How to Use Trace32 to Run U-boot in the i.MX6Q SABRE Platform

The document describes how to learn System Boot Flow of Linux by code using Trace32. The hardware platform is i.MX6Q SABRE and the software in PC is Trace32.

Contents

1. Introduction

By building a Yocto Project Image, a lot of system boot files were generated. These files can be used to learn system boot flow by Linux OS which running in the i.MX6Q SABRE platform.

This document is based on following environment.

Hardwares:

- Yocto 4.1.15 GA on i.MX6Q SABRE
- Power Debug USB 2.0(JTAG)
- ARM Debug Cable V4d
- PC(Windows 7)

Softwares:

- Trace32 in Windows7
- PuTTY v6.03

Files:

- U-boot
- System Boot files
- Attach Script

2. Hardware Connection

Figure 1 shows the connection among PC, JTAG and i.MX6Q SABRE.

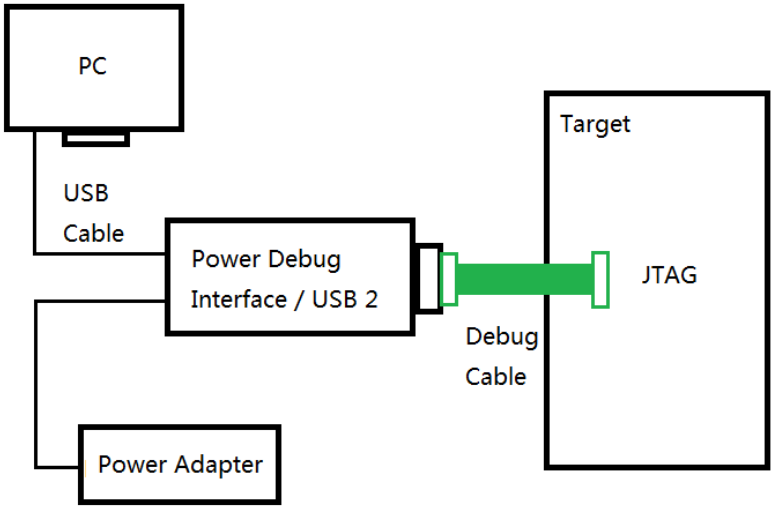

Figure 1. Hardware connection

But if you want to keep watch over the information during the SABRE's booting flow. A USB cable should be connect between PC and Target(SABRE) like Figure 2.

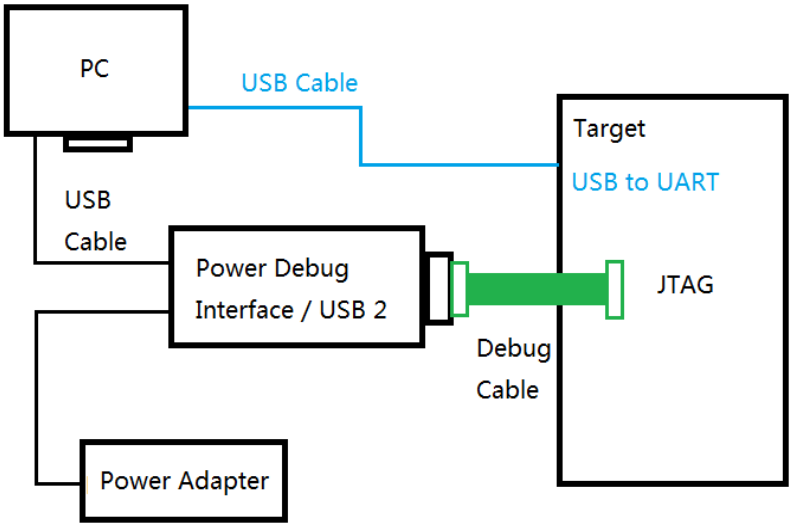

Figure 2. Hardware connection with Serial Port

3. Serial Connection Setup

To setup the serial connection, the first step is to verify USB Serial Port number. Right click on *Computer*, select *Manage* and then click on the *Device Manager* on left tree list as Figure 3 following. So the port number is COM4.

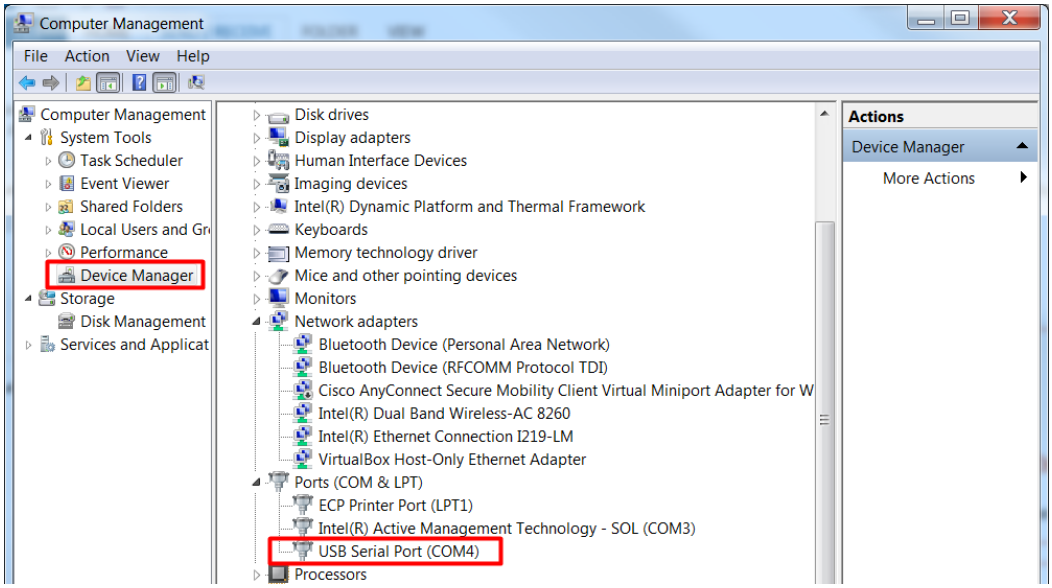

Figure 3. Verifying USB Serial Port number

Several software can be used to make a connection(PuTTY, Hyper Terminal etc.). PuTTY is used as an example in this paper. The Serial line is COM4 and Speed is 115200 as Figure 4. Then click the *Open* button.

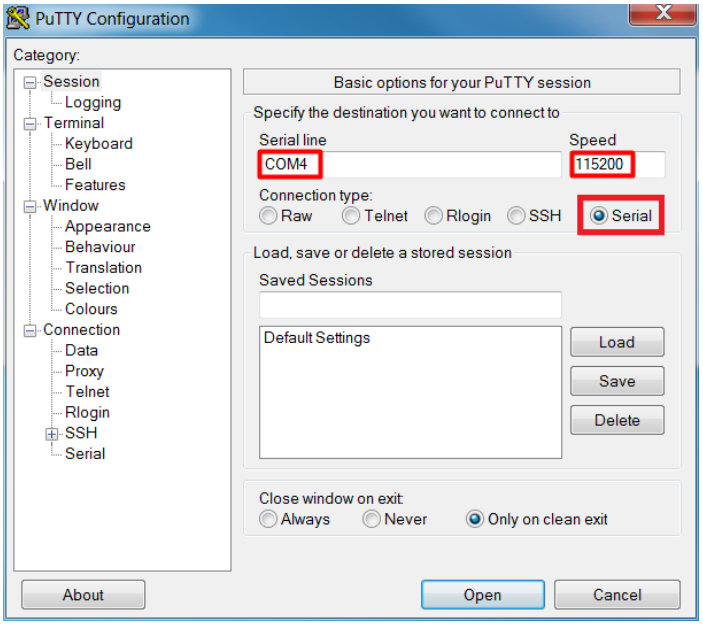

Figure 4. Set PuTTY configuration of Serial

Figure 5. shows the interface of serial console.

