

产品说明书

————RXK12232A(带中文字库)————

目 录

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

一、概述

T12232A 是一种内置 8192 个 16*16 点汉字库和 128 个 16*8 点 ASCII 字符集图形点阵液晶显示器,它主要由行驱动器/列驱动器及 122×32 全点阵液晶显示器组成。可完成图形显示,也可以显示 7.5×2 个(16×16 点阵)汉字. 与外部 CPU 接口采用并行或串行方式控制。

主要技术参数和性能:

1. 电源: VDD: +3.0~+5.5V。(电源低于 4.0 伏 LED 背光需另外供电)
2. 显示内容: 122(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: -20℃ ~ +70℃

二、外形尺寸图

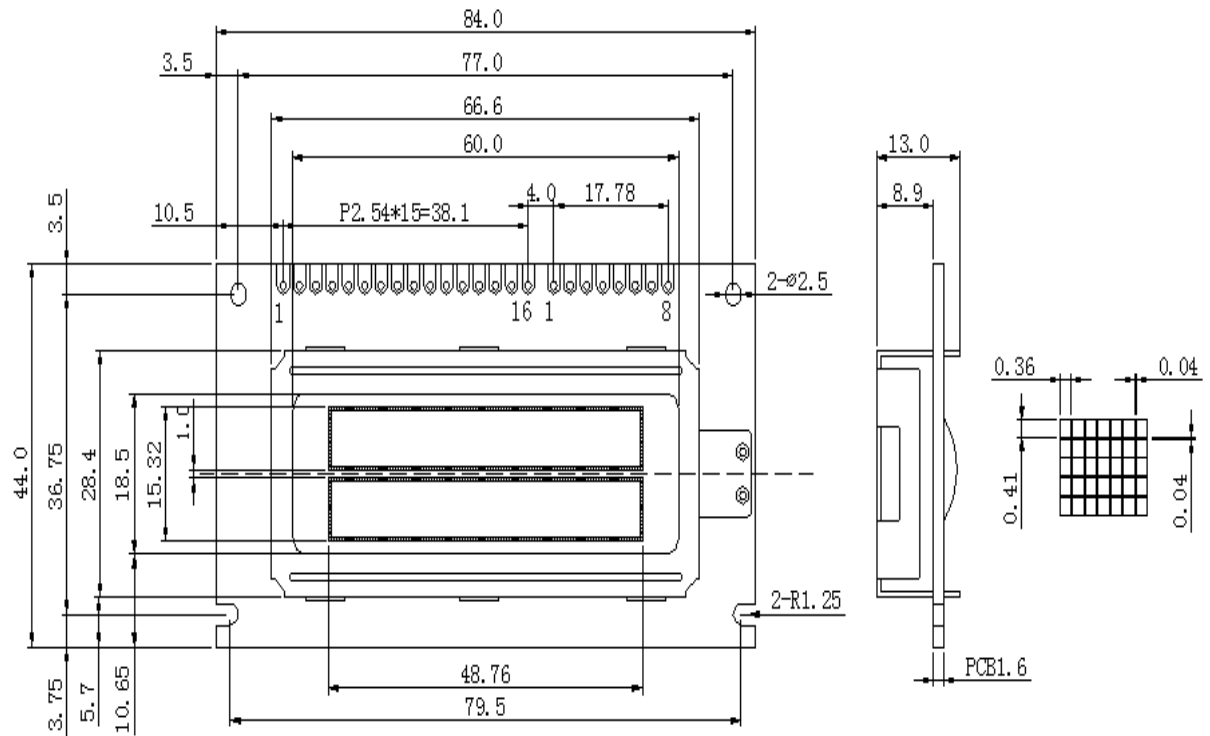


图 1

2. 外形尺寸图

表 1

项目	正常尺寸	单位
模块体积	84.0×44.0×13.0	mm
视域	60.0×18.5	mm
行列点阵数	122×32	DOTS
点距离	0.40×0.45	mm
点大小	0.36×0.41	mm

三. 模块主要硬件构成说明

控制器接口信号说明:

