

QRコードデコード ActiveX コントロール
QRdecoder
ユーザーズ・マニュアル
(Ver. 1.00.00)

株式会社デンソーウェーブ



目次

1. 概要	4
2. 機能	4
3. 実行環境	4
4. 開発環境・利用可能な言語	5
5. インストール/アンインストール	5
5.1. インストールの実行	5
5.2. インストールされるファイル	6
5.3. プロダクトキー及びレジストリの登録	7
5.4. アンインストール	8
6. プロパティ	10
6.1. カスタムプロパティ一覧	10
6.2. カスタムプロパティの機能説明	11
6.2.1. AreaNum プロパティ	12
6.2.2. BarProperty プロパティ	13
6.2.3. CodeKind プロパティ	14
6.2.4. DecDataV, DecDataプロパティ	15
6.2.5. DcdStrLinPos, DcdEndLinPos, DcdStrColPos, DcdEndColPosプロパティ	16
6.2.6. DigitizeMode プロパティ	17
6.2.7. DigiThreshold プロパティ	18
6.2.8. ErrorStatus プロパティ	19
6.2.9. InfDataV, InfDataプロパティ	21
6.2.10. LoadBMPFile プロパティ	22
6.2.11. LoadTIFFFile プロパティ	23
6.2.12. MemAddrV, MemAddrプロパティ	24
6.2.13. MemSizeX, MemSizeYプロパティ	25
6.2.14. MirrorMode プロパティ	26
6.2.15. ModSize プロパティ	27
6.2.16. MultiBar プロパティ	28
6.2.17. PageNoプロパティ	29
6.2.18. QRJoinMode プロパティ	30
6.2.19. RemoveNoise プロパティ	31
6.2.20. Resolution プロパティ	32
6.2.21. ReverseBW プロパティ	33
6.2.22. SelectArea プロパティ	34
6.2.23. SelectRGB プロパティ	35
6.2.24. Start プロパティ	36
6.2.25. Status プロパティ	37
6.3. 必須プロパティの設定順序	38
7. Visual C++での利用	39
7.1. 画像ファイルからのデコード(1エリア指定)	39



7.2. 画像ファイルからのデコード(複数エリア指定)	41
8. デコード情報テーブル	44
8.1. QRコードの場合	44
8.2. DataMatrixの場合	46
8.3. PDF417コードの場合	47
8.4. バーコードの場合	48
8.5. EAN.UCC Compositeの場合	50
9. 改訂履歴	51

1. 概要

QRコードを始めとする主要な二次元コード、および各種バーコードを画像ファイルもしくは256階調の画像イメージの中から探索し、デコードを行うための、ActiveXコントロール(OCXプログラム)でOLEカスタムコントロールとも呼ばれています。ファイル名はQRdecoder.ocxです。使用例の説明はVC++6.0を中心に記述してあります。

2. 機能

画像ファイルもしくは256階調の画像イメージの中から読み取りたい二次元コード・バーコードを探索しデコードすることができます。画像ファイル選択時は、同時に4つのエリアを指定してデコードする事が可能です。読み取り可能コードは以下の通りです。

二次元コード

- ・ QRコード(モデル 1, 2)
- ・ MicroQR
- ・ DataMatrix
- ・ PDF417
- ・ MicroPDF417
- ・ EAN.UCC Composite

バーコード

- ・ UPC/EAN
- ・ CODE39
- ・ CODABAR (NW-7)
- ・ Interleaved 2 of 5 (ITF)
- ・ CODE128/ EAN-128
- ・ RSS

画像ファイルは以下のフォーマットに対応しています。

ファイル形式	圧縮形式	モノクロ 2 値	256 色グレイ	256 色カラー	16bit カラー	24bit カラー
BMP	非圧縮	○	○	○	○	○
TIFF 注)	非圧縮	○	○	○	—	○
	G3	○	—	—	—	—
	G4	○	—	—	—	—
	LZW	○	—	○	—	○

○:対応 —:非対応

注)各フォーマットのマルチページ TIFF ファイルにも対応しています。

3. 実行環境

- OS: Windows 2000 Professional, Windows XP Professional/Home Edition
 CPU: PentiumⅢ 800MHz 以上を推奨
 メモリ: 上記OSが動作するメモリ
 ただし実行時最大で空き領域が約360Mバイト必要(デコード対象エリア数4, 各エリアの画像処理範囲指定なし, 画像ファイル8192×10240画素の場合)
 ディスク: 100Mバイト以上の空きエリアが必要



4. 開発環境・利用可能な言語

OS: Windows 2000 Professional, Windows XP Professional/Home Edition
言語: Visual Basic 6.0, Visual Basic.Net 2003
Visual C++ 6.0, Visual C++.Net 2003

5. インストール/アンインストール

5.1. インストールの実行

管理者権限をもったユーザがインストールする必要があります(ユーザ権限のみを所有するユーザは、インストールおよびプロダクトキー登録されていれば、ユーザ毎にプロダクトキーを登録することなく使用出来ます)。インストールに先立ち、他のアプリケーションソフトは全て終了してください。弊社ホームページより QRdecoder_Trial.zip をダウンロードして任意の場所に解凍して頂き、フォルダ内の Setup.exe を実行します。

後は、インストールウィザードに答えていくだけでインストールが完了し、プログラムフォルダに起動用アイコンが登録されます。

5.2. インストールされるファイル

インストールされる主なファイルは次の通りです。

ファイル名	内容説明	再配布
QRdecoder.ocx	ActiveX コントロールプログラム本体	不可
QRdecoder.lib	リンクライブラリ(C++での開発時に必要)	不可
QRdecoder.lic	ライセンスコントロールファイル(開発時に必要)	不可
QRdecoder.tlb	タイプライブラリ TLB ファイル(開発時に必要)	不可
QRdecoder_Users_Manual_J.pdf	ユーザーズガイド(本文書)	不可
QRDecOCX.h	デコード情報配列, プロパティ設定値, エラーステータスの定義	不可
UregDenso.exe	専用アンインストーラ	不可
DecOCXTes_Sample.exe	QRdecoder を使用した動作確認用ソフトウェア	不可

また, QRdecoder の使用例として下記サンプルファイルが用意されています。QRdecoder_Trial.zip を解凍して頂くと, Sample フォルダの下に言語ごとのサブフォルダが作成されます。なお, このサンプルファイルは動作確認していますが, パラメータの範囲チェックやエラー処理などは省略されていますので, 実際の業務に使用することはお避けください。QRdecoder を用いたプログラム作成の参考としてのみご利用ください。

サブフォルダ	ファイルの種類	機能
VC++6.0	C++ソース一式	Visual C++ 6.0 で作成されたサンプルプログラム (ソース, プロジェクトファイルなど)
VB6.0	VBソース一式	Visual Basic 6.0 で作成されたサンプルプログラム (ソース, プロジェクトファイルなど)
VC++.Net	C++ソース一式	Visual C++ .Net で作成されたサンプルプログラム (ソース, プロジェクトファイルなど)
VB.Net	VBソース一式	Visual Basic .Net で作成されたサンプルプログラム (ソース, プロジェクトファイルなど)



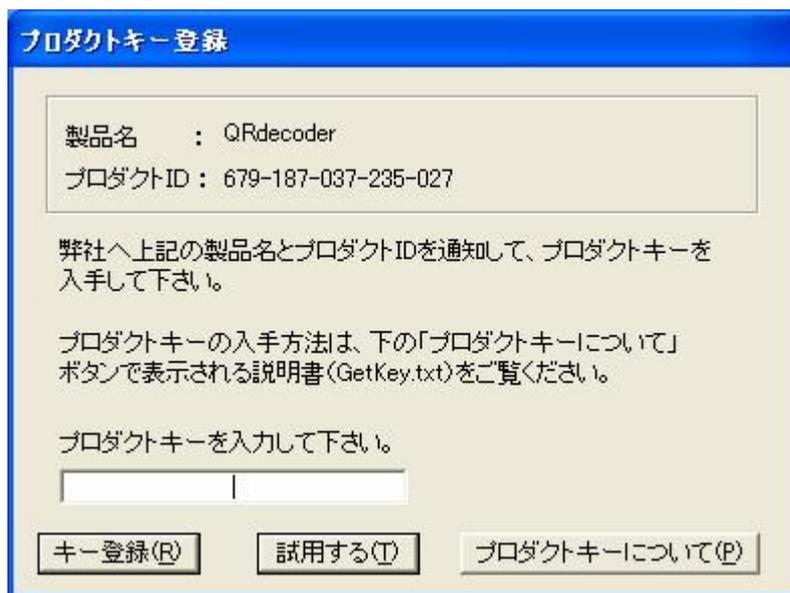
5.3. プロダクトキー及びレジストリの登録

インストール直後は 30 日間試用可能な試用版として動作しております。これを試用期間なしの製品版とするためにはプロダクトキーを入手し登録する必要があります。QRdecoder.ocx を組み込んで開発を行う場合もしくは組み込んだソフトを起動すると試用版では毎回、次のプロダクトキー未登録画面が表示されます。



