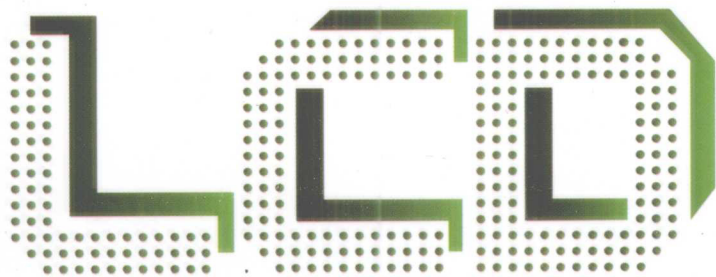


张新强 编著

点阵



# 驱动显控原理与实践



北京航空航天大学出版社

# 点阵 LCD 驱动显控 原理与实践

张新强 编著

北京航空航天大学出版社

## 内 容 简 介

本书主要从单片机编程方面介绍点阵 LCD 显示模块的驱动显控原理,总结出一套适用于各种单片机平台并适用于各种点阵 LCD 模块的驱动程序。在此基础上,重点讲解了将该驱动程序移植至不同单片机平台的方法,以及几种针对具体的 LCD 模块特性调整驱动程序的方法。

以 MCS-51 单片机为基础,以 ST 公司推出的 STM32 系列 32 位微控制器作为介绍应用的平台,以 Keil  $\mu$ Vision 集成开发环境作为程序设计和调试的环境。所有程序都使用 C 语言编写,所以,书中例程及编程方法亦适用于其他单片机平台。

本书适合普通高校计算机类、电子类、电气自动化等专业学生作为参考用书。同样,也适用于电子爱好者以及从事嵌入式应用设计的工程师作为实践工作的参考。

### 图书在版编目(CIP)数据

点阵 LCD 驱动显控原理与实践/张新强编著. —  
北京:北京航空航天大学出版社,2010.3  
ISBN 978-7-5124-0021-4

I. ①点… II. ①张… III. ①液晶显示器—电路设计  
IV. ①TN873

中国版本图书馆 CIP 数据核字(2010)第 023135 号

© 2010,北京航空航天大学出版社,版权所有。

未经本书出版者书面许可,任何单位和个人不得以任何形式或手段复制本书内容。  
侵权必究。

### 点阵 LCD 驱动显控原理与实践

张新强 编著

责任编辑 董立娟

\*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100191) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail:cmsbook@gmail.com

北京市媛明印刷厂印装 各地书店经销

\*

开本:787×960 1/16 印张:15.5 字数:347千字

2010年3月第1版 2010年3月第1次印刷 印数:4000册

ISBN 978-7-5124-0021-4 定价:28.00元

# 前 言

本书主要介绍一种通用点阵 LCD 显示模块的驱动编程方法(在本书中被称为通用 LCD 驱动程序),该驱动程序可以方便地移植到不同的单片机上使用,也可以经过简单的修改应用到不同的 LCD 模块应用中。本书以一款较典型规格的 COG 单色点阵 LCD 模块为例,详细介绍 LCD 模块的驱动方法、构建通用 LCD 驱动程序的思路,以及基于该通用 LCD 驱动程序的简单菜单应用程序、移植驱动程序的方法。本书从最底层 LCD 模块的接口驱动编程到稍上层的字符显示、绘图显示编程来介绍通用 LCD 驱动程序的编程思想;此外,在介绍了几种不改变通用 LCD 驱动程序架构的基础上,利用 LCD 模块及单片机本身的特性来提高驱动程序效率的方法;结合彩色 TFT 显示模块,介绍几种彩色图像的显示方法以及简单的综合应用方案供读者参考。笔者将本书的内容以及程序呈现给读者,但不希望读者以完全照搬的方式来使用书中介绍的程序,而是希望带给读者一种驱动控制的编程方法。

书中的部分内容已于 2007 年在网络上推出第一版,很多阅读过的朋友都反映不错,给予了笔者一定的肯定,此为笔者最感欣慰之事。而随着时间的推移,笔者又在多种单片机平台以及多种产品设计当中使用了本书介绍的通用 LCD 驱动程序;而对于不同种类的 LCD 模块,笔者也将通用 LCD 驱动程序套用于其中,实践证明了它的可移植性和通用性。不过,随着单片机平台以及 LCD 模块的变化,在很多应用中,或多或少都对原版的通用 LCD 驱动程序进行了改进,以便于充分发挥单片机及 LCD 模块的性能,这点在本书中将会有一章的内容来介绍。

