UM0003
Application Note

ISP Multi-Port Programmer Manual

Introduction

This user manual gives an overview of Artery ISP Multi-Port Programmer. ISP Multi-Port Programmer acts as a graphic interface application designed to facilitate the use of ARTERY MCU. With the help of this programmer, users can also configure and operate multiple ARTERY MCU devices through UART or USB ports.

Contents

List of tables

List of figures

1 Introduction

1.1 Environmental requirements

- **Software requirements**

Windows 7 and above are required.

Software version below 3.0.02, .Net framework 4.0 is required.

Software version 3.0.02 and above, .Net framework 4.6 is required.

- **Hardware requirements**

Serial communication port (COM)
USB communication port.

When connecting to multiple devices at the same time, it is required to use a stable data cable, USB HUB and USB to serial interface device.

Users can choose multiple devices according to the PC configuration.
It is recommended to connect less than 16 devices, otherwise the system and data transmission performance may be affected.

1.2 Glossary

- **ISP:**

This refer 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 manage 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: the device must be connected to the serial communication port (COM) on the computer.
DFU: the device must be connected to a USB port on the computer.

- **USB DFU driver installation**

If using the USB DFU communication, 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 "Artery ISP Multi-Port Programmer.exe"

3 interfaces

3.1 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+

3.2 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+

3.3 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+

3.4 AT32F403A/F407 interfaces

Table 4. AT32F403A/F407 GPIO 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+

3.5 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

3.6 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.

3.7 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+

3.8 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

3.9 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

3.10 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+

3.11 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+

3.12 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^2C1	All	PB6: I2C1_SCL PB7: I2C1_SDA
I^2C2	AT32F405KxU7-4, AT32F402KxU7-4	Not supported
	Others	PB10: I2C2_SCL PB3: I2C2_SDA
I^2C3	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

3.13 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+

3.14 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

3.15 AT32F455/F456/F457 interfaces

Table 1. AT32F455/F456/F457 GPIO Pin Map

IP	MCUs supported	Pin
USART1	All	PA9：USART1_TX PA10：USART1_RX
USART2	AT32F455ZxT7，AT32F455VxT7 AT32F456ZxT7，AT32F456VxT7 AT32F457ZxT7，AT32F457VxT7	PD5：USART2_TX PD6：USART2_RX
	Others	PA2：USART2_TX PA3：USART2_RX
USART3	AT32F455ZxT7，AT32F455VxT7，	PC10：USART3_TX

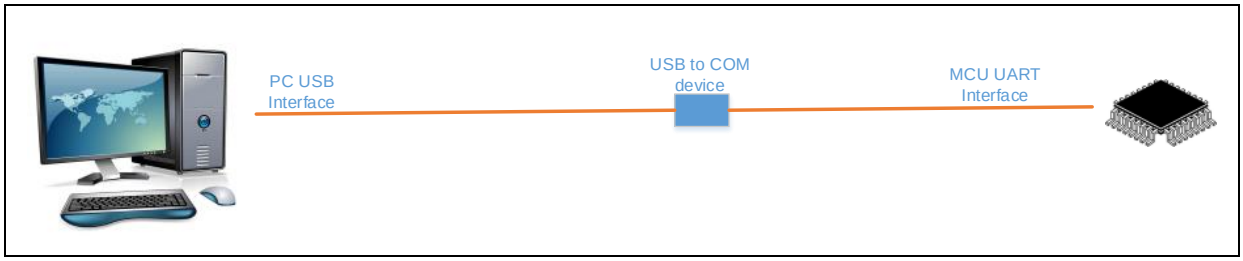
	AT32F455RxT7 AT32F456ZxT7, AT32F456VxT7, AT32F456RxT7 AT32F457ZxT7, AT32F457VxT7, AT32F457RxT7	PC11：USART3_RX 或 PB10：USART3_TX PB11：USART3_RX
	Others	PB10：USART3_TX PB11：USART3_RX
DFU	All	PA11：OTGFS1_D- PA12：OTGFS1_D+
I²C1	All	PB6：I2C1_SCL PB7：I2C1_SDA
I²C2	All	PB10：I2C2_SCL PB3：I2C2_SDA
I²C3	All	PA8：I2C3_SCL PB4：I2C3_SDA
CAN1	AT32F455ZxT7, AT32F455VxT7 AT32F456ZxT7, AT32F456VxT7 AT32F457ZxT7, AT32F457VxT7	PD0：CAN1_RX PD1：CAN1_TX
	Others	PB8：CAN1_RX PB9：CAN1_TX
CAN2	All	PB5：CAN1_RX PB13：CAN1_TX
SPI1	All	PA4：SPI1_CS PA5：SPI1_SCK PA6：SPI1_MISO PA7：SPI1_MOSI
SPI2	AT32F455ZxT7, AT32F455VxT7, AT32F455RxT7 AT32F456ZxT7, AT32F456VxT7, AT32F456RxT7 AT32F457ZxT7, AT32F457VxT7, AT32F457RxT7	PB12：SPI1_CS PC7：SPI1_SCK PC2：SPI1_MISO PC3：SPI1_MOSI
	Others	不支持

4 How to connect the device

ISP Multi-Port Programmer supports one device or multiple devices at the same time. It supports both UART and USB DFU. The device is connected as follows.

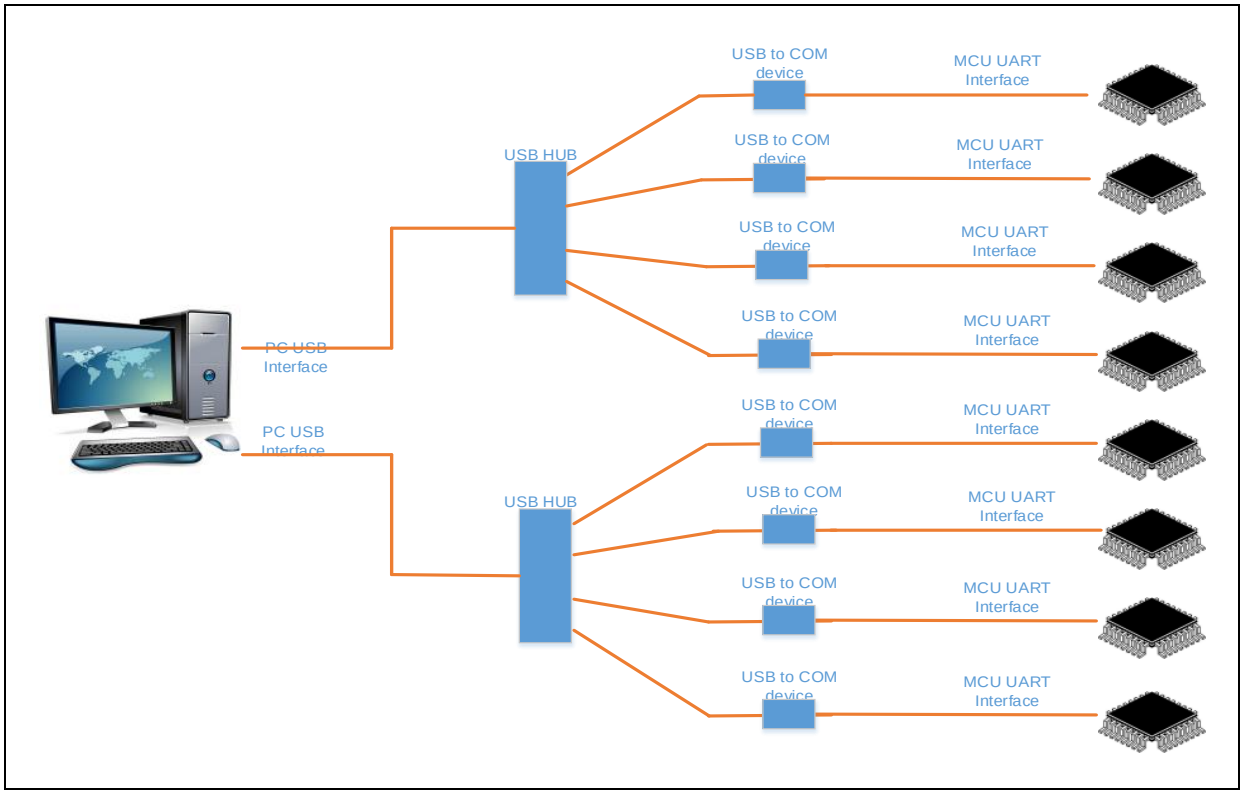
UART one device connection (As shown in Figure 1):

Figure 1. UART one device connection



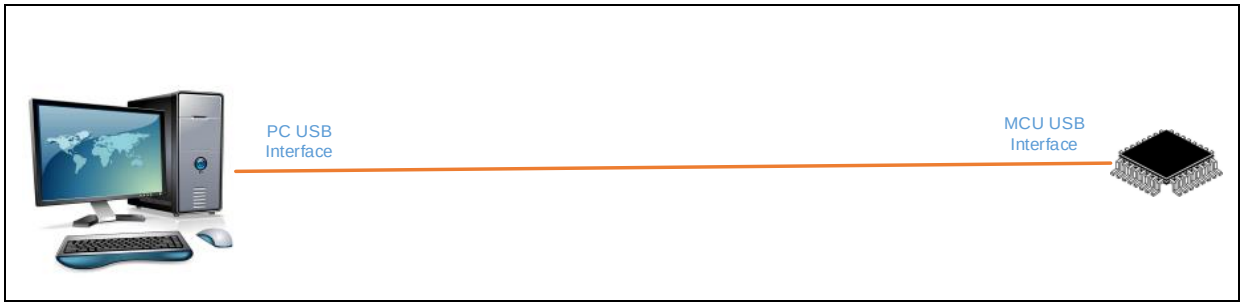
UART multiple devices connection: (As shown in Figure 2)

Figure 2. UART multi- device connection



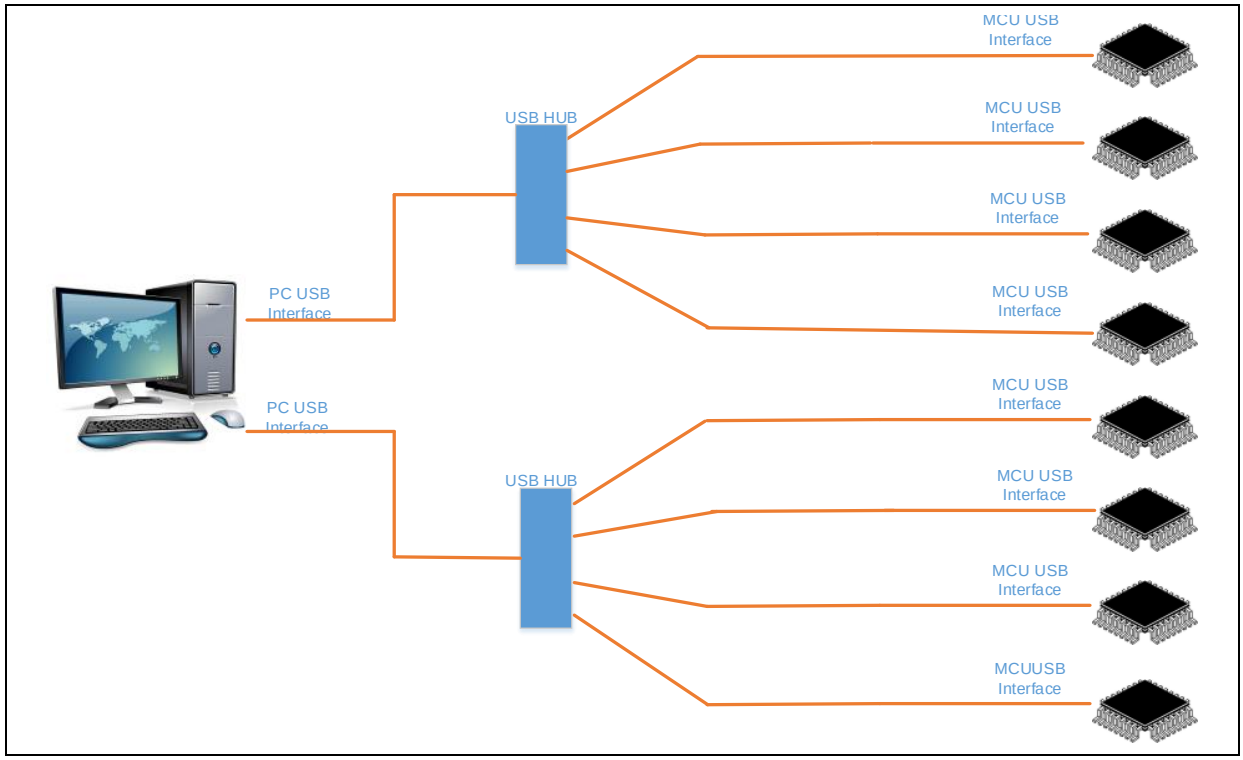
USB DFU one device connection: (As shown in Figure 3)

