NUC980 Non-OS BSP User Manual

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

www.nuvoton.com

1 Introduction to NUC980 Non-OS BSP

This BSP supports Nuvoton NUC980 series processors. The NUC980 series targeted for general purpose 32-bit microprocessor embeds an outstanding CPU core ARM926EJ-S, a RISC processor designed by Advanced RISC Machines Ltd., runs up to 300 MHz, with 16 KB I-cache, 16 KB D-cache and MMU, 16KB embedded SRAM and 16.5 KB IBR (Internal Boot ROM) for booting from USB, NAND and SPI Flash.

The NUC980 series integrates USB 2.0 HS HOST/Device controller with HS transceiver embedded, I^2S I/F controller, SD/MMC/NAND Flash controller, PDMA and 8 channels 12-bit ADC controller. It also integrates UART, SPI/MICROWIRE, I^2C, LIN, PWM, Timer, WDT/Windowed-WDT, GPIO, Smart Card I/F, 32.768 kHz XTL and RTC (Real Time Clock).

This Non-OS BSP includes following contents:

- NUC980 Non-OS drivers.
- Precompiled U-Boot images for different boot mode.
- Flash programming tool NuWriter, and its Windows driver.
- User manuals.

2 BSP Content

2.1 BSP directory structure

The Non-OS BSP contains four directories. The content of each directory listed in the table below.

Directory Name	Content
BSP	Directory contains Non-OS driver, third party software and sample applications.
Documents	BSP related documents
Loader	Pre-compiled U-Boot image files.
Tools	Tool for programming NAND, SPI, eMMC or download image to DDR. And its Windows driver.

2.2 Non-OS BSP content

The file under BSP directory shows following content.

Directory Name	Content
Driver	NUC980 peripheral drivers. Please refer to NUC980 Non-OS BSP Driver Reference Guide.chm under Documents directory for the usage of driver APIs.
Library	NUC980 libraries, including USB Host and smartcard.
SampleCode	Driver sample application.
Script	Link script and debug initialization file for Keil.
ThirdParty	Third party software. Including FATFS, yaffs2 file system and LwIP network protocol stack.

3 Development Environment

Keil IDE or Eclipse is used as Non-OS BSP development environment. and use ULINK2 (Keil only) or J-Link ICE for debug.

The NUC980 series supports J-TAG debug interface. Users could use this interface to download programs to DDR and debug.

The NUC980 Non-OS BSP uses the same open source loader as Linux BSP, U-Boot. U-Boot uses Linux as development environment. Users could ether the precompiled U-Boot images in this BSP. If it is necessary to modify and re-build U-Boot, please refer to NUC980 U-Boot user manual to set up the development environment. If the system boot from SPI NOR Flash or eMMC/SD, it is not required to use a loader and can execute main program directly after system booting up. But while booting from NAND Flash, it is recommend using a loader while booting from NAND Flash and let it handles the bad block during system boot up.

3.1 DEV Board Setting

The NUC980 series supports different boot modes, it can boot from SPI, NAND, eMMC/SD, or enter USB ISP mode. The booting mode is selected by PG[1:0] power-on-setting. Please refer to DEV board user's manual for boot source selection of DEV board.

3.2 Keil Project Setting

The NUC980 Non-OS sample code is executed in DDR. Before entering debug mode, it is necessery to initialize DDR. The initialize script files are loacted in BSP's Script/ direcoty. There is a unique file for each DDR type. Please select the file according to the debug target. The IDE does not belong to the content of this document. Please refer to official Keil website http://www.keil.com/ for the Keil IDE user manual.

