UM0001

ISP Programmer User manual

Introduction

This user manual gives an overview of Artery ISP Programmer. ISP Programmer is a kind of graphic Interface application to make it easier for users to use Artery MCU. With the help of this programmer, users can perform Artery MCU devices through UART or USB ports.

Contents

List of tables

List of figures

1. Introduction

1.1. Environmental requirements

- **Software prerequisites**

  Windows XP, Windows 7 and above are required.

  .Net framework 4.0 support is required.

- **Hardware prerequisites**

  Serial communication port (COM).

  USB communication port.

1.2 Glossary

- **ISP:**

  In-system programming. For the MCU chip with ISP function, users can directly write or erase program to/from the chip on the circuit board.

- **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**

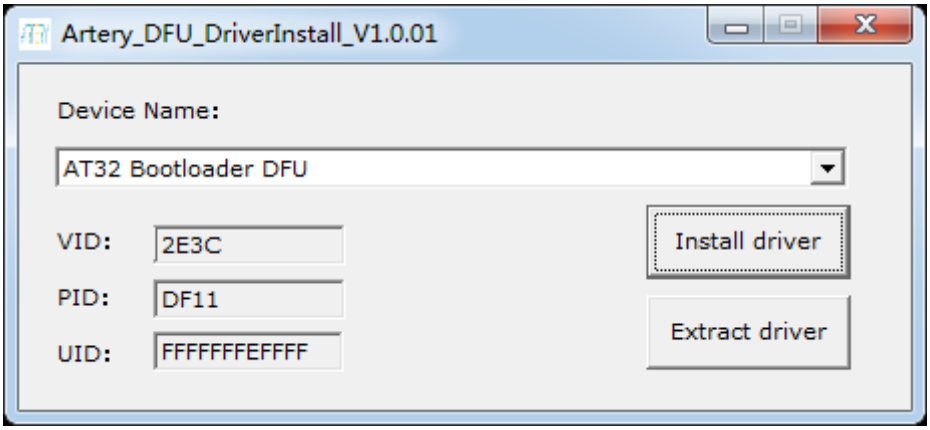
  This software is not required, just directly run the executable program "ArteryISPProgrammer.exe".

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 3-1)

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.

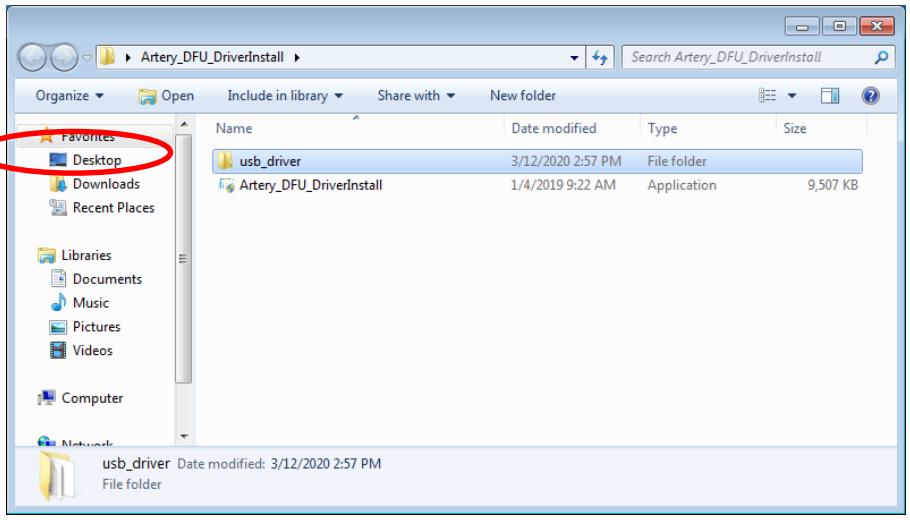
3.2 Install driver manually

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

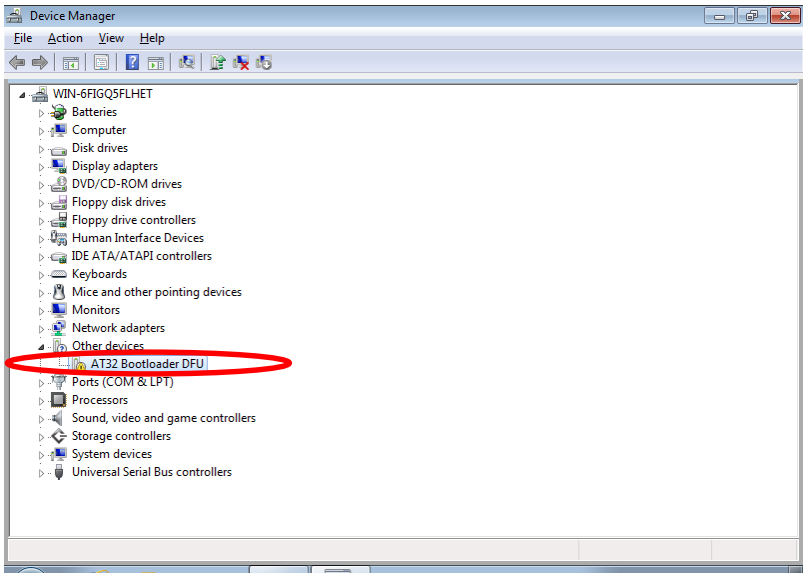


The procedures of manual installation are as follows (take windows7 as an example):

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

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

Figure 3. Manual install-device manager

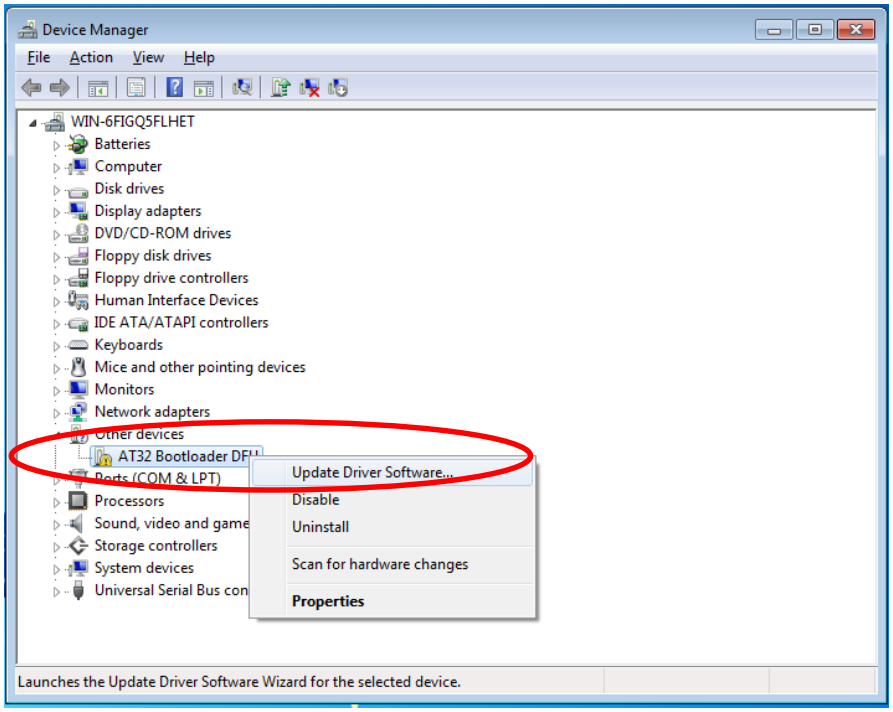


In this case, the "***Device Manager***" will scan the device "AT32 Bootloader DFU" without driver installed.

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***".

