

USBインターフェース  
32IN/32OUT DIOボード

# HUSB-DIO464 HUSB-DIO464(D)

## 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

**D I O 4 6 4 ・ ・ U S B   b u s**

H U S B - D I O 4 6 4

H U S B - D I O 4 6 4 ( D ) に適応しています。

---

本書及びプログラムの全部又は一部の無断転載、コピーを禁止します。  
本製品の内容に関しましては、改良等により将来予告なしに変更することがあります。  
本製品の内容についてお気づきの点がございましたら、お手数ながら当社までご連絡下さい。

Windows95, Windows98, WindowsNT 4.0, Windows2000, WindowsXP Home Edition, WindowsXP Professional, VisualC++,  
Visual Basic, Microsoft C/C++ は Microsoft Corporation の米国及びその他の国における登録商標です。  
その他、記載されている会社名、製品名は、各社の商標又は登録商標です。

株式会社 ハイバーテック  
東京都墨田区両国4-8-1  
ダイユービル  
TEL 03-3846-3801  
FAX 03-3846-3773  
sales@hivertec.co.jp

第1.0版 2003年 1月28日発行  
不許複製・転載

## 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。

## 免責事項

1. 本マニュアルに記載された内容に沿わない、製品の取り付け、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す恐れのある装置（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. 本マニュアル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。

## 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。本書は、本製品をご使用して頂く場合の取り扱い、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、本書が添付された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、誤った取り扱いをすると、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、誤った取り扱いをすると、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. 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ついて基本的な知識を有している方.

## 2. 適合BUS



### 注意



本製品は Universal Serial Bus 1.1 に適合したボードです。

## 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°C ~ +50°C
- ・ 動作周囲湿度 20%RH ~ 85%RH (結露せぬこと)
- ・ 保存周囲温度 -15°C ~ +75°C
- ・ 保存周囲湿度 10%RH ~ 90%RH (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 海拔3000m以下  
(3000m毎に2°Cの上限値を下げた範囲で使用して下さい)

## 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(エアークラップなど)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。重いものを乗せると、部品が損傷し故障の原因になります。



本製品のジャンパー及びディップスイッチの設定は、パソコン等に接続する前に行ってください。  
電源がONの状態を設定しますと、設定を正しく認識しないで誤動作の原因になります。



本製品のジャンパー及びディップスイッチの設定は、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コネクタの抜き差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、動作中に抜けたり、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












本製品を落とすなど乱暴に扱わないで下さい。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。



## 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にし、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、装置が思わ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、コネクタが外れる、または接触不良などにより誤動作の原因となります。
	外部配線用ケーブルは、引っ張る、または重い荷重を掛けないで下さい。 コネクタが外れる、または接触不良などにより誤動作の原因となります。
	外部配線用ケーブルは、モーターの配線や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。

## 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ると、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。

## 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。



# 目 次

1. はじめに	1
1. 1 このマニュアルについて	1
1. 2 添付ソフトウェア	1
2. ハードウェア編	2
2. 1 HUSB-DIO464及びHUSB-DIO464 (D) 仕様	2
2. 2 HUSB-DIO464及びHUSB-DIO464 (D) のボード構造	3
2. 2. 1 HUSB-DIO464 (D) 呼称	3
2. 2. 2 ボード形寸	4
2. 3 ブロック図	5
2. 4 入出力回路	5
2. 5 ポート構成	6
2. 6 ボード上の設定	7
2. 7 コネクタ信号表	8
2. 8 外部との接続	10
2. 8. 1 出力回路接続例	10
2. 8. 2 入力回路接続例	11
2. 8. 3 外部接続にあたっての注意事項	11
3. ソフトウェア編	12
3. 1 対応OS, ドライバ種別	12
3. 2 添付ソフトウェアの構成	12
3. 3 デバイスドライバのインストール	13
3. 4 デバイスドライバの更新	18
3. 4. 1 ドライバの更新	18
3. 4. 2 ドライバI/Fライブラリの更新	19
3. 5 ボードを複数枚使用する場合	23
3. 6 USBボードの認識	23
3. 7 ボードアクセス方法	24
3. 7. 1 ボード (デバイス) 認識用のデータ構造体	24
3. 7. 2 ドライバ関数の使用	25
3. 7. 3 C++アプリケーションでの使用	25
3. 7. 4 関数一覧	26
3. 7. 5 USBボードとの通信時間	26
3. 7. 6 ボードアクセスの準備手順	27
3. 8 ドライバI/F用DLL関数詳細	28
3. 9 関数の戻り値	33
3. 10 サンプルプログラム	34
3. 10. 1 サンプルプログラムの操作	35

# 目 次

1.	はじめに	1
2.	ハードウェア編	2
表2.	1-1 HUSB-DIO464仕様	2
写真2.	2-1 HUSB-DIO464	3
図2.	2-1 ボード形寸	4
図2.	2-2 DINレール取付台形寸	4
図2.	3-1 HUSB-DIO464ブロックダイア	5
図2.	4-1 出力回路	5
図2.	4-2 入力回路	5
表2.	5-1 ポート構成	6
図2.	6-1 ジャンパ設定箇所	7
図2.	6-2 P1ジャンパ	7
図2.	6-3 ボードIDの設定	7
表2.	7-1 出力コネクタ(J1)ピン配列	8
表2.	7-2 入力コネクタ(J2)ピン配列	9
表2.	7-3 TB1電源端子	9
図2.	8-1 出力回路接続例	10
図2.	8-2 シンクタイプとの接続	10
図2.	8-3 外部接続における対策	11
3.	ソフトウェア編	12
図3.	3-1 デバイスドライバインストール後の確認画面	14
図3.	3-2 WinXPのインストール手順	15
図3.	3-3 Win2Kのインストール手順	16
図3.	3-4 Win98のインストール手順	17
図3.	4-1 WinXPドライバの更新	20
図3.	4-2 Win2Kドライバの更新	21
図3.	4-3 Win98ドライバの更新	22
表3.	7-1 関数一覧	26
表3.	7-2 関数通信時間	26

## 1. はじめに

このマニュアルはUSB 適合の デジタル入出力ボードHUSB-DIO464及びHUSB-DIO464 (D)の取扱説明書です。 (※)

※HUSB-DIO464 (D) ボードはHUSB-DIO464ボードに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-DIO464またはDIO464と呼びます。

■ HUSB-DIO464は USB 1. 1 規格に基づいています。

### 1. 1 このマニュアルについて

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

### 1. 2 添付ソフトウェア

このボードには次のソフトウェアが添付されます。

Windows版

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。

## 2. ハードウェア編

### 2.1 HUSB-DIO464及びHUSB-DIO464 (D)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°C~50°C 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2.1-1 HUSB-DIO464仕様

## 2. 2 HUSB-DIO464及びHUSB-DIO464 (D) のボード構造

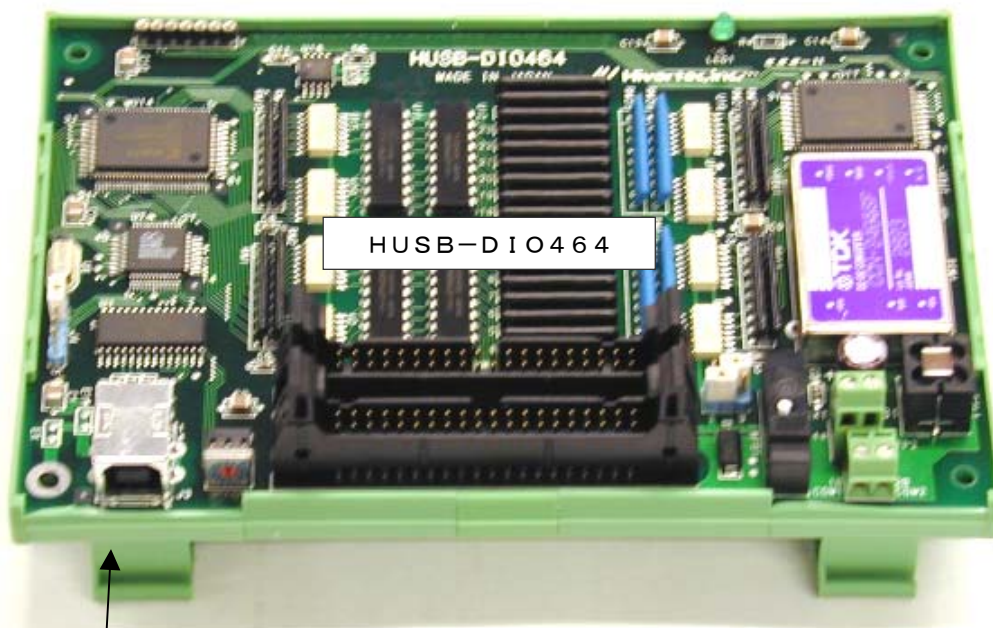


### USBコネクタの抜き挿しの注意

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. 2. 1 HUSB-DIO464 (D) 呼称

■ HUSB-DIO464 (D) の各部の呼称を 写真2. 2-1 に示します。



DINレール取付台 DIO464 (D)

写真2. 2-1 HUSB-DIO464 (D)

