

AT32F415固件库BSP&Pack应用指南

前言

这篇应用指南对如何使用BSP以及如何安装AT32 Pack进行了简单的描述，对用户起到引导性的作用。

目录

1	简介	6
2	BSP 使用简述	7
2.1	BSP 结构	7
2.2	BSP 使用方法	7
2.3	模板工程介绍	8
2.4	注意事项	9
2.4.1	型号切换	9
3	Pack 安装步骤	14
4	Flash 算法文件说明	24
5	版本历史	30

表目录

表 1. 型号宏定义对应表.....	7
表 2. 文档版本历史	30

图目录

图 1. BSP 内容结构	7
图 2. 外设使能宏定义	8
图 3. templates 文件内容	9
图 4. Keil_v5 模板工程示例	9
图 5. Keil 改 device	10
图 6. Keil 改宏定义	11
图 7. IAR 改 device	12
图 8. IAR 改宏定义	13
图 9. Pack 安装包	14
图 10. IAR Pack 安装界面	14
图 11. IAR Pack 安装流程	15
图 12. 查看 IAR Pack 安装情况	16
图 13. 查看 Keil_v5 Pack 安装情况	17
图 14. Keil_v4 Pack 安装界面	18
图 15. Keil_v4 Pack 安装流程	18
图 16. Keil_v4 Pack 安装完成	19
图 17. 查看 Keil_v4 Pack 安装情况	20
图 18. Segger 包安装界面	21
图 19. Segger 包安装流程	21
图 20. 打开 J-Flash	22
图 21. J-Flash 创建新工程	22
图 22. 查看 Device 信息	23
图 23. Keil 算法文件设置	24
图 24. Keil 算法文件配置栏	25
图 25. Keil 选择算法文件	25
图 26. Keil 新增算法文件	26
图 27. IAR 工程名	27
图 28. IAR 算法文件配置	27
图 29. IAR Flash Loader 新增	28
图 30. IAR Flash Loader 配置	28

图 31. IAR Flash Loader 配置成功.....	29
----------------------------------	----

1 简介

为了让用户高效快速的使用Artery MCU，雅特力官方提供了一套完整的BSP&Pack用于开发。主要包括：外设驱动库、内核相关文件、完整的应用例程以及能够支持Keil_v5、Keil_v4、IAR_v6和IAR_v7、IAR_v8等多种开发环境的Pack文件。

本应用指南会介绍BSP&Pack具体的使用方法。

2 BSP 使用简述

2.1 BSP 结构

BSP(Board Support Package)中内容结构大致如下图所示：

图 1. BSP 内容结构

document	21/05/18 10:32	文件夹
libraries	21/05/18 10:32	文件夹
middlewares	21/05/18 10:32	文件夹
project	21/05/18 10:32	文件夹
utilities	21/05/14 11:35	文件夹

document:

- 部分说明文档。

libraries:

- **drivers:** 外设驱动库。
- **cmsis:** 内核相关文件。包括 cortex-m4 库文件、系统初始化文件、启动文件等。

middlewares:

- 第三方软件包或公用协议包。如 USB 协议层驱动、网络协议层驱动、操作系统源码等。

project:

- **examples:** 型号相关的示例 demo。
- **templates:** 模板工程。

utilities:

- 各经典应用案例存放目录。

2.2 BSP 使用方法

1. 在创建工程时，需要导入启动代码（startup_at32f415.s）到工程，Code 编译之前，还需要根据 MCU 型号，开启对应的宏定义，MCU 型号与宏定义的对对应关系如下表

表 1. 型号宏定义对应表

MCU型号	宏定义	PINs	Flash大小(KB)
AT32F415RCT7	AT32F415RCT7	64	256
AT32F415RBT7	AT32F415RBT7	64	128
AT32F415R8T7	AT32F415R8T7	64	64
AT32F415RCT7-7	AT32F415RCT7_7	64	256
AT32F415RBT7-7	AT32F415RBT7_7	64	128
AT32F415R8T7-7	AT32F415R8T7_7	64	64
AT32F415CCT7	AT32F415CCT7	48	256
AT32F415CBT7	AT32F415CBT7	48	128
AT32F415C8T7	AT32F415C8T7	48	64
AT32F415CCU7	AT32F415CCU7	48	256
AT32F415CBU7	AT32F415CBU7	48	128
AT32F415KCU7-4	AT32F415KCU7_4	32	256
AT32F415KBU7-4	AT32F415KBU7_4	32	128

AT32F415K8U7-4	AT32F415K8U7_4	32	64
----------------	----------------	----	----

- 系列芯片头文件中（at32f415.h），USE_STDPERIPH_DRIVER 宏定义用于区别是否使用 Keil RTE 功能，在未使用 Keil RTE 功能时开启这个宏定义可规避 Keil-MDK 的某些版本误开启 RTE_ 的错误问题。
- 配置头文件中（at32f415_conf.h），定义了外设模块开启的宏定义，可用于控制外设模块的使用，关闭时只需屏蔽掉外设对应的 MODULE_ENABLED 宏定义即可，如下图所示：

图 2. 外设使能宏定义

```
#define CRM_MODULE_ENABLED
#define CMP_MODULE_ENABLED
#define TMR_MODULE_ENABLED
#define ERTC_MODULE_ENABLED
#define GPIO_MODULE_ENABLED
#define I2C_MODULE_ENABLED
#define USART_MODULE_ENABLED
#define PWC_MODULE_ENABLED
#define CAN_MODULE_ENABLED
#define ADC_MODULE_ENABLED
#define SPI_MODULE_ENABLED
#define DMA_MODULE_ENABLED
#define DEBUG_MODULE_ENABLED
#define FLASH_MODULE_ENABLED
#define CRC_MODULE_ENABLED
#define WWDT_MODULE_ENABLED
#define WDT_MODULE_ENABLED
#define EXINT_MODULE_ENABLED
#define SDIO_MODULE_ENABLED
#define USB_MODULE_ENABLED
#define MISC_MODULE_ENABLED
```

- 系统时钟配置文件（at32f415_clock.c/.h），配置了默认的系统时钟频率及时钟路径。用户如有自定义需求时可自行修改倍频流程及系数，后续也可结合 ArteryTek 提供的时钟配置上位机来生成相应的时钟配置文件。

2.3 模板工程介绍

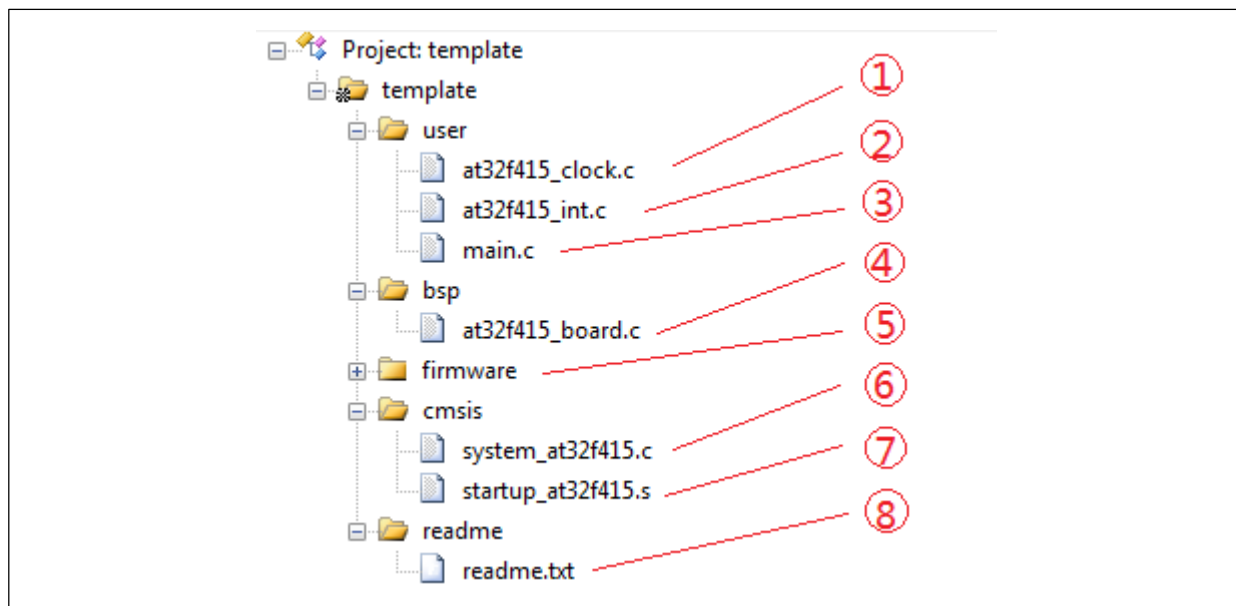
在 ArteryTek 提供的固件库 BSP 中都默认建立好了 Keil 和 IAR 常用版本下的模板工程。以 AT32F415 系列为例，其存放目录在 AT32F415_Firmware_Library_V2.x.x/project/at_start_xxx/templates 中，内容如下：

图 3. templates 文件内容

iar_v6.10	21/05/24 16:03	文件夹
iar_v7.4	21/05/24 16:03	文件夹
iar_v8.2	21/05/24 16:03	文件夹
inc	21/05/24 16:03	文件夹
mdk_v4	21/05/24 16:03	文件夹
mdk_v5	21/05/24 16:03	文件夹
src	21/05/24 16:03	文件夹
readme.txt	21/05/21 11:15	TXT 文件

在此创建了 Keil_v5、Keil_v4、IAR_6.10、IAR_7.4 和 IAR_8.2 版本的模板工程。inc 和 src 文件夹分别保存了模板工程中所用到的应用部分的头文件及源码文件。打开对应工程的文件夹并点击工程文件即可打开对应的 IDE 工程。如下是 Keil_v5 工程示例（具体内容及版本以实际固件包内容为准）：

图 4. Keil_v5 模板工程示例



工程内添加的内容描述如下（以 AT32F415 系列举例，其他系列与此类似）：

- ① at32f415_clock.c 时钟配置文件，设置了默认的时钟频率及时钟路径。
- ② at32f415_int.c 中断文件，默认编写了部分内核中断函数的代码流程。
- ③ main.c 模板工程的主代码文件。
- ④ at32f415_board.c 板级配置文件，设置了 AT-START 上的按键和 LED 等常用硬件配置。
- ⑤ firmware 下的 at32f415_xx.c 是各片上外设的驱动文件。
- ⑥ system_at32f415.c 系统初始化文件。
- ⑦ startup_at32f415.s 启动文件。
- ⑧ readme.txt 工程的说明文件，记录了模板工程的一些应用功能及设置方式等信息。

2.4 注意事项

2.4.1 型号切换

如若需要在已有的工程或 demo 中进行型号切换时需注意型号宏定义及 device 名的同步修改，在修改前请详细查看文中表 1 内容，其中详细罗列了 MCU 型号与宏定义的对应关系。

