PEX-H321416N Board Support Package Installation on RedHawk

Release Notes Revision A

February 28,2025

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-H321416N 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-H321416N 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-H321416N ボード
- RedHawk 6.x 以上
- Extmem version 8.4E 以上

PEX-H321416Nは、AD16ビットS16CH /DIOカウンタ複合(バス絶縁)を持つ、PCI Expressマルチファンクション製品です。

3. インストール方法

PEX-H321416N 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-H321416N 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

PEXH321416N BSP 実行パッケージのインストール

rpm -ivh bin-pexh321416n- **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-pexh321416n- **X.Y**_RH **x.y-z**.x86_64.rpmz

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

PEXH321416N BSP 実行パッケージのインストール

apt install ./bin-pexh321416n -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-pexh321416n -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

PEXH321416N BSP 実行パッケージのインストール
apt install ./bin-pexh321416n -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-pexh321416n -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,8.x または 9.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-H321416N BSP パッケージは ***/usr/local/CNC/drivers/extmem/interface/pexh321416n*** 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-H321416N 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/PEX-H321416n*** 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-pexh321416n -**X.Y**_RH***x.y-z***.x86_64 *(開発パッケージの削除)*
rpm -e bin-pexh321416n -**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-pexh321416n -**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-pexh321416n -rhx.y *(開発パッケージの削除)*
apt purge bin-pexh321416n -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-pexh321416n -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-pexh321416n -rhx.y *(開発パッケージの削除)*
apt purge bin-pexh321416n -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-pexh321416n -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。

pexh321416n(3) Library Functions Manual
pexh321416n(3)

NAME
pexh321416n - 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
pexh321416n は、external memory ドライバを利用した pexh321416n ボードアクセスライブラリです。

#include <sys/pexh321416n.h>
gcc [options ...] file -lpexh321416n -lextmem ...

****
PEXH321416N
****

割り込みハンドラの登録
int pexh321416n_setup_signal(int fd,void (*interrupt_handler)(int, siginfo_t *, void *))
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

デバイスの非初期化処理
int pexh321416n_uninit(int fd,PEXH321416NR *adc,PEXH321416NR *clk)
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
adc AD 制御用メモリ領域へのポインタ。
clk カウンタ制御用メモリ領域へのポインタ。

デバイスの初期化処理
int pexh321416n_init(int fd,PEXH321416NR **adc,PEXH321416NR **clk, 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
adc AD 制御用メモリ領域へのポインタ。128 バイト分占有。
clk カウンタ制御用メモリ領域へのポインタ。32 バイト分占有。
option 1を指定すると以下の情報が表示される
ADC mapped 4096 bytes,at 0x90a02000 CLK mapped 4096 bytes,at 0x90a00000
デバイスのリセット処理(pexh321416n_init(),pexh321416n_uninit()から呼び出される)
void pexh321416n_reset(PEXH321416NR *adc,PEXH321416NR *clk)
戻り値
なし
引数
adc AD 制御用メモリ領域へのポインタ。
clk カウンタ制御用メモリ領域へのポインタ。
注意
デバイスのリセット処理、初期化、非初期化を行うと、自動的に割り込みは禁止になり、DIO モードに設定される。

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 0x05)の値を戻す
int pexh321416n_intr_service(int fd,unsigned int *iflag,unsigned 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 割り込み制御レジスタ値を戻す変数
割り込んだ際の以下の割り込み要因レジスタの値
((PEXH321416N_ADC_IFR_OFFSET レジスタ) & 0xFFFF)|
((PEXH321416N_CLK_IFR_OFFSET レジスタ) & 0xFF << 16);
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 pexh321416n_disable_adc_intrrupt(PEXH321416NR *adc);
int pexh321416n_disable_clk_intrrupt(PEXH321416NR *clk);
戻り値
エラーなら-1 成功なら 0
引数
adc AD 制御用メモリ領域へのポインタ。
clk カウンタ制御用メモリ領域へのポインタ。

