PCI-3168C 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 PCI- 3168C PCI 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

PCI- 3168C 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。

- PCI- 3168C ボード
- RedHawk 6.x 以上
- Extmem version 8.3 以上

PCI-3168Cは、PCIバスに準拠したシングルエンド32チャンネル,差動16チャンネルの 12ビットAD変換製品です。

3. インストール方法

PCI-3168C 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。

PCI-3168C 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
PCI3168C BSP 実行パッケージのインストール
rpm -ivh bin-pci3168c -**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-pci3168c -**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

PCI3168C BSP 実行パッケージのインストール
apt install ./bin-pci3168c -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-pci3168c -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

PCI3168C BSP 実行パッケージのインストール
apt install ./bin-pci3168c -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-pci3168c -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。)

PCI-3168C BSP パッケージは ***/usr/local/CNC/drivers/extmem/interface/pci3168c*** 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

PCI-3168C 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/pci3168c 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-pci3168c -**X.Y**_RH***x.y-z***.x86_64 *(開発パッケージの削除)*
rpm -e bin-pci3168c -**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-pci3168c -**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-pci3168c -rhx.y *(開発パッケージの削除)*
apt purge bin-pci3168c -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-pci3168c -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-pci3168c -rhx.y *(開発パッケージの削除)*
apt purge bin-pci3168c -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-pci3168c -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 pci3168c
pci3168c(3) pci3168c(3)

NAME
pci3168c - 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
pci3168c は、external memory ドライバを利用した pci3168c ボードアクセスライブラリです。

#include <sys/pci3168c.h>
gcc [options ...] file -lpci3168c -lextmem ...

****
PCI3168C
****

DIP スイッチの読み込み
int pci3168c_get_sw(int fd,unsigned 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
int pci3168c_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, siginfo_t *, void *) 割り込みハンドラ
mask 割り込みを許可するビットマスク 以下のいずれかを指定する
PCI3168C_IMASK_TMR インターバルタイマー
PCI3168C_IMASK_BSY AD 変換終了割り込み
PCI3168C_IMASK_TRG 外部割り込み(EXINT IN)
PCI3168C_IMASK_ALL
(PCI3168C_IMASK_TMR|PCI3168C_IMASK_BSY|PCI3168C_IMASK_TRG)

デバイスの非初期化処理
int pci3168c_reset(int fd);
int pci3168c_uninit(int fd);　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
　2つの関数は同じ処理、複数枚同期サンプリングを使用 し な い,TRIG NONE,GATE OFF,割り込み禁止を設定する

デバイスの初期化処理
int pci3168c_init
　(
　　int fd,
　　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
　　BAR0 I/O Region addr 0x00004480 offset 0x00000000 32 bytes addr 0x00004480 offset 0x00000000 16 bytes Switch 1

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 0x0D) の値を戻す
int pci3168c_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 り込んだ際の割り込み要因レジスタ(オフセット 0x0D)の値を戻し、AD データを読み込む
関数とペアで使用する（下記使用例を参照）
int pci3168c_intr_service_and_read
　(
　　int fd,
　　unsigned int *iflag,
　　int *pending
　　unsigned short 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
　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
data AD 値を戻す変数

割り込みを禁止する

```
int pci3168c_disable_intrrupt
        (
            int fd,
            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
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
PCI3168C_IMASK_TMR インターバルタイマー
PCI3168C_IMASK_BSY AD 変換終了割り込み
PCI3168C_IMASK_TRG 外部割り込み(EXINT IN)
PCI3168C_IMASK_ALL
(PCI3168C_IMASK_TMR|PCI3168C_IMASK_BSY|PCI3168C_IMASK_TRG)

割り込みを許可する

```
int pci3168c_enable_intrrupt
        (
            int fd,
            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
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
PCI3168C_IMASK_TMR インターバルタイマー
PCI3168C_IMASK_BSY AD 変換終了割り込み
PCI3168C_IMASK_TRG 外部割り込み(EXINT IN)
PCI3168C_IMASK_ALL
(PCI3168C_IMASK_TMR|PCI3168C_IMASK_BSY|PCI3168C_IMASK_TRG)