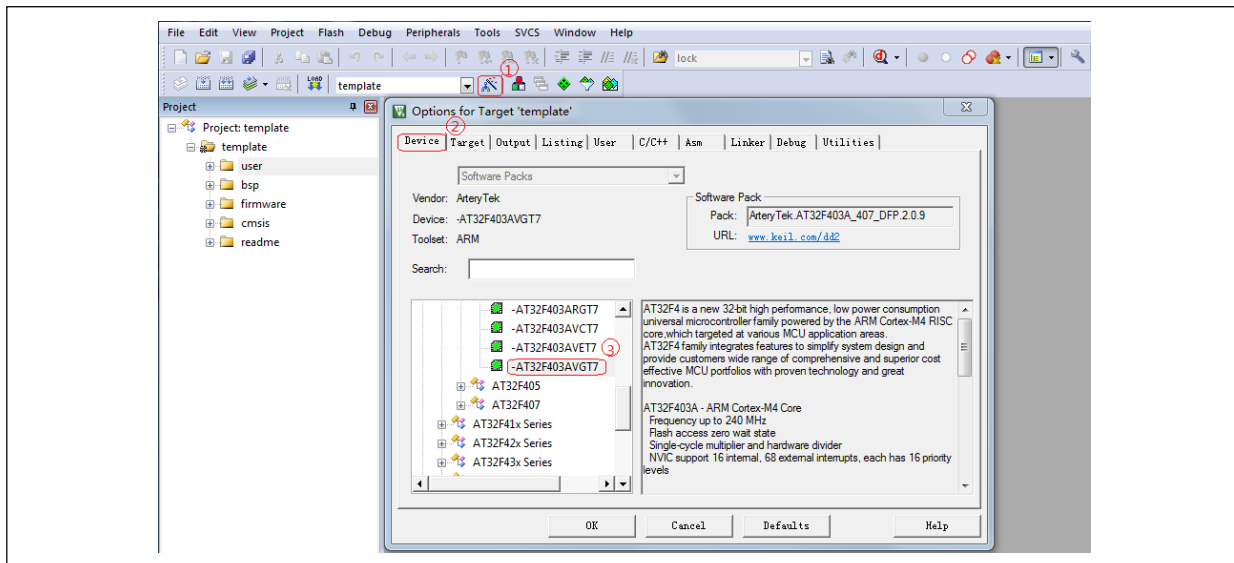
接下来将对两种常用的开发环境的修改方法来进行介绍（以下内容以 at32f403avg7 作为示例与图示，其余系列或型号的修改方法与此类似），修改方法主要有两步：1、改 device，2、改型号宏定义。

1. KEIL

修改 device，操作步骤和图示如下：

- ① 点击魔术棒“Options for Target”。
- ② 点击 Device 选项卡。
- ③ 选择需要切换的 Device 型号。

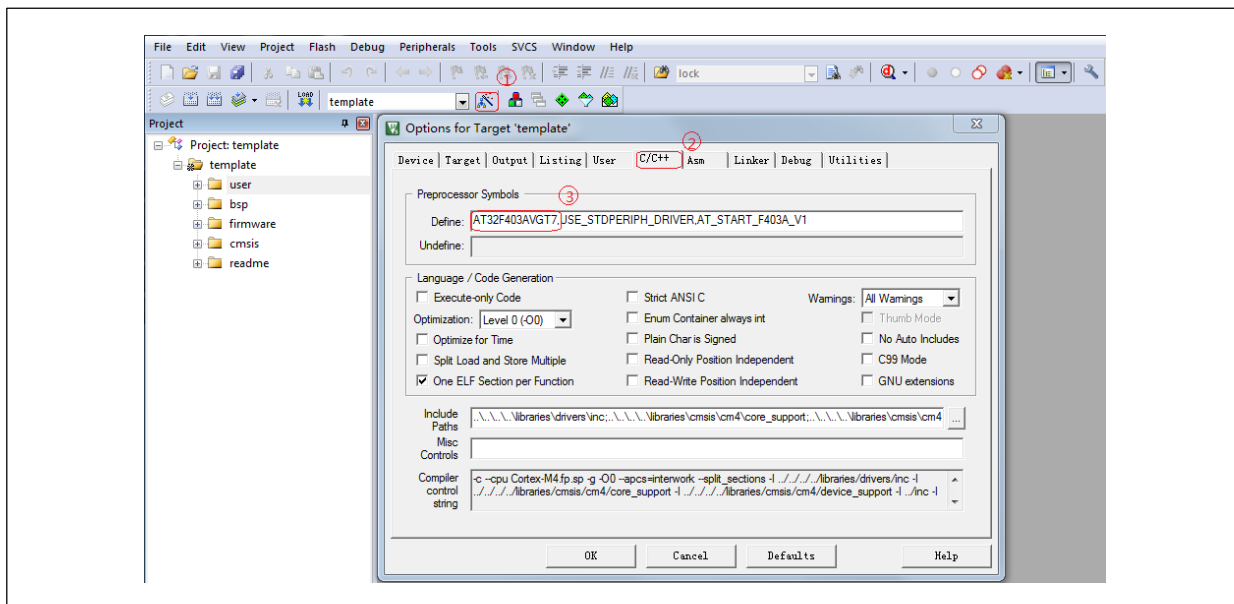
图 5. Keil 改 device



修改型号宏定义，操作步骤和图示如下：

- ① 点击魔术棒“Options for Target”。
- ② 点击 C/C++选项卡。
- ③ 将 Define 栏中将原有的型号宏定义删除，根据表 1 内容写入需要切换型号的宏定义。

图 6. Keil 改宏定义

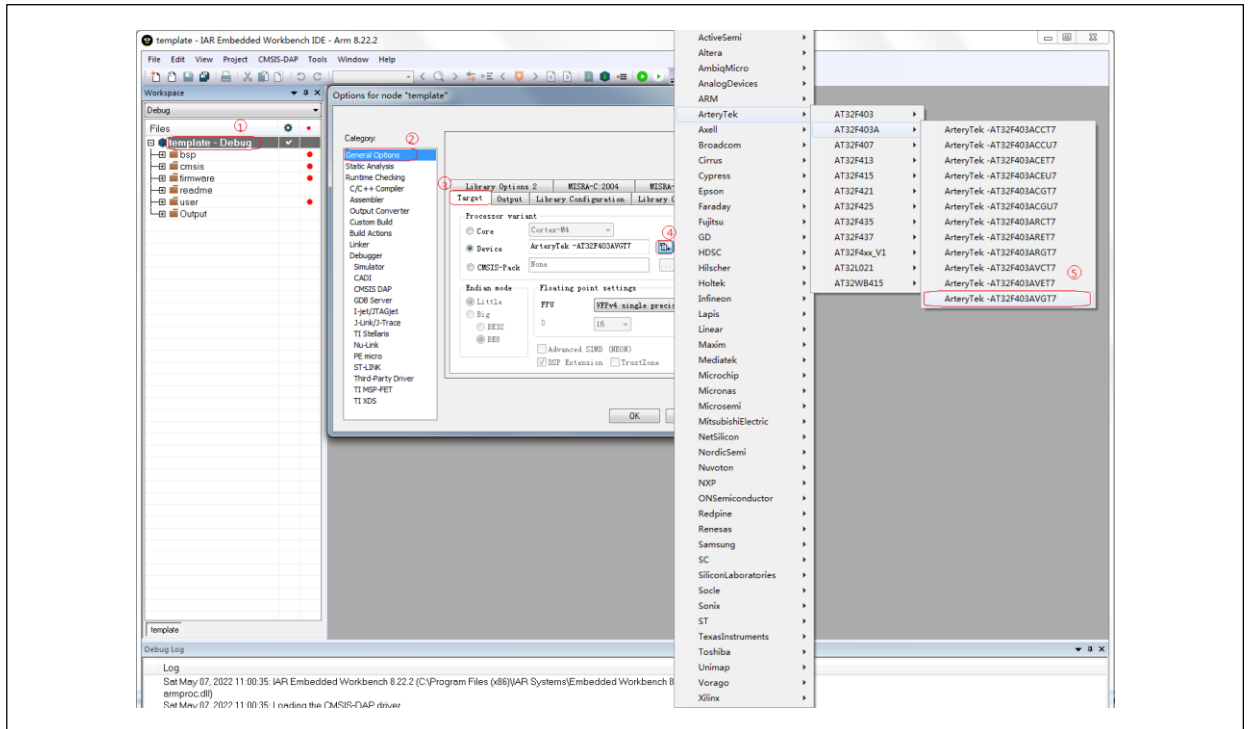


2. IAR

修改 device，操作步骤和图示如下：

- ① 鼠标右键点击工程名，并选择 **Options...**。
- ② 选择 **General Options**。
- ③ 选择 **Target**。
- ④ 点选复选框。
- ⑤ 选择需要切换的 **Device** 型号。

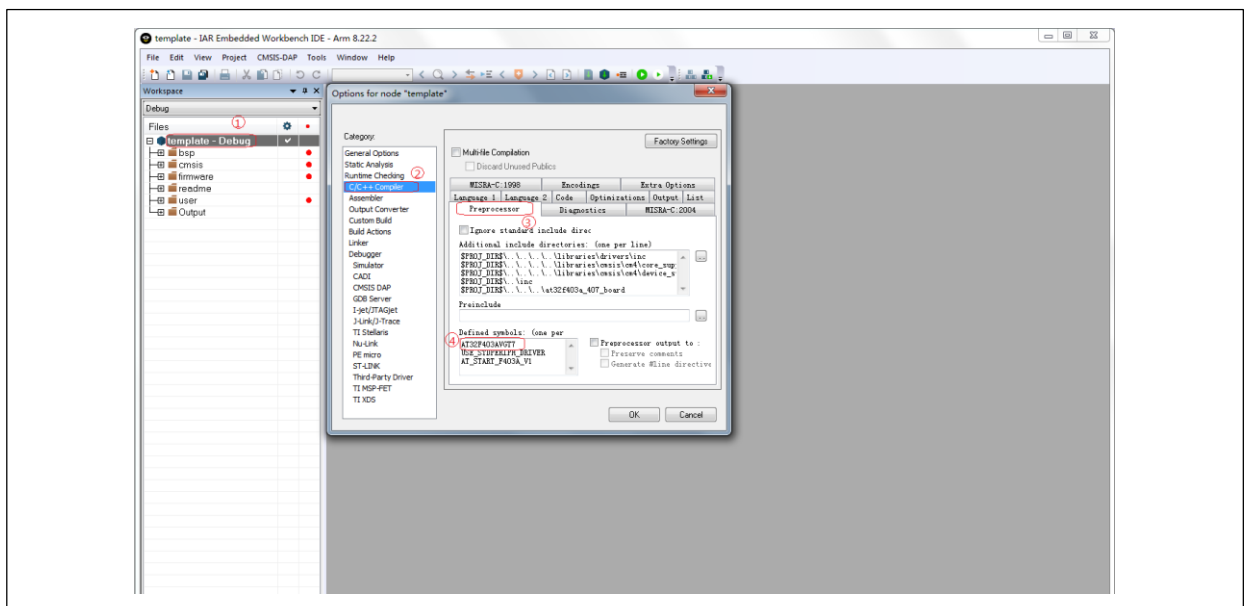
图 7. IAR 改 device



修改型号宏定义，操作步骤和图示如下：

- ① 鼠标右键点击工程名，并选择 Options...
- ② 选择 C/C++ Compiler。
- ③ 点击 Preprocessor 选项卡。
- ④ 将 Defined symbols 栏中将原有的型号宏定义删除，根据表 1 内容写入需要切换型号的宏定义。

图 8. IAR 改宏定义

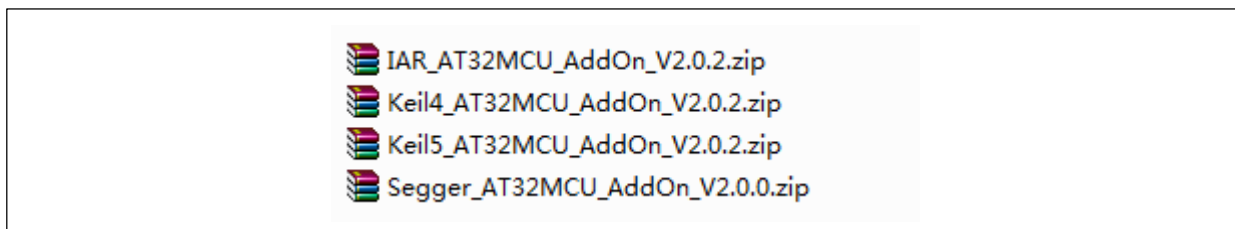


3 Pack 安装步骤

ArteryTek提供了支持Keil_v5、Keil_v4、IAR_v6、IAR_v7和IAR_v8等多种开发环境的Pack文件，对应的Pack采用‘双击’完成一键式安装。

