# PEX-251100 Board Support Package Installation on RedHawk

## Release Notes Revision B

September 9, 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- 251100 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-251100 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- 251100 ボード
- RedHawk 6.x 以上
- Extmem version 8.3 以上

PEX-251100は、4点フォトカプラ入力(シンク・ソース型出力対応)と、4点高電流C接点制御リレー出力を持つ、PCI Express対応複合製品です。
フォトカプラおよびリードリレーにより入出力部が絶縁されています
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-251100 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-251100 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。:

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**_RH***x.y-z***.x86_64.rpm
**PEX251100 BSP 実行パッケージのインストール**
# rpm -ivh bin-pex251100 -**X.Y**_RH***x.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-pex251100 -**X.Y**_RH***x.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

**PEX251100 BSP 実行パッケージのインストール**
# apt install ./bin-pex251100 -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-pex251100 -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

**PEX251100 BSP 実行パッケージのインストール**
# apt install ./bin-pex251100 -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-pex251100 -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-251100 BSP*** パッケージは ***/usr/local/CNC/drivers/extmem/interface/pex251100*** 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。

BSP をドライバに認識させるためのシェルスクリプトが***/usr/local/CNC/drivers/extmem/interface/pex251100*** ディレクトリに、config_pex251100.sh として用意されています。このシェルスクリプトを、/etc/rc.local などから起動時に呼び出すか、直接記述することで、ドライバが利用可能になります。
ただし、異なる BSP を同時に組み込むためには、EXTMEM の番号を変更する必要があります。
例えば、すでに別の BSP が存在する場合には、
EXTMEM_VID0=0x1147 EXTMEM_DID0=0x09cf EXTMEM_INT0=2
を
EXTMEM_VID1=0x1147 EXTMEM_DID1=0x09cf EXTMEM_INT1=2 に変更し、例えば

/usr/local/CNC/drivers/extmem/extmem_start EXTMEM_VID0=0x1147 EXTMEM_DID0=0x08c1 EXTMEM_INT0=2 EXTMEM_VID1=0x1147 EXTMEM_DID1=0x09cf EXTMEM_INT1=2 の様に記述するひつようがあります。
この時、/dev/extmem1 に対するシンボリックリンクも変更する必要があります。
また、同一 BSP を複数枚組み込み、スロット位置を固定するためには、config_pex251100.fix.sh を参照してください。

# 4. アンインストール方法

***PEX-251100 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/pex251100*** 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-pex251100 -**X.Y**_RH***x.y-z***.x86_64 *(開発パッケージの削除)*
# rpm -e bin-pex251100 -**X.Y**_RH***x.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-pex251100 -**X.Y**_RH***x.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-pex251100 -rhx.y *(開発パッケージの削除)*
# apt purge bin-pex251100 -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-pex251100 -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-pex251100 -rhx.y　　*(開発パッケージの削除)*
# apt purge bin-pex251100 -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-pex251100 -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 pex251100
pex251100(3)　　Library Functions Manual　　pex251100(3)

NAME
pex251100 - 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
pex251100　　は、external　　memory
ドライバを利用した pex251100 ボードアクセスライブラリです。

#include <sys/pex251100.h>
gcc [options ...] file -lpex251100 -lextmem ...

OPEN/CLOSE/MMAP
PEX251100　　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)　pex251100_init(),　　pex251100_reset(),　　pex251100_uninit(),
pex251100_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、pex251100_intr_ser-
vice()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 pex251100_intr_ser-
vice()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

**********
PEX251100
**********

割り込みハンドラの登録
int pex251100_setup_signal(int fd,void (*interrupt_handler)(int, siginfo_t *, void *),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
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 pex251100_reset(int fd);
int pex251100_reset_mmap(PEX251100R *dev);
int pex251100_uninit(int fd,PEX251100R *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 pex251100 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 pex251100_init(int fd,PEX251100R **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 pex251100 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 pex251100 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

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 pex251100_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 pex251100_disable_intrrupt ( int fd, unsigned long int mask);
int pex251100_disable_intrrupt_mmap(PEX251100R *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 pex251100 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

| | |
|---|---|
| PEX251100_IMASK_SIG1 | SIG1 からの入力信号 |
| PEX251100_IMASK_SIG2 | SIG2 からの入力信号 |
| PEX251100_IMASK_SIG3 | SIG3 からの入力信号 |
| PEX251100_IMASK_SIG4 | SIG4 からの入力信号 |
| PEX251100_IMASK_SIGT | タイマー割り込み |
| PEX251100_IMASK_SIGR | リセット割り込み |
| PEX251100_IMASK_ALL | 上記のすべて |

割り込みを許可する

int pex251100_enable_intrrupt ( int fd, unsigned long int mask);

int pex251100_enable_intrrupt_mmap(PEX251100R *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 pex251100 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

| | |
|---|---|
| PEX251100_IMASK_SIG1 | SIG1 からの入力信号 |
| PEX251100_IMASK_SIG2 | SIG2 からの入力信号 |
| PEX251100_IMASK_SIG3 | SIG3 からの入力信号 |
| PEX251100_IMASK_SIG4 | SIG4 からの入力信号 |
| PEX251100_IMASK_SIGT | タイマー割り込み |
| PEX251100_IMASK_SIGR | リセット割り込み |
| PEX251100_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 pex251100_set_interval_timer(int fd,unsigned int base,unsigned long int div);

int pex251100_set_interval_timer_mmap(PEX251100R *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 pex251100 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

| | |
|---|---|
| PEX251100_TIMER_BASE_STOP | 停止 |
| PEX251100_TIMER_BASE_010USEC | 10 マイクロ秒 |
| PEX251100_TIMER_BASE_100USEC | 100 マイクロ秒 |
| PEX251100_TIMER_BASE_001MSEC | 1 ミリ秒 |
| PEX251100_TIMER_BASE_010MSEC | 10 ミリ秒 |
| PEX251100_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 pex251100_get_interval_timer(int fd,unsigned long int *count);
int pex251100_get_interval_timer_mmap(PEX251100R *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 pex251100 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 pex251100_get_ioport(int fd,int offset,unsigned long int *value);
int pex251100_get_mmap(PEX251100R *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 pex251100 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 pex251100_set_ioport(int fd,int offset,unsigned long int *value);
int pex251100_set_mmap(PEX251100R *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 pex251100 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 pex251100_read_data(int fd,unsigned char *data);
int pex251100_read_data_mmap(PEX251100R *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 pex251100 のデバイスメモリへのポインタ
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 pex251100_write_data(int fd,unsigned char *data);
int pex251100_write_data_mmap(PEX251100R *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 pex251100 のデバイスメモリへのポインタ
data 出力変数へのポインタ

DIP スイッチの読み込み

int pex251100_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/pex251100 下のプログラム

AUTHORS