本书在介绍基于单色点阵 LCD 模块的通用 LCD 驱动程序时,以 MCS-51 单片机为介绍的重点,以便于众多具备 51 单片机编程基础的读者能够更快地理解;而对于近两年风靡 MCU 市场的 ST 公司的 STM32 系列 32 位 ARM 芯片,本书也针对它编写了大量的例程,并且在第 4 章中详细介绍如何将本书介绍的 LCD 驱动程序移植到该系列 MCU 之上。

## 本书更适合什么样的 LCD 模块?

本书主要针对单色的点阵 LCD 进行介绍,而且是针对本身就集成了驱动控制 IC 以及显示的 LCD 模块。那些字符型以及段码型的 LCD 不在介绍之列,但也可以在一定程序上参考本书的驱动编程方法来编写这些 LCD 的驱动程序。此外,有些小规模的一般是 3.5 寸以下)彩色 TFT LCD 也有内置驱动控制器以及显存,也可参考这里的介绍来编写它们的驱动程序。

## 这里的 LCD 驱动程序更适合什么样的 MCU?

MCU 即常说的单片机,本书介绍的通用 LCD 驱动程序适用于各种类型 MCU;只不过笔

者从应用的角度出发,为读者建议更适合使用本书介绍驱动程序的 MCU 类型。

目前市面上的 MCU 非常多,从功能和资源角度来看,大概可分为以下几类:

- ▶ 小资源 MCU。类似于传统 51 的 89S51 单片机、PIC 等的小资源单片机等;通常它们的资源都很少,片内的 ROM 少于或等于 4 KB, RAM 少于或等于 128 字节,速度较慢, MIPS 数通常在 10 MMIPS 以内。
- ▶ 中资源 MCU。这类 MCU 的涵盖面非常广,在实际的产品设计中应用非常多,如增强型 51 单片机、中资源的 AVR 单片机、16 位的 MSP430 系列单片机、凌阳的 SPCE061A、PIC 的中等资源单片机等,甚至包含至 ARM7 核心的 LPC 系列 MCU(如 LPC21XX 系列等);一般来说指的是片内的 ROM 资源在 8 KB 以上, RAM 在 256 字节以上, MCU 的运行速度较快,片内资源丰富,应用面非常广;而 STM32 是一个非常强劲的 MCU,推荐大家使用。
- ▶ 跑操作系统的大资源 MCU。这类 MCU 其实大部分指 ARM7、ARM9 核心或与这些核心同等级的处理器了,通常都会在设计中跑操作系统,也就是现在常说的 32 位嵌入式处理器。

本书介绍的 LCD 驱动程序更适合于中等资源的 MCU,因为它们有足够的片内资源和运行的速度,而且在应用它们的设计当中往往会涉及 LCD 的人机界面显示。

其实这里就是要说明,如果使用一些类似 2051、89S51 级别的 MCU,就没有太多必要使用本书所介绍的驱动程序,因为此类 MCU 存储空间有限,从而以致于很有可能连 LCD 驱动中的自带字库都装不下去,不过驱动的方法还是可以参考的。而如果使用的是较大资源的嵌入式处理器,而且本身就跑着图形操作系统的话,那就更没必要了,因为往往这类 MCU 都会自带 LCD 的控制器,而操作系统也带有完整的图形界面接口。

### 为什么用 C 语言?

本书中全部的源码都是以 C 语言为平台的。当前,没有 C 语言编译器支持的中等资源的 MCU 是极其少见的;使用 C 语言对 MCU 进行开发已经形成当前的主流模式。

ASM 的执行效率比 C 语言的代码要好? 当然,这是肯定的,不过现在 MCU 速度和性能已经提升上来了,而且很多优秀的 C 编译器编译的结果就未必比自己使用 ASM 编程的结果效率低。

ASM 的代码短小精悍? 从某个角度来说是的,不过时下优秀的 C 编译器对 C 代码的优化比自己使用 ASM 编写得还要好;况且,在编写结构复杂的程序时, C 语言肯定比 ASM 占优势,比如编程速度、可读性以及可移植性等。

本书共分 12 章,第 1~3 章详细介绍了通用 LCD 驱动程序的架构;读者可以通过这 3 章的介绍,理解贯穿本书的通用 LCD 驱动程序的编程思想。第 4~6 章介绍了通用 LCD 驱动程序移植到不同 MCU(以 STM32F103RC 为例)的方法;也以几种常用的 LCD 模块为例,介绍

