

Telink

Application Note

Telink Burning and Debugging Tool(BDT) User Guide

AN_22010600-E2

Ver 1.0.0

2022.1.6

Keyword

Linux BDT

Brief

This document is the development guide for Telink Burning and Debugging(BDT) in Linux.

Acknowledgements

Published by

Telink Semiconductor

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

© Telink Semiconductor
All Right 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 © 2021 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:

telinknsales@telink-semi.com

telinknsupport@telink-semi.com

Revision History

Version	Change	Description

V1.0.0		Initial release.

Telink Semiconductor

Contents

Acknowledgements	2
Legal Disclaimer	2
Information	2
Revision History	3
1 Software Introduction	5
2 Usage	6
2.1 Select the usb device	6
2.2 sws	6
2.3 reset	7
2.4 Activate	8
2.5 Read and write (Memory, flash).	8
2.6 Download	11
2.7 Erase flash	13
2.8 Debug	14
2.9 Program terminates	15
2.10 Command line	16
2.11 Device user-defined label	16
2.12 Download mode	16
2.13 Burning EVK firmware update	18
3 Software configuration	19
3.1 Configure optional chip model and flash firmware path	19

1 Software Introduction

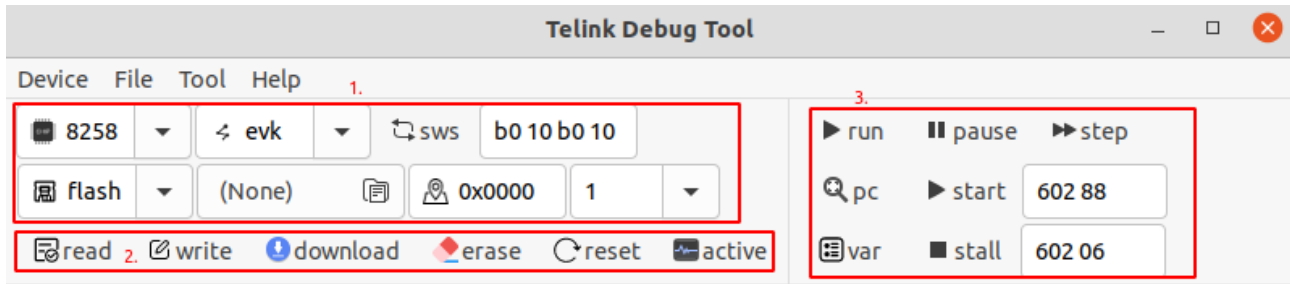


Figure 1

(1) Parameter Configuration - Area 1.

- Configuration Chip model(9518, 8258, 8266).
- Access the device's model(evk, usb).
- Access storage type(flash, ram, analog).
- Access start address.
- Access to the number of bytes.
- Configure the download file path.

(2) After setting parameters in area 1, access the device - Area 2.

- Click the read and write key, read and write operations for storage space.
- Click the Download button to download the bin file to flash or RAM.
- Click the erase button to erase flash area, the starting address.
- Click the Reset button to reset from flash or RAM.
- Click the active button to wake up the development board.

(3) Debug - Area 3.

- run: Run the program.
- pause: Pause program.
- step: Step through the program.
- pc: View the current program running pointer.
- var: View global variable values for the current program.
- start: Start running program.
- stall: Terminate the program.

2 Usage

2.1 Select the usb device

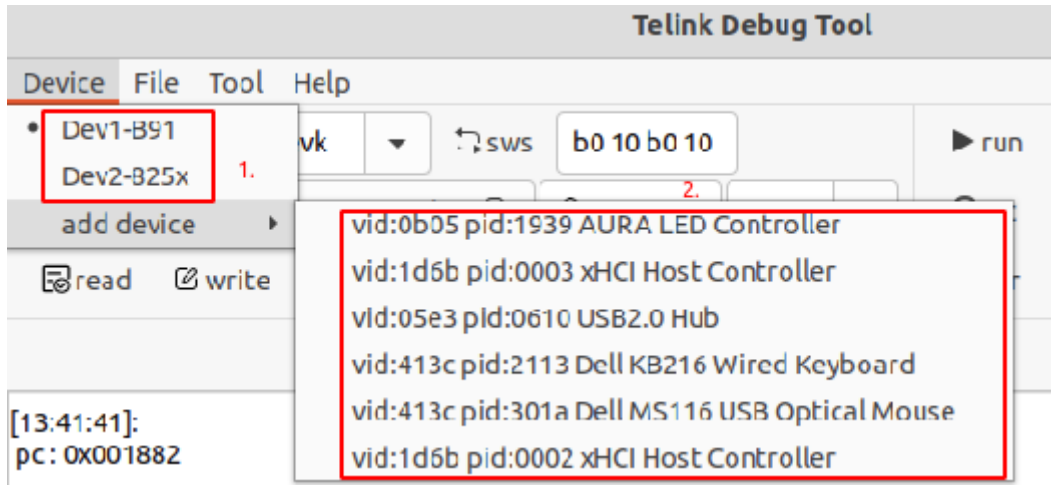


Figure 2

Click the Device menu bar to display the connected EVK devices. The EVK devices identified by this list (VID: 0x248A, PID: 0x8266) are shown in area 1 shown in the figure above. Note If the VID and PID values of the devices to be added are different, click Add Device.

The add-device list displays all usb devices scanned by the system. Add devices based on your requirements, as shown in area 2 in the figure above. Please note, please add correct need connection device, or the device may malfunction, you need to plug connection, to return to normal use.

2.2 sws

Set the single-line synchronization speed to ensure the normal connection between EVK device and development board. When the development board program is in low power mode, this function cannot be used normally, so it is necessary to wake up the development board first.

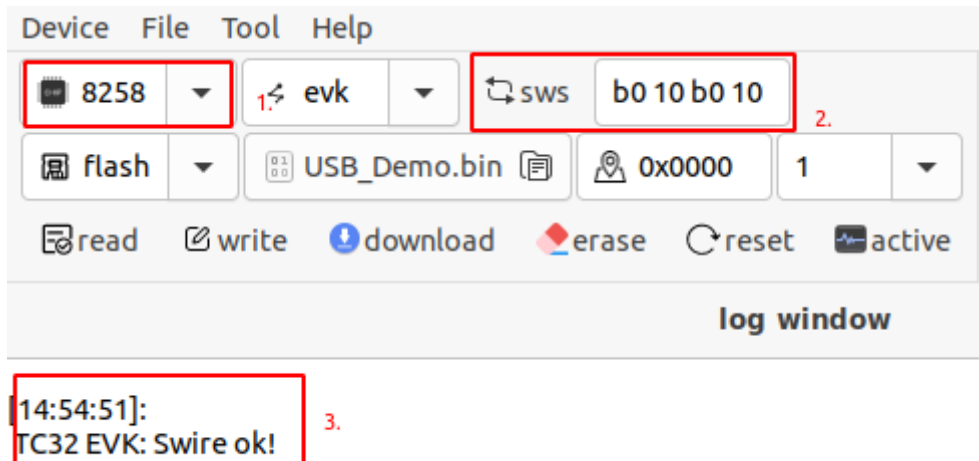


Figure 3

- (1) Select the chip model.
- (2) Configure SWS register address and clock value (default unchanged).
- (3) Successful results of SWS.

