

# Hadou-CAN Quickstart

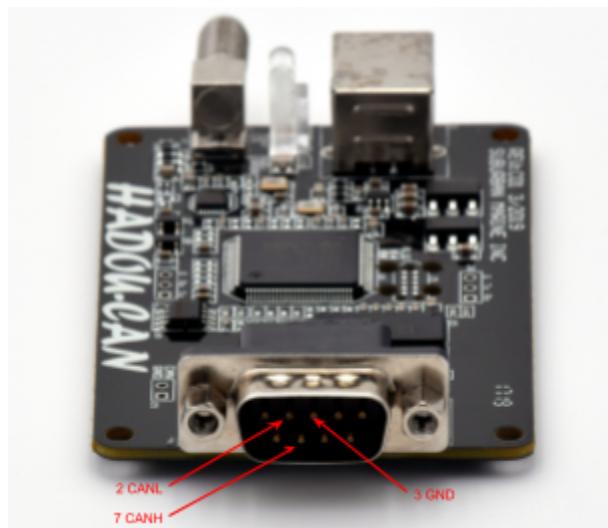
© 2019-2020, 2025-2026 Suburban Marine, Inc.



## Introduction

The Hadou-CAN is a USB to CAN bus adapter. It allows a standard Windows or Linux computer to send and receive CAN bus frames via a USB interface. CAN bus is a very useful network found in many common consumer and industrial settings. It can be used to connect to CAN buses found in cars and trucks via the OBD-II port, to diagnose NMEA2000 networks found on ships, or used to control a custom hobby robotics project.

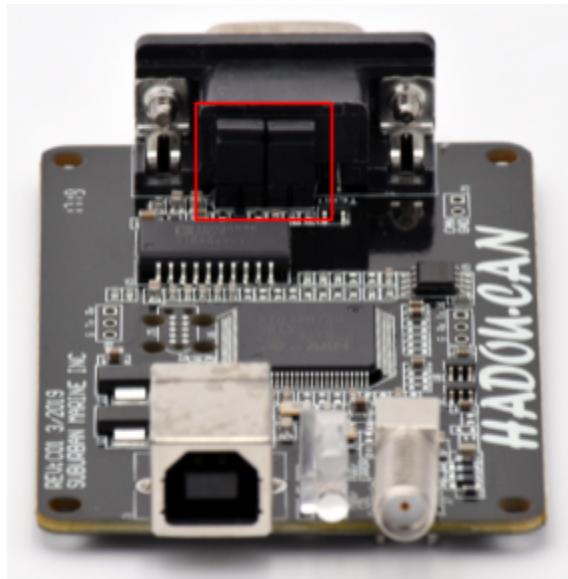
## CAN Pinout



Pin	Function
2	CAN Low
3	CAN Ground, not connected to USB/SMA ground. May be omitted on short CAN runs if common mode voltage requirements are met.
7	CAN High

All other pins on the DE-9 connector are not connected internally and should be left open.

## Internal CAN Bus Termination



Internal termination resistors can be selected by inserting 2x 0.1in jumpers onto pins 1&2 and 3&4 of J2, located adjacent to the DE9 connector. This will connect a 120 ohm split termination right after the D Sub connector. The device ships from the factory with the termination enabled.

## Usage Examples

Please see the User Guide for full protocol information:

[https://suburbanmarine.io/public/hadoucan/doc/Hadou-CAN\\_User\\_Guide.pdf](https://suburbanmarine.io/public/hadoucan/doc/Hadou-CAN_User_Guide.pdf)

### Check version

1. Plug in USB cable, wait for enumeration
2. Open minicom, putty, or other terminal emulation program to the correct serial port. Bit rate does not matter.
3. Type `!version` and press the enter key
4. A string should appear in the format `VXXXXY`, eg `V0104`

### Check Serial number

1. Plug in USB cable
2. Open minicom, putty, or other terminal emulation program
3. Type `!serial` and press the enter key
4. A hex string should appear, eg `003D00393038510736323733`

## Send first packet

1. Plug in USB cable
2. Open minicom, putty, or other terminal emulation program
3. Open channel by typing O and pressing enter. If you have two Hadou-CANs, make sure both devices have been opened, otherwise it will not be able to send a packet.
4. Send a can packet by typing t00021234 and pressing enter

## Use slcand / Wireshark on Linux

### Start slcand

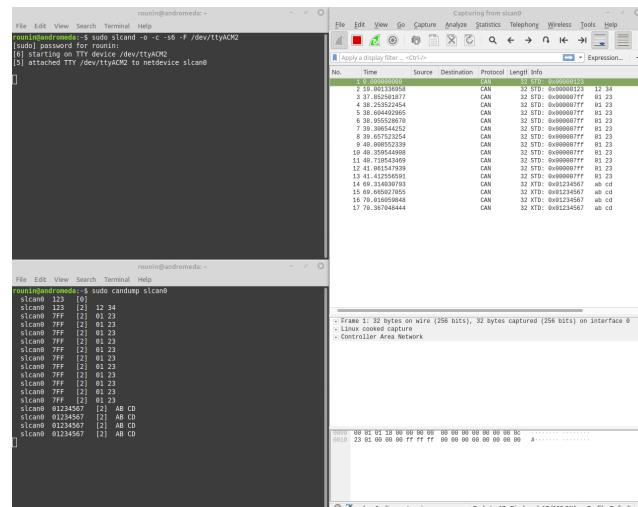
1. sudo slcand -o -c -s6 -F /dev/ttyACM0
2. sudo ip link set up slcan0

### Start candump & Wireshark

- candump slcan0

### Send packet at 500kbps

- cansend slcan0 123#112233



## Quick notes

- Auto poll is turned on - messages should print to the console when they arrive. A z will be sent back on the terminal when you send a CAN packet.
- Default bit rate is 500kbps. EXT ids are on, CAN FD is on, BRS is off.
- You need to open the channel by sending O\r (letter O and a carriage return, eg 0x4F 0x0D) before you can send or receive packets. If your terminal program adds line feeds along with the carriage returns it will not work, usually that is configurable.
- The Hadou-CAN is mostly compatible with the protocol described in this document: [http://www.can232.com/docs/can232\\_v3.pdf](http://www.can232.com/docs/can232_v3.pdf), with some extensions for CAN FD and new features such as timesync.
- You may need to send an extra carriage return \r to synchronize the application and flush any bytes the host operating system may send while probing / enumerating the USB device.
-

# References

## User Guide

<https://f000.backblazeb2.com/file/SM-PUB/hadoucan/Hadou-CAN+User+Guide.pdf>

## Datasheet

<https://f000.backblazeb2.com/file/SM-PUB/hadoucan/Hadou-CAN+Datasheet.pdf>

## Quickstart Guide

<https://f000.backblazeb2.com/file/SM-PUB/hadoucan/Hadou-CAN+Quickstart+Guide.pdf>

# Revision History

Date	Revision	Comments
20200416	A	Initial private release
20250712	B	Update links
20260119	C	Correct line ending typo