将通用 LCD 驱动程序移植到不同 LCD 模块的应用;另外,还针对某些 MCU 平台和 LCD 模块的特点对通用 LCD 驱动程序进行优化。第 7~11 章的内容介绍了彩色 TFT 模块的显示驱动方法;而针对彩色图像的显示应用,这里介绍了彩色图像取模显示、BMP 文件解码显示和 JPG 文件解码显示的例子。第 12 章介绍了一个简单的 SD 卡图像浏览器的方案。

编写本书的过程中得到很多朋友的帮助,在此致以诚挚的谢意:感谢 Powerint 提供 JPG 解码应用库,以及在长期的合作中给予笔者在编程思想上的启发;感谢身边的同事以及朋友,为笔者专心写书给予种种生活以及工作上的帮助;感谢“我们正在学习 ARM”QQ 群的各位网友,没有他们的支持,也不会有编写本书的创意。

本书的出版得到了北京航空航天大学出版社的大力支持,在此表示感谢。另,特别感谢牵线笔者与出版社的网友——“得蜜”。

本书在规划和编排过程中,稍显仓促,如果有不足之处,敬请各位读者指正。欢迎发邮件到 [xinqiang@mzdesign.com.cn](mailto:xinqiang@mzdesign.com.cn) 与笔者进一步交流,或联系 QQ:55693338。

笔 者

2009 年 11 月



## 第 1 章 典型单色点阵 LCD 模块

1.1 MzL02LCD 模块简介 .....	1
1.1.1 LCD 模块的结构 .....	1
1.1.2 MzL02 结构尺寸 .....	2
1.2 LCD 模块总线选择及时序图 .....	4
1.2.1 6800 并行总线接口 .....	4
1.2.2 8080 并行总线接口 .....	5
1.2.3 SPI 串行接口 .....	6
1.3 显存映射情况 .....	7
1.3.1 显存与 LCD 屏上点的关系 .....	7
1.3.2 行、列地址 .....	8
1.4 LCD 的 6800 总线接口 .....	9
1.5 LCD 控制器的特性 .....	10
1.6 LCD 驱动的基本流程 .....	16
1.6.1 LCD 模块的连接 .....	16
1.6.2 控制 LCD 模块显示一个点 .....	18

## 第 2 章 点阵 LCD 的驱动与显控

2.1 驱动程序架构思想 .....	21
2.2 基本驱动程序 LCD_Driver_User .....	22
2.2.1 端口配置头文件 LCD_Portconfig .....	22
2.2.2 MCU 与 LCD 的基本时序控制程序 .....	25
2.3 LCD 的初始化 .....	28
2.4 绘点子程序 .....	30
2.4.1 基本绘点函数 .....	30
2.4.2 一些扩展的基础功能函数 .....	32
2.5 驱动配置头文件 LCD_Config .....	33
2.6 LCD 驱动功能接口程序 .....	39

2.6.1	基本绘图功能函数	39
2.6.2	字符显示功能函数	44
2.7	字符显示原理	45
2.7.1	字符与字模	45
2.7.2	字模与字库	47
2.7.3	用绘点来绘制字符	53
2.7.4	显示驱动中自定义的中文字符	56
<b>第3章 Mz_MenuGUI 菜单应用</b>		
3.1	Mz_MenuGUI 简介	58
3.2	Mz_MenuGUI 的源码分析	59
3.2.1	Menu_Resource.c 菜单资源定义	59
3.2.2	Menu_GUI_Config.h 菜单 GUI 配置头文件	64
3.2.3	Menu_GUI.c 菜单接口函数	64
3.3	个性化 Menu 菜单界面	72
3.3.1	供参考的 GUI 响应控制代码	72
3.3.2	订制一个有二级菜单的工程	76
<b>第4章 将通用 LCD 驱动程序移植到 STM32</b>		
4.1	修改驱动中的底层代码	85
4.1.1	修改 LCD_PortConfig.h 的端口配置	86
4.1.2	修改底层驱动功能函数	87
4.2	与编译器相关的修改	89
4.3	例程软件结构说明	90
4.3.1	工程结构简介	90
4.3.2	工程中各文件简介	92
<b>第5章 将通用 LCD 驱动程序移植到其他 LCD 模块</b>		
5.1	LCD 控制器为 KS0108B 的模块	93
5.1.1	MzL613 模块简介	94
5.1.2	KS0108B 驱动控制方法	97
5.1.3	将 LCD 驱动程序移植到 KS0108B 控制器的 LCD 模块	101
5.2	LCD 控制器为 T6963C 的模块	108
5.2.1	MzL728 模块简介	108
5.2.2	T6963C 控制器的控制方法	113
5.2.3	控制器指令介绍	117
5.2.4	将 LCD 驱动程序移植到 T6963C 控制器的 LCD 模块	124



