

## 前言

**Eclipse**是著名的跨平台的自由集成开发环境（IDE），通过安装不同的插件**Eclipse**可以支持不同的计算机语言，比如**C/C++**等，**Eclipse**本身只是一个框架平台，众多的**Eclipse**插件使其拥有很高的灵活性，软件可以以**Eclipse**为架构开发自己的IDE。

这边文档主要描述怎样用现成的**Eclipse**插件来调试**AT32**系列芯片。

支持型号列表：

支持型号	AT32F 系列
------	----------

## 目录

<b>1</b>	<b>概述.....</b>	<b>5</b>
<b>2</b>	<b>Eclipse 调试环境安装.....</b>	<b>6</b>
2.1	Eclipse IDE for C/C++ Developers 安装.....	6
2.2	GNU ARM Eclipse 插件安装.....	6
2.3	ARM GCC 编译工具链安装.....	9
2.4	GNU ARM Eclipse Build Tools 安装 .....	12
2.5	安装 Jlink .....	13
<b>3</b>	<b>template 工程配置与编译.....</b>	<b>14</b>
3.1	打开 template 工程 .....	14
3.2	编译 .....	15
<b>4</b>	<b>调试.....</b>	<b>19</b>
4.1	Jlink 调试 .....	19
4.1.1	Debug 配置.....	19
4.2	ATLink 调试 .....	20
4.2.1	Debug 配置.....	20
<b>5</b>	<b>版本历史 .....</b>	<b>24</b>

## 表目录

表 1. 文档版本历史 .....	24
-------------------	----

## 图目录

图 1. AT32_Eclipse_Packet.zip 包含文件 .....	5
图 2. 进入 Install New Software 页面 .....	7
图 3. 选择 Add .....	7
图 4. Add Respository .....	7
图 5. 选择插件解压目录 .....	8
图 6. 勾选插件 .....	8
图 7. 安装完成 .....	8
图 8. accept the license agreement .....	9
图 9. Install anyway .....	9
图 10. 重启 Eclipse .....	9
图 11. Install .....	10
图 12. 安装向导 .....	10
图 13. 接受许可证协议 .....	10
图 14. 安装过程 .....	11
图 15. 选择 Add path to environment variable .....	11
图 16. 安装结果查看 .....	11
图 17. 运行安装包 .....	12
图 18. 选择安装路径 .....	12
图 19. 安装完成 .....	12

# 1 概述

本文档介绍通过使用Eclipse，ARM-GCC编译工具，GNU-ARM插件，Jlink，ATLink等资源来调试AT32系列芯片。

本文档主要介绍：

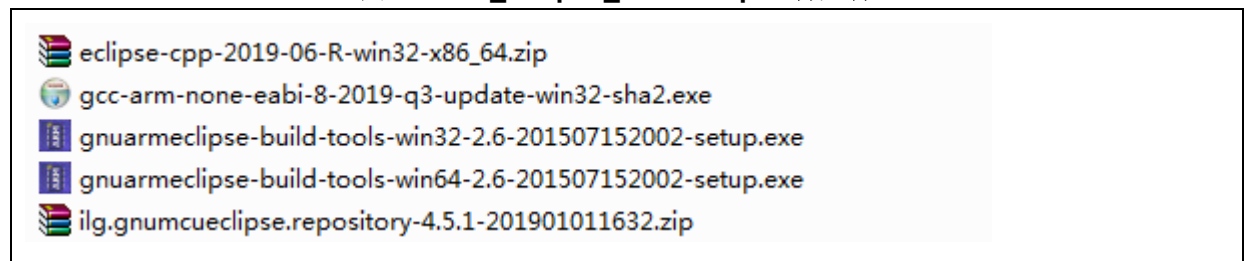
- Eclipse 调试环境安装
- Eclipse template工程
- Eclipse 编译配置
- Eclipse 调试配置

环境说明：本文档安装说明基于WINDOWS 7 x64系统下实现，使用AT32Fxx\_Firmware\_Library\project\at\_start\_xx\templates\eclipse\_gcc下的工程进行说明。

本文档所使用的软件都打包在AT32\_Eclipse\_Packet.zip，直接解压即可安装运行

AT32\_Eclipse\_Packet.zip包含文件：

图 1. AT32\_Eclipse\_Packet.zip 包含文件



## 2 Eclipse 调试环境安装

首先要明确需要安装哪些软件，如下是需要安装的软件：

- Eclipse IDE for C/C++ Developers
- GNU ARM Eclipse 插件
- GCC ARM 编译器
- GNU ARM Eclipse Build Tools 安装（make, rm等工具）

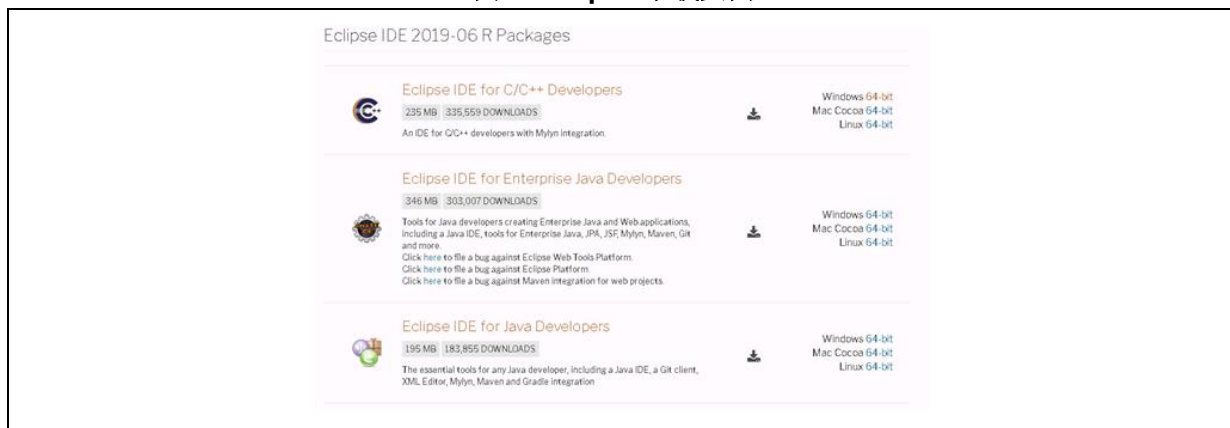
下面将详细介绍软件的安装部分。

### 2.1 Eclipse IDE for C/C++ Developers 安装

Eclipse有多个版本，我们只需下载C/C++的版本，下载最新版本，AT32\_Eclipse\_Packet.zip 包含有一个可以使用的版本eclipse-cpp-2019-06-R-win32-x86\_64.zip。

下载地址：<http://www.eclipse.org/downloads/eclipse-packages/>

图 2 Eclipse 下载页面



下载完成后直接解压eclipse-cpp-2019-06-R-win32-x86\_64.zip，这里我们直接点击eclipse.exe 就可以运行Eclipse，但此时还不能进行代码调试，需要进一步安装一下插件。

### 2.2 GNU ARM Eclipse 插件安装

下载最新的GNU ARM Eclipse插件：ilg.gnuarm.eclipse.repository-4.5.1-201901011632.zip并解压。

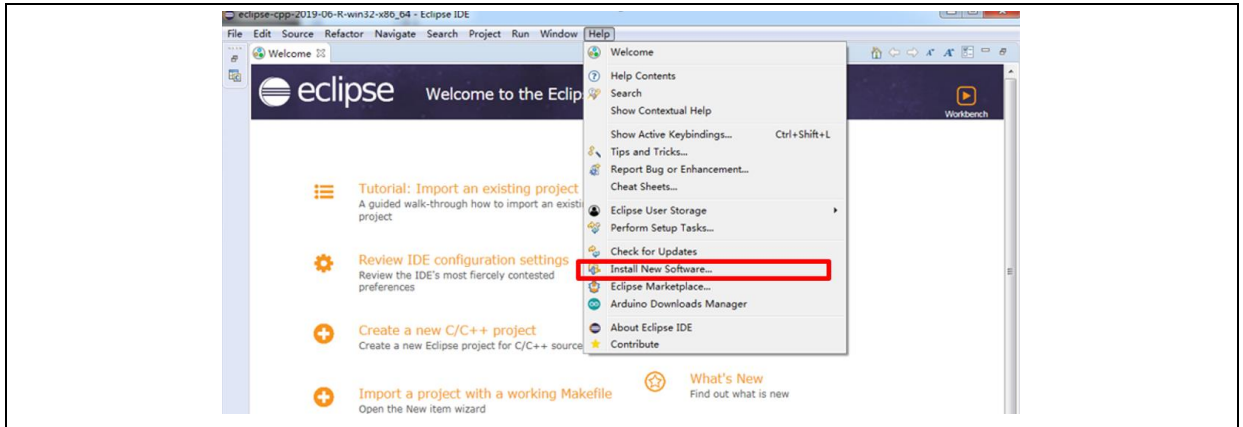
AT32\_Eclipse\_Packet.zip 包含有一个可以使用的版本ilg.gnuarm.eclipse.repository-4.5.1-201901011632.zip。

