

# Barcode.Ruby (gem版)

C++ WASM/FFI エンジン Ruby ラッパー

マニュアル

バージョン 1.0

---

有限会社 パオ・アット・オフィス

<https://www.pao.ac/>

# 目次

---

1. C++ バーコードエンジンを Ruby から手軽に利用。
2. はじめに
3. 導入方法
4. クイックスタート
5. APIリファレンス
6. Pure Ruby版との違い
7. 動作環境
8. ライセンス・お問い合わせ

# C++ バーコードエンジンを Ruby から手軽に利用。

---

## ユーザーズマニュアル

---

バージョン 1.0 — 2026年2月

有限会社 パオ・アット・オフィス

<https://www.pao.ac/>

# はじめに

## C++ WASM版 / FFI版とは

Barcode.Ruby gem版は、C++で書かれた高性能バーコードエンジンをRubyから利用するためのラッパーgemです。

2つのバリエーションがあります：

パッケージ	gem名	エンジン	特徴
Native版 (FFI)	barcode_pao_ffi	C++ DLL/SO (FFI経由)	最高速。Linux/Windows対応
WASM版	barcode_pao_wasm	C++ WASM (Node.js経由)	Node.jsがあればどこでも動く

どちらも Pure Ruby版と同じ API (setter メソッドパターン) で使用でき、全18種 のバーコードを PNG / JPEG / SVG で出力できます。

```
require 'barcode_pao_wasm'

qr = BarcodePaoWasm::QRCode.new(BarcodePaoWasm::FORMAT_PNG)
qr.draw("https://www.pao.ac/", 300)
base64 = qr.get_image_base64
```

## 特長

特長	説明
C++ エンジン	実績のある C++ バーコードライブラリを Ruby から利用
2つの選択肢	FFI (最高速) と WASM (ポータブル) から選べる
Pure Ruby版と同じAPI	setter メソッドパターンで統一
18種のバーコード	1D・2D・GS1・郵便まで網羅
PNG / JPEG / SVG	用途に応じた出力形式

## 対応バーコード一覧

Pure Ruby版と同じ 全18種 に対応しています。

カテゴリ	バーコード
1次元	Code39, Code93, Code128, GS1-128, NW-7, ITF, Matrix2of5, NEC2of5
商品	JAN-8, JAN-13, UPC-A, UPC-E
GS1 DataBar	標準型, 限定型, 拡張型
郵便	郵便カスタマバーコード

カテゴリ	バーコード
2次元	QRコード, DataMatrix, PDF417

# 導入方法

## 動作要件

**WASM版 (barcode\_pao\_wasm) :**

項目	要件
Ruby	3.0 以降
Node.js	18 以降 (WASMランタイム)

**Native版 (barcode\_pao\_ffi) :**

項目	要件
Ruby	3.0 以降
gem	ffi
OS	Linux (.so) または Windows (.dll)

## ダウンロード

<https://www.pao.ac/barcode.ruby/#download>

## ファイル構成 (WASM版)

```

barcode_ruby_gem/
├── easy2steps/
│   ├── Gemfile
│   └── app.rb
│   └── views/index.erb
└── allinone/
    ├── Gemfile
    ├── app.rb
    └── views/index.erb
    └── public/
└── barcode_pao_wasm/
    ├── lib/barcode_pao_wasm.rb
    ├── lib/barcode_pao_wasm/...
    └── wasm/
        ├── barcode.wasm
        ├── barcode.mjs
        └── _barcode_runner.mjs

```

## 起動方法

```
cd barcode_ruby_gem/easy2steps  
bundle install  
ruby app.rb  
# → http://localhost:5742
```

# クイックスタート

## QRコードをPNGで生成 (WASM版)

```
require 'barcode_pao_wasm'

qr = BarcodePaoWasm::QRCode.new(BarcodePaoWasm::FORMAT_PNG)
qr.set_error_correction_level("M")
qr.draw("https://www.pao.ac/", 300)

base64 = qr.get_image_base64
image_bytes = qr.get_image_memory
File.binwrite("qr.png", image_bytes)
```

## QRコードをPNGで生成 (FFI版)

```
require 'barcode_pao_ffi'

qr = BarcodePaoFfi::QRCode.new(BarcodePaoFfi::FORMAT_PNG)
qr.set_error_correction_level("M")
qr.draw("https://www.pao.ac/", 300)

base64 = qr.get_image_base64
```

## SVGベクター出力

```
bc = BarcodePaoWasm::Code128.new(BarcodePaoWasm::FORMAT_SVG)
bc.set_show_text(true)
bc.set_text_even_spacing(true)
bc.draw("Hello-2026", 400, 100)
svg = bc.get_svg
```

## REST APIサーバー

```
require 'sinatra'
require 'json'
require 'barcode_pao_wasm'

get '/api/qr' do
  content_type :json
  code = params[:code] || "https://www.pao.ac/"

  qr = BarcodePaoWasm::QRCode.new(BarcodePaoWasm::FORMAT_PNG)
  qr.draw(code, 300)
  base64 = qr.get_image_base64

  { base64: base64 }.to_json
end
```

# APIリファレンス

## Pure Ruby版との違い

gem版 (FFI/WASM) は setter メソッドパターンを使用します：

Pure Ruby版	gem版 (FFI/WASM)
bc.show_text = true	bc.set_show_text(true)
bc.text_even_spacing = true	bc.set_text_even_spacing(true)
bc.extended_guard = true	bc.set_extended_guard(true)
bc.foreground_color = [r,g,b,a]	bc.set_foreground_color(r, g, b, a)
bc.fit_width = true	bc.set_fit_width(true)

