

# COG12864C1013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
6	时序特性	6~9
7	指令功能及硬件接口与编程案例	9~末页

深圳市科飞研科技有限公司

## 1. 概述

科飞研科技专注于液晶屏及液晶模块的研发、制造。所生产COG12864C1013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。

COG12864C1013可以显示128列\*64行点阵单色图片，或显示8个/行\*4行16\*16点阵的汉字，或显示16个/行\*8行8\*8点阵的英文、数字、符号。

## 2. COG12864C1013图像型点阵液晶模块的特性

2.1 结构牢：焊接式 FPC。

2.2 IC 采用 UC1701X, 功能强大，稳定性好

2.3 显示内容：

- 128\*64 点阵单色图片；

- 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 8 字/行\*4 行。按照 12\*12 点阵汉字来计算可显示 10 字/行\*4 行。

2.4 指令功能强：可组合成各种输入、显示、移位方式以满足不同的要求；

2.5 接口简单方便：并行接口。

2.6 工作温度宽：- 0℃ - 50℃；

2.7 可靠性高：寿命为 50,000 小时(25℃)。

### 3. 外形尺寸及接口引脚功能

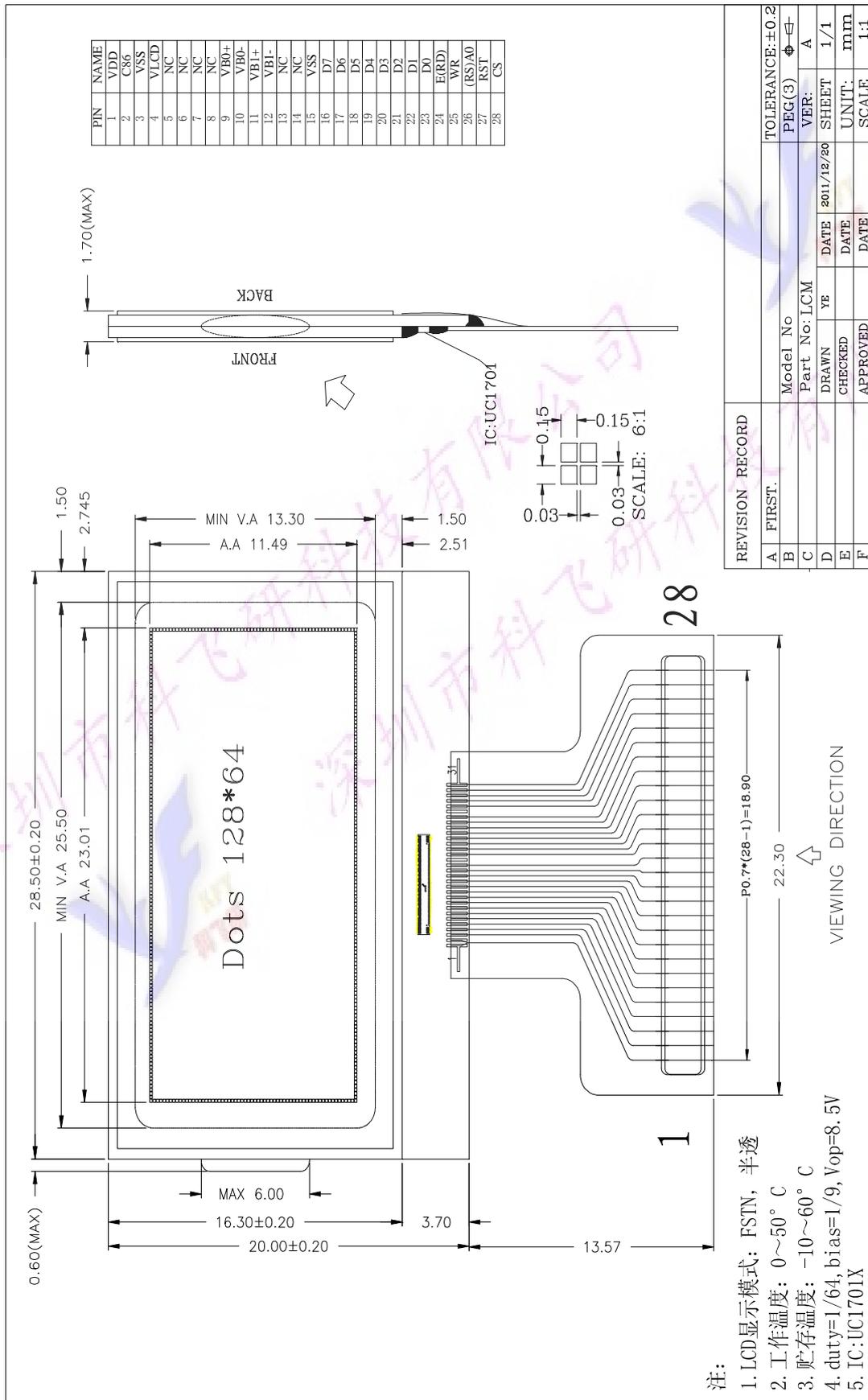


图 1. 外形尺寸

## 模块的接口引脚功能

引线号	符号	名称	功能
1	VDD	供电电源正极	供电电源正极
2	C86	选择 6800 或 8080	并行接口时：H:6800 系统,L:8080 系统。
3	VSS	接地	0V
4	VLCD	升压输出	LCD 倍压输出
5-8	NC	空脚	空脚
9	VB0+	倍压电路	倍压电路
10	VB0-	倍压电路	倍压电路
11	VB1+	倍压电路	倍压电路
12	VB1-	倍压电路	倍压电路
13-14	NC	空脚	空脚
15	VSS	接地	0V
16	D7	I/O	并行接口时：数据总线 DB7
17	D6	I/O	并行接口时：数据总线 DB6
18-23	D5-D0	I/O	并行接口时：数据总线 DB-D0
24	E (RD)	使能信号,或“读”	并行接口时并且选择 6800 时序时：使能信号,高电平有效. 并行接口时并且选择 8080 时序时：读数据,低电平有效.
25	WR	读/写,或“写”	并行接口时并且选择 6800 时序时：使能信号,高电平有效. 并行接口时并且选择 8080 时序时：读数据,低电平有效.
26	A0 (RS)	寄存器选择信号	H:数据寄存器 0:指令寄存器
27	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
28	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  $128 \times 64$  点阵,128 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 内部电路框图:

图2是COG12864C1013图像点阵型模块的电路框图,它由驱动UC1701X及几个电阻电容组成。

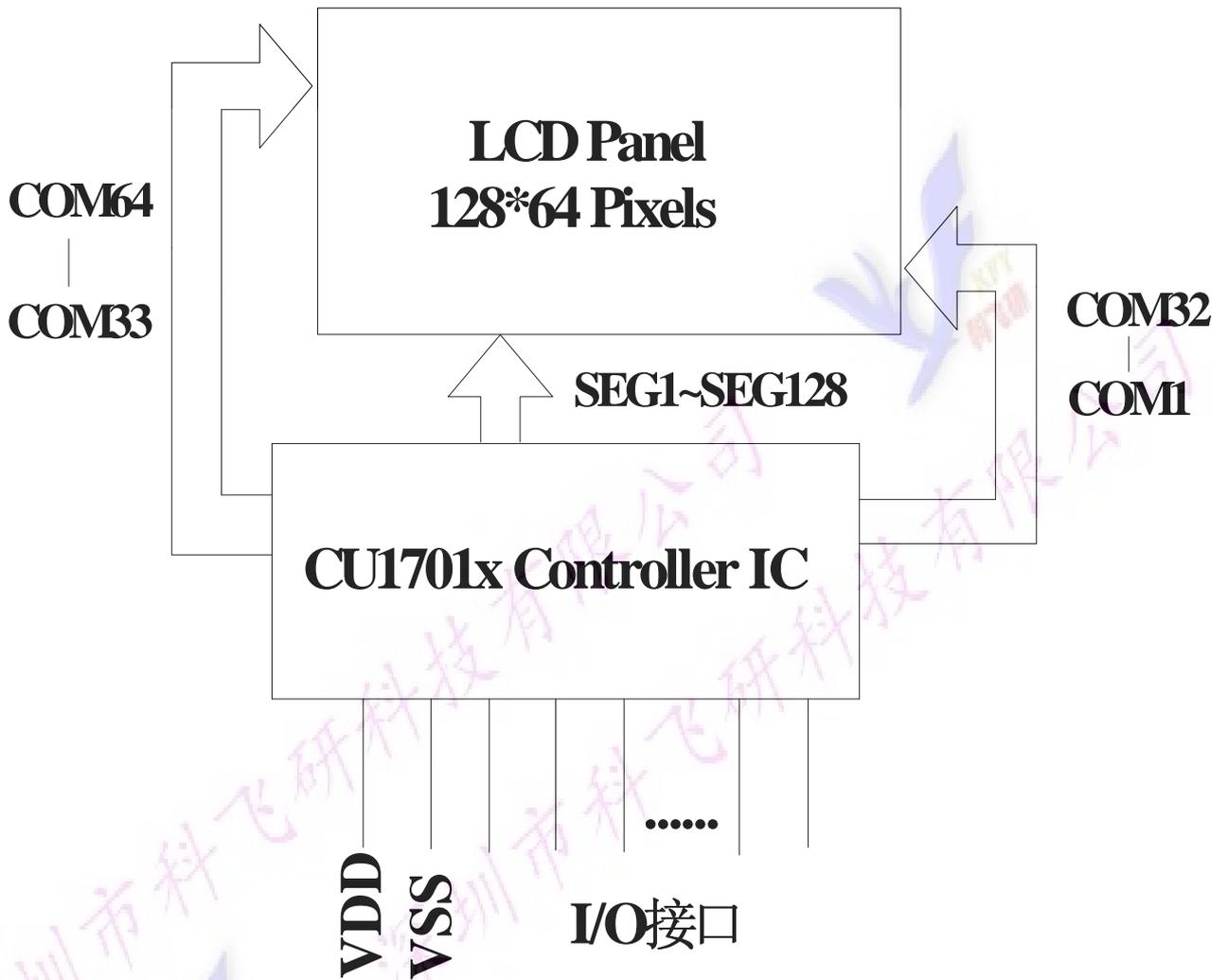


图 2: COG12864C1013图像点阵型液晶模块的电路框图

## 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		7.0	V
LCD 驱动电压	VDD - V0	VDD - 13.5		VDD + 0.3	V
静电电压		-	-	100	V
工作温度		0		+50	°C
储存温度		-20		+60	°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
输入高电平	VIH	-	2.2		VDD	V
输入低电平	VIO	-	-0.3		0.6	V
