

バーコード・リーダーAPI仕様書

株式会社ツインアース

DAO MINHQUAN



バーコード・リーダーAPI仕様書

目次

1. バーコード・リーダーについて.....	3
1.1. 画面構成.....	3
1.2. 機能概要.....	4
2. UI設定API.....	4
2.1. インタフェース全体の表示・非表示API.....	4
2.2. 個別インタフェースの表示・非表示API.....	5
3. 機能設定API.....	5
3.1. カメラ起動.....	5
3.2. カメラ停止.....	5
3.3. 認識領域のサイズ設定.....	5
3.4. カメラViewの枠を設定.....	5
3.5. カメラViewの自動回転.....	6
3.6. ズーム率設定.....	6
3.7. カメラ指定.....	6
3.8. カメラライト設定.....	6
3.9. 音声設定.....	6
3.10. バイブレーション設定.....	6
3.11. フォーカス設定.....	6
3.12. 読み取り方向設定.....	7
3.13. バーコードタイプの絞り込み.....	7
3.14. 設定画面の呼び出し.....	8
3.15. 認識領域の枠を独自に描画.....	8
3.16. 認識結果の取得.....	8
3.17. 認識結果表示・非表示.....	8
4. ライブラリの利用.....	8
4.1. XCodeプロジェクトに追加.....	8
4.2. 実装.....	9
4.3. サンプル1.....	9
4.3.1. カメラViewの初期化.....	10
4.3.2. 結果取得.....	10
4.3.3. カメラ開始・終了.....	12

1. バーコード・リーダーについて

本ドキュメントは、バーコード・リーダーライブラリー（ライブラリ）のUI設定、機能設定のAPIについて説明します。1. 節はライブラリで提供する機能を紹介し、2. 節はUIの詳細設定について説明し、3. 節は機能設定を説明します。4. 節はライブラリを組み入れるための実装を説明します。

1.1. 画面構成

下図はライブラリの画面構成を示す。

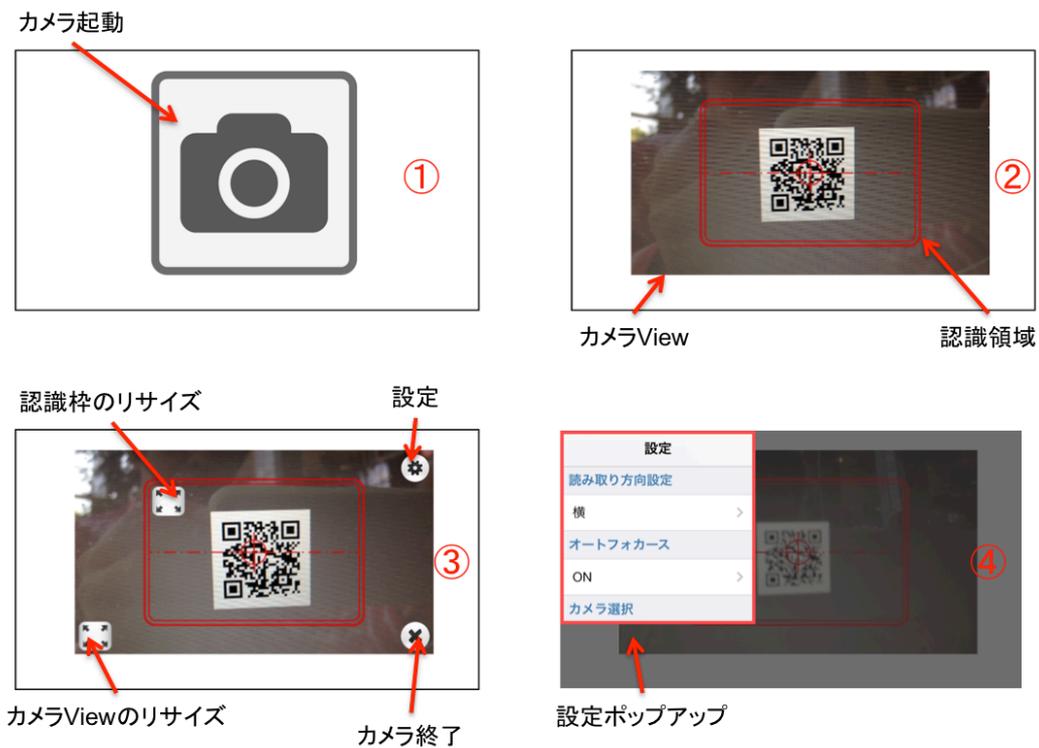


図1：バーコード・リーダー構成図

※ 画面①はアプリを起動した直後の画面です。カメラが停止する状態でカメラボタンを押すとカメラが起動します（画面②）
カメラ映像を表示する領域です。カメラViewのサイズ、表示位置、表示方向を設定することができます。

