

VGA 驱动及实现




^O^S^H
PEN. SOURCE. ARDWARE
开源硬件社区

[HTTP://www.oshbbs.com](http://www.oshbbs.com)

基础教程

VGA 驱动及实现 基础教程

版本 V 0.0.0 日期 6/29/2009

 **开源硬件社区** www.oshbbs.com 将随时可能对本教程中的打印错误、与最新资料不符之处、程序和设备改进做出必要改动。这些改动不再事先通知，但将会编入新版手册中，并上传到我们的网站上。**本文部分内容来源于互联网，版权归原作者所有。**

版权所有 **开源硬件社区** www.oshbbs.com

作者：小马哥，可以通过 XiaomaGee@Gmail.com 与我联系！

Copyright 2009 - 2010

免责声明



开源硬件社区 ([Http://www.oshbbs.com](http://www.oshbbs.com)) 保留本教程的最终解释权，并不对由于因阅读本文所带来的一切后果（包括商业目的）负责，请大家慎重使用。

目 录

第一章 VGA 简介	3
一、 VGA 简介	5
二、 VGA 接口	5
三、 VGA 电气特性	6
第二章 时序实现	7
一、 时序介绍	9
二、 时序实现	10
第三章 OVGA 项目	13
一、 项目简介	15
二、 硬件介绍	15
三、 显存	16
四、 DAC	17
五、 调试端口	18

第一章 VGA 简介

VGA 简介

本章介绍了 VGA 接口的历史及电气、机械特性。

本章分为以下几个部分：

- 一、VGA 简介
- 二、VGA 接口形式
- 三、VGA 电气特性

一、VGA 简介

显卡所处理的信息最终都要输出到显示器上，显卡的输出接口就是电脑与显示器之间的桥梁，它负责向显示器输出相应的图像信号。CRT 显示器因为设计制造上的原因，只能接受模拟信号输入，这就需要显卡能输入模拟信号。VGA 接口就是显卡上输出模拟信号的接口，VGA (Video Graphics Array) 接口，也叫 D-Sub 接口。虽然液晶显示器可以直接接收数字信号，但为了兼容性，大多数液晶显示器也配备了 VGA 接口。

VGA 是 IBM 在 1987 年随 PS / 2 机一起推出的一种视频传输标准，具有分辨率高、显示速率快、颜色丰富等优点，在彩色显示器领域得到了广泛的应用。目前 VGA 技术的应用还主要基于 VGA 显示卡的计算机、笔记本等设备。根据分辨率不同，VGA 分为 VGA (640x480)、SVGA (800x600)、XGA (1024x768)、SXGA (1280x1024) 等。

虽然说 VGA 的标准对于现在的个人计算机市场十分过时，但是 VGA 仍然是所有制造商所支持的最低标准，例如不管所有厂商的显卡，在不安装自己驱动的情况下，都是支持 VGA 标准显示的。

二、VGA 接口

VGA 接口是一种 D 型接口 (D-SUB)，上面共有 15 针空，分成三排，每排五个。如图 1-1 所示。而与之配套的底座则为孔型接口。



图 1-1 VGA 接口及底座

三、VGA 电气特性

VGA 引脚定义如表 1 所示。

表 1：

引脚	名称	描述	引脚	名称	描述
1	RED	RED Video	9	KEY	Key (No pin)
2	GREEN	Green Video	10	SGND	Sync Ground
3	BLUE	Blue Video	11	ID0	Monitor ID bit 0
4	ID2	Monitor ID bit 2	12	ID1	Monitor ID bit 1
5	GND	Ground	13	HSYNC	Horizontal Sync
6	RGND	Red Ground	14	VSYNC	Vertical Sync
7	GGND	Green Ground	15	ID3	Monitor ID bit 3
8	BGND	Blue Ground			

引脚 1、2、3 分别为红绿蓝三基色模拟电压，为 0~0.714V peak-peak (峰-峰值)，0V 代表无色，0.714V 代表满色。一些非标准显示器使用的是 1V_{pp} 的满色电平。

