AN0142
Application Note

Getting started with AT32L021 series-based development

Introduction

The purpose of this document is to provide users with a quick-start guide on AT32L021 series-based project development.

Note: The codes in this document are built around ARTERY's V2.x.x BSP. Attention should be paid to the differences between different versions of BSP when in use.

Applicable products：

Product series	AT32L021 series

Contents

List of tables

List of figures

1 Development resources

Resources download link:

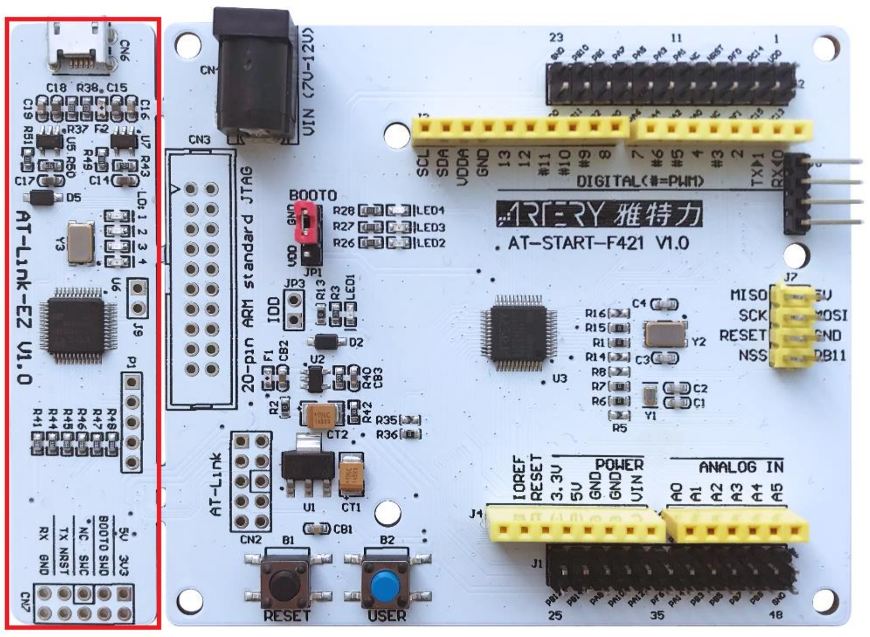
- Artery official website: http://www.arterychip.com

1.1 Set up AT32 development environment

1.1.1 Debug tools and evaluation board

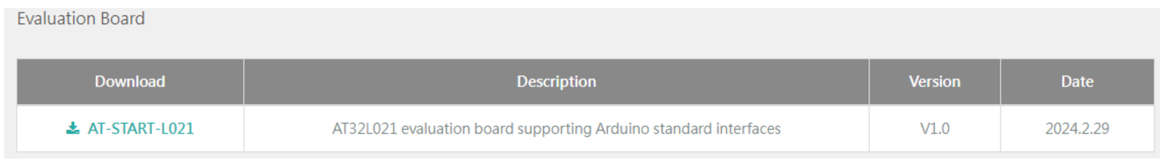
The debug tools for the AT32L021 series can be AT-Link/J-Link. In *Figure 1* below, the area in red rectangle box on the left side represents the AT-Link-EZ. The AT-Link-EZ can also be separated from the board to work in conjunction with other circuit boards. It supports a variety of functions such as IDE online debugging, online programming and USB-to-serial interface.

Figure 1. AT-START-L021 and AT-Link-EZ



Note: For details on AT-START-AT32L021 evaluation board, refer to the "UM_AT_START_L021_Vx.x", which is available from ARTERY's official website. You can access ARTERY official website → PRODUCT → Low power line → AT32L0xx series → Resources → Evaluation board, where you can download a ZIP-format AT-START-L021 and get AT_START_L021_Vx.x\03_Documents.

Figure 2. AT-START-L021 evaluation board package from ARTERY official website



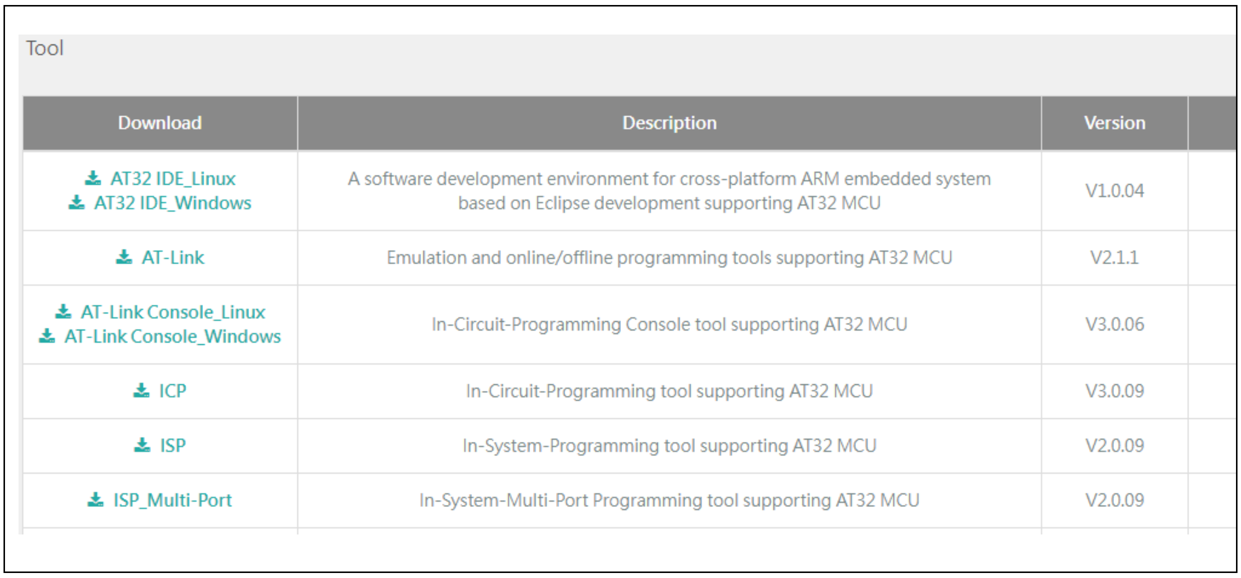
Evaluation Board

Download	Description	Version	Date
AT-START-L021	AT32L021 evaluation board supporting Arduino standard interfaces	V1.0	2024.2.29

1.1.2 Programming tools and software resources

- **ATERTY programming tools and software**: AT-Link /AT-Link+ /AT-Link-Pro /AT-Link-ISO /AT-Link-EZ, and ICP/ISP
- **ICP user guide:** Refer to the "*UM_ICP_Programmer",* which can be found at ARTERY official website → Product → Low power → AT32L0xx → Tool → download ICP document
- **ISP user guide:** Refer to the UM_ISP_Programmer, which can be found at ARTERY official website → Product → Low power → AT32L0xx → Tool → download ISP document
- **AT-Link user guide:** Refer to the *"UM0004_AT-Link_User_Manual*", which can be found at ARTERY official website → Product → Low power → AT32L0xx → Tool → download AT-Link document.

Figure 3. ICP/ISP/AT-Link-Family package from ARTERY official website



Tool

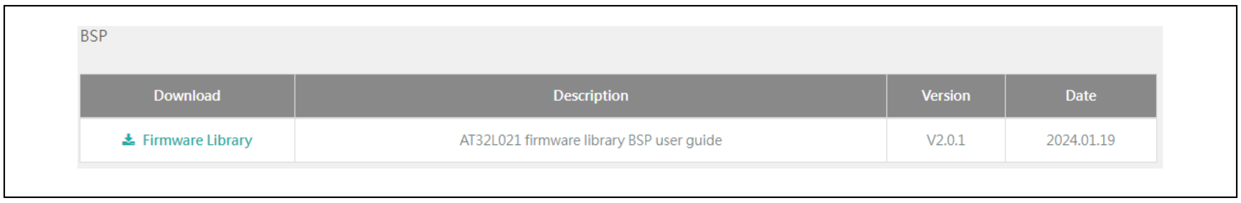
Download	Description	Version
AT32 IDE_Linux AT32 IDE_Windows	A software development environment for cross-platform ARM embedded system based on Eclipse development supporting AT32 MCU	V1.0.04
AT-Link	Emulation and online/offline programming tools supporting AT32 MCU	V2.1.1
AT-Link Console_Linux AT-Link Console_Windows	In-Circuit-Programming Console tool supporting AT32 MCU	V3.0.06
ICP	In-Circuit-Programming tool supporting AT32 MCU	V3.0.09
ISP	In-System-Programming tool supporting AT32 MCU	V2.0.09
ISP_Multi-Port	In-System-Multi-Port Programming tool supporting AT32 MCU	V2.0.09

1.1.3 AT32 MCU development environment

1.1.3.1 Template project

The general IDE template projects are included in ARTERY's firmware BSP. You can get these resources by visiting ARTERY official website → Product → Low power → AT32L0xx series → BSP.

