

Telink

Telink Burning and Debugging Tool (BDT)

User Guide

AN-18101200-E6

Ver1.5.0

2023.08.09

Keyword

BDT

Brief

This document is a user guide for Telink Burning and Debugging Tool (BDT).

Published by
Telink Semiconductor

Bldg 3, 1500 Zuchongzhi Rd,
Zhangjiang Hi-Tech Park, Shanghai, China

© Telink Semiconductor
All Rights Reserved

Legal Disclaimer

This document is provided as-is. Telink Semiconductor reserves the right to make improvements without further notice to this document or any products herein. This document may contain technical inaccuracies or typographical errors. Telink Semiconductor disclaims any and all liability for any errors, inaccuracies or incompleteness contained herein.

Copyright © 2023 Telink Semiconductor (Shanghai) Co., Ltd.

Information

For further information on the technology, product and business term, please contact Telink Semiconductor Company www.telink-semi.com

For sales or technical support, please send email to the address of:

telinksales@telink-semi.com

telinksupport@telink-semi.com

Revision History

Version	Change Description
V1.0.0	Initial release
V1.1.0	Supports multiple devices; added timestamps; path space issues; path buffer overflows, etc.
V1.2.0	Added chapter "Tools Guide"; updated section 2.4 Cache analyze.
V1.3.0	Added section 4.4 RISC-V Tool.
V1.4.0	Added section 4.5 EMI Tool, section 4.6 PA Configure Tool and section 4.7 EVK Offline Tool.
V1.5.0	Added section 4.8 Secure Boot Tool.

Contents

Revision History	3
1 Brief Introduction	12
1.1 Function Overview	12
2 Operation Guide	13
2.1 Download Firmware	13
2.1.1 Connect Hardware	13
2.1.2 Connect Device	16
2.1.3 Download Firmware to "Flash" via "USB" Mode	19
2.1.4 Download firmware into "Flash" via "EVK" mode	28
2.1.5 Download firmware into "SRAM" or "OTP"	36
2.1.6 Multi-address Download	37
2.1.7 OTP Start Information Download	43
2.2 Erase Flash Sector	46
2.3 Flash De-protection	49
2.4 Cache Analyze	50
2.5 Activate MCU	53
2.6 Debug	54
2.6.1 Access Memory	56
2.6.1.1 Read Data From Memory Space	56
2.6.1.2 Write Data into Memory Space	59
2.6.2 Operate Variable List	63
2.6.2.1 Update Variable List	64
2.6.2.2 Sort Variables	65
2.6.2.3 View Aariable Above 4 Bytes	65
2.6.2.4 Modify Variable Value	66
2.6.3 Debug MCU	69
2.6.3.1 Run MCU	69
2.6.3.2 Pause MCU	70
2.6.3.3 Step MCU	71
2.6.3.4 Trace PC	76
2.6.3.5 Stall MCU	80
2.6.3.6 Start MCU	81
2.7 Single-wire synchronization	82
2.8 Command Line	83
2.8.1 Single Command Line	83
2.8.2 Multi-command Line	85
2.9 Log Window	86
2.10 Help	89
2.11 Default Configuration	94
3 Upgrade Firmware	98
4 Tools Guide	102
4.1 BQB Tool	102
4.1.1 Function Description (BQB Tool)	102

4.1.2	Interface Composition and Instructions (BQB Tool)	102
4.1.2.1	Chip Selection	103
4.1.2.2	Firmware Saving and Downloading	103
4.1.2.3	Sending and Receiving Packet Test	103
4.2	Jtag Tool	104
4.2.1	Function Description (Jtag Tool)	104
4.2.2	Interface Composition and Instructions (Jtag Tool)	104
4.2.2.1	Jtag Device Driver Installation	105
4.2.2.2	Connecting Debug Equipment	106
4.2.2.3	Download Function	107
4.2.2.4	Erase Function	107
4.2.2.5	Read/Write Functions	107
4.2.2.6	Debug Function	108
4.3	TWS Tool	108
4.3.1	Download Tool	108
4.3.1.1	Function Description (Download Tool)	108
4.3.1.2	Interface Introduction	108
4.3.1.3	Instructions for Use	109
4.3.2	EQ Tool	109
4.3.2.1	Function Description (EQ Tool)	109
4.3.2.2	Interface Composition	109
4.3.2.3	Instructions for Use	110
4.4	RISC-V Tool	111
4.4.1	Read Memory	111
4.4.2	Write Memory	112
4.4.3	Download Firmware	113
4.4.4	VCD Function	116
4.4.5	Command Line Description	119
4.4.5.1	Read Command	119
4.4.5.2	Write Command	120
4.4.5.3	Flash Erase Command	121
4.4.5.4	Save the Read Data	122
4.4.6	Clear and Save Logs	123
4.5	EMI Tool	125
4.5.1	Functional description (EMI Tool)	125
4.5.2	Interface Composition and Instructions (BQB Tool)	125
4.5.2.1	Chip Selection	129
4.5.2.2	Save and Download Firmware	129
4.5.2.3	EMI Test	130
4.5.2.4	Non-signaling Test	130
4.5.2.5	RF Current Test	130
4.6	PA Configure Tool	130
4.6.1	Function Description (PA Configure Tool)	130
4.6.2	Interface Composition and Instructions (PA Configure Tool)	131
4.6.2.1	Configure the File that Generates the PA	131
4.7	EVK Offline Tool	131
4.7.1	Function Description (EVK Offline Tool)	131

4.7.2	Interface Composition and Instructions (EVK Offline Tool)	132
4.7.2.1	Update Tool Firmware	133
4.7.2.2	Download User Firmware	133
4.8	Secure Boot Tool	133
4.8.1	Interface and Instructions	134
4.8.1.1	Mode Selection	134
4.8.1.2	Firmware Encryption	134
4.8.1.3	Secure Debug	134
4.8.1.4	Run Description Address	134
4.8.1.5	Pubkey Hash	135
4.8.1.6	Root Key	135
4.8.1.7	Debug Plaintext	137
4.8.1.8	Run Code	139
4.8.1.9	Run Code description	140
4.8.1.10	Download-All	140
4.8.1.11	Download-Flash	140
4.8.1.12	Re-enable Debug-interface	140
4.8.1.13	Read-Info	141
4.8.1.14	Save	142
4.8.2	Commonly Used Secure Mechanism Options and Usage Instructions of the Tool	142
4.8.2.1	Secure Boot + Secure Debug	143
4.8.2.2	Firmware Encryption + Secure Debug	143
4.8.3	Secure Boot + Firmware Encryption + Secure Debug	144
5	Function Support List	146

List of Figures

Figure 1.1	Main interface	12
Figure 2.1	Connect target board with PC directly	13
Figure 2.2	Connect "Burning EVK" with PC	14
Figure 2.3	Connect target board with "Burning EVK" via USB	15
Figure 2.4	Connect target board with "Burning EVK" via Swire	16
Figure 2.5	Burning and Debugging Tool	17
Figure 2.6	Check available devices	18
Figure 2.7	Select the specified device	19
Figure 2.8	Select chip type	20
Figure 2.9	Select download mode as "USB"	21
Figure 2.10	Select starting offset address	22
Figure 2.11	Open file select window	23
Figure 2.12	Select target file and open new file path	24
Figure 2.13	Directly select target file path in the submenu of "Reopen"	25
Figure 2.14	Connection status indication	26
Figure 2.15	Download firmware into Flash via USB mode	27
Figure 2.16	Manual reset MCU	27
Figure 2.17	Auto reset MCU	28
Figure 2.18	Select chip type	29
Figure 2.19	Select download mode as "EVK"	30
Figure 2.20	Open the "Setting" Dialog	31
Figure 2.21	Select the target file	31
Figure 2.22	Select bin file	32
Figure 2.23	Check connection status	33
Figure 2.24	Download firmware into Flash via "Burning EVK" mode	34
Figure 2.25	Manual reset MCU	35
Figure 2.26	Auto reset MCU	36
Figure 2.27	Download firmware to "SRAM" via "auto mode"	37
Figure 2.28	Select target space	38
Figure 2.29	Open "Multi-address Download" interface	39
Figure 2.30	Add firmware file and set target address	39
Figure 2.31	Add firmware	40
Figure 2.32	Show bin file path	40
Figure 2.33	Set offset address	41
Figure 2.34	Show calculation results	41
Figure 2.35	Add the path of firmware 2	42
Figure 2.36	"Download" button	42
Figure 2.37	Start downloading firmware 1	42
Figure 2.38	Start downloading firmware 2	43
Figure 2.39	Reset	43
Figure 2.40	1st file	44
Figure 2.41	Select chip	45
Figure 2.42	OTP start information download	46
Figure 2.43	Select starting address	47

Figure 2.44	Set erase size	48
Figure 2.45	Erase 512kB flash space starting from 0x000000	49
Figure 2.46	Flash remove protection	50
Figure 2.47	Bin file selection	51
Figure 2.48	Cache data is correct	52
Figure 2.49	Cache data has error	53
Figure 2.50	Activate MCU	54
Figure 2.51	"Memory Access" interface	55
Figure 2.52	Select MCU type and communication mode	56
Figure 2.53	Set operation target and size	57
Figure 2.54	Set starting address and initiate reading	58
Figure 2.55	Save read data	59
Figure 2.56	Memory access window	60
Figure 2.57	Set operation size	61
Figure 2.58	Offset of the starting address	62
Figure 2.59	Set data and initiate writing	63
Figure 2.60	Case 1	63
Figure 2.61	Case 2	64
Figure 2.62	Update variable list	64
Figure 2.63	Sort variable by name	65
Figure 2.64	Read variable value more than 4 bytes	66
Figure 2.65	Double click specified "Value" box	67
Figure 2.66	Modify variable value	68
Figure 2.67	Write data to specified address	69
Figure 2.68	Run MCU	70
Figure 2.69	Pause MCU	71
Figure 2.70	Step MCU in "Single step" mode	72
Figure 2.71	Step MCU in "continuous" mode	73
Figure 2.72	"xxx.lst" file	73
Figure 2.73	Select "interp.lst"	74
Figure 2.74	Step MCU with "interp.lst" selected in "Single step" mode	75
Figure 2.75	Step MCU with "interp.lst" selected in "continuous" mode	76
Figure 2.76	Trace PC in "Single step" mode	77
Figure 2.77	Trace PC with "interp.lst" selected in "Single step" mode	78
Figure 2.78	Switch "Single step" mode to "continuous" mode	79
Figure 2.79	Trace PC with "interp.lst" selected in "continuous" mode	80
Figure 2.80	Stall MCU	81
Figure 2.81	Start MCU	82
Figure 2.82	Set Swire synchronization speed	83
Figure 2.83	Open cmdline input window	84
Figure 2.84	Enter single command line	85
Figure 2.85	Enter multiple commands	86
Figure 2.86	Open usb log window	87
Figure 2.87	Display output characters	88
Figure 2.88	Exit "usb log" mode	89
Figure 2.89	Command line	90
Figure 2.90	Show all commands	91

Figure 2.91	User guide	92
Figure 2.92	Open Help document	93
Figure 2.93	About Telink	94
Figure 2.94	Get information about Telink	94
Figure 2.95	Folder "config"	95
Figure 2.96	MCU	95
Figure 2.97	Mode of Communication	95
Figure 2.98	Objection	96
Figure 2.99	Size	96
Figure 2.100	Download address	96
Figure 2.101	Erase address	96
Figure 2.102	Page display	96
Figure 2.103	Button	97
Figure 2.104	Path	97
Figure 3.1	"Upgrade EVK" window	98
Figure 3.2	Check current firmware version	99
Figure 3.3	Load new firmware	100
Figure 3.4	Upgrade firmware into EVK	101
Figure 4.1	BQB tool firmware download interface	102
Figure 4.2	BQB tool test interface	103
Figure 4.3	Jtag tool interface	104
Figure 4.4	Jtag device	105
Figure 4.5	Jtag device driver installed	105
Figure 4.6	Jtag device driver is not installed	106
Figure 4.7	TWS_Tool/Download_Tool interface	109
Figure 4.8	TWS_Tool/EQ_Tool interface	110
Figure 4.9	Read memory	112
Figure 4.10	Write memory	113
Figure 4.11	Select firmware	114
Figure 4.12	Erase flash	115
Figure 4.13	Download firmware	116
Figure 4.14	Select the header file	117
Figure 4.15	Start the VCD function	118
Figure 4.16	View status	119
Figure 4.17	Read command	120
Figure 4.18	Write command	121
Figure 4.19	Flash erase command	122
Figure 4.20	Save the read data	123
Figure 4.21	Clear log	124
Figure 4.22	Save log	125
Figure 4.23	Firmware configuration download interface	126
Figure 4.24	EMI test interface	127
Figure 4.25	Non-signaling test interface	128
Figure 4.26	RF current test interface	129
Figure 4.27	"PA Configure Tool" interface	131
Figure 4.28	"EVK Offline Tool" tool firmware interface	132
Figure 4.29	"EVK Offline Tool" user firmware interface	133

Figure 4.30	Secure Boot tool interface	134
Figure 4.31	Public key hash file error	135
Figure 4.32	Import RootKey.txt file	136
Figure 4.33	Root Key format error 1	137
Figure 4.34	Root Key format error 2	137
Figure 4.35	Import DebugPlaintext.txt file	138
Figure 4.36	Debug Plaintext format error 1	139
Figure 4.37	Debug Plaintext format error 2	139
Figure 4.38	Re-enable Debug interface	141
Figure 4.39	Read the information	142
Figure 4.40	The path to the key folder	142
Figure 4.41	“Secure Boot + Secure Debug” burning steps	143
Figure 4.42	“Firmware Encryption + Secure Debug” burning steps	144
Figure 4.43	“Secure Boot + Firmware Encryption + Secure Debug” burning steps	145
Figure 5.1	Function Support List	146

Telink Semiconductor

List of Tables

Table 2.1	Introduction of Memory Access	55
-----------	---	----

Telink Semiconductor

1 Brief Introduction

"Telink Burning and Debugging Tool (BDT)" applies to all engineers who want to develop applications based on Telink SoCs including 8267, 8269, 8266, 8232, 8233, 8366, 836, B85M series and B91M series.

This document serves as a guide on how to use the "Telink Burning and Debugging Tool (BDT)".

1.1 Function Overview

During SDK development, by using "Telink Burning and Debugging Tool (BDT)", firmware can be directly downloaded into the target board (e.g. development board) via "USB" mode or "Burning EVK" (abbreviated as EVK) mode.

Its main functions include "Erase flash sector", "Download firmware", "Activate MCU when communication failed", "Access memory space including Flash/ Core/ Analog/ OTP", "Read/Write global variable" and "View USB log".

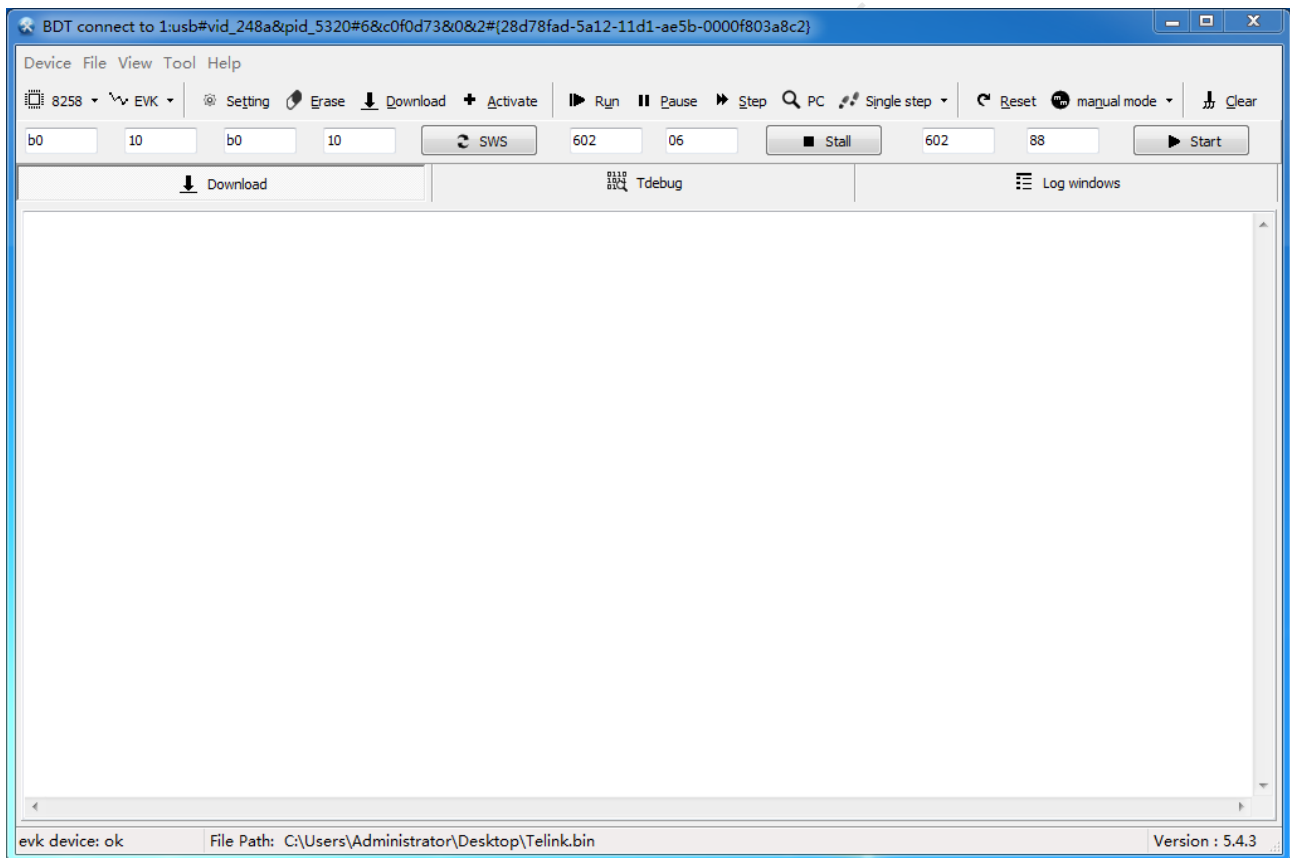


Figure 1.1: Main interface

2 Operation Guide

2.1 Download Firmware

2.1.1 Connect Hardware

Before using "Telink Burning and Debugging Tool (BDT)", it's needed to connect the target board with PC.

There are two methods to connect the target board with PC, as shown below.

Method 1: Directly connect the target board with PC via USB.

This method only applies to target board with USB interface and MCU supporting USB function, e.g. a dongle board.

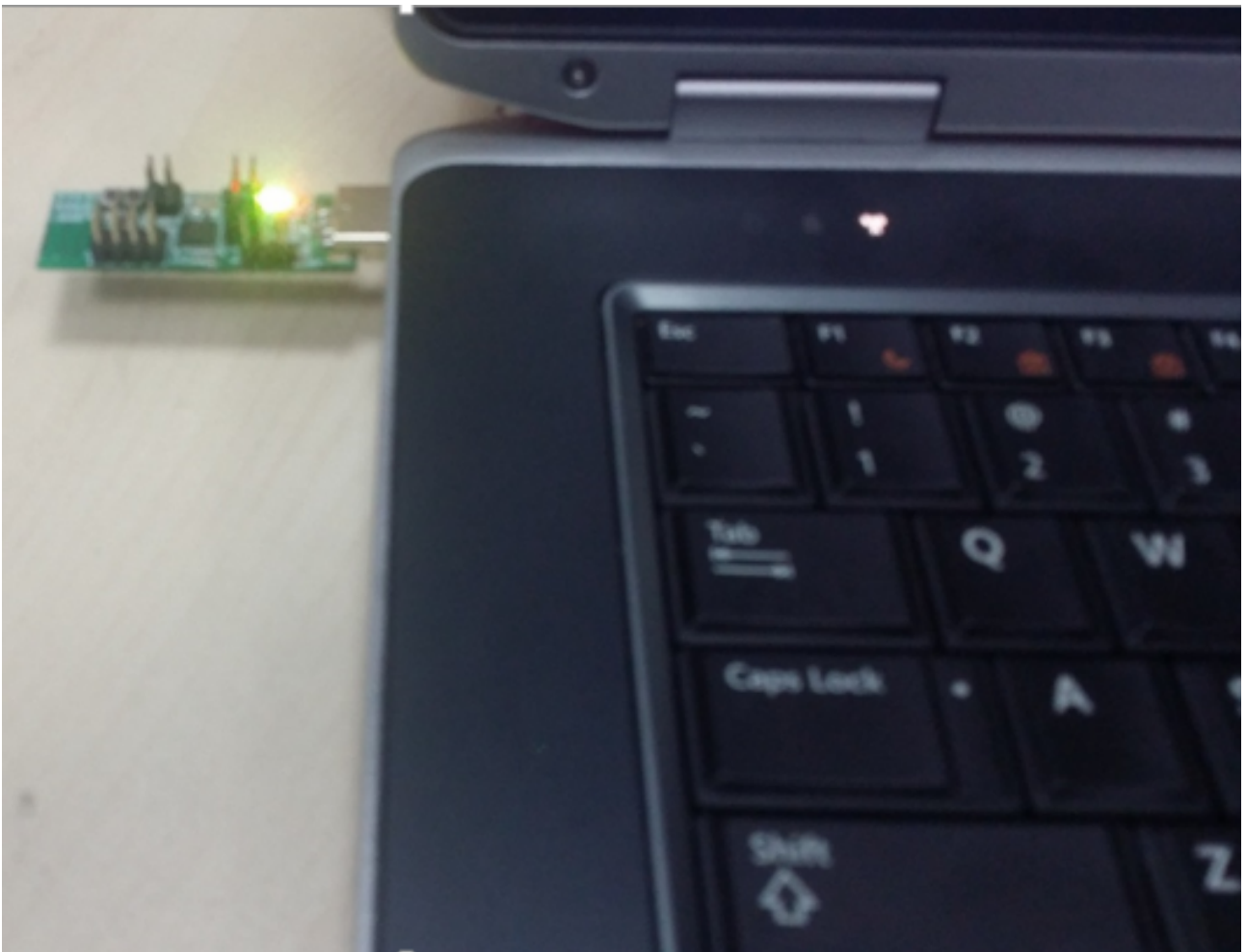


Figure 2.1: Connect target board with PC directly

Method 2: Connect the target board with PC via Telink "Burning EVK" TLSR8266BR56.

- (1) Connect "Burning EVK" with PC via USB. Observe the indicator light of "Burning EVK", the indicator light flashes once, indicating that "Burning EVK" is successfully connected to the PC.



Figure 2.2: Connect "Burning EVK" with PC

(2) Connect target board with "Burning EVK".

There are two methods to connect the target board with the "Burning EVK".

- a. Connect the target board with the "Burning EVK" via USB interface directly, as shown below. This connection method only applies to target board with USB interface and MCU supporting USB function, e.g. a dongle board.



Figure 2.3: Connect target board with "Burning EVK" via USB

- b. Connect the target board with the "Burning EVK" via Swire (single wire) interface, as shown below.



Figure 2.4: Connect target board with "Burning EVK" via Swire

Please refer to Telink document "AN_18010500_User Guide for Telink Burning EVK TLSR8266BR56" for the detailed guide of TLSR8266BR56.

After connecting the target board to PC via USB method or "Burning EVK" method, there are two methods to download firmware into the target board, which respectively correspond to the two hardware connection methods above.

2.1.2 Connect Device

Before downloading firmware into the target board, please ensure that the device have been found by BDT as the following figure.

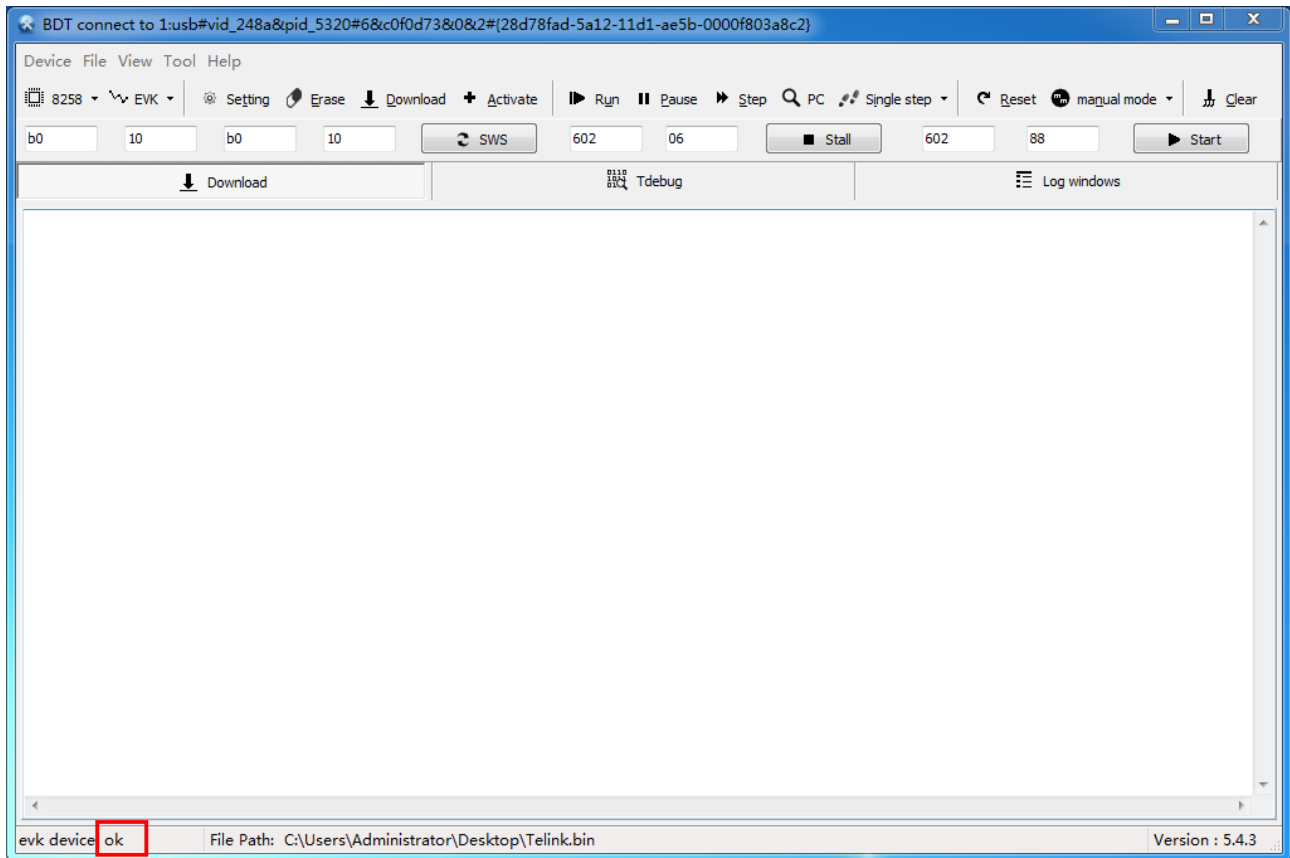


Figure 2.5: Burning and Debugging Tool

If BDT does not find the device, as shown below, you can click “Refresh” to check the available devices.

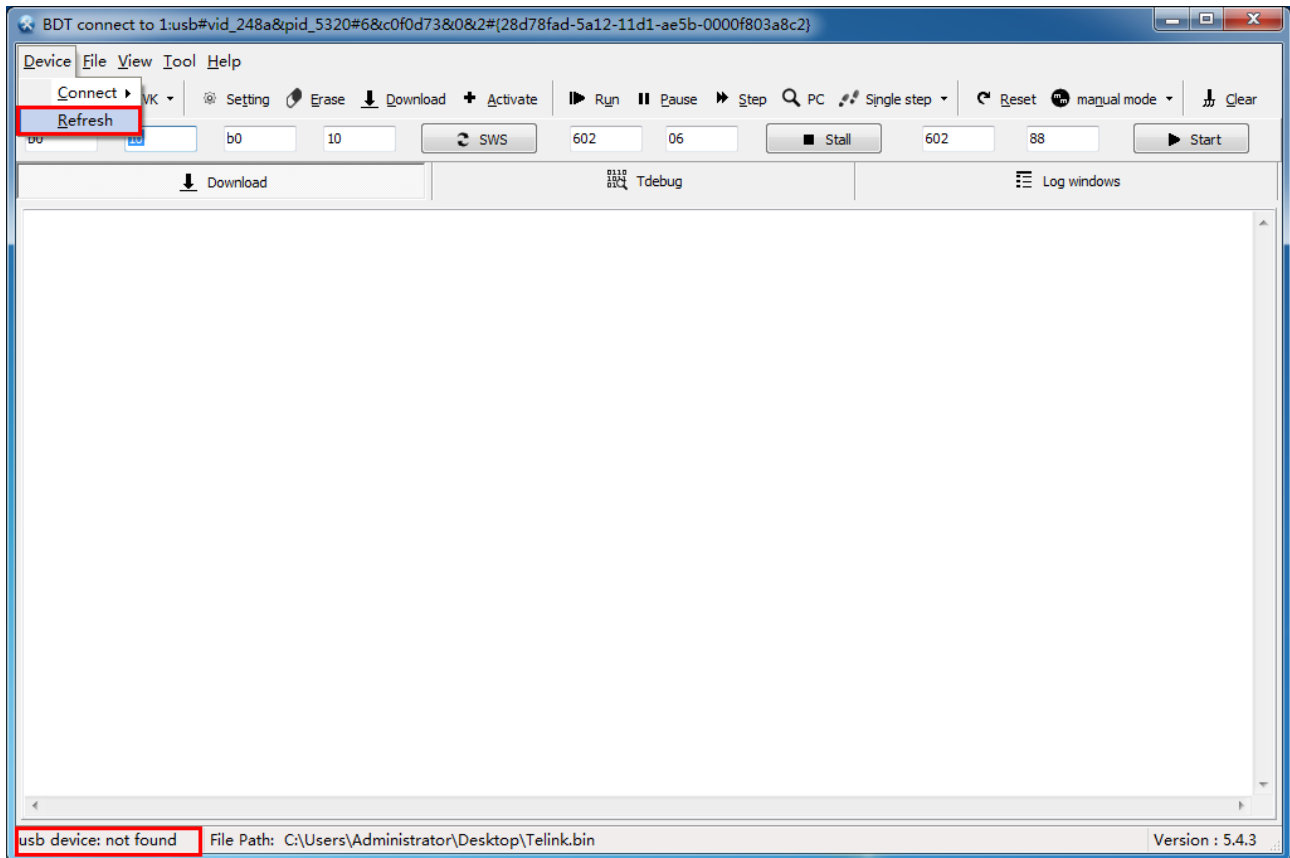


Figure 2.6: Check available devices

If BDT finds more than one device, all devices will be listed as shown in the figure below. After clicking the button “Refresh”, the first device will be connected to BDT by default. If you want to connect other devices, you can select the specified device as shown below.

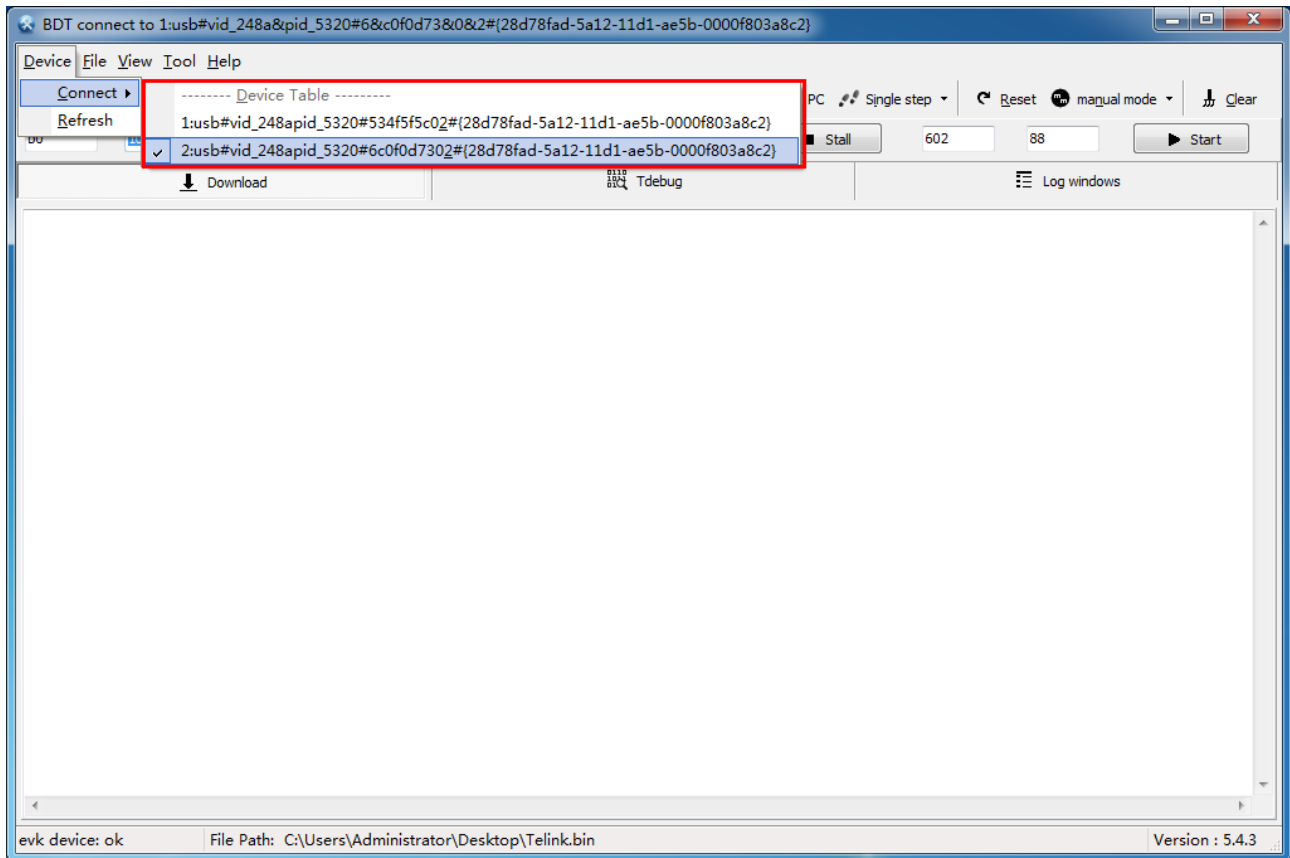


Figure 2.7: Select the specified device

2.1.3 Download Firmware to “Flash” via “USB” Mode

Before using “USB” mode to download or debug MCU, please ensure that the specified MCU supports USB function and its USB function is available. Refer to [section 5 Function Support List](#) for details.

Then user can follow the guide in this section to download firmware into specific flash space of the target board via “USB” mode.

Step 1: Select chip type of the target board, e.g. 8258 (default option). The following two methods are supported.

- Method 1: Use the drop-down menu to select chip type.
- Method 2: Press the shortcut key combination “Ctrl + Q” to switch chip type.

Note:

All chip type options and default options can be configured in the “config.ini” file.

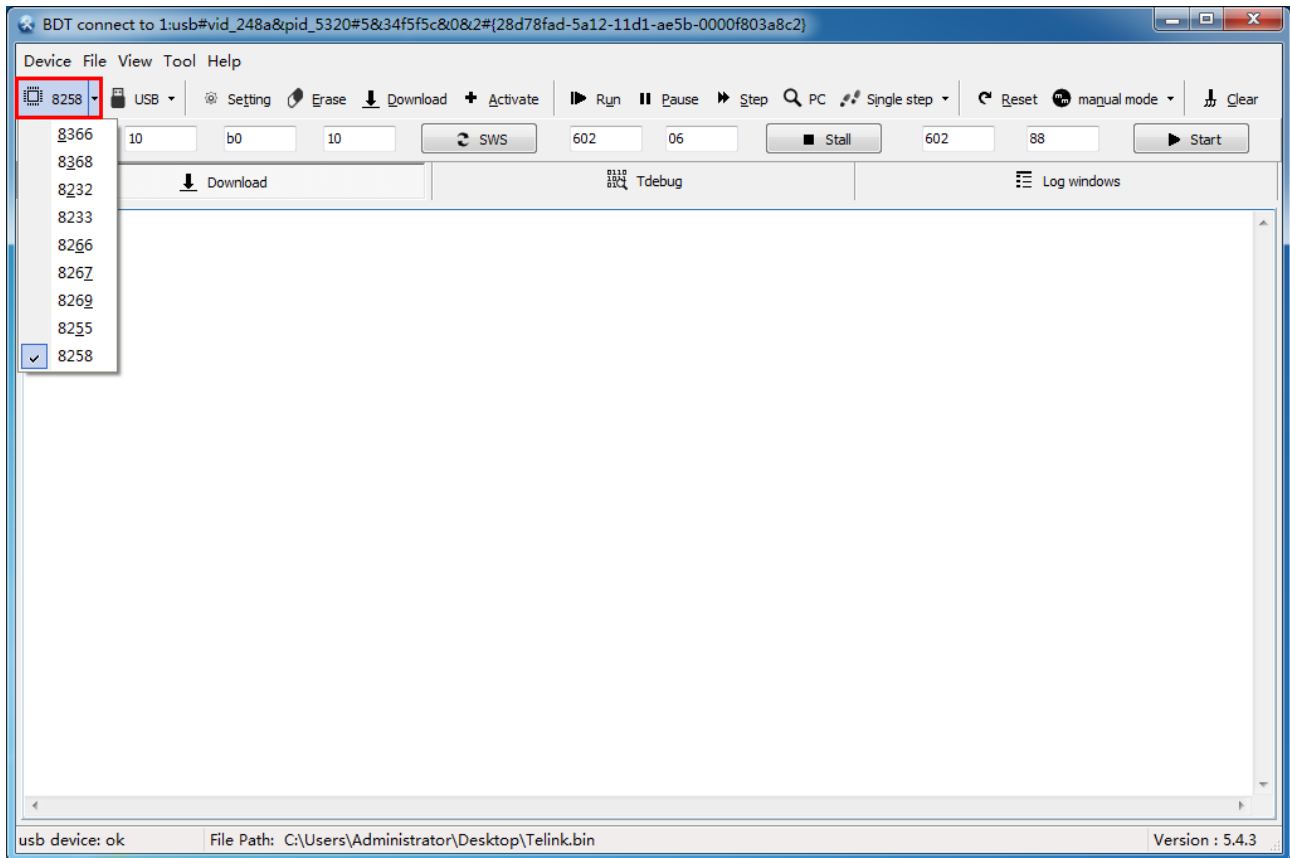


Figure 2.8: Select chip type

Step 2: Select the download mode as "USB" and the following two methods are supported.

- Method 1: Use the drop-down menu to select USB download mode.
- Method 2: Press the shortcut key combination "Ctrl + W" to switch to USB download mode. The drop-down menu will automatically show the current download mode after each switch.

Note:

All download modes and default options can be configured in the "config.ini" file.

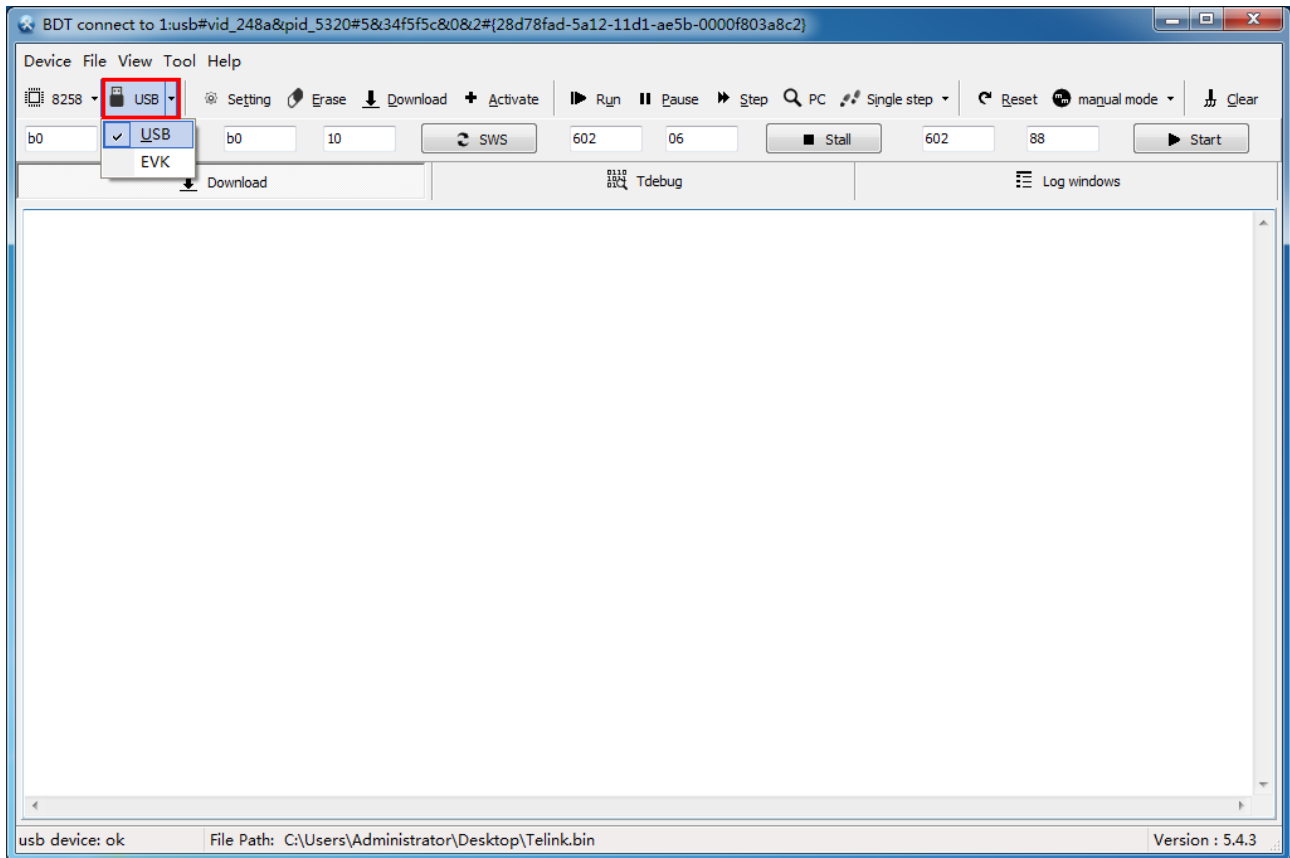


Figure 2.9: Select download mode as “USB”

Step 3: Click the “Setting” button to open the setting dialog window, and select the “Flash” option. The user can use any of the following methods to set the flash start address offset of the target firmware storage, e.g. 0x000000 (default option).

- Method 1: Use the “Download Addr” drop-down menu to select the available offset options.
- Method 2: Enter the offset address in the edit box for “Download Addr”.

Note:

- All settings and default options can be configured in the “config.ini” file.
- The user can also configure the “SRAM”/“OTP” setting option to download the target firmware to the target area in the SRAM or OTP.

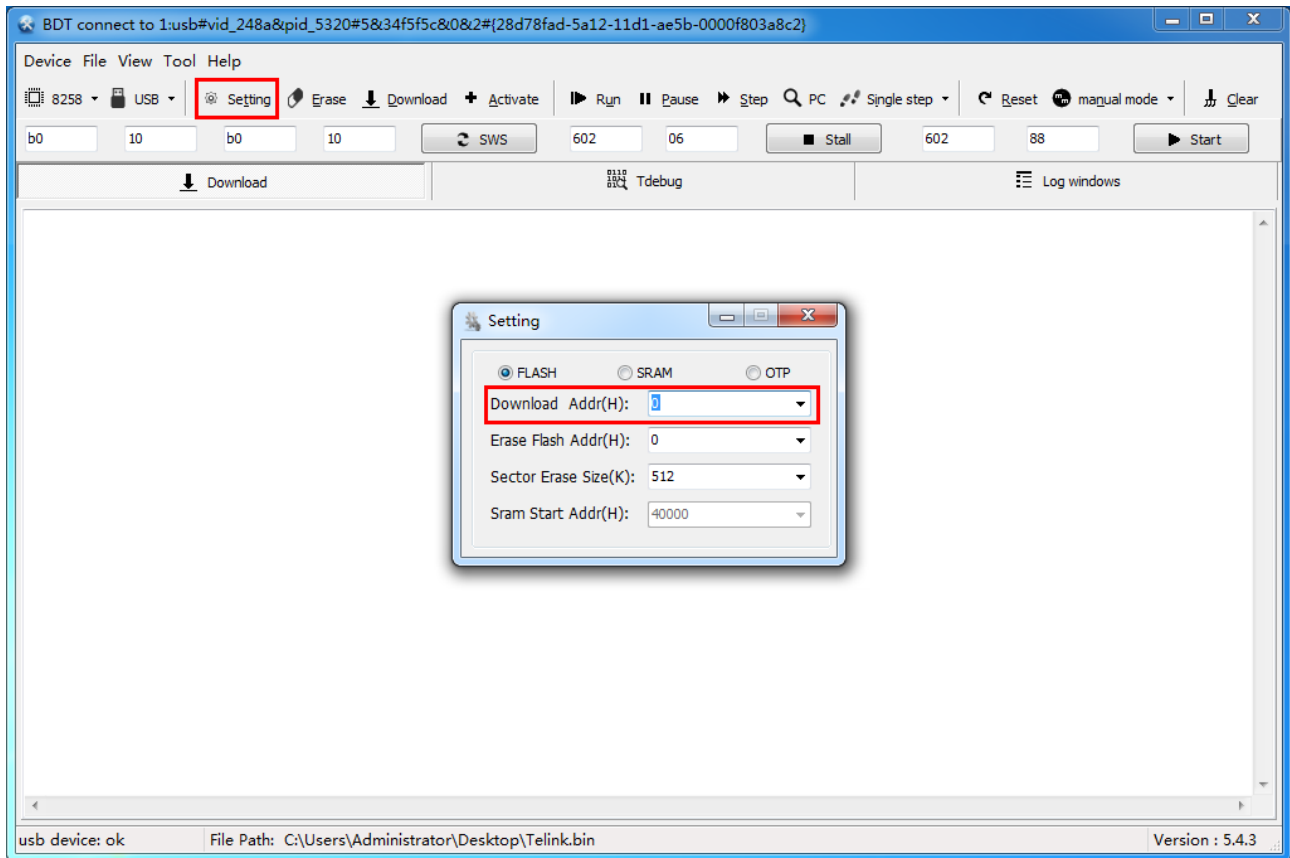


Figure 2.10: Select starting offset address

Step 4: Select the target firmware file to be downloaded into the target board.

- (1) Click File -> Open to open the file select window.

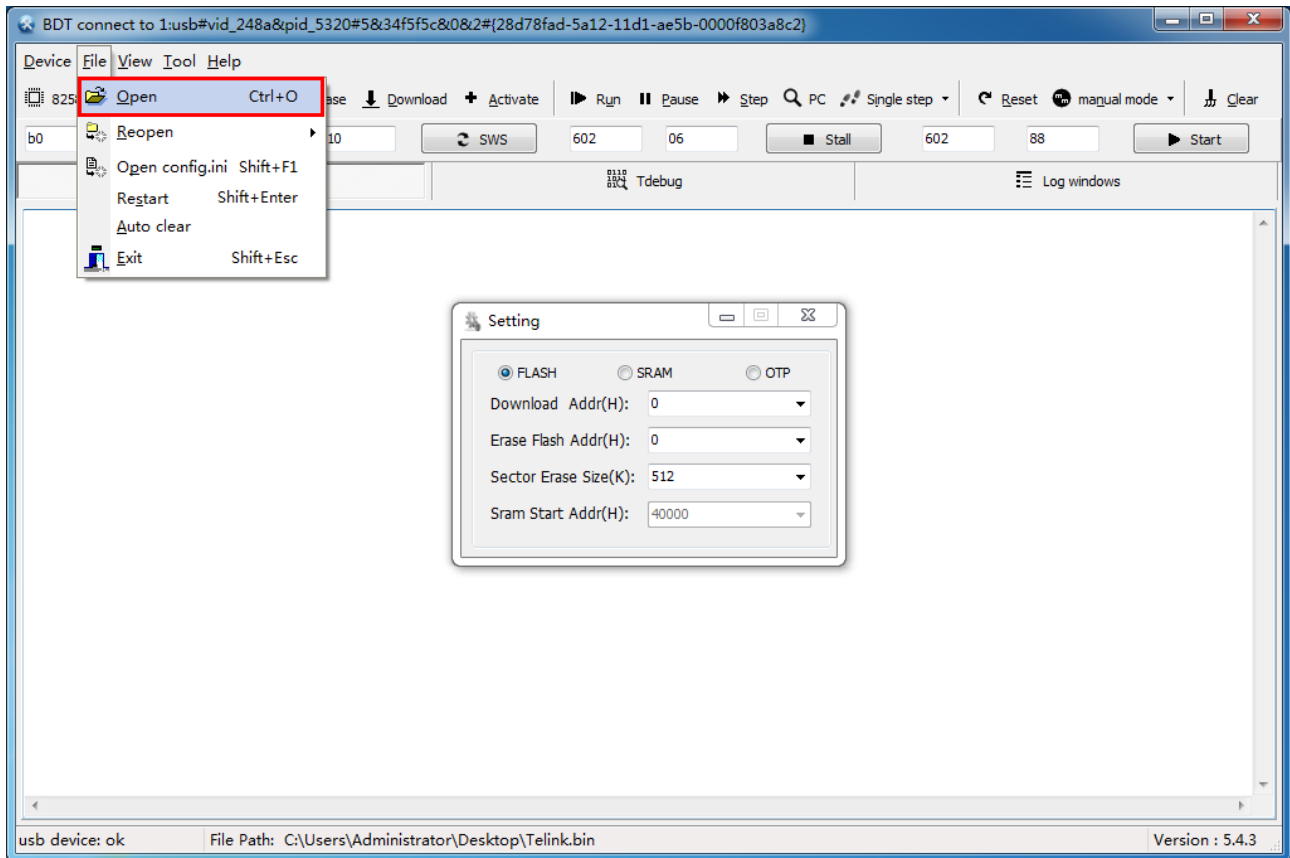


Figure 2.11: Open file select window

- (2) In the file select window, select the target bin file (e.g. Telink.bin) and then click the "Open" button.

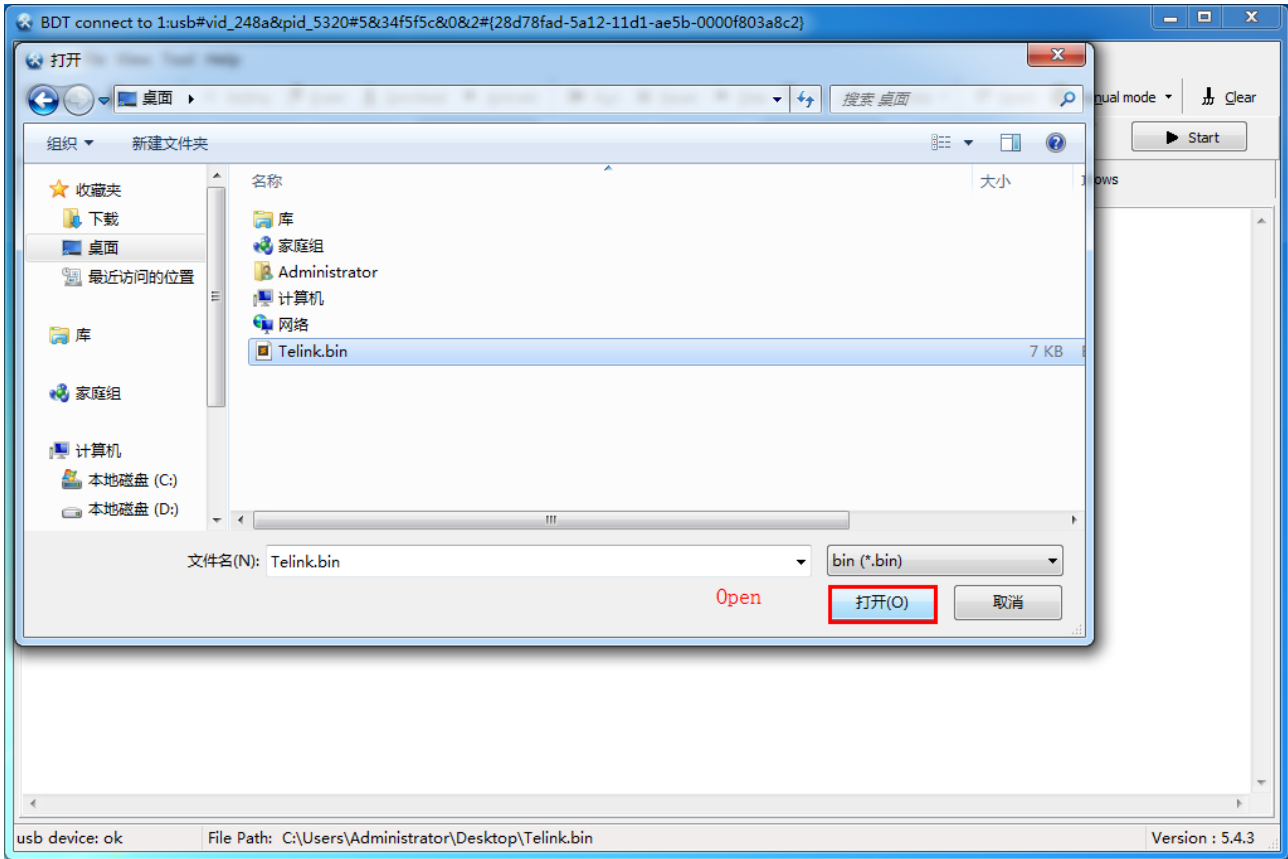


Figure 2.12: Select target file and open new file path

The file path will be displayed at the bottom of the main interface. Every time a new file path is opened via the file selection window, it will be added to the submenu of "File->Reopen". This simplifies firmware selection in subsequent operations by allowing the user to directly select the target file path in the submenu of "Reopen".

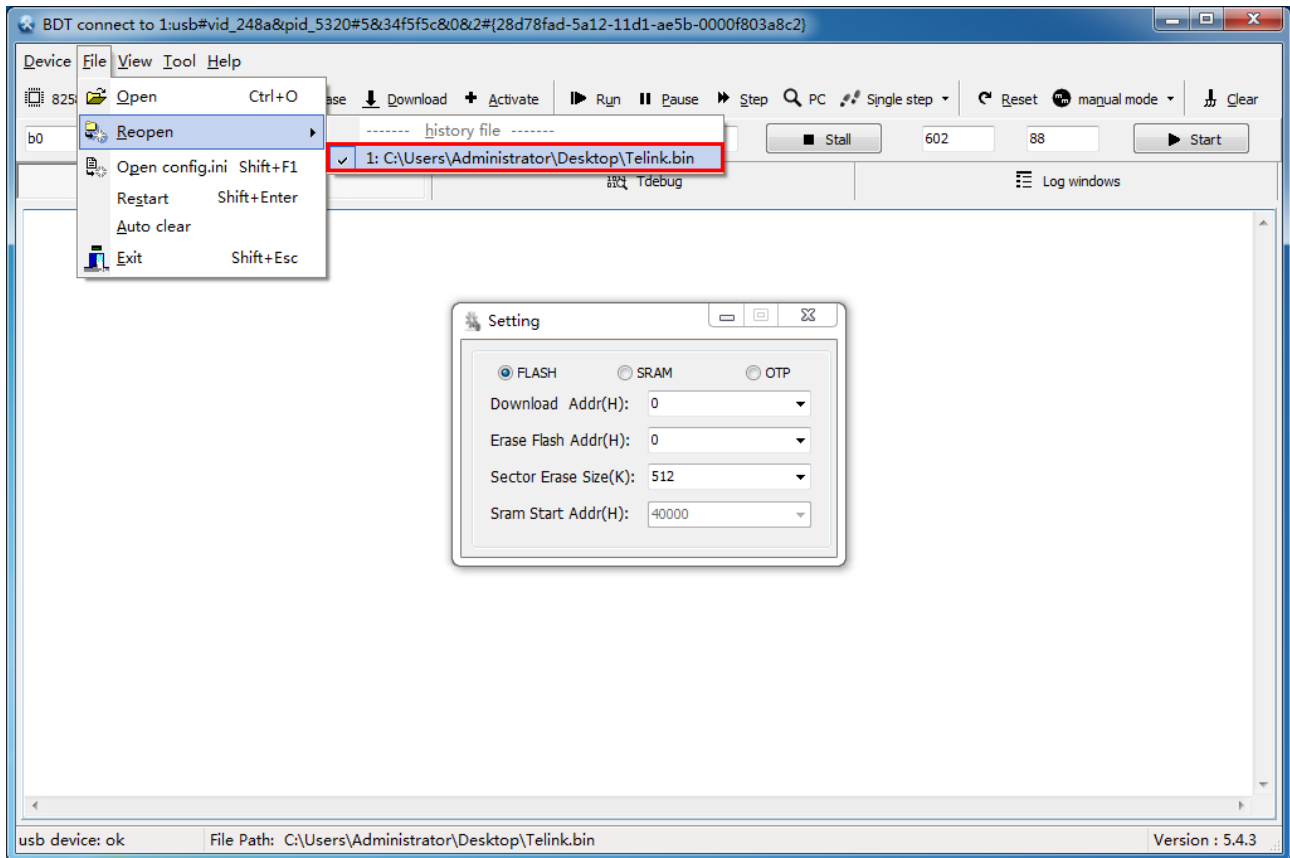


Figure 2.13: Directly select target file path in the submenu of “Reopen”

Step 5: Check the connection status between the target board and the PC to make sure it is working properly.

- If the connection status is normal, the left bottom of the main interface will display “usb device: ok” or “evk device: ok”.
- If the left bottom of the main interface displays “usb device: not found”, it indicates the connection between the target board and the PC is not available.

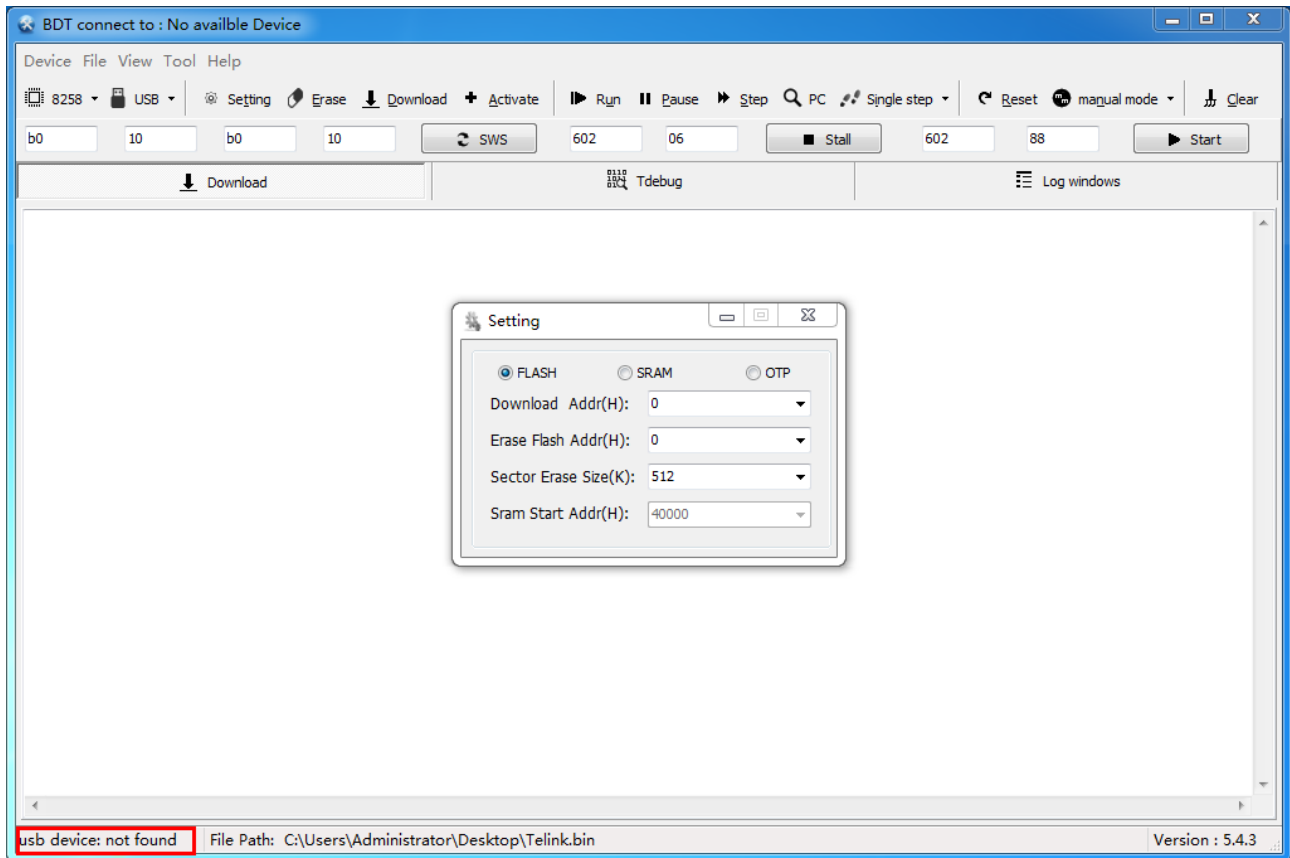


Figure 2.14: Connection status indication

Step 6: Download the selected file into the target board.

By clicking the “Download” button, the selected firmware file will be downloaded into the specified flash space of the target board via USB mode. The Log window will display the corresponding log information, as shown below.

