

FX-VXCC256 VGA 控制器使用手册

Ver1.0

一、简介.....	2
二、控制板接口与设置.....	2
1、控制板外形尺寸:	2
2、与 MPU 的接口 (J2) 引脚定义.....	3
3、VGA 接口 (J4) 引脚定义.....	5
三、指令操作说明.....	6
四、FX-VXC256 应用.....	6
1、直接访问方式.....	7
2、间接控制方式.....	8
3、256 色应用程序模块.....	10
子程序一: 用背景色清显示子程序 (直接方式)	10
子程序二: 坐标转换为 RAM 绝对地址 (51 汇编)	11
子程序三: 用指定色画任意长度水平/垂直直线 (直接方式)	12
子程序四、写 16×16 点阵字符子程序 (直接方式)	13



深圳市方显科技有限公司
 ShenZhen Fangxian Technology CO., LTD
 网站(Web): <http://www.fancyview.com>

一、简介

FX -VXC256 系列 VGA 控制器为方显科技自主设计开发,与单片机计算及机接口和操作指令简单;可方便地对显示存储器进行实时的读写。

FX-VXC256 VGA 控制器包括4种类型:FX-V3224C256, FX-V6448C256, FX-V8060C256, FX-V1024C256。其中FX-V3224C256 接CGA 显示器, 其他接LCD 显示器或CRT 显示器。

它们支持的分辨率为:

FX-V3224C256 : 320×240

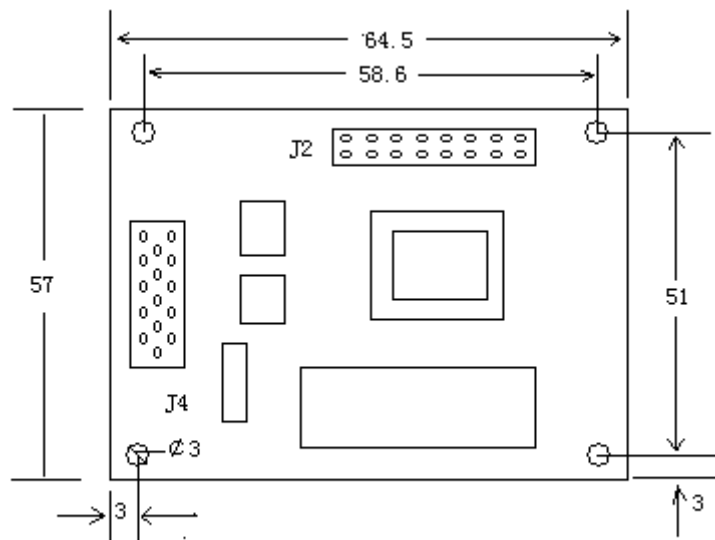
FX-V6448C256 : 640×480

FX-V8060C256 : 800×600

FX-V1024C256 : 1024×768

二、控制板接口与设置

1、控制板外形尺寸:



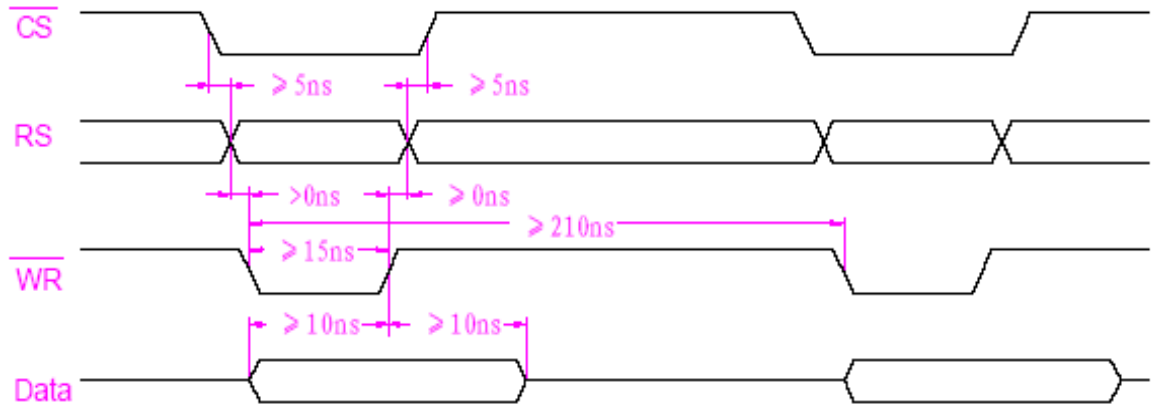
2、与MPU的接口 (J2) 引脚定义

引脚	定义	状态	功能
1	GND	0V	电源地
2	GND	0V	电源地
3	VCC	5V	电源
4	/RD	输入	读, 低电平有效
5	/WR	输入	写, 低电平有效
6	/CS	输入	片选, 低电平有效
7	RS	输入	端口选择 1-数据, 0-指令
8	D0	三态	数据总线(底位)
9	D1	三态	数据总线
10	D2	三态	数据总线
11	D3	三态	数据总线
12	D4	三态	数据总线
13	D5	三态	数据总线
14	D6	三态	数据总线
15	D7	三态	数据总线(高位)
16	NC	空	空