Figure 4. BSP resources from ARTERY official website



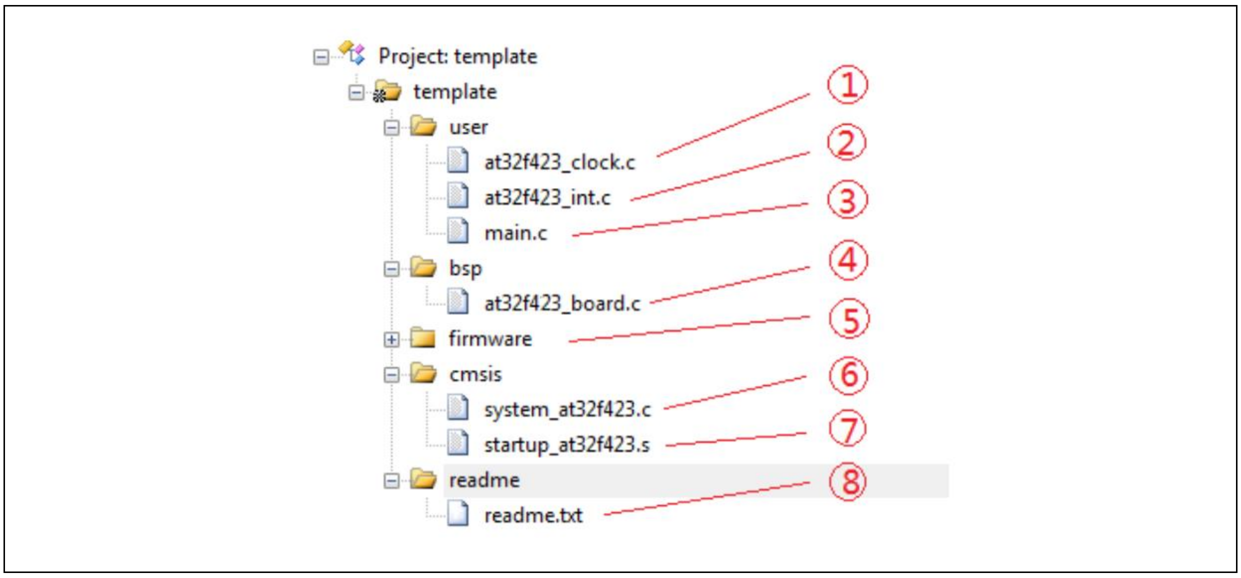
BSP

Download	Description	Version	Date
Firmware Library	AT32L021 firmware library BSP user guide	V2.0.1	2024.01.19

BSP contains template projects such as Keil_v5/Keil_v4/IAR_6.10/IAR_7.4/IAR_8.2/eclipse_gcc/at32_ide. They are located at AT32L021_Firmware_Library_V2.x.x\project\at_start_f4xx\templates.

Figure 5 below shows an example of Keil_v5 project.

Figure 5. Keil_v5 template



Taking Keil_v5 template as an example, it contains the following items:

① **at32l021_clock.c:** it is a clock configuration file defining the default clock frequency and clock path

② **at32l021_int.c:** it refers to an interrupt file containing codes related to core interrupt functions

③ **main.c:** it refers to the main code files of template projects

④ **at32l021_board.c:** board configuration file includes buttons, LED and other configurations

⑤ **firmware:** it contains "*at32l021_xx.c*" that is used as a driver file for peripherals

⑥ **system_at32l021.c:** system initialization file

⑦ **startup_at32l021.s:** startup file

⑧ **readme.txt:** a read-me txt file that describes information about template projects such as application functions, configuration method and application notes

In addition to "Templates", a large number of example codes are included in BSP for users' reference. These examples can be found at AT32L021_Firmware_Library_V2.x.x \ project \at_start_f4xx \examples.

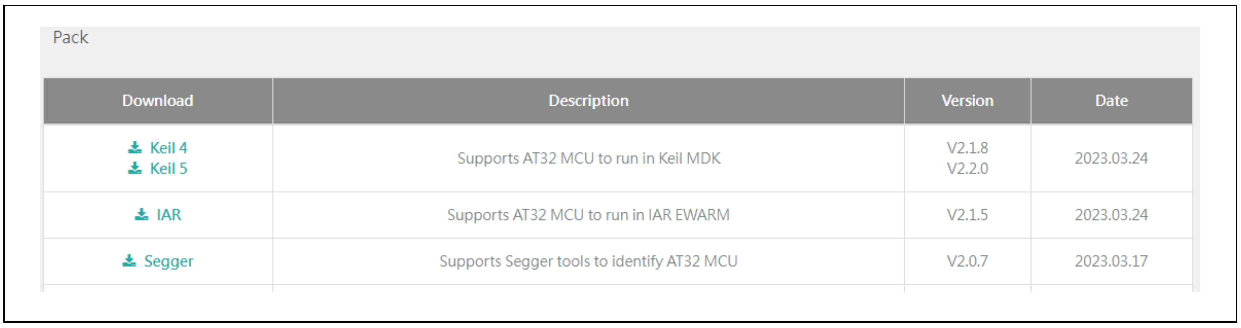
Note: For details on BSP, refer to Section 4 of the document "AT32L021_firmware_BSP&Pack_user_guide". This guideline is available from ARTERY official website → Product → Low power → AT32L0xx series → BSP.

1.1.3.2 Installing Pack

Pack can be used to add a specific AT32 MCU part number in Keil/IAR.

Pack can be downloaded from ARTERY official website → *Product* → *Low power* → *AT32L021 series* → *Pack.*

Figure 6. Pack resources from ARTERY official website



Pack

Download	Description	Version	Date
Keil 4 Keil 5	Supports AT32 MCU to run in Keil MDK	V2.1.8 V2.2.0	2023.03.24
IAR	Supports AT32 MCU to run in IAR EWARM	V2.1.5	2023.03.24
Segger	Supports Segger tools to identify AT32 MCU	V2.0.7	2023.03.17

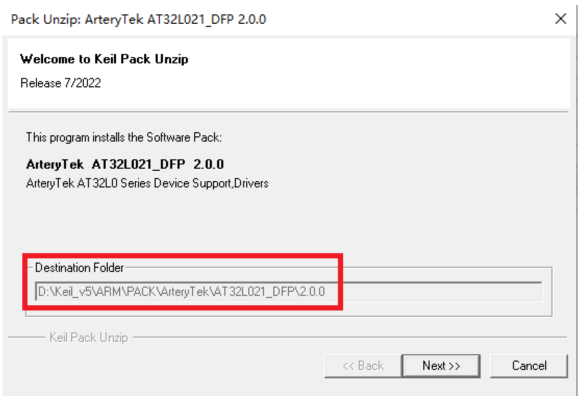
For Keil compiling system, the keil 4.74 or V5.23 above is recommended.

For Kei_v5 system, users need first unzip “Keil5_AT32MCU_AddOn” and then install the “ArteryTek.AT32L021_DFP”.

For Keil_v4 system, users need install “Keil4_AT32MCU_AddOn”.

By default, the installation path of Keil can be automatically recognized when installing. In case of recognition failure or path error, it is necessary for users to manually choose a Keil path.

Figure 7. Install ArteryTek.AT32L021_DFP



Pack Unzip: ArteryTek AT32L021_DFP 2.0.0

Welcome to Keil Pack Unzip

Release 7/2022

This program installs the Software Pack:

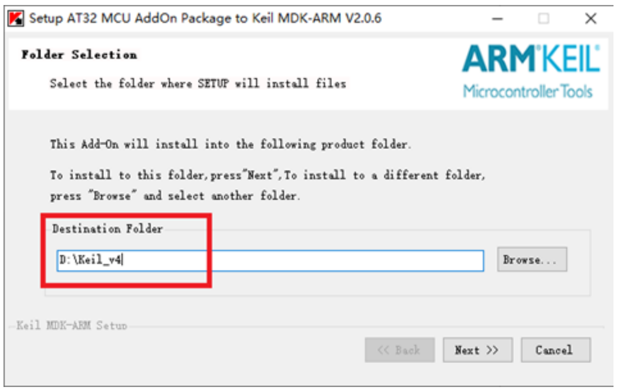
ArteryTek AT32L021_DFP 2.0.0

ArteryTek AT32L0 Series Device Support,Drivers

Destination Folder

D:\Keil_v5\ARM\PACK\ArteryTek\AT32L021_DFP\2.0.0

Figure 8. Install Keil4_AT32MCU_AddOn



Setup AT32 MCU AddOn Package to Keil MDK-ARM V2.0.6

Folder Selection

Select the folder where SETUP will install files

This Add-On will install into the following product folder.

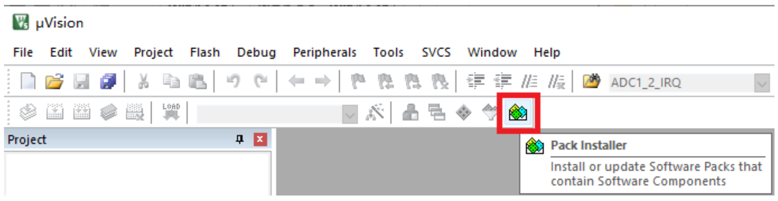
To install to this folder, press"Next", To install to a different folder, press "Browse" and select another folder.

Destination Folder

D:\Keil_v4

You can also open Keil, click the icon "Pack Installer", then click "File", and choose "Import" to import and install "Pack" you download from ARTERTY official website.

Figure 9. Click "Pack Installer" icon in Keil

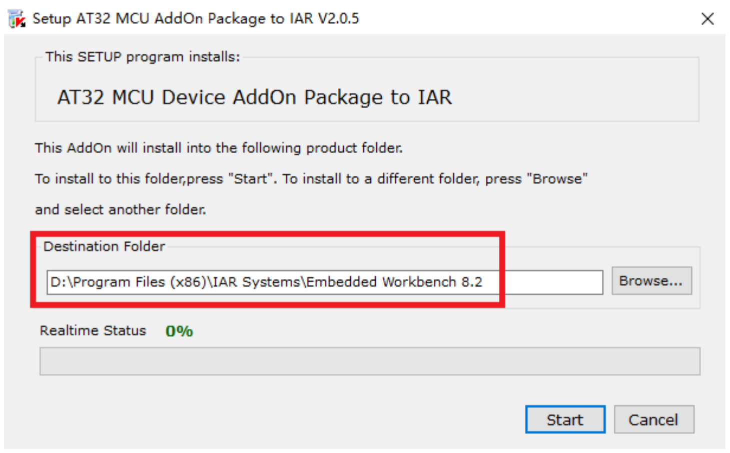


µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
ADC1_2_IRQ
Project
Pack Installer
Install or update Software Packs that contain Software Components

If IAR compiling system is to be used, the IAR7.0 or IAR6.1 above is recommended. While installing the "IAR_AT32MCU_AddOn", the installation path of Keil can be automatically recognized.

In case of recognition failure or path error, it is necessary for users to manually choose an IAR path.

Figure 10. Install IAR_AT32MCU_AddOn



Setup AT32 MCU AddOn Package to IAR V2.0.5

This SETUP program installs:

AT32 MCU Device AddOn Package to IAR

This AddOn will install into the following product folder.

To install to this folder,press "Start". To install to a different folder, press "Browse" and select another folder.

