
ISP Programmer User manual

Introduction

This user manual gives an overview of Artery ISP Programmer. ISP Programmer acts a graphic Interface application designed to facilitate the use of ARTERY MCU. With the help of this programmer, users can perform ARTERY MCU devices through UART or USB ports.

Contents

1	Introduction	7
1.1	Environmental requirements.....	7
1.2	Glossary	7
2	Installation	8
3	USB DFU driver installation	9
3.1	Install driver automatically	9
3.2	Install driver manually	9
4	interfaces	14
4.1	AT32F403 interfaces.....	14
4.2	AT32F413 interfaces.....	14
4.3	AT32F415 interfaces.....	14
4.4	AT32F403A/F407 interfaces.....	14
4.5	AT32F421 interfaces.....	15
4.6	AT32F435/F437 interfaces	15
4.7	AT32WB415 interfaces.....	15
4.8	AT32F425 interfaces.....	16
4.9	AT32L021 interfaces.....	16
4.10	AT32F423 interfaces.....	16
4.11	AT32A403A interfaces	16
4.12	AT32F402/F405 interfaces	17
4.13	AT32A423 interfaces	17
4.14	AT32M412/M416 interfaces.....	18
5	User Interface	19
5.1	Connection settings	19
5.1.1	UART connection	19
5.1.2	DFU connection.....	21
5.2	Flash status page.....	22

5.3	Device Information page.....	23
5.4	Operation configuration page	28
5.4.1	Erase	29
5.4.2	Edit User system data	30
5.4.3	Download to device	36
5.4.4	Disable sLib	39
5.4.5	Upload from device	39
5.4.6	Firmware CRC.....	40
5.4.7	Flash CRC	40
5.4.8	Protection	41
5.5	Operation progress page.....	42
5.6	SPIM encryption download.....	43
6	Revision history	45

List of tables

Table 1. AT32F403 GPIO Pin Map.....	14
Table 2. AT32F413 GPIO Pin Map.....	14
Table 3. AT32F415 GPIO Pin Map.....	14
Table 4. AT32F403A/F407GPIO Pin Map	14
Table 5. AT32F421 GPIO Pin Map.....	15
Table 6. AT32F435/F437 GPIO Pin Map	15
Table 7. AT32WB415 GPIO Pin Map	15
Table 8. AT32F425 GPIO Pin Map.....	16
Table 9. AT32L021 GPIO Pin Map.....	16
Table 10. AT32F423 GPIO Pin Map.....	16
Table 11. AT32A403A GPIO Pin Map	16
Table 12. AT32F402/F405 GPIO Pin Map	17
Table 13. AT32A423 GPIO Pin Map	17
Table 14. AT32M412/M416 GPIO Pin Map.....	18
Table 15. Document revision history.....	45

List of figures

Figure 1. DFU driver install	9
Figure 2. Manual install-driver location	10
Figure 3. Manual install-device manager.....	10
Figure 4. Manual install-update driver software	11
Figure 5. Manual install-browse my computer for driver	11
Figure 6. Manual install-select driver software	12
Figure 7. Manual install-driver software installing	12
Figure 8. Manual install successful.....	13
Figure 9. UART connection window	19
Figure 10. USB interface auto connection diagram.....	20
Figure 11. DFU connection window	21
Figure 12. Flash status window	23
Figure 13. Device information.....	24
Figure 14. SPIM selection.....	26
Figure 15. SPIM name	26
Figure 16. SPIM name	26
Figure 17. Operation configuration	28
Figure 18. Sector erase selection	29
Figure 19. Block erase selection.....	29
Figure 20. User system data.....	30
Figure 21. Erase and program protection bytes	32
Figure 22. User data	34
Figure 23. SPIM encryption key.....	34
Figure 24. QSPI encryption key.....	35
Figure 25. Download to device	36
Figure 26. Download file selection.....	37
Figure 27. Disable sLib	39
Figure 28. Upload from device.....	39
Figure 29. Firmware CRC	40
Figure 30. Flash CRC	40
Figure 31. Enable erase and program protection.....	41
Figure 32. Operation progress display	42
Figure 33. Encryption range config.....	43

Figure 34. SPIM encryption key config	44
---	----

1 Introduction

1.1 Environmental requirements

- **Software requirements**

Windows 7 and above are required.

Software version below 2.0.04, .Net framework 4.0 is required.

Software version 2.0.04 and above, .Net framework 4.6 is required.

- **Hardware requirements**

Serial communication port (COM).

USB communication port.

1.2 Glossary

- **ISP:**

This refers to in-system programming so that user can directly perform write or erase operations on the chip.

- **UART:**

Universal Asynchronous Receiver/Transmitter. It is a serial communication port (COM) for full-duplex asynchronous communication.

- **USB:**

Universal Serial Bus. It is an external bus standard used to regulate the connection and communication between computers and external devices.

- **DFU:**

Device Firmware Upgrade. It is a device firmware update protocol based on USB communication.

2 Installation

- **Hardware installation**

UART communication: the device must be connected to the serial communication port (COM) on the computer. DFU communication: the device must be connected to USB port on the computer.

- **USB DFU driver installation**

If the USB DFU communication is used, the USB DFU driver must be installed. Please refer to the chapter USB DFU driver installation for detailed information.

- **Software installation**

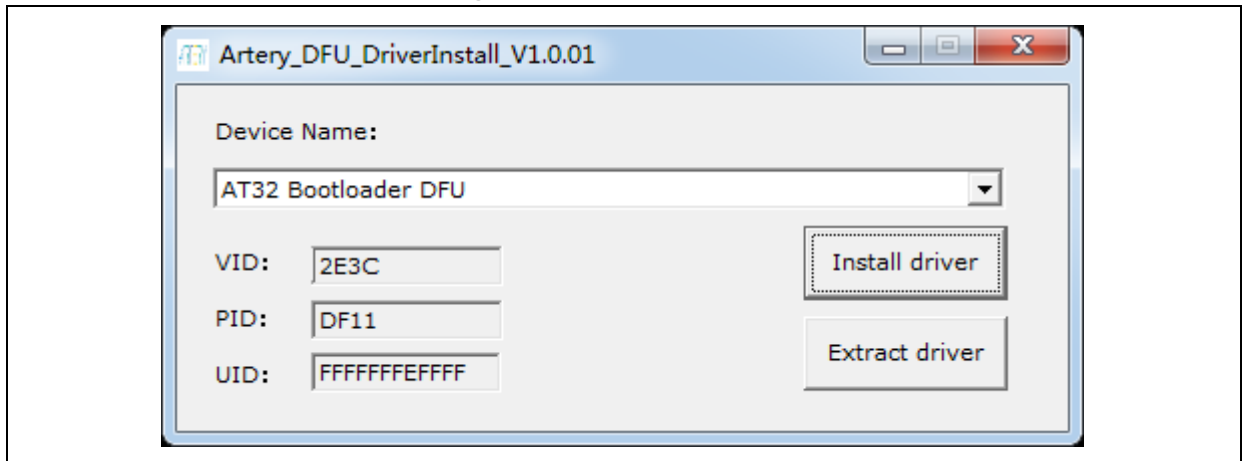
This software is not required, just directly run the executable program "ArteryISPProgrammer.exe".

3 USB DFU driver installation

Artery provides the USB DFU driver automatic installation program "Artery_DFU_DriverInstall.exe". Double-click it to enter the installation interface. (As shown in Figure 1)

The driver installation program will automatically scan all the "AT32 Bootloader DFU" devices connected to the computer. When the devices are connected, the "VID", "PID", and "UID" of each device can be displayed respectively.

Figure 1. DFU driver install



3.1 Install driver automatically

Click on "**Install driver**" button to start the automatic installation of the driver.

If the installation is successful, a successful installation message will be displayed.

If failed, an error message will be displayed.

If the driver is already installed, "**Install driver**" will become "**Reinstall driver**". Click on this button will reinstall the driver.

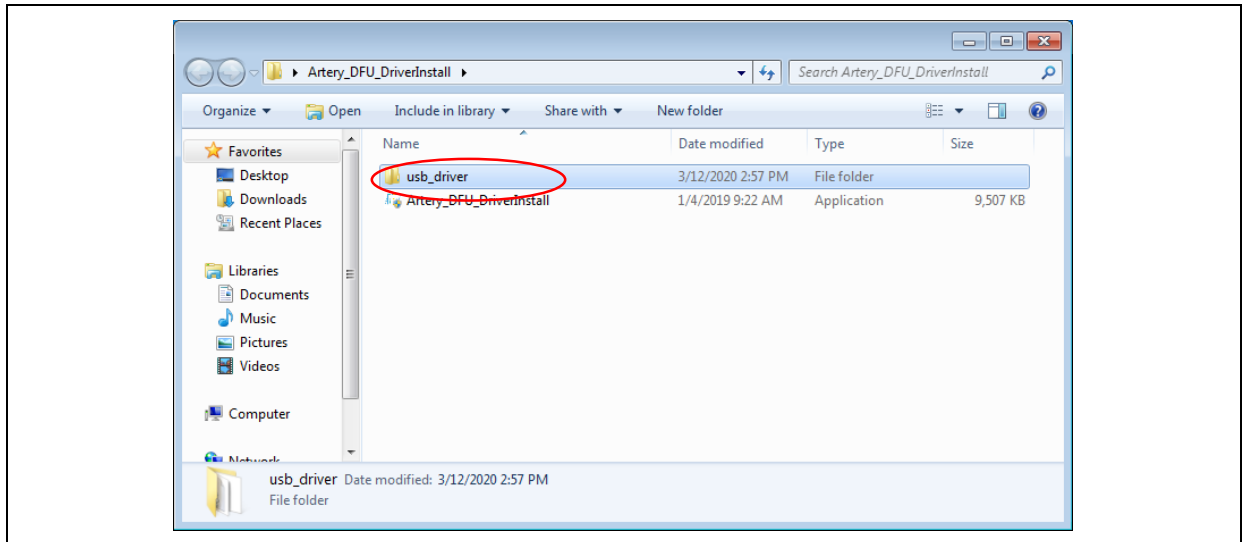
3.2 Install driver manually

When the automatic installation failed or the user needs to install the driver manually, refer to this chapter for manual Installation.

Click on "**Extract driver**" button, a driver installation package ("**usb_driver**" folder) will be generated in the current Directory (As shown in Figure 2).

This installation package is only available for the currently running operating system. If it is applied to other operating systems, the installation may fail.

