NuMicro® 8051
8-bit Microcontroller

8051 Family
KEIL™ µVision® Driver
User Manual

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

www.nuvoton.com

Table of Contents

1 OVERVIEW

The Nuvoton 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 KEIL™ µVision® Driver after Rev. 2.06, 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 ML51, MS51, N76E003, N76E885 and N76E616, the corresponding header file is also installed in KEIL™ general library. User can include it directly by writing e.g. “#include < ML51.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 interface.

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

NuMicro 8051 Low power Product Line	
ML51 series	ML51BB9AE, ML51DB9AE, ML51FB9AE, ML51OB9AE, ML51XB9AE, ML51EB9AE, ML51UB9AE, ML51PB9AE, ML51TB9AE, ML51EC0AE, ML51UC0AE, ML51PC0AE, ML51TC0AE
LPC 1T 8051 Product Line	
MS51 series	MS51FB9AE, MS51XB9AE, MS51XB9BE, MS51OB9AE
N76E003 series	N76E003AT20, N76E003AS20, N76E003AQ20, N76E003BQ20, N76E013CQ20
N76E616 series	N76E616AL48, N76E616AF44, N76E616AM44
N76E885 series	N76E885AT28, N76E885AT20, N76E885AQ20, N76E885AS28
LPC 4T 8051 Product Line (Nu-Link-51 Only)	
N79E85J	N79E715AT28, N79E715AT20, N79E715AS28, N79E715AS20, N79E715AS16, N79E855AWG, N79E854AWG, N79E845AWG, N79E844AWG, N79E8432ASG, N79E815AT28, N79E815AT20, N79E814AT28, N79E814AT20 , N79E8132AS16

Table 2-1 Supported Chips

3 INSTALL NUVOTON 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 ***Nu-Link_Keil_Driver.exe*** and then select the <*KEIL install path*>. The Hnew files will appear inside <*KEIL install path*>*\C51\BIN\ Nuvoton_8051_KeiL_uVision_Driver.DLL*.
- The latest Nu-Link Keil driver is available on the following web page:

  http://www.nuvoton.com/hq/support/tool-and-software/software/?__locale=en

4 START TO WORK WITH µVISION® 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 µVision Project**" to open the *Create New Project* form.

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

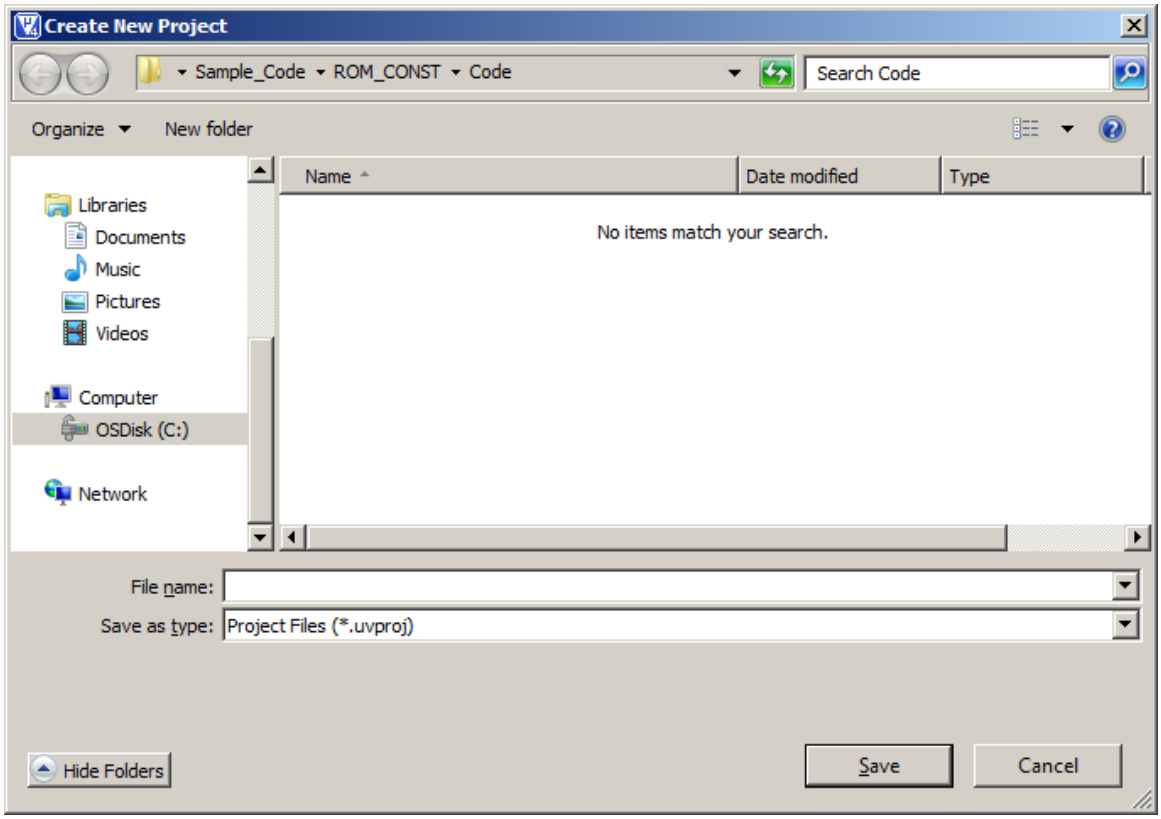


Figure 4-1 Create a New Project

Step–3: Select "**Nuvoton 8051 Devices**" and click the "**OK**" button.

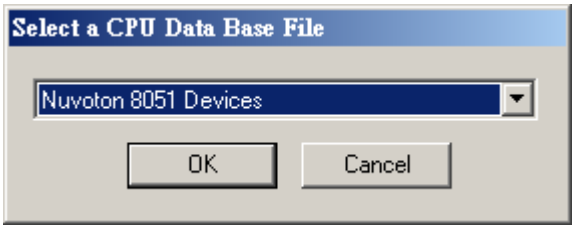


Figure 4-2 Select a CPU Database

Step–4: Select your device.

