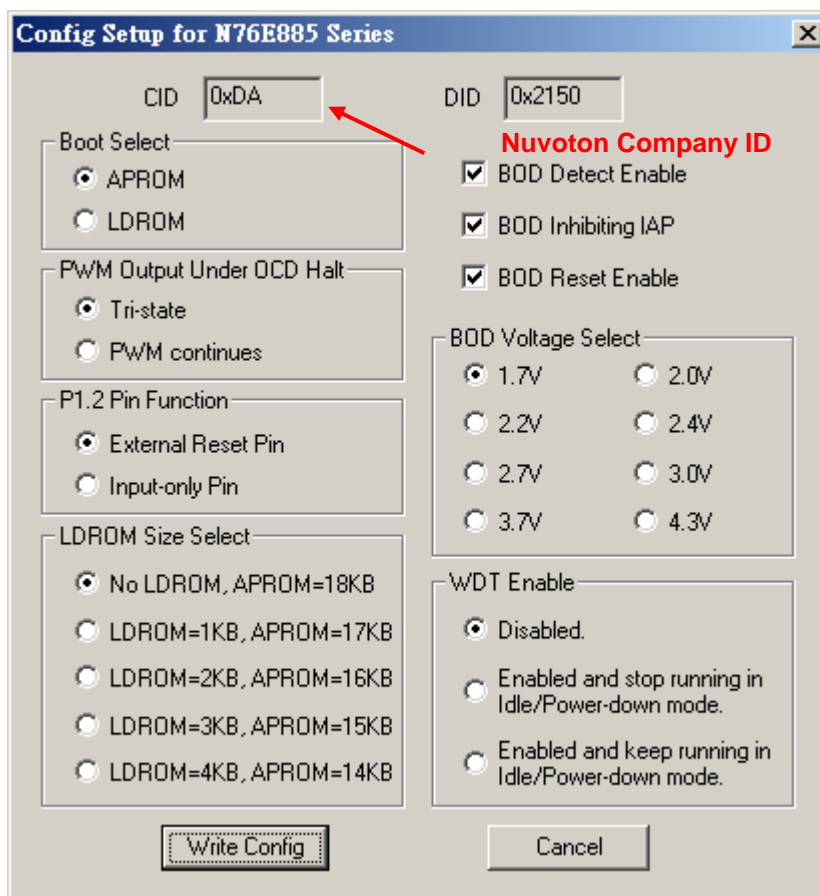


Figure 5-8 Set CONFIG Content

- NOTE:**
- a. Flash should be re-downloaded after clicking “Write Config”.
 - b. CID must be “0xDA” which is the “Company ID” (Nuvoton ID).
 - c. DID and CONFIG contents vary from different part numbers.

5.3 Read or Write SFR

Under debug mode, enter “D: 0x80” into the Address text field in the *Memory* window. “D: 0x80” means that the SFR address start 0x80 to 0xFF. SFRs is generally read or written.

