USBCAN-II C

USB-CAN Converter

User Manual



Document Version: V4.45 (2025/06/27)

English Translation and Editorial Work by Shenzhen MoCAN Electric Co., Ltd.

Revision History

Version	Date	Reason
V1.00	2013/06/16	Document created
V2.01	2013/12/20	Corrected device operating parameters
V3.01	2014/10/22	Added some parameters
V4.10	2018/01/13	Document corrected
V4.20	2021/01/09	Document corrected
V4.30	2022/01/10	Document corrected
V4.40	2022/06/06	Added dual-channel merging and data file saving formats
V4.41	2022/07/06	Document corrected
V4.42	2023/01/05	Software update
V4.43	2024/03/05	Updated contact information
V4.44	2024/12/16	Dimension change
V4.45	2025/06/27	Dimension change

Contents

Revision History 2
Contents 3
1. Product Overview 5
1.1 Introduction 5
1.2 Features 5
1.3 Typical Applications 6
2. Device Installation 7
2.1 Device Dimensions 7
2.2 Interface Definition and Functions 7
2.3 Driver and Software Installation 8
2.3.1 Driver and Software Installation 8
2.3.2 Driver and Software Uninstallation 9
3. Device Usage 10
3.1 Connecting to a PC 10
3.1.1 USB Bus-Powered Mode 10
3.1.2 External Power Mode 10
3.2 Connecting to CAN-bus 10
3.3 CAN Bus Terminal Resistor 11
3.4 System Status LEDs 12
4. GCANTools Software 14
4.1 Starting the Software 14
4.2 Sending and Receiving Data 15
4.3 Bus Analysis Function 17
4.4 Other Functions 17
5. Secondary Development 19
6. Technical Specifications 20
7. Frequently Asked Questions 21
8. Disclaimer 24

1. Product Overview

1.1 Introduction

The USBCAN-II C is an industrial-grade CAN-bus communication interface card with 2 integrated CAN channels. This CAN card is compatible with the USB 2.0 full-speed specification. With the USBCAN-II C interface card, the PC can be quickly connected to a CAN-bus network through the USB port, forming a CAN-bus network control node for data processing and data acquisition in fields such as fieldbus laboratories, industrial control, intelligent residential communities and automotive electronic networks.

The USBCAN-II C interface card is a powerful tool for CAN-bus product development and CAN-bus data analysis. It is compact and plug-and-play, making it the best choice for users of portable systems.

The USBCAN-II C interface card is equipped with a CAN interface electrical isolation protection module, which protects the device from damage caused by momentary over-current/over-voltage and enhances system reliability in harsh environments.

The USBCAN-II C interface card supports 32-bit/64-bit operating systems such as Windows 2000, XP, 7, 8 and 10. Mocan Electric provides users with a unified application programming interface (API) and complete application demo code, including development examples in C#, LabVIEW, Python, Qt, VB.NET, VB and VC, to facilitate application development.

The USBCAN-II C interface card works with the general-purpose test software GCANTools, CAN Test and CAN Pro, and can perform functions such as CAN-bus message transmission/reception and monitoring.

1.2 Features

- PC interface complies with the USB 2.0 full-speed specification, compatible with USB 1.1 and USB 3.0
- 2 CAN-bus channels integrated, using Phoenix-terminal wiring
- Supports CAN 2.0A and CAN 2.0B frame formats, compliant with the ISO/DIS 11898 standard
- CAN-bus communication baud rate freely programmable between 5 Kbps and 1 Mbps
- Powered from the USB bus, or by an external power supply (+9–24 V DC, 130 mA) when the USB supply is insufficient
- CAN-bus interface is electrically isolated; isolation module withstand voltage: 3000 V DC
- Electrostatic discharge (ESD) immunity level: contact discharge ±4 kV, air discharge ±8 kV
- Electrical fast transient (EFT) immunity level: CAN-bus interface ±2 kV
- Surge immunity level: CAN-bus interface ±4 kV
- Maximum received data rate: 8000 fps
- Supports Windows 2000, XP, 7, 8, 10 and other Windows operating systems
- Supports the GCANTools, CAN Test and CAN Pro test software

- Operating temperature range: -40 °C ~ +85 °C

1.3 Typical Applications

- CAN-bus network diagnosis and testing
- Automotive electronics applications
- Power communication networks
- Industrial control equipment
- High-speed, large-volume communication

2. Device Installation

This chapter describes how to connect the USB-CAN interface card to a computer, and the points for attention when connecting the USB-CAN interface card to a computer for the first time.

2.1 Device Dimensions

Overall dimensions of the device: (length, including the terminal block) 105 mm × (width) 64 mm × (height) 23.5 mm. The schematic diagram is shown in Figure 2.1.

