

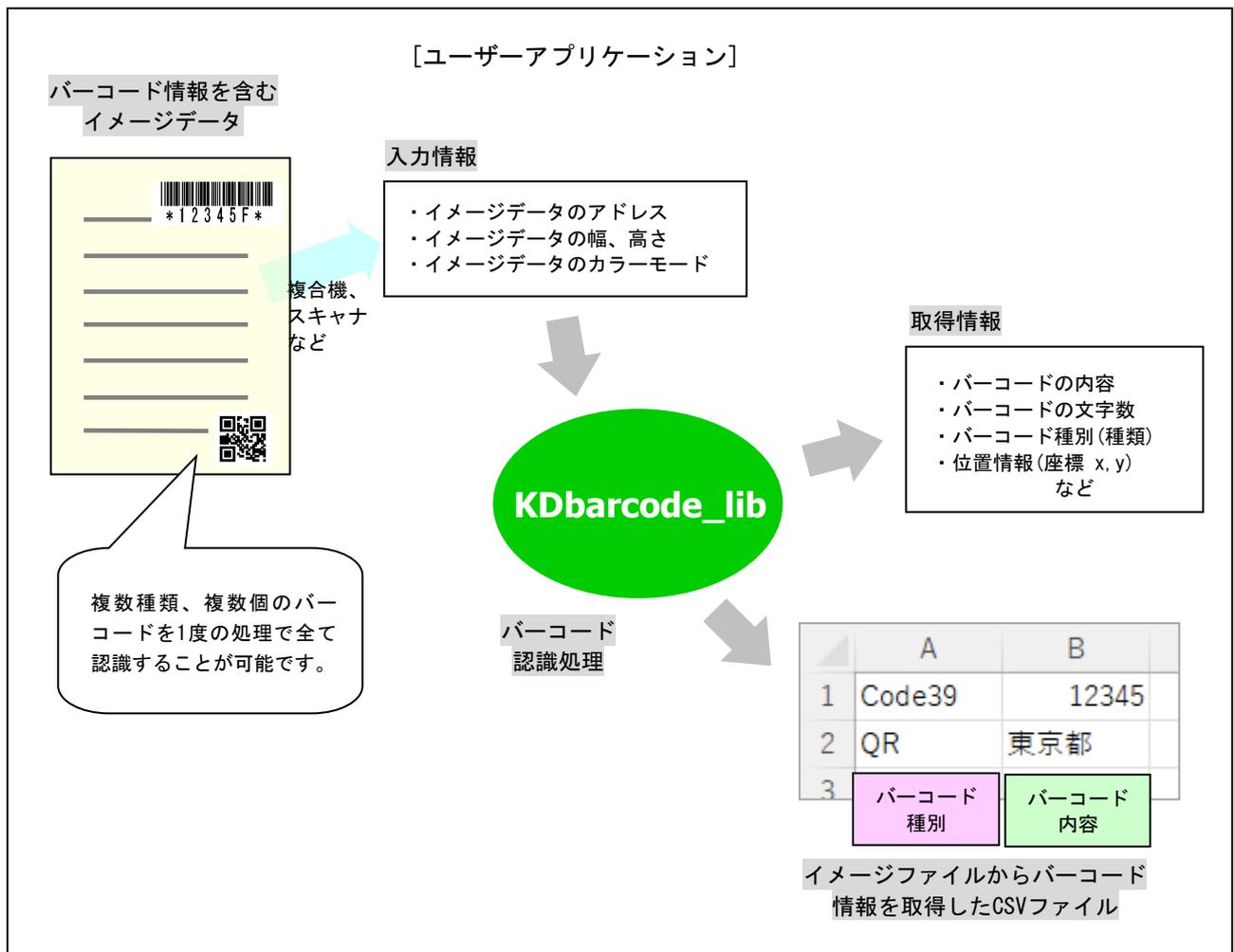
# バーコード読み取りライブラリ KBarcode\_lib

**KCS** Kernel Computer System  
カーネルコンピュータシステム株式会社

本社：パッケージ販売部  
〒221-0056  
横浜市神奈川区金港町6-3横浜金港町ビル  
TEL：045-442-0500 FAX：045-442-0501  
URL：<https://www.kernelcomputer.co.jp>

## [製品概要]

KBarcode\_libは、イメージデータ上のバーコードを読み取るライブラリです。  
バーコードがどこにあっても、また何個でも読み取ることができます。  
ユーザーアプリケーションにKBarcode\_libを結合すると、バーコード情報の取得を容易に行うことができます。  
ライブラリに渡す入力情報は、メモリ上のイメージデータの情報となります。  
また、取得したバーコード情報はCSVなどのテキスト情報としてファイルに出力することも可能です。



