

图形点阵液晶显示模块使用手册

ZX19264D

目 录

(一) 概述	(1)
(二) 外形尺寸图	(1)
(三) 模块主要硬件构成说明	(2)
(四) 模块的外部接口	(3)
(五) 指令说明	(3)
(六) 读写操作时序	(5)
(七) 应用举例	(7)

一. 概述

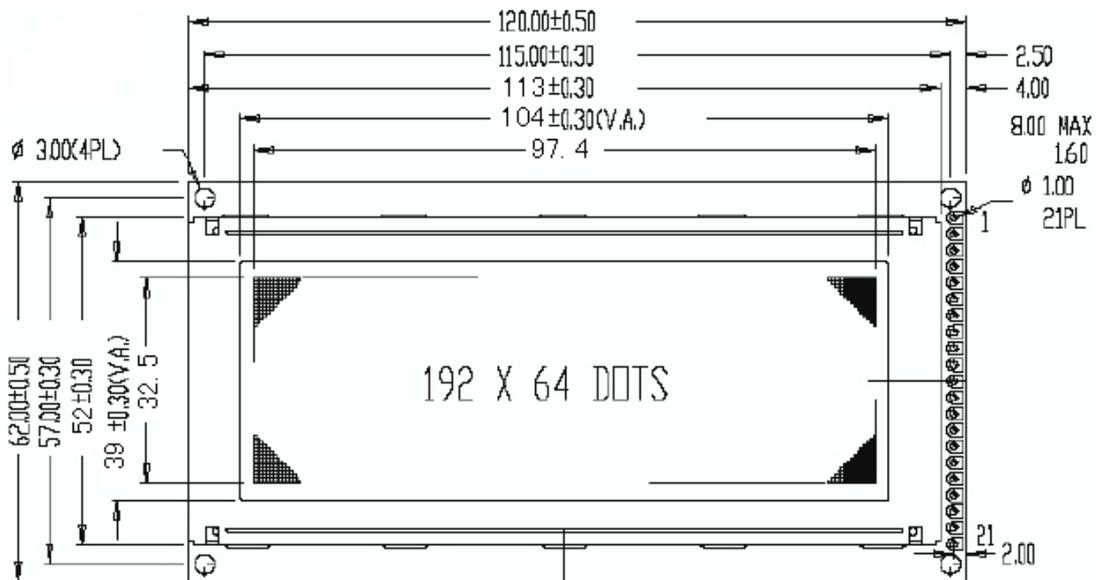
ZX19264D 是一种图形点阵液晶显示器,它主要由行驱动器/列驱动器及格 192×64 全点阵液晶显示器组成。可完成图形显示,也可以显示 12×4 个(16×16 点阵)汉字。

主要技术参数和性能:

1. 电源: VDD: +5V;
2. 显示内容: 192(列)×64(行)点
3. 全屏幕点阵
4. 七种指令
5. 与 CPU 接口采用 8 位数据总线并行输入输出和 8 条控制线
6. 占空比 1/64
7. 工作温度: $-10^{\circ}\text{C} \sim +55^{\circ}\text{C}$, 存储温度: $-20^{\circ}\text{C} \sim +70^{\circ}\text{C}$