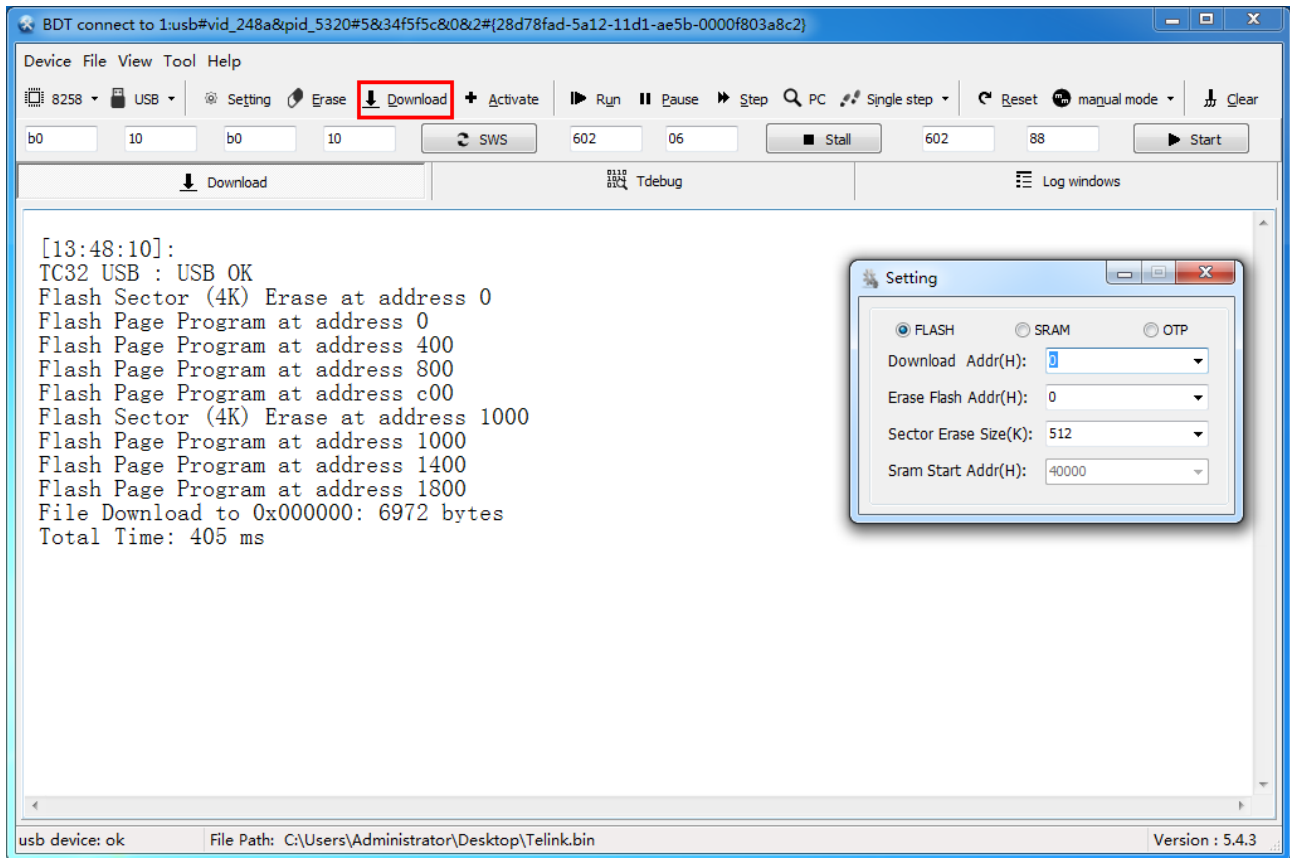


Figure 2.15: Download firmware into Flash via USB mode

Step 7: Reset MCU to make the newly downloaded program run without the need to power cycle MCU (unplug/plug from Burning EVK). User can follow either of the following two methods to make MCU boot and run.

- If “manual mode” is selected via the drop-down menu, after the firmware burning, the user can click the “Reset” button in the top right corner of the tool to manually reset the MCU.

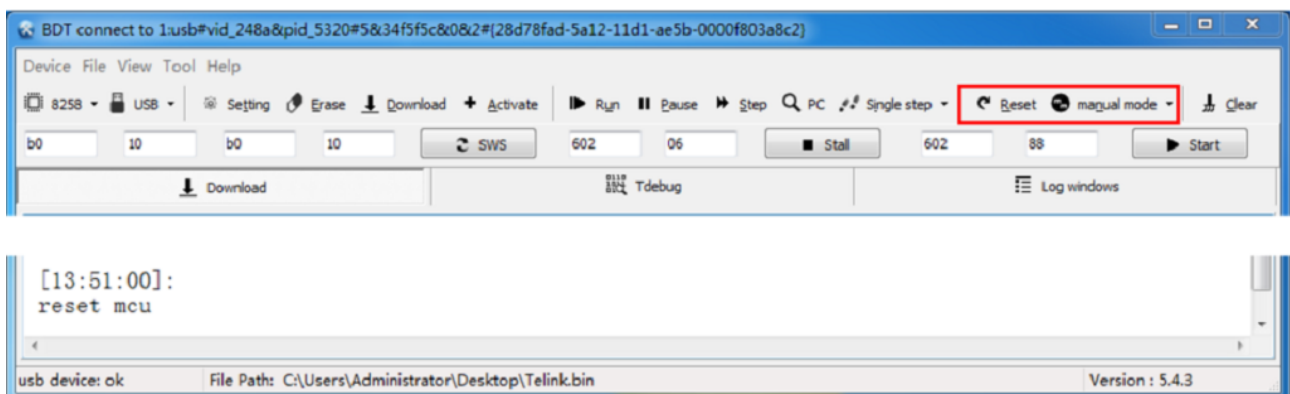


Figure 2.16: Manual reset MCU

- Users can also select “auto mode” via the drop-down menu before firmware burning, so as to enable auto reset after firmware burning.

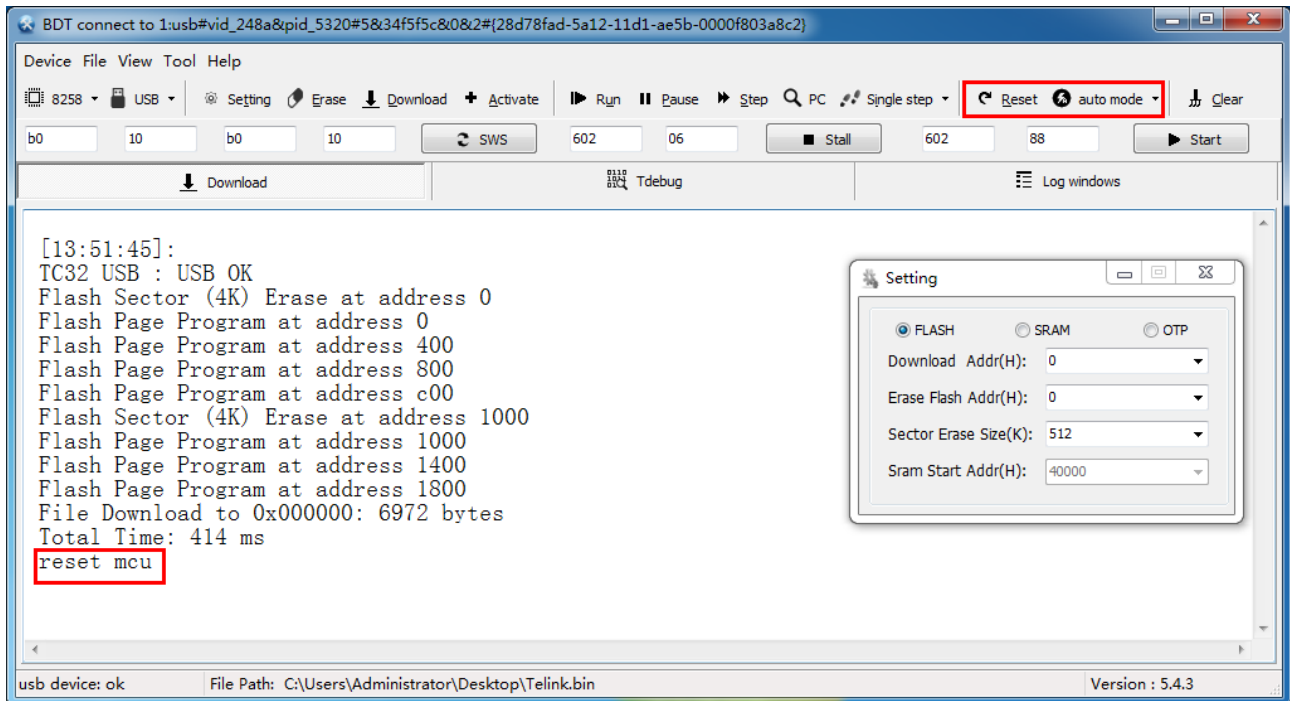


Figure 2.17: Auto reset MCU

2.1.4 Download firmware into “Flash” via “EVK” mode

Before using “EVK” mode to download firmware, please ensure that target board is connected to PC via “Burning EVK” method and the single wire communication of the target board is established.

User can follow the guide in this section to download firmware into specific flash space of the target board via “EVK” mode.

Step 1: Select chip type of the target board, e.g. 8258 (default option). The following two methods are supported.

- Method 1: Use the drop-down menu to select chip type.
- Method 2: Press the shortcut key combination “Ctrl + Q” to switch chip type, the drop-down menu will automatically display the current type after each switch.

Note:

All chip types and default options can be configured in the “config.ini” file.

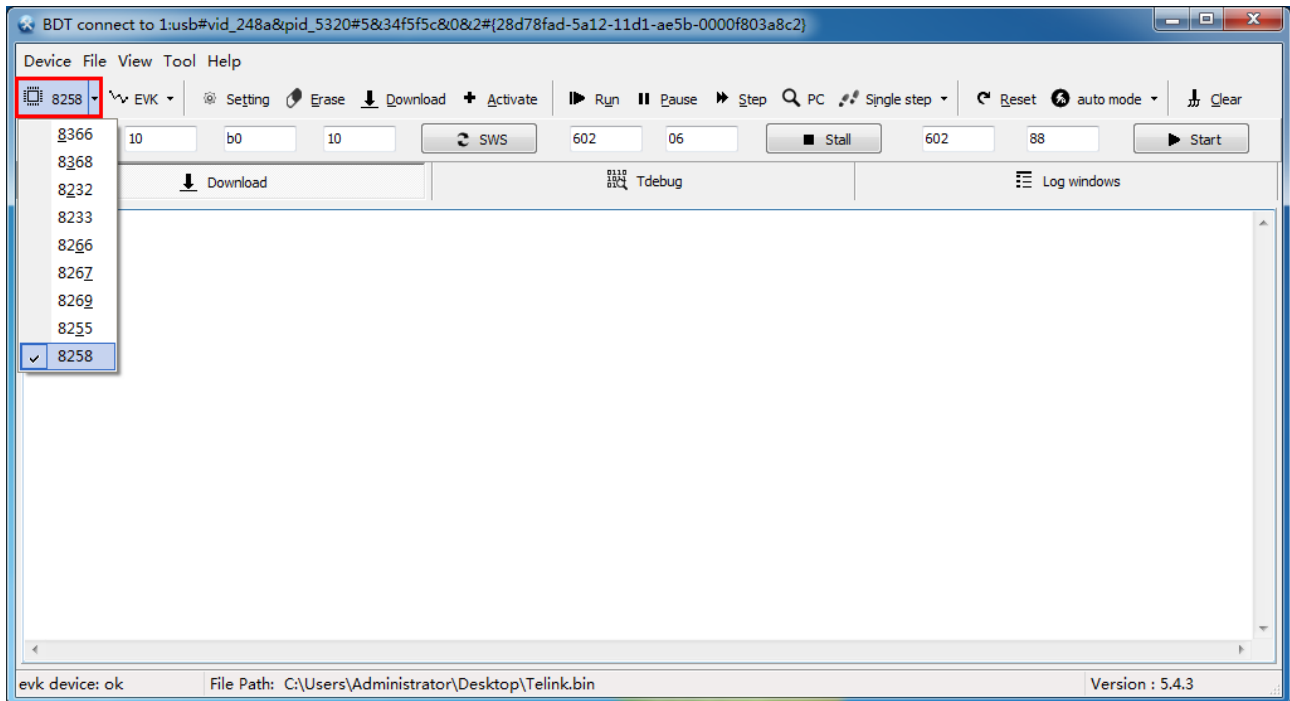


Figure 2.18: Select chip type

Step 2: Select download mode as “EVK”, the following two methods are supported.

- Method 1: Use the drop-down menu to select “EVK” download mode.
- Method 2: Press the shortcut key combination “Ctrl + W” to switch to “EVK” download mode. The drop-down menu will automatically show the current download mode after each switch.

Note:

All download modes and default options can be configured in the “config.ini” file.

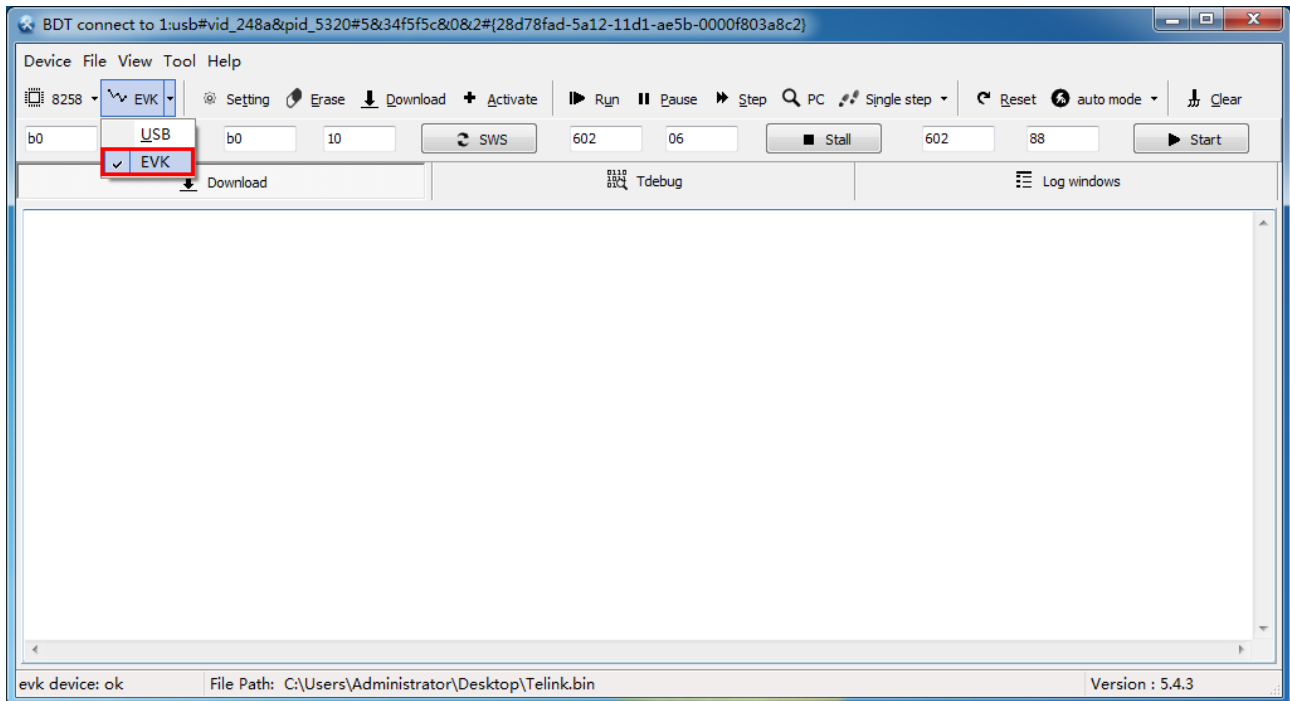


Figure 2.19: Select download mode as “EVK”

Step 3: Click the “Setting” button to open the setting dialog window, and select the “Flash” option. The user can use any of the following methods to set the flash start address offset of the target firmware storage, e.g. 0x000000 (default option).

- Method 1: Use the “Download Addr” drop-down menu to select the available offset options.
- Method 2: Enter the offset address in the edit box for “Download Addr”.

Note:

- All settings and default options can be configured in the “config.ini” file.
- The user can also configure the “SRAM”/“OTP” setting option to download the target firmware to the target area in the SRAM or OTP.

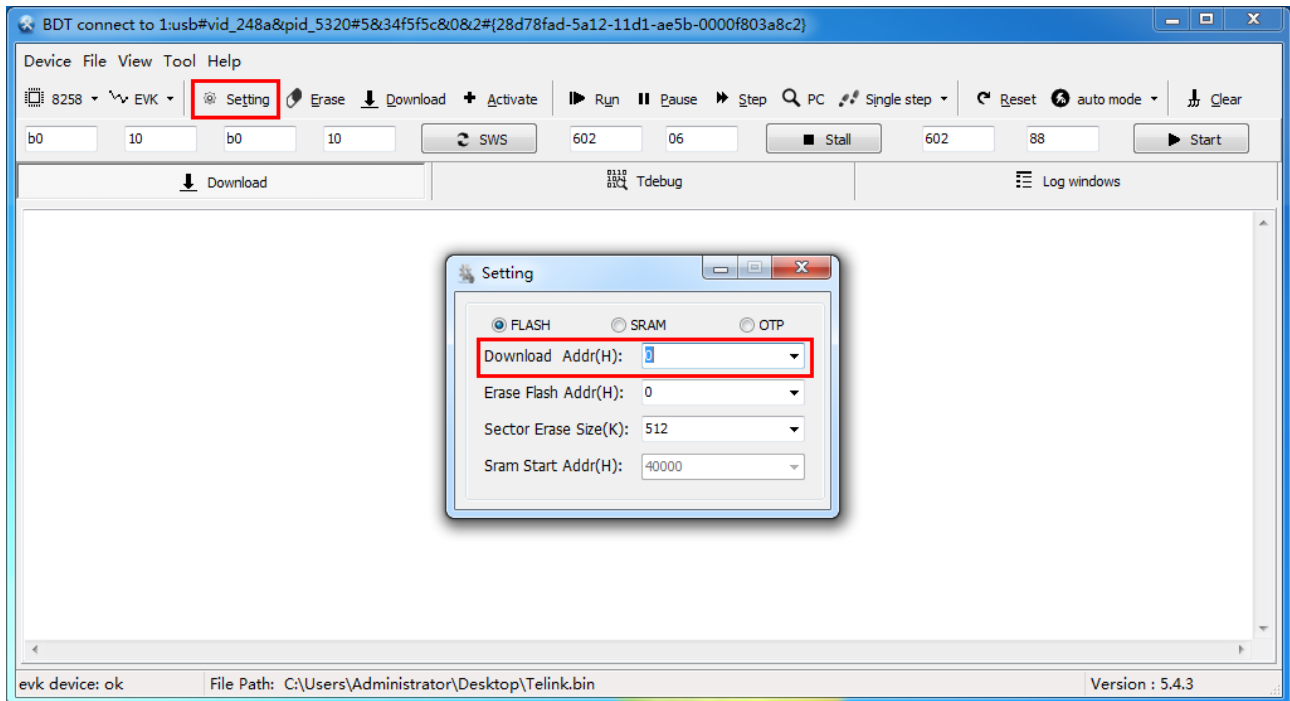


Figure 2.20: Open the "Setting" Dialog

Step 4: Select the target firmware file to be downloaded into the target board.

User can select the target file via the "File -> Open" or "File -> Reopen" (if it's available in the submenu), the file path will be visible at the bottom of the main interface.

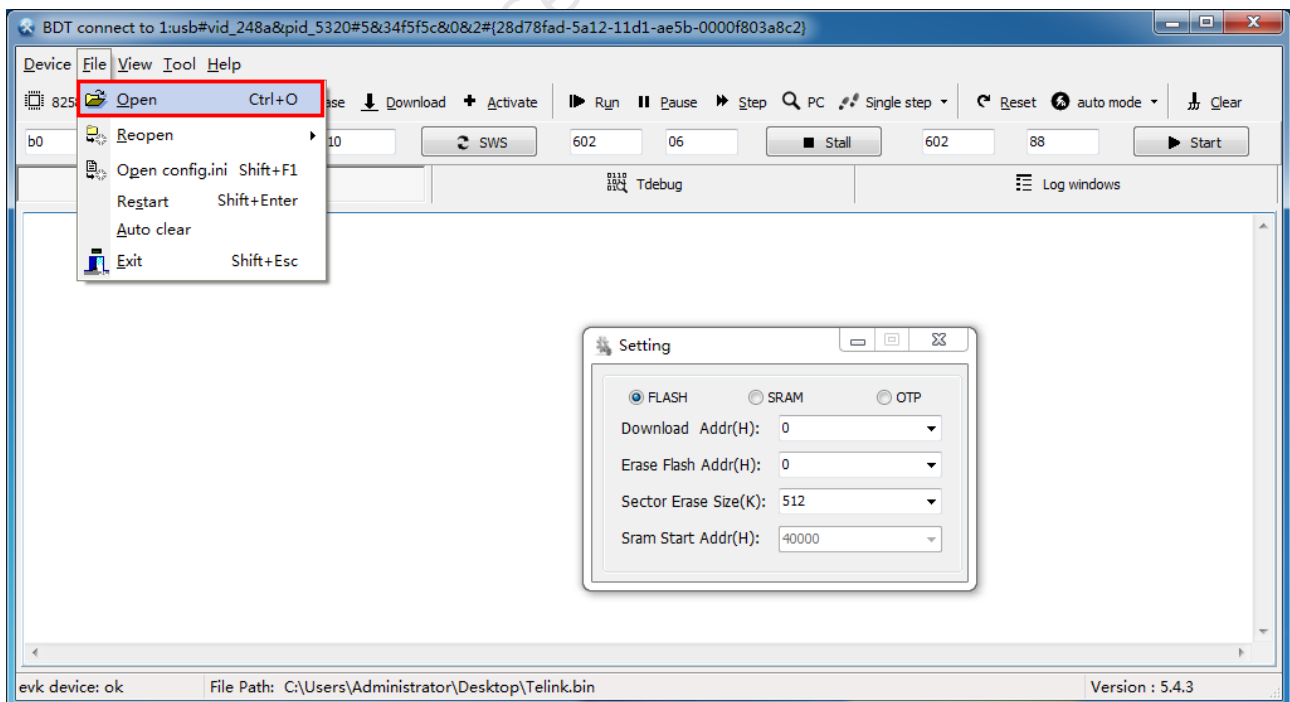


Figure 2.21: Select the target file

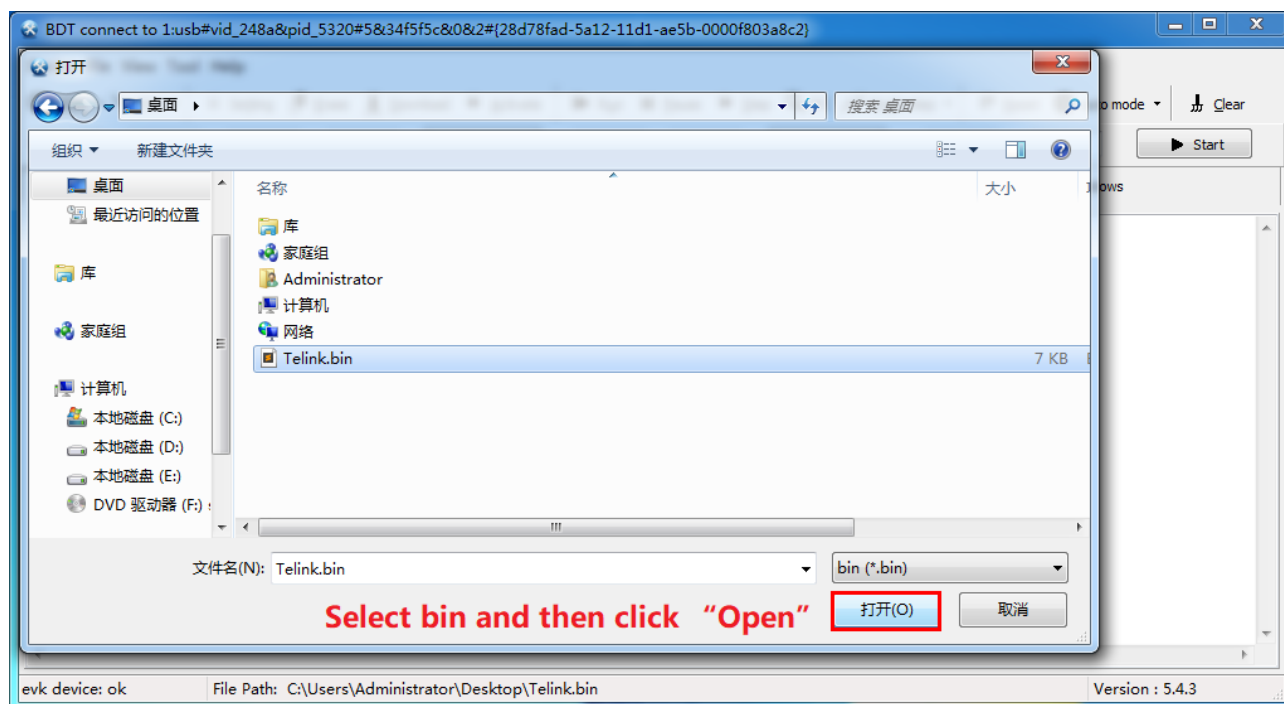


Figure 2.22: Select bin file

Step 5: Check the connection status between the target board and the PC to make sure it is working properly.

- If the connection status is normal, the left bottom of the main interface will display "usb device: ok" or "evk device: ok".
- If the left bottom of the main interface displays "usb device: not found", it indicates the connection between the target board and the PC is not available.

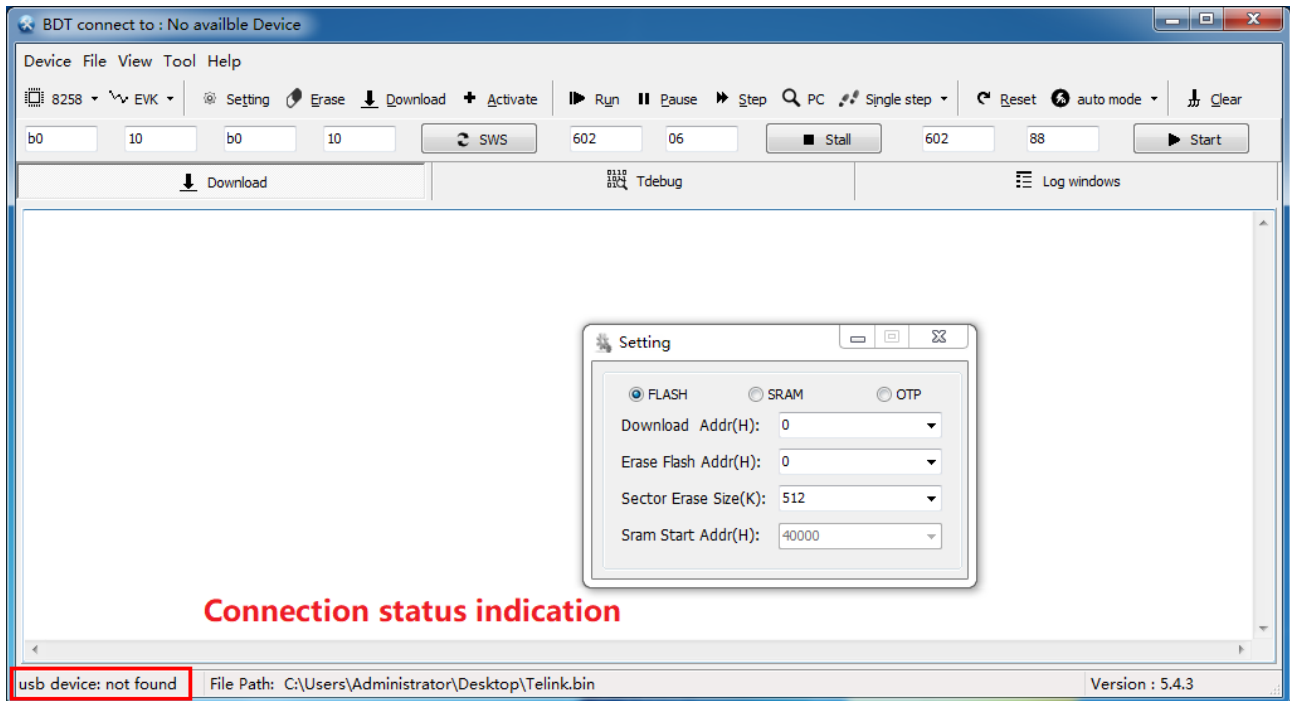


Figure 2.23: Check connection status

Step 6: Download the selected file into the target board.

By clicking the "Download" button, the selected firmware file will be downloaded into the specified flash space of the target board via "EVK" mode. The Log window will display the corresponding log information, as shown below.

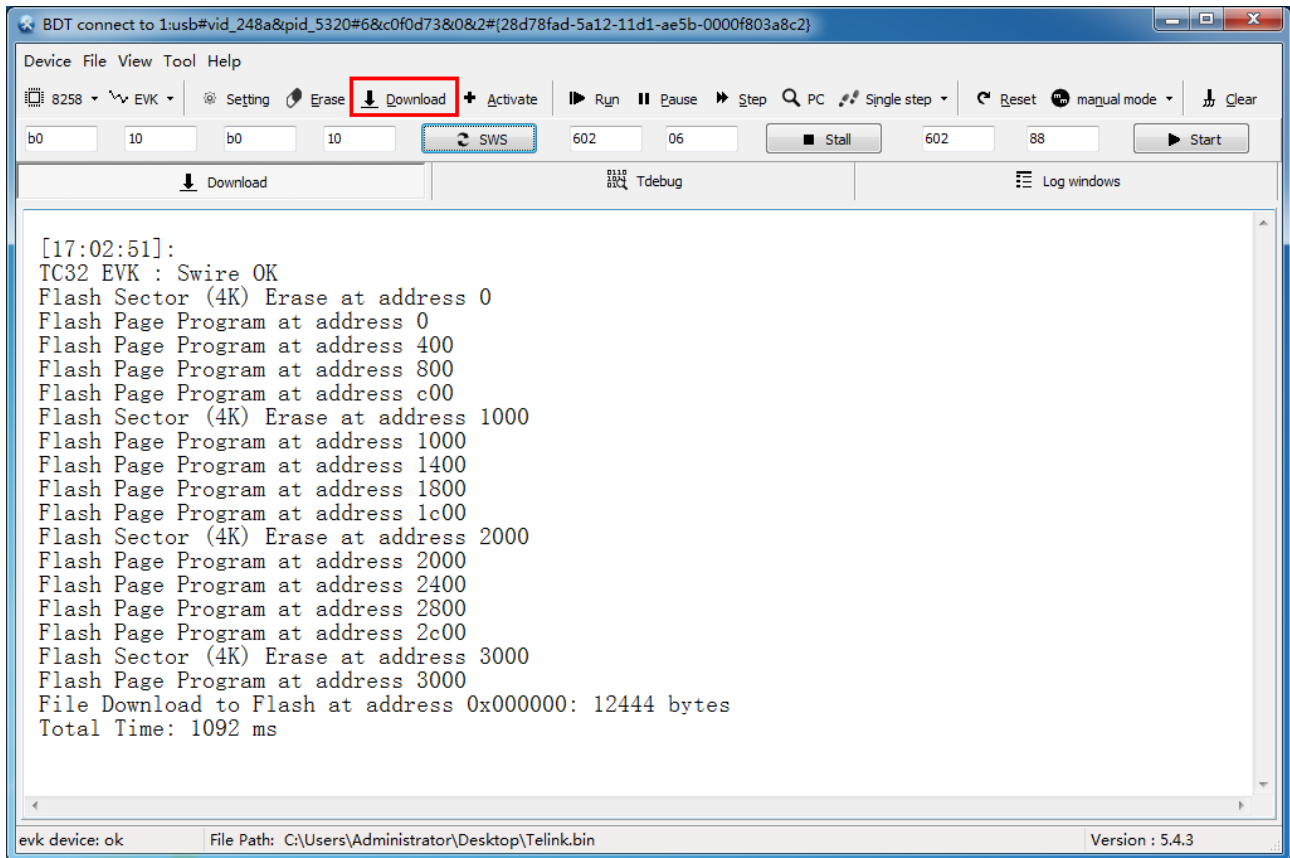


Figure 2.24: Download firmware into Flash via “Burning EVK” mode

Step 7: Reset MCU to make the newly downloaded program run without the need to power down the MCU. User can following either of the following two methods to make MCU boot and run.

- If “manual mode” is selected via the drop-down menu, after the firmware burning, the user can click the “Reset” button in the top right corner of the tool to manually reset the MCU.

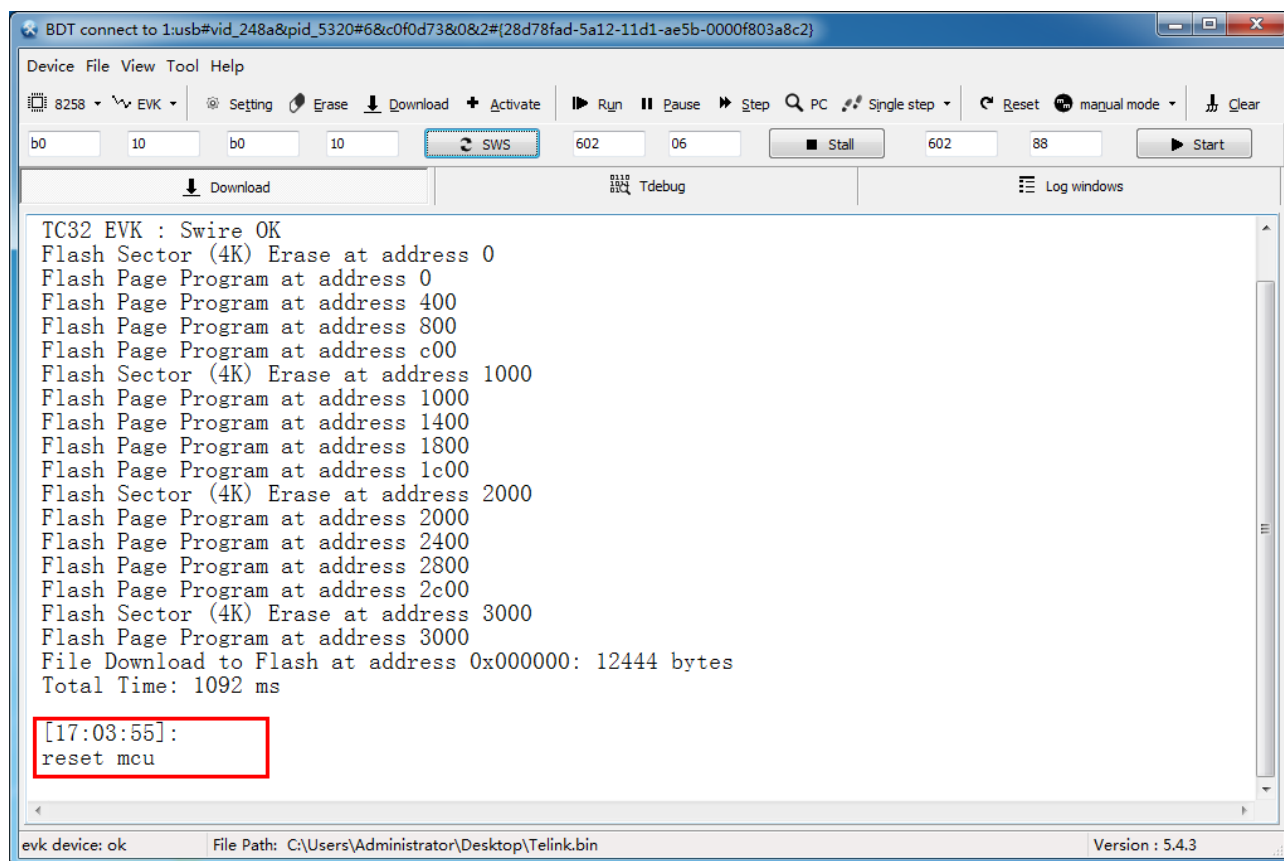


Figure 2.25: Manual reset MCU

- Users can also select "auto mode" via the drop-down menu before firmware burning, so as to enable auto reset after firmware burning.

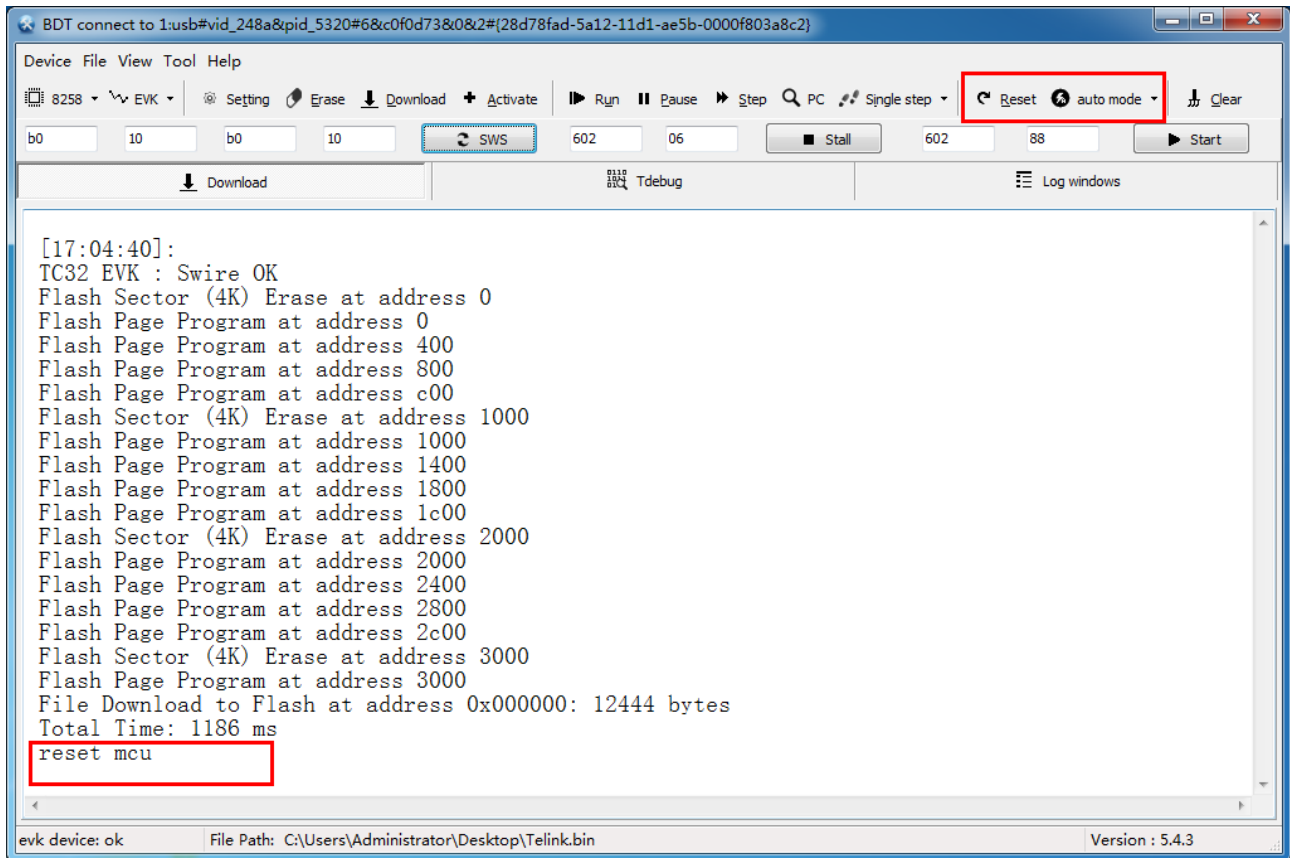


Figure 2.26: Auto reset MCU

2.1.5 Download firmware into “SRAM” or “OTP”

By default, the “Setting” option is set as “Flash”, i.e. by clicking the “Download” button, the target firmware will be downloaded into the “Flash” space of the target board.

User can also change the “Setting” option to “SRAM” or “OTP”. In this case, by clicking the “Download” button, the target firmware will be downloaded into the target board’s “SRAM” or “OTP” space correspondingly.

For the TLSR8258, the SRAM start address of the target firmware is 0x040000. For other Telink’s SoCs, the SRAM start address of the target firmware is 0x008000. Of course, the user can set other addresses in the edit box as needed.

After the target file is downloaded successfully, the target board will not automatically run the new firmware in “manual mode”, so user needs to click the “Reset” button to make program boot. User can also select “auto mode” via the drop-down menu before downloading, so as to enable auto MCU reset.

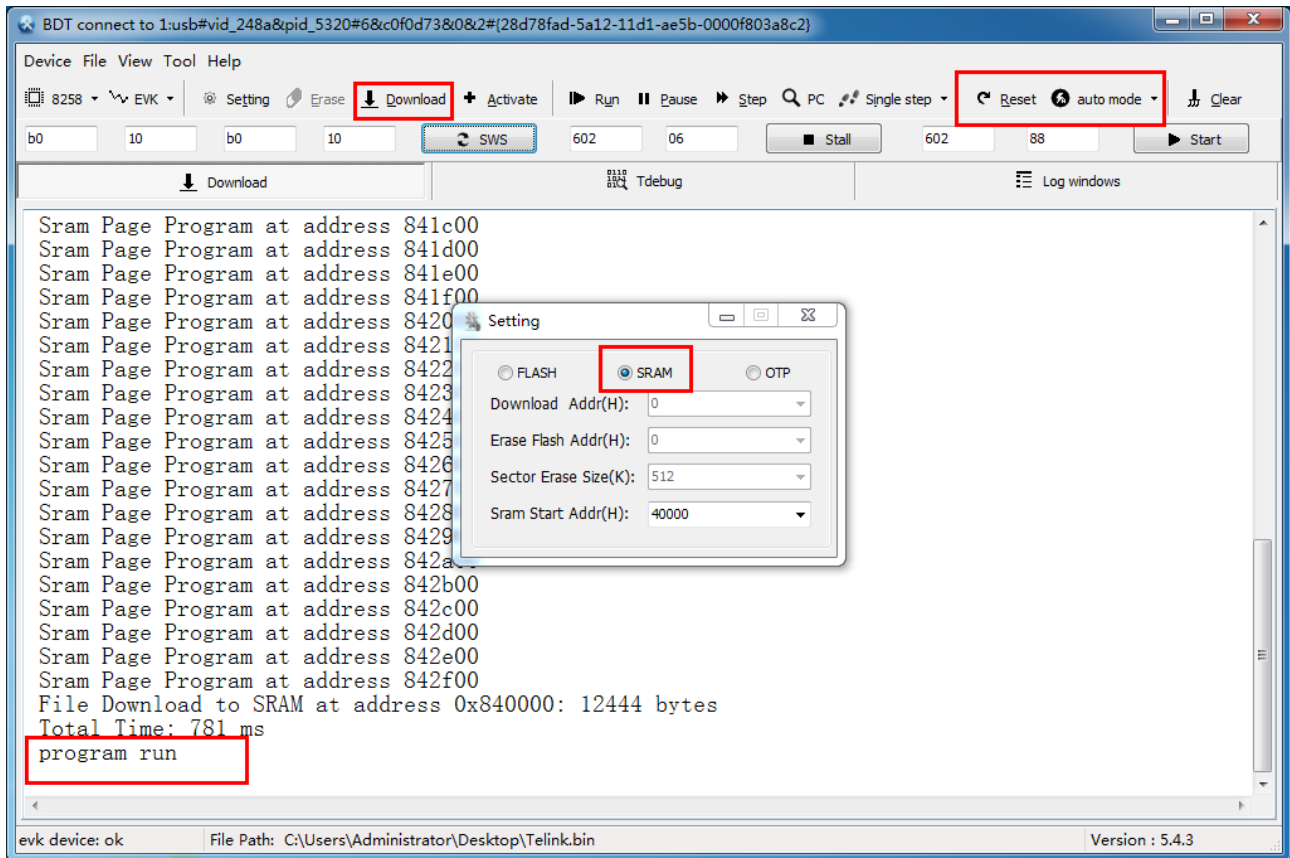


Figure 2.27: Download firmware to “SRAM” via “auto mode”

If target board supports OTP function, user can download firmware into “OTP”. Refer to [section 5 Function Support List](#) for details.

2.1.6 Multi-address Download

Sometimes, it is necessary to download more than one firmware into different addresses of Flash, SRAM or OTP.

For example, users can follow the steps below to download “Telink.bin” to the addresses “0x000000” and “0x005000” of Flash.

Step 1: Select the target memory space (such as Flash) to store the firmware to be downloaded.

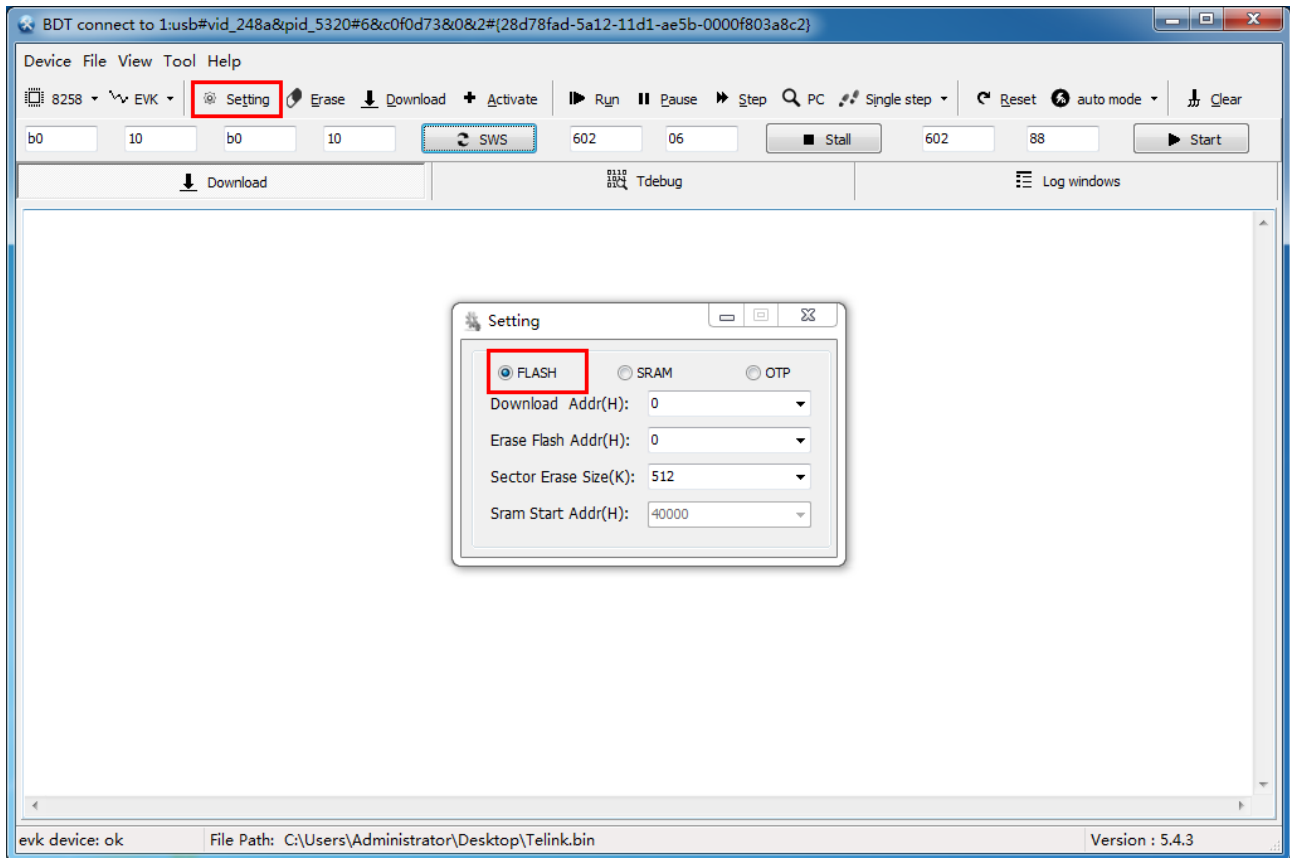


Figure 2.28: Select target space

Step 2: Click "Tool -> Multi-address download" or press the shortcut key "Ctrl+P", so as to open the "Multi-address Download" interface.

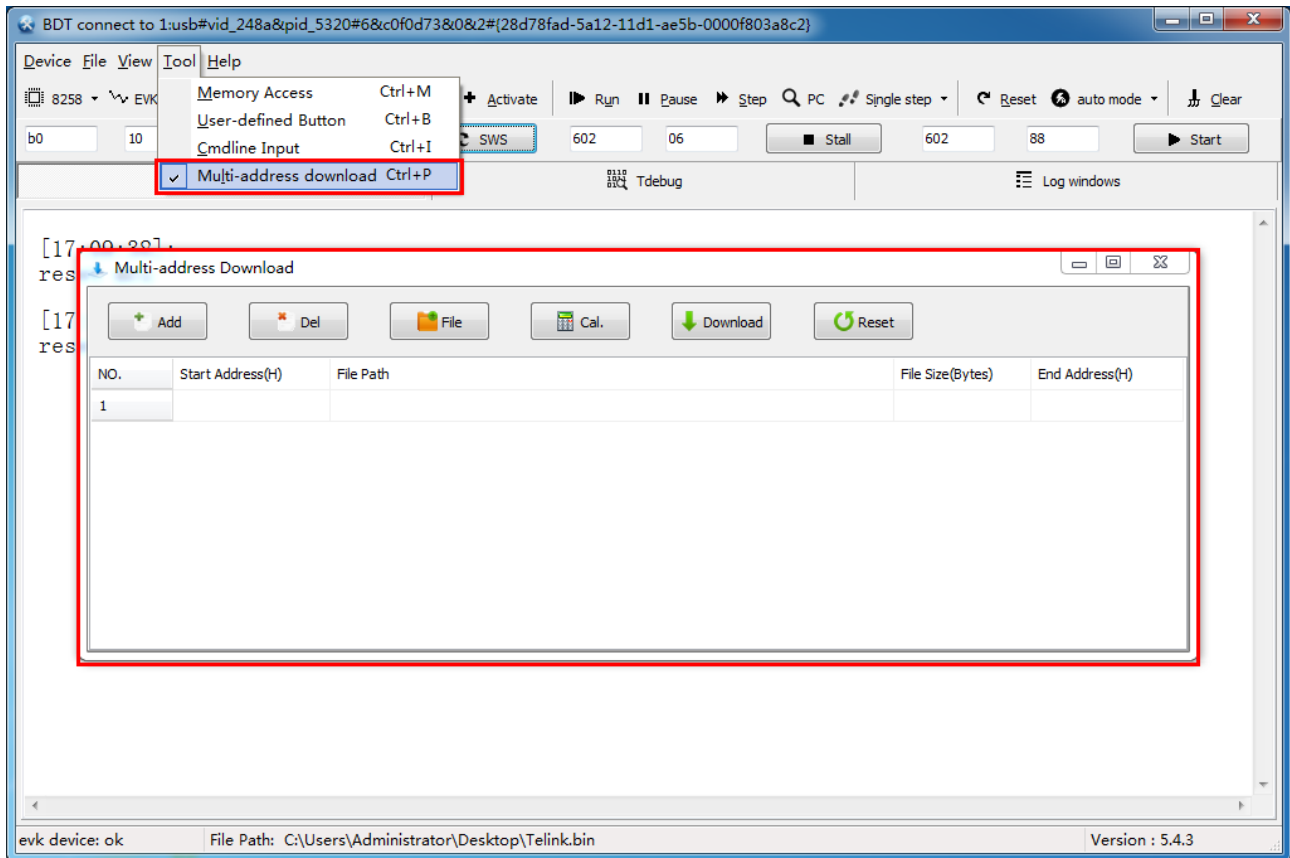


Figure 2.29: Open “Multi-address Download” interface

Step 3: Click the “Add” button so as to add firmware file and set target address.

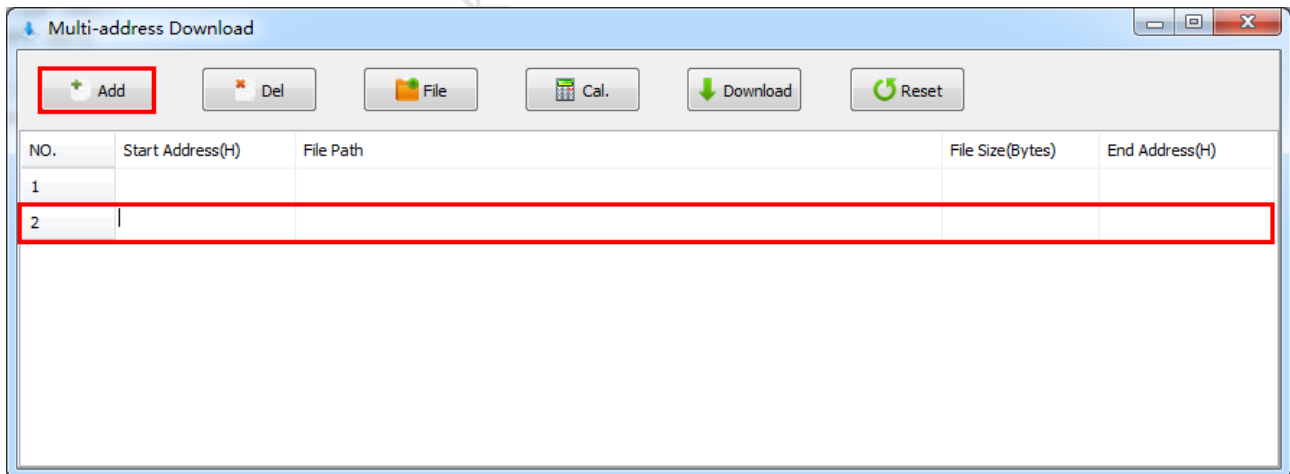


Figure 2.30: Add firmware file and set target address

Step 4: Add the path of firmware 1 to the list and set the offset of starting address to store this firmware.

- (1) First, select the first line (No.1), and click the “File” button to add target bin file (e.g. “Telink.bin”) to the list. The corresponding directory will be shown in the “File Path” column.

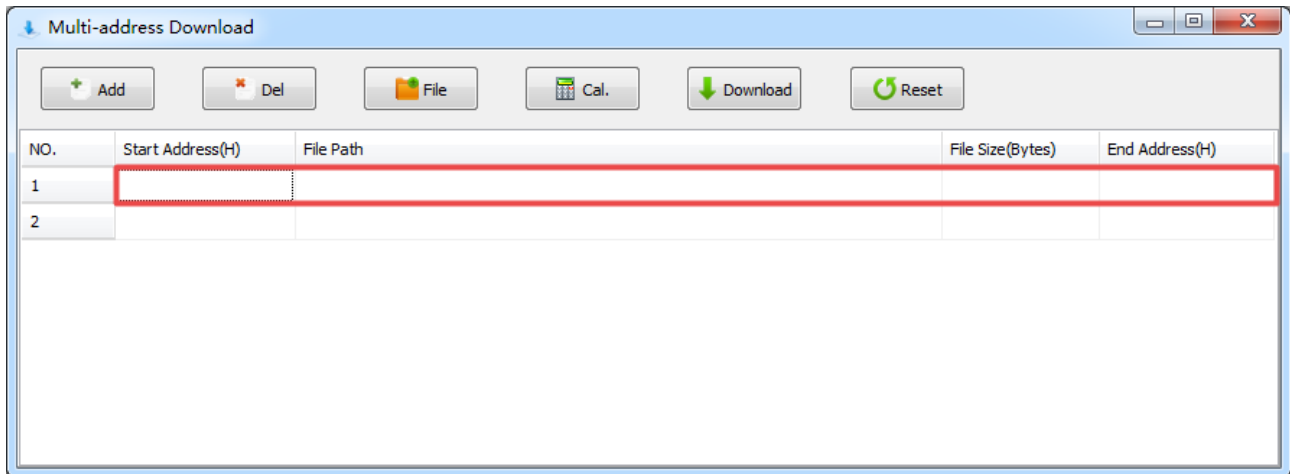


Figure 2.31: Add firmware

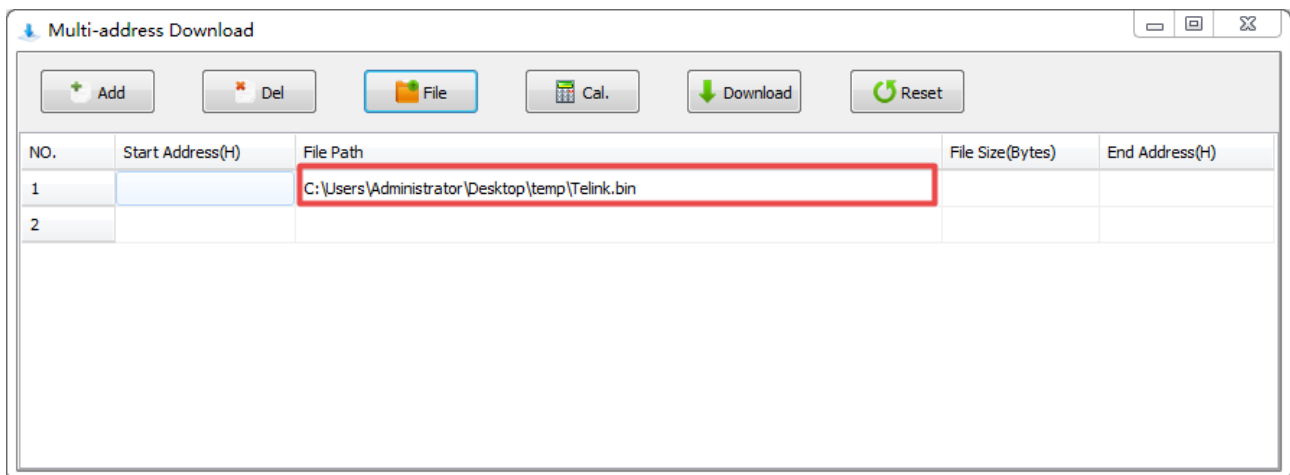


Figure 2.32: Show bin file path

(2) Then set the offset of starting Flash address to store this firmware in the "Start Address(H)" column.

To ensure that the target area for firmware storage is valid, the tool will automatically calculate the size and end address of the firmware, and then show the calculation result in the "File Size(Bytes)" and "End Address(H)" column.

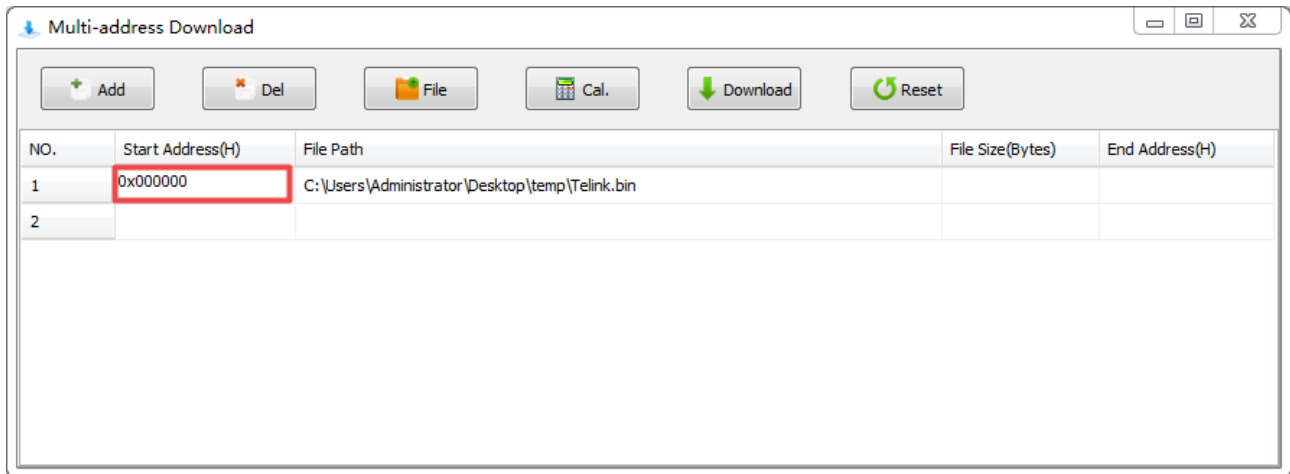


Figure 2.33: Set offset address

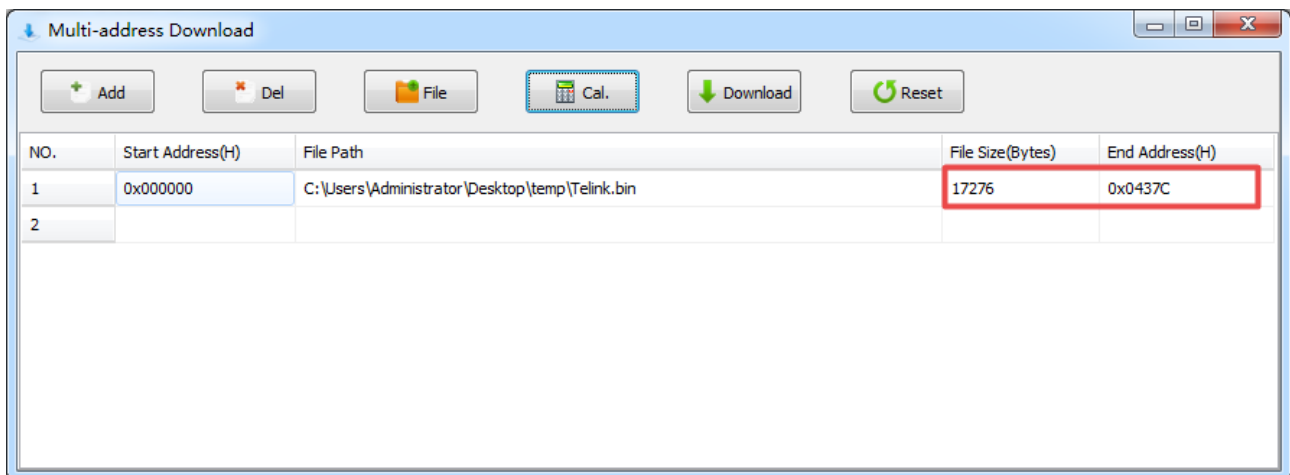


Figure 2.34: Show calculation results

Step 5: Refer to step 4 to add the path of firmware 2 to the list and set the offset of starting address to store this firmware.

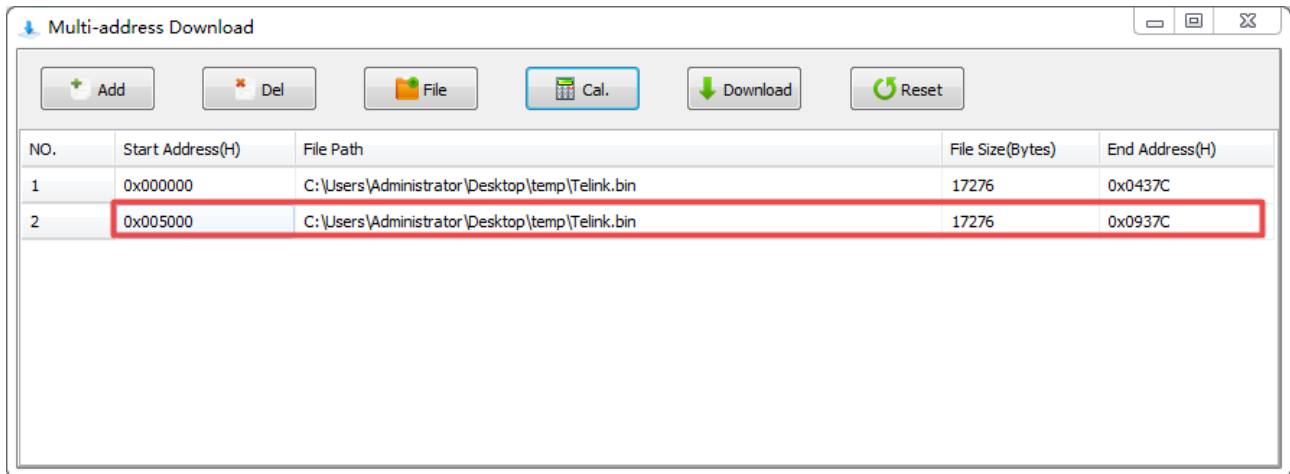


Figure 2.35: Add the path of firmware 2

Step 6: Click the “Download” button to download the added firmware 1 and firmware 2 into the specific memory location of the target board.

