

1 概述

为了让用户高效快速的使用Artery MCU，我们提供了一套完整的BSP&Pack用于开发。主要包括：at32f4xx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

 Libraries	2019/5/28 10:45	文件夹
 Middlewares	2019/5/28 10:45	文件夹
 Project	2019/5/28 10:46	文件夹
 Utilities	2019/5/28 10:46	文件夹

Libraries:

- AT32F4xx_StdPeriph_Driver: at32f4xx外设驱动库。
- CMSIS:
 - 1) CM4: 内核相关，包括Cortex-M4库文件，at32f4xx系统初始化文件、启动代码等。
 - 2) Documentation: 相关文档。

Middlewares:

- 第三方软件包，如FreeRTOS等。

Project:

- Examples: at32f4xx相关的应用案例。
- Templates: at32f4xx工程模板。

Utilities:

- 各APP Demo存放目录

2.2 BSP使用方法

1. 在创建工程时，需要根据MCU型号，选择正确的启动代码，MCU型号与启动代码对应关系如下表：

MCU型号	启动代码	PINs	Flash大小(KB)
AT32F403CC/E	startup_at32f403cx_hd.s	48	256/512
AT32F403CG	startup_at32f403cx_xl.s	48	1024
AT32F403RC/E	startup_at32f403rx_hd.s	64	256/512
AT32F403RG	startup_at32f403rx_xl.s	64	1024
AT32F403VC/E	startup_at32f403vx_hd.s	100	256/512

AT32F403VG	startup_at32f403vx_xl.s	100	1024
AT32F403ZC/E	startup_at32f403zx_hd.s	144	256/512
AT32F403ZG	startup_at32f403zx_xl.s	144	1024
AT32F413KB	startup_at32f413kx_md.s	32	128
AT32F413KC	startup_at32f413kx_hd.s	32	256
AT32F413C8/B	startup_at32f413cx_md.s	48	64/128
AT32F413CC	startup_at32f413cx_hd.s	48	256
AT32F413RB	startup_at32f413rx_md.s	64	128
AT32F413RC	startup_at32f413rx_md.s	64	256
AT32FEBKC8	startup_at32f413cx_hd.s	48	64
TS32F401CBU7	startup_ts32f401cbu7.s	48	128
AT32F415K8U7-4	startup_at32f415k8u7-4.s	32	64
AT32F415KBU7-4	startup_at32f415kbu7-4.s	32	128
AT32F415KCU7-4	startup_at32f415kcu7-4.s	32	256
AT32F415C8T7	startup_at32f415c8t7.s	48	64
AT32F415CBT7	startup_at32f415cbt7.s	48	128
AT32F415CCT7	startup_at32f415cct7.s	48	256
AT32F415R8T7	startup_at32f415r8t7.s	64	64
AT32F415RBT7	startup_at32f415rbt7.s	64	128
AT32F415RCT7	startup_at32f415rct7.s	64	256
AT32F415R8T7-7	startup_at32f415r8t7-7.s	64	64
AT32F415RBT7-7	startup_at32f415rbt7-7.s	64	128
AT32F415RCT7-7	startup_at32f415rct7-7.s	64	256
AT32F403ACCT7	startup_at32f403acct7.s	48	256
AT32F403ACET7	startup_at32f403acet7.s	48	512
AT32F403ACGT7	startup_at32f403acgt7.s	48	1024
AT32F403ACCU7	startup_at32f403accu7.s	48	256
AT32F403ACEU7	startup_at32f403aceu7.s	48	512
AT32F403ACGU7	startup_at32f403acgu7.s	48	1024
AT32F403ARCT7	startup_at32f403arct7.s	64	256
AT32F403ARET7	startup_at32f403aret7.s	64	512
AT32F403ARGT7	startup_at32f403argt7.s	64	1024
AT32F403AVCT7	startup_at32f403avct7.s	100	256
AT32F403AVET7	startup_at32f403avet7.s	100	512
AT32F403AVGT7	startup_at32f403avgt7.s	100	1024
AT32F407RCT7	startup_at32f407rct7.s	64	256
AT32F407RET7	startup_at32f407ret7.s	64	512
AT32F407RGT7	startup_at32f407rgt7.s	64	1024
AT32F407VCT7	startup_at32f407vct7.s	100	256
AT32F407VET7	startup_at32f407vet7.s	100	512

AT32F407VGT7	startup_at32f407vgt7.s	100	1024
--------------	------------------------	-----	------

2. 在at32f4xx.h中，是通过宏定义的方式选择MCU型号，示例如下图（实际会更多）：

```

41 /* Uncomment the line below according to the target AT32 device used in your
42 application
43 */
44 #if !defined (AT32F403Cx_HD) && !defined (AT32F403Cx_XL) && \
45     !defined (AT32F403Rx_HD) && !defined (AT32F403Rx_XL) && \
46     !defined (AT32F403Vx_HD) && !defined (AT32F403Vx_XL) && \
47     !defined (AT32F403Zx_HD) && !defined (AT32F403Zx_XL)
48 /* #define AT32F403Cx_HD */ /*!< AT32F403Cx_HD: LQFP48, High density devices: AT32F403CC, AT32F403CE */
49 /* #define AT32F403Cx_XL */ /*!< AT32F403Cx_XL: LQFP48, XL-density devices: AT32F403CG */
50 /* #define AT32F403Rx_HD */ /*!< AT32F403Rx_HD: LQFP64, High density devices: AT32F403RC, AT32F403RE */
51 /* #define AT32F403Rx_XL */ /*!< AT32F403Rx_XL: LQFP64, XL-density devices: AT32F403RG */
52 /* #define AT32F403Vx_HD */ /*!< AT32F403Vx_HD: LQFP100, High density devices: AT32F403VC, AT32F403VE */
53 /* #define AT32F403Vx_XL */ /*!< AT32F403Vx_XL: LQFP100, XL-density devices: AT32F403VG */
54 /* #define AT32F403Zx_HD */ /*!< AT32F403Zx_HD: LQFP144, High density devices: AT32F403ZC, AT32F403ZE */
55 /* #define AT32F403Zx_XL */ /*!< AT32F403Zx_XL: LQFP144, XL-density devices: AT32F403ZG */
56 #endif
57 /* Tip: To avoid modifying this file each time you need to switch between these
58 devices, you can define the device in your toolchain compiler preprocessor.
59
60 - High-density devices are at32f403xx microcontrollers where
61 the Flash memory density ranges between 256 and 512 Kbytes.
62 - XL-density devices are at32f403xx microcontrollers where
63 the Flash memory density ranges between 512 and 1024 Kbytes.
64 */
65
66 #if !defined (AT32F403Cx_HD) && !defined (AT32F403Cx_XL) && \
67     !defined (AT32F403Rx_HD) && !defined (AT32F403Rx_XL) && \
68     !defined (AT32F403Vx_HD) && !defined (AT32F403Vx_XL) && \
69     !defined (AT32F403Zx_HD) && !defined (AT32F403Zx_XL)
70 #error "Please select first the target at32f403 device used in your application (in at32f403.h file)"
71 #endif

```

所以，在Code编译之前，需要根据MCU型号，打开对应的宏定义，MCU型号与宏定义的对对应关系如下表。

MCU型号	宏定义	PINs	Flash大小(KB)
AT32F403CC/E	AT32F403Cx_HD	48	256/512
AT32F403CG	AT32F403Cx_XL	48	1024
AT32F403RC/E	AT32F403Rx_HD	64	256/512
AT32F403RG	AT32F403Rx_XL	64	1024
AT32F403VC/E	AT32F403Vx_HD	100	256/512
AT32F403VG	AT32F403Vx_XL	100	1024
AT32F403ZC/E	AT32F403Zx_HD	144	256/512
AT32F403ZG	AT32F403Zx_XL	144	1024
AT32F413KB	AT32F413Kx_MD	32	128
AT32F413KC	AT32F413Kx_HD	32	256
AT32F413C8/B	AT32F413Cx_MD	48	64/128
AT32F413CC	AT32F413Cx_HD	48	256
AT32F413RB	AT32F413Rx_MD	64	128
AT32F413RC	AT32F413Rx_HD	64	256
AT32FEBKC8	AT32FEBKCx_MD	48	64
TS32F401CBU7	TS32F401CBU7	48	128
AT32F415K8U7-4	AT32F415K8U7_4	32	64
AT32F415KBU7-4	AT32F415KBU7_4	32	128

