

DO-128 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、コンテック社製 DO- 128 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

DO- 128 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。

- DO-128 ボード
- RedHawk 6.x 以上
- Extmem version 6.7 以上

DO-128は、パソコンにデジタル信号の出力機能を拡張するPCI Expressバス対応ボードです。非絶縁オープンコレクタ出力128点を搭載しています。

3. インストール方法

DO-128 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。

DO-128 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
DO128 BSP 実行パッケージのインストール
# rpm -ivh bin-do128-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- do128-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
```

DO128 BSP 実行パッケージのインストール

```
# apt install ./bin-do128-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-do128-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
```

DO128 BSP 実行パッケージのインストール

```
# apt install ./bin-do128-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-do128-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。)

DO-128 BSP パッケージは **/usr/local/CNC/drivers/extmem/interface/do128** 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

DO-128 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/ do128** 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- do128-X.Y_RHx.y-z.x86_64 (開発パッケージの削除)
# rpm -e bin-do128-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-do128-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-do128-rhx.y (開発パッケージの削除)
# apt purge bin-do128-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-do128-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-do128-rhx.y (開発パッケージの削除)
# apt purge bin-do128-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-do128-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 do128

do128(3)

do128(3)

NAME

libcontec - 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

do128 は、external memory ドライバを利用した do128 ボードアクセスライブラリです。

```
#include <sys/do128.h>
```

```
gcc [options ...] file -ldo128 -lxtmem ...
```

DO128

割り込みハンドラの登録

```
int do128_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 do128_reset(int fd);
```

```
int do128_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 do128_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 0x0000ace0 offset 0x00000000 32 bytes

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 0x14-0x15)の値を戻す

int do128_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 do128_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

割り込みを許可する

int do128_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 割り込みを許可するビットマスク

インターバルタイマーをセットする

int do128_set_system_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 値を設定する変数へのポインタ

インターバルタイマーの現在値を読み出す

int do128_get_system_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 do128_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 do128_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 do128_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 入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 do128_write_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 出力変数へのポインタ

すべてのチャンネルの入力データを読み出す
int do128_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 入力配列変数へのポインタ

LP すべてのチャンネルのデータを出力する
int do128_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

SEE ALSO

/usr/local/CNC/drivers/extmem/contec/do128 下のプログラム

AUTHORS

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

28 Apr 2016

do128(3)