PCI-3126 Board Support Package Installation on RedHawk

Release Notes

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-3126 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- 3126 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- 3126 ボード
- RedHawk 6.x 以上
- Extmem version 8.3 以上

PCI-3126は、PCIバスに準拠した、電圧または電流を入力可能なチャンネル間独立 絶縁型12ビットAD変換製品です。

3. インストール方法

PCI-3126 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-3126 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

PCI3126 BSP 実行パッケージのインストール

rpm -ivh bin-pci3126 -**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-pci3126 -**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

PCI3126 BSP 実行パッケージのインストール

apt install ./bin-pci3126 -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-pci3126 -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

PCI3126 BSP 実行パッケージのインストール
apt install ./bin-pci3126 -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-pci3126 -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-3126 BSP パッケージは ***/usr/local/CNC/drivers/extmem/interface/pci3126*** 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-3126 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/pci3126 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-pci3126 -**X.Y**_RH***x.y-z***.x86_64 *(開発パッケージの削除)*
rpm -e bin-pci3126 -**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-pci3126 -**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-pci3126 -rhx.y *(開発パッケージの削除)*
apt purge bin-pci3126 -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-pci3126 -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-pci3126 -rhx.y *(開発パッケージの削除)*
apt purge bin-pci3126 -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-pci3126 -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 pci3126
pci3126(3) pci3126(3)

NAME

pci3126 - 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

pci3126 は、external memory ドライバを利用した pci3126 ボードアクセスライブラリです。

```
#include <sys/pci3126.h>
gcc [options ...] file -lpci3126-lextmem ...
```

****

PCI3126

****

DIP スイッチの読み込み

```
int pci3126_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

割り込みハンドラの登録

```
int pci3126_setup_signal
        (
            int fd,
            void (*interrupt_hadler)( 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_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

PCI3126_IMASK_TMR インターバルタイマー

PCI3126_IMASK_BSY AD 変換終了割り込み

PCI3126_IMASK_TRG 外部割り込み(EXINT IN)

PCI3126_IMASK_ALL (PCI3126_IMASK_TMR|PCI3126_IMASK_BSY|PCI3126_IMASK_TRG)

デバイスの非初期化処理

```
int pci3126_reset(int fd);
int pci3126_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 pci3126_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

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 pci3126_intr_service
(
int fd,
unsigned long int *iflag,
unsigned long 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 pci3126_intr_service_and_read
(
int fd,
unsigned long int *iflag,
unsigned long 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(最上位ビットは変換終了フラグ)
割り込みを禁止する

int pci3126_disable_intrrupt
(
int fd,
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

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
PCI3126_IMASK_TMR インターバルタイマー
PCI3126_IMASK_BSY AD 変換終了割り込み
PCI3126_IMASK_TRG 外部割り込み(EXINT IN)
PCI3126_IMASK_ALL
(PCI3126_IMASK_TMR|PCI3126_IMASK_BSY|PCI3126_IMASK_TRG)

割り込みを許可する
int pci3126_enable_intrrupt
(
int fd,
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
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
PCI3126_IMASK_TMR インターバルタイマー
PCI3126_IMASK_BSY AD 変換終了割り込み
PCI3126_IMASK_TRG 外部割り込み(EXINT IN)
PCI3126_IMASK_ALL
(PCI3126_IMASK_TMR|PCI3126_IMASK_BSY|PCI3126_IMASK_TRG)

汎用関数 Base Address 0 のオフセット値を指定してレジスタの値を読み出す
int pci3126_get_ioport(int fd,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
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 Base Address 0 のオフセット値を指定してレジスタに値を書き出す
int pci3126_set_ioport(int fd,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
offset レジスタオフセット
value 値を出す変数へのポインタ

汎用関数 Base Address 1 のオフセット値を指定してレジスタの値を読み出す
int pci3126_get_ioport2(int fd,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
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 Base Address 1 のオフセット値を指定してレジスタに値を書き出す
int pci3126_set_ioport2(int fd,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
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 pci3126_read_data_poll(int fd,unsigned long 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 pci3126_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 pci3126_set_channel(int fd,unsigned long 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 pci3126_set_sync(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 以下のいずれかを指定する
PCI3126_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)
PCI3126_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)
PCI3126_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 pci3126_set_trigger(int fd,unsigned long 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
PCI3126_EXTTRIG_ON 外部トリガ有効(タイマ無効)
PCI3126_EXTTRIG_OFF タイマ有効(外部トリガ無効)

PCI3126_EXTTRIG_BAN 設定禁止
PCI3126_TRIG_NONE 外部トリガ、タイマ両方無効

極性の設定
int pci3126_set_polarity(int fd,unsigned long int polarity);
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
polarity 以下を指定する
PCI3126_POLARITY_EXTG EXTRG IN 入力(0:立下りエッジ/1: 立上りエッジ)有効
PCI3126_POLARITY_EINT1 EXINT IN 入力(0:立下りエッジ/1:立上りエッジ)有効
PCI3126_POLARITY_DITG DI トリガ機能(0:無効/1:有効)

AD 変換タイマーをセットする
int pci3126_set_convert_timer(int fd,unsigned long int div1,unsigned long 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、pci3126_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 pci3126_set_gate(int fd,unsigned long 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
PCI3126_GATE_ON 0x01 //タイマーイネーブル
PCI3126_GATE_OFF 0x00 //タイマーディセーブル
使用例