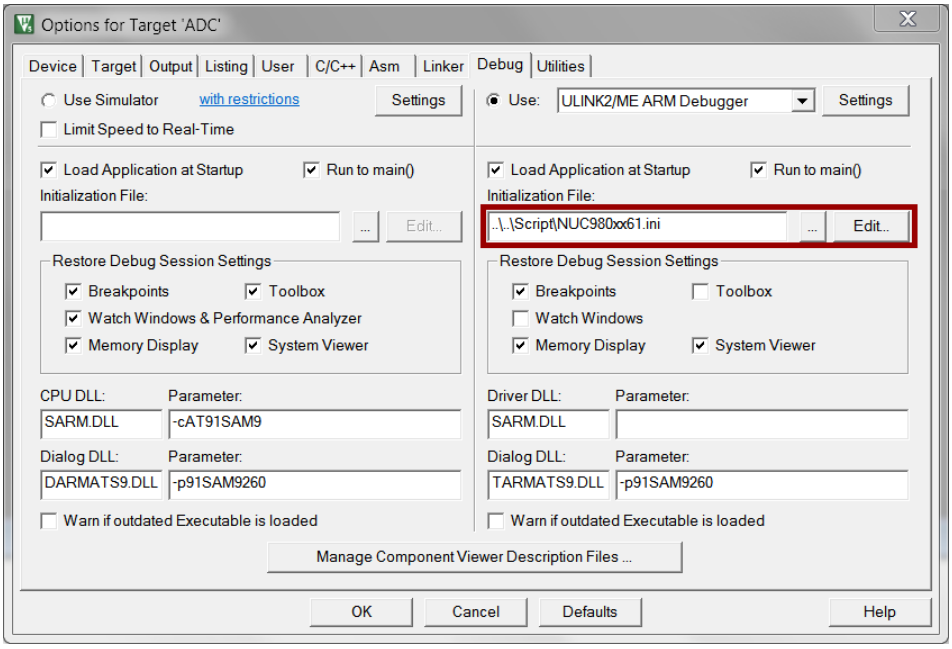

Figure 3-1 Initialization File Selection

3.3 Eclipse Development Environment

The NUC980 Non-OS BSP also supports using Eclipse as development IDE. This section introduces the installation steps of Eclipse development environment. First, download Eclipse IDE for C/C++ Developers Tool from Eclipse official website: https://www.eclipse.org/downloads/, select proper version according to your operating system. Since Eclipse is a Java based

application, please download JRE from Java website and install it.

Download the make tool from https://github.com/xpack-dev-tools/windows-build-tools-xpack/releases . Select “GNU MCU Eclipse Windows Build Tools v2.12 20190422” to download “gnu-mcu-eclipse-windows-build-tools-2.12-20190422-1053-win64.zip” and extract it to C:\eclipse.

Download gcc toolchain from https://developer.arm.com/downloads/-/gnu-rm . Select “gcc-arm-none-eabi-10.3-2021.10-win32.zip” and extract to C:\eclipse.

The cross compile - GNU ARM Embedded Toolchain can be downloaded from the website: https://gnu-mcu-eclipse.github.io/plugins/install/. After installing the software packages memtioned above, execute Eclipse and select Help -> Eclispe Marketplace.

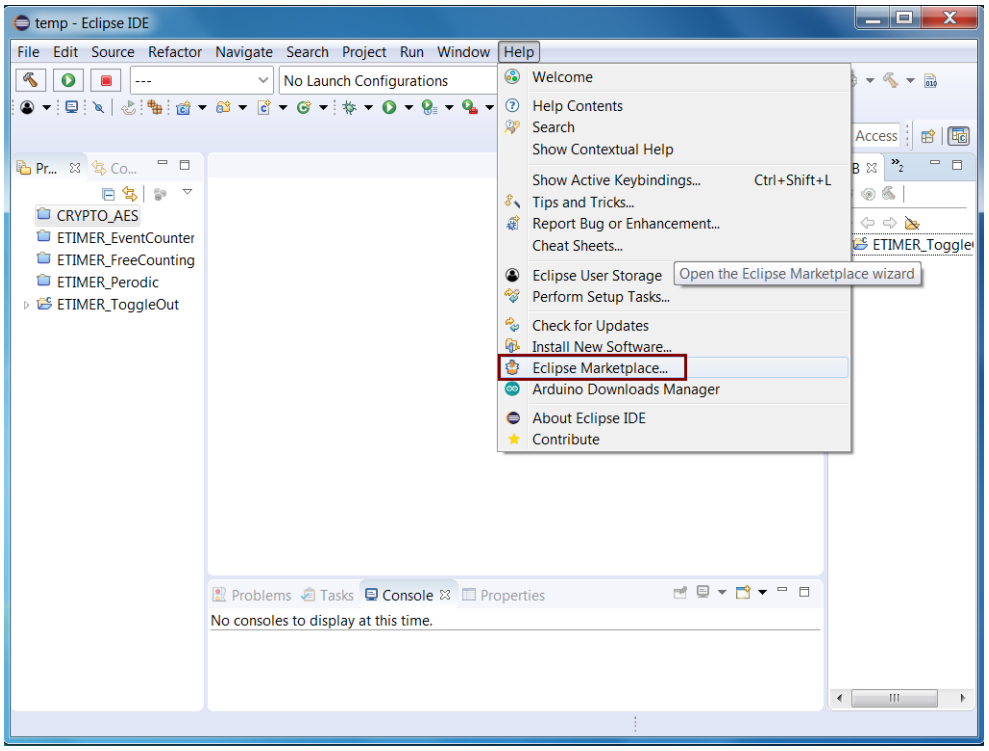

Figure 3-2 Select Eclipse Marketplace

Input “**gnu mcu eclipse**” in the “**Find**” field, and then the search result will be shown as Figure 3-3. Select the latest version and click “**Install**” button to install the required plug-in.

