N329 Series Non-OS BSP User Manual

Usage Introduction for 32-bit MPU Family

Document Information

Abstract	This document introduces the usage of Keil and Eclipse development environment and Non-OS BSP content for the N329 series microprocessor (MPU).
Apply to	N329 series.

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

www.nuvoton.com

Table of Contents

1 Introduction to N329 Non-OS BSP

This BSP supports Nuvoton N329 series 32-bit microprocessor (MPU) embedded with an outstanding CPU core ARM926EJ-S. The N3290 series runs up to 200 MHz, and the N3292 series runs up to 240 MHz, both with 8 KB I-cache, 8 KB D-cache and MMU, 8 KB embedded SRAM and 16 KB IBR (Internal Boot ROM) for booting from NAND, SD and SPI Flash.

The N3290 series is integrated with JPEG codec, BitBLT accelerator, 32-channel SPU (Sound Processing Unit), USB2.0 HS Device, ADC, DAC and TV encoder for meeting various kinds of application needs.

The N3292 series is integrated with video codec (H.264), JPEG codec, 32-channel SPU (Sound Processing Unit), AAC accelerator, SDIO host controller, USB2.0 HS Host/Device, ADC, DAC and TV encoder for meeting various kinds of application needs.

This non-OS BSP contains peripheral's sample code with Keil and Eclipse project for getting started easily.

1.1 Keil Development Environment

The Non-OS BSP supports using Keil as the development environment, and uses J-Link ICE for debugging. The IDE does not belong to the content of this document. Please refer to Keil official website http://www.keil.com/ for the *Keil IDE User Manual*.

The N329 series supports J-TAG debug interface. This interface can be used to download programs to DRAM and debug. It is recommended to boot on recovery mode for ICE debugging.

1.2 Eclipse Development Environment

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/, and select proper version according to your operating system. Since Eclipse is a Java based application, please download JRE from Java website and install it.

The cross compile - GNU ARM Embedded Toolchain can be downloaded from https://gnu-mcu-eclipse.github.io/plugins/install/. After installing the software packages mentioned above, follow the steps below.

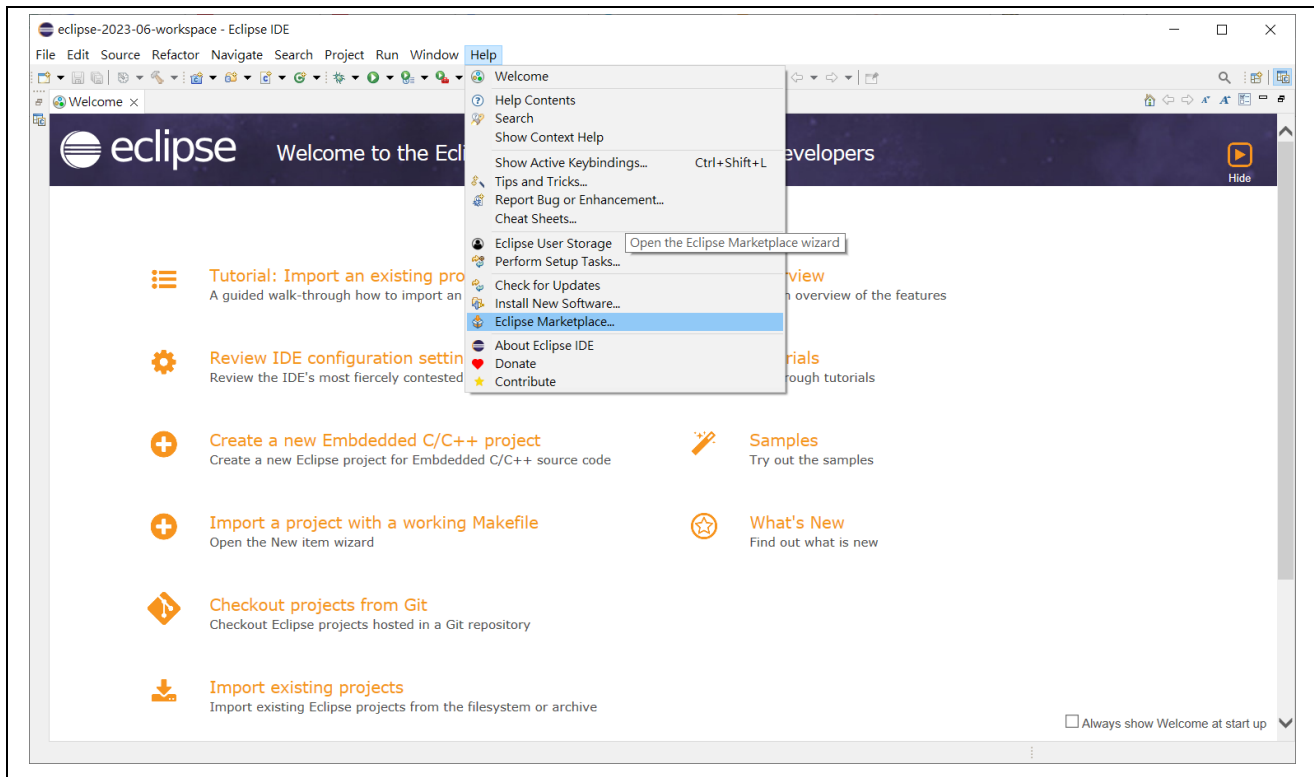1.   Execute Eclipse and select **Help** -> **Eclipse Marketplace**.

