

目 录

SC165C010 简介	2
SC165C010 的特点	2
SC165C010 的封装形式	3
SC165C010 的引脚说明	4
SC165C010 的内部结构框图	5
SC165C010 的典型应用框图	5
晶振、帧刷新频率和复位	6
极限参数	6
直流电气特性	6
交流电气特性	7
输出引脚复位状态	12
外围器件选用指南	13
液晶模块的硬件接口规范	14
应用原理图举例	15
软件接口方式	16
命令代码表	17
软硬件接口示例	20
封装外形尺寸	28

SC165C010 简介

SC165C010 是一种通用的单色图形点阵液晶显示控制器，单屏最大可支持 320×240 的点阵规模。由于显示数据随机存储器（RAM）和显示驱动器（Driver）均为外接，它可以应用于各种标准点阵规模的液晶显示模块。

SC165C010 支持的数据空间多达 448K 字节，分 7 页，每页 64K 字节。它可以有选择地外接 E^2 PROM 和中英文全字库 ROM。外接 E^2 PROM 将使其具有数据预置功能，丰富的显示数据便可通过页面编辑器（PgEditor.exe，可视化的计算机辅助设计软件）

设计产生并下载到该 E^2 PROM 中。

SC165C010 具有强大的显示功能和丰富的指令集，微处理器只需向它发送很少的参数便可实现各种复杂的显示效果。因此，SC165C010 不仅保留了传统的 INTEL8080 并行接口，还特别增加了串行接口方式，使其对微处理器的要求进一步降低。

SC165C010 由杭州士兰微电子股份有限公司独立设计，用户可免费享受计算机辅助设计软件带来的高效率、高可维护性和广阔的设计思维空间。

SC165C010 的特点

- ◆ 单屏最大可支持 320×240 的点阵规模，适用于从 128×32 、 64×64 到 320×240 各种标准点阵规模的液晶模块。
- ◆ 显示占空比：1/32 到 1/256。
- ◆ 支持 64K 字节的静态 RAM。
- ◆ 支持 64K 字节的显示数据预置存储器 E^2 PROM，可通过可视化的辅助开发软件下载预置数据。
- ◆ 最多可支持 16 种自定义字库。
- ◆ 支持中英文全字库 ROM。
- ◆ 与微处理器的接口可选用并行或串行方式，使对微处理器的要求降至最低。
- ◆ 固定外接 10MHz 的晶振。
- ◆ 支持 [T6A39, T6A40]、[HD66204, HD66205] 等驱动芯片组。
- ◆ 具有字符、字符串和位图显示功能及填充功能。
- ◆ 具有灵活的页面操作功能。
- ◆ 多种逻辑显示方式：普通、复制、与、或、非、异或。
- ◆ 支持点阵反显、下划线。
- ◆ 内置 256 种填充模板/过滤器。
- ◆ 具有绘制点、水平线、垂直线、矩形框、带阴影矩形框和任意两点间线段的功能。
- ◆ 支持自定义线型。
- ◆ 任意宽度和高度（最大为 24×24 ）的光标显示。
- ◆ 引入活动区域和可视活动区域的概念，超出可视活动区域的显示内容被自动裁减。
- ◆ 支持安全的内存操作，可在 RAM 和 E^2 PROM 中开辟一块区域作为一般数据存取区用，这在一些场合下可以减少外部存储器件。
- ◆ 支持复杂的显示对象，功能更强大，操作更方便。
- ◆ 支持点阵显示信息外灌，保持可扩展性。
- ◆ 封装形式：QFP64-14 \times 14-0.8。

SC165C010 的封装形式

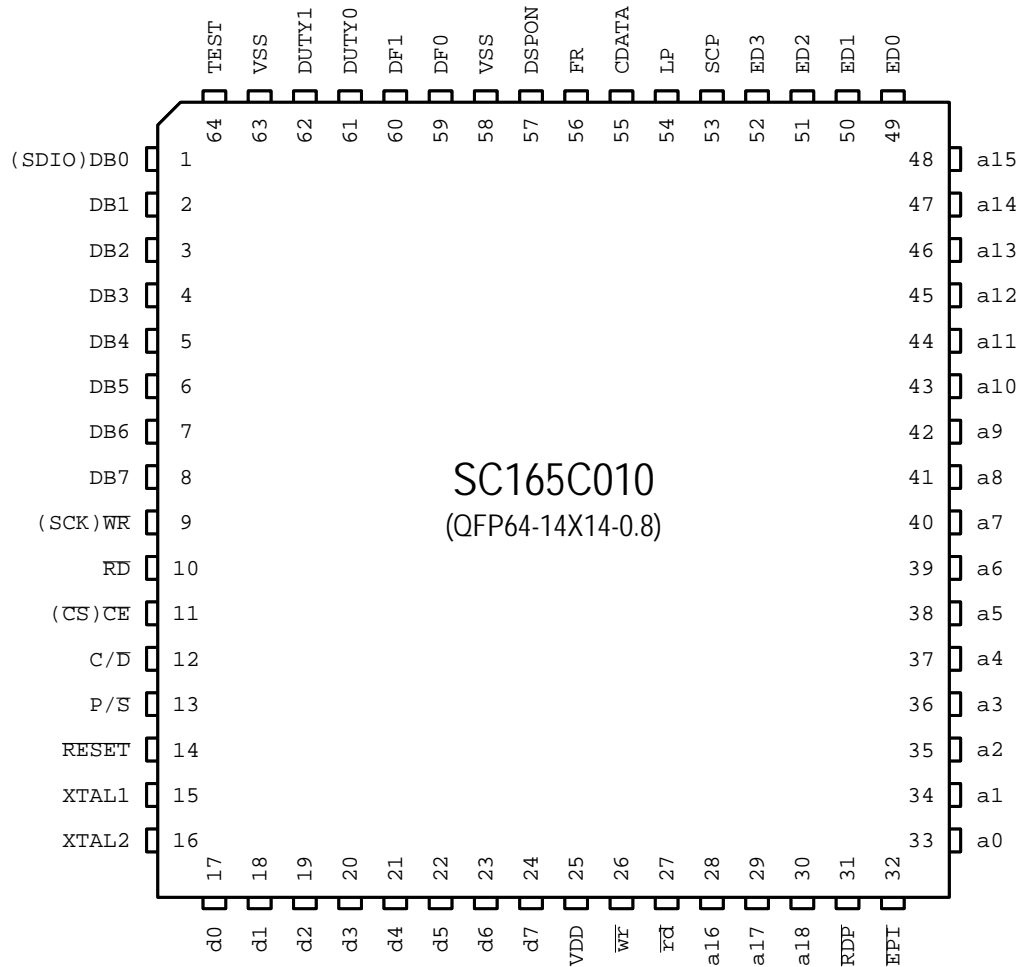


图 1 - QFP64-14×14-0.8 封装的引脚分配

SC165C010 的引脚说明

引脚名称	类型	编号	引脚功能
系统部分			
RESET	I	14	芯片复位引脚（低电平有效）
TEST	I	64	芯片测试引脚（正常使用请接低电平）
EPI	I	32	芯片测试引脚（正常使用请接高电平）
RDP	O	31	芯片测试引脚（正常使用请悬空）
VDD	PWR	25	电源正端
VSS	PWR	58, 63	电源负端
XTAL1	I	15	接晶振端 1 或外部系统时钟输入（典型频率为 10MHz）
XTAL2	O	16	接晶振端 2
MCU 接口部分			
CE (CS)	I	11	片选（低电平有效）
RD	I	10	读选通信号（低电平有效）
WR (SCK)	I	9	P/S = 1: 并行方式的写控制信号（上升沿写入） P/S = 0: 串行方式的时钟信号
C/D	I	12	操作通道选择（1: 命令通道； 0: 数据通道）
DB[0..7] (DB[0]=SDIO)	I/O	1..8	数据输入/输出端口
P/S	I	13	接口方式选择（1: 并行方式； 0: 串行方式）
存储器接口部分			
rd	O	27	外部数据存储器读控制信号
wr	O	26	外部数据存储器写控制信号
a[0..15]	O	33..48	存储器地址选择信号
a[16..18]	O	28..30	数据存储器页选择地址
d[0..7]	I/O	17..24	存储器数据输入/输出端口
液晶驱动控制部分			
CDATA	O	55	帧起始信号，即行扫描数据信号
DF[1..0]	I	60..59	显示数据输出格式（00: 1 位； 01: 2 位； 1x: 4 位）
DSPON	O	57	液晶显示驱动电源控制端（1: 显示； 0: 关闭）
DUTY[1..0]	I	62..61	显示占空比选择 (00: 1/32; 01: 1/64; 10: 1/128; 11: 1/256)
ED[0..3]	O	49..52	显示数据输出端
FR	O	56	液晶显示交流驱动波形信号
LP	O	54	锁存脉冲信号，用于列显示数据锁存信号和行扫描数据的移位信号
SCP	O	53	显示数据的移位脉冲信号