## 2. 2. 2 ボード形寸

### (1) HUSB-DIO464ボード形寸

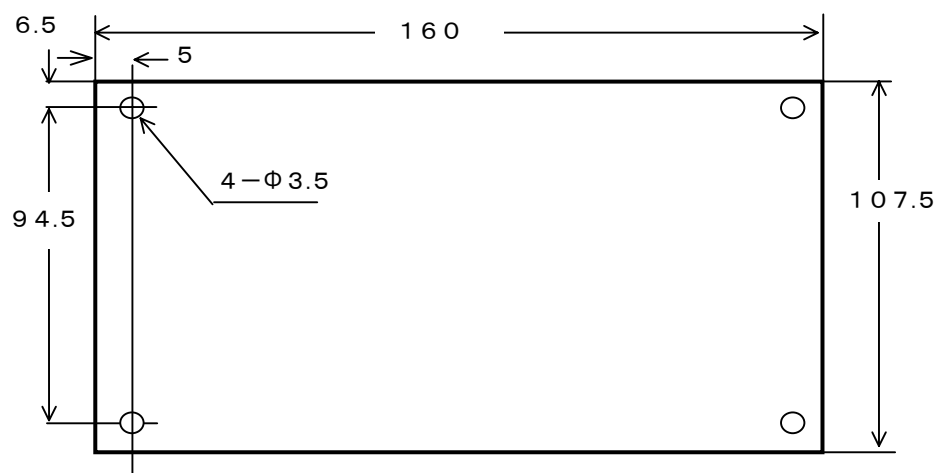


図2. 2-1 ボード形寸

### (2) DIN取付台形寸

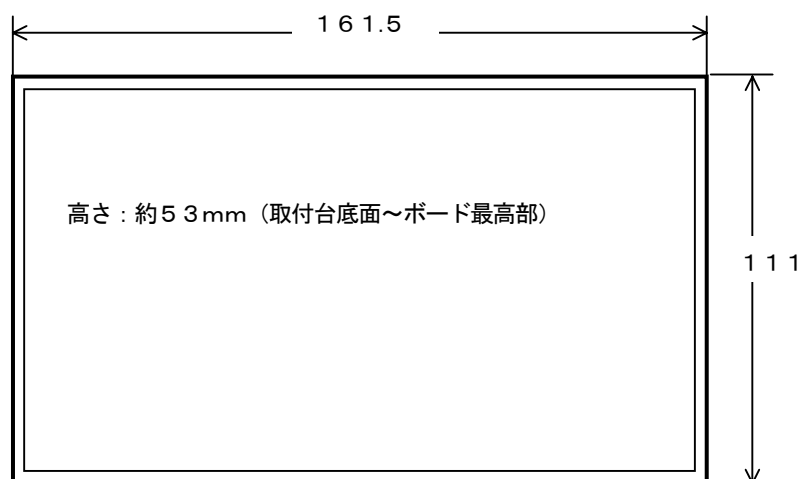


図2. 2-2 DINレール取付台形寸

## 2.3 ブロック図

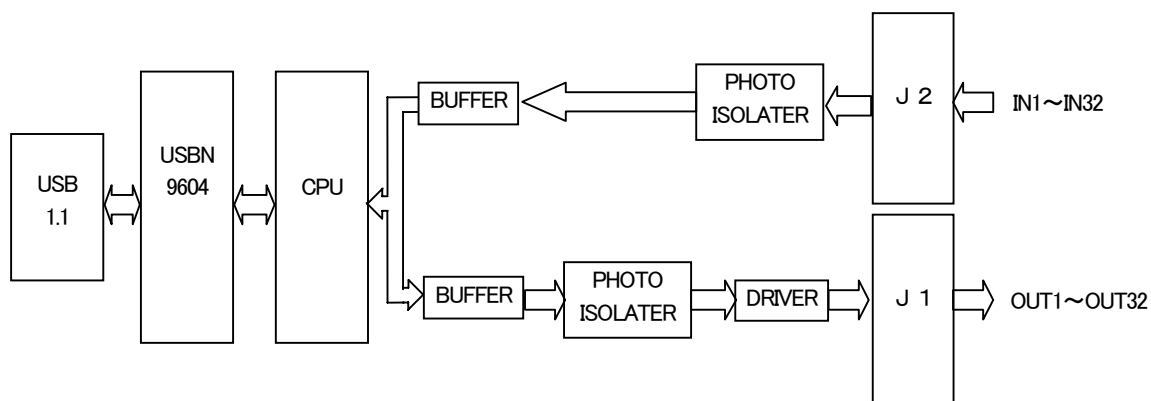


図2. 3-1 HUSB-DIO464ブロックダイア

## 2.4 入出力回路

(1) 本ボードの出力回路を図2. 4-1に示します。

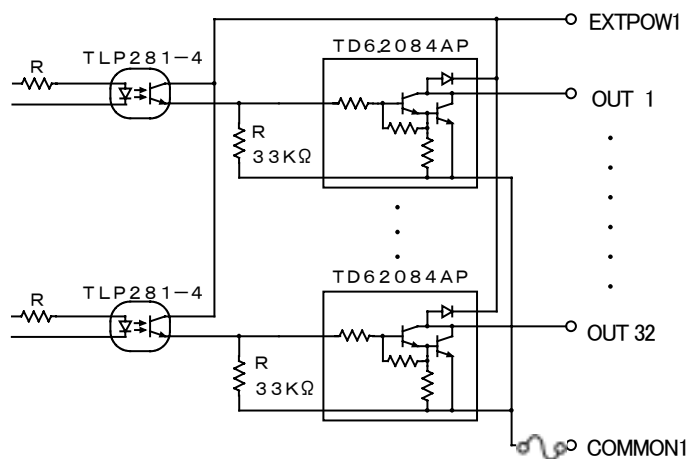


図2. 4-1 出力回路

(2) 本ボードの入力回路を図2. 4-2に示します。

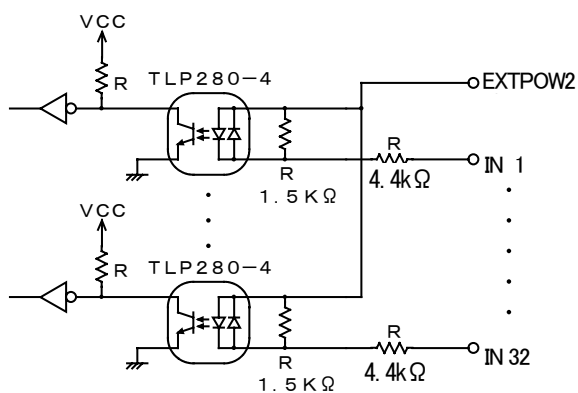


図2. 4-2 入力回路

## 2. 5 ポート構成

本ボードのポート構成を表2. 5-1に示します.

ポート名	ビット	入力(I N)	ポート名	ビット	入力(I N)	出力(O U T)
入力ポート 1	0	IN 1	出力ポート 1	0	OUT 1	OUT 1
	1	IN 2		1	OUT 2	OUT 2
	2	IN 3		2	OUT 3	OUT 3
	3	IN 4		3	OUT 4	OUT 4
	4	IN 5		4	OUT 5	OUT 5
	5	IN 6		5	OUT 6	OUT 6
	6	IN 7		6	OUT 7	OUT 7
	7	IN 8		7	OUT 8	OUT 8
入力ポート 2	0	IN 9	出力ポート 2	0	OUT 9	OUT 9
	1	IN10		1	OUT10	OUT10
	2	IN11		2	OUT11	OUT11
	3	IN12		3	OUT12	OUT12
	4	IN13		4	OUT13	OUT13
	5	IN14		5	OUT14	OUT14
	6	IN15		6	OUT15	OUT15
	7	IN16		7	OUT16	OUT16
入力ポート 3	0	IN17	出力ポート 3	0	OUT17	OUT17
	1	IN18		1	OUT18	OUT18
	2	IN19		2	OUT19	OUT19
	3	IN20		3	OUT20	OUT20
	4	IN21		4	OUT21	OUT21
	5	IN22		5	OUT22	OUT22
	6	IN23		6	OUT23	OUT23
	7	IN24		7	OUT24	OUT24
入力ポート 4	0	IN25	出力ポート 4	0	OUT25	OUT25
	1	IN26		1	OUT26	OUT26
	2	IN27		2	OUT27	OUT27
	3	IN28		3	OUT28	OUT28
	4	IN29		4	OUT29	OUT29
	5	IN30		5	OUT30	OUT30
	6	IN31		6	OUT31	OUT31
	7	IN32		7	OUT32	OUT32

表2. 5-1 ポート構成

## 2. 6 ボード上の設定

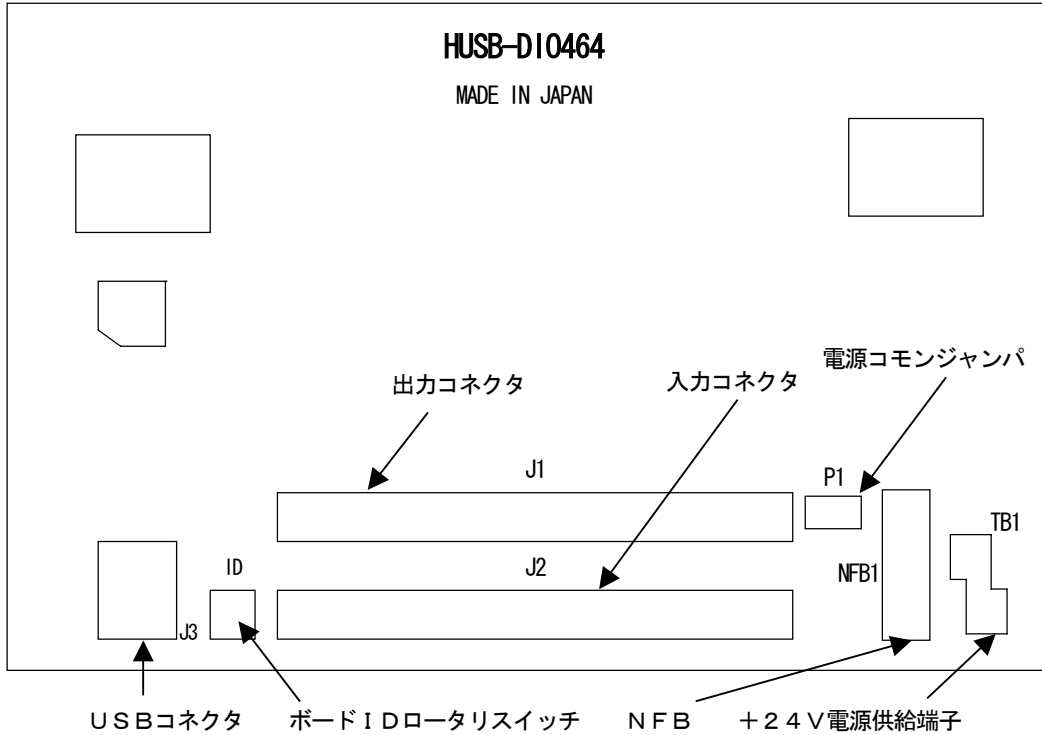


図2. 6-1 ジャンパ設定箇所

### (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。

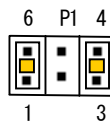
P1ジャンパの出荷時の設定は1-6接続, 3-4接続です。

(COMMON1とCOMMON2共通, EXTPOW1とEXTPOW2共通)