Figure 3. DFU one device connection



USB DFU multiple devices connection: (As shown in Figure 4)

Figure 4. DFU multi- device connection

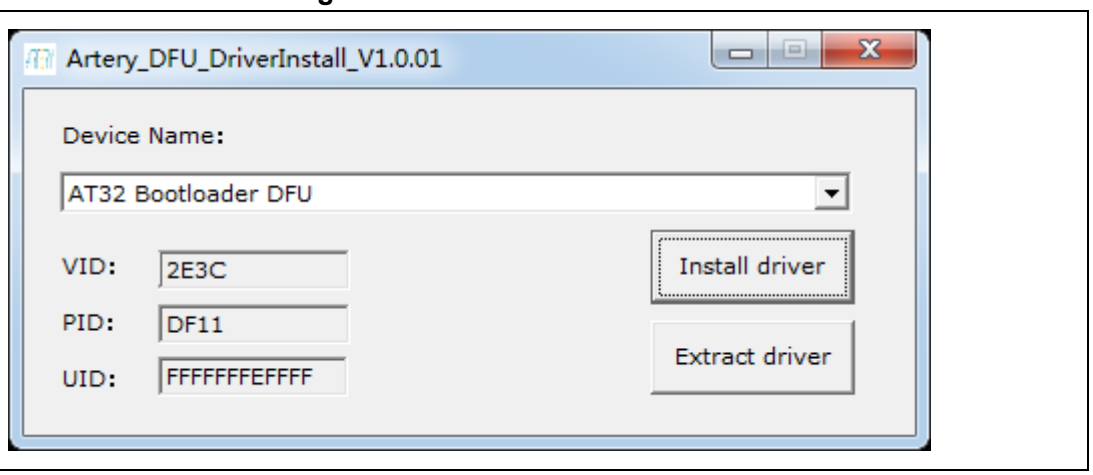


5 USB DFU driver installation

Artery provides the USB DFU driver automatic installation program "***Artery_DFU_DriverInstall.exe***". Double-click this program to enter the driver installation interface. (As shown in Figure 5)

This is 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 will be displayed respectively.

Figure 5. DFU driver install



5.1 Install driver automatically

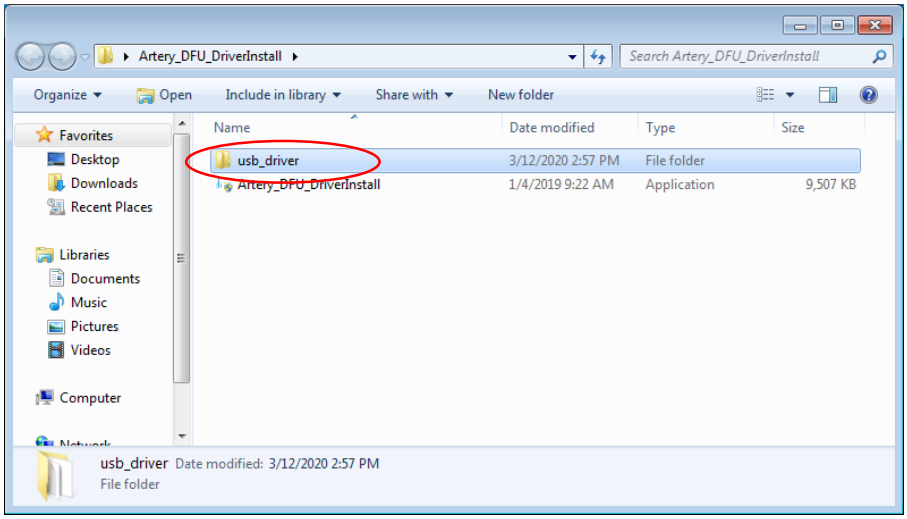
Click the "***Install driver***" to start the automatic installation of the driver.
If the installation is successful, a successful installation message will be displayed
If failed, an error message will be prompted.

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

5.2 Install driver manually

If the automatic installation failed or the user needs to install the driver manually, refer to this chapter for manual Installation.
Click "***Extract driver***", a driver installation package ("***usb_driver***" folder) will be generated in the current directory. (As shown in Figure 6)
This driver installation package is only available for the currently running operating system. If it is applied to other operating systems, the installation may fail.

Figure 6. Manual install-driver location

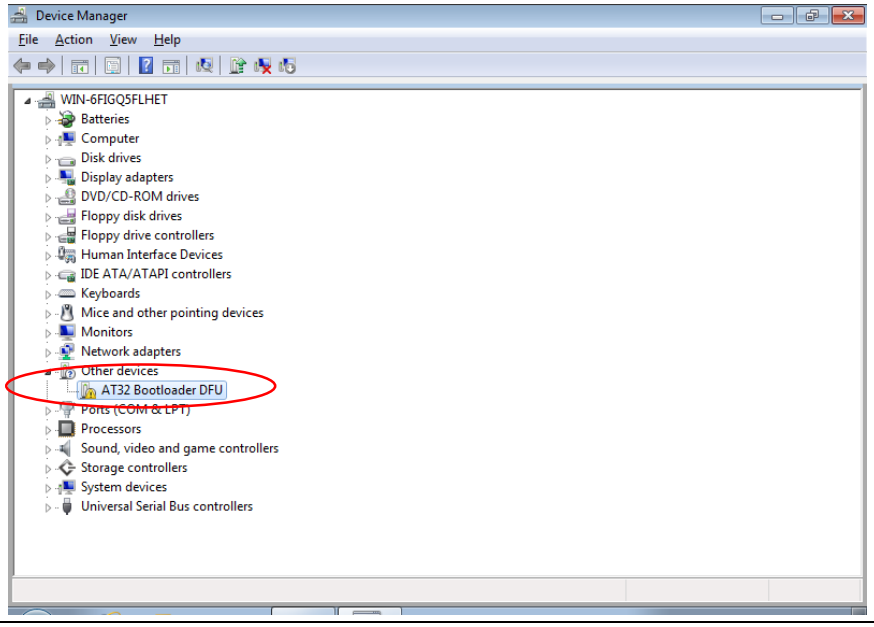


The procedures of manual installation are as follows: (take windows7 as an example)

- Open the "***Device Manager***" (As shown in Figure 7)

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

Figure 7. Manual install-device manager

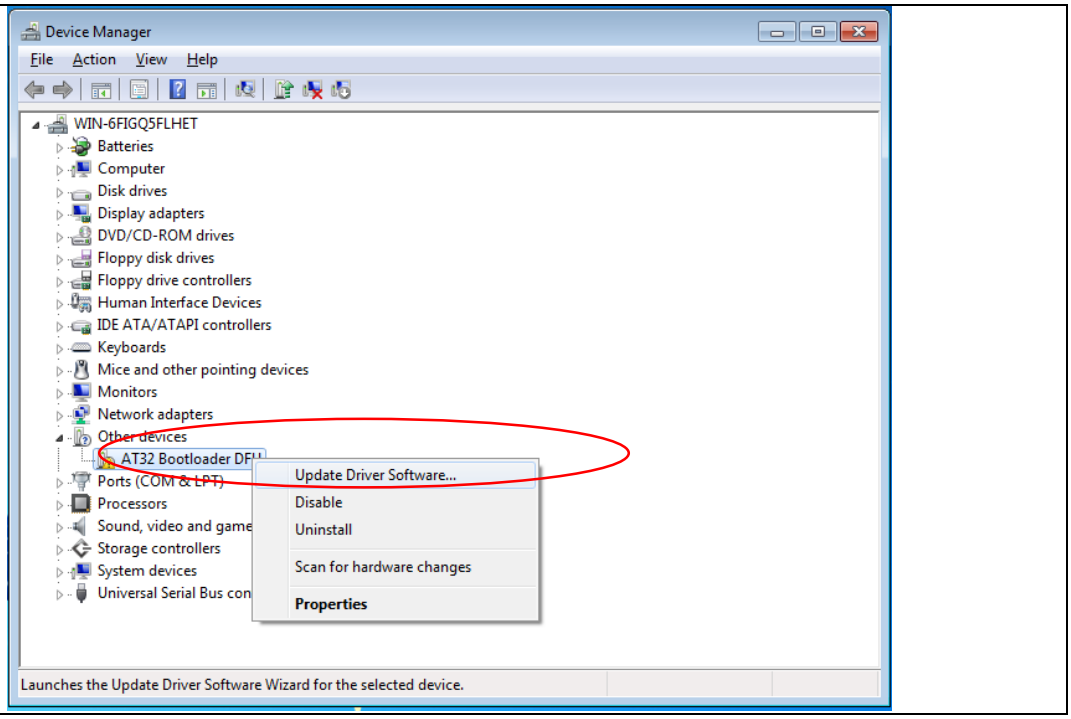


In this case, the "***Device Manager***" will scan the device "***AT32 Bootloader DFU***" without driver installed.

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

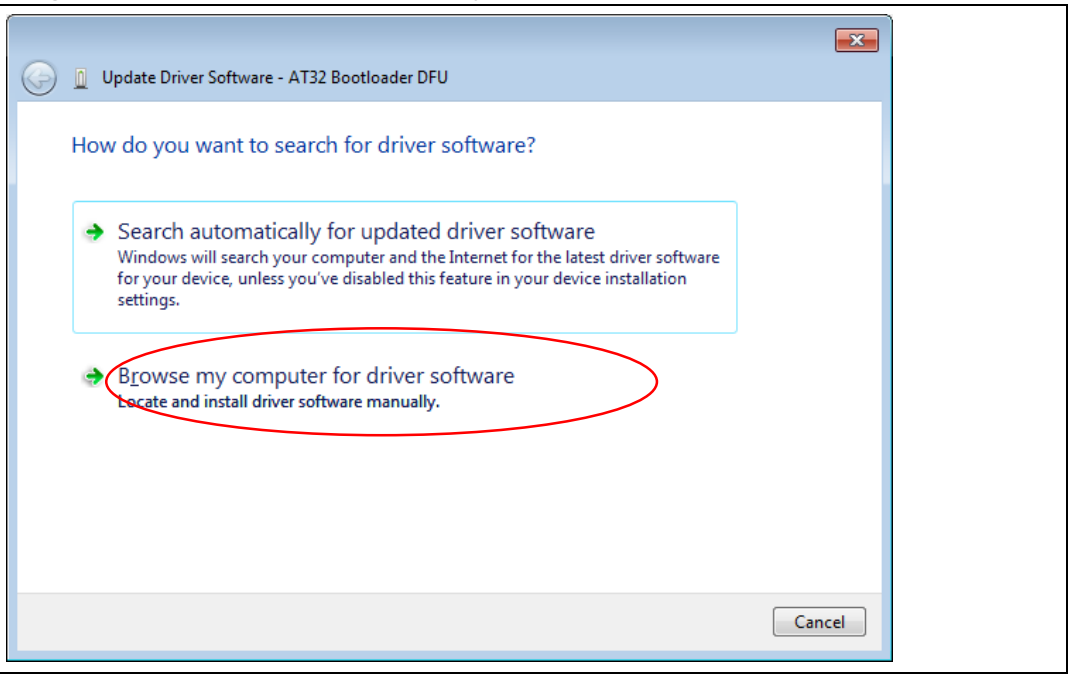
- Right-click on "***AT32 Bootloader DFU***" and select "***Update Driver Software***". (As shown in Figure 8)

Figure 8. Manual install-update driver



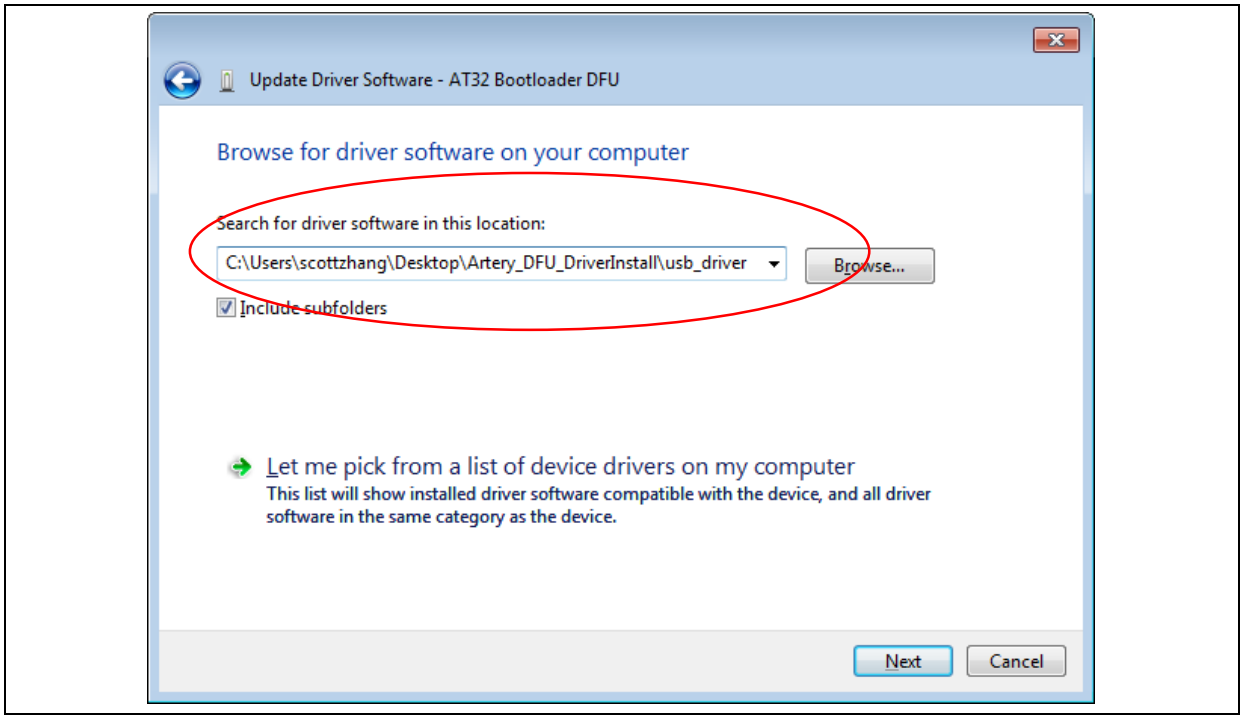
- Select "***Browse my computer for driver software***". (As shown in Figure 9)

