bmp2c documentation

Original file name:
20070911_bmp2c_v0002_doc_en.pdf

This document is relevant for bmp2c version 0.00.2 (alpha version), please visit www.nimp.co.uk/software for other versions, bug reports and feature requests.

Terms of licence

Overview

bmp2c is a simple win32 console utility which converts a bitmap file into a c/c++ source file. The picture is stored in an array. The type of the array depends on settings supplied by the user. This kind of utility is useful for embedded systems developers as it provides a portable way to include pictures in their application.

bmp2c has two major features:

- Configurable encoding: the user can specify any kind of encoding for the array.

- Preview feature: the user can preview the results immediately after the conversion, without running the code on the target and even before the hardware is designed!

In short, bmp2c converts a bmp file into a c file and optionally generates another bmp file which represents what you will see on your target hardware.

Installation

bmp2c is a self sufficient executable, just cut and past “bmp2c.exe” wherever you want on your system and your are done !

Requirements

bmp2c runs on any win32 system (As sources are provided you can easily recompile to your favorite platform of needed).
To use bmp2c, you need the following files:

- “bmp2c.exe”: well…
- A bitmap file: you can generate such file by editing a picture with paint and then save it using the “save as” command and by selecting “24-bit Bitmap (*.bmp; *.dib)” as “save as type”. 8bit, 24bit and 32bit encodings are supported.
- An “ini” file: this file is a simple text file where you describe what you expect from the tool.

Usage

The following convention is used to describe the command line:
<mandatory argument>
[optional argument]

Command line

```
bmp2c <name of the ini file> <name of the input bmp file> [name of the output source file]
```

Arguments description

<name of ini file>: must be the full name to a valid bmp2c ini file.
<name of the input bmp file>: must be the full name to a bitmap file.
[name of the output source file]: must be a full file name which leads to an existing directory where write access are granted. If this argument is not provided, the output file name is "<name of the input bmp file>_out.c".

Remarks

- If the preview feature is used, a bmp file is created in the same directory as the output source file. The name of this file is "<name of the output source file>_preview.bmp".
- You may want to write a batch file which converts all the bmp file of one folder. A basic example of such batch file is provided in this package.

How to write ini files

Syntax

The ini file is a text file with a special syntax:

- Everything written before “[bmp2c]” is ignored.
- Each parameter definition begins with ‘=’.
- Only the order and the number of parameters matters, everything before the ‘=’ character of parameters n (and after the last characters consumed during parameter n-1 reading) is ignored. This allow users to comments their ini files:

```
[a comment about this parameter] [parameter name reminder] = <parameter value> [comment2]
```

When the parameter is a string, all the characters after ‘=’ are read therefore the comments in place of “[comment2]” are not possible.

The string “#bmp2c_input_full_file_name#” will be replaced by the name of the input bitmap file, including the path if it is provided.

The string “#bmp2c_input_file_name#” will be replaced by the name of the input bitmap file.

The string “#bmp2c_input_file_name_id#” will be replace by a valid c/c++ variable name based on the name of the input bitmap file (spaces, quote and other prohibited characters are replaced by ‘_’).

Parameters list

The order of parameters matters, all parameters are mandatory.

min_version

Unsigned integer which specifies the minimum version of bmp2c required to process properly this ini file. This document describes the structure of ini files for bmp2c version 0 and compatible future releases. If a future version breaks backward compatibility, it will detect the version mismatch and complain.

```
min_version =0
```

x_decl

String which contains the declaration of a c/c++ symbol equal to the width of the pictures. This parameter can be an empty string.

```
x_decl        =static const unsigned short x_#bmp2c_input_file_name_id# =
```

