USBCAN-II FD

Industrial USB-CAN Converter

User Manual



Document Version 1.42 (2022/11/14)

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

Revision History

Version	Date	Reason
V1.00	2020/01/10	Document created
V1.10	2020/04/30	Released
V1.20	2022/05/30	Some descriptions corrected
V1.30	2022/06/06	Dual-channel merge and save-data file format added
V1.40	2022/07/05	Specification corrections
V1.41	2022/07/06	Document corrected
V1.42	2022/11/14	Baud rate corrections

Contents

Revision History 2
Contents 3
1. Function Introduction 5
1.1 Overview 5
1.2 Features 5
1.3 Typical Applications 5
2. Device Installation 6
2.1 Device Dimensions 6
2.2 Interface Definition and Functions 6
2.3 Driver and Software Installation 7
2.3.1 Driver and Software Installation 7
2.3.2 Driver and Software Uninstallation 7
3. Device Usage 7
3.1 Connection to PC 7
3.1.1 USB Bus Power Mode 8
3.1.2 External Power Supply Mode 8
3.2 Connection to CAN(FD)-bus 8
3.3 CAN-bus Terminating Resistors 8
3.4 System Status Indicators 9
4. Software Usage 10
4.1 Software Startup 10
4.2 Data Reception Functions 11
4.2.1 Dual-channel Merged Reception 12
4.2.2 Save Data and Realtime Save Functions 12
4.2.3 Pause Display Function 14
4.2.4 Display Mode 14
4.2.5 Clear Function 14
4.2.6 Filter Settings 14
4.3 Data Transmission Functions 15
4.3.1 Sending Data in Normal Mode 15
4.3.2 Sending Data in List Mode 16
5. Secondary Development 17
6. Technical Specifications 18

7. Frequently Asked Questions 19
8. Disclaimer 22
Copyright and Translation Notice 22
Product Information and User Responsibility 22
Interpretation 22
Appendix: CAN 2.0 Protocol Frame Format 24
A.1 CAN 2.0A Standard Frame 24
A.2 CAN 2.0B Extended Frame 24

1. Function Introduction

1.1 Overview

The USBCAN-II FD is a high-performance bus communication interface card integrating two CAN FD interfaces. Compliant with the high-speed USB 2.0 bus specification, the card provides two independent CAN/CAN FD channels whose signals are isolated from each other. Through the USB interface, a PC can quickly connect to a CAN(FD)-bus network, forming a CAN(FD)-bus network control node for data processing and data acquisition in field-bus laboratories, industrial control, smart communities, automotive electronics networks and other CAN(FD)-bus applications.

The USBCAN-II FD high-performance CAN interface card is a powerful tool for CAN(FD)-bus product development and CAN(FD)-bus data analysis. Compact and plug-and-play, it is also the best choice for portable-system users. The card carries an on-board USB port and integrates an electrical-isolation protection module for the CAN interface, protecting the device from damage caused by transient over-current/over-voltage and enhancing system reliability in harsh environments.

The USBCAN-II FD supports Windows XP, 7, 8 and 10 (32-bit/64-bit) operating systems. A unified application programming interface and complete sample application code are provided, with development examples to facilitate application development.

The USBCAN-II FD can be used with our independently developed ECANTools general-purpose test software to send, receive and monitor CAN(FD)-bus messages.

1.2 Features

- PC interface complies with the USB 2.0 full-speed specification; compatible with USB 1.1 and USB 3.0
- Integrates two CAN FD interfaces using pluggable terminal-block wiring
- Supports CAN 2.0A and CAN 2.0B frame formats; compliant with ISO/DIS 11898
- CAN communication baud rate freely programmable from 10 kbps to 1 Mbps
- CAN FD communication baud rate up to 5 Mbit/s
- Powered from the USB bus, or by an external power supply (5 V DC, 130 mA)
- CAN(FD)-bus interface electrically isolated; isolation withstand voltage 2500 V DC
- Maximum receive data throughput: 14000 fps
- CAN-side received-message timestamp accuracy up to 1 us
- Supports Windows XP, 7, 8 and 10 operating systems
- Supports the ECANTools test software
- Supports secondary development
- Operating temperature range: -40°C to +85°C

1.3 Typical Applications

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

2. Device Installation

2.1 Device Dimensions

The device outline dimensions are (length, including terminal block) 104 mm x (width) 64 mm x (height) 24 mm, as shown in Figure 2.1.



Figure 2.1 Outline dimensions of USBCAN-II FD

2.2 Interface Definition and Functions

The USBCAN-II FD interface card integrates one USB port, one DC 9-24 V auxiliary power port and two standard CAN(FD)-bus ports. The CAN(FD)-bus ports are brought out through one 6-pin pluggable terminal block, which can be used to connect two CAN(FD)-bus networks or CAN(FD)-bus devices. The positions and definitions of each interface are shown in Figure 2.2, Figure 2.3, Table 2.1 and Table 2.2.



Figure 2.2 Position of the USB and auxiliary power ports

Port	Name	Function

DC 9-24 V	DC 9-24 V	Auxiliary power for the USBCAN; normally not required
USB	USB	Powers the USBCAN and connects to the computer

Table 2.1 USB and auxiliary power port definitions

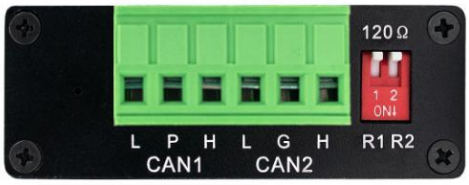


Figure 2.3 Position of the CAN(FD)-bus port

Pin (left to right)	Port name	Function
1	CAN1 L	CAN1_L signal line (CAN low)
2	P	Shield
3	H	CAN1_H signal line (CAN high)
4	CAN2 L	CAN2_L signal line (CAN low)
5	G	CAN-GND (ground)
6	H	CAN2_H signal line (CAN high)

Table 2.2 CAN(FD)-bus signal assignment of the USBCAN-II FD interface card

2.3 Driver and Software Installation

Before installing the driver and software, please make sure that the Windows account you are logged into is an administrator account, or that the user account has the relevant permissions to install drivers and software; otherwise the installation may fail. The method for checking Windows account permissions is: 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 install it manually. After the driver and software are installed correctly, plug the device into the PC's USB port; a new USBCAN device will appear in the PC's Device Manager under the driver name "GC-Tech USBCANFD-I Device". If there is no yellow exclamation mark or question mark, the device driver is working normally and the USBCAN device is correctly connected to the PC.

