Application Note:

lenzeTdebug User Guide

AN-Tdebug-UGE1

Ver 1.0

2013/09/22

Keyword:

Variable; Function; Breakpoint; Register; Flash;

Tcdb command

Brief:

This application note is the user guide for lenze Tdebug tool.

lenze technology co., LTD

Published by
lenze technology co., LTD

Room C,26th FIoor,West Speclal Zone PressBuilding,NO,6008Shennan Road,Futian District Shenzhen

Legal Disclaimer

lenze technology co., LTD reserves the right to make changes without further notice to any products herein to improve reliability, function or design. lenze technology co., LTD disclaims any and all liability for any errors, inaccuracies or incompleteness contained herein or in any other disclosure relating to any product.

lenze technology co., LTD does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others

The products shown herein are not designed for use in medical, life-saving, or life-sustaining applications. Customers using or selling lenze technology co., LTD products not expressly indicated for use in such applications do so entirely at their own risk and agree to fully indemnify lenze technology co., LTD for any damages arising or resulting from such use or sale.

Information:

For further information on the technology, product and business term, please contact lenze technology co., LTD Company (www.lenzetech.com).

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

Revision History

Version	Major Changes	Date	Author
1.0	Initial version	2013/9	Z.J.Q. ; Z.X.D; Cynthia

1 Table of contents

2 Table of Figures

1 Brief introduction

This document mainly presents the instruction of lenzeTdebug tool.

As a tool of program monitoring and debugging, Tdebug can be used to read/write register and FLASH, track and view program pointer, set breakpoint, as well as call simple tcdb commands.

2 Instructions for Use

2.1 Enter Tdebug interface

Before entering Tdebug interface, correct hardware parts connection should be ensured first. There are two methods available to connect daughter board (i.e. the board to be debugged and monitored, such as mouse board) with PC, corresponding to different modes to enter Tdebug interface.

(1) Connect the daughter board to PC via EVK board.

After connection is ready, turn on “Wtcdb” tool to enter its interface, and click the “BIN” button in the upper right corner of Wtcdb interface, as indicated in marker 1 of Figure 1. A dialog window will pop up, and then select the needed bin file in the corresponding directory (as shown in marker 2 of Figure 1). Click the “open” button in the dialog window.

Click the “SWB Speed” button (as shown in marker 1 of Figure 2), and check if connection is successful according to printed information on the right side of the Wtcdb interface. Marker 2 of Figure 2 indicates successful connection.

When connection is confirmed successful, click the “Tdebug” button (as shown in marker 3 of Figure 2) to enter Tdebug interface.

After entering Tdebug interface, connection can be reestablished via clicking the “UsbReconnect” button (as shown in marker 15 of Figure 7).

(2) Another connection method, which applies to a daughter board with USB interface, is to directly connect the daughter board with PC via USB cable.

After connection is ready, run the “Tdebug” tool and click the “UsbReconnect”

button to establish connection. Information of “usb device ok” indicates successful connection, while information of “no usb device” indicates unsuccessful.