1、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

2、E 信号

E 状态	执行动作	结果	
高——>低	I/O 缓冲——>DR	配合/W 进行写数据或指令	
高	DR——>I/O 缓冲	配合 R 进行读数据或指令	
低/低——>高	无动作		

● 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 RD 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)

模块内部显示数据 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 (本模块只用到其中的 7.5×2 个), 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 (16×8)、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 CGROM 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。

● 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、3 所示 (并行接口):

表 2

管脚号	管脚名称	LEVER	管脚功能描述
1	VSS	0V	电源地
2	VCC	3.0+5V	电源正
3	VEE	-	对比度调整
4	RS (CS)	H/L	RS= “H”, 表示 DB7~DB0 为显示数据 RS= “L”, 表示 DB7~DB0 为显示指令数据
5	R/W (SID)	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
6	E (CLK)	H/L	使能信号

7	DB0	H/L	数据线
8	DB1	H/L	数据线
9	DB2	H/L	数据线
10	DB3	H/L	数据线
11	DB4	H/L	数据线
12	DB5	H/L	数据线
13	DB6	H/L	数据线
14	DB7	H/L	数据线
15	BL+	VDD	背光源电压+4.2V~+5V
16	BL-	VSS	背光源公共端

串口接口管脚信号

表 3

管脚号	名称	LEVER	功能
1	VSS	0V	电源地
2	VDD	+5V	电源正(3.0V~5.5V)
3	VEE	-	对比度调整
4	CLK	H/L	串行同步时钟: 上升沿时读取 SID 数据
5	SID	H/L	串行数据输入端
6	CS	H/L	模组片选端, 高电平有效
7	BL+	VDD	背光源电压+4.2V~+5V
8	BL-	VSS	背光源公共端

五、指令说明

模块控制芯片提供两套控制命令, 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	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	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	1	D	C	B	D=1: 整体显示 ON C=1: 游标 ON B=1: 游标位置反白允许
进入点设定	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	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	DL=0/1: 4/8 位数据 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	设定 CGRAM 地址	
设定 DDRAM 地址	0	0	1	0	AC5	AC4	AC3	AC2	AC1	AC0	设定 DDRAM 地址 (显示位址) 第一行: 80H-87H 第二行: 90H-97H	
读取忙标志和地址	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	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)