Figure 9. Manual install-browse my computer for driver software



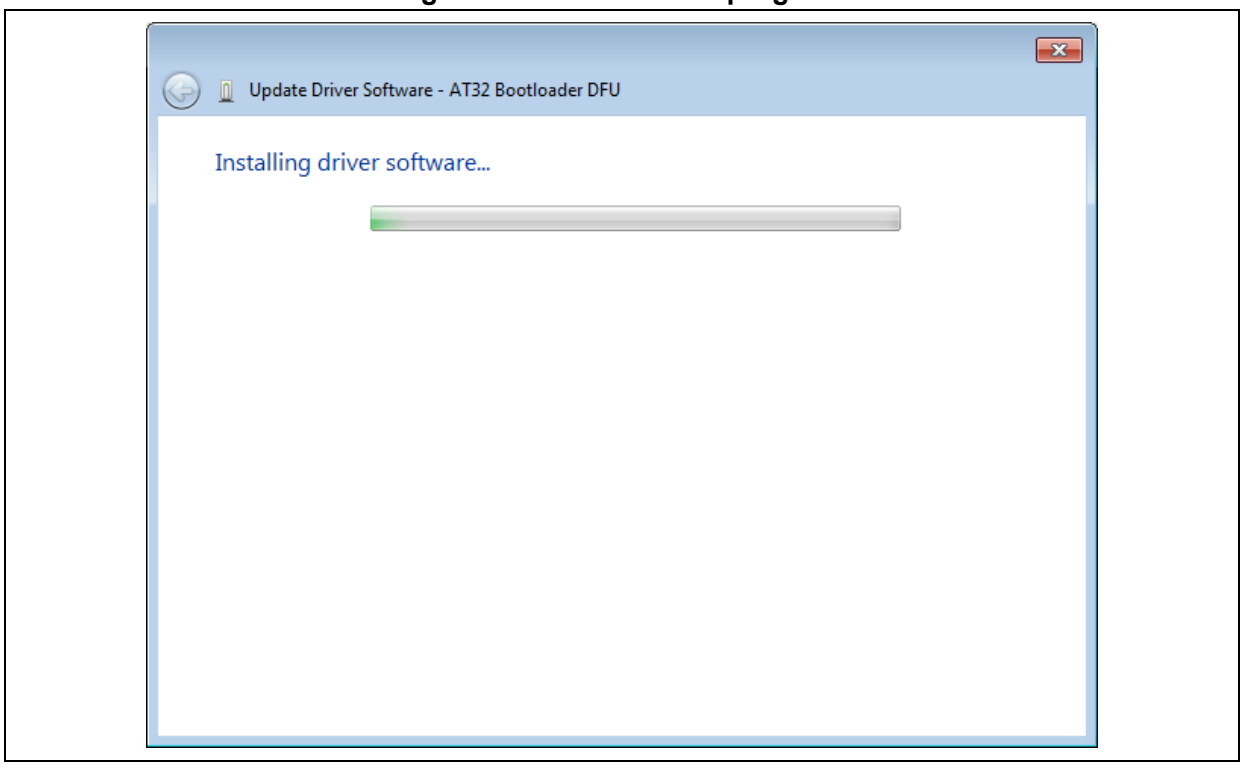
- Please select the driver software on your computer, that is, click on "***Extract driver***" to generate a driver installation package ("***usb_driver***" folder). Then click on "***Next***".(As shown in Figure 10)

Figure 10. Manual install-select driver



- Installing driver software. (As shown in Figure 11).

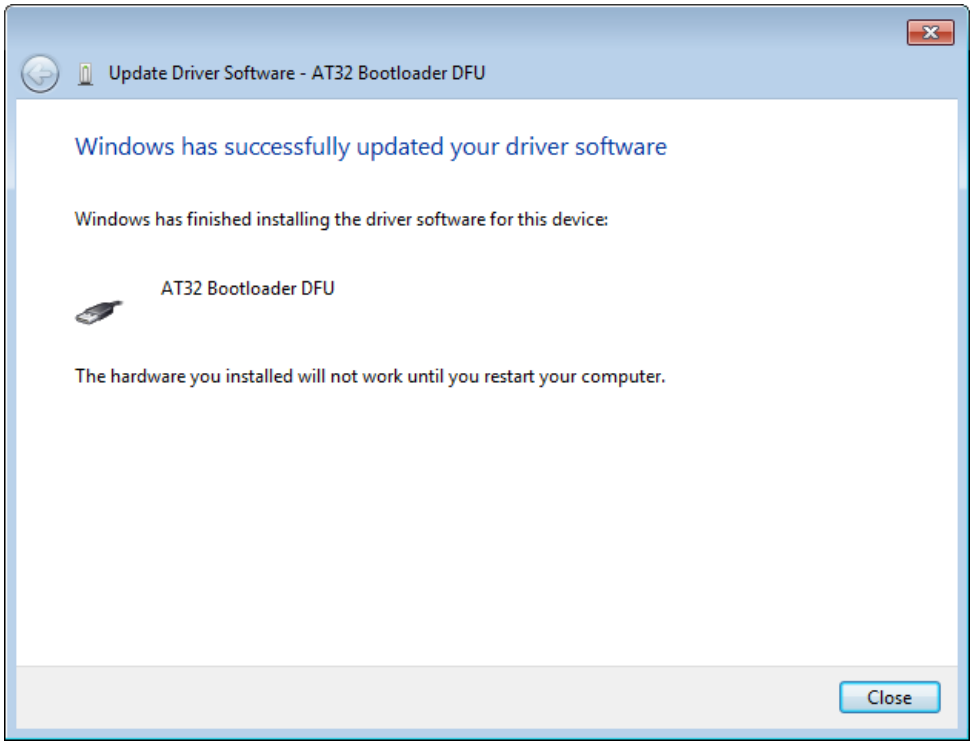
Figure 11. Manual install progress



Please wait for the driver installation to complete. After it is completed, click on "***Close***". (As shown in Figure 12)

The manual installation of the driver is now complete.

Figure 12. Manual install successful

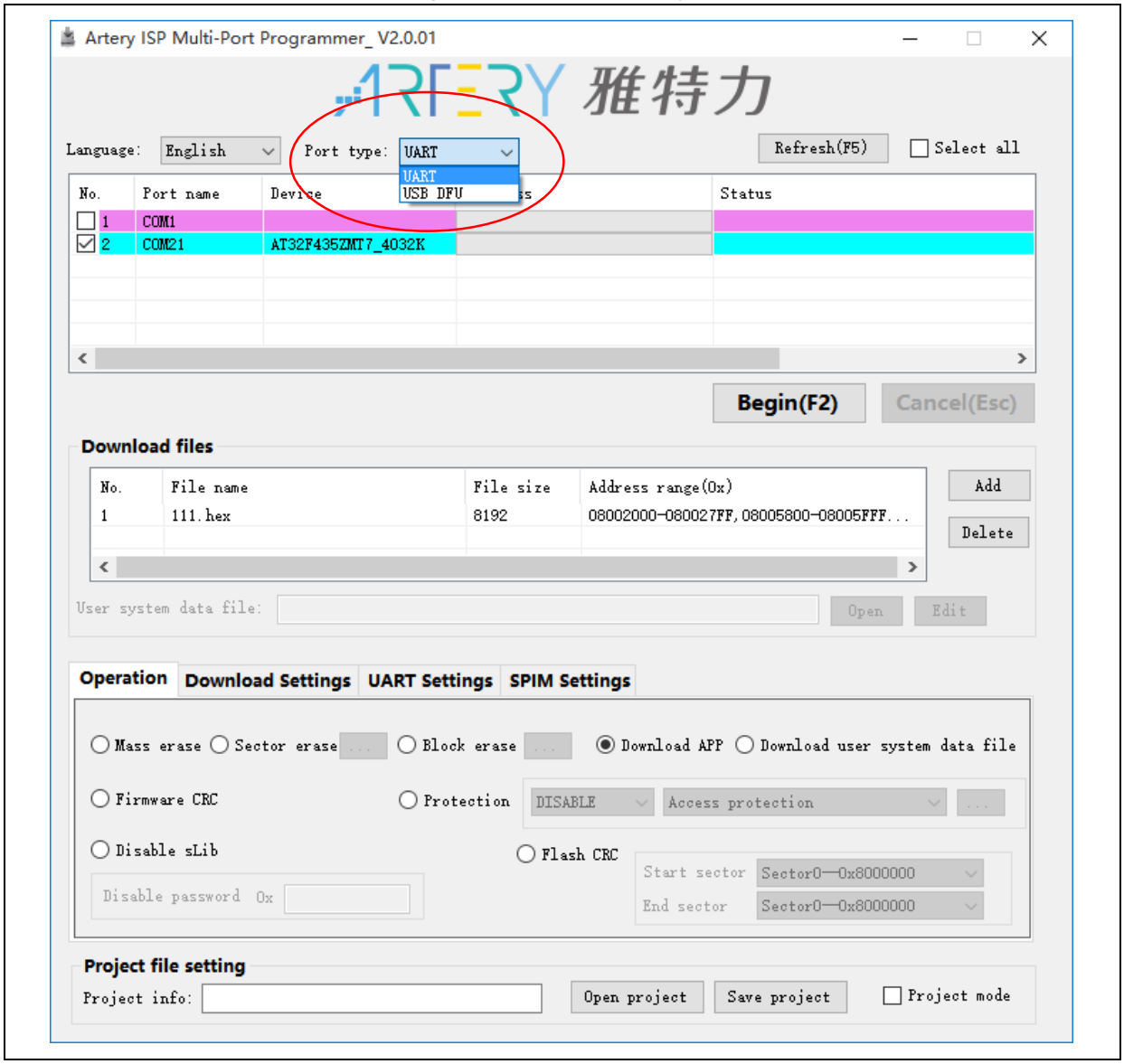


6 Device operation

6.1 Device connection

Users can select the corresponding connection method, that is, the interface type: UART or USB DFU. (As shown in Figure 13)

