



# Telink

## Application Note

# Telink Burning and Debugging Tool(BDT) CMD User Guide CN

AN-22010603-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

© 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](http://www.telink-semi.com)

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

[telinknsales@telink-semi.com](mailto:telinknsales@telink-semi.com)

[telinkcnsupport@telink-semi.com](mailto: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

Acknowledgements	2
Legal Disclaimer	2
Information	2
Revision History	3
<b>1 Telink BDT CMD User Guide</b>	<b>5</b>
<b>2 介绍</b>	<b>5</b>
<b>3 芯片型号参数</b>	<b>5</b>
<b>4 命令使用示例</b>	<b>6</b>
4.1 sws	6
4.2 activate	7
4.3 reset	7
4.4 read/write flash	7
4.5 read/write core	8
4.6 read/wirte analog	8
4.7 sboot	9
4.8 check pc/parameter	10
4.9 run stop	11
4.10 step	11
4.11 up	11
4.12 lsusb	11
4.13 FAQ	12

## 1 Telink BDT CMD User Guide

该工具开发与测试在 Ubuntu 20.04.3 LTS, 64-bit 操作系统. 在 macos X64, aarch 系列进行编译测试。  
使用前请更新 burning EVK 固件版本到 **V4.4**。

## 2 介绍

功能支持, 主要支持 evk 模式, 部分芯片支持 usb 模式. 在下表中列出. 在后续详细用例里面有说明.

	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 芯片型号参数

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](#)

如果使用 B92 1.8v 环境, 该参数选择 `B92_V18`。

## 4 命令使用示例

### 命令选项介绍

- u : 表示使用 usb 模式, 未使用 -u 默认为 evk 模式.
- s : 需要操作的字节数量, 数量跟在 -s 后面. eg: -s 16; -s 1k.
- e : 表示擦除操作, 常用于 flash, core 擦除.
- c : 表示 core, 常用 reset 命令.
- i : 指定输入文件, 后面紧跟文件路径, 常用于指定下载文件. eg: -i /home/8258\_gpio.bin.
- o : 指定输出文件, 后面紧跟文件路径, 常用于将读取二进制数据保存到文件. eg: -o /home/readflash.bin
- p : 表示打印具体过程, 常用于 flash 操作.
- b, -d : usb 设备的 bus, devid, 存在多个 usb 设备时候需要使用该参数.

支持 usb 模式的功能, 在命令后面添加 -u 选项即可.

假如存在多个 evk 设备, 由于 evk 设备 vid、pid 都是相同的。可以通过指定 evk 设备的 **bus**, **devid** 控制指定设备。

假如使用 usb 模式的调试方式, 也需要通过指定 **bus**, **devid** 的方式控制设备。

示例, 在命令的后面添加 -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

设置通讯速率, 以及检测 evk 于目标板连接是否正常.

# 设置指定的 sws 值.

# b0: 地址 10: 速率参数值. 前两个 (b0 10) 为设置 evk swire clk 值; 后两个 (b0 10) 是目标开发板 swire clk 值.

```
./bdt 8258 sws b0 10 b0 10
```

# 如果没有指定值, sws 值默认为 b0 10 b0 10.

```
./bdt 8258 sws
```

写入 sws 值必须跟在 sws 命令参数后面.

## 4.2 activate

唤醒设备，程序处于深度睡眠的时候，使用该命令。

```
./bdt 8258 ac
```

## 4.3 reset

重新启动，程序从 flash 或者 sram 启动。

**# 重启后从 flash 启动**

```
./bdt 8258 reset
```

**# 重启后从 sram 启动**

```
./bdt 8258 reset -c
```

## 4.4 read/write flash

### read flash(rf)

读取数量小于 1kb，读取数据将打印；大于 1kb 将会保存到默认文件。

默认文件名称实例：`save1020-11294102.bin`

**# 读取 flash 地址 0x00 16 字节数据**

```
./bdt 8258 rf 0x00 -s 16
```

```
./bdt 8258 rf 0x00 -s 1k
```

**# 将读取数据输出到制定文件 readflash.bin**

```
./bdt 8258 rf 0x00 -s 16 -o readflash.bin
```

### write flash(wf)

flash 在写入之前需要先进行擦除，擦除的默认单位是 4k。

**# 写入 4 字节数据到 flash 0x00.**

```
./bdt 8258 wf 0x00 01 02 03 04 -s 4
```

**# 先擦除，再写入数据.**

```
./bdt 8258 wf 0x00 01 02 03 04 -s 4 -e
```

**# 写入一个文件到 flash，下载功能.**

**# 写入文件不用选项 -e -s.**

```
./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 在程序运行过程中可能会被锁定，在调试的时候需要重新打开。

```
./bdt B92 ulf
```

## 4.5 read/write core

#### read core(rc)

读取数量小于 1kb, 读取数据将打印; 大于 1kb 不将会保存到默认文件.

默认文件名称实例: `save1020-11294102.bin`

**# 读取 sram 地址 0x40000 16 字节数据.**

