Application Note   AN2356

Graphics Library for OSRAM's OLED Displays

Author: Valeriy Kyrynyuk
Associated Project: Yes
Associated Part Family: All
PSoC Designer Version: 4.2
Associated Application Notes: AN2348

Abstract

This Application Note describes a graphics library for OLED display operation. Library functions include drawing, text, and bitmap operations.

Introduction

Modern designs often require a low-cost and high quality graphical display. OLED (organic light emitting diode) displays by OSRAM Opto Semiconductors (http://www.pictiva.com) meet these requirements and are the preferred displays for users and designers. This Application Note describes a library of functions to interface with a display that has a resolution of 96 x 48 pixels.

OLED Review

OSRAM OLED displays have the following characteristics:

o   They are monochrome and have 16 brightness levels.
o   The interface supply voltage is 2.4-3.5V.
o   The OLED supply voltage is 12-13V.
o   The operating temperature range is between -30°C and 70°C.
o   They have low power consumption at 220 mW maximum (OLED supply).
o   No additional lighting elements are required.
o   Lifetime at standard brightness is between 10 000 - 40 000 hours.
o   Parallel or serial (SPI) are the standard interface options.

Consider the display process in detail. An OLED display is a memory-mapped device. Four bits are necessary to display one pixel, because an OLED display has 16 brightness levels of one basic color; a single byte can hold information for two pixels. The columns and rows of the display can serve as the X and Y axes, respectively. One byte in video memory represents two pixels in the X direction. Therefore, to simplify algorithms for the display of images, the X coordinates of function arguments are assigned as one half of the real pixel position in its row. The pixel representation in a video memory byte is shown in Figure 1.
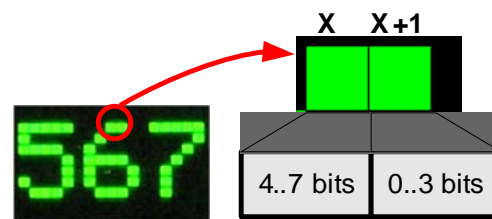
Figure 1. Video Memory Byte Format

Interfacing with PSoC®

The demonstration board is implemented using a CY8C29666 PSoC device. The interconnection circuit is shown in Figure 2. The connection is done by parallel interface. For a detailed description of this interface, refer to reference [1] from Solomon Systech Company at the end of this document. The maximum logic supply voltage of the display is 3.5V.
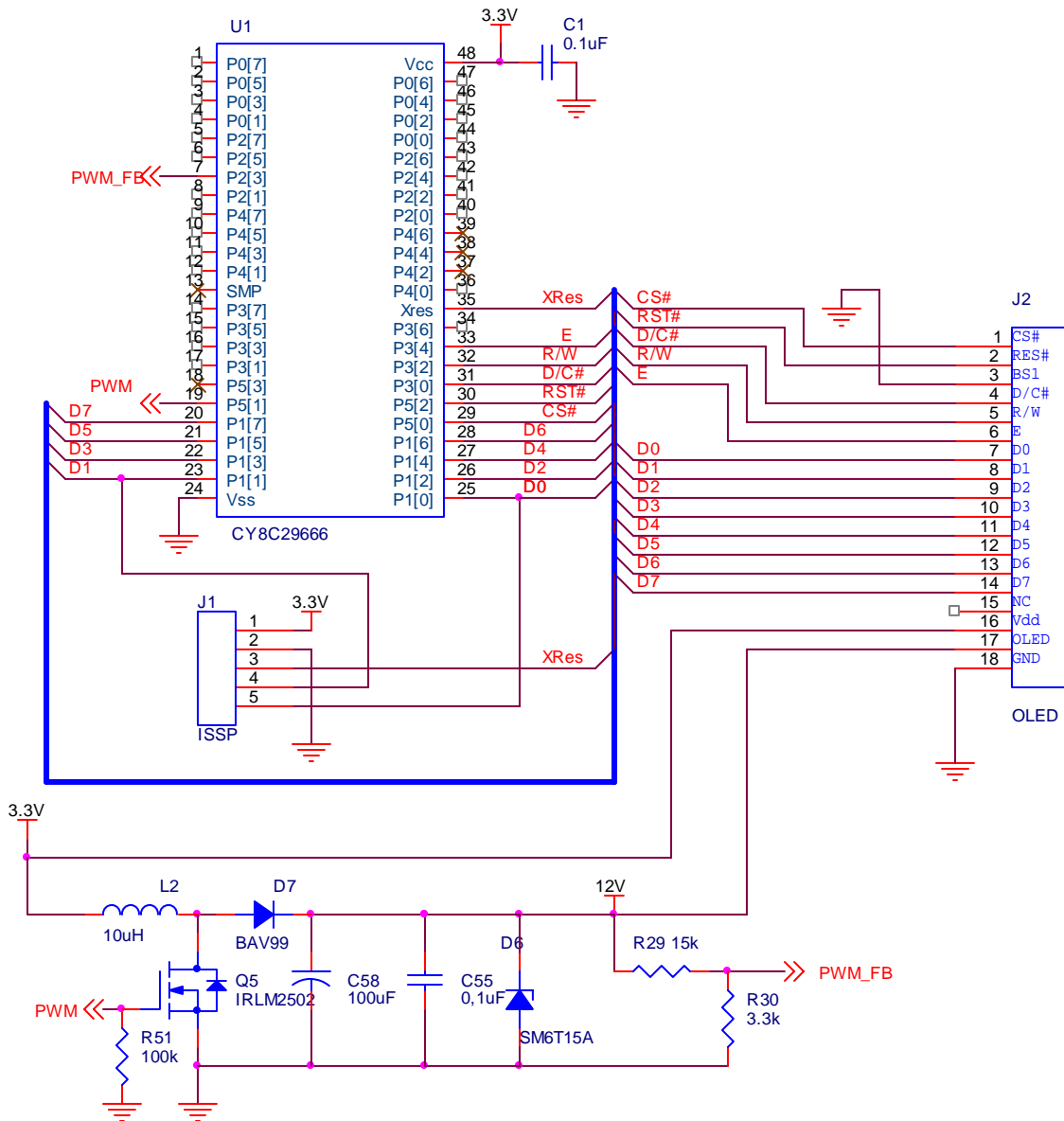
www.DataSheet4U.com

