

# REX-PCI64

*RS232C Serial Communication Board*

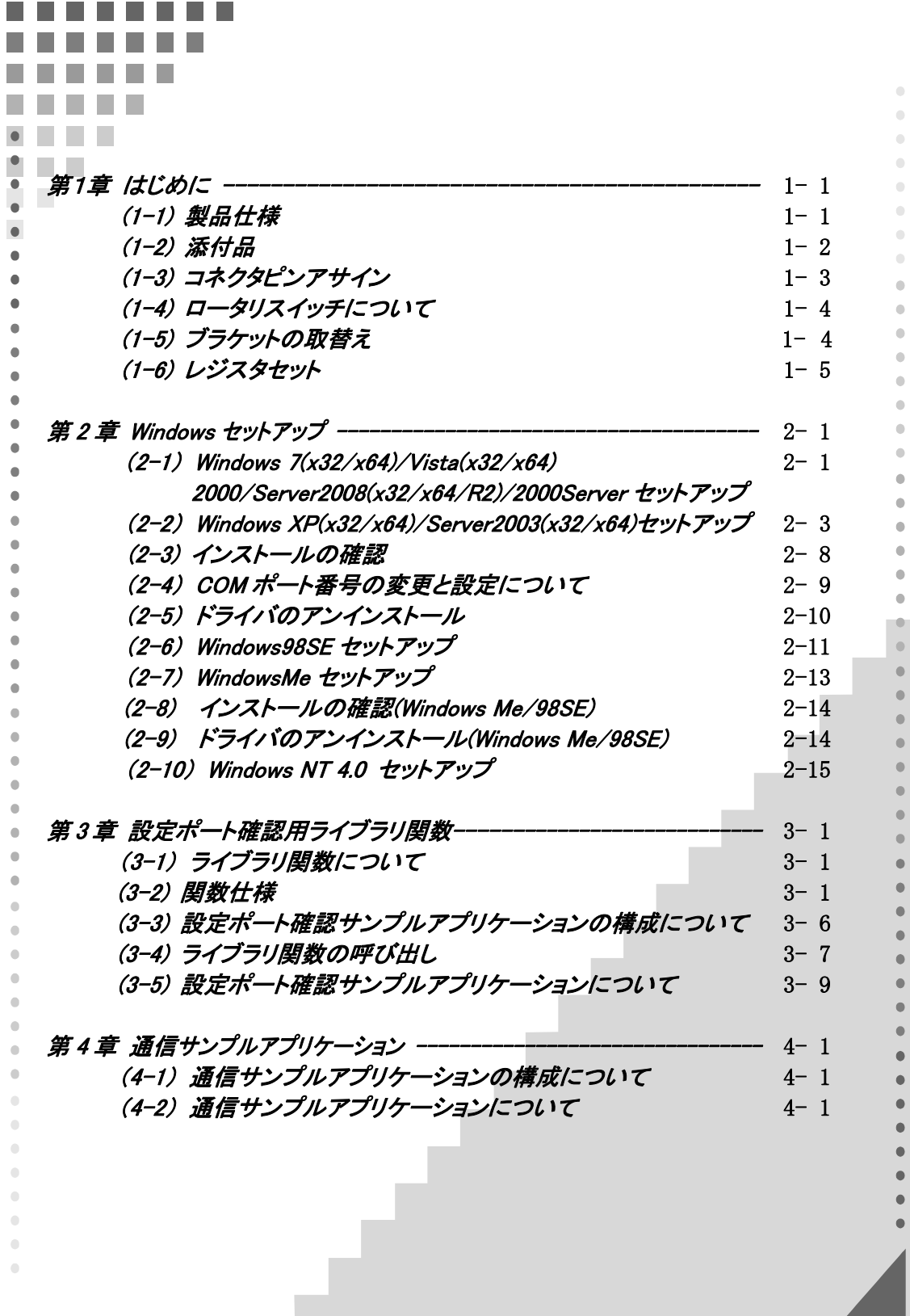
## ユーザーズマニュアル

2010年6月

第4.0版



ラトックシステム株式会社



<b>第1章 はじめに</b> -----	1- 1
(1-1) 製品仕様	1- 1
(1-2) 添付品	1- 2
(1-3) コネクタピンアサイン	1- 3
(1-4) ロータリスイッチについて	1- 4
(1-5) ブラケットの取替え	1- 4
(1-6) レジスタセット	1- 5
<b>第2章 Windows セットアップ</b> -----	2- 1
(2-1) Windows 7(x32/x64)/Vista(x32/x64) 2000/Server2008(x32/x64/R2)/2000Server セットアップ	2- 1
(2-2) Windows XP(x32/x64)/Server2003(x32/x64)セットアップ	2- 3
(2-3) インストールの確認	2- 8
(2-4) COM ポート番号の変更と設定について	2- 9
(2-5) ドライバのアンインストール	2-10
(2-6) Windows98SE セットアップ	2-11
(2-7) WindowsMe セットアップ	2-13
(2-8) インストールの確認(Windows Me/98SE)	2-14
(2-9) ドライバのアンインストール(Windows Me/98SE)	2-14
(2-10) Windows NT 4.0 セットアップ	2-15
<b>第3章 設定ポート確認用ライブラリ関数</b> -----	3- 1
(3-1) ライブラリ関数について	3- 1
(3-2) 関数仕様	3- 1
(3-3) 設定ポート確認サンプルアプリケーションの構成について	3- 6
(3-4) ライブラリ関数の呼び出し	3- 7
(3-5) 設定ポート確認サンプルアプリケーションについて	3- 9
<b>第4章 通信サンプルアプリケーション</b> -----	4- 1
(4-1) 通信サンプルアプリケーションの構成について	4- 1
(4-2) 通信サンプルアプリケーションについて	4- 1

## 安全にご使用いただくために

本製品は安全に充分配慮して設計を行っていますが、誤った使い方をすると火災や感電などの事故につながり大変危険です。ご使用の際は、警告/注意事項を必ず守ってください。

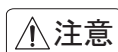
### 表示について

この取扱説明書は、次のような表示をしています。表示の内容をよく理解してから本文をお読みください。



**警告**

この表示を無視して誤った取扱いをすると、火災や感電などにより、人が死亡または重傷を負う可能性があ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、感電やその他の事故により、人が負傷または物的損害が発生する可能性がある内容を示しています。



**警告**

- 製品の分解や改造などは、絶対に行わないでください。
- 無理に曲げる、落とす、傷つける、上に重い物を載せることは行わないでください。
- 製品が水・薬品・油などの液体によって濡れた場合、ショートによる火災や感電の恐れがあるため使用しないでください。



**注意**

- 本製品は電子機器ですので、静電気を与えないでください。
- ラジオやテレビ、オーディオ機器の近く、モーターなどのノイズが発生する機器の近くでは誤動作することがあります。必ず離してご使用ください。
- 高温多湿の場所、温度差の激しい場所、チリやほこりの多い場所、振動や衝撃の加わる場所、スピーカなどの磁気を帯びた物の近くで保管しないでください。
- 煙が出たり異臭がする場合は、直ちにパソコンや周辺機器の電源を切り、電源ケーブルもコンセントから抜いてください。
- 本製品は、医療機器、原子力機器、航空宇宙機器、輸送機器など人命に関わる設備や機器、及び高度な信頼性を必要とする設備や機器での使用は意図されておりません。これらの設備、機器制御システムに本製品を使用し、本製品の故障により人身事故/火災事故/その他の障害が発生した場合、いかなる責任も負いかねます。
- 取り付け時、鋭い部分で手を切らないように、十分注意して作業を行ってください。
- 配線を誤ったことによる損失、逸失利益などが発生した場合でも、いかなる責任も負いかねます。

### その他のご注意

- 本書の内容に関して、将来予告なしに変更することがあります。
- 本書の内容につきましては万全を期して作成しておりますが、万一不審な点や誤りなどお気づきになりましたらご連絡お願い申し上げます。
- 本製品の運用を理由とする損失、逸失利益などの請求につきましては、いかなる責任も負いかねますので、予めご了承ください。
- 製品改良のため、将来予告なく外観または仕様の一部を変更する場合があります。
- 本製品は日本国内仕様となっており、海外での保守及びサポートは行っておりません。
- 本製品を廃棄するときは地方自治体の条例に従ってください。条例の内容については各地方自治体にお問い合わせください。
- 本製品の保証や修理に関しましては、添付の保証書に内容を明記しております。必ず内容をご確認の上、大切に保管してください。
- “REX”は株式会社リコーが商標権を所有しておりますが、弊社はその使用許諾契約により本商標の使用が認められています。
- Windowsは米国マイクロソフト社の米国およびその他の国における登録商標です。その他本書に記載されている商品名/社名などは、各社の商標または登録商標です。なお本書では、<sup>TM</sup>、<sup>®</sup>マークは明記しておりません。