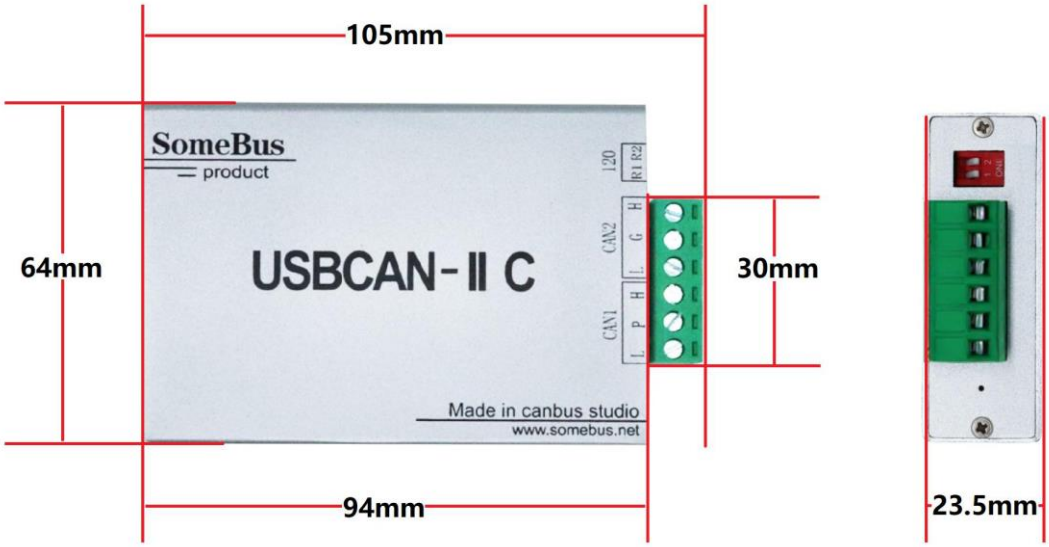

Figure 2.1 Dimensions of USBCAN-II C

2.2 Interface Definition and Functions

The USBCAN-II C interface card integrates 1 USB port, one +9–24 V DC auxiliary power port and 2 standard CAN-bus ports. The CAN-bus ports are brought out through one 6-pin pluggable terminal block, which can be used to connect 2 CAN-bus networks or CAN-bus devices.

The positions and definitions of the ports on the USBCAN-II C are shown in Figure 2.2, Figure 2.3 and Table 2.1, Table 2.2.

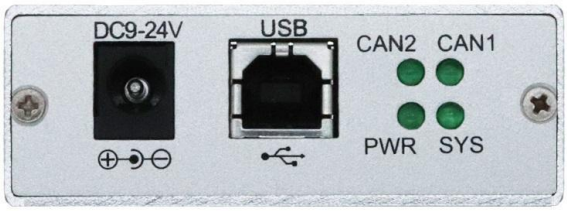

Figure 2.2 Positions of the USB and auxiliary power ports

Table 2.1 Definitions of the USB and auxiliary power ports

Port	Name	Function
9-24V DC	9-24V DC	Auxiliary power supply for the USBCAN; generally not required to be connected
USB	USB	Power supply for the USBCAN; connect to the computer



Figure 2.3 Position of the CAN-bus port

Table 2.2 CAN-bus signal assignment of the USBCAN-II C interface card

Port	Name	Function
CAN1	L	CAN1_L Signal line (CAN low)
	P	Shield
	H	CAN1_H Signal line (CAN high)
CAN2	L	CAN1_L Signal line (CAN low)
	G	CAN-GND Ground
	H	CAN1_H Signal line (CAN high)
120	R1	Terminal resistor of the CAN1 channel
	R2	Terminal resistor of the CAN2 channel

2.3 Driver and Software Installation

Before installing the driver and software, please make sure that the Windows account you are logged in with has super administrator rights, or that the user account has the permissions required to install drivers and software; otherwise the installation may fail.

To check Windows account permissions: Control Panel → User Accounts.

2.3.1 Driver and Software Installation

Users need to install the driver manually. Open the “Driver” folder on the CD and run the setup file that matches your system (32/64-bit) (DriverSetup.exe / DriverSetup64.exe) for manual installation.

Users can find and install the GCANTools, CAN Test and CAN Pro software directly on the CD. For details on software installation and usage, see the “Software Instructions” in the corresponding folder. Please read “Installation Must-Read” before installing.

2.3.2 Driver and Software Uninstallation

Users can uninstall the installed device driver by running DriverSetup.exe / DriverSetup64.exe as described above and clicking the “Uninstall” button.

Users can uninstall GCANTools, CAN Test and CAN Pro via “Add/Remove Programs” (XP) or “Programs and Features” (Win7).

Customers can also install the driver by installing the GCANTools software; the driver is installed automatically after the software installation is completed.

After the driver and software have been installed properly, plug the device into a USB port of the PC, and a new USBCAN device will be found in the PC Device Manager, with the driver name “GCAN USBCAN Device”. If there is no yellow exclamation mark or question mark, the device driver is working properly and the USBCAN device has been correctly connected to the PC.

3. Device Usage

3.1 Connecting to a PC

The USBCAN-II C interface card is plug-and-play. Therefore users can power the USBCAN-II C interface card directly from the USB port of the PC; if the USB power supply is insufficient, the external power supply mode must be used.

3.1.1 USB Bus-Powered Mode

USB bus-powered mode is suitable for most applications, for example, when the USBCAN-II C interface card is the only device connected to the USB port.

Connect the PC and the USBCAN-II C interface card directly with the supplied USB cable; the USB cable provides +5 V power to the USBCAN-II C interface card. At this point the PWR and SYS LEDs light up, indicating that the device is working properly and in the standby state.