AT32F415KCU7-4	AT32F415KCU7_4	32	256
AT32F415C8T7	AT32F415C8T7	48	64
AT32F415CBT7	AT32F415CBT7	48	128
AT32F415CCT7	AT32F415CCT7	48	256
AT32F415R8T7	AT32F415R8T7	64	64
AT32F415RBT7	AT32F415RBT7	64	128
AT32F415RCT7	AT32F415RCT7	64	256
AT32F415R8T7-7	AT32F415R8T7_7	64	64
AT32F415RBT7-7	AT32F415RBT7_7	64	128
AT32F415RCT7-7	AT32F415RCT7_7	64	256
AT32F403ACCT7	AT32F403ACCT7	48	256
AT32F403ACET7	AT32F403ACET7	48	512
AT32F403ACGT7	AT32F403ACGT7	48	1024
AT32F403ACCU7	AT32F403ACCU7	48	256
AT32F403ACEU7	AT32F403ACEU7	48	512
AT32F403ACGU7	AT32F403ACGU7	48	1024
AT32F403ARCT7	AT32F403ARCT7	64	256
AT32F403ARET7	AT32F403ARET7	64	512
AT32F403ARGT7	AT32F403ARGT7	64	1024
AT32F403AVCT7	AT32F403AVCT7	100	256
AT32F403AVET7	AT32F403AVET7	100	512
AT32F403AVGT7	AT32F403AVGT7	100	1024
AT32F407RCT7	AT32F407RCT7	64	256
AT32F407RET7	AT32F407RET7	64	512
AT32F407RGT7	AT32F407RGT7	64	1024
AT32F407VCT7	AT32F407VCT7	100	256
AT32F407VET7	AT32F407VET7	100	512
AT32F407VGT7	AT32F407VGT7	100	1024

3. 在at32f4xx.h中，通过宏定义LIBRARY_VERSION可以选择是否“包含”MCU外设相关的头文件，如下图：

```
#if defined (LIBRARY_VERSION) || defined (USE_STDPERIPH_DRIVER)
#include "at32f4xx_conf.h"
#endif
```

at32f4xx_conf.h“包含”了所有外设头文件，如下图：

```
#include "at32f4xx_acc.h"
#include "at32f4xx_adc.h"
#include "at32f4xx_bkp.h"
#include "at32f4xx_can.h"
#include "at32f4xx_comp.h"
#include "at32f4xx_crc.h"
#include "at32f4xx_dac.h"
#include "at32f4xx_dbgmcu.h"
#include "at32f4xx_dma.h"
#include "at32f4xx_ertc.h"
#include "at32f4xx_exti.h"
#include "at32f4xx_flash.h"
#include "at32f4xx_gpio.h"
#include "at32f4xx_i2c.h"
#include "at32f4xx_iwdg.h"
#include "at32f4xx_pwr.h"
#include "at32f4xx_rcc.h"
#include "at32f4xx_rtc.h"
#include "at32f4xx_sdio.h"
#include "at32f4xx_spi.h"
#include "at32f4xx_tim.h"
#include "at32f4xx_usart.h"
#include "at32f4xx_wwdg.h"
#include "at32f4xx_xmc.h"
#include "misc.h"
```

4. System_at32f4xx.c系统时钟初始化，需要打开一个宏定义选择初始化时钟，类似如下图：

```

/* #if defined (AT32F403xx) || defined (AT32F413xx) || \
   defined (AT32F415xx) || defined (AT32F403Axx) || \
   defined (AT32F407xx)
/* #define SYSCCLK_FREQ_HSE      HSE_VALUE */
/* #define SYSCCLK_FREQ_24MHz    24000000 */
/* #define SYSCCLK_FREQ_36MHz    36000000 */
/* #define SYSCCLK_FREQ_48MHz    48000000 */
/* #define SYSCCLK_FREQ_56MHz    56000000 */
/* #define SYSCCLK_FREQ_72MHz    72000000 */
/* #define SYSCCLK_FREQ_96MHz    96000000 */
/* #define SYSCCLK_FREQ_108MHz   108000000 */
/* #define SYSCCLK_FREQ_120MHz   120000000 */
/* #define SYSCCLK_FREQ_144MHz   144000000 */
/* #define SYSCCLK_FREQ_24MHz_HSI 24000000 */
/* #define SYSCCLK_FREQ_36MHz_HSI 36000000 */
/* #define SYSCCLK_FREQ_48MHz_HSI 48000000 */
/* #define SYSCCLK_FREQ_56MHz_HSI 56000000 */
/* #define SYSCCLK_FREQ_72MHz_HSI 72000000 */
/* #define SYSCCLK_FREQ_96MHz_HSI 96000000 */
/* #define SYSCCLK_FREQ_108MHz_HSI 108000000 */
/* #define SYSCCLK_FREQ_120MHz_HSI 120000000 */
/* #define SYSCCLK_FREQ_144MHz_HSI 144000000 */
#endif
/* #if defined (AT32F415xx)
/* #define SYSCCLK_FREQ_150MHz    150000000 */
/* #define SYSCCLK_FREQ_150MHz_HSI 150000000 */
#endif
/* #if defined (AT32F403xx) || defined (AT32F413xx) || \
   defined (AT32F403Axx) || defined (AT32F407xx)
/* #define SYSCCLK_FREQ_168MHz    168000000 */
/* #define SYSCCLK_FREQ_176MHz    176000000 */
/* #define SYSCCLK_FREQ_192MHz    192000000 */
/* #define SYSCCLK_FREQ_200MHz    200000000 */
/* #define SYSCCLK_FREQ_168MHz_HSI 168000000 */
/* #define SYSCCLK_FREQ_176MHz_HSI 176000000 */
/* #define SYSCCLK_FREQ_192MHz_HSI 192000000 */
/* #define SYSCCLK_FREQ_200MHz_HSI 200000000 */
#endif
/* #if defined (AT32F403Axx) || defined (AT32F407xx)
/* #define SYSCCLK_FREQ_224MHz    224000000 */
/* #define SYSCCLK_FREQ_240MHz    240000000 */
/* #define SYSCCLK_FREQ_224MHz_HSI 224000000 */
/* #define SYSCCLK_FREQ_240MHz_HSI 240000000 */
#endif

```

系统时钟初始化结果如下图：

1. After each device reset the HSI is used as System clock source.
2. Please make sure that the selected System clock doesn't exceed your device's maximum frequency.
3. If none of the define below is enabled, the HSI is used as System clock source.
4. The System clock configuration functions provided within this file assume that:
 - For at32f4xx devices, an external 8MHz crystal is used to drive the System clock.
 If you are using different crystal you have to adapt those functions accordingly.