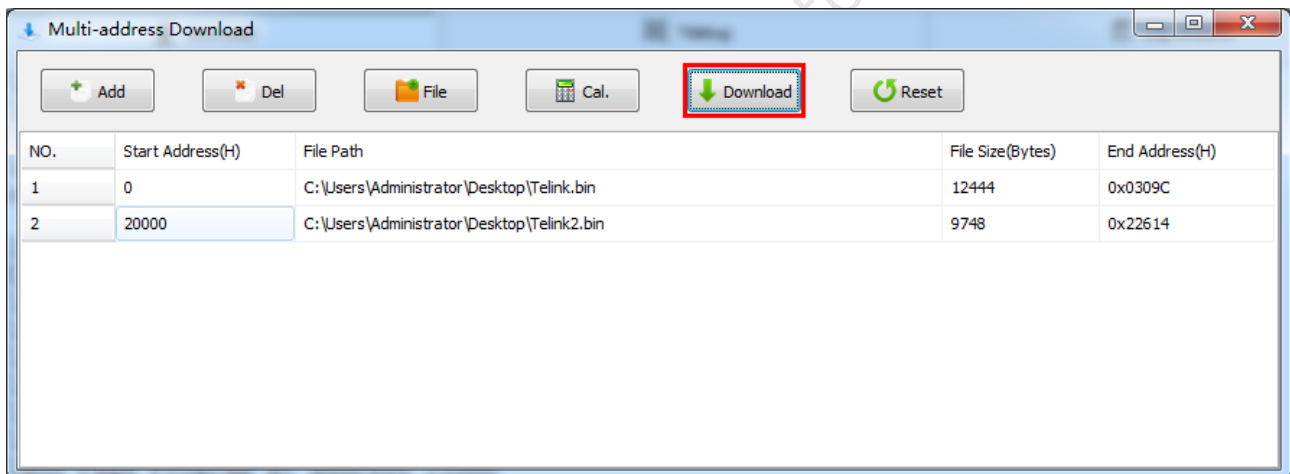


Figure 2.36: “Download” button

```
Flash Page Program at address 1800
Flash Page Program at address 1c00
Flash Sector (4K) Erase at address 2000
Flash Page Program at address 2000
Flash Page Program at address 2400
Flash Page Program at address 2800
Flash Page Program at address 2c00
Flash Sector (4K) Erase at address 3000
Flash Page Program at address 3000
File Download to Flash at address 0x000000: 12444 bytes
Total Time: 1186 ms
```

Figure 2.37: Start downloading firmware 1

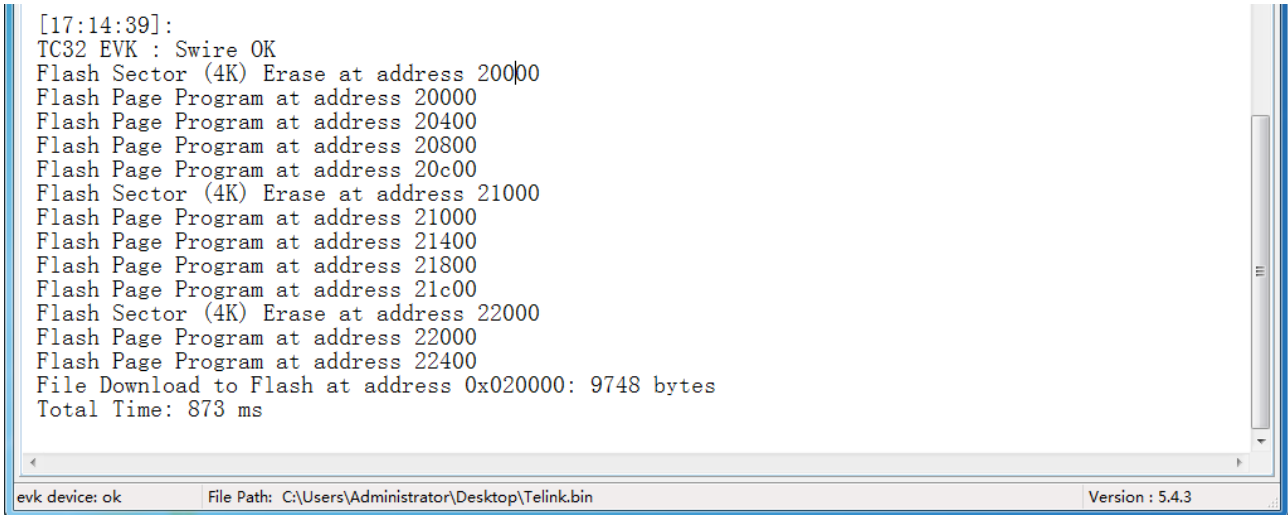


Figure 2.38: Start downloading firmware 2

Step 7: After downloading all firmware into the target board, click the “Reset” button to make program run without the need to power cycle the target board.

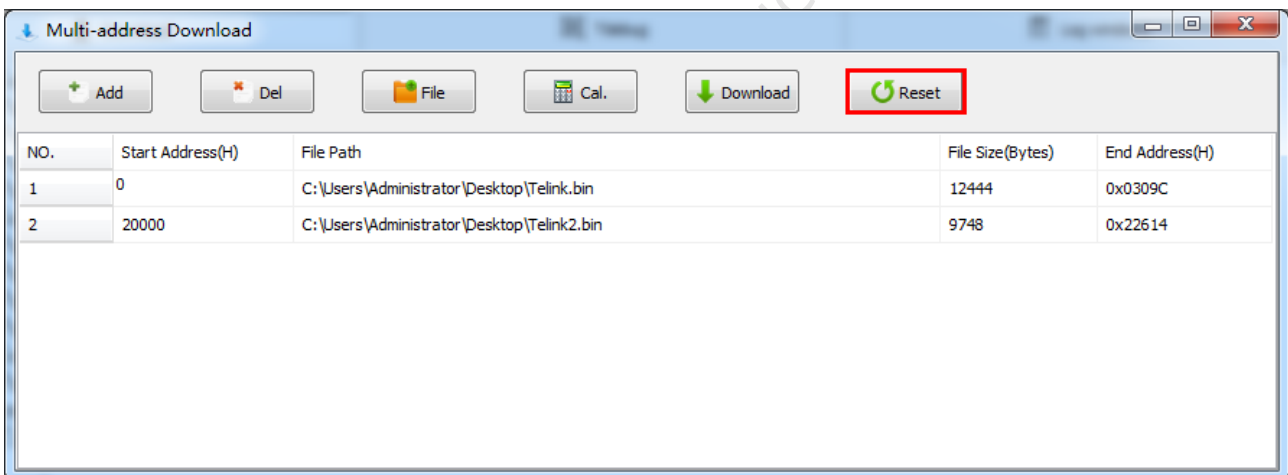


Figure 2.39: Reset

2.1.7 OTP Start Information Download

- (1) Configure OTP clock, take the chip 8208 as an example, OTP address 0x3ff8 write data 0x060301bf.
 - Function: Write the MCU digital register 0x603 to 0x01 to configure OTP clock.
- (2) Start code, take the chip 8208 as an example, OTP address 0x3ff0 write data 0x060298bf.
 - Function: Write the MCU digital register 0x602 to 0x98, the MCU starts the execution of the code, and the program starts to run.
- (3) Move the code into RAM, take the chip 8208 as an example, OTP address 0x3ff4 write data 0x0001f03f.

- Function: MCU moves the contents of the first 7936 bytes in OTP from address 0 to address 0 of RAM.
- Data length = $0x1f0 * 16 = 7936$ bytes

The following method can be used to calculate the size of the program that the OTP needs to move into ram.

When the OTP is powered on for the first time, both the vector and ram code fields need to be moved to the ram. The calculation formula for the size of these two fields is:

The base address of the LMA of the ram code + ram code size (the reason why the vector size + ram_code size is not used for calculation is that there may be alignment factors that cause the size to be calculated incorrectly)

Because the length information in the start information is: actual length/16. Therefore, the general calculation formula of the length information in the start information is summarized as follows:

((The base address of the LMA of the ram code + ram code size) (16 bytes alignment)) / 16

So the final calculation formula of the following lst file is: $0x220 + 0xAEC$, the size after 16 bytes alignment is 3344, $3344 / 16 = 209(0xd1)$

Just write the data $0x0000d13f$ to the corresponding OTP address $0x3ff4$.

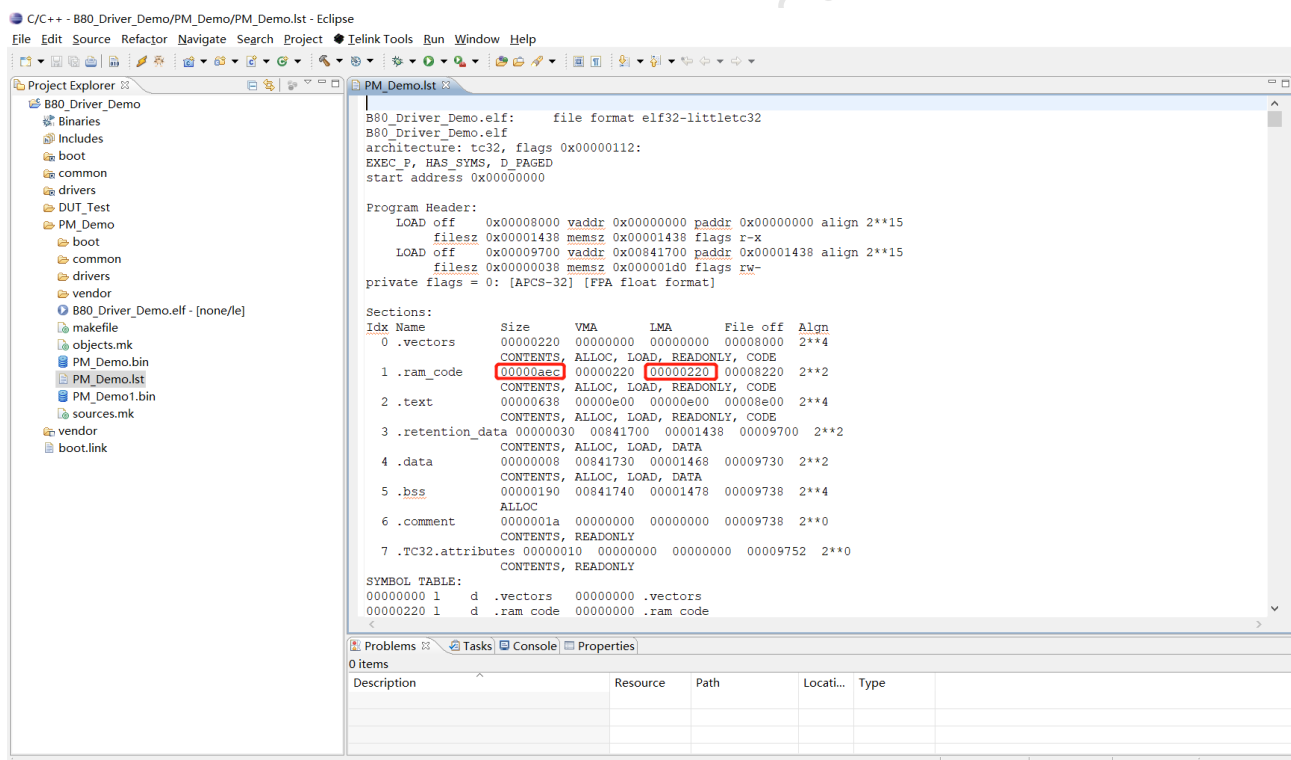


Figure 2.40: lst file

Taking the chip 8208 as an example, users can download the OTP start information according to the following steps.

Step 1: Open the BDT tool and select the 8208 chip.

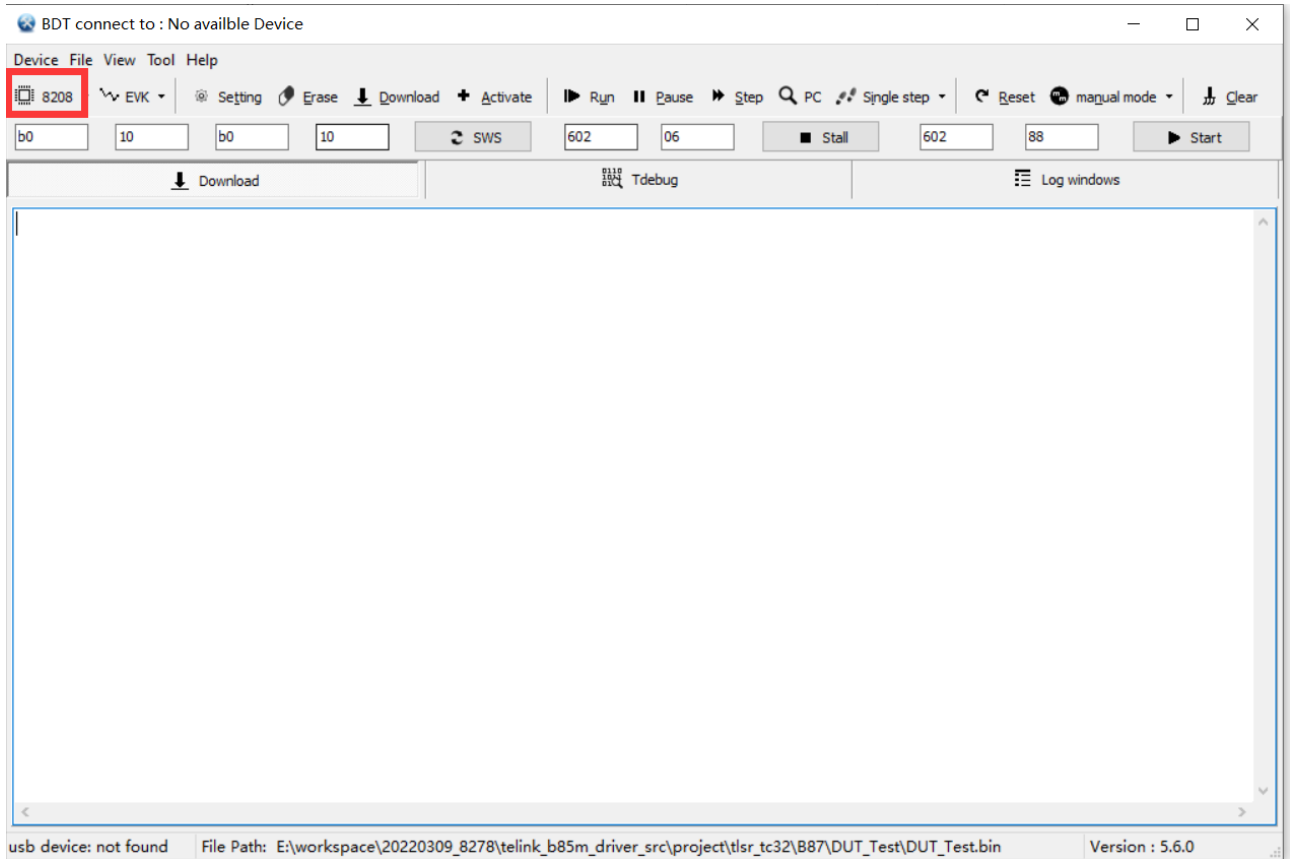


Figure 2.41: Select chip

Step 2: Click Tool -> Memory Access, enter the start information and its length to be written in the OTP address, and press the "Enter" key in the data edit box, the corresponding information will be burned into the OTP address.

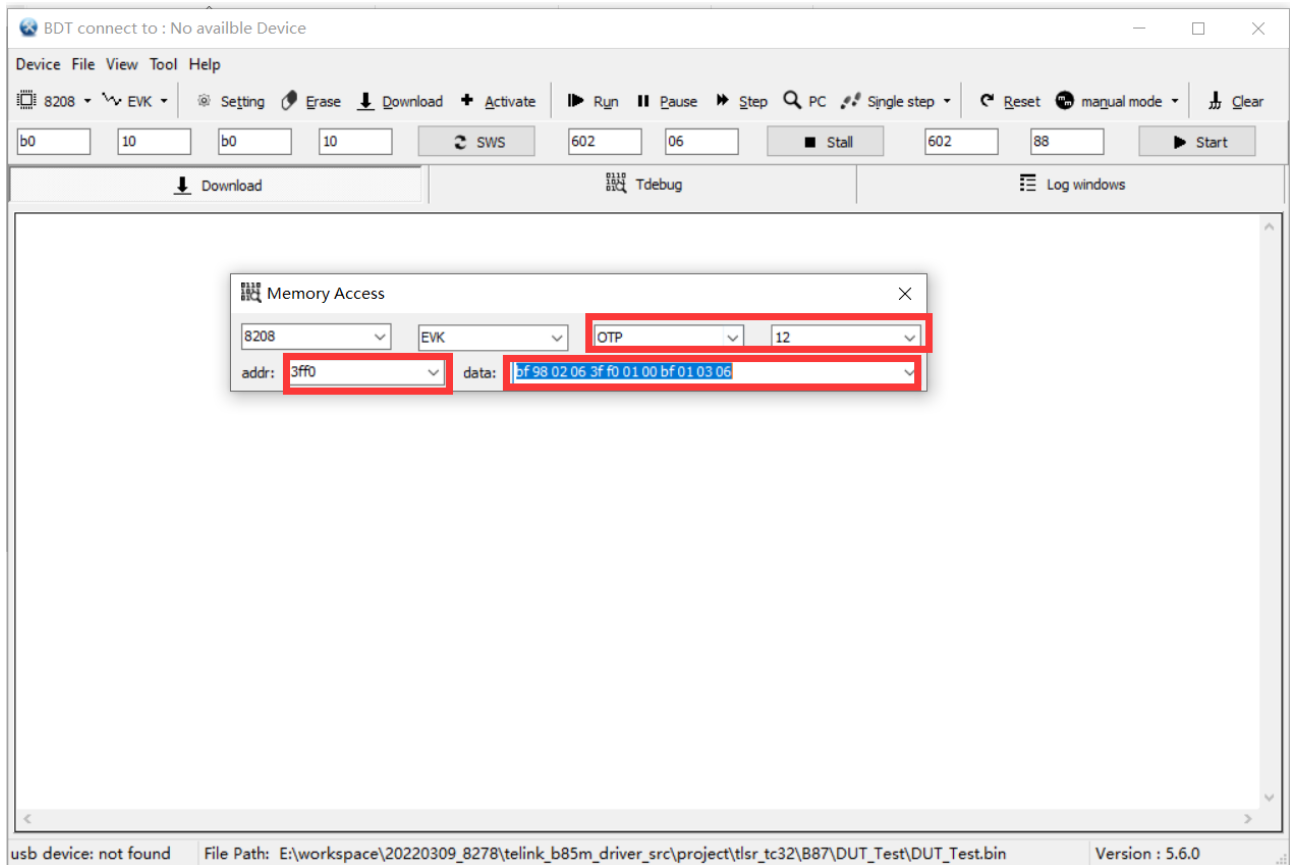


Figure 2.42: OTP start information download

2.2 Erase Flash Sector

The “Erase Flash Sector” function is used to erase specific Flash space starting from specific address in 4kB units.

For example, to erase 64kB Flash space starting from address 0x040000, user can follow the steps below.

Step 1: Click the “Setting” button to open the Setting dialog. Select the “Flash” setting option.

To set the starting address of Flash space to be erased, e.g. 0x040000 (default option: 0x000000), user can either select the offset option (if available) via the drop-down menu of “Erase Flash Addr” or directly input offset address in the edit box.

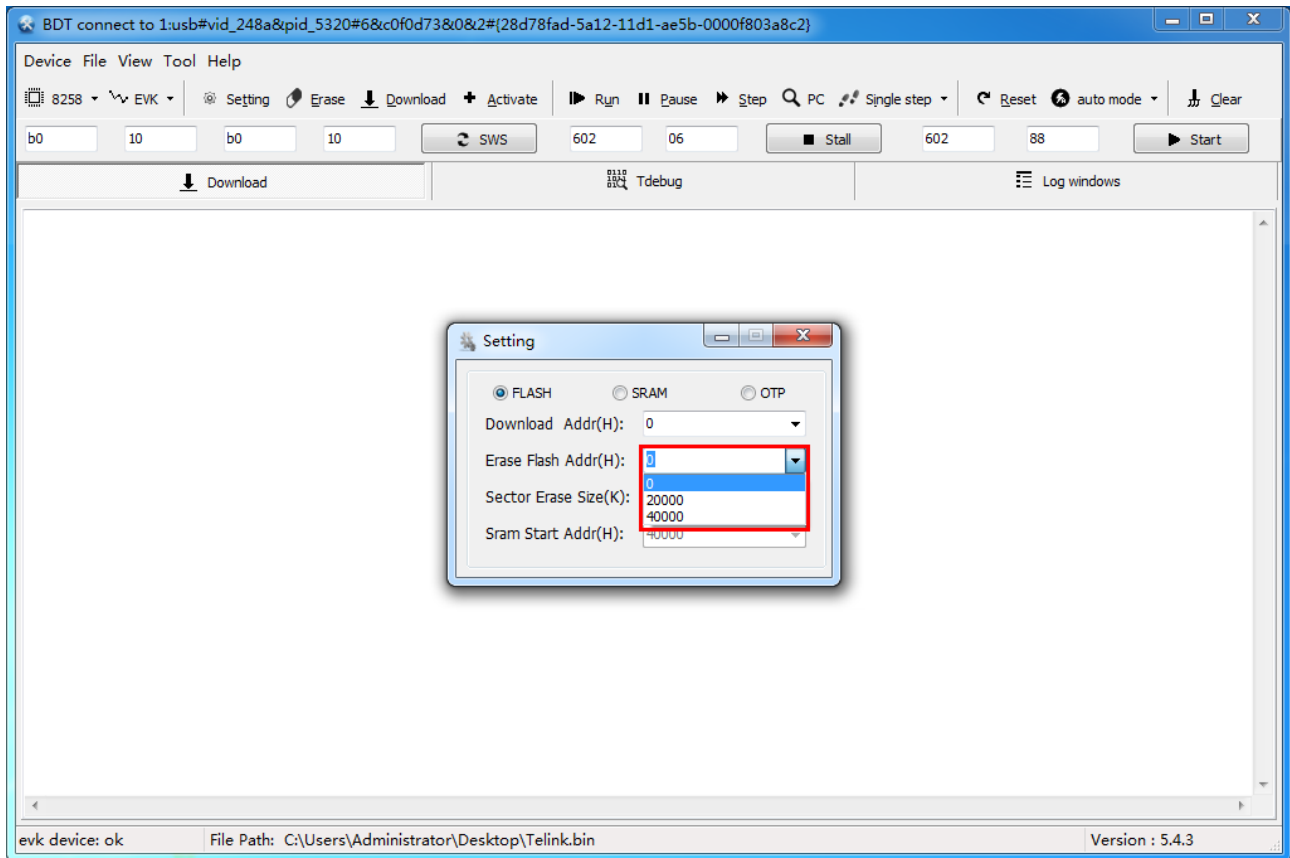


Figure 2.43: Select starting address

Step 2: To set the size of Flash space to be erased in 4kB units, e.g. 64kB (default option: 512kB), user can either select the size option (if available) via the drop-down menu of "Sector Erase Size" or directly input the size in the edit box.

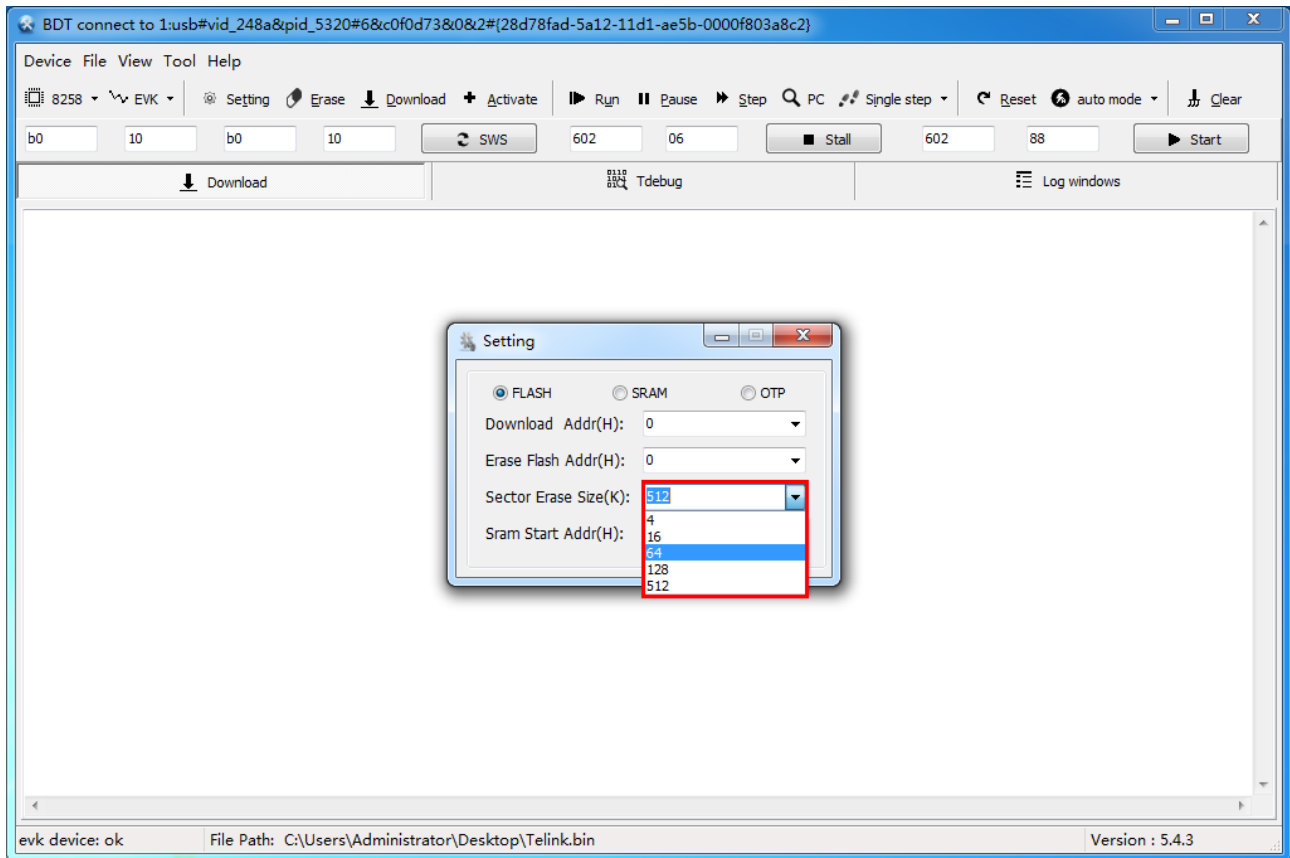


Figure 2.44: Set erase size

Step 3: Click the “Erase” button to start erasing the specified Flash space of the target board.

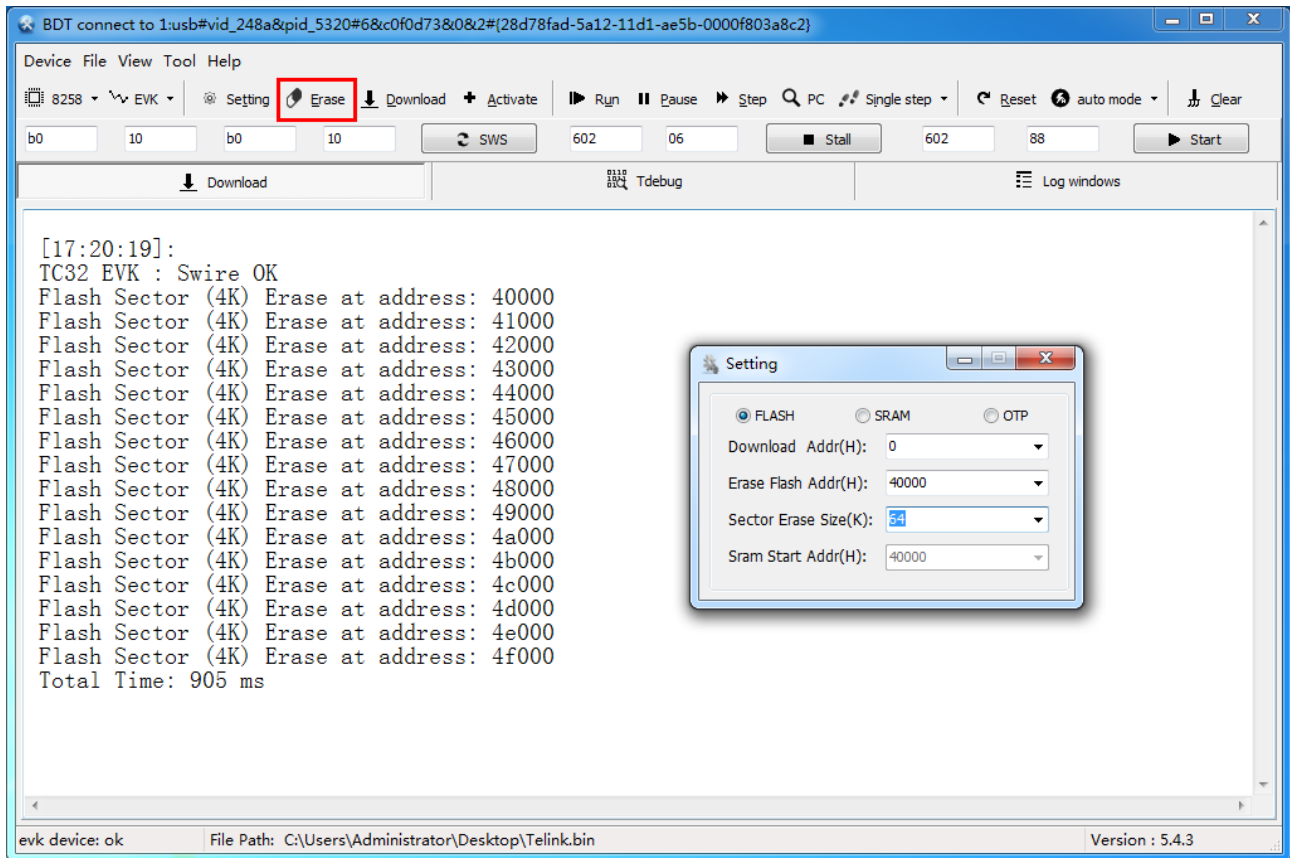


Figure 2.45: Erase 512kB flash space starting from 0x000000

2.3 Flash De-protection

After the Flash is write-protected, it will lead to unsuccessful erasing and burning, so it is necessary to remove the protection before operating the Flash. The steps to remove the protection from Flash are as follows. (This function has been supported since version v5.7.0)

Step 1: Open the BDT tool and select the chip type.

Step 2: Click the "Unlock" button, and "Flash remove protection successfully!" appears in the log, indicating that the protection was removed successfully.

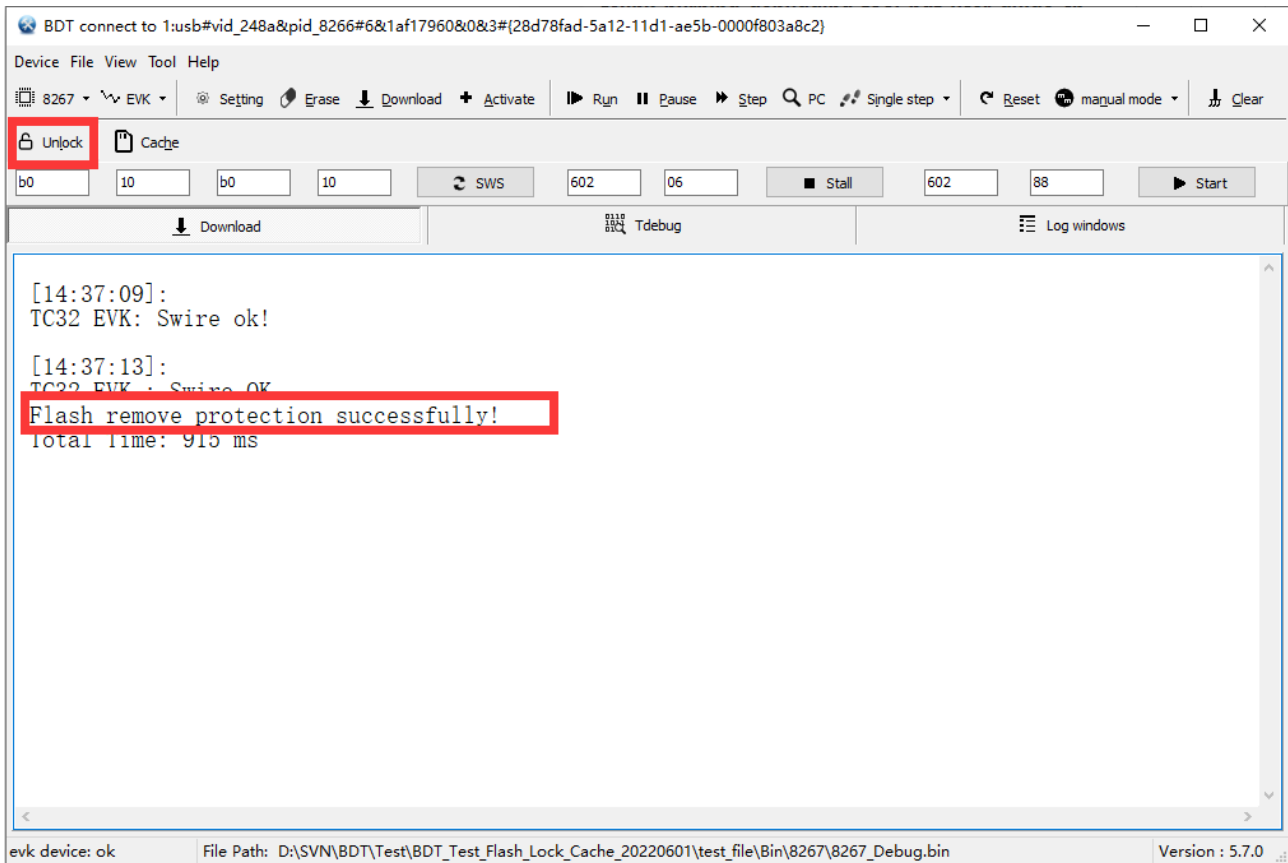


Figure 2.46: Flash remove protection

2.4 Cache Analyze

When the chip is powered on to carry the code, if the code to be run is not in the cache, it will automatically read the function from Flash or OTP into the cache. This function is mainly to analyze which address of the flash the cache move code fetches data from, and what the data content is, and compare the analyzed data with the original data to determine whether the cache move code is correct. The premise of this function is that the chip has downloaded the program and runs it, and then parses it through the following steps. (This function has been supported since version v5.7.0)

Step 1: Open the BDT tool, select the chip type and choose the firmware that the chip needs to download.

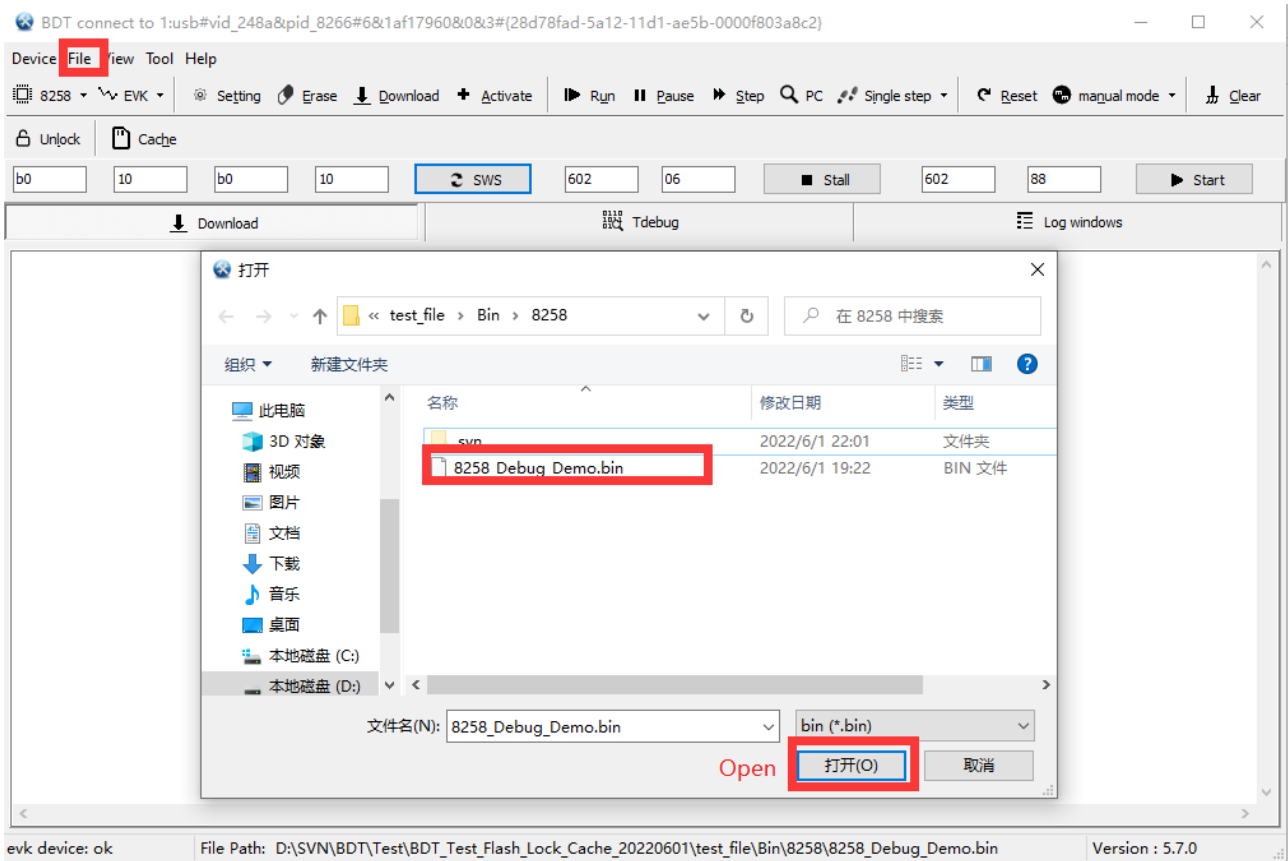


Figure 2.47: Bin file selection

Step 2: Click the "Cache" button. In the figure below, the marker "1" is the flash address where the moving code is located; the marker "2" is the data stored in the corresponding flash address; the marker "3" represents the test result. "OK" means the data in the moving code is correct; "FAIL" means the data in the moving code has error.

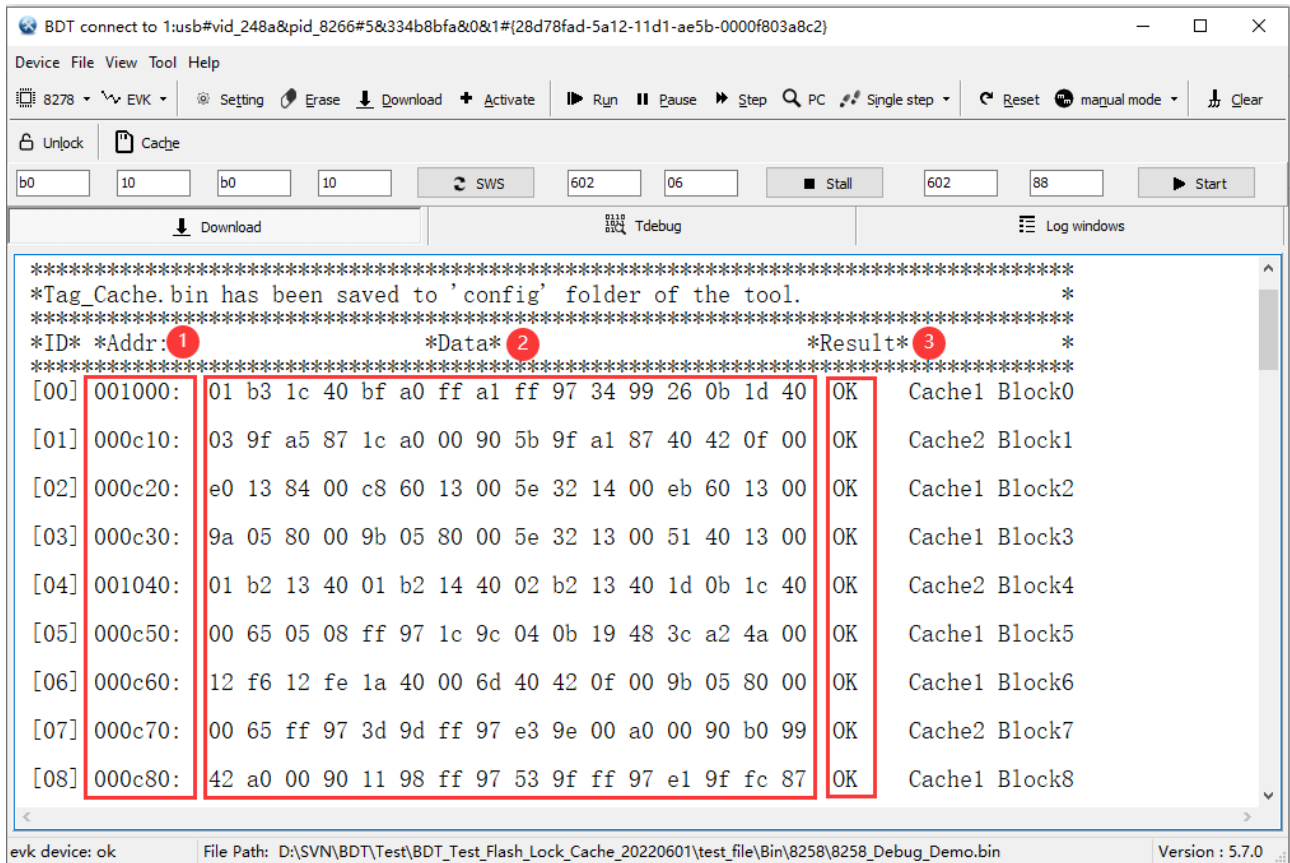


Figure 2.48: Cache data is correct

In the event of a cache data error, the "rawdata" represents the value of the correct data corresponding to the flash address; the "status" represents the result of comparing "rawdata" with the data in the cache. If the "status" is "T", the data of them is the same; if the "status" is "F", the data of them is inconsistent.

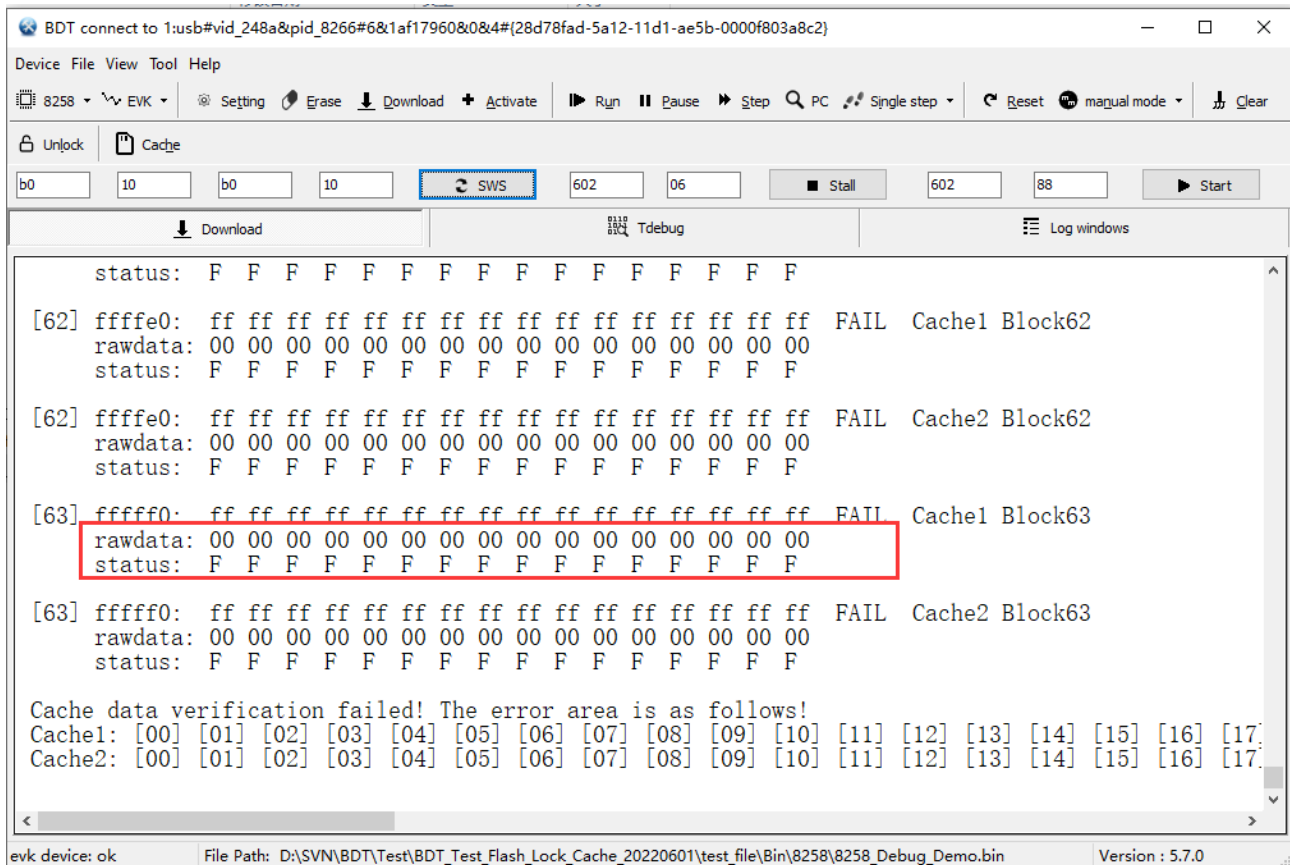


Figure 2.49: Cache data has error

2.5 Activate MCU

The function of "Activate MCU when communication with target board failed" only applies to Swire connection between "Burning EVK" and target board in "EVK" mode, i.e. it does not support "USB" mode or USB connection between "Burning EVK" and target board in "EVK" mode.

When firmware burning fails, please ensure the target board is connected with "Burning EVK" via Swire, and then click the "Activate" button to enable this function to activate the MCU.

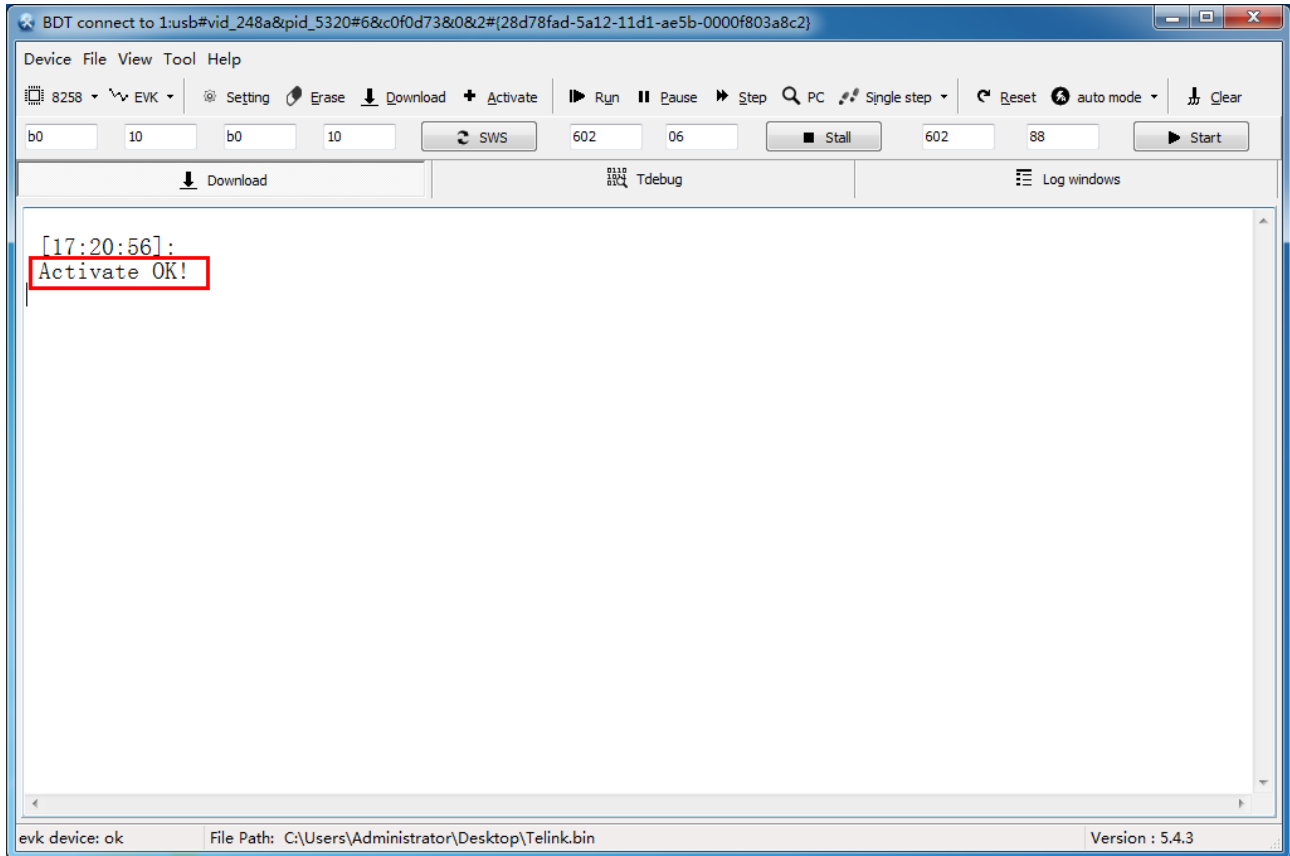


Figure 2.50: Activate MCU

2.6 Debug

After MCU starts running, to access memory space (Flash/Core/Analog/OTP), users can directly press the shortcut key combination "Ctrl + M" or click "Tool -> Memory Access" to open the "Memory Access" window as shown in the figure below.

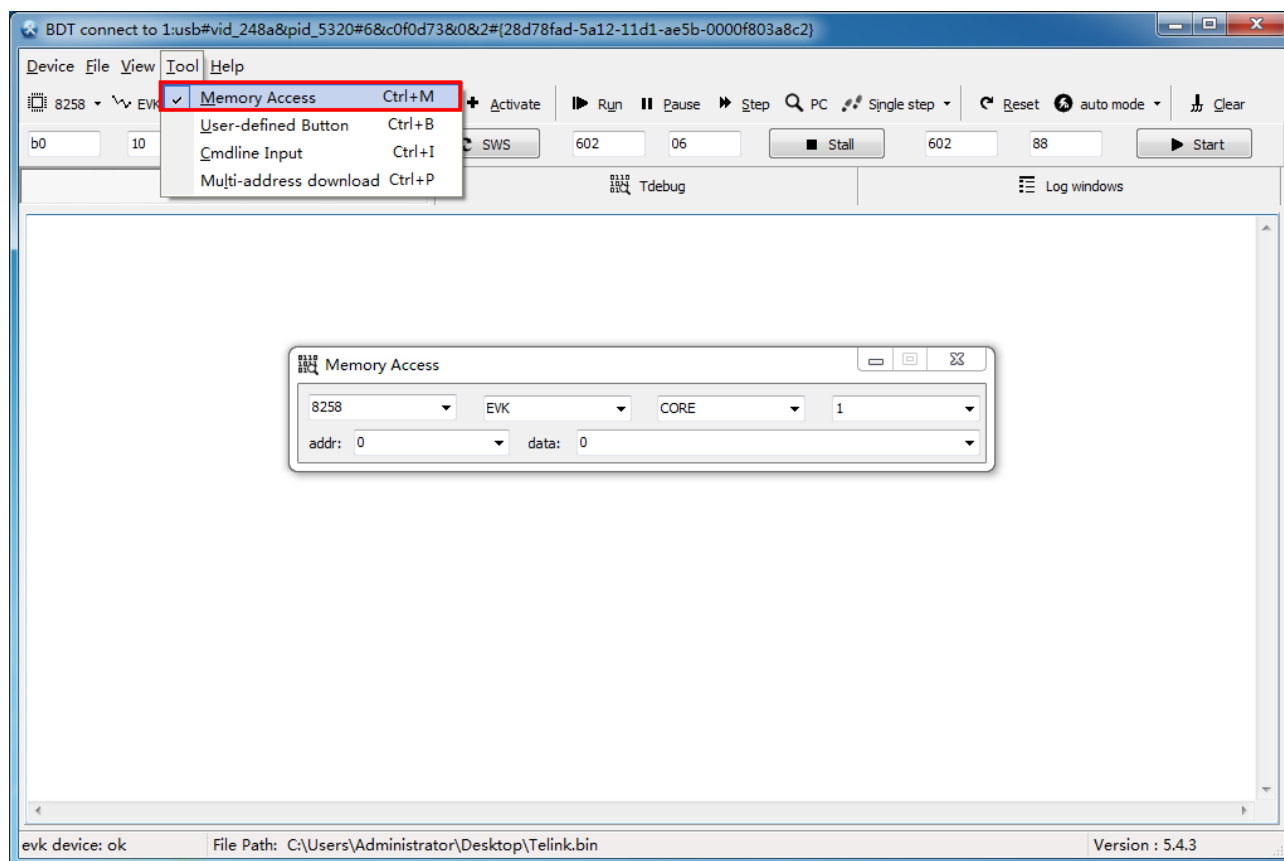


Figure 2.51: “Memory Access” interface

Table 2.1: Introduction of Memory Access

Items	Value list	Default	Shortcut
MCU Type	8366/8368/8266/8267/8269/8232/8233/8x5x	8258	Ctrl+Q
Communication Mode	EVK(Burning EVK)/USB	EVK	Ctrl+W
Operation Destination	Flash/Core ¹ /Analog ² /OTP	Core	Ctrl+E
Operation Size (Unit: Byte)	1/2/4/8/16	1	Ctrl+R
Address	0	0	Ctrl+A
Data	0	0	Ctrl+D

¹Core includes digital register and SRAM.

²Analog means analog register.

2.6.1 Access Memory

2.6.1.1 Read Data From Memory Space

To read data from specific memory space (Flash/Core/Analog/OTP), user can follow the steps below:

Step 1: Select MCU type (e.g. "8258") and communication mode (e.g. "EVK") with target board in the "Memory Access" window.

Note:

If you select "USB" mode to communicate with target board, please ensure that the selected MCU supports USB function and its USB function is available.

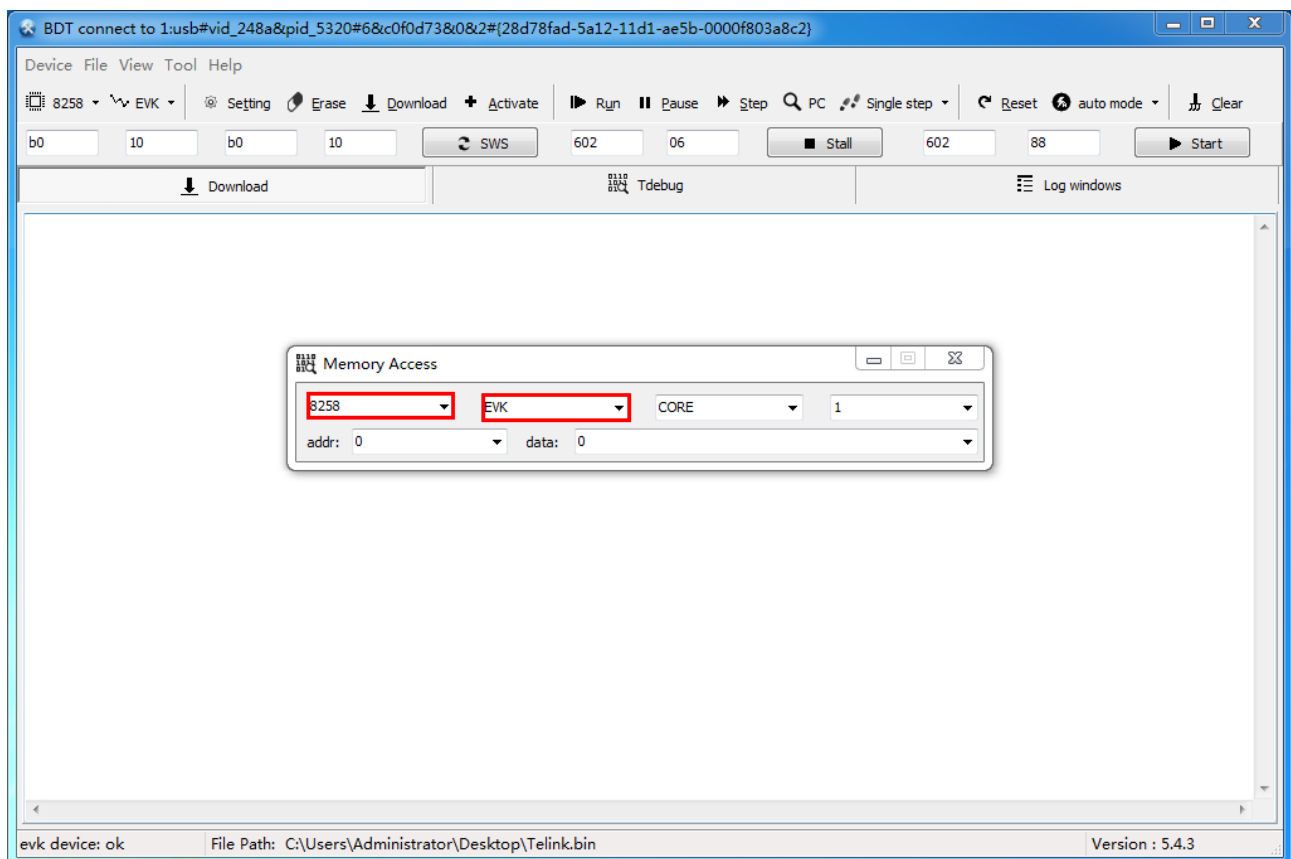


Figure 2.52: Select MCU type and communication mode

Step 2: In the "Memory Access" window select the operation target (e.g. "Core") and set the operation size (e.g. "16").

Note:

- If "OTP" is selected as the operation target, make sure that the selected MCU supports "OTP" function.
- To set the operation size, the user can select the size (if available) via the drop-down menu, or set the size directly in the edit box.

- The maximum operation size is 1MB.

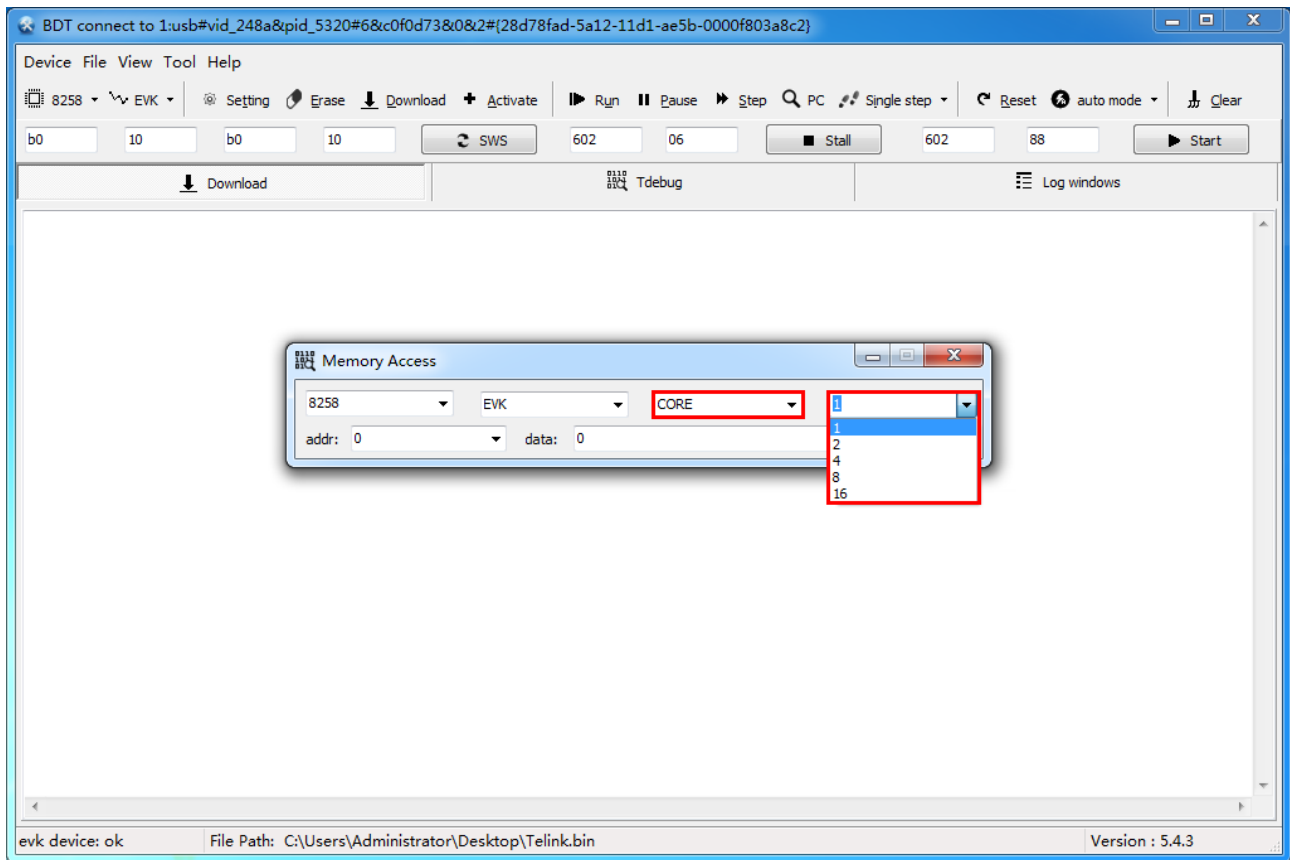


Figure 2.53: Set operation target and size

Step 3: Move mouse cursor to the “addr” box, and then input the offset of the starting address to be read (e.g. “0x040000”). Keep the cursor in the “addr” box, by clicking the “Tab” key, user can read data from the specified memory space.

