

TCB8000A

控制板应用参考

深圳市拓普微科技发展有限公司

版本	描述	日期	编者
0.1	新版本	2006-08-04	淮俊霞



目 录

1 简介	3
2 应用	3
2.1 接口	3
2.2 命令包格式	4
2.3 操作码及参数表	4
2.4 LCD 屏幕显示数据形式	5
2.5 应用举例	5
2.6 流程图	6
参考程序	7

1 简介

本公司生产的 TCB8000A 控制板采用 T8000 作为控制器, 设置了 8 位的 MCU I/O 接口(缺省设置)或 16 位的总线接口; 16-bpp RGB 的数据格式能够实现 6.5 万种颜色的显示。控制器内置 ASCII 码, 另有 256K ROM 中文字库版本可选。TCB8000A 具有图形加速器的功能, 简单命令便能实现点、线段、矩形和圆的 2D 作图, 方便用户制作菜单、列表、连线图等, 可加快 TFT 产品的开发。

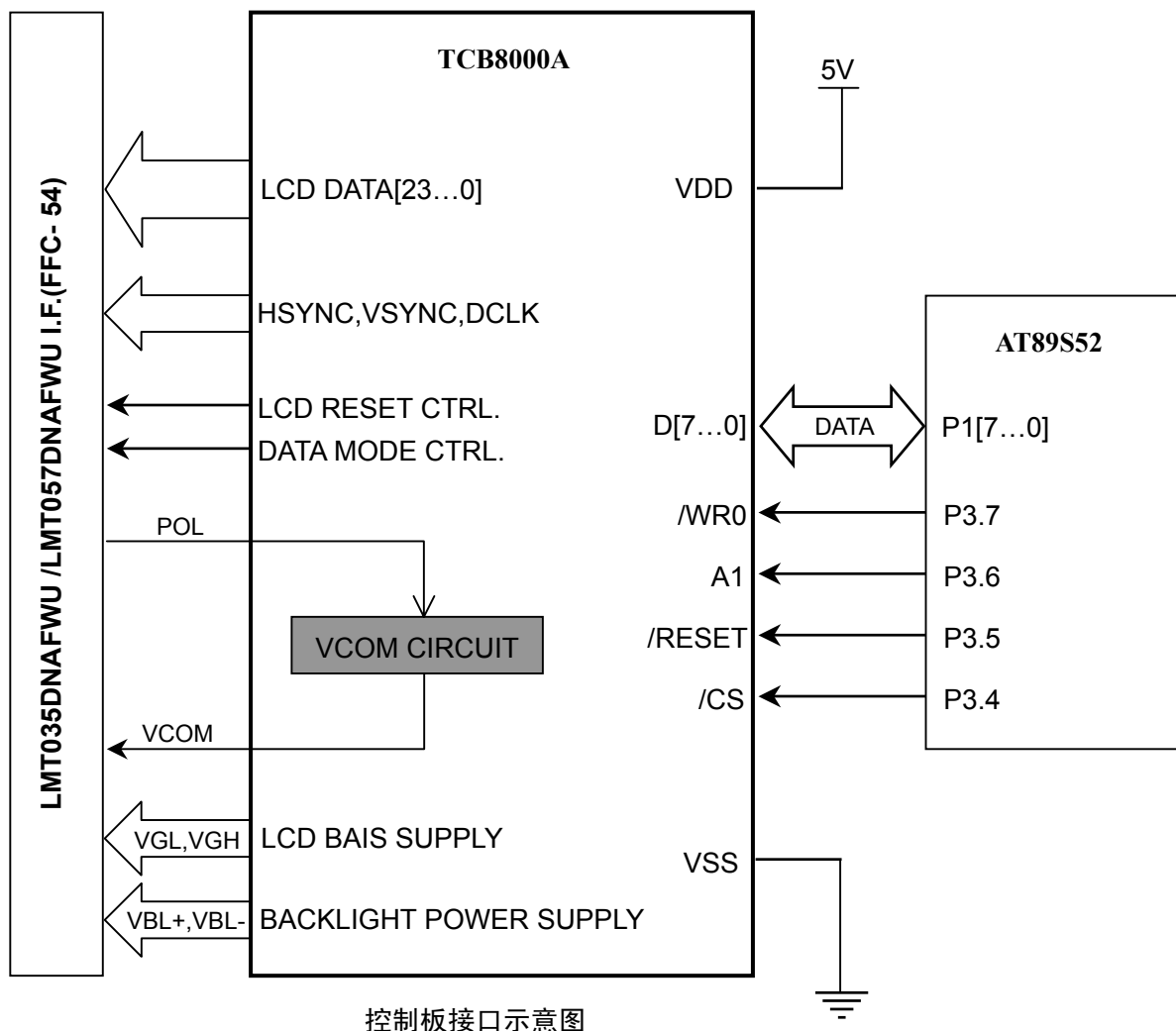
控制板特点:

- ✧ 单 5V 电源供电
- ✧ 内置背光驱动电路
- ✧ 嵌入 64K×16 SRAM
- ✧ 内置 LCD DC-DC 升压电路
- ✧ 3.3V MCU 接口 / 逻辑 I/O (可接受 5V 逻辑输入)
- ✧ 可直接连接 LMT035DNAFWU 或 LMT057DNAFWU TFT 液晶显示模块

2 应用

2.1 接口

以 8 位并行通信为例, TCB8000A 与 MCU 及 TFT 液晶显示模块接口如下图所示:



控制板接口示意图

2.2 命令包格式

控制器的所有命令应以如下格式发送:

操作码 (1 byte)	参数/数据 (最多 64 bytes)
--------------	---------------------

并行模式下,MPU 首先把完整的命令包送入地址为 F004H 的只写寄存器(A1=0),然后送”1”到地址为 F006H 的寄存器 (A1=1), 结束一个命令包, 并打开显示。对于地址线 A0,A2~A17 控制板中已设定好, 用户只需控制 A1 便可完成命令的送入。

2.3 操作码及参数表

操作码	操作	参数 / 数据
10H	字符设置	字符选择(1 byte): 00: 内置的 8×8 ASCII 码 01: 嵌入的 8×8 CGRAM 02: 嵌入的 8×16 CGRAM 03: 嵌入的 16×16 CGRAM 04: 外置的 16×16 ROM (GB2312) 05: 外置的 16×16 ROM (BIG5)
12H	字符写入位置	字符坐标 (4 bytes) - x (2 bytes) - y (2 bytes)
14H	字符前景颜色	字符前景颜色 (2 bytes) 16-bit TFT(5R:6G:5B)
15H	字符背景颜色	字符背景颜色 (2 bytes) 16-bit TFT(5R:6G:5B)
16H	显示字符	显示字符 (1 or 2 bytes)
17H	显示字符串	显示字符串 - 字符个数 0~63 (1 byte) - 字符串(≤63 bytes)
19H	定义 CGRAM	CGRAM 位图 - 字符编码 (1 byte) - 位图模式(≤32 bytes)

