AN0022
Application Note

Getting Started with AT32F415

Introduction

This application note is written to help users with rapid project development using AT32F415xx.

Note: The corresponding code in this application note is developed on the basis of V2.x.x BSP provided by Artery. For other versions of BSP, please pay attention to the differences in usage.

Applicable products:

Part number	AT32F415xx

Contents

List of tables

List of figures

1 Preliminary environment requirements

Download development environment

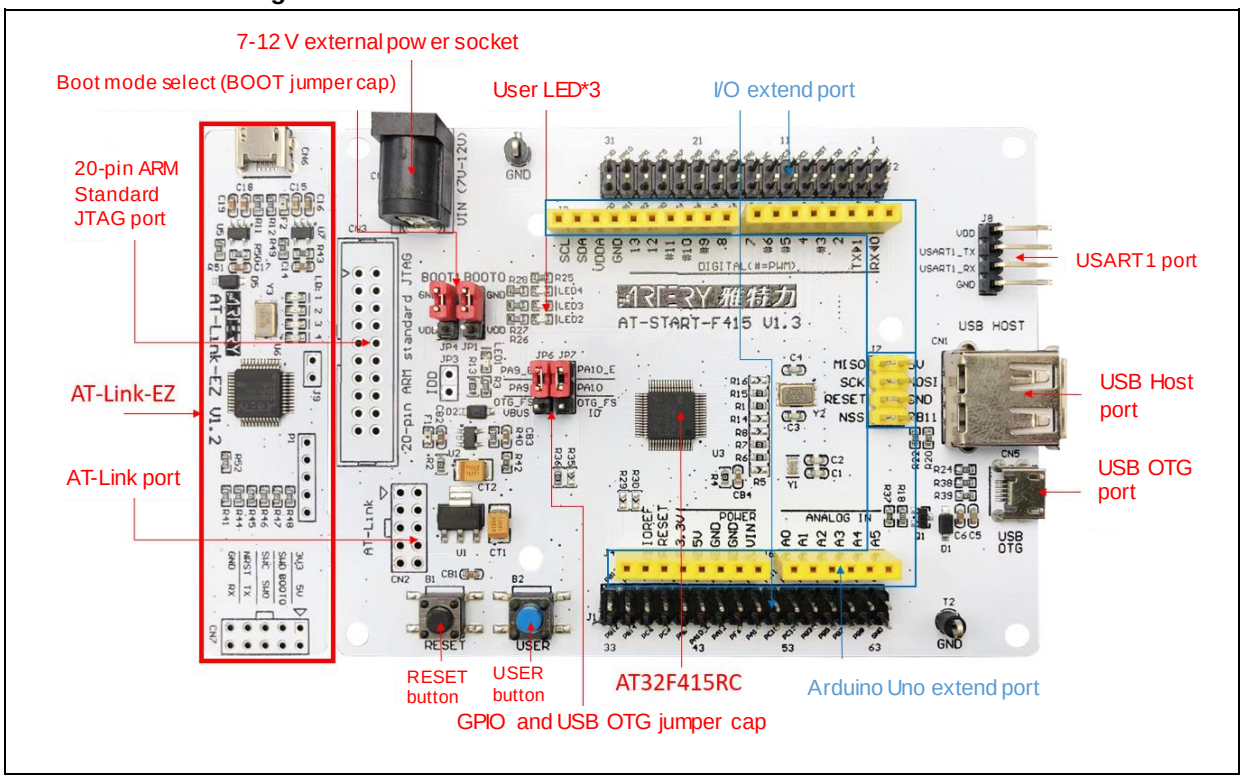
- ARTERY's official website

1.1 Build AT32 development environment

1.1.1 Debug tools and evaluation board

The AT32F415 evaluation board has an AT-Link-EZ debug tool, as shown in the red box in Figure 1 below. The AT-Link-EZ can be disassembled and used with other circuit boards, supporting IDE online debugging, online programming and USB-to-serial port.

Figure 1. AT-START-F415 evaluation board with AT-Link-EZ



Note: For details about resources for AT-START evaluation board, please refer to UM_AT_START_F415_Vx.x. Path: ARTERY's official website→PRODUCTS→Value line→AT32F4xx; download and unzip Evaluation Board package, and get the "\AT_START_F415_Vx.x\03_Documents".

Figure 2. AT-START-F415 evaluation board package

Evaluation Board

Download	Description	Version
AT-START-F415	AT32F415 evaluation board supporting Arduino standard interfaces	V1.3

1.1.2 Programming tools and software

- AT programming tools and software: AT-Link / AT-Link+ /AT-Link-Pro / AT-Link-ISO /AT-Link-

EZ, ICP/ISP.

- 3rd party programming tools: J-Link, Armfly, Alientek, XWOPEN, ICWORKSHOP, ZLG, MaxWiz, Amomcu, Acroview, Forcreat, Galecomm, Prosystems, Rx-prog, Sinaen, XELTEK, Zhifeng, etc.

Note: For more information, please visit ARTERY's official website→SUPPORT→Hardware Development Tool and 3RD Party Writer.

- For ICP usage instructions, please refer to *UM_ICP_Programmer*. Path: ARTERY's official website→PRODUCTS→Value line→AT32F4xx; download and unzip ICP tool, and get the "Artery_ICP_Programmer_Vx.x.xx\Document\UM_ICP_Programmer".
- For ISP usage instructions, please refer to *UM_ISP_Programmer*. Path: ARTERY's official website→PRODUCTS→Value line→AT32F4xx; download and unzip ISP tool, and get the "Artery_ISP_Programmer_Vx.x.xx\Document\UM_ISP_Programmer".
- For AT-Link usage instructions, please refer to *UM0004_AT-Link_User_Manual*. Path: ARTERY's official website →PRODUCTS →Value line →AT32F4xx; download and unzip AT-Link-Family, and get the "AT_Link_CH_ Vx.x.x\05_Documents\UM0004_AT-Link_User_Manual_ZH_Vx.x.x".

Figure 3. ICP/ISP/AT-Link-Family

Tool

Download	Description	Version
AT32 IDE_Linux AT32 IDE_Windows	支持AT32 MCU的基于Eclipse开发的跨平台ARM嵌入式系统的软件开发环境	V1.0.02
AT-Link Family	支持AT32 MCU 仿真与在线/离线烧录工具 (包含AT-Link-EZ/AT-Link/AT-Link-Pro/AT-Link-ISO四种工具)	V2.1.1
AT-Link Console	支持AT32 MCU「在电路编程」Console工具	V3.0.02
ICP	支持AT32 MCU「在电路编程」工具	V3.0.05
ISP	支持AT32 MCU「在系统编程」工具	V2.0.06
ISP Multi-Port	支持AT32 MCU一对多设备「在系统编程」工具	V2.0.06

1.1.3 AT32 development environment

1.1.3.1 Template projects

The commonly used IDE template projects are available from BSP provided by ArteryTek. BSP path: ARTERY's official website→PRODUCTS→Value line→AT32F4xx.

Figure 4. BSP package

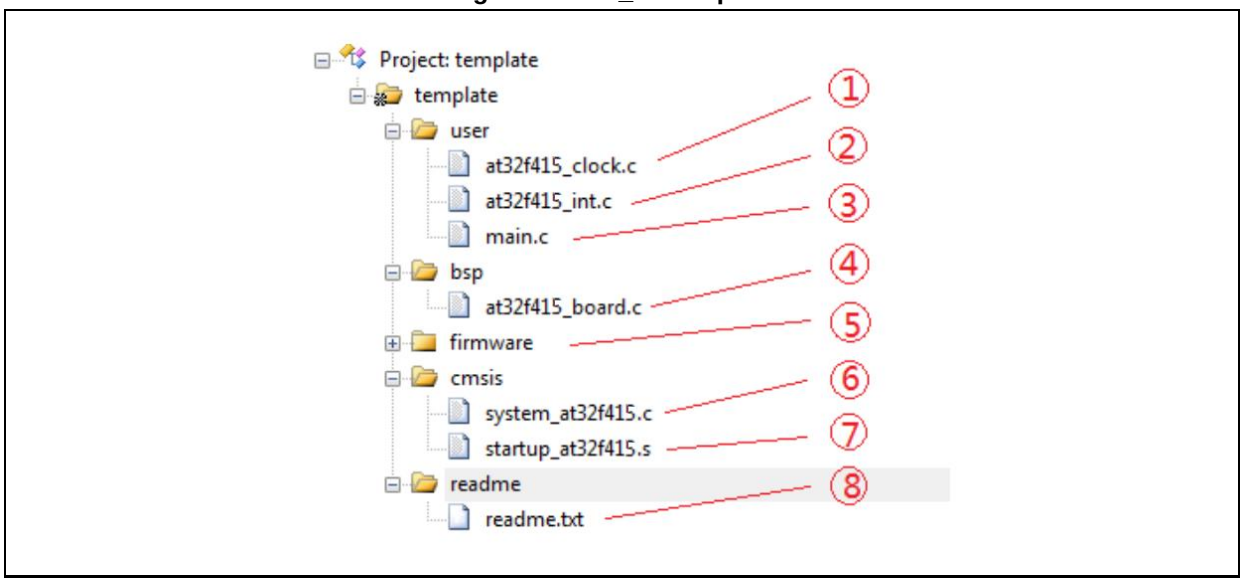
BSP

Download	Description	Version
Firmware Library	AT32F415 firmware library BSP user guide	V2.0.9

The template projects of Keil_v5/Keil_v4/IAR_6.10/IAR_7.4/IAR_8.2/eclipse_gcc/at32_ide are

created in BSP. Path: AT32F415_Firmware_Library_V2.x.x\project\at_start_f4xx\templates. Open the project folder and click on the project file to open the corresponding IDE project. The example of Keil_v5 template project is shown below.

Figure 5. Keil_v5 templates



Contents in the project:

① at32f415_clock.c: clock configuration file, contains default clock frequency and clock path;
② at32f415_int.c: interrupt file, part of core interrupt function code flow is written by default;
③ main.c: main code file of the template project;
④ at32f415_board.c: board-level configuration file, contains settings of AT-START on-board buttons and LEDs;
⑤ at32f415_xx.c under *firmware*: driver file of on-chip peripherals;
⑥ system_at32f415.c: system initialization file;
⑦ startup_at32f415.s: startup file;
⑧ readme.txt: project documentation, contains application functions, setting methods and associated application notes (ApNote) of the template project.

Except for templates, BSP also includes code examples (Keil_v5 project files) in terms of peripherals for reference (AT32F415_Firmware_Library_V2.x.x\project\at_start_f4xx\examples).

Note: For more details about BSP, please refer to "Section 4 BSP application" of AT32F415 Firmware BSP&Pack User Guide. Path: ARTERY's official website→PRODUCTS→Value line→AT32F4xx; download and unzip BSP, and get the "\AT32F415_Firmware_Library_Vx.x.x\document".