2.3.2 Driver and Software Uninstallation

Users can find the device driver in the PC's Device Manager and delete it. The ECANTools software can be uninstalled from "Add/Remove Programs" in the PC's Control Panel.

3. Device Usage

3.1 Connection to PC

The USB port of the USBCAN-II FD complies with the USB 2.0 full-speed protocol and can communicate with PCs supporting USB 1.1, USB 2.0 and USB 3.0 standards.

3.1.1 USB Bus Power Mode

USB bus power mode is suitable for most applications, for example when the USBCAN-II FD is the only device connected to the USB port. Connect the PC and the USBCAN-II FD directly with the supplied USB cable; the USB cable provides +5 V power to the card. The PWR and SYS indicators light up, indicating that the device is working normally and is waiting for connection.

3.1.2 External Power Supply Mode

External power supply mode is suitable when the PC uses a USB hub, or when several USB devices are already connected so that the USB port cannot supply enough current to the USBCAN-II FD. Connect an external power supply (9-24 V DC, 200 mA) to the DC 9-24 V power socket of the USBCAN-II FD; the PWR and SYS indicators light up. Then connect the PC and the USBCAN-II FD with the supplied USB cable, and the card works normally.

3.2 Connection to CAN(FD)-bus

To connect the USBCAN-II FD to a CAN bus, simply connect CAN_H to CAN_H and CAN_L to CAN_L to establish communication. The CAN(FD)-bus network uses a linear topology; the two most distant ends of the bus need 120-ohm terminating resistors. If there are more than two nodes, the intermediate nodes do not need 120-ohm terminating resistors. For branch connections, the length should not exceed 3 meters. The CAN(FD)-bus connection is shown in Figure 3.1.

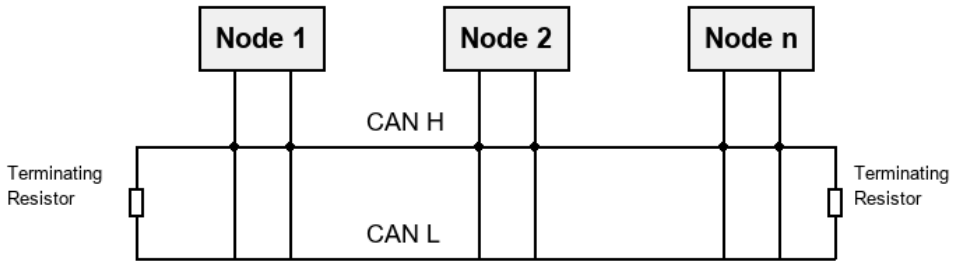


Figure 3.1 Topology of a CAN(FD)-bus network

3.3 CAN-bus Terminating Resistors

To enhance the reliability of CAN communication and eliminate signal-reflection interference at the bus ends, terminating matching resistors are usually added at the two most distant ends of the CAN-bus network, as shown in Figure 3.2. The value of the terminating resistor is determined by the characteristic impedance of the transmission cable. For example, if the characteristic impedance of the twisted pair is 120 Ω, then a 120-ohm terminating resistor should be integrated at both ends of the bus.

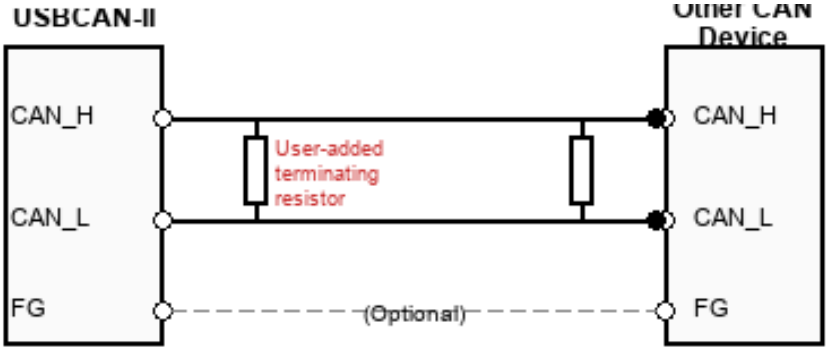


Figure 3.2 USBCAN-II FD connected to other CAN node devices

Note: The USBCAN-II FD has a 120-ohm terminating resistor integrated internally; a DIP switch selects whether the resistor is connected to the bus. The DIP switch is next to the 6-pin terminal block; R1 and R2 correspond to the terminating resistors of CAN2 and CAN1 respectively. Switch to the ON position to enable the resistor.

3.4 System Status Indicators

The USBCAN-II FD has one PWR indicator, one SYS indicator, one CAN1 indicator and one CAN2 indicator to show the operating status of the device. The functions of these four indicators are listed in Table 3.2, and the CAN-bus status corresponding to the various indicator states is shown in Table 3.3.

