
12864C-1 液晶中文显示模块

(一) 概述	2
(一) (二) 外形尺寸	
1 方框图	2
2 外型尺寸图	3
(二) (三) 模块的接口	4
(三) (四) 硬件说明	5
(五) 指令说明	6
(四) (五) 读写操作时序	7
(五) (六) 交流参数	9
(六) (七) 软件初始化过程	11
(七) (八) 应用举例	12
(八) (九) 附录	
1 半宽字符表	20
2 汉字字符表	21

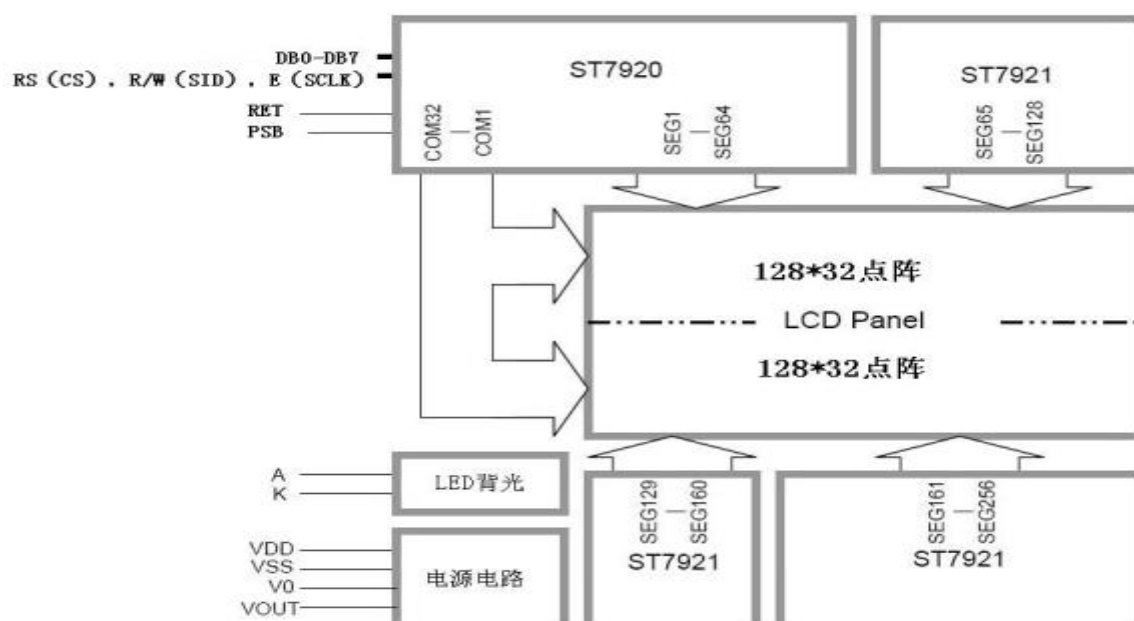
一、概述

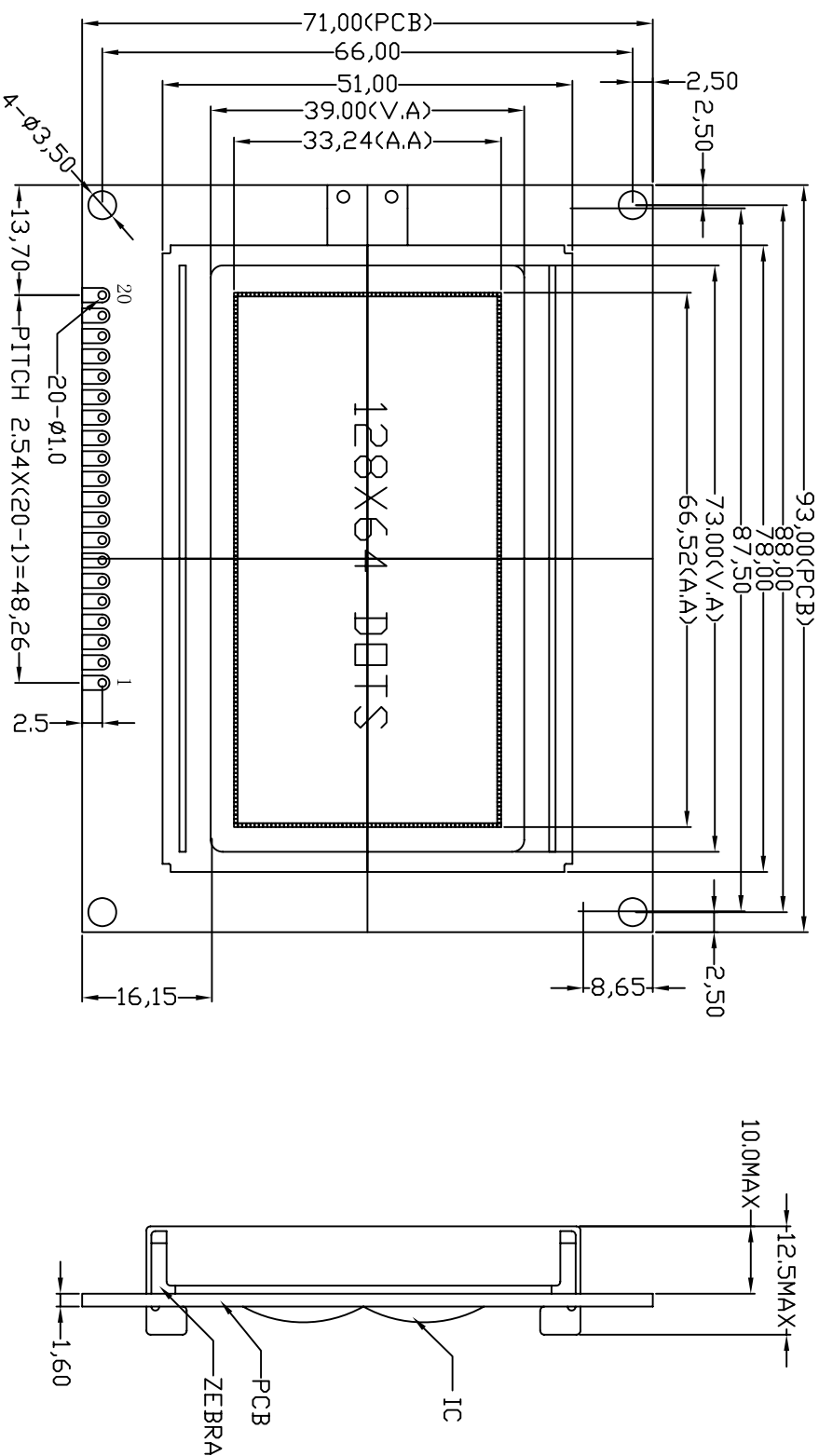
12864C-1 是一种具有 4 位/8 位并行、2 线或 3 线串行多种接口方式，内部含有国标一级、二级简体中文字库的点阵图形液晶显示模块；其显示分辨率为 128×64 ，内置 8192 个 16×16 点阵汉字，和 128 个 16×8 点 ASCII 字符集。利用该模块灵活的接口方式和简单、方便的操作指令，可构成全中文人机交互图形界面。可以显示 8×4 行 16×16 点阵的汉字，也可完成图形显示。低电压低功耗是其又一显著特点。由该模块构成的液晶显示方案与同类型的图形点阵液晶显示模块相比，不论硬件电路结构或显示程序都要简洁得多，且该模块的价格也略低于相同点阵的图形液晶模块。

基本特性：

- ● 低电源电压（VDD: +3.0—+5.5V），
- ● 显示分辨率: 128×64 点
- ● 内置汉字字库，提供 8192 个 16×16 点阵汉字(简繁体可选)
- ● 内置 128 个 16×8 点阵字符
- ● 2MHz 时钟频率
- ● 显示方式：STN、全透、蓝屏负显
- ● 驱动方式：1/32DUTY，1/5BIAS
- ● 视角方向：6 点
- ● 背光方式：侧部白色 LED，
- ● 通讯方式：串行、并口可选
- ● 内置 DC-DC 转换电路，无需外加负压
- ● 无需片选信号，简化软件设计
- ● 工作温度： $0^{\circ}\text{C} - +55^{\circ}\text{C}$ ，存储温度： $-20^{\circ}\text{C} - +60^{\circ}\text{C}$