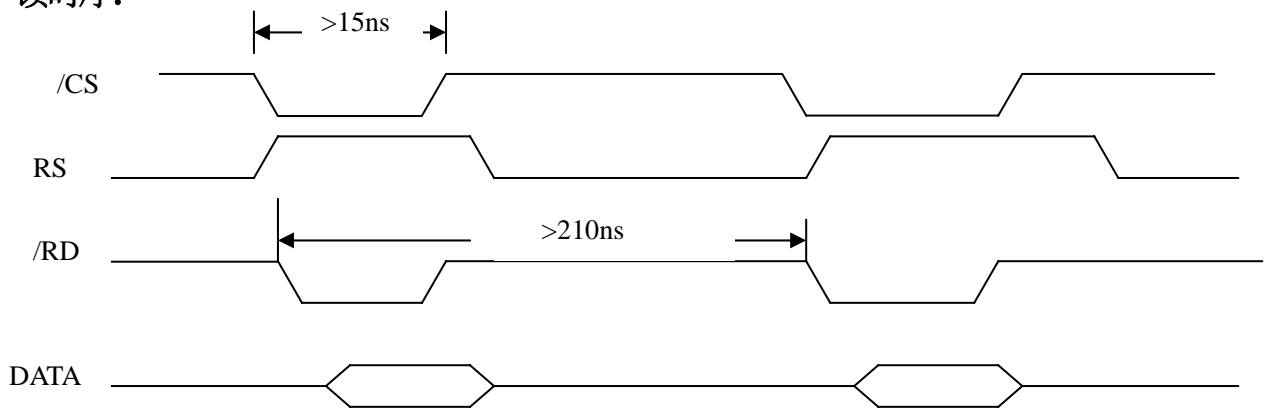
显示256色时，每字节显示一个像素，数据格式约定如下：DB7 (R2, 红色高位)，DB6 (R1)，DB5 (R0, 红色低位)，DB4 (G2, 绿色高位)，DB3 (G1)，DB2 (G0, 绿色高位)，DB1 (B1, 蓝色高位)，DB1 (B0, 蓝色低位)。

显示16色时，每字节显示两个像素，数据格式约定如下：DB7 (R)，DB6 (G1)，DB5 (B)，DB4 (G0)，DB3 (R)，DB2 (G1)，DB1 (B)，DB1 (G0)。

写时序:



读时序:



3、VGA 接口 (J4) 引脚定义

PIN NO.	Symbol	I/O	Function
1	R	0	Analog red signal output
2	G	0	Analog green signal output
3	B	0	Analog blue signal output
4	NC		No connect
5	NC		No connect
6	GND		Red ground
7	GND		Green ground
8	GND		Blue ground
9	NC		No connect
10	DGND		Digit ground
11	NC		No connect
12	NC		No connect
13	HS	0	Horizontal Sync signal
14	VS	0	Vertical Sync signal
15	NC		No connect

三、指令操作说明

FX 系列控制板使用统一的指令码格式，只需要通过控制器的指令对显示存储器进行控制即可，指令码格式为：

D7	D6	D5	D4	D3	D2	D1	D0
x	x	x	A18	A17	A16	CMD1	CMD0

其中：

D7, D6, D5: 保留

D6~D3: 显存高位地址(A18, A17, A16)

CMD1、CMD0 组合选择内部寄存器，组合功能如下：

CMD1	CMD0	功能说明
0	0	显存低 8 位地址
0	1	显存高 8 位地址
1	0	数据寄存器
1	1	控制寄存器

控制寄存器说明：

D7	D6	D5	D4	D3	D2	D1	D0
关闭显示	X	X	当前写入页	X	X	VIDEO	当前显示页

代码说明：

```

WCMD = 3;      // 选中控制寄存器
WDAT = 0x00;   // 当前显示的是第 0 页， 写入的也是第 0 页

WCMD = 3;
WDAT = 0x01;   // 当前显示的是第 1 页， 写入的是第 0 页

WCMD = 3;
WDAT = 0x10;   // 当前显示的是第 0 页， 写入的是第 1 页

WCMD = 3;
WDAT = 0x11;   // 当前显示和写入的都是第 1 页