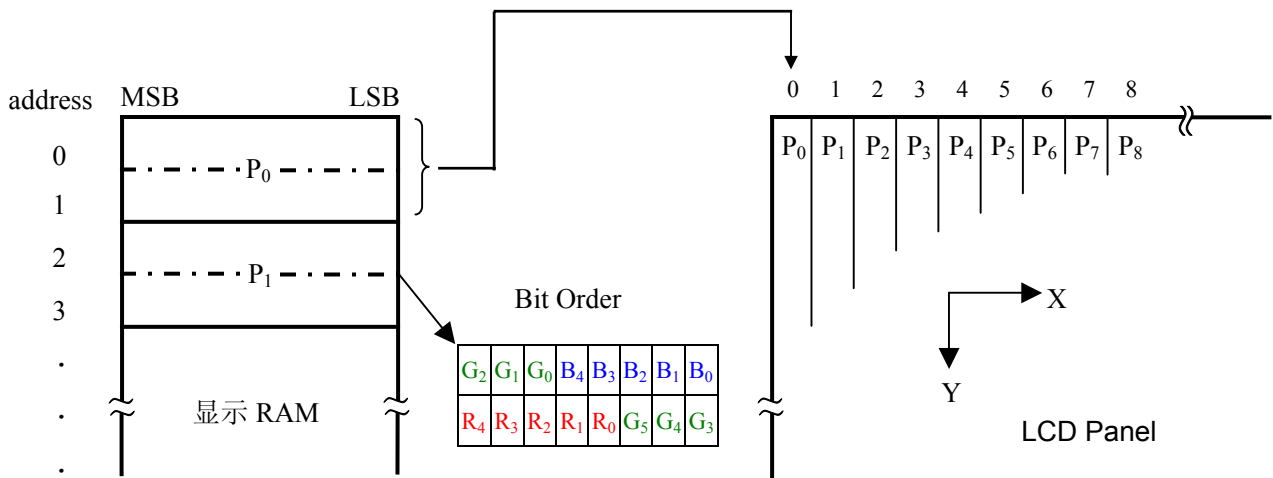
操作码	操作	参数 / 数据
20H	前景颜色	设置前景颜色 (2 bytes) 16-bit TFT(5R:6G:5B)
23H	画点	- x 坐标 (2 bytes) - y 坐标(2 bytes)
24H	画线	- x 起始坐标(2 bytes) - y 起始坐标(2 bytes) - x 终止坐标(2 bytes) - y 终止坐标(2 bytes)
26H	画矩形边框	- x 起始坐标(2 bytes) - y 起始坐标(2 bytes) - x 终止坐标(2 bytes) - y 终止坐标(2 bytes)
27H	画实心矩形	- x 起始坐标(2 bytes) - y 起始坐标(2 bytes) - x 终止坐标(2 bytes) - y 终止坐标(2 bytes)
28H	画圆	- 圆心 x 坐标(2 bytes) - 圆心 y 坐标(2 bytes) - 半径(1 byte)
29H	画实心圆	- 圆心 x 坐标(2 bytes) - 圆心 y 坐标(2 bytes) - 半径(1 byte)

操作码	操作	参数 / 数据
80H	更新设置	无
81H	设置内存指针	- 地址 (3 bytes)
83H	写寄存器	- 地址(2 bytes) - 数据(1 byte)
84H	写内存	- 数据个数(1 byte) - 数据(≤63 bytes)
8FH	内存时钟使能	“69 45 61 67 6C 65” (6 个字节 in HEX)

寄存器地址 A15-A0	读 / 写	描述
F004	只写	命令包入口: 写入命令包
F006	写	写控制和读状态入口: Bit[7:5]: 保留 Bit[4]: 0 显示开 1 显示关 Bit[3:1]: 保留 Bit[0]: 命令的结束标志 每个命令包后要写入 “1”
	读(串行)	Bit[7:1]: 保留 Bit[0]: 若为 1 表示 FIFO 满

注: 其它寄存器的设定请参考 TCB8000A 用户手册

2.4 LCD 屏幕显示数据形式



2.5 应用举例

1) 显示一幅 160×120 的彩色图片（常用函数见参考程序）

```

uchar code BMP[]={... ..};          // 数据略
uchar MemPtr[]={4,0x81,0x00,0x00,0x00};
uchar i,j;
uint k=0;
unsigned long addr;

ClearScreen(BLACK);                  // 用黑色清屏
addr=38560;                          // 图片起始位置为 2*X+640*Y;其中 X=80,Y=60
delay(10);

for(j=0;j<120;j++)
{
    MemPtr[2]=addr;
    MemPtr[3]=addr>>8;
    MemPtr[4]=addr>>16;
    WritePKG(MemPtr);                // 设置内存地址指针
    for(i=0;i<160;i++)
    {
        SdCmd(0x84);                // 送一个像素的数据到内存
        SdCmd(2);
        SdCmd(BMP[2*k+1]);
        SdCmd(BMP[2*k]);
        CmdEnd();                    // 结束一个命令包
        k++;
    }
    addr+=640;
}
    
```



