

Introduction

This user manual gives an overview of Artery ISP Console. ISP Console is a command-line application based on MCU Bootloader. With the help of this software, users can configure ARTERY MCU devices through UART or USB ports.

Contents

1	Introduction	6
1.1	Environmental requirements.....	6
1.2	Glossary	6
2	Installation	7
3	interfaces	8
3.1	AT32F403 interfaces.....	8
3.2	AT32F413 interfaces.....	8
3.3	AT32F415 interfaces.....	8
3.4	AT32F403A/F407 interfaces.....	8
3.5	AT32F421 interfaces.....	9
3.6	AT32F435/F437 interfaces	9
3.7	AT32WB415 interfaces	9
3.8	AT32F425 interfaces.....	10
3.9	AT32L021 interfaces.....	10
3.10	AT32F423 interfaces.....	10
3.11	AT32A403A interfaces	10
3.12	AT32F402/F405 interfaces	11
3.13	AT32A423 interfaces	11
3.14	AT32M412/M416 interfaces.....	12
3.15	AT32F455/F456/F457 interfaces.....	12
4	Software operation	14
4.1	Operating mode	14
4.1.1	Used in Windows.....	14
4.1.2	Used in Linux.....	15
4.2	Command line parameters	16
4.3	ISP Console return codes.....	18
4.4	Flow chart.....	19

5 Revision history20

List of tables

Table 1. AT32F403 GPIO Pin Map.....	8
Table 2. AT32F413 GPIO Pin Map.....	8
Table 3. AT32F415 GPIO Pin Map.....	8
Table 4. AT32F403A/F407 GPIO Pin Map	8
Table 5. AT32F421 GPIO Pin Map.....	9
Table 6. AT32F435/F437 GPIO Pin Map	9
Table 7. AT32WB415 GPIO Pin Map	9
Table 8. AT32F425 GPIO Pin Map.....	10
Table 9. AT32L021 GPIO Pin Map.....	10
Table 10. AT32F423 GPIO Pin Map.....	10
Table 11. AT32A403A GPIO Pin Map	10
Table 12. AT32F402/F405 GPIO Pin Map	11
Table 13. AT32A423 GPIO Pin Map	11
Table 14. AT32M412/M416 GPIO Pin Map.....	12
Table 15. AT322F455/F456/F457 GPIO Pin Map	12
Table 16 List of command line parameters.....	16
Table 17 List of return codes.....	18
Table 18. Document revision history.....	20

List of figures

Figure- 1 Command line window	14
Figure- 2 Batch file processing	14
Figure- 3 Linux system.....	15
Figure- 4 Shell script file	15
Figure- 5 Flow chart.....	19

1 Introduction

1.1 Environmental requirements

- **Software resources**

Windows OS

Windows 7 and above

Linux OS

Ubuntu, Fedora

- **Hardware resources**

Serial communication port (COM).

USB communication port.

1.2 Glossary

- **ISP:**

ISP is the abbreviation for In-System Programming. It enables users to directly perform write or erase operations on the chip with an ISP feature.

- **UART:**

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

- **USB:**

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

- **DFU:**

DFU is the abbreviation for Device Firmware Upgrade. It is a USB-based device firmware update protocol..

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

USB DFU communication: a USB DFU driver is required for windows systems, but not for Linux system.

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 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

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 15. 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, AT32F455RxT7 AT32F456ZxT7, AT32F456VxT7,	PC10: USART3_TX PC11: USART3_RX 或

	AT32F456RxT7 AT32F457ZxT7, AT32F457VxT7, AT32F457RxT7	PB10: USART3_TX PB11: USART3_RX
	Others	PB10: USART3_TX PB11: USART3_RX
DFU	All	PA11: OTGFS1_D- PA12: OTGFS1_D+
I ² C1	All	PB6: I2C1_SCL PB7: I2C1_SDA
I ² C2	All	PB10: I2C2_SCL PB3: I2C2_SDA
I ² C3	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 Software operation