Figure 1-1 Select Eclipse Marketplace

2. Input "gnu mcu**"** in the **Find** field, and then the search result will be shown as Figure 1-2. If there is a newer version found, click the **Install** button to install the required plug-in.
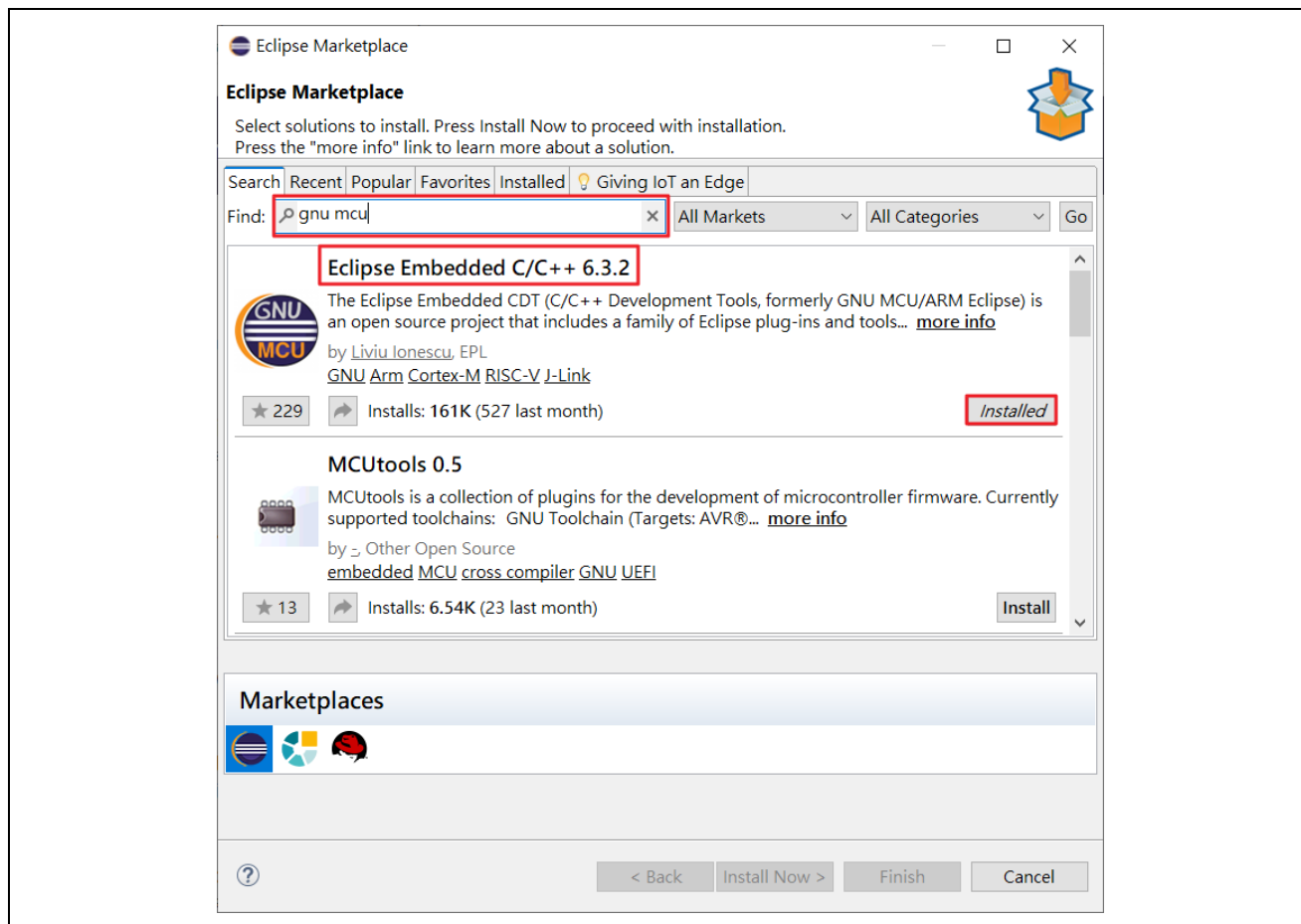
Figure 1-2 Install Plug-in