Destination Folder

D:\Program Files (x86)\IAR Systems\Embedded Workbench 8.2

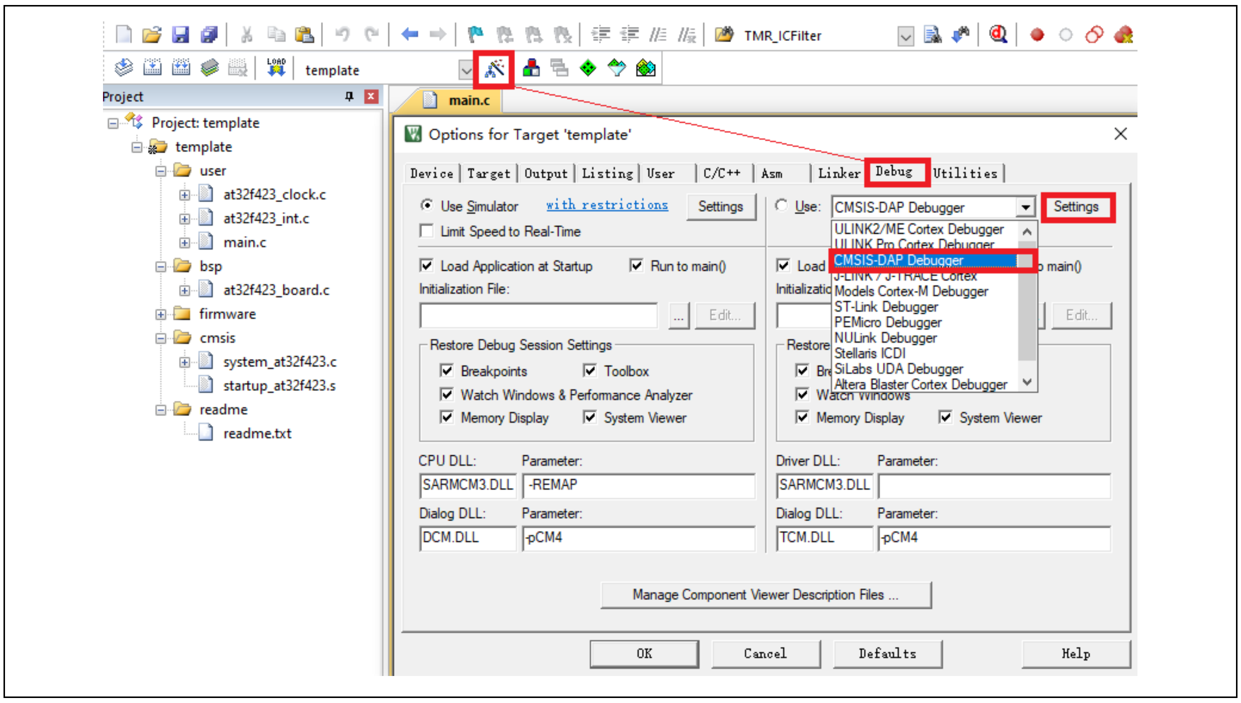
Realtime Status 0%

Note: For details on Pack, refer to Section 2 of the document "AT32L021_firmware_BSP&Pack_user_guide". This guideline is available from ARTERY official website → Product → Low power → AT32L021 series → BSP. Downloading BSP zip file, you can find the AT32L021_Firmware_Library_Vx.x.x\document in it.

1.1.3.3 Debug and download with AT-Link tool

To use AT-Link in Keil environment, click "**Debug**", and choose "**CMSIS-DAP Debugger**".

Figure 11. Keil Debug option



Options for Target 'template'

Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities

Use Simulator with restrictions Settings
Limit Speed to Real-Time
Load Application at Startup ✓ Run to main() ✓
Initialization File:
Restore Debug Session Settings: Breakpoints ✓ Toolbox ✓ Watch Windows & Performance Analyzer ✓ Memory Display ✓ System Viewer ✓
CPU DLL: SARMCM3.DLL Parameter: -REMAP
Dialog DLL: DCM.DLL Parameter: -pCM4

Use: CMSIS-DAP Debugger Settings
ULINK2/ME Cortex Debugger
ULINK Pro Cortex Debugger
CMSIS-DAP Debugger
J-LINK / J-TRACE Cortex
Models Cortex-M Debugger
ST-Link Debugger
PEMicro Debugger
NULink Debugger
Stellaris ICDI
SiLabs UDA Debugger
Altera Blaster Cortex Debugger
Driver DLL: SARMCM3.DLL Parameter:
Dialog DLL: TCM.DLL Parameter: -pCM4

Manage Component Viewer Description Files ...
OK | Cancel | Defaults | Help

Click "**Settings**" to enter "Cortex-M Target Driver Setup" interface, as shown in Figure 12.

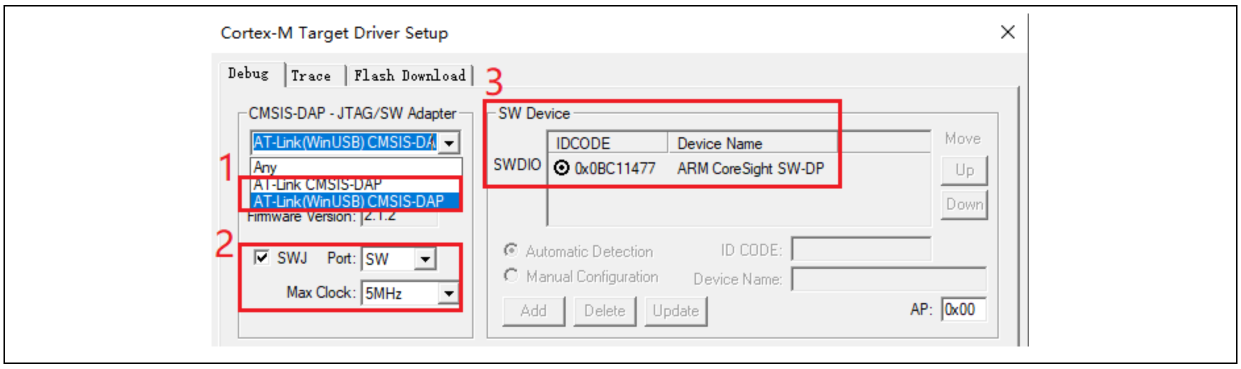
1. Choose "**AT-Link(WinUSB)-CMSIS-DAP/AT-Link-CMSIS-DAP**"

Note: For more information on WinUSB, refer to "FAQ0136_How_to_use_AT-LINK_WinUSB_EN_V2.0.0". This FAQ is stored at ARTERY official website → Support → FAQ → FAQ0136.

2. Choose "**SW**" In "**Port**" option, and then check "**SWJ**" option

3. As shown in step 3 below, the ARM SW-DP debug module is recognized.

Figure 12. Settings option in Keil Debug



Cortex-M Target Driver Setup

Debug | Trace | Flash Download

CMSIS-DAP - JTAG/SW Adapter: AT-Link(WinUSB) CMSIS-DAP
Any
AT-Link CMSIS-DAP
AT-Link(WinUSB) CMSIS-DAP
Firmware Version: 2.1.2
SWJ ✓ Port: SW
Max Clock: 5MHz

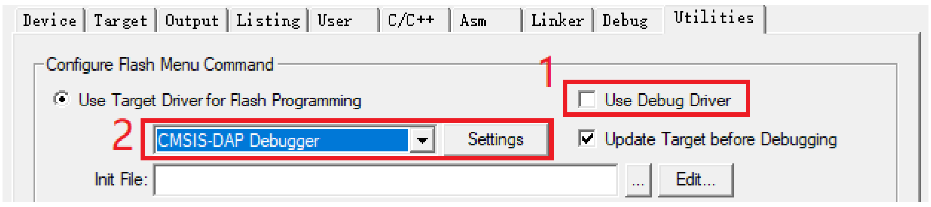
SW Device

	IDCODE	Device Name
SWDIO	0x0BC11477	ARM CoreSight SW-DP

Automatic Detection ID CODE:
Manual Configuration Device Name:
Add | Delete | Update AP: 0x00

In "**Utilities**" option, first de-check "**Use Debug Driver**" (see step 1), then choose "**CMSIS-DAP Debugger**" in Step 2, and re-check "**Use Debug Driver"** in Step 1 (noted that step 1 must be first de-checked before being checked again later)

Figure 13. Keil Utilities option



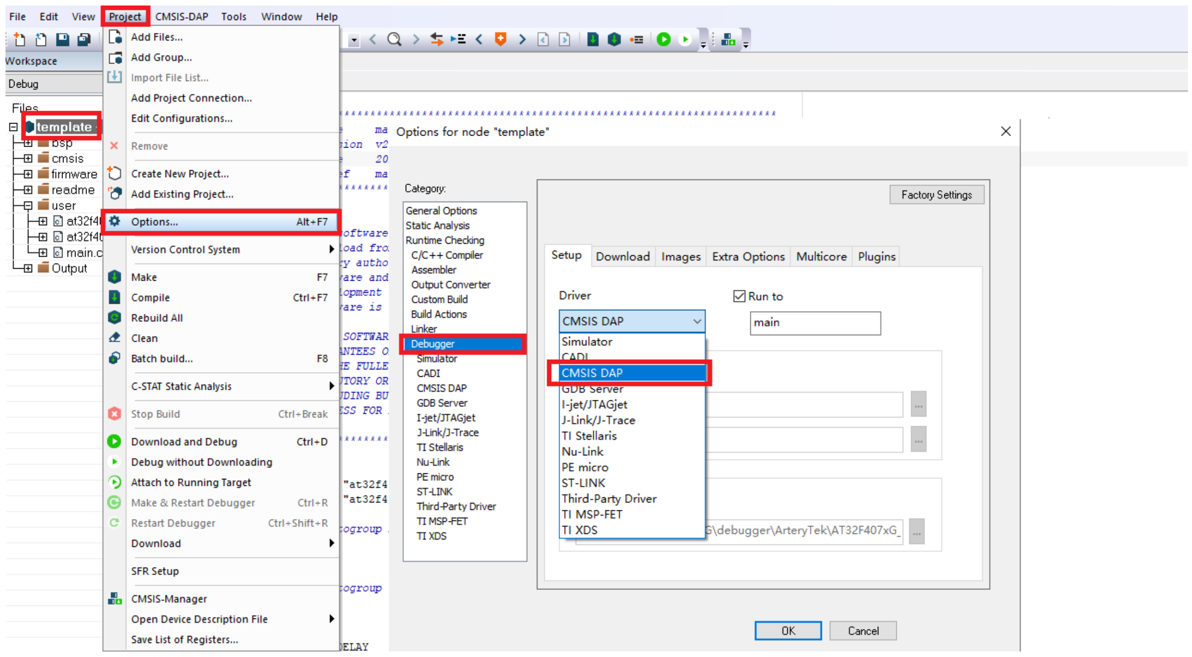
(● Use Target Driver for Flash Programming) CMSIS-DAP Debugger

☐ Use Debug Driver

☑ Update Target before Debugging

To use AT-Link in IAR environment, click “**Project**”, choose “**Options**”, go to “**Debugger**” and choose “**CMSIS-DAP**”, and then check “**SWD**” option.