Figure 2. PSoC-OLED Interconnection Circuit, Vcc = 3.3V

When working with dynamic graphics, the device does not have enough of its own memory to buffer the contents of video memory. This requires about 3 Kbytes (96x64/2) of RAM. The data must be read from the video memory of the display. Therefore, the D0..D7 lines should be bidirectional.

The user can choose other pins for OLED support. In this case, within the Device Editor of PSoC Designer™, the PSoC-OLED signal should be configured to other pins. After that, in the *oled.inc* file the port numbers and bit template for the interface signals must be assigned. It is important that wires D0..D7 be located on the same port without changing the order in which the pins are numbered.

www.DataSheet4U.com

OLED Driver Library Description

The library functions can be symbolically grouped into two levels: low and high. The low-level APIs implement simple functions for display operation, which include transmitting commands, writing and reading data, and transmitting data arrays from the PSoC memory to display memory. Table 1 lists these functions.

For image drawing, a special structure of image data was implemented. The argument *Buf* of the *oledWriteROMSprite* function points to the following structure:

```
struct Sprite{
        WORD    Xpos=0x03;
        BYTE    Xcmd=0x15;
        BYTE    Xstart;
        BYTE    Xend;
        WORD    Ypos=0x03;
        BYTE    Ycmd=0x75;
        BYTE    Ystart;
        BYTE    Yend;
        WORD    DataCount;
        BYTE    abDATA [DataCount&0x7FFF];
        }
```

As we can see, the structure consists of 12-header bytes and the data array.

- o *Xstart* and *Xend* start and end column position of the image. The position is divided by 2.
- o *Ystart* and *Yend* start and end row position of the image.
- o *DataCount* is the image size, 1..7FFFh bytes. The 0x8000 bit mask is set in this parameter.
- o *abData* is the image bitmap data array.

An example of such a record can be found in the *icon.asm* file.

All low-level functions are written in assembly and located in *oled.asm*. To incorporate the low-level functions into the project, three files must be added: *oled.asm*, *oled.inc* and *oled.h*.