Clock (MHz)

SYSCCLK	HCLK	PCLK2	PCLK1
24	24	24	24
36	36	36	36
48	48	48	24
56	56	56	28
72	72	72	36
96	96	48	48
108	108	54	54
120	120	60	60
144	144	72	72
150	150	75	75
168	168	84	84
176	176	88	88
192	192	96	96
200	200	100	100
224	224	112	112
240	240	120	120

3 Pack安装步骤

Artery 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。

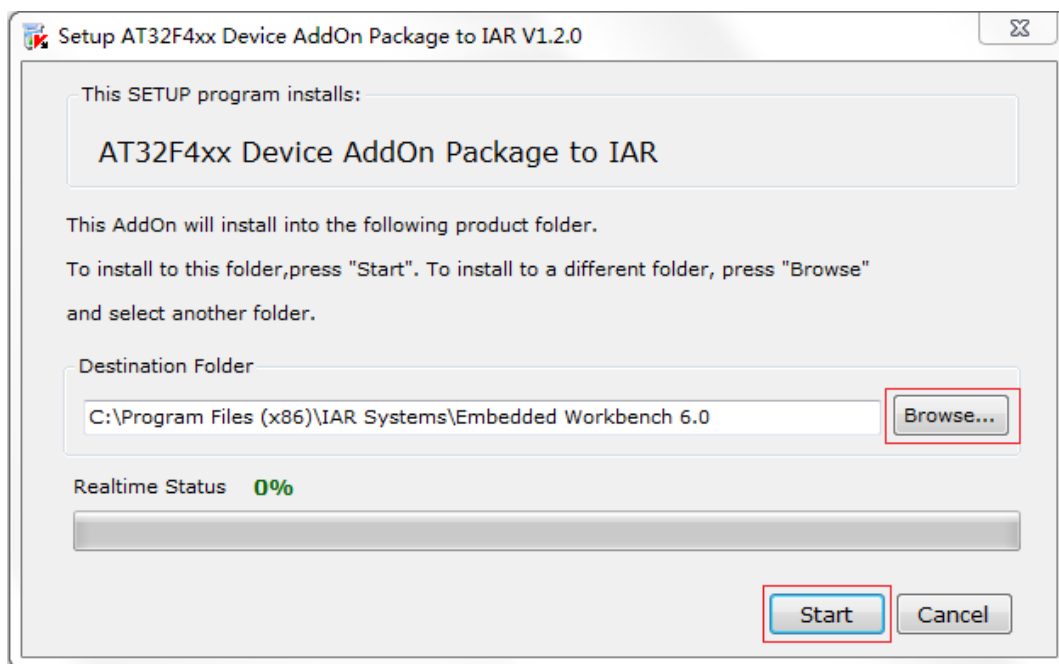
Pack安装文件如下图（具体版本信息按实际情况为准）。

 IAR_AT32F4xx_AddOn.exe	19/12/26 10:24	应用程序	5,561 KB
 Keil.AT32F4xx_DFP.1.3.0.pack	19/12/26 14:26	uVision Software...	858 KB
 Keil_AT32F4xx_AddOn.exe	19/12/26 10:26	应用程序	11,990 KB
 ReleaseNotes_AT32F4xx_Packs.pdf	20/01/07 14:05	Adobe Acrobat ...	292 KB
 Segger_AT32F4xx_AddOn.exe	20/01/07 13:48	应用程序	829 KB

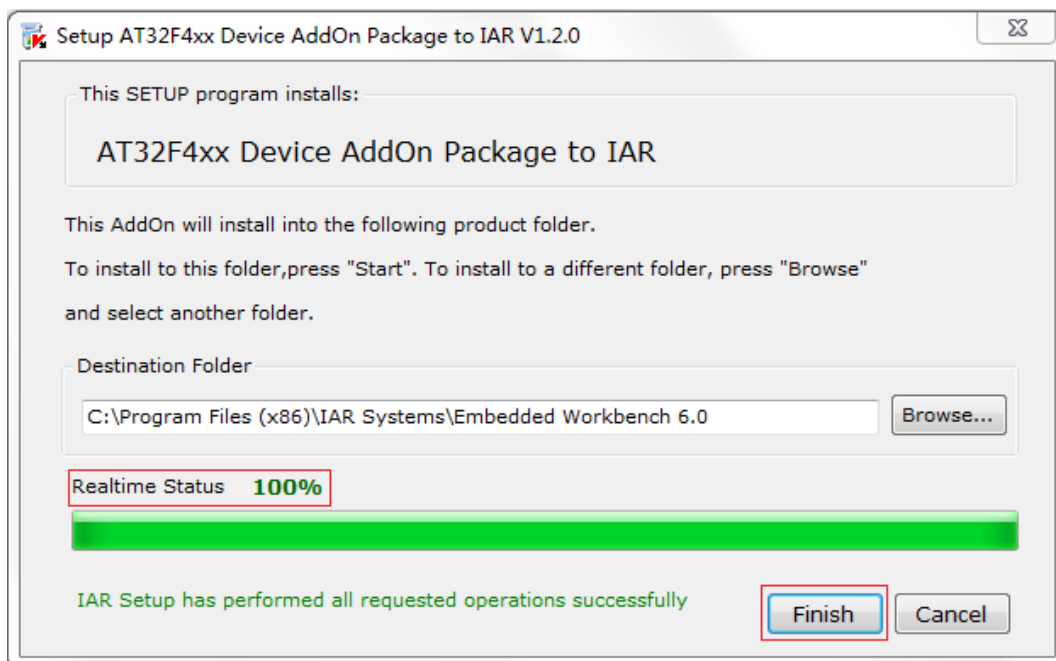
其中：

（1） IAR_AT32F4xx_AddOn：支援IAR_v6、IAR_v7和IAR_v8，安装步骤如下：

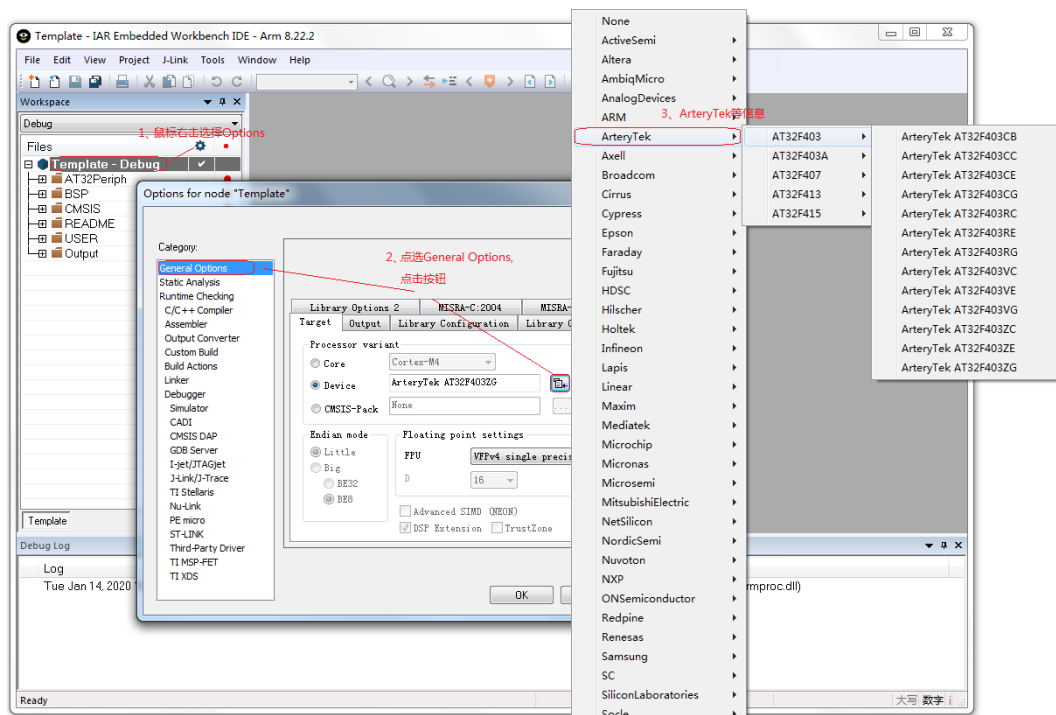
A、 双击IAR_AT32Fxx_AddOn.exe，弹出如下界面（具体版本信息按实际情况为准）。



