

# USBCAN Interface

**USBCAN-2A/II/I Intelligent CAN-bus Interface UM010206**

Rev 2.00 Date: 2/8/2007

Product Data Sheet

## Overview

USBCAN-2A/II/I is an intelligent PC-CAN interface module which is compatible with USB1.1 and supports one or two CAN channels. Each channel integrates independent electrical isolation circuit. And with the module and USB port, a control node of CAN-bus can be formed by connecting the PC to the CAN network.

USBCAN-2A/II/I supports CAN2.0A/B protocol and different Baud rates from 5Kbps to 1Mbps, provides driver and software for multi-systems, truly satisfying all application requirements for customers and providing high reliability and efficiency solutions for industrial communications CAN network.

## Applicable Field

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

## Features

- ◆ PC interface supports USB1.1 Protocol and can be worked with USB2.0;
- ◆ Integrated with one or two CAN-bus interface, supports CAN2.0A and CAN2.0B protocols, fully compatible with ISO/DIS 11898 standard;
- ◆ Programmable CAN-bus communication Baud rates from 5Kbps to 1Mbps;
- ◆ (DC+9V~+25V, 200mA) from USB bus or external power supply;
- ◆ Adopts electrical isolation, the isolation voltage is : 2500Vrms;
- ◆ Max data flow for a single channel: 3000 fps (standard frame);
- ◆ Supports plug and play;

## Order Info

TYPE	Temperature	Interface
USBCAN-2A	-25℃~+85℃	OPEN5
USBCAN-II	0℃~+70℃	DB9
USBCAN-I	0℃~+70℃	DB9

## Typical Applications

### Hardware

CAN controller: SJA1000T

CAN transceiver: PCA82C251

### Interface

Bus: USB 1.1 (12Mbps)

### Performance

Baud rate: 5Kbps ~ 1Mbps

Transfer rate: 3000 fps (standard frame)

### Configuration

USB: interrupt and I/O are distributed by BIOS

Operation mode: normal, listen only, self-transmitting and receiving

API: VCI function library

### Operation system supported:

Windows98/Me/2000/XP/2003

Linux 2.4、Linux 2.6

### Tools supported:

Testing tool for communication CAN: ZLGCANTest

OPC server: ZOPC Server

ICAN test tool: iCANTest

Virtual serial server: ZVCom

### Power supply and environment

Power supply requirement: USB power supply, DC5V@200mA

External power supply: DC9V@200mA (Optional)

Operating temperature: -25℃ ~ +85℃

Storage temperature: -40℃ ~ +85℃

[www.embedcontrol.com](http://www.embedcontrol.com)
**ZHIYUAN Electronics Co., Ltd.**  
**Industrial Control Department**




## Revision History

Version	Rev. Date	Modifications
Rev 1.0	2003-5-8	Initial version
Rev 1.2	2003-11-26	Unify the format and add contents to the appendix
Rev 2.0	2006-12-22	Unify the format and add contents for the new model: USBCAN-2A interface module



## Sales Info & Technical Supports

---

### ● Declaration

The copy rights of the USBCAN-2A/II/I intelligent CAN interface module and the relative software belong to Guangzhou ZHIYUAN Electronics Co. Ltd., and the property rights are protected by the civil law, which shall not be used or copied by any other companies, organizations, agents or individuals without authorization by ZHIYUAN Electronics.

Guangzhou ZHIYUAN Electronics Co., Ltd. reserves the right to modify this user manual at any time without notice.

### ● Sales Information:

If you need to purchase this product, you can call our sales department at working time (8:30AM~11:50AM, 1:30PM~5:30PM from Monday to Friday, and 8:30AM~11:50AM Saturday morning).

Telephone: +86 (020) 2264-4249, 22644399      FAX: +86 (020) 3860-1859

Address: Floor 2, No. 3 Building, Huangzhou Industrial Estate, Chebei Road, Tianhe, Guangzhou, CHINA

Postcode: 510660

### ● Technical supports:

To obtain latest news on our products, please visit our website:

[http:// www.embedcontrol.com](http://www.embedcontrol.com)

If you need technical supports, please call our service department at working time or write us an E-mail.

Support for CAN-bus: +86 (020)22644253

Support for Ethernet: +86 (020)22644385

Support for ARM: +86 (020)22644383

Support for USB: +86 (020)22644386

Visit <http://www.zlqmcu.com> for other products' supports



# Content

---

<b>Revision History .....</b>	<b>I</b>
<b>Sales Info &amp; Technical Supports .....</b>	<b>II</b>
<b>Content .....</b>	<b>III</b>
<b>Chapter 1: Features .....</b>	<b>1</b>
1.1 Product Overview .....	1
1.2 Parameters .....	1
1.3 Appearance of the product .....	2
1.4 Typical applications .....	3
<b>Chapter 2: Installation .....</b>	<b>4</b>
2.1 Power supply .....	4
2.1.1 External power supply .....	4
2.1.2 USB bus power supply .....	4
2.2 CAN-bus connector .....	5
2.3 Signal Indicator LED .....	6
2.4 System connections .....	8
2.4.1 CAN bus connections .....	8
2.4.2 Bus Terminal Resistance .....	8
2.4.3 USB Bus Connections .....	9
<b>Chapter 3: Drivers Installation .....</b>	<b>10</b>
3.1 Your first time to install the driver into the system .....	10
3.2 Check if the hardware is successfully installed .....	11
3.2.1 Open WINDOWS devices manager .....	11
3.2.2 Check whether the hardware is successfully installed .....	12
<b>Chapter 4: Common Problems .....</b>	<b>13</b>
<b>Chapter 5: Check and Maintenance .....</b>	<b>16</b>
<b>Appendix A: The Frame Format of CAN2.0 Protocol .....</b>	<b>18</b>
<b>Appendix B: SJA1000 standard Baud rate .....</b>	<b>20</b>
<b>Appendix C: Configuration for CAN Message Filter .....</b>	<b>21</b>

# Chapter 1: Features

---

## 1.1 Product Overview

USBCAN-2A/II/I module is an intelligent CAN-bus communication interface that compatible with USB1.1 bus and supports one/two CAN channels. 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.