※ 画面②は認識画面です。主にカメラViewと認識領域を表す赤い枠があります。枠のセンターラインは認識方向を表します。画面をタップするとライブラリを設定するためのUIが表示されます（画面③）

※画面③はライブラリを設定するためのUIです。主にカメラViewのリサイズ、認識領域のリサイズ、設定ポップアップを呼び出し、カメラを停止などのUIがあります。設定ボタンを押すと設定画面が表示されます（画面④）

※ 画面④はライブラリの様々な機能を設定する画面です。

1. 2. 機能概要

本ライブラリは、カメラ機能、バーコード認識機能などを設定するUIを提供するだけでなく、ユーザーが独自のUIで設定できるように様々なAPIを提供しています。主に次のAPIがあります。

- ※ デフォルトUI全体を表示・非表示API
- ※ デフォルトUIの一部を表示・非表示API
- ※ カメラViewリサイズAPI
- ※ 認識領域リサイズAPI
- ※ カメラオン・オフAPI
- ※ 設定ポップアップ呼び出しAPI
- ※ 読み取り方向設定
- ※ フォーカス設定
- ※ 前後カメラを選択
- ※ カメラをズーム、認識画面でのピッチングでもズームできます。
- ※ 読み取りバーコード種類の選択
- ※ 認識後に鳴らすサウンドをオン・オフ
- ※ バイブレーションをオン・オフ
- ※ デバッグ画像を表示・非表示
- ※ カメラライトオン・オフ
- ※ 認識結果表示・

2. UI設定API

本ライブラリは、カメラの機能を設定するUIが用意されていますが、ユーザーが独自にUIを作成することも出来ます。そのために、不要になったデフォルトUIを非表示することができます。以下は本ライブラリのデフォルトUIを表示・非表示の設定を説明します。

2. 1. インタフェース全体の表示・非表示API

- ※ インタフェース全体の表示・非表示を設定するプロパティです。
- ※ 参照ファイルBarcodeReaderView.h
- ※ 設定 property **ShowInterfaceMode** showInterfaceMode。設定値は下図に示します。

設定Id	対応するボタン
AllwayShowInferface	常にUIを表示
ShowInShortTime	表示した後、自動的に非表示になります。

HideInterface	認識領域をリサイズするボタン
---------------	----------------

2.2. 個別インタフェースの表示・非表示API

※ カメラViewのリサイズ、移動ボタン、認識領域リサイズボタン、設定ボタン、カメラを停止するボタンを個別に表示・非表示にするAPIです。

※ 参照ファイルBarcodeReaderView.h

※ 設定メソッド:

- (void)hideInterfaceWithId:(BarcodeInterfaceId)interfaceId

hidden:(BOOL)hidden;

hidden=YESの場合は、ボタンを非表示にし、hidden=NOの場合はボタンを表示にします。

interfaceId は表示・非表示したいボタンのIdを表します。interfaceIdは次の表示まとめます。デフォルトでは全てのUIが表示されます。

設定Id	対応するボタン
SettingButtonId	設定ボタン
CloseCameraButtonId	カメラ停止ボタン
ResizeScanAreaButtonId	認識領域をリサイズするボタン
ResizeCameraButtonId	カメラViewをリサイズするボタン

3. 機能設定API

3.1. カメラ起動

※ 参照ファイルBarcodeReaderView.h

※ 起動メソッド:- (void)startCamera

※ 起動完了のコールバック(オプション):- (void)finishedStartCamera

3.2. カメラ停止

※ 参照ファイルBarcodeReaderView.h

※ 設定メソッド:- (void)stopCamera

※ 起動完了のコールバック(オプション):- (void)finishedStopCamera

3.3. 認識領域のサイズ設定

※ 参照ファイルBarcodeReaderView.h

※ 設定メソッド:- (void)setScanViewWithSize:(CGSize)size

3.4. カメラViewの枠を設定

※ 参照ファイルBarcodeReaderView.h

※ 設定プロパティ CGRect frame

3.5. カメラViewの自動回転

- ※ デバイスが回転する際、自動的にデバイスの状態を取得してカメラViewを自動的に回転するAPI
- ※ 設定メソッド:- (void)autoRotate

3.6. ズーム率設定

- ※ 参照ファイルBarcodeReaderView.h
- ※ 設定メソッド:- (void)zoomCameraWithZoomFactor:(CGFloat)factor
- ※ CGFloat factorはズーム率です。