Figure 2. Manual install-driver location

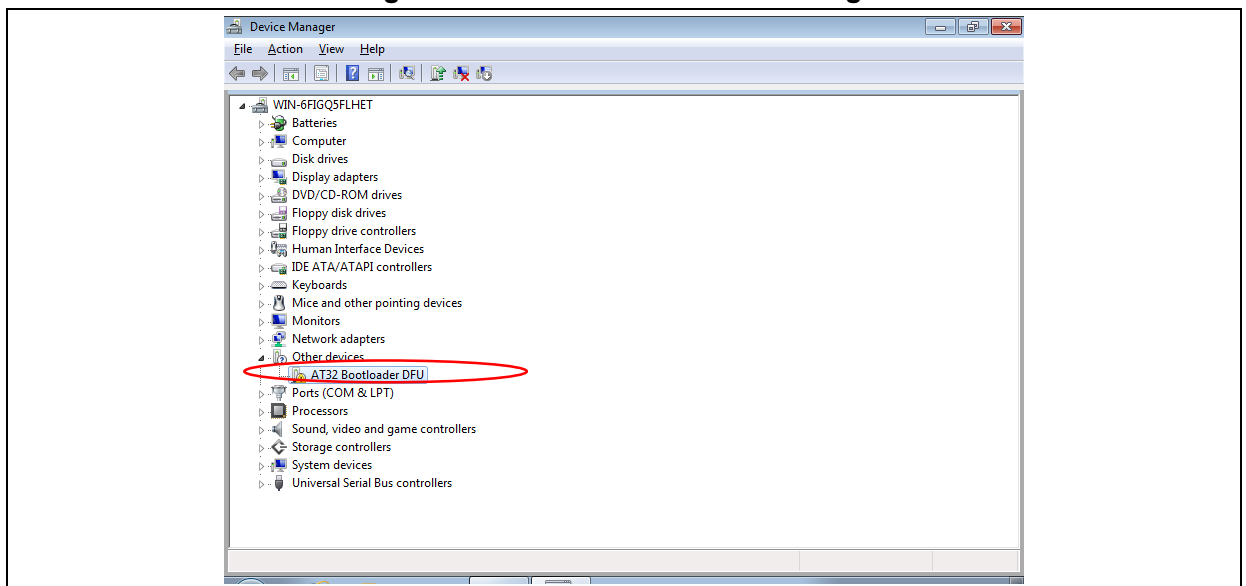


The procedures of manual installation are as follows (take windows7 as an example):

- Open the "**Device Manager**" (As shown in Figure 3)

First make sure that the "**AT32 Bootloader DFU**" device is properly connected to the computer.

Figure 3. Manual install-device manager

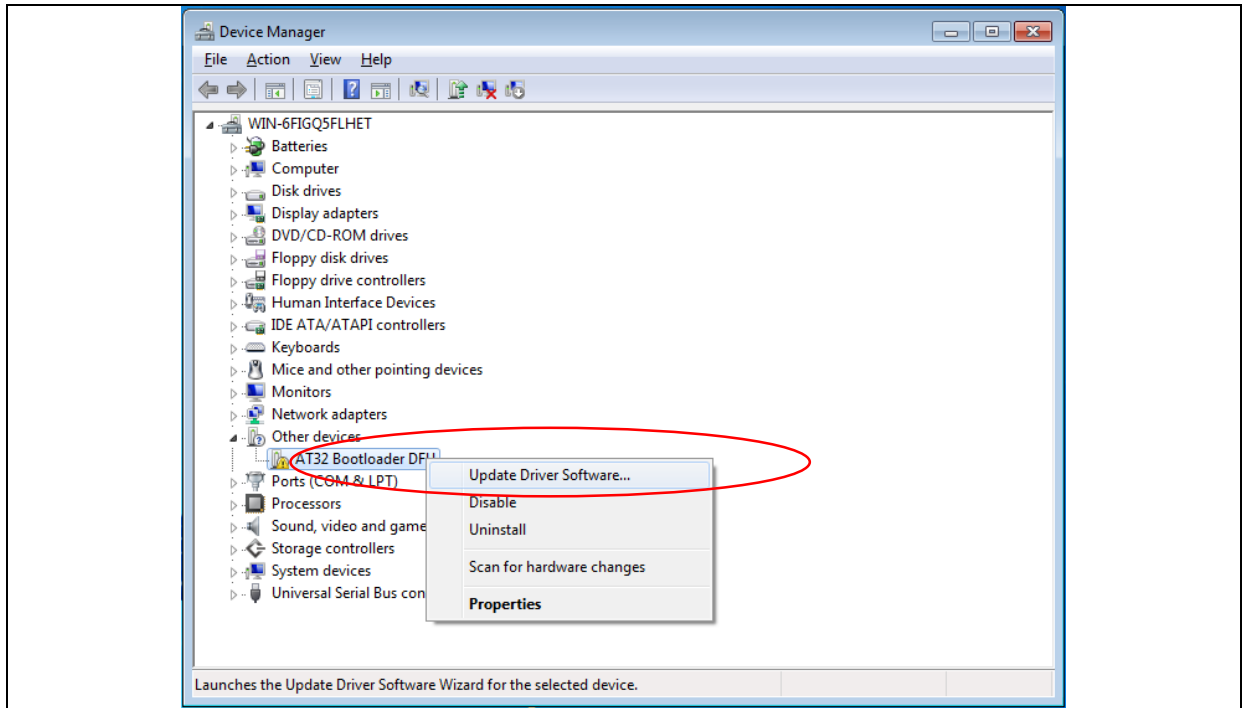


In this case, the "**Device Manager**" will scan the device "**AT32 Bootloader DFU**" without driver installed.

If the device "**AT32 Bootloader DFU**" is not found, please rescan it, that is, click on the "**Device Manager**"-"**Action**" menu and select "**Scan for hardware changes**".

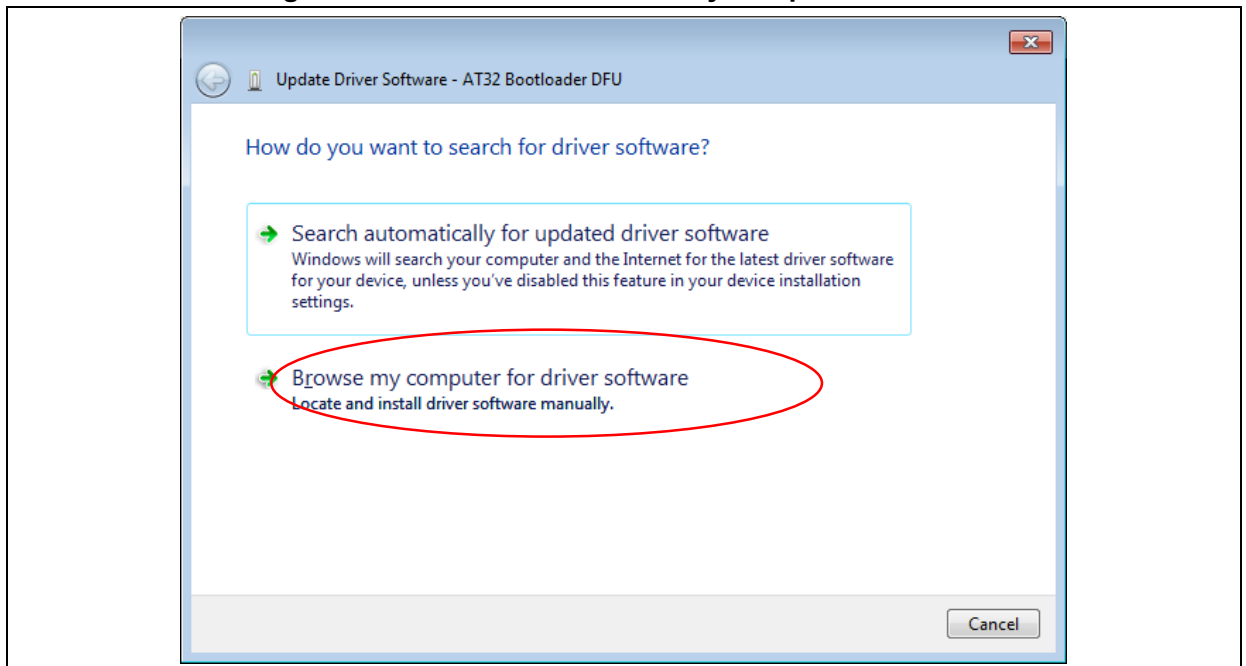
- Right-click on the device "**AT32 Bootloader DFU**" and select "**Update Driver Software**" (As shown in Figure 4).

Figure 4. Manual install-update driver software



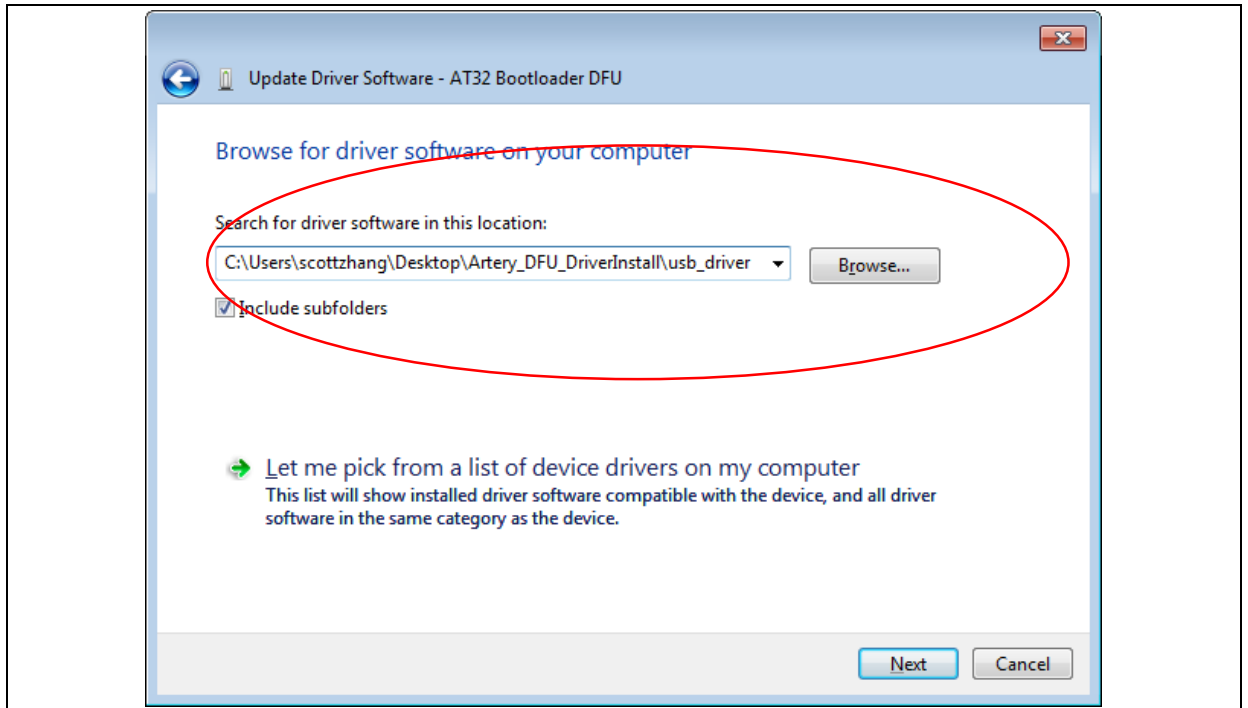
- Select "**Browse my computer for driver software**". (As shown in Figure 5)