2.3 reset

Reset the development board and select the launcher to run from Flash or SRAM.

- (1) Reset, from flash to start, as shown in the figure below.

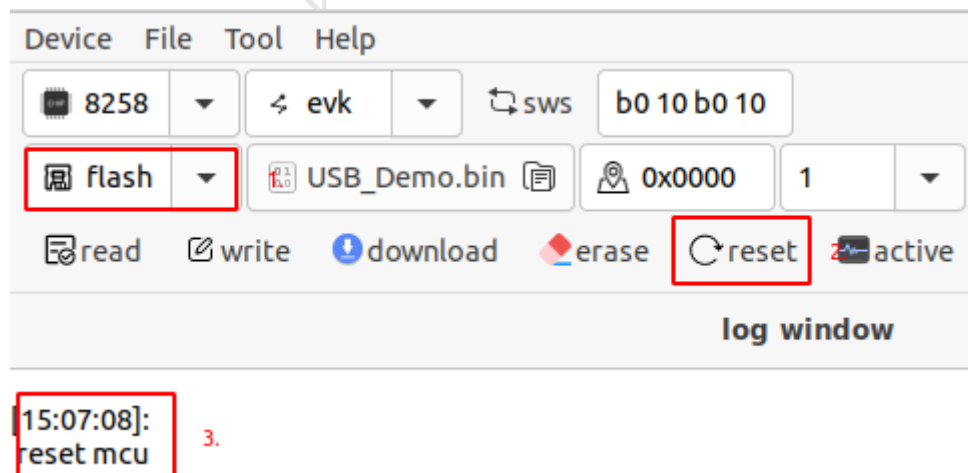


Figure 4

- (2) Reset, from sram, as shown in the figure below.

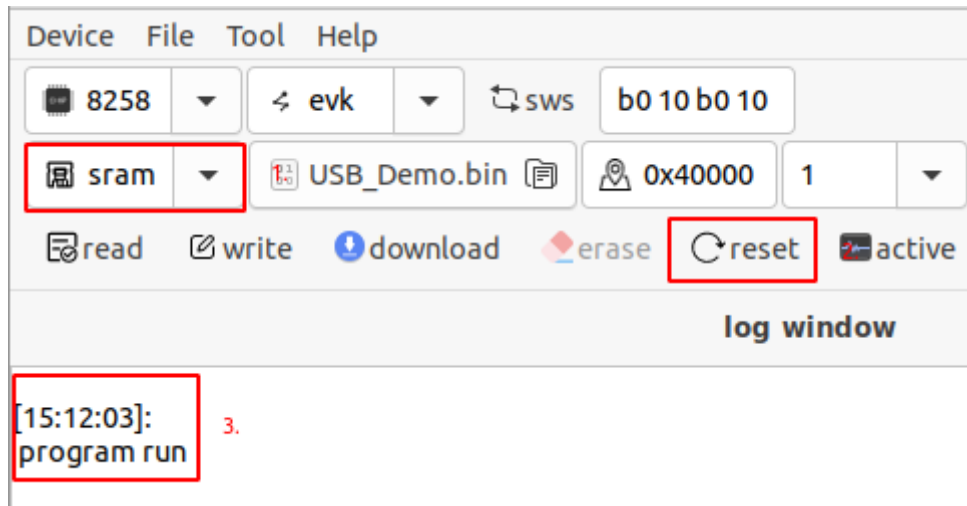


Figure 5

24 Activate

Development board to run the program in a low power mode, wake up.

2.5 Read and write (Memory, flash).

(1) Read flash.

- Read 16 bytes of data from address 0x0000 as shown in the figure below.

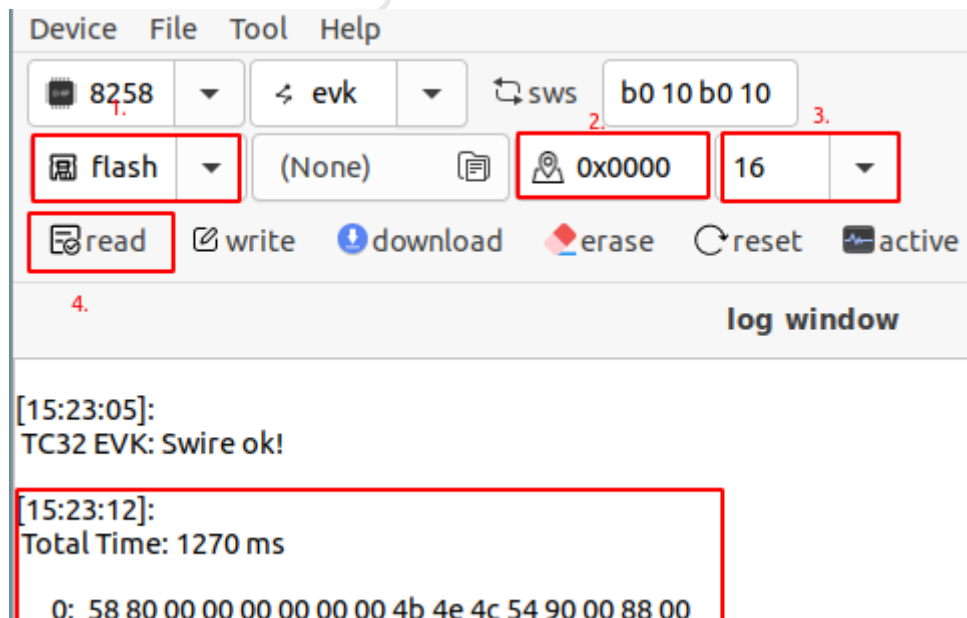


Figure 6

- Read 1K bytes of data from address 0x0000. The read quantity greater than or equal to 1K will not be directly printed and stored in a file, as shown in the figure below.

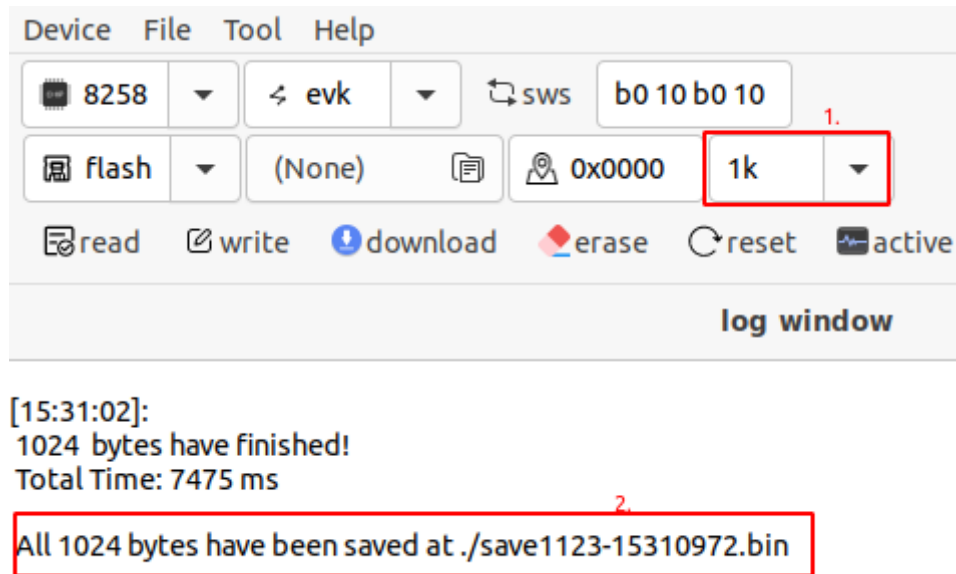


Figure 7

(2) Read SRAM.

