N9H2x Keil ICE Debug User Manual

Document Information

Abstract	This document instructs the user how to debug Keil application program using J-Link ICE.
Apply to	N9H2x series

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

www.nuvoton.com

Table of Contents

1 Introduction

N9H2x series provide Keil IDE as Non-OS BSP development environment and hardware supports J-TAG debug interface. Users could use this interface to download application program to DRAM for debugging. The N9H2x Non-OS sample code uses J-Link ICE for debugging and EVB connected to J-Link ICE as shown in Figure 1-2.

Figure 1-1 J-Link ICE

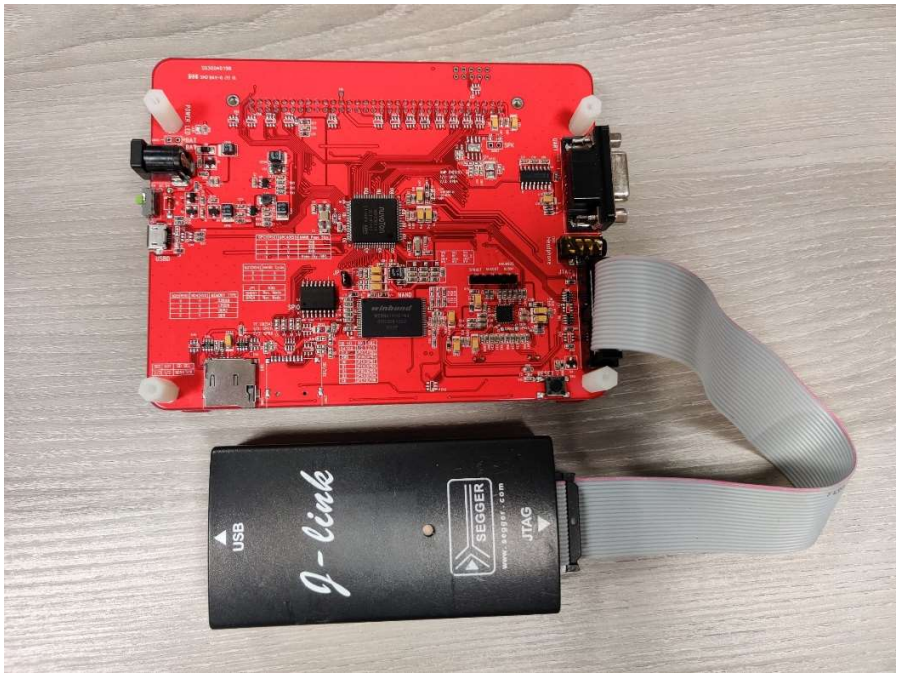

Figure 1-2 EVB Connect ICE

2 Setup Debug Environment

ARM9 based microcontrollers require the additional installation of the Keil MDK 4 Legacy Support. The user can install “NuMicro ARM9 Series Device Database Keil” program which download from the website https://www.nuvoton.com/products/microprocessors/arm9-mpus/n3290-mjpeg-series/n32901r1dn/?group=Software&tab=2 to instead of installation of Legacy Pack.

Keil supports debugging using J-Link ICE. The J-Link plug-in program can be downloaded from the website http://www.segger.com/downloads/jlink/#J-LinkSoftwareAndDocumentationPack and recommend select the version above 6.88a for “NuMicro ARM9 Series Device Database Keil” program.

2.1 DEV Board Setting

N9H2x series have two boot flows, one is Normal mode and the other is Recovery mode. If user wants to use ICE for debugging, the EVB should be set to Recovery mode. Further EVB information can be found at UM_NuMaker-emWin-NK-N9H20_User_Manual_EN_Rev2.01.pdf which can be downloaded from the website:

https://www.nuvoton.com/products/microprocessors/arm9-mpus/-n9h-series/n9h20k51n/?group=Document&rt=User%20Manual&tab=2

2.2 Keil Project Setting

After opening Keil project, the user can select corresponding target depend on the DUT part number if the project has multi-targets.