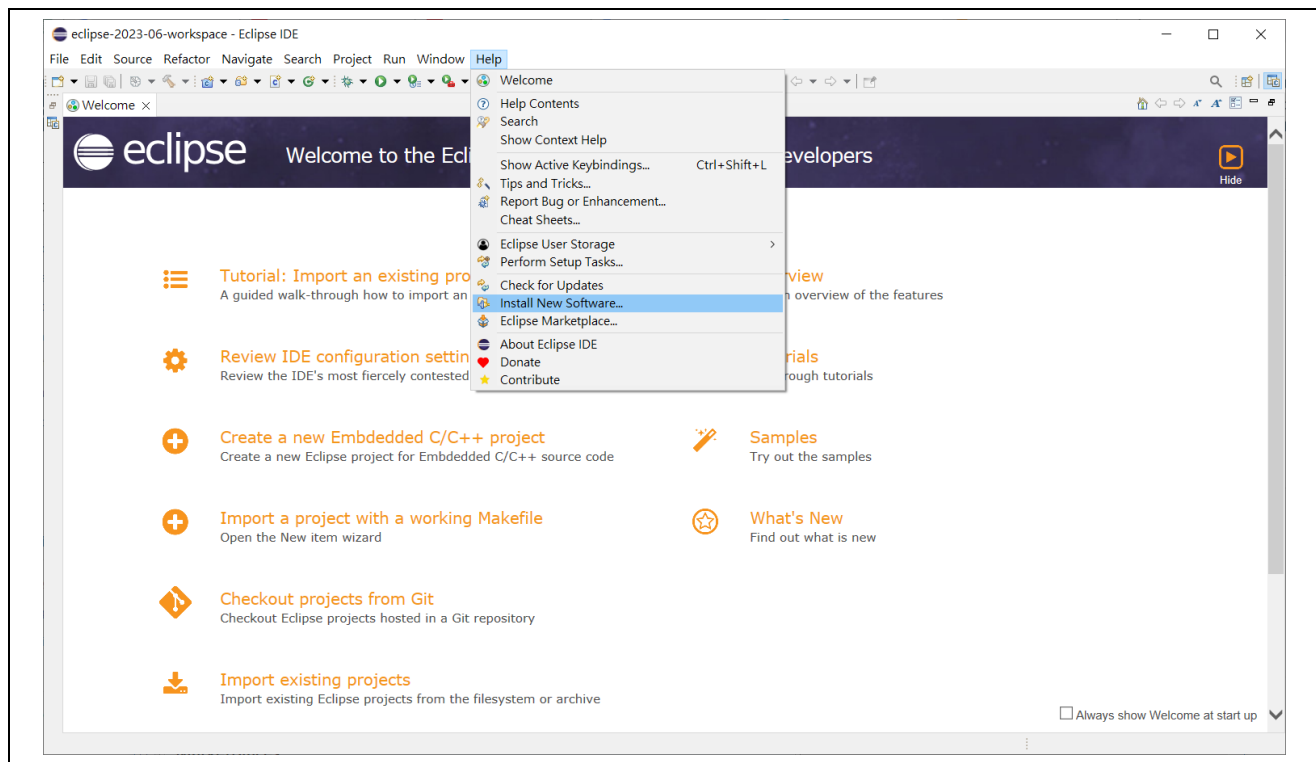3. Click **Help** -> **Install New Software** to install CDT to support C/C++ development.

