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<sup>™</sup> 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.

UVOTON

## ηυνοτοη

### Table of Contents

1	Overvie	w	5
1.1 1.2	Peature	uction es	5
1.3		rted Devices	
2	Installin	ng Nu-Link IAR Driver	6
2.1 2.2		n Requirements	
3	Example	e – Create and Debug a Project	7
3.1	. Start a	New Project	7
3.2	2 Debug	a project1	8
		e Watch2	
-		View Plug-in2	
-		nihosting2	
-		Console Plug-in	
-	.2.5 ETM	1 Trace	0
Э		OView Plug-in	
4	Firmwar	re Update	4
5	Revisior	n History	7

## ηυνοτοη

### List of Figures

Figure 3-1 Create a New Workspace ······7
Figure 3-2 Create a New Project
Figure 3-3 Select the Tool Chain for the Project
Figure 3-4 Save a Project ······9
Figure 3-5 Open Project Options Form
Figure 3-6 Options Form
Figure 3-7 Select Target Devices10
Figure 3-8 Select Linker Configuration File11
Figure 3-9 Select Debugger Driver
Figure 3-10 Use Nu-Link for Debugger Driver
Figure 3-11 Select "Use flash loader" Options
Figure 3-12 Add Files to the Project14
Figure 3-13 Add "Cstartup_M.s" File to the Project14
Figure 3-14 Add the Main Function in a New File15
Figure 3-15 Save the New File and Rename to "main.c"15
Figure 3-16 Add "main.c" File to the Project16
Figure 3-17 Save the Workspace16
Figure 3-18 Rebuild the Project17
Figure 3-19 Build the Project Successfully
Figure 3-20 Download Code and Debug ······18
Figure 3-21 Pause at Main Function while Debugging19
Figure 3-22 Pause the Program by Setting Breakpoints20
Figure 3-23 Terminal I/O Pane ······21
Figure 3-24 Open Memory Pane ·····22
Figure 3-25 Memory Pane22
Figure 3-26 Open Register Pane ······23
Figure 3-27 Register Pane ······23
Figure 3-28 Select Other Registers ······24

## nuvoTon

Figure 3-29 Access Other Registers ······24
Figure 3-30 Live Watch Pane·····25
Figure 3-31 Set Live Watch Update Frequency25
Figure 3-32 Live Watch Example25
Figure 3-33 NuTool - PinView
Figure 3-34 Debug Information in NuConsole Dialog Box29
Figure 3-35 Trace Setup with ETM
Figure 3-36 Initialize File for Trace Pins
Figure 3-37 Tracing Information Dialog ······32
Figure 3-38 NuTool - LCDView ······33
Figure 4-1 Firmware Update Selection Dialog Box ······ 34
Figure 4-2 Updating Firmware
Figure 4-3 Re-connect Nu-Link to Complete Firmware Update
Figure 4-4 Update Firmware Completely



### 1 Overview

#### **1.1 Introduction**

NuMicro Cortex-M IAR Driver allows IAR Embedded Workbench (EWARM) to communicate with Nuvoton on-chip debug logic.

In-system flash memory programming integrated into the driver allows the user to rapidly update target code. The IAR Embedded Workbench can be used to start and stop program execution, set breakpoints, check variables, inspect and modify memory contents, and single-step through programs to run your actual target hardware.

This document describes how to install and use NuMicro Cortex-M IAR Driver with programs written using IAR's compiling and flash tools.

#### **1.2 Features**

Nu-Link driver supports the following features. Some functions are triggered by IAR EWARM. The usage of these functions can be found in the IAR EWARM User Guide.

- Erase/program/verify Nuvoton chips. (via flash algorithm of Nu-Link IAR driver)
- Easy registers access of Nuvoton chips. (via the .ddf file of Nu-Link IAR driver)
- Support Hardware/Software/Flash breakpoints.
- Support Data breakpoints.
- Support Live Watch.
- Support various configurations for connection. (Reset options, SWD clock, etc.)

#### **1.3 Supported Devices**

Click to open the table of supported devices: Link of Supported devices.



### 2 Installing Nu-Link IAR Driver

#### 2.1 System Requirements

- **Software:** IAR Embedded Workbench (EWARM 6.1 and higher version)
- Hardware: Nu-Link ICE Adapter.

#### 2.2 Installation

Click installer file Nu-Link\_IAR\_Driver.exe. The delivered package contains a complete set of files. The user needs to run NuMicro Cortex-M IAR EWARM Driver with C-Spy interface. The following directories and files can be found after package is installed successfully:

- .\Samples: The sample project that uses Nu-Link driver for IAR.
- .\Nu-Link-IAR.dll: The driver DLL.



### 3 Example – Create and Debug a Project

### 3.1 Start a New Project

This section describes how to start a new project based on NUC100 series chips. The fast and easy way to start a new project is open an existing IAR project. To make sure the user knows about all the steps to create an IAR project, this section will start with an empty project.

1. Open IAR Embedded Workbench, and click "File"  $\rightarrow$  "New"  $\rightarrow$  "Workspace".

🔀 IAR Embedded Workbend	ch IDE				
File Edit View Project	Tools Wi	ndow Help			
New	×	File	Ctrl+N	> > = =	> <b>- - -</b>
Open	×	Workspace	N		f() 👻 🗙
Close			3	·	
Save Workspace					
Close Workspace					
Save	Ctrl+S				
Save As					
Save All					
Page Setup					
Print	Ctrl+P				
Recent Files	•				
Recent Workspaces	•				
Exit					
		_			
Create a new Workspace					

Figure 3-1 Create a New Workspace

2. Create a new project by clicking "**Project**"  $\rightarrow$  "Create New Project".

💥 IAR Embedded Wor	kbench IDE	
File Edit View Pro	oject Tools Window Help	
i D 🚅 🖬 🕼 🗍	Add Files	▼ ✓ > > > 2 ≥
Workspace	Add Group	f() - ×
	Import File List	
Files	Add Project Connection	
	Edit Configurations	
	Remove	
	Create New Project	
	Add Existing Project	~5°
	Options Alt-	F7
	Version Control System	•
	Make	F7
	Compile Ctrl-	-F7
	Rebuild All	
	Clean	
	Batch build	F8
) Create a new projec	Stop Build Ctrl+Br	eak

Figure 3-2 Create a New Project

3. Select "ARM" as the tool chain for this project, and then click "OK".

💥 IAR Embedded Wor	V IAR Embedded Workbench IDE					
File Edit View Pro	ject Tools Window Help					
🗋 🖻 🚅 🗐 🎒 Workspace	Create New Project					
	Tool chain: ARM					
Files	Project templates:					
	Empty project					
	ier asm					
	Externally built executable					
	Description:					
	Creates an empty project.					
	OK Cancel					
]						
Ready	a.					

