

ZS1311, USB CAN FD Adaptor

User Manual

1.1.0, 28 Jan 2026

ZILOGIC

Table of Contents

1. Revision History	1
2. Introduction	2
2.1. Technical Specification	2
2.2. Applications	2
2.3. Ordering Information	2
2.4. Package Contents	2
2.5. Additional softwares and manuals	2
3. Usage	3
3.1. Hardware Setup	3
3.2. LEDs	3
3.3. Connector	3
3.4. Termination Resistor	5
3.5. Usage in Linux	5
3.6. Usage in Windows	6
3.7. Loopback Connection	8
4. Known Issues and Limitations	9
4.1. CAN Speeds	9
4.2. Multiple Channel Access in Windows	9
A. Limited Hardware Warranty	10
A.1. LIMITED HARDWARE WARRANTY	10
A.2. EXCLUSIONS	10
A.3. HARDWARE RETURN PROCEDURES	10
A.4. ADDITIONAL RESPONSIBILITIES	11
A.5. LIMITATION OF LIABILITY	11

Chapter 1. Revision History

Revision	Date	Comment
1.0.0	21 Feb 2025	Initial Revision.
1.1.0	28 Jan 2026	zscanfd driver usage with python-can in windows.

Chapter 2. Introduction

ZS1311, USB CAN FD Adaptor is a signal and power-isolated CAN transceiver that allows a system with a USB to connect with a Controller Area Network (CAN). It is based on the open-source CANTact Pro hardware.

2.1. Technical Specification

The USB CAN FD Adaptor has the following specification.

- Host Interface is High-speed USB 2.0
- Supported CAN Standards
 - CAN HS, ISO 11898-2
 - CAN FD, ISO 11898-1
- 2 CAN Channels
- CAN bus connection via D-Sub connector
- Galvanic isolation between CAN and USB
- Arbitration bitrates from 30Kbps to 1 Mbps
- Data bitrates from 30Kbps to 5 Mbps
- Microchip CAN Transceiver MCP25612
- CAN termination can be activated through a DIP Switch
- 5V Power supply via USB
- Current Consumption: 5V 0.08A
- Dimension (L x W x H): 125mm x 79mm x 28mm

2.2. Applications

- Automate testing of CAN based systems.
- Troubleshoot ECUs by sniffing messages on a CAN bus.
- Exploring and rapid prototyping of CAN devices.

2.3. Ordering Information

- Product Name: ZS1311, USB CAN FD Adaptor
- Model No: ZS-CANFD-P1B

2.4. Package Contents

- 1x USB CAN FD Adaptor
- 1x USB Cable
- 1x Y-Splitter Cable

2.5. Additional softwares and manuals

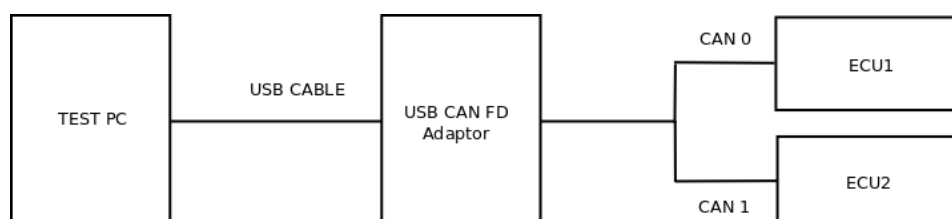
To download other manuals and softwares, please visit [Product Page \[www.zilogic.com/product-zs1311.html\]](http://www.zilogic.com/product-zs1311.html)

Chapter 3. Usage

3.1. Hardware Setup

The following diagram shows an example of a hardware setup. Here a test PC is connected to the USB CAN Adaptor using a USB cable. The two CAN channels of the USB CAN Adaptor is connected to two ECUs using a Y-Splitter cable.

Figure 3.1. Example Hardware Setup



3.2. LEDs

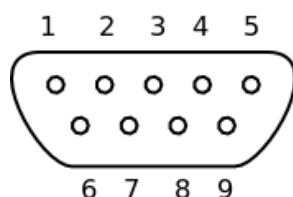
The LED indications, are primarily used indicate TX and RX activity in each of the channels. The details of the LED indications are shown in the below table.

