

# AIO-163202F Board Support Package Installation on RedHawk ReleaseNotes Revision B

September 12, 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 AIO- 163202F 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

AIO- 163202F 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。

- AIO-163202F ボード
- RedHawk 6.x 以上
- Extmem version 8.3 以上

AIO-163202Fは、高精度アナログ入出力、デジタル入出力、カウンタを搭載したPCI Expressバス準拠のマルチファンクションボードです。

## 3. インストール方法

**AIO-163202F 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。

**AIO-163202F 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
AIO-163202F BSP 実行パッケージのインストール
# rpm -ivh bin-aio163202f-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- aio163202f-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

AIO-163202F BSP 実行パッケージのインストール
# apt install ./bin-aio163202f-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-aio163202f-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
```

#### **AIO-163202F BSP 実行パッケージのインストール**

```
# apt install ./bin-aio163202f-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-aio163202f-rhx.y_X.Y_arm64.deb
# umount /mnt
```

(*x.y* は RedHawk のバージョン番号であり、6.0,6.3,6.5,7.0,7.2 または 7.3 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。)

**AIO-163202F BSP** パッケージは `/usr/local/CNC/drivers/extmem/interface/ aio163202f` 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

**AIO-163202F 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/ aio163202f` 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 rpm -ivh dev- aio163202f-X.Y_RHx.y-z.x86_64 (開発パッケージの削除)
```

```
# rpm -e bin-aio163202f-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-aio163202f-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-aio163202f-rhx.y (開発パッケージの削除)
```

```
# apt purge bin-aio163202f-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-aio163202f-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-aio163202f-rhx.y (開発パッケージの削除)
```

```
# apt purge bin-aio163202f-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-aio163202f-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 aio163202f
aio163202f(3)
```

aio163202f(3)

### NAME

aio163202f - 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

aio163202f は、external memory ドライバを利用した AIO163202F ボードアクセスライブラリです。

```
#include <sys/aio163202f.h>
gcc [options ...] file -laio163202f -lxtmem ...
```

```
*****
AIO163202F
*****
```

マルチファンクションボード AIO163202F をアクセスする関数群

デバイスの非初期化処理

```
int aio163202f_reset(int fd,int port);
port は、以下のビットオアまたは、全てを示す AIO163202F_ALL_PORT である
AIO163202F_AI_PORT
AIO163202F_AO_PORT
AIO163202F_DI_PORT
AIO163202F_DO_PORT
AIO163202F_COUNTER_PORT
AIO163202F_MEMORY_PORT
AIO163202F_ECU_PORT
```

```
int aio163202f_uninit(int fd);
```

デバイスの初期化処理

```
int aio163202f_init(int fd,int option);
```

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 aio163202f_get_ioport_long(int fd,int offset,unsigned int *value);
int aio163202f_get_ioport_short(int fd,int offset,unsigned int *value);
```

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 aio163202f_set_ioport_long(int fd,int offset,unsigned int *value);
int aio163202f_set_ioport_short(int fd,int offset,unsigned int *value);
```

割り込みハンドラの登録

```
int aio163202f_setup_signal(int fd,void (*interrupt_hadler)(int),unsigned int aimask,unsigned int aomask,unsigned int dimask,unsigned int comask,unsigned int memask);
```

割り込みを許可する

```
int aio163202f_enable_intrrupt(int fd,unsigned int aimask,unsigned int
aomask,unsigned int dimask,unsigned int comask,unsigned int memask);
```

割り込みを禁止する

```
int aio163202f_disable_intrrupt(int fd,unsigned int aimask,unsigned int
aomask,unsigned int dimask,unsigned int comask,unsigned int memask);
```

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の値を戻す

```
int aio163202f_intr_service(int fd,unsigned int *iflag, unsigned int
*aiflag,unsigned int *aoflag,unsigned int *diflag,unsigned int
*coflag,unsigned int *meflag, int *pending);
```

イベントコントローラ入出力信号の結線を行う

```
int aio163202f_set_signal_assign(int fd,unsigned int destination_sig-
nal,unsigned int source_signal);
```

イベントコントローラ入出力信号の結線を得る

```
int aio163202f_get_signal_assign(int fd,unsigned int destination_sig-
nal,unsigned int source_signal,unsigned int *destination_sig-
nal_ret,unsigned int *source_signal_ret);
```

設定できる信号は以下の通り

destination_signal	DST_SIGNAL_AI_STORE_ENABLE_TRIGGER	:	AI 格
納許可トリガ	DST_SIGNAL_AI_SWITCH_TRIGGER	:	AI 切
り換えトリガ 1 1	DST_SIGNAL_AI_STORE_DISABL_TRIGGER	:	AI 格
納不許可トリガ 2 2	DST_SIGNAL_SAMPLING_CLOCK	:	サンブ
リングクロック 4 4	DST_SIGNAL_AO_UPDATE_ENABLE_TRIGGER	:	AO 更
新許可トリガ 32 20	DST_SIGNAL_AO_UPDATE_DISABLE_TRIGGER	:	AO 更
新不許可トリガ 34 22	DST_SIGNAL_UPDATE_CLOCK	:	更新ク
ロック 36 24	DST_SIGNAL_AIEXTSTATUS00	:	
AIExtStatus00 112 70	DST_SIGNAL_AIEXTSTATUS01	:	
AIExtStatus01 113 71	DST_SIGNAL_AOEXTSTATUS00	:	
AOExtStatus00 114 72	DST_SIGNAL_AOEXTSTATUS01	:	
AOExtStatus01 115 73	DST_SIGNAL_CNTEXTSTATUS00	:	
CNExtStatus00 116 74	DST_SIGNAL_CNTEXTSTATUS01	:	
CNExtStatus01 117 75	DST_SIGNAL_COUNT_START_TRIGGER0	:	カウン
ト開始トリガ 0 128 80	DST_SIGNAL_COUNT_START_TRIGGER1	:	カウン
ト開始トリガ 1 129 81	DST_SIGNAL_COUNT_STOP_TRIGGER0	:	カウン
ト停止トリガ 0 130 82	DST_SIGNAL_COUNT_STOP_TRIGGER1	:	カウン
ト停止トリガ 1 131 83	DST_SIGNAL_COUNT_LOAD_TRIGGER0	:	カウン

