



Colorbit SDK

# Colorbit SDK Programming Manual

2012-05-14

Version 1.1.3

ビーコア株式会社発行

本ドキュメントの内容は、機密情報であるため、  
厳重な取り扱い、管理を行ってください。

## 目次

1	本書について.....	3
2	概要 .....	4
2.1	モジュール説明.....	4
2.1.1	名前空間 mw::cb .....	4
2.1.2	名前空間 mw::cb::Decoder.....	4
2.2	ビルド環境設定.....	4
2.2.1	インクルードパスの追加 .....	4
2.2.2	ライブラリファイルの追加 .....	4
3	Colorbit SDK の使用ガイド .....	5
3.1	インクルードファイル.....	5
3.2	初期化と終了処理.....	5
3.3	パラメータ設定 .....	6
3.3.1	デコードパラメータ .....	6
3.3.2	色抽出位置設定.....	6
3.4	Colorbit 認識.....	8
4	サンプルデモ .....	10
4.1	Simple.....	10
4.2	cbit.....	10
5	改訂履歴 .....	11



図 1	色抽出位置設定の例 .....	7
図 2	入力画像データフォーマット .....	8
図 3	色分離の結果データフォーマット .....	9
図 4	NARライブラリによる入力データ作成 .....	10

# 1 本書について

本書は Colorbit SDK for CTR API を使用した基本プログラミングについて説明するプログラミングガイドです。

本書の内容は C/C++ プログラミングに精通した読者を想定しています。

Colorbit SDK API の詳細な仕様については「HTML 関数リファレンスマニュアル」をご参照ください。

Colorbit の概要については「CTR\_Colorbit\_Application\_DevelopGuide.pdf」をご参照ください。

## 2 概要

Colorbit ライブラリは Colorbit を利用した ID 認識のために用意された SDK です。

本ライブラリを用いる事で Colorbit の ID (1050 通り) を画像データより取得する事が可能になります。

Colorbit の作成については「CTR\_MW\_Colorbit\_Application\_DevelopGuide.pdf」内の第 3 章 Colorbit の制作を参照してください。

この章では Colorbit ライブラリを使用する上でのモジュール説明、またアプリケーションのビルドに必要な情報を説明します。

### 2.1 モジュール説明

#### 2.1.1 名前空間 mw::cb

この名前空間では Colorbit API から返されるエラーコード、または設定するパラメータ値のみを定義しています。

#### 2.1.2 名前空間 mw::cb::Decoder

この名前空間では入力画像より Colorbit の認識を行うための関数群が定義されています。

### 2.2 ビルド環境設定

#### 2.2.1 インクルードパスの追加

コンパイラのインクルードパスに“\$(CTRMW\_CBIT\_ROOT)/include/mw/cb”を追加する必要があります。

##### コード 2-1 インクルードパスの追加

```
INCLUDES += $(CTRMW_CBIT_ROOT)/include/mw/cb
```

※コード例は CTR-SDK のビルドシステムを使用した場合に OMakefile に記述すべき内容です。

#### 2.2.2 ライブラリファイルの追加

リンカに渡すべきライブラリファイルは「libmw\_cb\_decoder.\*.a」(“\*”は“fast”または“small”)です。

##### コード 2-2 ライブラリファイルの追加

```
LIBFILES = $(addprefix $(CTRMW_CBIT_ROOT)$(DIRSEP)libraries$(DIRSEP)$(config.  
getTargetSubDirectoryture)$(DIRSEP), libmw_cb_decoder
```

※コード例は CTR-SDK のビルドシステムを使用した場合に Omakefile に記述すべき内容です。

## 3 Colorbit SDK の使用ガイド

この章では、Colorbit ライブラリを使用したアプリケーションの実装方法を処理の流れに沿って説明します。

また本ライブラリにより提供される API は非スレッドセーフになっています。

### 3.1 インクルードファイル

Colorbit SDK の機能を使用するには、「cbDecoder.h」ファイルをインクルードする必要があります。

#### コード 3-1 cbDecoder.h をソースコードにインクルード

```
#include "cbDecoder.h"
```

### 3.2 初期化と終了処理

Colorbit の認識機能を使用する際、事前に Initialize メソッドを呼び出す必要があります。このメソッドは、ライブラリが内部で使用するワークメモリ割り当て/割り当て解除のための関数を備えています。