Table 3.1. LED Indication

LED	Color	State	Description
TX1	Red	Blink	Indicates Tx in CAN1
		Solid	Indicates link-up in CAN1
RX1	Green	Blink	Indicates Rx in CAN1
		Solid	Indicates link-up in CAN1
TX2	Red	Blink	Indicates Tx in CAN2
		Solid	Indicates link-up in CAN2
RX2	Green	Blink	Indicates Rx in CAN2
		Solid	Indicates link-up in CAN2

3.3. Connector

Figure 3.2. D-Sub Male Pin Numbering



The pinout details of the Dual Channel CAN Connector is shown in the following table.

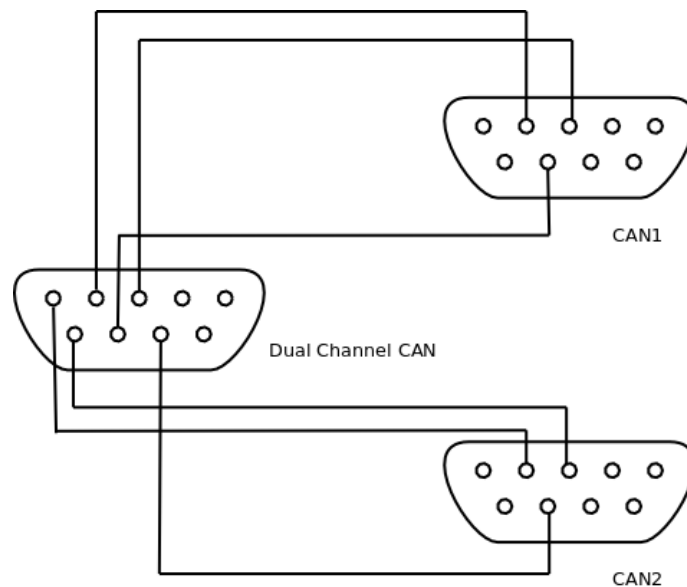
Table 3.2. Dual Channel CAN Pinout

Pin	Description
1	CANL2

Pin	Description
2	CANL1
3	GND
4	-
5	-
6	GND
7	CANH1
8	CANH2
9	-

The two CAN channels can be made available in separate DB9 connectors using a Y-Splitter.

Figure 3.3. Y-Splitter Schematic



The pinout details of the splitted out CAN Connectors is shown in the following tables.

Table 3.3. CAN Pinout

Pin	Description
1	-
2	CANL
3	GND
4	-
5	-
6	-
7	CANH
8	-
9	-

3.4. Termination Resistor

Termination resistor absorbs the CAN signal energy and ensure that this is not reflected back from cable ends. To enable / disable the termination resistor on CAN1 channel, turn ON / OFF the DIP switch 1. To enable / disable the termination resistor on CAN2 channel, turn ON / OFF the DIP switch 2.

3.5. Usage in Linux

3.5.1. Supported Linux Distributions

Supported versions of Linux

- Ubuntu 20.04 and above

3.5.2. Dependencies

Steps to install the dependencies are provided below.

- Download python-can package with zscanfd support from [Product Page](http://www.zilogic.com/product-zs1311.html) [www.zilogic.com/product-zs1311.html]
- Install CAN utils package in Ubuntu using the following command.

```
apt-get install can-utils
```

- Run the command to install the python-can package. Note: Replace `package-version` with actual release version.

```
pip install python_can-<package-version>-py3-none-any.whl
```

3.5.3. Setup

The CAN device is visible as a network interface. After connecting the USB CAN FD Adaptor to the system, check the names assigned to the device using the following command.

```
$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
  DEFAULT group default qlen 1000 link/loopback 00:00:00:00:00:00 brd
  00:00:00:00:00:00
9: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode
  DEFAULT group default qlen 10 link/can
10: can1: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode
  DEFAULT group default qlen 10 link/can
```

The above command shows the names "can0" and "can1" assigned to the two channels of the CAN FD Adaptor.

The CAN device can be then be configured and enabled using the `ip` command. For example, to configure the CAN device at 1Mbps arbitration bitrate and 5Mbps data bitrate, use the following command.

```
$ ip link set can0 up type can bitrate 100000 fd on dbitrate 500000
```

The CAN1 TX and RX LEDs will glow solid to indicate that the link is up.