三基色源端及终端匹配电阻均为 75 欧姆，如图 1-2 所示。

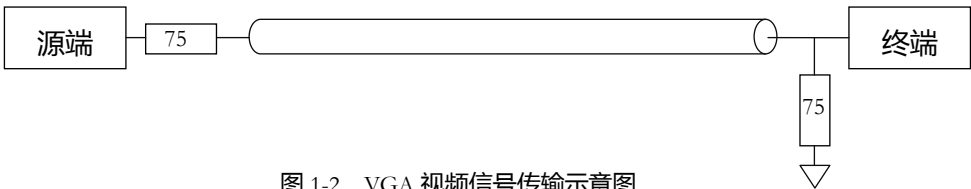


图 1-2 VGA 视频信号传输示意图

HSYNC 和 VSYNC 分别为行数据同步与帧数据同步，为 TTL 电平。

第二章 时序实现

时序实现

本章介绍 *VGA* 时序及实现方式。

本章分为以下几个部分：

- 一、时序介绍
- 二、时序实现

一、 时序介绍

VGA 的时序如图 2-1、图 2-2 所示。它分为行数据时序和帧数据时序。

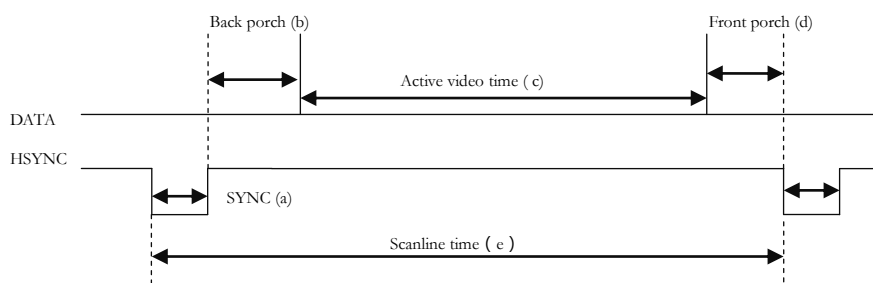


图 2-1 VGA 行数据时序

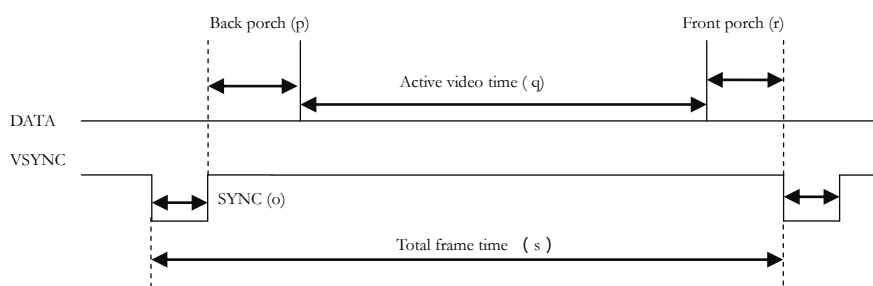


图 2-2 VGA 帧数据时序

行数据时序，顾名思义，也就是显示一行数据的时序。从图 2-1 可以看出，显示一行数据需要处理两件事情。第一：产生行同步 HSYNC。不难看出，HSYNC 是一个脉冲信号，此信号的周期为： $e=a+b+c+d$ ，低电平时间为 a 。其中 a 、 b 、 c 、 d 均为时间信号，这些信号根据需要显示的分辨率不同而不同。第二：产生显示的数据 (DATA) 信号，此信号为模拟信号，当在显示有效数据 (Active video) 内，DATA 信号为 $0\sim 0.714V_{pp}$ 的模拟电压 (R、G、B)，根据分辨率的不同，DATA 的采样率、点数也皆不相同。

帧数据时序与行时序类似，也就是显示一屏数据的时序。只是这里的基本单位为每行数据，而行数据里面的最基本单位为每个点。

上面提过，不同的分辨率，时序上的时间是不一样的。表 2、表 3 列出常用分辨率及时间参数。

表 2 常见刷新率时序表：

显示模式	时钟 (MHz)	行时序 (像素数)					帧时序 (行数)				
		a	b	c	d	e	o	p	q	r	s
640x480@60	25.175	96	48	640	16	800	2	33	480	10	525
640x480@75	31.5	64	120	640	16	840	3	16	480	1	500
800x600@60	40.0	128	88	800	40	1056	4	23	600	1	628
800x600@75	49.5	80	160	800	16	1056	3	21	600	1	625
1024x768@60	65	136	160	1024	24	1344	6	29	768	3	806
1024x768@75	78.8	176	176	1024	16	1312	3	28	768	1	800
1280x1024@60	108.0	112	248	1280	48	1688	3	38	1024	1	1066
1280x800@60	83.46	136	200	1280	64	1680	3	24	800	1	828
1440x900@60	106.47	152	232	1440	80	1904	3	28	900	1	932

