32IN/32OUT DIO ボード

USB インターフェース

HUSB-DIO464v2

ユーザーズマニュアル

NC ボードシリーズ

絶縁型入出力ボード

株式会社ハイバーテック

http://www.hivertec.co.jp/

この説明書は
HUSB-DIO464v2
HUSB-DIO464v2(D)に適応しています.

株式会社 ハイバーテック
東京都江東区新大橋 1-8-11
三井生命新大橋ビル
TEL 03-3846-3801
FAX 03-3846-3773
sales@hivertec.co.jp

第 1. 20 版 2010 年 6 月 10 日発行

本製品をご使用される前に「1.注意事項」を必ずご一読の上ご利用をお願い致します.

目　　次

図　表　目　次

1. 注意事項

1.1 保証範囲

1　本製品の保証期間は，お買い上げ頂いた日より3年間です．保証期間中に弊社の判断により欠陥が判明した場合には，本製品を弊社に引き取り，修理または交換を行います．

2　保証期間内外に関わらず，弊社製品の使用，供給（納期）または故障に起因する，お客様及び第三者が被った，直接，間接，2次的な損害あるいは，遺失利益の損害に付いて，弊社は本製品の販売価格以上の責任を負わないものとしますので，予めご了承下さい．

1.2 免責事項

1　本書に記載された内容に沿わない，製品の取付，接続，設定，運用により生じた損害に対しましては，一切の責任を負いかねますので，予めご了承下さい．

2　本製品は，一般電子機器用（工作機械・計測機器・FA／OA機器・通信機器等）に製造された半導体製品を使用していますので，その誤作動や故障が直接，生命を脅かしたり，身体・財産等に危害を及ぼしたりする恐れのある装置（医療機器・交通機器・燃焼機器・安全装置等）に適用できるような設計，意図，または，承認，保証もされていません．
ゆえに本製品の安全性，品質および性能に関しては，本マニュアル（またはカタログ）に記載してあること以外は明示的にも黙示的にも一切保証するものではありませんので，予めご了承下さい．

3　保証期間内外に関わらず，お客様が行った弊社の承認しない製品の改造または，修理が原因で生じた損害に対しましては，一切の責任を負いかねますので，予めご了承下さい．

4　本書に記載された内容について，弊社もしくは，第三者の特許権，著作権，商標権，その他の知的所有権の権利に対する保証または実施権の許諾を行うものではありません．
また本マニュアルに記載された情報を使用したことにより第三者の知的所有権等の権利に関わる問題が生じた場合，弊社は，その責任を負いかねますので，予めご了承下さい．

1.3 安全にお使い頂くために

この度は，弊社 NC ボードシリーズをご採用頂きまして，誠に有り難う御座います．本マニュアルは，本製品をご使用して頂く場合の取扱い，留意点に付いて記入してありますので，必ずご一読の上ご利用をお願い致します．
尚，本マニュアルは，本マニュアルが添付されたNCボード常設箇所付近の分かりやすい場所に常時保管し，必要に応じて適宜参照・確認頂きますよう，お願い致します．

安全上の注意

本製品のご使用前に，必ずこのユーザーズマニュアル及び付属書類を全て熟読し，内容を理解してから正しくご使用下さい．本製品の知識，安全の情報及び注意事項の全てに付いて習熟してからご使用下さい．
本ユーザーズマニュアルでは，安全注意事項のランクを「警告」，「注意」として区分してあります．

⚠ **警 告**	この表示を無視して，誤った取扱いをすると，人が死亡または重傷を負う可能性が想定される内容を示しています．
⚠ **注 意**	この表示を無視して，誤った取扱いをすると，人が傷害を負う可能性または物的損害が想定される内容を示しています．

1.3.1 対象ユーザー

⚠ **注 意**

本製品およびマニュアルは，以下の様な，ユーザーを対象としています．

⚠

・拡張用ボードの増設および配線に付いて基本的な知識を有している方．
・制御用電子機器およびパソコン等に付いて基本的な知識を有している方．

1.3.2 適合 Bus

⚠ **警 告**

⚠ 本製品は Universal Serial Bus 2.0 に適合したボードです

1.3.3 環境条件

⚠ **警 告**

本製品は，下記の環境条件下で保管・ご使用下さい．

⚠

- 動作周囲温度　0℃ ～ +50℃
- 動作周囲湿度　20%RH ～ 85%RH(但し結露せぬこと)
- 保存周囲温度　-15℃ ～ +75℃
- 保存周囲湿度　10%RH ～ 90%RH(但し結露せぬこと)
- 雰 囲 気　腐食性ガス・引火性ガス・オイルミスト・塵埃のないこと
- 標 高　海抜 3000m 以下(300m 毎に 2℃の上限値を下げた範囲で使用して下さい)

1.3.4 運搬・取り付け

警 告

本製品にふれる前に，金属に触り身体の静電気を取り除いて下さい.
静電気は，本ボードの故障の原因になります.

本製品を静電気の帯びやすい梱包材（エアーキャップなど）でくるまないで下さい.
静電気は，本ボードの故障の原因になります.

本製品の上に重いものを載せないで下さい. 重いものを乗せますと，部品が損傷し故障の原因になります.

本製品のジャンパ設定は，パソコン等に取り付ける前に行って下さい. 電源が ON の状態で設定しますと，設定を正しく認識しないで誤動作の原因になります.

本製品のジャンパ設定は，正しく行って下さい. 設定を間違えますと 誤動作の原因になります.

USB コネクタは本来 ホットプラグインが許されていますが制御用の接続には電源ON状態での USB コネクタの抜き挿しは避けてください.

本製品をパソコン等と接続する時は，コネクタを深くしっかりと挿入し，ケーブルの直近部分を固定する等して，動作中に抜けるなど接触不良が発生しない様な措置を施して下さい.
動作中に抜けたり，接触不良が発生したりすると，誤動作，故障の原因となります.

注 意

本製品を落としたり乱暴に扱ったりしないで下さい.
衝撃や振動が故障の原因となります.

本製品の半田面を手で直接触らないで下さい.
部品の突起などにより怪我をする恐れがあります.

1.3.5 配 線

警 告

外線用コネクタへの配線作業や外線用コネクタの着脱は，パソコン等の電源を OFF し電源コードを抜いてから行って下さい．
電源コードを抜かないで作業を行った場合，故障の原因になります．また，装置が思ぬ動作をすることがあります．

外線用コネクタへの配線は，コネクタ信号表などをよく確認し，正しく配線して下さい．間違った配線をしますと，故障・焼損の原因になります．

外部から供給する電源は，必ず定格以内でご使用下さい．定格以外で使用されますと，故障・焼損・誤動作の原因となります．

入出力回路に接続する回路は，必ず定格電流・電圧以内でご使用下さい．定格以外で使用されますと，故障・焼損・誤動作の原因となります．

外部配線用コネクタは，推奨のコネクタをご使用下さい．推奨以外のコネクタを使用されますと，接触不良などにより誤動作の原因となります．

外部配線用コネクタは，必ずロックしてご使用下さい．ロックしないで使用されますと，コネクタが外れたり接触不良を起こしたりして，誤動作の原因となります．

外部配線用ケーブルは，引っ張ったり重い荷重を掛けたりしないで下さい．コネクタが外れたり接触不良を起こしたりして，誤動作の原因となります．

外部配線用ケーブルは，モータの配線や AC 電源ケーブルなど，ノイズの多い配線とは出来るだけ離して下さい．配線が近いとノイズが　誤動作の原因となります．

1.3.6 試運転・調整

警 告

本製品を使用し装置を動作させる時は，プログラムのデバッグを充分行ってから動作させて下さい．プログラムに間違いがありますと，思わぬ動きをすることがあります．

本製品に添付してあるサンプルプログラムを使用し装置を動作させる時，最初は十分に安全な条件で，また機械系に合った設定を行って動作を確認して下さい．
機械系に合わない設定で動作を行うと思わぬ動きをすることがあります．

1.3.7 廃 棄

警 告

本製品を廃棄する時は，関連する法律・規則に従って処理して下さい．

2. 概　論

このマニュアルは USB 適合の ディジタル入出力ボード HUSB-DIO464v2 及び HUSB-DIO464v2(D)の取扱説明書です.（※）
※HUSB-DIO464v2(D)ボードは HUSB-DIO464v2 に DIN レール取付台が追加されたボードです.
以降, この説明書では, 双方ともに HUSB-DIO464v2 または DIO464v2 と呼びます.

■ HUSB-DIO464v2 は USB 1.1 規格に基づいています.

USBコネクタの抜き挿しの注意

USB コネクタは本来 ホットプラグインが許されていますが制御用の接続には電源 ON 状態での USB コネクタの抜き挿しは避けてください.

2.1 マニュアルの記載内容

このマニュアルには次の内容が記載されています.

■ ハードウェアに関する情報
(1) ポート構成
(2) ボード上の設定
(3) コネクタ割付
(4) 外部との接続

■ 添付ソフトウェア スタートアップガイド
(1) 添付ソフトのインストール方法
(2) ドライバI/F用DLL関数説明
(3) サンプルプログラム説明

2.2 添付ソフトウェア

2.2.1 添付ソフトウェア対応 OS

添付ソフトウェアの対応 OS は以下の通りです.
WIndows 7(64bit/32bit),, Windows Vista(64bit/32bit), Windows XP(32bit), Windows 2000,
Windows 98 Second Edition,

2.2.2 添付ソフトウェア内容

このボードには次のソフトウェアが添付されます.
デバイスドライバ(I/F 用 DLL)
サンプルプログラム・・・・・・・・・・ドライバ関数の使用法を解説するサンプルソフトです.

3. ハードウェア編

3.1 ブロック図

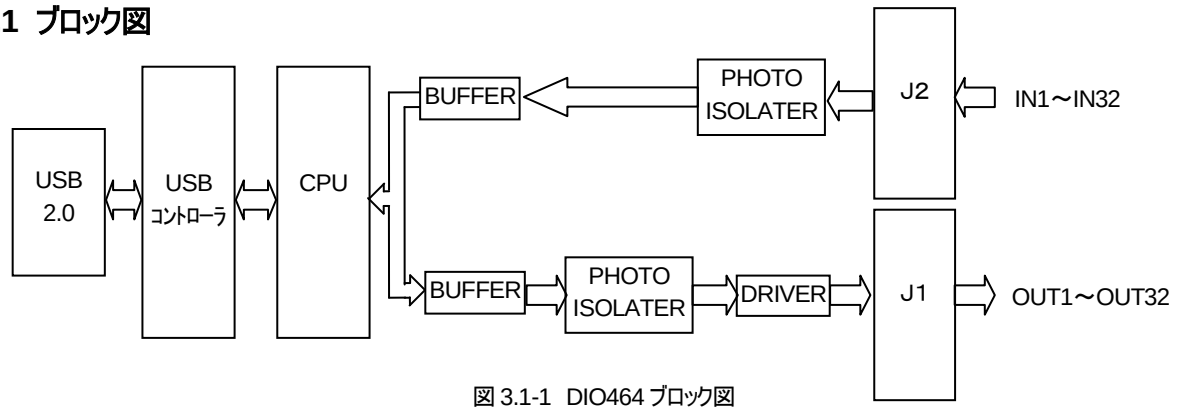


図 3.1-1 DIO464 ブロック図

3.2 HUSB-DIO464v2 呼称

■ HUSB-DIO464v2 の各部の呼称を下図 に示します.