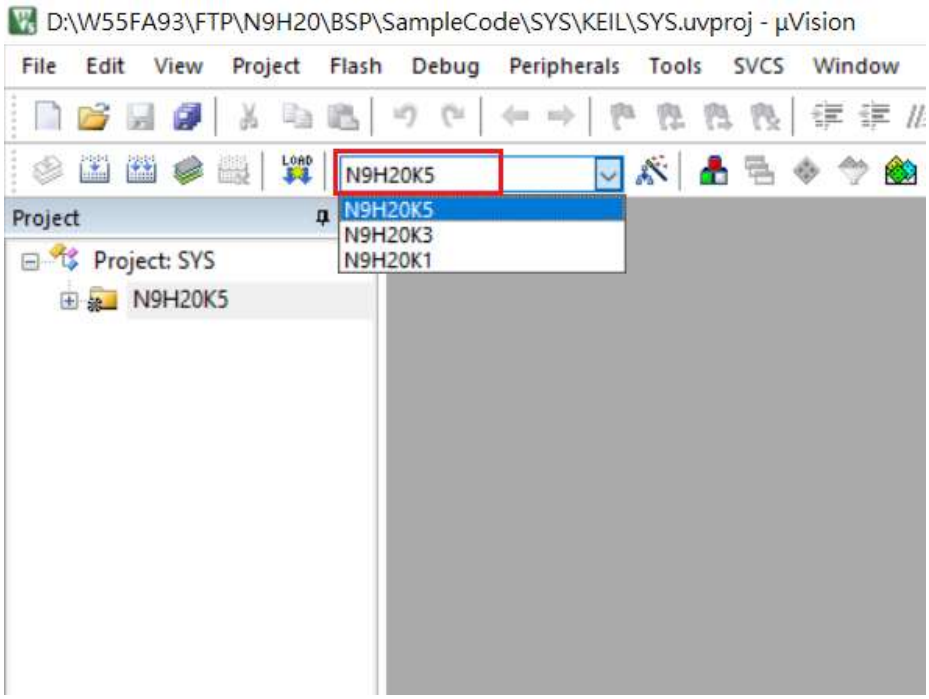

Figure 2-1 Select Target

The user can set device of project if they do not install Legacy Pack. As shown in Figure 2-2, click the **Options for Tartet** icon ➔ select **NuMicro ARM9 Database** ➔ click **Nuvoton_ARM9_Series** and then click **OK** button.

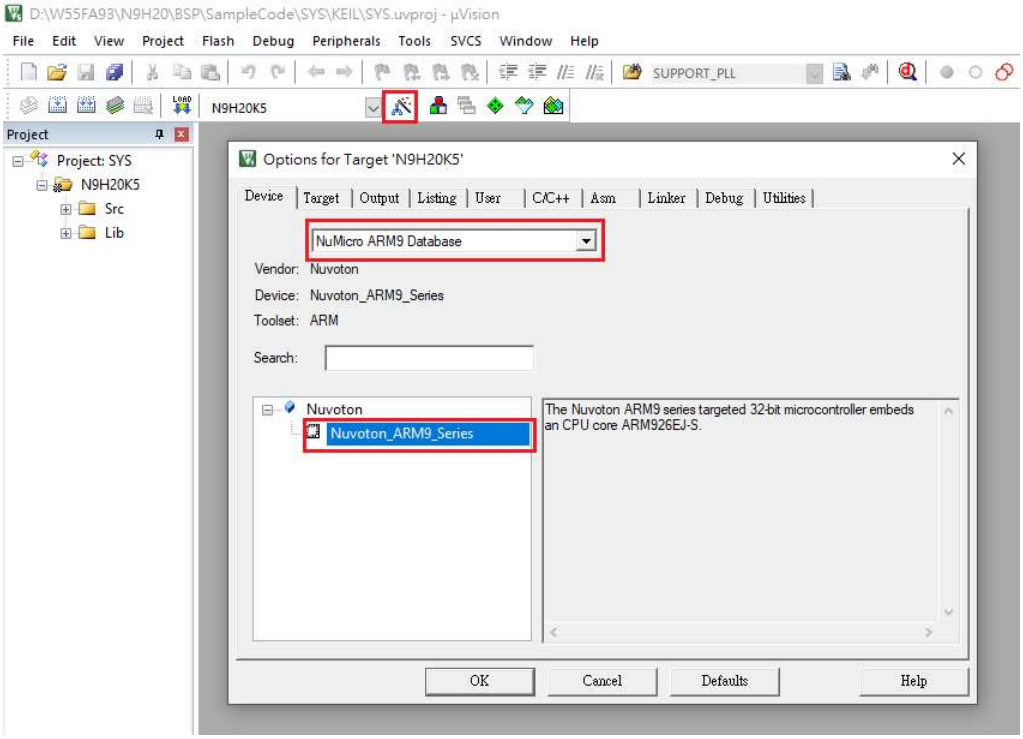

Figure 2-2 Device Setting

The next step is to set Debug options. As shown in Figure 2-3 and Figure 2-4 respectively, click the **Debug** button ➔ select **J-LINK/J-TRACE ARM** and then click **Settings** button to set JTAG interface configuration.