Figure 5. Manual install-browse my computer for driver



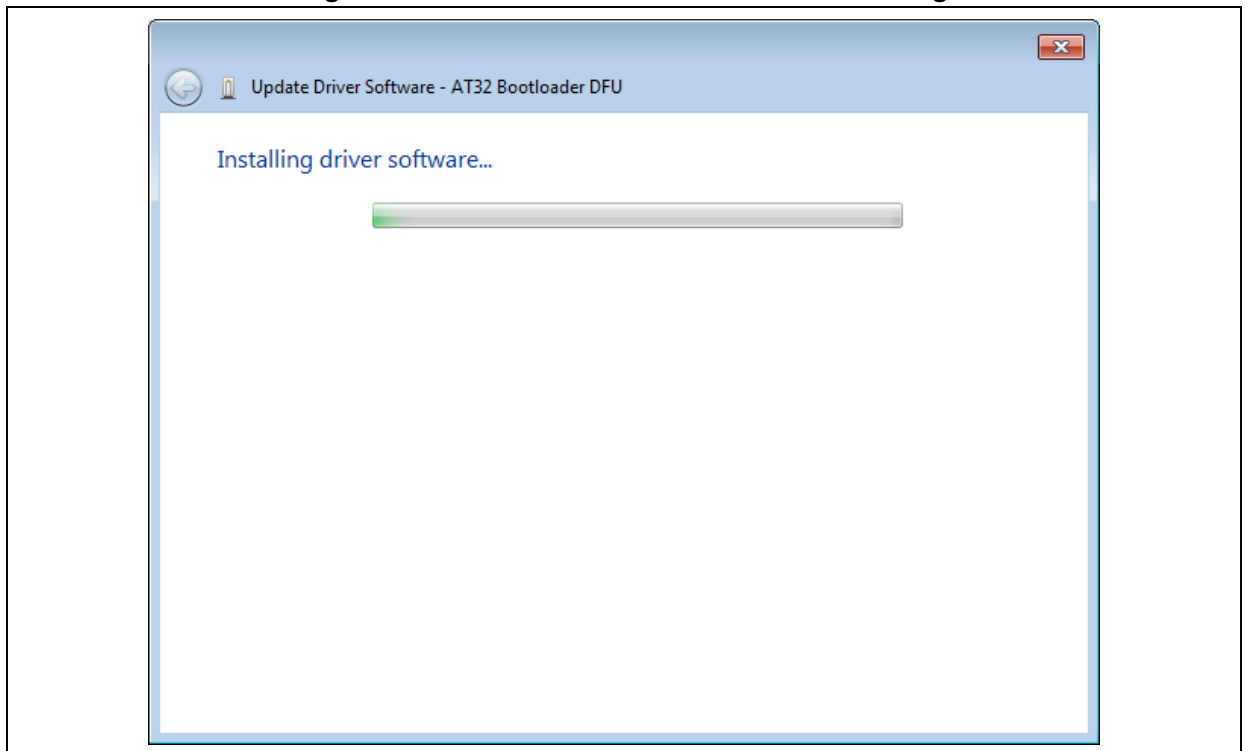
- Please select the location of the driver correctly, that is, click on "**Extract driver**" to generate a driver installation package ("**usb_driver**" folder). Then click on "**Next**" (As shown in Figure 6).

Figure 6. Manual install-select driver software



- Installing driver software. (As shown in Figure 7).

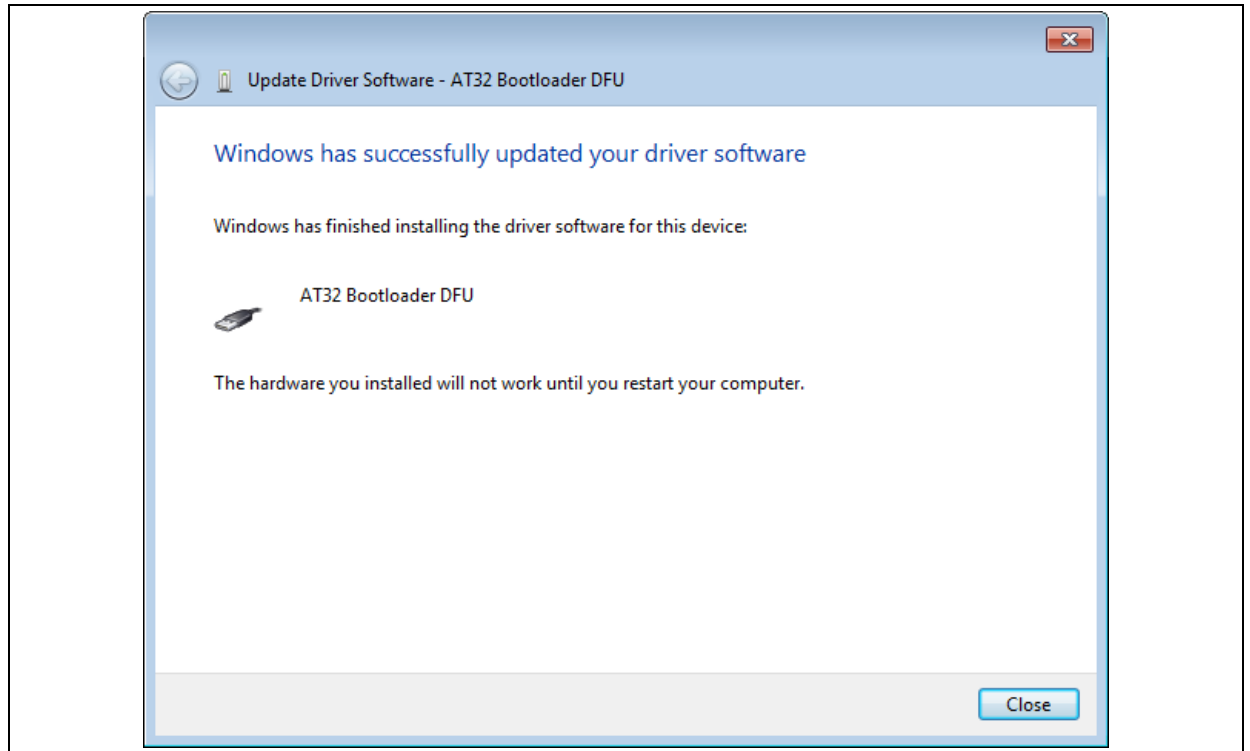
Figure 7. Manual install-driver software installing



Please wait for the driver installation to be complete. After it is completed, click on "**Close**" (As shown in Figure 8).

The manual installation of the driver is now completed.

Figure 8. Manual install successful



4 interfaces

4.1 AT32F403 interfaces

Table 1. AT32F403 GPIO Pin Map

IP	MCUs supported	Pin
USART1	All	PA9: USART1_TX PA10: USART1_RX
USART2	AT32F403ZGT6/AT32F403VGT6	PD5: USART2_TX PD6: USART2_RX
	Others	PA2: USART2_TX PA3: USART2_RX
DFU	All	PA11: OTGFS1_D- PA12: OTGFS1_D+

4.2 AT32F413 interfaces

Table 2. AT32F413 GPIO Pin Map

IP	MCUs supported	Pin
USART1	All	PA9: USART1_TX PA10: USART1_RX
USART2	All	PA2: USART2_TX PA3: USART2_RX
DFU	All	PA11: OTGFS1_D- PA12: OTGFS1_D+

4.3 AT32F415 interfaces

Table 3. AT32F415 GPIO Pin Map

IP	MCUs supported	Pin
USART1	All	PA9: USART1_TX PA10: USART1_RX
USART2	All	PA2: USART2_TX PA3: USART2_RX
DFU	All	PA11: OTGFS1_D- PA12: OTGFS1_D+

4.4 AT32F403A/F407 interfaces

Table 4. AT32F403A/F407GPIO Pin Map

IP	MCUs supported	Pin
USART1	All	PA9: USART1_TX PA10: USART1_RX
USART2	AT32F403AVGT7/AT32F407VGT7	PD5: USART2_TX PD6: USART2_RX
	Others	PA2: USART2_TX PA3: USART2_RX

DFU	All	PA11: OTGFS1_D- PA12: OTGFS1_D+
-----	-----	------------------------------------

4.5 AT32F421 interfaces

Table 5. AT32F421 GPIO Pin Map

IP	MCUs supported	Pin
USART1	All	PA9: USART1_TX PA10: USART1_RX
USART2	All	PA2: USART2_TX PA3: USART2_RX

4.6 AT32F435/F437 interfaces

Table 6. AT32F435/F437 GPIO Pin Map

IP	MCUs supported	Pin
USART1	All	PA9: USART1_TX PA10: USART1_RX
USART2	AT32F435/F437ZxT7、 AT32F435/F437VxT7	PD5: USART2_TX PD6: USART2_RX
	Others	PA2: USART2_TX PA3: USART2_RX
USART3	AT32F435/F437ZxT7、 AT32F435/F437VxT7、 AT32F435/F437RxT7	PC10: USART3_TX PC11: USART3_RX or PB10: USART3_TX PB11: USART3_RX
	Others	PB10: USART3_TX PB11: USART3_RX
DFU1	All	PA11: OTGFS1_D- PA12: OTGFS1_D+
DFU2	All	PB14: OTGFS1_D- PB15: OTGFS1_D+

Note 1: USART3 of AT32F435/ AT32F437ZxT7, AT32F435/ AT32F437VxT7, AT32F435/ AT32F437RxT7 supports PB10 and PB11 only in version B.

4.7 AT32WB415 interfaces

Table 7. AT32WB415 GPIO Pin Map

IP	MCUs supported	Pin
USART1	All	Not supported
USART2	All	PA2: USART2_TX PA3: USART2_RX
DFU	All	PA11: OTGFS1_D- PA12: OTGFS1_D+

4.8 AT32F425 interfaces

Table 8. AT32F425 GPIO Pin Map

IP	MCUs supported	Pin
USART1	All	PA9: USART1_TX PA10: USART1_RX
USART2	All	PA2: USART2_TX PA3: USART2_RX

4.9 AT32L021 interfaces

Table 9. AT32L021 GPIO Pin Map

IP	MCUs supported	Pin
USART1	All	PA9: USART1_TX PA10: USART1_RX
USART2	All	PA2: USART2_TX PA3: USART2_RX

4.10 AT32F423 interfaces

Table 10. AT32F423 GPIO Pin Map

IP	MCUs supported	Pin
USART1	All	PA9: USART1_TX PA10: USART1_RX
USART2	All	PA2: USART2_TX PA3: USART2_RX
USART3	AT32F423Vxx/AT32F423Rxx	PC10: USART3_TX PC11: USART3_RX
	Others	PB10: USART3_TX PB11: USART3_RX
DFU	All	PA11: OTGFS1_D- PA12: OTGFS1_D+

4.11 AT32A403A interfaces

Table 11. AT32A403A GPIO Pin Map

IP	MCUs supported	Pin
USART1	All	PA9: USART1_TX PA10: USART1_RX
USART2	AT32A403AVGT7	PD5: USART2_TX PD6: USART2_RX
	Others	PA2: USART2_TX PA3: USART2_RX
DFU	All	PA11: OTGFS1_D- PA12: OTGFS1_D+

