

PHP 版 バーコード作成ライブラリ

Barcode.php

説明書

Version 2.1

2017 年 1 月

Pao@Office

はじめに

Barcode.php は、PHP 環境下で動作する、バーコード作成ツール(クラス群)の総称です。

Barcode.php は、次のことを念頭において開発いたしました。

1. 精密なこと

単なるバーコードリーダでの検査でなく、RJS のレーザーインスペクター Model L2000 というバーコード検査機にて細かくバーコードの精度を検査しております。それにより、従来の他社のバーコード作成ツールに比べても精密なバーコードを作成することが可能です。

バーコード全体の幅を指定する方法以外にも、バーの最小幅を指定することにより、縮小することなく直接バーコードを描画し、より精度の高いバーコードを作成することが可能です。

2. 用途が様々

Barcode.php が、GD オブジェクトに対してバーコードを描画する仕組みになつておりますので、様々な用途に利用することが可能になっております。

3. 使いやすいこと

わかりやすいクラスのインターフェイスになっております。

後の使用例でも書かれておりますが、2~3 Step のロジックでバーコードの印刷等を行うことができます。

4. 軽いこと

何と言っても軽さが命です。Barcode.php を利用してバーコード作成を行う場合、Barcode.php 自体がシステムに与える負荷は微小です。ほんの数MBのメモリ上で動作します。

Barcode.php をご利用していただく皆さんが、PHP 環境でのバーコードの生成(印刷)プログラムの作成作業に、楽しさを感じていただければ幸いです。

平成 22 年 3 月 作者

目次

1.	Barcode.php の動作環境・インストール方法	1
1-1.	動作環境	1
1-2.	インストール方法	1
2.	Barcode.php の機能	2
2-1.	機能概要	2
2-1-1.	一次元バーコードの種類	3
2-1-2.	二次元バーコードの種類	5
2-1-3.	GS1 Databar (RSS) の種類	6
2-2.	一次元バーコード作成クラスの機能	7
2-3.	郵便カスタマバーコード作成クラスの機能	8
2-4.	QR コード作成クラスの機能	9
2-5.	DataMatrix 作成クラスの機能	10
2-7.	PDF417 作成クラスの機能	11
2-8.	GS1 データバー(RSS)作成クラスの機能	12
3.	アプリケーションプログラムから Barcode.php の使用方法	14
3-1.	クラス仕様	14
3-1-1.	概要	14
3-1-2.	一次元バーコードクラスメンバ	15
3-1-2-1.	コンストラクタ	15
3-1-2-2.	メソッド(関数)	16
3-1-2-3.	GS1-128(UCC/EAN128)コンビニバーコードメソッド	17
3-1-2-3.	プロパティ	18
3-1-3.	GS1 データバー(RSS) クラスメンバ	20
3-1-3-1.	コンストラクタ	20
3-1-2-2.	メソッド(関数)	21
3-1-2-3.	プロパティ	22
3-1-4.	郵便カスタマバーコードクラスメンバ	23
3-1-4-1.	コンストラクタ	23
3-1-4-2.	メソッド(関数)	23
3-1-5.	QR コードクラスメンバ	24
3-1-5-1.	コンストラクタ	24
3-1-5-2.	メソッド(関数)	25
3-1-5-3.	プロパティ	26
3-1-6.	DataMatrix クラスメンバ	27
3-1-6-1.	コンストラクタ	27
3-1-6-2.	メソッド(関数)	27
3-1-6-3.	プロパティ	28
3-1-7.	Pdf417 クラスメンバ	29
3-1-7-1.	コンストラクタ	29

3-1-7-2. メソッド(関数).....	29
3-1-7-3. プロパティ	30
3-2. 使用例・サンプルプログラム	31
4. 使用条件等.....	37
4-1. 試用版と製品版.....	37
4-2. 使用許諾	38
4-3. 代金支払い方法(ユーザ登録の方法).....	39

1. **Barcode.php の動作環境・インストール方法**

1-1. 動作環境

- PHP 5 / PHP6 / PHP7 / PHP8 が動作する環境であること。

1-2. インストール方法

以下の PHP ソースファイルをコピーしてお使いください。

○ 共通ルーチン → 必須

- Common.php

- CheckDigit.php

○ 各バーコード描画 → お使いのバーコードのみ必要

- Code39.php

- Code93.php

- Code128.php

- EAN128.php (GS1 128)

- ITF.php

- Jan13.php

- Jan8.php

- Matrix2of5.php

- NEC2of5.php

- NW7.php

- YubinCustomer.php

- QR.php

- DataMatrix.php + DataMatrix ディレクトリ

- Pdf417.php

- Databar14.php

- DatabarLimited.php

- DatabarExpanded.php

試用版：以下の URL より、VMWare Player イメージの OS(Linux)ごとダウンロードしてお試しください。

<http://www.pao.ac/barcode.php/>

Barcode.php のサンプルプログラムは、以下のいずれかのディレクトリにございます。

- /var/www/html/sample

- /usr/local/apache2/htdocs/sample

製品版：barcode.php(バージョン).zip を解凍後、任意のディレクトリにコピーしてお使いください。

プログラムからは、以下のように記述してお使いください。

```
require_once("Code39.php");
```

2. Barcode.php の機能

2-1. 機能概要

Barcode.php は、以下のバーコードの作成が可能です。

- (1) JAN-13(EAN-13)
- (2) JAN-8(EAN-8)
- (3) UPC-A
- (4) UPC-E
- (5) ITF(インターリーブド 2 of 5)
- (6) Matrix 2 of 5
- (7) NEC 2 of 5 (Coop 2 of 5)
- (8) NW-7(CODA-BAR)
- (9) CODE39
- (10) CODE93
- (11) CODE128
- (12) GS1-128 (UCC/EAN-128)
 - コンビニ向け標準料金代理収納用バーコード
 - 医療用 医薬品等のバーコード
 - 医療用 医療材料等のバーコード
 - 食肉標準物流バーコード「基本バーコード」
- (13) 郵便カスタマバーコード
- (14) GS1 Databar 標準型 (RSS-14) … ver 3.0 において追加
- (15) GS1 Databar 限定型 (RSS Limited) … ver 3.0 において追加
- (16) GS1 Databar 拡張型 (RSS Expanded) … ver 3.0 において追加
- (17) QR コード
- (18) 標準料金代理収納用バーコード(コンビニバーコード)
- (19) DataMatrix (GS1 DataMatrix)
- (20) PDF417