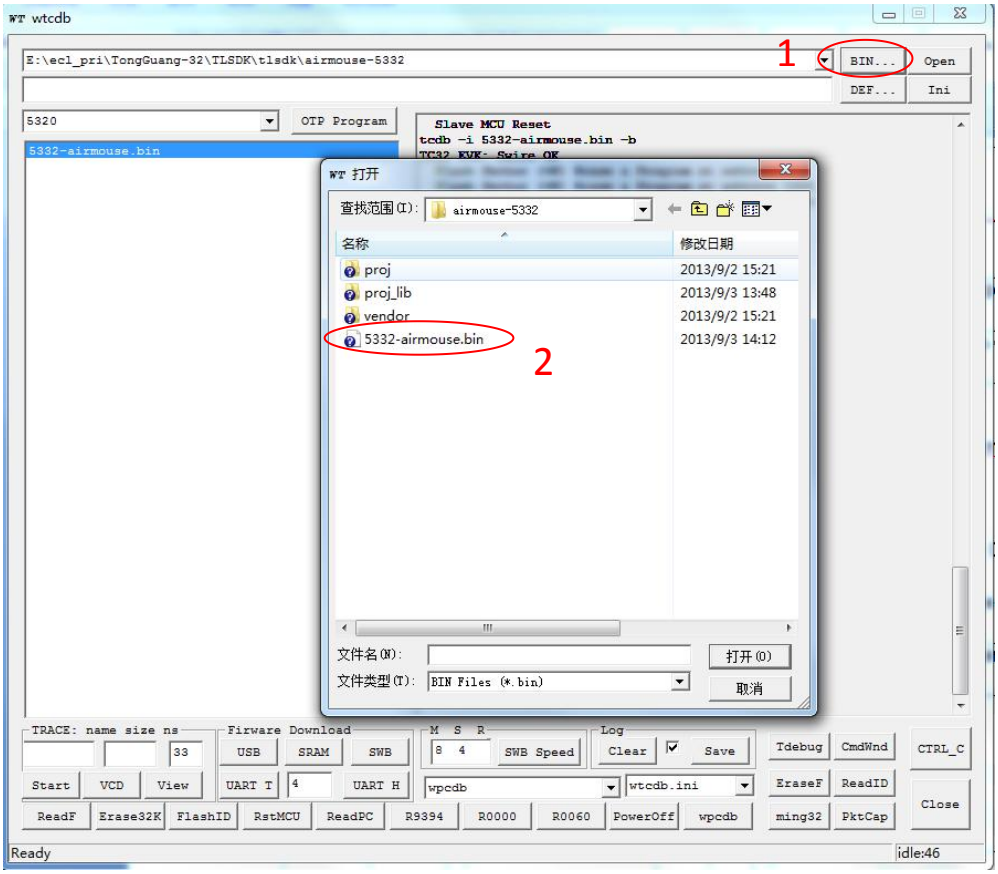


Figure 1 Wtcdb tool interface 1