4.1 Operating mode

4.1.1 Used in Windows

Mode 1: Input parameters on the command line

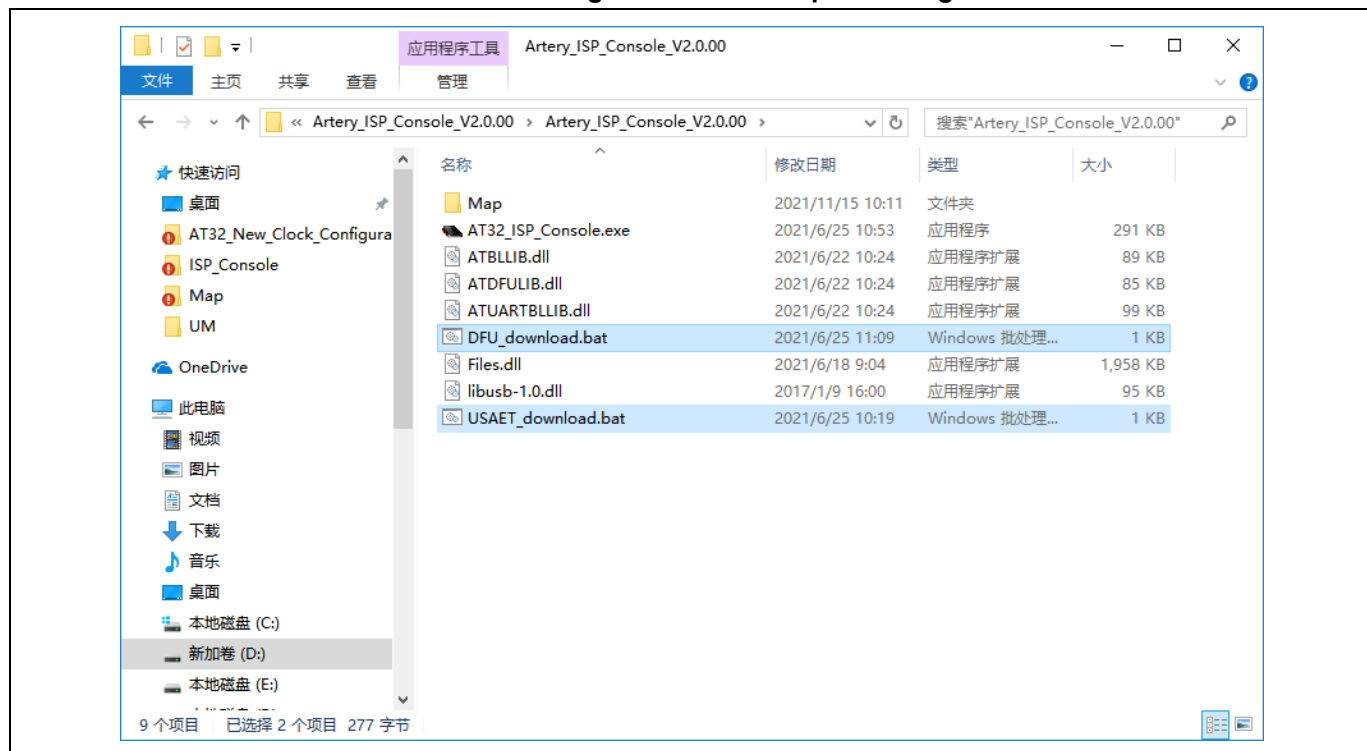
Figure- 1 Command line window

```
管理员: C:\Windows\System32\cmd.exe - AT32_ISP_Console -com --pn 5 --br 115200 --db 8 --pr EVEN --to 5 -p --dfap --depp -usd --ge...
8 --pr EVEN --to 5 -p --dfap --depp -usd --get --fn c:\usd.bin -e --all -d --a 08000000 --v --o --fn c:\test_64k.bin -p
--epp 2-10 --efap -r --a 08000000
-----
Artery ISP Programmer V2.0.00
Support USART and DFU
-----
USART Connect.....
Device: AT32F425R3T7   Flash: 64 KB
PID: 50092100   BID: 4B01   Version: 3.2
Device connected Successfully

Disable access protection
Running...
Disable access protection successfully!
Disable erase and program protection
Running...
Disable erase and program protection successfully!
Get user system data to file
running...
Get user system data to file succeed!
Erase
Running...100%
Erase successfully!
Download
c:\test_64k.bin      65536B
Running...100%
Verify
c:\test_64k.bin      65536B
Running...7%
```

Mode 2: Batch file processing (Refer to DFU_download.bat and USART_download.bat for details on common operations)

Figure- 2 Batch file processing



4.1.2 Used in Linux

1. The script AT32_ISP_Console.sh needs an execution permission. Command: `chmod +x AT32_ISP_Console.sh`;
2. Edit the script USART_download.sh, add operation steps based on command line parameters shown in Section 4.2 (See USART_download.sh in the example) and give an execution permission. Command: `chmod +x USART_download.sh`;
3. To execut the script USART_download.sh in the terminal, a sudo is required, for either a serial interface or USB device needs a root user authority. Command: `sudo ./ USART_download.sh`.