TB1接続は「表2. 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。



COMMON1とCOMMON2共通, EXTPOW1とEXTPOW2共通(出荷時の設定)

図2. 6-2 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-D10464 ボードを識別する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2. 6-3 ボードIDの設定(本図は“F”設定例)

## 2. 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表2. 7-1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	4	OUT 2	出力2
5	OUT 3	出力3	6	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	COMMON1	24V 用コモン (GND)	20	COMMON1	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	COMMON1	24V 用コモン (GND)	40	COMMON1	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COMMON1 (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2. 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表2. 7-2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	4	IN 2	入力2
5	IN 3	入力3	6	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	COMMON2	24V用コモン (GND)	20	COMMON2	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	COMMON2	24V用コモン (GND)	40	COMMON2	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2. 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表2. 7-3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.
1B	COMMON1	(同GND)	
2A	EXTPOW2	(+24V外部電源供給)	同様に EXTPOW1 と EXTPOW2 は P1 の1-6 接続により共通になります.
2B	COMMON2	(同GND)	

表2. 7-3 TB1 電源端子

## 2. 8 外部との接続

以下に入出力回路の一般的な接続例を図示します。

### 2. 8. 1 出力回路接続例

本ボードの出力回路の接続例を図2. 8-1 に示します。

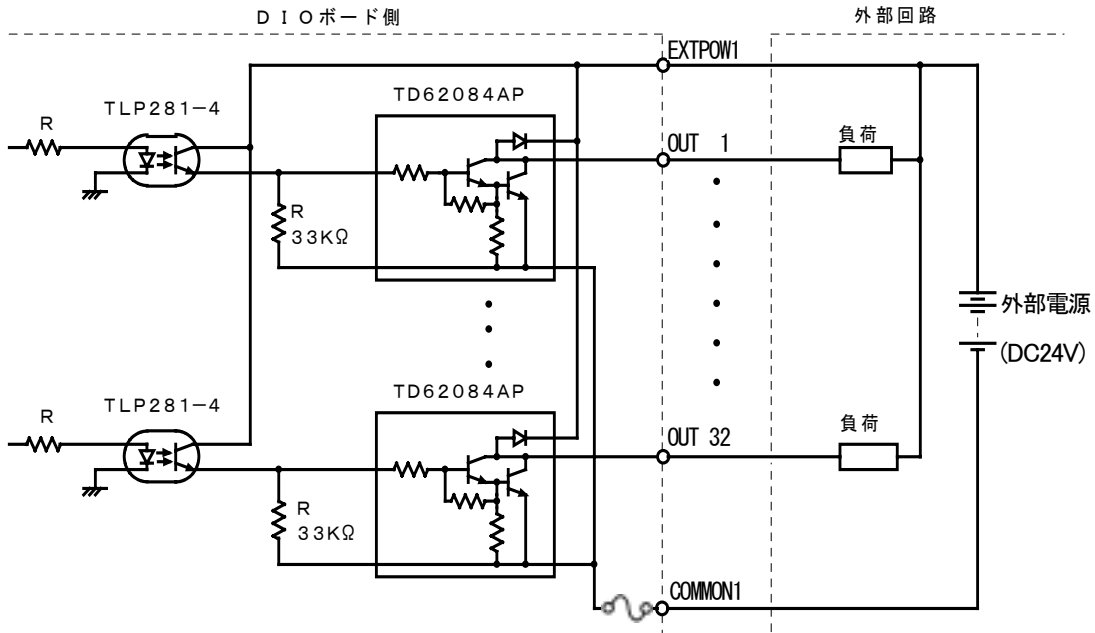


図2. 8-1 出力回路接続例

### 2. 8. 2 入力回路接続例

本ボードの入力回路と外部回路がシンクタイプとの接続例を図2. 8-2に示します。

#### (1) 外部回路がシンクタイプとの接続例

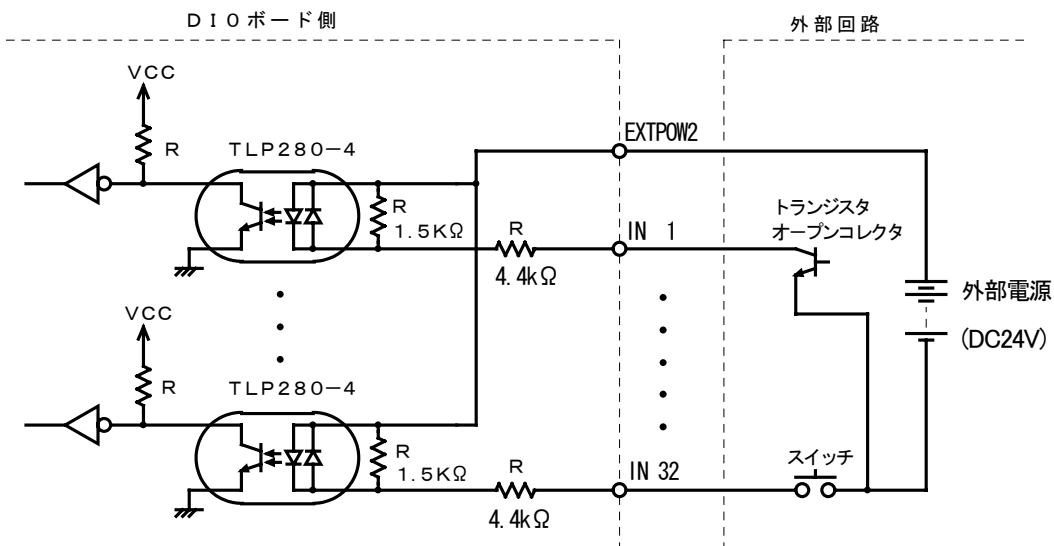


図2. 8-2 シンクタイプとの接続

### 2. 8. 3 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。

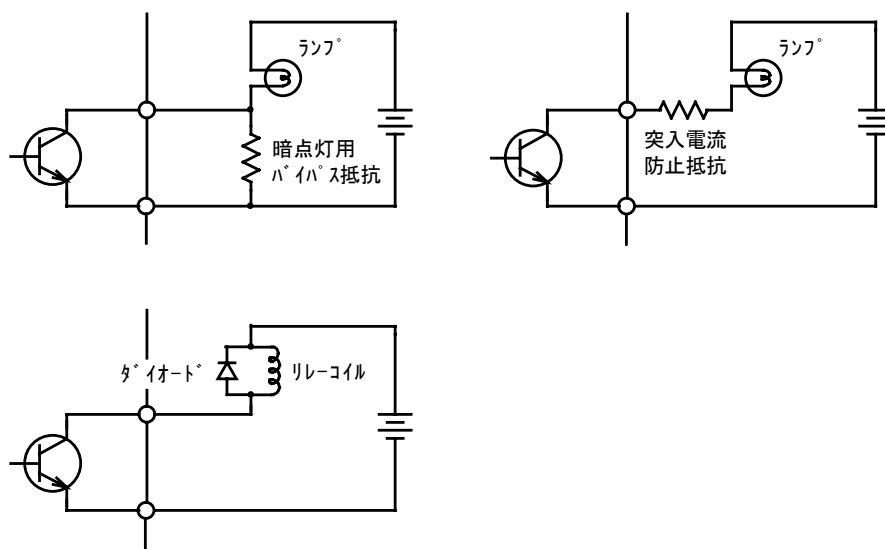


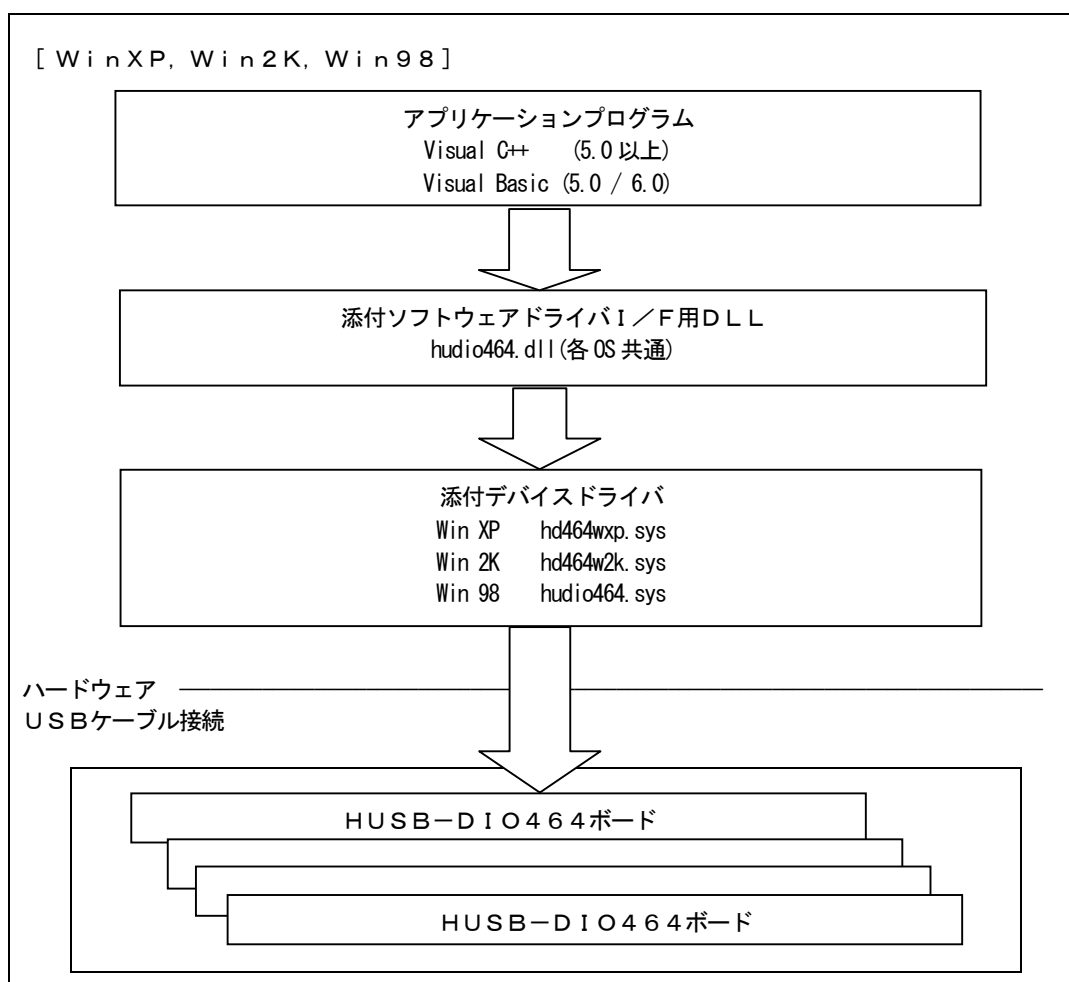
図 2. 8-3 外部接続における対策

### 3. ソフトウェア編

#### 3. 1 対応OS, ドライバ種別

- (1) 添付ソフトウェアの対応OS  
WindowsXP Professional, (以下WinXPとします),  
WindowsXP Home Edition (以下WinXPとします),  
Windows2000 Professional (以下Win2Kとします),  
Windows98 Second Edition (以下Win98とします)  
です.
- (2) Windows版デバイスドライバ の種別は  
WinXP では hd464w xp. sys です.  
Win2K では hd464w2k. sys です.  
Win98 では hudio464. sys です.
- (3) Windows版ドライバI/F用DLL として  
hudio464. dllを上記OS共通で使用します.

#### 3. 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、(A:) ¥Readme. txtを参照して下さい。

### 3. 3 デバイスドライバのインストール

#### (1) WinXPへのインストール

- ① Windows動作中を確認した後、DIO464ボードの電源切状態で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DIO464ボードの電源を投入します。
- ② DIO464ボード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-DI0464 (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。（図3. 3-2を参照）

#### (2) Win2Kへのインストール

- ① Windows動作中を確認した後、DIO464ボードの電源切状態で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DIO464ボードの電源を投入します。
- ② DIO464ボード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 A : ¥WIN2K (Aドライブである場合)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。（図3. 3-3を参照）

#### (3) Win98へのインストール

- ① Windows動作中を確認した後、DIO464ボードの電源切状態で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DIO464ボードの電源を投入します。
- ② DIO464ボード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 A : ¥WIN98 (Aドライブである場合)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。（図3. 3-4を参照）