输出高电平	VOH	IOH = 0.2mA	2.4		-	V
输出低电平	VOO	IOO = 1.2mA	-		0.4	V
模块工作电流	IDD	VDD = 3.0V	-		1.0	mA

表 3: 直流 (DC) 参数

## 6. 读写时序特性

### 6.1 并行接口:

从 CPU 写到 UC1701X (Writing Data from CPU to UC1701X)

System Bus Read/Write Characteristics 1 (For the 8080 Series MPU)

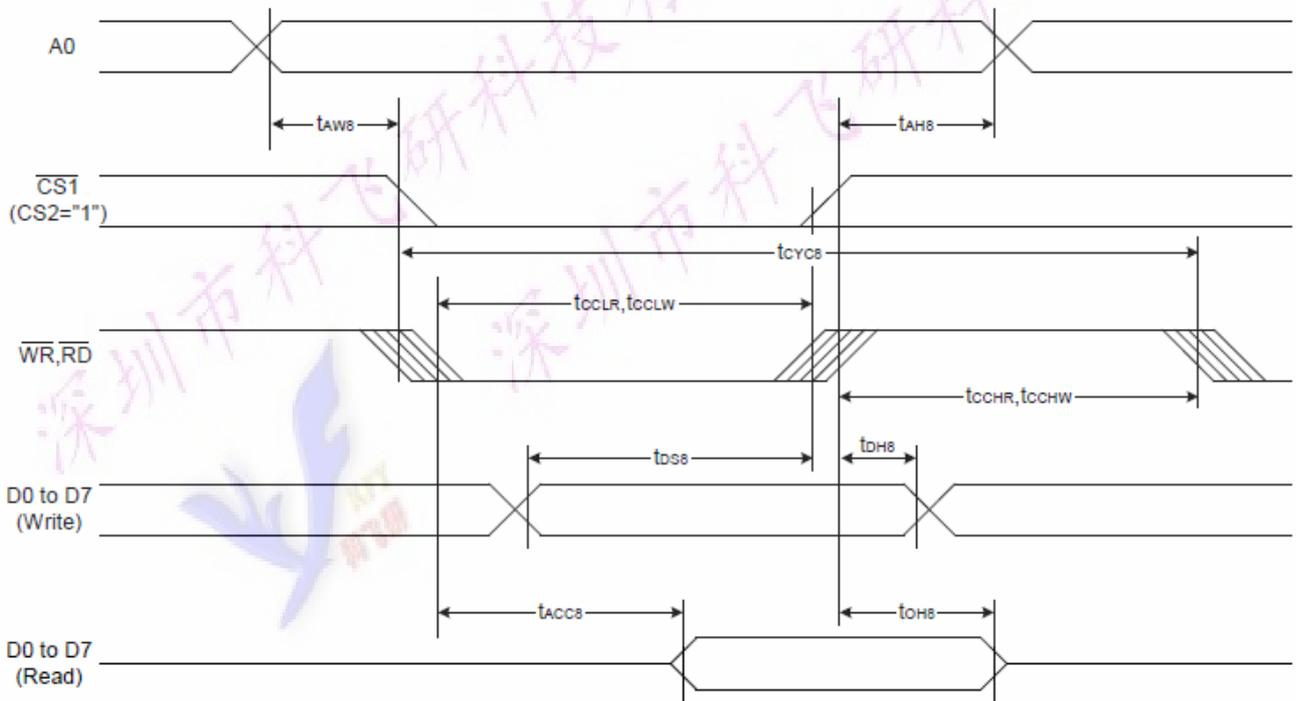


图 5. 从 CPU 写到 UC1701X (Writing Data from CPU to UC1701X)

