CLR-USB-CAN-I PRO

USB to CAN adapter

User manual

Revision History

Ver.	Date	Reason
V1.00	2013/6/16	Create document
V2.01	2013/12/20	Fixed working parameters
V3.01	2015/04/22	Add some parameters

Contents

1. Introduction

1.1 Functional Overview

USBCAN-I Pro adapter is a debug or analysis tool with two CAN-Bus channel. With this adapter, PC can quickly connect to CAN-Bus network through USB interface, and become a intelligent node of CAN-Bus to transmit/receive CAN-Bus data. Adapter comes with isolation, and can be used in different Windows systems. Device driver, software and programming interfaces(VC, VB, Net, Delphi, Labview, C++Builder) exist for different operating systems, so programs can easily access a connected CAN bus.

1.2 Properties at a Glance

- Adapter for USB connection (USB 1.1, compatible with USB 2.0)
- USB voltage supply
- Bit rates up to 1 Mbit/s Time stamp resolution 1μs
- Compliant with CAN specifications 2.0A (11-Bit ID) and 2.0B (29-Bit ID)
- CAN-Bus connection via OPEH4, PHOENIX
- NXP SJA1000 CAN controller
- NXP PCA82C251 CAN transceiver
- Support ECAN Tools software
- Galvanic isolation on the CAN connection up to 1500 V
- Extended operating temperature range from -40 to 85 °C
- Device driver and software support Windows 2000/2003/XP/7/8/10
- Size: (L)95mm * (W)57mm * (H)24mm

1.3 Typical application

- Test CAN-Bus network or device;
- Automotive electronics development;
- Electrical system communication test.
- industrial control network.
- Listen all CAN-Bus communication.

2. Installation

This chapter describes how to connect the USB-CAN adapter to the computer and the precautions when connecting the USB-CAN adapter to the computer for the first time.

2.1 Driver and software installation

Note: Before install the driver or software, please ensure that the user login windows account is administrator, or the user account has to install the driver and software related permissions, otherwise it may lead to the installation failed.

2.1.1 Install driver and software

ECAN Tools has been integrated hardware driver installation program, users can directly install ECAN Tools. If you only need to install the driver, please enter the “driver” folder, select the installation file that corresponds to the system type. (“DriverSetup.exe” for 32-bit. “DriverSetup64.exe” for 64-bit)

2.1.2 Uninstall driver and software

Users can run the DriverSetup.exe/ DriverSetup64.exe and click "Uninstall" button to uninstall the installed device driver.

2.2 Connect to PC

The adapter can be connected directly to a PC using a USB cable, if the USB power supply is insufficient, you need to use external power supply.

2.2.1 USB power supply mode

USB power supply mode is suitable for the most applications, such as: when USBCAN-I Pro is the only device in USB port.

2.2.2 External power supply mode (only the USBCAN - II Pro support)

External power supply mode is suitable for the USB port using an USB HUB and have already connect multiple USB device, this will lead to the adapter lack of electricity supply.

2.3 Connect to CAN-Bus

USBCAN-I Pro has one CAN-Bus by a 4 Pin plug type terminal leads, this CAN-Bus channels can connect to CAN-Bus network or devices. Terminal pin definition as Table 2.1 below.

Pin	Prot	Name	Function
1	CAN	PE	Shield earthing
2		L	CAN_L signal line
3		G	CAN_GND
4		H	CAN_H signal line

Table 2.1 USBCAN-I Pro adapter pin definition

Note: In practical use, most of the time just connected the CAN_H to CAN_H and CAN_L connected to CAN_L then communication can be realized

3. Adapter in use

3.1 Connect to USB

USBCAN-I Pro adapter can support to the USB2.0 full speed protocol specification, compatible USB1.1. When driver and software have been installed, connect the adapter to the USB interface, a new USBCAN device named "GC - Tech USBCAN Device" can be found in the PC Device manager. If there is no !or ?mark that the device run fine.

3.2 Connect to CAN

USBCAN-I Pro adapter connect to CAN-Bus as chapter 2.3, CAN_H to CAN_H,CAN_L to CAN_L. The CAN bus network adopts topological structure, only the two furthest terminal need to connect 120Ω terminal resistance between CAN_H and CAN_L. For branch connection, its length should not be more than 3m. CAN-bus nodes connection as shown in figure 3.1