表 1 - 引脚说明

SC165C010 的内部结构框图

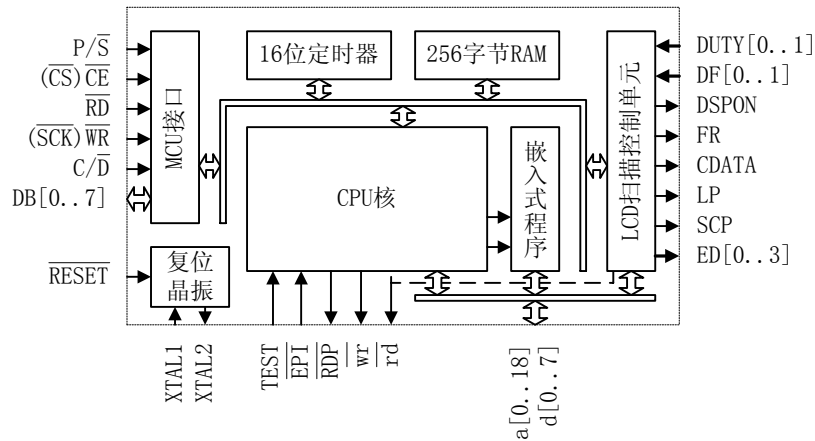


图 2 - 内部结构框图

SC165C010 的典型应用框图

SC165C010 在标准图形点阵液晶显示模块中的典型应用如图 3所示。其中：显示数据随机存储器（RAM）和显示数据预置存储器（可以是E²PROM、EPROM或ROM）的最大容量可达 64K字节，中英文全

字库ROM的字库存储格式为士兰公司定义，购买士兰公司的专用字库芯片（SC165LXXX系列）会比较经济。更详细的建议请看下文的“**外围器件选用指南**”部分。

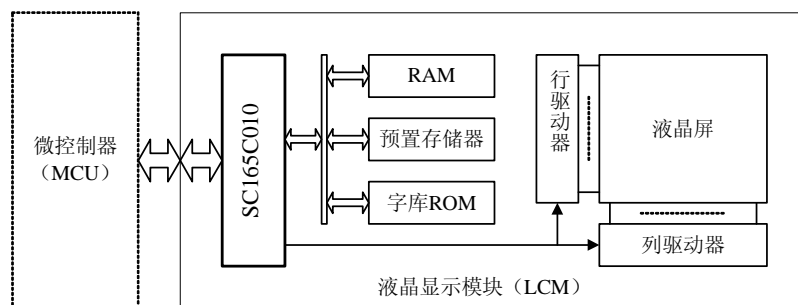


图 3 - SC165C010 的典型应用框图

晶振、帧刷新频率和复位

无论显示占空比设置成多少，SC165C010 均固定外接 10MHz 的晶振。在这样的晶振频率下，SC165C010 控制液晶屏的帧刷新频率为 61Hz。

复位引脚拉成低电平状态达 25ms 以上，而后将其拉高。上电完毕后，若还需要复位，则只需将复位引脚拉低 800ns 以上再拉高即可。

芯片上电时，为了保证晶振的可靠起振，应将

极限参数

参数	范围	单位
工作时的环境温度：T _A	0 ~ +70	℃
芯片放置温度：T _P	-65 ~ +125	℃
电源对地电压：VDD-VSS	-0.3 ~ +7	V
直流输入电压：V _I	VSS-0.3 ~ VDD+0.3	V

表 2 - 极限参数

直流电气特性

(T_A = 0 ~ 70℃, VDD = 2.7V ~ 5.5V, VSS = 0V)

符号	说明	测试条件	最小值	最大值	单位
V _{IH}	输入高电平电压要求		0.8VDD	VDD	V
V _{IL}	输入低电平电压要求		0	0.2VDD	V
V _{OH}	输出高电平电压	I _{OH} = -500 μA	VDD-0.1	-	V
V _{OL}	输出低电平电压	I _{OL} = 3.0mA	-	0.4	V
I _{LI}	输入漏电流绝对值	VSS ≤ V _I ≤ VDD	0	5	μA
I _{LO}	输出漏电流绝对值	VSS ≤ V _I ≤ VDD	0	5	μA
I _{DD}	工作电流	F _{osc} ⁽¹⁾ = 10MHz; VDD = 5V	6	9	mA

表 3 - 直流电气参数

表 3注：
1. F_{osc}为外接晶振频率。

交流电气特性

($T_A = 0 \sim 70^\circ\text{C}$, $V_{DD} = 2.7\text{V} \sim 5.5\text{V}$, $V_{SS} = 0\text{V}$, $F_{osc} = 10\text{MHz}$)

MCU 接口时序

- 并行接口时序 (P/S 固定接高电平)

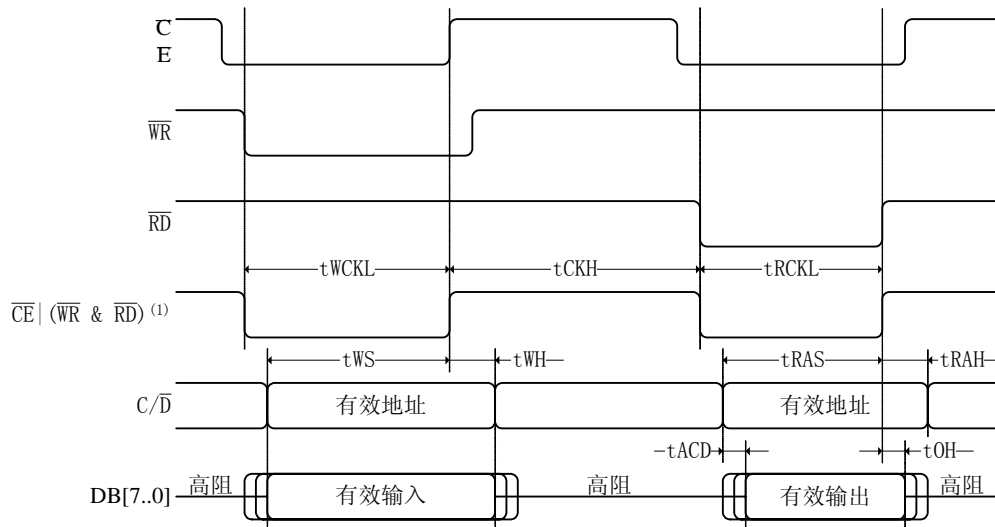


图 4 - MCU 并行接口读写操作时序⁽²⁾

图 4注:

1. $\overline{CE} | (\overline{WR} \& \overline{RD})$ 为**虚拟组合信号**，它是由 \overline{WR} 和 \overline{RD} 相与后再和 \overline{CE} 相或的结果；无论这三个信号的波形如何，我们只关心虚拟组合信号的高低电平宽度。**虚拟信号为低电平有效**。
2. 一般情况下，通讯速率不高的 INTEL8080 读写时序（如标准 80C51 工作在 12M 晶振下）均能够满足以上时序要求。

符号	说明	测试条件	最小值	最大值	单位
tWCKL	写操作时“虚拟组合信号”低电平宽度		120	-	ns
tCKH	“虚拟组合信号”高电平宽度		320	-	ns
tRCKL	读操作时“虚拟组合信号”低电平宽度		120	-	ns
tWS	写操作时，有效地址和有效输入数据的建立时间		90	-	ns
tWH	写操作时，有效地址和有效输入数据的保持时间		5	-	ns
tRAS	读操作时，有效地址的建立时间		90	-	ns
tRAH	读操作时，有效地址的保持时间		5	-	ns
tACD	读操作时，“虚拟组合信号”和地址均有效后，输出有效数据所需的时间		-	35	ns
tOH	读操作时，“虚拟组合信号”无效后，有效输出数据的保持时间		5	-	ns

表 4 - MCU 并行接口读写操作时序参数

□ 串行接口时序 (P/ \overline{S} 固定接低电平, \overline{RD} 固定接高电平)