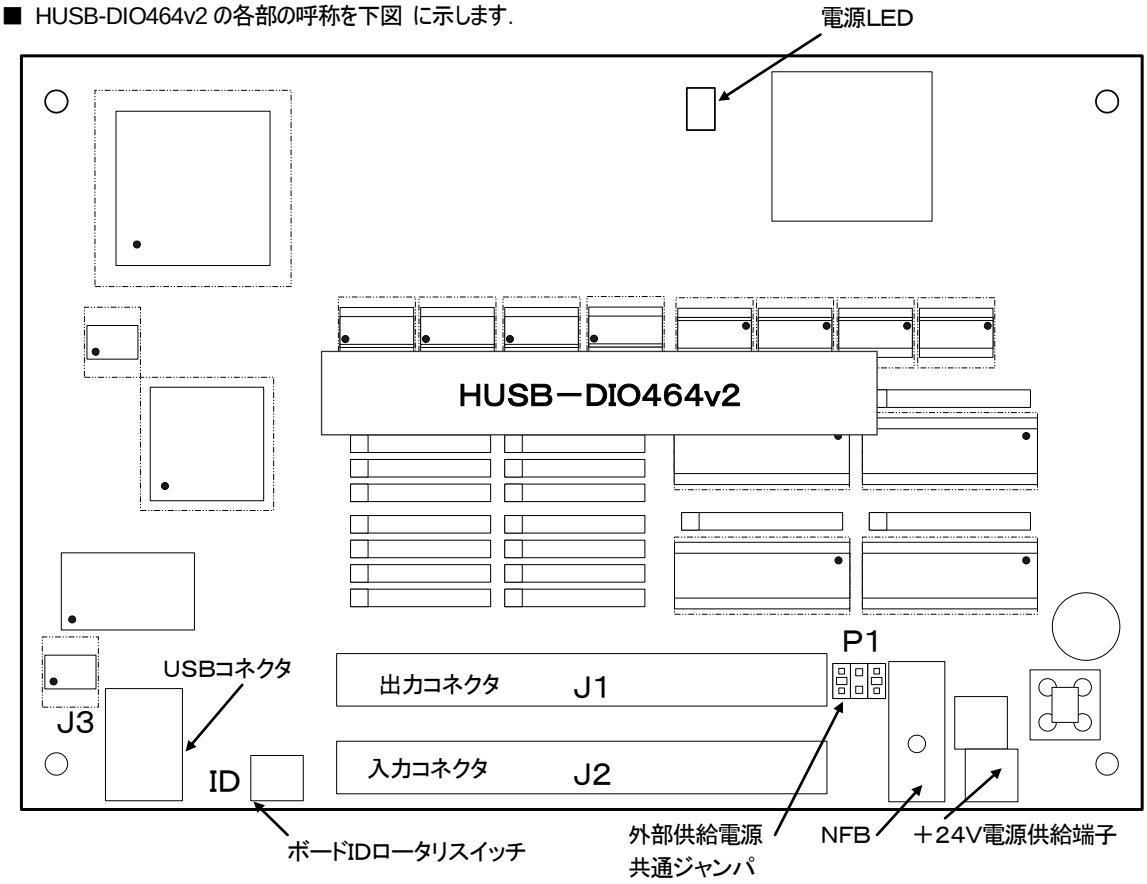


図 3.2-1 HUSB-DIO464 呼称

3.3 ボード内ローカル側ポート構成

本ボードのポート構成を表 3.3-1 に示します.

ポート名	ビット	入力(IN)
入力ポート１	0	IN 1
	1	IN 2
	2	IN 3
	3	IN 4
	4	IN 5
	5	IN 6
	6	IN 7
	7	IN 8
入力ポート２	0	IN 9
	1	IN10
	2	IN11
	3	IN12
	4	IN13
	5	IN14
	6	IN15
	7	IN16
入力ポート３	0	IN17
	1	IN18
	2	IN19
	3	IN20
	4	IN21
	5	IN22
	6	IN23
	7	IN24
入力ポート４	0	IN25
	1	IN26
	2	IN27
	3	IN28
	4	IN29
	5	IN30
	6	IN31
	7	IN32

ポート名	ビット	入力(IN)	出力(OUT)
出力ポート１	0	OUT 1	OUT 1
	1	OUT 2	OUT 2
	2	OUT 3	OUT 3
	3	OUT 4	OUT 4
	4	OUT 5	OUT 5
	5	OUT 6	OUT 6
	6	OUT 7	OUT 7
	7	OUT 8	OUT 8
出力ポート２	0	OUT 9	OUT 9
	1	OUT10	OUT10
	2	OUT11	OUT11
	3	OUT12	OUT12
	4	OUT13	OUT13
	5	OUT14	OUT14
	6	OUT15	OUT15
	7	OUT16	OUT16
出力ポート３	0	OUT17	OUT17
	1	OUT18	OUT18
	2	OUT19	OUT19
	3	OUT20	OUT20
	4	OUT21	OUT21
	5	OUT22	OUT22
	6	OUT23	OUT23
	7	OUT24	OUT24
出力ポート４	0	OUT25	OUT25
	1	OUT26	OUT26
	2	OUT27	OUT27
	3	OUT28	OUT28
	4	OUT29	OUT29
	5	OUT30	OUT30
	6	OUT31	OUT31
	7	OUT32	OUT32

表 3.3-1 ポート構成

3.4 ボード上の設定

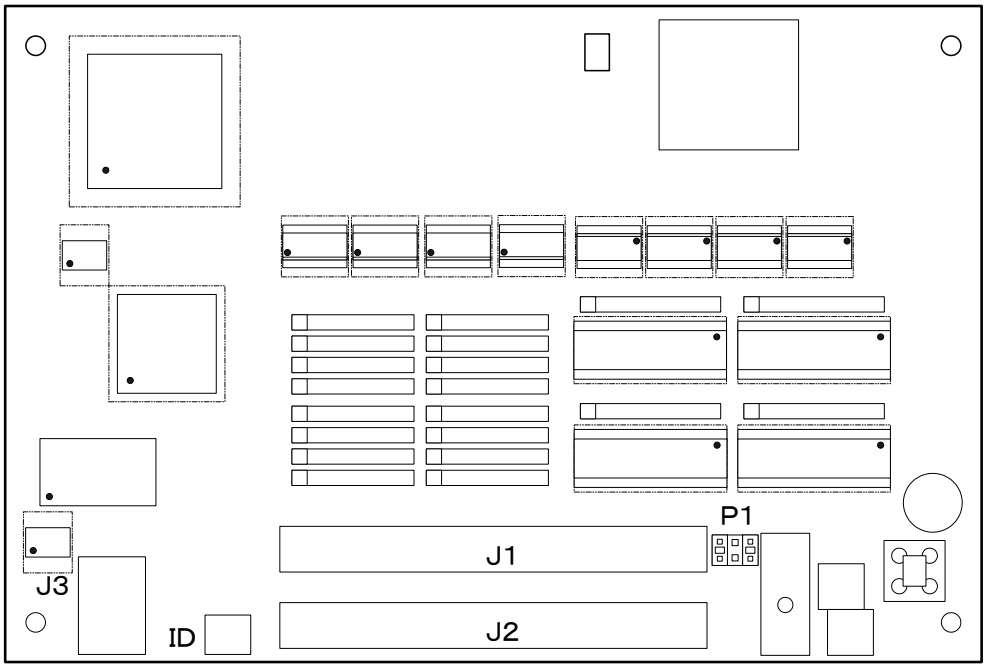


図 3.4-1 ジャンパ設定個所

(1) TB1 電源端子およびP1ジャンパ

TB1 は+24V 電源供給端子です.

■供給電源:+24V ±10%

P1 ジャンパは +24V カプラ絶縁電源 EXTPOW1, EXTPOW2 を接続します.

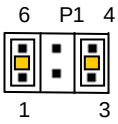
P1 ジャンパの出荷時の設定は 1-6 接続, 3-4 接続です.

（EXTGND1 とEXTGND2 共通，EXTPOW1とEXTPOW2 共通）

TB1 接続は 「表 3.7-3 TB1 電源端子」参照

■NFB は +24V 逆接続時過電流保護用. (1A)

過電流検出により白色突起が突出します. **原因除去後**突起押し込みにより復帰します.



EXTGND1 とEXTGND2 共通，EXTPOW1とEXTPOW2 共通 (出荷時の設定)

図 3.4-2 P1 ジャンパ

(2) ボード ID ロータリスイッチの設定

ボード ID は HUSB-DIO464 を識別する No.です.

矢印の位置が ID 設定値（16 進数:0～F(15)）となります. 出荷時の設定は“0”になっています.

図 3.4-3 ボード ID の設定(本図は“F”設定例)

3.5 入出力回路

本ボードの入出力回路を下図に示します.

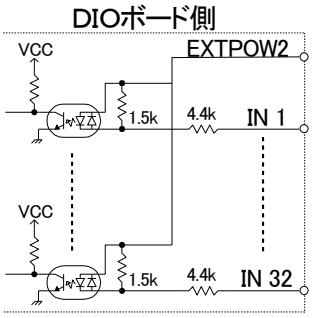


図 3.5-1 入力回路

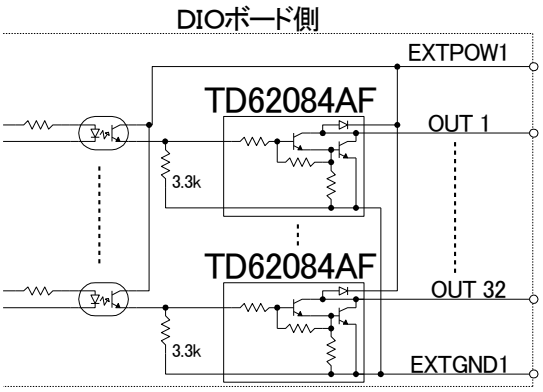


図 3.5-2 出力回路

3.6 外部との接続

3.6.1 入出力回路接続例

本ボードの入出力回路の接続例を下図に示します.