B、 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。

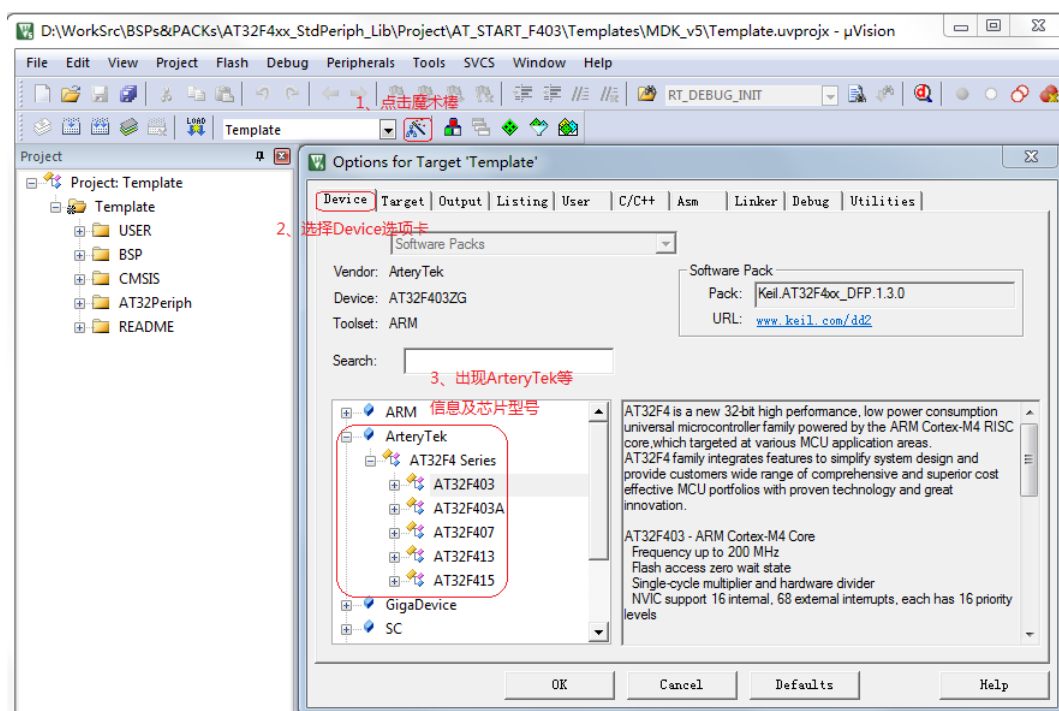


- C、 点击“Finish”完成安装。
- D、 查看IAR Pack是否安装成功。任意打开一个IAR工程，按如下步骤操作和查看：



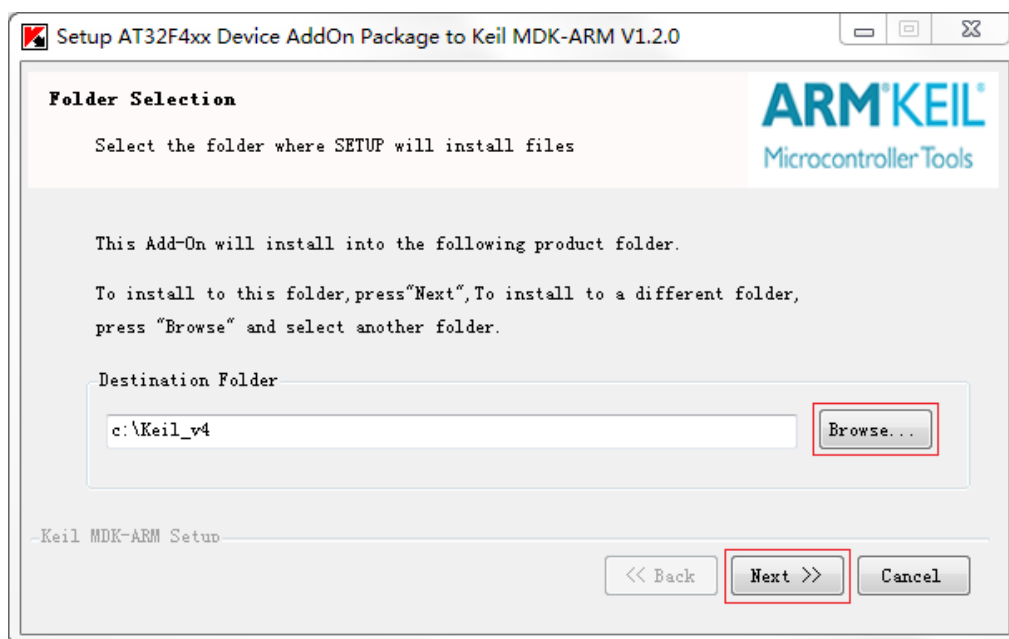
(2) Keil.AT32F4xx_DFP.1.x.x: 支援Keil_v5, 具体版本号见实际安装包, 安装步骤如下:

- A、 标准的Keil_v5 DFP安装文件, 双击Keil.AT32F4xx_DFP.1.x.x.pack完成一键式安装。
- B、 查看Keil_v5 Pack是否安装成功。按如下步骤操作和查看:

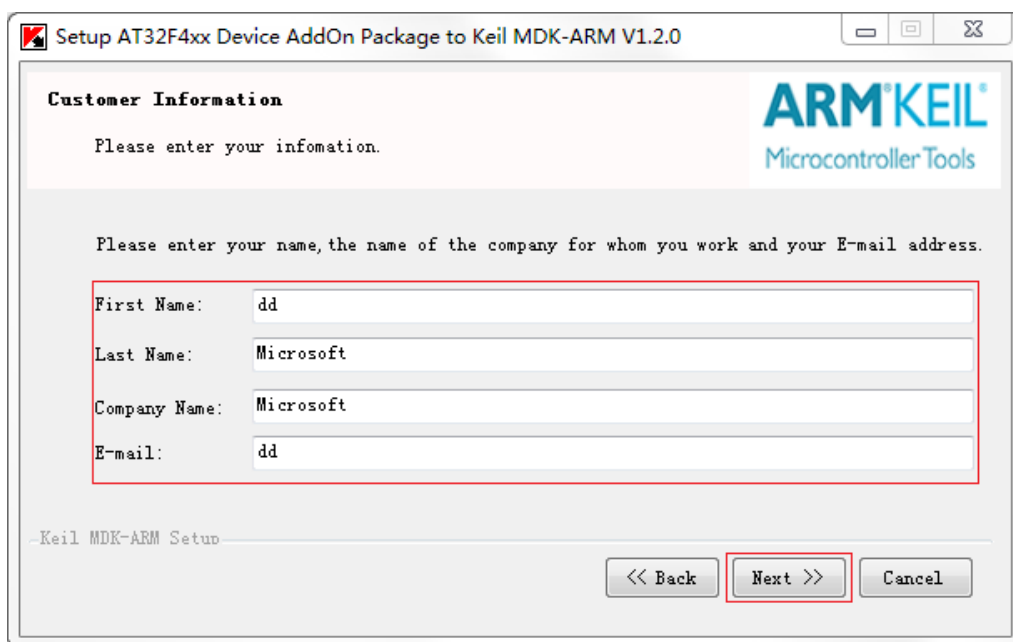


(3) Keil_AT32F4xx_AddOn: 支援Keil_v4, 安装步骤如下:

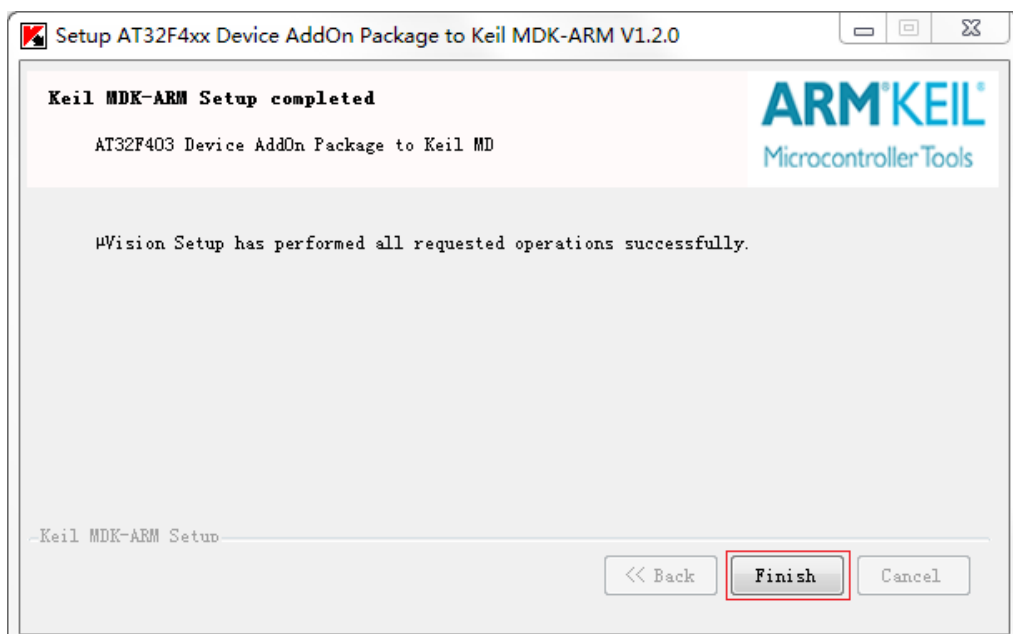
A、 双击Keil_AT32F4xx_AddOn.exe, 弹出如下界面 (具体版本信息按实际情况为准)。



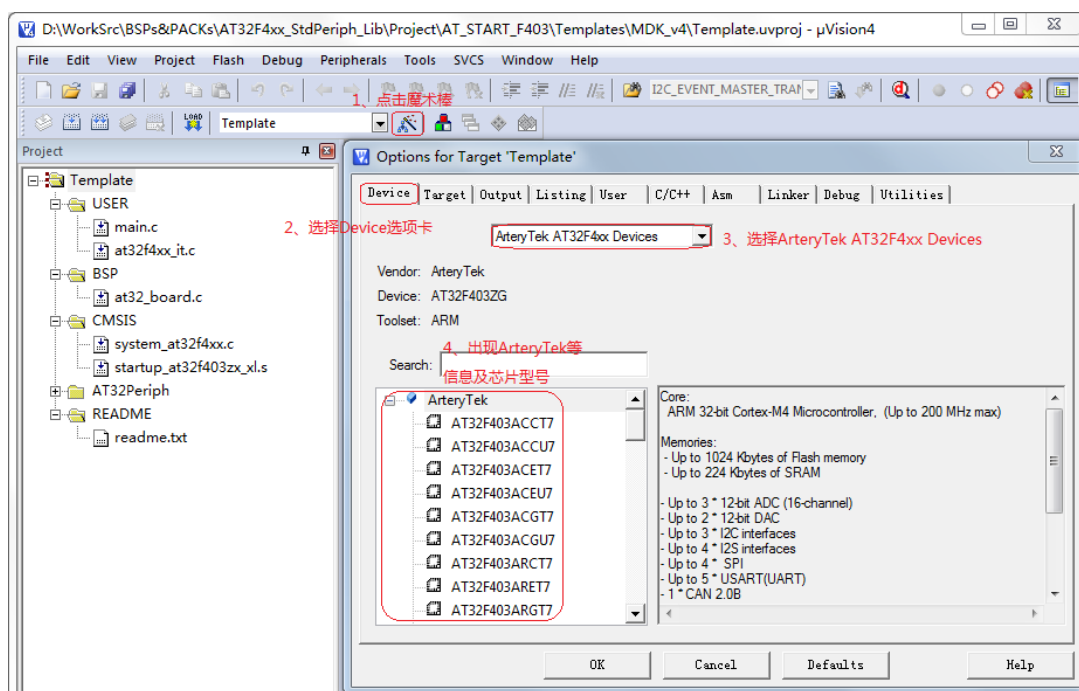
B、 如果Keil_v4的实际安装路径与“Destination Folder”对话框内的路径不一致, 点击“Browse”选择实际安装路径。然后点击“Next”, 弹出如下界面。