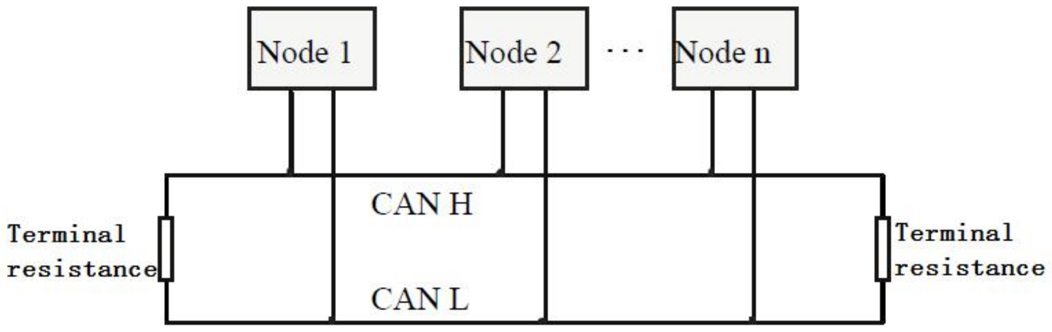


Figure 3.1 CAN-bus network

Note: the CAN-bus cable can use ordinary twisted-pair cable, shielded twisted-pair cable. Theory of the maximum communication distance depends on the bus baud rate, Their relationship as shown in the Table 3.1.

Baud rate	**Distance**
1 Mbit/s	40m
500 kbit/s	110m
250 kbit/s	240m
125 kbit/s	500m
50 kbit/s	1.3km
20 kbit/s	3.3km
10 kbit/s	6.6km
5 kbit/s	13km

Table 3.1 relationship of baud rate and distance

3.3 CAN-Bus terminal resistance

In order to improving the communication reliability and eliminating CAN-bus terminal reflection, the two furthest terminal need to connect terminal resistance between CAN_H and CAN_L as shown in figure 3.2. Terminal resistance values determined by the characteristic impedance of the cables. Such as, the characteristic impedance is 120Ω.

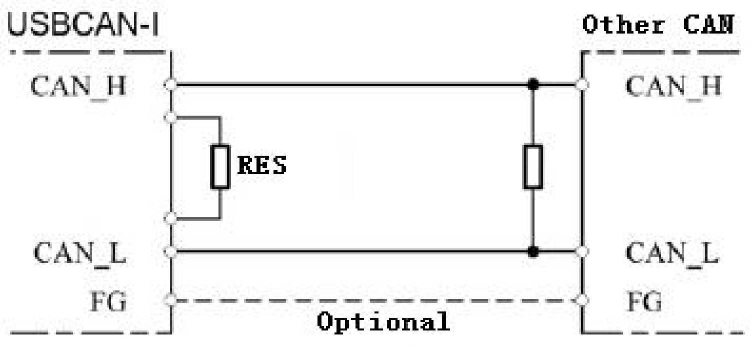


Figure 3.2

Note: USBCAN-I Pro adapter has integrated one 120Ω terminal resistance, users can choose whether enable.

3.4 System LED

USBCAN-I Pro adapter with one PWR indicator, one SYS indicator, one TX indicator one RX indicator to indicate the adapter status. More functions are shown in table 3.2 and 3.3.

Indicator	Colour	State
PWR	Green	Power indicator
SYS	Green	System indicator
TX	Green	CAN transmit indicator
RX	Green	CAN receive indicator

Table 3.2 USBCAN-I Pro adapter indicator LED

When USBCAN-I Pro adapter power on, PWR and SYS light, indicates the adapter has power supply, the system is initialized; Otherwise, a system power failure or system errors has exist. When USB-Bus data transfer, SYS indicator will blinking. When CAN-Bus data transceiver, the corresponding TX or RX will blinking.

Indicator	State	Meaning
PWR	ON	Power supply normal
	OFF	Power supply error
SYS	ON	Standby mode
	OFF	Initialization error
	Blinking	USB data transmission
TX / RX	OFF	CAN-Bus no data
	Blinking	CAN-Bus data transmission