表 3 常见刷新率时序表 (时间)：

显示模式	时钟 (MHz)	行时序时间 (μ s)					帧时序时间 (ms)				
		a	b	c	d	e	o	p	q	r	s
640x480@60	25.175	3.81	1.9	25.4	0.635	31.7	.006	1.048	15.25	0.317	16.6
640x480@75	31.5	2.03	3.8	20.3	0.507	26.7	0.08	0.426	12.8	0.026	13.3
800x600@60	40.0	3.2	2.2	20	1	26.4	0.1	0.6	15.84	0.026	16.6
800x600@75	49.5	1.62	3.2	16.16	0.323	21.3	0.06	0.45	12.8	0.021	13.3
1024x768@60	65	2.09	2.46	15.7	0.37	20.6	0.12	0.599	15.87	0.062	16.6
1024x768@75	65	1.22	2.23	12.99	0.203	16.6	0.05	0.466	12.78	0.016	13.3
1280x1024@60	108	1.04	2.3	11.85	0.444	15.6	0.05	0.6	16.0	0.015	16.6
1280x800@60	83.46	1.63	2.4	15.3	0.76	20.1	0.06	0.48	16.1	0.02	16.6
1440x900@60	106.47	1.43	2.18	13.52	0.75	17.9	0.05	0.5	16.1	0.017	16.6

二、 时序实现

VGA 时序的实现有很多方式，可以用专用芯片，可以用快速的 CPU，也可用可编程器件来实现。这里采用廉价的可编程器件，来产生 VGA 所需的时序。

由于目前液晶显示器的普及，而高于 60Hz 的刷新率对于液晶来说，没有任何意义，所以我们以 800x600 在 60Hz 的刷新率下为例，解说 VGA 时序的产生。从表 2 可以看出，800x600@60Hz，需要 40.0MHz 的驱动时钟，经过计算可知，HSYNC 信号频率为 37.8787kHz，低电平脉冲时间为 3.2 μ s。详细 verilog 代码如下：

```

module VGA(dclk,db,r,g,b,hs,vs,addr);
input dclk; //40MHz 800x600
input db;
output addr;
output r,g,b,hs,vs;

reg hs,vs;
reg[10:0] count_v,count_h;
reg[18:0]addr;
wire[15:0]db;
reg flag;
wire[4:0] r;
wire[5:0] g;
wire[4:0] b;

assign {r,g,b}=(flag==1?db[15:0]:0);
//Hsync clock generator
always@(posedge dclk)begin
    if (count_h == 1056)
        count_h <= 0;
    else
        count_h <= count_h+1;
end
//Vsync clock generator
always@(posedge dclk)begin
    if (count_v == 628) count_v <= 0;
    else if (count_h == 1056) count_v <= count_v+1;
end
//Hsync and Vsync generator.
always@(posedge dclk)begin
    if (count_h == 0) hs <= 0;
    if (count_v == 4) vs <= 1;
    if (count_h == 128) hs <= 1;
    if (count_v == 0)vs <= 0;
    if (count_v > 27 && count_v < 627)begin
        if ((count_h > 216) && (count_h < 1017)) begin
            flag <= 1;
            addr <= addr+1;
        end else flag <=0;
    end else addr<=0;
end

endmodule