ここで、[OK]ボタンを押すと、プロダクトキー登録画面が表示されます。製品版として使用するためには、プロダクトキーを以下の手順で入手し登録して下さい。

プロダクトキーの入手方法



1. プロダクト ID で表示されている数字列（5 個の 3 桁数字 上記例では 679-187-037-235-027）と製品パッケージのシリアル番号を弊社まで連絡いただきます。連絡方法は、「プロダクトキーについて」を押して参照してください。
2. その結果で発行されたプロダクトキーを入力し[キー登録]で登録完了となります。これで試用期間なしの製品版となります。なお図に表示されているプロダクト ID は一例であり、インストールされる PC 毎に異なる値となります。
3. 製品購入前に試用する場合は[試用する]を押してください。試用開始時から 30 日間試用可能です。



なお、OS のバージョンアップ、再インストール、ハードディスクの交換、フォーマットなどでは異なるプロダクトID となるため、同じプロダクトキーでは継続使用できなくなります。これら作業を行う前に、次項で説明する専用アンインストーラでアンインストールしてから、新たなプロダクトキーを入手し、再インストールしてください。

5.4. アンインストール

管理者権限をもったユーザがアンインストールする必要があります。登録されているアイコン「専用アンインストーラ」を選択してアンインストールを実行してください。アンインストーラで表示される「アンインストール情報」を取得します。別の PC などに再インストールするために必要な重要情報です。

[アンインストール]ボタンをクリックすることでアンインストールが始まります。ウィザードに答えて行きます。アンインストールウィザードの処理が終了すると、アンインストール情報が表示されます。



アンインストール情報 (DENSO WAVE製ソフト)

プログラム名 QRdecoder

登録名 Software#DENSO WAVE#QRdecoder

旧プロダクトID (重要) 654-096-090-254-983

アンインストールキー (重要) 032-763-141-001-826

終了

アンインストール

アンインストールが正常に終了しました。

表示された「アンインストール情報(重要)」は必ずメモしておいてください。別の PC などに再インストールするためには、新しいプロダクトキーを入手する必要があります。その際にこのアンインストール情報が必要となります。コントロールパネルの「アプリケーションの追加と削除」を使用して QRdecoder を削除することもできますが、アンインストール情報を取得できません。もし、このコントロールパネル機能で削除したり、直接インストールファイルを削除したりした場合は、もう一度 QRdecoder をインストールし直してから専用アンインストーラを起動して、「アンインストール情報」を取得してください。アンインストール情報の再確認だけであればプロダクトキーを入力する必要はありません。

なお、この専用アンインストーラでアンインストールを実行すると、それまで正常にインストールできていたプロダクトキーでは、同じPCであっても再インストールはできなくなりますので、不用意にアンインストールしないでください。間違えてアンインストールをしてしまった場合は、このアンインストール情報と共に、新たなプロダクトIDを準備していただき、弊社プロダクトキー取得ホームページ”<https://www.denso-wave.com/prokey/ja/>”でプロダクトキーを入手してください。

6. プロパティ

6.1. カスタムプロパティ一覧

名称	型	初期値 注 1)	R/W	エリア毎の設定	意味
AreaNum	long	1	R/W	×	デコード対象エリア数
BarProperty	long	0	R/W	○	バーコード詳細設定
CodeKind	long	CODE_QR(1)	R/W	○	コード種別
DecData	long	-	R	○	デコードデータ取得(VC++用)
DecDataV	Variant	-	R	○	デコードデータ取得(VB 用)
DcdEndColPos	long	8192	R/W	○	横方向デコード終了位置
DcdEndLinPos	long	10240	R/W	○	縦方向デコード終了位置
DcdStrColPos	long	0	R/W	○	横方向デコード開始位置
DcdStrLinPos	long	0	R/W	○	縦方向デコード開始位置
DigiThreshold	long	128	R/W	○	2 値化閾値
DigitizeMode	long	DIGI_OTSU (1)	R/W	○	2 値化方法
ErrorStatus	long	0	R	○	エラーステータス取得
InfData	long	-	R	○	デコード結果情報取得(VC++用)
InfDataV	Variant	-	R	○	デコード結果情報取得(VB 用)
LoadBMPFile	CString	""	R/W	×	BMP ファイルオープン
LoadTIFFFile	CString	""	R/W	×	TIFF ファイルオープン
MemAddr	long	-	W	×	画像メモリ開始番地(VC++用)
MemAddrV	Variant	-	W	×	画像メモリ開始番地(VB 用)
MemSizeX	long	0	R/W	×	画像メモリ X 方向サイズ
MemSizeY	long	0	R/W	×	画像メモリ Y 方向サイズ
MirrorMode	long	MIRROR_DIS(0)	R/W	○	表裏反転読みモード
ModSize	long	MOD_M(2)	R/W	○	モジュールサイズ
MultiBar	CString	""	R/W	○	多段バーコード設定パラメータ
PageNo	long	1	R/W	×	マルチページ TIFF ページ番号
QRJoinMode	long	QRJOIN_DIS(0)	R/W	○	QR 連結モード
RemoveNoise	long	0	R/W	○	ノイズフィルタ
Resolution	long	200	R/W	×	解像度
ReverseBW	long	REVERSE_DIS(0)	R/W	○	白黒反転読みモード
SelectArea	long	0	R/W	×	プロパティ設定対象エリア
SelectRGB	long	SELECT_R(0)	R/W	○	RGB 選択方式
Start	long	-	W	×	デコード開始
Status	long	0	R	○	デコード処理結果

注 1) エリア毎にプロパティを持つ場合、初期値は 0-3 のエリア全て同じ値になります。

注 2) カスタムプロパティの設定はプロパティページから実施しないで下さい。

6.2. カスタムプロパティの機能説明

名 称	機 能
AreaNum	デコードを実施するエリア数を設定
BarProperty	バーコード読み取りの詳細な設定
CodeKind	デコードするコード種別の設定
DecData	デコードデータの取得(VC++用)
DecDataV	デコードデータの取得(VB 用)
DcdEndColPos	LoadBMPFile, LoadTIFFFile プロパティで読み込んだ画像ファイルの一部のみを選択し処理する場合の桁終了位置
DcdEndLinPos	同上の行終了位置
DcdStrColPos	同上の桁開始位置
DcdStrLinPos	同上の行開始位置
DigiThreshold	DigitizeMode プロパティで固定閾値を選択した場合の閾値
DigitizeMode	グレイ画像の2値化方法の設定
ErrorStatus	エラー原因の取得
InfData	デコード結果情報の取得(VC++用)
InfDataV	デコード結果情報の取得(VB 用)
LoadBMPFile	デコードする BMP 形式の画像ファイルの設定
LoadTIFFFile	デコードする TIFF 形式の画像ファイルの設定
MemAddr	デコードする画像イメージ格納エリアの開始番地の設定(VC++用)
MemAddrV	デコードする画像イメージ格納エリアの開始番地の設定(VB 用)
MemSizeX	同上のX方向画素数の設定
MemSizeY	同上のY方向画素数の設定
MirrorMode	表裏反転ラベルの読み取り設定
ModSize	二次元コードまたはバーコードのモジュールサイズの設定
MultiBar	多段バーコードの組合せの設定
PageNo	マルチページ TIFF ファイルのページ番号設定
QRJoinMode	QRコード連結モードの一括読みの設定
RemoveNoise	ノイズフィルタの強さの設定
Resolution	解像度 (dpi 単位) の設定
ReverseBW	白黒反転デコードの設定
SelectArea	デコードエリアの選択設定
SelectRGB	カラーイメージのR,G,Bの選択設定
Start	デコード開始設定
Status	デコード結果の取得

6.2.1. AreaNum プロパティ

【機能】

デコードを実施するエリア数を設定します。指定されたエリア数だけデコードエリア 1 から順にデコードを実施していきます。デコード前に必ず各エリアのプロパティをセットしておいてください。プロパティをセットするエリアの切り替えは SelectArea プロパティをご使用ください。