Pack安装文件如下图（具体版本信息按实际情况为准）。

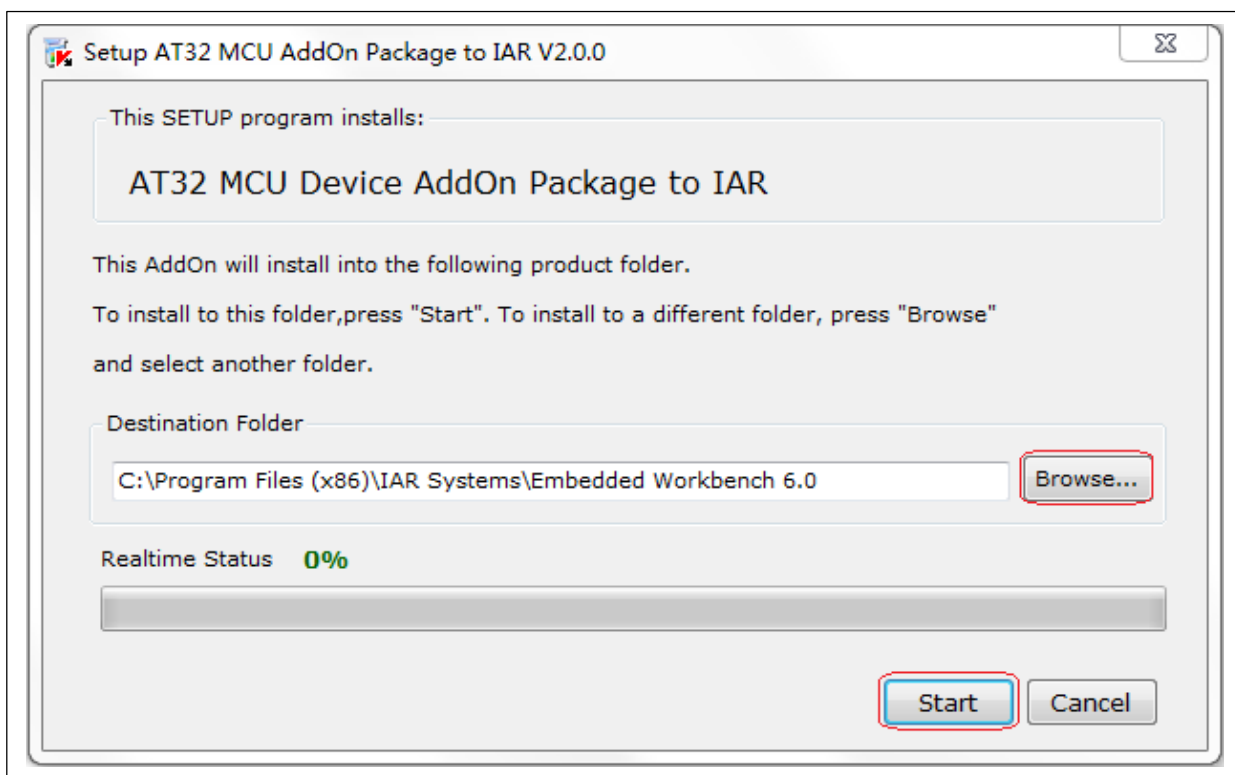
图 9. Pack 安装包



其中：

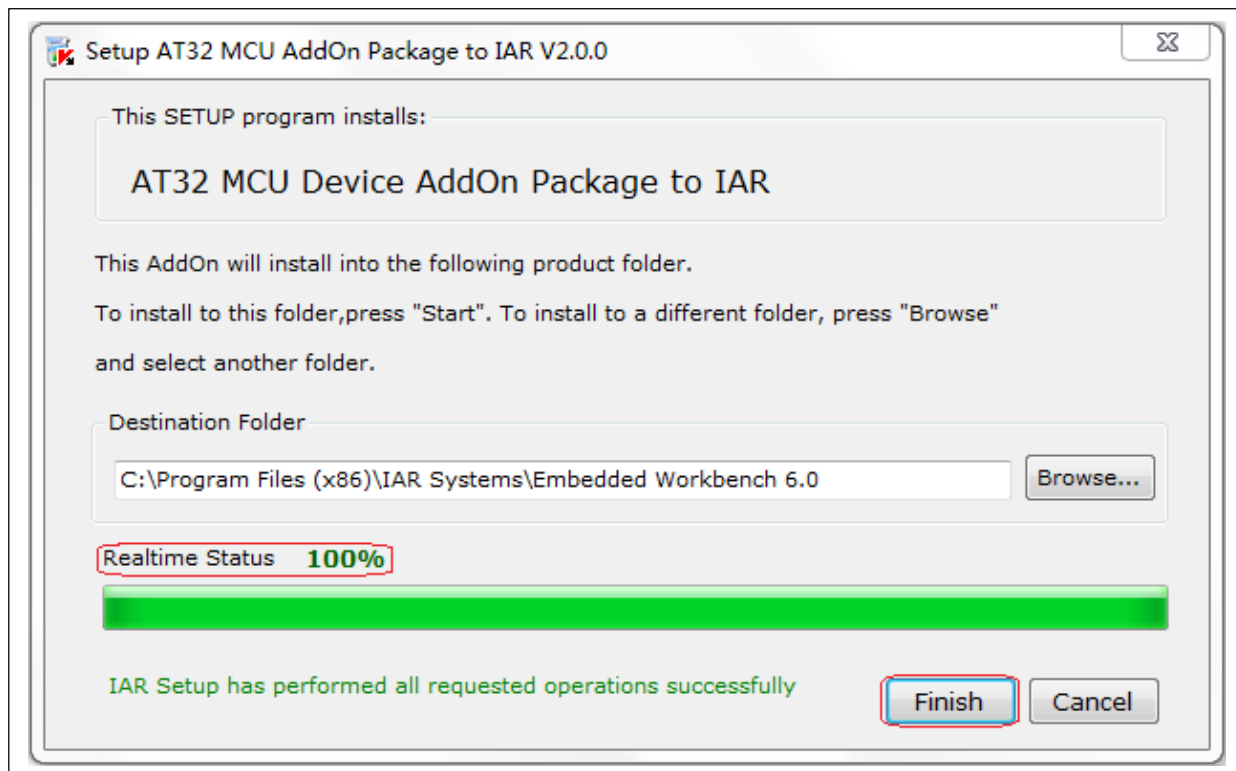
1. **IAR_AT32MCU_AddOn.zip**：支援 IAR_V6、IAR_V7 和 IAR_V8 的压缩包，安装步骤如下：
 - a) 解压 IAR_AT32MCU_AddOn.zip。
 - b) 双击 IAR_AT32MCU_AddOn.exe，弹出如下界面（具体版本信息按实际情况为准）。

图 10. IAR Pack 安装界面



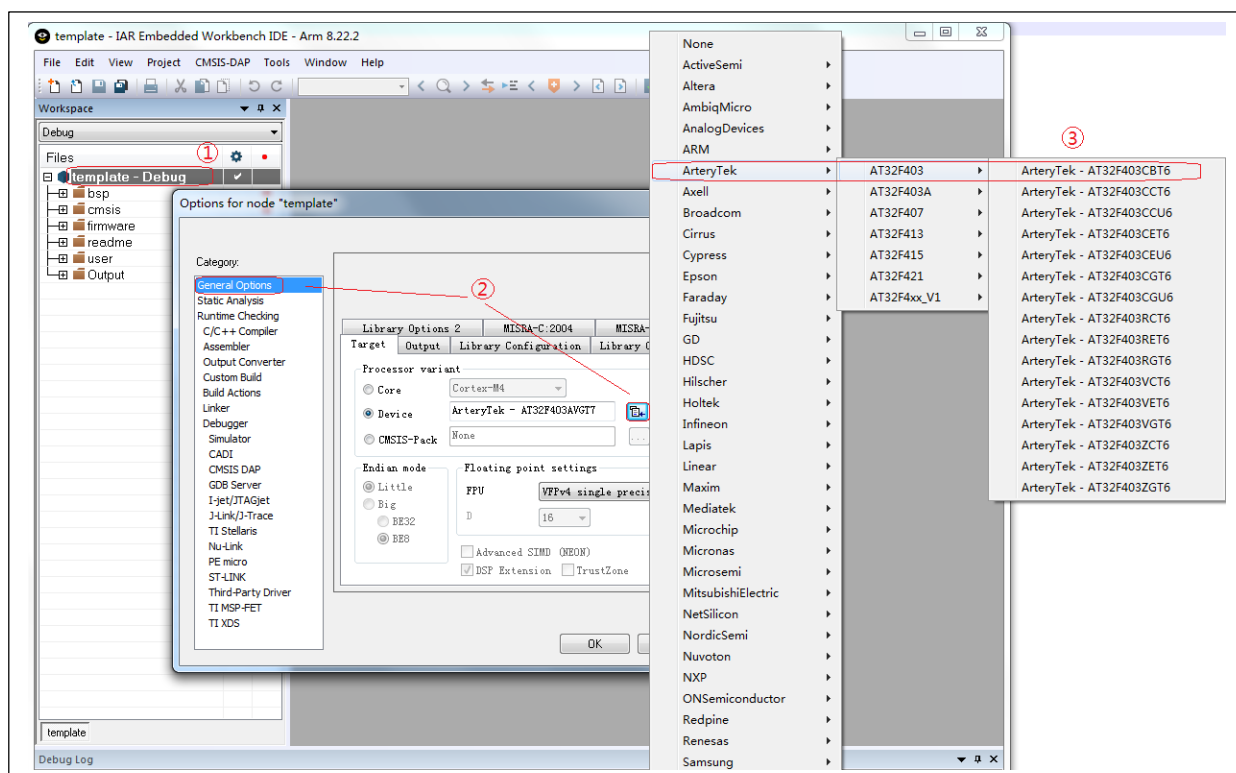
- c) 如果 IAR 的实际安装路径与“Destination Folder”对话框内的路径不一致，点击“Browse”选择实际安装路径。然后点击“Start”启动安装过程，如下图。

图 11. IAR Pack 安装流程



- d) 点击“Finish”完成安装。
- e) 查看 IAR Pack 是否安装成功。任意打开一个 IAR 工程，按如下步骤操作和查看：
- ① 鼠标右键点击工程名，并选择 Options...
 - ② 选择 General Options，并点击选复选框。
 - ③ 查看 ArteryTek 以及 ArteryTek – AT32F...相关的型号信息。

