N9H30 Non-OS BSP User Manual

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

1 Introduction to N9H30 Non-OS BSP 3
1.1 Develop Environment 3
1.2 Eclipse Develop Environment 4
1.3 DEV Board Setting 15
2 BSP Content 16
2.1 BSP directory structure 16
2.2 Non-OS BSP content 16
3 NuWriter 17
4 Revision History 18

1 Introduction to N9H30 Non-OS BSP

This BSP supports Nuvoton N9H30 family processors. The N9H30 series targeted for general purpose 32-bit microcontroller 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, 56KB embedded SRAM and 16 KB IBR (Internal Boot ROM) for booting from USB, NAND and SPI FLASH.

The N9H30 series integrates USB 2.0 HS HOST/Device controller with HS transceiver embedded, TFT type LCD controller, 2D graphics engine, I2S I/F controller, SD/MMC/NAND FLASH controller, GDMA and 8 channels 12-bit ADC controller with resistance touch screen functionality. It also integrates UART, SPI/MICROWIRE, I2C, 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:

- N9H30 Non-OS drivers.
- Precompiled U-Boot images for different boot mode.
- Flash programming tool NuWriter, and its Windows driver.
- User manuals.

1.1 Develop Environment

Non-OS BSP supports using Keil as develop environment, and use ULINK2 ICE for debug. 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.

N9H30 supports J-TAG debug interface. Users could use this interface to download programs to DDR and debug. After power up, the J-TAG interface is disabled by default. To use the J-TAG interface, ether of the criteria must be met. 1) N9H30 boot up in USB ISP mode and successfully set up a connection with NuWriter tool, or 2) N9H30 does not boot up in secure boot mode and successfully execute the loader (or application) store in SPI flash, NAND flash, or eMMC. Also ICE reset must be disabled; otherwise the JTAG interface will be disabled immediately after reset. To disable ULINK2 reset, uncheck Use Reset at Startup under the Misc Options as shown below.

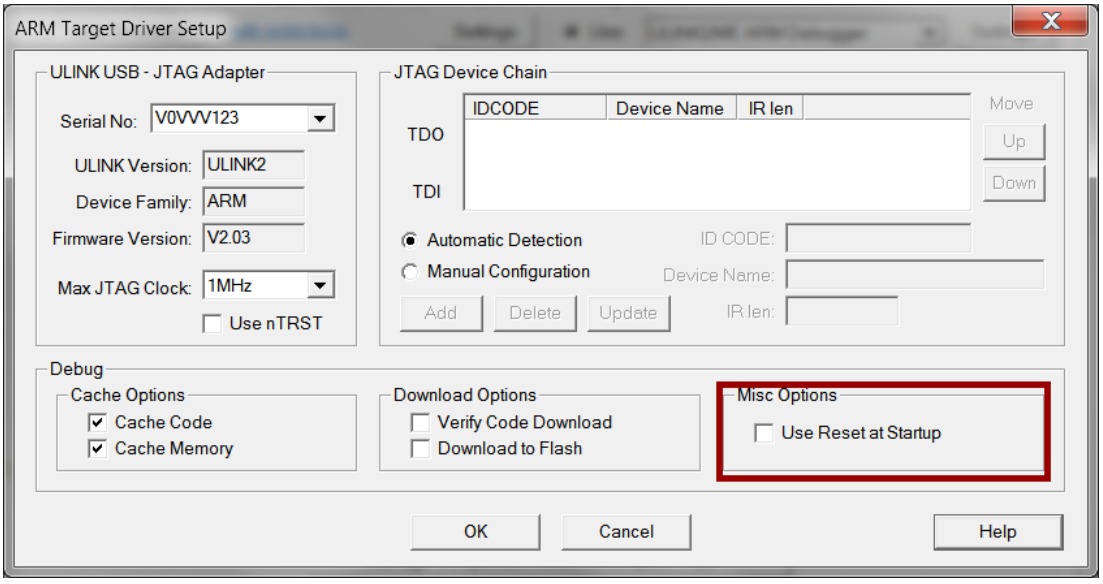

Figure 1-1 ARM Target Driver Setup

To disable J-Link reset, set Reset Strategy to No Reset as shown below.