4.12 AT32F402/F405 interfaces

Table 12. AT32F402/F405 GPIO Pin Map

IP	MCUs supported	Pin
USART1	AT32F405KxU7-4	Not supported
	Others	PA9: USART1_TX PA10: USART1_RX
USART2	All	PA2: USART2_TX PA3: USART2_RX
USART3	AT32F405RxT7, AT32F405RxT7-7	PC10: USART3_TX PC11: USART3_RX
	AT32F402RxT7, AT32F402RxT7-7	PC10: USART3_TX PC11: USART3_RX or PB10: USART3_TX PB11: USART3_RX
	AT32F402CxT7, AT32F402CxU7	PB10: USART3_TX PB11: USART3_RX
	Others	Not supported
DFU	All	PA11: OTGFS1_D- PA12: OTGFS1_D+
I ² C1	All	PB6: I2C1_SCL PB7: I2C1_SDA
I ² C2	AT32F405KxU7-4, AT32F402KxU7-4	Not supported
	Others	PB10: I2C2_SCL PB3: I2C2_SDA
I ² C3	AT32F405KxU7-4	Not supported
	Others	PA8: I2C3_SCL PB4: I2C3_SDA
CAN1	All	PB8: CAN1_RX PB9: CAN1_TX
SPI1	All	PA4: SPI1_CS PA5: SPI1_SCK PA6: SPI1_MISO PA7: SPI1_MOSI

4.13 AT32A423 interfaces

Table 13. AT32A423 GPIO Pin Map

IP	MCUs supported	Pin
USART1	All	PA9: USART1_TX PA10: USART1_RX
USART2	All	PA2: USART2_TX PA3: USART2_RX

USART3	AT32A423Vxx/AT32A423Rxx	PC10: USART3_TX PC11: USART3_RX
	Others	PB10: USART3_TX PB11: USART3_RX
DFU	All	PA11: OTGFS1_D- PA12: OTGFS1_D+

4.14 AT32M412/M416 interfaces

Table 14. AT32M412/M416 GPIO Pin Map

IP	MCUs supported	Pin
USART1	All	PA9: USART1_TX PA10: USART1_RX
USART2	All	PA2: USART2_TX PA3: USART2_RX
DFU	All	PA11: OTGFS1_D- PA12: OTGFS1_D+
I2C1	All	PB6: I2C1_SCL PB7: I2C1_SDA
I2C2	AT32M412ExP7/AT32M416ExP7	Not supported
	Others	PB10: I2C2_SCL PB3: I2C2_SDA
CAN1	AT32M412KxT7, AT32M412KxU7, AT32M416KxT7, AT32M412KxU7	Not supported
	Others	PB5: CAN1_RX PB13: CAN1_TX
SPI1	All	PA4: SPI1_CS PA5: SPI1_SCK PA6: SPI1_MISO PA7: SPI1_MOSI

5 User Interface

5.1 Connection settings

On this page, you can select the corresponding connection mode, that is, the interface type: UART or DFU.

5.1.1 UART connection

After using the UART connection, you can select the serial interface to be operated and make related settings (As shown in Figure 9). Please ensure that the device to be operated is properly connected to the selected serial interface.

When "**Boot Switch**" is set to "**Manual**", you need to manually reset the device to restart the "**BootLoader**" program in device. If the device supports automatic connection circuitry, reset can be controlled by controlling the DTR and RTS signals. You can select the control mode of the current device in "**Boot Option**".

After setting, click on "**Next**", if the connection is successful, skip to the next page. If failed, an error message will be displayed.

Figure 9. UART connection window

ISP Artery ISP Programmer_V2.0.08

ARTERY 雅特力

Select the communication port and set settings, then click next to open connection

Port Type: UART

Port Name: COM52 Parity: Even

Baud Rate: 115200 Boot Switch: Manual

Data Bits: 8 Timeout(s): 2

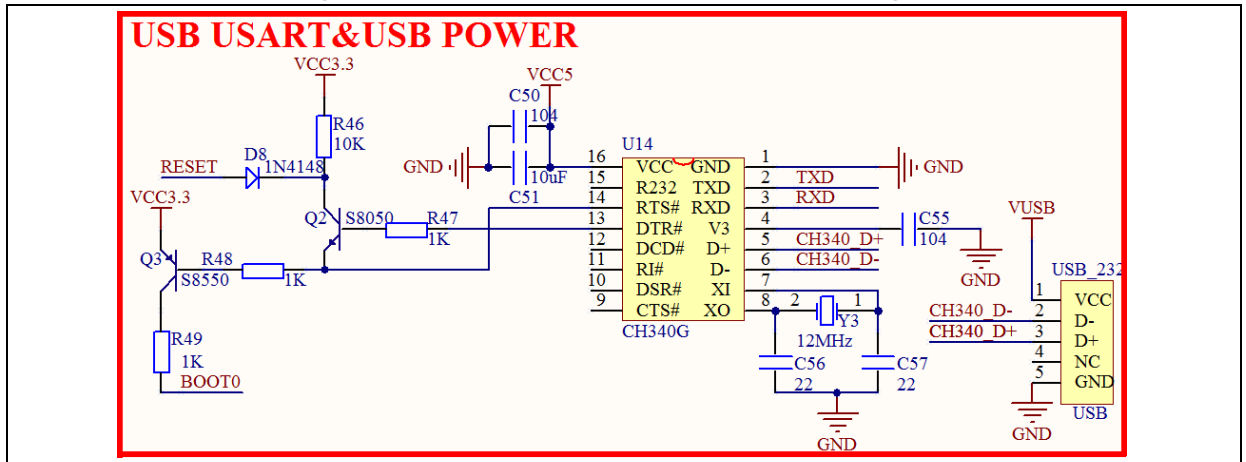
Boot Option: Not use RTS and DTR

Select Language: English

Back Next Cancel Close

The USB serial interface automatic connection circuit can be designed with reference to the following figure. (As shown in Figure 10):

Figure 10. USB interface auto connection diagram



The combination of Q2 and Q3 in figure 10 constitutes the automatic connection circuit of the development board, which only should be set in the ISP software: DTR low level reset, RTS high level to load bootloader. In this case, it can be connected automatically without setting B0 manually and pressing reset button. Among them, RESET is the reset signal of the board, whereas BOOT0 is B0 signal of the boot mode.

The following is the implementation process of automatic connection circuit when BOOT1 is low: First, the ISP controls DTR to output low level, then DTR_N output high, and RTS is set high, then RTS_N output is low, so Q3 is turned on and BOOT0 is pulled up, that is, BOOT0 is set to 1, and Q2 will also be turned on at the same time, the reset pin of chip is pulled low to realize reset. Then, after a delay of 100ms, the ISP controls DTR to be high level, then DTR_N output low level, and RTS maintains high, then RTS_N continues to be at low level, in this case, the reset pin of chip becomes high since Q2 is no longer on, and the chip ends reset, but BOOT0 remains at 1, and enters the BootLoader Mode, and then ISP starts to connect and download the code.

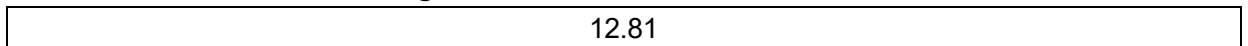
5.1.2 DFU connection

After selecting the DFU connection, you can select the DFU device to be connected (as shown in Figure 11). Please ensure that the device to be operated is connected to the corresponding USB port of PC.

The software will automatically obtain and display the relevant information of DFU device, including vendor ID (VID), product ID (PID), and product SN (UID).

After selecting the DFU device to be connected, click on "**Next**", and if the connection is successful, skip to the next page. If failed, an error message will be displayed.

Figure 11. DFU connection window





5.2 Flash status page

The connection is now set up, and the status of Flash is displayed on this page (As shown in Figure 12).

If "**Access protection**" is enabled, the device will restrict the use of some functions, that is, it is only allowed to use Firmware CRC function /Flash CRC/ Disable access protection function.

Figure 12. Flash status window



5.3 Device Information page

This page displays device-related information such as target device, PID, BID, protocol version, Flash mapping and Flash protection status (As shown in Figure 13).

If SPIM is connected, please check "**SPIM**" and select "**SPIM Type**". The SPIM size depends on the "SPIM Type". If SPIM encryption is required, set the SPIM FLASH_DA.

In this case, all sectors of main flash and SPIM are automatically displayed in the Flash map.

Figure 13. Device information

The screenshot shows the 'Artery ISP Programmer_V2.0.08' window. At the top is the Artery logo and the text '雅特力'. Below this, it says 'Please, select your device in the target list'. The 'Target' dropdown menu is set to 'AT32F403AVGT7_1024K'. Below the target selection, there are three input fields: 'PID (h)' with value '70050344', 'BID (h)' with value '4703', and 'Protocol Version' with value '3.2'. There is a checked checkbox for 'SPIM'. Below this, there are three input fields: 'SPIM Type' with value 'Common Type2', '16MB', and a 'Select' button. Below these is an input field for 'SPIM FLASH_DA 0x' with value '0'. A section titled 'Flash mapping' contains a table with 6 columns: Name, Start address, End address, Size, FAP, and EPP. The table lists 10 sectors (Sector0 to Sector9), each with a 2K size. The FAP and EPP columns are highlighted in green and contain 'N' for all sectors. Below the table, there is a legend: 'Y: Protected' and 'N: UnProtected'. At the bottom of the window are four buttons: 'Back', 'Next', 'Cancel', and 'Close'.

Name	Start address	End address	Size	FAP	EPP
Sector0	0x08000000	0x080007FF	0x800 (2K)	N	N
Sector1	0x08000800	0x08000FFF	0x800 (2K)	N	N
Sector2	0x08001000	0x080017FF	0x800 (2K)	N	N
Sector3	0x08001800	0x08001FFF	0x800 (2K)	N	N
Sector4	0x08002000	0x080027FF	0x800 (2K)	N	N
Sector5	0x08002800	0x08002FFF	0x800 (2K)	N	N
Sector6	0x08003000	0x080037FF	0x800 (2K)	N	N
Sector7	0x08003800	0x08003FFF	0x800 (2K)	N	N
Sector8	0x08004000	0x080047FF	0x800 (2K)	N	N
Sector9	0x08004800	0x08004FFF	0x800 (2K)	N	N

Y: Protected N: UnProtected

In UART communication mode:

1. AT32F403 series MCUs support SPIM.
2. AT32F413 series MCUs support SPIM.
3. AT32F415 series MCUs do not support SPIM.
4. AT32F403A series MCUs support SPIM.
5. AT32F407 series MCUs support SPIM.
6. AT32F421 series MCUs do not support SPIM.
7. AT32F435 series MCUs do not support SPIM.
8. AT32F437 series MCUs do not support SPIM.
9. AT32F425 series MCUs do not support SPIM.
10. AT32L021 series MCUs do not support SPIM.
11. AT32F423 series MCUs do not support SPIM.

12. AT32F402 series MCUs do not support SPIM.
13. AT32F405 series MCUs do not support SPIM.
14. AT32A403A series MCUs support SPIM.
15. AT32A423 series MCUs do not support SPIM.
16. AT32M412 series MCUs do not support SPIM.
17. AT32M416 series MCUs do not support SPIM.

In DFU communication mode:

1. AT32F403 series MCUs do not support SPIM.
2. AT32F413KCU7-4 and AT32F413KBU7-4 in the AT32F413 series do not support SPIM; other models of AT32F413 series MCUs support SPIM.
3. AT32F415 series MCUs do not support SPIM.
4. AT32F403A series MCUs support SPIM.
5. AT32F407 series MCUs support SPIM.
6. AT32F421 series MCUs do not support DFU and SPIM.
7. AT32F435 series MCUs do not support SPIM.
8. AT32F437 series MCUs do not support SPIM.
9. AT32F425 series MCUs do not support DFU and SPIM.
10. AT32L021 series MCUs do not support DFU and SPIM.
11. AT32F423 series MCUs do not support SPIM.
12. AT32F402 series MCUs do not support SPIM.
13. AT32F405 series MCUs do not support SPIM.
14. AT32A403A series MCUs do not support SPIM.
15. AT32A423 series MCUs do not support SPIM.
16. AT32M412 series MCUs do not support SPIM.
17. AT32M416 series MCUs do not support SPIM.