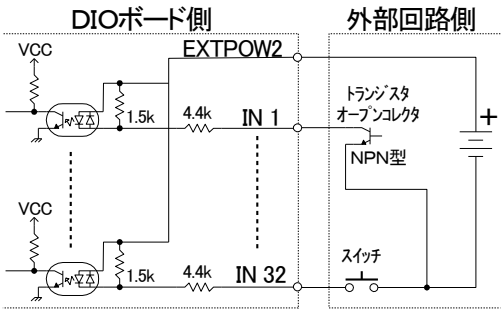


図 3.6-1 入力回路接続例

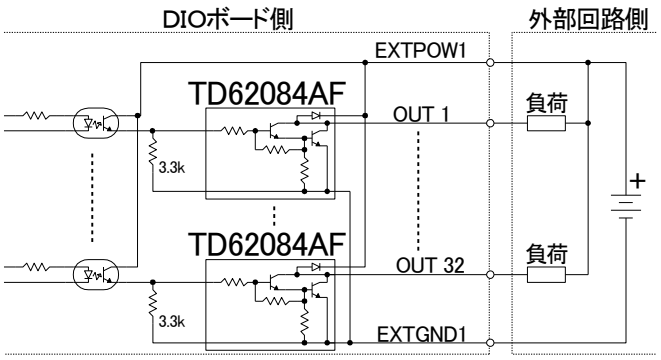
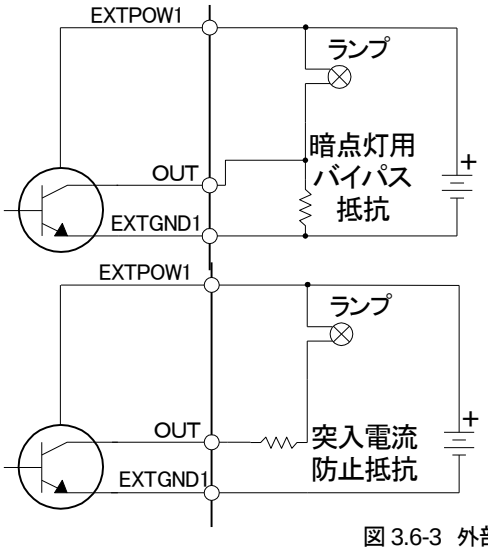


図 3.6-2 出力回路接続例

3.6.2 外部接続にあたっての注意事項

ランプやリレー等の誘導関係の負荷をコントロールする場合には，出力端子側で下記のような対策を行って下さい.



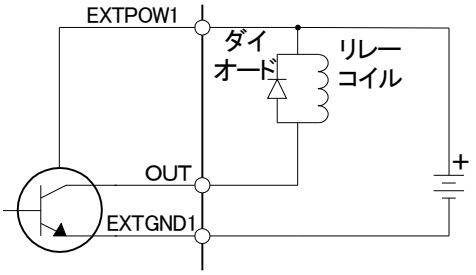


図 3.6-3 外部接続における対策

3.7 コネクタ信号表

(1) 出力コネクタ(J1)

出力コネクタピン配列表を下表に示します.

ピン番号	信 号 名		ピン番号	信 号 名	
1	EXTPOW1	+24V 入力	2	EXTPOW1	+24V 入力
3	OUT 1	出力 1	3	OUT 2	出力 2
5	OUT 3	出力 3	4	OUT 4	出力 4
7	OUT 5	出力 5	8	OUT 6	出力 6
9	OUT 7	出力 7	10	OUT 8	出力 8
11	OUT 9	出力 9	12	OUT10	出力 10
13	OUT11	出力 11	14	OUT12	出力 12
15	OUT13	出力 13	16	OUT14	出力 14
17	OUT15	出力 15	18	OUT16	出力 16
19	EXTGND1	24V 用コモン(GND)	20	EXTGND1	24V 用コモン(GND)
21	EXTPOW1	+24V 入力	22	EXTPOW1	+24V 入力
23	OUT17	出力 17	24	OUT18	出力 18
25	OUT19	出力 19	26	OUT20	出力 20
27	OUT21	出力 21	28	OUT22	出力 22
29	OUT23	出力 23	30	OUT24	出力 24
31	OUT25	出力 25	32	OUT26	出力 26
33	OUT27	出力 27	34	OUT28	出力 28
35	OUT29	出力 29	36	OUT30	出力 30
37	OUT31	出力 31	38	OUT32	出力 32
39	EXTGND1	24V 用コモン(GND)	40	EXTGND1	24V 用コモン(GND)

注意. EXTPOW1 は, 出力ポート用外部絶縁電源供給端子です.

J1 の EXTGND1(19, 20, 39, 40 ピン)はすべて接続して下さい.

コネクタ型式:40PIN フラットケーブルコネクタ XG4A-4031(オムロン)

ケーブル側コネクタ(参考) :XG4M-4030-T(フラットケーブル用)

:XG5M-4032-N(バラ線圧接用)

表 3.7-1 出力コネクタ(J1)ピン配列

(2) 入力コネクタ(J2)

入力コネクタピン配列表を下表に示します.

ピン番号	信 号 名		ピン番号	信 号 名	
1	EXTPOW2	+24V 入力	2	EXTPOW2	+24V 入力
3	IN 1	入力 1	3	IN 2	入力 2
5	IN 3	入力 3	4	IN 4	入力 4
7	IN 5	入力 5	8	IN 6	入力 6
9	IN 7	入力 7	10	IN 8	入力 8
11	IN 9	入力 9	12	IN10	入力 10
13	IN11	入力 11	14	IN12	入力 12
15	IN13	入力 13	16	IN14	入力 14
17	IN15	入力 15	18	IN16	入力 16
19	EXTGND2	24V 用コモン(GND)	20	EXTGND2	24V 用コモン(GND)
21	EXTPOW2	+24V 入力	22	EXTPOW2	+24V 入力
23	IN17	入力 17	24	IN18	入力 18
25	IN19	入力 19	26	IN20	入力 20
27	IN21	入力 21	28	IN22	入力 22
29	IN23	入力 23	30	IN24	入力 24
31	IN25	入力 25	32	IN26	入力 26
33	IN27	入力 27	34	IN28	入力 28
35	IN29	入力 29	36	IN30	入力 30
37	IN31	入力 31	38	IN32	入力 32
39	EXTGND2	24V 用コモン(GND)	40	EXTGND2	24V 用コモン(GND)

注意. EXTPOW2 は, 入力ポート用外部絶縁電源供給端子です.

コネクタ型式:40PIN フラットケーブルコネクタ XG4A-4031(オムロン)

ケーブル側コネクタ(参考) :XG4M-4030-T(フラットケーブル用)

:XG5M-4032-N(バラ線圧接用)

表 3.7-2 入力コネクタ(J2)ピン配列

(3) +24V 電源供給端子配列(TB1)

+24V 電源供給端子配列を下表に示します.

端子番号	信 号 名		備 考
1A	EXTPOW1	(+24V 外部電源供給)	COMMON1 と COMMON2 はジャンパ P1 の 3-4 接続により共通になります. 同様に EXTPOW1 と EXTPOW2 は P1 の 1-6 接続により共通になります.
1B	EXTGND2	(同 GND)	
2A	EXTPOW2	(+24V 外部電源供給)	
2B	EXTGND2	(同 GND)	

※マニュアル上でEXTGNDと表記されている部分は, 基板シルクがCOMになっている場合が有ります.