图 12. 查看 IAR Pack 安装情况

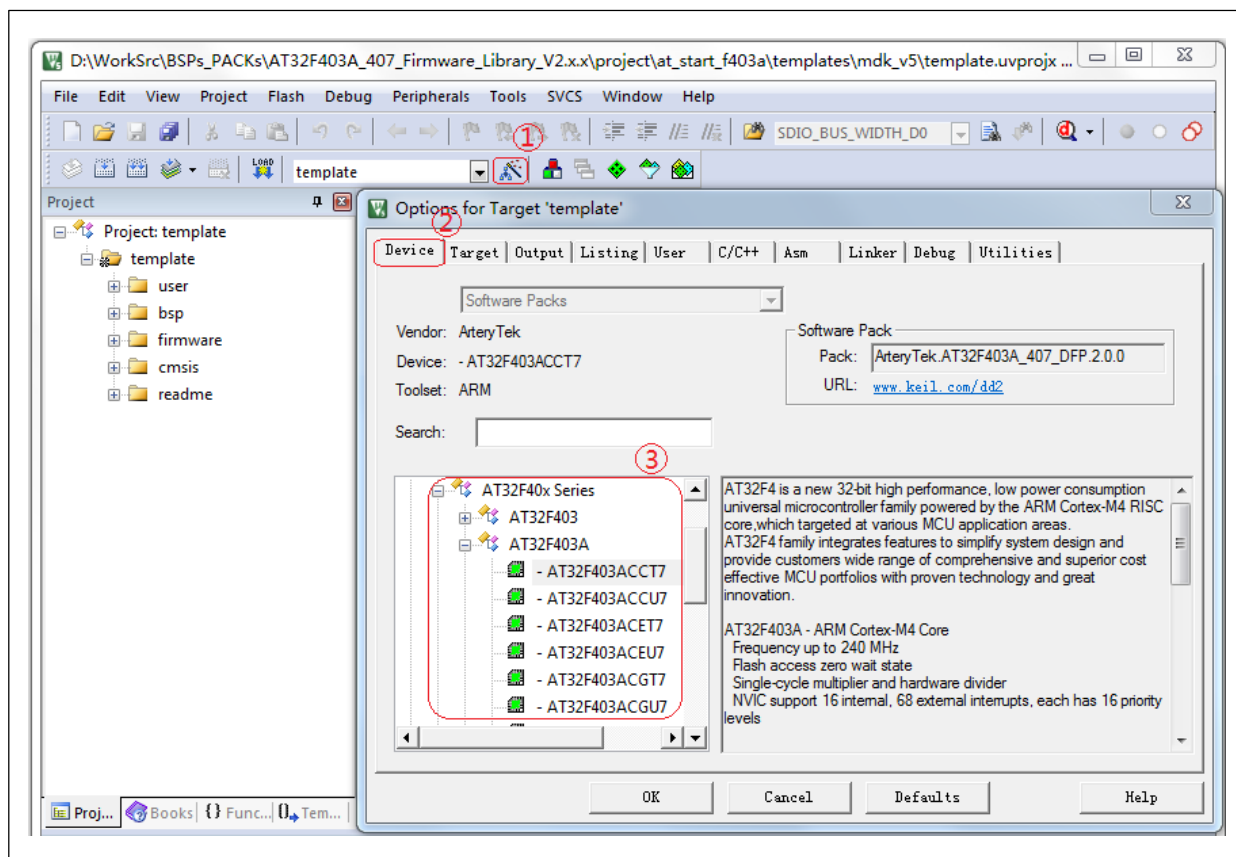


2. Keil5_AT32MCU_AddOn.zip: 支援 Keil_v5 的 pack 压缩包，具体版本见包内实际内容，安装

步骤如下：

- a) 解压 Keil5_AT32MCU_AddOn.zip，里面包含了所有目前支持的 Keil5 pack 安装包，都是标准的 Keil_v5 DFP 安装文件。
- b) 选择所需系列的安装包，双击 ArteryTek.AT32xxxx_DFP.2.x.x.pack 完成一键式安装。
- c) 查看 Keil_v5 Pack 是否安装成功。按如下步骤操作和查看：
 - ① 点击魔术棒。
 - ② 选择 Device 选项卡。
 - ③ 出现 ArteryTek 及相关型号信息。

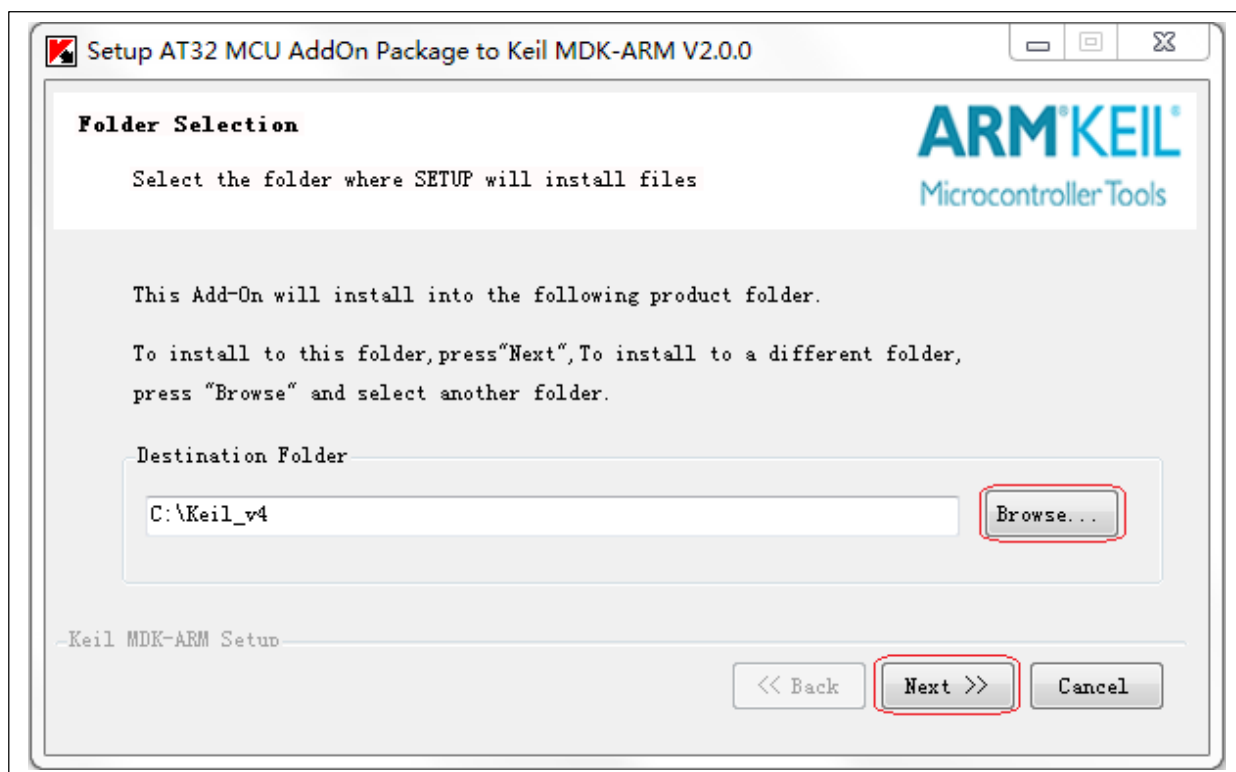
图 13. 查看 Keil_v5 Pack 安装情况



3. Keil4_AT32MCU_AddOn.zip: 支援 Keil_v4 的压缩包，安装步骤如下：

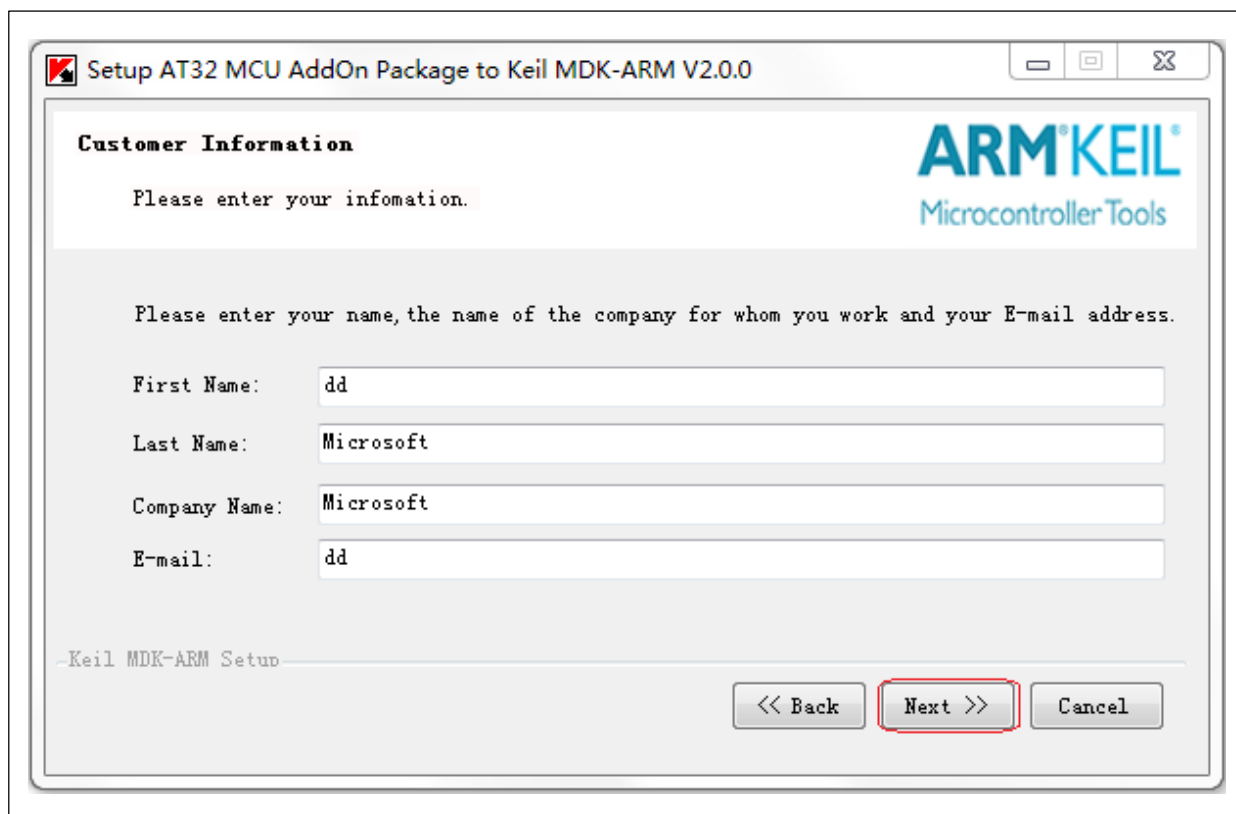
- a) 解压 Keil4_AT32MCU_AddOn.zip。
- b) 双击 Keil4_AT32MCU_AddOn.exe，弹出如下界面（具体版本信息按实际情况为准）。

图 14. Keil_v4 Pack 安装界面



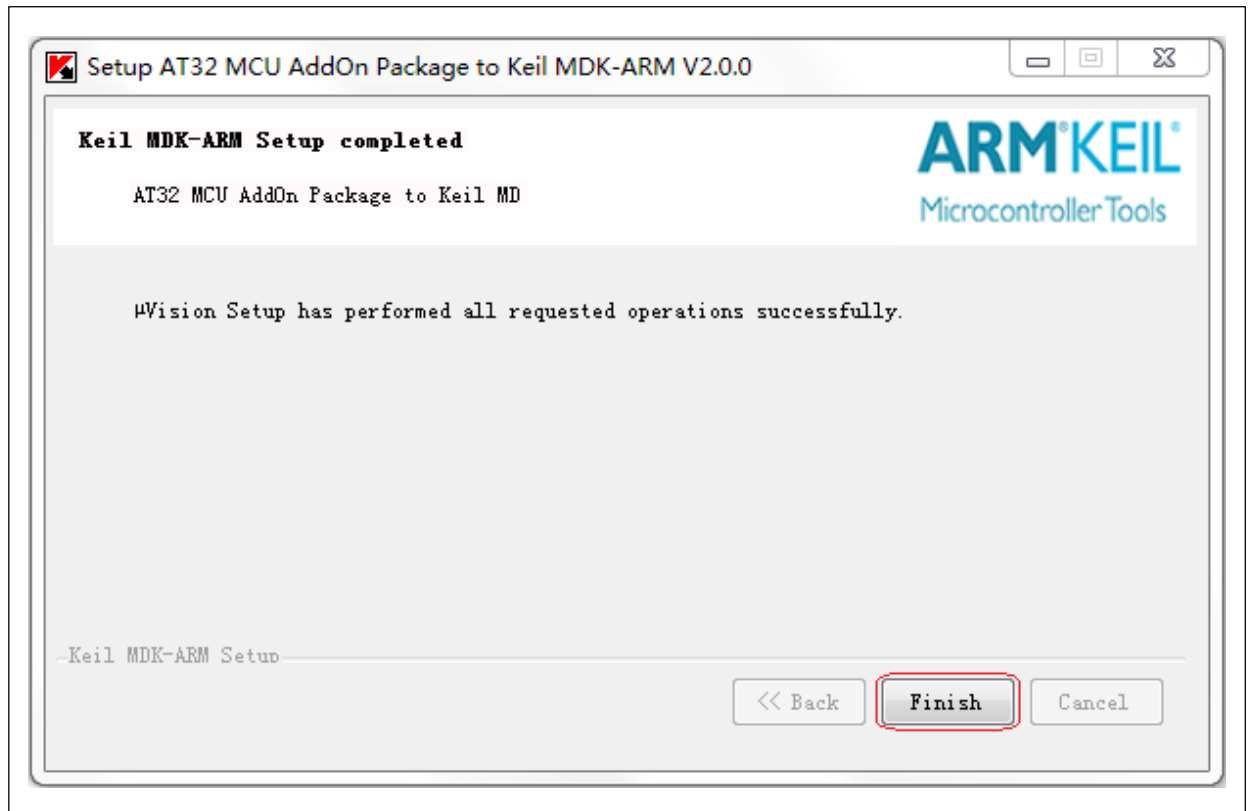
- c) 如果 Keil_v4 的实际安装路径与 “Destination Folder”对话框内的路径不一致，点击“Browse”选择实际安装路径。然后点击“Next”，弹出如下界面。

