

PCI-6202 Board Support Package Installation on RedHawk

Release Notes Revision B

September 9, 2022



1. はじめに

本書は、Concurrent Real Time Inc(CCRT)の RedHawk 上で動作する、インターフェース社製 PCI-6202 PCI ボードサポートパッケージ 用リリースノートです。

2. インストールのための条件

PCI- 6202 BSP をインストールするためには、以下の製品がインストールされている必要があります。

- PCI- 6202 ボード
- RedHawk 6.x 以上
- Extmem version 8.3 以上

PCI-6202は、PCIバスに準拠した、24ビットの多機能カウンタを2チャンネル持った カウンタ製品です。

3. インストール方法

PCI-6202 BSP は、IRQ 共有するように設計されています。もしこのデバイスの IRQ が、別のデバイスによって共有されている場合に、このドライバの性能は損なわれる場合があります。そのため、可能な限り、このボードはその IRQ が他の装置と共有されていないPCIスロットの中に実装する事が奨励されます。“lspci -v”コマンドをシステムで種々の装置の IRQ を確認するために使用することができます。

PCI-6202 BSP は、CDROM/DVD 上の RPM/DEB フォーマットで供給され、別途 extmem デバイスドライバがインストールされている必要があります。

以下に、インストールの手順を示します。:

x86_64 アーキテクチャの場合

```
==== root ユーザで実行してください====
# mount /dev/cdrom /mnt あるいは mount /dev/dvd /mnt
# cd /mnt
もし、extmem を同時にインストールする場合には、以下のコマンドを入力してください
# rpm -ivh bin-extmem-X.Y_RHx.y-z.x86_64.rpm
PCI6202 BSP 実行パッケージのインストール
# rpm -ivh bin-pci6202 -X.Y_RHx.y-z.x86_64.rpm
もし必要であれば、続けて開発パッケージのインストールを行ってください
# rpm -ivh dev-pci6202 -X.Y_RHx.y-z.x86_64.rpm
# umount /mnt
```

amd64 アーキテクチャの場合

```
==== root ユーザで実行してください====
# mount /dev/cdrom /mnt あるいは mount /dev/dvd /mnt
# cd /mnt
もし、extmem を同時にインストールする場合には、以下のコマンドを入力してください
# apt install ./bin-extmem-rhx.y_X.Y_amd64.deb
```

PCI6202 BSP 実行パッケージのインストール

```
# apt install ./bin-pci6202 -rhx.y_X.Y_amd64.deb
```

もし必要であれば、続けて開発パッケージのインストールを行ってください

```
# apt install ./dev-pci6202 -rhx.y_X.Y_amd64.deb
# umount /mnt
```

arm64 アーキテクチャの場合

```
==== root ユーザで実行してください====
# mount /dev/cdrom /mnt あるいは mount /dev/dvd /mnt
# cd /mnt
もし、extmem を同時にインストールする場合には、以下のコマンドを入力してください
```

```
# apt install ./bin-extmem-rhx.y_X.Y_arm64.deb
```

PCI6202 BSP 実行パッケージのインストール

```
# apt install ./bin-pci6202 -rhx.y_X.Y_arm64.deb
```

もし必要であれば、続けて開発パッケージのインストールを行ってください

```
# apt install ./dev-pci6202 -rhx.y_X.Y_arm64.deb  
# umount /mnt
```

(**x.y** は RedHawk のバージョン番号であり、6.x,7.x または 8.x で、**X.Y** は、BSP のバージョン、**z** は、BSP のリリース番号を示し、予告なく変更することがあります。)

PCI-6202 BSP パッケージは **/usr/local/CNC/drivers/extmem/interface/pci6202** ディレクトリにインストールされ、必要な場所に展開されます。

4. アンインストール方法

PCI-6202 BSP パッケージは、以下のコマンドでアンインストールします。この作業により **/usr/local/CNC/drivers/extmem/interface/ pci6202** ディレクトリは削除されます。

x86_64 アーキテクチャの場合

=== root ユーザで実行してください===

開発パッケージをインストールしていた場合には、

```
# rpm -e dev-pci6202 -X.Y_RHx.y-z.x86_64 (開発パッケージの削除)
```

```
# rpm -e bin-pci6202 -X.Y_RHx.y-z.x86_64 (実行パッケージの削除)
```