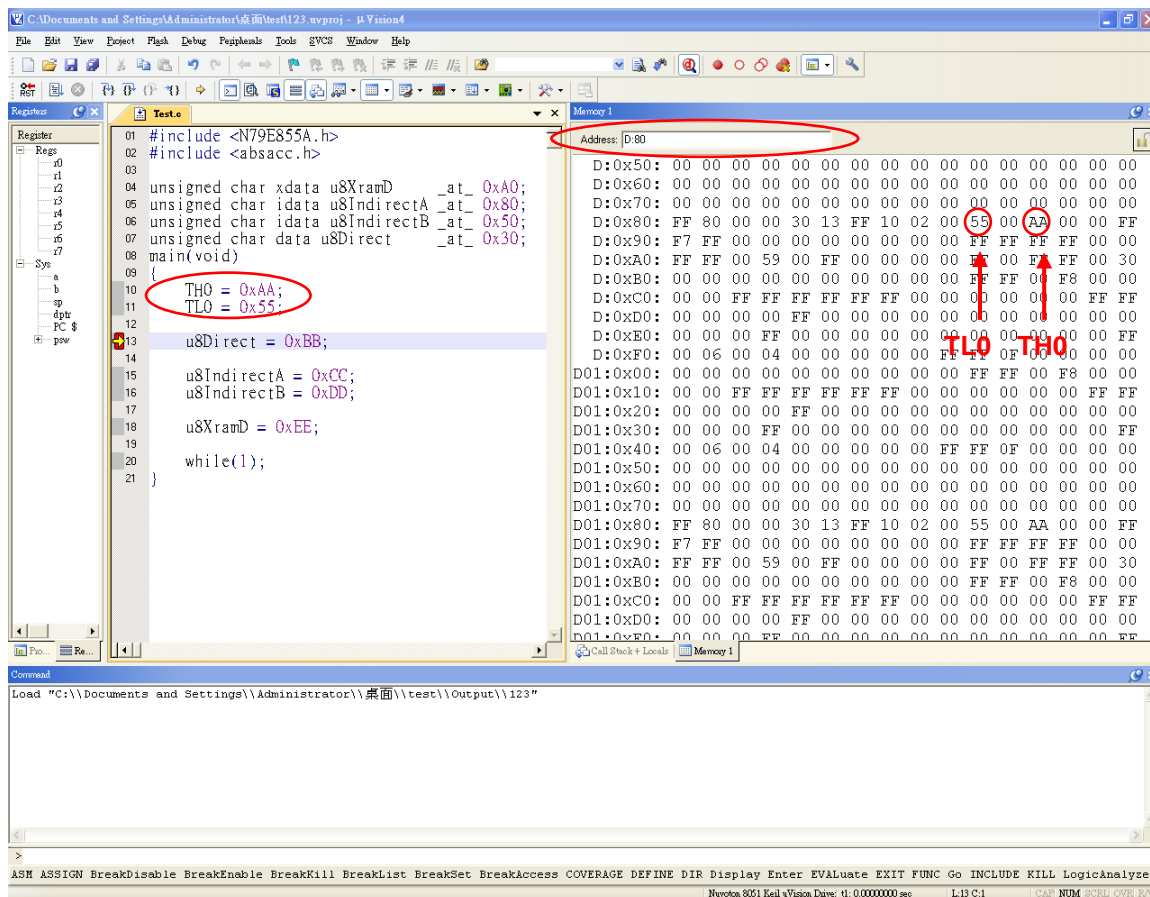


Figure 5-9 Read/Write SFRs

5.4 Read or Write Direct RAM

Under debug mode, enter “D: 0x00” into the Address text field in the *Memory* window. “D: 0x00” means that the direct RAM address start 0x00 to 0x7F. Direct RAM can be read or written.