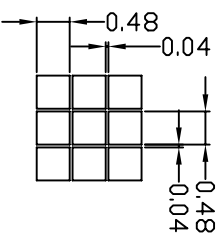
二、方框图





NOTES:

1. DISPLAY TYPE: STN/Negative BLUE
2. DISPLAY MODE: Transmissive
3. VIEWING DIRECTION: 6: 00
4. OPERATING VOL: 5.0V
5. OPERATING TEMPERATURE: -20° C~70° C
6. STORAGE TEMPERATURE: -30° C~80° C
7. DRIVING METHOD: 1/32 DUTY 1/6 BIAS
8. Drive IC: ST7920-0A
9. CONNECTOR: ZEBRA
9. BACKLIGHT: White, 15*4mA@5.0V




1	2	3	4	5	6	7	8	9	10
GND	VDD	VO/NC	RS	WR	E	DB0	DB1	DB2	DB3
11	12	13	14	15	16	17	18	19	20
DB4	DB5	DB6	DB7	PSB	NC	RST	VO _{UT}	LED+	LED-

接口定义

1	2	3	4	5	6	7	8	9	10
GND	VDD	VO/NC	RS	WR	E	DB0	DB1	DB2	DB3
11	12	13	14	15	16	17	18	19	20
DB4	DB5	DB6	DB7	PSB	NC	RST	VO _{UT}	LED+	LED-

NO.	REV.	REVISION RECORD	DATE	NO.	REV.	REVISION RECORD	DATE
1	00	新图	2014/3/11	5			
2				6			
3				7			
4				8			

CUSTOMER REF.	PROJECTION	APPROVAL	DRAFTING	ENGINEERING
SHEET 1 OF 1	 <p>PROJECTION</p>	APPROVED BY	L Y I	
		DATE	2014/3/11	
		DRAWN BY :	WANG	UNIT IN mm

12864C-1结构图
 比例 1:1

三、模块接口说明

2.1 串口接口管脚信号			
管脚号	名称	LEVEL	功能
1	VSS	0V	电源地
2	VDD	+5V	电源正 (3.0V——5.5V)
3	V0	-	对比度（亮度）调整
4	CS	H/L	模组片选端，高电平有效
5	SID	H/L	串行数据输入端
6	CLK	H/L	串行同步时钟：上升沿时读取 SID 数据
15	PSB	L	L：串口方式（见注释 1）
17	/RESET	H/L	复位端，低电平有效（见注释 2）
19	A	VDD	背光源电压+5V（见注释 3）
20	K	VSS	背光源负端 0V （见注释 3）

2.2 并行接口

管脚号	管脚名称	电平	管脚功能描述
1	VSS	0V	电源地
2	VCC	3.0+5V	电源正
3	V0 (NC)	-	对比度（亮度）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 (SCLK)	H/L	使能信号
7	DB0	H/L	三态数据线
8	DB1	H/L	三态数据线
9	DB2	H/L	三态数据线
10	DB3	H/L	三态数据线
11	DB4	H/L	三态数据线
12	DB5	H/L	三态数据线
13	DB6	H/L	三态数据线
14	DB7	H/L	三态数据线
15	PSB	H/L	H：8 位或 4 位并口方式，L：串口方式（见注释 1）
16	NC	-	空脚
17	/RESET	H/L	复位端，低电平有效（见注释 2）
18	VOUT (NC)	-	LCD 驱动电压输出端

19	A	VDD	背光源正端 (+5V)
20	K	VSS	背光源负端

* 四. 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, 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, 分别是半角英数字型 (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 0000H—0006H 的编码中 (其代码分别是 0000、0002、0004、0006 共 4 个) 将选择 CGRAM 的自定义字型, 02H—7FH 的编码中将选择半角英数字的字型, 至于 A1 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 BIG5 (A140—D75F), GB (A1A0—F7FFH)。

● 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 中的定义一样地通过 DDRAM 显示在屏幕中。

● 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。

● 光标/闪烁控制电路

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

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）

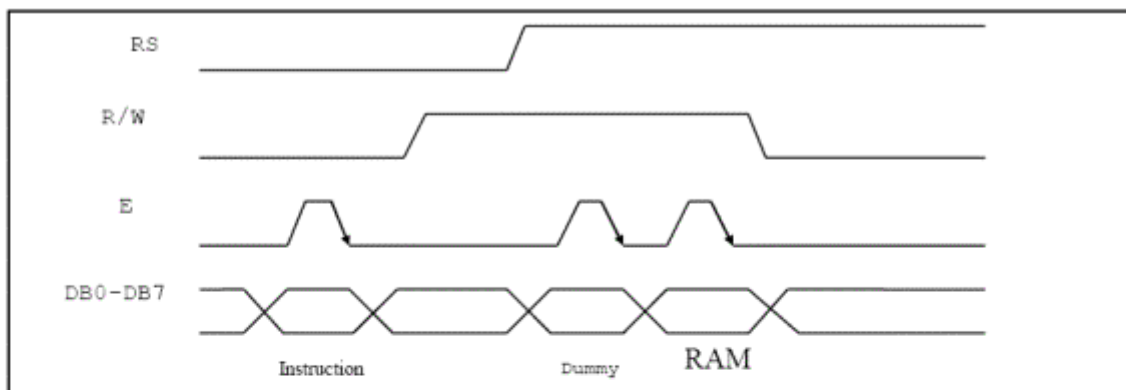
指令	指令码										功能
	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	
待命模式	0	0	0	0	0	0	0	0	0	1	进入待命模式，执行其他指令都裸终止待命模式

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 AC6	0 AC5	0 AC4	AC3 AC3	AC2 AC2	AC1 AC1	AC0 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

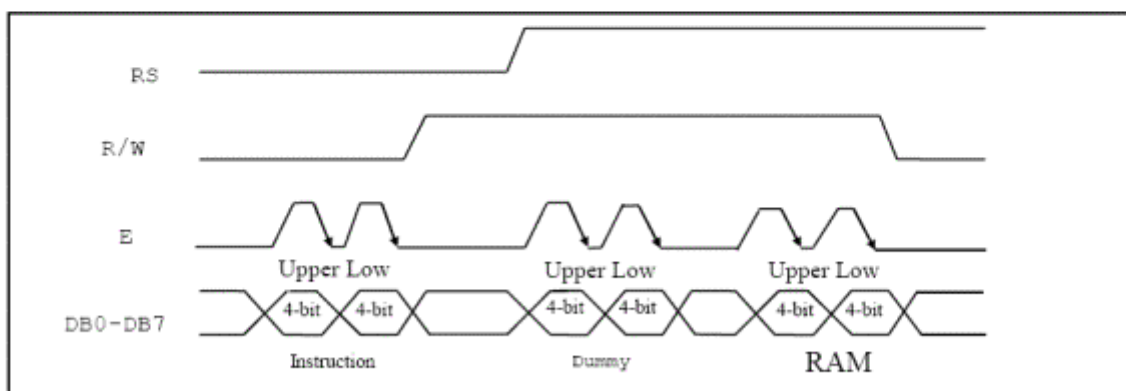
备注:当 IC1 在接受指令前,微处理器必须先确认其内部处于非忙碌状态,即读取 BF 标志时,BF 需为零,方可接受新的指令;如果在送出一个指令前并不检查 BF 标志,那么在前一个指令和这个指令中间必须延长一段较长的时间,即是等待前一个指令确实执行完成。