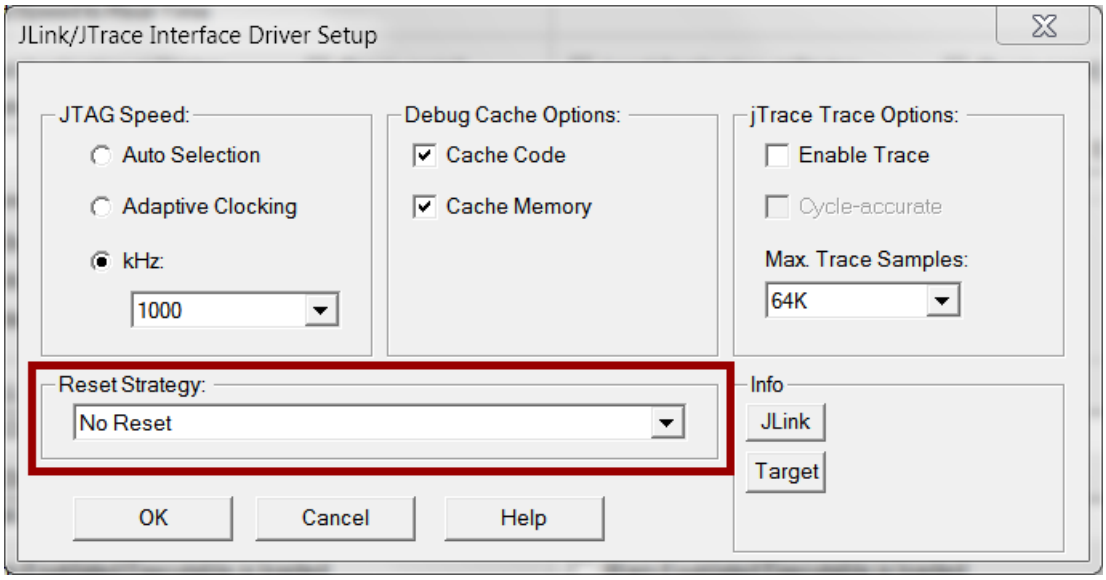

Figure 1-2 J-Link Interface Driver Setup

N9H30 Non-OS BSP uses the same open source loader as Linux BSP, U-Boot. U-Boot uses Linux as develop environment. Users could ether the precompiled U-Boot images in this BSP. If it is necessary to modify and re-build U-Boot, please download N9H30 Linux BSP and refer to N9H30 Linux BSP user manual to set up the develop environment. If the system boot from SPI flash or eMMC, 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.

1.2 Eclipse Develop Environment

The N9H30 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.

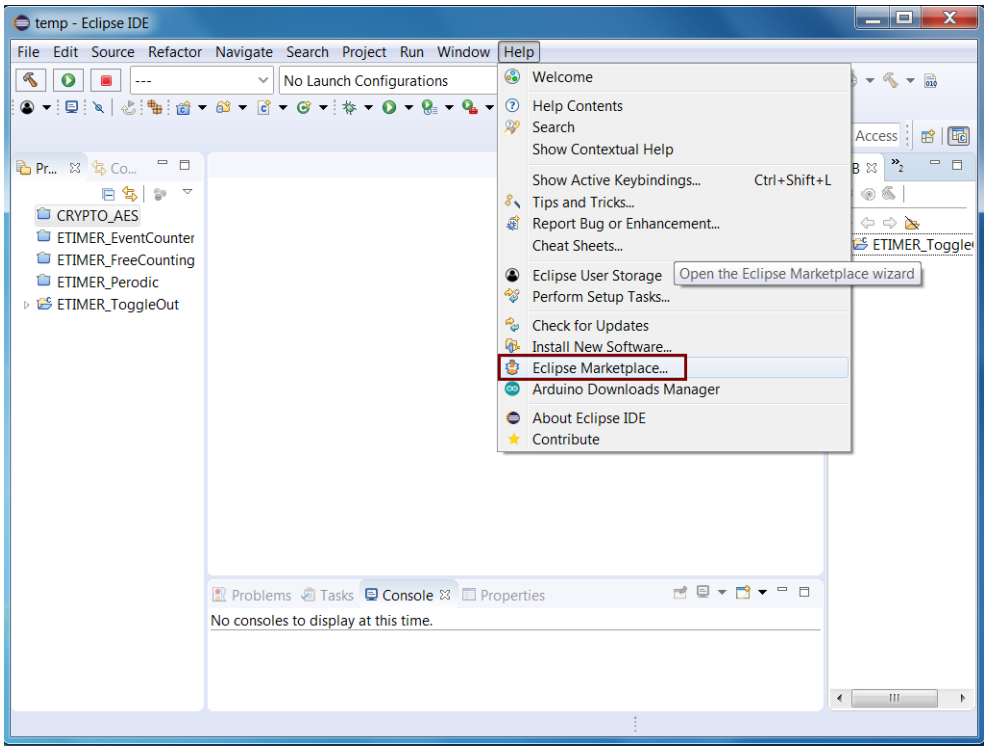

Figure 1-3 Select Eclipse Marketplace

Input gnu mcu exlipse in the Find field, and then the search result will be shown as Figure 1-4. Select the latest version and click Install button to install the required plug-in.

