

NuEclipse

用户手册

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

内容目录

1	序言	6
1.1	支持芯片	6
2	系统需求	7
3	快速使用	8
3.1	安装	8
3.1.1	执行NuEclipse安装软件在微软Windows上	8
3.1.2	解压NuEclipse Tar文件在GNU/Linux上	9
3.1.3	检验Eclipse Preferences	10
3.2	执行Eclipse	12
4	工程开发教学	13
4.1	选择Workspace	13
4.2	新工程向导	14
4.3	汇入现有工程	16
4.4	建置设定	17
4.5	调试配置	18
4.5.1	Debugger标签页	19
4.5.2	Startup标签页	20
4.6	调试视图	21
4.6.1	寄存器视图	21
4.6.2	内存视图	22
4.6.3	反汇编视图	23
4.6.4	外围寄存器视图	24
4.7	检测点	30
4.8	在RAM中调试	32
4.9	调试执行档	36
4.10	Cortex-A、多核SMP、AMP系统配置	42
4.10.1	事先准备	42
4.10.2	调试单核Cortex-A的配置	43
4.10.3	SMP模式下调试双核Cortex-A的配置	47
4.10.4	在单个 Cortex-A 内核和 Cortex-M (AMP) 上进行多核调试	47
4.10.5	双 Cortex-A 和 单 Cortex-M 上的多核调试	49
4.10.6	AARCH32 程序调试於A35双核 (SMP)	50
4.11	NuMicro 8051 1T配置	53

4. 11. 1 汇入和建置工程 53

4. 11. 2 调试配置 53

4. 11. 3 监控不同的内存空间 54

5 Q&A.....57

6 修订历史.....61

图片目录

图 3-1 NuEclipse安装向导..... 8

图 3-2 Install.sh脚本..... 9

图 3-3 Preferences设置Global Tools Paths的对话框.....10

图 3-4 Preferences设置OpenOCD Nu-Link的对话框11

图 3-5 Eclipse.exe和相关文件夹.....12

图 4-1 选择Workspace.....13

图 4-2 新工程向导.....14

图 4-3 目标处理器的设定.....15

图 4-4 汇入工程.....16

图 4-5 建置设定.....17

图 4-6 调试配置.....18

图 4-7 设置Debugger标签页.....19

图 4-8 设置Startup标签页.....20

图 4-9 寄存器视图.....21

图 4-10 内存视图.....22

图 4-11 点选Instruction Stepping Mode按键.....23

图 4-12 反汇编视图.....23

图 4-13 开启Packs观点.....24

图 4-14 如何下载套件.....25

图 4-15 数据库位置.....26

图 4-16 安装新唐SFR文件.....27

图 4-17 选择设备.....28

图 4-18 外围寄存器视图.....29

图 4-19 切换检测点.....30

图 4-20 C/C++ 检测点属性对话框.....31

图 4-21 在断点视图中新增的检测点.....31

图 4-22 内存配置.....32

图 4-23 修改ld脚本.....33

图 4-24 设定调试配置.....34

图 4-25 在RAM中调试.....35

图 4-26 汇入执行档进行调试.....36

图 4-27 选择执行档.....37

图 4-28 选择GDB Nuvoton Nu-Link Debugging作为启动配置.....38

图 4-29 指定GDB执行档所在位置39

图 4-30 选择欲载入的ELF档40

图 4-31 添加源码查找路径.....41

图 4-32 设置MA35D1的Startup标签页43

图 4-33 如何取消命令超时上限.....44

图 4-34 禁用CPU1设置.....45

图 4-35 设置Debugger标签页以在MA35D1上进行单核调试46

图 4-36 设置Debugger标签页以在M4内核上进行多核调试47

图 4-37 配置启动群组.....48

图 4-38 如何停用SMP49

图 4-39 多核调试.....49

图 4-40 指定AArch64 GDB执行档所在位置51

图 4-41 首次进入调试模式设置.....51

图 4-42 第二次进入调试模式设置52

图 4-43 AArch32 程序调试.....52

图 4-44 建置NuMicro 8051 1T工程.....53

图 4-45 Debugger标签页中的NuMicro 8051 1T调试配置54

图 4-46 使用内存浏览器访问内存空间.....55

图 4-47 使用内存视图访问内存空间.....55

图 4-48 使用监视表达式视图访问变量.....56

图 5-1 新增Udev规则58

图 5-2 Preferences设置String Substitution的对话框59

图 5-3 NuLink2 的更多选项60

1 序言

NuEclipse是个跨平台MCU嵌入式系统的软件开发环境。它包含一系列的Eclipse插件和工具。该插件可让用户创建、建置和调试基於ARM和NuMicro 8051 1T且在Eclipse开发环境下的工程。它拥有下面这些特色。

- 藉由新工程向导来创建工程: 新工程向导为不同的目标芯片提供一些模板工程。
- 藉由**GNU ARM 工具键来建置工程**: 该工具键包含ARM的GCC编译程序。用户可以不受限制地利用它来建置工程。
- 藉由**GDB来调试工程**: 用户可暂停、单步、运行和监视目标芯片。存取内存及闪存是允许的。支持设置硬件断点及检测点。此外,用户可以抹除目标芯片和编程用户配置。

通过**NuEclipse**,用户可在Eclipse开发环境下开发新唐微控制器的工程。

1.1 支持芯片

请参照user manual文件夹下的Supported_chips.htm。

2 系统需求

若用户想要在他们的计算机上执行NuEclipse，这里列出系统需求。

	最低需求	建议规格
操作系统	Windows®7 x64或GNU/Linux	Windows®10 x64或Ubuntu 18.04 LTS
GNU ARM Embedded 工具链	10.3-2021.10	目前最新版本

注意:为了在Linux上得到完整可用且舒适的使用环境，推荐Linux发行版为Ubuntu 18.04 LTS (64-bit)。

3 快速使用

3.1 安装

为安装NuEclipse，请根据你的操作系统执行下面步骤：

1. 执行NuEclipse安装软件在微软Windows上。
2. 解压NuEclipse tar文件在GNU/Linux上。

3.1.1 执行NuEclipse安装软件在微软Windows上

在Windows上，只要执行NuEclipse安装软件便能很轻松地安装好NuEclipse。安装过程中，向导会问用户是否要安装GNU ARM Eclipse Windows Build Tools和GNU ARM Embedded Toolchain。对NuEclipse而言，它们的存在是必需的。

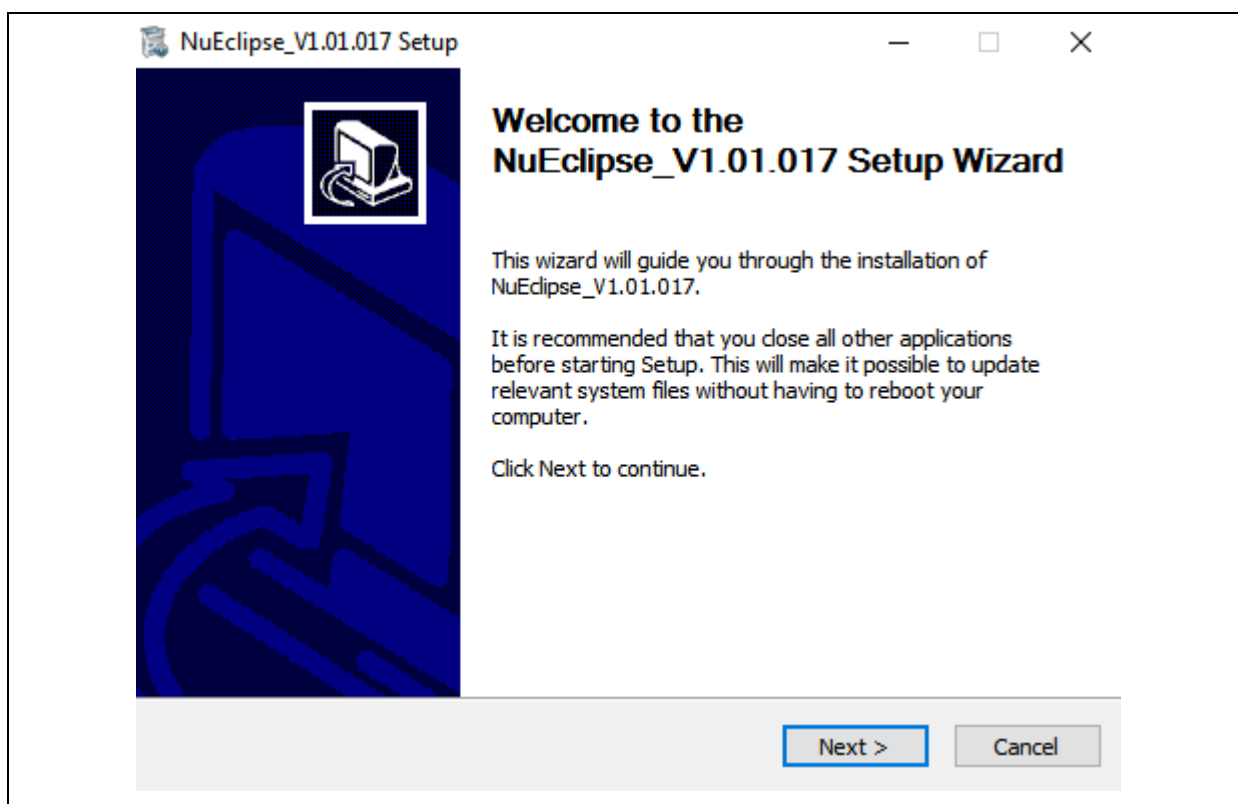


图 3-1 NuEclipse 安装向导

3.1.2 解压NuEclipse Tar文件在GNU/Linux上

在GNU/Linux上，只要解压NuEclipse tar文件便能很轻松地安装好NuEclipse。随后，执行install.sh脚本并登出重新登入以完成安装程序。请不要使用sudo命令来执行脚本。

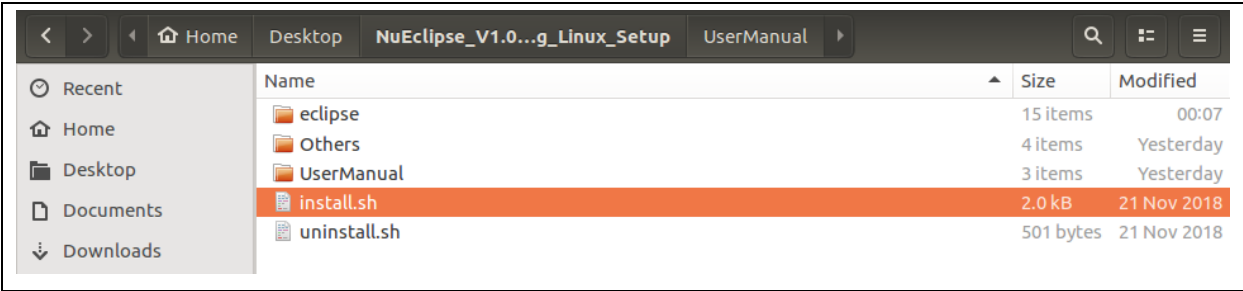


图 3-2 Install.sh 脚本

3.1.3 检验Eclipse Preferences

在Windows上安装结束后，Eclipse preferences会自动写入对应的值。为检验它们，单击Window > Preferences。Preferences对话框会出现。前往C/C++ > Build > Global Tools Paths并确认Build tools和Toolchain的文件夹路径是否正确地指向先前安装软件所装的位置。确认完毕，點選Apply按键来生效。在GNU/Linux上，Build tools文件夹路径并不需要。该路径应为空白。

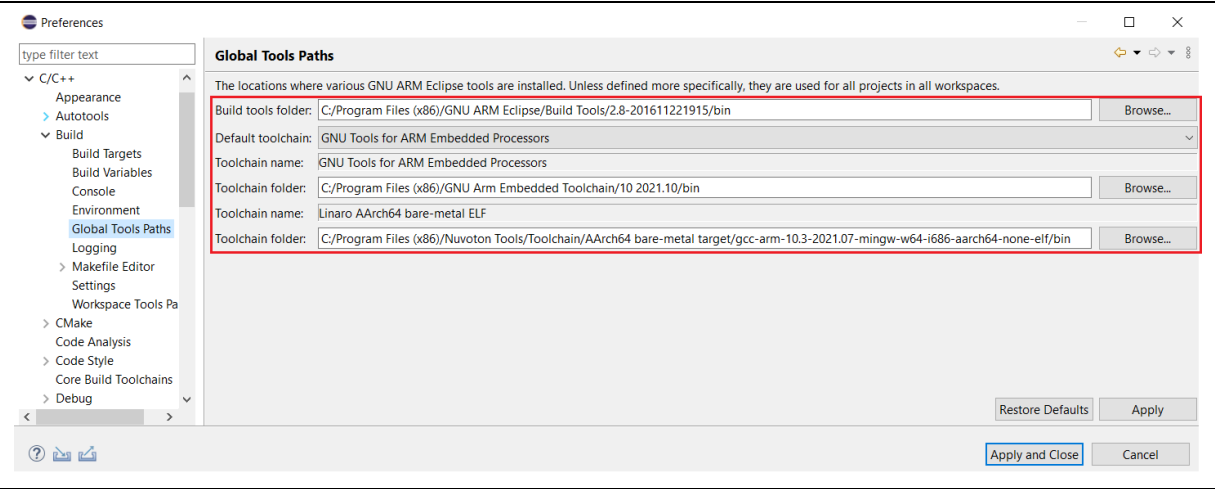


图 3-3 Preferences 设置 Global Tools Paths 的对话框

接下来，前往Run/Debug > OpenOCD Nu-Link并确认OpenOCD文件夹路径是否正确地指向安装软件所存放OpenOCD 执行档。例如，在微软Windows上，OpenOCD 文件夹的路径可能是C:/Program Files (x86)/Nuvoton Tools/OpenOCD。同样地，在GNU/Linux上它则可能是/usr/local/OpenOCD。该OpenOCD 执行档是由新唐提供且为Nu-Link客制化。若用户用别的OpenOCD执行档，OpenOCD和Nu-Link可能不能正常工作。确认完毕，点选Apply按钮来生效。

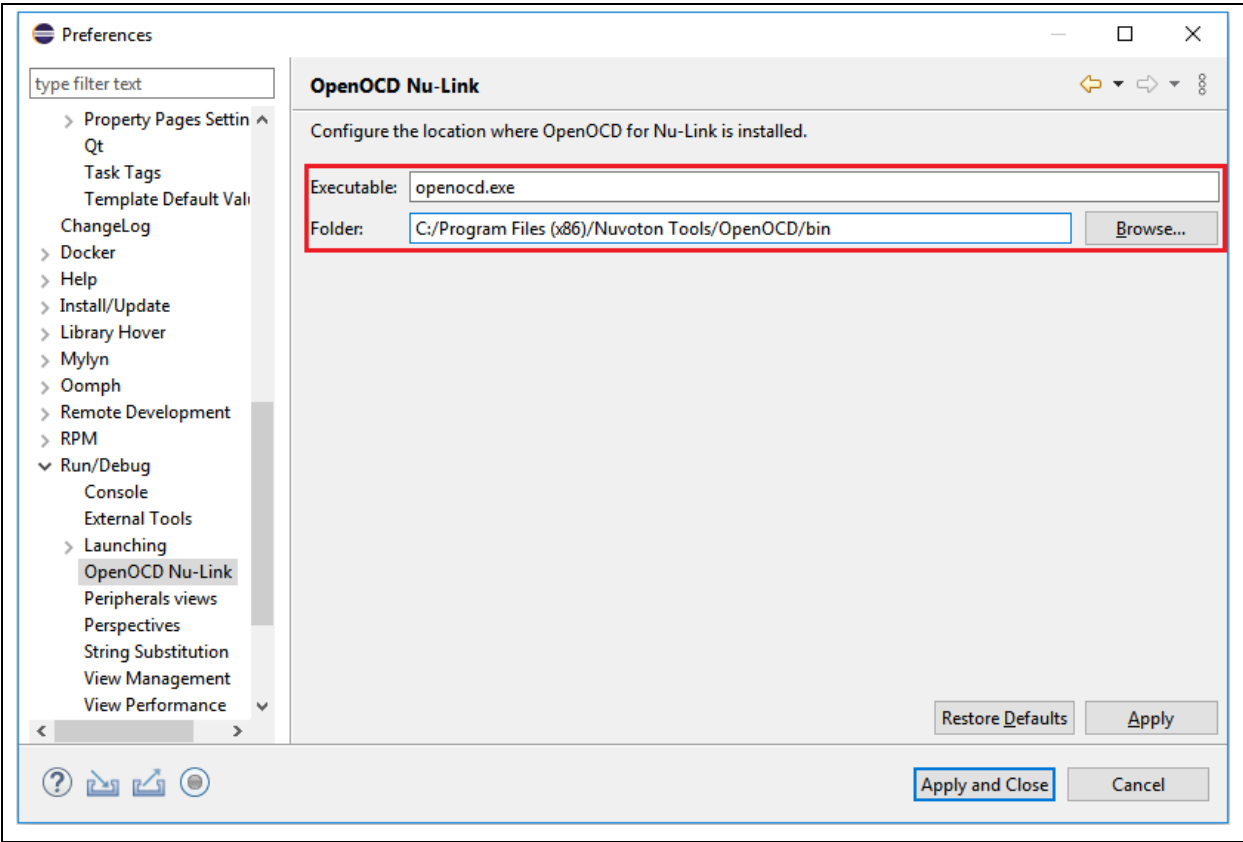


图 3-4 Preferences 设置 OpenOCD Nu-Link 的对话框

3.2 执行Eclipse

为执行NuEclipse，双击eclipse.exe。请注意执行档和相关的文件夹(例如OpenOCD文件夹)应该待在同一个目录底下。否则，软件将不能正常运作。

Name	Date modified	Type	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
eclipsesec.exe	2020/9/10 上午 11:05	Application	129 KB
notice.html	2020/9/7 下午 09:35	HTML Document	10 KB

图 3-5 Eclipse.exe 和相关文件夹

4 工程开发教学

4.1 选择Workspace

当Eclipse启动时，我们必须选择一个Workspace，其包含相关工程并构成一个应用程序。此外，Eclipse和工程们的一些配置设置也存储在这里。对于不同的计算机，配置设置可能会有所不同。我们应该创建自己的Workspace，而不是复制其他用户的Workspace。一次只能有一个工作空间处于活动状态。若要切换Workspace，点选File->Switch Workspace.

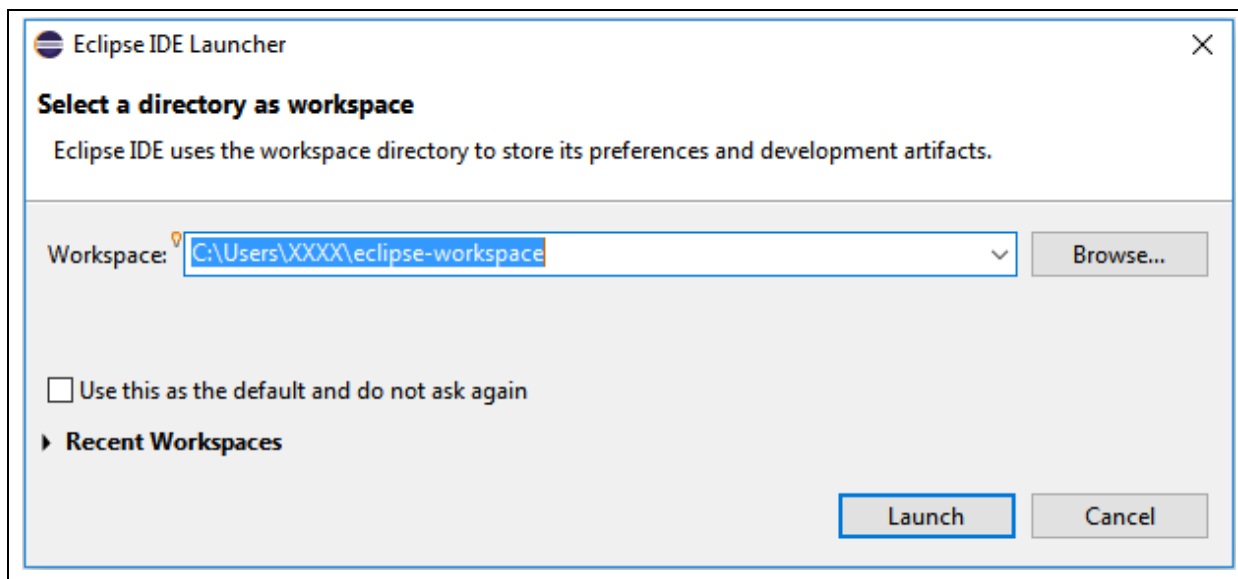


图 4-1 选择 Workspace

4.2 新工程向导

作为一个新手，快速创建C/C++工程的方式为使用新工程向导。以创建C工程为例，点选**File > New > C Project**。新工程向导的对话框会出现。在对话框内，点选**Hello World Nuvoton Cortex-M C Project**，然后输入工程名称，最后点选**Next >**按钮以继续下一步。

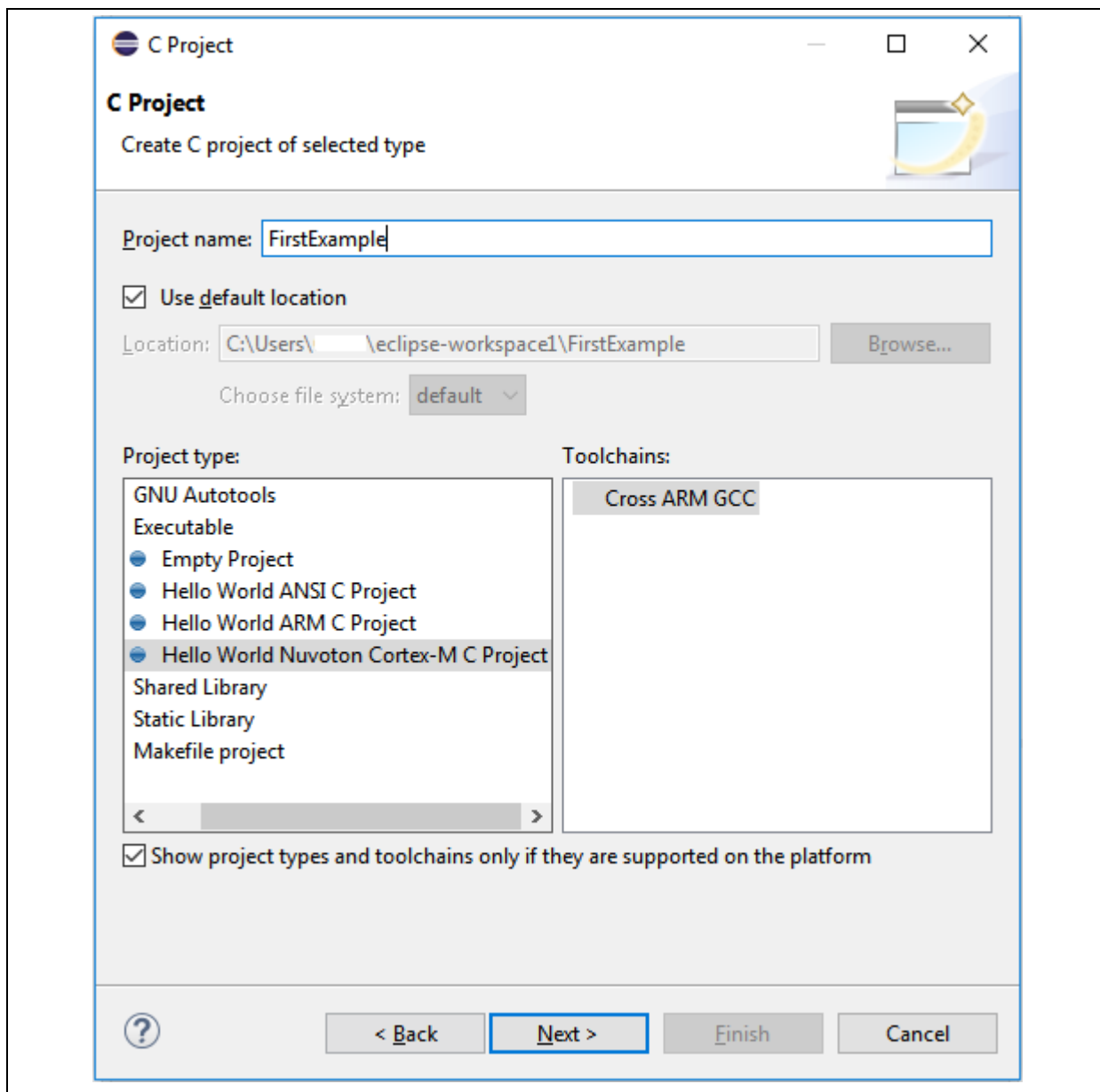


图 4-2 新工程向导

依据实际的目标芯片，我们选择相对应的芯片系列。对于某些芯片，例如M2351_NonSecure，我们需要输入额外的库路径。否则，建置过程可能会失败。此外，输入实际闪存和内存的值。否则，默认值将会被采用。当所有设定都完成后，点选Next > 按键直到点选Finish按键为止。