System Bus Read/Write Characteristics 2 (For the 6800 Series MPU)

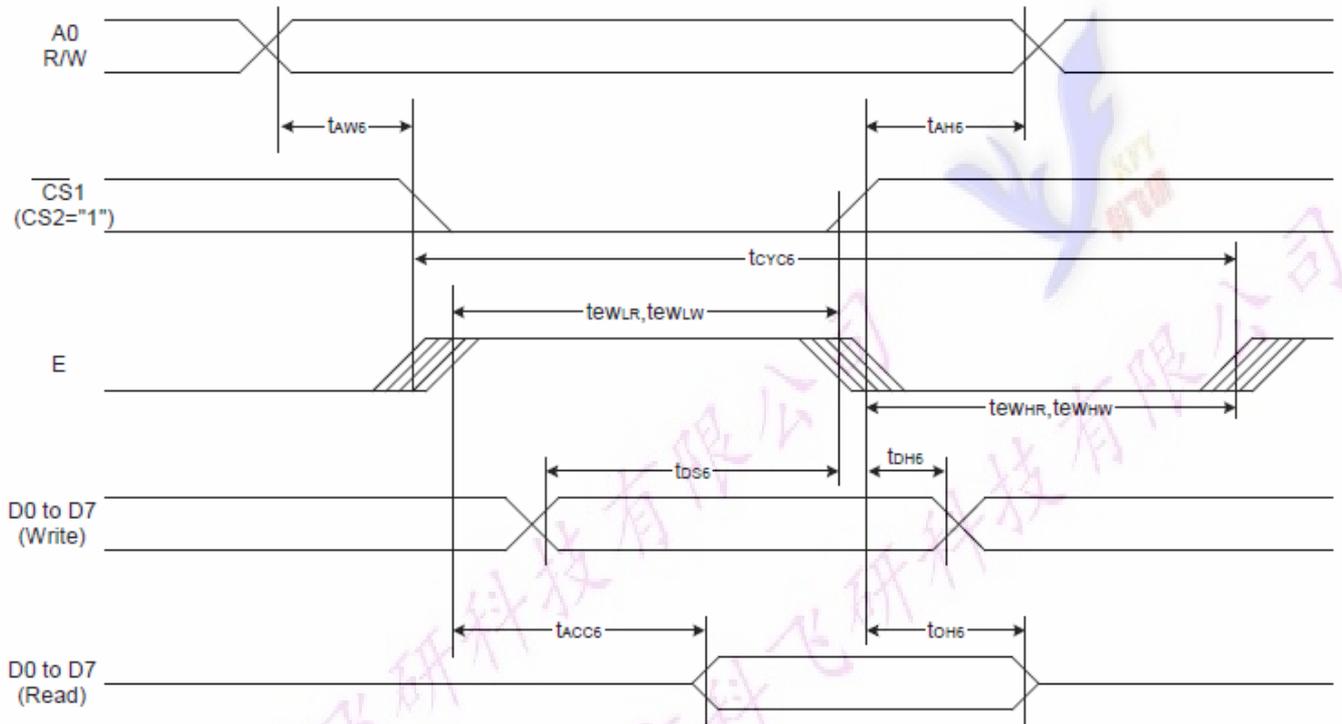


图 6. 从 CPU 写到 UC1701X (Writing Data from CPU to UC1701X)

6.4 并行接口：时序要求 (AC 参数)：

写数据到 UC1701X 的时序要求：(8080 系列 MPU)

项目	符号	测试条件	极限值			单位
			MIN	TYPE	MAX	
地址保持时间	A0	tAH8	0	--	--	ns
地址建立时间		tAW8	0		--	ns
系统循环时间		tCYC8	240		--	ns
使能“低”脉冲(写)	WR	tCCLW	80	--	--	ns
使能“高”脉冲(写)		tCCHW	80	--	--	ns
使能“低”脉冲(读)	RD	tCCLR	140	--	--	ns
使能“高”脉冲(读)		tCCHR	80	--		ns
写数据建立时间	D0-D7	tDS8	40		--	ns
写数据保持时间		tDH8	0		--	
读时间		tACC8	--		70	
读输出允许时间		tOH8	5		50	ns

**写数据到 UC1701X 的时序要求：（6800 系列 MPU）**

项目	符号	测试条件	极限值			单位
			MIN	TYPE	MAX	
地址保持时间	A0	tAH6	0	--	--	ns
地址建立时间		tAW6	0		--	ns
系统循环时间		tCYC6	240		--	ns
使能“低”脉冲（写）	WR	tEWLW	80	--	--	ns
使能“高”脉冲（写）		tEWHW	80	--	--	ns
使能“低”脉冲（读）	RD	tEWLR	80	--	--	ns
使能“高”脉冲（读）		tEWHR	140	--		ns
写数据建立时间	D0-D7	tDS6	40		--	ns
写数据保持时间		tDH6	0		--	
读时间		tACC6	--		70	
读输出允许时间		tOH6	5		50	ns