Indicator	Color	Indication
PWR	Green	Power indication
SYS	Green	System indication
CAN1	Green	CAN1 signal indication
CAN2	Green	CAN2 signal indication

Table 3.2 USBCAN-II FD interface card indicators

After the USBCAN-II FD is powered on, all four indicators light up simultaneously; then PWR stays on, SYS flashes slowly while CAN1 and CAN2 go off, indicating that the device is powered and the system has completed initialization. Otherwise, there is a system power fault or another fault, and you should contact our customer service.

After the USB interface is connected correctly, when PC software accesses the USBCAN device the USB signal indicator SYS will flash. At this time, when CAN1 or CAN2 sends or receives data, the corresponding CAN1 or CAN2 indicator will flash. If SYS flashes but CAN1 or CAN2 does not light up, there is no data on the CAN channel; check the wiring, communication baud rate and matching resistors.

Indicator	State	Indication
PWR	On	Power supply normal
	Off	Power supply fault
SYS	Off	Device initialization failed
	Flashing	Device initialization passed
CAN1, CAN2	Off	No data transmission on CAN channel

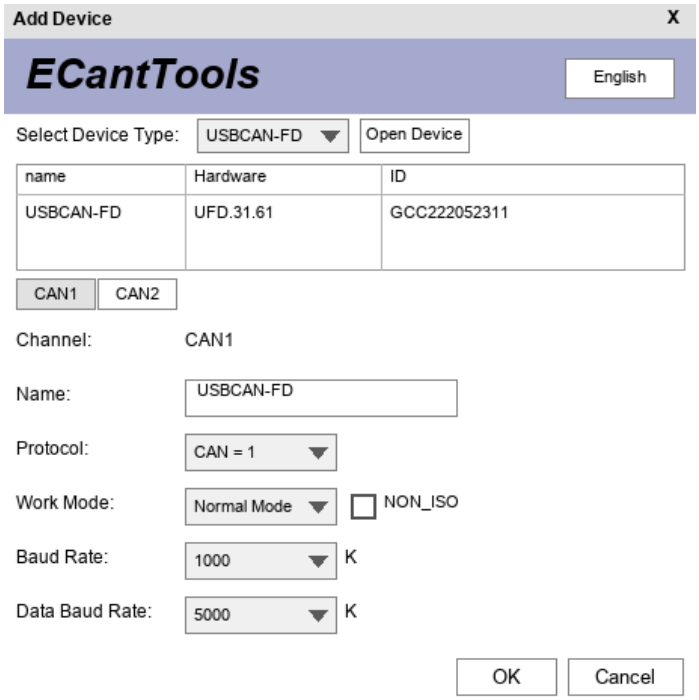
	Flashing	Data transmission on the corresponding CAN channel

Table 3.3 USBCAN-II FD indicator states

4. Software Usage

4.1 Software Startup

After installing the Windows driver, connect the CAN analyzer hardware and open the software; the setting interface shown below appears.



1) "Select Device Type": you need to select the version matching the hardware - USBCAB-FD.

2) "Open Device" button: click this button to access the USBCAN device. If "USB device open error!" is displayed, check that the device type selected in step 1 is correct and that the driver in Device Manager is installed correctly.

3) Device display window: this window shows the hardware number and serial number (SN) of the device.

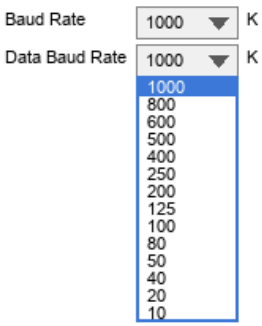
4) Channel selection area: here you can switch the setting window of the CAN channels.

5) Working mode: here you can select normal mode, listen-only mode, or self-transmit/self-receive mode. For normal use, select the default normal mode.

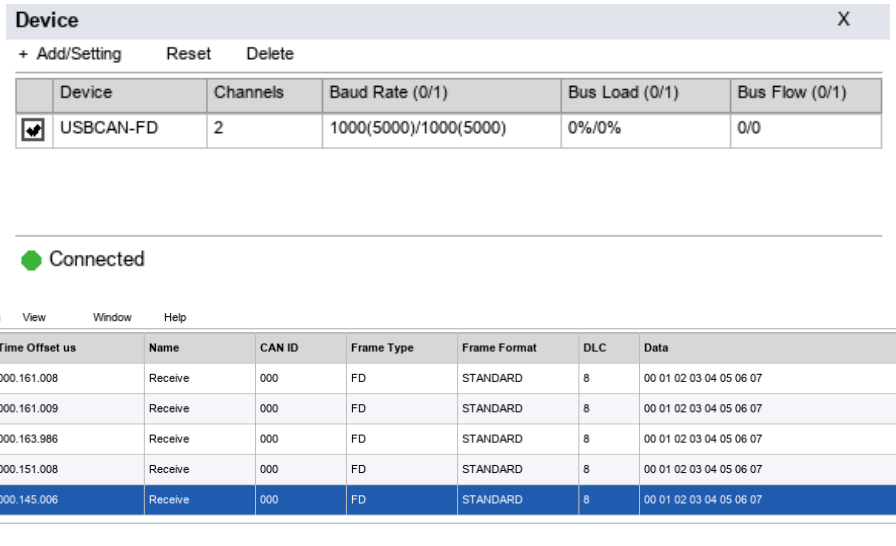
6) Baud rate selection: here you can select the baud rate. The baud rate is critical for CAN-bus communication; before communicating you need to determine the baud rate of the target device or target bus.

After selecting the device, you can set the working mode and baud rate below. If you have a dual-channel device, you can set CAN1 and CAN2 separately using the tabs. Setting the baud rate is particularly important; here the baud rate cannot be entered manually but must

be selected from a drop-down menu. Our products support the standard baud rates shown below.