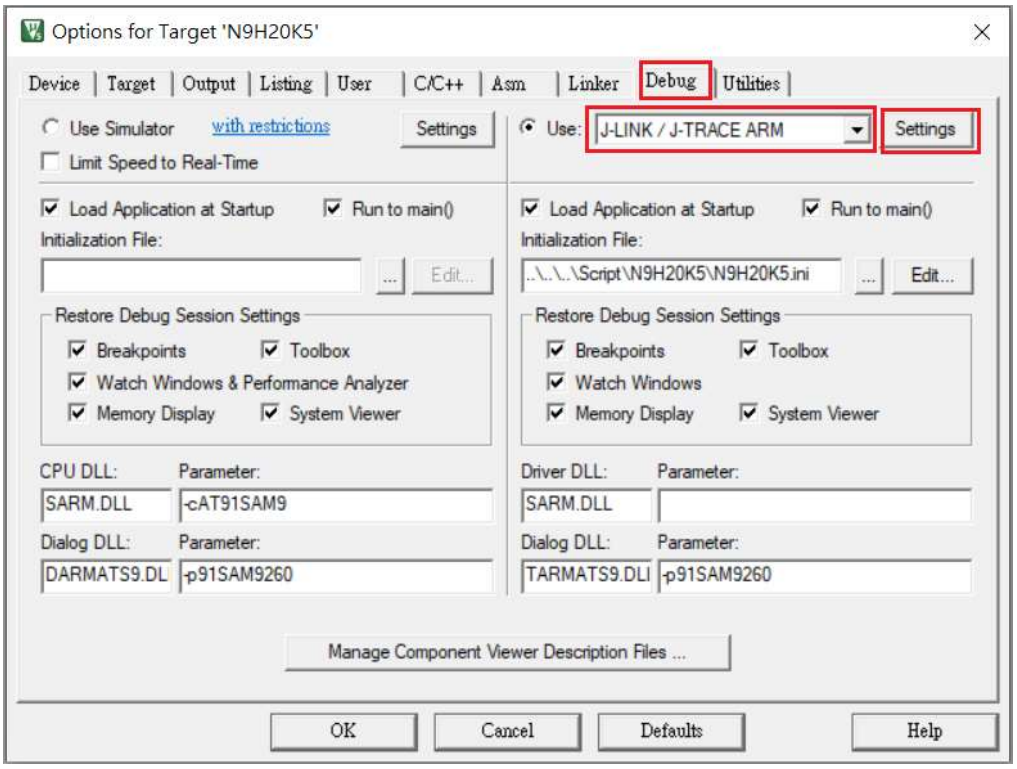

Figure 2-3 ICE Selection

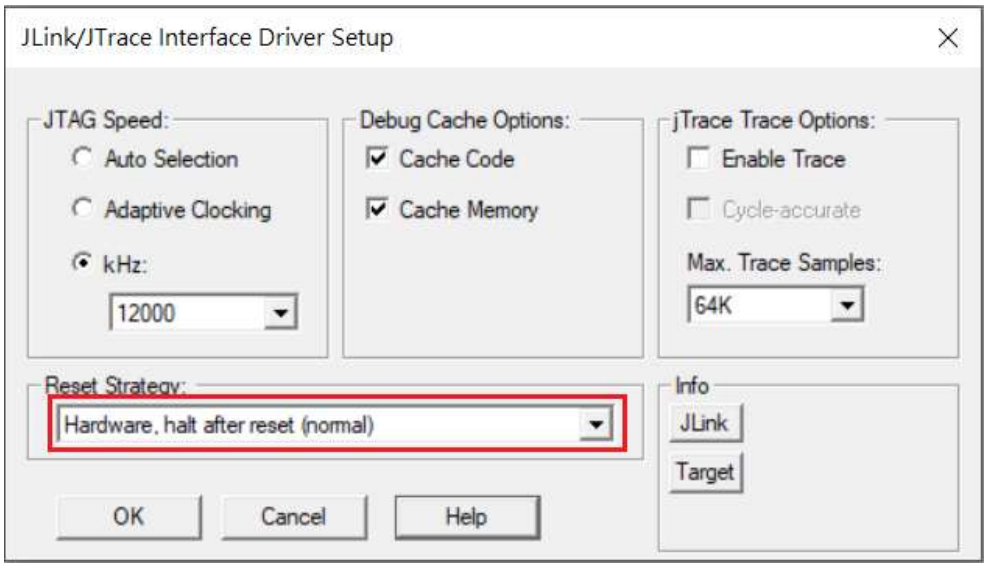

Figure 2-4 JLink Setting in Keil MDK

The N9H2x Non-OS sample code is executed in DRAM. Before entering debug mode, it is necessary to initialize DRAM. The initialize script files are located in BSP's Script/ directory. All initialization files of N9H2x series are listed in Table 2-1. There is a unique file for each DRAM type. Please select the file according to the debug target.

Directory Name	Description
N9H20K1	N9H20K1.ini is for IP usage example, which are executed at the address 0x0.
N9H20K3	N9H20K3.ini is for IP usage example, which are executed at the address 0x0.
N9H20K5	N9H20K5.ini is for IP usage example, which are executed at the address 0x0.
N9H26	N9H26K5.ini is for IP usage example, which are executed at the address 0x0.
N9H26	N9H26K6.ini is for IP usage example, which are executed at the address 0x0.