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

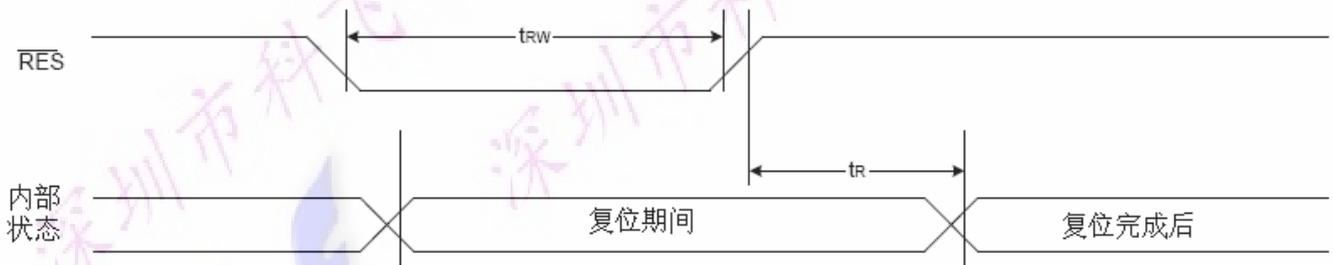


图 7：电源启动后复位的时序

表 6：电源启动后复位的时序要求

项目	符号	测试条件	极限值			单位
			MIN	TYPE	MAX	
复位时间	tr		--	--	1.0	us
复位保持低电平的时间	trw	引脚：RES	1.0	--	--	us

## 7. 指令功能:

## 7.1 指令表

指令表

表 8.

指令名称		指令码								说明	
		RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1		DB0
(1) 显示开/关 (display on/off)		0	1	0	1	0	1	1	1	0 1	显示开/关: 0:关, 1: 开
(2) 显示初始行设置 (Display start line set)		0	0	1	显示初始行地址, 共 5 位						设置显示存储器的显示初始行
(3) 页地址设置 (Page address set)		0	1	0	1	1	显示页地址, 共 4 位				设置显示页地址 (注: 每 8 行为一个页, 64 行分为 8 个页, 例 0000 为第一页, 0001 为第二页)
(4) 列地址高4位 设置	列地址高4位 设置	0	0	0	0	1	列地址的高 4 位				高 4 位与低 4 位共同组成列地址, 分别指定 128 列中任对应列。本液晶模块的第一列的地址为 00000000, 所以此指令表达为: 0x10, 0x00
	列地址低4位 设置										
(5) 读状态 (Status read)		0	状态				0	0	0	0	在本型号液晶模块不用此指令
(6) 写数据 (Display data write)		1	8 位显示数据								从 CPU 写数据到液晶模块
(7) 读数据 (Display data read)		1	8 位显示数据								在本型号液晶模块不用此指令
(8) 显示列地址增 减 (ADC select)			1	0	1	0	0	0	0	0 1	显示列地址增减: 0: 常规: 从左到右, 1: 反转: 从右到左
(9) 显示正显/反显 (Display normal/reverse)		0	1	0	1	0	0	1	1	0 1	显示正显/反显: 0: 常规: 正显 1: 反显
(10) 显示全部点阵 (Display all points)		0	1	0	1	0	0	1	0	0 1	显示全部点阵: 0: 常规 1: 显示全部点阵
(11) LCD 偏压比设置 (LCD bias set)		0	1	0	1	0	0	0	1	0 1	设置偏压比: 0: 1/9 BIAS 1: 1/7BIAS
(12) Read-modify-write		0	1	1	1	0	0	0	0	0	Column address increment At write: +1 At read: 0
(13) 退出上述指令 (End)		0	1	1	1	0	1	1	1	0	退出上述 “read/modify/write” 指令
(14) 软件复位 (Reset)		0	1	1	1	0	0	0	1	0	软件复位。