Note:

- Every time when clicking the “Tab” key to initiate a memory reading operation, the value of the editable box will be saved, so that you can directly select it via the drop-down menu next time.

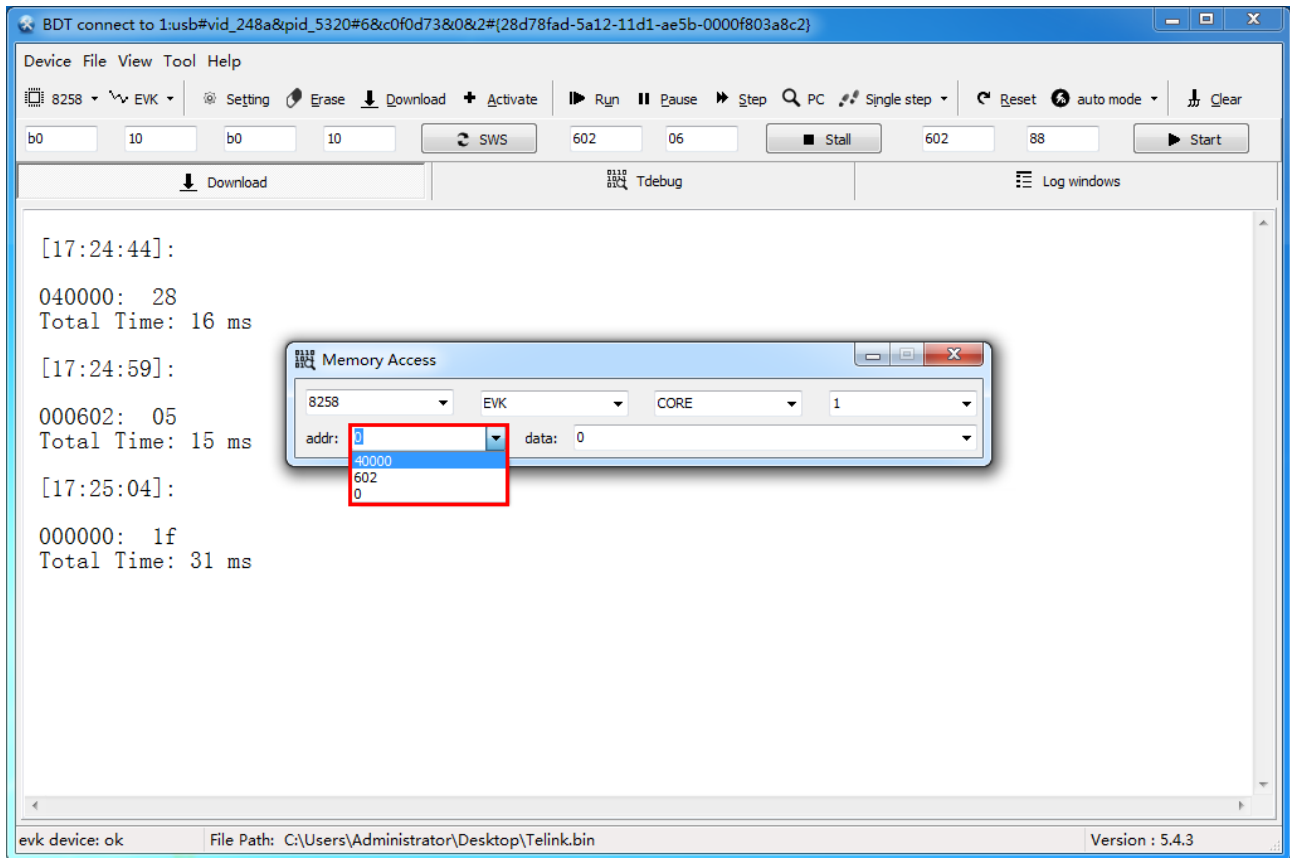


Figure 2.54: Set starting address and initiate reading

- When reading memory space (e.g. Core) larger than 1kB, the read data will be automatically saved to a file named "Read.bin" under the "user" folder. User can also use specific command line to change the name of file to save the read data. Refer to [section 2.8 Command line](#) for details.

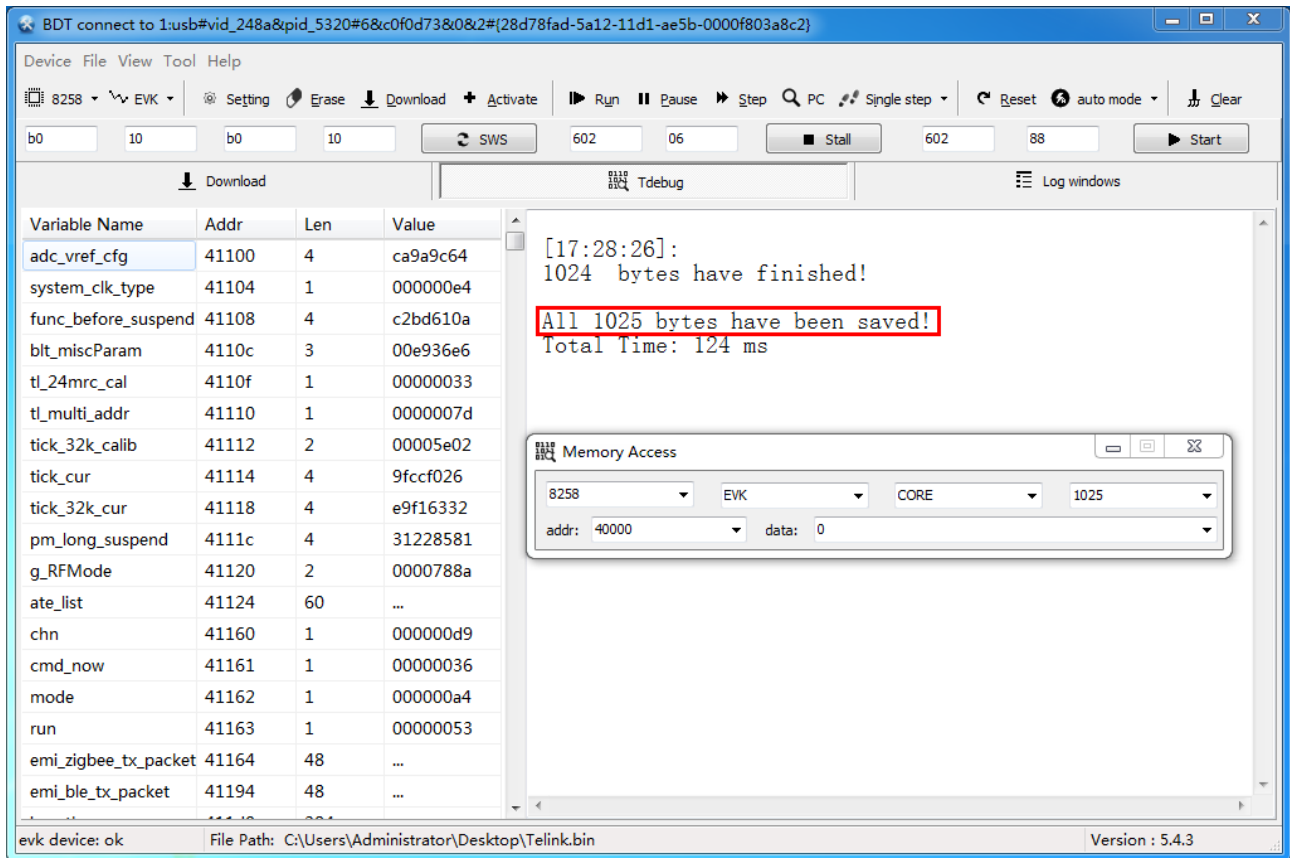


Figure 2.55: Save read data

2.6.1.2 Write Data into Memory Space

To write data into specific memory space, user can follow the steps below:

Step 1: Select MCU type (e.g. "8258") and communication mode (e.g. "EVK") with target board in the "Memory Access" window.

Note:

If you select "USB" mode to communicate with target board, please ensure that the selected MCU supports USB function and its USB function is available.

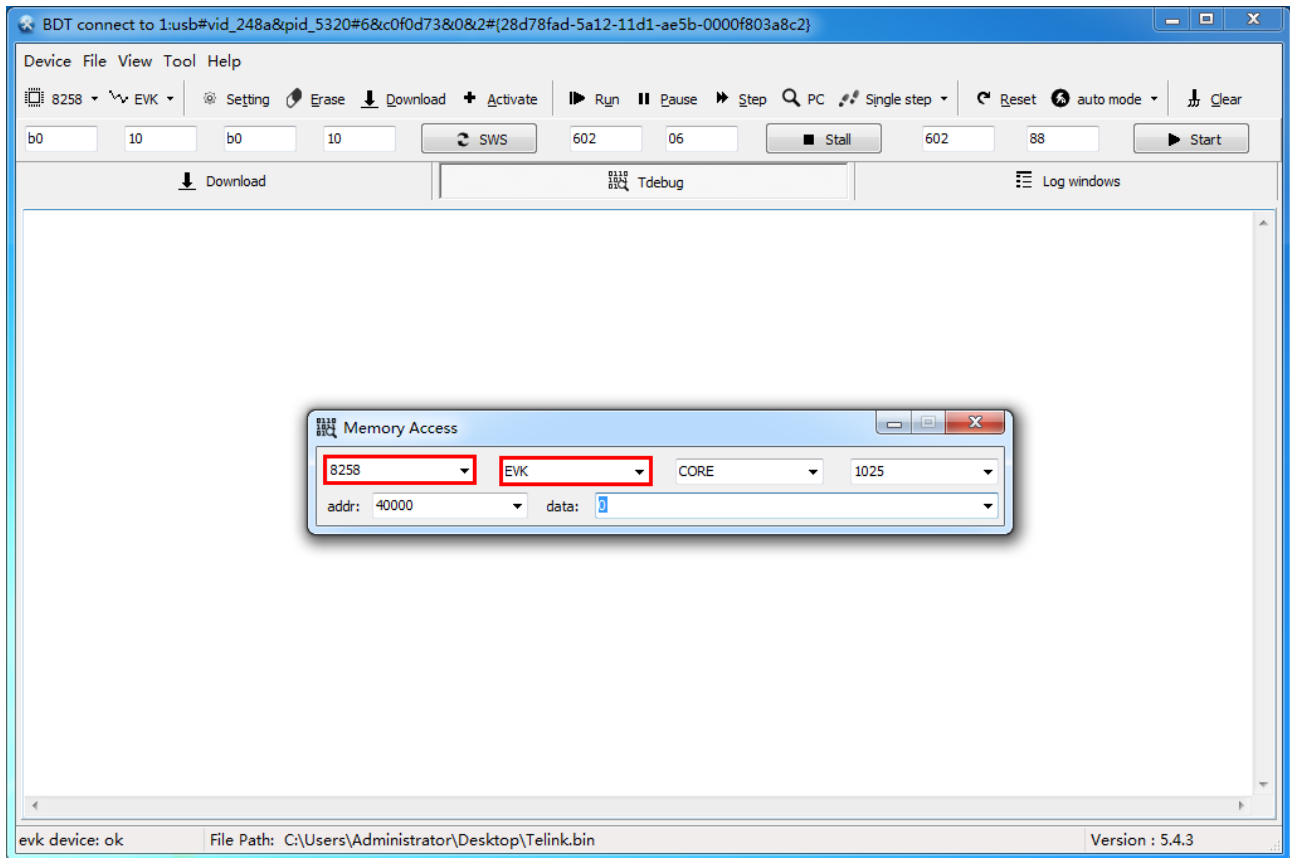


Figure 2.56: Memory access window

Step 2: In the “Memory Access” window select the operation target (e.g. “Core”) and set the operation size (e.g. “16”).

Note:

- If “OTP” is selected as the operation target, make sure that the selected MCU supports “OTP” function.
- To set the operation size, the user can select the size (if available) via the drop-down menu, or set the size directly in the edit box.
- The maximum operation size is 256 bytes.

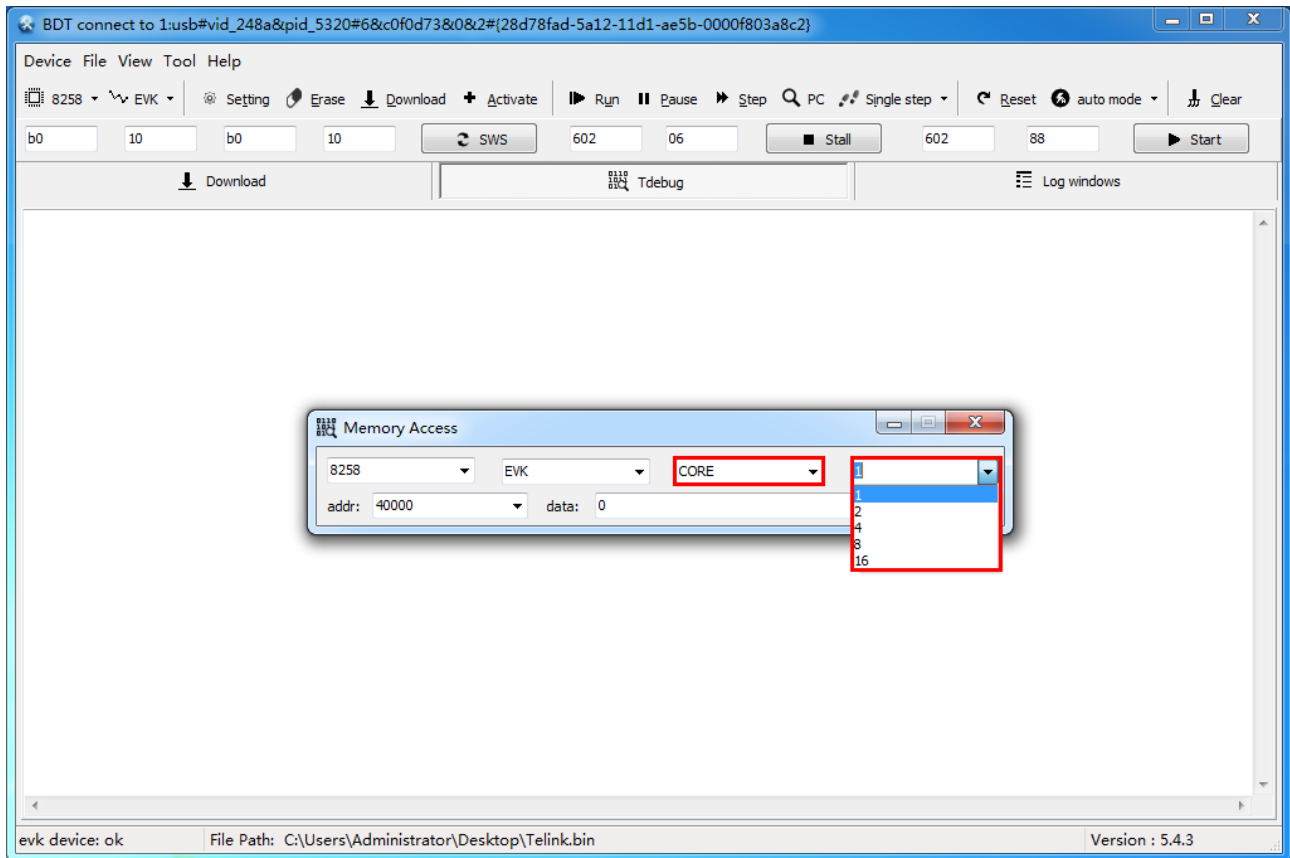


Figure 2.57: Set operation size

Step 3: Move mouse cursor to the “addr” box, and then input the offset of the starting address to be written (e.g. “0x040000”).

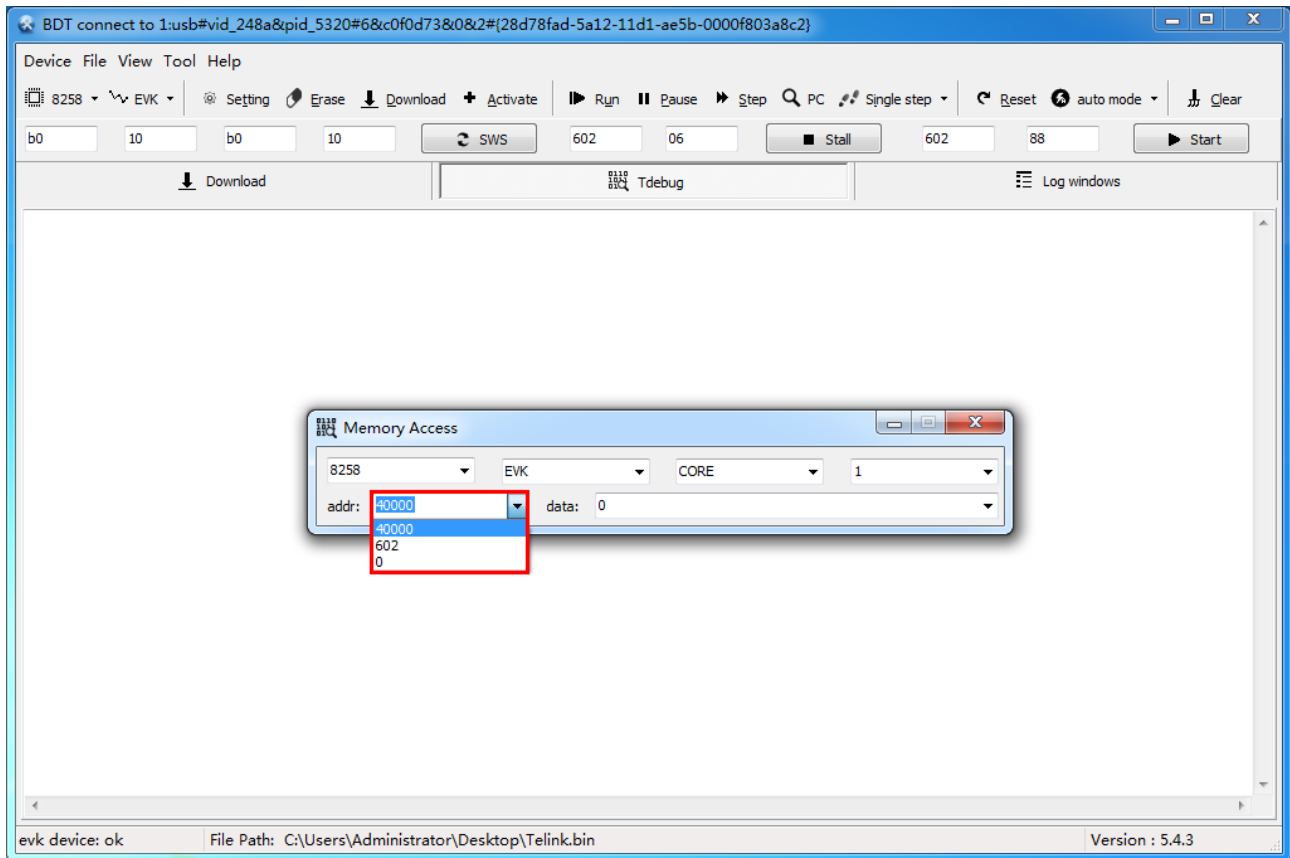


Figure 2.58: Offset of the starting address

Note:

When the operation target is selected as "Core", if the address you set is less than the starting address of "SRAM", the target memory space that is actually accessed is digital register; if the address you set is greater than the starting address of "SRAM", the target memory space that is actually accessed is SRAM.

Step 4: Write data to the specified address.

Move mouse cursor to the "data" box, and then input the target data to be written. Keep the cursor in the "data" box, by clicking the "Enter" key, user can write the specified data into the specified memory space.

Note:

When inputting the data, bytes must be separated from each other with space, as shown below:

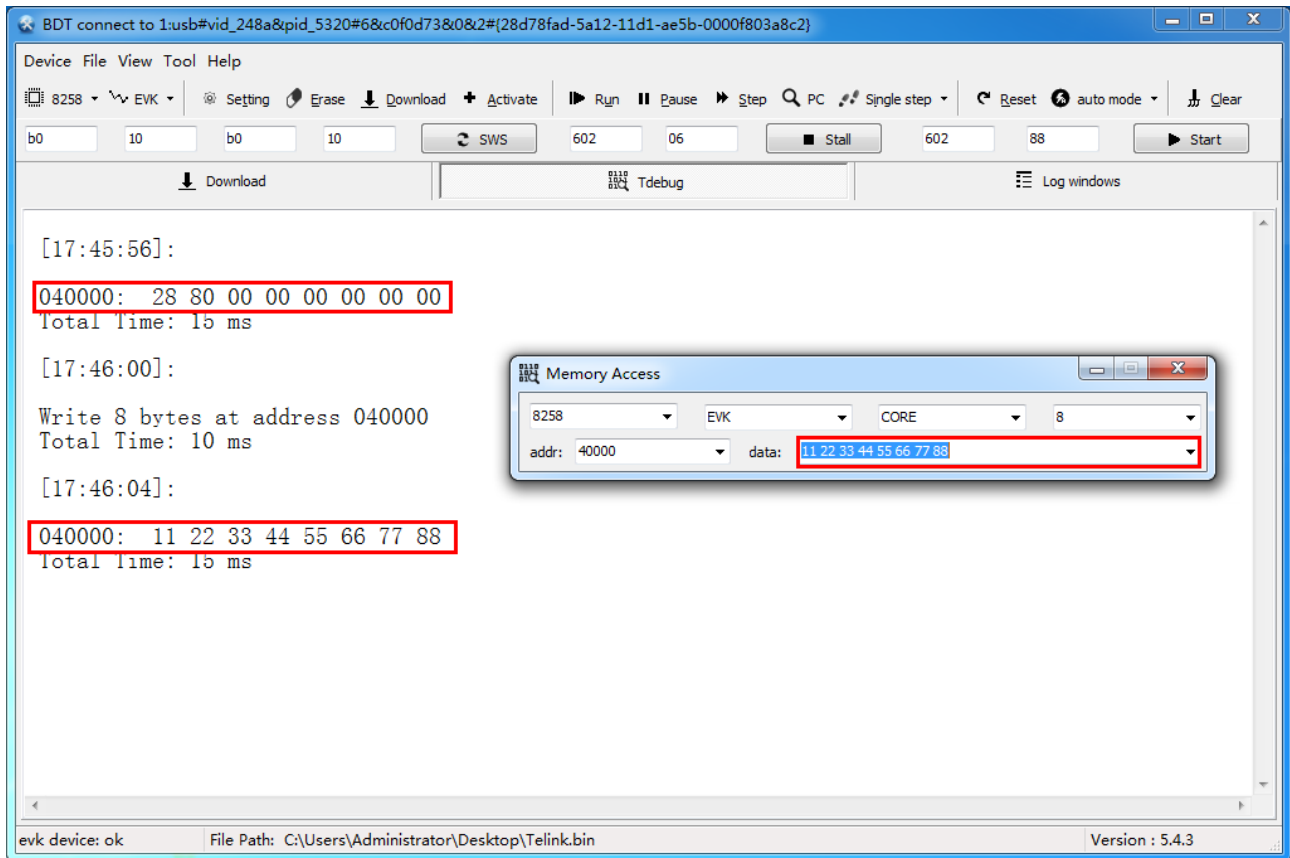


Figure 2.59: Set data and initiate writing

2.6.2 Operate Variable List

Before viewing all global variables, please ensure that the ".lst" file and the ".bin" file are placed under the same directory, the following two cases are supported:

Case 1: If only one pair of ".lst" file and the ".bin" file are placed under the same directory, this ".lst" file will be selected automatically.

 Interp_Cscript.lst	2018/9/11 10:03	LST 文件	317 KB
 Telink.bin	2018/9/11 10:03	BIN 文件	17 KB

Figure 2.60: Case 1

Case 2: If there are multiple ".lst" files and the ".bin" file are placed under the same directory, the ".lst" file with the name same as the selected ".bin" file will be selected automatically.

 Interp_Cscript.bin	2018/9/11 10:03	BIN 文件	17 KB
 Interp_Cscript.lst	2018/9/11 10:03	LST 文件	317 KB
 Telink.bin	2018/9/11 10:03	BIN 文件	17 KB
 Telink.lst	2018/9/11 10:03	LST 文件	317 KB

Figure 2.61: Case 2

For example, if you select the “Telink.bin” file, the “Telink.lst” file will be selected automatically. If there is no “Telink.lst” and only one “xxx.lst” file is placed under the directory of the selected “Telink.bin” file, this sole “xxx.lst” file will be selected.

2.6.2.1 Update Variable List

To look all global variables, user can select the “Tdebug” page, move mouse cursor to variable list, and then right click the mouse to choose “Refresh” or press the shortcut key “F3”. Each time you press “F3”, the variable list can be updated once.

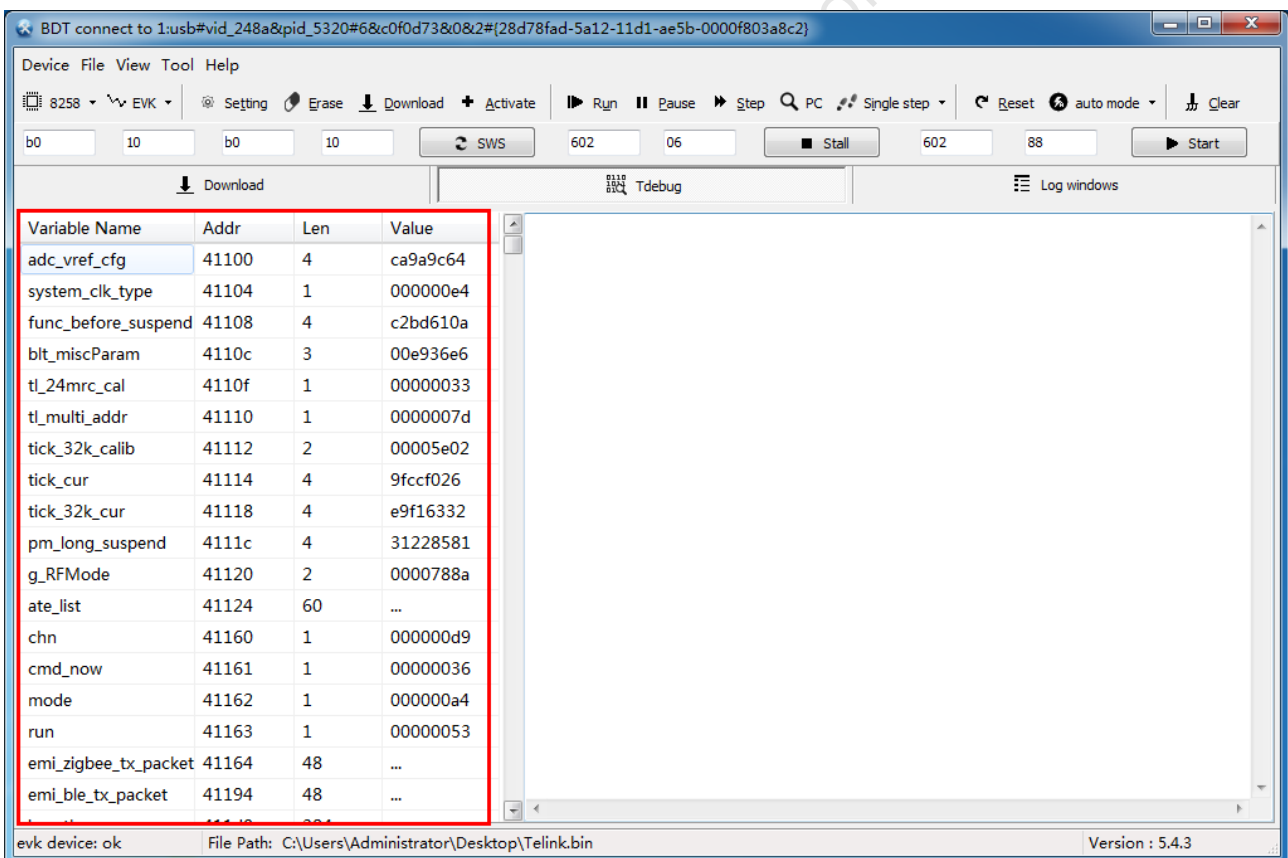


Figure 2.62: Update variable list

2.6.2.2 Sort Variables

As needed, user can also right click mouse, and choose the "Sort by address" to sort all variables by address from low to high (default), or choose the "Sort by name" to sort all variables by ASCII value of variable name from A to Z (compare the first different character).

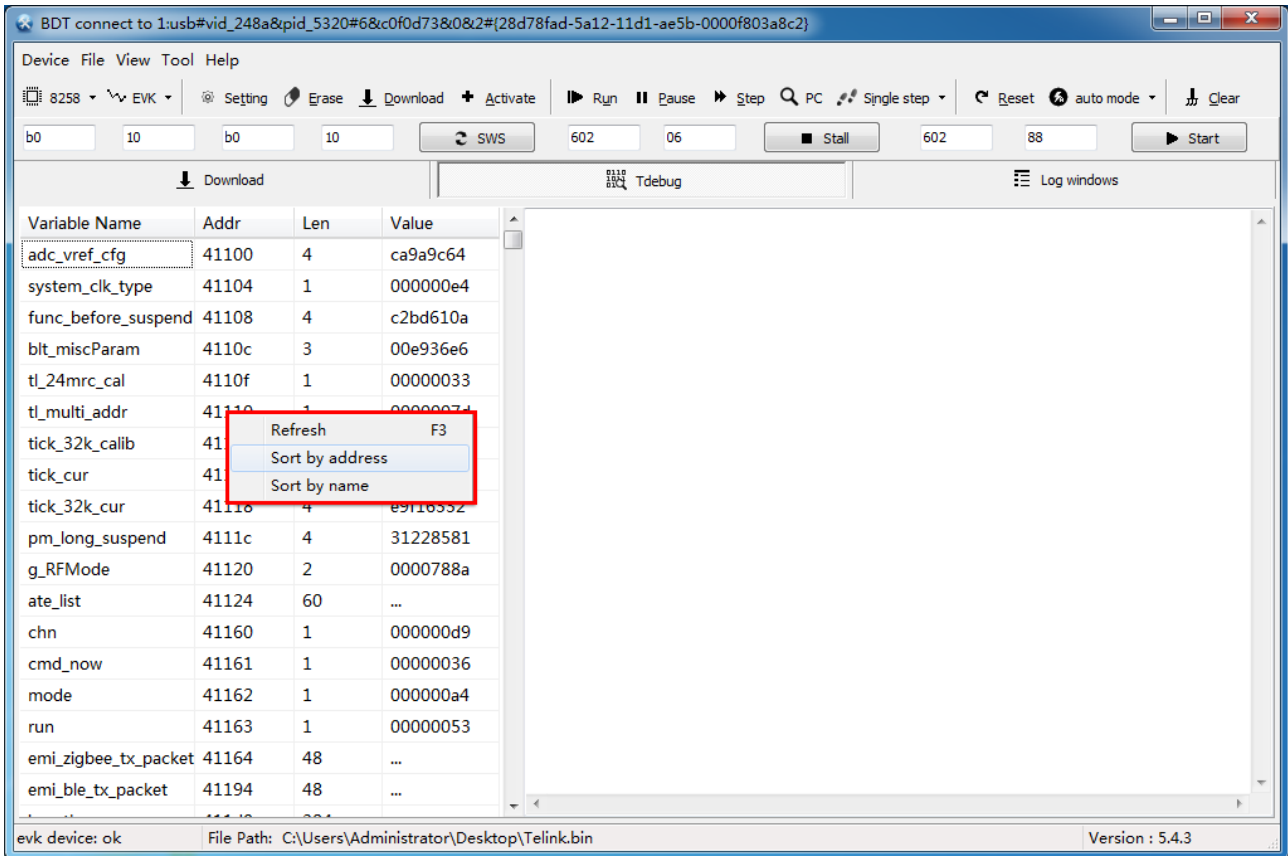


Figure 2.63: Sort variable by name

2.6.2.3 View Aariable Above 4 Bytes

If a variable's length is larger than 4 bytes, by clicking the "..." in the "Value" column of the "variable list", its value will be available in the log window on the right.

If a variable's length is greater than 1024 bytes, all values of the variable will be saved in a file named "Read.bin".

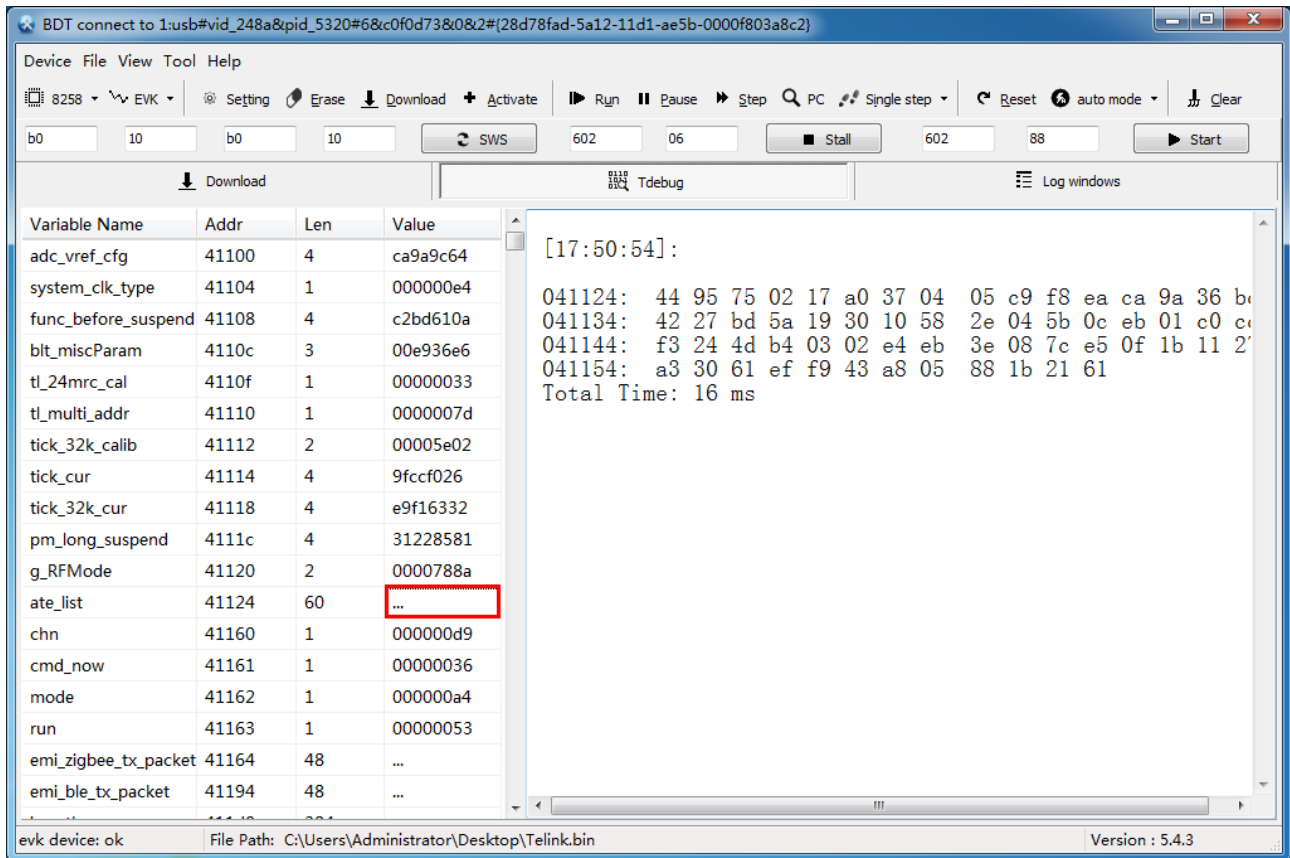


Figure 2.64: Read variable value more than 4 bytes

2.6.2.4 Modify Variable Value

Variables whose length does not exceed 4 bytes support direct write operations, double clicking the corresponding box in the “Value” column to directly modify its value.

For example, to modify the value of the variable “adc_vref_cfg” in the address “0x41100” to “0x00000004”, user can follow the steps below:

Step 1: Double click the box in the “Value” column corresponding to the specified variable (e.g. “adc_vref_cfg”) to make the box editable.

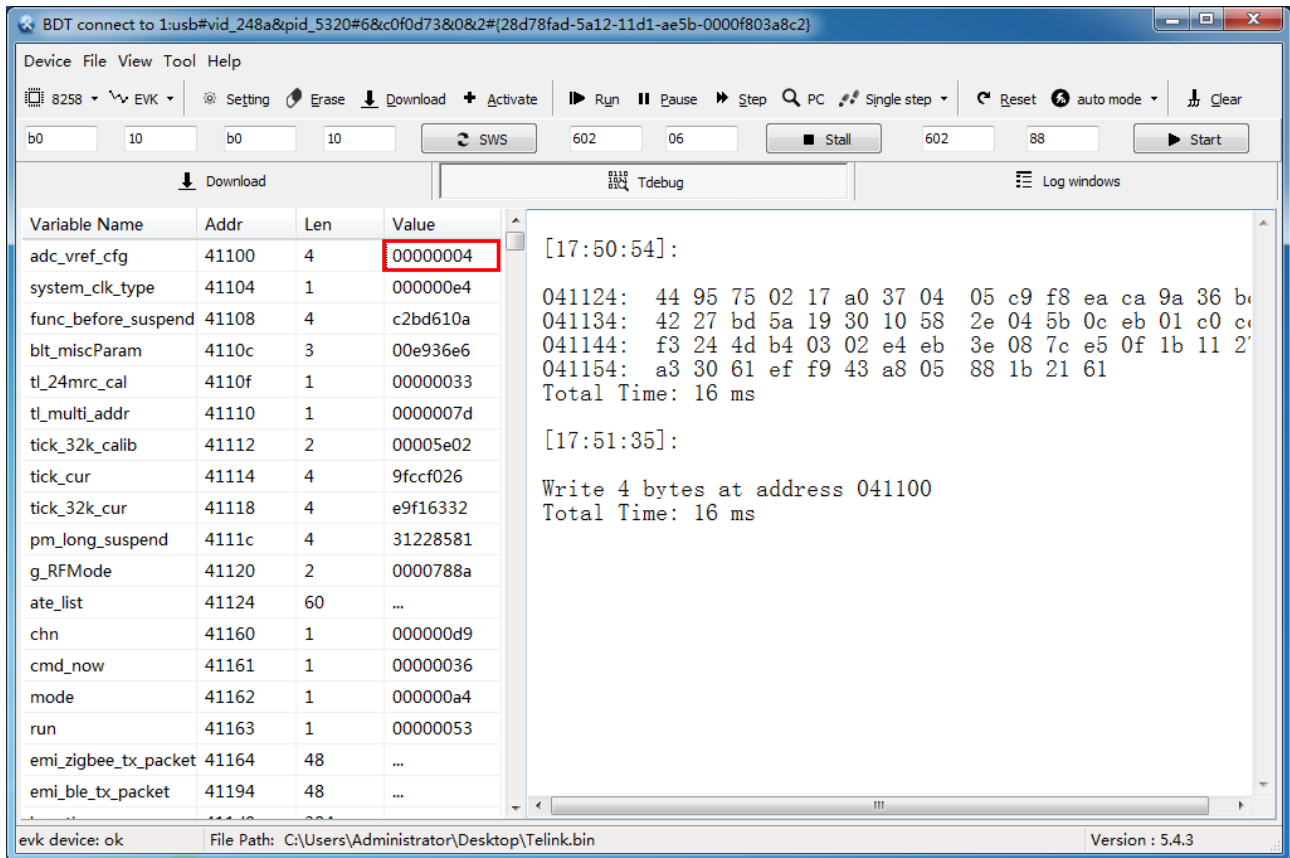


Figure 2.65: Double click specified "Value" box

Step 2: Input new variable value (e.g. "0x00000025") to replace the old value ("0x00000004").

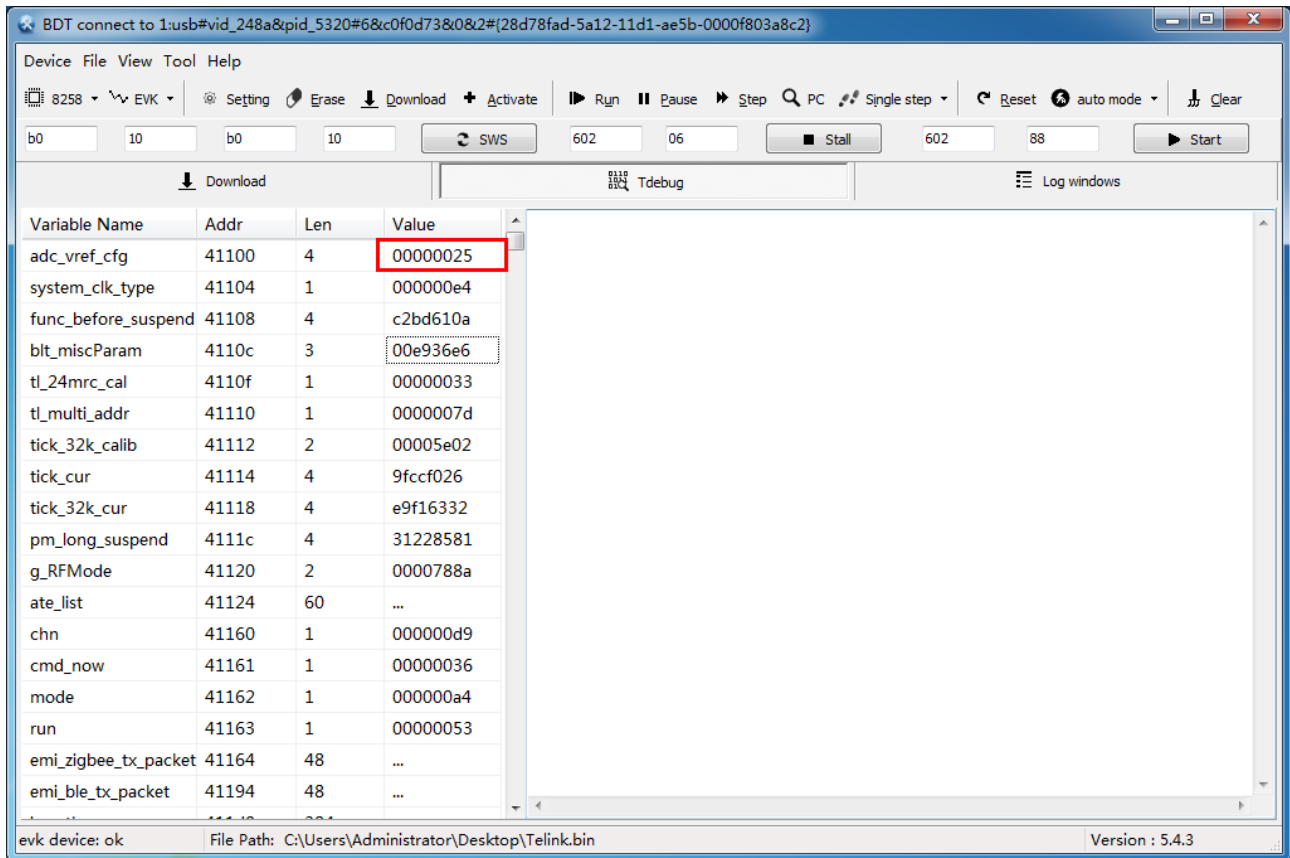


Figure 2.66: Modify variable value

Step 3: Press the “Enter” key to write the new value into the specified variable’s address.

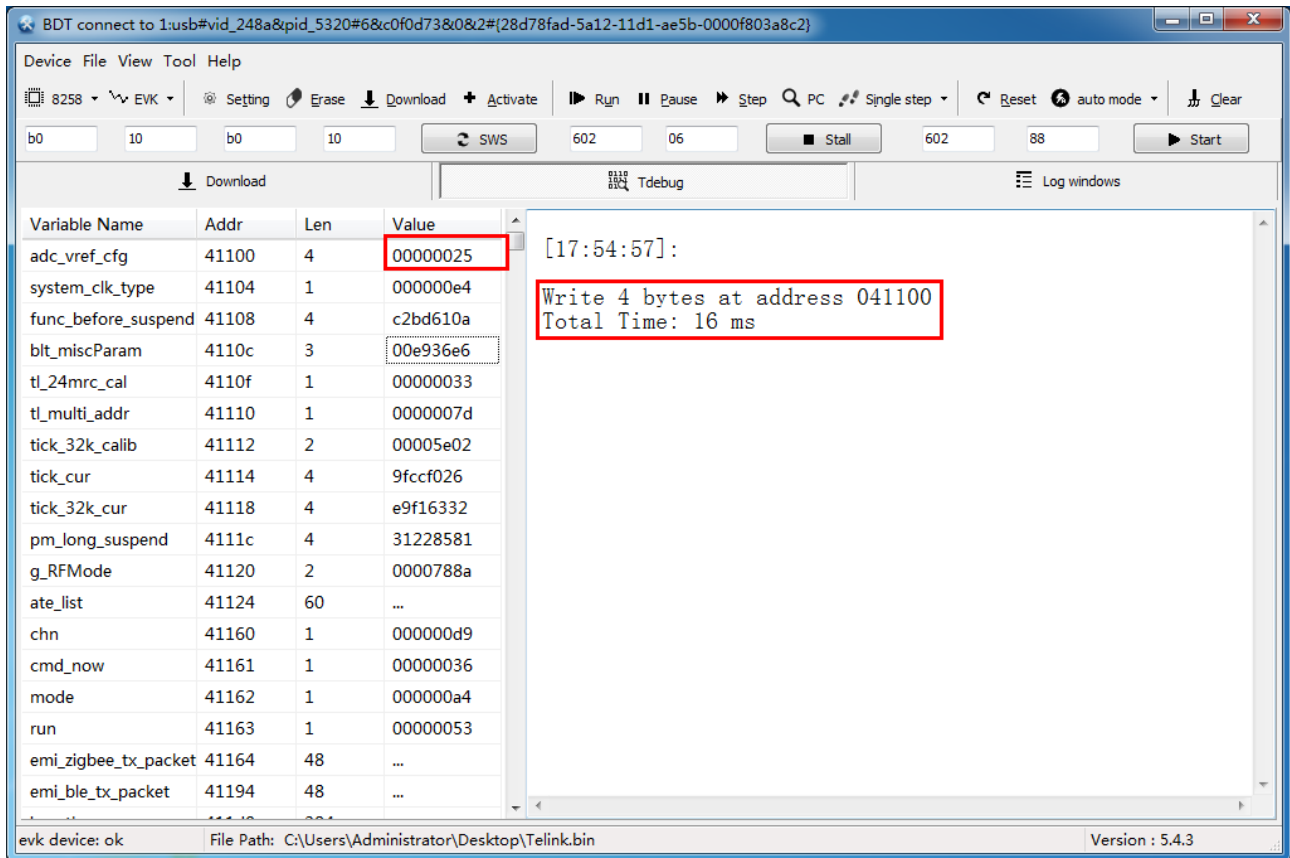


Figure 2.67: Write data to specified address

2.6.3 Debug MCU

Before using the buttons including “Run”, “Pause”, “Step” and “PC” to debug MCU, please ensure that the selected MCU supports this function. Refer to [section 5 Function Support List](#) for details.

In the sub-sections below, the MCU “8258” is taken as an example to introduce this function.

2.6.3.1 Run MCU

After clicking the “Pause” or “Step” button, user can make MCU continue to run from the current position by clicking the “Run” button. Please click the “PC” button to check the MCU status, read the PC and make sure the program runs again.

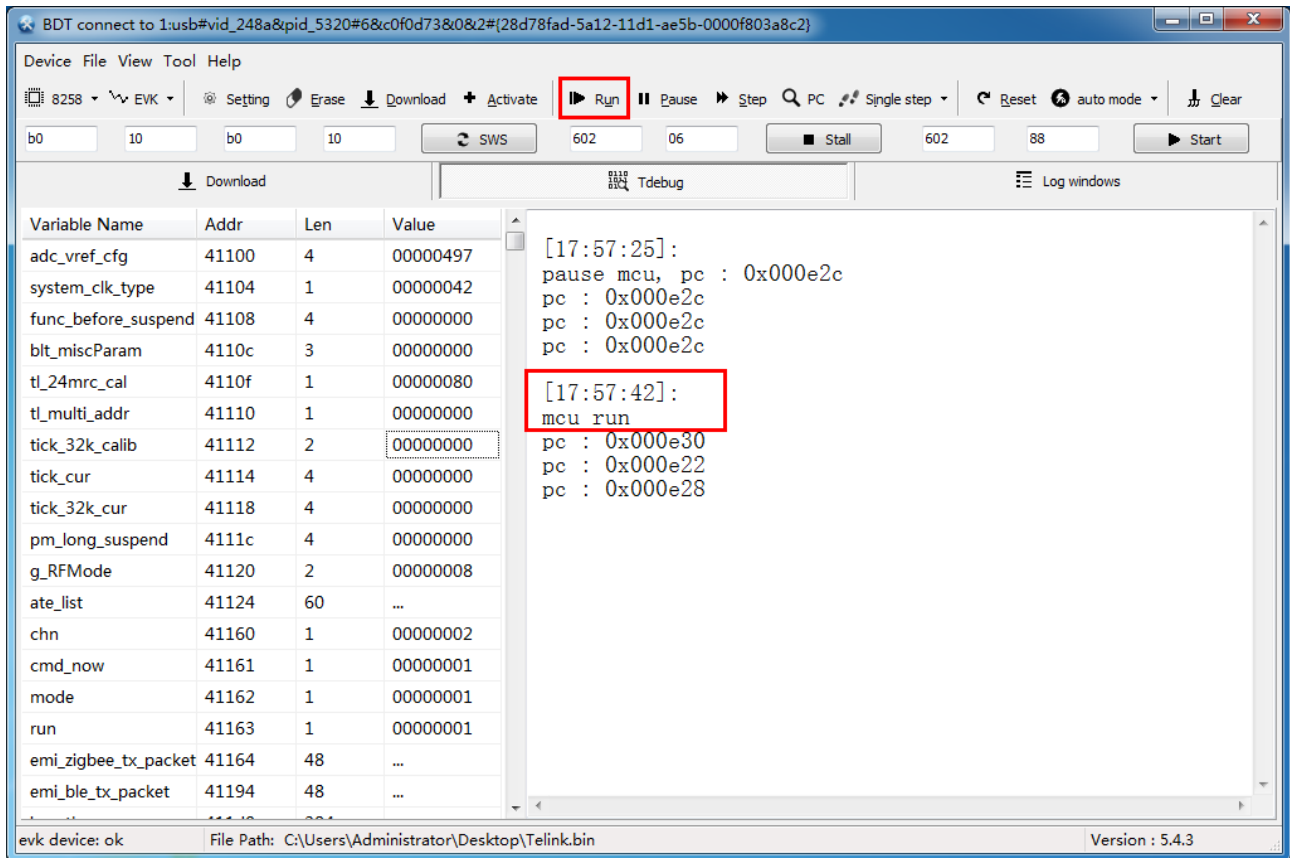


Figure 2.68: Run MCU

2.6.3.2 Pause MCU

To view the details of the MCU status, user can click the "Pause" button to pause MCU. Click the "Run" button, MCU will continue to run from the current position.

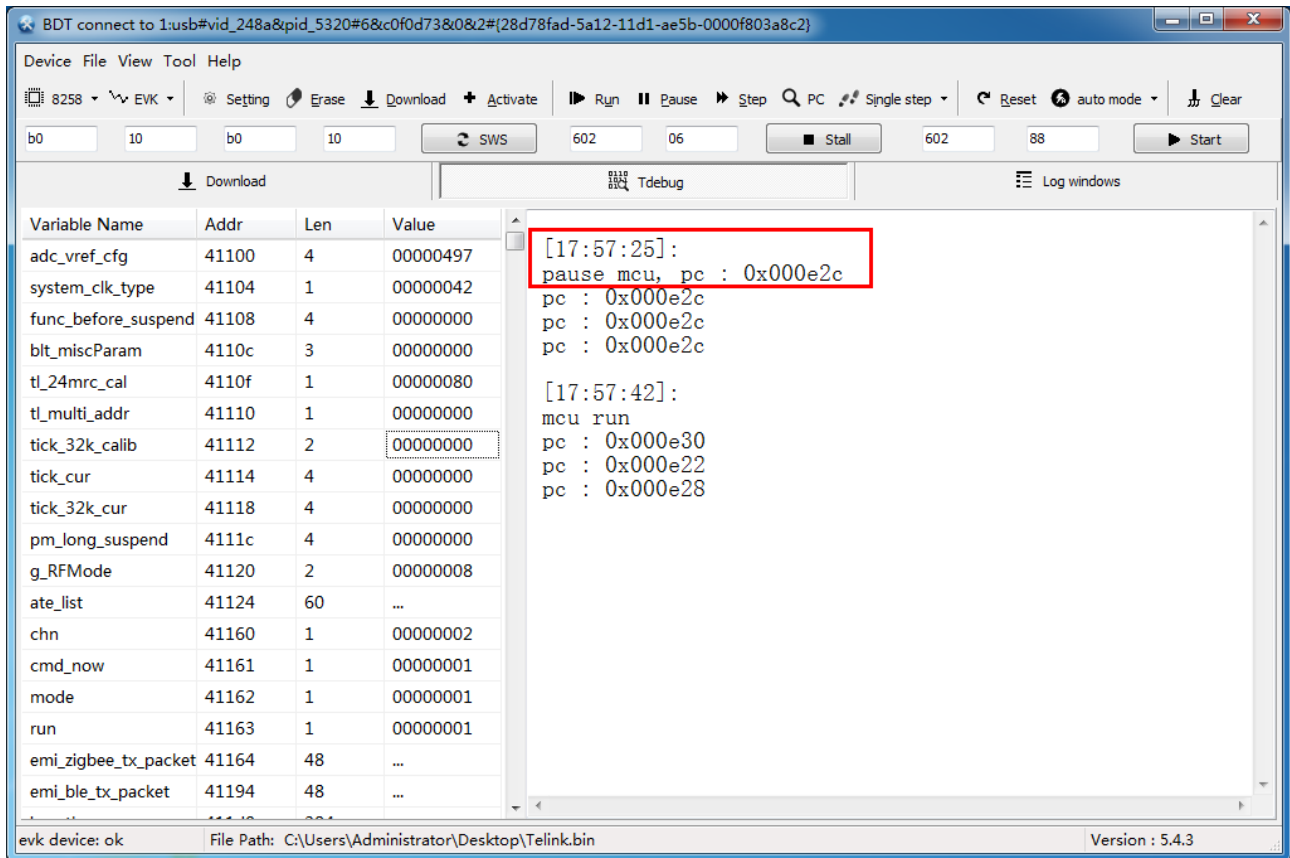


Figure 2.69: Pause MCU

2.6.3.3 Step MCU

If the communication with target board is available, user can click the "Step" button to view the current position of instruction step by step.

Both "Trace MCU" and "Step MCU" support two modes, as shown below.

- (1) If the "Single step" mode is selected, MCU will enter single step mode, in which one PC tracing will be implemented after clicking the "PC" or "Step" button.

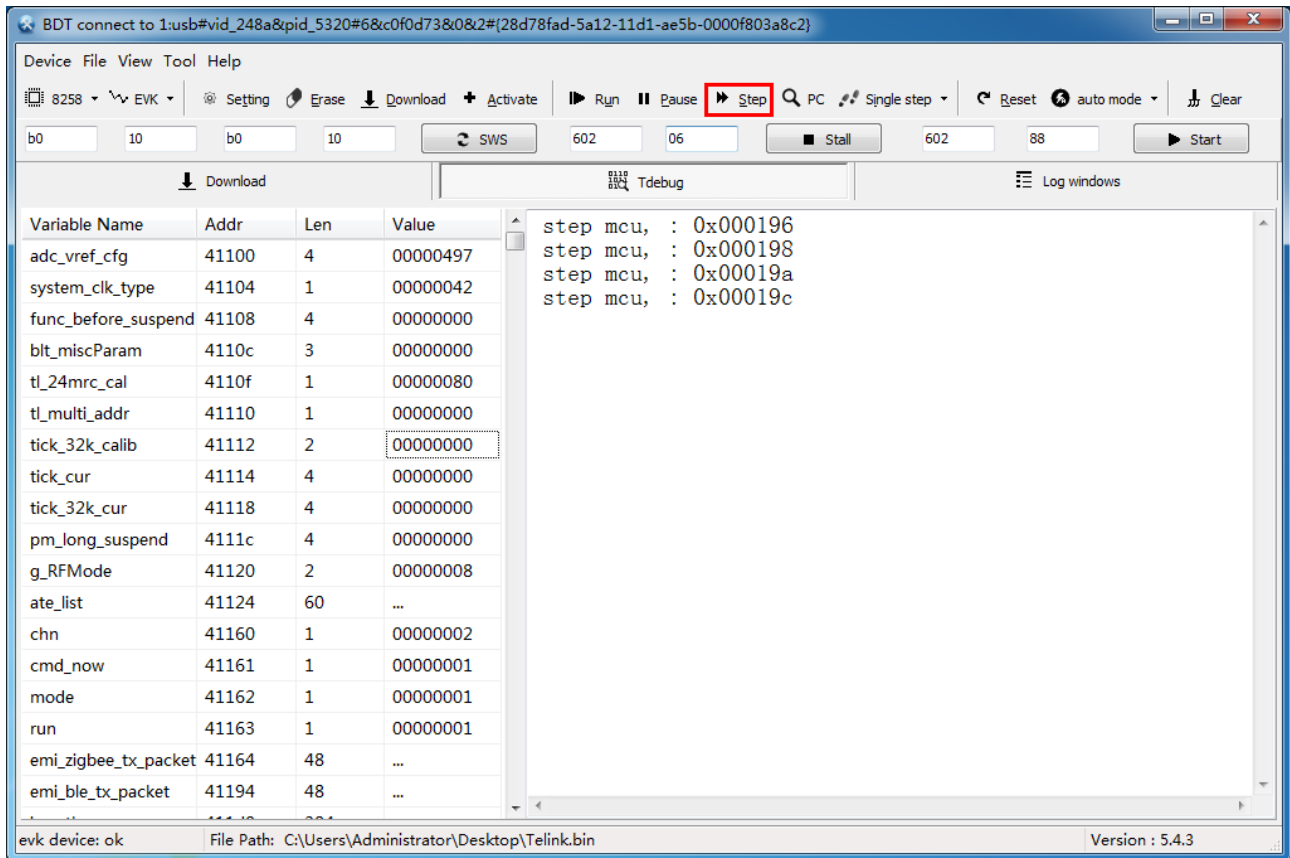


Figure 2.70: Step MCU in “Single step” mode

- (2) If the “continuous” mode is selected, MCU will enter continuous mode and implement continuous PC tracing or make MCU step continuously.

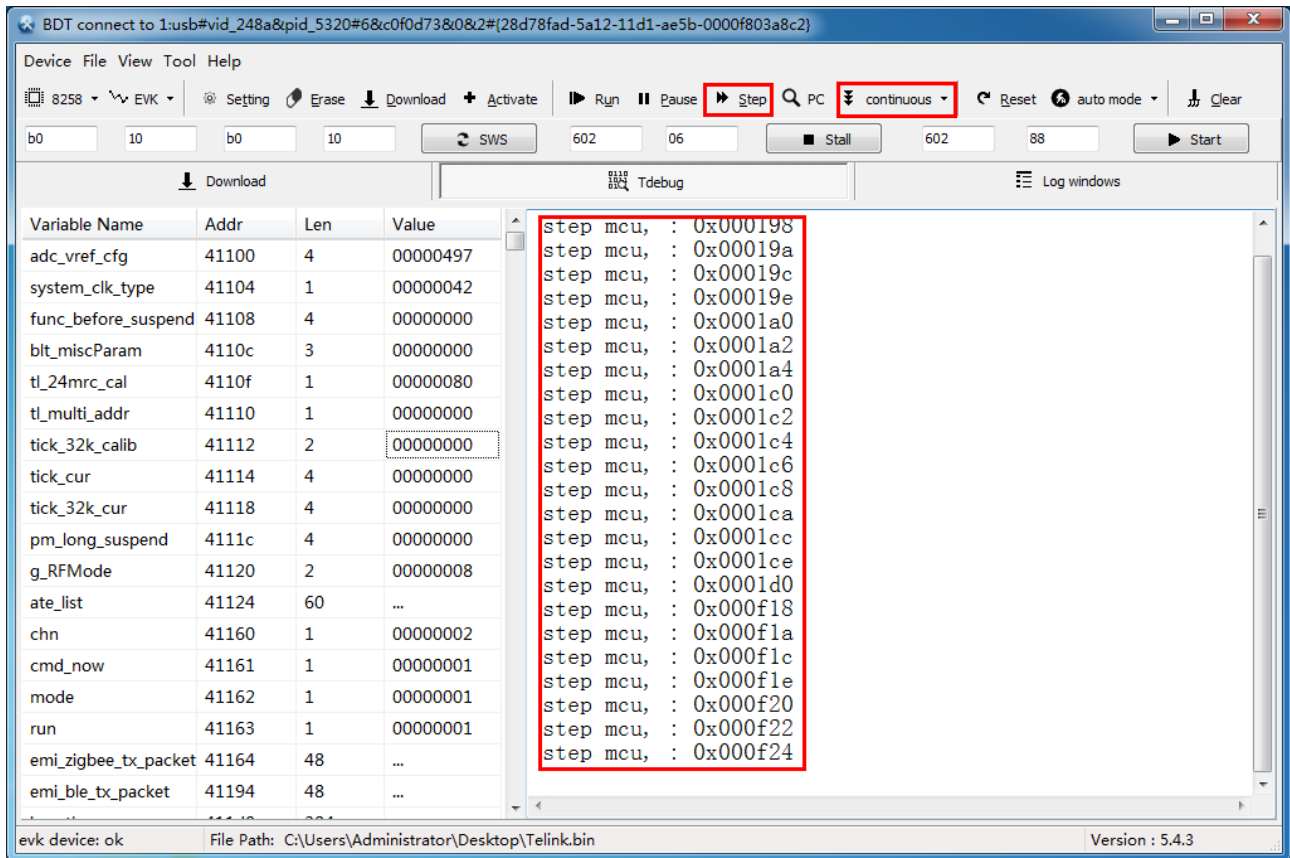


Figure 2.71: Step MCU in "continuous" mode

To cancel "continuous" mode, user can either switch mode to "Single step" mode, or click the "Run"/"Pause" button.

If the target firmware (bin file) and "xxx.lst" file are under the same directory, user can view more details by selecting "View -> interp .lst" and clicking the "PC" button to analyze the "xxx.lst" file, as shown below. Refer to [2.6.2 Operate Variable List](#).