3.1.2 External Power Mode

External power mode is suitable when the PC uses a USB hub (HUB), or when multiple USB devices have already been connected so that the USB port cannot provide enough current to the USBCAN-II C interface card.

Connect an external power supply (+9–24 V DC, 200 mA) to the +9–24 V DC power socket of the USBCAN-II C interface card; at this point the PWR and SYS LEDs light up. Then connect the PC and the USBCAN-II C interface card with the supplied USB cable, and the USBCAN-II C interface card works normally.

3.2 Connecting to CAN-bus

When connecting the USBCAN-II C to the CAN bus, simply connect CAN_H to CAN_H and CAN_L to CAN_L to establish communication.

The CAN-bus network uses a linear topology. The two farthest ends of the bus must be fitted with 120 Ω terminal resistors; if the number of nodes is greater than 2, the intermediate nodes do not need 120 Ω terminal resistors. For branch connections, the length should not exceed 3 m. The CAN-bus connection is shown in Figure 3.1.

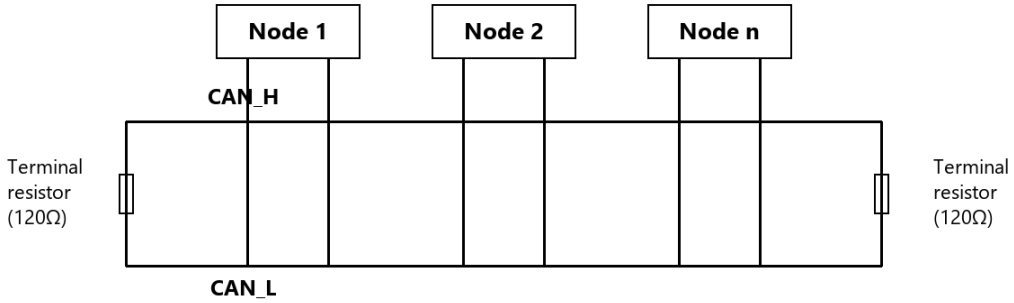

Figure 3.1 Topology of the CAN-bus network

NOTE: The CAN-bus cable can be ordinary twisted pair or shielded twisted pair. The theoretical maximum communication distance mainly depends on the bus baud rate. For the relationship between the maximum bus length and the baud rate, see Table 3.1. If the communication distance exceeds 1 km, make sure the wire cross-section is larger than 1.0 mm^2; the specific specification depends on the distance and is generally increased as the distance becomes longer.

Table 3.1 Baud rate versus maximum bus length

Baud rate	Bus length
1 Mbit/s	25 m
500 kbit/s	100 m
250 kbit/s	250 m
125 kbit/s	500 m
50 kbit/s	1.0 km
20 kbit/s	2.5 km
10 kbit/s	5.0 km
5 kbit/s	13 km

3.3 CAN Bus Terminal Resistor

To improve the reliability of CAN communication and eliminate signal reflection interference at the ends of the CAN bus, terminal matching resistors are usually added at the two farthest ends of the CAN-bus network, as shown in Figure 3.2. The value of the terminal matching resistor is determined by the characteristic impedance of the transmission cable. For example, if the characteristic impedance of the twisted pair is 120 Ω, the two ends of the bus should also be fitted with 120 Ω terminal resistors.

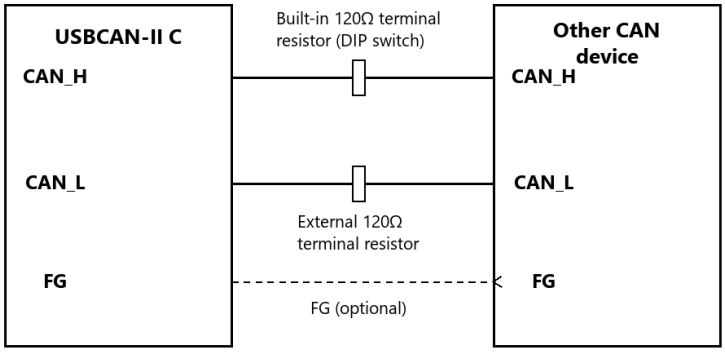

Figure 3.2 Connection of the USBCAN-II C with other CAN node devices

NOTE: A 120 Ω terminal resistor is already built into the USBCAN-II C, and whether the resistor is connected to the bus can be selected via DIP switches. The DIP switches are located next to the 6-pin terminal block. R1 and R2 are the terminal resistors of CAN1 and CAN2 respectively. Set them to ON to enable the resistors.

3.4 System Status LEDs

The USBCAN-II C interface card has 1 PWR LED, 1 SYS LED, 1 CAN1 LED and 1 CAN2 LED to indicate the operating status of the device. The specific indication functions of these 4 LEDs are shown in Table 3.2; when these 4 LEDs are in various states, the states of the CAN bus are shown in Table 3.3.

Table 3.2 LEDs of the USBCAN-II C interface card

LED	Color	Indication
PWR	Green	Power indicator
SYS	Green	System indicator
CAN1	Green	CAN1 signal indicator
CAN2	Green	CAN2 signal indicator

After the USBCAN-II C interface card is powered on, the PWR LED lights up and the system initialization status LED SYS lights up at the same time, indicating that the device has been powered and the system is initializing; otherwise, there is a system power fault or a serious system error has occurred.