二. 外形尺寸图

1. 外形尺寸图



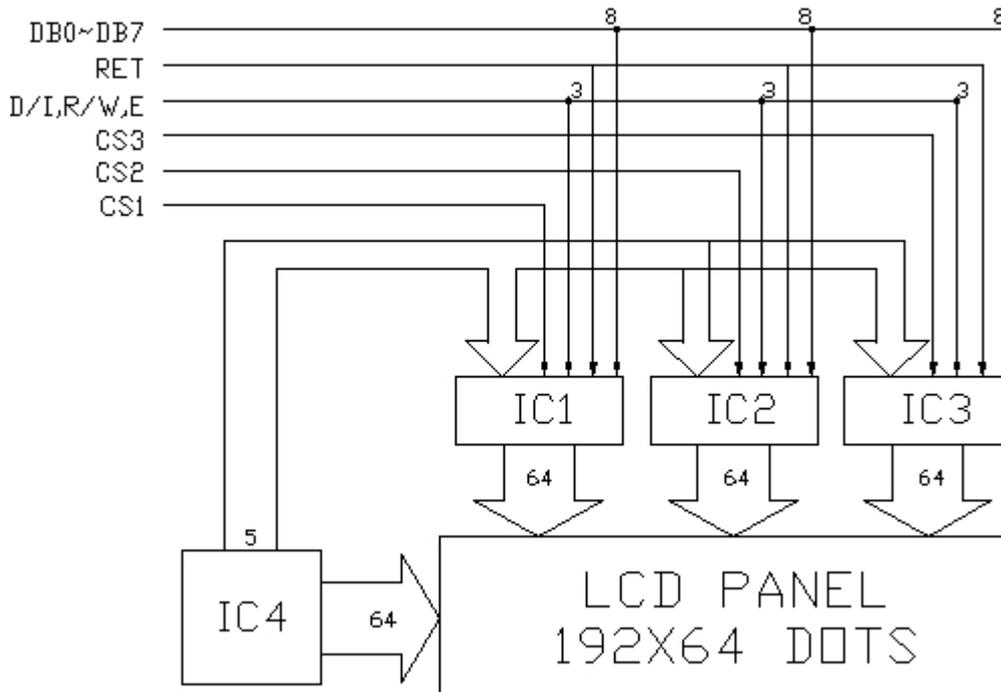
2. 外形尺寸

表 1

ITEM	NOMINAL DIMEN	UNIT
模块体积	130×65×12.5	mm
视域	104×39	mm
行列点阵数	192×64	dots
点距离	0.508×0.508	mm
点大小	0.458×0.458	mm

三. 模块主要硬件构成说明

(结构框图)



IC4 为行驱动器。IC1, IC2, IC3 为列驱动器。IC1, IC2, IC3, IC4 含有以下主要功能器件。了解如下器件有利于对 LCD 模块之编程。

1. 指令寄存器 (IR)

IR 是用于寄存指令码, 与数据寄存器数据相对应。当 D/I=0 时, 在 E 信号下降沿的作用下, 指令码写入 IR。

2. 数据寄存器 (DR)

DR 是用于寄存数据的, 与指令寄存器寄存指令相对应。当 D/I=1 时, 在下降沿作用下, 图形显示数据写入 DR, 或在 E 信号高电平作用下由 DR 读到 DB7~DB0 数据总线。DR 和 DDRAM 之间的数据传输是模块内部自动执行的。

3. 忙标志: BF

BF 标志提供内部工作情况。BF=1 表示模块在内部操作, 此时模块不接受外部指令和数据。BF=0 时, 模块为准备状态, 随时可接受外部指令和数据。

利用 STATUS READ 指令, 可以将 BF 读到 DB7 总线, 从检验模块之工作状态。

4. 显示控制触发器 DFF

此触发器是用于模块屏幕显示开和关的控制。DFF=1 为开显示 (DISPLAY ON), DDRAM 的内容就显示在屏幕上, DFF=0 为关显示 (DISPLAY OFF)。

DDF 的状态是指令 DISPLAY ON/OFF 和 RST 信号控制的。

5. XY 地址计数器

XY 地址计数器是一个 9 位计数器。高 3 位是 X 地址计数器, 低 6 位为 Y 地址计数器, XY 地址计数器实际上是作为 DDRAM 的地址指针, X 地址计数器为 DDRAM 的页指针, Y 地址计数器为 DDRAM 的 Y 地址指针。

X 地址计数器是没有记数功能的, 只能用指令设置。

Y 地址计数器具有循环记数功能, 各显示数据写入后, Y 地址自动加 1, Y 地址指针从 0 到 63。

6. 显示数据 RAM (DDRAM)

DDRAM 是存储图形显示数据的。数据为 1 表示显示选择，数据为 0 表示显示非选择。DDRAM 与地址和显示位置的关系见 DDRAM 地址表（见第 6 页）。

7. Z 地址计数器

Z 地址计数器是一个 6 位计数器，此计数器具备循环记数功能，它是用于显示行扫描同步。当一行扫描完成，此地址计数器自动加 1，指向下一行扫描数据，RST 复位后 Z 地址计数器为 0。