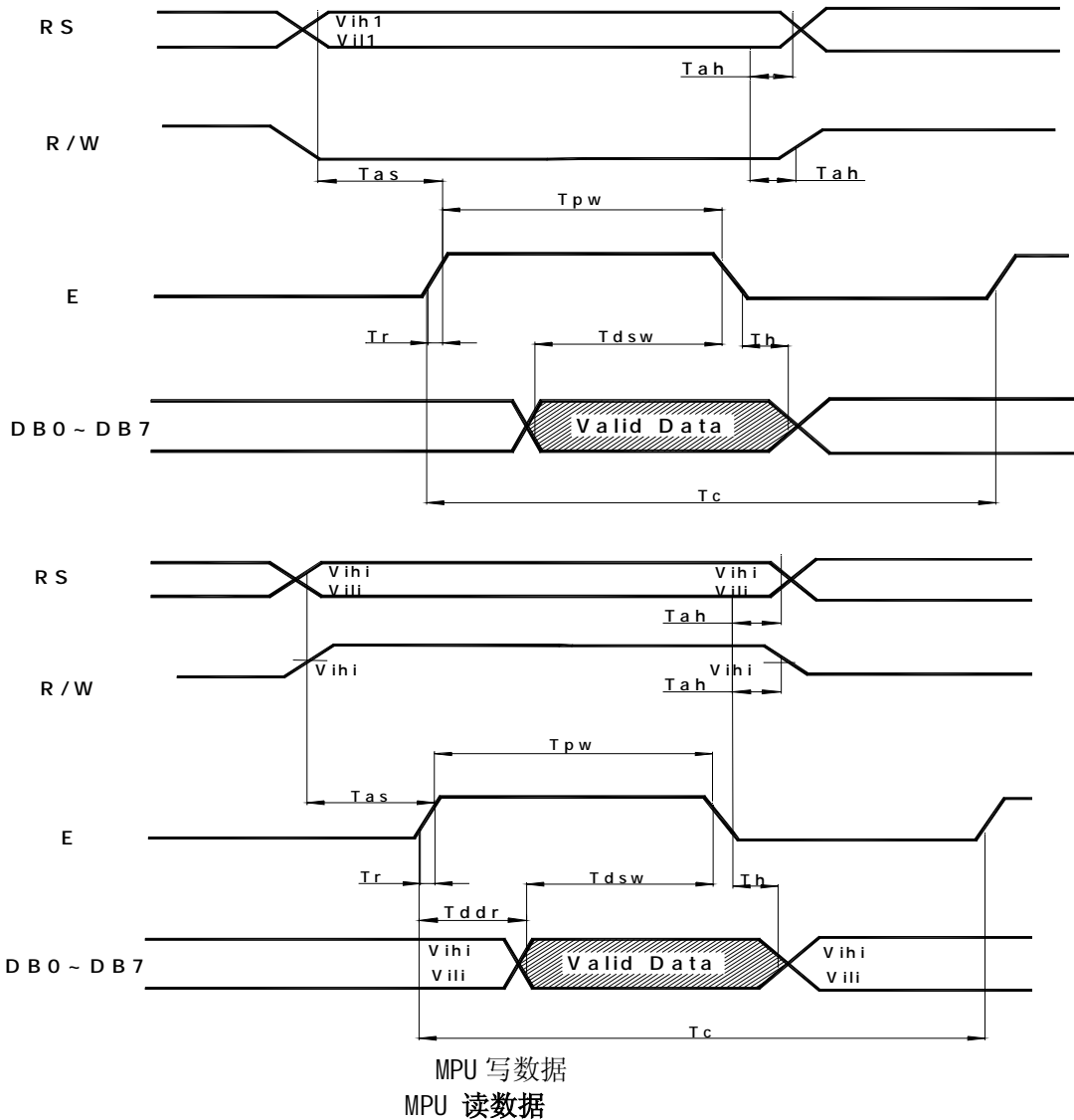
指令表 2: (RE=1: 扩充指令)

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	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: 允许输入垂直卷动地址 SR=0: 允许输入 IRAM 和 CGRAM 地址
反白选择	0	0	0	0	0	0	0	1	R1	R0	选择 2 行中的任一行作反白显示, 并可决定反白与否。初始值 R1R0=00, 第一次设定为反白显示, 再次设定变回正常
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
扩充功能设定	0	0	0	0	1	CL	X	RE	G	0	CL=0/1: 4/8 位数据 RE=1: 扩充指令操作 RE=0: 基本指令操作 G=1/0: 绘图开关
设定绘图 RAM 地址	0	0	1	0	0	0	AC3	AC2	AC1	AC0	设定绘图 RAM 先设定垂直(列)地址 AC6AC5...AC0 再设定水平(行)地址 AC3AC2AC1AC0 将以上 16 位地址连续写入即可

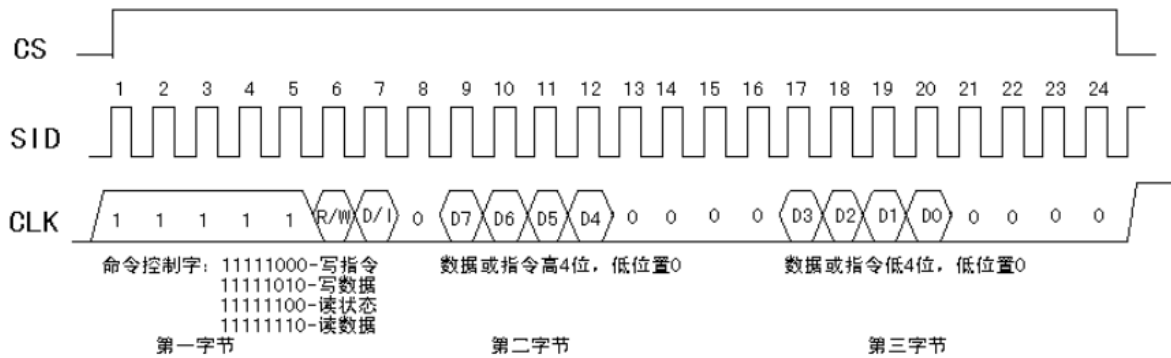
备注: 当 IC1 在接受指令前, 微处理器必须先确认其内部处于非忙碌状态, 即读取 BF 标志时, BF 需为零, 方可接受新的指令; 如果在送出一个指令前并不检查 BF 标志, 那么在前一个指令和这个指令中间必须延长一段较长的时间, 即是等待前一个指令确实执行完成。

