Application Note

Telink Burning and Debugging Tool(BDT) CMD User Guide

AN-22010602-E4

V1.7.0

2024.11.13

Keyword

BDT CMD

Brief

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

Acknowledgements

Published by
Telink Semiconductor

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

Legal Disclaimer

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:

telinkcnsales@telink-semi.com

telinkcnsupport@telink-semi.com

Revision History

Version	Change Description
V1.0.0	Initial release.
V1.0.1	Repair file dependency, and use absolute path for file path
V1.0.2	bdt_gui can pass bin file path parameter
V1.1.0	add vid 826b dev
V1.2.0	add vid 826a dev
V1.3.0	release mac bdt
V1.4.0	Support B92 function
V1.4.2	add some help cmd information
V1.4.3	add B92 usb mode function, B92 1.8v environment
V1.5.2	add B92 secure boot function, unlock flash
V1.6.0	support B930 B95 EVK function
V1.6.4	support tl321x evk function and others Flash unlock

Contents

1 Telink BDT CMD User Guide

The tool was developed and tested on Ubuntu 20.04.3 LTS, 64-bit operating system. Compile and test in macos M, intel series.

Please update the burning EVK firmware version to **V4.4** before using.

2 Introduce

It mainly supports EVK mode, and some chips support USB mode. They are listed in the following table. They are explained in subsequent detailed use cases.

	8 Series	B91	B92	B930	B95
read/write flash (rf, wf)	EVK/USB	EVK	EVK/USB	EVK	EVK
read/write sram (rc, wc)	EVK/USB	EVK	EVK/USB	EVK	EVK
read/wirte analog (ra, wa)	EVK/USB	EVK	EVK/USB	EVK	EVK
download in flash	EVK/USB	EVK	EVK/USB	EVK	EVK
sboot	-	-	EVK	-	-
download in core	EVK/USB	EVK	EVK/USB	EVK	EVK
erase in flash	EVK/USB	EVK	EVK/USB	EVK	EVK
lock/unlock flash(lf, ulf)	-	EVK	EVK	EVK	EVK
check pc (pc)	EVK/USB	EVK	EVK/USB	EVK	EVK
check global parameters (var)	EVK/USB	EVK	EVK/USB	EVK	EVK
reset in flash or sram	EVK/USB	EVK	EVK/USB	EVK	EVK
sws	EVK	EVK	EVK	EVK	EVK
run step stop	EVK	-	-	-	-
ac	EVK	EVK	EVK	EVK	EVK
lsusb					
up					

3 Chip parameters

```
B91 B92 B92_V18 B930 B95
B80 B85 B87 B89_A1 8232 8266 8267 8269 8366 8368 8367_i 8367_e 8369_i 8369_e
```

If B92 1.8v environment is used, select `B92_V18` for this parameter.

4 Command Example

Command Options

```
-u : Indicates usb mode,The default mode is EVK.
-s : The number of bytes read and written, which follows -s. eg: -s 16; -s 1k.
-e : Erasing, used in Flash and core erasing.
-c : Represents core, commonly used reset command.
-i : Specifies the input file followed by the file path, often used to specify the download
 ↪ file. eg: -i /home/8258_gpio.bin.
-o : Specifies the output file, followed by the file path, often used to save read binary data
 ↪ to a file. eg: -o /home/readflash.bin
-p : Represents the printing process, often used for flash operations.
-b,-d : Bus and devid of usb devices. This parameter is required when multiple USB devices
 ↪ exist.
```

Supports the function of USB mode. You can add the -u option after the command.

If there are multiple EVK devices, the VID and PID of EVK devices are the same. You can control a specified EVK device by specifying its **bus, devid**.