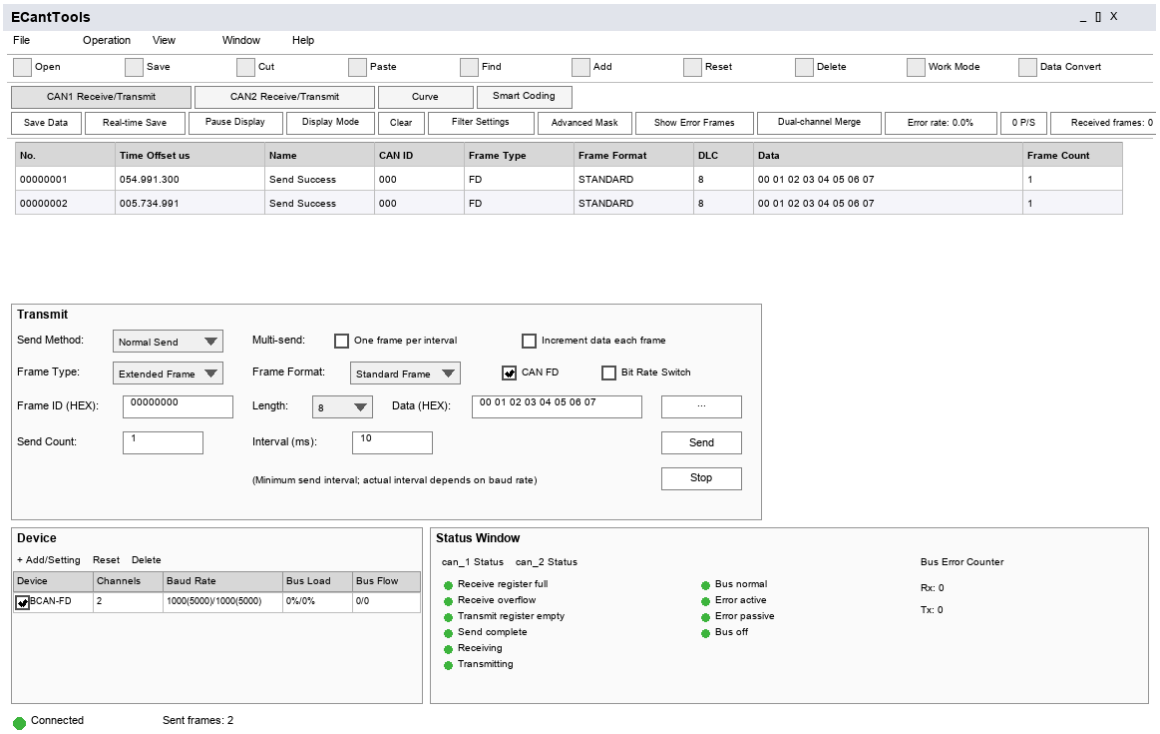


7) OK button: after selecting the baud rate, click OK to enter the software. If the hardware starts correctly, the SYS indicator changes from steady on to flashing (flashing twice continuously). After a successful connection the interface shown below appears.



4.2 Data Reception Functions

After the device parameters are set, the software enters the working state. If there is data on the bus, the receive-data window will display the data. The receive window is shown below.



In the red box, the CAN FD option: when checked, CAN FD is enabled and the frame type becomes FD; when unchecked, CAN FD data cannot be received and only CAN 2.0 data can be received. The Bit Rate Switch option, when checked, enables the variable bit rate.

4.2.1 Dual-channel Merged Reception

For dual-channel devices, when CAN1 and CAN2 receive data simultaneously, click "Dual-channel Merge" to receive data from both channels in the current interface, distinguished by name. When saving data, the merged dual-channel data is saved in the same file; when sending files, both channels send simultaneously according to the number of channels in the saved file.



If "Dual-channel Merge" is not clicked, each interface normally receives its own channel's data.



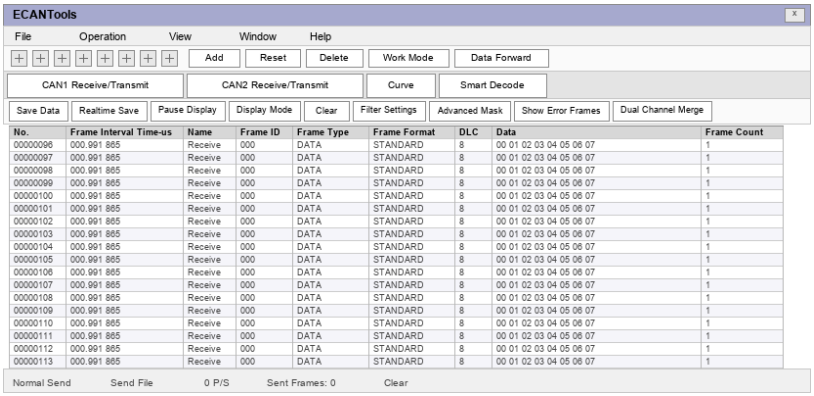
4.2.2 Save Data and Realtime Save Functions

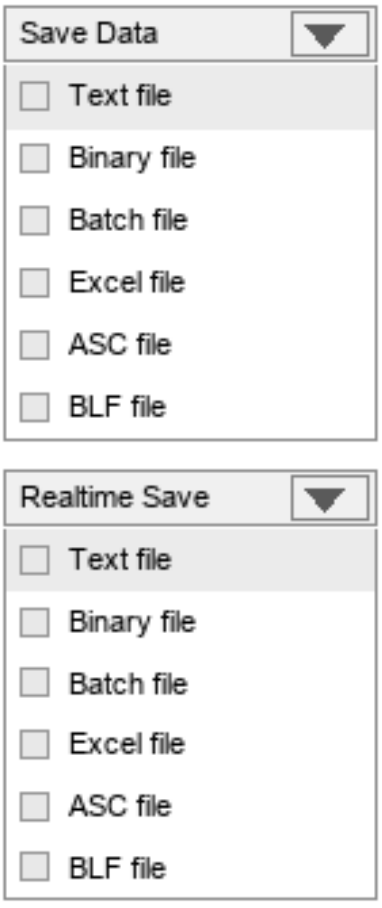
Users can save all data in the current send/receive list to local storage; the save formats are listed below.