### **【電波障害自主規制について】**

この装置は、情報処理装置等電波障害自主規制協議会(VCCI)の基準に基づくクラスB情報技術装置です。この装置は、家庭環境で使用することを目的としていますが、この装置がラジオやテレビジョン受信機に接近して使用されると、受信障害を引き起こすことがあります。

取扱説明書に従って正しい取り扱いをしてください。

## 第1章 はじめに

### (1-1) 製品仕様

REX-PCI64 は、シリアルコントローラに 16C550 互換 UART を搭載し、D-Sub9 ピン(オス)コネクタを 4 ポート採用した RS232C PCI ボードです。

### ハードウェア仕様

項目	仕様内容
バスインターフェース	PCI Local Bus Rev2. 3(32 ビットバスマスタ ユニバーサル PCI)
シリアルコントローラ	メモリマップ方式 16C550 互換 UART
接続コネクタ	D-Sub9Pin(オス) × 4 JISX5101 規格 D-Sub9Pin RS-232C コネクタ オス型
入出力レベル	【ドライバ】ハイレベル出力 : +5V (min) / +5.4V (TYP) ローレベル出力 : -5V (min) / -5.4V (TYP)
	【レシーバ】電圧レンジ : -15V ~ +15V
通信方式	非同期通信
通信速度	110/300/600/1200/2400/4800/9600/19200/38400/ 57600/115200/230400/460800/921600 bps ※実際に実行可能な最大通信速度はパソコンの仕様に依存します。
通信パラメータ	ビット長 : 7/8                      スタートビット : 1 ストップビット : 1/2              パリティ : 偶数/奇数/なし
ドライバ・レシーバ	MAX3245ECAI+ / ICL3245ECAZ (または相当品)
カード ID	複数の REX-PCI64 を使用した場合、ポートを特定するためのロータリスイッチを基板上に設置。
伝送距離	15m
割り込み番号	PCI 割り込みを自動割り当て
外形寸法	120mm (W) × 65 (H) [mm] (PCI ブラケット含まず)
重量	67g (標準 PCI ブラケットを含む)
電源電圧	+3.3V または 5V (PCI バスより供給)
ESD 保護特性	±15kV
動作環境	温度 : 0~55°C    湿度 : 20~80% (但し、結露なきこと)

D-Sub コネクタの 9 番ピンから電源 (5V) を出力できる製品の受注生産が可能です。詳しくは、サポートセンターまでお問い合わせください。

## ソフトウェア仕様

項 目	仕 様 内 容
通信サンプルプログラム	RS-232C 通信サンプルプログラム (VC++2005, VB2005, VC++6. 0, VB6. 0)
ポート認識用サンプルプログラム	ボード上ロータリスイッチおよびポート読み取り サンプルプログラム (VC++2005, VB2005, VC++6. 0, VB6. 0)
シリアル通信ドライバ	REX-PCI64 用デバイスドライバ

### 本製品の制限事項

本製品の IO アドレスは、OADG ハードウェア仕様で定められた IO ベースアドレスとは別のメモリアドレスにマップされますので、直接 IO ポートアドレスにアクセスしているアプリケーションはご使用になれません。

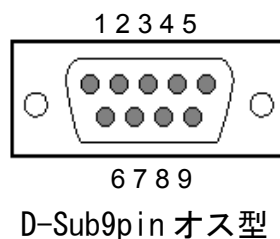
### (1-2) 添付品

ご使用前に下記添付品が添付されているかをご確認願います。

- RS-232C PCI ボード本体(標準 PCI ブラケット付)
- REX-PCI64 専用ケーブル(D-Sub9 ピン×4 ポート)
- Low profile PCI ブラケット
- ソフトウェア CD-ROM
- インストールガイド
- 保証書

### (1-3) コネクタピンアサイン

各信号のコネクタピンアサイン及び機能は下表のようになります。コネクタは OADG 仕様で定められている D-Sub9pin を採用しました。



ピン番	信号名	略称	DTE-外部	説明
1	Data Carrier Detect (DCD)	キャリア検出	←	キャリア検出の通知
2	Receive Data (RD)	受信データ	←	データの受信
3	Transmit Data (TD)	送信データ	⇒	データの送信
4	Data Terminal Ready (DTR)	受信準備	⇒	使用可能であることを通知
5	Signal Ground (SG)	信号用接地	-	グラウンド
6	Data Set Ready (DSR)	送信準備	←	使用可能であることを通知
7	Request to Send (RS)	送信要求	⇒	送信の停止・再開の要求
8	Clear to Send (CS)	送信許可	←	受信の停止・再開の通知
9	Ring Indicate (RI)	被呼表示	←	着信の通知

#### ⚠ D-SUB25PIN への変換について ⚠

製品添付ケーブルは D-SUB9PIN コネクタですので D-SUB25PIN に変換したい場合には、下記の変換表に基づいた変換コネクタをご使用ください。  
 変換コネクタは、一般の量販店やパソコンショップで入手可能です。

D-SUB9PIN			D-SUB25PIN	
ピン番号			ピン番号	
1	←→		8	
2	←→		3	
3	←→		2	
4	←→		20	
5	←→		7	
6	←→		6	
7	←→		4	
8	←→		5	
9	←→		22	

### (1-4) ロータリスイッチについて

本製品では、ロータリスイッチを使用して各ボードに ID 番号を割り振ることが可能です。本機能を利用すると以下の機能が使用できます。

- ・ 複数枚同時装着時に各ケーブル番号(各ケーブルについているシールの番号)に割り当てられている COM ポート番号が簡単に確認できます。
- ・ ソフトウェアから各ボードを識別することが可能。  
(サンプルアプリケーション Page. 3-5 参照。)

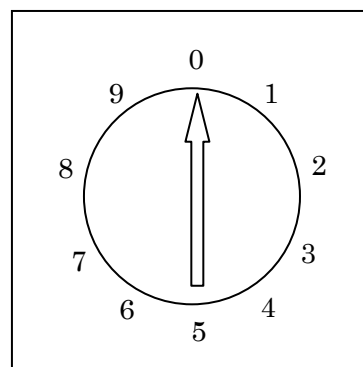
本機能を使用するには、下図のロータリスイッチで、ID 番号が重複しないように設定し、ボードを装着します。

※添付のサンプル (Page. 3-5 参照) を使用する場合は、必須となります。

※重複した場合でも、上記機能が使用できないだけで、COM ポート番号がアサインされていれば、正常にご使用いただけます。



ロータリスイッチ



ロータリスイッチ部拡大図

### (1-5) PCI ブラケットの取替え

ロープロファイル PCI スロットでご使用の場合は、ブラケットの取替えが必要となります。

PCI ボード上の六角ネジを取外してブラケットを取替えます。



## (1-6) レジスタセット

シリアルコントローラは 16C550 互換 UART が搭載されています。  
詳細につきましては、EXAR XR17D154 のデータシートをご参照ください。

TABLE 11: UART CHANNEL CONFIGURATION REGISTERS DESCRIPTION. SHADED BITS ARE ENABLED BY EFR BIT-4.

ADDRESS A3-A0	REG NAME	READ/ WRITE	BIT-7	BIT-6	BIT-5	BIT-4	BIT-3	BIT-2	BIT-1	BIT-0	COMMENT
0 0 0 0	RHR	R	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	LCR[7]=0
0 0 0 0	THR	W	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	LCR[7]=0
0 0 0 0	DLL	R/W	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	LCR[7]=1
0 0 0 1	DLM	R/W	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	LCR[7]=1
0 0 0 1	IER	R/W	0/ CTS/ DSR# Int. Enable	0/ RTS/ DTR# Int. Enable	0/ Xon/Xoff/ Sp. Char. Int. Enable	0	Modem Status Int. Enable	RX Line Status Int. Enable	TX Empty Int. Enable	RX Data Int. Enable	
0 0 1 0	ISR	R	FIFOs Enable	FIFOs Enable	0/ Delta- Flow Cntl	0/ Xoff/special char	INT Source Bit-3	INT Source Bit-2	INT Source Bit-1	INT Source Bit-0	
0 0 1 0	FCR	W	RXFIFO Trigger	RXFIFO Trigger	0/ TXFIFO Trigger	0/ TXFIFO Trigger	DMA Mode	TX FIFO Reset	RX FIFO Reset	FIFOs Enable	
0 0 1 1	LCR	R/W	Divisor Enable	Set TX Break	Set Parity	Even Par- ity	Parity Enable	Stop Bits	Word Length Bit-1	Word Length Bit-0	
0 1 0 0	MCR	R/W	0/ BRG Prescaler	0/ IR Enable	0/ XonAny	Internal Lopback Enable	(OP2) <sup>1</sup>	(OP1) <sup>1</sup> RTS/DTR Flow Sel	RTS# Pin Control	DTR# Pin Control	
0 1 0 1	LSR	R/W	RX FIFO ERROR	TSR Empty	THR Empty	RX Break	RX Fram- ing Error	RX Parity Error	RX Over- run	RX Data Ready	
0 1 1 0	MSR	R	CD	RI	DSR	CTS	Delta CD#	Delta RI#	Delta DSR#	Delta CTS#	
	MSR	W	RS485 DLY-3	RS485 DLY-2	RS485 DLY-1	RS485 DLY-0	Reserved	Reserved	Reserved	Reserved	
0 1 1 1	SPR	R/W	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	User Data
1 0 0 0	FCTR	R/W	TRG Table Bit-1	TRG Table Bit-0	Auto RS485 Enable	Invert IR RX Input	RTS/DTR Hyst Bit-3	RTS/DTR Hyst Bit-2	RTS/DTR Hyst Bit-1	RTS/DTR Hyst Bit-0	
1 0 0 1	EFR	R/W	Auto CTS/DSR Enable	Auto RTS/DTR Enable	Special Char Select	Enable IER [7:5], ISR [5:4], FCR[5:4], MCR[7:5,2], MSR[7:4]	Software Flow Cntl Bit-3	Software Flow Cntl Bit-2	Software Flow Cntl Bit-1	Software Flow Cntl Bit-0	
1 0 1 0	TXCNT	R	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	
1 0 1 0	TXTRG	W	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	
1 0 1 1	RXCNT	R	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	
1 0 1 1	RXTRG	W	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	