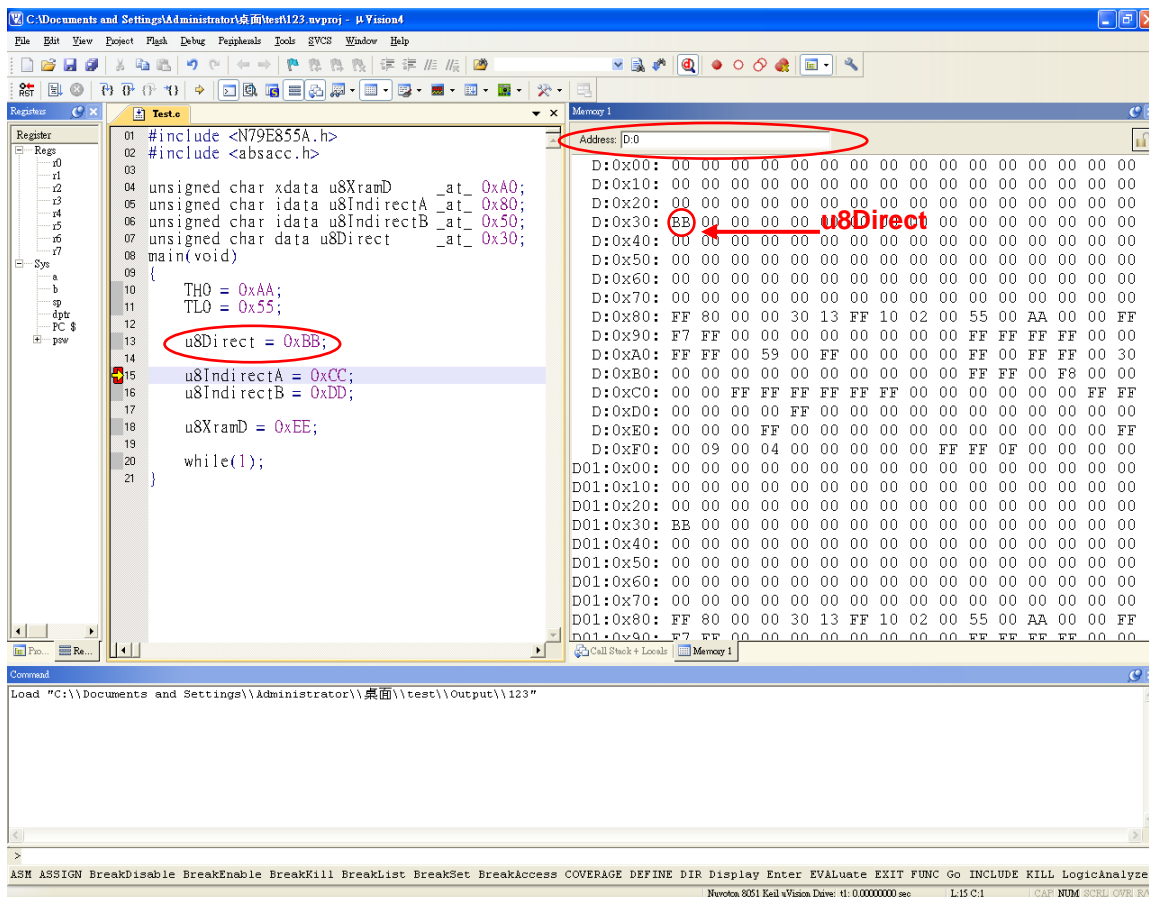


Figure 5-10 Read/Write Direct RAM

5.5 Read or Write Indirect RAM

Under debug mode, enter “I: 0x00” into the Address text field in the *Memory* window. “I: 0x00” means that the indirect RAM address start 0x00 to 0xFF. Indirect RAM can be read or written.