1.1.3.2 Pack installation

Install Pack and add the AT32 MCU part number to Keil/IAR. You can download Pack from ARTERY's official website→PRODUCTS→Value line→AT32F4xx.

Figure 6. Pack download

Pack

Download	Description	Version
Keil 4 Keil 5	Supports AT32 MCU to run in Keil MDK	V2.2.0 V2.2.3
IAR	Supports AT32 MCU to run in IAR EWARM	V2.1.6
Segger	Supports Segger tools to identify AT32 MCU	V2.0.7

For Keil compiling system, keil 4.74 /5.23 or above is recommended. If Keil_v5 is used, please unzip Keil5_AT32MCU_AddOn and install the corresponding ArteryTek.AT32F415_DFP. If Keil_v4 is used, please install Keil4_AT32MCU_AddOn. By default, the Keil installation path can be recognized automatically during installation. If the path is not recognized or incorrect, you need to manually select the Keil installation path.

Figure 7. Set up ArteryTek.AT32F415 _DFP

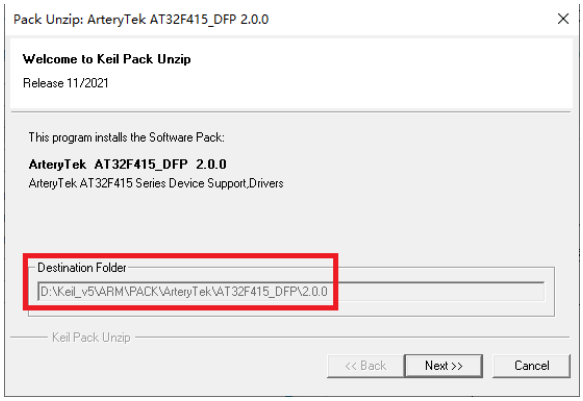
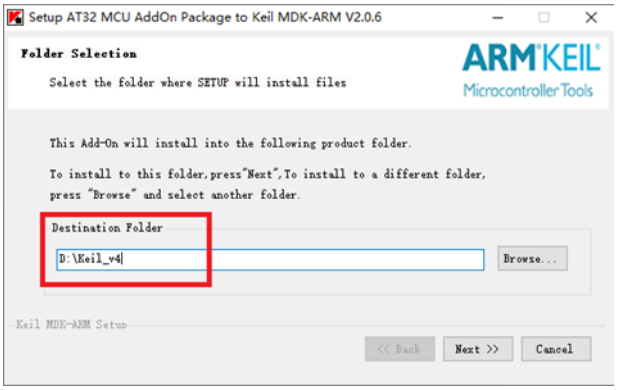
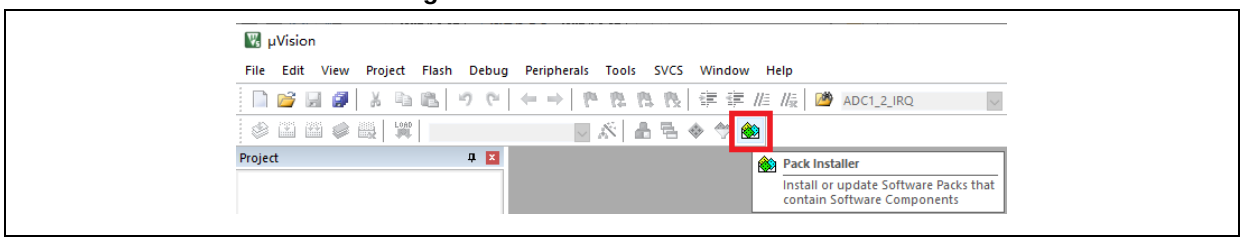


Figure 8. Set up Keil4_AT32MCU_AddOn



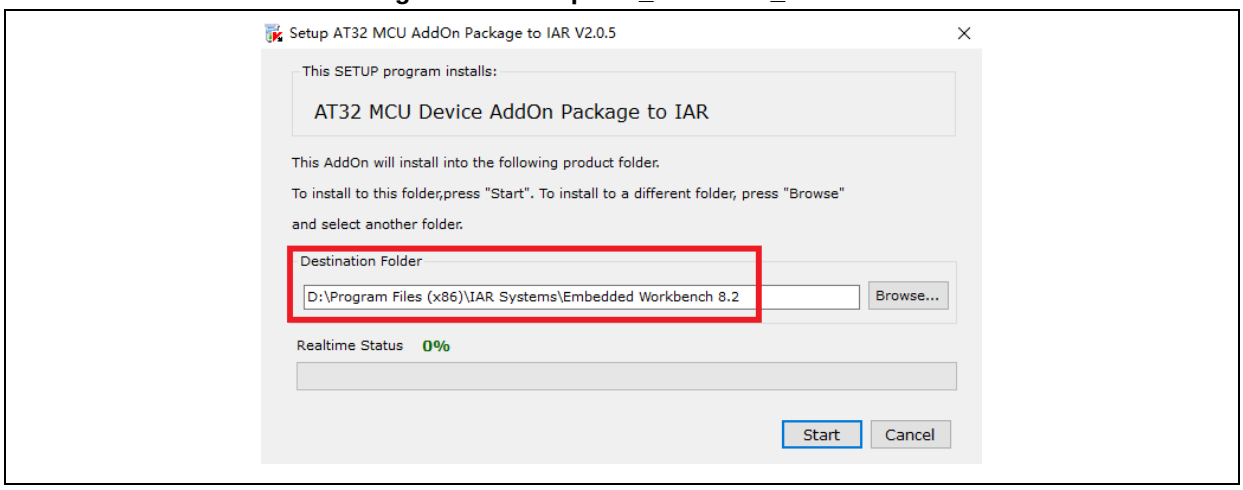
You can also open keil and click on “Pack Installer” icon; then click on the top left “file” and select “import” to import the corresponding pack downloaded from ARTERY's official website.

Figure 9. Pack Installer icon in Keil



For IAR compiling system, IAR7.0 or IAR6.1 above is recommended. It is necessary to install IAR_AT32MCU_AddOn. By default, the IAR installation path can be recognized automatically during installation. If the path is not recognized or incorrect, you need to manually select the IAR installation path.

Figure 10. Set up IAR_AT32MCU_AddOn

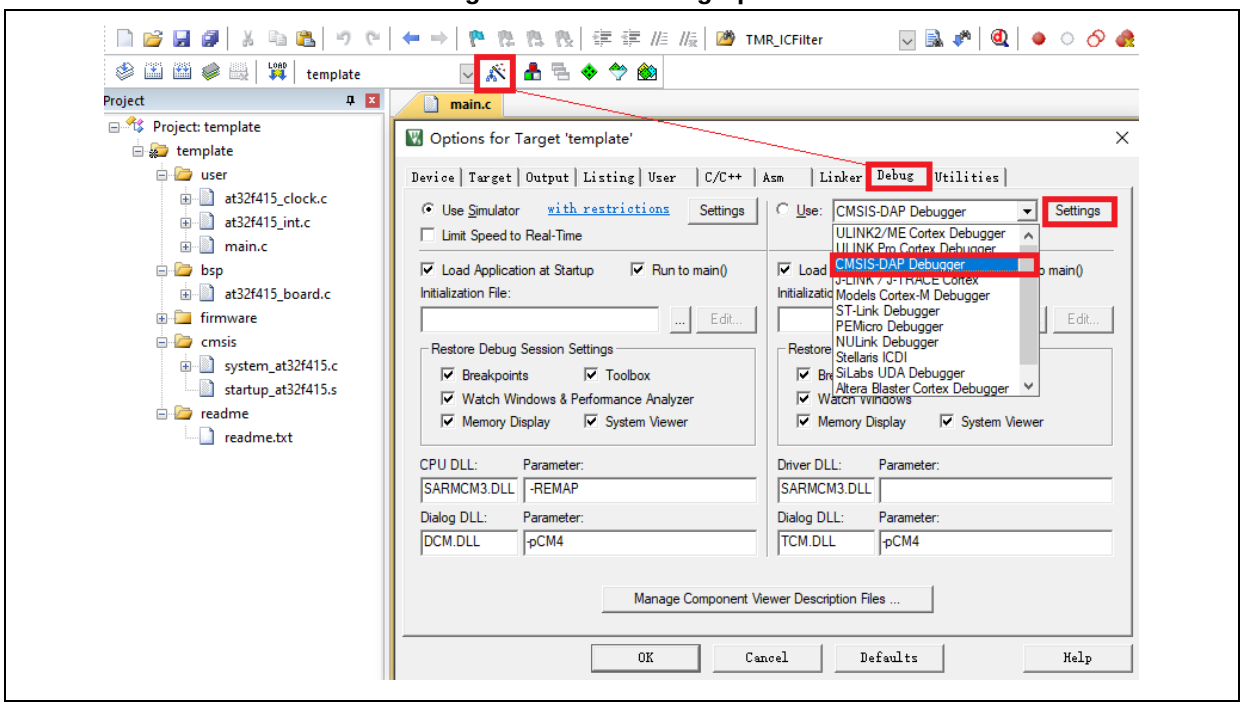


Note: For more details about Pack setup, please refer to "Section 2 Pack setup" of AT32F415 Firmware BSP&Pack User Guide. Path: ARTERY's official website→PRODUCTS→Value line→AT32F4xx; download and unzip BSP, and get the "\AT32F415_Firmware_Library_Vx.x.x\document".

1.1.3.3 Use AT-Link for debug and download

If you want to use AT-Link in IAR, select CMSIS-DAP in Debugger option.

Figure 11. Keil Debug option



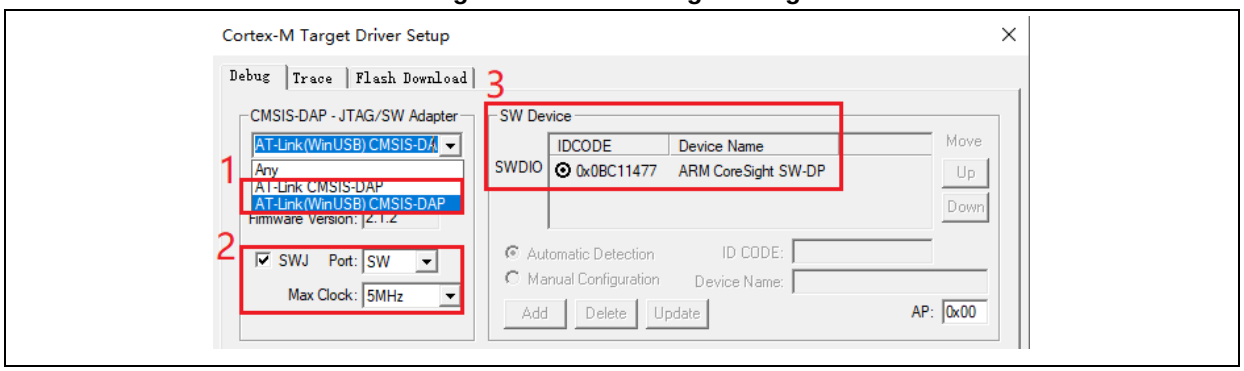
Go to Debug and click on Settings to enter the Cortex-M Target Driver Setup interface.