## 第2章 Windowsセットアップ

### (2-1) Windows 7 (x32/x64) / Vista (x32/x64) / 2000

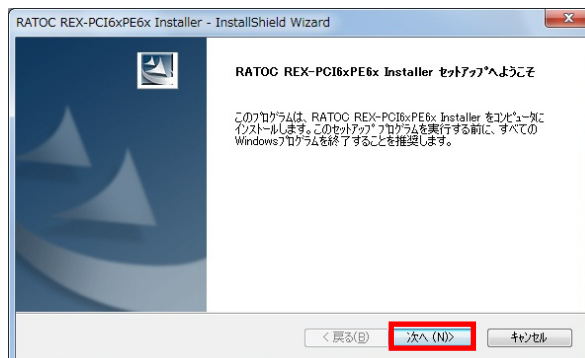
### Server2008 (x32/x64/R2) / 2000Server セットアップ

PCI ボードを接続する前に製品添付 CD-ROM 内のドライバをセットアップします。以下の手順でインストールを行ってください。

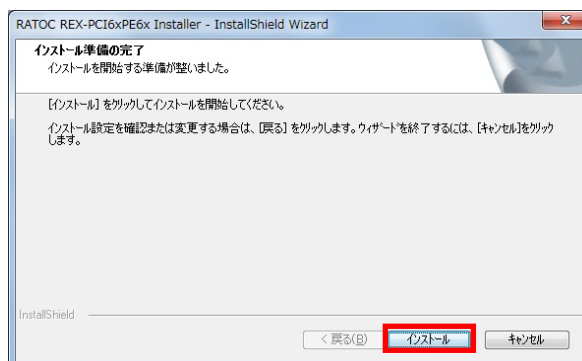
製品添付 CD-ROM 内の  
PCIPE6x\_Setup.exe をダブルクリックして実行します。  
ユーザーアカウント制御の画面が表示された場合は、「はい(Y)」ボタンをクリックします。



「RATOX REX-PCI6xPE6x  
Installer セットアップへようこそ」  
の画面で「次へ(N)」ボタンをクリックします。

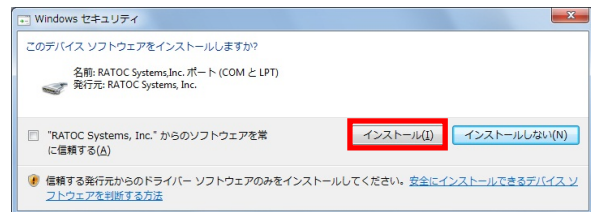
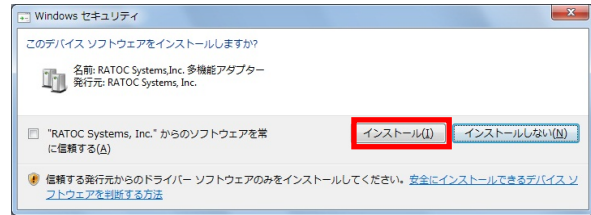


「インストール準備の完了」の画面  
で「インストール」ボタンをクリックします。



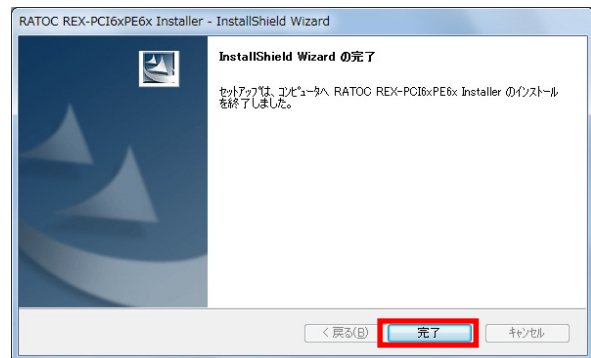
Windows セキュリティの確認画面が3回表示されますので、「インストール」ボタンをクリックします。

(※ Windows 2000/2000Server ではこれらの画面は表示されません。)



以上でドライバのセットアップは完了です。

PC の電源を切り PCI ボードを装着してください。



PC を起動後はセットアップしたドライバが自動的にインストールされます。

「(2-3) インストールの確認」へ進み、正常にインストールされていることを確認してください。

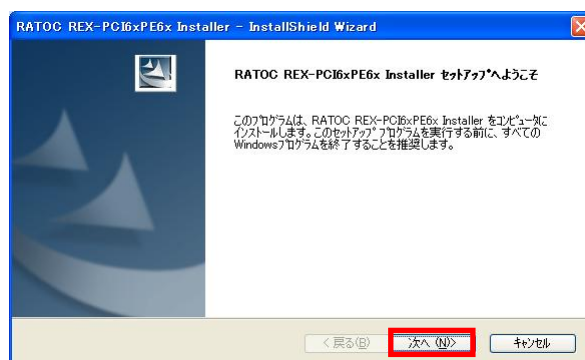
## (2-2) Windows XP (x32/x64) / Server2003 (x32/x64) セットアップ

PCI ボードを接続する前に製品添付 CD-ROM 内のドライバをセットアップします。以下の手順でインストールを行ってください。

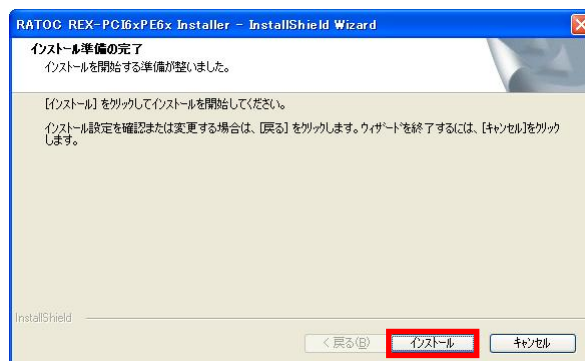
### < ドライバのセットアップ >

#### 「RATOC REX-PCI6xPE6x

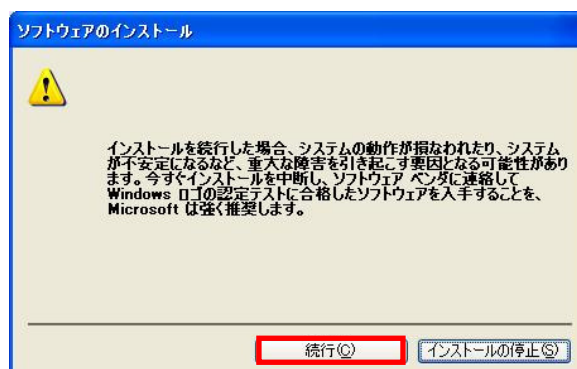
Installer セットアップへようこそ」の画面で「次へ(N)」ボタンをクリックします。