※郵便カスタマバーコード・GS1 Databar・二次元バーコード(QR / DataMatrix / PDF417)以外は、
以降総称して「一次元バーコード」と呼びます。

Barcode.php では、上記の各バーコードを作成するために、バーコードの種類ごとに全て別々のクラスとして利用することが可能となっております。

Barcode.php の各バーコード作成クラスは、バーコードを描画する draw メソッドを実装しています。Draw メソッドの戻り値は、バーコードが描画された GD ライブラリの画像(image)オブジェクトです。

2-1-1. 一次元バーコードの種類

次の種類のバーコードを出力できます。

1. JAN-13(EAN-13)	2. JAN-8(EAN-8)
数字 13 桁。12 桁指定時、13 桁目のチェックディジットを自動計算付与。	数字 8 桁。7 桁指定時、8 桁目のチェックディジットを自動計算付与。
 1 2 3 4 5 6 7 8 9 0 1 2 8	 1 2 3 4 5 6 7 0
3. UPC-A	4. UPC-E
数字 12 桁。11 桁指定時、12 桁目のチェックディジットを自動計算付与。	数字 7 桁。6 桁指定時、7 桁目のチェックディジットを自動計算付与。
 1 2 3 4 5 6 7 8 9 0 1 2	 0 1 2 3 4 5 6 5
5. ITF(インターリード 2 of 5)	6. Matrix 2 of 5
使用可能文字：数字  1 2 3 4 5 6 7 8 9 0	使用可能文字：数字  1 2 3 4 5 6 7 8 9 0
7. NEC 2 of 5 (Coop 2 of 5)	8. NW-7(CODA-BAR)
使用可能文字：数字  1 2 3 4 5 6 7 8 9 0	使用可能文字： ABCD. +:/-\$-0123456789  C 1 2 3 4 5 6 7 8 9 0 C 1 文字目 ABC いずれか入力した文字がスタート・ストップキャラクタとなる。入力がない場合 C が既定値。 パラメータ： スタート・ストップキャラクタ表示／非表示

9. CODE39	10. CODE93
使用可能文字： 1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ-. *\$/+%  * 1 2 3 4 5 6 7 8 9 0 *	使用可能文字： 1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ-. *\$/+%  1 2 3 4 5 6 7 8 9 0
パラメータ： スタート・ストップキャラクタ表示／非表示	12. GS1-128 (UCC/EAN-128) <ul style="list-style-type: none"> - コンビニ向け標準料金代理収納用バーコード - 医療用 医薬品等のバーコード - 医療用 医療材料等のバーコード - 食肉標準物流バーコード「基本バーコード」  (01)04512345670016(21)1
パラメータ： どのコードパターンを使用するか。 AUTO / CODE_A / CODE_B / CODE_C 既定値は、AUTO。AUTOは、数時4文字続ければCODE Aにコードチェンジするなど、バーコードが一番小さくなるように自動調整する。	コンビニバーコード 入力： {FNC1}9191234500000000000004520875004013100295006  (91)912345-000000000000000045208750 040131-0-029500-6
13. 郵便カスタマバーコード	
郵便番号＋住所の英数字部分のみ入力 例：27500263-29-2-401 パラメータ： 8~11.5 ポイント(大きさ) 	

2-1-2. 二次元バーコードの種類

次の種類のバーコードを出力できます。

1. QR コード

パラメータ : (既定値はアンダーバー)

- (1) エラー訂正レベル : L / M / Q / H
- (2) 文字種 : 数字 / 大文字英数字 / 8 ビットバイトデータ(漢字含む)
- (3) バージョン : 1~40 5
- (4) 全角文字エンコード : "shift-jis" / "utf8" / etc..



2. DataMatrix (GS1 DataMatrix)

パラメータ : (既定値はアンダーバー)

- (1) シンボルコードサイズ : 自動 / 10x10~144x144 / 8x18~16x48 等
- (2) 全角文字エンコード : "utf8" / "shift-jis" / etc..

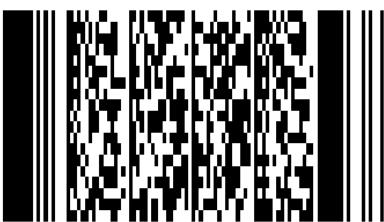


3. PDF417

パラメータ : (既定値はアンダーバー)

- (1) エラー訂正レベル自動決定 : する / しない
- (2) エラー訂正レベル : 1~8 2
- (3) アスペクト比 : 0.5 / 1.0 / 2.0 等
- (4) 列・行数決定方法 :

アスペクト比より自動決定 / 列数指定 / 行数指定 / 列数行数指定
- (5) 指定列数 : 1~30 5
- (6) 指定行数 : 3~90 5
- (7) 全角文字エンコード : "shift-jis" / "utf8" / etc..