Normal LED status: after power-on, the PWR LED lights up first, all four LEDs flash once together, and finally the PWR and SYS LEDs stay on steadily. After the software starts the device, the PWR LED stays on and the SYS LED flashes.

After the USB connection is normal, when data is being transmitted on the USB bus, the USB signal LED SYS flashes.

When data is transmitted or received on CAN1 or CAN2, the corresponding CAN1/CAN2 LED flashes.

Table 3.3 LED status of the USBCAN-II C interface card

LED	State	Indication
PWR	On	Power supply normal
	Off	Power supply fault
SYS	Steady on	Device initialized, standby state
	Off	Device initialization failed
	Flashing	Device in use
CAN1, CAN2	Off	No data transmission on the CAN channel
	Flashing	Data transmission on the corresponding CAN channel

4. GCANTools Software

GCANTools is dedicated debugging and analysis software developed by Mocan Electric for the Windows platform. With this software, users can send and receive CAN-bus data intuitively and quickly. The software is extremely easy to use and has very rich extension functions; users can get started quickly. This chapter introduces the main functions of GCANTools.

4.1 Starting the Software

1. If the GCANTools software has been installed, double-click the "GCANTools" icon on the desktop to open the software.

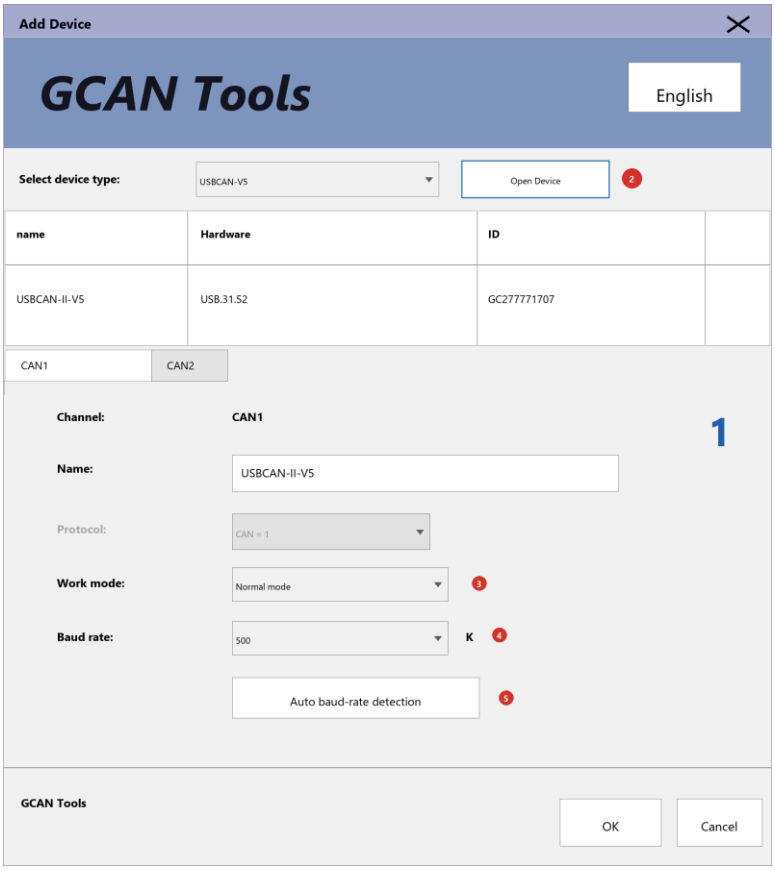

Figure 4.1 Add-device window of GCANTools

2. After selecting the corresponding device type, click "Open Device", and the CAN device plugged into the computer's USB port appears in the device list.

3. Select the working mode. The software provides 3 working modes: Normal mode, Listen-only mode and Self-send/self-receive mode. Normal mode is used for normal data transmission and reception with the software; Listen-only mode is used to monitor the bus and cannot send data — the USBCAN device does not act as a CAN-bus node and does not send acknowledgement or clock signals; Self-send/self-receive mode is used to test whether the CAN communication of the USBCAN device is normal.

4. Select the baud rate. Users need to set it according to the baud rate of the actually connected bus. If the baud rate does not match the bus, normal communication is impossible and the bus may even be disturbed. For a custom baud rate, please contact Mocan Electric. When the user does not know the bus baud rate, the automatic baud-rate detection function can also be used to scan.

5. Automatic baud-rate detection. If you do not know the baud rate of the target device, you can use the automatic baud-rate detection function, which provides two modes: “normal silent-mode detection” and “enhanced-mode detection”. The environment requirement of “normal silent-mode detection” is that there are at least two devices under test on the CAN bus that can communicate with each other normally, and there is active CAN data on the bus. The environment requirement of “enhanced-mode detection” is that the device under test is powered on and its CAN channel works properly, but it does not require the device to actively send CAN data, nor active CAN data on the bus. Before automatic detection, make sure the wiring is correct and two 120 Ω resistors are connected to the bus. You can use a multimeter to measure the resistance between CANL and CANH at both ends of the line without powering the devices in the CAN-bus system; it should be about 60 Ω.

Please note: for the CAN-bus system of a vehicle, please select “normal silent-mode detection” to identify the baud rate. Using “enhanced-mode detection” may cause error messages on the vehicle dashboard.