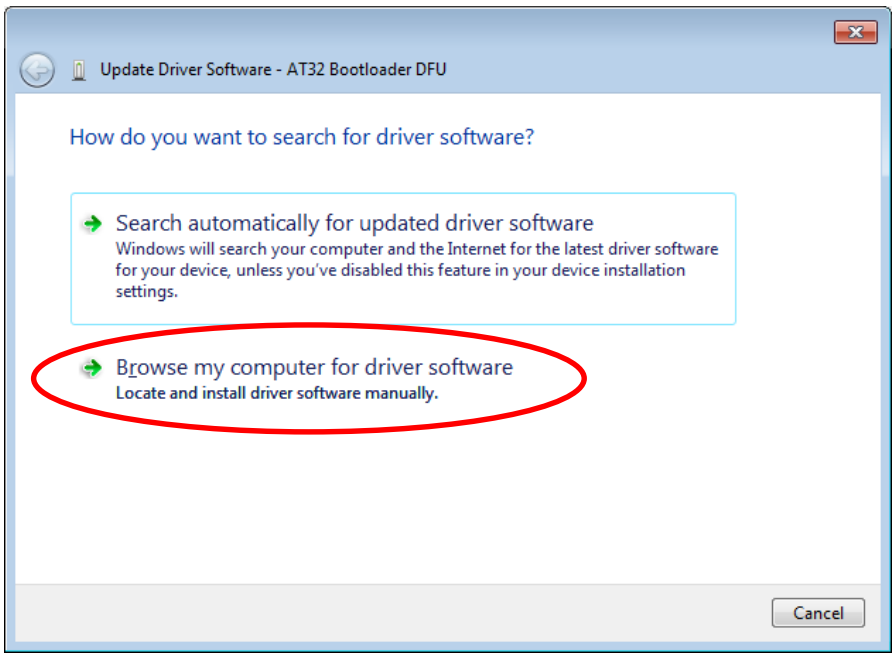
- Right-click on the device "***AT32 Bootloader DFU***" and select "***Update Driver Software***" (As shown in Figure 4).

Figure 4. Manual install-update driver software



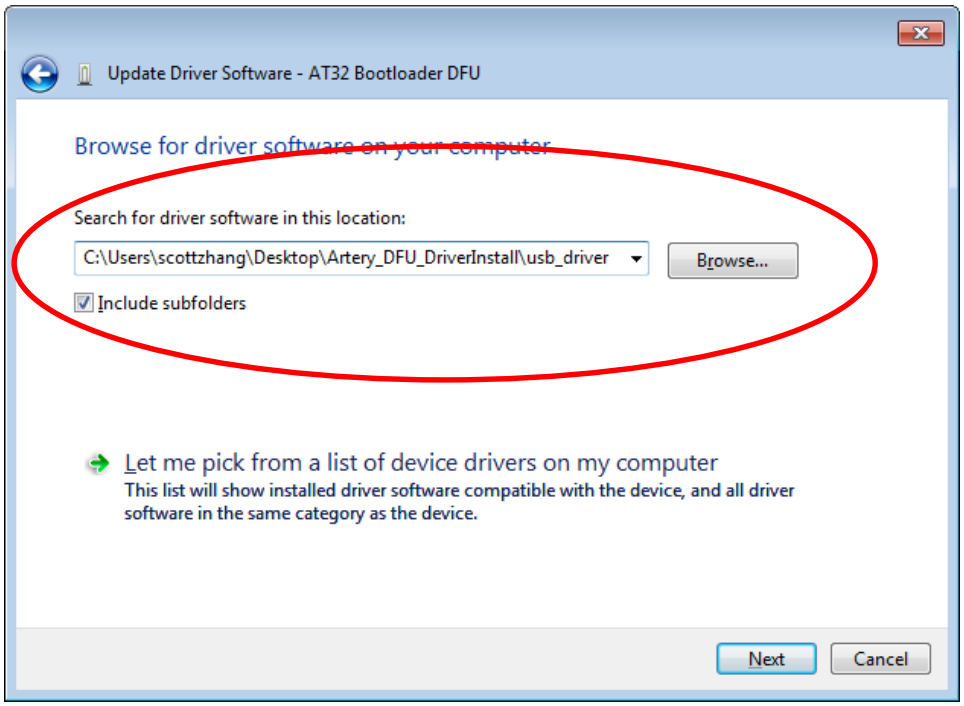
- Select "***Browse my computer for driver software***". (As shown in Figure 5)

Figure 5. Manual install-browse my computer for driver



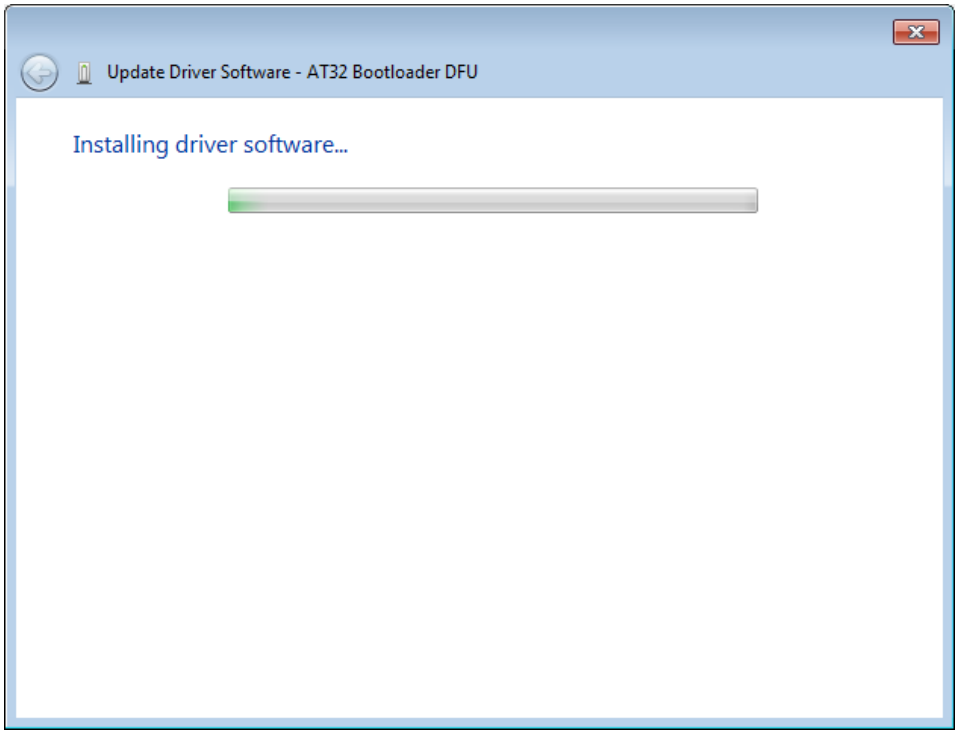
- Please select the location of the driver correctly, that is, click on "***Extract driver***" to generate a driver installation package ("usb_driver" folder). Then click on "***Next***" (As shown in Figure 6).

Figure 6. Manual install-select driver software



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

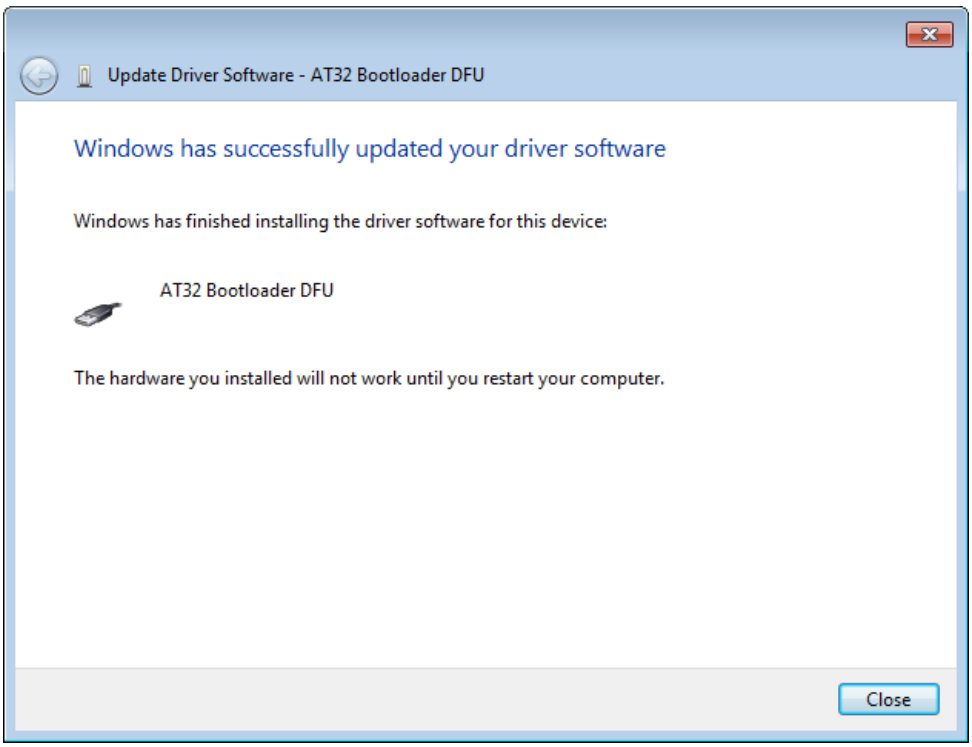
Figure 7. Manual install-driver software installing



