

# 液晶显示模块技术手册

## 图形点阵 ZX12832C

---

## 目 录

(一) 概述 .....	3
(二) 外形尺寸 .....	3
(三) 模块主要硬件构成说明 .....	4
(四) 模块的外部接口 .....	5
(五) 指令说明 .....	6
(六) 读写操作时序 .....	7
(七) 应用举例 .....	9

## 一、概述

12832C 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 128×32 全点阵液晶显示器组成。可完成 16×2 个(16×8 点阵)ASCII 码显示,也可以显示 8×2 个(16×16 点阵)汉字。与外部 CPU 接口可采用并行方式控制。当用串口控制时,请把模块背面的 J1 跳线从短接左边跳到短接右边,即可实现串口控制,反之,不变的话就是直接用并口控制。

主要技术参数和性能:

1. 电源:VDD:+2.7~+5V。
2. 显示内容:128(列)×32(行)点。
3. 全屏幕点阵。
4. 2M ROM(CGROM)总共提供 8192 个汉字(16×16 点阵)。
5. 16K ROM(HCGROM)总共提供 128 个字符(16×8 点阵)。
6. 2MHZ 频率。
7. 工作温度: -10℃ ~ +55℃,存储温度: -20℃ ~ +60℃

## 二、外形尺寸图

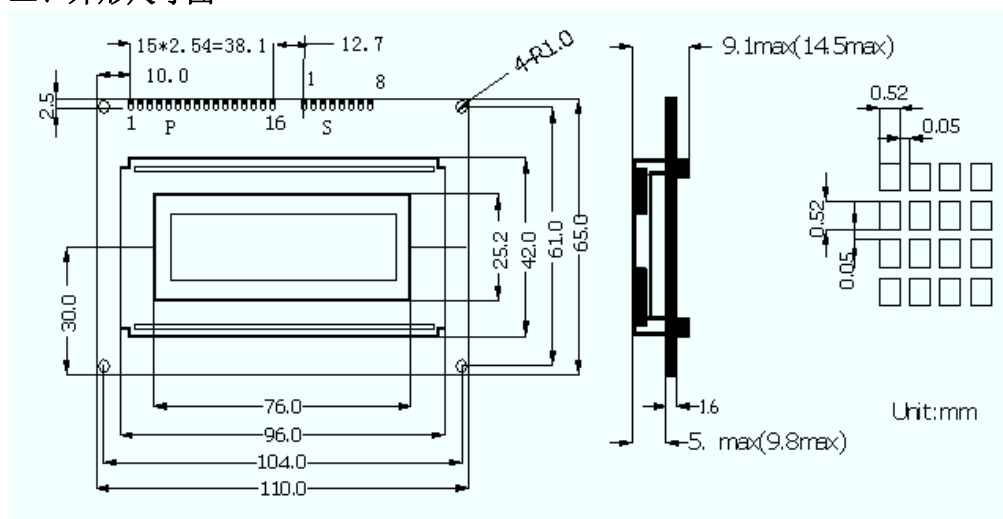


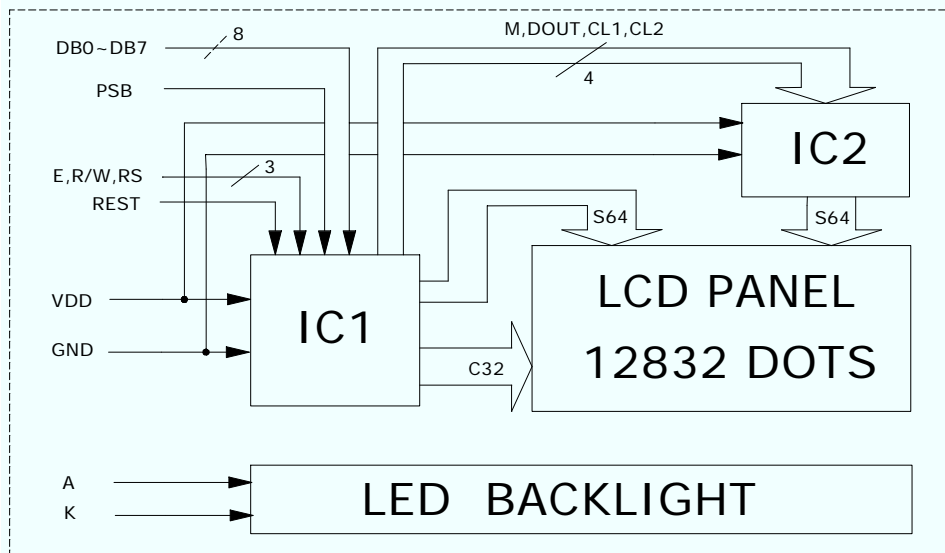
图 1

## 2. 外形尺寸图

表 1

项目	正常尺寸	单位
模块体积	110×65×14.5	mm
视域	76×25.2	mm
行列点阵数	128×32	DOTS
点距离	0.52×0.52	mm
点大小	0.57×0.57	mm