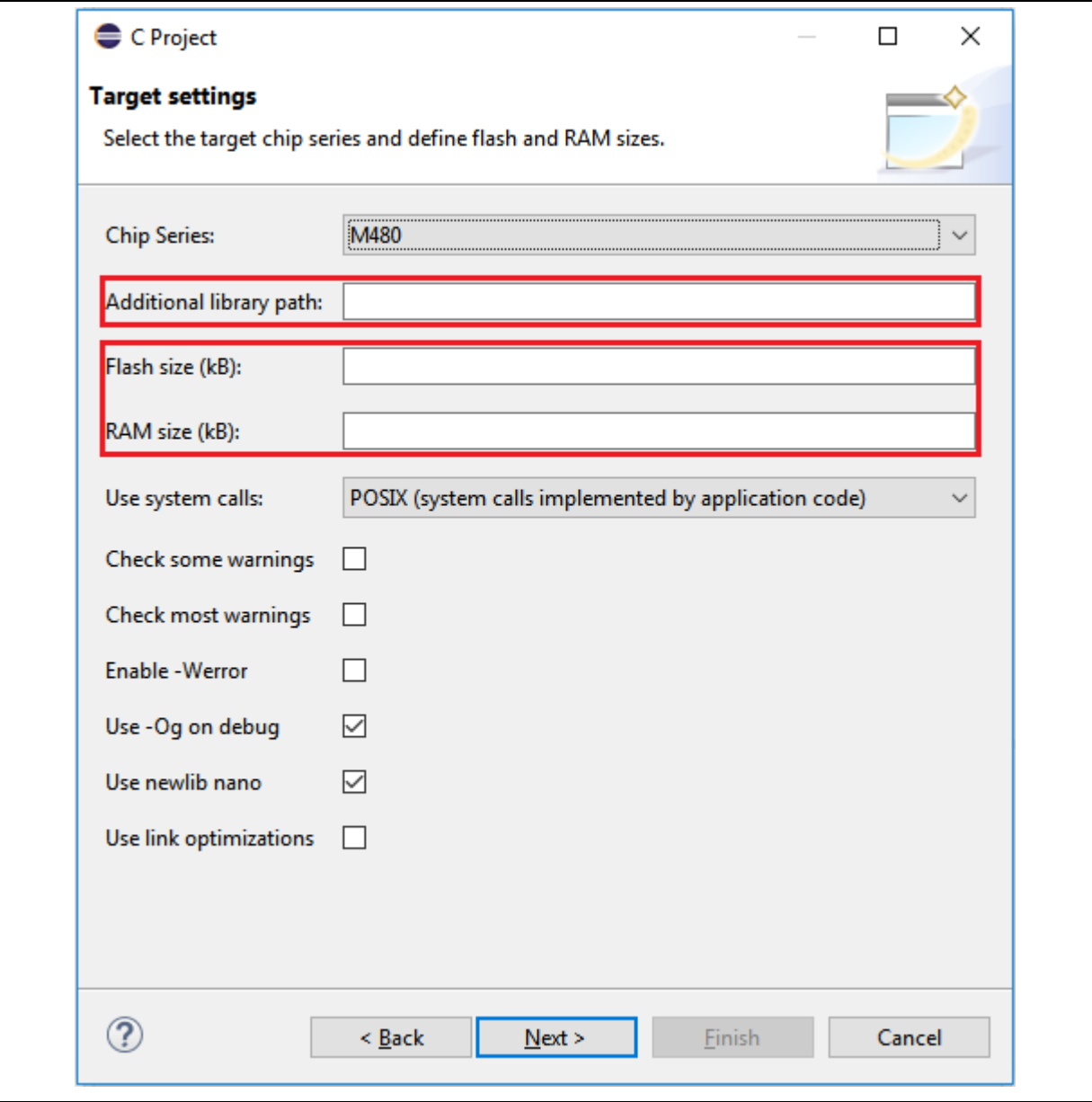


图 4-3 目标处理器的设定

4.3 汇入现有工程

当BSP可用时，我们可以使用以下步骤将它们汇入到Workspace中：

1. 从menu中，选择File > Import。 汇入向导出现。
2. 选择General> Existing Project into Workspace并点选Next。
3. 选择Select root directory或Select archive file，然后点选Browse 找到包含工程的目录或文件。 在Nuvoton BSP中，Eclipse工程放在GCC文件夹中。
4. 在Projects 下选择您想要汇入的工程，然后点选Finish。

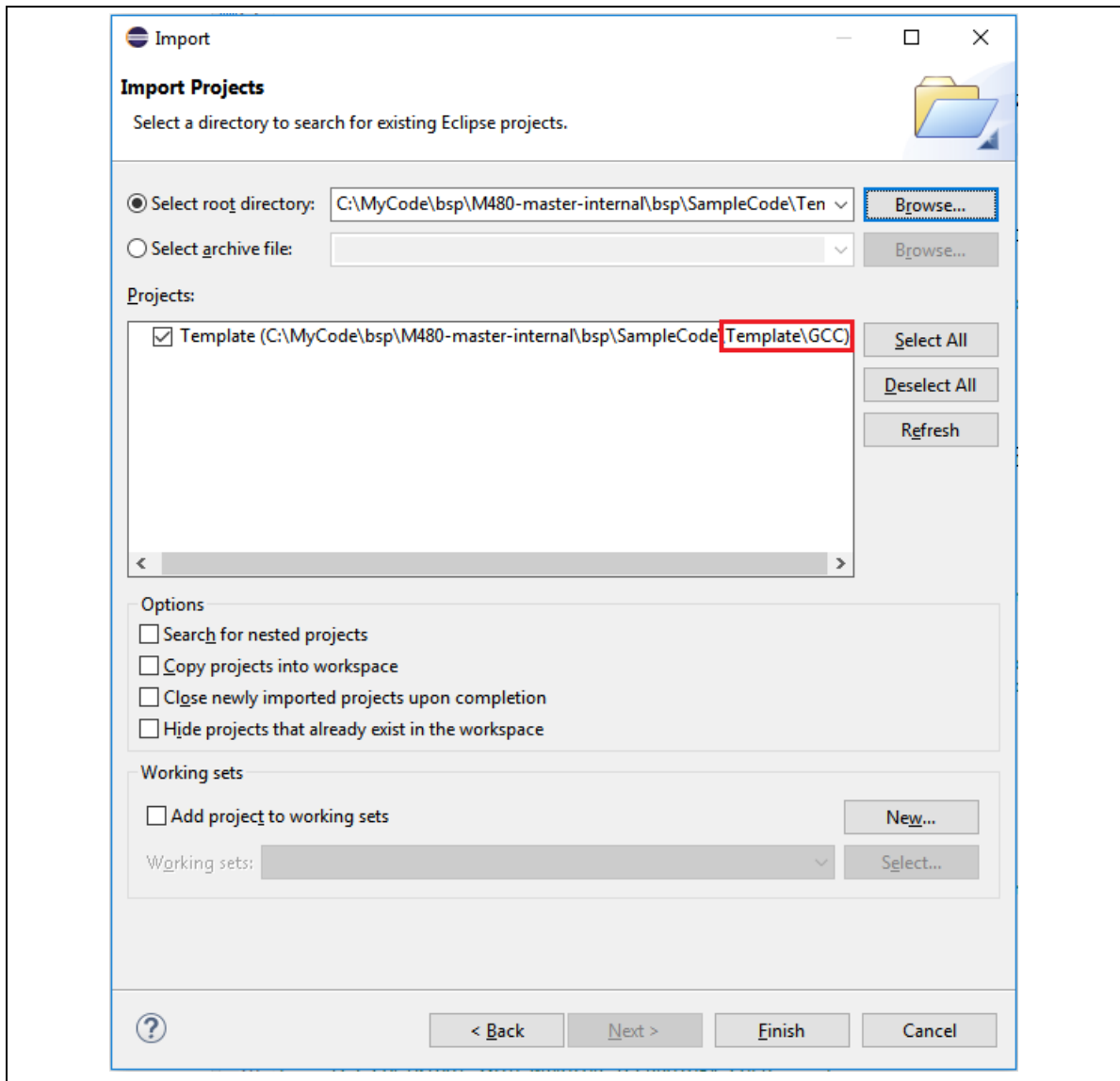


图 4-4 汇入工程

4.4 建置设定

工程创建完成后，我们仍有机会来更改建置设定，只要点选**Project > Properties**即可。当Properties对话框出现，前往**C/C++ Build > Settings**。在那里，我们可以根据现况来更改建置设定。点选**Apply**按键来生效。设定完成后，我们应该能够顺利地建置工程。

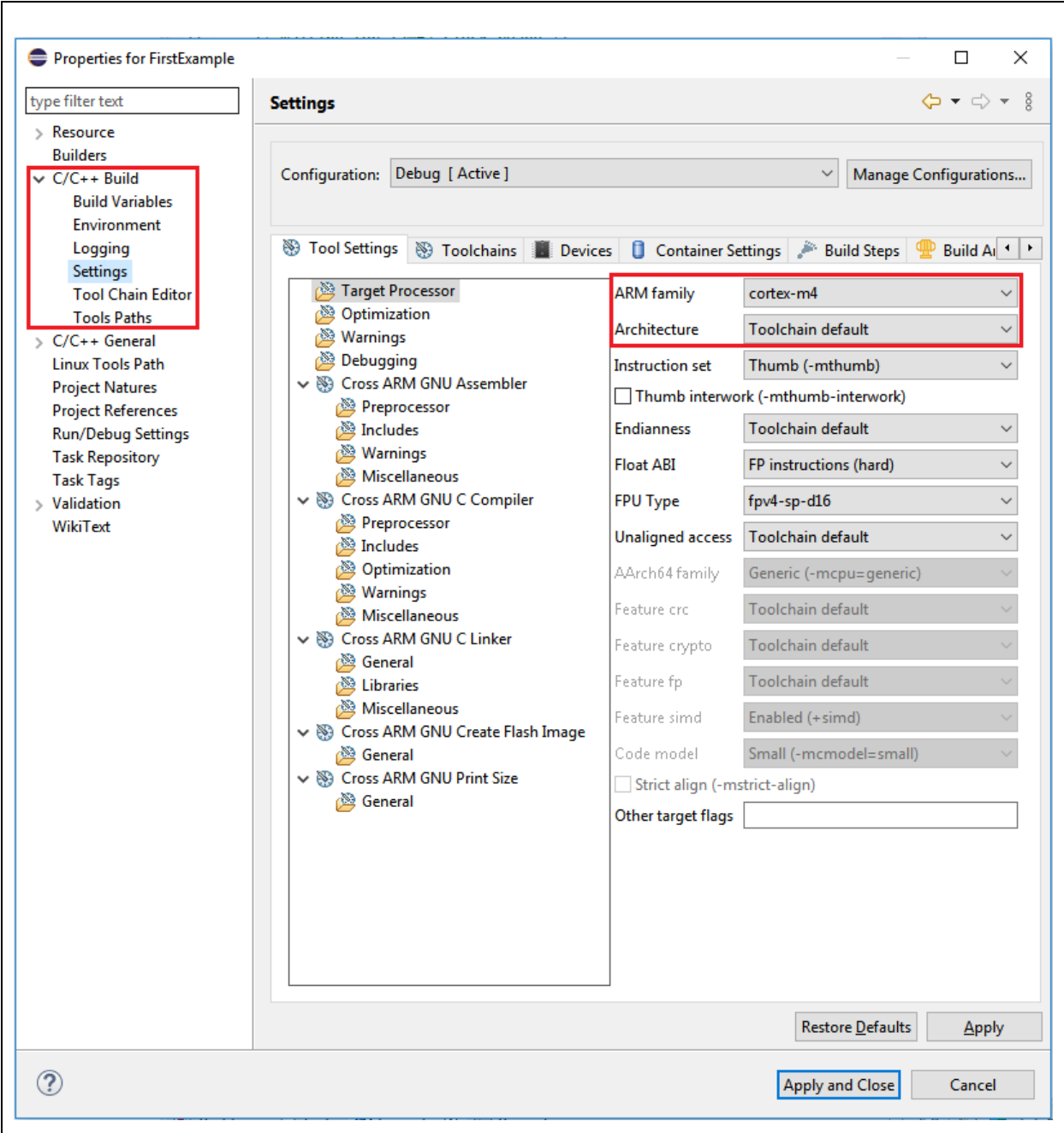


图 4-5 建置设定

4.5 调试配置

在启动应用程序进入调试模式前，我们必须先准备好调试配置，其包含所有关于调试所需的信息。點選 **Run > Debug Configuration...** 来呼叫调试配置对话框。双击 **GDB Nuvoton Nu-Link Debugging** 群组。新唐Nu-Link调试配置会出现在右手边 在Main标签页里，Project名称应与工程名称一致。C/C++ 应用程序应指向建置产生出来的 .elf 应用程序。倘若工程名称或C/C++应用程序不正确，请至工程视图选取想要调试的工程并建置它以产生执行档，然后展开树以确保生成的执行档存在。然后再重复先前的操作一遍。

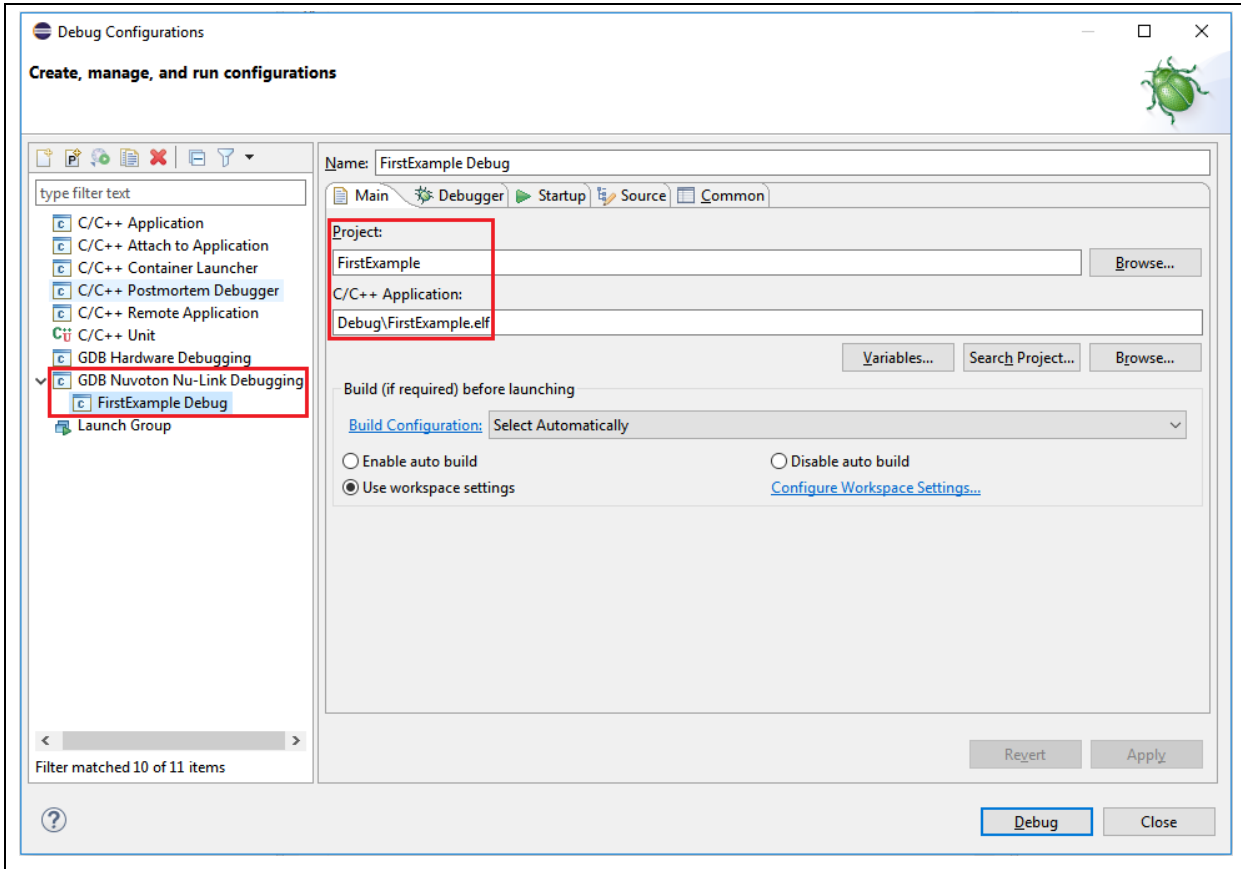


图 4-6 调试配置

4.5.1 Debugger标签页

Debugger 标签页用来提供OpenOCD和GDB客户端设置。OpenOCD需要正确的配置文件才能知道如何和适配器及目标芯片工作。配置文件定义在**Config options**字段。新唐适配器为Nu-Link，其接口配置文件为**nulink.cfg**。新唐有三个不同的ARM家族，即M0、M4和M23。相对应的目标配置文件为**numicroM0.cfg**、**numicroM4.cfg**和**numicroM23.cfg**。就M23二次开发而言，目标配置文件会是**numicroM23_NS.cfg**。就MA35D1开发而言，见4.10章节。

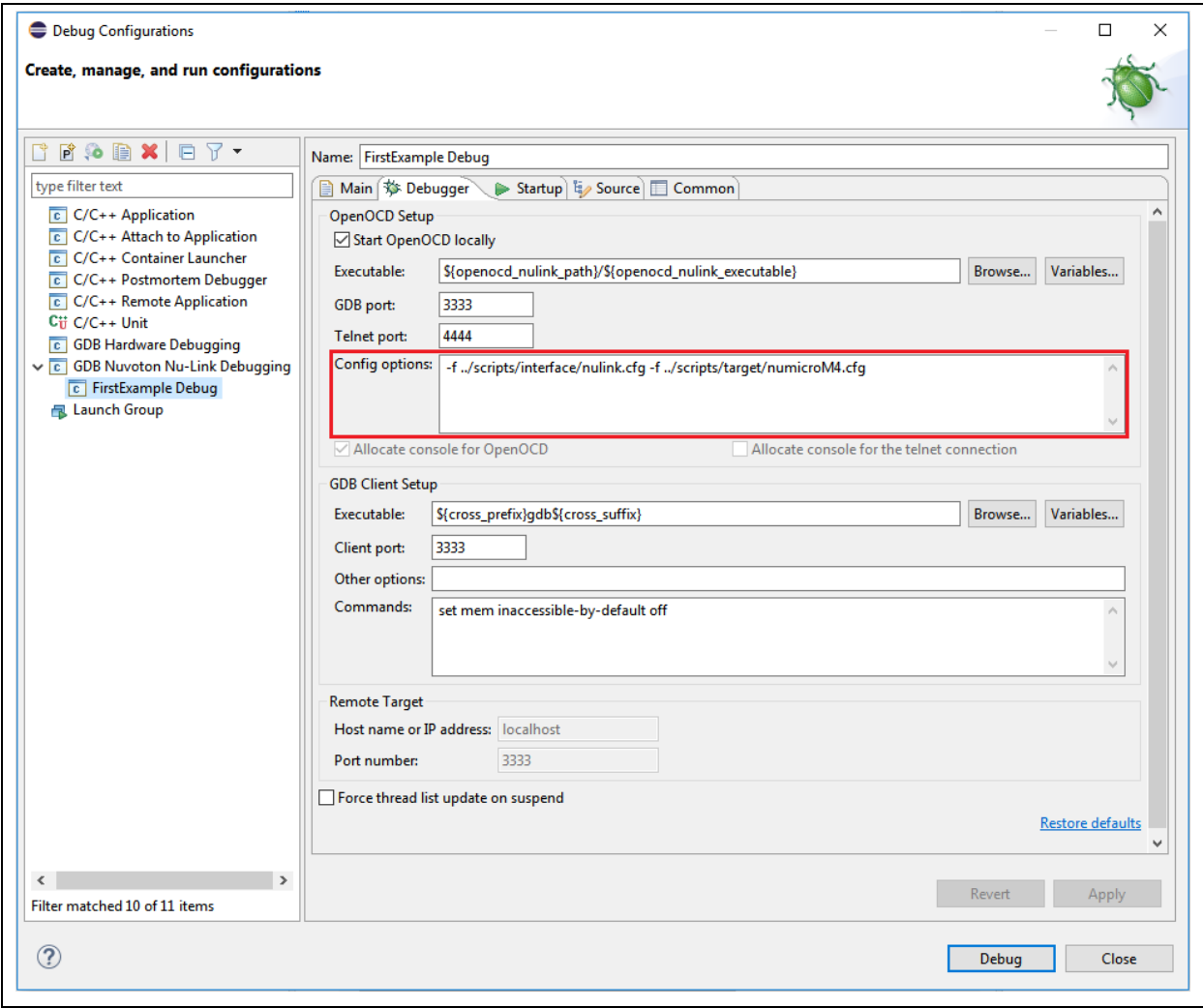


图 4-7 设置 Debugger 标签页

4.5.2 Startup标签页

作为开发的第一步，我们应到Startup标签页选择**对的Chip Series**。选完，相对应的目标配置文件将会自动地写入Debugger 标签页的**Config options**字段。为载入执行档至闪存，我们需勾选**Load executable to flash**复选框。为载入执行档至内存，我们需勾选**Load executable to SRAM**复选框。当所有的设定完成，点选**Apply**按键来生效。为启动应用程序进入调试模式，请点选**Debug**按键。

