AN0021
Application Note

Getting Started with AT32F435/F437

Introduction

The purpose of this application note is to provide a user guide on how to expedite AT32F435/F437 series-based application development. Compared to the AT32F435 series, the AT32F437 sereis offers an additional feature of EMAC.

Note: The codes in this application note are written based on the V2.x.x version of BSP developed by ARTERY. For other versions of BSP, please pay attention to the differences between versions when in use.

Applicable products:

Part number	AT32F435 series AT32F437 series

Contents

List of tables

List of figures

1 Environment requirements

Download link:

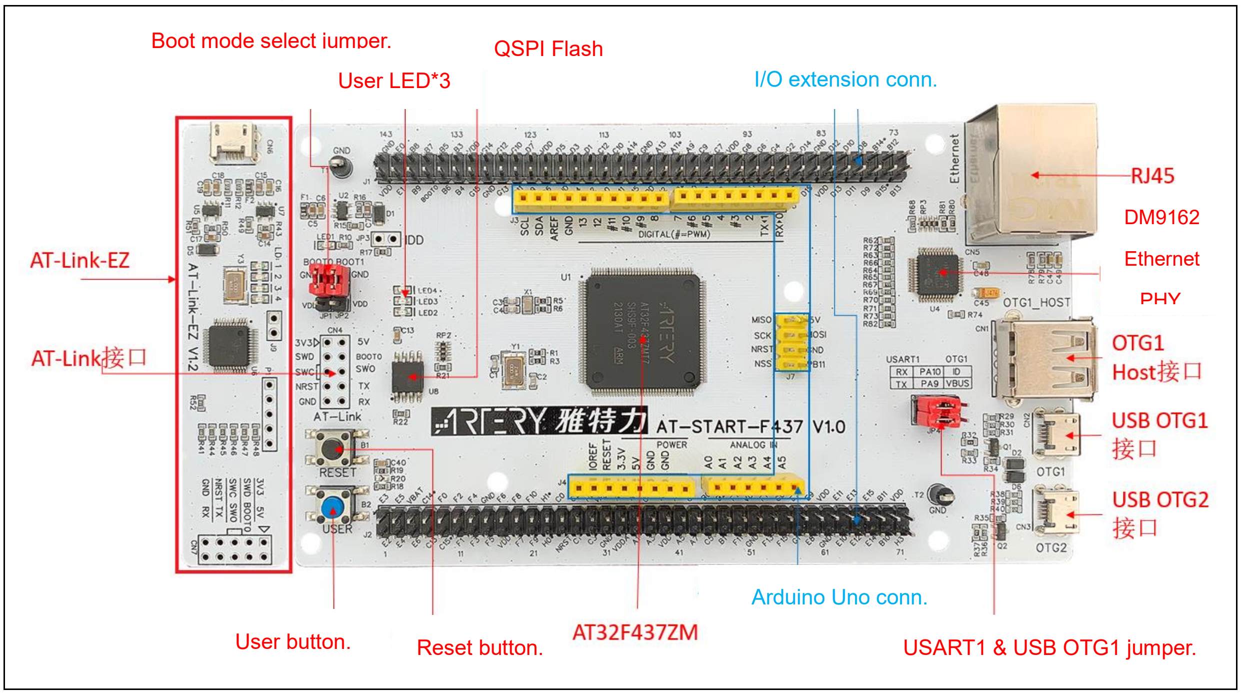
- ARTERY's official website: https://www.arterychip.com/

1.1 Set up AT32 MCU development environment

1.1.1 Debugging tool and evaluation board

The AT32F435/F435 series evaluation board is equipped with the AT-Link-EZ tool for debugging purpose, as shown in the left side of Figure 1 (marked in red box) below. This AT-Link-EZ can be disassembled from the board and used with other circuit boards. It supports IDE online debugging, online programming and USB-to-serial port.

Figure 1. AT-START-F437 evaluation board with AT-Link-EZ



Note: For details on the AT-START evaluation board, please refer to UM_AT_START_F435/F437_Vx.x, which can be found at ARTERY's official website → PRODUCTS → Mainstream → AT32F435/F437 → Evaluation Board → AT_START_F435/F437.

1.1.2 Programming tools and software

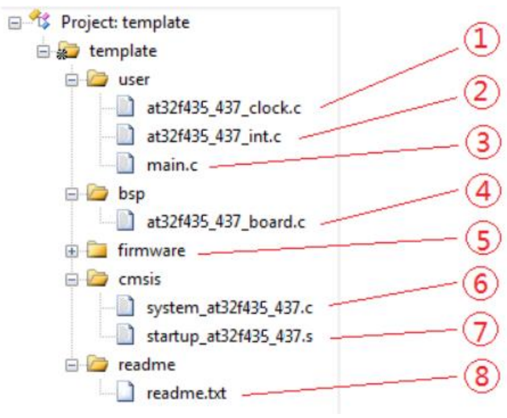
- AT programming tools and software: AT-Link, ICP/ISP.
- For ICP operation instruction, please refer to UM_ICP_Programmer
- For ISP operation instruction, please refer to UM_ISP_Programmer
- For AT-Link operation instruction, please refer to UM0004_AT-Link_User_Manual

1.1.3 AT32 MCU-based development environment

1.1.3.1 Template project

IDE template projects can be found in BSP.
BPS offers such template projects as Keil_v5, Keil_v4, IAR_6.10, IAR_7.4, IAR_8.2, eclipse_gcc, at32_ide. They are located at AT32F435_437_Firmware_Library_V2.x.xproject\at_start_f4xx\templates. Here is an example of Keil_v5 template project.

Figure 2. Keil_v5 templates



A template project contains the following content:

① at32f435_437_clock.c: clock configuration file, with default clock frequency and clock path being programmed
② at32f435_437_int.c: interrupt file that contains interrupt function-related code
③ main.c: refers to the main code file
④ at32f435_437_board.c: board-level configuration file that contains general hardware configuration such as AT-START on-board buttons and LEDs
⑤ at32f435_437_xx.c under firmware: peripheral driver files
⑥ system_at32f435_437.c: system initialization files
⑦ startup_at32f435_437.s: startup file
⑧ readme.txt: statement file that is used to introduce general operations and settings and application notes related to the template project.

In addition to templates, BSP contains a variety of code examples (Keil_v5 project files) for uses' reference. They are located at AT32F435_437_Firmware_Library_V2.x.x\project\at_start_f4xx\examples.

Note: For more details about BSP, please refer to "Section 4 BSP application" of AT32F435_437 Firmware BSP&Pack User Guide in BSP.

1.1.3.2 Installing PACK

Install Pack and add AT32 MCU part number to Keil/IAR. You can download Pack from the following link:

Keil4 Pack

Keil5 Pack

IAR Pack

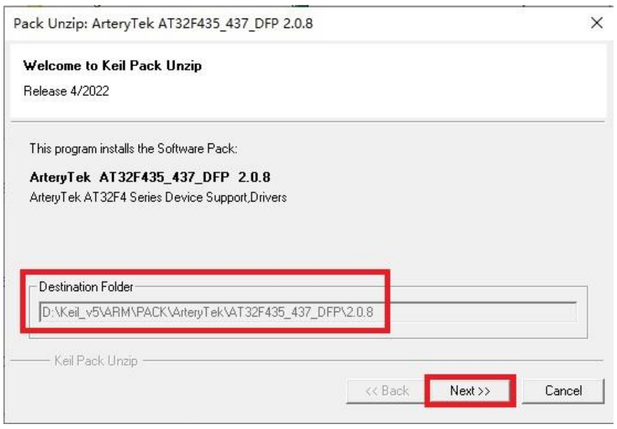
Segger Linux

Segger Win

For Keil compiling system, keil 4.74 /5.23 or above is recommended. For Keil_v5, users need to unzip the Keil5_AT32MCU_AddOn and then install the corresponding ArteryTek.AT32F435_437_DFP.

For Keil_v4, users can directly install Keil4_AT32MCU_AddOn. By default, the Keil installation path can be recognized automatically. In case of path recognition failure, users need to manually select the path.

Figure 3. ArteryTek.AT32F435_437_DFP



Pack Unzip: ArteryTek AT32F435_437_DFP 2.0.8

Welcome to Keil Pack Unzip

Release 4/2022

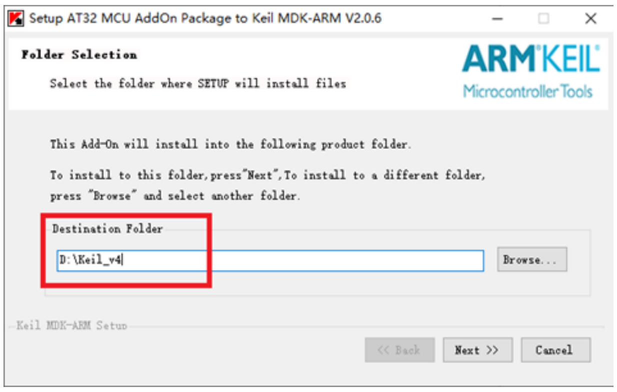
This program installs the Software Pack:

ArteryTek AT32F435_437_DFP 2.0.8

ArteryTek AT32F4 Series Device Support,Drivers

Destination Folder: D:\Keil_v5\ARM\PACK\ArteryTek\AT32F435_437_DFP\2.0.8

Figure 4. Set up Keil4_AT32MCU_AddOn



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

Folder Selection

Select the folder where SETUP will install files

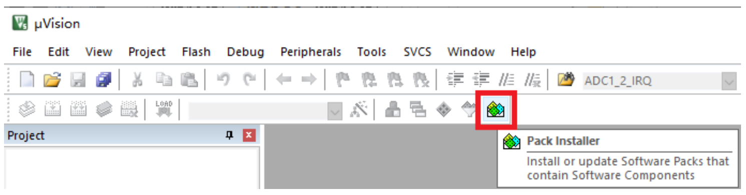
This Add-On will install into the following product folder.

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

Destination Folder: D:\Keil_v4

Another way to install PACK is as follows: Open keil and click on “Pack Installer” icon, click on “file” and select “import” to import the corresponding pack downloaded.

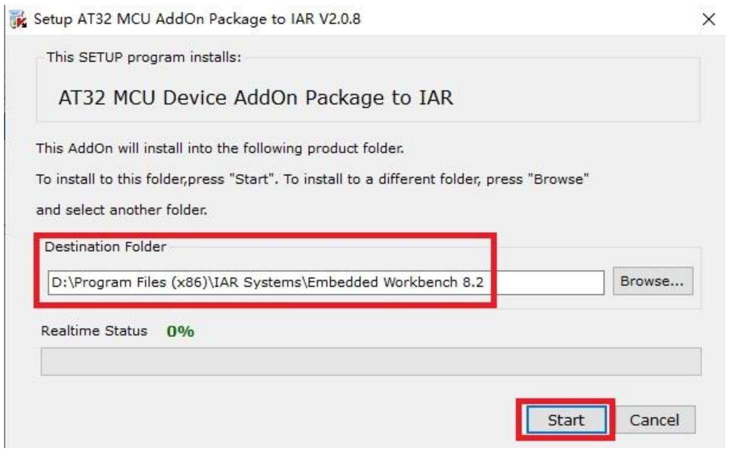
Figure 5. Pack Installer icon in Keil


µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
ADC1_2_IRQ
Project

Pack Installer
Install or update Software Packs that contain Software Components

For IAR compiling system, IAR7.0 / IAR6.1 or above is recommended. The IAR_AT32MCU_AddOn needs to be installed. By default, the IAR installation path can be recognized automatically during installation. In case of path recognition failure, users need to manually select the path.