- The operation of reading sRAM is similar to the above operation of reading Flash. Read 16 bytes of data from 0x40000 (8258 chip SRAM start address) as shown in the figure below.

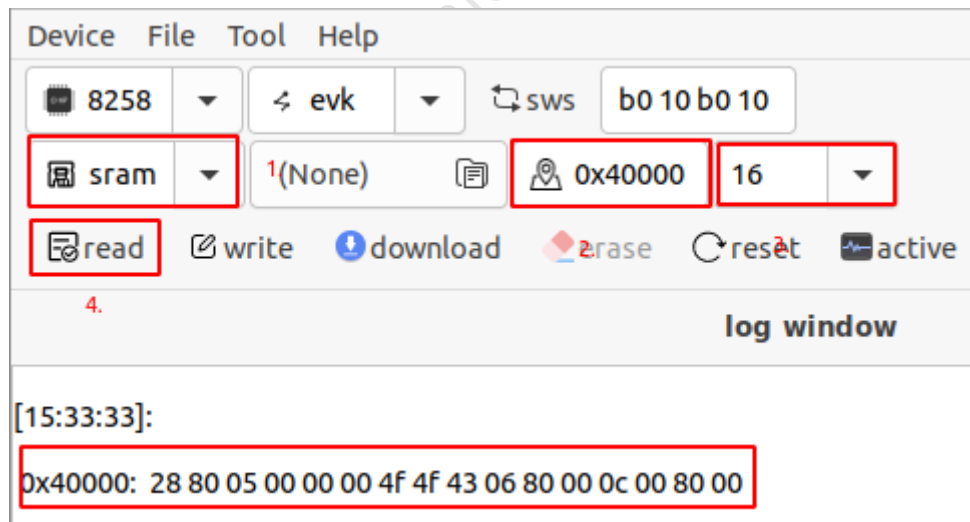


Figure 8

(3) Write flash

- Write four bytes of data at 0x0000, and configure the start address (the following figure - the area 2), the number of bytes written (the following figure - the area 3), and the content written (the following figure - the area 4), with Spaces between bytes).
- Success - (the following figure - the area 6).
- Failure - (the following figure - the area 7). Erase before writing, refer to 2.7 for details.

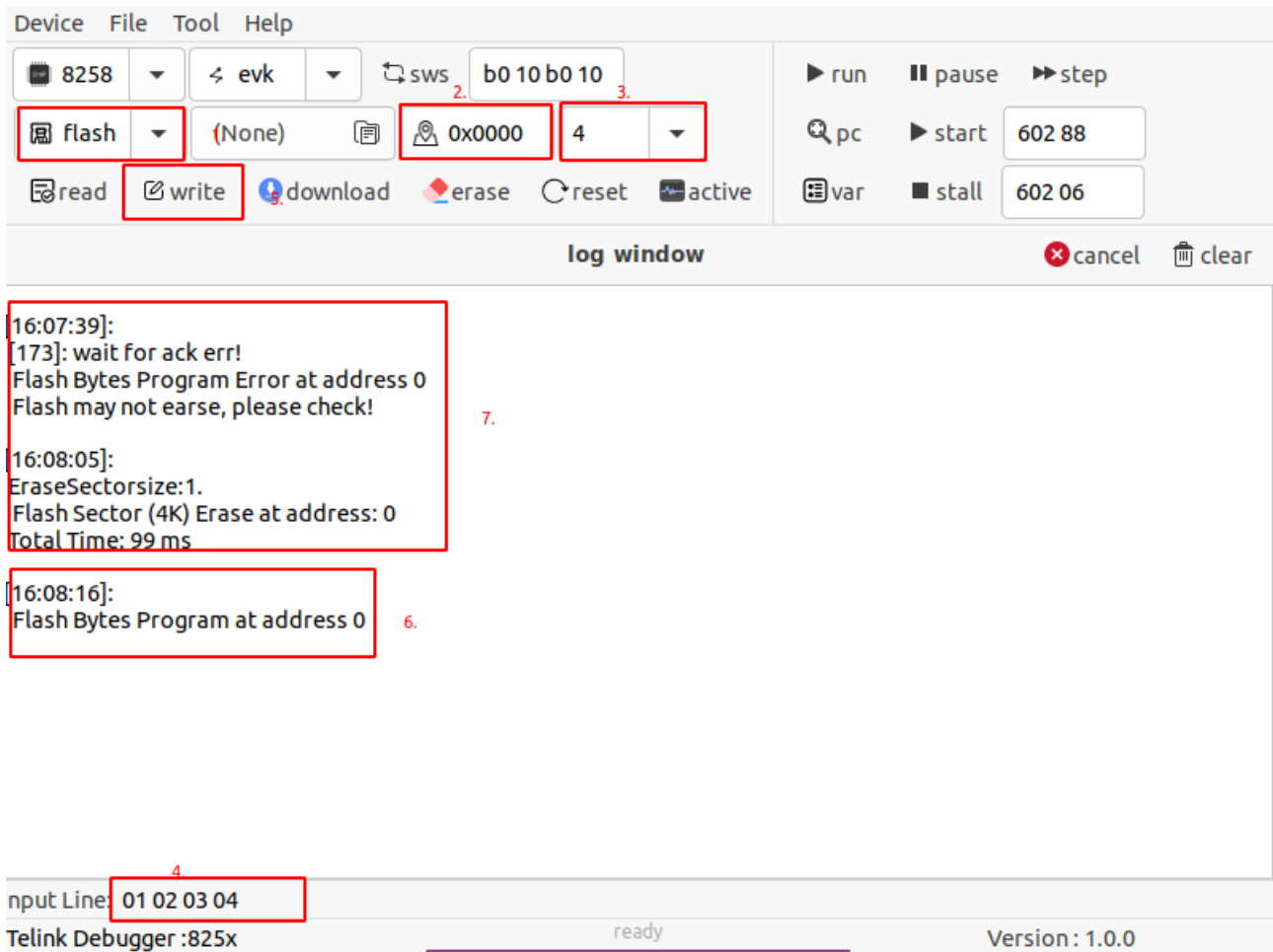


Figure 9

(4) Write sram

- Write 2 bytes of data at 0x40001, as shown in the following figure. Below 4 area after the input data, need not press enter, or as a command execution.

