

PEX-H291388 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-H291388 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-H291388 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- H291388 ボード
- RedHawk 6.x 以上
- Extmem version 8.3 以上

PEX-H291388は、PCI Express対応64点TTLシュミット入力と、64点TTLオープンコレクタ出力(プルアップ抵抗付き)を持つデジタル入出力製品です。

3. インストール方法

PEX-H291388 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-H291388 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
```

PEXH291388 BSP 実行パッケージのインストール

```
# apt install ./bin-pexh291388 -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-pexh291388 -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
```

PEXH291388 BSP 実行パッケージのインストール

```
# apt install ./bin-pexh291388 -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-pexh291388 -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-H291388 BSP パッケージは `/usr/local/CNC/drivers/extmem/interface/ pexh291388` 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-H291388 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/ pexH291388` 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-pexh291388 -X.Y_RHx.y-z.x86_64 (開発パッケージの削除)

rpm -e bin-pexh291388 -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-pexh291388 -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-pexh291388 -rhx.y (開発パッケージの削除)

apt purge bin-pexh291388 -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-pexh291388 -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-pexh291388 -rhx.y (開発パッケージの削除)

apt purge bin-pexh291388 -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-pexh291388 -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 pexh291388

pexh291388(3)

pexh291388(3)

NAME

pexh291388 - 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

pexh291388 は、external memory ドライバを利用した pexh291388 ボードアクセスライブラリです。

```
#include <sys/pexh291388.h>
```

```
gcc [options ...] file -lpexh291388 -lxtmem ...
```

OPEN/CLOSE/MMAP

PEXH291388 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。

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) pexh291388_init(), pexh291388_reset(), pexh291388_uninit(), pexh291388_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、pexh291388_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 pexh291388_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

PEXH291388

割り込みハンドラの登録

```
int pexh291388_setup_signal ( int fd, void (*interrupt_hadler)(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_hadler)(int) 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 pexh291388_reset(int fd);
```

```
int pexh291388_reset_mmap(PEXH291388R *dev);
```

```
int pexh291388_uninit(int fd,PEXH291388R *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 pexh291388 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 pexh291388_init(int fd,PEXH291388R **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 pexh291388 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 pexh291388 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

pexh291388 の出力を発生させる

```
int pexh291388_raise_signal_mmap(PEXH291388R *dev,int ack,int out1,int out2);
```

```
int pexh291388_raise_signal ( int fd, int ack, int out1, int out2);
```

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 pexh291388 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

PEXH291388_PULS_ACK1_NOACTION なにもしない

PEXH291388_PULS_ACK1_LOW_HIGH ACK1 端子 をクリア Low->High

PEXH291388_PULS_ACK1_HIGH_LOW ACK1 端子 をクリア High->Low

out1

PEXH291388_PULS_OUT1_NOACTION なにもしない

PEXH291388_PULS_OUT1_LEVEL_HIGH High レベル出力

PEXH291388_PULS_OUT1_LEVEL_LOW Low レベル出力

PEXH291388_PULS_OUT1_PULSE_LOW Low パルスを出力

out2

PEXH291388_PULS_OUT2_NOACTION なにもしない

PEXH291388_PULS_OUT2_LEVEL_HIGH High レベル出力

PEXH291388_PULS_OUT2_LEVEL_LOW Low レベル出力

PEXH291388_PULS_OUT2_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 pexh291388_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 pexh291388_disable_intrrupt ( int fd, unsigned long int mask);
```

```
int pexh291388_disable_intrrupt_mmap(PEXH291388R *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 pexh291388 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

PEXH291388_IMASK_STB1 STB1 からの入力信号

PEXH291388_IMASK_IRIN1 IR.IN1 からの入力信号

PEXH291388_IMASK_IRIN2 IR.IN2 からの入力信号

PEXH291388_IMASK_TIMER タイマー割り込み

PEXH291388_IMASK_RESET リセット割り込み

PEXH291388_IMASK_IACK2 ACK2 割り込み

PEXH291388_IMASK_ALL 上記のすべて

割り込みを許可する

```
int pexh291388_enable_intrrupt ( int fd, unsigned long int mask);
```

```
int pexh291388_enable_intrrupt_mmap(PEXH291388R *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 pexh291388 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

PEXH291388_IMASK_STB1 STB1 からの入力信号

PEXH291388_IMASK_IRIN1 IR.IN1 からの入力信号

PEXH291388_IMASK_IRIN2 IR.IN2 からの入力信号

PEXH291388_IMASK_TIMER タイマー割り込み

PEXH291388_IMASK_RESET リセット割り込み

PEXH291388_IMASK_IACK2 ACK2 割り込み

PEXH291388_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 pexh291388_set_interval_timer(int fd,unsigned int base,unsigned long int div);
```

```
int pexh291388_set_interval_timer_mmap(PEXH291388R *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 pexh291388 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

PEXH291388_TIMER_BASE_STOP	停止
PEXH291388_TIMER_BASE_010USEC	10 マイクロ秒
PEXH291388_TIMER_BASE_100USEC	100 マイクロ秒
PEXH291388_TIMER_BASE_001MSEC	1 ミリ秒
PEXH291388_TIMER_BASE_010MSEC	10 ミリ秒
PEXH291388_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 pexh291388_get_interval_timer(int fd,unsigned long int *count);
int pexh291388_get_interval_timer_mmap(PEXH291388R *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 pexh291388 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 pexh291388_get_ioport(int fd,int offset,unsigned long int *value);
int pexh291388_get_mmap(PEXH291388R *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 pexh291388 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 pexh291388_set_ioport(int fd,int offset,unsigned long int *value);
int pexh291388_set_mmap(PEXH291388R *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 pexh291388 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 pexh291388_read_data(int fd,int ch,unsigned char *data);
int pexh291388_read_data_mmap(PEXH291388R *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 pexh291388 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 pexh291388_write_data(int fd,int ch,unsigned char *data);  
int pexh291388_write_data_mmap(PEXH291388R *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 pexh291388 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 pexh291388_read_data_all(int fd,unsigned char *data);  
int pexh291388_read_data_all_mmap(PEXH291388R *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 pexh291388 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 pexh291388_write_data_all(int fd,unsigned char *data);  
int pexh291388_write_data_all_mmap(PEXH291388R *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
data 出力配列変数へのポインタ

DIP スイッチの読み込み

```
int pexh291388_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/pexh291388 下のプログラム

AUTHORS

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