3.5.4. Command-line Example

Refer Section 3.7, "Loopback Connection" to setup a loopback connection between CAN1 and CAN2 channels. The following command is used to transmit 8 bytes of data on CAN1 and receive the message on CAN2.

```
cansend can0 01b#1122334455667788
```

The following command is used to transmit 64 bytes from CAN1 to CAN2, using CAN FD.

```
cansend can0 01B##1.001122334455667788990011223344556677889900112233445566  
778899001122334455667788990011223344556677889900112233445566  
77889900112233
```

The received data can be viewed through the `candump` command. The following command displays the data received on the interface `can0`.

```
candump can1
```

3.5.5. Python Example

Refer Section 3.7, "Loopback Connection" to setup a loopback connection between CAN1 and CAN2 channels. The following code is used to transmit on CAN1 and receive the message on CAN2.

```
# Get access to the CAN bus  
tx_bus = can.interface.Bus(interface="socketcan",  
                           channel=tx_dev,  
                           receive_own_messages=False)  
rx_bus = can.interface.Bus(interface="socketcan",  
                           channel=rx_dev,  
                           receive_own_messages=False)  
  
# Send the message  
tx_msg = can.Message(arbitration_id=0xabcd, data=[1,2,3,4])  
tx_bus.send(tx_msg)  
  
# Receive back the message  
rx_msg = rx_bus.recv(0.1)  
print(f"Received message: {rx_msg}")
```

3.6. Usage in Windows

3.6.1. Supported Versions

Supported versions of Windows

- Windows 10
- Windows 11

The following steps are used for testing using python-can library in Windows.

3.6.2. Dependencies

Steps to install the dependencies are provided below.

- Download zscanfd package from [Product Page](http://www.zilogic.com/product-zs1311.html) [www.zilogic.com/product-zs1311.html] and extract it.
- Find the whl for your python version under wheels directory to install zscanfd driver. If your python version is 3.9 and zscanfd package version is 0.3.0, suitable wheel file will be zscanfd-0.3.0-cp39-cp39-linux_x86_64.whl.
- Check your python version using the following command.

```
python --version
```

- Run the following command to install the zscanfd driver. Note: Replace `package-version` and `python-version` with actual version.

```
pip install zscanfd-<package-version>-cp<python-version>-cp<python-version>-win_amd64.whl
```

- Download python-can package with zscanfd support from [Product Page](http://www.zilogic.com/product-zs1311.html) [www.zilogic.com/product-zs1311.html]
- Run the command to install the python-can package. Note: Replace `package-version` with actual release version.

```
pip install python_can-<package-version>-py3-none-any.whl
```

3.6.3. Python Example

Refer Section 3.7, "Loopback Connection" to setup a loopback connection between CAN1 and CAN2 channels. The following code is used to transmit and receive CAN messages between two channels.

```
# Get access to the CAN bus
tx_bus = can.interface.Bus(interface="zscanfd",
                           channel=tx_dev,
                           bitrate=1_000_000,
                           data_bitrate=4_000_000,
                           fd=True,
                           receive_own_messages=False)
rx_bus = can.interface.Bus(interface="zscanfd",
                           channel=rx_dev,
                           bitrate=1_000_000,
                           data_bitrate=4_000_000,
                           fd=True,
                           receive_own_messages=False)

# Send the message
tx_msg = can.Message(arbitration_id=0xabcd,
                    data=[1,2,3,4],
                    is_fd=True,
                    bitrate_switch=True,
                    is_extended_id=True,
                    dlc=4)
tx_bus.send(tx_msg)

time.sleep(0.5)

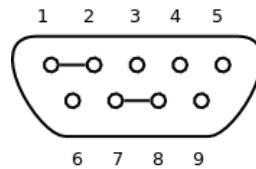
# Receive back the message
```

```
rx_msg = rx_bus.recv(0.1)
print(f"Received message: {rx_msg}")
```

3.7. Loopback Connection

The following diagram shows an example of a loopback connection. Pin 1 and pin 2 connected using the jumper wire. Likewise Pin 7 and pin 8 are connected. Pin 1 and 2 consider as low, pin 7 and 8 consider as high.

Figure 3.4. Loopback Connection



Chapter 4. Known Issues and Limitations

4.1. CAN Speeds

The USB CAN FD Adaptor support various speeds. Limitations: * Speeds less than 30Kbps are not supported. * 5Mbps is currently not supported because of a firmware limitation. with respect to transmit delay compensation.

The following are the tested speeds for CAN HS.