<b>第 6 章 基于通用 LCD 驱动程序的特殊应用</b>	
6.1 利用 MCU 内存来优化 LCD 的操作速度	133
6.1.1 优化驱动效率的前提条件	133
6.1.2 修改驱动程序的思路和方法	134
6.2 利用 MCU 的 DMA 及 SPI 驱动 COG 液晶模块	137
6.2.1 适用条件	137
6.2.2 STM32 的内部 RAM 作为 LCD 显存	138
6.2.3 修改驱动程序的思路和方法	139
<b>第 7 章 彩色 TFT 模块驱动显控</b>	
7.1 将 LCD 驱动程序移植到彩色 TFT 模块应用中	148
7.2 彩色 TFT 液晶显示模块介绍	148
7.2.1 MzT24 彩色 TFT 模块简介	148
7.2.2 显示 RAM 区映射情况	149
7.2.3 MzT24 操作时序	150
7.2.4 控制方法及 LCD 显示特性	151
7.3 STM32 驱动控制 MzT24	154
7.3.1 STM32F103RCT6 与 MzT24 模块的连接	155
7.3.2 修改 LCD_Driver_User.c 文件	156
7.3.3 修改 LCD_Config.h 的配置	164
7.3.4 一些简单的速度优化	164
<b>第 8 章 TFT 模块的扩展功能驱动程序</b>	
8.1 驱动功能扩展程序 LCD_Extend	167
8.1.1 一些有用的功能扩展	167
8.1.2 功能代码分析	167
8.2 窗口操作功能扩展程序 Dis_Window	171
8.2.1 窗口功能扩展程序的必要性	171
8.2.2 功能代码分析	171
8.2.3 Dis_Window 的作用	175
<b>第 9 章 RGB565 格式图像取模显示</b>	
9.1 RGB565 格式图像取模显示简介	176
9.2 图像取模	177
9.2.1 工具简介	177
9.2.2 彩色图像 RGB565 取模的步骤	177

9.3	代码分析 .....	179
9.4	显示效果参考 .....	182
<b>第 10 章 BMP 文件解码显示</b>		
10.1	BMP 文件解码显示简介 .....	184
10.2	BMP 文件格式 .....	185
10.2.1	BMP 文件构成分析 .....	185
10.2.2	位图文件头结构 .....	188
10.2.3	位图信息头结构 .....	189
10.2.4	色表结构分析 .....	190
10.2.5	图像数据分析 .....	191
10.3	将 BMP 文件数据整合进 Keil 的工程 .....	192
10.4	BMP 文件解码显示程序分析 .....	194
10.5	应用参考 .....	206
<b>第 11 章 JPG 文件解码显示</b>		
11.1	JPG 解码算法库 .....	208
11.1.1	JPG 解码库简介 .....	208
11.1.2	JPG 解码库的特点 .....	209
11.1.3	JPG 解码显示的软件架构 .....	209
11.2	将 JPG 文件数据整合进 Keil 的工程 .....	211
11.3	JPG 文件的解码显示程序分析 .....	211
11.4	应用参考 .....	216
<b>第 12 章 基于 SD 卡文件系统的简易图像浏览器方案</b>		
12.1	方案简介 .....	219
12.1.1	方案硬件平台 .....	219
12.1.2	方案软件结构 .....	222
12.2	基于 EFSL 的 SD 卡 FAT 文件系统简介 .....	223
12.2.1	SD 卡驱动程序 .....	223
12.2.2	EFSL 应用函数接口 .....	225
12.3	修改图像解码显示程序代码 .....	228
12.3.1	修改 BMP 文件解码显示程序 .....	228
12.3.2	修改 JPG 文件解码显示程序 .....	229
12.4	整体功能程序简介 .....	230

# 第 1 章

## 典型单色点阵 LCD 模块

### 1.1 MzL02LCD 模块简介

当前市面上的 LCD 模块种类非常多。各个厂家生产的编号都有所不同,即使使用同样的玻璃、同样的驱动控制 IC(芯片)都有可能存在不同的产品编号;但真正意义上对于应用设计者(软/硬件工程师)来说有用的,只是 LCD 模块当中的驱动控制 IC(或称为驱动控制器)型号以及驱动控制器芯片与玻璃的连接方法(也就是生产 LCD 模块时驱动控制器与玻璃引脚的连接,以及一些驱动控制器封装好的特性等)。无论如何,各种不同的 LCD 模块仍然可以总结出一些应用上的共性,这里以 MzL02 - 12864 LCD 模块为对象进行介绍,并不代表本书仅适用于该 LCD 模块,其他厂家生产的不同型号 LCD 模块也可以套用本书的介绍去理解、掌握 LCD 驱动程序的编程方法。