■ Checked “**SPIM**”

Allows operation on SPIM.

■ Unchecked “**SPIM**”

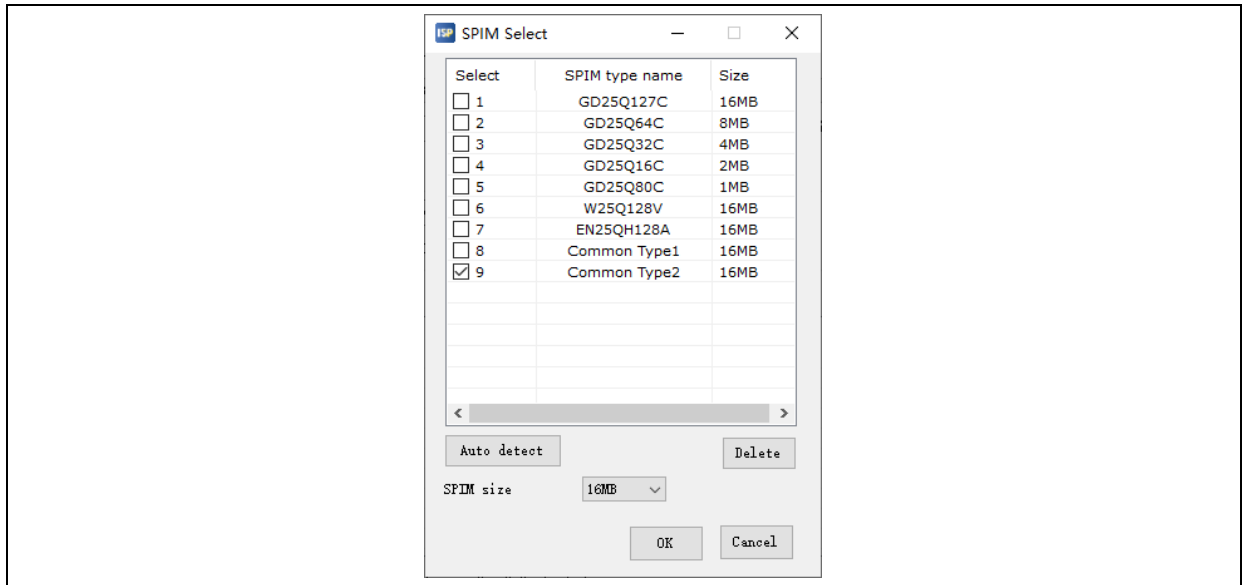
Operation on SPIM is not allowed.

■ SPIM Type

You can select SPIM type with “**Select**” button.

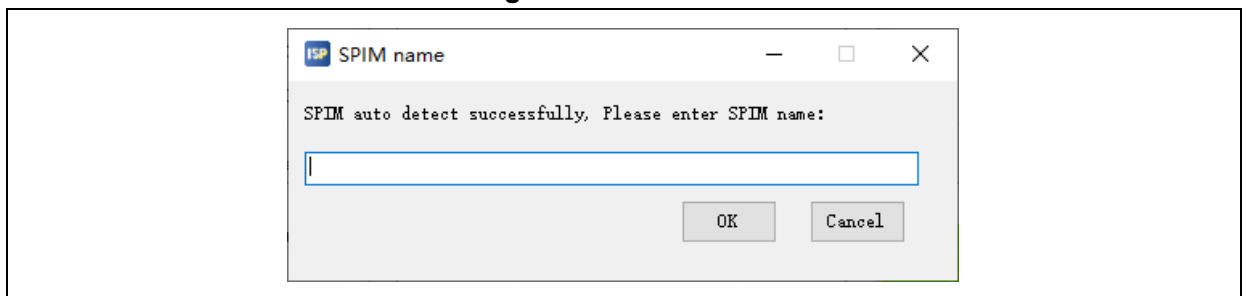
Click on “**Select**” button, a dialog box will pop up. (As shown in Figure 14)

Figure 14. SPIM selection



Auto detect: it will automatically detect whether the SPIM meets the requirements of this software operation. (Auto Detect will overwrite some data of SPIM, please use it with caution)
If the detection is successful, a dialog box will pop up. (As shown in Figure 15)

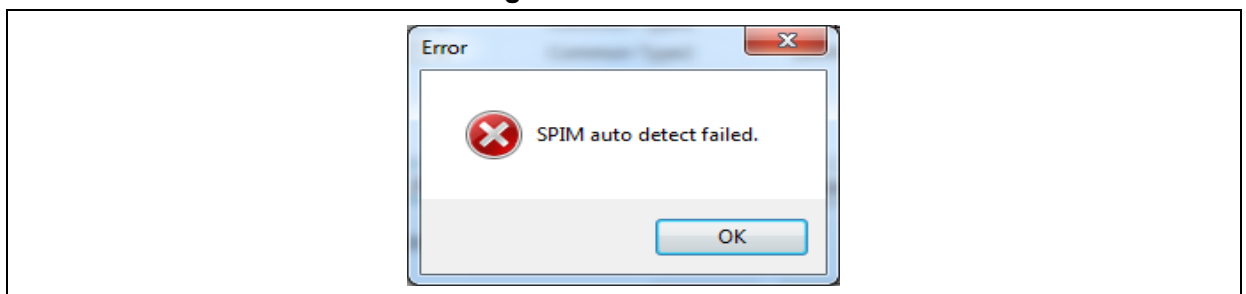
Figure 15. SPIM name



Click on "**OK**" to add the detected SPIM to the SPIM list.
Click on "**Cancel**" to cancel auto detect.

If Auto detect failed, a failure dialog box will pop up. (As shown in Figure 16)

Figure 16. SPIM name



SPIM size: this is used to select SPIM size, except for the default type.
Delete: delete the selected SPIM from the list, except for the default type.
OK: SPIM selected.
Cancel: cancel.

- SPIM Size

SPIM size is depending on the selected SPIM type.

- SPIM FLASH_DA

Set the encryption range when downloading files to the SPIM. The encryption range calculates starting from address 0x08400000.

- Remap0 (use PA11/PA12 pins)

- Remap1 (use PB10/PB11 pins)

Select the desired pins. This option is only available for AT32F413/F403/F407 series UART interfaces.

5.4 Operation configuration page

Choose what you need to do on this page. (As shown in Figure 17)

Figure 17. Operation configuration

ISP Artery ISP Programmer_V2.0.08

ARTERY 雅特力

☐ Erase ☒ All ☐ Sectors ... ☐ Edit User system data

☒ Download to device ☐ Disable sLib

sLib Status: DISABLE Start sector

Remaining usage times: 256 DATA start sector

Password 0x End sector

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

Add Delete

Erase option Erase the sectors of file size ☐ Enable sLib before download

☐ Optimize(Remove some FFs) ☐ Verify after download

☐ Write user serial number ☐ Jump to the user program

Address 0x 08010000 Current SN 0x 00000001 Increase step 0x 00000001

☐ Apply User system data ...

☐ Enable Access protection after Download

☐ Upload from device ...

☐ Firmware CRC Sector fill FF

☐ Flash CRC

Start sector Sector0-0x8000000 End sector Sector0-0x8000000

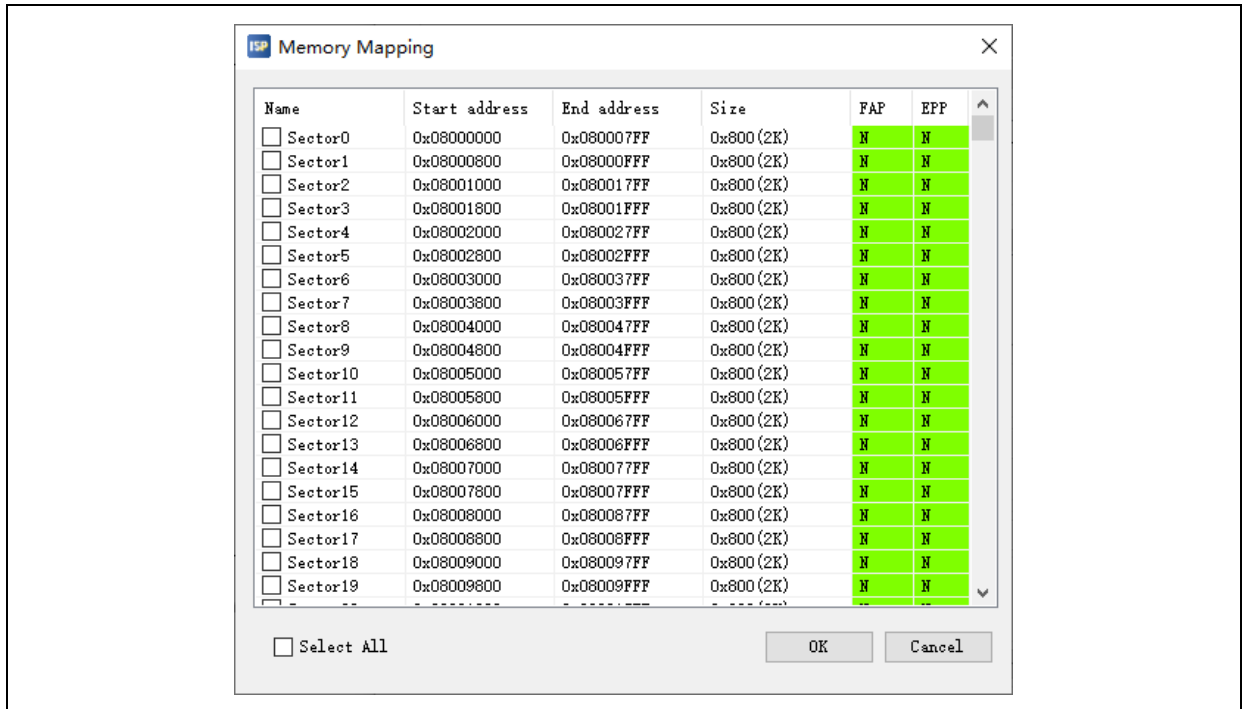
☐ Protection DISABLE Access protection ...

