

# AN0165

Application Note

### Getting Started with AT32F423

# Introduction

This document is aimed at helping users with a quick start of developing application solutions based on AT32F423 MCU.

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:

Part number AT32F423xx



# Contents

1	Dev	elopm	ent resources	6
	1.1	Set u	p AT32 development environment	6
		1.1.1	Debug tools and evaluation board	6
		1.1.2	Programming tools and software resources	7
		1.1.3	AT32 development environment	7
		1.1.4	AT32 Work Bench	13
		1.1.5	How to quickly replace AT32F415 with AT32F423	21
	1.2	AT32	F423 functionality enhancement	
		1.2.1	Instruction prefetch buffer	21
		1.2.2	PLL clock settings	22
		1.2.3	Encryption	23
		1.2.4	Set system memory as main memory extension	26
		1.2.5	How to distinguish AT32 MCU from other MCUs	
2	FAG	for do	ownload and compiling	
	2.1	Progr	am enters Hard Fault Handler	
	2.2	J-Link	c unable to recognize IC in Keil project	
	2.3	Possi	ble problems occurred during download	
		2.3.1	Error: Flash Download failed–"Cortex-M4"	
		2.3.2	No Debug Unit Device found	
		2.3.3	RDDI-DAP Error	
		2.3.4	ISP serial interface gets stuck during download	
		2.3.5	How to resume program download	33
3	Sec	urity li	brary (sLib)	35
	3.1	Introd	luction	
	3.2	Applic	cation principles	
	3.3	How t	to use sLib	35
4	Rev	ision h	nistory	



## List of tables

Table 1. Doc	cument revision history	y
--------------	-------------------------	---



# List of figures

Figure 1. AT-START-F423 and AT-Link-EZ	6
Figure 2. AT-START-F423 evaluation board package from ARTERY official website	6
Figure 3. ICP/ISP/AT-Link-Family package from ARTERY official website	7
Figure 4. BSP resources from ARTERY official website	8
Figure 5. Keil_v5 template	8
Figure 6. Pack resources from ARTERY official website	9
Figure 7. Install ArteryTek.AT32F423_DFP	9
Figure 8. Install Keil4_AT32MCU_AddOn	10
Figure 9. Click "Pack Installer" icon in Keil	10
Figure 10. Install IAR_AT32MCU_AddOn	10
Figure 11. Keil Debug option	11
Figure 12. Debug Settings in Keil	11
Figure 13. Keil Utilities option	12
Figure 14. IAR Debug option	12
Figure 15. IAR CMSIS-DAP option	13
Figure 16. AT32 Work Bench resources from ARTERY official website	14
Figure 17. dpkg command in Linux	14
Figure 18. Graphical installation	15
Figure 19. AT32 Work Bench getting started page	16
Figure 20. Project configuration	16
Figure 21. "Mode and Configuration" window	17
Figure 22. Mode setting	17
Figure 23. Parameters settings	17
Figure 24. GPIO settings	18
Figure 25. DMA settings	18
Figure 26. NVIC settings	18
Figure 27. Left click / right click effect	19
Figure 28. Clock configuration window	19
Figure 29. Code view	20
Figure 30. Project manager	20
Figure 31. Project file directory	21
Figure 32. Wait state bit in FLASH_PSR register	21
Figure 33. system_clock_config function	22



# Getting Started with AT32F423

Figure 34.	AT32F423 150 MHz output clock configuration	22
Figure 35.	AT32 PLL auto step-by-step frequency switch configuration example	23
Figure 36.	Use ISP tool to enable or disable access protection	24
Figure 37.	Use ISP to enable access protection	24
Figure 38.	Use ISP to unlock access protection	25
Figure 39.	Use ICP tool to enable erase/program protection	26
Figure 40.	Use ICP tool to disable erase/program protection	26
Figure 41.	Use ICP tool to set boot memory AP mode	27
Figure 42.	Boot memory AP mode operating progress	28
Figure 43.	Use ICP tool to set offline boot memory AP mode	29
Figure 44.	Use ICP tool to set offline project programming	29
Figure 45.	ICP tool to monitor offline download status	30
Figure 46.	Read Cortex ID	30
Figure 47.	Read UID and PID	31
Figure 48.	Flash Download failed"Cortex- M4" during download	32
Figure 49.	Configure debug settings	34
Figure 50.	Configure CMSIS DAP settings	34



### **1** Development resources

#### **Resources download link:**

https://www.arterychip.com/en/index.jsp

### 1.1 Set up AT32 development environment

### **1.1.1** Debug tools and evaluation board

The AT32F423 evaluation board comes with AT-Link-EZ for the debug purpose.

The picture of AT-Link-EZ is marked with red rectangle in Figure 1 below. It can also be separated from the board and works with other circuit boards. The debug tool can be used for several purposes such as IDE online debugging and programming, as well as USB-to-serial interface.

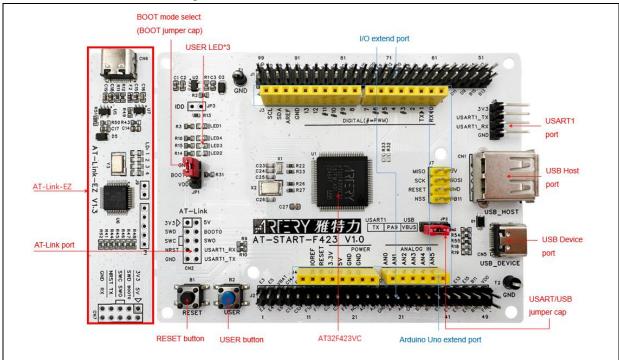


Figure 1. AT-START-F423 and AT-Link-EZ

Note: For details on AT-START-AT32F423 evaluation board, refer to the "UM\_AT\_START\_F423\_Vx.x" that is available from ARTERY's official website. You can go to <u>ARTERY official website</u>  $\rightarrow$  PRODUCTS $\rightarrow$  Value line  $\rightarrow$  AT32F423 series  $\rightarrow$  Resources  $\rightarrow$  Evaluation Board, where you can download a ZIP-format AT-START-F423 and get VAT\_START\_F423\_Vx.x\03\_Documents.

#### Figure 2. AT-START-F423 evaluation board package from ARTERY official website

Download	Description	Versior
AT-START-F423	AT32F423 evaluation board supporting Arduino standard interfaces	V1.0



### **1.1.2** Programming tools and software resources

- ARTERY programming tools and software: AT-Link /AT-Link+ /AT-Link-Pro /AT-Link-ISO /AT-Link-EZ, and ICP/ISP
- 3<sup>rd</sup> 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 <u>ARTERY official website</u>  $\rightarrow$  SUPPORT  $\rightarrow$  Hardware Development Tool and 3<sup>RD</sup> Party Writer.

- For ICP usage instructions, please refer to *UM\_ICP\_Programmer*. Path: <u>ARTERY official</u> <u>website</u>→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: <u>ARTERY official</u> <u>website</u>→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: <u>ARTERY official website</u>→PRODUCTS→Value line→AT32F4xx; download and unzip AT-Link-Family to get the "AT\_Link\_EN\_Vx.x.x\05\_Documents\UM0004\_AT-Link\_ User\_Manual\_EN\_Vx.x.x".

Download 🗘	Description	Version
<ul> <li>▶ 01_AT32 IDE</li> <li>▲ AT32 IDE_Linux</li> <li>▲ AT32 IDE_Win</li> </ul>	AT32 IDE User Manual A software development environment for cross-platform ARM embedded system based on Eclipse development supporting AT32 MCU	V1.0.09
<ul> <li>D2_AT32 Work Bench</li> <li>AT32 Work Bench_Linux</li> <li>AT32 Work Bench_Win</li> </ul>	AT32 Work Bench The AT32 Work Bench can generate initialization C code and corresponding IDE project through MCU graphical configuration.	V1.0.07
<ul> <li>03_AT32 IDE_Project Generate_Linux</li> <li>AT32 IDE_Project Generate_Win</li> </ul>	AT32 IDE Project Generate	V1.0.01
04_AT-Link Family AT-Link Family	AT-Link Family Emulation and online/offline programming tools supporting AT32 MCU	V2.1.2
<ul> <li>D5_AT-Link Console</li> <li>AT-Link Console_Linux</li> <li>AT-Link Console_Win</li> </ul>	AT-Link Console In-Circuit-Programming Console tool supporting AT32 MCU	V3.0.8
▶ 06_ICP ▲ ICP	ICP Programmer In-Circuit-Programming tool supporting AT32 MCU	V3.0.13
☑ 07_ISP ▲ ISP	ISP Programmer In-System-Programming tool supporting AT32 MCU	V2.0.12

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

### 1.1.3 AT32 development environment

### 1.1.3.1 Template project

The frequently-used IDE template projects are included in ARTERY's firmware BSP. You can get these resources by visiting <u>ARTERY official website</u>  $\rightarrow$  PRODUCTS  $\rightarrow$  Value line  $\rightarrow$  AT32F423 series  $\rightarrow$  BSP.



#### Figure 4. BSP resources from ARTERY official website

SP		
Download	Description	Version
🛓 Firmware Library	AT32F423 firmware library BSP user guide	V2.0.4

BSP offers such template projects as at32\_ide/eclipse\_gcc/Keil\_v5/Keil\_v4/IAR\_6.10/IAR\_7.4/ IAR\_8.2/IAR\_9.3, which are stored at AT32F423\_Firmware\_Library\_V2.x.x\project\at\_start\_f4xx\ templates. Just simply open the desired folder and IDE project. Figure 5 below shows an example of Keil\_v5 project.