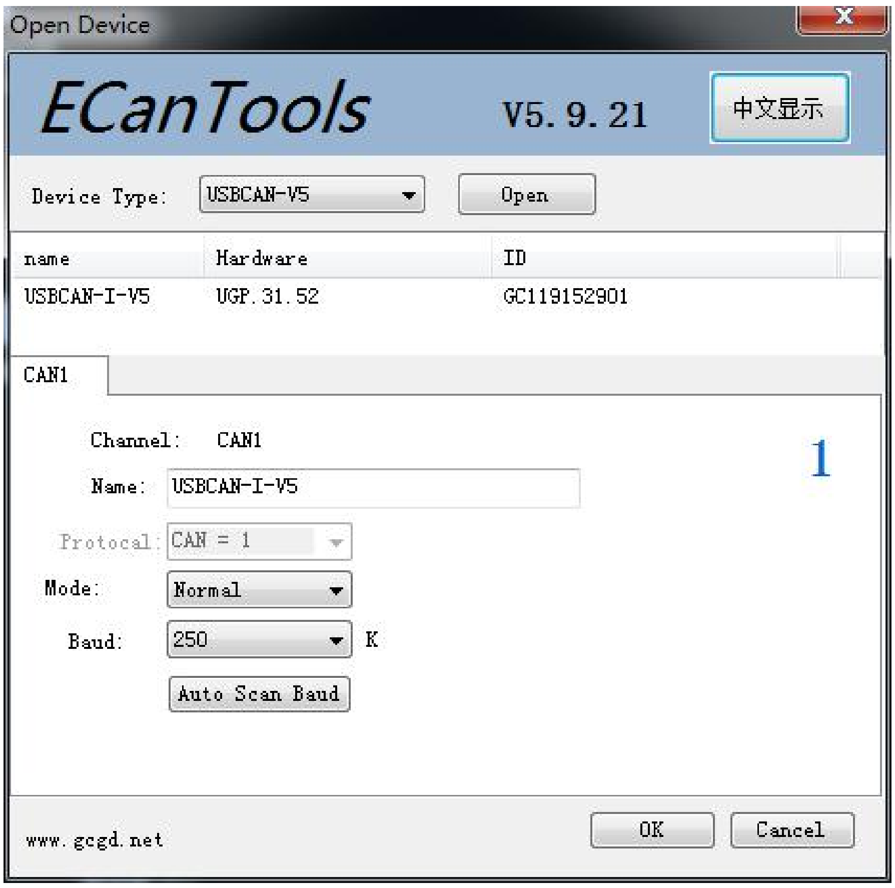
Table 3.3 USBCAN-I Pro adapter LED state

4. ECAN Tools introduction

Users can use ECAN Tools software to receive and transmit CAN data. Flexible use of functions can help to more with less.

4.1 Start

1. If ECAN Tools has been installed, users can directly run it on the desktop.



Device Type: USBCAN-V5

name	Hardware	ID
USBCAN-I-V5	UGP. 31. 52	GC119152901

CAN1

Channel: CAN1

Name: USBCAN-I-V5

Protocal: CAN = 1

Mode: Normal

Baud: 250 K

Figure 4.1

2. Choose the device type and click "Open", one adapter will shown in the below.

3. Choose work mode. Software provides three kinds of mode: normal, listen only, test self.

Normal: use this mode to transmit or receive data.

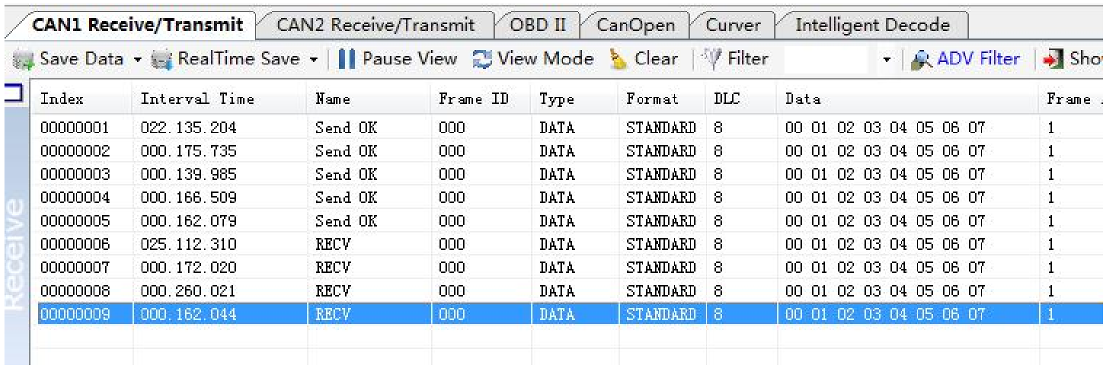
Listen Only: use this mode to receive data only, and don’t send response or clock.

Test Self: use this mode to test if the adapter is working well.

4. Choose baud rate according to the CAN-bus, don't match will lead to communication failed. If you don't know the baud rate, you can use "Auto Scan Baud Rate" to adapt.

4.2 Transmit/Receive data

Transmitting and receiving is the basic function of ECAN Tools, in this interface, users can directly see the received CAN data, and sent the data to CAN-bus.

