

Table of contents

Table of contents	1
Function Description	2
Platform.....	2
Hardware Platform	2
Software Platform	2
Burning Step	3
Step1	3
Step2	4
step3.....	4
Code Structure	5
System initialization	5
RF configuration	5
ESB packet setting	6
Result analysis	6

Function Description

Before reading this document, you need to know what Enhance Shock Burst (ESB) is.

This document is used to tell users about the use of ESB PTX mode and there is a simple example in this document. Through this document, user can learn how to send data in ESB PTX mode.

This document needs to be used with the ESB PTX mode demo (esb_ptx demo). The function of the ESB PTX mode demo is to send data to receiver which in the ESB PRX mode.

Platform

if you want to send data with ESB PTX mode, you need to configure the ESB PTX mode environment as follows.

Hardware Platform

- Telink TLSR8258 EVK(C1T139A30_V1.2)
- Telink Burning EVK(V1.0.0.0)

Note: You need an external antenna to work normally.



Figure 1 Telink burning EVK



Figure 2 Telink TLSR8258 EVK

Software Platform

- Telink Burning and Debugging Tool
- ESB PTX software (ESB_PTX.bin)

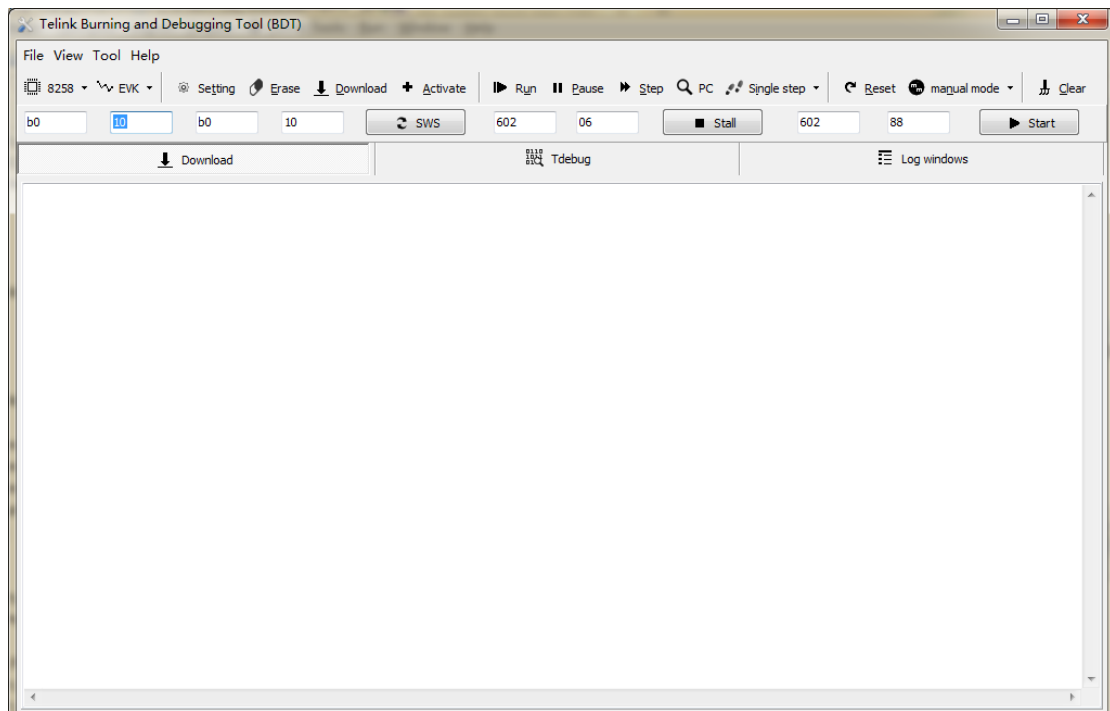


Figure 3 Telink Burning and Debugging Tool

Burning Step

Step1

Connect Telink Burning EVK and Telink TLSR8258 EVK with an usb cable, then connect with PC.
 Note: you also need connect Telink Burning EVK SWM pin and Telink TLSR8258 EVK SWS pin with a wire.