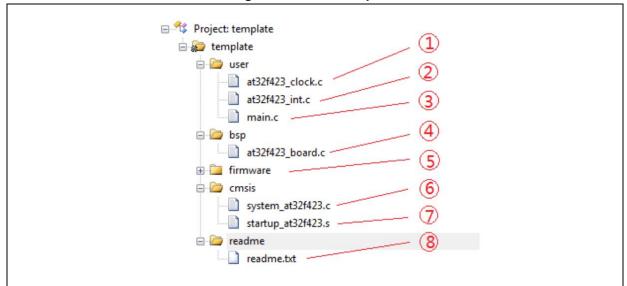


Figure 5. Keil\_v5 template

Taking Keil\_v5 template as an example, it mainly contains the following items:

- at32F423\_clock.c: it is a clock configuration file defining the default clock frequency and clock path
- ② at32F423\_int.c: : it refers to an interrupt file that contains the default process of handling some core interrupts
- 3 main.c: it refers to the main code file
- (4) **at32F423\_board.c:** it refers to the board configuration file that defines common hardware such as buttons and LEDs on AT-START evaluation board
- (5) firmware: it contains "at32F423\_xx.c" that is used as a driver file for peripherals
- 6 system\_at32F423.c: system initialization file
- ⑦ startup\_at32F423.s: a startup file
- (8) **readme.txt:** a read-me txt file that describes some application functions, configuration method and application notes relating to a template project.

In addition to "Templates", a large number of examples are included in BSP for users' reference. These example codes are stored at AT32F423\_Firmware\_Library\_V2.x.x\project\at\_start\_f4xx \examples.

Note: For details on BSP, refer to Section 4 of the document "AT32F423\_firmware\_BSP&Pack\_user\_guide". This guideline is available from <u>ARTERY official website</u>  $\rightarrow$  PRODUCTS  $\rightarrow$  Value line  $\rightarrow$  AT32F423 series  $\rightarrow$  BSP (unzip and get "\AT32F423\_Firmware\_Library\_Vx.x.x\document").



### 1.1.3.2 Pack installation

The installation of Pack is required to add AT32 MCU part number in Keil/IAR. This Pack can be downloaded from <u>ARTERY official website</u>  $\rightarrow$  PRODUCTS  $\rightarrow$  Value line  $\rightarrow$  AT32 AT32F4xx.

Download 👘	Description	t Versio
<b>≛</b> 1_Keil 4	Supports AT32 MCU to run in Keil MDK	V2.2.
<b>≛</b> 2_Keil 5	Supports AT32 MCU to run in Keil MDK	V2.3.
<b>≛</b> 3_IAR	Supports AT32 MCU to run in IAR EWARM	V2.2.
▲ 4_Segger	Supports Segger tools to identify AT32 MCU	V2.0.

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

Regarding the Keil compiling system, the keil 4.74 or above V5.23 is recommended. For Kei\_v5 system, users need first unzip "Keil5\_AT32MCU\_AddOn" and then install the "ArteryTek.AT32F423 DFP".

For Keil\_v4 system, users directly install "Keil4\_AT32MCU\_AddOn". By default, the installation path of Keil can be automatically recognized when installing. In case of recognition failure, it is necessary for users to manually choose an installation path for Keil.

#### Figure 7. Install ArteryTek.AT32F423\_DFP

Pack Unzip: ArteryTek AT32F423_DFP 2	.0.0	×
Welcome to Keil Pack Unzip Release 12/2022		
This program installs the Software Pack: ArteryTek AT32F423_DFP 2.0.0 ArteryTek AT32F423 Series Device Suppor	Drivers	
- Destination Folder D:\Keil_v5\APM\PACK\ArteryTek\AT32 Keil Pack Unzip		Cancel

#### Figure 8. Install Keil4\_AT32MCU\_AddOn

Setup AT32 MCU AddOn Package to Keil MDK-ARM V2.0.6	ARM'KEIL'
Select the folder where SETUP will install files This Add-On will install into the following product folder. To install to this folder, press"West". To install to a differ press "Browse" and select another folder.	Microcontroller Tools
Destination Folder D:\Keil_v4	Browse
<< Back	Next >> Cancel

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

Figure 9. Click "Pack Installer" icon in Keil

🖫 μVision		
File Edit View Project Flash Debu	Peripherals Tools SVCS Window Help	p
n 📬 🖉 🧳 🕺 🛍 🖄 🕫	←→ をなな夜 律連進ル	🕼 🖄 ADC1_2_IRQ 🗸 🗸
🖉 🖾 🖉 🥔 🔛 🛄	🖉 🖉 🖨 着 🐄 💆 🙆	
Project 📮 🗵		Pack Installer
		Install or update Software Packs that contain Software Components

If IAR compiling system is to be used, its IAR7.0 or IAR6.1 above versions are recommended. While installing the "IAR\_AT32MCU\_AddOn", the installation path of Keil can be automatically recognized. In case of recognition failure, it is necessary for users to manually choose an installation path for IAR.

#### Figure 10. Install IAR\_AT32MCU\_AddOn

🙀 Setup AT32 MCU AddOn Package to IAR V2.0.5 X
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 "AT32F423\_firmware\_BSP&Pack\_user\_guide". This guideline is available from <u>ARTERY official website</u>  $\rightarrow$  PRODUCTS  $\rightarrow$  Value line  $\rightarrow$  AT32F423 $\rightarrow$  BSP (unzip and get the "\AT32F423\_Firmware\_Library\_Vx.x.x\document").



### 1.1.3.3 Debug and download with AT-Link

If AT-Link is to be used in Keil environment, click "**Debug**", and then choose "**CMSIS-DAP Debugger**".

🗇 🔛 🎬 🥔 🔜 🙀 template	🚽 🔊 📥 🗟 🗇 🍘
Project 🛛 📮 🗵	main.c
Sector Strate → Sector Str	W Options for Target 'template'       X         Device Target Output Listing User       C/C++       Asm       Linker Debug       Utilities
	C Use Simulator with restrictions Settings     C Use: CMSIS-DAP Debugger ▼ Settings     UIINK2/ME Cortex Debugger     MUNK2/ME Cortex Debugger
<ul> <li>bsp</li> <li>at32f423_board.c</li> <li>firmware</li> <li>mis</li> <li>system_at32f423.c</li> <li>startup_at32f423.s</li> <li>readme</li> </ul>	Image: Construction at Startup       Image: Construction at Startup
	CPU DLL:     Parameter:     Driver DLL:     Parameter:       SARMCM3.DLL     -REMAP     SARMCM3.DLL     SARMCM3.DLL       Dialog DLL:     Parameter:     Dialog DLL:     Parameter:       DCM.DLL     \pCM4     TCM.DLL     \pCM4
	SARMCM3.DLL         -REMAP         SARMCM3.DLL           Dialog DLL:         Parameter:         Dialog DLL:         Parameter:

Next, click "**Settings**" to enter "**Cortex-M Target Driver Setup**" window, as shown in Figure 12 below.

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

Note: For more information on WinUSB B, refer to "FAQ0136\_How\_to\_use\_AT-LINK\_WinUSB\_EN\_V2.0.0". Path: <u>ARTERY official website</u>→SUPPORT→FAQ→FAQ0136.

- 2. In "Port" option, select "SW" and then tick "SWJ";
- 3. As shown in step 3 below, the ARM SW-DP debug module is recognized.

Cortex-M Target Driver Setup		×
Debug Trace Flash Download CMSIS-DAP - JTAG/SW Adapter AT-Link(WinUSB)CMSIS-DA Any AI-Link CMSIS-DAP AT-Link(WinUSB)CMSIS-DAP Himware Version: 2.1.2	-SW Device	Move Up Down
Max Clock: 5MHz	Automatic Detection ID CODE:     Manual Configuration Device Name:     Add Delete Update	AP: 0x00

Figure 12. Debug Settings in Keil

In "**Utilities**" option, first untick "**Use Debug Driver**" (see step 1), then select "**CMSIS-DAP Debugger**" in Step 2, and finally re-tick Step 1 option (noted that this step 1 must be first unticked before being ticked again later).

Figure 13. Keil Utilities option

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

If AT-Link is to be used in IAR environment, click "**Project**", choose "**Options**", go to "**Debugger**" and select "**CMSIS-DAP**", and then tick "**SWD**" option.