表 3.7-3 TB1 電源端子

3.8 ボード形寸

(1) HUSB-DIO464 ボード形寸

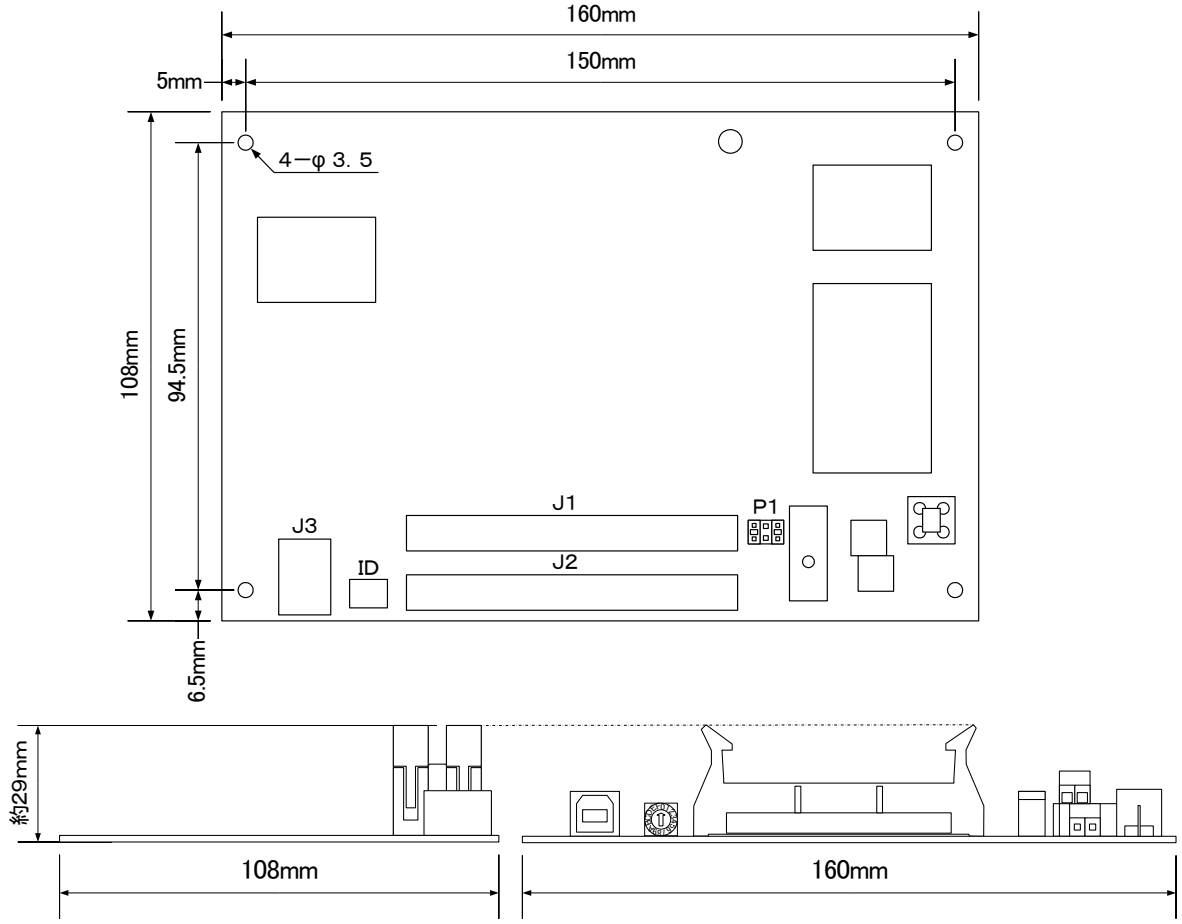


図 3.8-1 ボード形寸

(2) DIN レール取付台形寸

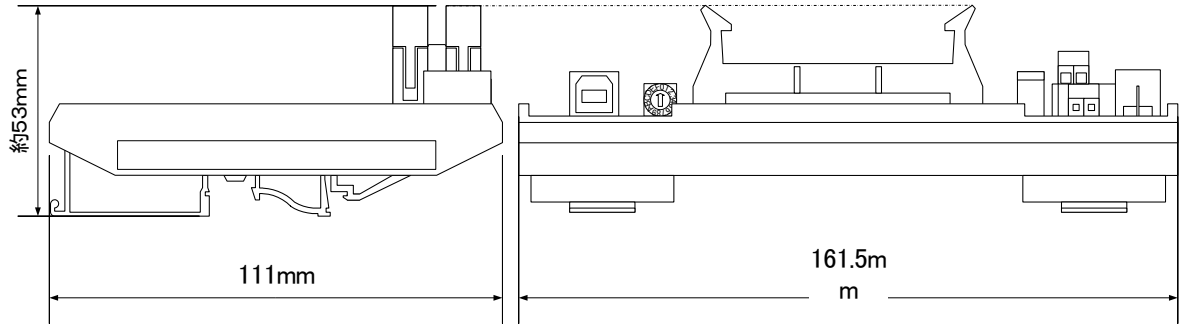


図 3.8-2 DIN レール取付台形寸

3.9 HUSB-DIO464 仕 様

項　目		仕　　様
【 入力部 】	入力点数	32 点
	入力形式	フォトカプラによる絶縁入力（TLP280-4 相当品を使用）
	定格入力電圧	ＤＣ２４Ｖ
	使用入力電圧範囲	ＤＣ２１．６Ｖ～DC26.4V
	定格入力電流	5mA
	入力抵抗	4.4kΩ
	応答時間	1msec以内（注 1）
	入力点数/コモン数	32 点／1 コモン
	入力論理	入力のフォトカプラ ON で内部論理“1”
【 出力部 】	出力点数	32 点
	出力形式	フォトカプラによる絶縁，オープンコレクタ・トランジスタアレー出力 （TD62084AP 相当品を使用）
	定格負荷電圧	ＤＣ２４Ｖ
	使用負荷電圧範囲	ＤＣ２１．６Ｖ～DC26.4V
	最大負荷電流	80mA
	応答時間	1msec以内（注 2）
	出力点数/コモン数	32 点／1 コモン
	出力論理	内部論理“1”で出力のトランジスタがON
【 周囲条件 】	供給電源	+24V±10%
	消費電流（約）	本体 0.35mA ＋ 入力 ON 点数×5mA ＋ 出力 ON 点数×80mA
	温度条件	0℃～50℃ ただし，結露ないこと.
	形　寸	ボードのみ　　：160mm X 107.5mm X 高さ 約 29mm
		DIN 取付台込み：161.5mm X 111mm X 高さ 約 53mm

注1.　入力コネクタピンに信号が入力されてから，信号がボード内部で認識されるまでの時間
(但し入力ポートのフィルタ設定が OFF の時)

注2.　ボード内部で出力が実行されてから，信号が出力コネクタピンに出力されるまでの時間

表 3.9-1 HUSB-DIO464 仕様

4. ソフトウェア編

ここでは次の説明をします.

◇ドライバのインストール方法
◇ボードへのアクセスとボード ID
◇サンプルプログラム（VC，VB，VC#によるソース添付）の使用方法.

また，この章では HUSB-DIO464v2，HUSB-DIO464 を DIO464 と呼びます.

4.1 対応 OS，ドライバ種別

添付ソフトウェアの対応 OS は以下の通りです.
Windows 7(64bit), Windows 7(32bit), Windows Vista(32bit), Windows Vista(32bit), Windows XP(32bit), Windows 2000, Windows 98 Second Edition

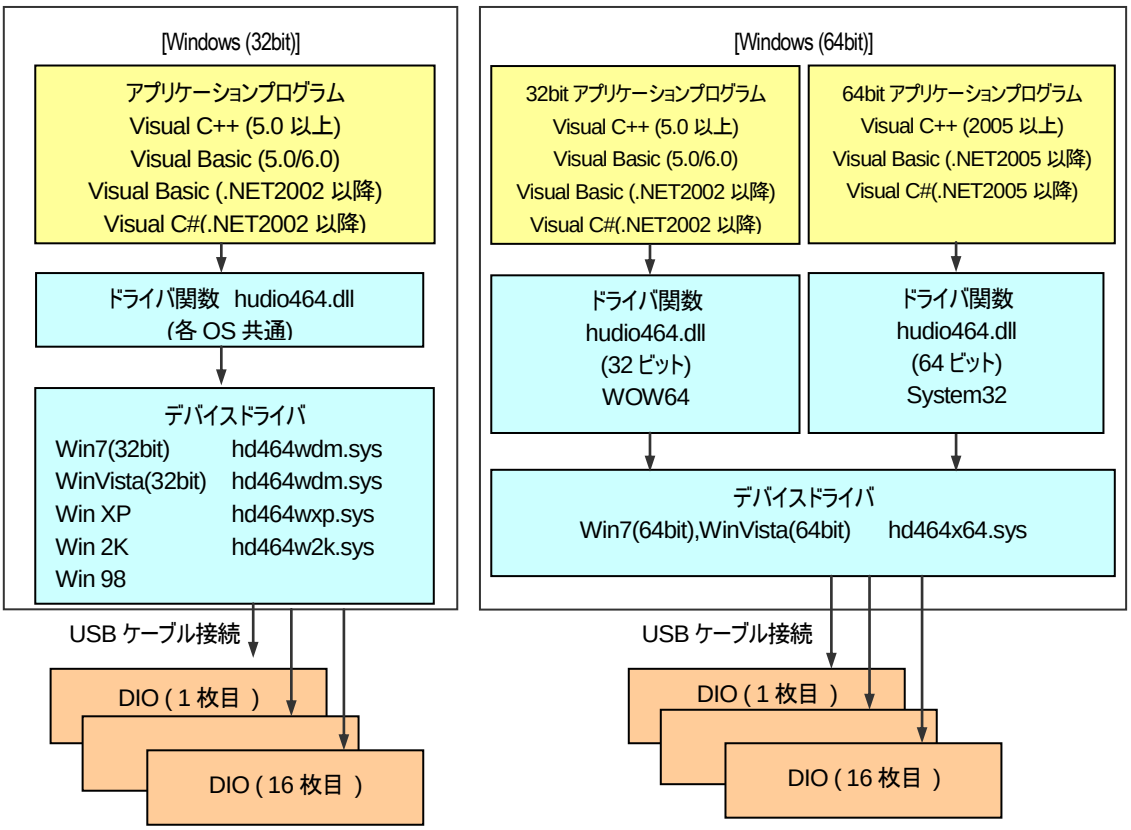
(1) デバイスドライバ種別

◇Windows 7(64bit)	(Win7(64bit)と表記)	に於いては・・hd464x64.sys
◇Windows Vista (64bit)	(WinVista(64bit)と表記)	に於いては・・hd464x64.sys
◇Windows 7(32bit)	(Win7(32bit)と表記)	に於いては・・hd464wdm.sys
◇Windows Vista (32bit)	(WinVista(32bit)と表記)	に於いては・・hd464wdm.sys
◇Windows XP	(WinXP と表記)	に於いては・・hd464wxp.sys
◇Windows 2000	(Win2K と表記)	に於いては・・hd464w2k.sys
◇Windows 98SE	(Win98 と表記)	に於いては・・hudio464.sys

(2) ドライバ I/F 用 DLL

◇hicpd530.dll(各 OS 共通で使用)

4.2 添付ソフトウェアの構成



添付ソフトウェアのフォルダ構成は，CD:¥Readme.txt を参照して下さい.

図 4.2-1 ソフトウェア構成

4.3 デバイスドライバのインストール

(1) Win7(32bit)と WinVista(32bit)へのインストール

① DIO464 をパソコンの USB ポートに装着する前に，パソコンの電源を ON にして Windows を起動します．
② CD ドライブ:¥win7_x86¥dpinst.exe を起動します．
③ "dpinst.exe"が起動されたら「次へ」をクリックして続行します．
「ドライバーソフトウェアの発行元を検証できません」とのメッセージが出る場合がありますが，
「このドライバーソフトウェアをインストールします」をクリックします．
④ インストーラー完了後，DIO464 の電源切状態で USB コネクタをパソコンの USB コネクタに装着します．
⑤ DIO464 の電源を投入します．
⑥ デバイスのインストールが自動的に行われます．(図 4.3-1 を参照)

[CD バージョン 3.10，デバイスドライババージョン 3.0.0.0 をインストールしている場合のデバイスドライバの更新]

① デバイスを取り外し，デバイスの電源を OFF します．
② 添付ディスクをディスクドライブに挿入します．
③ エクスプローラを起動し，D:¥cp430uin.exe（D ドライブである場合）を実行します．
④ Windows を再起動します．
⑤ Windows 起動後，前述の方法によりインストールを行います．

(2) Win7(64bit)と WinVista(64bit)へのインストール

① DIO464 をパソコンの USB ポートに装着する前に，パソコンの電源を ON にして Windows を起動します．
② CD ドライブ:¥win7_x64¥dpinst.exe を起動します．
③ "dpinst.exe"が起動されたら「次へ」をクリックして続行します．
④ インストーラー完了後，DIO464 の電源切状態で USB コネクタをパソコンの USB コネクタに装着します．
⑤ DIO464 の電源を投入します．
⑥ デバイスのインストールが自動的に行われます．

(3) WinXP へのインストール

① Windows 動作中を確認した後，DIO464 の電源切状態で USB コネクタをパソコンの USB コネクタに挿入します．
その後に DIO464 の電源を投入します．
② DIO464 がシステムにより検出され，自動的に必要なデバイスドライバのインストール画面が表示されます．
③ 添付ディスクをフロッピーディスクドライブに挿入します．
④ ソフトウェアを自動的にインストールする（推奨）を選択します．
⑤ HUSB-DIO464(WinXP)を指定してください．
後はシステムの指示に従ってインストールを完了させます．

(4) Win2K へのインストール

① Windows 動作中を確認した後，DIO464 の電源切状態で USB コネクタをパソコンの USB コネクタに挿入します．
その後に DIO464 の電源を投入します．
② DIO464 がシステムにより検出され，自動的に必要なデバイスドライバのインストール画面が表示されます．
③ システムがインストール元ディレクトリの指定を要求してきたら，添付ディスクをフロッピーディスクドライブに挿入します．
④ ◆" 検索場所の指定 "のチェックボックスを必ずチェックします．
⑤ CD:¥WIN2K を指定してください．
後はシステムの指示に従ってインストールを完了させます．

(5) Win98 へのインストール

① Windows 動作中を確認した後，DIO464 の電源切状態で USB コネクタをパソコンの USB コネクタに挿入します．
その後に DIO464 の電源を投入します．
② DIO464 がシステムにより検出され，自動的に必要なデバイスドライバのインストール画面が表示されます．（初回のみ）
③ システムがインストール元ディレクトリの指定を要求してきたら，添付ディスクをフロッピーディスクドライブに挿入します．
④ ◆" 検索場所の指定 "のチェックボックスを必ずチェックします．
⑤ CD:¥WIN98 を指定してください．
後はシステムの指示に従ってインストールを完了させます．

