

COG19264-33 使用说明书

目 录

序号	内 容 标 题	页码
1	概述	2
2	特点	2
3	外形及接口引脚功能	3~4
4	基本原理	4~5
5	技术参数	5
6	时序特性	6~7
7	指令功能及硬件接口与编程案例	8~末页

1. 概述

专注于液晶屏及液晶模块的研发、制造。所生产19264-33型液晶模块由于使用方便、显示清晰，广泛应用于各种人机交流面板。

19264-33可以显示192列*64行点阵单色图片，或显示 12 个/行*4 行 16*16 点阵的汉字，或显示 24 个/行*8 行 8*8 点阵的英文、数字、符号。

2. COG19264-33图像型点阵液晶模块的特性

2.1 结构轻、薄、带背光、焊接式 FPC。

2.2 IC 采用 UC1604c, 功能强大，稳定性好

2.3 功耗低:当电压为 3.3V 时，功耗低：不带背光 1mW (3.3V*0.3mA)，带背光不大于 100mW (3.3V*30mA)；

2.4 显示内容：

(1) 192*64 点阵单色图片，或其它小于 192*64 点阵的单色图片；

(2) 可选用 16*16 点阵或其他点阵的图片来自编汉字，按照 16*16 点阵汉字来计算可显示 12 字*4 行；

(3) 按照 12*12 点阵汉字来计算可显示 15 字*4 行；

(4) 按照 8*16 点阵汉字来计算可显示 24 字*4 行；

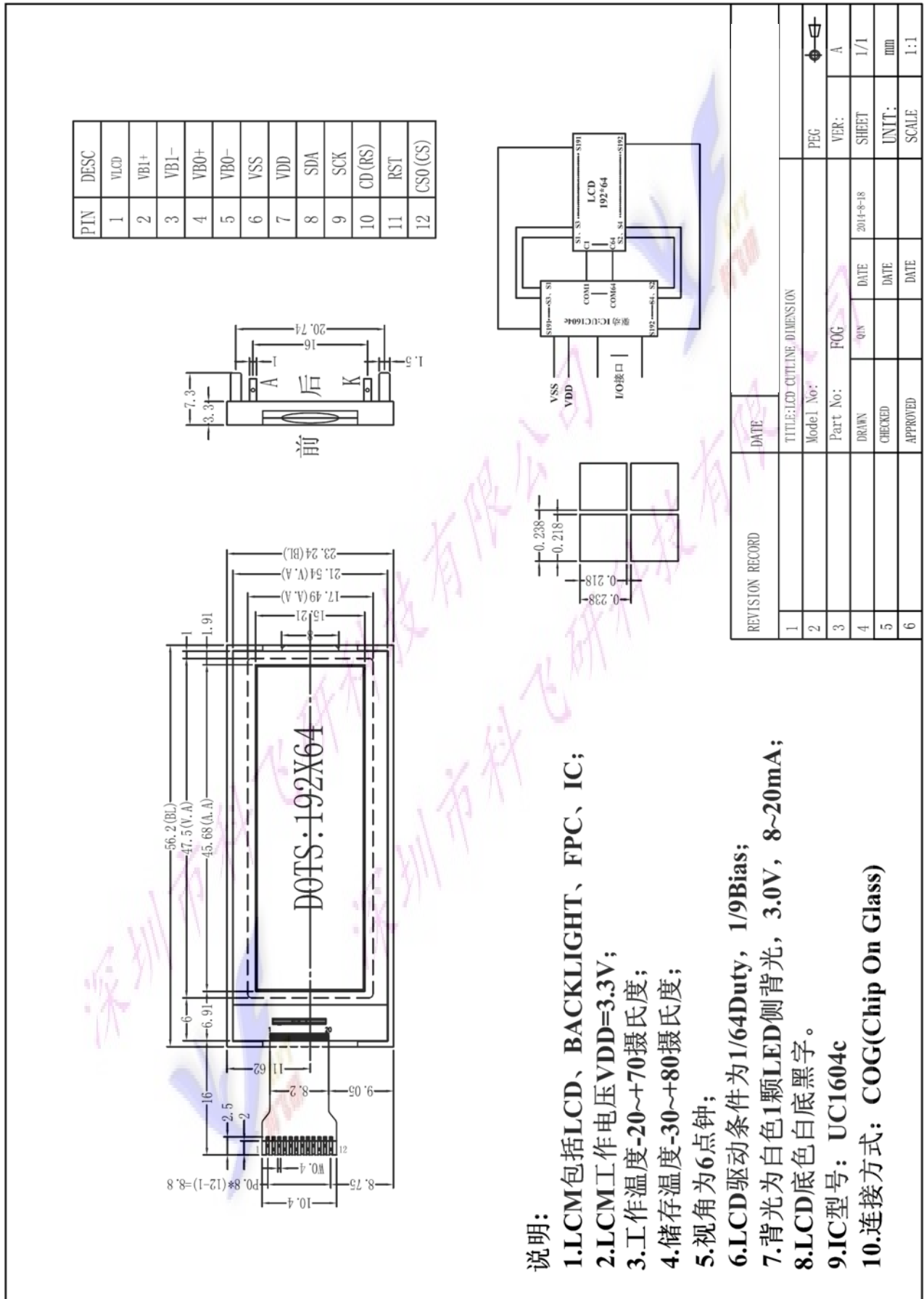
(5) 按照 5*8 点阵汉字来计算可显示 31 字*8 行；

2.5 指令功能强:可软件调对比度、正显/反显转换、行列扫描方向可改（可旋转 180 度使用）。

2.6 接口简单方便:串行接口。

2.7 工作温度宽:-20℃ - +70℃；

3. 外形尺寸及接口引脚功能



说明:

- 1.LCM包括LCD、BACKLIGHT、FPC、IC;
- 2.LCM工作电压VDD=3.3V;
- 3.工作温度-20~+70摄氏度;
- 4.储存温度-30~+80摄氏度;
- 5.视角为6点钟;
- 6.LCD驱动条件为1/64Duty, 1/9Bias;
- 7.背光为白色1颗LED侧背光, 3.0V, 8~20mA;
- 8.LCD底色白底黑字。
- 9.IC型号: UC1604c
- 10.连接方式: COG(Chip On Glass)

图 1. 外形尺寸

模块的接口引脚功能：

引 线 号	符 号	名 称	功 能
1	VLCD	VLCD	升压输出
2	VB1+	升压电容	
3	VB1-	升压电容	
4	VBO-	升压电容	
5	VBO+	升压电容	
6	VSS	接地	
7	VDD	供电电源正极	供电电源正极
8	SDA	I/O	串行接口时：串行数据(SDA)
9	SCK	I/O	串行接口时：串行时钟(SCLK)
10	CD(RS)	寄存器选择信号	H:数据寄存器 0:指令寄存器
11	RST	复位	低电平复位，复位完成后，回到高电平，液晶模块开始工作
12	CS0(CS)	片选	低电平片选

表 1：模块的接口引脚功能

4. 基本原理

4.1 液晶屏 (LCD)

在 LCD 上排列着 192×64 点阵, 192 个列信号与驱动 IC 相连, 64 个行信号也与驱动 IC 相连, IC 邦定在 LCD 玻璃上 (这种加工工艺叫 COG)。

4.2 工作电图：

图2是19264-33图像点阵型模块的电路框图, 它由驱动UC1604c及几个电阻电容组成。

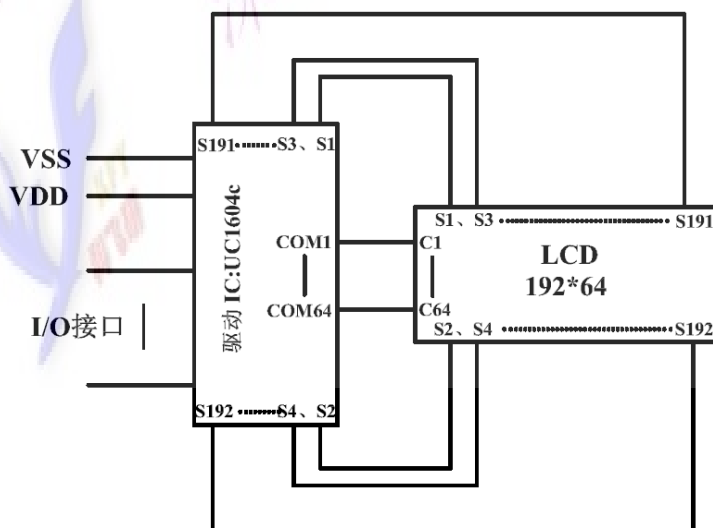


图2:19264-33图像点阵型液晶模块的电路框图

4.2 背光参数

该型号液晶模块带 LED 背光源。它的性能参数如下：

背光板可选择白色。

正常工作电流为：8~20mA（LED 灯数共 1 颗）；

工作电压：3.0V；

5. 技术参数

5.1 最大极限参数（超过极限参数则会损坏液晶模块）

名称	符号	标准值			单位
		最小	典型	最大	
电路电源	VDD - VSS	-0.3		3.6	V
工作温度		-20		+70	°C
储存温度		-30		+80	°C

表 2：最大极限参数

5.2 直流（DC）参数

名称	符号	测试条件	标准值			单位
			MIN	TYPE	MAX	
工作电压	VDD		2.4	3.3	3.6	V
背光工作电压	VLED		2.9	3.0	3.1	V
输入高电平	VIH	-	0.8xVDD		VDD	V
输入低电平	VI0	-	VSS		0.6	V
输出高电平	VOH	IOH = 0.2mA	0.8xVDD		VDD	V
输出低电平	VO0	I00 = 1.2mA	VSS		0.2xVDD	V
模块工作电流	IDD	VDD = 3.0V	-		0.3	mA
背光工作电流	ILED	VLED=3.0V	8	15	20	mA

表 3：直流（DC）参数

6. 读写时序特性

6.1 串行接口：

从 CPU 写到 UC1604c (Writing Data from CPU to UC1604c)