六、读写时序图

6.1 数据传输过程

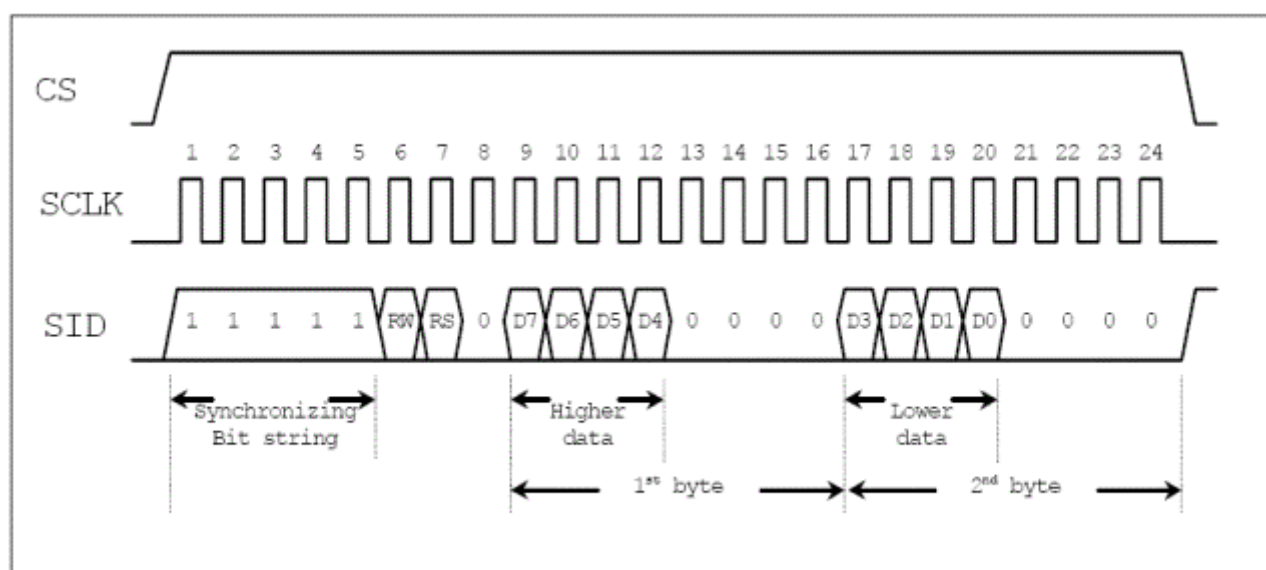


Timing Diagram of 8-bit Parallel Bus Mode Data Transfer



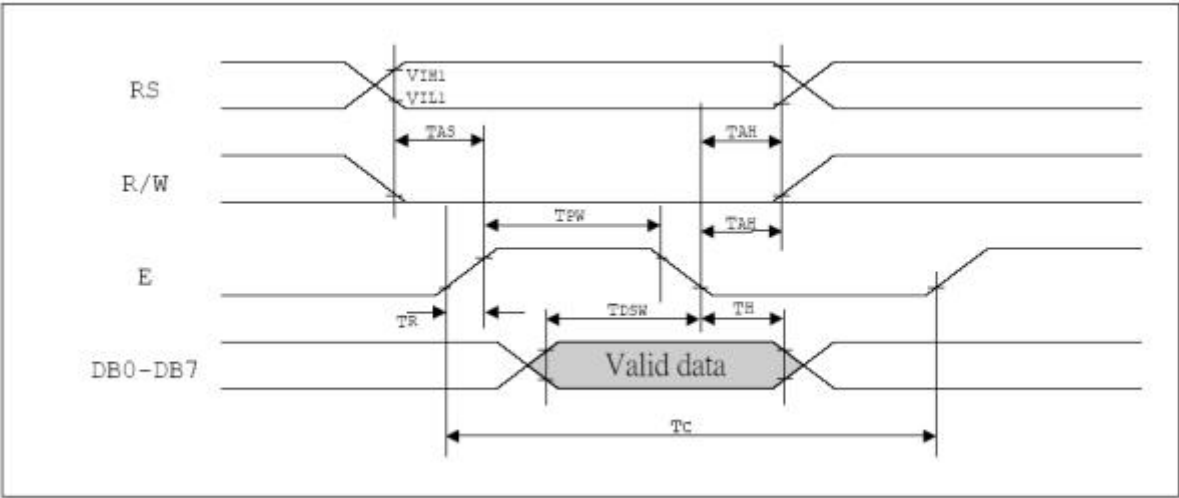
Timing Diagram of 4-bit Parallel Bus Mode Data Transfer