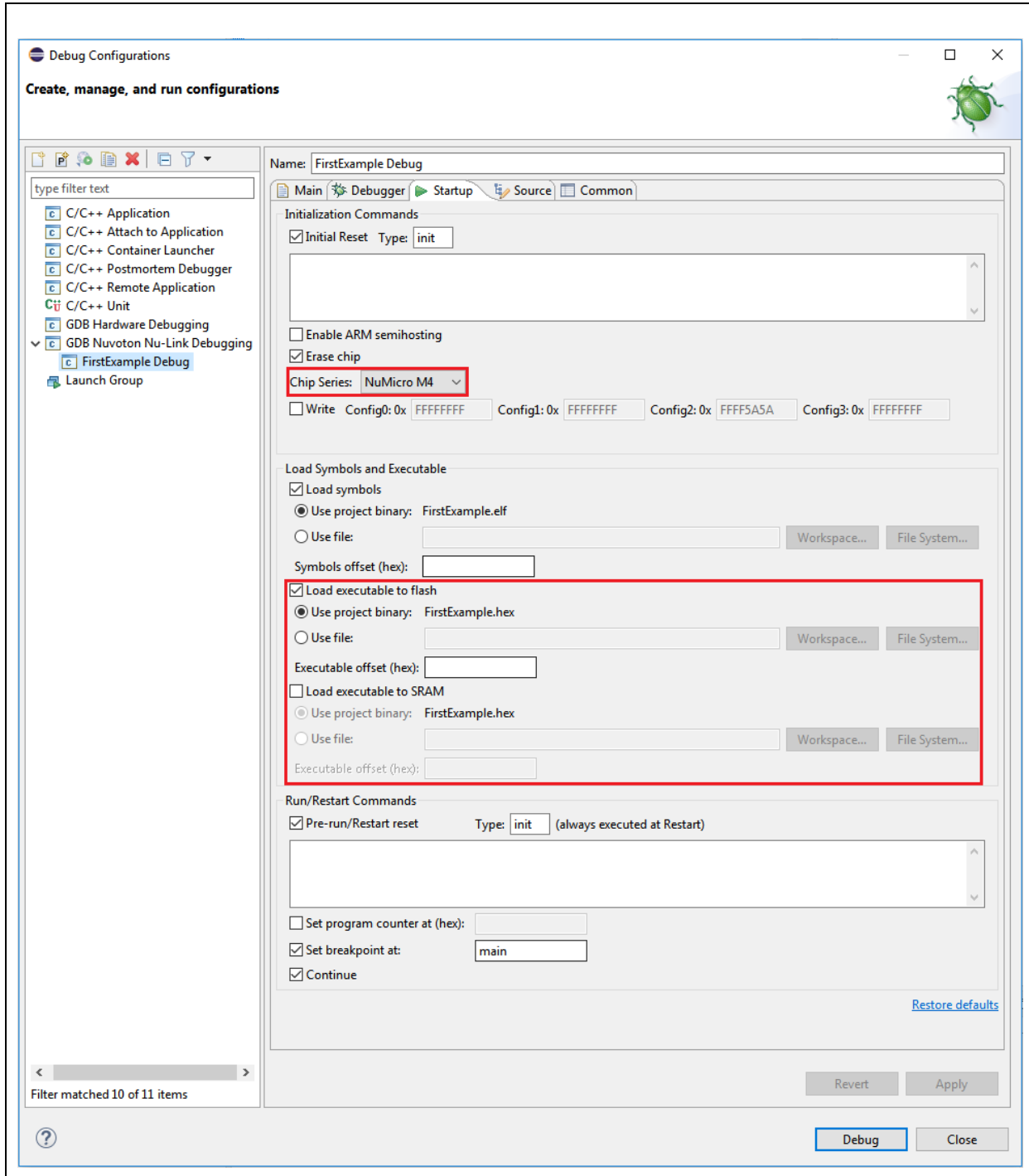


图 4-8 设置 Startup 标签页

4.6 调试视图

Eclipse提供多个调试视图。当中每一个都有特定的调试信息。

4.6.1 寄存器视图

当进入调试模式时，我们可以打开位於调试观点中底部的**寄存器视图**。寄存器视图显示目前栈上的寄存器信息。

Console 1010 0101 Registers Problems Executables Debugger Console Memory		
Name	Value	Description
General Registers		
1010 0101 r0	0	General Purpose and FPU Register Group
1010 0101 r1	0	
1010 0101 r2	536871108	
1010 0101 r3	537	
1010 0101 r4	0	
1010 0101 r5	0	
1010 0101 r6	0	
1010 0101 r7	0	
1010 0101 r8	0	
1010 0101 r9	0	
1010 0101 r10	536936448	
1010 0101 r11	0	
1010 0101 r12	0	
1010 0101 sp	0x2001fff8	
1010 0101 lr	643	
1010 0101 pc	0x32e <main+2>	
1010 0101 xPSR	1627389952	

图 4-9 寄存器视图

4.6.2 内存视图

调试观点中的**内存视图**可用作监视和修改进程内存。进程内存被表示成一系列的**内存监测器**。每一个监测器代表一小段内存，由它的**初始地址**来描述。为打开视图，点选调试观点下半部的内存标签页。

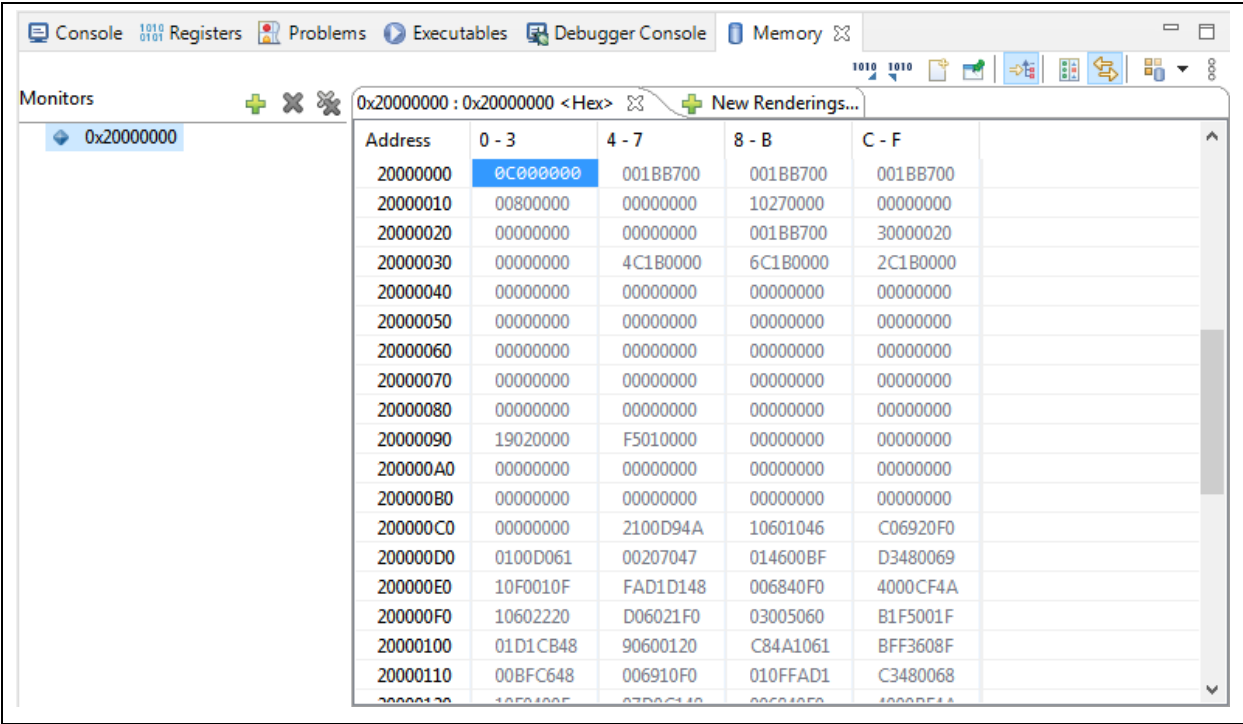


图 4-10 内存视图

4.6.3 反汇编视图

反汇编视图将载入程序显示为汇编指令并混合源代码作为比对。我们可在反汇编视图上作下列操作：

1. 设置断点在任何汇编指令的起点。
2. 启用或关闭断点。
3. 单步汇编指令。
4. 跳至特定的指令。

为打开视图，我们需在上方工具栏點選 **Instruction Stepping Mode** 按键，如下所示。



图 4-11 點選 Instruction Stepping Mode 按键

然后反汇编视图会出现在右边窗口。

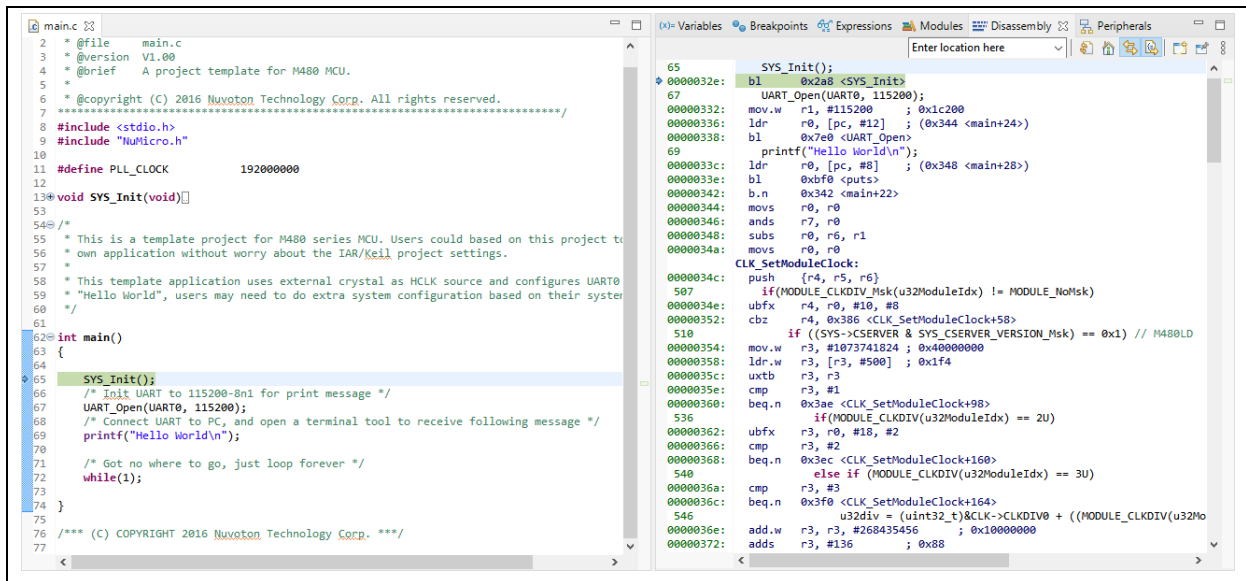


图 4-12 反汇编视图

4.6.4 外围寄存器视图

为显示外围寄存器视图，我们需使用Packs机制。该机制帮助用户从Keil数据库下载特殊函数寄存器(SFR)文件。首先，我们从Open Perspective对话框中开启Packs观点。

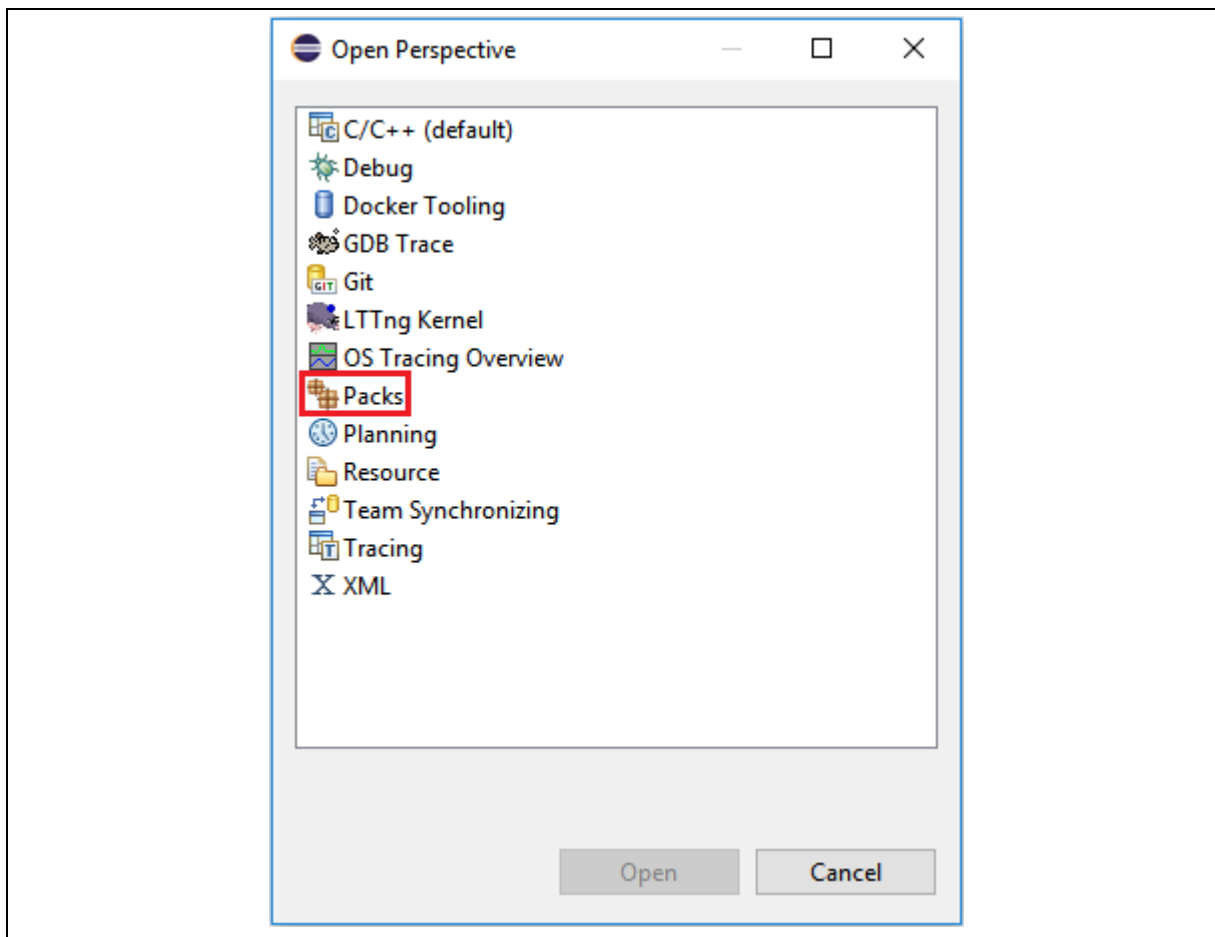


图 4-13 开启 Packs 观点

我们第一次看到Packs观点时，其内容是由NuEclipse安装软件所提供。倘若默认内容不见了，请切换到一个全新的Workspace然后再试一次。若想获得最新版本，点选右上角的Update the packages definitions from all repositories按键。之后，Eclipse会从在线数据库开始下载所有的SFR文件。

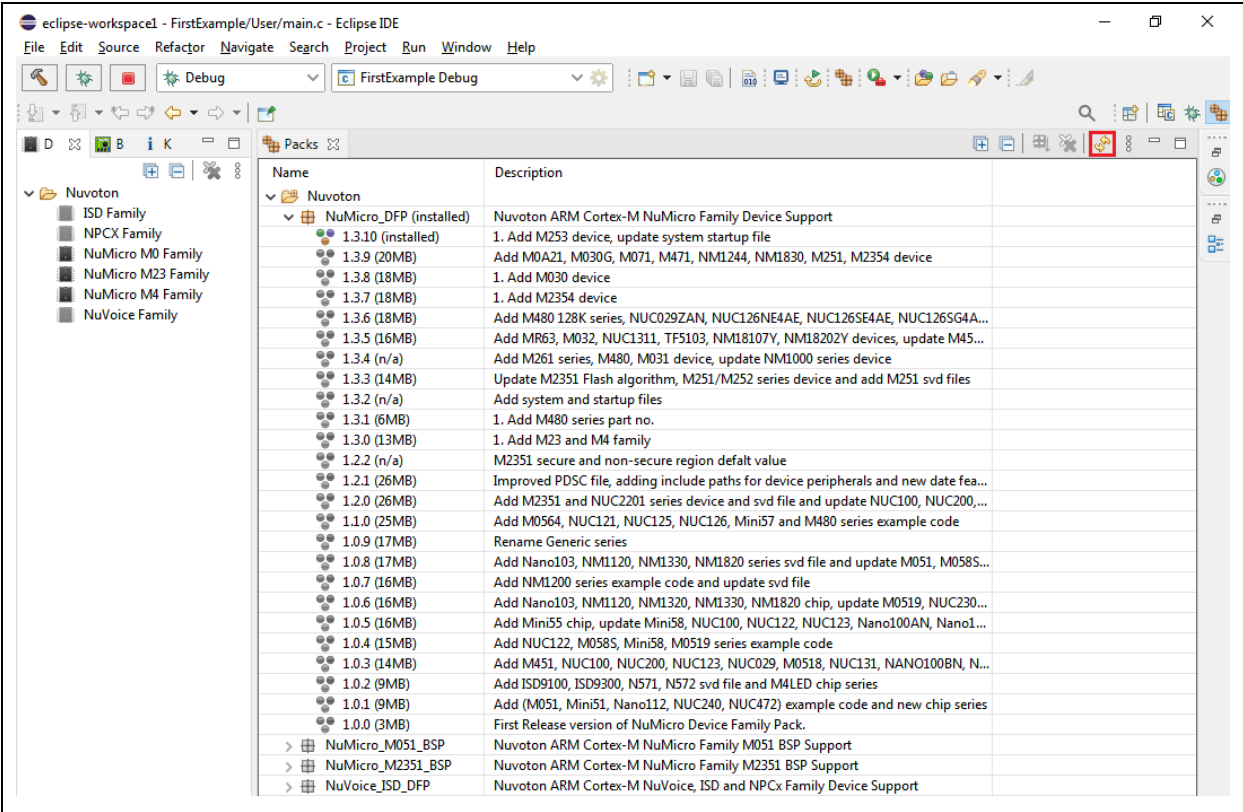


图 4-14 如何下载套件

数据库所在位置定义在Window > Preferences > C/C++ > Packages > Repositories。初始库是来自Keil’s CMSIS Pack。

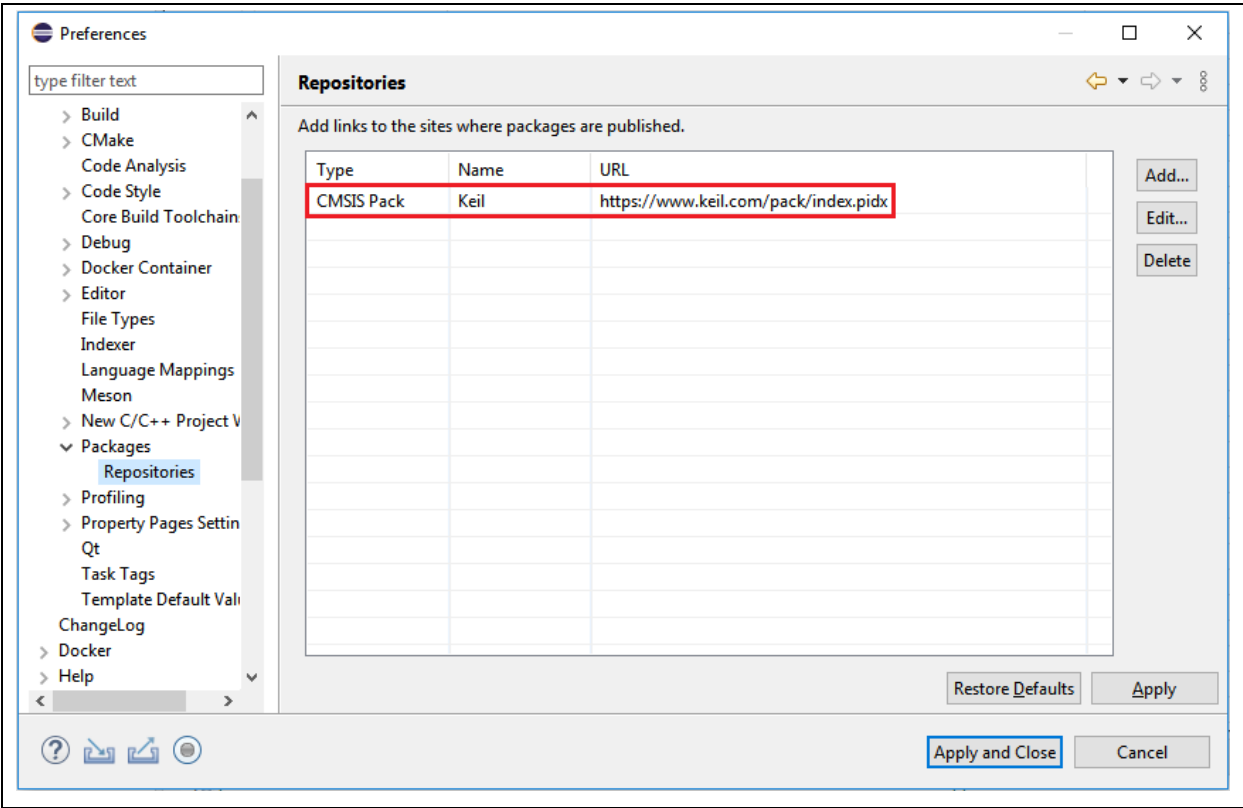


图 4-15 数据库位置

当下载完成时，我们可以找到新唐的SFR文件并在Eclipse安装它们，如果需要的话。

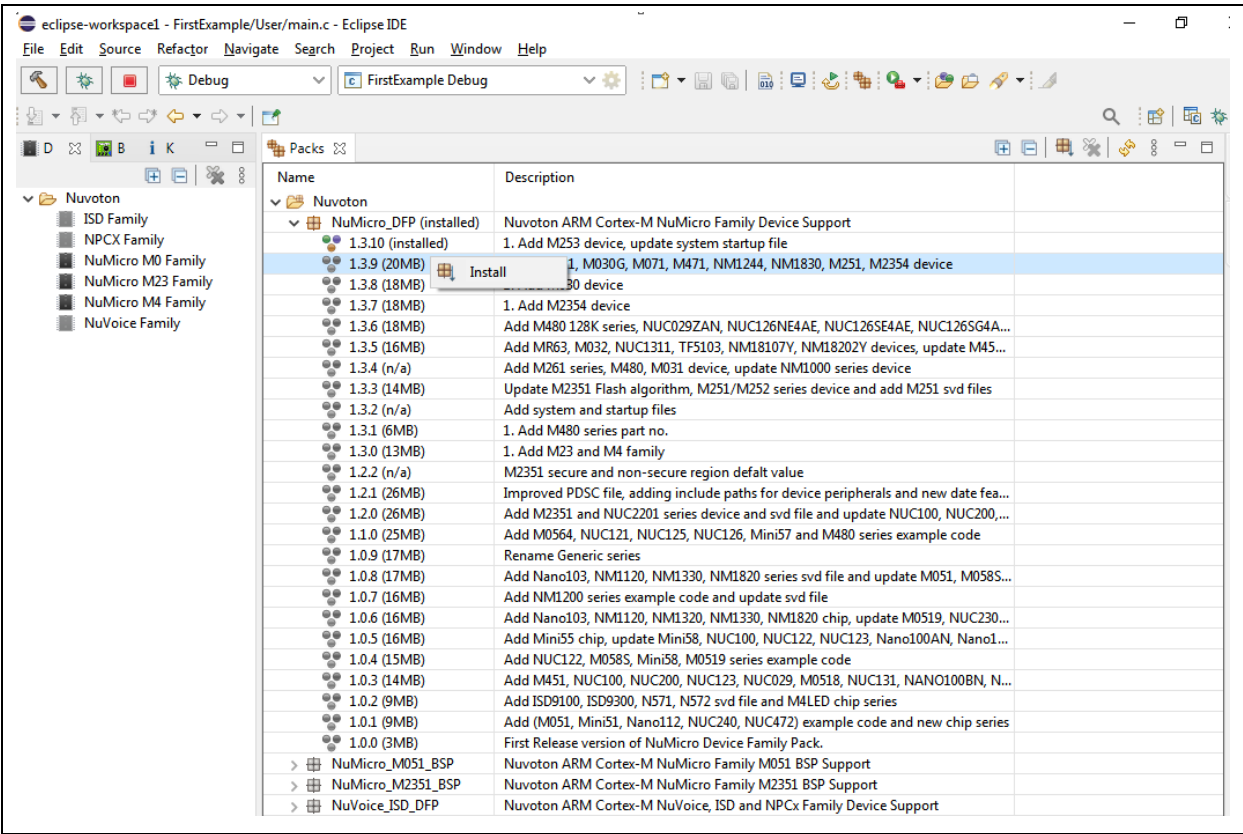


图 4-16 安装新唐 SFR 文件

为使用特定的SFR文件，打开工程的属性对话框，前往C/C++ Build > Settings。从那里，我们应该选择特定的设备以符合现实的芯片。在这个范例里，设备是 M487JIDAE。点选Apply按钮来生效。

