



FINTEK

F81601A

PCIe to 2 CANFD

Driver Installation Guide

for Linux

v1.10.1

Oct 13, 2026

1 / 11

Revision History

Date	Version	Revision History
2024/1/2	v1.00	Initial version.
2024/1/16	v1.01	Fix for RHEL9 kernel 5.14 not support can led issue.
2024/3/27	v1.02	Default set to internal clock.
2025/7/9	v1.03-20250515v2	Enable SSP for high baudrate / Isolator. Auto detect oscillator in driver.
2025/8/29	v1.04-20250829	Add parameter "auto_affinity" to change IRQ auto balance mode.
2026/3/2	v1.05-20260302	Fix kernel 6.16 build issue. Change CAN minimum baudrate prescale default from 8 to 4 and SSP value from 8 to 6 for compatibility.
2026/3/23	v1.06-20260323	Fix kernel 5.11 build issue. Fix compability for TEC.
2026/3/30	v1.07-20260330	Fix error count calculation. Fix TEC retransmission buffer handling.
2026/6/15	v1.08-20260615	Fix TEC retransmission buffer handling. Fix SJW/DSJW calculate method. Add EEPROM rename functional (via udev).
2026/7/17	v1.09-20260717	Fix kernel 6.19 build issue.
2026/8/25	v1.10-20260825	Added compatibility support for AMD PEG ports. Fixed an issue with AMD MMIO reads.
2026/10/13	V1.10.1-20261013	Documentation updates only. Driver identical to v1.10-20260825.

1. Preliminary

This document is for Fintek F81601A PCIe 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-H11H4-IM:/home/code/ddd/old/hpeter/fintek/F81601/driver# ls /sys/class/net/ -al
total 0
drwxr-xr-x  2 root root 0  五  9 11:39 .
drwxr-xr-x 61 root root 0  五  9 08:51 ..
lrwxrwxrwx  1 root root 0  五  9 11:39 can0 -> ../../devices/pci0000:00/0000:00:01.0/0000:01:00.0/net/can0
lrwxrwxrwx  1 root root 0  五  9 11:39 can1 -> ../../devices/pci0000:00/0000:00:01.0/0000:01:00.0/net/can1
lrwxrwxrwx  1 root root 0  五  9 11:39 enp2s0 -> ../../devices/pci0000:00/0000:00:1c.0/0000:02:00.0/net/enp2s0
lrwxrwxrwx  1 root root 0  五  9 11:39 enp3s0 -> ../../devices/pci0000:00/0000:00:1c.1/0000:03:00.0/net/enp3s0
lrwxrwxrwx  1 root root 0  五  9 11:39 lo -> ../../devices/virtual/net/lo
root@code-H11H4-IM:/home/code/ddd/old/hpeter/fintek/F81601/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
 - This value should be fine-tune by customer or following table, and the clock should set with half clock source (24MHz / 2 = 12Mhz)

```
root@code-H11H4-IM:/home/code# can-calc-bit-timing -c 12000000 sjal000
```

Bit timing parameters for sjal000 with 12.000000 MHz ref clock

nominal							real Bitrt	nom	real	SampP			
Bitrate	TQ[ns]	PrS	PhS1	PhS2	SJW	BRP	Bitrate	Error	SampP	SampP	Error	BTR0	BTR1
1000000	83	4	4	3	1	1	1000000	0.0%	75.0%	75.0%	0.0%	0x00	0x27
800000	83	5	6	3	1	1	800000	0.0%	80.0%	80.0%	0.0%	0x00	0x2a
500000	250	3	3	1	1	3	500000	0.0%	87.5%	87.5%	0.0%	0x02	0x05
250000	250	6	7	2	1	3	250000	0.0%	87.5%	87.5%	0.0%	0x02	0x1c
125000	500	6	7	2	1	6	125000	0.0%	87.5%	87.5%	0.0%	0x05	0x1c
100000	1250	3	3	1	1	15	100000	0.0%	87.5%	87.5%	0.0%	0x0e	0x05
50000	1250	6	7	2	1	15	50000	0.0%	87.5%	87.5%	0.0%	0x0e	0x1c
20000	3333	6	6	2	1	40	20000	0.0%	87.5%	86.6%	1.0%	0x27	0x1b
10000	5000	8	8	3	1	60	10000	0.0%	87.5%	85.0%	2.9%	0x3b	0x2f

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.

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

If the sample-point must be adjust, add “sample-point [custom value]” in step2.

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 00000000~1FFFFFFFh.

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/f81601a.ko
to entrust the driver and reboot.
```

Q4: How to make F81601A with maximum performance?

A4:

- Enable Intel VT-d in BIOS.
- Confirm the interrupt mode is MSIx2.

