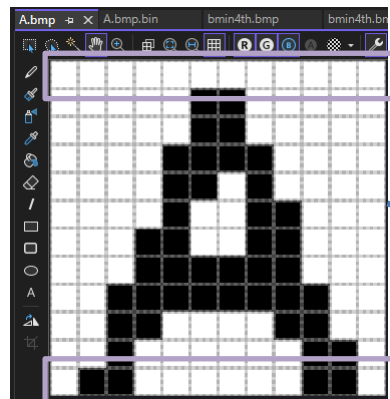
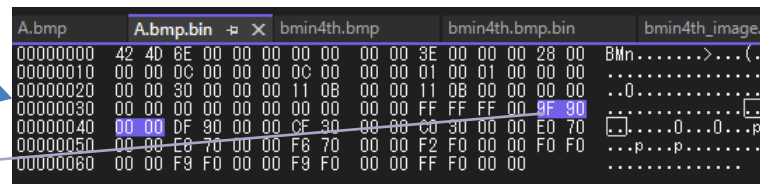


モジュール設計仕様書	システム	bitope	Bit-mapオペレーション	作成日	2025/02/04
補足	モジュール	ex_bitope	Bit-mapを操作する	作成者	HosiYamaKaze

DFD:ex\_bitope.cpp (1) バイトオーダーと白黒変換、bm\_{load,conv,save} ()



~~bm\_load()  
bitope  
(Windows)~~

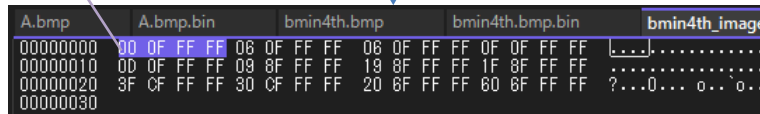


Sourec code

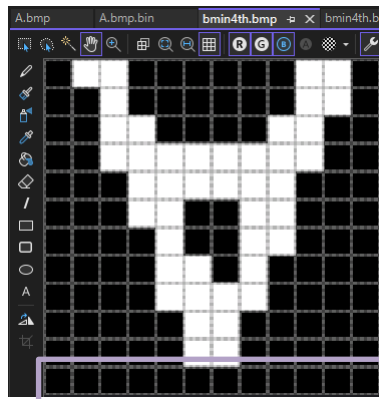
<https://github.com/hosiyamakaze/bitope>

Windows上のBitmapは、例えば'A'と画面表示される場合を分析すると、内部データは上下反転、白(=1)黒(=0)反転しているように見える。

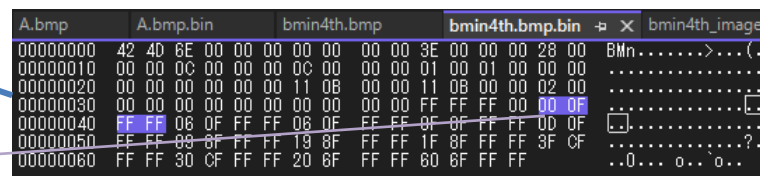
bm\_conv()  
bitope  
(Windows)



1. bitopeモジュールは、画面イメージと同様の上下順、白(=0)黒(=1)として内部データを扱う。
2. そのためにはbm\_create()で、BM\_PAD32, BM\_4TH, BM\_MSBFirst, BM\_ONを指定する。
3. ヘッダを付けずにbm\_save()するとイメージ本体データをファイルに保存する。