There are three independent types for USBCAN-2A/II/I module:

- USBCAN-2A: dual-channel industrial level CAN interface module
- USBCAN-II: dual-channel intelligent CAN interface module
- USBCAN-I: signal channel intelligent CAN interface module

**Notes:** \* USBCAN-2A interface module is the industrial level enhanced version of USBCAN-II.

USBCAN-2A/II/I intelligent CAN interface module is a powerful tool for CAN-bus products development and CAN-bus data analysis. It has a small size and supports plug and play function. It is also the best choice for the users who are using portable operation systems.

USBCAN-2A/II/I 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.

USBCAN-2A/II/I interface module supports Win9x/Me and Win2000/XP operation systems, as well as Linux2.4 and Linux2.6. USBCAN-2A/II/I provides an uniform application programmable interface and the complete demonstration code, including VC, VB, Delphi and C++ Builder, which make it convenient for user to develop programs.

USBCAN-2A/II/I also supports OPC interface and can be applied to a configuration software supporting OPC. In addition, test software ZLGCANTest is also provided for CAN-bus message to implement transmit/receive and monitoring functions.

## 1.2 Parameters

- PC interface supports USB1.1 protocol and is USB 2.0 compliant;
- 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;

- Adopts USB bus power supply or external power supply (DC+9V~+25V, 400mA);
- Adopts electrical isolation, the isolation voltage is : 2500Vrms;
- Max data flow for a single channel: 3000 fps (standard frame);
- Supports Win9x/Me, Win2000 and WinXP operation systems;
- Supports Linux2.4 and Linux2.6 operation systems;
- Supports ZLGCANTest test software and ZOPC server software;
- Small size, plug and play;
- Physical size: (length) 115mm \* (width) 76mm \* (height) 30mm.

### 1.3 Appearance of the product



Figure 1-1 : USBCAN-2A dual-channel industrial level CAN interface module



Figure 1-2 : USBCAN-II dual-channel intelligent CAN interface module



Figure 1-3 : USBCAN- I single channel intelligent CAN interface module



## 1.4 Typical applications

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



## Chapter 2: Installation

---

### 2.1 Power supply

USBCAN-2A/II/I intelligent CAN interface module is characterized by small size and the plug and play function. It can be powered by either PC USB port or an external power supply.

#### 2.1.1 External power supply

External power supply is used in case that the USB port is unable to provide enough current for USBCAN-2A/II/I interface when the PC is using an USB hub or connected with many USB terminal devices.

When the external power supply (DC+9~+25V@200mA, no polarity requirement for plugs) is connected to the POWER jack of the USBCAN-2A/II/I module, the LED SYS lights in red. Now user only need to connect the USBCAN-2A/II/I module to the PC with the attached USB cable to start the module.

#### 2.1.2 USB bus power supply

USB bus power supply can be used in many cases, for example, USBCAN-2A/II/I is the only device connected to the USB port.

Connecting the USBCAN-2A/II/I module to the PC with the attached cable can provide the module with a +5V voltage. Firstly, the LED SYS becomes to red, indicating the power supply works normally, later, the LED flickers for several times and becomes to stable green, indicating that communication has been built between the PC and the module.



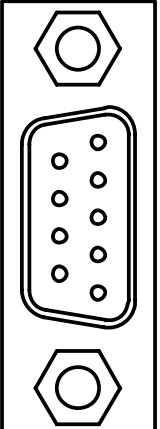
## 2.2 CAN-bus connector

USBCAN-II module integrates two CAN-bus channels, while USBCAN-I integrates one. USBCAN-II/I connects to CAN-bus network via DB9 male adapter or female adapter. CZ1 and CZ2 represent CAN channel numbers “0” and “1” respectively.

The pin signal definitions for DB9 conforms to DeviceNet and CANopen standards, see Table 2-1.


**Notes:** USBCAN-I interface module has two parallel DB9 male and female adapters, the pin sequence numbers for DB9 male adapter is opposite to those of DB9 female adapter. Please distinguish the pin numbers between the adapters.

**Table 2-1 : Signal Connections (DB9 adapter) for CAN-bus**

DB9 male adapter	Pins	Signals	Descriptions
	1	N.C.	unused
	2	CAN_L	CAN_L signal cable
	3	CAN_GND	Reference ground
	4	N.C.	unused
	5	CAN_SHIELD	Shield cable
	6	CAN_GND	Reference ground
	7	CAN_H	CAN_H signal cable
	8	N.C.	unused
	9	N.C.	unused

The CAN-bus signals from DB9 adapter can be switched to a 5-pin connector by using the optional DB9OPEN5 converter, making it much easier for connections.

**Table 2-2 : Signal Assignment for DB9OPEN5 Converter**

OPEN5 converter	Pins	Signals	Descriptions
 <div style="display: inline-block; vertical-align: top; margin-left: 10px;"> <p>DeviceNet Color-code</p> <p>V+ Red</p> <p>CANH White</p> <p>SHIELD Bare</p> <p>CANL Blue</p> <p>V- Black</p> </div>	1	V-	Network power supply (negative pole)
	2	CAN_L	CAN_L signal cable
	3	SHIELD	Shield cable (FG)
	4	CAN_H	CAN_H signal cable
	5	V+	Network power supply (positive pole)

USBCAN-2A interface module supports two CAN channels that can be operated independently, and it can be used to connect to a CAN-bus network or CAN-bus interface device. The two channels can be pinned out by using a 10-pin plug-pull AWG 14-22 connector. The pins assignment is shown as Table 2-3.

**Table 2-3 : Signal Assignment for USBCAN-2A Interface Module**

Pins	Ports	Names	Functions
1	CAN0	CAN_L	CAN_L signal cable
2		R-	Terminal resistance (internally connected to CAN_L)
3		SHIELD	Shield cable (FG)
4		R+	Terminal resistance (internally connected to CAN_H)
5		CAN_H	CAN_H signal cable
6	CAN1	CAN_L	CAN_L signal cable
7		R-	Terminal resistance (internally connected to CAN_L)
8		SHIELD	Shield cable (FG)
9		R+	Terminal resistance (internally connected to CAN_H)
10		CAN_H	CAN_H signal cable

## 2.3 Signal Indicator LED

USBCAN- I interface module has a dual-color SYS LED, RUN LED and an ERR LED to indicate the device operating status. Detailed functions are shown in Table 2-4.

