

patternlib 仕様書

25G1065 塩澤匠生

2025 年 7 月 14 日

概要

本書は、C 言語ライブラリ ‘patternlib’ の仕様を記述したものである。本ライブラリは、特定の構造を持つ 3 次元配列のデータに基づき、文字や記号を組み合わせたアスキーアートを端末（ターミナル）上に出力する機能を提供する。出力するアートの拡大・縮小、配色、縦横の表示方向など、多彩な表示オプションを指定できることを特徴とする。

目次

1	モジュールの概要	2
1.1	機能	2
2	設定	2
2.1	ビルド時設定	2
2.2	実行時設定	2
3	関数の説明	2
3.1	データ構造	2
3.2	関数概要	3
3.3	関数の詳細	3
3.3.1	show_array	3
3.3.2	show_array_vertical_scaled	4
3.3.3	show_array_horizontal_scaled	4
3.3.4	print_char	4
3.3.5	print_char_colored	5
4	使用例	5

1 モジュールの概要

本ライブラリは、コンソールアプリケーションにおいて、テキストベースで豊かな表現を行うことを目的として開発された。数値データを視覚的なパターンに変換して表示することにより、プログラムの出力をより直感的で分かりやすいものにするを目指す。

1.1 機能

‘patternlib’ は、主に以下の機能を提供する。

- 3次元配列で定義されたパターンの描画
- 描画文字、拡大率、配色の指定
- 複数文字パターンの縦連結および横連結表示

2 設定

2.1 ビルド時設定

ヘッダファイル ‘patternlib.h’ で定義される定数について説明する。これらの値を変更してライブラリを再コンパイルすることで、1文字あたりのデフォルトの描画サイズを調整できる。

COL_SIZE 1文字を描画する際の、基本となる列数（横幅）を定義する。デフォルト値は8。

ROW_SIZE 1文字を描画する際の、基本となる行数（縦幅）を定義する。デフォルト値は8。

2.2 実行時設定

ライブラリの動作は、‘show_array’関数に渡す引数によって実行時に制御される。詳細は「関数の説明」のセクションを参照。

3 関数の説明

3.1 データ構造

本ライブラリが描画に用いるパターンは、以下の構造を持つ3次元整数型配列で定義される。

```
int pattern[文字数][ROW_SIZE][COL_SIZE];
```

配列の各要素の値は、ピクセルごとの描画状態を意味する。

- 0: 背景（空白）
- 1以上: 描画ピクセル。色指定時にはANSIエスケープシーケンスのカラーコード（1-7）として解釈される。

3.2 関数概要

表 1 関数一覧

関数名	説明
show_array	3 次元配列データに基づき、パターンを描画する。
show_array_vertical_scaled	(内部関数) パターンを縦に並べて拡大表示する。
show_array_horizontal_scaled	(内部関数) パターンを横に並べて拡大表示する。
print_char	(内部関数) 1 文字をターミナルに出力する。
print_char_colored	(内部関数) 1 文字を色付きでターミナルに出力する。

3.3 関数の詳細

3.3.1 show_array

指定されたパターン配列に基づき、ターミナルに文字を描画する主要な関数。

```
void show_array(  
    int array[][ROW_SIZE][COL_SIZE],  
    char _char,  
    int len_c,  
    int scale,  
    int color_flag,  
    int horizontal_flag  
);
```

■詳細 本関数は、‘array’に格納されたパターン情報に従って、指定された文字 ‘_char’ を用いてターミナル上に図形を描画する。‘scale’引数で拡大表示、‘color_flag’で色付け、‘horizontal_flag’で複数パターンの表示方向（縦/横）を制御できる。

表 2 show_array 関数の引数

引数名	説明
array	描画するパターンを格納した 3 次元配列。
_char	パターンの描画に用いる文字。
len_c	array に含まれる文字パターンの総数。
scale	描画の拡大率。1 以上の整数を指定する。
color_flag	カラー表示の有効化フラグ。0 以外を指定すると、配列の値を ANSI カラーコードとして解釈し、色付きで描画する。
horizontal_flag	描画方向の指定フラグ。0 以外を指定すると複数の文字パターンを横に並べて描画する。0 の場合は縦に並べる。

