

N9H20 Non-OS BSP User Guide

Usage Introduction for 32-bit MPU Family

Document Information

Abstract	Introduce the usage of Keil and Eclipse development environment and Non-OS BSP content for the N9H20 series microprocessor (MPU).
Apply to	N9H20 series MPU.

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	INTRODUCTION TO N9H20 NON-OS BSP	3
1.1	Keil Development Environment.....	3
1.2	Eclipse Development Environment	3
2	BSP CONTENT	10
2.1	Non-OS BSP Directory Structure	10
2.2	BSP Content	10
2.3	Loader Content.....	11
2.4	Tools Content	11

1 Introduction to N9H20 Non-OS BSP

This BSP supports Nuvoton N9H20 series HMI-application 32-bit microprocessor (MPU) embedded with an outstanding CPU core ARM926EJ-S. The N9H20 series runs up to 200 MHz, 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 N9H20 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.

This non-OS BSP contains IPs 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 official Keil website <http://www.keil.com/> for the user manual of Keil IDE.

The N9H20 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/>.

1. After installing the software packages mentioned above, execute Eclipse and select **Help** -> **Eclipse Marketplace**.

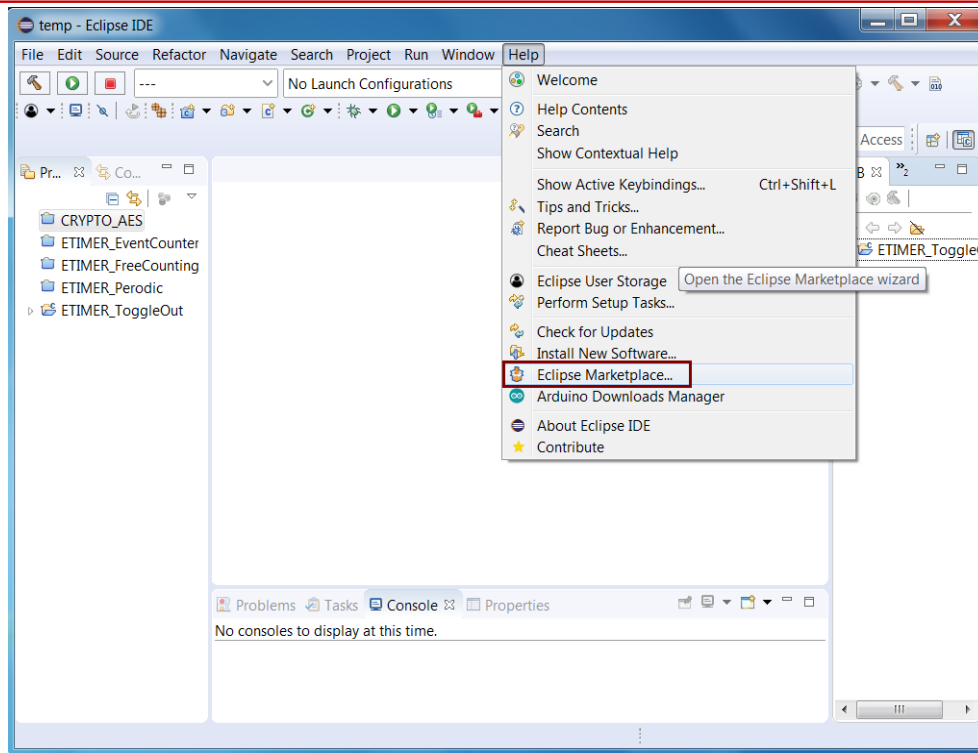


Figure 1-1 Select Eclipse Marketplace

- Input “gnu mcu eclipse” in the **Find** field, and then the search result will be shown as Figure 1-2. Select the latest version and click the **Installed** button to install the required plug-in.