**Table 2-4 LEDs for USBCAN- I Interface Module**

LED	Status	Descriptions
SYS	Red	Indicates the device initialization status
	Green	Indicates the USB interface signal
RUN	Green	CAN interface is running
ERR	Red	CAN interface error

- When USBCAN- I is power up, initialization status LED SYS becomes red, indicating that the power supply is working and the system is being initialized, otherwise it indicates that there is something wrong with the power supply or a serious system error has occurred.
- After the USB interface is well connected, USB signal LED SYS turns from red to green. USB signal LED SYS (green) begins to flicker when there is data transferring on the USB port.
- RUN LED lit in red, indicating that the initialization has finished, and CAN controller enters into the normally working status.
- ERR LED will be lit when errors occurred and off when errors are eliminated.

**Table 2-5 : LEDs for USBCAN- I Interface Module**

CAN LED status	CAN bus status
RUN&ERR are off	CAN controller disconnects to the bus
RUN is on	CAN bus works normally
RUN is on, ERR is flickering	Errors occurred on CAN-bus or data overflowed, frames may lose

USBCAN-2A/II interface module uses three two-color LEDs SYS, CAN1 and CAN2 to indicate the system running status. Detailed LED functions are shown as Table 2-6.

**Table 2-6 : USBCAN-2A/II Module LED**

LED	Status	Descriptions
SYS	Red	Indicates the system initialization status
	green	Indicates the USB interface signals
CAN1	Red	Indicates the running status for CAN1 interface
	green	
CAN2	Red	Indicates the running status for CAN2 interface
	green	

- When USBCAN-2A/II is power up, initialization status LED SYS becomes red, indicating that the power supply is working and the system is being initialized, otherwise it indicates that there is something wrong with the power supply or a serious system error has occurred.
- After the USB interface is well connected, USB signal LED SYS turns from red to green. USB signal LED SYS (green) begins to flicker when there is data transferring on the USB port.
- When the CANx controller is being initialized, the CAN interface RUN LED corresponding to the channel will alternately flicker in red and green. LED CANx becomes stable green when the initialization has finished and red when errors occurred until the errors are eliminated.

**Table 2-7 LED Status for USBCAN-2Av/II Module**

LED Status	CAN Bus Status
All off	CAN controller disconnects to the bus
Green is on	CAN bus works normally
Green is on and red flickers for one time	data overflowed, frames may lose
Green is on and red flickers	Errors occurred on CAN-bus or data overflowed, frames may lose

## 2.4 System connections

### 2.4.1 CAN bus connections

To connect USBCAN-2A/II/I module to the CAN-bus, user only need to connect CAN\_L and CAN\_H.

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 2-1.

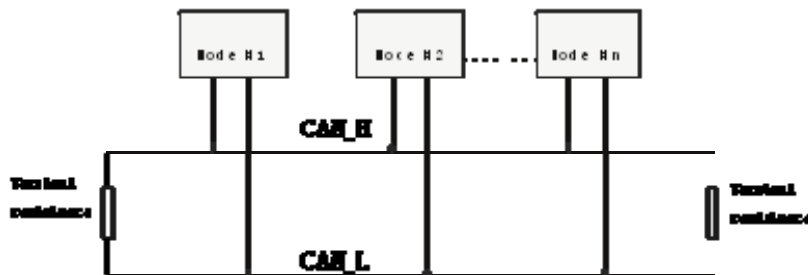


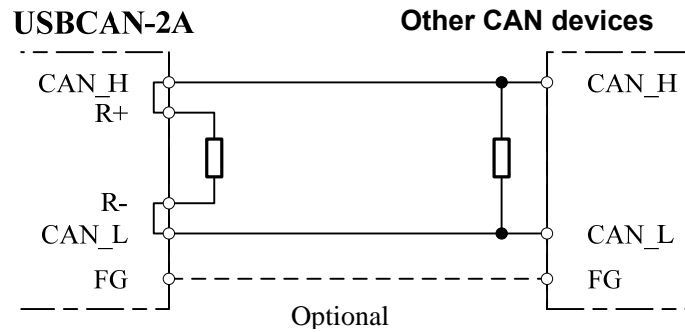
Figure 2-1 : The topology for CAN-bus Network

Note: CAN-bus cable can be either ordinary twisted-pair or shield twisted-pair. If the communication distance is longer than 1Km, then the area of section for the twisted-pair should be larger than  $\Phi 1.0\text{mm}^2$ . The particular specification depends upon the communication distance, and a longer distance usually requires a larger area.

### 2.4.2 Bus Terminal Resistance

To enhance the reliability for the CAN communication, two terminal resistances need to be respectively installed on to the two terminals on the CAN bus network, see Figure 2-1. The value for the terminal match resistance depends on the characteristic resistance of the transfer cable. For example, the characteristic resistance of the twisted-pair is 120Ω, so the terminal resistance on the bus terminal should be also 120Ω. In addition, because USBCAN-2A/II/I interface module adopts PCA82C251 transceiver, so the terminal resistance should be calculated in another way in case that the nodes on the network use different transceivers.

For USBCAN-2A/II/I intelligent interface, it is not necessary to add external terminal resistance, because each CAN channel has internally integrated a 120Ω terminal resistance. However, USBCAN-2A interface module does not have a terminal resistance internal. When USBCAN-2A interface locates on one of the nodes on the network, a 120Ω terminal resistance needs to be respectively connected to “R-” and “R+” pins. See Figure 2-2.

**Figure 2-2 : Connection Method**

Actually, CAN\_H and CAN\_L are directly connected to R+ and R- internal the device. The communication port of USBCAN-2A interface is physically compatible with DeviceNet port, but totally different in electrical structure, thus the communication port from DeviceNet device can not be directly plug to the port of USBCAN-2A interface module.