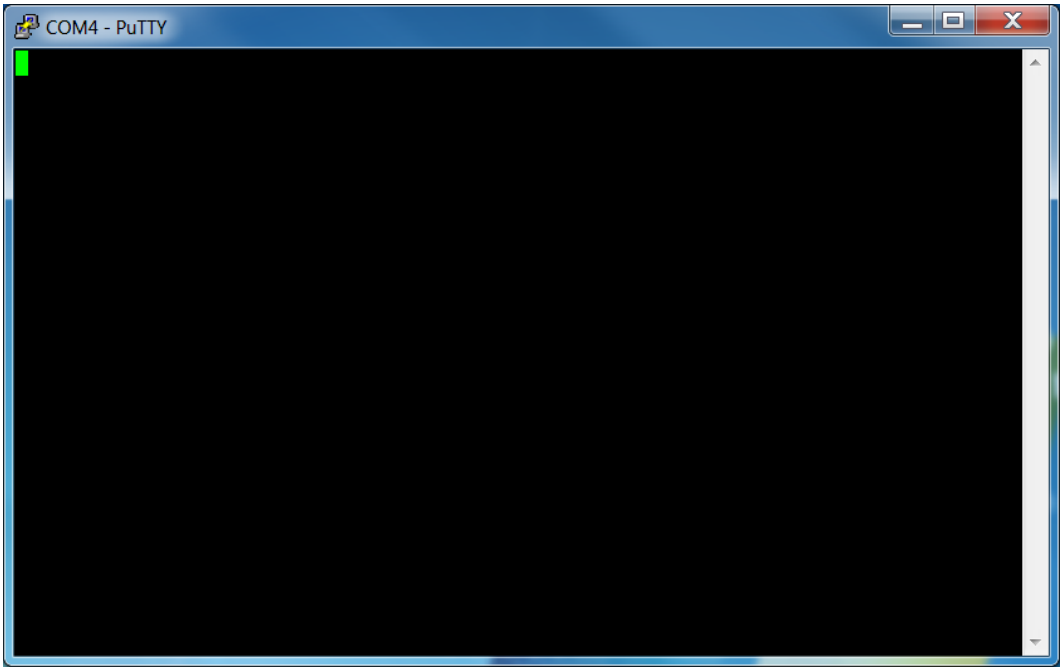

Figure 5. Serial console

Then power on the target(i.MX6Q SABRE) and several board information is outputted in the console like Figure 6.

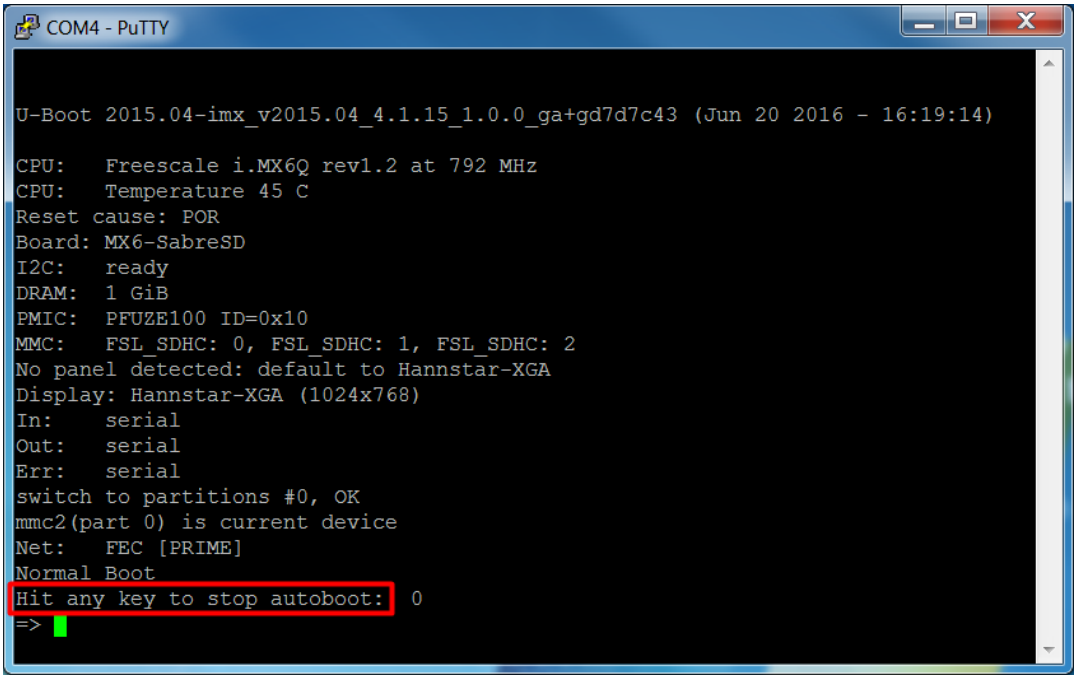

Figure 6. Board information

And when the information *Hit any key to stop autoboot* appears, you can press any key on your keyboard to avoid the autoboot for boot flow control. The time for pressing key is very slow, so

you may try several times. It is inconvenient for debugging, so once you succeed to press key in time, you can enter the command in the console as Figure 7 shows. The first command is to set the boot delay time as 5s and you can change the time parameter. The second is to save the changes.