1. Select AT-Link(WinUSB)-CMSIS-DAP/AT-Link-CMSIS-DAP;

Note: For details about WinUSB, please refer to FAQ0136_How to use AT-LINK WinUSB to improve download speed (ARTERY's official website→SUPPORT→FAQ→FAQ0136).

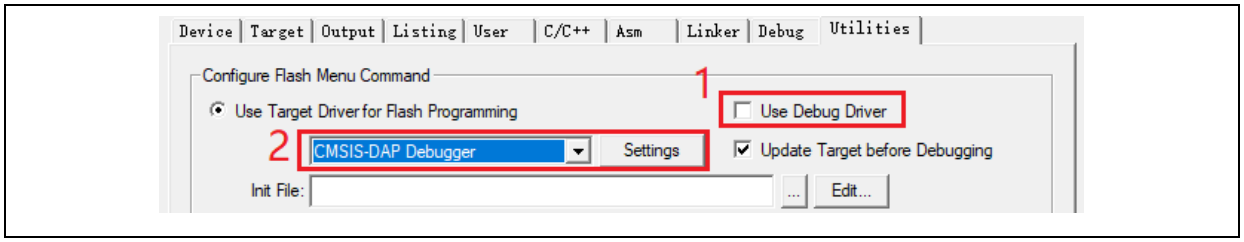
2. Find Port, select SW and then tick SWJ;
3. Confirm that the ARM SW-DP debug module is recognized.

Figure 12. Keil Debug Settings



Click on Utilities and untick option box 1; then select CMSIS-DAP Debugger in option box 2, and finally tick option box 1 (it should be unticked first and then ticked).

Figure 13. Keil Utilities



If you want to use AT-Link in IAR, please click on Project and select Options; then select CMSIS-DAP in Debugger option, and select SWD in CMSIS DAP option.

Figure 14. IAR Debug option

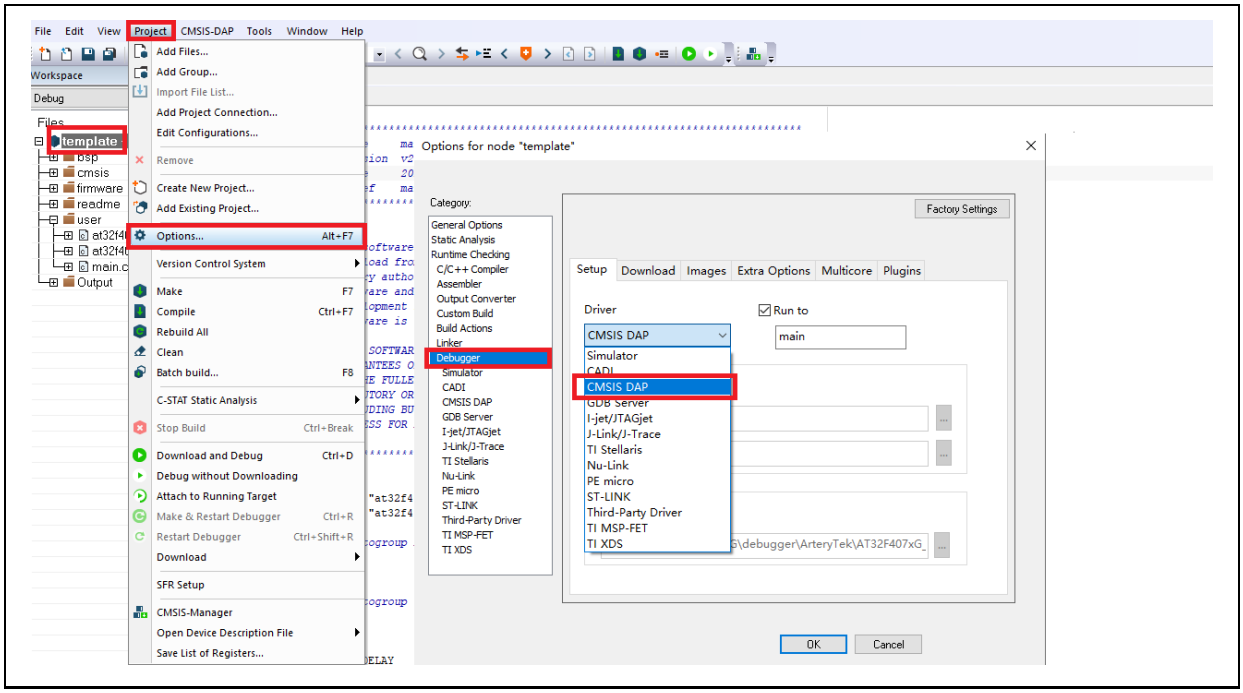
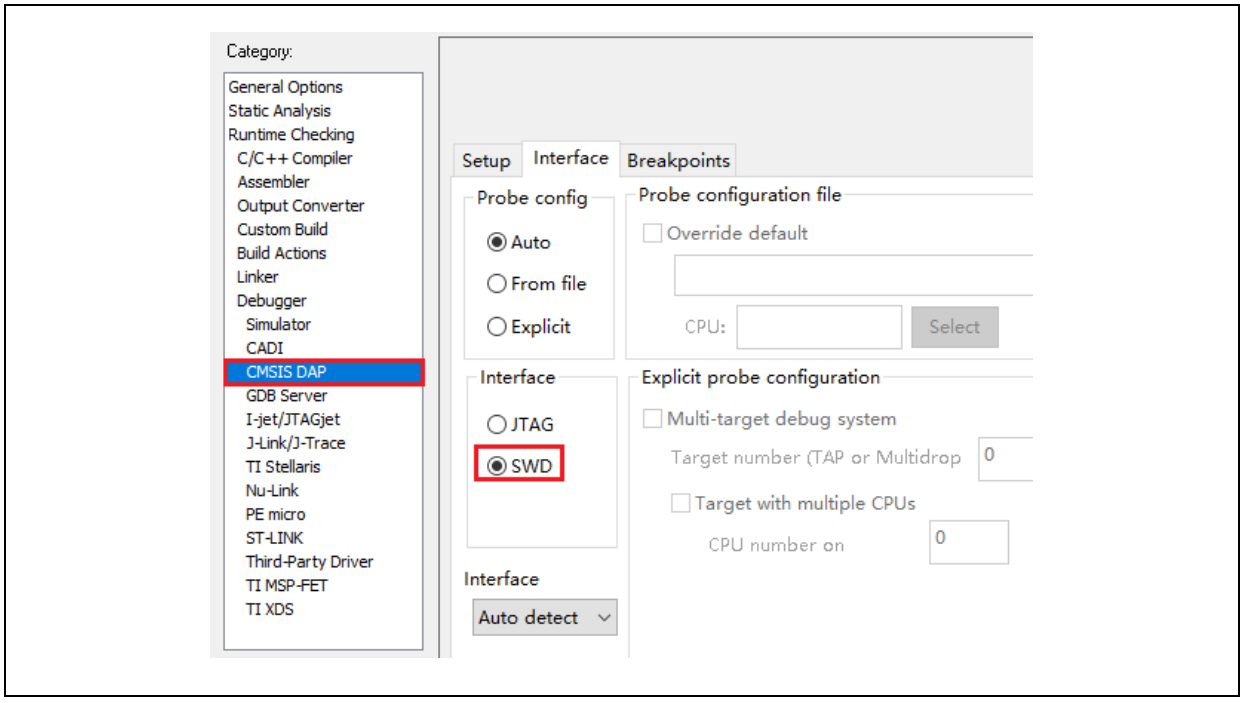


Figure 15. IAR CMSIS-DAP option



Note: For details about Flash algorithm file, MCU switch and solutions for "J-Link cannot find MCU", please refer to AT32F415 Firmware BSP&Pack User Guide. Path: ARTERY's official website→PRODUCTS→Value line→AT32F4xx; download and unzip BSP, and get the "\AT32F415_Firmware_Library_Vx.x.x\document".

1.1.4 How to replace SXX

- Please refer to *MG0005_ Migrating from SXX32F0xx&GX32F3x0 to AT32F415* (ARTERY's

official website→PRODUCTS→Value line→AT32F4xx);

- If the program still cannot run properly after completing the above steps, please refer to other sections of this application note or contact the agent and ARTERY technicians for help.

Note: Compared with SXX32F0xx, AT32F415 is more flexible to achieve better performance. Please refer to AN0004_Performance_Optimization (ARTERY's official website→SUPPORT→AP Note→AN0004) to learn how to promote performance of AT32F415.

1.2 AT32F415 chip enhanced functions

1.2.1 Prefetch buffer

When the prefetch buffer is set, the CPU can execute operations faster. For example, while the CPU is reading one byte, the next byte is waiting in the prefetch buffer. The Prefetch controller decides whether to access the Flash according to the available space in the prefetch buffer. When there is at least one available block in the prefetch buffer, the prefetch controller starts a read operation.

The user needs to set wait states according to different system clock frequencies by setting bit 2:0 of the FLASH_PSR register (corresponds to the FLASH_ACR register of SXX32).

Figure 16. Wait states of Flash performance select register (FLASH_PSR)

Bit	Abbr.	Reset value	Type	Description
Bit 2: 0	WTCYC	0x0	rw	Wait states The wait states depends on the size of the system clock, and they are in terms of system clocks. 0: Zero wait state when 0MHz<system clock≤32MHz 1: One wait state when 32MHz<system clock≤64MHz 2: Two wait states when 64MHz<system clock≤96MHz 3: Three wait states when 96MHz<system clock≤128MHz 4: Four wait states when 128MHz<system clock≤150MHz

AT library has corresponding settings in the system clock configuration function "system_clock_config()". For other BSP versions, please find out the correct position and complete the corresponding settings.

Figure 17. System clock configuration function "system_clock_config"

```
void system_clock_config(void)
{
  /* config flash psr register */
  flash_psr_set(FLASH_WAIT_CYCLE_4);

  /* reset crm */
  crm_reset();

  crm_clock_source_enable(CRM_CLOCK_SOURCE_HEXT, TRUE);

  /* wait till hext is ready */
  while(crm_hext_stable_wait() == ERROR)
  {
  }
```

Note: When the AHB clock prescaler is not equal to 1, the Flash prefetch buffer (FLASH_PSR[4]=0x1) must be enabled.