Table 2-1 Initialization File for Keil

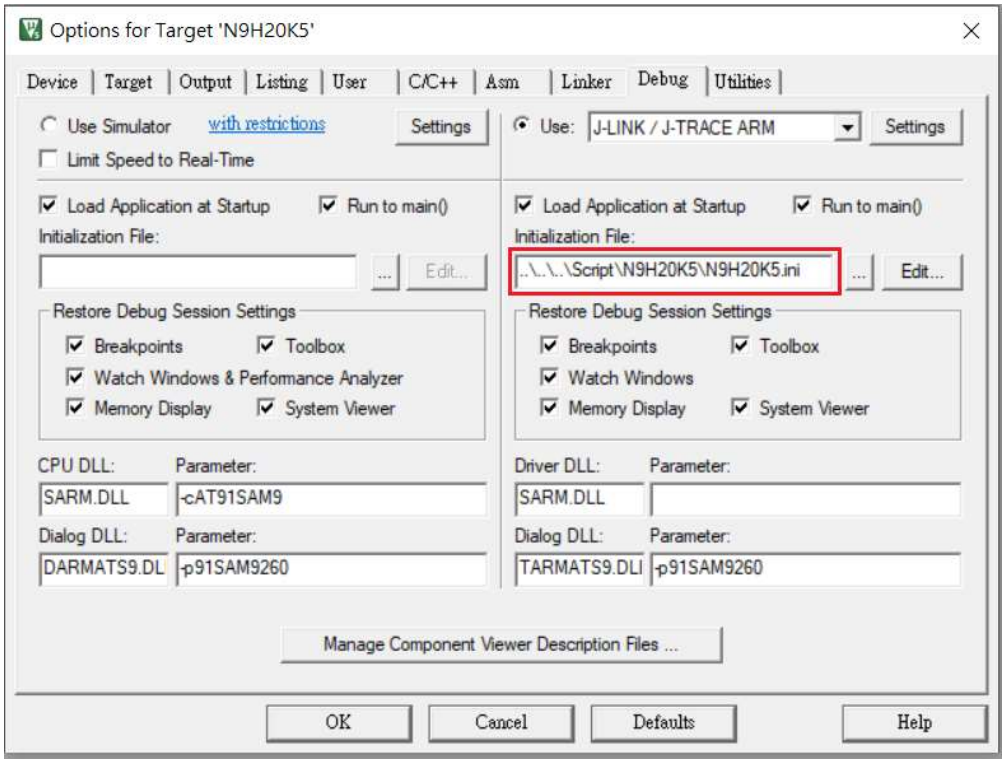

Figure 2-5 Initialization File Selection

The screen is split into options for the Simulator and for the target driver.

Use Simulator

Configures the Debugger as a software-only product that simulates the instruction set of an Arm based microcontroller.

Debug Driver

Selects a driver for debugging the target hardware. For example, a J-LINK adapter driver that uses the embedded on-chip debugging hardware.

Load Application at Startup

Loads the object file specified in the field **Options for Target ➔ Output ➔ Name of Executable** into the Debugger.

Run to main()

Executes the instructions until main() is reached and stops execution. When disabled, the application stops at the first instruction. Commands defined in **Initialization File** are executed regardless of this option.

2.3 Start Debugging

To build application program in Keil MDK, click the **Rebuild** icon as shown in Figure 2-6.

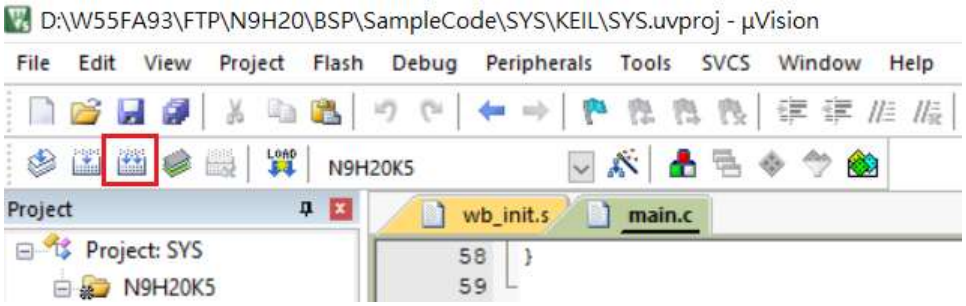

Figure 2-6 Rebuild Sample Cpde in Keil MDK

Start debugging using the menu **Debug ➔** click **Start/Stop Debug Session** to start debug with J-Link ICE as shown in Figure 2-7.