Figure 13. Select port type

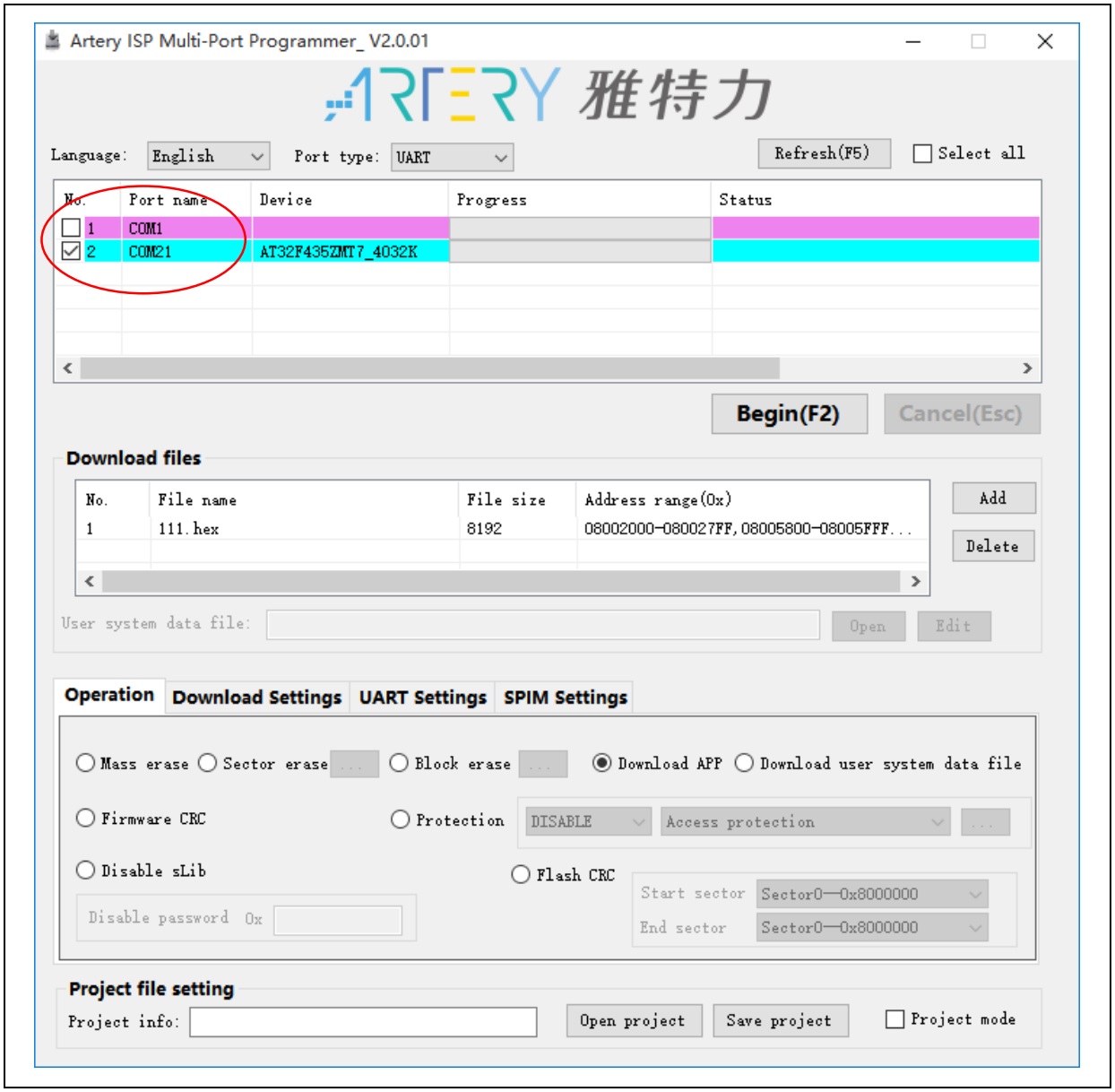


6.1.1 UART connection

When using the UART connection, the available serial interfaces will be enumerated automatically, or you can click on "***Refresh***" to re-enumerate the available serial interfaces and display the "***Port Name***". If "***Select All***" is checked, the devices will be connected and the device model will be displayed.

Please make sure that all devices to be operated are properly connected to the selected serial interface. (As shown in Figure 14)

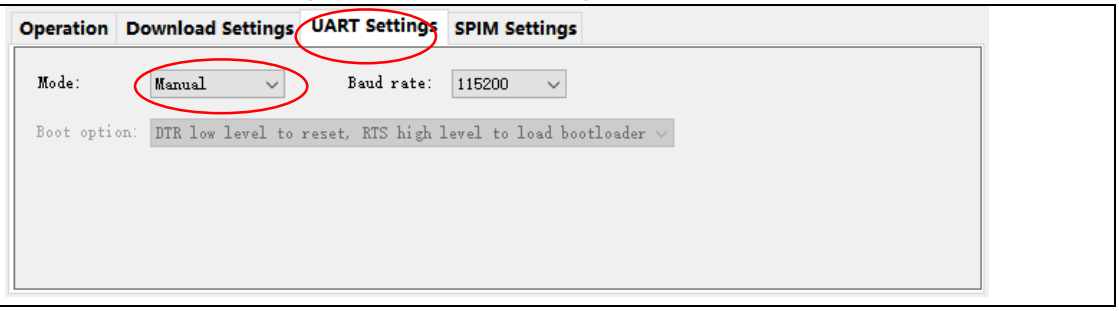
Figure 14. UART connection



UART Settings:

- Manual mode: (As shown in Figure 15)

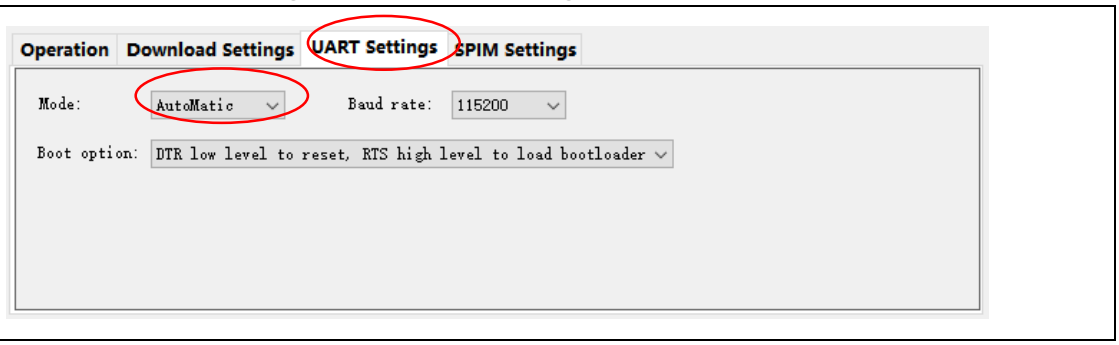
Figure 15. UART settings-manual



When "***Manual***" is selected in "***Mode***" option, you need to manually reset the device to restart the "Bootloader" embedded into the device.

- AutoMatic: (As shown in Figure 16)

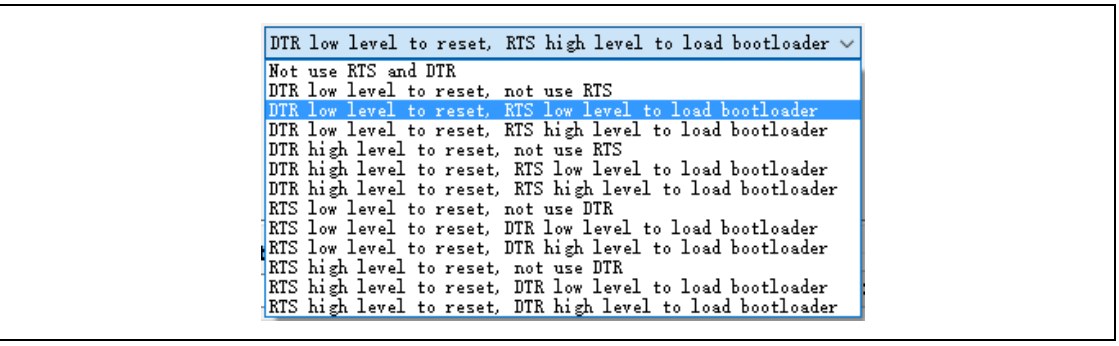
Figure 16. UART settings-automatic



When "***AutoMatic***" is selected in "***Mode***" option, the device must support automatic connection circuit, and the reset can be controlled by means of controlling DTR and RTS signals.

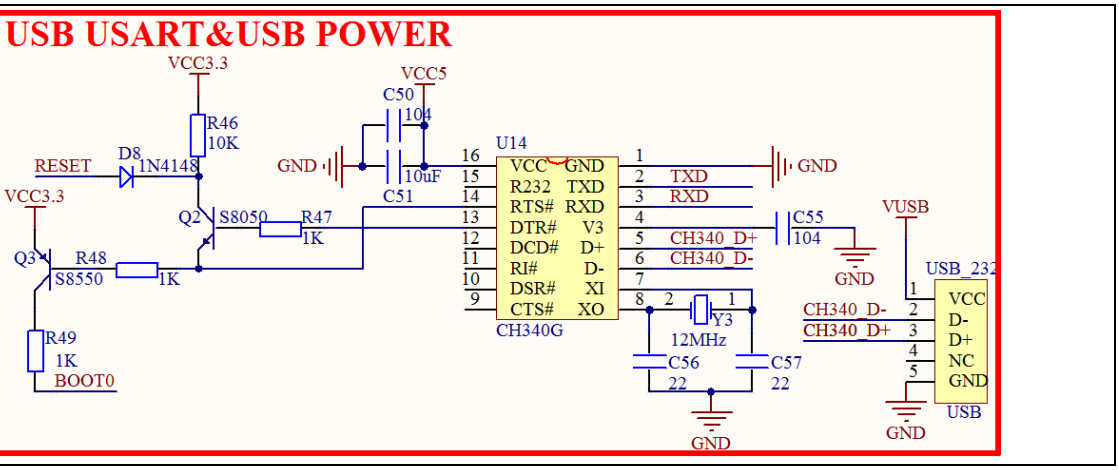
You can select the control mode supported by the current device in "***Boot Options***". (As shown in Figure 17)

Figure 17. UART settings-Boot option



The USB serial interface automatic connection circuit can be designed with reference to the following figure. (As shown in Figure 18)

Figure 18. USB auto connection circuit diagram



The combination of Q2 and Q3 in the figure constitutes the automatic connection circuit of the development board, which can 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 the need of setting B0 manually and pressing reset button. Among them, RESET is the reset signal of the development board, whereas BOOT0 is the B0 signal of the boot mode.

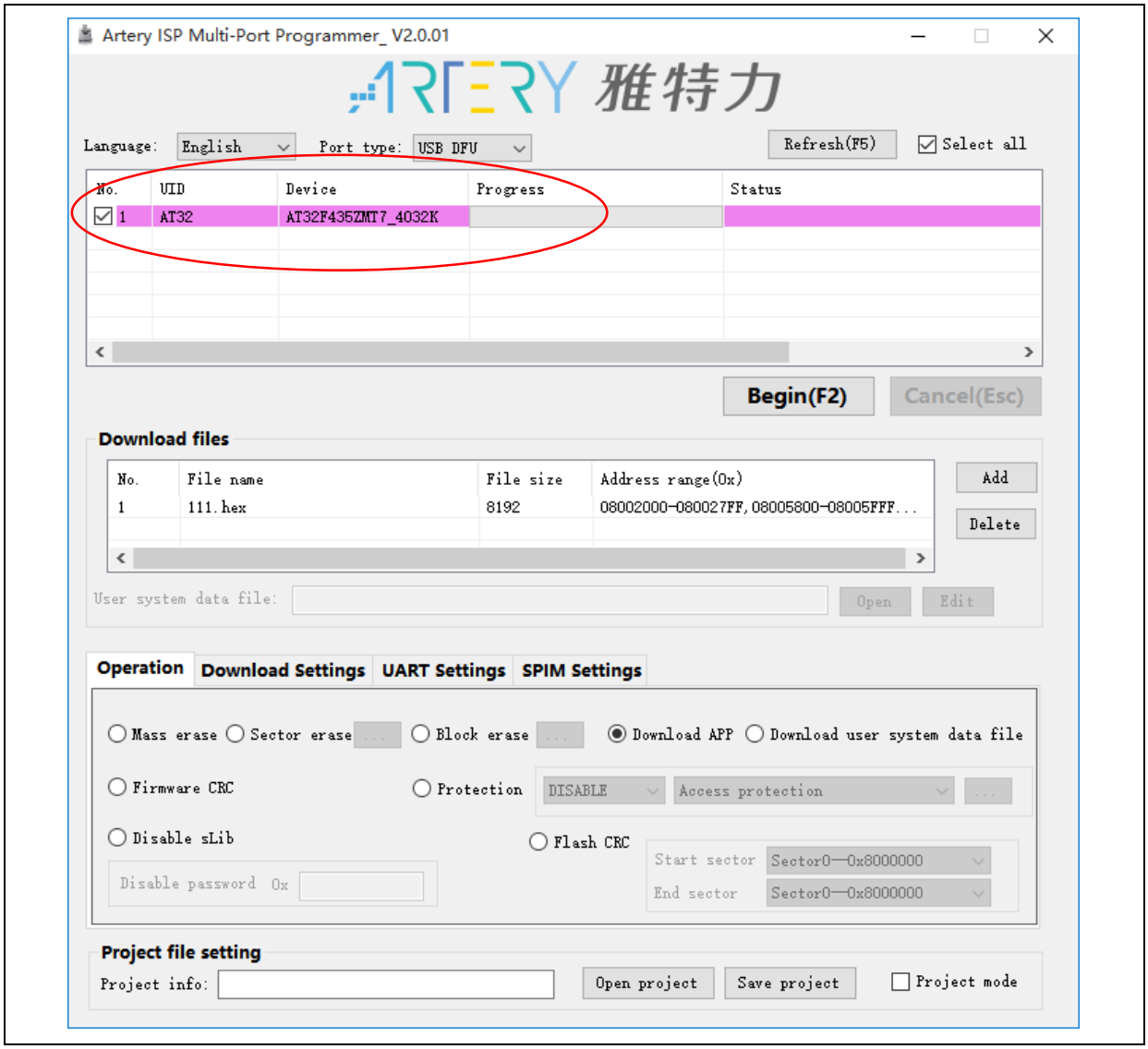
The following is the implementation process of automatic connection circuit when BOOT1 is low: First, the ISP controls the DTR output low level, then DTR_N output high, and RTS is set high, then the RTS_N output is low, at this time, Q3 is turned on and BOOT0 is pulled high, that is, BOOT0 is set to 1, and Q2 will also be turned on at the same time, and 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 the ISP can start to connect and download the code.