网络下载地址：<https://github.com/gnu-mcu-eclipse/eclipse-plugins/releases>

开始安装：

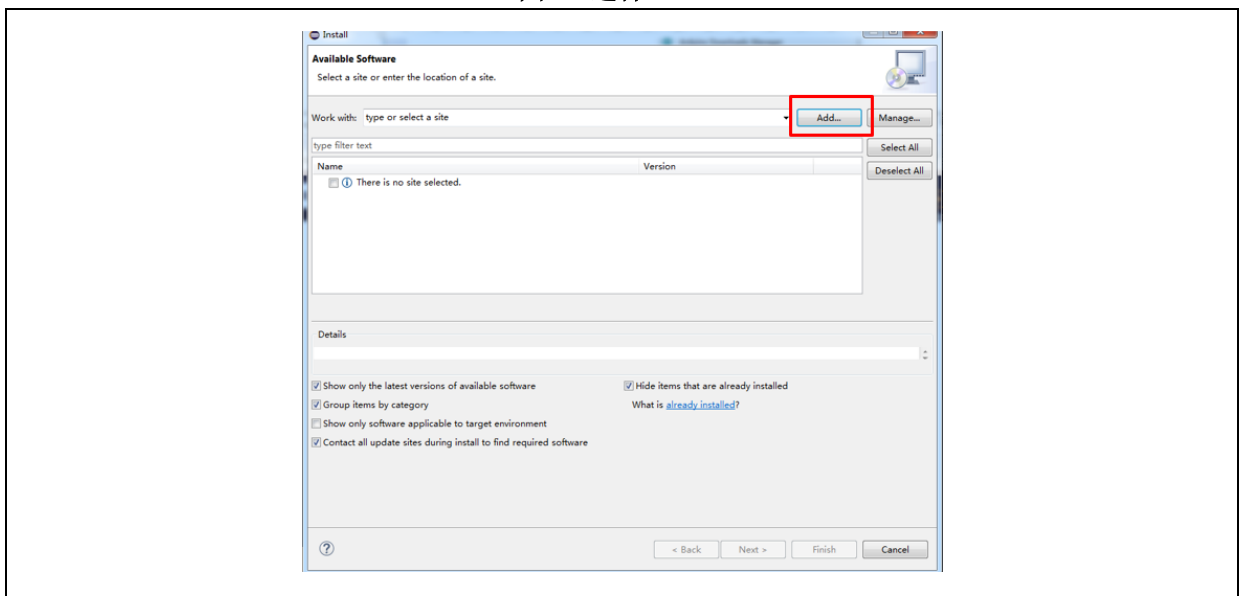
1. 打开Eclipse Help->Install New Software.

图 2. 进入 Install New Software 页面



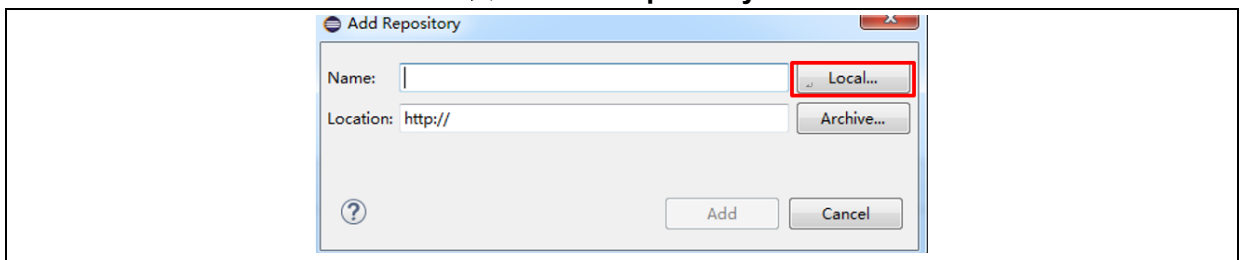
2. 点击 “Add...”

图 3. 选择 Add



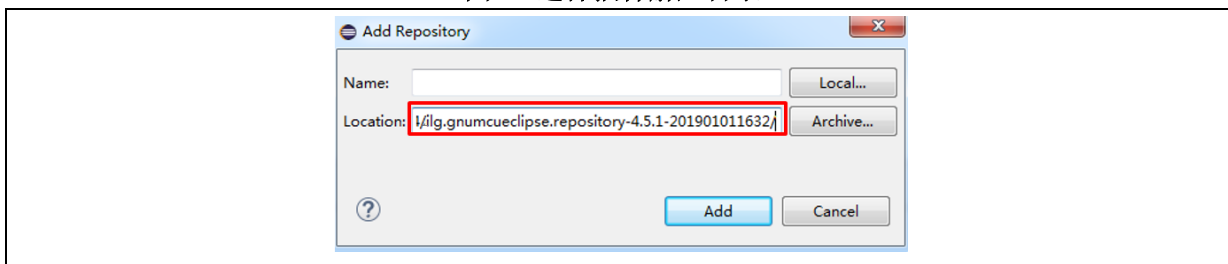
3. 添加一个本地插件，也可以使用网络路径自动下载安装

图 4. Add Repository



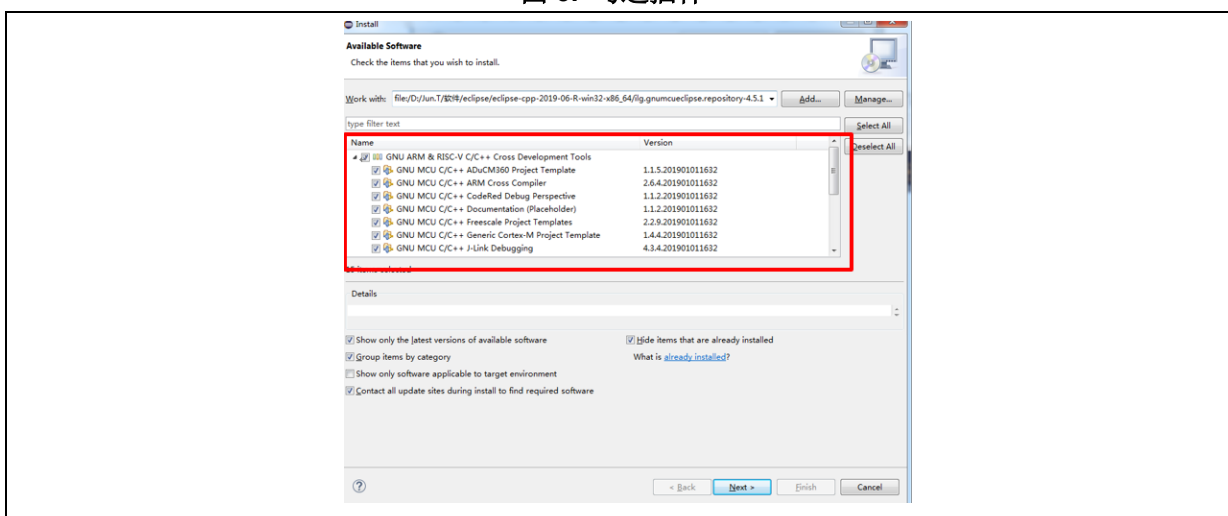
4. 选择本地插件解压目录，点击 “Add”

图 5. 选择插件解压目录



5. 勾选所有的插件，“Next”

图 6. 勾选插件



6. 安装完成 “Next”

图 7. 安装完成

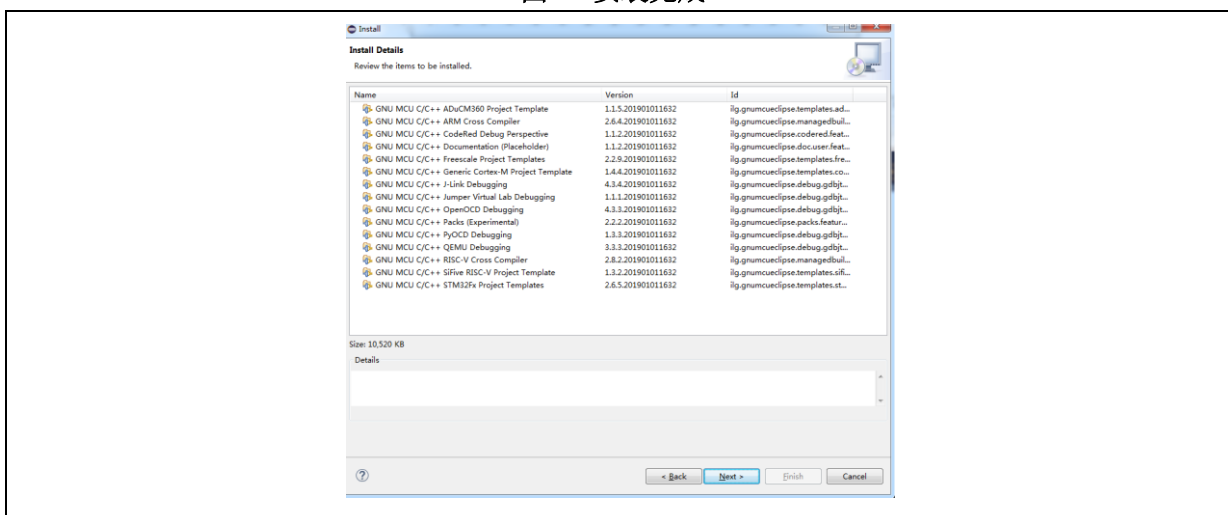
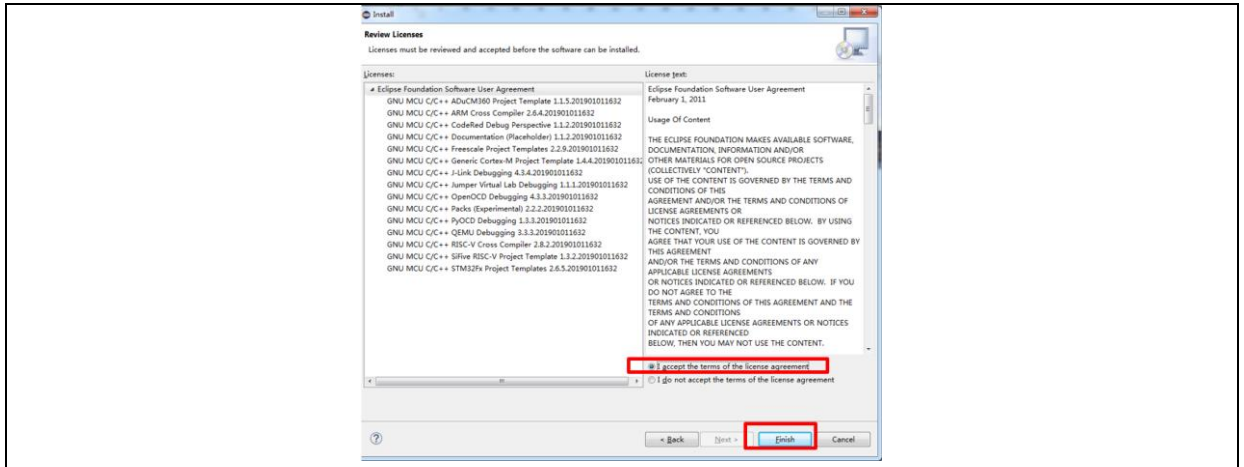


