

PEX-224140 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 PEX-224140 PCI Express 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

PEX- 224140 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。

- PEX- 224140 ボード
- RedHawk 6.x 以上
- Extmem version 8.3 以上

PEX-224140は、32点フォトカプラ型高耐圧接点入力(シンク型出力対応)を持つ、PCI Expressデジタル入力製品です。。

3. インストール方法

PEX-224140 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。

PEX-224140 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
```

PEX224140 BSP 実行パッケージのインストール

```
# rpm -ivh bin-pex224140 -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-pex224140 -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
```

PEX224140 BSP 実行パッケージのインストール

```
# apt install ./bin-pex224140 -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-pex224140 -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
```

PEX224140 BSP 実行パッケージのインストール

```
# apt install ./bin-pex224140 -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-pex224140 -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。)

PEX-224140 BSP パッケージは **/usr/local/CNC/drivers/extmem/interface/pex224140** 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

PEX-224140 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/pex224140** 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-pex224140 -X.Y_RHx.y-z.x86_64 (開発パッケージの削除)
# rpm -e bin-pex224140 -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-pex224140 -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-pex224140 -rhx.y (開発パッケージの削除)
# apt purge bin-pex224140 -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-pex224140 -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-pex224140 -rhx.y (開発パッケージの削除)
# apt purge bin-pex224140 -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-pex224140 -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 pex224140pex224140(3)
```

pex224140(3)

NAME

pex224140 - external memory board support library

SYNOPSIS

[ボードの詳細は、PEX224140 マニュアルを見てください]

DESCRIPTION

pex224140 は、external memory ドライバを利用した pex224140 ボードアクセスライブラリです。

本ボードのレジスタは、メモリマップタイプであるため、ユーザ空間からでもアクセス可能です。

一般的にユーザ空間からのアクセスは高速で、各関数の末尾が `_mmap` である場合、ユーザ空間からレジスタアクセスします。

しかし、割り込みを扱う関数は、カーネル空間からしかアクセス出来ないため、`ioctl()` を使いカーネル空間で、BUS WINDOW モードプログラムでレジスタアクセスします。

```
#include <sys/pex224140.h>
```

```
gcc [options ...] file -lpex224140 -lxtmem ...
```

OPEN/CLOSE/MMAP

PEX224140 は、通常のデバイスファイルと同様に `open/close` 可能です。デバイスは、実使用の前に必ずユーザーが初期化する必要があります。デフォルトでは、非共有モードですが、`IOCTL_EXTMEM_SHARED` を発行すると、複数のユーザーでデバイスを共有できます。但し、レジスタなどの整合性の責任はユーザーに任せられます。デバイスドライバでは最初に `open` したプロセスが最後に `close` することを仮定しています。典型的なレジスタ初期化の手続きは、ライブラリとして提供されているため、プログラムテンプレートを使用します。

ボードへの割り込みは、アクセスライブラリによって `extmem` デバイスドライバに登録された割り込み手続きによって処理されます。加えて必要であれば以下の例のように(SIGIO)シグナルハンドラを使用して追加の処理を行うことができます。アクセスライブラリでは、以下の場合に割り込みレジスタをアクセスします。

(1) `pex224140_init()`, `pex224140_reset()`, `pex224140_uninit()`, `pex224140_enable_intrrupt()`, など関数呼び出し時

(2) 実際の割り込みが発生した時

オフセット `0x0C(INTR)`を読み込み、ON になっているビットをクリアする
この値は、`pex224140_intr_service()`関数で、読み出すことができます。

ただし、関数を呼び出す前に連続して割り込みが発生した場合には、値は上書きされます。

また値が上書きされた場合には `pex224140_intr_service()` 関数の `pendig` 値で検出できます。

(3) アプリケーションプログラムがデバイスを `close()`した時、あるいは異常終了したとき

```
*****  
PEX224140  
*****
```

割り込みハンドラの登録

```
int pex224140_setup_signal ( int fd, void (*interrupt_handler)(int,  
siginfo_t *, void *), int mask);
```

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_handler)(int, siginfo_t *, void *) 割り込み
ハンドラ
mask 割り込みを許可するマスク値

デバイスの非初期化処理

```
int pex224140_reset(int fd);
int pex224140_reset_mmap(PEX224140R *dev);
int pex224140_uninit(int fd, PEX224140R *dev);
```

戻り値
エラーなら-1 成功なら 0

引数
dev pex224140 のデバイスメモリへのポインタ
fd ファイルディスクリプタ番号

デバイスの初期化処理

```
int pex224140_init(int fd, PEX224140R **dev, int *dev_size, 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
dev pex224140 のデバイスメモリへのポインタが返される
このポインタを利用すると高速にアクセスすることができる
dev_size pex224140 のデバイスメモリのサイズが返される(4096)
BAR0 MEM Region addr 0xebff000 offset 0x00000000 4096 bytes