8 位和 4 位数据线的传输过程

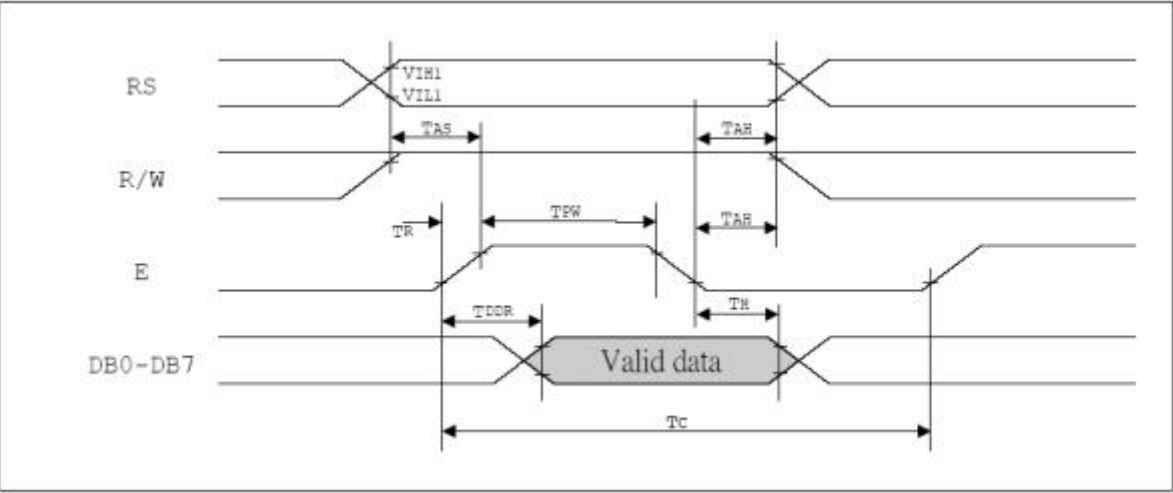


串口数据线模式数据传输过程

6.2 时序图

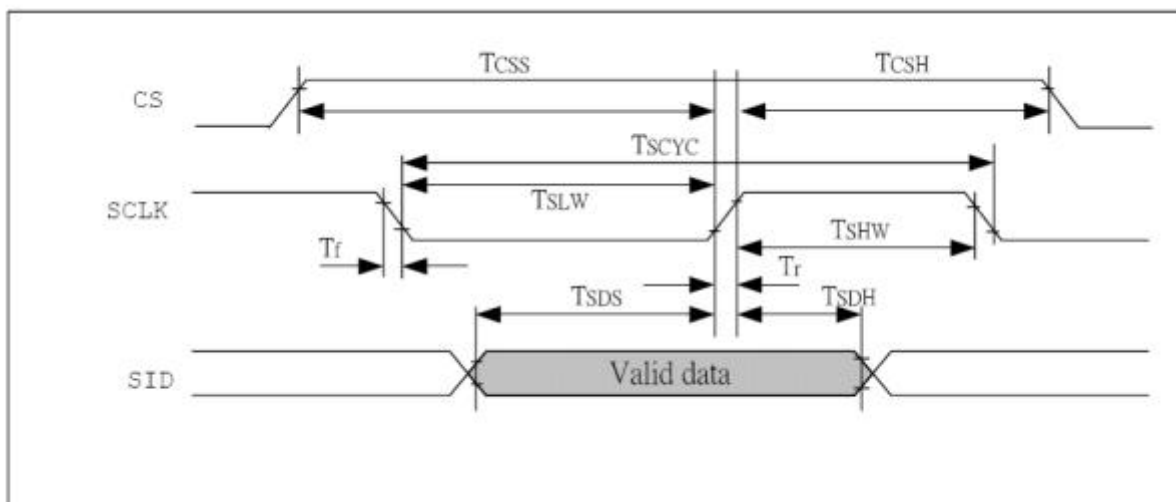


MPU 写资料到 ST7920（8 位数据线模式）



MPU 从 ST7920 读资料（8 位数据线模式）

6.3 串口读写时序：



串口方式下 MPU 写数据到 ST7920

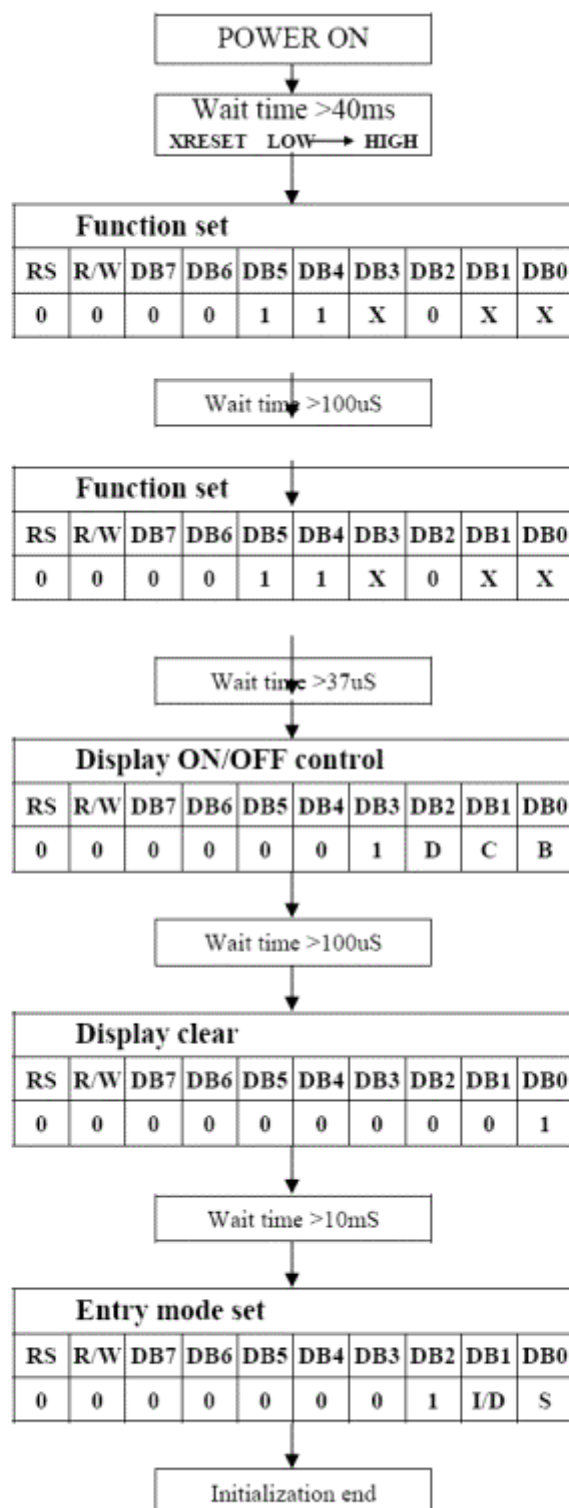
七、交流参数

Symbol	Characteristics	Test Condition	Min.	Typ.	Max.	Unit
<i>Internal Clock Operation</i>						
f _{OSC}	OSC Frequency	R = 33K Ω	480	540	600	KHz
<i>External Clock Operation</i>						
f _{EX}	External Frequency	-	480	540	600	KHz
	Duty Cycle	-	45	50	55	%
T _R ,T _F	Rise/Fall Time	-	-	-	0.2	μ s
<i>Write Mode (Writing data from MPU to ST7920)</i>						
T _C	Enable Cycle Time	Pin E	1200	-	-	ns
T _{PW}	Enable Pulse Width	Pin E	140	-	-	ns
T _R ,T _F	Enable Rise/Fall Time	Pin E	-	-	25	ns
T _{AS}	Address Setup Time	Pins: RS,RW,E	10	-	-	ns
T _{AH}	Address Hold Time	Pins: RS,RW,E	20	-	-	ns
T _{DSW}	Data Setup Time	Pins: DB0 - DB7	40	-	-	ns
T _H	Data Hold Time	Pins: DB0 - DB7	20	-	-	ns
<i>Read Mode (Reading Data from ST7920 to MPU)</i>						
T _C	Enable Cycle Time	Pin E	1200	-	-	ns
T _{PW}	Enable Pulse Width	Pin E	140	-	-	ns
T _R ,T _F	Enable Rise/Fall Time	Pin E	-	-	25	ns
T _{AS}	Address Setup Time	Pins: RS,RW,E	10	-	-	ns
T _{AH}	Address Hold Time	Pins: RS,RW,E	20	-	-	ns
T _{DDR}	Data Delay Time	Pins: DB0 - DB7	-	-	100	ns
T _H	Data Hold Time	Pins: DB0 - DB7	20	-	-	ns
<i>Interface Mode with LCD Driver(ST7921)</i>						
T _{CWH}	Clock Pulse with High	Pins: CL1, CL2	800	-	-	ns
T _{CWL}	Clock Pulse with Low	Pins: CL1, CL2	800	-	-	ns
T _{CST}	Clock Setup Time	Pins: CL1, CL2	500	-	-	ns
T _{SU}	Data Setup Time	Pin: D	300	-	-	ns
T _{DH}	Data Hold Time	Pin: D	300	-	-	ns
T _{DM}	M Delay Time	Pin: M	-1000	-	1000	ns

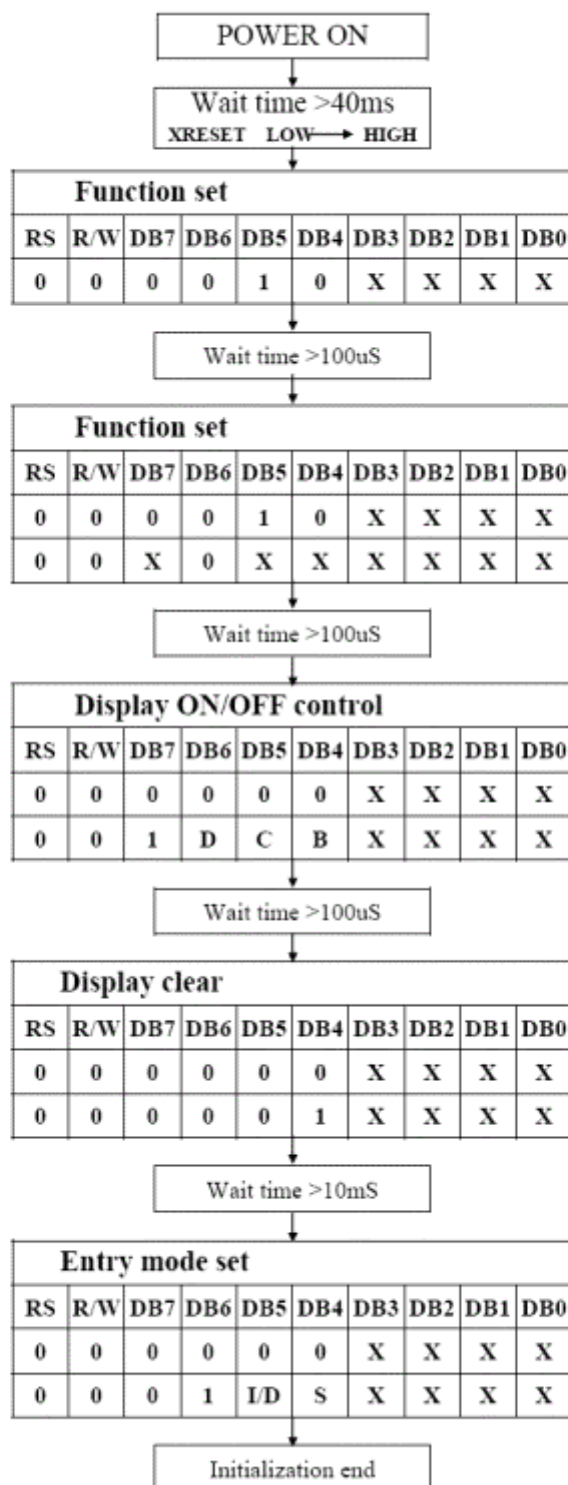
交流参数 (T_A = 25°C, V_{DD} = 4.5 V - 5.5 V)

