GPG-4141-050033 Driver Rev 9.0 Installation on RedHawk 6.0-8.4 for Interface Serial board

Release Notes

September 7, 2022

1. Introduction:

This document assists the user in installing the Interface Corporation's ***GPG-4141*** driver and test programs on a RedHawk 6.0.x,6.3.x,6.5.x,7.0.x,7.2.x,7.3.x,7.5.x,8.0.x,8.2.x,8.4.x for use with the Interface Corporation board.

2. Requirements:

The PCI/LPC/PEX/CTP/CPZ-4xxxx, an EIA RS-232(TIA/EIA-232)/RS-485 (TIA/EIA-485)/RS-422 (TIA/EIA-422) compliant asynchronous serial communications product for PCI-based computers. It must be physically installed in the system.

- RedHawk Revision 6.0,6.3,6.5,7.0,7.2,7.3,7.5,8.0,8.2,8.4
- Interface Serial board installed

  PCI

  PCI-4141, PCI-4141P, PCI-4141PE, PCI-4142, PCI-4142P, PCI-4142PE
  PCI-4144, PCI-4145, PCI-4146, PCI-4147, PCI-4148C, PCI-4149C
  PCI-4150, PCI-4155, PCI-4161, PCI-4646
  PCI-420108Q, PCI-420116Q, PCI-420208Q, PCI-420216Q
  PCI-466102, PCI-466102P, PCI-466120, PCI-466120P
  PCI-466104, PCI-466104A, PCI-466104P, PCI-466104PA
  PCI-466140, PCI-466140A, PCI-466140P, PCI-466140PA
  PCI-466108, PCI-466180, PCI-466101, PCI-466130, PCI-466110

  Low Profile PCI

  LPC-400111, LPC-466102, LPC-466104, LPC-466120, LPC-466140

  PCI Express

  PEX-400111, PEX-466102, PEX-466104, PEX-466120, PEX-466140

  CompactPCI

  CTP-4141, CTP-4141P, CTP-4142, CTP-4142P, CTP-4144, CTP-4145
  CTP-4146, CTP-4147, CTP-4148, CTP-4149, CTP-466102, CTP-466120
  CTP-420108Q, CTP-420116Q, CTP-420208Q, CTP-420216Q
  CPZ-4141, CPZ-4141P, CPZ-4142, CPZ-4142P, CPZ-4144, CPZ-4145
  CPZ-4146, CPZ-4147, CPZ-4148, CPZ-4149
  CPZ-420108Q, CPZ-420116Q, CPZ-420208Q, CPZ-420216Q
  CPZ-466102, CPZ-466102P, CPZ-466120, CPZ-466120P
  CPZ-466104, CPZ-466104A, CPZ-466104P, CPZ-466104PA
  CPZ-466140, CPZ-466140A, CPZ-466140P, CPZ-466140PA

3. Installation:

The ***GPG-4141*** driver is designed to support IRQ sharing. If this devices IRQ is being shared by another device then this driver's performance could be compromised. Hence, as far as possible, move this board into a PCI slot whose IRQ is not being shared with other devices. A ***'lspci -v'*** command can be used to determine the IRQs of various devices in the system.

The ***GPG-4141*** driver is supplied in a RPM format on a CDROM/DVD. It is a dynamically loadable driver that must be loaded with the ***modprobe cp4141 and/or cp4161*** command once it has been installed. It can be unloaded by issuing the ***modprobe –r cp4141 and/or cp4161*** command.

To extract the driver from a CDROM, typical command is as follows:

For RPM package

```
> === as root ===
> mount /dev/sr0 /mnt/dvd   (an entry must exist in /etc/fstab - most likely, mount point is
                             /mnt/dvd)
> cd /mnt/dvd
> rpm -ivh GPG-4141-050033-9.i686.rpm (install the package)
```

or

```
> rpm -ivh GPG-4141-050033-9.x86_64.rpm (install the package)

                ########################################### [100%]
   1:GPG-4141   ########################################### [100%]
Please wait a minute.
Now extract and apply Interface gpg4141 package.
patching file install
Now extract and apply RedHawk patchs
patching file makettynode.c
patching file makettynode.c
patching file dpg0101.c
patching file com_main.c
patching file makeinftbl.c
patching file comex_main.c
patching file makeinftbl.c
Create Standard kernel Drivers..................................Done.
Create Debug kernel Drivers.....................................Done.
Create Trace kernel Drivers.....................................Done.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Load Drivers
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@cp4141 info:5.00.33.00
cp4161 info:5.00.32.00
ttyG0: PEX-466120(bid=0h) CH1  [9600bps] tx:0 rx:0
ttyG1: PEX-466120(bid=0h) CH2  [9600bps] tx:0 rx:0

All the source for this product has been installed.
To build and install the objects manually:
the following steps. Then:
To build the driver:

cd /usr/local/CNC/drivers/gpg4141
./install_redhawk

> umount /mnt/dvd
```