実行パッケージのみをインストールしていた場合には、

```
# rpm -e bin-pci6202 -X.Y_RHx.y-z.x86_64 (実行パッケージの削除)
```

amd64 アーキテクチャの場合

=== root ユーザで実行してください===

開発パッケージをインストールしていた場合には、

```
# apt purge dev-pci6202 -rhx.y (開発パッケージの削除)
```

```
# apt purge bin-pci6202 -rhx.y (実行パッケージの削除)
```

実行パッケージのみをインストールしていた場合には、

```
# apt purge bin-pci6202 -rhx.y (実行パッケージの削除)
```

arm64 アーキテクチャの場合

=== root ユーザで実行してください===

開発パッケージをインストールしていた場合には、

```
# apt purge dev-pci6202 -rhx.y (開発パッケージの削除)
```

```
# apt purge bin-pci6202 -rhx.y (実行パッケージの削除)
```

実行パッケージのみをインストールしていた場合には、

```
# apt purge bin-pci6202 -rhx.y (実行パッケージの削除)
```

5. ライブラリマニュアル

ライブラリマニュアルは、オンラインで提供されます。

```
# man pci6202
```

```
pci6202(3)
```

```
pci6202(3)
```

NAME

pci6202 - external memory device access library

SYNOPSIS

[ボードの詳細は、各マニュアルを見てください]

DESCRIPTION

pci6202 は、external memory ドライバを利用した pci6202 ボードサポートライブラリです。

```
#include <sys/pci6202.h>
gcc [options ...] file -lpci6202 -lxtmem ...
```

```
*****
PCI6202
*****
```

割り込みハンドラの登録

```
int pci6202_setup_signal
```

```
(
    int fd,
    void (*interrupt_hadler)( int, siginfo_t *, void *),
    unsigned long int mask,
    unsigned long int mask0,
    unsigned long int mask1
```

```
);
```

戻り値

エラーなら-1 成功なら 0

引数

fd ファイルディスクリプタ番号

void (*interrupt_hadler)(int, siginfo_t *, void *) 割り込みハンドラ

mask タイマー 割り込みを許可するマスク値 (PCI6202_IMASK_TIMER)

mask0 CH0 割り込みを許可するマスク値(外部クロック:PCI6202_IMASK_CLKI,外部ラッチ:PCI6202_IMASK_CLKL)

mask1 CH1 割り込みを許可するマスク値(外部クロック:PCI6202_IMASK_CLKI,外部ラッチ:PCI6202_IMASK_CLKL)

デバイスの非初期化处理

```
int pci6202_reset(int fd);
```

```
int pci6202_uninit(int fd);
```

戻り値
エラーなら-1 成功なら 0

引数

fd ファイルディスクリプタ番号

2つの関数は同じ処理、全ての制御レジスタに 0 値を設定する。

デバイスの初期化处理

```
int pci6202_init
```

```
(
    int fd,
    int option
```

```
);
```

戻り値

エラーなら-1 成功なら 0

引数

fd ファイルディスクリプタ番号

option 1を指定すると以下の情報が表示される

BAR0 I/O Region addr 0x0000cce0 offset 0x00000000 16 bytes

BAR1 I/O Region addr 0x0000ccf0 offset 0x00000000 16 bytes Switch 1

割り込みサービス関数 割り込んだ際の割り込み要因レジスタの値を戻す

int pci6202_intr_service

```
(
    int fd,
    unsigned long int *iflag,
    unsigned long int *iflag0,
    unsigned long int *iflag1,
    unsigned long int *pending
);
```

戻り値

エラーなら-1 成功なら 0

引数

fd ファイルディスクリプタ番号

iflag タイマー割り込みフラグ値を戻す変数

iflag0 CH0 割り込みフラグ値を戻す変数

iflag1 CH1 割り込みフラグ値を戻す変数

pending 保留されている割り込みの数を戻す変数

割り込みを禁止する

int pci6202_disable_intrrupt

```
(
    int fd,
    unsigned long int mask,
    unsigned long int mask0,
    unsigned long int mask1
);
```