トロードトリガ 0 132 84	DST_SIGNAL_COUNT_LOAD_TRIGGER1	:	カウン
トロードトリガ 1 133 85	DST_SIGNAL_COUNT_UP0	:	カウン
ト 0 対象/アップカウント 134 86	DST_SIGNAL_COUNT_UP1	:	カウン
ト 1 対象/アップカウント 135 87	DST_SIGNAL_AI_WP_LATCH_TRIGGER	:	AI W.P
Latchトリガ 162 A2	source_signal		
	SRC_SIGNAL_NO_CONNECTION	:	Non
Connection 0 00 固定	SRC_SIGNAL_AI_INTERNAL_CLOCK	:	内部サ
ンプリングクロック(Internal CLK for AI) 4 04 クロック	SRC_SIGNAL_AI_BEFORE_TRIGGER_NUM_END	:	ビフォ
ートリガサンプリング回数終了 (AI Before Trigger Num End) 17 11	SRC_SIGNAL_AI_AFTER_TRIGGER_NUM_END	:	アフタ
ートリガサンプリング回数終了 (AI After Trigger Num End) 18 12	SRC_SIGNAL_AI_LEVEL_TRIGGER0	:	AI レ
ベルトリガ 0 (AI Level Trigger 0) 28 1C	SRC_SIGNAL_AI_LEVEL_TRIGGER1	:	AI レ
ベルトリガ 1 (AI Level Trigger 1) 29 1D	SRC_SIGNAL_AI_BUSY	:	変換動
作中(AI Busy) 48 30	SRC_SIGNAL_AI_BEFORE_TRIGGER_SAMMPLING_BUSY:		ビフォ
ートリガサンプリング (Before Sampling Busy) 49 31	SRC_SIGNAL_AI_AFTER_TRIGGER_SAMMPLING_BUSY:		アフタ
ートリガサンプリング (After Sampling Busy) 50 32	SRC_SIGNAL_AO_INTERNAL_CLOCK	:	内部更
新クロック(Internal CLK for AO) 66 42 クロック	SRC_SIGNAL_AO_BEFORE_TRIGGER_NUM_END	:	ビフォ
ートリガ更新回数終了 (AO Before Trigger Num End ) 80 50 フラグ	SRC_SIGNAL_AO_BUSY	:	AO
Busy 112 70	SRC_SIGNAL_AO_BEFORE_TRIGGER_RENEWAL	:	Before
Trigger Renewal 113 71	SRC_SIGNAL_DI_AI_EXTCLK_START_EDGE	:	AI Ext
Start Edge 144 90	SRC_SIGNAL_DI_AI_EXTCLK_STOP_EDGE	:	AI Ext
Stop Edge 145 91	SRC_SIGNAL_DI_AI_EXTCLK	:	AI Ext
CLK 146 92	SRC_SIGNAL_DI_AO_EXTCLK_START_EDGE	:	AO Ext
Start Edge 147 93	SRC_SIGNAL_DI_AO_EXTCLK_STOP_EDGE	:	AO Ext
Stop Edge 148 94	SRC_SIGNAL_DI_AO_EXTCLK	:	AO Ext
CLK 149 95	SRC_SIGNAL_DI_CNT_EXTUCLK1	:	CNT
Ext UCLK0 150 96	SRC_SIGNAL_DI_CNT_EXTUCLK2	:	CNT
Ext UCLK1 152 98	SRC_SIGNAL_COUNT_INTERNAL0	:	内部標
準タイマ 0 (Internal CLK For CNT0) 272 110	SRC_SIGNAL_COUNT_INTERNAL1	:	内部標
準タイマ 1 (Internal CLK For CNT1) 273 111			

SRCSIGNALCOUNT_COUNTUP0	:	比較力
ウント一致 0 (CNT0 Count Up) 288 120		
SRCSIGNALCOUNT_COUNTUP1	:	比較力
ウント一致 1 (CNT1 Count Up) 289 121		
SRCSIGNALCOUNT_BUSY0	:	CNT
Busy0 305 131		
SRCSIGNALCOUNT_BUSY1	:	CNT
Busy1 306 132		
SRCSIGNALMEMORY_AI_COMPARITION_DATA_NUM	:	AI 比
較データ数 Hold(1) (AI Comparison Data Num(1)) 322 142		
SRCSIGNALMEMORY_AI_COMPARITION_DATA_WP	:	AI 比
較 W.P(1) (AI Comparison W.P(1)) 329 149		
SRCSIGNALMEMORY_AO_DATA_EMPTY	:	AO
Data Empty 352 160		
SRCSIGNALMEMORY_AO_COMPARITION_DATA_NUM	:	AO 比
較データ数 Hold(1) (AI Comparison Data Num(1)) 353 161		
SRCSIGNALMEMORY_AO_COMPARITION_DATA_RP	:	AO 比
較 R.P(1) (AO Comparison R.P(1)) 360 168		
SRCSIGNALMEMORY_GENERAL_COMMAND0	:	汎用コ
マンド 0 (General Command 0) 384 180		
SRCSIGNALMEMORY_GENERAL_COMMAND1	:	汎用コ
マンド 1 (General Command 1) 385 181		