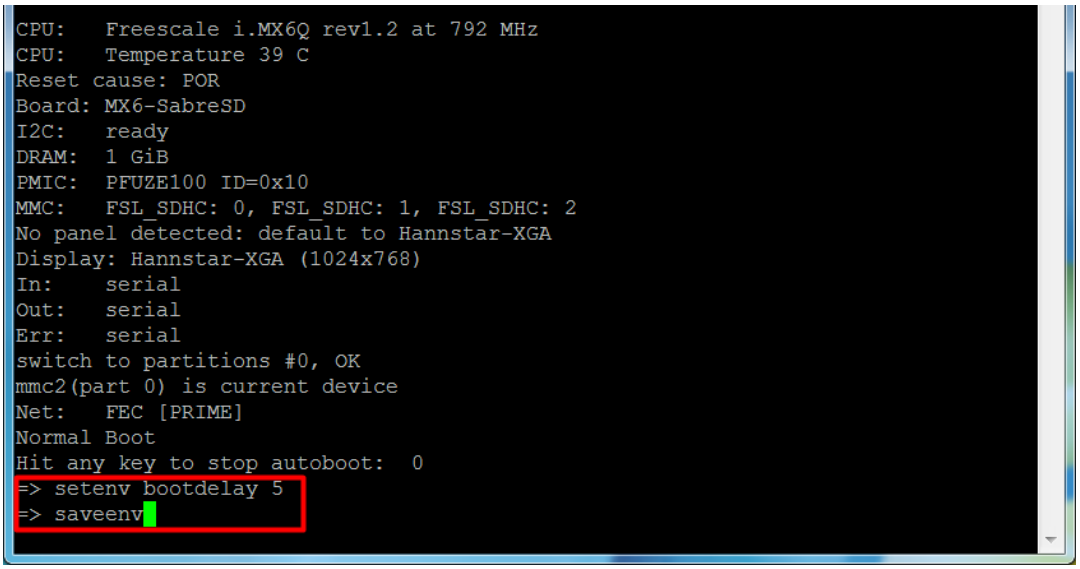

Figure 7. Set the boot delay time

4. U-boot Directory Setup

As you build the image by following the instructions in the *Freescale Yocto Project User's Guide*. A large number of files and directories have been generated. The two folders as Figure 8 should be copied to specific path in PC for u-boot debugging. This document using the x11 image as the section 5.6.1 in the *Freescale Yocto Project User's Guide*. You should change directory as Figure 8 shows in the top.

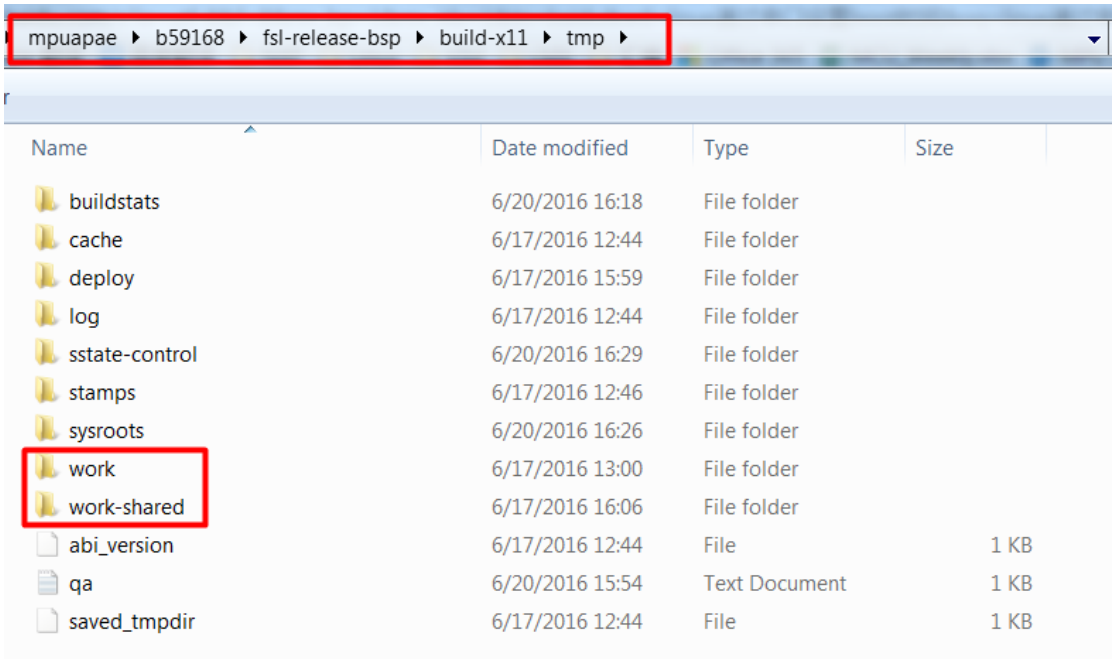

Figure 8. Two folders preparation for debugging

Figure 9 shows the specific path in PC because the u-boot file records the path information when building image.

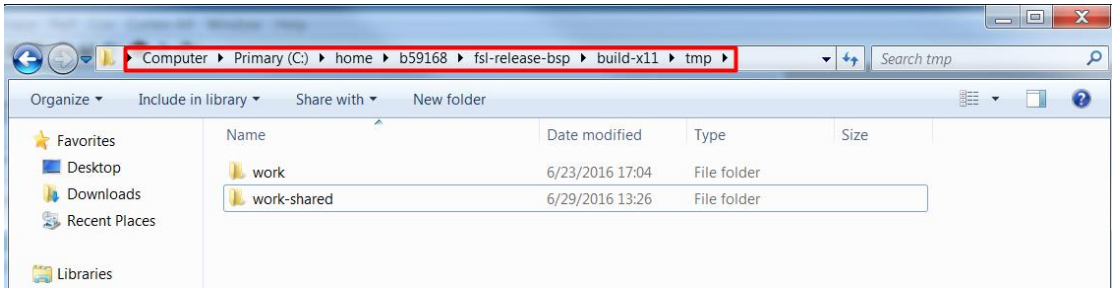

Figure 9. Specific files path in PC

Moreover, Trace32 and attach script should be downloaded from the server. Figure 10 shows the location in the network server. The two zipped files should be downloaded to your own PC.

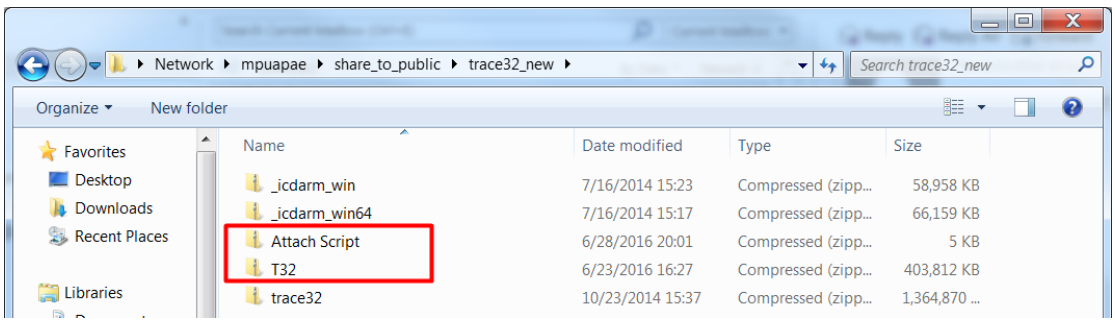

Figure 10. Files location in network server

Figure 11 shows the u-boot file's location after building image.