4.2 Sending and Receiving Data

The receive and transmit interface is the main function of GCANTools. In this interface, users can intuitively view the received CAN messages, and send data to the bus by normal sending or list sending.

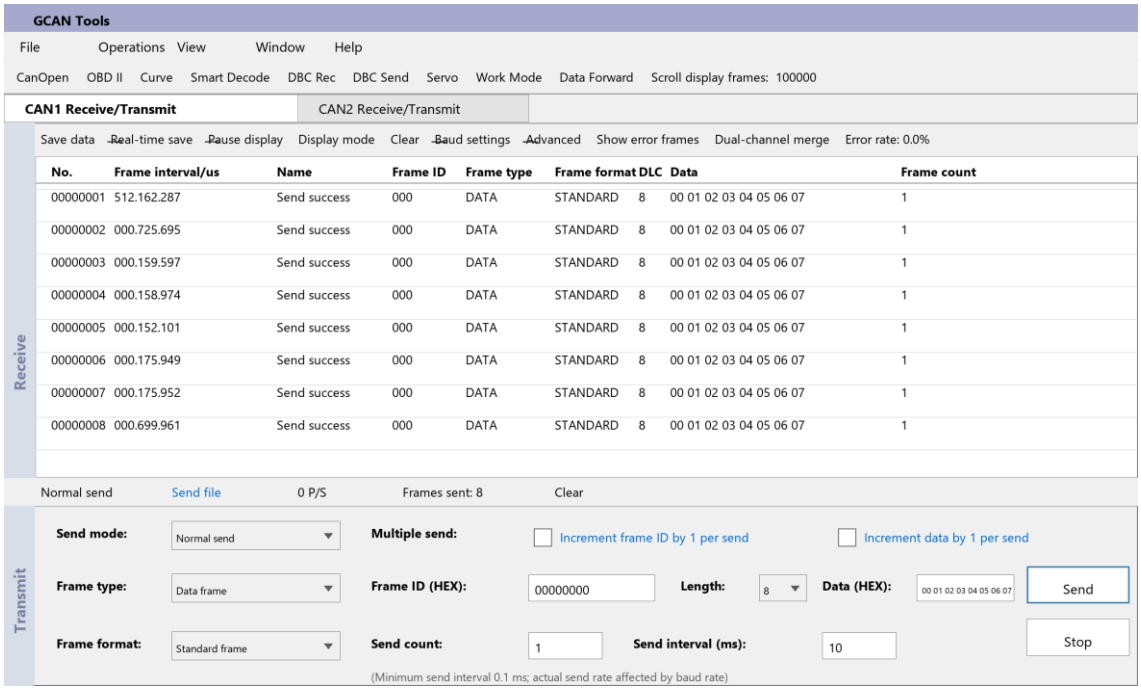

Figure 4.2 Main window of GCANTools

In addition to the basic functions above, the software also has a file sending function. File sending is divided into normal file sending and batch file sending. Normal file sending can send files to nodes on the bus; users can develop CAN-bus program flashing software based on this function. Batch file sending can re-send a batch file (.can) saved by the user back to the CAN bus with the timing and data as saved; see 4.4 for saving batch files.

Example of batch file sending:

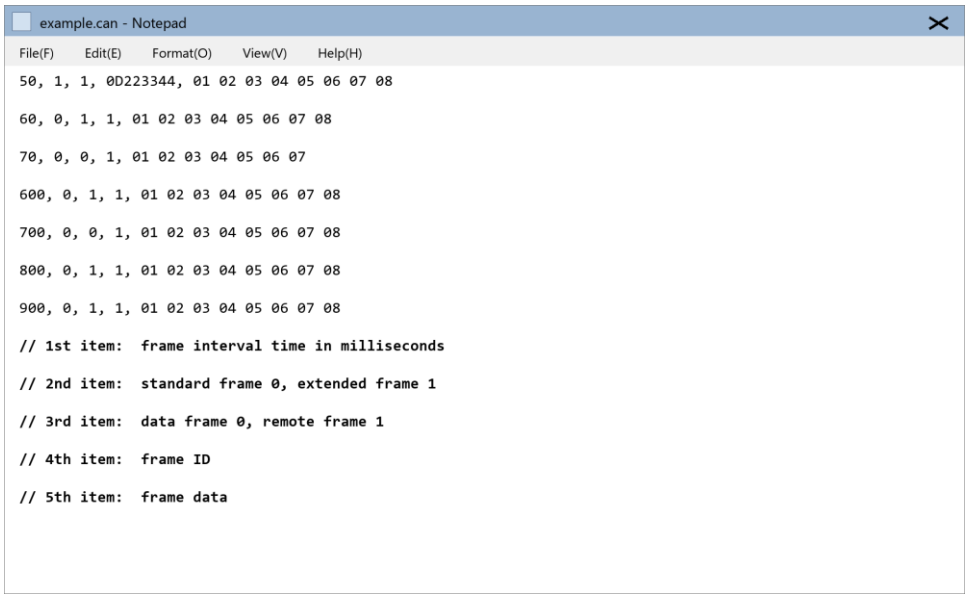

Figure 4.3 Example of a batch file (.can)

4.3 Bus Analysis Function

The bus diagnosis function can detect bus error frames and bus arbitration-lost bits, achieving fairly comprehensive CAN-bus analysis.

