CAN-USB-I/II Intelligent CAN-bus Interface

V6.4

Contents

Chapter 1 Product Overview

1.1 Product Overview

CANUSB-I/II interface module is an intelligent CAN-bus communication interface that compatible with USB2.0 bus and supports one/two CAN channel. Using this module will enable PC to connect to CAN-bus network via USB bus, forming the CAN-bus network control nodes for the data processing and data collection for the CAN-bus networks such as bus laboratory, industrial control, intelligent residential zone, auto electronics network, and etc.

CANUSB-I/II interface comes with an electrical isolation module, which could be used to avoid the damage caused by the ground loop and enhance the system reliability when working under a tough environment.

CANUSB-I/II interface module can use CANUSB Tester software provided by us to directly finish CAN Bus configuration, message sending, and receiving.

CANUSB-I/II interface module supports Win9X/Me、Win2000/XP 、Server 2003、 Vista operation systems, as well as Windows CE. And CANUSB-I/II provides dynamic link library (DLL) and the complete demonstration code, including VB, VB2003, VC, C++Builder, Delphi, Labview, eMbedded Visual C++ 4.0(for Wince), which make it convenient for user to develop programs.

1.2 Parameters

PC interface supports USB1.1 protocol and is USB 2.0 compliant;
Max data flow 6500 fps (extend frame);

Development kits for Win9X/Me, Win2000/XP, Server 2003, Vista, as well as Windows CE;
Examples of VB, VB2008, VC, C++Builder, Delphi, Labview, eMbedded Visual C++ are available;
Adopts electrical isolation, the isolation voltage is : 2500Vrms;
Supports CAN2.0A and CAN2.0B protocols, conforms to ISO/DIS11898 specification;
Integrates 1/2-channels CAN-bus interface, each channel can be operated independently;
Programmable CAN-bus communication Baud rates from 5Kbps to 1Mbps;
Small size, plug and play;
Directly powered by USB port, no need for external power supply;
Operating temperature: -20 to +70 ℃;
Physical size: (length) 112mm * (width) 84mm * (height) 28mm.

1.3 Typical applications

CAN-bus network diagnosis and test
Auto electronic applications
Electric power communication network
Industrial control devices
High-speed and large data communications

1.4 Ordering Information

Part Number	Operating temperature	Interface
CANUSB-I	-20℃ ~ +70℃	OPEN5
CANUSB-II	-20℃ ~ +70℃	OPEN5

1.5 Product Sales list

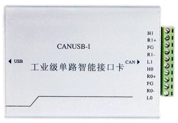
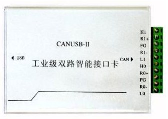
[1] CANUSB-I/II interface module ;
[2] USB cable ;
[3] CD-ROM.(Datasheet, Drivers, Dll, CANUSB-I/II Tester software, VB, VB2003, VC, C++Builder, Delphi, Labview, eMbedded Visual C++ 4.0(for Wince))

1.6 Support Information

Technical Support Mail: sales@buenoptic.com
Web site: http://www.buenoptic.com

1.7 PC-CAN Interface Card List

Part Number	Appearance	CAN Channel	Operating temperature	Interface

CANUSB-I (CAN TO USB Adapter)		1	-20℃ ~ +70℃	OPEN5
CANUSB-II (CAN TO USB Adapter)		2	-20℃ ~ +70℃	OPEN5
PCI-5001 (Passive PCI CAN Card)		1	-20℃ ~ +70℃	DB9 male
PCI-5002 (Passive PCI CAN Card)		2	-20℃ ~ +70℃	DB9 male
CANET-I (CAN TO Ethernet Converter)		1	-20℃ ~ +70℃	OPEN5
CANET-II (CAN TO Ethernet Converter)		2	-20℃ ~ +70℃	OPEN5

Chapter 2 Product Installation

2.1 Driver Installation

When you connect device with USB port of pc or notebook first, the windows will then auto run an installation wizard called "new hardware is found" after the hardware is detected. You can find the driver in CD ROM (CDROM₩Windows₩CANUSB₩Drivers₩). In addition, you can also install the driver from "My computer->Device Manager" by yourself.