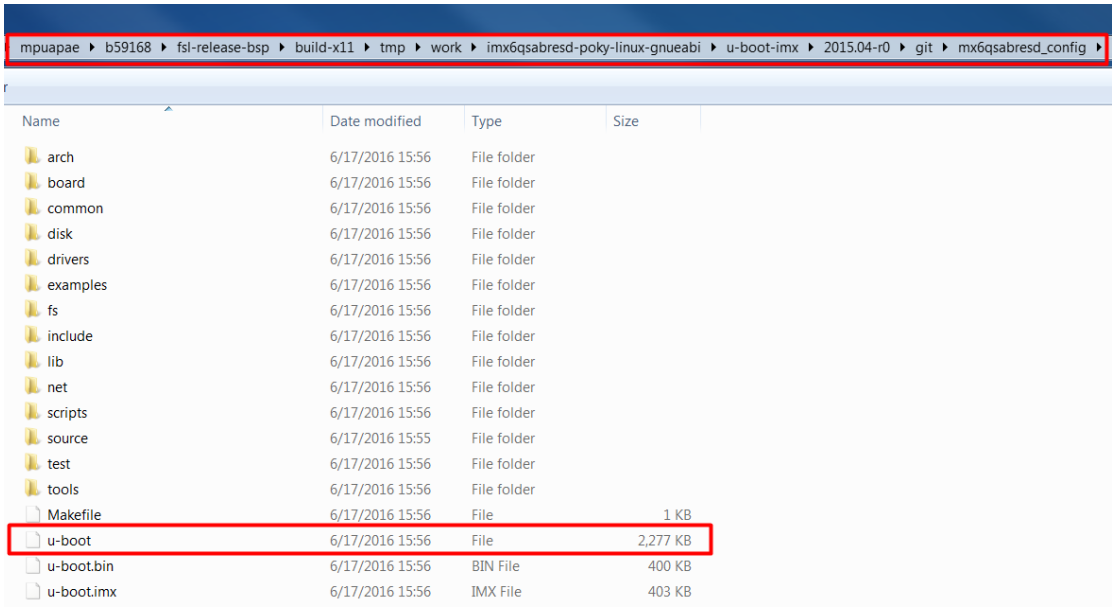

Figure 11. U-boot file location

Unzip the *Attach Script* file and copy the u-boot file to the folder as Figure 12.

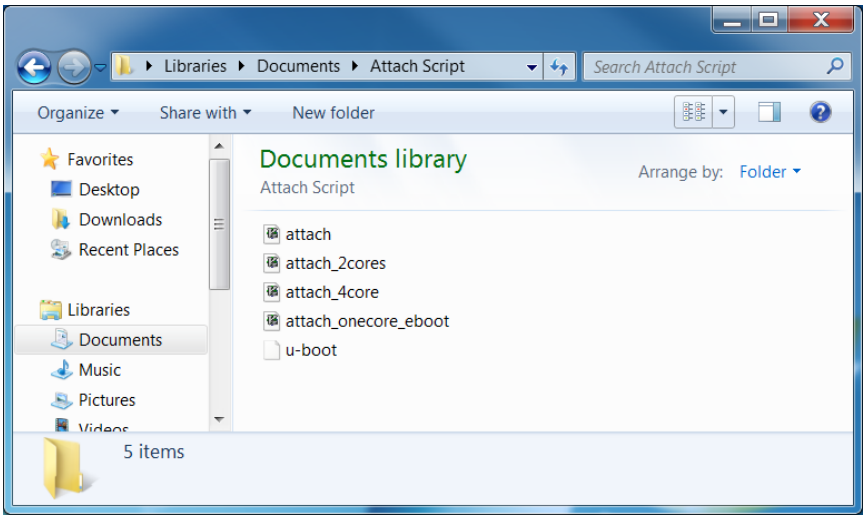

Figure 12. Copy the u-boot file to the same folder as attach script

5. Trace32 Installation & U-boot Debugging

Unzipped the T32 zipped file in Figure 10. Place the folder in a directory, for example, C:\trace32_new. Then the main program is *t32marm* in the path as Figure13.

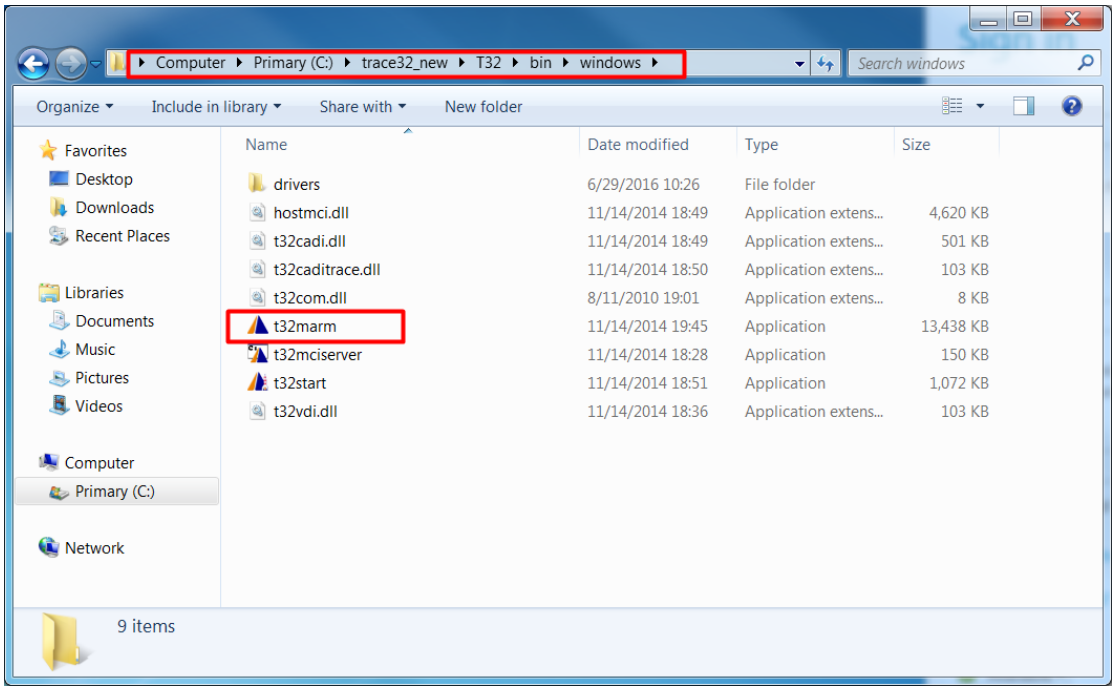

Figure 13. The location of main program

But if you open it immediately, you will meet a warning. Figure 14 is the warning window.

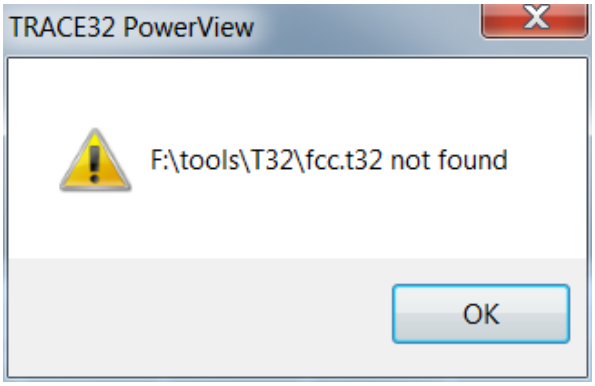

Figure 14. The warning

Because the environment variables in config file is specific, so the environment variables should be modified. Figure 15 is the location of config file.