1 D O I		Add Files	- < (	3, > \$ ⊨≡ < 0	> 🖻 🖻 💼 🗰 🖷 💿 📀 📮 🛲 📜
pace		Add Group			· · · · ·
1	[1]	Import File List			
		Add Project Connection			
template ·		Edit Configurations	*****		
psp	×	Remove	ion v2	Options for node "tem	plate" ×
cmsis	+>		20		
firmware readme			f ma	Category:	
user	0	Add Existing Project		General Options	Factory Settings
⊞ 🗟 at32f4		Options Alt+F7		Static Analysis	
⊞ 🗟 at32f4t ⊞ 🗟 main.c		Version Control System	oftware Load fro	Runtime Checking	
±li⊚imain.c ∎Output			y autho	C/C++ Compiler Assembler	Setup Download Images Extra Options Multicore Plugins
			rare and	Output Converter	
		Compile Ctrl+F7	lopment ware is	Custom Build	Driver I Run to
	9			Build Actions Linker	CMSIS DAP v main
	₫		SOFTWAR	Debugger	Simulator
	ê	Batch build F8	IE FULLE	Simulator	CADI
		C-STAT Static Analysis	JTORY OR	CADI CMSIS DAP	GDB Server
	۵	Stop Build Ctrl+Break	IDING BU ISS FOR	GDB Server	I-jet/JTAGjet
	G	Stop build Ctrl+Break		I-jet/JTAGjet J-Link/J-Trace	J-Link/J-Trace
	0	Download and Debug Ctrl+D	*****	TI Stellaris	TI Stellaris
	٠	Debug without Downloading		Nu-Link	PE micro
	۲	Attach to Running Target	"at32f4	PE micro ST-LINK	ST-LINK
	C	Make & Restart Debugger Ctrl+R	"at32f4	Third-Party Driver	Third-Party Driver TI MSP-FET
	c	Restart Debugger Ctrl+Shift+R	ogroup	TI MSP-FET	TI XDS S\debugger\ArteryTek\AT32F407xG
		Download	1.1	TI XDS	
		SFR Setup	1	L	
		CMSIS-Manager	ogroup		
		-			
		Open Device Description File Save List of Registers			OK Cancel

Figure 14. IAR Debug option

Figure 15. IAR CMSIS-DAP option

Category: General Options		
Static Analysis		
Runtime Checking		
C/C++ Compiler	Setup Interface	Breakpoints
Assembler	Probe config	Probe configuration file
Output Converter Custom Build		-
Build Actions	Auto	Override default
Linker	O From file	
Debugger	0	
Simulator CADI	⊖ Explicit	CPU: Select
CMSIS DAP	Interface	Explicit probe configuration
GDB Server	Interface	Explicit probe configuration
I-jet/JTAGjet	OJTAG	Multi-target debug system
J-Link/J-Trace		Target number (TAP or Multidrop 0
TI Stellaris	SWD	
Nu-Link PE micro		Target with multiple CPUs
ST-LINK		CPU number on 0
Third-Party Driver		CPU number on
TI MSP-FET	Interface	
TI XDS	Auto detect 🗸	

Note: The document AT32F423\_firmware\_BSP&Pack\_user\_guide provides details on Flash algorithm, MCU product series replacement and J-Link. Thus they are not repeated hereof. This document is located at <u>ARTERY official website</u>→PRODUCTS→Value line→AT32F423→BSP (unzip and get "VAT32F423\_Firmware\_Library\_Vx.x.x\document").

### 1.1.4 AT32 Work Bench

### 1.1.4.1 Introduction

This section introduces how to use AT32 Work Bench. Users can use the AT32 Work Bench to graphically configure MC, generate initialization C code and the corresponding IDE project, helping reducing development workload, duration and cost.

Environmental requirements

Software:

Windows: Windows 7 and above

Linux: Ubuntu or Fedora that support x86\_64

Hardware:

At least 2GB RAM and 4GB hard drive space

Note: For details about AT32 Work Bench, please refer to UM\_AT32\_Work\_Bench.pdf. Path: <u>ARTERY official</u> <u>website</u>→SUPPORT→Tool. Click <u>https://www.arterychip.com/en/support/index.jsp?index=5</u>, as shown below.



Г

Figure 16. AT32 Work Bench resources from ARTERY official website

CU
----

### 1.1.4.2 Installation

Windows: Run the executable AT32\_Work\_Bench.exe directly, without the need for installation. Linux: Ubuntu 16.4 and above are supported.

Installation steps:

 Enter the following "dpkg" command in the terminal for setup, as shown in Figure 17: sudo dpgk - i AT32\_Work\_Bench\_Linux-x86\_64\_Vx.x.xx.deb

Figure 17. dpkg command in Linux

л.	artery@artery-virtual-machine: ~	Q = - • 😣
	virtual-machine:~\$ sudo dpkg -i AT32_Work	k_Bench_linux_amd64.deb
[sudo] passwor		<b>b</b>
	lously unselected package at32-work-bench	
	ase 146852 files and directories curr	
Preparing to u	<pre>inpack AT32_Work_Bench_linux_amd64.deb</pre>	
Unpacking at32	-work-bench (1.0.0)	
Setting up at3	2-work-bench (1.0.0)	
Processing tri	ggers for gnome-menus (3.36.0-lubuntul)	
	ggers for desktop-file-utils (0.24-1ubuni	
	ggers for mime-support (3.64ubuntu1)	
	virtual-machine:~\$	
ar eer yeer eer y-	vereaue nachener y	

Graphical installation

Copy the "AT32\_Work\_Bench\_Linux-x86\_64\_Vx.x.xx.deb" to Linux and double click. Then, click on the "**Install**", as shown in Figure 18.

#### Figure 18. Graphical installation

<	at32-work-bench	•••
	<b>32-work-bench</b> 32 Work Bench Project	
Install		
AT32 Work Bend	h Project	
Details		
Version	1.0.0	
Updated	05/05/2023	
License	Proprietary	
Source	AT32_Work_Bench_linux_amd64.deb	
Download Size	0 bytes	

After the installation is complete, click the "**All Programs**" button at the bottom of the left taskbar, find and click on the "AT32 Work Bench" in the program list to start AT32 Work Bench.

### 1.1.4.3 Project configuration

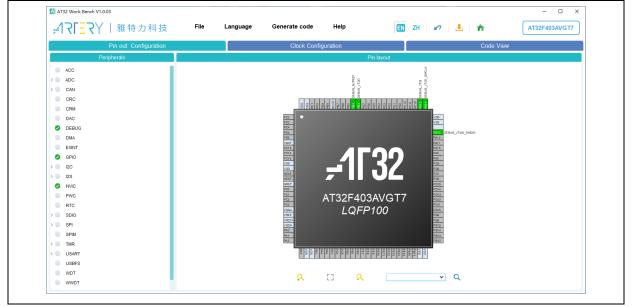
This section uses AT32F403AVGT7 to create an USART project to show how to use AT32\_Work\_Bench to generate initialization C code and the corresponding IDE project.

The getting started page is the first window that opens when AT32 Work Bench is started. It includes three options, i.e., "Start a New Design", "Open an Existing Design" and "Recent Designs".

#### Figure 19. AT32 Work Bench getting started page

		Language: English N
Ŭ		h, please use one of the below options to start or continue your design.
r Start a new De	esign	
Device Serials:	AT32F403A	▼ AT32F403A high-performance microcontrollers, powered by 32-bit ARM® Cortex®-M4
Device:	AT32F403AVGT7	<ul> <li>core, utilize advanced process to achieve 240 MHz computing speed. The embedded single precision floating-point unit (FPU) and digital signal processor (DSP), rich</li> </ul>
Package:	LQFP100	peripherals and flexible clock control mechanism can meet an extensive range of applications. The superior memory design supports up to 1 MB Flash memory and 224 KB SRAM, with the excellent Flash access zero wait far beyond the same level of the
Flash:	1024KB	chip industry. In addition to highly efficient computing performance, AT32F403A series also imports the Security Library to support the use of password to protect the
SRAM:	Default: 96KB Max: 224KB	specified program area, in which IDH programs the core algorithm and provides to
	New	
r Open an Existi	ng Design	
Open	a saved .ATWP file to continu	le veur design
opon	a saved service in continu	ae your design.
	15	
Recent Desigr		
Recent Desigr		
		/GT7_WorkBench/AT32A403AV/GT7_WorkBench_ATWP

Select the desired MCU and click to enter the project configuration window. The project configuration page contains "Pin out configuration", "Clock configuration" and "Code view".



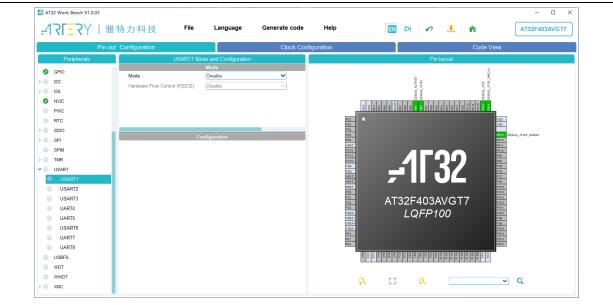


(1) **Pin out configuration:** It is used to configure the MCU peripherals and pins, and it includes "Peripherals", Mode and configuration" and "Pin layout".

Select a peripheral to open the "Mode and Configuration" window where users can set the mode and relevant parameters, as well as MCU pins.

One pin is allowed to be used by different peripherals and for multiple functions. Therefore, once the mode is configured, the most suitable peripheral pin layout will be set automatically. For example, Figure 21 shows the "Mode and Configuration" of USART1.





#### Figure 21. "Mode and Configuration" window

Follow the below steps to configure USART1.

Peripheral mode

#### Figure 22. Mode setting

USART1 Mo	ode and Configuration	
	Mode	
Mode	Asynchronous	~
Hardware Flow Control (RS232)	Disable	~

As shown in Figure 22, when the USART1 asynchronous mode is selected, PA9 and PA10 are automatically mapped onto "USART1\_TX" and "USART1\_RX", respectively.

