

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 seriesbased 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:



Contents

1	Dev	elopmo	ent resources	5
	1.1	Set up	o AT32 development environment	5
		1.1.1	Debug tools and evaluation board	5
		1.1.2	Programming tools and software resources	6
		1.1.3	AT32 MCU development environment	6
		1.1.4	How to quickly migrate from one MCU to another	12
	1.2	AT32L	_021 functional overview	12
		1.2.1	Instruction prefetch buffer	12
		1.2.2	PLL clock settings	13
		1.2.3	Encryption	13
		1.2.4	Setting system memory as main memory extension	17
		1.2.5	How to distinguish AT32 MCU from other MCUs	21
2	Free	quently	v-asked questions for download and compiling	
	2.1	Progra	am enters Hard Fault Handler	22
	2.2	Jlink u	unable to recognize IC in Keil project	
	2.3	Possil	ble questions during download	
		2.3.1	Error: Flash Download failed-"Cortex-M0+"	22
		2.3.2	No Debug Unit Device found	23
		2.3.3	RDDI-DAP Error	23
		2.3.4	ISP serial interface gets stuck during download	23
		2.3.5	How to resume program download	23
3	Sec	urity Li	ibrary (sLib)	24
	3.1	Introd	uction	24
	3.2	Princi	ples	
	3.3	How t	o use sLib	25
4	Rev	ision h	listory	



List of tables

Table 1. Document revision history	
------------------------------------	--



List of figures

Figure 1. AT-START-L021 and AT-Link-EZ	5
Figure 2. AT-START-L021 evaluation board package from ARTERY official website	5
Figure 3. ICP/ISP/AT-Link-Family package from ARTERY official website	6
Figure 4. BSP resources from ARTERY official website	6
Figure 5. Keil_v5 template	7
Figure 6. Pack resources from ARTERY official website	8
Figure 7. Install ArteryTek.AT32L021_DFP	8
Figure 8. Install Keil4_AT32MCU_AddOn	9
Figure 9. Click "Pack Installer" icon in Keil	9
Figure 10. Install IAR_AT32MCU_AddOn	9
Figure 11. Keil Debug option	10
Figure 12. Settings option in Keil Debug	10
Figure 13. Keil Utilities option	11
Figure 14. IAR Debug option	11
Figure 15. IAR CMSIS-DAP option	11
Figure 16. Wait state bit in FLASH_PSR register	12
Figure 17. system_clock_config function	12
Figure 18. AT32L021 70MHz output clock configuration	13
Figure 19. Use ISP tool to enable or disable access protection	14
Figure 20. Use ISP to enable access protection	15
Figure 21. Use ISP to unlock access protection	15
Figure 22. Use ICP tool to enable erase/write protection	16
Figure 23. Use ICP tool to disable erase/write protection	17
Figure 24. Use ICP tool to set system memory AP mode	18
Figure 25. System memory AP mode operating interface in ICP	18
Figure 26. Use ICP tool to set boot mode AP mode offline	19
Figure 27. Use ICP tool to set project file offline	20
Figure 28. ICP tool to monitor offline download status	20
Figure 29. Read Cortex ID	21
Figure 30. Read UID, PID	21
Figure 31. Flash Download failed–"Cortex- M4"	22



1 Development resources

Resources download link:

Artery official website: <u>http://www.arterychip.com</u>

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.





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 \rightarrow PRODUCT \rightarrow Low power line \rightarrow AT32L0xx series \rightarrow Resources \rightarrow Evaluation board, where you can download a ZIP-format AT-START-L021 and get AT_START_L021_Vx.x\03_Documents.



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 \rightarrow Product \rightarrow Low power \rightarrow AT32L0xx \rightarrow Tool \rightarrow 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.

loc		
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
LISP_Multi-Port	In-System-Multi-Port Programming tool supporting AT32 MCU	V2.0.09

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

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 \rightarrow Product \rightarrow Low power \rightarrow AT32L0xx series \rightarrow 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_g cc/at32_ide. They are located at AT32L021_Firmware_Library_V2.x.x\project\at_start_f4xx\tem plates.

Figure 5 below shows an example of Keil_v5 project.