出力メソッドは共通です：get\_image\_base64, get\_svg, get\_image\_memory

## 共通メソッド（全バーコード）

メソッド	説明
set_output_format(fmt)	出力形式 ("png", "jpeg", "svg")
set_foreground_color(r, g, b, a)	前景色 (RGBA)
set_background_color(r, g, b, a)	背景色 (RGBA)
set_fit_width(bool)	幅ぴったり描画
set_px_adjust_black(adj)	黒バー幅調整
set_px_adjust_white(adj)	白スペース幅調整
get_image_base64	Base64 データURI取得
get_svg	SVG文字列取得
get_image_memory	バイナリ取得

## 1次元バーコード共通メソッド

メソッド	説明	デフォルト
set_show_text(bool)	テキスト表示	true
set_text_even_spacing(bool)	テキスト均等割付	true
set_text_font_scale(scale)	フォントサイズ倍率	1.0
set_text_vertical_offset_scale(scale)	垂直オフセット倍率	1.0
set_min_line_width(width)	最小線幅	1

draw メソッド: draw(code, width, height)

## 各バーコード型

Code39 / NW-7: set\_show\_start\_stop(bool)

Code128: set\_code\_mode(mode) — CODE128\_AUTO / CODE128\_CODE\_A / CODE128\_CODE\_B / CODE128\_CODE\_C

JAN-8 / JAN-13 / UPC-A / UPC-E: set\_extended\_guard(bool)

GS1 DataBar 14: コンストラクタに symbol\_type 引数 (OMNIDIRECTIONAL / STACKED / STACKED\_OMNIDIRECTIONAL)

GS1 DataBar 拡張型: set\_no\_of\_columns(cols)

郵便カスタマバーコード: draw(code, height) — 幅は自動計算

## QRコード

メソッド	説明	デフォルト
set_string_encoding(enc)	"utf-8" / "shift-jis"	"utf-8"
set_error_correction_level(level)	"L" / "M" / "Q" / "H"	"M"
set_version(ver)	0 (自動) ~ 40	0
set_encode_mode(mode)	"N" / "A" / "B" / "K"	"B"

draw メソッド: draw(code, size) — 正方形

## DataMatrix

メソッド	説明	デフォルト
set_string_encoding(enc)	"utf-8" / "shift-jis"	"utf-8"
set_code_size(size)	DX_SZ_AUTO 等	DX_SZ_AUTO
set_encode_scheme(scheme)	DX_SCHEME_AUTO_BEST 等	DX_SCHEME_AUTO_BEST

draw メソッド: draw(code, size) — 正方形

## PDF417

メソッド	説明	デフォルト
set_string_encoding(enc)	"utf-8" / "shift-jis"	"utf-8"
set_error_level(level)	0~8	2
set_columns(cols)	列数	0 (自動)
set_rows(rows)	行数	0 (自動)
set_aspect_ratio(ratio)	縦横比	3.0
set_y_height(h)	Y高さ係数	3

draw メソッド: draw(code, width, height)

## Pure Ruby版との違い

項目	Pure Ruby版	gem版 (FFI/WASM)
エンジン	Ruby で実装	C++ バイナリ
依存	chunky_png, ttfunk	FFI gem / Node.js
速度	標準	高速 (特にFFI版)
API	attr_accessor	setter メソッド
ポータビリティ	どこでも動く	FFI: OS限定 / WASM: Node.js必須
ソースコード	Ruby ソース付き	バイナリ配布

### どちらを選ぶべきか？

- Pure Ruby版: Ruby のみの環境で動かしたい、ソースコードを確認したい
- FFI版: 最高速を求める、Linux/Windows サーバーで運用
- WASM版: Node.js がある環境でポータブルに使いたい

## 動作環境

### WASM版

項目	要件
Ruby	3.0 以降
Node.js	18 以降
OS	Windows / macOS / Linux

### Native版 (FFI)

項目	要件
Ruby	3.0 以降
gem	ffi
OS	Linux / Windows

# ライセンス・お問い合わせ

## 使用許諾

### Barcode.Ruby

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

1. 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.Ruby 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. 同意 — 利用者様が Barcode.Ruby 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。
3. ライセンスの購入 — 製品版を使用して開発を行う場合、1 台の開発用コンピュータにつき 1 ライセンスの購入が必要です。お客様環境等、開発コンピュータでないマシンでの使用にはライセンスは不要です（ランタイムライセンスフリー）。
4. 著作権 — Barcode.Ruby 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.Ruby の使用によって、直接的また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.Ruby およびその複製物を第三者に譲渡・貸与することはできません。開発ツールとしての再販・再配布を禁止します。ただし、モジュールとして組み込みを行い再販・再配布する場合は問題ございません。
7. 保証の範囲 — 弊社は Barcode.Ruby の仕様を予告なしに変更することがあります。
8. 適用期間 — 本使用許諾条件は利用者様が Barcode.Ruby を使用した日より有効です。

## ライセンス

試用版: 生成されるバーコードに「SAMPLE」の透かしが表示されます。機能制限はありません。

製品版: 透かしなしでバーコードを生成できます。

## お問い合わせ

Webサイト	<a href="https://www.pao.ac/">https://www.pao.ac/</a>
製品ページ	<a href="https://www.pao.ac/barcode.ruby/">https://www.pao.ac/barcode.ruby/</a>
メール	info@pao.ac

Barcode.Ruby (gem版) ユーザーズマニュアル バージョン 1.0 — 2026年2月

© 2026 有限会社 パオ・アット・オフィス