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. Product part number and interface

Table 1. Part number and interface

Part number	Flash size	Pin	Interface		
			UART1	UART2	DFU
AT32F403ZCT6	256KB	LQFP144	Y	Y	Y
AT32F403VCT6	256KB	LQFP100	Y	Y	Y
AT32F403RCT6	256KB	LQFP64	Y	Y	Y
AT32F403CCT6	256KB	LQFP48	Y	Y	Y
AT32F403ZGT6	1024KB	LQFP144	Y	Y	Y
AT32F403VGT6	1024KB	LQFP100	Y	Y	Y
AT32F403RGT6	1024KB	LQFP64	Y	Y	Y
AT32F403CGT6	1024KB	LQFP48	Y	Y	Y
AT32F403ZET6	512KB	LQFP144	Y	Y	Y
AT32F403VET6	512KB	LQFP100	Y	Y	Y
AT32F403RET6	512KB	LQFP64	Y	Y	Y
AT32F403CET6	512KB	LQFP48	Y	Y	Y
AT32F403CGU6	1024KB	QFN48	Y	Y	Y
AT32F403CEU6	512KB	QFN48	Y	Y	Y
AT32F403CCU6	256KB	QFN48	Y	Y	Y
AT32F403CBT6	128KB	LQFP48	Y	Y	Y
AT32F413RCT7	256KB	LQFP64	Y	Y	Y
AT32F413CCT7	256KB	LQFP48	Y	Y	Y
AT32F413KCU7-4	256KB	QFN32	Y	Y	Y
AT32F413CCU7	256KB	QFN48	Y	Y	Y
AT32F413RBT7	128KB	LQFP64	Y	Y	Y
AT32F413CBT7	128KB	LQFP48	Y	Y	Y
AT32F413KBU7-4	128KB	QFN32	Y	Y	Y
AT32F413CBU7	128KB	QFN48	Y	Y	Y
AT32F413C8T7	64KB	LQFP48	Y	Y	Y
AT32FEBKC8T7	64KB	LQFP48	Y	Y	Y
AT32F415RCT7	256KB	LQFP64	Y	Y	Y
AT32F415RCW	256KB	LQFP64	Y	Y	Y
AT32F415RCT7-7	256KB	LQFP64	Y	Y	Y
AT32F415CCT7	256KB	LQFP48	Y	Y	Y
AT32F415KCU7-4	256KB	QFN32	Y	Y	Y
AT32F415RBT7	128KB	LQFP64	Y	Y	Y
AT32F415RBW	128KB	LQFP64	Y	Y	Y
AT32F415RBT7-7	128KB	LQFP64	Y	Y	Y

AT32F415CBT7	128KB	LQFP48	Y	Y	Y
AT32F415KBU7-4	128KB	QFN32	Y	Y	Y
AT32F415R8T7	64KB	LQFP64	Y	Y	Y
AT32F415R8T7-7	64KB	LQFP64	Y	Y	Y
AT32F415C8T7	64KB	LQFP48	Y	Y	Y
AT32F415K8U7-4	64KB	QFN32	Y	Y	Y
AT32F415CCU7	256KB	QFN48	Y	Y	Y
AT32F415CBU7	128KB	QFN48	Y	Y	Y
AT32F403AVCT7	256KB	LQFP100	Y	Y	Y
AT32F403ARCT7	256KB	LQFP64	Y	Y	Y
AT32F403ACCT7	256KB	LQFP48	Y	Y	Y
AT32F403ACCU7	256KB	QFN48	Y	Y	Y
AT32F403AVET7	512KB	LQFP100	Y	Y	Y
AT32F403ARET7	512KB	LQFP64	Y	Y	Y
AT32F403ACET7	512KB	LQFP48	Y	Y	Y
AT32F403ACEU7	512KB	QFN48	Y	Y	Y
AT32F403AVGT7	1024KB	LQFP100	Y	Y	Y
AT32F403ARGT7	1024KB	LQFP64	Y	Y	Y
AT32F403ACGT7	1024KB	LQFP48	Y	Y	Y
AT32F403ACGU7	1024KB	QFN48	Y	Y	Y
AT32F407VCT7	256KB	LQFP100	Y	Y	Y
AT32F407RCT7	256KB	LQFP64	Y	Y	Y
AT32F407VET7	512KB	LQFP100	Y	Y	Y
AT32F407RET7	512KB	LQFP64	Y	Y	Y
AT32F407VGT7	1024KB	LQFP100	Y	Y	Y
AT32F407RGT7	1024KB	LQFP64	Y	Y	Y
AT32F407AVCT7	256KB	LQFP100	Y	Y	Y
AT32F407AVGT7	1024KB	LQFP100	Y	Y	Y
AT32F421C8T7	64KB	LQFP48	Y	Y	N
AT32F421C8W	64KB	LQFP48	Y	Y	N
AT32F421K8T7	64KB	LQFP32	Y	Y	N
AT32F421K8U7	64KB	QFN32	Y	Y	N
AT32F421K8U7-4	64KB	QFN32	Y	Y	N
AT32F421F8U7	64KB	QFN20	Y	Y	N
AT32F421F8P7	64KB	TSSOP20	Y	Y	N
AT32F421PF8P7	64KB	TSSOP20	Y	Y	N
AT32F421G8U7	64KB	QFN28	Y	Y	N
AT32F421C6T7	32KB	LQFP48	Y	Y	N
AT32F421K6T7	32KB	LQFP32	Y	Y	N

AT32F421K6U7	32KB	QFN32	Y	Y	N
AT32F421K6U7-4	32KB	QFN32	Y	Y	N
AT32F421F6U7	32KB	QFN20	Y	Y	N
AT32F421F6P7	32KB	TSSOP20	Y	Y	N
AT32F421G6U7	32KB	QFN28	Y	Y	N
AT32F421C4T7	16KB	LQFP48	Y	Y	N
AT32F421K4T7	16KB	LQFP32	Y	Y	N
AT32F421K4U7	16KB	QFN32	Y	Y	N
AT32F421K4U7-4	16KB	QFN32	Y	Y	N
AT32F421F4U7	16KB	QFN20	Y	Y	N
AT32F421F4P7	16KB	TSSOP20	Y	Y	N
AT32F421PF4P7	16KB	TSSOP20	Y	Y	N
AT32F421G4U7	16KB	QFN28	Y	Y	N

Table 2. GPIO Pin Map

IP	TX Pin/DM	RX Pin/DP
UART1	PA9	PA10
UART2 (Non-G Series LQFP144& LQFP100)	PA2	PA3
UART2 (G Series LQFP144& LQFP100)	PD5	PD6
DFU	PA11	PA12

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.

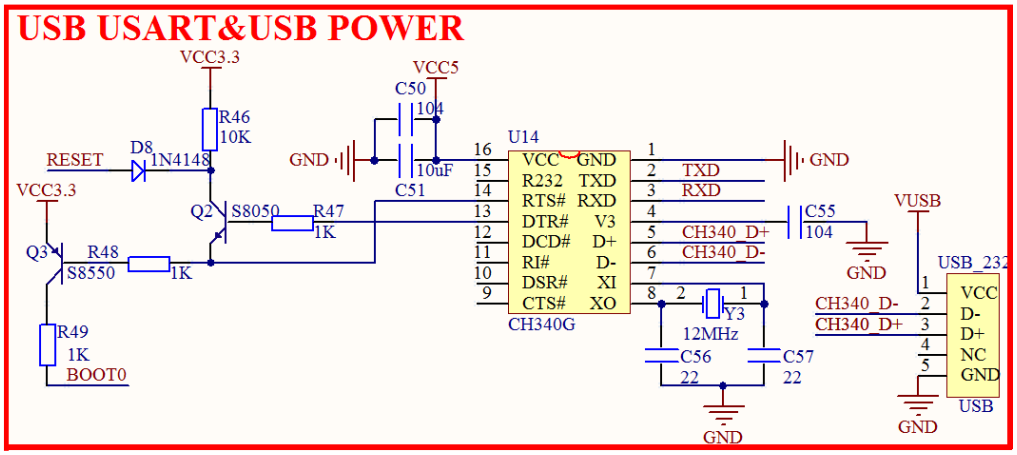
When "Boot Switch" is set to "Manual", you need to manually reset the device to restart the "BootLoader" program in device. If the device supports automatic connection circuitry, reset can be controlled by controlling the DTR and RTS signals. You can select the control mode of the current device in "Boot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

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 5-2 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 "***Read 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 Read protection" function.