If you use usb debugging mode, you also need to specify **bus, devid** to control the device.

```
Example, added after the command. -b:bus -d:devid
./bdt 8258 sws -b 1 -d 1
./bdt 8258 sws -b 1 -d 2

./bdt 8258 sws -b 1 -d 1 -u
./bdt 8258 sws -b 1 -d 2 -u
```

4.1 sws

Set the rate, and detect whether the EVK and the target board connection is normal.

```
# Sets the specified SWS value.
# b0:address 10:Rate parameter value.The first two (b0 10) are set evK SWire CLK values; The
 ↪ last two (B0 10) are the target development board swire CLK values.
./bdt 8258 sws b0 10 b0 10
```

```
# If no value is specified, the default SWS value is B0 10 b0 10.
./bdt 8258 sws
```

Writing SWS values must be followed by SWS command arguments.

4.2 activate

Run this command when the program is in low power mode.

```
./bdt 8258 ac
```

4.3 reset

Restart, the program starts from Flash or SRAM.

```
# Restart the device from the Flash
./bdt 8258 reset

# Restart the device from the Sram
./bdt 8258 reset -c
```

4.4 read/write flash

read flash(rf)

If the read quantity is less than 1KB, the read data will be printed. Larger than 1KB will be saved to the default file.

Default file name example: `save1020-11294102.bin`

```
# Read 16 bytes of flash address 0x00
./bdt 8258 rf 0x00 -s 16
./bdt 8258 rf 0x00 -s 1k

# Reads the data output to the specified file
./bdt 8258 rf 0x00 -s 16 -o readflash.bin
```

write flash(wf)

flash Erasure is required before writing, and the default unit of erasure is 4K.

```
# Write 4 bytes of data to flash 0x00.
./bdt 8258 wf 0x00 01 02 03 04 -s 4

# Erase first, then write data.
./bdt 8258 wf 0x00 01 02 03 04 -s 4 -e

# Write a file to Flash, download function.
# Write files without the -e and -s option.
./bdt 8258 wf 0x00 -i bin/USB_Demo.bin
```

lock flash(lf)

```
./bdt B92 lf addr size(k)
./bdt B92 lf 0 512k
```

unlock flash(ulf)

Flash may be locked during program execution and needs to be reopened during debugging.

```
./bdt B92 ulf
```

4.5 read/write core

read core(rc)

If the read quantity is less than 1KB, the read data will be printed. Larger than 1KB will be saved to the default file.

Default file name example: `save1020-11294102.bin`

```
# Read 16 bytes of sram address 0x40000
./bdt 8258 rc 0x40000 -s 16
./bdt 8258 rc 0x40000 -s 1k

# Reads the data output to the specified file.
./bdt 8258 rc 0x40000 -s 16 -o readsram.bin
```

write core(wc)

```
# Write 4 bytes of data to sram 0x40000
./bdt 8258 wc 0x40000 01 02 03 44 -s 4

# Write a file to sram, download function.
# Write files without the -e and -s option.
./bdt 8258 wc 0x40000 -i bin/USB_Demo.bin
```

4.6 read/wirte analog

read analog(ra)

```
# Read 16 bytes of analog address 0x40000
./bdt 8258 ra 0x00 -s 16
```

write analog(wa)

```
#  Write 4 bytes of data to analog 0x00.
./bdt 8258 wa 0x00 01 02 03 44 -s 4
```

4.7 check pc/parameter

View the PC pointer value, global parameter list (VAR).

You need to configure the. LST file to view the PC pointer value.

```
# Prints program run pointer.
./bdt 8258 pc

# Print the current PC pointer in detail.
./bdt 8258 pc -i USB_PRINT_LOG.lst

# Prints a list of current program parameters (address, length, value).
/bdt 8258 var -i USB_PRINT_LOG.lst
```