Figure 6. Set up IAR_AT32MCU_AddOn



Setup AT32 MCU AddOn Package to IAR V2.0.8

This SETUP program installs:

AT32 MCU Device AddOn Package to IAR

This AddOn will install into the following product folder.

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

and select another folder.

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

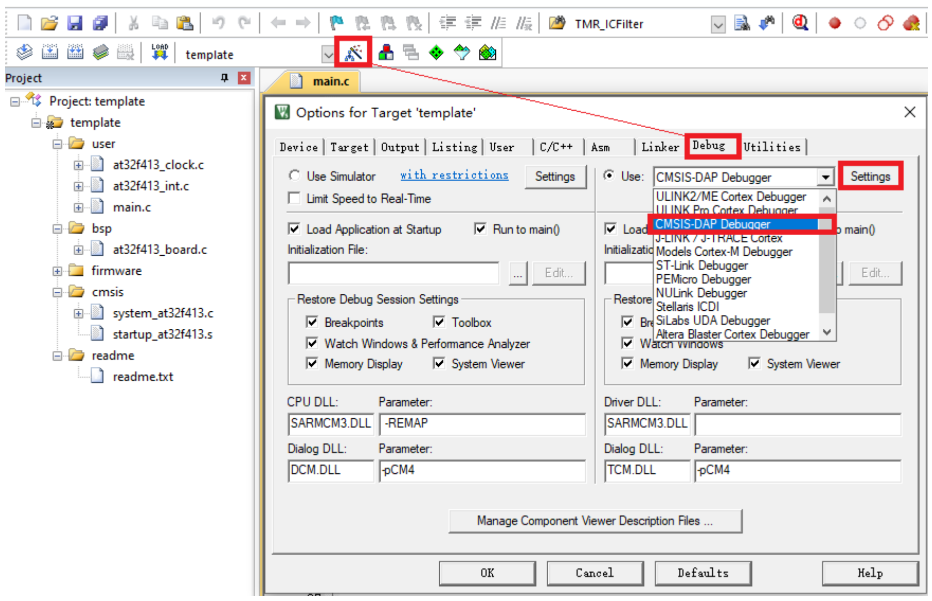
Realtime Status 0%

Note: For more details about PACK settings, please refer to "Section 2 Pack setup" of the AT32F435_437 Firmware BSP&Pack User Guide in BSP.

1.1.3.3 Debug and download with AT-Link

To use AT-Link in IAR, select CMSIS-DAP in Debugger option.

Figure 7. Keil Debug option



Options for Target 'template'

Use Simulator with restrictions

Limit Speed to Real-Time

Load Application at Startup; Run to main()

Initialization File:

Restore Debug Session Settings: Breakpoints; Toolbox; Watch Windows & Performance Analyzer; Memory Display; System Viewer

CPU DLL: SARMCM3.DLL Parameter: -REMAP

Dialog DLL: DCM.DLL Parameter: -pCM4

Use: CMSIS-DAP Debugger

ULINK2/ME Cortex Debugger
ULINK Pro Cortex Debugger
CMSIS-DAP Debugger
J-LINK / J-TRACE Cortex
Models Cortex-M Debugger
ST-Link Debugger
PEMicro Debugger
NULink Debugger
Stellaris ICDI
SiLabs UDA Debugger
Altera Blaster Cortex Debugger

Load; Run to main()

Restore: Breakpoints; Watch Windows; Memory Display; System Viewer

Driver DLL: SARMCM3.DLL Parameter:

Dialog DLL: TCM.DLL Parameter: -pCM4

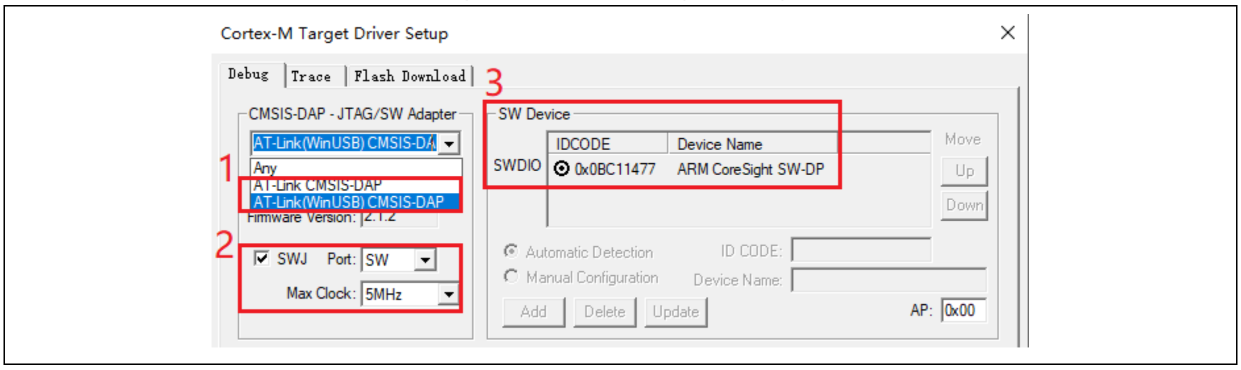
Go to "Debug" and click on "Settings" to enter "Cortex-M Target Driver Setup" interface.

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

Note: For details about WinUSB, refer to FAQ0136_How_to_use_AT_LINK_WinUSB_for_faster_download

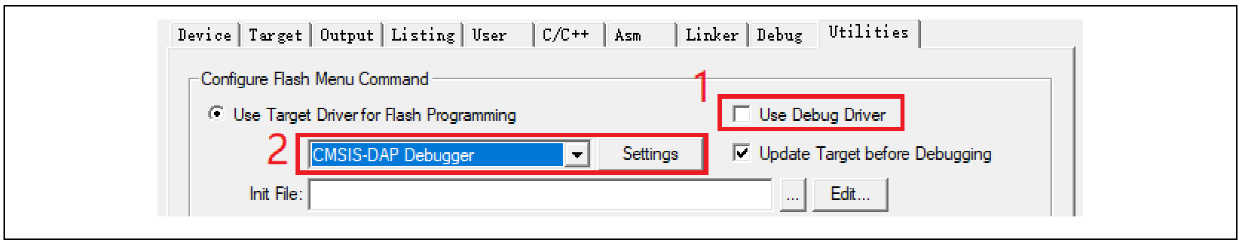
2. Select “SW” in Port, and tick “SWJ”
3. Confirm that the ARM SW-DP debug module is recognized.

Figure 8. Keil Debug Settings



Click on “Utilities”, untick “Use Debug Driver” option (Step 1 below), choose “CMSIS-DAP Debugger” in Settings (Step 2 below), and then re-tick Step 1 (This option must be first unticked and then re-ticked)

Figure 9. Keil Utilities



If users want to use AT-Link in IAR, click on “Project” and “Options”, then select “CMSIS-DAP”, and select “SWD”.

Figure 10. IAR Debug option

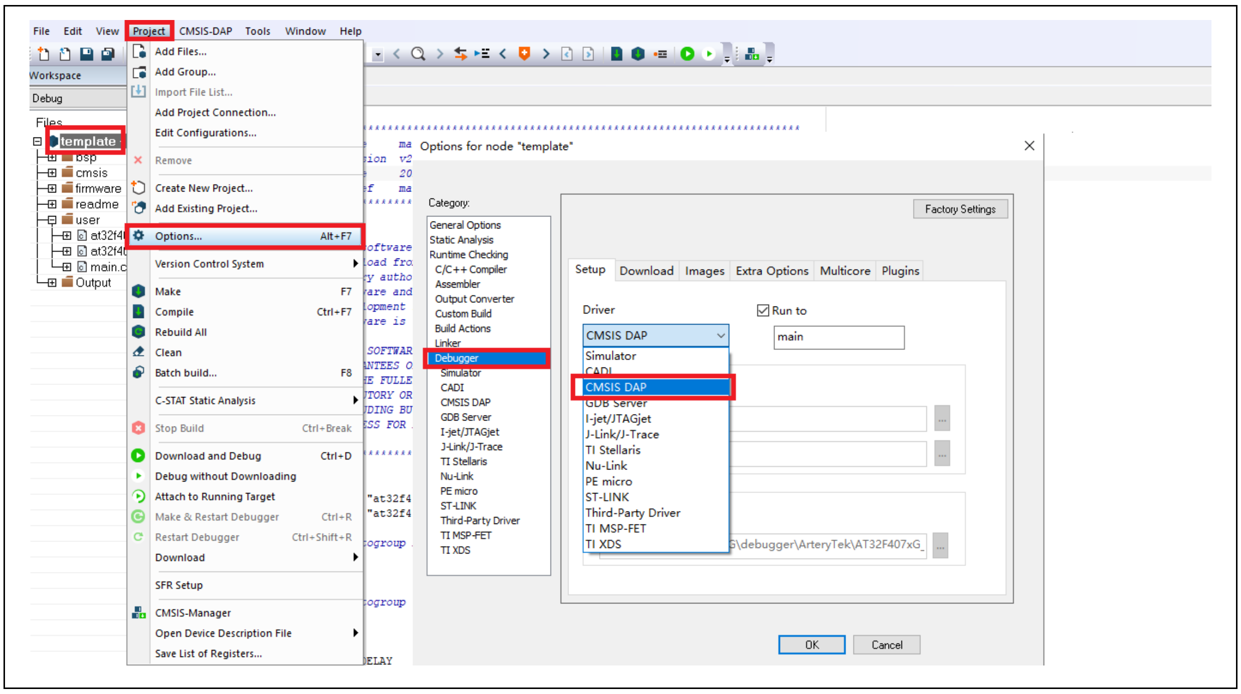
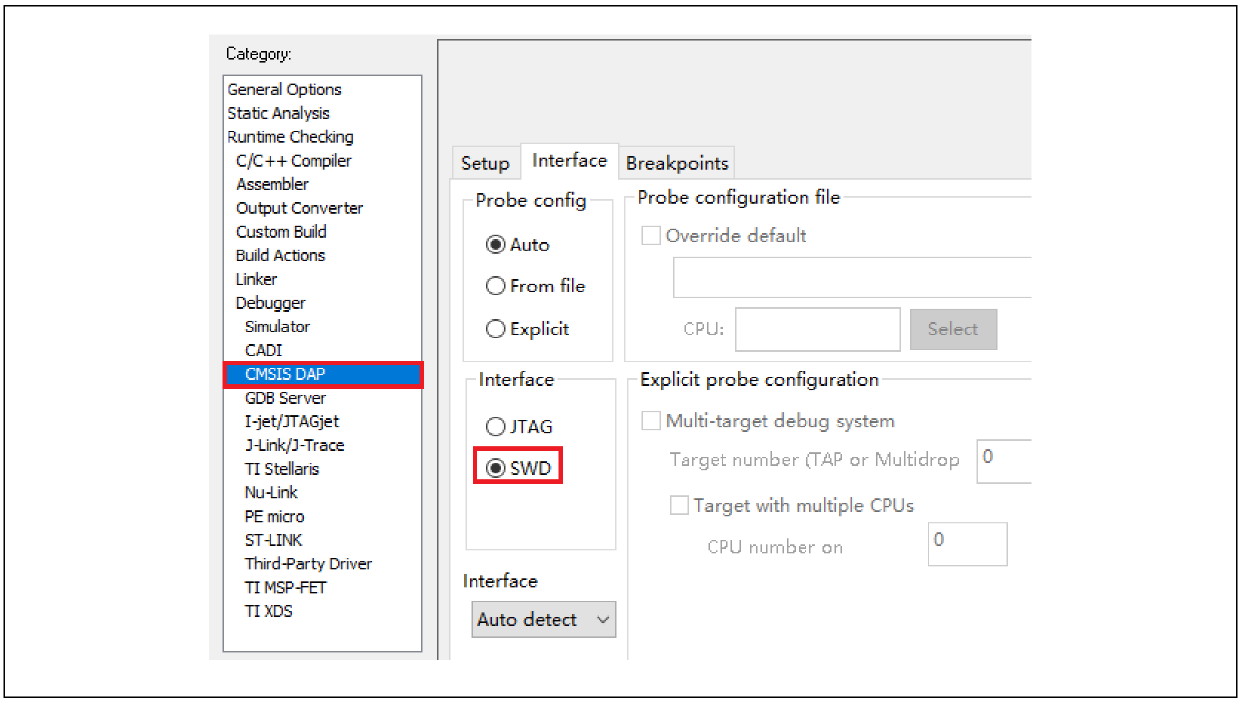


Figure 11. IAR CMSIS-DAP option


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/JTAGjet
J-Link/J-Trace
TI Stellaris
Nu-Link
PE micro
ST-LINK
Third-Party Driver
TI MSP-FET
TI XDS

Setup | Interface | Breakpoints

Probe config: (●) Auto ( ) From file ( ) Explicit

Probe configuration file: [ ] Override default; CPU: ; Select

Interface: ( ) JTAG (●) SWD

Explicit probe configuration: [ ] Multi-target debug system; Target number (TAP or Multidrop: 0; [ ] Target with multiple CPUs; CPU number on: 0

Interface: Auto detect

Note: For full details, please refer to AT32F435_437 Firmware BSP&Pack User Guide in BSP.