八、软件初始化

8-位元介面：



4-位元介面：



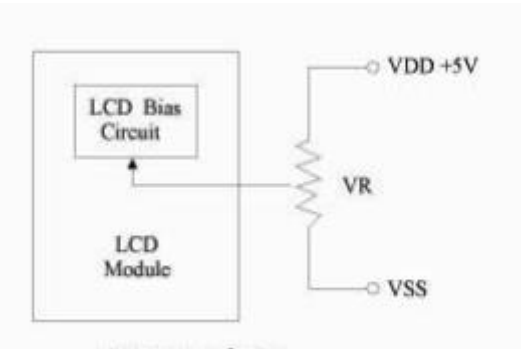
九. 应用举例：

1 、使用前的准备

先给模块加上工作电压，再按照下图的连接方法调节 LCD 的对比度，使其显示出黑色的底影。
此过程亦可以初步检测 LCD 有无缺段现象。

2、字符显示

12864C-1 每屏可显示 4 行 8 列共 32 个 16×16 点阵的汉字，每个显示 RAM 可显示 1 个中



文字符或 2 个 16×8 点阵全高 ASCII 码字符，即每屏最多可实现 32 个中文字符或 64 个 ASCII 码字符的显示。12864C-1 内部提供 128×2 字节的字符显示 RAM 缓冲区（DDRAM）。字符显示是通过将字符显示编码写入该字符显示 RAM 实现的。根据写入内容的不同，可分别在液晶屏上显示 CGROM（中文字库）、HCGROM（ASCII 码字库）及 CGRAM（自定义字形）的内容。三种不同字符/字型的选择编码范围为：0000~0006H（其代码分别是 0000、0002、0004、0006 共 4 个）显示自定义字型，02H~7FH 显示半宽 ASCII 码字符，A1A0H~F7FFH 显示 8192 种 GB2312 中文字库字形。字符显示 RAM 在液晶模块中的地址 80H~9FH。字符显示的 RAM 的地址与 32 个字符显示区域有着——对应的关系，其对应关系如下表所示。

80H	81H	82H	83H	84H	85H	86H	87H
90H	91H	92H	93H	94H	95H	96H	97H
88H	89H	8AH	8BH	8CH	8DH	8EH	8FH
98H	99H	9AH	9BH	9CH	9DH	9EH	9FH

3 、图形显示

先设垂直地址再设水平地址(连续写入两个字节的资料来完成垂直与水平的坐标地址)

垂直地址范围 AC5...AC0

水平地址范围 AC3...AC0

绘图 RAM 的地址计数器（AC）只会对水平地址(X 轴)自动加一,当水平地址=0FH 时会重新设为 00H 但并不会对垂直地址做进位自动加一，故当连续写入多笔资料时，程序需自行判断垂直地址是否需重新设定。**GDRAM** 的坐标地址与资料排列顺序如下图

		水平坐标				
		00	01	~	06	07
		D15 ~ D0	D15 ~ D0	~	D15 ~ D0	D15 ~ D0
垂直坐标	00					
	01					
	⋮					
	⋮					
	1E					
	1F					
	⋮					
	⋮					
	00					
	01					
	⋮					
	⋮					
	1E					
	1F					
		D15 ~ D0	D15 ~ D0	~	D15 ~ D0	D15 ~ D0
		08	09	~	0E	0F

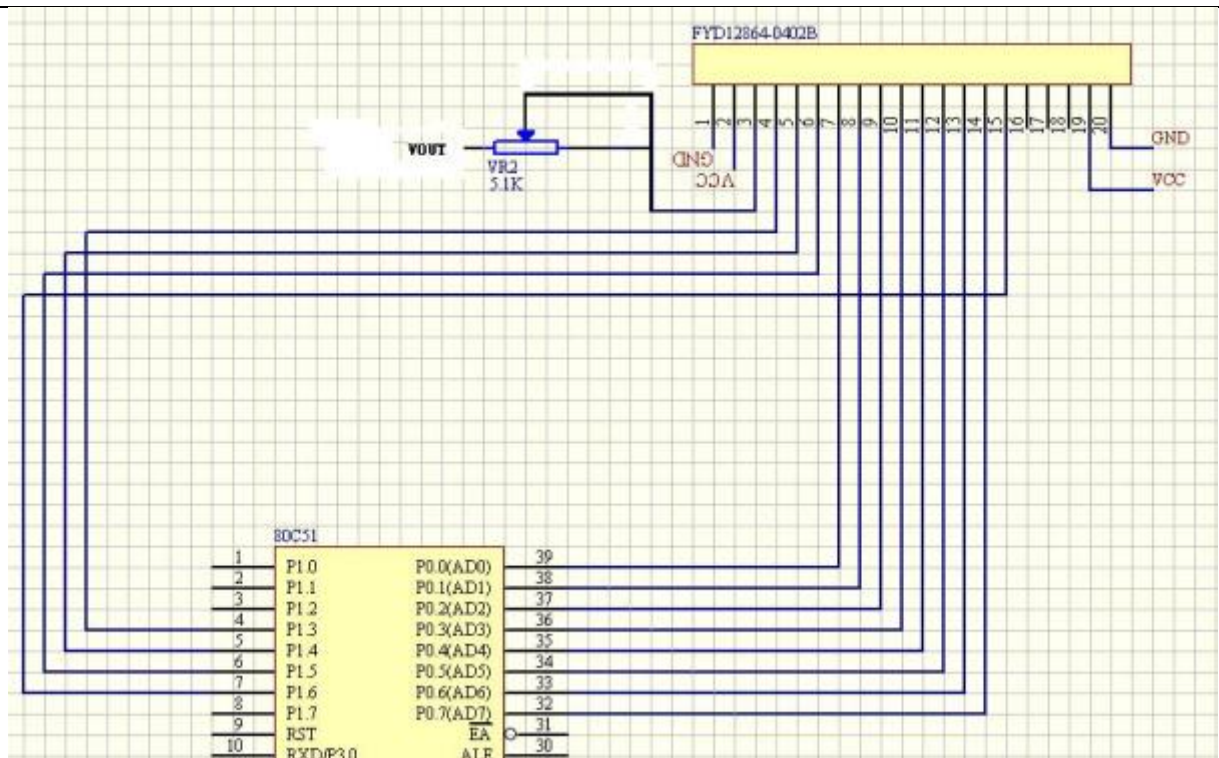
128*64点

3、应用说明

用 12864C-1 显示模块时应注意以下几点：

- ①欲在某一个位置显示中文字符时，应先设定显示字符位置，即先设定显示地址，再写入中文字符编码。
- ②显示 ASCII 字符过程与显示中文字符过程相同。不过在显示连续字符时，只须设定一次显示地址，由模块自动对地址加 1 指向下一个字符位置，否则，显示的字符中将会有有一个空 ASCII 字符位置。
- ③当字符编码为 2 字节时，应先写入高位字节，再写入低位字节。
- ④模块在接收指令前，向处理器必须先确认模块内部处于非忙状态，即读取 BF 标志时 BF 需为“0”，方可接受新的指令。如果在送出一个指令前不检查 BF 标志，则在前一个指令和这个指令中间必须延迟一段较长的时间，即等待前一个指令确定执行完成。指令执行的时间请参考指令表中的指令执行时间说明。
- ⑤“RE”为基本指令集与扩充指令集的选择控制位。当变更“RE”后，以后的指令集将维持在最后的状态，除非再次变更“RE”位，否则使用相同指令集时，无需每次均重设“RE”位。

4、4、12864C-1 与单片机 80C51 的一种接口如图所示



5、5、 编程参考（部分）

```

;*****
;Controller:ST7920
;MCU:AT89C52 ,晶体频率：12MHz
;LCM:128*64
;LCM 型号:12864C-1
;LCM 接口:1:GND 2:VCC 3:VO 4:RS 5:RW 6:E 7--14:DB0-DB7 15:PSB 16:NC 17:RST 18:Vout
;*****