Windows X P  
インストール結果



Windows2000  
インストール結果



Windows98  
インストール結果

図3. 3-1 デバイスドライバインストール後の確認画面

図3. 3-2 WinXP  
のインストール手順

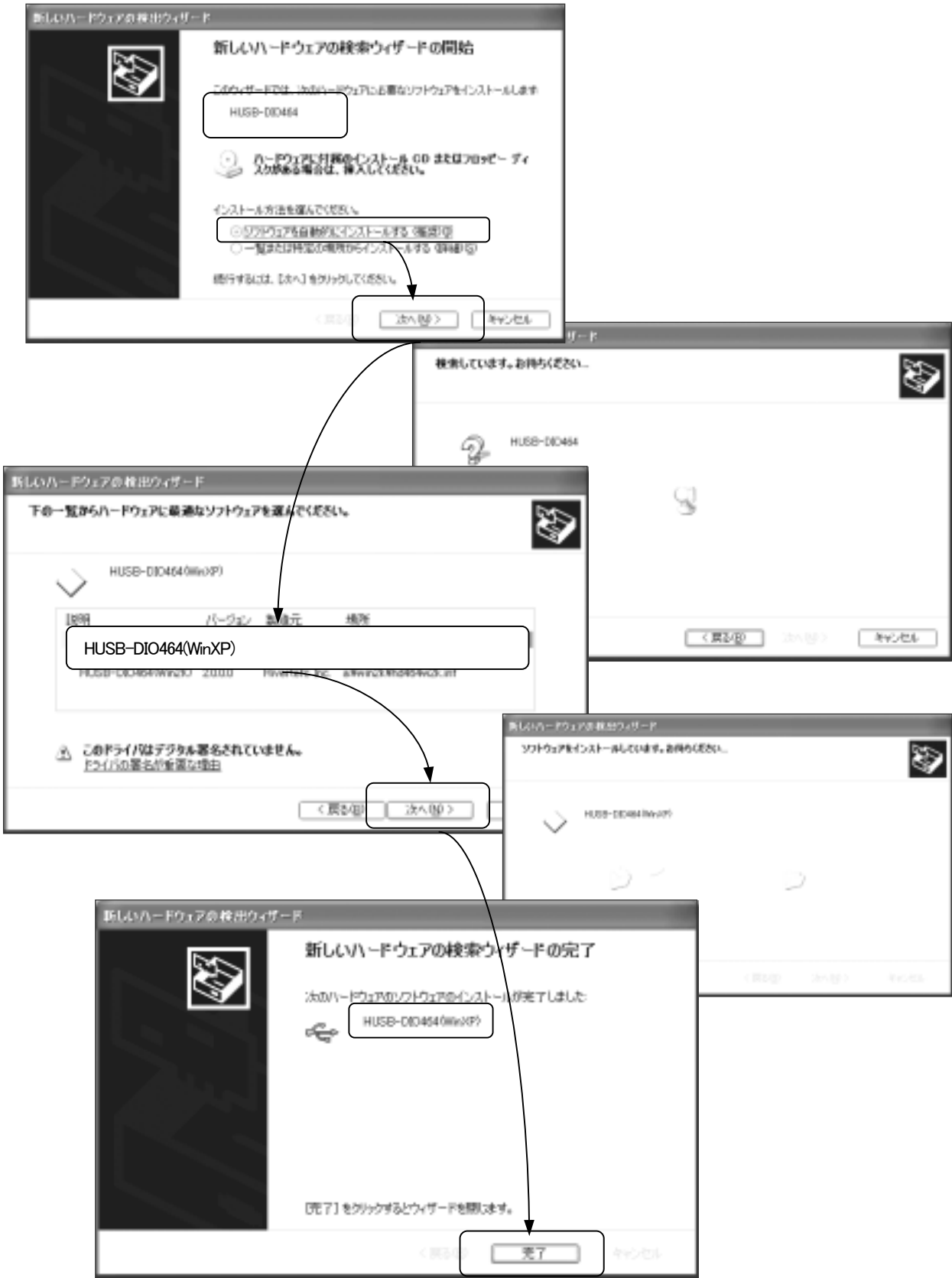


図3. 3-3 Win2K  
のインストール手順

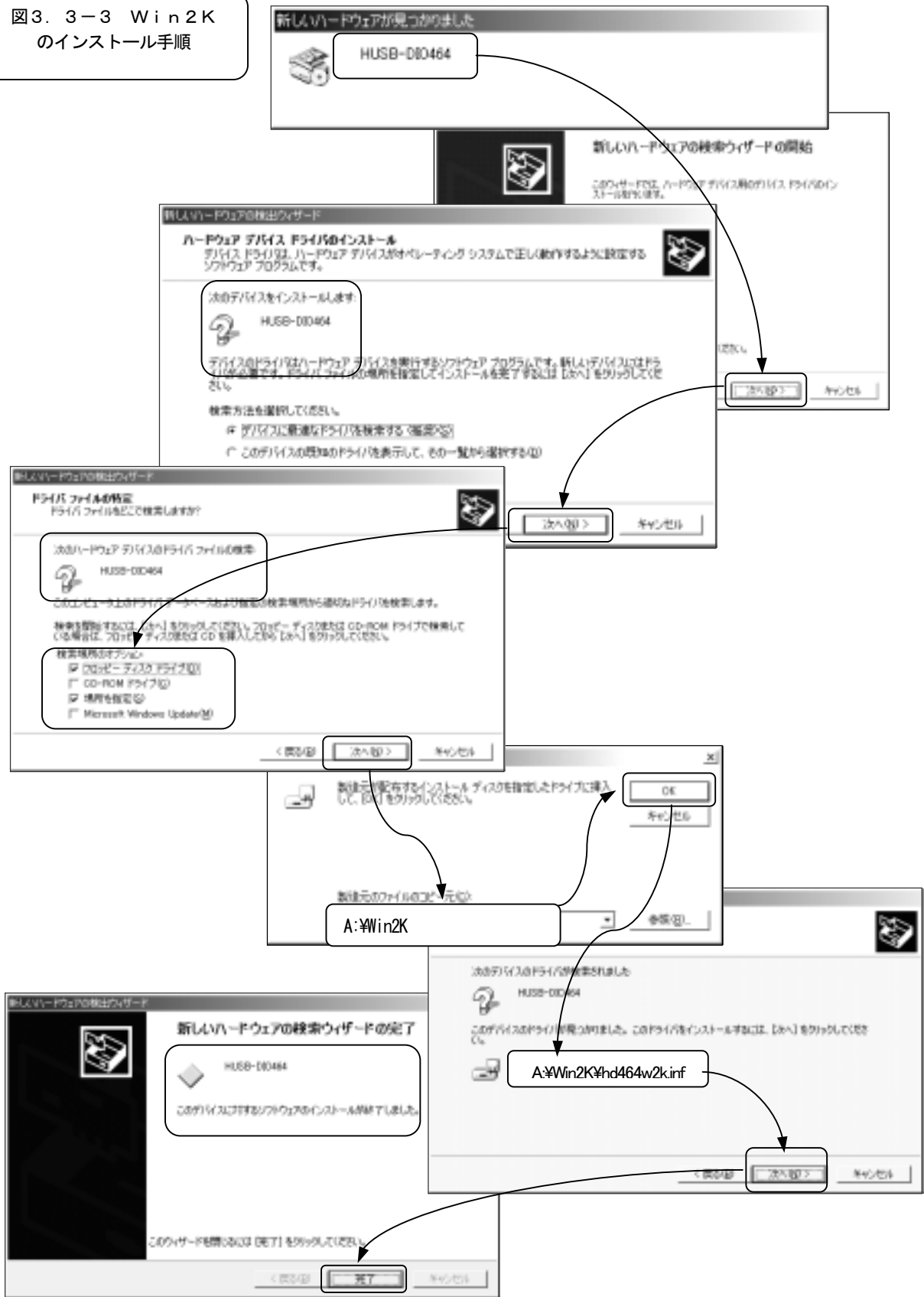
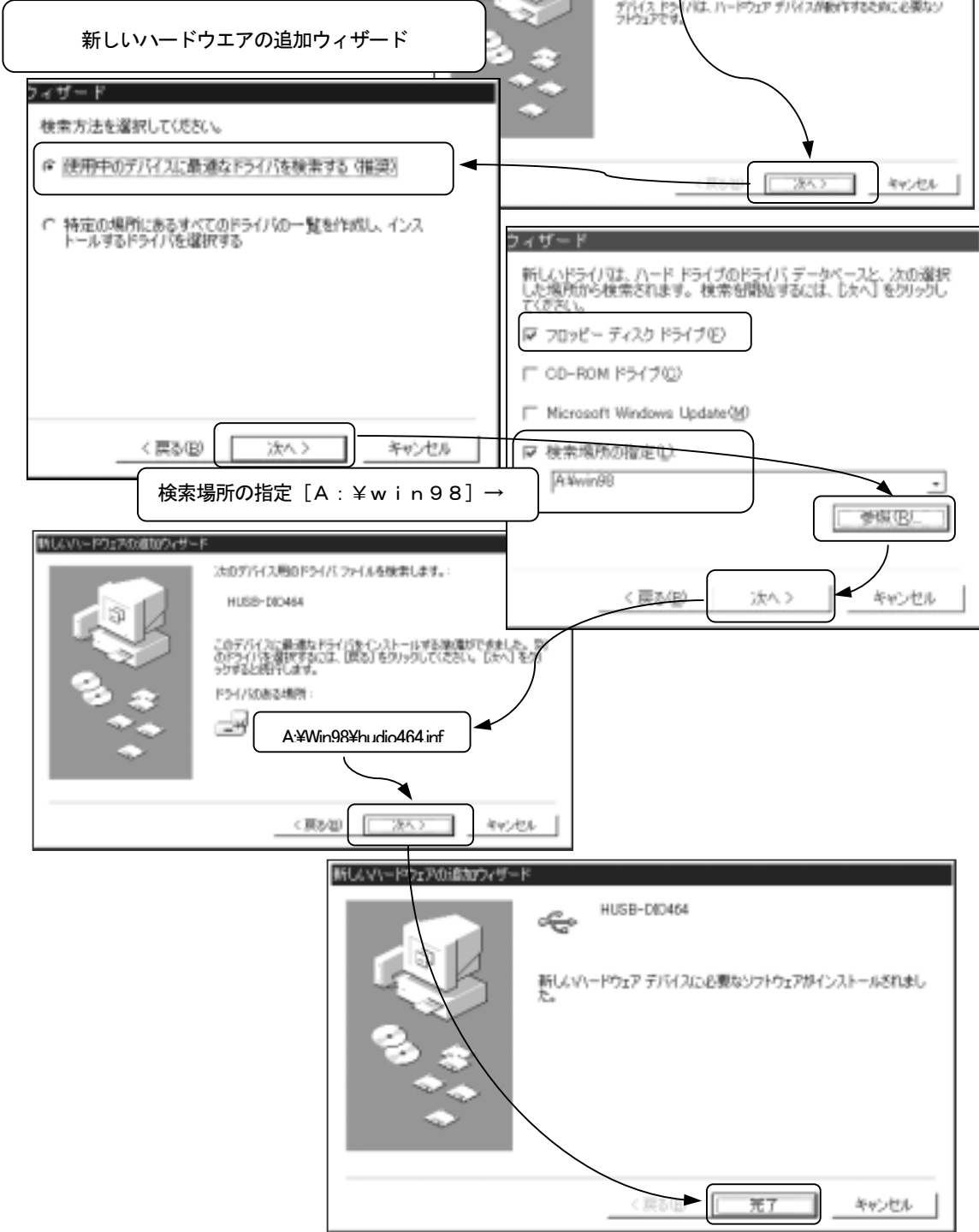


図 3. 3-4 Win98  
のインストール手順



### 3. 4 デバイスドライバの更新

ドライバ、ドライバ/ライブラリのバージョンアップ等でデバイスドライバを更新する場合の方法は次の通りです。

#### 3. 4. 1 ドライバの更新

更新（バージョンアップ）はドライバファイルの差し替えとなります。  
更新を行う場合には、D I Oボードをパソコンに接続した状態で行います。

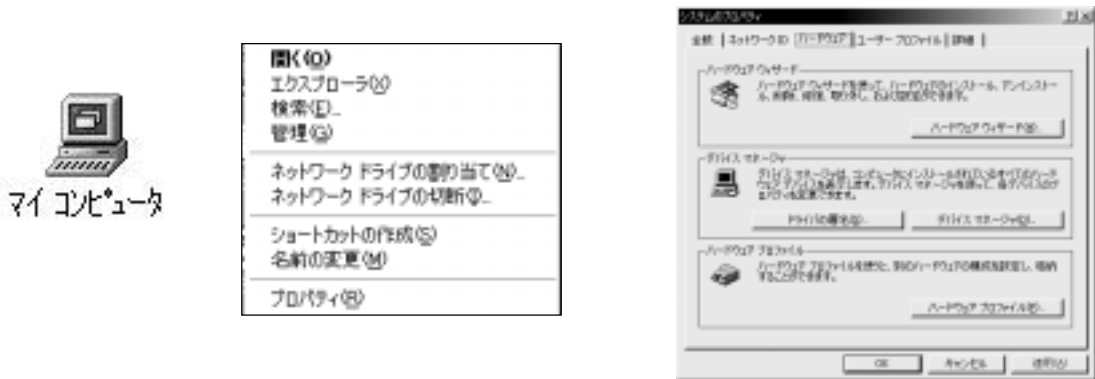
##### (1) Windows X P

スタートメニューの「マイコンピュータ」アイコンを右クリックし、「プロパティ」をクリックします。  