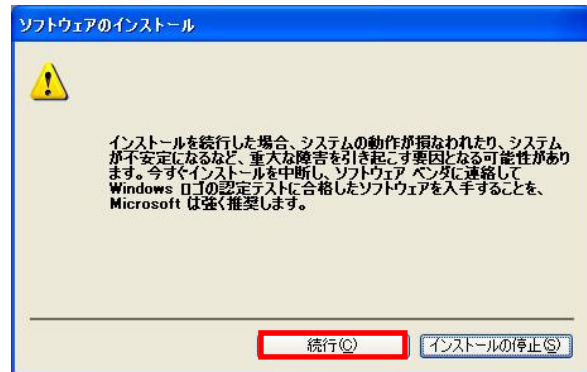


「インストール準備の完了」の画面で「インストール」ボタンをクリックします。



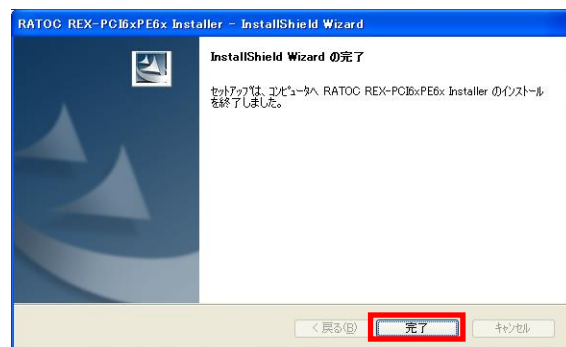
警告画面が 2 回表示されますが「続行(C)」ボタンをクリックします。





以上でドライバのセットアップは完了です。

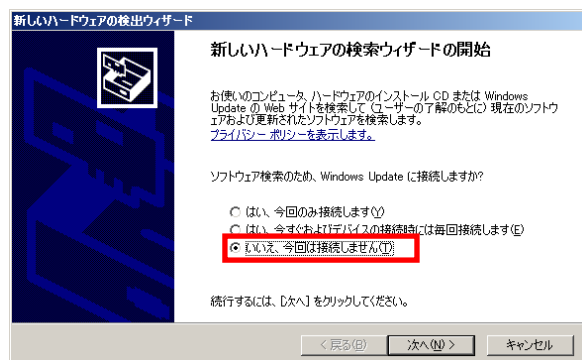
PC の電源を切り PCI ボードを装着してください。



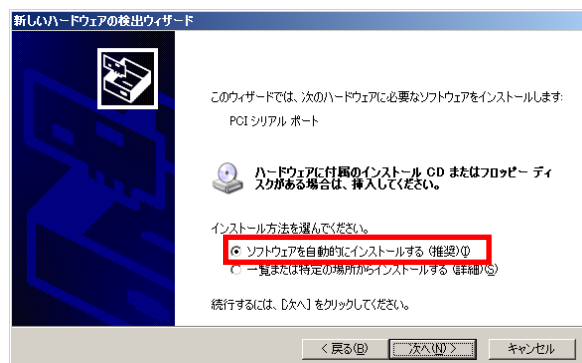
PC を起動後は以下の手順にてインストールを行います。

### <REX-PCI64 4 Port Serial PCI Board のインストール>

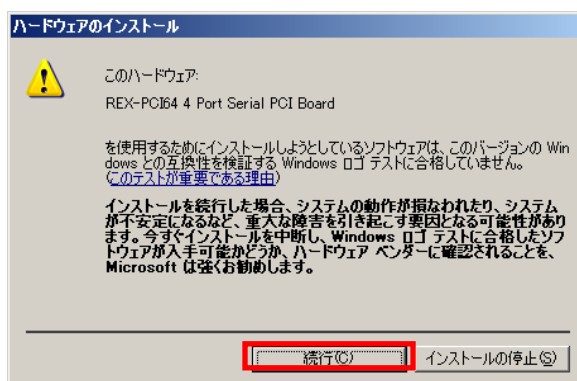
「新しいハードウェアの検索ウィザードの開始」で、「いいえ、今回は接続しません(T)」を選択し「次へ(N)」ボタンをクリックします。



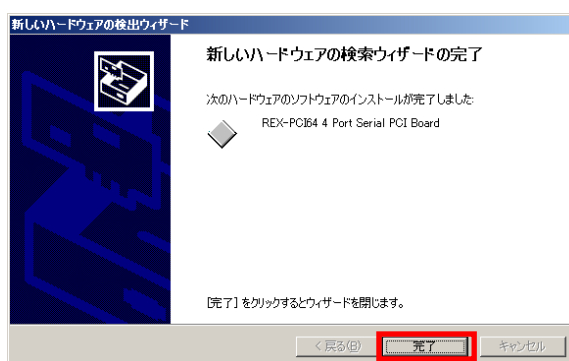
「ソフトウェアを自動的にインストールする(I)」を選択し「次へ(N)」ボタンをクリックします。



「ロゴテストに合格していません」と表示されますが、「続行(C)」ボタンをクリックします。

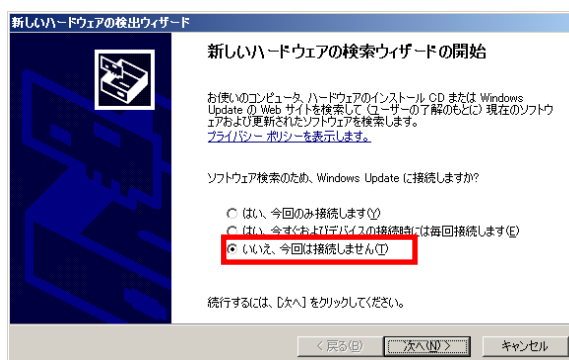


ドライバのコピーが開始され、完了のメッセージが表示されます。「完了」ボタンをクリックします。

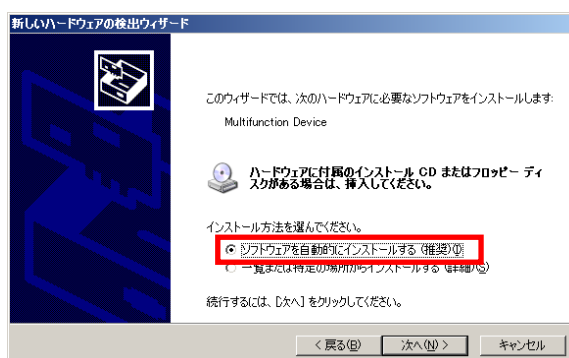


### <REX-PCI64 Board ID のインストール>

「新しいハードウェアの検索ウィザードの開始」で、「いいえ、今回は接続しません(T)」を選択し「次へ(N)」ボタンをクリックします。

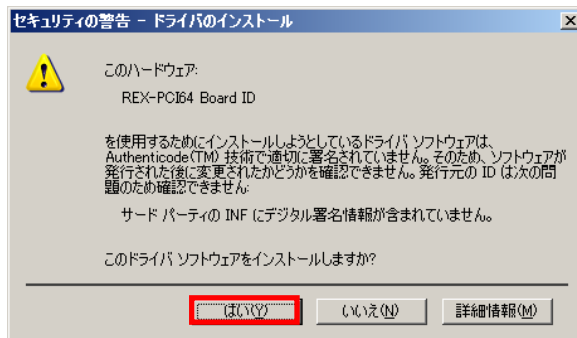


「ソフトウェアを自動的にインストールする(I)」を選択し「次へ(N)」ボタンをクリックします。

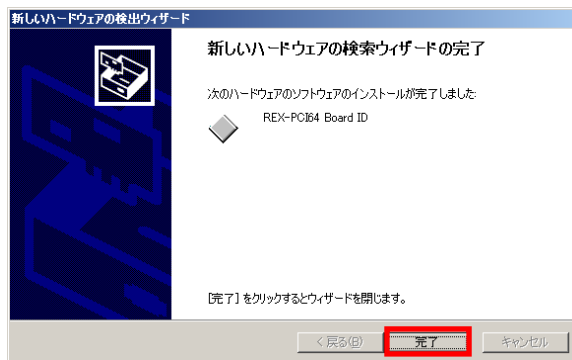


「このドライバソフトウェアをインストールしますか?」と表示される場合は「はい(Y)」ボタンをクリックします。

※WindowsXP ではこのダイアログは表示されません。

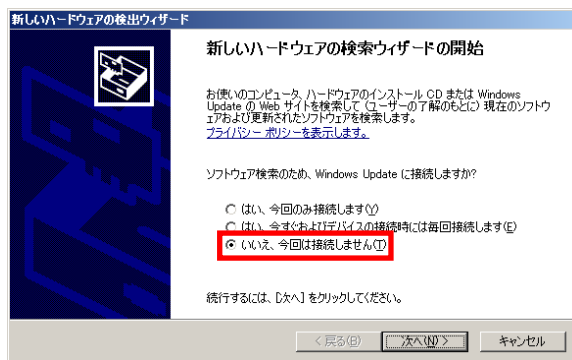


ドライバのコピーが開始され、完了のメッセージが表示されます。  
「完了」ボタンをクリックします。

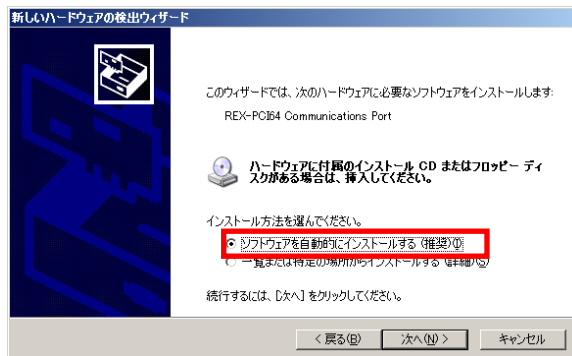


### <REX-PCI64 Communications Port のインストール>

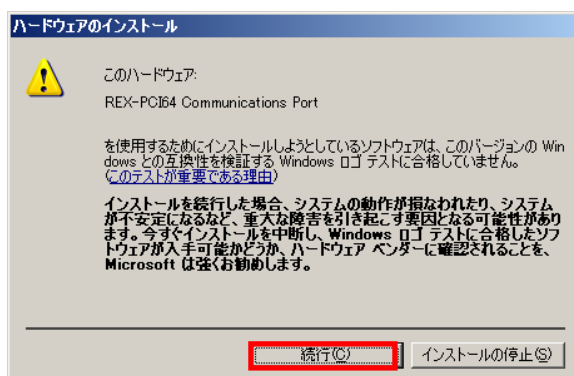
「新しいハードウェアの検索ウィザードの開始」で「いいえ、今回は接続しません(T)」を選択し「次へ(N)」ボタンをクリックします。