```

D1: VIDEO : 视频切换开关

D7: 0: 使能显示; 1: 关闭显示

四、FX-VXC256 应用

所有应用子程序作如下约定，其它占用资源，请参考各子程序说明，
直接访问方式地址定义，用户根据自己的硬件接口修改

51 汇编示例：

WCMD EQU 7E00H ; 写指令代码的地址

WDAT EQU 7F00H ; 写参数及读写显示数据的地址

C51 示例：

```
#define WCMD 0x7E00
```

```
#define WDAT 0x7F00
```

间接控制方式信号定义，用户根据自己的硬件接口修改

51 汇编示例：

RS EQU P3.0 ; 数据与指令选择寄存器

RD EQU P3.1 ; /RD

FX-TA256

WR EQU P3.2 ; /WR
 CS EQU P3.3 ; /CS
 DATBUS EQU P1 ; 间接控制方式数据总线。

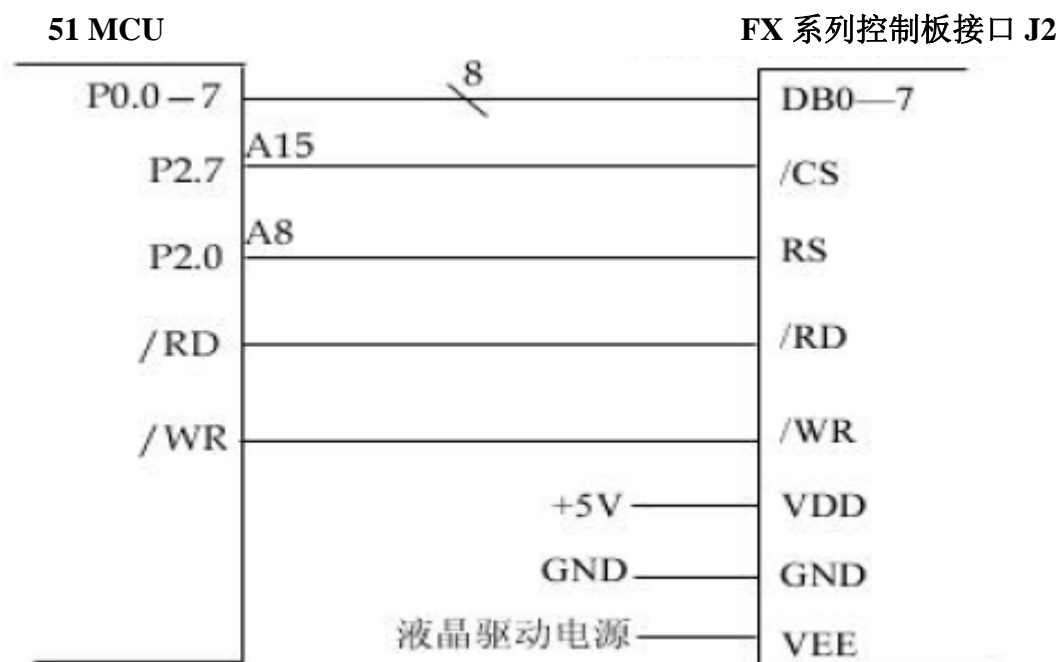
程序变量定义，用户根据自己的内部资源分配修改

ADDR1 EQU 78H ; 显示 RAM 绝对地址低 8 位
 ADDR2 EQU 77H ; 显示 RAM 绝对地址高 8 位
 A16 EQU 76H ; 显示 RAM 绝对地址第 17 位
 O_XL EQU 75H ; 横坐标低 8 位
 O_XH EQU 74H ; 横坐标高 8 位
 O_YL EQU 73H ; 纵坐标低 8 位
 COUNT1 EQU 71H ; 计数器
 COUNT2 EQU 70H
 COUNT3 EQU 69H CCODE EQU 68H ; 字符代码
 HxH EQU 01H ; 显示域宽度高 8 位
 HxL EQU 40H ; 显示域宽度低 8 位
 COLORF EQU 67H ; 前景色
 COLORB EQU 69H ; 背景色

1、直接访问方式

MPU 通过数据总线和控制信号直接采用 I/O 设备访问形式控制 FX 系列控制板。

如下图所示：



写指令子程序:

WR_CMD:

```
PUSH    DPL
PUSH    DPH
MOV     DPTR, #WCMD
MOV     A, CMD
MOVBX   @DPTR, A
POP     DPH
POP     DPL
RET
```

写数据子程序

WR_DAT:

```
PUSH    DPL
PUSH    DPH
MOV     DPTR, #WDAT
MOV     A, DAT
MOVBX   @DPTR, A
POP     DPH
POP     DPL
RET
```

读数据子程序

RD_DAT:

```
PUSH    DPL
PUSH    DPH
MOV     DPTR, #WDAT
MOVBX   A, @DPTR
MOV     DAT1, A
POP     DPH
POP     DPL
RET
```

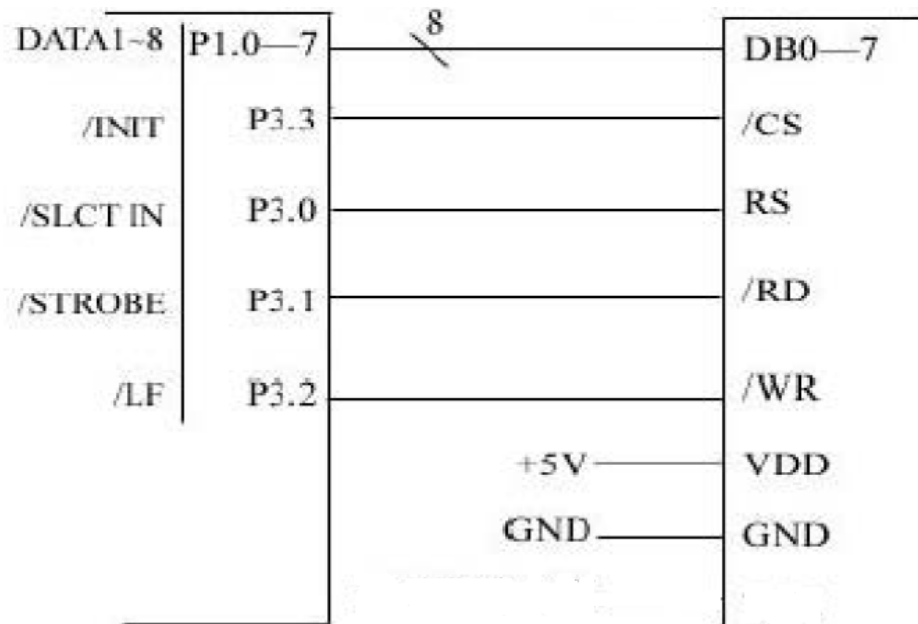
2、 间接控制方式

间接控制方式是MCU通过对I/O/并口的位操作间接实现对FX系列液晶显示控制板的访问控制,

如下图所示：

51 MCU/PC 并口

FX 系列控制板接口 J2



写指令子程序：

WR_CMD:

```

MOV    DATBUS, CMD
CLR    RS           ; RS=0
CLR    CS           ; /CS=0
CLR    WR           ; /WR=0
SETB   WR           ; /WR=1
SETB   CS           ; /CS=1
RET
    
```

写数据子程序

WR_DAT:

```

MOV    DATBUS, DAT
SETB   RS           ; RS=1
CLR    CS           ; /CS=0
CLR    WR           ; /WR=0
SETB   WR           ; /WR=1
SETB   CS           ; /CS=1
RET
    
```

读数据子程序

WR_DAT:

```

MOV    DATBUS, #0FFH
SETB   RS           ; RS=1
CLR    CS           ; /CS=0
CLR    RD           ; /WR=0
    
```

```

MOV     DAT, DATBUS
SETB   RD           ; /WR=1
SETB   CS           ; /CS=1
RET

```

3、256 色应用程序模块

子程序一：用背景色清显示子程序（直接方式）

51 汇编：；

占用资源：COUNT1, COUNT2, COUNT3, A；

用背景色填充 640×480 区域

CLEAR:

```

MOV     DPTR,#WCMD
MOV     A,#00H
MOVX   @DPTR,A
MOV     DPTR,#WDAT
MOV     A,#00H
MOVX   @DPTR,A           ; 写低 8 位地址
MOV     DPTR,#WCMD
MOV     A,#01H
MOVX   @DPTR,A
MOV     DPTR,#WDAT
MOV     A,#00H
MOVX   @DPTR,A           ; 写高 8 位地址
MOV     DPTR,#WCMD
MOV     A,#02H
MOVX   @DPTR,A           ; 准备写显示数据
MOV     A,#COLORB
MOV     DPTR,#WDAT
MOV     COUNT2,#480
CLR2:  MOV     COUNT3,#640
CLR3:  MOVX   @DPTR,A
        DJNZ   COUNT3,CLR3
        DJNZ   COUNT2,CLR2
RET