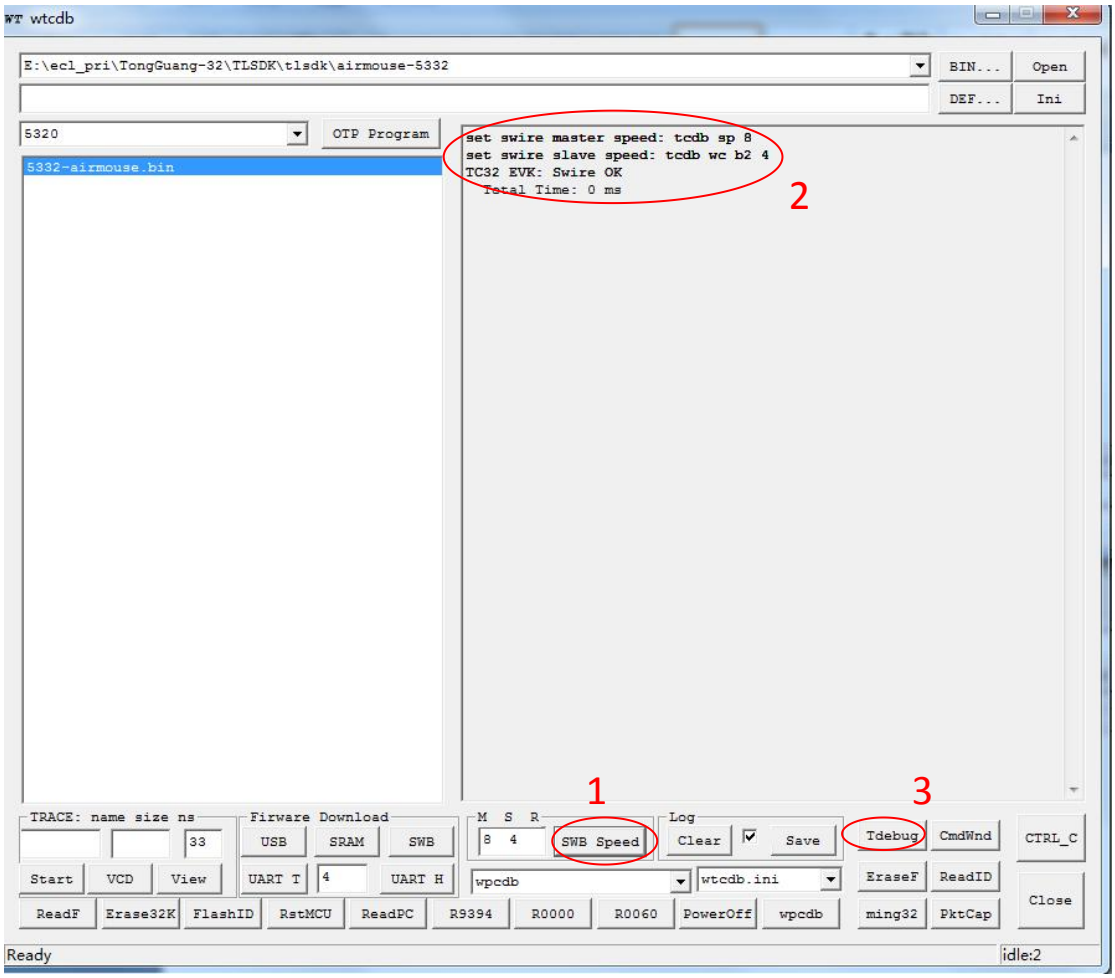
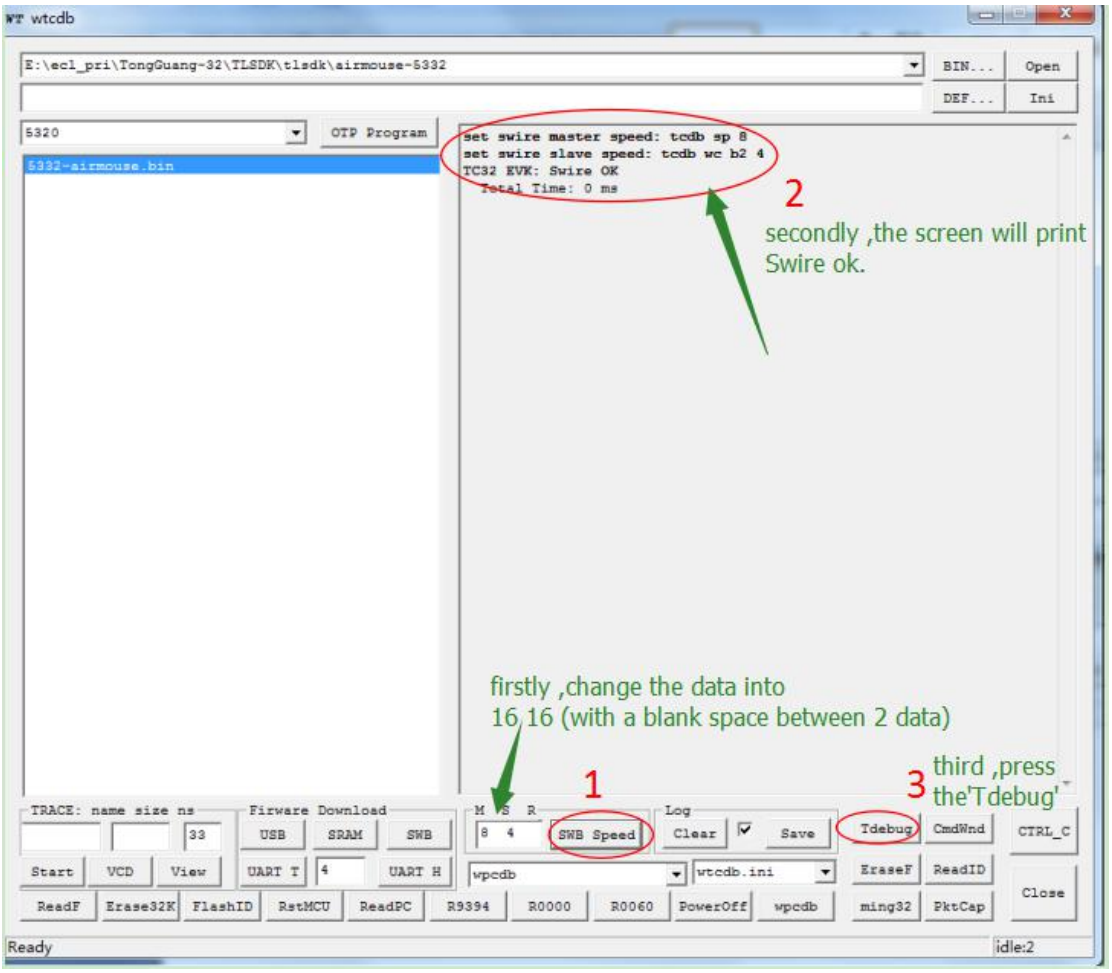