图 15. Keil_v4 Pack 安装流程



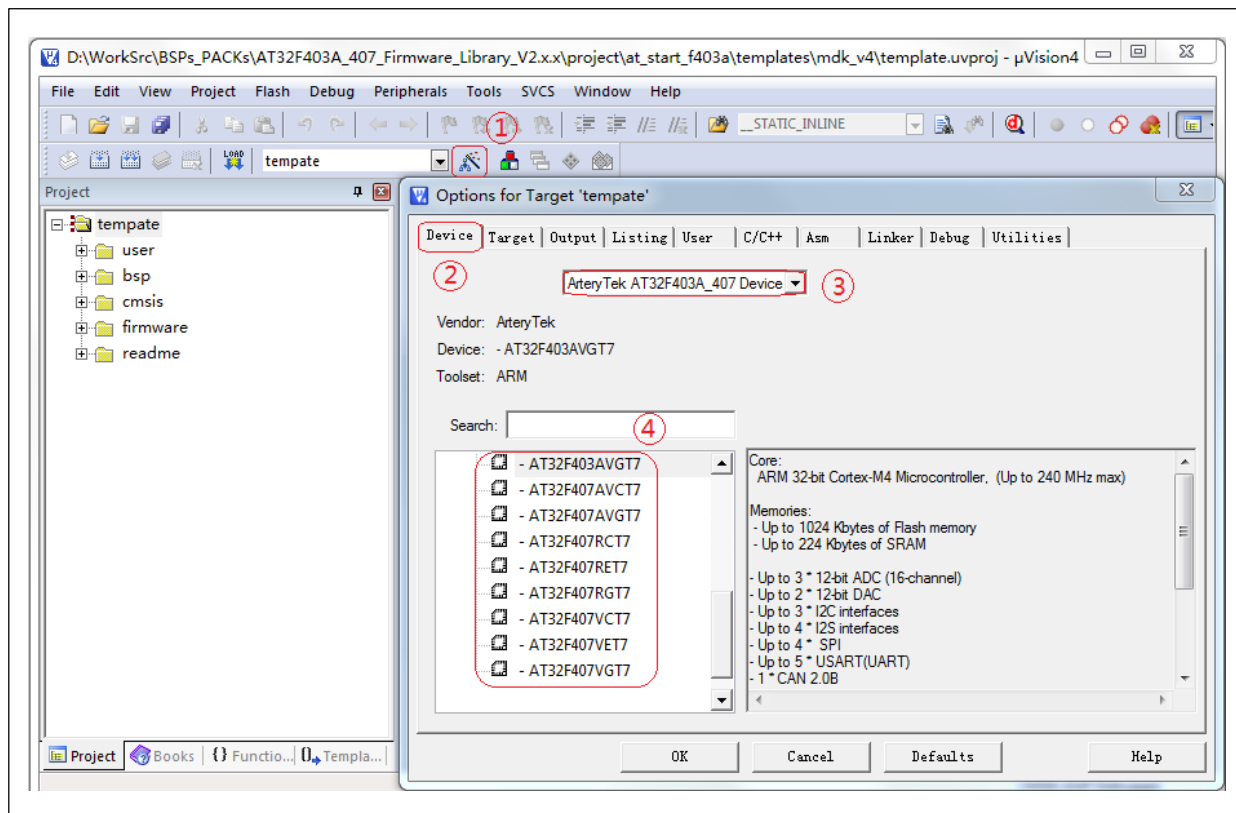
- d) 可以在上图的界面中修改 “Customer Information”，一般不需要修改此类信息。然后点击 “Next”启动安装过程，安装结果如下图：

图 16. Keil_v4 Pack 安装完成



- e) 点击“Finish”完成安装。查看 Keil_v4 Pack 安装是否成功。请按如下步骤进行操作和查看：
- ① 点击魔术棒。
 - ② 点选 Device 选项卡。
 - ③ 选择 ArteryTek 提供的对应系列的型号包文件。
 - ④ 出现 ArteryTek 信息及芯片型号。

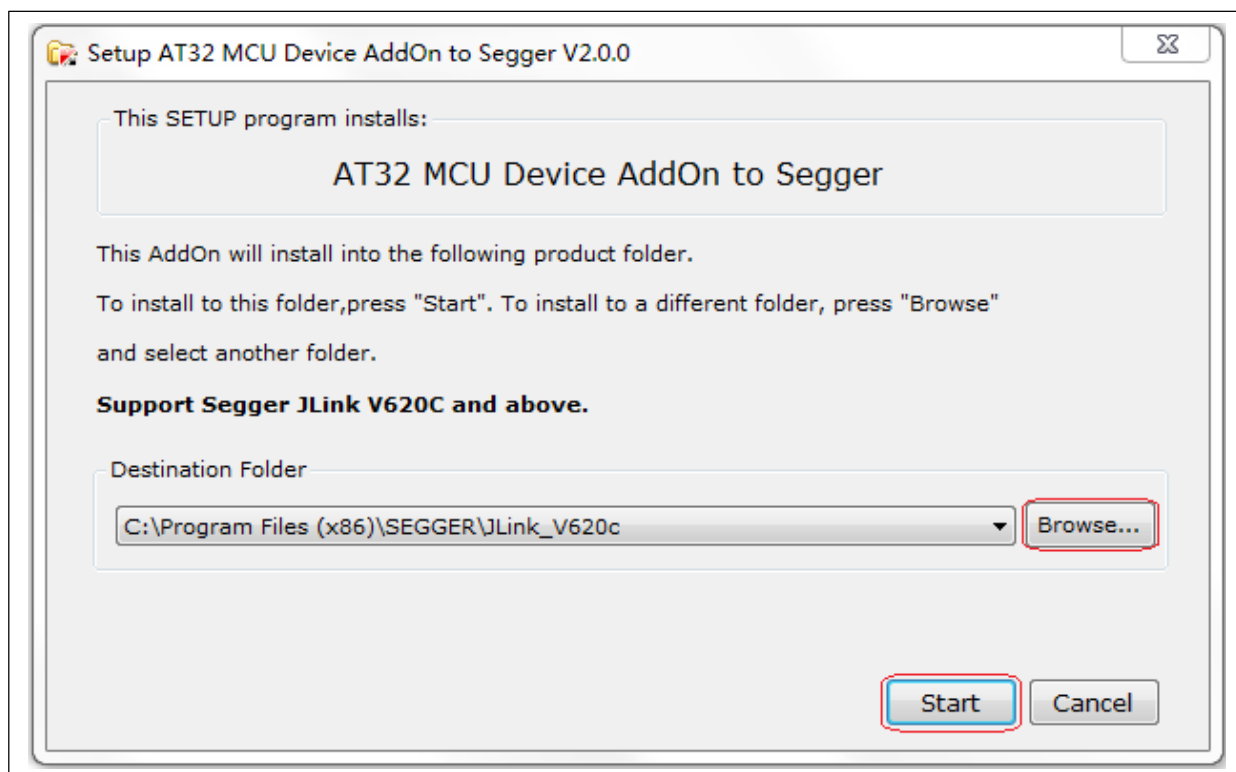
图 17. 查看 Keil_v4 Pack 安装情况



4. Segger_AT32MCU_AddOn.zip: 支援 J-Flash 下载的压缩包, 安装步骤如下:

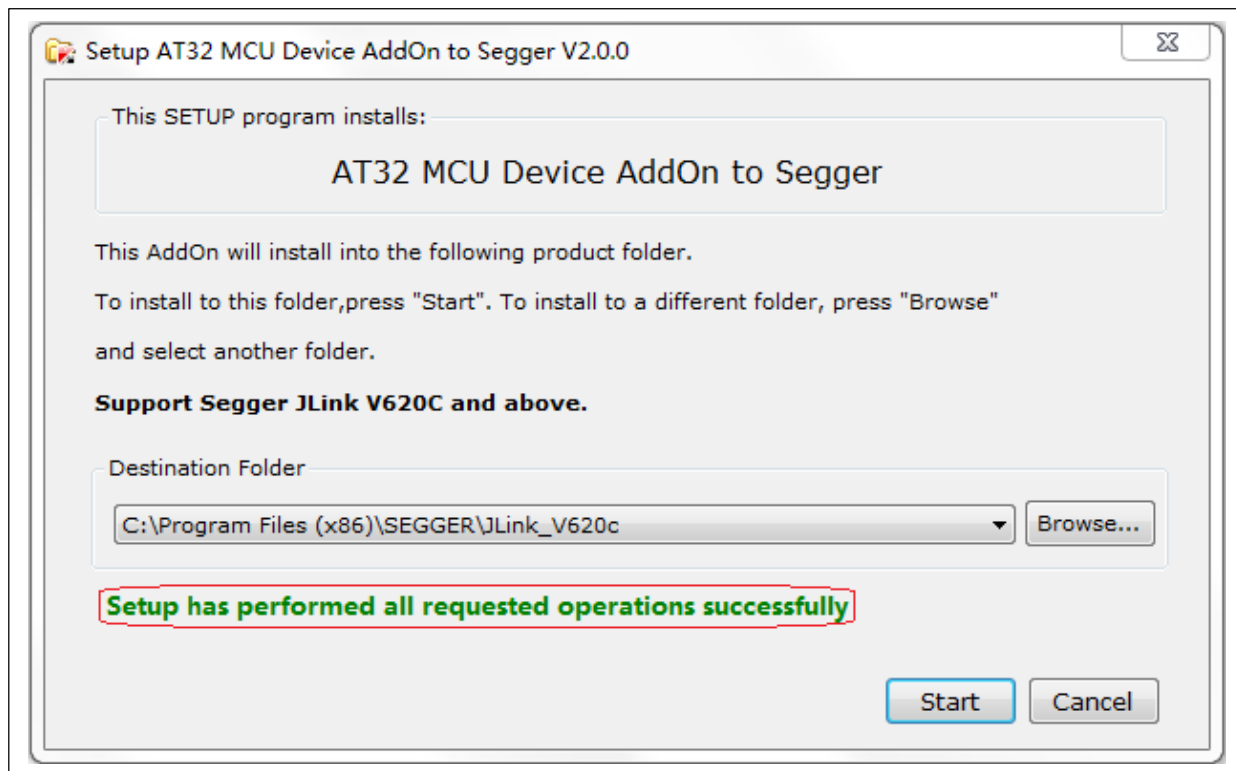
- 解压 Segger_AT32MCU_AddOn.zip。
- 双击 Segger_AT32MCU_AddOn.exe, 弹出如下界面 (具体版本信息按实际情况为准)。