3.7. カメラ指定

- ※ 参照ファイルBarcodeReaderView.h
- ※ 設定メソッド- (void)setCameraWithPosition:(CameraPosition)position
- ※ CameraPosition positionは選択するカメラです。

戻り値	現在の設定モード
FrontCamera	前方カメラを使用
BackCamera	後方カメラを使用

3.8. カメラライト設定

- ※ デバイスのバックにあるライトのオン・オフを設定
- ※ 参照ファイルBarcodeReaderView.h
- ※ 設定メソッド:- (void)setCameraLight:(BOOL)isOn

3.9. 音声設定

- ※ 認識が成功したとき、音声を鳴らす設定です。
- ※ 参照ファイルBarcodeReaderView.h
- ※ 設定メソッド:- (void)setSound:(BOOL)playSound

3.10. バイブレーション設定

- ※ 認識が成功したとき、端末のバイブレーションを鳴らす設定です。
- ※ 参照ファイルBarcodeReaderView.h
- ※ 設定メソッド:- (void)setVibration:(BOOL)playVibration

3.11. フォーカス設定

- ※ カメラのフォーカス調整を自動行う設定です。
- ※ 参照ファイルBarcodeReaderView.h
- ※ 設定メソッド:- (void)setCameraFocus:(BOOL)isAutoFocus

3.12. 読み取り方向設定

※ バーコードの読み取り方向を設定します。

※ 参照ファイルBarcodeReaderView.h

※ 設定メソッド:

-(void) setRegconizeDirection: (RegconizeDirection) regconizeDirection
regconizeDirectionは以下の値を設定できます。

戻り値	現在の設定モード
RegconizeOnPostrait	縦方向読み取り
RegconizeOnLandscape	横方向読み取り
RegconizeEither	縦・横両方読み取り

3.13. バーコードタイプの絞り込み

※ 読み取り対象のバーコードタイプが決まった場合は、読み取りタイプを制限することで読み込む時間をより短縮できます。

※ 参照ファイルBarcodeReaderView.h

※ 設定メソッド:-

(void) setRegconizeBarcode: (BarcodeFormat) barcodeFormat

regconize: (BOOL) regconize

※ regconize=YES読み込む、regconize=NO読み込まない

※ barcodeFormat :設定したいバーコードの種類です。次の値を設定できます。

	設定値	バーコードタイプ
0	kBarcodeFormatAztec	Aztec 2次元バーコード
1	kBarcodeFormatCodabar	CODABAR 1次元バーコード
2	kBarcodeFormatCode39	Code 39 1次元
3	kBarcodeFormatCode93	Code 93 1次元
4	kBarcodeFormatCode128	Code 128 1次元
5	kBarcodeFormatDataMatrix	Data Matrix 2次元
6	kBarcodeFormatEan8	EAN-8 1次元
7	kBarcodeFormatEan13	EAN-13 1次元
8	kBarcodeFormatITF	ITF 1次元
9	kBarcodeFormatMaxiCode	MaxiCode 2次元
10	kBarcodeFormatPDF417	PDF417 1次元
11	kBarcodeFormatQRCode	QR Code 2D
12	kBarcodeFormatRSS14	RSS 14 1次元
13	kBarcodeFormatRSSExpanded	RSS EXPANDED 1次元
14	kBarcodeFormatUPCA	UPC-A 1次元
15	kBarcodeFormatUPCE	UPC-E 1次元
16	kBarcodeFormatUPCEANExtension	UPC/EAN extension 1次元

注意：本ライブラリLimited, Compositeタイプを対応しません。

3. 14. 設定画面の呼び出し

※ 本ライブラリは上記の機能を設定するためのデフォルトUIを用意しています。独自のUIが必要ない場合は、こちらを利用して設定することができます。

※ 参照ファイルBarcodeReaderView.h

※ 呼び出しメソッド:- (void) showSettingPopup

3. 15. 認識領域の枠を独自に描画

※ 認識領域はUIViewの拡張クラスとして実装されるので、独自のUIViewを置き換えれば、枠の描画、センターアイコンの描画などを独自に実装できます。

※ 参照ファイル: CaptureView.h

※ 設定プロパティ:@property (nonatomic, assign) UIView *scanAreaView

3. 16. 認識結果の取得

※ 参照ファイルCaptureView.h

※ 実装メソッド:- (void) captureResult: (BarcodeResult*) result

※ 実装結果: BarcodeResult *result;