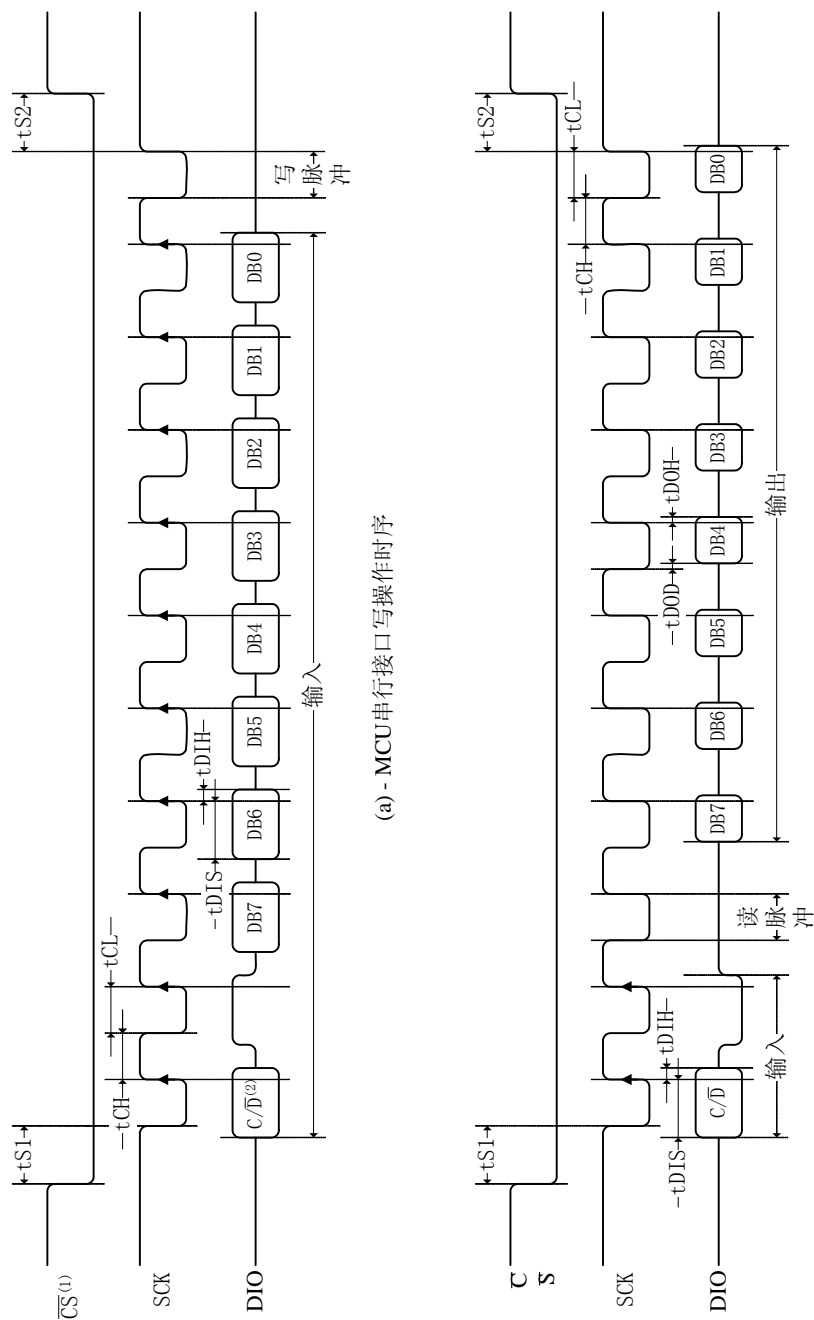


图 5

图 5注:

1. \overline{CS} 的低电平必须完全包含 SCK 的第一个下降沿到最后一个 (第十一个) 上升沿, 建议在每次读写操作之后, 都要把 \overline{CS} 拉高并保持 120ns 以上, 确保串口可靠复位。
2. SCK 的第一个时钟上升沿将会锁存操作通道, C/\overline{D} 为高电平表示接下来操作的是命令通道, 若为低电平, 则表示接下来操作的是数据通道。

符号	说明	测试条件	最小值	最大值	单位
tS1	\overline{CS} 的下降沿应超前 SCK 第一个下降沿的时间		220	—	ns
tS2	\overline{CS} 的上升沿应滞后 SCK 最后一个（第十一个）上升沿的时间		220	—	ns
tCH	SCK 的高电平宽度		320	—	ns
tCL	SCK 的低电平宽度		220	—	ns
tDIS	数据从 DIO 输入时，应在相应 SCK 上升沿之前有效建立的时间		90	—	ns
tDIH	数据从 DIO 输入时，应在相应 SCK 上升沿之后有效保持的时间		5	—	ns
tDOD	从 SCK 的下降沿到数据从 DIO 有效输出的延时		—	120	ns
tDOH	从 SCK 的上升沿到数据有效保持的时间		5	—	ns

表 5 - MCU 串行接口读写操作时序参数

◆ 存储器接口时序

▫ 内嵌 MCU 核访问外部存储器时序

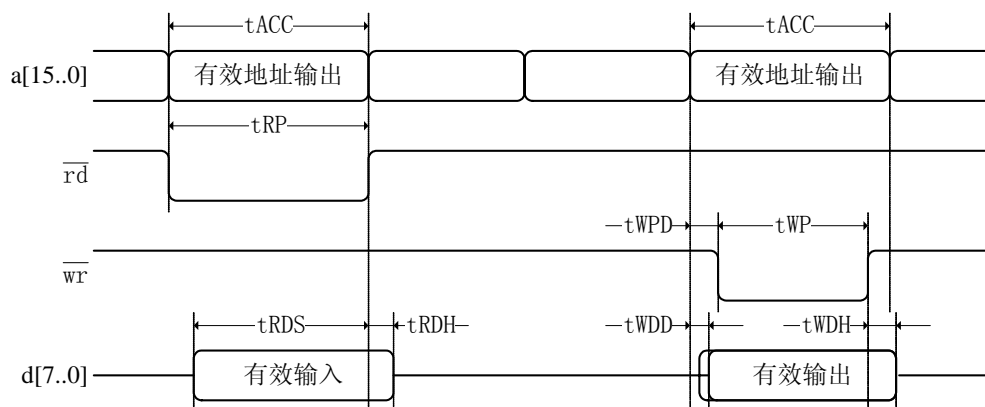


图 6 - 内嵌 MCU 核访问外部存储器时序

符号	说明	测试条件	最小值	典型值	最大值	单位
tACC	低 16 位有效地址 $a[15..0]$ 输出宽度		190	195	200	ns
tRP	读控制信号 \overline{rd} 输出低电平宽度		190	195	200	ns
tRDS			55	—	—	ns
tRDH			0	—	—	ns
tWPD			24	27	30	ns
tWP			145	148	150	ns
tWDD			16	18	20	ns
tWDH			20	23	26	ns

表 6 - 内嵌 MCU 核访问外部存储器时序参数

□ LCD 扫描控制单元读外部 RAM 时序

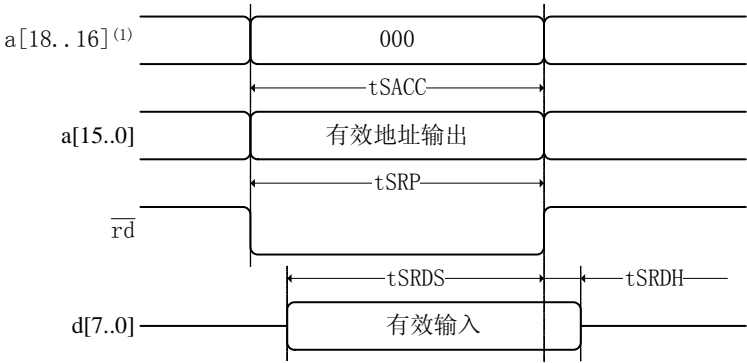


图 7 - LCD 扫描控制单元读外部 RAM 时序

图 7注:

1. LCD 扫描控制单元在访问外部 RAM 时，一定会把 $a[18..16]$ 清 0，待读操作完成后再恢复成原来的值。

符号	说明	测试条件	最小值	典型值	最大值	单位
tSACC			93	95	100	ns
tSRP			93	95	100	ns
tSRDS			55	—	—	ns
tSRDH			0	—	—	ns

