Table of contents

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 PRX mode and there is a simple example in this document. Through this document, you can learn how to receive data in ESB PRX mode.

This document needs to be used with the ESB PRX mode demo (esb_prx demo). The function of the ESB PRX mode demo is to receive data from the transmitter which in the ESB PTX mode.

Platform

if you want to receive data in ESB PRX mode, you need to configure the ESB PRX 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_PRX.bin)

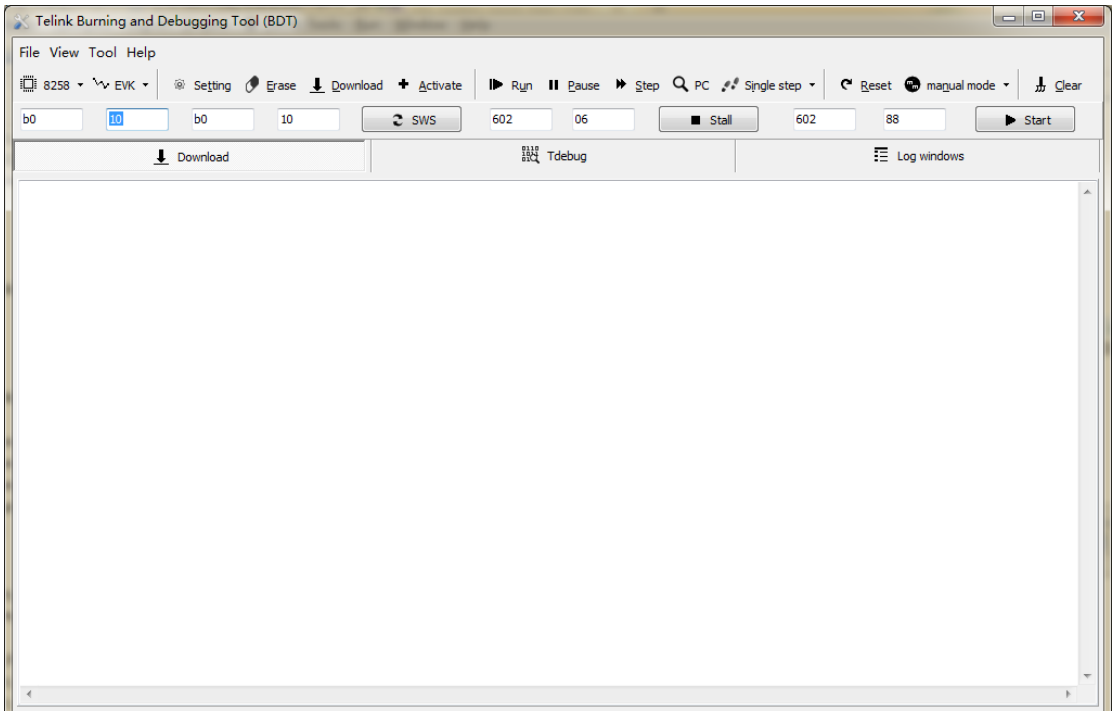

Figure 3 Telink Burning and Debugging Tool

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.

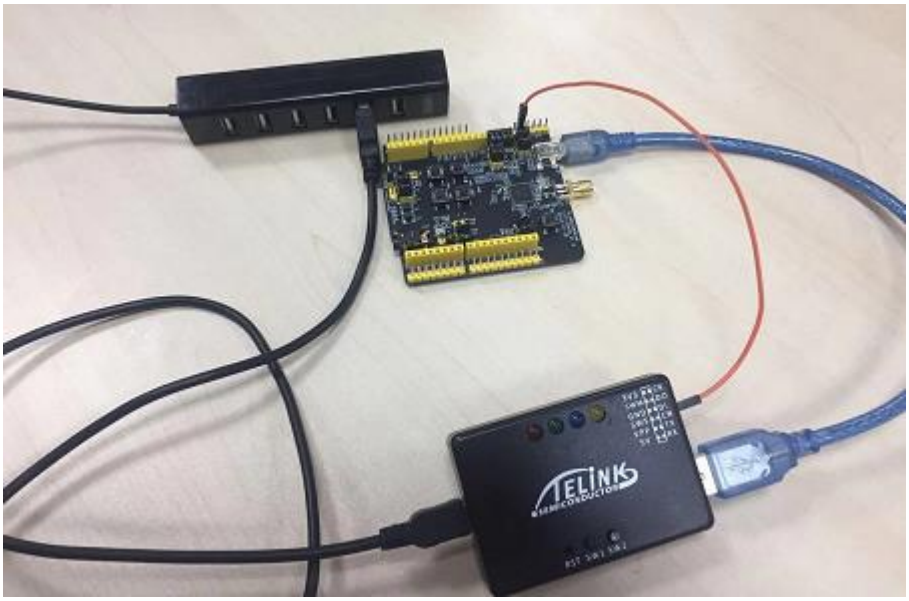

Figure 4 Connect way

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.