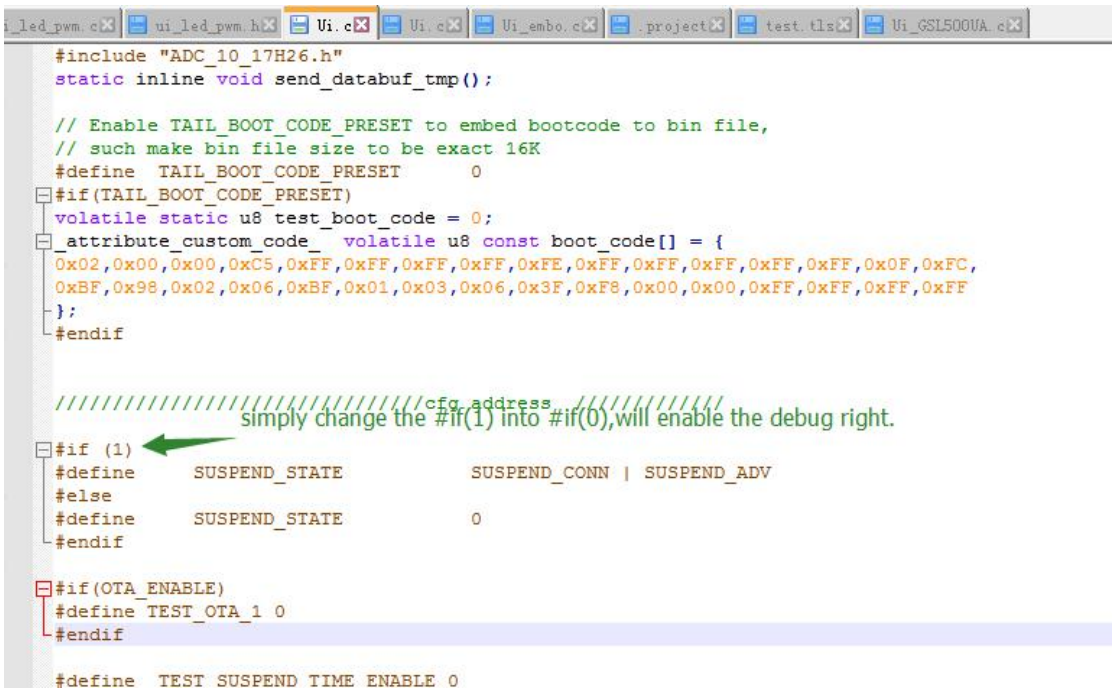
Figure 2 Wtcdb tool interface 2



If something wrong with the second step ,then ,go back to your program ,and make sure you have changed the Ui.c program into this ,which will enable the debug authority. And to make it clear ,this will also enable the full-speed of the 17h2x chip which will cost a lot of power consumption,but helpful when you want to watch the debug procedure.

```
20
21
22    ////////////////////////////////cfg address  /////////////
23
24  #if (0)
25    #define     SUSPEND_STATE           SUSPEND_CONN | SUSPEND_ADV
26    #else
27    #define     SUSPEND_STATE           0
28    #endif
29
```

And if you want to reduce the power and go into the low energy mode ,you have to change this back like below!



2.2 Monitoring of Tdebug interface

On the left side of Tdebug tool interface there is a switchable variable/function viewing region. Variables/Functions in this region are mapped to a boot-.lst file which is generated after program compiling. To ensure right display of these variables/functions, it's necessary to put the boot-.lst file in the same directory as the bin file to be burned. A control button region (as shown in Figure 7) and an information print region (below the control button region) reside on the right side of Tdebug interface.

2.2.1 Variable/Function viewing region switch

As indicated in marker 2 of Figure 3, the left side of Tdebug interface will switch to function viewing region (as shown in marker 1 of Figure 4) after clicking the "SeeFunctions" button. Then click the "SeeVariables" button, it will switch back to variable viewing region.