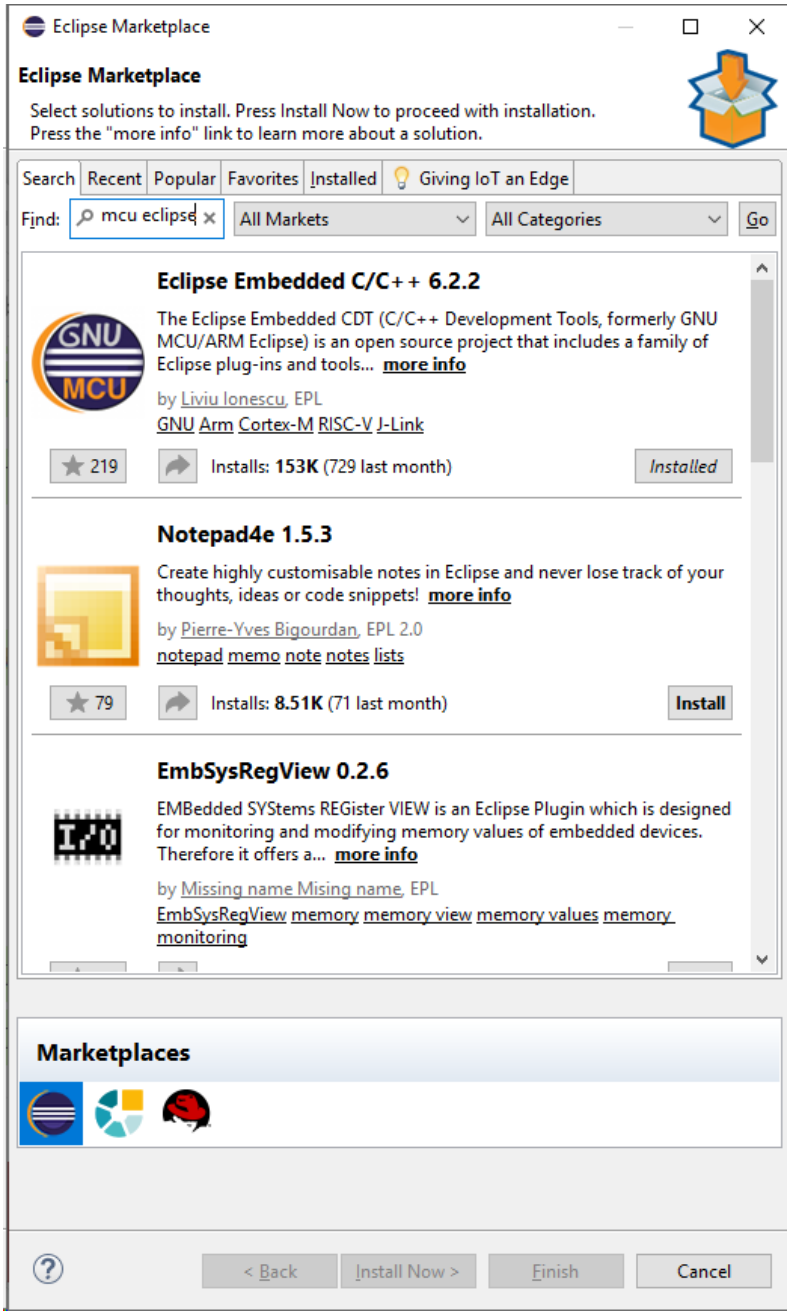

Figure 1-4 Install Plug-in

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

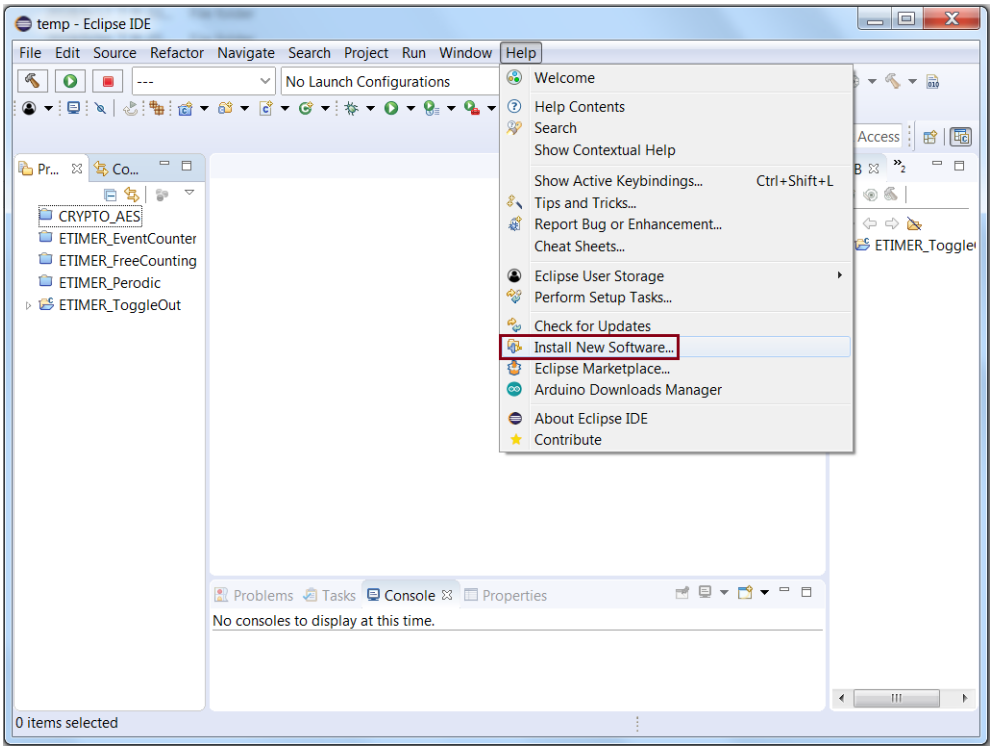

Figure 1-5 Install New Software

Input “CDT” in Work with field.

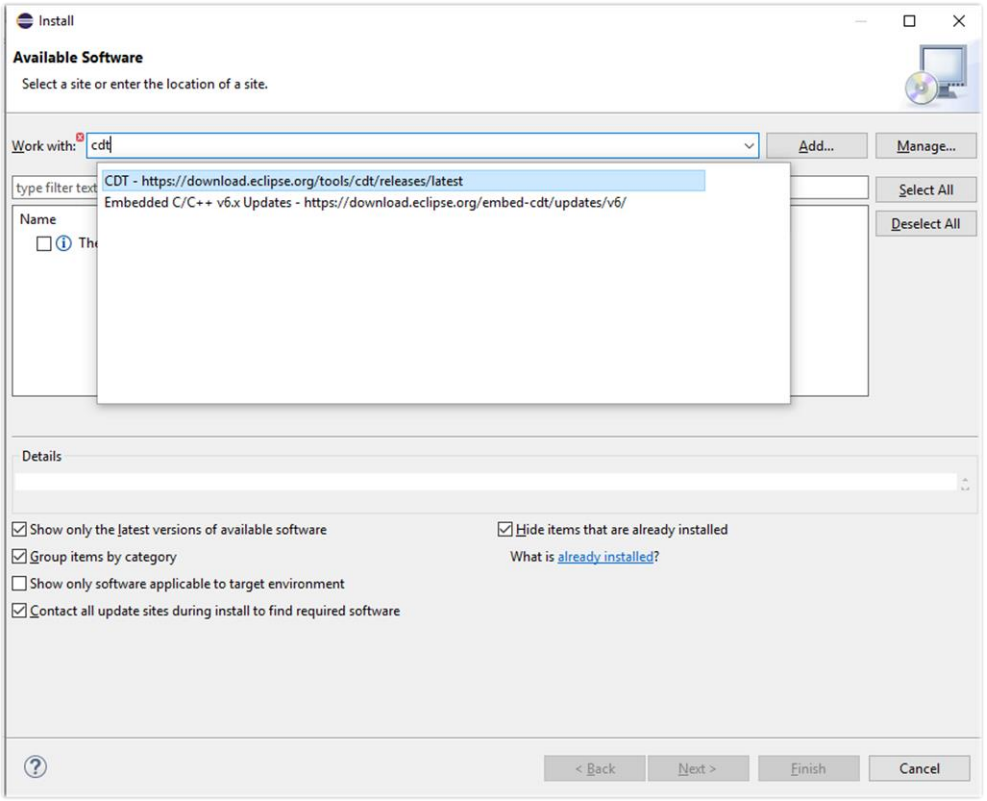

Figure 1-6 Search for CDT

Select CDT Main Features and CDT Optional Features as shown in Figure 1-7. User can also select other pakcages if necessary.