From the Device Manager you can see that "CAN TO USB Device" is already in the current hardware list in the "Universal Serial Bus Devices" option.

2.2 CAN-bus connector

CANUSB-I/II module integrates two CAN-bus channels, while CANUSB-I integrates one. The pin signal definitions see Figure 2-1, Figure 2-2and Table 2-1.

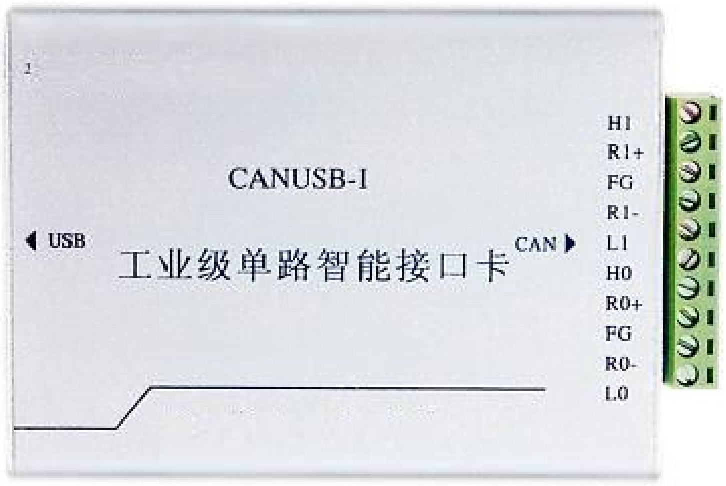


Figure 2-1：CANUSB-I CAN interface module

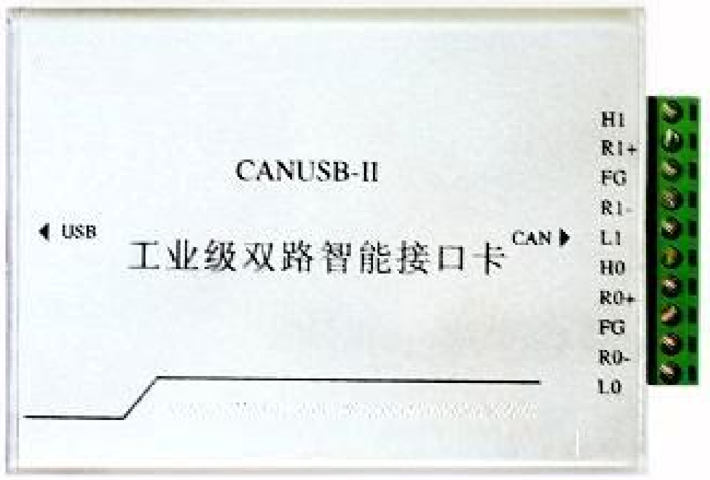


Figure 2-2：CANUSB-II CAN interface module

Table 2-1：Pin Description

(Note: CANUSB-II integrates two CAN-bus channels, CANUSB-I integrates one CAN-bus channel)

Pin	Channel	Name	Function
1	CAN0	CANL0	CAN bus Signal L
2		R0－	Terminal resistor R-(internally connected to CANL0)
3		FG	Shield cable (FG)
4		R0＋	Terminal resistor R+(internally connected to CANH0)
5		CANH0	CAN bus Signal H
6	CAN1	CANL1	CAN bus Signal L
7		R1－	Terminal resistor R-(internally connected to CANL1)
8		FG	Shield cable (FG)
9		R1＋	Terminal resistor R+(internally connected to CANH1)
10		CANH1	CAN bus Signal H

2.2 CAN bus connections

To connect CANUSB-I/II module to the CAN-bus, user only need to connect CAN_L and CAN_L, CAN-bus network adopts straight-line topology, and two terminal 120Ω resistances need to be installed on the two bus terminals. If the number of nodes larger than 2, the 120Ω resistance is not necessary to be installed on the middle node. The length of branch connection should not be longer than 3 meters. The connections for the CAN-bus are shown in Figure 2-3.