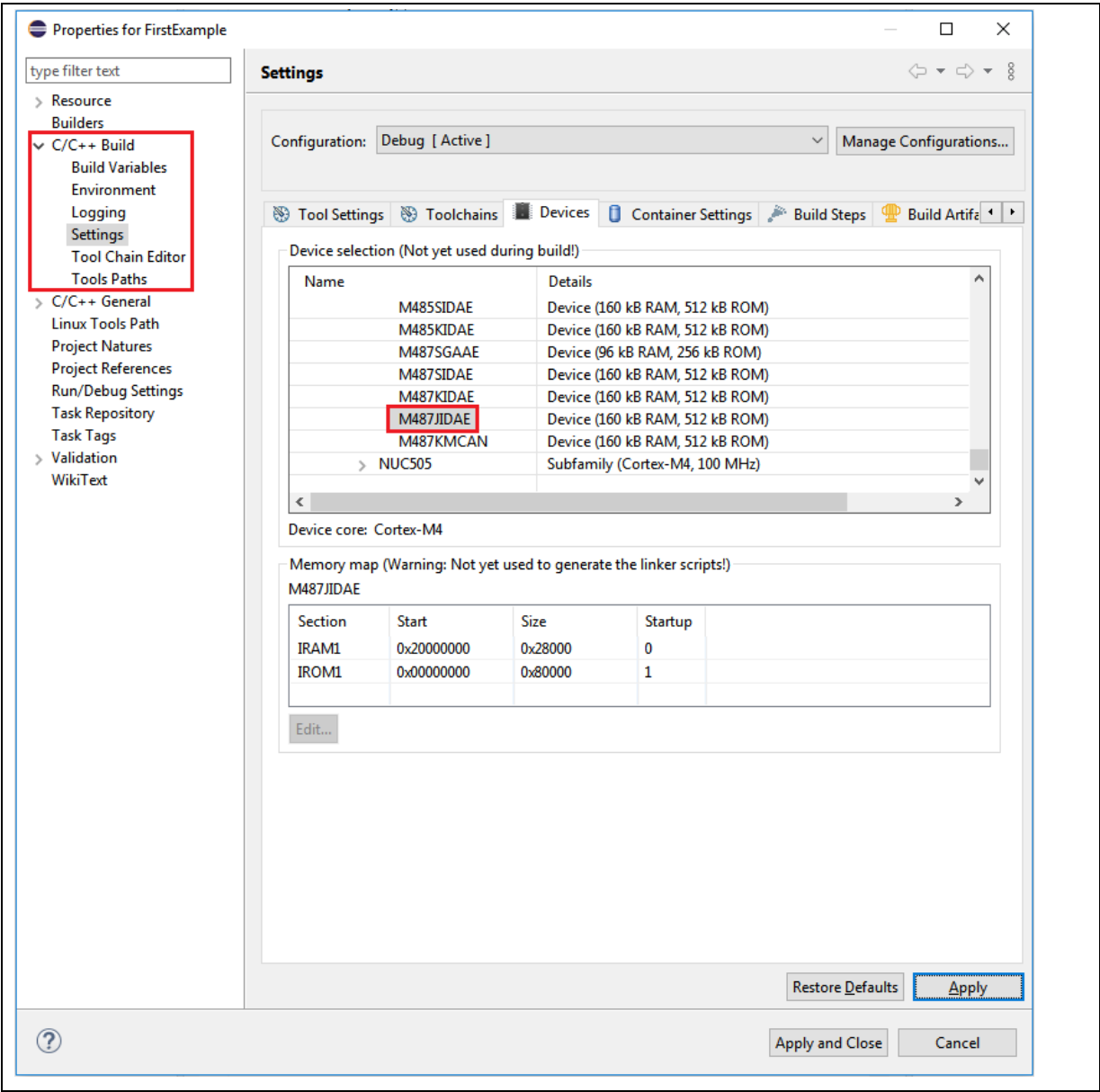


图 4-17 选择设备

最后结果，当我们在调试时可以监看外围寄存器。

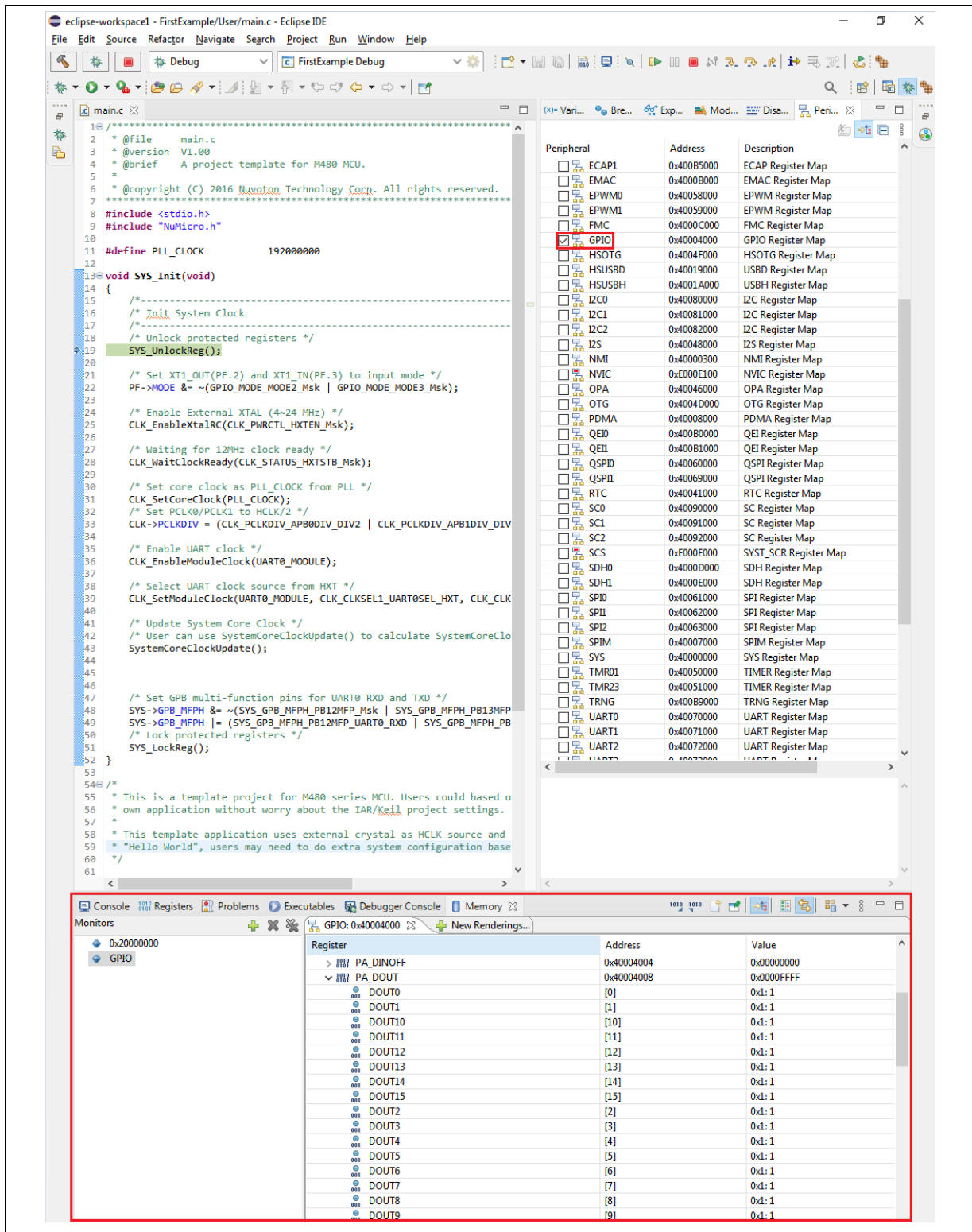


图 4-18 外围寄存器视图

4.7 检测点

为在Eclipse新增检测点，我们需作下面的步骤。

1. 在Outline视图中，选取一个**全局变量**，例如:g_seconds。
2. 鼠标右键单击它并选取Toggle Watchpoint。

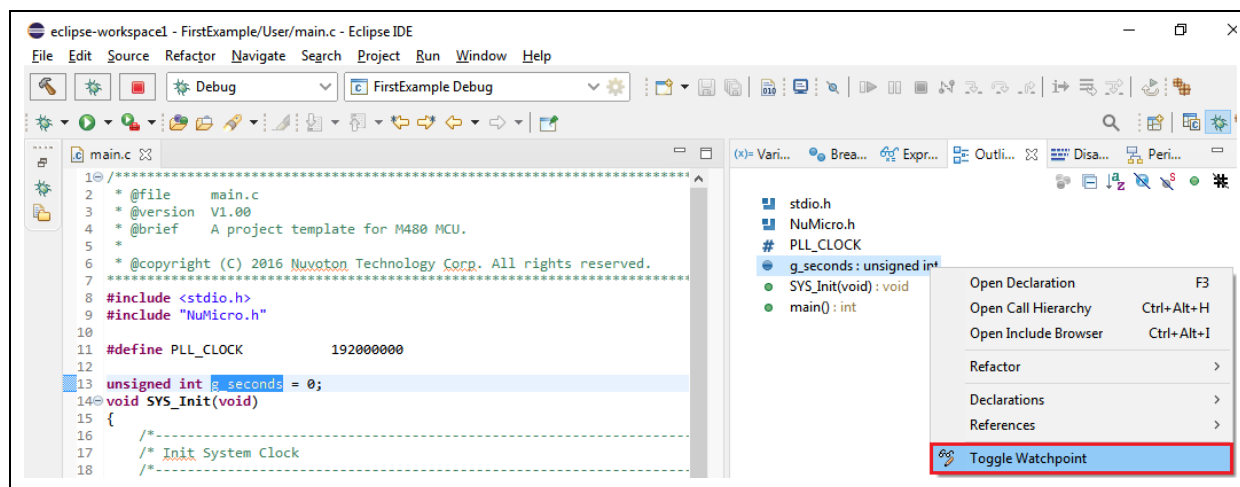


图 4-19 切换检测点

3. 设置检测点。若要检测表达式被读取时让程序停止运行的话，选取Read复选框。若要检测表达式被写入时让程序停止运行的话，选取Write复选框。

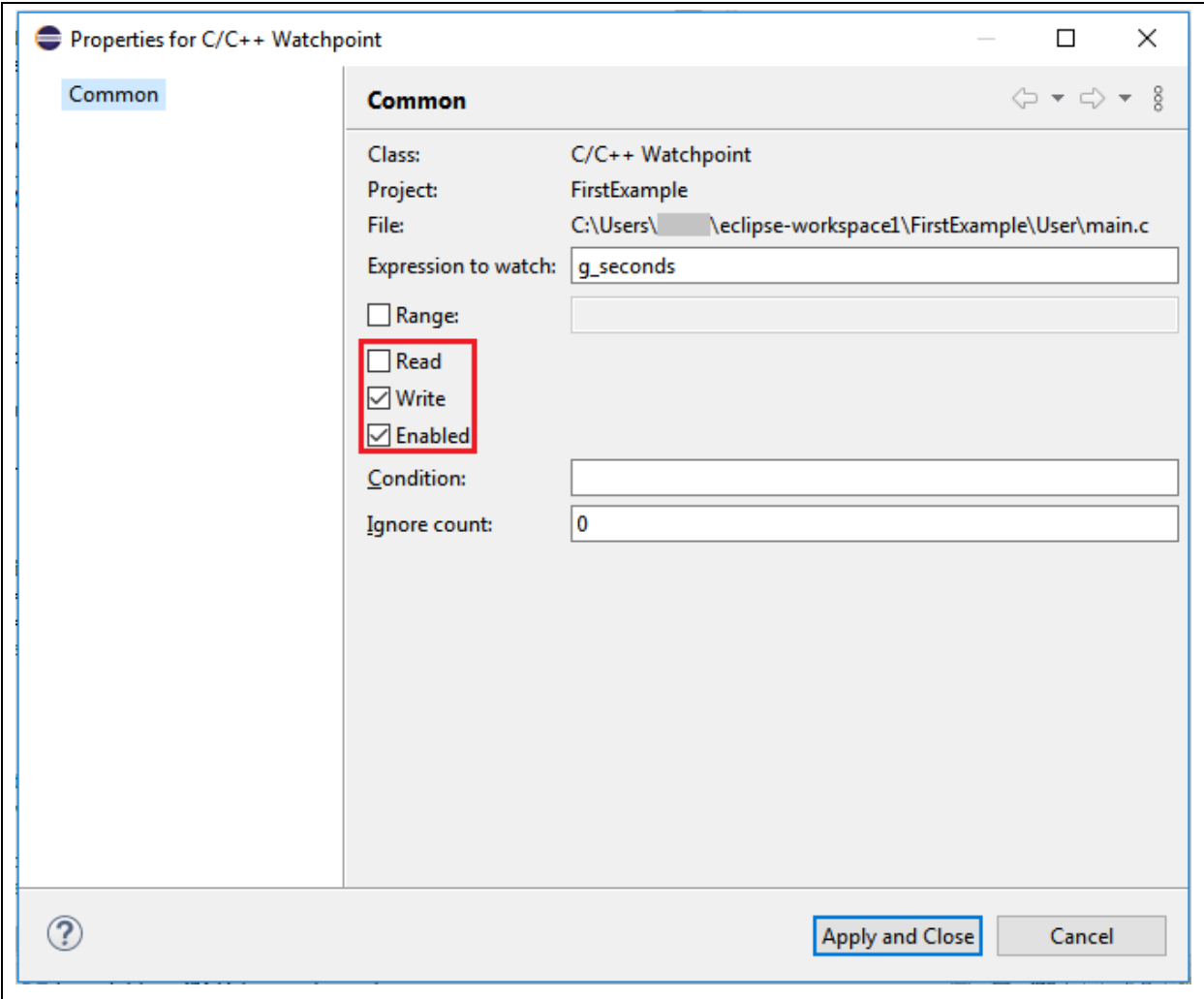


图 4-20 C/C++ 检测点属性对话框

当检测点被新增时，它会出现断点视图里面。

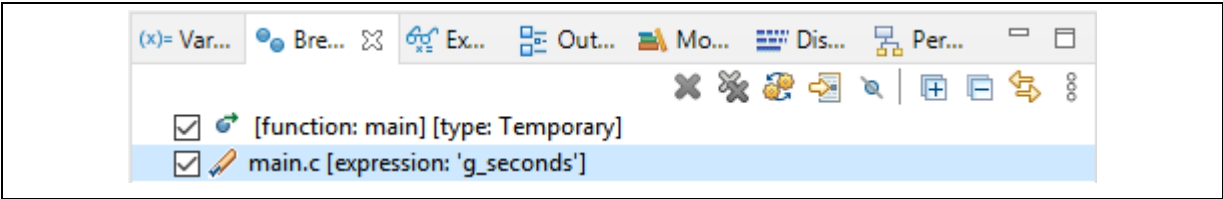


图 4-21 在断点视图新增的检测点

4.8 在RAM中调试

为在RAM上调试，有几个步骤需要完成：

1. 修改ld脚本。
2. 设定PC值至特定RAM地址。
3. 设定SP值至特定RAM地址。
4. 下载二进制文件到RAM。

ld脚本负责告诉链接器编译完成的执行文件如何配置。举例来说，一个内存配置如下：

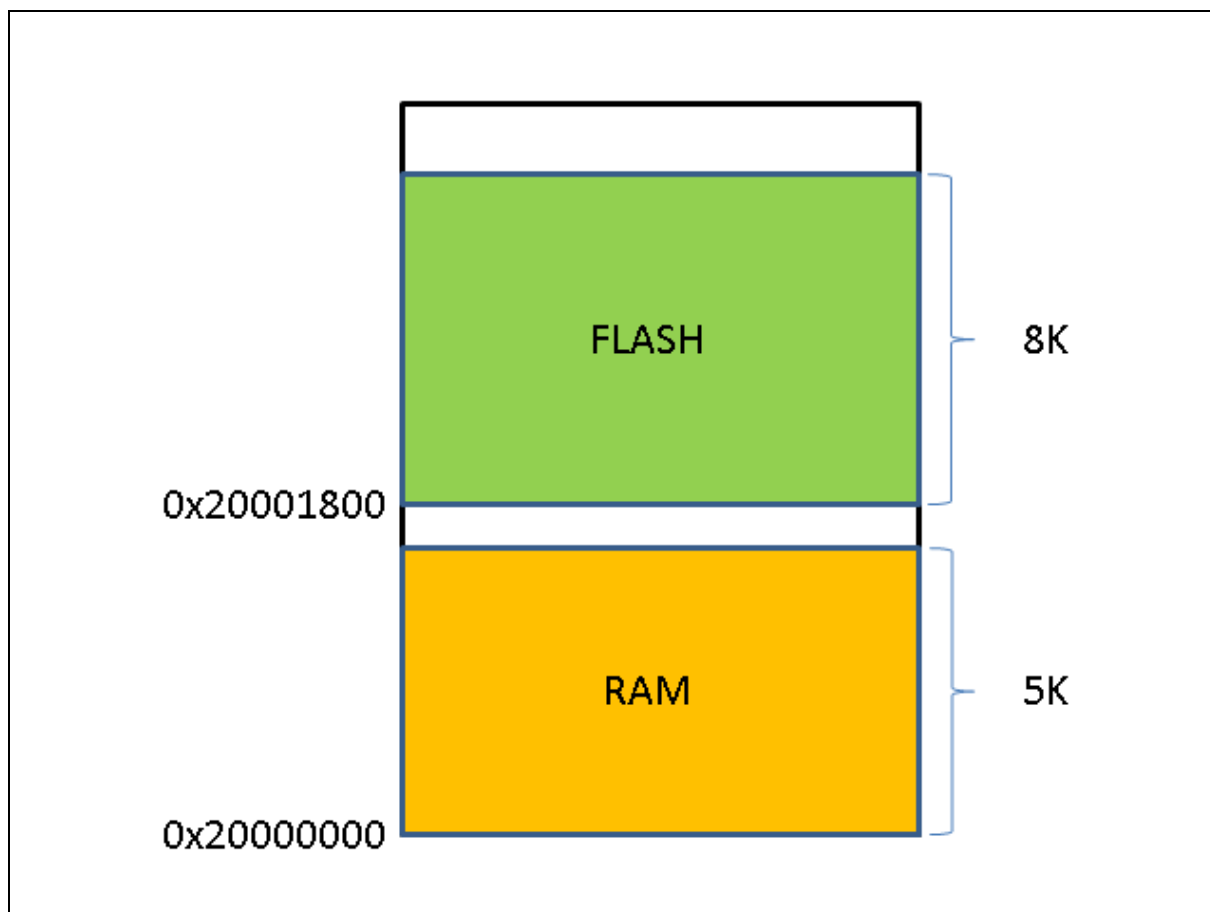


图 4-22 内存配置

修改后的ld脚本应该符合内存配置的设计。

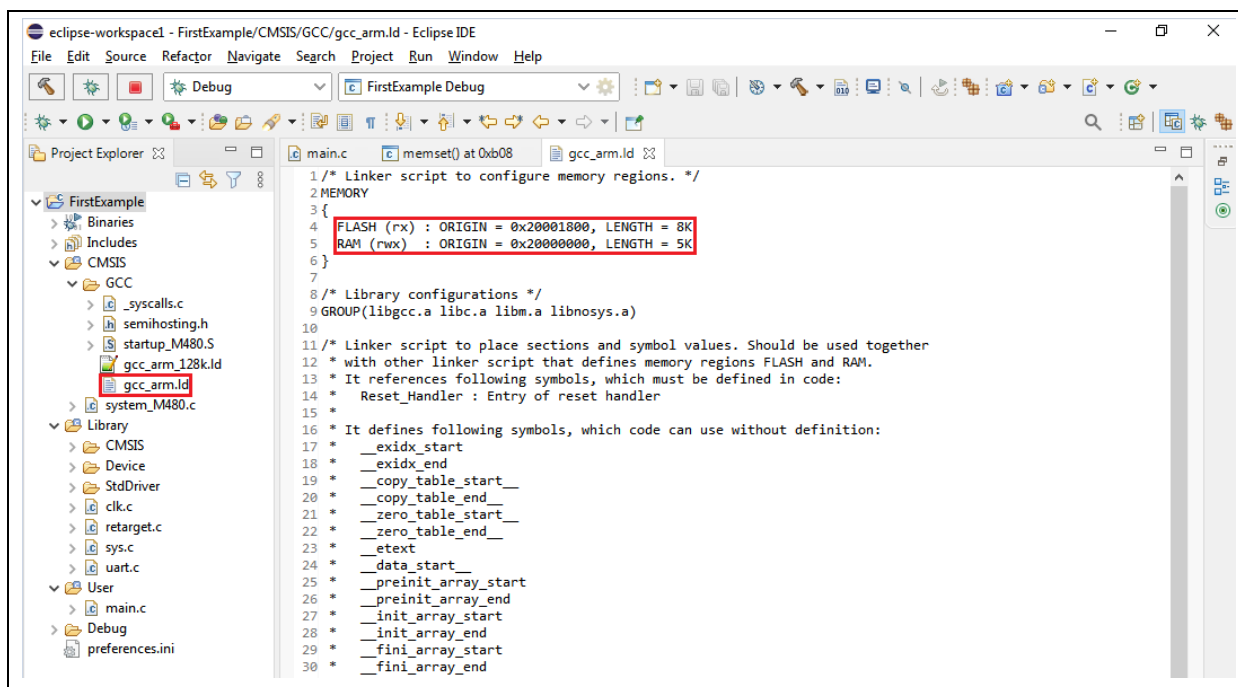


图 4-23 修改 ld 脚本

为设定PC和SP值至特定地址，我们需要输入它们至调试配置，如下。基於前面的内存配置，PC和SP的地址应分别为Reset_Handler和0x20001400。此外，设定Vector Table Offset Register (0xE000ED08)为0x20000000并且取消Pre-run/Restart reset按钮。为下载二进制文件到RAM，勾选Load executable to SRAM按钮和取消Load executable to flash按钮。点选Debug按钮来开始调试阶段。

