# NuEclipse 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

### Table of Contents

1	Int	roduction	5
2	Sys	stem Requirements and Installation Guide	6
2.1 2.2 2.3 <b>2</b> 2 2 2.4	2.3.1 2.3.2 2.3.3	System Requirements Supported Chips Installation Performing the NuEclipse Installer on Microsoft Windows Extracting the NuEclipse Tar File on GNU/Linux Verifying the Eclipse Preferences Running Eclipse	
3	Dev	velopment Tutorial	
3.1 3.2 3.3 3.4 3.5 3 3.6 3 3.6 3 3.6 3 3.7 3.8 3.9	5.5.1 5.5.2 5.6.1 5.6.2 5.6.3 5.6.4	Select Workspace New Project Wizard. Import Existing Projects Build Settings. Debug Configuration Debugger Tab. Startup Tab. Debug Views. Registers View Disassembly View Disassembly View Peripheral Registers View Watchpoints. Debug in RAM. Debug Executable Files Only.	12 13 15 16 17 18 19 20 20 20 20 21 22 23 23 29 31 35
4	Ųά	A	
5	Re۱	vision History	

# nuvoTon

### List of Figures

Figure 2-1 Nu	Eclipse Setup Wizard	7
Figure 2-2 Ins	stall.sh Script ·····	
Figure 2-3 Pro	eferences for Global Tools Paths	9
Figure 2-4 Pro	eferences for OpenOCD Nu-Link ·····	10
Figure 2-5 Ec	lipse.exe and Related Folders	
Figure 3-1 Se	lecting a Workspace	
Figure 3-2 Ne	ew Project Wizard ·····	
Figure 3-3 Ta	rget Processor Settings	
Figure 3-4 Im	porting Projects	15
Figure 3-5 Bu	ild Settings ·····	
Figure 3-6 De	bug Configuration	
Figure 3-7 Co	nfiguring the Debugger Tab	
Figure 3-8 Co	nfiguring the Startup Tab	19
Figure 3-9 Re	gisters View ·····	20
Figure 3-10 M	1emory View ·····	
Figure 3-11 C	licking the Instruction Stepping Mode Button	22
Figure 3-12 D	Disassembly View	22
Figure 3-13 C	pening the Packs Perspective	23
Figure 3-14 H	low to Download Packages	
Figure 3-15 L	ocations of Repositories ·····	25
Figure 3-16 I	nstalling SFR Files ·····	
Figure 3-17 D	Device Selection	27
Figure 3-18 P	eripheral Registers View	28
Figure 3-19 T	oggle Watchpoint	29
Figure 3-20 P	roperties for C/C++ Watchpoint	30
Figure 3-21 A	dded Watchpoint in the Breakpoints View	30
Figure 3-22 M	1emory Layout ·····	
Figure 3-23 M	lodifying the ld Script	32
Figure 3-24 D	bebug Configuration Settings	
Figure 3-25 D	ebugging in RAM ·····	
Figure 3-26 I	mporting Executable for Debugging	35
Figure 3-27 S	electing Executable ······	
Figure 3-28 C	hoosing GDB Nuvoton Nu-Link Debugging	37
Dec. 1, 2022	1 Page <b>3 of 53</b>	Rev 1.01.019

# nuvoTon

Figure 3-29 Locating the GDB Executable	38
Figure 3-30 Choosing the ELF File to Download	39
Figure 3-31 Adding Source Lookup Path	40
Figure 4-1 Adding Udev Rules ·····	42
Figure 4-2 Preferences for String Substitution	43
Figure 4-3 More Options for NuLink2······	44

#### 1 Introduction

The **NuEclipse** is designed for cross-platform embedded ARM development. It includes a series of Eclipse plug-ins and tools. The plug-ins allow the user to create, build, and debug ARM-based projects within the Eclipse framework. Its features are listed below:

- **Creating projects by the New Project Wizard**: The New Project Wizard provides several templates for different target chips.
- **Building projects by the GNU ARM Toolchain**: The toolchain contains the ARM Embedded GCC compiler. The user can use it to build projects without restriction.
- **Debugging projects by GDB**: The user can halt, step, run, and monitor target chips. Accessing memory and flash is allowed. Setting hardware breakpoints and watchpoints is supported. In addition, the user can erase target chips and program the user configuration.

Through the **NuEclipse**, the user can develop projects of the NuMicro<sup>®</sup> Family within the Eclipse framework.

### 2 System Requirements and Installation Guide

#### 2.1 System Requirements

The following table lists system requirements for the user to run the **NuEclipse**.

	Minimum Requirements	Recommended Specifications
Operating System	Windows®7 x86_64 or GNU/Linux	Windows®10 x86_64 or Ubuntu 18.04 LTS
GNU ARM Embedded Toolchain	6-2017-q1-major	The latest version

**Note:** To have a fully usable and pleasant experience on Linux, the recommended Linux distribution is Ubuntu 18.04 LTS (64-bit).

#### 2.2 Supported Chips

To see the list of supported chips, please refer to **Supported\_chips.htm** in the folder of user manual.

#### 2.3 Installation

To make the **NuEclipse** ready for work, perform the following steps based on your operating system:

- 1. Performing the NuEclipse installer on Microsoft Windows.
- 2. Extracting the NuEclipse tar file on GNU/Linux.

#### 2.3.1 Performing the NuEclipse Installer on Microsoft Windows

On Windows, it is very easy to install the NuEclipse only by performing the NuEclipse installer. The installer will ask the user to install the **GNU ARM Eclipse Windows Build Tools** and **GNU ARM Embedded Toolchain** because they are required by NuEclipse.

ጄ NuEclipse_V1.01.017 Setup	- 🗆 X
	Welcome to the NuEclipse_V1.01.017 Setup Wizard
	This wizard will guide you through the installation of NuEclipse_V1.01.017.
	It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
	Click Next to continue.
	Next > Cancel

Figure 2-1 NuEclipse Setup Wizard

#### 2.3.2 Extracting the NuEclipse Tar File on GNU/Linux

On GNU/Linux, it is very easy to install the NuEclipse only by extracting the NuEclipse tar file. After that, **run the install.sh script** to complete the installation process. Please do not use the **sudo** command to run the script.

<	> 🕻 🏠 Home	Desktop NuEclipse_V1.0g_Linux_Setup UserManual >>	٩	
Ø	Recent	Name A	Size	Modified
۵	Home	clipse C	15 items 4 items	00:07 Yesterdav
	Desktop	🔲 UserManual	3 items	Yesterday
D	Documents	📓 install.sh	2.0 kB	21 Nov 2018
*	Downloads	🖹 uninstall.sh	501 bytes	21 Nov 2018

Figure 2-2 Install.sh Script

#### 2.3.3 Verifying the Eclipse Preferences

After the installation, the Eclipse preferences are automatically written on Windows. To verify them, click **Window > Preferences**, the Preferences wizard appears. Go to **C/C++ > Build > Global Tools Paths** and make sure the Build tools and Toolchain folder be correctly configured to what the installer has installed in the previous step. Click the **Apply** button to take effect. On GNU/Linux, Build tools folder path is not required. The path should be empty.

Preferences				×
type filter text	Global Tools Pat	ths	← → → →	
General ^ C/C++	The locations wher for all projects in al	e various GNU ARM Eclipse tools are installed. Unless defined more specifical I workspaces.	ly, they are u	sec
Appearance > Autotools	Build tools folder:	C:\Program Files (x86)\GNU ARM Eclipse\Build Tools\2.8-201611221915\bi	<u>B</u> rowse	
Build	Default toolchain:	GNU Tools for ARM Embedded Processors		
Build Targets Build Variables	Toolchain name:	GNU Tools for ARM Embedded Processors		_
Console	Toolchain folder:	C:\Program Files (x86)\GNU Tools ARM Embedded\6 2017-q1-update\bin	<u>B</u> rowse	
Clobal Tools Paths Logging Makefile Editor Settings Workspace Tools Path CMake Code Analysis Code Style Core Build Toolchains Debug Docker Container Editor File Types Indexer		Restore <u>D</u> efaults	Δρρίγ	
? 🗠 🖆 🐵		Apply and Close	Cancel	

Figure 2-3 Preferences for Global Tools Paths

Subsequently, go to **Run/Debug > OpenOCD Nu-Link** and make sure the OpenOCD folder be configured to where the installer has put the OpenOCD executable. On Microsoft Windows, for example, the path of OpenOCD folder could be C:/Program Files (x86)/Nuvoton Tools/OpenOCD. Similarly, on GNU/Linux it could be /usr/local/OpenOCD. The OpenOCD executable provided by Nuvoton is customized for Nu-Link. If the user tries to use other OpenOCD executable, OpenOCD and Nu-Link may not cooperate well. Click the **Apply** button to take effect.

Preferences			_	
type filter text	penOCD	Nu-Link	<	<b>→</b> -
> Property Pages Settin A Qt	onfigure th	e location where OpenOCD for Nu-Link is installed.		
Task Tags Template Default Valu	xecutable:	openocd.exe		
ChangeLog F	older:	C:/Program Files (x86)/Nuvoton Tools/OpenOCD/bin		Browse
> Docker				<u>_</u>
> Help				
> Install/Update				
> Library Hover				
> Mylyn				
> Oomph				
> Remote Development				
> RPM				
✓ Run/Debug				
Console				
External Tools				
> Launching				
OpenOCD Nu-Link				
Peripherals views				
Perspectives				
String Substitution				
View Management				
View Performance V		R	estore <u>D</u> efaults	<u>A</u> pply
? 占 🖌 Θ		Арр	ly and Close	Cancel

Figure 2-4 Preferences for OpenOCD Nu-Link

#### 2.4 Running Eclipse

To run **NuEclipse**, double-click the **eclipse.exe**. Note that the .exe file and the related folders, such as the OpenOCD folder, should stay in the same directory; otherwise, the application will not work properly.