Figure 12. Flash status window

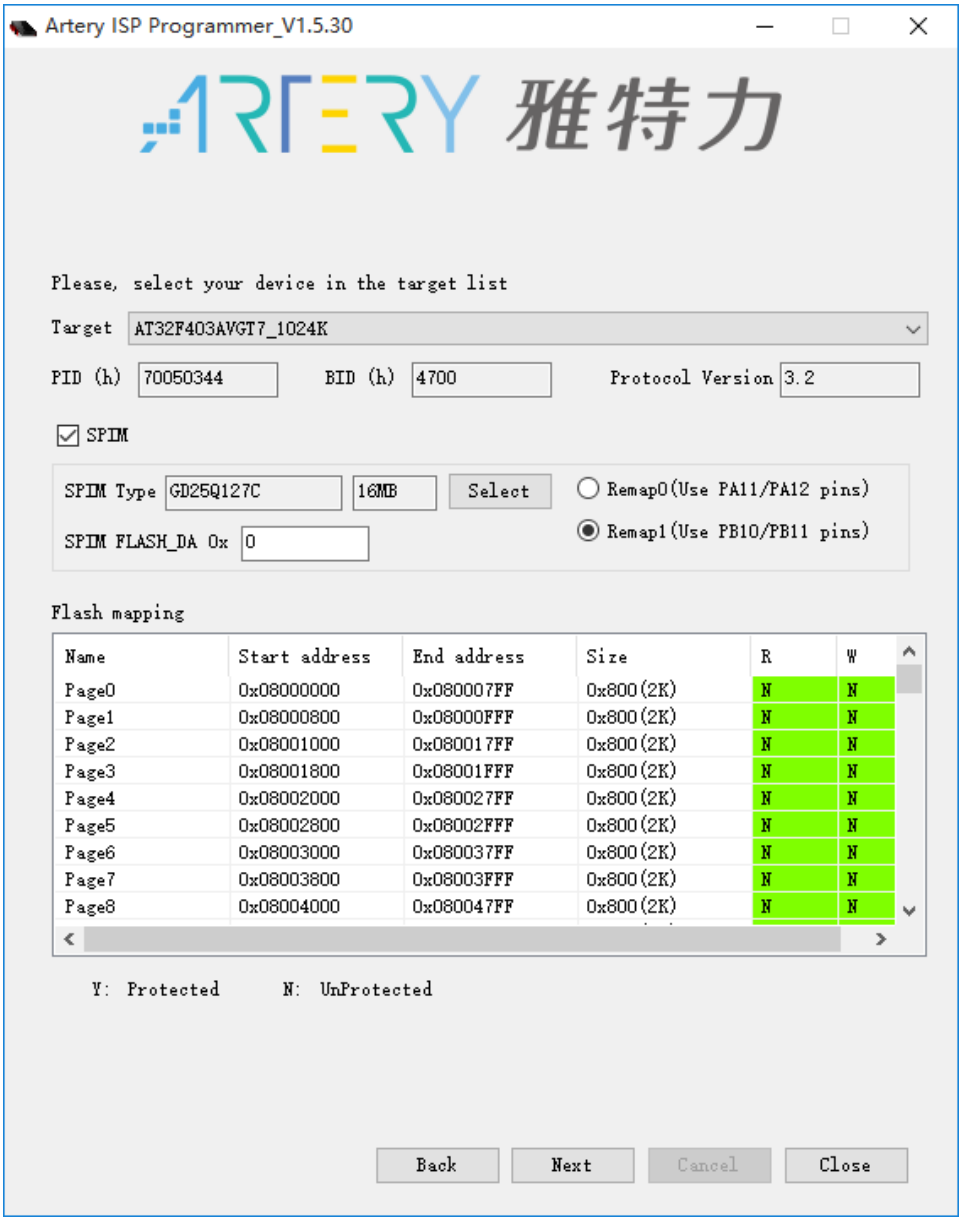
5.3 Device Information Page

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 pages of main flash and SPIM are automatically displayed in the Flash map.

Figure 13. Device information



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.

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.

- Checked "***SPIM***"

  Allows operation on SPIM.

- Unchecked "***SPIM***"

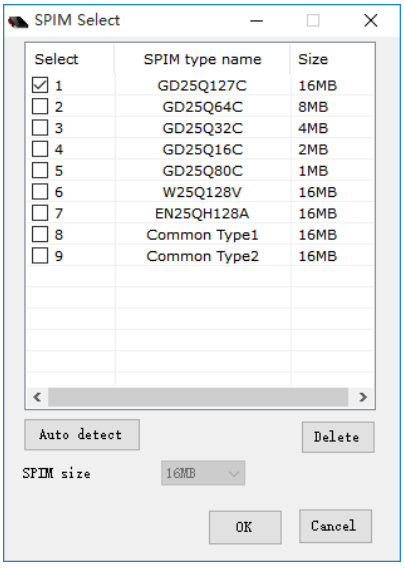
  Operation on SPIM is not allowed.

- 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)

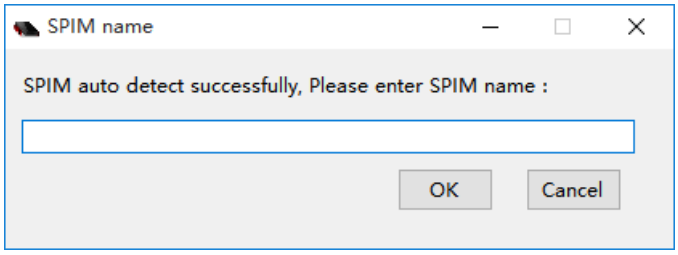
Figure 14. SPIM selection



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)

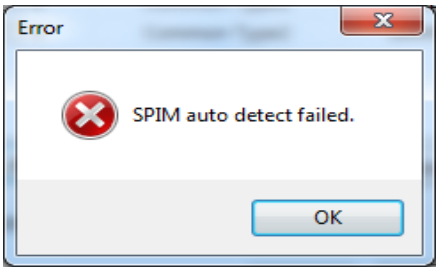
Figure 15. SPIM name



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.