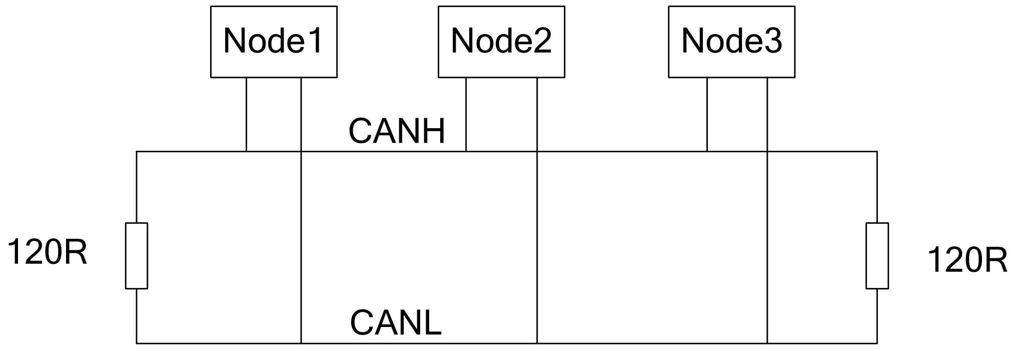
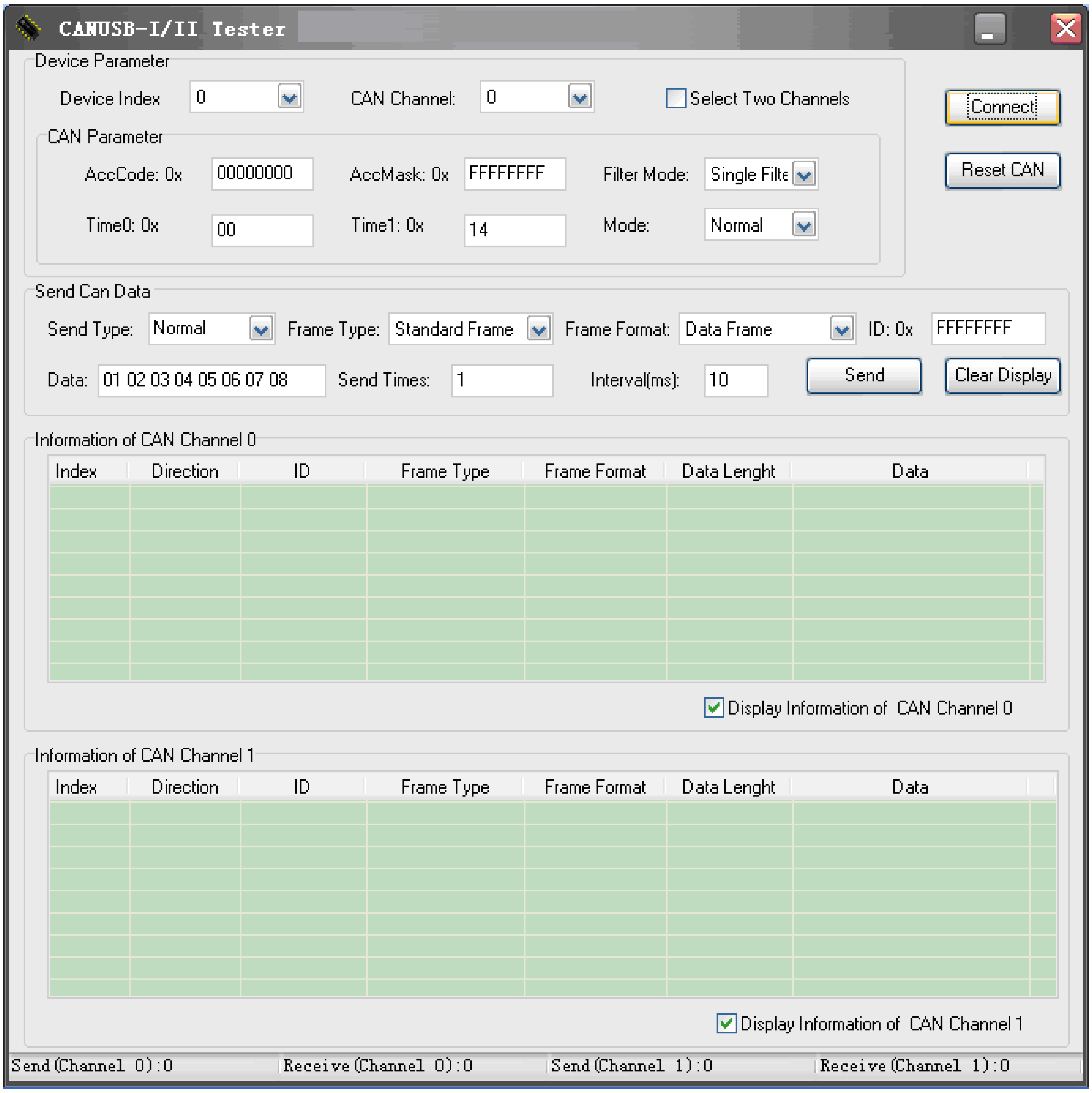


