

FINTEK

F81601

PCIe to 2 CANBUS

Driver Installation Guide

for Linux

v1.19

Mar 9, 2023

1 / 8

Revision History

Date	Version	Revision History
2018/6/4	v1.00	Initial version
2018/7/13	V1.01	Fix kernel 3.19 compile issue
2018/9/11	V1.03	Change IRQ mode from MSI to INTA
2019/3/21	V1.04	Parameterize to change IO/MEM, MSI/INTA
2019/6/5	V1.05	Fix kernel 3.15 compile issue
2019/8/12	V1.06	Fix S3/S4 resume will not functional
2019/11/27	V1.07	Document change only. Add change queue discipline command in 3. Configure CANBUS to avoid tx loss
2020/1/8	V1.08	Fix kernel 3.16 compile issue
2020/1/13	V1.09	Default using IO read/write mode Fix CANBUS read issue If your platform can't use IO instruction (like ARM/ARM64), Please change the define F81601_ACCESS_MEM_MODE in f81601.c to 1
2020/3/10	V1.10	Fix data overrun when idle into C-state with interrupt latency Add demo code "read_id.zip" for read card id
2020/5/21	V1.11	Add net-tools can-utils installation for "ifconfig" Fix for canconfig build issue in "config tools.zip" Fix build issue for kernel 4.4.24
2020/6/22	V1.12	Improve CANBUS read performance Fix build issue for kernel 3.10
2020/7/29	V1.13	Improve CANBUS read performance Fix overrun when bitrate = 250kbit
2020/8/27	V1.14	Fix not detect BUS off/restart state. Fix make install / depmod issue
2020/11/25	V1.15	Fix kernel 3.4 compile issue Fix CAN frame TX stability
2021/4/9	V1.16	Fix kernel 5.11 compile issue Fix kernel 3.10.0 (CentOS 7) MSI crash issue

2021/12/16	V1.17	Fix kernel 5.12/13 compile issue
2022/5/17	V1.17.1	Document & Makefile change. Driver is same with V1.17
2022/11/22	V1.18	Improve performance. Fix system hang when can port up with lots RX frame.
2023/3/9	V1.19	Improve RX frame stability. Change SJW setting from default 4 to user configure. Reduce error report. (controlled by parameter "more_err_report") Fix kernel v5.19 build issue. Add IRQ affinity control. (parameter "auto_affinity")

1. Preliminary

This document is for Fintek F81601 PCIe to 2 CANBUS 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

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

1. ifconfig 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	Error	BTR0	BTR1					
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. ifconfig 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.

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-H11H4-IM: /home/code [113x33]
連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)
When incrementing the CAN data the data length code minimum is set to 1.
CAN IDs and data content are given and expected in hexadecimal values.

Examples:
cangen vcan0 -g 4 -I 42A -L 1 -D i -v -v      (fixed CAN ID and length, inc. data)
cangen vcan0 -e -L i -v -v -v              (generate EFF frames, incr. length)
cangen vcan0 -D 11223344DEADBEEF -L 8      (fixed CAN data payload and length)
cangen vcan0 -g 0 -i -x                    (full load test ignoring -ENOBUEFS)
cangen vcan0 -g 0 -p 10 -x                (full load test with polling, 10ms timeout)
cangen vcan0                               (my favourite default :)

root@code-H11H4-IM: /home/code/ddd/old/hpeter/fintek/F81601/driver# cangen can0 -n 4
root@code-H11H4-IM: /home/code/ddd/old/hpeter/fintek/F81601/driver#

-----
root@code-H11H4-IM: /home/code#
root@code-H11H4-IM: /home/code#
root@code-H11H4-IM: /home/code#
root@code-H11H4-IM: /home/code#
root@code-H11H4-IM: /home/code#
root@code-H11H4-IM: /home/code#
root@code-H11H4-IM: /home/code# candump can1
can1 3DE [1] 10
can1 241 [8] 15 89 14 08 20 89 1D 09
can1 54A [0]
can1 60E [8] 68 CB 4C 0A A3 15 A8 37