Table 1. Low-Level Functions

Function	Description
oledStart(void)	Display initialization.
oledWriteMode(void)	Turn data direction from PSoC to OLED.
oledReadMode(void)	Turn data direction from OLED to PSoC.
oledSendByte(BYTE bX)	Write one byte to OLED (PSoC and OLED into write mode).
BYTE oledReadByte(void)	Read one byte from OLED (PSoC and OLED into read mode).
oledWriteCmd(BYTE bX)	Write one command byte to OLED.
oledWriteData(BYTE bX)	Write one data byte to OLED.
oledWriteROMBuf(const BYTE * Buf)	Write several bytes from the PSoC ROM (Flash) buffer to OLED. **Buf** points to an array of bytes located in ROM. The first two bytes in the buffer specify the length of data in bytes and the remaining are data bytes. If the 80h bit in the first byte is set, then data bytes will be written to the OLED's video memory (bitmap image), otherwise command bytes will be written.
oledWriteROMSprite(const BYTE * Buf)	Place a rectangle image located in the PSoC ROM (Flash) buffer onto the OLED. **Buf** points to the array of bytes located in ROM. These bytes hold information about the position of the image on the display, image size, and a bitmap of the image.

The high-level API functions include service routines for displaying text information, setting up a graphical data window, and clearing the selected area on display. These functions are realized in 'C' and located in the *main.c* file, and can easily be modified by the user. Table 2 lists these functions.

Table 2. High-Level Functions

Function	Description
`SetWindow(BYTE xs, BYTE ys, BYTE xe, BYTE ye)`	Specify the current window for display. The display of all graphical data will be confined to this window. The **xs**, **ys** parameters specify the lower-left corner coordinates, and the **xe**, **ye** parameters the upper-right corner of the window.
`oledCLSWindow(BYTE x, BYTE y, BYTE dx, BYTE dy, BYTE color)`	Fill the rectangular area with the selected color. The **x**, **y** parameters specify the lower-left corner coordinates of the window and the **dx**, **dy** parameters specify window size.
`oledString(BYTE xs, BYTE ys, BYTE color, char* text)`	Display a text string. The string is located in the PSoC RAM (SRAM) and terminated with a symbol of code 0x00. The starting coordinate (lower-left corner of the record), color, and address of the string are assigned as arguments. The 0x80 bit in the **xs** parameter defines the type/size of font. If the bit is set, the font is large. Otherwise, it is small.
`oledConstString(BYTE xs, BYTE ys, BYTE color, const char* text)`	Display a text string. The string is located in ROM (Flash) and terminated with a symbol of code 0x00. The starting coordinate (lower-left corner of the record), color, and address of the string are assigned as arguments. The 0x80 bit in the **xs** parameter defines the type/size of font. If the bit is set, the font is large. Otherwise, it is small.

Image Processing and Depiction

During the design of projects that use an OLED, it is often necessary to transfer graphical .bmp files from PC to PSoC. This Application Note includes a tool to convert bitmap images into .asm contents (*bmp.exe*). Follow these steps to incorporate a .bmp file into a PSoC project:

1. Copy image file into any graphic editor (Microsoft® Paint, for example).
2. Resize image to the appropriate size.
3. Save image as a 16-level grayscale bitmap file (*image.bmp*, for example).
4. Run *bmp.exe* program with the following command line:
   `bmp.exe filename.bmp filename.asm constname`. For example:
   `bmp.exe image.bmp logo.asm icon`.
5. Copy .asm file to PsoC project folder.
6. Open PSoC Designer and add file to project (Project >> Add to Project >> Files...).
7. Define the image constant into the program by including the following line:
   `extern const BYTE constname[];`.
   For example:
   `extern const BYTE icon[];`.

It is now possible to "draw" the image on the screen by calling the function `oledWriteROMSprite(constname)` in your project. For example: `oledWriteROMSprite(icon)`.

Figure 3 shows an image that was created on a PC, then converted to assembly format for the PSoC project and displayed on the OLED.

Figure 3. Example of Images on an OLED

Implementation of Fonts

The most important task for the display driver is to display text information. Two font types/sizes are implemented in the library. The range of their symbols is from 0x20 to 0x7F of the ASCII table. The size of the small font is 5x8 pixels and the large font is 8x16 (see Figure 4 (B)).

The bit images for each symbol of the small font are located in the *small_font.h* file. The bit images for the large font are located in the *big_font.h* file.

