

# **FINTEK**

## **F81605**

### **USB to 2 CANFD**

# **Driver Installation Guide**

## **for Linux**

v1.06

Jun 29, 2026

1 / 9

## Revision History

Date	Version	Revision History
2025/5/21	v1.00-v20250430	Initial version.
2025/7/9	v1.01-v20250709	Add J1939 SDK support. Fix kernel 5.10/5.14 build issue. Fix CANBUS status report issue.
2025/7/23	v1.02-v20250723	Fix RX non-used frame when idle. Change error message with dynamic debug, default no display. Add parameter "disable_canfd" to force no FD function in driver.
2025/9/19	v1.03-v20250919	Fix SocketCAN state not matched when state change. Fix TX err count issue when Oneshot mode. Support Presume ACK.
2026/1/7	v1.04-v20260107	Fix kernel 5.9 build issue. Fix F1939 read issue. Fix RX err mechanism. Fix don't read information when bus-off.
2026/2/26	v1.05-v20260226	Fix kernel 6.16 build issue. Add GPIO1/2 control with document In Q&A section.
2026/6/29	v1.06-v20260629	Add EEPROM SDK support Add EEPROM + udev rename network device support

## 1. Preliminary

This document is for Fintek F81605 USB to 2 CANFD driver installation in Linux and can-utils to verify the CANBUS.

## 2. Building Driver (Debian / Ubuntu based)

1. sudo su
2. Prepare the kernel tree & compiler tools for your distribution.
  1. apt-get update
  2. apt-get install build-essential gcc net-tools can-utils
3. unzip driver & driver.zip file
4. cd driver
5. make clean ; make ; make install
6. reboot
7. Use the following command to check CANBUS device is available (can0/can1/... etc.):  
ls /sys/class/net/ -al

```
root@code-h11h4im:/home/code/ddd/old/hpeter/fintek/F81605_2104/driver# dmesg | grep F81605 | tail -n 5
[172344.068930] F81605 1-2.4.1:1.0 can1: device disconnected
[172348.829596] F81605 1-2.4.1:1.0: f81605_probe: Fintek F81605 driver version: v1.00-v20250430
[172348.841651] F81605 1-2.4.1:1.0: Channel #0 registered as can0
[172348.843086] F81605 1-2.4.1:1.0: Channel #1 registered as can1
[172348.843186] usbcore: registered new interface driver F81605
root@code-h11h4im:/home/code/ddd/old/hpeter/fintek/F81605_2104/driver# ls /sys/class/net/ -al
total 0
drwxr-xr-x  2 root root 0  5月 21 09:25 .
drwxr-xr-x 79 root root 0  5月 21 09:25 ..
lrwxrwxrwx  1 root root 0  5月 21 09:25 can0 -> ../../devices/pci0000:00/0000:00:14.0/usb1/1-2/1-2.4/1-2.4.1/1-2.4.1:1.0/t/can0
lrwxrwxrwx  1 root root 0  5月 21 09:25 can1 -> ../../devices/pci0000:00/0000:00:14.0/usb1/1-2/1-2.4/1-2.4.1/1-2.4.1:1.0/t/can1
lrwxrwxrwx  1 root root 0  5月 19 09:33 enp1s0 -> ../../devices/pci0000:00/0000:00:01.0/0000:01:00.0/net/enp1s0
lrwxrwxrwx  1 root root 0  5月 19 09:33 enp7s2 -> ../../devices/pci0000:00/0000:00:1d.1/0000:06:00.0/0000:07:02.0/net/enp7s2
lrwxrwxrwx  1 root root 0  5月 19 09:33 lo -> ../../devices/virtual/net/lo
root@code-h11h4im:/home/code/ddd/old/hpeter/fintek/F81605_2104/driver#
```

### 3. Configure CANBUS

#### CAN 2.0A/B only

The following examples will use command to configure “can0” to bit-rate 250000, sample-point 0.875 and error restart with 100ms.

1. ip link set can0 down
2. ip link set can0 type can restart-ms 100
3. ip link set can0 type can bitrate 250000 sample-point 0.875
4. ip link set can0 type can berr-reporting on
5. ifconfig can0 txqueuelen 1000
6. tc qdisc add dev can0 root handle 1: pfifo
7. ip link set can0 up

If you want to change the CANBUS setting in your application, we can use “system()” to execute above command or “canconfig.c” in “config tools” programmatically via SocketCAN/Netlink.

#### CANFD

Most setting are the same with CAN2.0A/B only. The difference is following.

1. We must determine the CANFD is ISO or Non-ISO mode.
2. The data bitrate (1/2/4/5Mbits)
3. All device node’s sample-point must be the same.

So we modify the command from “CAN2.0A/B only” step3 as following config.

CANFD with frame rate 1M, data rate 5M with ISO mode, the sample-point & dsample-point must be the same.

- ip link set can0 type can bitrate 1000000 dbitrate 5000000 fd on fd-non-iso off sample-point 0.75 dsample-point 0.75

## 4. Using can-utils to operate CANBUS

We can get can-utils with following command.