Z 地址计数器可以用指令 DISPLAY START LINE 预置。因此，显示屏幕的起始行就由此指令控制，即 DDRAM 的数据从哪一行开始显示在屏幕的第一行。此模块的 DDRAM 共 64 行，屏幕可以循环滚动显示 64 行。

四. 模块的外部接口

外部接口信号如下表 2 所示：

表 2

管脚号	管脚名称	LEVER	管脚功能描述
1	VSS	0	电源地
2	VDD	5.0V	电源电压
3	V0	5.0V~10V	液晶显示器驱动电压
4	VEE	-10V	LCD 驱动负电压
5	D/I	H/L	D/I=“H”，表示 DB7~DB0 为显示数据 D/I=“L”，表示 DB7~DB0 为显示指令数据
6	R/W	H/L	R/W=“H”，E=“H” 数据被读到 DB7~DB0 R/W=“L”，E=“H→L” 数据被写到 IR 或 DR
7	E	H/L	R/W=“L”，E 信号下降沿锁存 DB7~DB0 R/W=“H”，E=“H” DDRAM 数据读到 DB7~DB0
8	DB0	H/L	数据线
9	DB1	H/L	数据线
10	DB2	H/L	数据线
11	DB3	H/L	数据线
12	DB4	H/L	数据线
13	DB5	H/L	数据线
14	DB6	H/L	数据线
15	DB7	H/L	数据线
16	CS1	H (19264A) L (19264A1)	选择 IC1，即（左）64 列，低有效
17	CS2	H (19264A) L (19264A1)	选择 IC2，即（中）64 列，低有效
18	CS3	H (19264A) L (19264A1)	选择 IC3，即（右）64 列，低有效
19	RESET	L	复位控制信号，RST=0 有效
20	LED+	+5.0V	LED 背光板电源
21	LED-	0V	LED 背光板电源

五. 指令说明

指令表

表 3

指令	指令码										功能
	R/W	D/I	D7	D6	D5	D4	D3	D2	D1	D0	
显示 ON/OFF	0	0	0	0	1	1	1	1	1	1/0	控制显示器的开关，不影响 DDRAM 中数据和内部状态
显示起始行	0	0	1	1	显示起始行 (0...63)					指定显示屏从 DDRAM 中哪一行开始显示数据	
设置 X 地址	0	0	1	0	1	1	1	X: 0...7			设置 DDRAM 中的页地址 (X 地址)
设置 Y 地址	0	0	0	1	Y 地址 (0...63)					设置地址 (Y 地址)	
读状态	1	0	B U S Y	0	ON/ OFF	R S T	0	0	0	0	读取状态 RST 1:复位 0:正常 ON/OFF 1:显示开 0:显示关 BUSY 0:READY 1:IN OPERATION
写显示数据	0	1	显示数据								将数据线上的数据 DB7~DB0 写入 DDRAM
读显示数据	1	1	显示数据								将数据线上的数据 DB7~DB0 写入 DDRAM

1. 显示开关控制 (DISPLAY ON/OFF)

代码	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
形式	0	0	0	0	1	1	1	1	1	D

D=1: 开显示 (DISPLAY ON) 意即显示器可以进行各种显示操作

D=0: 关显示 (DISPLAY OFF) 意即不能对显示器可以进行各种显示操作

2. 设置显示起始行

代码	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
形式	0	0	1	1	A5	A4	A3	A2	A1	A0

前面在 Z 地址计数器一节已经描述了显示起始行是由 Z 地址计数器控制的。A5~A0 的 6 位地址自动送入 Z 地址计数器，起始行的地址可以是 0~63 的任意一行。

例如：

选择 A5~A0 是 62，则起始行与 DDRAM 行的对应关系如下：

DDRAM 行：62 63 0 1 2 3 28 29

屏幕显示行: 1 2 3 4 5 6 31 32

3. 设置页地址

代码	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
形式	0	0	1	0	1	1	1	A2	A1	A0

所谓页地址就是 DDRAM 的行地址, 8 行为一页, 模块共 64 行即 8 页, A2 ~ A0 表示 0 ~ 7 页。读写数据对地址没有影响, 页地址由本指令或 RST 信号改变复位后页地址为 0。页地址与 DDRAM 的对应关系见 DDRAM 地址表。