【書式】

- | | |
|------------------|--|
| (1) Visual Basic | controlname.AreaNum=値 |
| (2) Visual C++ | controlname.SetAreaNum(long 値);
long 値=controlname.GetAreaNum(); |

【設定値】

デコードを実施するエリア数
1～4 の範囲の値
デフォルト値 = 1



6.2.2. BarProperty プロパティ

【機能】

バーコード読み取りの詳細な設定を行います。バーコード読み取りの詳細設定は複数設定が可能です。複数設定を行う場合はそれぞれの値を加算して下さい。

(例: BAR_C39_CKDG+BAR_NW7_CKDG+BAR_ITF_CKDG)

【書式】

- (1) Visual Basic controlname.BarProperty=値
- (2) Visual C++ controlname.SetBarProperty(long 値);
 long 値=controlname.GetBarProperty();

【設定値】

値	説明
BAR_C39_CKDG	CODE39 のチェックデジットをチェックする
BAR_NW7_CKDG	CODABAR のチェックデジットをチェックする
BAR_ITF_CKDG	ITF のチェックデジットをチェックする
BAR_POS_ADDON	UPC/EAN の AddOn コードを読む
デフォルト値 = 0(上記機能は全て禁止)	

6.2.3. CodeKind プロパティ

【機能】

デコードするコード種別を設定します。ただし、コード種別の指定は 1 コードのみです。複数指定した場合はエラーとなります。

【書式】

- (1) Visual Basic controlname.CodeKind =値
- (2) Visual C++ controlname.SetCodeKind(long 値);
long 値=controlname.GetCodeKind();

【設定値】

値	説明
CODE_QR	QRコード Model 1 又は Model 2 をデコード
CODE_MICROQR	MicroQR をデコード
CODE_DATAMATRIX	DataMatrix をデコード
CODE_PDF417	PDF417 をデコード
CODE_BARCODE	バーコード (UPC/EAN, CODE39, CODABAR(NW-7), Interleaved 2 of 5(ITF), CODE128, EAN-128, RSS)をデコード
CODE_MICROPDF	MicroPDF417 をデコード
CODE_COMPOSITE	EAN.UCC Composite をデコード
デフォルト値 = CODE_QR	



6.2.4. DecDataV, DecData プロパティ

【機能】

デコード結果を取得します。デコード結果には、連結QRコードや多段バーコードのように複数のコードを一括して読む場合には(EAN.UCC Composite は単独コード扱い)合成された結果がセットされています。InfData または InfDataV プロパティにて取得できるデータ数(合計データ数)分のデータがデコードデータとなります。デコード結果(Status プロパティで参照)が成功した場合に有効です。なお、設定値を変更することはできません。

【書式】 (VB 用と VC++用でプロパティが異なります)

- (1) Visual Basic エリア名= controlname.DecDataV
- (2) Visual C++ エリア開始番地=controlname.GetDecData();

【設定値】

デコード結果が格納されるエリア名(開始番地)



6.2.5. DcdStrLinPos, DcdEndLinPos, DcdStrColPos, DcdEndColPos プロパティ

【機能】

LoadBMPFile, LoadTIFFFile プロパティで読み込んだ画像ファイルの一部のみを選択し処理する場合に、その範囲を設定します。必ず LoadBMPFile, LoadTIFFFile プロパティを呼び出す前に呼び出して下さい。処理範囲を設定した場合、処理速度が向上します。デフォルト値の場合、処理範囲が画像ファイルの全範囲となります。処理範囲を設定せずにデコード対象エリア数を 4 にした場合、最大 320MB の動的メモリを占有しますのでご注意ください。

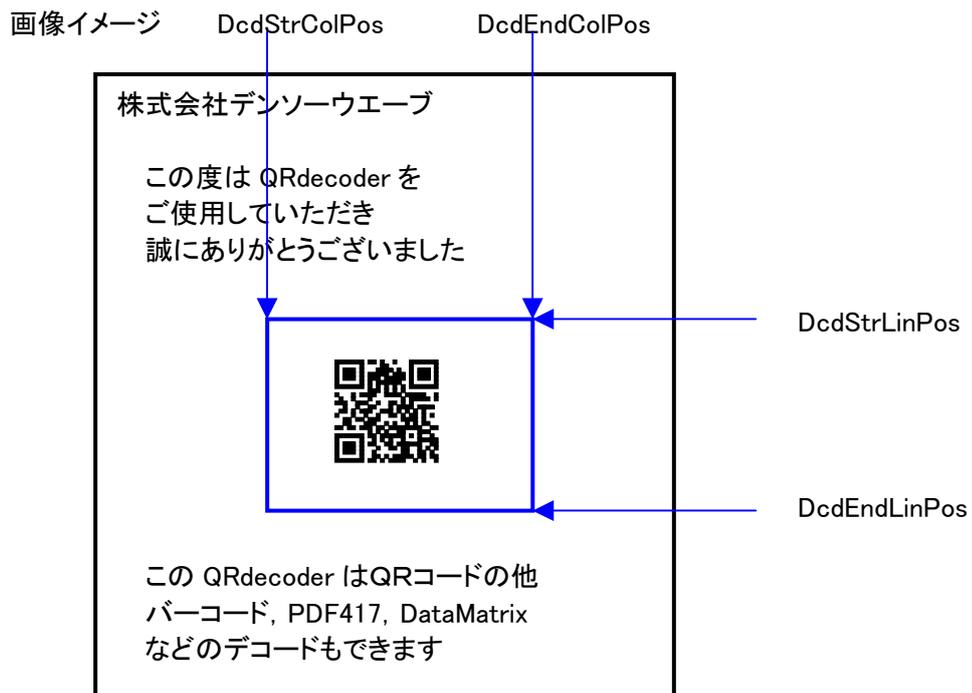
【書式】

- (1) Visual Basic controlname.DcdStrLinPos=値
- (2) Visual C++ controlname.SetDcdStrLinPos(long 値);
 long 値=controlname.GetDcdStrLinPos();

【設定値】

読み取り開始行, 終了行, 読み取り開始桁, 終了桁
行方向:0~10240, 桁方向:0~8192 の範囲の値
(ただし行桁方向共に画像サイズの範囲外を指定した場合はエラーとなります)
デフォルト値:
DcdStrLinPos=0, DcdEndLinPos=10240, DcdStrColPos=0, DcdEndColPos=8192

【設定例】





6.2.6. DigitizeMode プロパティ

【機能】

グレイ画像の2値化方法を設定します。スキャナーで撮った画像のように全体が一様な明るさの場合は判別閾値法による2値化(DIGI_OTSU)を、明るさにむらがある場合は局所2値化法による2値化(DIGI_LOCAL)を設定します。

【書式】

(1) Visual Basic controlname.DigitizeMode=値
 (2) Visual C++ controlname.SetDigitizeMode(long 値);
 long 値=controlname.GetDigitizeMode();

【設定値】

値	説明
DIGI_LOCAL	局所2値化法による2値化
DIGI_OTSU	判別閾値法による2値化
DIGI_FIXATION	固定閾値での2値化

デフォルト値 = DIGI_OTSU



6.2.7. DigiThreshold プロパティ

【機能】

DigitizeMode プロパティで固定閾値(DIGI_FIXATION)を設定した場合にのみ有効で、2 値化の閾値を設定します。

【書式】

- (1) Visual Basic controlname.DigiThreshold=値
- (2) Visual C++ controlname.SetDigiThreshold(long 値);
 long 値=controlname.GetDigiThreshold();

【設定値】

グレイ画像を2値化する場合の固定閾値
0～255 の範囲の値
デフォルト値 = 128

6.2.8. ErrorStatus プロパティ

【機能】

エラー原因を取得します。本プロパティは Start プロパティ呼び出し後に実行してください。プロパティの許容範囲オーバーの場合、許容範囲内の値を設定した場合にエラー原因はクリアされます。

【書式】

- (1) Visual Basic 値=controlname.ErrorStatus
 (2) Visual C++ long 値=controlname.GetErrorStatus();