Back Next Cancel Close

5.4.1 Erase

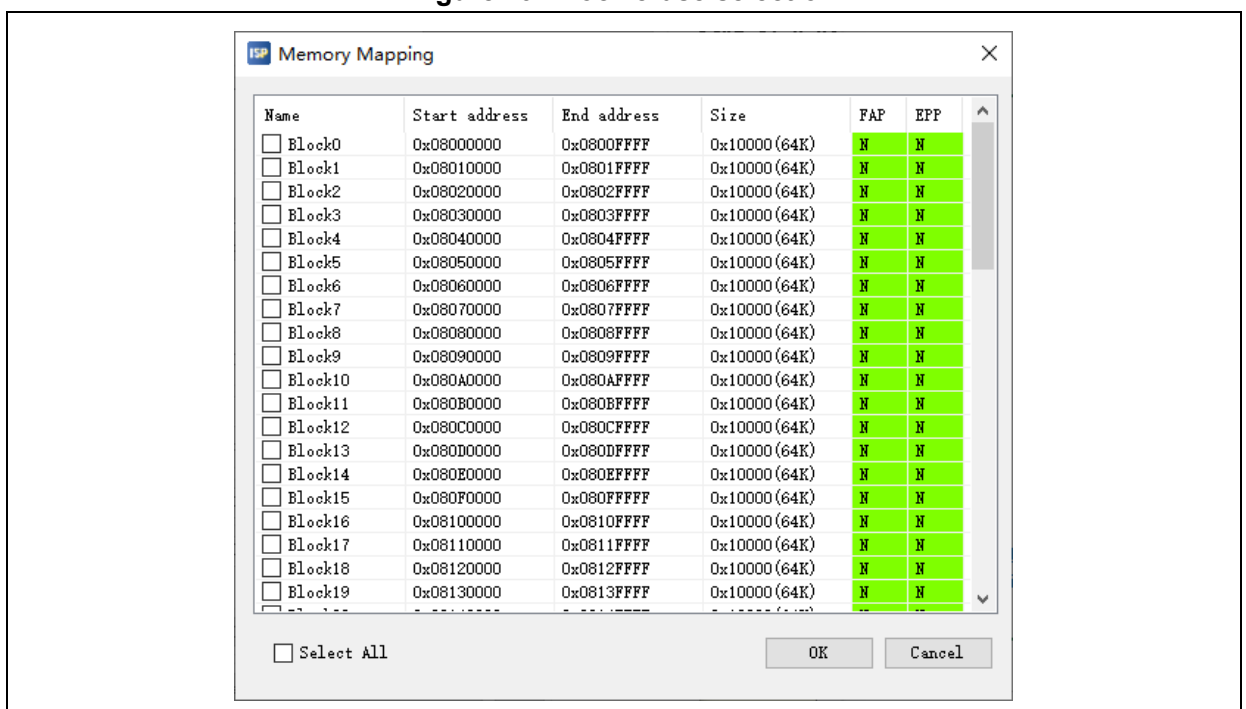
- Click on "**All**" to erase the whole memory (Including SPIM).
- Click on "**Sectors**" to customize the sectors to be erased. At this time, click on "..." to select the sector to be erased in the pop-up dialog box. (As shown in Figure 18)

Figure 18. Sector erase selection



- Click on "**Blocks**" to customize the blocks to be erased. At this time, click on "..." to select the block to be erased in the pop-up dialog box. (As shown in Figure 19)

Figure 19. Block erase selection



5.4.2 Edit User system data

Select "**Edit User system data**" and click on "**Next**".

On this page, users can configure the "User system data" through graphical interface (As shown in Figure 20).

Supports obtaining the "User system data" value from the device or file and displaying the value. After editing, apply to device or save to file.

Figure 20. User system data

The screenshot shows the 'Artery ISP Programmer_V2.0.08' window. The 'Access protection' section has 'FAP' set to 'A5' and 'Disable'. The 'EOPBO (SRAM)' section is set to '96KB SRAM'. The 'System setting byte' section has 'SSB' set to 'FF' and several checkboxes checked: 'nWDT_ATO_EN', 'nDEPSLP_RST', 'nSTDBY_RST', and 'BTOPT'. The 'Erase and program protection bytes' section shows a table of sectors with their start and end addresses, sizes, and EPP values. The 'User data' section shows a table of data bytes (0-7) all set to 'FF'. The 'SPI encryption key' section shows eight keys (KEY0-KEY7) all set to 'FF'. At the bottom, there are buttons for 'Load from device', 'Apply to device', 'Load from file', 'Save to file', 'Back', 'Next', 'Cancel', and 'Close'.

Name	Start a...	End add...	Size	EPP
<input type="checkbox"/> Sector0	0x08000000	0x080007FF	0x800 (2K)	N
<input type="checkbox"/> Sector1	0x08000800	0x08000FFF	0x800 (2K)	N
<input type="checkbox"/> Sector2	0x08001000	0x080017FF	0x800 (2K)	N
<input type="checkbox"/> Sector3	0x08001800	0x08001FFF	0x800 (2K)	N
<input type="checkbox"/> Sector4	0x08002000	0x080027FF	0x800 (2K)	N
<input type="checkbox"/> Sector5	0x08002800	0x08002FFF	0x800 (2K)	N
<input type="checkbox"/> Sector6	0x08003000	0x080037FF	0x800 (2K)	N
<input type="checkbox"/> Sector7	0x08003800	0x08003FFF	0x800 (2K)	N

Date	0	1	2	3	4	5	6	7
Data 0---7 (0x)	FF	FF	FF	FF	FF	FF	FF	FF

KEY	Value
KEY0	0x FF
KEY1	0x FF
KEY2	0x FF
KEY3	0x FF
KEY4	0x FF
KEY5	0x FF
KEY6	0x FF
KEY7	0x FF

■ Access protection

The access protection status is displayed. The access protection of the memory cannot be set here.

AT32F403/F413/F403A/F407/F435/F437/A403A:

Enabled: FAP---0xFF.

Disabled: FAP---0xA5.

AT32F415/F421/F425/L021/F423/A423/F402/F405/F490/M412/M416:

Access protection: FAP---0xFF.

High level access protection: FAP---0xCC (Access protection and user system data erase

protection). (AT32F425/L021/F423/A423/F402/F405/F490/M412/M416 high level access protection is irreversible. Once enabled, it will never be unlocked, with its debugging interface permanently disabled. Please use with caution.)

Disabled: FAP----0xA5.

When access protection is enabled, neither the flash memory or user system data can be read, unless the access protection is disabled. After access protection is disabled, both the main flash and user system data will be erased.

■ System setting byte

nWDT_ATO_EN:

Unchecked—Hardware watchdog.

Checked—Software watchdog.

nDEPSLP_RST:

Unchecked—Reset occurs when entering Deep Sleep mode.

Checked—No reset occurs when entering Deep Sleep mode.

nSTDBY_RST:

Unchecked—Reset occurs when entering Standby mode.

Checked—No reset occurs when entering Standby mode.

BTOPT (AT32F403/F413/F403A/F407/F435/F437/A403A)

Unchecked—when the device is set to boot from flash memory bank 1 or bank 2, if bank 2 has no startup program, boots from bank 1, otherwise, bank 2.

Checked—when the device is set to boot from flash memory (default value), it starts from bank 1.

nBOOT1 (AT32F421/F425/L021/F423/A423/F402/F405/F490/M412/M416)

Boot mode is determined together with BOOT0, and when BOOT0 = 1,

Unchecked----SRAM is selected as boot space.

Checked---Boot memory is selected as boot space.

nWDT_DEPSLP:

Unchecked----WDT stop count when entering Deep Sleep mode.

Checked---WDT does not stop count when entering Deep Sleep mode.

nWDT_STDBY:

Unchecked---- WDT stop count when entering Standby mode.

Checked--- WDT does not stop count when entering Standby mode.

SRAM_Parity: (AT32L021)

Unchecked ----- Enable odd check of RAM.

Checked ----- Disable odd check of RAM.

■ EOPB0(SRAM)

AT32F403/F403A/F407/A403A: (AT32F403CBT6 not support)

224 KB SRAM—SRAM 224 KB.

96 KB SRAM—SRAM 96 KB.

AT32F413: (AT32F413C8T7/AT32FEBKC8T7 not support)

64 KB SRAM—SRAM 64 KB.

32 KB SRAM—SRAM 32 KB.

16 KB SRAM—SRAM 16 KB.

AT32F415/F421/F425/L021/F423/A423/F402/F405/F490/M412/M416: (not support)

AT32F435/F437:

Flash size 256K and below:

512 KB SRAM—SRAM 512 KB.

448 KB SRAM—SRAM 448 KB.

384 KB SRAM—SRAM 384 KB.

Flash size 1024K and above:

512 KB SRAM—SRAM 512 KB.

448 KB SRAM—SRAM 448 KB.

384 KB SRAM—SRAM 384 KB.

320 KB SRAM—SRAM 320 KB.

256 KB SRAM—SRAM 256 KB.

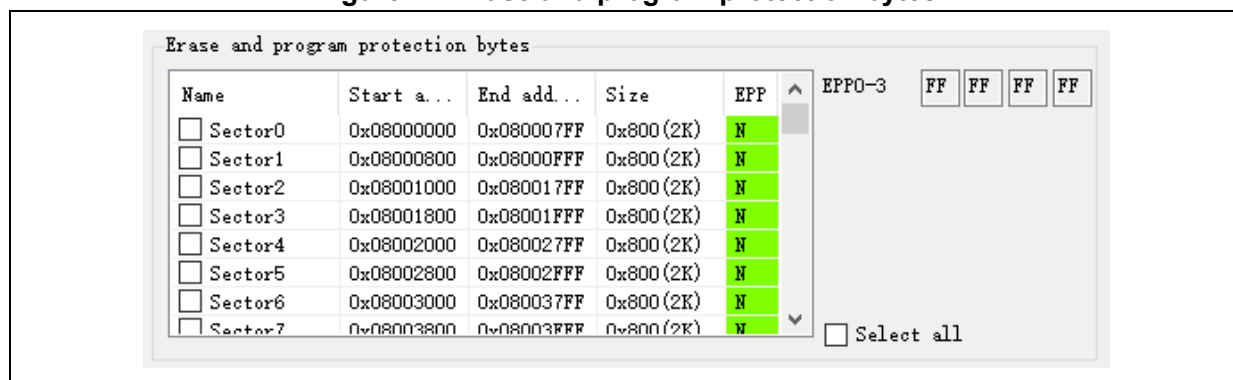
192 KB SRAM—SRAM 192 KB.

128 KB SRAM—SRAM 128 KB.

■ Erase and program protection bytes

You can choose which sectors need to be erase and program protected. (As shown in Figure 21)

Figure 21. Erase and program protection bytes



EPP0:

AT32F403/F413/F403A/F407/A403A: controls the erase and program protection of sectors in the range of Flash 1K-32K.

AT32F415/F423/A423/F402/F405: controls the erase and program protection of Sector0-Sector15.

AT32F421: controls the erase and program protection of Sector0-Sector31.

AT32F435/F437: controls the erase and program protection of sectors in the range of Flash 1K-32K. Each bit protects 4K bytes sectors.

AT32F425: controls the erase and program protection of Sector0-Sector31.

AT32L021: controls the erase and program protection of Sector0-Sector31.

AT32M412/M416: controls the erase and program protection of Sector0-Sector31.

EPP1:

AT32F403/F413/F403A/F407/A403A: controls the erase and program protection of sectors in the range of Flash 33K-64K.

AT32F415/F423/A423/F402/F405: controls the erase and program protection of Sector16-Sector31.

AT32F421: controls the erase and program protection of Sector32-Sector63.

AT32F435/F437: controls the erase and program protection of sectors in the range of Flash 33K-64K. Each bit protects 4K bytes sectors.

AT32F425: controls the erase and program protection of Sector32-Sector63.

AT32L021: controls the erase and program protection of Sector32-Sector63.