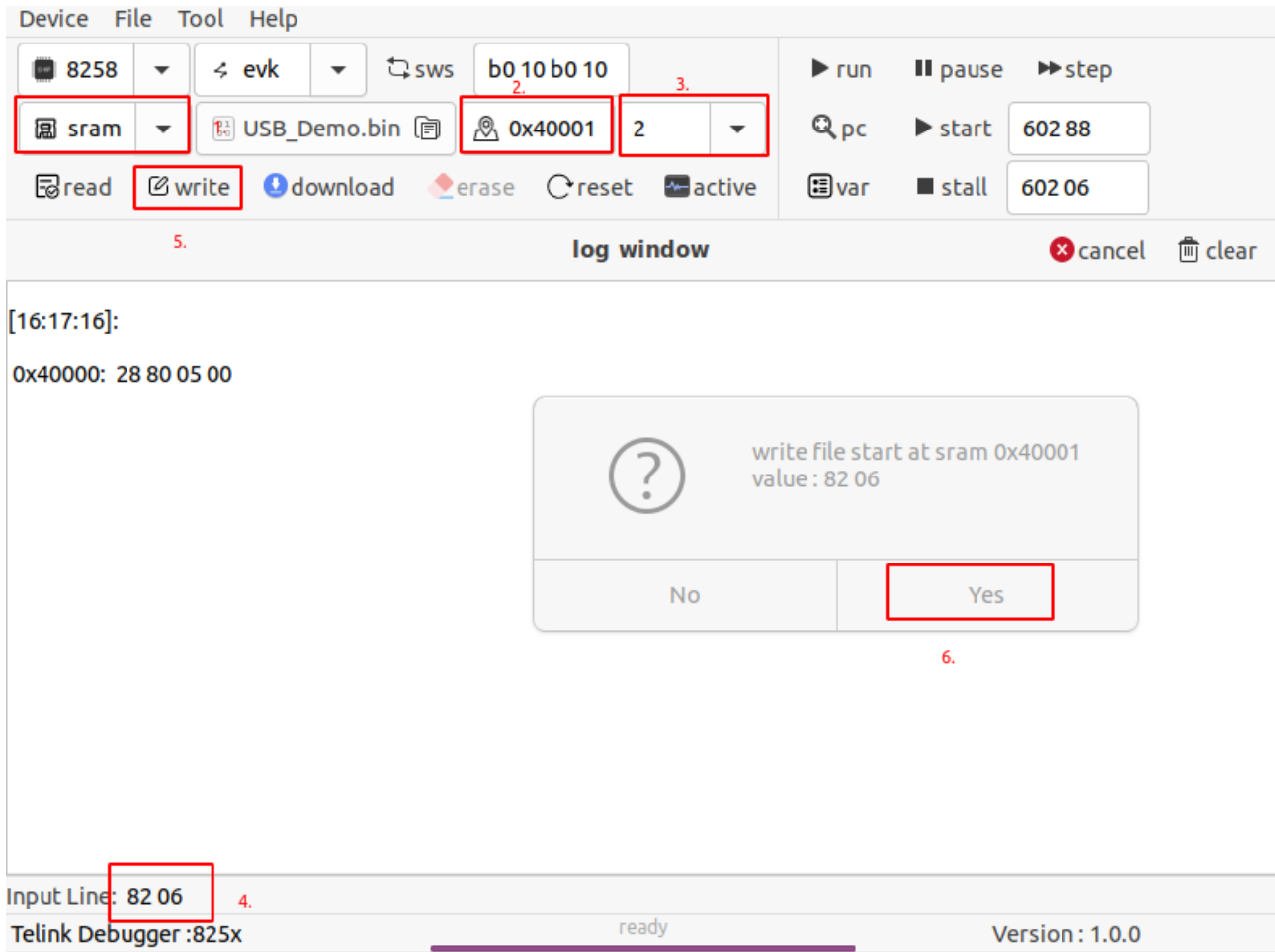


Figure 10

2.6 Download

- (1) Download the bin file to the flash, set the bin file path (the following figure - 2 area), and set the start address. If there is no response for a long time during the download, check whether the device is in low-power mode.

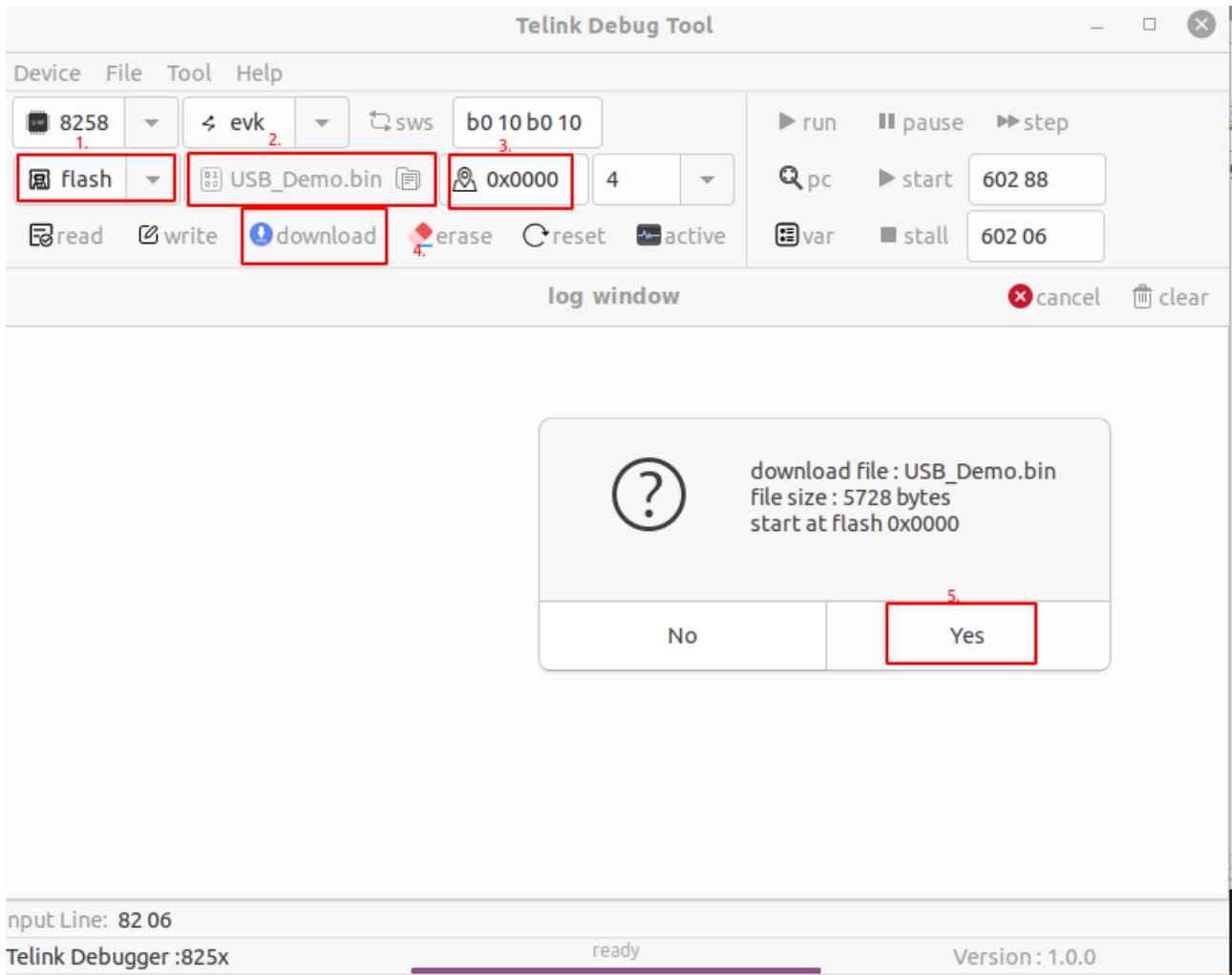


Figure 11

- (2) Download the bin file to the sram, set the bin file path (the following figure - 2 area), and set the start address.