## 開発環境と動作環境

### [Windows版]

#### <開発環境>

- ・ Microsoft Visual Studio 2005 (Visual C++, Visual Basic, Visual C#)
- ・ Microsoft Visual Studio 2008 (Visual C++, Visual Basic, Visual C#)
- ・ Microsoft Visual Studio 2010 (Visual C++, Visual Basic, Visual C#)
- ・ Microsoft Visual Studio 2012 (Visual C++, Visual Basic, Visual C#)
- ・ Microsoft Visual Studio 2013 (Visual C++, Visual Basic, Visual C#)
- ・ Microsoft Visual Studio 2015 (Visual C++, Visual Basic, Visual C#)
- ・ Microsoft Visual Studio 2017 (Visual C++, Visual Basic, Visual C#)
- ・ Microsoft Visual Studio 2019 (Visual C++, Visual Basic, Visual C#)
- ・ Microsoft Visual Studio 2022 (Visual C++, Visual Basic, Visual C#)

#### <動作環境>

- ・ Windows 7
- ・ Windows 8
- ・ Windows 8.1
- ・ Windows 10
- ・ Windows 11
- ・ Windows Server 2008
- ・ Windows Server 2008 R2
- ・ Windows Server 2012
- ・ Windows Server 2012 R2
- ・ Windows Server 2016
- ・ Windows Server 2019
- ・ Windows Server 2022

### [Solaris版]

#### <開発環境(コンパイラ)>

- ・ Sun C 5.7以降

#### <動作環境>

- ・ Solaris9以降 (SPARC)

### [HP-UX版]

#### <開発環境(コンパイラ)>

- ・ HP aC++/ANSI C B3910B A.05.55以降

#### <動作環境>

- ・ HP-UX11i V2以降

### [AIX版]

#### <開発環境(コンパイラ)>

- ・ IBM XL C/C++ for AIX V11.1以降

#### <動作環境>

- ・ AIX6.1以降

### [Linux(x86系)版]

#### <開発環境(コンパイラ)>

- ・ GNU Compiler Collection(gcc) 3.2以降
- ・ GNU C Library(glibc) 2.2以降

#### <動作環境>

- ・ Linux(x86系)

※ WindowsはDLLでの提供、UNIXはスタティックライブラリでの提供、Linuxはスタティックライブラリと共有ライブラリでの提供となります。

## 特長

### 【出力結果について】

バーコード画像を含むイメージデータからバーコード情報を取得します。また、GSVなどのテキストとしてファイルに出力することも可能です。取得する情報は以下の項目から選択、組み合わせることができます。

- ・バーコードの内容
- ・バーコード種別(種類)
- ・位置情報(座標 x, y)
- ・バーコードの向き(90度単位の角度: 0, 90, 180, 270)
- など

### 【カラーモードについて】

イメージデータのカラーモードについては、以下のものに対応しています。

インデックスカラー、RGBフルカラー、3ビットRGBカラー、グレースケール、2値、CMYKカラー、16ビットグレースケール

### 【認識について】

- ・複数種類、複数個のバーコードを1度の処理で認識することが可能です。
- ・バーコードの読み取り範囲を指定することができます。複数バーコードが存在する場合でも、特定の範囲にあるバーコードのみ認識することができます。
- ・1次元バーコードは、0/90/180/270度などの大まかな方向を指定して読み取ることができます。スキャナで読み取る際に傾いたイメージにも対応できます。
- ・認識するバーコードの種類、桁数、文字種などを制限することによって、効果的に誤認識を防止することができます。
- ・設定により、認識の障害となるディザノイズなどを除去することができます。

### 【特殊な機能】

- ・分割QRコードに対応しています。
- ・読み取ったバーコードを任意の文字で分割し、複数のバーコードとして扱うことができます。
- ・「空白のページ」や「バーコードのないページ」を検出することもできます。

### 【入力情報としてBitmapオブジェクトを使う】

入力情報としてイメージデータのアドレス、幅、高さ、カラーモードを使用する以外に、Bitmapオブジェクトを使用することもできます。ただしこの場合は、.NETFramework4.0以上が必要となります。