For DEB package

```
> === as root ===
# mount /dev/sr0 /mnt/dvd   (an entry must exist in /etc/fstab - most likely, mount point is
                             /mnt/dvd)
# cd /mnt/dvd
# apt install gpg4141-050033-9.deb
:
Please wait a minute.
Now extract and apply Interface gpg4141 package.
patching file install
Now extract and apply RedHawk patchs
patching file makettynode.c
patching file makettynode.c
patching file dpg0101.c
patching file com_main.c
patching file makeinftbl.c
patching file comex_main.c
patching file makeinftbl.c
Create Standard kernel Drivers..................................Done.
Create Debug kernel Drivers.....................................Done.
Create Trace kernel Drivers.....................................Done.

All the source for this product has been installed.
To build and install the objects manually:
```

```
the following steps. Then:
To build the driver:

cd /usr/local/CNC/drivers/gpg4141
./install_redhawk
> umount /mnt/dvd
```

The ***GPG-4141*** driver files will be installed into the ***/usr/src/interface*** directory from the DVD drive.

If you will need to rebuild the driver:

```
> === log in as root ===
> cd /usr/src/interface
> install_redhawk (build the driver & install the driver software and sample program)
```

NOTE!! If the ***make*** fails with some module version related error, then you will need to follow the directions *(see below)* ***"Building driver on a currently running RedHawk kernel".*** Once done, you will then need to re-make the driver as described above.

Once the driver is installed, you will need to ***/etc/init.d/ccrt_gpg4141*** it before any access to the PCI/LPC-4661xx board can be made. At this time, make sure that the hardware is physically installed in the machine, otherwise the load will fail.

```
> === log in as either user or root ===
> lspci -d 1147:1235 -vv
06:01.0 Communication controller: Interface Corp Device 1235 (rev 01)
        Subsystem: Interface Corp Device 2082
        Control: I/O- Mem+ BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR-
FastB2B- DisINTx-
        Status: Cap- 66MHz- UDF- FastB2B- ParErr- DEVSEL=slow >TAbort- <TAbort- <MAbort- >SERR-
<PERR- INTx-
        Interrupt: pin A routed to IRQ 48
        Region 0: Memory at fb202000 (32-bit, non-prefetchable) [size=64]
        Region 1: Memory at fb201000 (32-bit, non-prefetchable) [size=64]
        Region 2: Memory at fb200000 (32-bit, non-prefetchable) [size=16]
```

If the above device is not displayed by the ***lspci*** command, the device has not been properly installed in the system. Make sure that the device has been correctly installed and running prior to proceeding with the next step.

Once the driver and the hardware have been successfully installed, the driver can be loaded into the running kernel:

```
> === as root ===
> /etc/init.d/ccrt_gpg4141 restart (unload & load the GPG-4141 driver)
> lsmod |grep cp41            (this command will display the GPG-4141 driver)
> cp4161                145159  0
> cp4141                 99640  0

> cat /proc/tty/driver/cp4161
> cp4161 info:4.20.24.00
> ttyG0: PCI/LPC-466102(P)(bid=0h) CH1  [9600bps, FD] tx:0 rx:0
> ttyG1: PCI/LPC-466102(P)(bid=0h) CH2  [9600bps, FD] tx:0 rx:0
```