File Type	File Format	Editor
Text file	.txt	Notepad

Binary file	.dat	Notepad
Batch file	.can	Notepad
Excel file	.csv	WPS Excel
ASC file	.ASC	CANoe
BLF file	.blf	CANoe

Text files are convenient for data storage and later analysis. Batch files can be modified with Notepad and sent back to the bus. For a description of the batch-file format, see section 4.3.3.



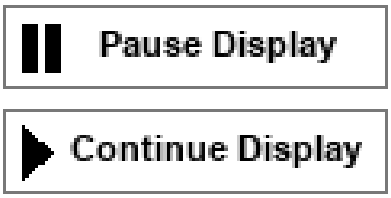


Click "Realtime Save" on the toolbar, set the text type and file name for realtime saving, and the realtime save function starts (i.e., save point A is set). Clicking again (i.e., setting save point B) stops saving and writes all data from the start (A) to the end (B) into the save file.

Please note that the system does not write data in real time. During realtime saving, the save file contains no data.

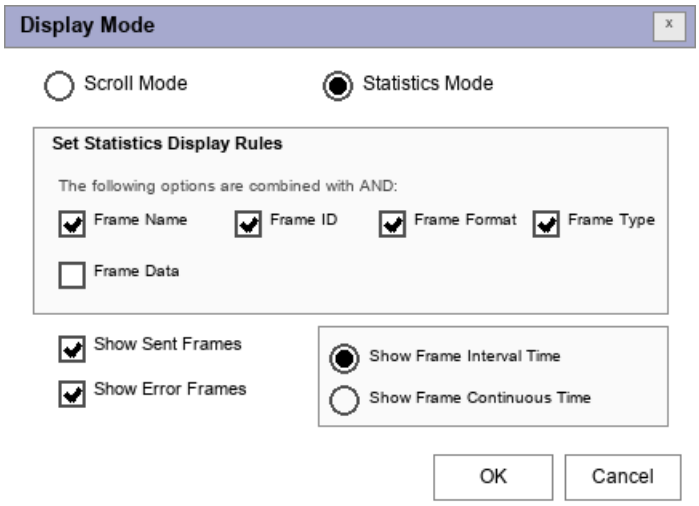
4.2.3 Pause Display Function

You can click Pause to freeze the scrolling data window. While paused, the device and software still receive data normally; only the data window does not refresh. Click Continue Display to resume scrolling.



4.2.4 Display Mode

Display modes include scrolling mode and statistics mode. Scrolling display continuously scrolls received data downwards in the receive list, so the current window shows the latest data. The statistics list displays data grouped according to set rules; for example, packets with the same ID can be displayed together with a count of the number of packets. Statistics mode makes it convenient for engineers to capture newly generated or changed data on the bus.



No.	Frame Interval Time-us	Name	Frame ID	Frame Type	Frame Format	DLC	Data	Frame Count
00000000	000.010 997	Receive	000	FD	STANDARD	8	00 01 02 03 04 05 06 07	3519

4.2.5 Clear Function

You can clear the data in the receive/send windows and the data in the buffer.