1.2.2 PLL clock settings

1.2.2.1 PLL setting methods

The embedded PLL clock of AT32F415 has a maximum output frequency of 150 MHz, and it can be configured by using the CRM_CFG register (corresponds to the RCC_CFGR register of SXX32) or the additional CRM_PLL register. Users can use the CRM_PLL register to configure more PLL clock frequencies, and the formula is shown below:

$$\text{PLL output clock} = \text{PLL ref. input clock} \times \frac{\text{PLL multiplication factor (PLL\_NS)}}{\text{PLL pre-division factor (PLL\_MS)} \times \text{PLL post-division factor (PLL\_FR)}}$$

When the SXX32F0xx BSP is used, the PLL setting example (HEXT=8 MHz, PLL=48 MHz):

```
RCC->CFGR |= (uint32_t)(RCC_CFGR_PLLSRC_HSE | RCC_CFGR_PLLMULL6);
```

If the user wants to use SXX32F0xx program to output a clock greater than 48 MHz on AT32F415, it is necessary to configure the CRM_CFG register (corresponds to the RCC_CFGR register of SXX32).

When PLL=144 MHz, the corresponding setting is as follows:

```
RCC->CFGR |= (uint32_t)(RCC_CFGR_PLLSRC_HSE | RCC_CFGR_PLLMULL18 );
```

When the AT BSP is used, the PLL setting example (HEXT=8 MHz, PLL=48 MHz):

```
crm_pll_config(CRM_PLL_SOURCE_HEXT, CRM_PLL_MULT_6);
```

When the AT BSP is used, the PLL setting example (HEXT=8 MHz, PLL=144 MHz):

```
crm_pll_config(CRM_PLL_SOURCE_HEXT, CRM_PLL_MULT_18);
```

The user also can use the additional CRM_PLL register of AT32 MCU to configure more clock frequencies. For example, if the AT32F415 BSP is used, the PLL setting example (HEXT=8 MHz, PLL=150 MHz):

Figure 18. AT32F415 150 MHz PLL clock configuration

```
#define  CRM_PLL_NS  ((uint16_t)0x4B)  /* PLL_NS=75 */
#define  CRM_PLL_MS  ((uint16_t)0x01)  /* PLL_MS=1 */
/* config pll clock resource    PLL_FR =4*/
crm_pll_config2(CRM_PLL_SOURCE_HEXT, CRM_PLL_NS, CRM_PLL_MS, CRM_PLL_FR_4);
```

Where, the parameter CRM_PLL_SOURCE_HEXT represents that the HEXT is used as external clock source; PLL_NS=75, PLL_MS=1, and CRM_PLL_FR_4 (0x02, divided by four) represents the PLL_FR value.

Please refer to *AN0117_AT32F415_CRM_Start_Guide* (ARTERY's official website→SUPPORT →AP Note→AN0117) for more details about AT32F415 clock source settings and modification and learn how to use New Clock Configuration (ARTERY's official website→PRODUCTS→Value line→AT32F4xx) to generate the desired clock code and apply it to the project.

1.2.2.2 Auto step-by-step switch

When the internal PLL of AT32F415 is set to 108 MHz and above, it is necessary to perform auto step-by-step switch.

When the SXX32F0xx BSP is used, the user needs to open system_sxx32f0xx.c and find out the current system clock configuration function (go through Section 1.2.2.1 PLL settings), and add the following codes in Italic black to the *static void SetSysClockToxxM(void)* function.

Figure 19. SXX PLL auto step-by-step switch configurations

```
/* Wait till PLL is ready */
while((RCC->CR & RCC_CR_PLLRDY) == 0)
{
}
*((unsigned int *)0x40021054) |= (0x30);  // Enable auto step-by-step clock switch function
/* Select PLL as system clock source */
RCC->CFGR &= (uint32_t)((uint32_t)~(RCC_CFGR_SW));
RCC->CFGR |= (uint32_t)RCC_CFGR_SW_PLL;
/* Wait till PLL is used as system clock source */
while ((RCC->CFGR & (uint32_t)RCC_CFGR_SWS) != (uint32_t)0x08)
{
}
*((unsigned int *)0x40021054) &=~ (0x30);  // Disable auto step-by-step clock switch function
```

When the AT32F415 BSP is used, the example of PLL auto step-by-step switch is shown below:

Figure 20. AT32 PLL auto step-by-step switch configurations

```
/* enable auto step mode */
crm_auto_step_mode_enable(TRUE);
/* select pll as system clock source */
crm_sysclk_switch(CRM_SCLK_PLL);
/* wait till pll is used as system clock source */
while(crm_sysclk_switch_status_get() != CRM_SCLK_PLL)
{
}
/* disable auto step mode */
crm_auto_step_mode_enable(FALSE);
/* update system_core_clock global variable */
system_core_clock_update();
```

Note: If the auto step-by-step system clock switch is enabled, it must be disabled after clock switch.

1.2.3 Encryption mode

1.2.3.1 Access protection

The access protection is commonly known as “encryption”, which is applied to the entire Flash storage area. Once the Flash access protection is enabled, the embedded Flash storage area can only be read through normal execution of the program, not through JTAG or SWD. When ICP/ISP tool is used to disable the access protection, the chip will perform erase operation on the Flash.

Note: Once enabled, the high-level access protection cannot be disabled, and the user is not allowed to re-erase or program the system data area in any way.

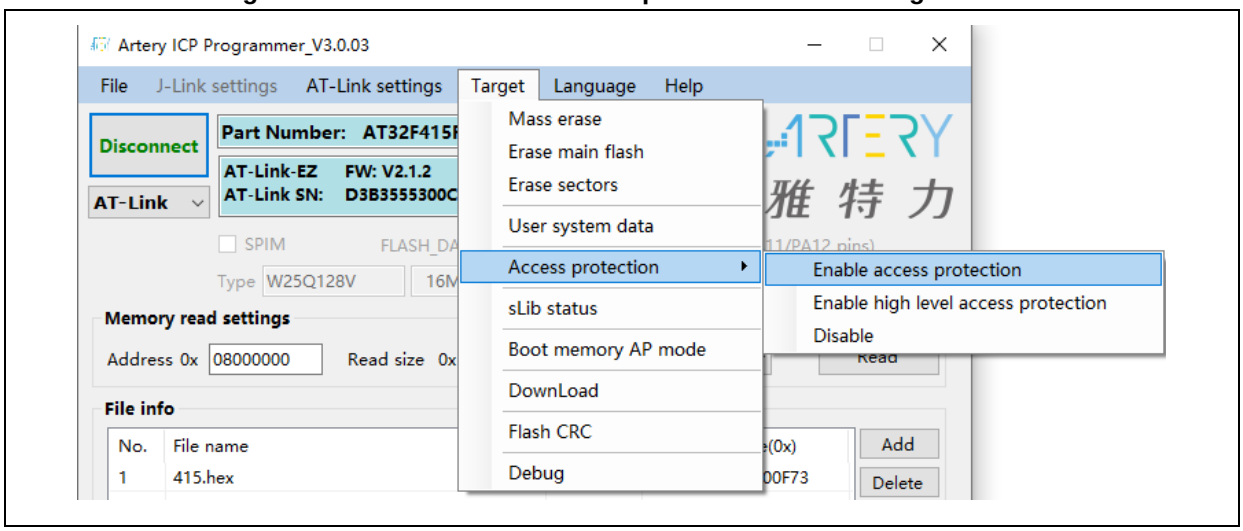
The ICP/ISP tool can be used to enable and disable IC access protection, as shown below:

- Artery ICP Programmer (BOOT0=0, BOOT1=0)

  Enable access protection: Open Artery ICP Programmer—Access protection—Enable high-level access protection.

  Disable access protection: Open Artery ICP Programmer—Access protection—Disable.

Figure 21. Enable/disable access protection in ICP Programmer



- Artery ISP Programmer (BOOT0=1, BOOT1=0)

  Enable access protection: Click on “Next” until enter the final interface, then select “Protection/Enable/Access protection” —Next—Yes, encrypted.

  Disable access protection: Protection/Disable/Access protection—Next—Yes, Flash decrypted.