Parameters settings

#### Figure 23. Parameters settings

Parameters Settings	GPIO Settings	NVIC Settings	DMA Settings	
➤ Basic Parameters				
Baud Rate	(1831-7500000)	115200 Bits/s		
Data bit nu	m	8 Bits (including	Parity)	~
Parity sele	ction	None		~
STOP bit n	um	1		~
➤ Advanced Paramete	ers			
Data Direct	tion	Receive and Tra	nsmit	~

As shown in Figure 23, the "Parameters Settings" window contains USART1 related parameters, including basic parameters such as baud rate and data bit number, and advanced parameters such as data transfer direction.

GPIO settings

Pin Name	Signal on Pin	Output level	GPIO type	Pull type	GPIO mode	Driver capability	Label	Modified
PA10	USART1_RX	n/a	n/a	Pull-none	Input mode	n/a		N
PA9	USART1_TX	n/a	Push Pull	Pull-none	Mux function mode	Moderate		N
		Output level GPIO type			n/a		<ul><li>✓</li><li>✓</li></ul>	
			ty		n/a Pull-r	none mode		

#### Figure 24. GPIO settings

As shown in Figure 25, the "GPIO Settings" windows contains available GPIOs, such as USART1\_TX" and "USART1\_RX".

DMA settings

Parameters Settings GPIO Sett	ings NVIC Settings	DMA Settings	
DMA Request	Channel	Direction	Priority
USART1_RX	DMA1 Channel 1	Peripheral To Me	mory Low
USART1_TX	DMA1 Channel 2	Memory To Perip	heral Low
DMA Request Parameters Peripheral Incre	ement	eripheral Inc Disable	~
Memory Increm	nent N	lemory Inc Enable	~
Peripheral Data	Alignment	yte	~
Memory Data A	lignment	yte	~
			~

Figure 25. DMA settings

As shown in Figure 25, the "DMA Settings" window contains configurable DMA requests, such as DMA channels and DMA request parameters of "USART1\_TX" and "USART1\_RX".

NVIC settings

#### Figure 26. NVIC settings

Parameters Settings GPIO Settings	NVIC Settings DMA Set	tings	
NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
DMA1_Channel1_IRQ	$\checkmark$	0	0
DMA1_Channel2_IRQ	$\checkmark$	0	0
USART1_IRQ	$\checkmark$	0	0

As shown in Figure 26, the "NVIC Settings" window contains configurable interrupts of the selected peripheral. When DMA channel is enabled, the corresponding DMA channel interrupt can be configured. The "preemption priority" and "sub priority" are configured in the "NVIC Mode and configuration" window.

The pin layout of the selected package (such as LQFP48, QFN32 and TSSOP20) is displayed in graphic. Each pin is represented by its name (such as PA9), configuration status, and current signal distribution.



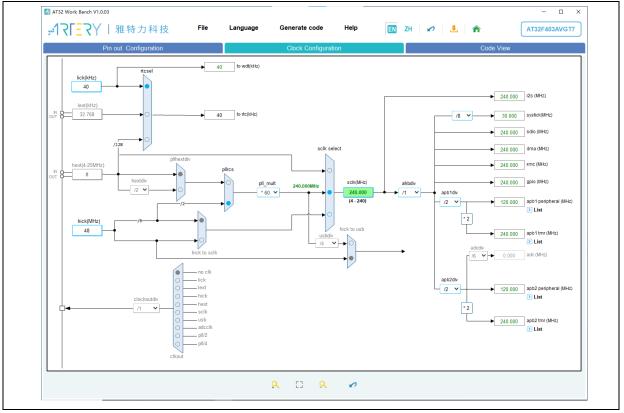
In the pin layout, left click on the pin (except for pins in fixed mode), and then a right-click menu pops up, showing configurable signals for the pin.

Right click on the configured pin, and then an "Enter Label" button pops up. Users can click on the button to specify a custom label for this signal.

PA10	USART1_RX	PA10	USART1_RX
PA9	Reset_State	PA9	Enter Label
PA8	USART1_TX	PA8	
PC9	I2C3_SMBA	PC9	1
	TMR1_CH2		
PC8	EVENTOUT	PC8	
PC7	GPIO_Input	PC7	
PC6	GPIO_Output GPIO_Analog	PC6	
PD15	EXINT9	PD15	1



(2) Users can configure the clock path and parameters in the Clock Configuration window, and use drop-down menus and input boxes to modify the actual clock tree configuration to meet application requirements. The Clock Configuration window is shown in Figure 28.



#### Figure 28. Clock configuration window

Note 1: Set the "LEXT" mode in the "CRM Mode and Configuration" window to enable LEXT. Note 2: Set the "HEXT" mode in the "CRM Mode and Configuration" window to enable HEXT. Note: Tick "Clock Output" in the "CRM Mode and Configuration" window to enable clockout.

(3) Click on the "Code View" to generate code automatically. Code files are listed in the left, and the corresponding code is shown in the right window.

#### Figure 29. Code view

AT32 Work Bench V1.0.03	File Language Generate code Help 🖾 7H 🛷 👢 🕋 attagrada.vcg.tz
<b>,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,</b>	File Language Generate code Help 🔣 ZH 🖍 🛃 AT32F403AVGT7
Pin out Configuration	Clock Configuration Code View
	1 /* add user code begin Header */
main.c	2 /**
	3 ************************************
at32f403a 407 wk config.c	5 * @brief main program
	6 ************************************
at32f403a 407 wk config.h	8 * 9 * The software Board Support Package (BSP) that is made available to
	<ul> <li>9 * The software Board Support Package (BSP) that is made available to</li> <li>10 * download from Artery official website is the copyrighted work of Artery.</li> </ul>
- +225402- 407	11 * Artery authorizes customers to use, copy, and distribute the BSP 12 * software and its related documentation for the purpose of design and
at32f403a_407_int.c	13 * development in conjunction with Artery microcontrollers. Use of the
	14 * software is governed by this copyright notice and the following disclaimer. 15 *
at32f403a_407_int.h	16 * THIS SOFTWARE IS PROVIDED ON "AS IS" BASIS WITHOUT WARRANTIES,
	17 * GUARANTEES OR REPRESENTATIONS OF ANY KIND. ARTERY EXPRESSLY DISCLAIMS, 18 * TO THE FULLEST EXTENT PERMITTED BY LAW, ALL EXPRESS, IMPLIED OR
at32f403a_407_conf.h	19 * STATUTORY OR OTHER WARRANTIES, GUARANTEES OR REPRESENTATIONS, 20 * INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANITABILITY,
	20 * INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILIT, 21 * FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.
	22 * *
	24 */
	25 /* add user code end Header */ 26
	27 /* Includes*/
	28 #include "at32f403a_407_wk_config.h"

Click on the "Generate code" in the menu bar or toolbar, and then the "Project Manager" window pops up, as shown in Figure 30.

Figure 30. Project manager

🔏 Project Mana	ger		$\times$
Project Setti	ngs		
Project Na	ame	AT32F403AVGT7_WorkBench	
Project Lo	cation	C:/Users/zhiqiangyou/Desktop Browse	
Toolchain	IDE	MDK_V5 V	
✓ Keep	User Code wher	n re-generating	
Linker Settin	qs		
Minimum	Heap Size (0x) Stack Size (0x)	0x200 0x400	
MCU and Fi	mware Package		
🔽 Сору	libraries into the	project folder	
Downl	oad the latest ve	rsion O Use folder package Use *.zip package	
Firmware	Relative File (*.zi	ip) [rs/zhiqiangyou/Downloads/AT32F403A_407_Firmware_Library_V2.1.6.zip] Browse	
		OK	

By default, the minimum heap size and stack size are 0x200 and 0x400, respectively. Users need to modify these values when middleware stack is used.

Tick "Copy libraries into the project folder", and libraries in the firmware package are copied automatically into the project folder when generating code.

Finally, click on "OK" to generate user code and project of the selected IDE automatically. The generated project file structure is shown in Figure 31.



#### Figure 31. Project file directory

名称	修改日期	类型	大小
libraries	2024/1/5 14:44	文件夹	
middlewares	2024/1/5 14:44	文件夹	
project	2024/1/5 14:44	文件夹	
AT32F403AVGT7_WorkBench.ATWP	2024/1/5 14:44	<b>ATWP</b> 文件	3 KB

### 1.1.5 How to quickly replace AT32F415 with AT32F423

- This migration guide from AT32F415 to AT32F423 is detailed in the document "MG0022\_Migrating\_from\_AT32F415\_to\_AT32F423\_V2.0.0", which is available from <u>ARTERY</u> <u>official website</u>→PRODUCTS→Value line→AT32F423 series page.
- If program failure occurs, please refer to the corresponding sections of this document, or you can contact your local or nearest ARTERY Tech team for assistance.

### **1.2** AT32F423 functionality enhancement

### 1.2.1 Instruction prefetch buffer

Instruction prefetch buffer is very useful for achieving quicker CPU execution speed. 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 will then determine whether or not to access Flash memory according to available space in the buffer. Once there is at least a free space in the instruction prefetch buffer, then 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₽	Abbr?	Reset value <i></i>	Type↩	Description ∂
Bit 2: 0∢	WTCYC↩	0x0 +²	ſ₩¢	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+<br="">1: One wait state when 32MHz<system clock≤64mhz+<br="">2: Two wait states when 64MHz<system clock≤96mhz+<br="">3: Three wait states when 96MHz<system clock≤128mhz<br="">4: Four wait states when 128MHz<system clock≤150mhz<="" td=""></system></system></system></system></system>

Figure 32.	Wait stat	e bit in F	LASH	PSR	reaister
	man ouu	• • • • • • •			egiete.