Name	Date modified	Туре	Size
configuration	2020/10/22 下午 05:41	File folder	
dropins	2020/10/22 下午 05:41	File folder	
features	2020/10/22 下午 05:41	File folder	
OpenOCD	2020/10/22 下午 05:41	File folder	
p2	2020/11/30 下午 03:48	File folder	
Packages	2020/10/22 下午 05:41	File folder	
plugins	2020/11/17 下午 04:26	File folder	
readme	2020/10/22 下午 05:46	File folder	
.eclipseproduct	2020/9/2 下午10:06	ECLIPSEPRODUCT	1 KB
📄 artifacts.xml	2020/10/22 下午 05:23	XML Document	170 KB
eclipse.exe	2020/9/10 上午 11:05	Application	417 KB
📓 eclipse.ini	2020/10/20 上午 10:51	Configuration sett	1 KB
eclipsec.exe	2020/9/10 上午 11:05	Application	129 KB
🧧 notice.html	2020/9/7 下午 09:35	HTML Document	10 KB

Figure 2-5 Eclipse.exe and Related Folders

#### 3 Development Tutorial

#### 3.1 Select Workspace

When Eclipse launches, we have to select a workspace which groups a set of related projects together that usually make up an application. In addition, some configuration settings for Eclipse and projects are stored here, too. For different computers, the configuration settings may change. We should create our own workspace rather than copying another user's workspace. Only one workspace can be active at one time. The current workspace for Eclipse can be switched by clicking **File->Switch Workspace**.

🖨 Eclipse IDE Launcher	×
Select a directory as workspace	
Eclipse IDE uses the workspace directory to store its preferences and development artifacts.	
Workspace: C:\Users\XXXX\eclipse-workspace	
Use this as the default and do not ask again	
Recent Workspaces	
Launch Cancel	

Figure 3-1 Selecting a Workspace

#### 3.2 New Project Wizard

As a beginner, the fastest way to create a C/C++ project is using the New Project Wizard. For instance, to create a C project, click **File > New > C Project**. The New Project Wizard appears. Here we choose **Hello World Nuvoton Cortex-M C Project** for Project type. Input the project name and click the **Next >** button to continue.

Project			
Create C project of select	ed type		
Project name: FirstExam	ple		
Use <u>d</u> efault location			
Location: C:\Users\	\eclipse-workspace1	L\FirstExample	B <u>r</u> owse.
Choose file s <u>y</u> :	stem: default 🖂		
Project type:		Toolchains:	
GNU Autotools		Cross ARM GCC	
Executable			
Empty Project			
<ul> <li>Empty Project</li> <li>Hello World ANSI C</li> </ul>	Project		
<ul> <li>Empty Project</li> <li>Hello World ANSI C</li> <li>Hello World ARM C</li> </ul>	Project Project		
<ul> <li>Empty Project</li> <li>Hello World ANSI C</li> <li>Hello World ARM C</li> <li>Hello World Nuvoto</li> </ul>	Project Project n Cortex-M C Project		
<ul> <li>Empty Project</li> <li>Hello World ANSI C</li> <li>Hello World ARM C</li> <li>Hello World Nuvoto</li> <li>Shared Library</li> <li>Static Library</li> </ul>	Project Project n Cortex-M C Project		
<ul> <li>Empty Project</li> <li>Hello World ANSI C</li> <li>Hello World ARM C</li> <li>Hello World Nuvoto</li> <li>Shared Library</li> <li>Static Library</li> <li>Makefile project</li> </ul>	Project Project n Cortex-M C Project		
<ul> <li>Empty Project</li> <li>Hello World ANSI C</li> <li>Hello World ARM C</li> <li>Hello World Nuvoto</li> <li>Shared Library</li> <li>Static Library</li> <li>Makefile project</li> </ul>	Project Project n Cortex-M C Project		
<ul> <li>Empty Project</li> <li>Hello World ANSI C</li> <li>Hello World ARM C</li> <li>Hello World Nuvoto</li> <li>Shared Library</li> <li>Static Library</li> <li>Makefile project</li> </ul>	Project Project n Cortex-M C Project >		
<ul> <li>Empty Project</li> <li>Hello World ANSI C</li> <li>Hello World ARM C</li> <li>Hello World Nuvoto</li> <li>Shared Library</li> <li>Static Library</li> <li>Makefile project</li> <li>Show project types ar</li> </ul>	Project Project n Cortex-M C Project > d toolchains only if t	hey are supported on the pl	atform
<ul> <li>Empty Project</li> <li>Hello World ANSI C</li> <li>Hello World ARM C</li> <li>Hello World Nuvoto</li> <li>Shared Library</li> <li>Static Library</li> <li>Makefile project</li> <li>Show project types ar</li> </ul>	Project Project n Cortex-M C Project > od toolchains only if t	hey are supported on the pl	atform
<ul> <li>Empty Project</li> <li>Hello World ANSI C</li> <li>Hello World ARM C</li> <li>Hello World Nuvoto</li> <li>Shared Library</li> <li>Static Library</li> <li>Makefile project</li> <li>Show project types an</li> </ul>	Project Project n Cortex-M C Project > ad toolchains only if t	hey are supported on the pl	atform
<ul> <li>Empty Project</li> <li>Hello World ANSI C</li> <li>Hello World ARM C</li> <li>Hello World Nuvoto</li> <li>Shared Library</li> <li>Static Library</li> <li>Makefile project</li> <li>Show project types ar</li> </ul>	Project Project n Cortex-M C Project > od toolchains only if t	hey are supported on the pl	atform
<ul> <li>Empty Project</li> <li>Hello World ANSI C</li> <li>Hello World ARM C</li> <li>Hello World Nuvoto</li> <li>Shared Library</li> <li>Static Library</li> <li>Makefile project</li> <li>Show project types ar</li> </ul>	Project Project n Cortex-M C Project > d toolchains only if t	hey are supported on the pl	atform

Figure 3-2 New Project Wizard

Based on the actual target chip, we select the corresponding chip series. For some chip series, e.g., M2351\_NonSecure, we need to input the additional library path. If not, the build process may fail. In addition, input the real values to Flash and RAM size. If not, the default values will be used. When all the settings are done, click the **Next >** buttons until clicking the **Finish** button.

Chip Series:	M480	<b>`</b> ~
Additional library path:		
Flash size (kB):		
RAM size (kB):		_
Use system calls:	POSIX (system calls implemented by application code)	$\sim$
Check some warnings		
Check most warnings		
Enable -Werror		
Use -Og on debug		
Use newlib nano		
Use link optimizations		

Figure 3-3 Target Processor Settings

#### **3.3 Import Existing Projects**

When BSP projects are available, we can import them into the workspace using the following steps:

- 1. From the main menu bar, select File > Import. The Import wizard shows up.
- 2. Select General > Existing Project into Workspace and click Next.
- 3. Choose either **Select root directory** or **Select archive file** and click the associated **Browse** to locate the directory or file containing the projects. In the Nuvoton BSP, the Eclipse projects are stored in the GCC folder.
- 4. Under Projects select the project or projects which you would like to import and click **Finish**.

			-
Select root directory:	C:\MyCode\bsp\M480-master-internal\b	osp\SampleCode\Ten ~	B <u>r</u> owse
Select <u>archive file</u> :		~	B <u>r</u> owse
Projects:			
Template (C:\MyC	Code\bsp\M480-master-internal\bsp\Samp	oleCode <mark>(Template\GCC)</mark>	<u>S</u> elect All
			Deselect All
			R <u>e</u> fresh
<		>	
Options Search for nested pr	ojects		
Copy projects into w	vorkspace		
Close newly importe	ed projects upon completion Iready exist in the workspace		
Working sets			
Add project to wor	king sets		Ne <u>w</u>
Washing sets		~	S <u>e</u> lect
working sets:			
working sets:			

Figure 3-4 Importing Projects

#### 3.4 Build Settings

After projects have been created, we still have a chance to alter the build settings by clicking **Project > Properties**. The Properties wizard shows up. Then go to **C/C++ Build > Settings**. From there, we can alter the build settings according to the actual target chip. Then click the **Apply** button to take effect. After applying build settings, we should be able to build projects successfully.