【設定値】

戻り値	デコードエラー原因
ERST_NOERROR	エラーなし
ERST_RESOLUTION	Resolution プロパティで設定した解像度の許容範囲オーバー
ERST_MODSIZE	ModSize プロパティで設定したモジュールサイズ番号の許容範囲オーバー
ERST_CODEKIND	CodeKind プロパティで設定したコード種別の許容範囲オーバー
ERST_DIGITIZE	DigitizeMode プロパティで設定した2値化方法の許容範囲オーバー
ERST_TRESHOLD	DigiThreshold プロパティで設定した2値化閾値の許容範囲オーバー
ERST_MEMSIZE	MemSizeX, MemSizeY プロパティで設定した画像イメージのサイズオーバー
ERST_FRAMING	DcdStrLinPos , DcdEndLinPos , DcdStrColPos , DcdEndColPos プロパティで設定した範囲指定エラー
ERST_LOADIMAGE	LoadBMPFile, LoadTIFFFile プロパティで設定した画像ファイルロードエラー
ERST_MLTBAR	MultiBar プロパティで設定した多段バーコード設定パラメータが不正
ERST_QRJOINMODE	QRJoinMode プロパティで設定した QRコード編集モードの許容範囲オーバー
ERST_REMOVE NOISE	RemoveNoise プロパティで設定したノイズフィルタ値の許容範囲オーバー
ERST_REVERSEBW	ReverseBW プロパティで設定した白黒反転設定値の許容範囲オーバー
ERST_SELRGB	SelectRGB プロパティで設定した RGB 選択値の許容範囲オーバー



戻り値	デコードエラー原因
ERST_BARPROPERTY	BarProperty プロパティで設定したバーコード詳細設定の許容範囲オーバー
ERST_PROPERTY	必須プロパティ(「6.3 必須プロパティの設定順序」参照)が設定されていない
ERST_MIRROR	MirrorMode プロパティで設定した表裏反転設定値の許容範囲オーバー
ERST_PAGENO	PageNo プロパティで設定したマルチページ TIFF ファイルのページ番号設定値の許容範囲オーバー
ERST_SELECTAREA	SelectArea プロパティで設定したデコードエリアの選択設定値の許容範囲オーバー
ERST_AREANUM	AreaNum プロパティで設定したデコードを実施するエリア数設定値の許容範囲オーバー
ERST_MEMORY	メモリの不足
ERST_NOPRODUCTNO	プロテクションキー(「5. インストール/アンインストール」参照)が未登録または試用期間切れ
ERST_DECODE	デコード失敗



6.2.9. InfDataV, InfData プロパティ

【機能】

デコード結果情報を取得します。単独のコードを読む場合には long[MAXDECINF]分の配列としてデータを処理してください(MAXDECINF は QRDecOCX.h に定義されています)。連結コードや多段バーコードのように複数のコードを合成し1つのデータを出す場合(EAN.UCC Composite は単独コード扱い)には long[最大コード数][MAXDECINF]の配列としてデータを処理してください。セットされる情報はコードマーク、バージョン番号、誤り訂正数などです(「8. デコード情報テーブル」参照)。なお、設定値を変更することはできません。

【書式】 (VB 用と VC++用でプロパティが異なります)

- (1) Visual Basic エリア名=controlname.InfDataV
- (2) Visual C++ エリア開始番地=controlname.GetInfData ();

【設定値】 (必須)

デコード結果情報が格納されるエリア名(開始番地)



6.2.10. LoadBMPFile プロパティ

【機能】

デコード対象の BMP 形式の画像ファイル名(パスを含む)を設定します。対応する BMP ファイルは以下のフォーマットです。画像イメージ画素数の上限はX方向 8192 画素, Y方向 10240 画素です。

圧縮形式	モノクロ 2 値	256 色グレイ	256 色カラー	16bit カラー	24bit カラー
非圧縮	○	○	○	○	○

○:対応

【書式】

- (1) Visual Basic controlname.LoadBMPFile="ファイルパス"
 (2) Visual C++ controlname.SetLoadBMPFile("ファイルパス");

【設定値】

BMP 形式の画像ファイル名(パスを含む)
 BMP 形式ファイルからデコードする場合は必須

6.2.11. LoadTIFFFile プロパティ

【機能】

TIFF 形式の画像ファイル名(パスを含む)を設定します。対応する TIFF ファイルは以下のフォーマットです。マルチページ TIFF ファイルにも対応しています。画像イメージ画素数の上限はX方向 8192 画素, Y方向 10240 画素です。

圧縮形式	モノクロ 2 値	256 色グレー	256 色カラー	16bit カラー	24bit カラー
非圧縮	○	○	○	—	○
G3	○	—	—	—	—
G4	○	—	—	—	—
LZW	○	—	○	—	○

○:対応 —:非対応

【書式】

- (1) Visual Basic controlname.LoadTIFFFile("ファイルパス")
- (2) Visual C++ controlname.SetLoadTIFFFile("ファイルパス");

【設定値】

TIFF 形式の画像ファイル名(パスを含む)
TIFF 形式ファイルからデコードする場合は必須



6.2.12. MemAddrV, MemAddr プロパティ

【機能】

画像イメージ格納エリアの開始番地を設定します。画像イメージ画素数の上限はX方向 8192 画素, Y方向 10240 画素です。なお, 設定値を読み出すことはできません。

【書式】 (VB 用と VC++用でプロパティが異なります)

- (1) Visual Basic controlname.MemAddrV=エリア名
- (2) Visual C++ controlname.SetMemAddr(エリア開始番地);

【設定値】

256 階調の画像イメージの開始番地
アプリケーションプログラムで画像イメージをメモリに展開し, そこからデコードする場合には後述する MemSizeX, MemSizeY プロパティと共に必須です。
LoadBMPFile 又は LoadTIFFFile プロパティを指定し, 直接画像ファイルの中からデコードする場合は不要です。



6.2.13. MemSizeX, MemSizeY プロパティ

【機能】

画像イメージのX方向とY方向画素数を設定します。画像イメージ画素数の上限はX方向 8192 画素, Y方向 10240 画素です。

【書式】

- (1) Visual Basic controlname.MemSizeX =値
- (2) Visual C++ controlname.SetMemSizeX (long 値);
 long 値=controlname.GetMemSizeX ();

【設定値】

X方向又はY方向画素数
アプリケーションプログラムで画像イメージをメモリに展開しそこからデコードする場合には前述の MemAddr プロパティと共に必須です。
LoadBMPFile 又は LoadTIFFFile プロパティを指定し、直接画像ファイルの中からデコードする場合は不要です。

【値の参照】

LoadBMPFile 又は LoadTIFFFile プロパティを実行した後はこのプロパティにロードしたイメージの画素数がセットされており、値を取得することが可能です。



6.2.14. MirrorMode プロパティ

【機能】

表裏反転ラベルの読み取りの許可/禁止を設定します。本設定は QR コード, MicroQR, DataMatrix の読み取りの場合に有効です。

【書式】

- (1) Visual Basic controlname.MirrorMode =値
- (2) Visual C++ controlname.SetMirrorMode (long 値);
 long 値=controlname.GetMirrorMode ();

【設定値】

値	説明
MIRROR_DIS	表裏反転ラベル読み取り禁止(表読み限定)
MIRROR_ENA	表裏反転ラベル読み取り許可(表読み禁止)
デフォルト値 = MIRROR_DIS	



6.2.15. ModSize プロパティ

【機能】

画像中の二次元コードまたはバーコードのモジュールサイズを設定します。サイズは下記の[設定値]の表から選択します。実際のサイズが下記表の範囲の中央付近になる値を選択して下さい。

【書式】

- (1) Visual Basic controlname.ModSize =値
 (2) Visual C++ controlname.SetModSize(long 値);
 long 値=controlname.GetModSize();

【設定値】

値	説明
MOD_SS	0.25 mm 以下
MOD_S	0.15 ~ 0.40 mm
MOD_M	0.25 ~ 0.75 mm
MOD_L	0.50 ~ 1.25 mm
MOD_LL	1 mm 以上
デフォルト値 = MOD_M	



6.2.16. MultiBar プロパティ

【機能】

多段バーコード(最大3段まで)の組合せを設定します。

【書式】

- (1) Visual Basic controlname.MultiBar="パラメータ文字列"
 (2) Visual C++ controlname.SetMultiBar("パラメータ文字列");

【設定値】 (多段バーコードを読ませる場合は必須)

多段バーコードの組合せパラメータ

形式: 1段目の指定, 2段目の指定[, 3段目の指定]

各段の形式: コードマーク[:[桁数[-最大桁数]][:比較文字1[比較文字2]]]