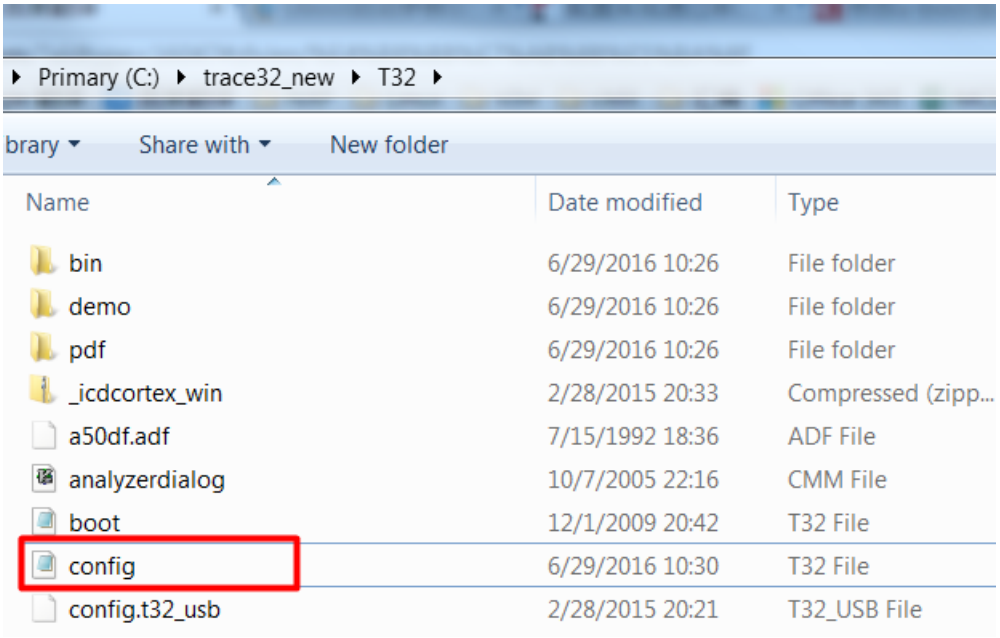

Figure 15. Config location

Double click to open the config file with the notepad. Figure 16 is the content.

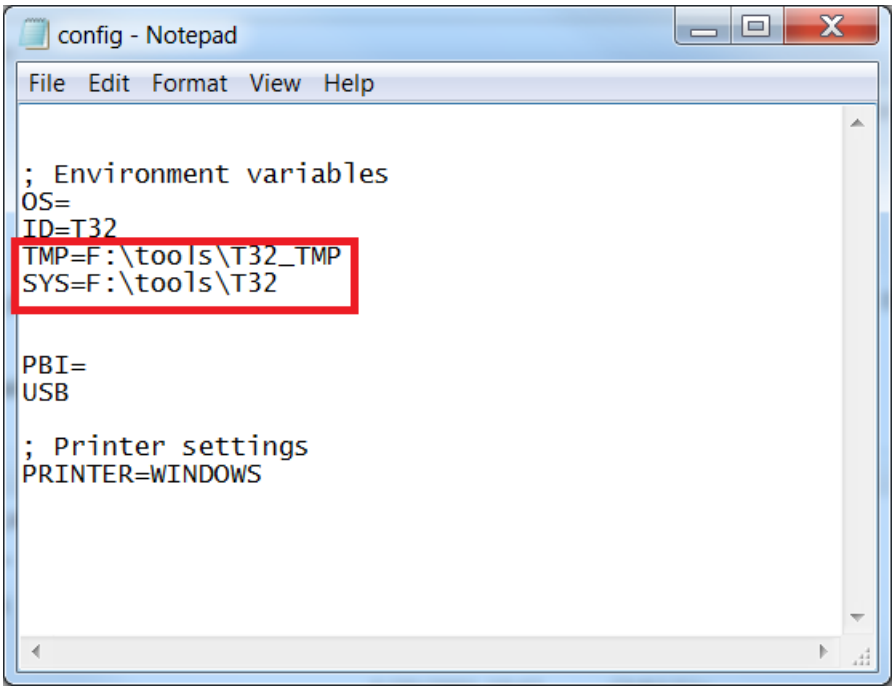

Figure 16. Config content

The variables should be modified is in the red rectangular box in Figure 16. And after modified, the text is as the Figure 17 shows.

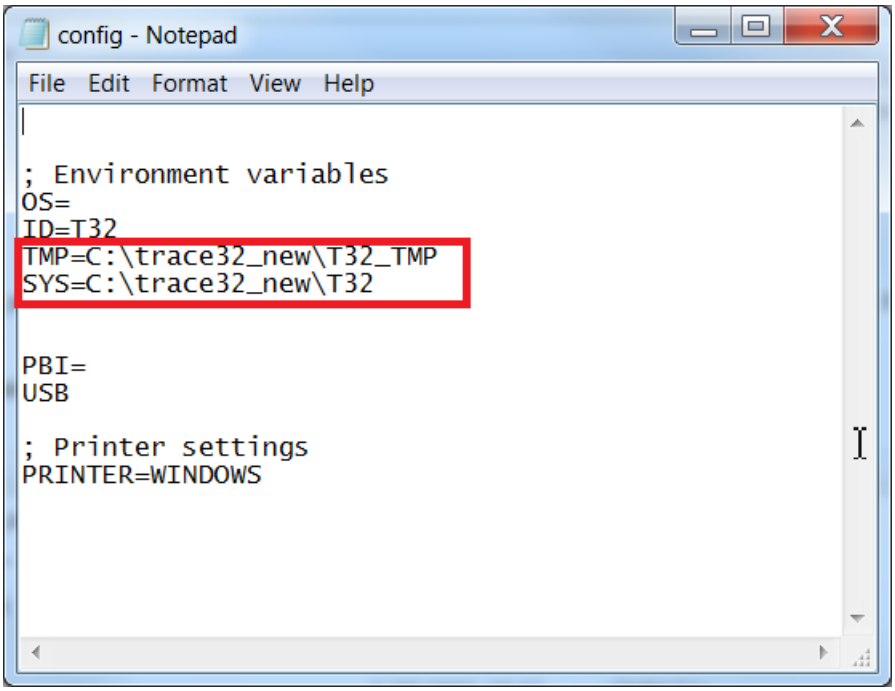

Figure 17. Config content after modified

Besides, you should new a folder named *T32_TMP* in the *trace32_new* folder directory as following Figure 18.

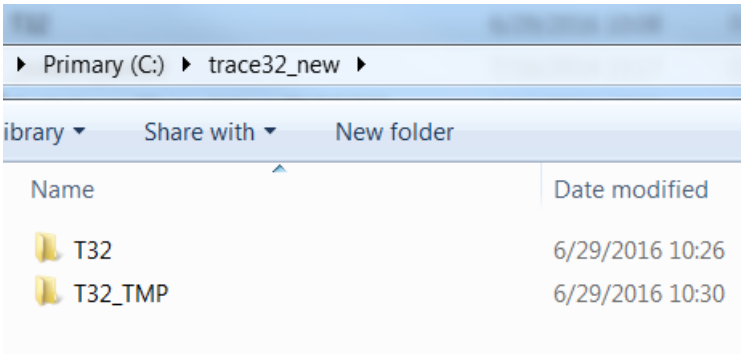

Figure 18. New a folder *T32_TMP*

Make sure hardware connection is right. The Debug USB and i.MX6Q SABRE are powered on and autoboot mode of i.MX6Q SABRE is stopped in the PuTTY.

Open the *t32marm* as Figure 13 shows. Figure 19 is the interface of Trace32.

Tips: you can create a shortcut of the application on desktop.