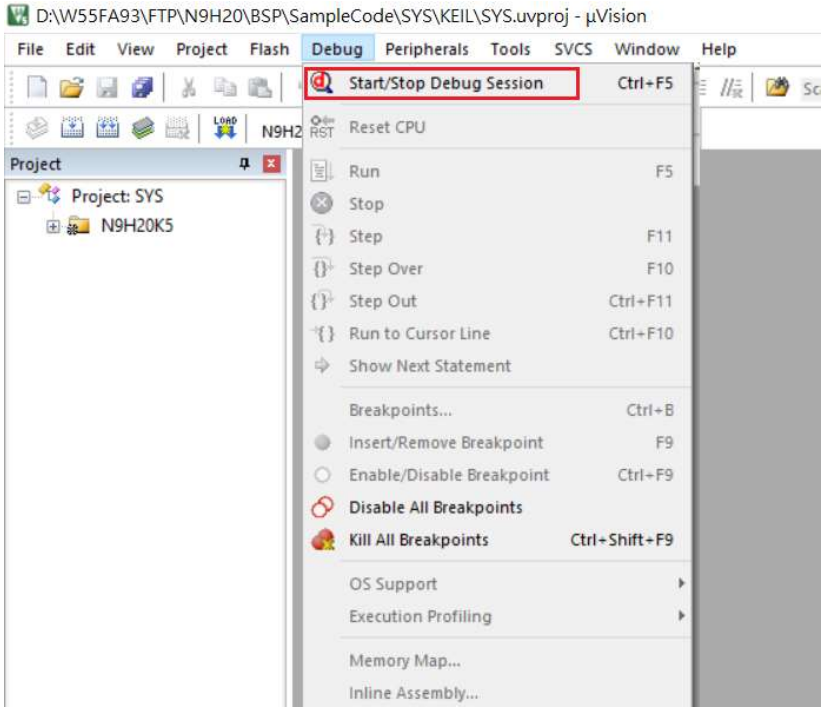

Figure 2-7 Download Binary to Device and Start/Stop Debug Session

This section introduces some simple operations for debugging.

1. Free Run the application program as shown in Figure 2-8.

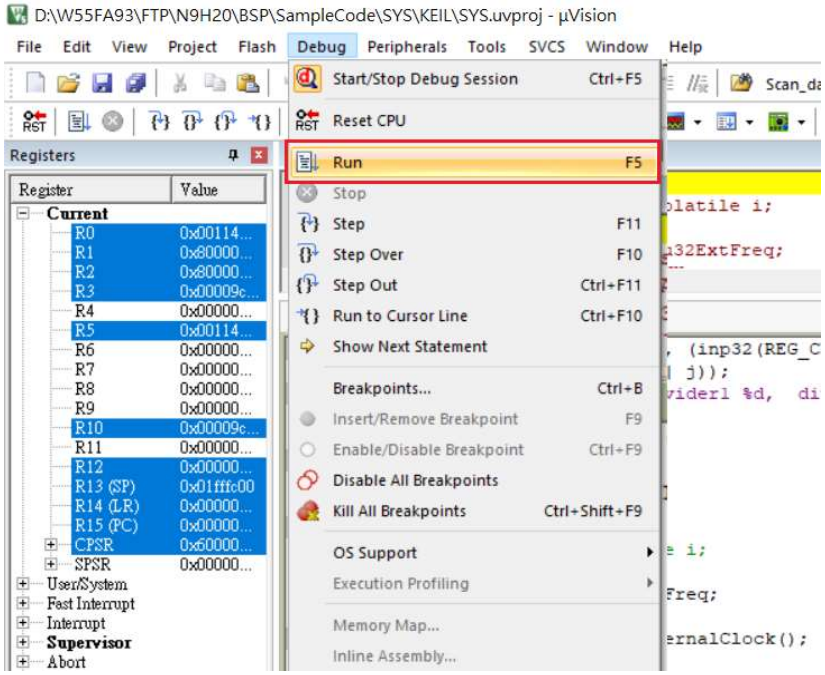

Figure 2-8 Free Run

2. Step & Step Over the function call as shown in Figure 2-9.

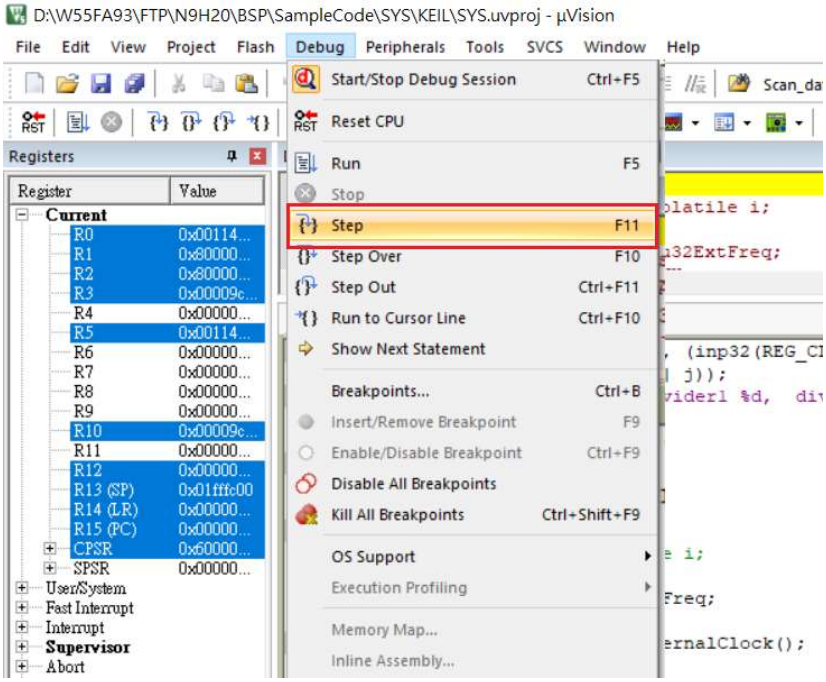

Figure 2-9 Step & Step Over

3. The **Memory** window shows the contents of the connected target's memory areas and allows the memory to be edited. Several memory windows can be used at a time. For example, the addresses 0xB0000200 shows content of clock control registers. Memory window and memory content as shown in Figure 2-10 and Figure 2-11 respectively.