HUSB－DIO464を使用されている場合の注意

CD バージョン 3.1.0.0(デバイスドライババージョン 3.0.0.0)以降のデバイスドライバであれば HUSB-DIO464 と HUSB-DIO464v2 は同時に使用できます．但し，同時に使用する場合に CD バージョン 3.1.0.0(デバイスドライババージョン 3.0.0.0)以降のデバイスドライバをインストール後，以前のデバイスドライバをインストールしないでください．
以前のデバイスドライバでは HUSB-DIO464v2 は動作しません．

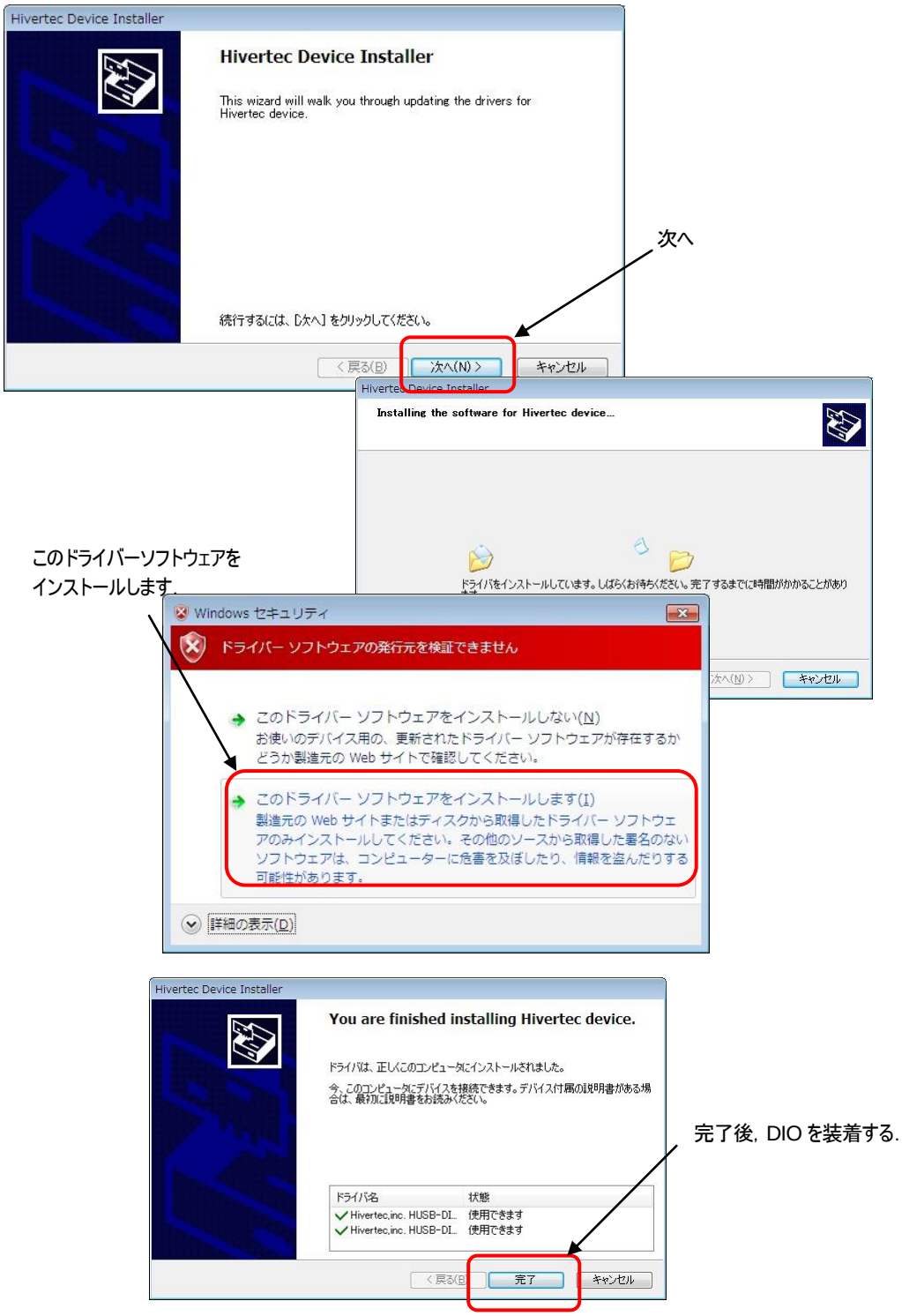


図 4.3-1 Win7,WinVista インストール手順

4.4デバイスドライバのアンインストール

(1) Win7/Vista デバイスドライバのアンインストール

[CD バージョン 4.0.0.0(デバイスドライババージョン 4.0.0.0)以降のデバイスドライバのアンインストール]

① デバイスを取り外し，デバイスの電源を OFF します．
② スタートメニューからコントロールパネルを起動します．
③ プログラムの追加と削除を選択します．
④ プログラムのアンインストールからアンインストールします．

図 4.4-1 Win7,WinVista アンインストール画面

[CD バージョン 3.1.0.0(デバイスドライババージョン 3.0.0.0)以前のデバイスドライバのアンインストール]

① デバイスを取り外し，デバイスの電源を OFF します．
② 添付ディスクをディスクドライブに挿入します．
③ エクスプローラを起動し，D:¥ d464uins.exe（CD ドライブが D ドライブである場合）を実行します．
④ Windows を再起動します．

(2) WinXP/2K/98 デバイスドライバのアンインストール

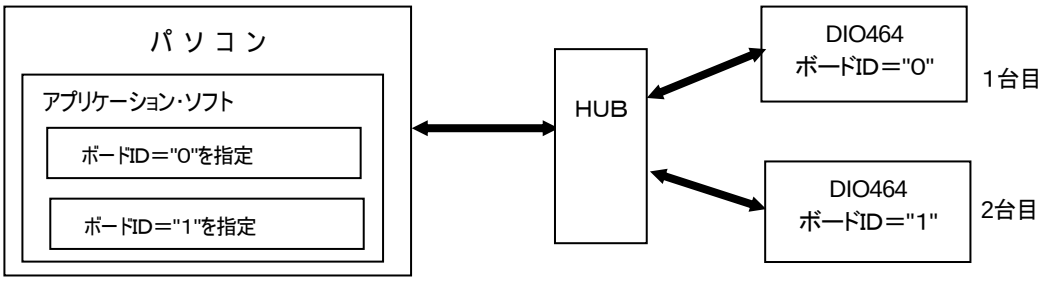
① デバイスを取り外し，デバイスの電源を OFF します．
② 添付ディスクをディスクドライブに挿入します．
③ エクスプローラを起動し，D:¥c d464uins.exe（CD ドライブが D ドライブである場合）を実行します．
④ Windows を再起動します．

アンインストール時の注意

デバイスドライバのアンインストール時は必ずデバイスを取り外し，DIO464 の電源を OFF にして行ってください．
電源を OFF にせずアンインストールを行うと，OS の再インストールが必要になる場合があります．

4.5ボードを複数枚使用する場合

DIO464 を同一のコンピュータに複数枚接続し，それぞれのボードと外部の接続を1対1に対応させたい場合について説明します．



(1) HUSB-DIO464 の指定
USB ケーブルで接続された2台以上の DIO464 は，パソコンによって内部で固定の番号が割り振られていますが，この番号をソフトウェアで利用できません．そのため，同一型名のボードを識別するために，各 DIO464 にそれぞれ固有な「ボード ID No.」を設定し識別します．(ボード ID は 0~15)

(2) ボード ID の確認方法
添付 CD 内のVCサンプル「¥sample¥Vc¥Release¥sud46400.exe」を実行し，画面のボード ID を確認します．

(3) ボード ID の設定方法
ボード ID の設定については ハードウェア編を参照してください．
※ ボード ID の設定は DIO464，DIO464v2 で共通です．
※ 1 枚でも複数枚でも DIO464 は複数プロセスから制御できません．

4.6 USB ボードの認識

USB ボードの認識・接続・切り離し

	項 目	備 考
1	DIO464（デバイス）の認識	アプリケーションプログラム起動時に行われます
2	起動時以降に接続されたデバイス	認識されません
3	起動時以降に切り離されたデバイス	制御不能となります
4	アプリケーション実行中に任意の USB 接続	アプリケーションの USB 通信に影響を与えます（通信停止状態が発生）
5	DIO464（デバイス）の着脱	接続では 5 秒以上経過してから使用（通信）してください 切り離し後の再接続は，5 秒以上経過してから行います

4.7 ボードアクセス方法

ある DIO ボードにアクセスするためには，どのようなハードウェアリソースを持つデバイスをオープンするかという情報が必要です．このハードウェアリソースをデバイス情報と呼びます．このデバイス情報を取得し，デバイス情報を使用してデバイスをオープンし，アクセスするためのデバイスハンドル値を取得します．

4.7.1 ボード(デバイス)認識用のデータ構造体

ボード認識のために次に示す HUSBDEVINF 型構造体が用意されています．

[Visual C++]

```
typedef struct _HUSBDEVINF {
   WORD          BrdID;          // ボード ID          ( 0～15 )
   WORD          EPcnt;          // エンドポイント数       (2)
   WORD          EPspc[8];       // エンドポイント仕様     (0x02,0x86,0x00,0x00)
   WORD          EPsiz[8];       // エンドポイント・バイト数(64,64, 0, 0)
   WORD          EPopt[8];       // エンドポイント・オプション( 予約 )
   HANDLE        EPhdl[8];       // エンドポイント・ハンドル(パイプハンドル)
} HUSBDEVINF, * PHUSBDEVINF;
```

[Visual Basic]

```
Type spc_mat
    dt(0 To 7)        As Integer
End Type
Type hdl_mat
    dt(0 To 7)        As Long
End Type
Public Type HUSBDEVINF
    BrdID         As Integer        ' ボード ID          ( 0～15 )
    EPcnt         As Integer        ' エンドポイント数       (2)
    EPspc         As spc_mat        ' エンドポイント仕様     (&H2,&H86,0,0)
    EPsiz         As spc_mat        ' エンドポイント・バイト数(64,64, 0, 0)
    EPopt         As spc_mat        ' エンドポイント・オプション( 予約 )
    EPhdl         As hdl_mat        ' エンドポイント・ハンドル(パイプハンドル)
End Type
```

[Visual Basic .NET]

```
Public Structure HUSBDEVINF
    BrdID       As Short      ' ボード ID          ( 0～15 )
    EPcnt       As Short      ' エンドポイント数       (2)
    EPspc0      As Short      ' エンドポイント仕様     (&H2,&H86,0,0)
    EPspc1      As Short
    EPspc2      As Short
    EPspc3      As Short
    EPspc4      As Short
    EPspc5      As Short
    EPspc6      As Short
    EPspc7      As Short
    EPsiz0      As Short      ' エンドポイント・バイト数(64,64, 0, 0)
    EPsiz1      As Short
    EPsiz2      As Short
    EPsiz3      As Short
    EPsiz4      As Short
    EPsiz5      As Short
    EPsiz6      As Short
    EPsiz7      As Short
    EPopt0      As Short      ' エンドポイント・オプション( 予約 )
    EPopt1      As Short
    EPopt2      As Short
    EPopt3      As Short
    EPopt4      As Short
    EPopt5      As Short
    EPopt6      As Short
    EPopt7      As Short
    EPhdl0      As Integer    ' エンドポイント・ハンドル(パイプハンドル)
    EPhdl1      As Integer
    EPhdl2      As Integer
    EPhdl3      As Integer
    EPhdl4      As Integer
    EPhdl5      As Integer
    EPhdl6      As Integer
    EPhdl7      As Integer
End Structure
```

