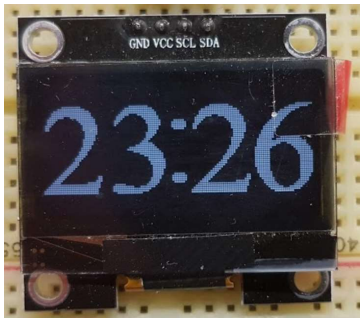
Bitmap Font Creator

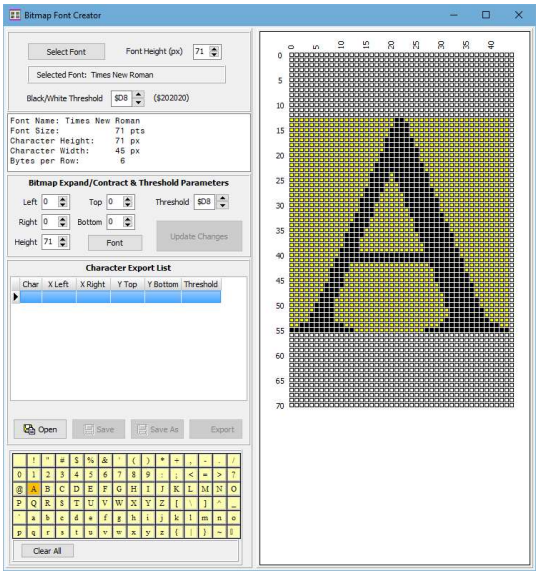
This application was created out of necessity. I was experimenting with small OLED displays (128x64) and needed characters which would, as much as possible, fill the display.



Creating a font is time consuming, a large one even more so. It also requires some considerable artistic flair which I don't have.

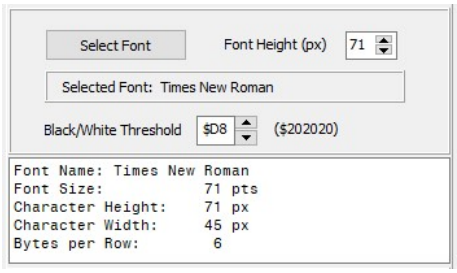
In my test the characters displayed are based on 71px high Arial with the space at the top and bottom trimmed down to 64px high. The actual active height is only 43px.

The BMFontCreator app will allow you to:

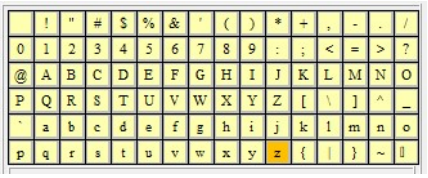


- Create bitmaps from a chosen list of characters
- Select font names and heights
- Allow you to trim or stretch the space around characters.
- Produce a header file containing the bitmaps.

Usage: The top panel allows you to select a font on which to base your bitmaps. It also allows you to select the height of the font in pixels. A Black/White threshold value is also provided.



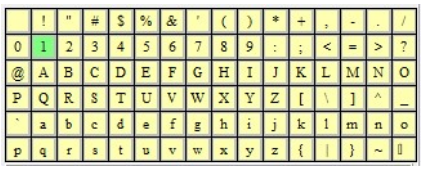
When rendering a font in black I discovered, much to my surprise, that not all of the rendered character is black. This is either shading or some side effect of scaling. The correct threshold value may require a little experimentation, start at around 80 hex. The character will be shown in the preview panel when you click on any character on the keyboard. The clicked character wil be highlighted as shown.



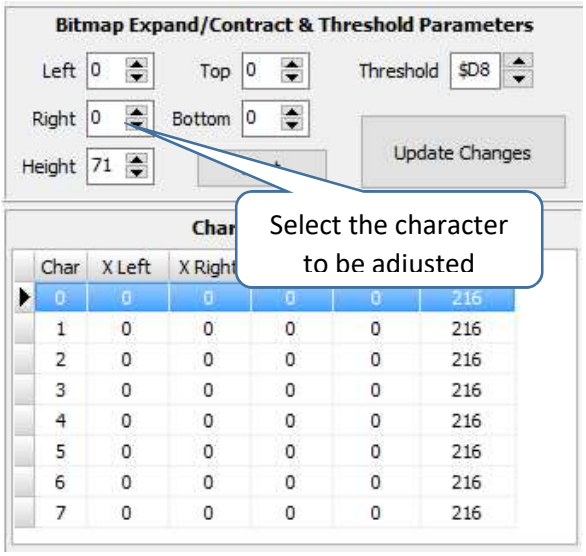
When creating an export list the settings for new characters are based on the first one added. When creating a new export list add the first character and make any adjustements. Once this is done add any additional characters.