6.1.2 USB DFU connection

After using the USB DFU connection, it will automatically enumerate the devices that have been connected to PC, or click on "***Refresh***" button to re-enumerate the devices connected to PC, then the "**UID**" is displayed.When "***Select All***", the devices will be connected and the device model will be displayed.

Please make sure that all the devices to be operated have been correctly connected to the USB ports of PC. (As shown in Figure 19)

Figure 19. DFU connection



6.2 SPIM

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.
18. AT32F455 series MCUs do not support SPIM.
19. AT32F456 series MCUs do not support SPIM.
20. AT32F457 series MCUs do not support SPIM.

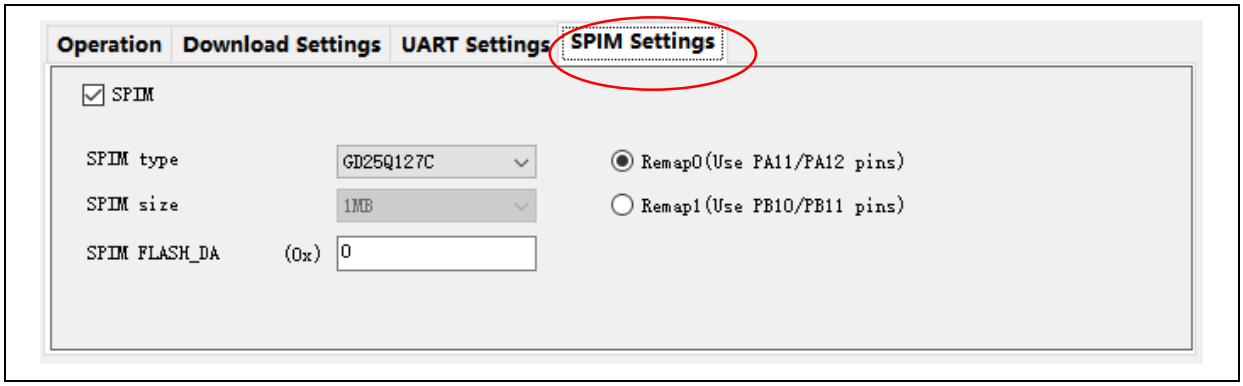
In DFU communication mode:

1. AT32F403 series MCUs do not support SPIM.
2. AT32F413KCU7-4/ AT32F413KBU7-4/AT32F413TBU7 in the AT32F413 series do not support SPIM; Other 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 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.
18. AT32F455 series MCUs do not support SPIM.
19. AT32F456 series MCUs do not support SPIM.

20. AT32F457 series MCUs do not support SPIM.

If SPIM is connected, please check "***SPIM***" and select "***SPIM type***". The "***SPIM size***" depends on "***SPIM type***". If SPIM encryption is required, set the "***SPIM FLASH_DA***" (As shown in Figure 20).

Figure 20. SPIM settings



- Checked "***SPIM***"

Allows operation on SPIM.

- Unchecked "***SPIM***"

Operation on SPIM is not allowed.

- ***Remap 0*** (use PA11/PA12 pins)
- ***Remap 1*** (use PB10/PB11 pins)

Select the desired pins. This option is only available for AT32F413/403A/407 series UART interfaces.

- ***SPIM Type***

You can select SPIM type. The supported types are as follows:
GD25Q127C
GD25Q64C
GD25Q32C
GD25Q16C
GD25Q80C
W25Q128V
EN25QH128A
Common Type1
Common Type2

If the SPIM you used is not displayed in the list, please select "Common Type1" or "Common Type2" to operate. In this case, if it still cannot be used normally, it can be determined that the SPIM used is not supported by this software.

- ***SPIM Size***

1 MB, 2 MB, 4 MB, 8 MB, 16 MB are available. For a fixed model of SPIM, SPIM size is also fixed.

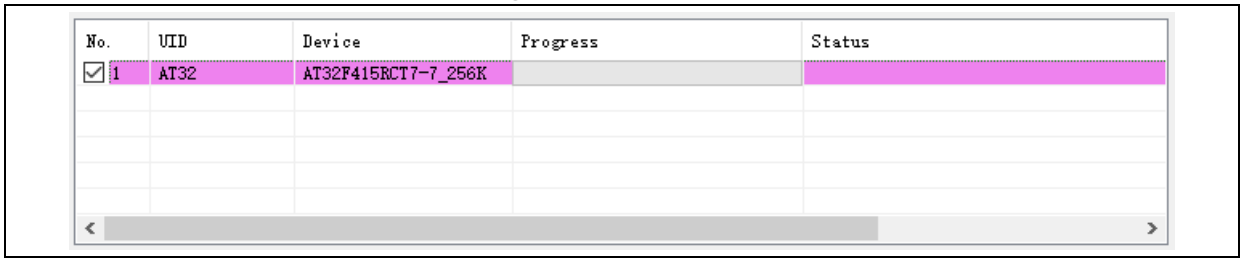
- ***SPIM FLASH_DA***

Set the encryption range when downloading files to the SPIM. The encryption range starts from address 0x08400000, with Byte as the unit.

6.3 Device list

(As show in Figure 21)

Figure 21. Device list



- ***No.***

The number of device in the list. If checked, the device will be connected, and the device information will be displayed under normal connection.

- ***UID*** (USB DFU connection):

The "UID" of the USB DFU device is displayed.

- ***Port name*** (UART connection):

The available port names are displayed.

- ***Device***

When the device is properly connected, the device information is displayed.

- ***Progress***

The real-time progress of various operations is displayed.

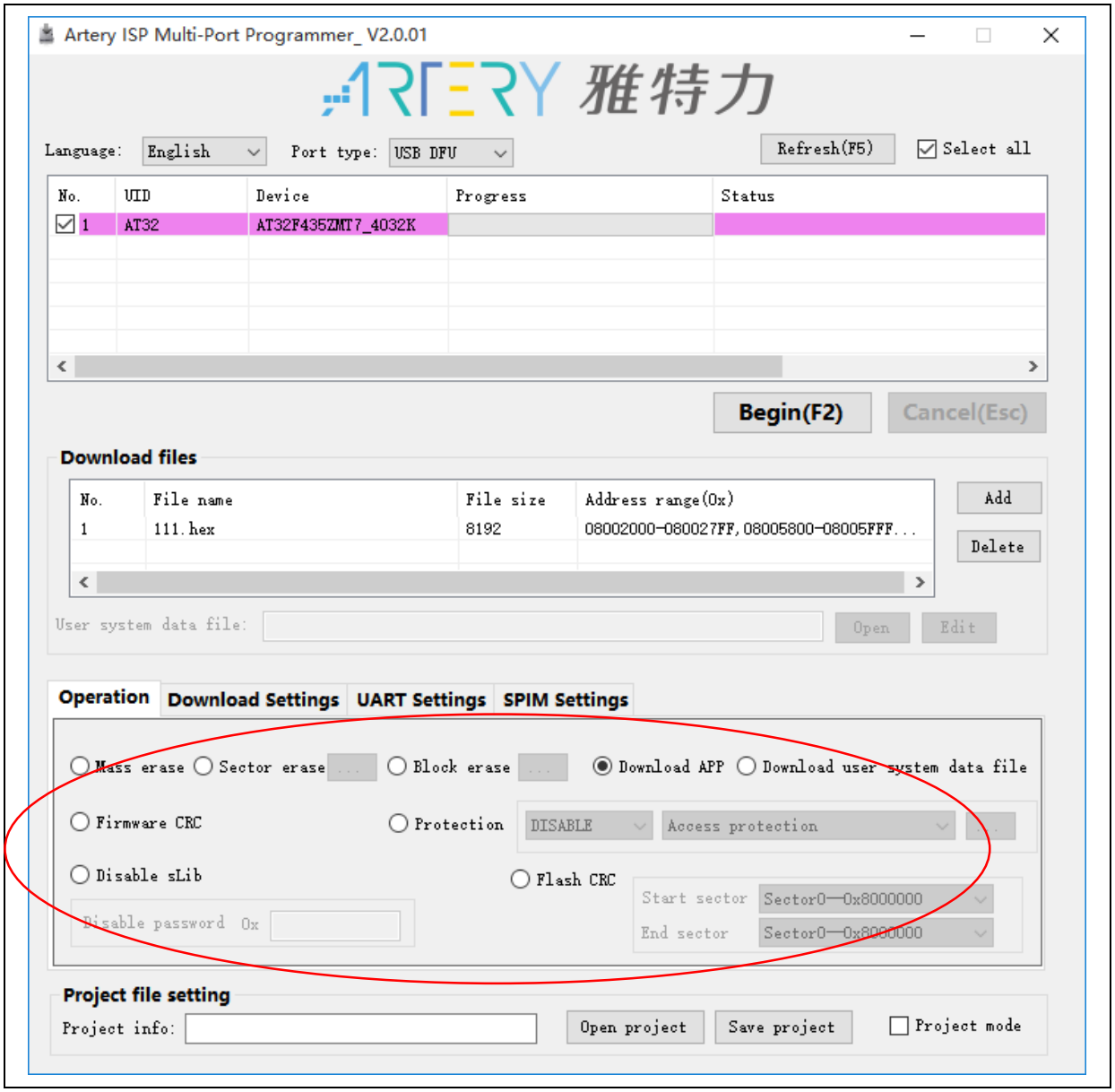
- ***Status***

When various operation is performed, the real-time status is displayed.

6.4 Functions

The functions contain Erase All, Sector Erase, Block erase, Download APP, Download user system data File, Firmware CRC, Flash CRC, Enable / Disable protection, Disable sLib and other functions (As shown in Figure 22).

Figure 22. Function operation



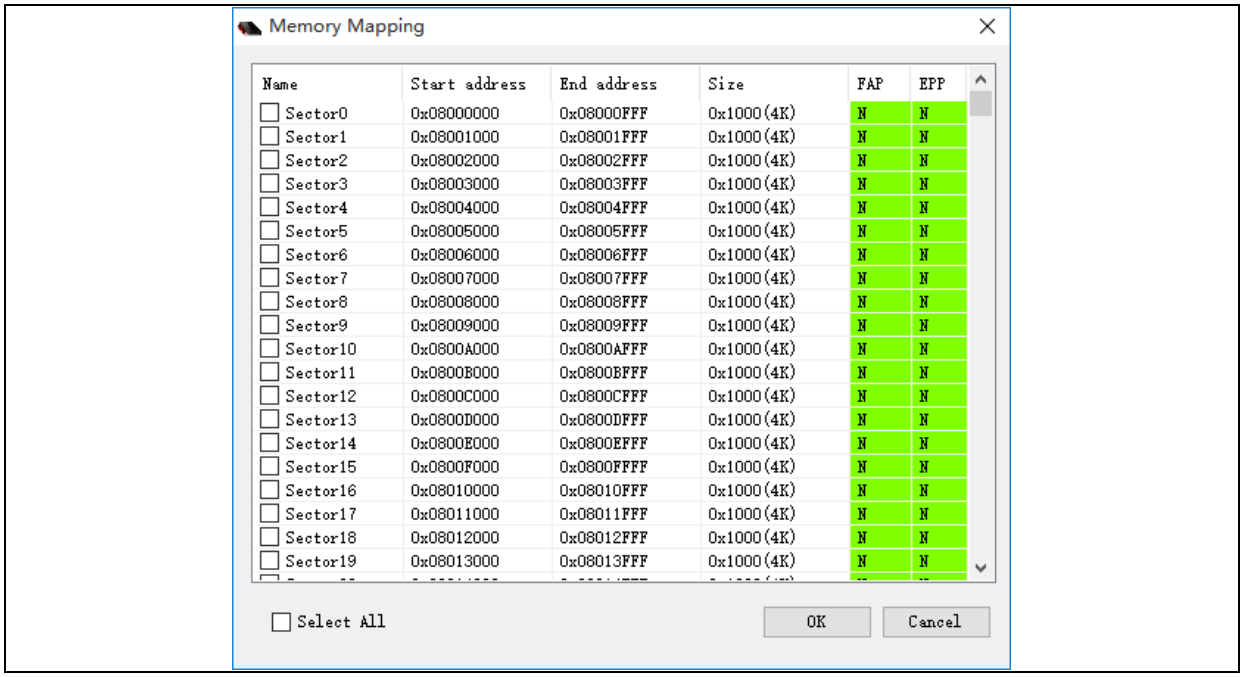
Select the function to be performed, and set the corresponding parameters, click on "Start" button, all devices in the list start to perform operations, and click on "***Cancel***" button to cancel the current operations of all devices.

6.4.1 Erase

- Select "***Erase All***"
  Click on "***Start***" button to erase the whole Flash memory. (Including SPIM)

- Select "***Sector Erase***"
  Customize the sectors to be erased. Click "**...**" to select the sector to be erased in the pop-up dialog box. (As shown in Figure 23)

Figure 23. Sector erase selection



Select all: select all sectorss.