[Visual C#]

```
// デバイス情報
public struct HUSBDEVINF
{
    public ushort BrdID;        // ボード ID            ( 0～15 )
    public ushort EPcnt;        // エンドポイント数       (>=2)
    public ushort EPspc0;       // エンドポイント仕様     (0x81,0x02,0x00,0x00)
    public ushort EPspc1;
    public ushort EPspc2;
    public ushort EPspc3;
    public ushort EPspc4;
    public ushort EPspc5;
    public ushort EPspc6;
    public ushort EPspc7;
    public ushort EPsiz0;       // エンドポイント･バイト数(64,64, 0, 0)
    public ushort EPsiz1;
    public ushort EPsiz2;
    public ushort EPsiz3;
    public ushort EPsiz4;
    public ushort EPsiz5;
    public ushort EPsiz6;
    public ushort EPsiz7;
    public ushort EPopt0;       // エンドポイント･オプション(  予約  )
    public ushort EPopt1;
    public ushort EPopt2;
    public ushort EPopt3;
    public ushort EPopt4;
    public ushort EPopt5;
    public ushort EPopt6;
    public ushort EPopt7;
    public uint EPhdl0;         // エンドポイント･ハンドル(パイプハンドル)
    public uint EPhdl1;
    public uint EPhdl2;
    public uint EPhdl3;
    public uint EPhdl4;
    public uint EPhdl5;
    public uint EPhdl6;
    public uint EPhdl7;
}
```

4.7.2 ドライバ関数の使用

(1) Visual C++（5.0 以上）によるアプリケーションの構築
次のファイルをプロジェクトへ追加します.
■プロジェクト追加ファイル
◇ドライバ 関数用・・**hudio464.lib** ・・ ドライバ関数インポートライブラリ
■インクルードファイル
◇ドライバ 関数用・・**hudio464.h** ・・ ドライバ関数結合用ヘッダーファイル

(2) Visual Basic (5.0/6.0) によるアプリケーションの構築
次のファイルをプロジェクトへ追加します.
◇ドライバ 関数用・・**hudio464.bas** ・・ ドライバI／F用DLL関数定義標準モジュールファイル
このファイルに外部関数宣言（Declare），及びユーザー定義型宣言が記述されています.

(3) Visual Basic .NET によるアプリケーションの構築
次のファイルをプロジェクトへ追加します.
◇ドライバ 関数用・・**hudio464.vb** ・・ ドライバI／F用DLL関数定義標準モジュールファイル
このファイルに外部関数宣言（Declare），及びユーザー定義型宣言が記述されています.

(4) Visual C# によるアプリケーションの構築
次のファイルをプロジェクトへ追加します.
◇ドライバ 関数用・・**hudio464.cs** ・・ ドライバI／F用DLL関数定義標準モジュールファイル
このファイルに外部関数宣言（DllImport），及びユーザー定義型宣言が記述されています.

4.7.3 C++アプリケーションでの使用

ドライバ関数は**「C 言語」**で作成されています．これらの関数をC++アプリケーションで使用できるように，
ドライバ関数のヘッダーファイル内で以下のように「C言語」として明示的な定義をしています.

```
//-------------------------------------------------------
//        関数プロトタイプ宣言
//-------------------------------------------------------
#ifdef __cplusplus
extern "C"
{
#endif

        個々の関数のプロトタイプ宣言

#ifdef __cplusplus
}
#endif
```

(1) #ifdef __cplusplus・・Visual C++ 用の定義です
#endif C++コーディングでは明示的にライブラリ関数が"Cモジュール"として解釈されます.

(2) extern "C" ・・ Cモジュールで定義されている関数を表します.

4.7.4 関数一覧

No	関数名称	機　能
1	dio464_GetDeviceCount ()	ボード枚数の取得
2	dio464_GetDeviceInfo ()	デバイス情報の取得
3	dio464_OpenDevice ()	デバイスオープン
4	dio464_CloseDevice ()	デバイスクローズ
5	dio464_rInB ()	入力ポートからの 1 バイトデータ読込
6	dio464_rOutB ()	出力ポートからの 1 バイトデータ読込
7	dio464_wOutB ()	出力ポートへの　1 バイトデータ書込
8	dio464_rInW ()	入力ポートからの 2 バイトデータ読込
9	dio464_rOutW ()	出力ポートからの 2 バイトデータ読込
10	dio464_wOutW ()	出力ポートへの　2 バイトデータ書込
11	dio464_rInDW ()	入力ポートからの 4 バイトデータ読込
12	dio464_rOutDW ()	出力ポートからの 4 バイトデータ読込
13	dio464_wOutDW ()	出力ポートへの　4バイトデータ書込
14	dio464_InOutDW ()	出力ポートへの 4 バイトデータ書込と入力ポートからの 4 バイトデータ読込
15	dio464_SetFilter ()	入力ポートのﾌｨﾙﾀの設定

表 4.7-1　関数一覧

4.7.5 USB ボードとの通信時間

HUSB-DIO464v2 は［USB2.0 規格　Hi-Speed USB：ﾊﾞﾙｸ通信　］より，ボードとの通信に多少の時間を要します．
ﾄﾞﾗｲﾊﾞ関数の概略所要時間を次の表に示します．（個々のﾊﾟｿｺﾝにより若干差があります．）

No	関数名称	接続される USB ポートが Hi-Speed の場合の通信時間（ ﾏｲｸﾛ秒 ）	接続される USB ポートが Full-Speed の場合の通信時間（ ﾏｲｸﾛ秒 ）
1	dio464_GetDeviceCount ()	----	----
2	dio464_GetDeviceInfo ()	----	----
3	dio464_OpenDevice ()	----	----
4	dio464_CloseDevice ()	----	----
5	dio464_rInB ()	500	4000
6	dio464_rOutB ()	500	4000
7	dio464_wOutB ()	250	2000
8	dio464_rInW ()	500	4000
9	dio464_rOutW ()	500	4000
10	dio464_wOutW ()	250	2000
11	dio464_rInDW ()	500	4000
12	dio464_rOutDW ()	500	4000
13	dio464_wOutDW ()	250	2000
14	dio464_InOutDW ()	500	4000
15	dio464_SetFilter ()	250	2000

表 4.7-2　関数通信時間

注意 1：本表の所要時間値は，ドライバ関数を連続して使用する場合であり，関数単独使用の場合は接続される
　　USB ポートが Hi-Speed の場合　"125 ﾏｲｸﾛ秒"，接続される USB ポートが Full-Speed の場合　"1 ﾐﾘ秒"**減じた値です．**
注意 2：HUSB-DIO464 は接続される USB ポートに関わらず Full-Speed の通信時間になります．
　　(Hi-Speed のハブに接続すると通信時間が早くなることがあります)

4.7.6 ボードアクセスの準備手順

(1) 使用する全ボードのデバイス情報の取得
"HUSBDEVINF"型構造体エリア(の配列)内に，全DIO464のデバイス情報をまず取得します．
◆ dio464_GetDeviceCount()・・・ボード枚数の確認
◆ dio464_GetDeviceInfo() ・・・全ボードのデバイス情報を取得

(2) ボード毎にデバイスオープン
ある1つの DIO464 のデバイス情報をデバイスオープン関数に渡します．
この結果その DIO464 がオープンされ，デバイスオープン関数はこのボードにアクセスするためのデバイスハンドル値を返してきます．
ボード枚数が2枚以上の場合には，個々のボード毎にこの処理を行います．
特定のボードを選択する場合には，ボードID値をチェックして下さい．
◆ dio464_OpenDevice() ・・・ボードのオープン処理
以降は，この「デバイスハンドル」を使用し，そのボードにアクセスすることができるようになります．

(3) 全ての処理が終了してアプリケーションを終了する場合には，オープンしたデバイスの「クローズ処理」を行って下さい．
◆ dio464_CloseDevice() ・・・ボードのクローズ処理(1 枚分)

******* デバイス判別の特例 *******

デバイス情報取得時に，デバイスが認識されない，ボード設定のボード ID が見つからない場合があります．
■次の場合には DIO デバイスとしては認めていません．(デバイス情報なし)
◆ベンダ ID・プロダクト ID が一致しても，デバイスからの情報取得に失敗．
◆デバイスのエンドポイント数が不足． (入出力不可)
◆各エンドポイントの仕様が一致しない．(入出力不可)
■上記の条件を満足しても，次の場合にはボード ID＝99 として，ドライバは処理しています．
◆DIO ボードの持つ一部の機能が認識できない．

4.8 関数の戻り値

ドライバの関数を使用する時，関数の戻り値が異常値（'0'以外）であった場合には，異常内容に対応した処理を行います．
通常，この異常が発生した場合にはアプリケーションプログラムの続行は困難であり，プログラム内容の再検討が必要となります．

No	戻り値			異 常 内 容 と 確 認 項 目
	記号表記	16 進数表記		
		VC++, VC#	VB, VB.NET	
1	NO_ERROR	0x000	&H0	正　常 異常は発生していません
2	NOT_FOUND	0x001	&H1	デバイスドライバが存在しない ◎デバイスドライバがインストールされていない ◎デバイスドライバが所定のフォルダに格納されていない
3	ALREADY_OPENED	0x002	&H2	既にオープン済のデバイスをオープン ◎オープン済みデバイスに更にオープン指令 ◇オープンしたデバイスはクローズするまで使用 （多重のオープンは禁止） ◎ボード2枚以上使用する場合,オープンするデバイス情報の更新を確認します．
4	NOT_MEMORY	0x004	&H4	デバイス情報格納メモリが不足 ◎アプリケーション用のメモリ不足 ◇パソコン主記憶メモリの不足 ◎システムリソース（OS用メモリ）の不足 ◇多数のアプリケーション起動 ◇1度に多数のウインドウを開いた
5	INVALID_HANDLE	0x008	&H8	無効なデバイスハンドルを指定 ◎デバイスオープンで得られた"デバイスハンドル"の不使用 ◎このデバイスは既にクローズされている
6	NOT_READY	0x010	&H10	デバイスの入出力ポートが使用できない ◎デバイス（ボード）内部の入出力ポートがない システムとの不整合等が考えられますので，弊社サポートまでお問い合わせください
7	ILLEGAL_DEVICE	0x020	&H20	ボード固有情報が不正 ◎デバイス情報は正常ですが,ボード機能が不一致です
8	ILLEGAL_PARAM	0x100	&H100	関数の引数の値が異常 ◇引数の設定値を確認（マニュアル照合）
9	DEVICE_STALL	0x80000000	&H80000000	通信異常

表 4.8-1　関数の戻り値

4.9 ドライバ I/F 用 DLL 関数詳細

(1) dio464_GetDeviceCount()　　ボード枚数の取得

機　能	現在パソコンに装着されている DIO464 の枚数を取得します.