#### 1.1.1 LCD 模块的结构

通常见到的 LCD 模块分为几部分:LCM(玻璃)、背光、PCB 板;而背光和 PCB 板部分其实是可有可无的,视具体的 LCD 模块而定。点阵的 LCD 模块按照驱动控制器的集成方式,可分为两种:COB 和 COG。COG 是将驱动控制器芯片集成到玻璃上,而模块背后的 PCB 板上只是一些驱动控制器芯片无法集成的电容电阻而已;COB 的 LCD 模块是将驱动控制器焊接在 LCD 模块后面的 PCB 板上。

MzL02 - 12864(后面简称 MzL02 模块)为一块  $128 \times 64$  点阵的单色 LCD 显示模块,模块上的 LCM 采用 COG 技术将控制(包括显存)、驱动器集成在 LCM 的玻璃上,接口简单、操作方便;为方便用户的使用,在 LCM 的基础上设计了 MzL02 模块,将模块所必需的外围电容电阻集成到模块上,并引出多种形式的引线接口以方便用户使用。MzL02 模块与各种 MCU 均

可进行方便简单的接口操作。

MzL02 LCD 模块的特性如下：

- ▶ 128×64 点阵 FSTN；
- ▶ 1/64 占空比,1/9 偏压比；
- ▶ 单电源供电对比度编程可调；
- ▶ 并行接口为 6800 时序 MPU 接口方式；
- ▶ 3.3 V 的白色 LED 背光,美观大方；
- ▶ 集成 S6B0724 驱动控制 IC。

## 1.1.2 MzL02 结构尺寸

图 1-1 为 MzL02 模块中 LCM 的结构尺寸示意图。MzL02 模块的正面示意图如图 1-2 所示,接口引脚说明如表 1-1 所列。

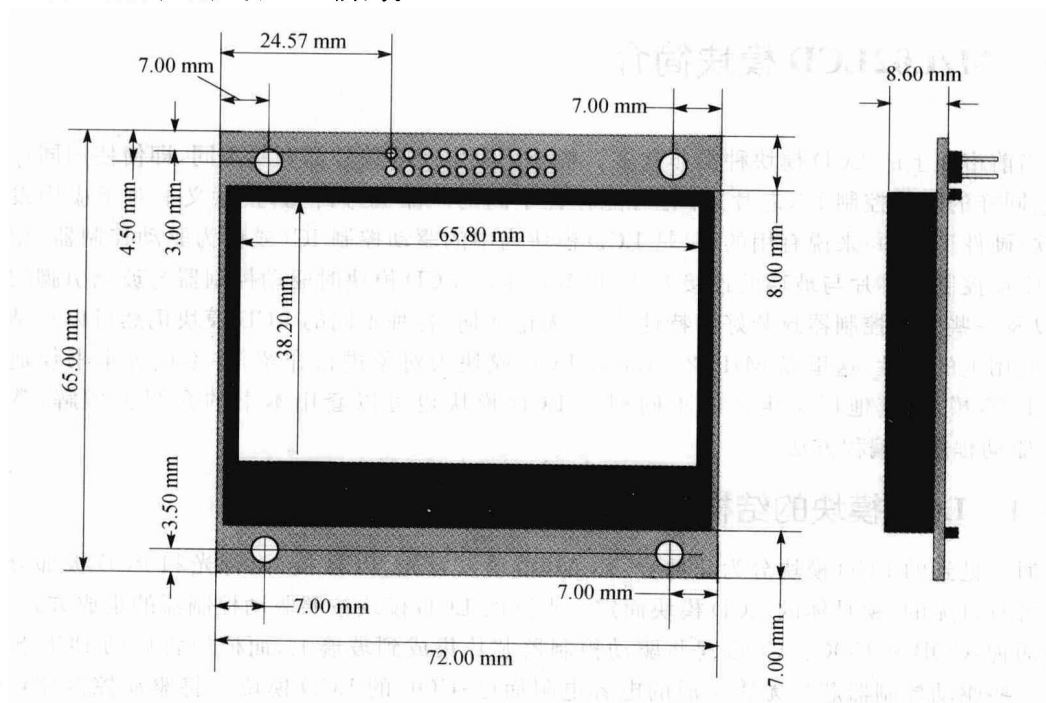


图 1-1 MzL02 模块 LCM 结构尺寸

LEDK	NC	C86	PS	EP	WR	A0	Reset	CS	LEDA
GND	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	+3.3 V