4. 设置 Y 地址 (SET Y ADDRESS)

代码	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
形式	0	0	0	1	A5	A4	A3	A2	A1	A0

此指令的作用是将 A5 ~ A0 送入 Y 地址计数器, 作为 DDRAM 的 Y 地址指针。在对 DDRAM 进行读写操作后, Y 地址指针自动加 1, 指向下一个 DDRAM 单元。

DDRAM 地址表:

表 4

		CS1=1					CS2=1					CS3=1					
Y=		0	1	..	62	63	0	1	..	62	63	0	1	..	62	63	行号
X=0		DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	0
	↓	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	7
↓		DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	8
	↓	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	55
X=7		DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	56
	↓	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	63

5. 读状态 (STATUS READ)

代码	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
形式	0	1	BUSY	0	ON/OFF	RET	0	0	0	0

当 R/W=1 D/I=0 时, 在 E 信号为“H”的作用下, 状态分别输出到数据总线 (DB7 ~ DB0) 的相应位。

BF: 前面已叙述过 (见 BF 标志位一节)。

ON/OFF: 表示 DFF 触发器的状态 (见 DFF 触发器一节)。

RST: RST=1 表示内部正在初始化, 此时组件不接受任何指令和数据。

6. 写显示数据 (WRITE DISPLAY DATE)

代码	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
形式	0	1	D7	D6	D5	D4	D3	D2	D1	D0

D7 ~ D0 为显示数据, 此指令把 D7 ~ D0 写入相应的 DDRAM 单元, Y 地址指针自动加 1。

7. 读显示数据 (READ DISPLAY DATE)

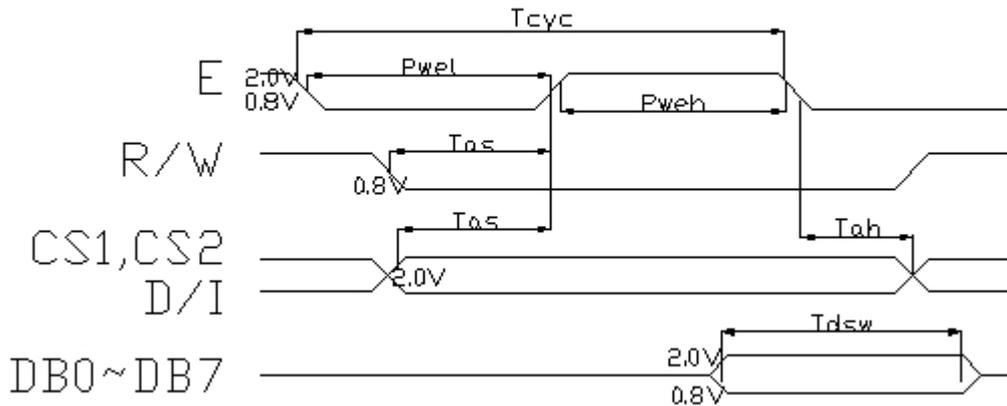
代码	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
形式	1	1	D7	D6	D5	D4	D3	D2	D1	D0