割り込みを許可する
int pexh321416n_enable_adc_intrrupt(PEXH321416NR *adc,int mask);
戻り値
エラーなら-1 成功なら 0
引数
adc AD 制御用メモリ領域へのポインタ。
mask 割り込みを許可するビットマスク。以下の値の論理和を指定する
PEXH321416N_ADC_EXTRG_IN 外部トリガ入力
PEXH321416N_ADC_SMPFINNUM サンプリング終了件数
PEXH321416N_ADC_OVERFLOW オーバーフロー
PEXH321416N_ADC_SCER サンプリングクロックエラー
PEXH321416N_ADC_ATRG アナログトリガ
PEXH321416N_ADC_SP FIFO 割り込み件数
PEXH321416N_ADC_SPS サンプリング開始
PEXH321416N_ADC_SMP サンプリング終了
PEXH321416N_ADC_FF フルスケール検出
PEXH321416N_ADC_EXINTIN 外部割り込み入力
PEXH321416N_ADC_BSY AD 変換終了
PEXH321416N_ADC_TMR インターバルタイマ
PEXH321416N_ADC_ICR_ALL 上記の全て
int pexh321416n_enable_clk_intrrupt(PEXH321416NR *clk,int mask);
戻り値
エラーなら-1 成功なら 0
引数
clk カウンタ制御用メモリ領域へのポインタ。
mask 割り込みを許可するビットマスク 以下の値の論理和を指定する
PEXH321416N_CLK_PERR 異常入力検出での割り込み
PEXH321416N_CLK_C_B キャリー/ボローでの割り込み
PEXH321416N_CLK_EXLT 外部ラッチでの割り込み
PEXH321416N_CLK_EQ カウンタと比較カウンター致での割り込み
PEXH321416N_CLK_ICR_ALL 上記の全て

インターバルタイマ設定
int pexh321416n_set_adc_itimer(PEXH321416NR *adc,unsigned int itimer);
タイマゲート制御時(タイマ無効時)にリセットされる。
I/O 変換方式, FIFO 変換方式共用
戻り値
エラーなら-1 成功なら 0
引数
adc AD 制御用メモリ領域へのポインタ。
itimer 8MHz を文集したインターバルタイマ値 00h～4Fh は設定禁止。

タイマゲート制御を設定する。
int pexh321416n_set_adc_itimer_gate(PEXH321416NR *adc,unsigned int val);
戻り値
エラーなら-1 成功なら 0
引数
adc AD 制御用メモリ領域へのポインタ。
val 下記の何れかの値
PEXH321416N_ADC_ITIMER_GATE_ENABLE
PEXH321416N_ADC_ITIMER_GATE_DISABLE

タイマゲート制御を確認する。
int pexh321416n_get_adc_itimer_gate(PEXH321416NR *adc);
戻り値
エラーなら-1 成功なら PEXH321416N_ADC_ITIMER_GATE_ENABLE あるいは
PEXH321416N_ADC_ITIMER_GATE_DISABLE
引数
adc AD 制御用メモリ領域へのポインタ。

変換モード設定
I/O 変換方式設定中は、FIFO サンプリングはできない。
また、FIFO 変換方式設定中は、I/O 変換方式を行うことはできない。
AD 変換方式を変更する場合は、この関数で変更後、AD 変換を行う必要がある。
int pexh321416n_set_adc_mode(PEXH321416NR *adc,unsigned int mode);

戻り値
エラーなら-1 成功なら 0
引数
adc AD 制御用メモリ領域へのポインタ。
val 下記の何れかの値
PEXH321416N_ADC_CM_FIFO
PEXH321416N_ADC_CM_IO