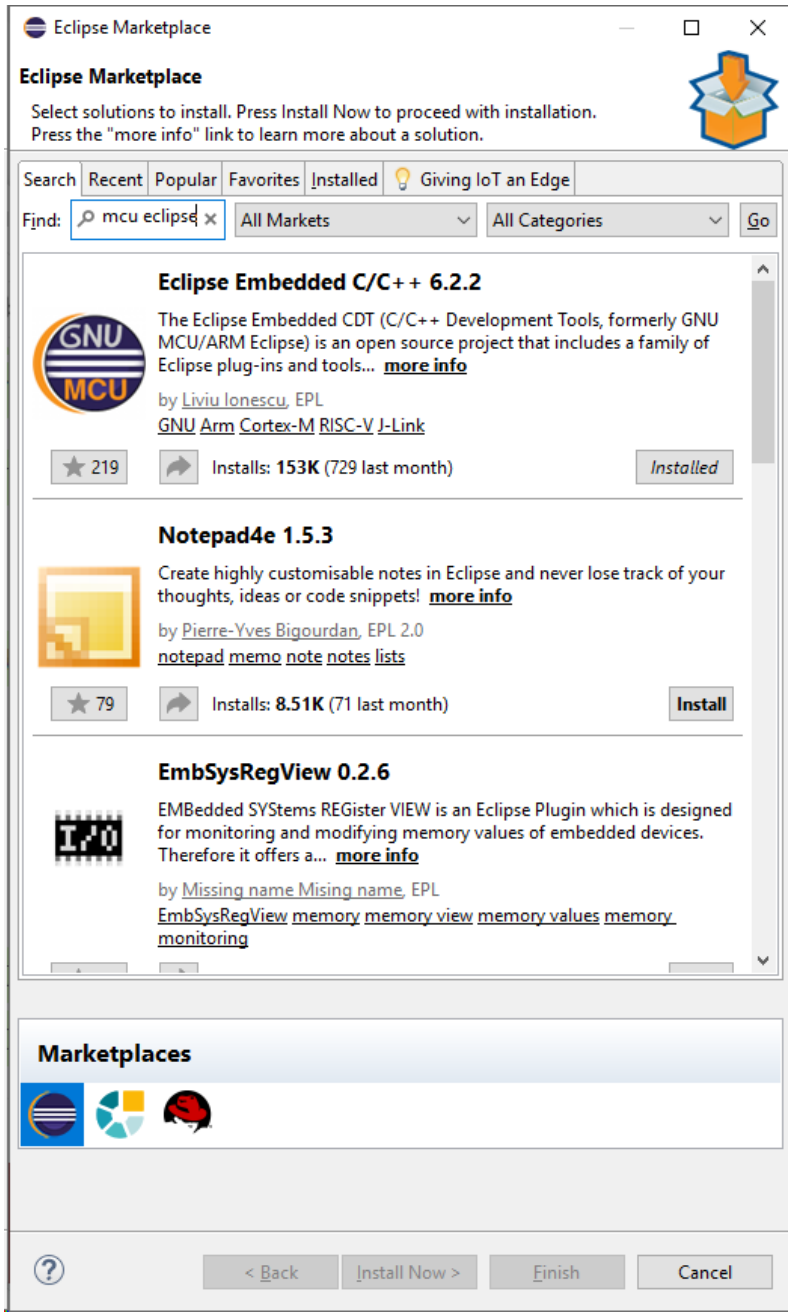

Figure 3-3 Install Plug-in

Click "**Help -> Install New Software**" to install CDT to support C/C++ development.