### 2.4.3 USB Bus Connections

The USB port of USBCAN-2A/II/I module supports USB1.1 protocol specification, and can be used to connect PCs that support USB1.1 and USB2.0 standards.

There are two modes for USBCAN-2A/II/I interface connecting to PC:

- Bus power supply: Directly connects to USB port on the PC by using the attached USB cable. This USB port provides a +5V voltage to USBCAN-2A/II/I module.
- External power supply: USBCAN-2A/II/I interface module connects to the PC via USB hub. If this USB hub is powered by bus, then USBCAN-2A/II/I module should use a external power supply (DC+9~+25V@200mA, no polarity requirement for plugs).

## Chapter 3: Drivers Installation

### 3.1 Your first time to install the driver into the system

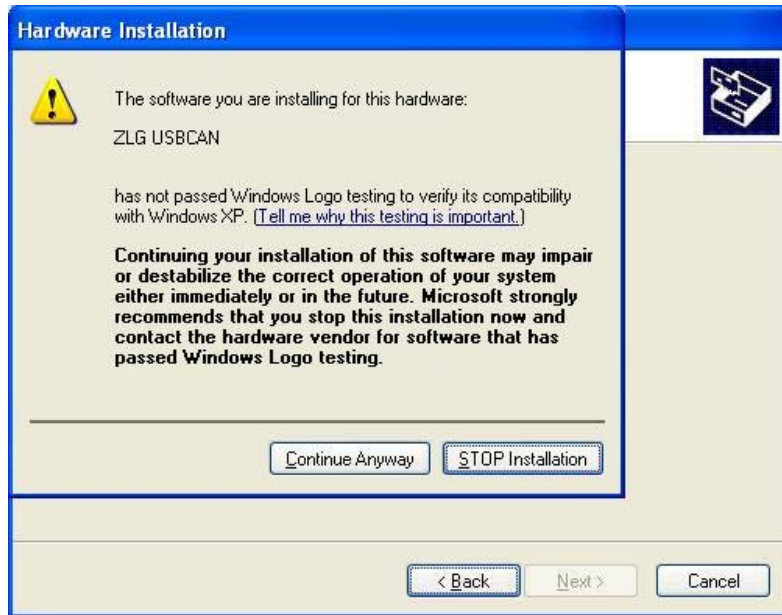
1. Copy USBCAN\Driver\usbcan.inf from the product disk to the directory windows\inf in the system, and copy usbcan.sys to windows\system32\driver;
2. When finishing the first step, connect USBCAN-2A/ II / I intelligent interface module to the PC with USB cable correctly. The Windows will then auto run an installation wizard called “new hardware is found” after the hardware is detected. Click “Next” to continue;



3. The wizard starts to search the new hardware;



4. A pop-up warning dialogue about the compatibility issue may appear when using Window XP/Windows2000. Just ignore this and click “still continue”.



5. And then the hardware will be found and installed successfully.



6. Click “Finish” to close the wizard. The LED SYS for USBCAN-2A/ II / I switches from red(indicates the initialization status) to green(indicates the USB status), indicating that the hardware is successfully installed and ready to use now.

## 3.2 Check if the hardware is successfully installed

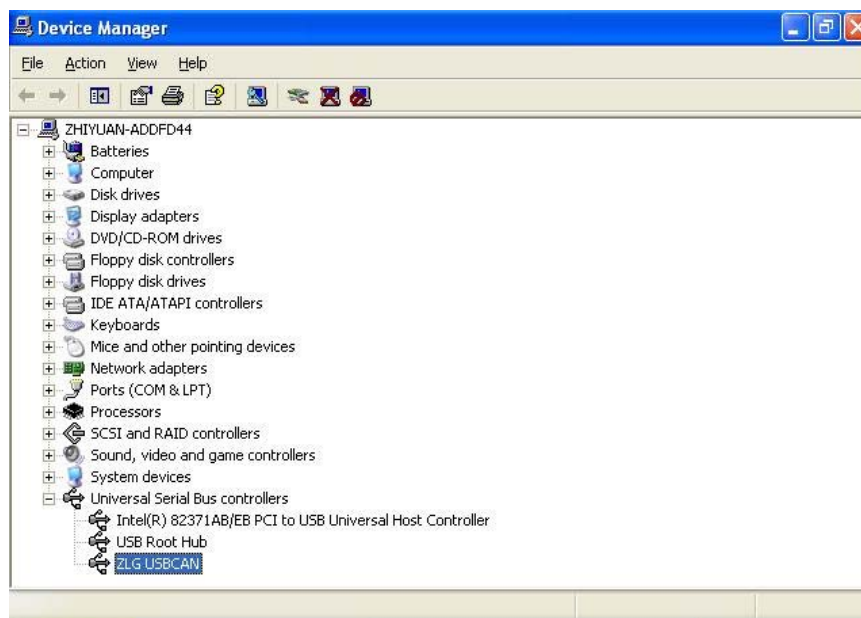
### 3.2.1 Open WINDOWS devices manager

1. Right click “My computer”;
2. Select “Property” from the pull-down menu;

3. Select the “hardware” tab;
4. click the “Device Manager” button to open the list for the current hardware devices.

### 3.2.2 Check whether the hardware is successfully installed

To find out if “ZLG USBCAN” device is already in the current hardware list in the “Universal Serial Bus Devices” option. If so, it means that the device has been successfully installed. A successful case is shown in the figure as below.



The USB LED SYS on the USBCAN-2A/II/I module will flicker in green when there is data transferring between the PC and the USBCAN-2A/II/I module.



## Chapter 4: Common Problems

---

### **1. A “fail to find device” message displays in the ZLGCANTEST test software.**

The reason causing this kind of problems maybe: the USBCAN was installed incorrectly. To solve this problem, open the “System” from the control panel to check the USBCAN device property. If a “!” or “?” was found before the USBCAN device, then check the hardware/software collisions and reinstall the USBCAN 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 USBCAN-2A/ II / I modules to one PC?**

The old version does not support the synchronous operations for multi-modules, while the current USBCAN-2A/ II / I version is able to support synchronous operation for up to 8 PC-CAN interface modules that with the same model number.