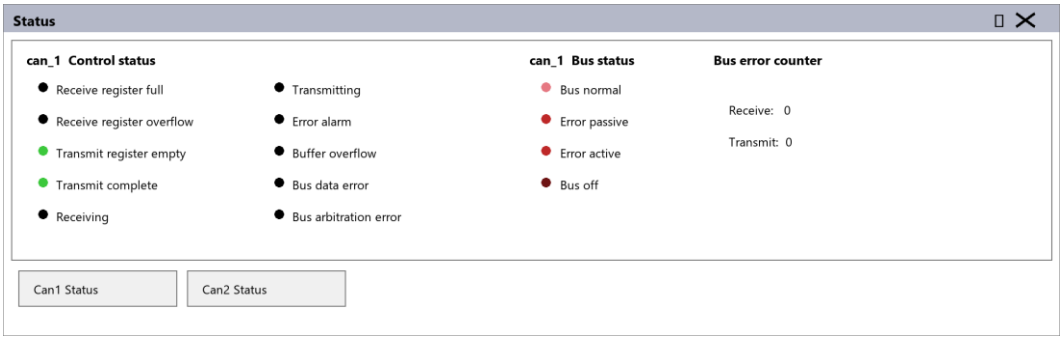

Figure 4.4 CAN-bus status window

CAN bus state display: indicates the current CAN bus state, including: bus normal, error passive, error active, bus off.

CAN controller internal FIFO overflow: bus messages within a certain period are too dense, so that the USBCAN cannot receive them in time and messages are lost.

CAN controller error warning: multiple transmit errors or receive errors occur on the bus, the error counter exceeds the alarm threshold, and the error count value can be displayed.

CAN controller error passive: multiple transmit errors or receive errors cause the CAN controller to enter the error-passive state, and the error count value can be displayed.

CAN controller arbitration lost: this node fails to win the bus when competing with other devices; the pending message will be retried at the next bus idle. The corresponding error bit information can be captured and displayed.

CAN controller bus error: a transmit error or receive error occurs on the bus, the error counter value is accumulated, and the error bit information such as ACK, CRC and bit-stuffing errors can be captured.

CAN data buffer overflow: because data has not been exchanged with the PC for a period of time, the number of messages stored in the device exceeds the hardware capacity limit, and newly received messages overwrite the earliest ones.

4.4 Other Functions

The GCANTools software has extremely rich extension functions, all of which are convenient and easy to use. Flexible use of these extension functions can help users analyze data conveniently and quickly.

Save data / real-time save: the receive list can be saved; saving formats: txt, can, excel, binary, ASC, BLF.

Pause display: pauses the scrolling of the receive list.

Display mode: scroll mode or list mode can be selected; in list mode data can be classified in real time by rules.

Clear: clears the data in the current receive list.

Filter setting: multi-segment filtering is implemented by editing filter IDs.

Advanced mask: masked IDs are not displayed by masking IDs.

Show error frames: shows/hides error frames on the bus.

Dual-channel merge: data received on the two channels can be merged and saved in the same file; when sending files, they are sent simultaneously according to the channels of the saved file.

NOTE: The USBCAN-II C does not support CANopen or OBD functions. For more functions of the GCANTools software, see the "GCANTools Software User Manual".

5. Secondary Development

Mocan Electric provides a standard interface function library for users performing secondary development, including ECANVCI.h, ECANVCI.lib and ECANVCI.dll. The interface function library is in standard format; users can declare and call these interface functions in programming environments such as VC and VB. For details, see the "ECAN Dynamic Library User Manual". Figure 5.1 shows the commonly used structure names and the function library call flow.