六. 时序图

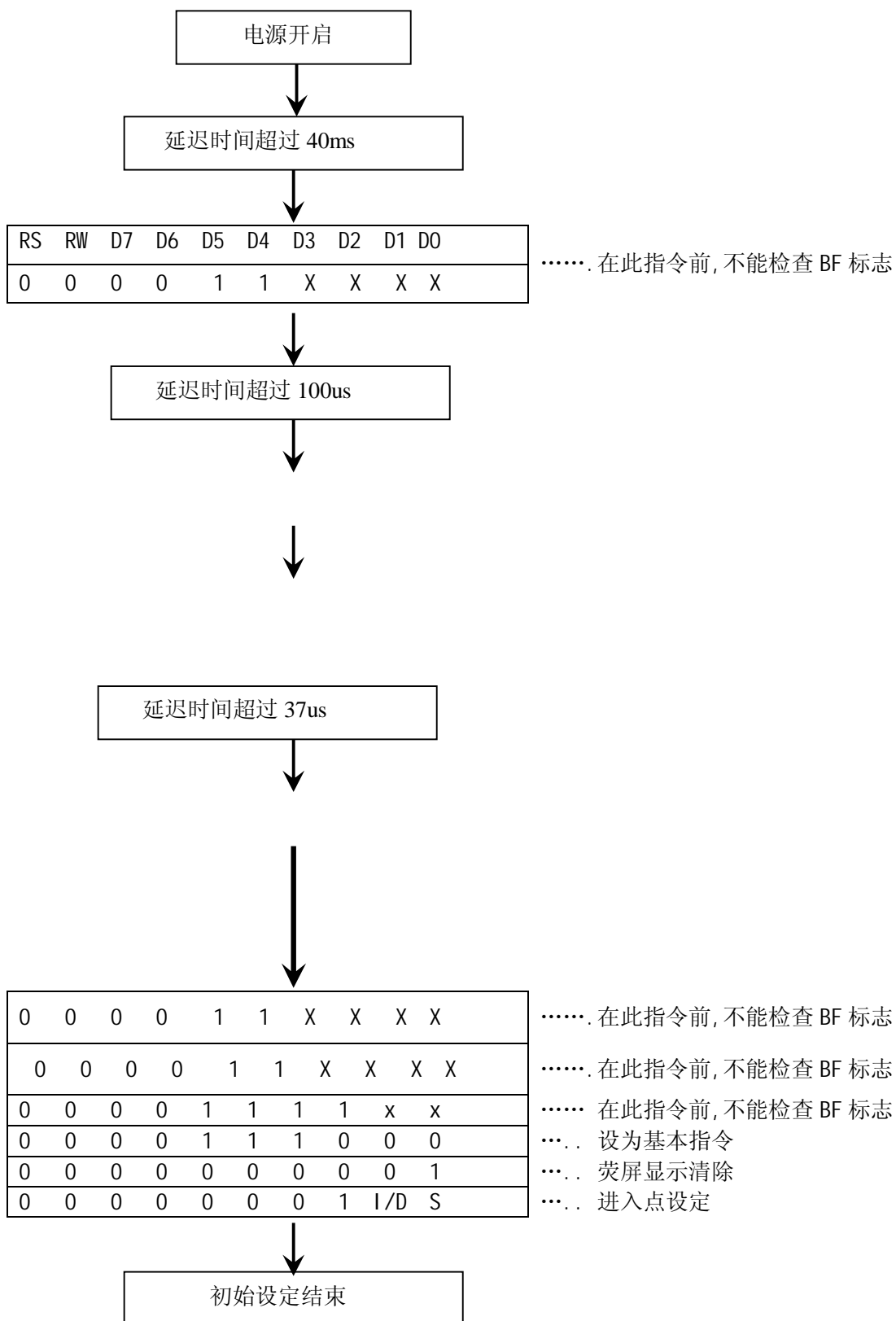
并口读写时序图:



串口读写时序:

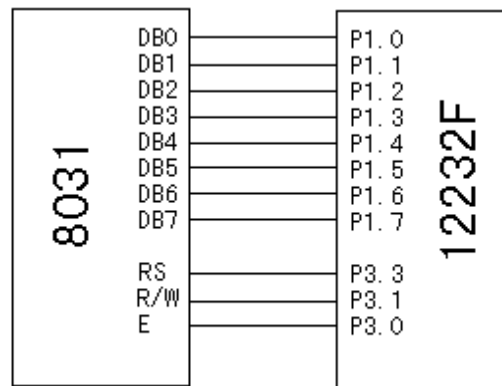


七、软件初始化:



八. 应用举例:

12232F 与 单片机 8031 的 一 种 接 口 如 图 5. 所 示



```

;This program is for 12232F
; RS-----P3.3
; R/W-----P3.1
; E-----P3.0
; DB0~7-----P1

DI      EQU P3.3
RW      EQU P3.1
E       EQU P3.0

ORG 0000H
AJMP  START
ORG 0003H
LCALL PAUSE
START:
MOV  IE, #81H      ;EXT. INTO PERMIT
MOV  IP, #01H     ;INTO IS FIRST INT. LEVEL
MOV  TCON, #00H   ;TIMER/COUNTER CONTROLER INIT.
mov  SP, #67h
LCALL DELAY
LCALL DELAY
LCALL SETUP
LCALL DEF_CHAR
MOV  A, #80H
LCALL WRITE_COM
MOV  R3, #8
TEST11:
MOV  DPTR, #CGRAM1 ;CGRAM TEST
LCALL WRITE_CGRAM
DJNZ R3, TEST11
MOV  A, #90H
    
```

```
LCALL WRITE_COM
MOV R3,#8
TEST12:
MOV DPTR,#CGRAM1
LCALL WRITE_CGRAM
DJNZ R3,TEST12
LCALL DELAY
LCALL DELAY
LCALL DELAY
LCALL DELAY
LCALL DELAY
MOV A,#80H
LCALL WRITE_COM
MOV R3,#8
TEST21:
MOV DPTR,#CGRAM2
LCALL WRITE_CGRAM
DJNZ R3,TEST21
MOV A,#90H
LCALL WRITE_COM
MOV R3,#8
TEST22:
MOV DPTR,#CGRAM2
LCALL WRITE_CGRAM
DJNZ R3,TEST22
LCALL DELAY
LCALL DELAY
LCALL DELAY
LCALL DELAY
LCALL DELAY
MOV A,#80H
LCALL WRITE_COM
MOV R3,#8
TEST31:
MOV DPTR,#CGRAM3
LCALL WRITE_CGRAM
DJNZ R3,TEST31
MOV A,#90H
LCALL WRITE_COM
MOV R3,#8
TEST32:
MOV DPTR,#CGRAM3
LCALL WRITE_CGRAM
DJNZ R3,TEST32
LCALL DELAY
LCALL DELAY
LCALL DELAY
LCALL DELAY
LCALL DELAY
MOV A,#80H
LCALL WRITE_COM
MOV R3,#8
```

```
TEST41:
    MOV  DPTR, #CGRAM4
    LCALL WRITE_CGRAM
    DJNZ R3, TEST41
    MOV  A, #90H
    LCALL WRITE_COM
    MOV  R3, #8
TEST42:
    MOV  DPTR, #CGRAM4
    LCALL WRITE_CGRAM
    DJNZ R3, TEST42
    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:    LJMP 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: ; WRIT///cv
; WRITE COMMANDS TO ST7920
LCALL DELAY1 ; INSTEAD OF CHECKING BF STATE
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

DEF_CHAR:                ;WRITE TO CGRAM
    MOV A,#01000000B    ;SET CGRAM ADDRESS
    LCALL WRITE_COM
    MOV R3,#8
DEF1:
    MOV A,#000H
    LCALL WRITE_DAT
    LCALL WRITE_DAT
    MOV A,#0FFH
    LCALL WRITE_DAT
    LCALL WRITE_DAT
    DJNZ R3,DEF1
    MOV R3,#8
DEF2:
    MOV A,#0AAH
    LCALL WRITE_DAT
    LCALL WRITE_DAT
    MOV A,#0AAH
    LCALL WRITE_DAT
    LCALL WRITE_DAT
    DJNZ R3,DEF2
    MOV R3,#8