2) 画点，线段和圆（常用函数见参考程序）

```

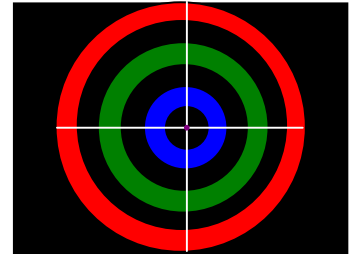
uint code color[]={WHITE,BLACK,BLUE,BLACK,GREEN,BLACK,RED,MAGENTA};
uchar circle[]={6,0x29,160,0,120,0,10};
uchar i;

ClearScreen(BLACK);           // 用黑色清屏

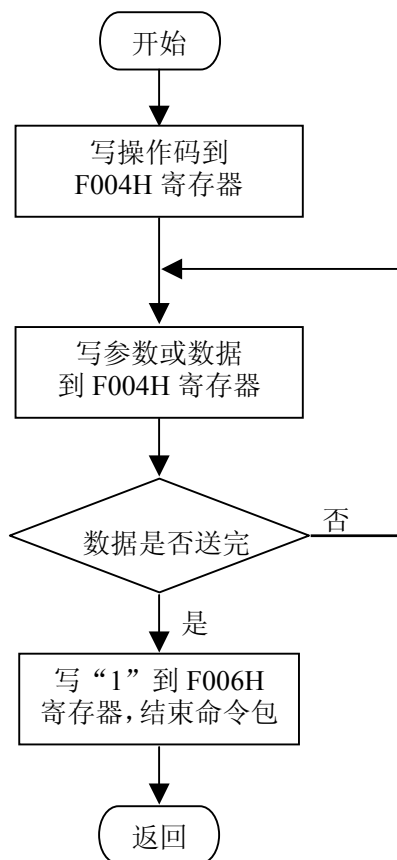
for(i=6;i>0;i--)
{
    SetFgColor(0x20,color[i]); // 设置前景颜色
    circle[6]=i*20;
    WritePKG(circle);          // 以 X=160,Y=120 为圆心画带填充颜色的同心圆
    delay(10);
}
SetFgColor(0x20,color[0]);
Draw_Rect(0x24,40,120,280,120); // 画线段 X1=40,Y1=120; X2=280,Y2=120
Draw_Rect(0x24,160,0,160,240);  // 画线段 X1=160,Y1=0; X2=160,Y2=240

SetFgColor(0x20,color[7]);
Draw_Dot(160,120);            // 画点 X=160,Y=120

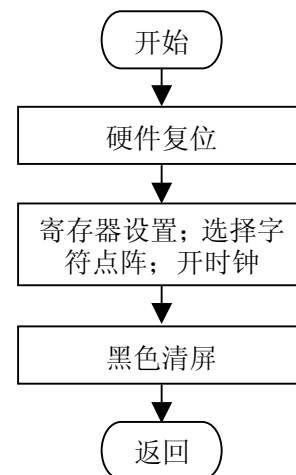
```