表 7 - LCD 扫描控制单元读外部 RAM 时序参数

◆ 液晶驱动控制时序

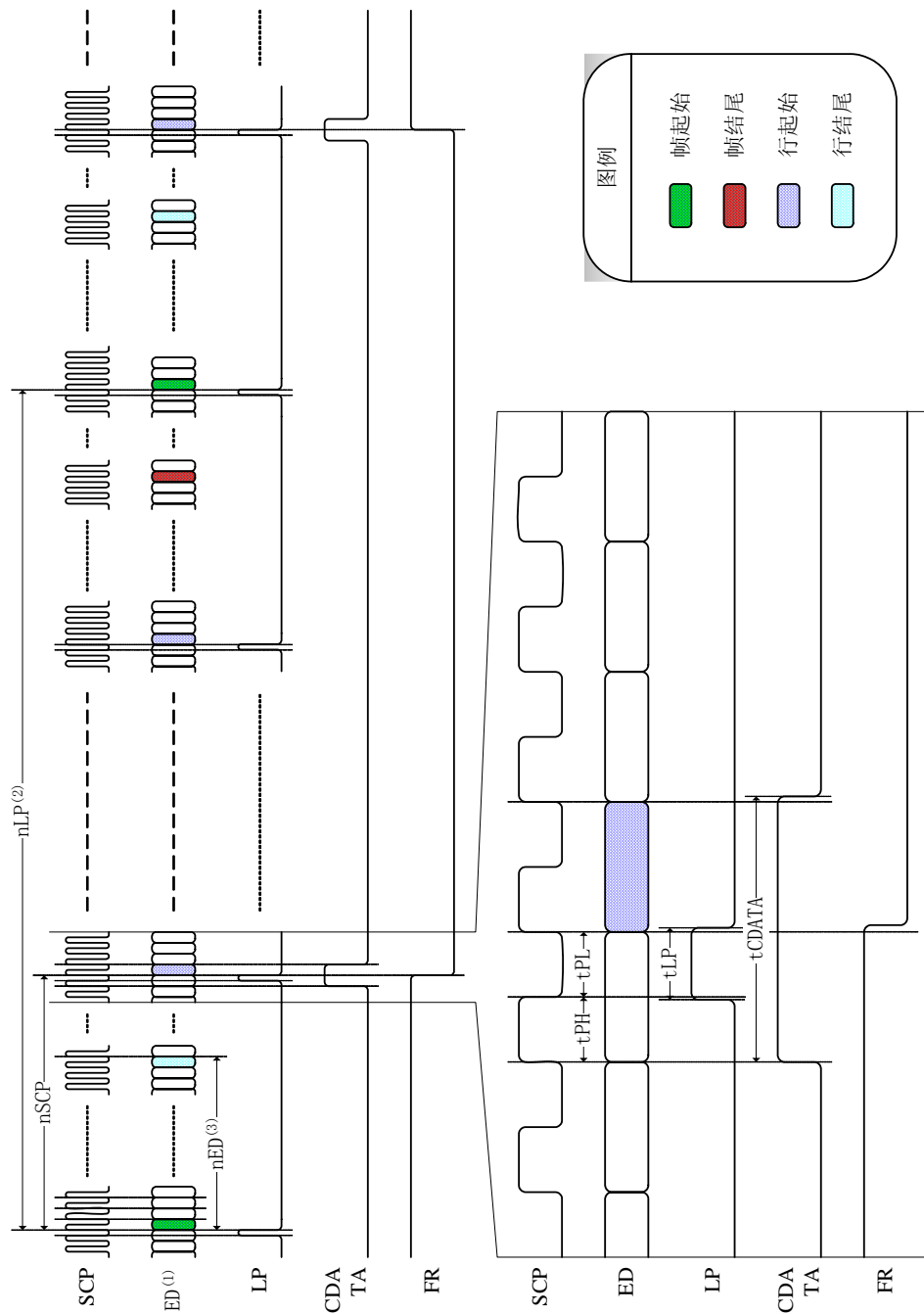


图 8 - 液晶驱动控制时序

图 8注:

1. 根据DF[1..0]的配置情况, ED有可能是 1 位、2 位或 4 位, 具体关系见表 8; 当 $nED < nSCP$ 时, ED在送出行结尾数据后就一直保持 0 值直至下一行开始。
2. nLP 是扫描一帧的LP 脉冲数, 也即一帧的行数, 它由DUTY[1..0]的配置情况决定。
3. nED 是一行内发送有效显示数据的SCP 脉冲数, 它可能会小于一行内SCP 脉冲的总数 $nSCP$ 。

DF[1..0]	nSCP	ED 的有效位	其它说明
00	320	ED[0]	ED[3..1] = 000；显存内每个字节的数据将按 [7], [6], [5], [4], [3], [2], [1], [0] 的次序从 ED[0] 送出
01	160	ED[1..0]	ED[3..2] = 00；显存内每个字节的数据将按 [7..6], [5..4], [3..2], [1..0] 的次序从 ED[1..0] 送出
10, 11	80	ED[3..0]	显存内每个字节的数据将按 [7..4], [3..0] 的次序从 ED[3..0] 送出

表 8 - DF[1..0] 的配置情况对 nSCP 和 ED 的影响

DUTY[1..0]	nLP	显示占空比
00	32	1/32
01	64	1/64
10	128	1/128
11	256	1/256

表 9 - DUTY[1..0] 的配置情况对 nLP 的影响

符号	说明	测试条件	典型值	单位
tPH	SCP 高电平宽度		$\frac{100 \times 256 \times 320}{nLP \times nSCP}$	ns
tPL	SCP 低电平宽度		$\frac{100 \times 256 \times 320}{nLP \times nSCP}$	ns
tLP	LP 脉宽		$\frac{100 \times 256 \times 320}{nLP \times nSCP}$	ns
tCDATA	CDATA 脉宽		$\frac{400 \times 256 \times 320}{nLP \times nSCP}$	ns

表 10 - 液晶驱动控制时序参数

输出引脚复位状态

引脚名称	复位状态	引脚名称	复位状态
\overline{rd}	“1”	ED[3..0]	“0000”
\overline{wr}	“1”	LP	“0”
DSPON	“0”	CDATA	“0”
SCP	“1”	FR	“1”

表 11 - 输出引脚复位状态

外围器件选用指南

从图 3 可以看出，在实际应用中，SC165C010 还需外接 RAM（必须是静态 RAM）、显示数据预置存储器、中英文全字库 ROM 和行列驱动器。其中，RAM 和行列驱动器是必不可少的，而显示数据预置存储器和中英文全字库 ROM 则可以根据需要进行取舍。

一般情况下，如果应用系统需要液晶显示较复杂的界面且希望降低开发成本，则强烈建议选用显示数据预置存储器。当应用系统中需要液晶显示大量的文字，而预置存储器容量比较紧张或干脆放不

下这么多字符点阵信息时，增加中英文全字库 ROM 可以有效降低成本。另外，如果在某种应用场合下液晶的显示内容以文字为主，则取字库 ROM 而舍预置存储器的方案也是可以的。

显示数据预置存储器可以选用 E²PROM、EPROM 或 ROM，选择的依据有：功能、用量、成本等。

在表 12、表 13 中，读者会对预置存储器和字库 ROM 的选择有一个更清晰的认识。

显示数据预置存储器	中英文全字库 ROM	应用场合
×	×	所有的点阵数据从外部送入 SC165C010，显示内容简单，对 MCU 的要求较高。
✓	×	显示数据可以预置，字符数有限，对 MCU 要求很低。
✓	✓	显示数据可以预置，字符数多，对 MCU 要求很低。
×	✓	显示内容以文字为主，其它点阵信息需要从外部送入，对 MCU 要求不高。

表 12 - 关于预置存储器和字库 ROM 的取舍

显示数据预置存储器	应用场合及特点
并行 E ² PROM	能够通过液晶仿真器直接下载预置数据； 能够将富余空间开辟给系统 MCU 自由存取； 成本较高，一般用于开发阶段。
EPROM	需要通过其它通用编程器烧录预置数据，成本不高，适用于小批量。
ROM	产品功能已经确定且量很大；掩膜成本高，须达到一定的量才能降低成本。

表 13 - 关于预置存储器类型的选择

◆ 静态随机存取存储器（RAM）的选择

SC165C010 的存储器接口时序为标准的 INTEL8080 接口时序，但 RAM 要被芯片内嵌 MCU 和 LCD 扫描控制单元访问，因此，它的速度等级要同时满

足表 6 和表 7 的要求。根据这两个条件，建议选用速度等级为 70 或更快的 62 系列 SRAM。

◆ 显示数据预置存储器的选择

若显示数据预置存储器为E²PROM，则选择 28 或 29 系列、容量不超过 64K 字节、速度等级不低于 150 的并行E²PROM。如：AT28C256-15、AT29C256-15、AT29C512-12 等。若想要预置存储器还用于充当MCU 的自由存取区，则只能选择 28 系列的E²PROM。

若显示数据预置存储器为 EPROM，则选择 27 系列的存储器，当然，容量也不要超过 64K 字节、速度等级不低于 150。如：CY27C256-120、W27C512-45 等。

◆ 中英文全字库 ROM 的选择

中英文全字库 ROM 的字库存储格式为士兰公司定义，字库的目标文件可以由可视化辅助开发工具——“页面编辑器”生成。因此，理论上用户可以自行制作字库 ROM，方法是：准备一颗 512K 字节容量的 EPROM，将“页面编辑器”生成的字库文件

(* .ROM) 烧录到该 EPROM 中，烧录起始地址为 20000H。但是，这样做的成本太高。

选用士兰的专用字库芯片（SC165LXXX 系列）可以降低成本、提高生产效率。

◆ 行列驱动器的选择

单纯的点阵液晶行列驱动器或驱动器组的数据接口一般都具有互相兼容的时序要求，如 [T6A39, T6A40]、[HD66204, HD66205]、[HD66206]、

[KS0086]、[KS0065]、[SED1180, SED1190]等，这些都符合 SC165C010 的接口要求。

液晶模块的硬件接口规范

若液晶模块中使用 SC165C010 作为显示控制器，则必须遵守本规范。

第一条：SC165C010 的 DSPON 引脚一定要能够控制驱动电源，并确保 SC165C010 的 DSPON 引脚为

低电平时液晶模块的显示能够被关闭。

第二条：液晶模块的对外接口，无论其封装形式如何，均应遵循表 14 的引脚定义。这是为了统一液晶模块和液晶仿真器的接口方式。

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
	GND	VCC		$\overline{\text{WR}}$	$\overline{\text{RD}}$	$\overline{\text{CS}}$	C/ $\overline{\text{D}}$	P/ $\overline{\text{S}}$	$\overline{\text{RESET}}$
P11	P12	P13	P14	P15	P16	P17	P18	P19	P20
DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7		

