

MR550 プログラミングマニュアル

ユニテック・ジャパン株式会社

2004.7 初版

1	概要		4
2	SDK &	· I+2	5
2		.lo.:	
2	7 ±	n ナネスハけ磁気カードリーダから データを詰っ込 む女法	5
3	74	ャノのるいは做気カートリータからナータを読み込む方法・・・・・・・・・・・・・・・	
3.	1 概要		5
3.	2 " Sc	anner3.dll "の関数コールリスト	7
	3.2.1	デコーダを使用可能にする	7
	3.2.2	デコーダを使用不可能にする	7
	3.2.3	ハーコード人刀チェック	/
	3.2.4	ハーコードテーダを読む	/
	3.2.5 2.2.6	幽気人トフイノリーダ(MSR) 八月のナエック	8 ە
	3.2.0 2.2.7	「ໝメストライブリーク(MSR) ブークを読む	۰٥ ۵
	3227	ブコーグチップからバーコートシンボル設定を得る	3 16
	329	リコーノノノノン 5/パーコー 1 ノノ が 2 k b c c r 2 · · · · · · · · · · · · · · · · · ·	16
	3.2.10	すべてのシンボルを標準値にリセット	
3.	3 " Sc	ankey3.dll "の関数コールリスト	17
	3.3.1	デコーダを有効にする	17
	3.3.2	デコーダを無効にする	17
	3.3.3	バーコードシンボルをデコーダチップにセット	17
	3.3.4	デコーダチップからバーコードシンボル設定を得る	17
	3.3.5	DLL バージョン番号を得る	17
	3.3.6	フリアンフル文字列をセット	18
	3.3.7	ノリアンフル乂子列を侍る	18
	3.3.8	小人トアノノル乂子州をビット	10
	3.3.9 3310	小人下アノフル文子列を守る	18
	3 3 11	区切り文子のビフト・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	18
	3 3 12	すべてのシンボルを標準値にセット	19
	0.0.12		
		○ プロゲニンンゲッキャック通知なニノブニリ	00
4	MKSS	U ノロクフミノクのにのの() 使利なフィノフリ	. 20
	4.1.1		20
	4.1.Z / 1 3	リレー 2 コノドロール	20
	 4 1 4	リレー 2 のステータスを得る	∠0 20
	4.1.5	デジタル入力を得る	
	4.1.6	セキュリティアラームを有効にする	21
	4.1.7	セキュリティアラームを無効にする	21
F		ヘレ_ヘノヘュ, with cov またけ wear とえ プロ ゲニミング	00
C	VI 3U	AL 0/0++ WIIΠ 3UN まには MF0 によるノロクフミノク	. ZZ

目 次

- 2 -

5.1	システムの必要条件:	22
5.2	インストール	23
5.3	VC++でWindows CE プログラムを作成する方法	24
6	VISUAL BASIC によるプログラミング	28
6.1	インストール	28
6.2	VB で Windows CE プログラムを作成する方法	28

1 概要

本マニュアルは Window CE 用に提供されている各種の開発キットやツールについて、そしていろいろなプログラミ ングインタフェースの概要について説明しています。Windows CE のアプリケーションプログラムは、Windows CE デバイスに合った開発、エミュレート、そしてリモート・デバッグをするために標準の Microsoft 開発環境 IDE – Embedded Visual C++ と Embedded Visual Basic を含む Embedded Visual Tools (EVT) で開発することができま す。ツールは Win32、COM、ActiveX、MFC、ATL をサポートしています。

Win32 API セットには、複数の機能が同じタスクを果たすことのできるという意味で重複した機能が含まれています。 一般にデスクトップ PC ソフトウェアの開発者はわずかの重複 OS コードは気にしませんが、Window CE の開発者 にとって、コードのサイズは重要です。Windows CE のリリースでは、Win32 API の特定のサブセットが提供されて います。(サポートされている API の詳細に関しては、Embedded Visual Tools のリリースノートをご覧ください。)

Microsoft 社の Component Object Model、あるいは COM は、実行時にクェリーができるしっかりしたコンポーネントを作成するための標準が提供されています。各 COM オブジェクトは、インターフェースや関連した論理メソッドを 公開しています。ベース インターフェース *IUnkown* は、サポートされたインターフェースについて特別なオブジェクトを COM オブジェクトがクェリーを行うことができます。COM モデルは言語には無関係で、コンポーネントは、コーラーに変更を要求することなく言語とは関係なく更新されます。

ActiveX コントロールは、特殊なタイプの COM オブジェクトです。Visual Basic 拡張 (VBX)で始まり、次に OLE コ ントロール (OCX) が続く拡張可能なコントロールの最新の仕様を表します。ActiveX コントロールは通常、ユーザ・ インターフェースを表し、プロパティ、メソッド、そしてイベントを公開します。コントロールは、COM インターフェース の特定セットを通して Visual Basic などのコントロール コンテナと関係します。プロパティ、メソッド、およびイベント を公開することにより、ActiveX コントロールは、スクリプトで操作することができます。

Microsoft Foundation Classes (MFC)は、C++で Windows アプリケーションを開発するクラスライブラリです。これ は Win32 API セットとほとんど同じ機能ですが、完全なオブジェクト指向アプリケーション・フレームワーク内にあり ます。Active Template Library (ATL)は、ActiveX コントロールとその他の COM コンポーネントを作成するために 特にデザインされた C++ のテンプレート ライブラリです。テンプレート クラスを使うことで ATL は、継承(インヘリタ ンス)のみを使う MFC より効率的にソフトウェアを作成できます。

主要な通信インターフェースは、Windows Sockets、あるいはシリアル接続、Ethernet、RF、あるいは赤外線ポート による通信に TCP/IP を使う Winsock です。Windows CE はまた、シリアル API、Telephony API(TAPI)、Remote Access Service (RAS)、FTP と HTTP サービスを提供する WinINet API、ネットワーク サービスをエミュレートして 接続の管理をする Windows ネットワーキング API などの通信関連 Win32 API セットもサポートします。