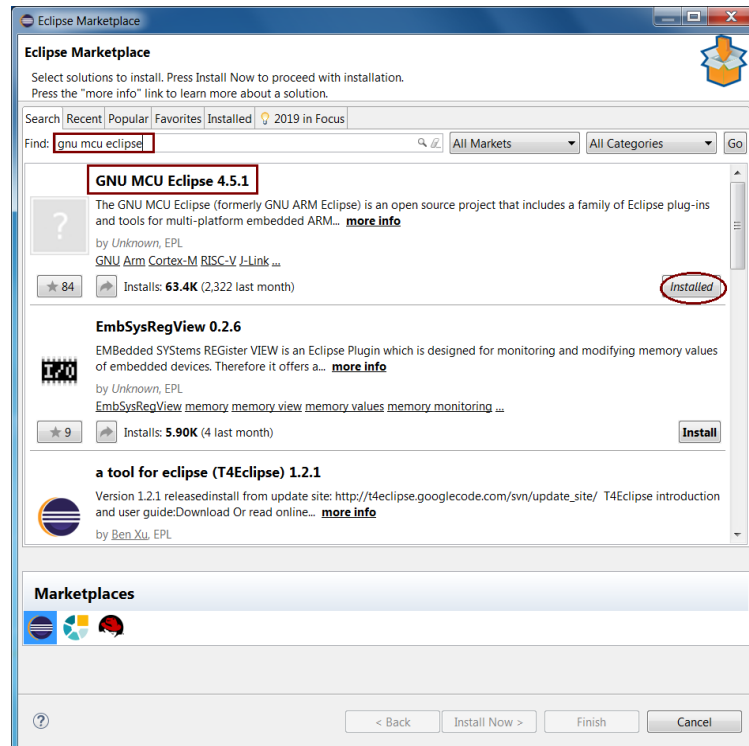


Figure 1-2 Install Plug-in

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

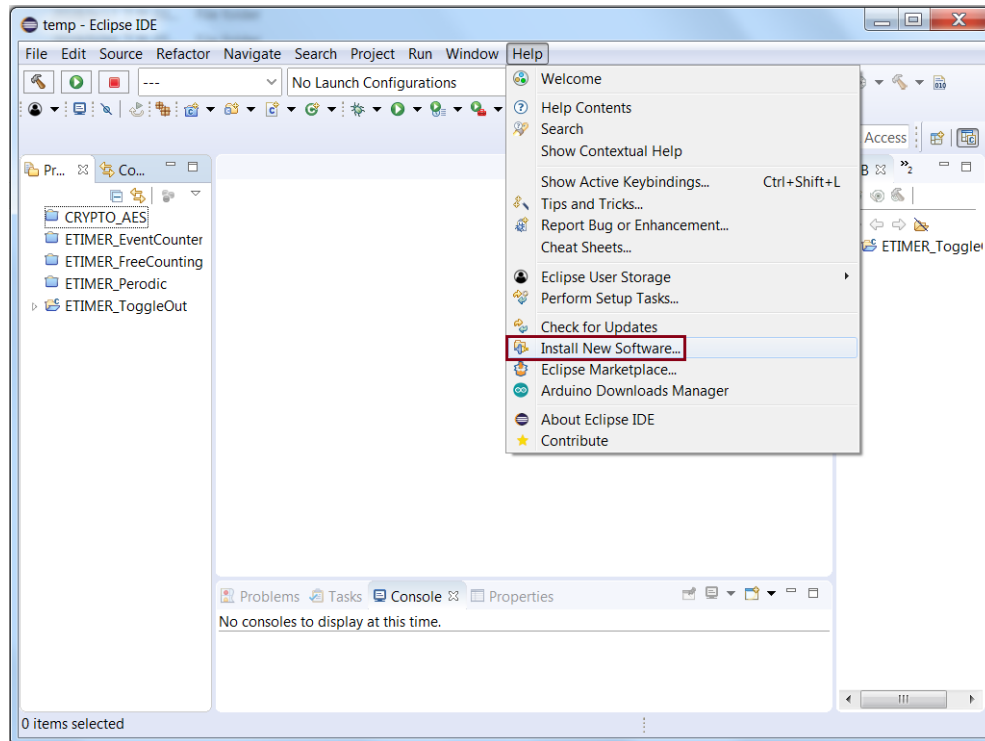


Figure 1-3 Install New Software

- Input "CDT" in the **Work with** field.