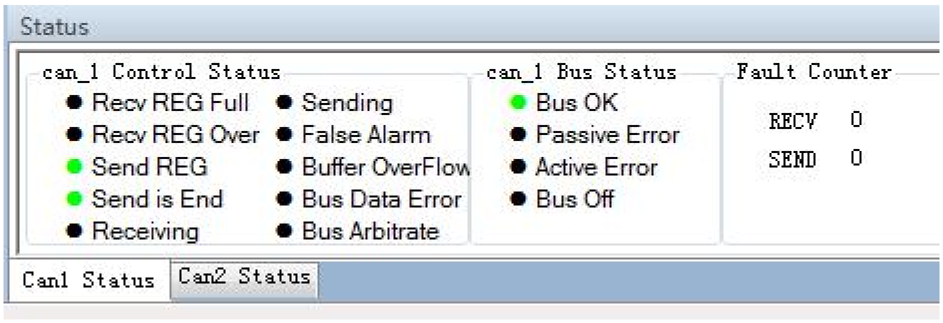


Index	Interval Time	Name	Frame ID	Type	Format	DLC	Data	Frame
00000001	022.135.204	Send OK	000	DATA	STANDARD	8	00 01 02 03 04 05 06 07	1
00000002	000.175.735	Send OK	000	DATA	STANDARD	8	00 01 02 03 04 05 06 07	1
00000003	000.139.985	Send OK	000	DATA	STANDARD	8	00 01 02 03 04 05 06 07	1
00000004	000.166.509	Send OK	000	DATA	STANDARD	8	00 01 02 03 04 05 06 07	1
00000005	000.162.079	Send OK	000	DATA	STANDARD	8	00 01 02 03 04 05 06 07	1
00000006	025.112.310	RECV	000	DATA	STANDARD	8	00 01 02 03 04 05 06 07	1
00000007	000.172.020	RECV	000	DATA	STANDARD	8	00 01 02 03 04 05 06 07	1
00000008	000.260.021	RECV	000	DATA	STANDARD	8	00 01 02 03 04 05 06 07	1
00000009	000.162.044	RECV	000	DATA	STANDARD	8	00 01 02 03 04 05 06 07	1

Figure 4.2

4.3 CAN-Bus diagnosis function

CAN-Bus diagnosis function can detect the bus error frames and bus arbitration lost.



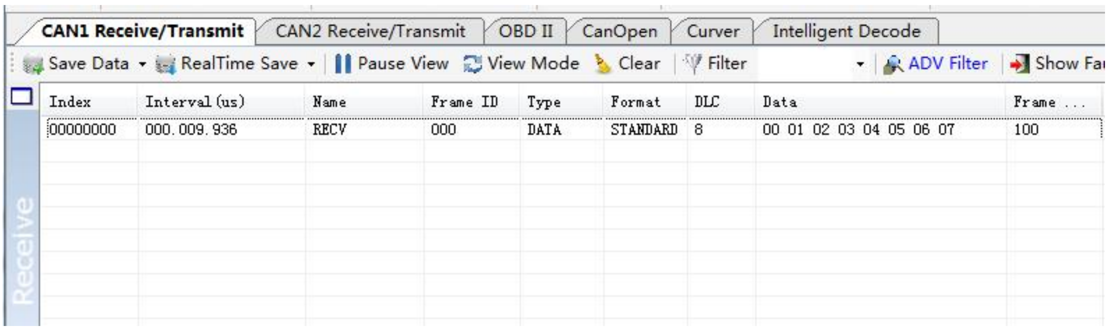
RECV 0
SEND 0

Figure 4.3

CAN bus status display: indicate the CAN bus status include: bus normal, passive error, active error, bus hung. **The CAN controller FIFO overflow:** message within a certain period of time is too dense, lead to data loss. **The CAN controller error alarm:** when many of errors on the bus, error counter exceeds the alarm threshold, and display the error count. **The CAN controller negative error:** when many of send or receive errors, lead to the CAN controller into the negative state, and display the error count. **CAN bus controller error:** when nodes send or receive errors, error counter value will be accumulate, and can catch the wrong information, such as ACK, CRC error and so on.

4.4 Statistical mode

When receiving data, software can classify these data in ID, data, name, format or type and counting the number of each data. This function is suitable for large data systems, engineers can easily observe and analyze other data after same data is combined.



Index	Interval (us)	Name	Frame ID	Type	Format	DLC	Data	Frame ...
00000000	000.009.936	RECV	000	DATA	STANDARD	8	00 01 02 03 04 05 06 07	100

Figure 4.4

4.5 Other functions

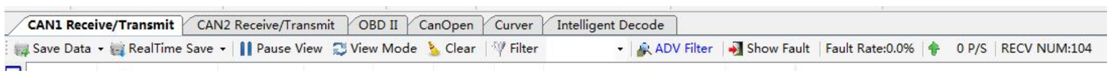


Figure 4.5

Save data: save the receiving list, save format: txt, can, csv and binary. Display mode: scroll mode and list mode, list mode can classified data together according to the rules. Filter settings: users can set multi-stage filtering by editing the filter ID. Data mask: masked ID is not displayed. Error frames: error frames on the bus can be displayed / hidden.