- C、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。

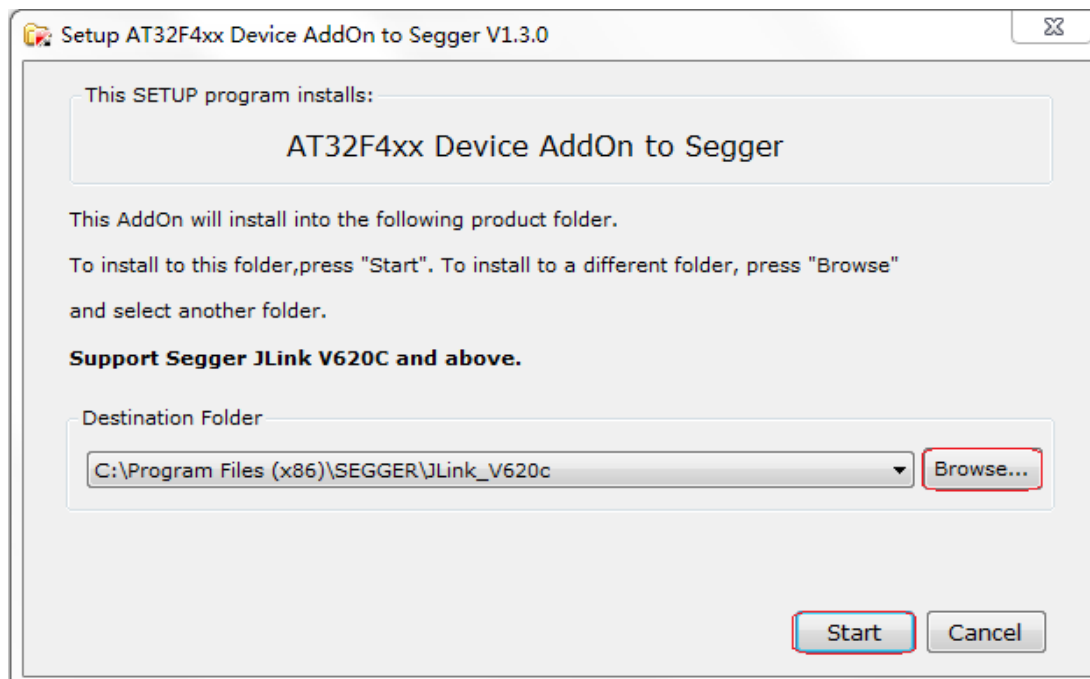


- D、点击“Finish”完成安装。查看Keil_v4 Pack安装是否成功。请按如下步骤进行操作和查看：

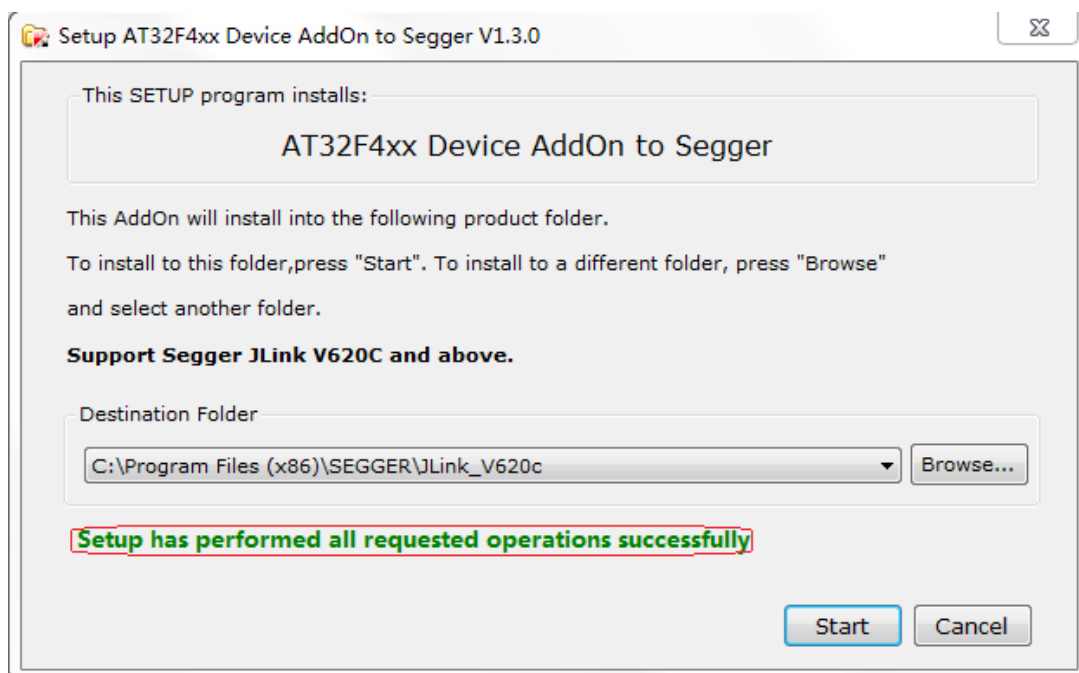


(4) Segger_AT32F4xx_AddOn: 支援J-Flash下载，安装步骤如下：

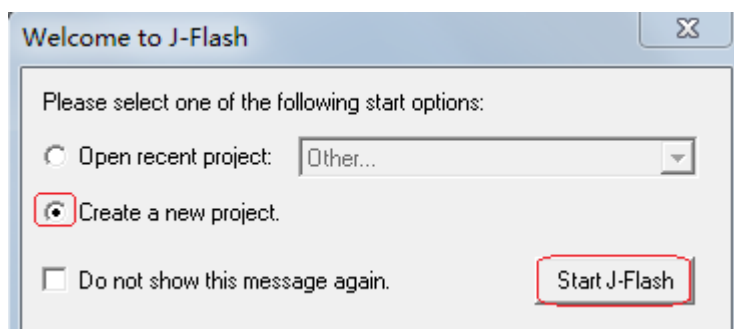
A、 双击Segger_AT32Fxx_AddOn.exe，弹出如下界面（具体版本信息按实际情况为准）。



B、 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。



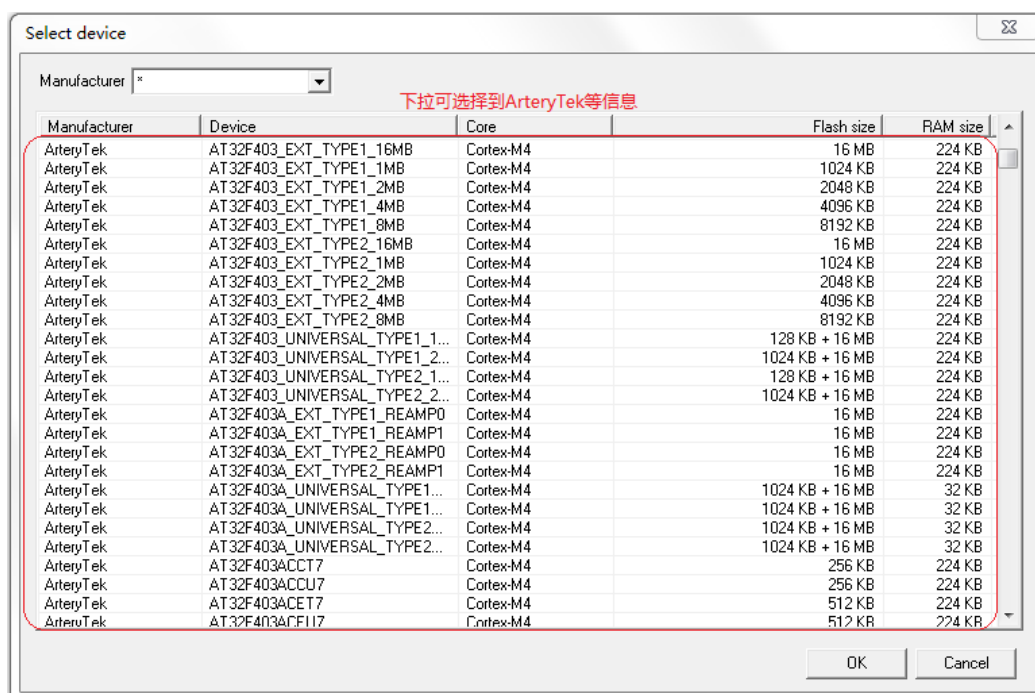
- C、出现“Setup has performed all requested operations successfully”则表示已安装成功。查看是否安装成功，请按如下步骤进行操作和查看：
- 1) 打开J-Flash.exe，出现如下对话框则选择Create a new project并点击Start J-Flash按钮，如下图：