3. 17. 認識結果表示・非表示

※ カメラ画面上に認識の結果を表示・非表示の設定

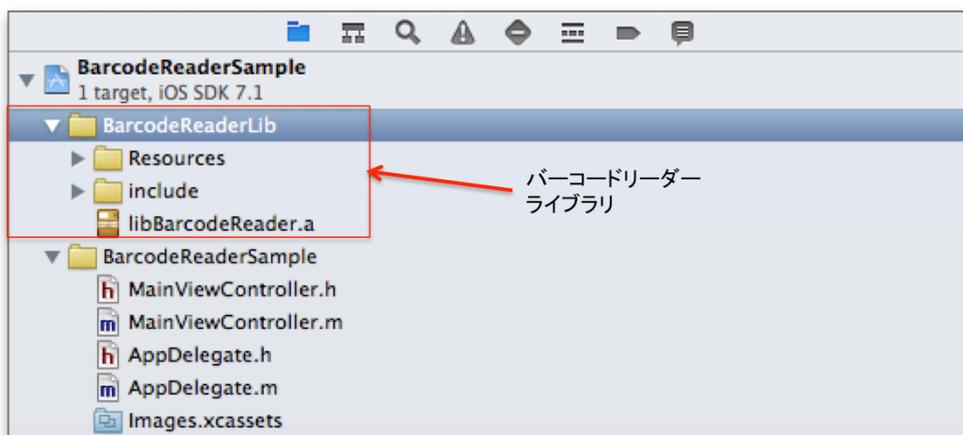
※ 参照ファイルCaptureView.h

※ 設定プロパティ: BOOL showResults

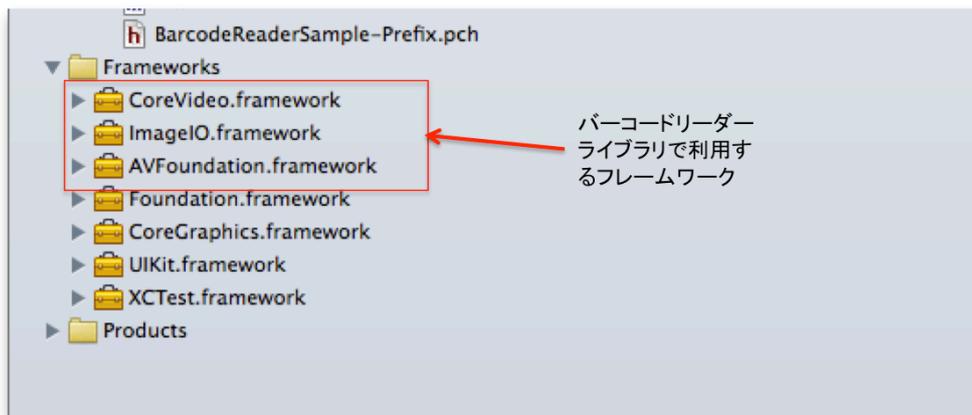
4. ライブラリの利用

4. 1. XCodeプロジェクトに追加

※ 下図のようにプロジェクトにライブラリのフォルダを追加



※ プロジェクトに必要なフレームワークを追加



4.2. 実装

※ 以下のようにインスタンスを生成します。

```
BarcodeReaderView *instance = [[BarcodeReaderView alloc] initWithFrame:rect];
```

※ BarcodeReaderViewはUIViewの派生クラスですので、使用したいUIViewにsubviewとして追加することができます。

```
[viewController.view addSubview:instance];
```

※ デバイスを回転させる場合は、回転する度に自動回転メソッド (3.5節) を実行してください。

```
[instance autoRotate];
```

※ 注意: UIViewcontrollerのUIViewに直接追加しない場合は、デバイスが回転した後、追加したUIViewの回転に応じてカメラViewのサイズ方などを再設定する必要があります。