- ・ コードマークは「8.4 バーコードの場合」のコードマーク一覧表をご参照下さい。ただし2次元コード及びRSSコードは多段バーコードとして設定できません。
- ・ 桁数を設定するとデコードした桁数が設定した値, 又はその範囲に入っている場合のみ読み取ることができます。)
- ・ 比較文字を設定するとCODABAR以外のコードの場合はコードの1桁目[と2桁目]の文字が設定した文字と一致する場合のみ読み取ることができます。(CODE39の*は含みません)
- ・ CODABARの場合は比較文字1がスタート文字に, 比較文字2がストップ文字に対応します。

【設定例】

M,M,M	CODE39の3段
N:10,I:5	1段目:CODABARの10桁 2段目:ITFの5桁
K:5-7,K:8-10	1段目:CODE128の5~7桁 2段目:CODE128の8~10桁
A::12,A::34	1段目:EAN-13/UPC-Aの1桁目が'1'で2桁目が'2' 2段目:EAN-13/UPC-Aの1桁目が'3'で2桁目が'4'



6.2.17. PageNo プロパティ

【機能】

マルチページ TIFF ファイルでデコードを実施するページ番号を設定します。0 が選択された場合、1 ページ目から順に全ページをデコードが成功するまで実施します。複数エリアの処理を実施している場合、1 エリアでも読み取りが成功した場合にデコードを終了します。LoadTIFFFile プロパティでマルチページ TIFF ファイルが設定された場合のみ有効です。

【書式】

- (1) Visual Basic controlname.PageNo=値
- (2) Visual C++ controlname.SetPageNo(long 値);
 long 値=controlname.GetPageNo();

【設定値】

デコードを実施するマルチページ TIFF のページ番号
0~99 の範囲の値
(0 の場合はデコードが成功するまで全ページをデコードします)
デフォルト値 = 1



6.2.18. QRJoinMode プロパティ

【機能】

QRコード連結コードの一括読みを設定します。連結コードとは、QRコードモデル1またはモデル2のデータを最大16個に分割したコードのことです。一括読みを設定した場合は画像エリア中に連結する全てのコードが入っていないと読み取りが成功しません(最大16個まで)。単独読みを設定した場合は、最初に読めたコードのみが出力されます。連結コードは同一のQRコードモデルのみ読み取り可能です。

【書式】

- (1) Visual Basic controlname.QRJoinMode =値
- (2) Visual C++ controlname.SetQRJoinMode(long 値);
 long 値=controlname.GetQRJoinMode();

【設定値】

値	説明
QRJOIN_DIS	単独読み(1コードずつ読み取る)
QRJOIN_ENA	全ての連結コードを一括して読み取る
デフォルト値 = QRJOIN_DIS	



6.2.19. RemoveNoise プロパティ

【機能】

移動平均法によるデジタルフィルタの設定をします。スムージングサイズは(設定値 × 2+1)の値になります。

【書式】

- | | |
|------------------|--|
| (1) Visual Basic | controlname.RemoveNoise=値 |
| (2) Visual C++ | controlname.SetRemoveNoise(long 値);
long 値=controlname.GetRemoveNoise(); |

【設定値】

デジタルフィルタのスムージングサイズ(0(フィルタなし)~50(スムージングサイズ 101))
デフォルト値 = 0



6.2.20. Resolution プロパティ

【機能】

デコードする画像の解像度(dpi 単位)を設定します。解像度の上限は 4800dpi です。

【書式】

(1) Visual Basic	controlname.Resolution =値
(2) Visual C++	controlname.SetResolution(long 値); long 値=controlname.GetResolution();

【設定値】

解像度(dpi 単位)
デフォルト値 = 200dpi



6.2.21. ReverseBW プロパティ

【機能】

白黒反転印刷されたコードを読む時に設定します。LoadBMPFile 又は LoadTIFFFile プロパティを使用してデコードした場合に本設定は有効です。

【書式】

- (1) Visual Basic controlname. ReverseBW=値
- (2) Visual C++ controlname.SetReverseBW(long 値);
 long 値=controlname.GetReverseBW();

【設定値】

値	説明
REVERSE_DIS	通常(白地に黒印刷)のコードを読む
REVERSE_ENA	白黒反転印刷されたコードを読む
デフォルト値 = REVERSE_DIS	



6.2.22. SelectArea プロパティ

【機能】

プロパティの設定または参照をするデコードエリアを指定します。希望するエリアのプロパティを設定または参照したい場合はこのプロパティでデコードエリアを指定後に実施してください。

【書式】

- (1) Visual Basic controlname. SelectArea=値
- (2) Visual C++ controlname.SetSelectArea(long 値);
 long 値=controlname.GetSelectArea();

【設定値】

プロパティをセットするデコードエリア番号
0～3 の範囲の値
デフォルト値 = 0



6.2.23. SelectRGB プロパティ

【機能】

カラーイメージの場合にR, G, Bの内どの色の情報を使用するかを設定します.
LoadBMPFile 又は LoadTIFFFile プロパティを使用してデコードした場合に本設定は有効です.

【書式】

- (1) Visual Basic controlname.SelectRGB =値
- (2) Visual C++ controlname.SetSelectRGB(long 値);
 long 値=controlname.GetSelectRGB();

【設定値】

値	説明
SELECT_R	R(赤)を使用
SELECT_G	G(緑)を使用
SELECT_B	B(青)を使用
デフォルト値 = SELECT_R	



6.2.24. Start プロパティ

【機能】

デコード処理を開始します。このプロパティの実行に先立って必要な設定プロパティを実行しておく必要があります。（「6.3 必須プロパティの設定順序」参照）

【書式】

- (1) Visual Basic controlname.Start=0
- (2) Visual C++ controlname.SetStart(0);

【設定値】

0



6.2.25. Status プロパティ

【機能】

デコード処理結果を取得します。本プロパティは Start プロパティ呼び出し後に実行してください。

【書式】

- (1) Visual Basic 値=controlname.Status
- (2) Visual C++ long 値=controlname.GetStatus();

【値の設定】

不可

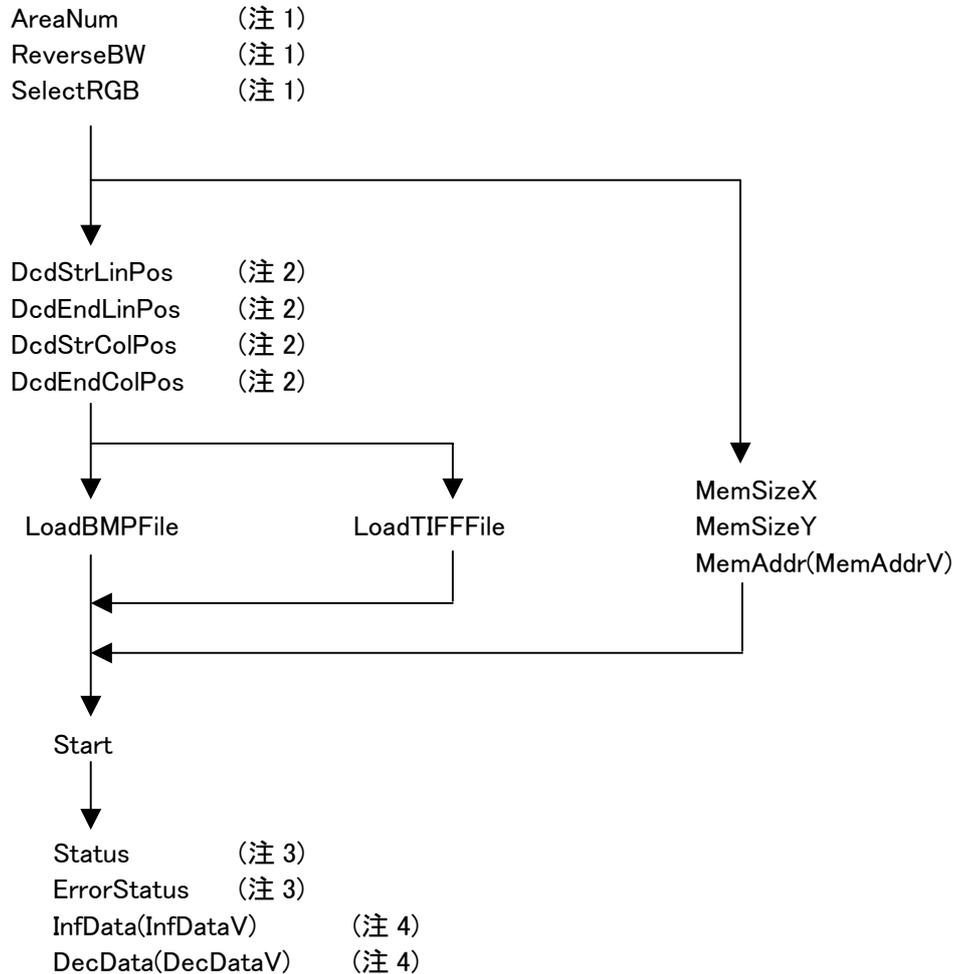
【値の参照】

デコード処理結果(1=デコード成功 / 0=失敗)