表 14 - 液晶模块对外接口引脚定义

表 14注：空白处为自定义引脚。

应用原理图举例

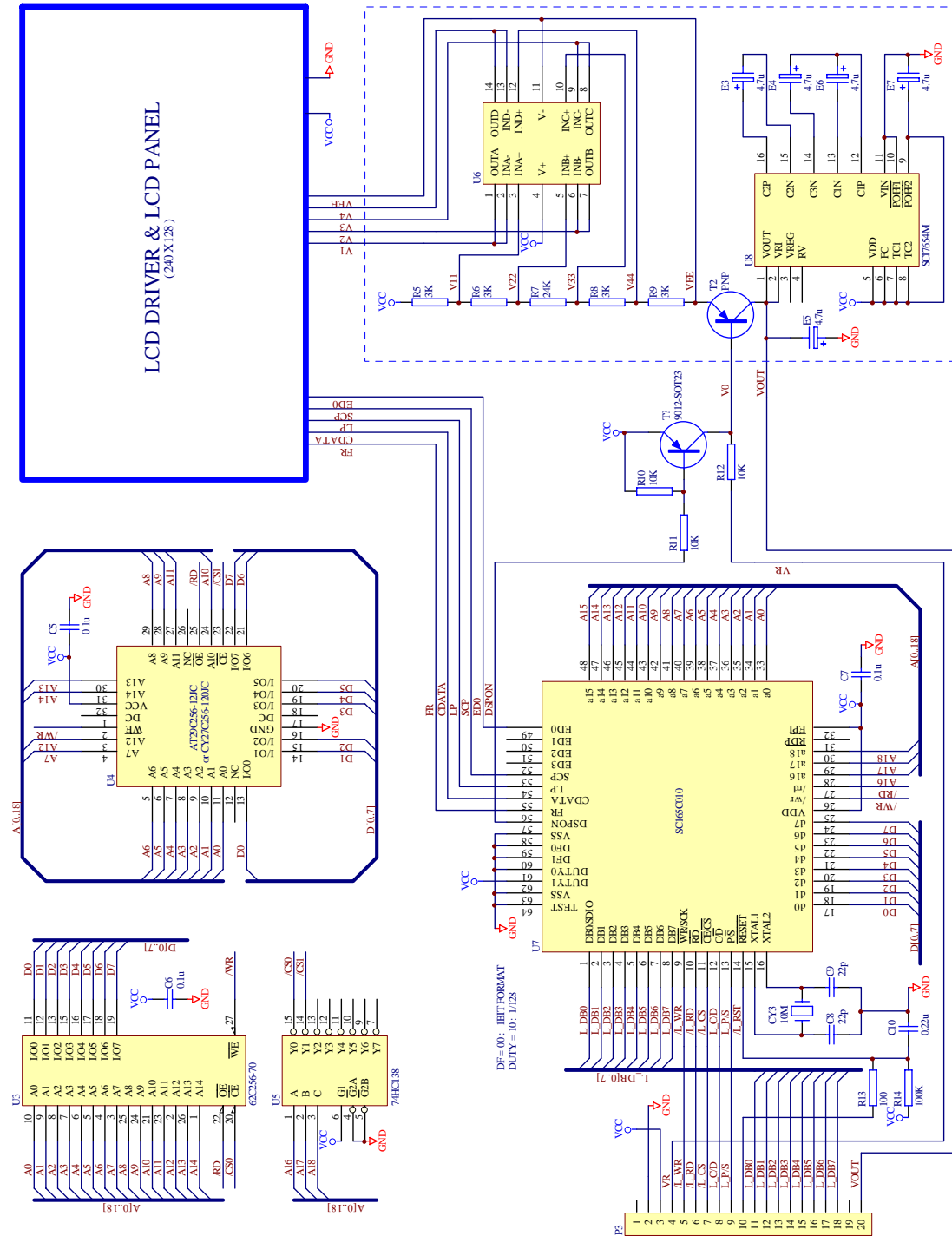


图 9 - SC165C010 应用原理图

软件接口方式

MCU 通过向 SC165C010 发送参数和命令达到控制液晶显示的目的。SC165C010 的命令多达 110 条，可实现从系统参数设置、基本显示到复杂的对象和页面显示等各个层面上的操作。每条命令的输入参数字节数从 0 到 16 不等，输出参数的字节数也从 0