Figure- 3 Linux system

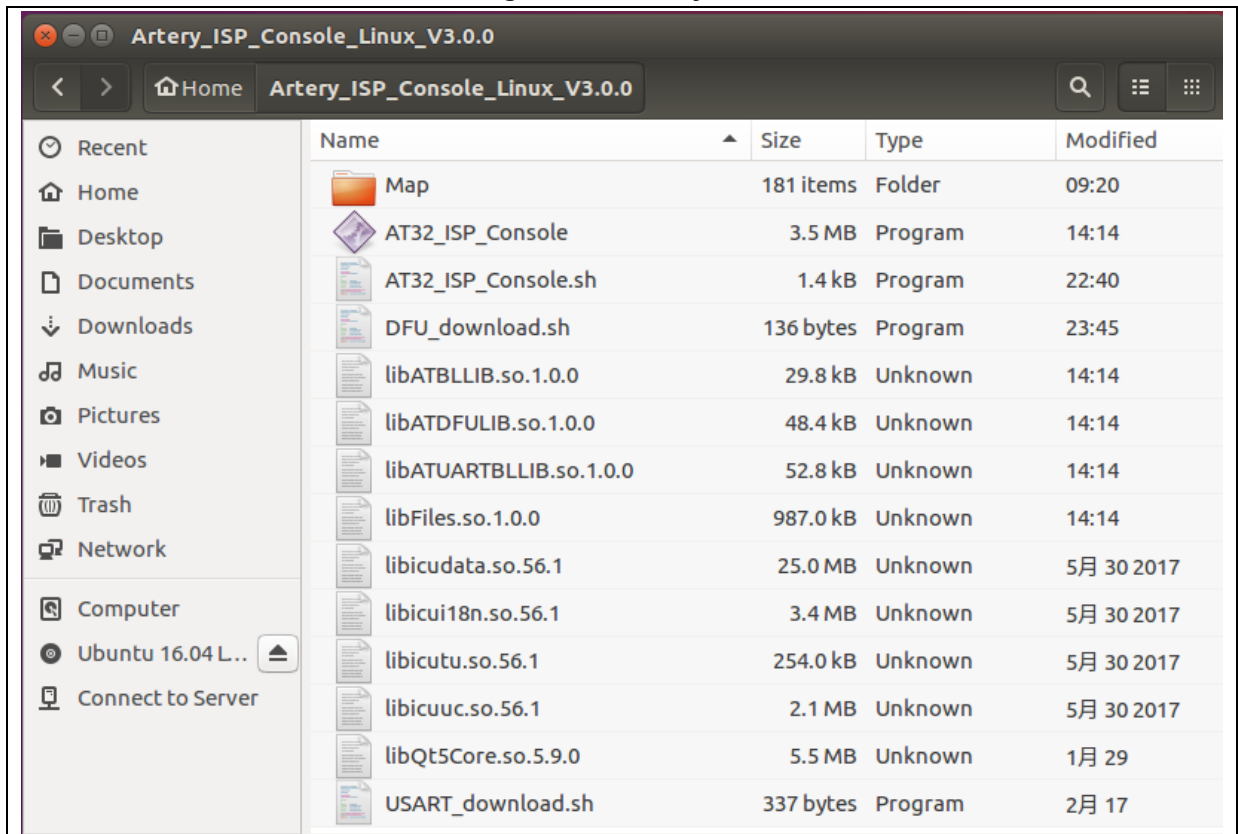
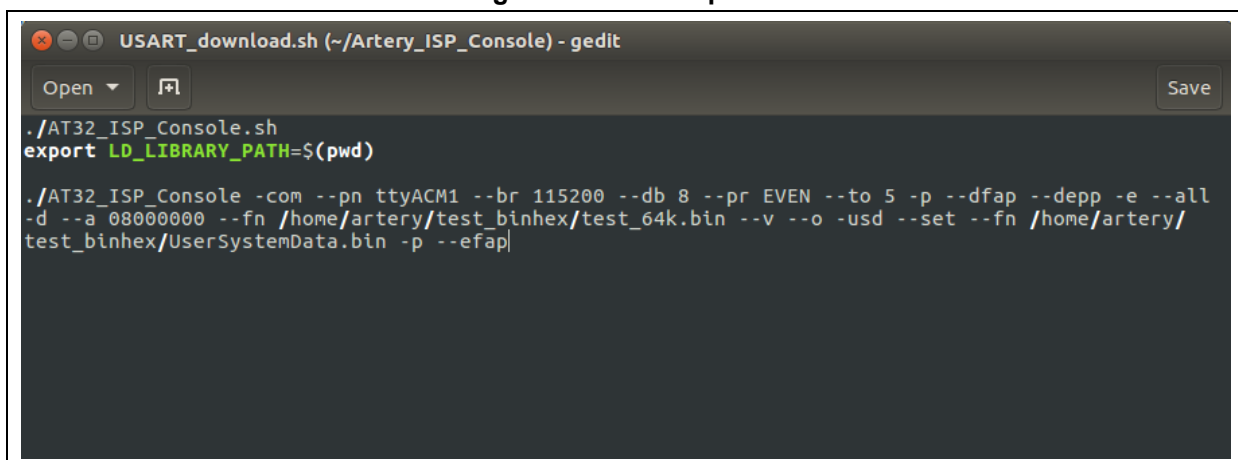