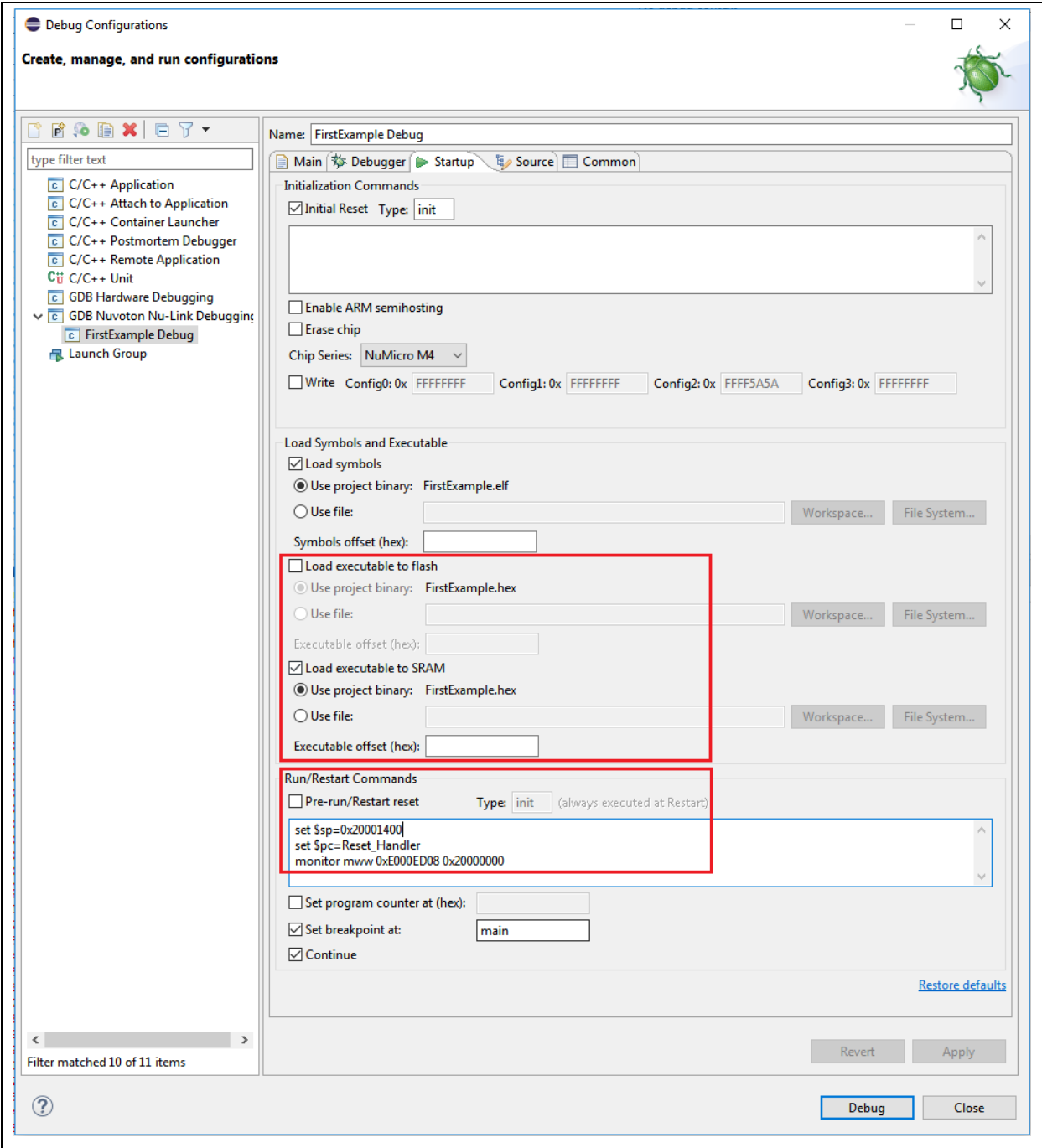


图 4-24 设定调试配置

当程序停在主函数时，我们打开内存视图。从那可以确认到二进制文件已成功地载入至RAM。第一个字代表SP地址，接下来的字代表处理函数的地址。

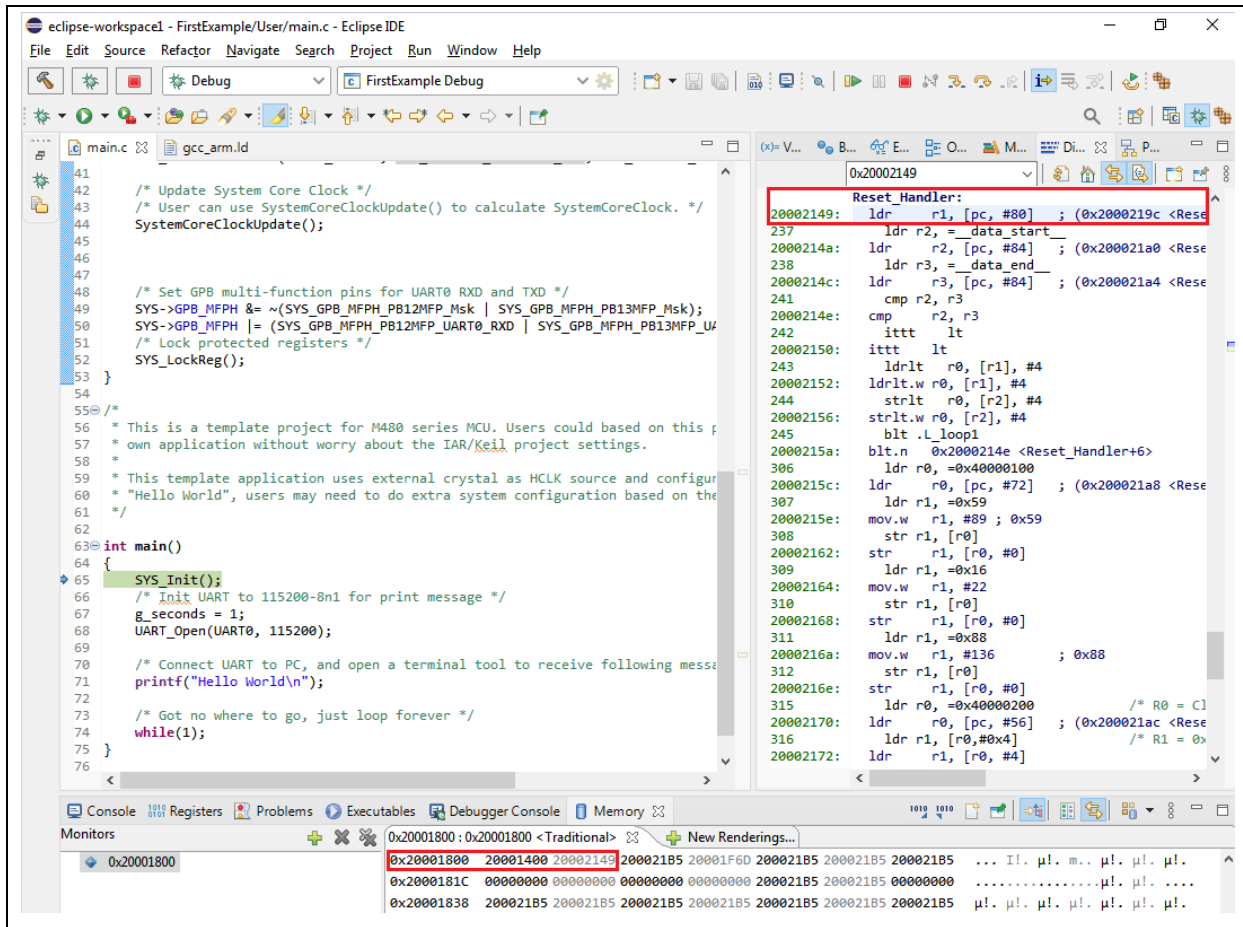


图 4-25 在 RAM 中调试

4.9 调试执行档

如果用户具有使用**调试符号**生成的执行档，但可能没有用于生成该执行档的相关环境，则他仍然可以通过以下步骤来调试该执行档：

1. 汇入执行档进行调试（请参阅图 4-26）。
2. 单击Select executable之后的Browse，然后选择一个执行档（请参阅图 4-27）。
3. 选择GDB Nuvoton Nu-Link Debugging作为启动配置（请参阅图 4-28）。
4. 在调试配置中指定GDB执行档（请参阅图 4-29）。
5. 在调试配置中选择欲载入的ELF文件（请参阅图 4-30）。
6. 添加相对于源码文件夹的源码查找路径（请参阅图 4-31）。
7. 点选Debug按钮。