2-1-3. GS1 Databar (RSS) の種類

次の種類のバーコードを出力できます。

1. GS1 データバーOmni-directional / 表示桁数：数字 14 桁(GTIN)	
Omni-directional(標準型)  (01) 14912345678901	Stacked (2 層型)  (01) 14912345678901
高さが狭い Truncated(カット型)という種類もございますが、この標準型の高さを調整して出力してください。	
2. GS1 データバーLimited / 表示桁数：数字 14 桁(GTIN)	
Limited(限定型)  (01) 12345678901231	
3. GS1 データバーExpanded / 表示桁数：最大数字 74 桁または英字 41 文字	
Expanded(拡張一層型)  (01) 00012345678905 (10) ABC123	Stacked Expanded(拡張多層型)  (01) 00012345678905 (10) ABC123

2-2. 一次元バーコード作成クラスの機能

Barcode.php の各一次元バーコード作成クラスは、以下の機能を有します。

(1) バーコードの描画

コード、横方向の最少描画ドット数、バーコードの高さを指定してバーコードを描画します。

少し特殊な GS1-128(UCC/EAN-128)において、AI(アプリケーション識別子)挿入方法は 2 通りございます。

(1) 可変長項目(データブロック)の後の AI には、FNC1 を挿入

⇒これまで通り "{FNC1}" を付ける。例 : "{FNC1}21" のようにコードを指定

(2) 固定長項目(データブロック)の後の AI には、固定長のため目印の FNC1 は不要

⇒新しく追加した "{AI}" を付ける。例 : "{AI}21" のようにコードを指定

"{AI}" を指定して FNC1 を挿入しない場合も、カッコ()付コード文字は出力されます。例えば入力コードに "{AI}21" を指定した場合、添え字には(21)と出力されます。

例) (01)04512345670016(21)1 ⇒(01)の前には FNC1 を挿入し(21)の前には挿入しない。

コード指定方法 → "{FNC1}0104512345670016{AI}211"

(2) 添字の描画

バーコードの下にコードの文字列を描画します(既定値)。

プロパティの設定で描画をしないようにすることも可能です。

添字のフォントをプロパティで指定することも可能です。

CODE39/NW-7(CODABAR) のみスタート・ストップキャラクタを印字するかどうかをプロパティで指定することができます。既定値は印字しません。

2-3. 郵便カスタマバーコード作成クラスの機能

Barcode.php の郵便カスタマバーコード作成クラスは、以下の機能を有します。

(1) バーコードの描画

コード、サイズを指定してバーコードを描画します。

コードの表記は・・・

[郵便番号の数字部分 7 桁]+[郵便番号では不明部分の住所の英数字を「-」区切り]
で、指定してください。

例) 〒116-0013 東京都荒川区西日暮里五丁目 37 番 5 号スタートアップオフィス A-207 号室

コード指定方法→「11600135-37-5-A-207」

※詳しくは、旧郵政省の web ページにマニュアルがございますのでご覧になってください。

2-4. QR コード作成クラスの機能

Barcode.php の QR コード作成クラスは、以下の機能を有します。

バーコードの描画

コードと最少描画ドット数、又は、コードと幅を指定して描画することが可能です。

プロパティで以下の項目を指定することが必要です。

- ・ バージョン(1~40)
- ・ エラー訂正レベル(L,M,Q,H)

2-5. DataMatrix 作成クラスの機能

Barcode.php の DataMatrix 作成クラスは、以下の機能を有します。

DataMatrix の描画

コードと最少描画ドット数、又は、コードと幅を指定して描画することが可能です。

プロパティで以下の項目を指定することができます。

- シンボルコードサイズ 既定値：`DxCodeSize`.`DxSzAuto`
(enum) `DxCodeSize`.

<code>DxSzRectAuto,</code>	<code>DxSz24x24,</code>	<code>DxSz88x88,</code>
<code>DxSzAuto,</code>	<code>DxSz26x26,</code>	<code>DxSz96x96,</code>
<code>DxSzShapeAuto,</code>	<code>DxSz32x32,</code>	<code>DxSz104x104,</code>
	<code>DxSz36x36,</code>	<code>DxSz120x120,</code>
<code>DxSz10x10,</code>	<code>DxSz40x40,</code>	<code>DxSz132x132,</code>
<code>DxSz12x12,</code>	<code>DxSz44x44,</code>	<code>DxSz144x144,</code>
<code>DxSz14x14,</code>	<code>DxSz48x48,</code>	
<code>DxSz16x16,</code>	<code>DxSz52x52,</code>	<code>DxSz12x36,</code>
<code>DxSz18x18,</code>	<code>DxSz64x64,</code>	<code>DxSz16x36,</code>
<code>DxSz20x20,</code>	<code>DxSz72x72,</code>	<code>DxSz16x48</code>
<code>DxSz22x22,</code>	<code>DxSz80x80,</code>	

- 全角エンコーディング ("utf-8", "shift-jis", etc..) 既定値：utf-8

2-7. PDF417 作成クラスの機能

Barcode.php の PDF417 作成クラスは、以下の機能を有します。

PDF417 の描画

コードと最少描画ドット数を指定して描画することが可能です。

プロパティで以下の項目を指定することができます。

- ・ サイズ種別 …データ列数・行数決定方法
 - 自動 `Pdf417.SIZE_KIND.AUTO` (既定値)
 - データ列数指定 `Pdf417.SIZE_KIND.COLUMNS`
 - データ行数指定 `Pdf417.SIZE_KIND.ROWS`
 - データ列数・行数指定 `Pdf417.SIZE_KIND.COLUMNS_AND_ROWS`
 - ・ 行数 …出力データ行数指定
 - サイズ種別が、
出力行数指定の場合=(自動サイズでない・列数指定でない場合)有効
3～90 既定値 : 5
 - ・ 列数 …出力データカラム数指定
 - サイズ種別が、
出力列数指定の場合=(自動サイズでない・行数指定でない場合)有効
1～30
 - ・ エラーレベル
 - 0～8 既定値 : 2
 - ・ エラーレベル自動決定
 - 自動でエラー訂正レベルを決定(する・しない) 既定値 : true(する)
 - ・ 縦横アクセプト比
 - シンボルの縦横比、シンボル アスペクト レシオ (比)
既定値 : 0.5
 - ・ 全角エンコーディング ("utf-8", "shift-jis", etc..)
既定値 : utf-8