Figure 5. Keil_v5 template

🖃 🍄 Project: template
😑 🔊 template 🧾 🚺
🕀 🗁 user
at32f423_clock.c
at32f423_int.c (3)
main.c
🖻 🗁 bsp 🛛 🚽 🦉
at32f423_board.c
🕀 🛄 firmware
🖃 🗁 cmsis
system_at32f423.c
startup_at32f423.s 🕖
e la readme
readme.txt

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 "*at32/021_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 \rightarrow Product \rightarrow Low power \rightarrow AT32L0xx series \rightarrow 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 \rightarrow *Product* \rightarrow *Low power* \rightarrow *AT32L021 series* \rightarrow *Pack*.

Figure 6.	Pack	resources	from	ARTERY	official	website
-----------	------	-----------	------	--------	----------	---------

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

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

Figure 8. Install Keil4_AT32MCU_AddOn

Setup AT32 MCU AddOn Package to Keil MDK-ARM V2.0.6	
Select the folder where SETUP will install files	Microcontroller Tools
This Add-On will install into the following product folder. To install to this folder, press"Next", To install to a differe press "Browse" and select another folder. Destination Folder D:\Keil_y4	nt folder, Browse
-Keil MDK-AEM Setup	Next >> Cancel

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
n 📬 🗐 🕺 🖄 🗈 🛍 🤊 (*	← → 啓 穆 穆 穆 律 <u>課</u> //版 20 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 Browse	
Realtime Status 0%]
Start Cancel	

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 \rightarrow Product \rightarrow Low power \rightarrow AT32L021 series \rightarrow 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".

n ng 📓 🖉 🕺 🛍 🖄 ท ng	← → 隆 陰 陰 律 律 //』 //☆ 🖄 TMR_ICFilter 🛛 🔍 🔜 🎺 🍳	i 🕘 o 🔗 🚓
🕸 🔛 🎬 🧼 🔜 📴 template	🖂 💉 📥 🖶 🗇 🏟	
Project 🛛 🗜 💌	main.c	
Ender Stranger - Stran	Provide the second seco	×
🖃 🗁 user	Device Target Output Listing User C/C++ Asm Linker Debug Utilities	L
	Use Simulator with restrictions Settings C Use: CMSIS-DAP Debugger ULINK2/ME Contex Debugger ULINK2/ME Contex Debugger ULINK2/ME Contex Debugger	Settings
⊖- 🦢 bsp ⊕- 🗋 at32f423_board.c	Ioad Application at Startup I Imitialization File: Initialization File: Initialization File: Initialization File:	o main()
 firmware cmsis 	Restore Debug Session Settings	Edit
system_at32f423.c	✓ Breakpoints ✓ Toolbox ✓ Match Windows & Performance Analyzer ✓ Watch Windows & Performance Analyzer	, v
ie- 🦢 readme readme.txt	I ✓ Memory Display I ✓ System Viewer ✓ Memory Display I ✓ System	Viewer
	CPU DLL: Parameter: Driver DLL: Parameter:	
	SARMCM3.DLL -REMAP SARMCM3.DLL	
	Dialog DLL: Parameter: Dialog DLL: Parameter:	
	DCM.DLL pCM4 TCM.DLL pCM4	
	Manage Component Viewer Description Files	

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 \rightarrow Support \rightarrow FAQ \rightarrow FAQ0136.

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

-

SWJ Port: SW

Max Clock: 5MHz

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

Figure 12	2. Sett	ings opti	on in Keil De	bug	
Cortex-M Target Driver Setup					×
Debug Trace Flash Download	3 SW De	vice			
AT-Link (Win USB) CMSIS-DA	SWDIO	IDCODE Ox0BC11477	Device Name ARM CoreSight SW-DP	Mov Uj	/e
AT-Link (WinUSB) CMSIS-DAP Firmware Version: 2.1.2				Dov	vn

Automatic Detection

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)

Add Delete Update

C Manual Configuration Device Name:

ID CODE:

AP: 0x00

Device Target Output Listing User C/C++ Asm Linker Debug Utilities Configure Rash Menu Command © Use Target Driver for Rash Programming 2 CMSIS-DAP Debugger Settings © Update Target before Debugging Init File: Edit...