	Telink.bin	2018/3/1 17:55	BIN 文件	3 KB
	Telink.lst	2018/3/1 17:55	LST 文件	44 KB

Figure 2.72: "xxx.lst" file

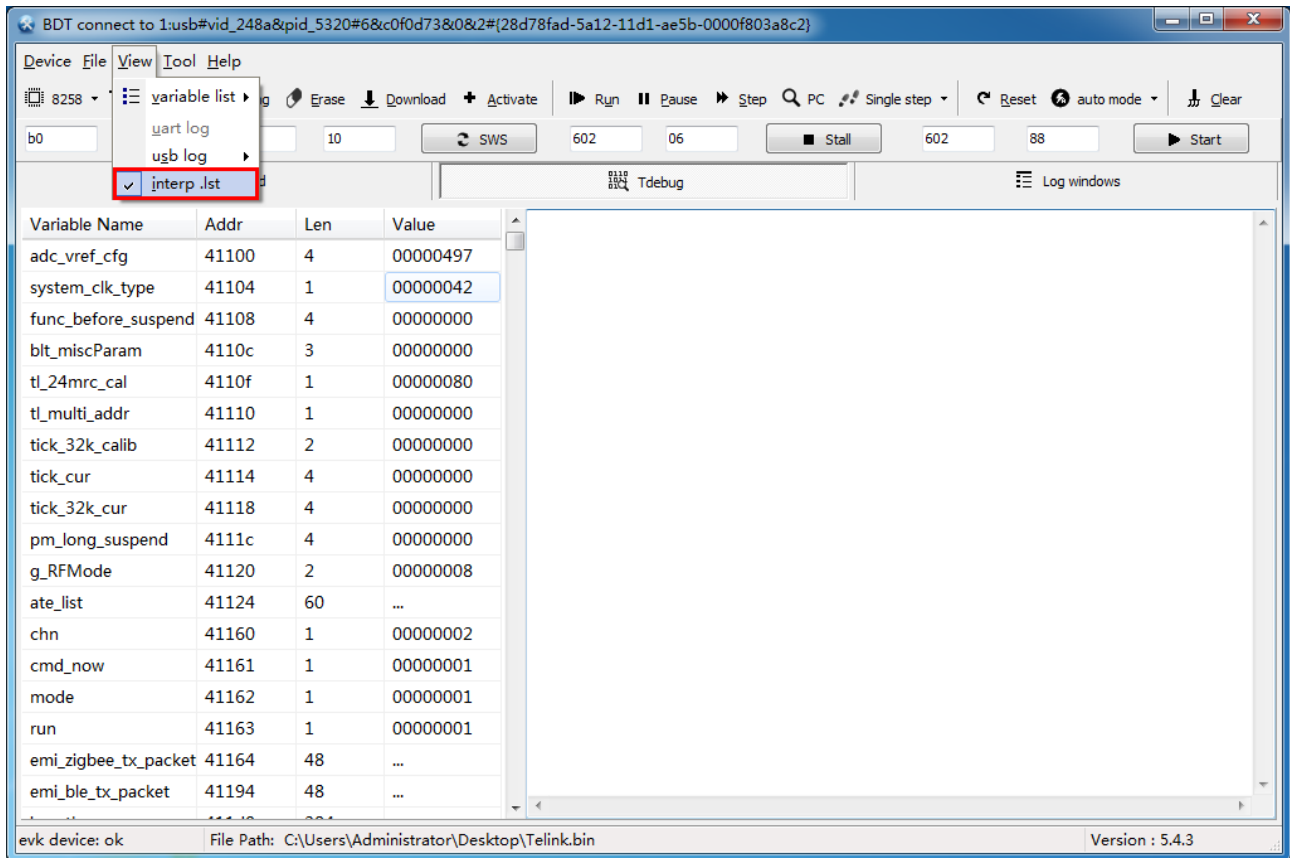


Figure 2.73: Select "interp.lst"

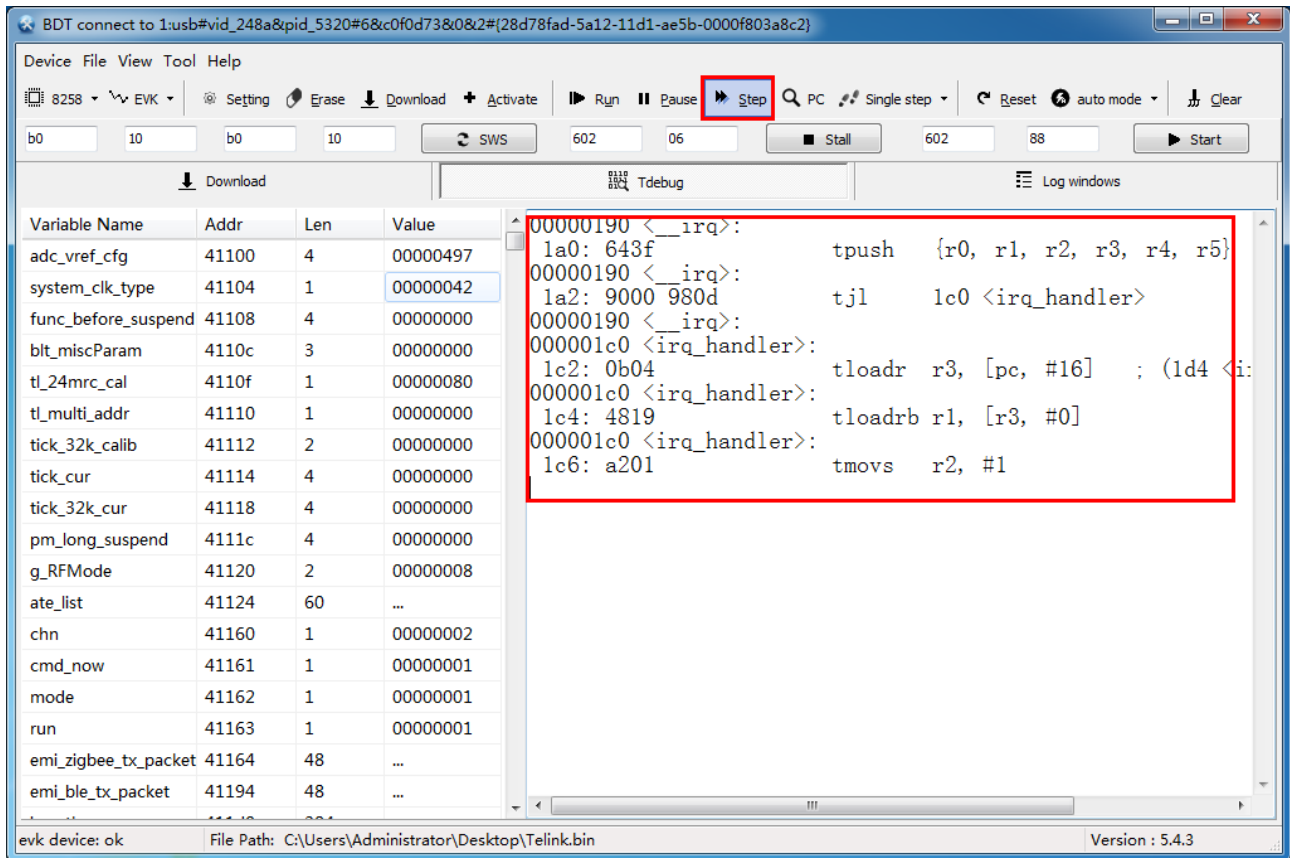


Figure 2.74: Step MCU with "interp.lst" selected in "Single step" mode

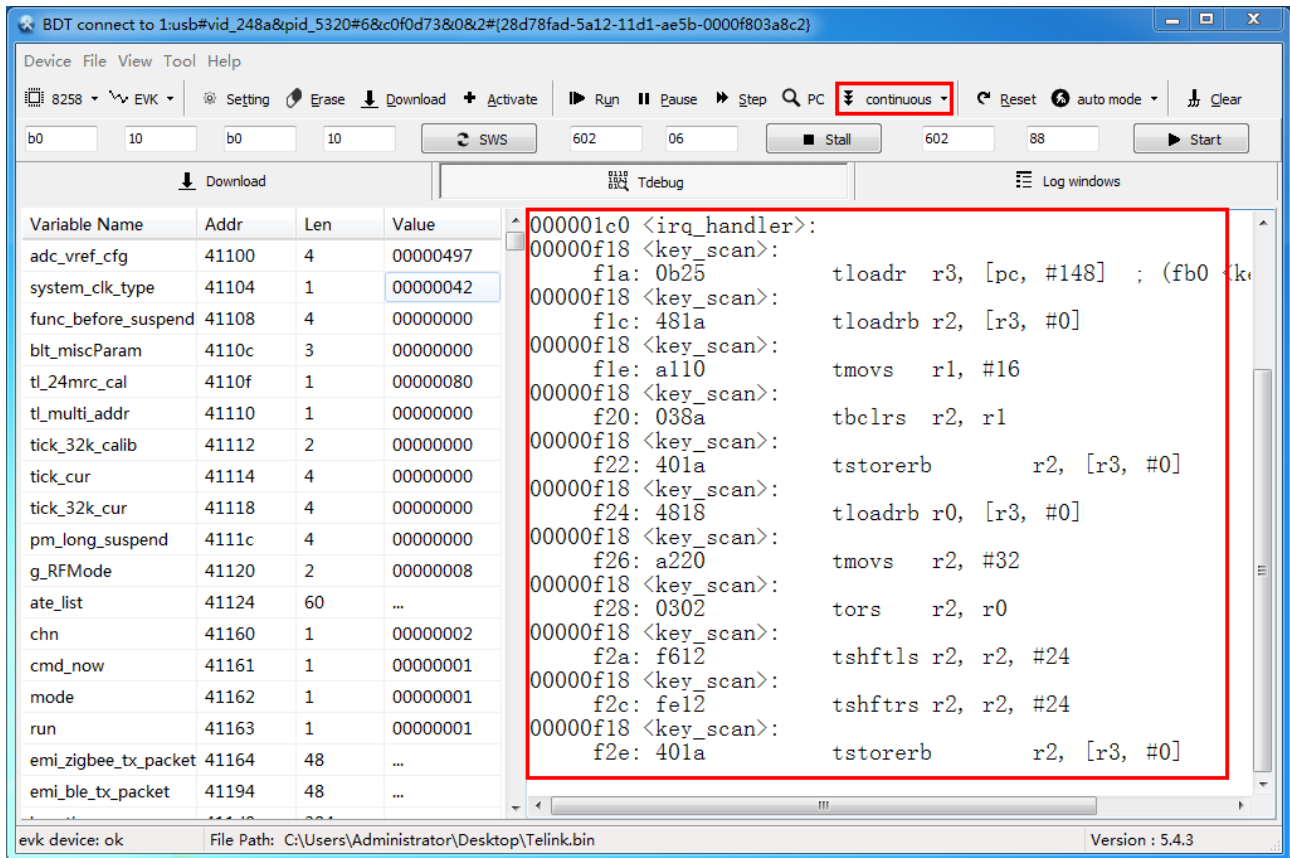


Figure 2.75: Step MCU with "interp.lst" selected in "continuous" mode

2.6.3.4 Trace PC

If the communication with target board is available, user can click the "PC" button to view the current position of instruction, as shown below:

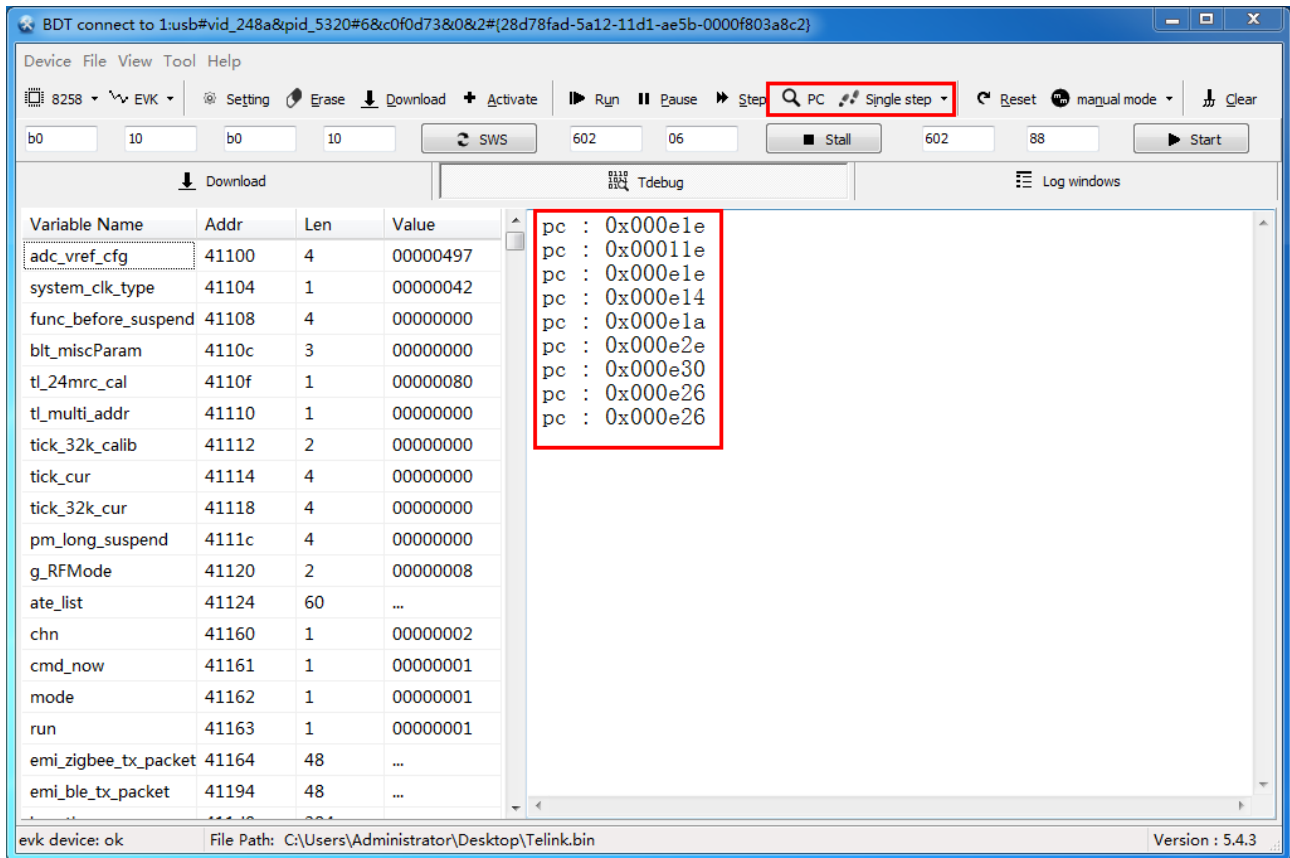


Figure 2.76: Trace PC in "Single step" mode

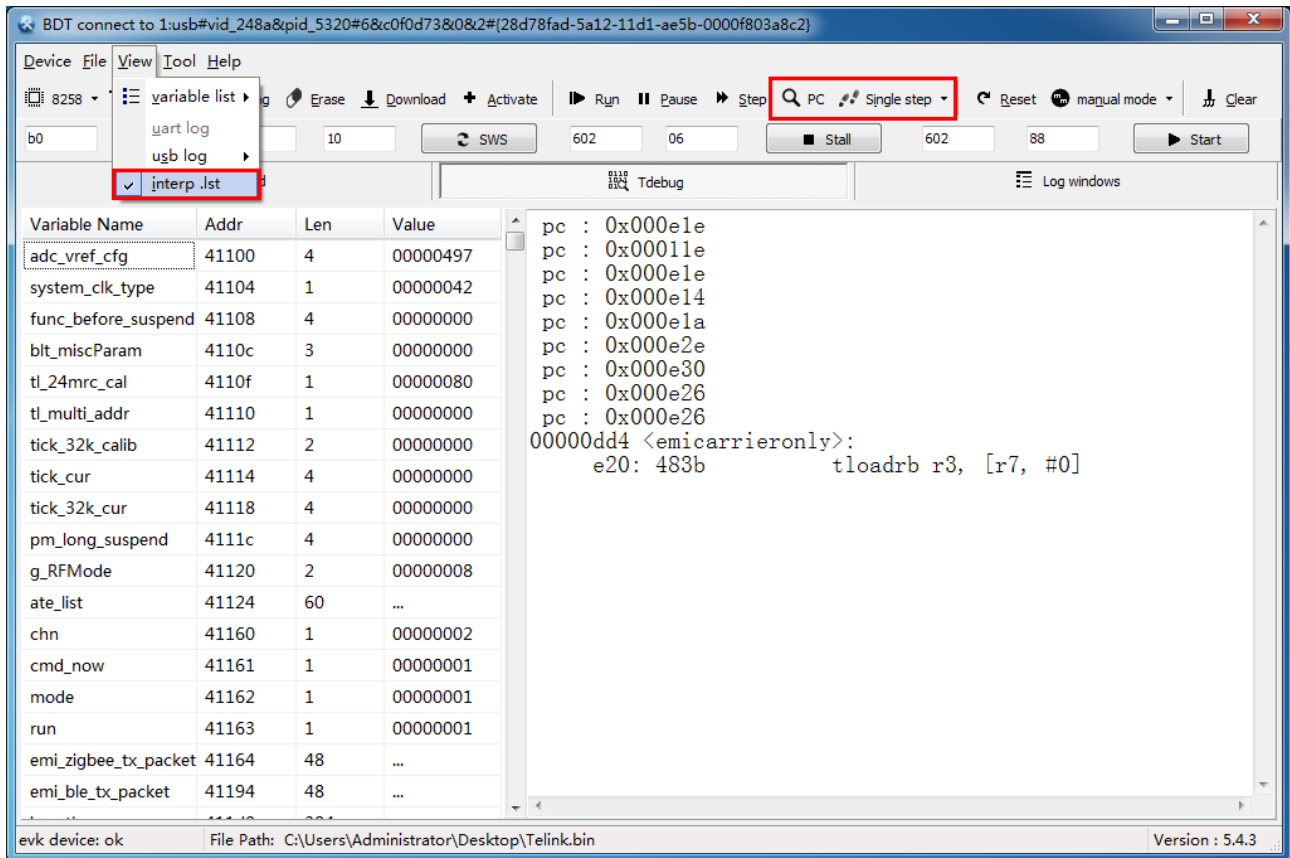


Figure 2.77: Trace PC with "interp.lst" selected in "Single step" mode

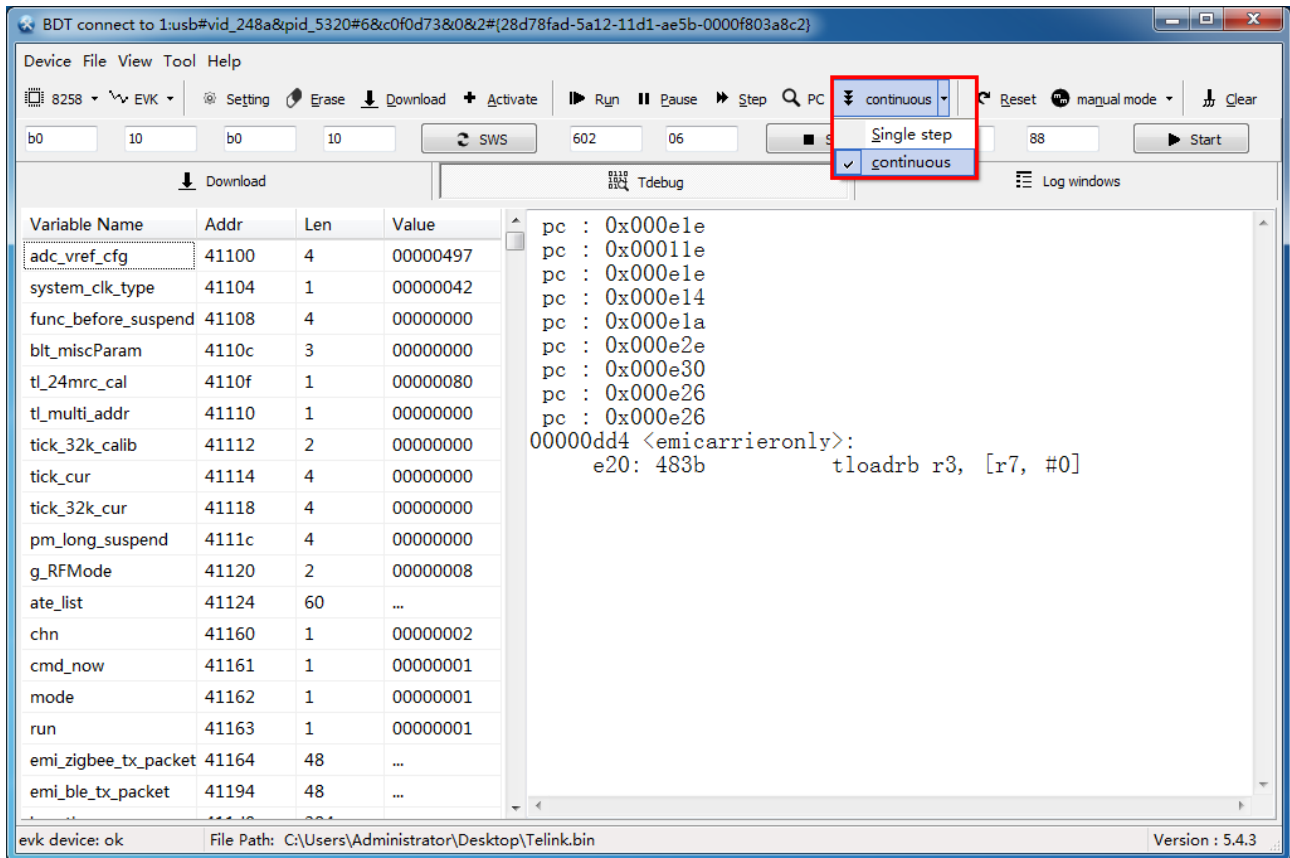


Figure 2.78: Switch "Single step" mode to "continuous" mode

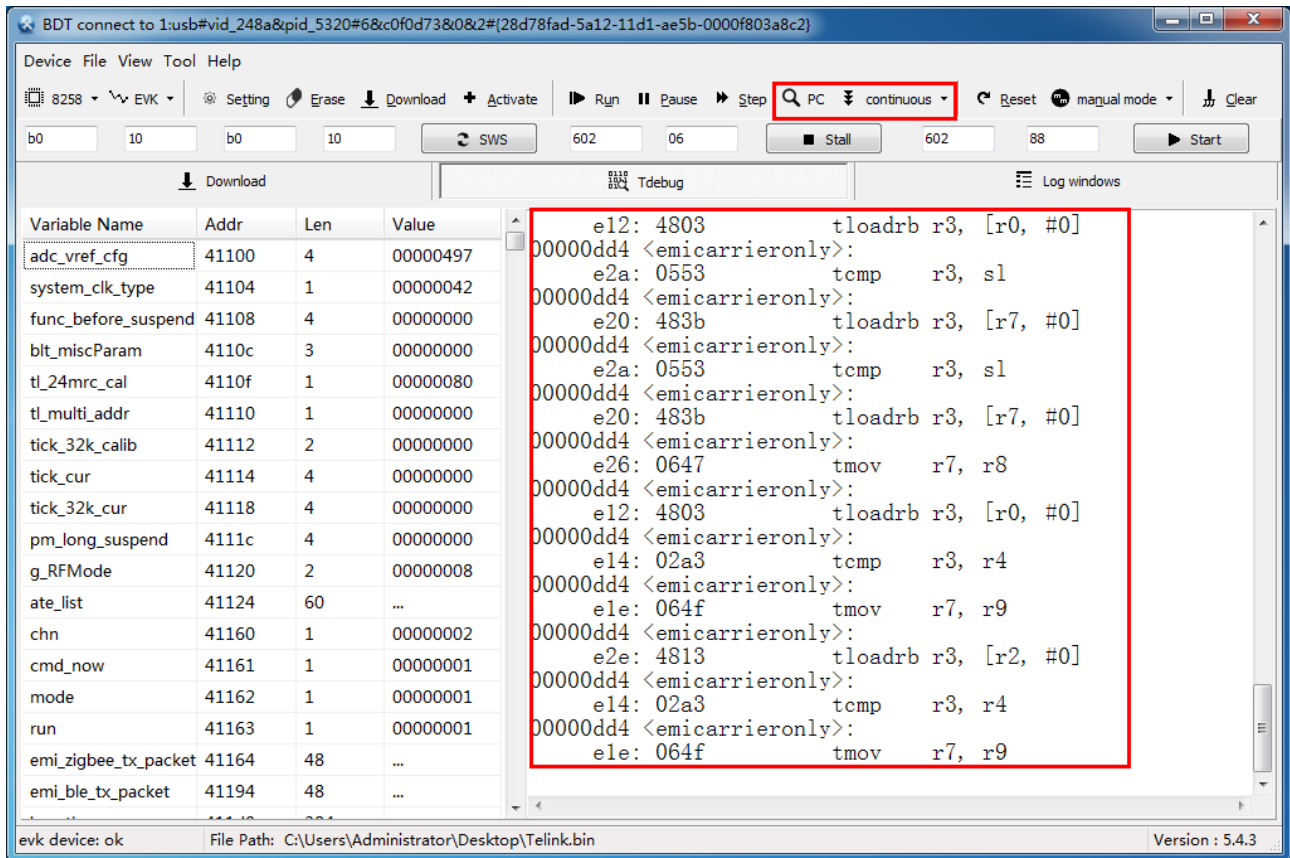


Figure 2.79: Trace PC with "interp.lst" selected in "continuous" mode

2.6.3.5 Stall MCU

When there is not enough time to view the status of MCU, user can click the "stall" button to stall MCU and view the status of MCU or change the configuration of MCU.

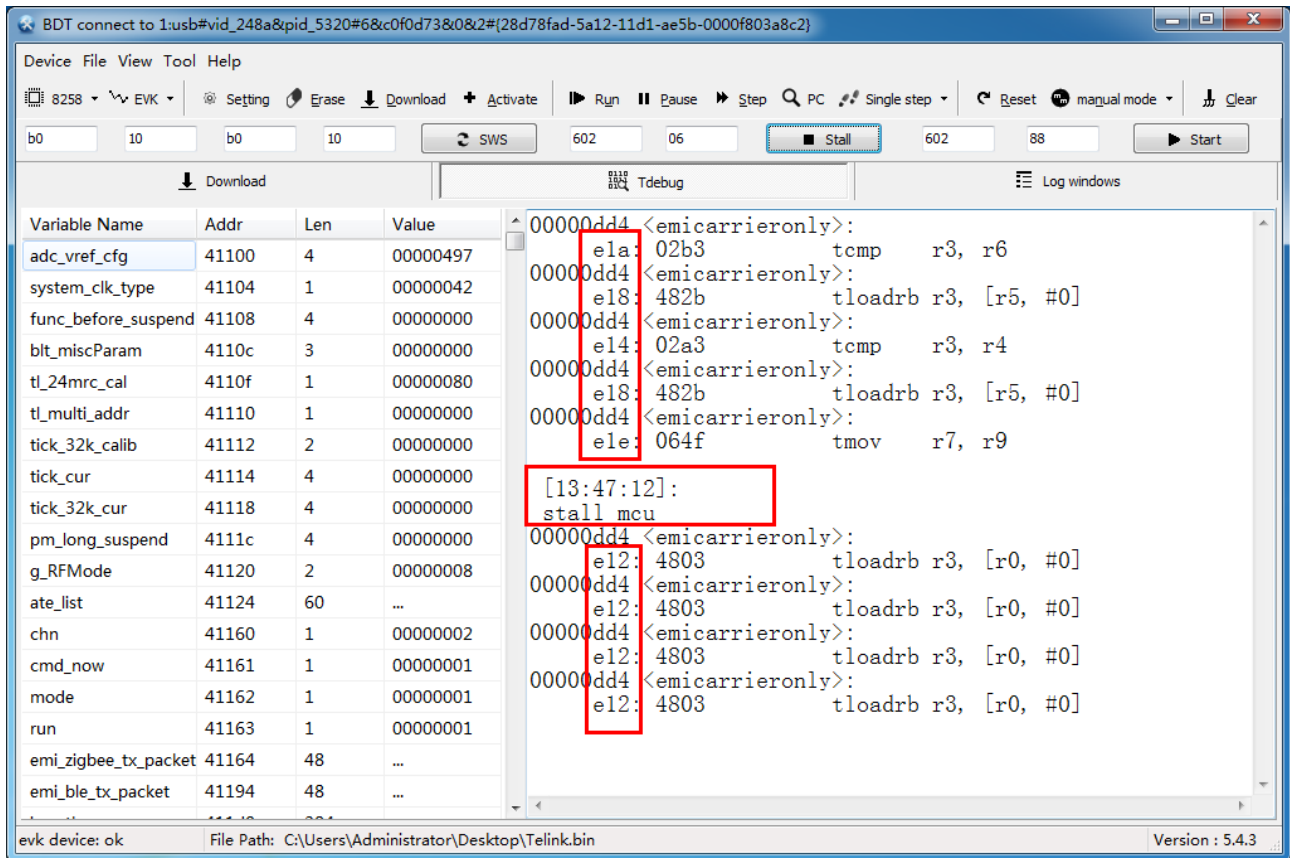


Figure 2.80: Stall MCU

Note:

In EVK mode, only the EVK burned with the "Firmware_v3.2.bin" (or higher version) supports this function.

2.6.3.6 Start MCU

After stalling MCU, click the "start" button to start MCU, so that the MCU will run from the SRAM.

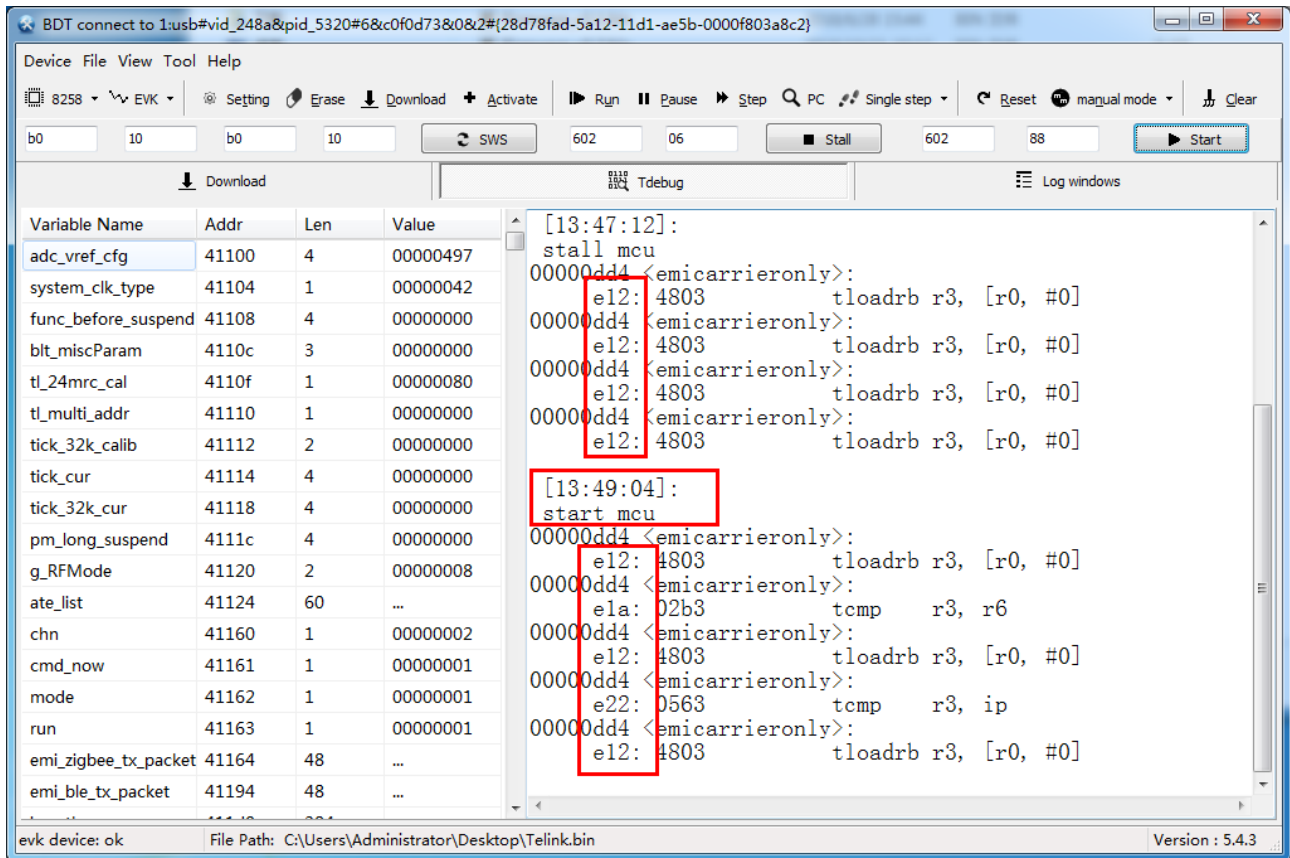


Figure 2.81: Start MCU

2.7 Single-wire synchronization

Before setting Single-wire synchronization speed, please ensure the following items:

- (1) Power supply is normal;
- (2) MCU is not in "Low Power" mode;
- (3) Single wire function of MCU is available;
- (4) System clock is normal.

When connection fails to be established between "Burning EVK" and the target board, user can try to set Swire(single wire) synchronization speed to improve communication. **Note:** Swire register address may vary with different chip types.

- (1) Change the speed of Master and the speed of Slave in the four editable boxes, as shown below. Note that Swire register address may vary with chip types.
- (2) Click the "SWS" button to implement Swire synchronization.

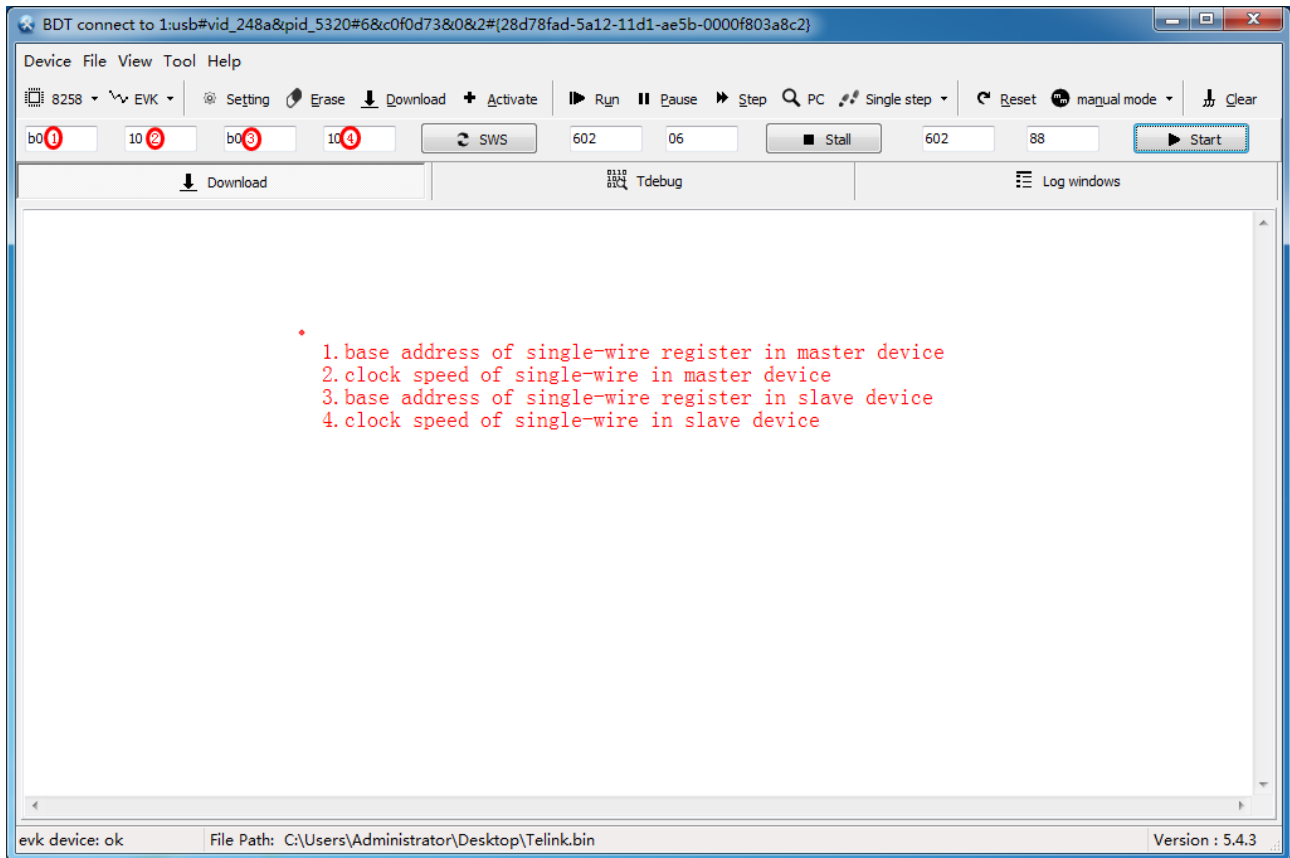


Figure 2.82: Set Swire synchronization speed

Before downloading firmware to target board or debugging MCU, it is suggested that each time the MCU is powered on, user should implement one Swire synchronization to check whether communication with target board is normal.

If the status of communication with target board is wrong, user can solve the problem according to the methods mentioned at the beginning of [section 2.5](#).

2.8 Command Line

Users can also use the command line to access memory or download files to the target board. In the following subsections, “read core” is used as an example.

2.8.1 Single Command Line

Step 1: Open command line input window by selecting the “Tool->Cmdline Input” or pressing the shortcut key combination “Ctrl + I”.

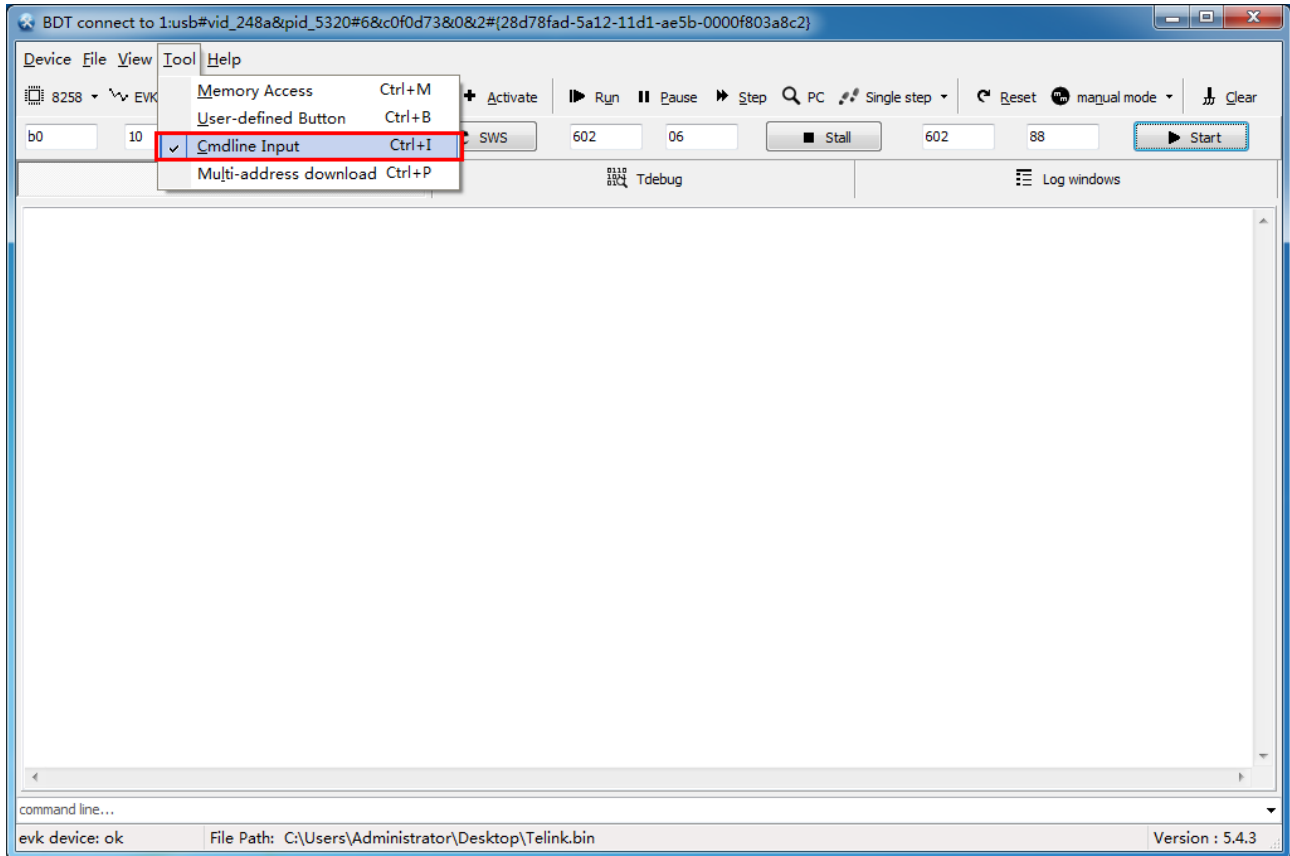


Figure 2.83: Open cmdline input window

Step 2: Input the needed single command as follows and then press the “Enter” key. The file with the specified named will be available in the specified location.

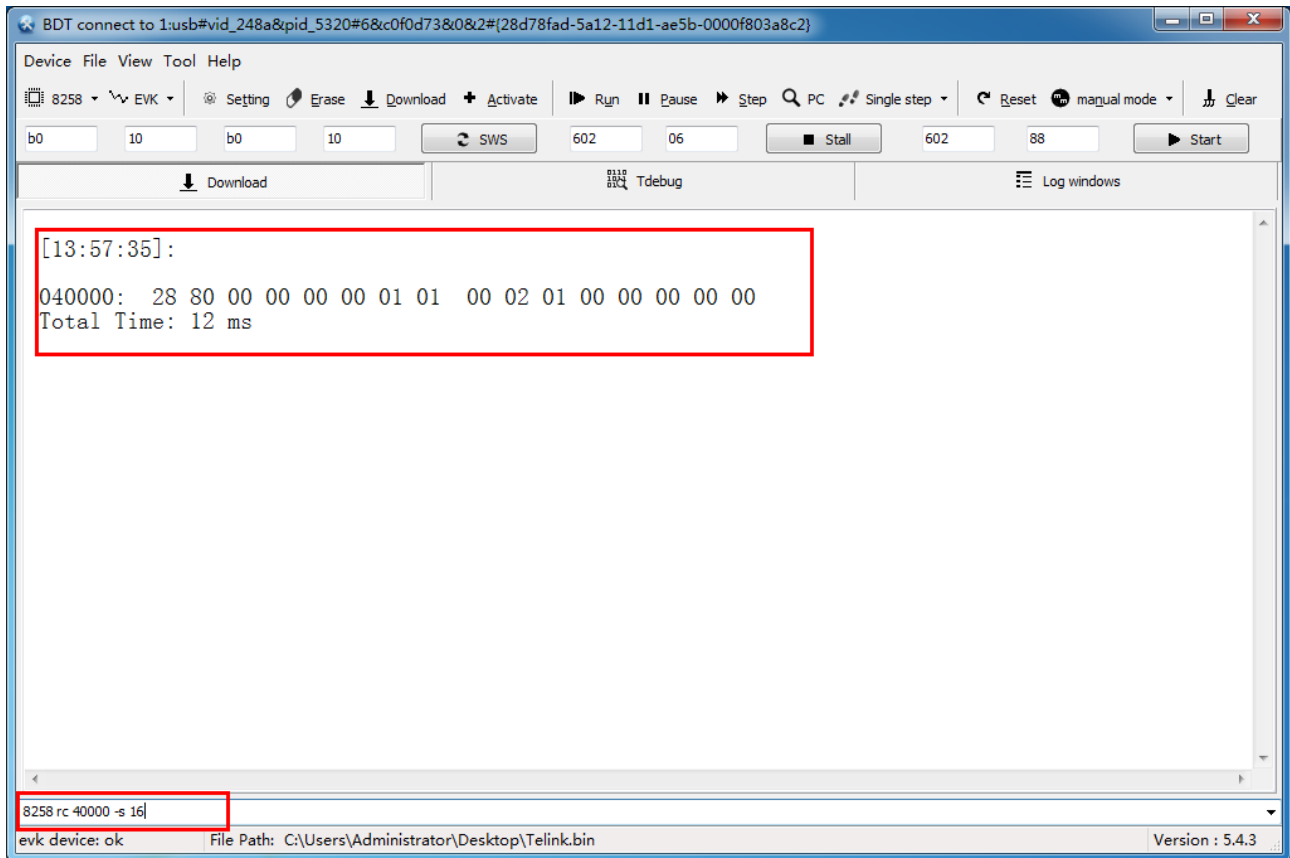


Figure 2.84: Enter single command line

2.8.2 Multi-command Line

The tool supports multi-command line operation.

All commands need to be separated from each other by semicolon, as shown below:

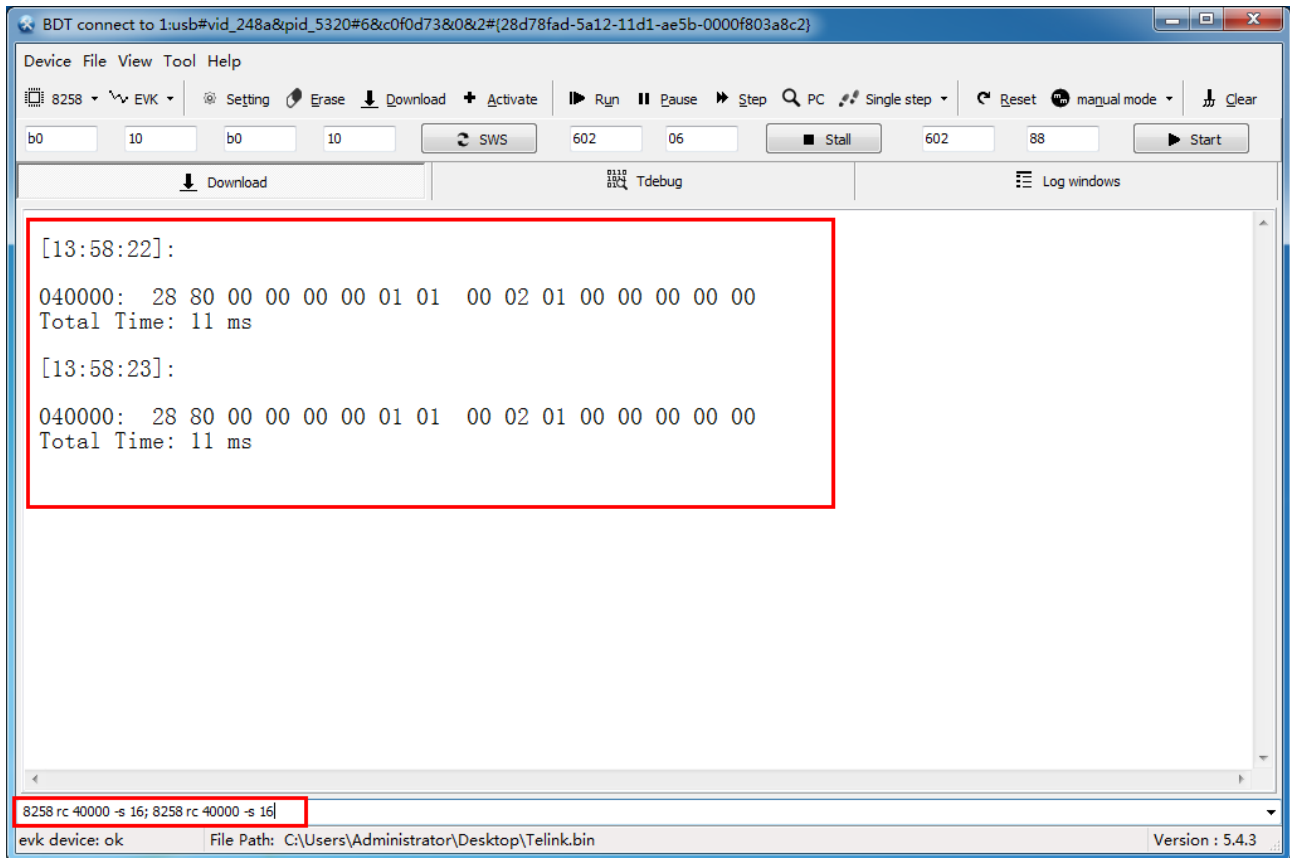


Figure 2.85: Enter multiple commands

2.9 Log Window

This tool supports USB print function, so MCUs with USB function can use USB print function to output information in log window on PC side.

Step 1: Click the "View -> usb log" to open usb log window.

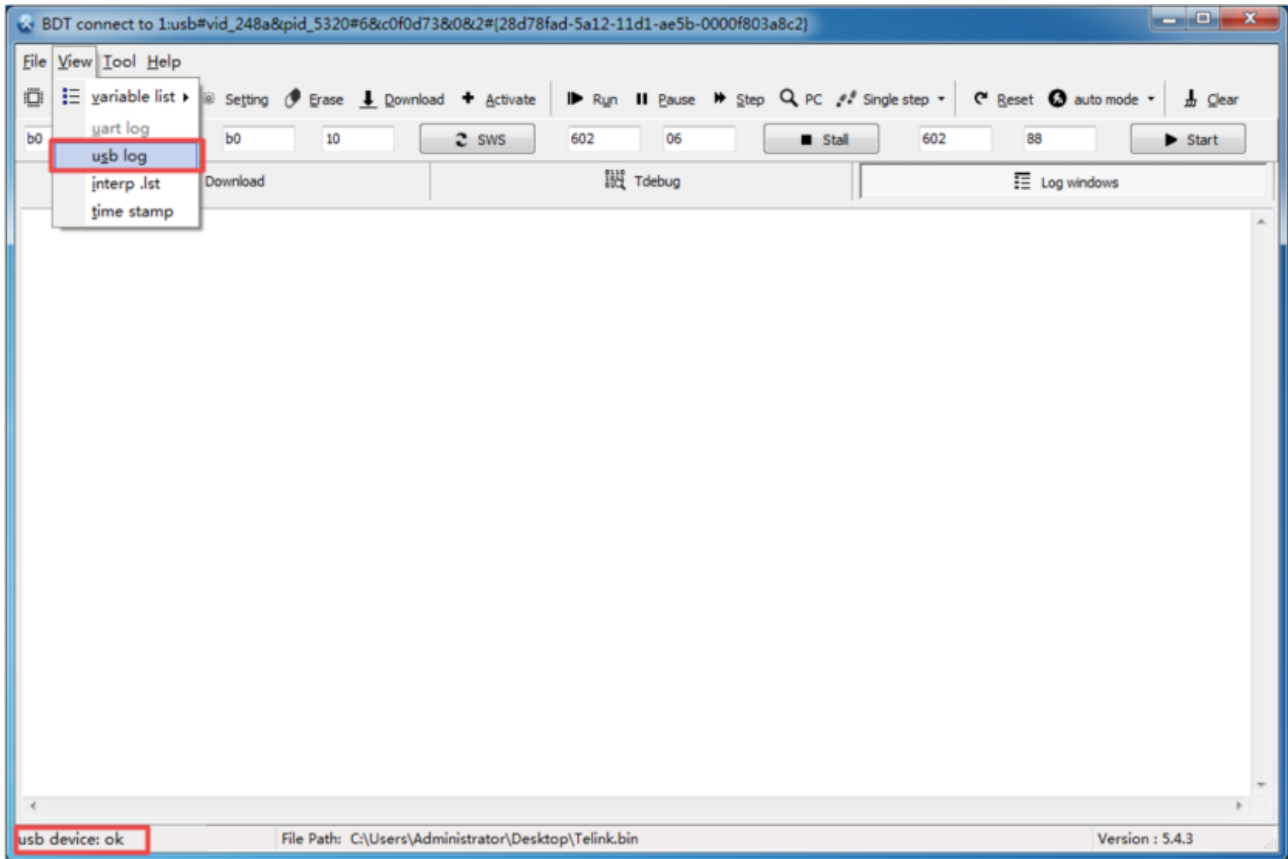


Figure 2.86: Open usb log window

Step 2: In the "usb log" mode, all output characters will be displayed in "Ascii" mode.

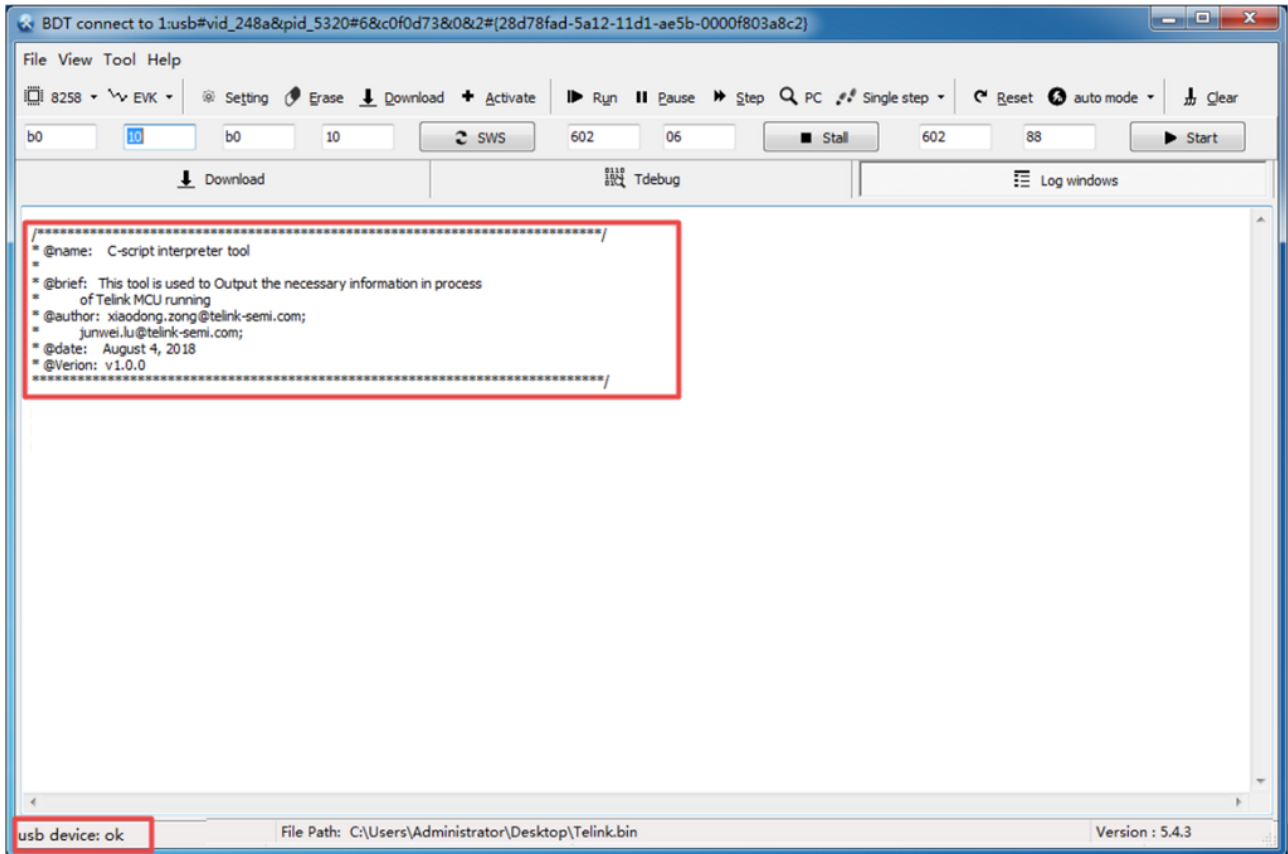


Figure 2.87: Display output characters

Step 3: Click the “View -> usb log” again to exit the “usb log” mode.

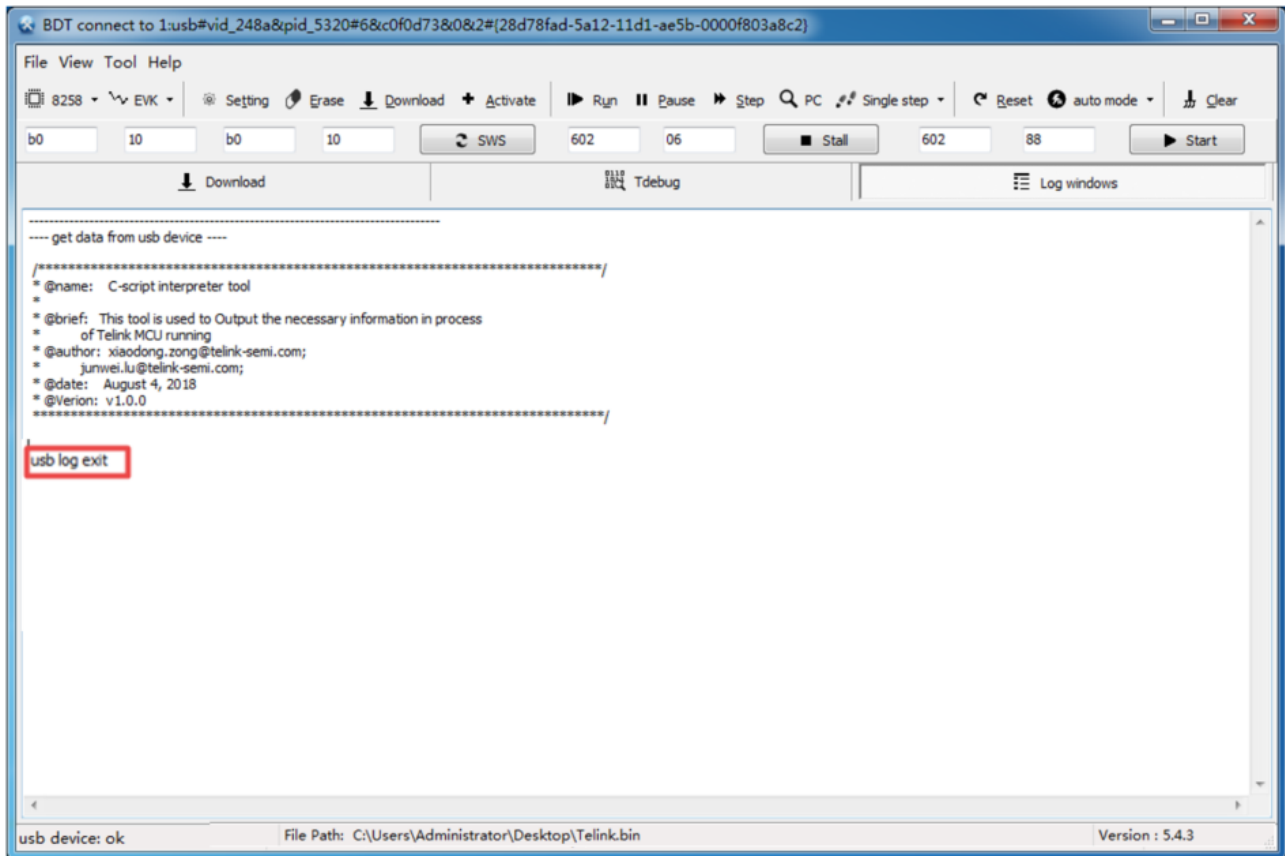


Figure 2.88: Exit “usb log” mode

2.10 Help

By clicking the “Help -> Command line”, all commands will be shown in the log windows of the “Download” or “Tdebug” page.