#### 汎用パルス出力コマンド

```
int aio163202f_raise_signal(int fd,int offset,int value);
```

通常 offset は AIO163202F\_ECU\_COMMAND\_OFFSET、で value は AIO163202F\_ECU\_GENERAL1\_COMMAND(0x5) か AIO163202F\_ECU\_GENERAL2\_COMMAND(0x6)である

イベントコントローラ(0x80)に、AI 異常停止設定コマンド(0x10)を値 value で出力する(outpd(0x84,value);)

```
int aio163202f_ai_abort_settings(int fd,int value);
```

value は、0x00000001 が AI CLK Error で、0x00010000 が AI Over Flow である。

イベントコントローラ(0x80)に、AO 異常停止設定コマンド(0x11)を値 value で出力する(outpd(0x84,value);)

```
int aio163202f_ao_abort_settings(int fd,int value);
```

value は、0x00000001 が AO CLK Error で、0x01000000 が AO Ext Stop である。

```
int aio163202f_ai_setcommand(int fd,int command,int value);
```

アナログ入力機能設定コマンド

command,value は以下の通り

command	:	value
AIO163202F_AI_CAL_COMMAND	:	AI キャリブレーション条件設定

以下の値のオアをとったもの(ajdust(data)|ch|range|select)を設定する

select	AIO163202F_AI_CAL_OFF-SET
range	AIO163202F_AI_CAL_GAIN
-10 to +10	AIO163202F_AI_CAL_RANGE_M10VtoP10V
0 to +10	AIO163202F_AI_CAL_RANGE_0VtoP10V

-5 to +5	AIO163202F_AI_CAL_RANGE_M5VtoP5V
0 to +5	AIO163202F_AI_CAL_RANGE_0VtoP5V
AIO163202F_AI_CAL_RANGE_M2p5VtoP2p5V	
-2.5 to +2.5	
0 to +2.5	AIO163202F_AI_CAL_RANGE_0VtoP2p5V
	ch:
	AIO163202F_AI_CAL_A
	AIO163202F_AI_CAL_B
	adjust
	AIO163202F_AI_CAL_ADJUST(data)
定	AIO163202F_AI_ADJUST_COMMAND : AI 調整用データ確認/設
ABLE	AIO163202F_AI_ADJUST_EPROM_WRITE_DISABLE
	AIO163202F_AI_ADJUST_EPROM_WRITE
	AIO163202F_AI_ADJUST_EPROM_READ
AIO163202F_AI_ADJUST_EPROM_WRITE_ENABLE	
AIO163202F_AI_RESET_COMMAND	: AI 初期化コマンド
	value=0 を設定
AIO163202F_AI_GATE_OPEN_COMMAND	: AI 内部ゲートオープンコマンド
	value=0 を設定
AIO163202F_AI_ABORT_COMMAND	: AI 強制停止コマンド
	value=0 を設定
AIO163202F_AI_INTCLK_COMMAND	: AI 用内部クロック設定
	value= 内部クロックのパ
ルス周期 = 周期(ms)/25 -1	
AIO163202F_AI_SCNCLK_COMMAND	: AI チャネルスキャンクロック設定
	value= 次のチャンネルの
変換までの時間 = 周期(ns)/25 -1	
AIO163202F_AI_CHNUM_COMMAND	: AI 格納チャンネル数設定
	value=チャンネル数-1(最大 31)
AIO163202F_AI_CHSEQ_COMMAND	: AI チャネルシーケンス設定
	value=変換チャンネル
ル<<16 変換番号	
AIO163202F_AI_MODE_COMMAND	: AI 入力方式設定
GLEEND シングルエンド	AIO163202F_AI_MODE_SINGLE-
FERNTIAL デファレンシアル	AIO163202F_AI_MODE_DIFF-
AIO163202F_AI_RANGE_COMMAND	: AI レンジ設定
-10 to +10	AIO163202F_AI_RANGE_M10VtoP10V
0 to +10	AIO163202F_AI_RANGE_0VtoP10V
-5 to +5	AIO163202F_AI_RANGE_M5VtoP5V
0 to +5	AIO163202F_AI_RANGE_0VtoP5V



-2.5 to +2.5 AIO163202F\_AI\_RANGE\_M2p5VtoP2p5V

0 to +2.5 AIO163202F\_AI\_RANGE\_0VtoP2p5V

AIO163202F\_AI\_BEFTRG\_COMMAND : AI ビフォートリガサンプルリング回数設定  
value = ビフォートリガ

サンプリング回数-1  
AIO163202F\_AI\_AFTTRG\_COMMAND : AI アフタートリガサンプルリング回数設定  
value = アフタートリガ

サンプリング回数-1  
AIO163202F\_AI\_REP\_COMMAND : AI リピート回数設定  
value = AI リピート回数

& 0xFFFF | AIO163202F\_AI\_REP\_INFINITY  
AIO163202F\_AI\_LVLTRG\_MODE\_COMMAND : AI レベルトリガ\_比較タイプ設定