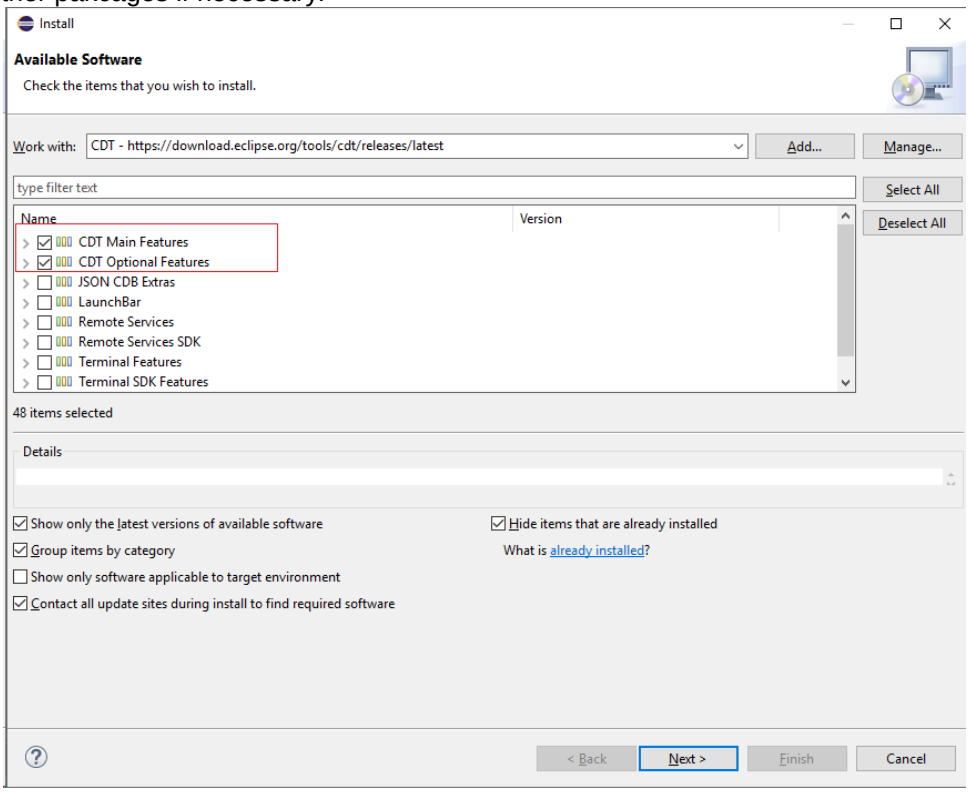

Figure 1-7 Select CDT

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 1-8.

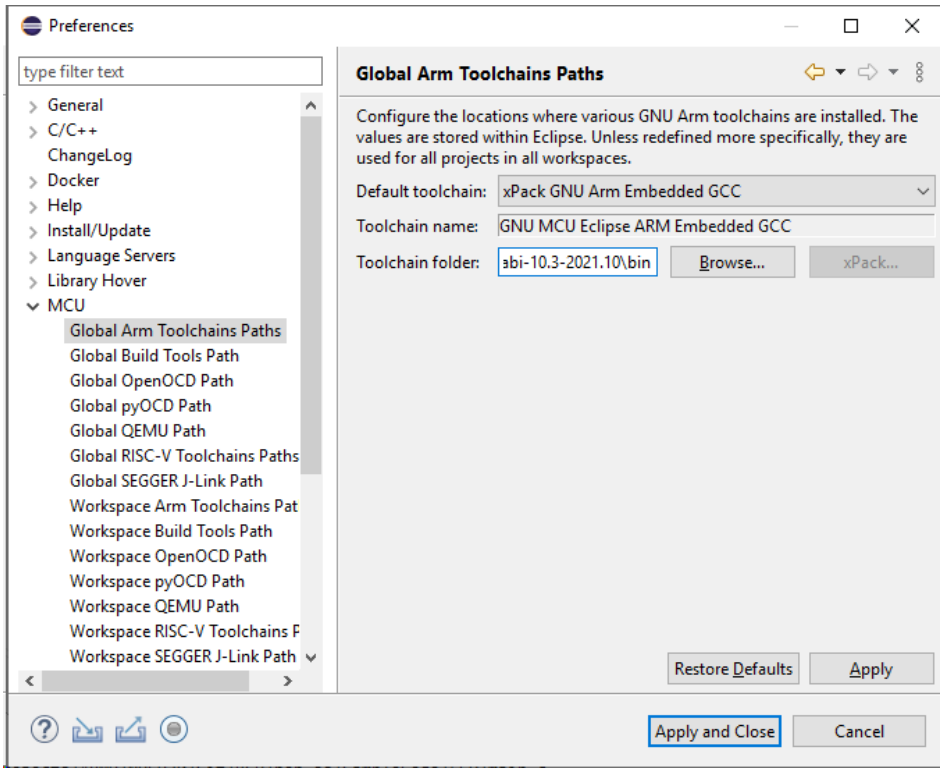

Figure 1-8 Configure tool chain path

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 1-9.