### 三、模块主要硬件构成说明



RS, R/W 的配合选择决定控制界面的 4 种模式:

RS	R/W	功能说明
L	L	MPU 写指令到指令暂存器 (IR)
L	H	读出忙标志 (BF) 及地址计数器 (AC) 的状态
H	L	MPU 写入数据到数据暂存器 (DR)
H	H	MPU 从数据暂存器 (DR) 中读出数据

- **忙标志:BF**

BF 标志提供内部工作情况. BF=1 表示模块在进行内部操作, 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 R D 指令, 可以将 BF 读到 DB7 总线, 从而检验模块之工作状态.

- **字型产生 ROM (CGROM)**

字型产生 ROM (CGROM) 提供 8192 个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)。

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

- **显示数据 RAM (DDRAM)**

显示数据 RAM 提供 64×2 个位元组的空间, 最多可控制 4 行 16 字 (64 个字) 的中文字型显示, 当写入显示数据 RAM 时, 可分别显示 CGROM 与 CGRAM 的字型; 此模块可显示三种字型, 分别是瘦长的英数字型、CGRAM 字型及 CGROM 的中文字型, 三种字型的选择, 由在 DDRAM 中写入的编码选择, 在 00~0F 的编码中将选择 CGRAM 的字定义字型, 10~7F 的编码中将选择瘦长英数字的字型, 至于 A0 以上的编码将自动的结合下一个位元组, 组成两个位元组的编码形成中文字型的编码 (A140~D75F)。

- **字型产生 RAM (CGRAM)**

字型产生 RAM 提供图象定义(造字)功能, 可以提供四组 16×16 点的自定义图象空间, 使用者可以将内部字型没有提供的图象字型自行定义到 CGRAM 中, 便可和 CGRAM 中的定义一般的通过 DDRAM 显示在荧屏中。

● **地址计数器 AC**

地址计数器是用来贮存 DDRAM/CGRAM 之一的地址, 它可由设定指令暂存器来改变, 之后只要读取或是写入 DDRAM/CGRAM 的值时, 地址计数器的值就会自动加一, 当 RS 为“0”时而 R/W 为“1”时, 地址计数器的值会被读取到 DB6~DB0 中。

● **ICON RAM (IRAM)**

IC1 提供 256 点的 ICON 显示, 它分别由 16 组的 IRAM 地址来组成, 每一组 IRAM 地址由 16 个位元构成, 每次写入一组 IRAM 时, 必须通过连续写入两个位元组的资料来完成, 先写入高位元组再写入低位元组。

● **LCD 驱动电路**

LCD 驱动电路提供 33 COMMON 以及 64 SEGMENT 信号来驱动 LCD 棉板, SEGMENT 数据从 CGRAM/CGRAM 转换储存到 64 位元的 SEGMENT 串列锁存, 当 33 个 COMMON 中的一个 COMMON 输出时, 相对应的 SEGMENT 数据将从 64 位元的串列锁存输出到 SEGMENT 驱动电路。

● **游标/闪烁控制电路**

此模块提供硬体游标及闪烁控制电路, 由地址计数器的值来指定 DDRAM 中的游标或闪烁位置。

#### 四·模块的外部接口

外部接口信号如下表 2 所示 (并行接口):

表 2

管脚号	管脚名称	LEVER	管脚功能描述
1	EL/A	AC/+5V	背光源电压
2	EL/K	AC/0V	背光源电压
3	VSS	0V	电源地
4	VCC	+5V	电源正
5	VEE	0~5.0V	液晶显示器驱动电压
6	RS	H/L	RS=“H”, 处理显示数据 RS=“L”, 处理控制指令
7	R/W	H/L	R/W=“H”, E=“H”, 数据被读到 DB7~DB0 R/W=“L”, E=“H→L”, DB7~DB0 的数据被写到 IR 或 DR
8	E	H/L	使能信号
9	DB0	H/L	数据线
10	DB1	H/L	数据线
11	DB2	H/L	数据线
12	DB3	H/L	数据线
13	DB4	H/L	数据线
14	DB5	H/L	数据线
15	DB6	H/L	数据线
16	DB7	H/L	数据线