Table 4.1. CAN HS Speeds

Arbitration Bitrate	Sampling point (Linux)
5Kbps	0.875
30Kbps	0.875
125Kbps	0.875
500Kbps	0.875
1Mbps	0.750

The following are the tested speeds for CAN FD.

Table 4.2. CAN FD Speeds

Arbitration Bitrate	Data Bitrate	Sampling point (Linux)
1Mbps	1Mbps	0.750
1Mbps	1.2Mbps	0.760
1Mbps	1.5Mbps	0.750
1Mbps	2Mbps	0.750
1Mbps	2.4Mbps	0.720
1Mbps	3Mbps	0.750
1Mbps	4Mbps	0.733
1Mbps	4.02Mbps	0.733
1Mbps	5Mbps	0.750
1Mbps	8Mbps	0.733

4.2. Multiple Channel Access in Windows

- The python-can package (>=4.5.0) does not support accessing multiple CAN channels, simultaneously.
- The workaround is to directly use the `zscanfd` package where both channels can be accessed simultaneously.

Appendix A. Limited Hardware Warranty

The warranties provided by Zilogic Systems in this Limited Hardware Warranty apply only to Hardware Products you purchase for your use, and not for resale. The term "Hardware Product" means a computing device with a specific function and limited configuration ability.

A.1. LIMITED HARDWARE WARRANTY

Zilogic Systems warrants that the hardware components of its Hardware Product shall be free from material defects in design, materials, and workmanship and will function, under normal use and circumstances, in accordance with the documentation provided, for a period of one (1) year from the date of purchase of the Hardware Product.

Your sole and exclusive remedy, and Zilogic Systems' sole and exclusive liability for defective hardware components, shall be that Zilogic Systems, subject to the terms and conditions of this Section, and solely upon confirmation of a defect or failure of a hardware component to perform as warranted, shall at its sole option, either repair or replace the nonconforming hardware component. All replacement parts furnished to you under this warranty shall be refurbished and equivalent to new, and shall be warranted as new for the remainder of the original warranty period. All defective parts, which have been replaced, shall become the property of Zilogic Systems. All defective parts that have been repaired shall remain your property.

A.2. EXCLUSIONS

The foregoing warranties and remedies shall be void as to any Hardware Products damaged or rendered unserviceable by one or more of the following: (1) improper or inadequate maintenance by anyone other than Zilogic Systems or Zilogic Systems' authorized engineers, (2) interfacing supplied by anyone other than Zilogic Systems, (3) modifications, alterations or additions to the Hardware Products by personnel not certified by Zilogic Systems or Zilogic Systems' authorized engineers to perform such acts, or other unauthorized repair, installation or other causes beyond Zilogic Systems' control, (4) unreasonable refusal to agree with engineering change notice programs, (5) negligence by any person other than Zilogic Systems or Zilogic Systems' authorized engineers, (6) misuse, abuse, accident, electrical irregularity, theft, vandalism, fire, water or other peril, (7) damage caused by containment and/or operation outside the environmental specifications for the Hardware Products, (8) alteration or connection of the Hardware Products to other systems, equipment or devices (other than those specifically approved by Zilogic Systems) not in accordance to the board and on-board device specifications (9) any use that is inconsistent with the user manual supplied with the Hardware Product. The warranty period is not extended if Zilogic Systems repairs or replaces a warranted product or any parts. Zilogic Systems may change the availability of limited hardware warranties, at its discretion, but any changes will not be retroactive.

A.3. HARDWARE RETURN PROCEDURES

If a Hardware Product or one of its component parts does not function as warranted during the warranty period, and such nonconformance can be verified by Zilogic Systems, Zilogic Systems, at its election, will provide either return and replacement service or replacement with a refurbished part/unit for the Hardware Product under the type of warranty service Zilogic Systems designates for that Hardware Product. A defective Hardware Product or one of its component parts may only be returned to Zilogic Systems upon Zilogic Systems' prior written approval. Any such approval shall reference an RMA number issued by an authorized Zilogic Systems service representative. If

you do not register the Hardware Product with Zilogic Systems, you may be required to present proof of purchase as evidence of your entitlement to warranty service. The Hardware Product's serial number will be required for all RMA cases.

