

FINTEK

PCIE to UART Driver Installation Guide for Linux

v1.0.14

Apr 11, 2024

F81504_Driver.odt

1 / 12

4/11/24

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。

Datasheet Revision History

Date	Version	Revision History
2014/12/25	1.0	Initial Version.
2014/12/26	1.0.1	Add compile kernel with Fedora Distribution.
2015/1/28	1.0.2	Add permanent installation with F81504 Module.
2015/4/2	1.0.3	Separate documents to match different kernel ver, this is for for Kernel 2.6.38 ~ 3.12. Fix port failed after S3(Suspend To RAM).
2015/5/18	1.0.4	Add support with High baud rate Add support with GPIO Mode Add 3.13~3.18 hot patch
2015/5/27	1.04.1	Fix compatible for lower than kernel 3.0 (CentOS6) Add Debian 7 install instructions
2015/8/14	1.0.4.2	Fix GPIO setting for F81508 Add RS485 API with kernel 3.19+ Add app to change RS485 Mode below kernel 3.19 Modify boot script
2016/3/23	1.0.4.3	Fix compile error on 2.6.36- kernel due to 2.6.36- not support high-baudrate.
2016/4/26	1.0.4.4	Add support for customer disable port with external EEPROM. Fix compile error on 3.4 kernel due to assign set_termios() with wrong data type.
2016/10/3	1.0.4.5	Add support for customer disable port with H/W pull low DSR/DCD/CTS/RI/RX Add CentOS 7 Linux documents for installation
2017/7/27	1.0.4.6	Fix some compatibility
2017/12/21	1.0.4.7	Fix some compatibility about PCIe Bridges
2018/7/18	1.0.4.8	Fix compatibility for kernel 3.1/3.6
2020/5/4	1.0.4.9	Fix support for imx6 kernel 4.9 issue



		Add fix com installation guide
2021/9/8	1.0.5	Fix for kernel 5.8 complie issue
2021/12/20	1.0.6	Fix for kernel 5.13 compile issue Add support for F81504/F75111 mode switch EVB.
2021/12/20	1.0.6.1	Document change for HW Fix transceiver mode
2022/5/17	1.0.6.2	Document & Makefile change only
2022/9/12	1.0.7	Fix com script change for Fintek / mainstream driver. Add MMIO access F75111 GPIO mode change.
2022/11/21	1.0.8	Fix H/W disable with F75111 will failed on gpio change. Fix H/W disable will report wrong port on boot.
2022/11/25	1.0.9	Fix H/W disable with F75111 will failed on gpio change. Fix H/W disable will report wrong port on boot.
2022/12/14	1.0.10	Add force GPIO mode via SERIRQ trap pin. Fix F75111 reset issue.
2022/12/15	1.0.11	Fix force GPIO mode via SERIRQ trap pin. Fix F75111 reset issue.
2022/12/29	1.0.11.1	Document change for installation with CentOS stream 8.
2023/7/21	1.0.12	Fix for kernel 6.0 / 6.1 compile issue
2023/7/25	1.0.13	Add non-standard baudrate 20kbits
2023/8/2	1.0.13.1	Fix makefile only. Driver not changed.
2024/4/11	1.0.14	Add workaround document for RHEL kernel 5.14 API type not compatible with mainstream kernel.

1. Driver installation

1. sudo su
2. Unzip the driver.zip
3. cd driver
4. Prepare the kernel tree & compiler tools for your distribution. (see appendix, Prepare Tools)
5. make clean ; make ; make install ; depmod
6. Boot script configure. (see appendix, Boot Configuration)
7. If customers need fix com port, see Fix COM script installation otherwise skip this.
8. Reboot and check if any new port added by “dmesg | grep ttyS”

```
root@code-desktop:/home/code/ddd/hpeter/fintek/1110/3.13- driver# dmesg | grep ttyS
[ 0.493174] 00:0a: ttyS0 at I/O 0x3f8 (irq = 4) is a 16550A
[ 0.513943] 00:0b: ttyS1 at I/O 0x2f8 (irq = 3) is a 16550A
[ 0.534706] 00:0c: ttyS2 at I/O 0x3e8 (irq = 10) is a 16550A
[ 0.555469] 00:0d: ttyS3 at I/O 0x2e8 (irq = 10) is a 16550A
[ 0.576239] 00:0e: ttyS4 at I/O 0x2d0 (irq = 11) is a 16550A
[ 0.597005] 00:0f: ttyS5 at I/O 0x2d8 (irq = 11) is a 16550A
[ 102.988045] 0000:01:00.0: ttyS6 at I/O 0xe000 (irq = 16) is a 16550A
[ 103.008588] 0000:01:00.0: ttyS7 at I/O 0xe008 (irq = 16) is a 16550A
[ 103.029241] 0000:01:00.0: ttyS8 at I/O 0xe010 (irq = 16) is a 16550A
[ 103.049840] 0000:01:00.0: ttyS9 at I/O 0xe018 (irq = 16) is a 16550A
```