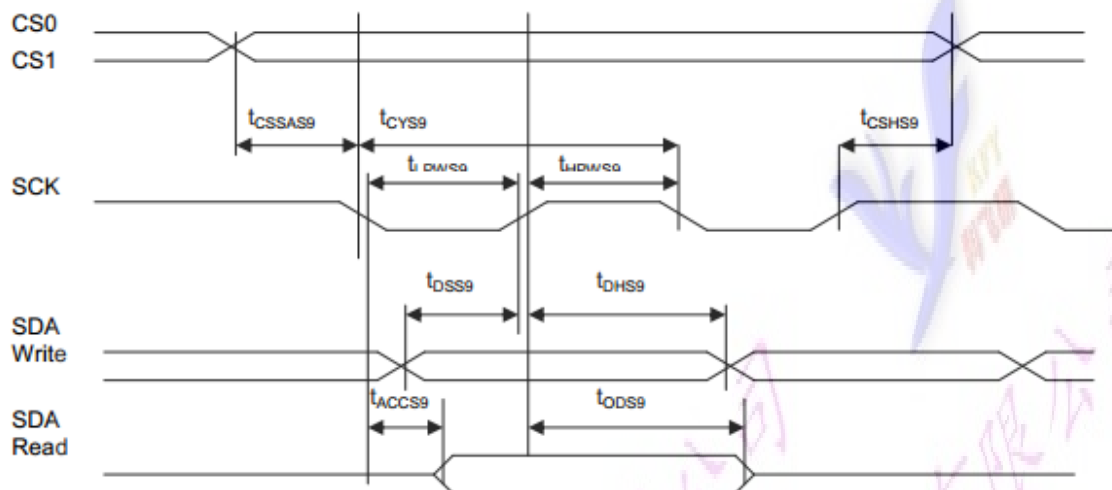


FIGURE 18: Serial Bus Timing Characteristics (for S9)

图 6. 从 CPU 写到 UC1604C (Writing Data from CPU to UC1604C)

6.2 串行接口：时序要求（AC 参数）：

写数据到 UC1604C 的时序要求：（6800 系列 MPU）

Symbol	Signal	Description	Condition	Min.	Max.	Unit
(2.5V ≤ V _{DD} ≤ 3.6V, T _a = -30 to +85 °C)				(Read / Write)		
t _{CSSAS9}	CS1, CS0	Chip select setup time		5	-	nS
t _{CSSHS9}		Chip select hold time		5	-	nS
t _{CYS9}		System cycle time		190 / 70	-	nS
t _{LPWS9}	SCK	Low pulse width		80 / 20	-	nS
t _{HPWS9}		High pulse width		80 / 20	-	nS
t _{DSS9}	SDA (Write)	Data setup time		20	-	nS
t _{DHS9}		Data hold time		10	-	nS
t _{ACC9}	SDA (Read)	Read access time	C _L = 100pF	-	80	nS
t _{OD9}		Output disable time		-	30	nS
(1.7V ≤ V _{DD} < 2.5V, T _a = -30 to +85 °C)				(Read / Write)		
t _{CSSAS9}	CS1, CS0	Chip select setup time		10	-	nS
t _{CSSHS9}		Chip select hold time		10	-	nS
t _{CYS9}		System cycle time		230 / 110	-	nS
t _{LPWS9}	SCK	Low pulse width		100 / 40	-	nS
t _{HPWS9}		High pulse width		100 / 40	-	nS
t _{DSS9}	SDA (Write)	Data setup time		24	-	nS
t _{DHS9}		Data hold time		15	-	nS
t _{ACC9}	SDA (Read)	Read access time	C _L = 100pF	-	100	nS
t _{OD9}		Output disable time		-	60	nS

Note: t_r (Rising time), t_f (falling time): ≤ 15nS

6.3 电源启动后复位的时序要求 (RESET CONDITION AFTER POWER UP):

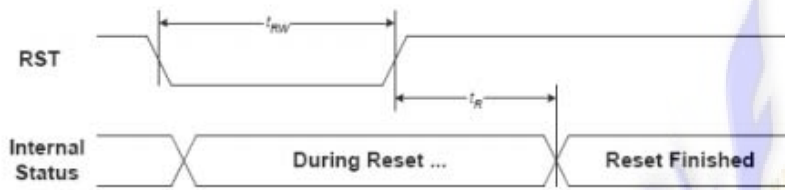


FIGURE 20: Reset Characteristics

($1.7V \leq V_{DD} \leq 3.6V$, $T_a = -30$ to $+85^\circ C$)

Symbol	Signal	Description	Condition	Min.	Max.	Unit
t_{RW}	RST	Reset low pulse width		3	-	μS
t_R	RST, Internal Status	Reset to Internal Status pulse delay		6	-	mS

图 7： 电源启动后复位的时序

7. 指令功能:

7.1 指令表

The following is a list of host commands supported by UC1604c

C/D: 0: Control, 1: Data **W/R**: 0: Write Cycle, 1: Read Cycle **D7-D0**: #: Useful Data bits -: Don't Care

No	Command	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0	Action	Default
1.	Write Data Byte	1	0	#	#	#	#	#	#	#	#	Write 1 byte	N/A
2.	Read Data Byte	1	1	#	#	#	#	#	#	#	#	Read 1 byte	N/A
3.	Get Status	0	1	ID	MX	MY	WA	DE	WS	MD	MS	Get Status PMO[5:0]	N/A
				VER	POR	#	#	#	#	#	#		
4.	Set Column Address LSB	0	0	0	0	0	0	#	#	#	#	Set CA [3:0]	0
	Set Column Address MSB	0	0	0	0	0	1	#	#	#	#	Set CA [7:4]	0
5.	Set Temp. Compensation	0	0	0	0	1	0	0	1	#	#	Set TC[1:0]	00b
6.	Set Power Control	0	0	0	0	1	0	1	#	#	#	Set PC[2:0]	110b
7.	Set Adv. Program Control (double-byte command)	0	0	0	0	1	1	0	0	R	R	Set APC[R][7:0], R = 0~3	N/A
				#	#	#	#	#	#	#	#		
8.	Set Scroll Line	0	0	0	1	#	#	#	#	#	#	Set SL[5:0]	0
9.	Set Page Address	0	0	1	0	1	1	#	#	#	#	Set PA[3:0]	0
10.	Set V _{BIAS} Potentiometer (double-byte command)	0	0	1	0	0	0	0	0	0	1	Set PM[7:0]	49H
				#	#	#	#	#	#	#	#		
11.	Set Partial Display Control	0	0	1	0	0	0	0	1	0	#	Set LC[5]	0b
12.	Set RAM Address Control	0	0	1	0	0	0	1	#	#	#	Set AC[2:0]	001b
13.	Set Frame Rate	0	0	1	0	1	0	0	0	#	#	Set LC[4:3]	01b
14.	Set All-Pixel-ON	0	0	1	0	1	0	0	1	0	#	Set DC[1]	0b
15.	Set Inverse Display	0	0	1	0	1	0	0	1	1	#	Set DC[0]	0b
16.	Set Display Enable	0	0	1	0	1	0	1	1	1	#	Set DC[2]	0b
17.	Set LCD Mapping Control	0	0	1	1	0	0	0	#	#	0	Set LC[2:1]	00b
18.	System Reset	0	0	1	1	1	0	0	0	1	0	System Reset	N/A
19.	NOP	0	0	1	1	1	0	0	0	1	1	No operation	N/A
20.	Set Test Control (double-byte command)	0	0	1	1	1	0	0	1	TT		For testing only. Do not use.	N/A
				#	#	#	#	#	#	#	#		
21.	Set LCD Bias Ratio	0	0	1	1	1	0	1	0	#	#	Set BR[1:0]	11b: 9
22.	Set COM End	0	0	--	--	#	#	#	#	#	#	Set CEN[5:0]	63D
23.	Set Partial Display Start	0	0	1	1	1	1	0	0	1	0	Set DST[5:0]	0
				--	--	#	#	#	#	#	#		
24.	Set Partial Display End	0	0	1	1	1	1	0	0	1	1	Set DEN[5:0]	63D
				--	--	#	#	#	#	#	#		
25.	Set MTP Operation Control	0	0	1	1	1	1	1	0	0	0	Set MTPC[4:0]	00H
				--	--	--	#	#	#	#	#		
26.	Set MTP Write Mask	0	0	1	1	1	1	1	0	0	1	Set MTPM[5:0]	0
				--	--	#	#	#	#	#	#		
27.	Set V _{MTP1} Potentiometer	0	0	1	1	1	1	0	1	0	0	Set VMTP1[7:0]	N/A
				#	#	#	#	#	#	#	#		
28.	Set V _{MTP2} Potentiometer	0	0	1	1	1	1	0	1	0	1	Set VMTP2[7:0]	N/A
				#	#	#	#	#	#	#	#		
29.	Set MTP Write Timer	0	0	1	1	1	1	0	1	1	0	Set MTPWT[7:0]	N/A
				#	#	#	#	#	#	#	#		
30.	Set MTP Read Timer	0	0	1	1	1	1	0	1	1	1	Set MTPRT[7:0]	N/A
				#	#	#	#	#	#	#	#		
Serial Read Command (Enabled only in S8/S9 mode)													
31.	Get Status	0	0	1	1	1	1	1	1	1	0	Get Status PMO[5:0]	N/A
		0	1	ID	MX	MY	WA	DE	WS	MD	MS		
32.	Read Data	0	0	1	1	1	1	1	1	1	1		FFH
		1	1	#	#	#	#	#	#	#	#		

Any bit pattern other than those listed above may result in NOP (No Operation).