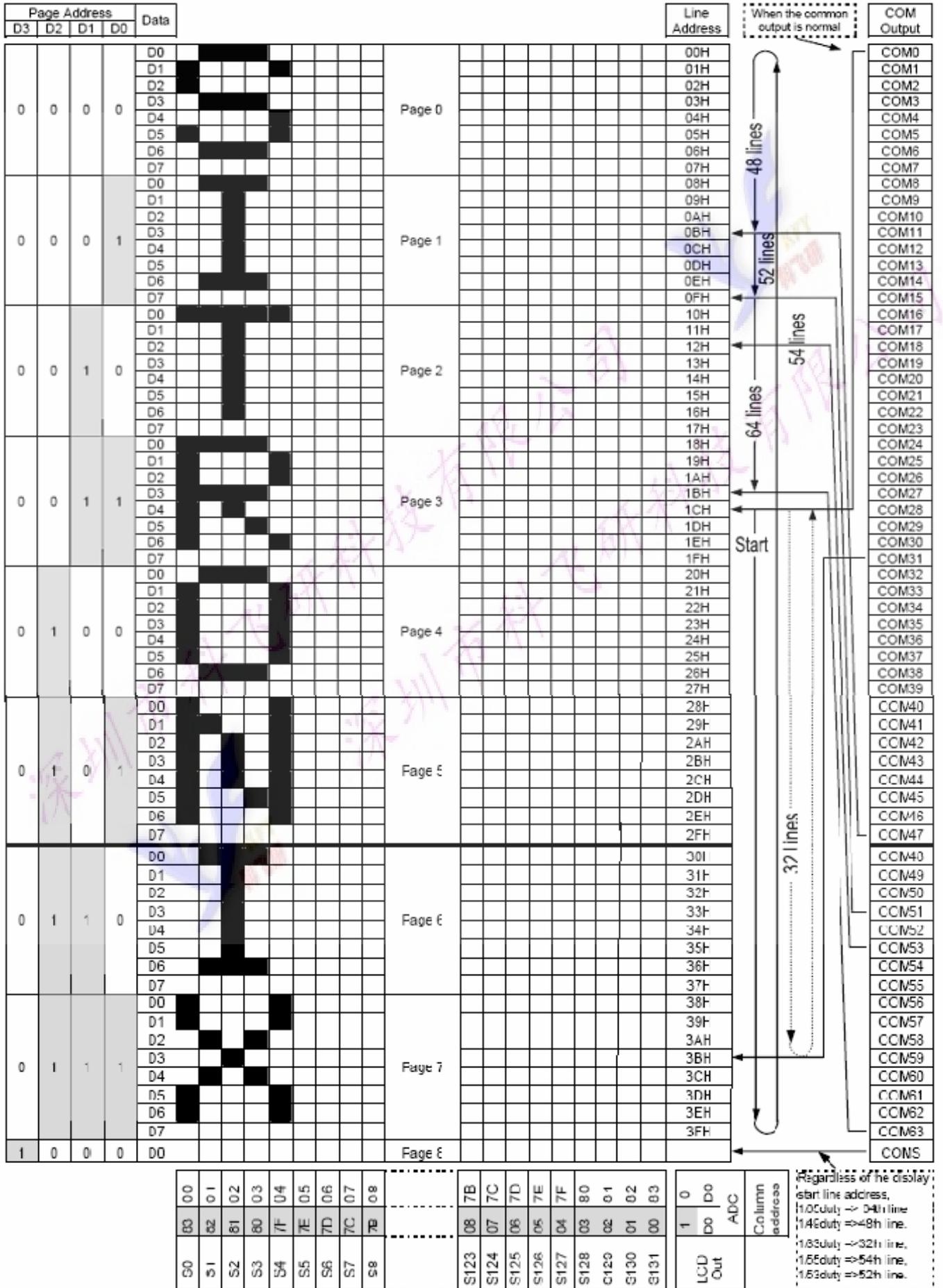
(15) 行扫描顺序选择(Common output mode select)		1	1	0	0	0	0	0	0	0	行扫描顺序选择： 0：普通顺序 1：反向扫描
(16) 电源控制 (Power control set)		0	0	1	0	1	<b>电压操作模式选择，共3位</b>			选择内部电压供应操作模式	
(17) 选择内部电阻比例	0	0	0	1	0	0	<b>内部电压值电阻设置</b>			选择内部电阻比例 (Rb/Ra)，本液晶模块通过外置电阻设置，此指令失效	
(18) 内部设置液晶电压模式 设置的电压值	0	1	0	0	0	0	0	0	1	设置内部电阻微调，以设置液晶电压，此两个指令需紧接着使用	
		0	0	<b>6位电压值数据，0~63共64级</b>							
(19) 静态图标显示：开/关	0	1	0	1	0	1	1	0	<b>0</b> <b>1</b>	0: 关, 1: 开。本液晶屏无此图标。此指令在进入及退出睡眠模式时起作用	
(20) 升压倍数选择 (Booster ratio set)	0	1	1	1	1	1	0	0	0	选择升压倍数： 00: 2倍, 3倍, 4倍 01: 5倍 11: 6倍。本模块外部已设置升压倍数为4倍，不必使用此指令	
		0	0	0	0	0	0	0	2位数设置升压倍数		
(21) 省电模式 (Power save)										省电模式，此非一条指令，是由“(10)显示全部点阵”、(19)静态图标显示：开/关等指令合成一个“省电功能”。详细看 IC 规格书第47页“POWER SAVE”	
(22) 空指令 (NOP)	0	1	1	1	0	0	0	1	1	空操作	
(23) 测试 (Test)	0	1	1	1	1	*	*	*	*	内部测试用，千万别用！	