AT32M412/M416: controls the erase and program protection of Sector21-Sector63.

EPP2:

AT32F403/F413/F403A/F407/A403A: controls the erase and program protection of sectors in the range of Flash 65K-96K.

AT32F415/F423/A423/F402/F405: controls the erase and program protection of Sector32-Sector47.

AT32F435/F437: controls the erase and program protection of sectors in the range of Flash 65K-96K. Each bit protects 4K bytes sectors.

AT32M412/M416: controls the erase and program protection of Sector64-Sector95.

EPP3:

AT32F403/F413/F403A/F407/A403A:

Bit 0-6 controls the erase and program protection of sectors in the range of 97K-124K;

Bit 7 controls the erase and program protection of all Sectors after Flash 124K, including SPIM.

AT32F415/F423/A423/F402/F405:

Bits 0-6 control the erase and program protection of Sector48-Sector61;

Bit 7 controls the erase and program protection of all subsequent sectors, including boot memory (boot memory in AP mode).

AT32F421: Bit 7 controls the boot memory area (boot memory in AP mode)

AT32F435/F437: controls the erase and program protection of sectors in the range of Flash 97K-128K. Each bit protects 4K bytes sectors.

AT32F425: Bit 7 controls the boot memory area (boot memory in AP mode)

AT32L021: Bit 7 controls the boot memory area (boot memory in AP mode)

AT32M412/M416: controls the erase and program protection of Sector96-Sector127.

EPP4:

AT32F435/F437: controls the erase and program protection of sectors in the range of Flash 129K-1152K. Each bit protects 128K bytes sectors.

EPP5:

AT32F435/F437: controls the erase and program protection of sectors in the range of Flash 1153K-2176K. Each bit protects 128K bytes sectors.

EPP6:

AT32F435/F437: controls the erase and program protection of sectors in the range of Flash 2177K-3200K. Each bit protects 128K bytes sectors.

EPP7:

AT32F435/F437: Bit 0-6 controls the erase and program protection of sectors in the range of Flash 3201K-4032K. Each bit protects 128K bytes sectors.

■ User data

Figure 22. User data

Date	0	1	2	3	4	5	6	7
Data 0---7 (0x)	11	22	FF	FF	FF	FF	FF	FF
Data 8---9 (0x)	FF	FF						

Buttons: Clear, Load file, Save to file

AT32F403/F413/F403A/F407/A403A: user data 8 bytes.

AT32F415: user data 10 bytes.

AT32F421: user data 250 bytes.

AT32F435/F437: Flash size is less than 4032K, user data 220 bytes. Flash size 4032K, user data 2012 bytes.

AT32L021: user data 250 bytes.

AT32F425: user data 250 bytes.

AT32F423: user data 250 bytes.

AT32F402/F405: user data 220 bytes.

AT32A423: user data 250 bytes.

AT32M412/M416: user data 250 bytes.

Clear: Reset all user system data to 0xFF, which is not saved to the device

Load file: Load the user system data file into the table for display

Save to file: Save the user system data in the table to the file.

■ SPIM encryption key (AT32F403/F413/F403A/F407/A403A)

You can set the encryption key when downloading the SPIM. (As shown in Figure 23)

Figure 23. SPIM encryption key

SPIM encryption key

KEY0 0x	FF	KEY1 0x	FF	KEY2 0x	FF	KEY3 0x	FF
KEY4 0x	FF	KEY5 0x	FF	KEY6 0x	FF	KEY7 0x	FF

■ QSPI encryption key (AT32F435/F437/F402/F405)

You can set the encryption key when downloading the QSPI. (As shown in Figure 24)

Figure 24. QSPI encryption key

The screenshot shows a window titled "QSPI encryption key". Inside, there are two rows of four keys each. Each key is labeled (KEY0 to KEY7), followed by "0x", and then a text box containing "FF".

KEY0	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6	KEY7
0x FF	0x FF	0x FF	0x FF	0x FF	0x FF	0x FF	0x FF

■ Load from device

Read the user system data from the device and update it to the interface for display.

■ Apply to device

Save the settings of the user system data to the device.

■ Load from file

Read the content of user system data from user system data and update it to the interface for display.

■ Save to file

Save the user system data settings to a file.

5.4.3 Download to device

(As show in Figure 25)

Figure 25. Download to device

● sLib settings

(AT32F403 not support sLib)

— sLib status

The sLib status of the current connected chip, disabled or enabled.

— Remaining usage times (AT32F413/F403A/F407/A403A)

It means the remaining number of times of sLib. It can be used up to 256 times, and will be reduced after each use. When the remaining number of times is 0, the sLib function will not be available.

— Password

Enter the enable password when the sLib function is enabled. Enter the disable password when the sLib function is disabled.

— Start sector

AT32F413/F415/F403A/F407/A403A:

The start sector of sLib area. The instruction area is from the "Start sector" to the "DATA start sector"(not including The DATA start sector). When sLib is enabled, the data in this area cannot be erased, written or read.

AT32F421/F435/F437/F425/L021/F423/A423/F402/F405/F490/M412/M416:

The start sector of sLib area. The area from "Start sector" to "INSTR start sector" (not including

"INSTR start sector") is a mixed instruction and data (read only area). Once sLib is enabled, the data in this area cannot be erased, written, but can be read.

— DATA start Sector/INSTR start Sector

AT32F413/F415/F403A/F407/A403A:

The start sector of the sLib data area. This data area is from "DATA start sector" to "End sector"(including "End sector"). After sLib is enabled, the data in this area cannot be erased and written, but can be read. When set to "none", it is set to no data area.

AT32F421/F435/F437/F425/L021/F423/A423/F402/F405/F490/M412/M416:

The start sector of sLib instruction area. The instruction area is from "INSTR start sector" to "End sector" (including "End sector"). After sLib is enabled, the data in this area cannot be erased, written or read. When it is set to "none", it is no instruction area.

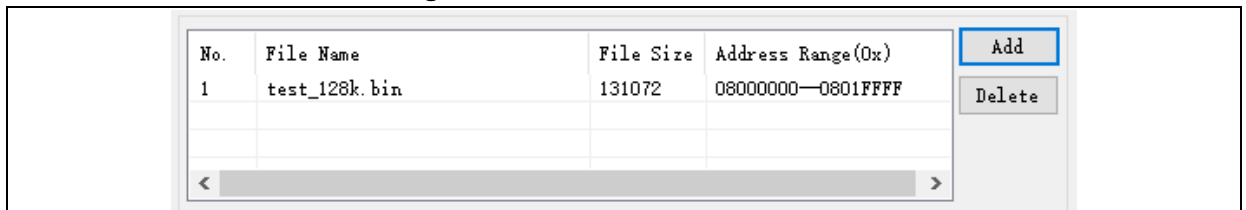
— End Sector

The end position of the sLib area.

● Other download settings

Three file types are supported: bin (binary), hex (hexadecimal), and s19 / srec (Motorola S file). (As shown in Figure 26)

Figure 26. Download file selection



If you are adding a bin file, you need to choose a download address.

If you are adding a hex or S19 / SREC file, the download address is obtained from the loaded file.

- Check "**Erase the sectors of file size**" to erase sectors where the downloaded file is located before download.
- Check "**No Erase**", no erase operation will be performed before download.
- Check "**Global Erase**" to erase the whole memory (including SPIM) before download.
- Check "**Jump to the user program**" to run the program directly after the download is complete.
- Check "**Enable sLib before download**" to enable sLib before download. You need to enter the password, start sector, DATA/INSTR start sector, and end sector to enable sLib.
- Check "**Verify after download**" to run the verify program after downloading to verify whether the downloaded data is correct.
- Check "**Optimize (Remove some FFs)**" to optimize the download process, skip the 0xFF field of the file and speed up the download.

- Check "**Write user serial number**" and download the serial number to the device after download.

Address: the address where the serial number is programmed into the memory.

Current SN: the serial number of the current programming.

Increase step: this is the amount added to the next serial number after each serial number is programmed

- Check "**Apply User system data**", load the user system data file after download, and set the value to the device.

- Check "**Enable Access Protection after Download**" to enable access protection after download.

For AT32F415/F421/F425/L021/F423/A423/F402/F405/F490/M412/M416, you can enable access protection and high level access protection (Access protection and user system data erase protection). (AT32F425/L021/F423/A423/F402/F405/F490/M412/M416 high level access protection is irreversible. Once enabled, it will never be unlocked, with its debugging interface permanently disabled. Please use with caution.)

5.4.4 Disable sLib

To disable sLib, enter the disable password. (That is, enter the password when sLib was last enabled) (As shown in Figure 27):

Figure 27. Disable sLib

☐ Download to device
 ☒ **Disable sLib**

sLib Status: **ENABLE**

Password 0x **55555555**

Start sector: Sector0—0x8000000
 INSTR start sector: Sector10—0x800A000
 End sector: Sector19—0x8013000

When disabling sLib successfully, the whole chip will be erased.

5.4.5 Upload from device

Three file types are supported: bin (binary), hex (hexadecimal), and s19 / srec (Motorola S file). Select the upload sectors. (As shown in Figure 28)

Figure 28. Upload from device

ISP Memory Mapping

Name	Start address	End address	Size	FAP	EPP
<input type="checkbox"/> Sector0	0x08000000	0x080007FF	0x800 (2K)	N	N
<input type="checkbox"/> Sector1	0x08000800	0x08000FFF	0x800 (2K)	N	N
<input type="checkbox"/> Sector2	0x08001000	0x080017FF	0x800 (2K)	N	N
<input type="checkbox"/> Sector3	0x08001800	0x08001FFF	0x800 (2K)	N	N
<input type="checkbox"/> Sector4	0x08002000	0x080027FF	0x800 (2K)	N	N
<input type="checkbox"/> Sector5	0x08002800	0x08002FFF	0x800 (2K)	N	N
<input type="checkbox"/> Sector6	0x08003000	0x080037FF	0x800 (2K)	N	N
<input type="checkbox"/> Sector7	0x08003800	0x08003FFF	0x800 (2K)	N	N
<input type="checkbox"/> Sector8	0x08004000	0x080047FF	0x800 (2K)	N	N
<input type="checkbox"/> Sector9	0x08004800	0x08004FFF	0x800 (2K)	N	N
<input type="checkbox"/> Sector10	0x08005000	0x080057FF	0x800 (2K)	N	N
<input type="checkbox"/> Sector11	0x08005800	0x08005FFF	0x800 (2K)	N	N
<input type="checkbox"/> Sector12	0x08006000	0x080067FF	0x800 (2K)	N	N
<input type="checkbox"/> Sector13	0x08006800	0x08006FFF	0x800 (2K)	N	N
<input type="checkbox"/> Sector14	0x08007000	0x080077FF	0x800 (2K)	N	N
<input type="checkbox"/> Sector15	0x08007800	0x08007FFF	0x800 (2K)	N	N
<input type="checkbox"/> Sector16	0x08008000	0x080087FF	0x800 (2K)	N	N
<input type="checkbox"/> Sector17	0x08008800	0x08008FFF	0x800 (2K)	N	N
<input type="checkbox"/> Sector18	0x08009000	0x080097FF	0x800 (2K)	N	N
<input type="checkbox"/> Sector19	0x08009800	0x08009FFF	0x800 (2K)	N	N

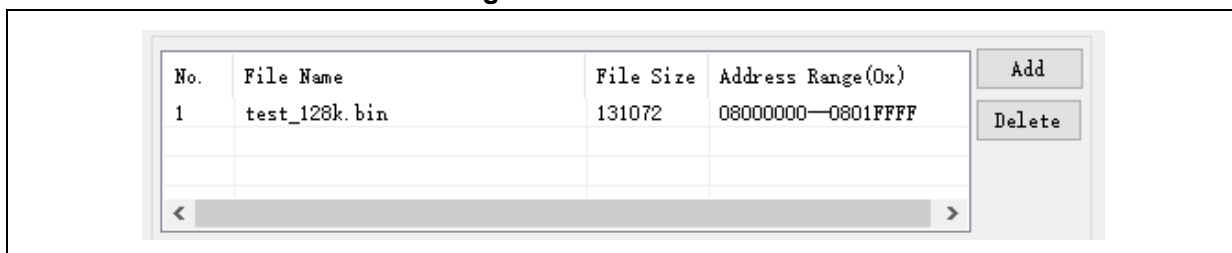
☒ Select All

5.4.6 Firmware CRC

This function is used to calculate the CRC code and compare it with the imported file to confirm the correctness of the downloaded files (This function can be used in the Flash access protection state).