Figure 13. Keil Utilities option

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

File Edit View	Pro	ject CMSIS-DAP Tools Window Hel)		
t t 🗈 🖬 🗐	C.	Add Files	• < C	() 🕏 🖻 く 📮 >	
Workspace	6	Add Group			
Debug	[1]	Import File List			
Filos		Add Project Connection			
Elemniate		Edit Configurations	*****		
	×	Remove	ion v2	Options for node "templ	ate" X
He formsis	•~		20		
HE Firmware		Create New Project		Category	
- 🖓 🖬 user	0	Add Existing Project		General Ontions	Factory Settings
—⊞ 🗟 at32f4i	۵	Options Alt+F7	often	Static Analysis	
-⊞ i at32140		Version Control System	load fro	Runtime Checking	
L-⊞ ∎ Output	-		y autho	Assembler	Setup Download Images Extra Options Multicore Plugins
	-	Make F/	lopment	Output Converter	Driver Down to
		Compile Ctri+F/	are is	Custom Build Build Actions	
		Rebuild All	COFTRAR	Linker	CMSIS DAP v main
	a a	Clean Patch build En	INTEES O	Debugger	Simulator
		Po Po	IE FULLE	CADI	CMSIS DAP
		C-STAT Static Analysis	IDING BU	CMSIS DAP	GDB Server
	۵	Stop Build Ctrl+Break	SS FOR .	GDB Server	I-jet/JTAGjet
	~	Deventered and Datum City D		J-Link/J-Trace	TI Stellaris
		Debug without Downloading		TI Stellaris	Nu-Link
	ି	Attach to Running Target	1	PE micro	PE micro
	é	Make & Restart Debugger Ctrl+R	"at32f4	ST-LINK	Third-Party Driver
	č	Restart Debugger Ctrl+Shift+R		TI MSP-FET	TI MSP-FET
		Download	cogroup .	TI XDS	TI XDS S\debugger\ArteryTek\AT32F407xG
		SFR Setup	ogroup		
		CMSIS-Manager	Cogroup (
		Open Device Description File			OK Council
		Save List of Registers	ELAY		

Figure 14. IAR Debug option

Figure 15. IAR CMSIS-DAP option

Category:			
Static Analysis			
Runtime Checking			
C/C++ Compiler	Setup	Interface	Breakpoints
Assembler	Probe	config —	Probe configuration file
Output Converter			
Build Actions	Au	to	Override default
Linker	0.5		
Debugger	On	mine	
Simulator	OEx	olicit	CPU: Select
CADI			
CMSIS DAP	Interfa	ice	Explicit probe configuration
GDB Server	0.17		Multi-target debug system
1-Jink/1-Trace	011	AG	Maia-target debug system
TI Stellaris	⊙ SV	/D	Target number (TAP or Multidrop 0
Nu-Link			Target with multiple CBUs
PE micro			Target with multiple CPOs
ST-LINK			CPU number on 0
TIMED EET	Interface		
TIXDS			
11700	Auto d	letect 🗸	

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 \rightarrow Product \rightarrow Low power \rightarrow AT32L021 series \rightarrow 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 \rightarrow Product \rightarrow Low power \rightarrow 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 \rightarrow Product \rightarrow AP Note \rightarrow 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.

Bit 2:043	WTCYC+	0x0+²	rw₊ ³	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↔<br="">001: One wait state, for 32 MHz<system clock≤64="" mhz↔<br="">010: Two wait states, for 64 MHz<system clock≤80="" mhz↔<="" th=""></system></system></system>
-----------	--------	-------	------------------	---

Figure 16. Wait state bit in FLASH_PSR register

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

<pre>void system_clock_config(void) </pre>
<pre>/* config flash psr register */ flash_psr_set(FLASH_WAIT_CYCLE_2);</pre>
<pre>/* reset crm */ crm_reset();</pre>
crm_clock_source_enable(CRM_CLOCK_SOURCE_HEXT, TRUE);
<pre>/* wait till hext is ready */ while(crm_hext_stable_wait() == ERROR) { }</pre>