開発環境	書　式
VC++	DWORD WINAPI dio464_GetDeviceCount(DWORD*　count);
VB6	Declare Function dio464_GetDeviceCount Lib "hudio464.dll" (ByRef count As Long) As Long
VB.NET	Declare Function dio464_GetDeviceCount Lib "hudio464.dll" (ByRef count As Integer) As Integer
VC#	[DllImport("Hudio464.dll")] public static extern uint dio464_GetDeviceCount(ref uint count);

引　数	説　明
count	取得した DIO464 枚数

VC++ 記述例	DWORD*　count;　　//DIO464 の枚数 DWORD　　ret;　　//関数の戻り値 ret = dio464_GetDeviceCount(&count);

備　考	

(2) dio464_ GetDeviceInfo()　　デバイス情報の取得

機　能	現在パソコンに装着されている指定枚数 DIO464 のデバイス情報を取得します. この結果，デバイス情報構造体の配列にデバイス情報が格納されます. このデバイス情報は，デバイスオープン時に利用します.

開発環境	書　式
VC++	DWORD WINAPI dio464_GetDeviceInfo(DWORD* count, HUSBDEVINF* HusbInf);
VB6	Declare Function dio464_GetDeviceInfo Lib "hudio464.dll" _ (ByRef count As Long,HusbInf As HUSBDEVINF) As Long
VB.NET	Declare Function dio464_GetDeviceInfo Lib "hudio464.dll" _ (ByRef count As Integer, ByRef HusbInf As HUSBDEVINF) As Integer
VC#	[DllImport("Hudio464.dll")] public static extern uint dio464_GetDeviceInfo(ref uint count, ref HUSBDEVINF HusbInf);

引　数	説　明
count	取得した DIO464 枚数
HusbInf	取得するデバイス情報

VC++ 記述例	DWORD　　ret;　　//関数の戻り値 DWORD　　count = 2;　　//使用枚数は 2 HUSBDEVINF　HusbInf[2];　　//2 枚の DIO464 のデバイス情報がセットされるべきエリア ret = dio464_GetDeviceInfo(&count, &HusbInf[0]);

備　考	

(3) dio464_ OpenDevice()　　デバイスのオープン

機　能	渡したデバイス情報を持つ DIO464 をオープンし，他と識別するためのデバイスハンドルを取得します． 以降このデバイスハンドルは，この DIO464 にアクセスするためのハンドルとなります． また，この時出力ポートはすべて"0"になります．

開発環境	書　式
VC++	DWORD WINAPI dio464_OpenDevice(DWORD * hDev, HUSBDEVINF* HusbInf);
VB6	Declare Function dio464_OpenDevice Lib "hudio464.dll" (ByRef hDev As Long, HusbInf As HUSBDEVINF) As Long
VB.NET	Declare Function dio464_OpenDevice Lib "hudio464.dll" (ByRef hDev As Integer, ByRef HusbInf As HUSBDEVINF) As Integer
VC#	[DllImport("Hudio464.dll")] public static extern uint dio464_OpenDevice(ref uint hDev, ref HUSBDEVINF HusbInf);

引　数	説　明
hDev	取得するデバイスハンドル
HusbInf	オープンするボードのデバイス情報

VC++ 記述例	DWORD*　count;　　//DIO464 の枚数 DWORD　ret;　　//関数の戻り値 ret = dio464_GetDeviceCount(&count);

備　考	

(4) dio464_ GetDeviceInfo()　　デバイスのクローズ

機　能	渡したデバイスハンドルを持つ DIO464 をクローズします． 以降このデバイスハンドルは，無効となり，この DIO464 にアクセスはできません． また，出力ポートの状態は最終出力状態が保持されます．

開発環境	書　式
VC++	DWORD WINAPI dio464_CloseDevice(DWORD hDev);
VB6	Declare Function dio464_CloseDevice Lib "hudio464.dll" (ByVal hDev As Long) As Long
VB.NET	Declare Function dio464_CloseDevice Lib "hudio464.dll" (ByVal hDev As Integer) As Integer
VC#	[DllImport("Hudio464.dll")] public static extern uint dio464_CloseDevice(uint hDev);

引　数	説　明
hDev	クローズするボードのデバイスハンドル

VC++ 記述例	DWORD　ret;　　//関数の戻り値 ret = dio464_CloseDevice(hDev);

備　考	

(5) dio464_rInB()　　入力ポートからの 1 バイトデータ読込

(6) dio464_rOutB()　　出力ポートからの 1 バイトデータ読込

(7) dio464_wOutB()　　出力ポートへの 1 バイトデータ書込

(8) dio464_rInW()　　入力ポートからの 2 バイトデータ読込

(9) dio464_rOutW()　　出力ポートからの 2 バイトデータ読込

(10)　dio464_wOutW()　出力ポートへの 2 バイトデータ書込

機　能	デバイスハンドルで指定された DIO464 の指定された入力ポートまたは出力ポートから 入力ポートからの1バイトデータ読込・・1 バイトデータを読込み指定したエリアに格納します. 出力ポートからの1バイトデータ読込・・1 バイトデータを読込み指定したエリアに格納します. 出力ポートへの1バイトデータ書込　・・1 バイトデータを書込ます. 入力ポートからの2バイトデータ読込・・2 バイトデータを読込み指定したエリアに格納します. 出力ポートからの2バイトデータ読込・・2 バイトデータを読込み指定したエリアに格納します. 出力ポートへの2バイトデータ書込　・・2 バイトデータを書込ます. 出力ポートの読込により最終出力状態の確認ができます.

開発環境	書　式
VC++	DWORD WINAPI dio464_rInB (DWORD hDev, WORD port, WORD*　wrdt); DWORD WINAPI dio464_rOutB(DWORD hDev, WORD port, WORD*　wrdt); DWORD WINAPI dio464_wOutB(DWORD hDev, WORD port, WORD　 wwdt); DWORD WINAPI dio464_rInW (DWORD hDev, WORD port, WORD*　wrdt); DWORD WINAPI dio464_rOutW(DWORD hDev, WORD port, WORD*　wrdt); DWORD WINAPI dio464_wOutW(DWORD hDev, WORD port, WORD　 wwdt);
VB6	Declare Function dio464_rInB Lib "hudio464.dll" (ByVal hDev As Long, ByVal port As Integer, _ ByRef wrdt As Integer) As Long Declare Function dio464_rOutB Lib "hudio464.dll" (ByVal hDev As Long, ByVal port As Integer, _ ByRef wrdt As Integer) As Long Declare Function dio464_wOutB Lib "hudio464.dll" (ByVal hDev As Long, ByVal port As Integer, _ ByVal wwdt As Integer) As Long Declare Function dio464_rInW Lib "hudio464.dll" (ByVal hDev As Long, ByVal port As Integer, _ ByRef wrdt As Integer) As Long Declare Function dio464_rOutW Lib "hudio464.dll" (ByVal hDev As Long, ByVal port As Integer, _ ByRef wrdt As Integer) As Long Declare Function dio464_wOutW Lib "hudio464.dll" (ByVal hDev As Long, ByVal port As Integer, _ ByVal wwdt As Integer) As Long
VB.NET	Declare Function dio464_rInB Lib "hudio464.dll" (ByVal hDev As Integer, ByVal port As Short, _ ByRef wrdt As Short) As Integer Declare Function dio464_rOutB Lib "hudio464.dll" (ByVal hDev As Integer, ByVal port As Short, _ ByRef wrdt As Short) As Integer Declare Function dio464_wOutB Lib "hudio464.dll" (ByVal hDev As Integer, ByVal port As Short, _ ByVal wwdt As Short) As Integer Declare Function dio464_rInW Lib "hudio464.dll" (ByVal hDev As Integer, ByVal port As Short, _ ByRef wrdt As Short) As Integer Declare Function dio464_rOutW Lib "hudio464.dll" (ByVal hDev As Integer, ByVal port As Short, _ ByRef wrdt As Short) As Integer Declare Function dio464_wOutW Lib "hudio464.dll" (ByVal hDev As Integer, ByVal port As Short, _ ByVal wwdt As Short) As Integer
VC#	[DllImport("Hudio464.dll")] public static extern uint dio464_rInB(uint hDev, ushort port, ref ushort wrdt); [DllImport("Hudio464.dll")] public static extern uint dio464_rOutB(uint hDev, ushort port, ref ushort wrdt); [DllImport("Hudio464.dll")] public static extern uint dio464_wOutB(uint hDev, ushort port, ushort wwdt); [DllImport("Hudio464.dll")] public static extern uint dio464_rInW(uint hDev, ushort port, ref ushort wwdt); [DllImport("Hudio464.dll")] public static extern uint dio464_rOutW(uint hDev, ushort port, ref ushort wwdt); [DllImport("Hudio464.dll")] public static extern uint dio464_wOutW(uint hDev, ushort port, ushort wwdt);

次ページに続く

前ページからの続き

引　数	説　明
hDev	対象デバイスのデバイスハンドル
port	ポート指定(0:ポート 1, 1:ポート 2, 2:ポート 3, 3:ポート 4)
wrdt	読込んだデータを格納する2バイトエリアのアドレス
wwdt	書込む2バイトデータ

VC++
記述例
	DWORD ret; //関数の戻り値
	WORD dt; //データ格納エリア
	ret = dio464_rInB(hDev, 0, &dt); //入力ポート 1 の読込

備　考	1バイトの入出力関数の場合 読込データは 上位バイト:0x00 下位バイト:読込データ 書込データは 上位バイト:無視 下位バイト:書込データ

(11) dio464_rInDW()　　入力ポートからの 4 バイトデータ読込

(12) dio464_rOutDW()　　出力ポートからの 4 バイトデータ読込

(13) dio464_wOutDW()　出力ポートへの 4 バイトデータ書込

機　能	デバイスハンドルで指定された DIO464 の入力ポートまたは出力ポートから 入力ポートからの 4 バイトデータ読込・・4 バイトデータを読込み指定したエリアに格納します. 出力ポートからの 4 バイトデータ読込・・4 バイトデータを読込み指定したエリアに格納します. 出力ポートへの 4 バイトデータ書込　・・4 バイトデータを書込ます. 出力ポートの読込により最終出力状態の確認ができます.