DEF3:
    MOV A,#055H
    LCALL WRITE_DAT
    LCALL WRITE_DAT
    MOV A,#0AAH
    LCALL WRITE_DAT
    LCALL WRITE_DAT
    DJNZ R3,DEF3
    mov R3,#8
DEF4:
    MOV A,#0FFH
    LCALL WRITE_DAT
    LCALL WRITE_DAT
    LCALL WRITE_DAT
    LCALL WRITE_DAT
    DJNZ R3,DEF4
    RET
WRITE_ASCII:
    MOV R4,#16
DDDD: CLR A
    MOVC A,@A+DPTR
    LCALL WRITE_DAT
```

```
        INC DPTR
        DJNZ R4, DDDD
        RET
WRITE_HZ:          ; WRITE 8 CHINESE TO LCD
        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
PAUSE: SETB P3.2          ; PAUSE KEY PROCESS
        SETB P3.2
        LCALL DELAY1
        MOV C, P3.2
        MOV C, P3.2
        JNC PAUSE        ; CHECK KEY WAS PRESSED
PAUSE1: MOV C, P3.2
        MOV C, P3.2
        LCALL DELAY1
        JC PAUSE1       ; CHECK KEY OPEN AFTER PRESSED
PAUSE2: SETB P3.2
        SETB P3.2
        LCALL DELAY1
        MOV C, P3.2
        MOV C, P3.2
        JNC PAUSE2     ; CHECK KEY WAS PRESSED AGAIN
        RETI

TABLE1:
; “这里是 16*8 点阵的字符代码”
CGRAM1: DB 000H, 000H          ; 这里是自造字符地址表
CGRAM2: DB 000H, 002H
CGRAM3: DB 000H, 004H
CGRAM4: DB 000H, 006H
CHINESE:
; “这里是 16*16 点阵的汉字代码表”
END
```

以下为串口写指令和数据的子程序

```
WRITE_COM:
    LCALL DELAY1          ; INSTEAD OF CHECKING BF STATE
    SETB CS
    PUSH ACC
    MOV R0, #8
    MOV A, #11111000B

COMM1:
    CLR C
    RLC A
    MOV SID, C
    CLR CLK
    SETB CLK
    DJNZ R0, COMM1
    POP ACC
    MOV R5, A
    ANL A, #0FOH
    MOV R0, #8

COMM2: CLR C
    RLC A
    MOV SID, C
    CLR CLK
    SETB CLK
    DJNZ R0, COMM2
    MOV A, R5
    SWAP A
    ANL A, #0FOH
    MOV R0, #8

COMM3: CLR C
    RLC A
    MOV SID, C
    CLR CLK
    SETB CLK
    DJNZ R0, COMM3
    CLR CS

    RET

WRITE_DAT:
    LCALL DELAY1
    SETB CS
    PUSH ACC
    MOV R0, #8
    MOV A, #11111010B

DATA1: CLR C
    RLC A
    MOV SID, C
    CLR CLK
    SETB CLK
    DJNZ R0, DATA1
    POP ACC
    MOV R5, A
    ANL A, #0FOH
```

```

MOV R0, #8
DATA2: CLR C
      RLC A
      MOV SID, C
      CLR CLK
      SETB CLK
      DJNZ R0, DATA2
      MOV A, R5
      SWAP A
      ANL A, #0F0H
      MOV R0, #8
DATA3: CLR C
      RLC A
      MOV SID, C
      CLR CLK
      SETB CLK
      DJNZ R0, DATA3
      CLR CS
RET
    
```