Figure 2-3: The topology for CAN-bus Network

2.3 Using and Testing the CANUSB-I/II

2.3.1 Introduction for the Tester Software.



Device Index: 0; CAN Channel: 0; Select Two Channels: unchecked
AccCode: 0x 00000000; AccMask: 0x FFFFFFFF; Filter Mode: Single Filte
Time0: 0x 00; Time1: 0x 14; Mode: Normal
Send Type: Normal; Frame Type: Standard Frame; Frame Format: Data Frame; ID: 0x FFFFFFFF
Data: 01 02 03 04 05 06 07 08; Send Times: 1; Interval(ms): 10

```
Send(Channel 0):0    Receive(Channel 0):0    Send(Channel 1):0    Receive(Channel 1):0
```

(1) Main View ->Device Parameter

Device Index	0: The first device. 1: The second device.
CAN Channel	Select the CAN Channel, and you can check the “Select the two channels” if you want to open the two CAN channels at the same time.

AccCode	AccCode corresponds to four registers in SJA1000. You can refer the datasheet of NXP SJA1000 controller.
	ACR0=(AccCode>>24)&0xFF
	ACR1=(AccCode>>16)&0xFF
	ACR2=(AccCode>>8)&0xFF
	ACR3=(AccCode>>0)&0xFF

AccMask	AccCode corresponds to four registers in SJA1000. You can refer the datasheet of NXP SJA1000 controller.
	AMR0=(AccMask >>24)&0xFF
	AMR1=(AccMask >>16)&0xFF
	AMR2=(AccMask >>8)&0xFF
	AMR3=(AccMask >>0)&0xFF

Filter Mode	Single Filter or Dual Filter.
Timer0	Baud rate timer 0.
Timer1	Baud rate timer 1.
Mode	Normal or Listen only.

(2) Main View ->Send CAN Data

Send Type	Normal or Self TX-RX.
Frame Type	Standard Frame or Extend Frame.
Frame Format	Data Frame or Remote Frame.
ID	CAN ID (For standard frame is 11bit, and for extend frame is 29bit) .
Data	Data for CAN packet.
Send Times	Packets need to send.
Interval(ms)	Interval of CAN packets.

You can select the message frame type, fill ID value and data value. When clicks the “Send”, a CAN message will be sent to CAN bus.

(3) Main View -> Information of CAN Channel

Display information of CAN Channel 0 and CAN Channel 1.

2.3.2 Using the CANUSB-I/II

(1) Connect the device to the CAN bus which you need to send and receive CAN packet.
(2) Connect the device to your computer through USB interface.
(2) Run CANUSB-I II Tester Advanced.exe. (CDROM ₩Windows₩CANUSB₩Tester₩).
(3) Setup the parameter on the Tester Software (The key parameters are the Time0 and Time1 which are in accordance with the CAN Bus baud rate).
(4) Click the “Connect” to open the device and initialize the CAN controller.
(5) Now you can send and receive any CAN packet based on CAN2.0A or CAN2.0B.