```
root@code-ms7c82:/home/code/ddd/old/hpeter/fintek/F81601-601A/driver# cat /proc/interrupts | grep can
132:      0          0      82266          0 IR-PCI-MSI 2621440-edge      can0
133:      0          0      82275 IR-PCI-MSI 2621441-edge      can1
root@code-ms7c82:/home/code/ddd/old/hpeter/fintek/F81601-601A/driver#
```

- Disable C-state power saving mode.

Q5: How to do IRQ assign for special CPU?

A5:

- a. Loading driver with parameter “auto_affinity=0” and using “dmesg” to find the F81601A two CANBUS IRQ. As following, the IRQ is 70/71.

```
[root@localhost driver_601a]# insmod f81601a.ko auto_affinity=0
[6909.107325] f81601a 0000:04:00.0: Fintek F81601 Driver version: v1.04-20250829
[6909.107333] f81601a 0000:04:00.0: auto_affinity mode: 0
[6909.107410] f81601a 0000:04:00.0: Using MMIO interface
[6909.107432] f81601a 0000:04:00.0: F81601 running with auto/external clock: 80Mhz, div: 1
[6909.107438] f81601a 0000:04:00.0: CAN/FD Not using TX delay ACKed
[6909.107856] f81601a 0000:04:00.0: Channel #1 at 0x00000000a4bbb854, irq 70 registered as can0
[6909.108056] f81601a 0000:04:00.0: Channel #2 at 0x00000000c5c5dfdb, irq 71 registered as can1
```

- b. View current MAX cpu number:

nproc

```
[root@localhost driver_601a]# nproc
16
```

- c. View current IRQ affinity via following command:

cat /proc/irq/70/smp_affinity

cat /proc/irq/71/smp_affinity

```
[root@localhost driver_601a]# cat /proc/irq/70/smp_affinity
0808
[root@localhost driver_601a]# cat /proc/irq/71/smp_affinity
0202
[root@localhost driver_601a]#
```

In this case, IRQ 70 will processed by CPU 3 & 11 (0808h => 0000 1000 0000 1000b) and IRQ 71 is CPU 1 & 9 (0202h => 0000 0010 0000 0010b).

- d. Assign IRQ to specific CPU. We'll try to fix IRQ70 to CPU 2/3 and IRQ71 to CPU 4/5 as following command:

echo 000c > /proc/irq/70/smp_affinity

echo 0030 > /proc/irq/71/smp_affinity

- e. After assignment, we must verify that the system has accepted the affinity.

```
[root@localhost driver_601a]# echo 000c > /proc/irq/70/smp_affinity
[root@localhost driver_601a]# echo 0030 > /proc/irq/71/smp_affinity
[root@localhost driver_601a]# cat /proc/irq/70/smp_affinity
000c
[root@localhost driver_601a]# cat /proc/irq/71/smp_affinity
0030
[root@localhost driver_601a]#
```

Q6. How to use udev & EEPROM to do network device rename?

A6.

1. Remove the existing module and installed rule
rmmod f81601a
rm -f /etc/udev/rules.d/99-fintek-f81601a-fix-name.rules
rm -f /run/udev/rules.d/99-fintek-f81601a-fix-name.rules
udevadm control --reload
udevadm settle
2. Rebuild the driver
make clean; make && make install
3. Select one rename method in `99-fintek-f81601a-fix-name.rules`
pcie_path: (Fix by PCIe bus)
can-fintek-0 / can-fintek-1
pcie_fix: (Fix by EEPROM data, the number 16/32 will depend on EEPROM)
can-ftk-fix-16 / can-ftk-fix-32
4. Install the rule and reload udev
install -m 0644 99-fintek-f81601a-fix-name.rules \
/etc/udev/rules.d/99-fintek-f81601a-fix-name.rules
udevadm control --reload
udevadm settle
5. Load the driver
modprobe can
modprobe f81601a
udevadm settle

Fix by PCIe bus

```
[ 3470.417286] f81601a 0000:01:00.0: F81601A running with auto/external clock: 80Mhz, div: 1
[ 3470.417291] f81601a 0000:01:00.0: CAN/FD Not using TX delay ACKed
[ 3470.552187] f81601a 0000:01:00.0: Channel #1 at 0x00000000ae9cc030, irq 41 registered as can0
[ 3470.552509] f81601a 0000:01:00.0: Channel #2 at 0x00000000f397a3e9, irq 42 registered as can1
[ 3470.556024] f81601a 0000:01:00.0 can-ftk-fix-36: renamed from can1
[ 3470.564000] f81601a 0000:01:00.0 can-ftk-fix-19: renamed from can0
```

Fix by EEPROM data

```
[ 3480.101420] f81601a 0000:01:00.0: F81601A running with auto/external clock: 80Mhz, div: 1
[ 3480.101424] f81601a 0000:01:00.0: CAN/FD Not using TX delay ACKed
[ 3480.243715] f81601a 0000:01:00.0: Channel #1 at 0x00000000ae9cc030, irq 41 registered as can0
[ 3480.243991] f81601a 0000:01:00.0: Channel #2 at 0x00000000f397a3e9, irq 42 registered as can1
[ 3480.246536] f81601a 0000:01:00.0 can-fintek-1: renamed from can1
[ 3480.249845] f81601a 0000:01:00.0 can-fintek-0: renamed from can0
```