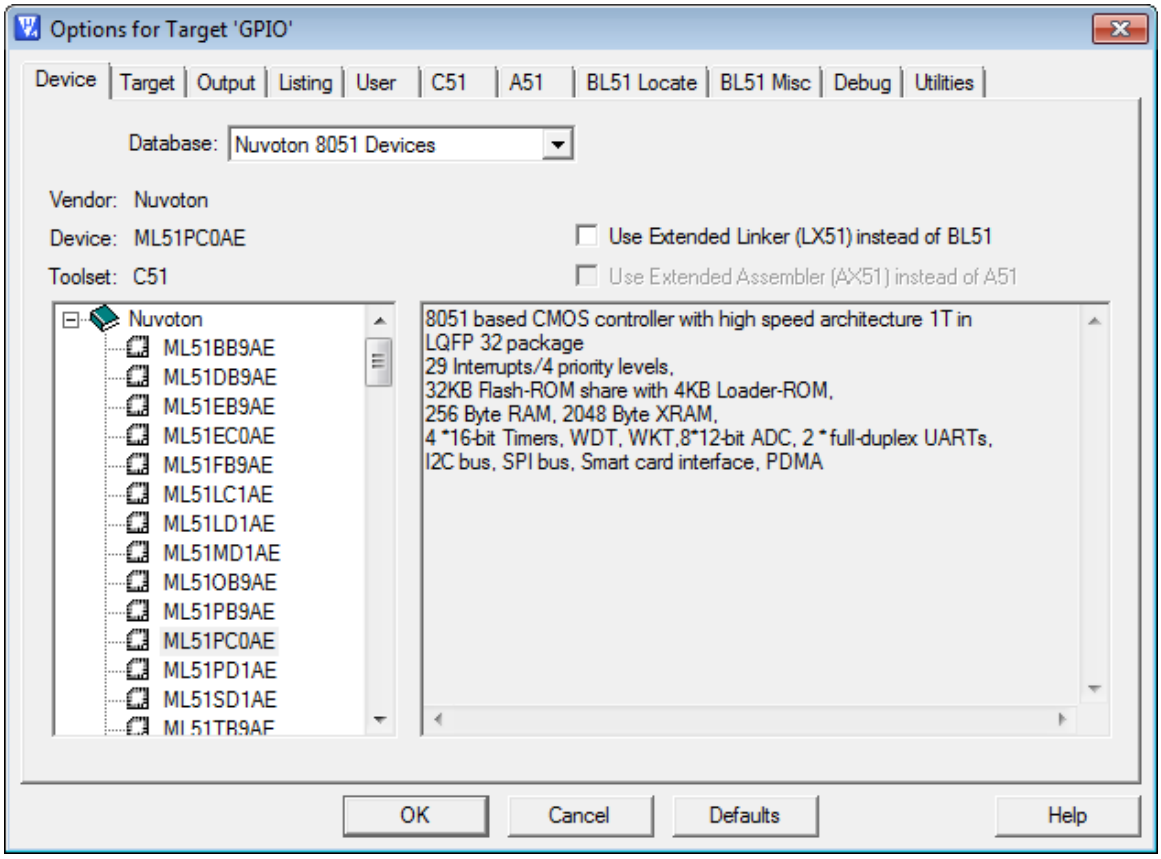


Figure 4-3 Select Your Device from the Database List

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

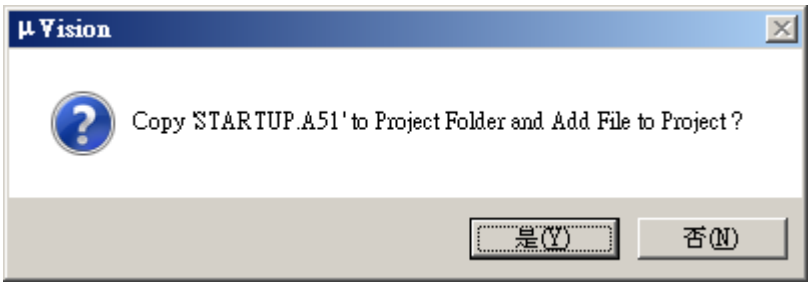


Figure 4-4 Add "STARTUP.A51" to Project

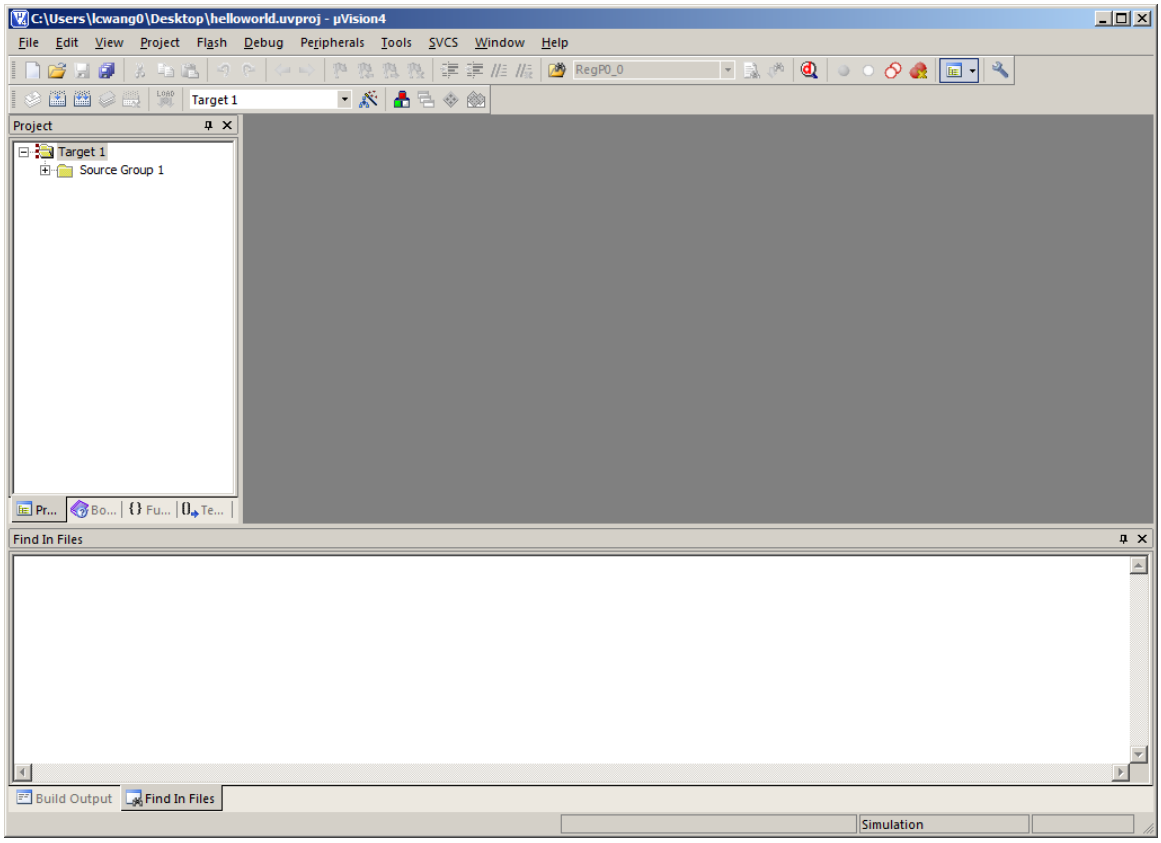


Figure 4-5 Complete the Process to Create a New Project

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 µVision Driver**".

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

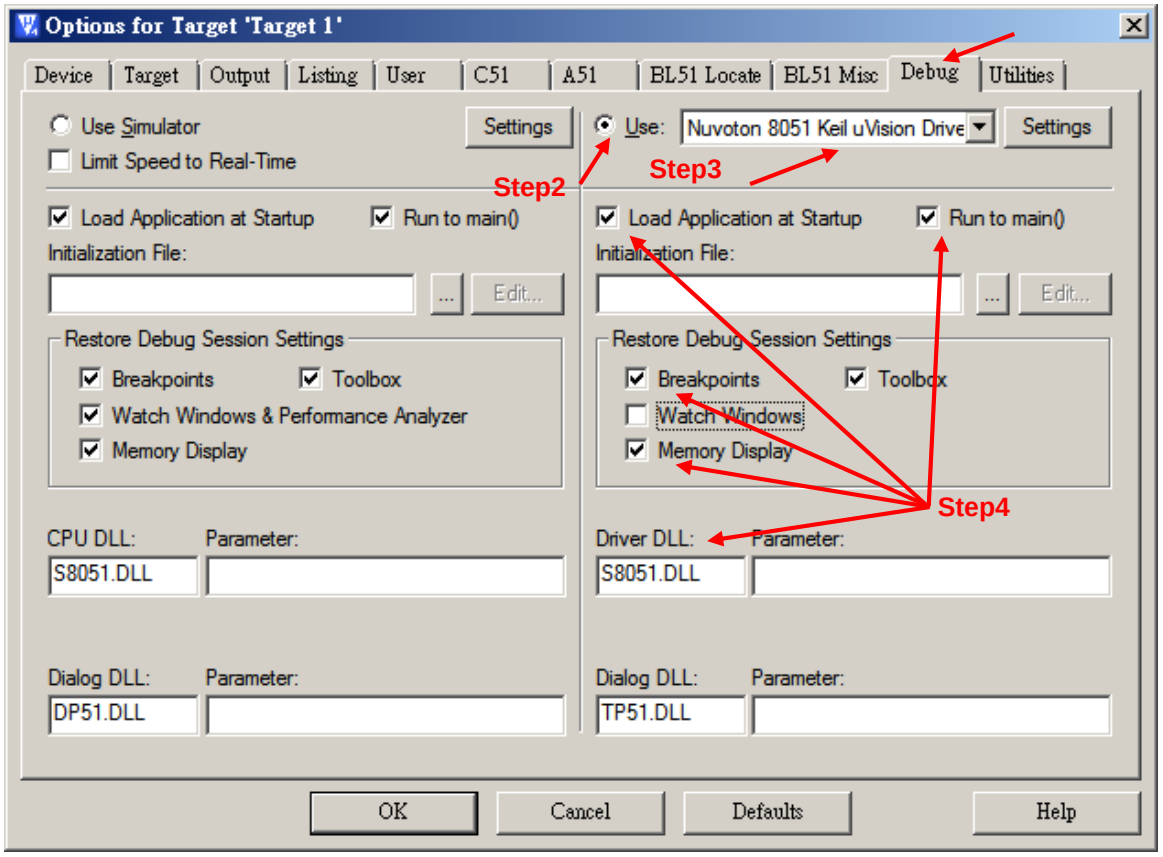


Figure 4-6 Set Debug Optins

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.

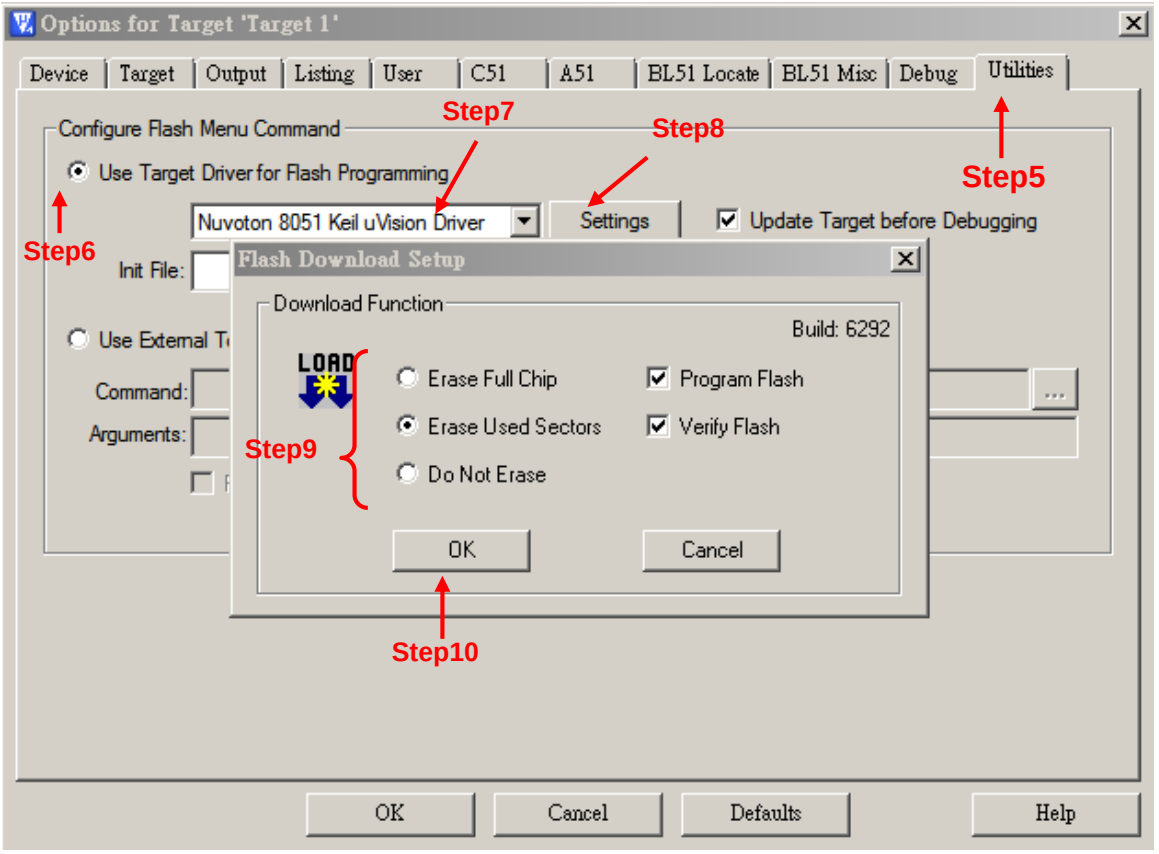


Figure 4-7 Set Download Options

5 USE THE DEBUG FEATURES

This chapter describes how to use Nuvoton 8051 KEIL™ µVision® 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.