-----
May 9 11:50:06 code-H11H4-IM kernel: [11012.487261] sja1000_f81601 0000:01:00.0 can0: setting BTR0=0x01 BTR1=0x1
c
May 9 11:50:06 code-H11H4-IM kernel: [11012.490660] sja1000_f81601 0000:01:00.0 can1: setting BTR0=0x01 BTR1=0x1
c

[0] 0:bash* 1:bash- "code-H11H4-IM" 14:01 09- fi -18

```

```

root@code-H11H4-IM: /home/code [113x33]
連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)
<can_id>#<flags>{data} for CAN FD frames
<can_id> can have 3 (SFF) or 8 (EFF) hex chars
{data} has 0..8 (0..64 CAN FD) ASCII hex-values (optionally separated by '.')
<flags> a single ASCII Hex value (0 .. F) which defines canfd_frame.flags

e.g. 5A1#11,2233,44556677,88 / 123#DEADBEEF / 5AA# / 123##1 / 213##311
1F334455#1122334455667788 / 123#R for remote transmission request.

root@code-H11H4-IM: /home/code# cansend can0 123#R
root@code-H11H4-IM: /home/code#

-----
root@code-H11H4-IM: /home/code# candump can1
can1 123 [8] 11 22 33 44 55 66 77 88
can1 123ABCDE [8] 11 22 33 44 55 66 77 88
can1 123 [0] remote request

-----
May 9 14:20:00 code-H11H4-IM kernel: [ 367.753899] sja1000_f81601 0000:01:00.0 can0: bit-timing not yet defined
May 9 14:20:16 code-H11H4-IM kernel: [ 383.815728] sja1000_f81601 0000:01:00.0 can0: setting BTR0=0x01 BTR1=0x1
c
May 9 14:20:16 code-H11H4-IM kernel: [ 383.818759] sja1000_f81601 0000:01:00.0 can1: setting BTR0=0x01 BTR1=0x1
c
May 9 14:20:38 code-H11H4-IM kernel: [ 405.567568] can: controller area network core (rev 20170425 abi 9)
May 9 14:20:38 code-H11H4-IM kernel: [ 405.567596] NET: Registered protocol family 29
May 9 14:20:38 code-H11H4-IM kernel: [ 405.575128] can: raw protocol (rev 20170425)

[0] 0:bash- 1:bash* "code-H11H4-IM" 14:22 09- fi -18

```

Fintek The information contained herein is the exclusive proprietary information of Feature Integration Technology Inc., is provided for customer's internal use only, and shall not be disclosed, distributed to any third party or reproduced in whole or in part by any third party without the prior written permission of Feature Integration Technology Inc. 本資料為精拓科技股份有限公司之智慧財產權，僅供客戶內部使用；非經精拓科技股份有限公司事前書面授權許可，不得透露、散佈予任何第三人或使任何第三人使用本資料之全部或一部，或將之複印、複製或轉變成其它任何形式使用。

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>

5. Read F81601 card ID

We can using the read_id.c to read F81601 card ID

1. unzip read_id.zip
2. cd read_id
3. make
4. ./read_id can0

```
root@code-desktop:/home/code/ddd/old/hpeter/fintek/F81601/app/read_id# ./read_id
usage:
    ./read_id <canbus>
e.g.
    ./read_id can0
root@code-desktop:/home/code/ddd/old/hpeter/fintek/F81601/app/read_id# ./read_id can0
can0 id: 7
root@code-desktop:/home/code/ddd/old/hpeter/fintek/F81601/app/read_id#
```

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/f81601.ko
to entrust the driver and reboot.
```

Q4: How to make F81601 with maximum performance?

A4:

- a. Enable Intel VT-d in BIOS.
- b. 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#
```

- c. Disable C-state power saving mode.