

イメージ・アルファ画像処理ライブラリ DataMatrix 処理マニュアル

第 1.00 版

株式会社 イメージ・アルファ

はじめに

本ドキュメントは、イメージ・アルファ画像処理ライブラリ IALIB の DataMatrix 復号機能について解説しています。

表 0-1 本ドキュメントの適用 IALIB バージョン

ソフト名	バージョン番号
IALIB	Version 1.2.6

目次

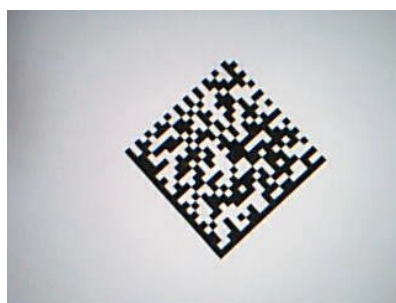
1. 機能.....	4
2. 使用方法.....	5
3. 関数.....	8
4. 列挙体、構造体.....	15

1. 機能

IALIB では、DataMatrix(データマトリクス)を復号する関数を用意しています。

項目	対応範囲
ECC レベル	ECC200
シンボルサイズ	10x10~26x26、正方形のみ
色	標準(白地に黒モジュール)
データファンクション	ASCII, Pad, 2-digitData, C40, UpperShift, ANSI X12, Text (連結等は未対応)

表 1-1 対応 DataMatrix



回転



かざし読み

図 1-1 復号できる DataMatrix 例

2. 使用方法

下のサンプルコードは、BMP ファイルから画像を読み込み、その画像にある DataMatrix を復号し、メッセージボックスで結果を表示するアプリケーションです。

```
int APIENTRY WinMain( HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow )
{
    int i;
    int imgID;
    IA_Size imgSize;
    int dmxCnt;
    IA_DMX_Decode_Ctl ctl;
    IA_DMX_Decode_Rslt rslt;
    char dmxCntStr[IA_DMX_MAXSTRBUFF];
    char message[100];

    // IALIB・有効化
    IA_Open(); → 2.1.1節

    // IALIB・DataMatrix復号機能・有効化
    IA_DMX_Open(); → 2.1.1節

    // 処理対象画面確保
    imgSize.xSize = 640;
    imgSize.ySize = 480;
    imgID = IA_AllocImg( IA_IMG_CHAR, &imgSize );

    // BMPファイルを読み込み処理対象画面へ格納
    IA_BMPFile_Load( imgID, "D:\Work\TestImg\DMXImg.bmp" );

    // 制御パラメータ設定
    IA_DMX_GetInitCtl( &ctl ); // ctlを初期化
    ctl.fFront = 1; // 正面を向いた(かざし読みでない)DataMatrixのみ復号
    IA_DMX_SetCtl( &ctl ); // 設定 → 2.1.2節

    // 復号
    dmxCnt = IA_DMX_Decode( imgID, IA_DMX_MAXSTRBUFF, dmxCntStr ); → 2.1.3節

```

⋮

⋮

```

// 結果表示
if( dmxNum > 0 ){
    IA_DMX_GetRslt( 0, &rslt );
    sprintf( message, "サイズ : %dx%d\n復号文字列 : %s", rslt.sizeR, rslt.sizeC, rslt.str );
    MessageBox( NULL, message, "復号成功", MB_OK );
} else if( dmxNum == 0 ){
    MessageBox( NULL, "復号失敗", "復号失敗", MB_OK );
} else{
    MessageBox( NULL, "エラー", "エラー", MB_OK );
}

// IALIB・DataMatrix復号機能・無効化
IA_DMX_Close();

// IALIB・無効化
IA_Close();

return 0;
}

```

→ 2.1.4節

→ 2.1.1節

→ 2.1.1節

2.1.1. 各種有効化

IALIB全体の有効化、DataMatrix復号機能の有効化を最初に行います。無効化されるまで機能が使用できます。

有効時には何回でも復号が行えます。復号のたびに有効化をする必要はありません。

2.1.2. 制御パラメータの設定

構造体 IA_DMX_Decode_Ctl (4.5 節)の変数を初期化して、環境に合わせてカスタマイズし、関数 IA_DMX_SetCtl (3.2.1 節)の引数とします。

2.1.3. 復号

関数 IA_DMX_Decode (3.2.3 節)により、制御パラメータ設定値に基づいた復号を行います。

2.1.4. 復号結果(詳細)の取得

関数 IA_DMX_GetRslt (3.2.4 節)により復号結果(詳細)を取得できます。

(復号の成否、復号の文字列を取得したいだけであれば、この関数は不要です)

3. 関数

本章では関数インターフェースを記しています。
 関連する列挙体、構造体は4章を参照してください。

区分	節番号	関数名	機能
有効/無効	3.1.1 節	IA_DMX_Open	有効化
	3.1.2 節	IA_DMX_Close	無効化
復号	3.2.1 節	IA_DMX_SetCtl	制御パラメータの設定
	3.2.2 節	IA_DMX_GetInitCtl	制御パラメータのデフォルト値を取得
	3.2.3 節	IA_DMX_Decode	復号
	3.2.4 節	IA_DMX_GetRslt	結果取得

