PEX-H251500 Board Support Package Installation on RedHawk

Release Notes Revision B

September 19, 2022

concurrent
REAL-TIME

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-H251500 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-H251500 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事が必要です。

- PEX- H251500 ボード
- RedHawk 6.x 以上
- Extmem version 8.3 以上

PEX-H251500は、PCI Expressに準拠した、32点フォトモスリレー出力(シンク・ソース型)を持つリレー出力製品です。
タイマカウンタを搭載しているので、インターバルタイマとして使用できます。

3. インストール方法

PEX-H251500 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-H251500 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ことが必要です。

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
```

PEXH251500 BSP 実行パッケージのインストール

```
# apt install ./bin-pexh251500 -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-pexh251500 -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
```

PEXH251500 BSP 実行パッケージのインストール

```
# apt install ./bin-pexh251500 -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-pexh251500 -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。)
)

PEXH251500 BSP パッケージは ***/usr/local/CNC/drivers/extmem/interface/pexh251500*** 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。

(***x.y*** は RedHawk のバージョン番号であり、 6.0,6.3,6.5,7.0,7.2,7.3 または 7.5 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-H251500 BSP パッケージは ***/usr/local/CNC/drivers/extmem/interface/pexh251500*** 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-H251500 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/pexh251500*** 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-pexh251500 -X.Y_RHx.y- z .x86_64 (開発パッケージの削除)
rpm -e bin-pexh251500 -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-pexh251500 -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-pexh251500 -rhx.y (開発パッケージの削除)
apt purge bin-pexh251500 -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-pexh251500 -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-pexh251500 -rhx.y (開発パッケージの削除)
apt purge bin-pexh251500 -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-pexh251500 -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 pexh251500
pexh251500(3)　　Library Functions Manual　　pexh251500(3)

NAME
　pexh251500 - external memory board support library

SYNOPSIS
　[ボードの詳細は、各マニュアルを見てください]

DESCRIPTION
　pexh251500 は、external memory ドライバを利用した pexh251500 ボードアクセスライブラリです。

　#include <sys/pexh251500.h>
　gcc [options ...] file -lpexh251500 -lextmem ...

OPEN/CLOSE/MMAP
　PEXH251500　　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。
　ディフォルトでは、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) pexh251500_init(), pexh251500_reset(), pexh251500_uninit(), pexh251500_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、pexh251500_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 pexh251500_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

　**********
　PEXH251500
　**********

　割り込みハンドラの登録
　int pexh251500_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_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 pexh251500_reset(int fd);
int pexh251500_reset_mmap(PEXH251500R *dev);
int pexh251500_uninit(int fd,PEXH251500R *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 pexh251500 のデバイスメモリへのポインタ

デバイスの初期化処理
int pexh251500_init(int fd,PEXH251500R **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 pexh251500 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 pexh251500 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

pexh251500 の出力を発生させる
int pexh251500_raise_signal (int fd, int ack, int out1, int out2);
int pexh251500_raise_signal_mmap(PEXH251500R *dev,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 pexh251500 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
PEXH251500_PULS_ACK1_NOACTION　　なにもしない
out1
PEXH251500_PULS_OUT1_NOACTION　　なにもしない
PEXH251500_PULS_OUT1_LEVEL_HIGH High レベル出力
PEXH251500_PULS_OUT1_LEVEL_LOW Low レベル出力
PEXH251500_PULS_OUT1_PULSE_LOW Low パルスを出力
out2
PEXH251500_PULS_OUT2_NOACTION　　なにもしない
PEXH251500_PULS_OUT2_LEVEL_HIGH High レベル出力
PEXH251500_PULS_OUT2_LEVEL_LOW Low レベル出力
PEXH251500_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 pexh251500_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 pexh251500_disable_intrrupt (int fd, unsigned long int mask);
int pexh251500_disable_intrrupt_mmap(PEXH251500R *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 pexh251500 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

PEXH251500_IMASK_IRIN1 IR.IN1 からの入力信号
PEXH251500_IMASK_IRIN2 IR.IN2 からの入力信号
PEXH251500_IMASK_TIMER タイマー割り込み
PEXH251500_IMASK_RESET リセット割り込み
PEXH251500_IMASK_IACK2 ACK2 割り込み
PEXH251500_IMASK_ALL 上記のすべて

割り込みを許可する

int pexh251500_enable_intrrupt (int fd, unsigned long int mask);
int pexh251500_enable_intrrupt_mmap(PEXH251500R *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 pexh251500 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

PEXH251500_IMASK_IRIN1 IR.IN1 からの入力信号
PEXH251500_IMASK_IRIN2 IR.IN2 からの入力信号
PEXH251500_IMASK_TIMER タイマー割り込み
PEXH251500_IMASK_RESET リセット割り込み
PEXH251500_IMASK_IACK2 ACK2 割り込み
PEXH251500_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 pexh251500_set_interval_timer(int fd,unsigned unsigned long int base,unsigned long int div);
int pexh251500_set_interval_timer_mmap(PEXH251500R *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 pexh251500 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

PEXH251500_TIMER_BASE_STOP 停止
PEXH251500_TIMER_BASE_010USEC 10 マイクロ秒
PEXH251500_TIMER_BASE_100USEC 100 マイクロ秒
PEXH251500_TIMER_BASE_001MSEC 1 ミリ秒
PEXH251500_TIMER_BASE_010MSEC 10 ミリ秒
PEXH251500_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 pexh251500_get_interval_timer(int fd,unsigned long int *count);
int pexh251500_get_interval_timer_mmap(PEXH251500R *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 pexh251500 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 pexh251500_get_ioport(int fd,int offset,unsigned long int *value);
int pexh251500_get_mmap(PEXH251500R *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 pexh251500 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 pexh251500_set_ioport(int fd,int offset,unsigned long int *value);
int pexh251500_set_mmap(PEXH251500R *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 pexh251500 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

チャネルを指定して入力データを読み出す
int pexh251500_read_data(int fd,int ch,unsigned char *data);
int pexh251500_read_data_mmap(PEXH251500R *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 pexh251500 のデバイスメモリへのポインタ
ch チャンネル(0 ~ 3)
data 値を出す変数へのポインタ

チャネルを指定してデータを出力する
int pexh251500_write_data(int fd,int ch,unsigned char *data);
int pexh251500_write_data_mmap(PEXH251500R *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 pexh251500 のデバイスメモリへのポインタ
ch チャンネル(0 ~ 3)
data 出力変数へのポインタ

すべてのチャネルの入力データを読み出す
int pexh251500_read_data_all(int fd,unsigned char *data);
int pexh251500_read_data_all_mmap(PEXH251500R *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 pexh251500 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

すべてのチャネルのデータを出力する

int pexh251500_write_data_all(int fd,unsigned char *data);
int pexh251500_write_data_all_mmap(PEXH251500R *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 pexh251500 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

DIP スイッチの読み込み

int pexh251500_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/pexh251500 下のプログラム

AUTHORS