Figure 1-3 Install New Software

4.　Input "CDT**"** in the **Work with** field, you can select the specified or latest version.
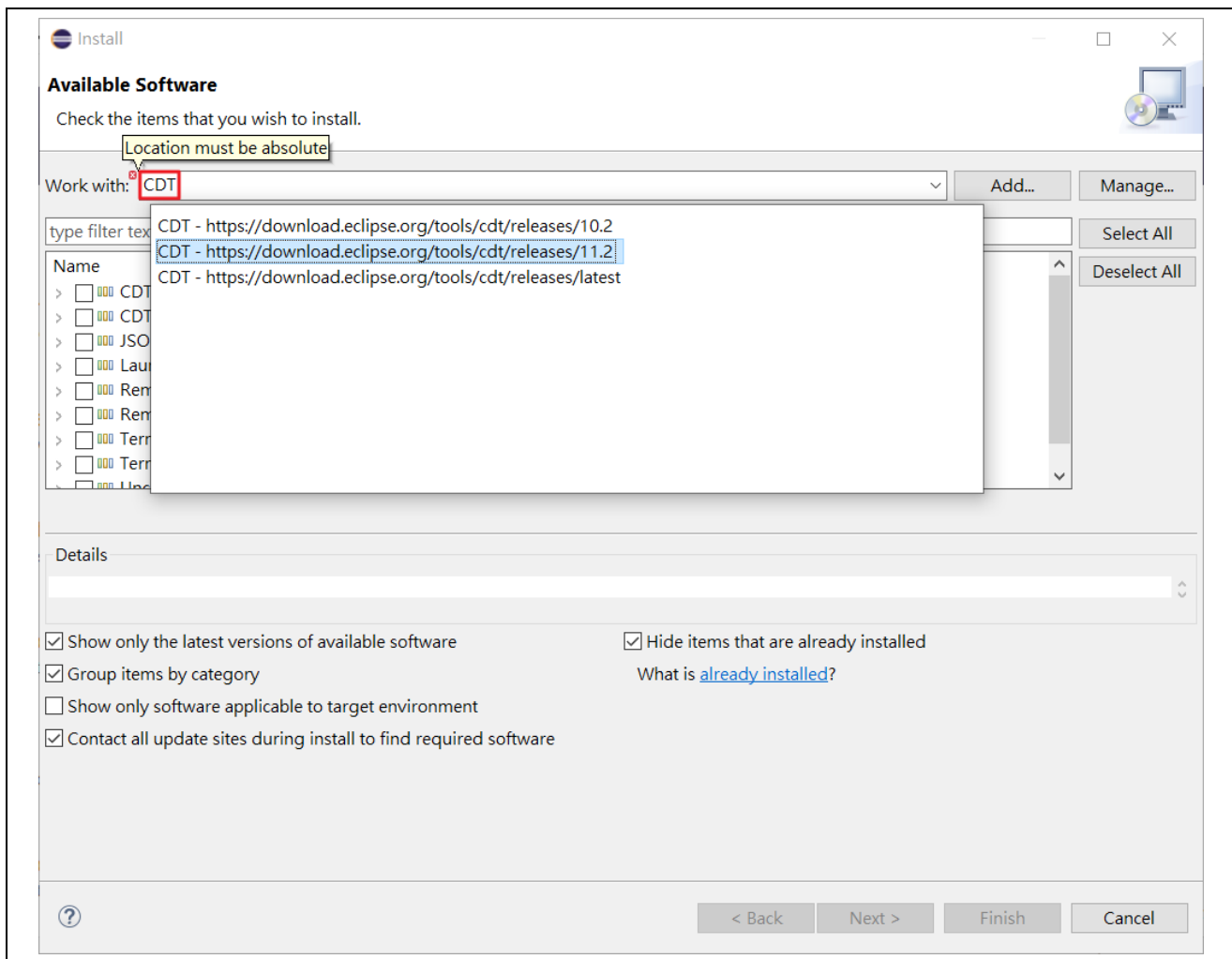
Figure 1-4 Search for CDT

5. Select **CDT Main Features** and **CDT Optional Features** as shown in Figure 1-5. Also select other pakcages if necessary.