表 3-1 DataMatrix 復号一覧

3.1. 有効/無効

3.1.1. 有効化 [IA_DMX_Open]

インタフェース

```
int IA_DMX_Open ();
```

リターン値

基本エラーコード

機能

DataMatrix 復号機能を有効にします。

本関数の呼び出しは、IALIB を有効化した後(つまり関数 IA_Open を呼んだ後)に行ってください。

3.1.2. 無効化 [IA_DMX_Close]

インタフェース

```
int IA_DMX_Close ();
```

リターン値

基本エラーコード

機能

DataMatrix 復号機能を無効にします。

IALIB 内部にて確保したメモリの開放などの後処理を行いますので、有効にした場合には本関数を必ず呼び出してください。

本関数の呼び出しは、IALIB を無効化する前(つまり関数 IA_Close を呼ぶ前)に行ってください。

3.2. 復号

3.2.1. 制御パラメータの設定 [IA_DMX_SetCtl]

インタフェース

```
int IA_DMX_SetCtl (
    IA_DMX_Decode_Ctl *pCtl
);
```

パラメータ

名称	タイプ	入/出	許容値	意味
pCtl	IA_DMX_Decode_Ctl* (4.5節)	入力		制御パラメータ

リターン値

基本エラーコード

機能

制御パラメータを設定します。この設定値で復号(3.2.3節)が行われます。

本関数を呼び出さずに復号する場合は、デフォルト値(表 3-2)で動作します。

3.2.2. 制御パラメータデフォルト値の取得 [IA_DMX_GetInitCtl]

インタフェース

```
int IA_DMX_GetInitCtl (
    IA_DMX_Decode_Ctl *pCtl
);
```

パラメータ

名称	タイプ	入/出	許容値	意味
pCtl	IA_DMX_Decode_Ctl* (4.5 節)	出力		制御パラメータデフォルト値

リターン値

基本エラーコード

機能

制御パラメータのデフォルト値(表 3-2)を引数の構造体に設定します。
主に構造体の初期化に使用します。

(網掛には現在指定不可)

変数名	値	説明
colorOrder	IA_DMX_COLORORDER_BLACK	標準
moduleRange.min	0	-
moduleRange.max	20	-
moduleShape	IA_DMX_MODULESHAPE_SQUARE	四角
maxNum	1	最大1つ復号
fFront	0	正面に限定しない
decodeSize[]	全て1	全サイズ復号する

表 3-2 制御パラメータデフォルト値(初期値)

3.2.3. 復号 [IA_DMX_Decode]

インタフェース

```
int IA_DMX_Decode (
    int imgID, int strBuffSize, char str[]
);
```

パラメータ

名称	タイプ	入/出	許容値	意味
imgID	int	入力		処理対象画像
strBuffSize	int	入力		復号文字列格納領域のバイト数
str	char []	出力		復号文字列

リターン値

0 以上 : 検知 DataMatrix 数
 0 未満 : 基本エラーコード

機能

画像から DataMatrix を検知、復号します。

現在は複数の DataMatrix を復号することはできません。

復号文字列が格納領域 str に収まらない場合(文字列バイト数 \geq strBuffSize)には、エラーとなります。

3.2.4. 結果取得 [IA_DMx_GetRslt]

インタフェース

```
int IA_DMx_GetRslt (
    int dmxID, IA_DMx_Decode_Rslt *pRslt
);
```

パラメータ

名称	タイプ	入/出	許容値	意味
dmxID	int	入力	0	0としてください
pRslt	IA_DMx_Decode_Rslt* (4.6節)	出力		復号結果

リターン値

基本エラーコード

機能

復号したDataMatrixの各種情報を取得します。

4. 列挙体、構造体

4.1. 色に関する復号順

```
typedef enum {
    IA_DMX_COLORORDER_BLACK,           // 黒 (通常の Data Matrix)
    IA_DMX_COLORORDER_WHITE,          // 白 (未対応)
    IA_DMX_COLORORDER_BLACK_WHITE,    // 黒→白 (黒で復号失敗したら、白で再度試みる) (未対応)
    IA_DMX_COLORORDER_WHITE_BLACK,    // 白→黒 (白で復号失敗したら、黒で再度試みる) (未対応)
    IA_DMX_COLORORDER_END
} IA_DMX_ColorOrder;
```

色に関する復号順を表します。
 構造体 IA_DMX_Decode_Ctl (4.5 節) のメンバとなっています。
 現在は IA_DMX_COLORORDER_BLACK のみ対応しています。

4.2. モジュールの形状

```
typedef enum {
    IA_DMX_MODULESHAPE_SQUARE,        // 四角
    IA_DMX_MODULESHAPE_RING,          // 環 (未対応)
    IA_DMX_MODULESHAPE_DOTL,          // ドット大 (未対応)
    IA_DMX_MODULESHAPE_DOTS,          // ドット小 (未対応)
    IA_DMX_MODULESHAPE_END
} IA_DMX_ModuleShape;
```