1.1.4 AT32 Work Bench

1.1.4.1 Introduction

This chapter describes how to use AT32 Work Bench. AT32 Work Bench makes it easier for developers to generate the initialized C codes and IDE projects through its graphic configuration features, reducing time and lower cost to development.

■ Environment requirements

Software:

Windows systems: Windows 7 or above

Linux systems: supports x86_64-based Ubuntu, Fedora, extra.

Hardware:

A minimum 2GB internal memory and 4GB hard disk is recommended.

Note: For details on AT32 Work Bench, refer to the document UM_AT32_Work_Bench which is available on ARTERY's official website. Our AT32 Work Bench tool has Linux and Windows versions.

1.1.4.2 Installing

Windows systems: no install is required. Just directly run the AT32_Work_Bench.exe。

Linux systems supports Ubuntu 16.4 or above versions. There are two ways to install it.

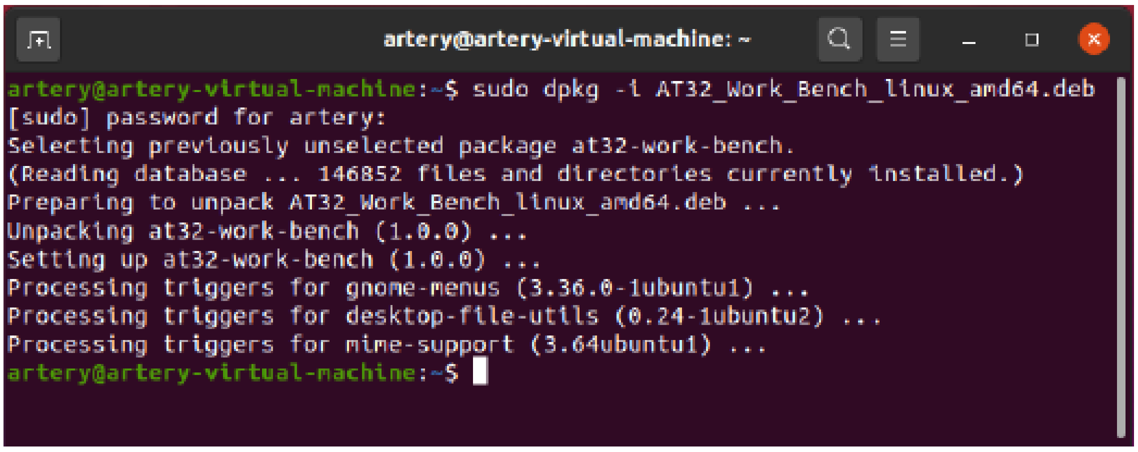
■ Use dpkg command to install in the terminal

Enter the following command:

sudo dpgk －i AT32_Work_Bench_Linux-x86_64_Vx.x.xx.deb

See Figure below:

Figure 12. Terminal command install

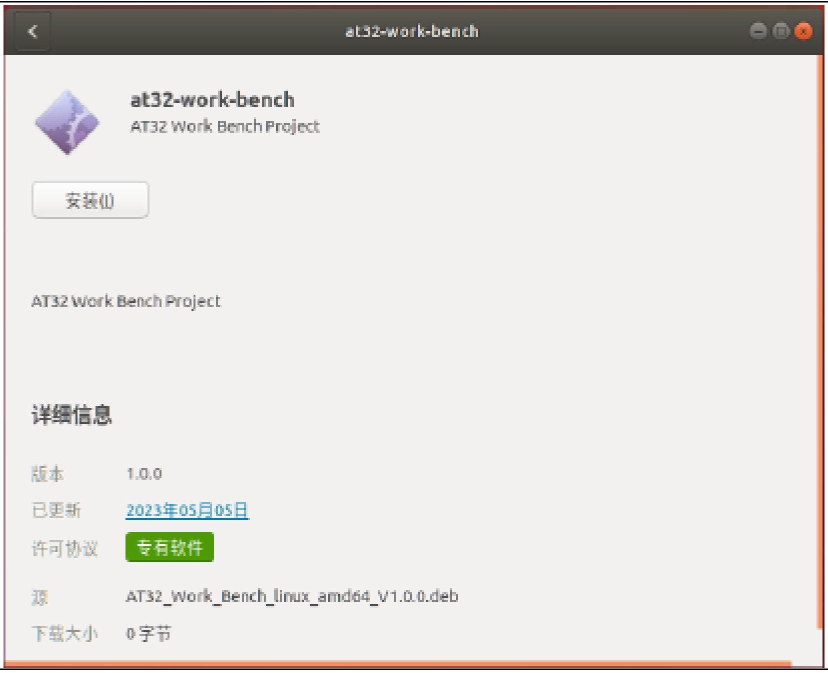


```
artery@artery-virtual-machine:~$ sudo dpkg -i AT32_Work_Bench_linux_amd64.deb
[sudo] password for artery:
Selecting previously unselected package at32-work-bench.
(Reading database ... 146852 files and directories currently installed.)
Preparing to unpack AT32_Work_Bench_linux_amd64.deb ...
Unpacking at32-work-bench (1.0.0) ...
Setting up at32-work-bench (1.0.0) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu2) ...
Processing triggers for mime-support (3.64ubuntu1) ...
artery@artery-virtual-machine:~$
```

■ Graphic install

Copy AT32_Work_Bench_Linux-x86_64_Vx.x.xx.deb into Linux system and double click it. In the pop-up window, click "install" button so that the software is automatically installed into the system, as shown in Figure below.

Figure 13. Graphic install



at32-work-bench

AT32 Work Bench Project

AT32 Work Bench Project

详细信息

版本	1.0.0
已更新	2023年05月05日
许可协议	专有软件
源	AT32_Work_Bench_linux_amd64_V1.0.0.deb
下载大小	0 字节

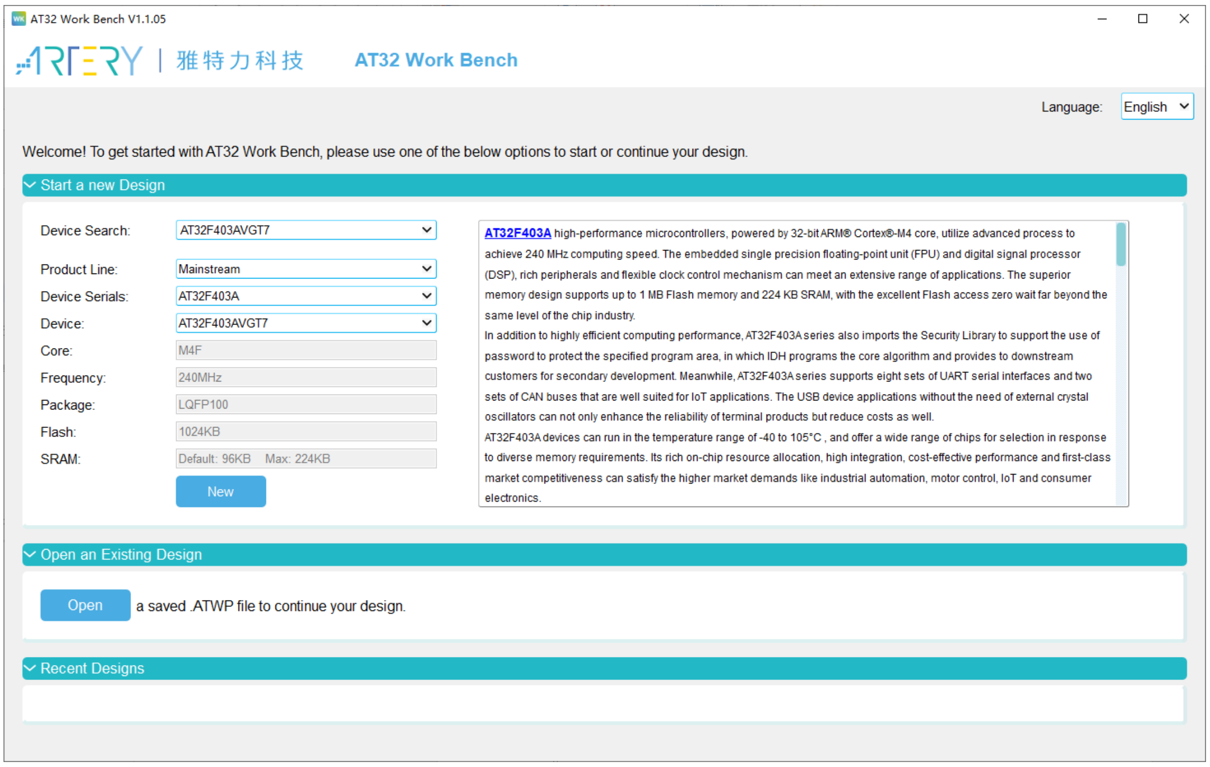
After finishing the install, click the lower bottom button "display all programs" on the left column, find the AT32_Work_Bench, click it to start.

1.1.4.3 Project configuration

This chapter uses the AT32F403AVGT7 to build a USART project as an example, demonstrating how to create the initialized C codes and IDE project with the AT32_Work_Bench tool.

The first window to come up when starting AT32 Work Bench is a guide page. In this page, there are three options: start a new design, open an existing design, and Recent designs.

Figure 14. AT32_Work_Bench guide page



Language: English

Welcome! To get started with AT32 Work Bench, please use one of the below options to start or continue your design.

Start a new Design

Device Search:	AT32F403AVGT7
Product Line:	Mainstream
Device Serials:	AT32F403A
Device:	AT32F403AVGT7
Core:	M4F
Frequency:	240MHz
Package:	LQFP100
Flash:	1024KB
SRAM:	Default: 96KB Max: 224KB

AT32F403A high-performance microcontrollers, powered by 32-bit ARM® Cortex®-M4 core, utilize advanced process to achieve 240 MHz computing speed. The embedded single precision floating-point unit (FPU) and digital signal processor (DSP), rich 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 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 specified program area, in which IDH programs the core algorithm and provides to downstream customers for secondary development. Meanwhile, AT32F403A series supports eight sets of UART serial interfaces and two sets of CAN buses that are well suited for IoT applications. The USB device applications without the need of external crystal oscillators can not only enhance the reliability of terminal products but reduce costs as well.

