

# PEX-H285011 Board Support Package Installation on RedHawk

## Release Notes Revision B

September 12, 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-H285011 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- H285011 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-H285011 ボード
- RedHawk 6.x 以上
- Extmem version 8.3 以上

PEX-H285011は、PCI Expressに準拠した、シンク型出力対応電流駆動フォトカプラ入力の64点デジタル信号入力と、高電流オープンコレクタシンク型出力64点のデジタル信号出力を持つデジタル入出力製品です。

(注意:本ボードは1ボードで2つの論理デバイスが存在し、偶数は入力、奇数は出力チャンネルになります。)

## 3. インストール方法

**PEX-H285011 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-H285011 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。:

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
```

**PEXH285011 BSP 実行パッケージのインストール**

```
# apt install ./bin-pexh285011 -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-pexh285011 -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
```

**PEXH285011 BSP 実行パッケージのインストール**

```
# apt install ./bin-pexh285011 -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-pexh285011 -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-H285011 BSP** パッケージは `/usr/local/CNC/drivers/extmem/interface/pexh285011` 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-H285011 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/pexH285011` 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-pexh285011 -X.Y\_RHx.y-z.x86\_64 (開発パッケージの削除)

# rpm -e bin-pexh285011 -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-pexh285011 -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-pexh285011 -rhx.y (開発パッケージの削除)

# apt purge bin-pexh285011 -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-pexh285011 -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-pexh285011 -rhx.y (開発パッケージの削除)

# apt purge bin-pexh285011 -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-pexh285011 -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 pexh285011

pexh285011(3)

pexh285011(3)

### NAME

pexh285011 - 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

pexh285011 は、external memory ドライバを利用した pexh285011 ボードアクセスライブラリです。

```
#include <sys/pexh285011.h>
```

```
gcc [options ...] file -lpexh285011 -lxtmem ...
```

### OPEN/CLOSE/MMAP

PEXH285011 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されます。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。

注意：本ボードは、一つの物理デバイスで二つの独立した論理デバイス（入力と出力）を構成します。

論理デバイスは、サブシステム ID+サブベンダーID で区別することができます。

入力：0x2D0B1147

出力：0x2D8B1147

サブシステム ID は、以下の API で取得できます。

```
extmem_device_info_t    device_info;
ioctl(fd, IOCTL_EXTMEM_GET_DEVICE_INFO, &device_info);
if (device_info.subsys_id==0x2D0B1147)
{
    printf("Input Board¥n");
}
if (device_info.subsys_id==0x2D8B1147)
{
    printf("Output Board¥n");
}
```

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) pexh285011\_init(), pexh285011\_reset(), pexh285011\_uninit(),

pexh285011\_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、pexh285011\_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 pexh285011\_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

\*\*\*\*\*

PEXH285011

\*\*\*\*\*

DIP スイッチの読み込み

int pexh285011\_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

割り込みハンドラの登録

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 割り込みを許可するマスク値

デバイスの非初期化处理

int pexh285011\_reset(int fd);

int pexh285011\_reset\_mmap(PEXH285011R \*dev);

int pexh285011\_uninit(int fd,PEXH285011R \*dev); 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 pexh285011 のデバイスメモリへのポインタ

デバイスの初期化处理

int pexh285011\_init(int fd,PEXH285011R \*\*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 pexh285011 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 pexh285011 のデバイスメモリのサイズが返される(4096)

BAR0 MEM Region addr 0xebfff000 offset 0x00000000 4096 bytes Switch 0 **Input**

Board

または

BAR0 MEM Region addr 0xebfff000 offset 0x00000000 4096 bytes Switch 0 **Output**

Board

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 pexh285011_intr_service ( int fd, unsigned int *iflag, 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 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 pexh285011_disable_intrrupt ( int fd, unsigned long int mask);
```

```
int pexh285011_disable_intrrupt_mmap(PEXH285011R *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 pexh285011 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

PEXH285011\_IMASK\_SIG1 IN1 からの入力信号

PEXH285011\_IMASK\_SIG2 IN2 からの入力信号

PEXH285011\_IMASK\_SIG3 IN3 からの入力信号

PEXH285011\_IMASK\_SIG4 IN4 からの入力信号

PEXH285011\_IMASK\_TIMER タイマー割り込み

PEXH285011\_IMASK\_RESET リセット割り込み

PEXH285011\_IMASK\_ALL 上記のすべて

割り込みを許可する

```
int pexh285011_enable_intrrupt ( int fd, unsigned long int mask);
```

```
int pexh285011_enable_intrrupt_mmap(PEXH285011R *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 pexh285011 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

PEXH285011\_IMASK\_SIG1 IN1 からの入力信号

PEXH285011\_IMASK\_SIG2 IN2 からの入力信号

PEXH285011\_IMASK\_SIG3 IN3 からの入力信号

PEXH285011\_IMASK\_SIG4 IN4 からの入力信号

PEXH285011\_IMASK\_TIMER タイマー割り込み

PEXH285011\_IMASK\_RESET リセット割り込み

PEXH285011\_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 pexh285011_set_interval_timer(int fd,unsigned int base,unsigned long int div);
```

```
int pexh285011_set_interval_timer_mmap(PEXH285011R *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 pexh285011 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

|                               |           |
|-------------------------------|-----------|
| PEXH285011_TIMER_BASE_STOP    | 停止        |
| PEXH285011_TIMER_BASE_010USEC | 10 マイクロ秒  |
| PEXH285011_TIMER_BASE_100USEC | 100 マイクロ秒 |
| PEXH285011_TIMER_BASE_001MSEC | 1 ミリ秒     |
| PEXH285011_TIMER_BASE_010MSEC | 10 ミリ秒    |
| PEXH285011_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 pexh285011_get_interval_timer(int fd,unsigned long int *count);
```

```
int pexh285011_get_interval_timer_mmap(PEXH285011R *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 pexh285011 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 pexh285011_get_ioport(int fd,int offset,unsigned long int *value);
```

```
int pexh285011_get_mmap(PEXH285011R *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 pexh285011 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 pexh285011_set_ioport(int fd,int offset,unsigned long int *value);
```

```
int pexh285011_set_mmap(PEXH285011R *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 pexh285011 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 pexh285011_read_data(int fd,int ch,unsigned char *data);
```

```
int pexh285011_read_data_mmap(PEXH285011R *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 pexh285011 のデバイスメモリへのポインタ

ch チャンネル

data 値を出す変数へのポインタ

チャンネルを指定してデータを出力する

```
int pexh285011_write_data(int fd,int ch,unsigned char *data);
int pexh285011_write_data_mmap(PEXH285011R *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 pexh285011 のデバイスメモリへのポインタ

ch チャンネル

data 出力変数へのポインタ

すべてのチャンネルの入力データを読み出す

```
int pexh285011_read_data_all(int fd,unsigned char *data);
```

```
int pexh285011_read_data_all_mmap(PEXH285011R *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 pexh285011 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

すべてのチャンネルのデータを出力する

```
int pexh285011_write_data_all(int fd,unsigned char *data);
```

```
int pexh285011_write_data_all_mmap(PEXH285011R *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 pexh285011 のデバイスメモリへのポインタ

data 出力配列変数へのポインタ

SEE ALSO

/usr/local/CNC/drivers/extmem/interface/pexh285011 下のプログラム

AUTHORS

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