開発環境	書　式
VC++	DWORD WINAPI dio464_rInDW (DWORD hDev, DWORD*　drdt); DWORD WINAPI dio464_rOutDW(DWORD hDev, DWORD*　drdt); DWORD WINAPI dio464_wOutDW(DWORD hDev, DWORD　 dwdt);
VB6	Declare Function dio464_rInDW Lib "hudio464.dll" _ (ByVal hDev As Long, ByRef drdt As Long) As Long Declare Function dio464_rOutDW Lib "hudio464.dll" _ (ByVal hDev As Long, ByRef drdt As Long) As Long Declare Function dio464_wOutDW Lib "hudio464.dll" _ (ByVal hDev As Long, ByVal dwdt As Long) As Long
VB.NET	Declare Function dio464_rInDW Lib "hudio464.dll" _ (ByVal hDev As Long, ByRef drdt As Integer) As Integer Declare Function dio464_rOutDW Lib "hudio464.dll" _ (ByVal hDev As Integer, ByRef drdt As Integer) As Integer Declare Function dio464_wOutDW Lib "hudio464.dll" _ (ByVal hDev As Integer, ByVal dwdt As Integer) As Integer
VC#	[DllImport("Hudio464.dll")] public static extern uint dio464_rInDW(uint hDev, ref uint drdt); [DllImport("Hudio464.dll")] public static extern uint dio464_rOutDW(uint hDev, ref uint drdt); [DllImport("Hudio464.dll")] public static extern uint dio464_wOutDW(uint hDev, uint dwdt);

引　数	説　明
hDev	対象デバイスのデバイスハンドル
drdt	読込んだデータを格納する 4 バイトエリアのアドレス
dwdt	書込む 4 バイトデータ

VC++ 記述例	DWORD　ret;　　//関数の戻り値 DWORD　dt;　　//データ格納エリア ret = dio464_rInDW(hDev, &dt);　//入力ポート読込

備　考	1バイトの入出力関数の場合 読込データは　　上位バイト:0x00 　　下位バイト:読込データ 書込データは　　上位バイト:無視 　　下位バイト:書込データ

(14) dio464_InOutDW()

DIO の出力ポートへ 4 バイトデータ書込と DIO 入力ポートからの 4 バイトデータ読込

機　能	デバイスハンドルで指定された DIO464 の出力ポートへ 4 バイトデータを書込ます. また出力直前の入力ポートから4バイトデータを読込み，指定したエリアに格納します.

開発環境	書　式
VC++	DWORD WINAPI dio464_InOutDW (DWORD hDev, DWORD* drdt, DWORD dwdt);
VB6	Declare Function dio464_ InOutDW Lib "hudio464.dll" (ByVal hDev As Long, _ ByRef drdt As Long _ ByVal dwdt As Long) As Long
VB.NET	Declare Function dio464_ InOutDW Lib "hudio464.dll" (ByVal hDev As Integer, _ ByRef drdt As Integer _ ByVal dwdt As Integer) As Integer
VC#	[DllImport("Hudio464.dll")] public static extern uint dio464_InOutDW(uint hDev, ref uint drdt, uint dwdt);

引　数	説　明
hDev	対象デバイスのデバイスハンドル
drdt	読込んだデータを格納する 4 バイトエリアのアドレス
dwdt	書込む 4 バイトデータ

VC++
記述例
	DWORD ret; //関数の戻り値
	DWORD rdt; //データ格納エリア
	ret = dio464_InOutDW(hDev, &rdt, 0x55555555); //55555555h のデータ出力と入力ポート読込

備　考	

(15) dio464_SetFilter()　入力ポートのフィルタの設定

機　能	デバイスハンドルで指定された DIO464 の指定された入力ポートのフィルタを設定します

開発環境	書　式
VC++	DWORD WINAPI dio464_SetFilter (DWORD hDev, WORD port, DWORD wtm);
VB6	Declare Function dio464_SetFilter Lib "hudio464.dll" (ByVal hDev As Long, _ ByVal port As Integer _ ByVal wtm As Long) As Long
VB.NET	Declare Function dio464_SetFilter Lib "hudio464.dll" (ByVal hDev As Integer, _ ByVal port As Short _ ByVal wtm As Integer) As Integer
VC#	[DllImport("Hudio464.dll")] public static extern uint dio464_SetFilter(uint hDev, ushort port, ushort wtm);

引　数	説　明
hDev	対象デバイスのデバイスハンドル
port	ポート指定 (0x1：ポート 1，0x2：ポート 2，0x4：ポート 3，0x8：ポート 4)
wtm	フィルタ時間 (wtm × 0.1msec)

VC++
記述例
	DWORD ret; //関数の戻り値
	ret = dio464_ SetFilter(hDev, 0x1, 100); //ポート 1 に 10msec のフィルタ設定

備　考	ポート指定が複数の場合は OR したデータを与えます. wtm の設定範囲は 1 ～ 65535 (ffffh) になります. 従って，サンプリング時間の設定範囲は 0.1msec ～ 6553.5msec になります. また，入力に変化が発生してから，ボード内に取り込まれた入力状態が確定するまでに， 最小 wtm ～ 最大 2 × wtm の時間を要します.

4.10　サンプルプログラム

ドライバ関数の使用方法を解説する目的のサンプルプログラムを添付しています.
サンプルプログラムは次の2種類があり, ほぼ同一の画面表示と操作になっています.
以降のサンプルプログラム説明では, (1) の「Cコーディング」を用います.

(1) Visual C++　6.0	【 sud46400.exe 】
(2) Visual Basic 6.0	【 sud46402.exe 】
(3) Visual Basic.NET2003	【 sud46403.exe 】
(4) Visual C#.NET2003	【 sud46404.exe 】

サンプルプログラムを使用する場合は, お客様のハードディスクにコピーして使用します.
個々のサンプル実行ファイルは"マウスのダブルクリック"操作を行う事で実行できます.

≪ ご注意 ≫

(1) Visual C++ サンプルは開発ツールとして Visual C++ 6.0 以上がインストールされている必要があります.
VC.NET2002, VC.NET2003, VC2005, VC2008 を使用する場合, 本サンプルプログラムを開くと変換ウィザードが起動しますので, それに従って自動で変換を行ってください.
また, 開発ツールとして Visual C++ 5.0 を使用している場合はプロジェクトを作成する必要があります.

(2) Visual Basic 6.0 サンプルは開発ツールとして Visual Basic 6.0 がインストールされている必要があります.
また, 開発ツールとして Visual Basic 5.0 を使用している場合はプロジェクトを作成する必要があります.

(3) Visual Basic .NET2003 サンプルは開発ツールとして Visual Basic .NET2003 以上 がインストールされている必要があります.
VB2005, VB2008 を使用する場合, 本サンプルプログラムを開くと変換ウィザードが起動しますので, それに従って自動で変換を行って下さい. 不十分な部分は手動で変換を行ってください.
また, 開発ツールとして Visual Basic .NET2002 を使用している場合はプロジェクトを作成する必要があります.

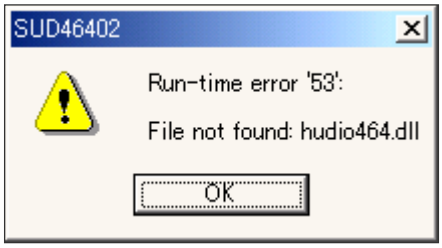
(4) Visual C#.NET2003 サンプルは開発ツールとして Visual C# .NET2003 以上 がインストールされている必要があります.
VC#2005, VC#2008 を使用する場合, 本サンプルプログラムを開くと変換ウィザードが起動しますので, それに従って自動で変換を行って下さい. 不十分な部分は手動で変換を行ってください.
また, 開発ツールとして Visual C# .NET2002 を使用している場合はプロジェクトを作成する必要があります.

(5) DIO464 を 2 枚以上で使用する場合, ボード ID は重複しないようにして下さい.

(6) 実行開始時に次頁のエラーメッセージが表示される場合には,プログラムは動作しません.

(7) Visual Basic .NET サンプル実行ファイル(sud46403.exe), Visual C#.NET(sud46404.exe)を動作させるためには, .NET Framework 2.0 以上をインストールする必要があります.

【 エラーメッセージの表示 】



※ DLL がインストールされていない.
DLL をインストールして下さい.

※ DIO464 が装着されていない. または, システムが認識していない.
※ デバイスドライバがインストールされていない.
ボードを装着してください. またはデバイスドライバをインストールしてください.

4.10.1 サンプルプログラムの操作

サンプルプログラムが正常に起動されると,図のような画面が表示されます.

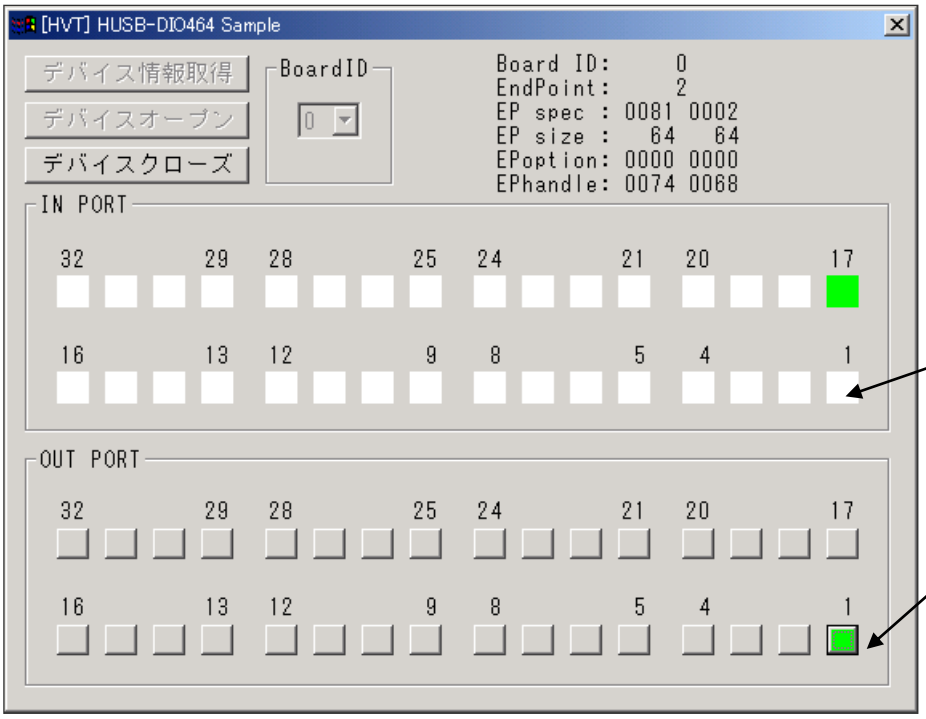
(1) ボード(デバイス)の選択

サンプルプログラムでは，ボード上の動作操作開始・終了は次の手順に従います.

①デバイス情報取得
②ボードの選択(2 枚以上の場合)
③デバイスオープン
④デバイスクローズ(全出力ポート(1-4)には'0'を書込みます)
⑤サンプルプログラムの終了(デバイスクローズ後)

(2) ボード上の操作と表示

デバイスオープンを行いますと次の画面となります.



入力表示(読込)
入力状態と表示色
入力中 ・・緑
入力なし・・白

出力ボタン(書込)
ON/OFF 切替

出力表示(読込)
出力状態と表示色
出力中 ・・緑
出力なし・・なし (背景色)