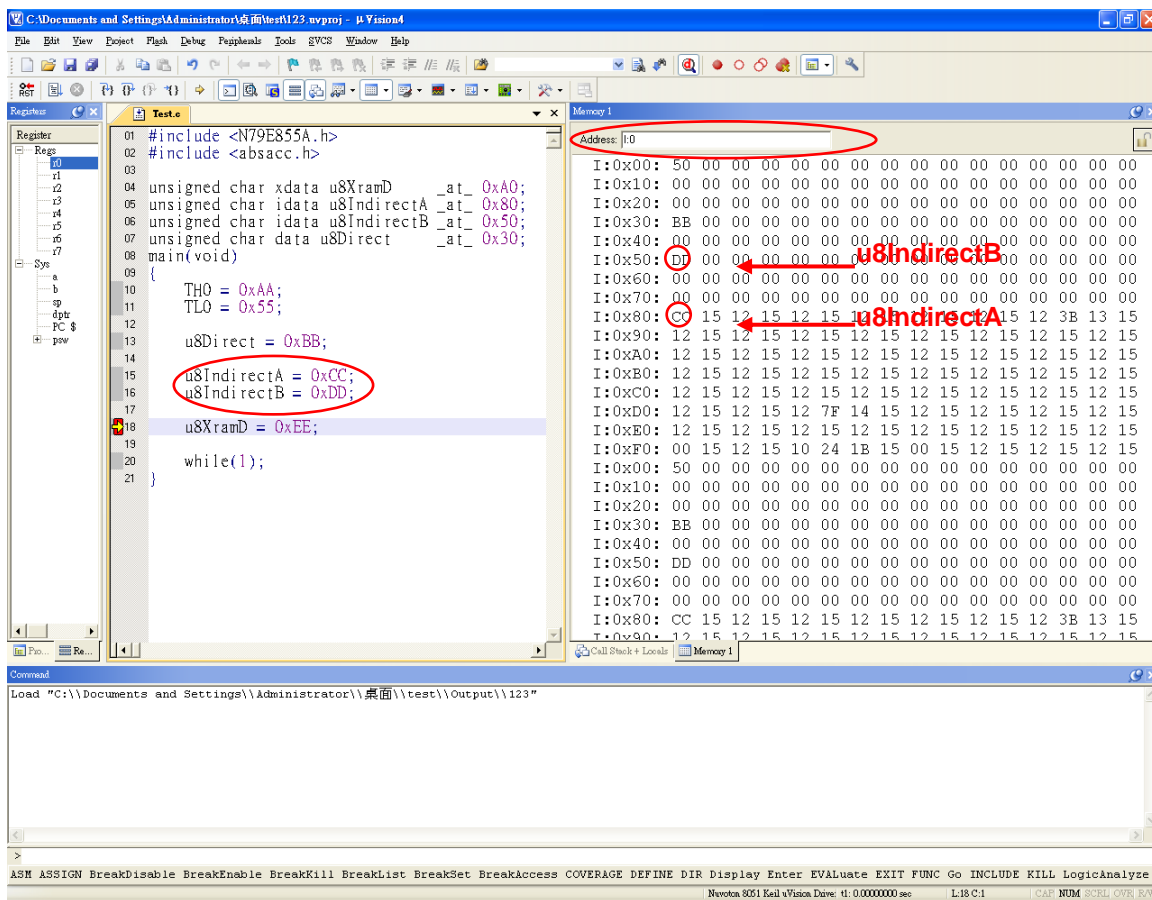


Figure 5-11 Read/Write Indirect RAM

5.6 Read or Write XRAM

Under debug mode, enter “X: 0x0000” into the Address text field in the *Memory* window. “X: 0x0000” means that the XRAM start address at 0x0000. XRAM can be read or written.