2. Serial Port Mode Change

The Linux official RS485 API released with Kernel 3.19.0. If you are using Kernel 3.19.0 or newer kernel, It can be set by TIOCSR485/TIOCGRS485 ioctl. Otherwise we can control the mode with our demo app “set_mode”.

This

```
root@code:/home/code/ddd/hpeter/fintek/F81504-Series/change_mode# make
gcc -I.. -c -o set_mode.o set_mode.c
gcc set_mode.o -o set_mode
root@code:/home/code/ddd/hpeter/fintek/F81504-Series/change_mode# ./set_mode
./set_mode
./set_mode <port> <mode>

UART Mode table:
1 --> eModeRS232
2 --> eModeRS485 - DE#/RE (TX Enable with RTS Low)
3 --> eModeRS485_1 - DE/RE# (TX Enable with RTS High)

root@code:/home/code/ddd/hpeter/fintek/F81504-Series/change_mode# ./set_mode ttyS4 2
Target:/sys/class/tty/ttyS4/device/config
291c411
Target:/sys/class/tty/ttyS4/port
change to mode: eModeRS485 - DE#/RE (TX Enable with RTS Low)
root@code:/home/code/ddd/hpeter/fintek/F81504-Series/change_mode#
```

screen-shot is based on Ubuntu 14.04.2.

Section1: Compile App

Section2: App Parameter

Section2: Demo change to RS485 Mode with DE# (We should also setup transceiver correctly)

Transceiver mode change (on EVB)

We designed special EVB for F81504/Transceiver mode switch. It can be controlled by set_mode / set_gpio to change the mode for application (RS422/485/232/ ...).

```

root@code-ms7c82:/home/code
Fintek F81504 set gpio tools
Set mode
./set_gpio <port> <mode>
Query mode
./set_gpio <port>

Uart Mode table:
F81439(M0/M1/M2)
0: 000 - RS422
1: 001 - RS232
2: 010 - RS485 Low Active
3: 011 - RS485 High Active
4: 100 - RS422 with termination
5: 101 - Reserved
6: 110 - RS485 Low Active with termination
7: 111 - Shutdown

F81435(M0/M1/M2)
0: 000 - RS422 High Active
1: 001 - RS232
2: 010 - RS485 High Active
3: 011 - ccTALK
4: 100 - RS422 Low Active
5: 101 - RS485 Low Active (Auto Sensing)
6: 110 - RS485 Low Active
7: 111 - RS422 Low Active (Auto Sensing)

F81433(M0/M1/R_DIS)
1: 001 - RS232
2: 010 - RS422 with termination
3: 011 - RS422
4: 100 - RS485 Low Active with termination (Auto Sensing)
5: 101 - RS485 Low Active (Auto Sensing)
7: 111 - Shutdown
[0] 0: bash 1: bash 2: [tmux]*
"code-ms7c82" 06:46 15- 2月 -41

```

The set_mode / set_gpio must set correctly for using this EVB, the default mode is RS232. The following examples will use F81439 and ttyS7 for demo.