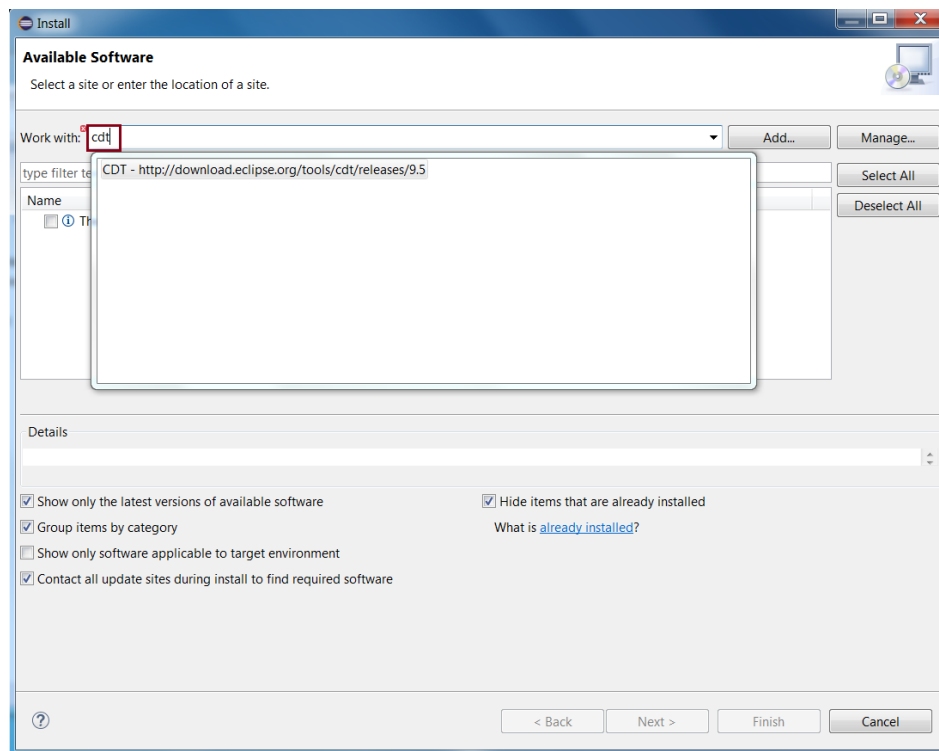


Figure 1-4 Search for CDT

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

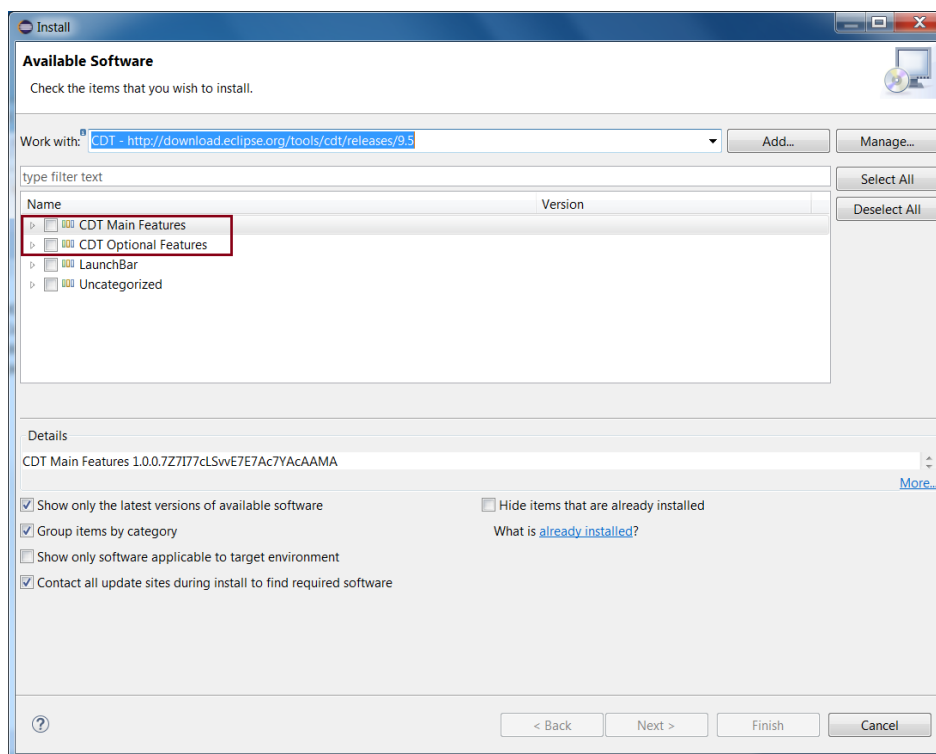


Figure 1-5 Select CDT

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://gnuarmeeclipse.github.io/plugins/install/> before starting debugging. After installation, set J-Link path in **Preference->MCU-> Global SEGGER J-Link**, and then click **Apply** button.