AIO163202F\_AI\_LVL-  
TRG\_MODE\_RISE 立ち上がりエッジ

AIO163202F\_AI\_LVL-  
TRG\_MODE\_FALL 立ち下がりエッジ

AIO163202F\_AI\_LVL-  
TRG\_MODE\_BOTH 立ち上がりエッジ,立ち下がりエッジ

AIO163202F\_AI\_LVL-  
TRG\_MODE\_INRANGE インレンジ

AIO163202F\_AI\_LVL-  
TRG\_MODE\_OUTRANGE アウトレンジ

AIO163202F\_AI\_LVTRG\_WDCH\_COMMAND : AI レベルトリガ\_監視チャンネル設定  
value = (AI レベルトリガ番号 & 0x3) | ((監視チャンネル番号 & 0x1F)<<16)

AIO163202F\_AI\_LVTRG\_STATE\_NUM\_COMMAND : AI レベルトリガ\_ステート維持回数設定  
value = (AI レベルトリガ番号 & 0x3) | (((設定回数-1) & 0xFFFF)<<16)

AIO163202F\_AI\_LVTRG\_LEVEL\_COMMAND : AI レベルトリガ\_比較レベル設定  
value = (AI レベルトリガ番号 & 0x3)|((レベル番号&1)<<8) | (((入力電圧レベル) & 0xFFFF)<<16)

int aio163202f\_ao\_setcommand(int fd,int command,int value);  
AIO163202F\_AO\_RESET\_COMMAND : AO 初期化コマンド  
value=0 を設定

AIO163202F\_AO\_GATE\_OPEN\_COMMAND : AO 内部ゲートオープンコマンド  
value=0 を設定

AIO163202F\_AO\_ABORT\_COMMAND : AO 強制停止コマンド  
value=0 を設定

AIO163202F\_AO\_INTCLK\_COMMAND : AO 用内部クロック設定  
value= 内部クロックのパルス周期 = 周期(ms)/25 -1

AIO163202F\_AO\_CHNUM\_COMMAND : AO 更新チャンネル数設定  
value=チャンネル数-1(最大 1)|(AIO163202F\_AO\_CHNUM\_MULTI or AIO163202F\_AO\_CHNUM\_SINGLE)

AIO163202F\_AO\_RANGE\_COMMAND : AO レンジ設定

value=(range<<16|ch)を設定する  
range

-10 to +10	AIO163202F_AO_RANGE_M10VtoP10V
0 to +10	AIO163202F_AO_RANGE_0VtoP10V
-5 to +5	AIO163202F_AO_RANGE_M5VtoP5V
0 to +5	AIO163202F_AO_RANGE_0VtoP5V
-2.5 to +2.5	AIO163202F_AO_RANGE_M2p5VtoP2p5V
0 to +2.5	AIO163202F_AO_RANGE_0VtoP2p5V

AIO163202F\_AO\_BEFTRG\_COMMAND : AO ビフォートリガ更新回数設定

value = ビフォートリガ

更新回数-1

AIO163202F\_AO\_REP\_COMMAND : AO リピート回数設定

value = AO リピート回数

& 0xFFFF | AIO163202F\_AO\_REP\_INFINITY or 0

AIO163202F\_AO\_REP\_COUNT\_SRC\_COMMAND : AO リピートカウントソース設定

value = 0:自動(AO ビフォートリガ更新回数)

value = 1:通常(更新不許可トリガの回数)

AIO163202F\_AO\_CAL\_COMMAND : AO キャリブレーション条件設定

以下の値のオアをとったものの(adjust(data)|ch|range|select)を設定する

select

SET AIO163202F\_AO\_CAL\_OFF-

AIO163202F\_AO\_CAL\_GAIN

range

-10 to +10 AIO163202F\_AO\_CAL\_RANGE\_M10VtoP10V

0 to +10 AIO163202F\_AO\_CAL\_RANGE\_0VtoP10V

-5 to +5 AIO163202F\_AO\_CAL\_RANGE\_M5VtoP5V

0 to +5 AIO163202F\_AO\_CAL\_RANGE\_0VtoP5V

AIO163202F\_AO\_CAL\_RANGE\_M2p5VtoP2p5V

-2.5 to +2.5

AIO163202F\_AO\_CAL\_RANGE\_0VtoP2p5V

0 to +2.5

AIO163202F\_AO\_CAL\_RANGE\_M1p25Vto1p25V

-1.25 to +1.25

ch:

AIO163202F\_AO\_CAL\_CH0

AIO163202F\_AO\_CAL\_CH1

adjust

AIO163202F\_AO\_CAL\_ADJUST(data)