AT32 library has made relevant settings in the system\_clock\_config() function. For BSP of other AT32 MCU series, you can also find this setting at the same location.

#### Figure 33. system\_clock\_config function

```
void system_clock_config(void)
{
    /* reset crm */
    crm_reset();
    /* config flash psr register */
    flash_psr_set(FLASH_WAIT_CYCLE_4);
    /* ensure system clock to highest, set power 1do output voltage to 1.3v */
    pwc_ldo_output_voltage_set(PWC_LDO_OUTPUT_1V3);
    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

### 1.2.2.1 PLL settings

AT32F423 embeds a PLL with a maximum of 150 MHz clock output. The CRM\_PLLCFG register (PLL clock configuration register) can be used to configure different PLL clock frequencies, with the following formula:

```
PLL \text{ output clock} = \left(PLL \text{ reference input clock} \times \frac{PLL \text{ multiplication frequency factor PLL_NS}}{PLL \text{ pre - division factor PLL_MS} \times PLL \text{ post - division factor PLL_FR}}\right)/2
```

In AT32 BSP, example of PLL settings is based on HEXT=8 MHz, PLL=150 MHz.

```
Figure 34. AT32F423 150 MHz output clock configuration
```

/\* config pll clock resource PLL\_FR =4\*/ crm\_pll\_config(CRM\_PLL\_SOURCE\_HEXT, 150, 1, CRM\_PLL\_FR\_4);

Where, the first parameter "CRM\_PLL\_SOURCE\_HEXT" represents that the HEXT is used as an external clock source, PLL\_NS=150, PLL\_MS=1, and PLL\_FR value is CRM\_PLL\_FR\_4 (0x02, divided by 4).

For more information on clock configurations, please refer to the document "AN0158\_AT32F423\_CRM\_Start\_Guide". Path: <u>ARTERY official website</u>  $\rightarrow$  SUPPORT $\rightarrow$ AP Note $\rightarrow$ AN0158. With this file, you can learn more about how to configure and modify AT32F423 clock source code, and how to use ARTERY's New Clock Configuration tool to quickly generate the desired clock code and apply it to project.

The New Clock Configuration document is available from <u>ARTERY official website</u>  $\rightarrow$  PRODUCTS  $\rightarrow$  Value line  $\rightarrow$  AT32F423 series  $\rightarrow$  Tool.

### 1.2.2.2 PLL automatic frequency switch feature

When the PLL multiplication frequency of AT32F423 series is greater than 108 MHz, it is recommended to enable PLL auto step-by-step frequency switch feature.



Figure 35 gives an example of PLL auto frequency switching function in AT32F423 BSP.

 Figure 35. AT 32 FLL auto step-by-step frequency switch configuration example
/* 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();

Figure 35. AT32 PLL auto step-by-step frequency switch configuration example

Note: If this auto frequency switching function is enabled, it should be disabled right after the completion of clock switching operation. In other words, enabling and disabling must be operated in pair.

### 1.2.3 Encryption

Note: The BOOT1 bit of AT32F423 series is located in the user system data area (0x1FFF F800). When ISP tool is used, it is mandatory 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 out through conducting 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 cannot 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. (See Figure 36 below)

Unlock access protection: click Target - Access protection - disable

File J-Link settings AT-Link settings	Target Language Help	
Disconne ct AT-Link Plus FW: V2.2.2 AT-Link SN: F5A81400004(	Mass erase Erase main flash Erase boot memory Erase sectors	■ <u>"17[</u> 7Y ■ 雅 特 力
Extra configuration SPIM Config QSPI Config	User system data Access protection	Enable access protection
Memory read settings	sLib status	Enable high level access protection Disable
Address 0x 08000000 Read size	Boot memory AP mode DownLoad	its ~ Read
File info	Flash CRC	
No. File name 1 423.hex	Debug	range(0x) Add 0-08000EEF Delete

#### Figure 36. Use ISP tool to enable or disable access protection

Artery ISP Programmer (BOOT0=1)

Enable access protection: Keep clicking "Next" until you enter the final interface. Then check "Protection", choose "Enable" and "Access protection", click "Yes" (see Figure 37 below).

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

■ Artery ISP Multi-Port Programmer (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".

🔯 Artery ISP Programmer_V2.0.08 – 🗌 🗙
<sup></sup> <sup> </sup>
O Erase O All O Sectors
O Download to device O Disable sLib
sLib Status: DISABLE Start sector IJSTR start sector Fassword Ox End sector
No.     File Name     File Size     Address Range(Dz)     Add       Balaxee       Confirm       Brase option       Optimize (Bay       Write user       Write user       Address Optimize (Bay
Apply User     3     是(?)     至(N)       Enable Access protection after Download Access protection
O Firmware CRC Sector fill FF
O Flash CBC       1     Start sector Sector21—0x000A000       Back 2    Close

#### Figure 37. Use ISP to enable access protection

Artery ISP	Programmer_V2.0.08	-	×
;	<u>#Y75171 #</u>	惟特力	
O Eraze	All O Sectors	🔿 Edit User sys	tem data
O Download t	o device	🔿 Disable sLib	
sLib Stat	us: DISABLE Start se	otor	~
		art sector	~
Password	Ox End sect	or	×
No. F	le Name File S	ite Add ess hange(ox)	Add
Er-	The flash memory will be mass erased you sure to disable the access protecti		
Enable	Access protection after Download Acc	ess protection ~	
	Access protection after Download Acce device C:\test_binhex\403A.hex	ess protection 🗸 🗸	
	a device C:\test_binhex\403A.hex	ess protection 🗸 🗸	
<ul> <li>Upload fro</li> <li>Firmware C</li> </ul>	a device C:\test_binhex\403A.hex	ess protection 🗸	
<ul> <li>Upload fro</li> <li>Firmware C</li> <li>Flash CBC</li> </ul>	n device C:\test_binhex\403A hex 8C Sector fill FF	ess protection	
<ul> <li>Vpload fro</li> <li>Firmware C</li> <li>Flash CBC</li> </ul>	n device C:\test_binhex\403A hex 8C Sector fill FF	End sector Sector0-0x8000000	

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

### 1.2.3.2 Erase and program 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/program protection based on procedure below.

■ Artery ICP Programmer (BOOT0=0)

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

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

Artery ISP Programmer (BOOT0=1)

**Enable erase/program protection:** check "Protection" option, choose "Enable" and "Erase/write protection", and then click "Yes"

**Disable erase/program protection:** check "Protection" option, choose "Disable" and "Erase/ write protection", and then click "Yes"

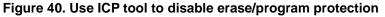
■ Artery ISP Multi-Port Programmer (BOOT0=1)

**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"

Image: system data       X         Access protection       FAP A5         PA       Disable         System setting byte         SSB       F         N/VDT_ATO_EN       NDEPSLP_RST         N/VDT_DEPSLP       N/VDT_STDBY         Erase and program protection bytes       Image: start addr.         Image: Start addr.       End addre         Start addr.       Size         Image: Start addr.       Size         Sectority       Ox80007EF         Sectority       Ox8000000         Sectority       Ox800022FF         Sectority       Ox800022FF         Sectority       Ox8000000         User data       Sectority         Image: Distable       Size         Image: Distable       Size         Image: Distable       Size         Sectority       Ox8000000         Sectority       Ox8000000         Sectority       Ox8000000         Sectority       Ox8000000         Sectority       Ox8000000         Size       Sectority         Size       Sectority         Sectority       Sectority         Sectority       Sectority	Access protection         FAP       A5         Disable         System setting byte         State addressing byte         State addressing byte         State addressing byte         Class and program protection bytes         Name         State addressing byte         Sector         Sector         Ox800000         Ox8000000         Sector         <	Access protection FAP AS Disable System setting byte System setting byte SSB FF INWDT_ATO_EN INDEPSLP_RST INSTDBY_RST INBOOT1 INWDT_DEPSLP INWDT_STDBY Erase and program protection bytes Erase and program protection bytes Sector 0 0x8000000 0x80007FF 0x800(2X) N Sector 0 0x80002000 0x80007FF 0x800(2X) N Sector 0 0x800000 0x80007FF 0x800(2X) N Sector 0 0x800000 0x80007FF 0x800(2X) N Sector 0 0x800000 0x80007FF 0x800(2X) N Sector 0 0x80000 0x80000 0x80007FF 0x800(2X) N Sector 0 0x80000 0x80000 0x800000 0x8000000 0x800000 0x800000 0x800000 0x800000 0x800000 0x800000 0x800000 0x8000000 0x8000000 0x8000000 0x800000 0x8000000 0x8000000 0x8000000 0x800000 0x8000000 0x8000000 0x8000000 0x8000000 0x80000000 0x80000000 0x8000000 0x8000000 0x800000000									_				
FAP       5       Disable         System setting byte         Sss       FF       InWDT_ATO_EN       InDEPSLP_RST       InSTDBY_RST       InBOOT1         Sss       FF       InWDT_DEPSLP       InWDT_STDBY       InBOOT1         Erase and program protection bytes       InBOOT1       Erase and program protection bytes       EPP0-3       F9       FF       F	FAP       AS       Disable         System setting byte         SSB       FF       MVDT_ATO_EN       MDEPSLP_RST       InSTDBY_RST       InBOOT1         SSB       FF       MVDT_DEPSLP       MVDT_STDBY         Erase and program protection bytes         Sector1       0x8000000       0x80007FF       0x800(2k)       N         Sector2       0x8001000       0x80007FF       0x800(2k)       N         Sector3       0x8001000       0x80007FF       0x800(2k)       Y         Sector4       0x800000       0x80007FF       0x800(2k)       Y         Sector5       0x800000       0x80007FF       0x800(2k)       Y         Sector6       0x8000000       0x80007FF       0x800(2k)       Y         Sector5       0x800000       0x80007FF       0x800(2k)       Y         Sector5       0x800000       0x80007FF       0x800(2k)       Y          Sector6       0x800000       0x80007FF       0x80(2k)       Y           User data       T       FF       F	FAP       A5       Disable         System setting byte         Ssse       FF       InVDT_ATO_EN       InDEPSLP_RST       InSTDBY_RST       InBOOT1         Ssse       FF       InVDT_DEPSLP       InVDT_STDBY       InBOOT1       InBOOT1       InBOOT1         Erase and program protection bytes       Instraduct       Erase and program protection bytes       EPPP-3       F9       FF       FF         Sector1       0x8000000       0x80007FF       0x800(2K)       N       Instraduct       EPP-3       F9       FF       FF         Sector1       0x8000000       0x80007FF       0x800(2K)       N       Instraduct       Instr	🚾 User system data												×