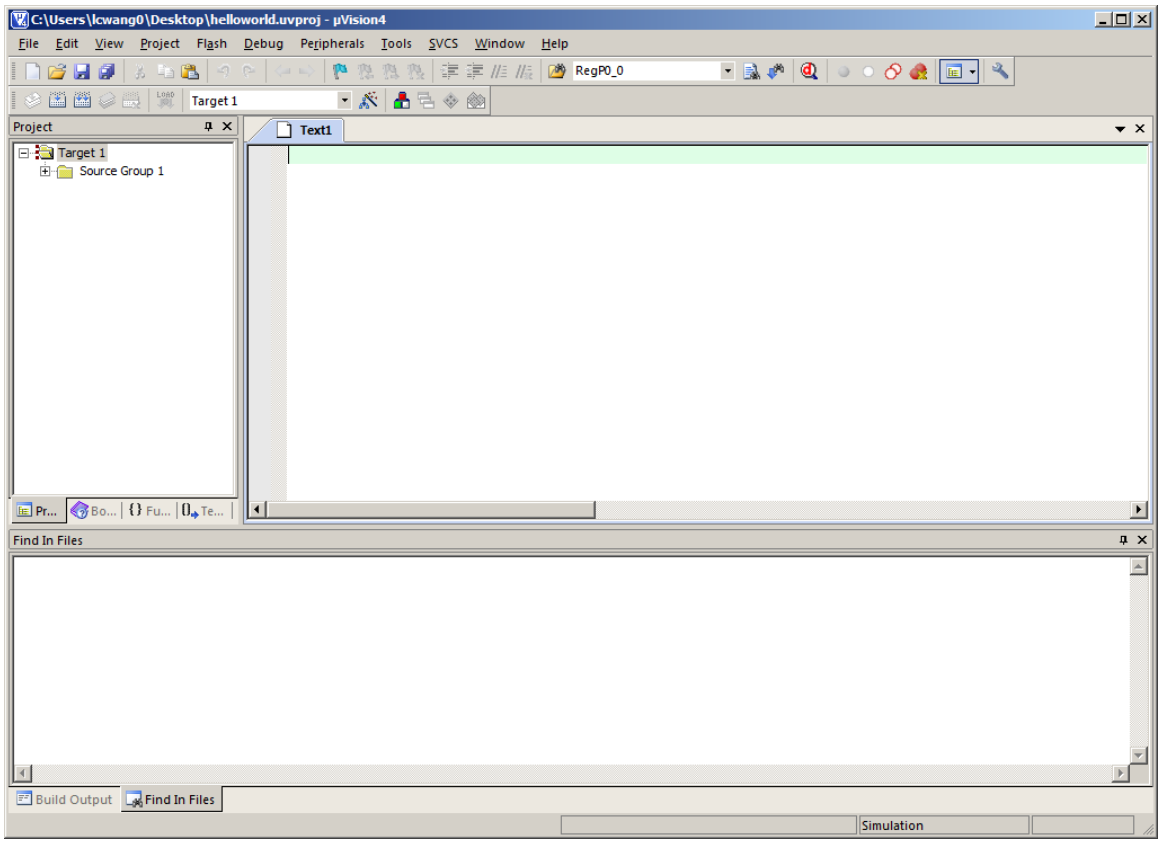


Figure 5-1 Create a New C File

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

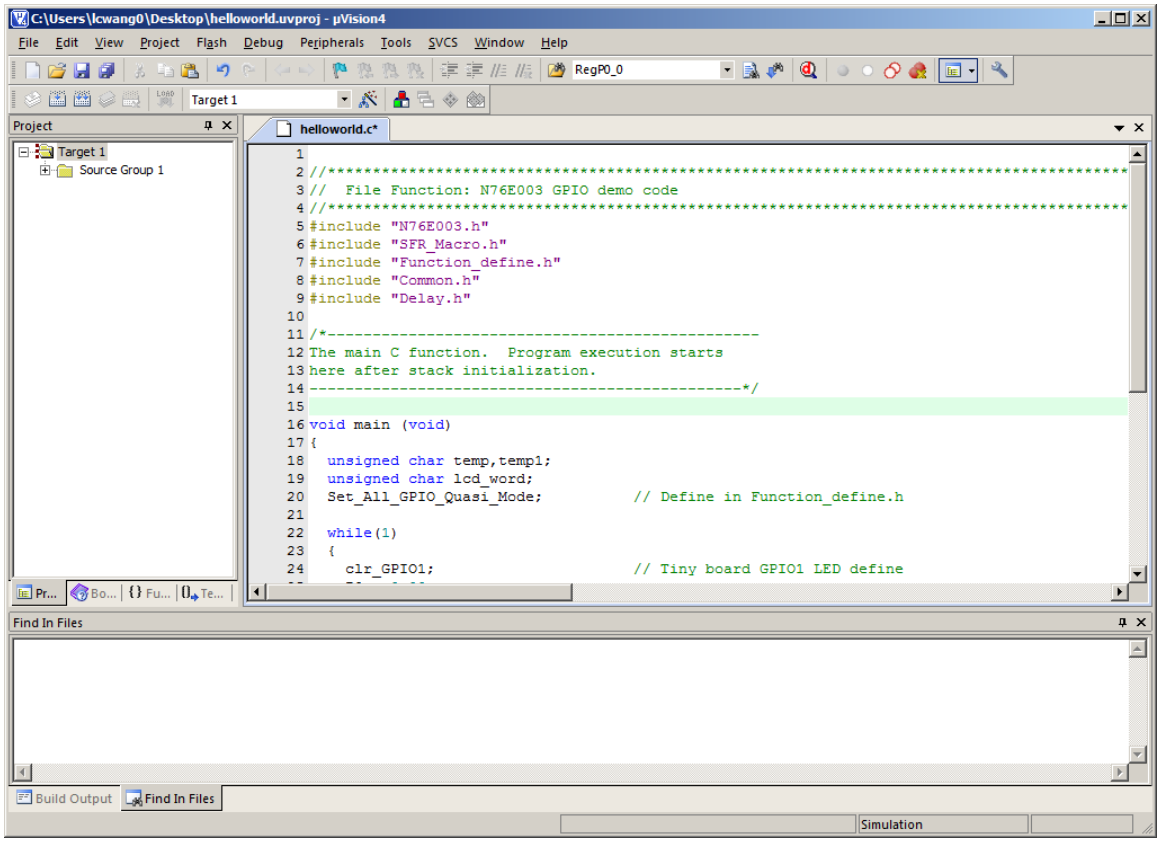


Figure 5-2 Write Sample Code

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

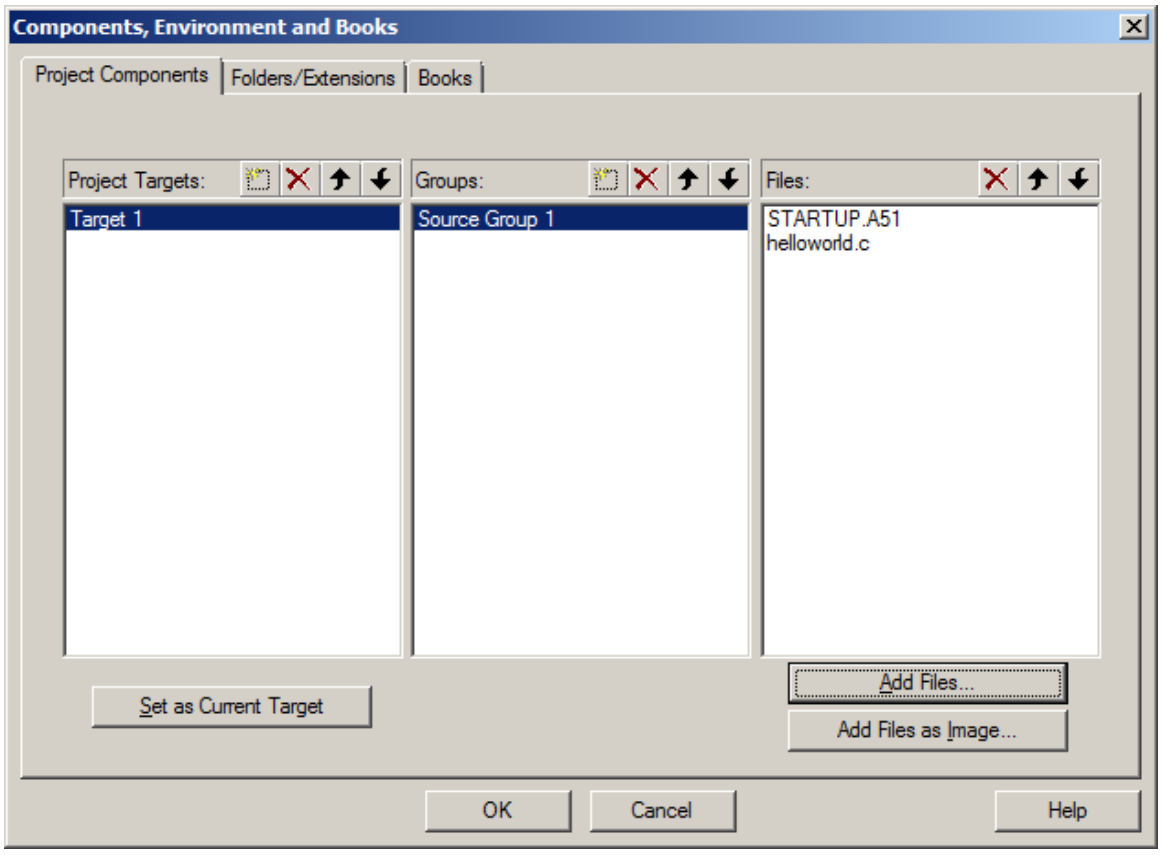


Figure 5-3 Add the C File in Your Project