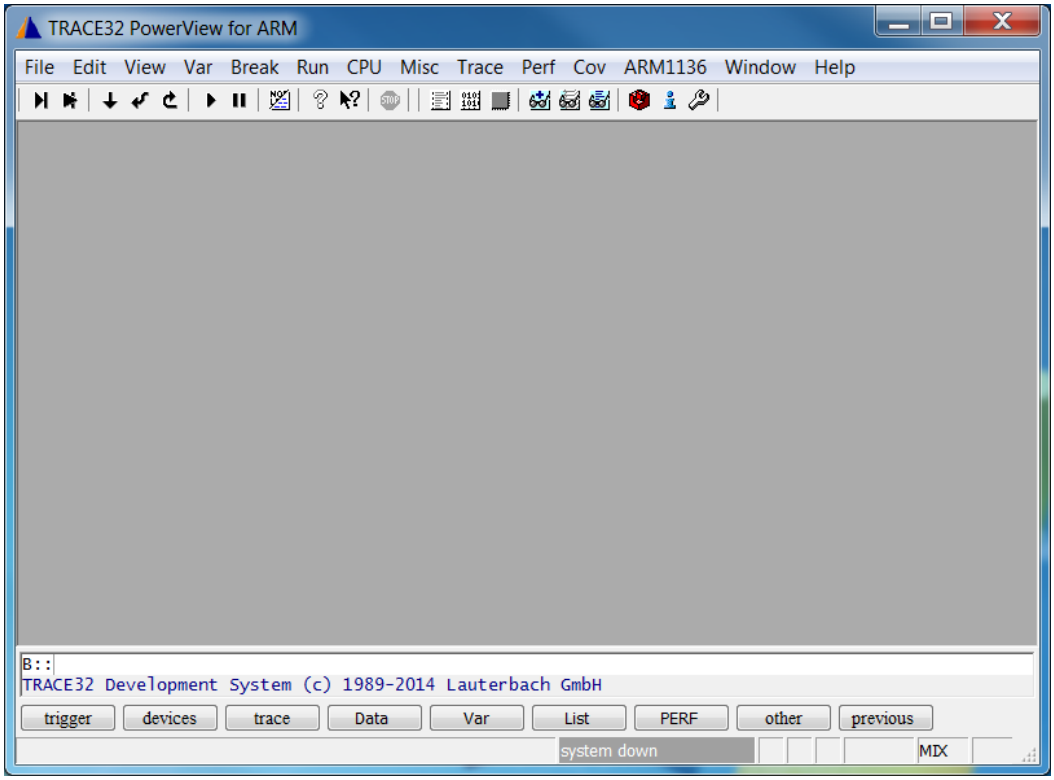

Figure 19. The interface of Trace32

Click *File* in the menu bar and select the *Run Batchfile...* to load a attach script in Figure 20.

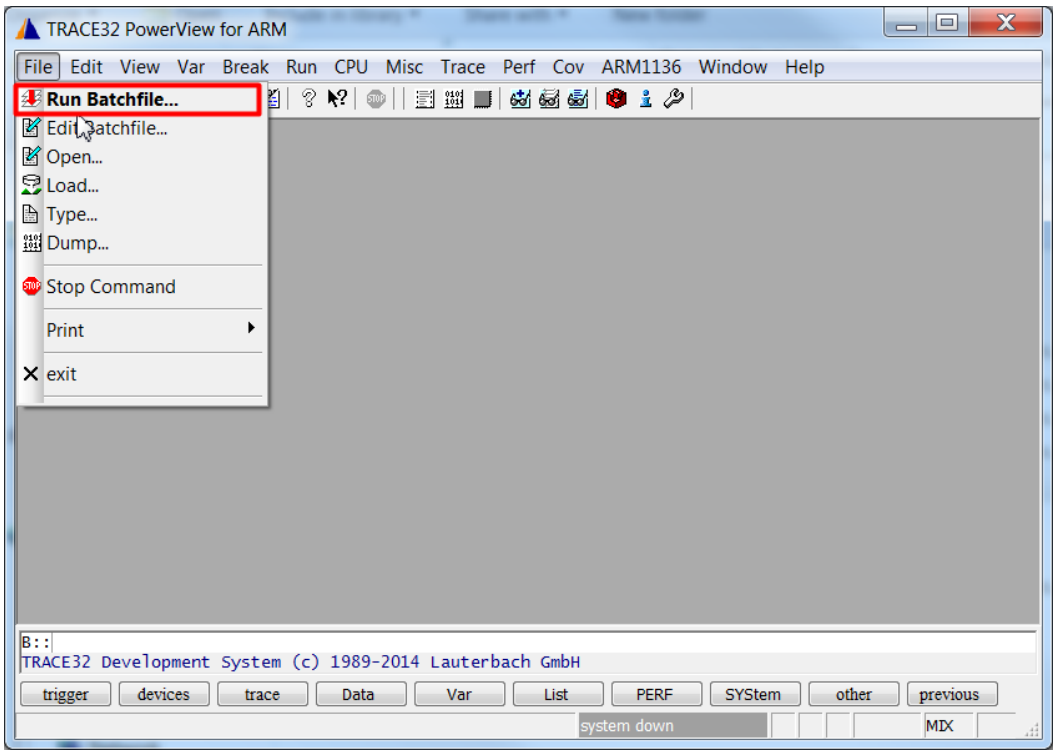

Figure 20. How to load a attach script

And then open the *attach_onecore_eboot.cmm* in the window as Figure 21. The file is mentioned in U-boot Directory Setup chapter.

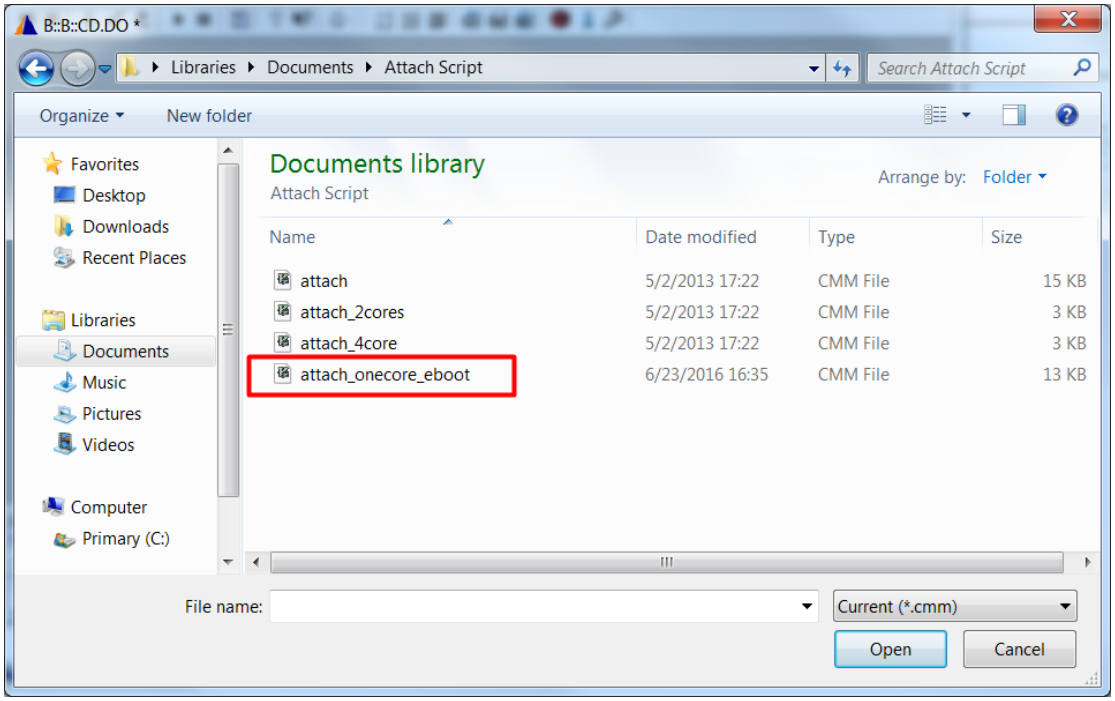

Figure 21. Open attach script

Then the debug process is started as shown in Figure 22.