I/O 変換方式用のチャンネル切り替え+AD 変換スタート関数
int pexh321416n_get_adc_channel_data(PEXH321416NR *adc,int ch,unsigned short int *data);
戻り値
エラーなら-1 成功なら 0
引数
adc AD 制御用メモリ領域へのポインタ。
ch 0 から 15 の何れかのチャンネル番号
data AD 変換値を戻す 16bit 変数へのポインタ

汎用関数 オフセット値を指定して 8 ビットレジスタの値を読み出す
int pexh321416n_get_byte_mmap(PEXH321416NR *dev ,int offset,unsigned long int *value);
戻り値
エラーなら-1 成功なら 0
引数
dev 制御用メモリ領域へのポインタ。
offset レジスタオフセット
オフセットはバイトの位置で指定する。
また、0x40 以上の値はエラーになる。
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 16 ビットレジスタの値を読み出す
int pexh321416n_get_word_mmap(PEXH321416NR *dev ,int offset,unsigned long int *value);
戻り値
エラーなら-1 成功なら 0
引数
dev 制御用メモリ領域へのポインタ。
offset レジスタオフセット
オフセットはバイトの位置で指定する。
また、0x40 以上の値はエラーになる。
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 32 ビットレジスタの値を読み出す
int pexh321416n_get_long_mmap(PEXH321416NR *dev ,int offset,unsigned long int *value);
戻り値
エラーなら-1 成功なら 0
引数
dev 制御用メモリ領域へのポインタ。
offset レジスタオフセット
オフセットはバイトの位置で指定する。
また、0x40 以上の値はエラーになる。
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 8 ビットレジスタに値を書き出す
int pexh321416n_set_byte_mmap(PEXH321416NR *dev ,int offset,unsigned long int *value);
戻り値
エラーなら-1 成功なら 0
引数
dev 制御用メモリ領域へのポインタ。
offset レジスタオフセット
オフセットはバイトの位置で指定する。
また、0x40 以上の値はエラーになる。
value 値を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 16 ビットレジスタに値を書き出す
int pexh321416n_set_word_mmap(PEXH321416NR *dev ,int offset,unsigned long int *value);
戻り値
エラーなら-1 成功なら 0
引数
dev 制御用メモリ領域へのポインタ。
offset レジスタオフセット
オフセットはバイトの位置で指定する。
また、DMA レジスタの存在する 0x20 から 0x3F と 0x80 以上の値はエラーになる。
value 値を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 32 ビットレジスタに値を書き出す
int pexh321416n_set_long_mmap(PEXH321416NR *dev ,int offset,unsigned long int *value);
戻り値
エラーなら-1 成功なら 0
引数
dev 制御用メモリ領域へのポインタ。

offset レジスタオフセット
オフセットはバイトの位置で指定する。
また、0x40 以上の値はエラーになる。
value 値を出す変数へのポインタ

DIO を読み出す
int pexh321416n_read_dio(PEXH321416NR *adc,unsigned char *data)
戻り値
エラーなら-1 成功なら 0
引数
adc AD 制御用メモリ領域へのポインタ。
data 8bit 変数へのポインタ
DIO に出力する
int pexh321416n_write_dio(PEXH321416NR *adc,unsigned char *data);
戻り値
エラーなら-1 成功なら 0
引数
adc AD 制御用メモリ領域へのポインタ。
data 8bit 変数へのポインタ
注意
デジタル入出力は、オフセット PEXH321416N_DIO_OFFSET のデジタル入出力機能が有効(値が 0)でないと機能しない
また、値を書き出して、直後に読み出した場合、値が正しく反映されないため、待ちが必要である。
LP カウンタ部端子状態・入力パルス有効/無効設定状態レジスタ 制御関数
int pexh321416n_clk_read_connector(PEXH321416NR *clk,unsigned char *value8);
int pexh321416n_clk_write_connector(PEXH321416NR *clk,unsigned char *value8);
戻り値
エラーなら-1 成功なら 0
引数
clk クロックカウンタ制御用メモリ領域へのポインタ。
*value 8bit 設定値へのポインタ(下記値の論理和を使用する)
(EN|L1|Z|B|A)
PEXH321416N_CLK_MODE_SETTING_EQS_ENABLE カウンタ値一致検出機能有効
PEXH321416N_CLK_CONNECTER_EN 入力パルス有効(0)/無効(1)
PEXH321416N_CLK_CONNECTER_L1 L1 端子状態(0:Low / 1: Hi)
PEXH321416N_CLK_CONNECTER_Z Z 端子状態(0:Low / 1: Hi)
PEXH321416N_CLK_CONNECTER_B B 端子状態(0:Low / 1: Hi)
PEXH321416N_CLK_CONNECTER_A A 端子状態(0:Low / 1: Hi)