图 18. Segger 包安装界面



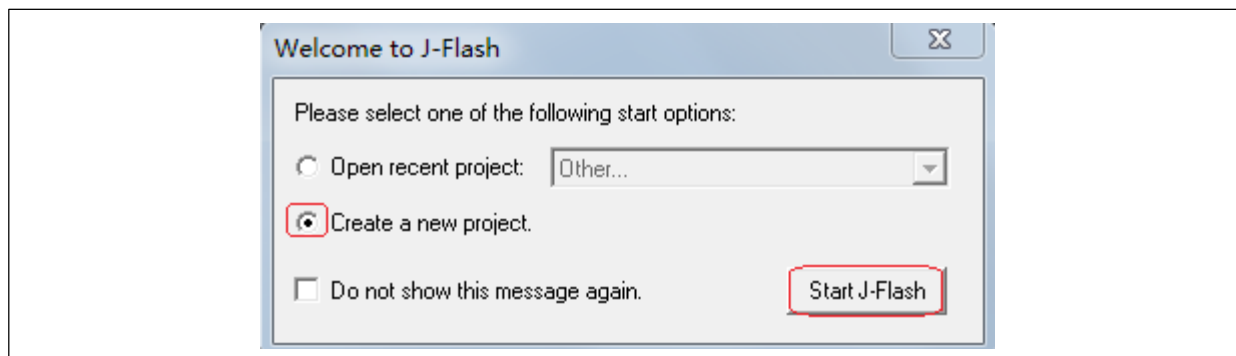
- 如果 Segger 的实际安装路径与 “Destination Folder”对话框内的路径不一致, 点击“Browse”选择实际安装路径。然后点击“Start”, 弹出如下界面。

图 19. Segger 包安装流程



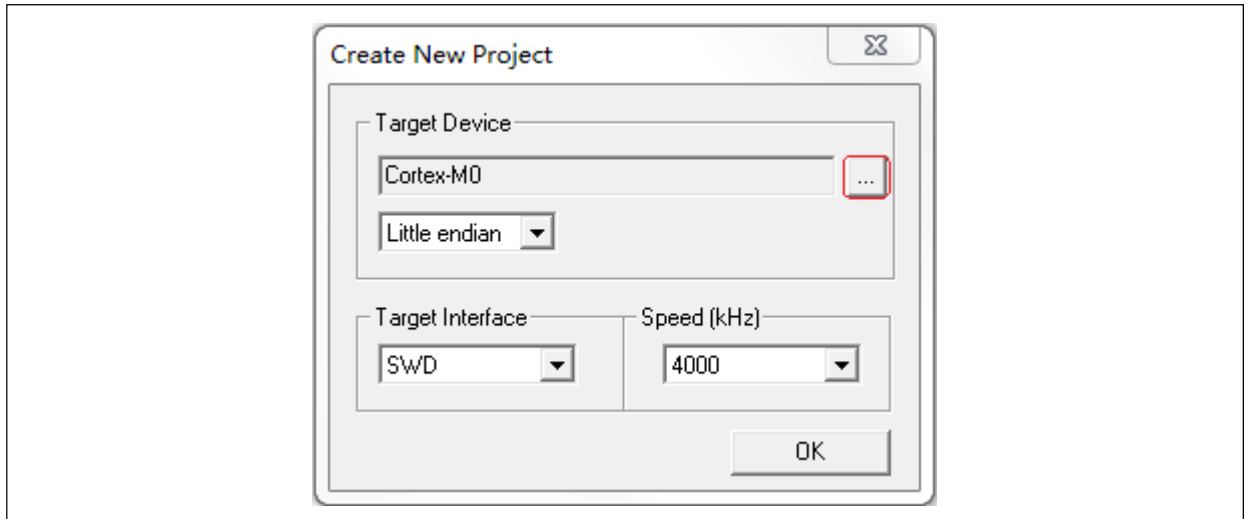
- d) 出现“Setup has performed all requested operations successfully”则表示已安装成功。查看是否安装成功，请按如下步骤进行操作和查看：
- i. 打开 J-Flash.exe，出现如下对话框则选择 Create a new project 并点击 Start J-Flash 按钮，如下图：

图 20. 打开 J-Flash



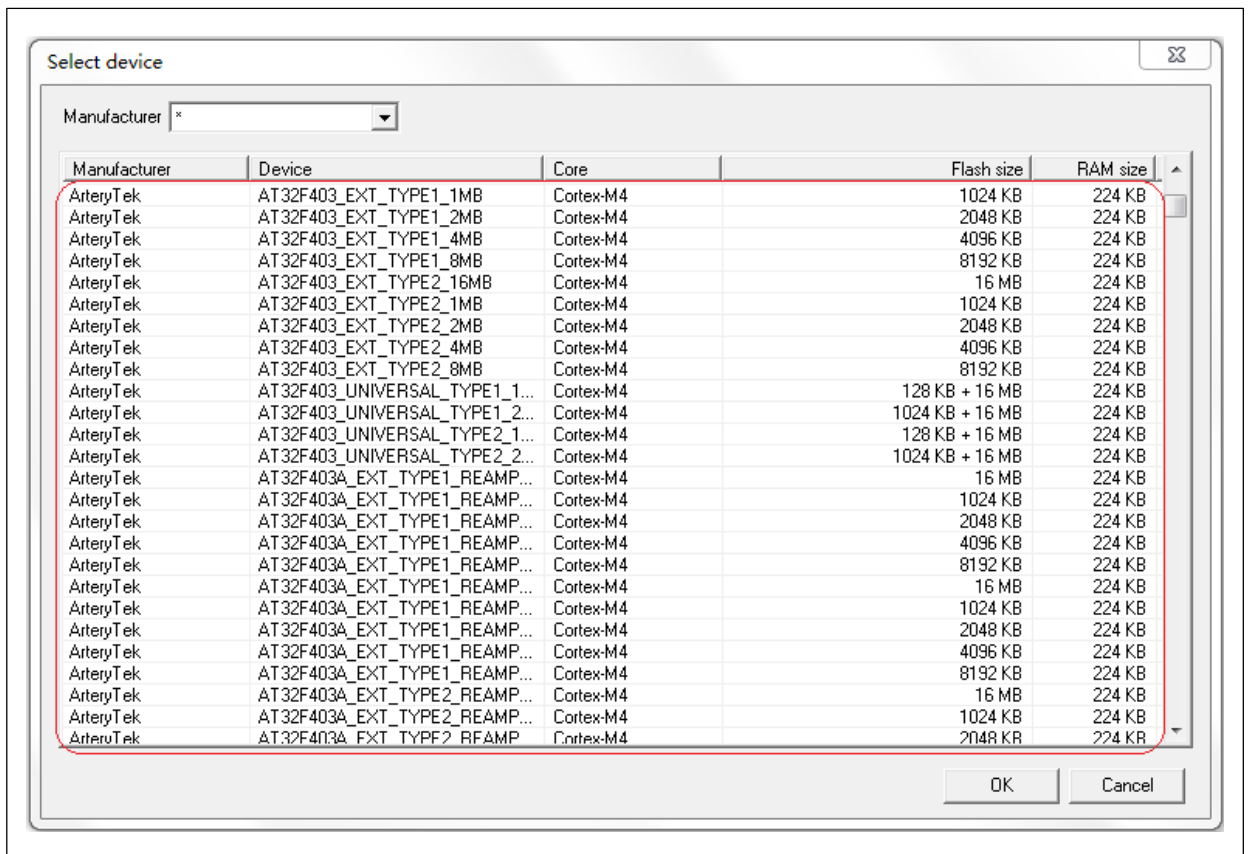
- ii. 启动 J-Flash 后，点击 Target Device 栏后的复选按钮，如下图：

图 21. J-Flash 创建新工程



- iii. 在复选框中上下拉动滚动条如查找到 ArteryTek 相关信息及算法文件则表示安装成功，如下图：

图 22. 查看 Device 信息



4 Flash 算法文件说明

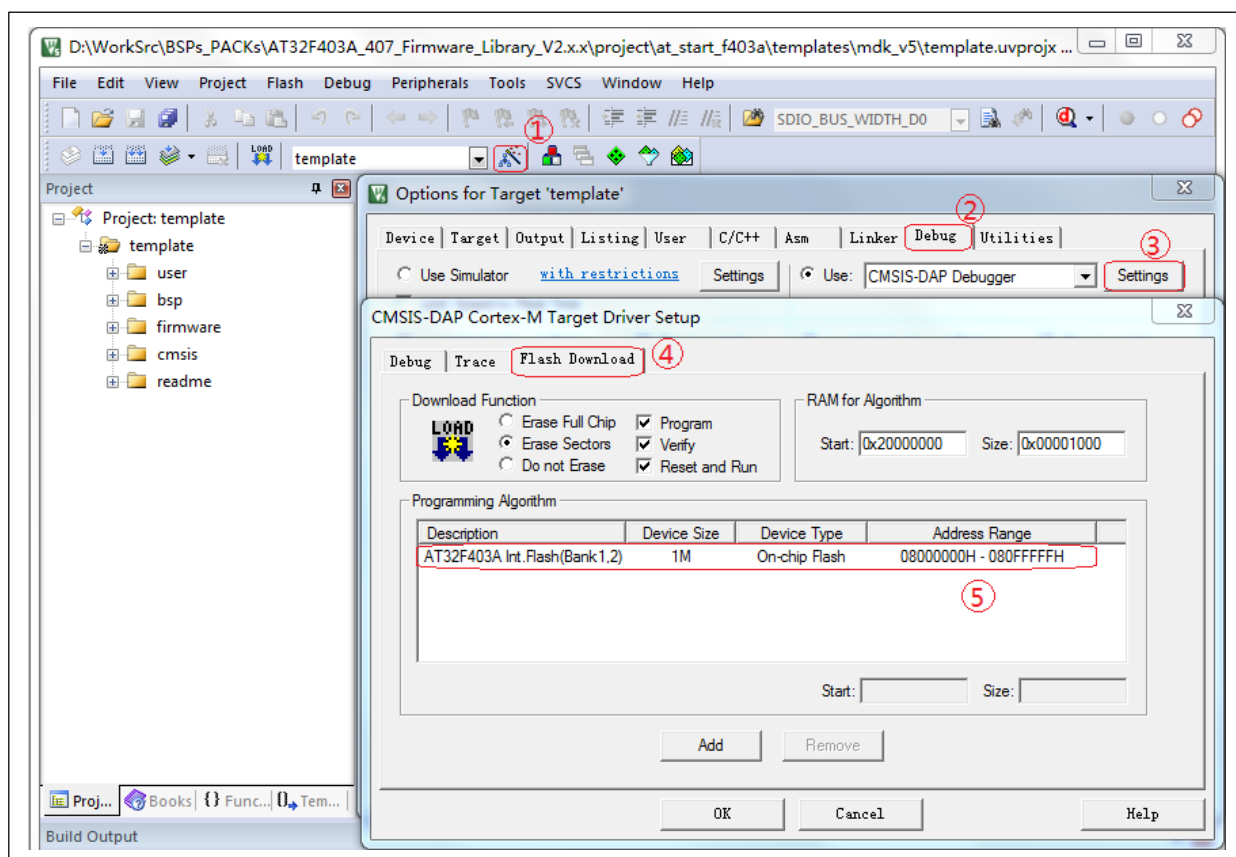
对于Artery MCU，我们都有在对应发布的Pack文件中整合了相关型号的Flash算法文件以供如KEIL/IAR等IDE工具进行在线code下载。虽各IDE工具对于算法文件的使用方法大致都一样，以下还是对算法文件的使用方法进行简单的说明。