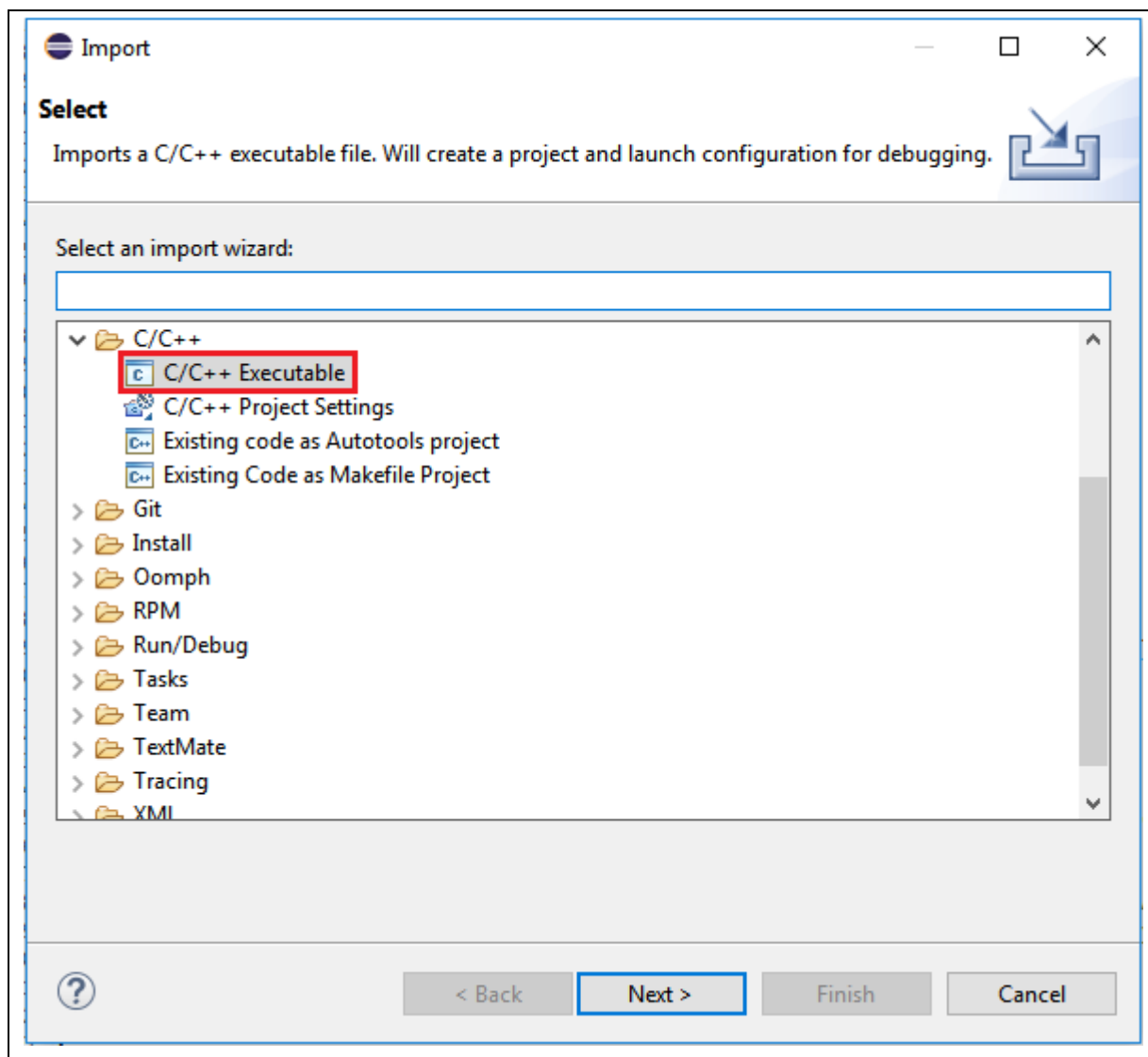


图 4-26 汇入执行档进行调试

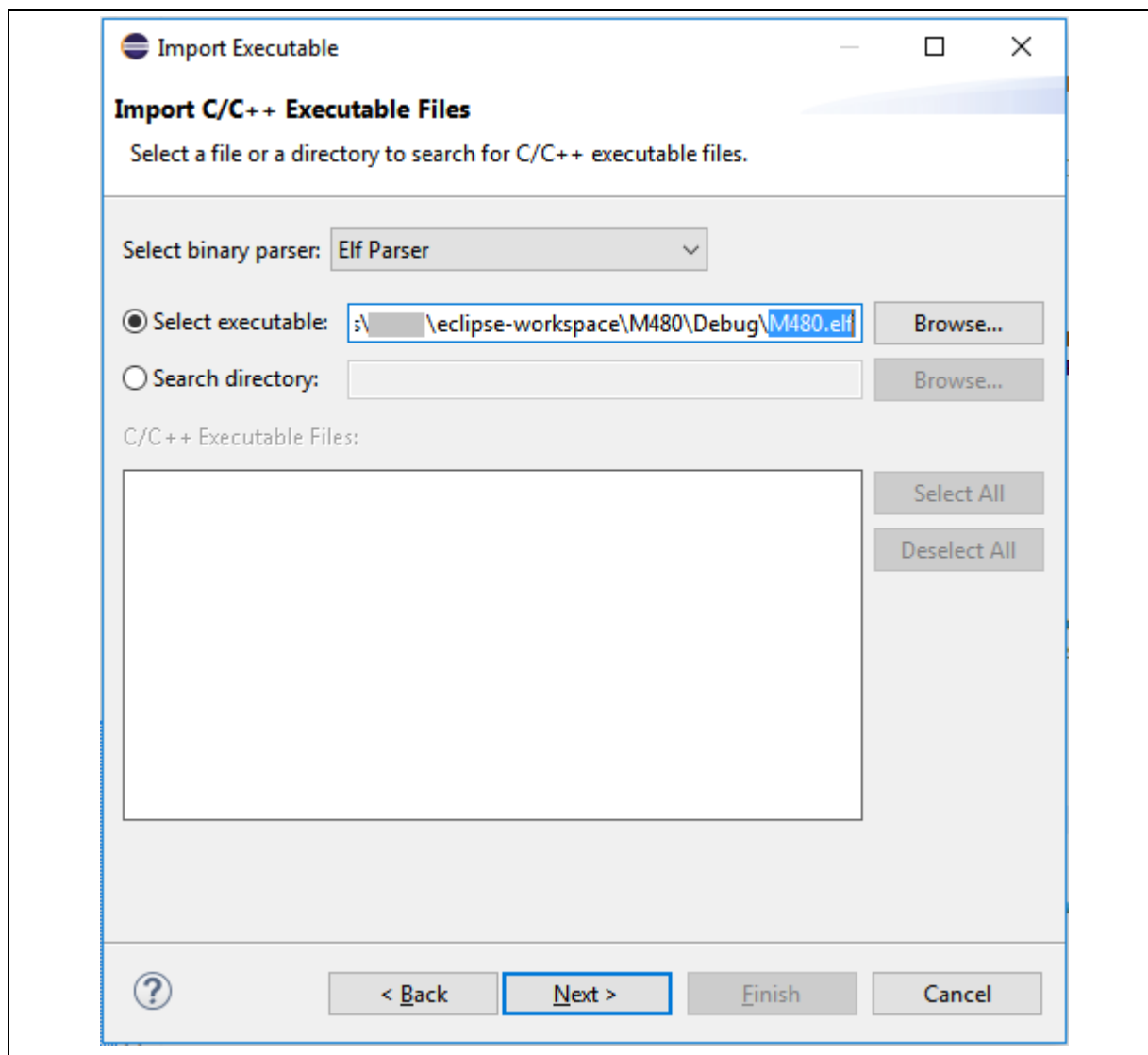


图 4-27 选择执行档

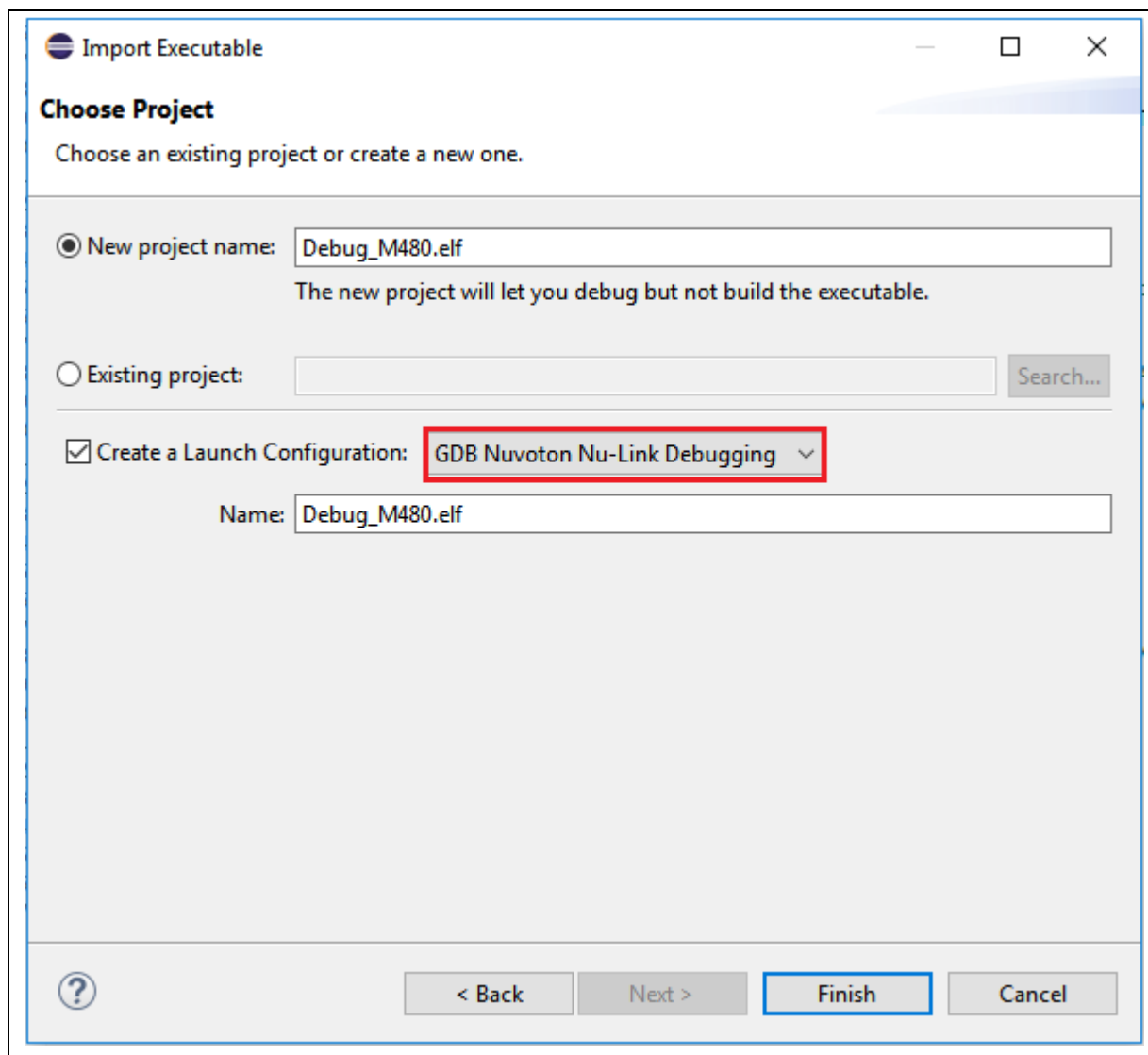


图 4-28 选择 GDB Nuvoton Nu-Link Debugging 作为启动配置

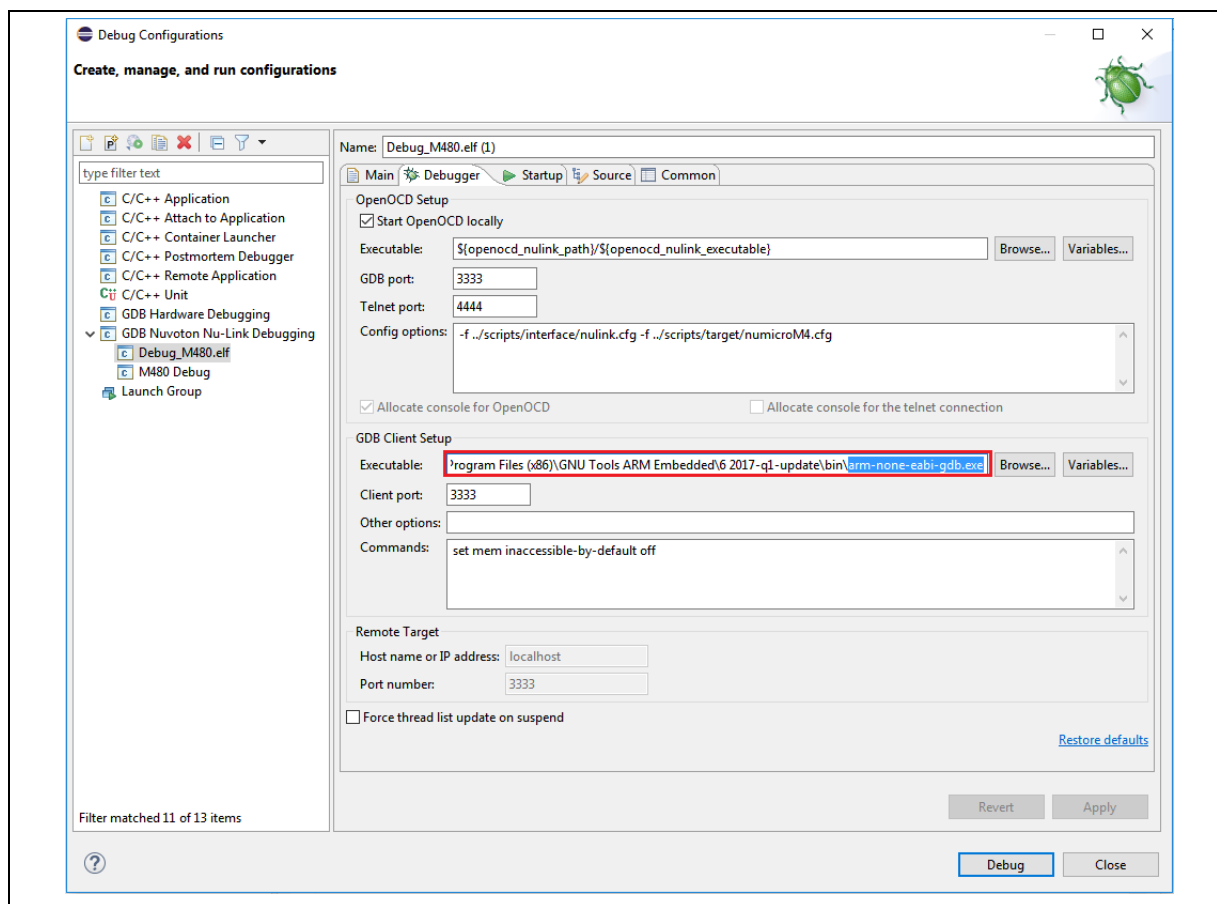


图 4-29 指定 GDB 执行档所在位置

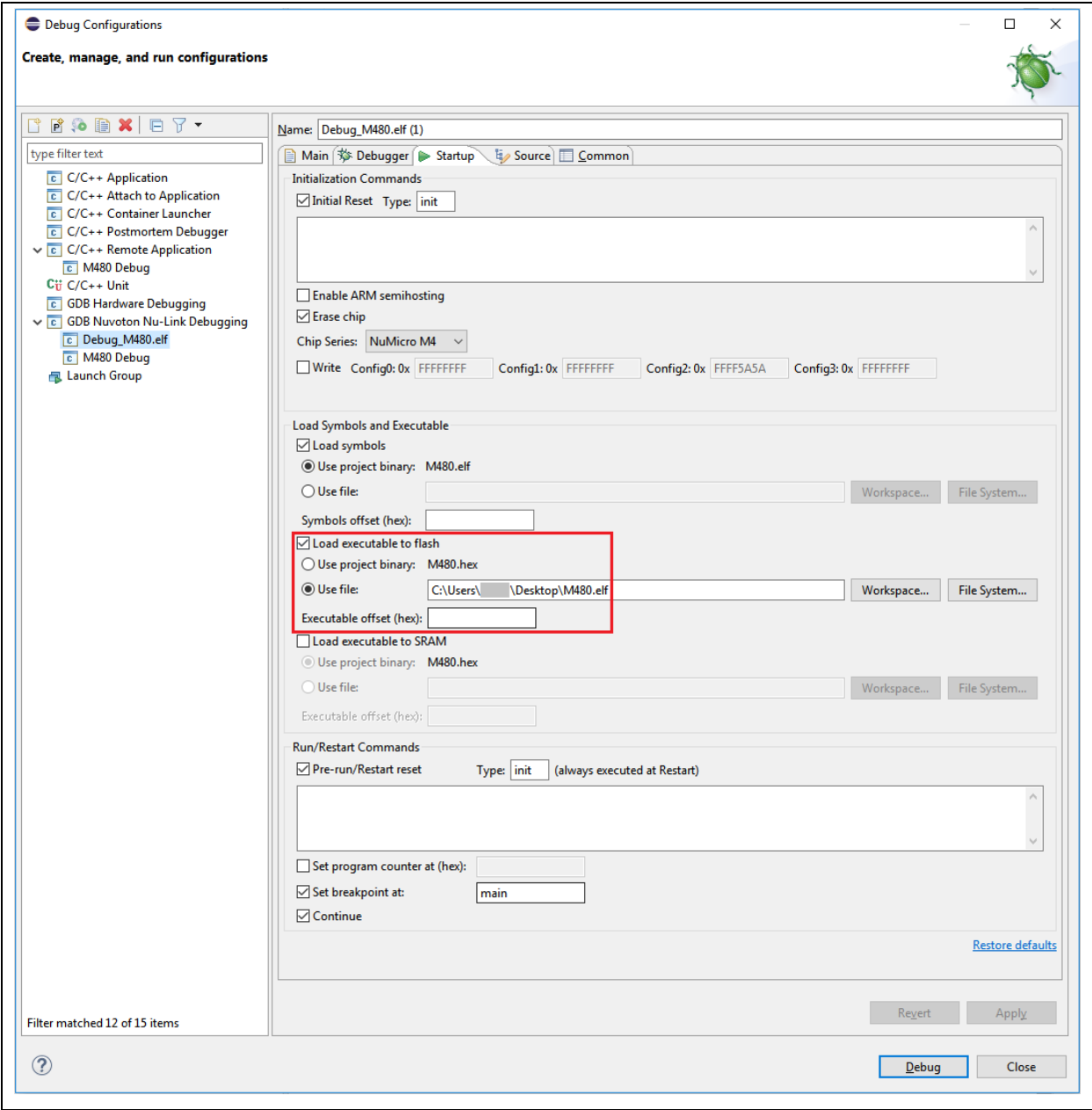


图 4-30 选择欲载入的 ELF 檔

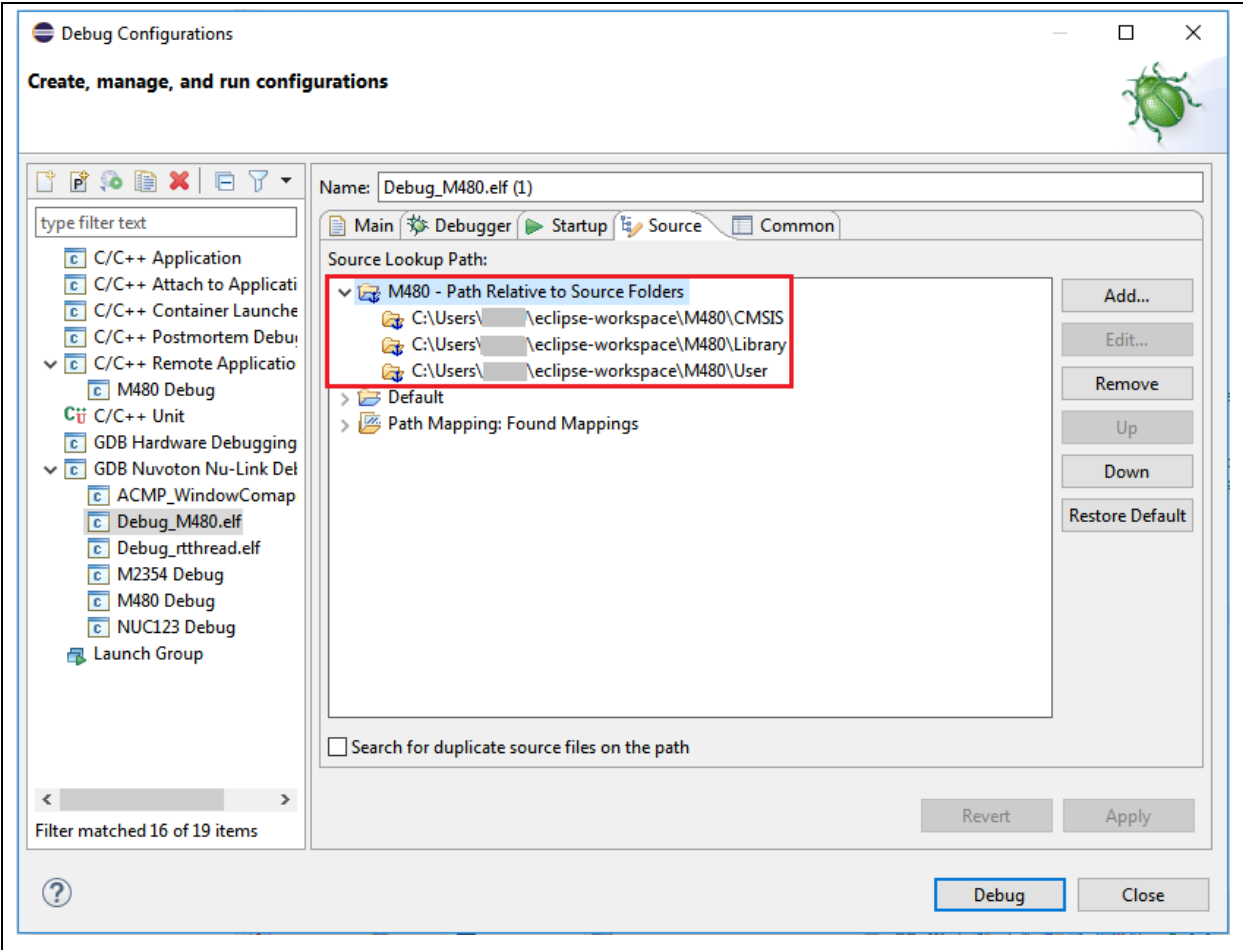


图 4-31 添加源码查找路径

4.10 Cortex-A、多核SMP、AMP系统配置

4.10.1 事先准备

建议使用 **Nu-Link2** 系列作为调试器，并确保升级 Nu-Link2 固件版本高于 **v3.09.7380r**。

Nu-Link2 从 **v3.09.7380r** 开始内置 CMSIS-DAP 接口，该 CMSIS-DAP 接口用于支持单 Cortex-A、多核SMP和AMP (Cortex-A + Cortex-M)。

此外，NuEclipse 为位于 **NuEclipse/Others** 文件夹中的 Cortex-A 系列提供了 GNU 工具链。配置此工具链后（参见第3.1.3节），我们可以构建 MA35D1 的项目。

4.10.2 调试单核Cortex-A的配置

本节介绍如何配置单个 Cortex-A 内核调试。

以新唐 MA35D1 为例，它具有双 Arm Cortex-A35 内核和一个 Arm Cortex-M4 内核。要调试它，我们应该选择 **NuMicro A35** 并在 Startup 选项中的可执行文件开始运行的位置重新开始目标芯片，如下所示。更多详细信息，请参阅第4.5节。要下载其他映像，命令为：**monitor load_image** 文件名地址。

要加载其他脚本，命令为：**monitor script filename**。

例如。加载 DDR 初始脚本，如下所示。

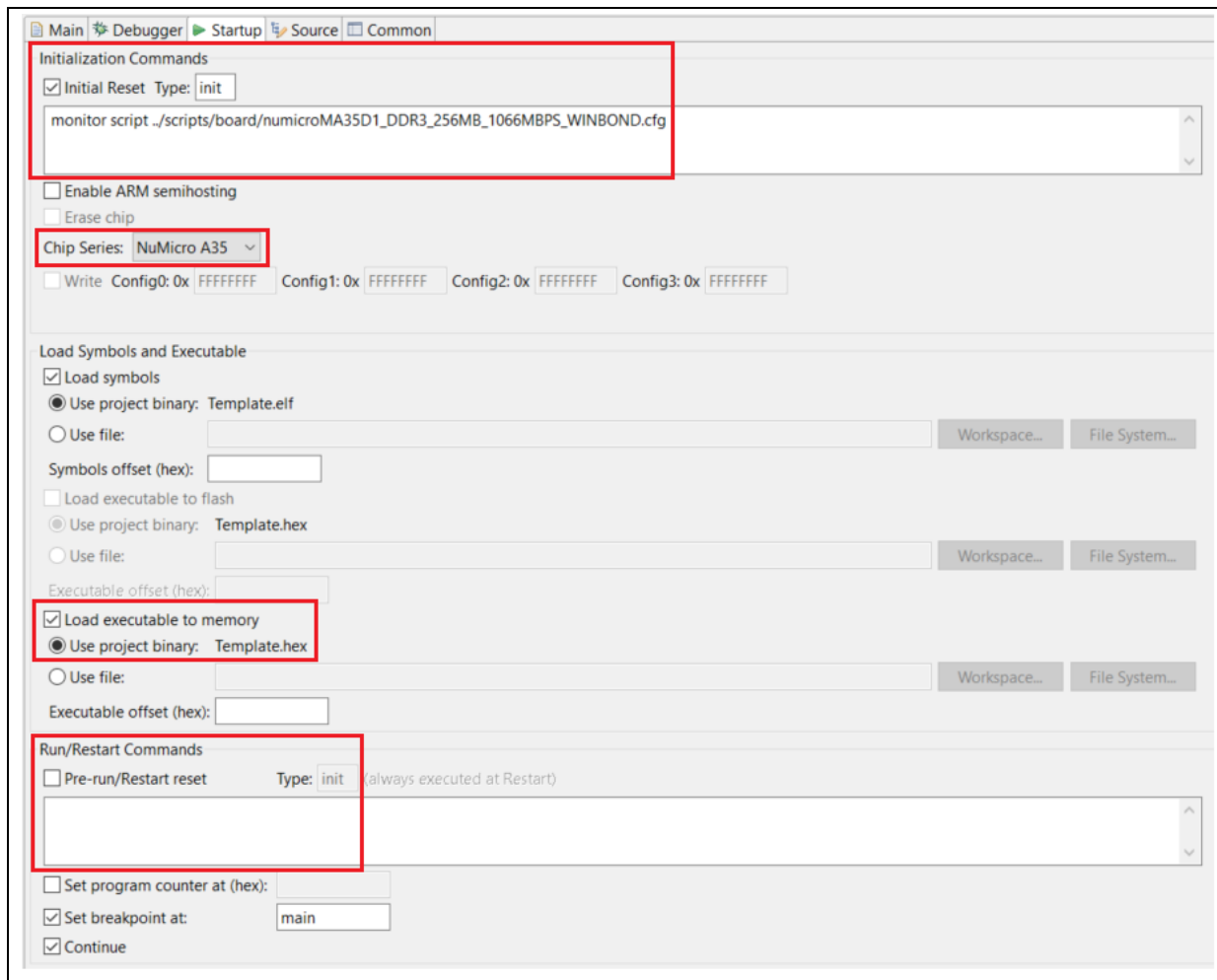


图 4-32 设置 MA35D1 的 Startup 标签页

当MA35D1的图像檔太大时，载入命令可能会遭遇超时。 为了避免这种情况，前往Window > Preferences > C/C++ > Debug > GDB并取消或增加命令超时上限。

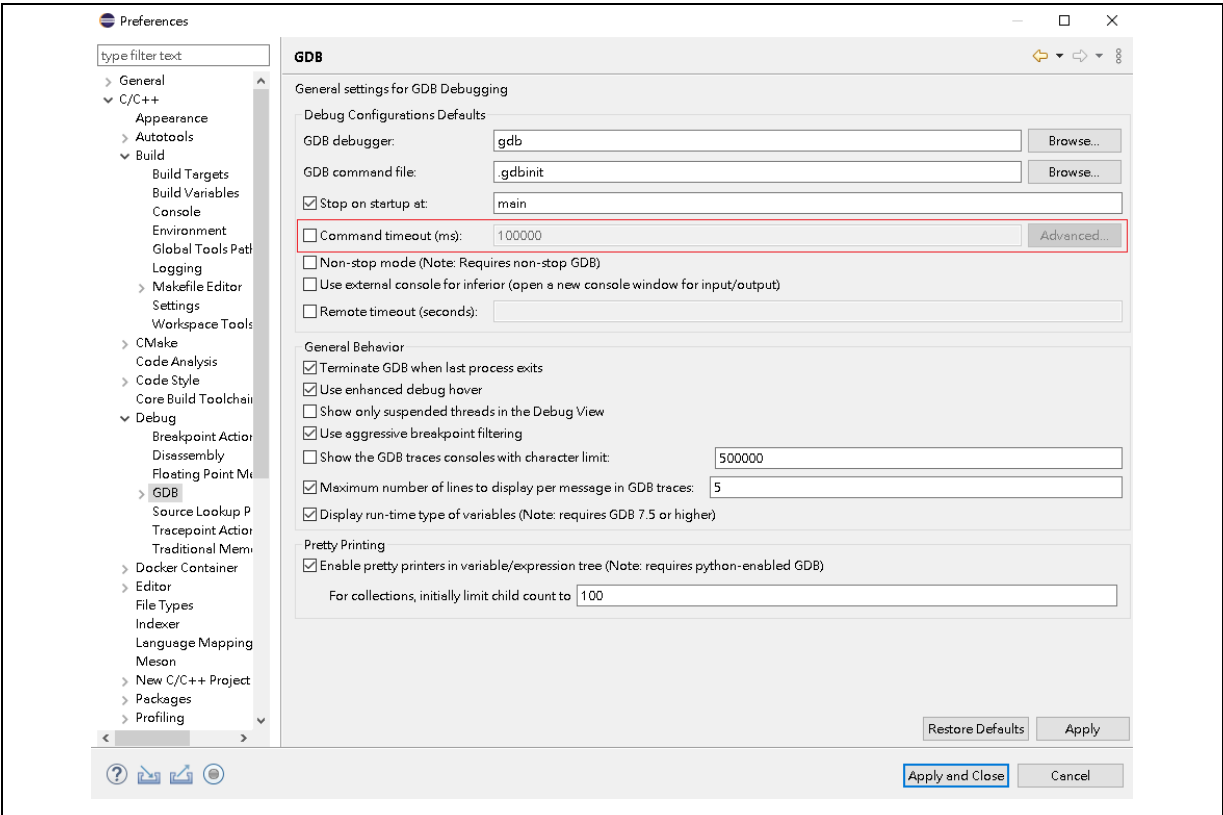


图 4-33 如何取消命令超时上限

要调试单个 A35 内核，我们必须通过注释掉 numicroMA35D1.cfg 中的以下代码来禁用 CPU1 设置。在 Windows 上，numicroMA35D1.cfg 所在的位置可能是 C:\Program Files (x86)\Nuvoton Tools\OpenOCD\scripts\target。

在 GNU/Linux 上，它可能是 /usr/local/OpenOCD/scripts/target。

```
cti create $_CHIPNAME.cti.sys -dap $_CHIPNAME.dap -ap-num 1 -baseaddr 0xE0801000
cti create $_CHIPNAME.cti.cpu0 -dap $_CHIPNAME.dap -ap-num 1 -baseaddr 0xE0C20000
#cti create $_CHIPNAME.cti.cpu1 -dap $_CHIPNAME.dap -ap-num 1 -baseaddr 0xE0D20000
cti create $_CHIPNAME.cti.cm4 -dap $_CHIPNAME.dap -ap-num 2 -baseaddr 0xE0042000

# so defer-examine it until the reset framework get merged
# NOTE: keep ap-num and dbgbase to speed-up examine after reset
# NOTE: do not change the order of target create
target create $_CHIPNAME.axi mem_ap -dap $_CHIPNAME.dap -ap-num 0
target create $_CHIPNAME.cpu0 aarch64 -dap $_CHIPNAME.dap -ap-num 1 -coreid 0 -dbgbase 0xE0C10000 -cti $_CHIPNAME.cti.cpu0
#target create $_CHIPNAME.cpu1 aarch64 -dap $_CHIPNAME.dap -ap-num 1 -coreid 1 -dbgbase 0xE0D10000 -cti $_CHIPNAME.cti.cpu1
target create $_CHIPNAME.cm4 cortex_m -dap $_CHIPNAME.dap -ap-num 2

targets $_CHIPNAME.cpu0

#target smp $_CHIPNAME.cpu0 $_CHIPNAME.cpu1
$_CHIPNAME.cpu0 aarch64 maskisr on
#$_CHIPNAME.cpu1 aarch64 maskisr on

adapter srst delay 200
reset_config srst only
adapter speed 1000

proc cpu0_init {} {
    $_CHIPNAME.cpu0 aarch64 dbginit
}

#proc cpu1_init {} {
#    $_CHIPNAME.cpu1 aarch64 dbginit
#}

$_CHIPNAME.cpu0 configure -rtos hwthread
#$_CHIPNAME.cpu1 configure -rtos hwthread
$_CHIPNAME.cpu0 configure -event reset-assert {cpu0_init}
#$_CHIPNAME.cpu1 configure -event reset-assert {cpu1_init}
$_CHIPNAME.cm4 configure -event reset-assert {halt}

$_CHIPNAME.cpu0 configure -event gdb-detach { shutdown }
```

图 4-34 禁用 CPU1 设置

在Startup标签页中选择NuMicro A35之后，numicroMA35D1.cfg将自动更新在Debugger标签页的Config options字段中。由于进行单核调试，因此OpenOCD GDB端口和GDB客户端端口应为相同的值，例如3333。完成所有设置后，单击Apply按钮以生效。为启动应用程序进入调试模式，请点选Debug按键。

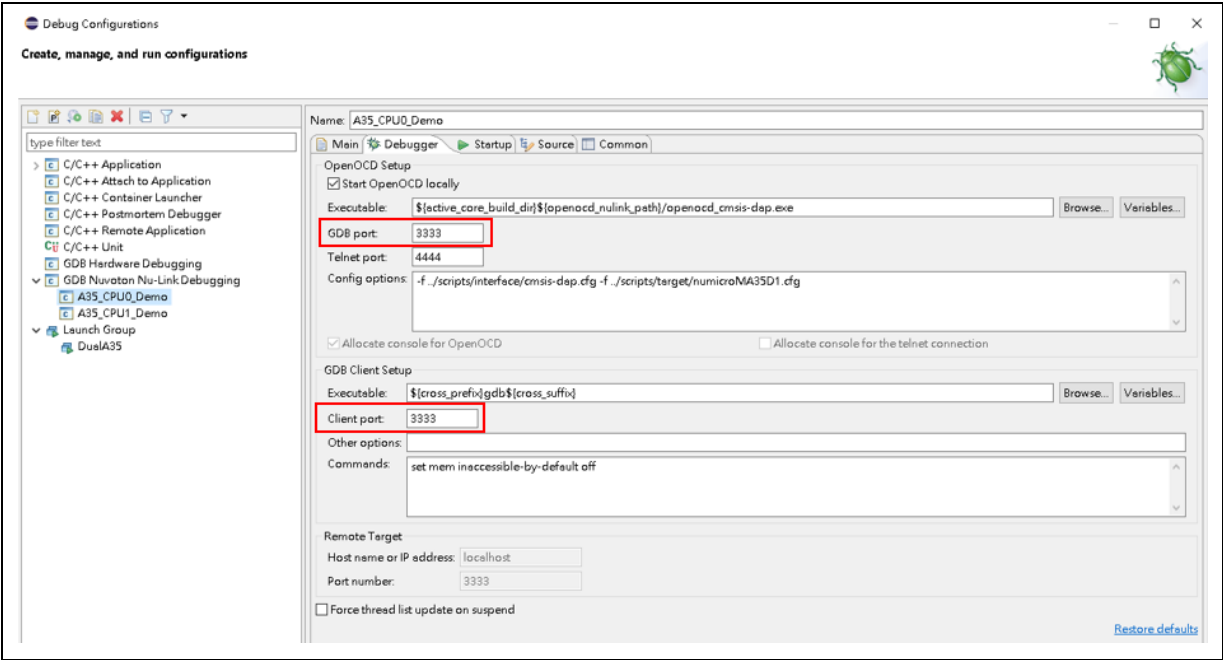


图 4-35 设置 Debugger 标签页以在 MA35D1 上进行单核调试

4.10.3 SMP模式下调试双核Cortex-A的配置

调试 SMP，例如 双 A35 核心，请参考第 4.10.2 节，但绕过禁用 CPU1 设置的步骤。

4.10.4 在单个 Cortex-A 内核和 Cortex-M (AMP) 上进行多核调试

调试AMP，例如 1个A35核心和M4同时，我们创建A35和M4工程并生成相应的调试配置（请参见第4.5节）。A35的调试配置遵循上一节的内容以完成设置。但是，M4调试配置的Debugger标签页与前者不同。取消Start OpenOCD locally复选框。GDB客户端端口和远程目标端口的值设置为3334。

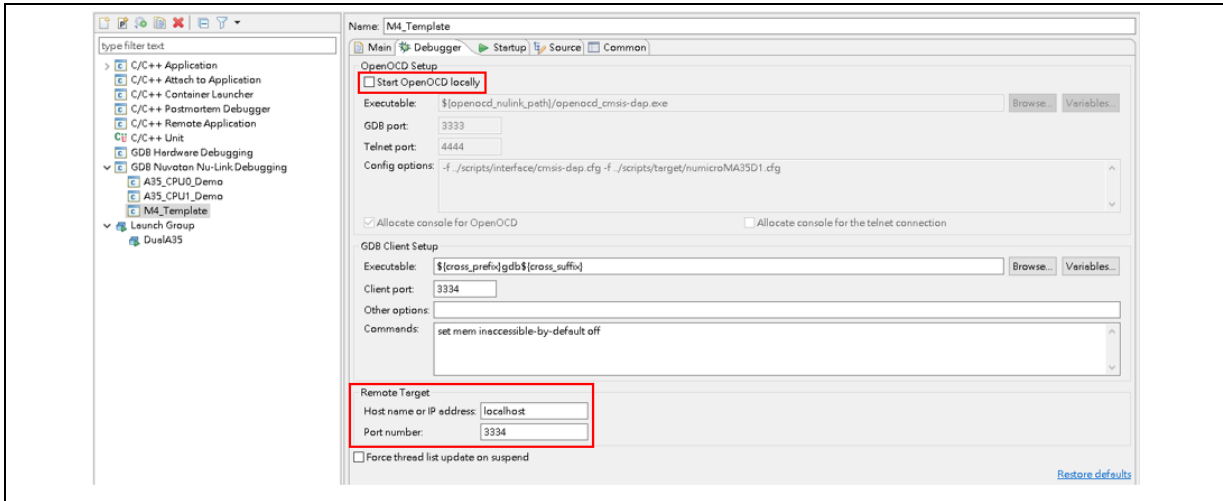


图 4-36 设置 Debugger 标签页以在 M4 内核上进行多核调试

要同时启动A35和M4应用程序，我们通过以下步骤使用启动群组：

- 1. 为新创建的启动群组命名。
- 2. 点选Add按钮。
- 3. 将预期的启动配置添加到启动群组。
- 4. 点选OK按钮。
- 5. 移动A35启动配置至**第一个**顺序。
- 6. 点选Apply按钮生效。
- 7. 点选Debug按钮以运行启动群组。