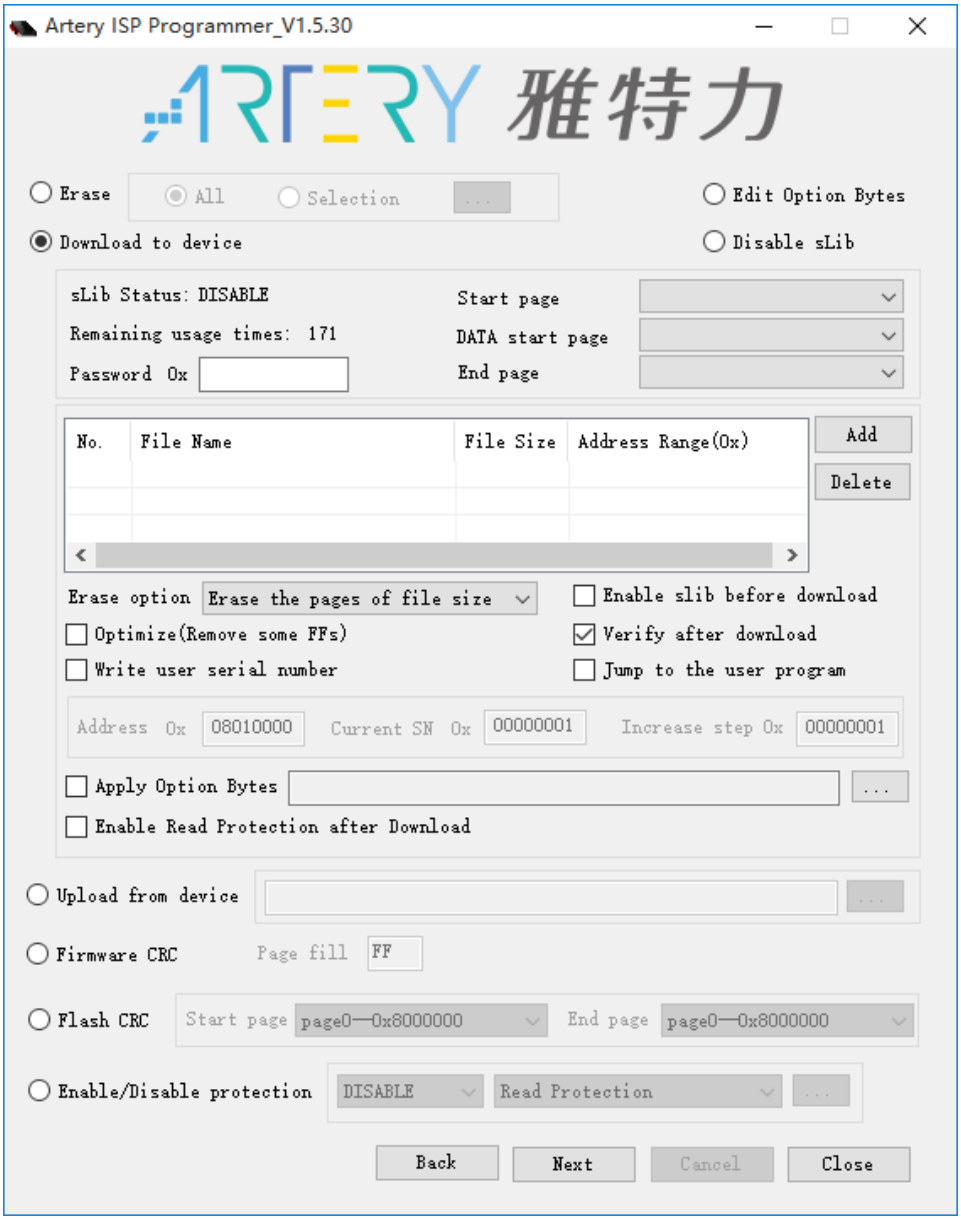
- SPIM FLASH_DA
  Set the encryption range when downloading files to the SPIM. The encryption range calculates starting from address 0x08400000.

- Remap0 (use PA11/PA12 pins)
  Remap1 (use PB10/PB11 pins)
  Select the desired pins. This option is only available for AT32F413/403/407 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 config

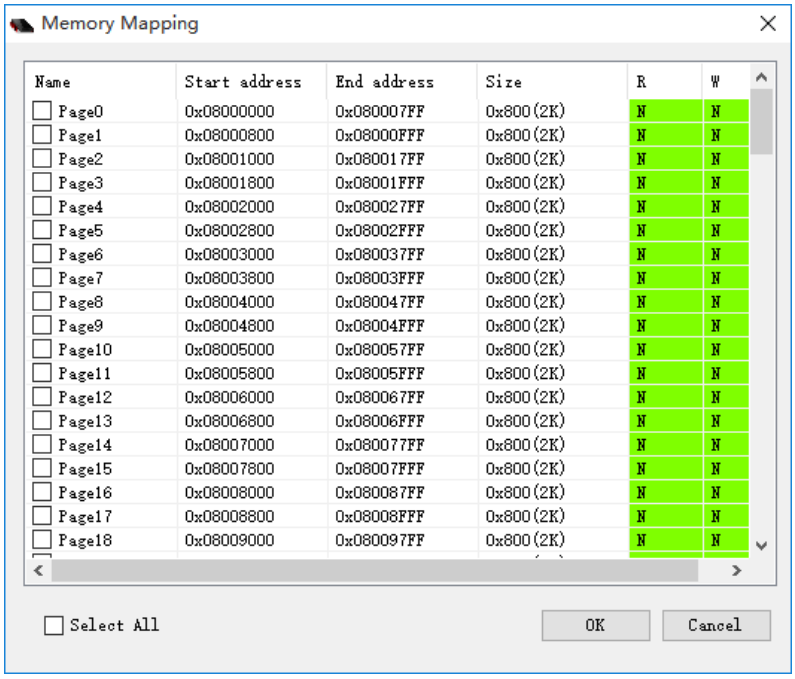


5.4.1 Erase

- Click on "***All***" to erase the whole memory (Including SPIM).

- Click on "***Selection***" to customize the pages to be erased. At this time, click on "**…**" to select the page to be erased in the pop-up dialog box. (As shown in Figure 18)

Figure 18. Page erase selection

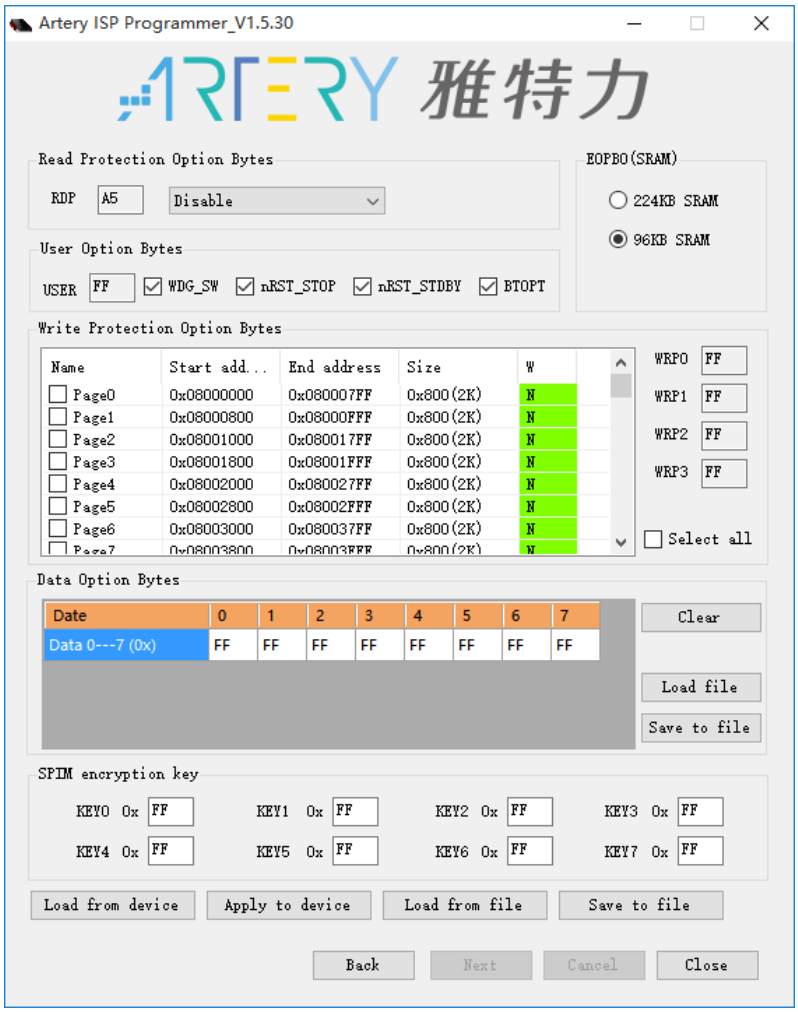


5.4.2 Edit Option Bytes

Select "***Edit Option Bytes***" and click on "***Next***".

On this page, users can configure the "Option bytes" through graphical interface (As shown in Figure 19).
Supports obtaining the "Option bytes" value from the device or file and displaying the value.
After editing, apply to device or save to file.

Figure 19. Option bytes page



- Read Protection Option Bytes

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

  AT32F403/413/403A/407:

  Enabled: RDP----0xFF.

  Disabled: RDP---0xA5.

  AT32F415/421:

  Common read protection: RDP----0xFF.

  Advanced read protection: RDP----0xCC (read protection and option bytes erase protection).

  Disabled: RDP----0xA5.

When read protection is enabled, neither the Flash memory or option bytes can be read, unless the read protection is disabled.

After read protection is disabled, both the main Flash and option bytes will be erased.

- User Option Bytes

WDG_SW:

Unchecked—Hardware watchdog.

Checked—Software watchdog.

nRST_STOP:

Unchecked—Reset occurs when entering STOP mode.

Checked—No reset occurs when entering STOP mode.

nRST_STDBY:

Unchecked—Reset occurs when entering Standby mode.

Checked—No reset occurs when entering Standby mode.