7.3 点阵与 DD RAM(显示数据存储器)地址的对应关系

请留意页的定义：PAGE, 与平时所讲的“页”并不是一个意思，在此表示 8 个行就是一个“页”，一个 192*64 点阵的屏分为 8 个“页”，从第 0“页”到第 7“页”。

DB7—DB0 的排列方向：数据是从下向上排列的。最低位 D0 是在最上面，最高位 D7 是在最下面。每一位 (bit) 数据对应一个点阵，通常“1”代表点亮该点阵，“0”代表关掉该点阵。如下图所示：

D0	0	1	1	1		0
D1	1	0	0	0		0
D2	0	0	0	0		0
D3	0	1	1	1		0
D4	1	0	0	0		0
-						

Display data RAM
(显示数据存储器)

COM0		■	■	■		
COM1	■					
COM2						
COM3		■	■	■		
COM4	■					
-						

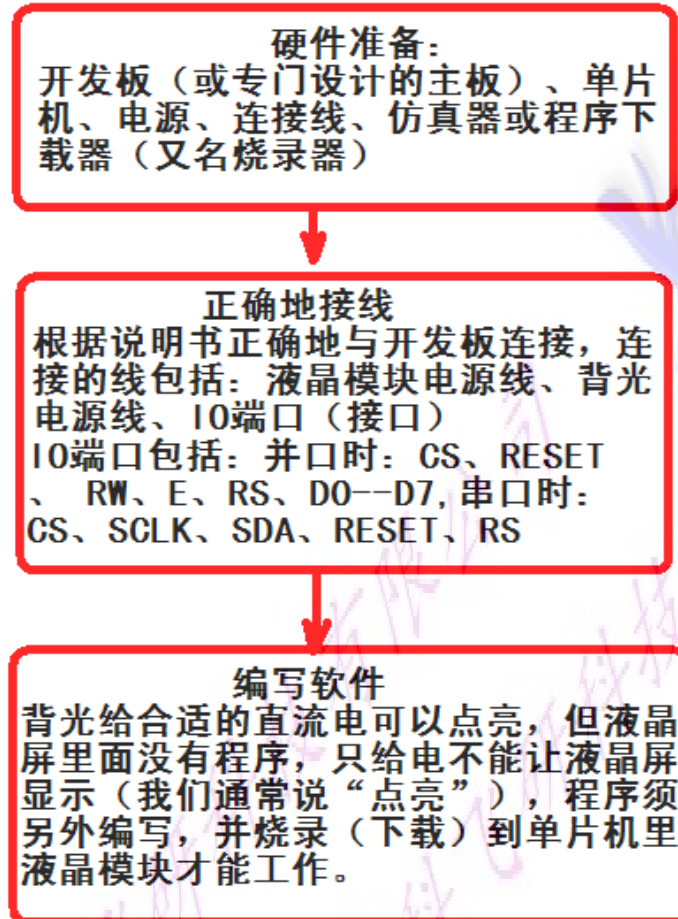
Liquid crystal display
(液晶屏)

下图摘自 UC1604C IC 资料，可通过“UC1604c_a1.3.pdf”之第 36 页获取最佳效果。

7.4 初始化方法

用户所编的显示程序, 开始必须进行初始化, 否则模块无法正常显示, 过程请参考程序

点亮液晶模块的步骤



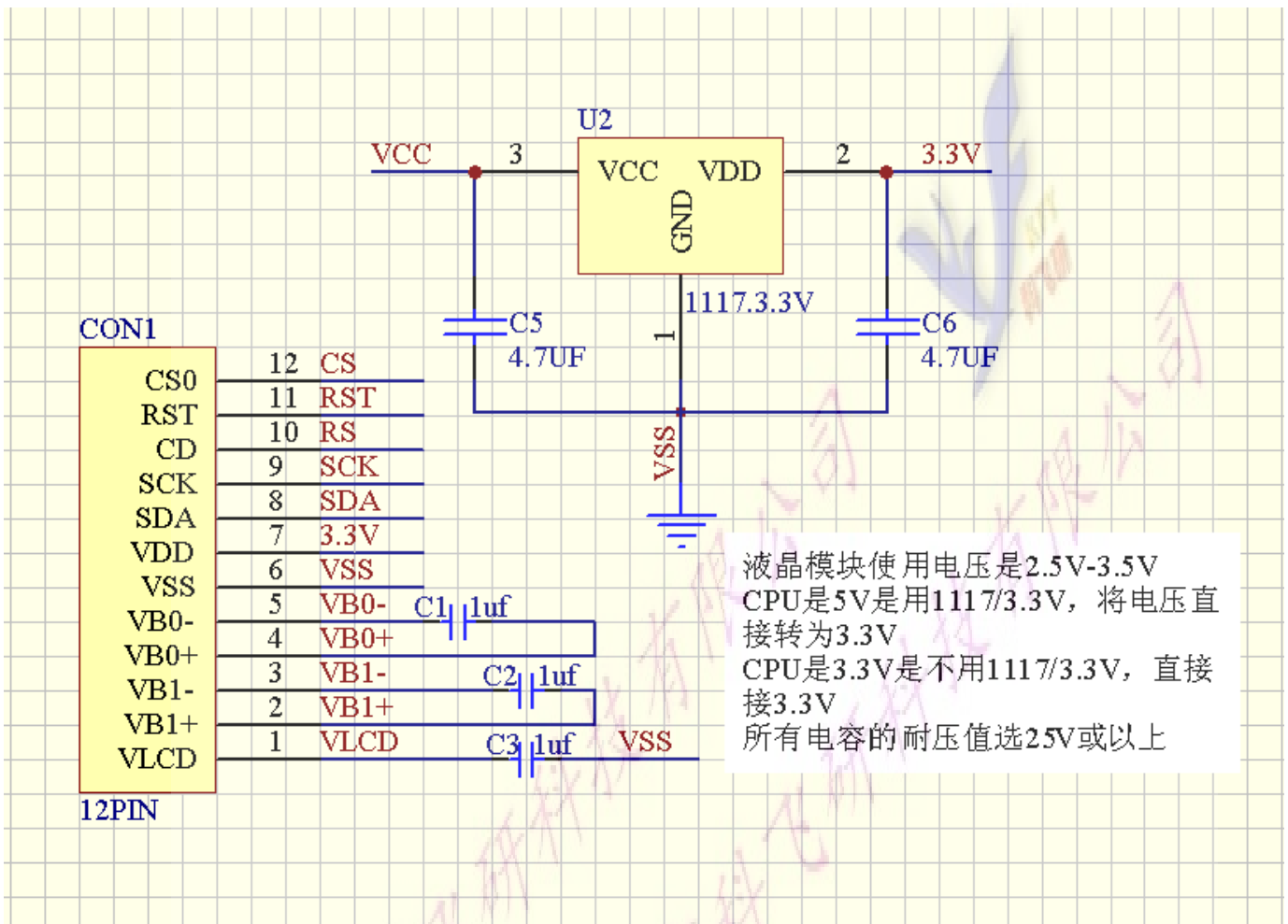
7.5 程序举例：

液晶模块与 MPU(以 8051 系列单片机为例)接口图如下：



图 8. 串行接口

串行电路图



7.5.1 程序

```
// 液晶演示程序COG19264-33，串行接口！
// 驱动 IC 是:UC1604c
```

```
#include <reg52.h>
#include <intrins.h>
#include <Ctype.h>
```

```
sbit key=P2^0;
```

```
sbit cs1=P1^1;
sbit reset=P1^0;
sbit rs=P3^0;
sbit sclk=P3^1;
sbit sid=P3^2;
```

```

#define uchar unsigned char
#define uint unsigned int
#define ulong unsigned long

uchar code ascii_table_8x16[95][16];
uchar code ascii_table_5x8[95][5];
uchar code bmp1[];

uchar code cheng1[]={
//-- 文字： 成 --
//-- 宋体 23； 此字体下对应的点阵为： 宽 x 高=31x31  --
//-- 高度不是 8 的倍数， 现调整为： 宽度 x 高度=32x32  --
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0C,
0xFC,0xFC,0x88,0x00,0x00,0x1C,0x78,0xF0,0xE0,0x00,0x80,0x80,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0xFF,0xFF,0x83,0x83,0x83,0x83,0x83,0x83,0x83,0xC3,0xC3,0x03,0x1F,
0xFF,0xFF,0x83,0x03,0x03,0x03,0xC3,0xF3,0xF3,0x63,0x03,0x03,0x00,0x00,0x00,0x00,
0x00,0x00,0xFC,0xFF,0x3F,0x00,0x80,0x00,0x00,0x80,0xFF,0xFF,0x03,0x00,0x00,0x03,
0x9F,0xFF,0xF8,0xF8,0xBE,0x1F,0x07,0x01,0x00,0x00,0xE0,0x20,0x00,0x00,0x20,0x38,
0x1F,0x07,0x01,0x00,0x00,0x01,0x01,0x07,0x07,0x23,0x31,0x18,0x0C,0x0E,0x07,0x03,
0x01,0x01,0x01,0x03,0x07,0x0F,0x0E,0x1C,0x1F,0x3F,0x30,0x00,0x00,0x00,0x00,0x00};

uchar code zhuang1[]={
//-- 文字： 状 --
//-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16  --
0x08,0x30,0x00,0xFF,0x20,0x20,0x20,0x20,0xFF,0x20,0xE1,0x26,0x2C,0x20,0x20,0x00,
0x04,0x02,0x01,0xFF,0x40,0x20,0x18,0x07,0x00,0x00,0x03,0x0C,0x30,0x60,0x20,0x00};

uchar code tai1[]={
//-- 文字： 态 --
//-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16  --
0x00,0x04,0x04,0x04,0x84,0x44,0x34,0x4F,0x94,0x24,0x44,0x84,0x84,0x04,0x00,0x00,
0x00,0x60,0x39,0x01,0x00,0x3C,0x40,0x42,0x4C,0x40,0x40,0x70,0x04,0x09,0x31,0x00};

uchar code shi1[]={
//-- 文字： 使 --
//-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16  --
0x40,0x20,0xF0,0x1C,0x07,0xF2,0x94,0x94,0x94,0xFF,0x94,0x94,0x94,0xF4,0x04,0x00,
0x00,0x00,0x7F,0x00,0x40,0x41,0x22,0x14,0x0C,0x13,0x10,0x30,0x20,0x61,0x20,0x00};

uchar code yong1[]={
//-- 文字： 用 --
//-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16  --
0x00,0x00,0x00,0xFE,0x22,0x22,0x22,0x22,0xFE,0x22,0x22,0x22,0x22,0xFE,0x00,0x00,
0x80,0x40,0x30,0x0F,0x02,0x02,0x02,0x02,0xFF,0x02,0x02,0x42,0x82,0x7F,0x00,0x00};