Resource         Builders         C/C++ Build         Build Variables         Environment         Logging         Settings         Tool Chain Editor         Tools Paths         C/C++ General         Linux Tools Path         Project Natures         Project References         Run/Debug Settings         Task Repository         Task Tags         Validation         WikiText         WikiText         WikiText         WikiText         WikiText	evices Container Settings   evices Container Settings   ARM family cortex-m4   Architecture Toolchain default   Instruction set Thumb (-mthumb)   Thumb interwork (-mthumb-interwork)   Endianness   Float ABI   FP instructions (hard)   FPU Type   fpv4-sp-d16   Unaligned access   Toolchain default   Arch64 family   Generic (-mcpu=generic)   Feature crc   Toolchain default
Builders         C/C++ Build         Build Variables         Environment         Logging         Settings         Tool Chain Editor         Tools Paths         C/C++ General         Linux Tools Path         Project Natures         Project References         Run/Debug Settings         Task Repository         Task Tags         Validation         WikiText         WikiText         WikiText         Optimization         Archite         Builders         Cross ARM GNU Compiler         Preprocessor         WikiText	✓       Manage Configuration         evices       I Container Settings       Image Build Steps       Image Build Al         ARM family       cortex-m4       Image Configuration         Architecture       Toolchain default       Image Configuration         Instruction set       Thumb (-mthumb)       Image Configuration         Instruction set       Thumb (-mthumb)       Image Configuration         Instruction set       Thoub (-mthumb)       Image Configuration         Float ABI       FP instructions (hard)       Image Configuration         FPU Type       fpv4-sp-d16       Image Configuration         AArch64 family       Generic (-mcpu=generic)       Image Configuration         Feature crc       Toolchain default       Image Configuration         Feature crc       Toolchain default       Image Configuration
C/C++ Build         Build Variables         Environment         Logging         Settings         Tool Chain Editor         Tools Paths         C/C++ General         Linux Tools Path         Project Natures         Project References         Run/Debug Settings         Task Repository         Task Rags         Validation         WikiText         WikiText         WikiText         WikiText         WikiText         WikiText	evices Container Settings P Build Steps P Build Al ARM family cortex-m4 Architecture Toolchain default Instruction set Thumb (-mthumb) Thumb interwork (-mthumb-interwork) Endianness Toolchain default Float ABI FP instructions (hard) FPU Type fpv4-sp-d16 Unaligned access Toolchain default AArch64 family Generic (-mcpu=generic) Feature crc Toolchain default Feature crypto Toolchain default
Build Variables         Environment         Logging         Settings         Tool Chain Editor         Tools Paths         C/C++ General         Linux Tools Path         Project Natures         Project References         Run/Debug Settings         Task Repository         Task Tags         Validation         WikiText         WikiText         WikiText         WikiText         Wiscellaneous         Feature         Warnings         Cross ARM GNU Compiler         FU Ty         WikiText	evices Container Settings Puild Steps Puild Al ARM family cortex-m4 Architecture Toolchain default Instruction set Thumb (-mthumb) Thumb interwork (-mthumb-interwork) Endianness Toolchain default Float ABI FP instructions (hard) FPU Type fpv4-sp-d16 Unaligned access Toolchain default AArch64 family Generic (-mcpu=generic) Feature crc Toolchain default Feature crypto Toolchain default
Logging         Settings         Tool Chain Editor         Tools Paths         C/C++ General         Linux Tools Path         Project Natures         Project References         Run/Debug Settings         Task Repository         Task Tags         Validation         Validation         VikiText         WikiText         VikiText         VikiText         VikiText         VikiText	evices Container Settings Puild Steps Puild A
Settings       Tool Chain Editor         Tools Paths       Image: C/C++ General         Linux Tools Path       Image: Cross ARM GNU Assembler         Project Natures       Preprocessor         Project References       Image: Cross ARM GNU Assembler         Run/Debug Settings       Image: Cross ARM GNU Assembler         Task Repository       Image: Cross ARM GNU Compiler         Task Tags       Image: Cross ARM GNU Compiler         Validation       Image: Cross ARM GNU Compiler         WikiText       Image: Cross ARM GNU Compiler         VikiText       Image: Cross ARM GNU Compiler         Image: Cross ARM GNU Compiler       FPU Ty         Image: Cross ARM GNU Compiler       FPU Ty         Image: Cross ARM GNU Compiler       FPU Ty         Image: Cross ARM GNU Compiler       Feature         Image: Cross ARM GNU Clinker       Feature	ARM family       cortex-m4         Architecture       Toolchain default         Instruction set       Thumb (-mthumb)         Thumb interwork (-mthumb-interwork)       Endianness         Float ABI       FP instructions (hard)         FPU Type       fpv4-sp-d16         Unaligned access       Toolchain default         AArch64 family       Generic (-mcpu=generic)         Feature crc       Toolchain default
Tool Chain Editor Tools Paths       Image Processor       ARM fr.         C/C++ General Linux Tools Path       Optimization       Archite         Project Natures       Image Processor       Image Processor         Project References       Image Preprocessor       Image Preprocessor         Run/Debug Settings       Image Preprocessor       Image Processor         Task Repository       Image Preprocessor       Image Preprocessor         Task Tags       Image Preprocessor       Image Preprocessor         Validation       Image Preprocessor       Image Preprocessor         WikiText       Image Preprocessor       Image Preprocessor         Image Preprocessor       Image Preprocessor       Image Preprocessor         Validation       Image Preprocessor       Image Preprocessor         WikiText       Image Preprocessor       Image Preprocessor         VikiText       Image Preprocessor       Image Preprocessor         Image Preprocessor       Image Preprocessor       Image Preprocessor         Image Preprocessor       Image Preprocessor       Image Preprocessor         VikiText       Image Preprocessor       Image Preprocessor       Image Preprocessor         Image Preprocessor       Image Preprocessor       Image Preprocessor       Image Preprocessor <td>ARM family       cortex-m4         Architecture       Toolchain default         Instruction set       Thumb (-mthumb)         Thumb interwork (-mthumb-interwork)       Instruction set         Endianness       Toolchain default         Float ABI       FP instructions (hard)         FPU Type       fpv4-sp-d16         Unaligned access       Toolchain default         AArch64 family       Generic (-mcpu=generic)         Feature crc       Toolchain default</td>	ARM family       cortex-m4         Architecture       Toolchain default         Instruction set       Thumb (-mthumb)         Thumb interwork (-mthumb-interwork)       Instruction set         Endianness       Toolchain default         Float ABI       FP instructions (hard)         FPU Type       fpv4-sp-d16         Unaligned access       Toolchain default         AArch64 family       Generic (-mcpu=generic)         Feature crc       Toolchain default
Tools Paths       Image: Optimization       Archite         C/C++ General       Image: Optimization       Archite         Linux Tools Path       Image: Optimization       Image: Optimization         Project Natures       Image: Optimization       Image: Optimization         Project References       Image: Optimization       Image: Optimization       Image: Optimization         Run/Debug Settings       Image: Optimization       Image: Optimization       Image: Optimization       Image: Optimization         Task Repository       Image: Optimization       Image: Optimization       Image: Optimization       Image: Optimization         Validation       Image: Optimization       Image: Optimization       Image: Optimization       Image: Optimization         WikiText       Image: Optimization       Image: Optimization       Image: Optimization       Image: Optimization         Image: Optimization       Image: Optimization       Image: Optimization       Image: Optimization         Image: Optimization       Image: Optimization       Image: Optimization       Image: Optimization         Image: Optimization       Image: Optimization       Image: Optimization       Image: Optimization       Image: Optimization         Image: Optimization       Image: Optimization       Image: Optimization       Image: Optimization       Ima	Architecture       Toolchain default         Instruction set       Thumb (-mthumb)         Thumb interwork (-mthumb-interwork)         Endianness       Toolchain default         Float ABI       FP instructions (hard)         FPU Type       fpv4-sp-d16         Unaligned access       Toolchain default         AArch64 family       Generic (-mcpu=generic)         Feature crc       Toolchain default
C/C++ General       Image: Construction of the second	Architecture       Toolchain default         Instruction set       Thumb (-mthumb)         Thumb interwork (-mthumb-interwork)         Endianness       Toolchain default         Float ABI       FP instructions (hard)         FPU Type       fpv4-sp-d16         Unaligned access       Toolchain default         AArch64 family       Generic (-mcpu=generic)         Feature crc       Toolchain default
Linux Tools Path Project Natures Project References Run/Debug Settings Task Repository Task Tags Validation WikiText WikiText WikiText Debugging Instruct Preprocessor Miscellaneous Preprocessor Miscellaneous Preprocessor Miscellaneous Preprocessor Miscellaneous Preprocessor Miscellaneous Preature Preature Miscellaneous Preature Miscellaneous Preature Preat	Instruction set       Thumb (-mthumb)         Thumb interwork (-mthumb-interwork)         Endianness       Toolchain default         Float ABI       FP instructions (hard)         FPU Type       fpv4-sp-d16         Unaligned access       Toolchain default         AArch64 family       Generic (-mcpu=generic)         Feature crc       Toolchain default
Project Natures <ul> <li>Secross ARM GNU Assembler</li> <li>Thu</li> <li>Preprocessor</li> <li>Includes</li> <li>Endian</li> <li>Warnings</li> <li>Warnings</li> <li>Wiscellaneous</li> <li>Validation</li> <li>Secross ARM GNU C Compiler</li> <li>Preprocessor</li> <li>Unalig</li> <li>Optimization</li> <li>Secross ARM GNU C Linker</li> <li>Feature</li> <li>Secross ARM GNU C Linker</li> <li>Feature</li> <li>Secret Florid Libraries</li> <li>Feature</li> <li>Secret Florid Libraries</li> </ul>	☐ Thumb interwork (-mthumb-interwork)         Endianness       Toolchain default         Float ABI       FP instructions (hard)         FPU Type       fpv4-sp-d16         Unaligned access       Toolchain default         AArch64 family       Generic (-mcpu=generic)         Feature crc       Toolchain default         Feature crypto       Toolchain default
Project References       Preprocessor       Endian         Run/Debug Settings       Includes       Endian         Task Repository       Warnings       Float A         Task Tags       S Cross ARM GNU C Compiler       FPU Ty         WikiText       Preprocessor       Unalig         WikiText       Optimization       AArchi         Warnings       Miscellaneous       Feature         WikiText       Warnings       Feature         WikiText       Wiscellaneous       Feature         WikiText       Wiscellaneous       Feature         Wiscellaneous       Feature       Feature         Wiscellaneous	EndiannessToolchain defaultFloat ABIFP instructions (hard)FPU Typefpv4-sp-d16Unaligned accessToolchain defaultAArch64 familyGeneric (-mcpu=generic)Feature crcToolchain defaultFeature cryptoToolchain default
Task Repository       Image: Cross ARM GNU C Compiler       Float A         Yalidation       Image: Cross ARM GNU C Compiler       FPU Ty         WikiText       Image: Cross ARM GNU C Compiler       FPU Ty         Image: Cross ARM GNU C Compiler       FPU Ty       Image: Cross ARM GNU C Compiler         Image: Cross ARM GNU C Compiler       Image: Cross ARM GNU C Compiler       Feature         Image: Cross ARM GNU C Compiler       Feature       Image: Cross ARM GNU C Compiler         Image: Cross ARM GNU C Compiler       Feature       Image: Cross ARM GNU C Compiler         Image: Cross ARM GNU C Compiler       Feature       Image: Cross ARM GNU C Compiler         Image: Cross ARM GNU C Compiler       Feature       Image: Cross ARM GNU C Compiler         Image: Cross ARM GNU C Compiler       Feature       Image: Cross ARM GNU C Compiler         Image: Cross ARM GNU C Compiler       Feature       Image: Cross ARM GNU C Compiler         Image: Cross ARM GNU C Compiler       Feature       Image: Cross ARM GNU C Compiler         Image: Cross ARM GNU C Compiler       Feature       Image: Cross ARM GNU C Compiler         Image: Cross ARM GNU C Compiler       Feature       Image: Cross ARM GNU C Compiler         Image: Cross ARM GNU C Compiler       Feature       Image: Cross ARM GNU C Compiler         Image: Cross ARM GNU C Compiler	Float ABI       FP instructions (hard)         FPU Type       fpv4-sp-d16         Unaligned access       Toolchain default         AArch64 family       Generic (-mcpu=generic)         Feature crc       Toolchain default         Feature crypto       Toolchain default
Task Tags       Image: Constant American Ame	FPU Type       fpv4-sp-d16         Unaligned access       Toolchain default         AArch64 family       Generic (-mcpu=generic)         Feature crc       Toolchain default         Feature crypto       Toolchain default
WikiText       Includes       Unalig         Includes       Optimization       AArchi         Warnings       Miscellaneous       Feature         Seneral       Elibraries       Feature         Wiscellaneous       Feature       Feature         Seneral       Elibraries       Feature         Seneral       Seneral       Feature	Unaligned access       Toolchain default         AArch64 family       Generic (-mcpu=generic)         Feature crc       Toolchain default         Feature crypto       Toolchain default
Includes       AArchi         Image: Construction       AArchi         Image: Construction       AArchi         Image: Construction       Feature         Image: Construction	AArch64 family Generic (-mcpu=generic) Feature crc Toolchain default Feature crypto Toolchain default
Warnings     Warnings     Warnings     Miscellaneous     Feature     Seneral     Wiscellaneous     Feature     Wiscellaneous     Feature     Wiscellaneous	Feature crc     Toolchain default       Feature crypto     Toolchain default
✓      ✓	Feature crypto Toolchain default
Image: Construction of the co	Feature crypto
Eibraries     Feature     Miscellaneous     Feature	
Miscellaneous     Feature     Feature	Feature fp Toolchain default
L. MA Concert ADM Chill Concerts Floor Income	Feature simd Enabled (+simd)
General     Code r	Gode model Small (-mcmodel=small)
V 🛞 Cross ARM GNU Print Size	Strict align (-mstrict-align)
🖉 General Other	Other target flags