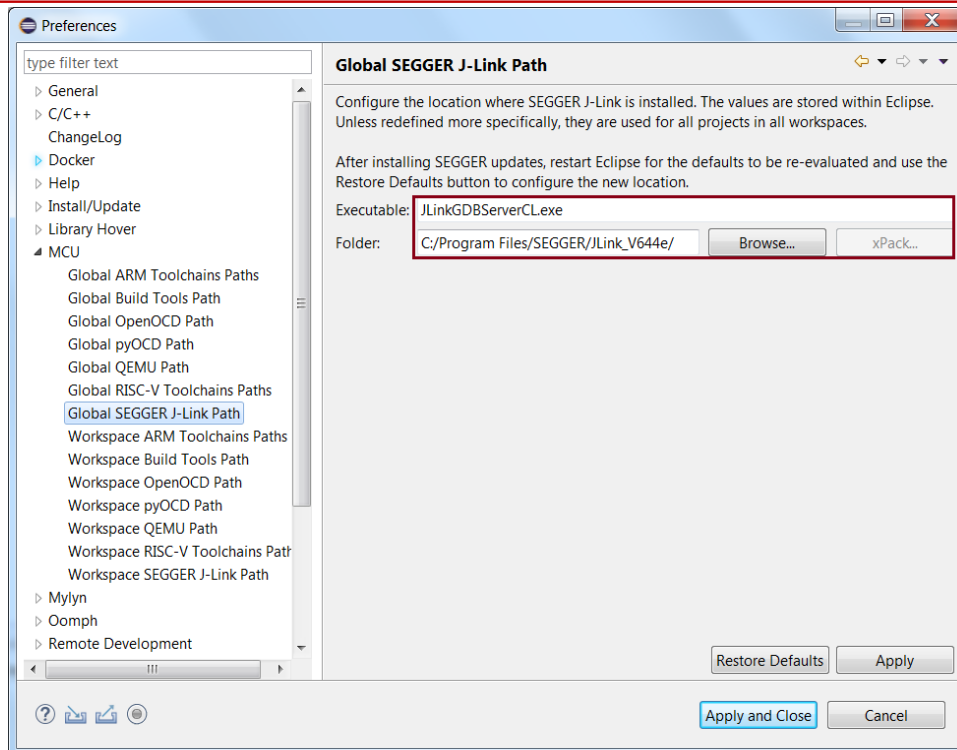


Figure 1-6 Global SEGGER J-Link Path Setting

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

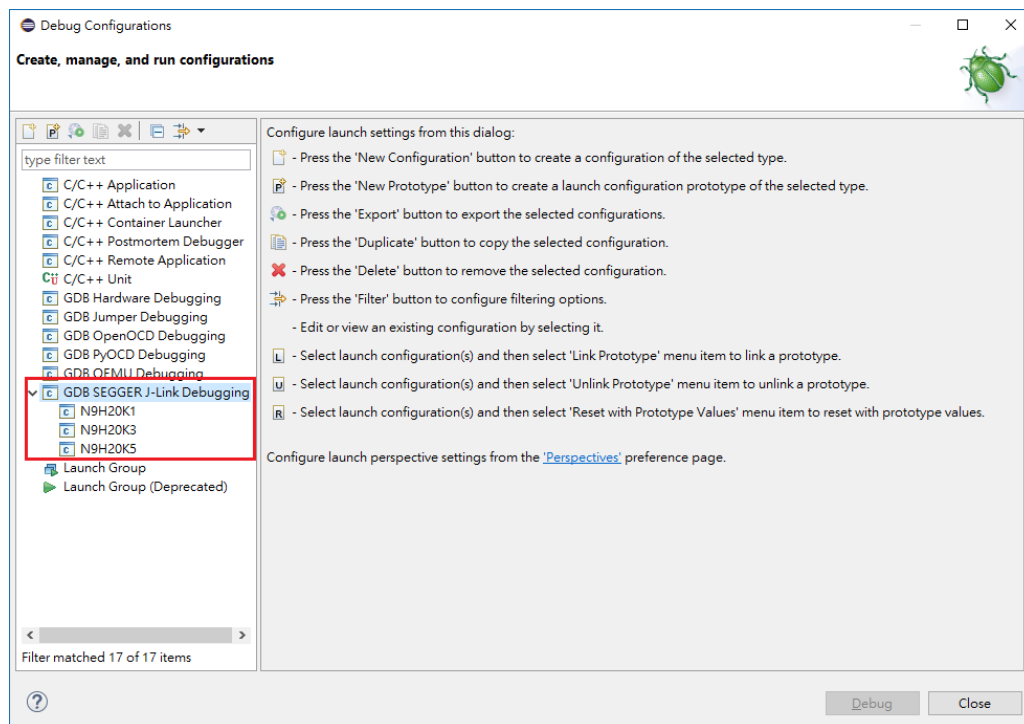


Figure 1-7 GDB SEGGER J-Link Debug

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