```

```

RS      EQU    P1.3
RW      EQU    P1.4
E       EQU    P1.5
PSB     EQU    P1.6
RST     EQU    P2.7
COM     EQU    20H
DAT     EQU    21H

```

*****以上是接口定义

START:

```

ORG 0000H
MOV SP, #60H
LJMP DISP

```

*****主程序入口地址

```

ORG 0040H
DISP: CLR    PSB
      LCALL  DELAY
      SETB   PSB           ;将 PSB 置 1，通信方式为 8 位数据并口
      LCALL  DELAY

      MOV    COM, #30H      ;功能设置---8BIT 控制界面，基本指令集
      ACALL  WRI           ;调用写指令子程序
      LCALL  DELAY         ;延迟 39uS

      MOV    COM, #0CH      ;显示打开，光标关，反白显示关
      ACALL  WRI           ;调用写指令子程序

```

```

        LCALL    DELAY                ;延迟 39uS

        MOV     COM, #01H            ;清除屏幕显示，将 DDRAM 的地址计数器归零
        ACALL    WRI                ;调用写指令子程序
        LCALL    DELAY                ;延迟 39uS

        MOV     COM, #06H            ;DDRAM 的地址计数器（AC）加 1
        ACALL    WRI                ;调用写指令子程序
        LCALL    DELAY                ;延迟 39uS
;*****
DISPLAY5A:
        MOV     R2, #64              ;DDRAM 地址计数器设为最大值
        MOV     R1, #10H
        MOV     COM, #80H            ;DD RAM 地址-----0000000
        ACALL    WRI
DISP5A:  MOV     DAT, R1
        ACALL    WRD                ;写入第二字节数据
        INC     R1
        DJNZ    R2, DISP5A
        LCALL    DELAY1              ;1S 延时子程序
;*****以上是写入半角字符
DISPLAY5:
        MOV     R2, #64              ;DDRAM 地址计数器设为最大值
        MOV     R1, #0DEH
        MOV     COM, #80H            ;DD RAM 地址-----0000000
        ACALL    WRI
DISP5:  MOV     DAT, #0B4H
        ACALL    WRD                ;写入第一字节数据
        MOV     DAT, R1
        ACALL    WRD                ;写入第二字节数据
        INC     R1
        DJNZ    R2, DISP5
        LCALL    DELAY1              ;1S 延时子程序
;*****以上是字库中“崔”字处开始显示
        MOV     COM, #01H            ;清屏
        ACALL    WRI                ;调用写指令子程序
        LCALL    DELAY

        MOV     COM, #32H            ;功能设置---8BIT 控制界面，绘图显示 ON
        ACALL    WRI                ;调用写指令子程序
        LCALL    DELAY                ;延迟 39uS

        MOV     COM, #36H            ;功能设置---8BIT 控制界面，扩充指令集
        ACALL    WRI                ;调用写指令子程序
        LCALL    DELAY                ;延迟 39uS

DISPLAY6:
        MOV     DPTR, #DAB2          ;图形数据入口地址
        MOV     R2, #32              ;32 行，(双屏结构中上半屏)
        MOV     R3, #80H            ;Y 地址寄存器
DISP6:  MOV     COM, R3              ;设置绘图区的 Y 地址坐标
        INC     R3                  ;Y 地址加 1
        LCALL    WRI
        MOV     COM, #80H            ;设置绘图区的 X 地址坐标
        LCALL    WRI
        MOV     R1, #16              ;16*8 列
DISP7:  CLR     A

```

```

        MOV     A, @A+DPTR
        MOV     DAT, A
        LCALL   WRD
        INC     DPTR
        DJNZ    R1, DISP7
        DJNZ    R2, DISP6      ;写满全屏的 16*8 字节 X64

        MOV     R2, #32        ;32 行, (双屏结构的下半屏)
        MOV     R3, #80H       ;Y 地址寄存器
DISP8:   MOV     COM, R3        ;设置绘图区的 Y 地址坐标
        INC     R3             ;Y 地址加 1
        LCALL   WRI
        MOV     COM, #88H      ;设置绘图区的 X 地址坐标
        LCALL   WRI
        MOV     R1, #16        ;16*8 列
DISP9:   CLR     A
        MOV     A, @A+DPTR
        MOV     DAT, A
        LCALL   WRD
        INC     DPTR
        DJNZ    R1, DISP9
        DJNZ    R2, DISP8      ;写满全屏的 16*8 字节 X64
        LCALL   DELAY1        ;1S 延时子程序

```

;*****以上是写入一幅图画(电脑桌面图)

```

DISPLAY6A:
        MOV     DPTR, #DAB3    ;图形数据入口地址
        MOV     R2, #32        ;32 行, (双屏结构中上半屏)
        MOV     R3, #80H       ;Y 地址寄存器
DISP6A:  MOV     COM, R3        ;设置绘图区的 Y 地址坐标
        INC     R3             ;Y 地址加 1
        LCALL   WRI
        MOV     COM, #80H      ;设置绘图区的 X 地址坐标
        LCALL   WRI
        MOV     R1, #16        ;16*8 列
DISP7A:  CLR     A
        MOV     A, @A+DPTR
        MOV     DAT, A
        LCALL   WRD
        INC     DPTR
        DJNZ    R1, DISP7A
        DJNZ    R2, DISP6A      ;写满全屏的 16*8 字节 X64

        MOV     R2, #32        ;32 行, (双屏结构的下半屏)
        MOV     R3, #80H       ;Y 地址寄存器
DISP8A:  MOV     COM, R3        ;设置绘图区的 Y 地址坐标
        INC     R3             ;Y 地址加 1
        LCALL   WRI
        MOV     COM, #88H      ;设置绘图区的 X 地址坐标
        LCALL   WRI
        MOV     R1, #16        ;16*8 列
DISP9A:  CLR     A
        MOV     A, @A+DPTR
        MOV     DAT, A

```

```

LCALL  WRD
INC    DPTR
DJNZ   R1, DISP9A
DJNZ   R2, DISP8A    ;写满全屏的 16*8 字节 X64
LCALL  DELAY1        ;1S 延时子程序
LJMP   START         ;从开始处循环显示

```

;*****以上是写入一幅图画(电子制作实验室)

```

WRI:    PUSH    ACC
        CLR     RS
        SETB    RW
WRI1:   MOV     PO, #0FFH
        SETB    E
        MOV     A, P0
        CLR     E
        JB      ACC. 7, WRI1
        CLR     RW
        MOV     PO, COM
        SETB    E
        CLR     E
        POP     ACC
        RET

```

;*****以上是写指令子程序

```

WRD:    PUSH    ACC
        CLR     RS
        SETB    RW
WRD1:   MOV     PO, #0FFH
        SETB    E
        MOV     A, P0
        CLR     E
        JB      ACC. 7, WRD1
        SETB    RS
        CLR     RW
        MOV     PO, DAT
        SETB    E
        CLR     E
        POP     ACC
        RET

```

;*****以上是写数据子程序

```

DELAY:  MOV     R7, #0FFH
DELAY4: MOV     R6, #00FH
LOOP2:  DJNZ    R6, LOOP2
        DJNZ    R7, DELAY4
        RET
DELAY1: MOV     R4, #08H
DELAY11:MOV     R7, #0FFH
DELAY41:MOV     R6, #0FFH
        DJNZ    R6, $
        DJNZ    R7, DELAY41
        DJNZ    R4, DELAY11
        RET
XX:     MOV     R5, #0FFH
XX1:    MOV     R6, #0FFH
XX2:    DJNZ    R6, XX2