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



- 3) 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：



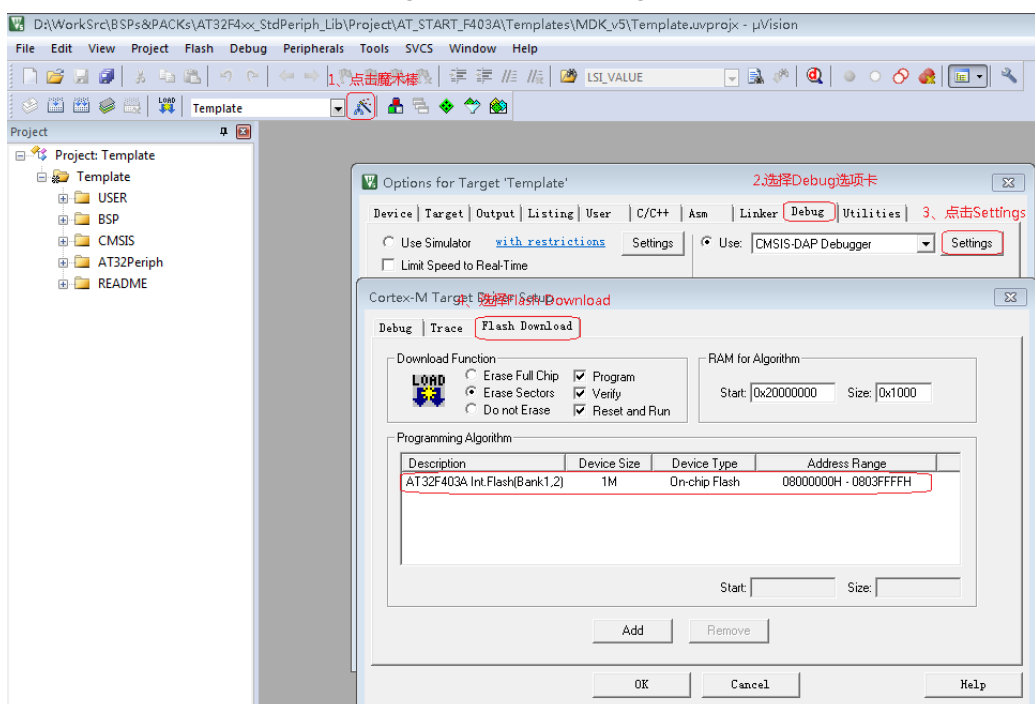
4 Flash算法文件说明

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。

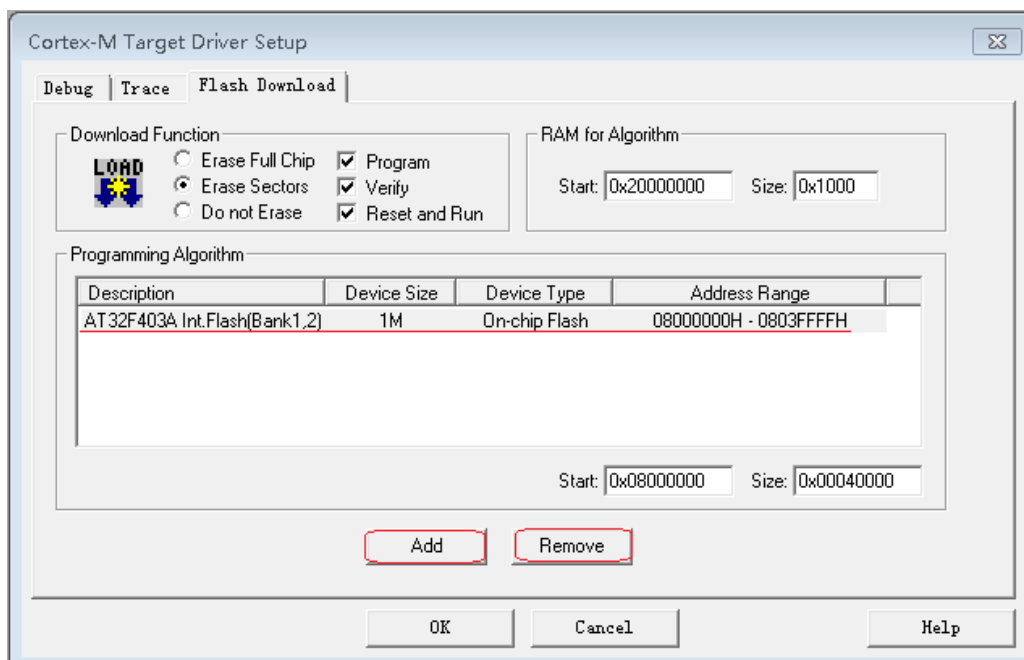
(1) Keil算法文件的使用方法

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。

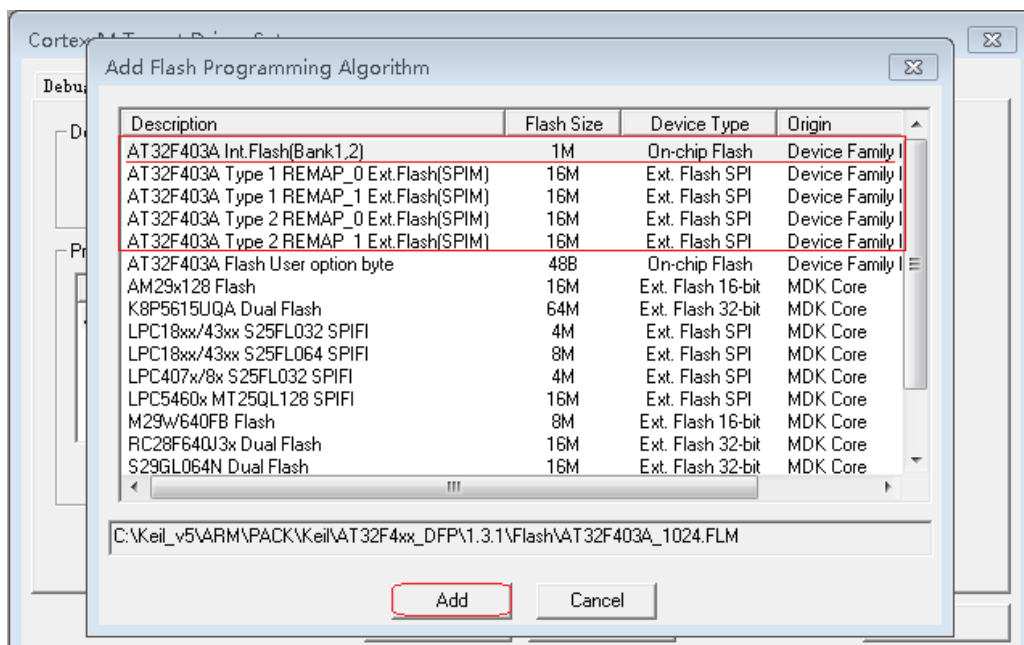
当在Keil IDE开发工具工程建立起来之后即可进行Debug方式配置和flash算法文件的选择。在开发工具内依次点击：配置魔术棒—>Debug选项卡—>Settings—>Flash Download，流程如下图：



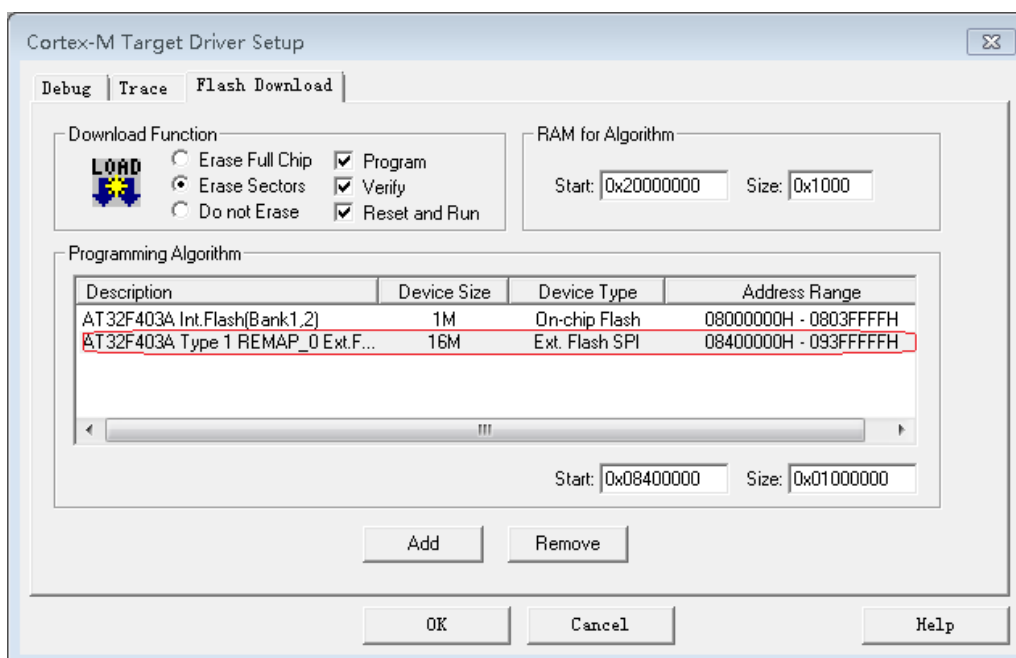
此处示例可看到所选择的Flash算法文件为默认的Flash算法文件，如需更改和移除可自行配置，点击到算法文件后可看到Add和Remove按钮可选择，如所选算法和实际MCU不匹配可使用以下方法重新配置