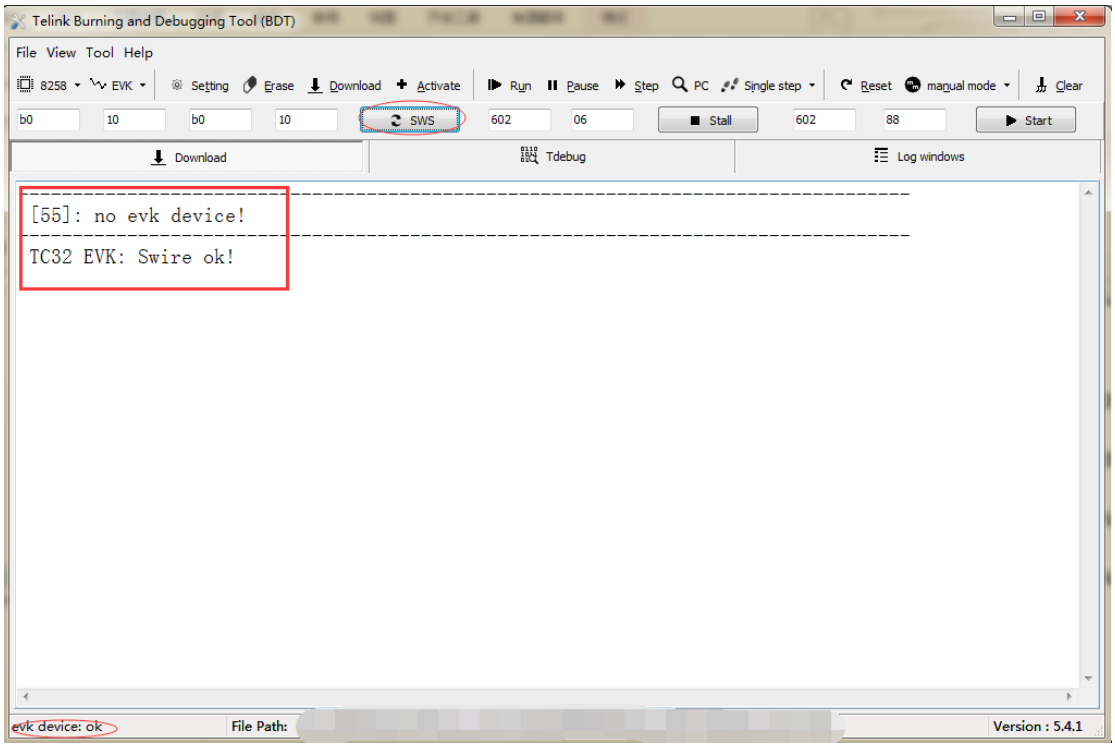

Figure 5 BDT information

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! ”, you need check your hardware or click on "Active" button ,it may solves your problem!