次に「ハードウェア」タブをクリックし、「デバイスマネージャ」をクリックします。  
“Hivertec HUSB-DI0464 Driver” 左隣の“+” 記号をクリックして、一覧を展開します。  
“HUSB-DI0464” を選択し、「ドライバの更新」をクリックします。  
以降の手順は、図 3. 4-1 Win X P ドライバの更新を参照して下さい。



##### (2) Windows 2000

デスクトップの「マイコンピュータ」アイコンを右クリックし、「プロパティ」をクリックします。  
次に「ハードウェア」タブをクリックし、「デバイスマネージャ」をクリックします。  
“Hivertec HUSB-DI0464 Driver” 左隣の“+” 記号をクリックして、一覧を展開します。  
“HUSB-DI0464” を選択し、「プロパティ」をクリックします。  
以降の手順は、図 3. 4-2 Win 2 K ドライバの更新を参照して下さい。

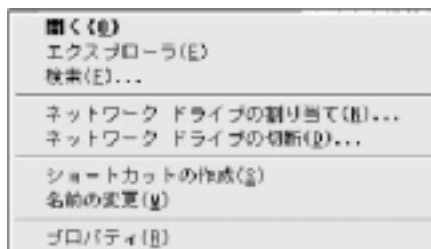


(3) Windows 98

デスクトップの「マイコンピュータ」アイコンを右クリックし、「プロパティ」をクリックします。次に「デバイスマネージャ」タブをクリックし、「Hivertec USB Driver」左隣の「+」記号をクリックして、一覧を展開します。