カウンタ制御関数
カウンタ読み出し 32 ビットレジスタを読み込み、32 ビットカウンタへデータをロードします。
int pexh321416n_clk_read_counter(PEXH321416NR *clk,unsigned int *value);
int pexh321416n_clk_write_counter(PEXH321416NR *clk,unsigned int *value);
戻り値
エラーなら-1 成功なら 0
引数
clk クロックカウンタ制御用メモリ領域へのポインタ。
*value 32bit 設定値へのポインタ

カウンタ部デジタルフィルタ設定レジスタ制御関数
int pexh321416n_clk_read_filter(PEXH321416NR *clk,unsigned char *value8);
int pexh321416n_clk_write_filter(PEXH321416NR *clk,unsigned char *value8)
カウンタ部デジタルフィルタ設定レジスタ制御関数
戻り値
エラーなら-1 成功なら 0
引数
clk クロックカウンタ制御用メモリ領域へのポインタ。
*value 8bit 設定値へのポインタ(下記値の論理和を使用する)
PEXH321416N_CLK_MODE_SETTING_EQS_ENABLE カウンタ値一致検出機能有効
*value = (フィルタ基準クロック 2bit|フィルタカウンタ 6bit)
フィルタ基準クロック
PEXH321416N_CLK_DFIL_STOP 下位ビットは、デジタルフィルタのカウンタ
PEXH321416N_CLK_DFIL_1US 下位ビットは、デジタルフィルタのカウンタ
PEXH321416N_CLK_DFIL_10US 下位ビットは、デジタルフィルタのカウンタ
PEXH321416N_CLK_DFIL_100US 下位ビットは、デジタルフィルタのカウンタ

カウンタ部モード設定レジスタ制御関数
int pexh321416n_clk_write_mode(PEXH321416NR *clk,unsigned char *value8);
int pexh321416n_clk_read_mode(PEXH321416NR *clk,unsigned char *value8);
戻り値
エラーなら-1 成功なら 0
引数
clk クロックカウンタ制御用メモリ領域へのポインタ。
*value 8bit 設定値へのポインタ(下記値の論理和を使用する)
(EQS|DIR|MDSEL[0-14])