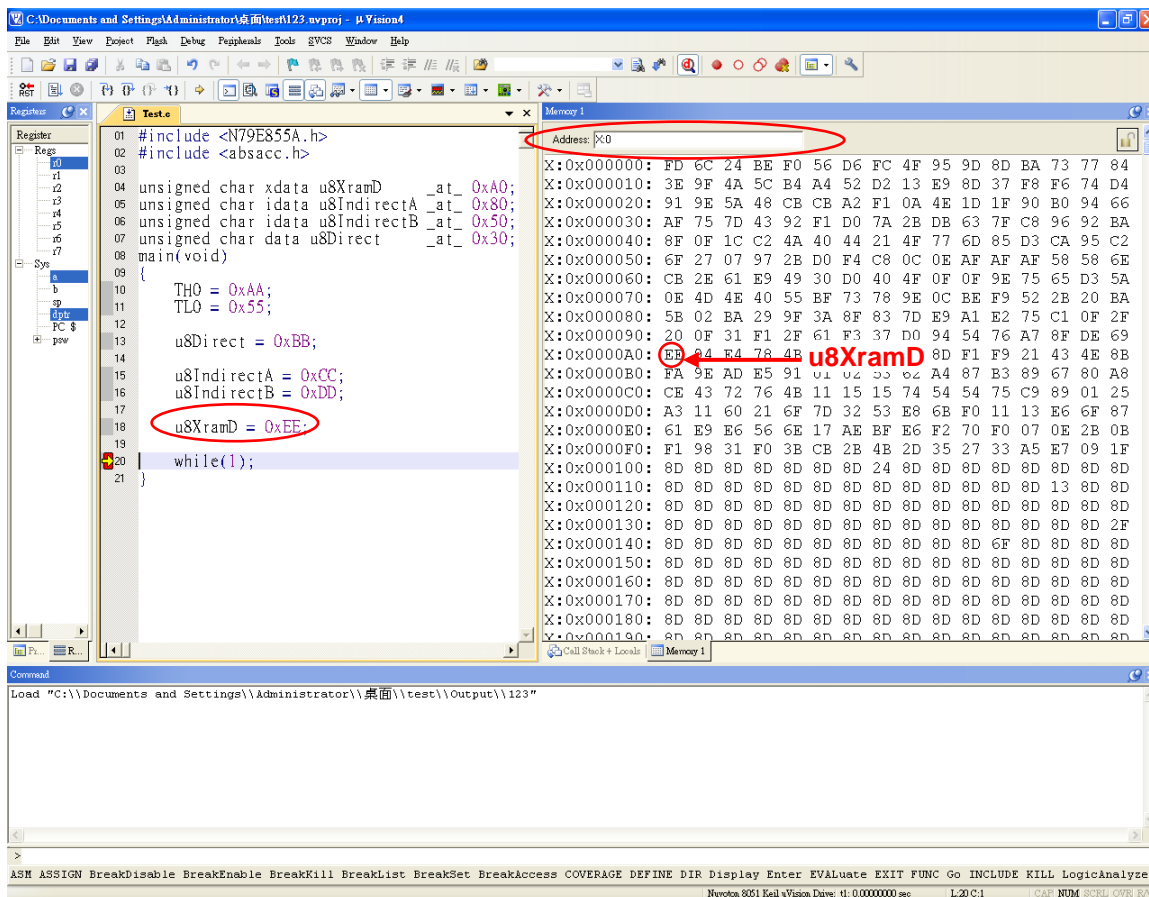


Figure 5-12 Read/Write XRAM

NOTE: If the MCU does not have XRAM, the setting is invalid.

5.7 Read Code (N76E003 / N76E885 / N76E616 Only)

Under debug mode, enter “C: 0x0000” into the Address text field in the *Memory* window. “C:0x0000” means that the CODE start address at 0x0000. CODE is read only. Reading CODE area benefits debugging especially when Data Flash storage is used during execution.