2 SDK とは?

Microsoft は異なる Windows CE プラットフォーム用に 3 つの SDK (Pocket PC, Palm PC と H/PC Pro)を提供して います。これはサポートされるモジュール、GUI そして各プラットフォームのコンポーネントに違いがあるからです。 ユーザがアプリケーションのコンパイルに不適当な SDK を使用すると予期できない問題が発生することがあります。 ユーザはアプリケーションプログラムを開発する場合にターゲットプラットフォームに合った SDK を選択しなければ なりません。Unitech のターミナルについても各プラットフォームにあった SDK を提供しております。皆様はユニテ ック・ジャパン(株)にお問い合わせになるか、以下の URL から最新バージョンを入手することができます。

http://adc.unitech.com.tw/pub/cs/software/MR550SDK/MR550SDK.zip

3 スキャナあるいは磁気カードリーダからデータを読み込む方法

3.1 概要

MR550 と標準の HPC/PalmPC の大きな違いはバーコード入力ができるかどうかです。Windows CE のリファレン スマニュアルにはバーコード入力についての情報は含まれていません。このセクションはバーコードサブシステム のプログラミング構造とMR550 のプログラミング・ユーティリティ・ライブラリを紹介しています。 MR550 の内部には、 バーコードスロットリーダと磁気カードリーダのコントロールとバーコードのデコードを行うための高機能なデコーダ チップがあります。以下はMR550 バーコードのシステム図です:



上記の図にあるように、MR550 はシリアル ポート COM2 経由でデコーダチップと通信します。その通信パラメータ は、38400,N,8,1 に固定されています。一般にデコーダチップは、COM2 がオープンしていないときはスリープ状 態です。デコーダチップは COM2 がオープンすると作動し、カードが読み込まれるとバーコードスロットリーダから のバーコード"信号"がデコードされます。デコードの後は、バーコード データとそのバーコードのタイプが直接 MR550 に送られます。

プログラマの皆様が、特に、バーコードやシリアルポートのコントロールに慣れていない場合は、プログラム言語だけでデコーダチップをコントロールすることが難しいことに気づかれるでしょう。このため、ユニテック社はデコーダチップをコントロールするためにユーザあるいはアプリケーションのプログラマの皆さんに以下のユーティリティライブラリとプログラムを提供しています。

- アプリケーションプログラム "Bar2Key.exe"はレーザースキャナから入力データを読み、そして MR550のキ ーボードバッファに直接データを入力する便利なアプリケーションプログラムです。Bar2Key.exe はバーコ ードデータの入力を簡単にし、そして COM ポートのプログラミングに慣れていないプログラマに特に役に立 ちます。ユーザプログラムは単にキーボードからバーコードデータを読みます。バーコードシンボルの設定 については、サポートされているシンボルと区切り文字のすべてを定義するために BARSETUP プログラムを 実行することができます。
- 2. ユーティリティライブラリ

MR550 用に二つのライブラリ "Scankey3.dll" と "Scanner3.dll"があります。

A. Scanner3.dll: この DLL はデコーダチップからスキャンしたデータを得るための関数コールを 提供します。(関数の詳細な説明はセクション 3.2 をご覧下さい) B. Scankey3.dll: この DLL は入力されたスキャンデータを標準のキーボードバッファに入力する ための関数コールを提供します。(関数の詳細な説明についてはセクション 3.3 をご覧下さ い。)

3.2 "Scanner3.dll"の関数コールリスト

"Scanner3.lib"と"scanner3.h"はアプリケーションをコンパイルする VC プログラマに必要なファイルです。これらのファイルは SDK をインストール後、標準の LIB と INCLUDE フォルダ にあります。

3.2.1 デコーダを使用可能にする

- **関数の説明:** この関数は COM2 ポートをオープンし、デコーダチップからのバーコード入力を得るためのスレッドを作成し、そしてシステムバッファに入力データを保存します。 アプリケーションはシステムバッファから入力データを得るために関数コール PT GetBarcode()を使用することができます。
- 関数コール: INT PT_EnableBarcode(VOID);
- **戻りコード**: =1 新しいスレッドの作成に失敗
 - =2 使用可能に出来ない
 - =3 COM2を開くことが出来ない
 - =4 Hamster からの パラメータ アップロード失敗
 - =0 OK

3.2.2 デコーダを使用不可能にする

関数の説明: この関数は COM2 ポートを閉じ、PT_EnableBarcode() によって作成されたスレッドを消去します。

関数コール: VOID PT_DisableBarcode(VOID);

3.2.3 バーコード入力チェック

関数の説明:この関数はデコーダチップによって正しくデコードされたバーコードデータがシス テムバッファにあるかどうかをチェックするために使用します。

- 関数コール: BOOL PT_CheckBarcode(VOID);
- **戻りコード:**=TRUE システムバッファに入力データがある。 =FALSE システムバッファにデータがない。
- 3.2.4 バーコードデータを読む
 - 関数の説明:システムバッファから入力したバーコードデータとそのタイプを得る。
 - **関数コール:** BOOL PT_GetBarcode(TCHAR *szBarcodeBuffer,TCHAR *cType);
 - パラメータ:(出力)
 - szBarcodeBuffer : 入力データを保存するためのストリングバッファ

cType : 入力データのタイプ

=00H	Full Code 39
=01H	STD Code 39
=02H	EAN-13
=03H	UPC-A
=04H	EAN-8
=05H	UPC-E
=06H	l-25
=07H	CODABAR
=08H	Code 128
=09H	Code 93
=0Ah	STD 25

=0BH MSI =0CH EAN-128 =0DH Code 32 =0EH DELTA =0FH LABEL =10H PLESSEY =11H Code 11 =12H TOSHIBA **戻りコード:** TRUE = バーコード入力がある FALSE = バーコード入力がない

3.2.5磁気ストライプリーダ(MSR) 入力のチェック

関数の説明: この関数はデコーダチップによって正しくデコードされた MSR データがシステムバ ッファにあるかどうかをチェックするために使用されます。

- 関数コール: BOOL PT_CheckMSR()
- **戻りコード:** TRUE = MSR 入力がある FALSE = MSR 入力がない

3.2.6磁気ストライプリーダ(MSR) データを読む

関数の説明:磁気ストライプリーダ (MSR) 入力を読む。 関数コール: BOOL PT GetMSR(TCHAR *szTrack1,TCHAR *szTrack2,TCHAR *szTrack3, TCHAR *cType) パラメータ:(出力) szTrack1: トラック1からの入力を保存するためのストリングバッファ szTrack2: トラック2からの入力を保存するためのストリングバッファ szTrack3: トラック3からの入力を保存するためのストリングバッファ cType : MSR 入力データのインジケータ 1 トラック1のみ 2 トラック2のみ トラック3のみ 3 4 トラック1 とトラック2 トラック2 とトラック3 5 6 トラック1、トラック2とトラック3 **戻りコード:** TRUE = MSR 入力がある FALSE = MSR 入力がない