```

```
uchar code mao_hao[]={
//-- 文字：：(冒号) --
//-- 宋体 12；此字体下对应的点阵为：宽 x 高=8x16 --
0x00,0x00,0x00,0xC0,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x30,0x30,0x00,0x00,0x00};

char code num0[]={
//-- 文字：0 --
//-- 宋体 12；此字体下对应的点阵为：宽 x 高=8x16 --
0x00,0xE0,0x10,0x08,0x08,0x10,0xE0,0x00,0x00,0x0F,0x10,0x20,0x20,0x10,0x0F,0x00
};
char code num1[]={
//-- 文字：1 --
//-- 宋体 12；此字体下对应的点阵为：宽 x 高=8x16 --
0x00,0x10,0x10,0xF8,0x00,0x00,0x00,0x00,0x00,0x20,0x20,0x3F,0x20,0x20,0x00,0x00
};
char code num2[]={
//-- 文字：2 --
//-- 宋体 12；此字体下对应的点阵为：宽 x 高=8x16 --
0x00,0x70,0x08,0x08,0x08,0x88,0x70,0x00,0x00,0x30,0x28,0x24,0x22,0x21,0x30,0x00
};
char code num3[]={
//-- 文字：3 --
//-- 宋体 12；此字体下对应的点阵为：宽 x 高=8x16 --
0x00,0x30,0x08,0x88,0x88,0x48,0x30,0x00,0x00,0x18,0x20,0x20,0x20,0x11,0x0E,0x00
};
char code num4[]={
//-- 文字：4 --
//-- 宋体 12；此字体下对应的点阵为：宽 x 高=8x16 --
0x00,0x00,0xC0,0x20,0x10,0xF8,0x00,0x00,0x00,0x07,0x04,0x24,0x24,0x3F,0x24,0x00
};

char code num5[]={
//-- 文字：5 --
//-- 宋体 12；此字体下对应的点阵为：宽 x 高=8x16 --
0x00,0xF8,0x08,0x88,0x88,0x08,0x08,0x00,0x00,0x19,0x21,0x20,0x20,0x11,0x0E,0x00
};

char code num6[]={
//-- 文字：6 --
//-- 宋体 12；此字体下对应的点阵为：宽 x 高=8x16 --
0x00,0xE0,0x10,0x88,0x88,0x18,0x00,0x00,0x00,0x0F,0x11,0x20,0x20,0x11,0x0E,0x00
};
char code num7[]={
//-- 文字：7 --
```

```
//-- 宋体 12; 此字体下对应的点阵为：宽 x 高=8x16  --
0x00, 0x38, 0x08, 0x08, 0xC8, 0x38, 0x08, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x00, 0x00, 0x00, 0x00
};
char code num8[]={
//-- 文字： 8  --
//-- 宋体 12; 此字体下对应的点阵为：宽 x 高=8x16  --
0x00, 0x70, 0x88, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x1C, 0x22, 0x21, 0x21, 0x22, 0x1C, 0x00
};

char code num9[]={
//-- 文字： 9  --
//-- 宋体 12; 此字体下对应的点阵为：宽 x 高=8x16  --
0x00, 0xE0, 0x10, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x00, 0x00, 0x31, 0x22, 0x22, 0x11, 0x0F, 0x00
};

//写指令到 LCD 模块
void transfer_command(int data1)
{
    char i;
    cs1=0;
    rs=0;
    for(i=0;i<8;i++)
    {
        sclk=0;
        if(data1&0x80) sid=1;
        else sid=0;
        sclk=1;
        data1=data1<<=1;
    }
    cs1=1;
}

//写数据到 LCD 模块
void transfer_data(int data1)
{
    char i;
    cs1=0;
    rs=1;
    for(i=0;i<8;i++)
    {
        sclk=0;
        if(data1&0x80) sid=1;
        else sid=0;
        sclk=1;
        data1=data1<<=1;
    }
}
```

```

    cs1=1;
}

//延时
void delay(int i)
{
    int j,k;
    for(j=0;j<i;j++)
        for(k=0;k<110;k++);
}

void waitkey()
{
repeat:
    if(key==1)goto repeat;
    else delay(400);
}

//LCD 模块初始化
void initial_lcd()
{
    reset=0;        //低电平复位
    delay(100);
    reset=1;        //复位完毕
    delay(100);
    transfer_command(0xe2); //软复位
    delay(500);
    transfer_command(0x2f); //打开内部升压
    delay(500);

    transfer_command(0x81); //微调对比度
    transfer_command(0x45); //微调对比度的值，可设置范围 0x00~0xFF
    transfer_command(0xeb); //1/9 偏压比 (bias)
    transfer_command(0xc4); //行扫描顺序：从上到下 0xc2
    transfer_command(0xa0); //列扫描顺序：从左到右
    transfer_command(0xaf); //开显示
}

void lcd_address(uchar page,uchar column)
{
    column=column-1; //我们平常所说的第 1 列，在 LCD 驱动 IC 里是第 0 列。
    所以在这里减去 1.
    page=page-1;
    transfer_command(0xb0+page); //设置页地址。每页是 8 行。一个画面的 64 行被分成 8 个页。
    我们平常所说的第 1 页，在 LCD 驱动 IC 里是第 0 页，所以在这里减去 1

```



```
transfer_command(((column>>4)&0x0f)+0x10); //设置列地址的高4位
transfer_command(column&0x0f);          //设置列地址的低4位
}

//全屏清屏
void clear_screen()
{
    unsigned char i,j;
    for(i=0;i<9;i++)
    {
        lcd_address(1+i,1);
        for(j=0;j<192;j++)
        {
            transfer_data(0x00);
        }
    }
}

void display_graphic_192x64(uchar *dp)
{
    uchar i,j;
    for(i=0;i<9;i++)
    {
        lcd_address(1+i,1);
        for(j=0;j<192;j++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
}

//=====display a picture of 128*64 dots=====
void full_display(uchar data_left,uchar data_right)
{
    int i,j;
    for(i=0;i<8;i++)
    {
        lcd_address(i+1,1);
        for(j=0;j<96;j++)
        {
            transfer_data(data_left);
            transfer_data(data_right);
        }
    }
}
```

//显示 32x32 点阵图像、汉字、生僻字或 32x32 点阵的其他图标

```
void display_graphic_32x32(uchar page,uchar column,uchar *dp)
```

```
{
    uchar i, j;
    for(j=0;j<4;j++)
    {
        lcd_address(page+j, column);
        for (i=0;i<31;i++)
        {
            transfer_data(*dp);    //写数据到 LCD, 每写完一个 8 位的数据后列地址自动加 1
            dp++;
        }
    }
}
```

//显示 16x16 点阵图像、汉字、生僻字或 16x16 点阵的其他图标

```
void display_graphic_16x16(uchar page,uchar column,uchar *dp)
```

```
{
    uchar i, j;
    for(j=0;j<2;j++)
    {
        lcd_address(page+j, column);
        for (i=0;i<16;i++)
        {
            transfer_data(*dp);    //写数据到 LCD, 每写完一个 8 位的数据后列地址自动加 1
            dp++;
        }
    }
}
```

//显示 8x16 点阵图像、ASCII, 或 8x16 点阵的自造字符、其他图标

```
void display_graphic_8x16(uchar page,uchar column,uchar *dp)
```

```
{
    uchar i, j;
    for(j=0;j<2;j++)
    {
        lcd_address(page+j, column);
        for (i=0;i<8;i++)
        {
            transfer_data(*dp);    //写数据到 LCD, 每写完一个 8 位的数据后列地址自动加 1
            dp++;
        }
    }
}
```

```
void display_string_8x16(uint page,uint column,uchar *text)
```

```

{
uint i=0, j, k, n;
while(text[i]>0x00)
{
if((text[i]>=0x20)&&(text[i]<=0x7e))
{
j=text[i]-0x20;
for(n=0;n<2;n++)
{
lcd_address(page+n, column);
for(k=0;k<8;k++)
{
transfer_data(ascii_table_8x16[j][k+8*n]); //显示 5x7 的 ASCII 字到 LCD 上, y 为页
地址, x 为列地址, 最后为数据
}
}
i++;
column+=8;
}
else
i++;
}
}

```

//显示一串 5x8 点阵的字符串

//括号里的参数分别为（页，列，是否反显，数据指针）

```
void display_string_5x8(uint page, uint column, uchar reverse, uchar *text)
```

```

{
uchar i=0, j, k, data1;
while(text[i]>0x00)
{
if((text[i]>=0x20)&&(text[i]<=0x7e))
{
j=text[i]-0x20;
lcd_address(page, column);
for(k=0;k<5;k++)
{
if(reverse==1) data1=~ascii_table_5x8[j][k];
else data1=ascii_table_5x8[j][k];
transfer_data(data1);
}
if(reverse==1) transfer_data(0xff);
else transfer_data(0x00);
i++;
column+=6;
}
}
}