6.3. 必須プロパティの設定順序

必須プロパティは以下の順に設定してください。その他のプロパティは必要に応じ Start プロパティの設定前までに設定して下さい。



注 1: LoadBMPFile, LoadTIFFFile プロパティにて画像ファイルからデコードする場合に有効です。設定は必須ではありませんが、設定する場合は LoadBMPFile, LoadTIFFFile プロパティ呼び出し前に行ってください。

注 2: LoadBMPFile, LoadTIFFFile プロパティにて画像ファイルからデコードする場合に有効です。設定は必須ではありませんが、設定する場合は LoadBMPFile, LoadTIFFFile プロパティ呼び出し前に行ってください。デフォルト設定の場合は画像ファイルの全領域が処理対象となります。

注 3: 取得は必須ではありませんが、取得する場合は Start プロパティ呼び出し後に行ってください。

注 4: 取得は必須ではありませんが、取得する場合は Start プロパティ呼び出し後で、Status プロパティ取得時にデコード成功の場合に行ってください。デコード失敗の場合本プロパティで取得した値は無効です。

7. Visual C++での利用

ここでは VC++6.0 を前提に説明します。

- 新規プロジェクトを MFC AppWizard(exe)で作成する。
- OCX コントロールを挿入するため、ダイアログベースまたは、SDI や MDI でのビュークラスを CFormView 指定で雛がたを作成する。
- [プロジェクト][プロジェクトへ追加][コンポーネントおよびコントロール]でコンポーネントおよびコントロールギャラリーを開き、Registered ActiveX Controls の中から、QRdecoder Control を挿入ボタンでプロジェクトに挿入します。
- OCX コントロールをプロジェクトに追加すると、CQRdecoder クラスを提供するファイル(.cpp と.h)が生成されます。
- リソースエディタで、AppWizard で生成したダイアログを開き、そこへコントロールツールバーから QRdecoder のアイコンを選択し、ダイアログへ貼り付けます。
- クラスウィザードを用いて挿入した OCX コントロールに対応するメンバー変数 m_QrdecoderCtrl を作成します。

以下はこのメンバー変数 m_QrdecoderCtrl を用いたプロパティの使用例です。

7.1. 画像ファイルからのデコード(1 エリア指定)

[条件]

画像ファイル形式: BMP 形式(非圧縮)
 画像ファイル名: image1.bmp(Cドライブの直下に配置)
 画像ファイルの解像度: 400dpi
 1セルのモジュールサイズ: 0.5mm
 コード種別: QRコード
 デコード処理範囲: 全エリア

```
void CTestDecode::OnDecode()
{
    //(1)ワークエリアを確保します.
    byte *DecData;           //デコード情報先頭アドレス
    long *InfData;          //デコードデータ先頭アドレス
    long DecdInfTbl[MAXDECINF]; // デコード情報テーブル
    unsigned char DecodeData[3000]; // デコードデータテーブル
    long Err;                // エラー要因
    int DataLen;             // デコードデータ長

    //(2)各プロパティの設定を行います.
    // スキャナー解像度
    m_QrdecoderCtrl.SetResolution((long)400);
    // モジュールサイズ
    m_QrdecoderCtrl.SetModSize((long)MOD_M);
    // コード種別
    m_QrdecoderCtrl.SetCodeKind((long)CODE_QR);
    // 画像ファイル
    m_QrdecoderCtrl.SetLoadBMPFile("C:¥¥¥image1.bmp");

    // (3)デコード実施.
    m_QrdecoderCtrl.SetStart(0);
}
```



```
// (4)デコード結果参照.
if (m_QrdecoderCtrl.GetStatus() == (long>true)
{
// デコード成功(デコードデータ取得)
// デコード情報取得
InfData = (long *)m_QrdecoderCtrl.GetInfData();
memcpy(DecdInfTbl, InfData, MAXDECINF*sizeof(long));
DataLen = DecdInfTbl[DIF_DATLEN];
// デコードデータ取得
DecData = (unsigned char *)m_QrdecoderCtrl.GetDecData();
memcpy(DecodeData, DecData, DataLen);
}
else {
// デコード失敗(エラー要因の取得)
Err = m_QrdecoderCtrl.GetErrorStatus();
}
}
```



7.2. 画像ファイルからのデコード(複数エリア指定)

[条件]

画像ファイル形式: BMP 形式(非圧縮)

画像ファイル名: image2.bmp(Cドライブの直下に配置)

画像ファイルの解像度: 200dpi

エリア 1:

1 セルのモジュールサイズ: 0.25mm

コード種別: QR コード

処理範囲 YS1-YE1 行, XS1-XE1 桁

エリア 2:

1 セルのモジュールサイズ: 0.5mm

コード種別: DataMatrix

処理範囲 YS2-YE2 行, XS2-XE2 桁



```

void CTestDecode::OnDecode2()
{
    // (1)ワークエリアを確保します.
    byte *DecData;           //デコード情報先頭アドレス
    long *InfData;          //デコードデータ先頭アドレス
    long DecdInfTbl[2][MAXDECINF]; // デコード情報テーブル
    char DecodeData[2][3000]; // デコードデータテーブル
    long Err1, Err2;        // エラー要因
    int  DataLen1, DataLen2; // デコードデータ長

    //(2)各プロパティの設定を行います.
    // エリア 1 指定
    m_QrdecoderCtrl.SetSelectArea((long)0);
    // モジュールサイズ
    m_QrdecoderCtrl.SetModSize((long)MOD_S);
    // コード種別
    m_QrdecoderCtrl.SetCodeKind((long)CODE_QR);
    // デコード処理範囲指定(開始行)
    m_QrdecoderCtrl.SetDcdStrLinPos((long)YS1);
    // デコード処理範囲指定(開始桁)
    m_QrdecoderCtrl.SetDcdStrColPos((long)XS1);
    // デコード処理範囲指定(終了行)
    m_QrdecoderCtrl.SetDcdEndLinPos((long)YE1);
    // デコード処理範囲指定(終了桁)
    m_QrdecoderCtrl.SetDcdEndColPos((long)XE1);

    // エリア 2 指定
    m_QrdecoderCtrl.SetSelectArea((long)1);
    // モジュールサイズ
    m_QrdecoderCtrl.SetModSize((long)MOD_M);
    // コード種別
    m_QrdecoderCtrl.SetCodeKind((long)CODE_DATAMATRIX);
    // デコード処理範囲指定(開始行)
    m_QrdecoderCtrl.SetDcdStrLinPos((long)YS2);
    // デコード処理範囲指定(開始桁)
    m_QrdecoderCtrl.SetDcdStrColPos((long)XS2);
    // デコード処理範囲指定(終了行)
    m_QrdecoderCtrl.SetDcdEndLinPos((long)YE2);
    // デコード処理範囲指定(終了桁)
    m_QrdecoderCtrl.SetDcdEndColPos((long)XE2);

    // 共通指定
    // スキャナー解像度
    m_QrdecoderCtrl.SetResolution((long)200);
    // デコード実施エリア数指定
    m_QrdecoderCtrl.SetAreaNum((long)2);
    // 画像ファイル
    m_QrdecoderCtrl.SetLoadBMPFile("C:¥¥image2.bmp");
}

```



```

// (3)デコード実施
m_QrdecoderCtrl.SetStart(0);

// (4)デコード結果参照.
// エリア 1 の結果参照
m_QrdecoderCtrl.SetSelectArea ((long)0);
if (m_QrdecoderCtrl.GetStatus() == (long>true)
{
    // デコード成功(デコードデータ取得)
    // デコード情報取得
    InfData = (long *)m_QrdecoderCtrl.GetInfData();
    memcpy(&DecdInfTbl[0][0], InfData,
           MAXDECINF*sizeof(long));
    DataLen1 = DecdInfTbl[0][DIF_DATLEN];
    // デコードデータ取得
    DecData = (unsigned char *)m_QrdecoderCtrl.GetDecData();
    memcpy(&DecodeData[0][0], DecData, DataLen1);
}
else {
    // デコード失敗(エラー要因の取得)
    Err1 = m_QrdecoderCtrl.GetErrorStatus();
}
// エリア 2 の結果参照
m_QrdecoderCtrl. SetSelectArea ((long)1);
if (m_QrdecoderCtrl.GetStatus() == (long>true)
{
    // デコード成功(デコードデータ取得)
    // デコード情報取得
    InfData = (long *)m_QrdecoderCtrl.GetInfData();
    memcpy(&DecdInfTbl[1][0], InfData,
           MAXDECINF*sizeof(long));
    DataLen2 = DecdInfTbl[1][DIF_DATLEN];
    // デコードデータ取得
    DecData = (unsigned char *)m_QrdecoderCtrl.GetDecData();
    memcpy(&DecodeData[1][0], DecData, DataLen2);
}
else {
    // デコード失敗(エラー要因の取得)
    Err2 = m_QrdecoderCtrl.GetErrorStatus();
}
}