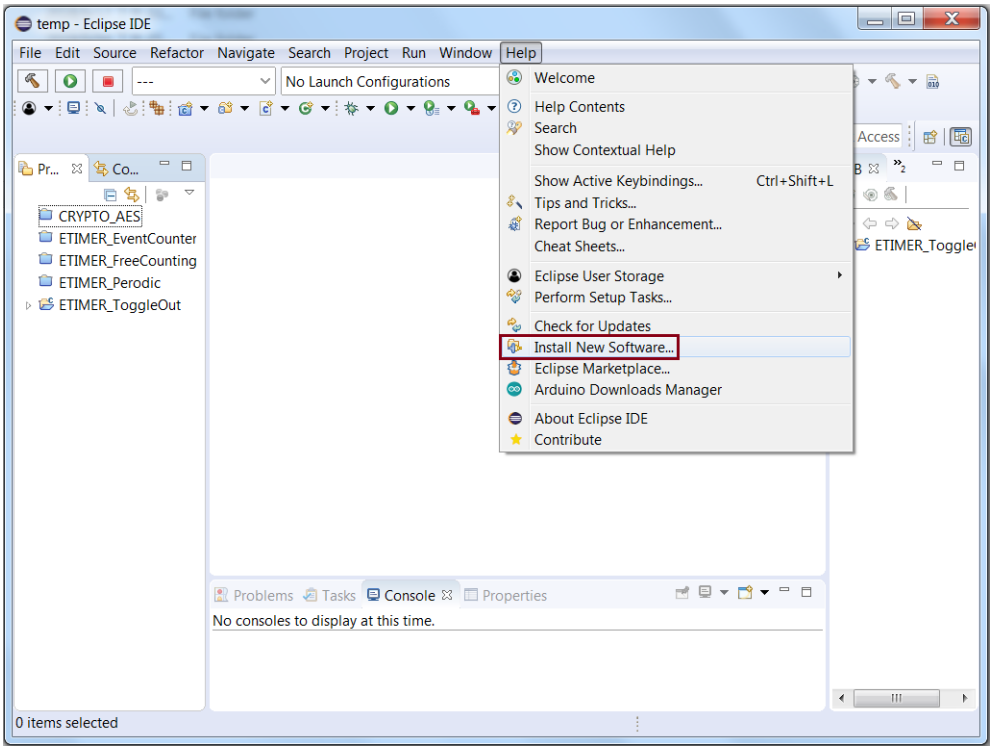

Figure 3-4 Install New Software

Input "**cdt**" in "**Work with**" field.

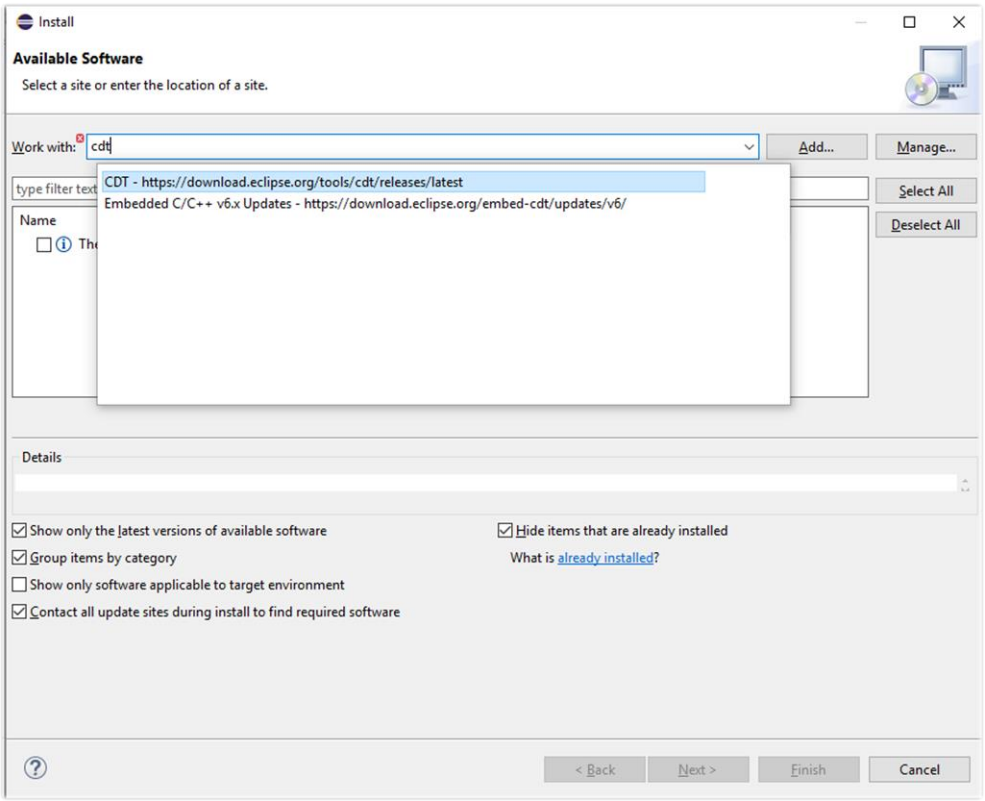

Figure 3-5 Search for CDT

Select “**CDT Main Features**” and “**CDT Optional Features**” as shown in Figure 3-6. User can also select other pakcages if necessary.