3.2.7 デコーダチップにバーコードシンボルをセット

関数の説明: この関数は個々のバーコードシンボルを設定するために使用されます。 **関数コール:** BOOL PT_SetBarcodePara(unsigned char type, unsigned char status); **戻り:** TRUE --- 設定 OK. FALSE - 設定失敗

パラメータ:(入力)

シンボル	タイプ	ステータス	動作
	0x39	Bit0 = 0	使用不可
		Bit0 = 1	使用可能
		Bit1 = 0	全 Code39
		Bit1 = 1	標準 Code39
		Bit3&2 = 0	チェックデジットを計算して送信
Code39 全設定		Bit3&2 = 1	チェックデジットを計算するが送
			信しない
		Bit3&2 = 2	チェックデジットを計算しない
		Bit4 = 0	スタートとストップビットを送信
		Bit4 = 1	スタートとストップビットを送信
			しない
Code39	0x3A	0	使用不可
		1	使用可能
Code39 全 ASCII	0x3B	0	使用可能
		1	使用不可
	0x3C	0	チェックデジットを計算して送信
Code39 チェックデジット		1	チェックデジットを計算するが送
	0/100		信しない
		2	チェックデジットを計算しない
	0x3D	0	スタートとストップビットを送信
Code39 SS		1	スタートとストップビットを送信
			しない
Code39 最少長	0x3E	0 48	標準値 = 0
Code39 最大長	0x3F	0 <i>48</i>	標準値 = 48

シンボル	タイプ	ステータス	動作
		Bit0 = 0	使用不可
		Bit0 = 1	使用可能
		Bit1 = 0	固定長可
		Bit1 = 1	固定長使用不可
		Bit2&3 = 0	チェックデジットを計算して送信
125	0x40	Bit2&3 = 1	チェックデジットを計算するが送信
			しない
		Bit2&3 = 2	チェックデジットを計算しない
		Bit4&5 = 0	最初の桁をサプレス
		Bit4&5 = 1	最後の桁をサプレス
		Bit4&5 = 2	サプレスしない
105	0.44	0	使用不可
125	0X41	1	使用可能
	010	0	固定長可
125 回正長	0X4Z	1	固定長使用不可
		0	チェックデジットを計算して送信
	0.42	1	チェックデジットを計算するが送信
125 CD	0X43		しない
		2	チェックデジットを計算しない
		0	最初の桁をサプレス
125 SS	0x44	1	最後の桁をサプレス
		2	サプレスしない
125 最少長	0x45	2 64	標準値 = 10
125 最大長	0x46	2 64	標準値 = 64
		Bit0 = 0	使用不可
		Bit0 = 1	使用可能
		Bit1 = 0	固定長可
S25&Toshiba	0x47	Bit1 = 1	固定長使用不可
02001031100	0,41	Bit2&3 = 0	チェックデジットを計算して送信
		Bit2&3 = 1	チェックデジットを計算するが送信
			しない
		Bit2&3 = 2	チェックデジットを計算しない
\$258Tochiba	0v48	0	使用不可
525010511100	0840	1	使用可能
S25&Toshiba 国宁長	0×40	0	固定長可
5250TOSITIDa 固定丧	0,49	1	固定長使用不可
		0	チェックデジットを計算して送信
S258Tochiba CD	0x44	1	チェックデジットを計算するが送信
UZJUTUJITIJA UD	0x4A		しない
		2	チェックデジットを計算しない
S25&Toshiba 最少長	0x4B	1 48	標準値 = 4
S25&Toshiba 最大長	0x4C	1 48	標準値 = 48

シンボル	タイプ	ステータス	動作
		Bit0 = 0	使用不可
		Bit0 = 1	使用可能
Code32	0v4d	Bit1 = 0	先頭文字を送信
000632	0840	Bit1 = 1	先頭文字を送信しない
		Bit2 = 0	末尾文字を送信
		Bit2 = 1	末尾文字を送信しない
Code32	0.40	0	使用不可
COUESZ	0846	1	使用可能
Code32 失雨文字详信	0v4f	0	先頭文字を送信
	0741	1	先頭文字を送信しない
Code32 末尾文字送信		0	末尾文字を送信
	0,00	1	末尾文字を送信しない
T . I		Bi t0=0	使用不可
rerpen	0.451	Bit0=1	使用可能
	0231	Bit1=0	文字セット = 標準
		Bit1=1	文字セット = 数字
Talaan	0x52	0	使用不可
Тетреп		1	使用可能
Talnan 文字セット	0x53	0	標準
		1	数字
FAN 128/UCC	0x54	0	使用不可
E/11 120/000		1	使用可能
FAN 128/UCC	0x55	0	使用可能
2/11 120/000	0,00	1	使用不可
EAN 128/UCC Code ID	0x56	0	使用不可
		1	使用可能
EAN 128/UCC ダブルラベル のセパレータ	0x57	ASCII	ASCII 文字
0a da . 400	0	0	使用不可
Code 128	0x58	1	使用可能
Code 128 最少長	0x59	0 64	標準値 = 1
Code 128 最大長	0x5a	0 64	標準値 = 64

シンボル	タイプ	ステータス	動作
		<i>bit0=0</i>	使用不可
		bit0=1	使用可能
		bit1=0	チェックデジット送信
		bit1=1	チェックデジット送信しない
MSI/Plessey	0x5B	bit3&2=0	チェックデジット・ダブルモジュー
			JV 10
		bit3&2=1	チェックデジット・モジュール 11
			プラス 10
		bit3&2=2	チェックデジット・モジュール 10
MSI/Plessev	0x5C	0	使用不可
		1	使用可能
MSI/Plessev CD	0x5D	0	チェックデジット送信
	0.002	1	チェックデジット送信しない
		0	チェックデジット・ダブルモジュー
			JL 10
MSI/Plessey CDMODE	0x5E	1	チェックデジット・モジュール 11
	_	2	チェックデジット・モジュール 10
MSI/Plessey 最少長	0x5F	1 64	標準値 = 1
MSI/Plessey 最大長	0x60	1 64	標準値 = 64
Code93	0x61	0	使用不可
		1	使用可能
Code93 最少長	0x62	1 48	標準値 = 1
Code93 最大長	0x63	1 48	標準値 = 48
		bit0=0	使用不可
		bit0=1	使用可能
Code11	0x64	bit1=0	1 チェックデジット
	0,01	bit1=1	2 チェックデジット
		bit2=0	チェックデジット送信
		bit2=1	チェックデジット送信しない
Code11	0x65	0	使用不可
Codell	0,00	1	使用可能
Code11 チェックデジット数	0x66	0	1 チェックデジット
	0,00	1	2 チェックデジット
Codell CD	0x67	0	チェックデジット送信
		1	チェックデジット送信しない
Code11 最少長	0x68	1 48	標準値 = 1
Code11 最大長	0x69	1 48	標準値 = 48