AT32F403A devices can run in the temperature range of -40 to 105°C , and offer a wide range of chips for selection in response to diverse memory requirements. Its rich on-chip resource allocation, high integration, cost-effective performance and first-class market competitiveness can satisfy the higher market demands like industrial automation, motor control, IoT and consumer electronics.

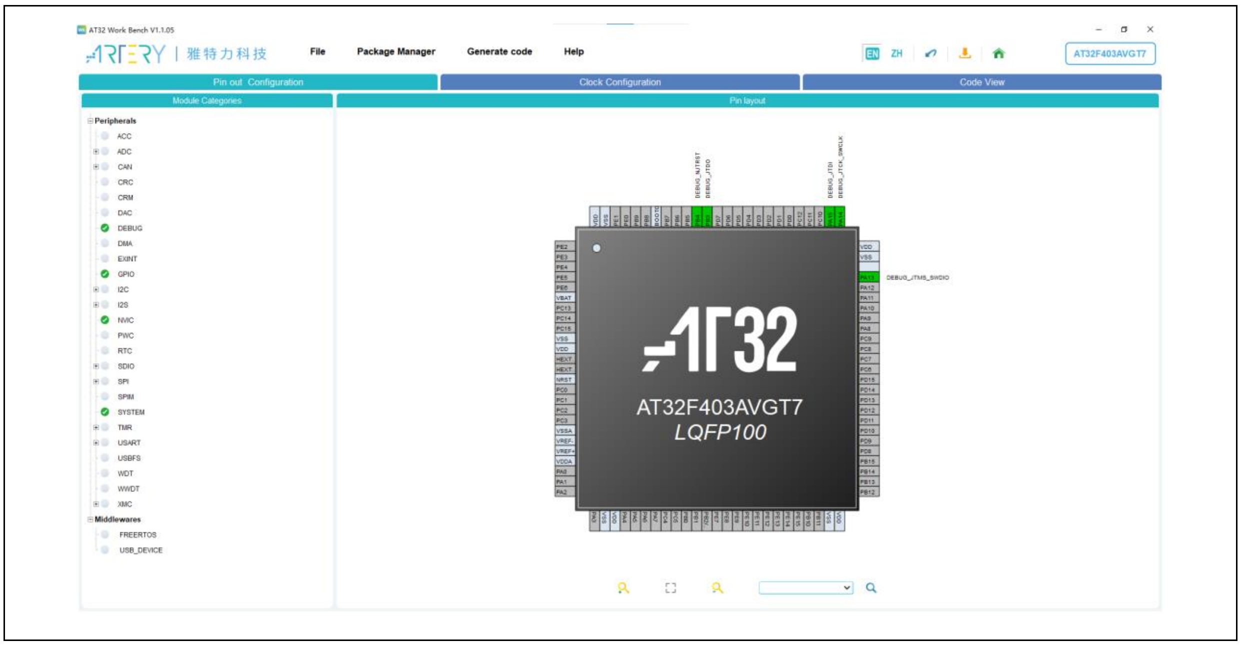
Open an Existing Design

a saved .ATWP file to continue your design.

Recent Designs

In guide page, select a MCU, click “Create” to enter project configuration interface. In the configuration page, there are three option boxes: pin out configuration, clock configuration and code view. They are three key functions that are detailed as follows.

Figure 15. Project configuration interface

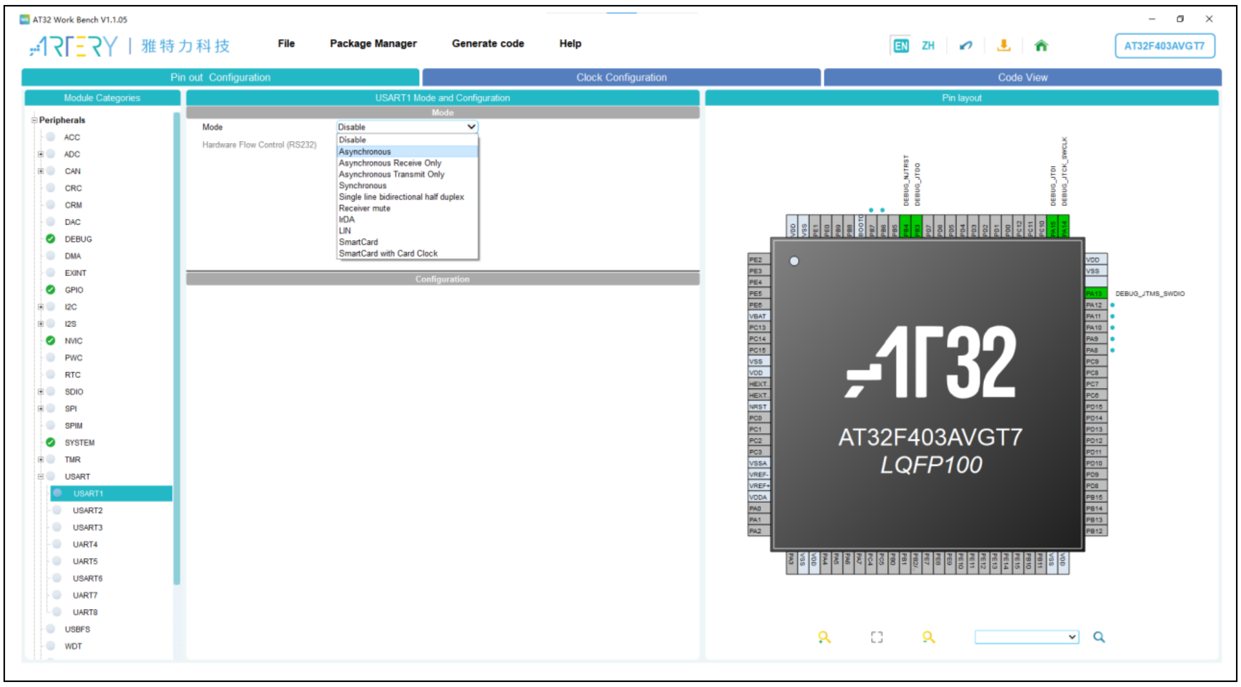


(1) Pin out configuration: used to configure peripherals and pins. It includes three windows: module categories, mode and configuration, and pin layout.

Select a peripheral from “module categories” list, pop up the corresponding “mode and configuration” window, select mode and parameters, such as, peripheral operating mode and the required pins.

As MCU supports the same pins to be used by different peripherals and functions, when the user selects a mode, the tool will automatically deliver a peripheral configuration tailored to the selected peripheral. As USART1 as an example, its mode and configuration window is shown in the figure below.

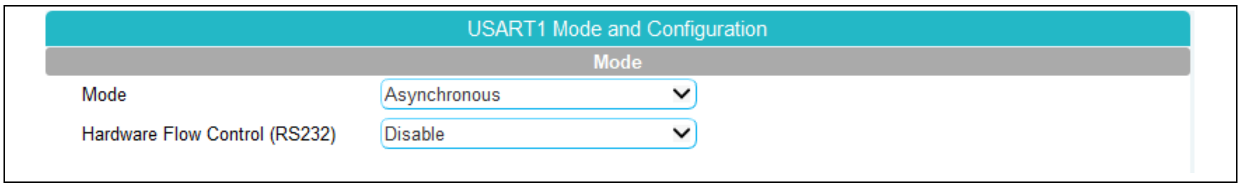
Figure 16. Mode and configuration window



Taking USART1 as an example, follow the steps below to configure USART1 peripheral.

■ Mode

Figure 17. Peripheral mode



USART1 Mode and Configuration

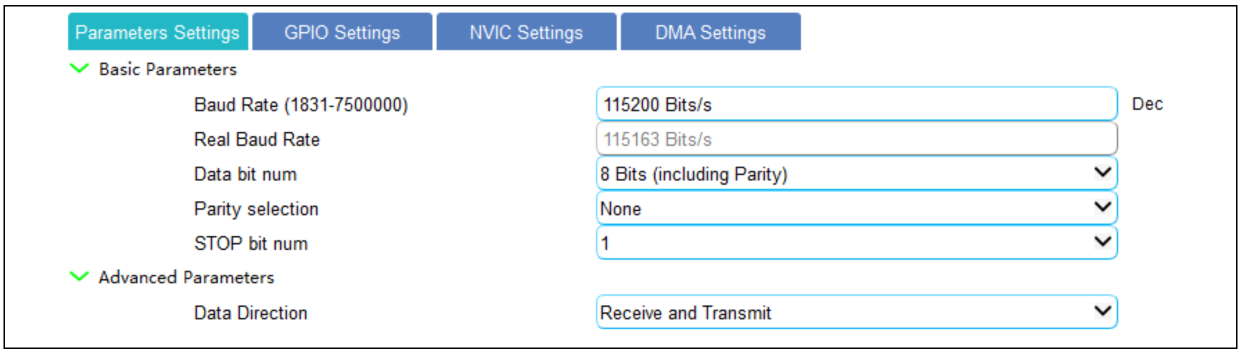
Mode

Mode	Asynchronous
Hardware Flow Control (RS232)	Disable

In Figure 17, we select Asynchronous mode for USART1. In pin layout view, PA9 and PA10 are automatically mapped onto USART1_TX and USART1_RX respectively.

■ Peripheral parameter settings

Figure 18. Peripheral parameter settings



Basic Parameters

Baud Rate (1831-7500000)	115200 Bits/s	Dec
Real Baud Rate	115163 Bits/s	
Data bit num	8 Bits (including Parity)	
Parity selection	None	
STOP bit num	1	

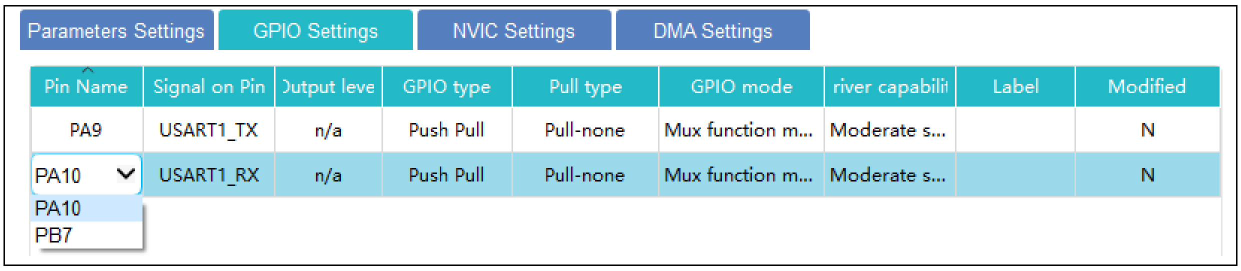
Advanced Parameters

Data Direction	Receive and Transmit

In Parameters Settings, we can see USART1 related parameters, such as, baud rate, data bit number, data direction.