Bitmap Font Creator

Initially the export list will be empty so in order to add characters to the list hold the control key down and click on the key to add. Characters appearing in the export list will be hightlighted in green on the keyboard. To remove a character from the list hold the control key down and click on the character. Should you wish to clear the export list click on the "Clear All" button



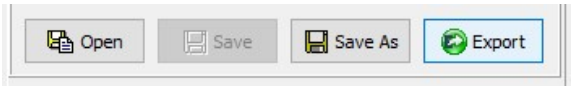
Once you have an export list you can then adjust the characters as necessary.



To adjust a character – First select it in the list.

Once selected you may adjust the space at the left, top, bottom and right of the character. The adjustment may be positive (row/columns added) or negative (rows/columns removed). The font name, height and Black/White threshold may also be changed as required.

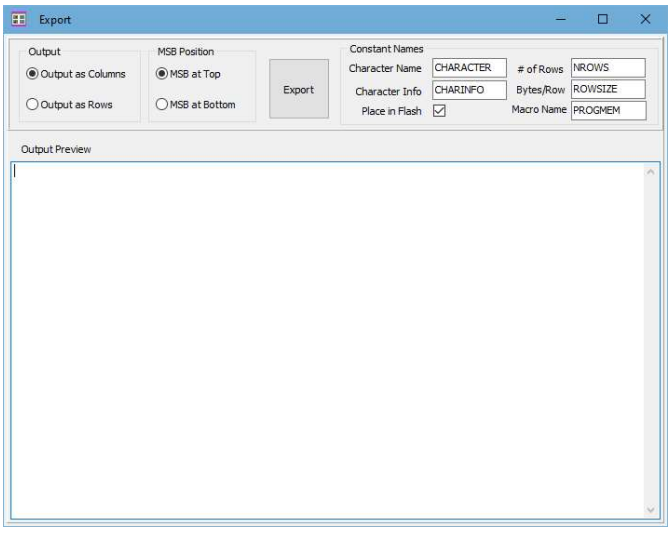
When adjustments are complete then press the "Update Changes" button, the result will then be displayed.



Saving and loading a list: When you create a list it will be unnamed. It is recommended that you save the list using the "Save As" button. Having once saved it then you may then just press the save button at any time in order to re-save it as changes are made. Already saved lists may be loaded using the "Open" button.

Exporting a list:

Once you have completed your list it may then be exported. Press the "Export" button, this will open the Export window.

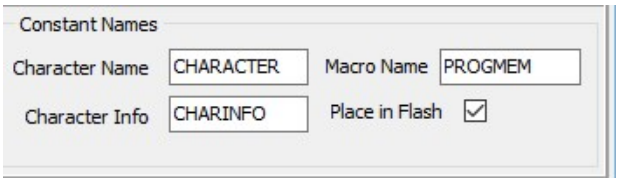


Settings:

Output – This allows you to order the characters in the header file by row or column. Some displays require their data to be presented as columns.

MSB at Top – Displays can have their column bytes with either the most significant bit (MSB) at the top or bottom. This setting is inoperative when "Output as rows" is selected.

The header file produced as the result of the export contains named constants. In order to reduce the risk of conflicting names within your code the constant names can be changed to suit.



Bitmap Font Creator

Character Name: e.g. const byte CHARACTER0[45][9] PROGMEM = {

The Character name here is "CHARACTER", this can be changed to suit. The following number is automatically appended.

Character Info: e.g. const byte CHARINFO[NCHARS][2] PROGMEM = {

The character information array contains column and row counts for each character. **The name shown here "CHARINFO" can be changed to suit.**

Place in Flash: Ticking this box allows a macro name to be inserted into the constant line. Upon compilation the macro allows for the arrays of characters to be stored in the microcontrollers Programme Flash memory rather than it's RAM.

Macro Name: This allows the name of the "In Flash" storage macro to be changed dependant up the microcontroller range. The default is "PROGMEM" (AVR's).

Other constants with fixed names:

BYCOLUMNS	0 = The output is ordered as rows 1 = The output is ordered as columns
COLINVERT	0 = The MSB is at the top of the bitmap column. 1 = The LSB is at the top of the column.
NCHARS	This is the number of characters in the list.

If you find this application useful but believe that it needs additional features or you have discovered a bug please contact me and I will endeavour to resolve the problem.

Version 2 will incorporate a bitmap editor.