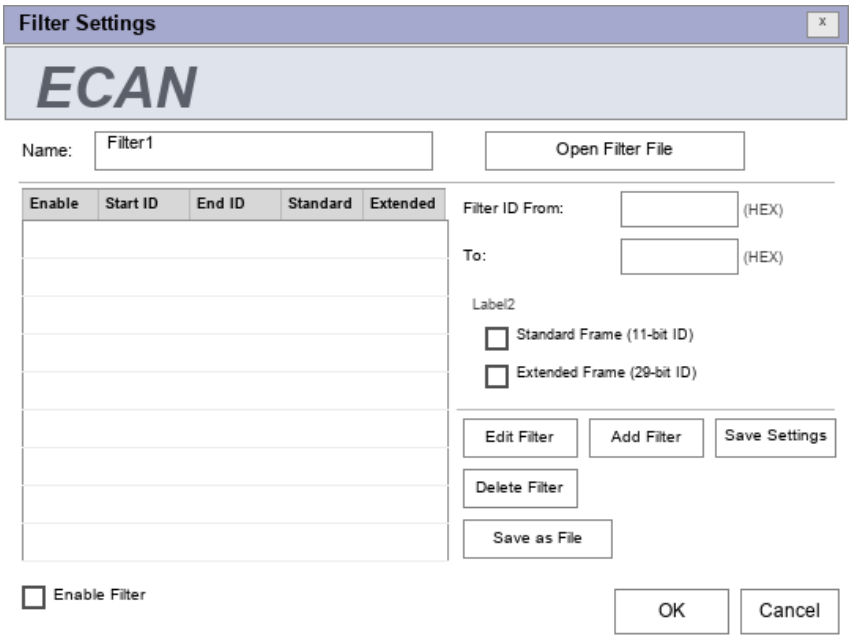


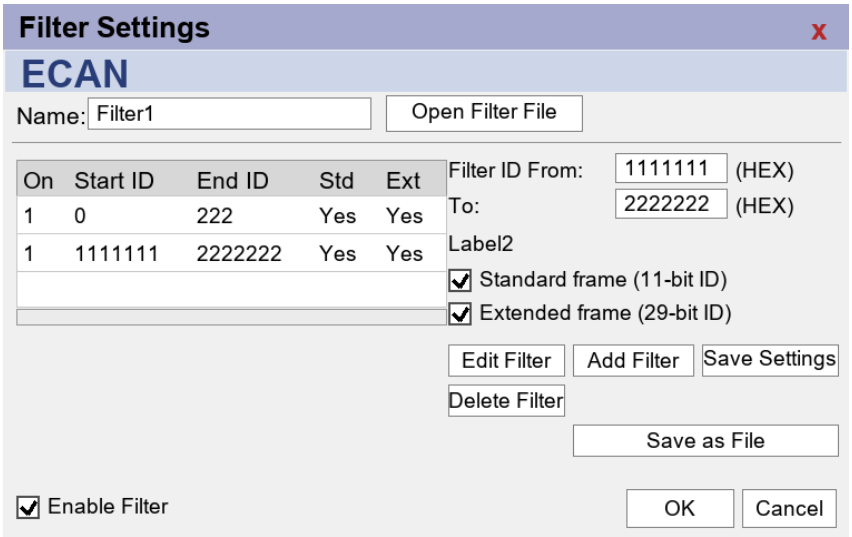
4.2.6 Filter Settings

Receive filter settings can set a filter ID or ID range. If a filter is set, the software displays only the set filter ID (range); IDs outside the filter range are filtered out. Click "Edit Filter" to open the filter setting window, where you can set the filter range.



Click "Add Filter" to start editing. Enter the ID or ID range you want to filter and click "Save Settings" to add the filter content to the left side. Several filter entries can be added to the left list; check the corresponding filter segment to enable it. The software can enable multiple filters at the same time. Please note that you need to check "Enable Filter" for the filters to take effect. Filter files can be saved and loaded through "Open Filter File".



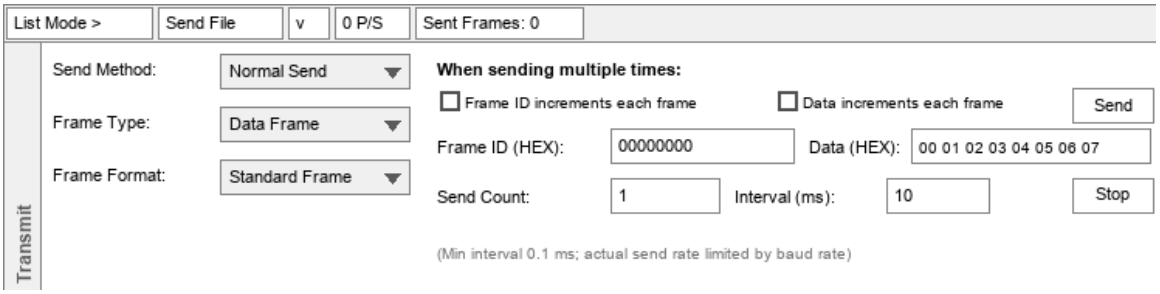


4.3 Data Transmission Functions

4.3.1 Sending Data in Normal Mode

Normal mode allows you to edit the frame data to be sent very intuitively and to set special functions such as cyclic sending. Normal mode is simple and frame editing is very intuitive.

Note that when entering data, a space must be entered between each byte; otherwise the "Data format error!" message appears.

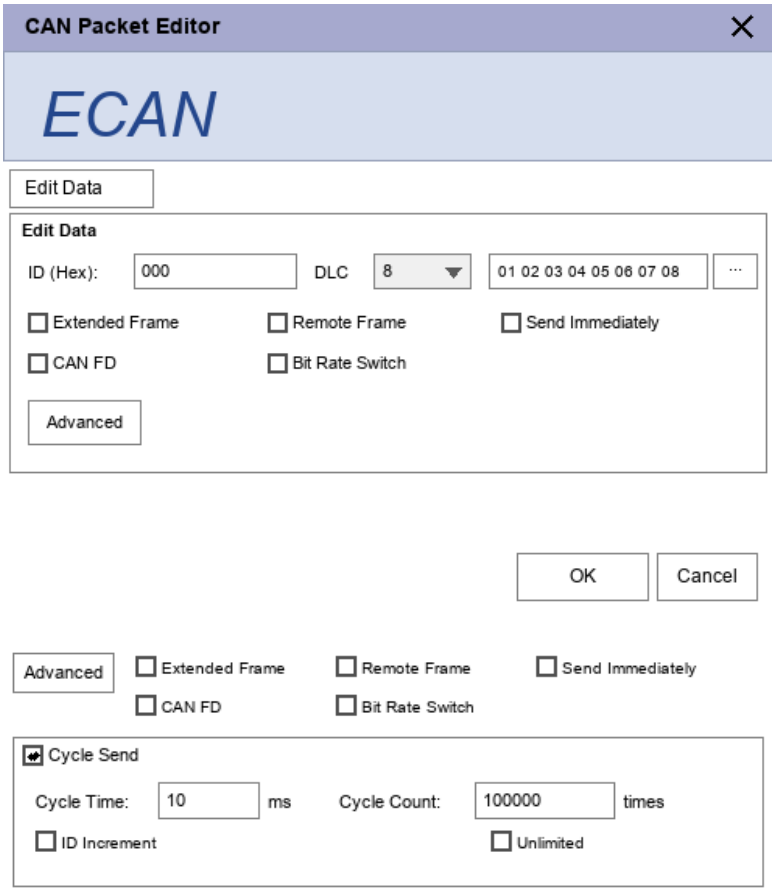


4.3.2 Sending Data in List Mode

After clicking List Mode, the original position switches to Normal Mode. Click New Frame to edit the data frame you want to send.



In the edit-data interface you can set the frame ID, frame type, frame format, frame length and so on; in Advanced Options you can set the cyclic sending method, cyclic sending interval, number of cyclic sends, ID increment, and other options.