Figure 3-3 Select the Tool Chain for the Project

4. Now you'll be prompted to save the project. Select a folder and input a project name to save it.

File	🔏 Save As					×
	🕞 🔵 🔻 📕 🕨 my_	iar_project	<b>▼</b> 49	Search my_iar_pro	ject	2
Wo	Organize 🔻 New	folder				0
F	🚖 Favorites	Name	*	Date modified	Туре	
	📜 Libraries		No items match you	ir search.		
	🖳 Computer					
	辑 Network					
		•	III			•
	File name:	nelloworld				-
	Save as type: F	Project Files (*.ewp)				•
Read	Alide Folders			Save	Cancel	

Figure 3-4 Save a Project

5. After the project is saved, right click on the project name of workspace area, and click "**Options**" to open the option setting form.

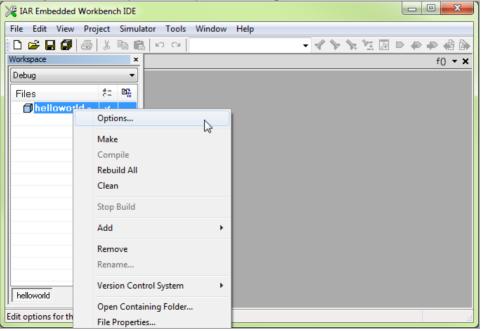


Figure 3-5 Open Project Options Form

### nuvoTon

VB IAD C.	Options for node "hellow	world"	
Verland Endi	Category: General Options C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel CMSIS DAP GDB Server IAR ROM-monitor I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro RDI ST-LINK Third-Party Driver	Target       Output       Library Configuration       Library Options       MISRA-C:200 ()         Processor variant       •       •       •         •       Core       ARM7TDMI       •         •       Device       None       •         •       Device       None       •         •       Big       •       •         •       BE3       •       •	2 23 Y
Ready		OK Cancel	

Figure 3-6 Options Form

6. The default options are neither for NUC100 series nor for Cortex-M0. Check the "**Device**" option and click the "Device" icon on the right to select the correct device name, such as "Nuvoton NUC100AN series (NUC100AN,NUC120AN)".

NO	Options for node "hellow	vorld"	1 23
File Edit			~~~~
Workspace Debug Files	C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel	Target       Output       Library Configuration       Library Options       MISRAC:200 4         Processor variant       Ore       Core       Cortex-M0         © Device       Nuvoton NUC100AN series (NUC1)       The	€ 🖗 🕪 f0 • ×
helloworl	CMSIS DAP GDB Server IAR ROM-monitor I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro RDI ST-LINK Third-Party Driver TI XDS 100	Endian mode FPU Little None  Big BE32 BE8	
Ready		OK Cancel	

Figure 3-7 Select Target Devices

7. To link and run the program in flash memory, override the default link script for this project. In this example, simply specify the link script used in the sample project "Samples\PWM".

	Options for node "hellow	vorld"	
File Edit	Category: General Options	Factory Settings	► 4 + + + + + + + + + + + + + + + + + +
Debug Files	C/C++ Compiler Assembler Output Converter	Config Library Input Optimizations Advanced Output List	
	Linker Debugger Simulator Angel CMSIS DAP GDB Server IAR ROM-monitor	Linker configuration file	
helloworl	I-jet/JTAGjet J-Link/J-Trace TI Stellaris C:\Prog Macraigor PE micro RDI ST-LINK Third-Party Driver TI XDS 100	/ aram Files\Nuvoton Tools\Nu-Link_IAR\Samples\PWM\config\NUC14	0_Flash.icf
Ready		OK Cancel	

Figure 3-8 Select Linker Configuration File

(Note: 8051 series uses .xc file)

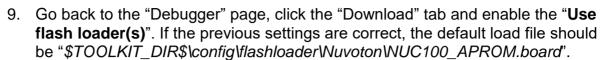
8. Select "Nu-Link" as the debugger driver for this project. On the "Debugger" page, select "**Third-Party Driver**"; and on the "Third-Party Driver" page, fill in the path of Nu\_Link-IAR.dll.

NR IAD C-	Options for node "hellow	vorld"		23
Verkspace	Category: General Options C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel CMSIS DAP GDB Server IAR ROM-monitor I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro RDI ST-LINK Third-Party Driver	Setup     Download     Images     Extra Options     Plugins       Driver     Images     Run to       Third-Party Driver     Images     Main       Simulator     Angel     Images       CMSIS DAP     CMSIS DAP       GDB Server     IAR ROM-monitor       Het/JTAGjet     J-Link/J-Trace       TI Stellaris     Macraigor       PE micro     RDI       ST-LINK     Third-Party Driver       TI XDS100     Images		× ×
helloworl Ready	TI XDS 100	ОК	Cancel	.#

Figure 3-9 Select Debugger Driver

	Options for node "hello	world"	23
File E Workspa Debug Files	Category: General Options C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel CMSIS DAP GDB Server IAR ROM-monitor I-jet/JTAGjet J-Link/J-Trace TI Stellaris	Factory Settings Third-Party Driver IAR debugger driver plugin C:\Program Files\Nuvoton Tools\Nu-Link_IAR\Nu-Link_IAR.dll	<ul> <li>▲ ●</li> <li>→ </li> <li>→ ×</li> </ul>
hellow	Macraigor PE micro RDI ST-LINK Third-Party Driver TI XDS 100	Log communication  SPROJ_DIR\$\cspycomm.log	
Ready			

Figure 3-10 Use Nu-Link for Debugger Driver



File Edit	Options for node "hello	world"	23
Workspace Debug Files The	Category: General Options C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel CMSIS DAP GDB Server IAR ROM-monitor I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro RDI ST-LINK Third-Party Driver TI XDS 100	Setup       Download       mages       Extra Options       Plugins         Attach to running target       Verify download         Suppress download       Use flash loader(s)         Override default .board file       STOOLKIT_DIRS\config\flashloader\Nuvoton\NUC1         Edit	× ×
Ready		OK Cancel	Ш

Figure 3-11 Select "Use flash loader" Options

(Note: 8051 series do not need to set flash loader)

10. After the option settings are done, click the "**OK**" button and save the project.

Since the project is initially empty and has no source code, an ASM startup code is required for a Cortex-M project. Simply copy the file "*cstartup\_M.s*" from the installation folder of Nu-Link driver to the project's directory.