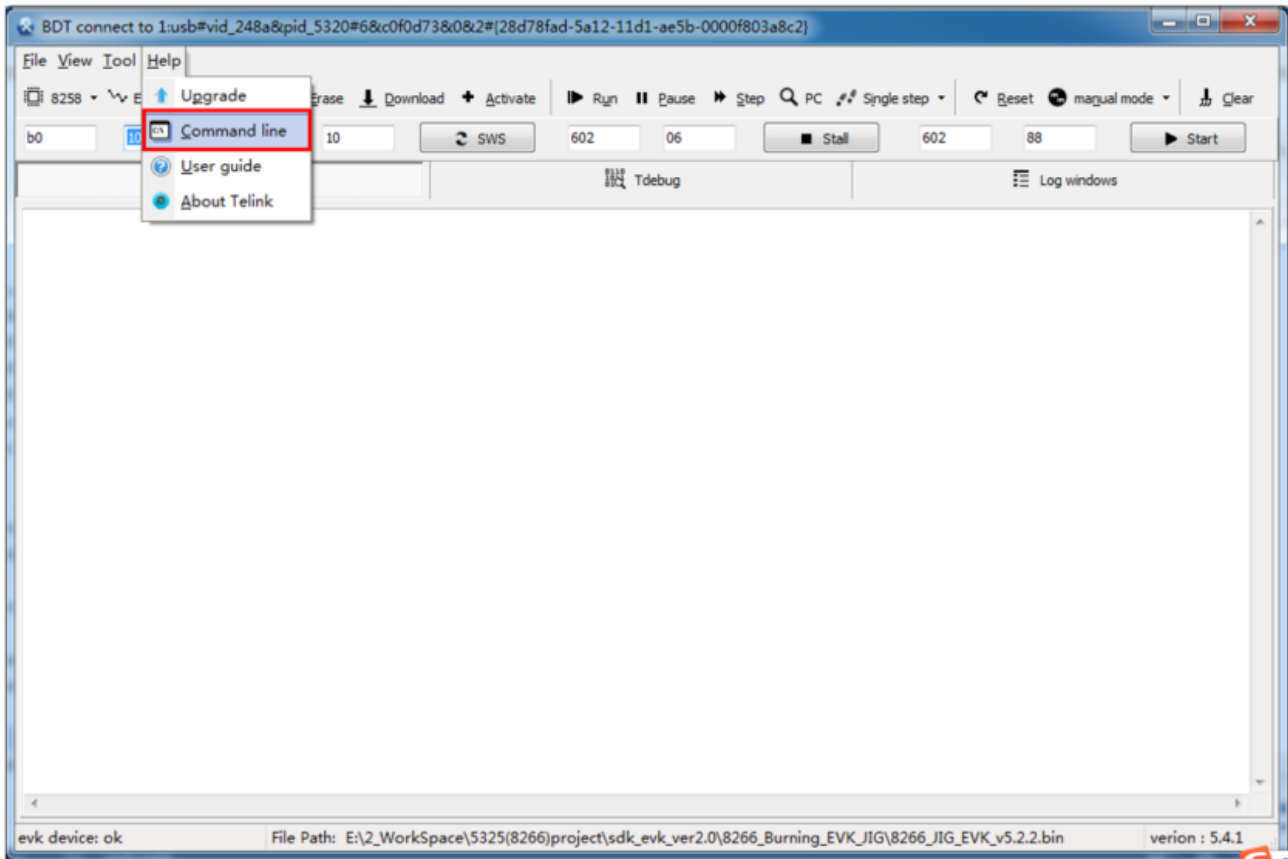


Figure 2.89: Command line

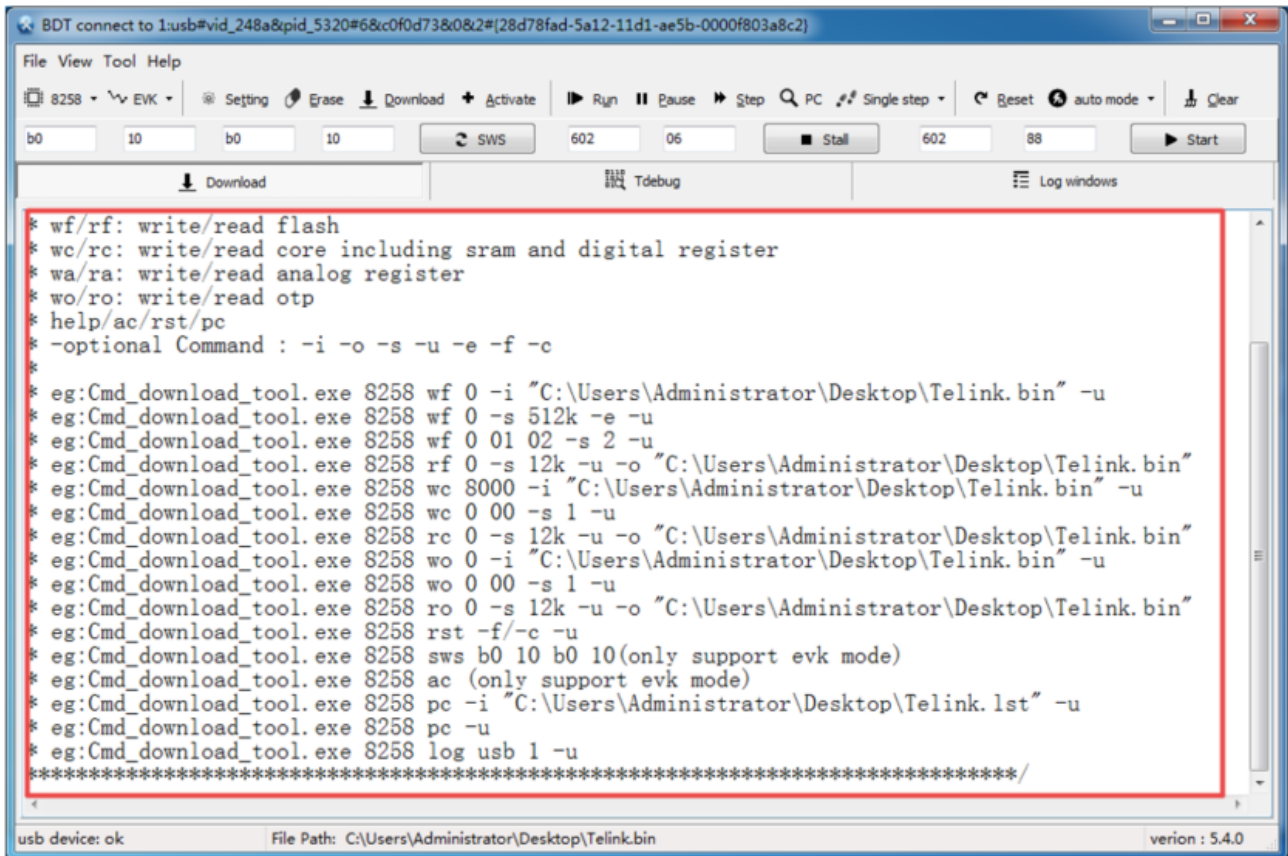


Figure 2.90: Show all commands

By clicking the "Help -> User guide", the embedded "Help" document "Telink Burning and Debugging Tool (BDT) User Guide" will be opened, as shown below:

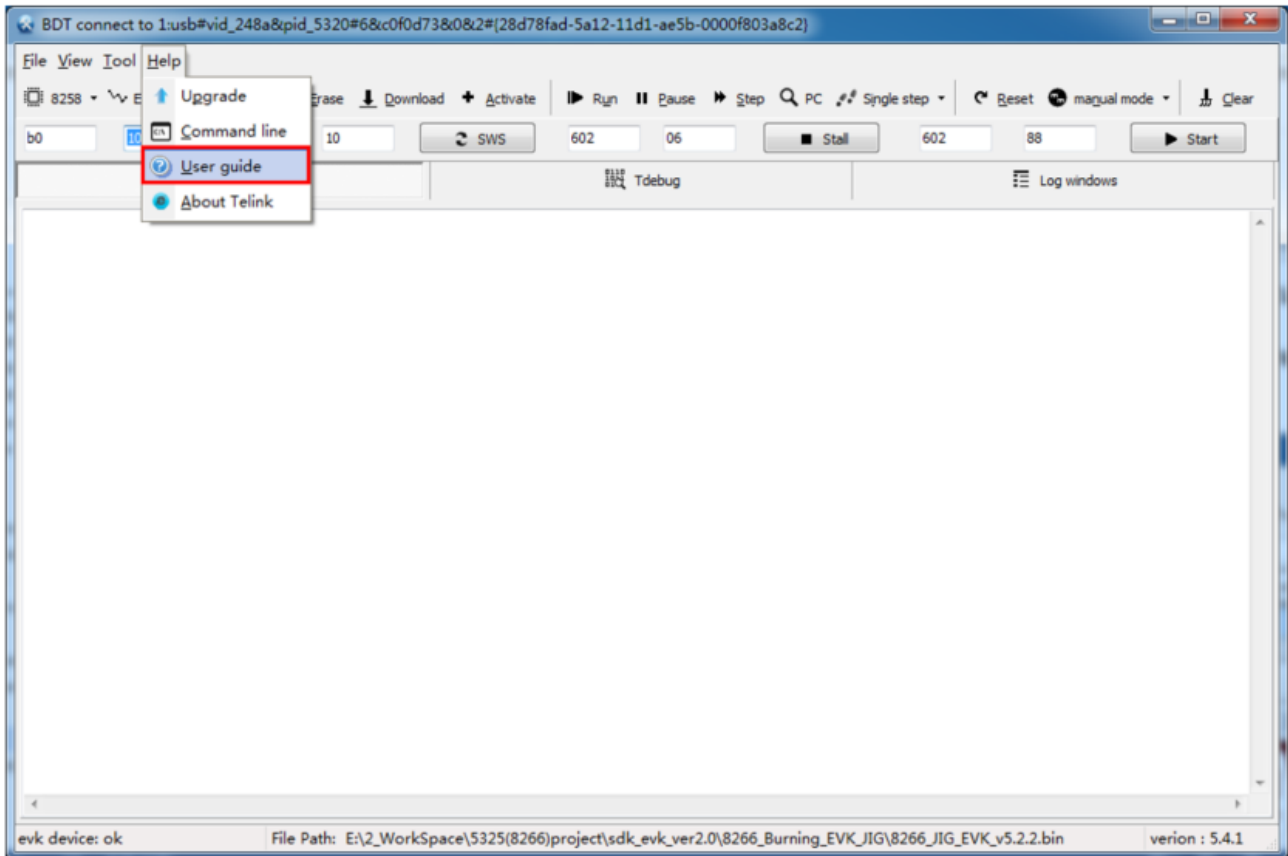


Figure 2.91: User guide

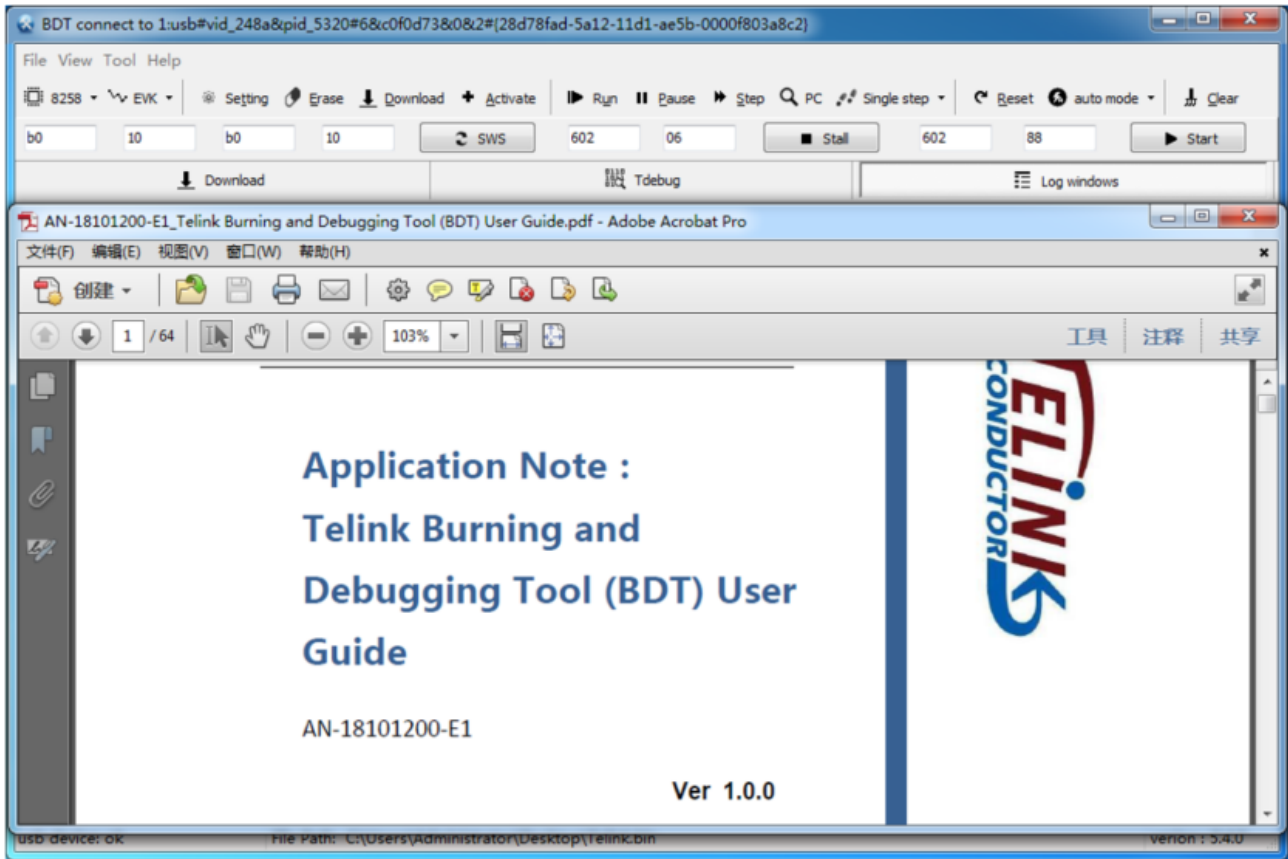


Figure 2.92: Open Help document

To get more information about Telink, user can click the "Help -> About Telink".

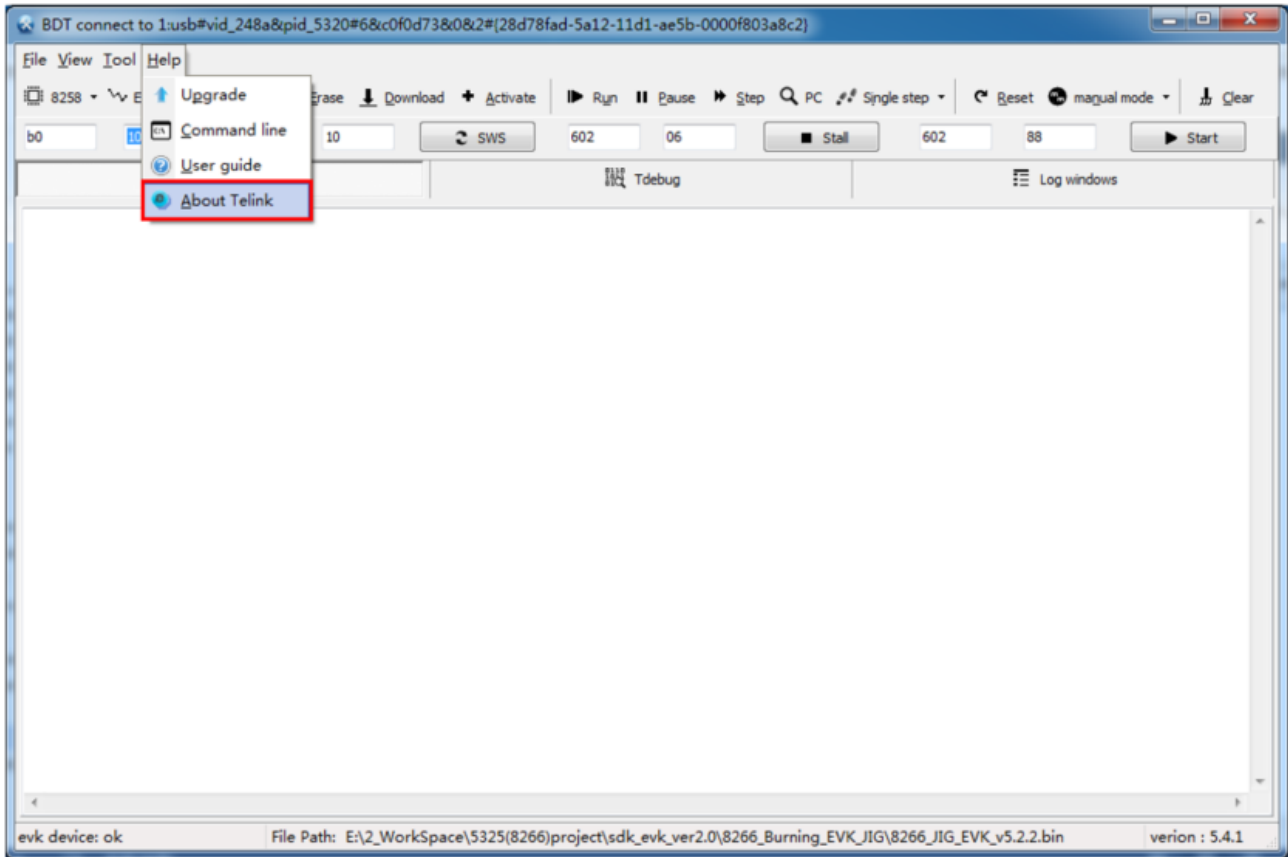


Figure 2.93: About Telink

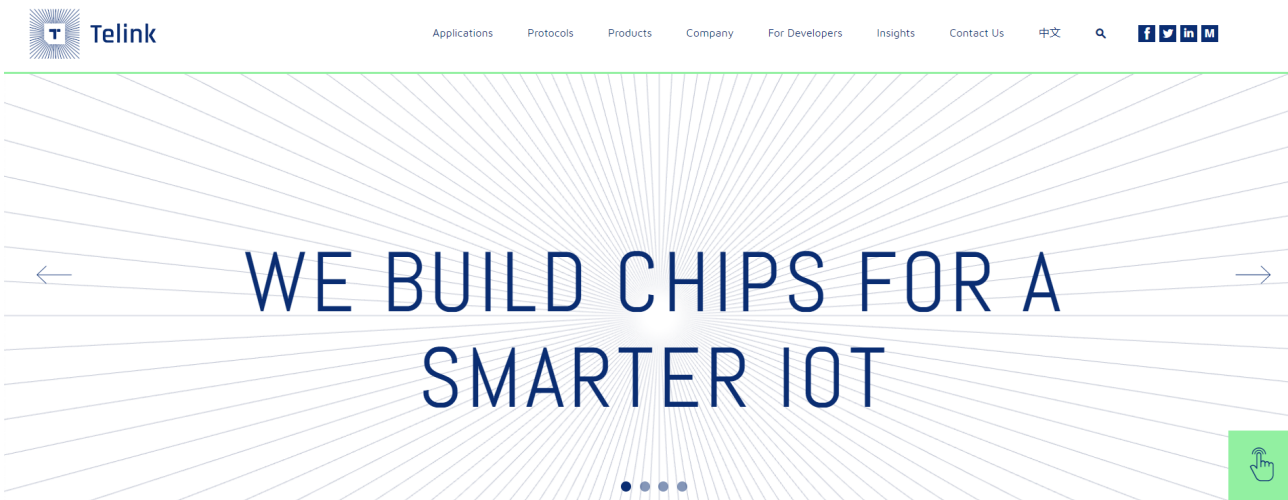


Figure 2.94: Get information about Telink

2.11 Default Configuration

Open the “config” folder in the root directory of this tool as shown below.

名称	修改日期	类型	大小
bin	2018/10/9 15:17	文件夹	
user	2018/10/9 9:31	文件夹	
Cmd_download_tool.exe	2018/10/10 10:24	应用程序	2,197 KB
config.ini	2018/10/9 18:07	INI 文件	3 KB
tl_usb2.0.dll	2017/12/1 14:02	DLL 文件	100 KB

Figure 2.95: Folder “config”

All default configurations in this tool can be changed by modifying the “config.ini” file. Following shows detailed explanations.

```
[MCU]
TYPE 1 0x01 - 8368 \bin\dut_8368_flash_v0244.bin 8000 b010b010 \bin\dut_8366_otp_v0005.bin \bin\dut_8366_otp_normal_v0002.bin End
TYPE 1 0x02 - 8368 \bin\dut_8368_flash_v0246.bin 8000 b010b010 \bin\dut_8368_otp_v0005.bin \bin\dut_8368_otp_normal_v0002.bin End
TYPE 2 0x03 - 8232 \bin\dut_8232_flash_v0002.bin 8000 b010b010 End
TYPE 2 0x04 - 8233 \bin\dut_8233_flash_v0002.bin 8000 b010b010 End
TYPE 2 0x05 - 8266 \bin\dut_8266_flash_v0244.bin 8000 b010b010 End
TYPE 2 0x06 - 8267 \bin\dut_8267_flash_v0002.bin 8000 b010b010 End
TYPE 3 0x07 - 8255 \bin\dut_8255_flash_v0002.bin 40000 b010b010 End
TYPE 3 0x08 - 8258 \bin\dut_8258_flash_v0003.bin 40000 b010b010 End
default - 8258 End
```

Figure 2.96: MCU

- (1) MCU type: 8368
- (2) The path of the firmware: \bin\dut_8368_flash_v0246.bin

This firmware is used to operate flash, e.g. download the specified firmware into flash, read/write flash, erase sectors of flash and check flash data when downloading firmware to flash or erase some sector of flash.

- (3) SRAM starting address: 0x8000, e.g. 8258: 0x40000;
- (4) Single wire communication:
 - a. Swire register base address of Master device: 0xb0
 - b. Swire clock of Master device: 0x10
 - c. Swire register base address of Slave device: 0xb0
 - d. Swire clock of Slave device: 0x10

- (5) The path of the firmware:

Firmware1: \bin\dut_8368_otp_v0005.bin, which is used to operate OTP, e.g. download firmware to OTP, read/write OTP, check OTP data during writing data to OTP or downloading firmware to OTP.

Firmware2: \bin\dut_8368_otp_normal_v0002.bin, which is used to check OTP data during writing data to OTP or downloading firmware to OTP.

- (6) default: 8258

```
[MODE]
MODE 0x01 - USB End
MODE 0x04 - SWK End
default - SWK End
```

Figure 2.97: Mode of Communication

- (1) Communication mode: USB
- (2) default: EVK (Burning EVK)

```
[OBJ]
OBJ 0x01 = 1 End
OBJ 0x02 = CORE End
OBJ 0x03 = ANALOG End
OBJ 0x04 = GPP End
default = 2 End
```

Figure 2.98: Objection

- (1) Access objection: Flash
- (2) default: Core

```
[SIZE]
SIZE 0x01 = 1 End
SIZE 0x02 = 1 End
SIZE 0x03 = 1 End
SIZE 0x04 = 8 End
SIZE 0x05 = 16 End
default = 2 End
```

Figure 2.99: Size

- (1) Size of Read / Write data: 2
- (2) default: 1

```
[DOWNLOAD_ADDR]
DOWNLOAD 0x01 = 0 End
DOWNLOAD 0x02 = 20000 End
DOWNLOAD 0x03 = 40000 End
default = 2 End
```

Figure 2.100: Download address

- (1) Download address: 0x20000
- (2) default: 0

```
[ERASE_ADDR]
ERASE 0x01 = 0 End
ERASE 0x02 = 20000 End
ERASE 0x03 = 40000 End
default = 2 End
```

Figure 2.101: Erase address

- (1) Erase address: 0x20000
- (2) default: 0

```
[PAGE_DISPLAY]
PAGE 0x01 = 1 Download End
PAGE 0x02 = 1 Debug End
PAGE 0x03 = 2 Log windows End
default = 2 End
```

Figure 2.102: Page display

- (1) Page display: 1 - Tdebug
- (2) default: 0 - Download

```
[BUTTON]
Button 1 = example1 external_exe E:\1_wtcd\tcdb.exe uart -1 -u End
Button 2 = example2 internal_exe 8266 wf 34000 -e 48k ;8266 rc 00 -s 10 -u; End
Button 3 = example3 external_file E:\1_wtcd\prj\kite\doc\TS5565_AFEpin_description_v1.0.xlsx End
/But 4 End
/Buttons 5 End
/Buttons 6 End
/Buttons 7 End
/Buttons 8 End
/Buttons 9 End
/Buttons 10 End
/Buttons 11 End
/Buttons 12 End
/Buttons 13 End
/Buttons 14 End
```

Figure 2.103: Button

- (1) The name of Button 1: example1
- (2) The property of Button 1: external executable application
- (3) The path of external executable application: E:\1_wtcd\tcdb.exe uart -1 -u
- (4) Undefined Button: Button 4

The property of Button includes three types:

- a. external_exe: external executable application
- b. internal_exe: internal executable application, which means this tool.
- c. external_file: open any external file by default

```
[PATH]
PATH 0x01 = C:\Users\Administrator\Desktop\Telink.bin End
default = 0x02 End
```

Figure 2.104: Path

- (1) The path of firmware: "C:\Users\Administrator\Desktop\Telink.bin"

Download this firmware into target board to update firmware. When there're too many file paths, you can directly delete these. However, the first path can't be deleted.

- (2) Default file path: "C:\Users\Administrator\Desktop\Telink.bin"

3 Upgrade Firmware

User can follow the steps below to upgrade firmware of "Burning EVK".

Step 1: Click "Help -> Upgrade" to open the "Upgrade EVK" dialog window.

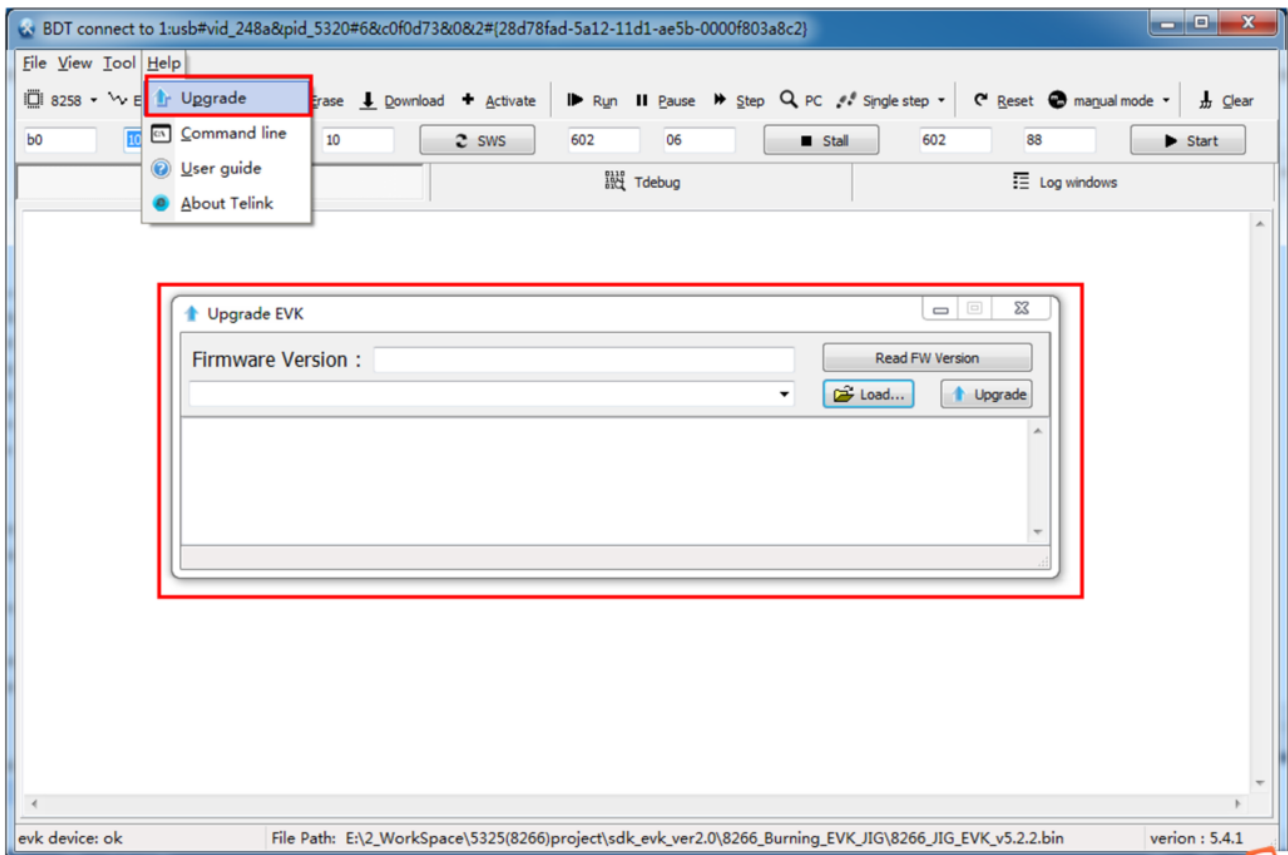


Figure 3.1: "Upgrade EVK" window

Step 2: Click the button "Read FW Version" to check the current firmware version.

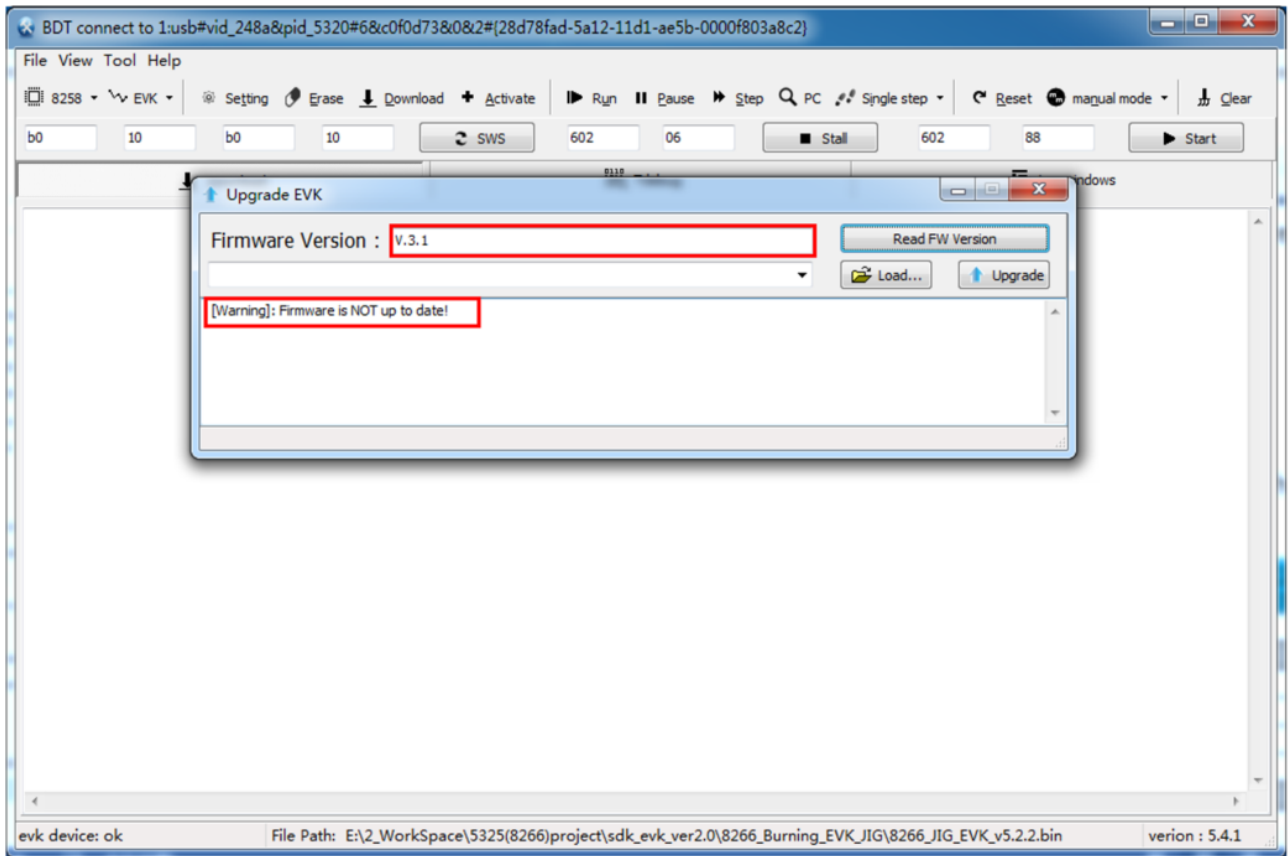


Figure 3.2: Check current firmware version

Step 3: Click the “Load...” button to load the new firmware.

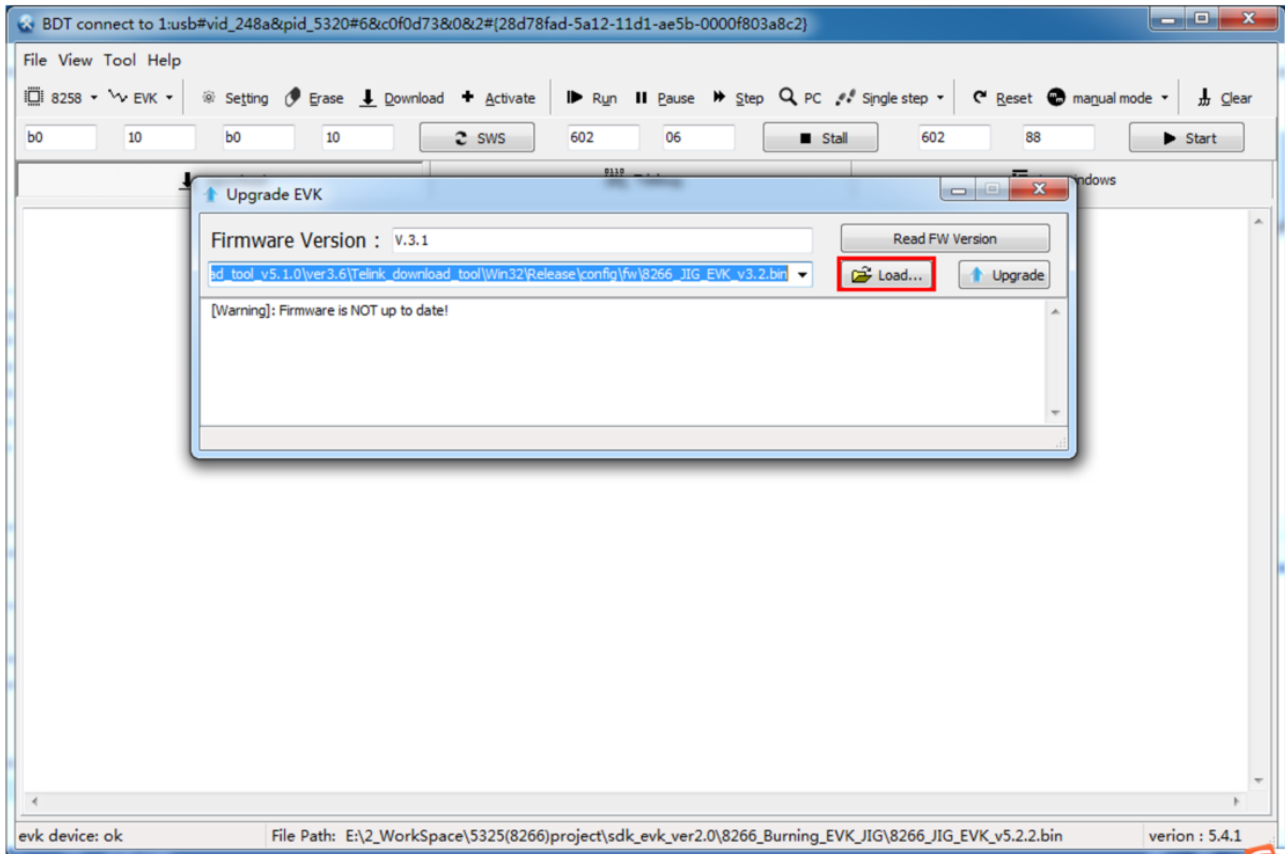


Figure 3.3: Load new firmware

Step 4: Click the “Upgrade” button to upgrade the new firmware into the “Burning EVK”. **Note:** User must power cycle the EVK before using it.

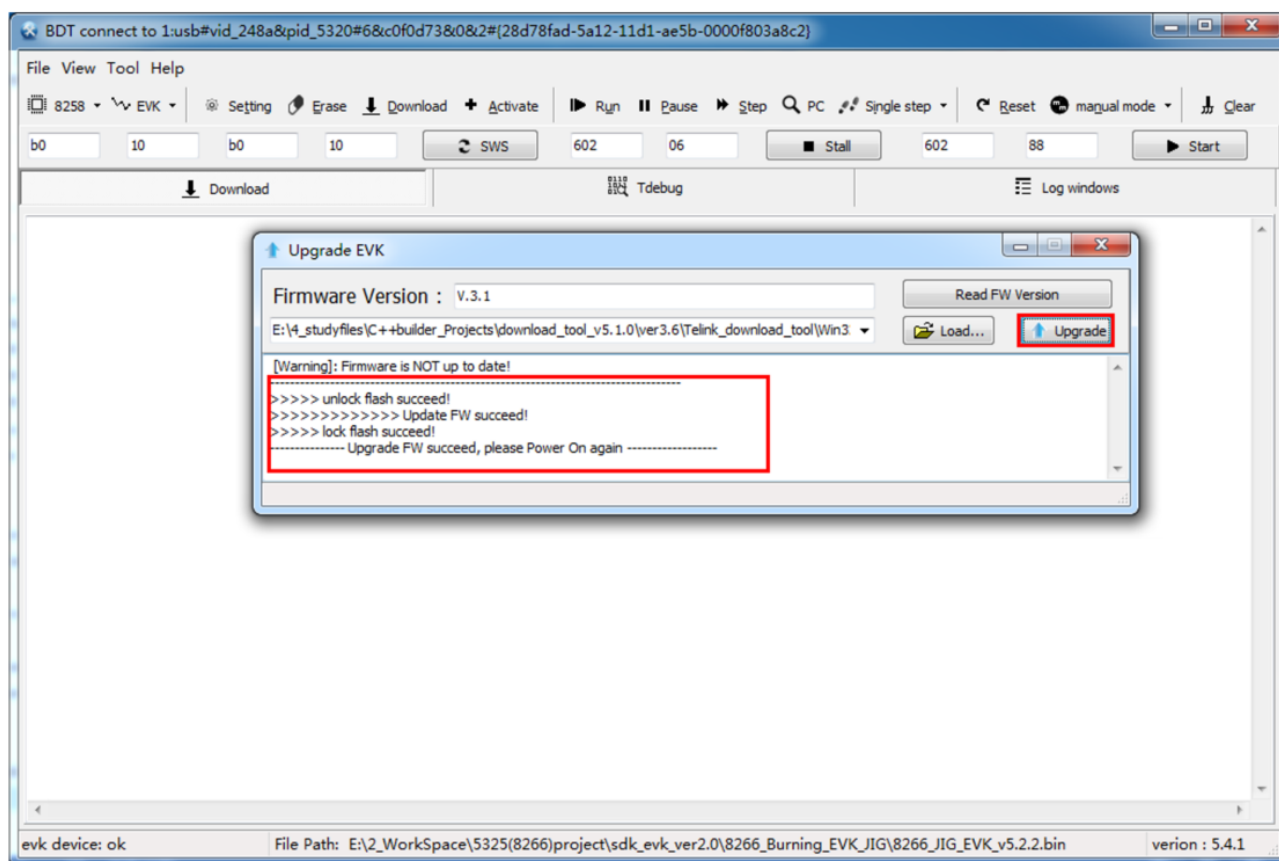


Figure 3.4: Upgrade firmware into EVK

4 Tools Guide

This section mainly introduces the built-in tools of the BDT, including the functions of the tools and how to use them.

4.1 BQB Tool

4.1.1 Function Description (BQB Tool)

This tool has two main functions:

- Configure and generate firmware for BQB testing according to requirements (bin files of both Flash and SRAM types), with configurable options including: UART pins, PA control pins, TX energy, internal and external capacitors, internal power supply mode and the location of calibration parameters. In addition, the generated firmware can be downloaded directly into the DUT through the tool (by way of Swire). **Note:** The above BQB test firmware is a UART two-wire test program, and the baud rate of serial communication is 115200.
- Sending and receiving packet test

4.1.2 Interface Composition and Instructions (BQB Tool)

The tool is mainly composed of two parts: the firmware download interface and the test interface, as shown in the following figure.

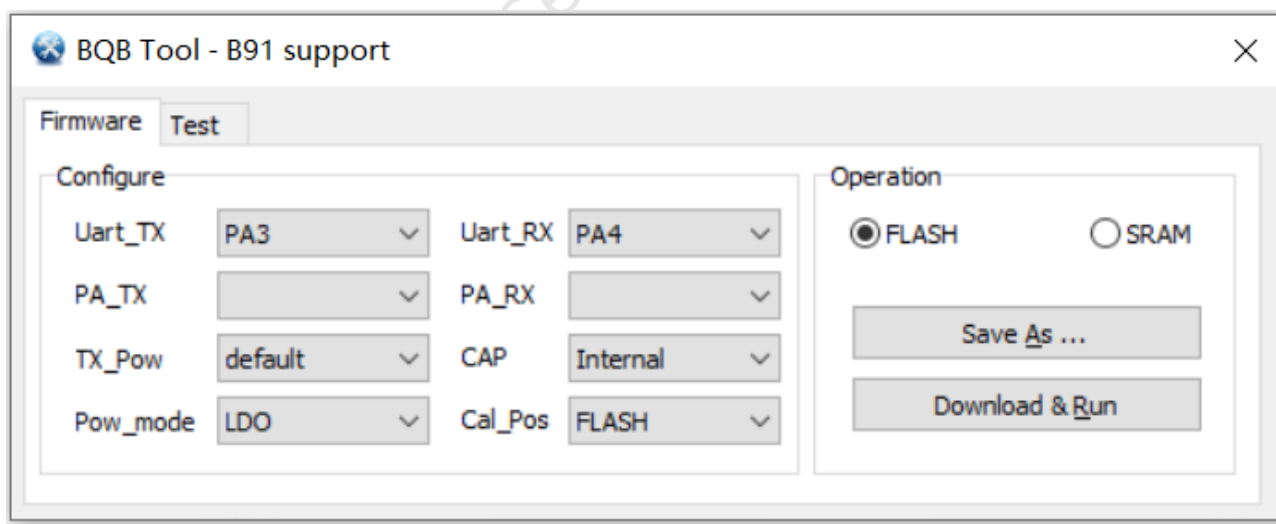


Figure 4.1: BQB tool firmware download interface

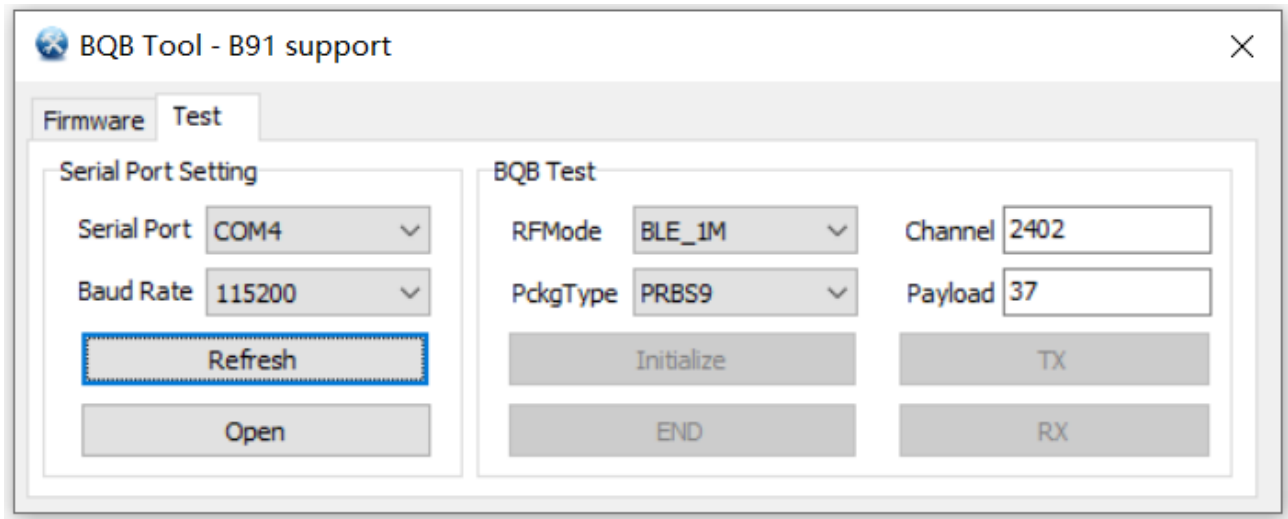


Figure 4.2: BQB tool test interface

4.1.2.1 Chip Selection

Selection of chip type through the MCU list in the BDT main interface, after selecting the chip, you can judge whether the tool supports the corresponding chip through the title bar of the BQB Tool interface.

4.1.2.2 Firmware Saving and Downloading

Step1 - Configure firmware: In the "Configure" area of the firmware download interface, Uart_TX(RX) is used to configure the TX and RX pins of the UART, PA_TX(RX) is used to configure the TX and RX pins of the PA, TX_Pow is used to configure the energy of sending packets (default is the default value, usually 7 dBm), CAP is used to configure the internal and external capacitors, Pow_mode is used to configure the power supply mode inside the chip, and Cal_Pos is used to configure the storage location of the calibration data (selected according to the actual situation, SRAM can be selected during testing).

Step2 - Select firmware type: In the "Operation" area of the firmware download interface, select the type through the check box, and the available types include Flash program and SRAM program.

Step3 - Save/Download firmware: In the "Operation" area of the firmware download interface, click the "Save As..." button to save the firmware, and click the "Download & Run" button to download the firmware.

4.1.2.3 Sending and Receiving Packet Test

Step1 - Connecting the serial port: In the "Serial Port Setting" area of the test interface, "SerialPort" is an optional serial port (if the list is empty, it can be updated by the "Refresh" button), and "BaudRate" is the baud rate of communication (115200 is selected by default, which is consistent with the firmware's serial communication rate), click the "Open" button to open the serial port after it is configured.

Step2 - Configuring the test mode: In the "BQB Test" area of the test interface, RFMode is used to select the RF mode, Channel is used to configure the test frequency point, PckgType is used to select the data type of the packet, and Payload is used to configure the data length of the packet.

Step3 - Testing: In the “BQB Test” area of the test interface, the “Initialize” button is used to initialize the state of the DUT, the “TX” button is used to control the DUT to send packets, the “RX” button is used to control the DUT to receive packets, and the “END” button is used to control the DUT to stop receiving or sending packets (Click the “END” button in the receiving state to count the number of received packets).

4.2 Jtag Tool

4.2.1 Function Description (Jtag Tool)

The main functions of this tool are as follows:

- Install Jtag hardware related drivers
- Download/Erase firmware through Jtag
- Read/Write parameters through Jtag
- Debug through Jtag

4.2.2 Interface Composition and Instructions (Jtag Tool)

The interface of the Jtag tool consists of two parts, the “Target” area and the “Debug” area, as shown in the figure below. The “Target” area is responsible for connecting devices and installing Jtag device drivers, while the “Debug” area is mainly responsible for reading, writing, erasing and debugging functions.

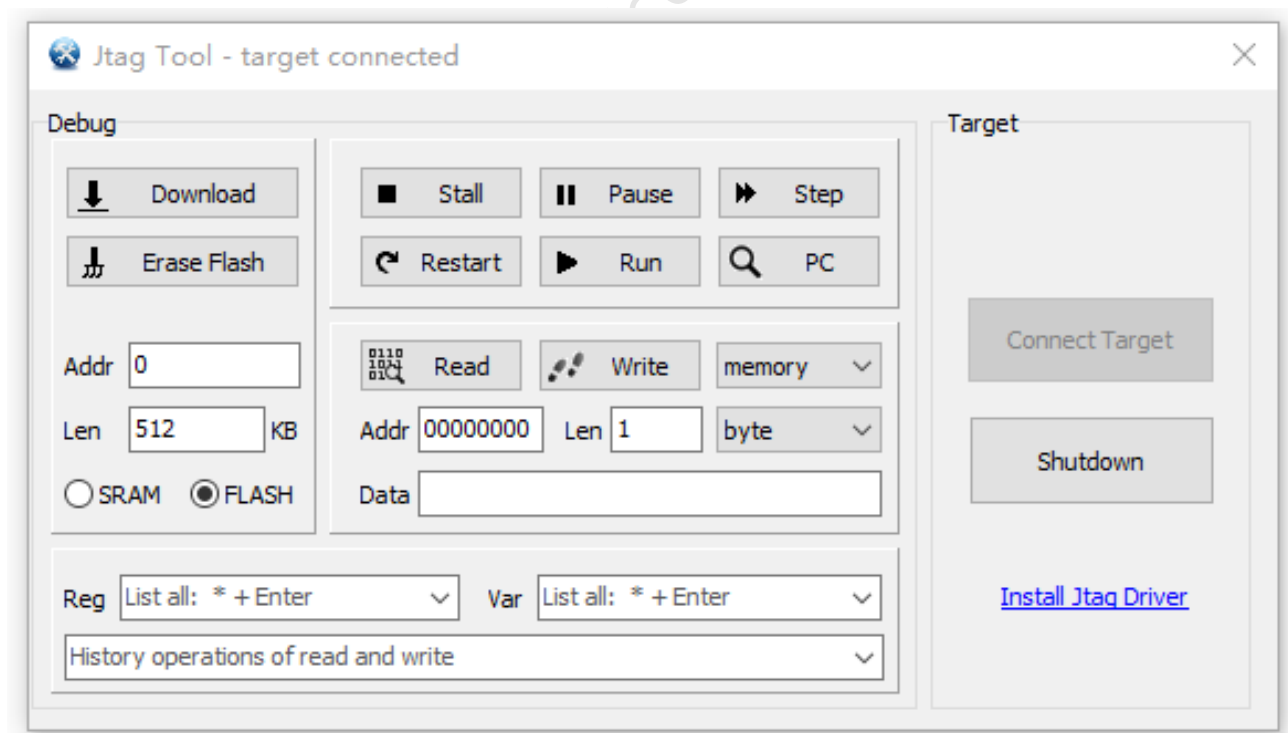


Figure 4.3: Jtag tool interface

4.2.2.1 Jtag Device Driver Installation

Step1 - Connect Jtag devices: The Jtag device is shown below and is connected to the PC via USB.



Figure 4.4: Jtag device

Step2 - Check device drivers: Open the device manager to view the device, if the driver is installed, it will be recognized as a "libusbK USB Devices" normally, as shown below.

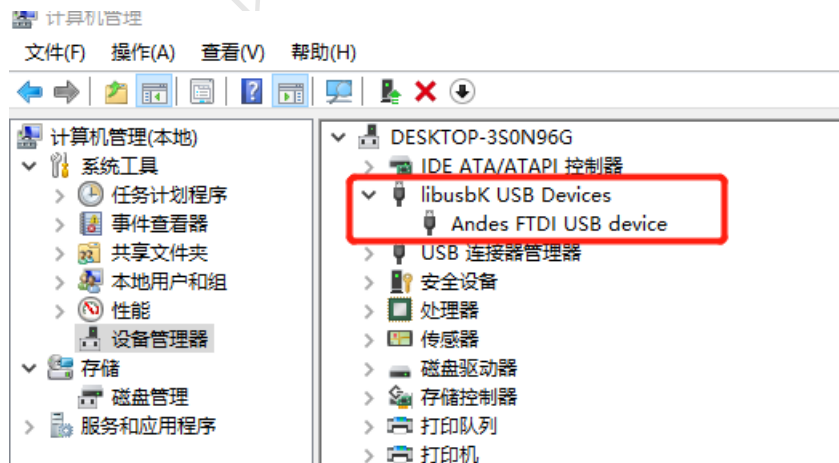


Figure 4.5: Jtag device driver installed

If the driver is not installed properly, it will be as shown in the figure below.

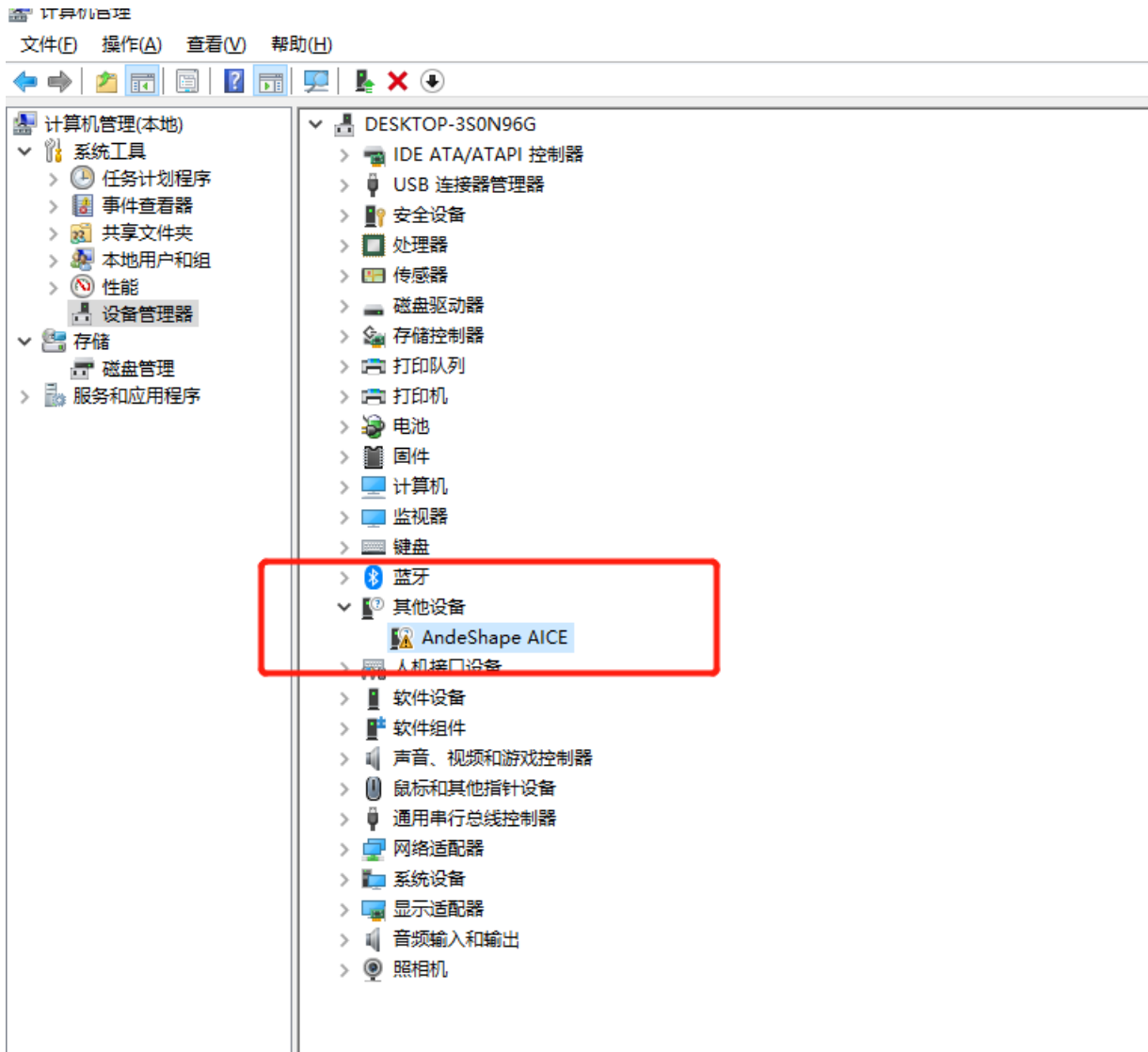


Figure 4.6: Jtag device driver is not installed

Step3 - Install driver: In the “Target” area of the Jtag tool interface, click “Install Jtag Driver” below to install the driver.

4.2.2.2 Connecting Debug Equipment

Step1 - Connecting debug equipment: Connect the debug device to the Jtag device through the Jtag interface. In the normal connection state, only the blue light of the Jtag device is always on. If the red light is always on at the same time, first check whether the Jtag driver is installed. If the driver installation is normal, then check whether the connection between the Jtag box and the test board is normal.

Step2 - Connecting target devices: After the debugging device is successfully connected, click the “Connect Target” button in the “Target” area of the Jtag tool interface to connect the debugging device. If the title bar of the Jtag tool interface displays “Jtag Tool – target connected”, it means that the connection is successful, and the controls in the “Debug” area become operable. If the connection is unsuccessful, it can be solved by plugging and unplugging the USB interface.

4.2.2.3 Download Function

Step1 - Select firmware: On the main interface of the BDT tool, click "File->Open" to select the firmware to be downloaded.

Step2 - Set the starting address: In the "Debug" area of the Jtag tool main interface, enter the starting address of the download in the input box corresponding to "Addr".

Step3 - Set the object loaded by the firmware: In the "Debug" area of the Jtag tool main interface, the object can be selected by clicking on the check box controls "FLASH" and "SRAM".

Step4 - Perform downloading: In the "Debug" area of the Jtag tool main interface, click the "Download" button to perform the download operation.

Note:

Programs downloaded into Flash are run by clicking the "Restart" button, while programs downloaded into SRAM are run by clicking the "Run" button.

4.2.2.4 Erase Function

Step1 - Set the starting address: In the "Debug" area of the Jtag tool main interface, enter the starting address of the download in the input box corresponding to "Addr".

Step2 - Set the erase length: In the "Debug" area of the Jtag tool main interface, enter the length to be erased in the input box corresponding to "Len" below the "Erase Flash" button.

Step3 - Perform erase: In the "Debug" area of the Jtag tool main interface, click the "Erase Flash" button to perform the erase operation.

4.2.2.5 Read/Write Functions

The tool supports reading and writing in three modes: Memory, Flash and Analog. Among them, the Memory mode can support reading and writing in byte, half-word (2 byte) and word (4 byte), while the other modes only support reading and writing by byte.

Step1 - Set the read/write mode: In the "Debug" area of the Jtag tool main interface, click the drop-down box behind the "Write" button to select the mode.

Step2 - Set the read/write width: In the "Debug" area of the Jtag tool main interface, click the drop-down box below the read/write mode selection box to select the read/write width.

Step3 - Set the read/write address and length: In the "Debug" area of the Jtag tool main interface, enter the address in the input box corresponding to "Addr" below the "Write" button, and enter the length in the input box corresponding to "Len". If you are writing data, you also need to enter the data to be written in the input box corresponding to "Data".

Step4 - Perform read/write operations: In the "Debug" area of the Jtag tool main interface, click the "Read" button to perform the read operation, and click the "Write" button to perform the write operation.

4.2.2.6 Debug Function

The debugging functions supported by the tool include: reading status registers, reading variable values, and common debugging operations.

Step1 - Reading the status register: In the "Debug" area of the Jtag tool main interface, enter the name of the status register in the input box corresponding to "Reg", then press the "Enter" key to find the corresponding register, and then click the drop-down box to select the corresponding register to view the value of this register. If you don't know the name of the register, you can also list all viewable registers by entering "**".

Step2 - Reading variable values: In the "Debug" area of the Jtag tool main interface, enter the name of the variable in the input box corresponding to "Var", then press the "Enter" key to find the corresponding variable, and then click the drop-down box to select the corresponding variable to view the value of this variable. If you don't know the variable name, you can also list all viewable variables by entering "**". **Note:** To implement this function, you need to ensure that a list file with the same name exists in the bin file directory.

Step3 - Common debugging operations: In the "Debug" area of the Jtag tool main interface, click the "Stall" button to stop the device, Click the "Restart" button to make the device reload the firmware and run from the flash, click the "Pause" button to pause the device program, click the "Run" button to make the device continue to run from the program pause, click the "Step" button to make the device program run in one step, and click the "PC" button to view the device's current PC pointer. In addition, if there is a list file with the same name in the bin file directory, you can also view the corresponding assembly code, but before using this function, you need to enter "**" in the input box corresponding to "Var" and press the "Enter" key.

4.3 TWS Tool

4.3.1 Download Tool

4.3.1.1 Function Description (Download Tool)

"Download Tool" is mainly used to burn bin files, parameter information, etc. to the specified address of Flash. This tool supports burning single and multiple objects at once, and supports burning up to 4 bin files, 1 set of "BT Name" parameters and 1 set of "BT MAC" parameters at a time.

4.3.1.2 Interface Introduction

The interface of "Download Tool" is shown in the figure below, which mainly includes two areas: "Setting" and "Operation". The "Setting" area is responsible for setting the download information, and the "Operation" part is responsible for executing the download operation.

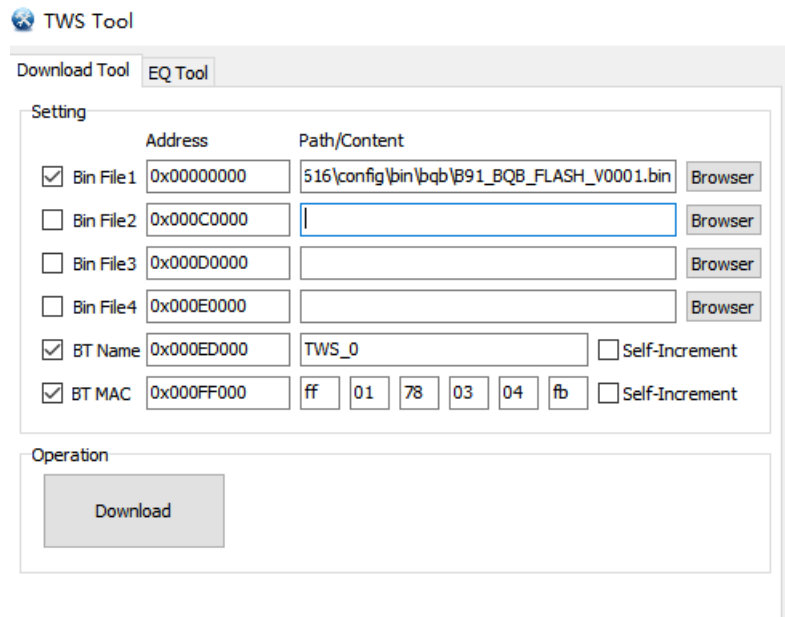


Figure 4.7: TWS_Tool/Download_Tool interface

4.3.1.3 Instructions for Use

Step1 - Set burning information: In the "Setting" area of the Download tool main interface, select the information to be burned through the radio button on the far left, enter the Flash address written in each information in the input box corresponding to "Address", and enter the corresponding information in the input box corresponding to "Path/Content". Both "BT Name/MAC" have a self-increasing function (The value corresponding to each set of data will increase automatically after burning), which can be selected via the check box on the right. Also note that the maximum length of the "BT Name" can not exceed 25 characters.

Step2 - Execute burning operation: In the "Operation" area of the Download tool main interface, click the "Download" button to perform the download operation.

4.3.2 EQ Tool

4.3.2.1 Function Description (EQ Tool)

The main functions of this tool are as follows:

- Generate EQ parameter
- Perform online EQ parameter tuning (requires support from relevant protocol)

4.3.2.2 Interface Composition

The interface of the EQ tool consists of 5 areas, including the graph display area, the "Configure" area, the "Parameter" area, the log output area and the "Operation" area, as shown in the figure below.

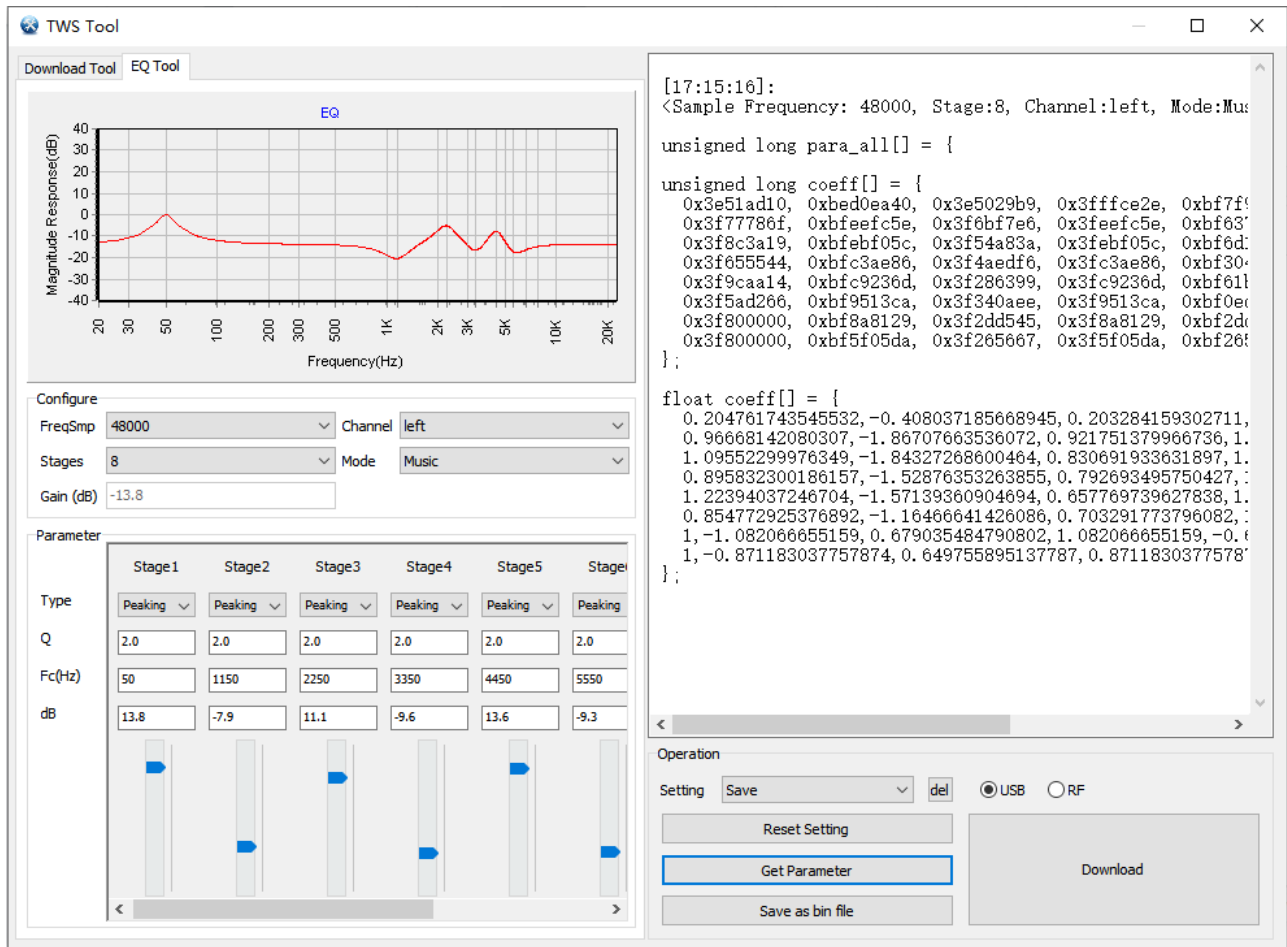


Figure 4.8: TWS_Tool/EQ_Tool interface

4.3.2.3 Instructions for Use

Step1 - EQ parameter adjustment: First select the mode of parameter tuning, and select it through the Mode in the "configure" area, the tool provides three optional modes by default, namely: "Music" mode (i.e. Line Out), "Speech MIC" mode (i.e. Line In) and "Speech SPK" mode (i.e. Line In to Line Out). Then select the channel, and select it through the Channel in the "configure" area. Next, select the sample frequency and the level of EQ parameter tuning, respectively, through the FreqSmp and Stages in the "configure" area. Finally, all levels of parameters, including filter type, Q value, center frequency and gain value, are configured by Type, Q, Fc and dB in the "parameter" area.