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

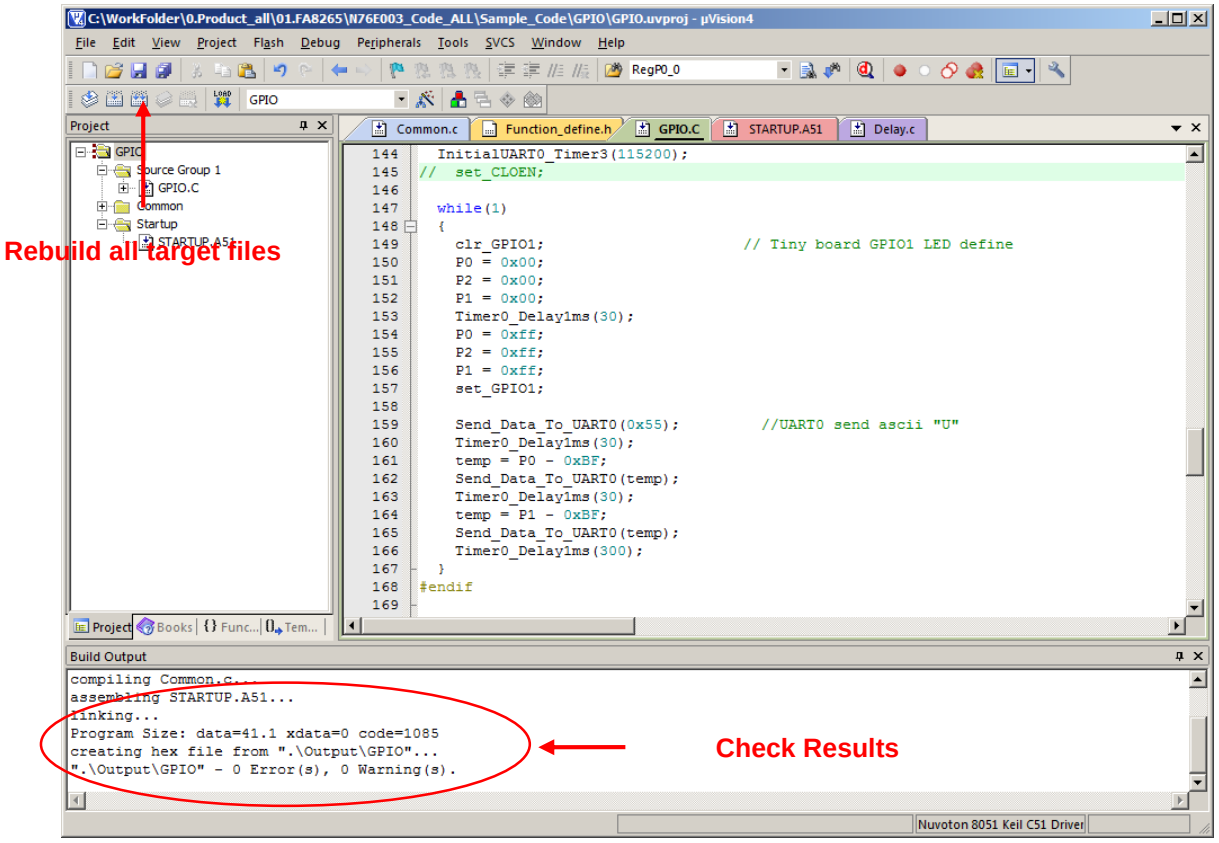


Figure 5-4 Build Target and Display Results

Step–5: Click "**Download**" to download the code to flash memory.

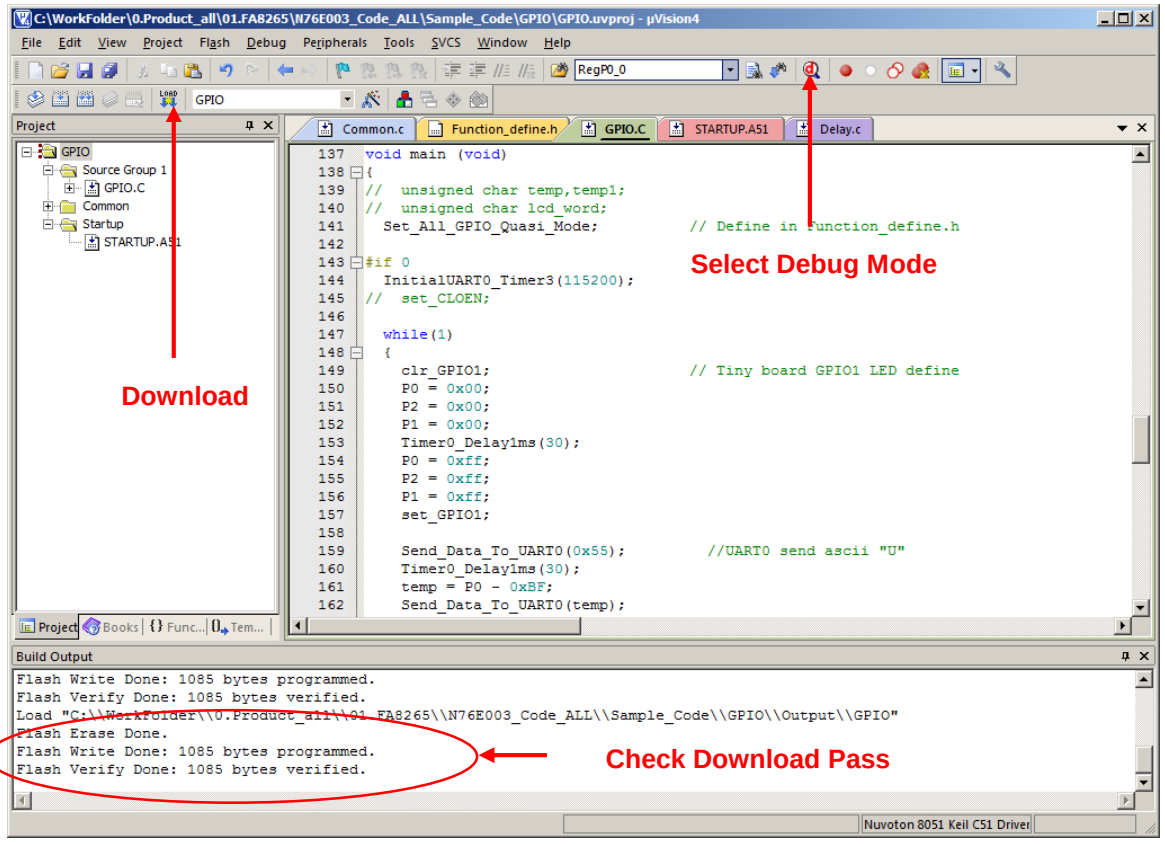


Figure 5-5 Download and Display Results

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*).