To build the project, you should add the startup code and user application code to the project. Please follow the steps below:

nuvoton

 Add the *cstartup\_M.s* to the project. The default path of the startup ASM is: "C:\Program Files (x86)\Wuvoton Tools\Wu-Link\_IAR\Samples\WUC140\_PWM\cstartup\_M.s" (Nuvoton BSP for IAR project uses file *startup\_seriesName.s* as the startup ASM file, which has the same functions as *cstartup\_M.s*).

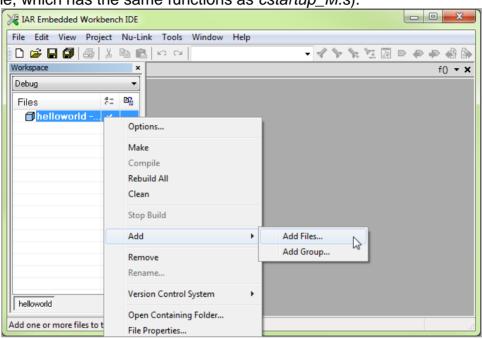


Figure 3-12 Add Files to the Project

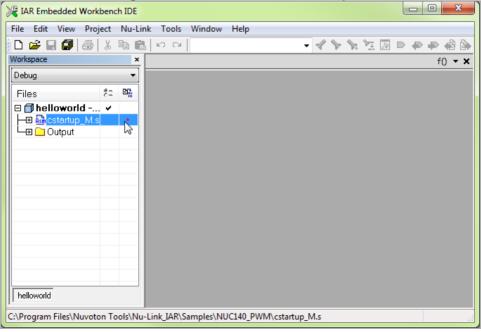


Figure 3-13 Add "Cstartup\_M.s" File to the Project

2. Now add the main function to this project. Click "File"  $\rightarrow$  "New"  $\rightarrow$  "File", and input the main function in the new file's editing pane.

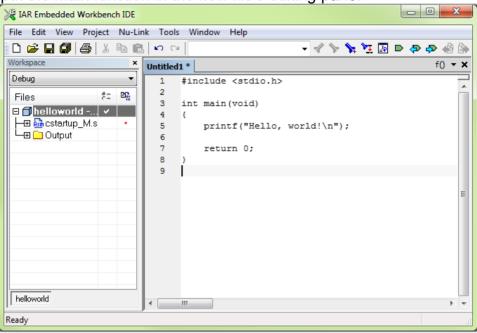


Figure 3-14 Add the Main Function in a New File

3. Click "File"  $\rightarrow$  "Save" to save the new text file as "main.c".

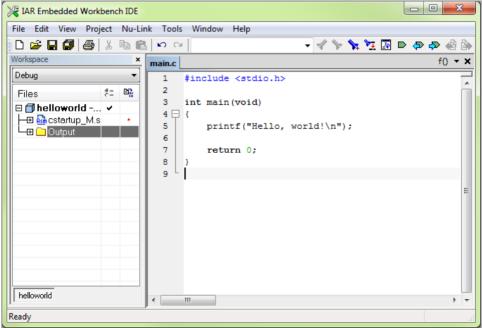


Figure 3-15 Save the New File and Rename to "main.c"

4. Add "*main.c*" to the current project. After "*main.c*" has been added, it will be listed in the "Workspace" pane.

IAR Embedded Workbench IDE		
File Edit View Project Nu-Lin	nk Tools Window Help	
D 🚅 日 🕼   🎒   X 🖻 🖻	। 🗠 🖓 🤸 🧏 🔝	D 🔊 🐢 🍓 🖿
Workspace ×	main.c	f() 🕶 🗙
Debug       ▼         Files       %*         Imain.c       *         Imain.c       *         Imain.c       *         Imain.c       *         Imain.c       *         Imain.c       *         Imain.c       *	<pre>1 #include <stdio.h> 2 3 int main(void) 4 □ { 5 printf("Hello, world!\n"); 6 7 return 0; 8 9</stdio.h></pre>	E
helloworld	<	
Ready		

Figure 3-16 Add "main.c" File to the Project

5. Click "File" → "Save Workspace". In the file dialog, input a name for this workspace and save it.

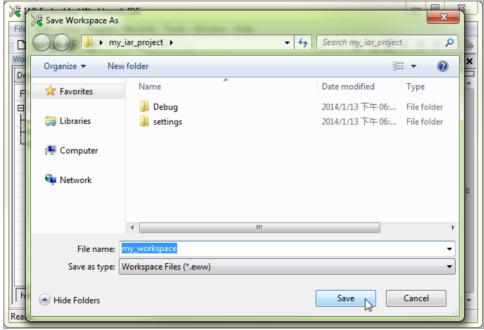


Figure 3-17 Save the Workspace

6. Now start to build the project by clicking "**Project**"  $\rightarrow$  "**Rebuild All**" or "**Make**".

🦉 my_workspace - IAR Embedded Workbench IDE					
File Edit View	Project Nu-Link Tools Window	Help	_		
0 🛥 🖬 🕼	Add Files		- 🗸 🏷 🏷 🔀 🛛	D 🗢 🗢 🌚	
Workspace	Add Group			f() 👻 🗙	
Debug	Import File List				
Files	Add Project Connection				
🗆 🗇 helloworl	Edit Configurations				
⊞ 🔂 cstartup	Remove		prld!\n");		
L-⊞ 🗀 Output	Create New Project				
	Add Existing Project				
	Options	Alt+F7		=	
	Version Control System	+			
	Make	F7			
	Compile	Ctrl+F7			
	Rebuild All	N			
	Clean	63			
helloworld	Batch build	F8			
, Clean and make the	Stop Build	Ctrl+Break			

Figure 3-18 Rebuild the Project

#### 7. The project is created successfully.

🔀 my_workspace - IAR Embedded Workbench IDE						
File Edit View Project Nu-Link Tools Window Help						
□ ☞ 및 Ø   & 1 = 1 = 1 = 1 = 1	• • •					
Workspace main.c	f() 🔻 🗙					
Debug   I #include <stdio.h></stdio.h>						
Files 👫 🛱 2						
□ □ □ helloworld ✓ 3 int main (void) 4 □ {						
H⊞ Macstartup_M.s	=					
H → C main.c 6						
7 return 0;						
	* *					
× Messages File	L *					
cstartup_M.s						
main.c						
Linking						
Total number of errors: 0	E					
Total number of warnings: 0						
	•					
Ready	Errors 0, V					

Figure 3-19 Build the Project Successfully

### 3.2 Debug a project

nuvoTon

IAR EWARM is a tool for users to debug the project easily. After the project is created successful and the target device and Nu-Link debugger is correctly connected. (Note: 8051 series didn't support the features after section 3.2.1)

 Click "Project" → "Download and Debug" to start to download and debug the project. Using the project described 3.1, the program will pause at function main().