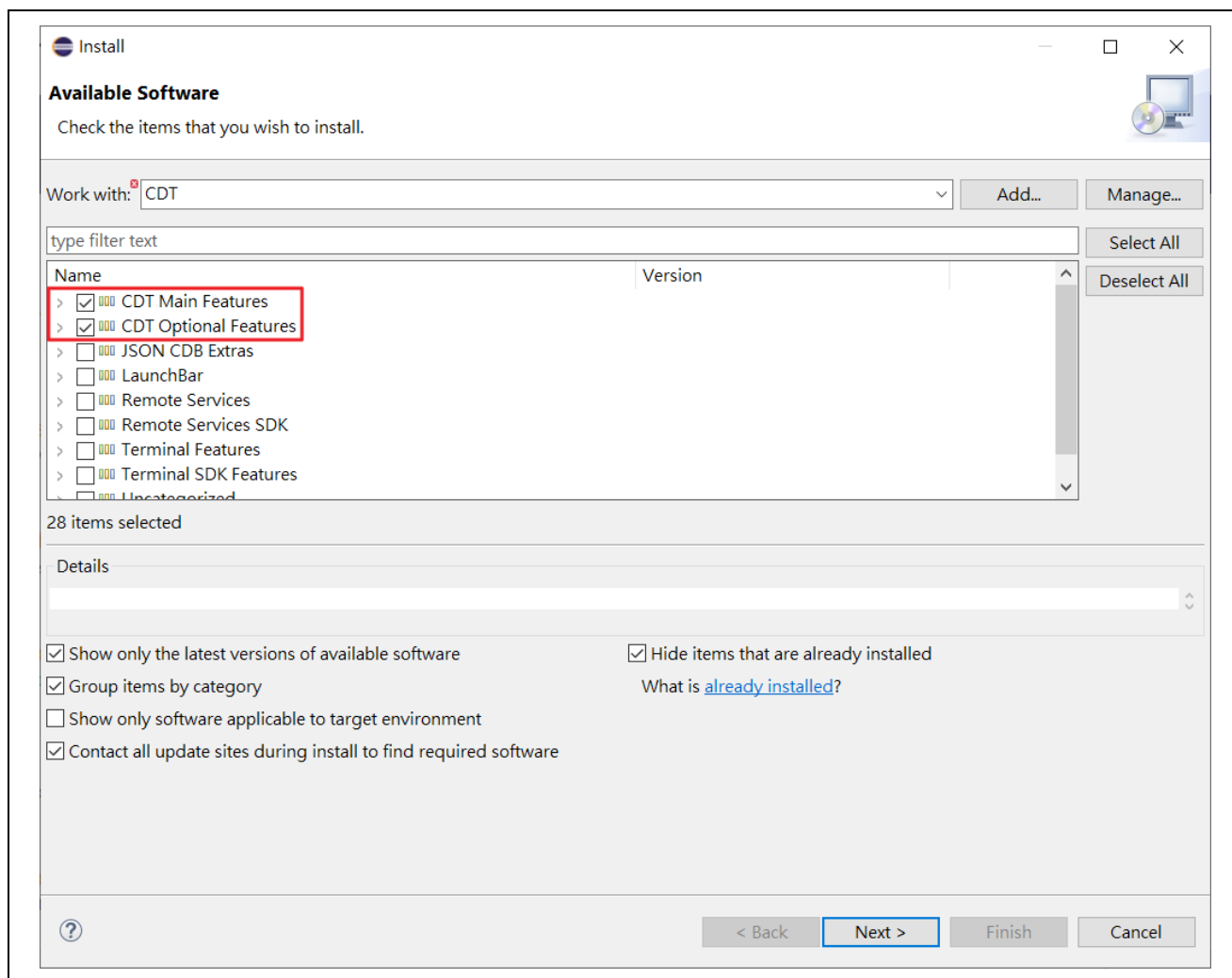Figure 1-5 Select CDT

6. After installing CDT, re-start Eclipse to import Eclipse project. The 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 J-Link path in **Windows -> Preference** -> **MCU** -> **Global SEGGER J-Link**, and then click **Apply** button.
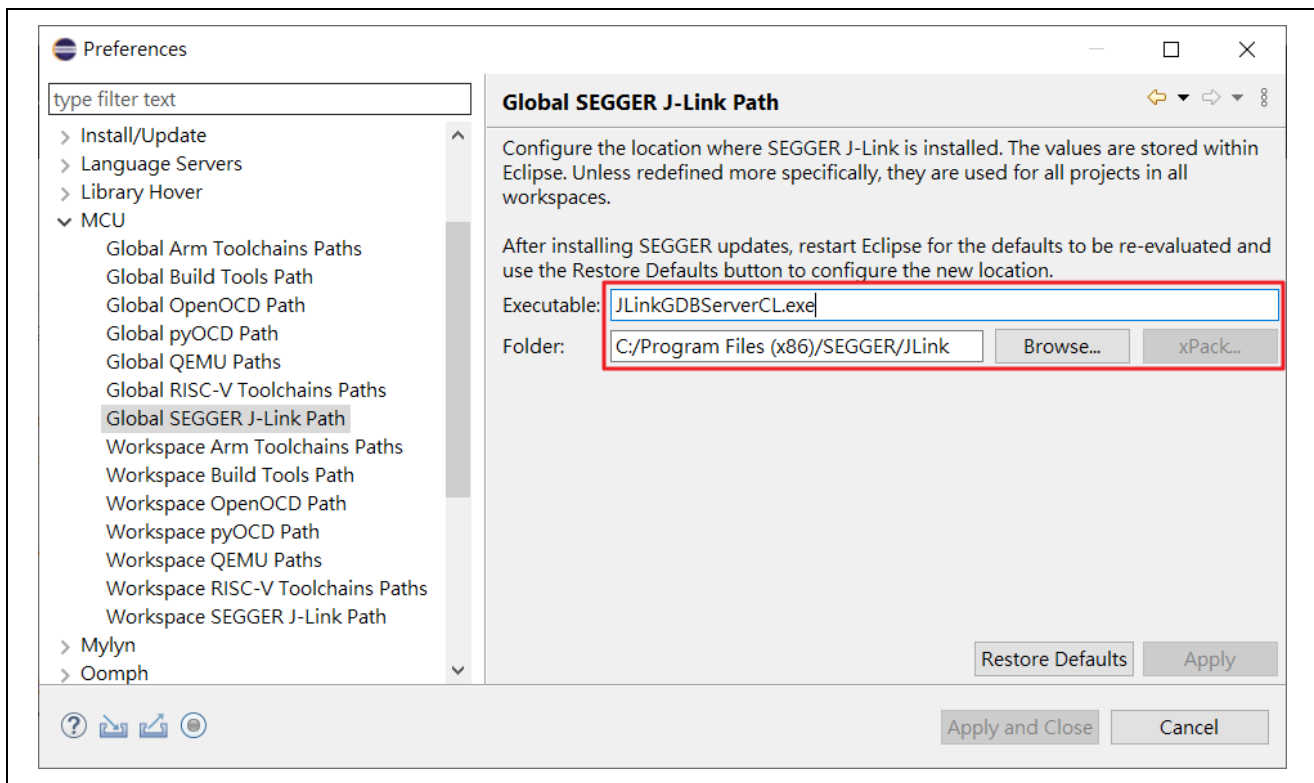
Figure 1-6 Global SEGGER J-Link Path Setting