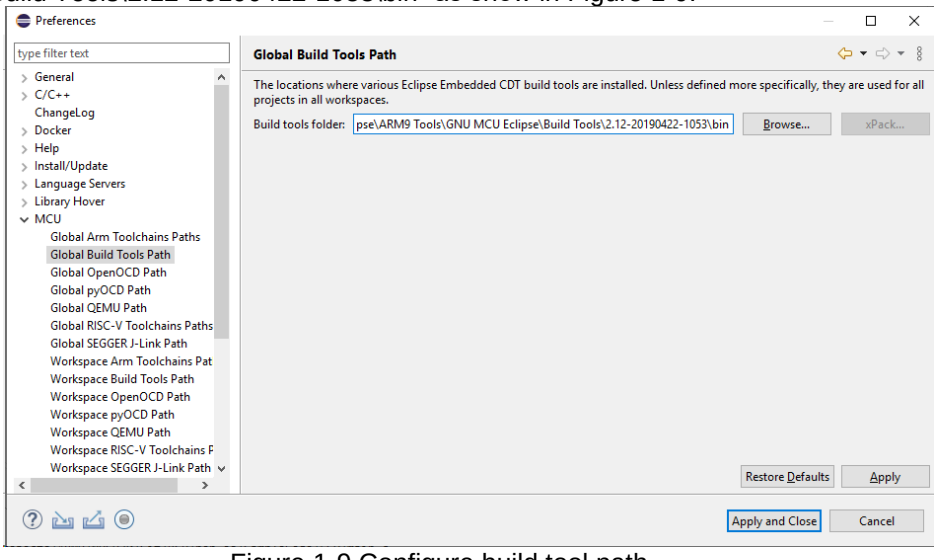

Figure 1-9 Configure build tool path

Now the Eclipse environment should be ready for compileing NUC9H30 BSP. Import an Eclipse project from File ➔ Import, the import window opend as shown in Figure 1-10.

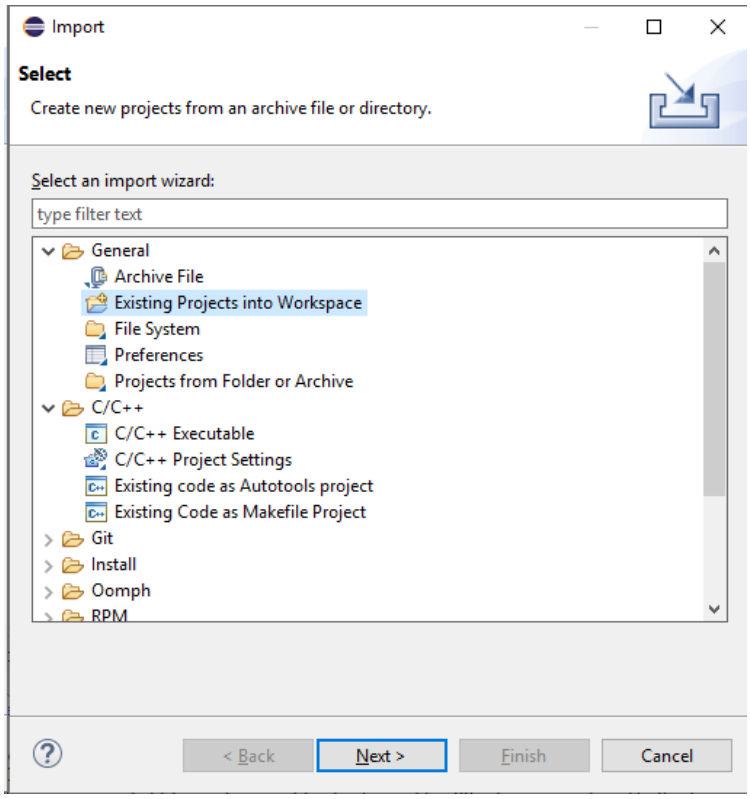

Figure 1-10 Import an existing project

Browse to select the GCC project and click “Finish” to open project, as showin in Figure 1-11. Now the Eclipse should be able to comile N9H30 GCC projects.