8	🗶 my_workspace - IAR Embedded Workbench IDE				
Fi	e Edit View	Project Nu-Link Tools Window Help			
i C	) 😅 🖬 🖪	Add Files	▼ <>>> >> >> >> >> >> >> >> >> >> <>>> <>>> <>>> <>>> <>>>> <>>>> <>>>> <>>>> <>>>> <>>>>> <>>>>> <>>>>>>		
W	rkspace	Add Group	f() 🕶 🗙		
	ebug	Import File List			
	Files	Add Project Connection	- All All All All All All All All All Al		
E	🗇 helloworl				
	- 🕀 🌆 cstartup	Remove	prld!\n");		
	–⊞ 💽 main.c –⊞ 🗀 Output				
	output	Create New Project			
	elloworld	Add Existing Project			
Ľ	lellowonu	Options Alt+F7			
×	Messages	reision control system	File L *		
	cstartup_N main.c	Make F7			
	Linking	Compile Ctrl+F7			
	<b>T</b>	Rebuild All	E		
	Total num Total num	Clean			
Build		Batch build F8	-		
	•	Stop Build Ctrl+Break	4		
Do	vnload the appli		Errors 0, V La		
		Download and Debug Ctrl+D			
		Debug without Downloading			
		Make & Restart Debugger Ctrl+R			
		Restart Debugger Ctrl+Shift+R			

Figure 3-20 Download Code and Debug

## nuvoTon

💥 my_workspace - IAR Embedded Workbench IDE					
File Edit View Project Debug Disassembly Nu-Link Tools Window Help					
D 🗳 🖬 🕼 👗 🐚 🛍 🗠 🗠 🚽 🗸 🐂 💀 🗢 🧔	> 🗳 🖆				
5 · · · · · · · · · · · · · · · · · · ·					
Workspace × main.c	f() • ×				
Debug 1 #include <stdio.h></stdio.h>					
Files 😤 🛱 2					
□ □ helloworl ✓ □ □ 3 int main (void) 4 □ (					
Image: Bar Startup     5     printf("Hello, world!\n");	=				
Em Coutout					
7 return 0;					
helloworld					
× Go to v Memory v 🗈					
Disassembly					
main:					
Image: https://www.commentscondition         Ox42c: 0xb580         PUSH         {R7, LR}           printf("Hello, world!\n");         0x42e: 0x4802         LDR.N         R0, ??main_0            0x430: 0xf2ff 0xffd2         BL         printf					
printf("Hello, world!\n"); 0x42e: 0x4802 LDR.N R0, ??main 0					
Dx428:         Dx4302         LDX.W         R0, finali_0            Dx430:         0x4302         LDX.W         R0, finali_0	-				
Ready					

Figure 3-21 Pause at Main Function while Debugging



The following section introduces the debug features using the project described above.

 Move the cursor to the line that contains "printf", press "F9" to set a breakpoint, and then press "F5" to run the program until one breakpoint occurs. Now the program pauses at the breakpoint line.

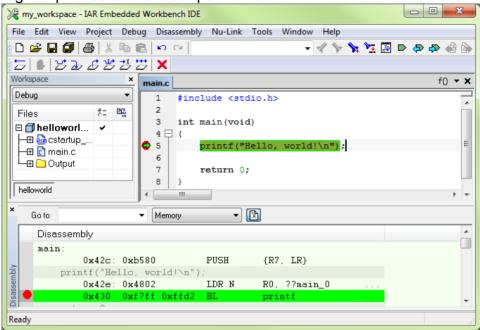


Figure 3-22 Pause the Program by Setting Breakpoints

Open the "Terminal I/O" pane by clicking "View" → "Terminal I/O". Press "F10" to step over current source line. The printf will output the message in the "Terminal I/O" pane in a semihosting way.

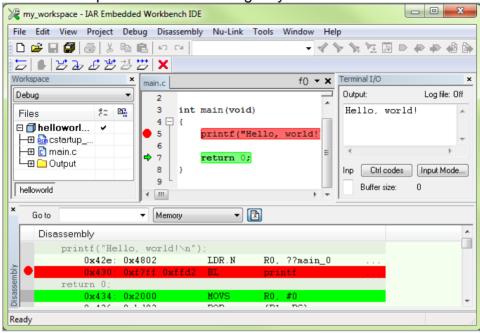


Figure 3-23 Terminal I/O Pane

Click "View" → "Memory" to open the Memory pane, in which user can view or edit the memory on the target device.

🔏 my_workspace - IAR Embedded Workbench IDE						
File Edit Vie	w Project Debug	Disassem	bly Nu-Link Tools Window Help			
🗅 🚅 🛯	Messages	+	- < > > ≥ < < < < > < < > < < > < < < < <			
	Workspace					
Workspace	Source Browser	+	f() • X Terminal I/O ×			
Debug	Breakpoints		Output: Log file: Off			
Files	Disassembly	- 1	t main(void) Hello, world!			
🗆 🗇 hell	Memory	2	printf("Hello, world!			
-⊞ 🔂 c   -⊞ 🖸 m	Symbolic Memory	43				
	Register		return 0;			
	Watch		Inp Ctrl codes Input Mode			
helloworld	Locals		Buffer size: 0			
x	Statics					
Go to	Auto		▼ 🗈			
Dist	Live Watch					
	Quick Watch		");			
a	Macro Quicklaunch		LDR.N R0, ??main_0 2 BL printf			
Disassembly	Call Stack		2 BL printf			
Disc	Stack		MOVS R0, #0 👻			
Open a new	Terminal I/O					
	T					

#### Figure 3-24 Open Memory Pane

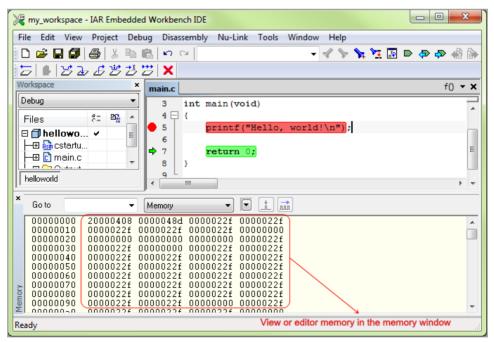
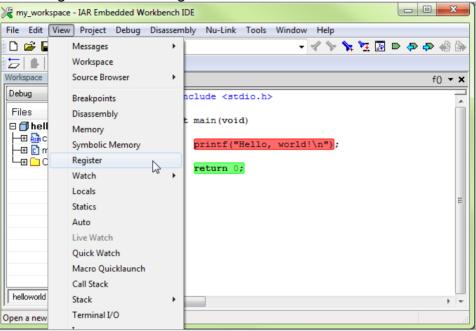


Figure 3-25 Memory Pane

● Click "View" → "Register" to open the Register pane, in which user can view or edit the registers on the target device.