八、附录部分

附录 1: ASCII 码表

☒	☒	☒	♥	♣	♠	♣	•	◦	◐	♂	♀	♫	♫	✂	
▶	◀	‡	!!	¶	§	—	‡	↑	↓	→	←	⊥	↕	▼	
	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
0	1	2	3	4	5	6	7	8	9	:	:	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{		}	~	△

16x8 半宽字型符号表

B400
 B410
 B420
 B430
 B440
 B450
 B460
 B470
 B480
 B490
 B500
 B510
 B520
 B530
 B540
 B550
 B560
 B570
 B580
 B590
 B600
 B610
 B620
 B630
 B640
 B650
 B660
 B670
 B680
 B690
 B700
 B710
 B720
 B730
 B740
 B750
 B760
 B770
 B780
 B790
 B800
 B810
 B820
 B830
 B840
 B850
 B860
 B870
 B880
 B890
 B900
 B910
 B920
 B930
 B940
 B950
 B960
 B970
 B980
 B990
 C000
 C010
 C020
 C030
 C040
 C050
 C060
 C070
 C080
 C090
 C100
 C110
 C120
 C130
 C140
 C150
 C160
 C170
 C180
 C190
 C200
 C210
 C220
 C230
 C240
 C250
 C260
 C270
 C280
 C290
 C300
 C310
 C320
 C330
 C340

C350
 C360
 C370
 C380
 C390
 C400
 C410
 C420
 C430
 C440
 C450
 C460
 C470
 C480
 C490
 C500
 C510
 C520
 C530
 C540
 C550
 C560
 C570
 C580
 C590
 C600
 C610
 C620
 C630
 C640
 C650
 C660
 C670
 C680
 C690
 C700
 C710
 C720
 C730
 C740
 C750
 C760
 C770
 C780
 C790
 C800
 C810
 C820
 C830
 C840
 C850
 C860
 C870
 C880
 C890
 C900
 C910
 C920
 C930
 C940
 C950
 C960
 C970
 C980
 C990
 D000
 D010
 D020
 D030
 D040
 D050
 D060
 D070
 D080
 D090
 D100
 D110
 D120
 D130
 D140
 D150
 D160
 D170
 D180
 D190
 D200
 D210
 D220
 D230
 D240
 D250
 D260
 D270
 D280
 D290
 D300
 D310
 D320
 D330
 D340

0260 九上清草勿 0270 司松因外嫩 0280 邪敢个松叙 0290 早才新燕伴 0300 野核功亦吉 0310 影八有公重 0320 区云平胜云 0330 榜旋扣翰舞 0340 愚操打官全 0350 愚操打官全 0360 愚操打官全 0370 愚操打官全 0380 愚操打官全 0390 愚操打官全 0400 愚操打官全 0410 愚操打官全 0420 愚操打官全 0430 愚操打官全 0440 愚操打官全 0450 愚操打官全 0460 愚操打官全 0470 愚操打官全 0480 愚操打官全 0490 愚操打官全 0500 愚操打官全 0510 愚操打官全 0520 愚操打官全 0530 愚操打官全 0540 愚操打官全 0550 愚操打官全 0560 愚操打官全 0570 愚操打官全 0580 愚操打官全 0590 愚操打官全 0600 愚操打官全 0610 愚操打官全 0620 愚操打官全 0630 愚操打官全 0640 愚操打官全 0650 愚操打官全 0660 愚操打官全 0670 愚操打官全 0680 愚操打官全 0690 愚操打官全 0700 愚操打官全 0710 愚操打官全 0720 愚操打官全