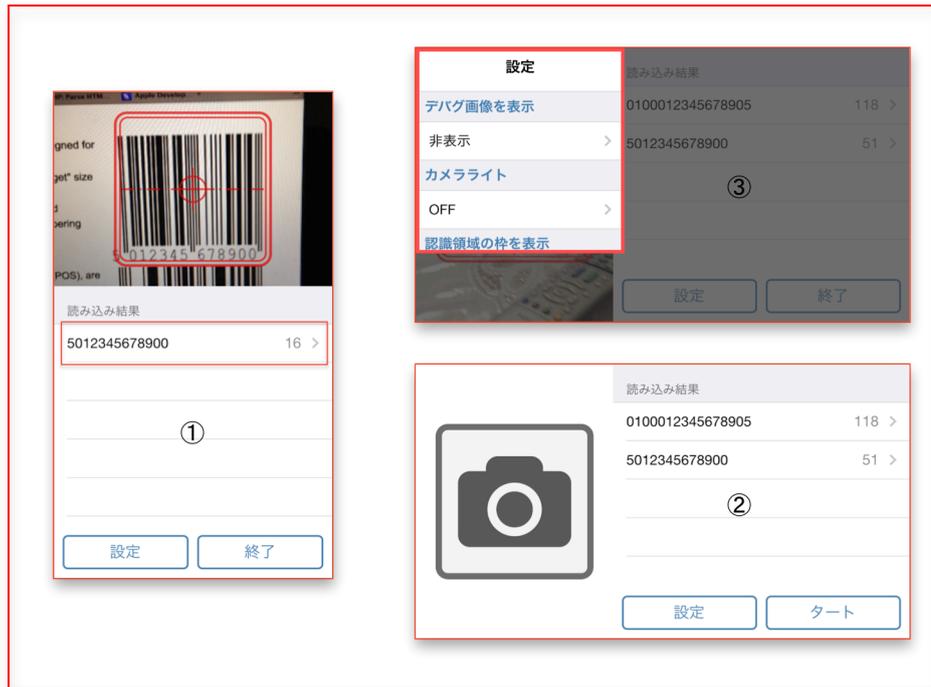
※ 読み込み結果を取得したい場合はdelegateを設定必要があります。

```
instance.delegate = self;
```

4.3. サンプル1

※ サンプル1は下図のように読み込むカメラ、読み込んだデータの結果を表示、設定画面を表示する設定ボタン、カメラ起動終了ボタンから構成されます。本サンプルはライブラリのデフォルトUIを使わずにアプリ側の「設定」ボタン、「スタート」ボタンでAPIを呼び出します。結果の右側の数値は、同じバーコードを読み込む回数です。

※ 上記の機能は、MainViewControllerクラスに実装されます。



4.3.1. カメラViewの初期化

```

//カメラのフレームを定義
CGRect rect = self.view.bounds;
rect.size.height *= CameraHeightCoe;
//インスタンスを生成
_barcodeReader = [[BarcodeReaderView alloc] initWithFrame:rect];
//ライブラリのデフォルトUIを非表示にする
_barcodeReader.showInterfaceMode = HideInterface;
//カメラ画面上に読み込み結果を表示しないように設定
_barcodeReader.showResults = NO;
//結果を取得するためにdelegateを設定
_barcodeReader.delegate = self;
[self.view addSubview:_barcodeReader];

```

4.3.2. 結果取得

```

// BarcodeResult読み込んだ結果
- (void)captureResult:(BarcodeResult *)result
{
    //結果を取り込む
    [self addObject:result];
    [_tableView reloadData];
}

- (void)addObject:(BarcodeResult*)result {

    for (int i=0;i<[_results count]; i++) {

        BarcodeResultEx *r = [_results objectAtIndex:i];
        //読み込んだバーコードは既に存在するかどうかを判定
        if ([r.result.code isEqualToString:result.code] &&
            r.result.barcodeFormat==result.barcodeFormat) {
            //読み込んだバーコードなら読み込む回数をカウントする
            r.count ++;
            [_results replaceObjectAtIndex:i withObject:r];
            return;
        }
    }

    //まだ読み込んでいないバーコードなら、結果の配列に追加する。
    BarcodeResultEx *exResult = [[BarcodeResultEx alloc] initWithResult:result];
    [_results addObject:exResult];
}

```

4.3.3. カメラ開始・終了

```
- (void)pushedClose
{
    //ボタンを押したとき、カメラが起動中なら終了させる
    if (_barcodeReader.running) {

        [_barcodeReader stopCamera];
        [_closeButton setTitle:@"タート" forState:UIControlStateNormal];
    } else {
        //ボタンを押したとき、カメラが停止中なら起動させる
        [_barcodeReader startCamera];
        [_closeButton setTitle:@"終了" forState:UIControlStateNormal];
    }
}
```

4.4. サンプル2