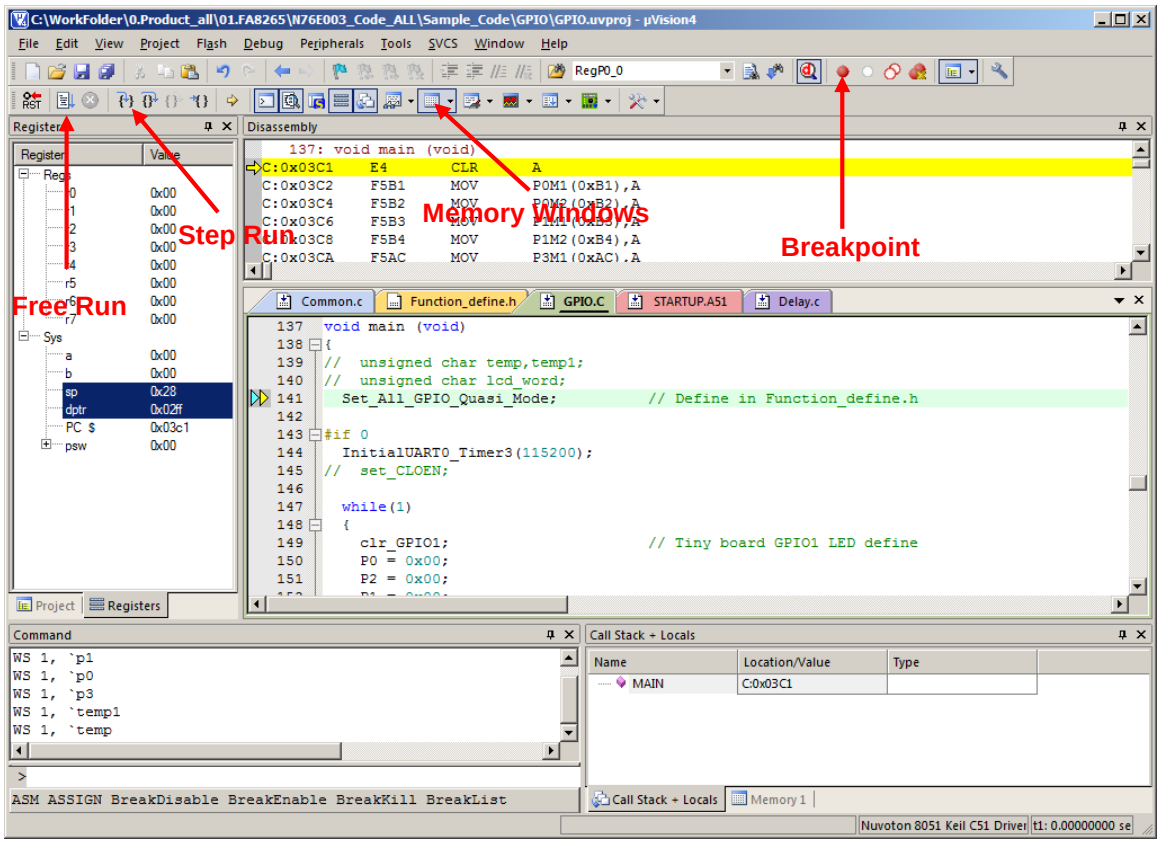


Figure 5-6 Execute Debug Mode

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.

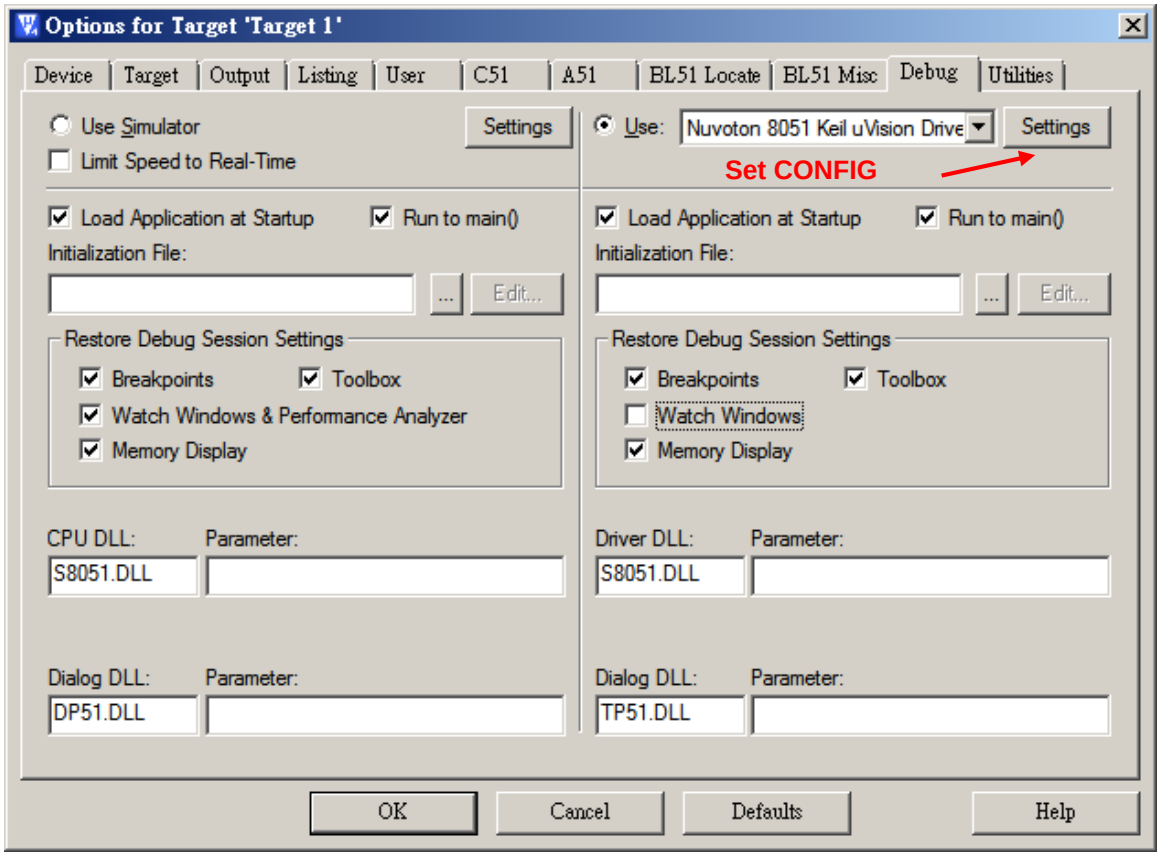


Figure 5-7 Set CONFIG Button

Step_3: Set *CONFIG* and click "**Write Config**".

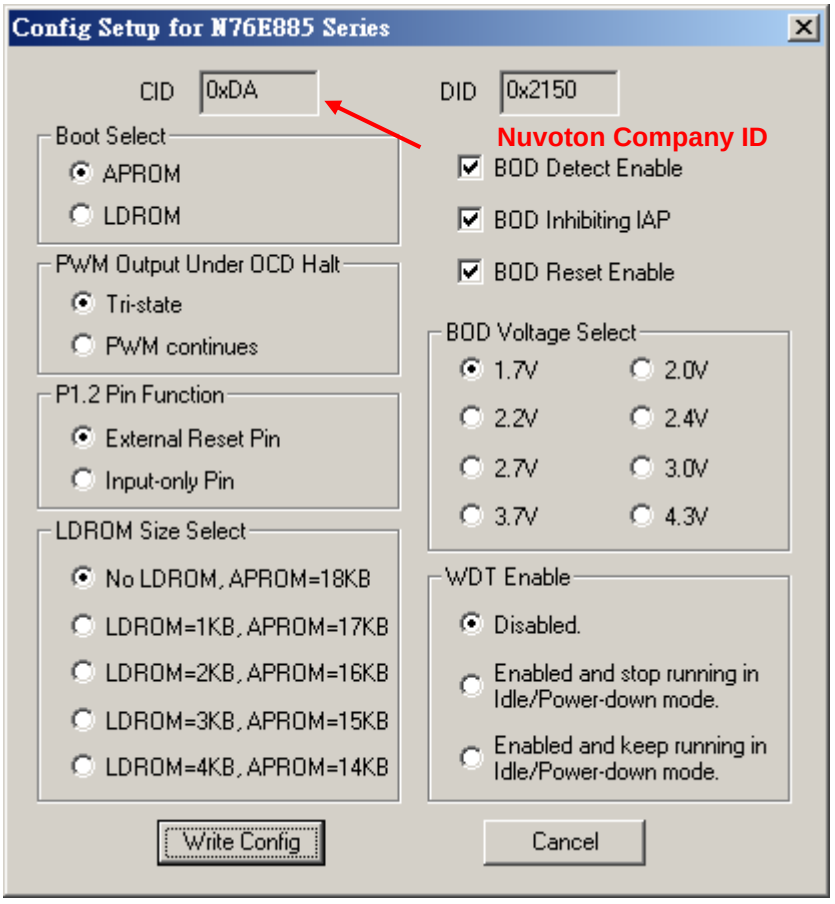


Figure 5-8 Set CONFIG Content

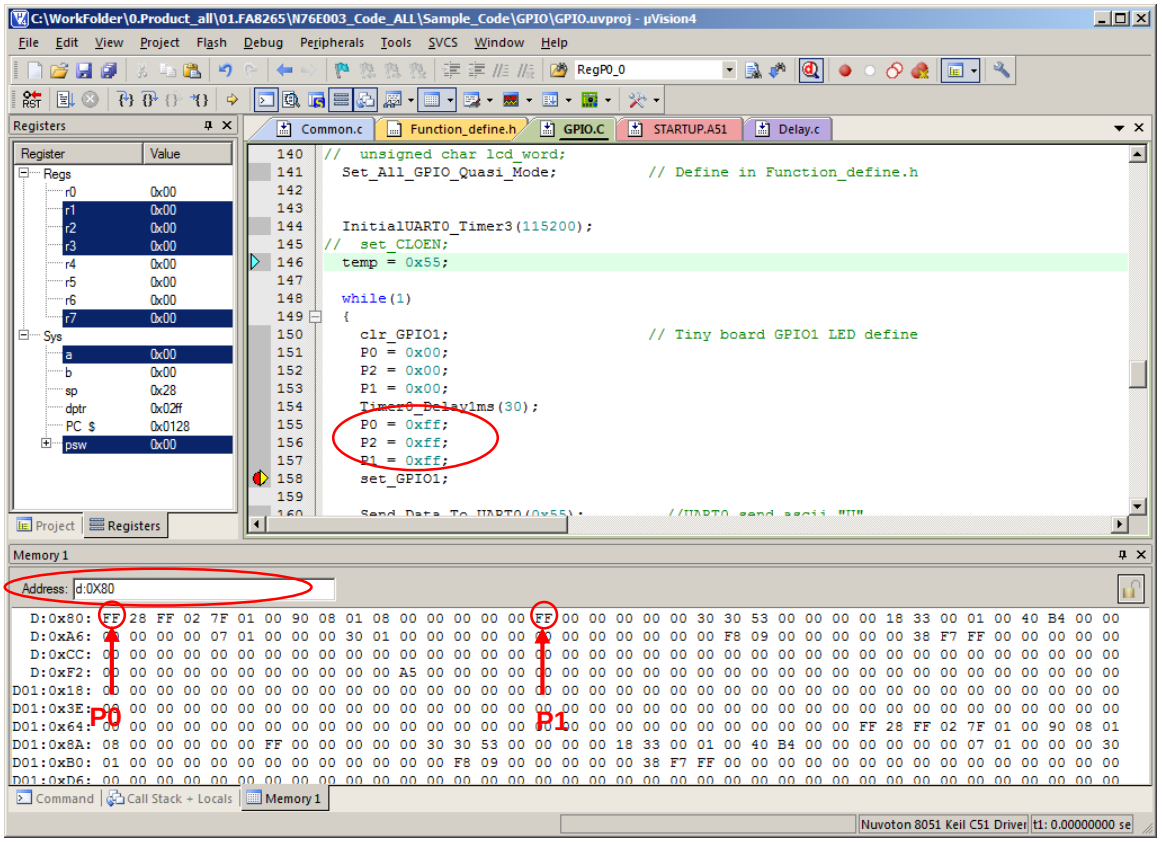
NOTE: **a.** Flash should be re-downloaded after clicking "**Write Config**".