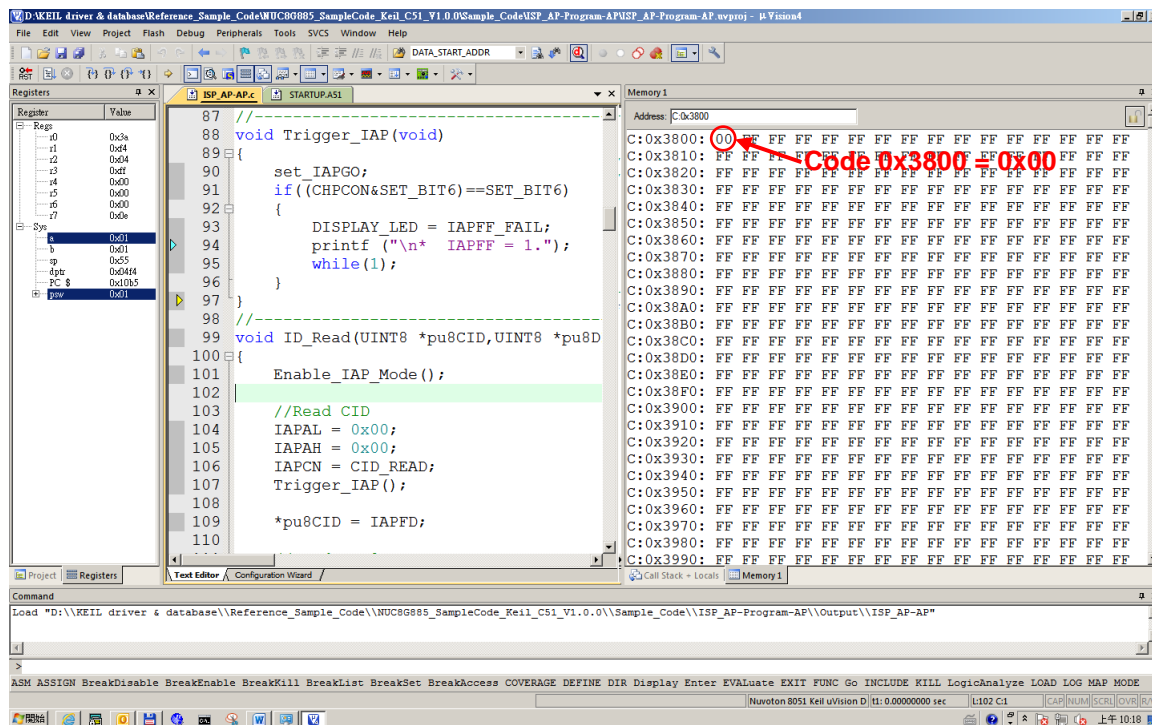


Figure 5-13 Read/Write Code

NOTE: This feature is valid only for N76E885 and N76E616.

6 PROJECT WORKING FLOW

The chapter provides flows for users to download or execute debug mode in KEIL™ C51 environment.

6.1 Program Code Download Flow

Step–1: Create a new project (refer to *Figure 4-1 ~ Figure 4-5*).

Step–2: Set debug options (refer to *Figure 4-6 ~ Figure 4-7*).

Step–3: Create a new C file and write sample code (refer to *Figure 5-1 ~ Figure 5-2*).

Step–4: Add C file in your project (refer to *Figure 5-3*).

Step–5: Set CONFIG (refer to *Figure 5-7 ~ Figure 5-8*).

Step–6: Build target and display results (refer to *Figure 5-4*).

Step–7: Download and display results (refer to *Figure 5-5*).

6.2 Program Code Debug Flow

Step–1: Create a new project (refer to *Figure 4-1 ~ Figure 4-5*).

Step–2: Set debug options (refer to *Figure 4-6 ~ Figure 4-7*).

Step–3: Create a new C file and write sample code (refer to *Figure 5-1 ~ Figure 5-2*).

Step–4: Add C file in your project (refer to *Figure 5-3*).

Step–5: Build target and display results (refer to *Figure 5-4*).

Step–6: Execute debug mode (refer to *Figure 5-5 ~ Figure 5-6*)

Step–7: Read or write SFRs, direct RAM, indirect RAM and XRAM (refer to *Figure 5-9 ~ Figure 5-12*).

7 CHECK VERSION

7.1 Nu-Link-Me Bridge Firmware Version

The ICE/OCD debug tool uses Nu-Link-Me as a bridge interface to communicate with KEIL™ debug environment and the target MCU. The main chip of Nu-Link-Me is a Nuvoton NUC12SR MCU. The firmware version programmed in this chip can be checked in “Flash Download Setup” window by clicking “Settings” in “Utilities” page.