AIO163202F\_AO\_ADJUST\_COMMAND : AO 調整用データ確認/設定

AIO-163202F-PE テクニカルリファレンスマニ

マニュアル 64/123 を参照してください

```
int aio163202f_ao_setcommand2(int fd,int command,int value1,int value2);
```

```
    AIO163202F_AO_RESET_COMMAND          : AO 初期化コマンド
        value1=0,value2=0 を設定
    AIO163202F_AO_GATE_OPEN_COMMAND      : AO 内部ゲートオープンコマンド
        value1=0,value2=0 を設定
    AIO163202F_AO_ABORT_COMMAND          : AO 強制停止コマンド
        value1=0,value2=0 を設定
    AIO163202F_AO_INTCLK_COMMAND         : AO 用内部クロック設定
        value1=内部クロックのパルス周期の下位 = ((周期(ms)/25 -1) & 0xFFFF)
        value2=内部クロックのパルス周期の上位 = ((周期(ms)/25 -1)>>16)
        value1=チャンネル 数-1(最大 1)|(AIO163202F_AO_CHNUM_MULTI or AIO163202F_AO_CHNUM_SINGLE)
        value2=0
    AIO163202F_AO_RANGE_COMMAND          : AO レンジ設定
        value1=ch を設定する
        value2=以下のレンジのいずれかを設定する
        AIO163202F_AO_RANGE_M10VtoP10V
        -10 to +10
        AIO163202F_AO_RANGE_0VtoP10V
        0 to +10
        AIO163202F_AO_RANGE_M5VtoP5V
        -5 to +5
        AIO163202F_AO_RANGE_0VtoP5V
        0 to +5
        AIO163202F_AO_RANGE_M2p5VtoP2p5V
        -2.5 to +2.5
        AIO163202F_AO_RANGE_0VtoP2p5V
        0 to +2.5
    AIO163202F_AO_BEFTRG_COMMAND         : AO ビフォートリガ更新回数設定
        value1 = ((ビフォートリガ更新回数-1) & 0xFFFF)
        value2 = ((ビフォートリガ更新回数-1) >>16)
    AIO163202F_AO_REP_COMMAND            : AO リピート回数設定
        value1 = AO リピート回数 & 0xFFFF
        value2 =
    AIO163202F_AO_REP_INFINITY or 0
    AIO163202F_AO_REP_COUNT_SRC_COMMAND  : AO リピートカウントソース設定
        value1 = 0:自動(AO ビフォートリガ更新回数)
        value1 = 1:通常(更新不許可トリガの回数)
        value2 = 0
    AIO163202F_AO_CAL_COMMAND           : AO キャリブレーション条件設定
        value1= 以下の値のオアをとったもの(ch|range|select)を設定する
        select
        AIO163202F_AO_CAL_OFF-
```

SET

	range	AIO163202F_AO_CAL_GAIN
-10 to +10		AIO163202F_AO_CAL_RANGE_M10VtoP10V
0 to +10		AIO163202F_AO_CAL_RANGE_0VtoP10V
-5 to +5		AIO163202F_AO_CAL_RANGE_M5VtoP5V
0 to +5		AIO163202F_AO_CAL_RANGE_0VtoP5V
AIO163202F_AO_CAL_RANGE_M2p5VtoP2p5V		
-2.5 to +2.5		AIO163202F_AO_CAL_RANGE_0VtoP2p5V
0 to +2.5		

AIO163202F\_AO\_CAL\_RANGE\_M1p25Vto1p25V  
-1.25 to +1.25

ch:  
AIO163202F\_AO\_CAL\_CH0  
AIO163202F\_AO\_CAL\_CH1  
value2=adjust

AIO163202F\_AO\_ADJUST\_COMMAND : AO 調整用データ確認/設定

AIO-163202F-PE テクニカルリファレンスマニ

ュアル 64/123 を参照してください

int aio163202f\_memory\_setcommand(int fd,int command,int value);  
command,value は以下の通り

command	: value
AIO163202F_MEMORY_RESET_COMMAND	:MEM 初期化コマンド value=0 を設定
AIO163202F_MEMORY_AI_TRANSFER_COMMAND	:AI 転送設定 AIO163202F_MEMORY_AI_TRANSFER_IO
AIO163202F_MEMORY_AO_TRANSFER_COMMAND	:AO 転送設定 AIO163202F_MEMORY_AO_TRANSFER_IO
AIO163202F_MEMORY_AI_MEM_WP_COMMAND	:AI MEM W.P 確認/設定 value=0 を設定
AIO163202F_MEMORY_AI_MEM_RP_COMMAND	:AI MEM R.P 確認/設定 value=0 を設定
AIO163202F_MEMORY_AO_MEM_WP_COMMAND	:AO MEM W.P 確認/設定 value=0 を設定
AIO163202F_MEMORY_AO_MEM_RP_COMMAND	:AO MEM R.P 確認/設定 value=0 を設定
AIO163202F_MEMORY_AI_MEM_CLEAR_COMMAND	:AI MEM クリア value=0 を設定
AIO163202F_MEMORY_AO_MEM_CLEAR_COMMAND	:AO MEM クリア value=0 を設定
AIO163202F_MEMORY_AI_MEMORY_TYPE_COMMAND	:AI メモリタイプ設定 AIO163202F_MEM-
ORY_AI_MEMORY_TYPE_FIFO FIFO	AIO163202F_MEM-
ORY_AI_MEMORY_TYPE_RING RING	AIO163202F_MEM-
AIO163202F_MEMORY_AO_MEMORY_TYPE_COMMAND	:AO メモリタイプ設定 AIO163202F_MEM-
ORY_AO_MEMORY_TYPE_FIFO FIFO	AIO163202F_MEM-
ORY_AO_MEMORY_TYPE_RING RING	AIO163202F_MEM-

AIO163202F\_MEMORY\_AI\_TRGGER\_WRITE\_POINT\_COMMAND :AIトリガ  
イトポイント設定

value=0 を設定

比  
比較タイプ設定  
AIO163202F\_MEMORY\_AO\_MEM\_COMPARETE\_TYPE\_COMMAND :AO 用 MEM

value=0 を設定

比  
比較データ設定  
AIO163202F\_MEMORY\_AO\_MEM\_COMPARETE\_DATA\_COMMAND :AO 用 MEM

value=0 を設定

AIO163202F\_MEMORY\_AO\_READ\_POINTER\_MAX\_ADDRESS\_COMMAND:AO リードポ  
インタ上限アドレス設定

value=0 を設定

int aio163202f\_memory\_setcommand2(int fd,int command,int value1,int  
value2);

command,value1,value2 は以下の通り

AIO163202F\_MEMORY\_AI\_MEM\_COMPARETE\_COMMAND:AI 用 MEM 比較タイプ設定

value1:比較番号  
FIFO

RING  
0:データ数(1)