### **4. What is the max data transfer rate for USBCAN-2A/ II / I module?**

The single CAN channel in USBCAN-2A/ II / I module supports a max CAN bus transfer rate up to 5000 fps, where the frame referred here denotes the data frame for extended frame, and the transfer rate maybe faster for standard frame of remote frame. In addition, the max data flow will be limited by the PC performance.

### **5. CAN status LED is off.**

Because all the operations on USBCAN-2A/ II / I interface module are controlled by PC, CAN status LED is meaningless until PC sends a command to start the CAN communication.

### **6. System illegal operation occurs during the function call**

You should read the function specification carefully before you start to use the interface functions and make sure the parameters input is legal, especially for the pass for the pointer (address). In addition, you can also refer to the provided example program. If the problem still exists after you have done all the steps above, then you should contact our technology support department.

### **7. How to configure the Baud rates for PC-CAN interface module?**

Appendix B 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 USBCAN-2A/ II / I interface module is 16MHz, which should be considered for calculation when user is going to use a customized Baud rate.

### **8. Is there any effect on the data receiving when the system is on standby or sleep mode?**

Yes, it does have effect. All the process will stop and it will even cause a hardware receiving buffer overflow error. The device will try to prevent the system from entering into standby or sleep mode when program is running, which will ensure the system to work normally. Please disable the standby and sleep function when using USBCAN-2A/II/I interface module.

#### **9. How to deal with the errors in the application?**

There are two main kinds of errors: function call error and CAN-bus communication error.

Function call error is usually caused by parameters, for example, device number exceeds the range or the model number is incorrect, the error code corresponding to this problem returned by Win32 function GetLastError is 87. Operations to a device that have not been started are actually the operations implemented on illegal handles. The corresponding Win32 standard error codes will be provided to each particular function call. User can use GetLastError to analysis the errors during the design.

CAN-bus communication error is usually caused by CAN network or incorrect configurations, for example: The Baud rates are set inconsistently or the transmitting function is invoked before the CAN controller is started. Simple treatments have been taken to the major errors in the device driver. If further error analysis and treatment are desired, user can invoke the VCI\_ReadCANStatus function.

Besides the errors mentioned above, data overflow error should also draw your attention. There are two possible causes for this problem: (1). Software receiving buffer overflow. It means that the application could not handle the receiving data in time. In this case you should optimize the application or change the communication strategy. (2). Hardware receiving buffer overflow. It is caused by a large delay of the interrupt from the receiving-end PC and only can be solved by enhancing the PC performance or properly slowing down the transmitting speed for the other nodes a little bit.

#### **10. What should be paid attention to when opening or closing devices?**

USBCAN-2A/II/I provides two CAN ports, user is able to operate different ports within one software. USBCAN-2A/II/I can not be started in a share mode, namely, one device can not be started for more than one time by invoking the VCI\_OpenDevice function. In generally speaking, user only needs to invoke the VCI\_OpenDevice and VCI\_CloseDevice functions once, respectively on the application initialization and exit.

When closing the device, user should invoke the VCI\_ResetCAN function to disconnect the current port from the CAN bus if this port does not needed to be used any more. The device driver will not auto-invoke the VCI\_ResetCAN function to disconnect the CAN bus until the last device handle is closed.

#### **11. How to operate the module in an interrupt way?**

USBCAN-2A/II/I does not provide any interface for directly operation for interrupt, because all the interrupts have been treated in the device driver. The major reasons to operate the interrupts in the applications are: The application fails to know how long it will take for the data to arrive the device. It is unable to read the data from the buffer without a receiving message trigger. A common solution to this issue is to use

multithreading (or multitasks). Namely, start a new thread, and then repeat invoking the VCI\_Receive function to check and receive the data from buffer. Barrage scheme is implemented internal the VCI\_Receive function, the thread will be hung up when there is no data in the buffer. So it will not occupy the CPU time and the application is still available to handle other tasks.

## **12. How to make good use of the VCI\_Transmit function?**

The driver for USBCAN-2A/ II / I interface module provides about 128 frames transmitting buffer FIFO, during each VCI\_Transmit no more than 128 frames data is transmitted. The speed for the transmitting device is depended upon the computer (software and hardware) performance. In general, the continuous transmitting speed is between 1000 and 2000 fps (standard data frame 11Bytes, 1Mbps). Too fast transmitting speed may lead to data overflow in the remote receiving device, but the delay can be programmed to longer to slow down the transmitting speed properly.

Timeout limit is set for each frame. For transmitting a single frame, it is 2 seconds; for transmitting multi-frames, it is 2 sec. for the last frame and 1 sec. for each of the others. Timeout often happens when CAN bus is busy and the priority for the current node is low but is not caused by function call or communication errors. User can re-transmit the data by programming (It is seldom for timeout situations to happen on the mid-low speed network). Therefore it should be ensured that the CAN bus occupation rate does not exceed 60-70% during system design.

## **13. How to make good use the VCI\_Receive function?**

The device driver provides 100000 frames software receiving buffer, offering the programmer with sufficient process time. When the data receiving overflowed, the device program will invoke VCI\_ResetCAN function to reset the CAN bus and set the data overflow interrupt flag bit of the CAN status. Note: Both software buffer overflow and CAN controller hardware buffer overflow use this flag bit.

The “Wait” parameter provided by the receiving function can be used for multithreading. A barrage function is inserted into the receiving function, and the definition for its parameter “Wait” is the same to that of “Win32/WaitForSingleObject/dwMilliseconds” (please refer to the Win32API specification), specifying a timeout value (in millisecond) for the VCI\_Receive function.

When “Wait” is 0, a function call will immediately returns the number of the frames read currently and successfully, or zero if the receiving buffer is empty. When “Wait” is non-zero, it immediately returns the number of the frames that successfully read if there is data in the receiving buffer, or if the receiving buffer is empty, it will not return the number of the read frames until the specified timeout limit is matched or data is received. “Wait” being set to 0xFFFFFFFF means it will keep waiting until there is data received. It is suggested not to set the parameter “Wait” too larger.