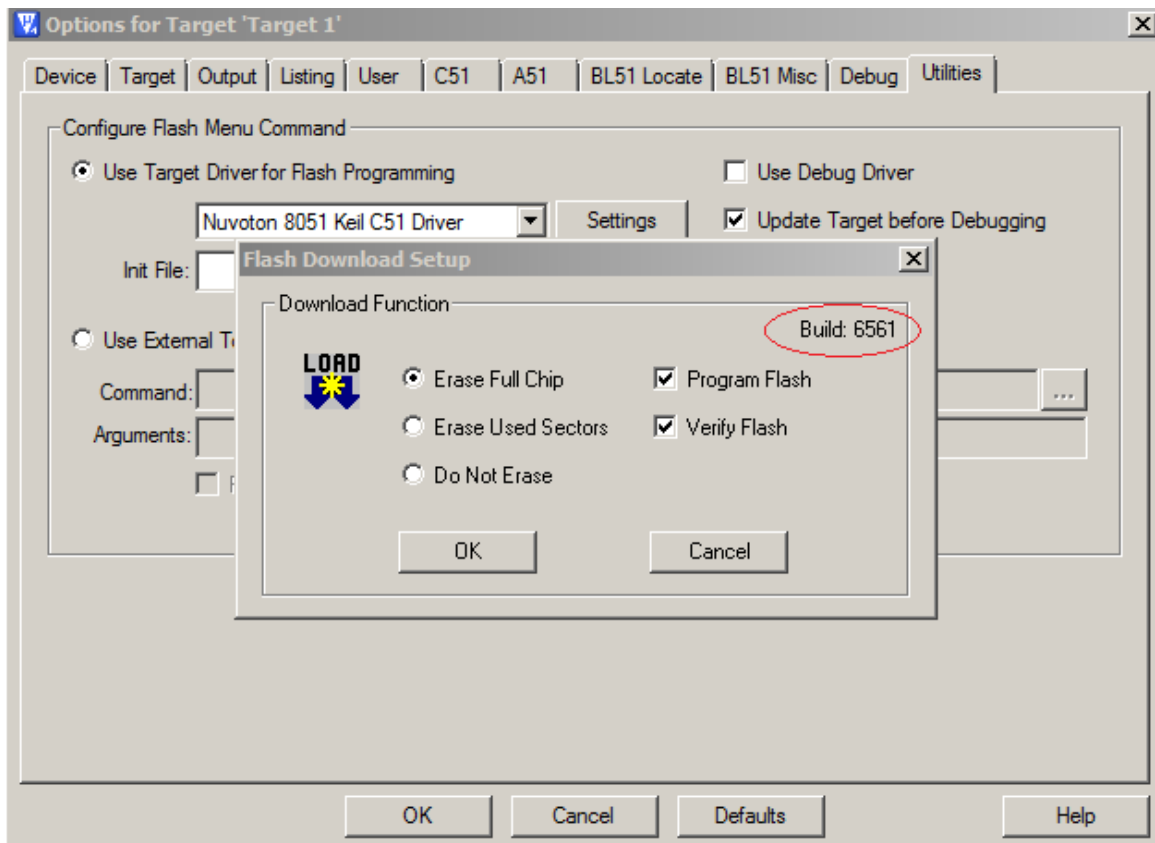


Figure 7-1 Check Bridge Firmware Version

7.2 Appendix Limitations of KEIL™ C51 Evaluation Edition

KEIL™ development tools without a current product license run as an Evaluation edition and have the following restrictions:

- The 8051 compiler, assembler, linker, and debugger are limited to 2 Kbytes of object code. Source code may be of any size.
- Programs that generate more than 2 Kbytes of object code will not compile, assemble, or link.
- The debugger supports programs that are 2 Kbytes or smaller.
- The startup code generated includes LJMPs. Code generated cannot be used in single-chip devices that support 2 Kbytes or less of program space.
- Programs start at offset 0x0800. Programs generated with the evaluation software may not be programmed into single-chip devices with less than 2 Kbytes of on-chip ROM.
- No hardware support for multiple DPTR registers is provided.
- No support for floating-point arithmetic and no support for user libraries are provided.
- No support for in-line assembly using #pragma ASM.
- The following components which are present in the PK51 Full Version are not included in the Evaluation Version: Linker for Code Banking, Library Manager, and RTX51 Tiny Real-time Operating System.

8 REVISION HISTORY

Date	Revision	Description
1.00	Nov 09,2011	Initial release.
1.01	Nov 17,2011	1. Revised W79E82J CONFIG. 2. Added the free-run button.
1.02	Nov 21,2011	Revised W79E85J CONFIG.
1.03	Mar 05,2012	Supported W79E659.
1.04	Apr 11,2012	1. Supported N79E37J. 2. Added Project Working Flow. 3. Updated format and descriptions.
1.05	Dec 8, 2014	1. Supported NUC8G885. 2. Supported Nuvoton's own CPU database and header files. 3. Supported CODE area read for NUC8G885. 4. Added how to check version of the debug environment.
1.06	Jan 14, 2015	Renamed NUC8G885 as N76E885.
1.07	Apr 15, 2015	1. Minor modified for N76E885 flash programming. 2. Supported N76E616.
1.08	Oct 08, 2015	Fixed a compile issue for KEIL™ μVision®5.
2.00	Nov 15, 2016	1. Add N76E003 supporting 2. Modify Nu-link-Me as bridge support

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.