```
#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>

#define MAXDATA     1600
int  fd;
int count=0;
int error=0;
unsigned short DATA[MAXDATA];

static void pci3126_interrupt_handler(int signo)
{
          unsigned long int pending=0;
          unsigned long int iflag=0;
        static int ch=0;

        pci3126_intr_service_and_read(fd ,&iflag,&pending,&DATA[count]);
        if ((DATA[count]&0x8000)==0)
        {
                error++;
                DATA[count]=0xFFFF;
        }
        ch ++;ch %= 8;
        pci3126_set_channel(fd,ch);
        if (count<MAXDATA) count++;
}

int
main(int argc, char **argv)
{
        char devname[1024];
        int mem_size;
        unsigned long int ch,value;
        static int cycle=1000,Us;
        unsigned short int data;
        int i;

          extern int optind,errno;
          extern char *optarg;
          int     c;

          while((c = getopt(argc, argv, "s:")) != EOF)
        {
                    switch(c) {
                              case 's':
                                        cycle = atoi( optarg ) ;
                                        if (( cycle <= 0 )||((cycle)>1666))
                                        {
                                                  fprintf(stderr,"INVALID          ARGU-
MENT!!0) ;
                                                  exit( 0 ) ;
                                        }
                              break;
                              default:
                                        fprintf(stderr,               "Usage:          %s
```

```
[options]0,argv[0]);
                                        fprintf(stderr, "Options:0);
                                        fprintf(stderr,"          -s  msec(msec: 0  or
1<=cycle<=16)0);
                                        exit(-1);
                        }
          }
          if (optind == argc) {
                        /* no more option */
                        strcpy(devname,"/dev/pci3126/0");
          } else {
                        strcpy(devname,argv[optind]);
          }

          if ((fd = open(devname,O_RDWR)) == -1)
        {
                fprintf(stderr,"Device not found %s(%s)0,
                devname,strerror(errno));
                        exit(0);
        }
        printf("%s ",devname);
        if (pci3126_init(fd,1)<0)
        {
                fprintf(stderr,"Device initialize error %s(%s)0,
                devname,strerror(errno));
                        exit(0);
        }

        ch =0;
        value = PCI3126_AD_RANGE_M10VtoP10V;
        pci3126_set_ioport(fd,PCI3126_RANGE_OFFSET,&value);

        pci3126_set_channel(fd,0);
        Us = 1000000/cycle;
        printf("%d Hz is %d     us0,cycle,Us);
          if (pci3126_set_convert_timer(fd,8,Us)<0)
          {
                        fprintf(stderr,"Interrupt initialize error %s(%s)0,
                        devname,strerror(errno));
                        exit(0);
          }

          if (pci3126_set_trigger(fd,PCI3126_EXTTRIG_OFF)<0)
          {
                        fprintf(stderr,"Interrupt initialize error %s(%s)0,
                        devname,strerror(errno));
                        exit(0);
          }
          if          (pci3126_setup_signal(fd,pci3126_interrupt_han-
dler,PCI3126_IMASK_BSY)<0)
          {
                        fprintf(stderr,"Interrupt initialize error %s(%s)0,
                        devname,strerror(errno));
                        exit(0);
          }
        pci3126_set_gate(fd,PCI3126_GATE_ON);
        for(i=0;count<MAXDATA; i++)
        {
                sleep(1); ",count);
                printf("%d
```

```
        }
        pci3126_set_gate(fd,PCI3126_GATE_OFF);

        pci3126_disable_intrrupt(fd ,PCI3126_IMASK_ALL);

        if (pci3126_uninit(fd)<0)
        {
                fprintf(stderr,"%s0,strerror(errno));
        }
        close(fd);
        if (error)
        {
                for(i=0;i<count;i++)
                {
                    printf("count %d:%04X0,i,DATA[i]);
                }
        }
        printf("Error %d0,error);
        printf("Success %d0,count-error);
    }
```

SEE ALSO

/usr/local/CNC/drivers/extmem/interface/pci3126 下のプログラム

AUTHORS