Note: If you want to know more about the software specific function and usage, please see the ECAN Tools software instructions document.

5. Linux system instructions

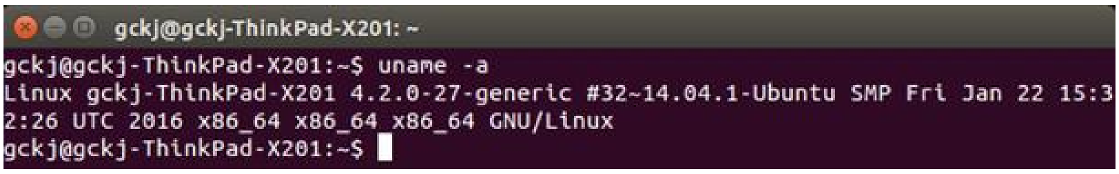
Vhandy Technology's USBCAN analyzer series products support various versions of Linux operating system. Our company will provide customers with 32/64-bit Linux system drivers and secondary development related documents. Users can develop and use in Linux systems by themselves.

The general method of using our company's equipment in Linux system is as follows: ① Obtain system administrator authority; ② Copy the necessary files to the system GCC compilation directory; ③ Switch the directory to the USBCAN driver folder to compile; ④ Run the test program. The specific operation method is as follows:

1. Check the linux version number and confirm the system type (32/64 bit).

Input: uname -a

(It can be seen from the result that our Linux system type is 64-bit)



```
gckj@gckj-ThinkPad-X201: ~
gckj@gckj-ThinkPad-X201:~$ uname -a
Linux gckj-ThinkPad-X201 4.2.0-27-generic #32~14.04.1-Ubuntu SMP Fri Jan 22 15:3
2:26 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux
gckj@gckj-ThinkPad-X201:~$
```

Figure 5.1

After confirming the Linux system type, copy the corresponding USBCAN driver file to the system.

(In this example, we put the driver file on the system desktop)

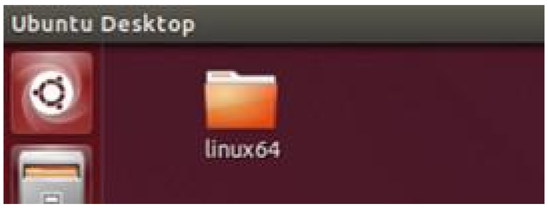
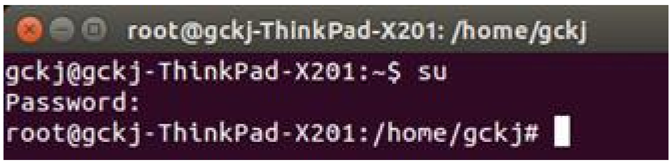


Figure 5.2

3. Obtain administrator rights to facilitate subsequent installation of drivers and other operations. Enter: su

(After entering the su command, the administrator password is required. Enter the correct password to obtain administrator privileges)



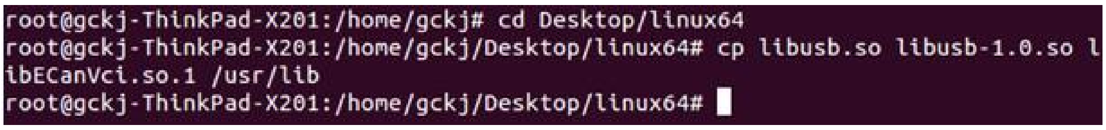
```
root@gckj-ThinkPad-X201: /home/gckj
gckj@gckj-ThinkPad-X201:~$ su
Password:
root@gckj-ThinkPad-X201:/home/gckj#
```

Figure 5.3

4. Enter the USBCAN driver folder, copy libusb.so, libusb-1.0.so, libECanVci.so.1 to the gcc compilation library directory. (The default path is /usr/lib)

Enter: cp libusb.so libusb-1.0.so libECanVci.so.1 /usr/lib

(The default path is /usr/lib)



```
root@gckj-ThinkPad-X201:/home/gckj# cd Desktop/linux64
root@gckj-ThinkPad-X201:/home/gckj/Desktop/linux64# cp libusb.so libusb-1.0.so l
ibECanVci.so.1 /usr/lib
root@gckj-ThinkPad-X201:/home/gckj/Desktop/linux64# 
```

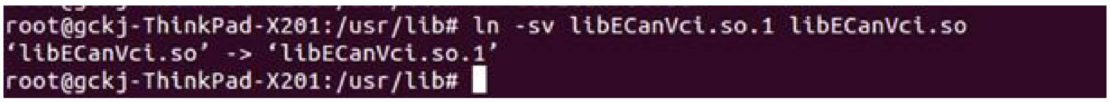
Figure 5.4

5. Enter the gcc compilation library folder and link libECanVci.so.1 and libECanVci.so together.

Enter: ln -sv libECanVci.so.1 libECanVci.so

6.Enter the USBCAN driver folder again and compile.