#### コード 3-2 Colorbit 初期化の例

```
void* Malloc(size_t size);  
void* Free(void* ptr);  
  
mw::cb::Decoder::Initialize(Malloc, Free);
```

終了時、Finalize メソッドを呼び出します。

#### コード 3-3 Colorbit 終了処理の例

```
mw::cb::Decoder::Finalize();
```

また実際に使用されるワークメモリサイズは GetWorkSize メソッドに入力画像のサイズを指定することで取得できます。

#### コード 3-4 ワークサイズ取得の例

```
int image_width; //input image width  
int image_height; //input image height  
  
size_t worksize = mw::cb::Decoder::GetWorkSize(image_width, image_height);
```

## 3.3 パラメータ設定

Colorbit の認識機能を使用する際、事前にパラメータを設定することで使用するデザイン、用途に合わせた利用が可能になります。

### 3.3.1 デコードパラメータ

撮影環境の影響などにより、入力画像が暗い事が予想される場合、mw::cb::IMAGE\_CORRECTION により暗所補正の ON/OFF を設定する事ができます。

※暗所補正処理は入力画像が十分明るい場合でも機能します。その場合認識率に大きな影響はありませんが、処理時間が増加します。事前に撮影環境を明るい場所に固定できる場合は、暗所補正処理を OFF とすることを推奨します。

また、Colorbit の認識時に使用するエラー訂正レベルを mw::cb::EC\_LEVEL により 0 ～ 2 の 3 段階で設定できます。

エラー訂正レベル 0 : エラー訂正無し

エラー訂正レベル 1 : 1 箇所の色間違いを訂正します。

エラー訂正レベル 2 : 2 箇所の色間違いを訂正します。

#### コード 3-5 デコードパラメータ設定の例

```
//暗所補正 OFF , エラー訂正レベル 2
mw::cb::Decoder::SetParams(mw::cb::DECODER_NONUSE_IMAGE_CORRECTION,
                             mw::cb::CODING_LD5_C15_CORRECTION_LEVEL_2);
```

### 3.3.2 色抽出位置設定

Colorbit は 15 の色配列により表現されています。

この 15 個の色取得位置を SetColorLocation メソッドを使用する事で、使用するデザインに合わせた設定が可能になります。

指定する座標系は左上頂点を原点とし、横方向を第 1 要素、縦方向を第 2 要素とし各抽出点を指定します。

座標値は入力画像内での絶対座標を指定して下さい。

また mw::cb::Decoder::QueryImage では location\_table の要素順に色配列を作成します。

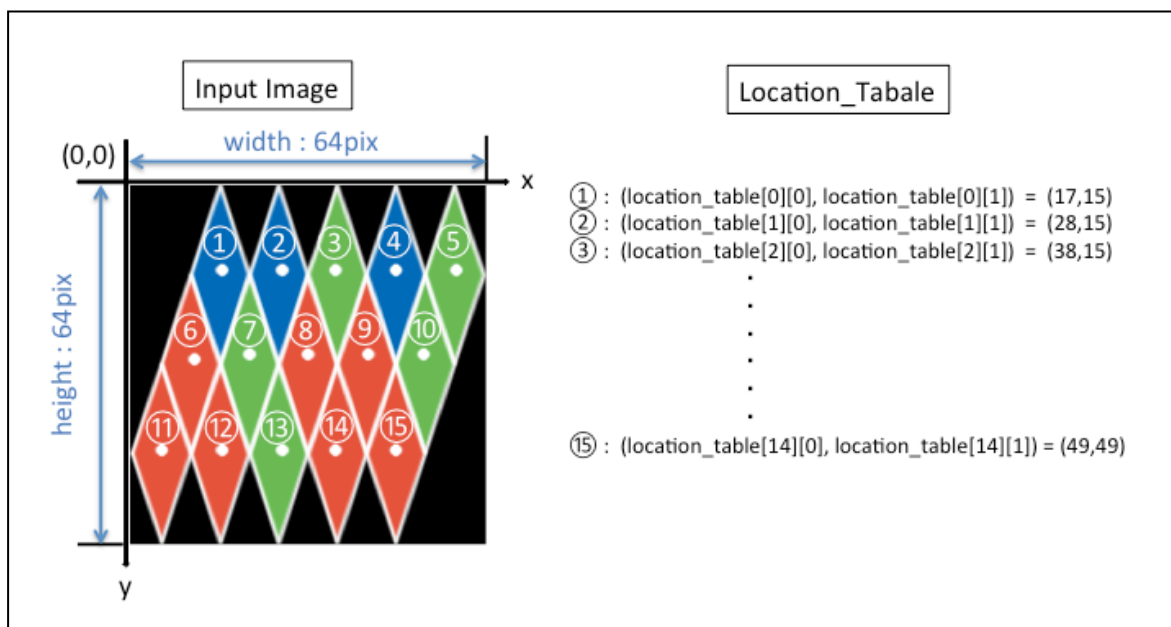
サンプルコードを 64 × 64 ピクセルの画像で捕らえる場合、色抽出位置の設定は下記ようになります。