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:

 $PLL input clock = PLL reference input clock \times \frac{PLL frequency multiplication factor PLL_NS}{PLL predivision factor PLL_{MS} \times PLL postdivision setting value PLL_FR}$

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 \rightarrow Support \rightarrow AP Note \rightarrow 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 \rightarrow Product \rightarrow Low power \rightarrow AT32L0xx series \rightarrow 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 1	9.	Use	ISP	tool	to	enable	or	disable	access	protection
----------	----	-----	-----	------	----	--------	----	---------	--------	------------

File J-Link settings AT-Link settings	Target Language Help	
Disconne ct AT-Link Plus FW: V2.2.2 AT-Link SN: F5A814000040	Mass erase Erase main flash Erase boot memory Erase sectors	■ <u>"1२ГΞ२</u> Ү ■ 雅特力
Extra configuration SPIM Config QSPI Config	User system data Access protection	Enable access protection
Memory read settings	sLib status Boot memory AP mode	Enable high level access protection Disable
Address 0x 08000000 Read size	DownLoad	its v Read
	Flash CRC	
No. File name	Debug	range(0x) Add

■ 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 "Highlevel access protection", click "Start".

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

Artery ISP Programmer_V2.0.08 – X
<mark> ⁻ · I · I · I · I · I · I · I · I · I · </mark>
O Fraze O All Sectors D Edit User system data
O Bownload to device O Bisable sLib
sLib Status: DISABLE Start sector INSTR start sector Password Ox Znd sector
No. File Nume File Size Address Eange(Ox)
Confirm
Erase option download
Optimize(Rei Prinze(Rei Are you sure to enable the access protection? Write user Are you sure to enable the access protection? rogram
Address Ov
Address 0x 3 是(Y) 晋(N)
Apply User
mente voces highering arter neuroan Whés? Montering
Upload from device U:\test_binhex\4USA.hex
Firnware CRC Sector fill FF
○ Flash CBC Start sector Sector21-0x800x800 ✓ End sector Sector0-0x8000000 ✓
Frotection EXABLE Access protection
Back 2 Next Cancel Close

Figure 20. Use ISP to enable access protection

Figure 21. Use ISP to unlock access protection

Artery	/ ISP Programmer_V2.0.08		-	>
	, <u>1715</u>	<mark>?</mark> Y 雅	特ナ	7
⊖ Erase	● All ○ Sectors		🔘 Edit	User system dat
O Downl	oad to device		O Disa	ble sLib
sLib	Status: DISABLE	Start sector		\sim
		INSTR start s	ector	
Pass	word Ox	End sector		\sim
No.	File Name	File Size	Address Range(0)	add (a
				Delete
Ere	Confirm The flash memory v	will be mass erased and a	Il contents will be l	ost ,Are
Ere	Confirm The flash memory you sure to disable	will be mass erased and a e the access protection?	ll contents will be la 是(Y)	× ost ,Are 否(N)
	Confirm The flash memory you sure to disable	will be mass erased and a the access protection? 3 tor Download Access p	ll contents will be la 토(Y)	ost ,Are 香(N)
Er a Ad Er	Confirm The flash memory you sure to disable sable Access protection af d from device C:\test_bi	will be mass erased and a t he access protection? 3 ter Download Access p nhex \403A. hex	ll contents will be la 是(Y)	ost ,Are 훕(N)
Er a	Confirm The flash memory you sure to disable hable Access protection af d from device C:\test_bi are CBC Sector fi	will be mass erased and a the access protection? a ter Download Access p ahex\403A hex 11 FF	all contents will be 是(Y) rotection	× ost,Are 晋(N)
Era Ad Uplose O Firaw	Confirm The flash memory you sure to disable able Access protection af d from device C:\test_bi are CBC Sector fi CBC	will be mass erased and a t the access protection? a lex Download Access p nhex \403A. hex	all contents will be 是(Y) rotection	ost ,Are 풉(N)
Ad Uploe Firew I	Confirm The flash memory you sure to disable nable Access protection of d from device C:\test_bi are CBC Sector fi CBC Start sector Sector21-c	will be mass erased and a the access protection? a line access protection?	#(m) #(m) rotection	ost Are 쿱(N) 8000000 ~
Eri Ad Bi O Uplose Firaw Flash	Confirm The flash memory you sure to disable access protection af d from device CEC Start sector Sector fi CEC Start sector Sector21-	will be mass erased and a the access protection? a local a l	創 contents will be l 是(Y) rotection actor Sectorのの	종(N) 종(N) 종(N)