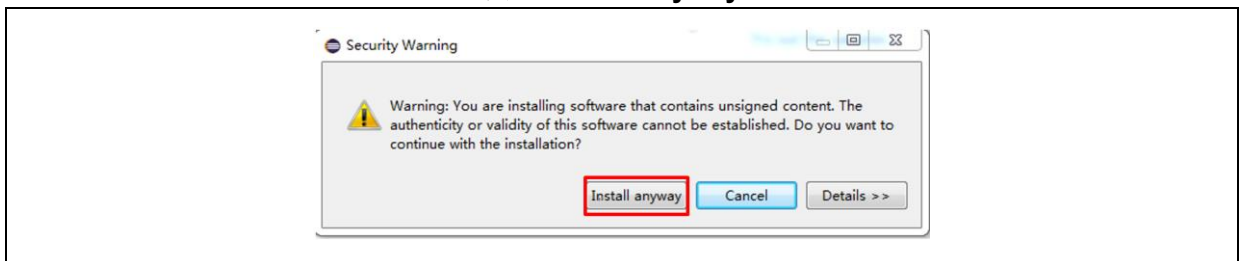


图 8. accept the license agreement



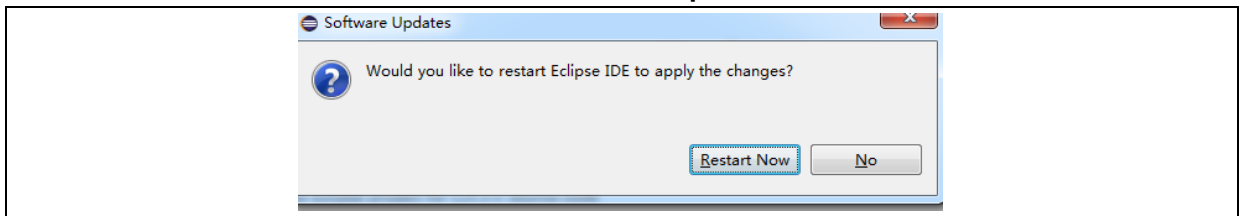
7. “Install anyway”

图 9. Install anyway



8. 重启Eclipse

图 10. 重启 Eclipse



## 2.3 ARM GCC 编译工具链安装

下载最新的编译工具链gcc-arm-none-eabi-8-2019-q3-update-win32-sha2.exe

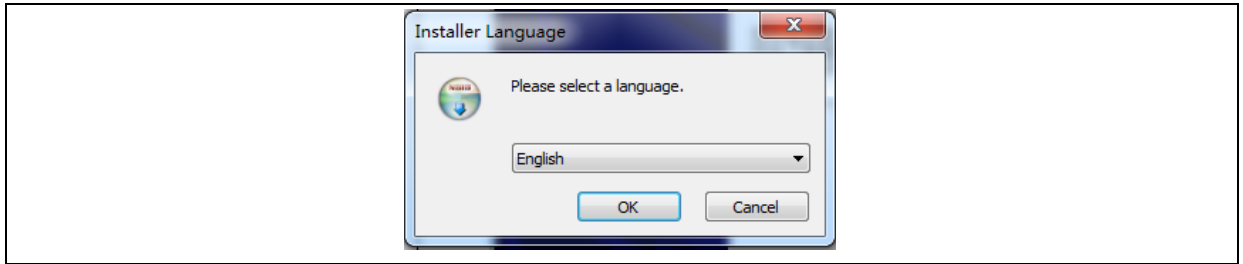
AT32\_Eclipse\_Packet.zip 包含有一个可以使用的版本gcc-arm-none-eabi-8-2019-q3-update-win32-sha2.exe。

下载地址: <https://launchpad.net/gcc-arm-embedded/+download>

开始安装:

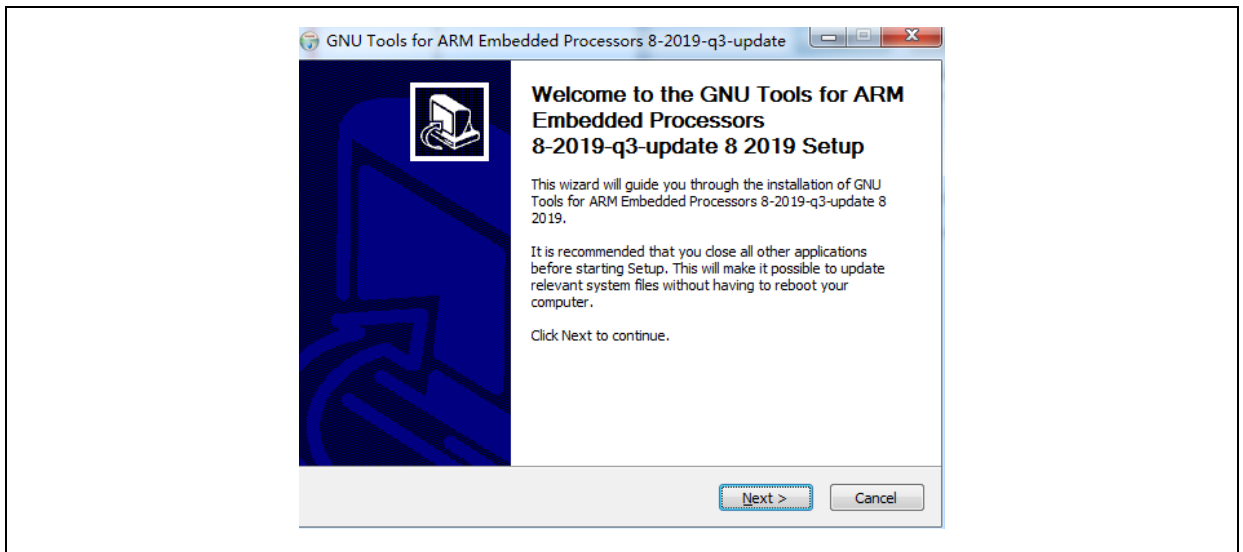
1. 选择语言

图 11. Install



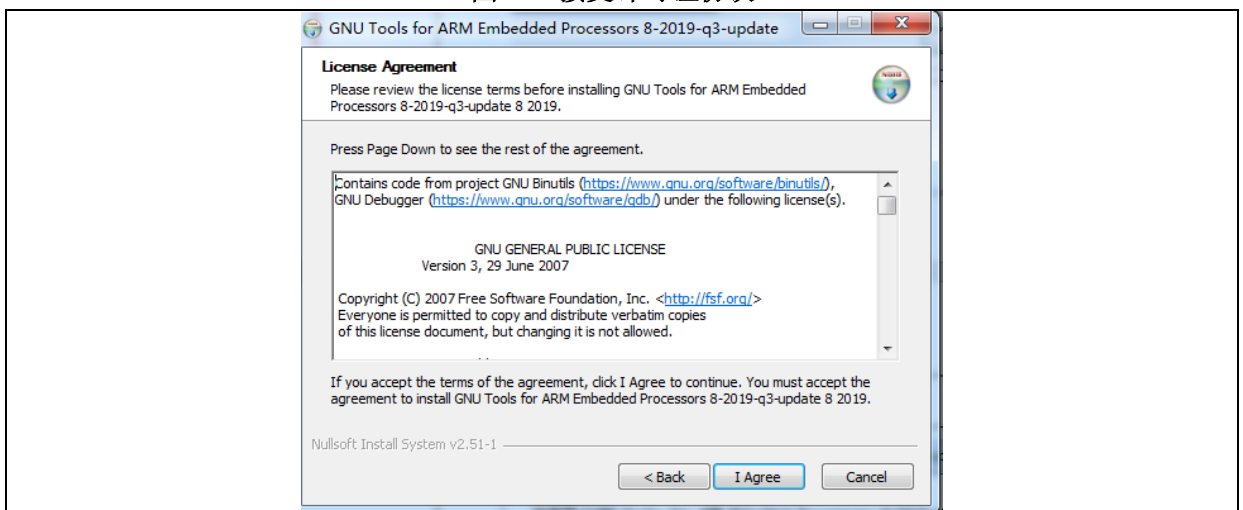
2. 安装向导，点击“下一步”