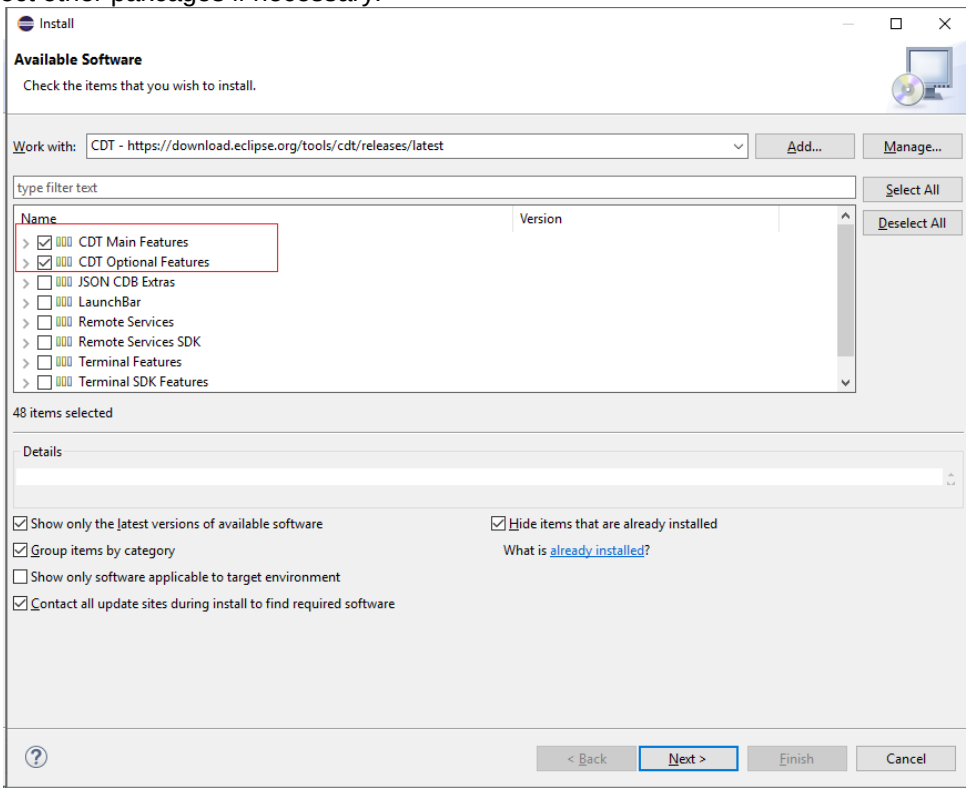

Figure 3-6 Select CDT

After installing CDT, re-start Eclipse. Setup the tool chian which should have been download and extract to c:\eclipse. Open the preference window from eclipse main menu “Windows” ➔ “Preferences” and select “MCU” ➔ “Global Arm Toolchains Paths” and browse to select path “C:\eclipse\gcc-arm-none-eabi-10.3-2021.10\bin” as shown in Figure 3-7.

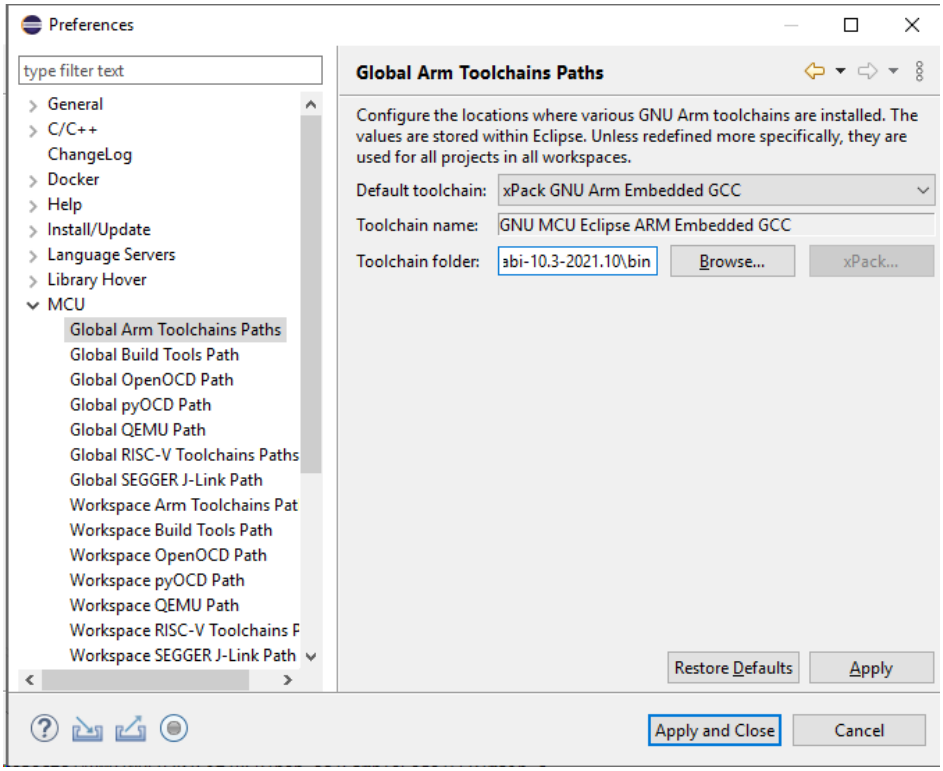

Figure 3-7 Configure tool chain path

Select the next “Global Build Tools Path” and browse to select path “C:\eclipse\GNU MCU Eclipse\Build Tools\2.12-20190422-1053\bin” as show in Figure 3-8.