「HUSB-D10464」を選択し、「プロパティ」をクリックします。

以降の手順は、図3.4-3 Win98ドライバの更新を参照して下さい。



3. 4. 2 ドライバ I/Fライブラリの更新

Windowsのシステムディレクトリ (WinXPならばC:\¥Windows¥system32, Win2KならばC:\¥Winnt¥system32, Win98ならばC:\¥Windows¥system (システムドライブがCドライブの場合)) にライブラリファイルを格納して使用している場合は上記のドライバの更新で更新されます。

もし、作業フォルダにライブラリファイルをコピーして使用されている場合は、作業フォルダに格納された同名のライブラリファイルを消去し、バージョンアップされたライブラリファイルを作業フォルダにコピーします。

(上書きコピーでも可能)

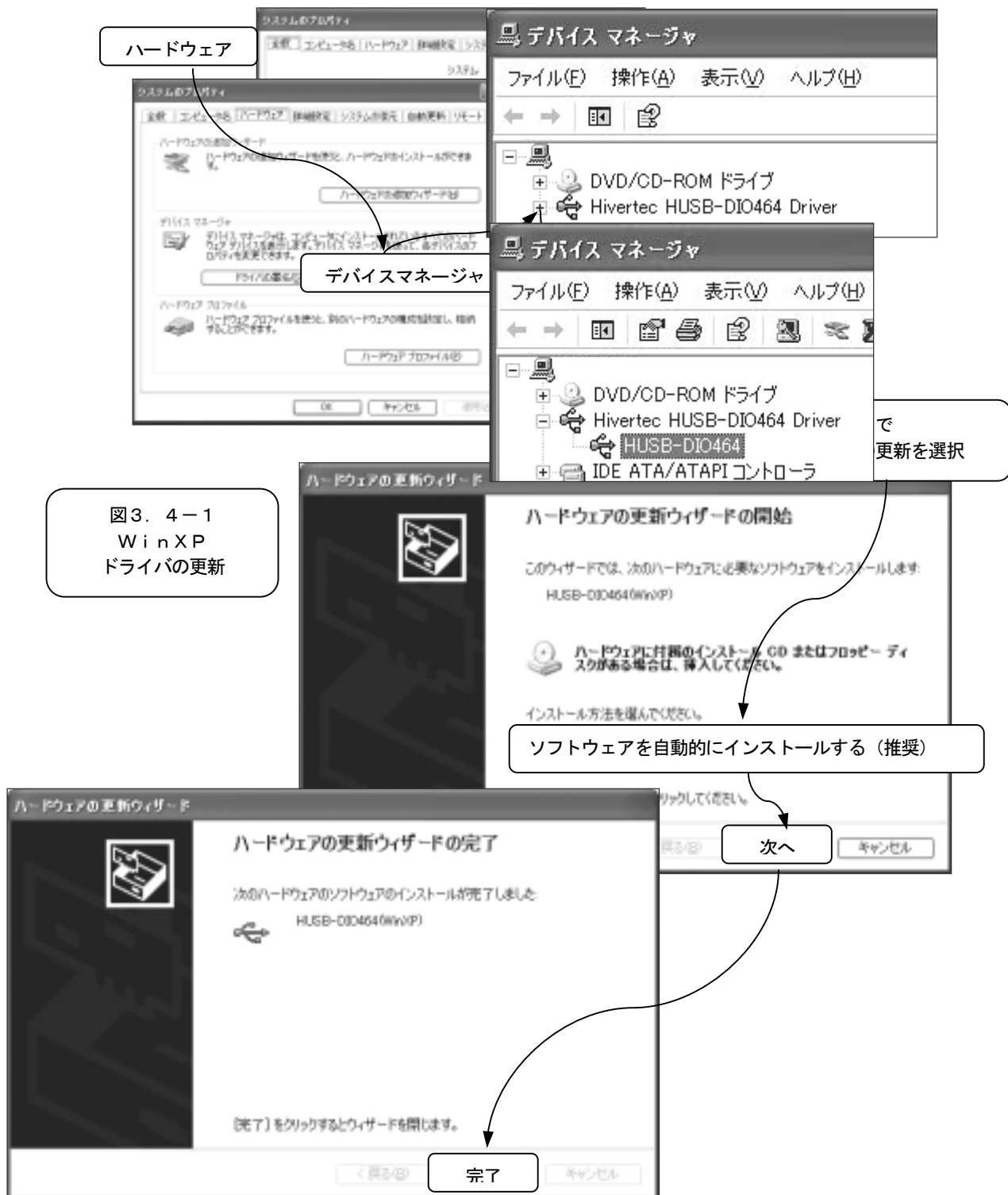


図3. 4-1  
WinXP  
ドライバの更新

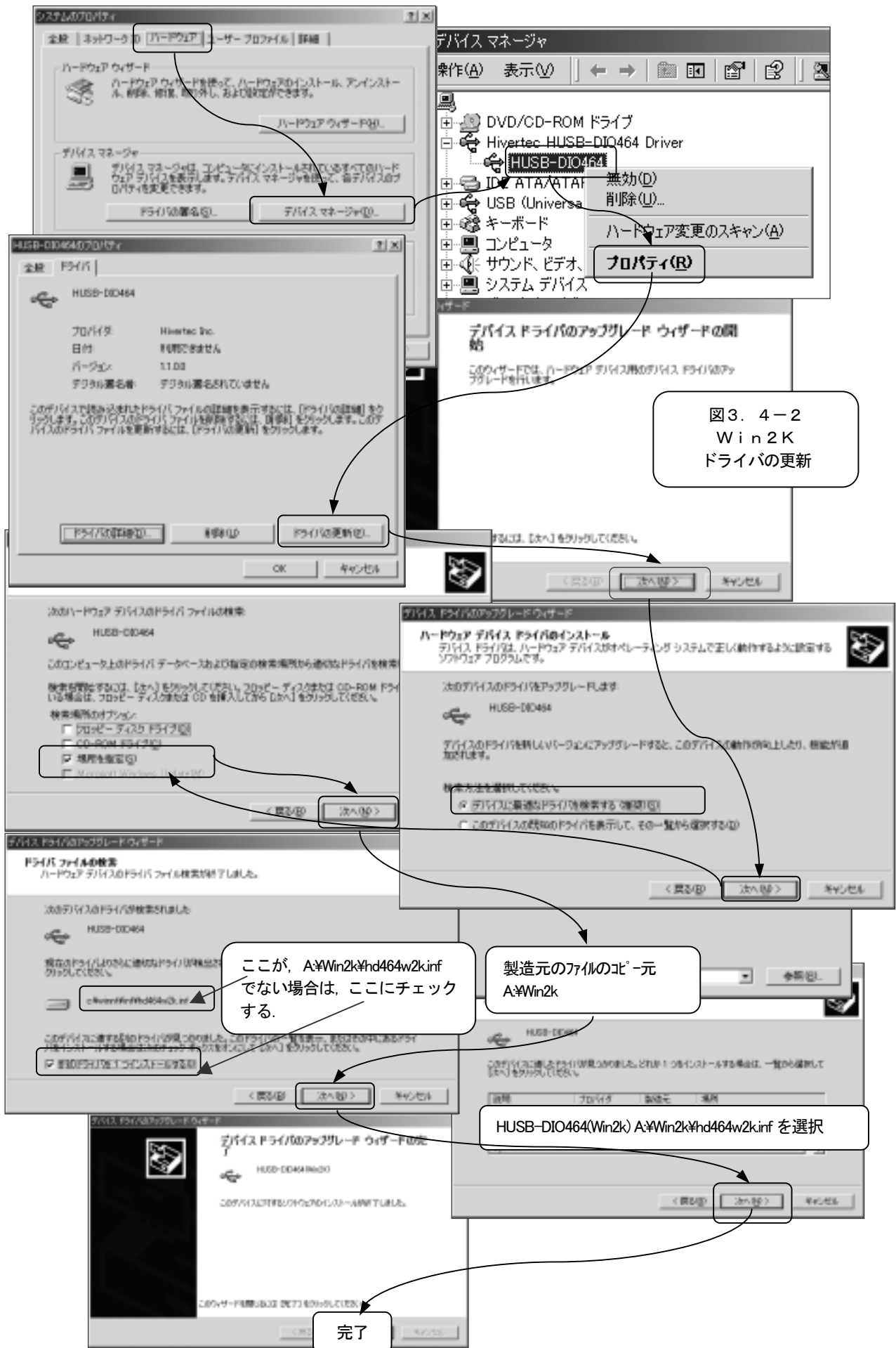
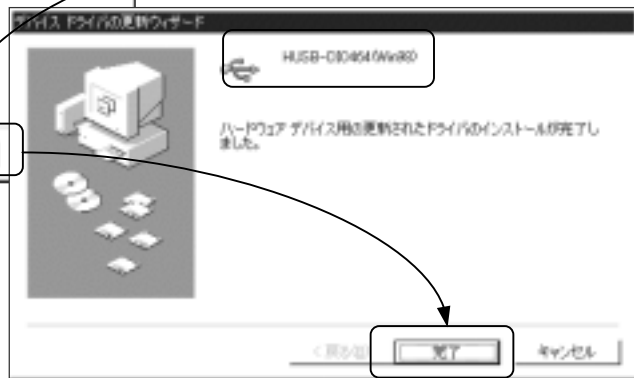
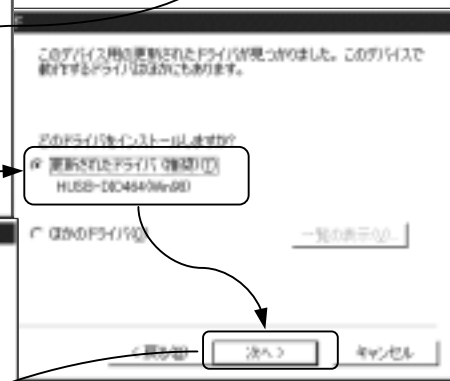
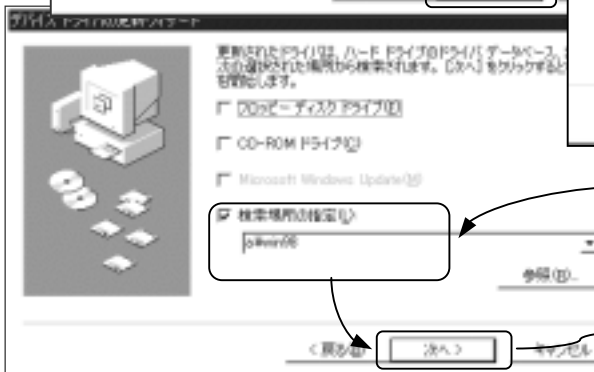
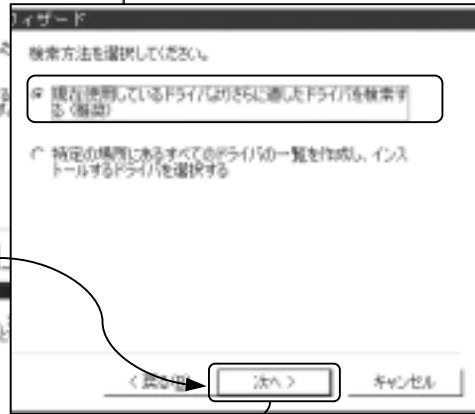
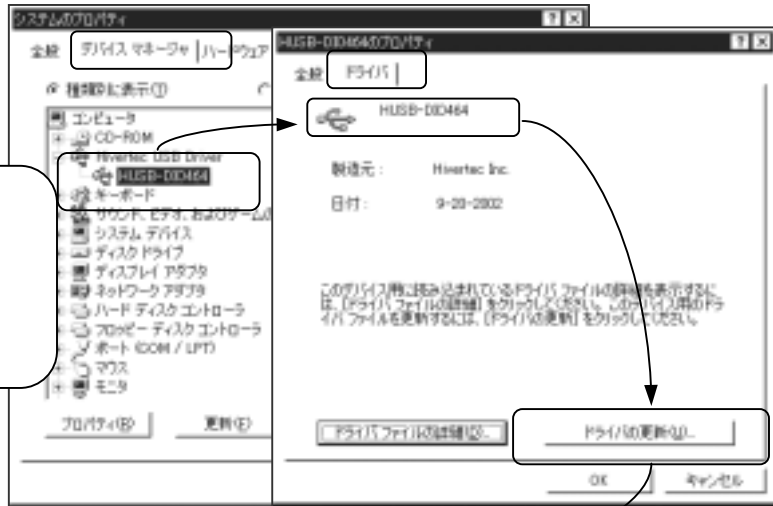