W.P(1)  
1:データ数(2)

W.P(2)  
value2:比較タイプ  
FIFO

RING  
0:無効 特

定ポイント  
1:上方一致 特

定ポイント毎  
2: 設

定ポイント毎  
AIO163202F\_MEMORY\_AO\_MEM\_COMPARETE\_COMMAND:AO 用 MEM 比較データ設定

value1:比較番号  
FIFO

RING  
0:データ数(1)

R.P(1)  
1:データ数(2)

R.P(2)  
value2:比較タイプ  
FIFO

RING  
0:無効 特

定ポイント  
1:上方一致 特

定ポイント毎  
int aio163202f\_di\_setcommand2(int fd,int command,int value1,int  
value2);

command,value1,value2 は以下の通り

AIO163202F\_DI\_RESET\_COMMAND: DI 初期化コマンド

value1=0,value2=0 を設定

AIO163202F\_DI\_EDGE\_DETECT\_COMMAND: DI エッジ検出設定

れるデジタル入力タイプを選択

value1=以下の外部から入力さ

AIO163202F\_DI\_EDGE\_TYPE\_DI  
AIO163202F\_DI\_EDGE\_TYPE\_AI  
AIO163202F\_DI\_EDGE\_TYPE\_AO  
AIO163202F\_DI\_EDGE\_TYPE\_CNT

の検出ビットを

value2= デジタル入力の以下

AIO163202F\_DI\_EDGE\_TYPE\_NONE  
AIO163202F\_DI\_EDGE\_TYPE\_RISE  
AIO163202F\_DI\_EDGE\_TYPE\_FALL  
AIO163202F\_DI\_EDGE\_TYPE\_BOTH

下記のポート分シフトした値を

ビットオアした値を設定

AIO163202F\_DI\_EDGE\_BIT00\_SHIFT  
AIO163202F\_DI\_EDGE\_BIT01\_SHIFT  
AIO163202F\_DI\_EDGE\_BIT02\_SHIFT  
AIO163202F\_DI\_EDGE\_BIT03\_SHIFT  
AIO163202F\_DI\_EDGE\_BIT04\_SHIFT  
AIO163202F\_DI\_EDGE\_BIT05\_SHIFT  
AIO163202F\_DI\_EDGE\_BIT06\_SHIFT  
AIO163202F\_DI\_EDGE\_BIT07\_SHIFT

例 え ば

、(AIO163202F\_DI\_EDGE\_TYPE\_RISE<<AIO163202F\_DI\_EDGE\_BIT00\_SHIFT)|...

AIO163202F\_DI\_DIGITAL\_FILTER\_COMMAND: DI Digital Filter 設定

value1=以下のデジタルフィ

ルタを入れる信号名を選択

AIO163202F\_DI\_DIGI-

TAL\_FILTER\_DI DI Digital Filter

AIO163202F\_DI\_DIGI-

TAL\_FILTER\_AI AI Digital Filter

AIO163202F\_DI\_DIGI-

TAL\_FILTER\_AO AO Digital Filter

AIO163202F\_DI\_DIGI-

TAL\_FILTER\_CNT CNTDigital Filter

value2= 以下のデジタルフィ

ルタ時間を選択

AIO163202F\_DI\_DIGI-

TAL\_FILTER\_OFF Digital Filter なし

AIO163202F\_DI\_DIGI-

TAL\_FILTER\_1U Digital Filter 1 マイクロ秒

AIO163202F\_DI\_DIGI-

TAL\_FILTER\_128U Digital Filter 128 マイクロ秒

AIO163202F\_DI\_DIGI-

TAL\_FILTER\_16M Digital Filter 16m 秒

int aio163202f\_do\_setcommand2(int fd,int command,int value1,int value2);

