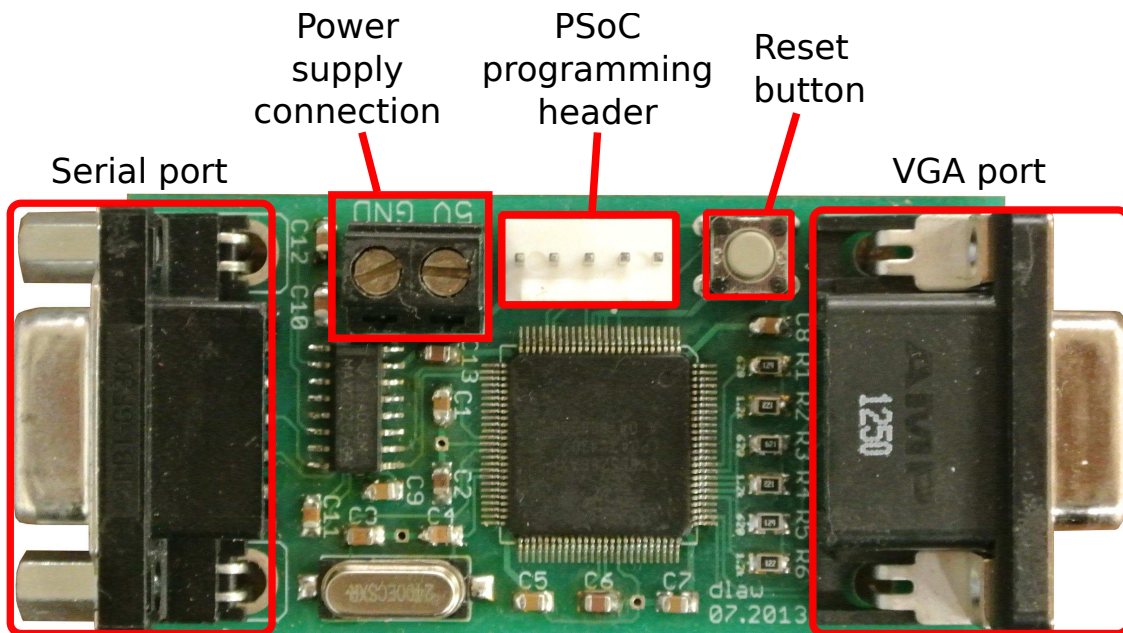


# How to use the PSoC-based VGA board

David Lawrence  
dlaw@mit.edu

November 2013



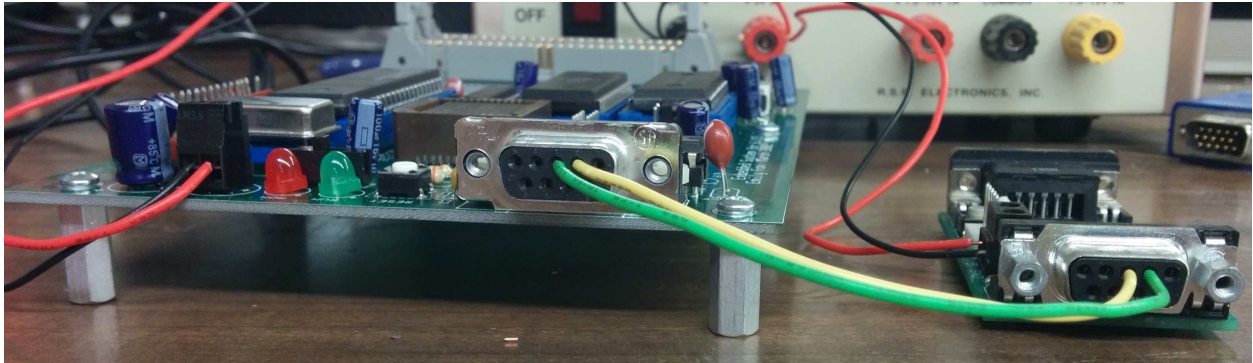
## 1 Hardware

The PSoC-based VGA board makes it easier to use a small microcontroller to drive a VGA monitor at low resolution.

To use it, first connect a 5 volt supply to the screw terminals. Be sure to observe the polarity markings, as the board does not have reverse voltage protection. The board begins displaying a black screen immediately, so an attached monitor should detect the output signal as soon as power is applied. The monitor's information screen should show that it is receiving a 1024 by 768 signal at 60 Hz.

At the other end of the board is a standard RS-232 connector for sending drawing commands. It can be attached to a computer using a regular serial cable. If it will be attached to another embedded device, a null modem cable should be used. The baud rate is fixed at 19200 baud.

The photo below shows the VGA board connected to an R31JP with the serial conductors in the null modem arrangement (pin 2 to pin 3 and pin 3 to pin 2). Note that the R31JP and VGA board share a common ground through the power supply.



## 2 Software

The serial communications protocol is byte-based. Any byte with the most significant bit set is a “command” byte, and any byte with the most significant bit clear is a “data” byte. A command byte is followed by some number of associated data bytes. There are four command bytes defined:

Command	Byte	# data bytes expected
Send 16 by 12 image	0x80	192
Send 32 by 24 image	0x81	768
Send 64 by 48 image	0x82	3072
Send 128 by 96 image	0x83	12288

The data bytes following each of these commands encode the image to be displayed. The color value for each pixel is encoded in one byte. The pixels are given in order, scanning across each row in order. Within each data byte, bits 0 and 1 encode the blue channel, bits 2 and 3 encode the green channel, and bits 4 and 5 encode the red channel. Example colors are given in the following table:

Color	Byte
White	0x3F
Light gray	0x2A
Dark gray	0x15
Black	0x00
Red	0x30
Green	0x0C
Blue	0x03

So, to display a pattern of vertical red and green stripes at 16 by 12 resolution, one would send 0x80 followed by 0x30, 0x0C, 0x30, 0x0C, ... repeated 192 times. To display a white

stripe at the top of a black screen at 128 by 96 resolution, one would send 0x83 followed by 0x3F 128 times (for the first row) and 0x00 12160 times (for the other 95 rows).

It is not necessary to send a command byte before every frame if the resolution will remain the same. For example, to draw two frames at 16 by 12 resolution, the byte 0x80 would be followed by 384 data bytes instead of 192. An arbitrary number of frames may follow each command byte. The PSoC displays a frame as soon as it has received the last byte of that frame.

### 3 Example

The following 8051 assembly snippet may be used to send an image once the serial port is configured for 19200 baud.

```
    mov a, #0x80
    acall sndchr           ; set resolution 16x12
    mov dptr, #data
    acall sndimg          ; send the image

;;; sndchr: send accumulator contents over the serial port.
sndchr: clr ti
        mov sbuf, a
        jnb ti, *         ; wait until the byte is sent
        ret

;;; sndimg: send the image currently pointed to by dptr.
sndimg: movx a, @dptr     ; load the next data byte
        cjne a, #0xff, *+4 ; jump over the ret unless at end of image
        ret
        acall sndchr      ; send the data byte
        inc dptr
        sjmp sndimg

;;; data: the image itself, terminated by 0xff.
data:  .db 0x30, 0x0c, 0x30, ..., 0x0c, 0xff
```

### 4 Extensibility

At 19200 baud, it takes about 0.1 seconds to send a 16 by 12 image and 10 seconds to send a 128 by 96 image. Obviously, this precludes displaying fast-moving objects at high resolution. Any user who would like to display higher bandwidth graphics is encouraged to modify the PSoC's source code and implement additional command bytes (e.g. "draw circle at coordinates") as needed for their specific application.