モジュールの形状を表します。
 構造体 IA_DMX_Decode_Ctl (4.5 節) のメンバとなっています。
 現在は四角にのみ対応しています。

4.3. モジュールの画像上サイズ

```
typedef struct {
    int min;
    int max;
} IA_DMX_ModuleRange;
```

DataMatrix のモジュールサイズの範囲を格納します。単位は画素。
 構造体 IA_DMX_Decode_Ctl (4.5 節) のメンバとなっています。
 この範囲を目安に復号が行われます。

4.4. シンボルサイズ

```
typedef enum {
    IA_DMX_SYMBOLSIZE_10x10,
    IA_DMX_SYMBOLSIZE_12x12,
    IA_DMX_SYMBOLSIZE_14x14,
    IA_DMX_SYMBOLSIZE_16x16,
    IA_DMX_SYMBOLSIZE_18x18,
    IA_DMX_SYMBOLSIZE_20x20,
    IA_DMX_SYMBOLSIZE_22x22,
    IA_DMX_SYMBOLSIZE_24x24,
    IA_DMX_SYMBOLSIZE_26x26,
    IA_DMX_SYMBOLSIZE_32x32, // (未対応)
    IA_DMX_SYMBOLSIZE_36x36, // (未対応)
    IA_DMX_SYMBOLSIZE_40x40, // (未対応)
    IA_DMX_SYMBOLSIZE_44x44, // (未対応)
    IA_DMX_SYMBOLSIZE_48x48, // (未対応)
    IA_DMX_SYMBOLSIZE_52x52, // (未対応)
    IA_DMX_SYMBOLSIZE_64x64, // (未対応)
    IA_DMX_SYMBOLSIZE_72x72, // (未対応)
    IA_DMX_SYMBOLSIZE_80x80, // (未対応)
    IA_DMX_SYMBOLSIZE_88x88, // (未対応)
    IA_DMX_SYMBOLSIZE_96x96, // (未対応)
    IA_DMX_SYMBOLSIZE_104x104, // (未対応)
    IA_DMX_SYMBOLSIZE_120x120, // (未対応)
    IA_DMX_SYMBOLSIZE_132x132, // (未対応)
    IA_DMX_SYMBOLSIZE_144x144, // (未対応)

    // Rectangular Symbols
    IA_DMX_SYMBOLSIZE_8x18, // (未対応)
    IA_DMX_SYMBOLSIZE_8x32, // (未対応)
    IA_DMX_SYMBOLSIZE_12x26, // (未対応)
    IA_DMX_SYMBOLSIZE_12x36, // (未対応)
    IA_DMX_SYMBOLSIZE_16x36, // (未対応)
    IA_DMX_SYMBOLSIZE_16x48, // (未対応)

    IA_DMX_SYMBOLSIZE_END
} IA_DMX_SymbolSize;
```

シンボルサイズを表します。

4.5. 復号制御パラメータ

```
typedef struct {
    IA_DMX_ColorOrder colorOrder; // 色に関する復号順 (現在 IA_DMX_COLORORDER_BLACK のみ対応)
    IA_DMX_ModuleRange moduleRange; // モジュールサイズの目安
    IA_DMX_ModuleShape moduleShape; // モジュールの形状 (現在 IA_DMX_MODULESHAPE_SQUARE のみ対応)
    int maxNum; // 最大復号Data Matrix 数 (現在1のみ対応)
    int fFront; // 正面のData Matrixのみ復号する
    int decodeSize[IA_DMX_SYMBOLSIZE_END]; // 復号サイズ
                                        // decodeSize[i]=1/0でサイズiを復号する/しない。
                                        // iには列挙体 IA_DMX_SymbolSizeを使用する

    // 構造的接続
    // (未対応)
} IA_DMX_Decode_Ctl, IA_DMX_DecodeB_Ctl;
```

復号制御パラメータを格納します。

関数 IA_DMX_GetInitCtl (2.1.2 節)にて初期化してから使用してください。

4.6. 復号結果

```
typedef struct {
    // 復号成否
    int          fDecode;          // 1 : 文字列への復号成功, 0 : 文字列への復号失敗 (未対応モード)
    // 各種情報
    IA_DMX_SymbolSize size;          // シンボルサイズ
    int          sizeR;            // 縦シンボルサイズ(モジュール数)
    int          sizeC;            // 横シンボルサイズ(モジュール数)
    // 文字列 (fDecode==0 の場合は空文字列)
    int          strLng;           // 文字列長(バイト数)。終端の NULL は含めない
    char         str[IA_DMX_MAXSTRBUFF]; // 文字列
    // データコード語データ部 (fDecode==0 の場合は空)
    int          dataCodeDataLng;
    char         dataCodeData[IA_DMX_MAXCORDWORDSADATA];
    // 構造的接続
    // (未サポート)
} IA_DMX_Decode_Rslt;
```

復号処理結果を格納します。

関数 IA_QR_GetRslt (3.2.4 節) より出力されます。