### Figure 3-26 Open Register Pane

File       Edit       View Project       Debug       Disassembly       Nu-Link       Tools       Window       Help         Image: Second Se	🔀 my_workspace - IAR Embedded Workbench IDE					
Imain.c       f0 + X         Debug       Imain.c       f0 + X         Files       file       formation         Imain.c       formation       formation         Imain.c       formation       formation         Imain.c       formation       formation         Imain.c       formation       formation       formation         Imain.c       formation       formation       formation       formation         Imain.c       formation       formation <t< td=""><td>File Edit View Project Debug Disassembly Nu-Link Tools Windo</td><td>w Help</td></t<>	File Edit View Project Debug Disassembly Nu-Link Tools Windo	w Help				
Workspace       ×         Debug <ul> <li>files</li> <li>finclude <stdio.h></stdio.h></li> <li>sint main (void)</li> <li>finclude <stdio.h></stdio.h></li> <l< td=""><td>[D 22 문 💭 🚭 👗 🛍 🛍 🗠 여 🗌</td><td>✓ &lt;&gt;&gt;&gt; &lt; &gt;&gt; &lt;&gt;&gt;&gt; &gt;&gt; &lt;&gt;&gt;&gt; &gt;&gt; &lt;&gt;&gt;&gt; &gt;&gt; &gt;</td></l<></ul>	[D 22 문 💭 🚭 👗 🛍 🛍 🗠 여 🗌	✓ <>>> < >> <>>> >> <>>> >> <>>> >				
Debug       Imain.c	〒  ●   芝 <u> </u>					
Debug       1       #include <stdio.h>         Files       %       %         B       helloworl          M       %       %         M       %<!--</td--><td>Workspace × main.c f0 • ×</td><td>Register ×</td></stdio.h>	Workspace × main.c f0 • ×	Register ×				
Files       Cm Wig         Image: Construct product of the second sec		Current CPU Registers 🔹				
3       int main (void)         4       {         9       printf("Hello, ws         8       >         9       Peturn 0;         8       >         9       Peturn 0;         1       0x00000001         R6       0x00000001         R6       0x0000000         R8       0x55000000         R9       0x4000020         R11       0x0000000         R9       0x4000020         R11       0x0000000         R9       0x4000020         R11       0x0000000         R9       0x4000020         R11       0x0000000         R9       0x40000020         R11       0x0000000         R13       (SP)         PSR       0x00000000         PSR       0x00000000         PSR       0x0000000         PSR       0x0000000         PSR       0x0000000         PSR       0x0000000	Files 2 2	R0 = 0x00000000				
Image: Control of the second secon	3 int main (woid)	R1 = 0x00000435				
Image: Constraint of the line of th						
Harman.c       6         Harman.c       7         Barman.c       7						
Image: Second state of the second s						
8       }         9       R7         9       0x000000000         R8       0x5500000         R9       0x4000020         R10       0x00030201         R11       0x0042000         R12       0x90410100         R13       (SP)         0x2000400       R14         (LR)       0x00000000         HAPSR       0x60000000         HPSR       0x00000000         HPSR       0x000000000         HPSR       0x00000000         HPSR       0x00000000						
9       R       0x55000000         R8       0x55000000         R9       0x4000020         R10       0x0030201         R11       0x0040200         R12       0x90A10100         R13       (SP)         View or edit registers       PSR         PSR       0x00000000         PSR       0x0000000         PSR       0x0000000         PSR       0x00000000         PSR       0x01000000						
R9       0x4000020         R10       0x0030201         R11       0x00402000         R12       0x00402000         R13       (SP) = 0x20000400         R14       (LR) = 0x000024F         APSR       = 0x60000000         IPSR       = 0x00000000         HAPSR       = 0x00000000         HSPSR       = 0x00000000         HSPSR       = 0x00000000						
R10       = 0x00030201         R11       = 0x00030201         R11       = 0x000402000         R12       = 0x90A10100         R13       (SP) = 0x2000400         R14       (IR) = 0x000002AF         HAPSR       = 0x60000000         HIPSR       = 0x00000000         HEPSR       = 0x00000000         HIPSR       = 0x00000000						
Niew or edit registers         Hall         Hall <td></td> <td></td>						
R12       = 0x90A10100         R13       (SP)         Niew or edit registers       R14         IPSR       = 0x0000000         IPSR       = 0x0000000         IPSR       = 0x0000000         IPSR       = 0x01000000         IPSR       = 0x01000000         IPSR       = 0x01000000						
R13 (SP) = 0x2000400         R14 (LR) = 0x00002AF         PAPSR = 0x6000000         PIPSR = 0x0000000         PEPSR = 0x0000000         PEPSR = 0x0000000         PEPSR = 0x0000000						
R14 (LR) = 0x000002AF         PAPSR = 0x60000000         PAPSR = 0x60000000         PAPSR = 0x00000000         PAPSR = 0x000000000         PAPSR = 0x000000000         PAPSR = 0x00000000         PAPSR = 0x000000000         PAPSR = 0x00000000         PAPSR = 0x000000000         PAPSR = 0x000000000         PAPSR = 0x000000000    <						
APSR         =         0x6000000           Usew or edit registers         =         1PSR         =         0x0000000           EPSR         =         0x0000000         =         IPSR         =         0x0000000						
View or edit registers     IPSR = 0x00000000       helloworld     + +						
helloworld	View or edit registers					
helloworld						
	helloworld					
Ready						
	Ready					

Figure 3-27 Register Pane

In the same Register pane, the user can also access registers other than CPU registers.

