8 位接口 TFT LCD 控制器 使用说明

一、概述

TFT LCD 控制板特点：

1、8 位并行处理器接口，适合一般 51 单片机和 ARM 单片机使用。

2、输入具有双缓冲功能，即使软件不查询 BUSY 信号也可获得较高的写入速率和正确的显示效果。

3、16 位并行 RGB 输出接口，可支持 8 位（256 色）显示或 16 位（65K 色）显示（分属不同产品型号）。

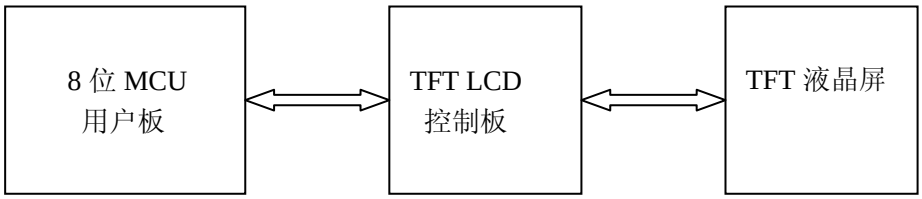
4、输出时序具有高稳定性，抗干扰能力强。

5、具备 2 页（或更多）显示页面，切换不闪烁，编程简单。

6、支持 640×480、800×600、320×240、800×480 等分辨率。

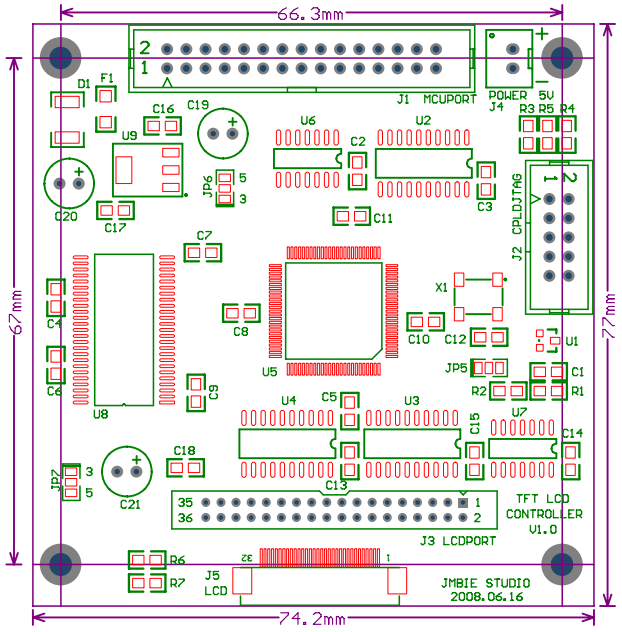
TFT LCD 控制板由 CPLD 产生稳定的 TFT 时序驱动 LCD 面板，以及 SDRAM（显存）时序读取显存的数据。并能接受 MCU 写入的数据，保存到 SDRAM 的指定位置。

TFT LCD 控制板在系统中的应用如下图所示：



TFT LCD 控制板可以通过用户板的扁平电缆供电，也可以单独供电。可以连接 5V 的 MCU，也可以连接 3.3V 的 MCU；可以驱动 5V 的 LCD 屏，也可以驱动 3.3V 的 LCD 屏。接口电压类型、TFT LCD 面板类型（不同的分辨率、位数、时序）以及显存容量均可定制。

二、TFT LCD 控制板接口说明。

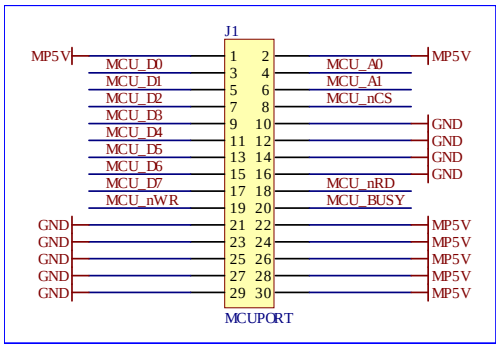


控制板 PCB 尺寸和引脚位置图

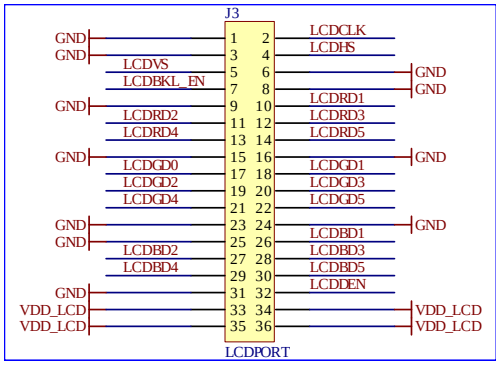
TFT LCD 控制板和 MCU 系统板的连接插座是 J1 MCUPORT，为 2.54mm 间距 30 脚的标准扁平线（线距 1.27mm）插座。引脚定义见下页图。