2.3.3 Self-Reception Test

After getting CANUSB-I/II Adapter, users can test the product by the following steps :

(1) Disconnect CANUSB-I/II from any other CAN node.
(2) Connect a 120ohm resistor between R0+ and R0- for channel 0 or R1+ and R1- for channel 1.
Note: In self-reception mode, you need the terminal resistor.
(3) Connect the USB port of PC to CANUSB-I/II, the Red LED is on, indicating that the power supply is working and the system is being initialized.
(4) If CANUSB-I/II is first used, you should install driver which can be found in CDROM (CDROM₩Windows₩CANUSB₩Drivers₩). After that, Green LED begins to flicker, indicating CAN controller enters into the normal working status and CANUSB-I/II connects to the PC successfully.

(5) Run CANUSB-I II Tester Advanced.exe. (CDROM ₩Windows₩CANUSB₩Tester₩)

(6) Click the “Connect” to open the device and initialize the CAN controller and change the Send Type to “Self TX-RX”

(7) Click the “Send” to Send message, and see whether you can receive the same message you just sent out.

If you can’t operate successfully, please check above steps carefully. And if it is still not work, please contact us to get help.

Chapter 4 Software development

If users intend to make a program for their own application, they need to read following descriptions very carefully, and refer the demo source code.

Develop files include CAN_TO_USB.h, CAN_TO_USB.lib (For VC) , CAN_TO_USBbc.lib (For BC), SiUSBXp.dll, CAN_TO_USB.dll.

We provides examples for VB, VB2003, VC, C++Builder, Delphi, Labview, eMbedded Visual C++ 4.0(for Wince), which make it convenient for user to develop programs.

3.1 Data Structure of Library

3.1.1 VCI_INIT_CONFIG

```
typedef struct _INIT_CONFIG{
 DWORD   AccCode;
 DWORD   AccMask;
 DWORD   Reserved;
 UCHAR   Filter;
 UCHAR   Timing0;
      UCHAR      Timing1;
 UCHAR   Mode;
}VCI_INIT_CONFIG,*P_VCI_INIT_CONFIG;
```

AccCode	acceptance code for filter
AccMask	mask code for filter
Reserved	not used
Filter	filter mode ,single or double
Timing0	timer0 (BTR0)
Timing1	timer1 (BTR1)
Mode	work mode 0: normal work, 1: listen only

Timing0 and remark Timing1 is used for setting CAN baud rate. The following table is about setting of 15 kinds of common baud rates

CAN Baud rate	Timer0	Timer1
5Kbps	0xBF	0xFF
10Kbps	0x31	0x1C
20Kbps	0x18	0x1C
40Kbps	0x87	0xFF
50Kbps	0x09	0x1C
80Kbps	0x83	0Xff
100Kbps	0x04	0x1C
125Kbps	0x03	0x1C
200Kbps	0x81	0xFA
250Kbps	0x01	0x1C

400Kbps	0x80	0xFA
500Kbps	0x00	0x1C
666Kbps	0x80	0xB6
800Kbps	0x00	0x16
1000Kbps	0x00	0x14

3.1.2 VCI_CAN_OBJ

```
typedef   struct   _VCI_CAN_OBJ{
        DWORD   ID;
 BYTE       SendType;
      BYTE  ExternFlag;
 BYTE       RemoteFlag;
 BYTE       DataLen;
 BYTE       Data[8];
}VCI_CAN_OBJ,*P_VCI_CAN_OBJ;
```

ID	packet ID, 4 bytes
SendType	0: Normal Send,1 :self reception
RemoteFlag	remote frame or not
ExternFlag	extended frame or not
DataLen	data length(<=8), it is the length of data
Data	data of packet

3.2 Function description

[1] Open the device.

BOOL   __stdcall   VCI_OpenDevice(DWORD DevIndex);

DevIndex   Device index.0: The first device.1: The second device.