到 16 不等。

MCU 向 SC165C010 发送一条命令的流程如图 10 所示，概括为：写输入参数→写命令→读输出参数。

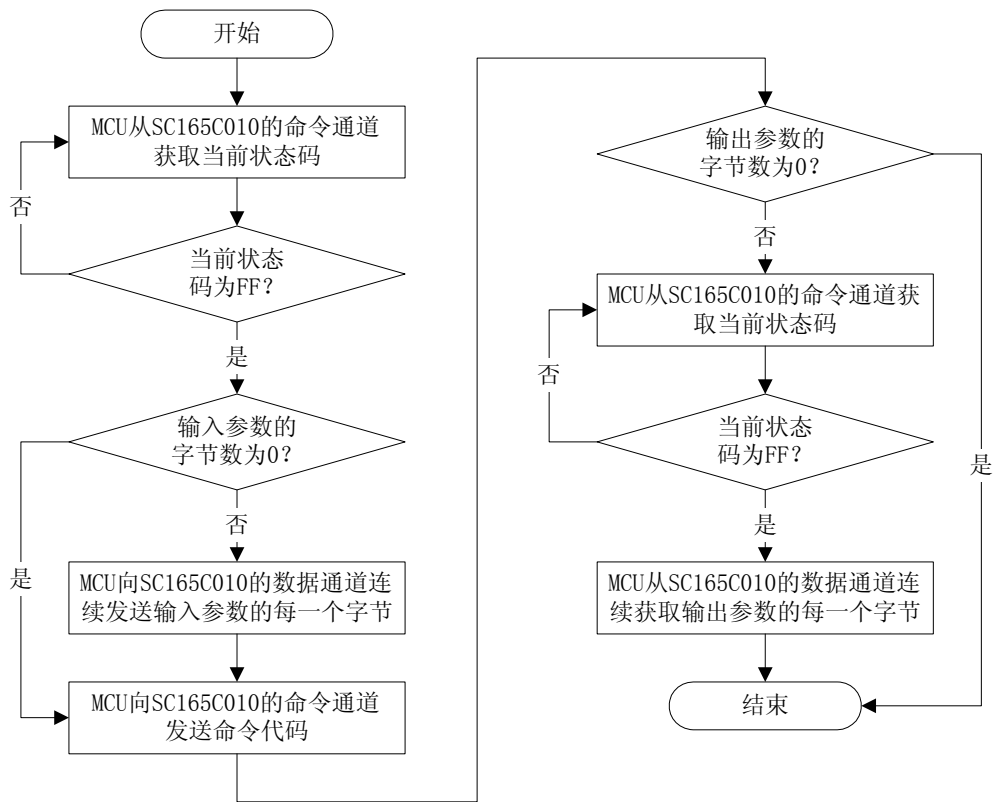


图 10 - 软件接口流程

命令代码表

功能名称	命令 代码	功能简介
F_SETBASIC	00H	设置液晶的基本参数
F_GETBASIC	01H	获取液晶基本参数的设置值
F_SETOSCFREQ	02H	设置外接晶振频率
F_GETOSCFREQ	03H	获取外接晶振频率的设置值
F_SETDISP	04H	设置液晶显示开关
F_GETLSW	05H	获取液晶状态字
F_SETPAGEINFO	08H	设置页面信息, 包括显示页号和操作页号
F_GETPAGEINFO	09H	获取页面信息, 包括显示页号和操作页号
F_SETACTRECT	0CH	设置活动区域
F_GETACTRECT	0DH	获取活动区域
F_SETACTFULL	0EH	设置整个页面区域为活动区域
F_SETFILTER	10H	设置过滤器编号
F_GETFILTER	11H	获取过滤器编号
F_SETINVERT	12H	设置反显方式
F_GETINVERT	13H	获取反显方式
F_SETUNDERLINE	14H	设置下划线方式
F_GETUNDERLINE	15H	获取下划线方式
F_SETCURSOR	18H	光标显示开关
F_SETCURGLIT	19H	光标闪烁开关
F_MOVECURSOR	1AH	移动光标
F_RESIZECURSOR	1BH	改变光标尺寸
F_MODISTRCUR	1CH	将字符串显示后产生的光标信息赋给光标
F_SETBDPTR	24H	设置通用显示信息缓冲区数据指针
F_GETBDPTR	25H	获取通用显示信息缓冲区的数据指针
F_WRBUFFA	28H	往通用显示信息缓冲区写一个数据, 数据指针循环增一
F_WRBUFFB	29H	往通用显示信息缓冲区写 N 个数据, 数据指针循环增加
F_WRBUFFC	2AH	往通用显示信息缓冲区写 16 个数据, 数据指针循环增加
F_WRBUFFD	2BH	往通用显示信息缓冲区写一个数据, 但不改变数据指针
F_RDBUFFB	2CH	从通用显示信息缓冲区读 N 个数据, 数据指针循环增加
F_RDBUFFD	2DH	从通用显示信息缓冲区读 N 个数据, 但不改变数据指针
F_SETFDPTR	34H	设置自由存取区数据指针
F_GETFDPTR	35H	获取自由存取区的数据指针
F_WRFREEA	38H	往自由存取区写一个数据, 数据指针循环增一
F_WRFREEB	39H	往自由存取区写 N 个数据, 数据指针循环增加
F_WRFREEC	3AH	往自由存取区写 16 个数据, 数据指针循环增加
F_WRFREED	3BH	往自由存取区写一个数据, 但不改变数据指针
F_RDFREEB	3CH	从自由存取区读 N 个数据, 数据指针循环增加
F_RDFREED	3DH	从自由存取区读 N 个数据, 但不改变数据指针
F_EEPWREN	42H	设置可编程自由存取区的写保护开关
F_SETEDPTR	44H	设置可编程自由存取区数据指针
F_GETEDPTR	45H	获取可编程自由存取区的数据指针

功能名称	命令 代码	功能简介
F_WREEPFA	48H	往可编程自由存取区写一个数据, 数据指针循环增一
F_WREEPFB	49H	往可编程自由存取区写 N 个数据, 数据指针循环增加
F_WREEPFC	4AH	往可编程自由存取区写 16 个数据, 数据指针循环增加
F_WREEPFD	4BH	往可编程自由存取区写一个数据, 但不改变数据指针
F_RDEEPFB	4CH	从可编程自由存取区读 N 个数据, 数据指针循环增加
F_RDEEPFD	4DH	从可编程自由存取区读 N 个数据, 但不改变数据指针
F_COPYPAGE	50H	复制整个页面
F_SAVEACTRECT	51H	保存活动区域的内容到其它页面
F_RESUACTRECT	52H	从其它页面中恢复活动区域的内容
F_CLRSCREEN	56H	清除整个操作页面的显示内容
F_FILLRECT	58H	填充矩形区域
F_FILLACTRECT	59H	填充整个活动区域(其实是填充整个可视活动区域)
F_FILLRECTOUT	5AH	填充矩形以外的区域
F_DISPBYTE	5DH	显示文本像素
F_DISPBMPIA	5EH	显示通用显示信息缓冲区内的位图
F_DISPBMPIB	5FH	显示预置的位图
F_DISPCHR	60H	显示字符
F_DISPSTRA	61H	显示通用显示信息缓冲区内的字符串
F_DISPSTRB	62H	显示预置的字符串
F_DISPTXTA	63H	显示通用显示信息缓冲区内的字符串, 具有自动换行功能
F_DISPTXTB	64H	显示预置的字符串, 具有自动换行功能
F_DRAWPIXEL	67H	显示图形像素
F_DRAWHLINE	68H	绘制水平线
F_DRAWVLINE	69H	绘制垂直线
F_DRAWRECT	6AH	绘制单边矩形框
F_DRAWRECTSHD	6BH	绘制单边矩形框的右下边线
F_DRAWSHDRECT	6CH	绘制带阴影的单边矩形框
F_DRAWLINE	6DH	绘制任意两点之间的直线段
F_SHOWHSCOLL	70H	显示水平滚动条
F_SHOWVSCOLL	71H	显示垂直滚动条
F_DISPFIX16	74H	无符号十六位定点数值十进制显示
F_DISPFIX16	75H	十六位定点数值十进制显示
F_VIEWSPAGE	80H	显示静态页面
F_VIEWAPAGE	81H	显示动态页面
F_VIEWPAGE	82H	显示页面
F_VIEWOBJECT	84H	显示对象
F_SETFOCUSED	88H	设置对象的焦点信息
F_GETFOCUSED	89H	获得对象的焦点信息
F_GETOBJECTITEMS	8CH	获得单选框, 复选框, 列表框的选项数目或数值编辑器的有效位数
F_SETOBJCUR	8DH	设置单选框, 复选框, 列表框的选中项序号或数值编辑器光标位序号
F_GETOBJCUR	8EH	获得单选框, 复选框, 列表框的选中项序号或数值编辑器光标位序号
F_INCOBJCUR	8FH	循环递增单选框, 复选框, 列表框的选中项序号或数值编辑器光标位序号
F_DECOBJCUR	90H	循环递减单选框, 复选框, 列表框的选中项序号或数值编辑器光标位序号

功能名称	命令 代码	功能简介
F_SETCHECK	92H	设置复选框所有选项状态
F_GETCHECK	93H	获得复选框所有选项状态
F_ALTCHKISTA	94H	切换复选框某项状态
F_GETCHKISTA	95H	获得复选框某项状态
F_ALTCHKESTA	96H	切换复选框选中项状态
F_GETCHKESTA	97H	获得复选框选中项状态
F_INCLSTIS_V	9AH	限制递增列表框选中项行号
F_DECLSTIS_V	9BH	限制递减列表框选中项行号
F_SETSCVALUE	9EH	设置滚动条当前值和最大值
F_GETSCVALUE	9FH	获得滚动条当前值和最大值
F_SETVALFMT	A2H	设置动态数值框的显示格式
F_SETAVALUE	A3H	设置动态数值框的显示数据
F_SETSTDELMT	A6H	设置标准十进制定点数值编辑器的数值极限, 并重新调整编辑数值
F_SETSTDEVAL	A7H	设置标准十进制定点数值编辑器的数值
F_GETSTDEVAL	A8H	获得标准十进制定点数值编辑器的数值和极限值
F_SETCURVALUE	A9H	设置标准数值编辑器光标处数值
F_INCCURVALUE	AAH	标准数值编辑器光标处数值增一并且向前进位
F_DECCURVALUE	ABH	标准数值编辑器光标处数值减一并且向前借位
F_ALTSTDVDESIGN	ACH	标准编辑器数值符号取反
F_GETATXT_LEN	B0H	获得动态文本框的长度和最大长度
F_SETATXT_XY	B1H	设置动态文本框显示内容的坐标
F_GETATXT_XY	B2H	获得动态文本框显示内容的坐标
F_ATXTCLR	B3H	清动态文本框的字符串内容
F_ATXTAPPEND	B4H	在动态文本框的末尾添加文本
F_ATXTINSERT	B5H	在动态文本框中插入文本
F_ATXTDELETE	B6H	在动态文本框中删除字符串

表 15 - SC165C010 命令代码索引表

表 15注: 关于每一条命令的详细说明请参看《SC165C010 命令使用指南》。

软硬件接口示例

软件接口程序的代码取决于 MCU 的类型和硬件 提供相应的软件接口代码。

接口的电路连接方式。下面根据两个特定的电路连

◆ 示例一

- 主要硬件电路（图中的LCM的详细电路图请看图 9）

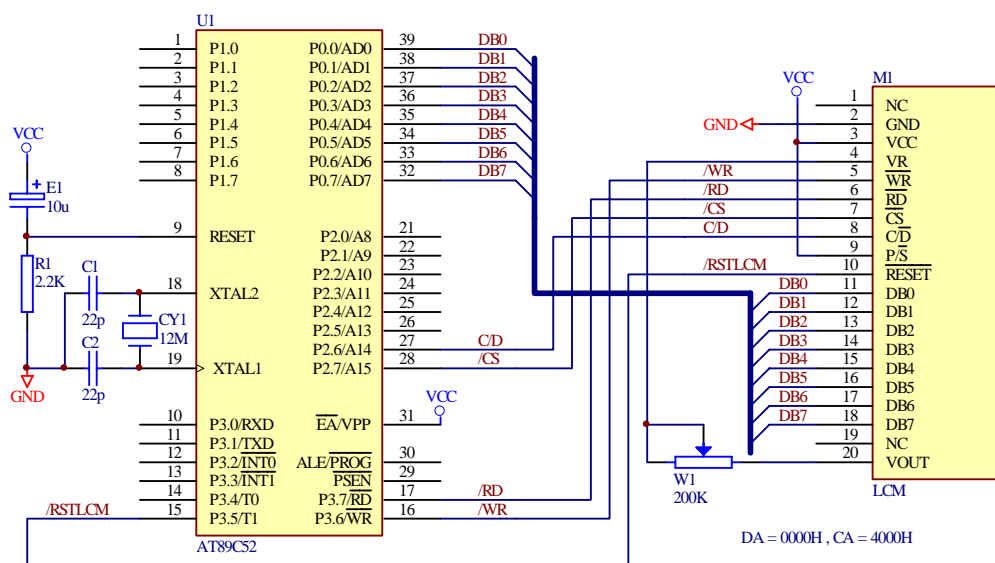


图 11 - 89C52 与液晶模块采用并行硬件接口示例

- 依据图 11 的软件接口示例

1. 符号定义 (对以下代码中使用到的符号进行定义)