You can contact me at:

meterman.accenttechnology@gmail.com

Bitmap Font Creator

```
/* ****************************************
 * File Sample.h generated at 13/12/2020 07:02:49
 * Total bytes used 294 (0x0126)
 * ****************************************
 *
 * File organised as Rows of Columns bytes
 *
 * ****************************************
 */
#define BYCOLUMNS   1 // Ordered by columns    1 = yes, 0 = no
#define COLINVERT   1 // Column bytes inverted 1 = yes, 0 = no
#define NCHARS      2   // characters in the array

// number of rows and columns in each character of the array
const byte CHARINFO[NCHARS][2] PROGMEM = {
       { 49,  22},
       { 49,  22}
};


const byte CHARACTER0[22][7] PROGMEM = {
// Times New Roman Height: 50px
// Character "0" Bytes: 147 (22x49 px)
/*   0 */ {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},
/*   1 */ {0x00, 0x00, 0xF0, 0x1F, 0x00, 0x00, 0x00},
/*   2 */ {0x00, 0x00, 0xFE, 0xFF, 0x01, 0x00, 0x00},
/*   3 */ {0x00, 0x80, 0xFF, 0xFF, 0x07, 0x00, 0x00},
/*   4 */ {0x00, 0xE0, 0xFF, 0xFF, 0x0F, 0x00, 0x00},
/*   5 */ {0x00, 0xF0, 0xFF, 0xFF, 0x1F, 0x00, 0x00},
/*   6 */ {0x00, 0xF8, 0x03, 0x80, 0x3F, 0x00, 0x00},
/*   7 */ {0x00, 0x38, 0x00, 0x00, 0x78, 0x00, 0x00},
/*   8 */ {0x00, 0x0C, 0x00, 0x00, 0x60, 0x00, 0x00},
/*   9 */ {0x00, 0x0C, 0x00, 0x00, 0x60, 0x00, 0x00},
/*  10 */ {0x00, 0x04, 0x00, 0x00, 0x40, 0x00, 0x00},
/*  11 */ {0x00, 0x04, 0x00, 0x00, 0x40, 0x00, 0x00},
/*  12 */ {0x00, 0x0C, 0x00, 0x00, 0x60, 0x00, 0x00},
/*  13 */ {0x00, 0x1C, 0x00, 0x00, 0x70, 0x00, 0x00},
/*  14 */ {0x00, 0x78, 0x00, 0x00, 0x3C, 0x00, 0x00},
/*  15 */ {0x00, 0xF8, 0xFF, 0xFF, 0x1F, 0x00, 0x00},
/*  16 */ {0x00, 0xF0, 0xFF, 0xFF, 0x1F, 0x00, 0x00},
/*  17 */ {0x00, 0xC0, 0xFF, 0xFF, 0x07, 0x00, 0x00},
/*  18 */ {0x00, 0x80, 0xFF, 0xFF, 0x03, 0x00, 0x00},
/*  19 */ {0x00, 0x00, 0xFC, 0x7F, 0x00, 0x00, 0x00},
/*  20 */ {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},
/*  21 */ {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}
       };
const byte CHARACTER1[22][7] PROGMEM = {
// Times New Roman Height: 50px
// Character "1" Bytes: 147 (22x49 px)
/*   0 */ {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},
/*   1 */ {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},
/*   2 */ {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},
/*   3 */ {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},
/*   4 */ {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},
/*   5 */ {0x00, 0x18, 0x00, 0x00, 0x40, 0x00, 0x00},
/*   6 */ {0x00, 0x18, 0x00, 0x00, 0x40, 0x00, 0x00},
/*   7 */ {0x00, 0x18, 0x00, 0x00, 0x60, 0x00, 0x00},
/*   8 */ {0x00, 0x38, 0x00, 0x00, 0x60, 0x00, 0x00},
/*   9 */ {0x00, 0xFC, 0xFF, 0xFF, 0x7F, 0x00, 0x00},
/*  10 */ {0x00, 0xFC, 0xFF, 0xFF, 0x7F, 0x00, 0x00},
/*  11 */ {0x00, 0xFC, 0xFF, 0xFF, 0x7F, 0x00, 0x00},
/*  12 */ {0x00, 0xFC, 0xFF, 0xFF, 0x7F, 0x00, 0x00},
/*  13 */ {0x00, 0x00, 0x00, 0x00, 0x60, 0x00, 0x00},
/*  14 */ {0x00, 0x00, 0x00, 0x00, 0x60, 0x00, 0x00},
/*  15 */ {0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00},
/*  16 */ {0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00},
/*  17 */ {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},
/*  18 */ {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},
/*  19 */ {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},
/*  20 */ {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00},
/*  21 */ {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}
       };
```