图 12. 安装向导



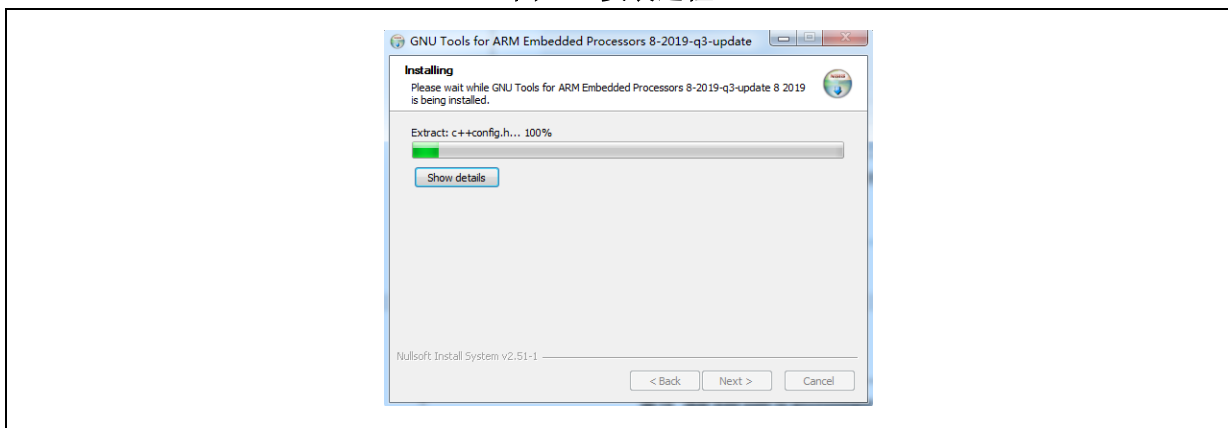
3. 许可证协议，点击“我接受“

图 13. 接受许可证协议



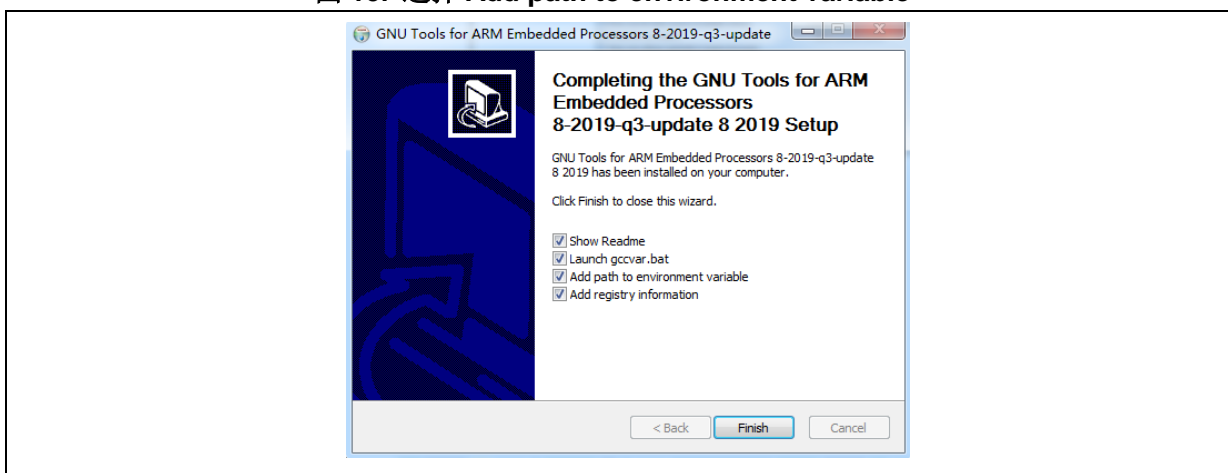
4. 选择安装位置，安装到默认位置，点击 “安装“，进入安装过程

图 14. 安装过程



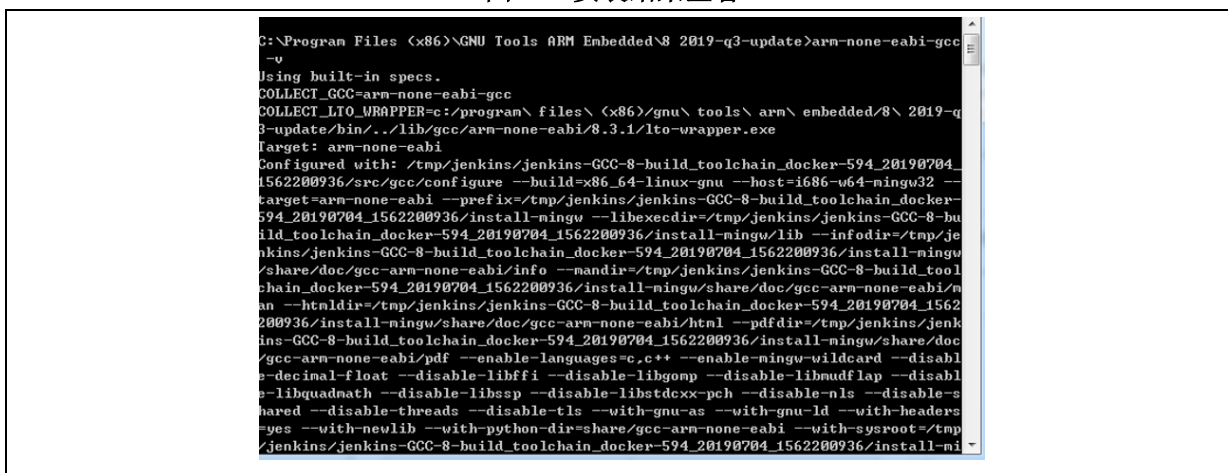
5. 安装完成界面，要将” Add path to environment variable “选项勾选上，会自动添加到环境变量之中，否则需要手动添加环境变量。

图 15. 选择 Add path to environment variable



6. 安装完成之后在弹出的命令窗口中输入 `arm-none-eabi-gcc -v`，会显示一些版本信息，表示安装成功。

图 16. 安装结果查看



## 2.4 GNU ARM Eclipse Build Tools 安装

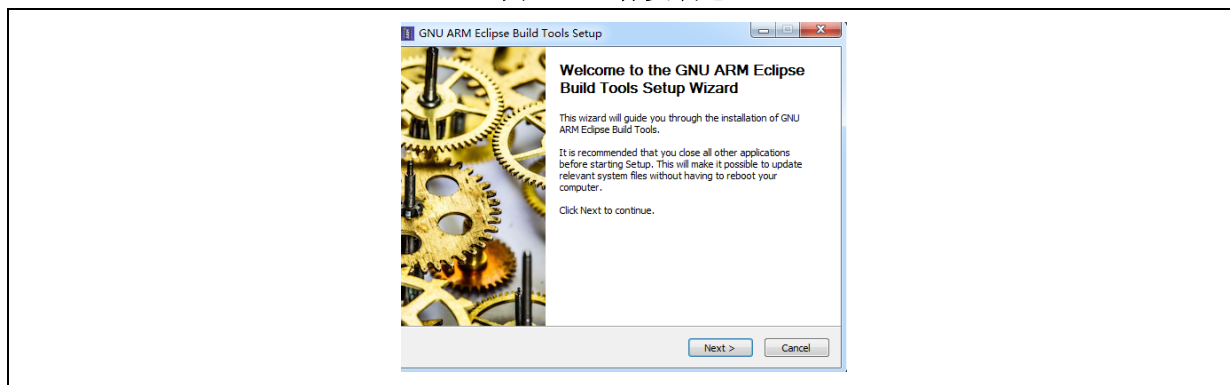
此部分主要是安装make，rm等命令。

下载地址：[https://sourceforge.net/projects/gnuarmeclipse/files/Build Tools/](https://sourceforge.net/projects/gnuarmeclipse/files/Build%20Tools/)

AT32\_Eclipse\_Packet.zip 包含有一个可以使用的版本gnuarmeclipse-build-tools-win64-2.6-201507152002-setup.exe，或者下载其它适用版本。

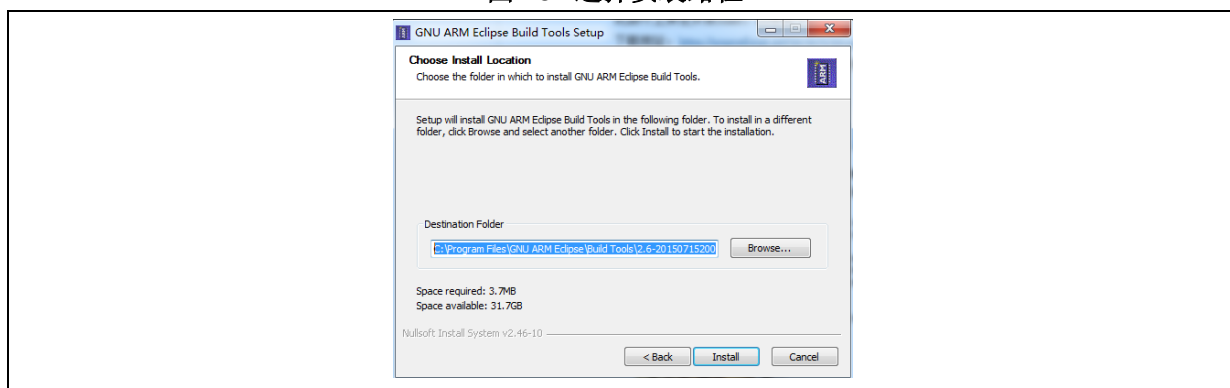
### 1. 运行安装包

图 17. 运行安装包



### 2. 选择安装路径

图 18. 选择安装路径



### 3. 安装完成 需要重启Eclipse

图 19. 安装完成



## 2.5 安装 Jlink

需要将AT32 系列芯片拷贝到Jlink目录，目前可使用ICP完成拷贝工作

### 1. Jlink 安装（略）

此部分按照下载最新的Jlink进行安装即可

### 2. 算法文件拷贝

为了Jlink能够识别和下载程序到AT32芯片，可以将AT32 芯片的下载算法拷贝到Jlink目录中，这里有个简单的方法是使用最新的ICP，直接运行ICP，ICP会将相应的AT32 算法拷贝的Jlink目录。