Once set, the CAN message can be added to the send queue: click the checkbox to the left of "send" to start or stop sending. When sending finishes, the checkbox resets automatically, and the number of successfully sent packets is shown in "Frames Sent".

Send	No.	Name	Frame ID	Frame Type	Frame Format	DLC	Data	Total Frames	Sent Frames	Interval Time
☐	00000000		000	FD	STANDARD	8	01 02 03 04 05 06 07 08	-1	2333	10
☐	00000001		000	FD	STANDARD	8	01 02 03 04 05 06 07 08	-1	0	10

5. Secondary Development

We provide a standard interface function library for secondary-development users, including ECanFDVci.h and ECANFDVCI.dll. These libraries are all in standard format; users can declare and call these interface functions in programming environments such as VC, VB and LabVIEW. For specific usage, see the "ECANFDVCI Dynamic Library User Manual". Figure 5.1 shows the common structure names and the function-library call flow.

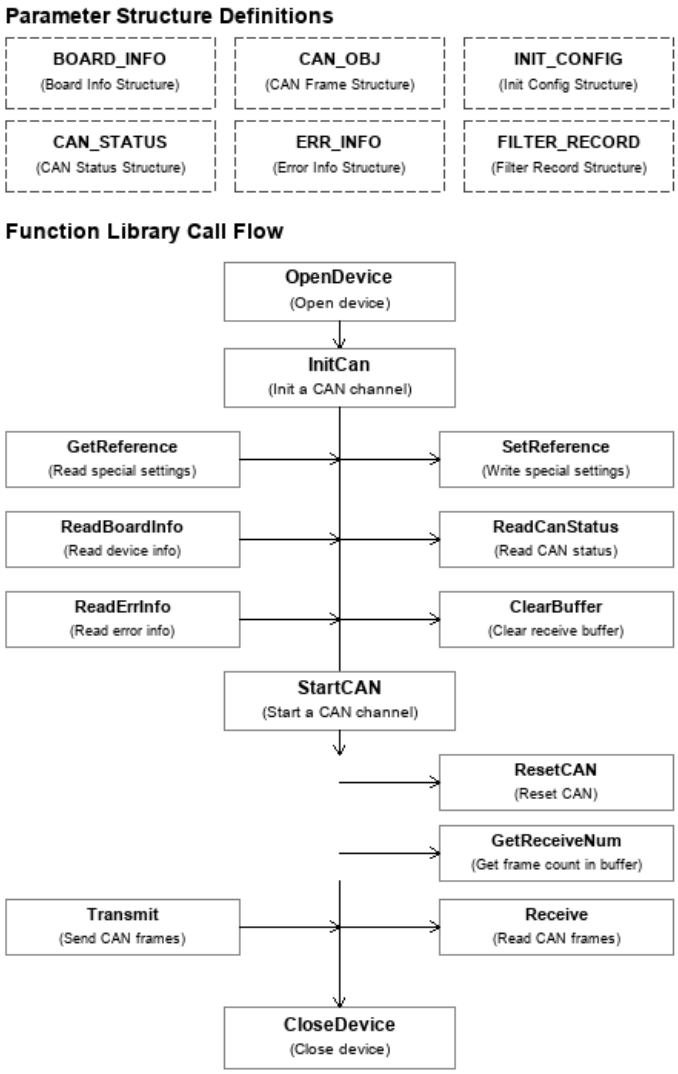


Figure 5.1 Secondary-development function calls

6. Technical Specifications

Category	Item	Specification
Connection	PC side	USB port, Type A
	CAN side	OPEN6 pluggable terminal
Interface features	USB interface	USB 2.0 full-speed, compatible with USB 1.1 and USB 3.0
	CAN interface	Compliant with ISO 11898, supports CAN 2.0A/B
	CAN baud rate	CAN communication baud rate 10 kbit/s to 1 Mbit/s; CAN FD data baud rate up to 5 Mbit/s
	Electrical isolation	2500 V, DC-DC
	CAN terminating resistor	Integrated; enabled via DIP switch
Power supply	Supply voltage	+5 V DC (USB port)

	Supply current	Maximum 130 mA
Environmental test	Operating temperature	-40°C to +85°C
	Operating humidity	15% to 90% RH, non-condensing
EMC test	Standard	EN 55024:2011-09; EN 55022:2011-12
Protection	Ingress rating	IP 20
General	Dimensions	104 mm x 64 mm x 24 mm (including terminal block)
	Weight	120 g

Table 6.1 Technical specifications

7. Frequently Asked Questions

1. Why does the ECANTools test software display "Device open error"?

Possible causes are: the wrong device type was selected, or the USBCAN device driver was not installed correctly. Check the USBCAN device properties in the PC's 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. Select USBCAN-FD as the device type.

2. Is a 120-ohm terminating resistor always necessary?

A 120-ohm terminating resistor is recommended to absorb end reflections and provide a stable physical link. For a single-node self-transmit/self-receive test, the 120-ohm terminating resistor must be connected to form a loop; otherwise the test cannot be performed. The USBCAN high-performance CAN interface card already has a 120-ohm terminating resistor connected internally.

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

Older interfaces do not support multi-card operation, but current USBCAN interface cards support up to eight USBCAN cards of the same model operating simultaneously.

4. Why is the CAN status indicator not lit?

Because all operations of the USBCAN interface card are controlled by the PC, the CAN status indicator is meaningful only after the PC has sent a command to start CAN communication.

5. Why does the system report an illegal operation when an interface function is called?

First, when using the interface functions, carefully read the function descriptions to ensure the input parameters are legal, paying special attention to the passing of pointers (addresses), or refer to the provided example programs. If the problem is still not solved, please contact our technical support.