PEXH321416N_CLK_MODE_SETTING_EQS_ENABLE　カウンタ値一致検出機能有効(1)
PEXH321416N_CLK_MODE_SETTING_DIR_REVERSE　カウント方向を設定リバース方向(1)
PEXH321416N_CLK_MODE_SETTING_DIR_FOWARD　カウント方向を設定通常方向(0)
PEXH321416N_CLK_MODE_SETTING_MDSEL0　　ゲート付き単相パルス 1 逓倍モード
PEXH321416N_CLK_MODE_SETTING_MDSEL1　　ゲート付き単相パルス 2 逓倍モード
PEXH321416N_CLK_MODE_SETTING_MDSEL4　　位相差パルス 1 逓倍モード 非同期クリア ※ 1
PEXH321416N_CLK_MODE_SETTING_MDSEL5　　位相差パルス 2 逓倍モード 非同期クリア ※ 1
PEXH321416N_CLK_MODE_SETTING_MDSEL6　　位相差パルス 4 逓倍モード 非同期クリア ※ 1
PEXH321416N_CLK_MODE_SETTING_MDSEL8　　2 パルスカウントモード*/
PEXH321416N_CLK_MODE_SETTING_MDSEL12　　位相差パルス 1 逓倍モード 同期クリア ※ 2
PEXH321416N_CLK_MODE_SETTING_MDSEL13　　位相差パルス 2 逓倍モード 同期クリア ※ 2
PEXH321416N_CLK_MODE_SETTING_MDSEL14　　位相差パルス 4 逓倍モード 同期クリア ※ 2
※ 1 非同期クリア:
A 相, B 相のステータスに関わりなく、Z 相または Z 相+L1 が有効になった時、カウンタはクリアされます。
※ 2 同期クリア:
B 相が“Low”で、Z 相または Z 相+L1 が有効になった時、A 相の(↑)と(↓)にてカウンタはクリアされます

カウンタ部ステータスレジスタ 制御関数
int pexh321416n_clk_read_status(PEXH321416NR *clk,unsigned char *value8);
カウンタ部ステータスレジスタ 制御関数
戻り値
エラーなら-1 成功なら 0
引数
clk　　クロックカウンタ制御用メモリ領域へのポインタ。
*value　　8bit 設定値へのポインタ(下記値の論理和を使用する)
(PERR|EQF|EXLTS|EQ|CBF|UD)
PEXH321416N_CLK_STATUS_PERR　　位相差パルスカウントモードでの異常入力検出です。値をリードするとクリアされます。
PEXH321416N_CLK_STATUS_EQF　　カウンタ値一致。値をリードするとクリアされます
PEXH321416N_CLK_STATUS_EXLTS　　外部ラッチ。値をリードするとクリアされます
PEXH321416N_CLK_STATUS_EQ　　カウンタと比較カウンタが一致しているかどうかを表します。
PEXH321416N_CLK_STATUS_CBF　　キャリー/ボロー。値をリードするとクリアされます
PEXH321416N_CLK_STATUS_UD　　現在のカウンタの方向を表します。0:ダウンカウント 1:アップカウント

カウンタ部ソフトウェアラッチクリアレジスタ 制御関数
int pexh321416n_clk_write_status(PEXH321416NR *clk,unsigned char *value8);
戻り値
エラーなら-1 成功なら 0
引数
clk　　クロックカウンタ制御用メモリ領域へのポインタ。
*value　　8bit 設定値へのポインタ(下記値の何れかを使用する)
PEXH321416N_CLK_STATUS_SOFT_LATCH　　カウンタ値をリードし、レジスタに移動する。(ソフトウェアラッチ)
PEXH321416N_CLK_STATUS_SOFT_CLEAR　　カウンタ,リードレジスタをクリアする。(ソフトウェアクリア)

カウンタ部外部ラッチ, クリア, Z 相論理設定レジスタ制御関数
int pexh321416n_clk_write_lpzpltscls(PEXH321416NR *clk,unsigned char *value8);
戻り値
エラーなら-1 成功なら 0
引数
clk　　クロックカウンタ制御用メモリ領域へのポインタ。
*value　　8bit 設定値へのポインタ(下記値の論理和を使用する)
(LP|ZP|LTS[012]|CLS[012])
PEXH321416N_CLK_LPZPLTSCLS_LP_RISE　L 論理設定 L の立ち上がりを有効(“High”で有効)
PEXH321416N_CLK_LPZPLTSCLS_LP_FALL　L 論理設定 L の立ち下がりを有効(“Low”で有効)
PEXH321416N_CLK_LPZPLTSCLS_ZP_HIGH　Z 相論理設定 Z 相反転(“Low”で有効)
PEXH321416N_CLK_LPZPLTSCLS_ZP_LOW　Z 相論理設定 Z 相通常(“High”で有効)
PEXH321416N_CLK_LPZPLTSCLS_LTS0　外部ラッチ設定 カウンタの外部ラッチを行わない。
PEXH321416N_CLK_LPZPLTSCLS_LTS1　外部ラッチ設定 L のみでカウンタをラッチする。
PEXH321416N_CLK_LPZPLTSCLS_LTS2　外部ラッチ設定 L と Z 相が有効の時、カウンタをラッチする。
PEXH321416N_CLK_LPZPLTSCLS_CLS0　外部クリア設定 カウンタの外部クリアを行わない。
PEXH321416N_CLK_LPZPLTSCLS_CLS1　外部クリア設定 Z 相のみでカウンタをクリアする。
PEXH321416N_CLK_LPZPLTSCLS_CLS2　外部クリア設定 L と Z 相が有効の時、カウンタをクリアする。

