

FINTEK

PCIE to UART Driver Installation Guide for Linux

v1.04.8

Jul 18, 2018

F81504_Driver.odt

1 / 9

10/9/18

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。

Datasheet Revision History

Date	Version	Revision History
2014/12/25	1.0	Initial Version.
2014/12/26	1.01	Add compile kernel with Fedora Distribution.
2015/1/28	1.02	Add permanent installation with F81504 Module.
2015/4/2	1.03	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	1.04	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	1.04.2	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	1.04.3	Fix compile error on 2.6.36- kernel due to 2.6.36- not support high-baudrate.
2016/4/26	1.04.4	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	1.04.5	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	1.04.6	Fix some compatibility
2017/12/21	1.04.7	Fix some compatibility about PCIe Bridges



2018/7/18	1.04.8	Fix compatibility for kernel 3.1/3.6
-----------	--------	--------------------------------------

1. Driver installation

1. Unzip the driver.zip
2. cd driver
3. Prepare the kernel tree & compiler tools for your distribution. (see appendix, Prepare Tools)
4. make clean ; make ; make install ; depmod
5. dmesg | grep 8250 | grep ports
If the output of ports is less than 32. Please following appendix, Boot Configuration.
6. Boot script configure. (see appendix, Boot Script)
7. Reboot and check if any new port added by “dmesg | grep ttyS”, if you cant see any new add ports, please following the Troubleshooting A1 then insmod driver again.

```
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 TIOCSRS485/TIOCGRS485 ioctl. Otherwise we can control the mode with our demo app “set_mode”.

```
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#
```

This 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)

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

Boot Script Modify:

If You are using Debian (Kernel >= 3.13):

1. cp F81504-boot-script.sh /etc/init.d
2. chmod +x /etc/init.d/F81504-boot-script.sh
3. update-rc.d F81504-boot-script.sh defaults 99

If you are using Ubuntu (systemd, tested with 14.04.2):

1. cp F81504-boot-script.sh /etc/init.d/
2. chmod +x /etc/init.d/F81504-boot-script.sh
3. vi /etc/init/F81504-patch.conf
add texts below

description "F81504 Serial Hotpatch"

start on runlevel [2345]

stop on runlevel [!2345]

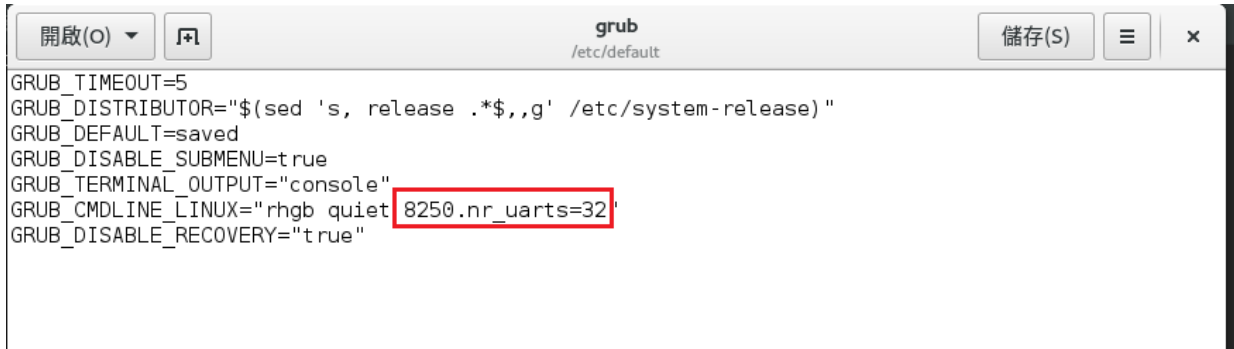
exec /etc/init.d/F81504-boot-script.sh

4. save and reboot

Boot Configuration

Fedora 21 Configurations / CentOS 7

1. `sudo su`
2. `gedit /etc/default/grub`
3. add command "8250.nr_uares=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_uares=32"
GRUB_DISABLE_RECOVERY="true"

```

4. `grub2-mkconfig > /boot/grub2/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
3. Add text "8250.nr_uares=32" to kernel option.

```
*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_uares=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_uares=32
    initrd /initramfs-2.6.32-279.el6.i686.img
```

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

Debian 7 Configurations

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

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet 8250.nr_uares=32"
GRUB_CMDLINE_LINUX=""
```

4. update-grub
5. reboot