SSB       FF       Y       NWDT_ATO_EN       Y       DEPSLP_RST       Y       NBODT1         Y       NWDT_DEPSLP       Y       NWDT_STDBY       Y       DEPSLP_RST       Y       NBODT1         Frase and program protection bytes       Image: Start addr       End addre       Start addr       End addre       Start addr       Find addre       Start addr       Find addre       Start addre       addre </td <td>SSB     FF     INVDT_ATO_EN     MDEPSLP_RST     INSTDBY_RST     INBOOT1       INVDT_DEPSLP     INVDT_STDBY    Frase and program protection bytes  Frase and program protection bytes  Sector1 Ox8000000 Ox80007FF Ox800(2k) N Sector2 Ox800100 Ox800007FF Ox800(2k) V Sector3 Ox80000 Ox800000 Ox800007FF Ox800(2k) V Sector3 Ox80000 Ox800000 Ox800007FF Ox800(2k) V Sector3 Ox80000 Ox80000 Ox800007FF Ox800(2k) V Sector3 Ox80000 Ox80000 Ox800007FF Ox800(2k) V Sector3 Ox80000 Ox80000 Ox80000 Ox80000 Ox80000 V Sector3 Ox8000 Ox80000 Ox80000 Ox80000 Ox80000 Ox8000 Ox8000 Ox8000 Ox8000 Ox8000 Ox8000 Ox8000 Ox8000 Ox800 Ox80 Ox8</td> <td>SSB       FF       NWDT_ATO_EN       N DEPSLP_RST       N SDDBY_RST       M BOOT1         MWDT_DEPSLP       N WDT_STDBY         Erase and program protection bytes         Name       Stat addr.       End addre.       Size       EPP       EPP       EPP       EPP0-3       F9 FF FF FF         Sector1       0x8000000       0x80007FF       0x8002(X)       N       EPP0-3       F9 FF FF FF         Sector2       0x8001600       0x80017FF       0x8002(X)       N       EPP0-3       F9 FF FF FF         Sector3       0x8001600       0x80017FF       0x8002(X)       N       V       Sector3       0x8001800       0x80012(X)       N       EStat addresconsection       Sector4       Sector3       Sector4       Sector4       Sector4       Sector5       Sector4       Sector5       Sector4       Sector5       Sector5</td> <td></td> <td>able</td> <td></td> <td></td> <td>~</td> <td>]</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	SSB     FF     INVDT_ATO_EN     MDEPSLP_RST     INSTDBY_RST     INBOOT1       INVDT_DEPSLP     INVDT_STDBY    Frase and program protection bytes  Frase and program protection bytes  Sector1 Ox8000000 Ox80007FF Ox800(2k) N Sector2 Ox800100 Ox800007FF Ox800(2k) V Sector3 Ox80000 Ox800000 Ox800007FF Ox800(2k) V Sector3 Ox80000 Ox800000 Ox800007FF Ox800(2k) V Sector3 Ox80000 Ox80000 Ox800007FF Ox800(2k) V Sector3 Ox80000 Ox80000 Ox800007FF Ox800(2k) V Sector3 Ox80000 Ox80000 Ox80000 Ox80000 Ox80000 V Sector3 Ox8000 Ox80000 Ox80000 Ox80000 Ox80000 Ox8000 Ox8000 Ox8000 Ox8000 Ox8000 Ox8000 Ox8000 Ox8000 Ox800 Ox80 Ox8	SSB       FF       NWDT_ATO_EN       N DEPSLP_RST       N SDDBY_RST       M BOOT1         MWDT_DEPSLP       N WDT_STDBY         Erase and program protection bytes         Name       Stat addr.       End addre.       Size       EPP       EPP       EPP       EPP0-3       F9 FF FF FF         Sector1       0x8000000       0x80007FF       0x8002(X)       N       EPP0-3       F9 FF FF FF         Sector2       0x8001600       0x80017FF       0x8002(X)       N       EPP0-3       F9 FF FF FF         Sector3       0x8001600       0x80017FF       0x8002(X)       N       V       Sector3       0x8001800       0x80012(X)       N       EStat addresconsection       Sector4       Sector3       Sector4       Sector4       Sector4       Sector5       Sector4       Sector5       Sector4       Sector5		able			~	]							
			2		TO_EN	nD	EPSLP	_RST	⊠ nS	TDBY_F	RST	🗹 ni	nBOO	īΤ1	
Name         Stat addr         End addre         Size         EPP         EPP-0.3         F9         FF	Name         Start addr         End addre         Size         EPP         EPP0-3         F9         FF         FF	Name         Stat addr.         End addre.         Size         EPP         EPPo.3         F9         FF         FF <t< td=""><td></td><td>nWDT_D</td><td>EPSLF</td><td>P ⊠ nV</td><td>/DT_ST</td><td>DBY</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>		nWDT_D	EPSLF	P ⊠ nV	/DT_ST	DBY							
□ Sector0       0x8000000       0x80007FF       0x80007FF       0x80007FF         □ Sector1       0x8000100       0x80007FF       0x80007FF       0x80007FF         □ Sector2       0x800100       0x80017FF       0x80027F       0x8007FF         □ Sector2       0x800100       0x80017FF       0x80027F       0x80027F         □ Sector5       0x8002000       0x80027FF       0x80027F       0x80027F         □ Sector6       0x8002000       0x80027F       0x80027F       0x80027F         □ Data 515 (bx)	Sector0       0x8000000       0x80007FF       0x80007FF       0x80007FF         Sector1       0x8000000       0x800007FF       0x80007FF       0x80007FF         Sector2       0x8001000       0x80007FF       0x8002(X)       Y         Sector3       0x8000000       0x80007FF       0x800(2K)       Y         Sector3       0x800000       0x80007FF       0x800(2K)       Y         Sector5       0x8002800       0x80007FF       0x800(2K)       Y         Sector6       0x8002800       0x80027FF       0x800(2K)       Y         Sector6       0x8002800       0x80027FF       0x800(2K)       Y         User data               User data                User data                  User data                                 <	Sector0       0x8000000       0x80007FF       0x80007FF       0x80007FF       0x80007FF         Sector1       0x800017F0       0x80007FF       0x80007FF       0x80007FF       0x80007FF         Sector3       0x800100       0x80007FF       0x80007FF       0x80007FF       0x80007FF         Sector3       0x800100       0x80007FF       0x80007FF       0x80007FF       0x80007FF       0x80007FF         Sector4       0x80002000       0x800027FF       0x80007FF       0x80007FF       0x80007FF       0x80007FF       0x80007FF         Sector4       0x80002000       0x800027FF       0x800027F       0x800072FF       0x800072F       0x800072F       0x800072FF       0x800072FF       0x800072FF       0x800072FF       0x800078       0x800078	Erase and program	protection	n bytes										
☑ Sector2       0x8001000       0x80017FF       0x800(2K)       N         ☑ Sector3       0x8001800       0x8001FFF       0x800(2K)       N         ☑ Sector3       0x8001800       0x8001FFF       0x800(2K)       N         ☑ Sector5       0x8002800       0x8002FF       0x800(2K)       N         ☑ Sector6       0x8003000       0x8002FF       0x800(2K)       N       □         ☑ Sector6       0x800300       0x80037FF       0x800(2K)       N       □       □         User data       0       1       2       3       4       5       6       7       ∩         Date       0       1       2       3       4       5       6       7       ∩         Data       0       1       2       3       4       5       6       7       ∩         Data       0       1       2       3       4       5       6       7       ∩         Data       0       1       2       3       4       5       6       7       ∩         Data       0       1       2       3       4       5       6       7       ∩	✓ Sector2       0x8001000       0x80017FF       0x80012K)       Y         ✓ Sector3       0x8001800       0x8001FFF       0x80012K)       Y         ✓ Sector4       0x8002000       0x8002FFF       0x8002K)       Y         ✓ Sector5       0x8003000       0x8002FFF       0x8002K)       Y         ✓ Sector6       0x8003000       0x8003FFF       0x8002K)       Y          ✓ Sector6       0x8003000       0x8003FFF       0x8002K)       Y           ✓ Sector6       0x8003000       0x8003FFF       0x8002K)       Y            ✓ Sector6       0x8003000       0x8003FFF       0x8003FFF       0x8002K)       N            ✓ User data       0       1       2       3       4       5       6       7          Date       0       1       2       3       4       5       6       7          Date       0       1       2       3       4       5       6       7         Clear         Data       6       1       7       FF       FF       FF       FF       FF<	☑ Sector/2       0x8001000       0x80017FF       0x800(2K)       Y         ☑ Sector/2       0x8001800       0x8001FFF       0x800(2K)       Y         ☑ Sector/2       0x80000       0x8002FFF       0x800(2K)       Y         ☑ Sector/2       0x80000       0x8002FFF       0x800(2K)       Y         ☑ Sector/2       0x80000       0x8002FFF       0x800(2K)       Y         ☑ Sector/2       0x80000       0x80007FF       0x800(2K)       Y	Sector0	0x80000	0 0	x80007F	F 0x	300(2K)	N		EPP0-	3 F	F9 F	FFFF	]
□         Select all           User data         0         1         2         3         4         5         6         7         6           Date         0         1         2         3         4         5         6         7         6           Date         0         1         2         3         4         5         6         7         6           Date         0         1         2         3         4         5         6         7         6           Date         0         7         FF         FF<	□         □         ○         ○         Select all           User data         0         1         2         3         4         5         6         7           Date         0         1         2         3         4         5         6         7           Date         0         1         2         3         4         5         6         7         Clear           Date         0         1         2         3         4         5         6         7         Clear           Date         0         7         FF	□         Sectors         0x8003000         0x80037F         0x800(2k)         N         □         Select all           User data           □         0         1         2         3         4         5         6         7         6           □         0 table         0         1         2         3         4         5         6         7         6           □         0 table         0         1         2         3         4         5         6         7         6           □         0 table         0         1         2         3         4         5         6         7         6         Clear           □         0 table         0         1         2         7         F         FF	Sector2	0x800100 0x800180	00 0	x80017F x8001FF	F Ox	800(2K) 800(2K)	Y Y						