command,value1,value2 は以下の通り

AIO163202F\_DO\_RESET\_COMMAND DO 初期化コマンド

value1=0;value2=0

AIO163202F\_DO\_OUTSETTING\_COMMAND: DO 出力設定

value1= 汎用 DO 番号

AIO163202F\_DO\_OUTSET-

TING\_DO\_BIT00 DO 出力 bit0

AIO163202F\_DO\_OUTSET-

TING\_DO\_BIT01 DO 出力 bit1

TING_DO_BIT02 DO 出力 bit2	AIO163202F_DO_OUTSET-
TING_DO_BIT03 DO 出力 bit3	AIO163202F_DO_OUTSET-
TING_DO_BIT04 DO 出力 bit4	AIO163202F_DO_OUTSET-
TING_DO_BIT05 DO 出力 bit5	AIO163202F_DO_OUTSET-
TING_DO_BIT06 DO 出力 bit6	AIO163202F_DO_OUTSET-
TING_DO_BIT07 DO 出力 bit7	AIO163202F_DO_OUTSET-
TING_AI_STATUS00	AIO163202F_DO_OUTSET-
TING_AI_STATUS01	AIO163202F_DO_OUTSET-
TING_AO_STATUS00	AIO163202F_DO_OUTSET-
TING_AO_STATUS01	AIO163202F_DO_OUTSET-
TING_CNT0_STATUS	AIO163202F_DO_OUTSET-
TING_CNT1_STATUS	AIO163202F_DO_OUTSET-
	value2= DO 出力から何を出力するかを設定
DO 出力	AIO163202F_DO_OUTSETTING_DO
ECU 出力	AIO163202F_DO_OUTSETTING_ECU
	AIO163202F_DO_OUTSTATE_COMMAND: DO 出力状態設定
	value1= 汎用 DO 番号
STATE_DO_BIT00 DO 出力 bit0	AIO163202F_DO_OUT-
STATE_DO_BIT01 DO 出力 bit1	AIO163202F_DO_OUT-
STATE_DO_BIT02 DO 出力 bit2	AIO163202F_DO_OUT-
STATE_DO_BIT03 DO 出力 bit3	AIO163202F_DO_OUT-
STATE_DO_BIT04 DO 出力 bit4	AIO163202F_DO_OUT-
STATE_DO_BIT05 DO 出力 bit5	AIO163202F_DO_OUT-
STATE_DO_BIT06 DO 出力 bit6	AIO163202F_DO_OUT-
STATE_DO_BIT07 DO 出力 bit7	AIO163202F_DO_OUT-
STATE_AI_STATUS00	AIO163202F_DO_OUT-
STATE_AI_STATUS01	AIO163202F_DO_OUT-
STATE_AO_STATUS00	AIO163202F_DO_OUT-
STATE_AO_STATUS01	AIO163202F_DO_OUT-
STATE_CNT0_STATUS	AIO163202F_DO_OUT-
	AIO163202F_DO_OUT-

STATE\_CNT1\_STATUS

value2=以下のロジックとタイプを論理 OA したものを設定

を設定

AIO163202F\_DO\_OUT-  
STATE\_LOGIC\_HI or AIO163202F\_DO\_OUTSTATE\_LOGIC\_LO  
AIO163202F\_DO\_OUT-  
STATE\_LOGIC\_THROUGH or AIO163202F\_DO\_OUTSTATE\_LOGIC\_PLUSE  
int aio163202f\_cnt\_setcommand(int fd,int command,int value);

AIO163202F\_COUNTER\_RESET\_COMMAND: //CNT 初期化コマンド  
value=0 を設定

AIO163202F\_COUNTER\_GATE\_OPEN\_COMMAND://CNT 内部ゲートオープンコマンド

value=チャンネル選択:各チャンネルに内部ゲート  
オープンするかしなないかを設定

0ビット:CNT00 1ビット:CNT01 を表す  
AIO163202F\_COUNTER\_ABORT\_COMMAND: //CNT 強制停止コマンド  
value=チャンネル選択:各チャンネルに強制停止す

るかしなないかを設定

0ビット:CNT00 1ビット:CNT01 を表す

int aio163202f\_ai\_getcommand(int fd,int command,int \*value);  
以下のコマンドが利用できる、value の詳細は aio163202f\_ai\_setcommand()を参照

AIO163202F\_AI\_INTCLK\_COMMAND: //AI 用内部クロック設定  
AIO163202F\_AI\_SCNCLK\_COMMAND: //AI チャネルスキヤンクロック設定

ク設定

AIO163202F\_AI\_CHNUM\_COMMAND: //AI 格納チャネル数設定  
AIO163202F\_AI\_CHSEQ\_COMMAND: //AI チャネルシーケンス設定  
AIO163202F\_AI\_MODE\_COMMAND: //AI 入力方式設定  
AIO163202F\_AI\_RANGE\_COMMAND: //AI レンジ設定  
AIO163202F\_AI\_BEFTRG\_COMMAND: //AI ビフオートリガサンプリ

ング回数設定

AIO163202F\_AI\_AFTTRG\_COMMAND: //AI アフタートリガサンプリ

ング回数設定

AIO163202F\_AI\_REP\_COMMAND: //AI リピート回数設定  
AIO163202F\_AI\_LVLTRG\_MODE\_COMMAND: //AI レベルトリガ\_比較タイプ設定

プ設定

AIO163202F\_AI\_LVTRG\_WDCH\_COMMAND: //AI レベルトリガ\_監視チャンネル設定

ネル設定

AIO163202F\_AI\_LVTRG\_STATE\_NUM\_COMMAND: //AI レベルトリガ\_ステート

維持回数設定

AIO163202F\_AI\_LVTRG\_LEVEL\_COMMAND: //AI レベルトリガ\_比較レベル設定

ル設定

int aio163202f\_ao\_getcommand(int fd,int command,int \*value);  
以下のコマンドが利用できる、value の詳細は aio163202f\_ao\_setcommand()を参照

AIO163202F\_AO\_RESET\_COMMAND: //AO 初期化コマンド  
AIO163202F\_AO\_GATE\_OPEN\_COMMAND: //AO 内部ゲートオープンコマンド

ド

AIO163202F\_AO\_ABORT\_COMMAND: //AO 強制停止コマンド  
AIO163202F\_AO\_INTCLK\_COMMAND: //AO 用内部クロック設定  
AIO163202F\_AO\_CHNUM\_COMMAND: //AO 更新チャネル数設定  
AIO163202F\_AO\_RANGE\_COMMAND: //AO レンジ設定  
AIO163202F\_AO\_BEFTRG\_COMMAND: //AO ビフオートリガ更新回数設定

