

PCI-3341A Board Support Package Installation on RedHawk

Release Notes Revision B

September 9, 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 PCI- 3341A 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- 3341A 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。

- PCI- 3341A ボード
- RedHawk 6.x 以上
- Extmem version 8.3 以上

PCI-3341Aは、PCIバスに準拠した、12ビットDA変換機能を持った製品です。

3. インストール方法

PCI-3341A 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-3341A 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。:

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_RHx.y-z.x86_64.rpm  
PCI3341A BSP 実行パッケージのインストール  
# rpm -ivh bin-pci3341a -X.Y_RHx.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-pci3341a -X.Y_RHx.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
```

PCI3341A BSP 実行パッケージのインストール

```
# apt install ./bin-pci3341a -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-pci3341a -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
```

PCI3341A BSP 実行パッケージのインストール

```
# apt install ./bin-pci3341a-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-pci3341a-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-3341A BSP パッケージは `/usr/local/CNC/drivers/extmem/interface/pci3341a` 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-3341A 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/pci3341a` 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-pci3341a -X.Y_RHx.y-z.x86_64 (開発パッケージの削除)  
# rpm -e bin-pci3341a -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-pci3341a -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-pci3341a-rhx.y (開発パッケージの削除)  
# apt purge bin-pci3341a-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-pci3341a-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-pci3341a-rhx.y (開発パッケージの削除)  
# apt purge bin-pci3341a-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-pci3341a-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 pci3341a  
pci3341a(3)
```

pci3341a(3)

NAME

pci3341a - 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

pci3341a は、external memory ドライバを利用した pci3341a ボードアクセスライブラリです。

```
#include <sys/pci3341a.h>
gcc [options ...] file -lpci3341a -lxtmem ...
```

```
*****
PCI3341A
*****
```

DIP スイッチの読み込み

```
int pci3341a_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 pci3341a_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

PCI3341A_IMASK_TMR インターバルタイマー

PCI3341A_IMASK_SPS DA 変換開始割り込み

PCI3341A_IMASK_TRG 外部割り込み(EXINT IN)

PCI3341A_IMASK_ALL

(PCI3341A_IMASK_TMR|PCI3341A_IMASK_TRG|PCI3341A_IMASK_SPS)

デバイスの非初期化処理

```
int pci3341a_reset(int fd);
```

```
int pci3341a_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、全ての制御レジスタに 0 値を設定する。

デバイスの初期化処理

```
int pci3341a_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 0x0000a8c0 offset 0x00000000 32 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 0x0D)の値を戻す

```

int pci3341a_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 pci3341a_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
        PCI3341A_IMASK_TMR インターバルタイマー
        PCI3341A_IMASK_SPS DA 変換開始割り込み
        PCI3341A_IMASK_TRG 外部割り込み(EXINT IN)
        PCI3341A_IMASK_ALL
        (PCI3341A_IMASK_TMR|PCI3341A_IMASK_TRG|PCI3341A_IMASK_SPS)

```

割り込みを許可する

```

int pci3341a_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
        PCI3341A_IMASK_TMR インターバルタイマー

```

PCI3341A_IMASK_SPS DA 変換開始割り込み
PCI3341A_IMASK_TRG 外部割り込み(EXINT IN)

PCI3341A_IMASK_ALL
(PCI3341A_IMASK_TMR|PCI3341A_IMASK_TRG|PCI3341A_IMASK_SPS)

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 pci3341a_get_ioport(int fd,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

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 pci3341a_set_ioport(int fd,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

offset レジスタオフセット

value 値を出す変数へのポインタ

データを同時出力モードで出力する

int pci3341a_write_data_all(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

ch チャンネル

data 出力変数へのポインタ

チャンネルを指定してデータを出力する

int pci3341a_write_data(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 出力データ

出力データを読み出す

int pci3341a_read_data(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 出力変数へのポインタ

チャンネル切り替え

int pci3341a_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 pci3341a_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
PCI3341A_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)
PCI3341A_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)
PCI3341A_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 pci3341a_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
PCI3341A_TRIG_EXINT EXINT IN 入力(立ち上がりエッジ:1/立ち下がりエッジ:0)有効
PCI3341A_TRIG_NONE なし

DA 変換タイマーをセットする

int pci3341a_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

DA 変換タイマーをスタートまたは停止する

int pci3341a_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
PCI3341A_GATE_ON 0x01 //タイマーイネーブル
PCI3341A_GATE_OFF 0x00 //タイマーディセーブル

SEE ALSO

/usr/local/CNC/drivers/extmem/interface/pci3341a 下のプログラム

AUTHORS

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

28 Apr 2016

pci3341a(3)