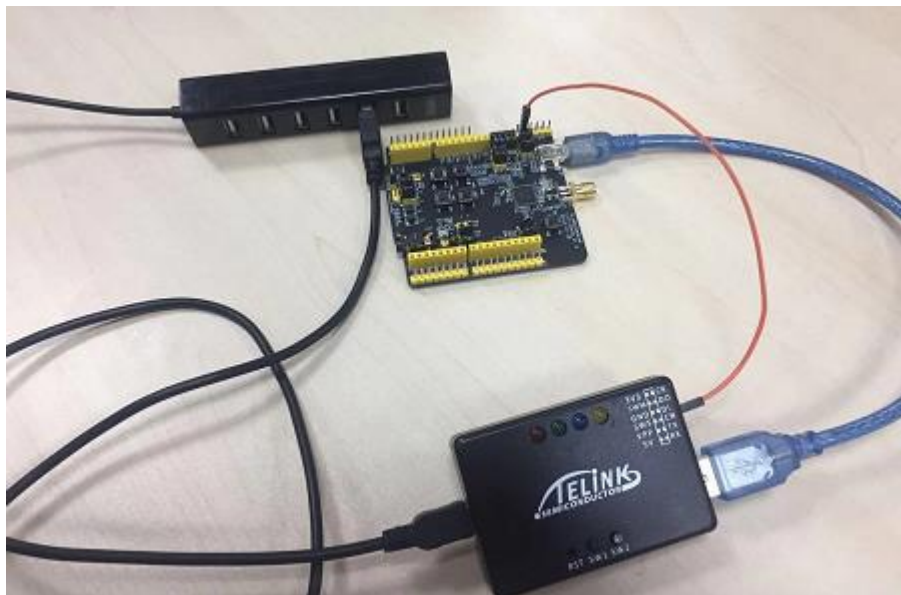


Figure 4 Connect way

Step2

Open Telink Burning and Debugging Tool (BDT) and select 8258 chip, then click on SWS. If BDT shows “no evk device!”, you should check whether the hardware connection is correct. If BDT shows “Swire ok!” that means hardware is ok.