Figure 22. Enable access protection in ISP Programmer

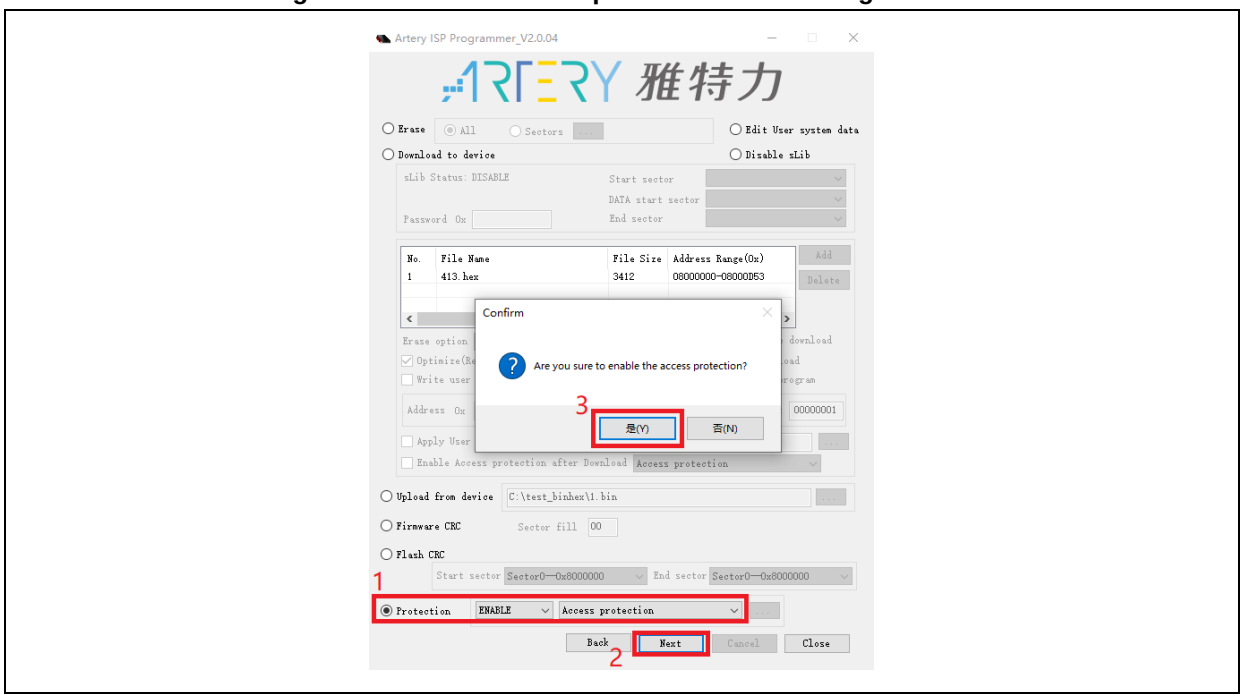
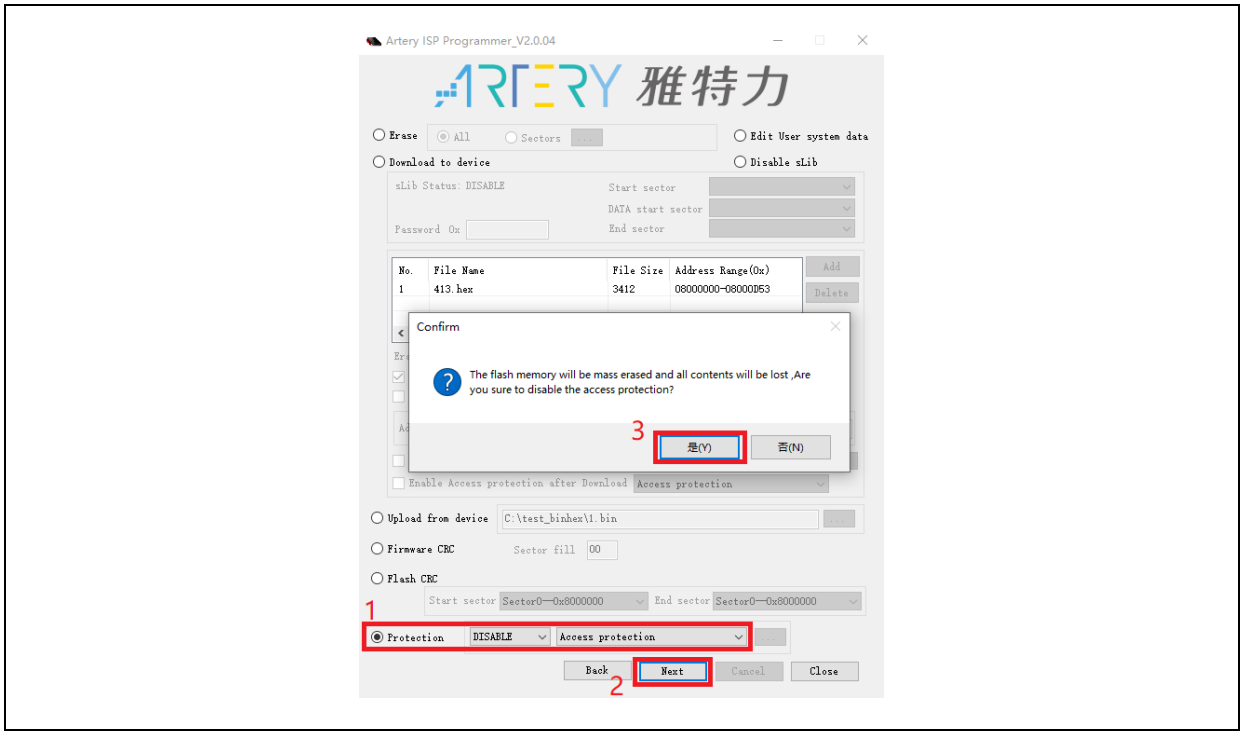


Figure 23. Disable access protection in ISP Programmer



■ Artery ISP Multi-Port Programmer (BOOT0=1, BOOT1=0)

Enable access protection: Protection/Enable/High-level access protection—Start, encrypted;

Disable access protection: Protection/Disable/Access protection—Start, Flash decrypted.

Note: Once enabled, the access protection cannot be disabled through erase operation.

1.2.3.2 Erase and program protection

The program protection is applied to the entire Flash storage area or certain pages in the Flash storage area. Once the Flash program protection is enabled, the internal Flash storage area cannot be programmed in any way.

The user can use ICP/ISP programmer to enable/disable erase and program protection, as shown below:

- Artery ICP Programmer (BOOT0=0, BOOT1=0)

  Enable erase and program protection: Open Artery ICP Programmer—User system data—Tick the sectors that require erase and program protection—Apply to device.

  Disable erase and program protection: Open Artery ICP Programmer—User system data—Untick the sectors that do not require erase and program protection—Apply to device.

- Artery ISP Programmer (BOOT0=1, BOOT1=0)

  Enable erase and program protection: Protection/Enable/Erase and program protection—Next—Yes, erase and program protection enabled.

  Disable erase and program protection: Protection/Disable/Erase and program protection—Next—Yes, erase and program protection disabled.

- Artery ISP Multi-Port Programmer (BOOT0=1, BOOT1=0)

  Enable erase and program protection: Protection/Enable/Erase and program protection—Start—Yes, enabled.

  Disable erase and program protection: Protection/Disable/Erase and program protection—Start—Yes, disabled.

Figure 24. Enable erase and program protection in ICP Programmer

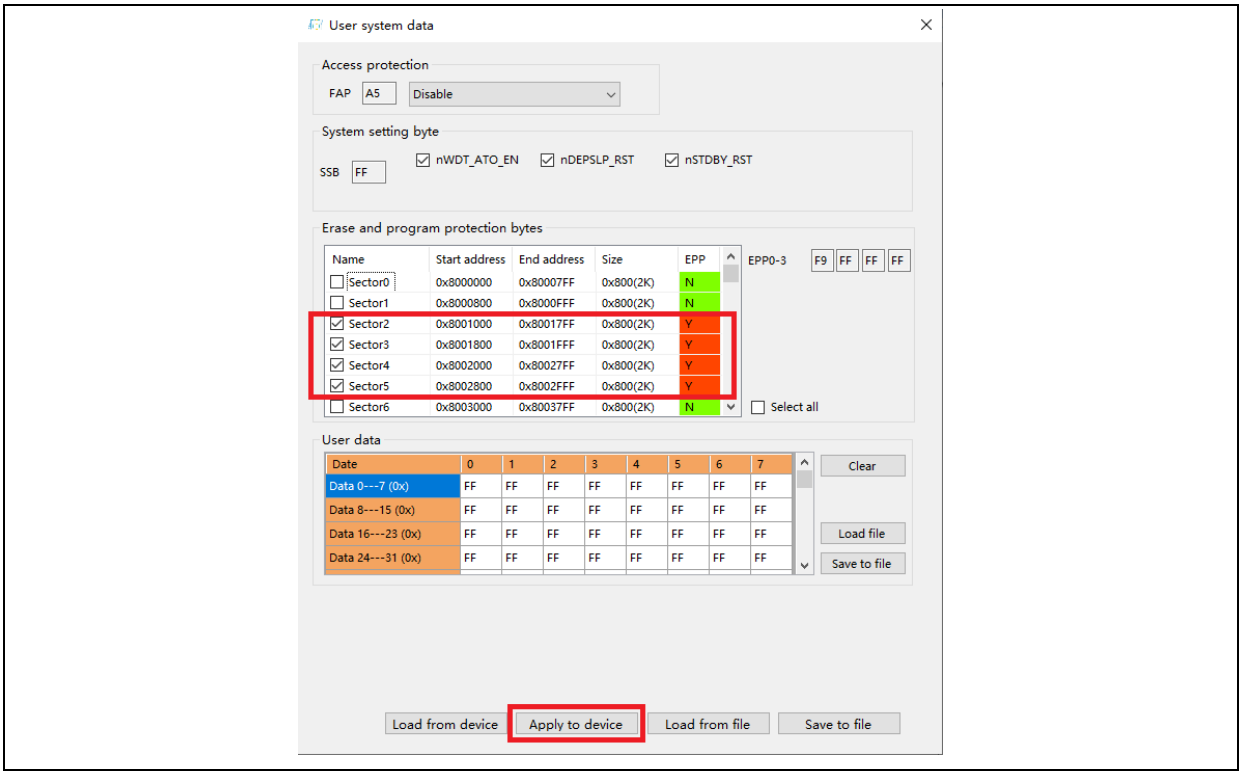
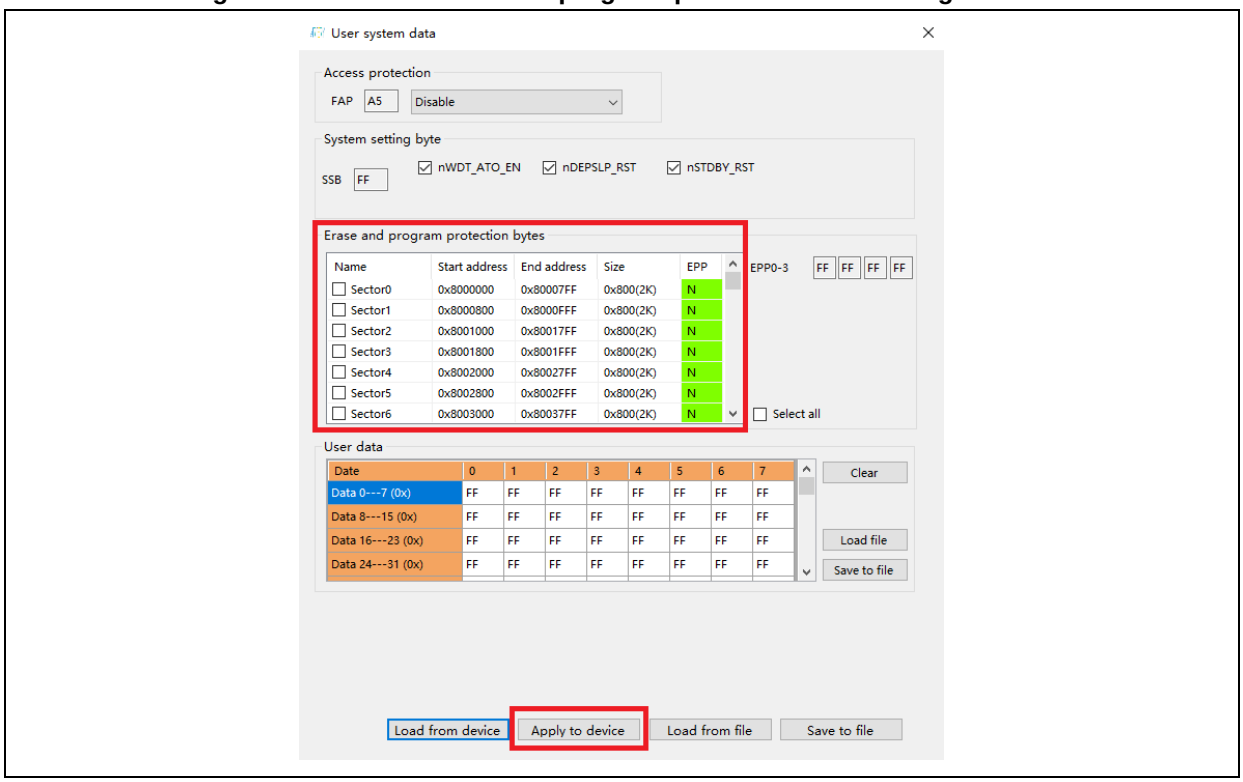


Figure 25. Disable erase and program protection in ICP Programmer



Note: Once enabled, the erase and program protection cannot be disabled through erase operation.