「ソフトウェアを自動的にインストールする(I)」を選択し「次へ(N)」ボタンをクリックします。



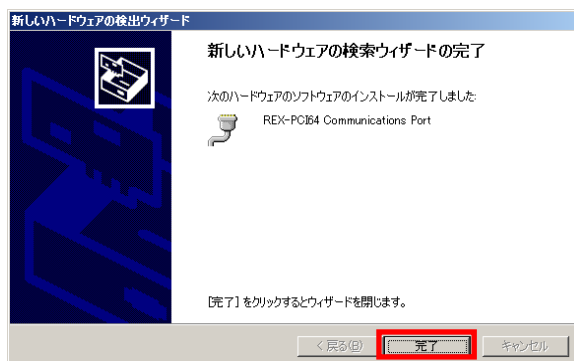
「ロゴテストに合格していません」と表示されますが、「続行(C)」ボタンをクリックします。



ドライバのコピーが開始され、完了のメッセージが表示されます。

「完了」ボタンをクリックします。

4 ポート分についてのインストール作業が必要となりますので、同様の手順で行ってください。



以上で REX-PCI64 のインストールは終了です。

「(2-3) インストールの確認」へ進み、インストールの確認を行ってください。



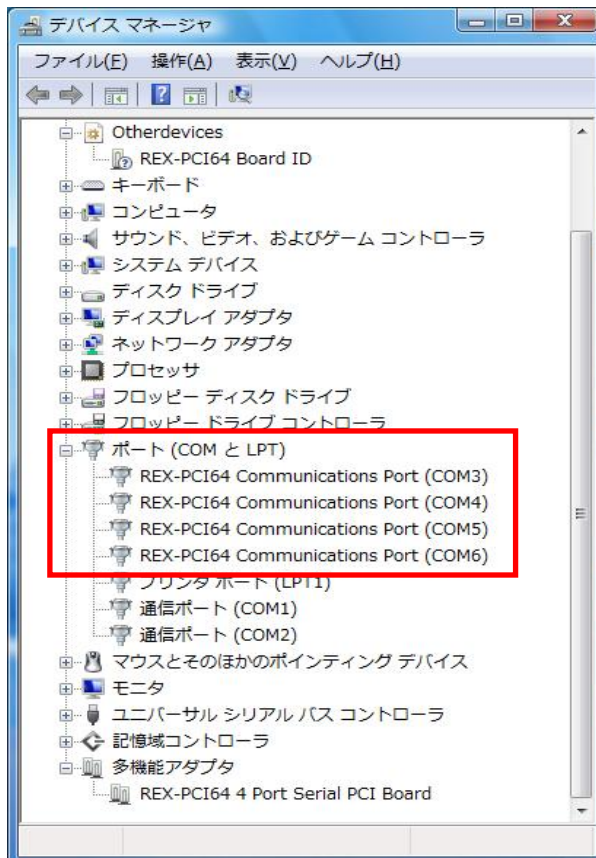
### (2-3) インストールの確認

コントロールパネルの「デバイスマネージャ」を起動します。

「ポート(COM と LPT)」をクリックして新しくポートが追加されているのを確認してください。

(※ Windows7/Vista/Server2008 以外の OS では、コントロールパネルのシステムを起動し、

「システムのプロパティ」の「ハードウェア」タブから「デバイスマネージャ」ボタンをクリックします。)



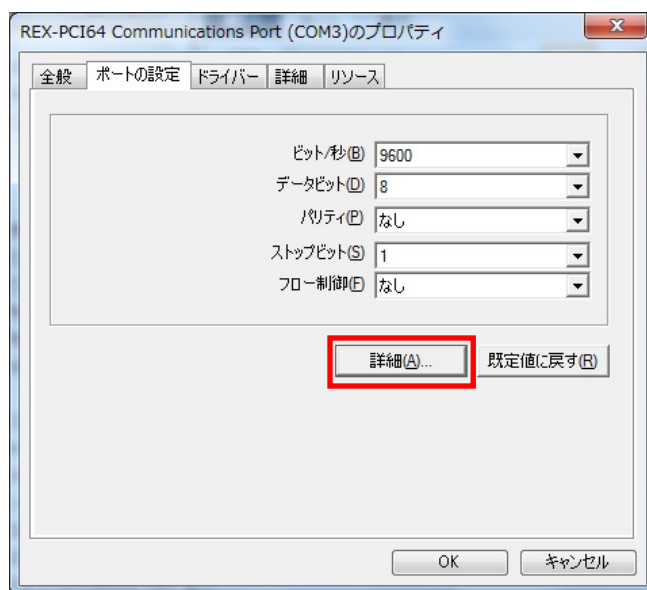
画面では「COM3」「COM4」「COM5」「COM6」となっておりますが、ご使用の環境により COMx の x の数字が異なりますのでご注意ください。

## (2-4) COM ポート番号の変更と設定について

本製品に割り当てられた COM ポート番号の変更はデバイスマネージャ上より行うことができます。

※変更したい COM ポート番号が分からない場合は、サンプルアプリケーション(Page.3-5)を利用して、ロータリスイッチ設定番号・ケーブル番号(各ケーブルに付いているシールの番号)から割り当てられている COM ポート番号を知ることができます。

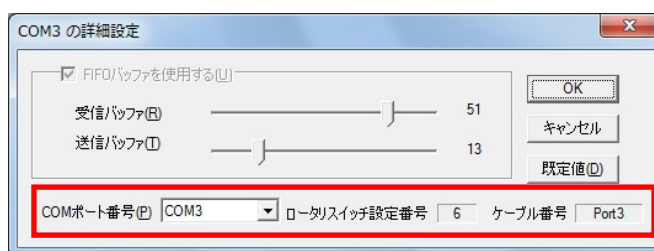
「(2-3) インストールの確認」と同様にポートのプロパティ画面を開き「ポートの設定」タブをクリックします。「詳細」ボタンをクリックするとポートの詳細設定ダイアログが表示されます。



【ポートのプロパティ画面】

COM ポート番号を変更するには「COM ポート番号」コンボボックスより変更先 COM 番号を選択後に「OK」ボタンをクリックしてください。

「ロータリスイッチ設定番号」には REX-PCI64 PCI ボードにあるロータリスイッチの値が表示され、「ケーブル番号」にはこの COM ポート番号が割り当てられているケーブル番号が表示されます。



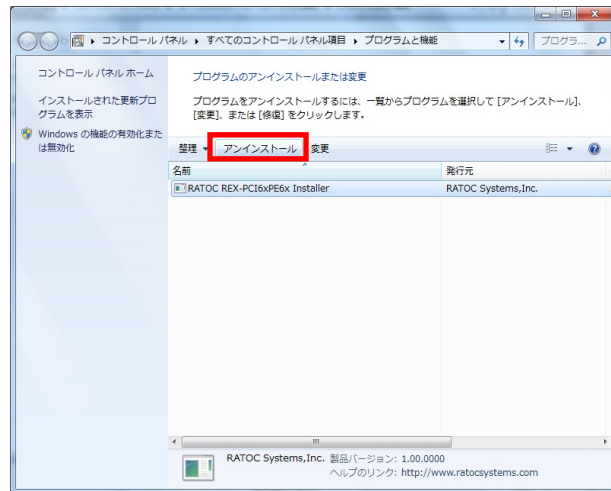
【ポートの詳細設定ダイアログ】

### (2-5) ドライバのアンインストール

コントロールパネルの「プログラムと機能」を起動し、セットアップされたドライバをアンインストールします。

(Windows XP/2000/Server2003 では「プログラムの追加と削除」を起動します。)

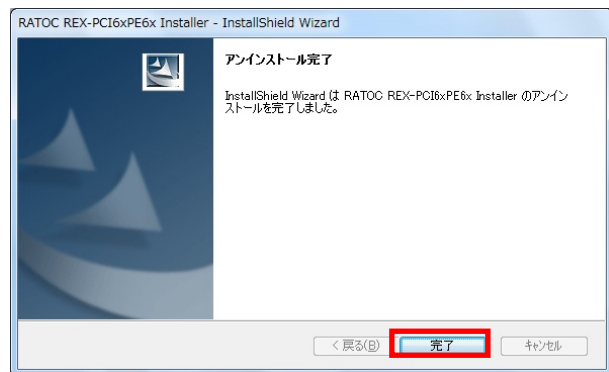
「RATOC REX-PCI6xPE6x Installer」を選択し、「アンインストール」をクリックします。