シンボル	タイプ	ステータス	動作
		Bit0 = 0	使用不可
		Bit0 = 1	使用可能
		Bit1 = 0	スタートとストップ文字を送信
		Bit1 = 1	スタートとストップ文字を送信しな
			61
CodaBar	0x6a	Bit2&3 = 0	チェックデジットを計算して送信
		Bit2&3 = 1	チェックデジットを計算するが送信
		Bit2&3 = 2	チェックテジットを計算しない
		Bit4 = 0	
		B/t4 = 1	
CodaBar	0x6b	0	
		1	
CodaBar SS	0x6c	0	
COURDAT 55	0,000	1	スタートとストツノ乂子を送信しな
		0	チェックデジットを計算して送信
		0	チェックデジットを計算するが送信
CodaBar CD	0x6d		しない
		2	チェックデジットを計算しない
	- 0x6e	0	オン
		1	オフ
Codabar 最少長	0x6f	3 48	標準値 = 3
Codabar 最大長	0x70	3 48	標準値 = 48
		Bit0 = 0	使用不可
Label Code	0x71	Bit0 = 1	使用可能
		Bit1 = 0	チェックデジット送信
		Bit1 = 1	チェックデジット送信しない
Label Code	0v72	0	使用不可
	0/112	1	使用可能
Label Code CD	0x73	0	チェックデジット送信
	0/11/0	1	チェックデジット送信しない
		bit0=0	使用不可
		bit0=1	使用可能
UPC-A	0x74	<i>bit1=0</i>	先頭桁を送信
		bit1=1	先頭桁を送信しない
		bit2=0	チェックデジットを送信
		bit2=1	<u>チェックデジットを送信しない</u>
UPC-A	0x75	0	使用不可
		1	使用可能
UPC-A LD	0x76	0	
		1	先頭桁を迭信しない
UPC-A CD	0x77	0	<i>チェックデジットを送信</i>
		1	ナエックテンットを送信しない

シンボル	タイプ	ステータス	動作
		bit0=0	使用不可
		bit0=1	使用可能
		bit1=0	先頭桁を送信
		bit1=1	先頭桁を送信しない
	0v78	bit2=0	チェックデジット送信
	0270	bit2=1	チェックデジット送信しない
		bit3=0	ゼロ表示オン
		bit3=1	ゼロ表示オフ
		bit4=0	NSC 使用可能
		bit4=1	NSC 使用不可
	∩v79	0	使用不可
	0713	1	使用可能
	0v7a	0	先頭桁を送信
	0774	1	先頭桁を送信しない
	0v7h	0	チェックデジット送信
	0710	1	チェックデジット送信しない
UPC_F 拉碑	0x7c	0	ゼロ表示オン
		1	ゼロ表示オフ
HPC_F NSC	0v7d	0	使用不可
	UX/U	1	使用可能
		bit0=0	使用不可
		bit0=1	使用可能
		bit1=0	先頭桁を送信
FΔN13	0x7e	bit1=1	先頭桁を送信しない
LANIS	UXIE	bit2=0	チェックデジット送信
		bit2=1	チェックデジット送信しない
		Bit3=0	ブックランド有効
		Bi t3=1	ブックランド無効
EVN13	0v7E	0	使用不可
EAN 15		1	使用可能
	0×80	0	先頭桁を送信
	0880	1	先頭桁を送信しない
FAN13 チェックデジット	0,01	0	チェックデジット送信
		1	送信しない
EANI12 ブックランド	0.48.2	0	使用可能
	0702	1	使用不可

シンボル	タイプ	ステータス	動作
		Bit0=0	使用不可
		Bit0=1	使用可能
	0.292	Bit1=0	先頭桁を送信
EANO	0x03	Bit1=1	先頭桁を送信しない
		Bit2=0	チェックデジット送信
		Bit2=1	チェックデジット送信しない
	0.01	0	使用不可
EANO	0x84	1	使用可能
	0,495	0	先頭桁を送信
EANO LU	COXO	1	先頭桁を送信しない
	0,496	0	チェックデジット送信
EANO UD	0000	1	チェックデジット送信しない
		bit0=0	Supplement 2 使用不可
		bit0=1	Supplement 2 使用可能
Supplement 2 / 5	0x87	Bit1=0	Supplement 5 使用不可
	0,01	Bit1=1	Supplement 5 使用可能
		Bi t2=0	あれば送信
		Bit2=1	必ず送信
Supplement 2	0x88	Bi t0=0	使用不可
		Bit0=1	使用可能
Supplement 5	0x89	BitO=0	使用不可
		Bit0=1	使用可能
Supplement MH	0x8A	Bit0=0	あれば送信
		Bit0=1	必ず送信
Supplement SSI	0x8B	Bit0=0	スペースセパレータ挿入
		Bit0=1	挿入しない
		Bi t0=0	使用不可
		Bit0=1	使用可能
Delta Code	0x8C	Bit1=0	チェックデジット計算
		Bit1=1	チェックテシット計算しない
		Bit2=0	ナエックテンット达信
		BI [2=1	テェックテジット这信しない
Delta Code	0x8D	BITU=U Dit0_1	使用不可
		Bit0-0	伏田門能 エーック<i>ニ</i>ジット社会
Delta Code CDC	0x8E	<i>B110=0</i> Bit0=1	<i>テェックテンツド計算</i> チェックデジットを計算したい
		Bit0_0	チェックデジットを详定
Delta Code CDD	0x8F		
		BITU=1	テェックテシットを送信しない
		0	
		1	
		2	
M2K	6	3	<u> </u>
		4	
		5	トラック2のみ
		6	トラック 3 のみ