BTOPT (AT32F403/413/403A/407)

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)

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

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

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

- 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.

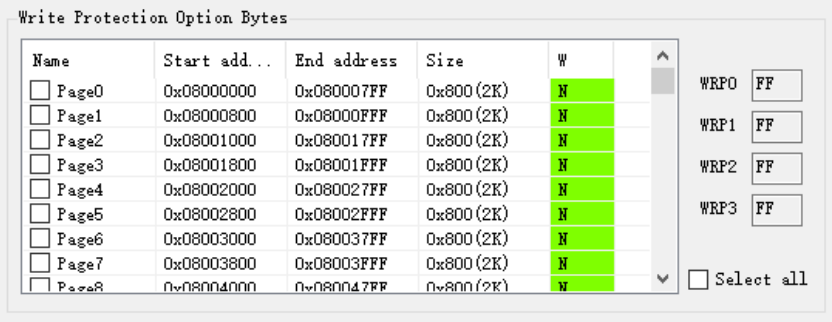
16 KB SRAM—SRAM 16 KB.

AT32F415/421: (not support)

- Write Protection Option Bytes

  You can choose which pages need to be write protected. (As shown in Figure 20)

Figure 20. Write option bytes



WRP0:

AT32F403/413/403A/407: controls the write protection of pages in the range of Flash 0K-32K.

AT32F415: controls the write protection of Page0-Page15.

AT32F421: controls the write protection of Page0-Page31.

WRP1:

AT32F403/413/403A/407: controls the write protection of pages in the range of Flash 32K-64K.

AT32F415: controls the write protection of Page16-Page31.

AT32F421: controls the write protection of Page32-Page63.

WRP2:

AT32F403/413/403A/407: controls the write protection of pages in the range of Flash 64K-96K.

AT32F415: controls the write protection of Page32-Page47.

WRP3:

AT32F403/413/403A/407:

Bit 0-6 controls the write protection of pages in the range of 96K-124K;

Bit 7 controls the write protection of all pages after Flash 124K, including SPIM.

AT32F415:

Bits 0-6 control the write protection of Page48-Page61;

Bit 7 controls the write protection of all subsequent pages, including system memory (in system memory AP mode).

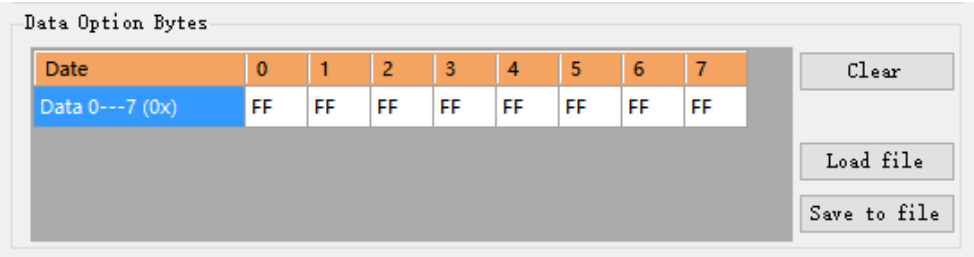
AT32F421:

Bit 7 controls the system memory area (system memory area in AP mode)

(For Flash 256 KB and above, each page size is 2048 bytes (2 KB), whereas for Flash less than 256 KB, each page size is 1024 bytes (1 KB))

- Data Option Bytes

Figure 21. Data option bytes



(AT32F403/413/403A/407 data option bytes: 8 bytes; AT32F415: 10 bytes; AT32F421: 250 bytes)

Clear: Reset all data option bytes to 0xFF, which is not saved to the device

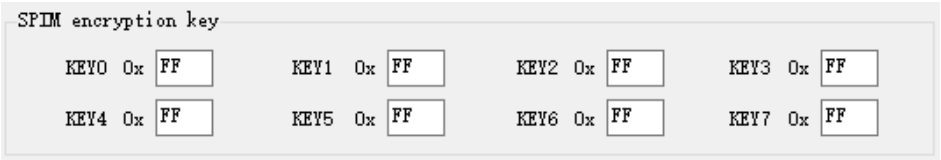
Load file: Load the data option bytes file into the table for display

Save to file: Save the data option bytes in the table to the file.

- SPIM encryption key (AT32F415/421 not support)

You can set the encryption key when downloading the SPIM. (As shown in Figure 22)

Figure 22. SPIM encryption key



- Load from device

Read the option bytes from the device and update it to the interface for display.

- Apply to device

Save the settings of the option bytes to the device.

- Load from file

Read the content of option bytes from option bytes and update it to the interface for display.

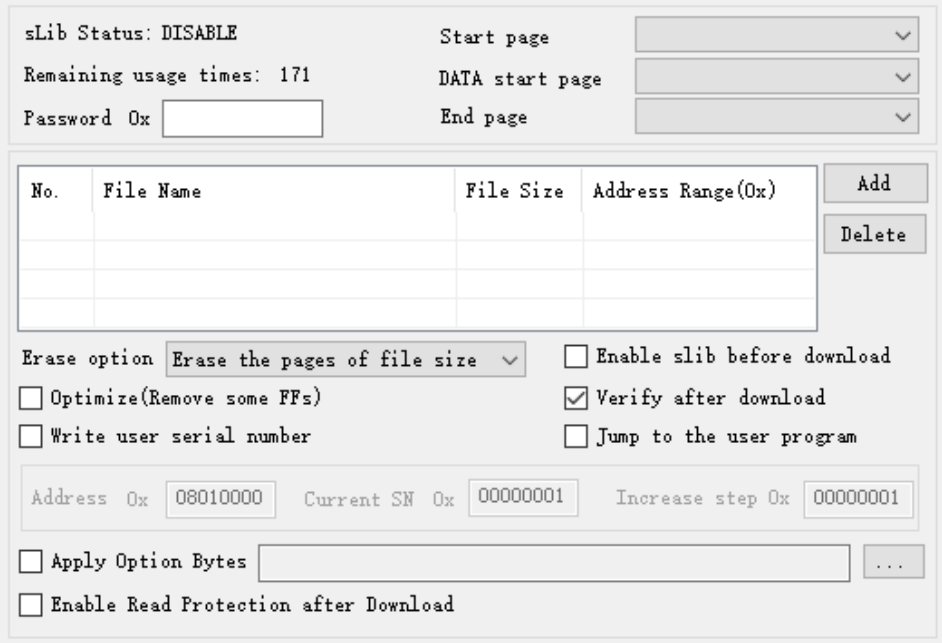
- Save to file

Save the option bytes settings to a file.

5.4.3 Download to device

(As show in Figure 23)

Figure 23. Download to device



1) sLib settings

(AT32F413/415/403A/407/421 support sLib)

- sLib status

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

- Remaining usage times (AT32F413/403A/407)

  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 page

  AT32F413/415/403A/407:

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

  AT32F421:

  The start page of sLib area。The area from “Start page” to “INSTR start page” (not including “INSTR start page”) 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 page/INSTR start page

  AT32F413/415/403A/407：

  The start page of the sLib data area. This data area is from "DATA start page" to "End page"(including "End page"). 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:

  The start page of sLib instruction area. The instruction area is from "INSTR start page" to "End page" (including "End page"). 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.

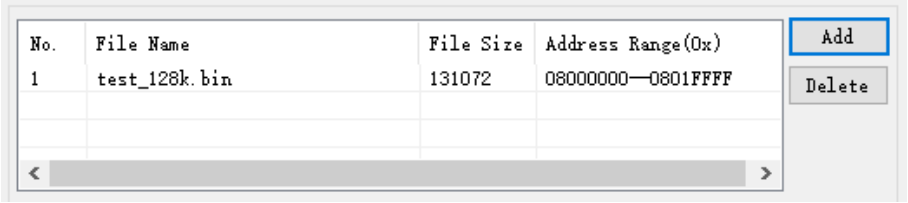
- End page

  The end position of the sLib area.