6. How is the communication baud rate of the USBCAN interface card set?

The device provides a set of commonly used baud-rate values. To use other baud rates, please contact Guangcheng customer service for calculation. Note that the CAN controller of

the USBCAN interface card uses a 24 MHz clock; user-defined baud rates must be calculated based on this clock frequency.

7. Does the system entering standby or sleep mode affect reception?

Yes. All processing will stop, which may at worst cause a hardware receive-buffer overflow error. If a program has opened the device, it will try to prevent the system from entering standby or sleep mode to ensure normal operation. When using the USBCAN interface card, please disable the system's standby and sleep functions.

8. How should application errors be handled?

Errors are mainly divided into function-call errors and CAN(FD)-bus communication errors. Function-call errors are generally caused by parameter errors, such as a device number out of range or an incorrect type number; the error number returned by the Win32 function GetLastError is 87. Other cases involve operating on a device that has not been opened, i.e. operating on an illegal handle; corresponding Win32 standard error codes are provided according to the specific function call, and users can use GetLastError for error analysis. This debugging work should generally be completed during the design stage.

CAN(FD)-bus communication errors are generally caused by the CAN network, but may also be caused by improper user settings, such as inconsistent baud rates or calling the send function without starting the CAN controller. Most errors are handled simply in the device driver; for deeper error analysis and handling, the ReadCANStatus function can be called.

Another point to note is the data-overflow interrupt error, which can occur in two cases: (1) software receive-buffer overflow, which means the application cannot process received data in time; in this case users should optimize the application or change the communication strategy; (2) hardware receive-buffer overflow, caused by excessive PC interrupt latency. This can only be solved by improving computer performance or coordinating other nodes to appropriately reduce the sending speed.

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

The USBCAN-II FD interface card provides two CAN ports, and users can operate different ports in the same program. The card does not allow the device to be opened in shared mode; the same device cannot be opened multiple times by different processes calling OpenDevice. The OpenDevice and CloseDevice functions generally need to be called only once at application initialization and exit. When closing the device, if the current port is no longer in use, the ResetCAN function should be called first to detach the 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.

10. How should the Transmit sending function be used effectively?

The driver of the USBCAN interface card provides a transmit FIFO of about 128 frames, and each Transmit call can send up to about 128 frames. The sending speed is determined by the current computer's hardware and software performance; 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 stop responding; in this case users can add an appropriate delay in the application to reduce the sending speed.

Each frame has a timeout limit during sending: for a single frame the timeout is about 2 seconds; when sending multiple frames at once the last frame has a 2-second timeout and

the others 1 second. Send timeouts generally occur when the CAN bus is busy and the current node has a low priority; they are not function-call or communication errors, and users can implement retransmission in their programs (send timeouts are rare in low- and medium-speed networks). Therefore, when designing a system, note that CAN-bus occupancy should not exceed 60-70% of the bus capacity.

11. How should the Receive function be used effectively?

The device driver provides a 100000-frame software receive buffer, giving application programmers sufficient reaction and processing time. When the software receive buffer overflows, the driver calls ResetCAN to reset the CAN bus and sets the data-overflow interrupt flag in the CAN status. Note that both software buffer overflow and CAN-controller hardware buffer overflow use this flag.

When Wait is 0, the function returns immediately with the number of frames successfully read, or 0 if the receive buffer is empty. When Wait is non-zero, if data is already in the receive buffer the function returns immediately with the number of frames read; if the buffer is empty, the function waits for a specified timeout or until data is received before returning the number of frames successfully received. When Wait is 0xFFFFFFFF it waits indefinitely until data is received; it is advisable not to set Wait too large, and indefinite waiting should be used with even more caution.

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 with Wait 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.

Appendix: CAN 2.0 Protocol Frame Format

A.1 CAN 2.0A Standard Frame

A CAN standard frame is 11 bytes, consisting of two parts: the information part and the data part. The first three bytes are the information part.

Byte 1 is the frame information. Bit 7 (FF) indicates the frame format; in a standard frame FF = 0. Bit 6 (RTR) indicates the frame type; RTR = 0 means a data frame and RTR = 1 means a remote frame. DLC indicates the actual data length in a data frame.

Bytes 2 and 3 are the message identifier, of which 11 bits are valid.

Bytes 4 to 11 are the actual data of a data frame; they are invalid in a remote frame.

	7	6	5	4	3	2	1	0
Byte 1	FF	RTR	x	x	DLC (Data Length)			
Byte 2	(Message ID) ID.10 - ID.3							
Byte 3	ID.2 - ID.0			x	x	x	x	x
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							

Figure A.1 CAN 2.0A standard frame format

A.2 CAN 2.0B Extended Frame

A CAN extended frame is 13 bytes, consisting of two parts: the information part and the data part. The first five bytes are the information part.

Byte 1 is the frame information. Bit 7 (FF) indicates the frame format; in an extended frame FF = 1. Bit 6 (RTR) indicates the frame type; RTR = 0 means a data frame and RTR = 1 means a remote frame. DLC indicates the actual data length in a data frame.

Bytes 2 to 5 are the message identifier, of which the high 29 bits are valid.

Bytes 6 to 13 are the actual data of a data frame; they are invalid in a remote frame.

	7	6	5	4	3	2	1	0
Byte 1	FF	RTR	r1	r0	DLC (Data Length)			
Byte 2	(Message ID) ID.28 - ID.21							
Byte 3	ID.20 - ID.13							
Byte 4	ID.12 - ID.5							
Byte 5	ID.4 - ID.0					x	x	x
Byte 6	Data 1							
Byte 7	Data 2							
Byte 8	Data 3							
Byte 9	Data 4							
Byte 10	Data 5							
Byte 11	Data 6							
Byte 12	Data 7							
Byte 13	Data 8							

Figure A.2 CAN 2.0B extended frame format