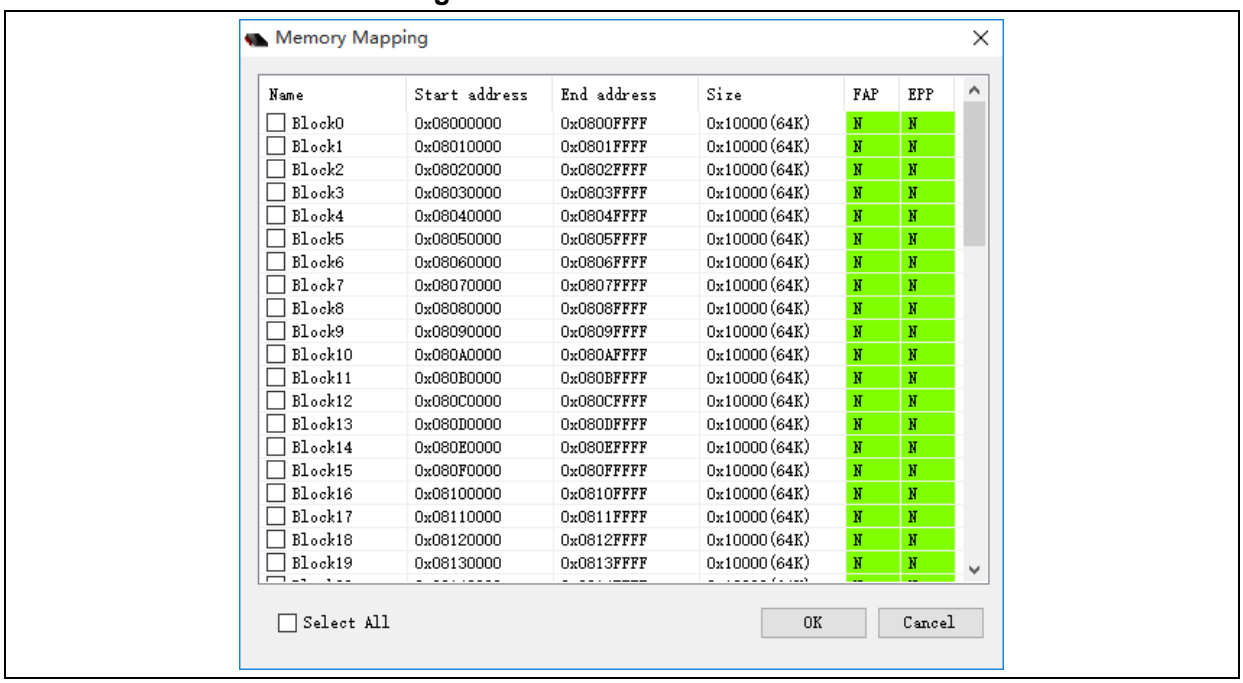
OK: make the selected sectorss take effect.

Cancel: cancel and close this dialog.

- Select "***Block Erase***"

  Customize the blocks to be erased. Click "**...**" to select the block to be erased in the pop-up dialog box. (As shown in Figure 24)

Figure 24. Block erase selection



Select all: select all blocks.

OK: make the selected blocks take effect.

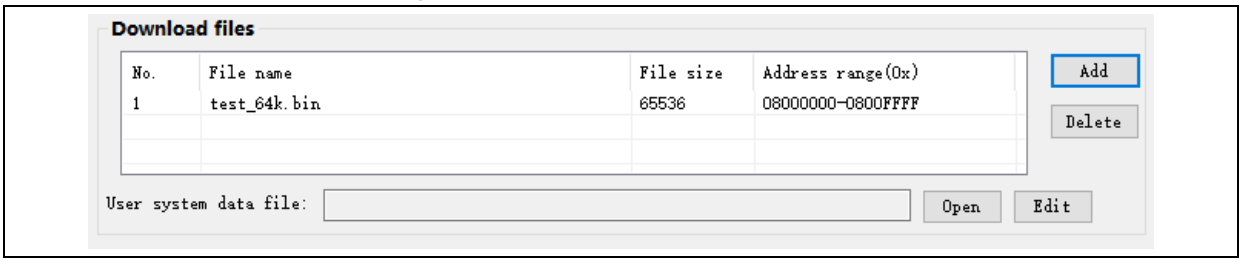
Cancel: cancel and close this dialog.

6.4.2 Download

Before downloading files, please set the download options:

Download files: (As shown in Figure 25)

Figure 25. Download files selection

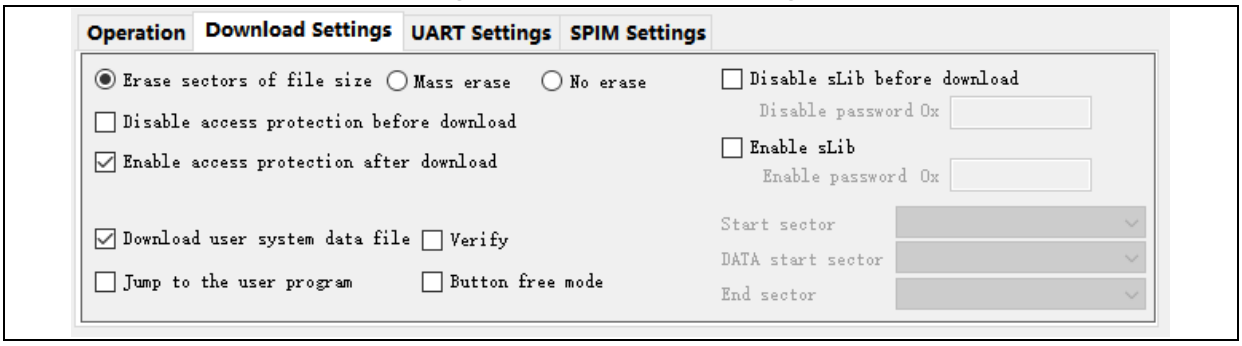


- Download files:
  The bin (binary) and hex (hexadecimal), s19/srec (Motorola S file) are supported.

- Address range:
  If you are loading a bin file, the default download address is the start address of the first sector in the memory. In this case, the download address can be modified.
  If you are loading a hex or s19/srec file, the download address is obtained from the loaded file. The download address cannot be modified at this time.

  Download Settings: (As shown in Figure 26)

Figure 26. Download settings



- Erase before download:
  "***Erase sectors of file size***": to erase sectors of the downloaded file.
  "***Mass erase***": to erase the whole flash memory (if SPIM is selected, the whole SPIM is also erased).
  "***No erase***": no erase operation is performed.

- The "***Disable access protection before download***" option is to disable access protection before download.

- The "***Enable access protection after download***" option is to enable access protection after download.
  For AT32F415/421/F425/L021/F423/A423/F402/F405/F490/M412/M416/F455/F456/F457, 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/F455/F456/F457 high level access protection is irreversible. Once enabled, it will never be unlocked, with its debugging interface permanently disabled. Please use with caution.)

- If "***Download user system data file***" option is checked, load the user system data file after download, and set the value to the device.

- If "***Verify***" option is checked, run the verify program after download to check whether the downloaded data is correct.

- If "***Jump to the user program***" option is checked, run the program directly after the download is completed.

- If "***Button free mode***" option is checked, after one device is downloaded, replace it with a new one, the software will automatically download the new one.
  UART: one or more devices can be connected in "Button free mode".
  DFU: only one DFU device can be connected in "Button free mode".

(AT32F403 series not support sLib function)

- The "***Disable sLib before download***" option is to disable sLib before downloading files. You need to enter the correct sLib password.

- The "***Enable sLib***" option is to enable sLib before downloading files. You need to enter the encryption password, start sector, DATA start sector/INSTER start sector and end sector.

- ***Start sector***
  AT32F413/415/403A/407/AT32A403A：
  The start sector of sLib area. The instruction area is from the "Start sector" to the "DATA start sector"(excluding the DATA start sector). When sLib is enabled, the data in this area cannot be erased, written or read.

  AT32F421/435/437/F425/L021/F423/A423/F402/F405/F490/M412/M416/F455/F456/F457:
  The start sector of sLib area. The area from the “Start sector” to “INSTR start sector” (not including “INSTR start sector”) is a mix area of instruction and data (read only area). When sLib is enabled, the data in this area cannot be erased, written, but can be read.

- ***DATA start sector/INSTR start sector***
  AT32F413/415/403A/407/AT32A403A：
  The start sector of the sLib data area. The 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.

  AT32F421/435/437/F425/L021/F423/A423/F402/F405/F490/M412/M416/F455/F456/F457：
  The start sector of the 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 set to "none", it is no instruction area.

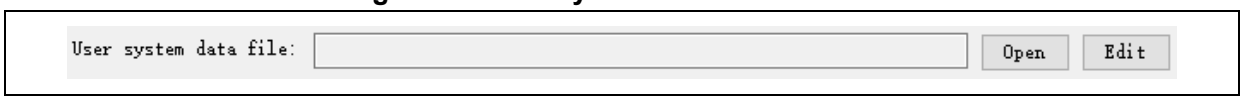
- ***End sector***
  The end sector of the sLib area.

6.4.3 Download user system data file

This function is used to download the user system data file to each device separately.

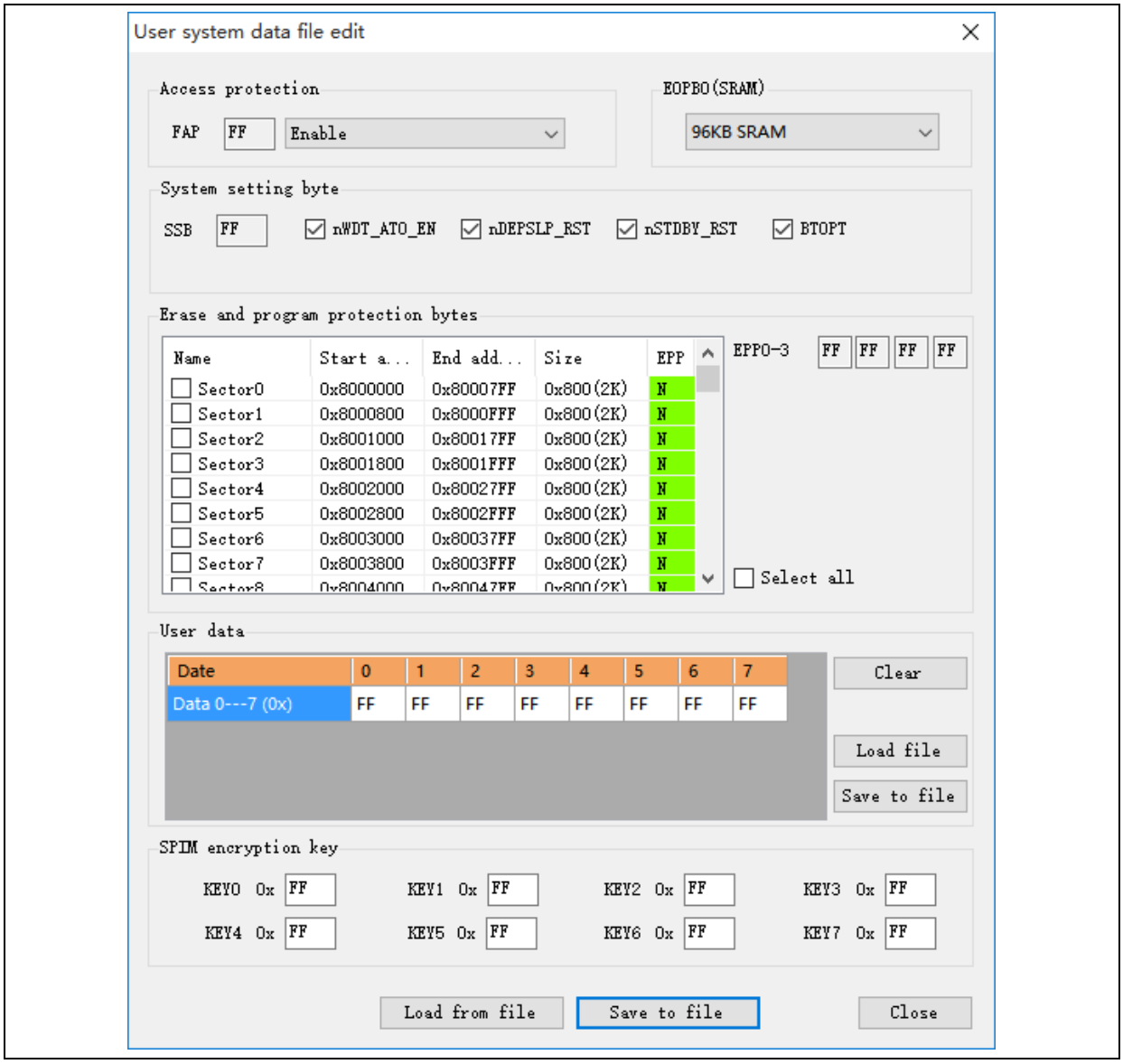
- User system data file: the file selected in "User system data file". (As show in Figure 27)

Figure 27. User system data file selection



User system data file supports bin (binary) and hex (hexadecimal).
"***Open***": open a new user system data file.
"***Edit***": edit user system data file data. If no user system data file is opened, the value of the user system data in the edit interface is all 0xFF, and a new user system data file can be edited on this basis. (As shown in Figure 28) (AT32F403A user system data):

Figure 28. User system data edit



■ Access protection

AT32F403/413/403A/407/435/437:

Enabled: FAP----0xFF.