### コード 3-6 色抽出位置設定の例

```
int location_table[15][2] = {
    {17,15},{28,15},{38,15},{49,15},{60,15},
    {12,32},{22,31},{33,31},{44,31},{54,31},
    {6,49},{17,49},{28,49},{38,49},{49,49}
};

mw::cb::Decoder::SetColorLocation(location_table);
```

図 1 色抽出位置設定の例



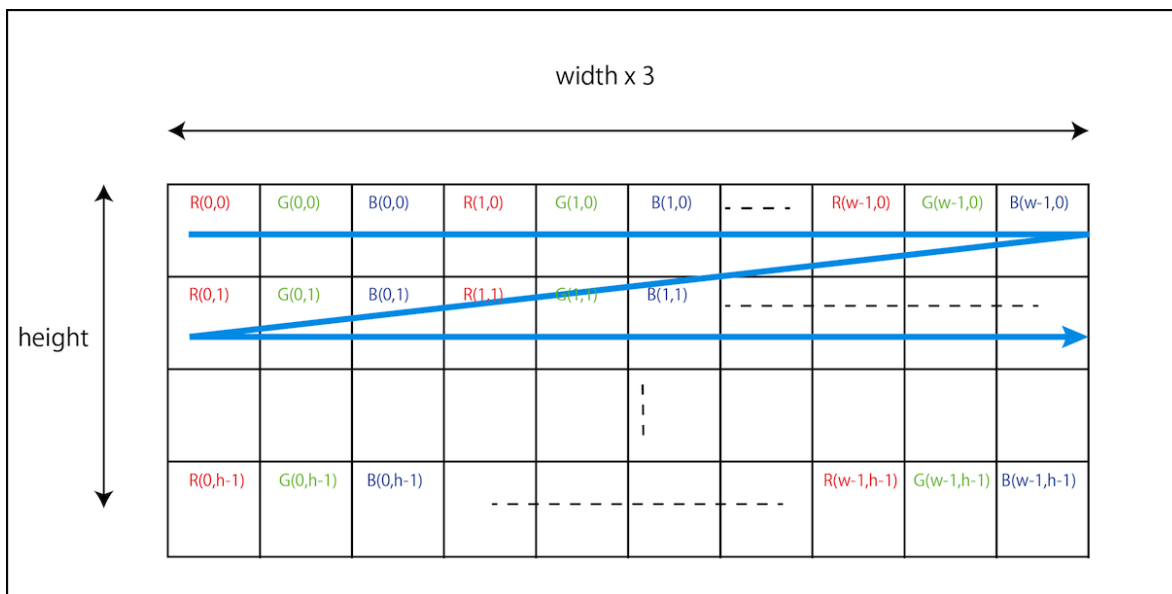
Colorbit のデザインを新たに作成した場合、デザインに応じて色抽出位置を設定する必要があります。

### 3.4 Colorbit 認識

パラメータを設定した後、Colorbit が捉えられている画像データを `mw::cb::Decoder::QueryImage` メソッドに入力することで、ID 認識を行います。