アンインストールの確認画面が表示されますので「はい(Y)」ボタンをクリックします。



以上で REX-PCI64 のアンインストールは完了です。



## (2-6) Windows98SE セットアップ

### Windows98SE でのインストール方法

PCI ボードを PCI スロットに挿入後、PC を起動すると右の「新しいハードウェアの追加ウィザード」が表示されます。「次へ」を押します。



ドライバの検索方法は「使用中のデバイスに最適なドライバを検索する(推奨)」を選択し、「次へ」を押します。



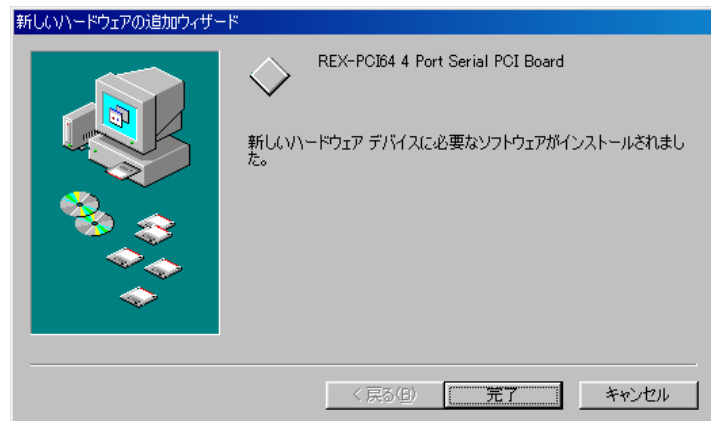
検索する場所に「検索場所の指定(L)」を選択後、添付の CD-ROM を挿入し、「参照(R)」ボタンをクリックしてください。表示されたダイアログで [Win9x] フォルダを指定後、「OK」ボタンを押してください。「次へ」を押します。



ドライバのある場所が表示されます。「次へ」を押します。



ドライバのコピーが開始され、完了のメッセージが表示されます。「完了」を押します。



以上で REX-PCI64 のインストールは終了です。

「(2-8) インストールの確認 (Windows Me/98SE)」へ進み、インストールの確認作業を行ってください。

## (2-7) WindowsMe セットアップ

### WindowsMe でのインストール方法

PCI ボードを PCI スロットに挿入後、PC を起動すると右の「新しいハードウェアの追加ウィザード」が表示されます。

ここでは「適切なドライバを自動的に検索する(推奨)」にチェックを入れ「次へ」を押します。



ドライバのコピーが開始され、完了のメッセージが表示されます。「完了」を押します。



以上で REX-PCI64 のインストールは終了です。

「(2-8) インストールの確認 (Windows Me/98SE)」へ進み、インストールの確認作業を行ってください。

## (2-8) インストールの確認(Windows Me/98SE)

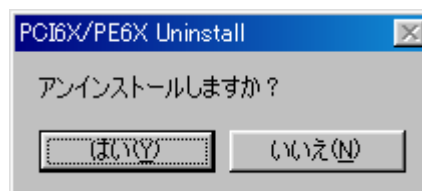
コントロールパネルのシステムを起動し、デバイスマネージャを選択します。ボードの設定が正常に行われていれば、コンピュータのレジストリツリー「ポート(COMとLPT)」の下に「通信ポート(COMx)」が登録されます。

画面では、「COM2」「COM3」「COM5」「COM6」となっておりますが、お客様の環境により COMx の x の数字が異なりますのでご注意ください。



## (2-9) ドライバのアンインストール(Windows Me/98SE)

[Windows 98SE/Me] の場合  
(CD-ROM)¥Win9x¥Win9x  
にある Uninstall.exe を実行します。

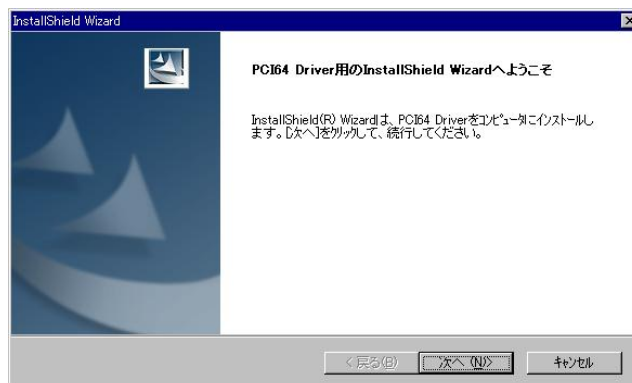


## (2-10) Windows NT 4.0 セットアップ

WindowsNT4.0 では、添付の CD-ROM からインストーラを起動する必要があります。システムのリソースの空き状況によっては REX-PCI64 を使用できないことがあります。

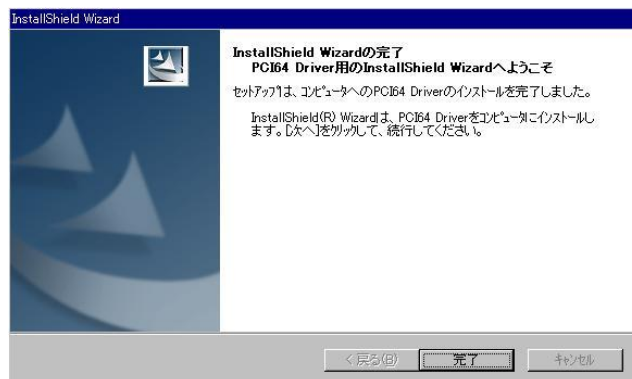
PCI ボードを接続する前に添付の CD-ROM を挿入し、WinNT¥PCI64 からインストーラを起動します。

「次へ」ボタンをクリックします。



「完了」ボタンをクリックします。

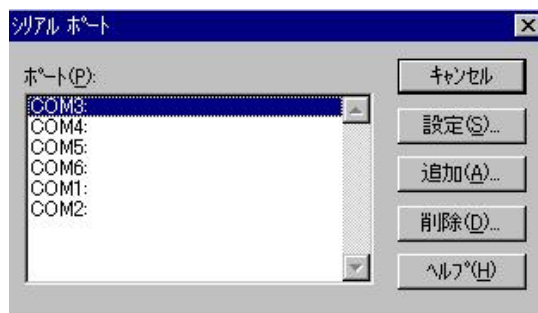
PC をシャットダウンし、PCI スロットに REX-PCI64 を接続してください。



以上で REX-PCI64 のインストール作業は完了となります。

### ◆ PCI ボード設定内容の確認

「スタートメニュー」→「設定」→「コントロールパネル」→「シリアルポート」に新しいポートが追加されていることを確認してください。





## 第3章 設定ポート確認用ライブラリ関数

### (3-1) ライブラリ関数について

本製品に添付しているライブラリ関数を利用すると、ロータリスイッチ番号とケーブル番号から割り当てられている COM ポート番号を知ることができます。

また、割り当てられている COM ポート番号のロータリスイッチ番号とケーブル番号を知ることができます。

### (3-2) 関数仕様

ライブラリ関数と機能の一覧は次の通りとなります。

関数名	機能
GetComPort	REX-PCI64 に割り当てられているポート数およびポート番号の取得
GetBoardIdCableNo	指定したポート番号のロータリスイッチの値とケーブル番号を取得
GetBoardId	REX-PCI64 の枚数およびロータリスイッチの値を取得
GetCableNo	各ケーブルに割り当てられている COM ポート番号を取得

※ライブラリ関数仕様の詳細につきましては次ページ以降をご参照ください。

<b>GetComPort</b>	REX-PCI64 に割り当てられているポート数 およびポート番号の取得
-------------------	---

**書式** CHAR GetComPort( PCHAR pComPort, CHAR ComPortSize )

**引数** PCHAR pComPort COM ポート番号を格納する配列へのポインタ  
CHAR ComPortSize 第一引数で確保された配列のサイズ

**戻値** REX-PCI64 にアサインされている全 COM ポート数が返されます。

**機能** 引数の pComPort に NULL をセットすると、REX-PCI64 にアサインされている COM ポートのトータル数が戻り値として返されます  
また、ComPortSize に REX-PCI64 にアサインされているポート数を指定すると、COM ポート番号の値が pComPort[0], pComPort[1]…にセットされます。

#### VB での宣言 (Visual Studio 6.0)

```
Declare Function GetComPort Lib "rsui.dll" (ComPort As Any, ByVal  
ComPortSize As Byte) As Byte
```

#### VB での宣言 (Visual Studio 2005)

```
Declare Function GetComPort Lib "rsui.dll" (ByVal ComPort As String,  
ByVal ComPortSize As Byte) As Byte
```

#### 使用例

```
// REX-PCI64 にアサインされている COM ポート数を取得  
ComNum = GetComPort( NULL, 0 );  
if( ComNum != 0 ) {  
    // COM ポート分のメモリ確保  
    pCom = (PCHAR)LocalAlloc( LPTR, ComNum );  
    // アサインされた全ての COM ポートの番号を取得  
    GetComPort( pCom, ComNum );  
    // COM ポート番号を用いた処理  
    .....  
    // メモリ解放  
    LocalFree( pCom ); }
```