■ GPIO settings

Figure 19. GPIO settings

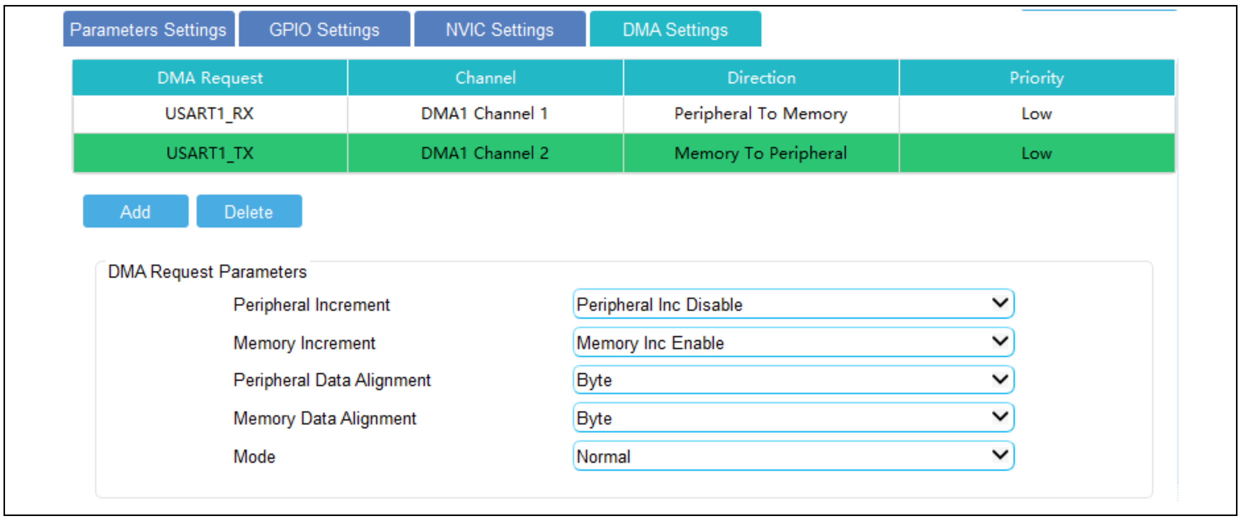


Pin Name	Signal on Pin	Dutput leve	GPIO type	Pull type	GPIO mode	river capabili	Label	Modified
PA9	USART1_TX	n/a	Push Pull	Pull-none	Mux function m...	Moderate s...		N
PA10	USART1_RX	n/a	Push Pull	Pull-none	Mux function m...	Moderate s...		N

In GPIO settings, it lists GPIOs that can be configured for USART1, such as, USART1_T
X and USART1_RX.

■ DMA settings

Figure 20. DMA settings



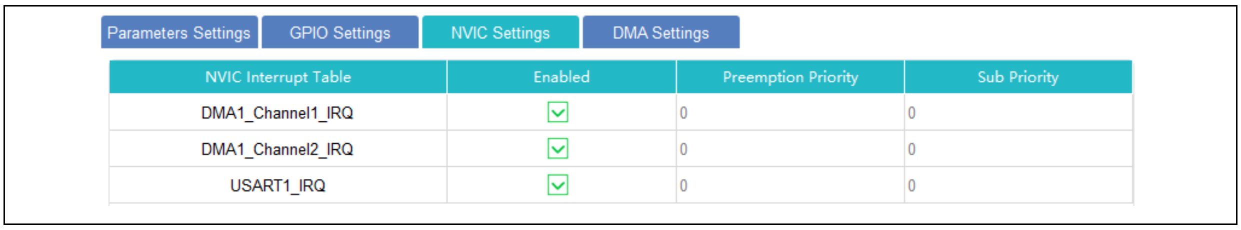
DMA Request	Channel	Direction	Priority
USART1_RX	DMA1 Channel 1	Peripheral To Memory	Low
USART1_TX	DMA1 Channel 2	Memory To Peripheral	Low

DMA Request Parameters

Peripheral Increment	Peripheral Inc Disable
Memory Increment	Memory Inc Enable
Peripheral Data Alignment	Byte
Memory Data Alignment	Byte
Mode	Normal

In DMA settings, the user can configure DMA request, such as DMA channel and DMA request parameters.

■ NVIC settings



NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
DMA1_Channel1_IRQ	✓	0	0
DMA1_Channel2_IRQ	✓	0	0
USART1_IRQ	✓	0	0

Figure 21. NVIC settings

In NVIC settings, the user can configure interrupts, including DMA channel interrupt when DMA channel is enabled. “Preemption priority” and “Sub priority” need to be configured in NVIC.

In Pin layout window, it will show the pin layout of a given package, such as LQFP48, QFN32, and TSSOP20. Also the pin name, configuration status and signal allocation are displayed.

In pin view window, a menu pops up by left clicking on the pin (except a pin with fixed mode), showing its signals to be configurable. For a pin with its function configured, an “Enter label” pops up by right clicking on the pin, as shown below:

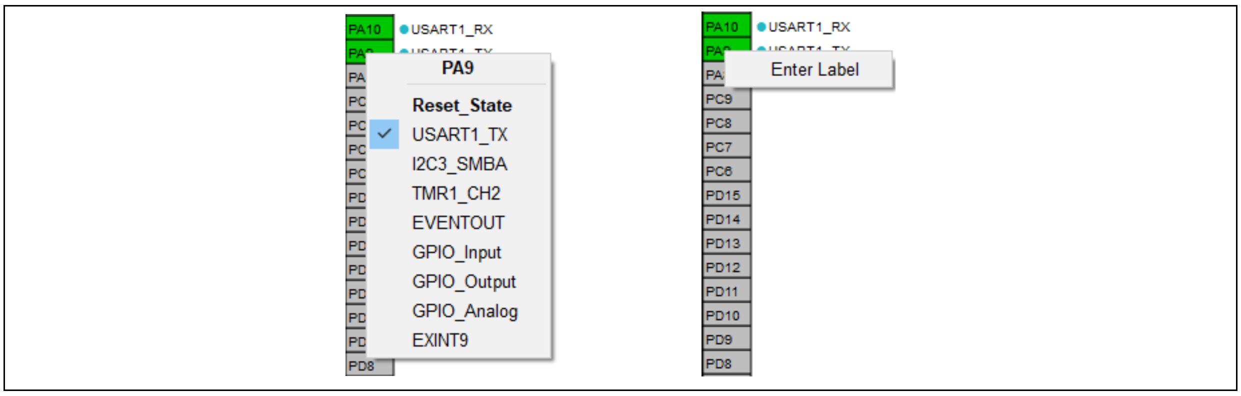


Figure 22. Left click and right click

(2) Clock configuration interface is used to configure clock path and related parameters. The drop-down menu and enter box are used to change clock tree configuration to meet actual requirements.

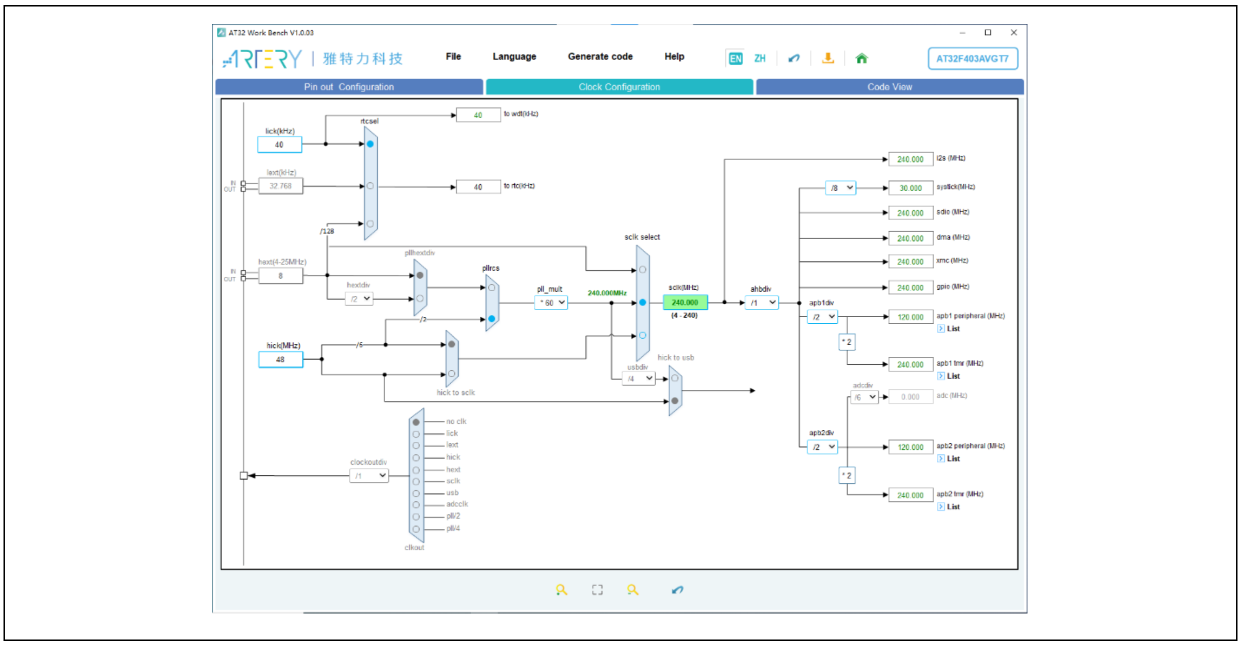
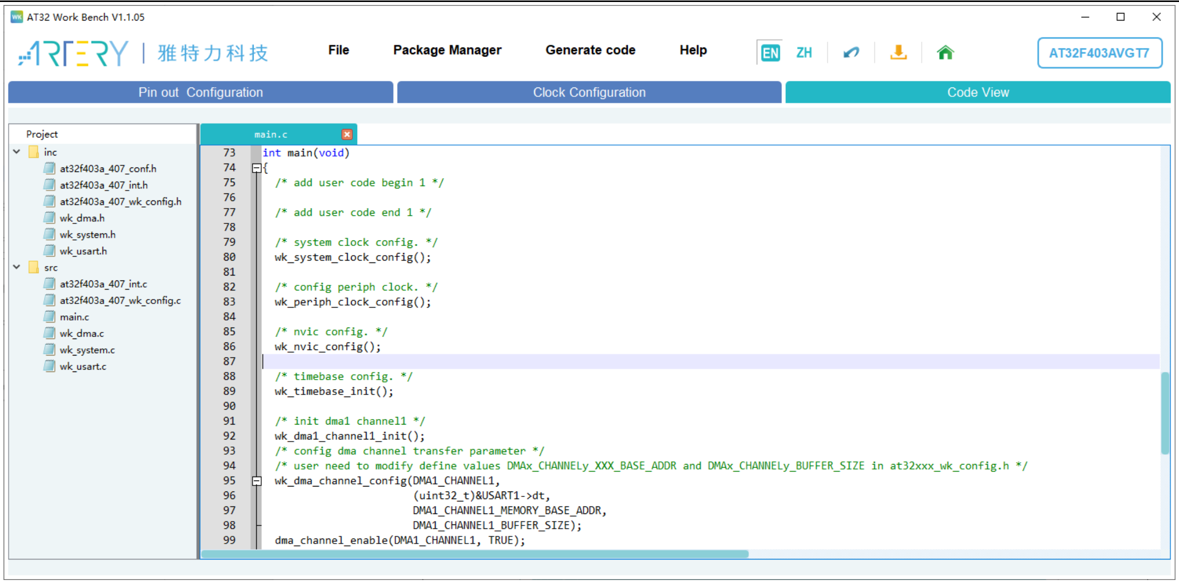


Figure 23. Clock configuration interface

Note 1: To use LEXT, it is necessary to configure LEXT mode in CRM mode window.
Note 2: To use HEXT, it is necessary to configure HEXT mode in CRM mode window.
Note 3: To use clockout, it is necessary to tick clock output in CRM mode window.

(3) Click on "Code view" to generate the code that is configured, and to show code content. In the left column shows code files that have been generated, and the right column shows the content of the selected file, as shown below.