7. Set GDB SEGGER J-Link Debugging options. Click **Run**-> **Debug Configurations** and then expand **GDB SEGGER J-Link Debugging** configuration.

Figure 1-7 GDB SEGGER J-Link Debug

8. Select the corresponding configuration matched with your device. Go to **Debugger** tab, as shown in Figure 1-8.

Figure 1-8 J-Link Debugger Setting

9. Go to **Startup** tab, as shown in Figure 1-9.

Figure 1-9 J-Link Startup Setting

10. After the setting is complete, click **Debug** button to start debugging with J-Link.

2 BSP Content
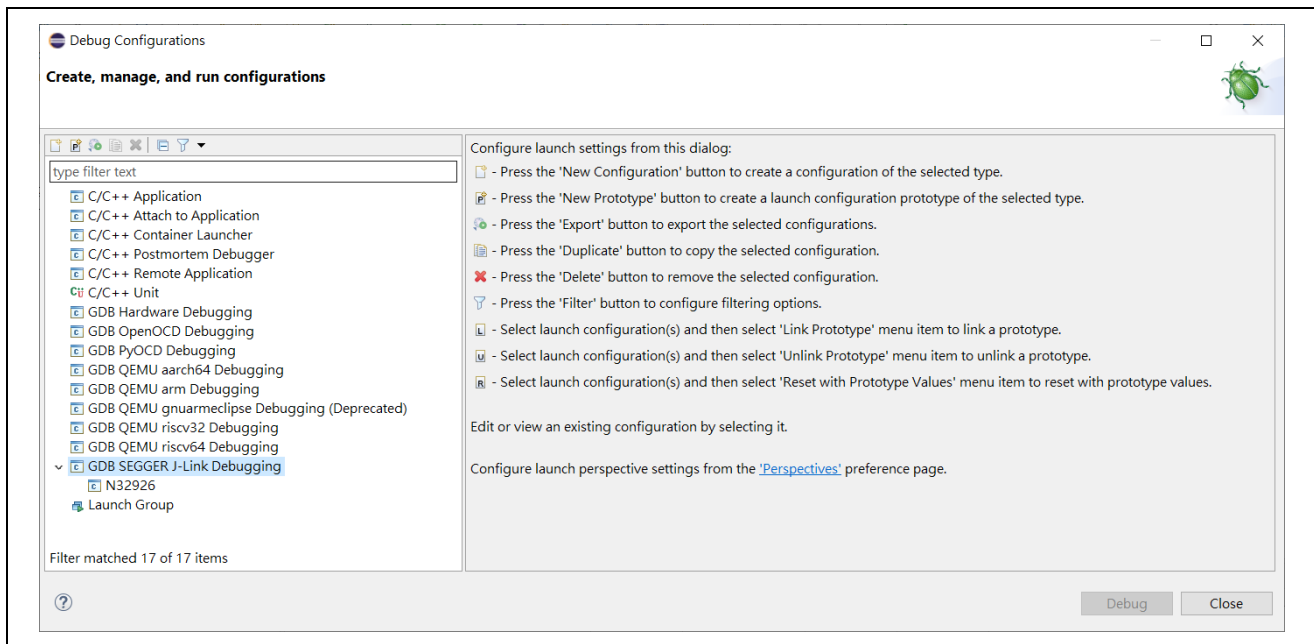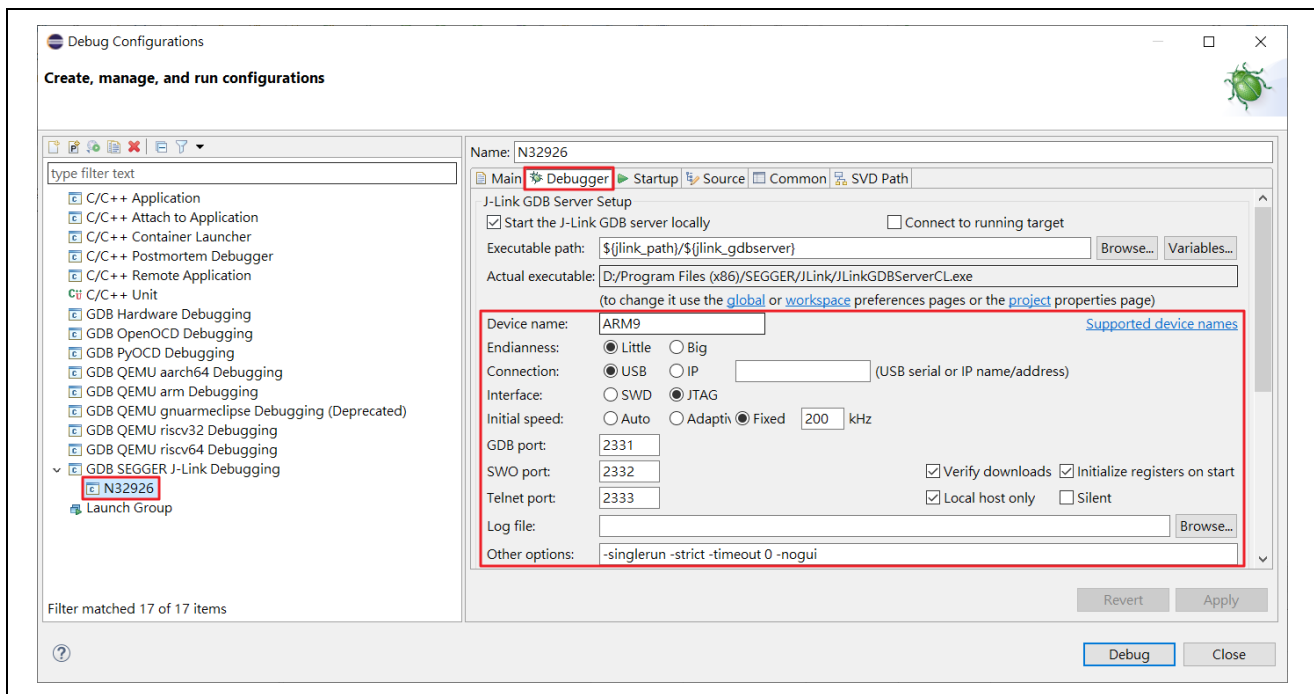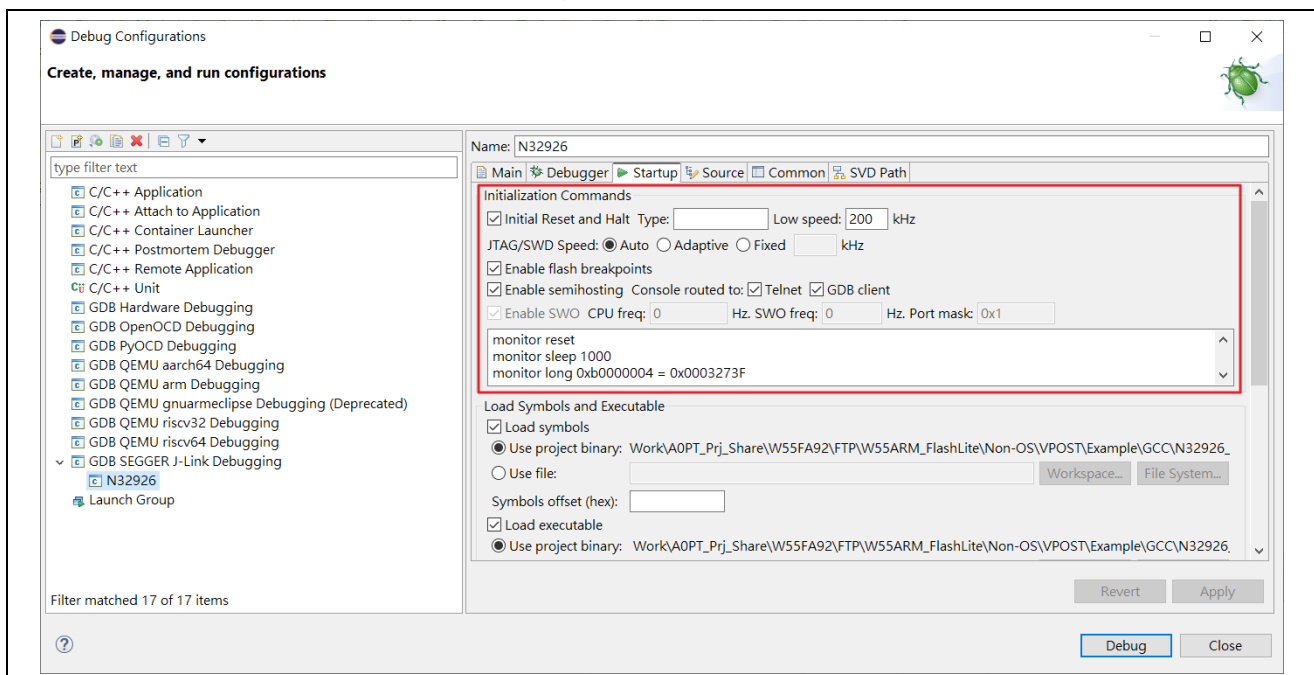
2.1 Non-OS BSP Directory Structure