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.



Z

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.

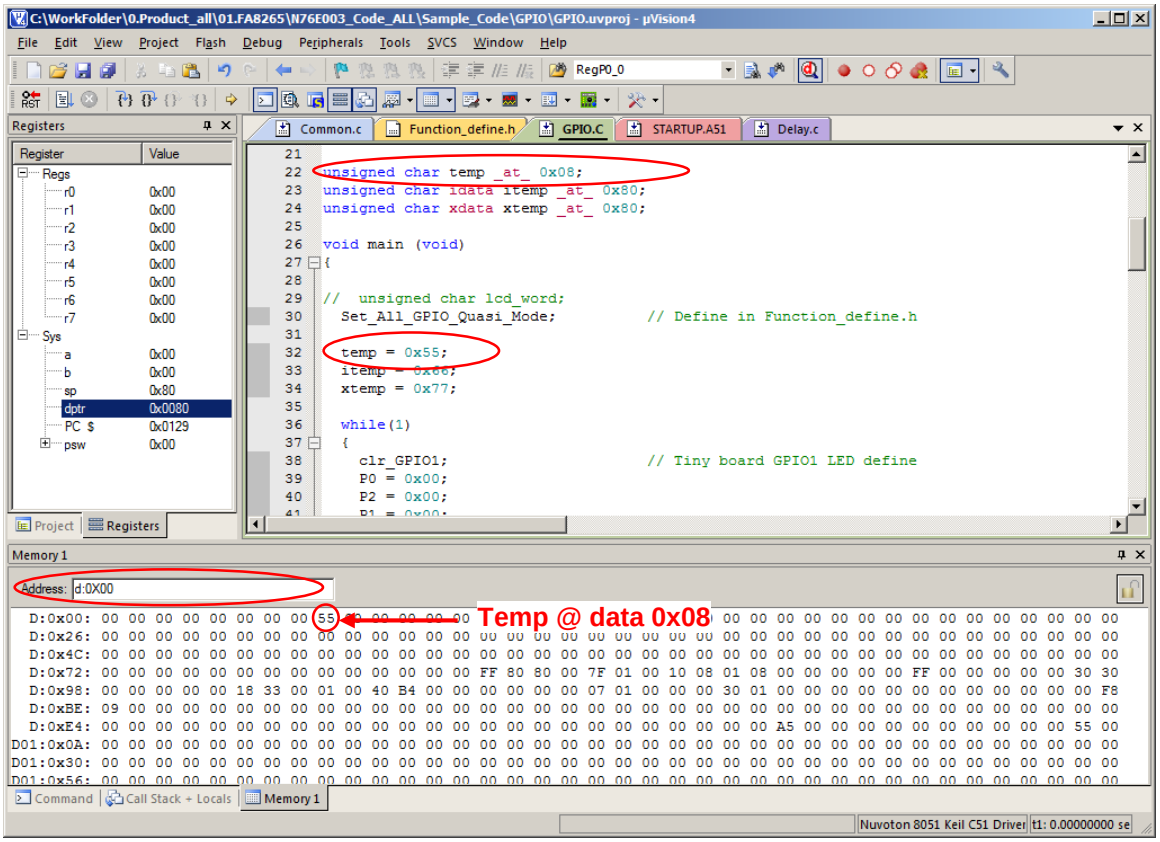


Figure 5-10 Read/Write Direct RAM

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.

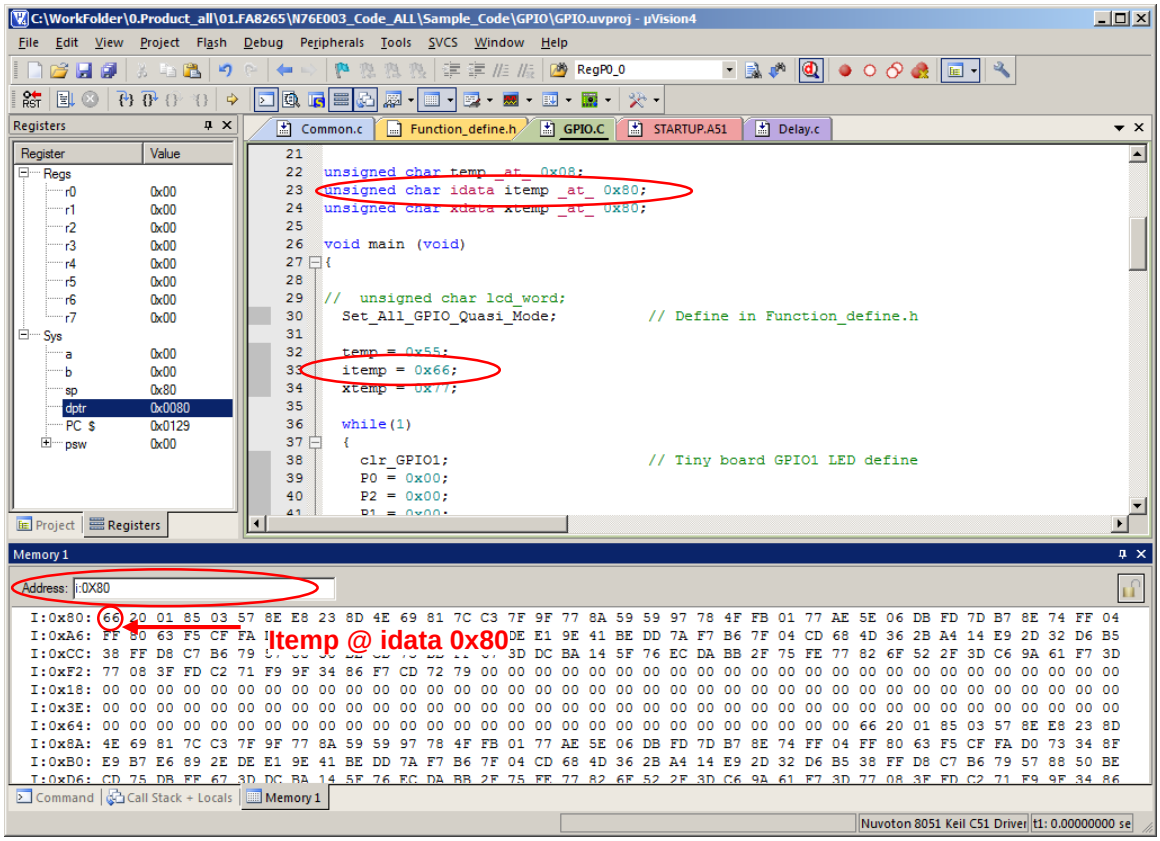


Figure 5-11 Read/Write Indirect RAM

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.