RS232:

```
./set_mode ttyS7 1
```

```
./set_gpio ttyS7 1
```

RS485:

```
./set_mode ttyS7 2
```

```
./set_gpio ttyS7 2
```

RS422:

```
set_mode: don't care.  
./set_gpio ttyS7 0
```

Query transceiver mode & type

```
./set_gpio <com port>
```

```
root@code-ms7c82:/home/code/ddd/old/hpeter/fintek/tmp/f81504/change_mode# ./set_gpio ttyS7  
Transceiver: F81439  
current mode: 1  
root@code-ms7c82:/home/code/ddd/old/hpeter/fintek/tmp/f81504/change_mode# █
```

H/W fix transceiver mode

Linux Driver will only detect F81504 Transceiver & H/W fix mode when driver load, reboot & S3/4 resume.
The detecting flow is as following:

1. Set the M0/M1/M2 with the com port as input mode.
2. Read the three pins value.
3. If the three pins are digital 1, this com port can be controlled by set_gpio, otherwise is H/W fix mode.

Appendix:

Prepare Tools:

If You are using Debian:

1. nano /etc/apt/sources.list
2. add '#' in the front of "deb cdrom:[Debian GNU/Linux 7.8.0 _Wheezy_ - Official i386 DVD Binary-1 20150110-13:32]/ wheezy contrib main"

```
# deb cdrom:[Debian GNU/Linux 7.8.0 _Wheezy_ - Official i386 DVD Binary-1 20150110-13:32]/ wheezy contrib main
#deb cdrom:[Debian GNU/Linux 7.8.0 _Wheezy_ - Official i386 DVD Binary-1 20150110-13:32]/ wheezy contrib main
deb http://ftp.tw.debian.org/debian/ wheezy main
deb-src http://ftp.tw.debian.org/debian/ wheezy main
```

3. CTRL-X & save
4. apt-get update
5. apt-get dist-upgrade
6. reboot
7. apt-get install gcc make kernel-package linux-headers-`uname -r` build-essential

If You are using Ubuntu:

1. apt-get update
2. apt-get dist-upgrade
3. reboot
4. apt-get install gcc make kernel-package

If You are using Fedora series (e.g. CentOS / Fedora 21):

1. yum update
2. reboot
3. yum install gcc make kernel-devel kernel-devel rpm-build

Fix COM script installation:

After the installation of driver, If customer need to fix com port index. Please follow the section.

We provide the fix com script "fix_com.sh" for the ttySx index to match desire IO port address.

For the example, the following script will make the serial port sequence:

ttyS1 => 0x3000

ttyS2 => 0x3008

...

ttyS8 => 0x3038

Customers need modify the script to fit the target machine first and install the script via following OS type.

```
customer_fix_com(){
# 客戶請自行修改對應的 ttySx <=> IO address
# 請記得 io address 都是大寫
# 也請記得要安裝 setserial

:;
1      2
set_serial_io "ttyS1" 0x3000
set_serial_io "ttyS2" 0x3008
set_serial_io "ttyS3" 0x3010
set_serial_io "ttyS4" 0x3018
set_serial_io "ttyS5" 0x3020
set_serial_io "ttyS6" 0x3028
set_serial_io "ttyS7" 0x3030
set_serial_io "ttyS8" 0x3038

set_serial_io "ttyS9" 0x4000
set_serial_io "ttyS10" 0x4008
set_serial_io "ttyS11" 0x4010
set_serial_io "ttyS12" 0x4018
set_serial_io "ttyS13" 0x4020
set_serial_io "ttyS14" 0x4028
set_serial_io "ttyS15" 0x4030
set_serial_io "ttyS16" 0x4038
}
```

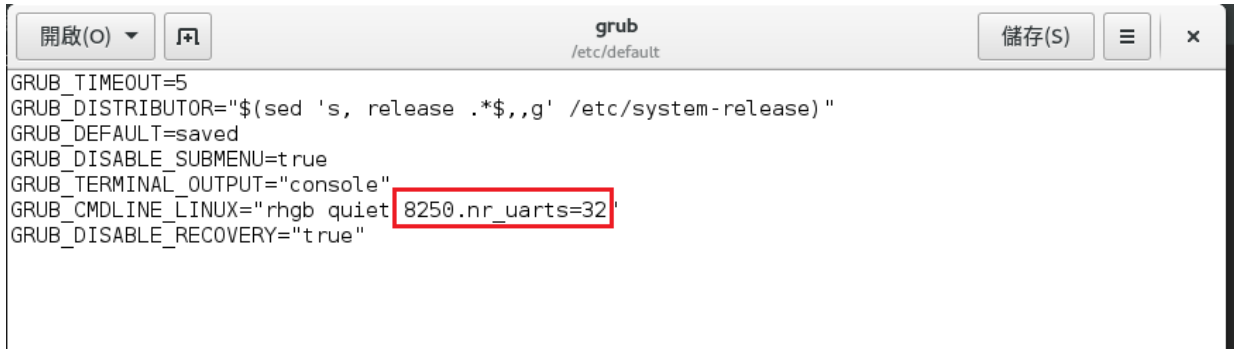

If you are using Ubuntu 18.04:

1. apt-get install setserial
2. cp fix_com.sh /etc/init.d/
3. chmod 755 /etc/init.d/fix_com.sh
4. cp f81504_fix_com.service /etc/systemd/system/
5. chmod 644 /etc/systemd/system/f81504_fix_com.service
6. systemctl daemon-reload
7. systemctl enable f81504_fix_com
8. systemctl start f81504_fix_com

Boot Configuration

Fedora 21 Configurations / CentOS 7 / CentOS stream 8

1. sudo su
2. gedit /etc/default/grub
3. add command "8250.nr_uaarts=32" to "GRUB_CMDLINE_LINUX" like:



```

GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rhgb quiet 8250.nr_uaarts=32"
GRUB_DISABLE_RECOVERY="true"

```

4. CentOS 7: grub2-mkconfig > /boot/grub2/grub.cfg
CentOS stream 8: grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg
5. reboot
6. dmesg | grep ttyS

```

[code@localhost ~]$ dmesg | grep ttyS
[ 0.445584] 00:06: ttyS0 at I/O 0x3f8 (irq = 4, base_baud = 115200) is a 16550A
[ 0.467713] 0000:02:00.0: ttyS4 at I/O 0xd000 (irq = 16, base_baud = 115200) is a 16550A
[ 0.488809] 0000:02:00.0: ttyS5 at I/O 0xd008 (irq = 16, base_baud = 115200) is a 16550A
[ 0.509899] 0000:02:00.0: ttyS6 at I/O 0xd010 (irq = 16, base_baud = 115200) is a 16550A
[ 0.530986] 0000:02:00.0: ttyS7 at I/O 0xd018 (irq = 16, base_baud = 115200) is a 16550A
[ 0.552177] 0000:06:00.0: ttyS8 at I/O 0xb000 (irq = 18, base_baud = 115200) is a 16550A
[ 0.573278] 0000:06:00.0: ttyS9 at I/O 0xb008 (irq = 18, base_baud = 115200) is a 16550A
[ 0.594377] 0000:06:00.0: ttyS10 at I/O 0xb010 (irq = 18, base_baud = 115200) is a 16550A
[ 0.615463] 0000:06:00.0: ttyS11 at I/O 0xb018 (irq = 18, base_baud = 115200) is a 16550A
[ 0.636550] 0000:06:00.0: ttyS12 at I/O 0xb020 (irq = 18, base_baud = 115200) is a 16550A
[ 0.657641] 0000:06:00.0: ttyS13 at I/O 0xb028 (irq = 18, base_baud = 115200) is a 16550A
[ 0.678728] 0000:06:00.0: ttyS14 at I/O 0xb030 (irq = 18, base_baud = 115200) is a 16550A
[ 0.699833] 0000:06:00.0: ttyS15 at I/O 0xb038 (irq = 18, base_baud = 115200) is a 16550A
[ 0.720936] 0000:06:00.0: ttyS16 at I/O 0xb040 (irq = 18, base_baud = 115200) is a 16550A
[ 0.742037] 0000:06:00.0: ttyS17 at I/O 0xb048 (irq = 18, base_baud = 115200) is a 16550A
[ 0.763133] 0000:06:00.0: ttyS18 at I/O 0xb050 (irq = 18, base_baud = 115200) is a 16550A
[ 0.784234] 0000:06:00.0: ttyS19 at I/O 0xb058 (irq = 18, base_baud = 115200) is a 16550A
[code@localhost ~]$ sudo su
[sudo] password for code:
[root@localhost code]# uname -a
Linux localhost.localdomain 3.19.7-200.fc21.x86_64 #1 SMP Thu May 7 22:00:21 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux

```

CentOS 6 Configurations

1. su
2. gedit /boot/grub/grub.conf

```
*grub.conf x
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#         all kernel and initrd paths are relative to /boot/, eg.
#         root (hd0,0)
#         kernel /vmlinuz-version ro root=/dev/mapper/vg_livedvd-lv_root
#         initrd /initrd-[generic-]version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.32-504.16.2.el6.i686)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-504.16.2.el6.i686 ro root=/dev/mapper/vg_livedvd-lv_root rd_NO_LUKS LANG=en_US.UTF-8
rd_LVM LV=vg_livedvd/lv_root rd_NO_MD SYSFONT=latacyrheb-sun16 crashkernel=auto rd_LVM_LV=vg_livedvd/lv_swap
KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet 8250.nr_uaarts=32
    initrd /initramfs-2.6.32-504.16.2.el6.i686.img
title CentOS (2.6.32-279.el6.i686)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-279.el6.i686 ro root=/dev/mapper/vg_livedvd-lv_root rd_NO_LUKS LANG=en_US.UTF-8
rd_LVM LV=vg_livedvd/lv_root rd_NO_MD SYSFONT=latacyrheb-sun16 crashkernel=auto rd_LVM_LV=vg_livedvd/lv_swap
KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet 8250.nr_uaarts=32
    initrd /initramfs-2.6.32-279.el6.i686.img
```

3. Add text "8250.nr_uaarts=32" to kernel option.
4. Save
5. cp /usr/share/dbus-1/system-services/org.freedesktop.ModemManager.service
/usr/share/dbus-1/system-services/org.freedesktop.ModemManager.service.bak
6. Reboot

Ubuntu 18.04 / Debian 7 or later Configurations

1. su
2. nano /etc/default/grub
3. add 8250.nr_uaarts=32 initcall_blacklist=serial_pci_driver_init to
"GRUB_CMDLINE_LINUX_DEFAULT"

```
GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=menu
GRUB_TIMEOUT=20
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash 8250.nr_uaarts=32 initcall_blacklist=serial_pci_driver_init"
#GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""
```

4. update-grub
5. reboot

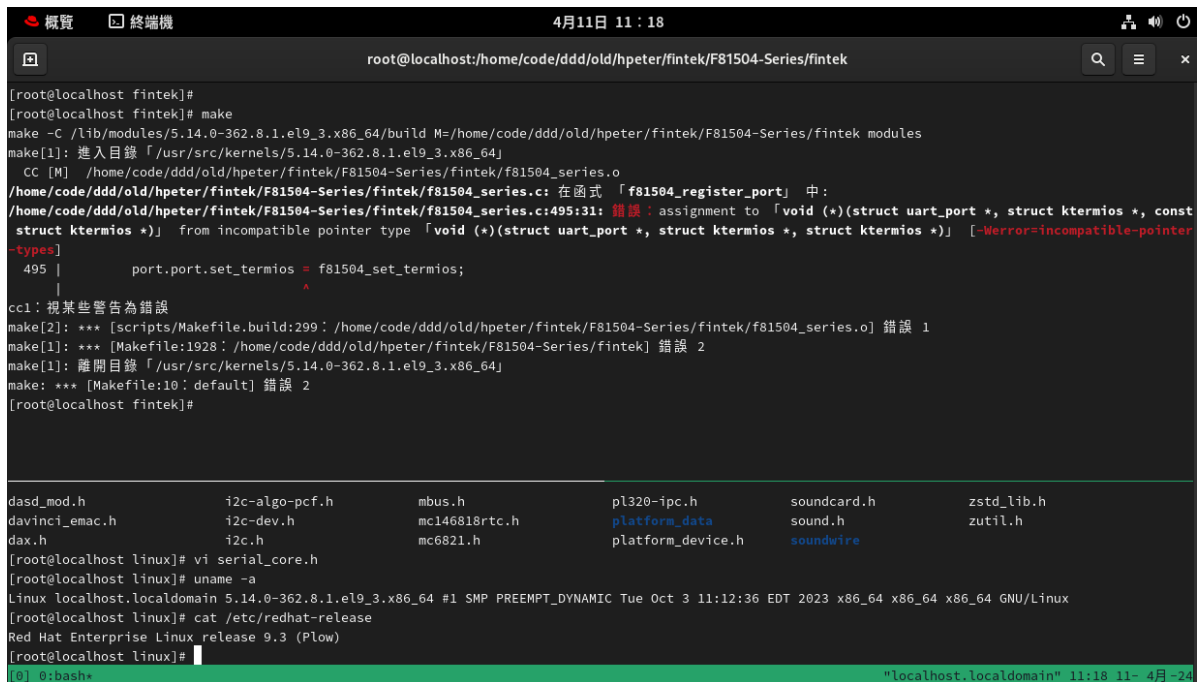
Q&A:

Q1: Can't load driver when system reboot with Kylin (銀河麒麟).