```

8. デコード情報テーブル

情報テーブルはデコード結果が格納される long 型の配列テーブルで、格納される情報はデコードしたコード種別により異なります。

8.1. QRコードの場合

配列数は単独ラベルのデコードの場合は「QRDecOCX.h」に定義されています。

MAXDECINF 分(下表に相当)格納されています。

例: long decodeinf[`MAXDECINF`];

位置	内容	
DIF_ERRST	デコードステータス(0=成功/−1=失敗)	
DIF_DATLEN_SUM	合計データ数	
DIF_CODEMARK	コードマーク (単独コード, 連結コード一括読み時: 'Q', 連結コード単独読み時: 'S')	
DIF_MODELNO	モデル番号 (0=モデル 1, 1=モデル 2, 2=MicroQR)	
DIF_VERNO	バージョン番号	
DIF_CELNO	1辺のセル数	
DIF_CORRLVL	誤り訂正レベル(0=L, 1=M, 2=Q, 3=H)	
DIF_DIVNO	分割数 (連結でない場合は 0)	
DIF_DIVPOS	分割時の位置 (連結でない場合は 0)	
DIF_GLEN	総コードワード数	
DIF_CORSU	誤り訂正数	
DIF_DATLEN	分割デコードデータ長(1つのコード内の値)	
DIF_SKEW	コードの傾き(度)(時計回り方向)	
DIF_PARITY	パリティ(連結でない場合は 0)	
DIF_C1POS_X	第1位置検出パターン(次図参照)のコーナ	X 座標
DIF_C1POS_Y	〃	Y座標
DIF_C2POS_X	第2位置検出パターン(次図参照)のコーナ	X 座標
DIF_C2POS_Y	〃	Y座標
DIF_C3POS_X	第3位置検出パターン(次図参照)のコーナ	X 座標
DIF_C3POS_Y	〃	Y座標
DIF_C4POS_X	第1位置検出パターン(次図参照)反対側コーナ	X 座標
DIF_C4POS_Y	〃	Y 座標
DIF_PAGENO	デコードが成功したページ番号 (TIFF ファイルでない場合は 0)	

【例】位置検出パターン

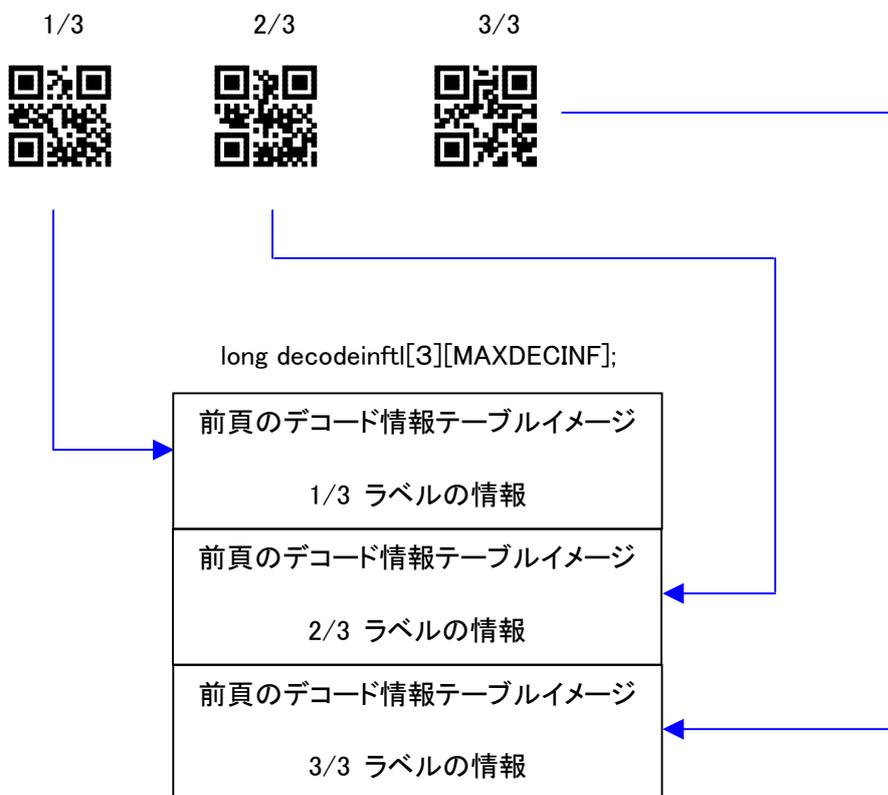


連結コードの一括読み込みの場合は上記サイズ×分割数分格納されています。

例: `long decodeinftrl[分割数][MAXDECINF];` (下図参照)

「位置」も、「QRDecOCX.h」に定義されています。

【例】3分割コード一括読み込みの場合

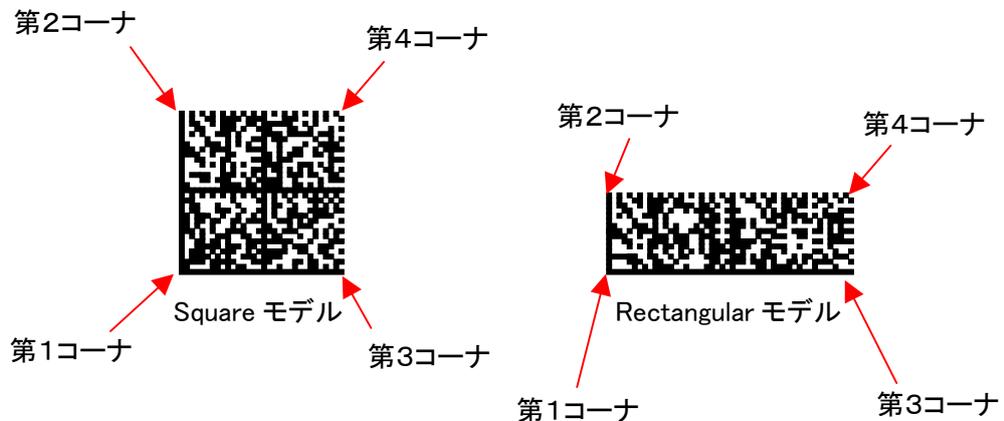




8.2. DataMatrix の場合

配列数は「QRDecOCX.h」に定義されています。
MAXDECINF 分(下表に相当)格納されています。

位置	内容
DIF_ERRST	デコードステータス(0 = 成功 / -1 = 失敗)
DIF_DATLEN_SUM	データ数
DIF_CODEMARK	コードマーク 'Z'
DIF_MODELNO	モデル番号 (0 = Square, 1 = Rectangular)
DIF_VERNNO	縦方向セル数
DIF_GELNO	横方向セル数
DIF_CORRLVL	0
DIF_DIVNO	0
DIF_DIVPOS	0
DIF_CLEN	総コードワード数
DIF_CORSU	誤り訂正数
DIF_DATLEN	データ数
DIF_SKEW	コードの傾き(度)(時計回り方向)
DIF_PARITY	0
DIF_C1POS_X	第1コーナ(下図参照) X 座標
DIF_C1POS_Y	" Y座標
DIF_C2POS_X	第2コーナ(下図参照) X 座標
DIF_C2POS_Y	" Y座標
DIF_C3POS_X	第3コーナ(下図参照) X 座標
DIF_C3POS_Y	" Y座標
DIF_C4POS_X	第4コーナ(下図参照) X 座標
DIF_C4POS_Y	" Y座標
DIF_PAGENO	デコードが成功したページ番号 (TIFF ファイルでない場合は 0)