Build and run the driver test programs:

```
> === as user ===
> cd /usr/src/interface/PCI4661
> make                                        (build the test programs)
> You connect ch1 and ch2 using Cross Cable.  (CAB-17xx)
> ./com_loop -t /dev/ttyG0 -n /dev/ttyG1 -F   (Full Duplex loopback test)
```

```
> ./com_loop -t /dev/ttyG0 -n /dev/ttyG1 -H4      (Half Duplex 4-waire loopback test)
> You connect ch1 and ch2 using Straight Cable.   (CAB-18xx)
> ./com_loop -t /dev/ttyG0 -n /dev/ttyG1 -H2      (Half Duplex 2-waire loopback test)
> ./com_loop -h
                          usage: com_loop -t [Test device name] -n [Non
                          test device name] [-H2 -H4 or -F] [ -S -B
                          base_clock -D divisor ] [ -p test_pattern]

Examples
Default Speed Test(38400 bps)
                          ./com_loop -H2 -n /dev/ttyG0 -t /dev/ttyG1

Default Speed Test(38400 bps) and Fixed Pattern Test
                          ./com_loop -H2 -n /dev/ttyG0 -t /dev/ttyG1 -p
                          0x5555

Normal Speed Test
                          ./com_loop -H2 -n /dev/ttyG0 -t /dev/ttyG1 -S
                          -B
                          [50|75|110|134|150|200|300|600|1200|1800|2400
                          |4800|9600|19200|38400|57600|115200|230400|46
                          0800] -D 0

Special Speed Test(custom base_clock/divisor bps)
                          ./com_loop -H2 -n /dev/ttyG0 -t /dev/ttyG1 -S
                          -B
                          [2000000|1228800|921600|768000|512000|3686400
                          |3072000] -D n
```

4. Removal of the Package.

The ***GPG-4141*** driver is a dynamically loadable driver that can be unloaded as follows:

For RPM package

```
> === as root ===
> rpm -e GPG-4141-050033-7.i686 (remove the package)
or
> rpm -e GPG-4141-050033-7.x86_64 (remove the package)
```

For DEB package

```
> === as root ===
>  apt purge gpg4141 (remove the package)
```

NOTE!! If any changes have been made to the driver package, they need to be backed up prior to removing /usr/src/interface directory else all changes will be lost.

5. Notes.

1. You can control the state at startup with the following description.

/etc/interface/ttyG*duplex.cfg

half2w or half4w or half or full

and/or

```
  ioctl (fd_b, CP4141_SET_DUPLEX_MODE, & DuplexMode);
```

So, the connection is controlled by software, and

DuplexMode is

2-wire half-duplex	CP4141_HALF_DUPLEX_2W;
4-wire half-duplex	CP4141_HALF_DUPLEX_4W;
Full Duplex	CP4141_FULL_DUPLEX;

Can be selected.

2. A special speed uses the following API
Set the speed to B38400 and use iocrl().

Please refer to com_loop.c for details

```
struct serial_struct {
        int     type;
        int     line;
        unsigned int    port;
        int     irq;
        int     flags;
        int     xmit_fifo_size;
        int     custom_divisor;
        int     baud_base;
        unsigned short  close_delay;
        char    io_type;
        char    reserved_char[1];
        int     hub6;
        unsigned short  closing_wait; /* time to wait before closing */
        unsigned short  closing_wait2; /* no longer used... */
        unsigned char   *iomem_base;
        unsigned short  iomem_reg_shift;
        unsigned int    port_high;
        unsigned long   iomap_base;     /* cookie passed into ioremap */
};

ret= ioctl(Handle,TIOCGSERIAL,&serial);
/*
      set 2Mbps (See table below)
      CAUTION: The set B38400 values remains until resetting.
       (serial.baud_base, serial.custom_divisor, serial.flags)
*/
serial.baud_base=2000000;
serial.custom_divisor=1;
serial.flags |= ASYNC_SPD_CUST;
ret= ioctl(Handle,TIOCSSERIAL,&serial);
```

Special Speed Table

Target Speed	termios.c_cflag	serial.flags	serial.baud_base	serial.custom_divisor
57600	B38400&(CBAUD|CBAUDEX)	ASYNC_SPD_HI	Not Use	Not Use
115200		ASYNC_SPD_VHI		
230400		ASYNC_SPD_SHI		
460800		ASYNC_SPD_WARP		
2000000	B38400&(CBAUD|CBAUDEX)	ASYNC_SPD_CUST	2000000	1
1000000				2
500000				4
200000				10
100000				20
:				:
305.18				66535
1228800			1228800	1
614400				2
:				:
921600			921600	1
460800				2
:				:
768000			768000	1
384000				2
:				:
512000			512000	1
256000				2
:				:
3686400			3686400	1
1843200				2
				:
3072000			3072000	1
1536000				2
				:

6. Resolved Issues.

Rel2 :

When executing ccrt_gpg_4141, it fails with modprobe.