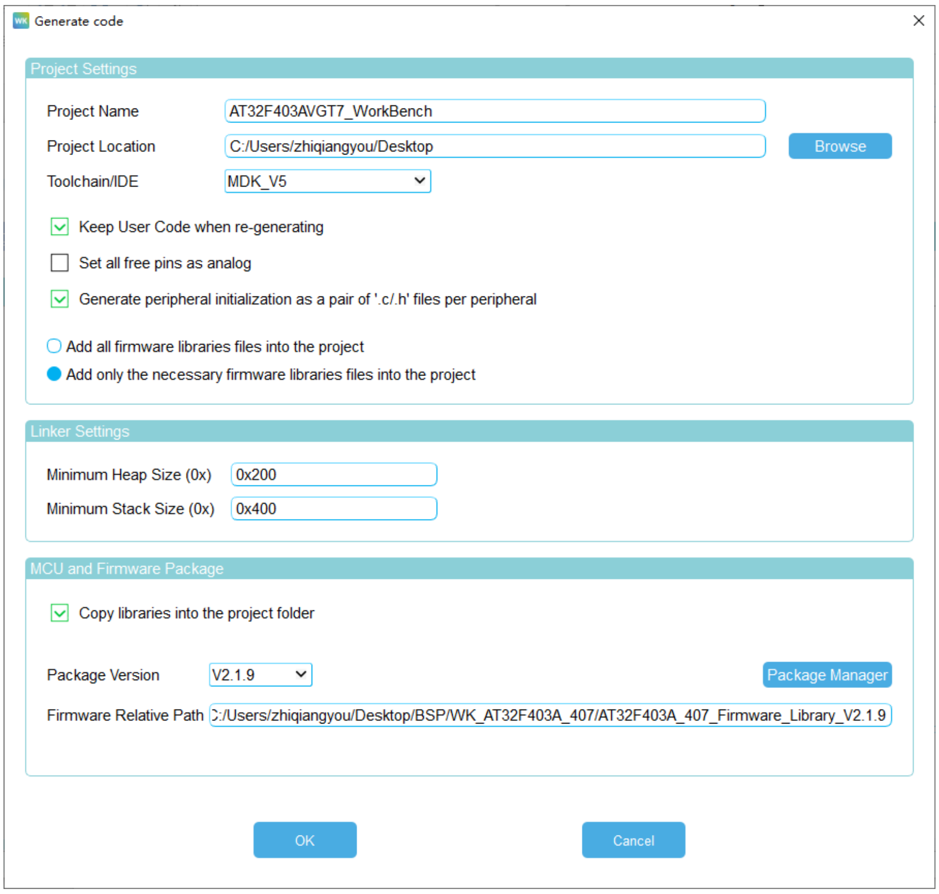
Figure 24. Code view



```
int main(void)
{
  /* add user code begin 1 */

  /* add user code end 1 */

  /* system clock config. */
  wk_system_clock_config();

  /* config periph clock. */
  wk_periph_clock_config();

  /* nvic config. */
  wk_nvic_config();

  /* timebase config. */
  wk_timebase_init();

  /* init dma1 channel1 */
  wk_dma1_channel1_init();
  /* config dma channel transfer parameter */
  /* user need to modify define values DMAx_CHANNELy_XXX_BASE_ADDR and DMAx_CHANNELy_BUFFER_SIZE in at32xxx_wk_config.h */
  wk_dma_channel_config(DMA1_CHANNEL1,
                        (uint32_t)&USART1->dt,
                        DMA1_CHANNEL1_MEMORY_BASE_ADDR,
                        DMA1_CHANNEL1_BUFFER_SIZE);
  dma_channel_enable(DMA1_CHANNEL1, TRUE);
```

Click on menu button or "generate code" button, a project management window is shown below:

Figure 25. Project management window



Project Name: AT32F403AVGT7_WorkBench
Project Location: C:/Users/zhiqiangyou/Desktop
Toolchain/IDE: MDK_V5

[✓] Keep User Code when re-generating
[ ] Set all free pins as analog
[✓] Generate peripheral initialization as a pair of '.c/.h' files per peripheral

○ Add all firmware libraries files into the project
● Add only the necessary firmware libraries files into the project

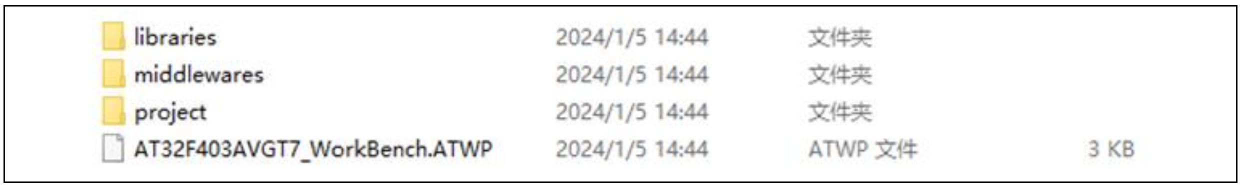
Minimum Heap Size (0x): 0x200
Minimum Stack Size (0x): 0x400

[✓] Copy libraries into the project folder

Package Version: V2.1.9
Firmware Relative Path: :/Users/zhiqiangyou/Desktop/BSP/WK_AT32F403A_407/AT32F403A_407_Firmware_Library_V2.1.9

The size of a heap and a stack is recommended to be 0x200 and 0x400 by default. However, these values may need to be increased when a middleware stack is to be used. After a library file is selected and copied into a project folder, the library file in the firmware is automatically copied into the project filer when generating code. After finishing project configuration, click on “confirm” button to generate user code and IDE project file.

Figure 26. Project file structure



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	ATWP 文件	3 KB

1.1.5 How to quickly replace AT32F403A/407 with AT32F435_437

- refer to MG018_Migrating_from_AT32F403A/407_to_AT32F435/437
- Step 6: If the program fails to run after above steps, please refer to other sections of this application note or contact local agents or ARTERY technicians for help.

1.2 AT32F435_437 MCU general function

1.2.1 Clock configuration

Refer to AN0084_AT32F435/437_CRM_Quick_Start_Guide

For clock configuration and code generation, it is recommended to use AT32_Work_Bench tool, or Win/Linux New_clock_configuration tool.

1.2.2 How to enable FPU function (Floating point unit)

There are two ways to enable FPU in Keil:

Please refer to AN0037_How_to_use_FPU

1.2.3 ZW/NZW Flash and embedded SRAM configuration

For more information about SRAM extension, please refer to FAQ0054_How_to_modify_SRAM_size_of_AT32 MCU?

1.2.4 Flash memory

1.2.4.1 Access protection

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

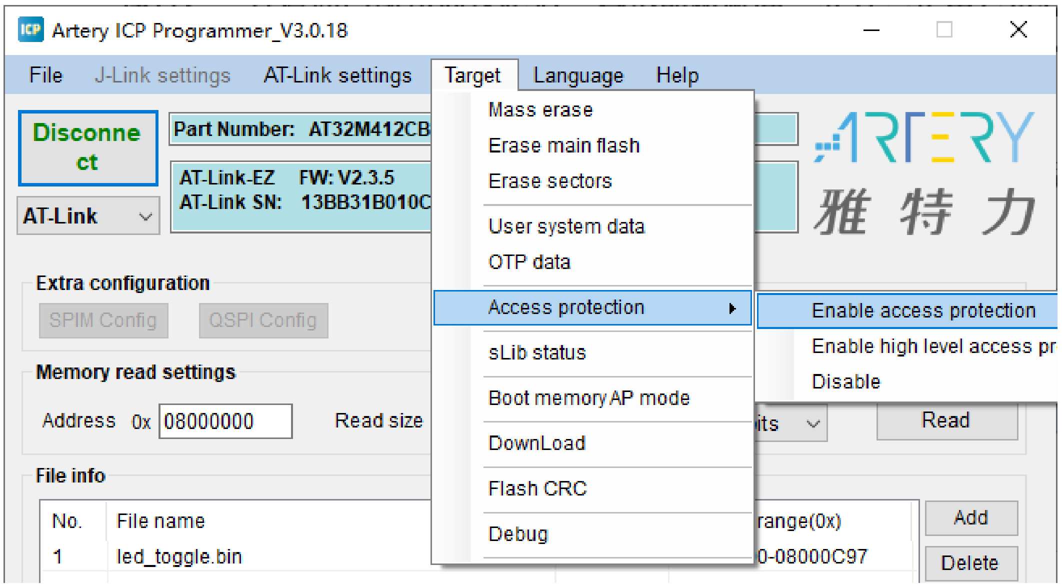
The ICP/ISP tool can be used to enable/ disable access protection based on the following procedures:

- Artery ICP Programmer

  Enable access protection: Target — Access protection — Enable (Y).

  Disable access protection: Target — Access protection — Disable (Y).

Figure 27. Enable/disable access protection with ICP



Part Number: AT32M412CB

AT-Link-EZ FW: V2.3.5

AT-Link SN: 13BB31B010C

Address 0x 08000000

No.	File name	range(0x)
1	led_toggle.bin	0-08000C97

■ Artery ISP Programmer (BOOT0=1)

Enable access protection: Enable/disable access protection — enable Access protection –Next —Yes

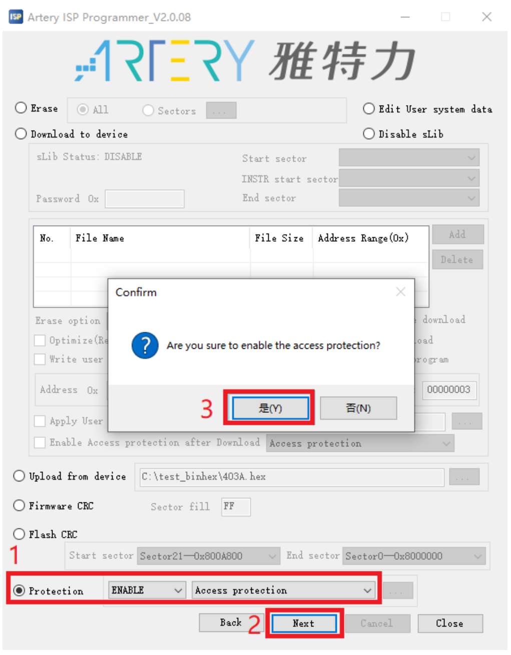
Disable access protection: Enable/disable access protection — disable Access protection - Next—Yes

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

Enable access protection: Enable/disable access protection — enable access protection — Start —Yes

Disable access protection: Enable/disable access protection — disable access protection — Start — Yes

Figure 28. Enable access protection with ISP



sLib Status: DISABLE

Password 0x

No.	File Name	File Size	Address Range(0x)

Are you sure to enable the access protection?

Address 0x 00000003

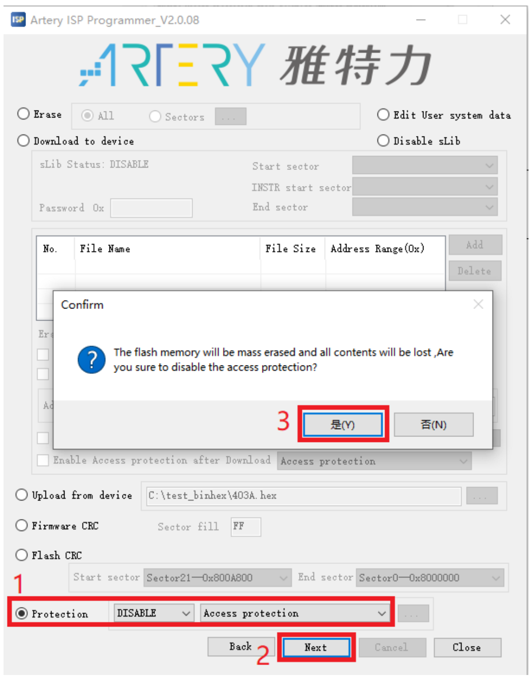
C:\test_binhex\403A.hex

Sector fill FF

Start sector Sector21—0x800A800 End sector Sector0—0x8000000

Protection [✓] ENABLE Access protection

Figure 29. Disable access protection with ISP



Artery ISP Programmer_V2.0.08

Confirm

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

sLib Status: DISABLE

C:\test_binhex\403A.hex

Sector fill FF

Start sector Sector21—0x800A800 End sector Sector0—0x8000000

Protection DISABLE Access protection

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

1.2.4.2 Erase and program protection

Write protection is applied to the entire Flash memory area or a given page in Flash memory. Once the Flash write protection is enabled, the internal Flash memory area cannot be written in any way.