3.2.8 デコーダチップからバーコートシンボル設定を得る

関数の説明: この関数はデコーダチップから個々のバーコードシンボルを得るために使用されます。 **関数コール:** BOOL PT_GetBarcodePara(unsigned char type, int *status); 入力/出力:セクション 3.2.7の表をご覧下さい。 戻り: TRUE --- OK. FALSE - 失敗

3.2.9 DLL バージョン番号を得る

関数の説明: この関数は DLL のバージョン番号を得るために使用されます。 **関数コール:** INT PT_DI I Version(void); **戻り:** Integer

3.2.10 すべてのシンボルを標準値にリセット

関数の説明: この関数コールはデコーダチップのシンボル設定を標準値にリセットします。 VC 用関数コール: int PT_ SetToDefault (VOID) VB 用関数コール: PT_ SetToDefault

3.3 "Scankey3.dll"の関数コールリスト

ユニテック社では以下の三つのファイルを提供しています。

"Scankey3.lib"	VC プログラミング用
"Scankey3.h"	VC プログラミング用
"Scankey3.bas"	VB プログラミング用

3.3.1 デコーダを有効にする

- **関数の説明:** この関数は COM2 ポートを開き、デコーダチップからバーコード入力を得るためにスレッドを作成し、そしてスキャンしたデータをキーボードバッファに送ります。ユーザアプリケーションは標準のキーボード入力としてデータを得ることが出来ます。
- VC 用関数コール: int PT_EnableBarToKey(VOID)
- VB用関数コール: PT_EnableBarToKey() as Long
- **戻りコード:**=1 新しいスレッド作成に失敗
 - =2 再度有効にできない
 - =3 COM2 を開けない
 - =4 Hamster からパラメータのアップロード失敗
 - =0 OK

3.3.2 デコーダを無効にする

関数の説明: この関数は COM2 ポートを閉じ、PT_EnableBarToKey()で作成されたスレッドを消去 します。 VC 用関数コール: VOID PT_DisableBarToKey (VOID)

VB 用関数コール: PT DisableBarToKey ()

3.3.3 バーコードシンボルをデコーダチップにセット

関数の説明:この関数は個々のバーコードシンボルをセットするために使用されます。

VC 用関数コール: BOOL PT_SetBarToKeyPara(unsigned char , unsigned char)

VB 用関数コール: PT_SetBarToKeyPara (ByVal strType As String, ByVal strStatus As String) As Boolean

戻り: TRUE --- 設定 OK. FALSE - 設定失敗

パラメータ:(入力) セクション 3.2.7と同じ。

3.3.4 デコーダチップからバーコードシンボル設定を得る

関数の説明: この関数はデコーダチップから個々のバーコードシンボルを得るために使用されます。 VC用関数コール: BOOL PT_GetBarToKeyPara(unsigned char cType,int* iStatus) VB用関数コール: VB用の関数コールはありません。 入力/出力: セクション 3.2.7の表をご覧下さい。 戻り: TRUE --- OK. FALSE - 失敗

3.3.5 DLL バージョン番号を得る

関数の説明:この関数は DLL バージョン番号を得るために使用されます。

VC 用関数コール: INT PT_Version(void); VB 用関数コール: PT_Version As Long 戻り: Integer

3.3.6 プリアンブル文字列をセット

関数の説明: この関数は このファンクションは、"Scankey3.dll"がキーボードバッファにスキャンしたデータの前にプリアンブル文字列を送らせます。標準のプリアンブル文字列はNULLで、その最大長は48文字です。標準のプリアンブル文字列NULLで、最大長は48文字です。

VC 用関数コール: void PT_SetPreamble(TCHAR *sPreambleStr)

VB 用関数コール: void PT_SetPreamble(ByVal strPreamble As String)

入力 : プリアンブル文字列のストリングバッファ

戻り: なし

3.3.7 プリアンブル文字列を得る

関数の説明: 現在のプリアンブル文字列を得る VC 用関数コール: void PT_GetPreamble(TCHAR * sPreambleStr) VB 用関数コール: No 関数コール available for VB 出力: プリアンブル文字列のストリングバッファ 戻り: なし

3.3.8 ポストアンブル文字列をセット

関数の説明: この関数は、 "Scankey3.dll"がキーボードバッファにスキャンしたデータの後にポ ストアンブル文字列を送らせます。標準のポストアンブル文字列は NULL で、その最 大長は 48 文字です。 VC 用関数コール: void PT_SetPostamble(TCHAR *sPostambleStr)

VB用関数コール: void PT_SetPostamble(ByVal strPostamble As String)

入力 : ボスとアンブル文字列のストリングバッファ

戻り:なし

3.3.9 ポストアンブル文字列を得る

関数の説明: 現在のポストアンブル文字列を得る VC 用関数コール: void PT_GetPostamble(TCHAR * sPostambleStr) VB 用関数コール: VB 用の関数コールはありません。 出力: ポストアンブル文字列のストリングバッファ 戻り: なし

3.3.10 区切り文字のセット

関数の説明: このファンクションは、"Scankey3.dll"がキーボードバッファにポストアンブルの 後に区切り文字を追加させます。標準の区切り文字はNULLで、その最大長は10文字です。 VC 用関数コール: void PT_SetDelimiter(TCHAR *sDelimiterStr) VB 用関数コール: PT_SetPostamble (ByVal strPostamble As String) 入力 : 区切り文字列のストリングバッファ 戻り: なし

3.3.11 区切り文字列を得る

関数の説明:現在の区切り文字列を得る

VC 用関数コール: void WINAPI PT_GetDelimiter(TCHAR *sDelimiterStr) VB 用関数コール: VB 用の関数コールはありません 出力: 区切り文字列のストリングバッファ 戻り: なし

3.3.12 すべてのシンボルを標準値にセット

- **関数の説明:** この関数コールはデコーダチップのシンボル設定をシステムの標準値にリセットします。
- VC 用関数コール: int PT_ SetToDefault (VOID) VB 用関数コール: PT_ SetToDefault

4 MR550 プログラミングのための便利なライブラリ

MR550 はリレー、フォトカップル入力、セキュリティアラーム等の I/O デバイスをコントロールす るための GIOAPI.DLL も提供しています。

4.1.1 リレー1 コントロール

関数の説明: この関数はリレーポート 1 のステータスをコントロールするために使用されます。
関数コール: DWORD GIO_Relay1 (BOOL status)
パラメータ: (入力)
status: TRUE = リレー 1オープン
FALSE = リレー 1クローズ
戻りコード: なし