```

```

    }
    else
    i++;
}
}

void display_string_5x8_1(uint page,uint column,uchar *text)
{
    uint i=0,j,k;
    while(text[i]>0x00)
    {
        if((text[i]>=0x20)&&(text[i]<0x7e))
        {
            j=text[i]-0x20;
            lcd_address(page,column);
            for(k=0;k<5;k++)
            {
                transfer_data(ascii_table_5x8[j][k]); //显示 5x7 的 ASCII 字到 LCD 上, y 为页地址, x 为
                列地址, 最后为数据
            }
            i++;
            column+=6;
        }
        else
        i++;
    }
}

void main(void)
{
    while(1)
    {
        initial_lcd();
        clear_screen();
        display_string_5x8(1,1,1,"          MENU"); //显示 5x8 点阵的字符串, 括
        号里的参数分别为 (页, 列, 是否反显, 数据指针)
        display_string_5x8(3,1,0,"  Select>>>>");
        display_string_5x8(3,100,1,"1. Graphic  ");
        display_string_5x8(4,100,0,"2. Chinese  ");
        display_string_5x8(5,100,0,"3. Movie   ");
        display_string_5x8(6,100,0,"4. Contrast ");
        display_string_5x8(7,100,0,"5. Mirror  ");
        display_string_5x8(8,1,1,"  PRE  USER  DEL  NEW  ");
    }
}

```

```

display_string_5x8(8,59,0," ");
display_string_5x8(8,94,0," ");
display_string_5x8(8,97+48,0," ");
waitkey();

clear_screen(); //clear all dots
display_graphic_192x64(bmp1);
delay(2000);
waitkey();
clear_screen();
display_graphic_32x32(1,1,cheng1); //在第1页,第49列显示单个汉字"成"
delay(2000);
waitkey();
clear_screen(); //clear all dots
display_graphic_16x16(5,1,zhuang1); //在第5页,第1列显示单个汉字"状"
display_graphic_16x16(5,(1+16),tail); //在第5页,第17列显示单个汉字"态"
display_graphic_8x16(5,(1+16*2),mao_hao); //在第5页,第25列显示单个字符":"
display_graphic_16x16(5,(1+16*2+8),shi1); //在第5页,第41列显示单个汉字"使"
display_graphic_16x16(5,(1+16*3+8),yong1); //在第5页,第49列显示单个汉字"用"
display_graphic_8x16(5,(89),num0); //在第5页,第89列显示单个数字"0"
display_graphic_8x16(5,(89+8*1),num0); //在第5页,第97列显示单个数字"0"
display_graphic_8x16(5,(89+8*2),mao_hao); //在第5页,第105列显示单个字符":"
display_graphic_8x16(5,(89+8*3),num0); //在第5页,第113列显示单个数字"0"
display_graphic_8x16(5,(89+8*4),num0); //在第5页,第121列显示单个数字"0"
waitkey();
delay(2000);
clear_screen(); //clear all dots
display_string_8x16(1,1,"(<"0123456abc~`!@#%^\`>")); //在第1页,第1列显示字符串
display_string_8x16(3,1,"{[(<"'&*|\@#_+='\`>]}"); //在第*页,第*列显示字符串
display_string_5x8_1(5,1,"[!#$%&'()*+,-./0123456789:;<=>?]);
display_string_5x8_1(6,1,"[ABCDEFGHJKLMNOPQRSTUVWXYZabcd]);
display_string_5x8_1(7,1,"(abcdefghijklmnopqrstuvwxyabcd));
display_string_5x8_1(8,1,"{[(<"'&*|\@abcde012#_+='\`>]}");
waitkey();
delay(2000);
full_display(0xff,0xff);
waitkey();
delay(2000);
full_display(0x55,0xaa);
waitkey();
delay(2000);
full_display(0xaa,0x55);
waitkey();
delay(2000);
full_display(0xff,0x00);
waitkey();

```

```

    delay(2000);
    full_display(0x00, 0xff);
    waitkey();
    delay(2000);
}
}

uchar code ascii_table_8x16[95][16]={

//粗体8x16点阵的ASCII码的点阵数据，从“XXX-GB2312”型号的字库IC中读出来的国标的。
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // - ( 即
“空格”) ASCII 码: 0X20
0x00, 0x00, 0x38, 0xFC, 0xFC, 0x38, 0x00, 0x00, 0x00, 0x00, 0x0D, 0x0D, 0x00, 0x00, 0x00, // ! -
ASCII 码: 0X21
0x00, 0x0E, 0x1E, 0x00, 0x00, 0x1E, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // " -
0x20, 0xF8, 0xF8, 0x20, 0xF8, 0xF8, 0x20, 0x00, 0x02, 0x0F, 0x0F, 0x02, 0x0F, 0x0F, 0x02, 0x00, // # -
0x38, 0x7C, 0x44, 0x47, 0x47, 0xCC, 0x98, 0x00, 0x06, 0x0C, 0x08, 0x38, 0x38, 0x0F, 0x07, 0x00, // $ -
0x30, 0x30, 0x00, 0x80, 0xC0, 0x60, 0x30, 0x00, 0x0C, 0x06, 0x03, 0x01, 0x00, 0x0C, 0x0C, 0x00, // % -
0x80, 0xD8, 0x7C, 0xE4, 0xBC, 0xD8, 0x40, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x07, 0x0F, 0x08, 0x00, // & -
0x00, 0x10, 0x1E, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // ' -
0x00, 0x00, 0xF0, 0xF8, 0x0C, 0x04, 0x00, 0x00, 0x00, 0x00, 0x03, 0x07, 0x0C, 0x08, 0x00, 0x00, // ( -
0x00, 0x00, 0x04, 0x0C, 0xF8, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x08, 0x0C, 0x07, 0x03, 0x00, 0x00, // ) -

0x80, 0xA0, 0xE0, 0xC0, 0xC0, 0xE0, 0xA0, 0x80, 0x00, 0x02, 0x03, 0x01, 0x01, 0x03, 0x02, 0x00, // * -
ASCII 码: 0X2A
0x00, 0x80, 0x80, 0xE0, 0xE0, 0x80, 0x80, 0x00, 0x00, 0x00, 0x00, 0x03, 0x03, 0x00, 0x00, 0x00, // + -
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x1E, 0x0E, 0x00, 0x00, 0x00, // , -
0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // ---
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0C, 0x0C, 0x00, 0x00, 0x00, // . -
0x00, 0x00, 0x00, 0x80, 0xC0, 0x60, 0x30, 0x00, 0x0C, 0x06, 0x03, 0x01, 0x00, 0x00, 0x00, 0x00, // / -
0xF8, 0xF8, 0x0C, 0xC4, 0x0C, 0xF8, 0xF0, 0x00, 0x03, 0x07, 0x0C, 0x08, 0x0C, 0x07, 0x03, 0x00, // 0 -
ASCII 码: 0X30
0x00, 0x10, 0x18, 0xFC, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x08, 0x08, 0x0F, 0x0F, 0x08, 0x08, 0x00, // 1 -
0x08, 0x0C, 0x84, 0xC4, 0x64, 0x3C, 0x18, 0x00, 0x0E, 0x0F, 0x09, 0x08, 0x08, 0x0C, 0x0C, 0x00, // 2 -
0x08, 0x0C, 0x44, 0x44, 0x44, 0xFC, 0xB8, 0x00, 0x04, 0x0C, 0x08, 0x08, 0x08, 0x0F, 0x07, 0x00, // 3 -

0xC0, 0xE0, 0xB0, 0x98, 0xFC, 0xFC, 0x80, 0x00, 0x00, 0x00, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x00, // 4 -
ASCII 码: 0X34
0x7C, 0x7C, 0x44, 0x44, 0x44, 0xC4, 0x84, 0x00, 0x04, 0x0C, 0x08, 0x08, 0x08, 0x0F, 0x07, 0x00, // 5 -
0xF0, 0xF8, 0x4C, 0x44, 0x44, 0xC0, 0x80, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x08, 0x0F, 0x07, 0x00, // 6 -
0x0C, 0x0C, 0x04, 0x84, 0xC4, 0x7C, 0x3C, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x00, 0x00, 0x00, 0x00, // 7 -
0xB8, 0xFC, 0x44, 0x44, 0x44, 0xFC, 0xB8, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x08, 0x0F, 0x07, 0x00, // 8 -
0x38, 0x7C, 0x44, 0x44, 0x44, 0xFC, 0xF8, 0x00, 0x00, 0x08, 0x08, 0x08, 0x0C, 0x07, 0x03, 0x00, // 9 -
0x00, 0x00, 0x00, 0x30, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x06, 0x06, 0x00, 0x00, 0x00, // : -
0x00, 0x00, 0x00, 0x30, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x0E, 0x06, 0x00, 0x00, 0x00, // ; -
0x00, 0x80, 0xC0, 0x60, 0x30, 0x18, 0x08, 0x00, 0x00, 0x00, 0x01, 0x03, 0x06, 0x0C, 0x08, 0x00, // < -