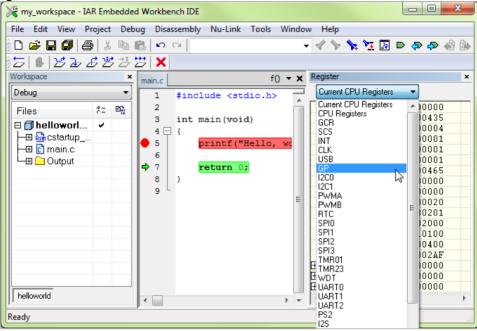


Figure 3-28 Select Other Registers

🔀 my_workspace - IAR Embedded Workbench IDE					
File Edit View Project Deb	ug Disassembly Nu-Link Tools Window Help				
0 🛩 🖬 🕼   🎄   👗 🖻 (	■ 1 × ×   × × × × × × × × × × × × × × × ×	i > <b>+</b> + <b>+ +</b>			
5 8 8 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	₩ <b>X</b>				
Workspace ×	main.c f() - × Register	×			
Debug	1 #include <stdio.< td=""><td>-</td></stdio.<>	-			
		= 0xFFFFFFFF			
Files 🕅 🖾	3 int main (void)	= 0x00000000			
🗆 🗇 hellowori 🗸		= 0x0000 🗟			
	GP_GPIOA_DOUT	= 0x0000FFFF			
🕂 🕂 🔁 🖸 main.c	E E E E E E E E E E E E E E E E E E E				
🖵 🖵 Output	HGP_GPIOA_PIN	= 0x0000FFFF			
	B } EFFECTION BEN	= 0x00000000 = 0x00000000			
	9	= 0x00000000			
	E E GP GPIOA ISRC	= 0x00000000			
	<b>⊞</b> GP_GPIOB_PMD	= 0xFFFFFFFF			
	<b>⊞GP_GPIOB_OFFD</b>	= 0x00000000			
	<b>⊞GP_GPIOB_DOUT</b>	= 0x0000FFFF			
	<b>⊞GP_GPIOB_DWASK</b>				
	<b>⊞GP_GPIOB_PIN</b>	= 0x0000FFFF			
	<b>⊞GP_GPIOB_DBEN</b>	= 0x00000000			
	<b>⊞GP_GPIOB_IMD</b>	= 0x00000000 = 0x00000000			
helloworld	<b>H</b> GP_GPIOB_IEN	- 0X0000000			
II		P			
Ready					

Figure 3-29 Access Other Registers

#### 3.2.1 Live Watch

nuvoton

Nu-Link supports IAR "Live Watch" function which can periodically update the static or global variables and display the values in the "Live Watch" pane while target CPU running.

 In Debug mode, Select "View" → "Live Watch", and then drag and drop the variables you want to observe in the "Live Watch" pane.

main.c		main() 🗸 🗙	Live Watch			×
14			Expression	Value	Location	Туре
<b>\$</b> 15	i	nt main (void)	count	0	0x200000A0	int
16	무 🕻		Click to			
17		static int count = 0;				
18						
19		for(;1;)				
20		<pre>printf("%d\n", count++);</pre>				
0 21		E				
22		return 0;				
23	}					
۲ <u>أأ</u>	1		•	III		•

Figure 3-30 Live Watch Pane

Right-click on the "Build" window, select "Options" → "Debugger", and you can set the update frequency.

×	Messages Building config Updating build		Step into functions  All functions  Functions with source only		
		Register Filter	Update intervals (milliseconds)		
	Configuration i		Live watch:	1000	
			Memory window:	1000	

Figure 3-31 Set Live Watch Update Frequency

• Click the "Go" icon, and you can observe the variables. The automatic update will be performed periodically.

main.c	main() 👻 🗙	Live Watch		×	Terminal I/O		
14	-	Expression	Value	Location	Output:		
15	int main (void)	count	1427	0x20000	1426 1427		
16	₽‹ L	≺click to			1427		
17	static int count = 0;				1429		
18					1430		
19	for(;1;)				1431		
20	printf("%d\n", count				4		
0 21	E E						
22	return 0;				Input: Ctrl codes		
23	2				Buffer size:		
•		۰ III.		F.	Duriti size.		

Figure 3-32 Live Watch Example



#### 3.2.2 PinView Plug-in

In Debug mode, user can call "NuTool – PinView" from "Nu-Link"  $\rightarrow$  "NuTool – PinView", and then check the correctness of pin assignment through GUI.

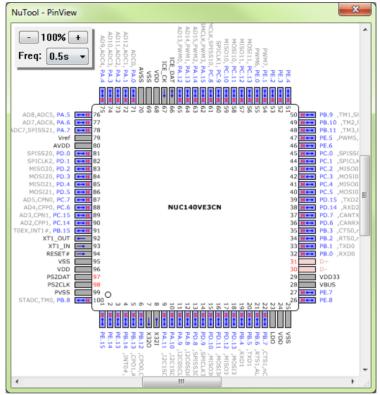


Figure 3-33 NuTool - PinView

#### 3.2.3 Semihosting

The Nu-Link supports some semihosting functions for user to easily output messages in IAR EWARM IDE. The default option settings support semihosting library. As the example project above, printf outputs the message in the "Terminal I/O" pane.

The only problem in this example is that it lacks the support of outputting message on real target device. If the program does not run in ICE debug mode, but run directly in flash memory, it would be halted by function printf.

To solve the problem described above, please follow the steps below:

- 1. Open the "C:\Program Files (x86)\Wuvoton Tools\Wu-Link\_IAR\Samples\WUC140\_PWM" directory and copy the following two files into your project folder:
  - SH\_retarget.c
  - SH\_startup\_NUC1xx.s
- 2. Then open your project, and
  - Replace *retarget.c* (if exists) with *SH\_retarget.c*.
  - Replace startup\_NUC1xx.s or startup\_M.s (if exists) with SH\_startup\_NUC1xx.s.
- 3. To reduce the code size, you can undefine the macros in *SH\_retarget.c* to disable the real UART or semihosting terminal or both.
  - #define DEBUG\_ENABLE\_UART
  - #define DEBUG\_ENABLE\_SEMIHOST
- 4. Build and run again. Now you can test the semihosting function again.
  - If debugging with Nu-Link:

Select "View"  $\rightarrow$  "Terminal I/O" to open the "Terminal I/O" pane.

When the program calls printf, it will show messages in the "Terminal I/O" pane.

• If running program in flash directly without the debugger being connected:

When program calls printf, it will show messages in real UART if DEBUG\_ENABLE\_UART is defined.