请详细参考 IC 资料“UC1701X\_V15.PDF”的第42~49页。

### 7.3 点阵与 DD RAM 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128\*32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。下图摘自 UC1701X IC 资料，可通过“UC1701X\_V15.PDF”之第27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

## 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:

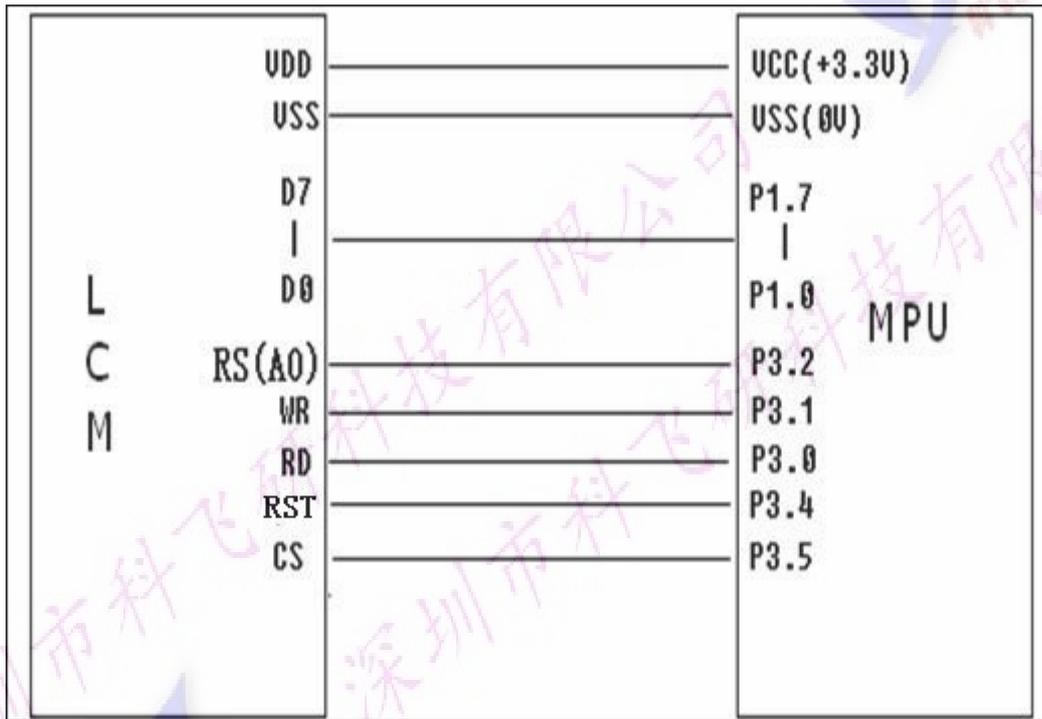
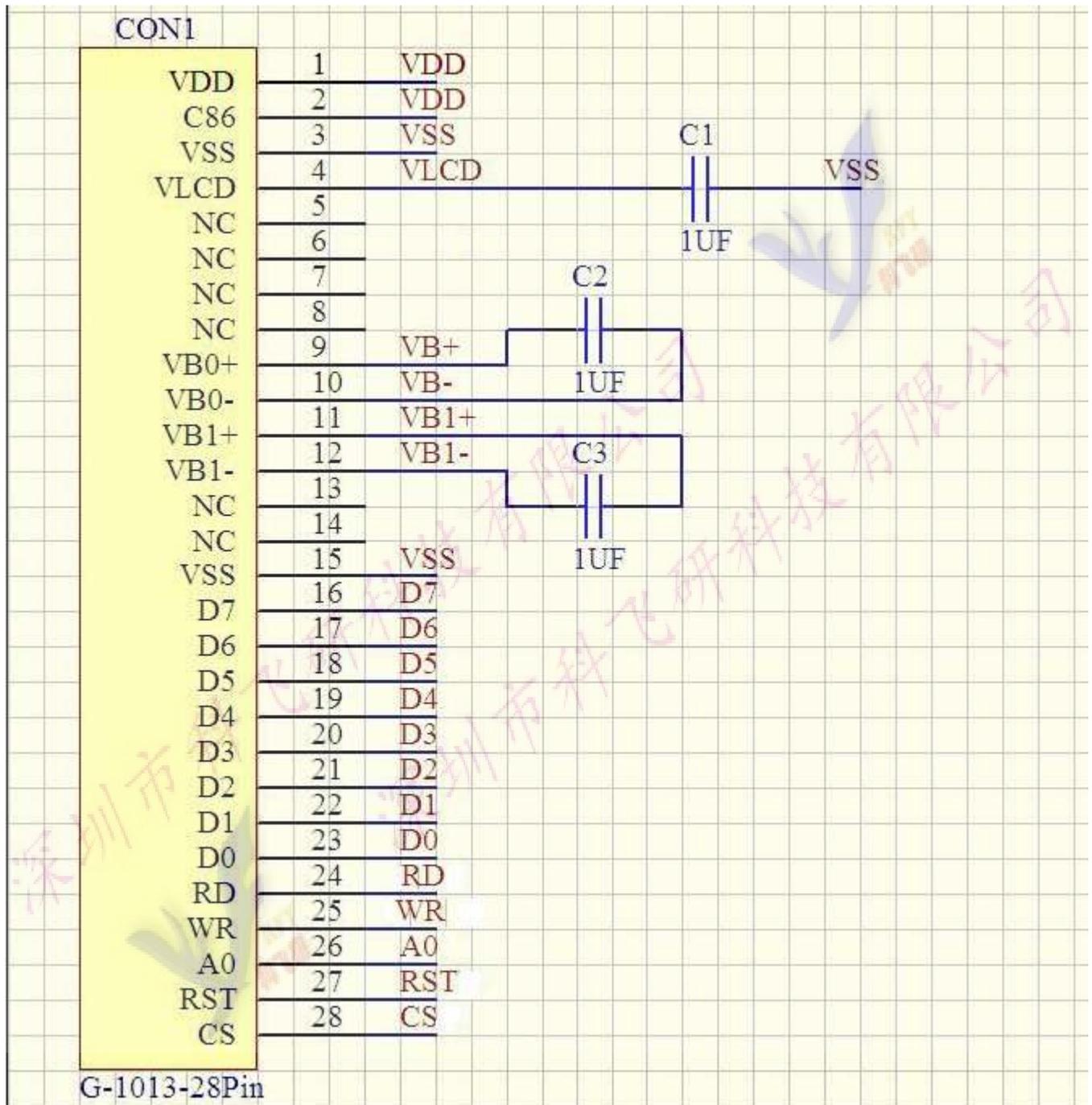