Enter: make

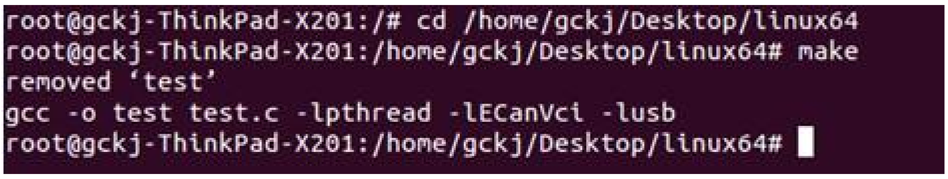


```
root@gckj-ThinkPad-X201:/usr/lib# ln -sv libECanVci.so.1 libECanVci.so
'libECanVci.so' -> 'libECanVci.so.1'
root@gckj-ThinkPad-X201:/usr/lib# 
```

Figure 5.5

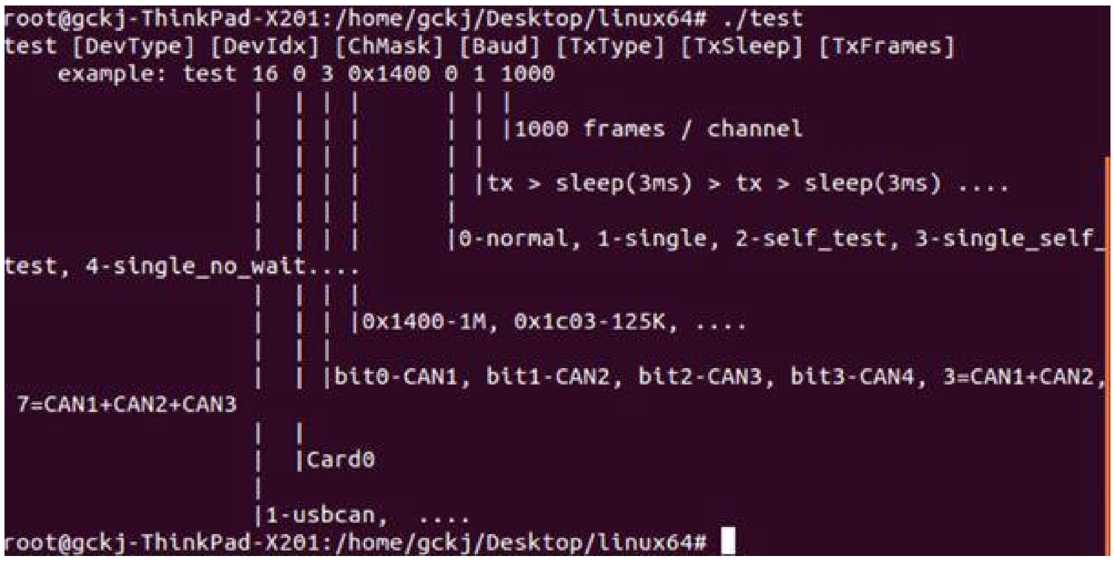
7.Run the test program to test the USBCAN transceiver.

Enter: ./test



```
root@gckj-ThinkPad-X201:/# cd /home/gckj/Desktop/linux64
root@gckj-ThinkPad-X201:/home/gckj/Desktop/linux64# make
removed 'test'
gcc -o test test.c -lpthread -lECanVci -lusb
root@gckj-ThinkPad-X201:/home/gckj/Desktop/linux64# 
```

Figure 5.6



```
root@gckj-ThinkPad-X201:/home/gckj/Desktop/linux64# ./test
test [DevType] [DevIdx] [ChMask] [Baud] [TxType] [TxSleep] [TxFrames]
    example: test 16 0 3 0x1400 0 1 1000
                  |  |  | |      | | |
                  |  |  | |      | | |1000 frames / channel
                  |  |  | |      | |
                  |  |  | |      | |tx > sleep(3ms) > tx > sleep(3ms) ....
                  |  |  | |      |
                  |  |  | |      |0-normal, 1-single, 2-self_test, 3-single_self_
test, 4-single_no_wait....
                  |  |  | |
                  |  |  | |0x1400-1M, 0x1c03-125K, ....
                  |  |  |
                  |  |  |bit0-CAN1, bit1-CAN2, bit2-CAN3, bit3-CAN4, 3=CAN1+CAN2,
 7=CAN1+CAN2+CAN3
                  |  |
                  |  |Card0
                  |
                  |1-usbcan,  ....
root@gckj-ThinkPad-X201:/home/gckj/Desktop/linux64# 
```