1.2.4 Set boot memory as main Flash memory extension area

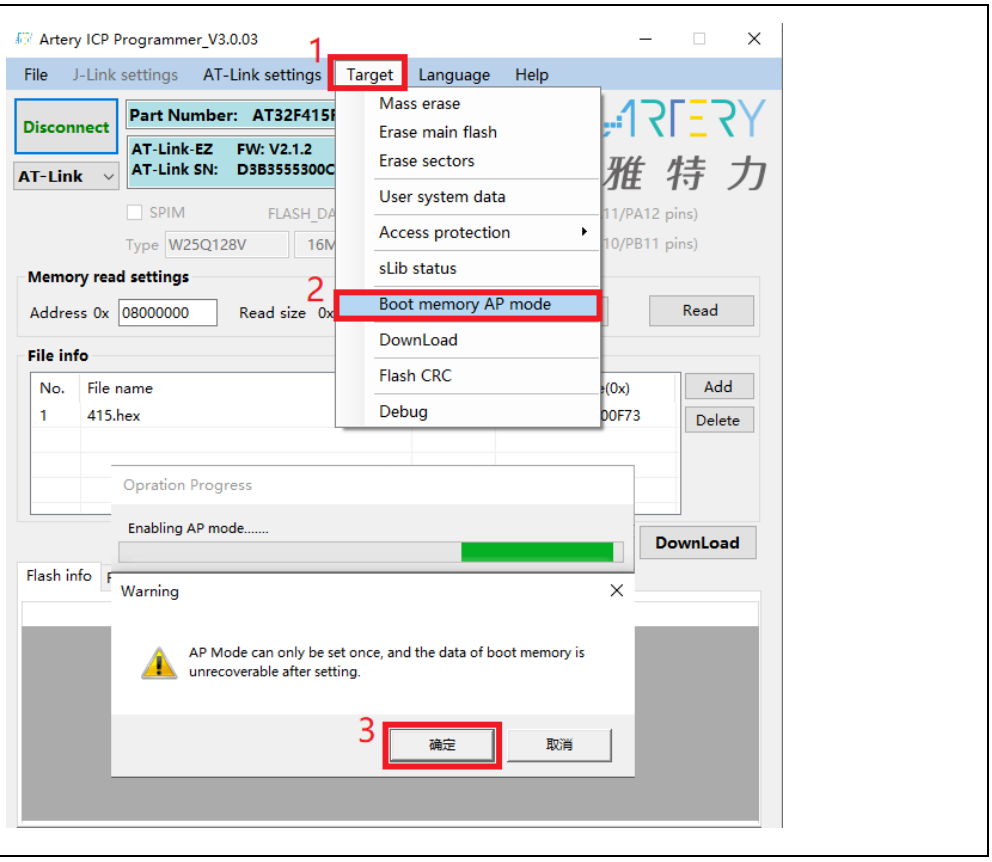
By default, boot memory stores the original boot codes in BOOT mode. For AT32F415 series MCUs, the boot memory also can be used as the main Flash memory extension area (AP mode) to store user-defined codes.

Note: The boot memory AP mode can only be set once, and the BOOT mode function of the original boot memory cannot be restored after setting.

The user can use Artery ICP Programmer to set the boot memory as the main Flash memory extension area as follows:

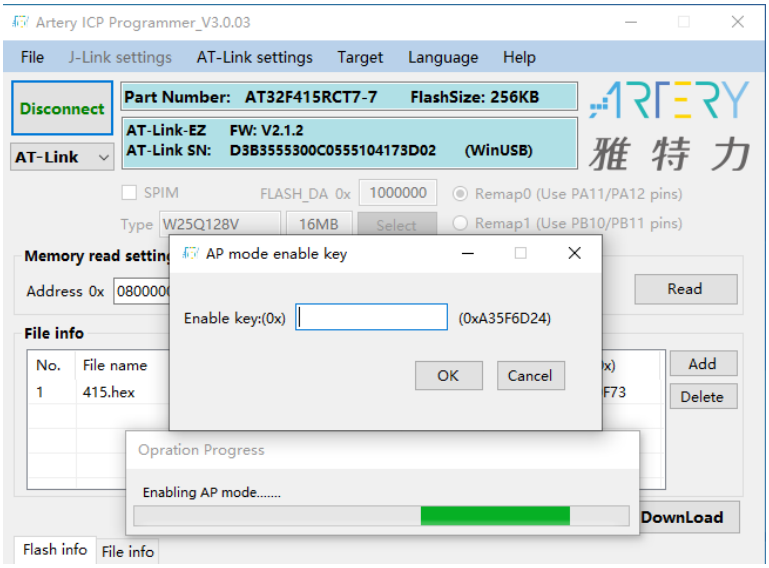
- Connect AT-Link/J-Link emulator to AT-START-F415 board and power on;
- Open Artery ICP programmer and select AT-Link/J-Link to connect;
- Target—Boot memory AP mode—OK.

Figure 26. Set AP mode in ICP Programmer



- To prevent misoperations, the user needs to manually enter the enable key 0xA35F6D24, and then the “Flash info” interface will display a success or failure message.

Figure 27. AP mode enabling in ICP Programmer



The user can use Artery ICP Programmer to set the boot memory as the main Flash memory extension area (in mass production) as follows:

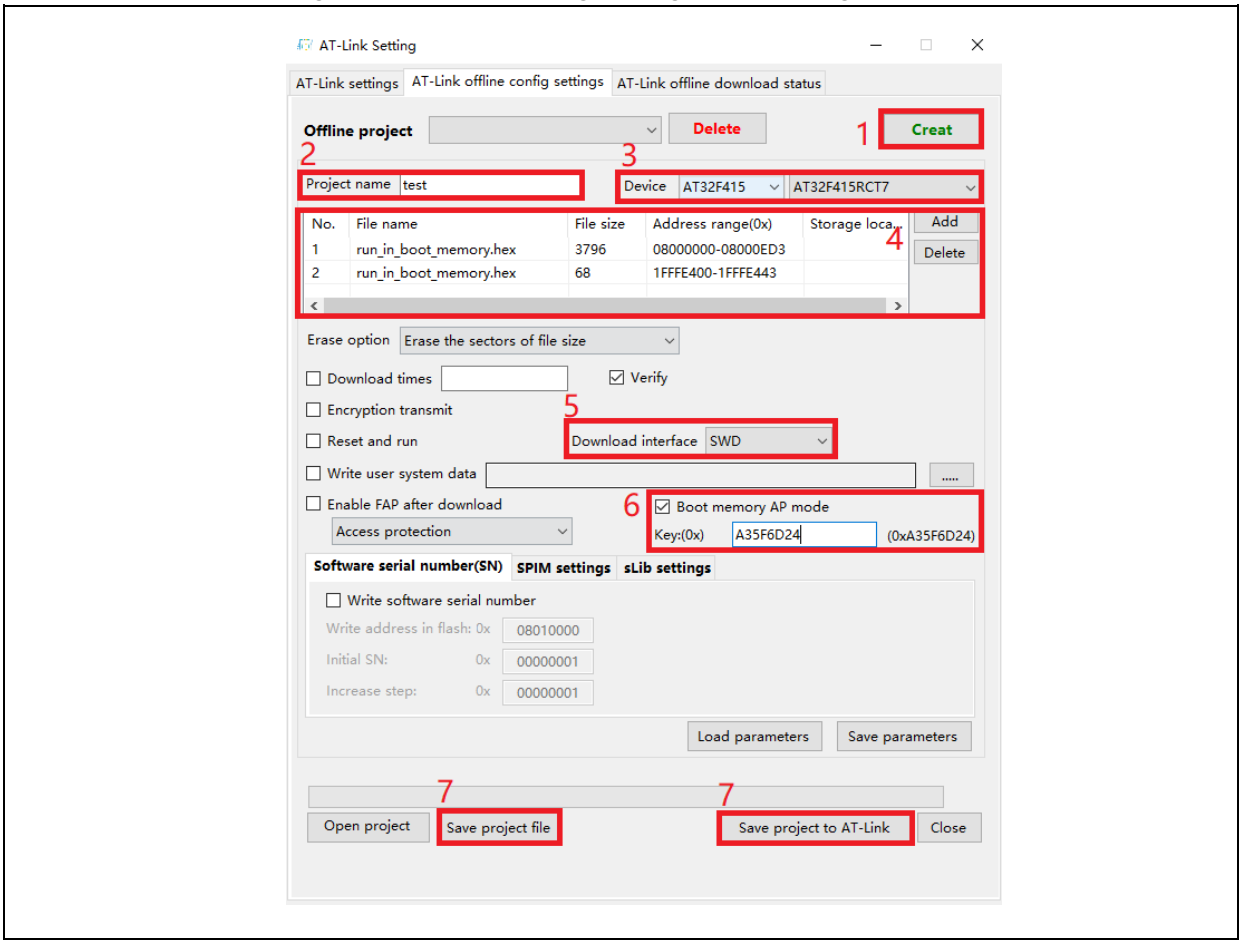
- Connect AT-Link emulator to AT-START-F415 board and power on.

Note: The on-board AT-Link EZ does not support offline programming. If necessary, please use AT-Link other than EZ version.

- Open Artery ICP programmer and select AT-Link to connect;
- AT-Link Setting—AT-Link offline config settings;
- Generate an offline project as follows:
  1. Click on “Create”;
  2. Enter project name;
  3. Select MCU part number;
  4. Add .hex file
  5. Select SWD as the download interface;
  6. Tick boot memory AP mode and enter the key;
  7. Save project to AT-Link or save project file.

Complete other relevant settings as necessary.