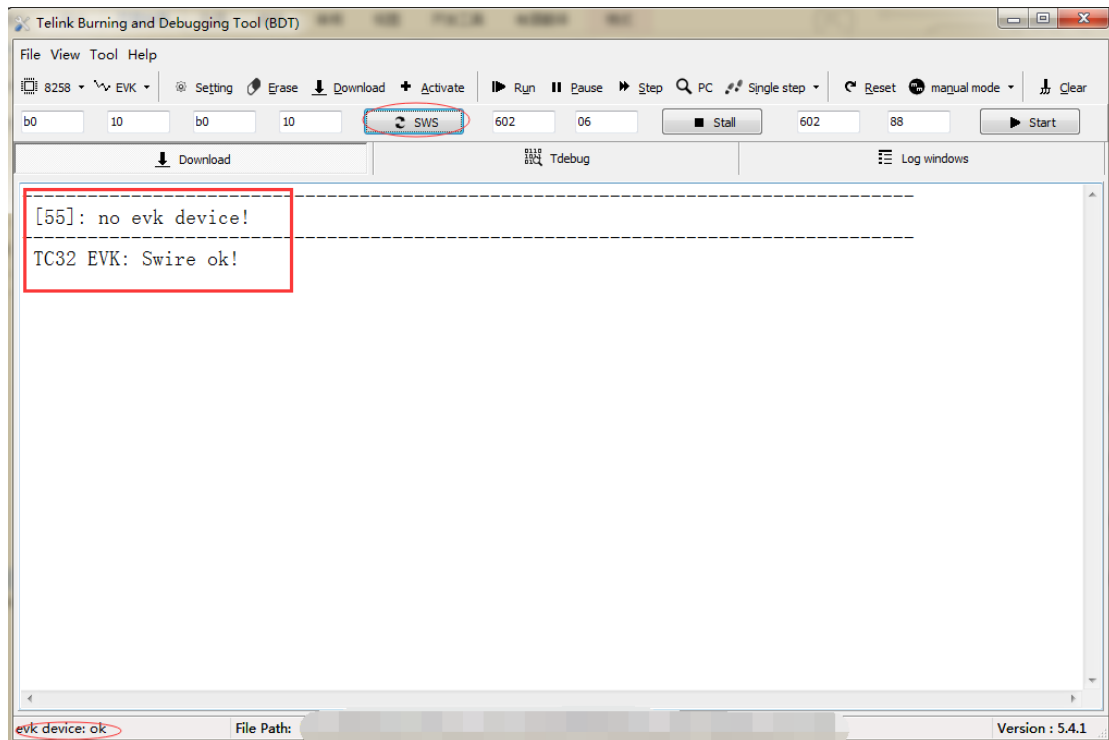


Figure 5 BDT information

step3

Burning the program to the 8258 TLSR8258 EVK. You should click on "file" button to select your target bin file, then click on “Download” button, it will show download information correctly. Whereas, if the BDT shows “Swire err!” check your hardware or click on "Active" button, it may solves your problem!

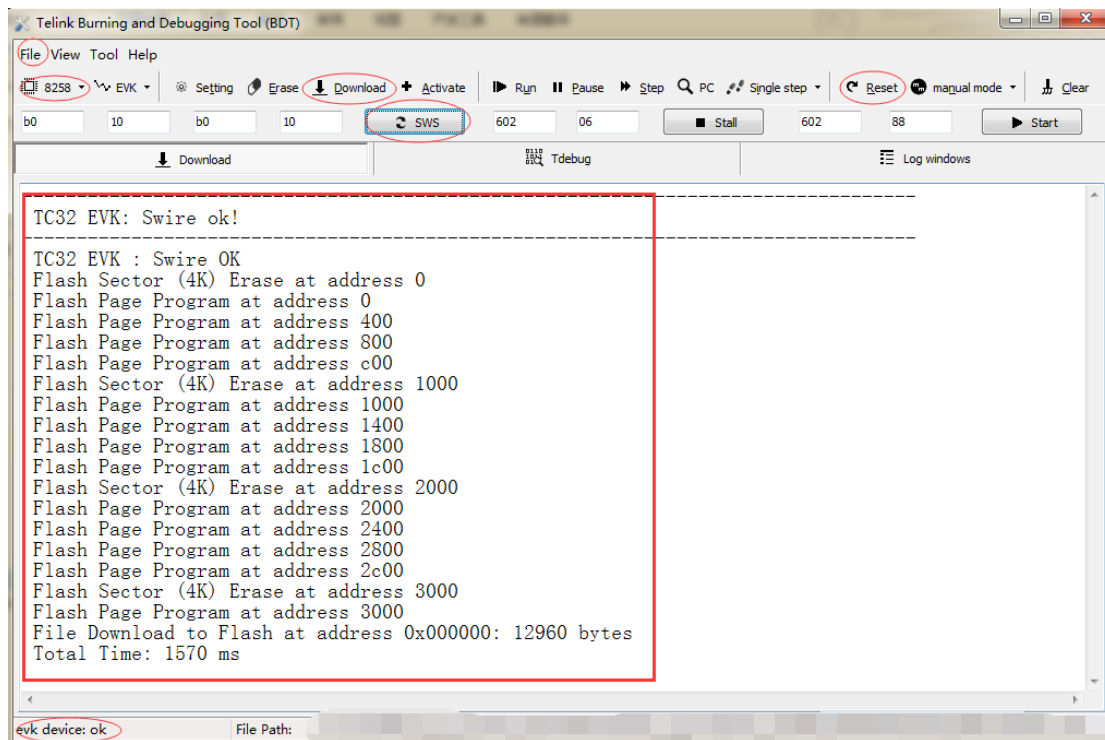


Figure 6 BDT information

Code Structure

System initialization

Include cup initialization, clock initialization and user initialization.

```
cpu_wakeup_init();
clock_init(SYS_CLK_24M_Crystal);
User_Init(2);
```