## 3 template 工程配置与编译

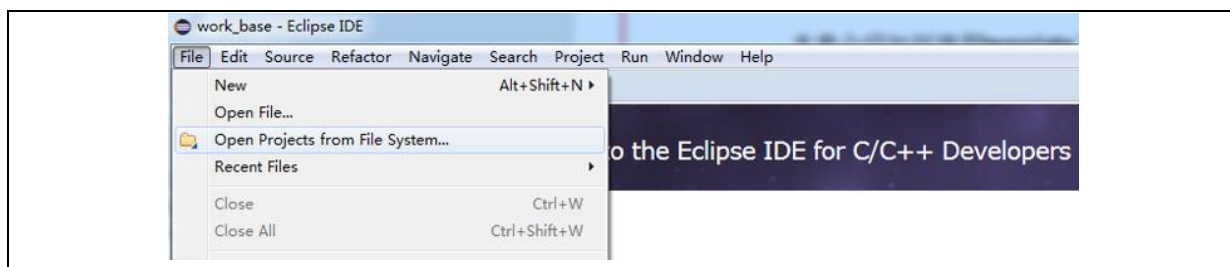
本章介绍如何使用template工程，工程路径：

AT32Fxx\_Firmware\_Library\project\at\_start\_xx\templates\eclipse\_gcc

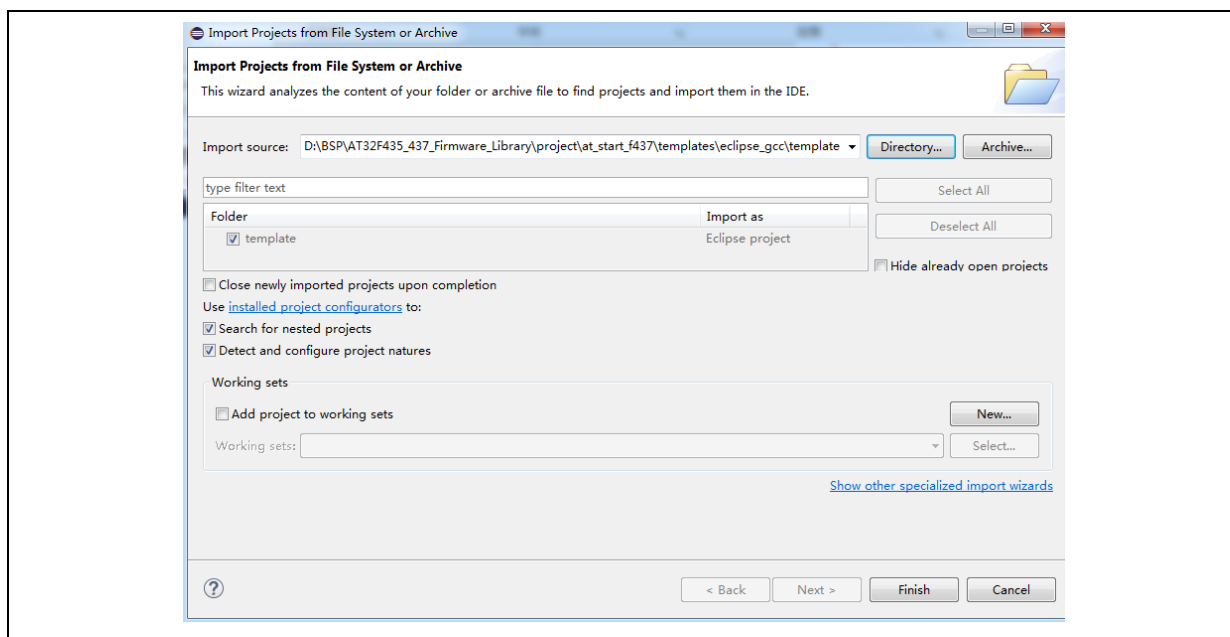
本章以AT32F437为例说明工程的配置与编译

### 3.1 打开 template 工程

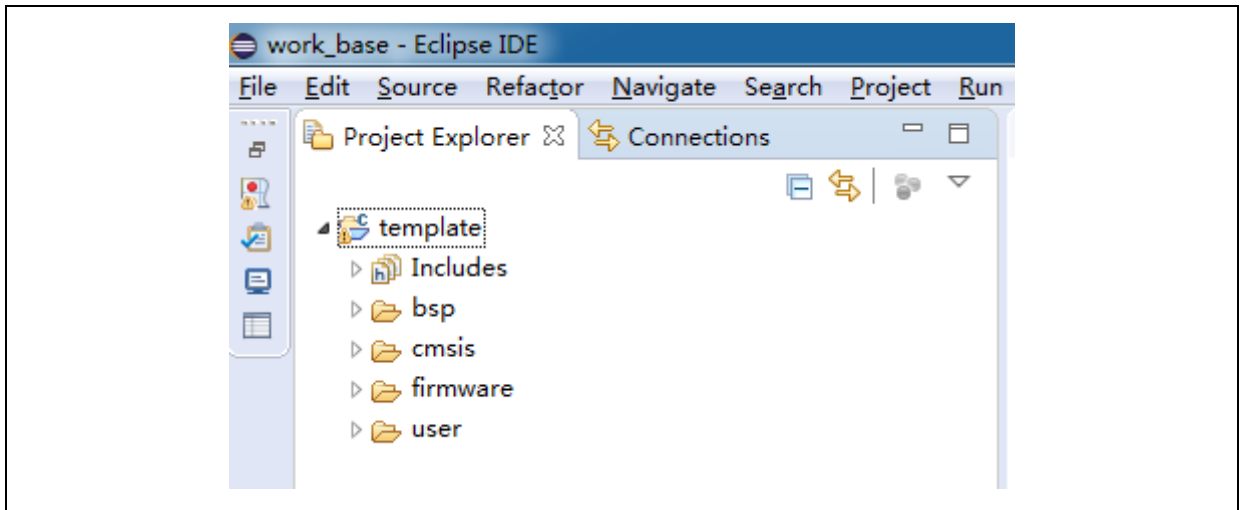
1. 选择File→Open Projects from File System..



2. 在Import source中选择路径，然后点击Finish。AT32F437xx template路径如下，其它系列路径类似xxx\AT32F435\_437\_Firmware\_Library\project\at\_start\_f437\templates\eclipse\_gcc\template



3. 工程打开之后看到一个template的项目工程如下

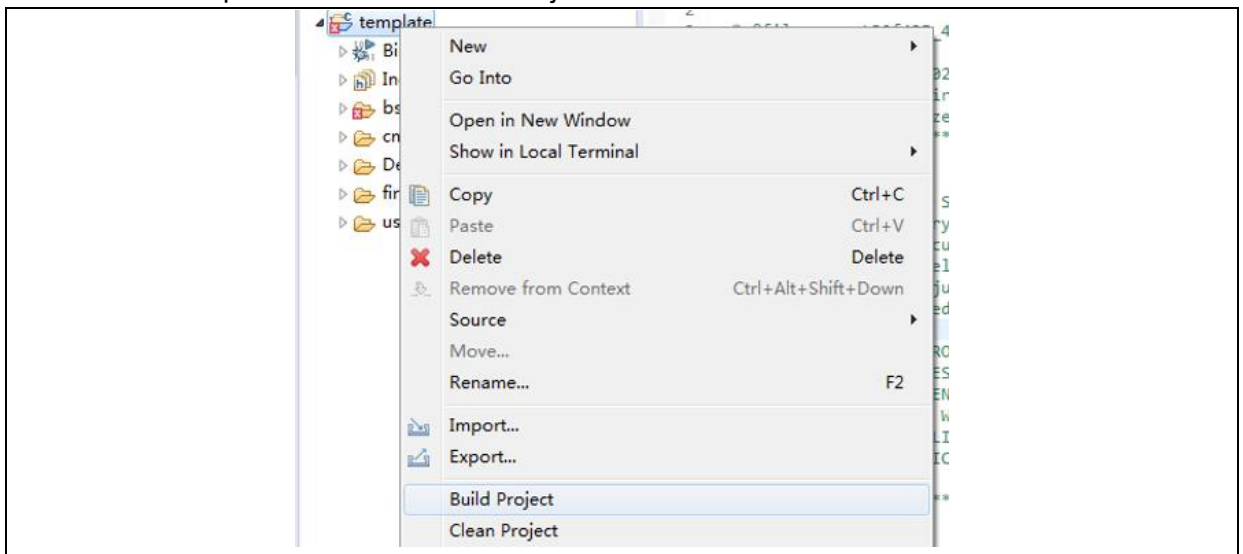


## 3.2 编译

在编译过程中，需要对头文件的路径，以及一些芯片宏的定义，此部分在template工程有对应都有配置。配置包括如下内容

- 芯片配置
- 头文件路径配置
- 宏定义配置
- 链接脚本文件配置（不同型号之前ld文件会涉及到要修改）

### 1. 右键选择 template 工程，选择 Build Project



2 编译结束之后会生成template.elf

## CDT Build Console [template]

```
arm-none-eabi-gcc -mcpu=cortex-m4 -mthumb -O0 -ffunction-sections -g -DAT_START_F437_V1 -DAT32F437ZMT7 -DUSE_STDPERIPH_DR
Finished building: D:\BSP\AT32F435_437_Firmware_Library\libraries\cmsis\cm4/device_support/system_at32f435_437.c
```

```
Building file: D:\BSP\AT32F435_437_Firmware_Library\project\at32f435_437_board\at32f435_437_board.c
```

```
Invoking: GNU ARM Cross C Compiler
```

```
arm-none-eabi-gcc -mcpu=cortex-m4 -mthumb -O0 -ffunction-sections -g -DAT_START_F437_V1 -DAT32F437ZMT7 -DUSE_STDPERIPH_DR
```

```
Finished building: D:\BSP\AT32F435_437_Firmware_Library\project\at32f435_437_board\at32f435_437_board.c
```

```
Building target: template.elf
```