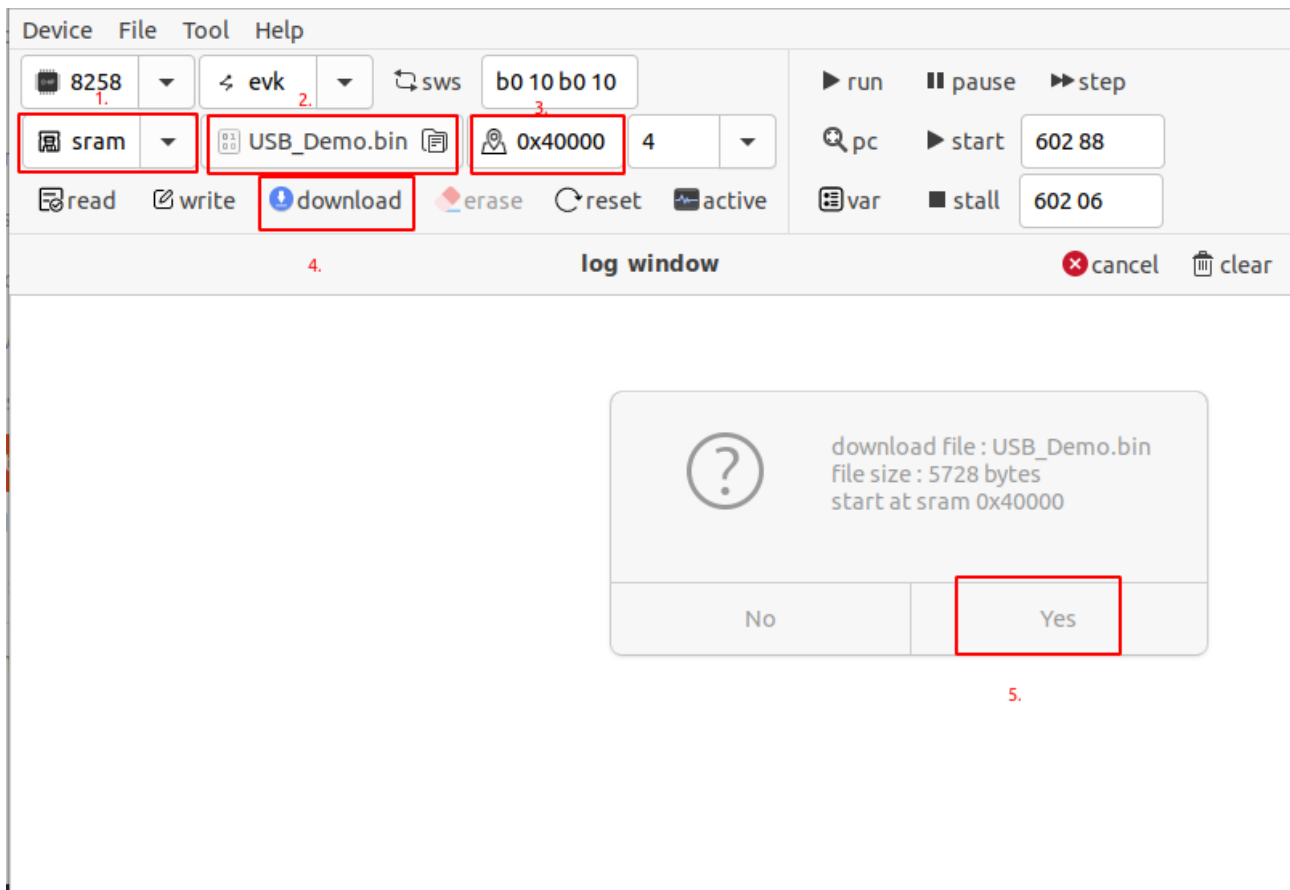
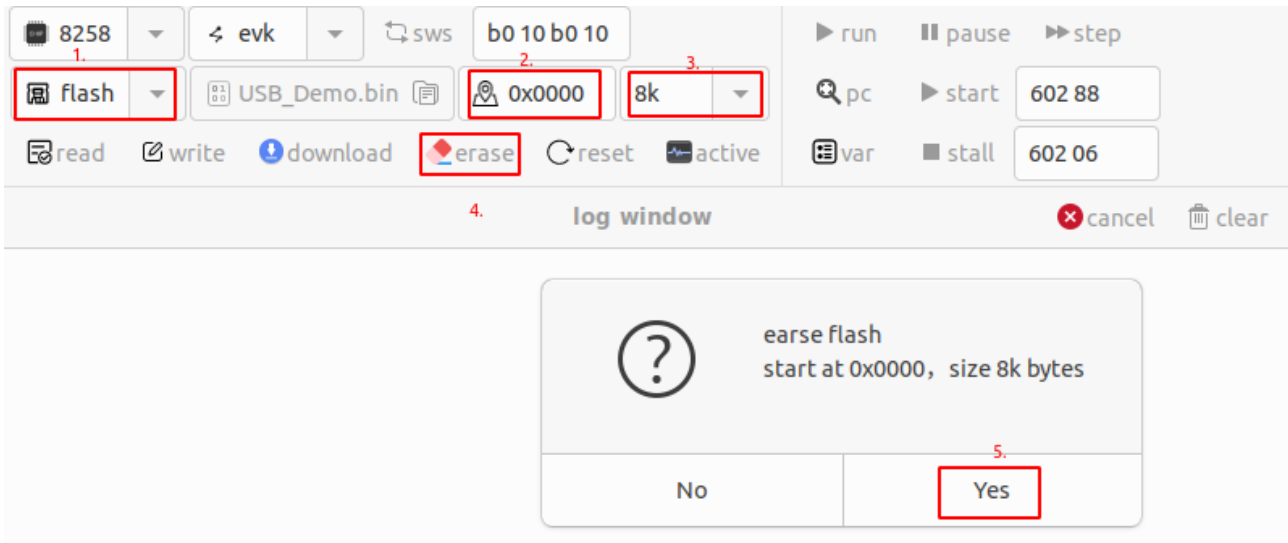