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"

		-								-
🕫 User system dat	а									
Access protection FAP A5 Di	sable			~]					
System setting byte SSB FF	;] nWDT_A] nWDT_D	TO_EN	⊠ nDi ⊻ nW	EPSLP /DT_ST	_RST DBY	⊠ nS	STDBY_I	RST	⊻ n	BOOT1
Erase and program	n protection	n bytes								
Name	Start add 0x800000 0x80008	r E 00 0: 00 0	nd addre x80007Ff x8000EEf	F Oxi F Oxi	te 800(2K) 800(2K)	EF N N	°P ^	EPP0-	3 [F9 FF FF FF
Sector2 Sector3 Sector4 Sector5	0x800100 0x800180 0x800200	00 00 00 00 00 00	(80017FF x8001FFF x80027FF x8002FFI	F Oxi F Oxi F Oxi F Oxi	800(2K) 800(2K) 800(2K) 800(2K)	Y Y Y				
Sector6	0x80030	0 0	x80037FF	F 0x	800(2K)	N	~	🗌 Se	lect a	II
User data										
Date	0	1	2	3	4	5	6	7	^	Clear
Data 815 (0x)	FF	FF	FF	FF	FF	FF	FF	FF	-11	
Data 1623 (0x)	FF	FF	FF	FF	FF	FF	FF	FF		Load file
Data 2431 (0x)	FF	FF	FF	FF	FF	FF	FF	FF		Save to file
Load	from devic	e	Apply to	device		Load	from file	e	S	ave to file

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

Access protection FAP A5 Disable System setting byte SSB FF I NWDT_ATO_EN I NDEPSLP_RST I NSTDBY_RST I NBOOT1 SSB FF I NWDT_DEPSLP I NWDT_STDBY Erase and program protection bytes Erase and program protection bytes Sector1 0x8000000 0x80007FF 0x800(2K) N Sector2 0x8001000 0x80007FF 0x800(2K) N Sector3 0x8001800 0x80007FF 0x800(2K) N Sector3 0x8001800 0x8001FFF 0x800(2K) N Sector3 0x800180 0x8001FFF 0x800(2K) N Se
FAP A5 Disable System setting byte SSB FF ✓ NWDT_ATO_EN ✓ NEPSLP_RST ✓ Stat Addr End addre Size EPP Sector0 0x800000 0x80007FF 0x800(2k) N Sector1 0x800000 0x80007FF 0x800(2k) N Sector2 0x8001000 0x80017FF 0x800(2k) N Sector3 0x8001800 0x8001FF 0x800(2k) N
System setting byte SSB FF
SSB FF nWDT_ATO_EN nDEPSLP_RST nSTDBY_RST nBOOT1 Image: start addr nWDT_DEPSLP nWDT_STDBY nWDT_STDBY Erase and program protection bytes Name Start addr End addre Size EPP Sector1 0x8000000 0x80007FF 0x800(2K) N EPP0-3 FF FF FF Sector1 0x8000800 0x80007FF 0x800(2K) N EPP0-3 FF FF FF Sector1 0x8000100 0x80007FF 0x800(2K) N EP0-3 FF FF FF Sector3 0x80017FF 0x800(2K) N EP0-3 FF FF FF
SSB FF InWDT_DEPSLP InWDT_STDBY Erase and program protection bytes Image: Start addr End addre Size EPP EPP0-3 FF_FF_FF_FF Sector0 0x8000000 0x80007FF 0x800(2K) N EPP0-3 FF_FF_FF_FF Sector1 0x8000000 0x8000FFF 0x800(2K) N EP00-3 FF_FF_FF_FF Sector1 0x8000800 0x8001FFF 0x800(2K) N E0000800 Sector10 0x80011FFF 0x800(2K) N E0000800 Sector10 Sector10 </td
Erase and program protection bytes Name Start addr End addre Size EPP EPP-3 FF Start 10 <
Erase and program protection bytes Name Start addr End addre Size EPP EPP-3 FF Startand and and and an
Name Start addr End addre Size EPP EPP0-3 FF Start addrew addrewaddrew addrew addrewaddrewaddrew addrew addrew addrewa
Sector1 0x8000000 0x8000FFF 0x800(2K) N Sector1 0x8000800 0x8000FFF 0x800(2K) N Sector2 0x8001000 0x8001FFF 0x800(2K) N Sector3 0x8001800 0x8001FFF 0x800(2K) N Sector3 0x8001800 0x8001FFF 0x800(2K) N
Sector1 0x8001000 0x80017FF 0x80012X N Sector3 0x8001800 0x8001FFF 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 V Select all
User data
Date 0 1 2 3 4 5 6 7 ^ Clear
Data 07 (0x) FF FF FF FF FF FF FF FF FF
Data 815 (0x) FF FF FF FF FF FF FF FF FF
Data 1623 (0x) FF FF FF FF FF FF FF FF FF Load file
Data 2431 (0x) FF Save to file