戻り値

エラーなら-1 成功なら 0

引数

fd ファイルディスクリプタ番号

mask タイマー割り込みを禁止するマスク値 (PCI6202_IMASK_TIMER)

mask0 CH0 割り込みを禁止するマスク値(外部クロック:PCI6202_IMASK_CLKI,外部ラッ
チ:PCI6202_IMASK_CLKL)

mask1 CH1 割り込みを禁止するマスク値(外部ク ロック:PCI6202_IMASK_CLKI,外部ラッ
チ:PCI6202_IMASK_CLKL)

割り込みを許可する

int pci6202_enable_intrrupt

```
(
    int fd,
    unsigned long int mask,
    unsigned long int mask0,
    unsigned long int mask1
);
```

戻り値

エラーなら-1 成功なら 0

引数

fd ファイルディスクリプタ番号

mask0 CH0 割 り 込 み を 許 可 す る マ ス ク 値(外部クロック:PCI6202_IMASK_CLKI,外
部ラッチ:PCI6202_IMASK_CLKL)

mask1 CH1 割 り 込 み を 許 可 す る マ ス ク 値(外部ク ロック:PCI6202_IMASK_CLKI,外部
ラッチ:PCI6202_IMASK_CLKL)

mask 割り込みを禁止するビットマスク 以下のいずれかを指定する

インターバルタイマーをセットする

```
int pci6202_set_interval_timer(int fd,unsigned long int base,unsigned long int div);
```

戻り値

エラーなら-1 成功なら 0

引数

fd ファイルディスクリプタ番号

base ベースクロック値 以下のいずれかを指定する

PCI6202_TIMER_BASE_STOP 停止

PCI6202_TIMER_BASE_010USEC 10 マイクロ秒

PCI6202_TIMER_BASE_100USEC 100 マイクロ秒

PCI6202_TIMER_BASE_001MSEC 1 ミリ秒

PCI6202_TIMER_BASE_010MSEC 10 ミリ秒

PCI6202_TIMER_BASE_100MSEC 100 ミリ秒

div ベースクロックを分周する値 カウントダウンし 0 の時割り込

みが発生する

最大15分周しかできない

インターバルタイマーの現在値を読み出す

```
int pci6202_get_interval_timer(int fd,unsigned long int *count);
```

戻り値

エラーなら-1 成功なら 0

引数

fd ファイルディスクリプタ番号

count 値を読み出す変数へのポインタ

汎用関数 オフセット値を指定してレジスタの値を読み出す

pci6202_get_ioport() は、PCI ベースアドレス 0 エリアを、pci6202_get_ioport1()は、PCI ベースアドレス 1 エリアを 1 バイトで読み出す。

pci6202_get_ioport_long は、PCI ベースアドレス0エリアを、4 バイトで読み出す。

```
int pci6202_get_ioport(int fd,int offset,unsigned long int *value);
```

```
int pci6202_get_ioport1(int fd,int offset,unsigned long int *value);
```

```
int pci6202_get_ioport_long(int fd,int offset,unsigned long int *value);
```

戻り値

エラーなら-1 成功なら 0

引数

fd ファイルディスクリプタ番号

offset レジスタオフセット

value 値を読み出す変数へのポインタ

汎用関数 オフセット値を指定してレジスタに値を書き出す

```
int pci6202_set_ioport(int fd,int offset,unsigned long int *value);
```

```
int pci6202_set_ioport1(int fd,int offset,unsigned long int *value);
```

pci6202_set_ioport() は、PCI ベースアドレス 0 エリアを、pci6202_set_ioport1()は、PCI ベースアドレス1エリアを 1 バイトで書き出す。

戻り値

エラーなら-1 成功なら 0

引数

fd ファイルディスクリプタ番号

offset レジスタオフセット

value 値を出す変数へのポインタ

DIP スイッチの読み込み

```
int pci6202_get_sw(int fd,unsigned long int *data);
```

戻り値

エラーなら-1 成功なら 0

引数

fd ファイルディスクリプタ番号

data 出力変数へのポインタ

SEE ALSO

/usr/local/CNC/drivers/extmem/interface/pci6202 下のプログラム

AUTHORS

Copyright (C) 1995-2016 Concurrent Real Time Inc

28 Apr 2016

pci6202(3)