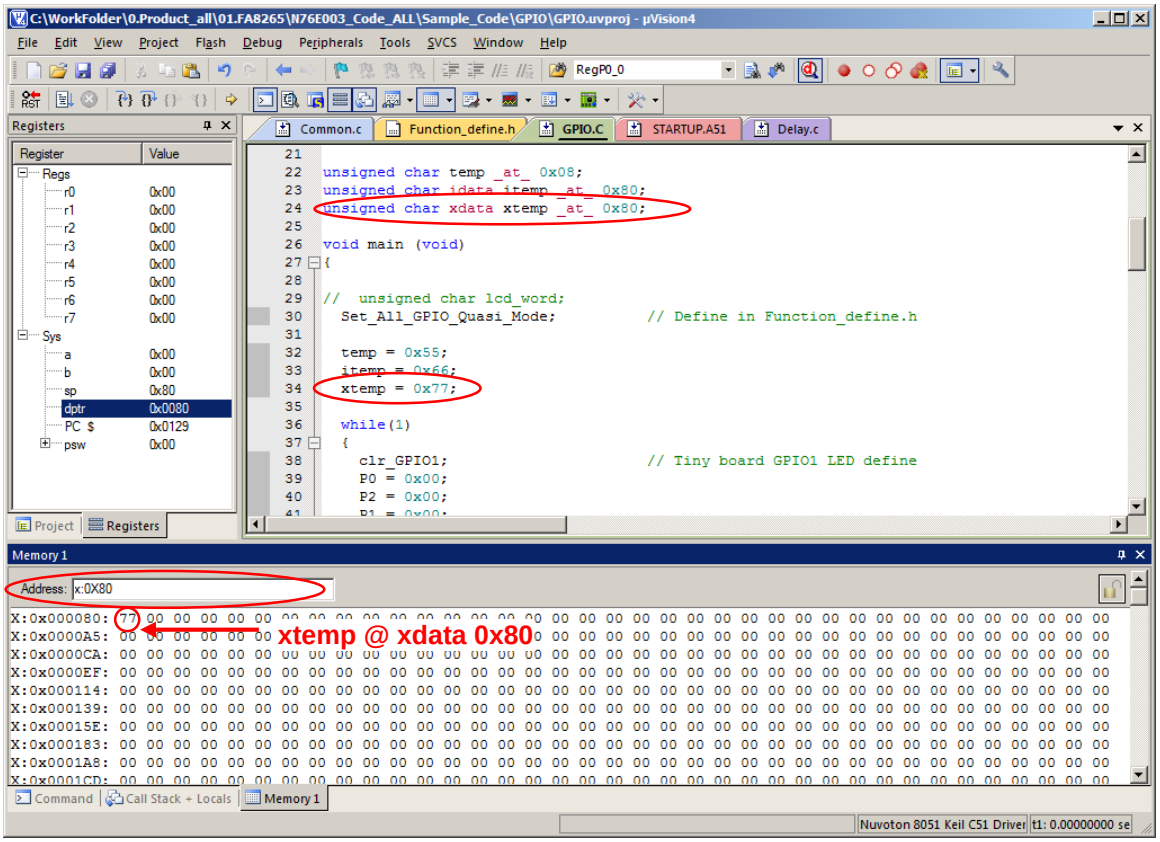


Figure 5-12 Read/Write XRAM

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

5.7 Read Code

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.

NOTE: This feature is valid for ML51 series, MS51 series and N76 series.

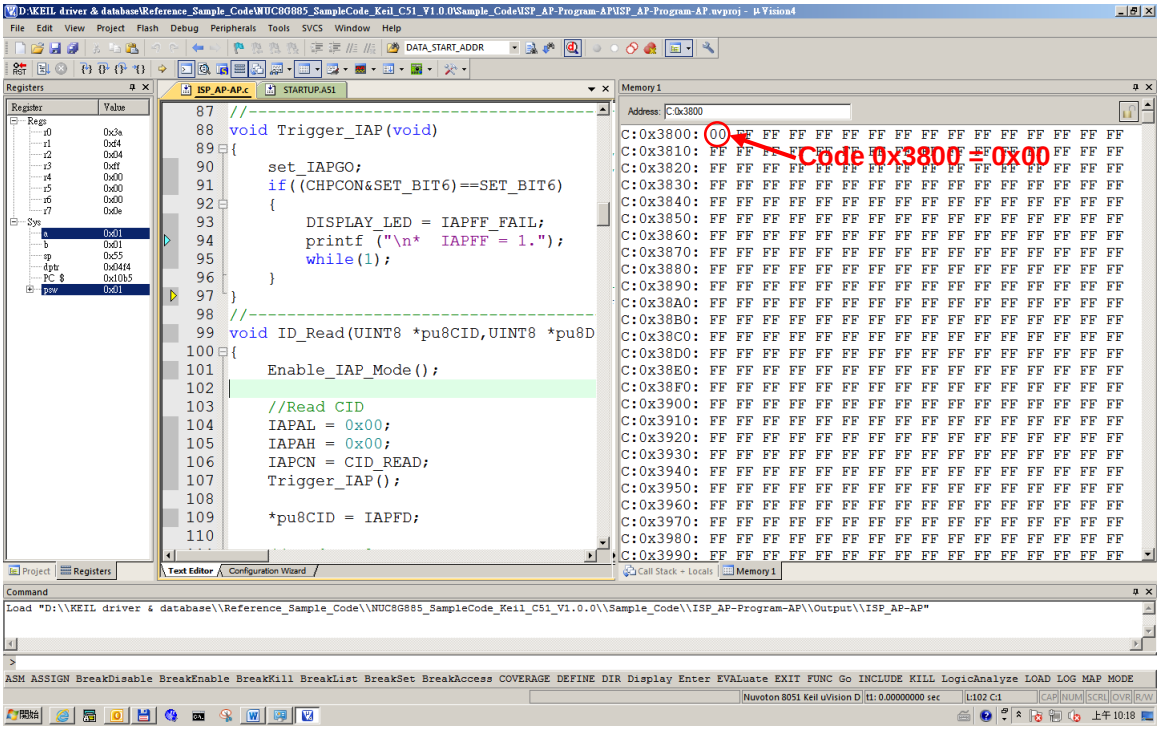
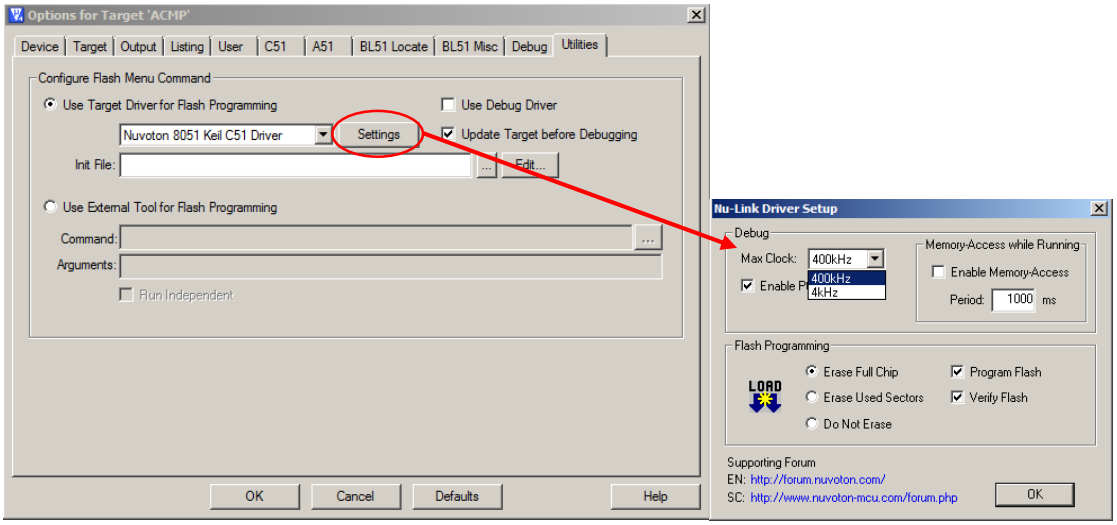


Figure 5-13 Read/Write Code

5.8 ICE Clock Setting

If target MCU use LIRC or system clock divider under 1MHz, need define the ICE max Clock to 4kHz to confirm the ICE read / write data correctly.

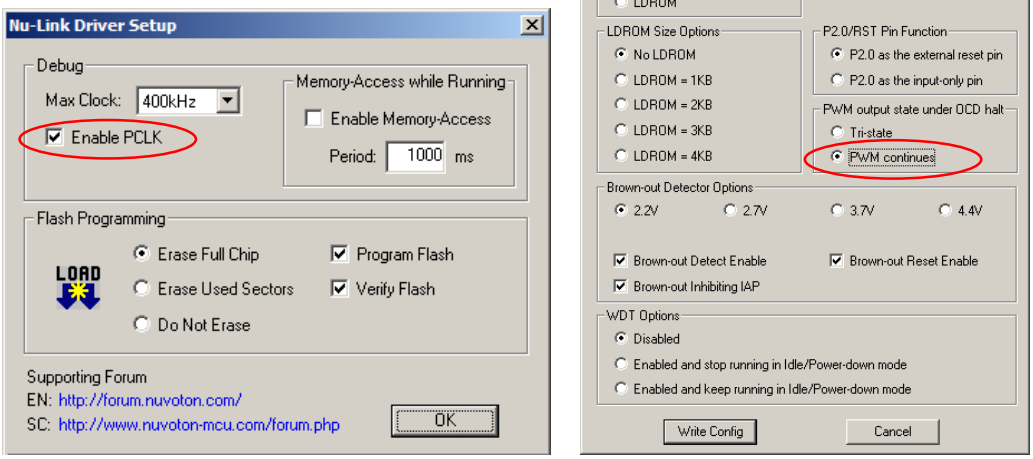
NOTE: This feature is valid for ML51 series, MS51 series and N76 series.



5.9 Timer Clock Keep Run Under OCD Halt

For some MCU such as N76E003, CONFIG setting support PWM continues run under OCD halt, means if break point or step run, PWM keep output as normal setting, this function also need ICE driver setup “Enable PCLK” selected.

NOTE: When this function is enabled, not only the PWM clock keep running and output, all Timer include Timer 0/1/2/3 & WKT) also keep count. This feature is valid for ML51 series, MS51 series and N76 series.