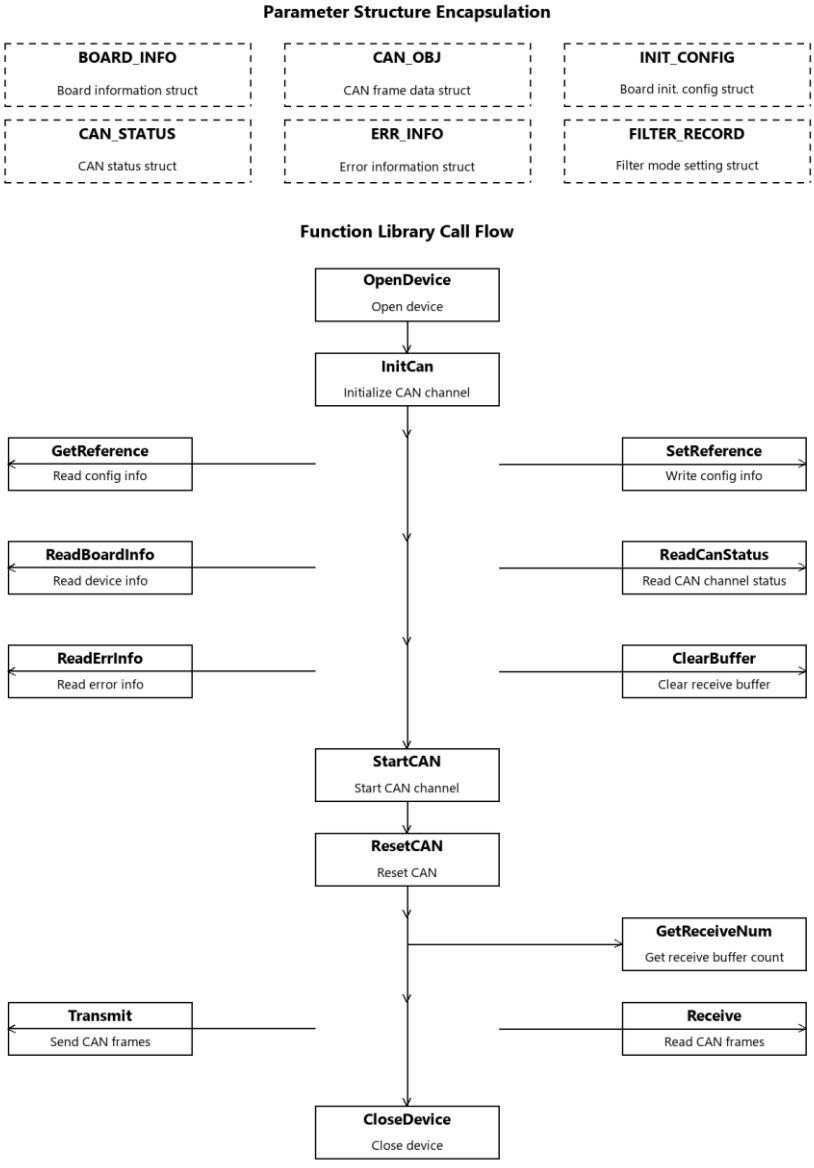

Figure 5.1 Secondary development function calls

6. Technical Specifications

Connection	
PC side	USB port, Type A
CAN side	OPEN6 Phoenix terminal
Interface features	
USB port	USB 2.0 full-speed interface, compatible with USB 1.1 and USB 3.0
CAN port	Compliant with ISO 11898 standard, supports CAN 2.0A/B
CAN baud rate	5 Kbit/s ~ 1 Mbit/s
Electrical isolation	3000 V DC
CAN terminal resistor	Built-in; enabled or disabled via DIP switch
Power supply	
Supply voltage	+5 V DC (USB port)
Supply current	Max. 130 mA
Environmental tests	
Operating temperature	-40 °C ~ +85 °C
Operating humidity	15% ~ 90% RH, non-condensing
EMC test	EN 55024:2011-09 EN 55022:2011-12
Protection level	IP 20
General	
Dimensions	105 mm × 64 mm × 23.5 mm
Weight	100 g

7. Frequently Asked Questions

1. When opening CAN Test or CAN Pro, the message "Failed to open device" is displayed.

Possible causes of this fault: controlcan.dll has not been replaced, or the USBCAN device driver is not properly installed. Check the properties of the USBCAN device in the PC Device Manager to see whether there is a "!" or "?" in front of the USBCAN device; if so, check for hardware/software conflicts and reinstall the USBCAN device driver.

2. Is the 120 Ω terminal matching resistor always required?

It is recommended to use the 120 Ω terminal matching resistor to absorb reflections at the endpoints and provide a stable physical link. When performing a single-node self-send/self-receive test, this 120 Ω terminal resistor must be connected to form a loop; otherwise the self-send/self-receive test cannot be performed. A 120 Ω terminal resistor is already connected inside the USBCAN-II C interface card.

3. Can multiple USBCAN interface cards be installed in one computer?

Old versions did not support simultaneous multi-card operation, but the current USBCAN interface cards support simultaneous operation of multiple PC-CAN interface cards of the same model.

4. What is the maximum data conversion rate of the USBCAN-II C interface card?

A single CAN channel of the USBCAN interface card supports CAN-bus data conversion at up to 8000 fps. The frames mentioned here refer to data frames with 8 data bytes in standard frames; frames with less than 8 data bytes or remote frames may be faster. In addition, the maximum data rate is limited by PC performance.

5. Why are the CAN status LEDs not lit?

All operations of the USBCAN interface card are controlled by the PC. The CAN status LEDs are meaningful only after the PC has sent the command to start CAN communication.

6. Why does an illegal operation occur when calling interface functions?

When using interface functions, first read the function descriptions carefully, make sure the input parameters are valid, and pay special attention to the passing of pointers (addresses), or refer to the provided example programs. If the problem is still not solved, contact our technical support.

7. How to set the communication baud rate of the PC-CAN interface card?

A set of commonly used baud rate setting values is provided. To use other baud rates, please calculate them yourself according to the "CAN-bus General Test Software and Interface Function Library User Manual". Note: the CAN controller of the USBCAN interface card uses a

24 MHz clock; when defining a baud rate yourself, it must be calculated based on this clock frequency.

8. Does the system standby or sleep state affect reception?

Yes, it does. In these states all processing stops, which will most likely cause a hardware receive buffer overflow error. If a program has opened the device, it will try to prevent the system from entering the standby or sleep state, so as to ensure normal operation of the system. When using the USBCAN interface card, please disable the standby and sleep functions of the system.

9. How to handle errors in the application?

Errors are mainly divided into two types: function call errors and CAN-bus communication errors. Function call errors are generally caused by parameter errors, such as device number out of range or wrong type number; the error number returned by the Win32 function GetLastError is 87. Some errors are caused by operating on a device that has not been opened, which is actually operating on an illegal handle. Corresponding Win32 standard error codes are provided depending on the specific function call; users can use GetLastError for error analysis, and this debugging work should generally be completed at the design stage.