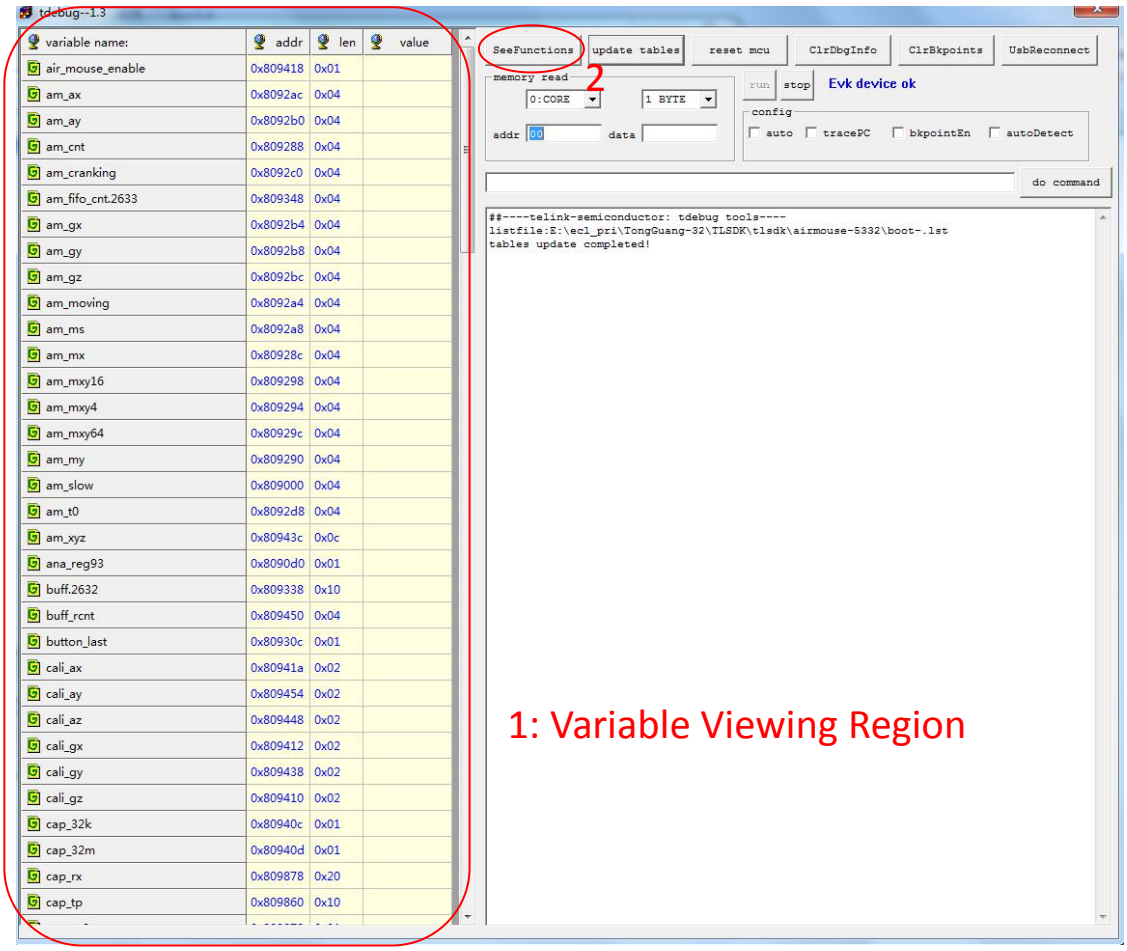


Figure 3 Tdebug tool interface 1

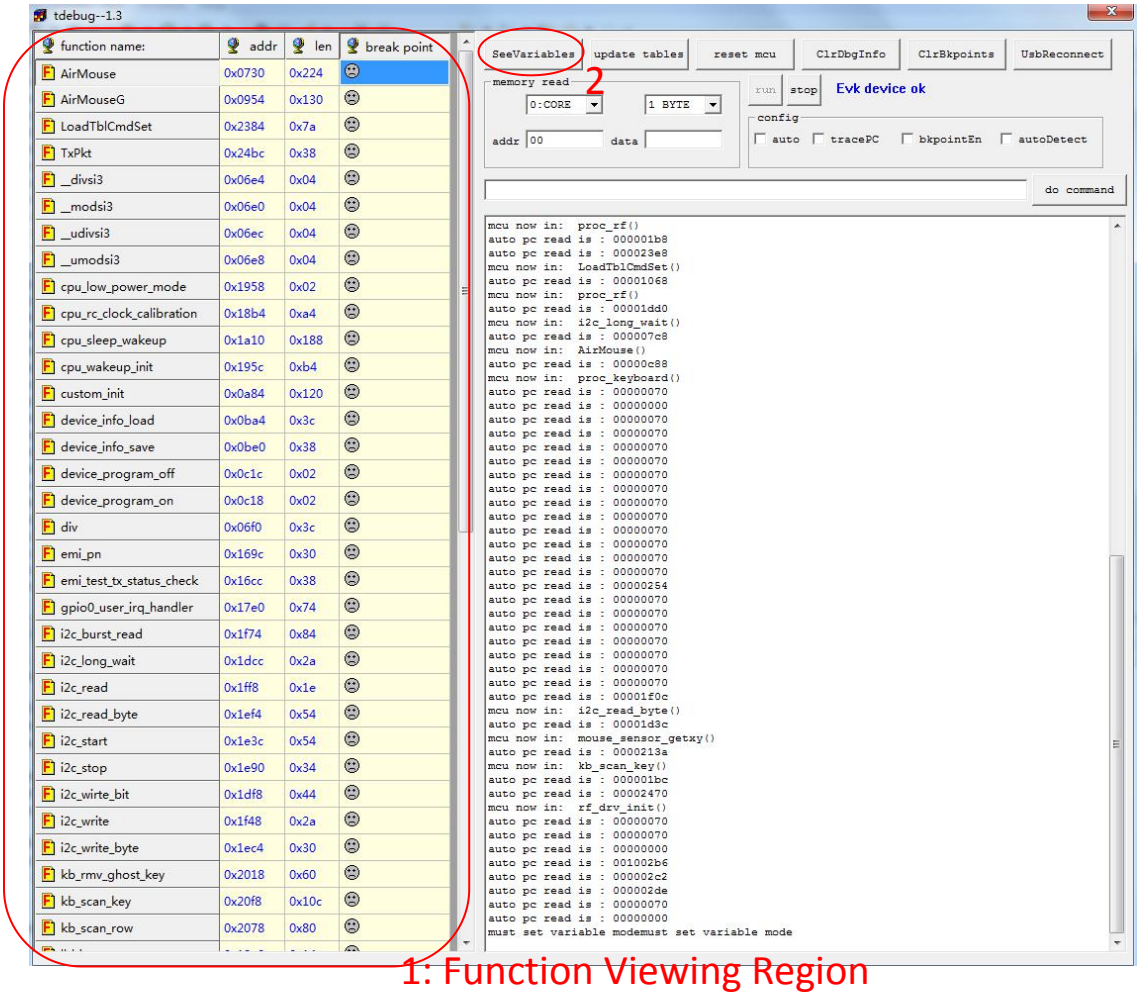


Figure 4 Tdebug tool interface 2

As shown in Figure 5, the variable viewing region contains four options: variable name, variable address in memory (i.e. “addr”), variable length (i.e. “len”), and variable value.

Figure 5 Options of variable viewing region

As shown in Figure 6, the function viewing region also contains four options: function name, function initial address in flash (i.e. “addr”), function length (i.e. “len”), and break point.

Figure 6 Options of function viewing region

2.2.2 Variable update

Variables' value may change with program running. To update variables in the variable viewing region, either click the "update tables" button (as shown in marker 1 of Figure 7) to implement manual update, or tick the "auto" option in config bar (as shown in marker 2 of Figure 7) to enable automatic update.