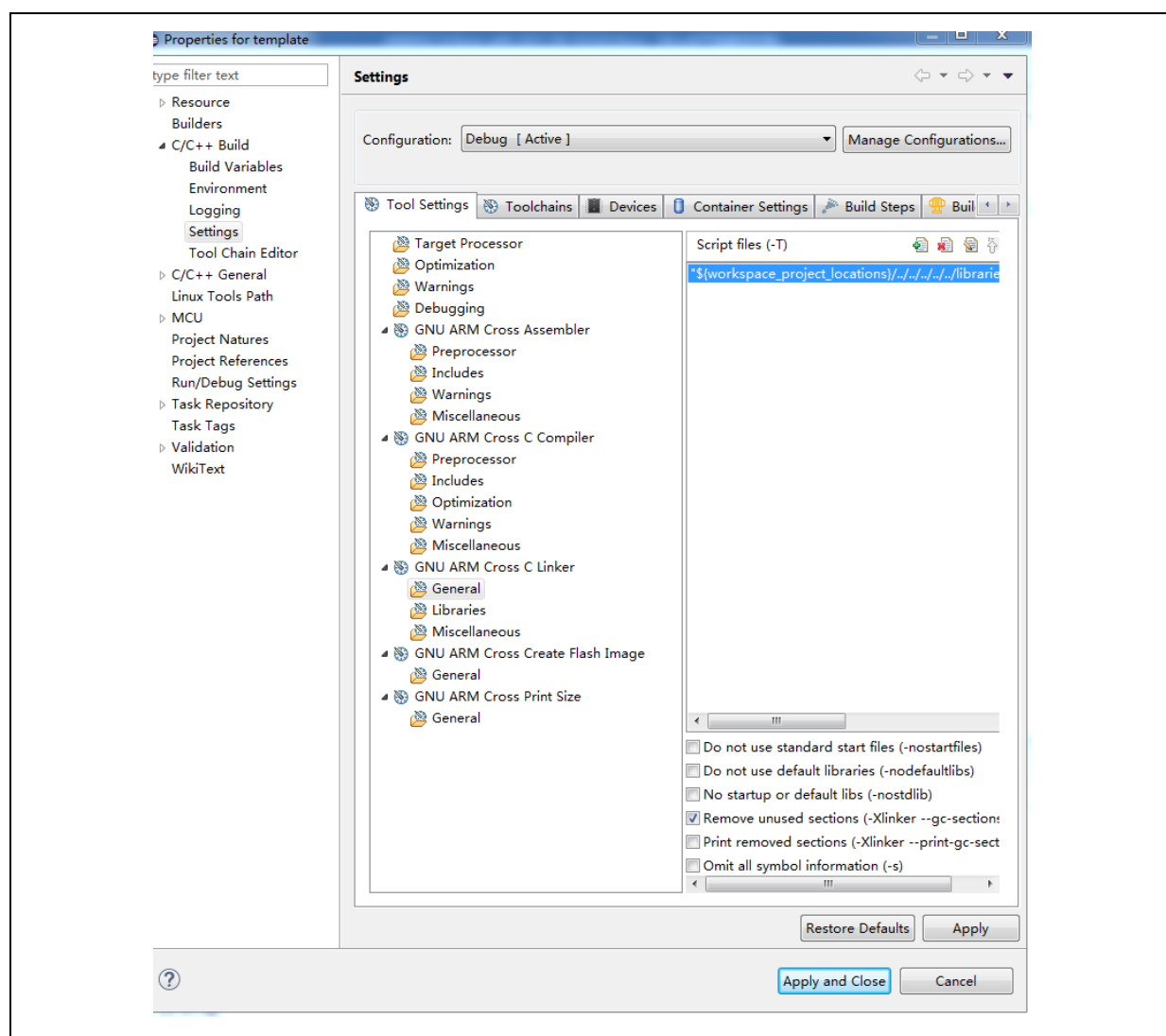
```
Invoking: GNU ARM Cross C Linker
```

```
arm-none-eabi-gcc -mcpu=cortex-m4 -mthumb -O0 -ffunction-sections -g -T "D:\BSP\AT32F435_437_Firmware_Library\project\at_
```

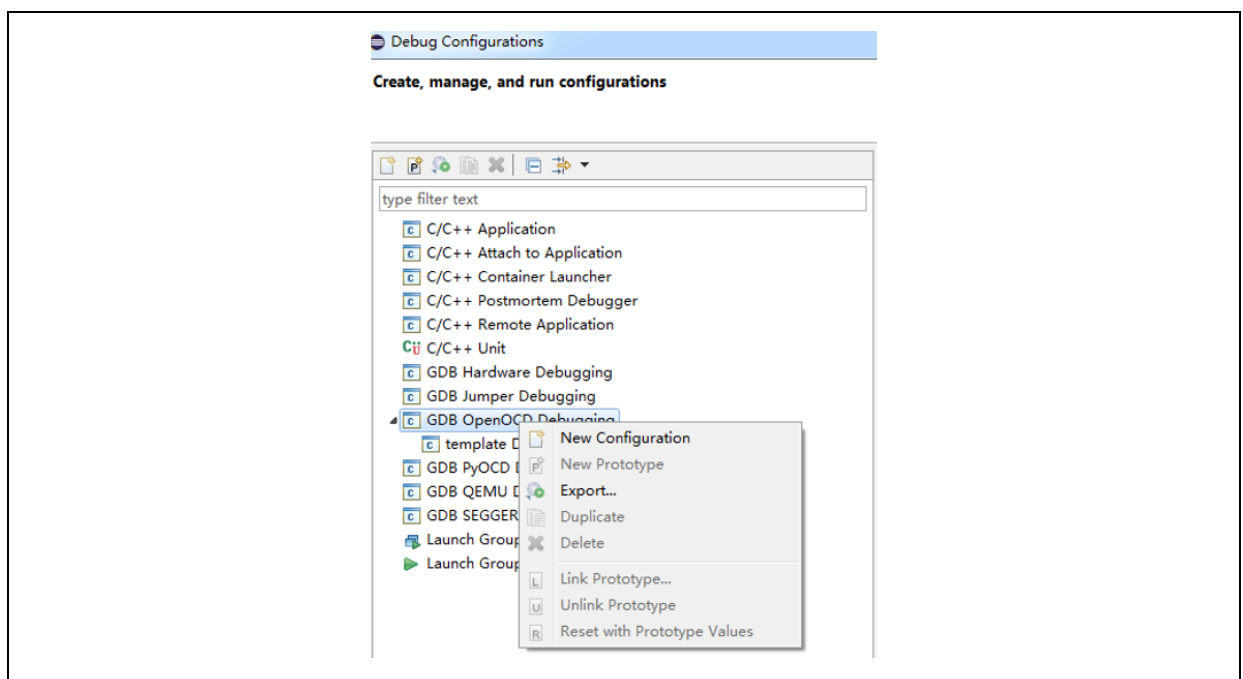
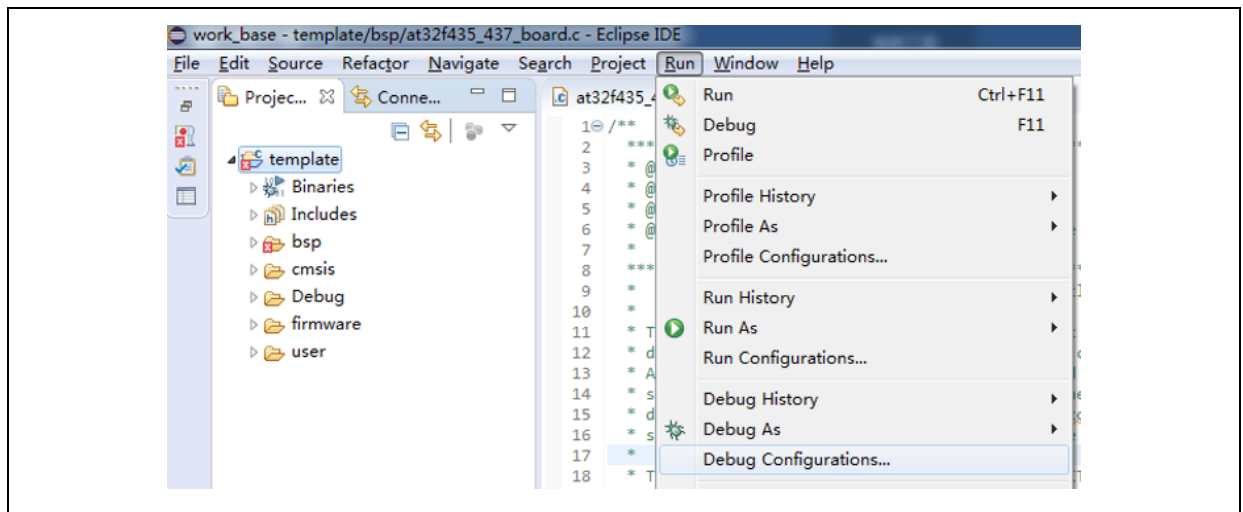
```
Finished building target: template.elf
```

```
15:09:24 Build Finished. 0 errors, 0 warnings. (took 11s.303ms)
```

3. 对于同一系列不同型号的配置，只需要修改ld文件即可，在下图的setting中修改。另外，其它如果要修改的如头文件路径等也是在下图setting中对应选项中修改即可。