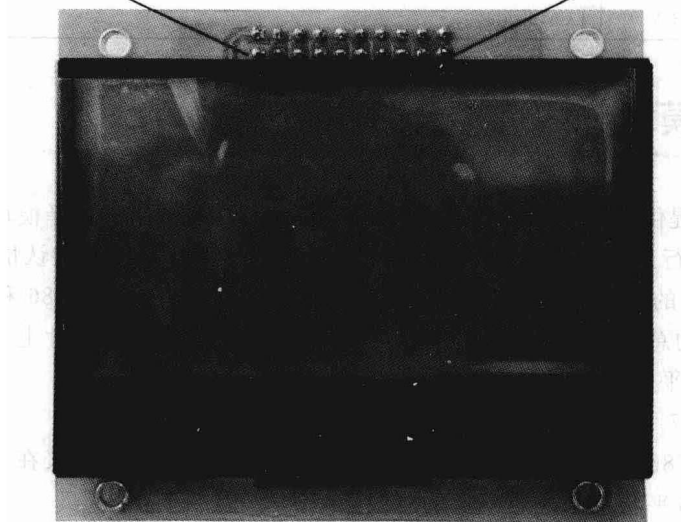


图 1-2 MzL02 模块正面示意图

表 1-1 MzL02 模块接口引脚说明

引 脚	接口引脚名	说 明
1	LEDK	背光负极输入
2	NC	空
3	C86	并行接口选择 高电平:6800 总线;低电平:8080 总线
4	PS	串行/并行接口选择 高电平:并行总线接口;低电平:串行接口
5	EP	6800 系列 MPU 的时钟信号使能脚(EP)。当选择总线接口为 8080 时,该线为 RD 信号
6	WR	6800 系列 MPU 的读/写控制信号(R/W)。8080 系列 MPU 总线的写使能信号
7	A0	数据命令选择脚(一些资料称其为 RS、C/D 等)
8	Reset(RST)	复位脚(低电平复位)
9	CS	片选(低电平有效)

引脚	接口引脚名	说明
10	LEDA	背光正极输入
11	GND	地
12~19	DB7~DB0	8 位数据总线
20	+3.3 V	LCM 供电

## 1.2 LCD 模块总线选择及时序图

MzL02 模块提供了两个 LCM 的引脚(C86 和 PS 引脚)供用户选择模块的接口类型,分别可以选择 6800 并行总线、8080 并行总线或者 SPI 串行接口。模块在默认情况下(即连接 LCD 模块时 C86 和 PS 的引针悬空时),由模块背后 PCB 中的上拉电阻将 C86 和 PS 脚上拉至高电平,此时为 6800 的总线接口,如图 1-3 所示。用户也可以通过给引针上 C86、PS 不同的电平来选择 LCD 模块的总线接口类型。

图 1-3 中,R7 为 C86 的上拉电阻,R9 为 PS 的上拉电阻。

如果用户在 C86 和 PS 引针上连接有固定的电平,则模块根据连接在这两个引脚上的电平状态来决定模块的总线接口类型。

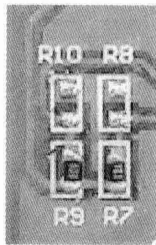


图 1-3 总线模式  
预选电阻

### 1.2.1 6800 并行总线接口

当 PS 为高电平、C86 为高电平时,LCD 模块的总线接口为 6800 并行总线接口。LCD 的时序如图 1-4 所示,参数见表 1-2。

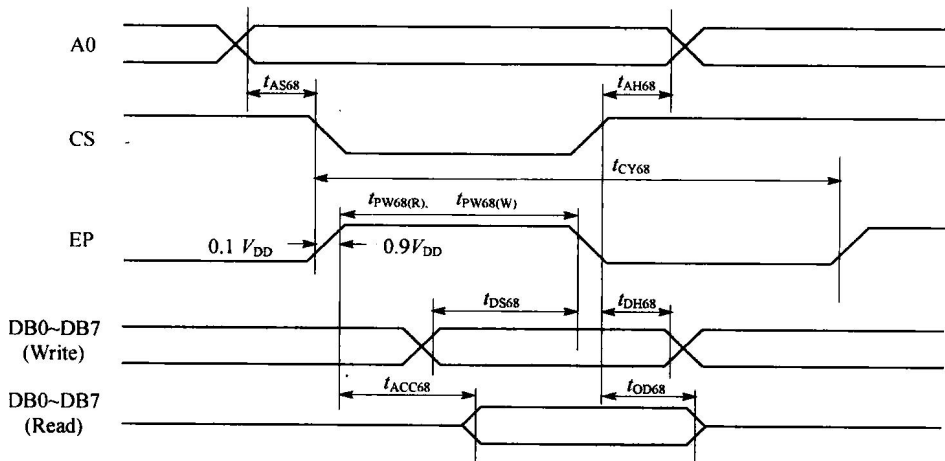


图 1-4 6800 时序示意图

表 1-2 时序参数 ( $V_{DD}=2.4\sim 3.6V$ )