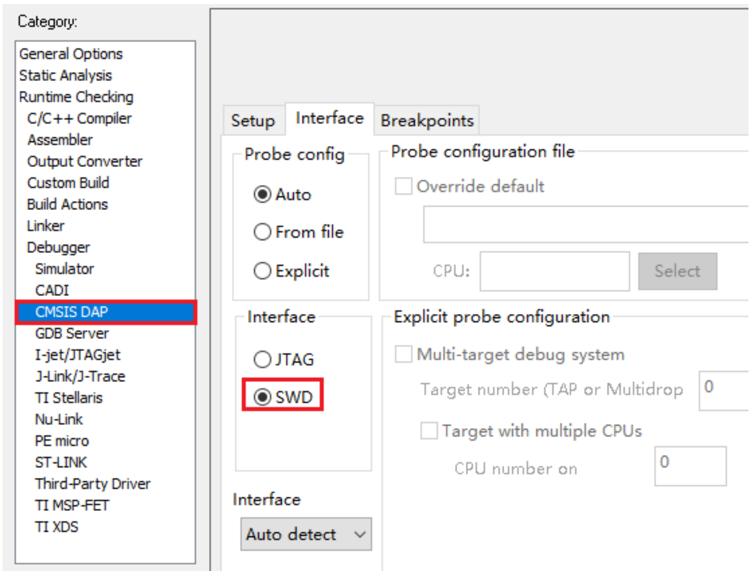
Figure 14. IAR Debug option



Driver: CMSIS DAP

☑ Run to: main

Figure 15. IAR CMSIS-DAP option



Probe config: [● Auto] ○ From file ○ Explicit

Interface: ○ JTAG [● SWD]

Interface: Auto detect

Note: The document AT32L021_firmware_BSP&Pack_user_guide provides details on Flash algorithm files, MCU product series replacement and J-Link. Thus they are not explained here. This user guide is located at ARTERY official website → Product → Low power → AT32L021 series → BSP.

1.1.4 How to quickly migrate from one MCU to another

- Refer to the "*MG0016_Migrating_from_SXX32F030_to_AT32L021*", which can be found at ARTERY official website → Product → Low power → AT32L021 series page
- If the program fails to run normally, refer to the corresponding sections of this document, or contact your local or nearest ARTERY Tech team for assistance.

Note: For more information on how to achieve optimal performance of AT32L021 series, please refer to the application note "AN0004_Performance_Optimization". You can read or download this file by visiting ARTERY official website → Product → AP Note → AN0004.

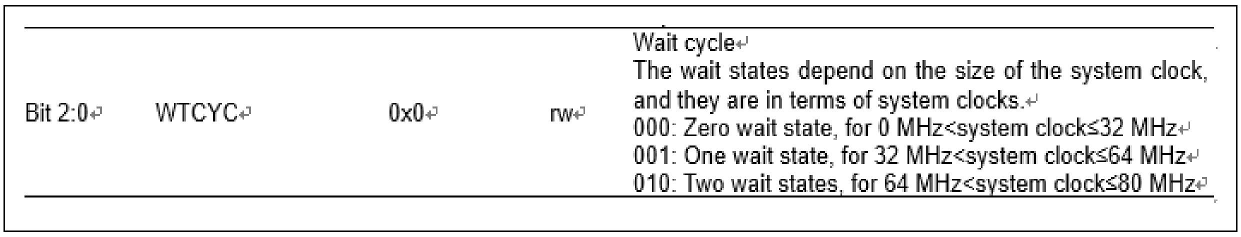
1.2 AT32L021 functional overview

1.2.1 Instruction prefetch buffer

Instruction prefetch buffer is useful for achieving quicker CPU execution. After instruction prefetch buffer is enabled, the subsequent word is already awaiting in the buffer while CPU is reading the existing word. The instruction prefetch controller determines whether or not to access Flash memory according to space available in the buffer. When there is at least a free space in the instruction prefetch buffer, the instruction prefetch controller will trigger a read access.

Different system clocks require different wait states, which can be set through the bit [2:0] (WTCYC) in the FLASH_PSR register.

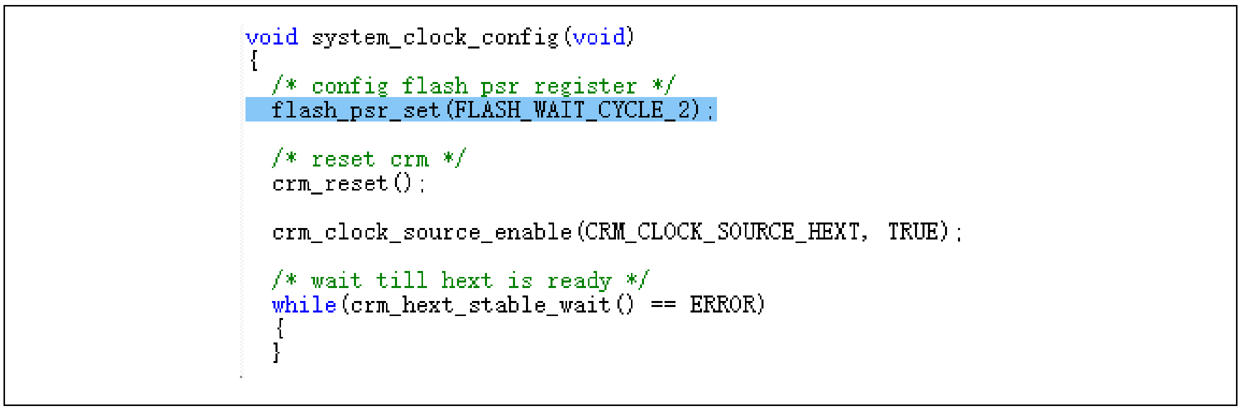
Figure 16. Wait state bit in FLASH_PSR register



Bit 2:0	WTCYC	0x0	rw	Wait cycle The wait states depend on the size of the system clock, and they are in terms of system clocks. 000: Zero wait state, for 0 MHz<system clock≤32 MHz 001: One wait state, for 32 MHz<system clock≤64 MHz 010: Two wait states, for 64 MHz<system clock≤80 MHz

AT32 library has made relevant settings in the system_clock_config() function. For BSP of other AT32 MCU series, you can also find these settings at the same location of the function.

Figure 17. system_clock_config function



```
void system_clock_config(void)
{
  /* config flash psr register */
  flash_psr_set(FLASH_WAIT_CYCLE_2);

  /* reset crm */
  crm_reset();

  crm_clock_source_enable(CRM_CLOCK_SOURCE_HEXT, TRUE);

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

1.2.2 PLL clock settings

The AT32L021 series embeds a PLL with a maximum of 72MHz clock output. There are two methods to configure the PLL. One is to use the CRM_CFG register (clock configuration register), the other is to use the CRM_PLL register. The latter is capable of setting various PLL clock frequencies, based on the following formula:

$$\text{PLL input clock} = \text{PLL reference input clock} \times \frac{\text{PLL frequency multiplication factor PLL\_NS}}{\text{PLL predivision factor } \text{PLL}_{\text{MS}} \times \text{PLL postdivision setting value PLL\_FR}}$$

Example 1: using CRM_CFG register to set PLL clock (HEXT=8MHz, PLL=72MHz)

```
crm_pll_config(CRM_PLL_SOURCE_HEXT, CRM_PLL_MULT_9);
```

Example 2: using CRM_PLL register to set PLL clock (HEXT=8MHz, PLL=72MHz)

Figure 18. AT32L021 70MHz output clock configuration

```
#define   CRM_PLL_NS   ((uint16_t)0x23)   /* PLL_NS=35 */

#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 first parameter "**CRM_PLL_SOURCE_HEXT**" represents HEXT as an external clock source, PLL_NS is **35,** PLL_MS is **1,** and PLL_FR value is CRM_PLL_FR_4 (0x02, divided by 4).

For more information on clock configuration, please refer to the document "*AN0134_AT32L021_CRM_Start_Guide*". This user guide can be found at ATERTY official website → Support → AP Note → AN0134. This document describes how to configure and modify the clock source code of the AT32L021 series, and how to generate the desired code and apply them into project by using ARTERY's New Clock Configuration tool.

The New Clock Configuration document can be found at ATERTY official website → Product → Low power → AT32L0xx series → Tool.

1.2.3 Encryption

Note: The BOOT1 bit of AT32L021 series is located in the user system data area (0x1FFF F800). Before using ISP tool, it is necessary to ensure that nBOOT1=1 is asserted (default value) so that the program is booted from system memory instead of SRAM.

1.2.3.1 Access protection

Access protection is usually known as an encryption operation. It applies to the entire Flash memory. Once the access protection is enabled, the embedded Flash memory can only be read through normal program execution, rather than through JTAG or SWD.

Using ISP or ICP tool to unlock access protection will trigger erase operation to the Flash memory.

Note: Once enabled, high-level access protection can not be unlocked. Meanwhile, it is forbidden for users to erase and write system area in any forms.