Figure 3-5 Build Settings

#### 3.5 Debug Configuration

Before launching an application into the debug mode, we have to prepare a debug configuration, which contains all the necessary information about the debug mode. Click **Run > Debug Configuration...** to open the debug configuration dialog. Double click on the **GDB Nuvoton Nu-Link Debugging** group. The Nuvoton Nu-Link debug configuration appears on the right-hand side. In the Main tab, the name of Project should coincide with the project name. The C/C++ Application should point to the .elf application generated by the build process. If the project name or C/C++ Application is incorrect, please select the expected project first in the project view, build the project to generate the executable, and expand the tree to make sure the existence of the generated executable. Then repeat the former operations again.

Debug Configurations			– 🗆 X
Create, manage, and run configuratio	ns		Ť
type filter text C/C++ Application C/C++ Attach to Application C/C++ Container Launcher C/C++ Postmortem Debugger C/C++ Remote Application Cit C/C++ Unit GDB Hardware Debugging C GDB Nuvoton Nu-Link Debugging C FirstExample Debug Launch Group	Name:       FirstExample Debug         Main       Debugger         Project:       FirstExample         FirstExample       C/C++ Application:         Debug\FirstExample.elf       Build (if required) before launching         Build Configuration:       Select Automatically         © Enable auto build       © Use workspace settings	urce Common	Browse Browse
< > Filter matched 10 of 11 items		Re⊻ert	Apply
?		<u>D</u> ebug	Close

#### Figure 3-6 Debug Configuration

#### 3.5.1 Debugger Tab

The Debugger tab is used to provide the OpenOCD and GDB Client setup. OpenOCD requires correct configuration files to know how to work with adapters and target chips. The configuration files are specified in the **Config options** field. Nuvoton's adapter is Nu-Link, which uses the interface configuration file named **nulink.cfg**. In addition, Nuvoton has three different ARM families, such as M0, M4, and M23. The corresponding target configuration files are **numicroM0.cfg**, **numicroM4.cfg**, and **numicroM23.cfg**. For M23 2<sup>nd</sup> development, the target configuration file would be **numicroM23\_NS.cfg**.

Debug Configurations		– 🗆 X
Create, manage, and run configuratio	ns	Ť
Image: Second	Name:       FirstExample Debug         Main       Startup       Source       Common         OpenOCD Setup       Start OpenOCD locally       Browse         GDB port:       3333       Browse         GDB port:       3333       Telnet port:       4444         Config options:       -f.,/scripts/interface/nulink.cfg -f.,/scripts/target/numicroM4.cfg       Browse         GDB Client Setup       S(cross_prefix)gdb\$(cross_suffix)       Browse         Client port:       3333       Other options:       Commands:         Commands:       set mem inaccessible-by-default off       sata and commands:         Remote Target       Host name or IP address:       localhost         Port number:       3333       Gasaa         Force thread list update on suspend       Sata and commands       sata and commands	Variables
< >> Filter matched 10 of 11 items	Revert	Apply
?	Debug	Close

Figure 3-7 Configuring the Debugger Tab

#### Startup Tab 3.5.2

As the first step, we should choose the right Chip Series in the Startup tab. When done, the corresponding target configuration file will be automatically written in the Config options field of the Debugger tab. To load executable to flash, we need to select the Load executable to flash checkbox. To load executable to RAM, we need to select the Load executable to SRAM checkbox. When all the settings are done, click the Apply button to take effect. To launch the application into the debug mode, click the **Debug** button.

Debug Configurations	- □ >
eate, manage, and run configuratior	15
: 🖻 🗭 🗎 🗙 🗖 🏹 🗸	Name: FirstExample Debug
ype filter text	Main (参 Debugger ) Startup Source 🔲 Common
C/C++ Application	Initialization Commands
C/C++ Attach to Application	☑ Initial Reset Type: init
C/C++ Postmortem Debugger	^
C/C++ Remote Application	
GDB Hardware Debugging	
C GDB Nuvoton Nu-Link Debugging	Erase chin
Launch Group	Chip Series: NuMicro M4
	Write Config0: 0x FFFFFFF Config1: 0x FFFFFFFF Config2: 0x FFFF545A Config3: 0x FFFFFFFF
	Load Symbols and Executable
	Use project binary: FirstExample.elf
	O Use file: Workspace File System
	Symbols offset (hex):
	□ Load executable to flash
	Use project binary: FirstExample.hex
	O Use file: Workspace File System
	Executable offset (hex):
	Load executable to SRAM
	Use project binary: FirstExample.hex
	Use file: Workspace File System
	Executable offset (hex):
	Run/Restart Commands
	Pre-run/Restart reset Type: init (always executed at Restart)
	^
	Set program counter at (hey):
>	Revert Apply
ter matched 10 of 11 items	hereit heppy

Figure 3-8 Configuring the Startup Tab

#### 3.6 Debug Views

Eclipse provides many debug views. Each of them contains specific information for debugging.

#### 3.6.1 Registers View

When entering the debug mode, we can open the **Registers view** in the bottom of Debug perspective. The Registers view lists information about the registers in a selected stack frame.

🗐 Console 🚦 Registers 🔀 🖹 Problems	🜔 Executables 🛛 🙀 Debugger Console 📋 N	vlemory 🐁 🎫 🕒 📩 😭 🖇 🗖	
Name	Value	Description	^
🗸 👬 General Registers		General Purpose and FPU Register Group	
1010 <b>rO</b>	0		
1010 rl	0		
1010 r2	536871108		
1010 r3	537		
1010 <b>r</b> 4	0		
1010 <b>r5</b>	0		
1010 <b>r6</b>	0		
1010 <b>r7</b>	0		
1010 0101 <b>r8</b>	0		
1010 <b>r9</b>	0		
1010 <b>r10</b>	536936448		
1010 <b>r11</b>	0		
<sup>1010</sup> r12	0		
0101 sp	0x2001fff8		
1000 lr	643		
<sup>1010</sup> pc	0x32e <main+2></main+2>		
10101 xPSR	1627389952		~

Figure 3-9 Registers View

#### 3.6.2 Memory View

The **Memory view** of the Debug perspective is used to monitor and modify the process memory. The process memory is presented as a list of so called **memory monitors**. Each monitor represents a section of memory specified by its location called **base address**. To open it, click the Memory tab on the lower side of Debug perspective.

					1012 1010 📑 🖬	- <b>1</b>
nitors 🚽	🗶 💥 0x20000000 : (	0x20000000 <he< th=""><th>x&gt; 🕄 🔶 🛛</th><th>lew Renderings</th><th><u>)</u></th><th></th></he<>	x> 🕄 🔶 🛛	lew Renderings	<u>)</u>	
0x2000000	Address	0 - 3	4 - 7	8 - B	C - F	
	2000000	00000000	001BB700	001BB700	001BB700	
	20000010	0080000	00000000	10270000	00000000	
	20000020	00000000	00000000	001BB700	30000020	
	20000030	00000000	4C1B0000	6C1B0000	2C1B0000	
	2000040	00000000	00000000	00000000	00000000	
	20000050	00000000	00000000	00000000	00000000	
	20000060	00000000	00000000	00000000	00000000	
	20000070	00000000	00000000	00000000	00000000	
	2000080	00000000	00000000	00000000	00000000	
	20000090	19020000	F5010000	00000000	00000000	
	200000A0	00000000	00000000	00000000	00000000	
	20000B0	00000000	00000000	00000000	00000000	
	200000C0	00000000	2100D94A	10601046	C06920F0	
	20000D0	0100D061	00207047	014600BF	D3480069	
	200000E0	10F0010F	FAD1D148	006840F0	4000CF4A	
	200000F0	10602220	D06021F0	03005060	B1F5001F	
	20000100	01D1CB48	90600120	C84A1061	BFF3608F	
	20000110	00BEC648	006910F0	010FFAD1	C3480068	

Figure 3-10 Memory View

#### 3.6.3 Disassembly View

The **Disassembly view** shows the loaded program as assembler instructions mixed with source code for comparison. We can do the following tasks in the Disassembly view:

- 1. Setting breakpoints at the start of any assembler instruction.
- 2. Enabling and disabling breakpoints.
- 3. Stepping through the disassembly instructions of the program.
- 4. Jumping to specific instructions in the program.

To open it, we need to click the **Instruction Stepping Mode** button on the upper toolbar, as follows:



Figure 3-11 Clicking the Instruction Stepping Mode Button

Then the Disassembly view will appear on the right-hand side.



Figure 3-12 Disassembly View

#### 3.6.4 Peripheral Registers View

To display the Peripheral Registers view, we need to utilize **Packs** mechanism. Packs can help the user download special function register (SFR) files from the Keil repository. Firstly, we open the Packs perspective by choosing it in the **Open Perspective** dialog.