3.2.4 NuConsole Plug-in

In addition to semihosting, the Nu-Link provides another I/O mechanism without affecting the target's real time behavior. By using standard debug port SWD, It doesn't need any additional pin or hardware. The only requirement is that the target application should reserve a buffer space (hereinafter referred to as **InfoBlock**) in SRAM in order to store control settings and communicate I/O data between NuConsole and target. To use the NuConsole functions, please follow the steps below:

- 1. Open the "C:\Program Files (x86)\Nuvoton Tools\Nu-Link\_IAR\NuConsole\_Sample" directory and copy the following files into your project folder:
  - NuConsole.h/.c
  - NuConsole\_Config.h
  - NuConsole\_Retarget.c
- 2. Set up the project
  - Replace *retarget.c* (if exists) with *NuConsole\_Retarget.c*.
  - Add *NuConsole.c* to the project and include *NuConsole.h* in the corresponding files.
- 3. Configure InfoBlock
  - In *NuConsole\_Config.h*, adjust the appropriate size of TX/RX buffers according to the application requirements and hardware limitation. Also, the TX buffer can be configured to be blocking or non-blocking.
  - Call NuConnsole\_Init() function to initialize InfoBlock before doing I/O operations (e.g. printf()).
- 4. Enable the "Generate linker map file" option and build the project
  - In the linker map "project\_name.map" file under "project\_path/Debug/List" directory, find the value of symbol NuConsole\_InfoBlock variable declared in NuConsole.c to get the memory address of InfoBlock.
- 5. Download and run

## In Debug mode, select "Nu-Link" → "NuConsole" to invoke the control dialog. Set up the address of InfoBlock, and then click the start button to process I/O data.

7. Now when you execute the program and run the I/O statements, you can see debug information in the NuConsole dialog as shown below:

NuConsole				8
Stop				
Hello, Nuvoton!				
Info Block	nsmit Text			
	IISINC TEXT		Ente	er 📄
Address 0x20000478	pend nothing 👻		Time-out Limit (s)	; <b>•</b>
× Expression	Value	Location		
	<struct></struct>	0x20000478		

Figure 3-34 Debug Information in NuConsole Dialog Box

IUVOTON

### 3.2.5 ETM Trace

To start Embedded Trace Macrocell (ETM) tracing on Nuvoton Cortex®-M4/M23 devices, please connect to the device using the Nu-Link2-Pro with 20-pin connector and follow the steps below.

1. Configure the NuTrace.

- Select "NuTrace" from "Nu-Link" menu.
- In "Trace Port" select Sync Trace Port with 4 bit data. It is possible to use other bit sizes but best to use the largest to increase the bandwidth.
- In "Capture Mode", specify whether trace data is collected before or after a trigger.
  - Trace After: Capture the trace information after the trigger point and stop capturing when trace buffer is full.
  - Trace Before: Capture the most recent trace information before CPU is stopped.
- Select Trace Enable and ETM Trace Enable.
- Click OK to save the changes.

NuTrace Setup		
Core Clock: 192.000000 MHz	✓ Trace Enable	ETM Trace Enable
Capture Mode	Timestamps Enable Prescaler: 1	Trace Events CPI: Cycles per Instruction
Trace Port Sync Trace Port with 4-bit Data	PC Sampling Prescaler: 64*1 Periodic Period: <disabled> on Data R/W Sample</disabled>	<ul> <li>EXC: Exception overhead</li> <li>SLEEP: Sleep Cycles</li> <li>LSU: Load Store Unit Cycles</li> <li>FOLD: Folded Instructions</li> <li>EXCTRC: Exception Tracing</li> </ul>
ITM Stimulus Ports Enable: 0x00000000 Privilege: 0x00000000 Port 3		Port 8 7 Port 0
		Cancel OK

Figure 3-35 Trace Setup with ETM

2. In **Setup macros**, please insert the script file to initialize the device's trace pins when starting the debugger. The following is an example script file.

## nuvoton

Options for node "TIMER_Perio	odicINT"
Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator CADI CMSIS DAP GDB Server Viet/TACEA	Setup       Download       Images       Extra Options       Multicore       Plugins         Driver       Images       Run to       Images       Multicore       Plugins         Third-Party Driver       Imain       Imain       Images       Setup macros       Images       Use macro file(s)         ImpleCode\StdDriver\TIMER_PeriodicINT\IAR\M480_TP.mac       Images       Images       Images       Images
4 }	set() eMemory32(0x000EEEEE, 0x40000054, "Memory");

Figure 3-36 Initialize File for Trace Pins

Note: The Nu-Link driver with the version v2.07 or later will automatically setup the trace pins when starting the debugger. The user does not need to do the above configuration.

3. After doing above settings, user must start the debugger. In Debug mode, please select "Nu-Link"  $\rightarrow$  "NuTrace" to invoke the tracing information dialog, and it will show every single executed instruction in the current application as shown below.

NuTrace									
									Search
ЕТМ	index	address	opcode	instruction			source		
ETM	#0	0x00001370	480D,	LDR	RO,	[PC, #52]; 0x000013A8	LDR	RO,	=0x40000100
ETM	#1	0x00001372	2159,	MOV{S}	R1,	#89	LDR	R1,	=0x59
ETM	#2	0x00001374	6001,	STR	R1,	[R0, #0]	STR	R1,	[R0]
ETM	#3	0x00001376	2116,	MOV{S}	R1,	#22	LDR	R1,	=0x16
ETM	#4	0x00001378	6001,	STR	R1,	[R0, #0]	STR	R1,	[R0]
ETM	#5	0x0000137A	2188,	MOV{S}	R1,	#136	LDR	R1,	=0x88
ETM	#6	0x0000137C	6001,	STR	R1,	[R0, #0]	STR	R1,	[R0]
ETM	#7	0x0000137E	480B,	LDR	RO,	[PC, #44]; 0x000013AC	LDR	RO,	=0x40000200
ETM	#8	0x00001380	6841,	LDR	R1,	[R0, #4]	LDR	R1,	[R0,#0x4]
ETM	#9	0x00001382	F4414180,	ORR	R1,	R1, #0x00004000	ORR	R1,	R1, #0x4000
ETM	#10	0x00001386	6041,	STR	R1,	[R0, #4]	STR	R1,	[R0,#0x4]
ETM	#11	0x00001388	4809,	LDR	RO,	[PC, #36]; 0x000013B0	LDR	RO,	=0x40007000
ETM	#12	0x0000138A	6841,	LDR	R1,	[R0, #4]	LDR	R1,	[R0,#4]
ETM	#13	0x0000138C	F0410102,	ORR	R1,	R1, #0x0000002	ORR	R1,	R1,#2
ETM	#14	0x00001390	6041,	STR	R1,	[R0, #4]	STR	R1,	[R0,#4]
ETM	#15	0x00001392	6841,	LDR	R1,	[R0, #4]	LDR	R1,	[R0,#4]
ETM	#16	0x00001394	F0410104,	ORR	R1,	R1, #0x0000004	ORR	R1,	R1, #4
ETM	#17	0x00001398	6041,	STR	R1,	[R0, #4]	STR	R1,	[R0,#4]
ETM	#18	0x0000139A	4806,	LDR	RO,	[PC, #24]; 0x000013B4	LDR	RO,	=SystemInit
ETM	#19	0x0000139C	4780,	BLX	RO		BLX	RO	-
ETM	#20	0x00000CB0	4816,	LDR	RO,	[PC, #88]; 0x00000D0C	SCB->0	PACR	= ((3UL << 10*2)
ETM	#21	0x00000CB2	6801,	LDR	R1,	[R0, #0]			
ETM	#22	0x00000CB4	F4410170,	ORR	R1,	R1, #0x00F00000			
ETM	#23	0x00000CB8	6001,	STR	R1,	[R0, #0]			
ETM	#24	0x00000CBA	4815,	LDR	RO,	[PC, #84]; 0x00000D10	FMC->C	YCCTL	. = (FMC->CYCCTL & ~FM
ETM	#25	0x00000CBC	6801,			[R0, #0]			
ETM	#26	0x00000CBE	0909,	LSR{S}	R1,	R1, #4			
ETM	#27	0x00000CC0	0109,	LSL(S)	R1,	R1, #4			
ETM	#28	0x00000CC2				R1, #0x0000008			
ETM	#29	0x00000CC6	6001,			[R0, #0]			