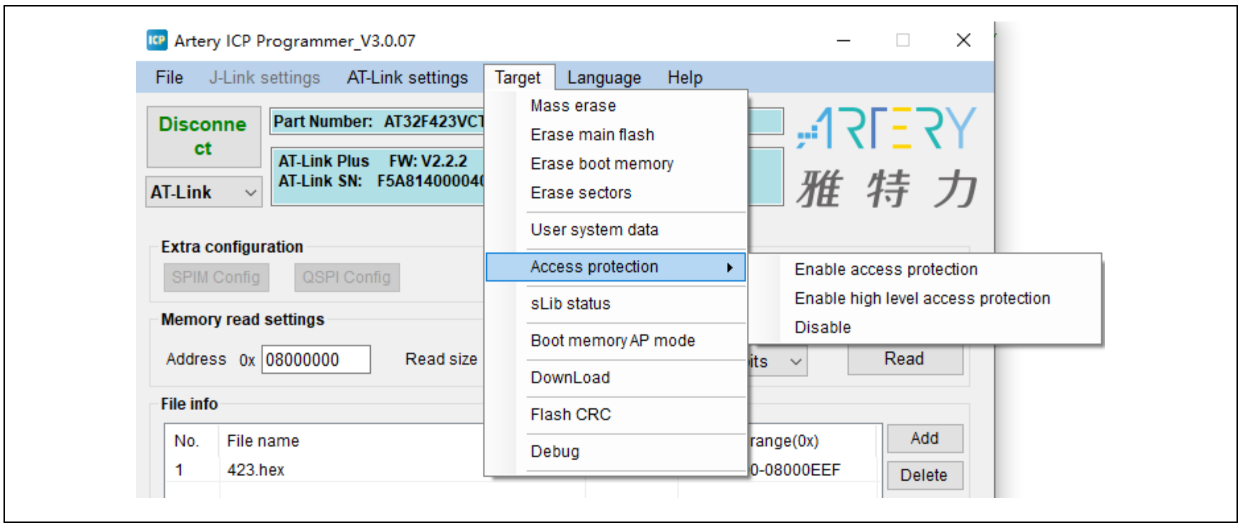
ISP or ISP tool can be used to enable and disable access protection.

- Artery ICP Programmer (BOOT0=0)

  Enable access protection: click Target – Access protection – Enable access protection or enable high-level access protection.

  Unlock access protection: click Target – Access protection – disable

Figure 19. Use ISP tool to enable or disable access protection



- Artery ISP Programmer tool (BOOT0=1)

  Enable access protection: Keep clicking “Next” until you enter the final interface. Then check “Protection”, choose “Enable” and “Access protection”, click “Yes”.

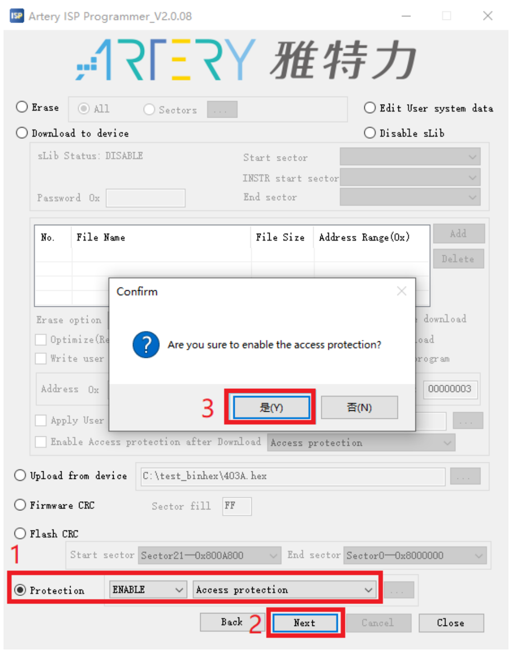
  Unlock access protection: check “Protection”, choose “Disable” and “Access protection”, click “Yes”.

- Artery ISP Multi-Port Programmer tool (BOOT0=1)

  Enable access protection: check “Protection”, choose “Enable” and “Access protection” or “High-level access protection”, click “Start”.

  Unlock access protection: check “Protection”, choose “Disable” and “Access protection”, click “Start”.

Figure 20. Use ISP to enable access protection

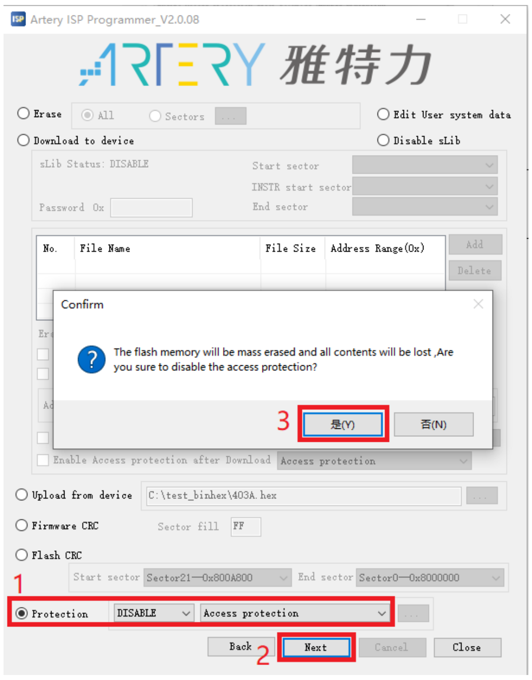

Artery ISP Programmer_V2.0.08

Confirm

Are you sure to enable the access protection?

Protection: ENABLE, Access protection

Figure 21. Use ISP to unlock access protection


Artery ISP Programmer_V2.0.08

Confirm

The flash memory will be mass erased and all contents will be lost ,Are you sure to disable the access protection?

Protection: DISABLE, Access protection

Note: Access protection, after enabled, cannot be unlocked through erase operation.

1.2.3.2 Erase/write protection

Write protection applies to the entire Flash memory or to part of Flash area. Once Flash write protection is enabled, the embedded Flash memory are write-protected against any writing operation.

ISP or ICP tool can be used to enable or disable erase/write protection:

- Artery ICP Programmer tool (BOOT0=0)

  Enable erase/write protection: click “Target” – User system area – choose the sectors to be erase/write-protected – apply to device

  Disable erase/write protection: click “Target” – User system area – cancel the sectors to be erase/write-protected – apply to device

- Artery ISP Programmer tool (BOOT0=1)

  Enable erase/write protection: check “Protection” option, choose “Enable” and “Erase/write protection”, and then click “Yes”

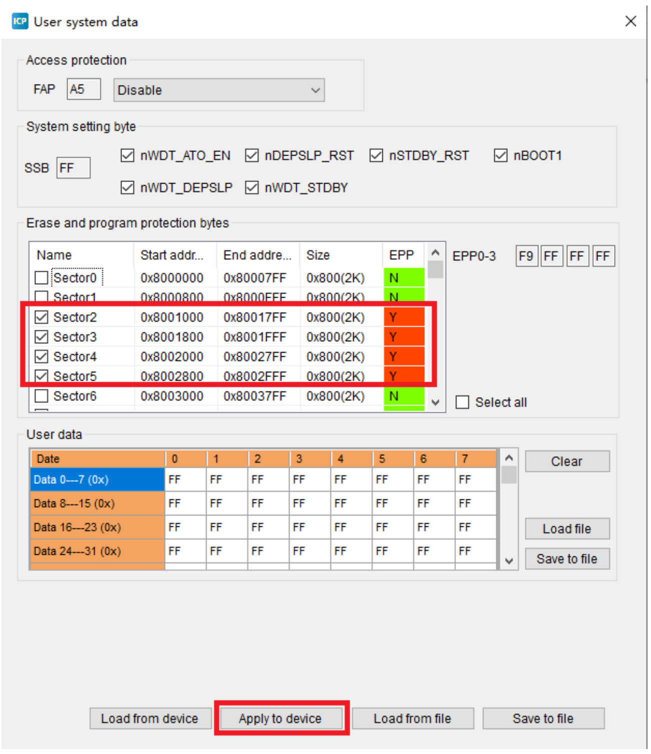
  Disable erase/write protection: check “Protection” option, choose “Disable” and “Erase/write protection”, and then click “Yes”

- Artery ISP Multi-Port Programmer tool (BOOT0=1)

  Enable erase/write protection: check “Protection” option, choose “Enable” and “Erase/write protection”, and then click “Yes”

  Disable erase/write protection: check “Protection” option, choose “Disable” and “Erase/write protection”, and then click “Yes”

Figure 22. Use ICP tool to enable erase/write protection



User system data

Access protection

FAP: A5 — Disable

System setting byte

SSB: FF

☑ nWDT_ATO_EN ☑ nDEPSLP_RST ☑ nSTDBY_RST ☑ nBOOT1

☑ nWDT_DEPSLP ☑ nWDT_STDBY

Erase and program protection bytes

EPP0-3: F9 FF FF FF

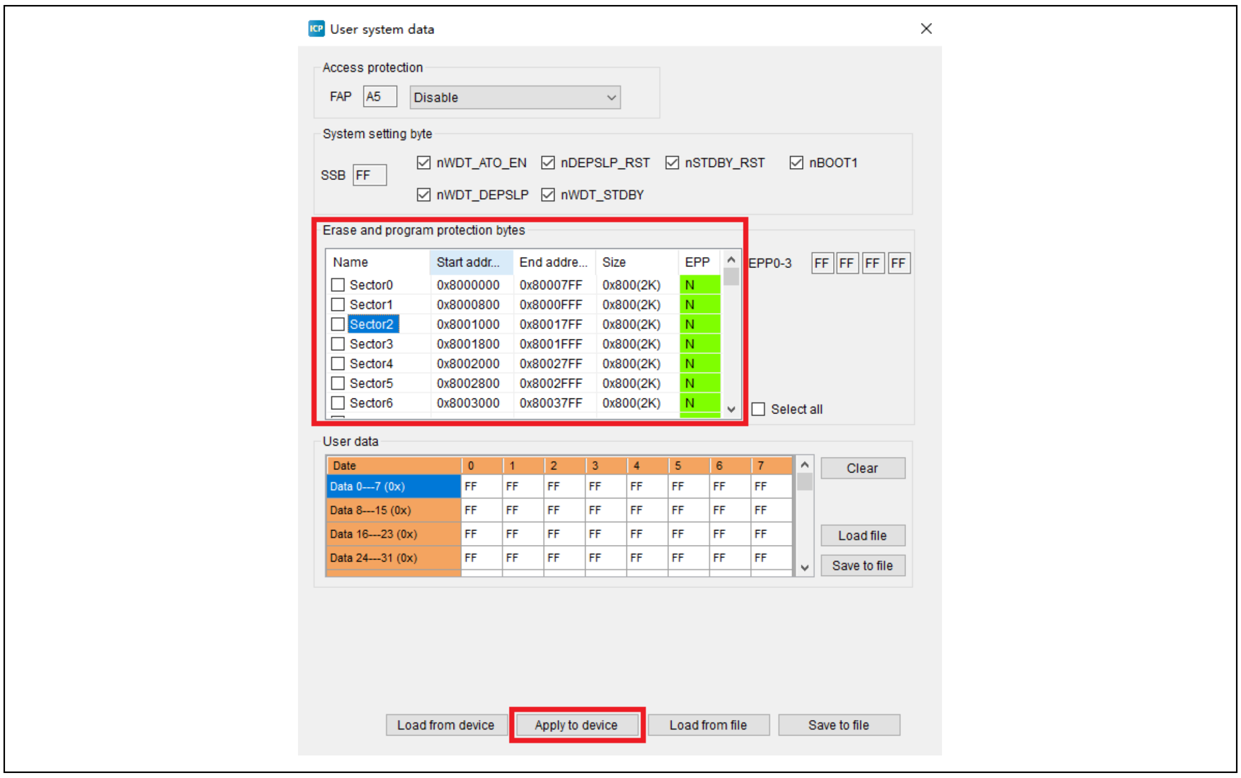
Name	Start addr...	End addre...	Size	EPP
☐ Sector0	0x8000000	0x80007FF	0x800(2K)	N
☐ Sector1	0x8000800	0x8000FFF	0x800(2K)	N
☑ Sector2	0x8001000	0x80017FF	0x800(2K)	Y
☑ Sector3	0x8001800	0x8001FFF	0x800(2K)	Y
☑ Sector4	0x8002000	0x80027FF	0x800(2K)	Y
☑ Sector5	0x8002800	0x8002FFF	0x800(2K)	Y
☐ Sector6	0x8003000	0x80037FF	0x800(2K)	N