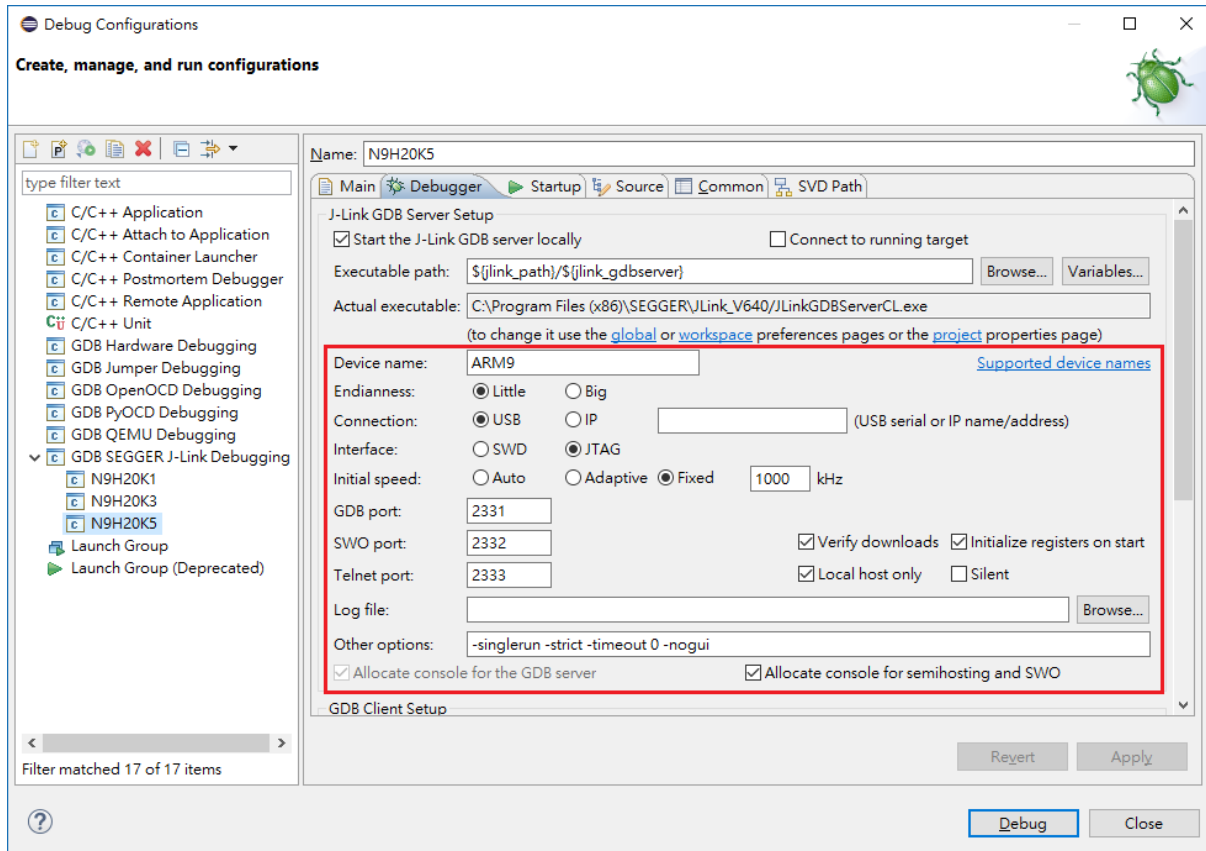


Figure 1-8 J-Link Debugger Setting

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

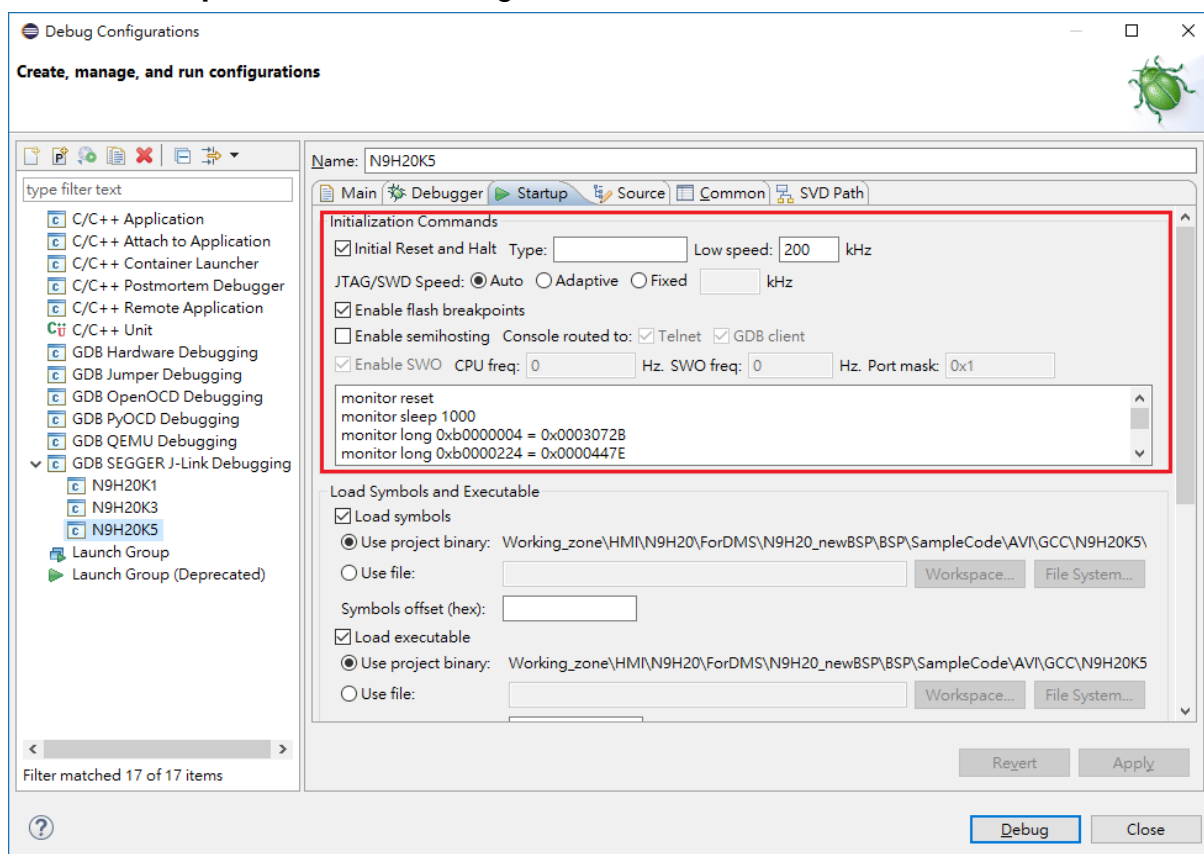


Figure 1-9 J-Link Startup Setting

- 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
BSP	Folder contains Non-OS driver, third party software and sample applications.
Documents	BSP related documents.
Loader	Contain source code and binary file for different loader in booting path.
Tools	Tools include PC tools and the Mass production tools.