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

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

J. ·		, ,	
Artery I	CP Programmer_V3.0.07	-	
File J-Li	nk settings AT-Link settings	Target Language Help	
Disconn ct	e Part Number: AT32F423VC1 AT-Link Plus FW: V2.2.2	Erase main flash Erase sectors	
AT-Link	AT-Link SN: F5A81400004(User system data 推 疗力	
Extra con	figuration	Access protection	
SPIM Co	nfig QSPI Config	sLib status	
- Memory r	ead settings 2	Boot memory AP mode	
Address	0x 08000000 Read size	DownLoad its V Read	
File info		Flash CRC	
No. F	ile name	range(0x) Add	
	Opration Progress	3624 0800000-08000EEF Delete	
	Enabling AP mode		
		orify Download	
Flash info	File	×	
	AP Mode can only unrecoverable afte	be set once, and the data of boot memory is r setting.	
		3 确定 取消	

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

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

File J-Li	ink s	ettings AT-Link settings Target Language Help		
Disconn	e	Part Number: AT32F423VCT7 Flash Size: 256KB	<u> 12.</u>	Y 5-]
ct		AT-Link Plus FW: V2.2.2 AIN: D1FB5F42A31D771D	π	4+ _L
AT-Link	\sim	AI-LINK SN: F5A81400004001210C95D107 (WINDSB)	雅 :	行刀
Extra conf	figura	tion		
SPIM Cor	nfig	🖙 AP mode enable key — 🗆 🗙		
Memory re	ead s			
Address	Ox C	Enable key:(0x) (0xA35F6D24)	s ~	Read
File info		OK Cancel		
No. F	ile na		ange(0x)	Add
1 4	123.h	Opration Progress	-08000EEF	Delete
		Enabling AP mode		
			erify De	ownLoad

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 setti Offline pr	oject
Project na	me test Device AT32F423 V AT32F423VCT7 V
No. Fil 1 ru 2 ru ¢	e name File size Address range(0x) Storage locat Add n_in_boot_memory.hex 4044 0800000-08000FCB 4 Delete n_in_boot_memory.hex 68 1FFFA400-1FFFA443 >
Erase opti	on Erase the sectors of file size \checkmark
Downlo	bad times Verify
	ion transmit 5
Reset	and run Download interface SWD ~
U Write u	ser system data
	FAP after download
Software	Reg(UX) A35F6D24 (UXA5F6D24)
	e software serial number
Write a	ddress in flash: 0x 08010000
Initial S	N: 0x 00000001
Increas	ue step: 0x 00000001
	Load parameters Save parameters
	77
Open pr	oject file 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.

紀 AT-Link project file s	ettings —		×
☑ This project is on AT-Link SN:	y used at the specified AT-Link. 88A150320000B32905177402		
☐ This project is on AT-Link AIN:	y used once. 07318178B80B910B OK	Cancel	

Figure 27. Use ICP tool to set project file offline

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".