1. Keil 算法文件的使用方法

因常用的Keil_v4和Keil_v5 IDE开发环境在算法文件选择方法和使用上基本一样，以下对应Keil_v5环境的使用来进行说明。

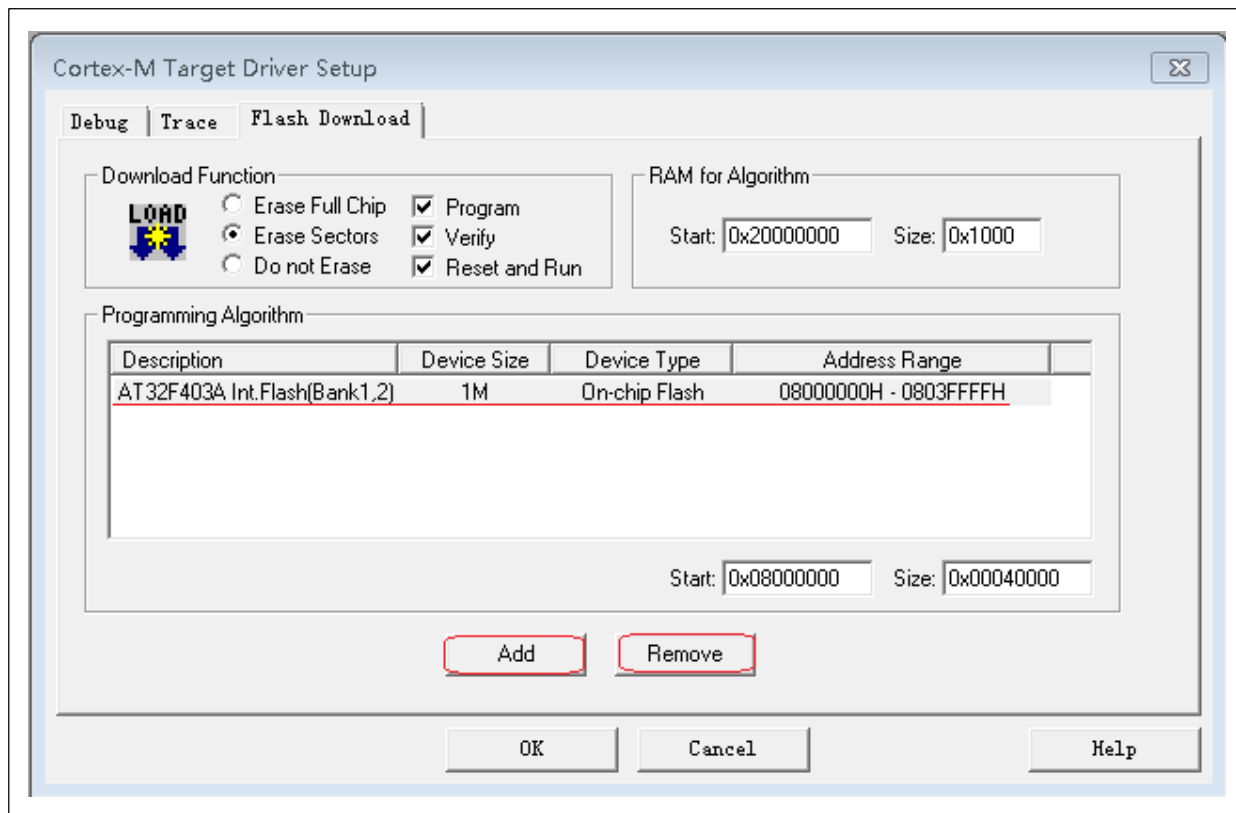
当在Keil IDE开发工具工程建立起来之后即可进行Debug方式配置和flash算法文件的选择。在开发工具内依次点击：配置魔术棒—>Debug选项卡—>Settings—>Flash Download，流程如下图：

图 23. Keil 算法文件设置



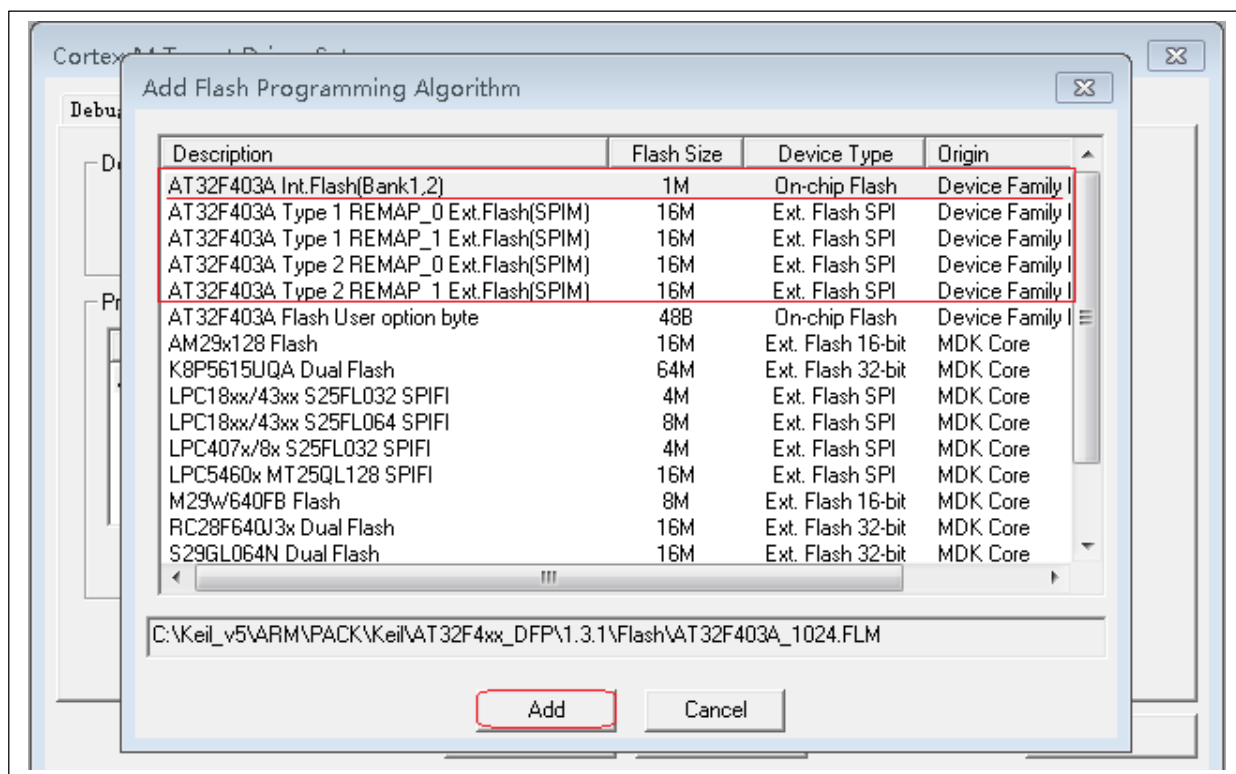
此处示例可看到所选择的Flash算法文件为默认的Flash算法文件，如需更改和移除可自行配置，点击到算法文件后可看到Add和Remove按钮可选择，如所选算法和实际MCU不匹配可使用以下方法重新配置

图 24. Keil 算法文件配置栏



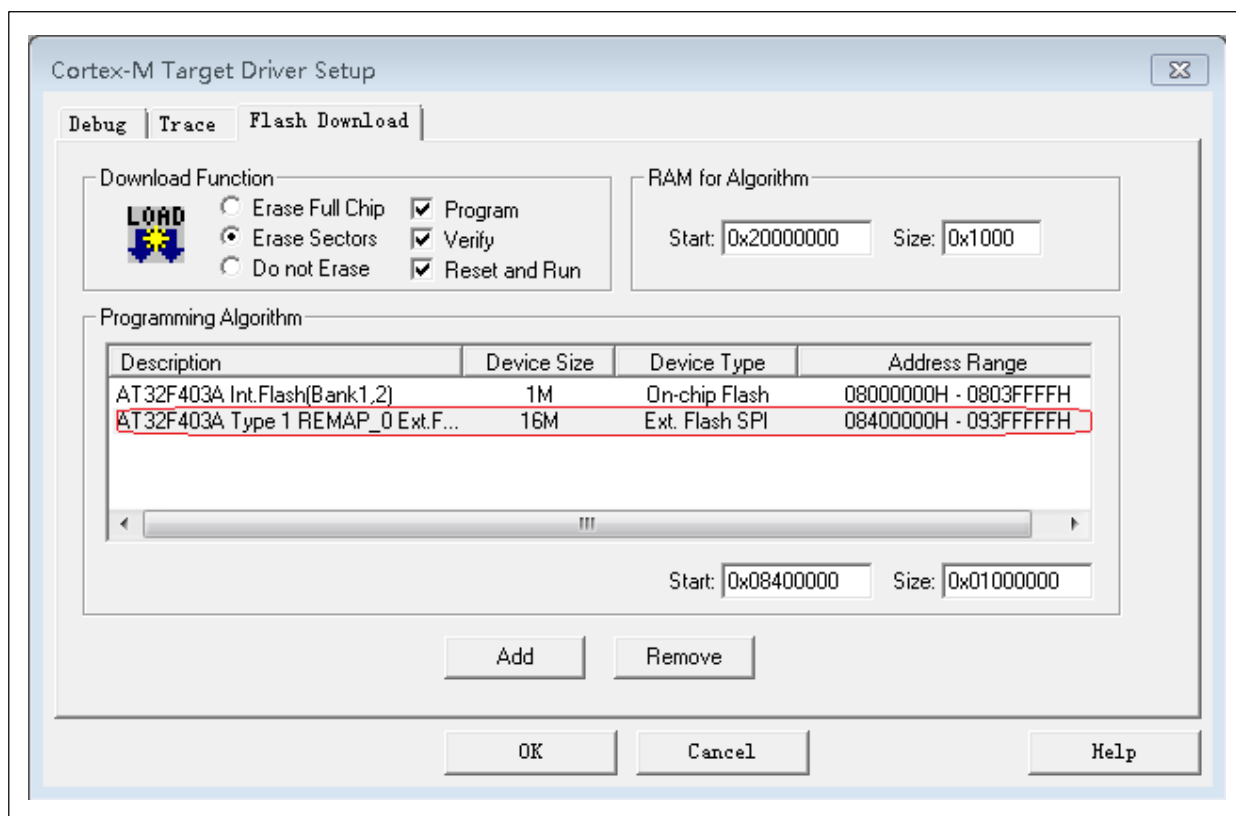
点击Remove可将当前选择到的算法文件从工程配置中移除，点击Add可查看支持此型MCU的算法文件并进行选择，示例如下：