Disabled: FAP---0xA5.

AT32F415/421/F425/L021/F423/A423/F402/F405/F490/M412/M416/F455/F456/F457:

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/F455/F456/F457 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.

■ 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 (AT32F403/413/403A/407/435/437)
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.

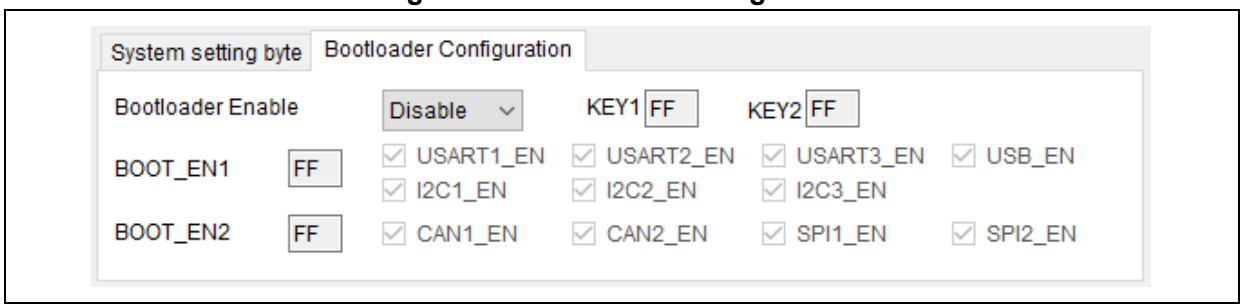
nBOOT1 (AT32F421/F425/L021/F423/A423/F402/F405/F490/M412/M416/F455/F456/F457)
Boot mode is determined together with BOOT0, and when BOOT0 = 1,
Unchecked----SRAM is selected as boot space.
Checked---Boot memory is selected as boot space.

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.

- Bootloader Configuration

Figure 29. Bootloader Configuration



Bootloader Enable：
Enable--------Bootloader peripherals enablement can be configured.
Disable--------Bootloader peripherals enablement cannot be configured. By default, all peripherals are enabled.

USART1_EN:
Unchecked -----Disable USART1.
Checked--------Enable USART1.

USART2_EN:
Unchecked -----Disable USART2.
Checked--------Enable USART2.

USART3_EN:
Unchecked -----Disable USART3.
Checked--------Enable USART3.

USB_EN:
Unchecked -----Disable USB.
Checked--------Enable USB.

I2C1_EN:
Unchecked -----Disable I2C1.
Checked--------Enable I2C1.

I2C2_EN:
Unchecked -----Disable I2C2.
Checked--------Enable I2C2.

I2C3_EN:
Unchecked -----Disable I2C3.
Checked--------Enable I2C3.

CAN1_EN:
Unchecked -----Disable CAN1.
Checked--------Enable CAN1.

CAN2_EN:
Unchecked -----Disable CAN22.
Checked--------Enable CAN2.

SPI1_EN:
Unchecked -----Disable SPI1.
Checked--------Enable SPI1.

SPI2_EN:
Unchecked -----Disable SPI2.
Checked--------Enable SPI2.

- EOPB0(SRAM)

AT32F403/403A/407: (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.

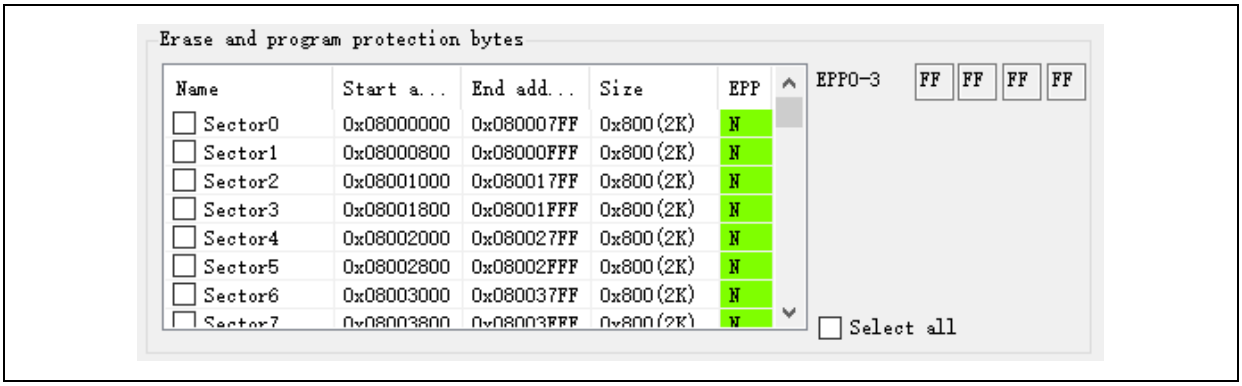
AT32F415/421/F425/L021/F423/A423/F402/F405/F490/M412/M416/F455/F456/F457: (not support)

AT32F435/437:
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 30)

Figure 30. Erase and program protection bytes



EPP0:
AT32F403/413/403A/407: controls the erase and program protection of sectors in the range of Flash 1K-32K.
AT32F415/F423/A423/F402/F405/F455/F456/F457: controls the erase and program protection of Sector0-Sector15.
AT32F421: controls the erase and program protection of Sector0-Sector31.
AT32F435/437: 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:

AT32F403/413/403A/407: controls the erase and program protection of sectors in the range of Flash 33K-64K.
AT32F415/F423/A423/F402/F405/F455/F456/F457: controls the erase and program protection of Sector16-Sector31.
AT32F421: controls the erase and program protection of Sector32-Sector63.
AT32F435/437: 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 Sector32-Sector63.

EPP2:
AT32F403/413/403A/407: controls the erase and program protection of sectors in the range of Flash 65K-96K.
AT32F415/F423/A423/F402/F405/F455/F456/F457: controls the erase and program protection of Sector32-Sector47.
AT32F435/437: 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:
AT32F403/413/403A/407:
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/F455/F456/F457:
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)
AT32F435/437: 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:
AT32F435/437: controls the erase and program protection of sectors in the range of Flash 129K-1152K. Each bit protects 128K bytes sectors.

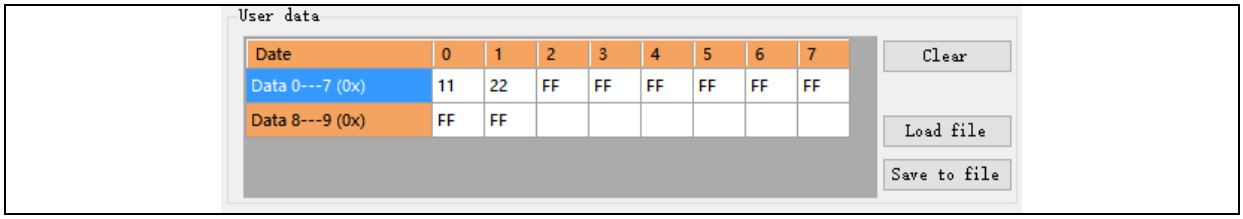
EPP5:
AT32F435/437: controls the erase and program protection of sectors in the range of Flash 1153K-2176K. Each bit protects 128K bytes sectors.

EPP6:
AT32F435/437: controls the erase and program protection of sectors in the range of Flash 2177K-3200K. Each bit protects 128K bytes sectors.

EPP7:
AT32F435/437: 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 31. User data



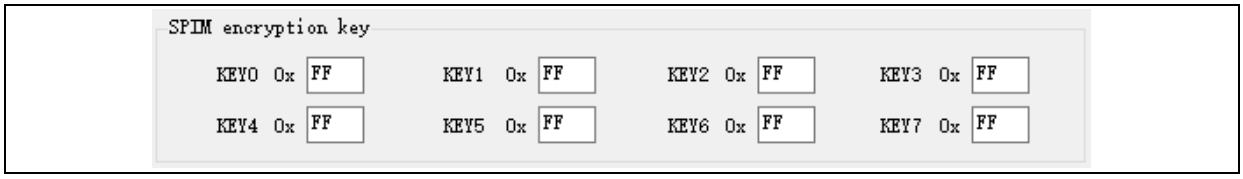
AT32F403/413/403A/407: user data 8 bytes.
AT32F415: user data 10 bytes.
AT32F421: user data 250 bytes.
AT32F435/437: Flash size is less than 4032K, user data 220 bytes. Flash size 4032K, user data 2012 bytes.
AT32F425: user data 250 bytes.
AT32L021: user data 250 bytes.
AT32F423: user data 250 bytes.
AT32A423: user data 250 bytes.
AT32F402/F405: user data 220 bytes.
AT32M412/M416: user data 250 bytes.
AT32/F455/F456/F457: user data 216 bytes.

Clear：all the data user system data are set to 0xFF, and not saved to device.
Load file: load the saved data user system data into the table for display.
Save to file: save the data user system data in the table into files.

- SPIM encryption key (AT32F403/413/403A/407)

You can set the encryption key when downloading. (As shown in Figure 32)

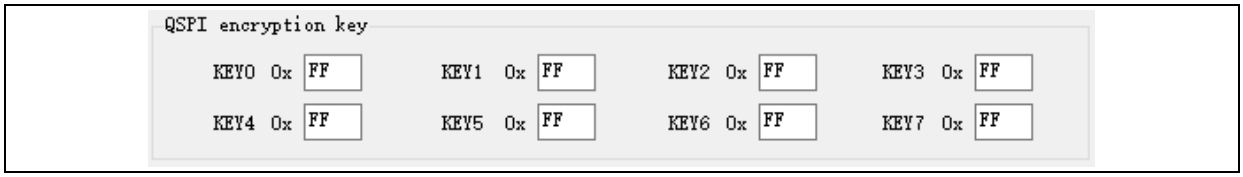
Figure 32. SPIM encryption key config



- QSPI encryption key (AT32F435/437/F402/F405/F455/F456/F457)

You can set the encryption key when downloading the QSPI. (As shown in Figure 33)

Figure 33. QSPI encryption key



- Load from file:

Load another user system data file. When the loading is successful, the user system data file path will be displayed in the main UI interface.

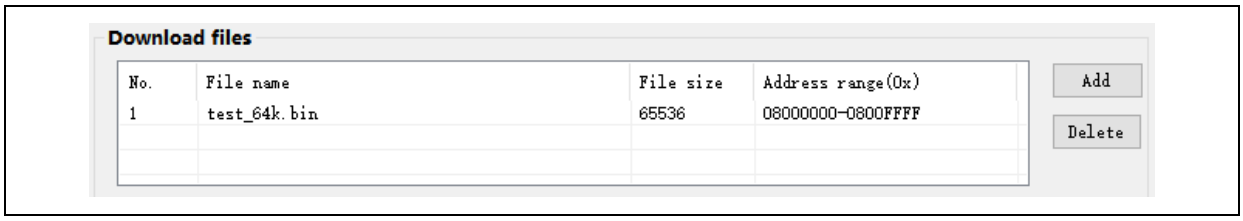
- Save to file:

Save the configuration to the user system data file. When the save is successful, the user system data file path will be displayed in the main UI interface.

6.4.4 Firmware CRC

This function is used to calculate the CRC code and compare it with the downloaded files to confirm if the downloaded content is correct or not. (As shown in Figure 34)
(This function can be used in flash access protection state)

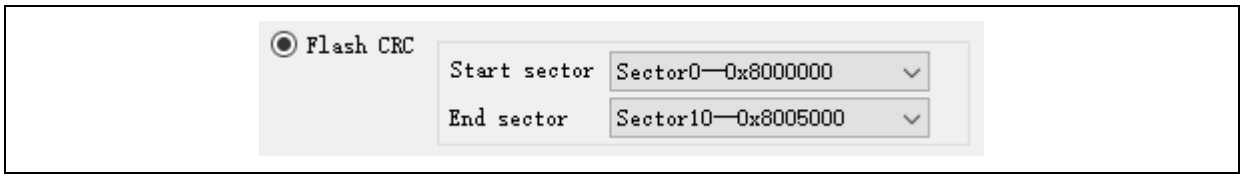
Figure 34. Firmware CRC



6.4.5 Flash CRC

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

Figure 35. Flash CRC



The start sector and end sector must be set.