使用例
カウンタ値の取得方法
3 モードパルスカウンタインタフェースモジュールのカウンタ値を読み込みます。
(1)カウントモードの設定を行います。
カウンタ/デジタル入出力選択
カウンタ機能有効に設定
value8=PEXH321416N_CLK_SELECT_COUNTER;
pexh321416n_set_byte_mmap(clk,PEXH321416N_CLK_SELECT_OFFSET,&value8);
カウンタ部モード設定
位相差パルス, 4 逓倍モード, 非同期クリアに設定
value8=PEXH321416N_CLK_MODE_SETTING_MDSEL6;

```
            pexh321416n_clk_write_mode(clk,&value8);
        カウンタ部ソフトウェアラッチ/クリア
            カウンタクリアコマンドを発行
            value8=PEXH321416N_CLK_STATUS_SOFT_CLEAR;
            pexh321416n_clk_write_status(clk,&value8);
    (2)パルスの入力を開始します。
    (3)カウンタ値を取得したいタイミングでカウンタラッチコマンドを発行します。
            カウンタラッチコマンドを発行
            value=PEXH321416N_CLK_STATUS_SOFT_LATCH;
            pexh321416n_clk_write_status(clk,&value);
    (4)カウンタ値の取得を行います。
            pexh321416n_clk_read_counter(clk,&value32);
    (5)取得したカウンタ値の表示を行います。
            printf("0x08X0,value32);

カウンタ値の設定方法
    3 モードパルスカウンタインタフェースモジュールへカウンタ値を書き込みます。
    (1)カウントモードの設定を行います。
            カウンタ機能有効に設定
                value8=PEXH321416N_CLK_SELECT_COUNTER;
                pexh321416n_set_byte_mmap(clk,PEXH321416N_CLK_SELECT_OFFSET,&value8);
            カウンタ部モード設定
            位相差パルス, 4 逓倍モード, 非同期クリアに設定
            value8=PEXH321416N_CLK_MODE_SETTING_MDSEL6;
            pexh321416n_clk_write_mode(clk,&value8);
    (2)書き込みを行うカウンタを選択します。
            書き込み先にカウンタプリレジスタを指定
            value=PEXH321416N_CLK_CONNECTER_INPUT_COUNT;
            pexh321416n_clk_write_connector(clk,&value);
    (3)カウンタ値を設定します。
            value32 = カウンタ値;
            pexh321416n_clk_write_counter(clk,&value);

比較カウンタ設定方法
    3 モードパルスカウンタインタフェースモジュールへカウンタ値を書き込みます。
    (1)カウントモードの設定を行います。
            カウンタ機能有効に設定
                value8=PEXH321416N_CLK_SELECT_COUNTER;
                pexh321416n_set_byte_mmap(clk,PEXH321416N_CLK_SELECT_OFFSET,&value8);
            カウンタ部モード設定
            位相差パルス, 4 逓倍モード, 非同期クリアに設定
            value8=PEXH321416N_CLK_MODE_SETTING_MDSEL6;
            pexh321416n_clk_write_mode(clk,&value8);
    (2)書き込みを行うカウンタを選択します。
            書き込み先に比較レジスタを指定
            value=PEXH321416N_CLK_CONNECTER_INPUT_COMPARE;
            pexh321416n_clk_write_connector(clk,&value);