项目	信号	标识	最小	典型	最大	单位	备注
地址建立时间	RS	$t_{AS68}$	0	—	—	ns	
地址保持时间		$t_{AH68}$	0	—	—	ns	
读写周期	RS	$t_{CY68}$	300	—	—	ns	
数据建立时间	DB7~DB0	$t_{DS68}$	40	—	—	ns	
数据保持时间		$t_{DH68}$	15	—	—	ns	
总线时间		$t_{ACC68}$	—	—	140	ns	$C_L=100\text{ pF}$
输出禁止时间		$t_{OD68}$	10	—	100	ns	
使能脉宽	读写	$t_{PW68(R)}$	120	—	—	ns	
		$t_{PW68(W)}$	60	—	—	ns	

6800 总线接口的模式下,WR 线为读写控制信号线,当 WR 为高电平时为读状态,当 WR 为低电平时为写入状态。

## 1.2.2 8080 并行总线接口

当 PS 为高电平、C86 为低电平时,LCD 模块的总线接口为 8080 并行总线接口。LCD 的时序如图 1-5 所示。

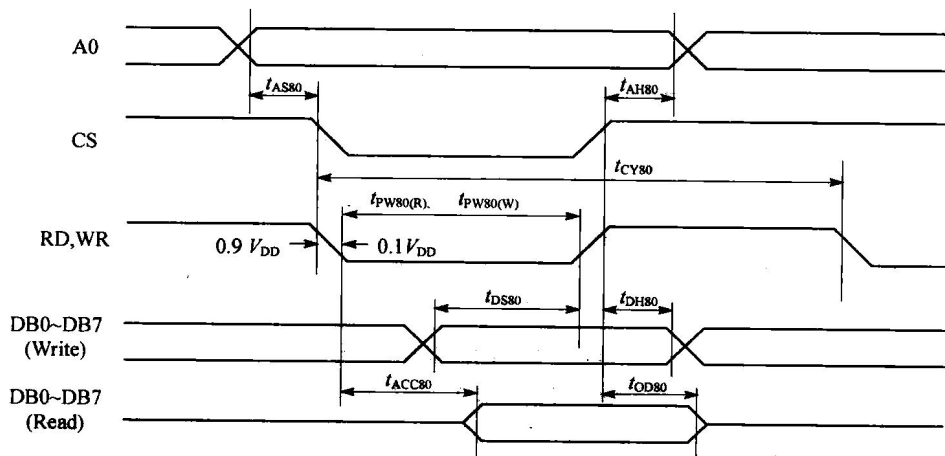


图 1-5 8080 时序示意图

LCD 模块处于 8080 总线模式下时,原 LCD 模块上的 EP 引脚将作为 8080 总线的 RD(读使能信号线)。8080 时序的参数如表 1-3 所列。

表 1-3 时序参数 ( $V_{DD}=2.4\sim 3.6\text{ V}$ )

项目	信号	标识	最小	典型	最大	单位	备注
地址建立时间	RS	$t_{AS80}$	0	—	—	ns	
地址保持时间		$t_{AH80}$	0	—	—		
读写周期	RS	$t_{CY80}$	300	—	—	ns	
读周期	E_RDB	$t_{PW80(R)}$	60	—	—	ns	
写周期	RW_WRB	$t_{PW80(W)}$	60	—	—	ns	
数据建立时间	DB7~DB0	$t_{DS80}$	40	—	—	ns	
数据保持时间		$t_{DH80}$	15	—	—		
总线时间		$t_{ACC80}$	—	—	140	ns	
输出禁止时间	$t_{OD80}$	10	—	100			

## 1.2.3 SPI 串行接口

当 PS 为低电平时, LCD 模块的总线接口为 SPI 串行总线接口; 而当模块处于 SPI 串行接口模式下时, LCD 模块控制器不提供读操作的功能, 仅为写入。而 SPI 模式下, 原并行总线的数据线 DB6 将作为时钟输入线 (SCLK), 而 DB7 将作为数据输入线 (SID)。