2) Other download settings

- Three file types are supported: bin (binary), hex (hexadecimal), and s19 / srec (Motorola S file). (As shown in Figure 24)

Figure 24. 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 pages of file size***" to erase pages 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 page, start data page, and end page 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

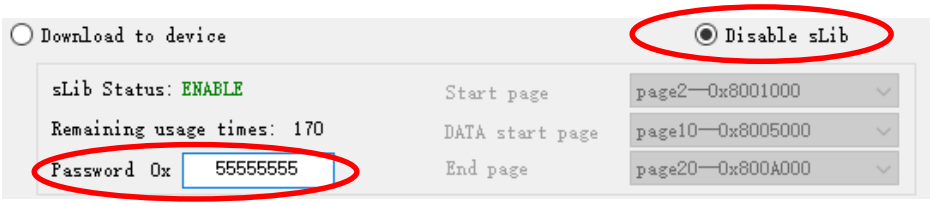
- Check "***Apply Option Byte***s", load the option bytes file after download, and set the value to the device.

- Check "***Enable Read Protection after Download***" to enable read protection after download.
For AT32F415/421, you can enable common read protection and advanced read protection (Read protection and option bytes erase protection).

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 25):

Figure 25. Disable sLib



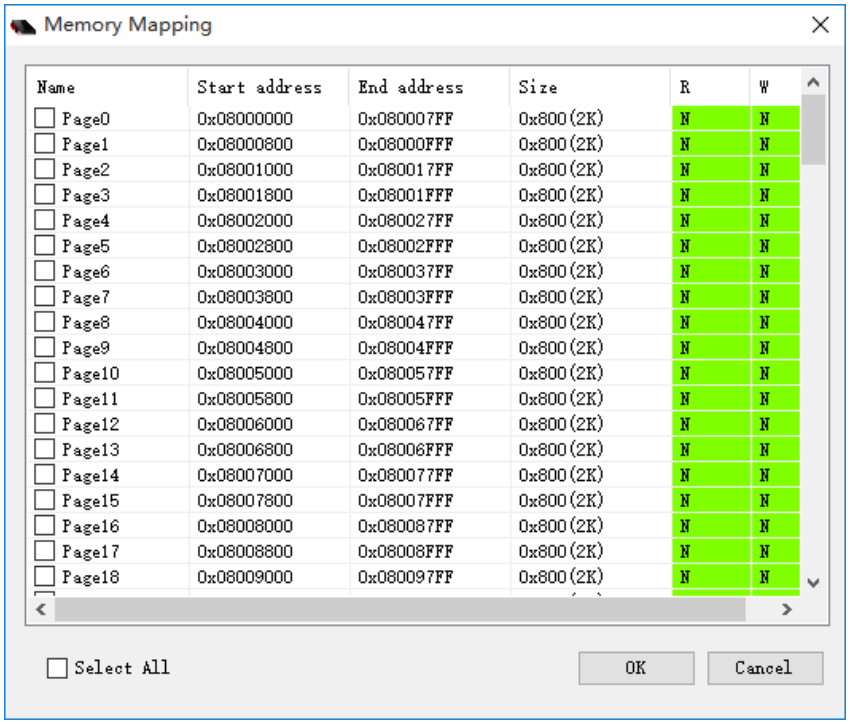
When disabling sLib successfully, the whole chip will be erased.

5.4.5 Upload from device

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

Select the upload pages. (As shown in Figure 5-18)

Figure 26. Upload from device

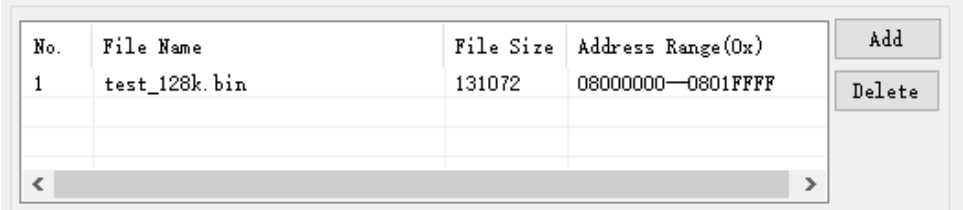


5.4.6 Firmware CRC

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 read protection state).

- First you need to select the file to be compared. (As shown in Figure 27)

Figure 27.Firmware CRC



- "Page fill": the Firmware CRC is performed in units of pages. What is filled in here is the download data that is not filled in the page part. Generally, it is "FF".

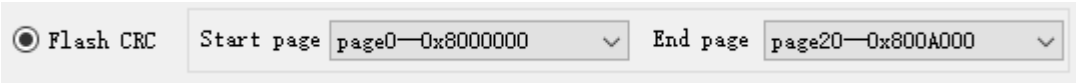
5.4.7 Flash CRC

This function is used to calculate CRC value, including main Flash and SPIM.

(This function can be used in the Flash read protection state)

(As shown in 28):

Figure 28.Flash CRC



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

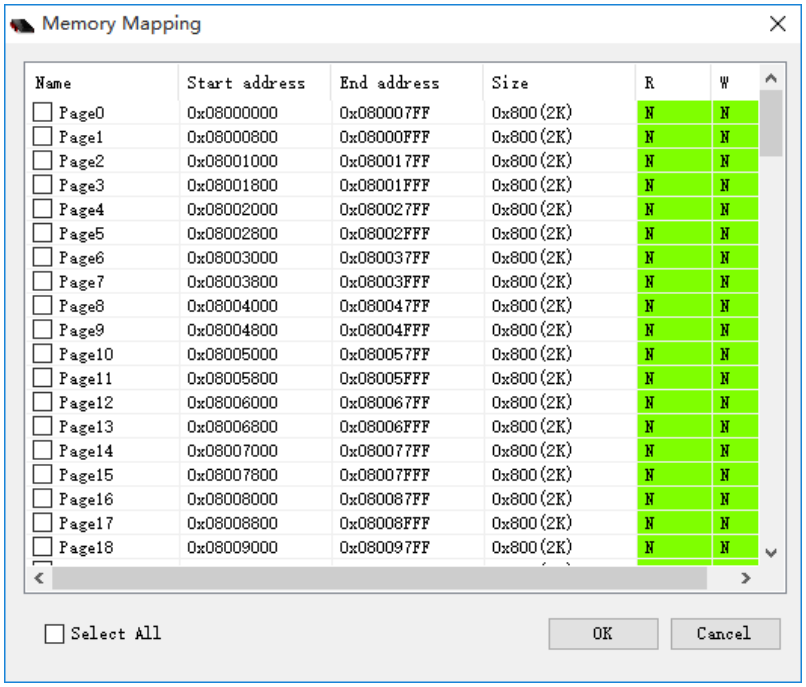
5.4.8 Enable/Disable protection

- Select "***Enable***" - "***Read Protection***" to enable the Flash read protection. The whole Flash will be read Protected.
  AT32F415/421: enable common read protection and advanced read protection (Read protection and option bytes erase protection).

- Select "***Disable***" - "***Read Protection***" to disable the read protection of the whole Flash.

- Select "***Enable***" - " ***Write Protection***", and click **"..."** , you can select the pages to enable write protection in the dialog box that pops up. (As shown in Figure 29)

Figure 29. Enable write protection



- Select "***Disable***" - "***Write Protection***" to disable the write protection of the whole Flash.

5.5 Operation progress page

This page displays information related to the operation progress. (As shown in Figure 30)

Figure 30. Operation progress display

5.6 SPIM encryption download

SPIM encryption principle:

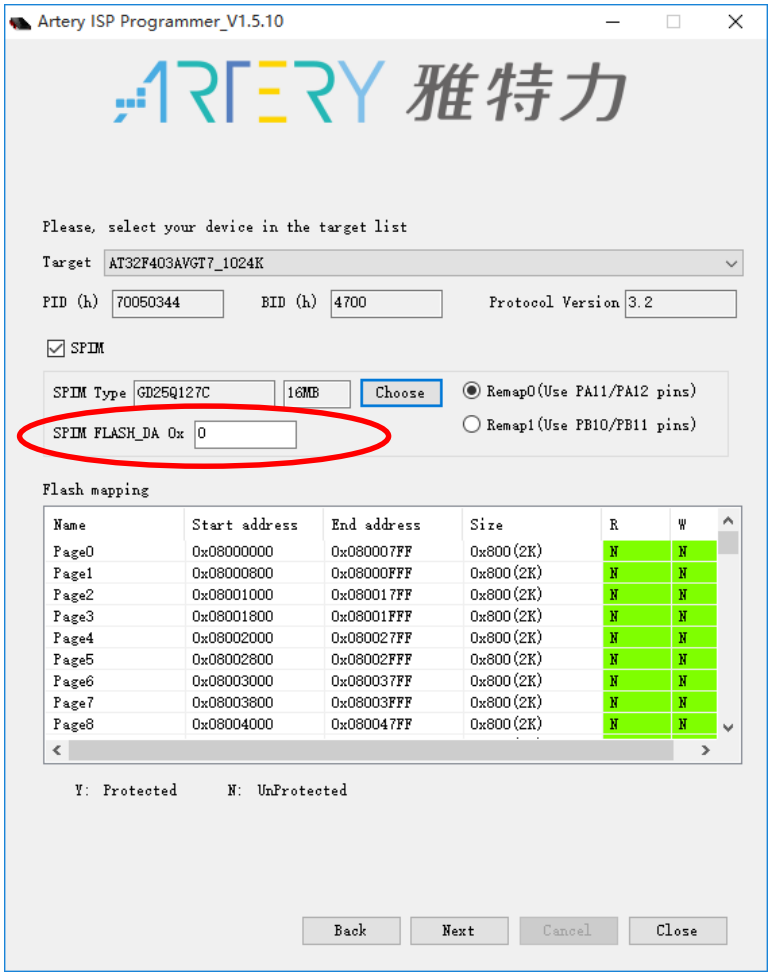
When SPIM encrypted download is required, users must first configure the SPIM FLASH_DA and SPIM encryption key (Key is set in the option bytes), and then perform download operation. In this case, the MCU will encrypt the downloaded original data according to SPIM FLASH_DA and encryption key as well as 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 (AT32F415/421 not support SPIM)

Step 1: set the ***SPIM FLASH_DA*** (As shown in figure 31).

Figure 31. Encryption range config

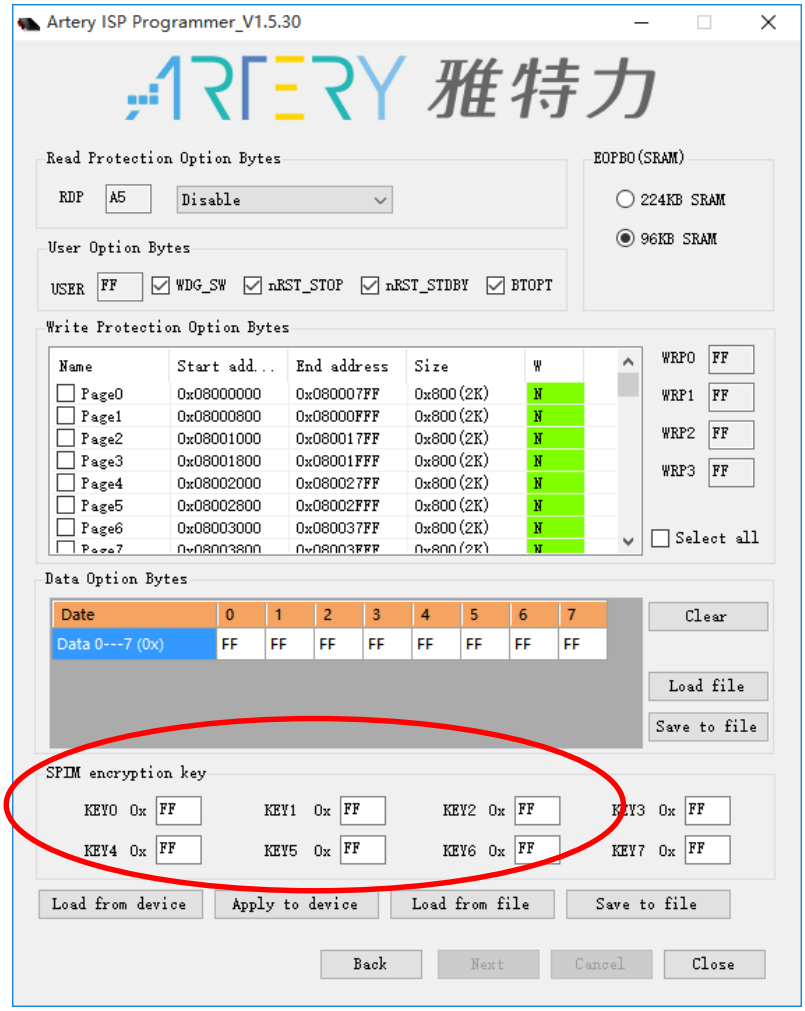


Starting from the address 0x08400000, plus the set FLASH_DA, it is the encryption area.

If encryption is not required, set to 0.

Step 2: set the SPIM encryption key through the "option bytes" page. (As shown in figure 32)

Figure 32. SPIM encryption key config



This is the encryption / decryption key for downloading and reading data in the encryption range of SPIM.

When the read protection is disabled, the key is also erased.

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

6. Revision history

Table 3. Document revision history

Date	Revision	Changes
2017/03/02	V1.00	1. Initial release.
2017/11/30	V1.10	1. Changed the description of some contents. 2. Added the description and interface of Firmware CRC function. 3. Added description and interface of related functions of SPIM.
2018/01/17	V1.20	1. Software name adjustment. 2. Updated the software interface and adjust the name and text. 3. Added SPIM selection. 4. Added SPIM automatic detection function description. 5. Added SPIM encryption download function description. 6. Added SPIM encryption key setting in option bytes, and add the option bytes description. 7. Removed the manual SPIM enable function and change it to automatic enable.
2018/03/19	V1.30	1. Added DFU communication related description. 2. Added DFU driver installation. 3. Updated some pictures.
2018/05/11	V1.35	1. Added device and interface support list. 2. The option bytes EOPB0 setting is changed. 3. Updated some descriptions.
2018/06/27	V1.36	1. "Download to device" added "Enable Read Protection after Download" function description.
2018/11/29	V1.37	1. Added AT32F413 series and UART SPIM Pin remap instructions. 2. Added option bytes -EOPB0, AT32F413 MCU SRAM description. 3. Added multi-file download UI and description.
2019/01/03	V1.38	1. Changed part of the model list. 2. Updated some pictures.
2019/02/18	V1.39	1. Added description of sLib.
2019/03/21	V1.40	1. Updated some pictures.
2019/04/16	V1.41	1. Added support for AT32F413CBU7 and AT32FEBKC8T7.
2019/06/26	V1.42	1. Added support for AT32F415 MCU.
2019/08/09	V1.43	1. Updated AT32F415 read protection name and description. 2. Added the principle of encrypted download of SPIM.
2019/10/12	V1.44	1. Added support for AT32F415CCU7 and AT32F415CBU7.
2019/11/29	V1.50	1. Added support for AT32F403A MCU. 2. Added support for AT32F407 MCU.
2020/03/06	V1.51	1. Updated SPIM Remap descriptions and pictures.
2020/5/15	V1.52	Adjusted the document formats.

2020/8/12	V1.60	1 .Added AT32F421. 2. Added “Flash CRC” function. 3. Added the description of data option bytes.
2020/10/20	V1.61	1. Added AT32F421PF4P7 and AT32F421PF8P7 2. Added AT32F407AVCT7 and AT32F407AVGT7.
2020/11/25	V1.62	1. Added AT32F421C8W 2. Added AT32F415RCW and AT32F415RBW 3. Updated the document format.

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.

Artery's products and services are provided "AS IS" and Artery provides no warranties express, implied or statutory, including, without limitation, any implied warranties of merchantability, satisfactory quality, non-infringement, or fitness for a particular purpose with respect to the Artery's products and services.

Notwithstanding anything to the contrary, purchasers acquires no right, title or interest in any Artery's products and services or any intellectual property rights embodied therein. In no event shall Artery's products and services provided be construed as (a) granting purchasers, expressly or by implication, estoppel or otherwise, a license to use third party's products and services; or (b) licensing the third parties' intellectual property rights; or (c) warranting the third party's products and services and its intellectual property rights.

Purchasers hereby agrees that Artery's products are not authorized for use as, and purchasers shall not integrate, promote, sell or otherwise transfer any Artery's product to any customer or end user for use as critical components in (a) any medical, life saving or life support device or system, or (b) any safety device or system in any automotive application and mechanism (including but not limited to automotive brake or airbag systems), or (c) any nuclear facilities, or (d) any air traffic control device, application or system, or (e) any weapons device, application or system, or (f) any other device, application or system where it is reasonably foreseeable that failure of the Artery's products as used in such device, application or system would lead to death, bodily injury or catastrophic property damage.