

# PCI-2994CV 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- 2994cv 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- 2994CV 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- 2994CV ボード
- RedHawk 6.x 以上
- Extmem version 8.3 以上

**PCI-2994CV**は、PCIバスに準拠した、フォトカプラ型高耐圧接点入力の64点デジタル 信号入力と、高電流オープンコレクタシンク型出力64点のデジタル信号出力を持つ デジタル入出力共用製品です。

## 3. インストール方法

**PCI-2994CV 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-2994CV 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_RHx.y-z.x86_64.rpm
PCI2826CV BSP 実行パッケージのインストール
# rpm -ivh bin-pci2826cv -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-pci2826cv -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
```

```
PCI2826CV BSP 実行パッケージのインストール
# apt install ./bin-pci2826cv -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-pci2826cv -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
```

#### **PCI2826CV BSP 実行パッケージのインストール**

```
# apt install ./bin-pci2826cv-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-pci2826cv-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-2994CV BSP** パッケージは `/usr/local/CNC/drivers/extmem/interface/pci2994cv` 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-2994CV 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/pci2994cv` ディレクトリは削除されます。

5. x86\_64 アーキテクチャの場合
6. === root ユーザで実行してください ===
7. 開発パッケージをインストールしていた場合には、
8. # rpm -e dev-pci2826cv -X.Y\_RHx.y-z.x86\_64 (開発パッケージの削除)
9. # rpm -e bin-pci2826cv -X.Y\_RHx.y-z.x86\_64 (実行パッケージの削除)
10. 実行パッケージのみをインストールしていた場合には、
11. # rpm -e bin-pci2826cv -X.Y\_RHx.y-z.x86\_64 (実行パッケージの削除)
- 12.
13. amd64 アーキテクチャの場合
14. === root ユーザで実行してください ===
15. 開発パッケージをインストールしていた場合には、
16. # apt purge dev-pci2826cv -rhx.y (開発パッケージの削除)
17. # apt purge bin-pci2826cv -rhx.y (実行パッケージの削除)
18. 実行パッケージのみをインストールしていた場合には、
19. # apt purge bin-pci2826cv -rhx.y (実行パッケージの削除)
- 20.
21. arm64 アーキテクチャの場合
22. === root ユーザで実行してください ===
23. 開発パッケージをインストールしていた場合には、
24. # apt purge dev-pci2826cv -rhx.y (開発パッケージの削除)
25. # apt purge bin-pci2826cv -rhx.y (実行パッケージの削除)
26. 実行パッケージのみをインストールしていた場合には、
27. # apt purge bin-pci2826cv -rhx.y (実行パッケージの削除)

## 28. 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 pci2994cv
```

```
pci2994cv(3)
```

```
pci2994cv(3)
```

NAME

pci2994cv - 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

pci2994cv は、external memory ドライバを利用した pci2994cv ボードアクセスライブラリです。

```
#include <sys/pci2994cv.h>
gcc [options ...] file -lpci2994cv -lxtmem ...
```

```
*****
PCI2994CV
*****
```

### 割り込みハンドラの登録

```
int pci2994cv_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マスク値

### デバイスの非初期化処理

```
int pci2994cv_reset(int fd);
```

```
int pci2994cv_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 pci2994cv_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 16 bytes

Switch 1

### pci2994cv の出力を発生させる