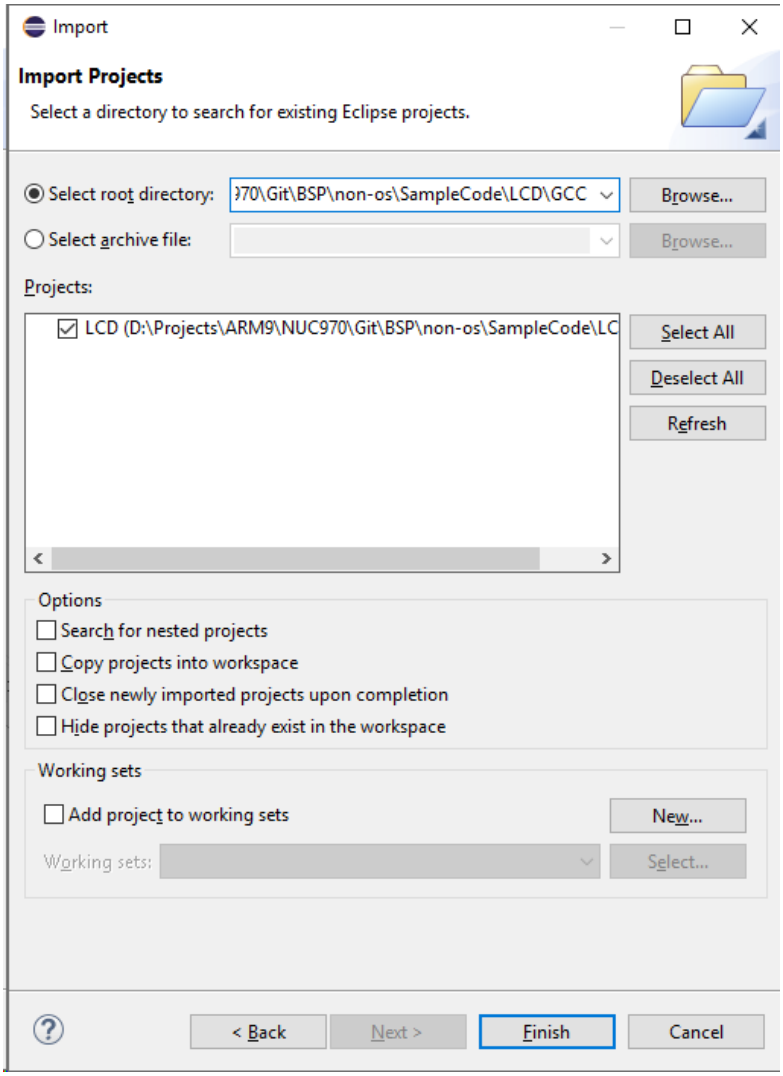

Figure 1-11 Import the GCC project

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, and then click Apply button.

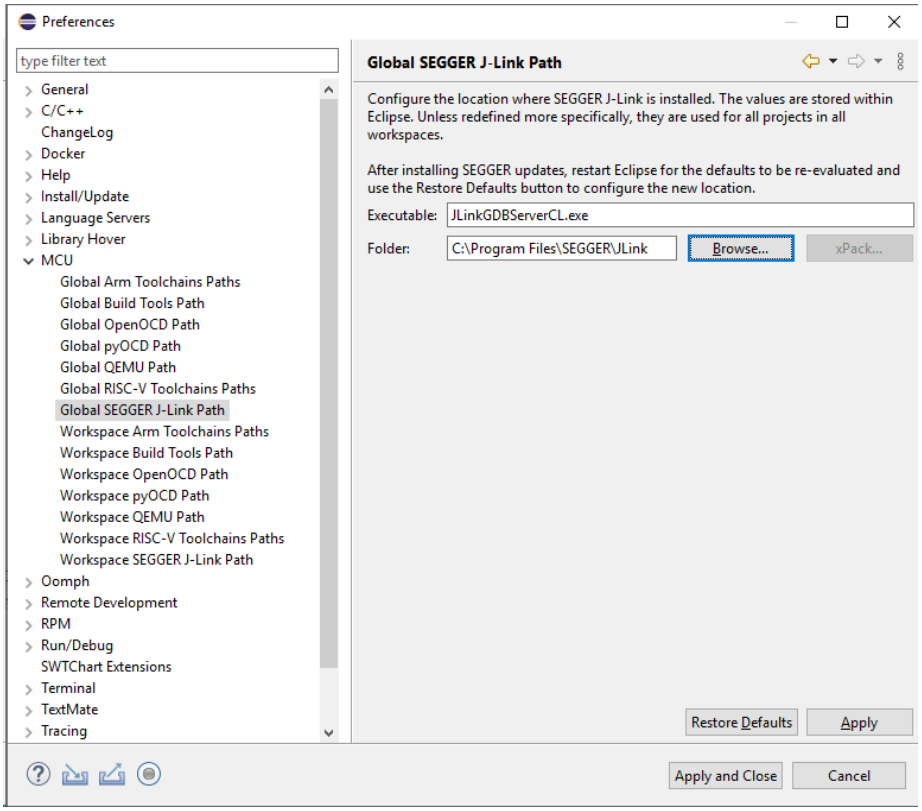

Figure 1-12 Global SEGGER J-Link Path Setting

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.

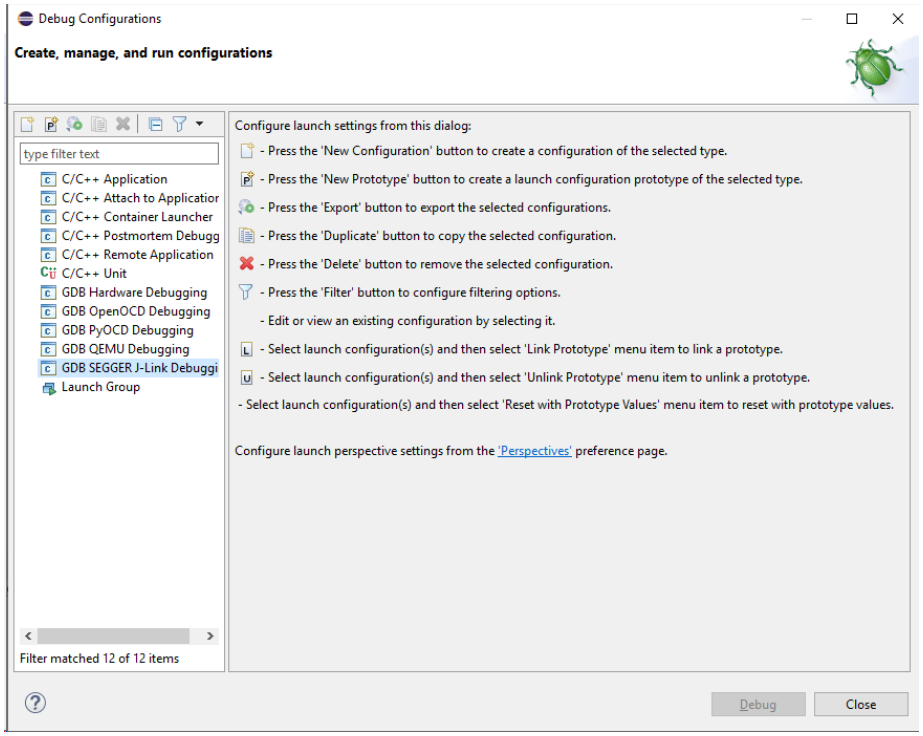

Figure 1-13 GDB SEGGER J-Link Debug

Go to Debug tab, and configure this tab as Figure 1-14.