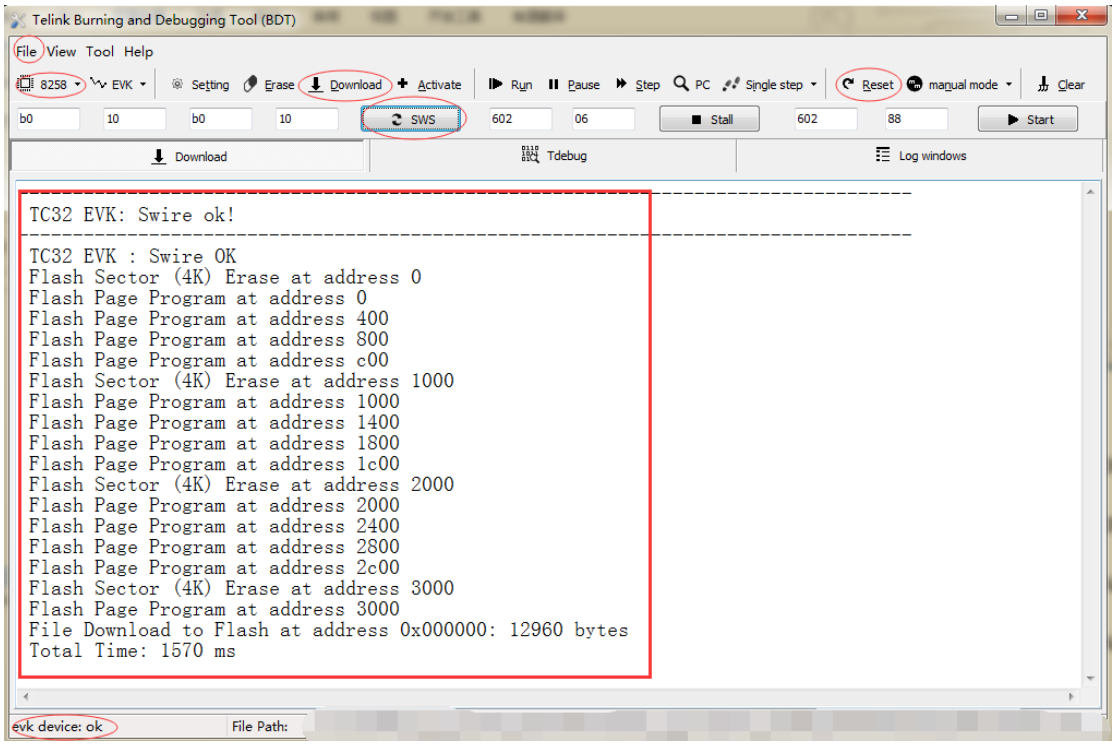

Figure 6 BDT information

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

We need to set the basic parameter of RF before we receive the data in ESB PRX mode

```
    //rf configuration
    ESB_Init(ESB_BITRATE_2MBPS);
    ESB_SetOutputPower(ESB_RF_POWER_0DBM);
    ESB_SetAddressWidth(ADDRESS_WIDTH_5BYTES);
    ESB_ClosePipe(ESB_PIPE_ALL);
#if PTX_CHANNEL == 0
    unsigned char rx_address[5] = {0xe7,0xe7,0xe7,0xe7,0xe7};
    ESB_SetAddress(ESB_PIPE0,rx_address);
    ESB_OpenPipe(ESB_PIPE0);
#endif
```

```
ESB_ModeSet(ESB_MODE_PRX);
ESB_SetRFChannel(chn);
ESB_TxSettleSet(149);
```

API *ESB_Init* is used for setting RF bitrate, we provide 2 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.

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

Note: This version of 8258 only has pipe0.

ESB packet setting

Use API ESB_PRXTrig to start receiving data when rf configuration is complete.

```
ESB_PRXTrig();
```

API ESB_ReadRxPayload is used for receiving data. If you want to send data to PTX, you need to set pipe, data and data length immediately after receiving the data using the API ESB_WriteAckPayload.

```
ESB_ReadRxPayload(&rx_data);
while(!ESB_TxFifoEmpty(0));
ESB_WriteAckPayload(ESB_PIPE0, ack_payload, ack_payload_length);
```

Result analysis

If the data are received, LED D2 will start blinking per packet.

We can debug with BDT.

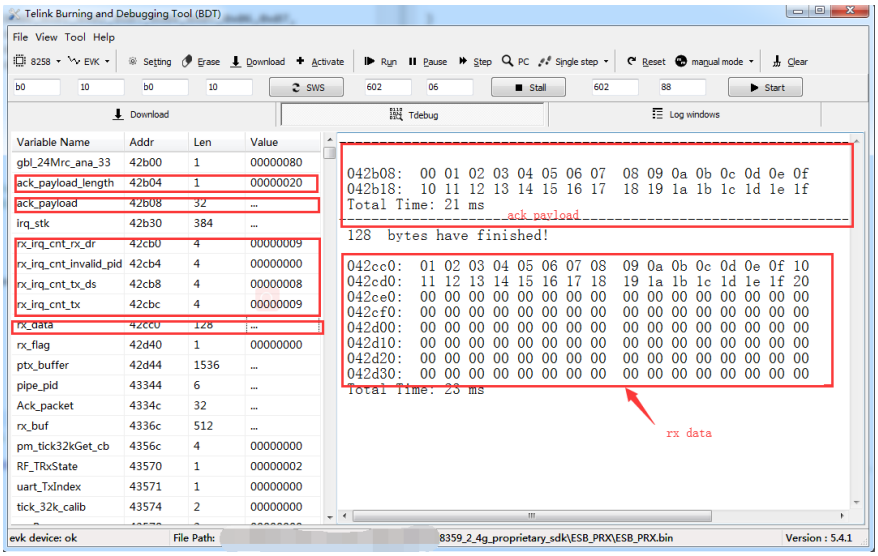

Figure 7 PRX debug information

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