Rel3 :

It is not reflected unless special speed setting is done twice
(Fixed the driver.)

There is no program to test special speed setting
(The com_loop test program has been updated.)

Rel4:

Support RedHawk7.2

Rel5:

Support RedHawk7.5

Rel6:

Support RedHawk8.0

In 4.14 or later, expanding a data segment results in "Invalid address limit on user-mode return" error and insmod/modprobe error because SIGKILL is sent
This feature uses the /root/mybasic/gartbasic or
/etc/interface/"cp4161_ttyname"+"minor"duplex.cfg file
It is a function to define the default port status, but it can be replaced by the option specification at modprobe/insmod below.
cp4161_duplexmode=half2w
cp4161_duplexmode=half4w
cp4161_duplexmode=half

Rel7:

Support RedHawk8.2

Rel8:

Support RedHawk8.4..

Rel9:

Fixed a bug that caused a PANIC when the device was mapped to a 64-bit memory address space.

The cause was in the original device driver's code that read the map address from the PCI configuration register.

For this reason, it was PANIC because it was not possible to access the registers allocated above the 32-bit address space

Support DEB(Ubuntu18.04/20.04) Package.
Support aarch64(ARM64) Architecture.
XAVIER's PCIexpress bus cards may have some problems.
And it's using a legacy bus bridge, and two things happen.

No IRQ assigned.
The PCIexpress bus board is Disabled.

```
# lspci -vvv -n
0005:02:0c.0 ff00: 1147:09d3 (rev 01)
    Subsystem: 1147:2c80
    Control: I/O- Mem- BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
    Status: Cap- 66MHz- UDF- FastB2B- ParErr- DEVSEL=slow >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-
    Interrupt: pin A routed to IRQ 0
    Region 0: Memory at 1f40000000 (32-bit, non-prefetchable) [disabled] [size=32]
```

This issue is resolved by making changes to the card's device driver.
IRQs must be hard-coded.

Its value is 39 and must be assigned to pdev->irq in PCI configuration registers and PCI device structures.
Then you have to manipulate the PCI configuration register to enable the PCIexpress bus.

```
#ifdef __aarch64__
   temp_base_addr = pci_resource_start(pdev, PCI_BASE_NUMBER);
   temp_addr_size = ((pci_resource_end(pdev, PCI_BASE_NUMBER)) - temp_base_addr);
      request_region ( temp_base_addr, temp_addr_size, MODULE_NAME);
      if (pdev->resource[ PCI_BASE_NUMBER].flags & IORESOURCE_IO)
      {
                unsigned short cmd;
                pci_read_config_word(pdev, PCI_COMMAND, &cmd);
                pci_write_config_word(pdev, PCI_COMMAND, cmd | PCI_COMMAND_IO);
      }
      if (pdev->resource[ PCI_BASE_NUMBER].flags & IORESOURCE_MEM)
      {
                unsigned short cmd;
                pci_read_config_word(pdev, PCI_COMMAND, &cmd);
                pci_write_config_word(pdev, PCI_COMMAND, cmd | PCI_COMMAND_MEMORY);
      }
      irq=39;
      request_irq (irq,(void*) irq_handler、,IRQF_SHARED, MODULE_NAME, private_struct))   ;
      pci_write_config_byte(pdev, PCI_INTERRUPT_LINE, irq);
      pdev->irq=irq;
#endif
```

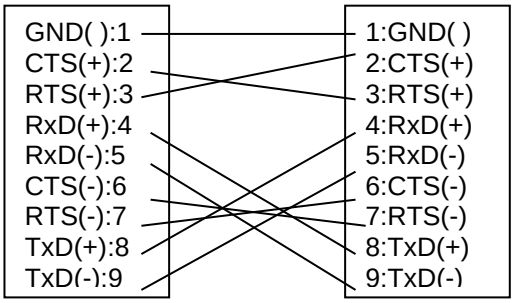
7. RS-422/485 Pin Assignment.