4.1.2 リレー2コントロール

関数の説明: この関数はリレーポート 2 のステータスをコントロールするために使用されます。
関数コール: DWORD GIO_Relay2 (BOOL status)
パラメータ: (入力)
status: TRUE = リレー 2オープン
FALSE = リレー2クローズ
戻りコード: なし

4.1.3 リレー1のステータスを得る

関数の説明: この関数はリレーポート1のステータスを得るために使用されます。 **関数コール:** DWORD GIO_Relay1Status () **パラメータ: なし 戻りコード:** 0 = リレー 1 は開かれている 1 = リレー 1 は閉じられている

4.1.4 リレー2のステータスを得る

関数の説明: この関数はリレーポート2のステータスを得るために使用されます。 **関数コール**: DWORD GIO_Relay2Status () パラメータ: なし 戻りコード: 0 = リレー2は開かれている 1 = リレー2は閉じている

4.1.5 デジタル入力を得る

関数の説明: この関数はデジタル入力(フォトカップル)のステータスを得るために使用され ます。 **関数コール: DWORD GIO_Digit() パラメータ: なし 戻りコード:** 0 = low 1 = high

4.1.6 セキュリティアラームを有効にする

関数の説明: この関数はセキュリティアラームを有効なするために使用されます。従って、 MR550 はバックプレートが開けられたら音を発します。

関数コール: DWORD GIO_EnableAlarm()

パラメータ: なし

戻りコード: なし

4.1.7 セキュリティアラームを無効にする

関数の説明:この関数はセキュリティアラームを無効にするために使用されます。従って、 MR550 はバックプレートが開けられても音を出しません。

- 関数コール: DWORD GIO_DisableAlarm()
- パラメータ: なし
- 戻りコード: なし

5 Visual C/C++ with SDK または MFC によるプログラミング

5.1 システムの必要条件:

HW	CPU	Pentium 90-MHz 以上のプロセッサを推奨。					
	メモリ	Windows NT Workstation 4.0, Windows 2000 - 128MB					
		Windows 95/98 - 64MB					
	CDROM ドライブ	マルチメディア・ディスクトップ・コンピュータ仕様に対応しているも					
		Ø					
	ディスプレイ	VGA 以上の解像度が必要。Super VGA モニタを推奨					
	マウス	マウスまたは相当品。					
SW	OS	Microsoft® Windows NT/2000/XP ® Workstation 4.0, Microsoft®					
		Windows® 95/98/Me					
	開発ツール	Microsoft eMbedded Visual Tools および MR550 SDK for VC/VB					

注意: 開発環境には Microsoft Windows 2000, XP, NT Workstation 4.0 をデバッグ用のホストとし て推奨いたします。ツールキットは Windows 95/98/Me にインストールすることができ、これらからアプ リケーションを構築することができます。しかし、エミュレーションは Windows 95/98/Me では動作しま せん; 代わりにホスト機として Windows NT,2000,XP を使用して下さい。

5.2 インストール

ユーザはアプリケーション・プログラムの開発に Microsoft eMbedded Visual Tools (EVT) と MR550 SDK for VC++ を使用して下さい。

Microsoft® eMbedded Visual C++® 3.0 はデスクトップ Visual C++アプリケーション開発で使用するの と同様な統合開発環境(IDE)を使用して Windows CE ベースのアプリケーションを開発することができま す。しかし、この IDE はアプリケーションを作成、テスト、および改良するために使用する多数の標準 開発ツールの Windows CE 専用のバージョンを含んでいます。これは Windows CE プラットフォームとデ バイスに独特な新しいソフトウェアを開発するための各種ツールも含んでいます。

ユーザ独自の Windows CE アプリケーションを開発するために、eMbedded Visual C++ IDE は学習が容易 で、他の Visual C++ファミリの経験のあるプログラマはすぐに慣れることができます。アプリケーショ ンは Handheld PC Pro (H/PC Pro)、 Palm-size PC 1.2、 Pocket PC そして MR550 プラットフォーム 上で実行するために eMbedded Visual C++ で作成することができます。独自の Windows CE ベースのプ ラットフォーム、あるいは Windows CE ベースのプラットフォームをシミュレートするデスクトップ・エ ミュレータの中で実行するアプリケーションを作成するために eMbedded Visual C++ を使用することも できます。

Microsoft eMbedded Visual Tools のインストールが終わった後で、ダイアローグボックスは H/PC と Palm-size PC の SDK をインストールするための入力要求を出します。 この時点でこれらの SDK をイン ストールする場合は、"はい"を選択して下さい。

これは Microsoft eMbedded Visual Tools のセットアップ・プログラムから立ち上がり、その結果 H/PC、 Palm-size PC と Pocket PC 用の Windows CE SDK のセットアップが立ち上がります。 三つの設 定プログラムのすべてを正しいインストール作業を行うために行わなくてはなりません。Windows CE 用 の MFC を使用したい場合は、これらの SDK をインストールしなければなりません。

さらに、ユニテック社は当社の MR550 プラットフォーム用の SDK も提供しています。ユニテックの提供 する CD-ROM のフォルダ ¥SDK¥VC からインストール・プログラムを実行することができます。ユニテッ クの SDK は Palm-Size SDK と特別なスキャナ・ライブラリ を含んでおり、これは"scanner3.lib", "scanner3.h", "scankey3.lib" と "scankey3.h" を皆様の開発用のフォルダにコピーする必要がないこ とを意味しています。言い換えると、これらは標準の ¥lib と ¥include フォルダに置かれます。

注意: Windows CE Toolkit for Visual C++ が正しく機能するために少なくとも一つの Windows CE SDK をインストールしなければなりません。

より詳しい情報については、eMbedded Visual C++のリリース文書をご覧下さい。

5.3 VC++で Windows CE プログラムを作成する方法

アプリケーションを作成するために以下のステップに従って下さい:

1. eMbedded Visual C++を実行します。

2. メインメニューから "ファィル / 新規作成"を選択します。

😥 Microsoft eMbedded Visual C++		
ファイル(E) 編集(E) 表示(⊻) 挿2	<φ プロジェクト(P) ビルド(B) ツール(T) ウィンドウ(W) ヘルプ(H)	
■ 新規作成(N) Ctrl+N ■ 聞(Q) Ctrl+O 閉じる(Q)		
ページ レイアウトの設定(1) ④ 印刷(2) OrrI+P 最近使ったファイル(2) ↓ 最近使ったワークスペース(2) ↓ アブリケーションの終了(2)		
 ビルド (デバッグ): アプリケーションを終了し、変更したつ 	ファイルから検索1 入 ファイルから検索2 / 【【】 マイルの保存を確認します	