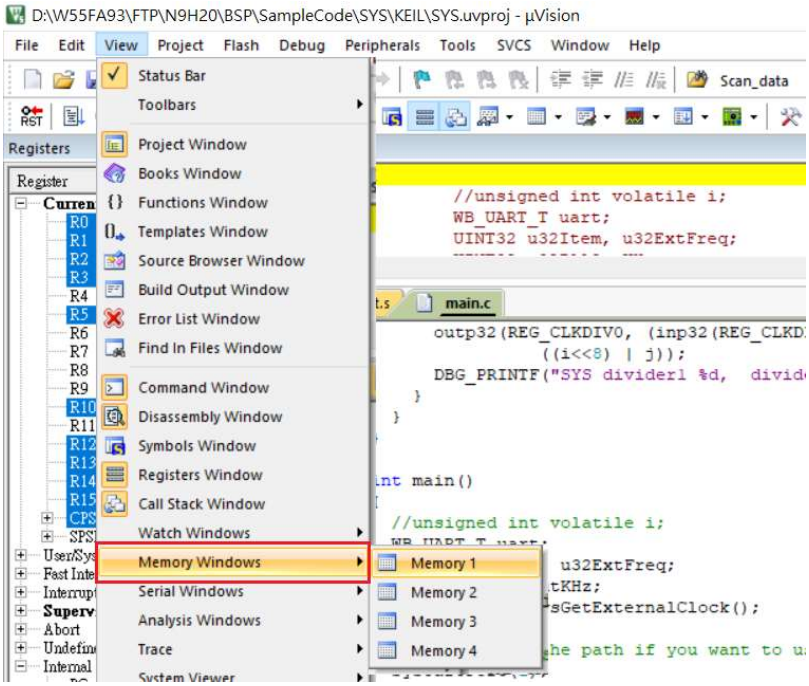

Figure 2-10 Memory Windows

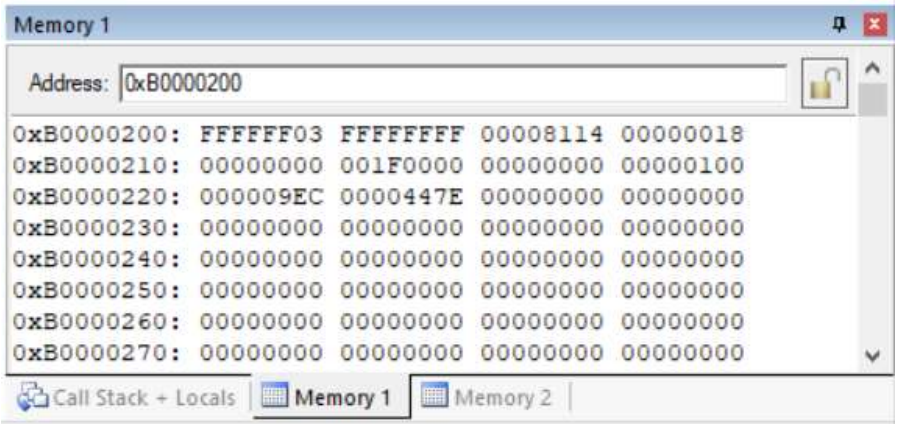

Figure 2-11 Content of Clock Control Registers

4. The **Registers** window includes the following registers:
   - 13 general-purpose registers R0-R12.
   - One Stack Pointer (R13 or SP).
   - One Link Register (R14 or LR).
   - One Program Counter (R15 or PC).

   The registers are available and accessible in any processor mode. When the program stops at a breakpoint, or is stepped, the **Registers** windows update to show the current values of the registers.