Signal Name		Direction	9 Pin D-sub Name		Description
T(A)	T-	OUT/(IN : Half Duplex)	TxD-	9	Send Data-
C(A)	C-	OUT	RTS-	7	Control Out-
R(A)	R-	IN	RxD-	5	Receive Data-
I(A)	I-	IN	CTS-	6	Control In-
SG	SG	-		1	Signal Graund
T(B)	T+	OUT/(IN : Half Duplex)	TxD+	8	Send Data+
C(B)	C+	OUT	RTS+	3	Control Out+
R(B)	R+	IN	RxD+	4	Receive Data+
I(B)	I+	IN	CTS+	2	Control In+

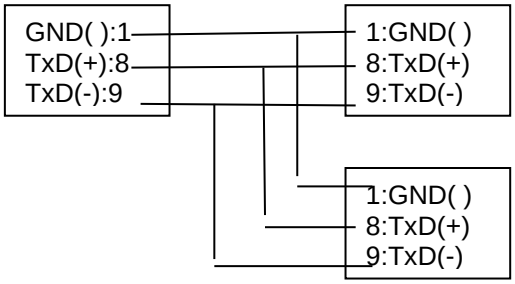
9 Pin D-sub Cable: CWB-3025RU

8. Connection

Full Duplex



Half Duplex



9. FAQ

Q1
通信を行う際、root 権限がないと open 時にエラーとなります。
また、”chmod 666 /dev/ttyG*”を行っても、電源を入れなおすたびに設定が戻ってしまいます。
また、デバイスの権利を変更した場合、open は実施できるのですが、ioctl(TIOCSSERIAL)でエラーとなります。(Operation not permitted)

一般ユーザで各通信ボードでも、通信を行う方法はありますでしょうか？

A1
インターフェース社の GPG4141 ドライバは、makettynode でノードを作成しますが、RedHawk では、起動時に/etc/rc.d/init.d/ccrt_gpg4141 を呼び出しています。
従って、下記の通り、５６行目を追加していただくのが最も簡単です。

```
# vi /etc/rc.d/init.d/ccrt_gpg4141
     53         fi
     54         if [ -e /usr/local/bin/makettynode ]; then
     55                 /usr/local/bin/makettynode
     56                 chmod 666 /dev/ttyG*
     57         fi
```

この makettynode は、/usr/src/interface/gpg4141/x86_64/linux/drivers/src/cp4161/makettynode.c に存在し、コンパイル後、/usr/local/bin/にコピーしていますので、本プログラムの"0664"を"0666"に変更後、再コンパイルする方法もあります。

```
# cd /usr/src/interface/gpg4141/x86_64/linux/drivers/src/cp4161/
# make makettynode
cc -o makettynode makettynode.c
# cp -f makettynode /usr/local/bin/
```

ioctl が機能しない原因は、一般ユーザにケーパビリティ CAP_SYS_TTY_CONFIG が付与されていないことが原因です。
これは、本質的に Linux のセキュリティ設定に関する問題ですので、RedHawk のマニュアル 13 章にも記載されています。

https://www.concurrent-rt.co.jp/external/TechSup/PDF/RedHawkLinux_UsersGuide7.2_Jpn.pdf

いくつか方法はございますが、以下の手順が最も簡単です。
（ただし、root ユーザと同じ権限になってしまいます。実行プログラム単位に、必要なケーパビリティを setcap コマンドで設定する方法もありますが、全てのシステムコールに必要なケーパビリティを個別に設定するのは、かなりの手間です）
それぞれ、下記の追加を行ってください

```
vi /etc/pam.d/password-auth
# 追加
session required pam_capability.so

vi /etc/security/capability.conf
#user ログイン名  ROLE 名
user guest  admin
```

上記、guest はユーザのログイン名です。

もし、細かい設定が必要であれば、
/etc/security/capability.conf 内に、必要なケーパビリティを設定した ROLE を定義して、そのロールをユーザと共に指定します。
詳細は、マニュアルをご参照ください。

/root/mybasic/gartbasic または、 /etc/interface/"cp4161_ttyname"+"minor"duplex.cfg ファイルの実装について

4.14 以降では、データセグメントを拡張すると "Invalid address limit on user-mode return" エラーになり、SIGKILL が送信されるため　　insmod/modprobe がエラーになるため、この機能を削除しています。
この機能は、/root/mybasic/gartbasic または、 /etc/interface/"cp4161_ttyname"+"minor"duplex.cfg ファイルを読み込みディフォルトのポートの状態を定義する機能ですが、下記の modprobe/insmod 時のオプション指定で代替できます。

```
cp4161_duplexmode=half2w
cp4161_duplexmode=half4w
cp4161_duplexmode=half
```