LCD 的时序如图 1-6 所示, 其时序如表 1-4 所列。

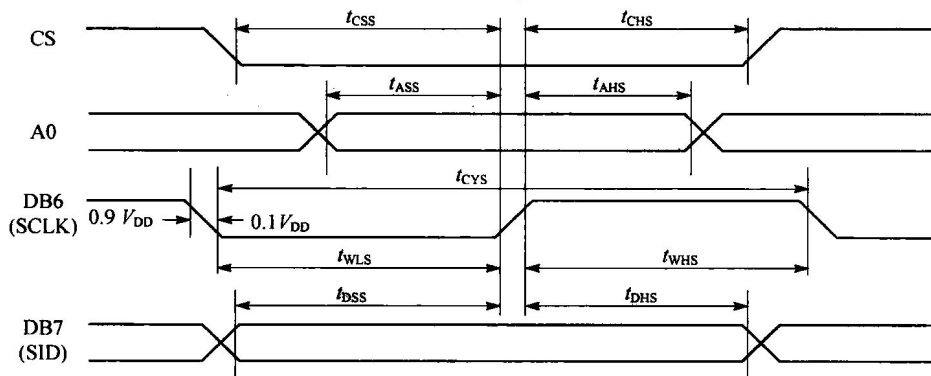


图 1-6 串行时序示意图

表 1-4 时序参数 ( $V_{DD}=2.4\sim 3.6\text{ V}$ )

项目	信号	标识	最小	典型	最大	单位
时钟周期	DB6 (SCLK)	$t_{CYS}$	250	—	—	ns
时钟高电平持续时间		$t_{WHS}$	100	—	—	
时钟低电平持续时间		$t_{WLS}$	100	—	—	



续表 1-4

项目	信号	标识	最小	典型	最大	单位
地址建立时间	RS	$t_{ASS}$	150	—	—	ns
地址保持时间		$t_{AHS}$	150	—	—	
数据建立时间	DB7 (SID)	$t_{DSS}$	100	—	—	ns
数据保持时间		$t_{DHS}$	100	—	—	
片选建立时间	CS	$t_{CSS}$	150	—	—	ns
片选保持时间		$t_{CHS}$	150	—	—	

## 1.3 显存映射情况

### 1.3.1 显存与 LCD 屏上点的关系

对于 LCD 模块,了解清楚驱动控制 IC 当中的显存与 LCD 玻璃上点的对应关系非常重要,是编写 LCD 驱动程序的基础。

MzL02 模块显示器(玻璃)上的显示点与驱动控制芯片中的显示缓存 RAM 是一一对应的。驱动控制芯片当中共有  $65(8 \text{ Page} \times 8 \text{ bit} + 1) \times 132$  个位的显示 RAM 区,而显示器的显示点阵大小为  $64 \times 128$  点,所以实际上在液晶显示模块中有用的显示 RAM 区为  $64 \times 128$  个位;按字节为单位划分,共分为 8 个 Page,每个 Page 为 8 行,而每一行为 128 位(即 128 列)。

驱动控制器芯片的显示 RAM 区每个字节的数据对应屏上点的排列方式为:纵向排列,低位在上高位在下,如图 1-7 所示。

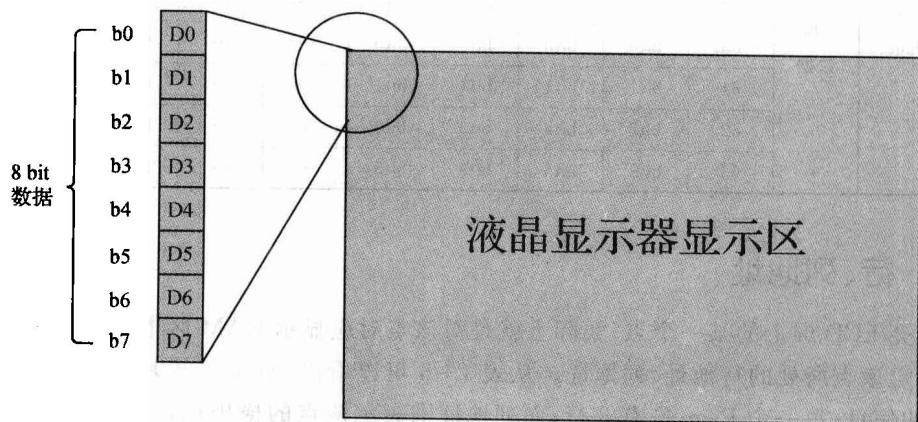


图 1-7 字节数据排列情况