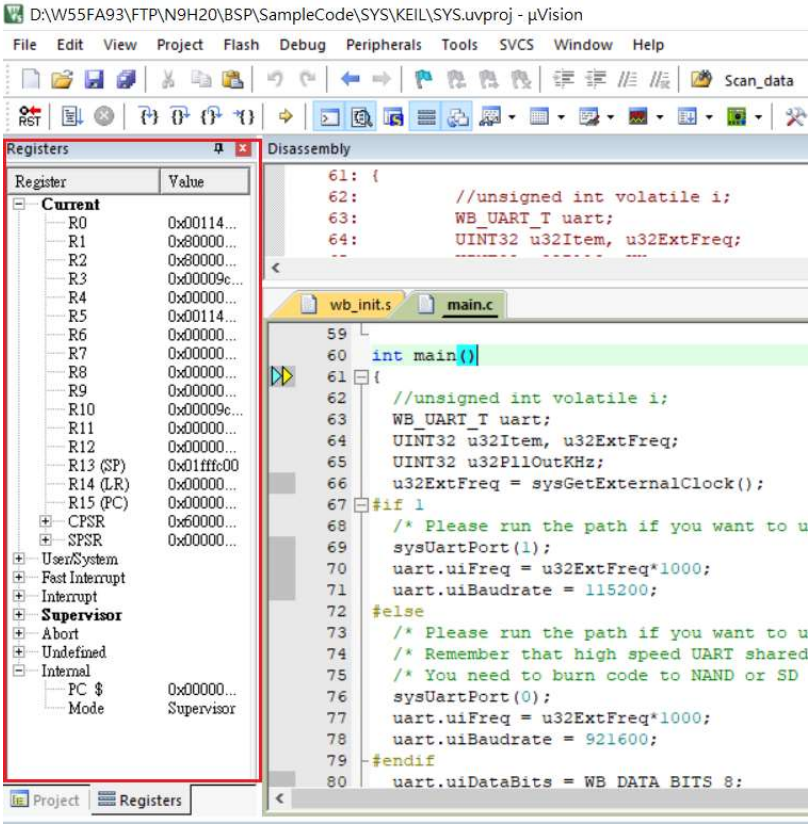

Figure 2-12 Registers Window

5. The **Disassembly** window shows the program execution in assembly code. Open the window with the menu **View** ➔ click **Disassembly Window**

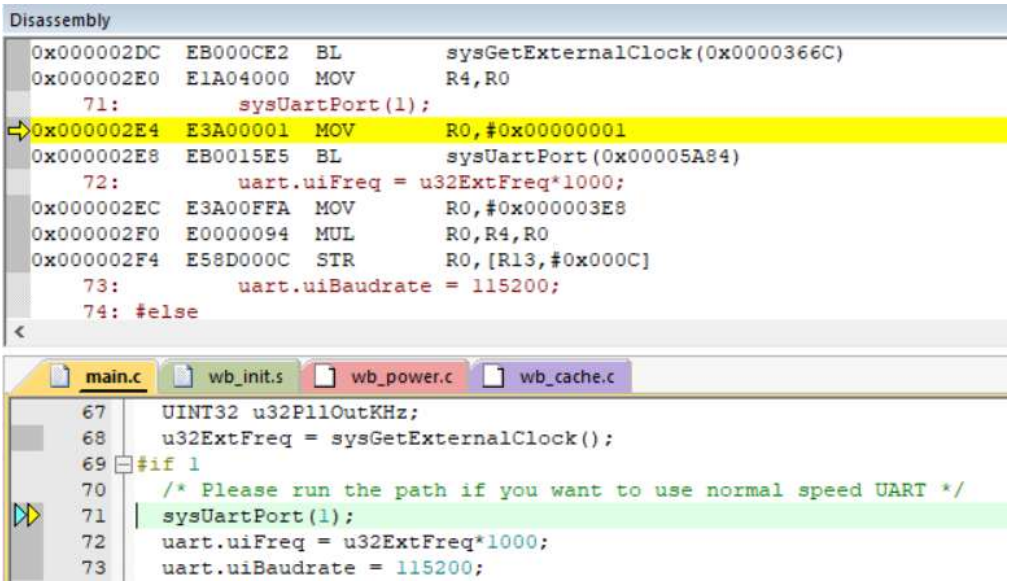

Figure 2-13 Disassembly Window

6. A breakpoint can be enabled and disabled by left-clicking on its icon as shown in Figure 2-14.

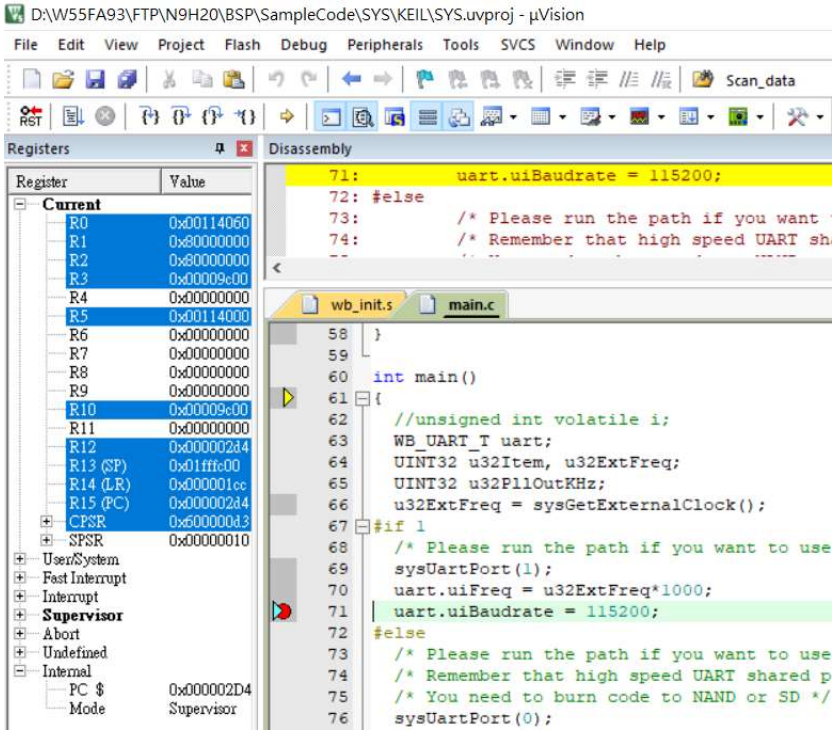

Figure 2-14 Set Breakpoint

As for other operations in the debugger, the user can refer to **Help** window to get further information.