User data           Date         0         1         2         3         4         5         6         7           Data         0         1         2         3         4         5         6         7           Data         0         1         2         3         4         5         6         7           Data         0         1         2         3         4         5         6         7           Data         0         1         2         3         4         5         6         7         7           Data         0         7         FF	User data Date 0 1 2 3 4 5 6 7 Data 67(0x) FF FF FF FF FF FF FF FF Data 615(0x) FF FF FF FF FF FF FF FF Data 1623(0x) FF FF FF FF FF FF FF FF FF Data 16-23(0x) FF FF FF FF FF FF FF FF FF Data 16-23(0x) FF	User data Data 0 1 2 3 4 5 6 7 Data 0 FF FF FF FF FF FF FF FF Data 815 (Dx) FF FF FF FF FF FF FF FF Data 323 (Dx) FF FF FF FF FF FF FF FF FF FF Data 07 (Dx) FF							Y						
Date         0         1         2         3         4         5         6         7         ^           Data 07 (bx)         FF         FF <td< td=""><td>Date         0         1         2         3         4         5         6         7         Clear           Data 07 (0x)         FF         FF</td><td>Date         0         1         2         3         4         5         6         7         ^           Data 07 (bx)         FF         <td< td=""><td></td><td>0x800300</td><td>0 0</td><td>x80037F</td><td>F 0x</td><td>B00(2K)</td><td>N</td><td>~</td><td>🗌 Se</td><td>lect al</td><td>all</td><td></td><td></td></td<></td></td<>	Date         0         1         2         3         4         5         6         7         Clear           Data 07 (0x)         FF	Date         0         1         2         3         4         5         6         7         ^           Data 07 (bx)         FF         FF <td< td=""><td></td><td>0x800300</td><td>0 0</td><td>x80037F</td><td>F 0x</td><td>B00(2K)</td><td>N</td><td>~</td><td>🗌 Se</td><td>lect al</td><td>all</td><td></td><td></td></td<>		0x800300	0 0	x80037F	F 0x	B00(2K)	N	~	🗌 Se	lect al	all		
Data 07 (bx)         FF	Data 67 (ox)         FF         FF	Data 07 (bx)         FF	User data												
Data 815 (Dx)         FF         FF         FF         FF         FF         FF         FF           Data 1623 (Dx)         FF         <	Data 6—15 (0x)         FF         FF	Data 815 (Dx)         FF         FF         FF         FF         FF         FF         FF           Data 1623 (Dx)         FF         FF         FF         FF         FF         FF         FF         FF         Load file           Data 1623 (Dx)         FF         FF         FF         FF         FF         FF         FF         Load file	Date	0	1	2	3	4	5	6	7	^		Clear	
Data 16—23 (0x) FF Load file	Data 16—23 (0x) FF FF FF FF FF FF FF FF FF Load file	Data 16—23 (0x)         FF         FF         FF         FF         FF         FF         FF         FF         Load file           Data 16—23 (0x)         FF	Data 07 (0x)	FF	FF	FF	FF	FF	FF	FF	FF				
				FF	FF	FF	FF	FF	FF	FF	FF				
Data 2431 (0x) FF Save to file	Data 2431 (0x)         FF         FF         FF         FF         FF         FF         FF         FF         V         Save to file	Data 2431 (0x)         FF         FF         FF         FF         FF         FF         FF         FF         FF         V         Save to file			_			_	-		_		L	.oad file	
			Data 2431 (0x)	FF	FF	FF	FF	FF	FF	FF	FF	~	S	ave to file	
					_										
			Load	rom devic	e	Apply to	device		Load	from file	e	S	Save t	o file	
Load from device Apply to device Load from file Save to file	Load from device Apply to device Load from file Save to file	Load from device Apply to device Load from file Save to file													
Load from device Apply to device Load from file Save to file	Load from device Apply to device Load from file Save to file	Load from device Apply to device Load from file Save to file													

#### Figure 39. Use ICP tool to enable erase/program protection



😢 User system d	lata									>
System setting t	Disable				_RST	<mark>⊘</mark> nS	TDBY_F	IST	⊠ ni	BOOT1
Erase and progr	ram protectio	n byte	s	_		_				
Name Sector0 Sector1 Sector2 Sector3 Sector5 Sector5	Start add 0x80000 0x80008 0x80010 0x80018 0x80020 0x80028 0x80020	000 000 000 000 000	End addr 0x80007F 0x8000FF 0x80017F 0x8001FF 0x80027F 0x8002FF 0x8002FF	FF 0x8 FF 0x8 FF 0x8 FF 0x8 FF 0x8 FF 0x8 FF 0x8	e 100(2K) 100(2K) 100(2K) 100(2K) 100(2K) 100(2K) 100(2K)	Ν		EPP0-3		FF FF FF FF
User data Date	0	1	2	3	4	5	6	7	^	01
Data 97 (0x) Data 815 (0x) Data 1623 (0x	FF FF ) FF	FF FF FF	FF FF FF	FF FF FF FF	4 FF FF FF	5 FF FF FF	FF FF FF FF	FF FF FF FF		Clear Load file
Data 2431 (0x	) FF	FF	FF	FF	FF	FF	FF	FF	v	Save to file
Lo	ad from devis	ce	Apply to	o device		Load	from file		S	ave to file

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

### **1.2.4** Set system memory as main memory extension

System memory is used as a boot mode by default to store microcontroller manufacturer' Startup Code. On top of this, in AT32F423 series, a new feature is added to the system memory by using it to store user-defined codes as an extended memory area (AP mode).

# 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, we can use Artery ICP Programmer to enable system memory as an extended memory according to the following procedures.

- Connect AT-Link/J-Link to AT-START-F423 evaluation board and supply power to it;
- Open Artery ICP programmer and choose AT-Link/J-Link connection;
- Go to menu bar: Target Boot memory AP mode OK.

#### Figure 41. Use ICP tool to set boot memory AP mode

To prevent unexpected wrong operations, you need manually enter the passkey "0xA35F6D24", then you can check whether the operation is successful or not in "File info" column.

	rogrammer_V3.0.07 settings AT-Link settings Target Language Help	-
Disconne	Part Number: AT32F423VCT7 Flash Size: 256KB	Y5=151;
ct	AT-Link Plus FW: V2.2.2 AIN: D1FB5F42A31D771D AT-Link SN: F5A81400004001210C95D107 (WinUSB)	
AT-Link ~	AI-LINK 3N. 13A014000040012100332101 (Wind 3D)	雅特力
Extra configur	ation	_
SPIM Config	🕼 AP mode enable key — 🗆 🗙	
Memory read Address 0x [	Eachta (au) (au)	s v Read
File info	OK Cancel	
No. File n		ange(0x) Add
1 423.h	Opration Progress	-08000EEF Delete
	Enabling AP mode	

#### Figure 42. Boot memory AP mode operating progress

During mass-production stage, we can also use Artery ICP Programmer to enable system memory as memory extension area according to the following procedures:

Connect AT-Link to AT-START-F423 evaluation board and supply power to it.

Note: The on-board AT-Link EZ does not support offline programming. As a result, you can only use non-EZ AT-Link.

- 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 an offline project:
  - 1. Click "Create";
  - 2. Enter a project name;
  - 3. Select a particular MCU series and MCU part number;
  - 4. Add .hex files;
  - 5. Select SWD as download interface;
  - 6. Tick "Boot memory AP mode" option and enter the passkey;
  - 7. Save project file or save project to AT-Link.

In addition to above settings, you can also make other configurations according to the actual needs.

•
C AT-Link Setting - 🗆 🗙
AT-Link settings AT-Link offline config settings AT-Link offline download status
Offline project <u>Vertex</u> Delete 1 Creat
Project name test Device AT32F423 V AT32F423VCT7 V
No. File name File size Address range(0x) Storage locat Add
1 run_in_boot_memory.hex 4044 08000000-08000FCB 4 Delete
2 run_in_boot_memory.hex 68 1FFFA400-1FFFA443
< >>
Erase option Erase the sectors of file size V
Download times
Reset and run     Download interface     SWD
Write user system data
Enable FAP after download 6 🗹 Boot memory AP mode
Access protection 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
Increase step: 0x 0000001
Load parameters Save parameters
7 7
One suisting Constant fin
Open project file Save project file Save project to AT-Link Close

Figure 43. Use ICP tool to set offline boot memory AP mode

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 actual 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 AT-Link. In this case, you need to enter AT-Link serial number that will be bonded to.

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 44. Use ICP tool to set offline project programming
--

🛯 AT-Link project file	settings	—		×
This project is on	y used at the specified AT-	Link.		
AT-Link SN:	F5A81400004001210			]
This project is onl	y used once.			
AT-Link AIN:	D1FB5F42A31D771D			
		ОК	Cancel	1

Г



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

CP AT-Link Setting	1 AT-Link of	fline download status	-		×
Select offline download item:	g settings AI-Link o		OWD		_
	Save and activate	Download interface: ISP uart baud rate:	SWD 115200		$\sim$
		ISP boot mode:	Manual		$\sim$
Activated project: test					
Total downloads: Unlimited Downloaded times: 2	Succe	ssful downloads: 2			
File download successfully!					
		3 Start	download	i	
		Start butto	n free dov	vnload	
		Cancel but	on free do		

#### Figure 45. 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 is stored at ARTERY official website→SUPPORT→AP Note→AN0066.
- For DEMO on running user program in the system memory, please refer to BSP, which is stored at <u>ARTERY official website</u>→PRODUCTS→Value line→AT32F423 series→BSP (unzip and get the AT32F423\_Firmware\_Library\_V2.x.x\utilities\at32F423\_boot\_memory\_ap\_demo).

### **1.2.5** How to distinguish AT32 MCU from other MCUs

■ Reading Cortex-M series CPU ID number, you can determine whether it is based on M0, M0+, M1, M3 or M4 core.

Figure 46. Read Cortex ID

cortex	<_id = *(uint32_t *)0xE000ED00;// read Cortex part number
if((cor	tex_id == 0x410FC240)    (cortex_id == 0x410FC241))
{	
	printf("This chip is Cortex-M4F.\r\n");
}	
else	
{	
	printf("This chip is Other Device.\r\n");
}	



#### Read UID and PID