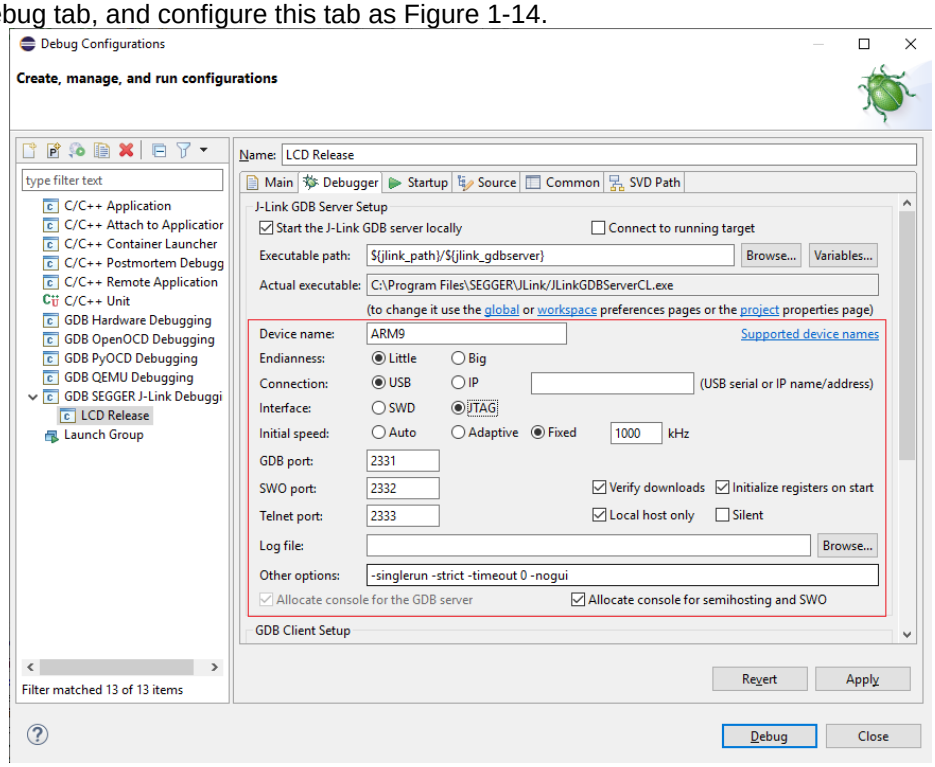

Figure 1-14 J-Link Debugger Setting

Go to Startup tab, and configure this tab as Figure 1-15.

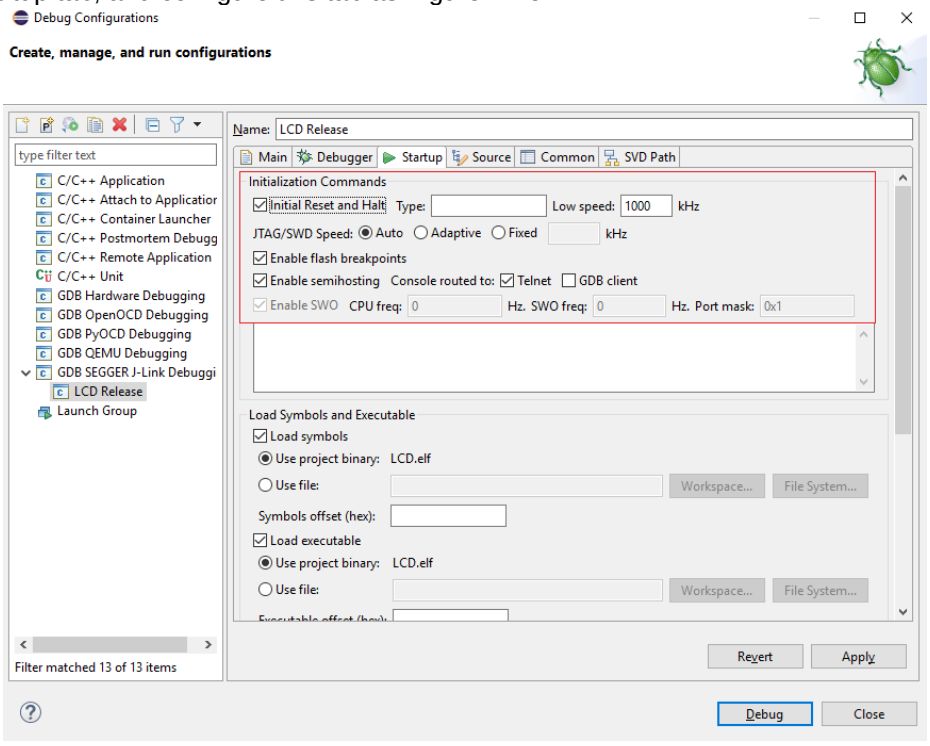

Figure 1-15 J-Link Startup Setting

After completing the setting, have N9H30 board boot from USB and connect to NuWriter, which

can help initialize DDR. Now click Debug button to start debug with J-Link.

Convert a C project to C++ project. First, import an Eclipse project from File ➔ Import, the import window opend as shown in Figure 1-10. Select convert C++ from File ➔ New ➔Convert to a C/C++ Project (Adds C/C++ Nature), The convert window opend as shown in Figure 1-16 and Figure 1-17. And then click “Finish” to convert project.