■引数

3.3.2 show_array_vertical_scaled

複数の文字パターンを縦に連結し、指定された倍率で拡大して表示する。

```
void show_array_vertical_scaled(  
    int array[][ROW_SIZE][COL_SIZE],  
    char _char,  
    int len_c,  
    int scale,  
    int color_flag  
);
```

■詳細 ‘show_array’から ‘horizontal_flag’が 0 の場合に呼び出される内部関数。文字パターンを縦方向に並べて描画する。

■引数 ‘show_array’関数と同様（‘horizontal_flag’を除く）。

3.3.3 show_array_horizontal_scaled

複数の文字パターンを横に連結し、指定された倍率で拡大して表示する。

```
void show_array_horizontal_scaled(  
    int array[][ROW_SIZE][COL_SIZE],  
    char _char,  
    int len_c,  
    int scale,  
    int color_flag  
);
```

■詳細 ‘show_array’から ‘horizontal_flag’が 0 以外の場合に呼び出される内部関数。文字パターンを横方向に並べて描画する。

■引数 ‘show_array’関数と同様（‘horizontal_flag’を除く）。

3.3.4 print_char

1 ピクセルに相当する文字をターミナルに 1 文字表示する。

```
void print_char(int flag, char _char);
```

■詳細 ピクセルデータ（‘flag’）に基づき、指定された文字（‘_char’）または空白を出力する。‘show_array’から呼び出される低レベル描画関数。

表 3 print_char 関数の引数

引数名	説明
flag	ピクセルデータ。0 の場合は空白を出力し、それ以外の場合は ‘_char’ を出力する。
_char	描画に用いる文字。

■引数

3.3.5 print_char_colored

1 ピクセルに相当する文字をターミナルに 1 文字、色付きで表示する。

```
void print_char_colored(int flag, char _char, int color);
```

■詳細 ピクセルデータ (‘flag’) に基づき、指定された文字 (‘_char’) または空白を、指定色 (‘color’) で出力する。

表 4 print_char_colored 関数の引数

引数名	説明
flag	ピクセルデータ。0 の場合は空白を出力し、それ以外の場合は色付きで ‘_char’ を出力する。
_char	描画に用いる文字。
color	ANSI エスケープシーケンスのカラーコード (0-7) を指定する。

■引数

4 使用例

以下に、‘patternlib’ ライブラリを使用して簡単なパターンを描画する C 言語のコード例を示す。

```
#include <stdio.h>
#include "patternlib.h"

int main(void) {
    // 'A' の文字パターンを定義 (8x8)
    int pattern_A[1][ROW_SIZE][COL_SIZE] = {
        {
            {0,0,0,1,1,0,0,0},
            {0,0,1,0,0,1,0,0},
        }
    };
}
```

```

        {0,1,0,0,0,0,1,0},
        {0,1,1,1,1,1,1,0},
        {0,1,0,0,0,0,1,0},
        {0,1,0,0,0,0,1,0},
        {0,0,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,0}
    }
};

printf("--- デフォルト表示 ---\n");
show_array(pattern_A, '#', 1, 1, 0, 0);

printf("\n--- 2倍拡大表示 ---\n");
show_array(pattern_A, '*', 1, 2, 0, 0);

// 色付きのパターンを定義（値2を緑色とする）
int pattern_color[1][ROW_SIZE][COL_SIZE] = {
    {
        {0,2,2,2,2,2,2,0},
        {0,2,0,0,0,0,2,0},
        {0,2,2,2,2,2,2,0},
        {0,2,0,0,0,0,2,0},
        {0,2,0,0,0,0,2,0},
        {0,2,0,0,0,0,2,0},
        {0,2,0,0,0,0,2,0},
        {0,2,0,0,0,0,2,0},
        {0,0,0,0,0,0,0,0}
    }
};

printf("\n--- カラー表示 ---\n");
show_array(pattern_color, ' ', 1, 1, 1, 0);

return 0;
}

```