The Non-OS BSP contains four directories. The content of each directory is listed below.

Directory Name	Content
AVI	The AVI player library which supports JPEG decode and audio decode.
Doc	BSP related documents.
FreeRTOS	The source code and samples of the FreeRTOS, which is a real-time operating system for microcontrollers.
GCC	Linker script and startup files for GCC project.
GNAND	The library to make NAND device as a disk.
Loader/NVTLoader	Contain source code and binary file for different loader in booting path.
N329xx	Linker script and debug initialization files for Keil project.
[NAND/SD/SPI]Writer	The tool can copy the prepared data to NAND/SD/SPI storage.
NVTFAT	The library of Nuvoton FAT file system.
[Peripheral IP]	Contain source code and library file for N329 peripheral drivers, such as VPOST, SPU, GPIO, I^2C and SPI.
SYSLIB	The System library to control on-chip functions, such as Timers, UARTs, AIC, Cache and power management.
USB_Core	The library of USB host. The N3290 series has USB 1.1 only, and the N3292 series has USB 1.1 and USB 2.0.

2.2 Peripheral Content

The content of each [Peripheral IP] directory is listed below.

Directory Name	Content
Example	Keil/GCC project and source file for sample application.

Lib	Driver library and header files.
Src	Keil/GCC project and source file for driver.

2.3 Loaders Content

The booting sequence is **IBR → Loader → NVTLoader (option)**. These folders provide the reference sample code for these Loaders. In most cases, this code does not need to be modified.

Directory Name	Content
Loader\NandLoader	Source code of NANDLoader. Please refer to *FA92/FA93 NANDLoader Reference Guide* under Documents directory for the usage.
Loader\SDLoader	Source code of SDLoader. Please refer to *FA92/FA93 SDLoader Reference Guide* under Documents directory for the usage.
Loader\SpiLoader	Source code of SPILoader. Please refer to *FA92/FA93 SPILoader Reference Guide* under Documents directory for the usage.
Loader\SpiLoader_gzip	Source code of SPILoader with gzip.
NVTLoader	Source code of NVTLoader. Please refer to *FA92/FA93 NVTLoader Reference Guide* under Documents directory for the usage.

2.4 Script Content

The booting sequence is **IBR → Loader → NVTLoader (option)**. These folders provide the reference sample code for these Loaders. In most cases, this code does not need to be modified.

Directory Name	Content
N32901	• N32901.ini is for peripheral usage example, which are executed at the address 0x0. • N32901_loader.ini is for NANDLoader, SDLoader and SPILoader under Loaders folder, which are executed at the address 0x180000. • N32901_NVTloader.ini is for NVTLoader, which are executed at the address 0x60000.
N32903	• N32903.ini is for peripheral usage example, which are executed at the address 0x0. • N32903_loader.ini is for NANDLoader, SDLoader and SPILoader under Loaders folder, which are executed at the

	address 0x700000. ● N32903_NVTloader.ini is for NVTLoader, which are executed at the address 0x600000.
N32905	● N32905.ini is for peripheral usage example, which are executed at the address 0x0. ● N32905_NVTloader.ini is for NVTLoader, which are executed at the address 0x800000.
N32926	● N32926.ini is for peripheral usage example, which are executed at the address 0x0. ● N32926_NVTloader.ini is for NVTLoader, which are executed at the address 0x800000 address.

2.5 Writers Content

Prepare the content you want to program on storage on SDx in advance. This tool can copy the prepared data to NAND/SD/SPI through SD booting.

Directory Name	Description
NANDWriter	Please refer to *NandWriter User Guide* under *NandWriter\Doc* folder for details.
SDWriter	Please refer to *SDWriter User Guide* under *SDWriter\Doc* folder for details.
SPIWriter	Please refer to *SpiWriter User Guide* under *SpiWriter\Doc* folder for details.

Revision History

Date	Revision	Description
2023.07.19	1.00	● Initial version.

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.