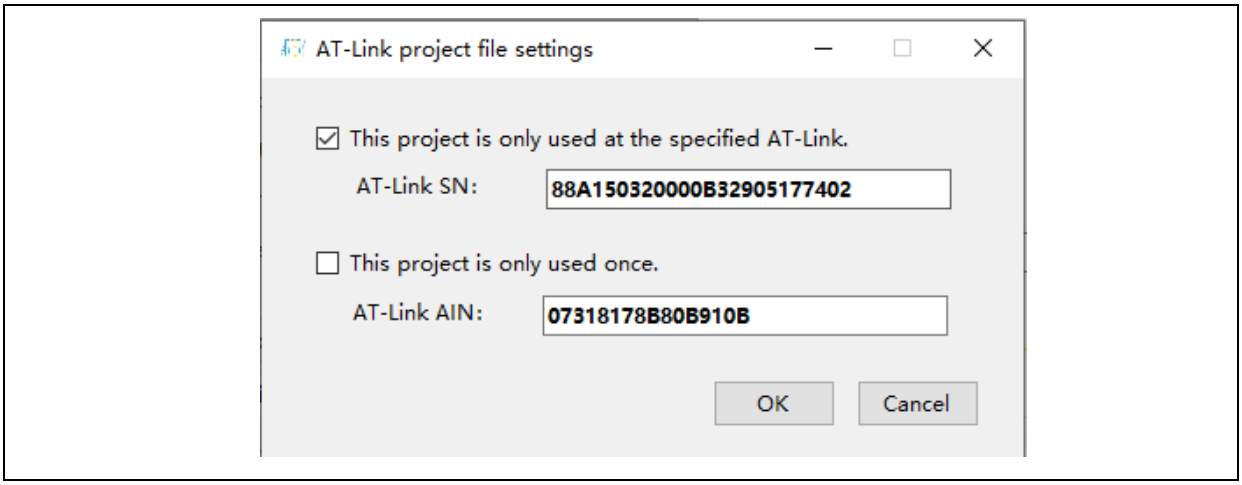
Figure 28. Offline config settings in ICP Programmer



- If you select “Save project file”, the project will be saved as an .atcp file for easy loading to other versions of AT-Link.

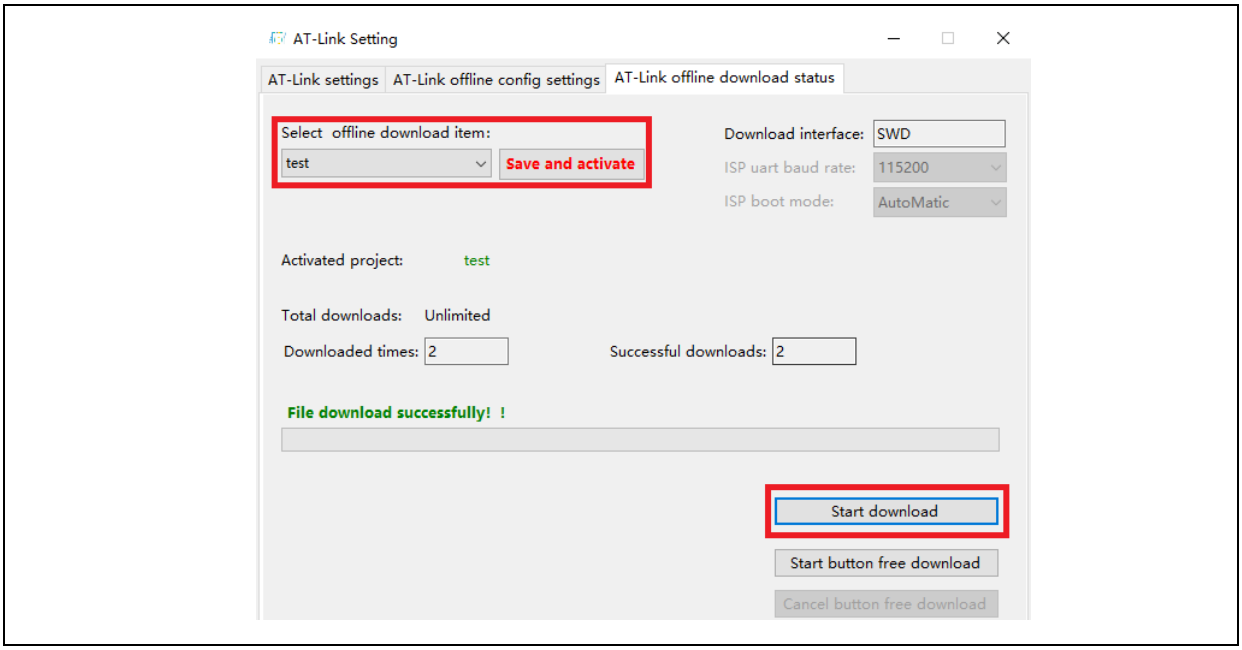
  The following window (as shown in Figure 24) will pop up in operation. If you tick “This project is only used at the specific AT-Link”, this project file will be bound to a specific AT-Link and can only be used on this AT-Link, and you need to set the AT-Link serial number. If you tick “This project is only used once”, this project file can only be used once on the same AT-Link.

Figure 29. AT-Link project file settings



- If the project is saved to AT-Link successfully, in the offline download status interface (as shown in Figure 25), select offline download item—Save and activate, to start download.

Figure 30. AT-Link offline download status



- Refer to *AN0066_config_boot_memory_as_extension_of_main_memory (AP_mode)* (ARTERY's official website→SUPPORT→AP Note→AN0066) for more information about the memory extension.
- Refer to BSP for the demo of running user application in boot memory. Path: ARTERY's official website→PRODUCTS→Value line→AT32F4xx; download and unzip BSP, and then get the AT32F415_Firmware_Library_V2.x.x\utilities\at32f415_boot_memory_ap_demo.

1.2.5 Recognize AT32 MCU in program

■ Read Cortex-M CPU ID to identify M0, M3 and M4 core

Figure 31. Read Cortex ID

```
cortex_id = *(uint32_t*)0xE000ED00;// Read Cortex ID
if((cortex_id == 0x410FC240) || (cortex_id == 0x410FC241))
{
        printf("This chip is Cortex-M4F.\r\n");
}
else
{
        printf("This chip is Other Device.\r\n");
}
```

■ Read PID and UID

Figure 32. Read PID and UID

```
/* Get the base address of AT32 MCU PID/UID */
#define DEVICE_ID_ADDR1 0x1FFFF7F3          // Define Artery MCU part number, UID base address
#define DEVICE_ID_ADDR2 0xE0042000          // Define MCU device number, PID base address

/* Used to store ID */
uint8_t     ID[5] = {0};

/* AT32F415 MCU type table */
const uint64_t AT32_MCU_ID_TABLE[] =
{
      0x0000000570030240,   //AT32F415RCT7      256KB     LQFP64
      0x00000005700301C1,   //AT32F415RBT7      128KB     LQFP64
      …
};

  /* Get PID/UID */
ID[0] = *(int*)DEVICE_ID_ADDR1;
ID[1] = *(int*)(DEVICE_ID_ADDR2+3);
ID[2] = *(int*)(DEVICE_ID_ADDR2+2);
ID[3] = *(int*)(DEVICE_ID_ADDR2+1);
ID[4] = *(int*)(DEVICE_ID_ADDR2+0);

/* Combine PID/UID */
  AT_device_id =
((uint64_t)ID[0]<<32)|((uint64_t)ID[1]<<24)|((uint64_t)ID[2]<<16)|((uint64_t)ID[3]<<8)|((uint64_t)ID[4]<<0);

/* Judge AT32 MCU */
for(i=0;i<sizeof(AT32_MCU_ID_TABLE)/sizeof(AT32_MCU_ID_TABLE[0]);i++)
{
    if(AT_device_id == AT32_MCU_ID_TABLE[i])
    {
        printf("This chip is AT32F4xx.\r\n");
    }
     else
    {
        printf("This chip is Other Device.\r\n");
    }
}
```

Note: The AT32F4xx MCU contains multiple ID codes. Combine the obtained ID information to a 64-bit data to help identify the specific MCU model. For more information, please refer to the Reference

Manual (DEBUG section) of each series and *AN0016_Recognize_AT32_MCU* (ARTERY's official website→SUPPORT→AP Note→AN0016).

2 FAQs in downloading/compiling

2.1 Hard Fault Handler

- The data being accessed is out of range.

  Find out the access violation point and modify the data to the normal range.
- The SRAM being used exceeds the specified MCU SRAM.
- The system clock configuration is out of specification.

2.2 J-Link cannot find IC

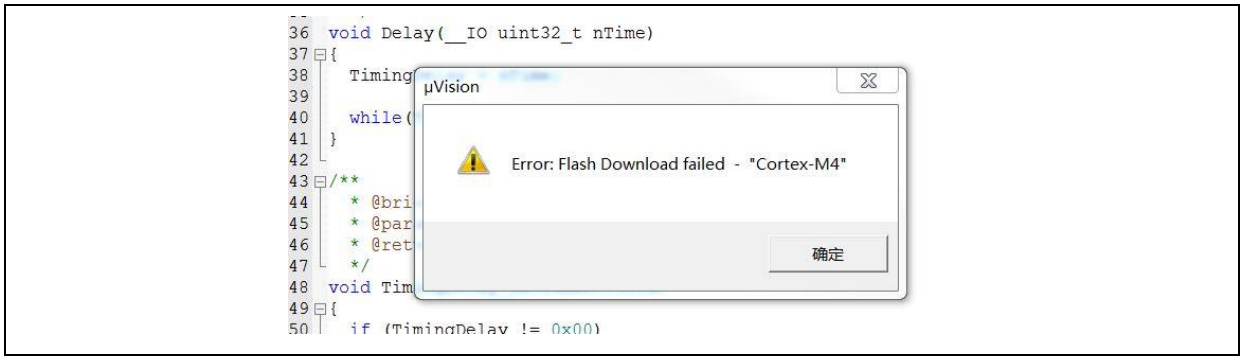
- Please refer to *FAQ0008_ J-Link cannot find IC* (ARTERY's official website→ SUPPORT→FAQ→FAQ0008).
- Please refer to *FAQ0132_Add Artery MCU to J-Link* (ARTERY's official website→ SUPPORT→FAQ→FAQ0132).

2.3 Problems in program downloading

2.3.1 Error: Flash Download failed–"Cortex-M4"

The following pop-up window appears in KEIL emulation or downloading:

Figure 33. Flash Download failed–"Cortex- M4" pops up in downloading



This pop-up window may appear in one of the following conditions:

- The access protection is enabled. Please disable MCU access protection and then download
- The selected Flash file algorithm is incorrect or no Flash file algorithm is loaded. Please add the proper Flash file algorithm in "Flash Download".
- BOOT0 and BOOT1 are incorrect. Set BOOT0=0 and BOOT1=0 to allow MCU to boot from the main Flash memory.
- The J-Link version is outdated. Please use 6.20C and above.
- JTAG/SWD PIN is disabled in the program. Please refer to "2.3.5 Resume download".

2.3.2 No Debug Unit Device found

- The download port is occupied. For example, ICP is connecting to the target device.
- JTAG/SWD connection error or not connected.

2.3.3 RDDI-DAP Error

- Compiler optimization level is too high. For example, the default optimization level of Keil AC6 is –Oz, which should be modified to -O0/-O1.
- JTAG/SWD PIN is disabled in the program. Please refer to “2.3.5 Resume download”.

2.3.4 ISP serial port gets stuck in downloading

When the ISP serial port is used for downloading, it may be stuck occasionally, and the PC cannot release serial port in this case.

The following operations are recommended:

- Check whether the power supply is stable;
- Replace with better USB-to-serial port tools, i.e., CH340 chip.