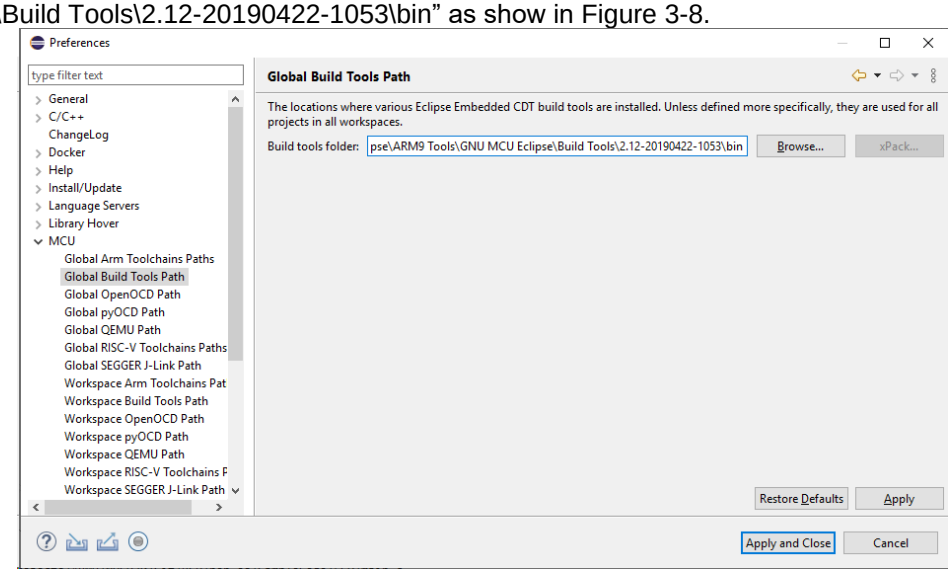

Figure 3-8 Configure build tool path

After installing CDT, re-start Eclipse to import Eclipse project.
Eclipse supports debugging using J-Link ICE. Download and install J-Link plug-in from the website: http://gnuarmeclipse.github.io/plugins/install/ before starting debugging. After installation, set the J-Link path through “**Preference->MCU-> Global SEGGER J-Link Path**”, and then click “**Apply**” button.

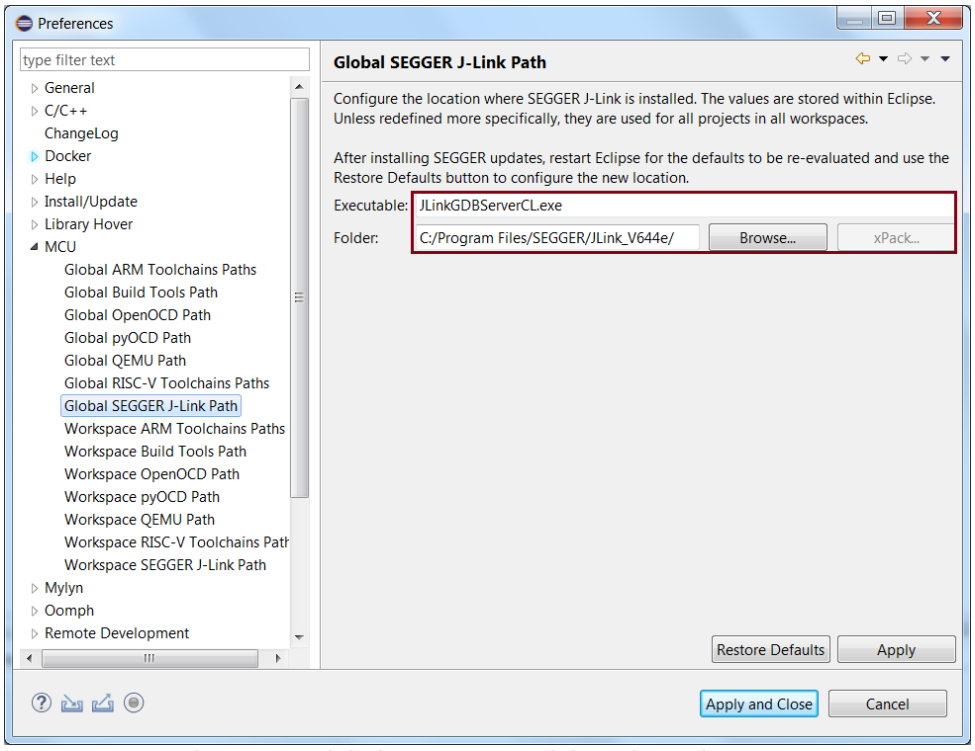

Figure 3-9 Global SEGGER J-Link Path Setting

The next step is to set GDB SEGGER J-Link Debugging options. Click "**Run -> Debug Configurations**" and then double click "**GDB SEGGER J-Link Debugging**".