(A)

(B)

Figure 4. Large and Small Fonts
(A) "Negative" Mode
(B) "Positive" Mode

The font symbol is represented as an array of bytes that hold the monochrome image of each symbol (see Figure 5).
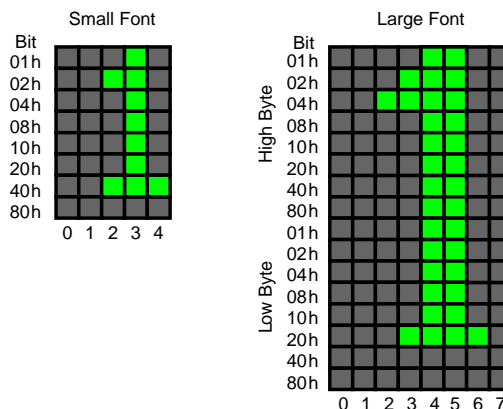

Figure 5. Format of Symbol Bitmap Image

As we can see, a symbol in the small font needs 5 bytes for implementation, and a symbol in the large font needs 14 bytes. Both the high-level functions for working with fonts were previously described. The program *Fontbuilder.exe* allows custom fonts to be built and is included with the project.

OLED Recommendations

Reducing pixel emission time and brightness are particularly useful ways to extend the lifetime of OLED displays. Users should also avoid using "Positive" (B) Screen Mode (Figure 4). For detailed information about optimizing the display operation, see reference [4] "*Effective Use of Pictiva™ OLED Displays: Power, Image and Lifetime Optimization.*" To include screen savers in OLED projects, refer to reference [5] "*OLED Display Module Screen Saver.*"

An OLED display requires a regulated power supply of +12V. There are many ways to generate this voltage supply. The demonstration project discussed in this Application Note uses a step-up converter to transform +3.3V to +12V. This converter only uses hardware resources of the PSoC: one switched capacitor block (analog comparator), two digital blocks (PWM and DigBuf), and one row LUT. The operation principle is the same as with the switch mode pump. The step-up converter is controlled by enabling\disabling the PWM output. The PWM duty cycle is set to $(1 - \frac{3.3V}{12V}) \times 100\% = 72.5\%$ to provide optimal converter operation.

Conclusion

The library we have described can be used for an OLED with different resolutions. It can also be adapted for an OLED with a serial interface by using the PSoC SPIM User Module. The combination of PSoC and OLED displays can be applied to low-price devices, cell phones, hand-held terminals, vehicle displays, self-contained instruments, and similar applications. This library has been successfully tested in the compass design from Application Note AN2348 "Tilt-Compensated Magnetic Compass with Built-In Temperature Sensor and OLED Display."

About the Author

Name: Valeriy Kyrynyuk
Title: Engineer
Background: Seven years experience on fieldbus and communications device design.
Contact: lopik@lviv.farlep.net

References

[1] Advance Information.128 x 80, 16 Gray Scale Dot Matrix OLED/PLED Segment/Common Driver with Controller.
http://www.solomon-systech.com.
To obtain the SSD0323 data sheet, go to:
http://www.solomon-systech.com/pop_datasheet.htm.

[2] Bitmap File Format and Manipulation.
http://catalog.osram-os.com/media/_en/Graphics/00030916_0.pdf.

[3] Software Reference Manual: Text and Graphics Generation.
http://catalog.osram-os.com/media/_en/Graphics/00030913_0.pdf.

[4] Effective Use of Pictiva™ OLED Displays: Power, Image and Lifetime Optimization.
http://catalog.osram-os.com/media/_en/Graphics/00027057_0.pdf.

[5] OLED Display Module Screen Saver.
http://catalog.osram-os.com/media/_en/Graphics/00030925_0.pdf.

Appendix A. Tools List

Tool	Description
bmp.exe (ARG1, ARG2, ARG3)	Converts an image from a .bmp file to an .asm file for "drawing" on an OLED display by using the function `oledWriteROMSprite` (ARG3). ARG1 is the source .bmp file. ARG2 is the destination .asm file. ARG3 is the name of resulting array constant.
Fontbuilder.exe	Builds font symbols. Converts the symbol image to a 'C'-format BYTE array (and from a 'C'-format BYTE array to a symbol image). The generated array can be directly inserted into a font definition file.