控制板有两种连接 LCD 屏的接口，一个是 J3，为 2.0mm 间距 36 脚的扁平线（线距 1.0mm）插座；另一个是 J5，为 0.5mm 间距 32 脚的薄膜线插座。从 J3 可以用扁平线或分立线将 LCD 信号引出，方便调整线序。J5 的线序可以兼容很多种数字接口的 16 位 TFT LCD 屏。

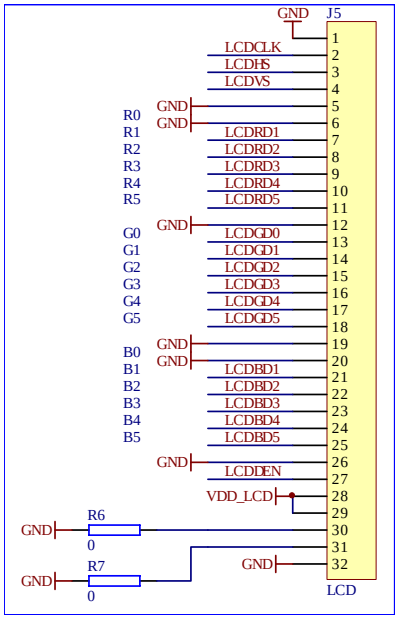
J3 和 J5 的具体形式根据所配屏决定，其引脚定义如下页图所示：



J1　MCU 接口定义



J3　LCD 接口定义



J5　LCD 薄膜线插座引脚定义

图中 MCU_D0 到 MCU_D7 是单片机的数据线，MCU_nWR、nRD 分别是写、读信号。MCU_A0、A1、nCS 分别是地址线和片选信号（片选也可以是地址线）。MCU_BUSY 是控制板输出的握手信号，为高电平时表示控制板输入缓冲区满，软件应暂停写入数据，直到查询该信号电平为低时再写入数据，以免显示错误。

图中 LCDCLK、LCDHS、LCDVS、LCDDEN 分别是 LCD 屏的位时钟、水平同步、垂直同步、数据使能等信号。LCDRD1 到 LCDRD5 是红色信号，LCDGD0 到 LCDGD5 是绿色信号，LCDBD1 到 LCDBD5 是蓝色信号，按 RGB 5－6－5 格式构成 16 位色信号。

对于 8 位色版本的控制板实际只输出 8 位色，RGB 3－3－2 格式，有效信号是 RD3 到 RD5，RG3 到 RG5 和 RB4、RB5。其余信号均输出低电平。

控制板上的 R6 和 R7 可以设置 LCD 屏的画面方向，可以水平或垂直翻转画面，具体参考 LCD 屏的手册。

三、软件操作说明

1. MCU 接口时序：TFT LCD 控制板支持标准 8051 系列 MCU 的读写时序，即用

```
MOVX    @DPTR，A
MOVX    A，  @DPTR
```

指令访问控制板的 4 个寄存器（端口）。具体时序见 8051 手册。

其他处理器，比如带有并行总线但没有 LCD 控制器的 ARM 微控制器，外部总线往往是 16 位甚至更高，可以只使用其低 8 位，或直接配置为 8 位总线使用。同时，ARM 微控制器的总线时间参数可以设置，在后面的程序样例中给以说明。

当然，我们有 16 位并行总线接口的 LCD 控制板，在显示 16 位色的时候效率会高一倍多，非常适合 16 位单片机。另外还有 SPI 接口的 LCD 控制板，适合没有并行总线的 ARM 微控制器。这两类控制板在本文档中不做详细介绍。

2．MCU 信号电平组合：

nCS	nWR	nRD	A1	A0	D0－D7
H	任意	任意	任意	任意	操作无效
L	L	H	0	0	写数据到显存
L	H	L	0	0	从显存读数据
L	L	H	0	1	设置 LCD 行号（垂直 y 坐标）低字节
L	L	H	1	0	设置 LCD 列号（水平 x 坐标）低字节
L	L	H	1	1	设置操作字节（包含行列地址高位）