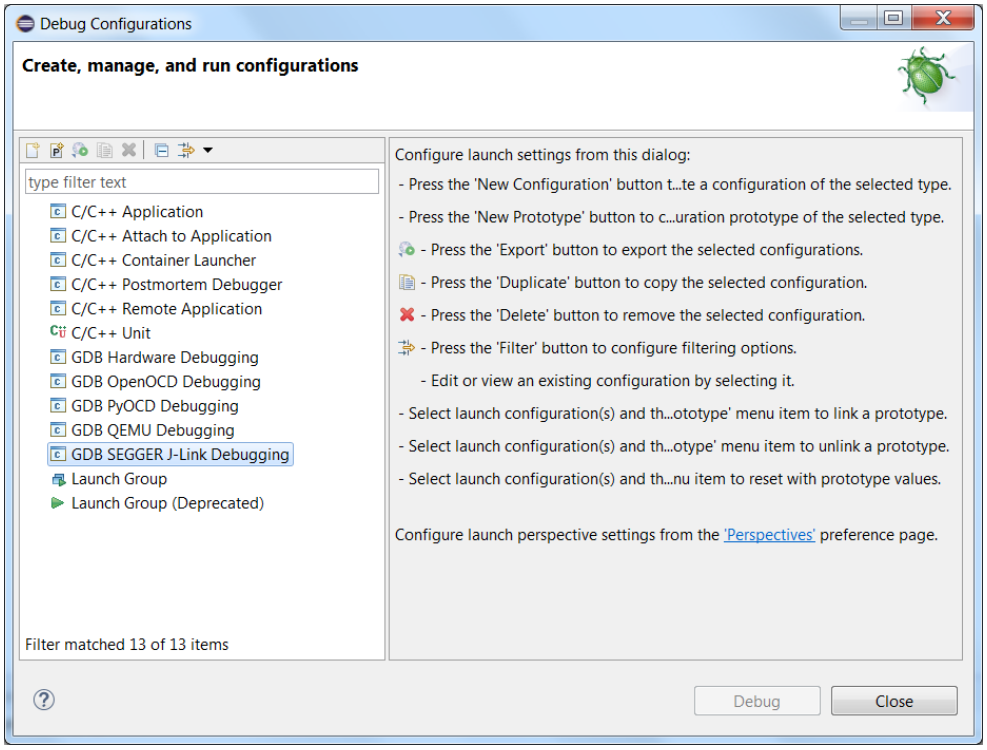

Figure 3-10 GDB SEGGER J-Link Debug

Go to "**Debug**" tab, and configure this tab as Figure 3-11.