RF configuration

You need to set the basic parameter of RF in ESB PTX mode before you send the data.

```
//rf configuration
ESB_Init(ESB_BITRATE_2MBPS);
ESB_SetOutputPower(ESB_RF_POWER_ODBM);
ESB_SetAddressWidth(ADDRESS_WIDTH_5BYTES);
ESB_ClosePipe(ESB_PIPE_ALL);

#if PTX_CHANNEL == 0
//unsigned char tx_address[3] = {0xe7,0xe7,0xe7};
unsigned char tx_address[5] = {0xe7,0xe7,0xe7,0xe7,0xe7}; //{0xaa,0xbb,0xcc,0xdd,0xee};
ESB_SetAddress(ESB_PIPE0, tx_address);
ESB_OpenPipe(ESB_PIPE0);
ESB_SetTXPipe(ESB_PIPE0);
#endif
```

```

ESB_ModeSet(ESB_MODE_PTX);
ESB_SetRFChannel(chn);
ESB_SetAutoRetry(0,150); //5,150
ESB_RxTimeoutSet(500);
ESB_RxSettleSet(80);
ESB_TxSettleSet(149);

```

API *ESB_Init* is used for setting RF bitrate, we provide 4 bitrate options, 2Mbps, 1Mbps.

API *ESB_SetOutputPower* is used for setting RF emission power.

API *ESB_SetAddressWidth* is used for setting address width, you can select 3-5 bytes address width to set.

API *ESB_SetAddress* is used for setting address and pipe. You need API *ESB_OpenPipe* to open pipe after setting address and pipe.

API *ESB_SetTXPipe* is used for launching pipe.

API *ESB_ModeSet* is used to set ESB mode, we have two mode to choose, PTX mode and PRX mode.

API *ESB_SetRFChannel* is used for setting RF channel, notice that the unit of channel is 500 kHz.

API *ESB_SetAutoRetry* is used for automatic retransmission when PTX lose an ack, the first parameter is used to set the number of retransmissions and another parameter is used to set the retransmissions interval.

Other APIs are used to set up the RF basic configuration.

Note: This version of 8258 only has pipe0.

ESB packet setting

If you want to send data, first, you need to set pipe, data and data length with API *ESB_WriteTxPayload*; finally, use API *ESB_PTXTrig* to send data.

```

tmp = ESB_WriteTxPayload(PTX_CHANNEL, tx_data, tx_len);
ESB_PTXTrig();

```

If you want to receive data from PRX, when an ack is received, you must save data with API *ESB_ReadRxPayload* immediately.

Result analysis

If the data is sent successfully and an ack from PRX is received, LED D2 will start blinking per second.

To get some information when system running, you can debug with BDT.

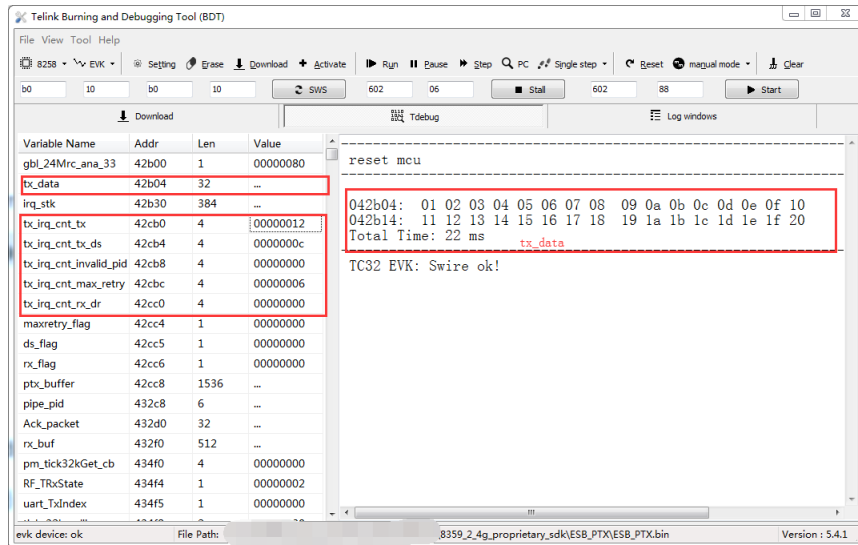


Figure 7 PTX debug information

From Figure 7, you can know what data you have sent, how many times you have sent data, how many times you have received an ack and so on.