When nFrames is equal to zero, it returns an inform message and it is not necessary to read the receiving buffer. Note: If the function VCI\_Receive is invoked in the main thread and “Wait” is non-zero, the application may loss responds temporarily. In query mode, “Wait” should be normally set to zero.

## Chapter 5: Check and Maintenance

The major electrical components for USBCAN-2A/II/I are semiconductor devices, which have long life cycle but are also easy to aging if used under incorrect environment. Regular check should be taken to ensure the devices meet all requirements. It is recommended to check the devices at least once per 6 months or one year. More frequently checks are preferred in terrible environment.

Follow the specification below to find out and solve the problems. If the problems still exist, please contact to Guangzhou ZHIYUAN Electronics Co. Ltd.

NO.	Item	Check	Standard	Actions
1	Power supply	Check the voltage wave of the power supply	The allowable range for the voltage wave for the external power supply is: +9V~+25V DC or The power supply from the USB is +5V DC	Check the power supply with the voltage meter and take necessary actions to make the voltage wave in the allowable range
3	Surrounding environment	Check the surrounding temperature (including the temperature within the close environment)	-25°C ~ +85°C	Check the temperature with thermometer to ensure that the environment temperature is in the allowable range
		Check the surrounding humidity (including the humidity within the close environment)	Relative humidity should be from 10% to 90% without air conditioner	Check the humidity with hygrometer and ensure that the surrounding humidity is in the normal range
		Check if there is dust, salt and metal scraps	No dust, salt or metal scraps	Clean and protect the devices
		Check if there is water, oil or chemical spray adhering to the device.	No spray adhering to the device	Clean the devices if necessary
		Check if there are erosive or combustible gas	No erosive or combustible gas	Check by nose or a sensor
		Check the shake-impact	The shake-impact accords with in prescribed limit	Install an interleaver or other shock absorption equipments
		Check the noise source	There is no significant	Segregate the device

NO.	Item	Check	Standard	Actions
		near the device	noise signal source	from the noise source or protect the device
4	Installation and wiring	Check the connection for each unit	Firmly connected	Push the connectors to one side and lock them
		Check if the cable connector is fully plugged and locked	Firmly connected	correct the incorrect installations for connectors
		Check if there is loose screw in the external connection	Firmly connected	Tighten the screws with a screwdriver
		Check the ZD connector	There is enough space between the connectors	Examine with the unaided eye and adjust if necessary
		Check if the external cable is damaged	No damage	Examine with the unaided eye and replace the cable if necessary

## Appendix A: The Frame Format of CAN2.0 Protocol

### CAN2.0B standard frame

The length for CAN standard frame message is 11 bytes, including two parts: message and data. The first 3 bytes are used for message.

	7	6	5	4	3	2	1	0
Byte 1	FF	RTR	×	×	DLC (Data length)			
Byte 2	(message identifier)				ID.10—ID.3			
Byte 3	ID.2—ID.0			×	×	×	×	×
Byte 4	Data 1							
Byte 5	Data 2							
Byte 6	Data 3							
Byte 7	Data 4							
Byte 8	Data 5							
Byte 9	Data 6							
Byte 10	Data 7							
Byte 11	Data 8							

Byte 1 is frame information. Byte 7 (FF) denotes frame format, FF=0 in the standard frame. Byte 6 (RTR) denotes the frame type, 0 for data frame and 1 for remote frame. DLC stands for the data length in data frame mode.

Byte 2 and 3 are message identifiers, 11 bits are effective.

Byte 4 to 11 is the data for data frame, invalid for remote frame.

### CAN2.0B extended frame

The length for CAN extended frame message is 13 bytes, including two parts: message and data. The first 5 bytes are used for message.

	7	6	5	4	3	2	1	0
Byte 1	FF	RTR	×	×	DLC（Data length）			
Byte 2	（Message identifier）ID.28—ID.21							
Byte 3	ID.20—ID.13							
Byte 4	ID.12—ID.5							
Byte 5	ID.4—ID.0					×	×	×
Byte 6	Data 1							
Byte 7	Data 2							
Byte 8	Data 3							
Byte 9	Data 4							



	7	6	5	4	3	2	1	0
Byte 10	Data 5							
Byte 11	Data 6							
Byte 12	Data 7							
Byte 13	Data 8							

Byte 1 is frame information. Byte 7 (FF) denotes the frame format, FF=1 for extended frame. Byte 6 (RTR) denotes the frame type, 0 for data frame and 1 for remote frame. DLC stands for the data length in the data frame.

Byte 2 and 5 are message identifiers, the higher 29 bits are effective.

Byte 6 to 13 is the data for data frame, invalid for remote frame.

## Appendix B: SJA1000 standard Baud rate

No.	Baudrate (Kbps)	Oscillator Frequency=16MHz	
		BTR0 (Hex)	BTR1 (Hex)
1	5	BF	FF
2	10*	31	1C
3	20*	18	1C
4	40	87	FF
5	50*	09	1C
6	80	83	FF
7	100*	04	1C
8	125*	03	1C
9	200	81	FA
10	250*	01	1C
11	400	80	FA
12	500*	00	1C
13	666	80	B6
14	800*	00	16
15	1000*	00	14

Note: Those with “\*” are the Baud rates that recommended by CIA union.



## Appendix C: Configuration for CAN Message Filter

The CAN message filter of the converter is designed based on the PeliCAN mode of the CAN controller SJA1000 (made by PHILIPS). SJA1000 filter is composed of 4 sets (4 Bytes) of acceptance code registers (ACR) and 4 sets (4 Bytes) of acceptance mask registers (AMR). The value of ACR is the preset acceptance code, and the value of AMR indicates that if the corresponding value of ACR is used for acceptance filtering.

When SJA1000 is in some certain modes, part of registers in the filter will be left unused. For convenience, we only care about the actual value for the filter and discard the unnecessary value.

The general rules for filtering are: Every acceptance mask bit is corresponding to each acceptance code bit respectively. When the mask bit is 1 (namely set to irrelative), then no matter if the corresponding acceptance frame ID bit is the same to the corresponding acceptance code bit or not, it will denotes an acceptance. But when the mask bit is 0 (namely set to relative), it will not indicate an acceptance unless the two correspond bits above have the same value. And CAN controller receives this frame message only when all the bits denote acceptance.