入力画像データはピクセルフォーマット RGB888 のリニアフォーマットを使用します。

図 2 入力画像データフォーマット



`mw::cb::Decoder::QueryImage` メソッドにより取得した色配列は下記のように表現されています。

R = 0、G = 1、B = 2、NO\_COLOR = 255

#### コード 3-7 Colorbit 認識の例

```
(void*) image_data; //Pixel Format = RGB888
int image_width;
int image_height;

//Colorbitのid取得にunsigned long long型(64bit)のデータ領域を確保
unsigned long longcb_id;
//Colorbitの色配列取得にunsigned char型(8bit)* 15のデータ領域を確保
unsigned char color_seq[15];

int error = mw::cb::Decoder::QueryImage(image_data, width, height,
                                         &cb_id, &color_seq);

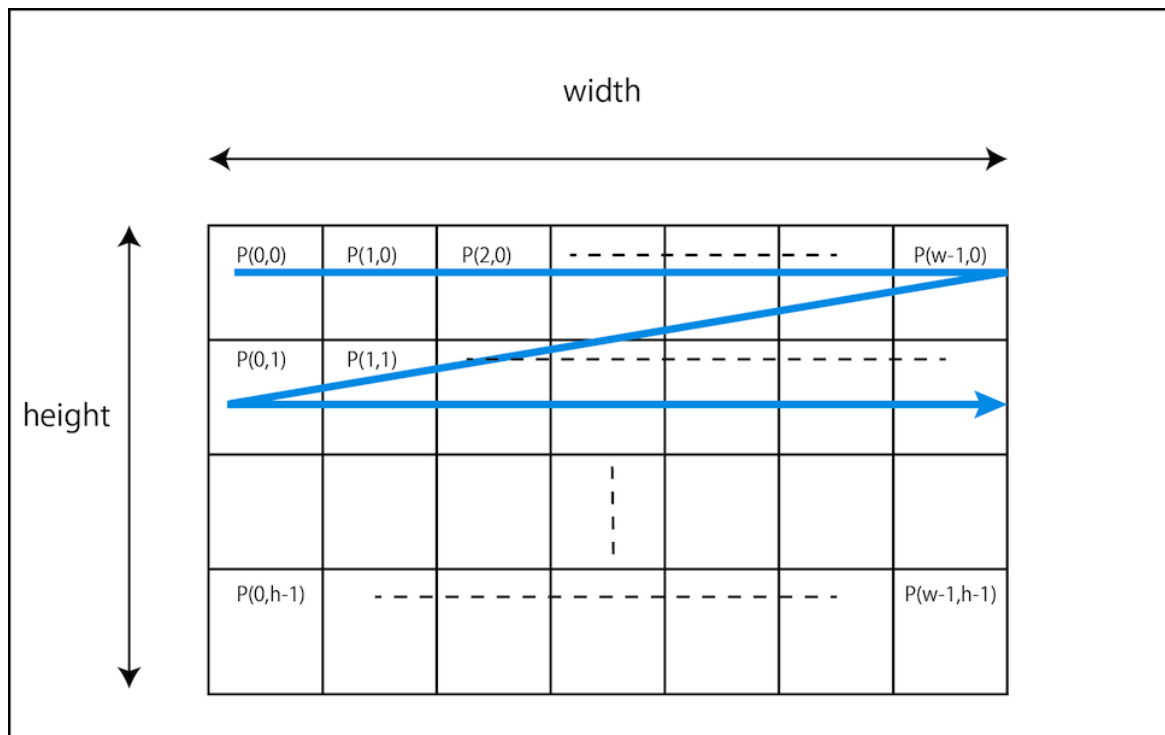
if(error < 0){
    //Colorbit Decode Error
}else{
}
}
```



また `mw::cb::Decoder::QueryImageEx` メソッドを使用する事で、入力画像全体の色分離の結果を取得する事ができます。

色分離の結果は、 $R = 150$ 、 $G = 100$ 、 $B = 50$ 、 $NO\_COLOR = 0$  の 8bit 画像データで表現されます。

**図 3 色分離の結果データフォーマット**



**コード 3-8 Colorbit 認識の例 2**

```
unsigned char* color_dump;
color_dump =
(unsigned char*)malloc(sizeof(unsignedchar)*image_width*image_height);

int error = mw::cb::Decoder::QueryImageEx(image_data, width, height,
                                           &cb_id, &color_seq, color_dump);

free(color_dump);
```

## 4 サンプルデモ

### 4.1 Simple

サンプルデモ Simple ではカメラからの画像取得を行い、上画面に表示されているカメラプレビュー内の枠内・色抽出位置にColorbitを正しい向きで捕らえる事でColorbitの認識を行います。