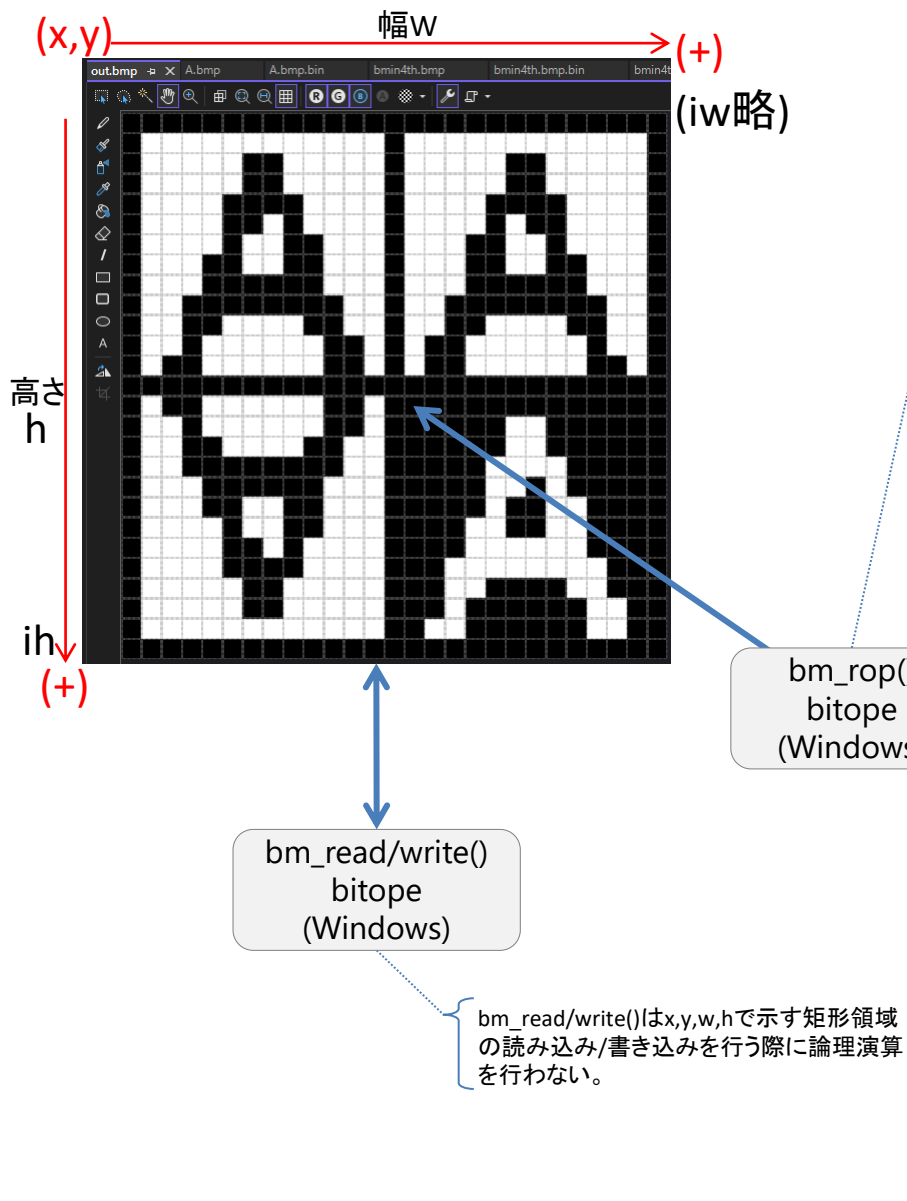
bm\_save()  
bitope  
(Windows)



1. Bitmap用のヘッダを付けてbm\_save()すると、画面には上下反転、白黒反転しているように表示される。
2. bm\_conv()はこの変換を実行するが、Y方向X方向の反転を指定することもできる。

モジュール設計仕様書	システム	bitope	Bit-mapオペレーション	作成日	2025/02/04
補足	モジュール	ex_bitope	Bit-mapを操作する	作成者	HosiYamaKaze

DFD:ex\_bitope.cpp (2) Bit-map操作、bm\_{rop,read,write}()



モジュール設計仕様書	システム	bitope	Bit-mapオペレーション	作成日	2025/02/04
補足	モジュール	ex_base64	Bit-mapを使ってbit列を操作する	作成者	HosiYamaKaze

DFD:全体像(ex\_base64.cpp:高さh=1のbit-mapとして、bit列を操作する)

Enter source text (for Enter to Base64): 0123abcABC  
Base64 Encode: MDEyM2FiY0FCQw==  
Base64 Decode: 0123abcABC  
Enter source text (for Enter to Base64):  
Enter Base64 text (for Enter to exit): MDEyM2FiY0FCQw==  
Base64 Decode: 0123abcABC  
Base64 Encode: MDEyM2FiY0FCQw==  
Enter source text (for Enter to Base64):  
Enter Base64 text (for Enter to exit):  
Done.

1. 入力されたtextのbit列を高さh:1のビットマップを利用して、幅w:6bitまたは8bitで取り扱う。
2. Bit操作例を示すため、下記bitmapイメージ化した。