There are two filter configuration modes: single filter and dual filter. And the filtering for standard frame and extended frame is a little different. Open all the filter functions under “customize filter mask code” in the configuration software.

### 1. Single filter configuration

This kind of filter configuration can be defined as a long filter. The relationships for the corresponding bits between the filter byte and the message byte are dependent on the current frame format.

**Standard frame:** When the frame format is standard, only part of the data bits (lower 11 bits) from the first two bytes in ACR (ACR3 and ACR4) will be used to store the filter acceptance code. Also, filter mask code only use the lower 11 bits from AMR3 and AMR4.

When the bits in AMR are 0 (relative), if the corresponding bits between ACR and acceptance frame ID (eg.ACR1.0 and AMR1.0 and ID.00) are the same, it indicates “acceptable” (logic 1), otherwise it indicates “unacceptable” (logic 0). When the bits in AMR are 1, it always indicates “acceptable” (logic) regardless of the discussions above.

For a successfully received message, receiving signals must be sent after comparing each single bit. See Figure 5-1.

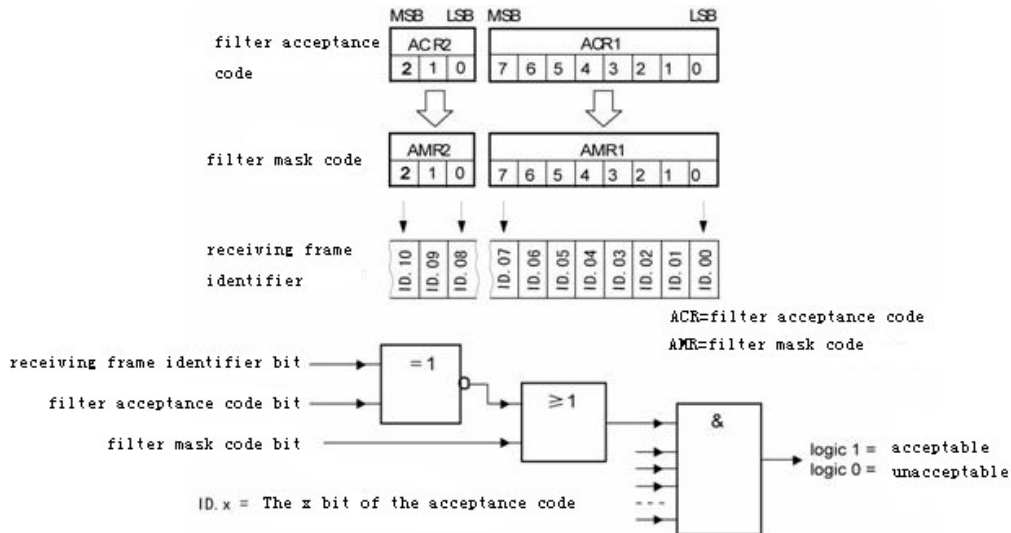


Figure 5-1 : Sketch for standard frame single filter

**Extended frame:** When the frame format is extended, the length for the frame identifier is 29 bits, so the lower 29 bits of the four bytes of ACR are used to store the filter acceptance code. And it is similar to the AMR.

The acceptance logic relationship is the same to that of standard frame. See Figure 5-2.