#### 串行接口

管脚号	名称	LEVER	功能
1	BL POWER	VDD	背光源电压+4.2V~+5V
2	BL EN	H/L	背光源允许端

3	VSS	0V	电源地
4	VDD	+5V	电源正 (3.0V~5.5V)
5	VEE	-	对比度调整
6	CS	H/L	模组片选端, 高电平有效
7	SID	H/L	串行数据输入端
8	CLK	H/L	串行同步时钟: 上升沿时读取 SID 数据

## 五、指令说明

IC1 提供两套控制命令, 基本指令和扩充指令如下:

指令表 1: (RE=0: 基本指令)

指令	指令码										功能		
	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0			
清除显示	0	0	0	0	0	0	0	0	0	0	1	将 DDRAM 填满“20H”, 并且设定 DDRAM 的地址计数器 (AC) 到“00H”	
地址归位	0	0	0	0	0	0	0	0	0	1	X	设定 DDRAM 的地址计数器 (AC) 到“00H”, 并且将游标移到开头原点位置; 这个指令不改变 DDRAM 的内容	
显示状态开/关	0	0	0	0	0	0	0	1	D	C	B	D=1: 整体显示 ON C=1: 游标 ON B=1: 游标位置 ON	
进入点设定	0	0	0	0	0	0	0	0	1	I/D	S	指定在数据的读取与写入时, 设定游标的移动方向及指定显示的移位	
游标或显示移位控制	0	0	0	0	0	0	1	S/C	R/L	X	X	设定游标的移动与显示的移位控制位; 这个指令不改变 DDRAM 的内容	
功能设定	0	0	0	0	1	DL	X	RE	X	X	X	DL=1 (必须设为 1) RE=1: 扩充指令操作 RE=0: 基本指令操作	
设定 CGRAM 地址	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	AC0	设定 DDRAM 地址到地址计数器	
设定 DDRAM 地址	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	AC0	设定 CGRAM 地址到地址计数器	
读取忙标志和地址	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	AC0	读取忙标志 (BF) 可以确认内部动作是否完成, 同时可以读出地址计数器 (AC) 的值	
写数据到 RAM	1	0	数据										将数据 D7~D0 写入到内部的 RAM (DDRAM/CGRAM/IRAM/GRAM)
读出 RAM 的值	1	1	数据										从内部 RAM 读取数据 D7~D0 (DDRAM/CGRAM/IRAM/GRAM)

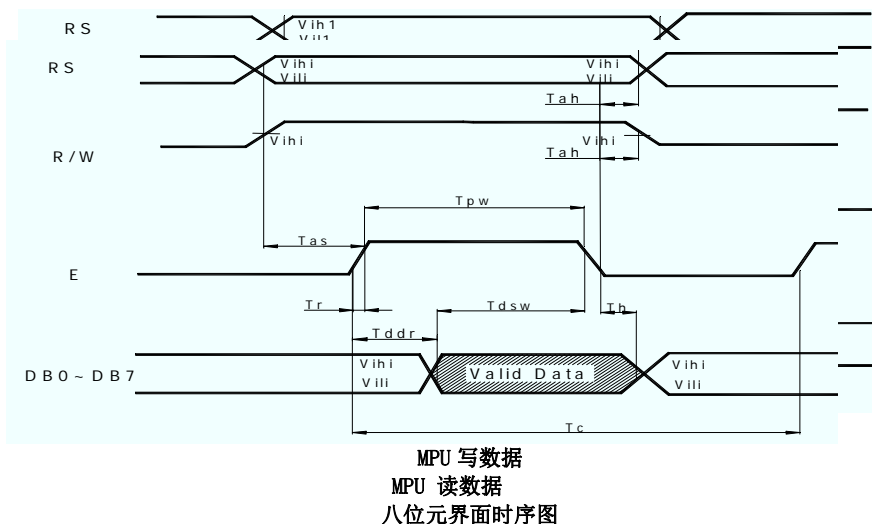
指令表 2: (RE=1: 扩充指令)

指令	指令码										功能	
	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0		
待命模式	0	0	0	0	0	0	0	0	0	0	1	进入待命模式, 执行其他指令都裸终止待命模式