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：

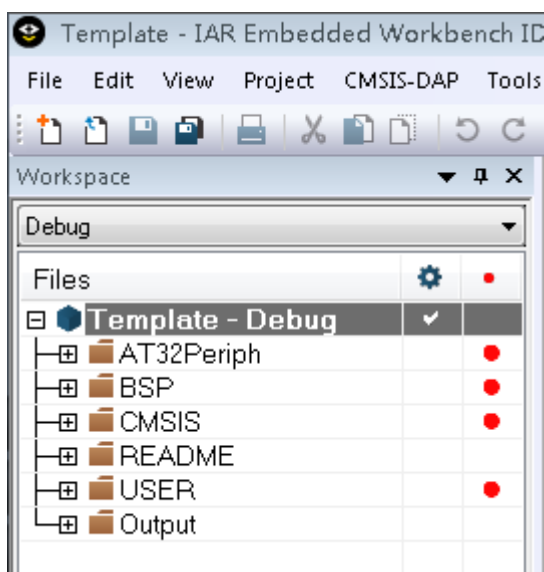


当选择到相应的算法文件后点击Add即可将新算法文件加入到当前工程配置，如下示例是新增SPIM算法到工程中：

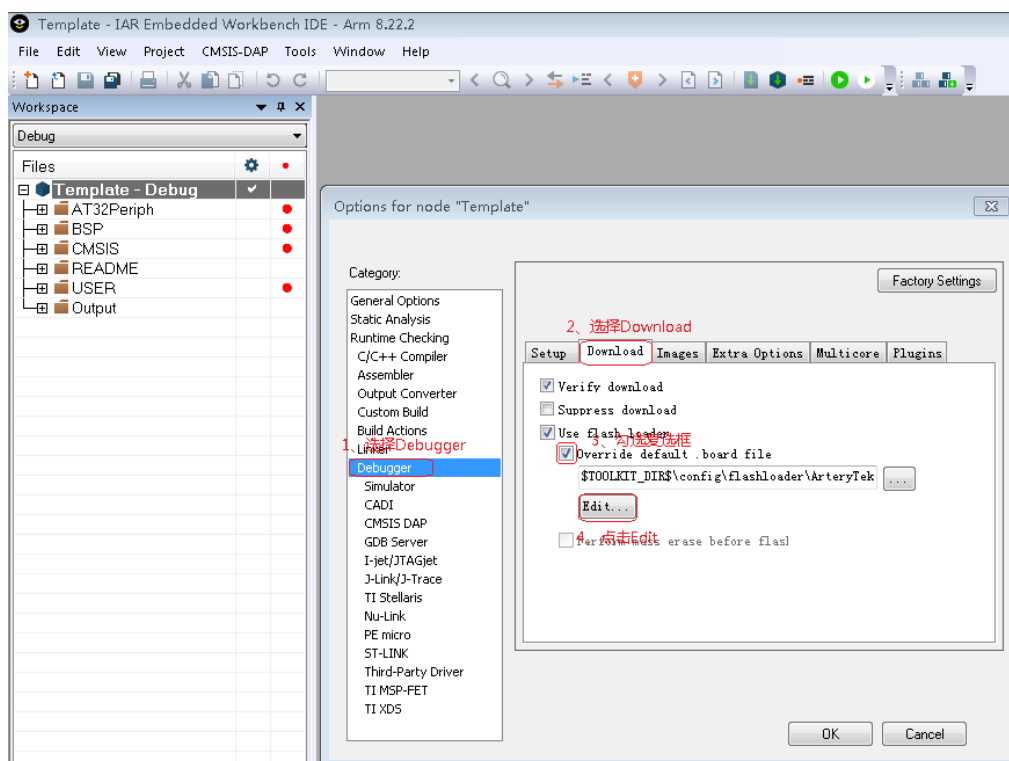


(2) IAR算法文件的使用方法

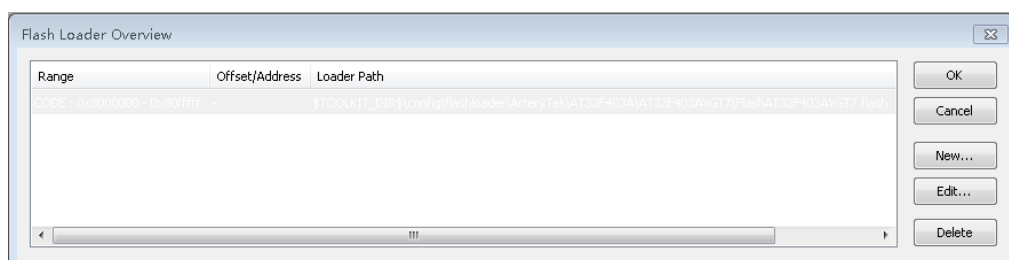
IAR开发环境对算法文件的选择方法是在当新建工程的配置中选定指定的MCU型号后自动选定的对应的默认flash算法文件。如需手动去进行算法文件配置，可在IAR工程建立起来之后，鼠标右击如下灰色选框位置的工程名：



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

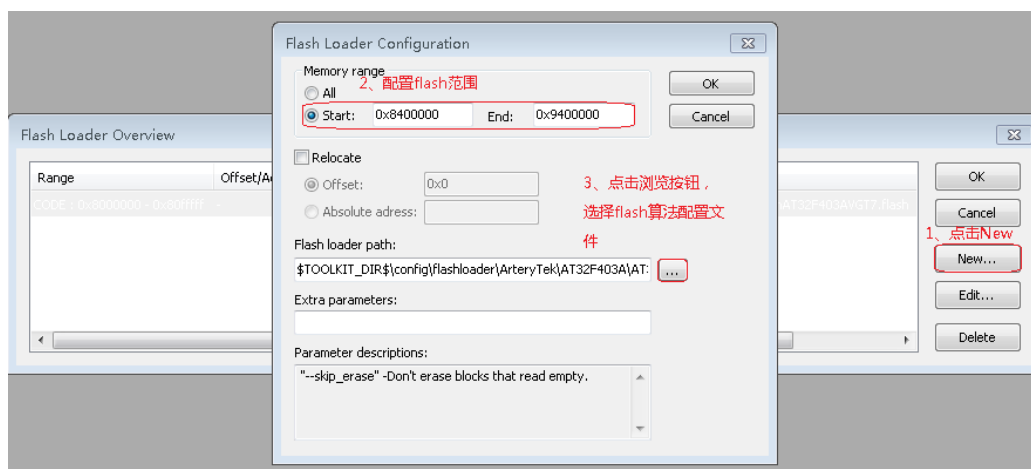


进入后可看见如下的配置界面：

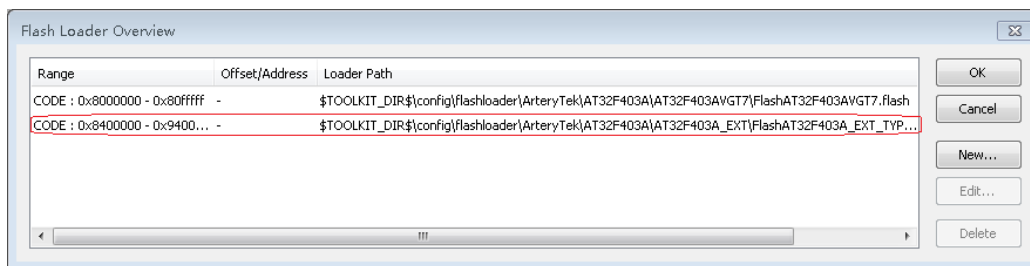


其中的flash算法配置方法是选定MCU芯片型号后默认指定，如需手动进行修改可点击旁边的New/Edit/Delete三个选项进行修改。

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：



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_AT32F4xx_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：



(3) SPIM算法说明

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：AT32F4xxType_NREMAP_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

日期	版本	变更
2018.12.17	1.0.0	最初版本
2019.07.22	1.0.1	修改一些描述和图示信息，方便多系列兼容使用此文档
2019.12.25	1.0.2	增加新系列AT32F403A&AT32F407，修改一些截图
2020.01.14	1.0.3	新加入 Segger_AT32F4xx_AddOn.exe 的描述，并完善 Pack 工具安装完成后是否安装成功的查看步骤。
2020.02.27	1.0.4	新增 Flash 算法文件描述章节，增加对 TS32F401CBU7 型号的支持

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。

© 2017 雅特力科技（重庆）有限公司 保留所有权利