6.4.6 Protection

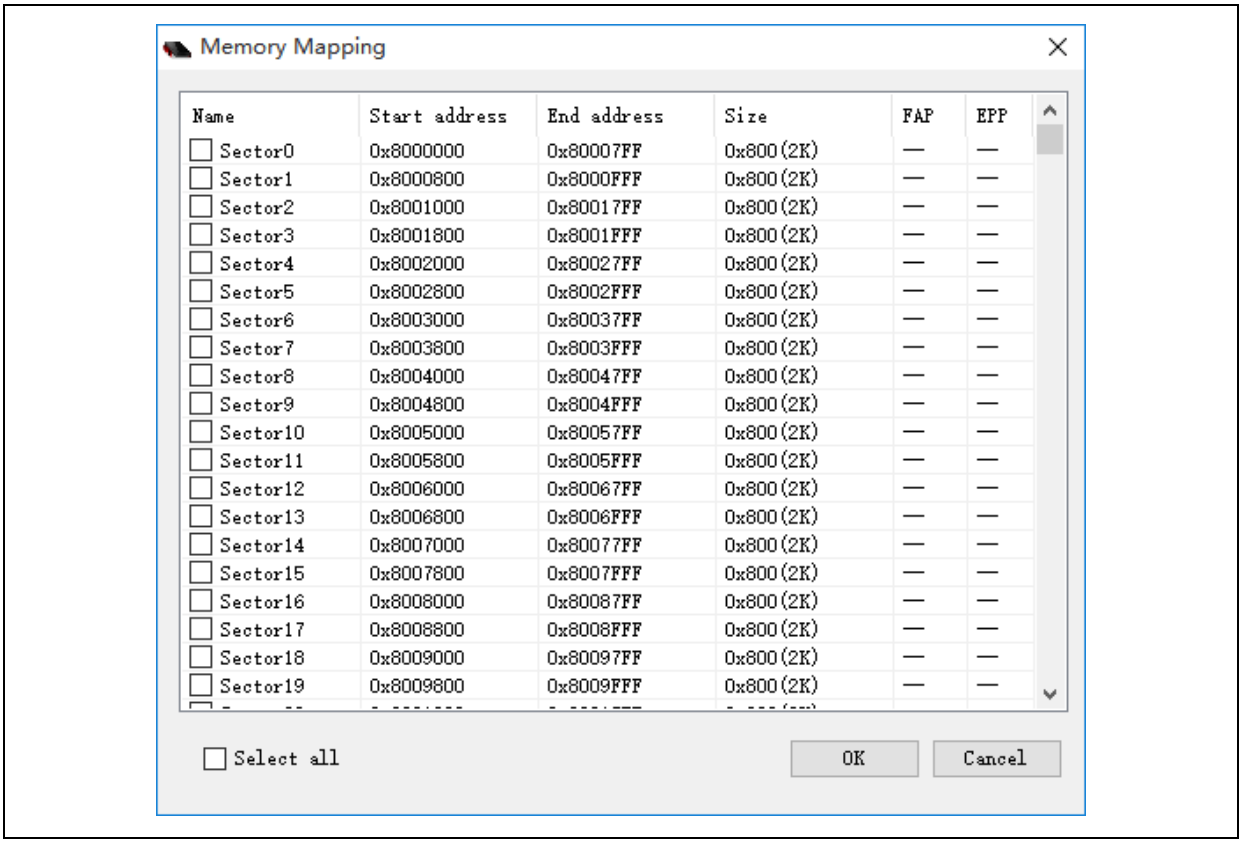
Go to "***Enable***" - "***Access protection***" to enable the flash access protection. The whole flash will be access protected.
AT32F415/421/F425/L021/F423/A423/F402/F405/F490/M412/M416/F455/F456/F457: support enable basic access protection and advanced access protection (access protection and user system data erase protection).
(AT32F425/L021/F423/A423/F402/F405/F490/M412/M416/F455/F456/F457 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 flash access protection. The whole flash memory will be access protected.

Select "***Enable***" - "***Erase and program protection***", and click on "***...***", you can select the sectors to enable erase and program protection in the dialog box that pops up (As shown in Figure 36).

Figure 36. Enable erase and program protection



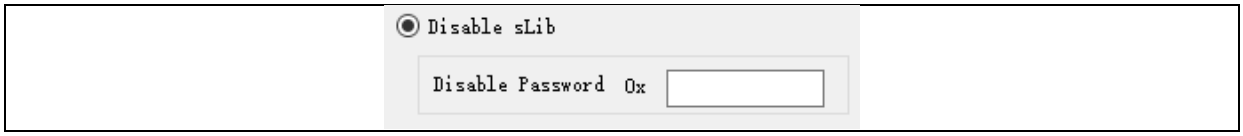
Select all: select all sectors.
OK: make the selected sector take effect.
Cancel: cancel and close this dialog.

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

6.4.7 Disable sLib

Support to disable sLib function separately (AT32F413/415/403A/407/421/435/437/F425/L021 support sLib, as shown in Figure 37)

Figure 37. Disable sLib

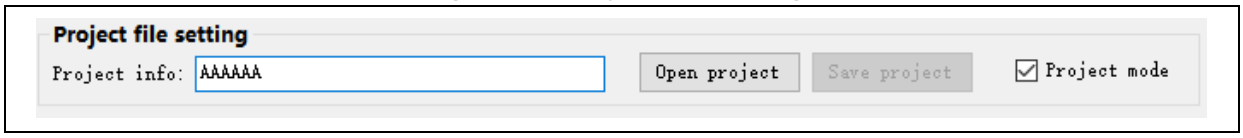


The password must be entered correctly.

7 Project file setting

You can save the download settings and downloaded files in the software as a project file. The project file can be loaded for next use (Project files are encrypted automatically).
(As show in Figure 38)

Figure 38. Project file settings



Project info: it is customized by users.

Save project: save the download settings and download files in the software as a project file.

Open project: Load the saved project file.
After the project file is successfully loaded, the "***Project mode***" is automatically checked, in this case, the parameter settings cannot be changed.

Project mode: in project mode, all settings cannot be changed.
Check this option: enter project mode. The project mode can only be entered via "***Open project***".
Unchecked this option: cancel project mode, and the settings can be changed according to user needs.

8 SPIM encryption Download

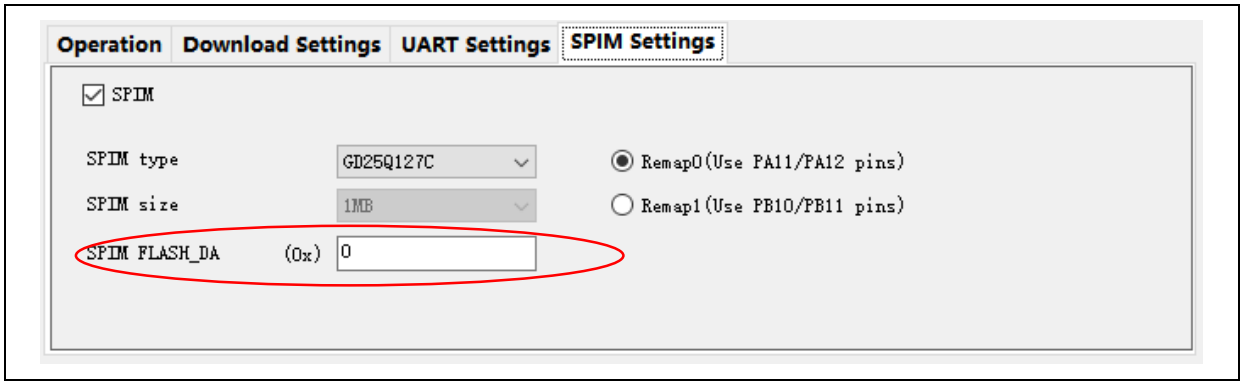
SPIM encryption principle:

When SPIM encryption 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 the download. In this case, the MCU will encrypt the downloaded original data according to the SPIM FLASH_DA and encryption key as well as internal algorithm in the MCU and then write the encrypted data to SPIM. If users want to read the encrypted data of 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 implement encryption download. (AT32F403/F413/F403A/F407/A403A support SPIM)

Step 1: Set *SPIM FLASH_DA* (As shown in Figure 39)

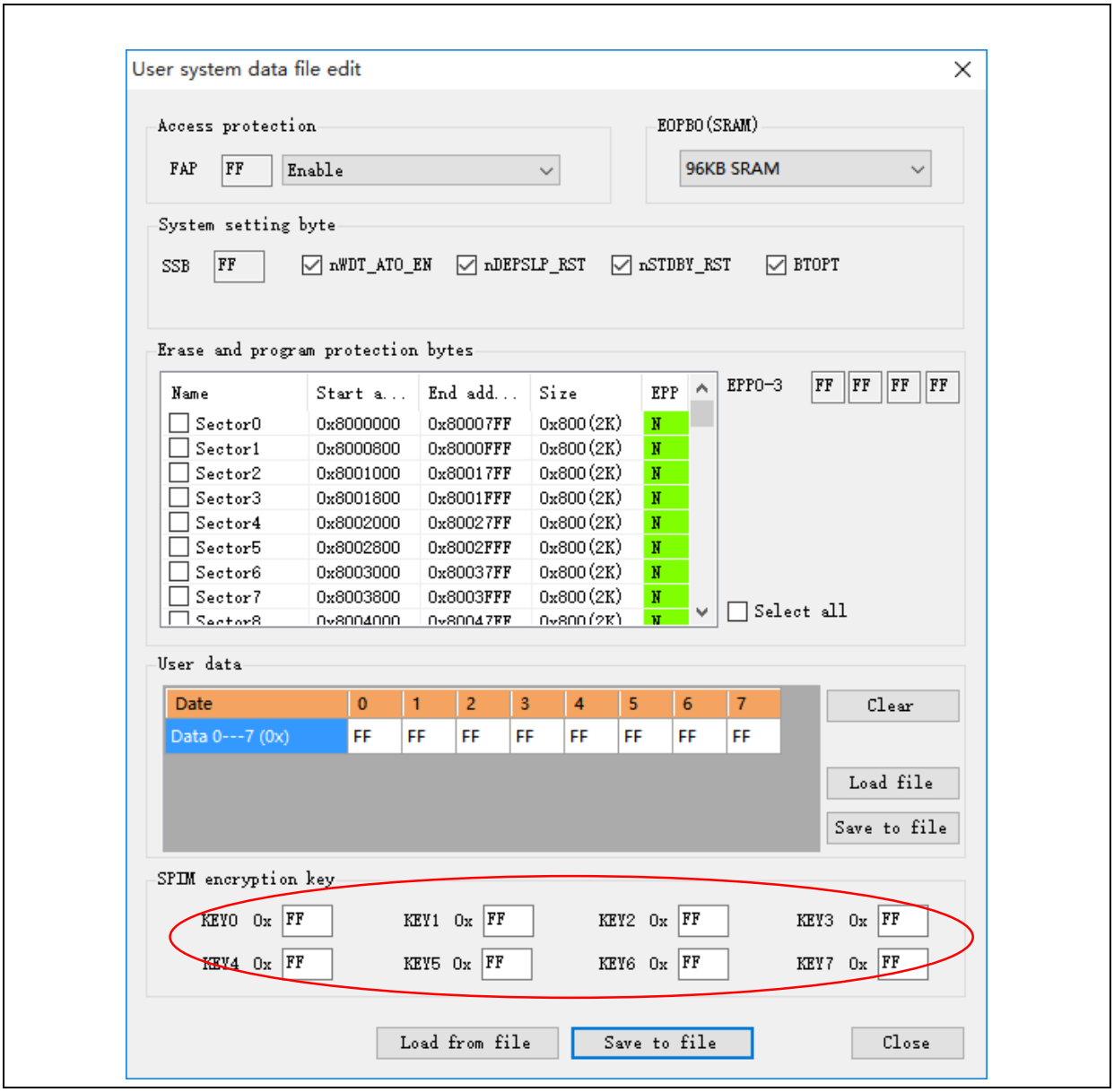
Figure 39. SPIM encryption range config



The encryption range starts from 0x08400000, plus FLASH_DA that is set, which is the encryption area.If encryption is not required, set to 0.

Step 2: Set the SPIM encryption key through "user system data" page. Go to "***User system data file edit***"---"***SPIM encryption key***" (As shown in Figure 40)

Figure 40. 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 file to SPIM to implement encryption download.

9 Revision history

Table 15. Document revision history

Date	Revision	Changes
2025/02/17	V2.13	1. Support for AT32/F455/F456/F457 serial.
2024/10/29	V2.12	1. Support for AT32M412/M416 serial. 2. Added downloading One-Time Programmable data.
2024/04/11	V2.11	1. Support for AT32A423 serial.
2023/08/10	V2.10	1. Support for AT32F423VCW. 2. Support for AT3F402/F405 serial.
2023/07/06	V2.09	1. 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	2. 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.
2021/11/26	V2.01	1. Support for AT32F425 serial. 2. Support for AT32F403AVGW. 3. Support for AT32WB415 serial.
2021/10/13	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.