```
/* get AT32 MCU's UID/PID base address */
  #define DEVICE_ID_ADDR1 0x1FFFF7F3
                                                 //define Artery MCU device ID and UID base address
                                                 //define MCU device ID and PID base address
 #define DEVICE_ID_ADDR2 0xE0042000
 /* store ID */
 uint8_t ID[5] = \{0\};
 /* AT32F423 MCU type table */
 const uint64_t AT32_MCU_ID_TABLE[] =
  {
      0x0000012700A3240, //AT32F423VCT7
                                                            LQFP100
                                                  256KB
      0x0000012700A3240, //AT32F423VBT7
                                                            LQFP100
                                                  128KB
 };
  /* 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 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: AT32F4xx MCU contains several ID codes. By organizing the obtained ID information into a 64-bit data, it is possible for users 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* (path: <u>ARTERY official website</u> $\rightarrow$ SUPPORT $\rightarrow$ AP Note $\rightarrow$ AN0016).

#### Figure 47. Read UID and PID



## 2 FAQ for download and compiling

### 2.1 **Program enters Hard Fault Handler**

- Access data outside its boundary limit. Locate where the program exceeds the boundary, and move it to normal data area.
- The program uses SRAM exceeding its maximum capacity threshold.
- System clock is set out of spec.

### 2.2 J-Link unable to recognize IC in Keil project

For a possible solution, please refer to the following documents:

- "FAQ0008\_J-Link\_cannot\_ find\_IC", which is stored at <u>ARTERY official website</u> →SUPPORT→FAQ→FAQ0008.
- "FAQ0132\_How\_to\_add\_Artery\_MCU\_into\_JLINK", which is stored at <u>ARTERY official website</u> →SUPPORT→FAQ→FAQ0132.

### 2.3 Possible problems occurred during download

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

When Keil debugging or downloading, a warning message below pops up:

37 ⊟ { 38 Timing
38 Timing Wision
40 while(
41 } 42 - Error: Flash Download failed - "Cortex-M4"
43 🖓 / * *
44 * 0bri 45 * 0par
46 * @ret 确定
4/ _ */
48 void Tim

Figure 48. Flash Download failed-"Cortex- M4" during download

There are several possible factors behind this pop-up message:

- Access protection is enabled. If so, unlock access protection before download.
- You may not load Flash algorithm file or choose an incorrect one. If so, add a correct Flash algorithm to Flash Download location.
- Wrong BOOT0 setting. Note that the BOOT0 must be set to 0 to boot MCU from main Flash memory.
- Older version of J-Link driver. Note that 6.20C or above is recommended for J-Link.
- JTAG/SWD pin is disabled. Please 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 it is not connected.



### 2.3.3 RDDI-DAP Error

- Compiler offers a higher level of optimization. For instance, the optimization level for Keil AC6 compiler is the default "-Oz", so it should be changed to "-O0/-O1".
- JTAG/SWD pin is disabled. Please refer to Section 2.3.5 for how to resume download.

### 2.3.4 ISP serial interface gets stuck during download

When using ISP interface for download, it is stuck at a certain location so that it cannot be released.

Proceed with this issue as follows:

- Check if power supply you are using is stable or not.
- Turn to 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:

- Disable JTAG/SWD PIN disable, so that program download failed and JTAG/SWD device cannot be located.
- Enter Standby mode, so that program download failed and JTAG/SWD device cannot be located.

The principle of solution is to halt the chip when the program is not running yet.

Recommended solutions:

- 1. Set BOOT mode: Configure to boot from boot memory or SRAM, then reset the MCU by setting the reset pin, and then perform erase operation to resume download.
- Use ICP tool to add AT-Link: Connect AT-Link RST pin to the MCU reset pin, then choose AT-Link connection on ICP Programmer interface, and then erase the MCU to resume download.
- 3. Use Keil to add AT-Link: Connect AT-Link RST pin to the MCU reset pin, then configure debug settings in Keil (as marked with red rectangle in Figure 49), and then perform erase operation to resume download.

CMSIS-DAP Cortex-M Target I Debug Trace Flash Downl	
CMSIS-DAP - JTAG/SW Adapt AT-Link-Plus(WinUSB) CMSIS Serial No: 4AB41500004026 Firmware Version: 2.2.11	
SWJ Port: SW -	Add Delete Update AP: 0x00
✓ <u>R</u> eset after Connect	Reset:       HW RESET       Cache Options         Image: Cache Code       Image: Cache Code       Image: Cache Code         Image: Cache Memory       Image: Cache Memory       Image: Download to Flash         Image: Stop after Reset       Image: Cache Memory       Image: Download to Flash
	OK Cancel Help

Figure 49. Configure debug settings

4. Use IAR to add AT-Link: Connect AT-Link RST pin to the MCU reset pin, then configure "CMSIS DAP" settings in IAR (as marked with red rectangle in Figure 50), and then perform erase operation to resume download.

Options for node "temple Category:	Factory Settings
General Options Static Analysis Runtime Checking C/C+ C-Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator CADI	Setup     Interface     Breakpoints       Reset     Connect during reset (default)     ~       Duration:     300 ms     Delay     200 ms
CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	Emulator Always prompt for probe selection Serial no: Log communication \$PROJ_DIR\$\cspycomm.log
	OK Cancel



# 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, AT32F423 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 Application 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.
- The security library 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.
- The 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 access (including ISP/ICP/debug mode or boot from internal RAM) via D-Code, for 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 also unlock this protection through wring a correct password in the SLIB\_PWD\_CLR register. Once sLib is unlocked, MCU will erase the whole main memory, including sLib. This kind of design is to protect 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  $AN0164\_AT32F423\_Security\_Library\_Application\_Note$ , which is stored at <u>ARTERY official website</u> $\rightarrow$ SUPPORT $\rightarrow$ AP Note $\rightarrow$ AN0164.



# 4 Revision history

Date	Version	Revision note
2023.03.20	2.0.0	Initial release.
2024.01.05	2.0.1	Updated Section 2.3.5 "How to resume program download" and added section
		1.1.4 "AT32 Work Bench".

#### Table 1. Document revision history

#### 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) Aerospace applications or environment; (D) Weapons, and/or (E) Other applications that may cause injuries, deaths or property damages. 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.

© 2024 ARTERY Technology - All Rights Reserved