2-8. GS1 データバー(RSS)作成クラスの機能

Barcode.php の各 GS1 データバー(RSS)作成クラスは、以下の機能を有します。

(1) バーコードの描画

コード、横方向の最少描画ドット数、バーコードの高さを指定してバーコードを描画します。

(2) バーコードの高さ指定とその出力結果

高さ指定とその出力結果について Barcode.php 独自の仕様がございます。

次の表にまとめました。

1. GS1 データバーOmni-directional / 表示桁数：数字 14 桁(GTIN)

Omni-directional (標準型)

一次元バーコードなので、幅高さは通常の一次元バーコードと同様に自在です。



Stacked (2 層型)

バーコード上段・センター部分・下段の高さ割合が決まっているため、高さの指定があっても割合をキープして出力することが基本です。

Barcode.php では、DrawDirect / DrawDelicate メソッドを使用した時に、センター部分を含む 3 段の割合を確実にキープします。

Draw メソッドの場合は、センターの小さな正方形群の各正方形割合はキープしながら、ユーザより引数で指定された高さにより上段・下段の割合に従って上段下段の高さを決定します。従って Draw メソッドでは、本来決まっているセンター部分を含む 3 段の高さ割合は保たれません。指定された高さにより、上・下段が決められている割合を保ちながら伸び縮みします。



Stacked Omni-directional (標準 2 層型)

バーコードの上段・下段の割合は同一で、センター部分(小さな正方形 3 段)がある仕様のバーコードです。従ってセンター部分の高さを固定し、ユーザより引数で指定された高さにより、上・下段がその割合によって同じ高さで伸び縮みします。



2. GS1 データバーLimited / 表示桁数：数字 14 桁(GTIN)

Limited (限定型)

一次元バーコードなので、幅高さは通常の一次元バーコードと同様に自在です。



3. GS1 データバーExpanded / 表示桁数：最大数字 74 桁または英字 41 文字

Expanded(拡張一層型)

一次元バーコードなので、幅高さは通常の一次元バーコードと同様に自在です。



Stacked Expanded(拡張多層型)

このバーコードは、各段(層)の割合は同一な仕様です。段と段(層と層)間の小さな 3 段の正方形の高さを固定し、ユーザより引数で指定された高さにより、各段(層)がその割合によって同じ高さで伸び縮みします。



(3) 拡張型の AI 識別子(ファンクションコード)について

決められている AI 識別子(ファンクションコード)は、何も指定しなくとも、Barcode.php がコード体系から判断し自動的に挿入します。

ただし、任意で AI 識別子を挿入する場合は、{AI}を入力してください。

例：(01)商品識別コード+任意の AI

0100012345678905 {AI} 10ABC123 ⇒ (01) 00012345678905 (10) ABC12

(4) 添字の描画

バーコードの下にコードの文字列自体を描画します(既定値)。プロパティの設定で描画をしないようにすることも可能です。

添字のフォントをプロパティで指定することも可能です。

3. アプリケーションプログラムから `Barcode.php` の使用方法

3-1. クラス仕様

3-1-1. 概要

`Barcode.php` は、以下のそれぞれバーコードごとに独立したクラスで構成されております。

`Jan13`

`Jan8`

`UpcA`

`UpcE`

`ITF`

`Matrix2of5`

`NW7`

`Code39`

`Code128`

`EAN128`

`GS1_128` …(旧名)`EAN128` を継承

`YubinCustomer`

`QRCode`

`DataMatrix`

`Pdf417`

一次元バーコードのクラスは基本的に同一名のプロパティやメソッドといったメンバを所有し、それらの機能も基本的に同一です。

そこで以降の各メンバの説明では、

- 一次元バーコードのクラスと
 - `YubinCustomer`(郵便カスタマバーコードクラス)と、
 - 二次元バーコード(`QRCode` / `DataMatrix` / `Pdf417`) のクラス
- の 3 つに分けてご説明いたします。

3-1-2. 一次元バーコードクラスメンバ

3-1-2-1. コンストラクタ

初期処理を行う。

バーコードの種類別に以下のインターフェイスが存在します。

- (1) **JAN13()**
- (2) **JAN8()**
- (3) **UpcA()**
- (4) **UpcE()**
- (5) **ITF()**
- (6) **Matrix2of5()**
- (7) **NEC2of5()**
- (8) **NW7()**
- (9) **Code39()**
- (10) **Code98()**
- (11) **Code128()**
- (12) **EAN128()**
- (13) **GS1_128()** … (旧名)EAN128 を継承

・引数

なし。

3-1-2-2. メソッド(関数)

(1) `function draw($code, $minWidthDot, $height)`

バーコードの描画を行います。

・引数

① `$code`

描画を行うバーコードのコードを文字列で指定します。

少し特殊な GS1-128(UCC/EAN128)において、AI(アプリケーション識別子)挿入方法は 2 通りございます。

a) 可変長項目(データブロック)の後の AI には、

FNC1 を挿入

⇒これまで通り "{FNC1}" を付ける。例 : "{FNC1}21" のようにコードを指定

b) 固定長項目(データブロック)の後の AI には、固定長のため目印の

FNC1 は不要

⇒新しく追加した "{AI}" を付ける。例 : "{AI}21" のようにコードを指定 "{AI}" を指定して FNC1 を挿入しない場合も、カッコ()付コード文字は出力されます。例えば入力コードに "{AI}21" を指定した場合、添え字には(21)と出力されます。

例) (01)04512345670016(21)1

↑(01)の前には FNC1 を挿入し(21)の前には挿入しない。

コード指定方法 → "{FNC1}0104512345670016{AI}211"

② `$minWidthDot`

横方向の最少描画ドット数指定します。(単位はドット)

③ `$height`

バーコードのバーの高さを指定します。(単位はドット)

・戻り値

バーコードの画像(image)