```

DJNZ R5, XX1

RET

;*****以上是延时子程序

DAB2:

```
DB 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 03FH, 0FFH, 0F0H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 040H, 000H, 018H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 0BFH, 0FFH, 0DCH, 000H, 001H
DB 080H, 000H, 01FH, 0FFH, 0FFH, 000H, 000H, 000H, 000H, 000H, 000H, 0BFH, 0FFH, 0BCH, 000H, 001H
DB 080H, 000H, 030H, 000H, 001H, 080H, 000H, 000H, 000H, 000H, 001H, 07FH, 0FFH, 0B8H, 000H, 001H
DB 080H, 000H, 030H, 000H, 000H, 080H, 000H, 000H, 000H, 000H, 001H, 07FH, 0FFH, 078H, 000H, 001H
DB 080H, 000H, 019H, 0FFH, 0FEH, 0C0H, 000H, 000H, 000H, 000H, 002H, 0FFH, 0FFH, 070H, 000H, 001H
DB 080H, 000H, 018H, 000H, 000H, 040H, 000H, 000H, 000H, 000H, 002H, 0FFH, 0FEH, 0F0H, 000H, 001H
DB 080H, 000H, 00CH, 001H, 0FFH, 060H, 000H, 000H, 000H, 000H, 005H, 0FFH, 0FEH, 0E0H, 000H, 001H
DB 080H, 000H, 00CH, 07FH, 0D0H, 020H, 000H, 000H, 000H, 000H, 005H, 0FFH, 0FDH, 0E0H, 000H, 001H
DB 080H, 000H, 006H, 000H, 00FH, 0B0H, 000H, 000H, 000H, 000H, 00BH, 0FFH, 0FDH, 0C0H, 000H, 001H
DB 080H, 000H, 006H, 003H, 0FCH, 010H, 000H, 000H, 000H, 000H, 00BH, 0FFH, 0FBH, 0C0H, 000H, 001H
DB 080H, 000H, 003H, 01FH, 000H, 018H, 000H, 000H, 000H, 000H, 017H, 0FFH, 0FBH, 080H, 000H, 001H
DB 080H, 000H, 003H, 000H, 000H, 008H, 000H, 000H, 000H, 000H, 017H, 0FFH, 0F7H, 080H, 000H, 001H
DB 080H, 000H, 001H, 080H, 000H, 00CH, 000H, 000H, 000H, 000H, 017H, 0FFH, 0F7H, 000H, 000H, 001H
DB 080H, 000H, 001H, 080H, 000H, 004H, 000H, 000H, 000H, 000H, 009H, 0FFH, 0EFH, 000H, 000H, 001H
DB 080H, 000H, 000H, 0C0H, 000H, 0F6H, 000H, 000H, 000H, 000H, 006H, 07FH, 0EEH, 000H, 000H, 001H
DB 080H, 000H, 000H, 0C0H, 003H, 0C2H, 000H, 000H, 000H, 000H, 001H, 09FH, 0DEH, 000H, 000H, 001H
DB 080H, 000H, 000H, 060H, 000H, 03BH, 000H, 000H, 000H, 000H, 000H, 067H, 0DFH, 000H, 000H, 001H
DB 080H, 000H, 000H, 060H, 000H, 0E3H, 000H, 000H, 000H, 000H, 000H, 019H, 0BFH, 000H, 000H, 001H
DB 080H, 000H, 000H, 030H, 003H, 08EH, 000H, 000H, 000H, 000H, 000H, 006H, 03FH, 000H, 000H, 001H
DB 080H, 000H, 000H, 030H, 006H, 03CH, 000H, 000H, 000H, 000H, 000H, 001H, 0FFH, 000H, 000H, 001H
DB 080H, 000H, 000H, 018H, 000H, 0F0H, 000H, 000H, 000H, 000H, 000H, 000H, 0FFH, 000H, 000H, 001H
DB 080H, 000H, 000H, 018H, 003H, 0C0H, 000H, 000H, 000H, 000H, 000H, 00FH, 07FH, 080H, 000H, 001H
DB 080H, 000H, 000H, 00CH, 00FH, 000H, 000H, 000H, 000H, 000H, 000H, 030H, 01FH, 0C0H, 000H, 001H
DB 080H, 000H, 000H, 00CH, 03CH, 000H, 000H, 000H, 000H, 000H, 000H, 040H, 007H, 0E0H, 000H, 001H
DB 080H, 000H, 000H, 006H, 0F0H, 000H, 000H, 000H, 000H, 000H, 000H, 0E0H, 01FH, 0E0H, 000H, 001H
DB 080H, 000H, 000H, 007H, 0C0H, 000H, 000H, 000H, 000H, 000H, 000H, 0F8H, 03FH, 0C0H, 000H, 001H
DB 080H, 000H, 000H, 003H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 07EH, 0FFH, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 01FH, 0FCH, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 007H, 0F0H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H, 0C0H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 005H, 040H, 088H, 004H, 001H, 008H, 000H, 000H, 00AH, 081H, 010H, 004H, 000H, 020H, 001H
DB 080H, 079H, 021H, 008H, 002H, 021H, 049H, 000H, 000H, 0F2H, 042H, 010H, 004H, 047H, 010H, 001H
DB 080H, 009H, 007H, 0DFH, 07FH, 0F7H, 0AAH, 000H, 000H, 012H, 00FH, 0BEH, 07FH, 0E5H, 0FEH, 001H
DB 080H, 07FH, 0F4H, 051H, 008H, 081H, 02CH, 000H, 000H, 0FFH, 0E8H, 0A2H, 044H, 047H, 000H, 001H
DB 080H, 009H, 004H, 061H, 008H, 081H, 07FH, 000H, 000H, 012H, 008H, 0C2H, 07FH, 0C5H, 04AH, 001H
DB 080H, 00BH, 024H, 051H, 008H, 083H, 081H, 000H, 000H, 016H, 048H, 0A2H, 044H, 045H, 06AH, 001H
DB 080H, 01DH, 047H, 0C9H, 005H, 005H, 001H, 000H, 000H, 03AH, 08FH, 092H, 044H, 047H, 052H, 001H
DB 080H, 068H, 084H, 049H, 005H, 005H, 07FH, 000H, 000H, 0D1H, 008H, 092H, 07FH, 0C5H, 06AH, 001H
DB 080H, 009H, 094H, 041H, 002H, 001H, 001H, 000H, 000H, 013H, 028H, 082H, 044H, 005H, 04AH, 001H
DB 080H, 00EH, 057H, 0C1H, 00DH, 081H, 001H, 000H, 000H, 01CH, 0AFH, 082H, 004H, 029H, 042H, 001H
```



```

DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 07EH, 0FDH, 0F8H, 007H, 0E7H, 007H, 03FH, 01FH, 039H, 0F0H, 0F8H, 003H, 0C7H, 03EH, 001H
DB 080H, 07EH, 0FDH, 0F8H, 003H, 0C3H, 00DH, 080H, 036H, 06CH, 0D9H, 0B0H, 006H, 0CDH, 0BFH, 001H
DB 080H, 03CH, 078H, 0F0H, 001H, 083H, 00FH, 080H, 01CH, 03CH, 0D8H, 0E0H, 006H, 00DH, 0BFH, 001H
DB 080H, 03CH, 078H, 0F0H, 003H, 0C3H, 00CH, 000H, 030H, 06CH, 0D9H, 080H, 006H, 00DH, 0BFH, 001H
DB 080H, 03CH, 078H, 0F1H, 087H, 0E7H, 087H, 080H, 03EH, 03FH, 0FDH, 0F3H, 003H, 0C7H, 03FH, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 033H, 000H, 001H, 098H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 01EH, 000H, 000H, 0F0H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH

```

END

十、附录部分

附录 1：ASCII 码表

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

16*8 半宽字符表

附录 2：汉字码址表

ST7920 GB 中文字型碼表

A1A0	“	”	()	<	>	《	》	「	」	『	』	【	】
A1B0	±	×	÷	:	∧	∨	Σ	Π	∩	∈	::	√	⊥	//
A1C0	∠	⊙	∫	≡	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
A1D0	∴	♂	♀	°	′	″	℃	\$	□	⊗	⊕	⊖	⊗	⊗
A1E0	○	●	◎	◇	◆	□	■	△	▲	※	→	←	↑	↓
A1F0														
A2A0														
A2B0	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.
A2C0	16.	17.	18.	19.	20.	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
A2D0	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)	(21)	(22)	(23)
A2E0	(24)	(25)	(26)	(27)	(28)	(29)	(30)	(31)	(32)	(33)	(34)	(35)	(36)	(37)
A2F0	!	!!	!!!	IV	V	VI	VII	VIII	IX	X	XI	XII		
A3A0	!	"	#	¥	%	&	'	()	*	+	-	.	/
A3B0	0	1	2	3	4	5	6	7	8	9	:	:	<	>
A3C0	@	A	B	C	D	E	F	G	H	I	J	K	L	M
A3D0	P	Q	R	S	T	U	V	W	X	Y	Z	[\]
A3E0	'	a	b	c	d	e	f	g	h	i	j	k	l	m
A3F0	p	q	r	s	t	u	v	w	x	y	z	{		}
A4A0	あ	い	う	え	お	か	が	き	く					
A4B0	け	こ	さ	し	ず	せ	そ	ぞ	た					
A4C0	ち	つ	て	と	な	に	ぬ	ね	の					
A4D0	は	び	ぶ	へ	べ	ほ	ば	ぼ	ま					
A4E0	め	も	や	ゆ	よ	ら	り	る	わ					
A4F0	ゐ	を	ん											
A5A0	ア	イ	ウ	エ	オ	カ	ガ	キ	ク					
A5B0	ケ	コ	サ	セ	ソ	タ	チ	ツ	テ					
A5C0	ダ	デ	ド	ナ	ニ	ノ	ハ	ヒ	フ					
A5D0	バ	ビ	ブ	ペ	ボ	マ	ミ	ム	メ					
A5E0	モ	ヤ	ユ	ヨ	ラ	リ	ル	ワ	ヰ					
A5F0	ヱ	ヰ	ヱ	ヰ	ヱ	ヰ	ヱ	ヰ	ヱ					
A6A0	A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ
A6B0	Π	P	Σ	T	Φ	X	Ψ	Ω						
A6C0	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ
A6D0	π	ρ	σ	τ	υ	φ	χ	ψ	ω					
A6E0														
A6F0														
A7A0	A	B	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н
A7B0	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы
A7C0	Ю	Я												
A7D0	a	b	в	г	д	е	ж	з	и	й	к	л	м	н
A7E0	o	p	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы
A7F0	ю	я												
A8A0	ā	ō	ū	ē	ē	ē	ē	ē	ē	ē	ē	ē	ē	ē
A8B0	ō	ū	ū	ū	ū	ū	ū	ū	ū	ū	ū	ū	ū	ū
A8C0	g													
A8D0	司	く	丁	虫	イ	尸	日	尸	尸	尸	尸	尸	尸	尸
A8E0	玄	又	馬	与	尤	厶	儿	丨						
A8F0														
A9A0														
A9B0	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐
A9C0	└	└	└	└	└	└	└	└	└	└	└	└	└	└
A9D0	+	+	+	+	+	+	+	+	+	+	+	+	+	+
A9E0	+	+	+	+	+	+	+	+	+	+	+	+	+	+
A9F0														
BOA0	啊	阿	埃	挨	哎	哀	皑	癌	藺	矮	艾	碍	爰	隘
BOB0	鞍	氨	安	俺	按	暗	岸	熬	案	肮	昂	盎	凹	敖
BOC0	袄	傲	奥	懊	澳	芭	捌	扒	叭	吧	笆	八	疤	巴

[illegible]

BAC0 豪盒亨吼
 BAE0 豪貉厚孤
 BAF0 话槐涣渥
 BBA0 话涣渥渥
 BBC0 话涣渥渥
 BBD0 话涣渥渥
 BBE0 话涣渥渥
 BBF0 话涣渥渥
 BCA0 及祭夹监
 BCB0 及祭夹监
 BCC0 及祭夹监
 BCD0 及祭夹监
 BCE0 及祭夹监
 BCF0 及祭夹监
 BDA0 及祭夹监
 BDB0 及祭夹监
 BDC0 及祭夹监
 BDD0 及祭夹监
 BDE0 及祭夹监
 BDF0 及祭夹监
 BEA0 及祭夹监
 BEB0 及祭夹监
 BEC0 及祭夹监
 BED0 及祭夹监
 BEE0 及祭夹监
 BEF0 及祭夹监
 BFA0 及祭夹监
 BFB0 及祭夹监
 BFC0 及祭夹监
 BFD0 及祭夹监
 BFE0 及祭夹监
 BFF0 及祭夹监
 COA0 及祭夹监
 COB0 及祭夹监
 COC0 及祭夹监
 COD0 及祭夹监
 COE0 及祭夹监
 COF0 及祭夹监
 C1A0 及祭夹监
 C1B0 及祭夹监
 C1C0 及祭夹监
 C1D0 及祭夹监
 C1E0 及祭夹监
 C1F0 及祭夹监
 C2A0 及祭夹监
 C2B0 及祭夹监
 C2C0 及祭夹监
 C2D0 及祭夹监
 C2E0 及祭夹监
 C2F0 及祭夹监
 C3A0 及祭夹监
 C3B0 及祭夹监
 C3C0 及祭夹监
 C3D0 及祭夹监
 C3E0 及祭夹监
 C3F0 及祭夹监
 C4A0 及祭夹监

C4B0 目南倪娘
 C4C0 木奈霓念
 C4D0 慕耐妮捻
 C4E0 慕耐妮捻
 C4F0 慕耐妮捻
 C5A0 慕耐妮捻
 C5B0 慕耐妮捻
 C5C0 慕耐妮捻
 C5D0 慕耐妮捻
 C5E0 慕耐妮捻
 C5F0 慕耐妮捻
 C6A0 慕耐妮捻
 C6B0 慕耐妮捻
 C6C0 慕耐妮捻
 C6D0 慕耐妮捻
 C6E0 慕耐妮捻
 C6F0 慕耐妮捻
 C7A0 慕耐妮捻
 C7B0 慕耐妮捻
 C7C0 慕耐妮捻
 C7D0 慕耐妮捻
 C7E0 慕耐妮捻
 C7F0 慕耐妮捻
 C8A0 慕耐妮捻
 C8B0 慕耐妮捻
 C8C0 慕耐妮捻
 C8D0 慕耐妮捻
 C8E0 慕耐妮捻
 C8F0 慕耐妮捻
 C9A0 慕耐妮捻
 C9B0 慕耐妮捻
 C9C0 慕耐妮捻
 C9D0 慕耐妮捻
 C9E0 慕耐妮捻
 C9F0 慕耐妮捻
 CAA0 慕耐妮捻
 CAB0 慕耐妮捻
 CAC0 慕耐妮捻
 CAD0 慕耐妮捻
 CAE0 慕耐妮捻
 CAF0 慕耐妮捻
 CBA0 慕耐妮捻
 CBB0 慕耐妮捻
 CBC0 慕耐妮捻
 CBD0 慕耐妮捻
 CBE0 慕耐妮捻
 CBF0 慕耐妮捻
 CCA0 慕耐妮捻
 CCB0 慕耐妮捻
 CCC0 慕耐妮捻
 CCD0 慕耐妮捻
 CCE0 慕耐妮捻
 CCF0 慕耐妮捻
 CDA0 慕耐妮捻
 CDB0 慕耐妮捻
 CDC0 慕耐妮捻
 CDD0 慕耐妮捻
 CDE0 慕耐妮捻
 CDF0 慕耐妮捻

F5C0	𨇖	𨇗	𨇘	𨇙	𨇚	𨇛	𨇜	𨇝	𨇞	𨇟	𨇠	𨇡	𨇢	𨇣	𨇤
F5D0	𨇥	𨇦	𨇧	𨇨	𨇩	𨇪	𨇫	𨇬	𨇭	𨇮	𨇯	𨇰	𨇱	𨇲	𨇳
F5E0	𨇴	𨇵	𨇶	𨇷	𨇸	𨇹	𨇺	𨇻	𨇼	𨇽	𨇾	𨇿	𨇿	𨇿	𨇿
F5F0	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿
F6A0		𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿
F6B0	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿
F6C0	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿
F6D0	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿
F6E0	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿
F6F0	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿
F7A0		𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿
F7B0	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿
F7C0	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿
F7D0	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿
F7E0	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿
F7F0	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿	𨇿