```

0x00, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x00, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00,	//=-
0x00, 0x08, 0x18, 0x30, 0x60, 0xC0, 0x80, 0x00, 0x00, 0x08, 0x0C, 0x06, 0x03, 0x01, 0x00, 0x00,	//>-
ASCII 码: 0X3E	
0x18, 0x1C, 0x04, 0xC4, 0xE4, 0x3C, 0x18, 0x00, 0x00, 0x00, 0x00, 0x0D, 0x0D, 0x00, 0x00, 0x00,	//-?-
0xF0, 0xF0, 0x08, 0xC8, 0xC8, 0xF8, 0xF0, 0x00, 0x07, 0x0F, 0x08, 0x0B, 0x0B, 0x0B, 0x01, 0x00,	//-@-
0xE0, 0xF0, 0x98, 0x8C, 0x98, 0xF0, 0xE0, 0x00, 0x0F, 0x0F, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x00,	//-A-
ASCII 码: 0X41	
0x04, 0xFC, 0xFC, 0x44, 0x44, 0xFC, 0xB8, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x08, 0x0F, 0x07, 0x00,	//-B-
0xF0, 0xF8, 0x0C, 0x04, 0x04, 0x0C, 0x18, 0x00, 0x03, 0x07, 0x0C, 0x08, 0x08, 0x0C, 0x06, 0x00,	//-C-
0x04, 0xFC, 0xFC, 0x04, 0x0C, 0xF8, 0xF0, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x0C, 0x07, 0x03, 0x00,	//-D-
0x04, 0xFC, 0xFC, 0x44, 0xE4, 0x0C, 0x1C, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x08, 0x0C, 0x0E, 0x00,	//-E-
0x04, 0xFC, 0xFC, 0x44, 0xE4, 0x0C, 0x1C, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x00, 0x00, 0x00, 0x00,	//-F-
0xF0, 0xF8, 0x0C, 0x84, 0x84, 0x8C, 0x98, 0x00, 0x03, 0x07, 0x0C, 0x08, 0x08, 0x07, 0x0F, 0x00,	//-G-
0xFC, 0xFC, 0x40, 0x40, 0x40, 0xFC, 0xFC, 0x00, 0x0F, 0x0F, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x00,	//-H-
ASCII 码: 0X48	
0x00, 0x00, 0x04, 0xFC, 0xFC, 0x04, 0x00, 0x00, 0x00, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x00, 0x00,	//-I-
0x00, 0x00, 0x00, 0x04, 0xFC, 0xFC, 0x04, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x0F, 0x07, 0x00, 0x00,	//-J-
0x04, 0xFC, 0xFC, 0xC0, 0xE0, 0x3C, 0x1C, 0x00, 0x08, 0x0F, 0x0F, 0x00, 0x01, 0x0F, 0x0E, 0x00,	//-K-
0x04, 0xFC, 0xFC, 0x04, 0x00, 0x00, 0x00, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x08, 0x0C, 0x0E, 0x00,	//-L-
0xFC, 0xFC, 0x38, 0x70, 0x38, 0xFC, 0xFC, 0x00, 0x0F, 0x0F, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x00,	//-M-
0xFC, 0xFC, 0x38, 0x70, 0xE0, 0xFC, 0xFC, 0x00, 0x0F, 0x0F, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x00,	//-N-
0xF8, 0xFC, 0x04, 0x04, 0x04, 0xFC, 0xF8, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x08, 0x0F, 0x07, 0x00,	//-O-
0x04, 0xFC, 0xFC, 0x44, 0x44, 0x7C, 0x38, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x00, 0x00, 0x00, 0x00,	//-P-
0xF8, 0xFC, 0x04, 0x04, 0x04, 0xFC, 0xF8, 0x00, 0x07, 0x0F, 0x08, 0x0E, 0x3C, 0x3F, 0x27, 0x00,	//-Q-
0x04, 0xFC, 0xFC, 0x44, 0xC4, 0xFC, 0x38, 0x00, 0x08, 0x0F, 0x0F, 0x00, 0x00, 0x0F, 0x0F, 0x00,	//-R-
0x18, 0x3C, 0x64, 0x44, 0xC4, 0x9C, 0x18, 0x00, 0x06, 0x0E, 0x08, 0x08, 0x08, 0x0F, 0x07, 0x00,	//-S-
0x00, 0x1C, 0x0C, 0xFC, 0xFC, 0x0C, 0x1C, 0x00, 0x00, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x00, 0x00,	//-T-
0xFC, 0xFC, 0x00, 0x00, 0x00, 0xFC, 0xFC, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x08, 0x0F, 0x07, 0x00,	//-U-
0xFC, 0xFC, 0x00, 0x00, 0x00, 0xFC, 0xFC, 0x00, 0x01, 0x03, 0x06, 0x0C, 0x06, 0x03, 0x01, 0x00,	//-V-
0xFC, 0xFC, 0x00, 0x00, 0x00, 0xFC, 0xFC, 0x00, 0x07, 0x0F, 0x0E, 0x03, 0x0E, 0x0F, 0x07, 0x00,	//-W-
0x0C, 0x3C, 0xF0, 0xE0, 0xF0, 0x3C, 0x0C, 0x00, 0x0C, 0x0F, 0x03, 0x01, 0x03, 0x0F, 0x0C, 0x00,	//-X-
0x00, 0x0C, 0x7C, 0xC0, 0xC0, 0x7C, 0x3C, 0x00, 0x00, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x00, 0x00,	//-Y-
0x1C, 0x0C, 0x84, 0xC4, 0x64, 0x3C, 0x1C, 0x00, 0x0E, 0x0F, 0x09, 0x08, 0x08, 0x0C, 0x0E, 0x00,	//-Z-
0x00, 0x00, 0xFC, 0xFC, 0x04, 0x04, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x08, 0x08, 0x00, 0x00,	//-[-
0x38, 0x70, 0xE0, 0xC0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x03, 0x07, 0x0E, 0x00,	//-\-
0x00, 0x00, 0x04, 0x04, 0xFC, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x08, 0x08, 0x0F, 0x0F, 0x00, 0x00,	//-]-
0x08, 0x0C, 0x06, 0x03, 0x06, 0x0C, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,	//-^-
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20,	//-_-
0x00, 0x00, 0x03, 0x07, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,	//-`-
0x00, 0xA0, 0xA0, 0xA0, 0xE0, 0xC0, 0x00, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x07, 0x0F, 0x08, 0x00,	//-a-

ASCII 码：0X61

```

0x04, 0xFC, 0xFC, 0x20, 0x60, 0xC0, 0x80, 0x00, 0x00, 0x0F, 0x0F, 0x08, 0x08, 0x0F, 0x07, 0x00, //b-
0xC0, 0xE0, 0x20, 0x20, 0x20, 0x60, 0x40, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x08, 0x0C, 0x04, 0x00, //-c-
0x80, 0xC0, 0x60, 0x24, 0xFC, 0xFC, 0x00, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x07, 0x0F, 0x08, 0x00, //-d-
0xC0, 0xE0, 0xA0, 0xA0, 0xA0, 0xE0, 0xC0, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x08, 0x0C, 0x04, 0x00, //-e-

0x40, 0xF8, 0xFC, 0x44, 0x0C, 0x18, 0x00, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x00, 0x00, 0x00, 0x00, //-f-
0xC0, 0xE0, 0x20, 0x20, 0xC0, 0xE0, 0x20, 0x00, 0x27, 0x6F, 0x48, 0x48, 0x7F, 0x3F, 0x00, 0x00, //-g-

0x04, 0xFC, 0xFC, 0x40, 0x20, 0xE0, 0xC0, 0x00, 0x08, 0x0F, 0x0F, 0x00, 0x00, 0x0F, 0x0F, 0x00, //-h-
0x00, 0x00, 0x20, 0xEC, 0xEC, 0x00, 0x00, 0x00, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x00, 0x00, //-i-
0x00, 0x00, 0x00, 0x00, 0x20, 0xEC, 0xEC, 0x00, 0x00, 0x30, 0x70, 0x40, 0x40, 0x7F, 0x3F, 0x00, //-j-
0x04, 0xFC, 0xFC, 0x80, 0xC0, 0x60, 0x20, 0x00, 0x08, 0x0F, 0x0F, 0x01, 0x03, 0x0E, 0x0C, 0x00, //-k-
0x00, 0x00, 0x04, 0xFC, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x00, 0x00, //-l-
0xE0, 0xE0, 0x60, 0xC0, 0x60, 0xE0, 0xC0, 0x00, 0x0F, 0x0F, 0x00, 0x07, 0x00, 0x0F, 0x0F, 0x00, //-m-
0x20, 0xE0, 0xC0, 0x20, 0x20, 0xE0, 0xC0, 0x00, 0x00, 0x0F, 0x0F, 0x00, 0x00, 0x0F, 0x0F, 0x00, //-n-
0xC0, 0xE0, 0x20, 0x20, 0x20, 0xE0, 0xC0, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x08, 0x0F, 0x07, 0x00, //-o-

0x20, 0xE0, 0xC0, 0x20, 0x20, 0xE0, 0xC0, 0x00, 0x40, 0x7F, 0x7F, 0x48, 0x08, 0x0F, 0x07, 0x00, //-p-
0xC0, 0xE0, 0x20, 0x20, 0xC0, 0xE0, 0x20, 0x00, 0x07, 0x0F, 0x08, 0x48, 0x7F, 0x7F, 0x40, 0x00, //-q-

0x20, 0xE0, 0xC0, 0x60, 0x20, 0xE0, 0xC0, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x00, 0x00, 0x00, 0x00, //-r-
0x40, 0xE0, 0xA0, 0x20, 0x20, 0x60, 0x40, 0x00, 0x04, 0x0C, 0x09, 0x09, 0x0B, 0x0E, 0x04, 0x00, //-s-
0x20, 0x20, 0xF8, 0xFC, 0x20, 0x20, 0x00, 0x00, 0x00, 0x00, 0x07, 0x0F, 0x08, 0x0C, 0x04, 0x00, //-t-
0xE0, 0xE0, 0x00, 0x00, 0xE0, 0xE0, 0x00, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x07, 0x0F, 0x08, 0x00, //-u-
0x00, 0xE0, 0xE0, 0x00, 0x00, 0xE0, 0xE0, 0x00, 0x00, 0x03, 0x07, 0x0C, 0x0C, 0x07, 0x03, 0x00, //-v-
0xE0, 0xE0, 0x00, 0x80, 0x00, 0xE0, 0xE0, 0x00, 0x07, 0x0F, 0x0C, 0x07, 0x0C, 0x0F, 0x07, 0x00, //-w-
0x20, 0x60, 0xC0, 0x80, 0xC0, 0x60, 0x20, 0x00, 0x08, 0x0C, 0x07, 0x03, 0x07, 0x0C, 0x08, 0x00, //-x-
0xE0, 0xE0, 0x00, 0x00, 0x00, 0xE0, 0xE0, 0x00, 0x47, 0x4F, 0x48, 0x48, 0x68, 0x3F, 0x1F, 0x00, //-y-

0x60, 0x60, 0x20, 0xA0, 0xE0, 0x60, 0x20, 0x00, 0x0C, 0x0E, 0x0B, 0x09, 0x08, 0x0C, 0x0C, 0x00, //-z-
//
0x00, 0x40, 0x40, 0xF8, 0xBC, 0x04, 0x04, 0x00, 0x00, 0x00, 0x00, 0x07, 0x0F, 0x08, 0x08, 0x00, //-{-
}
0x00, 0x00, 0x00, 0xBC, 0xBC, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x00, 0x00, 0x00, //-|-
0x00, 0x04, 0x04, 0xBC, 0xF8, 0x40, 0x40, 0x00, 0x00, 0x08, 0x08, 0x0F, 0x07, 0x00, 0x00, 0x00, //-}-
0x08, 0x0C, 0x04, 0x0C, 0x08, 0x0C, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, //-~