```
int pci2994cv_raise_signal
```

```
(
    int fd,
```

```

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
ack,out1,out2 割り込みの種類 以下のいずれかを指定する
ack1
PCI2994CV_PULS_ACK1_NOACTION なにもしない
PCI2994CV_PULS_ACK1_LOW_HIGH ACK1 端子をクリア Low->High
PCI2994CV_PULS_ACK1_HIGH_LOW ACK1 端子をクリア High->Low
out1
PCI2994CV_PULS_OUT1_NOACTION なにもしない
PCI2994CV_PULS_OUT1_LEVEL_HIGH High レベル出力
PCI2994CV_PULS_OUT1_LEVEL_LOW Low レベル出力
PCI2994CV_PULS_OUT1_PULSE_LOW Low パルスを出力
out2
PCI2994CV_PULS_OUT2_NOACTION なにもしない
PCI2994CV_PULS_OUT2_LEVEL_HIGH High レベル出力
PCI2994CV_PULS_OUT2_LEVEL_LOW Low レベル出力
PCI2994CV_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 pci2994cv_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もとに戻す変数  
割り込みを禁止する

```

int pci2994cv_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  
PCI2994CV\_IMASK\_STB1 STB1 からの入力信号  
PCI2994CV\_IMASK\_IRIN1 IR.IN1 からの入力信号  
PCI2994CV\_IMASK\_IRIN2 IR.IN2 からの入力信号  
PCI2994CV\_IMASK\_TIMER タイマー割り込み  
PCI2994CV\_IMASK\_RESET リセット割り込み

割り込みを許可する

```
int pci2994cv_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

PCI2994CV\_IMASK\_STB1 STB1 からの入力信号

PCI2994CV\_IMASK\_IRIN1 IR.IN1 からの入力信号

PCI2994CV\_IMASK\_IRIN2 IR.IN2 からの入力信号

PCI2994CV\_IMASK\_TIMER タイマー割り込み

PCI2994CV\_IMASK\_RESET リセット割り込み

PCI2994CV\_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 pci2994cv_set_interval_timer(int fd,unsigned int base,unsigned 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

base ベースクロック値 以下のいずれかを指定する

PCI2994CV\_TIMER\_BASE\_STOP 停止

PCI2994CV\_TIMER\_BASE\_010USEC 10 マイクロ秒

PCI2994CV\_TIMER\_BASE\_100USEC 100 マイクロ秒

PCI2994CV\_TIMER\_BASE\_001MSEC 1 ミリ秒

PCI2994CV\_TIMER\_BASE\_010MSEC 10 ミリ秒

PCI2994CV\_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 pci2994cv_get_interval_timer(int fd,unsigned 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

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 pci2994cv_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 pci2994cv_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

チャンネルを指定して入力データを読み出す

```
int pci2994cv_read_data(int fd,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  
ch チャンネル  
data 値を出す変数へのポインタ

すべてのチャンネルの入力データを読み出す

```
int pci2994cv_read_data_all(int fd,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  
data 値を出す配列変数へのポインタ

pci2994cv.3.UTF8

使用例

```
int pci2994cv_write_data(int fd,unsigned long 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  
ch チャンネル  
data 出力変数へのポインタ

すべてのチャンネルのデータを出力する

```
int pci2994cv_write_data_all(int fd,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  
data 出力配列変数へのポインタ

DIP スイッチの読み込み

```
int pci2994cv_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

```

*****/
/*
*/
/* init.c -- PCI-2994cv */
/*
*/
/*****/
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <memory.h>
#include <sys/mman.h>
#include <sys/param.h>
#include <sys/file.h>
#include <sys/types.h>
#include <sys/extmem.h>
#include <signal.h>
#include <stdlib.h>
#include <sys/libinterface.h>

int
main(argc, argv)
    int argc;
    char **argv;
{
    int fd;
    char devname[1024];
    int mem_size;

    if ( argc < 2 ) {
        strcpy(devname, "/dev/pci2994cv/0");
    } else {
        strcpy(devname, argv[1]);
    }
    if ((fd = open(devname, O_RDWR)) == -1)
    {
        fprintf(stderr, "Device not found %s(%s)\n",
            devname, strerror(errno));
        exit(0);
    }
    printf("%s ", devname);
    if (pci2994cv_init(fd, 1) < 0)
    {
        fprintf(stderr, "Device initialize error %s(%s)\n",
            devname, strerror(errno));
        exit(0);
    }
    if (pci2994cv_uninit(fd) < 0)
    {
        fprintf(stderr, "%s\n", strerror(errno));
    }
    close(fd);
}

```

SEE ALSO

/usr/local/CNC/drivers/extmem/interface/pci2994cv 下のプログラム

AUTHORS

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

pci2994cv(3)	10 Jul 2016	pci2994cv(3)	10 Jul 2014
--------------	-------------	--------------	-------------