## 4 调试

本章分别描述用jlink和atlink调试at32系列芯片。

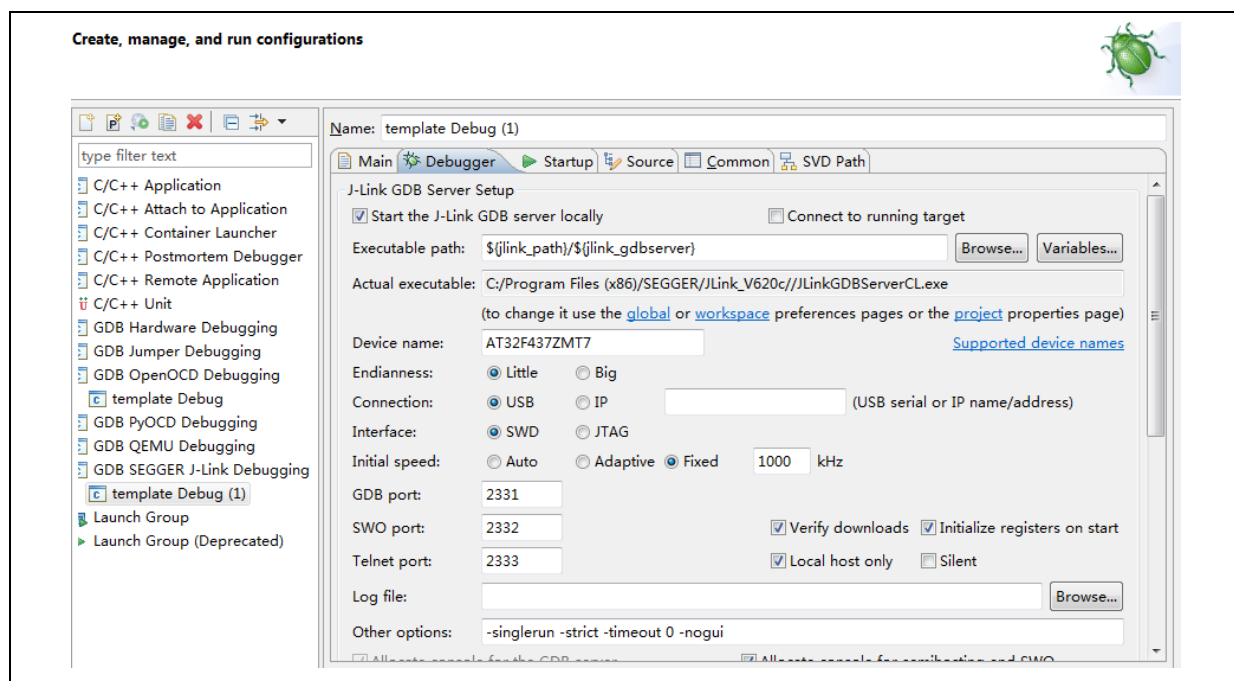
### 4.1 Jlink 调试

本部分主要说明调试的一些配置：

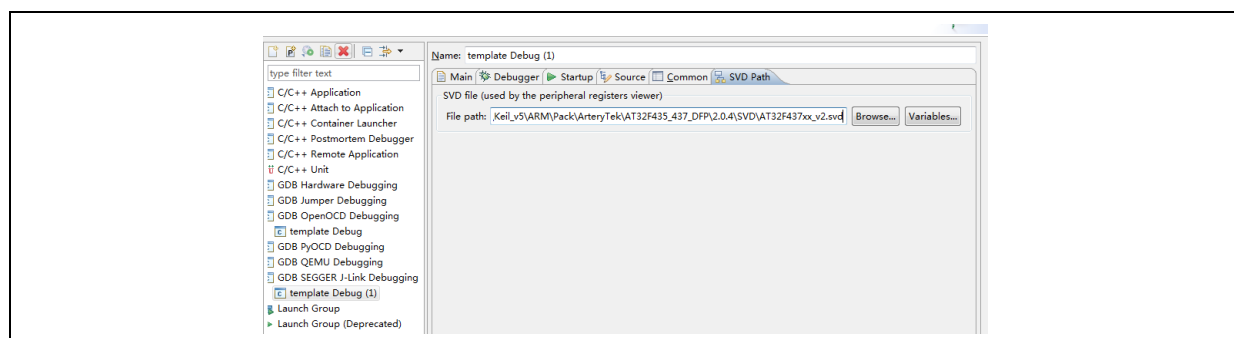
- Jlink配置
- GDB配置
- SVD 外设寄存器配置

#### 4.1.1 Debug 配置

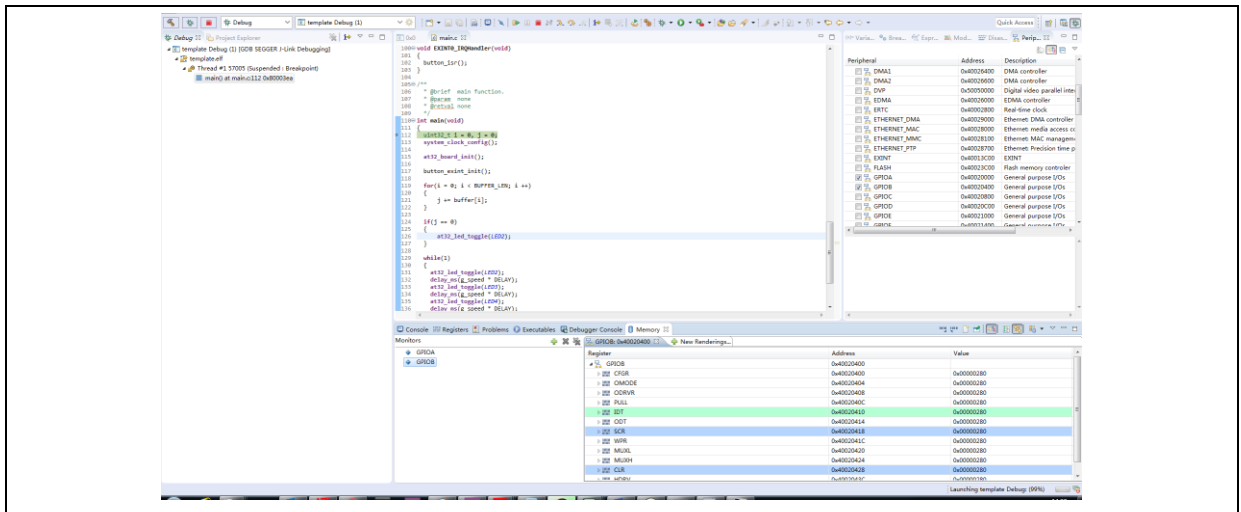
1. “Run” → “Debug Configurations” → “GDB SEGGER J-Link Debugging” → “New Configuration”  
建立一个新的Debug配置，配置JlinkGDBServerCL，Device name根据需要调试的具体芯片型号填写，例如AT32F437ZMT7，AT32F413RCT7，AT32F415RCT7等



2. 配置GDB，选择GCC 安装目录下的arm-none-eabi-gdb.exe
3. SVD Path选择，用于Debug寄存器的描述，这里可以直接使用keil下的svd文件，当安装了AT32的keil Packet之后，会自动将svd拷贝到keil 目录下。



#### 4. Debug 配置完成 → “Apply”→“Debug” 进入调试



## 4.2 ATLink 调试

本部分主要说明使用OpenOCD + Eclipse + ATLink来调试AT32。关于ATLink的使用说明请参考《AT-Link\_User\_Manual\_SC.pdf》