<b>GetBoardIdCableNo</b>	指定したポート番号のロータリスイッチの値とケーブル番号を取得
--------------------------	--------------------------------

**書式** BOOL GetBoardIdCableNo( PCHAR pComPort, PCHAR pBoardId, PCHAR pCableNo )

**引数**

PCHAR pComPort	COM ポートの番号を文字列でセット
PCHAR pBoardId	対応するロータリスイッチの値を格納する配列へのポインタ
PCHAR pCableNo	対応するケーブル No を格納する配列へのポインタ

**戻値** pComPort で指定されたポート番号が REX-PCI64 にアサインされている場合は TRUE が返されます。その他のシリアルポートにアサインされている番号、または存在しないポート番号を指定した場合は FALSE が返されます。

**機能** 指定したポート番号の REX-PCI64 ロータリスイッチの値とケーブル No を取得します。

#### VB での宣言 (Visual Studio 6.0)

```
Declare Function GetBoardIdCableNo Lib "rsui.dll" (ByVal ComPort As String, BoardId As Any, CableNo As Any) As Boolean
```

#### VB での宣言 (Visual Studio 2005)

```
Declare Function GetBoardIdCableNo Lib "rsui.dll" (ByVal ComPort As String, ByVal BoardId As String, ByVal CableNo As String) As Boolean
```

#### 使用例

```
CHAR BoardId;
CHAR CableNo;
// 指定した COM ポートにアサインされた BoardId, CableNo を取得
if( GetBoardIdCableNo( "COM5", &BoardId, &CableNo ) ) {
    // ロータリスイッチ又はケーブル No を用いた処理
    ..... }
}
```

<b>GetBoardId</b>	<b>REX-PCI64 の枚数およびロータリスイッチの値を取得</b>
-------------------	--------------------------------------

**書式** CHAR GetBoardId( PCHAR BoardId, CHAR BoardIdSize )

**引数** PCHAR pBoardId     ロータリスイッチの値を格納する配列へのポインタ  
CHAR BoardIdSize    第一引数で確保された配列のサイズ

**戻値** PC 上で認識されている REX-PCI64 の枚数が返されます。複数枚の REX-PCI64 が認識されていて、ロータリスイッチの値が重複している場合はエラーとなります。VC の場合は-1、VB の場合は 0xFF が返されます。

**機能** 引数の BoardId に NULL をセットして本関数を呼び出すと、PC 上で認識されている REX-PCI64 の枚数が返されます。

また、PC 上で認識されている複数枚の REX-PCI64 についてのロータリスイッチの値を列挙し第一引数に格納します。

(例) 2 枚の REX-PCI64 のロータリスイッチの値が “1” , “2” と設定されていた場合、BoardId[0] = 1, BoardId[1] = 2 とセットされます。

#### VB での宣言 (Visual Studio 6.0)

```
Declare Function GetBoardId Lib "rsui.dll" (BoardId As Any , ByVal BoardIdSize As Byte) As Byte
```

#### VB での宣言 (Visual Studio 2005)

```
Declare Function GetBoardId Lib "rsui.dll" (ByVal BoardId As String , ByVal BoardIdSize As Byte) As Byte
```

#### 使用例

```
// REX-PCI64 の接続数取得
BoardNum = GetBoardId ( NULL, 0 );
if(BoardNum != 0 ) {
    pBoard = (PCHAR)LocalAlloc( LPTR, BoardNum ); // メモリ確保
    // 接続した全てのロータリスイッチの値を取得
    GetBoardId ( pBoard, BoardNum );
    //ロータリスイッチの値を用いた処理
    .....
    LocalFree( pBoard ); } // メモリ解放
```

**GetCableNo**

各ケーブルに割り当てられている COM ポート番号を取得

**書式** BOOL GetCableNo( CHAR BoardId, PCHAR ComPort, CHAR ComPortSize )

**引数** CHAR BoardId           ロータリスイッチの値をセット  
 PCHAR ComPort           COM ポート番号を格納する配列へのポインタ  
 CHAR ComPortSize        第二引数で確保された配列のサイズ

**戻値** 指定したロータリスイッチの値が設定された REX-PCI64 の検出と、そのボードに対応する COM ポート番号を取得できた場合は TRUE が返されます。指定したロータリスイッチの値が設定された REX-PCI64 が検出されなかった場合は FALSE が返されます。

**機能** 指定したロータリスイッチの値が設定されている REX-PCI64 について、ケーブル No が Port1 から Port4 の順に、第二引数で指定された配列に COM ポート番号がセットされます。

**VB での宣言 (Visual Studio 6.0)**

```
Declare Function GetCableNo Lib "rsui.dll" (ByVal BoardId As Byte,
ComPort As Any, ByVal ComPortSize As Byte) As Byte
```

**VB での宣言 (Visual Studio 2005)**

```
Declare Function GetCableNo Lib "rsui.dll" (ByVal BoardId As Byte,
ByVal ComPort As String, ByVal ComPortSize As Byte) As Byte
```

**使用例**

```
CHAR BoardId;
CHAR ComPort[4];
BoardId = 0;
// 指定した BoardId の Port1~Port4 に割り当てられた COM 番号を取得
if( GetCableNo( BoardId, ComPort, sizeof(ComPort) ) ) {
    //COM ポート番号を用いた処理
    ..... }
}
```

### (3-3) 設定ポート確認サンプルアプリケーションの構成について

製品添付の CD-ROM の[ViewSample]フォルダ内の構成は次のようになります。

- VB6.0 フォルダ・・・VisualBasic6.0 サンプル
- VC6.0 フォルダ・・・VisualC++6.0 サンプル
- VB2005 フォルダ・・・VisualBasic2005 サンプル
- VC2005 フォルダ・・・VisualC++2005 サンプル

### (3-4) ライブラリ関数の呼び出し

#### VCからの呼び出し

Visual C/C++のアプリケーションから、製品に添付された DLL(rsui.dll)の API を呼び出すには、以下の関数についての呼び出し宣言を行い、それらを参照する必要があります。(rsui.dll は REX-PCI64 のドライバインストール時に system32 フォルダへコピーされます。)

```
CHAR GetComPort( PCHAR pComPort, CHAR ComPortSize )
BOOL GetBoardIdCableNo( PCHAR pComPort, PCHAR pBoardId, PCHAR pCableNo )
CHAR GetBoardId( PCHAR BoardId, CHAR BoardIdSize )
BOOL GetCableNo( CHAR BoardId, PCHAR ComPort, CHAR ComPortSize )
```

#### VC6.0 / VC2005 サンプルでの呼び出し例 (SampleDlg.cpp より抜粋)

```
// rsui.dll 関数呼び出し宣言
CHAR (__stdcall*fnGetComPort)( CHAR*, CHAR );
BOOL (__stdcall*fnGetBoardIdCableNo)( CHAR*, CHAR*, CHAR* );
CHAR (__stdcall*fnGetBoardId)( CHAR*, CHAR );
BOOL (__stdcall*fnGetCableNo)( CHAR, CHAR*, CHAR );

BOOL CSampleDlg::OnInitDialog()
{
    // rsui.dll をロード
    hRsui = LoadLibrary( "Rsui.DLL" );
    if( !hRsui ){
        index = GetLastError();
        MessageBox( "LoadLibrary エラー", "失敗", MB_OK );
        return FALSE;
    }
    // rsui.dll 内のエクスポート済み関数のアドレスを取得
    fnGetComPort = ( CHAR (__stdcall*)( CHAR*, CHAR ) )GetProcAddress( hRsui, "GetComPort" );
    fnGetBoardIdCableNo = ( BOOL (__stdcall*)( CHAR*, CHAR*, CHAR* ) )GetProcAddress( hRsui,
        "GetBoardIdCableNo" );
    fnGetBoardId = ( CHAR (__stdcall*)( CHAR*, CHAR ) )GetProcAddress( hRsui, "GetBoardId" );
    fnGetCableNo = ( BOOL (__stdcall*)( CHAR, CHAR*, CHAR ) )GetProcAddress( hRsui,
        "GetCableNo" );
}
```

### VB からの呼び出し

Visual Basic のアプリケーションから、製品に添付された DLL(rsui.dll)の API を呼び出すには、関数についての呼び出し宣言を行い、それらを参照する必要があります。

#### VB6.0 サンプルでの宣言 (RSUI.bas)

```
Declare Function GetComPort Lib "rsui.dll" (ComPort As Any, ByVal ComPortSize As Byte) As Byte
Declare Function GetBoardIdCableNo Lib "rsui.dll" (ByVal ComPort As String, BoardId As Any,
                                                    CableNo As Any) As Boolean
Declare Function GetBoardId Lib "rsui.dll" (BoardId As Any, ByVal BoardIdSize As Byte) As Byte
Declare Function GetCableNo Lib "rsui.dll" (ByVal BoardId As Byte, ComPort As Any, ByVal
                                                    ComPortSize As Byte) As Boolean
```