Figure 12

2.7 Erase flash

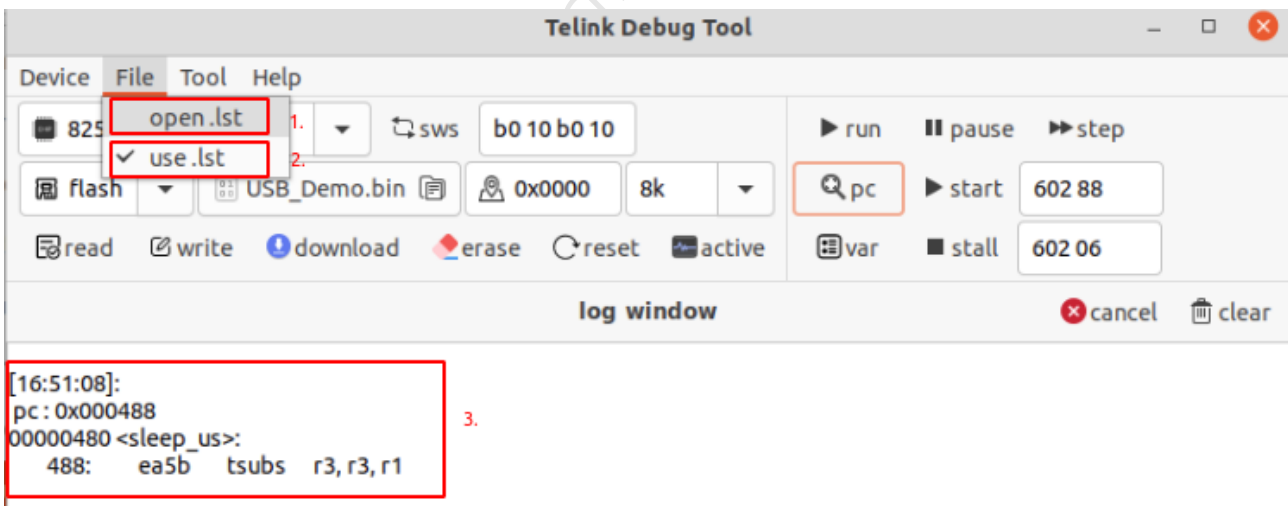
Erase the starting address 0 x0000, erase the byte 8 KB size (the following figure - 3 area, erasing the custom input), erase the smallest unit of 4 KB.


Figure 13

2.8 Debug

(1) pc : View the program running pointer

- To view details about the PC value, open the .LST file for running the program (the following figure - 1 area) and select Use. LST file (the following figure - 2 area).


Figure 14

(2) View global variable values for the current program.

- This function requires opening and using. LST files, as shown in the previous section.

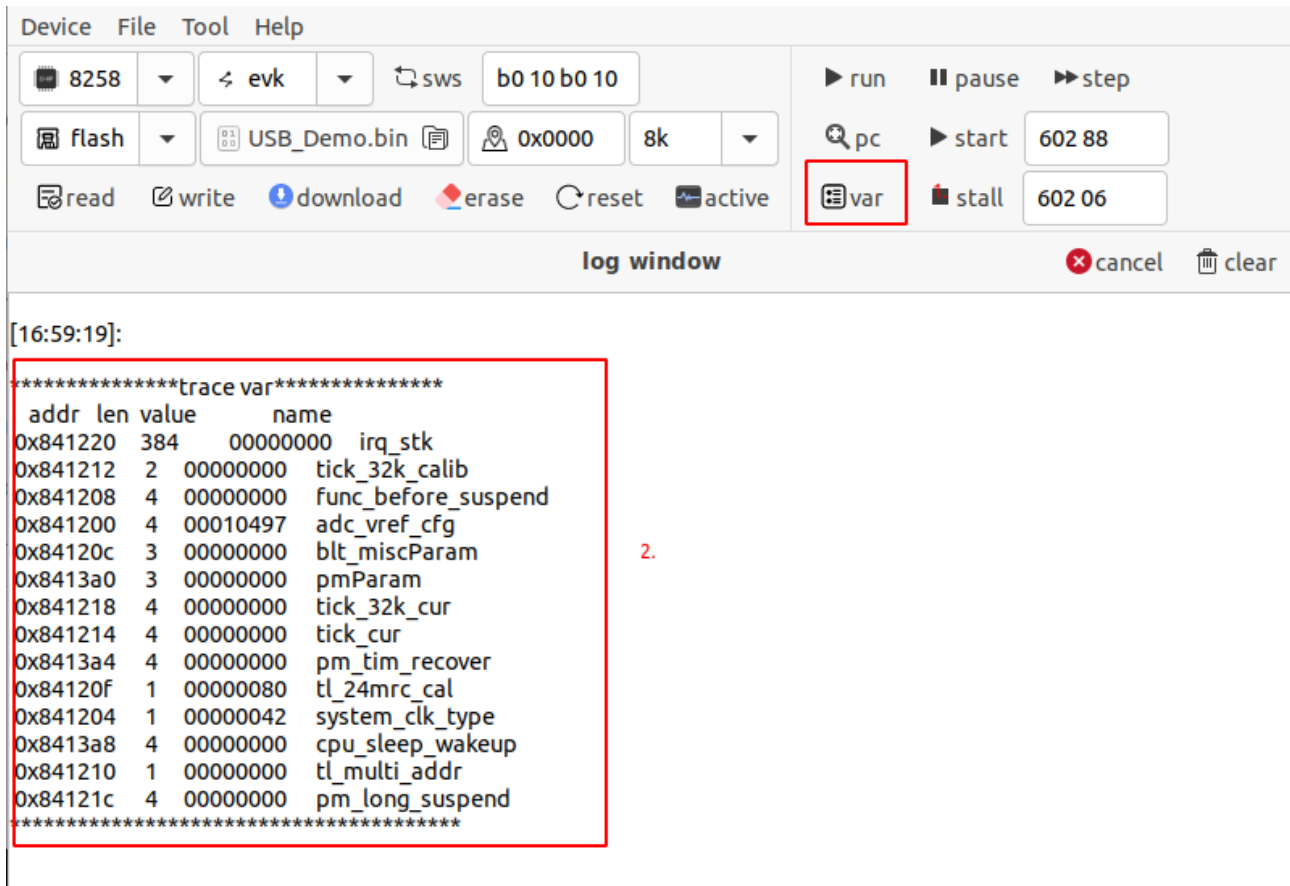


Figure 15

(3) run

- Continue the program after pause is used.

(4) pause

- Pause the program.

(5) step

- One step operation program.

(6) start

- The program is restarted. The operation address 0x602 and value 0x88 are generally unchanged.

(7) stall

- The program terminates. Operation address 0x602 and operation value 0x06 generally remain unchanged.

2.9 Program terminates

Cancel button. If there is no response for a long time, you can directly exit the current command.

2.10 Command line

In command line mode, enter the command and press Enter to execute the command. For details about the command usage format, see the bdt-cmd User guide.

```
[17:09:36]:
TC32 EVK: Swire ok!

[17:09:49]:

0x40000: 58 80 00 00 00 00 00 00 4b 4e 4c 54 90 00 88 00

[17:09:59]:
Total Time: 1270 ms

0: 58 80 00 00 00 00 00 00 4b 4e 4c 54 90 00 88 00
```