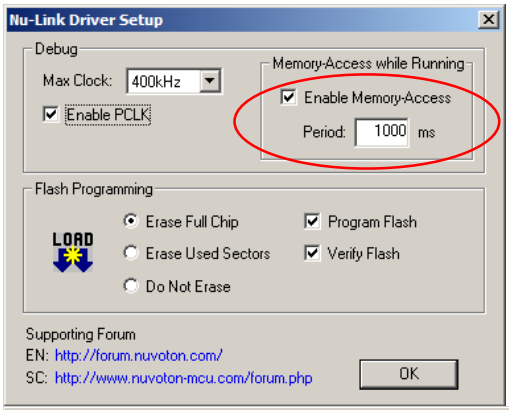


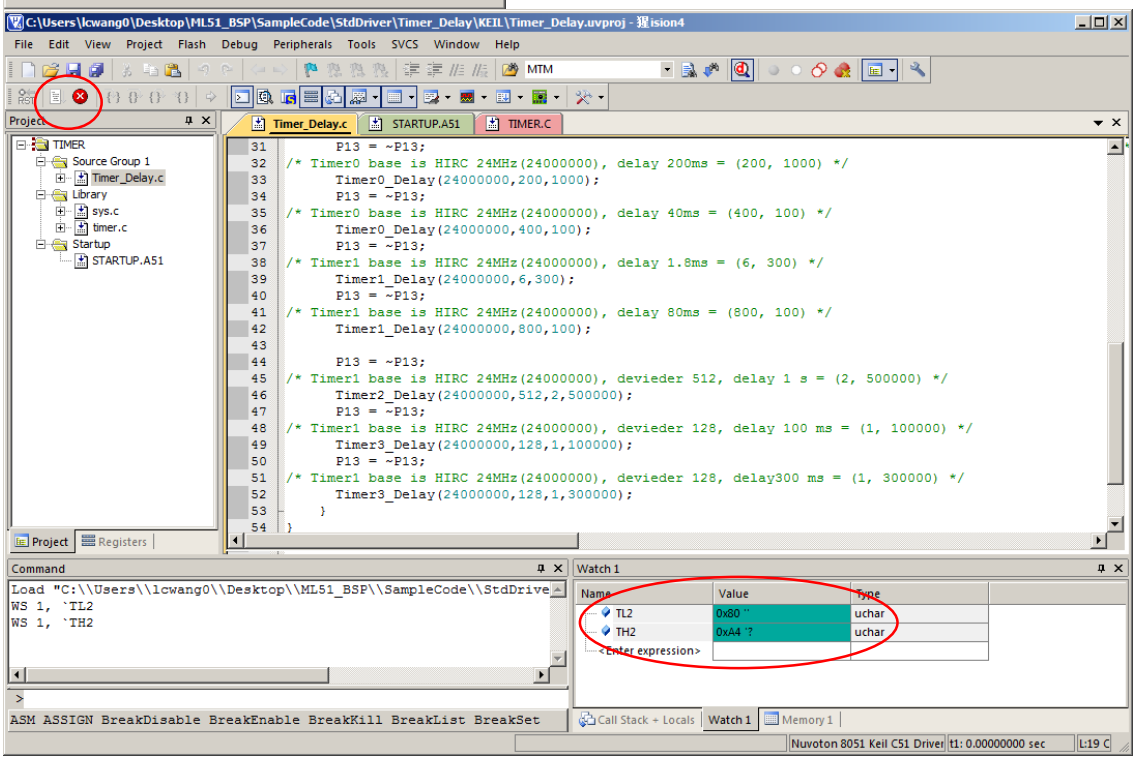
5.10 Memory Access while Runing

When driver setup window with memory access running function item means support SFR and RAM value keep update when ICE in free run mode.

Customer can define the period of refresh timing, the minima value is 100ms.

NOTE: Since the refresh process is stop the MCU running, update memory windown then running MCU again, so this maybe cause some timing issue not match the real clock when this function is enabled. This feature is valid for ML51 series, MS51 series and N76 series.


Nu-Link Driver Setup
Debug
Max Clock:
400kHz
Enable PCLK
Memory-Access while Running
Enable Memory-Access
Period:
1000
ms
Flash Programming
LOAD
Erase Full Chip
Erase Used Sectors
Do Not Erase
Program Flash
Verify Flash
Supporting Forum
EN: http://forum.nuvoton.com/
SC: http://www.nuvoton-mcu.com/forum.php
OK


C:\Users\lcwang0\Desktop\ML51_BSP\SampleCode\StdDriver\Timer_Delay\KEIL\Timer_Delay.uvproj - 狸ision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
MTM
Project
TIMER
Source Group 1
Timer_Delay.c
Library
sys.c
timer.c
Startup
STARTUP.A51
Timer_Delay.c
STARTUP.A51
TIMER.C
31 P13 = ~P13;
32 /* Timer0 base is HIRC 24MHz(24000000), delay 200ms = (200, 1000) */
33 Timer0_Delay(24000000,200,1000);
34 P13 = ~P13;
35 /* Timer0 base is HIRC 24MHz(24000000), delay 40ms = (400, 100) */
36 Timer0_Delay(24000000,400,100);
37 P13 = ~P13;
38 /* Timer1 base is HIRC 24MHz(24000000), delay 1.8ms = (6, 300) */
39 Timer1_Delay(24000000,6,300);
40 P13 = ~P13;
41 /* Timer1 base is HIRC 24MHz(24000000), delay 80ms = (800, 100) */
42 Timer1_Delay(24000000,800,100);
43
44 P13 = ~P13;
45 /* Timer1 base is HIRC 24MHz(24000000), devieder 512, delay 1 s = (2, 500000) */
46 Timer2_Delay(24000000,512,2,500000);
47 P13 = ~P13;
48 /* Timer1 base is HIRC 24MHz(24000000), devieder 128, delay 100 ms = (1, 100000) */
49 Timer3_Delay(24000000,128,1,100000);
50 P13 = ~P13;
51 /* Timer1 base is HIRC 24MHz(24000000), devieder 128, delay300 ms = (1, 300000) */
52 Timer3_Delay(24000000,128,1,300000);
53 }
54 }
Project
Registers
Command
Load "C:\\Users\\lcwang0\\Desktop\\ML51_BSP\\SampleCode\\StdDrive
WS 1, `TL2
WS 1, `TH2
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
Watch 1
Name
Value
Type
TL2
0x80
uchar
TH2
0xA4
uchar
<Enter expression>
Call Stack + Locals
Watch 1
Memory 1
Nuvoton 8051 Keil C51 Driver
t1: 0.00000000 sec
L:19 C

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 Bridge Firmware Version

The ICE/OCD debug tool uses Nu-Link as a bridge interface to communicate with KEIL™ debug environment and the target MCU. The main chip of Nu-Link 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.

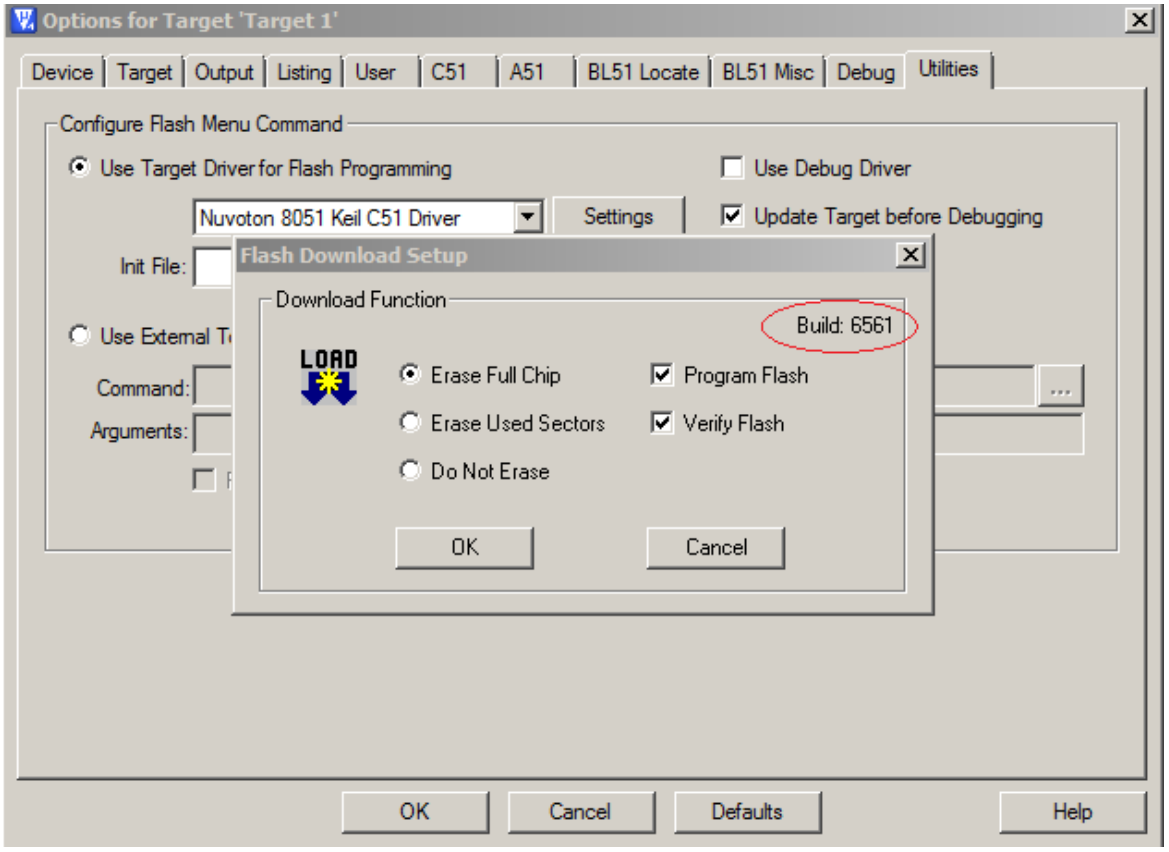


Figure 7-1 Check Bridge Firmware Version

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
2.01	Nov 8, 2017	1. Modify the name of keil driver. 2. Modify RAM picture and description.
2.06	Jan 21, 2019	1. Added ML51 and MS51 in chapter 2 support series table 2. Added chapter 5.8~5.10

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.