0730 愚操打官全 0740 愚操打官全 0750 愚操打官全 0760 愚操打官全 0770 愚操打官全 0780 愚操打官全 0790 愚操打官全 0800 愚操打官全 0810 愚操打官全 0820 愚操打官全 0830 愚操打官全 0840 愚操打官全 0850 愚操打官全 0860 愚操打官全 0870 愚操打官全 0880 愚操打官全 0890 愚操打官全 0900 愚操打官全 0910 愚操打官全 0920 愚操打官全 0930 愚操打官全 0940 愚操打官全 0950 愚操打官全 0960 愚操打官全 0970 愚操打官全 0980 愚操打官全 0990 愚操打官全 1000 愚操打官全

E110
 E120
 E130
 E140
 E150
 E160
 E170
 E180
 E190
 E200
 E210
 E220
 E230
 E240
 E250
 E260
 E270
 E280
 E290
 E300
 E310
 E320
 E330
 E340
 E350
 E360
 E370
 E380
 E390
 E400
 E410
 E420
 E430
 E440
 E450
 E460
 E470
 E480
 E490
 E500
 E510
 E520
 E530
 E540
 E550
 E560
 E570
 E580
 E590
 E600
 E610
 E620
 E630
 E640
 E650
 E660
 E670
 E680
 E690
 E700
 E710
 E720
 E730
 E740
 E750
 E760
 E770
 E780
 E790
 E800
 E810
 E820
 E830
 E840
 E850
 E860
 E870
 E880
 E890
 E900
 E910
 E920
 E930
 E940
 E950
 E960
 E970
 E980
 E990
 E1000

E500
 E510
 E520
 E530
 E540
 E550
 E560
 E570
 E580
 E590
 E600
 E610
 E620
 E630
 E640
 E650
 E660
 E670
 E680
 E690
 E700
 E710
 E720
 E730
 E740
 E750
 E760
 E770
 E780
 E790
 E800
 E810
 E820
 E830
 E840
 E850
 E860
 E870
 E880
 E890
 E900
 E910
 E920
 E930
 E940
 E950
 E960
 E970
 E980
 E990
 E1000

F5C0 趵 趵 趵 趵 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮
 F5D0 趼 趼 趼 趼 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮
 F5E0 踵 踵 踵 踵 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮
 F5F0 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮 踮
 F6A0 躅 躅 躅 躅 躅 躅 躅 躅 躅 躅 躅 躅 躅 躅 躅 躅 躅 躅
 F6B0 霫 霫 霫 霫 霫 霫 霫 霫 霫 霫 霫 霫 霫 霫 霫 霫 霫 霫
 F6C0 隼 隼 隼 隼 隼 隼 隼 隼 隼 隼 隼 隼 隼 隼 隼 隼 隼 隼
 F6D0 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇
 F6E0 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇 魇
 F6F0 鲙 鲙 鲙 鲙 鲙 鲙 鲙 鲙 鲙 鲙 鲙 鲙 鲙 鲙 鲙 鲙 鲙 鲙
 F7A0 鳌 鳌 鳌 鳌 鳌 鳌 鳌 鳌 鳌 鳌 鳌 鳌 鳌 鳌 鳌 鳌 鳌 鳌
 F7B0 鞞 鞞 鞞 鞞 鞞 鞞 鞞 鞞 鞞 鞞 鞞 鞞 鞞 鞞 鞞 鞞 鞞 鞞
 F7C0 骼 骼 骼 骼 骼 骼 骼 骼 骼 骼 骼 骼 骼 骼 骼 骼 骼 骼
 F7D0 膻 膻 膻 膻 膻 膻 膻 膻 膻 膻 膻 膻 膻 膻 膻 膻 膻 膻
 F7E0 鬣 鬣 鬣 鬣 鬣 鬣 鬣 鬣 鬣 鬣 鬣 鬣 鬣 鬣 鬣 鬣 鬣 鬣
 F7F0 黠 黠 黠 黠 黠 黠 黠 黠 黠 黠 黠 黠 黠 黠 黠 黠 黠 黠