Input Line: `./bdt 8258 rf 0x0000 -s 16`

Figure 16

2.11 Device user-defined label

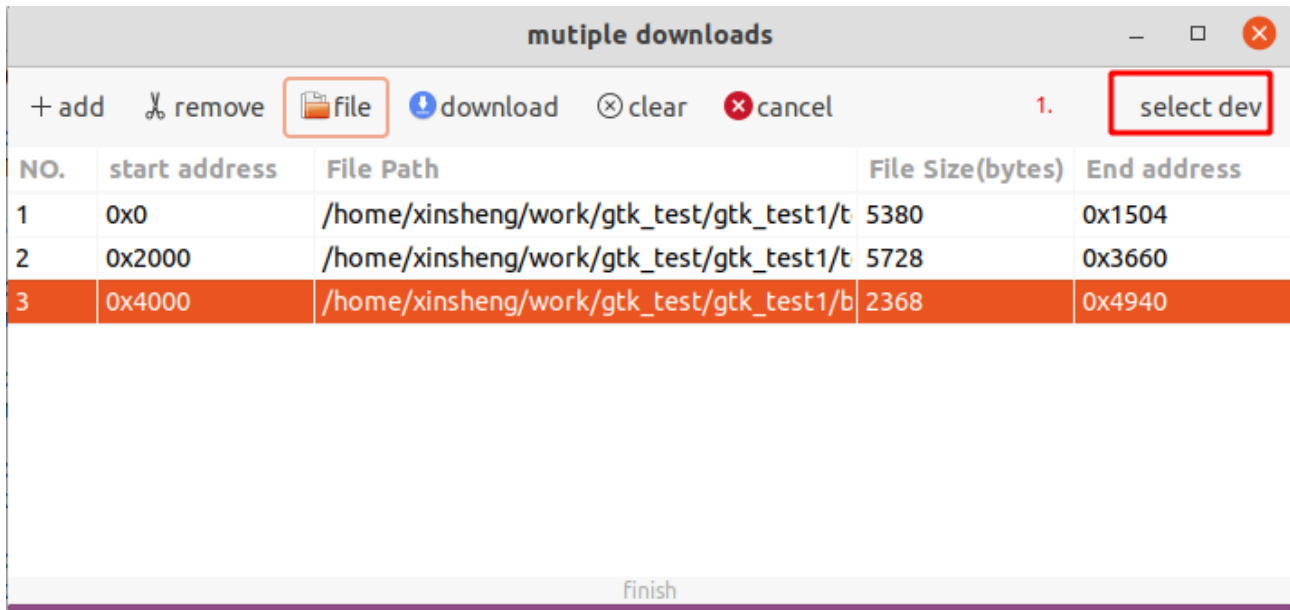
In the case of insert multiple devices to prevent equipment. The user defines the name of the connected device. Click to modify the label text, and press Enter.

Input Line: `dongle-master` ready Version: 1.0.0

Figure 17

2.12 Download mode

Tool -> mutiple downloads. This function is applicable to the device downloading multiple bin files, and multiple devices downloading files at the same time.


Figure 18

- (1) add: Add download items and arrange them in sequence number (NO) .
- (2) remove: Delete the specified download item, click the download item with the mouse, and then click Remove.
- (3) file: Specify the download bin file path.
- (4) download: Execute download command.
- (5) clear: Delete all downloaded items.
- (6) cancel: Cancel all downloading processes.

Use steps

- (1) Click the Start Address bar, enter the start address, and **press Enter to end**.
- (2) Click File and select the bin file path. The file size is automatically resolved and the end address is calculated. Address range conflicts will be marked in red.
- (3) Configure the device, as shown in the figure above - label 1. Click to enter the configuration interface, input the chip signal, **enter to end**

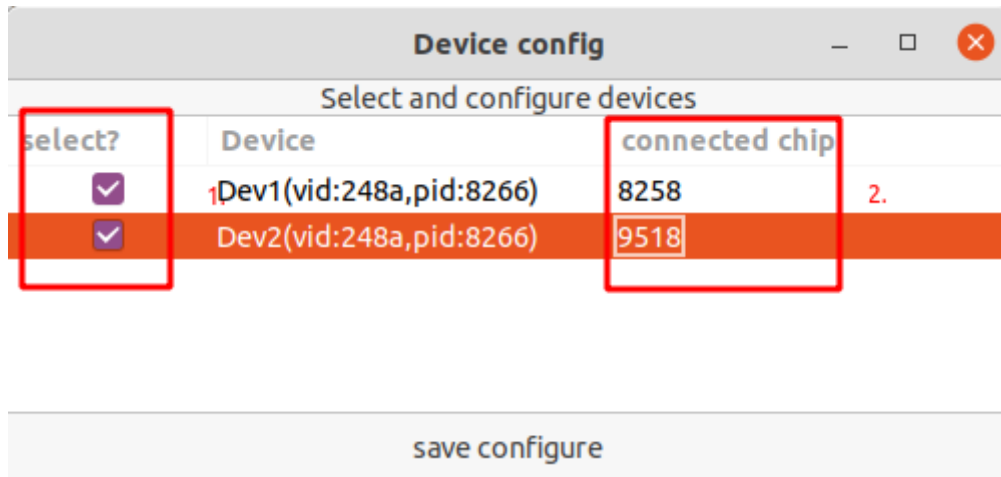


Figure 19

- (5) As shown in the figure above, select the download device with label 1. Label 2, configure the chip model connected to the burning evk. Click Save configure.
- (6) Click Download to download and wait for the result, as shown in the following figure. (Please ensure that all devices are properly connected before downloading)

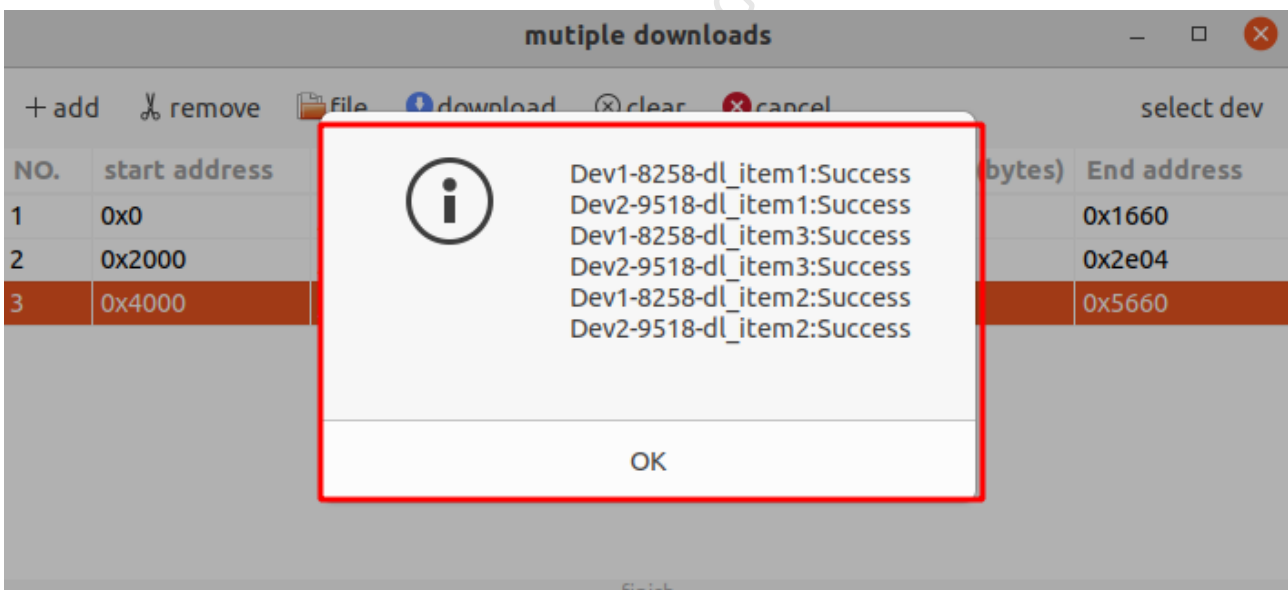


Figure 20

2.13 Burning EVK firmware update

View the current firmware version and update Burning EVK firmware.