Figure 3-37 Tracing Information Dialog

## ηυνοτοη

#### 3.2.6 LCDView Plug-in

In Debug mode, user can call "NuTool – LCDView" from "Nu-Link"  $\rightarrow$  "NuTool – LCDView". For details, please refer to the following link to the user manual:

https://www.nuvoton.com/resourcedownload.jsp?tp\_GUID=UG1320220524115039

NuTool - LCDView	-	
Project Name 🥖		.04.0000 ^
Create Mode Emulator Mode		
🖶 🗄 🛃 🔤 🛃 🗐		
Canvas           Width:         1000         px           Height:         500         px		
□ Show grid		
PROJ_M258		
M254SD3AE		
PID: M254SD3AE		
	Icon ID: Path ID: Com: Seg: Address: bit: [Coordinate] Top-Left: Top-Right: Bottom-Left: Bottom-Right:	
AW 🖬 💠	Seg 0         Seg 1         Seg 2         Seg 3         Seg 4         Seg 5         Seg 6	Seg 7
% <sup>ppm</sup> °C°F√	Com 0	
	Com 1	

Figure 3-38 NuTool - LCDView



### 4 Firmware Update

When trying to debug a project using IAR EWARM, it will check the firmware version first. If the current firmware version is not consistent with the installed Nu-Link IAR Driver, a dialog box will pop up as follows:

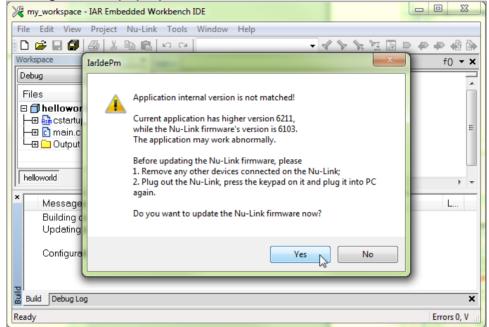


Figure 4-1 Firmware Update Selection Dialog Box

Click "Yes" to update firmware or click "No" to cancel.

## nuvoTon

🔀 my_workspace - IAR Embedded Workbench IDE	, o X
File Edit View Project Nu-Link Tools Window Help	
□ ☞ 目 Ø ● ↓ 咱 电   □ □	D 🗭 🚯 🖗
Workspace main.c	f() 🕶 🗙
Debug 1 #include <stdio.h></stdio.h>	
Files 😤 🛱 2	
B helloworld - v 3 int main (void)	
H B Const Nu-Link	E
Verify 45%	
helloworld	
Mess	L
Building configuration, nonowonal Debag	
Updating build tree	
Configuration is up-to-date.	
Build Debug Log	×
Ready	Errors 0, V

Figure 4-2 Updating Firmware

When update is complete, it is necessary to recreate a connection between Nu-Link and PC. Please plug out the Nu-Link from PC and plug in again.

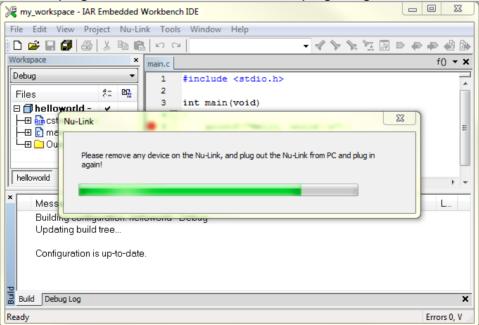


Figure 4-3 Re-connect Nu-Link to Complete Firmware Update

## nuvoTon

🔀 my_workspace - IAR Embedded Workbench IDE	E 🗆 🖾 🕮
File Edit View Project Nu-Link Tools W	Vindow Help
D 🖙 🖬 🗿 🎒 🐰 🖻 💼 🗠 🖂	- < > > E 🛛 > + + + + + + + + + + + + + + + + + +
Workspace x main.c	f() 🕶 🗙
Debug • 1 #ir	nclude <stdio.h></stdio.h>
Files 🥋 📴 2	IarldePm X
□	
	Update firmware, OK!
La Cu	
Update 100%	
helloworld	ОК + -
× Mess	
Building consignation. Inclowed a con	
Updating build tree	-5
Configuration is up-to-date.	
Configuration is up to date.	
Build Debug Log	×
Ready	Errors 0, V

Figure 4-4 Update Firmware Completely

If you use Nu-Link2 adapter, you can also use drag and drop to upgrade firmware. Press the button on Nu-Link2 adapter and plug in USB cable, you will see a disk name "Nu-Link2". Drag and drop bin file into it will upgrade Nu-Link2 firmware. (Note: if you see disk name "NuMicro MCU" it will upgrade the target device firmware instead of Nu-Link2 itself.)

## ηυνοτοη

### 5 Revision History

Revision	Date	Description
1.02	2010/07/07	Added the IAR's project settings.
1.03	2010/07/22	Added the M05x series.
1.17	2011/08/12	Used new configuration for NUC100 series.
1.18	2013/07/01	Added the "Nuvoton Announcement" and "Firmware Update" chapters.
1.19	2014/02/24	Changed document format, and updated all figures.
1.20	2014/10/17	Changed document and figure format.
2.01	2017/02/23	Added the "NuConsole" chapter.
2.07	2019/04/02	Added the "ETM Trace" chapter.

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.

> 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.