FTDI EVE Unicode Rendering

--------------------------

The FTDI EVE chips have several fonts in ROM, but these fonts only contain a

subset of ASCII characters. Notably, this excludes diacritics and accents

used in most Western languages.

While the FTDI EVE has the capability for user-defined fonts, such fonts only

support 127 character positions, making them as limiting as the built-in fonts.

As a further complication, high resolution TFT displays require high resolution

fonts. It is not feasible to put a complete international font into the limited

flash memory of most microprocessors.

To work around these limitations, this library uses a custom font renderer with

the following characteristics:

1) Rather than providing bitmaps for different font sizes, it uses a single

bitmap for the largest font size (romfont 31) and emulates other sizes by

scaling the bitmaps using BITMAP\_TRANSFORM.

2) Rather than loading an entire font, it combines symbols from romfont 31

with a limited number of symbols from a custom font. For accented letters,

the rendering code combine basic letter shapes from romfont 31 with

bitmaps containing only the accent themselves.

3) The custom bitmap is RLE compressed into PROGMEM. For accents, which have

a fairly small number of non-white pixels, the savings are significant.

These characteristics enable an alphabet for Western languages to be

synthesized from only a few dozen custom symbols and modest PROGMEM use (~10k)

The text layout is done by the code in "unicode.cpp" with the help of one of

more character renderers (e.g. "western\_char\_set.cpp"). Each character render

is responsible for loading the necessary bitmap data into RAMG and drawing

characters as requested.

To add symbols for other languages, it will only be necessary to make a bitmap

and implement a corresponding character renderer.