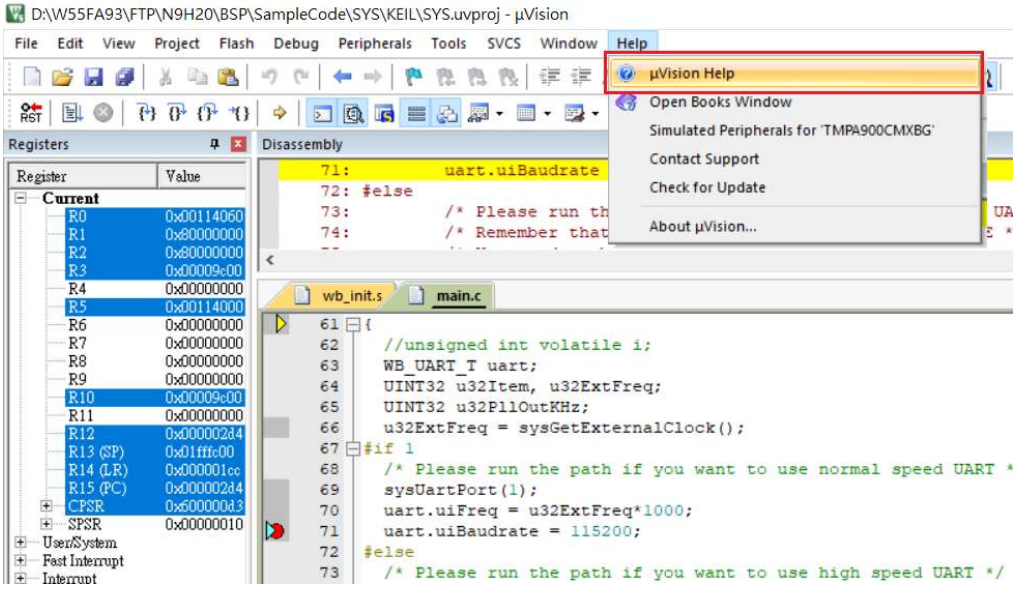

Figure 2-15 Help Window

The result of SYS sample code execution as shown in Figure 2-16 and related terminal setting as shown in Figure 2-17.

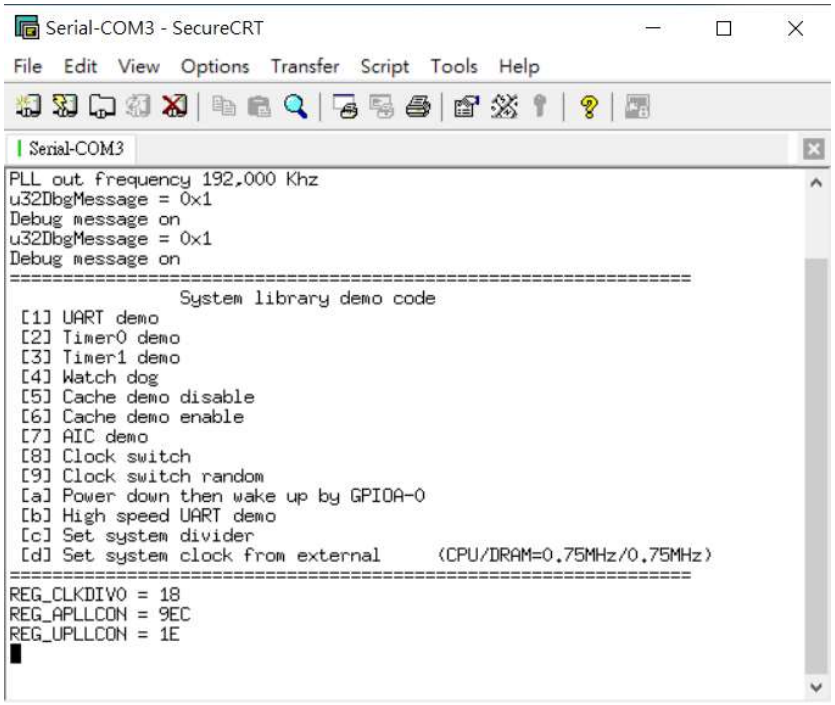

Figure 2-16 Result of SYS Sample Code

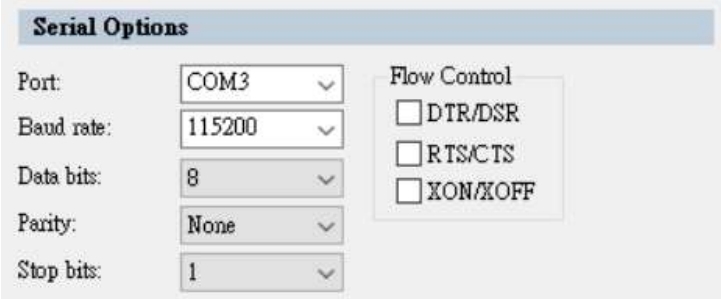

Figure 2-17 Terminal Setting

Revision History

Date	Revision	Description
2021.07.26	1.00	1. Initially issued.

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.