Transportation costs, if any, incurred in connection with the return of a defective item to Zilogic Systems shall be borne by You. Any transportation costs incurred in connection with the redelivery of a repaired or replacement item to You by Zilogic Systems shall be borne by Zilogic Systems; provided, however, that if Zilogic Systems determines, in its sole discretion, that the allegedly defective item is not covered by the terms and conditions of the warranty or that a warranty claim is made after the warranty period, the cost of the repair by Zilogic Systems, including all shipping expenses, shall be reimbursed by You.

A.3.1. HARDWARE REPLACEMENT PROCEDURES

Zilogic Systems will attempt to diagnose and resolve your problem over the phone or e-mail. Upon determination of the hardware issue is related to a malfunction of one of the Hardware Product components, an RMA process will be initiated by Zilogic Systems.

For Warranty Replacement service, it is required that you deliver the faulty unit to a location Zilogic Systems designates, and provide courier name and tracking number to Zilogic Systems. After the Faulty unit is returned to Zilogic Systems, Zilogic Systems will use commercially reasonable efforts to ship the replacement hardware within fourteen (14) business days. Actual delivery times may vary depending on availability of the spares and customer's location.

A.4. ADDITIONAL RESPONSIBILITIES

You agree:

- To provide Zilogic Systems or its partner with sufficient and safe access to your facilities to permit Zilogic Systems to fulfill its obligations.
- To ship back the faulty Hardware Product (or replaceable unit) suitably packaged, quoting the RMA number, to the Zilogic Systems designated location.
- You shall ship the faulty Hardware Product once Zilogic Systems approves the RMA and provide the courier name and tracking number.
- To securely erase from any Hardware Product you return to Zilogic Systems for any reason all programs and data not provided by Zilogic Systems with the Hardware Product. You acknowledge that in order to perform its responsibilities under this Limited Hardware Warranty, Zilogic Systems may ship all or part of the Hardware Product or its software to third party locations around the world, and you authorize Zilogic Systems to do so.

A.5. LIMITATION OF LIABILITY

Zilogic Systems' development kits are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, not in applications where failure or malfunction of a Zilogic Systems product can reasonably be expected to result in personal injury, death or severe property or environmental damage.

NOTWITHSTANDING ANYTHING ELSE IN THIS AGREEMENT OR OTHERWISE, NEITHER ZILOGIC SYSTEMS NOR ITS SUPPLIERS WILL BE LIABLE WITH RESPECT TO ANY SUBJECT MATTER OF THIS AGREEMENT UNDER ANY CONTRACT, NEGLIGENCE, STRICT LIABILITY, OR OTHER LEGAL OR EQUITABLE THEORY, REGARDLESS OF WHETHER ZILOGIC SYSTEMS OR ITS SUPPLIERS WERE ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, FOR: (i) ANY PUNITIVE, INCIDENTAL OR CONSEQUENTIAL DAMAGES OR LOST DATA OR LOST PROFITS; OR (ii) FOR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY OR SERVICES; OR (iii) FOR ANY CLAIMS BASED ON ANY ERROR,

DEFECT OR NONCONFORMITY IN THE PRODUCTS OR SERVICE, FOR ANY AMOUNT IN EXCESS OF THE PRICE PAID TO ZILOGIC SYSTEMS FOR SUCH DEFECTIVE PRODUCT(S) OR SERVICE; OR (IV) FOR ALL OTHER CLAIMS NOT RELATED TO AN ERROR, DEFECT OR NONCONFORMITY IN THE PRODUCTS, ANY AMOUNTS IN EXCESS IN THE AGGREGATE OF THE AMOUNT PAID TO ZILOGIC SYSTEMS HEREUNDER DURING THE THREE (3) MONTHS PRECEDING THE DATE THE CAUSE OF ACTION AROSE.

WARRANTY DISCLAIMER. EXCEPT AS STATED HEREIN, ZILOGIC SYSTEMS MAKES NO WARRANTIES WITH RESPECT TO any PRODUCT, license or SERVICE AND DISCLAIMS ALL Statutory or IMPLIED WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, or arising from a course of dealing or usage of trade and any WARRANTIES OF NONINFRINGEMENT. ZILOGIC SYSTEMS DOES NOT WARRANT THAT THE ZILOGIC SYSTEMS PRODUCT(s) WILL MEET any REQUIREMENTS or THAT THE OPERATION OF ZILOGIC SYSTEMS PRODUCTS WILL BE UNINTERRUPTED OR ERROR FREE.