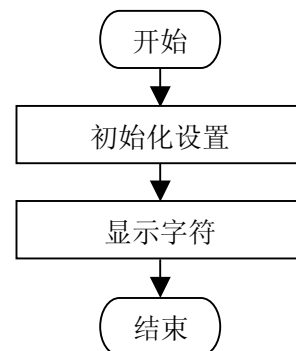
2.6 流程图



写命令包流程图



初始化流程图



主程序流程图

参考程序

```
//-----  
//采用带中文字库的 TCB8000A, 本程序的演示结果为:  
//  
//          深圳市拓普微科技开发有限公司  
//          TOPWAY TCB8000A  
//          http://www.topwaydisplay.com  
//-----  
  
#include <reg52.h>  
#include <intrins.h>  
  
#define uchar unsigned char          // 0~255  
#define uint unsigned int           // 0~65535  
  
#define LCDBUS    P1  
sbit    _CS      = P3^4;  
sbit    _RES     = P3^5;  
sbit    A1       = P3^6;  
sbit    _WR      = P3^7;  
  
#define RED      0xf800  
#define GREEN    0x07e0  
#define BLUE     0x001f  
#define YELLOW   0xffe0  
#define MAGENTA  0xf81f  
#define BLACK    0x0000  
#define WHITE    0xffff  
  
uchar  PkgFgColor[]={3,0x20,0xe0,0x07};    // 前景颜色  
uchar  PkgBgColor[]={3,0x15,0xe0,0x07};    // 背景颜色  
uchar  CharCoord[]={5,0x12,0x00,0x00,0x00,0x00}; // 字符写入位置  
  
//-----  
// 延时子程序  
//-----  
void delay(uint m)                          // 12MHz Xtal, close to ms value  
{  
    uint j;  
    uint i;  
  
    for(i=0; i<m; i++)  
        for(j=0; j<109; j++)  
            _nop_();  
}  
  
//-----  
// 写命令  
//-----  
void SdCmd(uchar Command)  
{  
    A1 = 0;                                // 置寄存器地址 F004H, 写入操作码, 参数或数据  
    LCDBUS = Command;  
    _CS = 0;  
    _WR = 0;  
    _nop_();  
    _WR = 1;  
    _nop_();  
    _CS = 1;  
}  
  
//-----  
//结束命令  
//-----  
void CmdEnd()  
{  
    A1 = 1;                                // 置寄存器地址 F006H  
    LCDBUS = 1;  
    _CS = 0;  
    _WR = 0;  
    _nop_();  
    _WR = 1;  
    _nop_();  
    _CS = 1;  
}
```

```
//-----  
//写入一个命令包  
//-----  
void WritePKG(uchar *pkg)  
{  
    uchar i;  
    delay(2);  
  
    for(i=*pkg;i;i--)  
        SdCmd(*(++pkg));  
    CmdEnd();  
}  
  
//-----  
//设置前景颜色  
//-----  
void SetFgColor(uchar par,uint color)  
{  
    PkgFgColor[1]=par;  
    PkgFgColor[2]=color;  
    PkgFgColor[3]=color>>8;  
    WritePKG(PkgFgColor);  
}  
  
//-----  
//设置背景颜色  
//-----  
void SetBgColor(uint color)  
{  
    PkgBgColor[2]=color;  
    PkgBgColor[3]=color>>8;  
    WritePKG(PkgBgColor);  
}  
  
//-----  
//由填充颜色清屏  
//-----  
uchar code PkgCLS[]={9,0x27,0x00,0x00,0x00,0x00,0x3f,0x01,0xef,0x00};  
void ClearScreen(uint color)  
{  
    SetFgColor(0x20,color);  
    WritePKG(PkgCLS);  
    delay(5);  
}  
  
//-----  
//画点  
//-----  
uchar dot[]={5,0x23,0x00,0x00,0x00,0x00};  
void Draw_Dot(uint X,uint Y)  
{  
    dot[2]=X;  
    dot[3]=X>>8;  
    dot[4]=Y;  
    dot[5]=Y>>8;  
    WritePKG(dot);  
    delay(10);  
}  
  
//-----  
//画线段或矩形  
//-----  
uchar rect[]={9,0x27,0x00,0x00,0x00,0x00,0x3f,0x01,0xef,0x00};  
void Draw_Rect(uchar para,uint x0,uint y0,uint x1,uint y1)  
{  
    rect[1]=para;  
    rect[2]=x0;  
    rect[3]=x0>>8;  
    rect[4]=y0;  
    rect[5]=y0>>8;  
    rect[6]=x1;  
    rect[7]=x1>>8;  
    rect[8]=y1;  