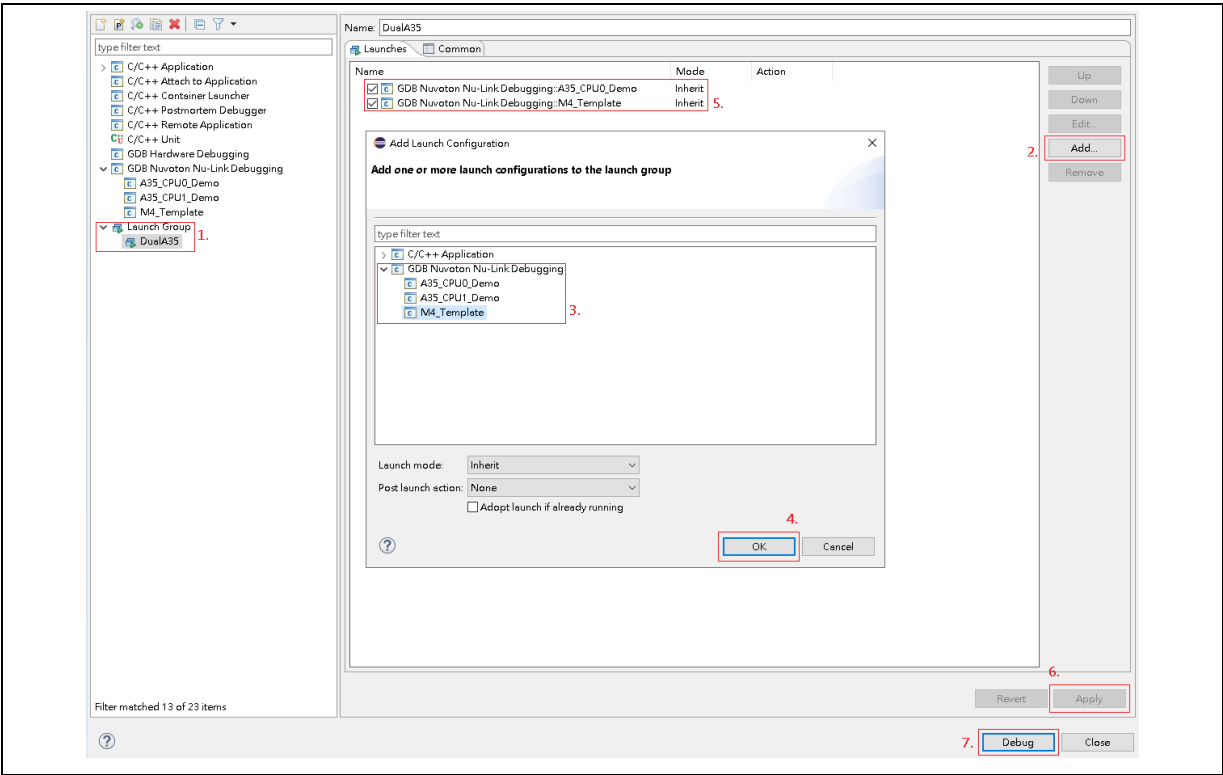


图 4-37 配置启动群组

4.10.5 双 Cortex-A 和 单 Cortex-M 上的多核调试

调试 AMP，例如1个A35核心和M4同时使用，我们必须通过在numicroMA35D1.cfg中注释掉以下代码来停用CPU对称多核处理器（SMP）。在Windows上，numicroMA35D1.cfg所在的位置可能是C:\Program Files (x86)\Nuvoton Tools\OpenOCD\scripts\target。

在GNU/Linux上，它可能是/usr/local/OpenOCD/scripts/target。

```
targets $_CHIPNAME.cpu0

#target smp $_CHIPNAME.cpu0 $_CHIPNAME.cpu1
$_CHIPNAME.cpu0 aarch64 maskisr on
$_CHIPNAME.cpu1 aarch64 maskisr on
```

图 4-38 如何停用 SMP

A35 CPU0的调试配置遵循第4.10.2节来完成设置。A35 CPU1和M4的调试配置遵循第4.10.3节来完成设置，但是GDB 客户端和远程目标端端口的值不同。A35 CPU1和M4端口值分别是3334和3335。启动群组应包括A35 CPU0，CPU1和M4的启动配置。确保A35 CPU0启动配置处于第一顺序，然后按Debug按钮运行启动群组。在进入调试模式时，我们可以通过在调试视图中单击相应的线程执行绪来切换其他CPU内核以进行调试。

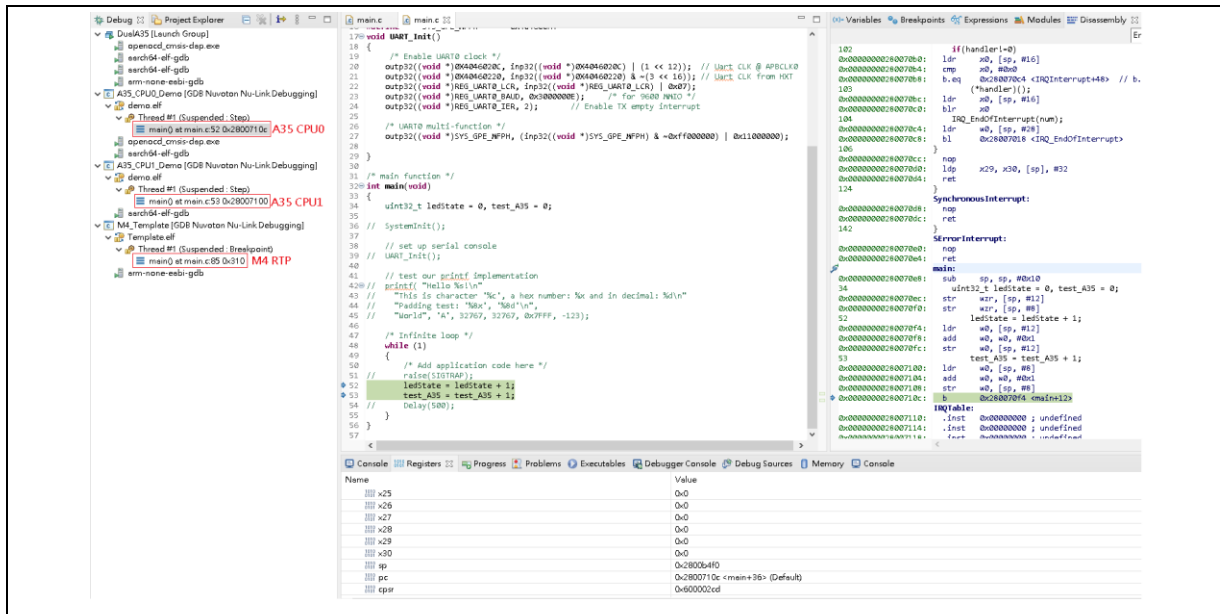


图 4-39 多核调试

4.10.6 AARCH32 程序调试於A35双核 (SMP)

章节4.10中的设置主要针对 ARM v8a AARCH64 模式，默认情况下。在本节中，我们将描述如何调试 AARCH32 应用程序。

例如，在 Cortex-A35 上调试 AARCH32 程序，我们首先进入调试模式以运行 AARCH64 代码并退出调试模式。然后重新进入调试模式切换到AARCH32代码进行调试。我们导入可执行文件（参见第 4.9节）并创建相应的调试配置（参见第4.5节）。在 Windows 上，numicroMA35D1.cfg 所在的位置可能是 C:\Program Files (x86)\Nuvoton Tools\OpenOCD\scripts\target。在 GNU/Linux 上，它可能是 /usr/local/OpenOCD/scripts/target。有几个步骤要遵循：

1. 汇入执行档进行调试（请参阅图 4-26）。
2. 单击Select executable之后的Browse，然后选择一个执行档（请参阅图 4-27）。
3. 选择GDB Nuvoton Nu-Link Debugging作为启动配置（请参阅图 4-28）。
4. 在调试配置中指定GDB执行档（请参阅图 4-40）。
5. 在调试配置中选择欲载入的ELF文件（请参阅图 4-41）。请根据开发板的DDR选择对应的DDR初始文件。

这些文件可以在 C:\Program Files (x86)\Nuvoton Tools\OpenOCD\scripts\board 文件夹中找到。

```
(monitor script ../scripts/board/numicroMA35D1_DDR3_256MB_1066MBPS_WINBOND.cfg)
```

6. 点选Debug按键。
7. 在aarch64程序的最后一个地址设置断点，继续到该地址。
8. 点选运行按键然后代码并退出调试模式。
9. 取消选中的重置设置并重新进入调试模式（请参阅图 4-42）。
10. 调试 AARCH32 程序（请参阅图 4-43）。

以上描述用于在双核 SMP 模式下调试 AARCH32 程序。AARCH32程序调试也可以在以下两种设置中使用

1. 参考图 4-34 禁用单核模式的 CPU1 设置。
2. 参考图 4-38 禁用 AMP 模式下的 CPU SMP。

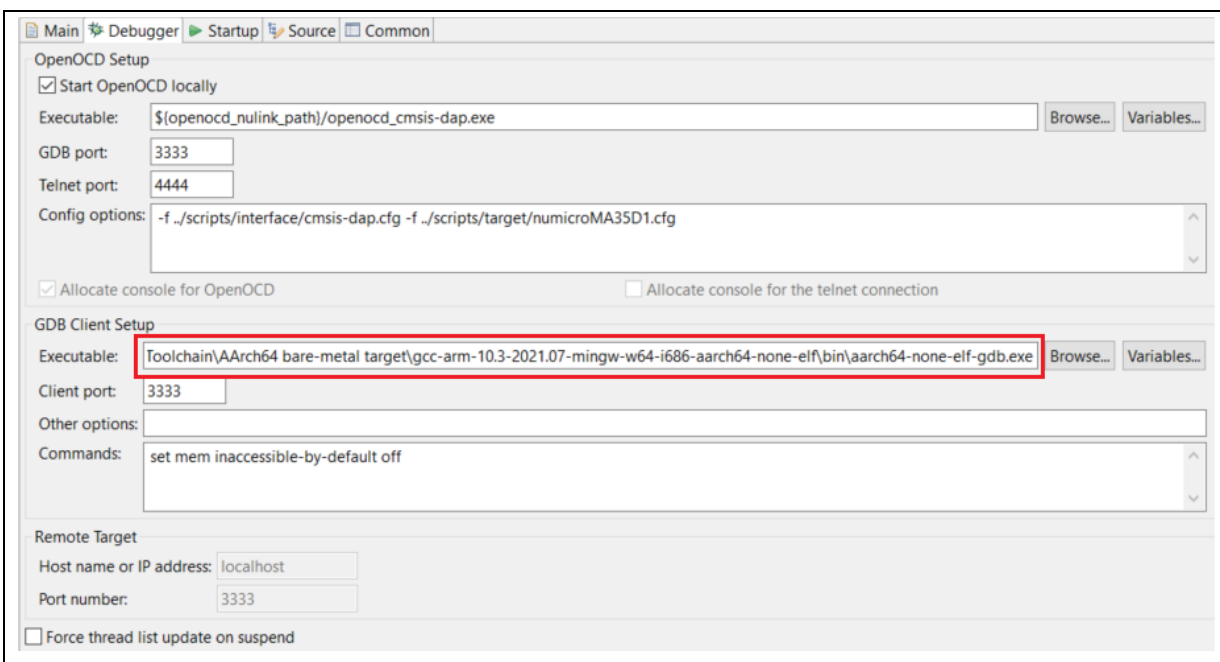


图 4-40 指定 AARCH64 GDB 执行档所在位置

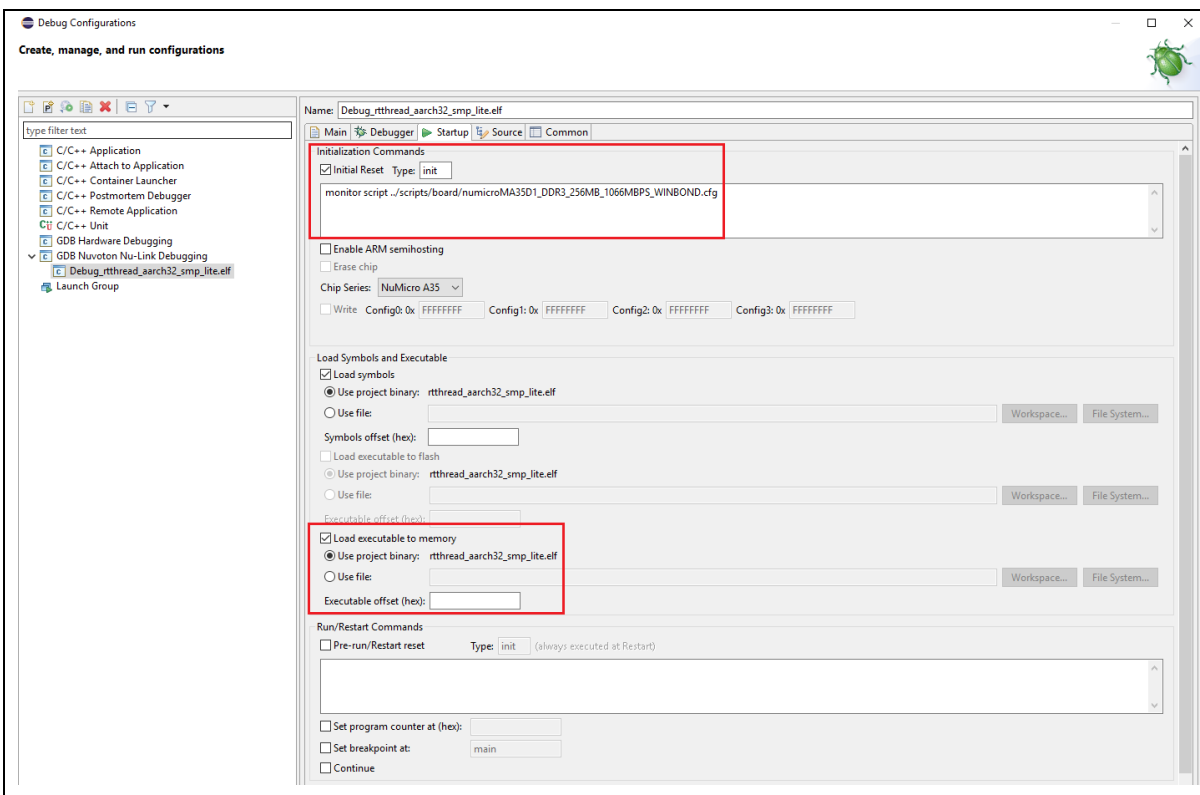


图 4-41 首次进入调试模式设置

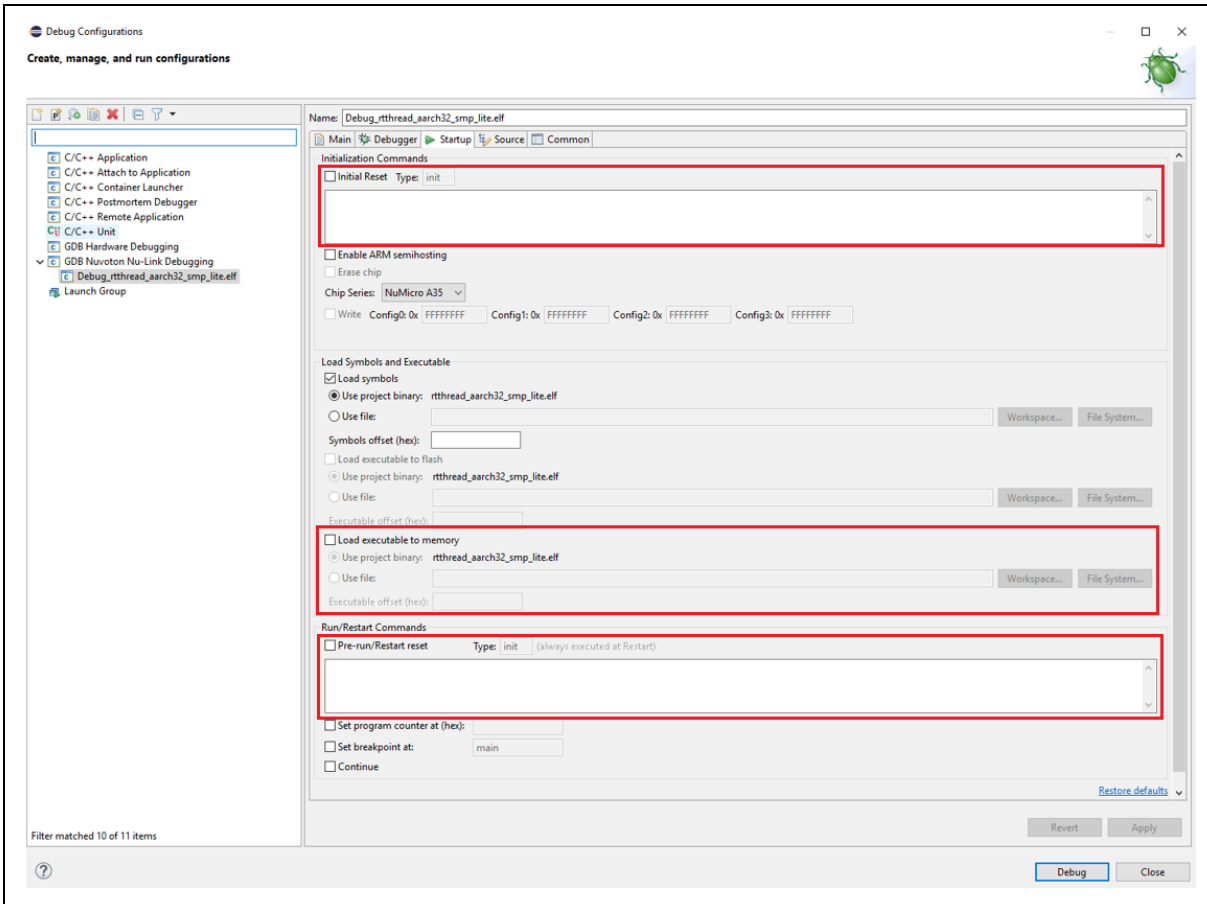


图 4-42 第二次进入调试模式设置

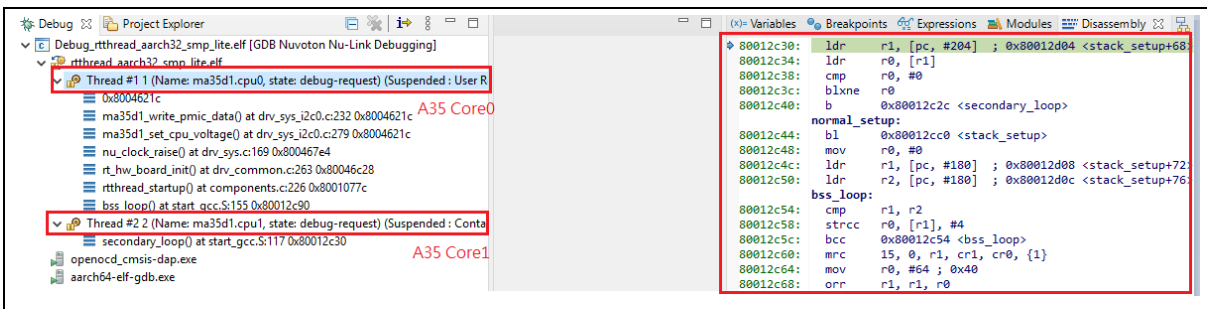


图 4-43 AARCH32 程序调试

4.11 NuMicro 8051 1T配置

请从 Nuvoton 网站下载 NuEclipse_Windows (For NuMicro 8051)。

4.11.1 汇入和建置工程

当 NuMicro 8051 1T BSP 项目可用时，我们可以将它们导入到Workspace中。更多详细信息，请参阅第 4.3 章节。

参考第4.4章节构建工程。构建项目成功后，Console View 会显示校验值并生成bin文件。

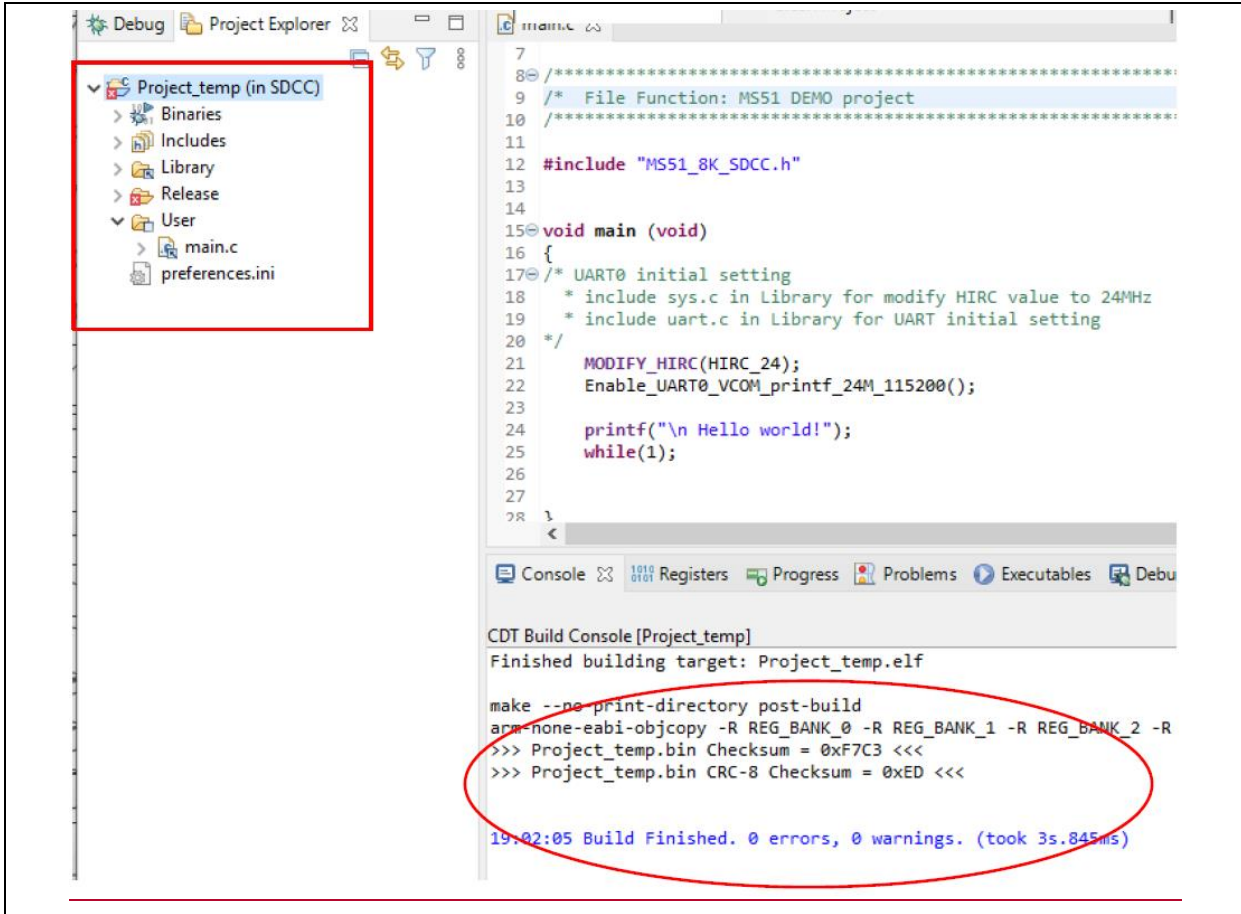


图 4-44 建置 NuMicro 8051 1T 工程

4.11.2 调试配置

本节介绍如何配置NuMicro 8051 1T调试。

在Debugger标签页中，**Config options**字段的配置文件是numicro8051_1T.cfg。gdb.exe的安装路径必须在GDB Client Setup Executable字段中设置。参见图 4-45。