	<ul> <li>Docker Tooling</li> <li>GDB Trace</li> <li>Git</li> <li>LTTng Kernel</li> <li>OS Tracing Overview</li> <li>Packs</li> <li>Planning</li> <li>Resource</li> <li>Team Synchronizing</li> <li>Tracing</li> </ul>	□ C/C++ (default) 参 Debug	Open Perspective
pen			—
Cance			
			×

Figure 3-13 Opening the Packs Perspective

For the first time we see the Packs perspective, its content is provided by the NuEclipse installer. If the default content is missing, please switch to a new workspace and try again. To get the latest version, click the **Update the packages definitions from all repositories** button at the upperright corner. After clicking, Eclipse begins downloading all the SFR files from an online repository.

🎄 🔳 🔅 Debug	✓ FirstExample Debug		•			
• 🖗 • 🕫 👉 • 🔿 •				Q	8	Ec 🕇
🛛 🔝 В іК 🖳 🗆	🏪 Packs 🙁	Œ		H 🔆 🛛	8 8	- 0
🖭 🖻 🎘 i	Name	Description				
Nuvoton	✓ 2 Nuvoton					
ISD Family	✓	Nuvoton ARM Cortex-M NuMicro Family Device Support				
NPCX Family	1.3.10 (installed)	1. Add M253 device, update system startup file				
NuMicro M0 Family	1.3.9 (20MB)	Add M0A21, M030G, M071, M471, NM1244, NM1830, M251, M2354 device				
NuMicro M23 Family	1.3.8 (18MB)	1. Add M030 device				
NuMicro M4 Family	• 1.3.7 (18MB)	1. Add M2354 device				
NuVoice Family	9 1.3.6 (18MB)	Add M480 128K series, NUC029ZAN, NUC126NE4AE, NUC126SE4AE, NUC126SG4A				
	1.3.5 (16MB)	Add MR63, M032, NUC1311, TF5103, NM18107Y, NM18202Y devices, update M45				
	•• 1.3.4 (n/a)	Add M261 series, M480, M031 device, update NM1000 series device				
	🔮 1.3.3 (14MB)	Update M2351 Flash algorithm, M251/M252 series device and add M251 svd files				
	🔮 1.3.2 (n/a)	Add system and startup files				
	1.3.1 (6MB)	1. Add M480 series part no.				
	🔮 1.3.0 (13MB)	1. Add M23 and M4 family				
	🔮 1.2.2 (n/a)	M2351 secure and non-secure region defalt value				
	1.2.1 (26MB)	Improved PDSC file, adding include paths for device peripherals and new date fea				
	1.2.0 (26MB)	Add M2351 and NUC2201 series device and svd file and update NUC100, NUC200,				
	🔮 1.1.0 (25MB)	Add M0564, NUC121, NUC125, NUC126, Mini57 and M480 series example code				
	1.0.9 (17MB)	Rename Generic series				
	1.0.8 (17MB)	Add Nano103, NM1120, NM1330, NM1820 series svd file and update M051, M058S				
	1.0.7 (16MB)	Add NM1200 series example code and update svd file				
	1.0.6 (16MB)	Add Nano103, NM1120, NM1320, NM1330, NM1820 chip, update M0519, NUC230				
	9 1.0.5 (16MB)	Add Mini55 chip, update Mini58, NUC100, NUC122, NUC123, Nano100AN, Nano1				
	1.0.4 (15MB)	Add NUC122, M058S, Mini58, M0519 series example code				
	1.0.3 (14MB)	Add M451, NUC100, NUC200, NUC123, NUC029, M0518, NUC131, NANO100BN, N				
	1.0.2 (9MB)	Add ISD9100, ISD9300, N571, N572 svd file and M4LED chip series				
	1.0.1 (9MB)	Add (M051, Mini51, Nano112, NUC240, NUC472) example code and new chip series	:			
	1.0.0 (3MB)	First Release version of NuMicro Device Family Pack.				
	> H NuMicro_M051_BSP	Nuvoton ARM Cortex-M NuMicro Family M051 BSP Support				
	A MuMicro M2351 BSP	Nuvoton ARM Cortex-M NuMicro Family M2351 BSP Support				

Figure 3-14 How to Download Packages

The locations of repositories are specified in the **Window > Preferences > C/C++ > Packages > Repositories**. The default is from Keil's CMSIS Pack.

pe filter text	Repositories			¢ -	• 🖒 • 🖇
> Build ^ > CMake	Add links to the s	ites where packa	ges are published.		
Code Analysis	Туре	Name	URL		Add
Code Style Core Build Toolchain	CMSIS Pack	Keil	https://www.keil.com/pack/index.pidx		Edit
> Debug > Docker Container					Delete
> Editor File Types					
Indexer Language Mappings Meson					
> New C/C++ Project V					
<ul> <li>Packages</li> <li>Repositories</li> </ul>					
<ul> <li>Profiling</li> <li>Property Pages Settin</li> </ul>					
Qt Task Tags					
Template Default Valu					
ChangeLog Docker					
Help v				Restore <u>D</u> efaults	<u>A</u> pply

Figure 3-15 Locations of Repositories

When the download is completed, we can find the Nuvoton SFR files and install them on Eclipse if needed.

🗞 🔅 🔳 🔅 Debug	V FirstExample Debug	· · · · · · · · · · · · · · · · · · ·	🔗 🕶 🌛		
				Q	1
D 🛛 🔝 В іК 🖳 🗆	🏪 Packs 🛛		œ ⊨   ₩,	ا 🗞	8 <b>-</b> E
🕀 🖻 🎉 ŝ	Name	Description			
> Nuvoton	V 🚝 Nuvoton				
ISD Family	V H NuMicro DEP (installed)	Nuvoton ARM Cortex-M NuMicro Family Device Support			
NPCX Family	2 1.3.10 (installed)	1. Add M253 device, update system startup file			
NuMicro M0 Family	9 1.3.9 (20MB)				
NuMicro M23 Family	1.3.8 (18MB)	all 30 device			
NuMicro M4 Family	• 1.3.7 (18MB)	1. Add M2354 device			
NuVoice Family	1.3.6 (18MB)	Add M480 128K series, NUC029ZAN, NUC126NE4AE, NUC126SE4AE, NUC126SG4	۹		
	9 1.3.5 (16MB)	Add MR63, M032, NUC1311, TF5103, NM18107Y, NM18202Y devices, update M45	i		
	•• 1.3.4 (n/a)	Add M261 series, M480, M031 device, update NM1000 series device			
	1.3.3 (14MB)	Update M2351 Flash algorithm, M251/M252 series device and add M251 svd files			
	•• 1.3.2 (n/a)	Add system and startup files			
	1.3.1 (6MB)	1. Add M480 series part no.			
	1.3.0 (13MB)	1. Add M23 and M4 family			
	•• 1.2.2 (n/a)	M2351 secure and non-secure region defalt value			
	1.2.1 (26MB)	Improved PDSC file, adding include paths for device peripherals and new date fe	a		
	1.2.0 (26MB)	Add M2351 and NUC2201 series device and svd file and update NUC100, NUC200			
	9 1.1.0 (25MB)	Add M0564, NUC121, NUC125, NUC126, Mini57 and M480 series example code			
	1.0.9 (17MB)	Rename Generic series			
	1.0.8 (17MB)	Add Nano103, NM1120, NM1330, NM1820 series svd file and update M051, M058	S		
	1.0.7 (16MB)	Add NM1200 series example code and update svd file			
	1.0.6 (16MB)	Add Nano103, NM1120, NM1320, NM1330, NM1820 chip, update M0519, NUC23	D		
	1.0.5 (16MB)	Add Mini55 chip, update Mini58, NUC100, NUC122, NUC123, Nano100AN, Nano	L		
	9 1.0.4 (15MB)	Add NUC122, M058S, Mini58, M0519 series example code			
	9 1.0.3 (14MB)	Add M451, NUC100, NUC200, NUC123, NUC029, M0518, NUC131, NANO100BN, I	N		
	9 1.0.2 (9MB)	Add ISD9100, ISD9300, N571, N572 svd file and M4LED chip series			
	9 1.0.1 (9MB)	Add (M051, Mini51, Nano112, NUC240, NUC472) example code and new chip ser	ies		
	1.0.0 (3MB)	First Release version of NuMicro Device Family Pack.			
	> H NuMicro_M051_BSP	Nuvoton ARM Cortex-M NuMicro Family M051 BSP Support			
	> H NuMicro M2351 BSP	Nuvoton ARM Cortex-M NuMicro Family M2351 BSP Support			
	NuVoice ISD DEP	Nuvoton ARM Cortey-M NuVoice ISD and NPCy Family Device Support			

Figure 3-16 Installing SFR Files

### nuvoTon

To use the specific SFR file, open the project's properties dialog and go to the **C/C++ Build > Settings**. From there, we should choose the specific device matching the real one. In this case, it is M487JIDAE. Click the **Apply** button to take effect.

e filter text	Settings					(
Resource						
Builders						
C/C++ Build	Configuration:	Debug [Active]			✓ Man	age Configurations
Build Variables						
Environment						_
Logging	🛞 Tool Setting	js 🛞 Toolchains	Devices	Container Settin	gs 🎤 Build Steps	🚇 Build Artifa 🕚
Settings	- Device select	ion (Not vet used d	uring build!)			
Tool Chain Editor	Device select	ion (Not yet used u	anng bana.)			
Tools Paths	Name		Details			
L/C++ General		M485SIDAE	Device (1	60 kB RAM, 512 kB R	(OM)	
Droject Natures		M485KIDAE	Device (1	60 kB RAM, 512 kB R	(OM)	
Project Natures		M487SGAAE	Device (9	6 kB RAM, 256 kB RO	DM)	
Run/Debug Settings		M48/SIDAE	Device (1	60 KB RAM, 512 kB R	(OM)	
Task Repository		M487KIDAE	Device (1	60 kB RAM, 512 kB R	(OM)	
Task Tags		M487JIDAE	Device (1	00 KB KAIVI, 512 KB H 60 KB RANA 512 KB H		
/alidation		MI487 KIVICAN	Device (1 Subfamil	V Cortex-M4 100 M	(UM) LI-1	
WikiText		100505	Subranni	y (Contex-IVI4, 100 IVI	112)	~
	<					>
	Device core:	Cortex-M4				
	M487JIDAE Section	Start	Size	Startup		
	IRAM1	0x2000000	0x28000	0		
	IROM1	0x0000000	0x80000	1		
	INCOME	0,0000000	0,00000	-		
	Edit					
					Restore Defa	ults Apply