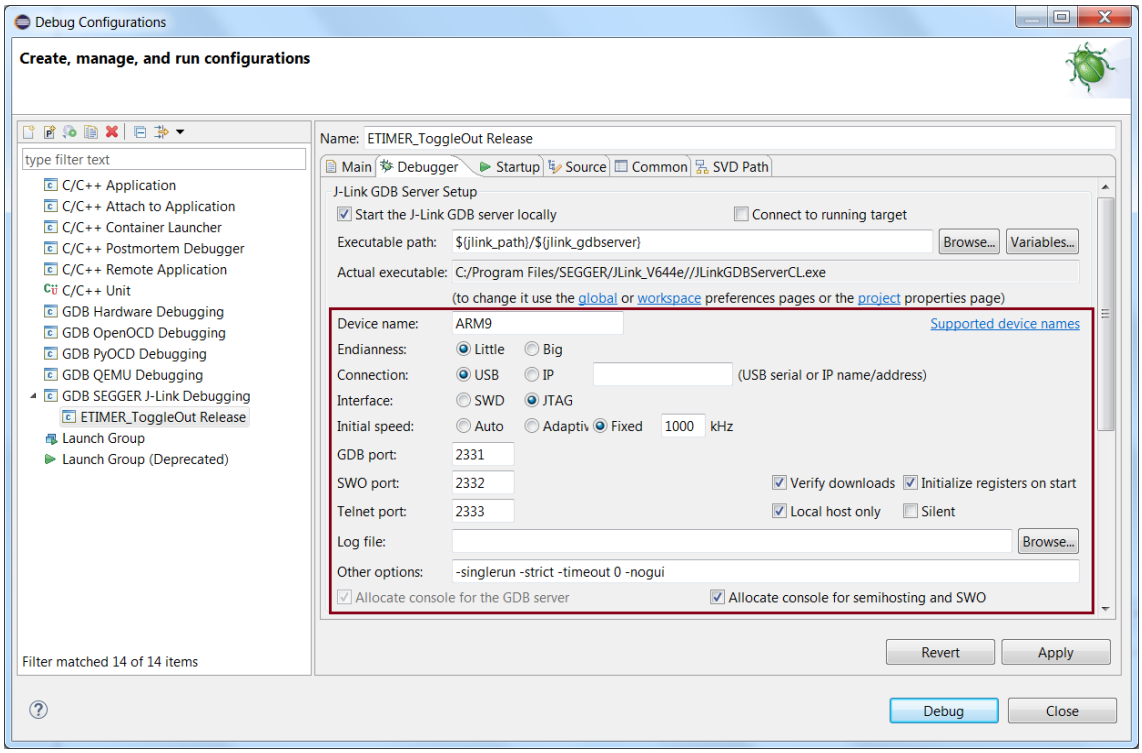

Figure 3-11 J-Link Debugger Setting

Go to "**Startup**" tab, and configure this tab as Figure 3-12. Please note that the DDR initial parameters depend on the DUT part number. The correct setting can be modified from the ini files under the *Script/* directory in BSP by replacing "**`E INT`**" with "**`monitor long`**" in the ini file. For example, to debug NUC980DF61YC, user should open *NUC980xx61.ini* with editor and replace all "**`E INT`**" with "**`monitor long`**" and then copy the modified contnet into the command box.

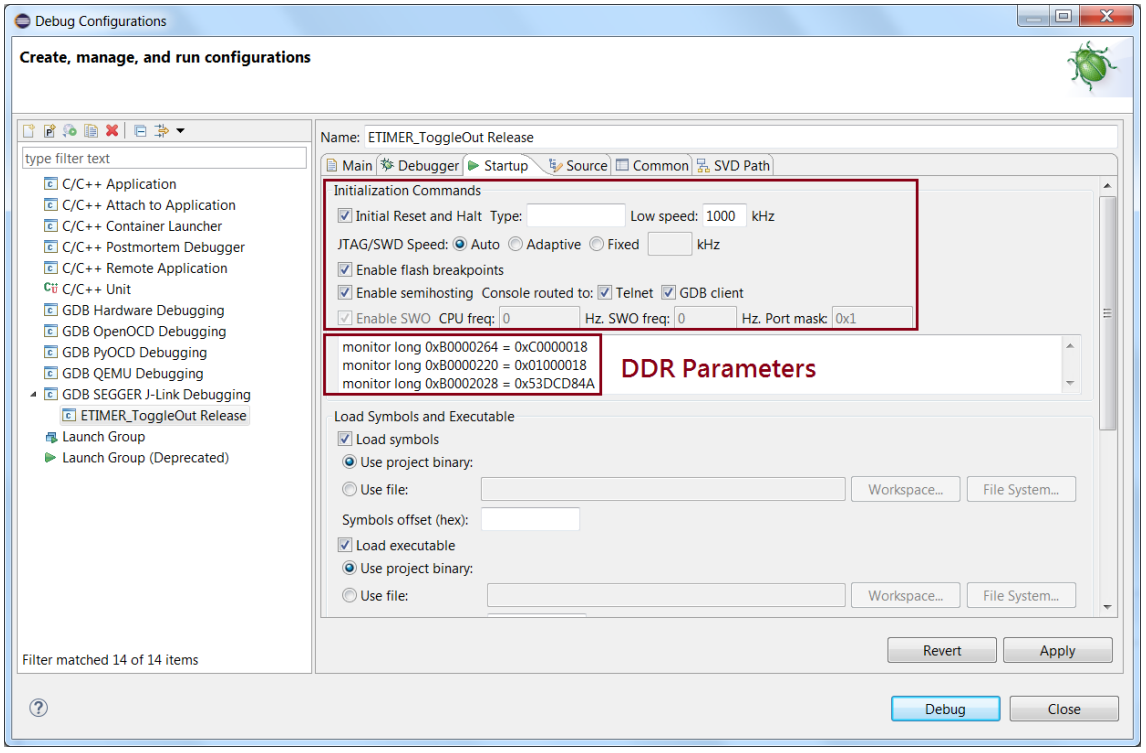

Figure 3-12 J-Link Startup Setting

After completing the setting, click “**Debug**” button to start debug with J-Link.

4 NuWriter

NuWriter can download images to NAND Flash, SPI Flash, eMMC/SD, or DDR while the NUC980 is in USB ISP mode. Please refer to *NUC980 NuWriter User Manual* for the usage of NuWriter.

5 Revision History

Date	Revision	Description
2018.09.25	1.00	1. Initially issued.
2019.06.06	1.01	1. Added Eclipse IDE description.
2022.08.16	1.02	1. Modify Eclipse installation gudie

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.