☐ Select all

User data

Date	0	1	2	3	4	5	6	7
Data 0---7 (0x)	FF	FF	FF	FF	FF	FF	FF	FF
Data 8---15 (0x)	FF	FF	FF	FF	FF	FF	FF	FF
Data 16---23 (0x)	FF	FF	FF	FF	FF	FF	FF	FF
Data 24---31 (0x)	FF	FF	FF	FF	FF	FF	FF	FF

Figure 23. Use ICP tool to disable erase/write protection



User system data

Access protection

FAP A5 Disable

System setting byte

SSB FF ☑ nWDT_ATO_EN ☑ nDEPSLP_RST ☑ nSTDBY_RST ☑ nBOOT1 ☑ nWDT_DEPSLP ☑ nWDT_STDBY

Erase and program protection bytes

Name	Start addr...	End addre...	Size	EPP
☐ Sector0	0x8000000	0x80007FF	0x800(2K)	N
☐ Sector1	0x8000800	0x8000FFF	0x800(2K)	N
☐ Sector2	0x8001000	0x80017FF	0x800(2K)	N
☐ Sector3	0x8001800	0x8001FFF	0x800(2K)	N
☐ Sector4	0x8002000	0x80027FF	0x800(2K)	N
☐ Sector5	0x8002800	0x8002FFF	0x800(2K)	N
☐ Sector6	0x8003000	0x80037FF	0x800(2K)	N

EPP0-3 FF FF FF FF

☐ Select all

User data

Date	0	1	2	3	4	5	6	7
Data 0---7 (0x)	FF	FF	FF	FF	FF	FF	FF	FF
Data 8---15 (0x)	FF	FF	FF	FF	FF	FF	FF	FF
Data 16---23 (0x)	FF	FF	FF	FF	FF	FF	FF	FF
Data 24---31 (0x)	FF	FF	FF	FF	FF	FF	FF	FF

Note: Erase/write protection, after enabled, cannot be unlocked through erase operation.

1.2.4 Setting system memory as main memory extension

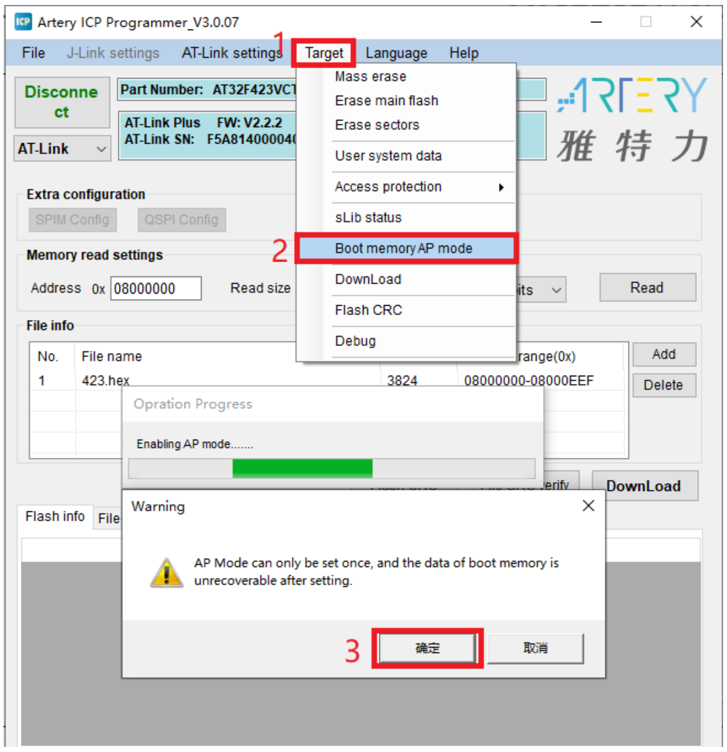
System memory is used as a boot mode by default to store microcontroller manufacturer' Startup Code. In the AT32L021 series, the system memory can also be used as a memory extension area (in AP mode) to store user-defined codes.

Note: System memory AP mode is irreversible and can only be used once, meaning that after AP mode is selected, its original BOOT mode cannot be resumed.

During product development stage, Artery ICP Programmer is used to configure system memory as memory extension, based on the following steps.

- Connect AT-Link or J-Link to AT-START-L021 evaluation board and supply power to it
- Open Artery ICP programmer, choose AT-Link/J-Link connection
- In menu bar: Target – Boot memory AP mode -- OK

Figure 24. Use ICP tool to set system memory AP mode



Part Number: AT32F423VCT

AT-Link Plus FW: V2.2.2

AT-Link SN: F5A814000040

Address 0x 08000000

No.	File name		range(0x)
1	423.hex	3824	08000000-08000EEF

Opration Progress

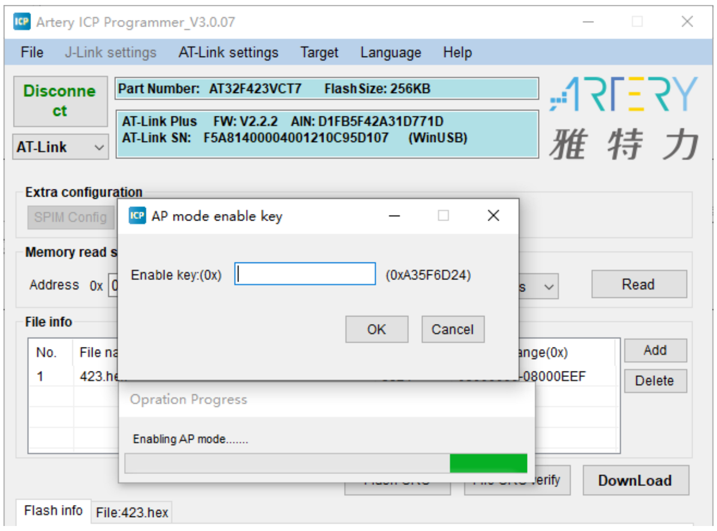
Enabling AP mode.......

Warning

AP Mode can only be set once, and the data of boot memory is unrecoverable after setting.

- To avoid unexpected wrong operations, users need enter the password “0xA35F6D24”, and then check “File info” column if the operation is successful or not.

Figure 25. System memory AP mode operating interface in ICP



Part Number: AT32F423VCT7 Flash Size: 256KB

AT-Link Plus FW: V2.2.2 AIN: D1FB5F42A31D771D

AT-Link SN: F5A81400004001210C95D107 (WinUSB)

AP mode enable key

Enable key:(0x) (0xA35F6D24)

Opration Progress

Enabling AP mode.......

File:423.hex

During mass-production stage, Artery ICP Programmer can also be used to enable system memory as memory extension area according to the following steps:

- Connect AT-Link to AT-START-L021 evaluation board and supply power to it

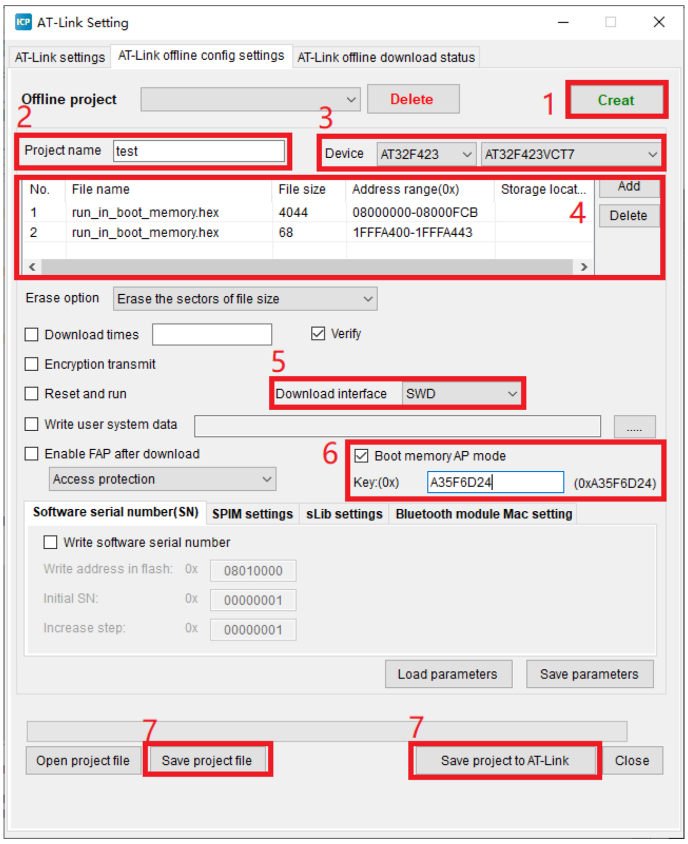
Note: The users can only choose non-EZ AT-Link as the edition of AT-Link EZ on board does not support programming offline.