2.2 BSP Content

The content of each BSP directory is listed below.

Directory Name	Content
Driver	N9H20 peripheral drivers. Please refer to N9H20 Non-OS Library Reference Guide.pdf under Documents directory for the usage of driver APIs.
Library	N9H20 Keil libraries, including IP, AVI, GNAND, NVTFAT and USB Host. The IP library source code can be found in <i>Driver\Source</i> folder.
Library_GCC	N9H20 GCC libraries, including IP, AVI, GNAND, NVTFAT and USB Host. The IP library source code can be found in <i>Driver\Source</i> folder.
SampleCode	Driver sample application.
Script	Link script and debug initialization file for Keil.
ThirdParty	Third party software, including emWin and FreeRTOS.

The folders under Script are for N9H20 different part number IC as below.

Directory Name	Description
N9H20K1	<ul style="list-style-type: none"> N9H20K1.ini is for IP usage example, which are executed at the address 0x0. N9H20K1_loader.ini is for NandLoader, SDLoader and SPILoader under LOADER folder, which are executed at the

	address 0x180000. ● N9H20K1_NVTloader.ini is for NVT LOADER, which are executed at the address 0x60000.
N9H20K3	● N9H20K3.ini is for IP usage example, which are executed at the address 0x0. ● N9H20K3_loader.ini is for NandLoader, SDLoader and SPILoader under LOADER folder, which are executed at the address 0x700000. ● N9H20K3_NVTloader.ini is for NVT LOADER, which are executed at the address 0x600000.
N9H20K5	● N9H20K5.ini is for IP usage example, which are executed at the address 0x0. ● N9H20K5_loader.ini is for NandLoader, SDLoader and SPILoader under LOADER folder, which are executed at the address 0x900000. ● N9H20K5_NVTloader.ini is for NVT LOADER, which are executed at the address 0x800000.

2.3 Loader 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
Binary	Contains the pre-build binary file for NandLoader, SDLoader, SPILoader and NVTLoader.
NANDLoader	Source code of NandLoader. Please refer to N9H20 NAND Loader Reference Guide.pdf under Documents directory for the usage.
NVTLoader	Source code of NVTLoader. Please refer to N9H20 NVTLoader Reference Guide.pdf under Documents directory for the usage.
SDLoader	Source code of SDLoader. Please refer to N9H20 SD Loader Reference Guide.pdf under Documents directory for the usage.
SPILoader	Source code of SPI Loader. Please refer to N9H20 SPI Loader Reference Guide.pdf under Documents directory for the usage.
SPILoader_gzip	Source code of SPI Loader with gzip. Please refer to N9H20 SPI Loader Reference Guide.pdf under Documents directory for the usage.

2.4 Tools Content

This directory contains PC_tools and MassProduction_tools.

PC_tools:

Directory Name	Description
AutoWriter V3.xx.xxx_N9H20Kx	Please refer to the AutoWriter User Guide.pdf for details.
TurboWriter V2.xx.xxx_N9H20Kx	Please refer to the TurboWriter Tool User Guide.pdf for details.

MassProduction_tool:

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.pdf under <i>NandWriter\Doc</i> folder for details.
NRomMaker	Please refer to NRomMakder Tool User Guide.pdf under <i>NRomMaker</i> folder for details.
SDWriter	Please refer to SDWriter User Guide.pdf under <i>SDWriter\Doc</i> folder for details.
SPIWriter	Please refer to SPIWriter User Guide.pdf under <i>SPIWriter\Doc</i> folder for details.

Revision History

Date	Revision	Description
2018.05.04	1.00.000	Initially issued.
2018.08.17	1.00.001	Minor update.
2018.09.21	1.01.000	BSP folder structure change.
2019.09.30	1.02.000	Added Eclipse IDE description.
2020.09.15	1.03.000	Added folder for Library_GCC and NRomMaker.
2020.11.10	1.04.000	Updated document format and description.

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.*