3．CPLD 内部含有 4 个寄存器，对 MCU 而言就是 4 个地址端口：

数据寄存器	地址 A1=0	A0=0	MCU 读写显存数据的缓冲寄存器。
行号寄存器	地址 A1=0	A0=1	LCD 行号就是象素的垂直 y 坐标（低 8 位）
列号寄存器	地址 A1=1	A0=0	LCD 列号就是象素的水平 x 坐标（低 8 位）
设置寄存器	地址 A1=1	A0=1	包含行号的高 1 位和列号的高 2 位及其他控制信号

640×480 分辨率版本的设置寄存器的位定义：

BIT7　保留
BIT6　保留
BIT5　如果为 1，bit4、bit3 才有效，为 0 则忽略 bit4、bit3。
BIT4　当前显示的页号
BIT3　当前写入数据的页号
BIT2　行号最高 1 位
BIT1-BIT0　列号最高 2 位

800×600 分辨率版本的设置寄存器的位定义：

BIT7　保留
BIT6　保留
BIT5　保留
BIT4　保留
BIT3-BIT2　行号最高 2 位
BIT1-BIT0　列号最高 2 位

对于分辨率是 640×480 的 LCD 屏，行号（象素 y 坐标）的取值范围是 0－479，列号（象素 x 坐标）的取值范围是 0－639。这个数值范围用 1 个字节无法存放，所以在设置寄存器中设置行号的最高 1 位和列号的最高 2 位。

对于分辨率是 800×600 的 LCD 屏，行号（象素 y 坐标）的取值范围是 0－599，列号（象素 x 坐标）的取值范围是 0－799。这个数值范围用 1 个字节无法存放，所以在设置寄存器中设置行号的最高 2 位和列号的最高 2 位。

由于控制板上显存 SDRAM 的容量为 2M，在 800×600 分辨率的版本中，无法多页显示，所以没有显示页号的设置。当然，我们另有更大容量显存芯片的控制板型号。

控制板有一个重要的功能就是，每次写数据寄存器之后，列号（象素的 x 坐标）会自动加 1，如果软件需要连续写一行内的若干象素，可以连续写数据寄存器，而不用反复设置列号寄存器。但读数据寄存器时没有 x 坐标自动加 1 功能，一般情况读数据只限于局部，数据

量不大，可以软件反复设置坐标。

对于分辨率是 640×480 的控制板，控制板支持双页显示，就是在显存中可保持两个显示页面的数据，软件可以设置其中一个页面显示在 LCD 屏上，而 MCU 将数据写入另一个页面。显示页面和写入页面相同时可能会看到数据写入（画面刷新）的过程，而显示页面和写入页面不同时则可避免这个现象，但一般只适合全屏刷新，比如清屏。

切换显示页需要通过写设置寄存器来实现，写入设置寄存器的字节中 bit5 如果为 1，bit4、bit3 才有效，bit5 为 0 则忽略 bit4、bit3。这样，当需要切换显示页时，保持 bit5 为 1 即可，而无需关心行列地址高位，而写入行列地址高位时，bit5 为 0 即可，bit4 和 bit3 不会影响显示页号，所以软件中每次更新行列地址时不必把显示页号合并到地址字节中，提高了效率。

4. 对于 8 位色（共 256 种颜色）的控制板，每操作一个象素读写一个字节的数据即可。数据字节位定义：

BIT7 红色高 对应 LCD 信号的 RD5
BIT6 红色中 对应 LCD 信号的 RD4
BIT5 红色低 对应 LCD 信号的 RD3
BIT4 绿色高 对应 LCD 信号的 GD5
BIT3 绿色中 对应 LCD 信号的 GD4
BIT2 绿色低 对应 LCD 信号的 GD3
BIT1 蓝色高 对应 LCD 信号的 BD5
BIT0 蓝色低 对应 LCD 信号的 BD4

对于 16 位色（共 65536 种颜色）的控制板，每操作一个象素需要读写两个字节的数据，8 位接口则需要分两次传输。先写入低字节，再写入高字节，只有在写入高字节后，控制板才会认为是一个象素写入，所以软件务必保证写入数据的完整性。

16 位数据高字节位定义：

BIT15 对应 LCD 信号的 RD5
BIT14 对应 LCD 信号的 RD4
BIT13 对应 LCD 信号的 RD3
BIT12 对应 LCD 信号的 RD2
BIT11 对应 LCD 信号的 RD1
BIT10 对应 LCD 信号的 GD5
BIT9 对应 LCD 信号的 GD4
BIT8 对应 LCD 信号的 GD3