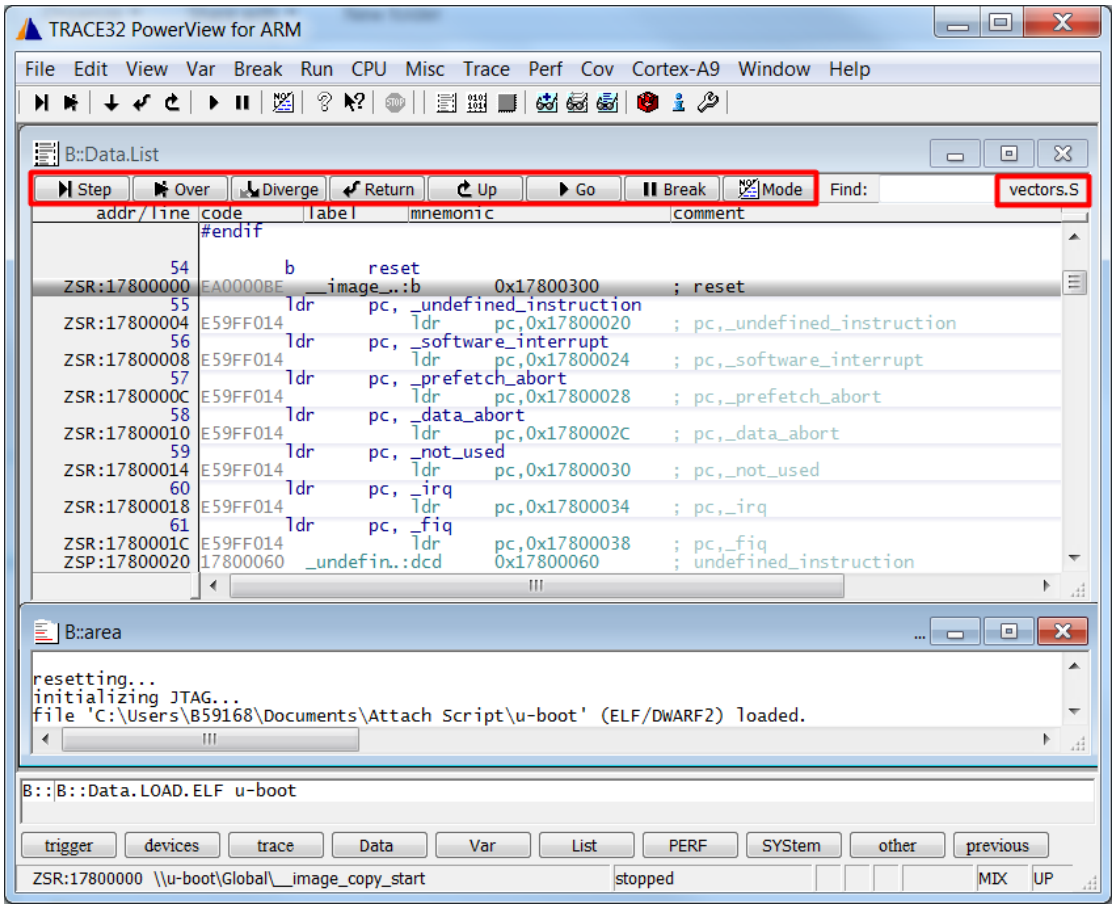

Figure 22. Debug process started

The left red rectangle box shows operation of debugging. *Step* button is to debug step by step and *Over* button can be used to skip function. The right red rectangle box points out the file that code executing at present. And if you click on it, you can locate the file in the directory as shown in Figure 23.

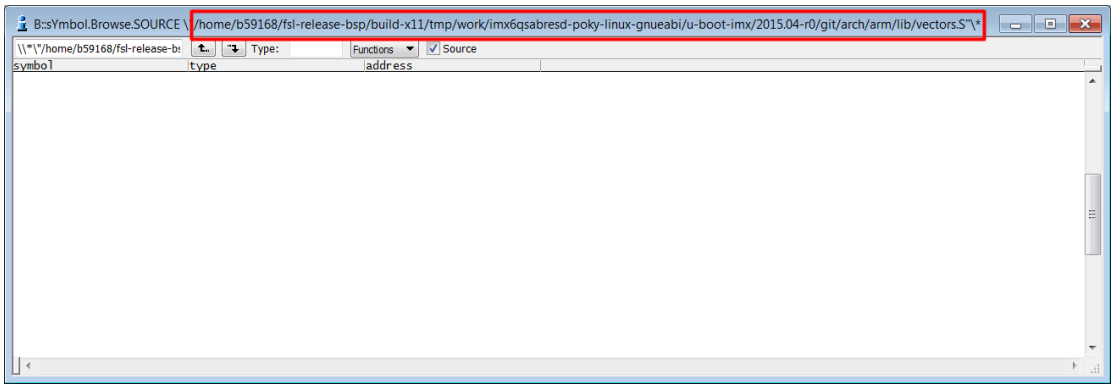

Figure 23. How to locate the debugging file

Tips: If you may meet a loop, e.g. *clr_gd as* shown in Figure 24, you can double click in the left side of code to create breakpoints. And then you can click *Go* in the debug bar to skip the loop.