2.3.5 Resume download

When using AT32F415, the user may not be able to download the program again in the following conditions:

- After disabling JTAG/SWD PIN in the program, the program cannot be downloaded and JTAG/SWD device is not found.
- After entering Standby mode the program cannot be downloaded and JTAG/SWD device is not found.

Solutions in KEIL and IAR are as follows:

- Solution 1: Use the ConfigureJLink.exe provided by ARTERY;
- Solution 2: Switch boot mode.
  Switch boot mode to Boot[1:0]=01b or Boot[1:0]=11b, and then press “Reset” button to resume download (note: switch back to Boot[1:0]=00b). Similarly, ISP download can be resumed.
- Solution 3: ICP tool + AT-Link
  AT-Link is specially designed for AT32 MCUs. The ICP can be used together with AT-Link to resume download.

Solution 2 and Solution 3 require Boot PIN or device (AT-Link) additionally; therefore, this application note mainly introduces Solution 1.

2.3.5.1 Solution in KEIL

Use the ConfigureJLink.exe provided by ARTERY.

The following procedures are recommended:

- Place the ConfigJLink_V1.0.0.exe into the directory where the project file (*.uvprojx) is located;
- Double click the ConfigJLink_V1.0.0.exe, and the following window (as shown in Figure 34) will

pop up;

- Tick “Agree” and then click on “OK”; then the progress bar will pop up (as shown in Figure 35), and wait for the erase progress is completed, and then the program can be downloaded.

Figure 34. Operate ConfigJLink_V1.0.0 in KEIL

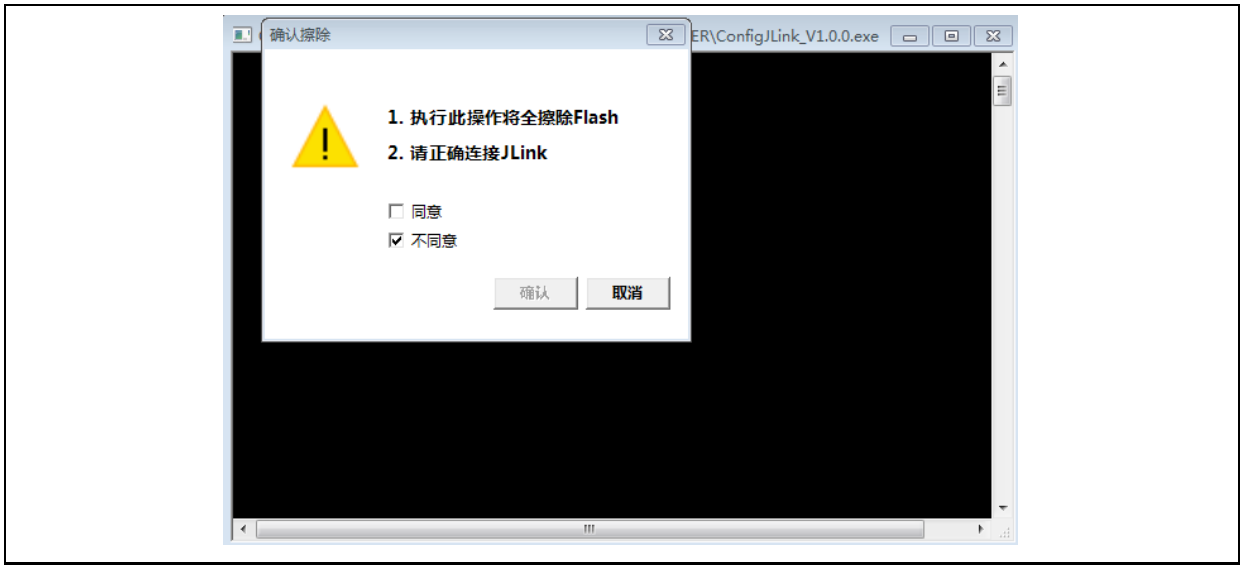
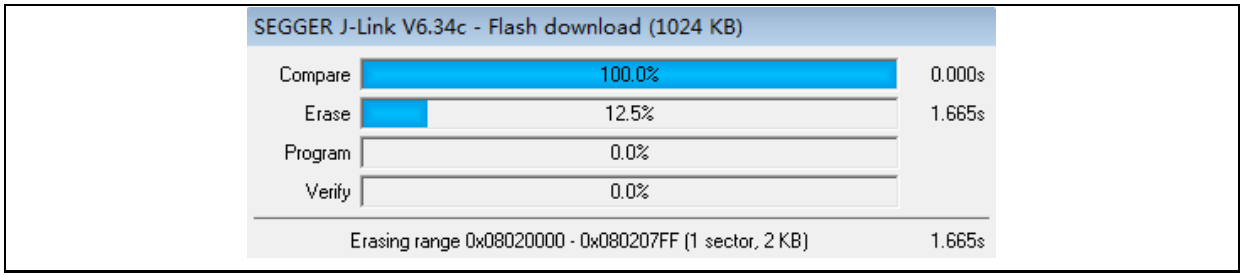


Figure 35. ConfigJLink_V1.0.0 execution progress in KEIL



Note 1: Make sure that SEGGER J-Link interface DLL is not lower than V6.14.

Note 2: If JTAG/SWD PIN is disabled every time the program is downloaded, perform the preceding steps before downloading the program.

Note 3: If the MCU enters Standby mode every time the program is downloaded, perform the preceding steps before the chip is powered on.

Note 4: In Keil, after AT32F415 MCU enters Standby mode, the solution using ConfigureJLink.exe is invalid.

2.3.5.2 Solutions in IAR

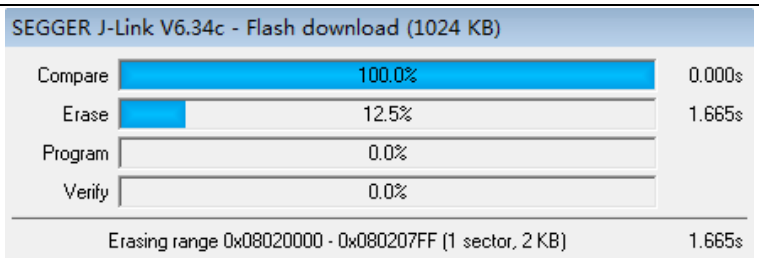
Use the ConfigJLink_V1.0.0.exe provided by ARTERY.

The following procedures are recommended:

- Place the ConfigJLink_V1.0.0.exe to the *settings* folder under the project directory.
- Double click the ConfigJLink_V1.0.0.exe and the following window (as shown in Figure 36) will pop up;
- Tick “Agree” and then click on “OK”; then the progress bar will pop up (as shown in Figure 37), and wait for the erase progress is completed, and then the program can be downloaded.

Figure 36. Operate ConfigJLink_V1.0.0 in IAR

Figure 37. ConfigJLink_V1.0.0 execution progress in IAR



Note 1: Make sure that SEGGER J-Link interface DLL is not lower than V6.14.

Note 2: If JTAG/SWD PIN is disabled every time the program is downloaded, perform the preceding steps before downloading the program.

Note 3: If the MCU enters Standby mode every time the program is downloaded, perform the preceding steps before the chip is powered on.

3 Security Library (sLib)

3.1 Introduction

As more and more MCU applications require complex algorithms and middleware solutions, it has become an important issue that how to protect IP-Codes (such as core algorithms) developed by software solution providers.

The AT32F415 series is designed with a security library (sLib) to protect important IP-Codes against being changed or read by the end user's program.

3.2 Principles of application

- Security library (sLib) is a defined area protected by a code in the main memory. Software solution providers store core algorithms in sLib for protection. The rest of the area can be used for secondary development by terminal customers.
- Security library includes the instruction security library (SLIB_INSTRUCTION) and data security library (SLIB_DATA). Users can select part of or the whole security library for instruction storage, but using the whole security library for storing data is not supported.
- Program code in the instruction security library (SLIB_INSTRUCTION) can only be fetched by MCU through I-Code bus (can only be executed) and cannot be read through D-Code bus (including ISP/ICP debug mode and programs that boot from internal RAM). When accessing the SLIB_INSTRUCTION in the manner of reading data, values are all read 0xFF.
- Data in the data security library (SLIB_DATA) can only be read through D-Code bus and cannot be programmed.
- The program code and data in security library cannot be erased unless the correct code is keyed in. If a wrong code is keyed in, in an attempt of writing or erasing the security library, a warning message will be issued by EPPERR=1 in the FLASH_STS register.
- The program code and data in security library are not erased when the end users perform a mass erase on the main Flash memory.
- Users can write the previously defined password in the SLIB_PWD_CLR register to disable security library protection. When the security library protection is disabled, the chip will perform a mass erase on the main Flash memory (including the contents of security library). Therefore, even if the code defined by the software solution provider is leaked, the program code will not be leaked.

3.3 Security library application

For details, please refer to *AN0065_AT32F415_Security_Library_Application_Note* (ARTERY's official website→SUPPORT→AP Note→AN0065).

4 Revision history

Table 1. Document revision history

Date	Version	Revision note
2021.12.20	2.0.0	Initial release.
2022.08.04	2.0.1	Updated 3rd party programming tools.
2022.10.10	2.0.2	Added description of development environment and file path.
2022.10.21	2.0.3	Optimized description of UID and PID.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Purchasers are solely responsible for the selection and use of ARTERY's products and services; ARTERY assumes no liability for purchasers' selection or use of the products and the relevant services.

No license, express or implied, to any intellectual property right is granted by ARTERY herein regardless of the existence of any previous representation in any forms. If any part of this document involves third party's products or services, it does NOT imply that ARTERY authorizes the use of the third party's products or services, or permits any of the intellectual property, or guarantees any uses of the third party's products or services or intellectual property in any way.

Except as provided in ARTERY's terms and conditions of sale for such products, ARTERY disclaims any express or implied warranty, relating to use and/or sale of the products, including but not restricted to liability or warranties relating to merchantability, fitness for a particular purpose (based on the corresponding legal situation in any unjudicial districts), or infringement of any patent, copyright, or other intellectual property right.

ARTERY's products are not designed for the following purposes, and thus not intended for the following uses: (A) Applications that have specific requirements on safety, for example: life-support applications, active implant devices, or systems that have specific requirements on product function safety; (B) Aviation applications; (C) Auto-motive application or environment; (D) Aerospace applications or environment, and/or (E) weapons. Since ARTERY products are not intended for the above-mentioned purposes, if purchasers apply ARTERY products to these purposes, purchasers are solely responsible for any consequences or risks caused, even if any written notice is sent to ARTERY by purchasers; in addition, purchasers are solely responsible for the compliance with all statutory and regulatory requirements regarding these uses.

Any inconsistency of the sold ARTERY products with the statement and/or technical features specification described in this document will immediately cause the invalidity of any warranty granted by ARTERY products or services stated in this document by ARTERY, and ARTERY disclaims any responsibility in any form.