定



```

    AIO163202F_AO_REP_COMMAND:          //AO リポート回数設定
    AIO163202F_AO_REP_COUNT_SRC_COMMAND://AO リポートカウントソース設
定
    AIO163202F_AO_CAL_COMMAND:          //AO キャリブレーション条件設
定
    AIO163202F_AO_ADJUST_COMMAND:      //AO 調整用データ確認/設定
int aio163202f_memory_getcommand(int fd,int command,int *value);
以下のコマンドが利用できる、valueの詳細は aio163202f_memory_setcommand()
を参照
    AIO163202F_MEMORY_AI_TRANSFER_COMMAND:          //AI 転送設
定
    AIO163202F_MEMORY_AO_TRANSFER_COMMAND:          //AO 転送設
定
    AIO163202F_MEMORY_AI_MEM_WP_COMMAND:           //AI MEM
W.P 確認/設定
    AIO163202F_MEMORY_AI_MEM_RP_COMMAND:           //AI MEM
R.P 確認/設定
    AIO163202F_MEMORY_AO_MEM_WP_COMMAND:           //AO MEM
W.P 確認/設定
    AIO163202F_MEMORY_AO_MEM_RP_COMMAND:           //AO MEM
R.P 確認/設定
    AIO163202F_MEMORY_AI_MEM_CLEAR_COMMAND:        //AI MEM ク
リア
    AIO163202F_MEMORY_AO_MEM_CLEAR_COMMAND:        //AO MEM ク
リア
    AIO163202F_MEMORY_AI_MEMORY_TYPE_COMMAND:     //AI メモリ
タイプ設定
    AIO163202F_MEMORY_AO_MEMORY_TYPE_COMMAND:     //AO メモリ
タイプ設定
    AIO163202F_MEMORY_AI_TRGGER_WRITE_POINT_COMMAND: //AI トリガ
ライトポイント設定
    AIO163202F_MEMORY_AO_MEM_COMPARETE_TYPE_COMMAND: //AO 用
MEM
    比較タイプ設定
    AIO163202F_MEMORY_AO_MEM_COMPARETE_DATA_COMMAND: //AO 用 MEM
    比較データ設定
    AIO163202F_MEMORY_AO_READ_POINTER_MAX_ADDRESS_COMMAND://AO リード
    ポインタ上限アドレス設定
int aio163202f_memory_getcommand2(int fd,int command,int *value1,int
*value2);
以下のコマンドが利用できる、value1,value2の詳細
は aio163202f_memory_setcommand2()を参照
    AIO163202F_MEMORY_AI_MEM_COMPARETE_COMMAND:    //AI 用 MEM
    比較タイプ設定 2パラメータ
    AIO163202F_MEMORY_AO_MEM_COMPARETE_COMMAND:    //AO 用
MEM
    比較データ設定
int aio163202f_di_getcommand(int fd,int command,int *value);
以下のコマンドが利用できる、valueの詳細は aio163202f_di_setcommand()を参
照
    AIO163202F_DI_RESET_COMMAND:                  //DI 初期化コマンド
    AIO163202F_DI_EDGE_DETECT_COMMAND:            //DI エッジ検出設定
    AIO163202F_DI_DIGITAL_FILTER_COMMAND:         //DI Digital Filter 設定
int aio163202f_do_getcommand(int fd,int command,int *value);

```

以下のコマンドが利用できる、value の詳細は aio163202f\_do\_setcommand() を参照

```
AIO163202F_DO_OUTSETTING_COMMAND: //DO 出力設定
AIO163202F_DO_OUTSTATE_COMMAND://DO 出力状態設定
int aio163202f_cnt_getcommand(int fd,int command,int *value);
```

以下のコマンドが利用できる、value の詳細は aio163202f\_cnt\_setcommand() を参照

```
AIO163202F_COUNTER_GATE_OPEN_COMMAND://CNT 内部ゲートオープンコマンド
```

```
AIO163202F_COUNTER_ABORT_COMMAND: //CNT 強制停止コマンド
```

```
int aio163202f_set_clock(int fd,int value);
```

クロックの値を value に設定する。

```
int aio163202f_get_counter(int fd,int ch,int *value);
```

クロックの値を読み出し \*value に設定する。

```
int aio163202f_set_counter(int fd,int ch,int value);
```

カウンタ ch の値を value に設定する

```
int aio163202f_set_counter_comparete(int fd,int ch,int value);
```

クロック ch の値を読み出し \*value に設定する

```
int aio163202f_ai_read(int fd,unsigned long int *data,int nch);
```

AD 変換器から nch 分データを読み出す

```
int aio163202f_ai_read_data(int fd,unsigned long int *data);
```

AD 変換器から1データだけ読み出す

```
int aio163202f_ai_read_csr(int fd,unsigned short int *data);
```

AD 変換器の CSR から1データだけ読み出す

```
int aio163202f_ai_calibration(int fd,int value);
```

AI のキャリブレーションを行う

```
int aio163202f_ai_start(int fd);
```

AD 変換をスタートする

```
int aio163202f_ao_write(int fd,unsigned long int *addr,int nch);
```

配列 addr のデータを nch 分 AO に書き出す

```
int aio163202f_ao_start(int fd);
```

AO をスタートする

```
int aio163202f_di_setmask(int fd,int value);
```

DI のマスクを設定する

```
int aio163202f_di_getvalue(int fd,int *value);
```

DI の値を読み出す

```
int aio163202f_do_setmask(int fd,int value);
```

DO のマスクを設定する

```
int aio163202f_do_setvalue(int fd,int value);
```

DO に値を設定する

```
int aio163202f_ao_read(int fd,unsigned long int *addr,int nch);
```

出力した AO の値を読み出す

## SEE ALSO

/usr/local/CNC/drivers/extmem/contec/aio163202f 下のプログラム

## AUTHORS

Copyright (C) 1995-2016 Concurrent Realtime Inc.