A1: Run:

sudo kysec_set -n exectl -v original /lib/modules/`uname -r`/updates/f81504_series.ko
to entrust the driver and reboot.

Q2: F81504 will build failed on RHEL kernel 5.14.



```

[root@localhost fintek]# make
make -C /lib/modules/5.14.0-362.8.1.el9_3.x86_64/build M=/home/code/ddd/old/hpeter/fintek/F81504-Series/fintek modules
make[1]: 進入目錄「/usr/src/kernels/5.14.0-362.8.1.el9_3.x86_64」
CC [M] /home/code/ddd/old/hpeter/fintek/F81504-Series/fintek/f81504_series.o
/home/code/ddd/old/hpeter/fintek/F81504-Series/fintek/f81504_series.c: 在函式「f81504_register_port」中:
/home/code/ddd/old/hpeter/fintek/F81504-Series/fintek/f81504_series.c:495:31: 錯誤: assignment to 「void (*)(struct uart_port *, struct ktermios *, const struct ktermios *)」 from incompatible pointer type 「void (*)(struct uart_port *, struct ktermios *, struct ktermios *)」 [-Werror=incompatible-pointer-types]
495 |         port.port.set_termios = f81504_set_termios;
    |                               ^
cc1: 視某些警告為錯誤
make[2]: *** [scripts/Makefile.build:299: /home/code/ddd/old/hpeter/fintek/F81504-Series/fintek/f81504_series.o] 錯誤 1
make[1]: *** [Makefile:1928: /home/code/ddd/old/hpeter/fintek/F81504-Series/fintek] 錯誤 2
make[1]: 離開目錄「/usr/src/kernels/5.14.0-362.8.1.el9_3.x86_64」
make: *** [Makefile:10: default] 錯誤 2
[root@localhost fintek]#

dasd_mod.h      i2c-algo-pcf.h      mbus.h          pl320-ipc.h      soundcard.h      zstd_lib.h
davinci_emac.h  i2c-dev.h          mc146818rtc.h   platform_data    sound.h          zutil.h
dax.h           i2c.h              mc6821.h        platform_device.h soundwire

[root@localhost linux]# vi serial_core.h
[root@localhost linux]# uname -a
Linux localhost.localdomain 5.14.0-362.8.1.el9_3.x86_64 #1 SMP PREEMPT_DYNAMIC Tue Oct 3 11:12:36 EDT 2023 x86_64 x86_64 x86_64 GNU/Linux
[root@localhost linux]# cat /etc/redhat-release
Red Hat Enterprise Linux release 9.3 (Plow)
[root@localhost linux]#
[0] 0:bash$

```

A2: It's due to the RHEL kernel API not the same with mainstream's. Customer must modify f81504_serial.c near line 290, **change the #if 1 to #if 0 and re-build driver.**

```

284: #if LINUX_VERSION_CODE >= KERNEL_VERSION(2, 6, 36)
285: #if LINUX_VERSION_CODE >= KERNEL_VERSION(6, 1, 0)
286: void f81504_set_termios(struct uart_port *port, struct ktermios *termios,
287:                        const struct ktermios *old)
288: #else
289: #if 1
290: void f81504_set_termios(struct uart_port *port, struct ktermios *termios,
291:                        struct ktermios *old)
292: #else
293: // for centos stream kernel 5.14.0-390-el9
294: void f81504_set_termios(struct uart_port *port, struct ktermios *termios,
295:                        const struct ktermios *old)
296: #endif
297: #endif
298: {
299:     struct pci_dev *dev = container_of(port->dev, struct pci_dev, dev);
300:     struct f81504_priv *priv = pci_get_drvdata(dev);
301:     unsigned int baud = tty_termios_baud_rate(termios);

```