```



第三章 OVGA 项目

OVGA 项目

通过本章，可以了解、学习 OVGA 开源项目。

本章分为以下几个部分：

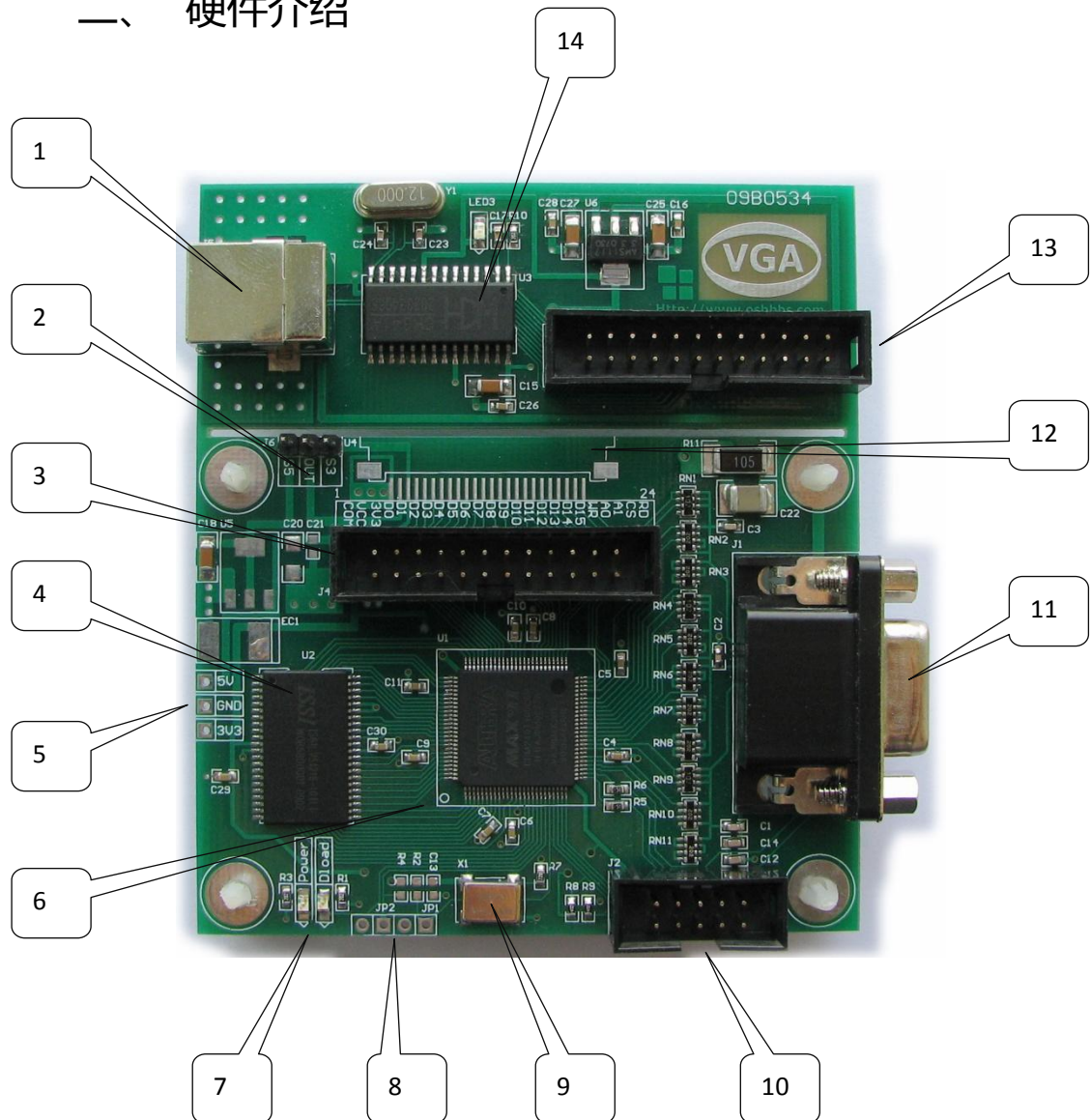
- 一、项目简介
- 二、硬件介绍
- 三、显存
- 四、DAC
- 五、调试端口

一、项目简介

OVGA 是【开源硬件社区】推出的开源 VGA 驱动项目。项目包括硬件实现、软件驱动等，所有资源开源。

当前显示器越来越廉价，如果用显示器作为嵌入式设备的显示终端，不管从显示效果和成本上，都比常用的单色液晶、低分辨率彩色液晶等都有优势。本项目利用廉价的可编程逻辑器件，实现了 800x600 分辨率，60Hz 刷新率下的真彩色（16bit 色）显示。通过外部预留的总线接口，可以与单片机、ARM、DSP 等嵌入式处理器相连，作为人机界面。