First you need to select the file to be compared. (As shown in Figure 29)

Figure 29. Firmware CRC



No.	File Name	File Size	Address Range(0x)
1	test_128k.bin	131072	08000000—0801FFFF

< >

Add
Delete

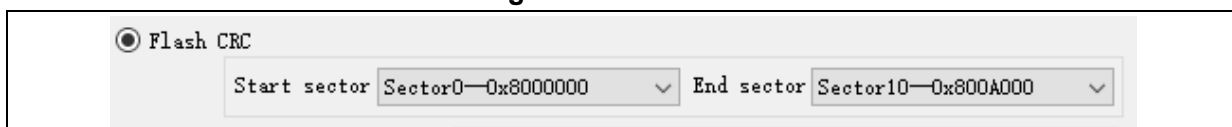
"Sector fill": the Firmware CRC is performed in units of sectors. What is filled in here is the download data that is not filled in the sector part. Generally, it is "FF".

5.4.7 Flash CRC

This function is used to calculate CRC value, including main Flash and SPIM.
(This function can be used in the Flash access protection state)

(As shown in Figure 30):

Figure 30. Flash CRC



☒ Flash CRC

Start sector: Sector0—0x8000000 End sector: Sector10—0x800A000

The start sector and end sector of memory must be set up.

5.4.8 Protection

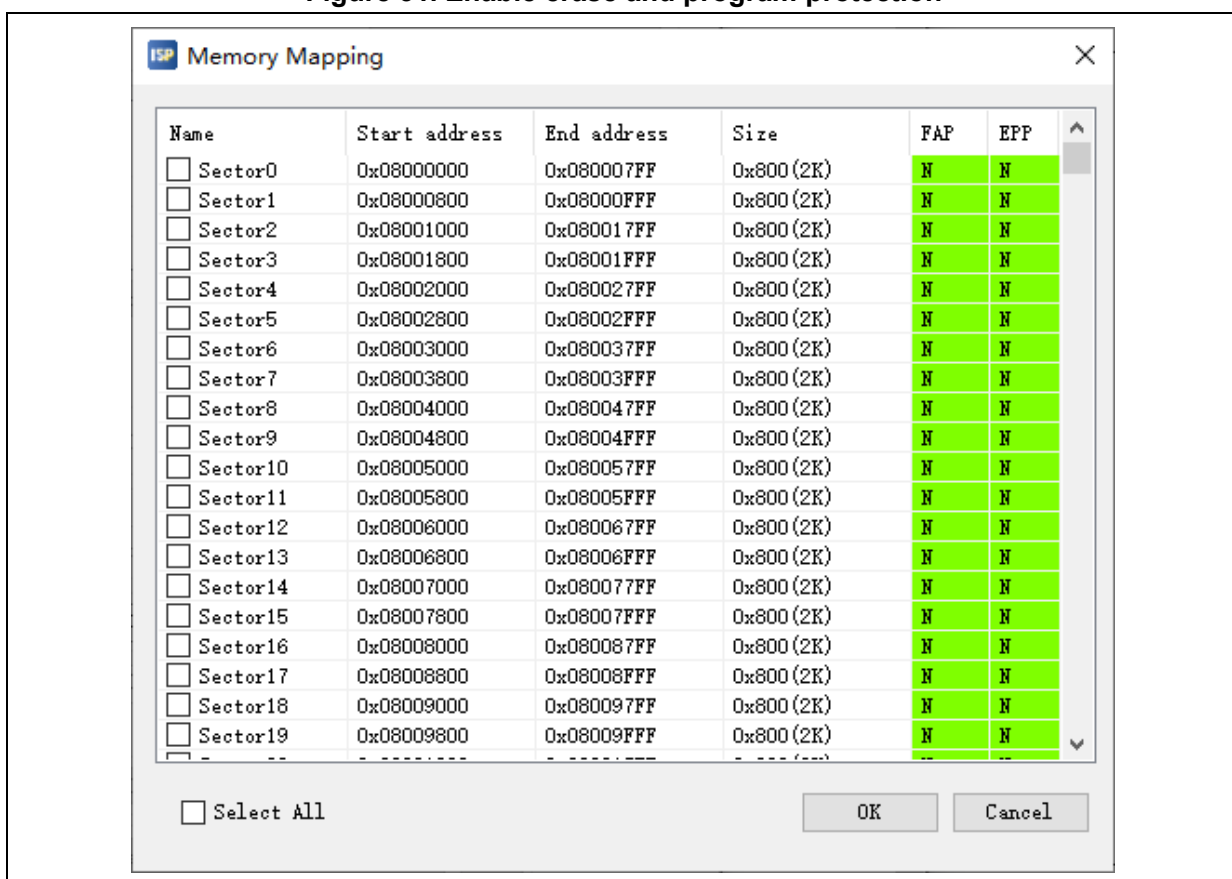
- Select "**Enable**" - "**Access Protection**" to enable the flash access protection. The whole flash will be access protected.

AT32F415/F421/F425/L021/F423/A423/F402/F405/F490/M412/M416: enable access protection and high level access protection (Access protection and user system data erase protection).

(AT32F425/L021/F423/A423/F402/F405/F490/M412/M416 high level access protection is irreversible. Once enabled, it will never be unlocked, with its debugging interface permanently disabled. Please use with caution.)

- Select "**Disable**" - "**Access Protection**" to disable the access protection of the whole flash.
- Select "**Enable**" - "**Erase and program protection**", and click "...", you can select the sectors to enable erase and program protection in the dialog box that pops up. (As shown in Figure 31)

Figure 31. Enable erase and program protection



- Select "**Disable**" - "**Erase and program protection**" to disable the erase and program protection of the whole flash.

5.5 Operation progress page

This page displays information related to the operation progress. (As shown in Figure 32)

Figure 32. Operation progress display



5.6 SPIM encryption download

SPIM encryption principle:

When SPIM encrypted download is required, users must first configure the SPIM FLASH_DA and SPIM encryption key (Key is set in the user system data), and then perform download operation. In this case, the MCU will encrypt the downloaded original data according to SPIM FLASH_DA and encryption key as well as internal algorithm in MCU, then write the encrypted data to SPIM.

When users want to read the encrypted data in the SPIM, users also need to configure the SPIM FLASH_DA and encryption key. Based on the SPIM FLASH_DA and encryption key, the MCU uses the MCU's internal algorithm to decrypt the encrypted data and restore it to the correct original data.

When downloading files to SPIM, the following steps can be set to encrypt the downloaded contents (AT32F403/F413/F403A/F407/A403A support SPIM)

Step 1: set the SPIM FLASH_DA (As shown in Figure 33).

Figure 33. Encryption range config

The screenshot shows the 'Artery ISP Programmer_V2.0.08' window. The 'Target' is set to 'AT32F403AVGT7_1024K'. The 'PID (h)' is '70050344', 'BID (h)' is '4703', and 'Protocol Version' is '3.2'. The 'SPIM' checkbox is checked. The 'SPIM Type' is 'Common Type2', '16MB', and 'Select' button is visible. The 'SPIM FLASH_DA' field is set to '0' and is circled in red. Below this is a 'Flash mapping' table with columns: Name, Start address, End address, Size, FAP, and EPP. The table lists sectors 0 through 9, each with a size of 0x800 (2K). The 'FAP' and 'EPP' columns are marked with 'N' for all sectors. At the bottom, there are buttons for 'Back', 'Next', 'Cancel', and 'Close'.

Name	Start address	End address	Size	FAP	EPP
Sector0	0x08000000	0x080007FF	0x800 (2K)	N	N
Sector1	0x08000800	0x08000FFF	0x800 (2K)	N	N
Sector2	0x08001000	0x080017FF	0x800 (2K)	N	N
Sector3	0x08001800	0x08001FFF	0x800 (2K)	N	N
Sector4	0x08002000	0x080027FF	0x800 (2K)	N	N
Sector5	0x08002800	0x08002FFF	0x800 (2K)	N	N
Sector6	0x08003000	0x080037FF	0x800 (2K)	N	N
Sector7	0x08003800	0x08003FFF	0x800 (2K)	N	N
Sector8	0x08004000	0x080047FF	0x800 (2K)	N	N
Sector9	0x08004800	0x08004FFF	0x800 (2K)	N	N

Y: Protected N: UnProtected

Starting from the address 0x08400000, plus the set FLASH_DA, it is the encryption area.
If encryption is not required, set to 0.

Step 2: set the SPIM encryption key through the "User system data". (As shown in Figure 34)

Figure 34. SPIM encryption key config



This is the encryption / decryption key for downloading and reading data in the encryption range of SPIM. When the access protection is disabled, the key is also erased.

Step 3: download the files to SPIM to implement encryption download.

6 Revision history

Table 15. Document revision history

Date	Revision	Changes
2024/07/11	V2.12	1. Support for AT32M412/M416 serial. 2. Added downloading One-Time Programmable data.
2024/04/26	V2.11	1. Support for AT32A423 serial.
2023/08/10	V2.10	1. Support for AT32F423VCW. 2. Support for AT32F402/F405 serial.
2023/07/06	V2.09	2. Support for AT32A403A serial.
2023/03/28	V2.08	1. Support for AT32F435ZDT7、AT32F435VDT7、AT32F435RDT7、AT32F435CDT7、AT32F435CDU7、AT32F437ZDT7、AT32F437VDT7、AT32F437RDT7.
2023/02/17	V2.07	1. Support for AT32F423 serial.
2022/08/25	V2.06	1. Support for AT32F4212C8T7.
2022/07/06	V2.04	1. Support for AT32L021 serial.
2022/01/26	V2.02	1. The serial port number supports a maximum of 1024.
2021/11/23	V2.01	1. Support for AT32F425 serial. 2. Support for AT32F403AVGW. 3. Support for AT32WB415 serial.
2021/10/09	V2.00	1. Initial release. Support for AT32F403/F413/F415/F421/F403A/F407/F435/F437.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Purchasers understand and agree that purchasers are solely responsible for the selection and use of Artery's products and 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.