Figure 3-17 Device Selection

As a result, we can monitor the peripheral registers when debugging.

		to be bug ↓ □	FirstExample Debug $\checkmark$ 🖄 🗄 🖙 🗸		<i>₅</i> % ■ 11 <1	. 🔿 . e   i> 🗮 🖘   🥾 🐜
	· •					× : 8 40
.c) m	iain.c 🔀			(x)= Vari 💁 Bre	👾 Exp 🖹 Mod.	🎬 Disa 🧏 Peri 💥 🖳 🗖
10	* @fil		***************************************			🏝 🎫 🖻
3	* @ver	rsion V1.00		Peripheral	Address	Description
4	* @bri	ief A project template	for M480 MCU.	ECAP1	0x400B5000	ECAP Register Map
5	*	unisht (C) 2016 Numeter T	asheelen. Com. All sights recoved	🗆 🚼 EMAC	0x4000B000	EMAC Register Map
7	******	byright (C) 2016 <u>NUVOTON</u> I	ecnnology <u>Corp</u> . All rights reserved.	EPWM0	0x40058000	EPWM Register Map
8	#includ	le <stdio.h></stdio.h>		EPWM1	0x40059000	EPWM Register Map
9	#includ	le "NuMicro.h"		E Rec	0x4000C000	FMC Register Map
10	#define	PLL CLOCK 1920	00000	GPIO	0x40004000	GPIO Register Map
12	#ucl In	1720 IS20	00000		0x4004F000	HSOTG Register Map
130	void S	<pre>/S_Init(void)</pre>			0x40019000	USBU Register Map
14	{				0x4001A000	DC Register Man
16	/*	Init System Clock			0x40081000	I2C Register Map
17	/*.			□ <u>2</u> 12C2	0x40082000	I2C Register Map
18	/*	Unlock protected register	s */	□ <mark>2</mark> 12S	0x40048000	I2S Register Map
20	SYS	_UNIOCKReg();			0x40000300	NMI Register Map
21	/*	Set XT1_OUT(PF.2) and XT1	_IN(PF.3) to input mode */	🗌 🛃 NVIC	0xE000E100	NVIC Register Map
22	PF	->MODE &= ~(GPI0_MODE_MODE	2_Msk   GPIO_MODE_MODE3_Msk);		0x40046000	OPA Register Map
23	1*	Enable Extensel VIAL (1.2	4 MU-) */	🗆 🔀 ОТБ	0x4004D000	OTG Register Map
24 25	()* CI	Chapte External XIAL (4~2 CEnableXtalRC/CLK PWRCTI	HXTEN Msk);	🗌 🚼 PDMA	0x40008000	PDMA Register Map
26		mic) L_		🗌 🚼 QEI0	0x400B0000	QEI Register Map
27	/*	Waiting for 12MHz clock r	eady */	C 🔁 CEIL	0x400B1000	QEI Register Map
28	CLI	WaitClockReady(CLK_STATU	S_HXTSTB_Msk);	C 🔤 🔁 QSPI0	0x40060000	QSPI Register Map
30	/*	Set core clock as PLL CLO	CK from PLL */		0x40069000	QSPI Register Map
31	CLI	<pre></pre>		LI TA RTC	0x40041000	KTC Register Map
32	/*	Set PCLK0/PCLK1 to HCLK/2			0x40090000	SC Register Map
53 34	CLI	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	APD001V_DIV2   CLK_PCLKDIV_APB1DIV_DIV		0x40091000	SC Register Map
35	/*	Enable UART clock */			0x+0092000	SVST_SCR Register Man
36	CLI	_EnableModuleClock(UART0_	MODULE);		0x40000000	SDH Register Map
37	1+	Coloct HADT clash and	from HVT */		0x4000F000	SDH Register Map
58 39	/* CU	Setect UAKI CLOCK SOURCe	ULE, CLK CLKSEL1 UARTØSFI HXT. CLK CLK	SPID	0x40061000	SPI Register Map
40	CL1	N0001001000(0AN10_100	,censee_ownosee_int, cen_cen	SP11	0x40062000	SPI Register Map
41	/*	Update System Core Clock	*/	SPI2	0x40063000	SPI Register Map
42	/*	User can use SystemCoreCl	ockUpdate() to calculate SystemCoreClo		0x40007000	SPIM Register Map
43 44	Sys	scemcoreciockUpdate();		SYS	0x40000000	SYS Register Map
45				TMR01	0x40050000	TIMER Register Map
46		cation while the start		🗌 🔀 TMR23	0x40051000	TIMER Register Map
47 48	/*	Set GPB multi-function pi	ns tor UARIO RXD and TXD */	TRNG	0x400B9000	TRNG Register Map
49	SYS	5->GPB MFPH  = (SYS GPB MF	PH PB12MFP UARTO RXD   SYS GPB MFPH PB	UARTO	0x40070000	UART Register Map
50	/*	Lock protected registers	*/	UART1	0x40071000	UART Register Map
51	SYS	5_LockReg();			0x40072000	UART Register Map
53	3			<		>
55 56 57 58 59 60 61	* Thi: * own * Thi: * "Hel */	s is a template project fo application without worry s template application use llo World", users may need	<pre>r MABW series MCU. Users could based o about the IAR/Keil project settings. s external crystal as HCLK source and to do extra system configuration base v</pre>	<		,
🕒 Co	onsole 🐰	🔐 Registers 🔝 Problems 🔘 Ex	xecutables 🙀 Debugger Console 🔋 Memory 🔀		1012 1010 📑 🗖	1 📑 🕄 🖏 🗤 🔹 🗆 🗆
vionito	ors	🕂 🗶 🦂	🕱 🚰 GPIO: 0x40004000 🔀 🔶 New Renderings			
	0x20000	000	Register	Address		Value
٠	0010		> 1000 PA_DINOFF	0x40004	004	0x0000000
<b>ب</b>	GPIO			0.40004	008	0x0000FFFF
\$ \$	GPIO		✓ iiii PA_DOUT	0x40004		
<b>\$</b>	GPIO		✓ ININ PA_DOUT BOUTO	[0]		0x1:1
•	GPIO		VIII PA_DOUT	[0] [1]		0x1:1 0x1:1
<b>\$</b>	GPIO		✓ ### PA_DOUT	[0] [1] [10]		0x1:1 0x1:1 0x1:1
<b>\$</b>	GPIO			(0) (1) (10) (11)		0x1:1 0x1:1 0x1:1 0x1:1 0x1:1
•	GPIO			[0] [1] [10] [11] [12]		0xd: 1 0xd: 1 0xd: 1 0xd: 1 0xd: 1 0xd: 1
•	GPIO			(0) (1) (11) (11) (12) (13) (14)		0xd:1 0xd:1 0xd:1 0xd:1 0xd:1 0xd:1 0xd:1
•	GPIO			(0) (1) (1) (11) (12) (13) (14) (15)		0d: 1 0d: 1 0d: 1 0d: 1 0d: 1 0d: 1 0d: 1 0d: 1 0d: 1 0d: 1
•	GPIO		↓ IIII PA_DOUT           ● DOUT0           ● DOUT1           ● DOUT1           ● DOUT1           ● DOUT12           ● DOUT13           ● DOUT14           ● DOUT15           ● DOUT15	(0) (0) (1) (10) (11) (12) (13) (14) (14) (15) (2)		Od: 1
•	GPIO			(0) (0) (1) (10) (11) (12) (13) (14) (15) (2) (2) (3)		0d: 1 0d: 1
•	GPIO		✓	(1) (1) (1) (11) (12) (13) (14) (15) (2) (3) (4)		0d: 1 0d: 1
•	GPIO		✓ iiiii PA_DOUT           ●         DOUT0           ●         DOUT1           ●         DOUT10           ●         DOUT11           ●         DOUT12           ●         DOUT13           ●         DOUT14           ●         DOUT15           ●         DOUT3           ●         DOUT3           ●         DOUT3           ●         DOUT3           ●         DOUT3           ●         DOUT3           ●         DOUT4           ●         DOUT3           ●         DOUT3	0340004 [0] [1] [10] [11] [12] [13] [44] [5] [6]		Od: 1
<b>\$</b>	GPIO		↓ IIII PA_DOUT           ● DOUT0           ● DOUT1           ● DOUT1           ● DOUT1           ● DOUT12           ● DOUT13           ● DOUT14           ● DOUT5           ● DOUT4           ● DOUT4           ● DOUT5	(0) (0) (1) (11) (12) (13) (14) (15) (2) (3) (4) (4) (5) (6)		Od: 1           Od: 1
<b>\$</b>	GPIO		✓ iiiii PA_DOUT           ●         DOUT0           ●         DOUT1           ●         DOUT1           ●         DOUT1           ●         DOUT1           ●         DOUT10           ●         DOUT11           ●         DOUT12           ●         DOUT12           ●         DOUT13           ●         DOUT14           ●         DOUT2           ●         DOUT3           ●         DOUT3           ●         DOUT3           ●         DOUT3           ●         DOUT5           ●         DOUT5           ●         DOUT5           ●         DOUT5           ●         DOUT5	0340004 (0) (1) (10) (11) (12) (13) (14) (15) (2) (3) (4) (5) (6) (7)		Od: 1

Figure 3-18 Peripheral Registers View

### nuvoton

#### 3.7 Watchpoints

To add watchpoints on Eclipse, we need to do the following steps:

- 1. Selecting a **globe variable**, i.e. g\_seconds, in the Outline view.
- 2. Right-clicking on the global variable and choosing Toggle Watchpoint.



Figure 3-19 Toggle Watchpoint



3. Configuring the settings for watchpoints. To stop execution when the watch expression is read, select the **Read** checkbox. To stop execution when the watch expression is written to, select the **Write** checkbox.

Properties for C/C++ Wa	tchpoint	— — X
Common	Common	<> ▼ <> ▼ 8
	Common Class: Project: File: Expression to watch: Range: Read Write Enabled Condition: Ignore count:	C/C++ Watchpoint FirstExample C:\Users\\eclipse-workspace1\FirstExample\User\main.c g_seconds 0
?		Apply and Close Cancel

Figure 3-20 Properties for C/C++ Watchpoint

When the watchpoint is added, it appears in the Breakpoints view.