二、硬件介绍



OVGA 硬件分为两部分 :OVGA 模块和测试平台。两部分独立工作 ,虽然 PCB 连在一起,但是可以掰开独立工作。两部分是通过 24p 连接线连接在一起的。OVGA 模块供电可以选择 3.3V 或者 5V 供电,通过板子上的跳线选择。

为了便于与其他系统连接,OVGA 模块包含了两套接口,24 脚底座和 FPC 连接器。下面介绍下系统资源:

- 1. USB 调试器接口;
- 2. 3.3V、5V 供电选择;
- 3. OVGA 模块上,24p 连接器;
- 4. 显存芯片 61LV25616;
- 5. 外接电源(可接独立外接电源);
- 6. 主逻辑芯片 EPM240T100C5;
- 7. 指示 LED,分别为电源指示、下载指示;
- 8. 两个设定跳线,备用;
- 9. 系统时钟,80MHz 有源晶振;
- 10. CPLD 调试接口;
- 11. VGA 接口,工作时接到显示器上;
- 12. FPC 24 脚接口;功能等同 3;
- 13. 测试平台 24p 接口,用计算机下载图片时,连接于 3;
- 14. USB 转并口芯片 CH341A。

三、 显存

800x600 分辨率 16bit 下,一屏幕的数据有 960k 字节。我们选用 61LV51216 作为显存,存储一屏幕的数据。由于 61LV51216 时钟最快可以到 125MHz,而我们实现的显示方式只需要 40MHz 的驱动时钟,所以我们可以用 80MHz 的时钟去

驱动 61LV51216，这样可以用时钟的偶数周期对 RAM 进行写，而用奇数周期对 RAM 进行读。进而避免了刷屏幕时（读 RAM）写屏幕数据的逻辑冲突问题。

四、DAC

CPLD 只能输出数字信号，而 VGA 需要的 R、G、B 是模拟信号，所以我们需要进行 模拟-数字 转换即 DAC 功能。实现视频 DAC 我们可以选用专用的芯片，但是那样价格昂贵，这里我们选用 R-2R 电阻网络作为视频 DAC，从最终测试结果上看，此方法的显示效果是理想的。没有雪花、抖动、颜色也正。当然如果在要求较高的场合，是必须选用专用 DAC 芯片的。

16bit 真彩色显示，RGB 分别占的位数为 5:6:5 模式，也就是红色用 5 位、绿色用 6 位、蓝色用 5 位来表示。下面就用红色 5 位来说明 R-2R 的选取。