3-1-2-3. GS1-128(UCC/EAN128)コンビニバーコードメソッド

(1) `function drawConvenience($code, $minWidthDot, $height)`

コンビニバーコードの描画を行います。

・引数

① `$code`

描画を行うバーコードのコードを文字列で指定します。

少し特殊な GS1-128(UCC/EAN128)において、AI(アプリケーション識別子)挿入方法は 2 通りございます。

a) 可変長項目(データブロック)の後の AI には、

FNC1 を挿入

⇒これまで通り "{FNC1}" を付ける。例 : "{FNC1}21" のようにコードを指定

b) 固定長項目(データブロック)の後の AI には、固定長のため目印の

FNC1 は不要

⇒新しく追加した "{AI}" を付ける。例 : "{AI}21" のようにコードを指定 "{AI}" を指定して FNC1 を挿入しない場合も、カッコ()付コード文字は出力されます。例えば入力コードに "{AI}21" を指定した場合、添え字には(21)と出力されます。

例) (01)04512345670016(21)1

↑ (01)の前には FNC1 を挿入し(21)の前には挿入しない。

コード指定方法 → "{FNC1}0104512345670016{AI}211"

② `$minWidthDot`

横方向の最少描画ドット数指定します。(単位はドット)

③ `$height`

バーコードのバーの高さを指定します。(単位はドット)

・戻り値

バーコードの画像(image)

3-1-2-3. プロパティ

(1) **\$TextWrite**

true : 添字の描画を行う。(既定値)

false : 添字を描画しない。

(2) **\$FontName**

添字のフォント名。

既定値は、”MS ゴシック”

(3) **\$FontSize**

添字のフォントサイズ。

既定値は、10。

(4) **\$KurobarChousei** (2013/03/07 ver 1.2.1 追加プロパティ)

黒バーの幅をドット単位で調整できます。

プリンタによって PDF から印刷すると黒バーが太すぎて読み取れないようなケースに使用します。

既定値は、0。

(使用例ソースコード)

```
// ↓↓↓黒バーを1ドット細くします。↓↓↓

// ただし、全体のバーコードのサイズを大きくしないと
// 1ドット細くすると細くなりすぎてしまいます。
$barcode->KuroBarCousei = -1;
// そこで、バーコードの横幅を4倍にします。
// 具体的には、バーコードを描画する最小ドット幅を
// 1ドットから4ドットへ変更します。
$barcode->minWidthDot = 4;
// できあがったバーコードの画像ファイルが大きくて
// その後、指定サイズでPDFにその画像を読み込むため問題ありません。
```

(5) **\$dispStartStopCode**

Code39/NW7 のみ使用可能なプロパティ

true : スタート／ストップコードの描画を行う。

false : スタート／ストップコードを描画しない。(既定値)

(6) **\$outputCode**

Code39/NW7 のみ使用可能なプロパティ

添字に出力するコード(スタート／ストップコード含む)。

出力プロパティなので、指定する必要はない。

◇次のプロパティは、CODE128 / GS1_128(EAN128)のみで使用するプロパティです。

(1) **\$CodeABC (既定値 : CodeSet128.AUTO)**

CODE128 のコードセットには、次の種類があります。

CODE-A: フルアスキー

CODE-B: 1 桁の アルファ・ニューメリック

CODE-C: 2 桁の数字

本プロパティでは、次の指定が可能です。

CodeSet128 クラスの CONST 値

const AUTO = 0	Barcode.net が自動でコードセットを組み合わせ最小幅のバーコードとする。
cons CODE_A = 1	CODE-A
const CODE_B = 2	CODE-B
const CODE_C = 3	CODE-C

3-1-3. GS1 データバー (RSS) クラスメンバ

GS1 データバーのクラスのインターフェイス(コンストラクタ・メソッド・プロパティ)は、基本的に一次元バーコードのものと変わりません。特別な機能のプロパティのみ追加されております。

従って、メソッドやプロパティの説明は、冗長にならない目的のためにも割愛させていただいている部分がございます。

3-1-3-1. コンストラクタ

初期処理を行う。

GS1 データバーの種類別に以下のインターフェイスが存在します。

- (1) **Databar140**
- (2) **DatabarLimited()**
- (3) **DatabarExpanded()**

・引数

なし。

3-1-2-2. メソッド(関数)

(1) **function draw(\$code, \$minWidthDot, \$height)**

GS1 データバーの描画を行います。

・引数

① **\$code**

描画を行う GS1 データバーのコードを文字列で指定します。

《拡張型の AI 識別子(ファンクションコード)について》

決められている AI 識別子(ファンクションコード)は、何も指定しなくとも、コード体系から自動的に挿入されます。

ただし、任意で AI 識別子を挿入する場合は、{AI}を入力してください。

例：(01)商品識別コード+任意の AI

0100012345678905 {AI} 10ABC123 ⇒ (01)00012345678905 (10)ABC12

② **\$minWidthDot**

横方向の最少描画ドット数指定します。(単位はドット)

③ **\$height**

GS1 データバーのバーの高さを指定します。(単位はドット)

・戻り値

GS1 データバーの画像(image)

3-1-2-3. プロパティ

(1) **\$TextWrite**

添字の描画を行う。(既定値 : true)

(2) **\$FontName**

添字のフォント名。(既定値は : "MS ゴシック")

(3) **\$FontSize**

添字のフォントサイズ。(既定値 : 10)

(4) **\$KurobarChousei**

黒バーの幅をドット単位で調整できます。(既定値 : 0)

(5) **\$outputCode**

Code39/NW7 のみ使用可能なプロパティ

添字に出力するコード(スタート／ストップコード含む)。

出力プロパティなので、指定する必要はない。

◇次のプロパティは、GS1 データバー(RSS)のみで使用するものです。

(1) **\$SymbolType**

GS1 データバーのタイプです。

限定型では使用しません。標準型・拡張型で次のように定義されています。

出力するデータバーのタイプを標準型・拡張型、各々で指定します。

GS1 DataBar RSS 14 のタイプ	
class Databar14	
const OMNIDIRECTIONAL = 0	標準型
const STACKED = 1	二層型
const STACKED_OMNIDIRECTIONAL = 2	標準二層型