(x)= Var	● <sub>●</sub> Bre	23	∰ Ex	₽ Out	=\	Мо			Dis	묾	Per.			
						X	×	2	d	`Q	( <del>+</del>	F	\$₽}	000
✓ 💣	[function	: ma	in] [type: `	Temporary]										
🗹 🎣	main.c [e	pres	sion: 'g_se	econds']										

Figure 3-21 Added Watchpoint in the Breakpoints View

#### 3.8 Debug in RAM

To debug in RAM, there are several steps to follow:

- 1. Modifying the ld script.
- 2. Assigning PC to the specific RAM address.
- 3. Assigning SP to the specific RAM address.
- 4. Downloading the binary file to RAM.

The ld script is responsible for telling the linker the layout of the compiled executable. For example, the memory layout looks like:



#### Figure 3-22 Memory Layout





Figure 3-23 Modifying the Id Script

To assign PC and SP to the specific addresses, we need to input them in the debug configuration, as follows. Based on the previous memory layout, the PC and SP addresses should be Reset\_Handler and 0x20001400, respectively. In addition, set Vector Table Offset Register (0xE000ED08) should be 0x2000000 and unselect the **Pre-run/Restart reset** Button. To download the binary to RAM, we select the **Load executable to SRAM** button and unselect the **Load executable to flash** button. Click the **Debug** button to start a debug session.

Debug Configurations		— 🗆 X
Create, manage, and run configuration	ns	TO.
		2
	Name: FirstExample Debug	
type filter text	Main 🕉 Debugger 🌔 Startup V 💱 Source 🔲 Common	
C/C++ Application	Initialization Commands	
C/C++ Container Launcher		
C/C++ Postmortem Debugger		^
C C/C++ Remote Application		
GDB Hardware Debugging		×
C GDB Nuvoton Nu-Link Debugging		
Launch Group	Chip Series: NuMicro M4	
	Write Config0: 0x FEFFFFF Config1: 0x FEFFFFFF Config2: 0x FEFF5A5A Con	fig3: 0x EEEEEEE
	Load Symbols and Executable	
	✓ Load symbols	
		File System
		space The system
	Symbols offset (nex):	
	Use project binary: FirstExample.hex	
	O Use file:	snace File System
	Evacutable officet /bev/u	
	✓ Load executable to SRAM	
	Use project binary: FirstExample.hex	
	O Use file: Work	space File System
	Evecutable offset (bey)	
	Run/Restart Commands	
	Pre-runy restart reset Type: Init (arways executed at Restart)	
	set Ssp=0x20001400 set Spc=Reset_Handler	<u>^</u>
	monitor mww 0xE000ED08 0x20000000	
	Set program counter at (hex):	
	Set breakpoint at:	
		Restore defaults
		<u>Restore defaults</u>
Filter matched 10 of 11 items		Revert Apply
The multicate of 11 licens		
?	Г	Debug Close



When the program stops in the main function, we open the Memory view. From there, we can verify that the binary file is successfully downloaded into RAM. The first word denotes the SP address. The following words denote the addresses of handlers.

eclipse-workspace1 - FirstExample/User/main.c - Eclipse IDE	– 0 X
<u>File Edit Source Refactor Navigate Search Project Run Window H</u> elp	
🍕 🎄 📕 🎋 Debug 🗸 🖸 FirstExample Debug 🗸 🄅 🗋 ▾ 🔚 🐘 🗒 🗒 🗒 🗒 🗒 🖗 🕪 💷 🖷	3. 👁 .r.   🔛 🗟 🗷   🕹 🐂
物 ▼ Ο ▼ 🂁 🗁 🔗 ▼ 🕖 🖗 ▼ 🖗 ▼ 🖓 ♥ ♥ ♥ ♥ ♥ ♥   🖻	Q i 🖻   💀 🐐
🔐 main.c 🛛 📄 gcc_arm.ld 🖓 B 🏠 E	0 🛋 M 🏧 Di 🛛 🚼 P 🖓 🗖
* 41 0x20002149	✓ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
42 /* Update System Core Clock */	r: ^ ^
44 SystemCoreClockUpdate(); 45	=data_start
46 2000214a: ldr 2000214a: ldr	., [pc, #84] ; (0x200021a0 <rese = data end</rese 
47 48 /* Set GPB multi-function pins for UARTO RXD and TXD */ 2000214c: ldr	, [pc, #84] ; (0x200021a4 <rese< td=""></rese<>
49 SYS->GPB_MFPH &= ~(SYS_GPB_MFPH_PB12MFP_Msk   SYS_GPB_MFPH_PB13MFP_Msk); 2000214e: cmp	r3 2, r3
50 SYS->GPB_MFPH  = (SYS_GPB_MFPH_PB12MFP_UART0_RXD   SYS_GPB_MFPH_PB13MFP_U/ 51 /* Lock protected registers */	lt
52 SYS_LockReg(); 20002150: 1ttt	r0, [r1], #4
53 } 20002152: ldrlt.w	), [r1], #4
55⊕/* 244 strlt 20002156: strlt.w	r0, [r2], #4
56 * This is a template project for M480 series MCU. Users could based on this p 245 blt .	.oop1
58 * 2000215a: blt.n	2000214e <reset_handler+6> =0x40000100</reset_handler+6>
59 * This template application uses external crystal as HCLK source and configur 2000215c: ldr	), [pc, #72] ; (0x200021a8 <rese< td=""></rese<>
61 */	=0x59 #89 • 0x59
62 63@ int main() 308 str r	[r0]
64 { 20002162: str	., [r0, #0] =0x16
\$ 65 SYS Init(); 20002164: mov.w	., #22
67 g_seconds = 1; 310 str r	[r0]
68 UART_Open(UART0, 115200); 311 ldr r	=0x88
70 /* Connect UART to PC, and open a terminal tool to receive following messa 2000216a: mov.w	., #136 ; 0x88
71 printf("Hello World\n"); 2000216e: str	['0] , [r0, #0]
// 315 ldr m 315 // 315	=0x40000200 /* R0 = C1
74 while(1); 316 ldr r	[r0,#0x4] ; (0x200021aC <rese [r0,#0x4] /* R1 = 0x</rese 
75 } v 20002172: ldr	, [r0, #4] v
· · · · · · · · · · · · · · · · · · ·	>
📮 Console 🐰 Registers 🦹 Problems 🜔 Executables 🙀 Debugger Console 🔋 Memory 🛿	" 📑 🛃 📰 🔩 📲 🗸 8 🗖 🗖
Monitors 🕂 🙀 🕅 🕸 🖗 🕅 🖉 🖓 🖉 🖓 🖉 Mew Renderings	
	5 Ι!. μ!. m μ!. μ!. μ!. ^
0x2000181C 00000000 00000000 00000000 00000000 200021B5 200021B5 00000	Θμ!. μ!
0x20001838 20002185 2000200000000000000000000000000000000	5 μΙ. μΙ. μΙ. μΙ. μΙ. μΙ. μΙ.

Figure 3-25 Debugging in RAM

#### 3.9 Debug Executable Files Only

If the user has an executable built with **debug symbols** but may not have the relevant environment used to build the executable, he still is able to debug the executable by the following steps:

- 1. Import an executable for debugging (referring to Figure 3-26).
- 2. Click Browse following Select executable, then select an executable (referring to Figure 3-27).
- 3. Choose GDB Nuvoton Nu-Link Debugging as a Launch Configuration (referring to Figure 3-28).
- 4. Locate the GDB executable in the debug configuration (referring to Figure 3-29).
- 5. Choose the ELF file to download in the debug configuration (referring to Figure 3-30).
- 6. Add source lookup path relative to source folders (referring to Figure 3-31).
- 7. Press on the Debug button.

■ Import – □ ×	
Select Imports a C/C++ executable file. Will create a project and launch configuration for debugging.	
Select an import wizard:	
<ul> <li>C/C++</li> <li>C/C++ Executable</li> <li>C/C++ Project Settings</li> <li>Existing code as Autotools project</li> <li>Existing Code as Makefile Project</li> <li>Existing Code as Makefile Project</li> <li>Install</li> <li>Oomph</li> <li>P RPM</li> <li>Run/Debug</li> <li>Tasks</li> <li>Team</li> <li>TextMate</li> <li>Tracing</li> <li>XMI</li> </ul>	
Output > Seck Next > Finish Cancel	

Figure 3-26 Importing Executable for Debugging

### nuvoTon

Select binary parser:	Elf Parser $\checkmark$	
Select executable:	s\\\eclipse-workspace\M480\Debug\ <mark>M480.elf</mark>	Browse
O Search directory:		Browse
C/C++ Executable Fil	es:	
		Select All
		Deselect All

Figure 3-27 Selecting Executable

### nuvoTon

Import Executable			×
Choose Project			
Choose an existing proj	ect or create a new one.		
New project name:	Debug_M480.elf		
	The new project will let you debug but not build the executable.		
O Existing project:		Sear	ch
Create a Launch Co	onfiguration: GDB Nuvoton Nu-Link Debugging 💛		
Name:	Debug_M480.elf		
?	< Back Next > Finish	Cance	I

Figure 3-28 Choosing GDB Nuvoton Nu-Link Debugging

reate, manage, and run configuration	IS	Ť
ype filter text C C/C++ Application C C/C++ Attach to Application C C/C++ Attach to Application C C/C++ Attach to Application C C/C++ Postmortem Debugger C C/C++ Remote Application C C/C++ Unit C GDB Hardware Debugging ✓ GDB Nuvcton Nu-Link Debugging C Debug M480.eff C M480 Debug T Launch Group	Name:       Debug_M480.elf (1)         Main % Debugger       Startup % Source Common         OpenOCD Setup       Start OpenOCD locally         Executable:       \$(openocd_nulink_path)/\$(openocd_nulink_executable)         BDB port:       3333         Telnet port:       4444         Config options:       -f/scripts/interface/nulink.cfg -f/scripts/target/numicroM4.cfg	/ariables
	Client port: Client port: Commands: Commands: Set mem inaccessible-by-default off	/ariables
	Remote Target         Host name or IP address:         Iocalhost         Port number:         3333         Force thread list update on suspend         Remote Target	store defaults
ilter matched 11 of 13 items	Revert	Apply