x_decl_postfix

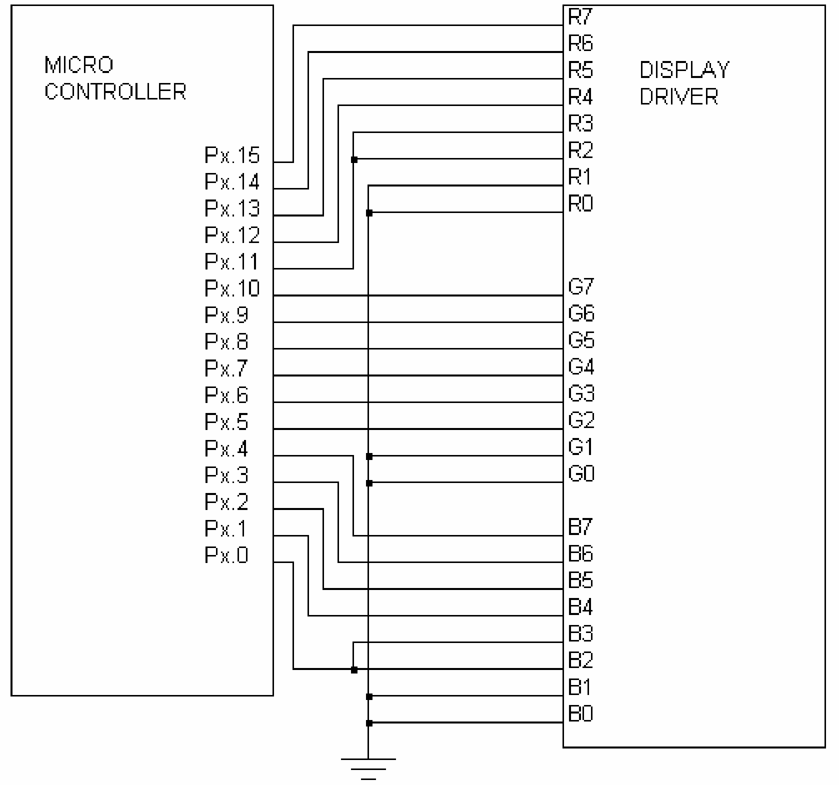
String which contains the end of the declaration for symbol x.
This parameter can be empty.

```
x_decl_postfix =;//width of the picture
```

y_decl, y_decl_postfix

Counterpart parameters for the height of the picture.

Examples of parameters below are given for the following hardware:



Those examples are also written in the sample ini file “NEC touch it.ini”

array_decl

String which contains the declaration of the array.

```
array_decl     =static const unsigned short pic_#bmp2c_input_file_name_id#[] =
```

data_size

Unsigned integer which specifies the number of bit per encoded pixel (size in bit of each element of the array).

```
data_size              =16
```

data_map

String which describes how to compute encoded pixels. It should contain not more than <data_size> bit definition. The following bit definitions are available:
r<x>: copy the bit x of the red component of the input bitmap. (0<=x<8).
R<x>: same as r<x> but the copied bit is complemented.
g<x>: copy the bit x of the green component of the input bitmap. (0<=x<8).
G<x>: same as g<x> but the copied bit is complemented.
b<x>: copy the bit x of the blue component of the input bitmap. (0<=x<8).
B<x>: same as b<x> but the copied bit is complemented.
a<x>: copy the bit x of the alpha (transparency) component of the input bitmap. (0<=x<8).
A<x>: same as b<x> but the copied bit is complemented.
c0: copy constant bit equal to 0.
c1: copy constant bit equal to 1.

```
data_map     = r7  r6  r5  r4  r3          g7  g6  g5  g4  g3  g2        b7  b6  b5  b4  b3
```

This tell the tool to store the 5 MSB of the red component into the 5MSB of the 16bit which represent the pixel, then the 6 green MSB and finally the 5 blue MSB.

preview_map

String which describe how to compute preview pixels. It should contain not more than 24 bit definition. The following bit definitions are available:
d<x>: copy the bit x of the encoded pixel. (0<=x< <data_size>).
D<x>: same as d<x> but the copied bit is complemented.
c0: copy constant bit equal to 0.
c1: copy constant bit equal to 1.

```
preview_map  = d15 d14 d13 d12 d11 d11 c0 c0 d10 d9 d8 d7 d6 d5 c0 c0 d4 d3 d2 d1 d0 d0 c0 c0
```

As you can see, d11 and d0 are repeated twice to reflect the wiring between the microcontroller and the display driver.

generate_preview_bmp

Boolean value (0 or 1). If 0, <preview_map> will not be processed (but must be present) and no preview bmp file will be generated.

pause

Boolean value (0 or 1). If 1, the program will ask the user to hit a key before returning control to the system.

Src_endl_param

String which specify how end of line should be encoded in the output source file. This parameter must be one of the following strings:

- "CR_LF" (Dos, win32 standard)
- "CR" (Mac)
- "LF" (Unix)