Return value 1: Success, 0: Fail

[2] Close the device

BOOL   __stdcall   VCI_CloseDevice(DWORD DevIndex);

DevIndex   Device index.0: The first device.1: The second device.

Return value 1: Success, 0: Fail

[3] Initialize can.

BOOL   _stdcall   VCI_InitCAN(DWORD DevIndex,DWORD CANIndex, P_VCI_INIT_CONFIG InitConfig);

DevIndex   Device index.0: The first device.1: The second device.

CANIndex   Can channel.0: The first channel.1: The second channel

InitConfig   Init parameters structure.

InitConfig->AccCode	AccCode corresponds to four registers in SJA1000.
InitConfig->AccMask	ACR0=((InitConfig->AccCode)>>24)&0xFF

	AMR0=((InitConfig->AccMask)>>24)&0xFF
InitConfig->Reserved	reserved
InitConfig->Filter	Filter mode, 0- single filter, 1-dual filter
InitConfig-> timer0	Baud rate timer 0
InitConfig-> timer1	Baud rate timer 1
InitConfig->Mode	0:Normal mode.1:Listen only

Return value 1: Success, 0: Fail

[4] Reset can.

BOOL __stdcall VCI_ResetCAN(DWORD DevIndex , DWORD CANIndex);

DevIndex	Device index.0: The first device.1: The second device.
CANIndex	Can channel.0: The first channel.1: The second channel
Return value	1: Success, 0: Fail

[5] Send can packet.

BOOL __stdcall VCI_Transmit(DWORD DevIndex , DWORD CANIndex, P_VCI_CAN_OBJ *pSend);

DevIndex	Device index.0: The first device.1: The second device.
CANIndex	Can channel.0: The first channel.1: The second channel
pSend	Packet to Send.
Return value	1: Success, 0: Fail

[6] Receive can packet.

DWORD __stdcall VCI_Receive(DWORD DevIndex , DWORD CANIndex, PVCI_CAN_OBJ pReceive , DWORD Len , DWORD WaitTime);

DevIndex	Device index.0: The first device.1: The second device.
CANIndex	Can channel.0: The first channel.1: The second channel
pReceive	Receive packet.
Len	Number of packets you need to read.
WaitTime	Time out.
Return value	Actually packets number returned。

[7] Get the number packet in the internal buffer

DWORD __stdcall VCI_GetReceiveNum(DWORD DevIndex,DWORD CANIndex);

DevIndex	Device index.0: The first device.1: The second device.
CANIndex	Can channel.0: The first channel.1: The second channel
Return value	number of the packets in the internal buffer

[8] Clear the internal buffer.

BOOL __stdcall VCI_ClearBuffer(DWORD DevIndex,DWORD CANIndex);

DevIndex	Device index.0: The first device.1: The second device.
CANIndex	Can channel.0: The first channel.1: The second channel
Return value	1: Success, 0: Fail