Figure 5.7

After entering ./test, system prompts and examples will appear, among them:

The first digit (16): device type, single-channel device input 3, dual-channel input 4; second digit (0): device index number, 0 when only one USBCAN is connected;

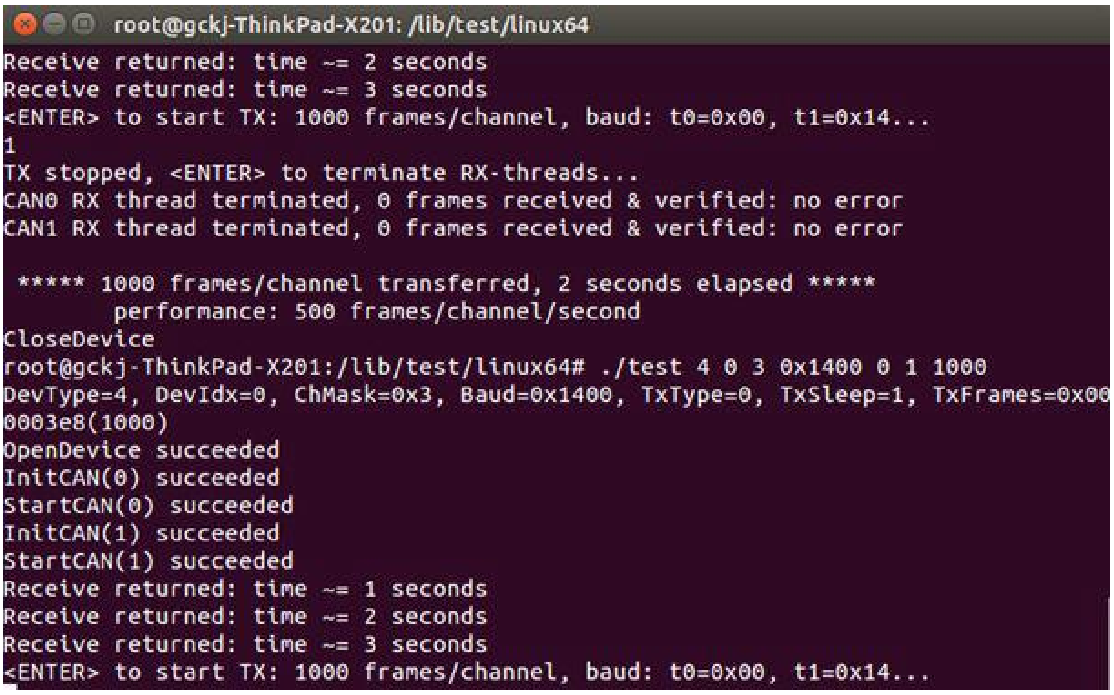
The third digit (3): turn on the number of CAN channels, turn on CAN1 input 1, turn on CAN2 input 2 and turn on CAN1 and CAN2 input 3 at the same time;

The fourth digit (0x1400): Set the CAN bus baud rate, 0x1400 means the baud rate is 1000K, for other baud rate values, please refer to "EcanVCI Dynamic Library User Manual";

Fifth bit (0): working mode, 0 is normal mode, for other working modes, please refer to "EcanVCI Dynamic Library User Manual";

The sixth digit (1): the sending time interval, in ms; the seventh digit (1000): the number of sending.

8. After running the test program, you can use other USBCAN devices to receive the random data sent by him. This routine only completes the process of sending the random number play, and there is no subsequent receiving function part, which needs to be written by the user. (Refer to the dynamic library manual)



```
root@gckj-ThinkPad-X201: /lib/test/linux64
Receive returned: time ~= 2 seconds
Receive returned: time ~= 3 seconds
<ENTER> to start TX: 1000 frames/channel, baud: t0=0x00, t1=0x14...
1
TX stopped, <ENTER> to terminate RX-threads...
CAN0 RX thread terminated, 0 frames received & verified: no error
CAN1 RX thread terminated, 0 frames received & verified: no error

 ***** 1000 frames/channel transferred, 2 seconds elapsed *****
        performance: 500 frames/channel/second
CloseDevice
root@gckj-ThinkPad-X201:/lib/test/linux64# ./test 4 0 3 0x1400 0 1 1000
DevType=4, DevIdx=0, ChMask=0x3, Baud=0x1400, TxType=0, TxSleep=1, TxFrames=0x00
0003e8(1000)
OpenDevice succeeded
InitCAN(0) succeeded
StartCAN(0) succeeded
InitCAN(1) succeeded
StartCAN(1) succeeded
Receive returned: time ~= 1 seconds
Receive returned: time ~= 2 seconds
Receive returned: time ~= 3 seconds
<ENTER> to start TX: 1000 frames/channel, baud: t0=0x00, t1=0x14...
```

Figure 5.8

6 Secondary development

We will provide interface, example and library for secondary development customers. dll and library named: “ECANVCI.h”, “ECANVCI.lib”, “ECANVCI.dll”. These libraries standards compliant, users can use these in VC, VB and some other programming environment, to use these libraries, please see “ECAN dynamic library manual” and Figure 6.1.