CAN-bus communication errors are generally caused by the CAN network, and may also be caused by improper user settings, such as inconsistent baud rate settings, or calling the send function before starting the CAN controller. Most errors have been handled simply in the device driver; for deeper error analysis and handling, the ReadCANStatus function can be called.

In addition, note the data overflow interrupt error, which may occur in two cases: (1) software receive buffer overflow — this means the application cannot process the received data in time; the user should optimize the application or change the communication strategy. (2) hardware receive buffer overflow — this error is caused by excessive PC interrupt latency at the receiving end, and can only be solved by improving computer performance or coordinating the other nodes to reduce the sending speed appropriately.

10. What should be noted when opening and closing the device?

The USBCAN-II C interface card provides 2 CAN ports, and users can operate different ports in the same program. The USBCAN-II C interface card does not allow opening devices in shared mode; the same device cannot be opened multiple times by different processes calling OpenDevice. OpenDevice and CloseDevice generally only need to be called once during application initialization and exit. When closing the device, if the current port will no longer be used, call ResetCAN first to detach the current port from the CAN bus; the device driver automatically calls ResetCAN to exit the CAN bus connection only when the last device handle is closed.

11. How to use the Transmit send function better?

The driver of the USBCAN interface card provides a send buffer FIFO of about 128 frames; each Transmit call sends at most about 128 frames. The sending speed of the device is determined by the current software and hardware performance of the computer; generally the continuous sending speed is about 2000 fps (standard data frame, 11 bytes, 1 Mbps). If the sending speed is too fast, the remote receiving device may overflow and lose response; users can add appropriate delays in application programming to reduce the sending speed.

During sending, every frame has a timeout limit: for single-frame sending the timeout is about 2 s; when sending multiple frames at once, the timeout of the last frame is 2 s and that of the other frames is 1 s. Send timeouts generally occur when the CAN bus is busy and the current node has a low priority; this is not a function call or communication error, and users can implement retransmission by programming (send timeout events rarely occur on low/medium-speed networks). Therefore, in system design, make sure the CAN bus utilization does not exceed 60–70% of the bus capacity.

12. How to use the Receive function better?

The device driver provides a software receive buffer of 100,000 frames, which provides application programmers with sufficient reaction/processing time. When the software receive buffer overflows, the device driver calls ResetCAN to reset the CAN bus and sets the data-overflow interrupt flag bit in CAN status; note that both software buffer overflow and CAN controller hardware buffer overflow use this flag bit.

When Wait is 0, the function returns immediately at call time with the number of frames successfully read; if the receive buffer is empty, it returns 0. When Wait is non-zero, if there is data in the receive buffer at call time the function returns immediately with the number of frames successfully read; if the receive buffer is empty at that time, the function waits until a specified timeout expires or data is received, then returns the number of frames successfully received. When Wait is 0xFFFFFFFF, it waits indefinitely until data is received; it is recommended not to set Wait too large, and be especially careful with indefinite waiting.

When nFrames equals 0, the function is actually a notification message return and does not read the receive buffer; this is a special, skillful usage. Note: if the Receive function is called in the main thread and Wait is non-zero, the application may temporarily stop responding. For polled reception, Wait should generally be set to 0.

8. Disclaimer

Thank you for purchasing GCAN series hardware and software products.

1. Copyright and Translation Notice

The original Chinese-language version of this document was prepared by Shenyang Guangcheng Technology Co., Ltd., which retains the copyright to the original document and its contents.

This English version has been independently translated and prepared by Shenzhen MoCAN Electric Co., Ltd. based on its understanding of the original Chinese documentation. It is intended solely for overseas customer reference and technical support purposes. It has not been officially reviewed, verified or approved by Shenyang Guangcheng Technology Co., Ltd., and should not be regarded as an official or authoritative English version of the original documentation.

The English translation, editorial work and original layout elements created by Shenzhen MoCAN Electric Co., Ltd. are protected by applicable copyright laws. No part of this English version may be reproduced, redistributed, republished or used for commercial purposes without prior written permission from Shenzhen MoCAN Electric Co., Ltd., except as permitted by applicable law.

GCAN is a registered trademark of Shenyang Guangcheng Technology Co., Ltd.

2. Product Information and User Responsibility

Product specifications, technical parameters, functions and other information provided in this document are for reference purposes and may be subject to change without prior notice.

Users are responsible for evaluating the suitability of the products for their specific applications and ensuring proper installation, configuration and operation in accordance with the relevant technical documentation and applicable safety requirements.

For product selection, technical clarification or application-specific requirements, users are advised to consult the manufacturer or an authorized supplier before making purchasing or implementation decisions.

3. Interpretation

This English document is provided as a reference translation of the original Chinese-language documentation. While reasonable efforts have been made to ensure the accuracy and consistency of the translation, differences in technical terminology, interpretation or expression may occur.

In the event of any discrepancy, ambiguity or inconsistency between this English version and the original Chinese-language documentation, the original Chinese-language version shall prevail.

For technical questions or clarification regarding the product specifications and intended functions, users are encouraged to contact Shenyang Guangcheng Technology Co., Ltd. or their authorized supplier.