```
./bdt 8258 rc 0x40000 -s 16
./bdt 8258 rc 0x40000 -s 1k
```

**# 将读取数据输出到制定文件 readsram.bin.**

```
./bdt 8258 rc 0x40000 -s 16 -o readsram.bin
```

#### write core(wc)

**# 写入 4 字节数据到 sram 0x40000**

```
./bdt 8258 wc 0x40000 01 02 03 44 -s 4
```

**# 写入一个文件到 sram, 下载功能**

**# 写入文件不用选项 -e -s**

```
./bdt 8258 wc 0x40000 -i bin/USB_Demo.bin
```

## 4.6 read/wirte analog

#### read analog(ra)

**# 读取 analog 数据, 起始地址 0x00 16 字节.**

```
./bdt 8258 ra 0x00 -s 16
```

#### write analog(wa)

**# 写入 4 字节数据到 analog 0x00.**

```
./bdt 8258 wa 0x00 01 02 03 44 -s 4
```

## 4.7 sbboot

目前该功能 (security boot) 暂时只支持 b92，详细用法、例子也可使用 `bdt help sbboot` 查看。

`--mode 0/1`

0: 普通模式。

1: security boot 模式 (验签)，需要配合 `--pkh` 参数使用。

`--crypto 0/1`

0: flash 固件不加密读写模式。

1: flash 固件加密读写模式，需要配合 `--rk` 参数使用。

`--pkh /path/to/public_key_file`

下载 public key hash 到 efuse，该文件由 security boot post 工具可生成。

`--rk (16 bytes key)`

root key 用于 flash 读写功能。

`--run-code addr-/path/to/bin_file`

下载固件到指定 flash 地址，该固件为 mcu 实际运行的 bin 文件。

`--run-code-des addr-/path/to/des_bin_file`

下载描述符信息到指定 flash 地址，该 des\_bin\_file 文件由 security boot post 工具生成；需要配合 `--run-des-addr` 使用；

`--run-des-addr(3 bytes addr)`

配置地址到 efuse，该参数 addr 根据 flash 容量大小变化。

1M: f8000

2M: 1f8000

4M: 4f8000

16M: ff8000

### 示例

可以依次配置参数，也可以一次性配置所有的参数。

1. mode(0) + crypto(1) + rk(16 bytes) + run-code((flash addr)-path(BIN))

bdt B92 sbboot --mode 0

bdt B92 sbboot --crypto 1

bdt B92 sbboot --rk 000102030405060708090a0b0c0d0e0f

bdt B92 sbboot --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

# 如果已经设置 root key, 则不用设置

```
./bdt b92 sboot --rk 000102030405060708090a0b0c0d0e0f
./bdt b92 sboot --debug-text 000102030405060708090a0b0c0d0e0f
./bdt b92 sboot --secure-debug 1
```

运行 -secure-debug 命令后, sw 相关调试命令会立即失效。root-key 与 debug-text 在后续 re-enable 命令需  
继续使用, 请保存 2 个字符串。

### secure-debug re-enable

# --re-enable-debug rook\_key-debug\_text

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

成功运行该命令后, 开发板没有重启前, 调试功能继续生效; 重启后, 调试功能将再次失效。

### read sboot info

可以运行如下命令读取当前关于 sboot 的设置信息。

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

## 4.8 check pc/parameter

查看 pc 指针值, 全局参数表 (var).

需要查看 pc 指针值, 参数具体含义, 需要配置.lst 文件

# 打印当前 pc 指针值.

```
./bdt 8258 pc
```

# 打印当前 pc 指针值, 详细说明.

```
./bdt 8258 pc -i USB_PRINT_LOG.lst
```

# 打印当前程序参数列表 (地址, 长度, 值).

```
/bdt 8258 var -i USB_PRINT_LOG.lst
```

## 4.9 run stop

运行, 停止程序执行

# 开始运行.

```
./bdt 8258 run
```

# 停止运行.

```
./bdt 8258 stop
```

## 4.10 step

单步运行程序.

```
./bdt 8258 step
```

## 4.11 up

更新 evk 固件, usb 连接 evk 设备

-i : 指定需要更新的固件文件夹

-v : 查询当前 evk 版本号

# evk 芯片为 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

查看已经电脑 usb 设备列表, -v 选项可以查看 usb 描述符字符串。

```
./bdt lsusb  
./bdt lsusb -v
```

## 4.13 FAQ

Telink Semiconductor