### 【認識可能なバーコードの種類】

バーコードの種類	備考
QRコード	モデル2のすべての型番に対応。 分割QRコードの連結も可能。 マイクロQRコードにも対応。
Code39	桁数は任意。 フルアスキーモードにも対応。
Code128	GS1-128(UCG/EAN128)にも対応。
JAN	8桁および13桁のJANコード。 EANおよびUPC-Aも認識可能。
ITF	桁数は任意。
NW-7	多くの方式のチェックデジットに対応。
郵便カスタマバーコード	郵便物に印刷される、住所を表すバーコード。
GS1 DataBar (RSS)	すべてのバージョンに対応。
GS1合成シンボル	任意のリニアコンポーネントに対応。
PDF417	PDF417バーコード。
DataMatrix	DataMatrix(ECC200)。
その他	空白ページや「バーコードのないページ」の検出が可能。

## 主な実行オプション

KDbarcode\_libは、変換時に以下のような実行オプションを指定することができます。  
複数のオプションを記述する場合、その順番は処理の内容に影響しません。

-M : 出力ファイルの最後に情報を追加出力します。  
-QR[file] : QRコードの詳細情報をファイルに出力します。  
-BSX[X] -BSY[Y] : バーコードの解析開始位置を指定します。  
-BSW[W] -BSH[H] : バーコードの解析範囲の大きさを指定します。  
-BTYP[typ] : 検出するバーコードの種類を指定します。  
-BDIR[dir] : 検出するバーコードの方向を指定します。

## 主なAPI

### ・認識処理の実行

```
int KDDBC_execute(KDDBC_Handle h, char *in_atr, char *out_file, char *out_atr, char *exec_opt)
```

KDDBC_Handle	h	入力イメージデータ情報(データ、幅、高さ、カラーモードなど)、 取得バーコード情報(内容、文字数、種類など)
char*	in_atr	入力属性ファイル名
char*	out_file	出力ファイル名
char*	out_atr	出力属性ファイル名
char*	exec_opt	実行オプション
int	return	リターンコード(0:正常終了, -1:エラー)

### ・イメージデータの先頭アドレスを設定

```
void KDDBC_setAddress(KDDBC_Handle h, unsigned char *address)
```

KDDBC_Handle	h	入力イメージデータ情報(ここにアドレスが設定される)、 取得バーコード情報
unsigned char*	address	イメージデータのアドレス

### ・検出したindex番目のバーコードのデータ内容を取得

```
int KDDBC_getData(KDDBC_Handle h, int index, int format, unsigned char *buffer)
```

KDDBC_Handle	h	入力イメージデータ情報、取得バーコード情報
int	index	バーコードのインデックス
int	format	データの書式(KDDBC_DATA_BIN : バイナリデータ、など)
unsigned char*	buffer	取得データ
int	return	リターンコード(0:正常終了, -1:エラー)

## 組み込みイメージ

```
{
  int n, i;
  unsigned char bardata[512];
  KDBC_Handle h;

  :
  :

  KDBC_setAddress(h, [データのアドレス]);
  KDBC_setHeight(h, [データの高さ]);
  KDBC_setWidth(h, [データの幅]);
  KDBC_setColorFormat(h, [データのカラーフォーマット]);
  A

  KDBC_execute(h, NULL, NULL, NULL, NULL);
  B

  n = KDBC_getCount(h);
  C

  for (i=0; i<n; i++) {
    KDBC_getData(h, i, KDBC_DATA_BIN, bardata);
    D
  }
}
```

- A …… 入力データの設定
- B …… バーコード認識処理の実行
- C …… 検出したバーコード数をnに取得
- D …… 検出したi番目のバーコードデータ内容をbardataに取得

## 属性ファイル

変換時のパラメータを記述するために、以下のような属性ファイルが用意されています。

- ・ ras\_in. atr 主に入力処理に関連したパラメータを記述します。
- ・ ras\_bar. atr バーコード認識のパラメータを記述します。

属性ファイルの内容は、以下のような形式のテキストファイルです。

```
キーワード = パラメータ
キーワード = パラメータ
...
```

- ・ 先頭が#の行はコメントになります。
- ・ 一つの項目を複数行にわたって記述することはできません。
- ・ 余分な空白や空行は無視されます。
- ・ キーワードには大文字/小文字の区別はありません。

入力属性ファイル(ras\_in. atr)の例:

```
COLOR_FORMAT = 1
RAS_WIDTH = 1000
RAS_HEIGHT = 1500
CHG_BK_WH = 0
```