LCD_DA	EQU	0000H	:根据图 11确定的液晶模块数据通道地址
LCD_CA	EQU	4000H	:根据图 11确定的液晶模块命令通道地址
F_GETFILTER	EQU	11H	:获得过滤器编号命令代码, 见表 15
F_CLRSCREEN	EQU	56H	:清屏命令代码, 见表 15
F_DRAWLINE	EQU	6DH	:画线命令代码, 见表 15

L_IN_BYTES	EQU	30H	:输入参数首字节寄存器(共 16 字节)
L_OUT_BYTES	EQU	40H	:输出参数首字节寄存器(共 16 字节)
L_IN_COUNT	EQU	50H	:输入参数字节数寄存器
L_OUT_COUNT	EQU	51H	:输出参数字节数寄存器
L_LCD_CMD	EQU	52H	:命令代码寄存器

2. 等待液晶空闲子程序

```

;;
;;名称: F_WAITLCD_P
;;功能: 等待液晶空闲(状态码为 0FFH)
;;类型: 全局函数
;;输入: (无)
;;输出: (无)
;;
;;调用: (无)
;;影响: PSW, A, DPTR
;;堆栈: 2
;;
;;注意: 只适用于并行接口方式
;;
F_WAITLCD_P:
    MOV     DPTR,    #LCD_CA
WAITLCDP_LP:
    MOVX    A,       @DPTR           ;获取液晶当前状态码, 它从命令通道读出
    CJNE    A, #0FFH, WAITLCDP_LP   ;当前状态码不是 FF, 忙, 循环等待
    ;
WAITLCDP_RET:
    RET

```

3. 清屏程序（无输入无输出参数例子）

```

LCALL     F_WAITLCD_P           ;等待 LCD 空闲
MOV       DPTR,    #LCD_CA      ;液晶的命令通道地址
MOV       A,       #F_CLRSCREEN
MOVX      @DPTR,    A

```

4. 从起点坐标为 (0, 0) 到终点坐标为 (32, 48) 画一条斜线段程序（有输入无输出参数例子）

```

LCALL     F_WAITLCD_P           ;等待 LCD 空闲
MOV       DPTR,    #LCD_DA      ;液晶的数据通道地址
MOV       A,       #00H         ;[0], X1H
MOVX      @DPTR,    A
MOV       A,       #00H         ;[1], X1L
MOVX      @DPTR,    A
MOV       A,       #00H         ;[2], Y1H
MOVX      @DPTR,    A
MOV       A,       #00H         ;[3], Y1L
MOVX      @DPTR,    A
MOV       A,       #00H         ;[4], X2H
MOVX      @DPTR,    A
MOV       A,       #20H         ;[5], X2L
MOVX      @DPTR,    A
MOV       A,       #00H         ;[6], Y2H
MOVX      @DPTR,    A
MOV       A,       #30H         ;[7], Y2L
MOVX      @DPTR,    A
MOV       A,       #FFH         ;[8], STYLE
MOVX      @DPTR,    A

```

```

MOV    DPTR,    #LCD_CA      ;液晶的命令通道地址
MOV    A,        #F_DRAWLINE
MOVX   @DPTR,    A

```

5. 获得过滤器编号程序（无输入有输出参数例子）

```

LCALL   F_WAITLCD_P          ;等待 LCD 空闲
MOV     DPTR,    #LCD_CA      ;液晶的命令通道地址
MOV     A,        #F_GETFILTER
MOVX    @DPTR,    A
LCALL   F_WAITLCD_P          ;等待 LCD 空闲
MOV     DPTR,    #LCD_DA      ;液晶的数据通道地址
MOV     A,        @DPTR       ;A = 过滤器编号

```

6. 依据图 10的通用子程序

```

;-----
;名称: F_CTRLLCD_P
;功能: 向 SC165C010 发送命令的通用子程序
;类型: 全局函数
;输入: L_IN_BYTES[0..15]      输入参数, L_IN_BYTES[0]先送
;      L_IN_COUNT              输入参数的字节数
;      L_LCD_CMD               命令代码
;      L_OUT_COUNT             输出参数的字节数
;输出: L_OUT_BYTES[0..15]      输出参数, 先获得 L_OUT_BYTES[0]
;-----
;调用: F_WAITLCD_P
;影响: PSW, R0, A, B, DPTR
;堆栈: 6
;-----
;注意: 只适用于并行接口方式, 使用时注意输入参数和输出参数缓冲区所在地址
;-----
F_CTRLLCD_P:
    PUSH    L_IN_COUNT
    PUSH    L_OUT_COUNT
    ;
    LCALL   F_WAITLCD_P        ;等待 LCD 空闲
    MOV     A,    L_IN_COUNT
    JZ      CTRLLCDP_NT        ;若输入参数字节数为 0, 直接发送命令
    MOV     DPTR,    #LCD_DA    ;连续发送输入参数
    MOV     R0,    #L_IN_BYTES ;R0 指向输入参数缓冲区首地址
CTRLLCDP_LP1:
    MOV     A,    @R0
    MOVX    @DPTR,    A
    INC     R0
    DJNZ    L_IN_COUNT, CTRLLCDP_LP1
    ;
CTRLLCDP_NT:
    MOV     DPTR,    #LCD_CA    ;发送命令
    MOV     A,    L_LCD_CMD
    MOVX    @DPTR,    A
    ;
    MOV     A,    L_OUT_COUNT

```

```

JZ      CTRLLCDP_RET      ;若输出参数为 0, 直接返回
LCALL   F_WAITLCD_P      ;等待 LCD 空闲
MOV     DPTR, #LCD_DA     ;连续读取输出参数
MOV     R0, #L_OUT_BYTES  ;R0 指向输出参数缓冲区首地址

CTRLLCDP_LP2:
MOVX    A, @DPTR
MOV     @R0, A
INC     R0
DJNZ    L_OUT_COUNT, CTRLLCDP_LP2
;
CTRLLCDP_RET:
POP     L_OUT_COUNT
POP     L_IN_COUNT
RET
    
```

◆ 示例二

- 主要硬件电路（图中的LCM的详细电路图请看图 9）

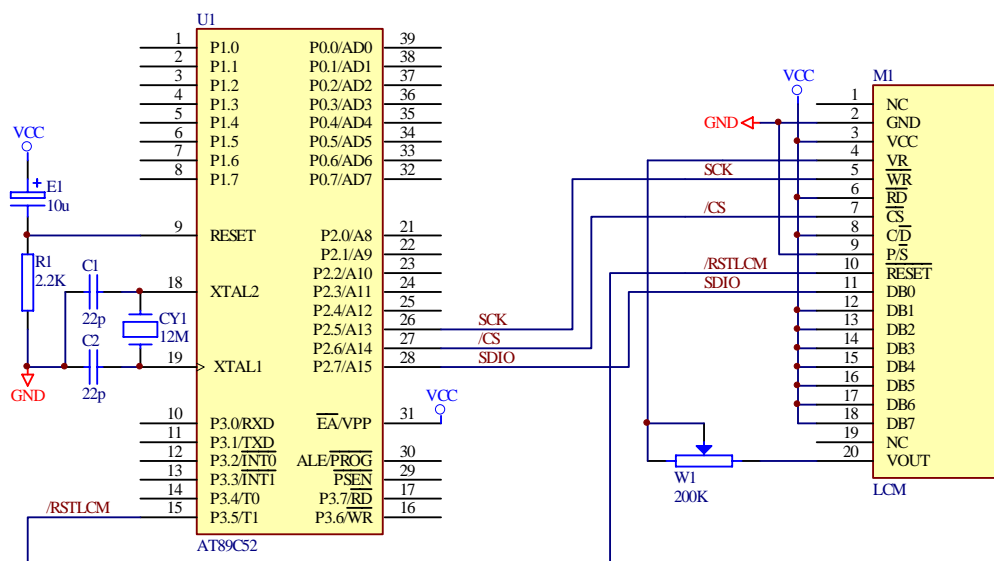


图 12 - 89C52 与液晶模块采用串行硬件接口示例

- 依据图 12 的软件接口示例

1. 符号定义（对以下代码中使用到的符号进行定义）

SCK	EQU	P2. 5	;SCK 引脚定义
NOT_CS	EQU	P2. 6	; /CS 引脚定义
SDIO	EQU	P2. 7	;SDIO 引脚定义
LCD_DA	EQU	00H	;液晶模块数据通道地址, 只与程序有关
LCD_CA	EQU	01H	;液晶模块命令通道地址, 只与程序有关
F_CLRSCREEN	EQU	56H	;清屏命令代码, 见表 15