GS1 DataBar RSS Expanded のタイプ	
class DatabarExpanded	
const UNSTACKED = 0	一層型
const STACKED = 1	多層型

3-1-4. 郵便カスタマバーコードクラスメンバ

3-1-4-1. コンストラクタ

初期処理を行う。

YubinCustomer()

・引数

なし。

3-1-4-2. メソッド(関数)

(1) **function draw(\$code, \$size)**

郵便カスタマバーコードの描画を行います。

・引数

① **\$code**

描画を行うバーコードのコードを文字列で指定します。

コードは・・・

[郵便番号の数字部分 7 桁]+[郵便番号では不明部分の住所の英数字を「-」区切り]で、指定してください。

例) 〒116-0013 東京都荒川区西日暮里五丁目 37 番 5 号スタートアップオフィスA-207 号室
コード指定方法→「11600135-37-5-A-207」

② **\$size**

バーコードの大きさを指定します。

・戻り値

バーコードの画像(image)

3-1-5. QR コードクラスメンバ

3-1-5-1. コンストラクタ

初期処理を行う。

QRCode()

- ・引数

なし。

3-1-5-2. メソッド(関数)

(1) `function draw_base($code)`

ベースとなる一番小さな QR コードを描画します。

- 引数

① `$code`

描画を行う QR コードのコードを文字列で指定します。

- 戻り値

バーコードの画像(image)

(2) `function draw_by_size($code, $size)`

サイズ(1,2,4,8,16)を指定して読み取り精度の高い QR コードを描画します。

`draw_by_width` に比べ画像の拡縮を行わないので精度が高くなります。

- 引数

① `$code`

描画を行う QR コードのコードを文字列で指定します。

② `$size`

1,2,4,8,16 のいずれかを指定

- 戻り値

バーコードの画像(image)

(3) `function draw_by_width($code, $width)`

指定された幅に伸縮した QR コードを描画します。

`draw_by_size` に比べ、理論上、読み取り精度は低下する可能性があります。

- 引数

① `$code`

描画を行う QR コードのコードを文字列で指定します。

② `$width`

バーコードの幅 (単位: ドット)

- 戻り値

バーコードの画像(image)

3-1-5-3. プロパティ

(1) **\$version**

バージョン 1~40 を指定。

既定値は、5。

(2) **\$error_level**

エラー訂正レベル : L・M・Q・H のいずれかを指定。

既定値は、M。

3-1-6. DataMatrix クラスメンバ

3-1-6-1. コンストラクタ

初期処理を行う。

DataMatrix()

- ・引数

なし。

3-1-6-2. メソッド(関数)

(1) **function draw_by_size(\$code, \$size)**

サイズ(1~)を指定して読み取り精度の高い DataMatrix を描画します。

draw_by_width に比べ画像の拡縮を行わないため精度が高くなります。

- ・引数

① **\$code**

描画を行う DataMatrix のコードを文字列で指定します。

② **\$size**

サイズを指定(1~)

- ・戻り値

バーコードの画像(image)

(2) **function draw_by_width(\$code, \$width)**

指定された幅に伸縮した DataMatrix を描画します。

draw_by_size に比べ、理論上、読み取り精度は低下する可能性はあります。

- ・引数

① **\$code**

描画を行う DataMatrix のコードを文字列で指定します。

② **\$width**

バーコードの幅 (単位: ドット)

- ・戻り値

バーコードの画像(image)

3-1-6-3. プロパティ

(1) **\$CodeSize**

シンボルコードサイズ

既定値 : DxCodeSize::DxSzAuto(自動)

`class DxCodeSize.`

DxSzRectAuto,	DxSz24x24,	DxSz88x88,
DxSzAuto,	DxSz26x26,	DxSz96x96,
DxSzShapeAuto,	DxSz32x32,	DxSz104x104,
	DxSz36x36,	DxSz120x120,
DxSz10x10,	DxSz40x40,	DxSz132x132,
DxSz12x12,	DxSz44x44,	DxSz144x144,
DxSz14x14,	DxSz48x48,	
DxSz16x16,	DxSz52x52,	DxSz12x36,
DxSz18x18,	DxSz64x64,	DxSz16x36,
DxSz20x20,	DxSz72x72,	DxSz16x48
DxSz22x22,	DxSz80x80,	

(2) **\$StringEncoding**

全角エンコーディング ("utf-8", "shift-jis", etc..)

既定値 : utf-8

3-1-7. Pdf417 クラスメンバ

3-1-7-1. コンストラクタ

初期処理を行う。

Pdf417()

・引数

なし。

3-1-7-2. メソッド(関数)

function draw (\$code, \$size)

バーコードの描画を行います。

バーコード全体の幅を指定するのではなく、バーを描画する最小単位のドット数を指定します。(1~)

・引数

① **\$code**

描画を行う PDF417 のコードを文字列で指定します。

② **\$size (\$minLinePitch)**

1~ サイズを指定

・戻り値

バーコードの画像(image)

3-1-7-3. プロパティ

(1) **`$SizeKind = SIZE_KIND_AUTO;`**

データ列数・行数決定方法

既定値: SIZE_KIND_AUTO

```
/* データ列数・行数決定方法 */
/*! 自動サイズ(アスペクト比より決定) */
const SIZE_KIND_AUTO = 0;
/*! 指定列数(dataColumns)に従う */
const SIZE_KIND_COLUMNS = 1;
/*! 指定行数(dataRows)に従う */
const SIZE_KIND_ROWS = 2;
/*! 指定列数・行数両方に従う(長方形の最小値) */
const SIZE_KIND_COLUMNS_AND_ROWS = 3;
```