```

ASCII 码：0X7E

};

```

uchar code ascii_table_5x8[95][5]={
/*全体 ASCII 列表:5x8 点阵*/
0x00, 0x00, 0x00, 0x00, 0x00, //- - //space

```


0x00, 0x00, 0x4f, 0x00, 0x00, //-!-
0x00, 0x07, 0x00, 0x07, 0x00, //-"-
0x14, 0x7f, 0x14, 0x7f, 0x14, //-#-
0x24, 0x2a, 0x7f, 0x2a, 0x12, //-\$\$-
0x23, 0x13, 0x08, 0x64, 0x62, //-%-
0x36, 0x49, 0x55, 0x22, 0x50, //-&-
0x00, 0x05, 0x07, 0x00, 0x00, //-'-
0x00, 0x1c, 0x22, 0x41, 0x00, //-(-
0x00, 0x41, 0x22, 0x1c, 0x00, //-)-
0x14, 0x08, 0x3e, 0x08, 0x14, //-*-
0x08, 0x08, 0x3e, 0x08, 0x08, //-+-
0x00, 0x50, 0x30, 0x00, 0x00, //-,-
0x08, 0x08, 0x08, 0x08, 0x08, //----
0x00, 0x60, 0x60, 0x00, 0x00, //-.-
0x20, 0x10, 0x08, 0x04, 0x02, //-/-
0x3e, 0x51, 0x49, 0x45, 0x3e, //-0-
0x00, 0x42, 0x7f, 0x40, 0x00, //-1-
0x42, 0x61, 0x51, 0x49, 0x46, //-2-
0x21, 0x41, 0x45, 0x4b, 0x31, //-3-
0x18, 0x14, 0x12, 0x7f, 0x10, //-4-
0x27, 0x45, 0x45, 0x45, 0x39, //-5-
0x3c, 0x4a, 0x49, 0x49, 0x30, //-6-
0x01, 0x71, 0x09, 0x05, 0x03, //-7-
0x36, 0x49, 0x49, 0x49, 0x36, //-8-
0x06, 0x49, 0x49, 0x29, 0x1e, //-9-
0x00, 0x36, 0x36, 0x00, 0x00, //-:-
0x00, 0x56, 0x36, 0x00, 0x00, //-;-
0x08, 0x14, 0x22, 0x41, 0x00, //-<-
0x14, 0x14, 0x14, 0x14, 0x14, //-==
0x00, 0x41, 0x22, 0x14, 0x08, //->-
0x02, 0x01, 0x51, 0x09, 0x06, //-?-
0x32, 0x49, 0x79, 0x41, 0x3e, //-@-
0x7e, 0x11, 0x11, 0x11, 0x7e, //-A-
0x7f, 0x49, 0x49, 0x49, 0x36, //-B-
0x3e, 0x41, 0x41, 0x41, 0x22, //-C-
0x7f, 0x41, 0x41, 0x22, 0x1c, //-D-
0x7f, 0x49, 0x49, 0x49, 0x41, //-E-
0x7f, 0x09, 0x09, 0x09, 0x01, //-F-
0x3e, 0x41, 0x49, 0x49, 0x7a, //-G-
0x7f, 0x08, 0x08, 0x08, 0x7f, //-H-
0x00, 0x41, 0x7f, 0x41, 0x00, //-I-
0x20, 0x40, 0x41, 0x3f, 0x01, //-J-
0x7f, 0x08, 0x14, 0x22, 0x41, //-K-
0x7f, 0x40, 0x40, 0x40, 0x40, //-L-
0x7f, 0x02, 0x0c, 0x02, 0x7f, //-M-
0x7f, 0x04, 0x08, 0x10, 0x7f, //-N-

0x3e, 0x41, 0x41, 0x41, 0x3e, //-0-
0x7f, 0x09, 0x09, 0x09, 0x06, //-P-
0x3e, 0x41, 0x51, 0x21, 0x5e, //-Q-
0x7f, 0x09, 0x19, 0x29, 0x46, //-R-
0x46, 0x49, 0x49, 0x49, 0x31, //-S-
0x01, 0x01, 0x7f, 0x01, 0x01, //-T-
0x3f, 0x40, 0x40, 0x40, 0x3f, //-U-
0x1f, 0x20, 0x40, 0x20, 0x1f, //-V-
0x3f, 0x40, 0x38, 0x40, 0x3f, //-W-
0x63, 0x14, 0x08, 0x14, 0x63, //-X-
0x07, 0x08, 0x70, 0x08, 0x07, //-Y-
0x61, 0x51, 0x49, 0x45, 0x43, //-Z-
0x00, 0x7f, 0x41, 0x41, 0x00, //-[-
0x02, 0x04, 0x08, 0x10, 0x20, //-\-
0x00, 0x41, 0x41, 0x7f, 0x00, //-]-
0x04, 0x02, 0x01, 0x02, 0x04, //-^-
0x40, 0x40, 0x40, 0x40, 0x40, //-_-
0x01, 0x02, 0x04, 0x00, 0x00, //-`-
0x20, 0x54, 0x54, 0x54, 0x78, //-a-
0x7f, 0x48, 0x48, 0x48, 0x30, //-b-
0x38, 0x44, 0x44, 0x44, 0x44, //-c-
0x30, 0x48, 0x48, 0x48, 0x7f, //-d-
0x38, 0x54, 0x54, 0x54, 0x58, //-e-
0x00, 0x08, 0x7e, 0x09, 0x02, //-f-
0x48, 0x54, 0x54, 0x54, 0x3c, //-g-
0x7f, 0x08, 0x08, 0x08, 0x70, //-h-
0x00, 0x00, 0x7a, 0x00, 0x00, //-i-
0x20, 0x40, 0x40, 0x3d, 0x00, //-j-
0x7f, 0x20, 0x28, 0x44, 0x00, //-k-
0x00, 0x41, 0x7f, 0x40, 0x00, //-l-
0x7c, 0x04, 0x38, 0x04, 0x7c, //-m-
0x7c, 0x08, 0x04, 0x04, 0x78, //-n-
0x38, 0x44, 0x44, 0x44, 0x38, //-o-
0x7c, 0x14, 0x14, 0x14, 0x08, //-p-
0x08, 0x14, 0x14, 0x14, 0x7c, //-q-
0x7c, 0x08, 0x04, 0x04, 0x08, //-r-
0x48, 0x54, 0x54, 0x54, 0x24, //-s-
0x04, 0x04, 0x3f, 0x44, 0x24, //-t-
0x3c, 0x40, 0x40, 0x40, 0x3c, //-u-
0x1c, 0x20, 0x40, 0x20, 0x1c, //-v-
0x3c, 0x40, 0x30, 0x40, 0x3c, //-w-
0x44, 0x28, 0x10, 0x28, 0x44, //-x-
0x04, 0x48, 0x30, 0x08, 0x04, //-y-
0x44, 0x64, 0x54, 0x4c, 0x44, //-z-
0x08, 0x36, 0x41, 0x41, 0x00, //-{-
0x00, 0x00, 0x77, 0x00, 0x00, //-|-

```
0x00, 0x41, 0x41, 0x36, 0x08, //-}-
0x04, 0x02, 0x02, 0x02, 0x01, //-~-
};
```

```
uchar code bmp1[]={
/*-- 调入了一幅图像：D:\e\新开发部\显示图案收藏\19264-32小熊及JERRY.bmp --*/
/*-- 宽度 x 高度=192x64 --*/
0xFF, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x81, 0xC1, 0xE1, 0xF1, 0xF9, 0xF9, 0xF9, 0xFD, 0xFD,
0xFD, 0xFD, 0xF9, 0xF9, 0xF1, 0xF1, 0xE1, 0xC1, 0xC1, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
0x41, 0x41, 0xC1, 0x81, 0x81, 0x81, 0xC1, 0xE1, 0xF1, 0xF9, 0xF9, 0xFD, 0xFD, 0xFD, 0xFD, 0xFD, 0xF9, 0xF9, 0xF1, 0xE1, 0xC1, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,
0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0xC1, 0xE1, 0xE1, 0xF1, 0xF1, 0xF9, 0xF9, 0xF9,
0xF9, 0xF9, 0xF9, 0xF1, 0xF1, 0xE1, 0xC1, 0x81, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,
0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,
0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,
0x01, 0x01, 0x81, 0x81, 0xF1, 0x79, 0x07, 0xC3, 0x81, 0x81, 0x81, 0x03, 0x07, 0x3D, 0xC1, 0x01,
0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x81, 0x81, 0x81,
0x81, 0xC1, 0x41, 0x41, 0xC1, 0xC1, 0x81, 0x81, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0xFF,
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFE, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x3F, 0x1F,
0x0F, 0x07, 0x07, 0x03, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0x03, 0x07, 0x07, 0x0F, 0x1F, 0x7F, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x80, 0xF0, 0xF0, 0x60, 0x20, 0x10, 0x10, 0x08, 0x08,
0x0C, 0x04, 0x04, 0x04, 0x04, 0x06, 0x06, 0x07, 0x07, 0x07, 0x07, 0x07, 0x07, 0x07, 0x0F, 0x0F, 0x0F,
0x1F, 0x3F, 0x3F, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0x7E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x01, 0x01, 0x83, 0xC3, 0x67, 0x27, 0x17, 0x1F, 0x0F, 0x0F, 0x0F, 0x07, 0x07, 0x07,
0x07, 0x07, 0x07, 0x07, 0x06, 0x04, 0x0C, 0x0C, 0x0C, 0x09, 0x09, 0x1F, 0x3E, 0x3F, 0x7C,
0xF8, 0xF8, 0xF8, 0xE0, 0xF0, 0x78, 0x38, 0x1C, 0xCC, 0xEE, 0x7F, 0x3F, 0x1F, 0x0F, 0x03, 0x03,
0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0x03, 0x03, 0x04, 0xF8, 0xC0, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x07, 0xFF, 0x3F, 0x07, 0xC1, 0xE0, 0xF0, 0xF8,
0xF8, 0xF8, 0xFC, 0xFC, 0xFC, 0xFC, 0xF8, 0xF8, 0xF8, 0xF0, 0xE0, 0xC0, 0x00, 0x00, 0x80, 0xC0,
0xE0, 0xF0, 0xF0, 0xF8, 0xF8, 0xF8, 0xF8, 0xF8, 0xF8, 0xF8, 0xF0, 0xF0, 0xE0, 0xC0, 0x83, 0x0F,
0x7F, 0xDF, 0xEF, 0xFF, 0xF7, 0x19, 0x04, 0x03, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xE0, 0xF0, 0xF8, 0xFC, 0xFC, 0xFC, 0xFE, 0xFE, 0xFE,
0xFE, 0xFC, 0xFC, 0xF8, 0xF8, 0xF1, 0xC3, 0x0F, 0x38, 0xE0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x80, 0xE0, 0x60, 0xE0, 0x60, 0xE0, 0xE0, 0xC0, 0x00, 0x00,
0x00, 0xE0, 0xFC, 0x07, 0x01, 0x00, 0x80, 0x08, 0x0C, 0x0E, 0x0C, 0x0C, 0x1C, 0x18, 0x70, 0x00,
0x00, 0x00, 0x10, 0x18, 0x08, 0x0C, 0x0C, 0x18, 0x38, 0x30, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x01, 0x03, 0xE1, 0x78, 0x3C, 0x0E, 0x07, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xFF, 0x00, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7F, 0xD8, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF,
```