    rect[9]=y1>>8;  
    WritePKG(rect);  
    delay(10);  
}
```



```
//-----
//设置字符坐标
//-----
void SetCharCoord(uint coordX,uint coordY)
{
    CharCoord[2]=coordX;
    CharCoord[3]=coordX>>8;
    CharCoord[4]=coordY;
    CharCoord[5]=coordY>>8;
    WritePKG(CharCoord);
}

//-----
//写入 16*16 的字符串
//-----
void Printstr(uchar *pstr)
{
    while(*pstr>0)
    {
        SdCmd(0x16);
        SdCmd(*pstr++);
        SdCmd(*pstr++);
        CmdEnd();
        delay(5);
    }
}

//-----
//写入 8*8 或 8*16 的字符串
//-----
void Print(uchar *pstr)
{
    while(*pstr>0)
    {
        SdCmd(0x16);
        SdCmd(*pstr++);
        CmdEnd();
        delay(5);
    }
}

//-----
//初始化设置
//-----
uchar code SRAM01[]={4,0x83,0x00,0xf5,0x00};
uchar code SRAM02[]={4,0x83,0x01,0xf5,0xe0};
uchar code ROM01[]={4,0x83,0x04,0xf5,0x44};
uchar code ROM02[]={4,0x83,0x05,0xf5,0x80};

uchar code LCDConfig01[]={4,0x83,0x80,0xf0,0xfc};
uchar code LCDConfig02[]={4,0x83,0x8e,0xf0,0x32};

uchar code LCDConfig03[]={4,0x83,0x90,0xf0,0x14};
uchar code LCDConfig04[]={4,0x83,0x91,0xf0,0x24};
uchar code LCDConfig05[]={4,0x83,0x92,0xf0,0x1e};
uchar code LCDConfig06[]={4,0x83,0x94,0xf0,0x05};
uchar code LCDConfig07[]={4,0x83,0x95,0xf0,0x0e};
uchar code LCDConfig08[]={4,0x83,0x96,0xf0,0x03};

uchar code Clock[]={7,0x8f,0x69,0x45,0x61,0x67,0x6c,0x65};

uchar CharSet[]={2,0x10,0x04};
//uchar CharSet[]={2,0x10,0x00};

void initLCDM(void)
{
    _RES=0;
    delay(100);
    _RES=1;
    delay(300);

    WritePKG(SRAM01);
    WritePKG(SRAM02);
    WritePKG(ROM01);
    WritePKG(ROM02);
    WritePKG(LCDConfig01);
    WritePKG(LCDConfig02);
}
```

```
// memory 脉冲宽度
// memory 16 位数据线
// 外部字库 脉冲宽度
// 外部字库 8 位数据线

// panel 选择
// Pixel Clock 6MHZ

// Horizontal Front Porch
// Horizontal Back Porch
// Horizontal Sync 脉冲宽度
// Vertical Front Porch
// Vertical Back Porch
// Vertical Sync 脉冲宽度

// 时钟使能

// 字库设置: 中文字库
// 字库设置: ASCII 码
```

```
WritePKG(LCDConfig03);
WritePKG(LCDConfig04);
WritePKG(LCDConfig05);
WritePKG(LCDConfig06);
WritePKG(LCDConfig07);
WritePKG(LCDConfig08);

WritePKG(CharSet);           // 16×16 GB2312
WritePKG(Clock);            // 开时钟

ClearScreen(BLACK);         // 清屏
}

//-----
//主程序
//-----
void main()
{
    SP=0x60;
    EA=0;

    LCDBUS=0xff;
    _CS=1;
    _RES=1;
    A1=0;
    _WR=1;

    initLCDM();              // 初始化 LCD 控制器和面板

    SetFgColor(0x14,BLUE);
    SetBgColor(BLACK);
    SetCharCoord(48,70);
    Printstr("深圳市拓普微科技发展有限公司"); // 显示 16×16 汉字

    SetFgColor(0x14,RED);
    SetBgColor(BLACK);
    SetCharCoord(100,120);
    Print("TOPWAY TCB8000A"); // 显示 8×16 半宽字符

    SetFgColor(0x20,YELLOW); // 画带填充颜色的矩形
    Draw_Rect(0x27,52,189,278,198);
    SetFgColor(0x14,BLACK);
    SetBgColor(YELLOW);      // 字体填充颜色
    SetCharCoord(54,190);
    CharSet[2]=0;
    WritePKG(CharSet);
    Print("http:\\\\www.topwaydisplay.com"); // 显示 8×8 ASCII 字符

    while(1)
    {
    }
}
```