Step2 - View and save EQ parameter values: The parameter value can be obtained by clicking the "Get Parameter" button in the "Operation" area, and the generated parameter value can be saved as a bin file by clicking the "Save as bin file" button.

Step3 - Save EQ configuration: The EQ configuration can be saved and restored through the checkbox corresponding to Setting in the "Operation" area. For the save operation, first select "Save" in the checkbox, then enter the configuration name in the pop-up window, and finally click OK. For restoring a saved configuration, select the name of the saved configuration via the checkbox to restore the configuration.

Step4 - EQ online tuning: First select the method of online parameter tuning, and select the online param-

eter tuning method through the check box controls ("USB" and "RF") in the "operation" area. Then download the EQ parameters to the device by clicking the "Download" button in the "Operation" area.

Note:

All online parameter tuning methods supported by EQ tool need to be supported by relevant protocols, and you need to consult relevant staff for details.

4.4 RISC-V Tool

The RISC-V tool supports the chip to realize the following functions through USB:

- Read/write register
- Read/write/erase flash
- Report debugging information
- VCD function

Note:

The SDK must have a corresponding interaction protocol before this tool can be used.

4.4.1 Read Memory

Read data from a specific storage space (RAM/Analog/Flash), the steps are as follows.

Step1 - Select the storage space type: In the "Memory" area of the interface, click on the drop-down menu to select the type.

Step2 - Set the access area: Enter the starting address of the space to be accessed in the "Addr" input box, and enter the number of required access addresses in the "Size" input box.

Step3 - Read data: Hold the cursor in the "Addr" box, and the user can read data from the specified storage space by pressing the "Tab" key.

Note:

The data in the "Size" input box is hexadecimal. Writing "10" represents reading 16 bytes of data. A maximum of 256 bytes of data can be read at a time.

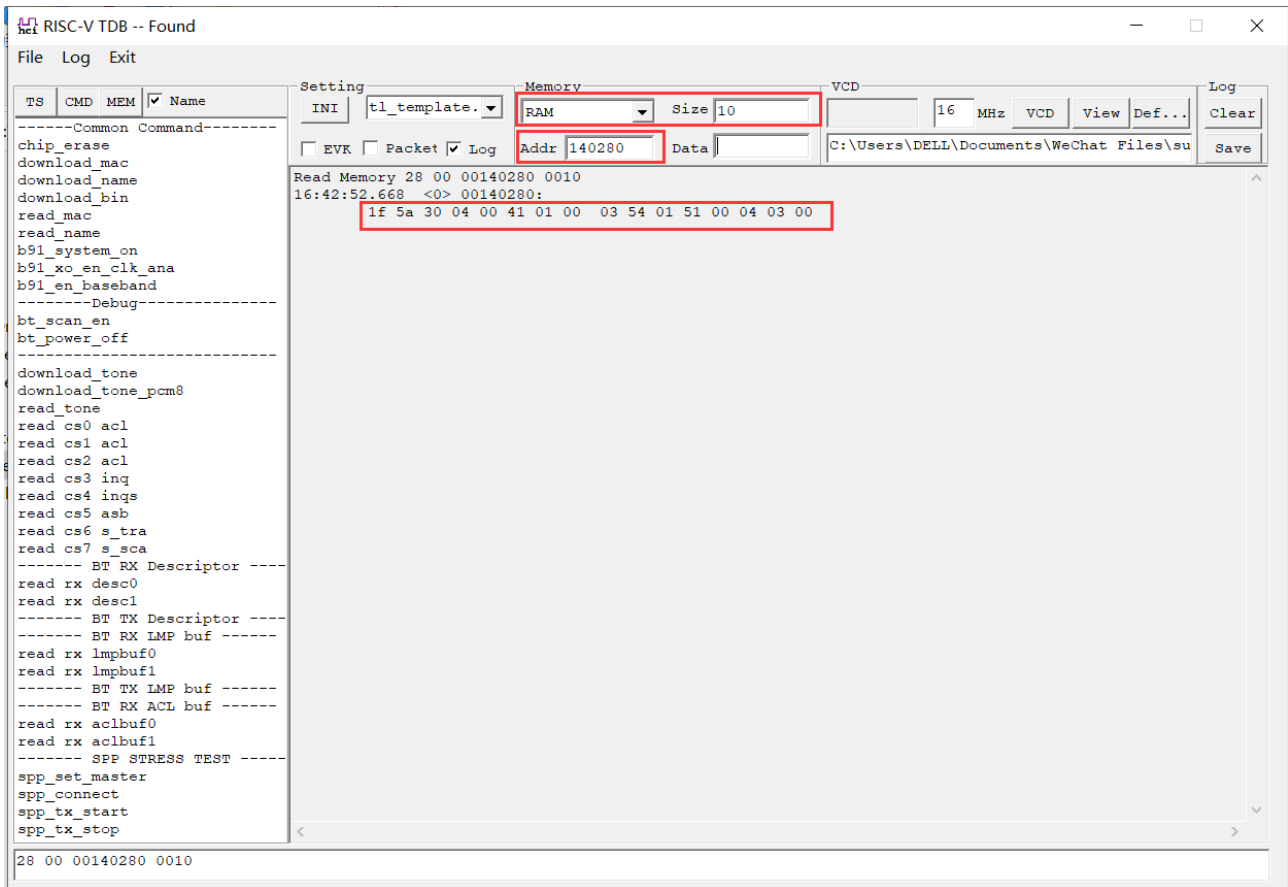


Figure 4.9: Read memory

4.4.2 Write Memory

Write data in a specific storage space (RAM/Analog/Flash), the steps are as follows.

Step1 - Select the storage space type: In the "Memory" area of the interface, click on the drop-down menu to select the type.

Step2 - Set the access area: Enter the starting address of the space to be accessed in the "Addr" input box, and input the length of the data to be written in the "Size" input box.

Step3 - Write data: Enter the target data to be written in the "Data" box. and the user can write the specified data to the specified storage space by pressing the "Enter" key.

Note:

Write data can only write up to four bytes of data.

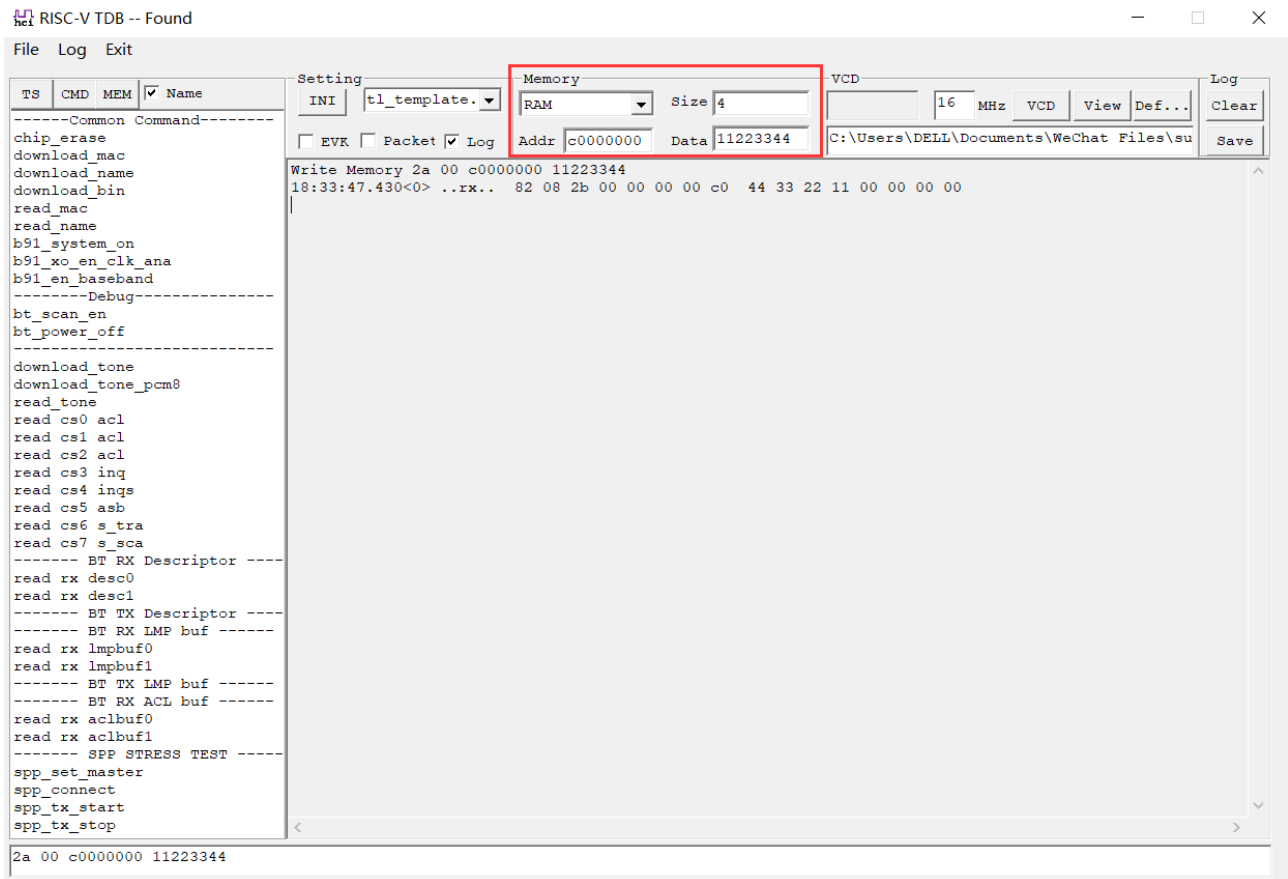


Figure 4.10: Write memory

4.4.3 Download Firmware

Download the firmware to flash, the steps are as follows.

Step1 - Select firmware: Click "File->Import bin" and select the firmware to download.

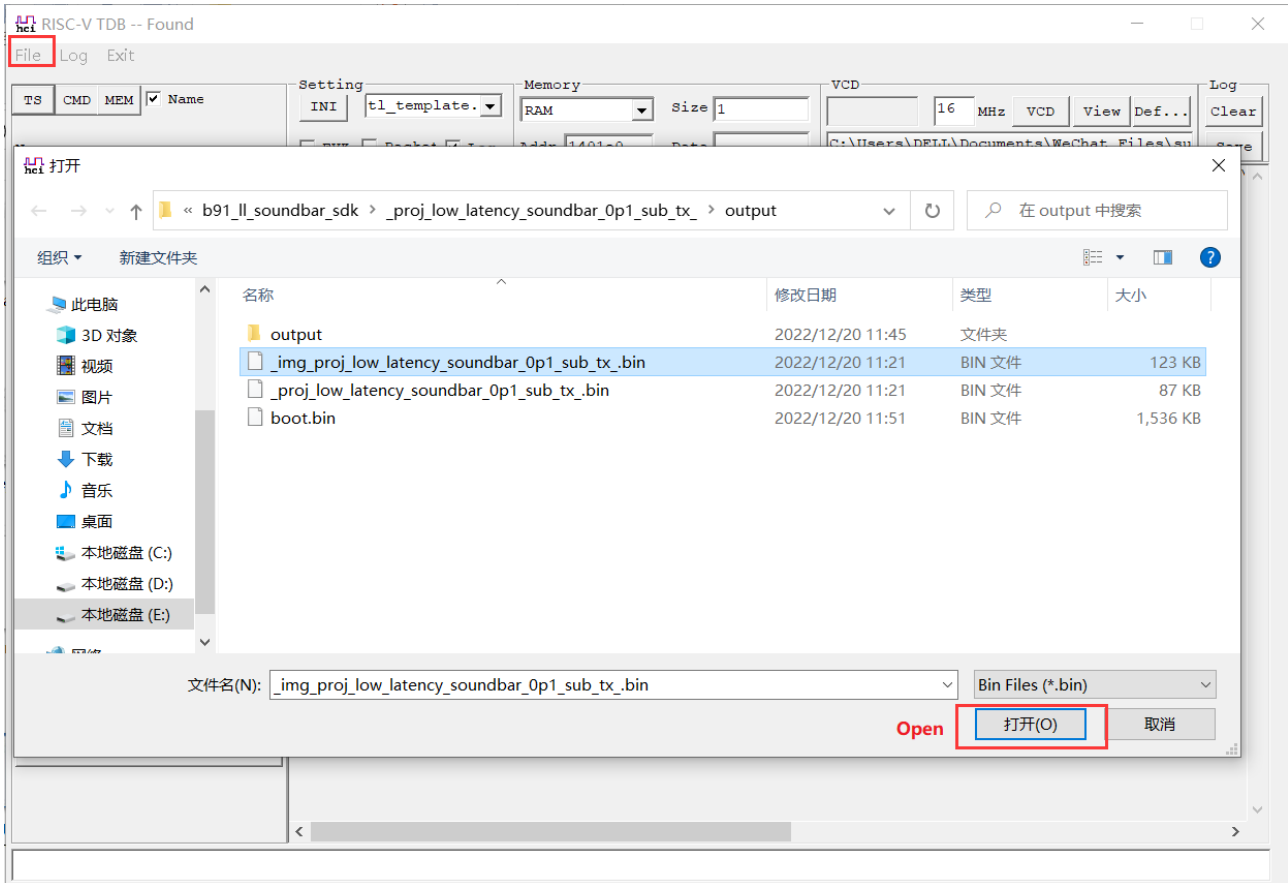


Figure 4.11: Select firmware

Step2 - Erase flash: Click the "TS" button, and then click the "Chip_Erase" button to erase all flash data.

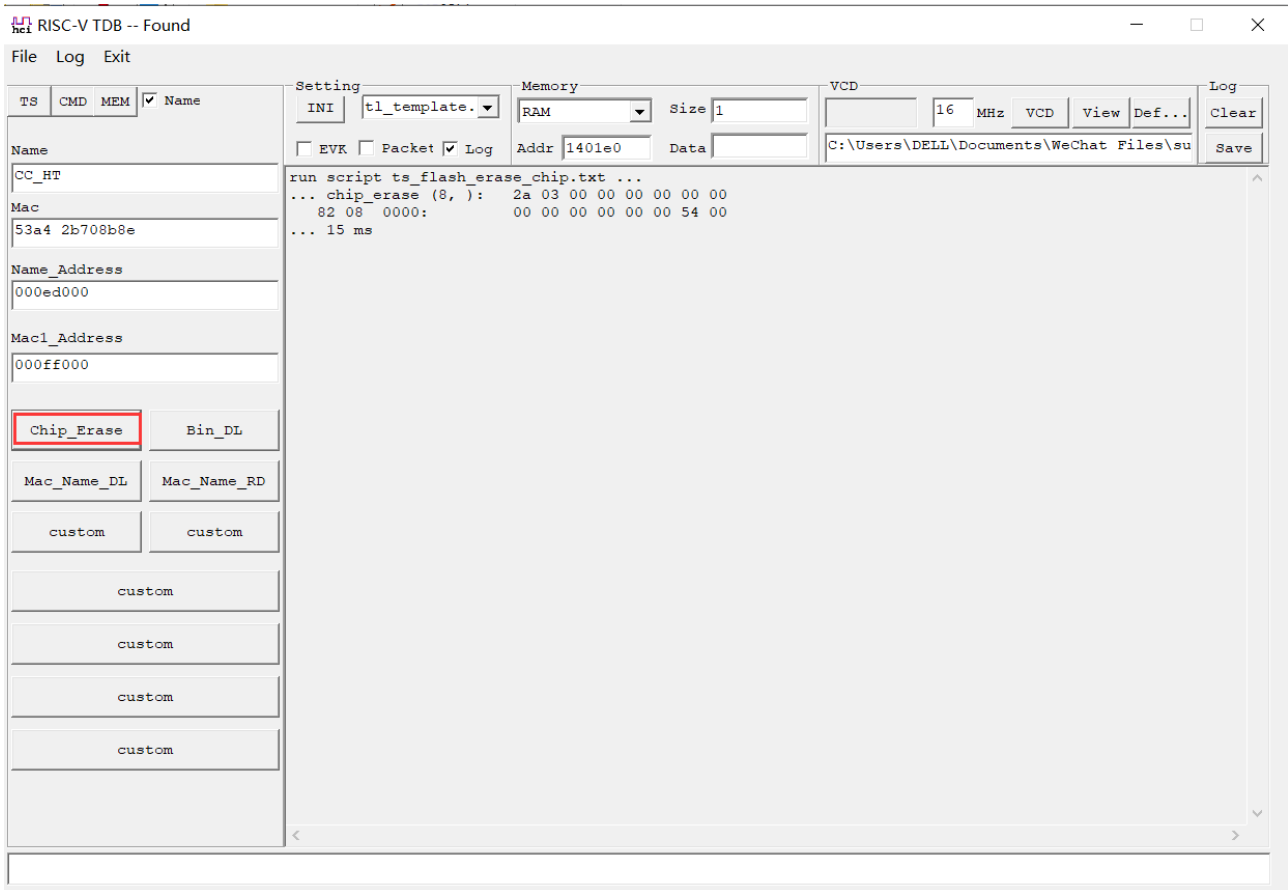


Figure 4.12: Erase flash

Step3 - Download firmware: Click the "Bin_DL" button to download the firmware.

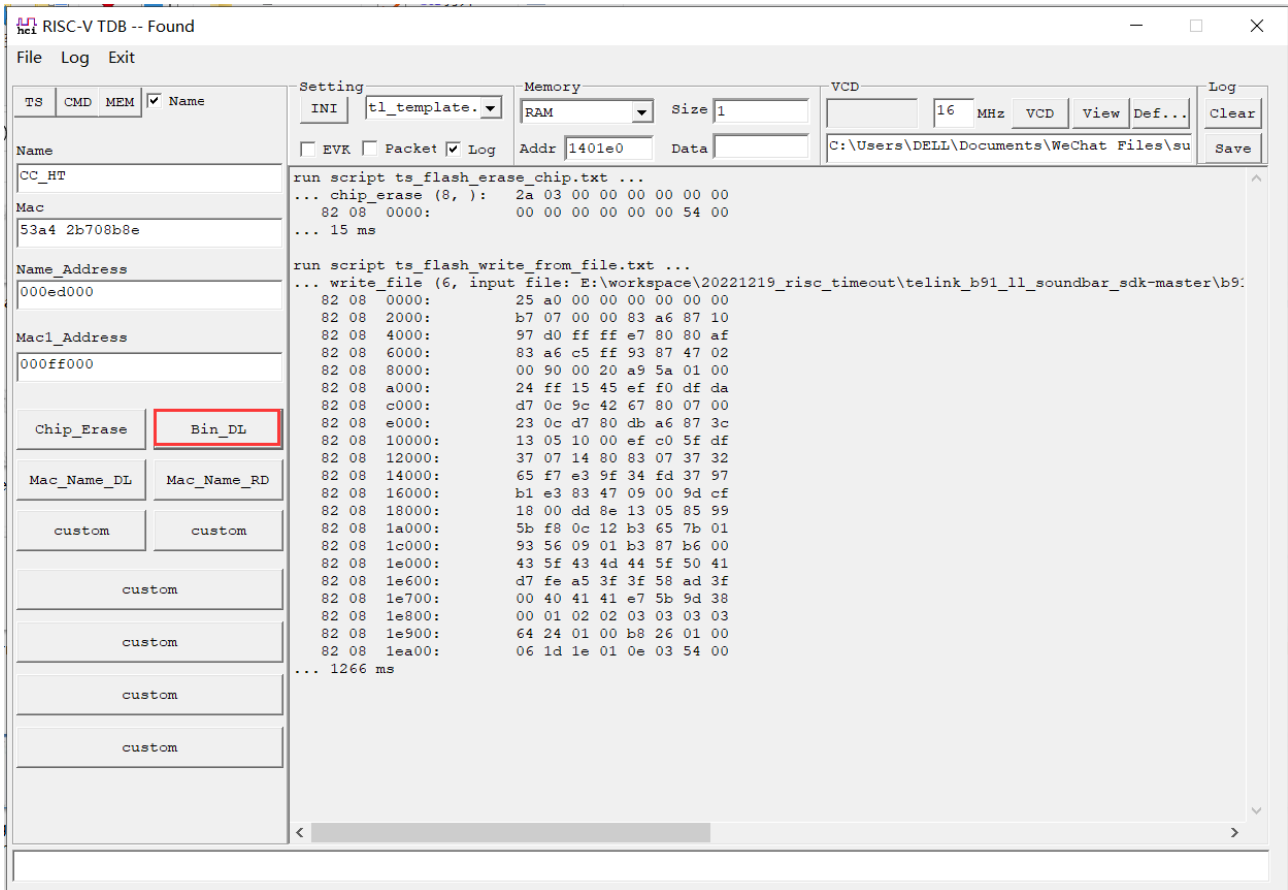


Figure 4.13: Download firmware

4.4.4 VCD Function

VCD function operation steps are as follows.

Step1 - Select the header file: Click the "Def" button in the VCD function area and select the corresponding header file.

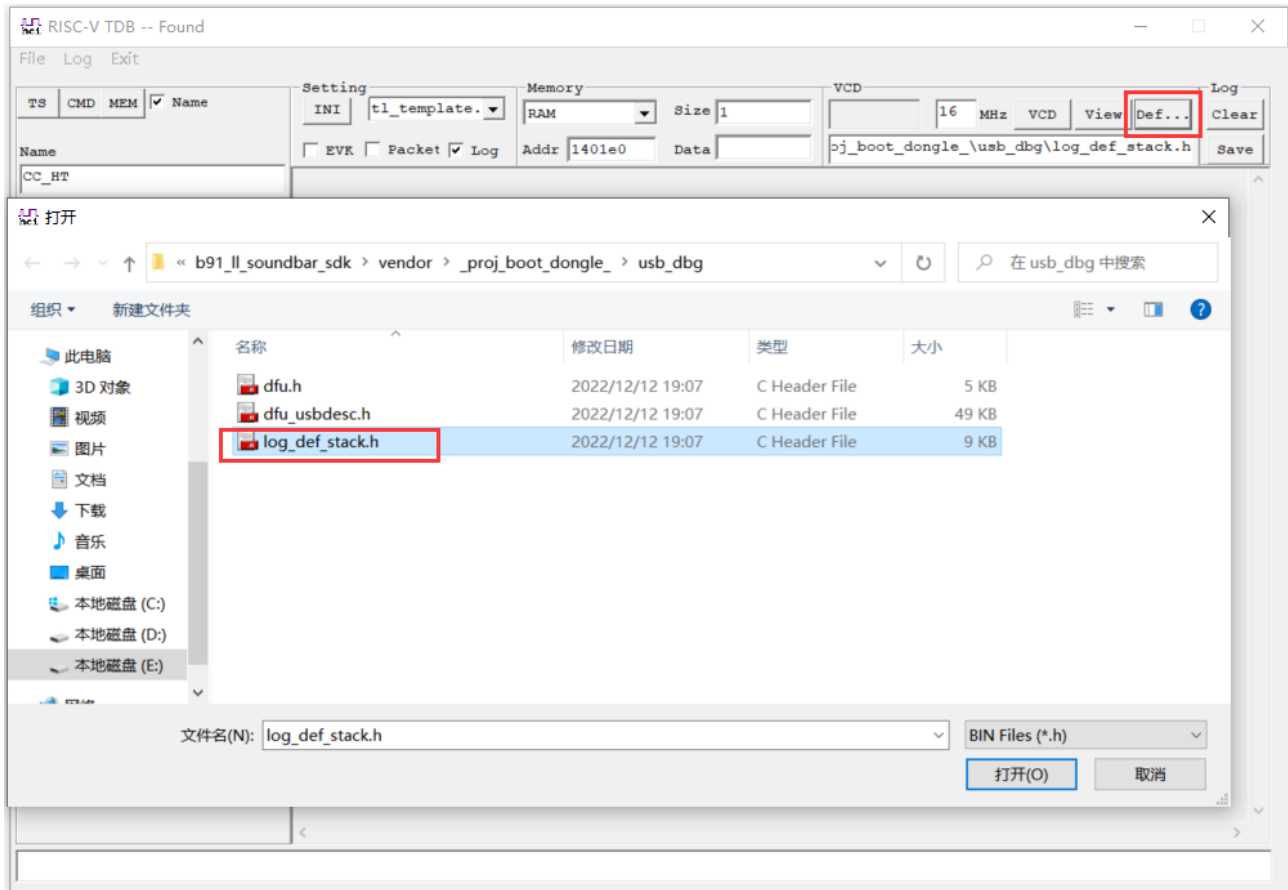


Figure 4.14: Select the header file

Step2 - Start the VCD function: Click the "VCD" button to start.

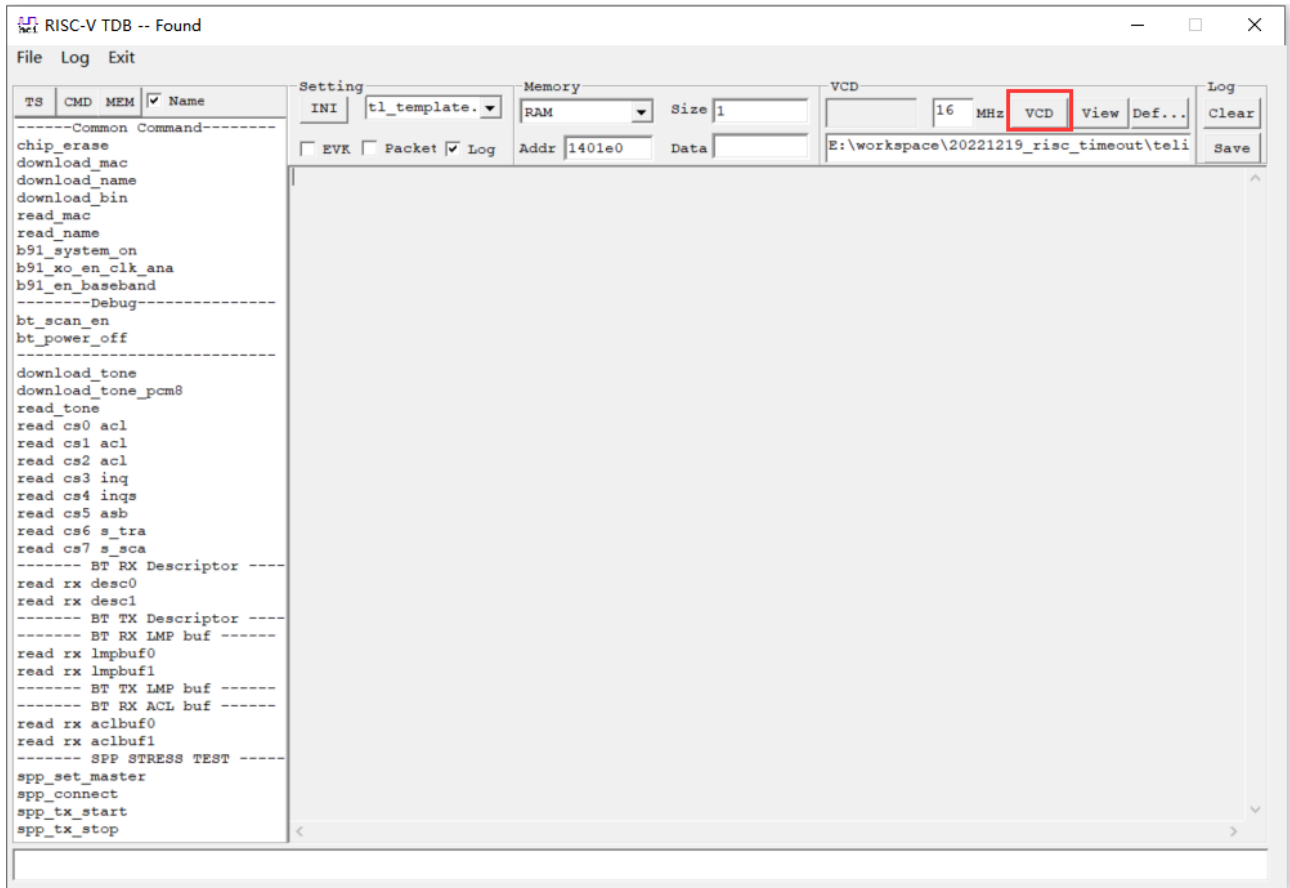


Figure 4.15: Start the VCD function

Step3 - View the running status of related tasks: Click the “View” button to view the task running status.

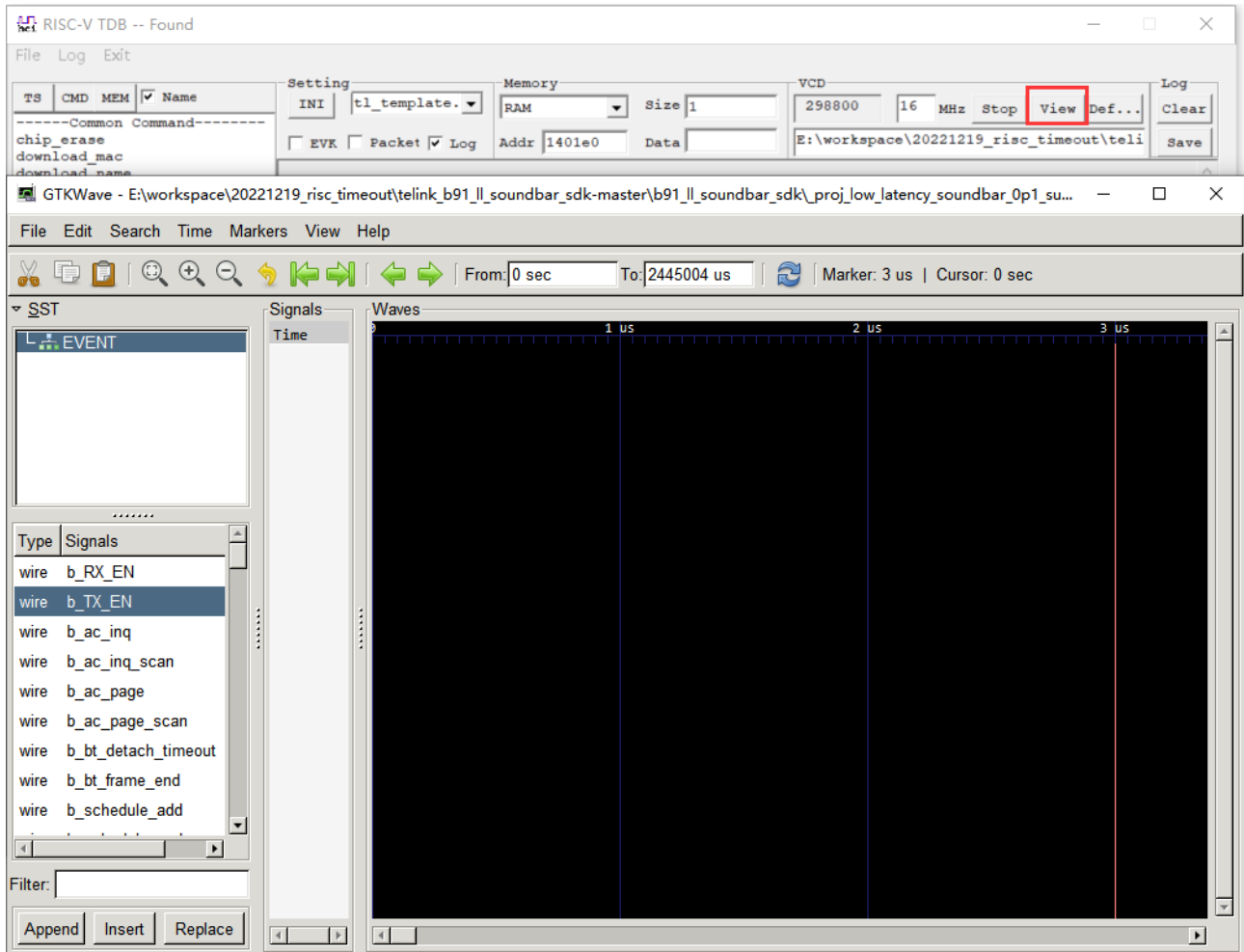


Figure 4.16: View status

Note:

Please consult the SDK responsible person for the definition and use of the header file.

4.4.5 Command Line Description

To facilitate debugging, you can enter the corresponding command to execute the corresponding operation.

4.4.5.1 Read Command

Command format: [28] [Type] [Address] [Size]

Type: access space type, "00" represents access to digital register; "01" represents access to analog register; "02" represents access to flash.

Address: the address of the access space, accounting for 4 bytes.

Size: the number of access addresses, accounting for 4 bytes.

Example: Read a byte of data from the address of digital register 0x140280.

Command: 28 00 00140280 000001

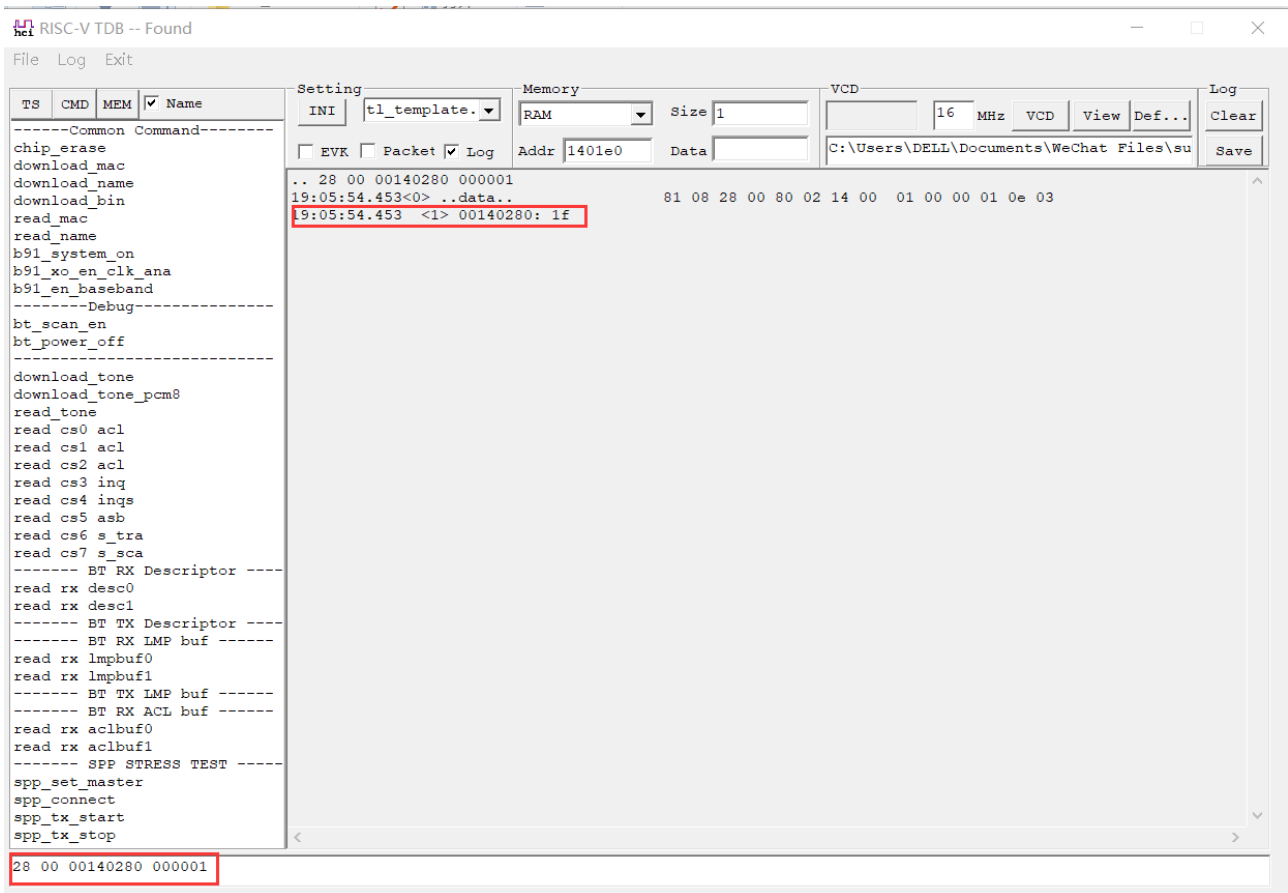


Figure 4.17: Read command

4.4.5.2 Write Command

Command format: [2a] [Type] [Address] [Data1] [Data2].....[DataN]

Type: access space type, "00" represents access to digital register; "01" represents access to analog register; "02" represents access to flash.

Address: the address of the access space, accounting for 4 bytes.

DataN: data to be written.

Example: the flash 0x1000 address writes data "0x1122334455".

Command: 2a 02 00001000 55 44 33 22 11

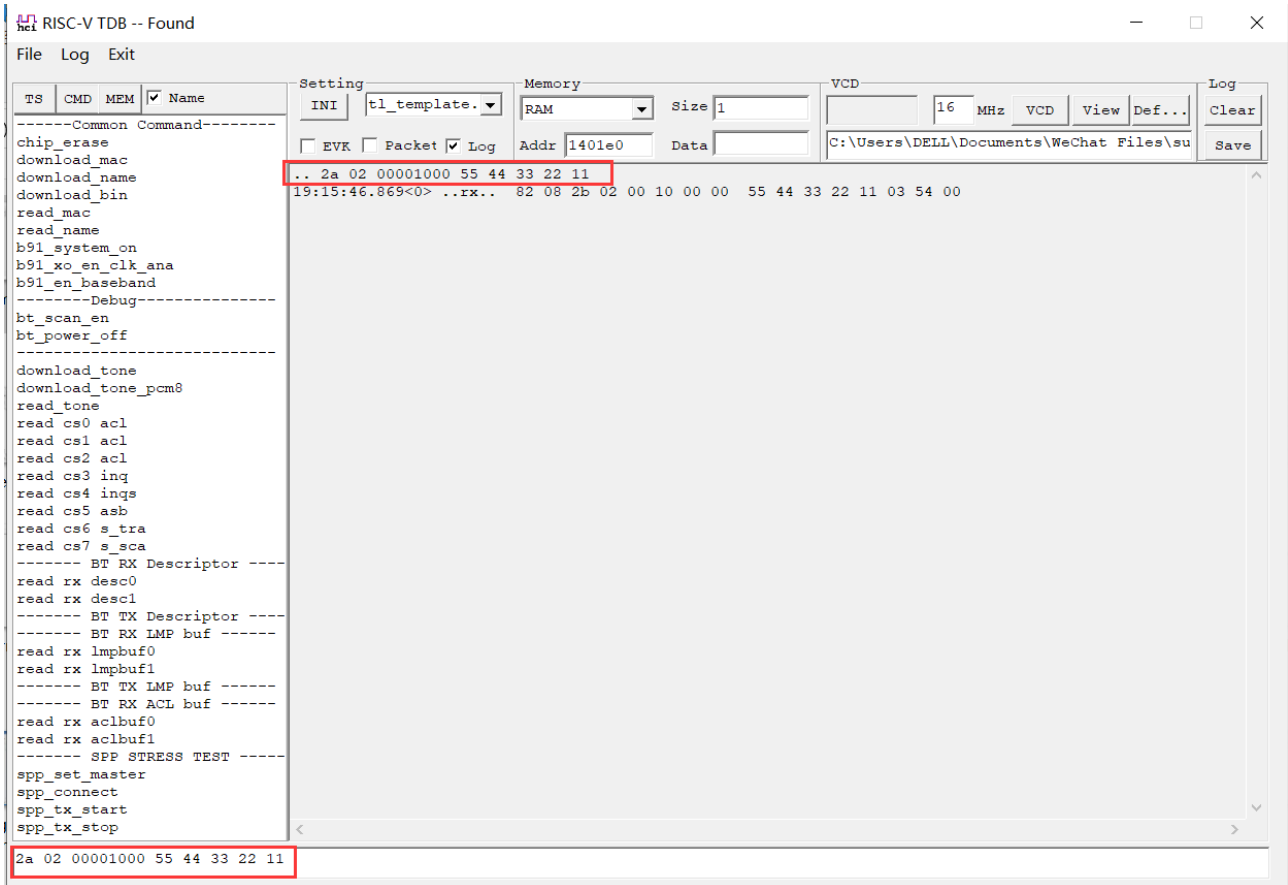


Figure 4.18: Write command

4.4.5.3 Flash Erase Command

Command format: [2a] [03] [Address] [Size]

Address: the address of the access space, accounting for 4 bytes.

Size: the size of the flash space erased, accounting for 4 bytes.

Example: flash 0x1000 address erase 4k data.

Command: 2a 03 00001000 00001000

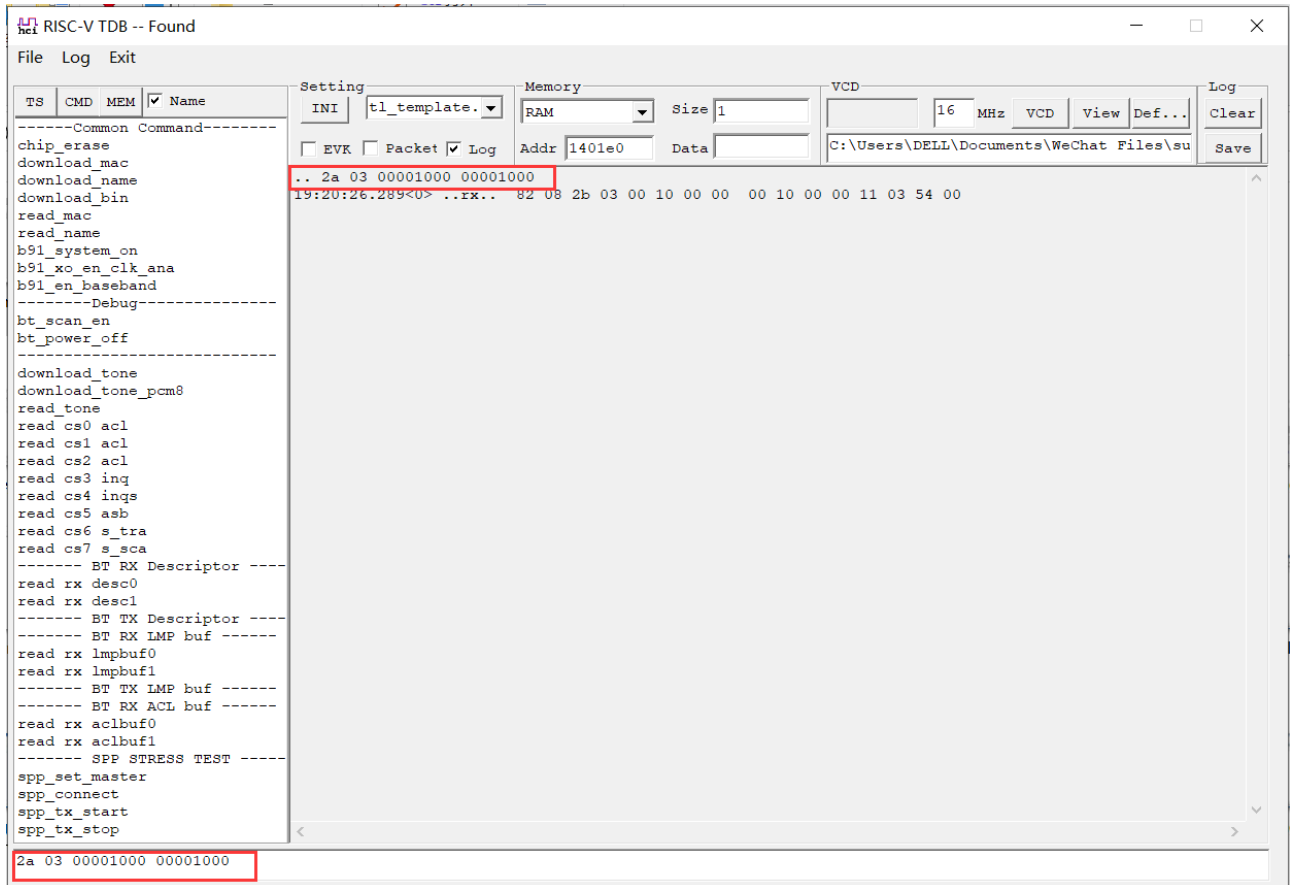


Figure 4.19: Flash erase command

Example: erase the whole flash.

Command: 2a 03 00000000 00000000

4.4.5.4 Save the Read Data

Command format: [r] [String] [28] [Type] [Address] [Size] [BinPath]

r: read command.

String: string, which can be customized.

Type: access space type, "00" represents access to digital register; "01" represents access to analog register; "02" represents access to flash.

Address: the address of the access space, accounting for 4 bytes.

Size: the size of the data to be read, accounting for 4 bytes.

BinPath: the path where the read data is to be saved.

Example: read 4k data from flash 0 address and save it to D drive.

Command: r read 0228 00000000 00001000 -d:\read.bin

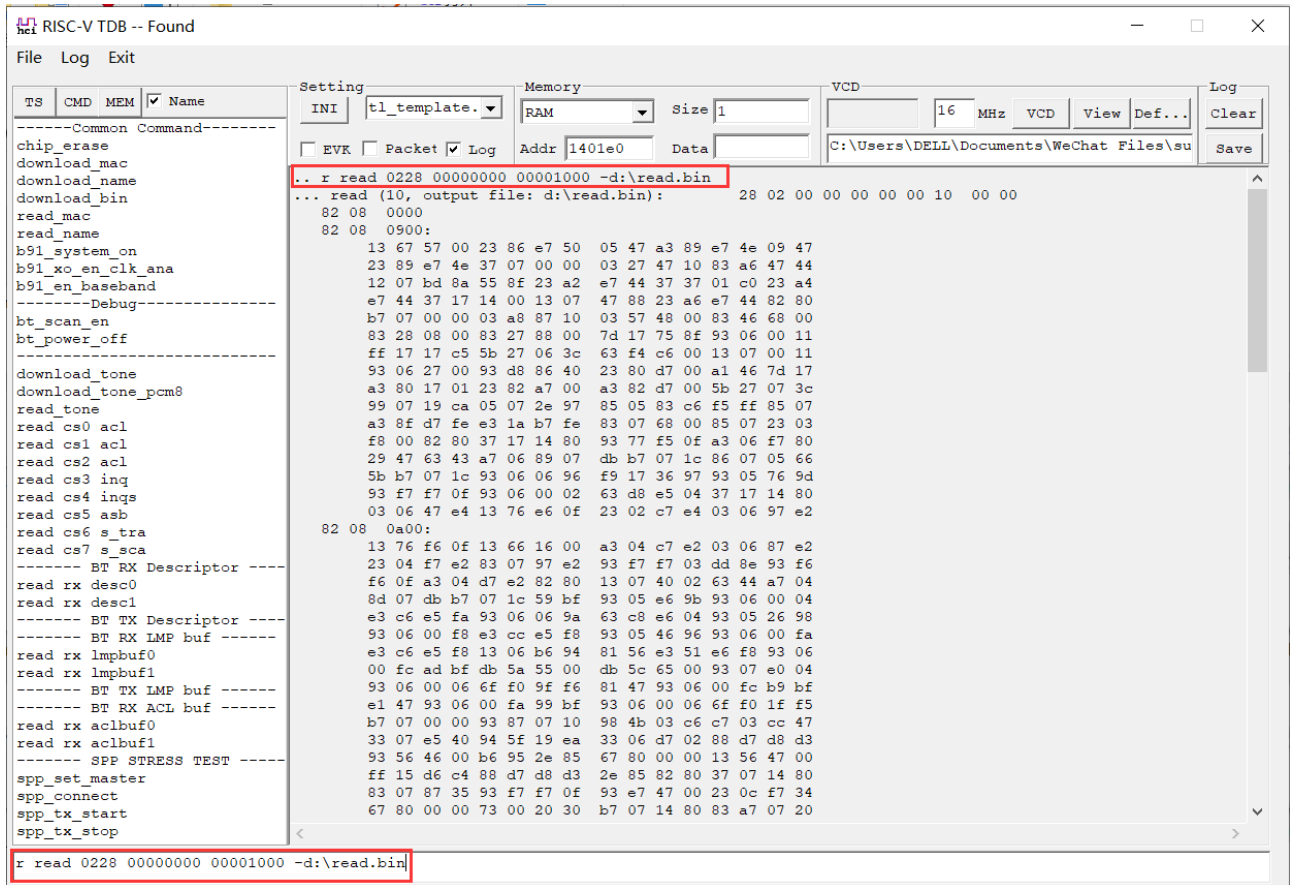


Figure 4.20: Save the read data

4.4.6 Clear and Save Logs

Clear log: click the "Clear" button on the interface to clear the log.

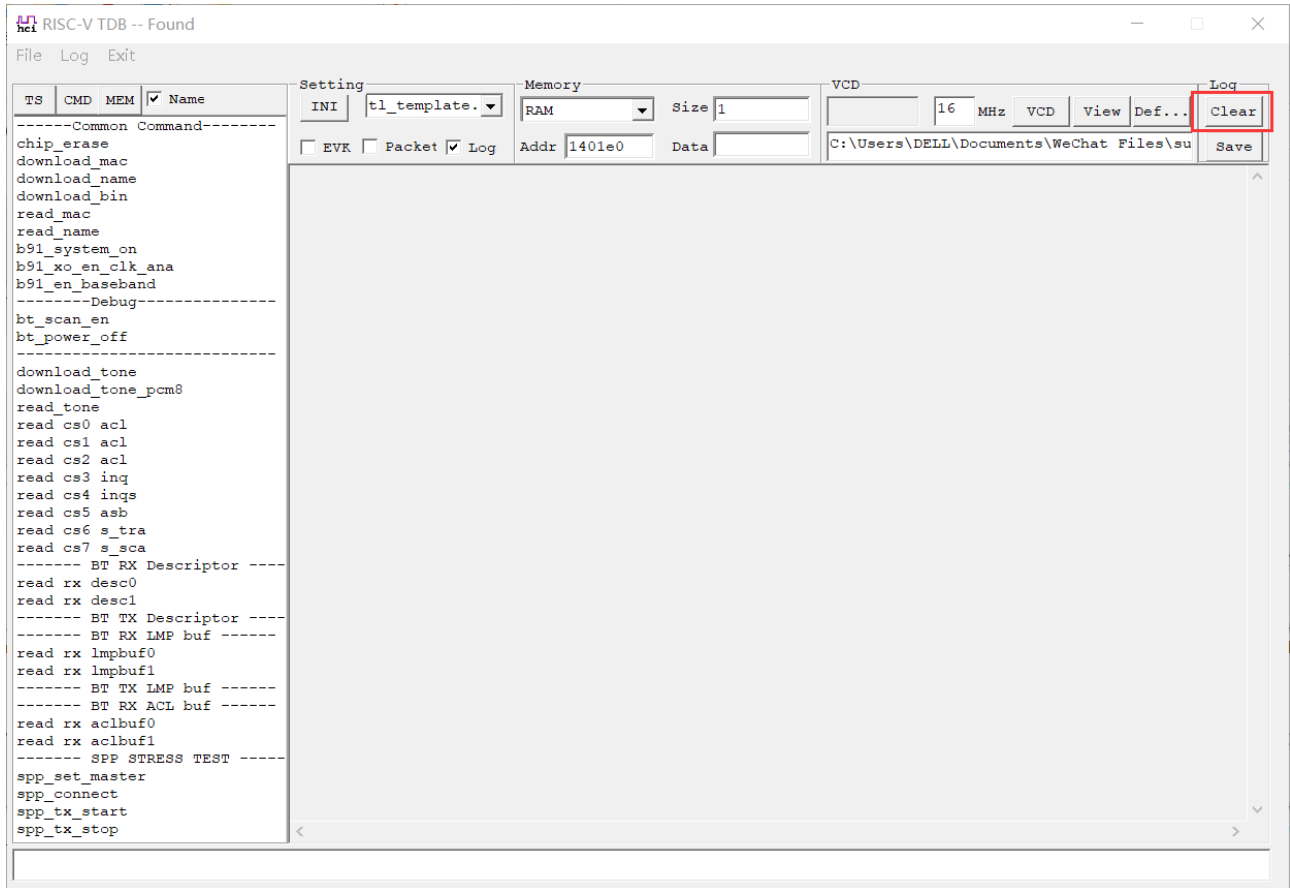


Figure 4.21: Clear log

Save log: click the "Save" button on the interface, select the file saving path and enter the file name.

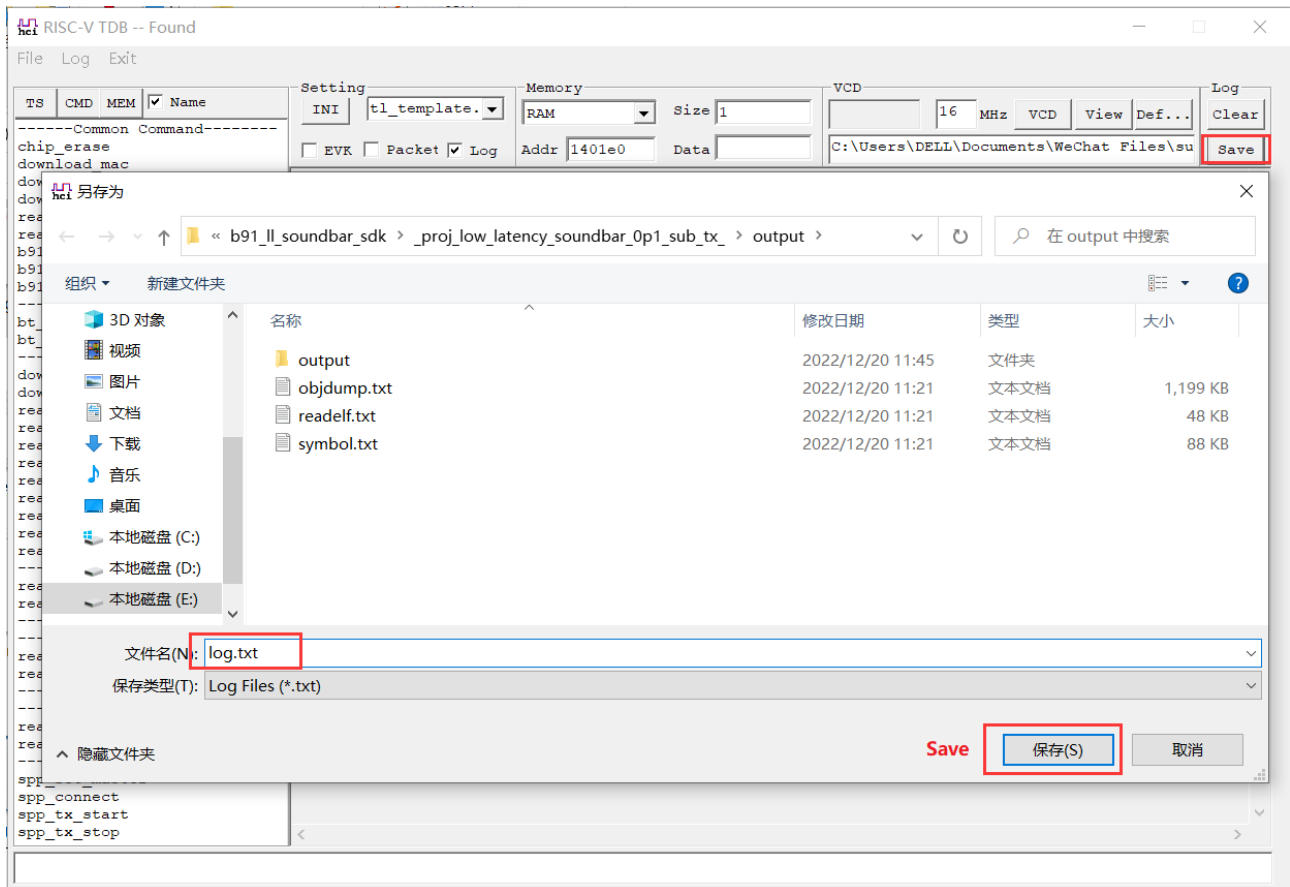


Figure 4.22: Save log

4.5 EMI Tool

4.5.1 Functional description (EMI Tool)

The tool has two main functions:

- Configure and download the test firmware.
- EMI test.
- Non-signaling test.
- RF current test.

4.5.2 Interface Composition and Instructions (BQB Tool)

The tool is mainly composed of four parts: firmware configuration download interface, EMI test interface, non-signaling test interface and RF current test interface, as shown in the figure below.

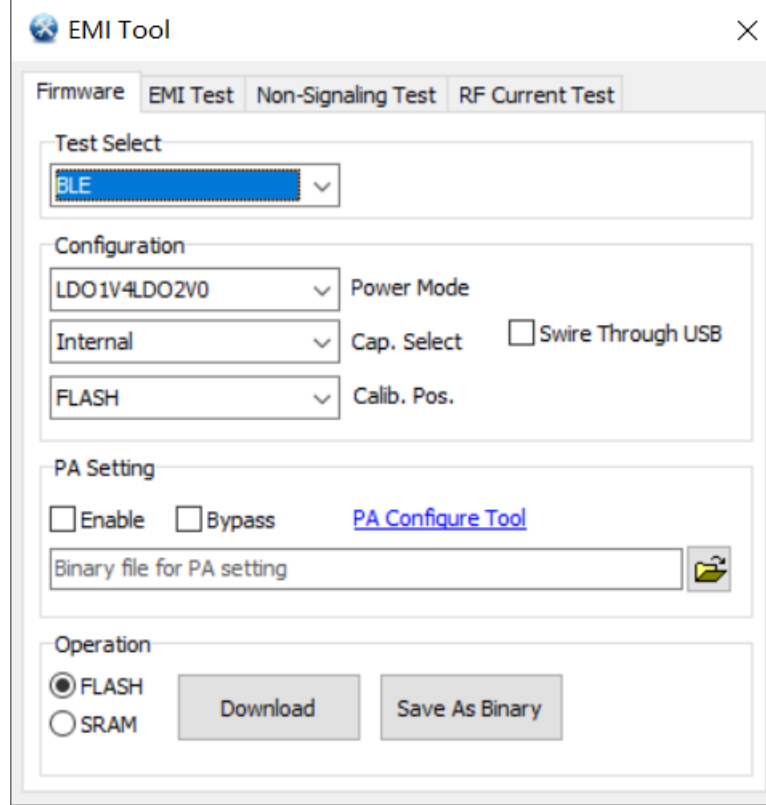


Figure 4.23: Firmware configuration download interface

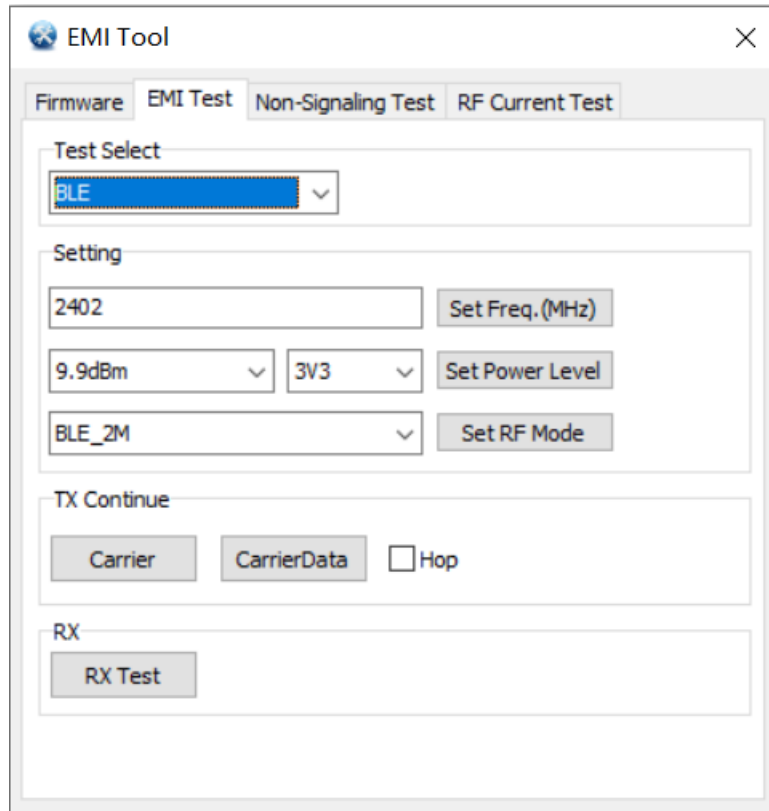


Figure 4.24: EMI test interface

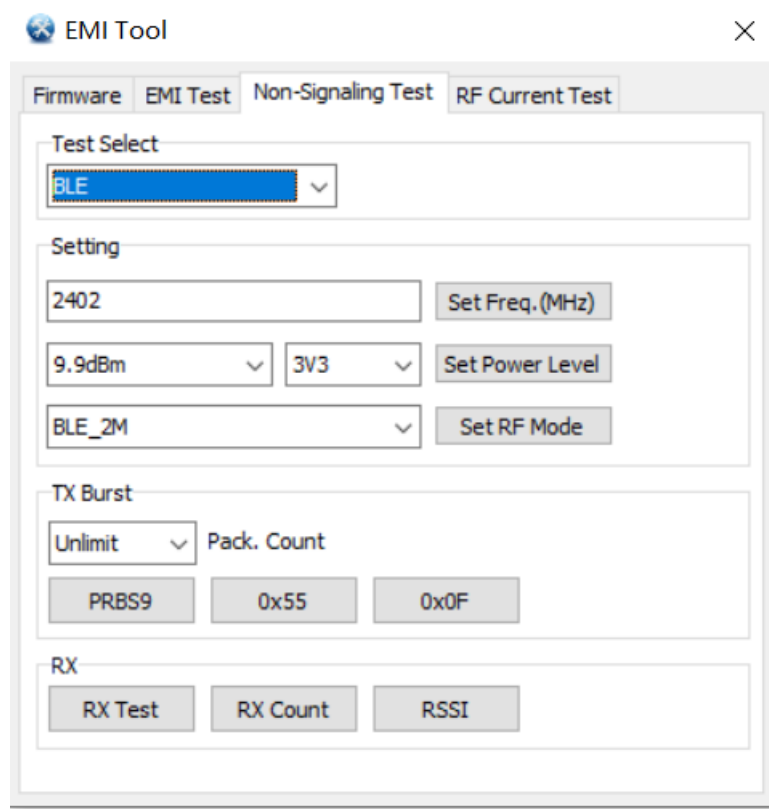


Figure 4.25: Non-signaling test interface

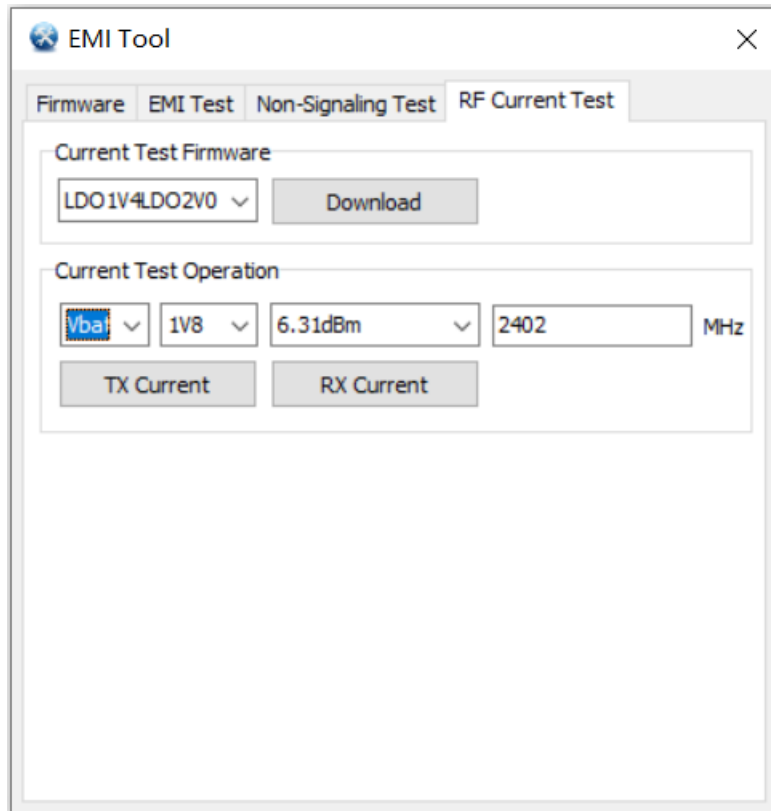


Figure 4.26: RF current test interface

4.5.2.1 Chip Selection

Chip model selection is performed through the MCU list on the BDT main interface.