Do not remove or insert the device during the update. After the update, reinsert the device.

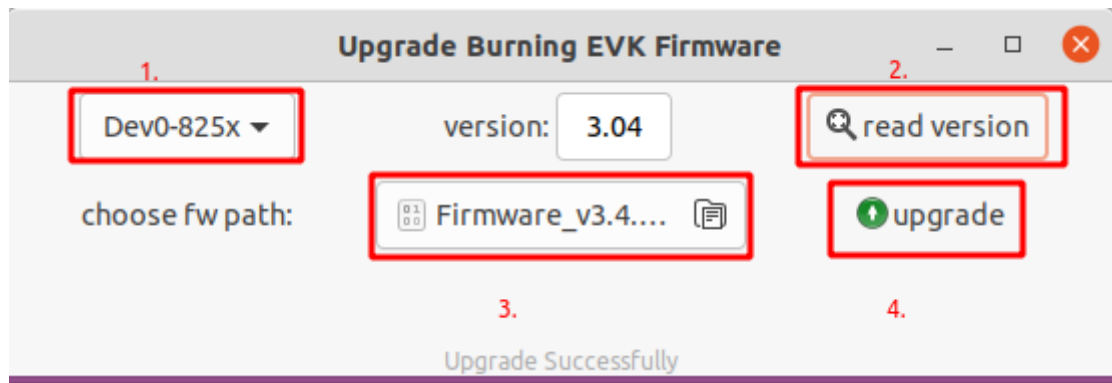


Figure 21

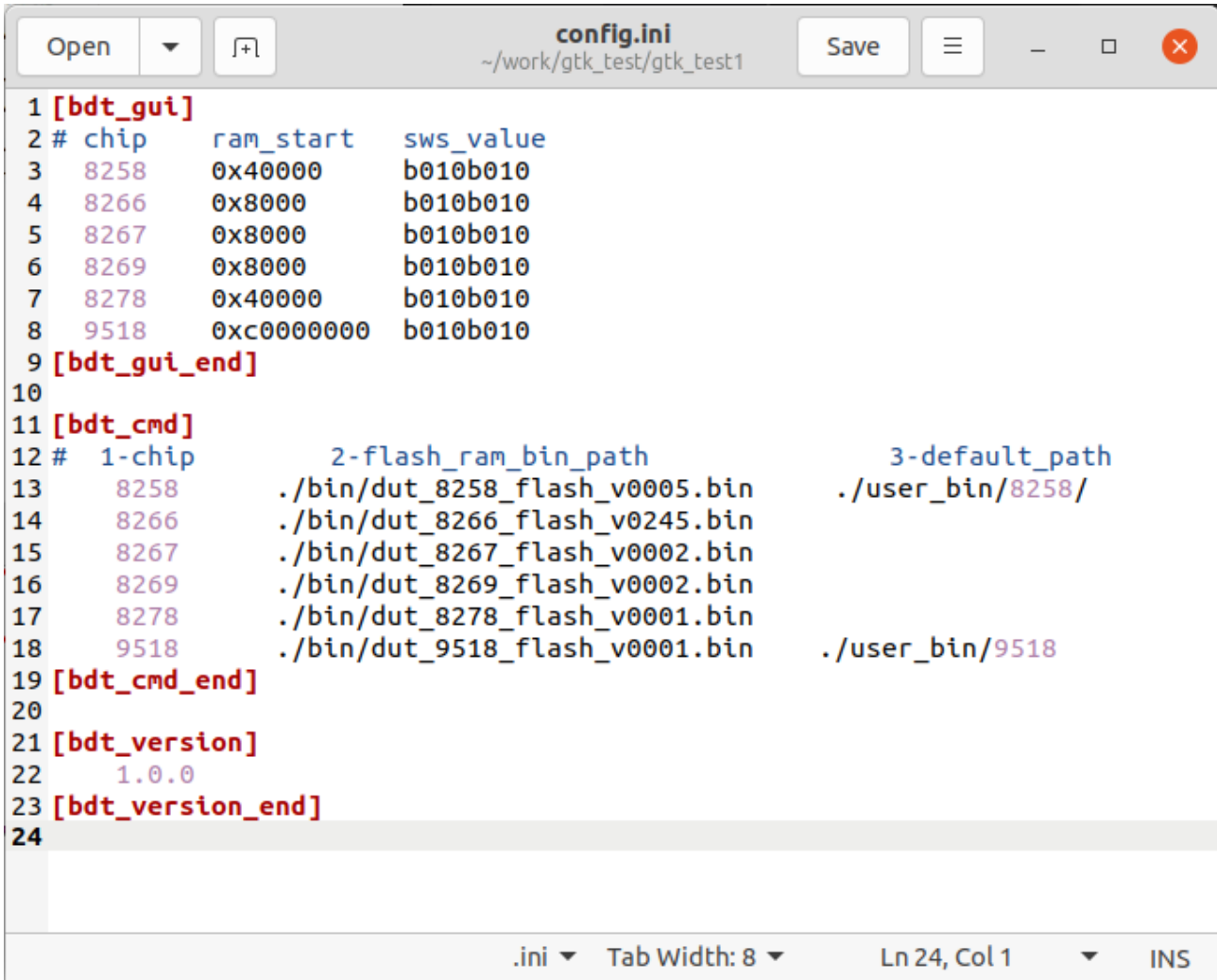
Step

- Above - 1 area, select need to update the firmware of the device.
- Above - 2 area, click can query the current firmware version.
- Above - 3 area, click on the select the firmware file path.
- Above - 4 area, click update to specify the firmware to the device.

3 Software configuration

3.1 Configure optional chip model and flash firmware path

Click to open config.ini file.



```

1 [bdt_gui]
2 # chip      ram_start  sws_value
3 8258      0x40000      b010b010
4 8266      0x8000       b010b010
5 8267      0x8000       b010b010
6 8269      0x8000       b010b010
7 8278      0x40000      b010b010
8 9518      0xc0000000    b010b010
9 [bdt_gui_end]
10
11 [bdt_cmd]
12 # 1-chip      2-flash_ram_bin_path      3-default_path
13 8258      ./bin/dut_8258_flash_v0005.bin      ./user_bin/8258/
14 8266      ./bin/dut_8266_flash_v0245.bin
15 8267      ./bin/dut_8267_flash_v0002.bin
16 8269      ./bin/dut_8269_flash_v0002.bin
17 8278      ./bin/dut_8278_flash_v0001.bin
18 9518      ./bin/dut_9518_flash_v0001.bin      ./user_bin/9518
19 [bdt_cmd_end]
20
21 [bdt_version]
22 1.0.0
23 [bdt_version_end]
24

```

Figure 22

- (1) In bdt_cmd area adds the chip model and the corresponding flash firmware path, and each item is separated by a space. (if the chip model in use is not configured). The chip model configured at this position will appear in the chip model optional box of the graphical interface.
- (2) bdt_gui area is used to configure the starting address of ram and the set value of SWS register.
- (3) bdt_version area is not modified.