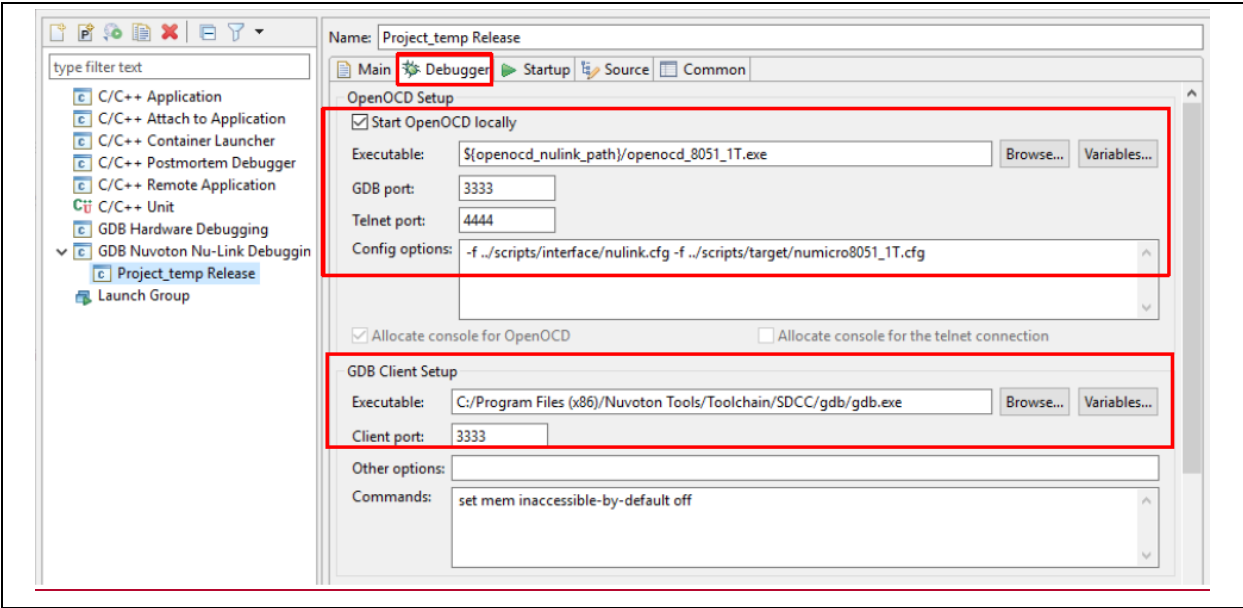


图 4-45 Debugger 标签页中的 NuMicro 8051 1T 调试配置

4. 11. 3 监控不同的内存空间

NuEclipse提供不同的8051内存空间访问。变量可以显式分配给特定的内存空间（通过在声明中包含内存空间说明符）或隐式分配（基于内存模型）。
根据不同的内存空间，必须指定相应的偏移位置，才能访问特定的内存空间数据。请参考下表。

Memory space of SDCC declaration	Data access address
__code	(CODE address) + 0
__data	(DATA address) + 0xF0000000
__idata	(IDATA address) + 0xF1000000
__xdata	(XDATA address) + 0xF2000000

表 4-1 使用偏移量访问内存空间

例如，要访问 __xdata 内存的 0x10 位置，用户需要将 0xF2000010 填入内存浏览器或内存视图。参见图 4-46。

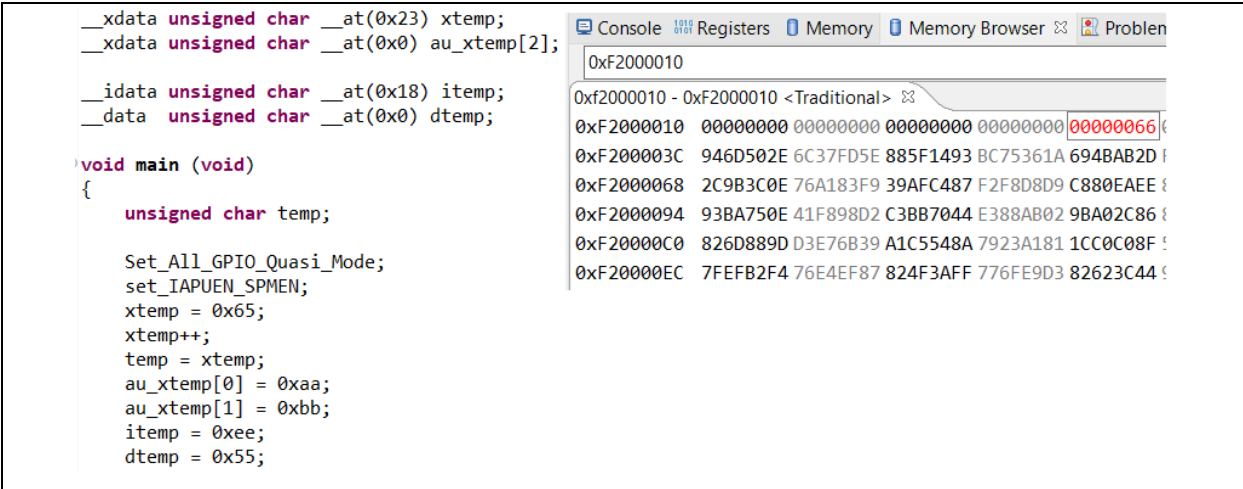


图 4-46 使用内存浏览器访问内存空间

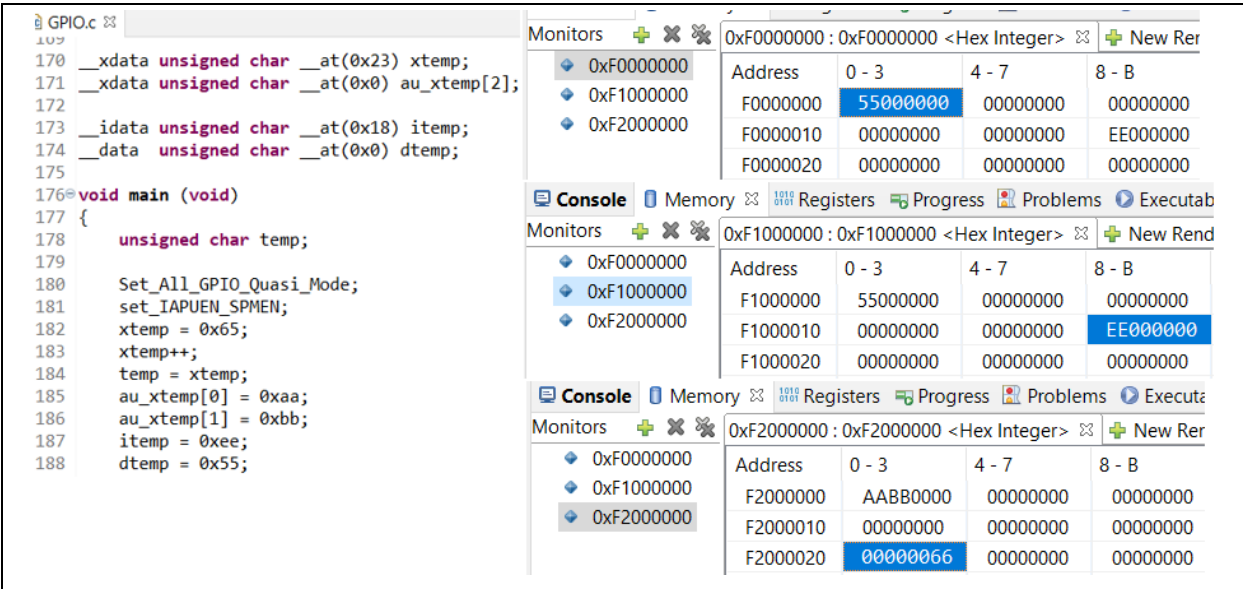


图 4-47 使用内存视图访问内存空间

要监视变量值，用户可以通过**Watch Expressions View**观察变量，并添加指定的偏移地址来访问不同内存空间中的变量值。

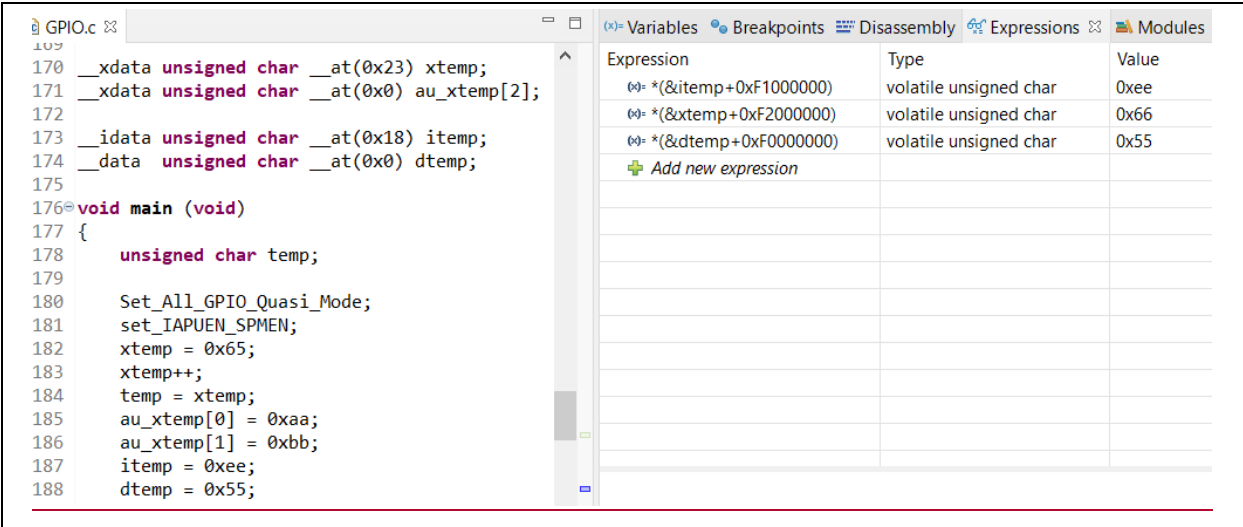


图 4-48 使用监视表达式视图访问变量

5 Q&A

1. Q: 我们能同时在Eclipse、Keil和Iar调试吗?

A: 不行, 我们必须停下Eclipse的调试模式, 才能在别的IDE上调试。

2. Q: 我们能在Eclipse上调试时同时使用新唐的开发软件(例如ICP Programming Tool)吗?

A: 不行, 我们必须停下Eclipse的调试模式, 才能使用它们, 反之亦然。

3. Q: 总共支持几个断点和检测点?

A: 视硬件而定。M0有4个断点, 2个检测点。M4有6个断点, 4个检测点。M23有4个断点, 4个检测点。8051 1T有8个断点。目前, 我们不支持闪存断点。

4. Q: 如何更新Nu-Link的固件?

A: 请使用ICP Programming Tool或Keil来更新固件, 访问
https://github.com/OpenNuvoton/Nuvoton_Tools 网站获取更多信息。

5. Q: 工程创建后要如何修改闪存和内存定义的大小?

A: 请至ldscripts文件夹找到并打开ld文件, 那里可以修改它们。

6. Q: 为何应用程序进不去调试模式?

A: 首先, 我们需安装所有必要的软件, 请参照3.1节的说明。接着检查下面的列表:

- 如果正在调试的话, 离开上一次的调试模式。
- 闪存和内存定义的大小要正确。
- 进入调试前, OpenOCD不该处在运行中。为检查它有没有运行, 请打开微软任务管理器或系统监视器。如果看到OpenOCD进程正在跑, 请关掉该进程。
- 目标芯片不应该被别的IDE和工具占据住。
- Debugger标签页的**Config options**字段要正确。
- Debugger标签页的**GDB Client Setup Executable**字段要正确。
- Startup标签页的**Initilal Reset type**应为 **init**。**Pre-run/Restart reset type**应为 **init**。
- Eclipse Preferences的内容要正确, 请参阅3.1.3节。
- **将Nu-Link固件和驱动程序升级到最新版本。**
- 检查工程下是否成功生成hex文件。
- 检查执行文件是否已正确地被写入目标芯片。
- 检查SP。如果它是错的, 请指定它至正确的位置。
- 写入正确的Config值至目标芯片。
- 如果操作系统是Windows 7, 请使用USB 2.0端口, 而不是USB 3.0端口。
- 工程路径不能存在任何特殊字符或空白, 例如#\$&`. {}。
- 重开机计算机。

7. Q: 在GNU/Linux上, 如何为Nu-Link新增udev规则?

A: 当透过Nu-Link存取目标芯片时, GNU/Linux需要USB权限。我们可为Nu-Link新增udev规则以获得该权限。这里列出步骤:

- 新增User至plugdev群组。在Terminal输入下面指令:

```
sudo useradd -G plugdev $USER
```

- 新增Nu-Link至udev。在Terminal输入下面指令:

```
cd /etc/udev/rules.d和sudo gedit 10-openocd-nulink.rule
```

- 新增下面字符串至该文件:

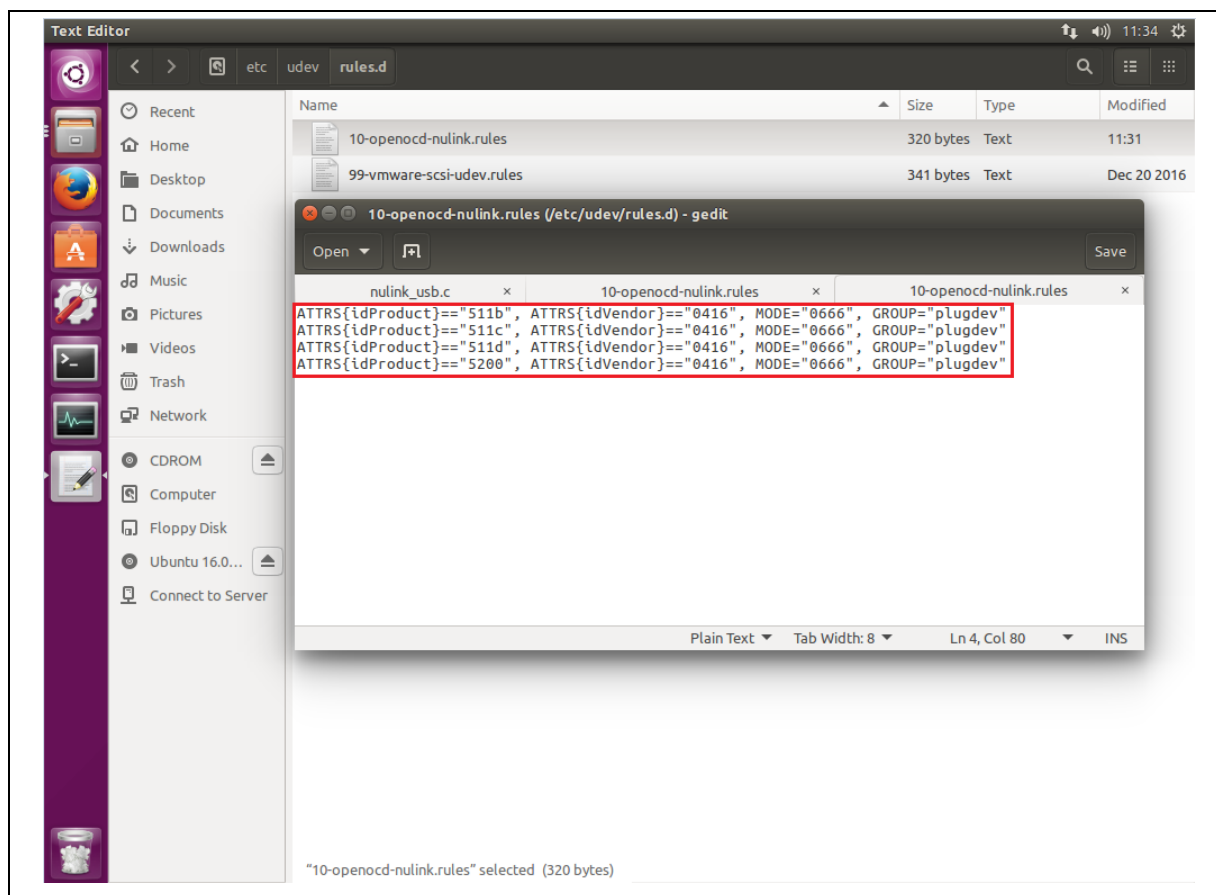


图 5-1 新增 Udev 规则

- 重载新的udev规则。在Terminal输入下面指令:

```
sudo udevadm trigger
```

8. Q: 如何编辑openocd_nulink_path字符串?

A: openocd_nulink_path字符串存放OpenOCD执行文件的所在位置。升级NuEclipse后, 该字符串可能会维持在上一版OpenOCD的路径。为修正它, 单击Window > Preferences。Preferences对话框会出现。前往Run/Debug > String Substitution。找到并编辑openocd_nulink_path字符串, 使其指向新的OpenOCD路径。

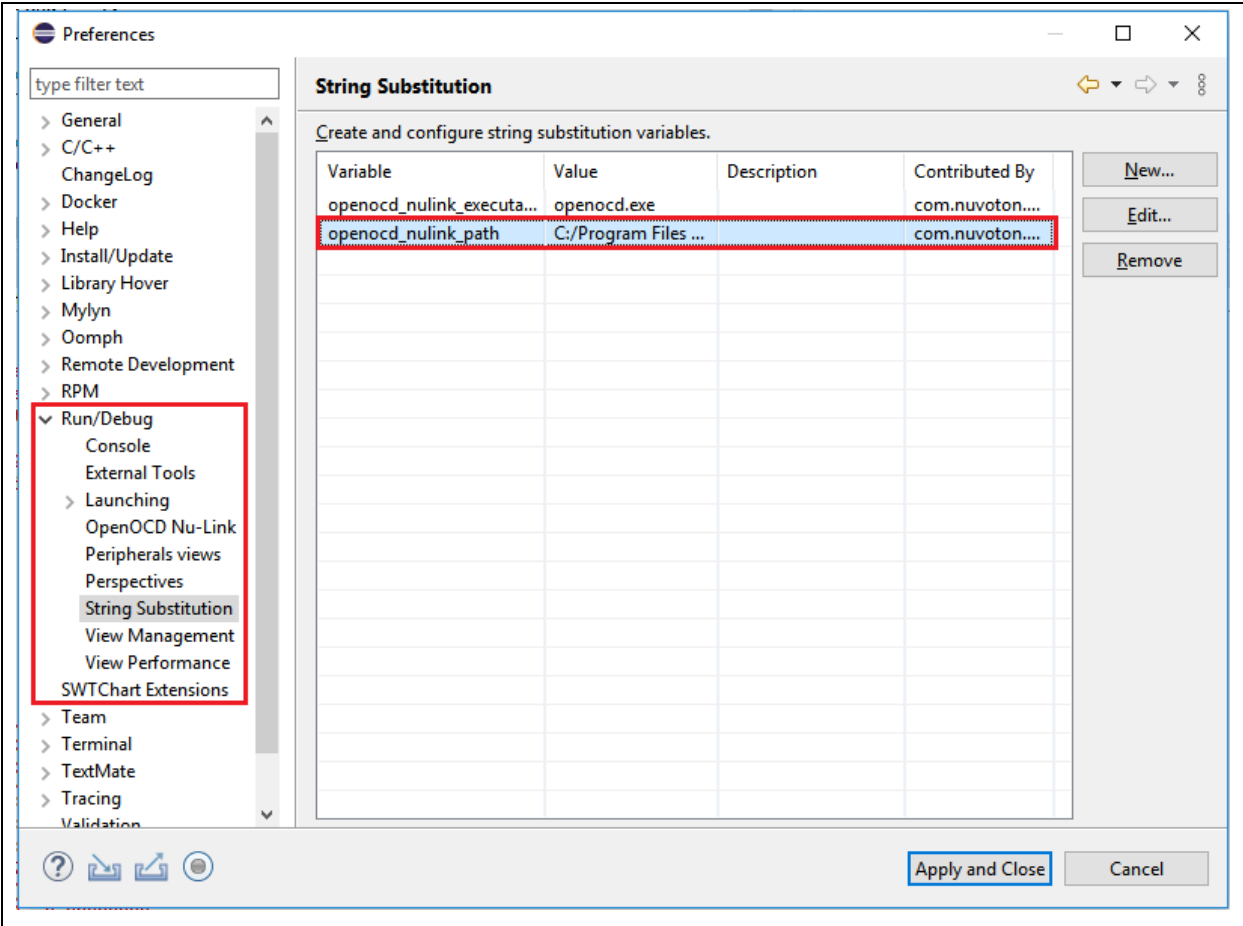


图 5-2 Preferences 设置 String Substitution 的对话框

9. Q: 在Windows上, 为何更新或安装软件Packs会失败呢?

A: 可能的原因之一是Windows文件夹的写入权限被拒绝。 我们需要找到正确的文件夹并允许写入权限。 在Windows上, 我们放置软件Packs的位置是C: \ Program Files (x86) \ Nuvoton Tools \ Packages。

10. Q: Nu-Link2如何调整输出电压?

A: 升级NuLink2固件v3.08.7249以上版本, 在弹出的“NuMicro MCU” 盘中打开NU_CFG.TXT文件, 会看到NU_CFG.TXT中有一些选项。

设置CMSIS-DAP=1, 然后重新插入USB线, 它会多出一个接口HID_CMSIS-DAP, 这很方便使用CMSIS-DAP协议。

设置LEVEL=3然后重新插入USB线, 它显示输出电压之一, 这很方便使用5V电压。

```
[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
```

图 5-3 NuLink2 的更多选项

6 修订历史

Date	Revision	Description
2017.03.31	0.01.000	Alpha版发布。
2017.06.30	1.01.000	Beta版发布
2018.09.15	1.01.013	官方版发布
2018.11.30	1.01.014	1. 支持NUC505。 2. 更新新工程向导
2019.08.09	1.01.015	支持M031、M261和M480LD。
2020.03.06	1.01.016	支持M031BT、NUC1311、M2354和M479。
2020.09.30	1.01.017	支持M030G、M071、M0A21、M251和M471。
2021.03.12	1.01.018	支持M030GM031G和NUC1262。
2021.12.01	1.01.019	支持M460, I94100 and KM1M7。
2022.04.28	1.02.019	支持MA35D1。
2023.02.07	1.02.022	支持8051 1T。
2023.05.26	1.02.023	支持M2L31。
2023.08.18	1.02.024	支持MG51, N76S003, MUG51。

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.