Switch 0

pex224140 の出力を発生させる

```
int pex224140_raise_signal ( int fd, int ack, int out1);
int pex224140_raise_signal_mmap(PEX224140R *dev,int ack,int out1);
```

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
dev pex224140 のデバイスメモリへのポインタ
ack,out1,out2 割り込みの種類 以下のいずれかを指定する
ack1

	PEX224140_PULS_ACK1_NOACTION	なにもしない
	PEX224140_PULS_ACK1_LOW_HIGH	ACK1 端 子 を クリア

Low->High

PEX224140_PULS_ACK1_HIGH_LOW	ACK1 端 子 を クリア
------------------------------	----------------

High->Low

out1

PEX224140_PULS_OUT1_NOACTION	なにもしない
PEX224140_PULS_OUT1_LEVEL_HIGH	High レベル出力
PEX224140_PULS_OUT1_LEVEL_LOW	Low レベル出力
PEX224140_PULS_OUT1_PULSE_LOW	Low パルスを出力

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(オフセット 0x0c)
の値を戻す

```
int pex224140_intr_service ( int fd, unsigned int *iflag, int *pend-  
ing);
```

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 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 pex224140_disable_intrrupt ( int fd, unsigned long int mask);  
int pex224140_disable_intrrupt_mmap(PEX224140R *dev , unsigned long int  
mask);
```

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

dev pex224140 のデバイスメモリへのポインタ

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

PEX224140_IMASK_STB1 STB1 からの入力信号

PEX224140_IMASK_IN1 IN1 からの入力信号

PEX224140_IMASK_IN2 IN2 からの入力信号

PEX224140_IMASK_IN3 IN3 からの入力信号

PEX224140_IMASK_IN4 IN4 からの入力信号

PEX224140_IMASK_TIMER タイマー割り込み

PEX224140_IMASK_RESET リセット割り込み

PEX224140_IMASK_ALL 上記のすべて

割り込みを許可する

```
int pex224140_enable_intrrupt ( int fd, unsigned long int mask);  
int pex224140_enable_intrrupt_mmap(PEX224140R *dev,unsigned long int  
mask);
```

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

dev pex224140 のデバイスメモリへのポインタ

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

PEX224140_IMASK_STB1 STB1 からの入力信号

PEX224140_IMASK_IN1 IN1 からの入力信号

PEX224140_IMASK_IN2 IN2 からの入力信号

PEX224140_IMASK_IN3 IN3 からの入力信号

PEX224140_IMASK_IN4 IN4 からの入力信号

PEX224140_IMASK_TIMER タイマー割り込み

PEX224140_IMASK_RESET リセット割り込み

PEX224140_IMASK_ALL 上記のすべて

インターバルタイマーをセットする

```
int pex224140_set_interval_timer(int fd,unsigned unsigned long int  
base,unsigned unsigned long int div);  
int pex224140_set_interval_timer_mmap(PEX224140R *dev,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

dev pex224140 のデバイスメモリへのポインタ

base ベースクロック値 以下のいずれかを指定する

PEX224140_TIMER_BASE_STOP 停止

PEX224140_TIMER_BASE_010USEC 10 マイクロ秒

PEX224140_TIMER_BASE_100USEC 100 マイクロ秒

PEX224140_TIMER_BASE_001MSEC 1 ミリ秒

PEX224140_TIMER_BASE_010MSEC 10 ミリ秒

PEX224140_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 pex224140_get_interval_timer(int fd,unsigned long int *count);
```

```
int pex224140_get_interval_timer_mmap(PEX224140R *dev,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

dev pex224140 のデバイスメモリへのポインタ

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

```
int pex224140_get_ioport(int fd,int offset,unsigned long int *value);
```

```
int pex224140_get_mmap(PEX224140R *dev ,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

dev pex224140 のデバイスメモリへのポインタ

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 pex224140_set_ioport(int fd,int offset,unsigned long int *value);
```

```
int pex224140_set_mmap(PEX224140R *dev ,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

dev pex224140 のデバイスメモリへのポインタ

offset レジスタオフセット

value 値を出す変数へのポインタ

チャンネルを指定して入力データを読み出す

```
int pex224140_read_data(int fd,int ch,unsigned char *data);
```

```
int pex224140_read_data_mmap(PEX224140R *dev,int ch,unsigned char *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

dev pex224140 のデバイスメモリへのポインタ

ch チャンネル(0-5)
data 値を出す変数へのポインタ

すべてのチャンネルの入力データを読み出す

```
int pex224140_read_data_all(int fd,unsigned char *data);
```

```
int pex224140_read_data_all_mmap(PEX224140R *dev,unsigned char *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

dev pex224140 のデバイスメモリへのポインタ

data 値を出す配列変数へのポインタ

DIP スイッチの読み込み

```
int pex224140_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/pex224140 下のプログラム

AUTHORS

Copyright (C) 1995-2016 Concurrent Real Time Inc.