4.5.2.2 Save and Download Firmware

Step1 - Configuration firmware: In the "Configure" area of the firmware downloading interface, "Cap. Select" is used to configure the internal or external capacitors, "Power Mode" is used to configure the internal power supply mode of the chip, and "Calib. Pos." is used to configure the storage location of calibration data (selected according to actual situation). If the "Swire Through USB" function is used, the corresponding option should be ticked.

Step2 - PA configuration: In the "PA Setting" area of the firmware downloading interface, if you need to use the PA function, tick the box for "Enable". If you want to use the PA function but working in Bypass mode, tick the box for "Bypass". In addition to the above operations, users also need to enter the binary file path for PA operations into the corresponding text box. The PA-related binary files can be generated using the "PA Configure Tool". You can click "PA Configure Tool" to open the tool, and refer to the corresponding chapter for the use of the tool.

Step3 - Select the firmware type: In the "Operation" area of the firmware downloading interface, select the type through a radio box. The available types include "Flash" and "SRAM".

Step4 - Save/download the firmware: In the "Operation" area of the firmware downloading interface, save

the firmware by clicking the "Save As Binary" button, and download the firmware by clicking the "Download" button.

4.5.2.3 EMI Test

Step1 - Configure frequency, power, and RF mode: In the "Setting" area of the EMI test interface, input the test frequency, select the TX power (required for TX testing) and select the RF mode.

Step2 - TX test: The TX test includes two types of RF signals, namely single carrier signal and continuous modulated signal. They can be configured separately using the "Carrier" and "CarrierData" buttons in the "TX Continue" area. Additionally, the continuous modulated signal supports frequency hopping mode, which can be set by ticking the "Hop" box.

Step3 - RX test: The "RX Test" button in the "RX" area controls the DUT into RX test mode.

4.5.2.4 Non-signaling Test

Step1 - Configure frequency, power, and RF mode: In the "Setting" area of the non-signaling test interface, input the test frequency, select the TX power (required for TX testing) and select the RF mode.

Step2 - TX test: In the "TX Burst" area of the non-signaling test interface, select the number of consecutive packets to be sent by using "Pack. Count" (only two modes of sending 1000 packets per time or an unlimited number of packets are supported). Additionally, you can select to send different types of data packets by clicking on "PRBS9", "0x55", or "0x0F" respectively.

Step3 - RX test: The "RX Test" button in the "RX" area controls the DUT into RX test mode, clicking on the "RX Count" button to view the current number of received packets, and clicking on the "RSSI" button to view the power of the RF signal source.

4.5.2.5 RF Current Test

Step1 - Download the firmware required for RF current test: In the "Current Test Firmware" area of the RF current test interface, configure the internal power supply of the MCU and click on the "Download" button to download the firmware for the current test (the DUT needs to be powered down and on at the beginning of each time).

Step2 - TX current test: In the "Current Test Operation" area of the RF current test interface, configure the power supply method, power and frequency point, and enter the TX current test by clicking on the "TX Current" button.

Step3 - RX current test: In the "Current Test Operation" area of the RF current test interface, configure the frequency point and enter the RX current test by clicking on the "RX Current" button.

4.6 PA Configure Tool

4.6.1 Function Description (PA Configure Tool)

This tool is mainly used to configure the binary files required to generate the PA.

4.6.2 Interface Composition and Instructions (PA Configure Tool)

The interface of the tool is shown below.

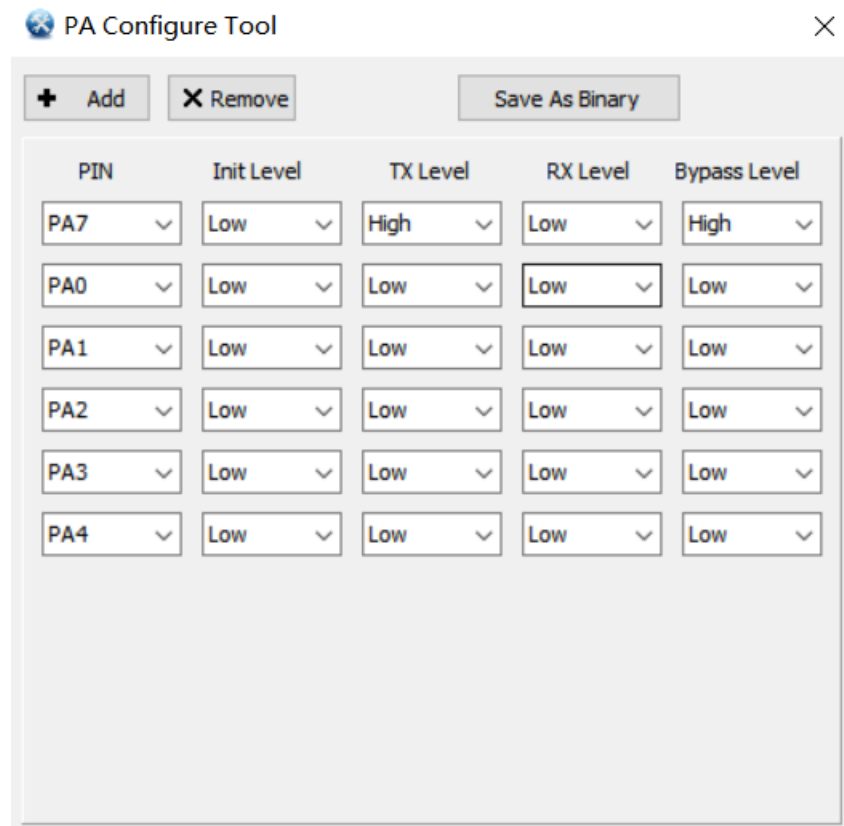


Figure 4.27: "PA Configure Tool" interface

4.6.2.1 Configure the File that Generates the PA

Step1 - Add PA-controlled pin: Add the pin required for the PA to work by clicking on the "Add" button.

Step2 - Configure the level of each pin in the different operating modes: Configure the level of each pin in "Init", "TX", "RX" and "Bypass" modes respectively.

Step3 - Generate binary files: The configuration can be saved as a binary file by clicking on the "Save As Binary" button.

4.7 EVK Offline Tool

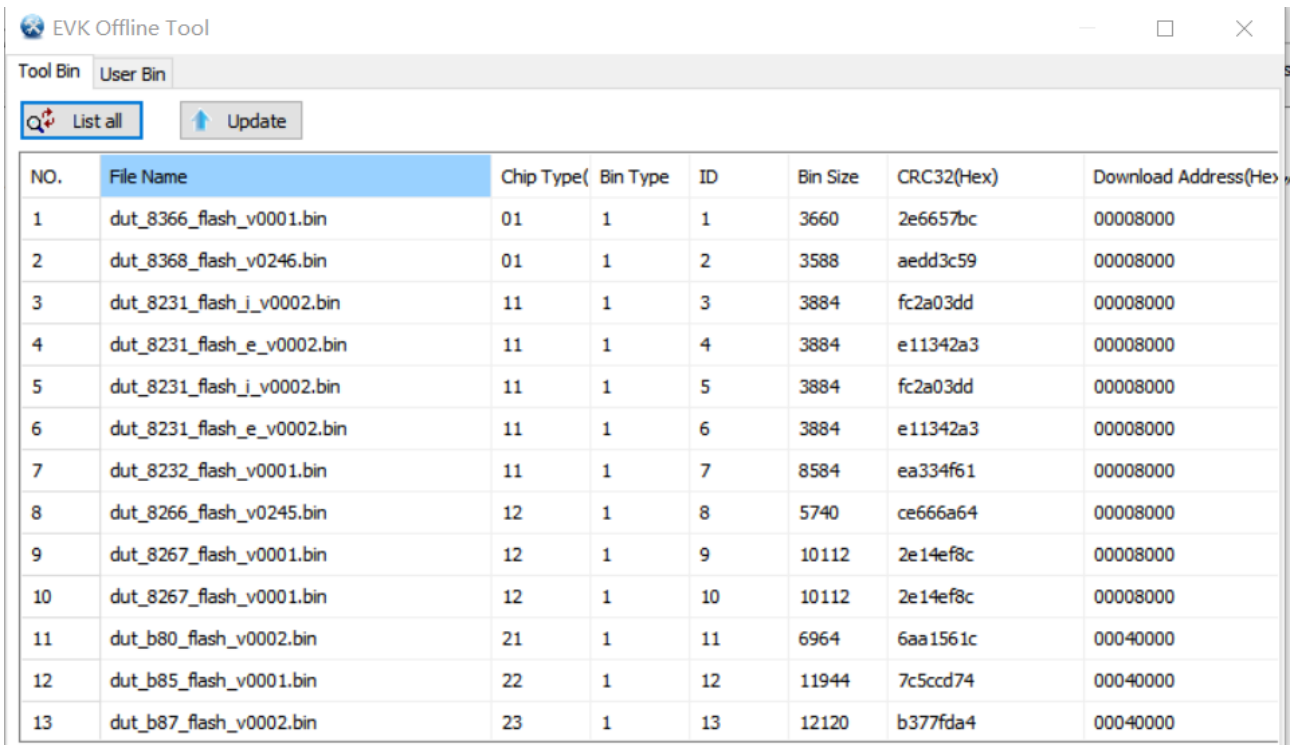
4.7.1 Function Description (EVK Offline Tool)

This tool is used in conjunction with BurningEVK with offline download function, The main features include:

- Update the tool firmware in the BurningEVK.
- Download the user program into BurningEVK.

4.7.2 Interface Composition and Instructions (EVK Offline Tool)

The interface of the tool includes the tool firmware interface and the user tool interface, as shown below.



The screenshot shows the 'EVK Offline Tool' window. It has two tabs: 'Tool Bin' (selected) and 'User Bin'. Below the tabs are two buttons: 'List all' (with a magnifying glass icon) and 'Update' (with an upward arrow icon). The main area contains a table with 8 columns: NO., File Name, Chip Type, Bin Type, ID, Bin Size, CRC32(Hex), and Download Address(Hex). The table lists 13 files, each with a unique ID and CRC32 value.

NO.	File Name	Chip Type	Bin Type	ID	Bin Size	CRC32(Hex)	Download Address(Hex)
1	dut_8366_flash_v0001.bin	01	1	1	3660	2e6657bc	00008000
2	dut_8368_flash_v0246.bin	01	1	2	3588	aedd3c59	00008000
3	dut_8231_flash_j_v0002.bin	11	1	3	3884	fc2a03dd	00008000
4	dut_8231_flash_e_v0002.bin	11	1	4	3884	e11342a3	00008000
5	dut_8231_flash_j_v0002.bin	11	1	5	3884	fc2a03dd	00008000
6	dut_8231_flash_e_v0002.bin	11	1	6	3884	e11342a3	00008000
7	dut_8232_flash_v0001.bin	11	1	7	8584	ea334f61	00008000
8	dut_8266_flash_v0245.bin	12	1	8	5740	ce666a64	00008000
9	dut_8267_flash_v0001.bin	12	1	9	10112	2e14ef8c	00008000
10	dut_8267_flash_v0001.bin	12	1	10	10112	2e14ef8c	00008000
11	dut_b80_flash_v0002.bin	21	1	11	6964	6aa1561c	00040000
12	dut_b85_flash_v0001.bin	22	1	12	11944	7c5ccd74	00040000
13	dut_b87_flash_v0002.bin	23	1	13	12120	b377fda4	00040000

Figure 4.28: "EVK Offline Tool" tool firmware interface

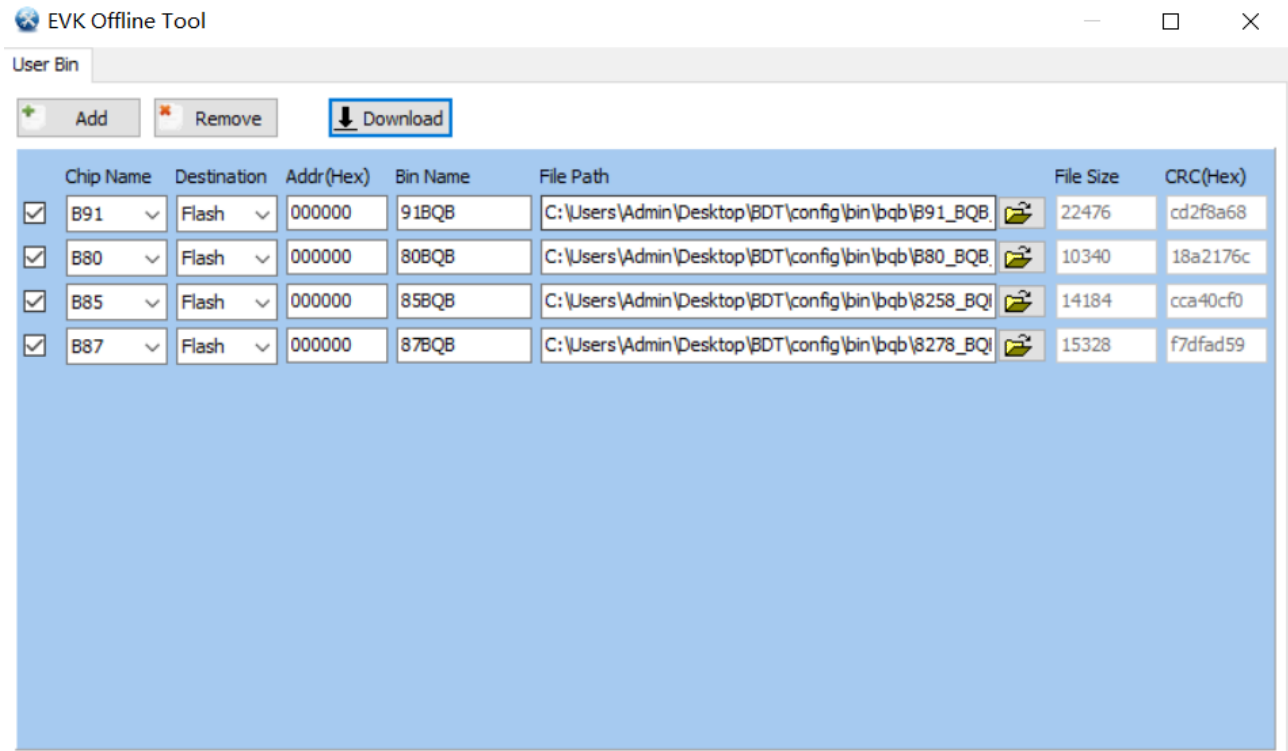


Figure 4.29: “EVK Offline Tool” user firmware interface

4.7.2.1 Update Tool Firmware

Step1 - View the tool firmware in the BDT: Click on the “List all” button to view the existing tool firmware.

Step2 - Update firmware: Update the tool firmware to the BurningEVK by clicking on the “Update” button.

Note:

The tool firmware interface is hidden by default and needs to be displayed by modifying the configuration file.

4.7.2.2 Download User Firmware

Step1 - Add user firmware: Add the user firmware by clicking on the “Add” button and configure the “Chip Name”, “Bin Name”, “Destination”, “Addr” and “File Path” information.

Step2 - Download user firmware: First tick the firmware that needs to be downloaded to BurningEVK and then click on the “Download” button to complete the download.

4.8 Secure Boot Tool

The main functions of this tool are as follows:

- Supports the download of chip configuration information and firmware.

- Supports re-enabling the Debug interface function.
- Supports reading the configuration information of the chip.

4.8.1 Interface and Instructions

The Secure Boot tool interface is shown in the following figure.

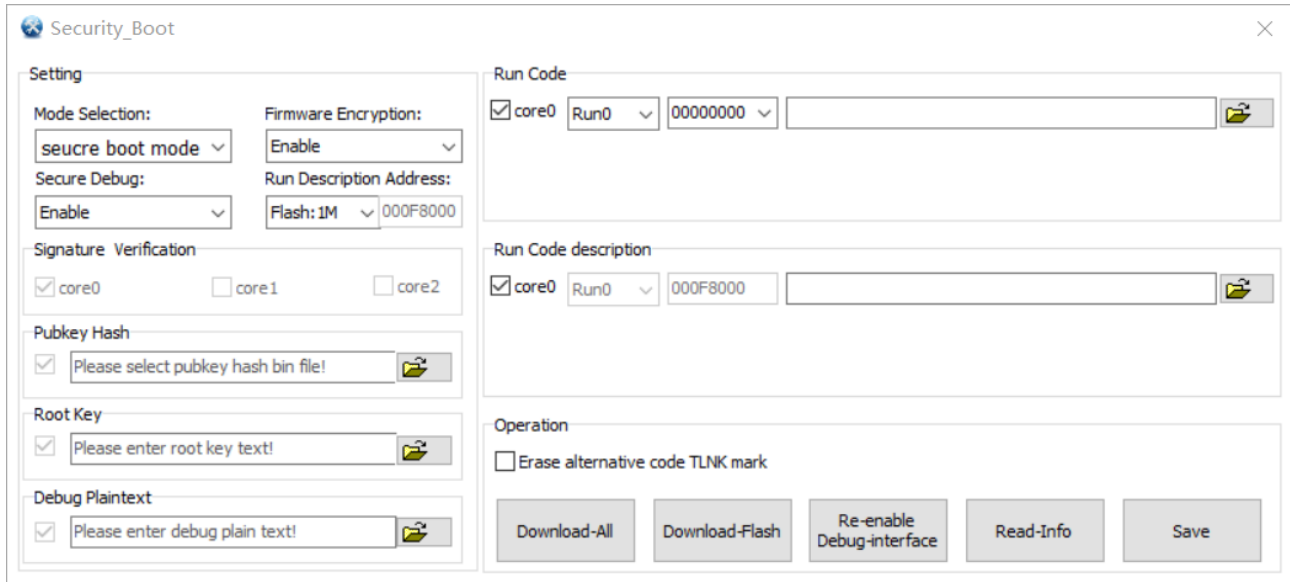


Figure 4.30: Secure Boot tool interface

4.8.1.1 Mode Selection

Select either "secure boot mode" or "normal mode".

4.8.1.2 Firmware Encryption

Select to enable or disable the "Firmware Encryption" function.

4.8.1.3 Secure Debug

Select to enable or disable the "Secure Debug" function.

4.8.1.4 Run Description Address

Select the flash size and the corresponding code description address will be confirmed automatically.

Note:

This function is only available in "secure boot mode".

4.8.1.5 Pubkey Hash

Select the "public_key_hash.bin" file to be burned. After clicking the "Download-All" button, the program will automatically check whether the length of the file size meets the conditions. If it doesn't, the program will report an error.

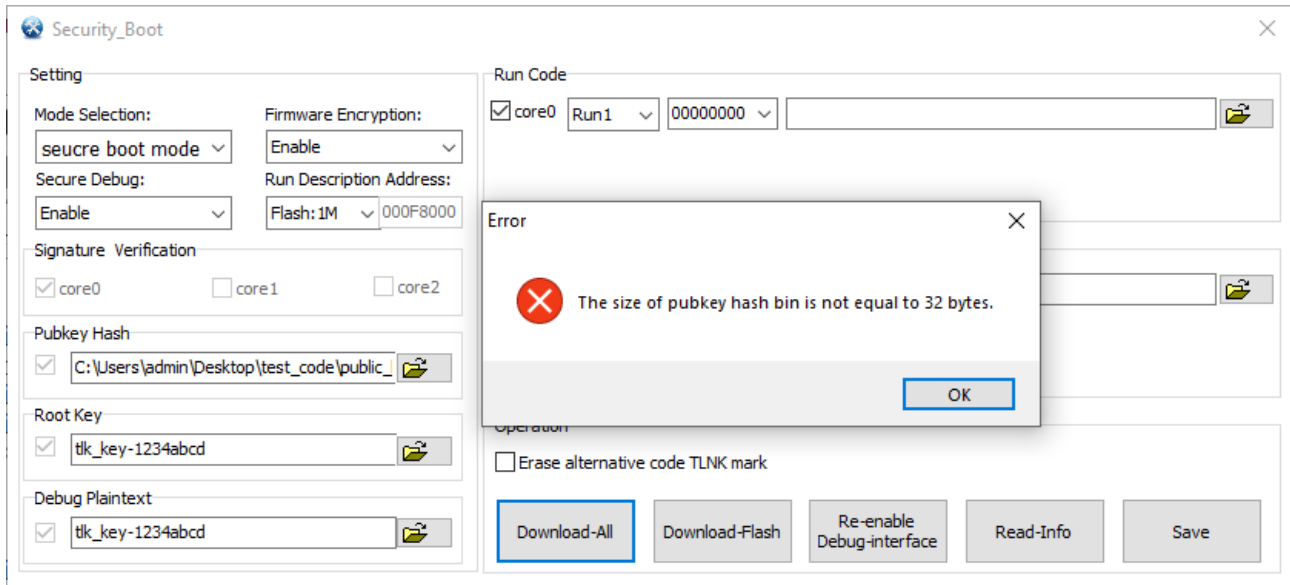


Figure 4.31: Public key hash file error

Note:

This function is only available in "secure boot mode".

4.8.1.6 Root Key

Enter a string of 16 characters or 32 hexadecimal characters as the root key in the input box. It also supports selecting a text file to import the key.

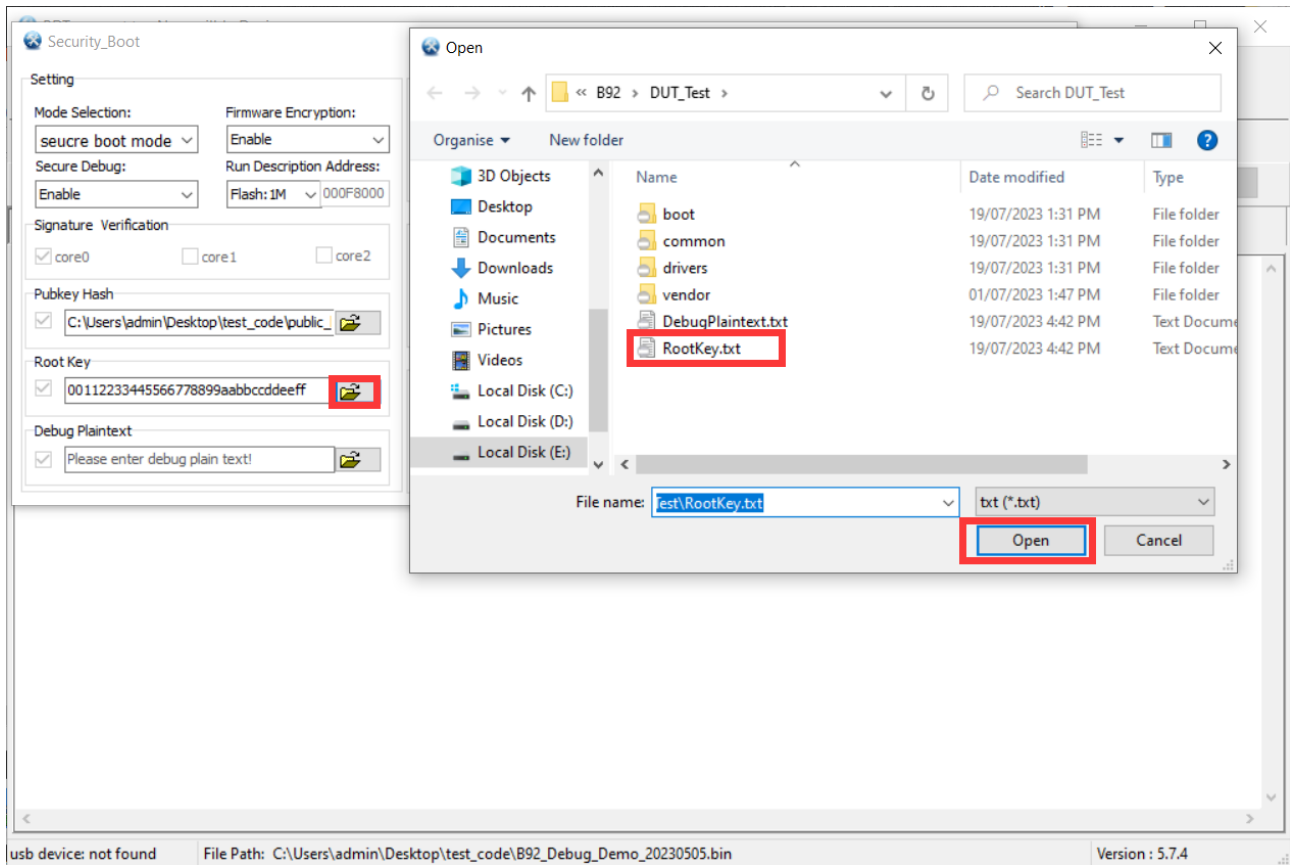


Figure 4.32: Import RootKey.txt file

Enter a string of 32 hexadecimal characters, for example: 00112233445566778899aabbccddeeff or 00112233445566778899AABBCCDDEEFF.

Enter a string of 16 characters, for example: tlk_key-1234abcd.

After clicking the "Download-All" button, the program will automatically check whether the input string meets the conditions. If the conditions are not met, an error will be reported.

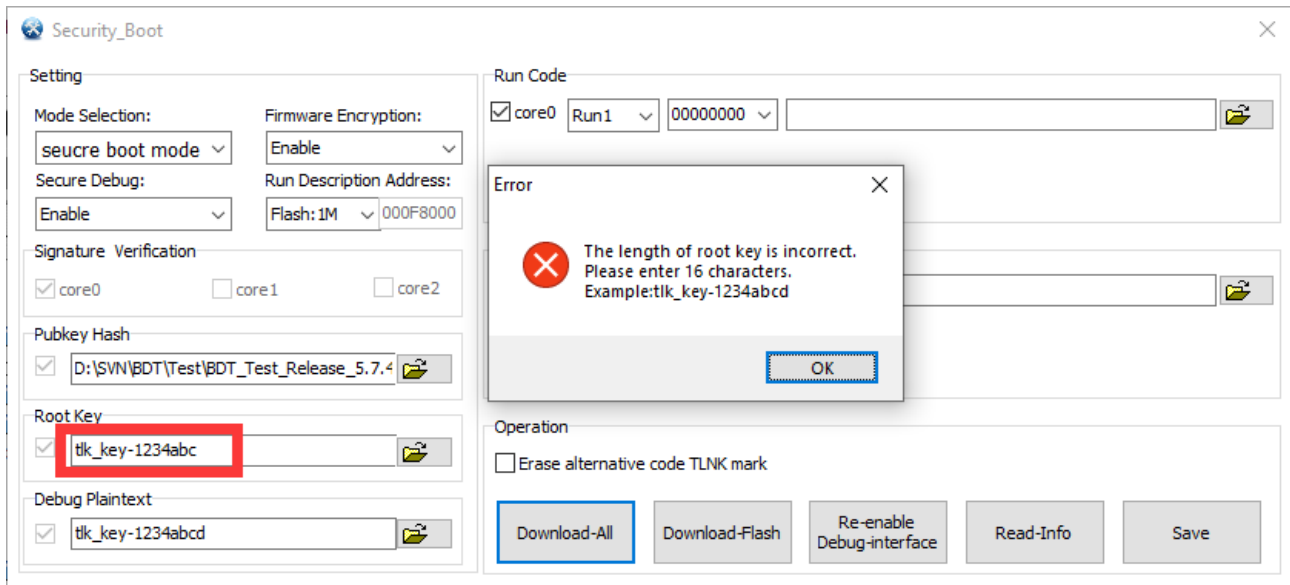


Figure 4.33: Root Key format error 1

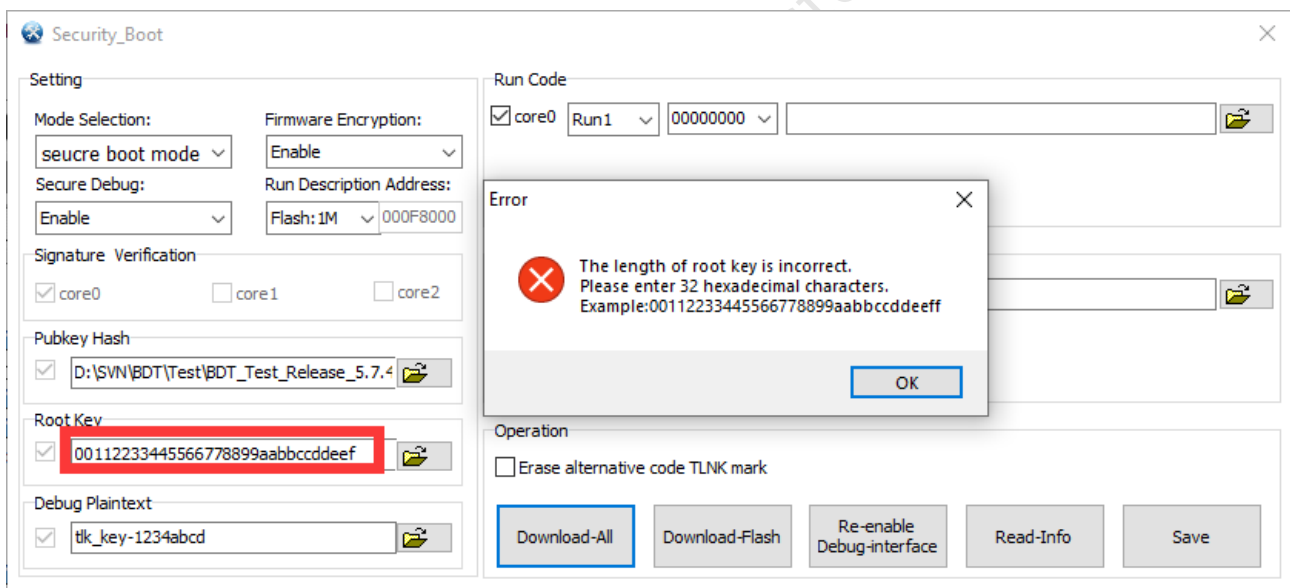


Figure 4.34: Root Key format error 2

Note:

The "Root Key" needs to be set only when the "Firmware Encryption" or "Secure Debug" function is enabled.

4.8.1.7 Debug Plaintext

Enter a string of 16 characters or 32 hexadecimal characters as the root key in the input box. It also supports selecting a text file to import the key.

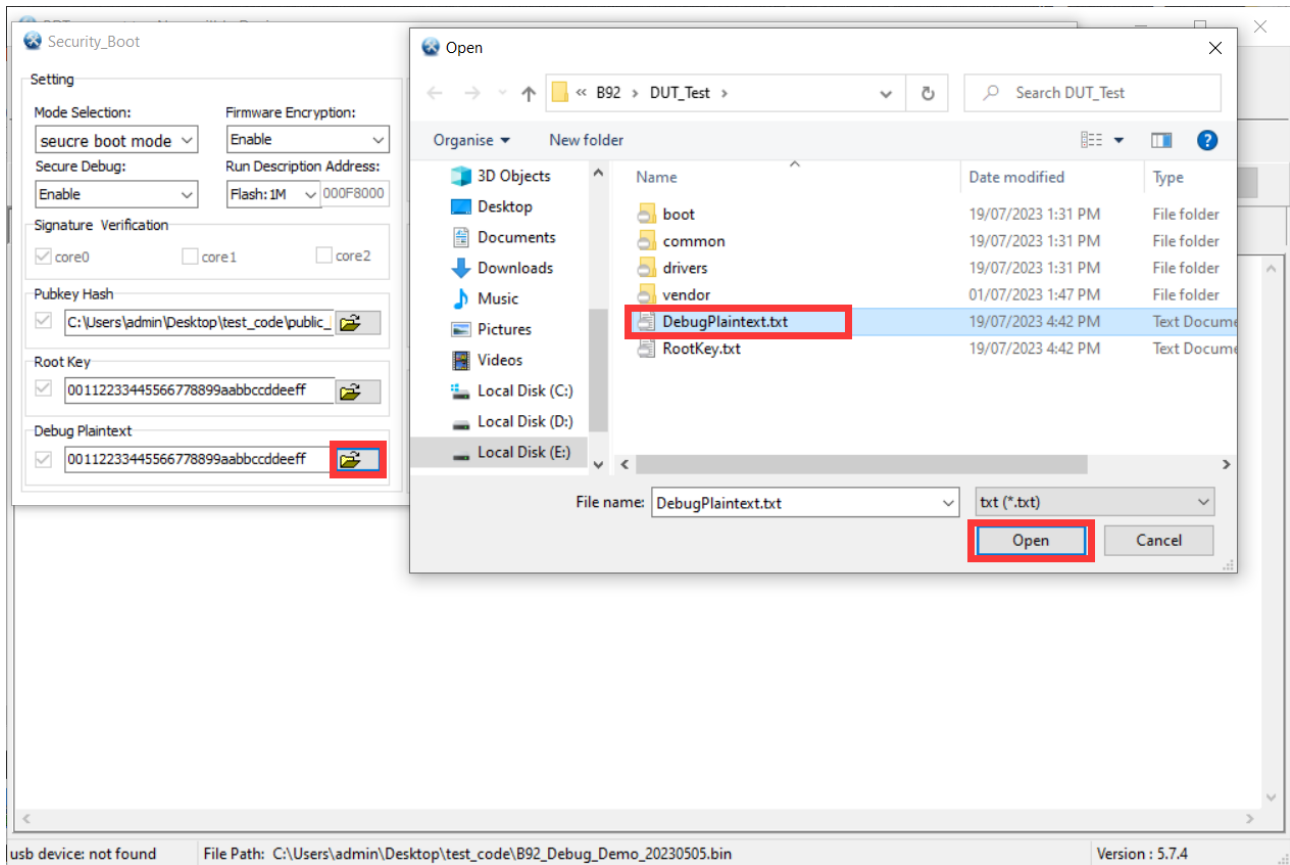


Figure 4.35: Import DebugPlaintext.txt file

Enter a string of 32 hexadecimal characters, for example: 00112233445566778899aabbccddeeff or 00112233445566778899AABBCCDDEEFF.

Enter a string of 16 characters, for example: tlk_key-1234abcd.

After clicking the "Download-All" button, the program will automatically check whether the input string meets the conditions. If the conditions are not met, an error will be reported.

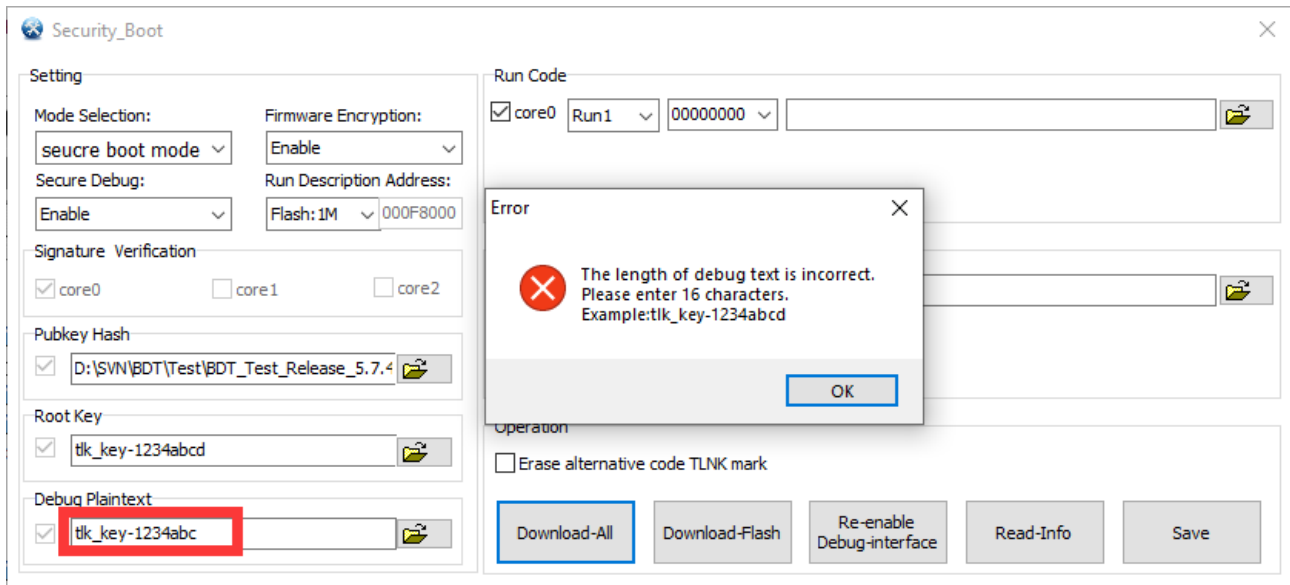


Figure 4.36: Debug Plaintext format error 1

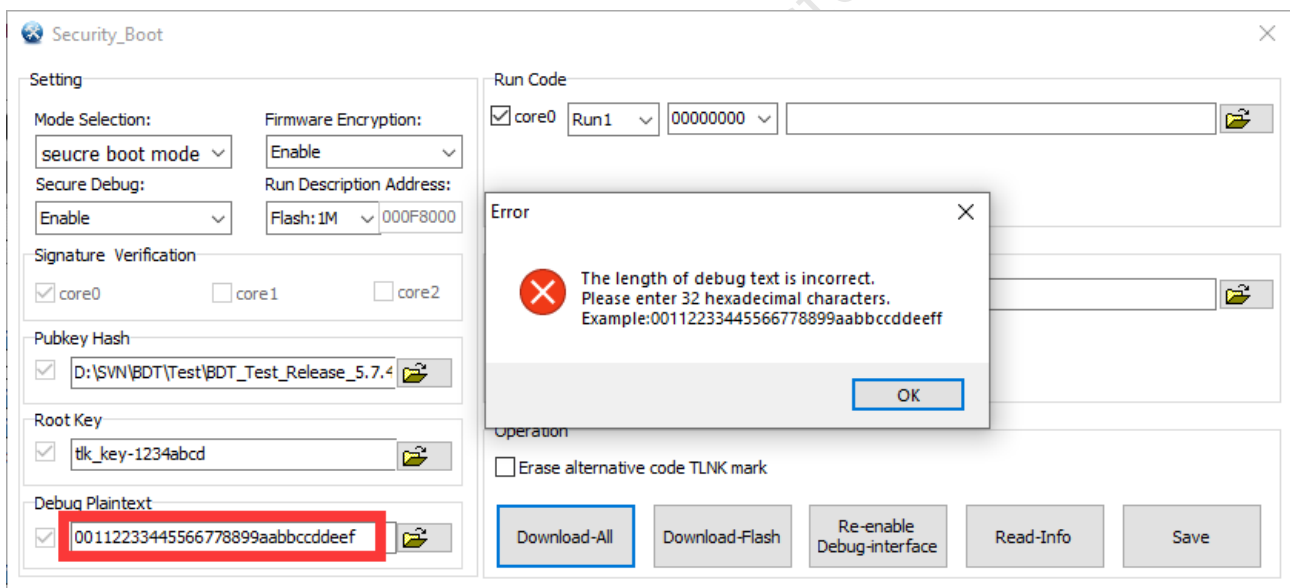


Figure 4.37: Debug Plaintext format error 2

Note:

The "Debug Plaintext" needs to be set only when the "Secure Debug" function is enabled.

4.8.1.8 Run Code

Tick the check box, select run code 0 or code 1 and run code address, and import the firmware path.

4.8.1.9 Run Code description

Tick the check box, select the run code description file path and import it.

Note:

This function is only available in "secure boot mode".

4.8.1.10 Download-All

The configuration information of the chip in the "Setting" area is downloaded to eFuse. The files selected in the "Run Code" and "Run Code description" areas are downloaded to flash.

Note:

When the "Secure Debug" function is enabled, click the "Download-All" button, regardless the download is successful or not, the "Debug Key" file will be generated in the path of tool config\sboot\key.

4.8.1.11 Download-Flash

The files selected in the "Run Code" and "Run Code description" areas are downloaded to flash.

Note:

If eFuse has been burned, just update the firmware program and click the "Download-Flash" button.

4.8.1.12 Re-enable Debug-interface

When the "Secure Debug" function is enabled, the "Debug" function of the chip will be disabled. Click the "Re-enable Debug-interface" button will pop up a dialog box. Select "Yes" and use the DebugKey.txt file in the default folder as the "Debug Key" to restore the "Debug" function. Select "No", you need to manually import the file as the "Debug Key" to restore the "Debug" function.

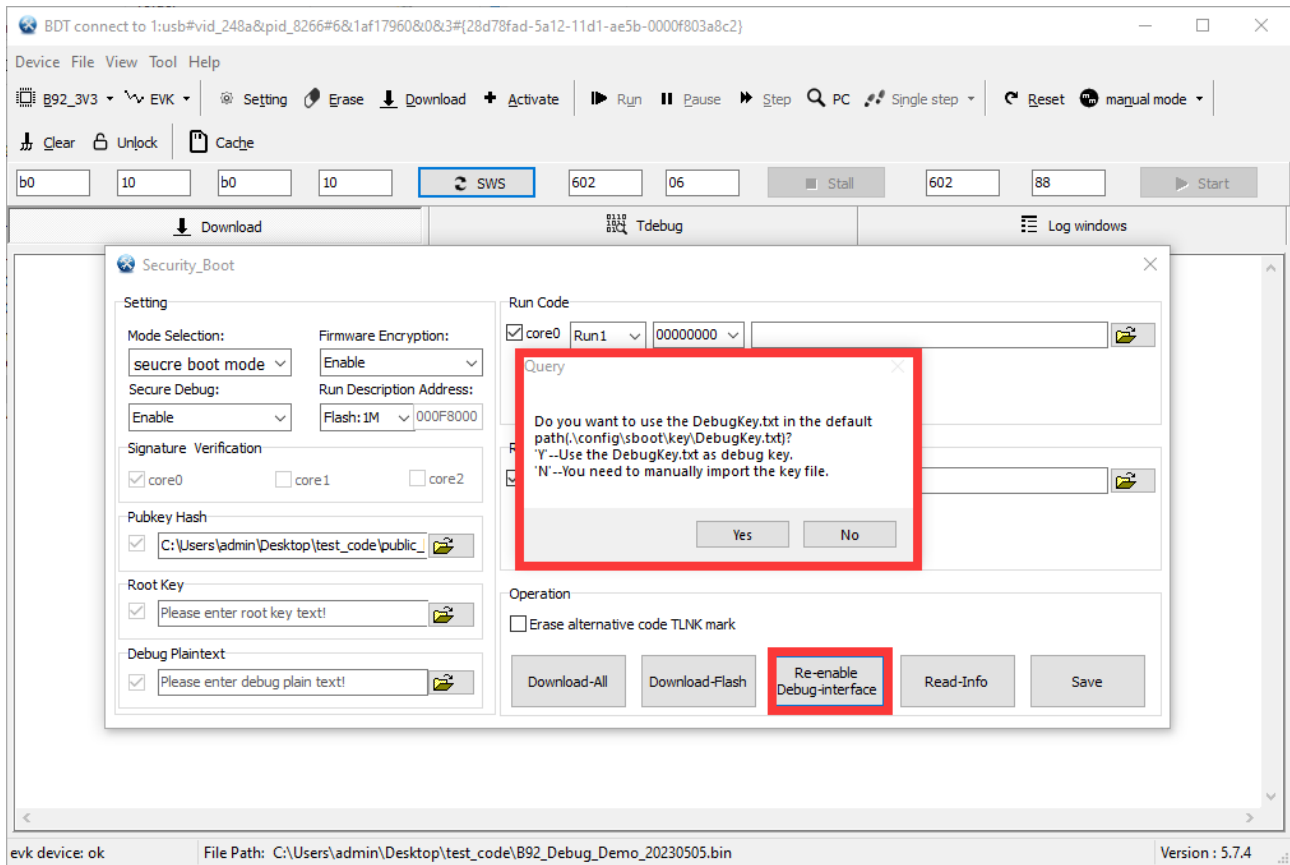


Figure 4.38: Re-enable Debug interface

4.8.1.13 Read-Info

Read the configuration information in the chip eFuse, excluding the information in the "Root Key" and "Debug Plaintext".

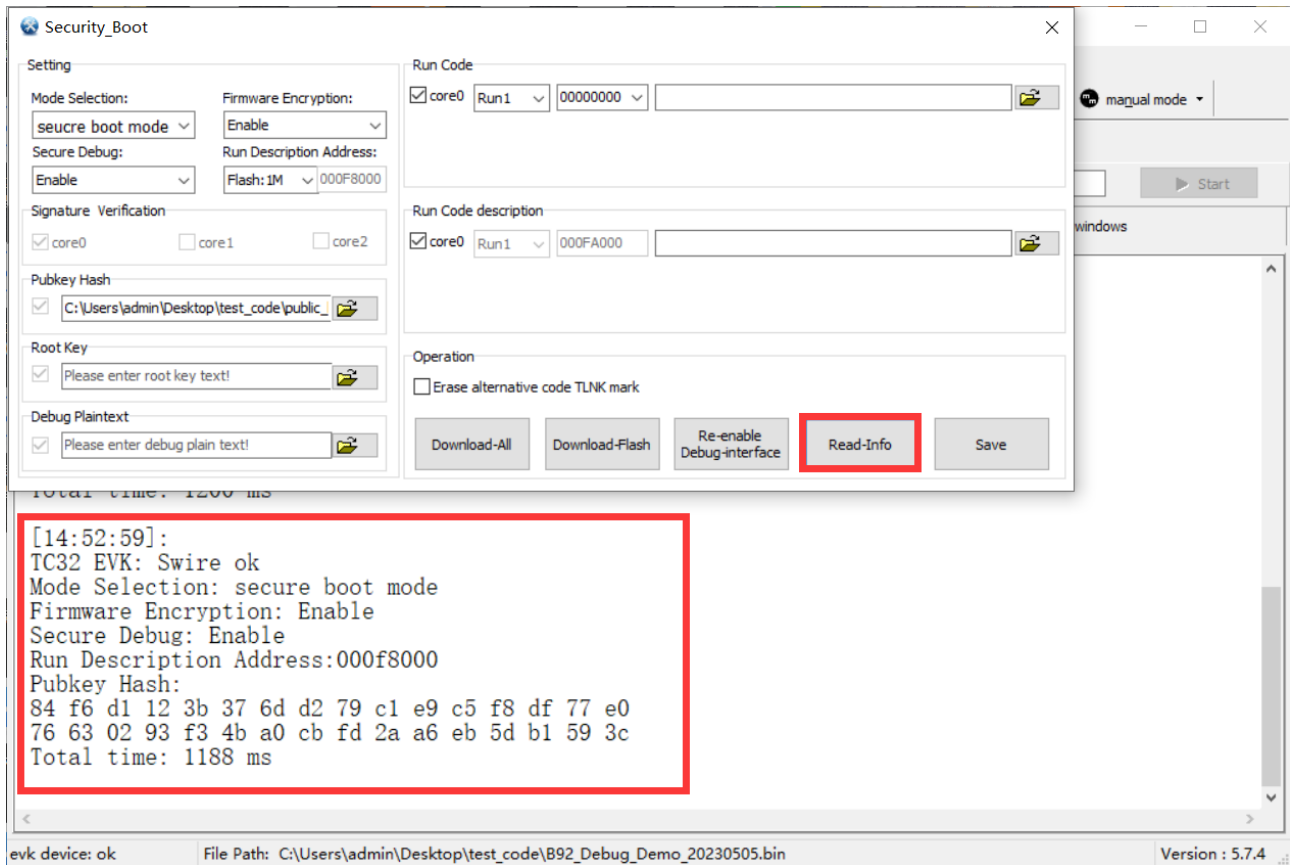


Figure 4.39: Read the information

4.8.1.14 Save

Save the configuration information in the interface, and save the information in "Root Key" and "Debug Plaintext" to the config\sboot\key path.



Figure 4.40: The path to the key folder

4.8.2 Commonly Used Secure Mechanism Options and Usage Instructions of the Tool

The Secure Boot, Firmware Encryption, and Secure Debug security mechanisms can be used individually and can be freely combined. However, in practice it is recommended to use the following combinations:

- Option 0: Secure Boot + Secure Debug
- Option 1: Firmware Encryption + Secure Debug

- Option 2: Secure Boot + Firmware Encryption + Secure Debug

4.8.2.1 Secure Boot + Secure Debug

Step1: Select "secure boot mode" in the "Mode Selection" drop-down box.

Step2: Select "Disable" in the "Firmware Encryption" drop-down box.

Step3: Select "Enable" in the "Secure Debug" drop-down box.

Step4: Select the corresponding flash size according to the chip package. The default selection here is "Flash: 1M".

Step5: In the "Run Code" area, select run code 0 or code 1 and the run code address, and import the firmware path.

Step6: After the previous step is completed, the "public_key_hash.bin" file path is automatically imported into the "Pubkey Hash" input box, and if it is not successful, it should be imported manually.

Step7: Import the run code description file path in the "Run Code description" area.

Step8: Enter the key in the "Root Key" and "Debug Plaintext" input boxes.

Step9: Click the "Download-ALL" button to complete the download.

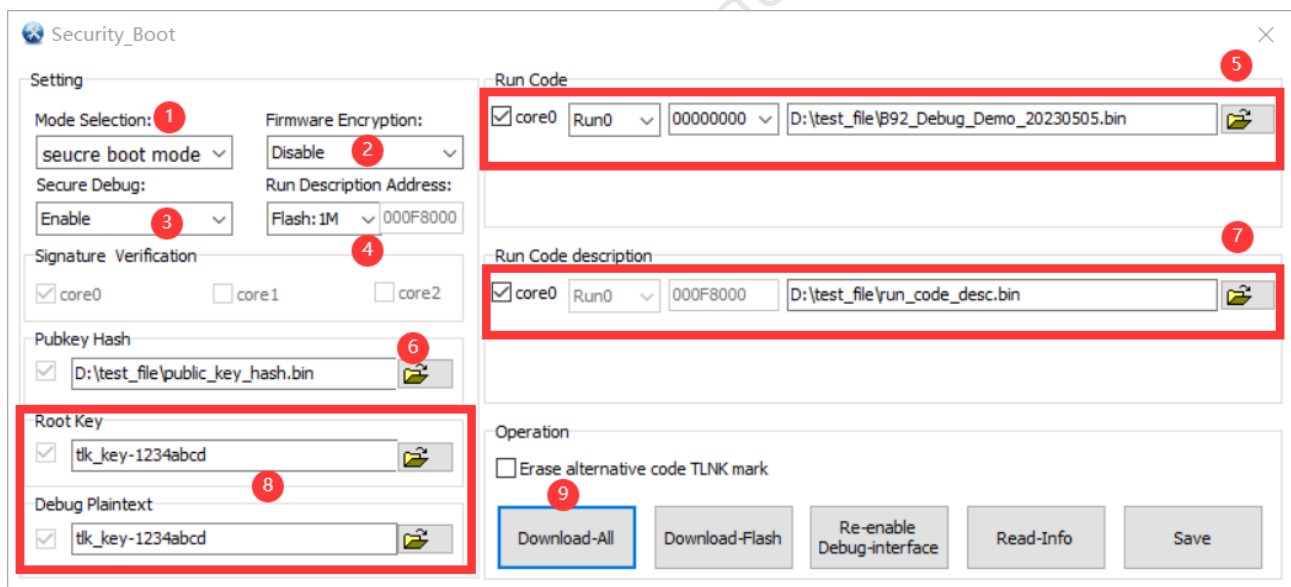


Figure 4.41: "Secure Boot + Secure Debug" burning steps

4.8.2.2 Firmware Encryption + Secure Debug

Step1: Select "normal mode" in the "Mode Selection" drop-down box.

Step2: Select "Enable" in the "Firmware Encryption" drop-down box.

Step3: Select "Enable" in the "Secure Debug" drop-down box.

Step4: Select the firmware download address and import the firmware path.

Step5: Enter the key in the “Root Key” and “Debug Plaintext” input boxes.

Step6: Click the “Download-ALL” button to complete the download.

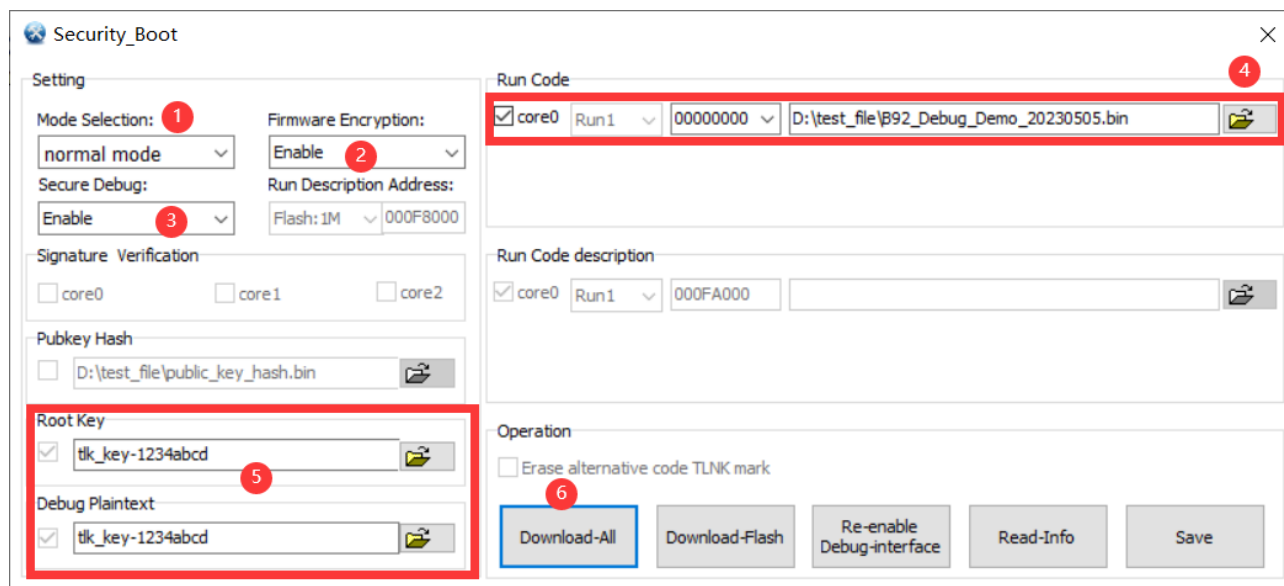


Figure 4.42: “Firmware Encryption + Secure Debug” burning steps

4.8.3 Secure Boot + Firmware Encryption + Secure Debug

Step1: Select “secure boot mode” in the “Mode Selection” drop-down box.

Step2: Select “Enable” in the “Firmware Encryption” drop-down box.

Step3: Select “Enable” in the “Secure Debug” drop-down box.

Step4: Select the corresponding flash size according to the chip package. The default selection here is “Flash: 1M”.

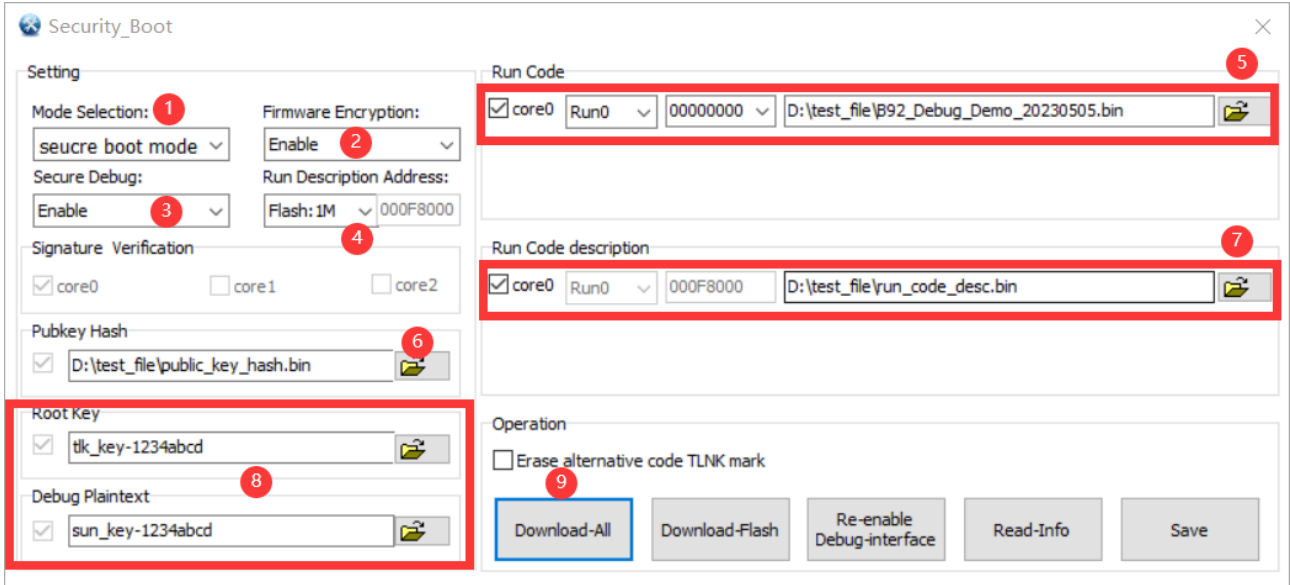
Step5: In the “Run Code” area, select run code 0 or code 1 and the run code address, and import the firmware path.

Step6: After the previous step is completed, the “public_key_hash.bin” file path is automatically imported into the “Pubkey Hash” input box, and if it is not successful, it should be imported manually.

Step7: Import the run code description file path in the “Run Code description” area.

Step8: Enter the key in the “Root Key” and “Debug Plaintext” input boxes.

Step9: Click the “Download-ALL” button to complete the download.



The screenshot shows the 'Security_Boot' window with the following settings and highlighted steps:

- Setting:**
 - Mode Selection:** 1. 'seucre boot mode' (Note: typo in image)
 - Firmware Encryption:** 2. 'Enable'
 - Secure Debug:** 3. 'Enable'
 - Run Description Address:** 4. 'Flash: 1M' and '000F8000'
 - Signature Verification:** 5. 'core0' (checked)
 - Pubkey Hash:** 6. 'D:\test_file\public_key_hash.bin'
 - Root Key:** 8. 'tlk_key-1234abcd' (checked)
 - Debug Plaintext:** 8. 'sun_key-1234abcd' (checked)
- Run Code:** 5. 'core0' (checked), 'Run0', '00000000', 'D:\test_file\B92_Debug_Demo_20230505.bin'
- Run Code description:** 7. 'core0' (checked), 'Run0', '000F8000', 'D:\test_file\run_code_desc.bin'
- Operation:** 9. 'Download-All' (highlighted button)

Figure 4.43: "Secure Boot + Firmware Encryption + Secure Debug" burning steps

5 Function Support List

Not all MCU types support all functions of this tool. Please refer to the table below.

ITEMS \ MCU Type	8266			8267/8269			8366/8368			8232/8233	8258		
	USB	SWIRE	SWIRE2USB	USB	SWIRE	SWIRE2USB	USB	SWIRE	SWIRE2USB	SWIRE	USB	SWIRE	SWIRE2USB
Download(Flash/SRAM)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Download(OTP)	×	×	×	×	×	×	✓	✓	✓	×	×	×	×
Erase FLASH	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Activate	×	✓	×	×	✓	×	×	✓	×	✓	×	✓	×
PC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Run/pause/step	×	×	×	✓	✓	✓	×	×	×	✓	✓	✓	✓
Memory access (OTP)	×	×	×	×	×	×	✓	✓	✓	×	×	×	×
Memory access (FLASH/CORE/ANALOG)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Figure 5.1: Function Support List