L_LCDADDR	EQU	30H	;液晶通道地址寄存器
L_LCDBYTE	EQU	31H	;液晶操作数据寄存器

2. 根据图 5（a）的串行写操作子程序

```
;;-----  
;;名称: F_WR_LCDBYTE  
;;功能: 用串行方式向 SC165C010 写一个字节  
;;类型: 全局函数  
;;输入: L_LCDBYTE 输入数据  
;;      L_LCDADDR 数据输入的通道地址  
;;输出: (无)  
;;-----  
;;  
;;调用: (无)  
;;影响: PSW, A  
;;堆栈: 3  
;;-----  
;;注意: 只适用于串行接口方式;  
;;      当 MCU 的指令执行速度较高时, 应注意该子程序能否保证 SCK 的高低电平保持时间  
;;-----  
F_WR_LCDBYTE:  
    PUSH    B  
    ;  
    SETB    NOT_CS          ;/CS = 1  
    SETB    SCK  
    MOV     A,    L_LCDADDR  
    ;  
    CLR     NOT_CS          ;/CS = 0  
    NOP  
    ;  
    CLR     SCK              ;WRITE L_LCDADDR  
    MOV     C,    ACC.0  
    MOV     SDIO,  C  
    NOP  
    SETB    SCK  
    NOP  
    ;  
    CLR     SCK              ;WRITE WR FLAG  
    SETB    SDIO  
    NOP  
    SETB    SCK  
    NOP  
    ;  
    MOV     A,    L_LCDBYTE  
    MOV     B,    #8  
WRLCDBYTE_LP:                ;WRITE BIT7..BIT0  
    CLR     SCK  
    RLC     A  
    MOV     SDIO,  C  
    NOP  
    SETB    SCK  
    NOP
```



```

CJNZ B, WRLCDBYTE_LP
;
CLR SCK ;WRITE PULSE
NOP
SETB SCK
NOP
;
SETB NOT_CS ;/CS = 1
NOP
;
WRLCDBYTE_RET:
POP B
RET

```

3. 根据图 5 (b) 的串行读操作子程序

```

;;
;;名称: F_RD_LCDBYTE
;;功能: 用串行方式从 SC165C010 读一个字节
;;类型: 全局函数
;;输入: L_LCDADDR 数据输入的通道地址
;;输出: L_LCDBYTE 输出数据
;; A A = L_LCDBYTE
;;
;;调用: (无)
;;影响: PSW
;;堆栈: 3
;;
;;注意: 只适用于串行接口方式;
;; 当 MCU 的指令执行速度较高时, 应注意该子程序能否保证 SCK 的高低电平保持时间
;;
F_RD_LCDBYTE:
PUSH B
;
SETB NOT_CS ;/CS = 1
SETB SCK
MOV A, L_LCDADDR
;
CLR NOT_CS ;/CS = 0
NOP
;
CLR SCK ;WRITE L_LCDADDR
MOV C, ACC.0
MOV SDIO, C
NOP
SETB SCK
NOP
;
CLR SCK ;WRITE RD FLAG
CLR SDIO
NOP
SETB SCK
NOP

```

```

;
CLR    SCK                      ;READ PULSE
NOP
SETB   SCK
NOP
;
SETB   SDIO                     ;PORT INPUT
;
MOV     B,    #8
RDLCD_BYTE_LP:                  ;READ BIT7..BIT0
CLR     SCK
NOP
MOV     C,    SDIO
RLC     A
SETB    SCK
NOP
DJNZ    B,    RDLCD_BYTE_LP
;
SETB    NOT_CS                  ;/CS = 1
NOP
;
MOV     L_LCD_BYTE, A
;
RDLCD_BYTE_RET:
POP     B
RET

```

4. 等待液晶空闲子程序

```

;;
;;名称: F_WAITLCD_S
;;功能: 等待液晶空闲(状态码为 0FFH)
;;类型: 全局函数
;;输入: (无)
;;输出: (无)
;;
;;调用: F_RD_LCD_BYTE
;;影响: PSW, A, L_LCDADDR, L_LCD_BYTE
;;堆栈: 5
;;
;;注意: 只适用于串行接口方式
;;
F_WAITLCD_S:
MOV     L_LCDADDR, #LCD_CA      ;液晶的命令通道地址
WAITLCD_LP:
LCALL   F_RD_LCD_BYTE           ;获取液晶当前状态码, 它从命令通道读出
CJNE    A, #0FFH, WAITLCD_LP   ;忙, 循环等待
;
WAITLCD_RET:
RET

```

5. 清屏程序（无输入无输出参数例子）

```
LCALL F_WAITLCD_S      ;等待 LCD 空闲
MOV    L_LCDADDR, #LCD_CA ;液晶的命令通道地址
MOV    L_LCDBYTE, #F_CLRSCREEN
LCALL  F_WR_LCDBYTE     ;写一个字节
```

6. 获得过滤器编号程序（无输入有输出参数例子）

```
LCALL F_WAITLCD_S      ;等待 LCD 空闲
MOV    L_LCDADDR, #LCD_CA ;液晶的命令通道地址
MOV    L_LCDBYTE, #F_GETFILTER
LCALL  F_WR_LCDBYTE     ;写一个字节
LCALL  F_WAITLCD_S      ;等待 LCD 空闲
MOV    L_LCDADDR, #LCD_DA ;液晶的数据通道地址
LCALL  F_RD_LCDBYTE     ;读一个字节, A = L_LCDBYTE = 过滤器编号
```

封装外形尺寸

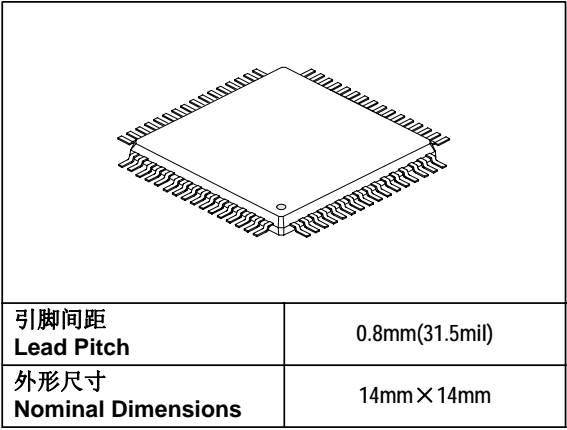


图 13 - QFP64-14×14-0.8 封装外形

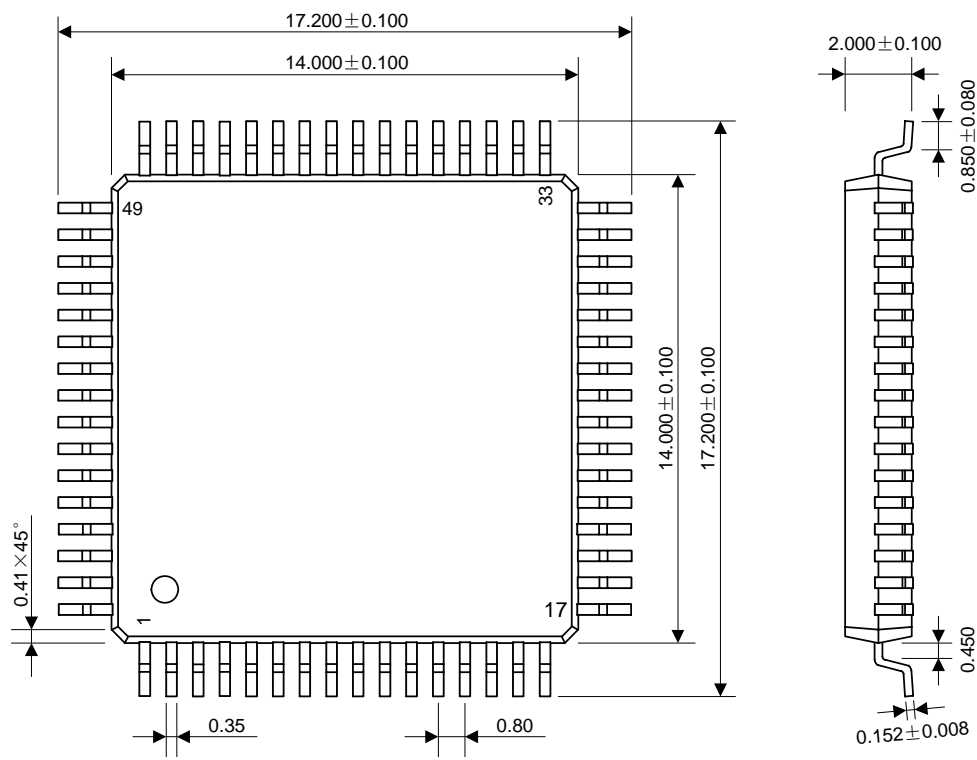


图 14 - QFP64-14×14-0.8 封装尺寸