16 位数据低字节位定义：

BIT7 对应 LCD 信号的 GD2
BIT6 对应 LCD 信号的 GD1
BIT5 对应 LCD 信号的 GD0
BIT4 对应 LCD 信号的 BD5
BIT3 对应 LCD 信号的 BD4
BIT2 对应 LCD 信号的 BD3
BIT1 对应 LCD 信号的 BD2
BIT0 对应 LCD 信号的 BD1

5. 由于每次处理器写入数据后，CPLD 都需要时间将显示数据保存到 SDRAM 中的指定位置，所以控制板有象素写入速度的限制，约 3.75M 次每秒，超过此极限，会出现写入

数据丢失的现象。对于 51 单片机而言，晶体频率 20M，X2 模式，汇编指令连续写入时不会发生显示错误现象。对于频率更高的单片机，如果出现显示花点或丢点的问题，可以通过两种方式解决，一是插入延时，二是查询 BUSY 信号，为高电平时表示控制板忙，暂停写入数据。由于控制板采用输入双缓冲结构，所以可保证在不是很高的写入速率下能正确保存显示数据，而不用查询 BUSY 信号。

6. MCU 编程方法：

首先需要对坐标值进行字节拆分和移位。

1）将 X、Y 坐标的最高位（拼成 1 字节）写入操作寄存器（A1=1 A0=1）。

2）将 X 坐标的低 8 位写入列号寄存器（A1=1 A0=0）。

3）将 Y 坐标的低 8 位写入行号寄存器（A1=0 A0=1）。

4）将象素颜色数据写入数据寄存器（A1=0 A0=0）。列号寄存器（控制板内部的 x 坐标）会自动加 1。当写入数据寄存器后，控制板才开始进行保存动作，所以必须先行写入行列地址再写入数据。

5）对于 16 位色版本，需要再写入象素颜色数据的高字节，之后控制板才进行保存动作。

读数据过程和以上类似，但是设置坐标之后，控制板需要时间将显存的数据转移到数据寄存器，所以根据效果调节写行列号寄存器到读数据寄存器之间的时间间隔，在此 BUSY 信号不起作用。

四、程序样例

以 STC89C516RD+单片机为例，控制板的 A1 接 MCU 的 P2.1，A0 接 MCU 的 P2.0，CS 接 MCU 的 P2.7。如此定义以下访问地址的宏：

```
#define LcdDataPort      *((unsigned char xdata *)0x4000)
#define LcdRowLowPort    *((unsigned char xdata *)0x4100)
#define LcdColLowPort    *((unsigned char xdata *)0x4200)
#define LcdOperatPort    *((unsigned char xdata *)0x4300)
```

这里为了避开 STC89C516RD+的内部扩展 RAM 空间，令 P2.6 为 1。

然后编写下面的写入象素的函数，基于此可完成其他画面操作。

```
void LcdPixel(int x,y,unsigned char color) //640*480
{
  LcdOperatPort = (x/256)|((y/64)&0x04);
  LcdColLowPort = x;
  LcdRowLowPort = y;
  LcdDataPort   = color;
}

void LcdPixel(int x,y,unsigned char color) //800*600
{
  LcdOperatPort = (x/256)|((y/64)&0x0c);
  LcdColLowPort = x;
  LcdRowLowPort = y;
  LcdDataPort   = color;
}
```

```
void LcdPixel(int x,y,unsigned int color) //640*480 16位色
{
  LcdOperatPort = (x/256)|((y/64)&0x04);
  LcdColLowPort = x;
  LcdRowLowPort = y;
  LcdDataPort   = color;
  LcdDataPort   = color/256;
}

unsigned int GetPixel(int x,y) //16位色 读数据
{
  unsigned int c=0;
  LcdOperatPort = (x/256)|((y/64)&0x04);
  LcdColLowPort = x;
  LcdRowLowPort = y;
  c=LcdDataPort
  c=c+(LcdDataPort*256);
  return c;
}
```

随此文档，我们提供与 TFT LCD 控制板配套的 Keil C51 工程，基于 8051 的单片机（比如 AT89S52、STC89C516RD+等等），包含基本的清屏、画点、画线、画矩形、画圆等函数以及显示 BMP 图片和文字等内容。仅供用户了解如何对控制板进行软件编程。

感谢您使用我们的 TFT LCD 控制板，我们会不断增加新的支持型号和功能。