The ICP/ISP programmer can be used to enable/disable erase and program protection based on the following procedures.

- Artery ICP Programmer

  Enable erase and program protection: Target — User system data — Tick “erase and program protection bytes” — Apply to device.

  Disable erase and program protection: Target — User system data — Untick “erase and program protection bytes” — Apply to device.

- Artery ISP Programmer (BOOT0=1)

  Enable erase and program protection: Enable/disable protection — enable Erase and program protection — Next — Yes

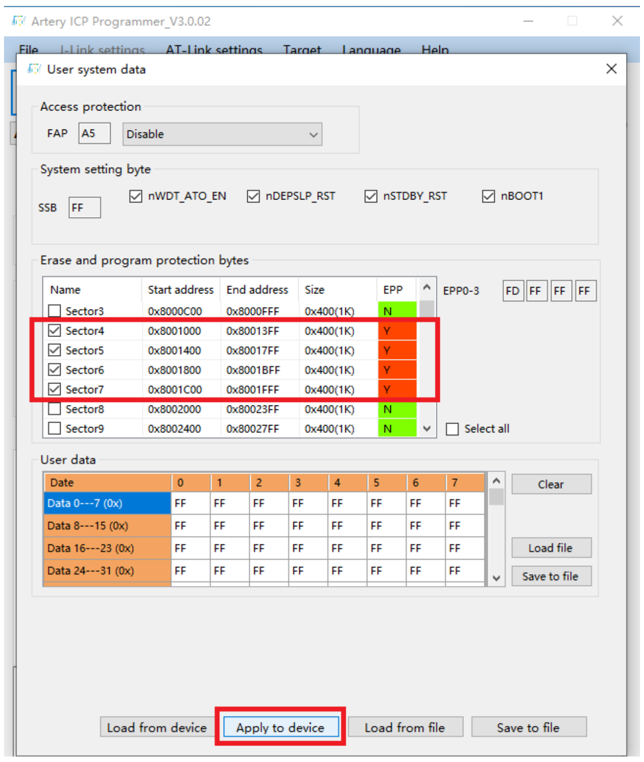
  Disable erase and program protection: Enable/disable protection — disable Erase and program protection — Next —Yes

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

  Enable erase and program protection: Enable/disable protection — enable Erase and program protection — Start —Yes

  Disable erase and program protection: Enable/disable protection — disable Erase and program protection — Start —Yes

Figure 30. Enable erase and program protection with ICP



Artery ICP Programmer_V3.0.02

File | J-Link settings | AT-Link settings | Target | Language | Help

User system data

Access protection

FAP A5 Disable

System setting byte

SSB FF ☑ nWDT_ATO_EN ☑ nDEPSLP_RST ☑ nSTBY_RST ☑ nBOOT1

Erase and program protection bytes

Name	Start address	End address	Size	EPP
☐ Sector3	0x8000C00	0x8000FFF	0x400(1K)	N
☑ Sector4	0x8001000	0x80013FF	0x400(1K)	Y
☑ Sector5	0x8001400	0x80017FF	0x400(1K)	Y
☑ Sector6	0x8001800	0x8001BFF	0x400(1K)	Y
☑ Sector7	0x8001C00	0x8001FFF	0x400(1K)	Y
☐ Sector8	0x8002000	0x80023FF	0x400(1K)	N
☐ Sector9	0x8002400	0x80027FF	0x400(1K)	N

EPP0-3 FD FF FF FF

☐ Select all

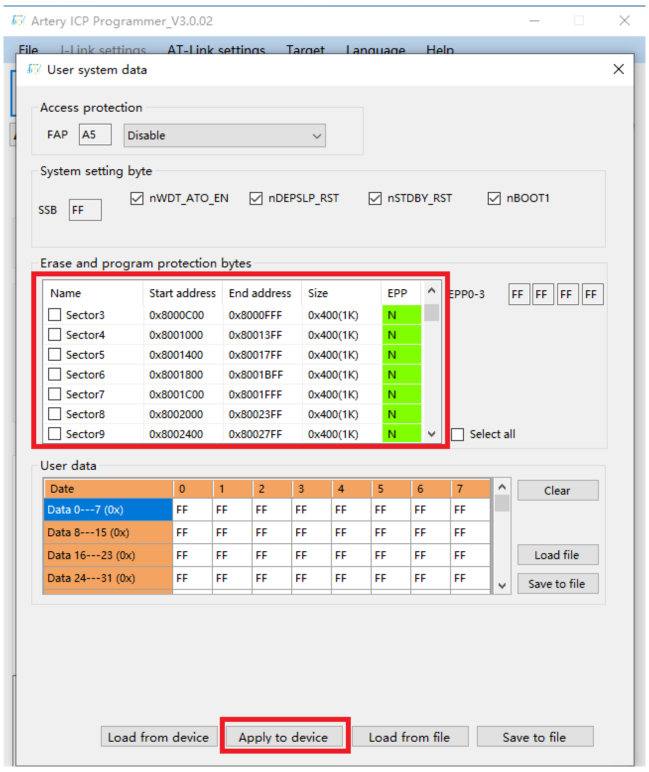
User data

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

Clear | Load file | Save to file

Load from device | Apply to device | Load from file | Save to file

Figure 31. Disable erase and program protection with ICP



Artery ICP Programmer_V3.0.02

File | J-Link settings | AT-Link settings | Target | Language | Help

User system data

Access protection

FAP A5 Disable

System setting byte

SSB FF ☑ nWDT_ATO_EN ☑ nDEPSLP_RST ☑ nSTBY_RST ☑ nBOOT1

Erase and program protection bytes

Name	Start address	End address	Size	EPP
☐ Sector3	0x8000C00	0x8000FFF	0x400(1K)	N
☐ Sector4	0x8001000	0x80013FF	0x400(1K)	N
☐ Sector5	0x8001400	0x80017FF	0x400(1K)	N
☐ Sector6	0x8001800	0x8001BFF	0x400(1K)	N
☐ Sector7	0x8001C00	0x8001FFF	0x400(1K)	N
☐ Sector8	0x8002000	0x80023FF	0x400(1K)	N
☐ Sector9	0x8002400	0x80027FF	0x400(1K)	N

EPP0-3 FF FF FF FF

☐ Select all

User data

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

Clear | Load file | Save to file

Load from device | Apply to device | Load from file | Save to file

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

1.2.5 Distinguish AT32 MCU from other MCU

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

Figure 32. Read Cortex ID

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

if((cortex_id == 0x410FC240) || (cortex_id == 0x410FC241))

{

        printf("This chip is Cortex-M4F.\r\n");

}

else

{

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

}
```

■ **Read PID and UID**

Figure 33. Read PID and UID

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

/* Store ID */
uint8_t    ID[5] = {0};

/* AT32F435 MCU type table */
const uint64_t AT32_MCU_ID_TABLE[] =
{
     0x0000000D70084540,          //AT32F435ZMT7     4032KB    LQFP144
     0x0000000D7008454F,          //AT32F437ZMT7     4032KB    LQFP144
     …
};

 /* 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)|((u
int64_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");
    }
}
```

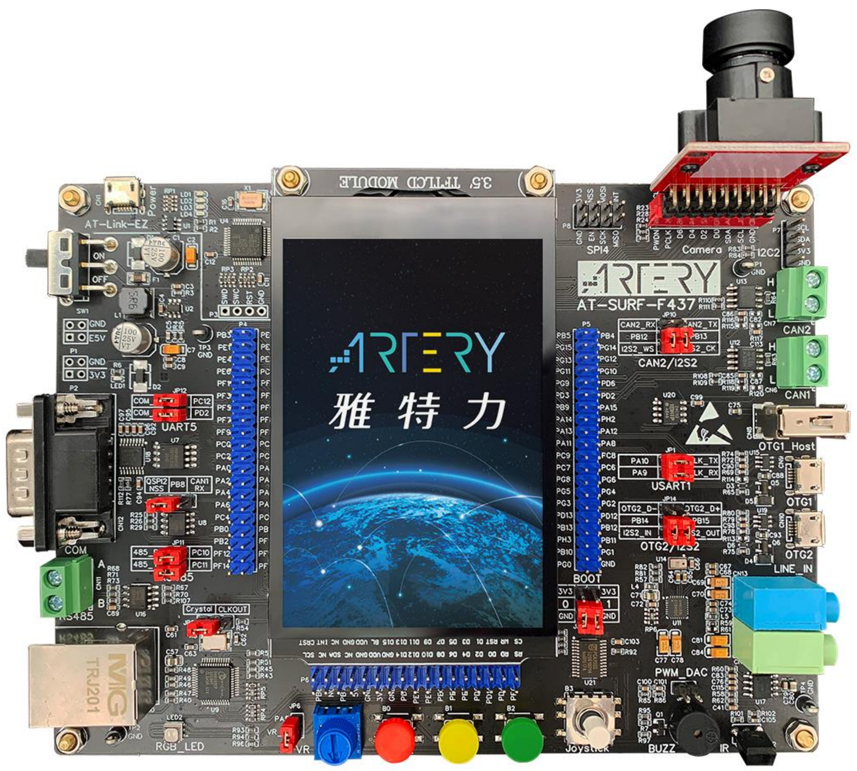
Note: AT32F4xx MCU contains multiple ID codes. By packing the obtained ID information into a 64-bit data, users can distinguish AT32 MCU from other ones. For details, refer to the corresponding Reference

Manual (DEBUG section) and AN0016_Recognize_AT32_MCU

1.2.6 Advanced features of AT32F435/F437

To make it easier to perform AT32F437 series-based application development, we offer the AT-SURF-F437 evaluation board alongside various demos and application codes. These documents can be found at AN0049_ AT_SURF_F437_Board_Application_Note located at ARTERY' official website → Support → AP Note→AN0049.