### 4.2 cbit

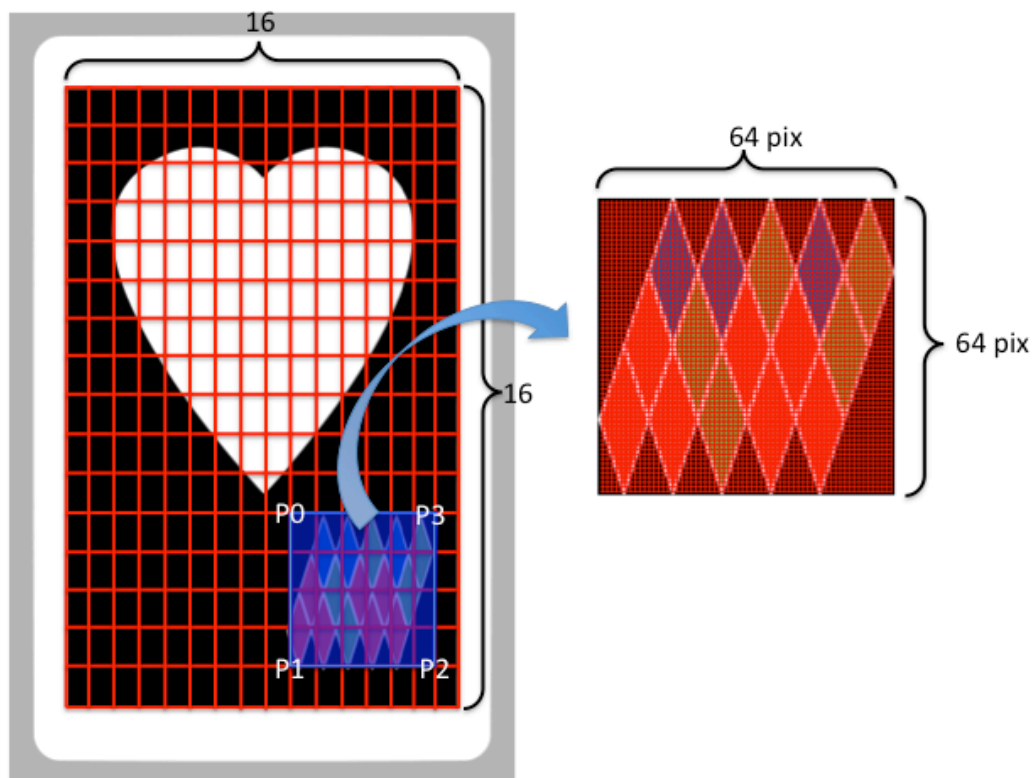
サンプルデモ cbit では NAR ライブラリを使用し、Colorbit の認識を行います。

本ライブラリでは画像中からColorbitの位置・向きを検出する機能は用意されていないため、NAR ライブラリを用いる事でマーカの検出を行い、検出された領域内部の情報をを用い Colorbit の認識を行います。

具体的な処理の流れは以下のようになります。

- 1) NAR ライブラリによりマーカの検出を行い、得られた領域を 縦横それぞれ 16 分割します。
- 2) 16 分割した点から適切な 4 点を指定し、その内部をさらに 64 分割し交点の色情報を取得します。
- 3) 得られた  $64 \times 64$  ピクセルのデータを入力データとし、Colorbit の認識を行います。

図 4 NARライブラリによる入力データ作成



## 5 改訂履歴

版	改訂日	分類	改訂内容
1.1.3	2012-05-14	－	・バージョンナンバーの変更
1.1.2	2012-04-04	追加	・4サンプルデモ
		変更	・3.3.2色抽出位置設定 サンプルコードでの設定例を追記
1.0.1	2012-02-28	－	・バージョンナンバーの変更
1.0.0	2011-12-22	－	・初版

Colorbit はビーコア株式会社の登録商標です。

その他、記載されている会社名、製品名等は、各社の登録商標または商標です。

©2012 Nintendo

ビーコア株式会社の許諾を得ることなく、本書に記載されている内容の一部あるいは全部を無断で複製・複写・転写・頒布・貸与することを禁じます。