ダイアローグボックスが AP タイプ選択のためにポップアップします。Windows CE アプリケーションに ついては、選択可能ないくつかのプロジェクトタイプがあります。ユーザは "WCE application"を選 択することによって SDK プログラムを作成するか、 "WCE MFC AppWizard (exe)"を選択することによ って MFC プログラムを作成、もしくは 他の7オプションのいずれかを選択します。"プロジェクト名" にこのプロジェクトの適当な名前を入力して、 "CPU"の下で"Win32 (WCE MIPS)"を選択します。 こ こで次のステップにすすむために "OK" ボタンをクリックします。

新規作成	?×
ファイル プロジェクト ワークスペース) ■WCE Application 影 WCE ATL COM AppWizard	 プロジェクト名(<u>N</u>): PT930 デモブログラム
WCE Dynamic-Link Library WCE MFC ActiveX ControlWizard WCE MFC AppWizard (dll) WCE MFC AppWizard (exe) WCE Pocket PC Application WCE Pocket PC MFC AppWizard (exe) WCE Static Library	位置(©): C¥Program Files¥Microsoft eMbedk … ● 新しいワークスペースを作成(®) ● 現在のワークスペースへ追加(A) ■ 依存関係(©):
	CPU(<u>P</u>): Win32 (WCE ARM) Win32 (WCE MIPS) Win32 (WCE MIPS16) Win32 (WCE MIPSFP)
	OK キャンセル

WCE Applicationを選択した場合、適当な項目をクリックあるいは選択して、"次へ"ボタンをクリックします。

WCE Application - ステップ 1/1		<u>?</u> ×
	作成する Windows アブリケーションの種類を選択し てください。 ○ 空のブロジェクト(E) ● 単純な Windows CE アブリケーション(S) ● 標準的な "Hello World!" アブリケーション(W)	
 〈 戻る(<u>B</u>)	次へ (11)> 終了(5) キャンセル	

WCE MFC AppWizard(exe)を選択した場合、適当な項目をクリックあるいは選択して、"次へ"ボタンを クリックします。そして、すべての必要な MFC DLL は実行ファイルに組み込まれます。

WCE MFC AppWizard (exe) - ステップ 1/4		? ×
WCE MFC AppWizard (exe) - ステップ 1/4	作成するアブリケーションの種類 ● <u>SDIS</u> ● ダイアログ ベース(型) ■ ドキュメント/ビュー アーキテクチャのサポート(型) リソースで使用する言語の指定(L) ■本語 (ceappwzjpndl)	?×
〈 戻る(B)	次へ (N)> 終了(E) キャンセル	

WCE MFC AppWizard (exe) - ステップ 3/4		?×
File Edit Yiew Jasert Bwild Help Project Project	ソース ファイルのコメントを生成しますか?	
< 戻る(<u>B</u>) (次へ (N)> 終了(F) キャンセル	

"終了"ボタンを押し、そしてソースコードを生成するために"OK"ボタンを押します。

WCE MFC AppWizard (exe) - ステップ 4.	/4 AppWizard で作を応され CPT930App CMainFrame CPT930Doc CPT930View	る新規アプリケーション クラス(<u>C</u>):	?×
\	クラス名(L): CPT930App	ヘッダー ファイル: PT930 デモプログラムA	
	基本クラス: CWinApp	インプリメント ファイル : PT930 デモブログラムA	
< 戻る(B) 次へ (N)>	終了(E) キャンセル	

プロジェクトをビルドする前に、ターゲットプラットフォームとして"MR550 " または "Palm-size PC 2.11 " を選択し、そしてアクティブ構成として "Win32 (CE MIPS) Release "を選択して下さい。

😥 PT930 デモプログラム - Microsoft eMbedded Visual C++	<u>- 0 ×</u>
」ファイル(E) 編集(E) 表示(V) 挿入(Q) プロジェクト(P) ビルド(B) ツール(T) ウィンドウ(W) ヘルブ(H)	
CAboutDlg ・ (すべてのクラス メンバ) ・ CAboutDlg ・ ベ・	
PT930 デモブロ・PT930 JPN ・Win32 (WCE MIPS) Release ・PT930 JPN (Default Device)	💽 🖉 🛗 🗄
Palm size PC 211	
× ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	

これは皆様のアプリケーションプログラムに必要なソースコードを生成し、そして実行可能なプログラムファ イルを生成するためにこのプロジェクトをビルドします。

このプログラムを MR550 に送信するために"ActiveSync"を使用することができます。.

6 Visual Basic によるプログラミング

6.1 インストール

VB プログラミングについては、Microsoft eMbedded Visual Basic と MR550 SDK for VB をインストールしなければなりません。

Microsoft eMbedded Visual Basic は Visual Basic 6.0 と CE toolkit for Visual Basic が組み合わ されていますが、Windows CE プログラミングのみ使用可能です。

注意: 本リリースに含まれている H/PC SDK を使用するために、VBCE 6.0 インストールの一部として これをインストールしなければなりません (完全またはカスタムのいずれか)。あるいは、これを最初に インストールして、そして VBCD 6.0 をインストールします。

より詳しい情報については、WinCE Toolkit for Visual Basic 6.0のオンラインヘルプをご覧下さい。

6.2 VB で Windows CE プログラムを作成する方法

Visual Basic のプログラムについては、VB プログラムを MR550 にインストールする時にインストー ル・プログラムと ActiveSync が必要になります。

1. Visual Basic 6.0を実行し、そして新しいプロジェクトを作成するために "ファイル/新しいプロジェクト "を選択します。

新しいプロジェクト	? ×
Microsoft Contraction All Rights Reserved.	0
新規作成 既存のファイル 最近使ったファイル	
フォームのな Windows CE windows ce windows ce Windows CE い windows c PsPC フロン palm-size p for the pock PT900 Projec	t
	開(@) キャンセル ヘルブ(H)
□ 次回からこのダイアログ ボックスを表示しない(型)	

2. "フォームのない Windows CE プロジェクト"のアイコンをダブルクリックし、ターゲット・プラットフォームとして" MR550"を選択して下さい。

プラットフォームの選択
このプロジェクトは、正しい Windows CE ブラットフォームをターゲットにしていま せん。使用可能なブラットフォームを1 つ選択してください:
PT930JPN Palm-size PC 2.11 Pocket PC PT900
ОК Теруси