由于 DAC 是一个线性的模型，所以当红色 5bit 输出都是高电平的时候，我们需要得到 0.714V 的电压，拓扑结构如图 3-1 所示。

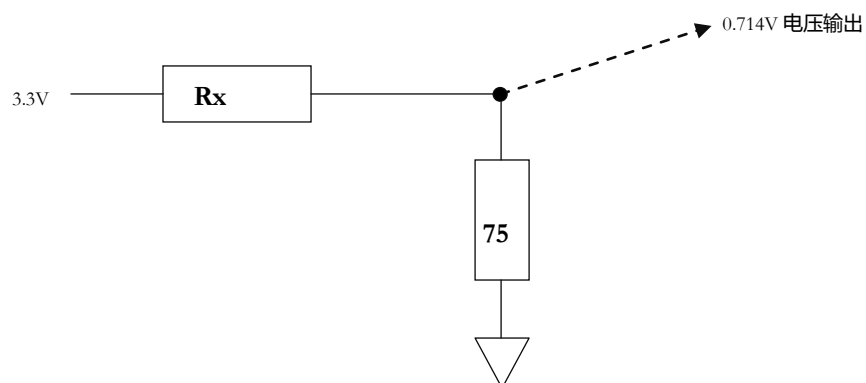


图 3-1 R-2R 视频 DAC 模式拓扑结构

CPLD 的输出电压为 3.3V，Rx 为 5bit R-2R 电阻网络并联而得，所以可以得到如下方程：

$$\frac{R_x + 75}{3.3} = \frac{75}{0.714} \quad (1)$$

设基准电阻为 Ra，则：

$$R_a // 2R_a // 4R_a // 8R_a // 16R_a = R_x \quad (2)$$

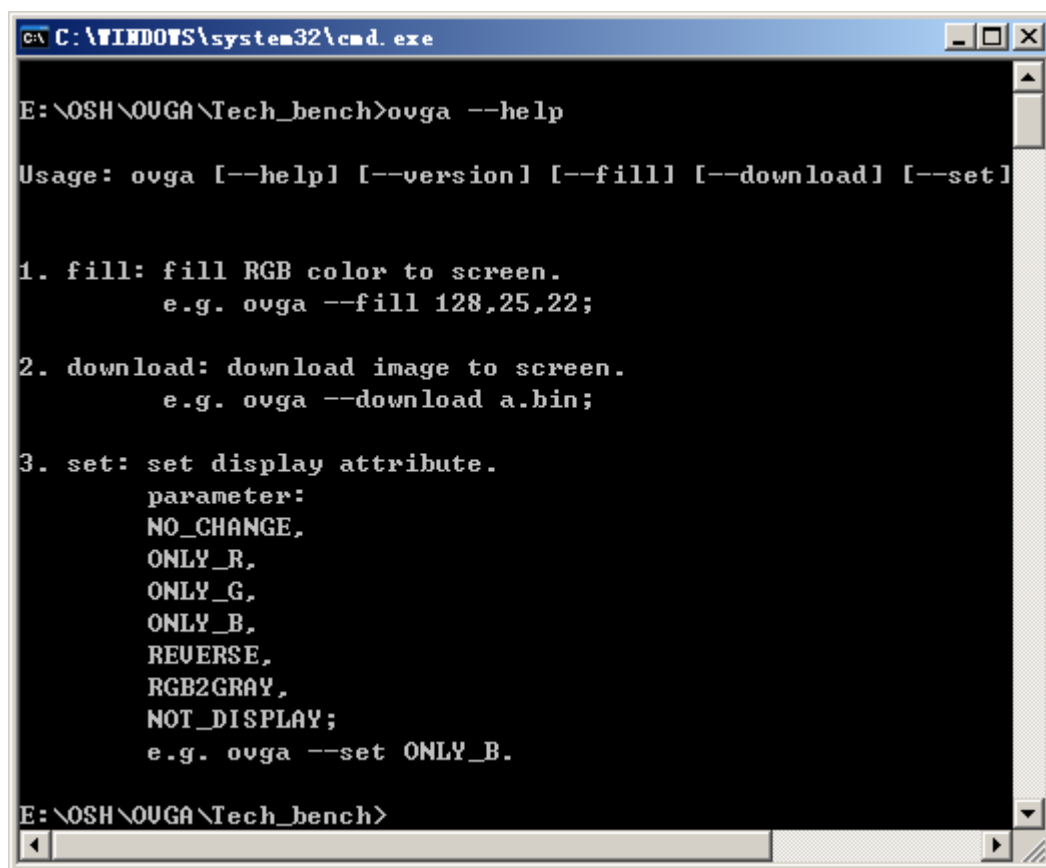
根据式 (1)、(2)，求解得到，Rx=271.6，Ra=526.2。所以我们选择 500，1k，2k，4k，8k 作为电阻网络，为了保证电阻的一致性，我们选用了 1k、2k 的电阻并联、串联的形式。

五、 调试端口

为了便于调试、测试，我们选用了 USB 转并口芯片 CH341A，对模块进行数据下载。CH341A 为 易用的国产新片，其使用简单，价格低廉，我们可以通过简单的 API 调用，来实现数据的下载。

CH341A 支持的 API 为 open , close , write0 , wirtel , read0 , get_driver_ver、get_device_name 等。

OVGA PC 测试平台软件为 OVGA.exe，可以通过这个软件下载图片到显存、填充颜色到显存及做些简单的图像变换。其主要功能如下：



```
C:\WINDOWS\system32\cmd.exe

E:\OSH\OVGA\Tech_bench>ovga --help

Usage: ovga [--help] [--version] [--fill] [--download] [--set]

1. fill: fill RGB color to screen.
    e.g. ovga --fill 128,25,22;

2. download: download image to screen.
    e.g. ovga --download a.bin;

3. set: set display attribute.
    parameter:
    NO_CHANGE,
    ONLY_R,
    ONLY_G,
    ONLY_B,
    REVERSE,
    RGB2GRAY,
    NOT_DISPLAY;
    e.g. ovga --set ONLY_B.

E:\OSH\OVGA\Tech_bench>
```