CP AT-Link Setting 1	– 🗆 X
AT-Link settings AT-Link offline config settings AT-Link of	ffline download status
Select offline download item:	Download interface: SWD
test v Save and activate	ISP uart baud rate: 115200 $$
	ISP boot mode: Manual ~
Total downloads: Unlimited Downloaded times: 2 Succe File download successfully! !	ssful downloads: 2
	3 Start download
	Start button free download

Figure 28. ICP tool to monitor offline download status

- 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[] =
                     0x000001010012001, // AT32L021F4P7
                                                                                                                                                                16KB
                                                                                                                                                                                        TSSOP20
                     0x000001010012114, // 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]<16)|(uint64\_t)ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||uint64\_t|ID[4]<16||u
<0);
     /* judge if it is AT32 MCU */
      for(i=0;i<sizeof(AT32_MCU_ID_TABLE)/sizeof(AT32_MCU_ID_TABLE[0]);i++)
      ł
               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 \rightarrow Support \rightarrow AP note \rightarrow 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 \rightarrow S upport \rightarrow FAQ \rightarrow 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"

38 Timing µVision	1		
40 while(41 } 42 □/** 43 □/** 44 * @bri	Error: Flash Download failed	- "Cortex-M4"	
45 * @ret 46 * @ret 47 */			

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 Fla sh 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.
- 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 X	
Debug Trace Flash Download CMSIS-DAP - JTAG/SW Adapter SW Device AT4Unk-PustWinUSB) CMSIE IDCODE Device Name Serial No: (4AB41500004026) SWDIO Ox2BA01477	
Firmware Version: 22.11 Image: Swide of the system Image: Swide of the system Max Clock: 10MHz Add Delete Update AP: Ox000	
Debug Connect & Reset Options Connect: under Reset: ▼ Reset: ₩ Reset: ₩ Reset: ₩ Cache Options ₩ Cache Qode ₩ Cache Memory ₩ Download Options ▶ Cache Memory ▶ Download to Bash	
OK Cancel Help	



 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 pro gram and resume download.

Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator CADI CMSIS DAP GDB Server I-jet J-Link/ PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	Factory Settings Setup Interface Breakpoints Reset Connect during reset (default) Ouration: 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 \rightarrow Support \rightarrow AP Note \rightarrow AN0146.



4 Revision history

Table 1.	Document	revision	history
----------	----------	----------	---------

Date	Revision	Changes
2022.08.08	2.0.0	Initial release



IMPORTANT NOTICE - PLEASE READ CAREFULLY

Purchasers are solely responsible for the selection and use of ARTERY's products and services, and ARTERY assumes no liability whatsoever relating to the choice, selection or use of the ARTERY products and services described herein

No license, express or implied, to any intellectual property rights is granted under this document. If any part of this document deals with any third party products or services, it shall not be deemed a license granted by ARTERY for the use of such third party products or services, or any intellectual property contained therein, or considered as a warranty regarding the use in any manner of such third party products or services or any intellectual property contained therein.

Unless otherwise specified in ARTERY's terms and conditions of sale, ARTERY provides no warranties, express or implied, regarding the use and/or sale of ARTERY products, including but not limited to any implied warranties of merchantability, fitness for a particular purpose (and their equivalents under the laws of any jurisdiction), or infringement on any patent, copyright or other intellectual property right.

Purchasers hereby agree that ARTERY's products are not designed or authorized for use in: (A) any application with special requirements of safety such as life support and active implantable device, or system with functional safety requirements; (B) any aircraft application; (C) any aerospace application or environment; (D) any weapon application, and/or (E) or other uses where the failure of the device or product could result in personal injury, death, property damage. Purchasers' unauthorized use of them in the aforementioned applications, even if with a written notice, is solely at purchasers' risk, and Purchasers are solely responsible for meeting all legal and regulatory requirements in such use.

Resale of ARTERY products with provisions different from the statements and/or technical characteristics stated in this document shall immediately void any warranty grant by ARTERY for ARTERY's products or services described herein and shall not create or expand any liability of ARTERY in any manner whatsoever.

© 2022 Artery Technology -All rights reserved