- Open Artery ICP programmer, and choose AT-Link connection
- Go to menu bar: AT-Link setting -- AT-Link offline configuration setting

- Follow the steps below to generate off-line project
    1. Click "Create"
    2. Enter a project name
    3. Select a particular MCU series and MCU part number
    4. Add.hex files
    5. Choose SWD as download interface
    6. Check "Boot mode AP mode" option and enter the passkey
    7. Save project file or save project to AT-Link

For other settings, users can make corresponding configurations according to their actual needs.

Figure 26. Use ICP tool to set boot mode AP mode offline



AT-Link Setting

AT-Link settings | AT-Link offline config settings | AT-Link offline download status

Offline project: (empty) — Delete — 1 Creat

2 Project name: test

3 Device: AT32F423 — AT32F423VCT7

No.	File name	File size	Address range(0x)	Storage locat...
1	run_in_boot_memory.hex	4044	08000000-08000FCB	
2	run_in_boot_memory.hex	68	1FFFA400-1FFFA443	

4 Add / Delete

Erase option: Erase the sectors of file size

☐ Download times — ☑ Verify

☐ Encryption transmit

☐ Reset and run — 5 Download interface: SWD

☐ Write user system data

☐ Enable FAP after download — Access protection

6 ☑ Boot memory AP mode — Key:(0x) A35F6D24 (0xA35F6D24)

Software serial number(SN) | SPIM settings | sLib settings | Bluetooth module Mac setting

☐ Write software serial number

Write address in flash: 0x 08010000

Initial SN: 0x 00000001

Increase step: 0x 00000001

Load parameters — Save parameters

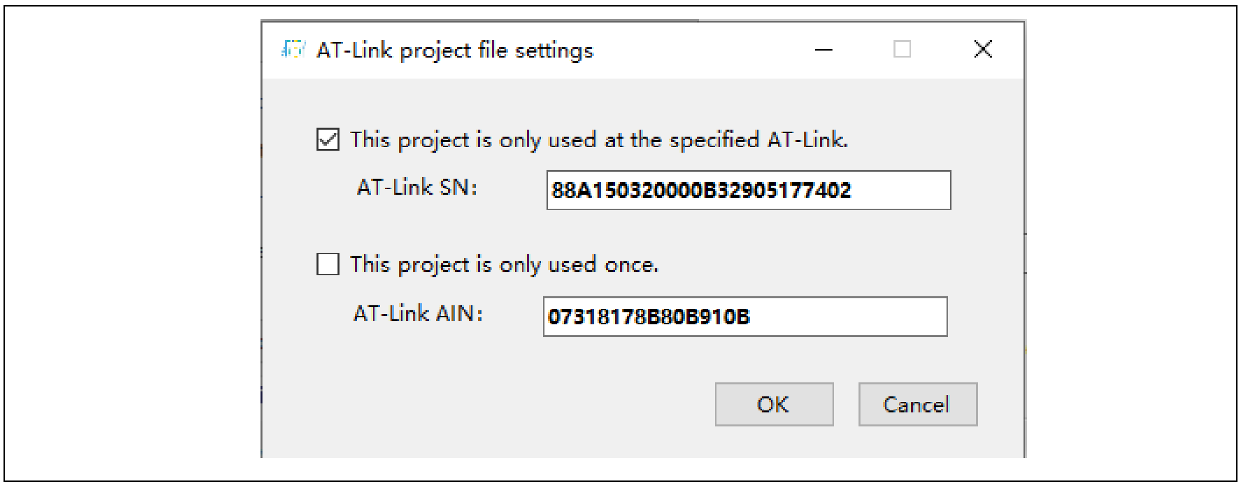
Open project file — 7 Save project file — 7 Save project to AT-Link — Close

- In Step 7, if you choose "Save project file", this project will be saved as .atcp file so that it can be loaded onto other AT-Link.

  The following dialogue window will pop out during operation. If "This project is only used at the specified AT-Link" option checked, it means that this project is bonded to a particular AT-Link and can only be used in this particular AT-Link. In this case, users need to enter a correct AT-Link serial number.

  If "This project is only used once" option is checked, it means that this project could only be used once in the same AT-Link.

Figure 27. Use ICP tool to set project file offline



AT-Link project file settings

☑ This project is only used at the specified AT-Link.

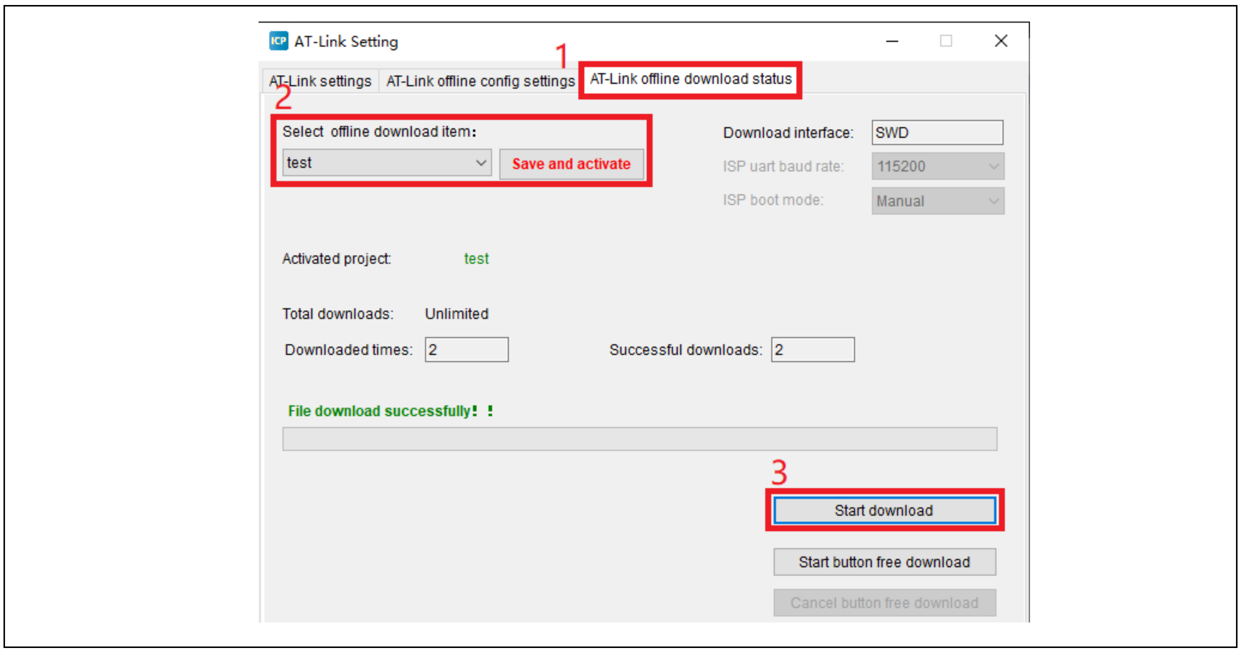
AT-Link SN: 88A150320000B32905177402

☐ This project is only used once.

AT-Link AIN: 07318178B80B910B

■ In step 7, if you choose “Save project to AT-Link” and operation is successful, in “AT-Link offline download status” option, users need to choose a project name for offline download, click “Save and activate”, and click “Start download”.

Figure 28. ICP tool to monitor offline download status



AT-Link Setting
AT-Link settings
AT-Link offline config settings
1
AT-Link offline download status
2
Save and activate
3
Start download
Start button free download
Cancel button free download

Select offline download item: test

Download interface: SWD

ISP uart baud rate: 115200

ISP boot mode: Manual

Activated project: test

Total downloads: Unlimited

Downloaded times: 2

Successful downloads: 2

File download successfully! !

■ For more information on system memory extension, please refer to the document “AN0066_config_boot_memory_as_extension_of_main_memory(AP_mode)”, which can be found at ARTERY official website → Support → AP Note → AN0066.

■ For DEMO on running user program in the system memory, please refer to BSP which is available from ARTERY official website → Product → Low power → AT32L021 series → BSP. After unzipping BSP file, you can get the desired demo under AT32L021_Firmware_Library_V2.x.x\utilities\at32l021_boot_memory_ap_demo.

1.2.5 How to distinguish AT32 MCU from other MCUs

■ Read Cortex-M series CPU ID number. This can be used to identify whether the MCU is based on M0, M0+,M1, M3 or M4 core.

Figure 29. Read Cortex ID

```
cortex_id = *(uint32_t *)0xE000ED00;// read Cortex ID

if((cortex_id == 0x410CC600) || (cortex_id == 0x410CC601))

{

        printf("This chip is Cortex-M0+.\r\n");

}

else

{

        printf("This chip is Other Device.\r\n");

}
```

■ **Reading UID and PID**

Figure 30. Read UID, PID

```
/* get AT32 MCU's UID/PID base address*/
#define DEVICE_ID_ADDR1 0x1FFFF7F3          //define Artery MCU project model, UID base address
#define DEVICE_ID_ADDR2 0x40015800          //Define MCU model, PID base address

/* it is used to store ID */
uint8_t ID[5] = {0};

/* AT32L021 MCU type table */
const uint64_t AT32_MCU_ID_TABLE[] =
{
      0x0000001010012001,  // AT32L021F4P7    16KB    TSSOP20

      0x0000001010012114,  // AT32L021C8T7    64KB    LQFP48
      …
};

 /* get UID/PID */
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 UID/PID */

 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 if it is 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 AT32L0xx.\r\n");
    }
      else
    {
       printf("This chip is Other Device.\r\n");
    }
}
```

Note: AT32L0xx series contains multiple ID codes. By organizing the obtained ID information into a 64-bit data, users are able to determine which MCU series is being used. For more information, please refer to the "Debug" section of the corresponding reference manual and AN0016_Recognize_AT32_MCU.
This AN0016 can be downloaded from ARTERY official website → Support → AP note → AN0016.