Figure 3-29 Locating the GDB Executable

Debug Configurations	>
create, manage, and run configurations	·
	~~~
🖞 🕼 🗎 🗶 🖻 🏹 🕶	Name: Debug_M480.elf (1)
type filter text	📄 Main 🏇 Debugger 🍉 Startup 👍 Source 🔲 Common
C/C++ Application	Initialization Commands
C/C++ Attach to Application           C/C++ Container Launcher	☐ Initial Reset Type: init
C/C++ Postmortem Debugger	^ ·
C/C++ Remote Application	
Ct C/C++ Unit	Eastele ADM consideration
C GDB Hardware Debugging	
C Debug_M480.elf	Chip Series: NuMicro M4 V
C M480 Debug	Write Config0: 0x FFFFFFFF Config1: 0x FFFFFFFF Config2: 0x FFFF5A5A Config3: 0x FFFFFFFF
Launch Group	
	Load Symbols and Executable
	∠oad symbols and Executable
	Use project binary: M480.elf
	O Use file: Workspace File System
	Symbols offset (hex):
	✓ Load executable to flash
	O Use project binary: M480.hex
	Use file:     C:\Users\ \Desktop\M480.elf     Workspace     File System
	Executable offset (hex):
	Load executable to SRAM
	Use project binary: M480.hex
	Use file: File System
	Executable offset (hex):
	Run/Restart Commands
	Pre-run/Restart reset Type: init (always executed at Restart)
	^ ^
	Set program counter at (hev)
	Set program council at (no.).
	Restore default
	<u>Restore default</u>
lter matched 12 of 15 items	Re <u>v</u> ert Apply
3)	<u>D</u> ebug Close

Figure 3-30 Choosing the ELF File to Download

Debug Configurations Create, manage, and run configurations	gurations		
<ul> <li>Image: Construct of the second second</li></ul>	Name: Debug_M480.elf (1)          Main * Debugger       Startup       Source       Con         Source Lookup Path:       M480 - Path Relative to Source Folders       C:\Users\\eclipse-workspace\M480\CMSIS         C:\Users\\eclipse-workspace\M480\Library       C:\Users\\eclipse-workspace\M480\Library         C:\Users\\eclipse-workspace\M480\User       Search for duplicate source files on the path	nmon]	Add Edit Remove Up Down Restore Default
< >> Filter matched 16 of 19 items		Revert	Apply
?		Debug	Close

Figure 3-31 Adding Source Lookup Path

#### 4 Q&A

1. Q: Can we simultaneously debug on Eclipse, Keil and Iar?

A: No, we must stop the debug mode on Eclipse first. Then we can debug on another IDE.

2. Q: Can we simultaneously debug on Eclipse and use the Nuvoton development tools, such as ICP Programming tool?

A: No, we must stop the debug mode on Eclipse first. Then we can use them and vice versa.

3. Q: How many breakpoints and watchpoints are supported?

A: It depends on the hardware. For M0 chips, the supported number of breakpoints and watchpoints is 4 and 2, respectively. For M4 chips, the supported number of breakpoints and watchpoints is 6 and 4, respectively. For M23 chips, the supported number of breakpoints and watchpoints is 4 and 4, respectively. For now, we do not support flash breakpoints.

4. Q: How to update firmware for Nu-Link?

A: Please use ICP Programming tool or Keil to update firmware.

5. Q: How to change Flash and RAM size after projects are created?

A: Please find and open the ld script in the ld scripts folder. From there, we can change Flash and RAM size.

6. Q: Why can the application not enter the debug mode?

A: Firstly, we must install all the stuff by following the previously mentioned steps. Then check the following list:

- I. Leave the previous debug mode first if exists.
- II. Flash and RAM size must be correct.
- III. OpenOCD should not be launched before debugging. To check that, please go to Windows Task Manager or System Monitor. If an OpenOCD process has already been running, please end the process.
- IV. The target chip should not be held by other tools or IDE.
- V. The Config options field of the Debugger tab must be correct.
- VI. In the Startup tab, the **Initial Reset** type should be **init**. The **Pre-run/Restart rest** type should be **init**.
- VII. The Eclipse preferences must be correct. Please refer to the previous discussion.

#### VIII. Upgrade Nu-Link firmware and USB driver to the latest one.

- IX. Check whether the executable has been downloaded to the target chip correctly.
- X. Check SP. If it is wrong, please assign it to the correct location.
- XI. Write a correct Config value into the target chip.
- XII. If the operating system is Windows 7, please use USB 2.0 port, instead of USB 3.0 port.
- XIII. The project path must not contain any special character or whitespace, such as #\$&`.{}.
- XIV. Restart the computer.

## nuvoTon

7. Q: How to add udev rules for Nu-Link on GNU/Linux?

A: When accessing target chips via Nu-Link, GNU/Linux requires the USB permission. We can get the permission by adding udev rules for Nu-Link. Here are the steps to do that:

I. Add the User to the plugdev Group. Type the command in the Terminal:

#### sudo useradd -G plugdev \$USER

II. Add Nu-Link to udev. Type the commands in the Terminal:

cd /etc/udev/rules.d and sudo gedit 10-openocd-nulink.rules

III. Add the following text to the file

		udev	rules.d		_	_	_	t ຊ	♠)) 11:34 改 . ☷ :::
	Recent	Name					Size	Туре	Modified
	Home		10-openocd-nulink.rules				320 bytes	Text	11:31
<u> </u>	Desktop		99-vmware-scsi-udev.rules				341 bytes	Text	Dec 20 2016
ビ р	Documents	8	10-openocd-nulink.rul	es (/etc/udev/rules.d)	gedit				
A 🕹	Downloads	Ope	n 🔻 🕕						Save
	Music		nulink_usb.c ×	10-openocd-	nulink.rules	×	10-openo	cd-nulink.rules	×
🕢 🖸	Pictures	ATTRS ATTRS	{idProduct}=="511b", {idProduct}=="511c",	ATTRS{idVendor}== ATTRS{idVendor}==	'0416", MOD '0416", MOD	E="0666", GR E="0666", GR	OUP="pluge OUP="pluge	dev" dev"	
2	Videos	ATTRS ATTRS	{idProduct}=="511d", {idProduct}=="5200",	ATTRS{idVendor}== ATTRS{idVendor}==	'0416", MOD '0416", MOD	E="0666", GR E="0666", GR	OUP="pluge OUP="pluge	dev" dev"	
	Trash								
-∿	Network								- 84
. 0		•							
2	Computer								
	Floppy Disk								- 84
	Ubuntu 16.0	•							- 84
<u>Ý</u>	Connect to Server								
					Plain Text 🔻	Tab Width: 8	Ln 4	I, Col 80 🔹 🔻	INS

#### Figure 4-1 Adding Udev Rules

IV. Reloaded the new udev rules by entering the command in the Terminal:

#### sudo udevadm trigger

8. Q: How to edit string substitution for openocd\_nulink\_path?

A: The **openocd\_nulink\_path** string stores where the OpenOCD executable resides. After upgrading NuEclipse, the string may keep the previous OpenOCD path. To fix it, click **Window > Preferences**, the Preferences wizard appears. Go to **Run/Debug > String Substituion**. Find and edit the openocd\_nulink\_path to the new OpenOCD path.

be filter text	String Substitution				← → → →
General	Create and configure string s	ubstitution variables.			
ChangeLog	Variable	Value	Description	Contributed By	<u>N</u> ew
Docker	openood pulink executa	openocd.exe		com.nuvoton	
Help	openocd nulink path	C:/Program Files		com.nuvoton	<u>E</u> dit
Install/Update		ci, riogianti ico in		connerotonin	Remove
Library Hover					<u>H</u> emore
Mylyn					
Oomph					
Remote Development					
RPM					
Run/Debug					
Console					
External Tools					
> Launching					
OpenOCD Nu-Link					
Peripherals views					
Perspectives					
String Substitution					
View Management					
View Performance					
SWTChart Extensions					
Team					
Terminal					
TextMate					
Tracing					
Validation	L	1	1		

Figure 4-2 Preferences for String Substitution

9. Q: Why does Eclipse fail to update or install new software packs on Windows?

A: One of possible reasons is that the write permission for Windows folders is denied. We need to find the correct folder and allow the write permission. On Windows, the location where we place software packs is C:\Program Files (x86)\Nuvoton Tools\Packages.

10. Q: How to adjust output voltage of Nu-Link2?

A: Upgrade the NuLink2 firmware greater than version v3.08.7249, open NU\_CFG.TXT file in popup "NuMicro MCU" disk, you will see some options in NU\_CFG.TXT.

Set CMSIS-DAP=1 then re-plug in USB cable, it presents one more interface HID\_CMSIS-DAP, this is handy if you want to use CMSIS-DAP protocol.

Set LEVEL=3 then re-plug in USB cable, it presents one of output voltage, this is handy if you want to use 5V voltage.

```
[Power Control]
LEVEL=3
; Default I/O voltage (Level0: 1.8V, Level1: 2.5V, Level2: 3.3V, Level3: 5V)
[Interface configuration]
CMSIS-DAP=1
; 0 = disable
; 1 = enable
```

Figure 4-3 More Options for NuLink2

### 5 Revision History

Date	Revision	Description
2017.03.31	0.01.000	Alpha version released.
2017.06.30	1.01.000	Beta version released.
2018.09.15	1.01.013	Official version released.
2018.11.30	1.01.014	<ol> <li>Supported NUC505.</li> <li>Updated the new project wizard.</li> </ol>
2019.08.09	1.01.015	Supported M031 ,M261 and M480LD.
2020.03.06	1.01.016	Supported M031BT, NUC1311, M2354 and M479.
2020.09.30	1.01.017	Supported M030G, M071, M0A21, M251 and M471.
2021.03.12	1.01.018	Supported M030GM031G and NUC1262.
2021.12.01	1.01.019	Supported M460, I94100 and KM1M7.



Notice: Using this software indicates your acceptance of the disclaimer hereunder:

THIS SOFTWARE IS FOR YOUR REFERENCE ONLY AND PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. YOUR USING THIS SOFTWARE/FIRMWARE IS BASED ON YOUR OWN DISCRETION, IN NO EVENT SHALL THE COPYRIGHT OWNER OR PROVIDER BE LIABLE TO ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.