此指令把 DDRAM 的内容 D7~D0 读到数据总线 DB7~DB0, Y 地址指针自动加 1。

六. 读写操作时序

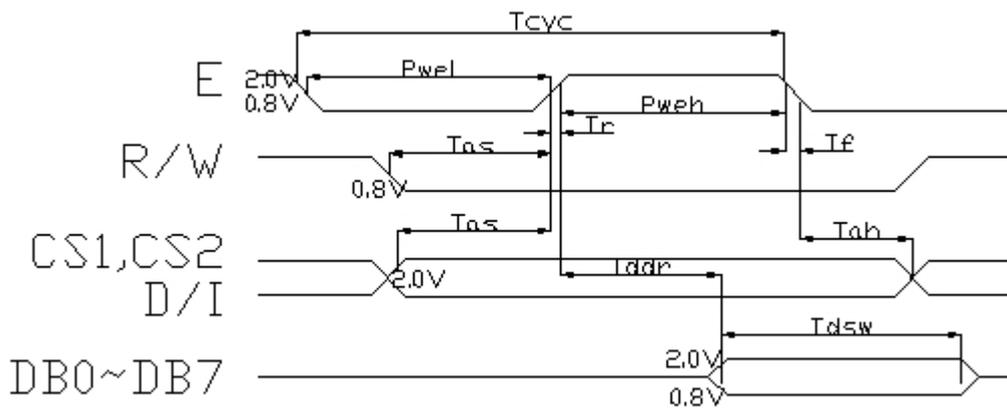
1. 写操作时序

图 3



2. 读操作时序

图 4



3. 读写时序参数表

表 5

名称	符号	最小值	典型值	最大值	单位
E 周期时间	Tcyc	1000	---	---	ns
E 高电平宽度	Pweh	450	---	---	ns
E 低电平宽度	Pwel	450	---	---	ns
E 上升时间	Tr	---	---	25	ns
E 下降时间	Tf	---	---	25	ns
地址建立时间	Tas	140	---	---	ns
地址保持时间	Tah	10	---	---	ns
数据建立时间	Tdsw	200	---	---	ns

数据延迟时间	Tddr	---	---	320	ns
写数据保持时间	Tdhw	10	---	---	ns
读数据保持时间	Tdhw	20	---	---	ns

七. 应用举例

19264-4 与单片机 8031 的一种接口如图 5. 所示:

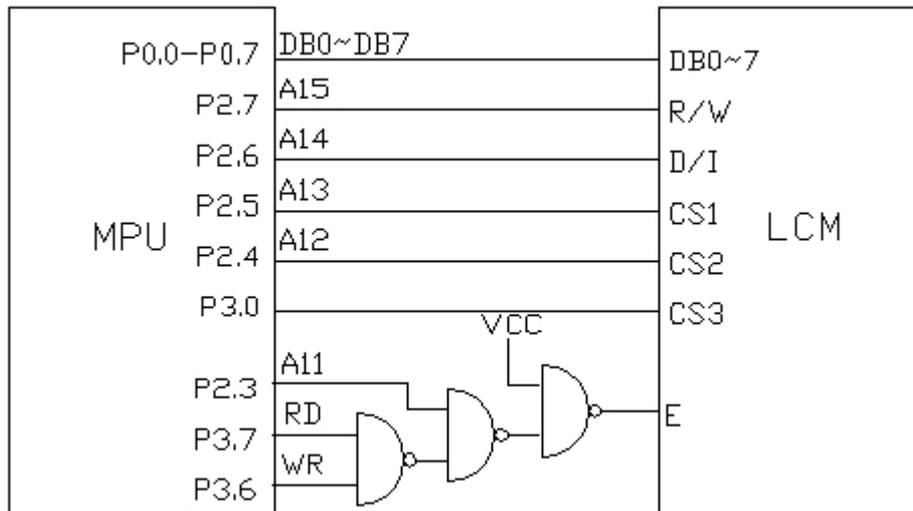


图 5

利用图 5 举例介绍几编程实例

1. 显示开/关控制 (DISPLAY ON/OFF)

所谓显示开与关, 意即当显示器置于“开”状态时, 可以对显示进行各种操作。当显示器置于“关”状态时, 不能对显示器显示进行各种显示操作。

```
MOV A, #3EH      ; 关显示代码为 3EH, 开显示操作只需将代码 3EH 换为 3FH 即可
SETB P3.0       ; Select Chip3
MOV DPTR, #3800H ; R/W For Write, And Select Chip1 and Chip2
MOVX @DPTR, A   ; Output the Instruction
CLR P3.0
```

2. 页地址设置

前面已对页地址进行定义, 即 8 行为一页。LCD (或 DDRAM) 共为 8 页。

```
MOV A, #0B8H    ; 设置页地址为 0 页
SETB P3.0      ; Select Chip3
MOV DPTR, #3800H ; R/W For Write, And Select Chip1 and Chip2
MOVX @DPTR, A  ; Output the Instruction
CLR P3.0
```

3. 写显示数据与对应 DDRAM

假设页地址, Y 地址已设置完毕, 显示数据为 DATA。

```
MOV A, #DATA    ; 设送显示数据为
SETB P3.0      ; Select Chip3
MOV DPTR, #7800H ; R/W For Write, And Select Chip1 and Chip2
MOVC @DPTR, A  ; Output the Instruction
```