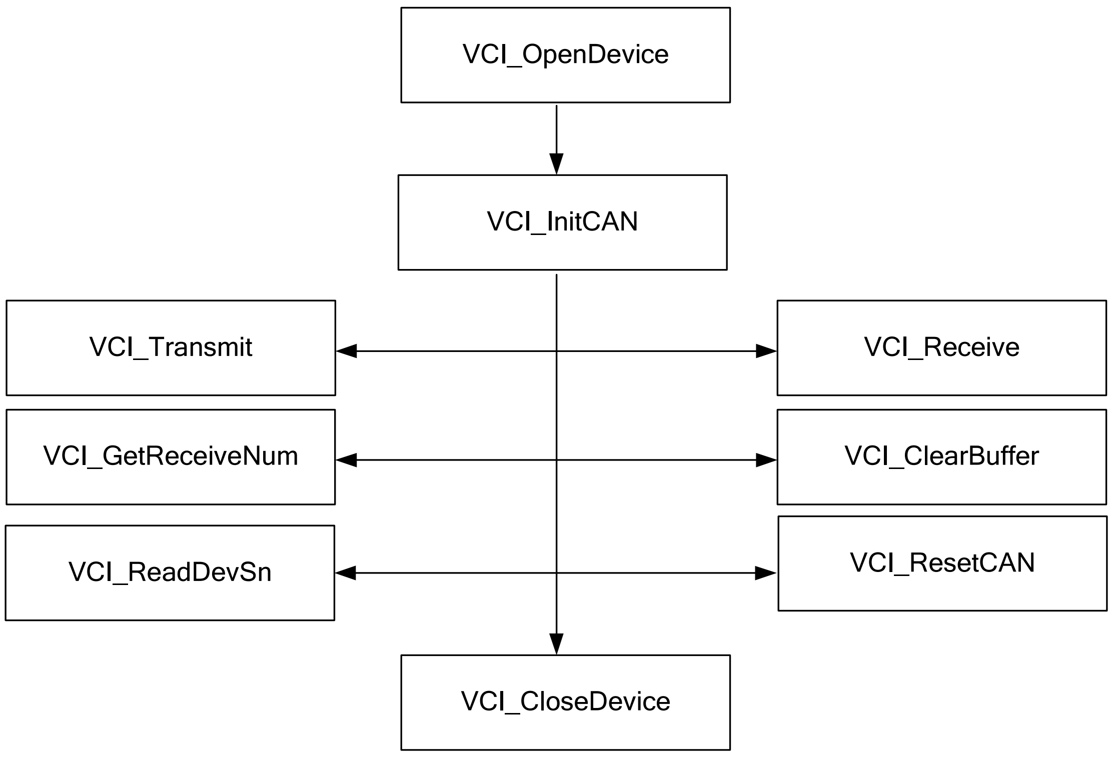
[9] Read the serial number of the device.

BOOL _stdcall VCI_ReadDevSn(DWORD DevIndex, PCHAR DevSn);

DevIndex	Device index.0: The first device.1: The second device.

DevSn	Serial Number of the device
Return value	1: Success, 0: Fail

3.4 Interface library function using flow



Chapter 5 Common Problems

1. A “Open Device Fail” message displays in the test software.

The reason causing this kind of problems maybe: the CANUSB-I/II was installed incorrectly. To solve this problem, open the “System” from the control panel to check the CANUSB-I/II device property. If a “!” or “?” was found, then check the hardware/software collisions and reinstall the driver.

2. Is the 120Ω terminal resistance absolutely necessary?

It is recommended to use the 120Ω resistance to absorb the endpoint reflection and provide stable physical links. To successfully take a self-transmitting and self-receiving test, the 120Ω resistance is needed to form a loop.

3. Is it possible to install more than one CANUSB-I/II adapter to one PC?

Current CANUSB-I/II version is able to support synchronous operation for up to 8 adapters.

4. What is the max data transfer rate for CANUSB-I/II adapter?

The single CAN channel in CANUSB-I/II adapter supports a max CAN bus transfer rate up to 6500fps (data frame for extended frame).

5. How to configure the Baud rates for CANUSB-I/II adapter?

Appendix A provides a set of frequently used Baud rates. For more Baud rates, please refer to SJA1000 data sheet. Note: the clock frequency for the CAN controller of CANUSB-I/II adapter is 16MHz, which should be considered for calculation when user is going to use a customized Baud rate.

Appendix A: SJA1000 standard Baud rate

(Oscillator Frequency=16MHz)

CAN Baud rate	BTR0 or Timer0(Hex)	BTR1 or Timer1(Hex)
5Kbps	0xBF	0xFF
10Kbps	0x31	0x1C
20Kbps	0x18	0x1C
40Kbps	0x87	0xFF
50Kbps	0x09	0x1C

80Kbps	0x83	0Xff
100Kbps	0x04	0x1C
125Kbps	0x03	0x1C
200Kbps	0x81	0xFA
250Kbps	0x01	0x1C
400Kbps	0x80	0xFA
500Kbps	0x00	0x1C
666Kbps	0x80	0xB6
800Kbps	0x00	0x16
1000Kbps	0x00	0x14