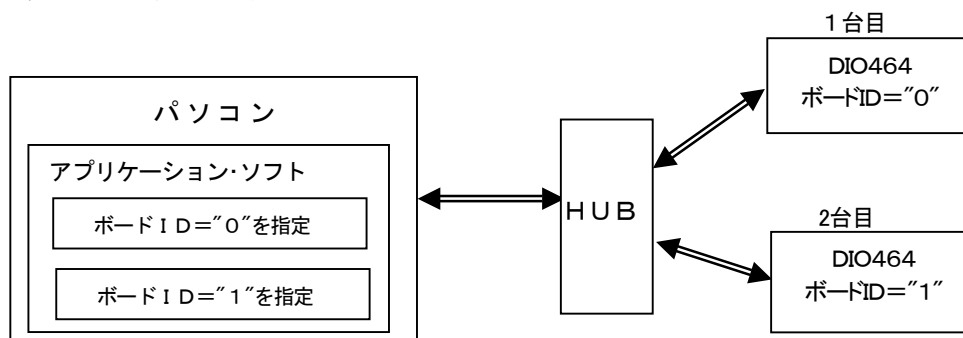
図3. 4-3  
Win98  
ドライバの更新

システムのプロパティ  
→デバイス指定  
→プロパティ  
→ドライバ  
→ドライバの更新



### 3. 5 ボードを複数枚使用する場合

DIO464ボード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コントローラの指定

USBケーブルで接続された2台以上のDIOボードは、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。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DIOボードにそれぞれ固有な「ボードID No.」を設定し識別します。(ボードIDは0~15)

#### (2) デバイスIDの確認方法

添付ソフトウェアフロッピーディスクの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 3. 3 ボード上の設定「図3. 3-2 ボードIDの設定」を参照して下さい。

### 3. 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	DIOボード (デバイス) の認識	アプリケーションプログラム (ドライバI/F用DLL) 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	DIOボード (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) して下さい 切り離し後の再接続は、5秒以上経過してから行います

### 3. 7 ボードアクセス方法

複数のD I O 4 6 4 ボードを制御することができます。

ある1つのD I O 4 6 4 ボードにアクセスするためには、まずこのデバイスをオープンして、アクセスするための足がかりとなるデバイスハンドル値を取得する必要があります。