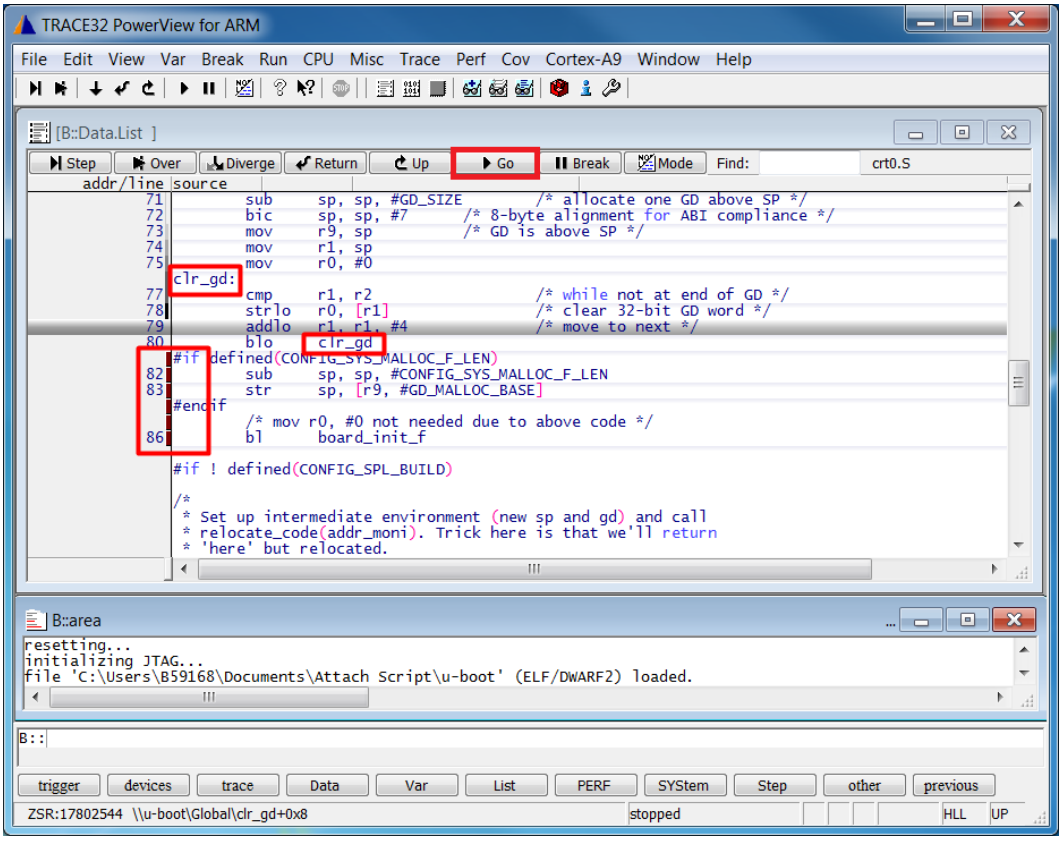

Figure 24. How to skip a loop

During the debugging process, you may see the feedback information printed in the console through the serial port. So,you can learn the system boot process better.

This document describes the basic step about debugging process of Trace32. If you need more help, you can click the *Help* in the menu bar and select *Contents* as followed in Figure 25.

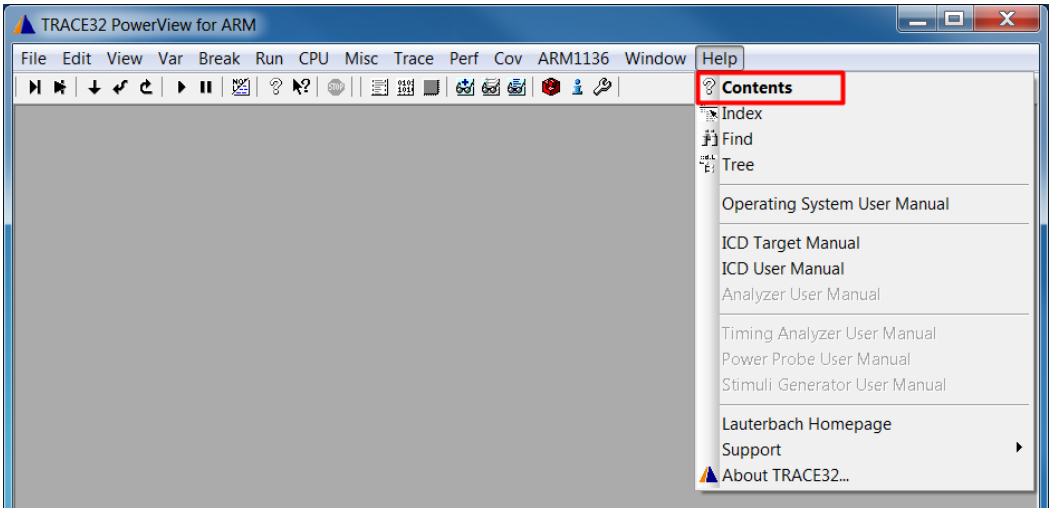

Figure 25. More contents about Trace32 help

Then a hierarchical directory of training documents appears and you can choose specific aspect you need.

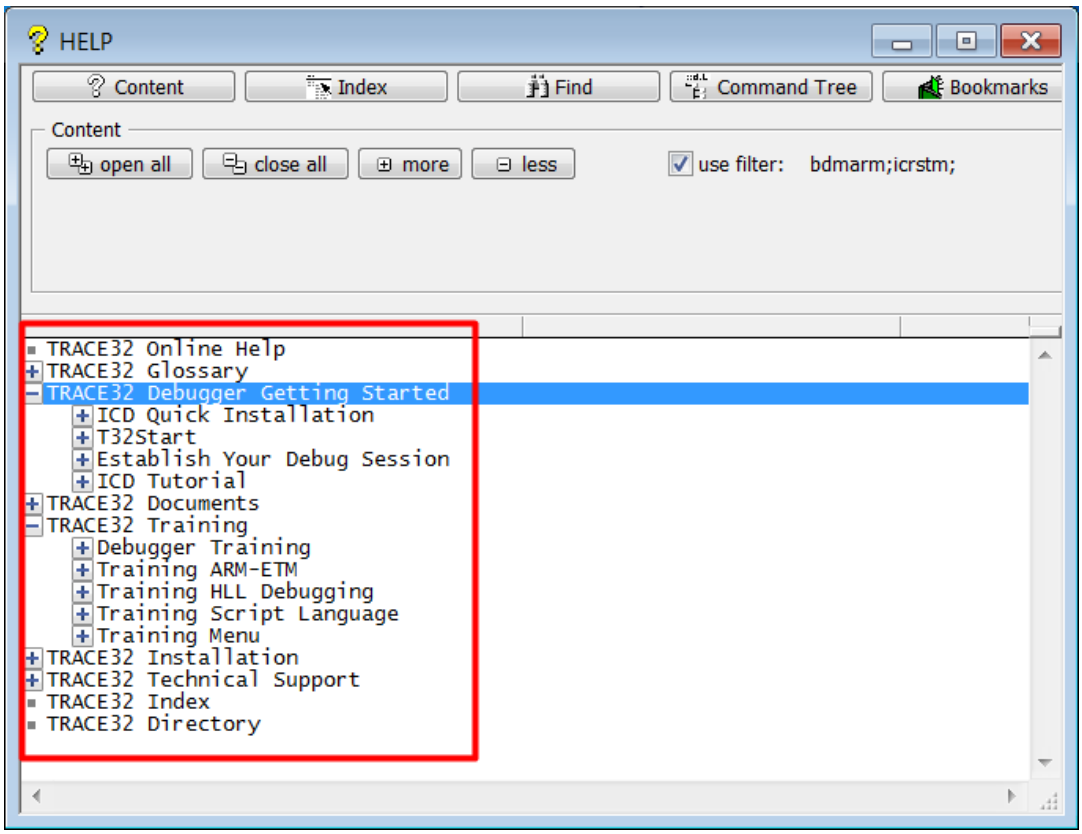

Figure 26. Hierarchical directory of help files