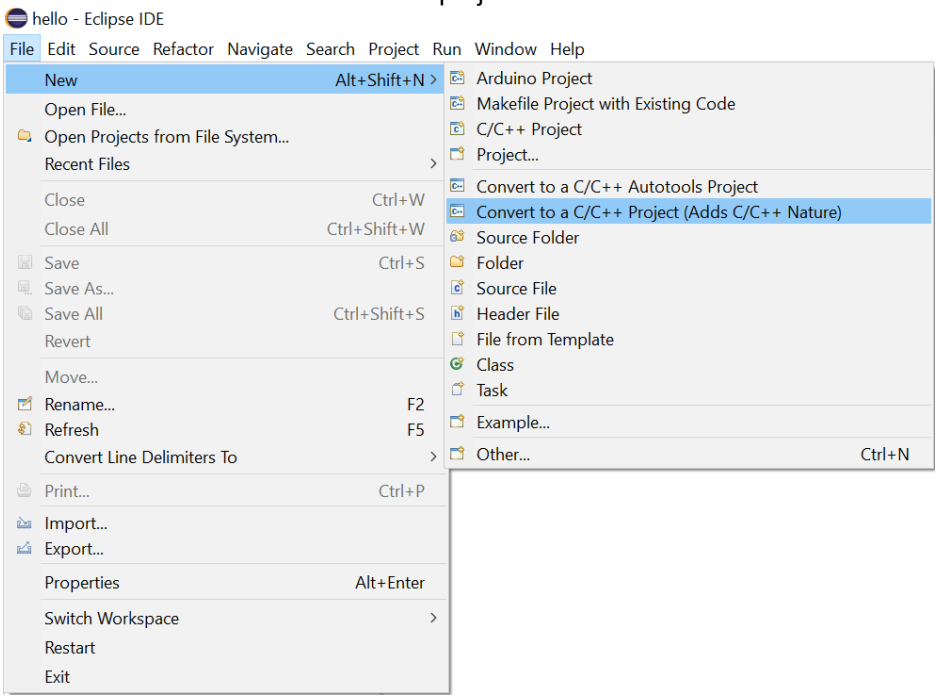

Figure 1-16 Select convert project

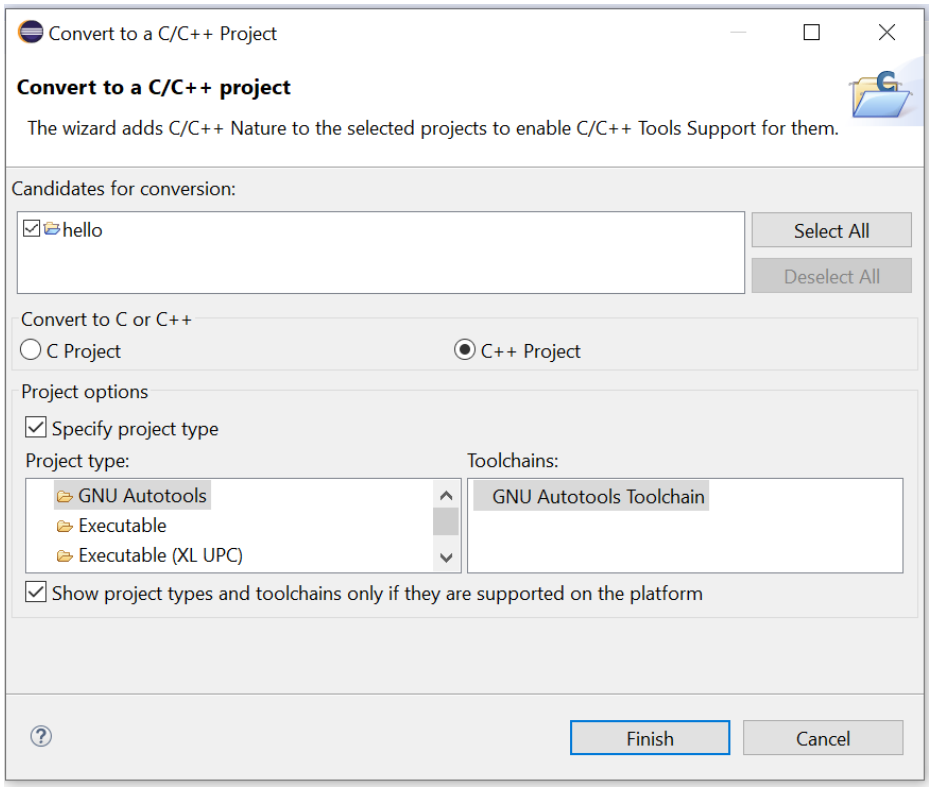

Figure 1-17 Convert to a C/C++ project

1.3 DEV Board Setting

N9H30 family supports different boot modes, it can boot from SPI, NAND, eMMC, or enter USB ISP mode. The booting mode is selected by PA[1:0] power-on-setting. Please refer to DEV board user's manual for boot source selection of DEV board.

2 BSP Content

2.1 BSP directory structure

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
Images	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	N9H30 peripheral drivers. Please refer to N9H30 Non-OS BSP Driver Reference Guide.chm under Documents directory for the usage of driver APIs.
Library	N9H30 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 NuWriter

NuWriter can download images to NAND flash, SPI flash, eMMC, or DDR while N9H30 is in USB ISP mode. Please refer to NUC970 N9H30 NuWriter User Manual for the usage of NuWriter.

4 Revision History

Version	Date	Description
1.00	Jan. 31, 2018	Initial release
1.01	Jul. 5, 2018	Minor update
1.02	Dec. 24, 2018	Minor update
1.10	May 31, 2019	Add Eclipse IDE description
1.20	Mar. 7, 2022	Remove emWin
1.30	Aug. 8, 2022	Update Eclipse developemnt environment
1.31	Dec. 16, 2022	Add convert C/C++ project

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.