- Debian/Ubuntu
  - apt-get install can-utils
- Fedora
  - yum install can-utils
- Centos/RHEL
  - Source code download link: <https://github.com/linux-can/can-utils>

We'll use "candump" to receive data, "cangen" & "cansend" to send data. The "cangen" will send random data & ID and "cansend" will send specific data & ID to CANBUS.

```
root@code-ms7c82:/home/code# cansend can0 555#55
root@code-ms7c82:/home/code# cansend can0 555##1.55
root@code-ms7c82:/home/code# cansend can0 00000555##1.55
root@code-ms7c82:/home/code# █

^Croot@code-ms7c82:/home/code#
root@code-ms7c82:/home/code# candump any,0:0,#FFFFFFFF -t z -e -x
(000.000000) can0 TX - - 555 [1] 55
(000.000046) can1 RX - - 555 [1] 55
(004.991369) can1 RX B - 555 [01] 55
(004.991360) can0 TX B - 555 [01] 55
(012.312236) can1 RX B - 00000555 [01] 55
(012.312227) can0 TX B - 00000555 [01] 55
```

We can send canframe by "cansend" with following operations.

- CAN2.0 A/B
  - SFF: cansend can0 555#55
  - EFF: cansend can0 00000000#55
- CANFD
  - SFF: cansend can0 555##1.55
  - EFF: cansend can0 00000000##1.55

The range of SFF's ID are from 000~7ffh, EFF's are from 00000000~1FFFFFFh.

We can access the website to get more detail usage and source code.

Manpage manual:

<http://manpages.ubuntu.com/manpages/bionic/man1/candump.1.html>

<http://manpages.ubuntu.com/manpages/bionic/man1/cangen.1.html>

<http://manpages.ubuntu.com/manpages/bionic/man1/cansend.1.html>

Source code:

<https://github.com/linux-can/can-utils/blob/master/candump.c>

<https://github.com/linux-can/can-utils/blob/master/cangen.c>

<https://github.com/linux-can/can-utils/blob/master/cansend.c>

## 6. Q&A

Q1: “No Buffer space available” with “cangen” tools.

```
root@code-H11H4-IM:/home/code#  
root@code-H11H4-IM:/home/code# cangen can0 -g 0  
write: No buffer space available  
root@code-H11H4-IM:/home/code#
```

A1: To enlarge tx buffer by command “ifconfig can0 txqueuelen 1000” or ignore the message with parameter “cangen -i”

Q2: “non-retpoline compiler” error with make

```
root@code-desktop:/home/code/d/driver# make  
make -C /lib/modules/4.15.0-50-generic/build M=/home/code/d/driver modules  
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-50-generic'  
arch/x86/Makefile:245: *** You are building kernel with non-retpoline compiler, please update your compiler.. Stop.  
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-50-generic'  
Makefile:8: recipe for target 'default' failed  
make: *** [default] Error 2
```

A2: The current kernel & compiler is not matched. Please update the compiler with following command:

```
apt-get install gcc-5 gcc-5-base
```

Q3: Can't load driver when system reboot with Kylin (銀河麒麟).

A3: Run:

```
sudo kysec_set -n exectl -v original /lib/modules/`uname -r`/updates/F81605.ko  
to entrust the driver and reboot.
```

Q4: How to control GPIO1/2?

A4: The control method is shown below (assuming F81605 is can0/can1):

**Control path:**

```
GPIO1: /sys/class/net/can0/gpio_ctrl
```

```
GPIO2: /sys/class/net/can1/gpio_ctrl
```

**Read method:**

```
cat /sys/class/net/can0/gpio_ctrl
```

**Set method:**

```
echo 1 > /sys/class/net/can0/gpio_ctrl # set GPIO1 to output high
```

```
echo 0 > /sys/class/net/can0/gpio_ctrl # set GPIO1 to output low
```

Q5: How to use EEPROM to rename network device name.

A5:

1. Find USB Hierarchical Path (usb\_path)

- Load the driver first.
- Read the usb\_path from sysfs:

```
cat /sys/class/net/can0/usb_path
```

```
cat /sys/class/net/can1/usb_path
```

Expected output example:

```
1-5__0
```

```
1-5__1
```

The format is: <USB-port-path>\_\_<channel-id>

2. Select one rename method in `99-fintek-f81605-fix-name.rules`

1. usb\_path:

Match by actual USB port path. Replace \*-xxx\_\_0 with your real path.

```
can-fintek-0 / can-fintek-1
```

2. usb\_fix:

Match by EEPROM fix\_com\_number value (always available when EEPROM is programmed).

```
can-ftk-fix- $\$attr\{usb\_fix\}$  (e.g., can-ftk-fix-19 / can-ftk-fix-36)
```

Note: usb\_fix is only available when EEPROM contains fix\_com\_number setting

3. Install the rule:

- `install -m 0644 99-fintek-f81605-fix-name.rules /etc/udev/rules.d/99-fintek-f81605-fix-name.rules`
- `udevadm control --reload`

- udevadm settle
4. Remove the rule:
    1. `rm -f /etc/udev/rules.d/99-fintek-f81605-fix-name.rules`
    2. `rm -f /run/udev/rules.d/99-fintek-f81605-fix-name.rules`
    3. `udevadm control --reload`
    4. `udevadm settle`

If it install complete, we'll see the rename message from dmesg.

```
[ 4121.614586] F81605 1-5:1.0: f81605_probe: Fintek F81605 driver version: v1.05-v20260226
[ 4121.667061] F81605 1-5:1.0: Channel #0 registered as can0
[ 4121.667405] F81605 1-5:1.0: Channel #1 registered as can1
[ 4121.667454] usbcore: registered new interface driver F81605
[ 4121.671210] F81605 1-5:1.0 can-ftk-fix-19: renamed from can0
[ 4121.673662] F81605 1-5:1.0 can-ftk-fix-36: renamed from can1
```