3. そしてアプリケーションに従って皆様の Visual Basic プログラムを編集して下さい。

🞆 Project1 - Microsoft eMbed	ded Visual Basic [デザイン]				<u>- 🗆 ×</u>
ファイル(E) 編集(E) 表示(⊻)	プロジェクト(P) 書式(Q) デバッグ(D) 実行(<u>R</u>) ツー	ール(工) ウィンドウ(₩ ヘルプ(田)	
🏷 - 🖻 🖬 👗 🖻	🖹 🏘 🗠 斗 🍵 📕 PT930	IJPN (De 🗸 💩	i 🖻 🐕 🛠		>>
Noject1 -	Form1 (Form)		ブロジェクト - Proje	ect1	×
全般 ► 60 ► A 「abl ^(M) □ I ^(M) Com Com Com EII 111 ■ ^(M) Fran	L L X		■ SProject1 ■ フォー ■ フォー ■ Frame Frame1 Frame 全体 項目別 (ブラジェクト名) Appearance BackColor BorderStyle Caption ClipControls Enabled (ブラジェクト名) プログラムで使用 めの名前を返しま	(Project1) -ム orm1 (Form1) モジュール odMain (modMain)) × ・ ・ * *

4. Visual Basic プログラムの編集が終わった後でこのプロジェクトを保存し(これは "*.vbp" また は "*.ebp" としてディスクに保存します)、そして "*.vb" をディスクに保存するために "ファイ ル" から "プロジェクトの作成 "を選択します。

🔛 Project1 - M	Aicrosoft eMbedded Visual Basic [デザイン]	- 🗆 ×
ファイル(E) 編集	集任)表示(V) プロジェクト(P) 書式(Q) デバッグ(D) 実行(R) ツール(T) ウインドウ(W) ヘルプ(H)	
	 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	× ×

5. "ツール - リモート・ツール" から "アプリケーション・インストール・ウイザード"を選択し、 そし て "ステップ1" ダイアローグボックスの "次へ" ボタンをクリックして下さい。

eMbedded Visual Basic アプリケ	ーション インストール ウィザード - ステップ 1	x
	eMbedded Visual Basic アプリケーション インストール ウィザードを 使用すると、eMbedded Visual Basic アプリケーションのインストー ル プログラムを各ステップに従って作成できます。 開始するには Dたへ] をクリックしてください。	
	< 戻る(日) (ホス(1)) キャンセル	

6. "ステップ 2 " と "ステップ 3 "のダイアローグボックスで、ユーザはステップ 4 で生成される "*.ebp " と "*.vb "ファイルを表示しなければなりません。

eMbedded Visual Basic アプリカ	rーション インストール ウィザード - ステップ 2 🔀 🔀
	eMbedded Basic Project (EBP) ファイルへの完全パスを入力してく ださい。
	C¥tmp¥Pt930Demo.ebp
	< 戻る(B) 次へ(N) > キャンセル

eMbedded Visual Basic アプリ	ケーション インストール ウィザード - ステップ 3	×
	コンパイル済みの eMbedded Visual Basic アプリケーション ファイ ル (VB) への完全パスを入力してください。	
	C:¥tmp¥PT930Demo.vb	
	< 戻る(B) 次へ(N) > キャンセル	

"ステップ4" ダイアローグボックスで出力ファィルのディレクトリを表示します。そして"ステップ5"のダイアローグボックスでプロセッサとして "MIPS 3000" をクリックします。

eMbedded Visual Basic アプリク	ケーション インストール ウィザード - ステップ 4	×
	すべての出力ファイルが作成されるディレクトリ名を入力してくださ い。	
	C¥tmp	
	< 戻る(B) 次へ(N) > キャンセル	

eMbedded Visual Basic アプリケ	rーション インストール ウィザード - ステップ 5	×
	サポートするプロセッサを選択してください。	
Sø-	▼MIPS3000 (4K) v3.00	
A .		
	< 戻る(B) 次へ(N) > キャンセル	-

"ステップ 6"のダイアローグボックスで必要な ActiveX コントロールをクリックします。そして、 ステップ 7"のダイアローグボックスで MR550 に送るすべての必要なファイルを選択します。

eMbedded Visual Basic アプリケ	ーション インストール ウィザード - ステップ 6	×
	インストールする Active X コントロールまたはリファレンスを選択し てください。	
	Microsoft CE ADOX Control 3.1	
	✓Microsoft CE Comm Control 3.0	
	✓Microsoft CE CommandBar Control 3.0	
	✓Microsoft CE Common Dialog Control 3.0	
	Microsoft CE Data Access Control 3.1	
	Microsoft CE File Control 3.0	
	✔Microsoft CE File System Control 3.0 🗨	
	< 戻る(B) 次へ(N) > キャンセル	

eMbedded Visual Basic アプリケ	ーション インストール ウィザード - ステップ 7	×
	追加するデータ ファイルを指定してください	
	ファイル 種類	
	< 戻る(B) (次へ(ND)> キャンセル	

インストールディスクの適当な"ディレクトリ"と"ステップ8"ダイアローグボックスの要求に従った他の情報を入力します。そして"ステップ9"ダイアローグボックスで"インストール・プログラムの作成" ボタンをクリックします。

eMbedded Visual Basic アプリケ	ーション インストール ウィザード - ステップ 8	×
	次の情報を入力してください。 既定のインストール ディレクトリ (¥Program Files に作成されるフォ ルダ名)	
	PT930_Demo	
	アラリク ーション名 PT930 Demonstration	
	, 月1100日	
	(F130) モクロクラム 会社名	
	Unitech Japan KK	
		_
	〈戻る(B) 次へ(N) > キャンセル	

eMbedded Visual Basic アプリケ	ーション インストール ウィザード - ステップ 9	×
	CE アプリケーション インストール ウィザードの完了 インストール ブログラムを作成する準備ができました。 下のボタンを クリックすると開始します。	
	インストール プログラムの作成(<u>©</u>)	
	て戻る(日) 完了 キャンセル	

インストールディスクはディレクトリ "CD1 "に保存されました。 CD-ROM またはディスケットにコピー して下さい。

MR550 に VB プログラムをインストールしようとする PC で ActiveSync が実行していることを確認して、 "setup.exe"を実行して下さい。