    (3)比較カウンタ値を設定します。
            value32 = 比較カウンタ値;
            pexh321416n_clk_write_counter(clk,&value);

割り込み処理設定方法
割り込み処理を行う場合、以下の処理を行う必要があります。
シグナルハンドラーの登録
    pexh321416n_setup_signal(fd,pexh321416n_interrupt_handler);

シグナルハンドラー定義
    static void pexh321416n_interrupt_handler(int signo,siginfo_t *info, void *ptr)
    {
        unsigned int iflag;

        /*割り込みが発生すると、割り込みフラグレジスタの各ビットが ON になります。このフラグで発生した割り込みの要因を確認します。
*/
        pexh321416n_intr_service(pexh321416n_fd,&iflag,&pexh321416n_pending);
            adc_iflag = iflag & 0xFFFF;
            clk_iflag = (iflag>>16) & 0xFF;
    }
シグナル番号の確認、ディフォルトでは SIGIO。
    ioctl(fd,IOCTL_EXTMEM_GET_SIGNAL_NUMBER,&signo);
シグナルマスクの設定
      sigfillset(&imask);
      sigdelset(&imask,SIGINT);
      sigdelset(&imask,signo);
割り込みの許可
    pexh321416n_enable_clk_intrrupt(clk,PEXH321416N_CLK_ICR_ALL);
PEXH321416N_CLK_IFR_ALL 割り込みシグナルの待ちと、要因別処理
    sigsuspend(&imask);
```

```
        if (clk_iflag & PEXH321416N_CLK_PERR)
          /* 異常入力検出での割り込み 0:割り込み未検出 1:割り込み検出*/
        if (clk_iflag & PEXH321416N_CLK_C_B)
          /* キャリー/ボローでの割り込み 0:割り込み未検出 1:割り込み検出*/
        if (clk_iflag & PEXH321416N_CLK_EXLT)
          /* 外部ラッチでの割り込み 0:割り込み未検出 1:割り込み検出*/
        if (clk_iflag & PEXH321416N_CLK_EQ)
          /* カウンタと比較カウンター致での割り込み 0:割り込み未検出 1:割り込み検出*/
    割り込みの禁止
        pexh321416n_disable_clk_intrrupt(clk);

SEE ALSO
    /usr/local/CNC/drivers/extmem/interface/pexh321416n 下のプログラム

AUTHORS
    Copyright (C) 1995-2025 Concurrent Real Time Inc.

                                                    28 Feb 2025
pexh321416n(3)
```

6. インタフェースモジュール制御レジスタ情報

インタフェースモジュール制御レジスタは、インタフェース社よりご入手ください。
なお、PEX-H321416N の公開資料はありませんが、PEX-H321416N と PEX-361416 は AD 制御とカウンタ制御のレジスタマップは共通ですので、PEX-361416 の IO 公開資料を入手ください。
但し、PEX-H321416N には DA 機能はございませんので DA 制御のレジスタは使用できません。

なお下記の違いがございます。

デバイス ID
PEX-361416　:0E1Eh
PEX-H321416N：0C8Eh

PEX-H321416N の AD 制御用の電子ボリューム調整レジスタ（ベースアドレス 0　+3CH～+3FH）が無くなっています

DIO の入出力制御は AD の BASE0 にあります。

デジタル入力レジスタ：Offset :+38h
デジタル出力レジスタ：Offset :+38h

「AD 変換モード設定」の部分に記載されている内容については
次の様に訂正させていただきます。

・AD 変換モード設定
（ソフトスタート、指定件数終了）
（ADR+2Ch）に”40h”を設定