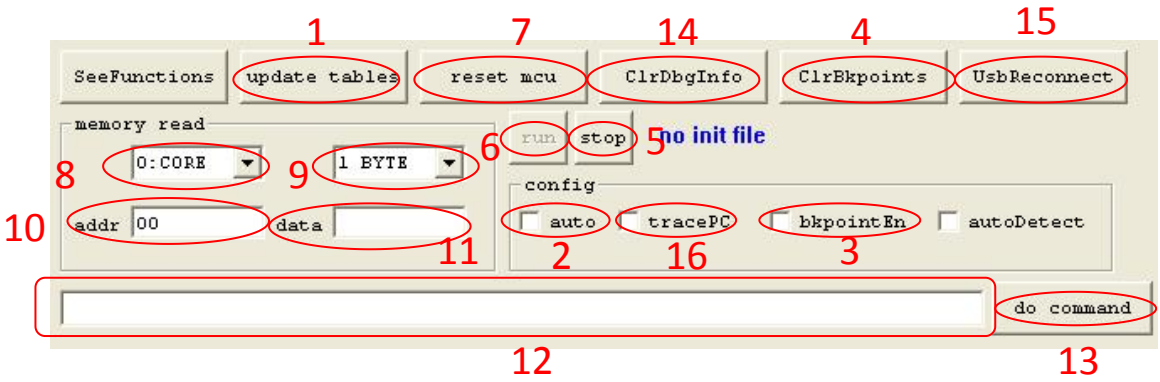


Figure 7 Control button region

2.2.3 Set/Clear function breakpoint

To set a breakpoint for some function in the function viewing region, first tick the "bkpointEn" button in the control button region (as shown in marker 3 of Figure 7) to enable breakpoint function. Then click the symbol in the "break point" column corresponding to the function line: click the symbol, and the symbol will turn red (as shown in Figure 8), which indicates setting a breakpoint at the top of the function; click the symbol again, and the symbol will turn green (as shown in Figure 9), which indicates setting a breakpoint at the end of the function; click the symbol once again, and the symbol will turn gray, which indicates cancelling the breakpoint setting.

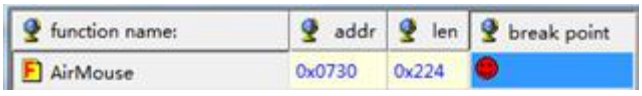


Figure 8 Chart of setting a breakpoint at the top of a function

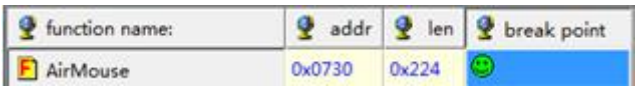


Figure 9 Chart of setting a breakpoint at the end of a function

As shown in marker 4 of Figure 7, all breakpoints can be cleared via clicking the "ClrBkpoints" button in the control button region.

2.2.4 Program run/stop

When program is running, click the "stop" button in the control button region (as shown in marker 5 of Figure 7), thus the program can stop running immediately.

Then click the "run" button (as shown in marker 6 of Figure 7), so that the program will continue running from the breakpoint where it stopped.

Clicking the "reset mcu" button (as shown in marker 7 of Figure 7) indicates that the program will restart running from the top.

2.2.5 Read/Write register or FLASH

The "memory read" bar in the control button region can be used to read/write registers or Flash.

Either register (0: Core) or Flash (2: FLASH) can be selected via the drop-down box as shown in marker 8 of Figure 7. The drop-down box as shown in marker 9 of Figure 7 can select following number of bytes: 1 byte, 2 bytes, or 4 bytes.

Enter the address corresponding to register/Flash in the box behind the "addr" option (as shown in marker 10 of Figure 7), and press the "Tab" key to make a confirmation; thus the printing window will print out the content in designated address of register/Flash.

As shown in marker 11 of Figure 7, enter some data in the box behind the "data" option, and press the "Enter" key to make a confirmation; thus the data can be written to the designated address of register/Flash.

2.2.6 Call simple tcdb commands

Enter some simple tcdb command in the box as shown in marker 12 of Figure 7, and press the “Enter” key or click the “do command” button (as shown in marker 13 of Figure 7); thus the printing window will print out information indicating command execution result.

Note: Please refer to the document “lenzeTcdb User Guide” for tcdb command format, and the word “tcdb” can be omitted when entering a command herein.

2.2.7 Others

“ClrDbgInfo” button (as shown in marker 14 of Figure 7) function: clear the content of the printing window.

“UsbReconnect” button (as shown in marker 15 of Figure 7) function: reestablish the connection.

“tracePC” option (as shown in marker 16 of Figure 7) function: trace Program Counter, view program pointer. Ticking the option can help decide whether the program is working or failing.