0xFF, 0xFF, 0x8F, 0x07, 0x63, 0x23, 0x27, 0x07, 0xFF, 0xFF, 0xFF, 0xFF, 0x7F, 0x7E, 0xFF, 0xFF,
0xFF, 0xFF, 0x8F, 0x47, 0x23, 0x23, 0x47, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x7C,
0xC0, 0x7F, 0x11, 0xFF, 0x00, 0xC0, 0xF0, 0xF8, 0xFC, 0xFE, 0xFE, 0xFF, 0x7F, 0x3F, 0x3F, 0x3F,
0x3F, 0x3E, 0xFE, 0xFC, 0xF8, 0xF0, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF3, 0xC1, 0x80, 0x9C, 0x88,
0x80, 0xC1, 0xFF, 0xFF, 0xFF, 0xFF, 0x7F, 0x0F, 0x80, 0x80, 0x7F, 0x40, 0x40, 0x40, 0xC0, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0xC0, 0xC0, 0x60,
0xE0, 0x00, 0x00, 0xE0, 0x38, 0x17, 0x00, 0x03, 0x00, 0x01, 0xC0, 0xFD, 0x3E, 0x07, 0x00, 0x00,
0x3C, 0xE7, 0x80, 0x00, 0xF0, 0x1C, 0x03, 0xE3, 0xE3, 0xFE, 0xFC, 0x00, 0x00, 0x80, 0xF0, 0x78,
0x1C, 0x0C, 0x0C, 0x8E, 0x8E, 0x1C, 0xF8, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x0F, 0x39, 0x60, 0xC0, 0xC0, 0xC0, 0xC0, 0xE0, 0xE0, 0xF0, 0x78, 0x1C, 0x07, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0xE0, 0x7F, 0x07, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x6E, 0x99, 0x93, 0x97, 0x9F,
0x9F, 0xBF, 0x7F, 0x1F, 0x1E, 0x0E, 0x0E, 0x0F, 0x07, 0x07, 0x13, 0xB9, 0x78, 0x38, 0xB9, 0x03,
0x07, 0x0F, 0x0F, 0x1F, 0x1E, 0x1E, 0x1E, 0xFE, 0xFF, 0x3F, 0x3F, 0x2F, 0x27, 0x33, 0xF9, 0xCE,
0x03, 0x00, 0x00, 0x0F, 0x78, 0xCF, 0x3F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0xF0, 0xE1, 0xE3,
0xF0, 0xF0, 0xFC, 0xFF, 0xFF, 0x7F, 0x1F, 0x00, 0x70, 0xE1, 0xF3, 0x77, 0x07, 0x07, 0x07, 0x07,
0x07, 0x07, 0x07, 0x03, 0x01, 0x00, 0x06, 0xC5, 0xF4, 0x9C, 0xC4, 0xC2, 0xE2, 0xE1, 0xE0, 0xE0,
0xE0, 0xE0, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x7E, 0x03, 0x0F, 0x01, 0x0F, 0x07, 0xC0,
0x3F, 0x00, 0x03, 0x0F, 0x0C, 0x1C, 0x70, 0xC0, 0x80, 0x00, 0x00, 0x01, 0x07, 0xDE, 0x58, 0x71,
0xDC, 0xF6, 0xF3, 0xD0, 0xD1, 0xD3, 0x13, 0x23, 0xC3, 0x93, 0x39, 0x38, 0x18, 0x83, 0x8F, 0x18,
0xD0, 0xFE, 0xBF, 0xFF, 0xEF, 0xEF, 0xA7, 0xA3, 0x50, 0x50, 0x50, 0x50, 0x48, 0x68, 0x24, 0x20,
0x20, 0x00, 0xF8, 0xFE, 0x0F, 0x07, 0x07, 0x03, 0x03, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x80, 0xC0, 0x60, 0x18, 0x04, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0xF0, 0xF8, 0xF8, 0xFC, 0xFD,
0xFD, 0xFF, 0xFE, 0xFC, 0xFC, 0xE8, 0xC8, 0x18, 0x10, 0x30, 0xF0, 0xF0, 0xF1, 0xF1, 0xF0, 0x30,
0x10, 0x10, 0x18, 0x08, 0x08, 0x0C, 0x04, 0x0E, 0x1F, 0x3F, 0x7F, 0xFF, 0xFF, 0xFF, 0xFE, 0xFE,
0xFC, 0xFC, 0xFC, 0xF8, 0xF8, 0xE1, 0x1F, 0xFC, 0xF9, 0xF1, 0xE3, 0xA3, 0x93, 0xCB, 0xCF, 0xBF,
0x83, 0x03, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x80, 0x82, 0xC0, 0xC0, 0xC0,
0xE0, 0xF0, 0xF0, 0xC8, 0x0C, 0x1E, 0x3F, 0x3F, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0x7F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x20, 0x20, 0x60, 0x40, 0xC0, 0x80, 0x03,
0x01, 0x03, 0x06, 0x0C, 0x08, 0x18, 0x30, 0xC1, 0x87, 0x0F, 0x7E, 0xF0, 0xF0, 0xC0, 0x00, 0x01,
0x09, 0x01, 0x87, 0x87, 0x8F, 0xCE, 0xDE, 0xFC, 0xF8, 0xF8, 0xF9, 0xB9, 0xB9, 0xF1, 0xF0, 0xF8,
0x08, 0xF8, 0xD8, 0x09, 0x09, 0x19, 0x19, 0x11, 0xF1, 0xE1, 0xC3, 0x62, 0x22, 0x36, 0x1C, 0x0C,
0x06, 0x0F, 0x07, 0x01, 0x06, 0x0C, 0x08, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18, 0x08, 0x08, 0x0C,
0x0C, 0x06, 0x03, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x7F, 0x3F, 0x70, 0xFC, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFC, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x03, 0x0F, 0x7F,
0xFF, 0x3F, 0x3F, 0x1F, 0x1F, 0x67, 0xF8, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x3F, 0x1F, 0x07, 0x01,
0x01, 0x01, 0x01, 0x0F, 0x1F, 0x3F, 0x3F, 0x3F, 0x3F, 0x1F, 0x0F, 0x0F, 0x0F, 0x0F, 0x0F, 0x0F, 0x0F,
0x0F, 0x0F, 0x0F, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x07, 0x1F, 0xFF, 0x87, 0x03,
0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x03,
0x0C, 0x18, 0x70, 0xE0, 0x80, 0x00, 0x00, 0x00, 0x01, 0x03, 0x06, 0x98, 0xA1, 0xEF, 0xFF, 0x1C,
0x7E, 0x2F, 0x37, 0x03, 0x03, 0x01, 0x01, 0x01, 0x01, 0x10, 0x18, 0x0C, 0x0C, 0x07, 0x83, 0xC1,
0xF0, 0x1F, 0x07, 0x00, 0x00, 0x00, 0x80, 0xE3, 0x7F, 0x11, 0x01, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,  
0xFF, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0xBF, 0x81, 0x80, 0x81, 0x81, 0x83,  
0x83, 0x83, 0x83, 0x81, 0x81, 0x80, 0x80, 0x80, 0x80, 0x81, 0x83, 0x87, 0x87, 0x87, 0x87, 0x83,  
0x81, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80,  
0xBF, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x81, 0xB1, 0xBF, 0x81, 0x80, 0x80, 0x80, 0x80,  
0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80,  
0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80,  
0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80,  
0x80, 0x80, 0x80, 0x80, 0x83, 0x86, 0x8C, 0xB8, 0xA0, 0xB8, 0xBE, 0xA7, 0xB3, 0xA0, 0x80, 0x80,  
0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0xA0, 0xB0, 0x88, 0x87, 0xB0, 0x9E,  
0x83, 0x80, 0x80, 0x80, 0xB0, 0x98, 0x87, 0x83, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80,  
0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80,  
0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0xFF,  
};
```