Figure 34. AT-SURF-F437 evaluation board



Note: The impro~~~~*ment to system performance is the result of a combination of factors that are conducive to performance optimization. For details on how to enhance the performane of AT32F435/AT32F437, please refer to* *FAQ0004_compile_function_or_variable to_designated address* and *AN0092_AT32F435/437_Performance_Improvement*

2 Frequently asked questions

2.1 Enter Hard Fault Handler

- SRAM size you are using exceeds the SRAM size configured in user system data
  See Section 1.2.3 and use ICP/ISP to extend SRAM space before programming.
- The single precision function is enabled in Keil or IAR while the M4-core FPU register is not enabled in the code. Please enable FPU function in the code:

Figure 35. Add FPU enable code

```
void SystemInit (void)
{
  /* Enable FPU*/
#if defined (__FPU_USED) && (__FPU_USED == 1U)
```

```
    SCB->CPACR |= ((3U << 10U * 2U) |          /* set CP10 Full Access */
                   (3U << 11U * 2U)   );        /* set CP11 Full Access */
    #endif
```

- Access data outside the boundary

  Find the location where data boundary is violated and change them back to normal range.
- System clock setting out of spec

2.2 J-Link fails to identify IC in Keil

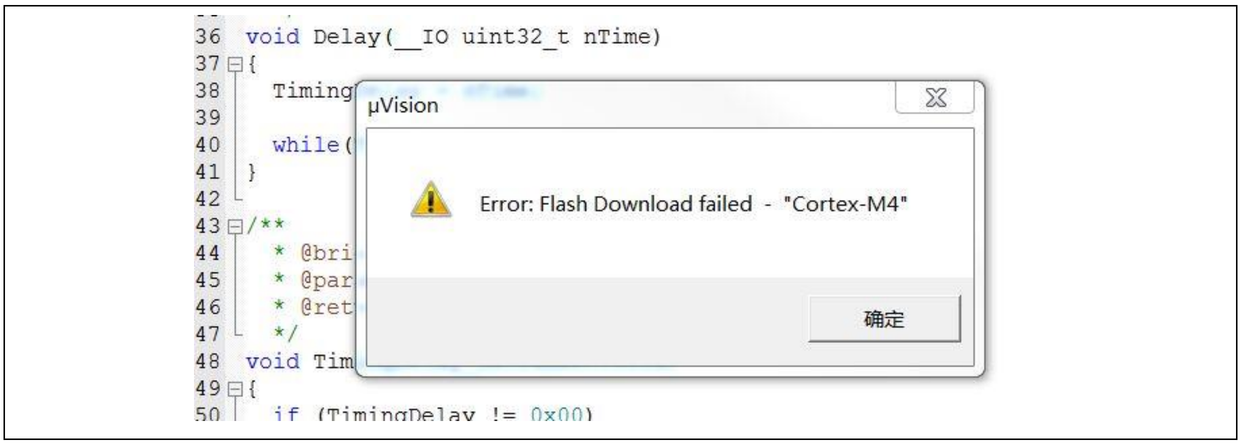
- Please refer to FAQ0008_J_LINK_not_identify_IC
- Please refer to FAQ0132_How_to_add_Artery_MCU_into_JLINK

2.3 Problems occurred in program download

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

The following warning message pops up during KEIL emulation or download:

Figure 36. Flash Download failed–"Cortex- M4"



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

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

µVision

Error: Flash Download failed - "Cortex-M4"

Here are the possible reasons behind this problem:

- Access protection is enabled. Please disable access protection before download
- Flash file algorithm is incorrect or Flash file algorithm is not loaded. Please add a correct Flash file algorithm
- BOOT0 and BOOT1 settings are incorrect. Set BOOT0=0 and BOOT1=0 to allow MCU to boot from the main Flash memory.
- Use older version of J-Link. Please use 6.20C or above.
- JTAG/SWD PIN is disabled. Please refer to "2.3.5 Resume download".

2.3.2 No Debug Unit Device found

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

2.3.3 RDDI-DAP Error

- Compiler optimization level is too high. For example, select "–Oz" as the default optimization level of Keil AC6. This actually should be changed to -O0/-O1.
- JTAG/SWD PIN is disabled. Please refer to "2.3.5 Resume download".

2.3.4 ISP serial port gets stuck during download

When the ISP serial port is used to download, it may be stuck so that it cannot be released. Following the procedure below:

- Check whether the power supply is stable
- Use a high-quality USB-to-serial port tool, i.e., CH340 chip.

2.3.5 How to resume download

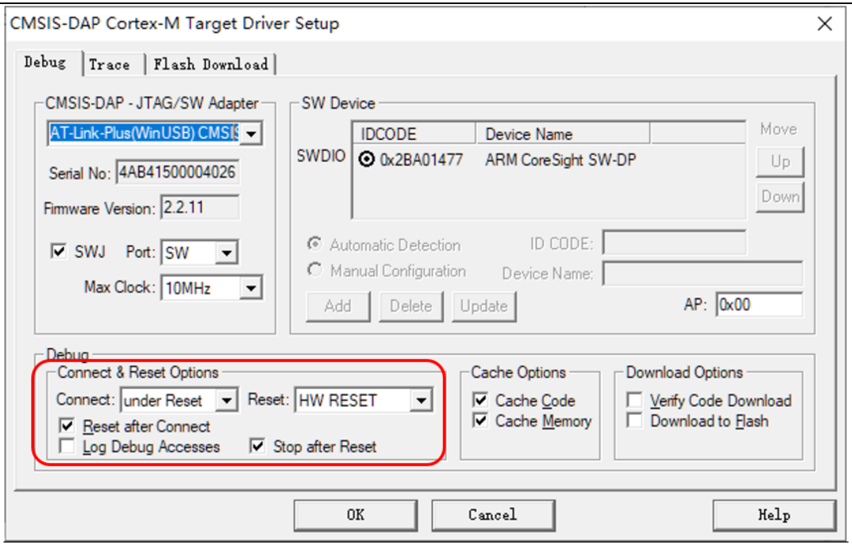
When users fail to download programs following the operations below on the AT32F403A/F407 series:

- After JTAG/SWD PIN is enabled, the program cannot be downloaded and JTAG/SWD device cannot not be identified
- After Standby mode and other low-power mode entry, the program cannot be downloaded and JTAG/SWD device cannot be identified

The solution to this problem is to halt MCU before program execution. Here are several methods for reference.

1. Change boot mode so that the microcontroller boots from Flash memory or SRAM, and then reset the microcontroller with Reset pin. By doing so, program can be erased and download operation is resumed
2. Use ICP tool and AT-Link debugger. Connect the AT-Link RST pin to the reset pin of the microcontroller. Click “connect” button on ICP interface. After successful connection, erase program to resume download
3. Use Keil and AT-Link debugger. Connect the AT-Link RST pin to the reset pin of the microcontroller. On Keil “Debug” window interface, select the following options marked in red box in the Figure below. This operation will erase program and resume download.

Figure 37. Debug interface



CMSIS-DAP Cortex-M Target Driver Setup

Debug | Trace | Flash Download

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

SW Device

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

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

Debug

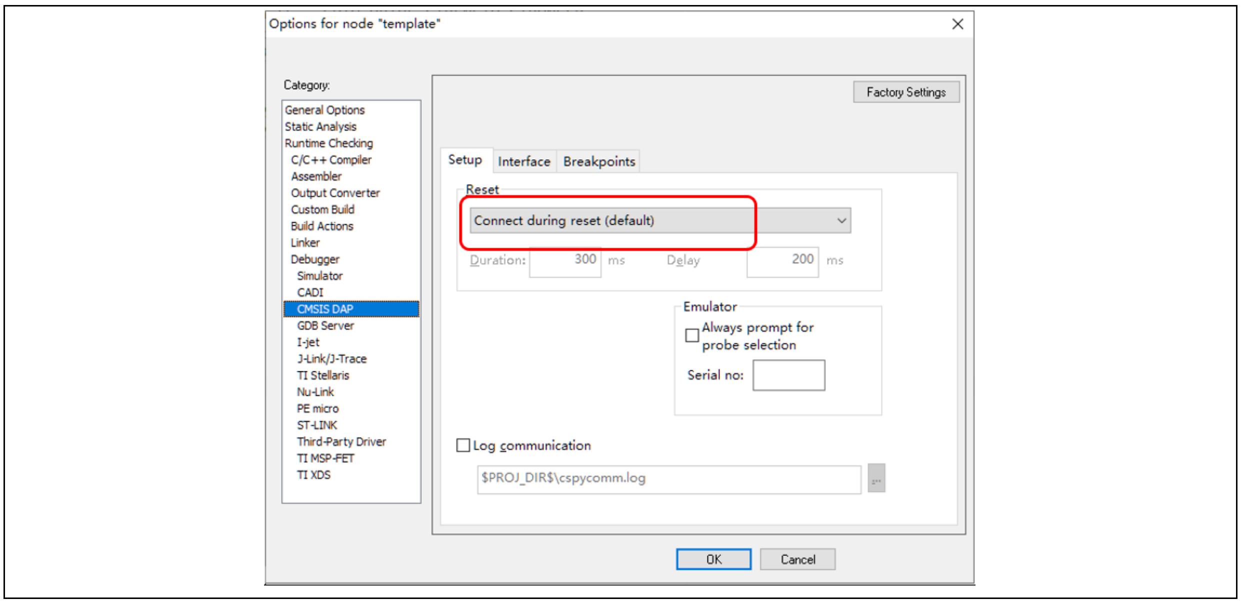
Connect & Reset Options
Connect: under Reset Reset: HW RESET
☑ Reset after Connect
☐ Log Debug Accesses ☑ Stop after Reset

Cache Options
☑ Cache Code
☑ Cache Memory

Download Options
☐ Verify Code Download
☐ Download to Flash

4. Use IAR and AT-Link debugger. Connect the AT-Link RST pin to the reset pin of the microcontroller. On IAR “CMSIS DAP” window interface, select the following options marked in red box in the Figure below. This operation will erase program and resume download.

Figure 38. CMSIS DAP interface



Reset: Connect during reset (default)
Duration: 300 ms Delay 200 ms
Emulator
☐ Always prompt for probe selection
Serial no:
☐ Log communication
$PROJ_DIR$\cspycomm.log

3 Security Library (sLib)

3.1 Introduction

As more and more MCU applications require complex algorithms and middleware solutions, how to protect core algorithms and related intellectual property rights of (such as core algorithms) software solution providers is emerging as an important topic and concern.

To provide robust protection function, the AT32F435/F437 series is equipped with a security library (sLib) to protect IP-Codes against being changed or read by end user's programs.

3.2 Principles of application

- Security library (sLib) is a secure area protected by an encryption key in the main memory. This area is used for software solution providers to store their core algorithms for better protection purpose. In the meantime, the rest of the area can be used for secondary-level development by terminal customers.
- Security library is divided into the instruction security library (SLIB_INSTRUCTION) and data security library (SLIB_DATA). Users can select part of or the whole security library for storing read-only area or instruction area.
- Read-only area (SLIB_READ_ONLY) can be read through I-Code and D-Code, but not written.
- Program codes in the instruction security library (SLIB_INSTRUCTION) can only be fetched (can only be executed) by MCU through I-Code bus but cannot be read through D-Code in the way it reads data (such as ISP/ICP/ debug mode or boot program from internal RAM). When accessing the SLIB_INSTRUCTION in the form of reading data, values return all 0xFF.
- Data in the data security library (SLIB_DATA) can only be read through D-Code bus but cannot be written.
- The program code and data in security library cannot be erased unless the correct code is keyed in. If a wrong password is entered in an attempt to write or erase the security library, a warning message will be issued by setting EPPERR=1 in the FLASH_STS register.
- The program code and data in security library are not erased when end users perform a mass

erase on the main Flash memory.

- After sLib protection is enabled, it is possible to unlock it by writing the previously preset password in the SLIB_PWD_CLR register. After the security library protection is unlocked, the MCU will perform a mass erase on the main Flash memory (including the contents of security library). As a result the program code will not be leaked even if the password set by the software solution provider is leaked.

3.3 How to use Security library

For details on security library, refer to AN0081_AT32F435/437_Security_Library_Application_Note

4 Revision history

Table 1. Document revision history

Date	Version	Revision note
2022.04.20	2.0.0	Initial release.
2022.07.08	2.0.1	Updated some descriptions.
2022.10.11	2.0.2	Added description of development environment and file path.
2022.10.21	2.0.3	Updated descriptions of UID and PID.
2024.01.05	2.0.4	Updated the descriptions on Resume AT32 MCU download;
2025.01.13	2.0.5	Optimized some descriptions