Figure- 4 Shell script file



- Note:
1. To use DFU communication in Windows system, it is required to run Artery_DFU_DriverInstall.exe and install a DFU driver;
 2. When using DFU communication, one PC can only be connected to one device.

4.2 Command line parameters

Table 16 List of command line parameters

Command	Sub Command	Remarks
-?		Show the help.
-dfu		Establish connection to the DFU port.
-com		Establish connection to the COM port.
	--pn port number	If OS is Windows, this is port number, e.g: 1, 2... default 1. Else this is port name, e.g: ttyACM0, ttyS0.
	--br baud rate	E.g: 256000, 128000, 115200, 57600 ..., default 115200.
	--db data bit	Value in {5,6,7,8} ... default 8.
	--pr parity	Value in {ODD,EVEN} default EVEN.
	--to time out	(s) e.g 1, 2, 3 ..., default 5.
	--auto option	Auto connect option, e.g 0, 1, 2, 3 ..., default 0.
		0: Not use RTS and DTR.
		1: DTR low level to reset, not use RTS.
		2: DTR low level to reset, RTS low level to load bootloader.
		3: DTR low level to reset, RTS high level to load bootloader.
		4: DTR high level to reset, not use RTS.
		5: DTR high level to reset, RTS low level to load bootloader.
		6: DTR high level to reset, RTS high level to load bootloader.
		7: RTS low level to reset, not use DTR.
		8: RTS low level to reset, DTR low level to load bootloader.
		9: RTS low level to reset, DTR high level to load bootloader.
		10: RTS high level to reset, not use DTR.
		11: RTS high level to reset, DTR low level to load bootloader.
		12: RTS high level to reset, DTR high level to load bootloader.
-e		Erase flash.
	--all	Erase all sectors of flash,SPIM(SPIM enabled),boot memory(AP mode enabled).
	--sec n-m	Erase selected sectors, begin_sector-end_sector, e.g 0-20.