(2) **`$CodeRows = 3;`**

出力行数指定。

SizeKind が 出力行数指定の場合 = (自動サイズでない・列数指定でない場合) 有効

既定値: 3

(2) **`$CodeCols = 3;`**

出力データカラム数指定。

SizeKind が 出力列数指定の場合 = (自動サイズでない・行数指定でない場合) 有効

既定値: 3

(3) **`$ErrorLevel = 2;`**

エラー訂正レベル: 0~8 既定値は: 2。

(2) **`$UseAutoErrorLevel = true;`**

自動でエラー訂正レベルを決定(する・しない) 既定値は: する(true)

(3) **`$AspectRatio = 0.5;`**

縦横アクセプト比: 0.5 / 1.0 / 2.0 等を指定。 既定値: 0.5

(2) **`$StringEncoding = "utf-8"`**

全角エンコーディング ("utf-8", "shift-jis", etc..) 既定値: utf-8

3-2. 使用例・サンプルプログラム

以下のサイトにサンプルプログラムをご用意してございます。

<http://www.pao.ac/barcode.office/sample/>

ここではその一つ、「ブラウザにバーコードを出力する使用例」を記載しておきます。

【draw_browser_barcode.php(バーコード描画部分)】

```
<?php

$kind = (string)$_GET['kind'];
$code = (string)$_GET['code'];
$x = (float)$_GET['x'];
$y = (float)$_GET['y'];
$minWidthDot = (float)$_GET['minWidthDot'];
$height = (float)$_GET['height'];
$dispStartStopCode = (bool)$_GET['dispStartStopCode'];
$rotateAngle = (int)$_GET['rotateAngle'];

require_once("Code39.php");
require_once("Code128.php");
require_once("GS1_128.php");
require_once("ITF.php");
require_once("Jan13.php");
require_once("Jan8.php");
require_once("UpcA.php");
require_once("UpcE.php");
require_once("Matrix2of5.php");
require_once("NEC2of5.php");
require_once("NW7.php");
require_once("QR.php");
require_once("YubinCustomer.php");
$barcode = null;
$qr = null;

switch ($kind)
{
    case "code39":
        $barcode = new Code39();
        $barcode->dispStartStopCode = $dispStartStopCode;
        break;
}
```

```

        case "code128":
            $barcode = new Code128();
            break;
        case "gs1-128":
            $barcode = new GS1_128();
            break;
        case "itf":
            $barcode = new ITF();
            break;
        case "jan13":
            $barcode = new Jan13();
            break;
        case "jan8":
            $barcode = new Jan8();
            break;
        case "UPC-A":
            $barcode = new UpcA();
            break;
        case "UPC-E":
            $barcode = new UpcE();
            break;
        case "matrix2of5":
            $barcode = new Matrix2of5();
            break;
        case "nec2of5":
            $barcode = new NEC2of5();
            break;
        case "nw7":
            $barcode = new NW7();
            $barcode->dispStartStopCode = $dispStartStopCode;
            break;
        default :
            print("Nothing Barcode !!\n");
            break;
    }

    // バーコード描画 (PNGファイルへ)
    $img = $barcode->draw($code, $minWidthDot, $height);

    // 回転

```

```
if($rotateAngle != 0) {
    $img = imagerotate($img, $rotateAngle, 0);
}

$fileName = "gazou/gazou.png";
ImagePNG($img, $fileName);

$size = GetImageSize($fileName);
print("<html>n");
print("<head>n");
print("<title>Barcode Test Output</title>n");
print("</head>n");
print("<body>n");

print("<DIV id=Barcode style=position : absolute;width : ".$size[0]."px;height :
".$size[1]."px;z-index : 1;visibility : visible;left : ".$x."px;top : ".$y."px;>n"
    ."<IMG src='".$fileName."?".rand()."'" width=".$size[0]."
height=".$size[1]." border=0></DIV>n");

print("</body>n");
print("</html>n");
```

【browser_barcode.php(ユーザインターフェイス部分)】

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS" />
<title>BarCode.php バーコードをブラウザに出力するサンプル</title>
</head>

<body>
<form id="form1" name="form1" method="get" action="draw_browser_barcode.php">
<p>(BarCode.php)</p>
<p>PHPで、バーコードをブラウザに出力するサンプル (座標の単位はピクセル)</p>
<table width="436" border="0">
<tr>
<td width="141">Barcode Kind </td>
<td width="436">
<label></label>
<label>
<select name="kind" id="kind">
<option value="code39">Code39</option>
<option value="code128">Code128</option>
<option value="gs1-128">GS1_128</option>
<option value="itf">ITF</option>
<option value="jan13">Jan13</option>
<option value="jan8">Jan8</option>
<option value="UPC-A">UPC-A</option>
<option value="UPC-E">UPC-E</option>
<option value="matrix2of5">Matrix2of5</option>
<option value="nec2of5">NEC2of5</option>
<option value="nw7">NW7</option>
</select>
</label>
</td>
</tr>
<tr>
<td>コード</td>
<td>
<input name="code" type="text" id="code" value="1234567890" />
</td>

```

```
</tr>
<tr>
  <td>X</td>
  <td>
    <input name="x" type="text" id="x" value="20" />
  </td>
</tr>
<tr>
  <td>Y</td>
  <td>
    <input name="y" type="text" id="y" value="20" />
  </td>
</tr>
<tr>
  <td>幅</td>
  <td>
    <select name="minWidthDot" id="minWidthDot">
      <option value="1">通常 (最小描画ピッチ= 1)</option>
      <option value="2">広い (最小描画ピッチ= 2)</option>
      <option value="3">もっと広い(最小描画ピッチ= 3)</option>
      <option value="4">もっともっと広い(最小描画ピッチ= 4)</option>
    </select>
  </td>
</tr>
<tr>
  <td>高さ</td>
  <td>
    <input name="height" type="text" id="height" value="100" />
  </td>
</tr>
<tr>
  <td>スタート・ストップコード出力</td>
  <td>
    <select name="dispStartStopCode" id="dispStartStopCode">
      <option value="">しない□‘</option>
      <option value="true">する</option>
    </select>
  </td>
</tr>
<tr>
  <td>回転角度</td>
  <td>
```