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  pci3168c_get_ioport(int  fd,int   base,int   offset,unsigned   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
base レジスタベース PCIBAR0 か PCIBAR1 を指定する
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   pci3168c_set_ioport(int   fd,int   base,int  offset,unsigned  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

base　レジスタベース PCIBAR0 か PCIBAR1 を指定する
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(変換終了フラグをポーリングする)
int pci3168c_read_data_poll(int fd,int ch,unsigned short 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
ch チャンネル
data　入力変数へのポインタ

入力データを読み出す
int pci3168c_read_data(int fd,unsigned short 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

チャネル切り替え＋AD 変換開始
int pci3168c_set_channel(int fd,int ch);
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
ch チャンネル

同期サンプリング設定
int pci3168c_set_sync(int fd,unsigned 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　以下のいずれかを指定する
PCI3168C_SYNC_NORMAL 複数枚同期サンプリングを使用しない場合
(同期信号はスルーされる)
PCI3168C_SYNC_MASTER 複数枚同期サンプリングを使用する
(同期信号を出力するマスターになる)
PCI3168C_SYNC_SLAVE　複数枚同期サンプリングを使用する
(同期信号を入力するスレーブになる)

割り込みのトリガー設定
int pci3168c_set_trigger(int fd,unsigned int trigger);
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
trigger　以下のいずれかを指定する
PCI3168C_TRIG_TMST タイマーによる AD 変換スタート有効
PCI3168C_TRIG_EXTRG EXTRG IN 入力有効
PCI3168C_TRIG_NONE なし

AD 変換タイマーをセットする
int pci3168c_set_convert_timer(int fd,unsigned int div1,unsigned int div2);

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

div1 8MHz のベースクロックを分周する値 カウントダウンする
最大 65535 分周しかできない

div2 div1 の出力を分周する値 カウントダウンし 0 の時割り込みが発生する
最大 65535 分周しかできない

この間数を呼び出すと割り込みのロジックが変更になり、割り込み処理でデータを読み出すため、
割り込み関数では、pci3168c_intr_service_and_read()を必ず使用すること

AD 変換タイマーをスタートまたは停止する
int pci3168c_set_gate(int fd,unsigned int on);

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

on 以下のいずれかを指定する

PCI3168C_GATE_ON 0x01 // タイマーイネーブル

PCI3168C_GATE_OFF 0x00 //タイマーディセーブル

AD 変換のレンジをチャンネル指定して設定する/読み出す
int pci3168c_set_channel_range(int fd,int ch,unsigned long int range);
int pci3168c_get_channel_range(int fd,int ch,unsigned long int *range);

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

ch チャンネル

range 以下のいずれかを指定する

PCI3168C_RANGE_SETTING_M5TOP5 -5V から +5V

PCI3168C_RANGE_SETTING_M10TOP10 -10V から +10V

PCI3168C_RANGE_SETTING_M25TOP25 -2.5V から +2.5V

PCI3168C_RANGE_SETTING_0TOP5 0V から +5V

PCI3168C_RANGE_SETTING_0TOP10 0V から +10V

AD 変換のレンジをチャンネル指定して読み出 す int pci3168c_set_mode(int fd,unsigned long int mode);
int pci3168c_get_mode(int fd,unsigned long int *mode);

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

ch チャンネル

mode 以下のいずれか

PCI3168C_AI_MODE_SINGLEEND	シングルエンド
PCI3168C_AI_MODE_DIFFERNTIAL	デファレン シアル
PCI3168C_AI_MODE_GROUND	(0V)
PCI3168C_AI_MODE_PLUS5	(+5V)
PCI3168C_AI_MODE_MINUS5	(-5V)

SEE ALSO

/usr/local/CNC/drivers/extmem/interface/pci3168c 下のプログラム

AUTHORS