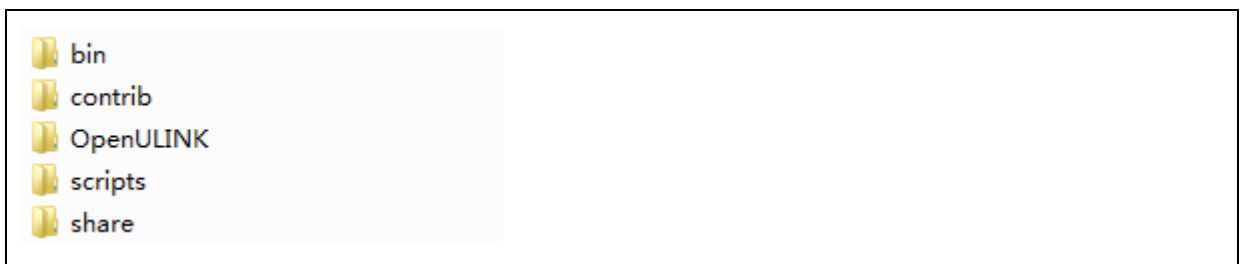
本节主要说明如下内容：

- Eclipse Openocd配置
- GDB配置
- SVD 外设寄存器配置

在解压OpenOCD包文件<<OpenOCD\_V2.x.x.zip>>

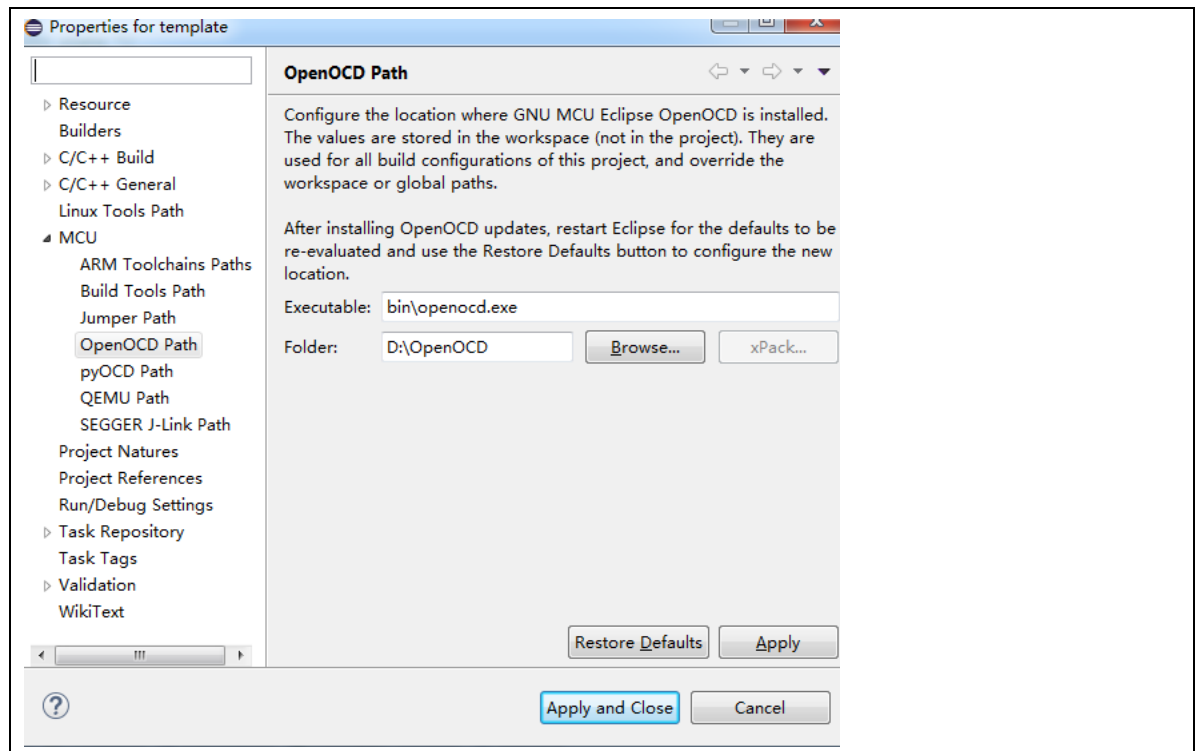
包含5个目录，bin文件下为可执行exe，scripts目录为配置文件目录

OpenOCD包含如下目录文件：



### 4.2.1 Debug 配置

1. 配置 OpenOCD 路径，Project→Properties→MCU→OpenOCD Path



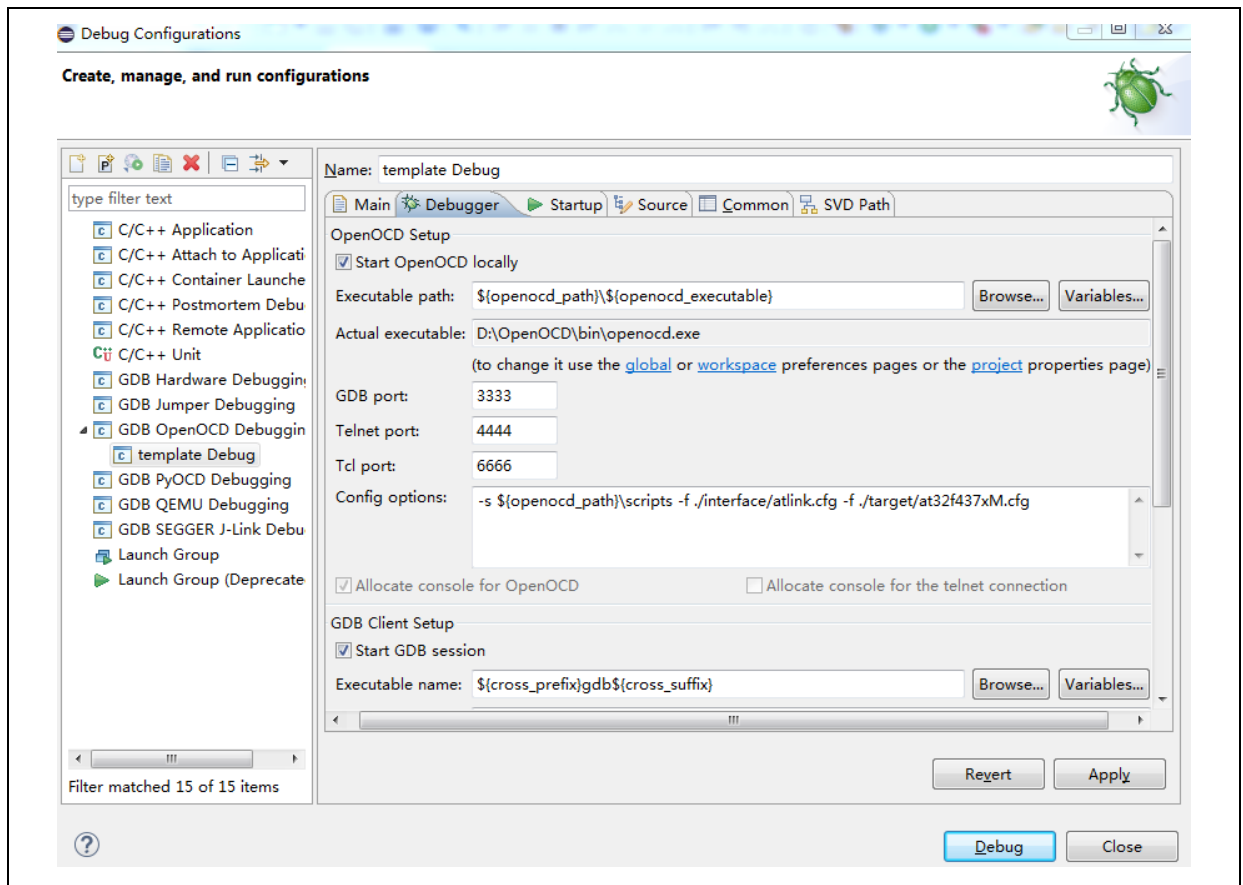
2. “Run” → ” Debug Configurations” →”GDB OpenOCD Debugging”→“New Configuration” 建立一个新的Debug配置

可配置项如下:

openocd的可执行文件路径: D:\OpenOCD\bin\openocd.exe

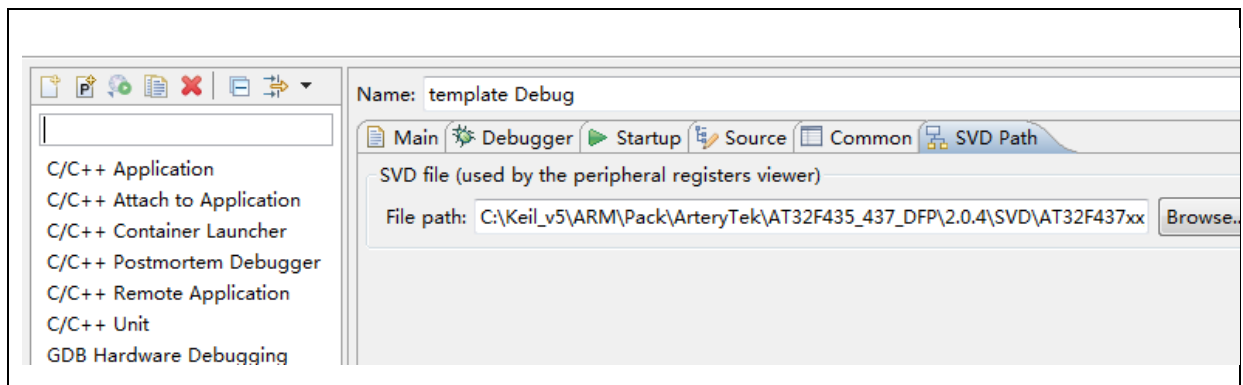
Config options: -s \${openocd\_path}\scripts -f ./interface/atlink.cfg -f ./target/at32f437xM.cfg

atlink.cfg表示使用atlink调试工具, at32f437xM.cfg表示at32f437 FLASH有4032KB的型号, 其它AT32F437的型号可统一使用at32f437xx.cfg。对于不同的系列, 如AT32F403A,AT32F415等此处对应的target/xxx.cfg需要做对应的修改。

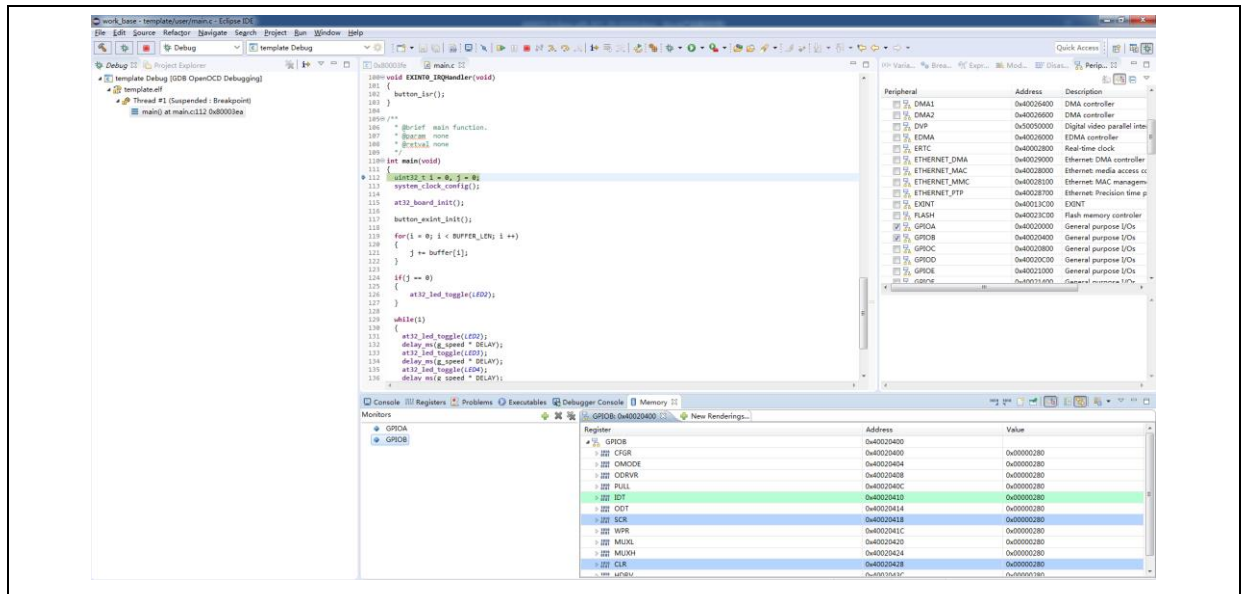


### 3. 配置SVD文件

可下载对应型号的SVD文件用于Debug



### 4. Debug 配置完成 → “Apply”→”Debug” 进入调试



## 5 版本历史

表 1. 文档版本历史

日期	版本	变更
2021.12.13	2.0.0	最初版本



**重要通知 - 请仔细阅读**

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途(及其依据任何司法管辖区的法律的对应情况)，或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力的产品不得应用于武器。此外，雅特力产品也不是为下列用途而设计并不得应用于下列用途：(A) 对安全性有特别要求的应用，例如：生命支持、主动植入设备或对产品功能安全有要求的系统；(B) 航空应用；(C) 汽车应用或汽车环境，且/或(D) 航天应用或航天环境。如果雅特力产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，采购商仍将独自承担因此而导致的任何风险，雅特力的产品设计规格明确指定的汽车、汽车安全或医疗工业领域专用产品除外。根据相关政府主管部门的规定，ESCC、QML 或 JAN 正式认证产品适用于航天应用。

经销的雅特力产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2021 雅特力科技 (重庆) 有限公司 保留所有权利