```
<input name="rotateAngle" type="text" id="rotateAngle" value="0" />
</td>
</tr>
</table>

<input type="submit" name="submit" id="submit" value="Submit" />

</body>
</html>
<?php
```

4. 使用条件等

4-1. 試用版と製品版

Barcode.php は、ご購入前に試用することが可能です。

試用版は、以下の URL より OS のイメージごとダウンロードしてください。

<http://www.pao.ac/barcode.php/>

OS は、VMWare Player で動作するいくつかの Linux ディストリビューションをご用意しております。

サンプルプログラムは、各 OS 内の以下のディレクトリにご用意してございます。

- /var/www/html/sample
- /usr/local/apache2/htdocs/sample

試用版は、ソースコードを見えないようにするために、弊社独自仕様で暗号化しております。

試用版の制限は、バーコードに「SAMPLE」という文字が入ります。



QR コードのお試し版の制限は、指定したコードの先頭に半角の「9」という文字が入ります。

製品版は、Barcode.php を購入(ユーザ登録)していただいたお客様に、お客様専用の製品版ダウンロード URL をメールでお送りいたします。

バージョンアップの際は、WEB サイトにてお知らせいたします。

バージョンアップをご希望のお客様は、大変お手数でございますが、メールにてアップグレード依頼をしてください。

すぐにバージョンアップ版をメールにてお送りいたします。

4-2. 使用許諾

Barcode.php の使用について、Barcode.php の使用者(以下「利用者様」と称します)と有限会社パオ・アット・オフィス(以下「弊社」と称します)は、以下の各項目についての内容に同意するものとします。

1. Barcode.php の使用に関する使用許諾書

この使用許諾書は、利用者様がお使いのパソコンにおいて、Barcode.php を使用する場合に同意しなければならない契約書です。

2. 使用許諾書の同意

利用者様が Barcode.php を使用する時点で、本使用許諾書に同意されたものとします。同意されない場合は、Barcode.php を使用する事はできません。

3. ライセンス(使用権)の購入

利用者様が Barcode.php の製品版を使用して開発を行う場合には、1人または1台のコンピュータで Barcode.php を使用するにあたり、1ライセンスを購入する必要があります。同時に複数の人が複数のコンピュータで使用する場合はどちらか少ない方のライセンスを購入しなければなりません。

4. 著作権

Barcode.php 及の著作権は、いかなる場合においても弊社に帰属いたします。

5. 免責

Barcode.php の使用によって、直接的、あるいは、間接的に生じた、いかなる損害に対しても、弊社は補償賠償の責任を負わないものとします。

6. 禁止事項

Barcode.php 及びその複製物を第三者に譲渡・貸与する事は出来ません。Barcode.php を開発ツールとして再販/再配布することを禁止します。なお、モジュールとして組み込みを行い再販/再配布する場合は、開発ツールとしての再販/再配布には含まれませんので、OKです。

7. 保証の範囲

弊社は Barcode.php の仕様を予告無しに変更することがあります。その場合の利用者様に対する情報提供は、弊社 WebSite にて行う事とします。

8. 適用期間

本使用許諾条件は利用者様が Barcode.php を使用した日より有効です。利用者様が本使用許諾条件のいずれかの条項に違反した場合、又は、本許諾条件に同意出来ない場合は、利用者様は Barcode.php を一切使用出来ないものとします。

4-3. 代金支払い方法(ユーザ登録の方法)

Barcode.php の製品版をご利用頂ける場合は、ライセンスを購入して頂く必要があります。ライセンス形態及び代金支払方法は以下のとおりです。

- 必要なライセンス数の考え方
 - Barcode.php をインストールするパソコンの台数。

- 1ライセンス当たりの価格
 - 20,000 円(税込:21,600 円)(ソースコード含む)
 - ✧ バグフィックス等のバージョンアップは原則として無償とさせて頂きます。
 - ✧ 大幅な機能追加等によるバージョンアップの場合には別ライセンスとさせていただく場合がございます。
 - ✧ 本価格は Barcode.php の使用権に対するものです。カスタマイズや保守等の費用は一切含まれておりません。

- お支払方法 …先払い方法。後払い(納品後支払)方法は後に説明がございます。
(21,600 円 × ライセンス数)を下記口座へ銀行振込、
または、郵便振替による送金をして下さい。

銀行名	支店名 (コード)	口座番号	名義
三菱 UFJ 銀行	新宿支店 (341)	普通 3831891	ユ) パオアットオフィス
ジャパンネット銀行	すずめ支店 (002)	普通 6461359	ユ) パオアットオフィス

郵便口座番号	名義
00150-0-576845	有限会社 パオ・アット・オフィス

✧ 振込手数料は利用者様負担でお願い致します。

- お支払いの通知と製品の送付
 - 振り込みが完了した時点で、必ず弊社 WebSite の「Barcode.php 入金連絡フォーム」から入金のご連絡をお願いいたします。
<http://www.pao.ac/barcode.php/buy.html#form>
 - 弊社では上記連絡を受けて入金確認を行い、利用者様専用の Barcode.php の製品ダウンロード URL を利用者様へ電子メールにてお送りさせて頂きます。
 - ✧ 利用者様へは電子メール以外での製品の提供は原則として行いません。
 - ✧ 製品の再送付は原則として行いません。製品のファイルは消去してしまわないように大切にお取り扱いください。
 - お振り込み頂いても入金の連絡がない場合、こちらか振り込み人様の情報が分からぬいため、製品の送付が行えません。必ず入金連絡を行って頂くようお願いいたします。

- 見積書/納品書/請求書/領収証の発行、納品後のお支払いについて
見積書/納品書/請求書/領収証の発行は可能でございます。本製品納品後のお支
払いも可能でございます。

<http://www.pao.ac/barcode.php/buy.html>

上記サイトでの手続きにより、弊社からの見積書/納品書/請求書/領収証の発行、
及び、納品後のお客様からのお支払いを行えるようになっております。