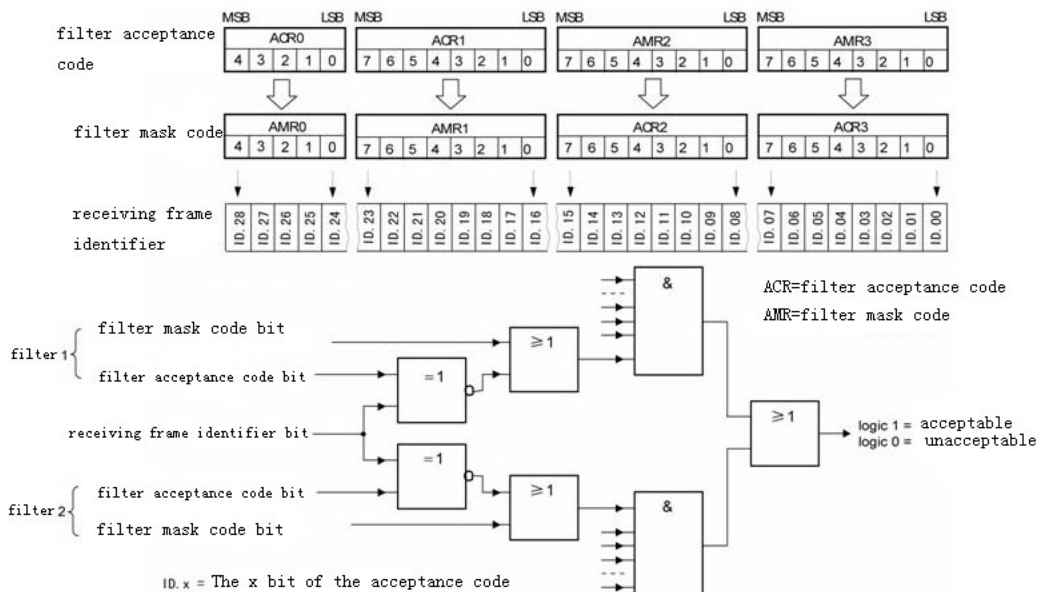


Figure 5-2 : Sketch for extended frame single filter

## 2. Dual filter configuration.

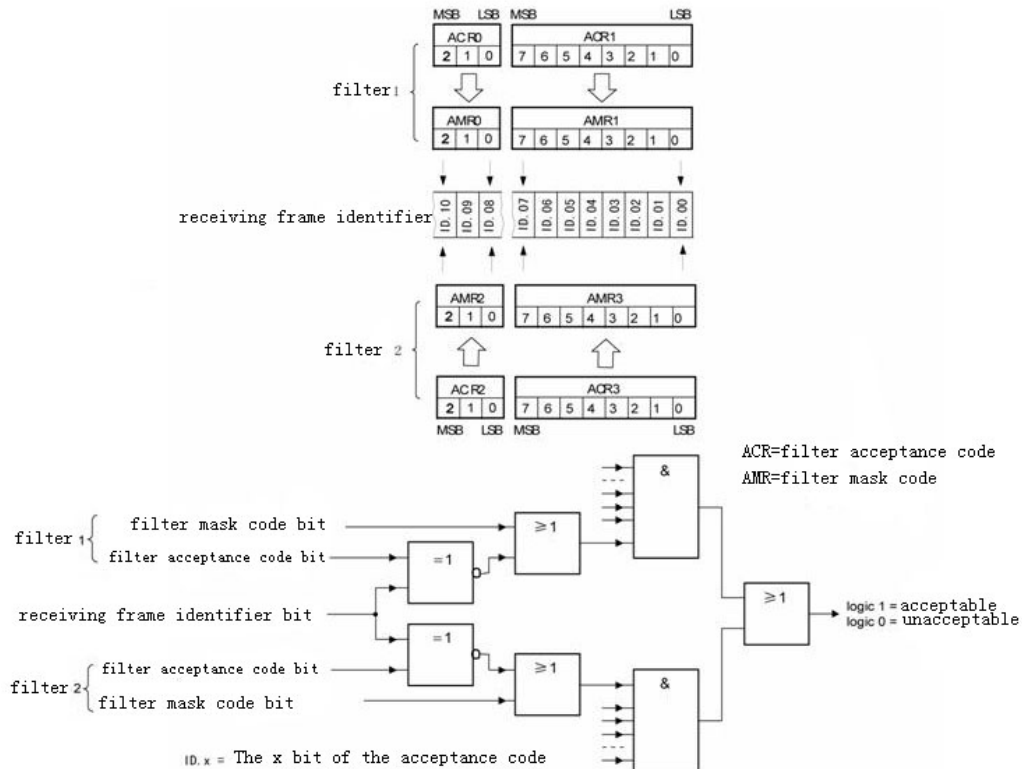
This configuration mode is able to define two short filters. A message to be received has to compare with two filters before it can be stored into the receiving buffer. The message received is valid only when at least one filter sent out an acceptance signal. The relationships for the corresponding bits between the filter byte and the message

byte are dependent on the current frame format.

**Standard frame:** For standard frame, it can be considered that the acceptance frame identifier is filtered with two single filters. See Figure 5-3.

To successfully receive message, all the bit comparisons should indicates “acceptable”.

This frame can be received only when at least one filter of two indicates “acceptable”.



**Figure 5-3 : Sketch for standard frame dual filter**

**Extended:** For extended frame, the two filters defined are the same. It only compares the first two bytes of the extended identifier (ID.28 to ID.13) for the two filters. See Figure 5-4.

To successfully receive message, all the bit comparisons should indicates “acceptable”.

This frame can be received only when at least one filter of two indicates “acceptable”.

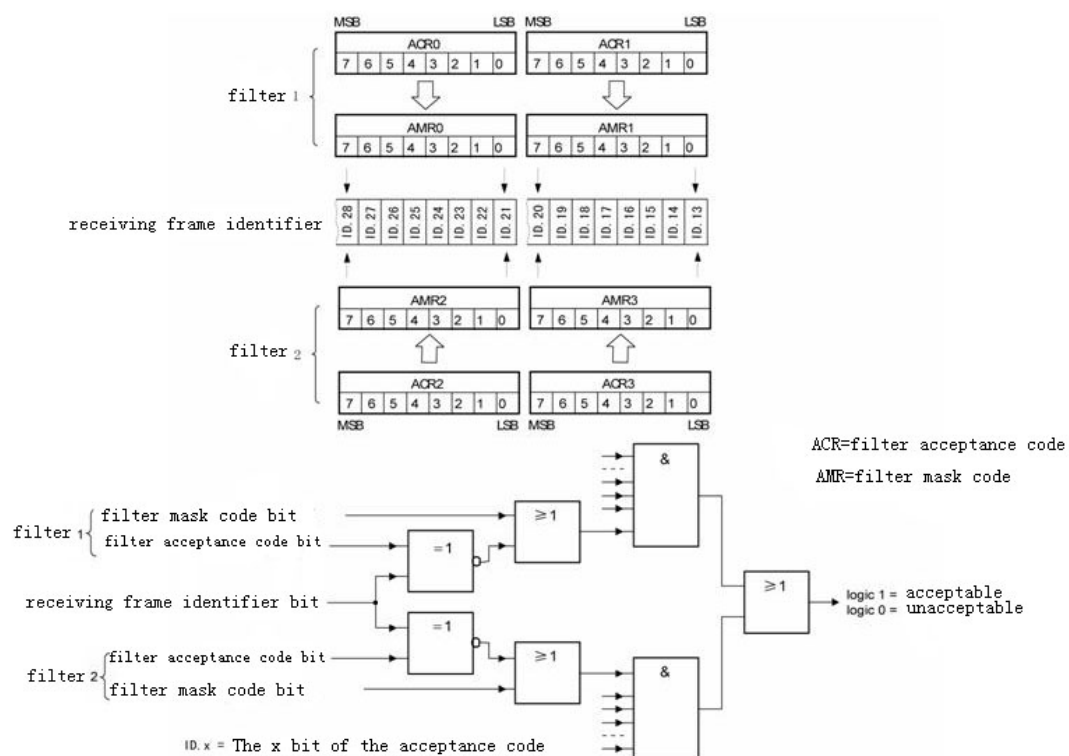


Figure 5-4 : Sketch for extended frame dual filter

Company name:	Guangzhou ZHIYUAN Electronics Co., Ltd. Industrial communication network department.
Address:	(R&D Department) Floor 2, No.7 Building, Huangzhou Industrial Estate Guangzhou, CHINA
Post code:	510660
Website:	30H <a href="http://www.embedcontrol.com">www.embedcontrol.com</a>
Tele:	+86 (020) 22644381 22644385 22644253
Fax:	+86 (020) 3860-4380
Website:	<a href="http://www.embedcontrol.com">www.embedcontrol.com</a>