2 Frequently-asked questions for download and compiling

2.1 Program enters Hard Fault Handler

- Access data outside its boundary limit

  Locate where the program exceeds the access boundary, and change it to normal data area

- SRAM used in the program exceeds its maximum threshold

- System clock is set out of spec

2.2 Jlink unable to recognize IC in Keil project

- “FAQ0008_J-Link_cannot_ find_IC”, which can be found at ARTERY official website → Support → FAQ → FAQ0008

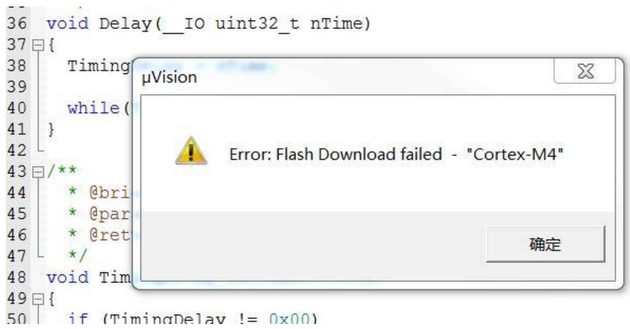
- “FAQ0132_How_to_add_Artery_MCU_into_JLINK”, which can be found at ARTERY official website → Support → FAQ → FAQ0132

2.3 Possible questions during download

2.3.1 Error: Flash Download failed–“Cortex-M0+”

A warning message below pops up during Keil debugging or downloading.

Figure 31. Flash Download failed–“Cortex- M4”



```
void Delay(__IO uint32_t nTime)
{
  Timing

  while(
}

/**
  * @bri
  * @par
  * @ret
  */
void Tim
{
  if (TimingDelay != 0x00)
```

µVision

Error: Flash Download failed - "Cortex-M4"

There are several possible factors behind this error:

- Access protection is enabled. If so, unlock access protection before download operation

- Flash algorithm file is not loaded or incorrect. If so, add a correct Flash algorithm to Flash Download location

- BOOT0 setting is incorrect. The BOOT0 must be set to 0 to boot MCU from main Flash memory

- The version of J-Link driver is older. The version 6.20C or above is recommended for J-Link

- JTAG/SWD pin is disabled. Refer to Section 2.3.5 for how to resume download.

2.3.2 No Debug Unit Device found

- Download interface is being occupied, for instance, ICP is being connected to a target device.
- JTAG/SWD connection error or not connected.

2.3.3 RDDI-DAP Error

- Compiler's optimization level is too high, for instance, the optimization level for keil AC6 compiler is "-Oz" (default), so it should be changed to -O0/-O1.
- JTAG/SWD pin is disabled. Refer to Section 2.3.5 for how to resume download.

2.3.4 ISP serial interface gets stuck during download

The ISP interface gets stuck occasionally during download so that it cannot be released

- Check if the power supply you are using is stable or not
- Use a good-quality USB-to-serial interface tool such as CH340 chip.

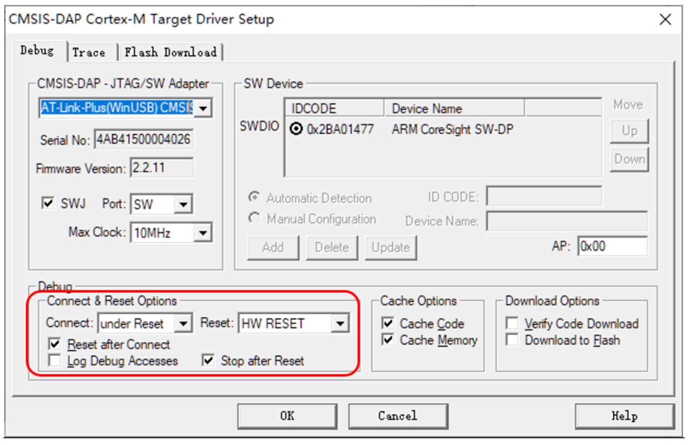
2.3.5 How to resume program download

After executing the following operations, users may not be able to download programs:

- After JTAG/SWD pin is disabled, the program download failed and JTAG/SWD device cannot be located
- After entering Standby mode and other low power modes, the program download failed and JTAG/SWD device cannot be located

The basic principle of solving these issues is to halt MCU device before program starts running. Here are some solutions for reference:

1. Change BOOT mode to "boot from memory" or "boot from SRAM", reset the device via Reset pin to erase program and resume download
2. Use ICP and AT-Link. Connect AT-Link_RST_pin to the reset pin of the device. In ICP interface, click connection to erase program and resume download.
3. Use Keil and AT-Link. Connect AT-Link_RST_pin to the reset pin of the device. In Keil's Debug interface, choose the following options marked by red box to erase program and resume download.


CMSIS-DAP Cortex-M Target Driver Setup

Debug | Trace | Flash Download

CMSIS-DAP - JTAG/SW Adapter: AT-Link-Plus(WinUSB) CMSIS
Serial No: 4AB41500004026
Firmware Version: 2.2.11
[✓] SWJ Port: SW
Max Clock: 10MHz

SW Device

	IDCODE	Device Name
SWDIO	● 0x2BA01477	ARM CoreSight SW-DP

[● Automatic Detection] Manual Configuration
ID CODE:
Device Name:
AP: 0x00

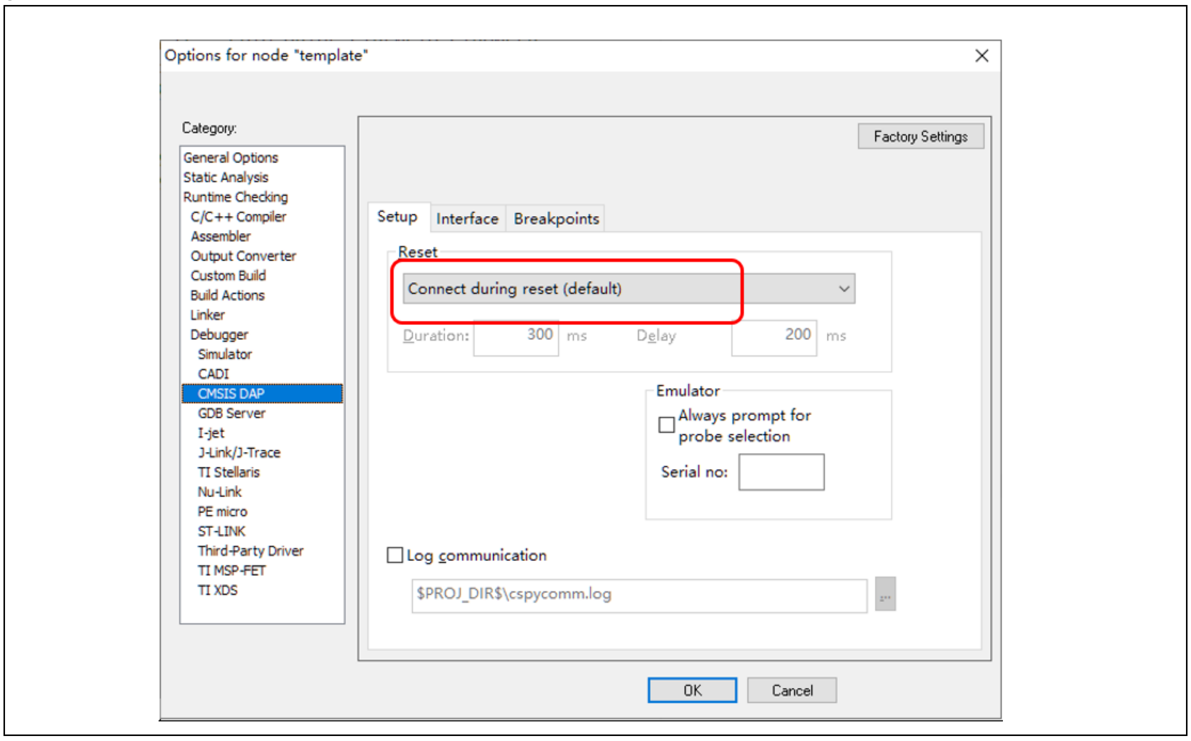
Debug

Connect & Reset Options
Connect: under Reset Reset: HW RESET
[✓] Reset after Connect
[ ] Log Debug Accesses [✓] Stop after Reset

Cache Options
[✓] Cache Code
[✓] Cache Memory

Download Options
[ ] Verify Code Download
[ ] Download to Flash

4. Use IAR and AT-Link. Connect AT-Link_RST_pin to the reset pin of the device. In IAR's CMSIS DAP interface, choose the following options marked by red box to erase program and resume download.



Options for node "template"

Category: CMSIS DAP

Setup

Reset: Connect during reset (default)

Duration: 300 ms Delay 200 ms

Emulator: ☐ Always prompt for probe selection

Serial no:

☐ Log communication

$PROJ_DIR$\cspycomm.log

3 Security Library (sLib)

3.1 Introduction

At present, as an increasing number of microcontrollers (known as MCU) require complex algorithms and middleware solutions, how to protect core algorithms and other IP codes of solution providers has emerged as one of the most important concerns in the field of MCU applications.

In response to this demand, AT32L021 series is equipped with a security library, known as sLib, with the aim of preventing important IP codes from being altered or read by end user program, so as to safeguard the rights of solution providers.

3.2 Principles

- Any part of Flash memory can be designated as a security library (sLib) with password. This sLib is used for storing critical algorithms by solution providers while the remaining memory area can be used for secondary development by end users.
- sLib is divided into a read-only area (SLIB_READ_ONLY) and an instruction area (SLIB_INSTRUCTION). Part of or the entire sLib can be set as read-only area or instruction area
- SLIB_READ_ONLY area can be read through I-Code and D-Code, but it is write-protected.
- The codes in the SLIB_INSTRUCTION area can only be fetched (only executable) by MCU through I-CODE. They cannot be read by reading operation (including ISP/ICP/debug mode or boot from internal RAM) via D-Code, so accessing SLIB_INSTRUCTION by reading operation will return all 0xFF.

- Codes and data within sLib cannot be erased until a correct password is entered. Performing write or erase operation in case of wrong password entry will trigger a warning from EPPERR=1 of the FLASH_STS register.
- Mass erase to the main Flash memory by end users will not affect codes and data in the sLib, meaning that programs and data in this secure area will not be erased.
- After sLib feature is enabled, users can unlock this protection by writing a correct password in the SLIB_PWD_CLR register. Once sLib is unlocked, the whole main memory (including contents in the sLib) will be erased. Such design is aimed at protecting program codes against leakage even if the password set by solution providers is leaked.

3.3 How to use sLib

For details on sLib, please refer to AN0146_AT32L021_Security_Library_Application_Note, which can be found at ARTERY official website → Support → AP Note → AN0146.

4 Revision history

Table 1. Document revision history

Date	Revision	Changes
2022.08.08	2.0.0	Initial release