4.8 sboot

this feature (security boot) only supports B92 temporarily. Detailed usage and examples can also be viewed using `bdt help sboot` .

```
--mode 0/1
0: normal mode。
1: security boot mode(Check signature), need to be used with the --pkh parameter.
```

```
--crypto 0/1
0: Flash firmware does not encrypt read-write mode.
1: Flash firmware encryption read-write mode, needs to be used with the --rk parameter.
```

```
--pkh /path/to/public_key_file
Download the public key hash to efuse, which is generated by the security boot post tool.
```

```
--rk (16 bytes key)
Root key is used for flash read and write functions.
```

```
--run-code addr-/path/to/bin_file
Download the firmware to the specified flash address, which is the bin file that the MCU
↪ actually runs.
```

```
--run-code-des addr-/path/to/des_bin_file
ownload descriptor information to the specified flash address, des_bin_file is generated by the
↪ security boot post tool; Need to work with --run-des-addr;
```

```
--run-des-addr(3 bytes addr)
Configure the address to efuse, and the value of the parameter addr varies according to the size
↪ of the flash capacity.
1M: f8000
2M: 1f8000
4M: 4f8000
16M:ff8000
```

Example

You can configure the parameters in turn, or you can configure all the parameters at once.

```
1. mode(0) + crypto(1) + rk(16 bytes) + run-code((flash addr)-path(BIN))
  bdt B92 sboot --mode 0
  bdt B92 sboot --crypto 1
  bdt B92 sboot --rk 000102030405060708090a0b0c0d0e0f
  bdt B92 sboot --run-code 0-/path/to/flash_bin

  You can input more than one parameter. (During this process, if one parameter configuration
↪ fails, subsequent parameter configurations will be terminated).\n "
  f.g
    bdt B92 sboot --mode 0 --crypto 1 --rk 000102030405060708090a0b0c0d0e0f --run-code 0-/path/
↪ to/flash_bin
```

```
2. mode(1) + crypto(1) + pkh(public key path) + rk(16 bytes) + run-des-addr(addr) + run-
↪ code((flash addr)-path(BIN)) + run-code-des(addr-path(des_bin))
  f.g
    bdt B92 sboot --mode 1 --crypto 1 --pkh /path/to/public_key_file --rk
↪ 000102030405060708090a0b0c0d0e0f --run-des-addr f8000 --run-code 0-/path/to/flash_bin --run-
↪ code-des f8000-/path/to/flash_bin
```

```
3. mode(1) + crypto(0) +pkh(public key path) + run-des-addr(addr) + run-code((flash addr)-
↪ path(BIN)) + run-code-des(addr-path(des_bin))
  f.g
    bdt B92 sboot --mode 1 --crypto 0 --pkh /path/to/public_key_file --run-des-addr f8000 --run-
↪ code 0-/path/to/flash_bin --run-code-des f8000-/path/to/flash_bin
```

secure-debug enable

```
# If the root key has already been set, there is no need to set it
./bdt b92 sboot --rk 000102030405060708090a0b0c0d0e0f
./bdt b92 sboot --debug-text 000102030405060708090a0b0c0d0e0f
./bdt b92 sboot --secure-debug 1
```

After running the secure-debug command, the sw related debugging commands will immediately become invalid。 Root-key and debug-text need to be used again in subsequent re-enable commands. Please save two strings.

secure-debug re-enable

```
# --re-enable-debug rook_key-debug_text
./bdt b92 sboot --re-enable-debug
↪ 000102030405060708090a0b0c0d0e0f-000102030405060708090a0b0c0d0e0f
```

After running the command successfully, the debugging function continues to take effect until the development board is restarted; After restart, the debugging function will fail again.

read sboot info

you can read somd information about sboot in setting.

```
./bdt b92 sboot --read-info
```

4.9 run stop start stall

Run, stop the program.

```
./bdt 8258 run
./bdt 8258 stop
```

start, stall the program

```
./bdt 8258 start
./bdt 8258 stall
```

4.10 step

Step through the program.

```
./bdt 8258 step
```

4.11 up

Update EVK firmware

-i：Specifies the firmware file path to update.

-v：Query evk version number

```
# The chip used by burning evk is 8266
./bdt 8266 up -i fw/Firmware_v3.4.bin
./bdt 8266 up -i fw/Firmware_v3.4.bin -ev
./bdt 8266 up -ev
```

4.12 lsusb

List connected USB devices.

```
./bdt lsusb

# -v : View usb descriptors
./bdt lsusb -v
```

5 FAQ