Command	Sub Command	Remarks
-u		Upload flash contents to the specified file.
	--sec n-m	Upload selected sectors, begin_sector-end_sector, e.g 0-20.
	--fn file_name	Full path name (bin, hex or s19 file; the file type is recognized by its extension).
-d		Download the content of the specified file into flash.
	--a address(hex)	Start address, default 0x08000000; ignored if the target file is not a binary file.
	--fn file_name	Full path name (bin, hex or s19 file; the file type is recognized by its extension).
	--v	verify after download.
	--o	Optimize; removes FFs data..
-p		Enable or disable protection.
	--dfap	Disable flash access protection .
	--depp	Disable erase and program protection .
	--efap	Enable flash access protection, all arguments following this one will fail .
	--efap1	Enable basic access protection, all options following this one will fail.
	--efap2	Enable high level access protection, all options following this one will fail.
	--y	If the device is AT32F425/AT32F423/AT32A423/AT32L021/AT32F402/AT32F405, you must enter "--y" for confirmation. (--efap2 --y).
	--eepp n-m	Enable erase and program protection for sector codes, begin_sector-end_sector, e.g 0-20.
-usd		Set user system data to MCU.
	--get --fn file_name	Get user system data from the device and write it in the specified file , full path name (bin/hex file,the file type is recognized by its extension).
	--set --fn file_name	Load user system data from the specified file and write it to the device , full path name (bin/hex file,the file type is recognized by its extension).
-r		Run the flash code at the specified address.
	--a address(hex)	Address, default 0x08000000.
-enspim		Enable to access SPIM.
	--ft type	SPIM flash type value 1 or 2.default value 1.
	--fs size	SPIM flash size (MB).
	--fda value(hex)	SPIM FLASH_DA, hexadecimal.
	--remap 0/1	Remap IO pin used by SPIM(bank3). 0: remap0(Use PA11/PA12 pins). 1: remap1(Use PB10/PB11 pins).

Note: Depending on the used USB chips, the serial interface hardware flow control DTR and RTS may have opposite electrical levels in Windows and Linux systems. For example, for CH340, its high and low levels are opposite in Windows and Linux systems; while for FT232, its high and low levels are the same in both systems. For others, their levels must be configured according to the actual situations.

4.3 ISP Console return codes

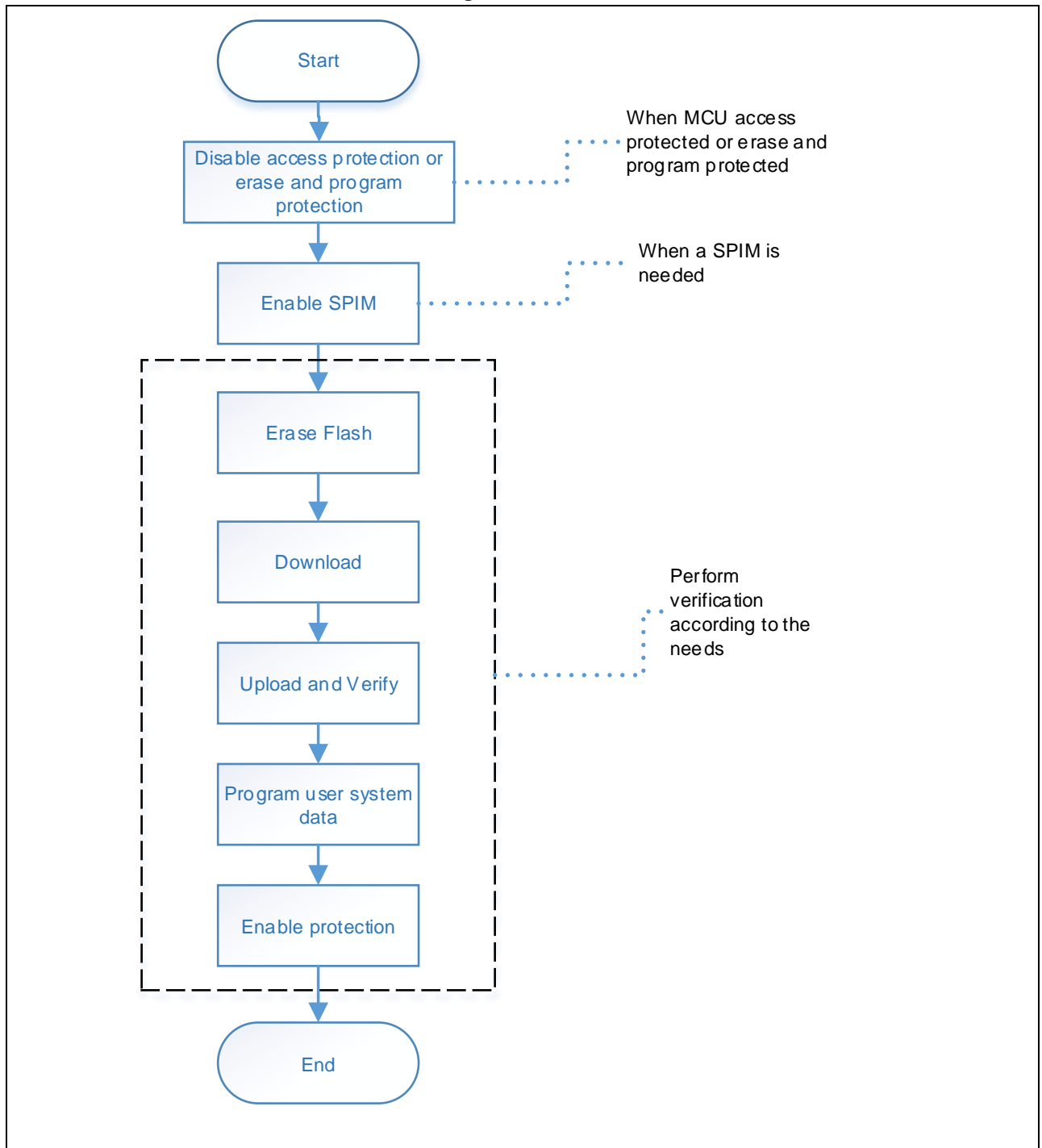
In case of error, while executing AT-Link Console commands, the return code (Errrolevel) is greater than 0.