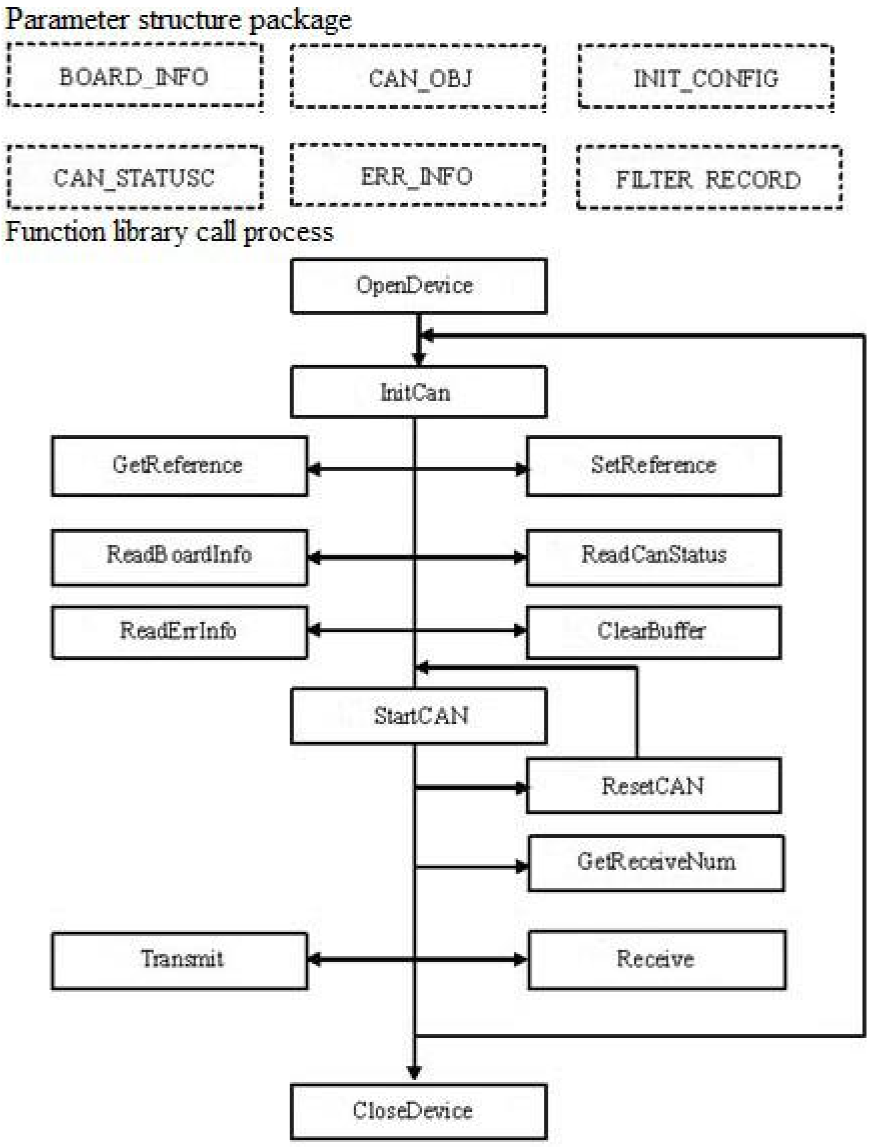


Figure 6.1 Secondary development function call process

7. Technical Specifications

Connection	
PC	USB, type A
CAN	OPEN4 PHOENIX
Interface	
USB	USB2.0 full speed, USB 1.1
CAN	ISO 11898 standard, support CAN2.0A/B
CAN baud rate	5Kbit/s~1Mbit/s
Isolation	1500V, DC-DC
CAN terminal resister	Integrated, code switch to enable
Power	
Voltage	+5V DC (USB port)
Current	200mA (Max)
Environment	
Temperature	-40℃~+85℃
Humidness	15%~90%RH, without condensation
EMC test	EN 55024:2011-09 EN 55022:2011-12
IP grade	IP 20
Basic	
Size	95mm *57mm *24mm
Weight	90g