图 25. Keil 选择算法文件



当选择到相应的算法文件后点击Add即可将新算法文件加入到当前工程配置，如下示例是新增SPIM算法到工程中：

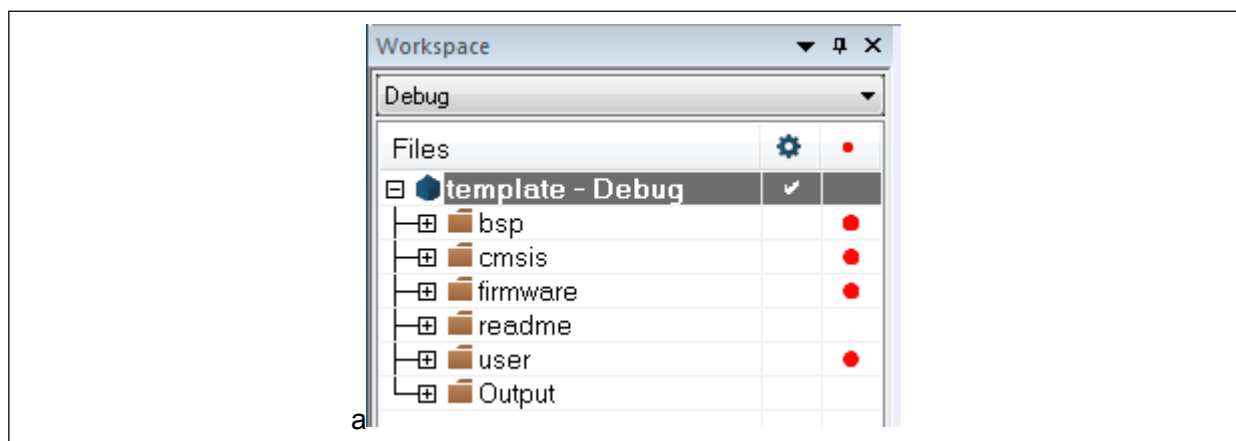
图 26. Keil 新增算法文件



2. IAR 算法文件的使用方法

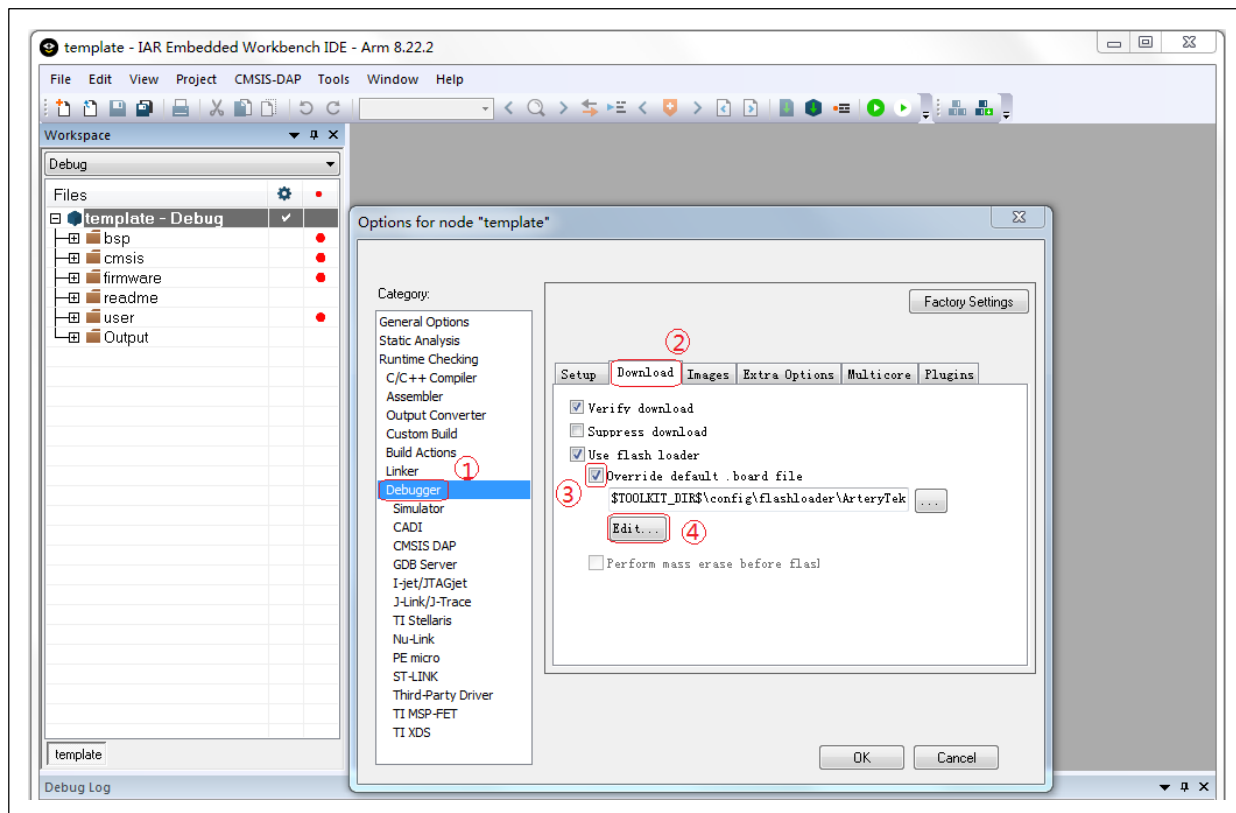
IAR开发环境对算法文件的选择方法是在当新建工程的配置中选定指定的MCU型号后自动选定的对应的默认flash算法文件。如需手动去进行算法文件配置，可在IAR工程建立起来之后，鼠标右击如下灰色选框位置的工程名：

图 27. IAR 工程名



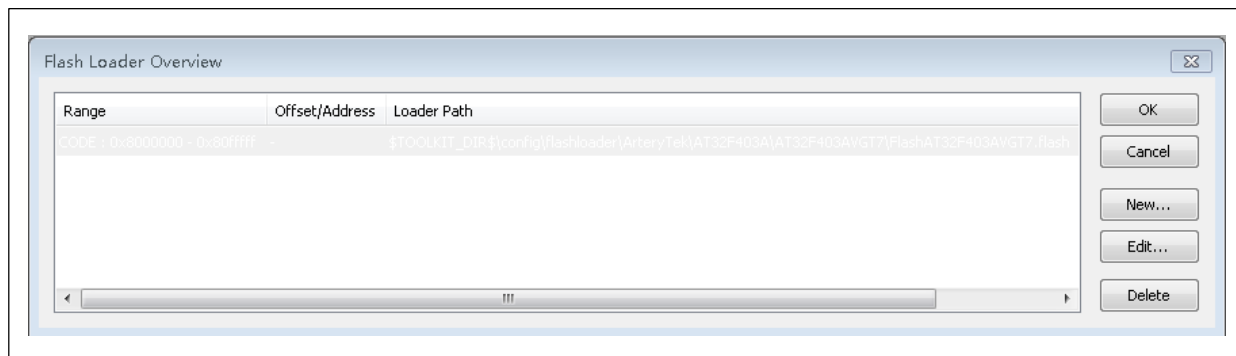
选择Options—>Debugger—>Download—>勾选Override default .board file—>点击Edit，流程如下图所示：

图 28. IAR 算法文件配置



进入后可看见如下的配置界面：

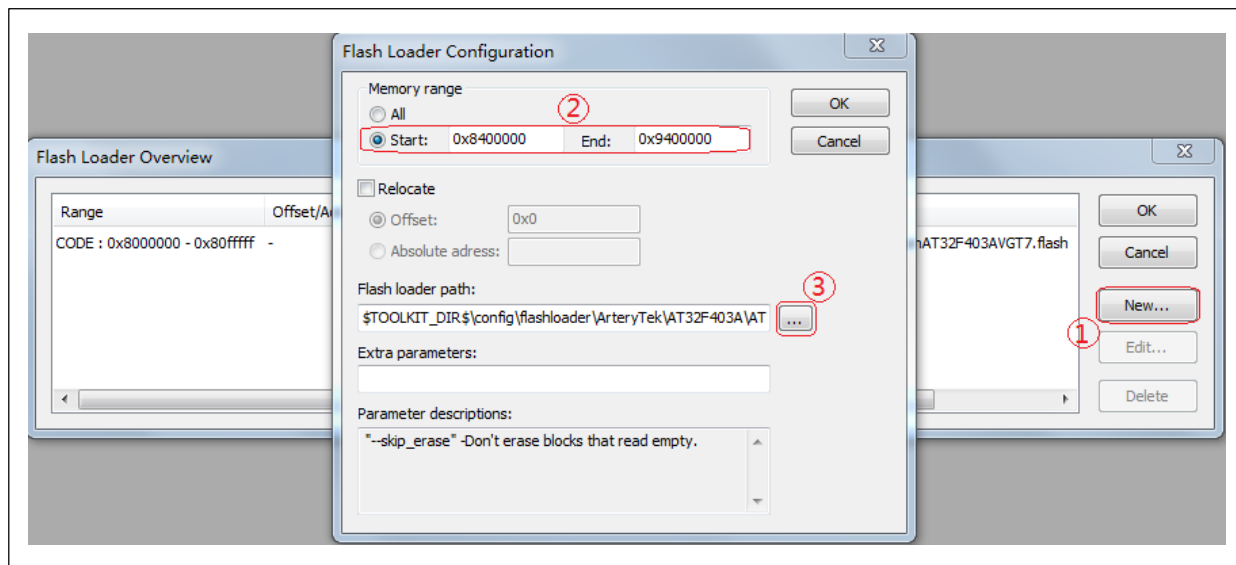
图 29. IAR Flash Loader 新增



其中的flash算法配置方法是选定MCU芯片型号后默认指定，如需手动进行修改可点击旁边的New/Edit/Delete三个选项进行修改。

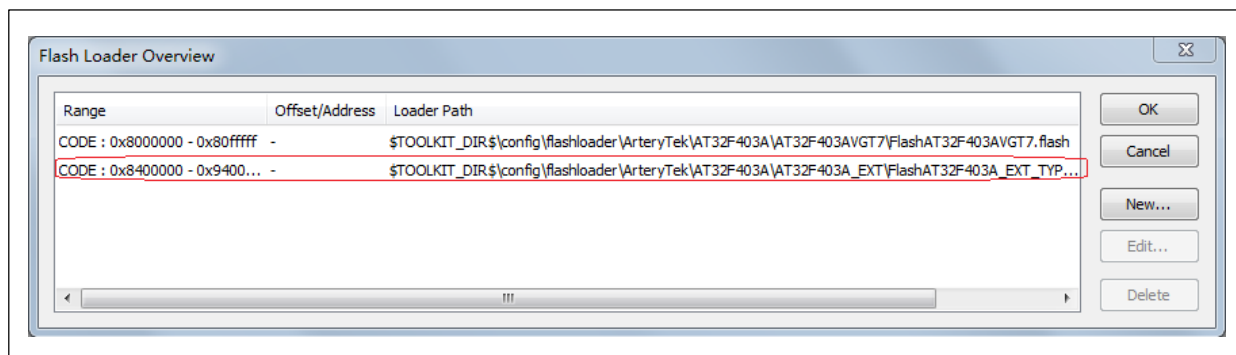
以点击New新增配置Flash算法文件举例。1.点击New—>2.配置Flash范围—>3.选择对应的Flash算法配置文件。流程如下图所示：

图 30. IAR Flash Loader 配置



此处示例是新增SPIM flash算法文件举例。需选择对应型号且正确的Flash算法文件进行配置。被选择的flash算法配置文件是由IAR_AT32MCU_AddOn工具安装到IAR开发环境内。示例新增的SPIM flash算法完成配置后如下图：

图 31. IAR Flash Loader 配置成功



3. SPIM 算法文件说明

Artery部分MCU 支持Bank3（详情请参考官方Reference Manual或DataSheet），其接口外挂flash可作为内部flash不足或特殊应用需求情况下的flash存储介质的扩充，当软件程序中部分code或数据指定编译链接地址在SPIM存储空间时，IDE工具在线下载的过程中需要使用到此算法文件进行外部flash编程。Artery SPIM算法文件的命名方式如下：AT32F4xxTypeNREMAP_P Ext.Flash。

N=1,2

P=0,1

TYPEN：外接的SPI Flash类型，按外接flash类型和型号进行选择。详细信息请参考对应MCU Reference Manual的FLASH_SELECT寄存器描述。

REMAP_P：MCU SPIM PIN脚的复用选择，按连接外部flash的硬件电路PIN脚连线方式进行选择。详细信息请参考对应MCU Reference Manual的外部SPIF重映射章节。

REMAP0：EXT_SPIF_GRP=000

REMAP1：EXT_SPIF_GRP=001

5 版本历史

表 2. 文档版本历史

日期	版本	变更
2021.11.26	2.0.0	最初版本
2022.05.09	2.0.1	新增2.4.1型号切换章节

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。

雅特力产品并非设计或专门用于下列用途的产品：(A) 对安全性有特别要求的应用，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；(D) 航天应用或航天环境，且/或(E) 武器。因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，风险由购买者单独承担，并且独力负责在此类相关使用中满足所有法律和法规要求。

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。

© 2022 雅特力科技 (重庆) 有限公司 保留所有权利