Table 17 List of return codes

Return code	Command	Error
0x00	All	Finished successfully
0x01	All	Command arguments error.
0x02	All	Connection problem.
0x03	-d	Flash memory programming/verification error.
0x04	-u	Failed to upload Flash memory contents
0x06	-usd --get	Failed to get user system data from the device.
0x07	-usd --set	Failed to write user system data to the device.
0x08	-enspim	Failed to enable SPIM.
0x0C	-r	Failed to run at address.
0x20	-p --efap1	Failed to enable access protection.
0x21	-p --efap2	Failed to enable high level access protection.
0x22	-p --dfap	Failed to disable flash access protection.
0x23	-p --depp	Failed to disable erase and program protection.
0x24	-p --eepp	Failed to enable erase and program protection.
0x30	-e --all	Failed to erase all sectors of Flash.
0x31	-e --all	Failed to erase all sectors of Flash,SPIM(SPIM enabled),boot memory(AP mode enabled).
0x33	-e --sec	Failed to erase selected sectors.
0x40	-otp	Failed to download One-Time Programmable data.

4.4 Flow chart

Figure- 5 Flow chart



5 Revision history

Table 18. Document revision history

Date	Revision	Changes
2025/02/17	V3.11	1. Added AT32F455/F456/F457 series.
2024/10/29	V3.10	1. Added AT32M412/M416 series. 2. Added downloading One-Time Programmable data.
2024/04/26	V3.09	1. Added AT32A423 series.
2023/08/10	V3.08	1. Delete NONE from the parity check option of serial port parameter configuration.
2023/08/10	V3.07	1. Support for AT32F423VCW. 2. Support for AT32F402/F405 series.
2023/07/06	V3.06	1. Support for AT32A403A serial.
2023/03/28	V3.05	1. Support for AT32F435ZDT7、AT32F435VDT7、AT32F435RDT7、AT32F435CDT7、AT32F435CDU7、AT32F437ZDT7、AT32F437VDT7、AT32F437RDT7.
2023/02/17	V3.04	1. Added AT32F423 series.
2022/08/25	V3.03	2. Added AT32F4212C8T7.
2022/08/12	V3.02	1. Added return codes.
2022/07/06	V3.01	1. Added AT32F425 series.
2022/03/16	V3.00	1. Supports multiple platforms,including Windows、Linux(Ubuntu, Fedora) OS.
2022/01/26	V2.03	1. Adjustment description.
2022/01/04	V2.02	1. Added SPIM support. 2. Added access protection and advanced access protection.
2021/11/26	V2.01	1. Added AT32F403/F413/F415/F421/F403A/F407/F435/F437. 2. Added AT32F425 series. 3. Added AT32F403AVGW. 4. Added AT32WB415 series.

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.

© 2025 ARTERY Technology – All Rights Reserved