卷动地址开关开启	0	0	0	0	0	0	0	0	1	SR	SR=1: 允许输入卷动地址 SR=0: 允许输入 IRAM 地址
反白选择	0	0	0	0	0	0	0	1	R1	R0	选择 4 行中的任一行作反白显示, 并可决定反白与否
睡眠模式	0	0	0	0	0	0	1	SL	X	X	SL=0: 进入睡眠模式 SL=1: 脱离睡眠模式
点距书面移位控制	0	0	0	0	0	1	OA	LR	L1	L0	OA=1: 选择单行移位 OA=0: 全部 4 行一起移位 LR=1: 点距右移 LR=0: 点距左移 L1, L0: 选择移位行
扩充功能设定	0	0	0	0	1	CL	X	RE	G	GP	CL=1 (必须设为 1) RE=1: 扩充指令操作 RE=0: 基本指令操作 G=0: (必须设为 0) GP=0: (必须设为 0)
设定 IRAM 地址或是卷动地址	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	SR=1: AC5~AC0 为卷动地址 SR=0: AC5~AC0 为 ICON RAM 地址
设定绘图 RAM 地址	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	本版本不提供此功能 0

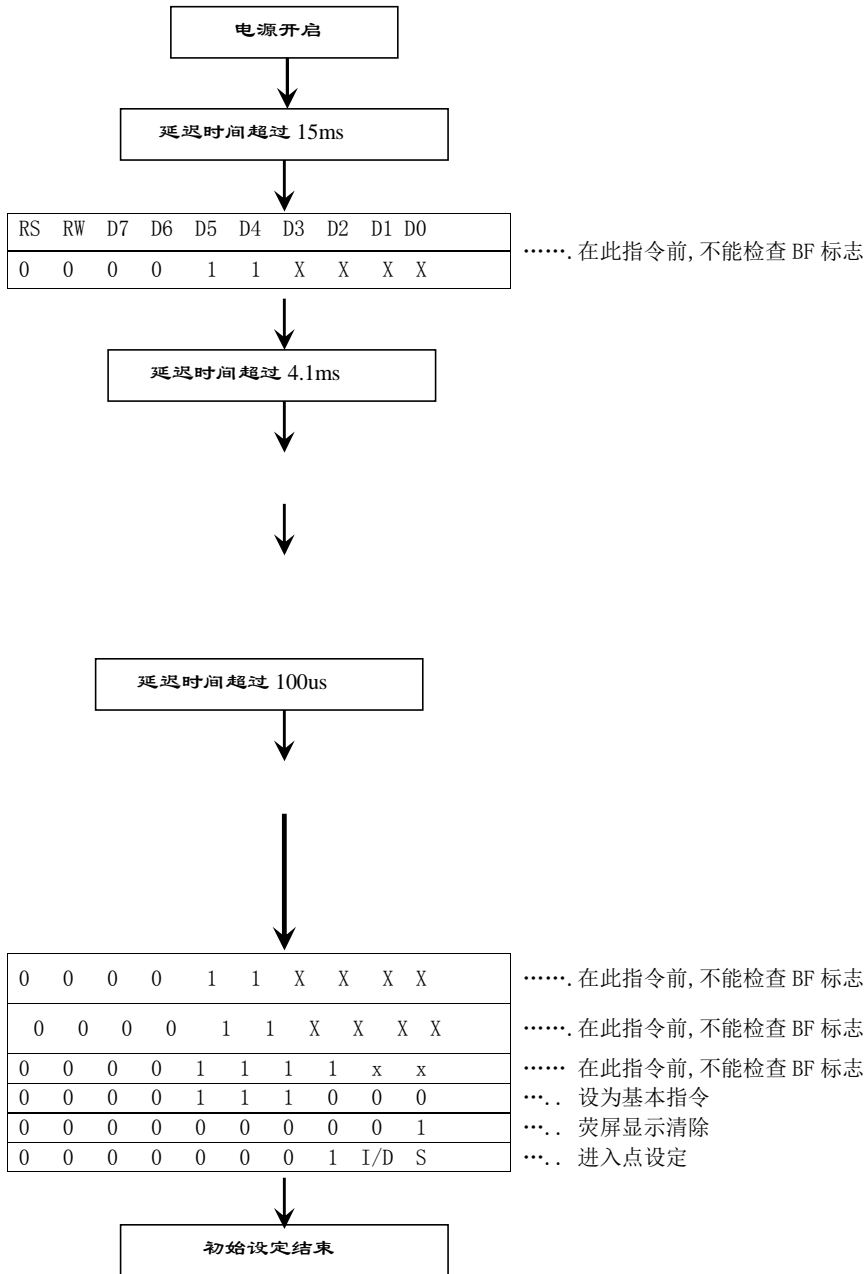
备注: 当 IC1 在接受指令前, 微处理器必须先确认其内部处于非忙碌状态, 即读取 BF 标志时, BF 需为零, 方可接受新的指令; 如果在送出一个指令前并不检查 BF 标志, 那么在前一个指令和这个指令中间必须延长一段较长的时间, 即是等待前一个指令确实执行完成。