#### VB2005 サンプルでの宣言 (rsui.vb)

```
Module rsui
Declare Function GetComPort Lib "rsui.dll" (ByVal ComPort As String , ByVal ComPortSize As Byte)
                                                    As Byte
Declare Function GetBoardIdCableNo Lib "rsui.dll" (ByVal ComPort As String, ByVal BoardId As
                                                    String, ByVal CableNo As String) As Boolean
Declare Function GetBoardId Lib "rsui.dll" (ByVal BoardId As String , ByVal BoardIdSize As Byte)
                                                    As Byte
Declare Function GetCableNo Lib "rsui.dll" (ByVal BoardId As Byte, ByVal ComPort As String, ByVal
                                                    ComPortSize As Byte) As Byte
End Module
```



### (3-5) 設定ポート確認サンプルアプリケーションについて

(COMポート番号から情報取得)

GetComPort()を用いてREX-PCI64に割り当たっているCOMポート番号を全て列挙します。GetBoardIdCableNo()を用いてコンボボックスに列挙されたCOMポート番号を指定することで、ロータリスイッチの値とケーブル番号を呼び出します。



(ロータリスイッチから情報取得)

GetBoardId()を用いてREX-PCI64のロータリスイッチの値を全て列挙します。GetCableNo()を用いてコンボボックスに列挙されたロータリスイッチの値を指定することで、割り当てられているPort1からPort4までのCOMポート番号を列挙します。

関数を機能させるためには、4ポートすべてが有効になっている必要がありますので、デバイスマネージャ上で無効としないでください。

## 第4章 通信サンプルアプリケーション

### (4-1) 通信サンプルアプリケーションの構成について

製品添付の CD-ROM の[CommSample]フォルダ内の構成は次のようになります。

- VB6 フォルダ・・・VisualBasic6.0 サンプル
- VC6 フォルダ・・・VisualC++6.0 サンプル
- VB2005 フォルダ・・・VisualBasic2005 サンプル
- VC2005 フォルダ・・・VisualC++2005 サンプル

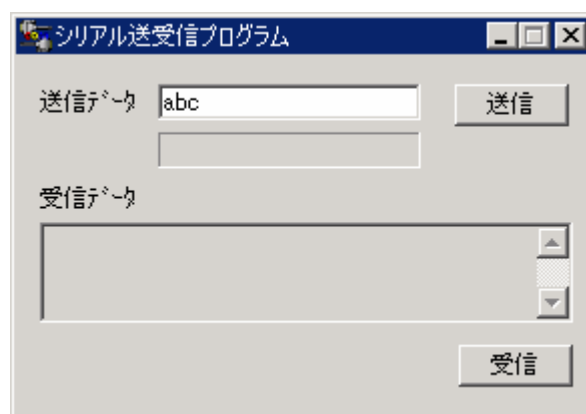
### (4-2) 通信サンプルアプリケーションについて

通信サンプルアプリケーションは ASCII 文字列を送受信する簡易プログラムです。

使用するポートを選択します。



入力した ASCII 文字列の送信、および接続先から送信されたデータの受信を行います。



## 通信サンプルプログラム抜粋(VC)

(Visual Basic についてはサンプルソース内をご参照ください)

```

LRESULT CALLBACK WndProc(HWND hWnd, UINT uMessage, WPARAM wParam, LPARAM lParam)
{
    switch (uMessage)
    {
        case WM_CREATE:
            // COMポートオープン
            hComPort = CreateFile( lpszComName,
                                GENERIC_READ|GENERIC_WRITE,
                                0,
                                NULL,
                                OPEN_EXISTING,
                                FILE_FLAG_OVERLAPPED,
                                NULL);

            if ( hComPort == INVALID_HANDLE_VALUE ) {
                // ハンドルエラー
                ShowError("COM Open Error.");
                return FALSE;
            }
            // DCB 設定
            memset(&dcb, 0, sizeof(dcb));
            dcb.DCBlength = sizeof(dcb);
            dcb.BaudRate = 9600;
            dcb.fBinary = 1;
            dcb.fDtrControl = DTR_CONTROL_ENABLE;
            dcb.fOutxCtsFlow = 1;
            dcb.fRtsControl = DTR_CONTROL_HANDSHAKE;
            dcb.Parity = NOPARITY;
            dcb.StopBits = ONESTOPBIT;
            dcb.ByteSize = 8;
            // 新たに通信パラメータを設定する
            if ( !SetCommState( hComPort, &dcb ) )
                ShowError("Set COM parameter error.");

            break;

        case WM_COMMAND:
            switch (wParam)
            {
                case IDB_TXDATA:
                    // 送信データ取得
                    memset( TxBuf, 0x00, sizeof( TxBuf ) );
                    GetDlgItemText( hWnd, IDE_TXDATA, TxBuf, sizeof(TxBuf) );
                    SetDlgItemText( hWnd, IDS_TXDATA, TxBuf );
                    SetDlgItemText( hWnd, IDE_TXDATA, "" );
                    nToWrite = strlen(TxBuf);
                    // COMポートにデータ送信
                    iRet = WriteFile( hComPort, TxBuf, nToWrite, &dwBytesWrote, &ov);
                    if ( iRet == 0 ) {
                        WaitForSingleObject(ov.hEvent, 1000);
                    }
                    break;
            }
    }
}

```

/\* 次ページに続く \*/

```
        case IDB_RXDATA:
            // 受信スレッドを作成します
            hThread = CreateThread( NULL,
                                   0,
                                   (LPTHREAD_START_ROUTINE) ReadThread,
                                   hWnd,
                                   0,
                                   &ThreadId );

            break;
    }
    break;
case WM_DESTROY:
    if (hThread != NULL) {
        CloseHandle( hThread );
        fReadThread = FALSE;
    }
    PostQuitMessage( 0 );
    break;
default:
    return DefWindowProc( hWnd, uMessage, wParam, lParam );
}return 0;
}

DWORD WINAPI ReadThread( LPVOID lpParameter )
{
    // バイト受信イベントを待つ受信データを取り出し格納
    while( fReadThread ) {
        // イベントを待つ
        WaitCommEvent( hComPort, &dwEvent, &ov );
        if ( WaitForSingleObject( ov.hEvent, INFINITE ) == WAIT_OBJECT_0 ) {
            do {
                memset( RxBuf, 0, sizeof( RxBuf ) );
                if ( !ReadFile( hComPort, RxBuf, sizeof( RxBuf ), &dwBytesRead, &ov ) ) {
                    if ( WinError = GetLastError() == ERROR_IO_PENDING ) {
                        if ( !GetOverlappedResult( hComPort, &ov, &dwBytesRead, TRUE ) ) {
                            ShowError( "GetOverlappedResult failed" );
                            break;
                        }
                    }
                }
            }
            else {
                if ( WinError != ERROR_INVALID_HANDLE ) {
                    ShowError( "ReadFile failed" );
                    break;
                }
            }
        }
        if ( dwBytesRead > 0 ) {
            // 受信データ表示
            RxBuf[dwBytesRead] = 0x00;
            SetDlgItemText( hWnd, IDS_RXDATA, RxBuf );
        }
    }while ( dwBytesRead > 0 && fReadThread != FALSE );
}
return 0;
}
```

製品に対するお問い合わせ

REX-PCI64 の技術的なご質問やご相談の窓口を用意していますのでご利用ください。

ラトックシステム株式会社  
I&L サポートセンター  
〒556-0012  
大阪市浪速区敷津東 1-6-14 朝日なんばビル  
TEL. 06-6633-6741  
FAX. 06-6633-8285  
〈サポート受付時間〉  
月曜～金曜（祝祭日は除く）AM 10:00 - PM 1:00  
PM 2:00 - PM 5:00

また、インターネットのホームページでも受け付けています。  
HomePage ➡ <http://www.ratocsystems.com>

🔔 ご注意 🔔

- ☑本書の内容については、将来予告なしに変更することがあります。
- ☑本書の内容につきましては万全を期して作成しましたが、万一ご不審な点や誤りなどお気づきになられましたらご連絡願います。
- ☑本製品および本製品添付のマニュアルに記載されている会社名および製品名は、各社の商品または登録商標です。
- ☑運用の結果につきましては、責任を負いかねますので、予めご了承ください。


REX-PCI64 質問用紙
----------------

●下記ユーザ情報をご記入願います。

法人登録 の方のみ	会社名・学校名			
	所属部署			
ご担当者 名				
E-Mail				
住所	〒			
TEL		FAX		
製品型番		シリアルNo.		
ご購入情 報	販売店名		ご購入日	

●下記運用環境情報とお問い合わせ内容をご記入願います。

【パソコン/マザーボードのメーカー名と機種名】
【ご利用のOS】
【接続機器】
【お問合せ内容】
【添付資料】

 個人情報取り扱いについて

ご連絡いただいた氏名、住所、電話番号、メールアドレス、その他の個人情報は、お客様への回答など本件に関わる業務のみに利用し、他の目的では利用致しません。