```

C51 例程（直接方式）：

```

#define WCMD 0x7e00
#define WDAT 0x7f00
#define SCREENWIDTH 640
typedef unsigned char    uchar;
typedef unsigned int     uint;
uchar data charcolor;   // 字符前景色
uchar data charbkcolor; // 字符背景色
uchar data gracolor;    // 图形前景色
uchar data grabkcolor;  // 图形背景色
uchar data char_h_dot;  // 字符水平点数

```

FX-TA256

```

uchar data char_v_dot; // 字符垂直点数
uint data curx;        // 当前光标横坐标
uint data cury;        // 当前光标纵坐标

```

```

void clear(void) //colorb: 背景色
{
    uint data i;
    uchar data j;

    WCMD=0x00; WDAT=0x00;
    WCMD=0x01; WDAT=0x00;
    WCMD=0x02;
    for(i=0;i<480;i++){
        for(j=0;j<100;j++){
            WDAT=charbkcolor;
            WDAT=charbkcolor;
            WDAT=charbkcolor;
            WDAT=charbkcolor;
            WDAT=charbkcolor;
            WDAT=charbkcolor;
            WDAT=charbkcolor;
            WDAT=charbkcolor;
        }
    }
}

```

子程序二、坐标转换为 RAM 绝对地址 (51 汇编)

```

CO_ADDR:    ;坐标转换为 RAM 绝对地址
MOV    A,O_YL
MOV    B,#HxL
MUL    AB
ADD    A,O_XL
MOV    ADDRl,A          ; 低 8 位地址存 ADDRl
MOV    A,B
ADDC   A,O_XH
MOV    ADDRH,A          ;
高 8 位地址存 ADDRH
CLR    A
ADDC   A,#00H
MOV    A16,A
MOV    A,O_YL
MOV    B,#HxH
MUL    AB
ADD    A,ADDRH
MOV    ADDRH,A
CLR    A
ADDC   A,A16
RL     A

```

```

RL    A
MOV   A16,A           ;将 A16 移位到与指令代码相应的位
RET

```

子程序三 任意位置画点程序

```

#define SCREEN_WIDE 640

void putpixel(int x,int y, uchar color)
{
    uchar a16;
    ulong p;

    p = (ulong)y*SCREEN_WIDE+(ulong)x;
    a16 = (uchar)(p>>16);
    a16 <<= 2;
    WCMD = 0x00|a16;
    WDAT = (uchar)p;
    WCMD = 0x01|a16;
    WDAT = (uchar)(p>>8);
    WCMD = 0x02|a16;
    WDAT = color;
}

```

子程序四 显示汉字的 c 语言例程:

```

//" 入口参数: 列 ,行 ,点阵(方阵), 字模数据首地址,前景色,背景色
curx-----当前光标的 x 坐标(全局变量)
cury-----当前光标的 y 坐标(全局变量)
SCREENWIDTH-----屏幕宽度(320, 480, 640,800 等)
WCMD-----命令端口
WDAT-----数据端口

```

```

typedef unsigned char    uchar;
typedef unsigned int     uint;
typedef unsigned long    ulong;

```

char_v_dot 和 char_h_dot 说明汉字点阵类型(16*16 或 32*32)

```

void dispch_code(char code *s)           // 字符点阵存放在 ROM 中
{
    uchar i, j, v, ha, k;
    long addr;

    addr = (long)cury*SCREEN_WIDTH+(long)curx; // 坐标转换为显存地址
    for(i=0;i<char_v_dot;i++){              // 字符高度
        ha = addr>>16;                       // 高位地址
        ha <= 2;
    }
}

```

```
WCMD = ha|0; WDAT = addr;           // 设置显存低 8 位地址
WCMD = ha|1; WDAT = addr>>8;       // 设置显存低高位地址
WCMD = ha|2;                         // 数据通道
for(j=0;j<char_h_dot;j++){         // 字符宽度
    v = *s++;
    for(k=0;k<8;k++){
        if(v&0x80){                 // 显示前景色还是背景色
            WDAT = charcolor;
        }
        else{
            WDAT = charbkcolor;
        }
        v<<=1;
    }
}
addr+=SCREEN_WIDTH;                 // 下一行
}
```