COLOR_FORMAT	イメージデータのカラーモード(RGBフルカラー、モノクロ、など)
RAS_WIDTH	イメージデータの幅の指定(ピクセル数)
RAS_HEIGHT	イメージデータの高さの指定(ピクセル数)
CHG_BK_WH	色を反転するかどうか(0, 1)

出力属性ファイル(ras\_bar.atr)の例:

```

BAR_TYPE = CODE39, CODE128, QR
BAR_AREA = 0, 0, 0, 0
BAR_DIRECTION = 0, 90, 180, 270
BAR_UNDITHER = 0
BAR_INTERVAL = 10
THRESHOLD_MODE = 0
THRESHOLD_REF = 0
BAR_DIGIT =
BAR_ANK_NUM =
BAR_AMOUNT = 0, 2
CODE39_START = 0
CODE39_CHECK = 0
CODE39_FULLASCII = 0
ITF_CHECK = 1
NW7_START = 1
NW7_CHECK = 1
NW7_CHECK_TYPE = 2
INFO_FORMAT = %N, %S, %C, %X, %Y, %K, %R
FILENAME_MAX = 30
EXEC_CMD = copy C:\sample.tif %Q.tif
EXEC_MODE = 3
DETECT_ERROR = 0
BAR_SPLIT =
QR_CONCAT = 0

```

BAR_TYPE	検出するバーコードのタイプを指定 1次元: JAN, ITF, NW7, CODE39, CODE128, CUSTOMER, RSS 2次元: QR, PDF417, DM, GS 特殊: BLANK, NODATA, ANY, ALL
BAR_AREA	検出する領域の位置とサイズを指定
BAR_DIRECTION	検出する方向を指定 (0, 90, 180, 270) [度]
BAR_UNDITHER	ディザノイズの除去 (0, 1)
BAR_INTERVAL	平均化によるノイズの除去 (1~255)
THRESHOLD_MODE	2値化モードの指定 (0~2)
THRESHOLD_REF	2値化閾値の調整 (-255~255)
BAR_DIGIT	検出するバーコードの桁数の範囲を指定
BAR_ANK_NUM	検出するバーコードの各桁の文字種を指定
BAR_AMOUNT	検出するバーコードの個数を指定
CODE39_START	Code39のスタート/ストップキャラクタの取り扱いを指定 (0, 1)
CODE39_CHECK	Code39のチェックキャラクタの取り扱いを指定 (0, 1, 2)
CODE39_FULLASCII	Code39のフルアスキーモード指定 (0, 1)
ITF_CHECK	ITFのチェックキャラクタの取り扱いを指定 (0, 1, 2)
NW7_START	NW-7のスタート/ストップキャラクタの取り扱いを指定 (0, 1)
NW7_CHECK	NW-7のチェックキャラクタの取り扱いを指定 (0, 1, 2)
NW7_CHECK_TYPE	NW-7のチェックキャラクタの種類を指定 (0~9)
INFO_FORMAT	出力するバーコード情報を指定 (通常の文字列と、シーケンスでの置き換えを組み合わせで指定可能) %D: バーコード内容 (制御文字などはエスケープ表現に置換) %C: バーコード内容 (CSV形式用に加工) %U: バーコード内容 (URLエンコード処理) %Q: バーコード内容 (ファイル名用に加工) %B: バーコード内容 (バイナリデータとして別のファイルに出力) %S: 読み取ったバーコードの文字数 %N: バーコードの連番 %X: x方向位置 %Y: y方向位置 %K: バーコードの種類 %R: バーコードの向き
FILENAME_MAX	%Qでの最大文字数制限
EXEC_CMD	検出したバーコード情報を引数として任意のコマンドを実行
EXEC_MODE	EXEC_CMDを実行するタイミングの指定 0: バーコード毎 1: バーコードのあるページ毎 2: すべてのページ毎 3: バーコードのあるファイル毎
DETECT_ERROR	不正なバーコードの検出設定 (0, 1)
BAR_SPLIT	バーコード情報の分割指定 (区切り文字)
QR_CONCAT	分割QRコードの連結指定 (0~6)

## 価格(税抜き)

- ・ 本体ライセンス  
35万円
- ・ 年間保守  
12万円
- ・ 配布ライセンス  
5万円 (複数ライセンスを購入する場合の費用については、お問い合わせください。)

※ライブラリ製品のライセンスは、本体ライセンスと配布ライセンスの2種類があります。  
本体ライセンスは、1台のマシンでライブラリを組み込んだモジュールの作成と実行ができるライセンスです。  
配布ライセンスは、1台のマシンでライブラリを組み込んだモジュールの実行ができるライセンスです。