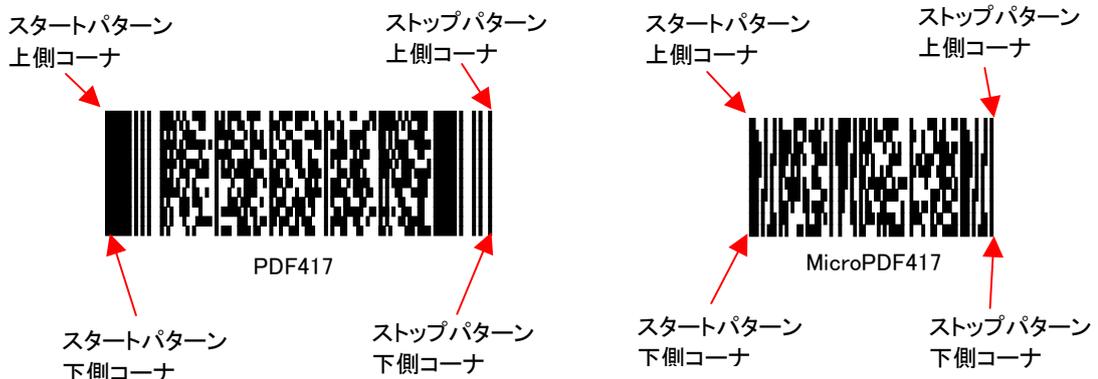




8.3. PDF417 コードの場合

配列数は「QRDecOCX.h」に定義されています。
MAXDECINF 分(下表に相当)格納されています。

位置	内容
DIF_ERRST	デコードステータス(0= 成功/-1= 失敗)
DIF_DATLEN_SUM	データ数
DIF_CODEMARK	コードマーク 'Y'
DIF_MODELNO	モデル番号 (0=PDF417, 1=MicroPDF417)
DIF_VERNO	段数
DIF_GELNO	桁数
DIF_CORRLVL	誤り訂正レベル(MicroPDF417 は 0)
DIF_DIVNO	0
DIF_DIVPOS	0
DIF_GLEN	総コードワード数
DIF_CORSU	誤り訂正数
DIF_DATLEN	データ数
DIF_SKEW	コードの傾き(度)(時計回り方向)
DIF_PARITY	0
DIF_C1POS_X	スタートパターン上側コーナ X 座標
DIF_C1POS_Y	" Y座標
DIF_C2POS_X	スタートパターン下側コーナ X 座標
DIF_C2POS_Y	" Y座標
DIF_C3POS_X	ストップパターン上側コーナ X 座標
DIF_C3POS_Y	" Y座標
DIF_C4POS_X	ストップパターン下側コーナ X 座標
DIF_C4POS_Y	" Y座標
DIF_PAGENO	デコードが成功したページ番号 (TIFF ファイルでない場合は 0)





8.4. バーコードの場合

配列数は単独ラベルのデコードの場合は「QRDecOCX.h」に定義されています。

MAXDECINF 分(下表に相当)格納されています。

例: long decodeinf[`MAXDECINF`];

位置	内容
DIF_ERRST	デコードステータス(0= 成功/-1= 失敗)
DIF_DATLEN_SUM	合計データ数
DIF_CODEMARK	コードマーク(次ページ)
DIF_MODELNO	0
DIF_VERNO	0
DIF_CELNO	0
DIF_CORRLVL	0
DIF_DIVNO	0
DIF_DIVPOS	0
DIF_CLEN	0
DIF_CORSU	0
DIF_DATLEN	データ数(多段の場合は1つのコード内の値)
DIF_SKEW	コードの傾き(度)(時計回り方向)
DIF_PARITY	0
DIF_C1POS_X	スタートパターン上側コーナ X 座標
DIF_C1POS_Y	" Y座標
DIF_C2POS_X	スタートパターン下側コーナ X 座標
DIF_C2POS_Y	" Y座標
DIF_C3POS_X	ストップパターン上側コーナ X 座標
DIF_C3POS_Y	" Y座標
DIF_C4POS_X	ストップパターン下側コーナ X 座標
DIF_C4POS_Y	" Y座標
DIF_PAGENO	デコードが成功したページ番号 (TIFF ファイルでない場合は 0)

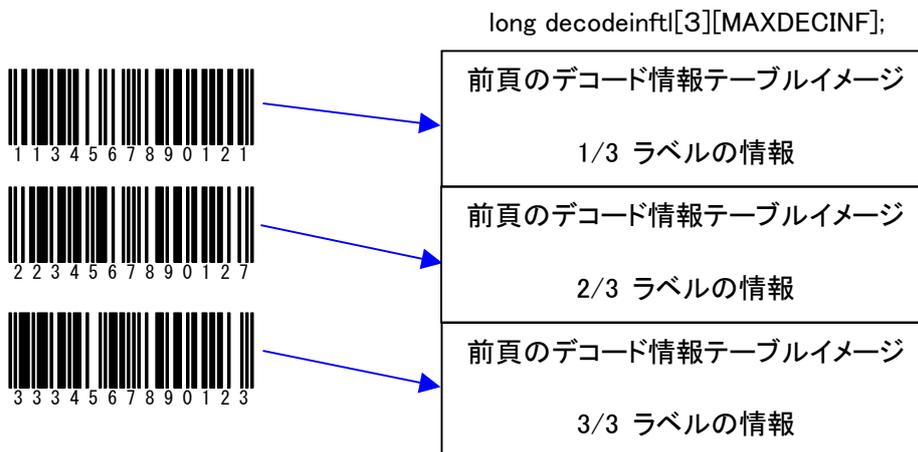




多段バーコードの一括読みの場合には上記サイズ×段数(最大3段)分格納されています。

例: long decodeinftrl[段数][MAXDECINF]; (下図参照)

【例】3段コード一括読みの場合



【例】バーコードのコードマーカー一覧表

コード種別	コードマーク	コード種別	コードマーク
EAN-13/UPC-A	‘A’	EAN-8	‘B’
UPC-E	‘C’	CODE39	‘M’
CODABAR	‘N’	ITF	‘I’
CODE128	‘K’	EAN-128	‘W’
RSS	‘R’		

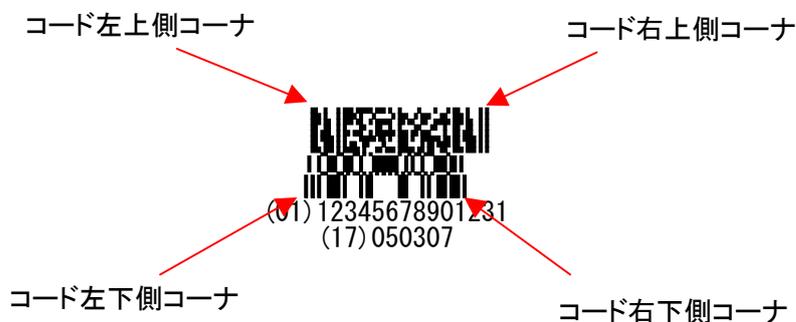


8.5. EAN.UCC Composite の場合

配列数は「QRDecOCX.h」に定義されています。

MAXDECINF 分(下表に相当)格納されています。

位置	内容
DIF_ERRST	デコードステータス(0= 成功/-1= 失敗)
DIF_DATLEN_SUM	合計データ数
DIF_CODEMARK	コードマーク('V')
DIF_MODELNO	0
DIF_VERNO	0
DIF_GELNO	0
DIF_CORRLVL	0
DIF_DIVNO	0
DIF_DIVPOS	0
DIF_CLEN	0
DIF_CORSU	0
DIF_DATLEN	合計データ数
DIF_SKEW	コードの傾き(度)(時計回り方向)
DIF_PARITY	0
DIF_C1POS_X	コード左上側コーナ X 座標
DIF_C1POS_Y	" Y座標
DIF_C2POS_X	コード左下側コーナ X 座標
DIF_C2POS_Y	" Y座標
DIF_C3POS_X	コード右上側コーナ X 座標
DIF_C3POS_Y	" Y座標
DIF_C4POS_X	コード右下側コーナ X 座標
DIF_C4POS_Y	" Y座標
DIF_PAGENO	デコードが成功したページ番号 (TIFF ファイルでない場合は 0)





9. 改訂履歴

日付	内容
2006.09.15	第1版

【商標および登録商標など】

このユーザズガイドの著作権は、(株)デンソーウェーブにあります。
また、QRコード、QR Code は(株)デンソーウェーブの登録商標です。

Windows 2000, Windows XP, Visual Basic, Visual C++, Visual Studio, ActiveX は、
米国マイクロソフト社の商標または登録商標です。

その他、本書に記載されている商品名は、一般に該当する各社の商標または登録商標です。