デバイスをオープンするためには、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となります。

#### 3. 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。

[ C言語 : Visual C/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  (0x81, 0x02, 0x00, 0x00)
    WORD        EPsiz[8];       // エンドポイント・バイト数 (32, 32, 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  (0x81, 0x02, 0x00, 0x00)
    EPsiz      As spc_mat      ' エンドポイント・バイト数 (32, 32, 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
```

### 3. 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. 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。

### 3. 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
	dio464_rOutB ()	出力ポートからの1バイトデータ読込
	dio464_wOutB ()	出力ポートへの1バイトデータ書込
	dio464_rInW ()	入力ポートからの2バイトデータ読込
	dio464_rOutW ()	出力ポートからの2バイトデータ読込
	dio464_wOutW ()	出力ポートへの2バイトデータ書込
6	dio464_rInDW ()	入力ポートからの4バイトデータ読込
	dio464_rOutDW ()	出力ポートからの4バイトデータ読込
	dio464_wOutDW ()	出力ポートへの4バイトデータ書込
7	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
8	dio464_SetFilter ()	入力ポートのフィルターの設定

表 3. 7-1 関数一覧

### 3. 7. 5 USBボードとの通信時間

HUSB-DIOボードは[USB 1.1 規格 : バルク通信]より、ボードとの通信に多少の時間を要します。ドライバ関数の概略所要時間を次の表に示します。(個々のパソコンで変化します。)

No	関数名称	通信時間 (msec)
1	dio464_GetDeviceCount ()	—
2	dio464_GetDeviceInfo ()	—
3	dio464_OpenDevice ()	—
4	dio464_CloseDevice ()	—
5	dio464_rInB ()	4
	dio464_rOutB ()	4
	dio464_wOutB ()	2
	dio464_rInW ()	4
	dio464_rOutW ()	4
	dio464_wOutW ()	2
6	dio464_rInDW ()	4
	dio464_rOutDW ()	4
	dio464_wOutDW ()	2
7	dio464_InOutDW ()	4
8	dio464_SetFilter ()	2

表 3. 7-2 関数通信時間

(注) 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“1msec”減じた値です。

### 3. 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D I O 4 6 4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D I Oボード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D I Oボードがオープンされ、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D I O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。

- ◆ D I Oボードの持つ一部の機能が認識できない。  
このケースでは、デバイス情報の取得はできますが、D I Oボードとしての正常な動作は保証しかねます。

### 3. 8 ドライバI/F用DLL関数詳細

No	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D I Oボードの枚数を取得します。		
VC++	書式	DWORD WINAPI dio464_GetDeviceCount (DWORD* count );	
	引数	DWORD* count ・ ・取得したD I Oボード枚数の格納エリア	
	呼出例	<pre> DWORD* count; //D I Oボードの枚数 DWORD ret; //関数の戻り値 ret = dio464_GetDeviceCount( &amp;count ); </pre>	
VB	書式	Declare Function dio464_GetDeviceCount Lib "hudio464.dll" (ByRef count As Long) As Long	
	引数	ByRef count As Long ・ ・取得したD I Oボード枚数の格納エリア	
	呼出例	<pre> Dim count As Long 'D I Oボードの枚数 Dim ret As Long '関数の戻り値 ret = dio464_GetDeviceCount( count ) </pre>	
備考			

No	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D I Oボード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。		
VC++	書式	DWORD WINAPI dio464_GetDeviceInfo (DWORD* count, HUSBDEVINF* HusbInf);	
	引数	DWORD* count ・ ・ D I Oボードの枚数 (デバイス情報取得ボード枚数) HUSBDEVINF* HusbInf ・ ・ 取得するデバイス情報がセットされるエリア先頭アドレス	
	呼出例	<pre> DWORD ret; //関数の戻り値 DWORD count = 2; //最大枚数は2 HUSBDEVINF HusbInf [2]; //2枚のD I Oボードのデバイス情報がセットされるべきエリア ret = dio464_GetDeviceInfo( &amp;count, &amp;HusbInf [0] ); </pre>	
VB	書式	Declare Function dio464_GetDeviceInfo Lib "hudio464.dll" (ByRef count As Long, HusbInf As HUSBDEVINF) As Long	
	引数	ByRef count As Long ・ ・ D I Oボードの枚数 (デバイス情報取得ボード枚数) HusbInf As HUSBDEVINF ・ ・ 取得するデバイス情報がセットされるエリア先頭アドレス	
	呼出例	<pre> Dim ret As Long '関数の戻り値 Dim count As Long '枚数 Dim HusbInf (2) As HUSBDEVINF 'デバイス情報のエリア count = 2 '最大枚数は2枚 ret = dio464_GetDeviceInfo( count, HusbInf (0) ) </pre>	
備考			

No	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D I Oボードをオープンし、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D I Oボード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。		
VC++	書式	DWORD WINAPI dio464_OpenDevice (DWORD * hDev, HUSBDEVINF* HusbInf);	
	引数	DWORD hDev; .. 取得するデバイスハンドルの格納エリア HUSBDEVINF* HusbInf; .. オープンするボードのデバイス情報格納アドレス	
	呼出例	<pre> DWORD ret; //関数の戻り値 DWORD hDev[2]; //デバイスハンドル取得エリア ret = dio464_OpenDevice (&amp;hDev[0], &amp;HusbInf[0]); // 1 番目のデバイス情報 ret = dio464_OpenDevice (&amp;hDev[1], &amp;HusbInf[1]); // 2 番目のデバイス情報 </pre>	
VB	書式	Declare Function dio464_OpenDevice Lib "hudio464.dll" (ByRef hDev As Long, HusbInf As HUSBDEVINF) As Long	
	引数	ByRef hDev As Long .. 取得するデバイスハンドルの格納エリア HusbInf As HUSBDEVINF .. オープンするボードのデバイス情報格納アドレス	
	呼出例	<pre> Dim ret As Long ' 関数の戻り値 Dim hDev(2) As Long ' デバイスハンドル取得エリア ret = dio464_OpenDevice (hDev(0), HusbInf(0)) ' 1 番目のデバイス情報 ret = dio464_OpenDevice (hDev(1), HusbInf(1)) ' 2 番目のデバイス情報 </pre>	
備考			

No	4	dio464_CloseDevice ()	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D I Oボードをクローズします。 以降このデバイスハンドルは、無効となり、このD I Oボード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。		
VC++	書式	DWORD WINAPI dio464_CloseDevice (DWORD hDev);	
	引数	DWORD hDev .. クローズするボードのデバイスハンドル	
	呼出例	<pre> DWORD ret; //関数の戻り値 ret = dio464_CloseDevice ( hDev ); </pre>	
VB	書式	Declare Function dio464_CloseDevice Lib "hudio464.dll" (ByVal hDev As Long) As Long	
	引数	ByVal hDev As Long .. クローズするボードのデバイスハンドル	
	呼出例	<pre> Dim ret As Long ' 関数の戻り値 ret = dio464_CloseDevice ( hDev ) </pre>	
備考			

No	5	dio464_rInB()	入力ポートからの1バイトデータ読込
		dio464_rOutB()	出力ポートからの1バイトデータ読込
		dio464_wOutB()	出力ポートへの1バイトデータ書込
		dio464_rInW()	入力ポートからの2バイトデータ読込
		dio464_rOutW()	出力ポートからの2バイトデータ読込
		dio464_wOutW()	出力ポートへの2バイトデータ書込
機能		<p>デバイスハンドルで指定されたD I Oボード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バイトデータを読み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バイトデータを読み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みます。  入力ポートからの2バイトデータ読込・・・2バイトデータを読み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バイトデータを読み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みます。  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。</p>	
VC++	書式	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 );	
	引数	DWORD hDev・・・対象デバイスのデバイスハンドル WORD 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) WORD* 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 WORD wwdt・・・書込む2バイトデータ	
	呼出例	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	
VB	書式	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	
	引数	ByVal hDev As Long・・・対象デバイスのデバイスハンドル ByVal port As Integer・・・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) ByRef wrdt As Integer・・・読込んだデータを格納する2バイトエリアのアドレス ByVal wwdt As Integer・・・書込む2バイトデータ	
	呼出例	Dim ret As Long '関数の戻り値 Dim dt As Integer '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
-----	---

No	6	dio464_rInDW ()	入力ポートからの4バイトデータ読込
		dio464_rOutDW ()	出力ポートからの4バイトデータ読込
		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D I Oボード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みます。 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。		
VC++	書 式	DWORD WINAPI dio464_rInDW ( DWORD hDev, DWORD* drdt ); DWORD WINAPI dio464_rOutDW ( DWORD hDev, DWORD* drdt ); DWORD WINAPI dio464_wOutDW ( DWORD hDev, DWORD dwdt );	
	引 数	DWORD hDev・・・対象デバイスのデバイスハンドル DWORD* 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 DWORD dwdt・・・書込む4バイトデータ	
	呼出例	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	
VB	書 式	Declare Function dio464_rInDW Lib "hudio464.dll" (ByVal hDev As Long, ByVal drdt As Long) As Long Declare Function dio464_rOutDW Lib "hudio464.dll" (ByVal hDev As Long, ByVal drdt As Long) As Long Declare Function dio464_wOutDW Lib "hudio464.dll" (ByVal hDev As Long, ByVal dwdt As Long) As Long	
	引 数	ByVal hDev As Long・・・対象デバイスのデバイスハンドル ByVal drdt As Long・・・読込んだデータを格納する4バイトエリアのアドレス ByVal dwdt As Long・・・書込む4バイトデータ	
	呼出例	Dim ret As Long '関数の戻り値 Dim dt As Long 'データ格納エリア ret = dio464_rInDW ( hDev, dt ); '入力ポート読込	
備 考			

No	7	<b>dio464_InOutDW()</b>	D I O の出力ポートへ4バイトデータ書込と D I O 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D I O の出力ポートへ4バイトデータを書込みます。 また出力直前の入力ポートから4バイトデータを読み込み、指定したエリアに格納します。		
VC++	書式	DWORD WINAPI dio464_InOutDW ( DWORD hDev, DWORD* drdt, DWORD dwdt );	
	引数	DWORD hDev ... 対象デバイスのデバイスハンドル DWORD* 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 DWORD dwdt ... 書込む4バイトデータ	
	呼出例	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	
VB	書式	Declare Function dio464_InOut Lib "hudio464.dll" (ByVal hDev As Long, _ ByRef drdt As Long _ ByVal dwdt As Long) As Long	
	引数	ByVal hDev As Long ... 対象デバイスのデバイスハンドル ByRef drdt As Long ... 読込んだデータを格納する4バイトエリアのアドレス ByVal dwdt As Long ... 書込む4バイトデータ	
	呼出例	Dim ret As Long '関数の戻り値 Dim dt As Long 'データ格納エリア ret = dio464_InOutDW( hDev, dt, &H55555555 ) '55555555h のデータ出力と入力ポート読込	
備考			

No	8	<b>dio464_SetFilter()</b>	入力ポートのフィルターの設定
機能	デバイスハンドルで指定されたD I O の指定された入力ポートのフィルターを設定します		
VC++	書式	DWORD WINAPI dio464_SetFilter (DWORD hDev, WORD port, DWORD wtm);	
	引数	DWORD hDev ... 対象デバイスのデバイスハンドル WORD 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) DWORD wtm ... フィルター時間 ( wtm × 0.1msec )	
	呼出例	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	
VB	書式	Declare Function dio464_SetFilter Lib "hudio464.dll" (ByVal hDev As Long, ByVal port As Integer _ ByVal wtm As Integer) As Long	
	引数	ByVal hDev As Long ... 対象デバイスのデバイスハンドル ByVal port As Integer ... ポート指定 (&H1:ポート1, &H2:ポート2, &H4:ポート3, &H8:ポート4) ByVal wtm As Integer ... サンプリング時間( wtm × 0.1msec )	
	呼出例	Dim ret As Long '関数の戻り値 ret = dio464_SetFilter( hDev, &H1, 100 ) 'ポート1に10msecのフィルター設定	
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。		

### 3. 9 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++		VB
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	既にオープン済み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）

### 3. 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++ (5.0)・・・Cコーディング 【 sud46400.exe 】
- ② Visual Basic(5.0) 【 sud46402.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 (sud46400.exe, sud46402.exe) 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 ▶

- ◆Visual C++ サンプルは開発ツールとして Visual C++ 5.0 以上がインストールされている必要があります。
- ◆Visual Basic サンプルは開発ツールとして Visual Basic 5.0 がインストールされている必要があります。
- ◆D I O 4 6 4 ボードを2枚以上で使用する場合、ボード I D は重複しないようにして下さい。
- ◆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。

#### 【 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。



※ D I O 4 6 4 ボード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。

### 3. 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
- ⑤サンプルプログラムの終了 (デバイスクローズ後)

(注) ④デバイスクローズ処理において、全出力ポート (1~4) には'0'を書込みます。

#### (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。

The screenshot shows a software window titled "[HWT] HUSB-DID464 Sample". It contains several controls and data fields:

- Buttons: "デバイス情報取得", "デバイスオープン", "デバイスクローズ".
- Board ID: A dropdown menu showing "0".
- Device Information:
  - Board ID: 0
  - EndPoint: 2
  - EP spec: 0081 0002
  - EP size: 64 64
  - EPotion: 0000 0000
  - EPhandle: 0074 0068
- IN PORT: A grid of 32 checkboxes (ports 32 to 1) with port 17 highlighted in green.
- OUT PORT: A grid of 32 checkboxes (ports 32 to 1) with port 1 highlighted in green.

Annotations on the right side of the image:

- Input indicator (read): Input status and display color. Input in progress: Green. Input not in progress: White.
- Output button (write): ON/OFF toggle.
- Output indicator (read): Output status and display color. Output in progress: Green. Output not in progress: None (background color).