## 六、时序图



## 七、软件初始化:

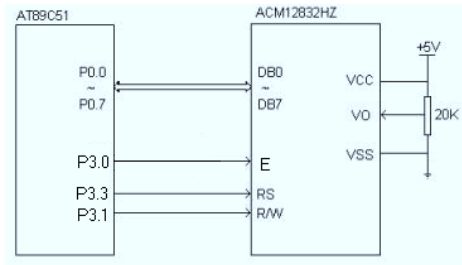
(8 位并行接口)



八. 应用举例:

12832B 与单片机 8031 的一种接口如图 5. 所示





;This program is for 12832B

```
RS EQU P3.3
RW EQU P3.1
E EQU P3.0
```

```
ORG 0000H
AJMP START
ORG 0100H
START:
    mov SP, #67h
    LCALL DELAY
    LCALL DELAY
    LCALL SETUP

    MOV A, #80H
    LCALL WRITE_COM
    MOV R3, #8

    LCALL DELAY
    LCALL DELAY
    LCALL DELAY
    LCALL DELAY
    LCALL DELAY

    MOV A#80H ;WORD TEST
    LCALL WRITE_COM
    MOV DPTR, #CHINESE
    LCALL WRITE_HZ
    MOV A, #90H
    LCALL WRITE_COM
    MOV DPTR, #TABLE1
    LCALL WRITE_ASCII
    LCALL DELAY
    LCALL DELAY
    LCALL DELAY
    LCALL DELAY
    LCALL DELAY
    MOV A#80H
    LCALL WRITE_COM
```

---

```

MOV DPTR,#table1
LCALL WRITE_ascii
MOV A,#90H
LCALL WRITE_COM
MOV DPTR,#chinese
LCALL WRITE_hz
LCALL DELAY
LCALL DELAY
LCALL DELAY
LCALL DELAY
LCALL DELAY
AAA:  LJMPL START

SETUP:
LCALL DELAY
LCALL DELAY
LCALL DELAY
MOV A,#01H ;CLEAR DISPLAY
LCALL WRITE_COM
MOV A,#00110000B ;FUNCTION SETTING
LCALL WRITE_COM
MOV A,#00000010B ;DDRAM SET TO '00H'
LCALL WRITE_COM
MOV A,#00000100B ;
LCALL WRITE_COM
MOV A,#00001100B ;DISPLAY ON
LCALL WRITE_COM
MOV A,#00000001B ;CLEARING SCREEN
LCALL WRITE_COM
MOV A,#10000000B ;SET DDRAM ADDRESS
LCALL WRITE_COM
RET

WRITE_COM:
LCALL DELAY1
CLR RS
CLR RS
CLR RW
CLR RW
MOV P1,A
MOV P1,A
SETB E
SETB E
NOP
NOP
CLR E
CLR E
;LCALL DELAY1
RET
WRITE_DAT: ;WRITE DISPLAY DATAS TO ST79220
LCALL DELAY1
SETB RS

```

---

```

SETB RS
CLR RW
CLR RW
MOV P1, A
MOV P1, A
SETB E
SETB E
NOP
NOP
CLR E
CLR E
RET

DELAY1:
MOV R7, #010H
D11: MOV R6, #010H
DJNZ R6, $
DJNZ R7, D11
RET

DELAY:
MOV R1, #00H
D2: MOV R2, #00H
DJNZ R2, $
DJNZ R1, D2
RET

WRITE_ASCII:
MOV R4, #16
DDDD: CLR A
MOVC A, @A+DPTR
LCALL WRITE_DAT
INC DPTR
DJNZ R4, DDDD
RET

WRITE_HZ:
MOV R4, #8
DD: CLR A
MOVC A, @A+DPTR
INC DPTR
LCALL WRITE_DAT
CLR A
MOVC A, @A+DPTR
INC DPTR
LCALL WRITE_DAT
DJNZ R4, DD
RET

WRITE_CGRAM: ;CGRAM TESTING
CLR A
MOVC A, @A+DPTR
LCALL WRITE_DAT

```

---

```
INC DPTR
CLR A
MOVC A, @A+DPTR
LCALL WRITE_DAT
RET
```

TABLE1:

;Here is ASCII codes

CGRAM1: DB 000H, 000H

CGRAM2: DB 000H, 002H

CGRAM3: DB 000H, 004H

CGRAM4: DB 000H, 006H

CHINESE:

;Here is Chinese codes

DB 0D2H, 0BAH, 0BEH, 0A7H, 0C4H, 0A3H, 0BFH, 0E9H ;液晶模块

DB 0CFH, 0D4H, 0CAH, 0BEH ;显示

END