图 8. 并行接口



电容 C9/C1 请使用 16V 电压, 其他电容请使用 10V 电压。

```
/* Test program for COG12864C1013, 并行接口
```

```
   驱动 IC 是:UC1701x(or competible)
```

```
*/
```

```
#include <reg51.H>
```

```
sbit rs=P3^2;    /*接口定义:lcd_rs 就是 LCD 的 rs*/
```

```
sbit rd=P3^0;    /*接口定义:lcd_e 就是 LCD 的 rd*/
```

```
sbit wr=P3^1;    /*接口定义:lcd_rw 就是 LCD 的 wr*/
```

```
sbit reset=P3^4; /*接口定义:lcd_reset 就是 LCD 的 reset*/
```

```
sbit cs1=P3^5; /*接口定义:lcd_cs1 就是 LCD 的 cs1*/

void transfer_data(int data1);
void transfer_command(int data1);
char code graphic1[];
char code graphic2[];
char code graphic3[];
char code graphic4[];
char code graphic5[];
char code graphic6[];
void delay(int i);
void Delay1(int i);
void disp_grap(char *dp);
void initial_lcd();
void clear_screen();
void waitkey();

//=====main program=====
void main(void)
{
    initial_lcd();
    while(1)
    {
        clear_screen(); //clear all dots
        disp_grap(graphic1); //display a picture of 128*64 dots
        waitkey();
        disp_grap(graphic2); //display a picture of 128*64 dots
        waitkey();
        disp_grap(graphic4); //display a picture of 128*64 dots
        waitkey();
        disp_grap(graphic5); //display a picture of 128*64 dots
        waitkey();
        disp_grap(graphic6); //display a picture of 128*64 dots
        waitkey();
    }
}

//=====initial
void initial_lcd()
{
    reset=0; /*低电平复位*/
    delay(20);
    reset=1; /*复位完毕*/
}
```

```

delay(20);
transfer_command(0xe2); /*软复位*/
delay(5);
transfer_command(0x2c); /*升压步聚 1*/
delay(5);
transfer_command(0x2e); /*升压步聚 2*/
delay(5);
transfer_command(0x2f); /*升压步聚 3*/
delay(5);
transfer_command(0x24); /*粗调对比度，可设置范围 0x20~0x27*/
transfer_command(0x81); /*微调对比度*/
transfer_command(0x1B); /*微调对比度的值，可设置范围 0x00~0x3f*/
transfer_command(0xa2); /*1/9 偏压比 (bias) */
transfer_command(0xc0); /*行扫描顺序：从上到下*/
transfer_command(0xa1); /*列扫描顺序：从左到右*/
transfer_command(0x40); /*起始行：第一行开始*/
transfer_command(0xaf); /*开显示*/
}

```

//=====clear all dot martrices=====

```

void clear_screen()
{
    unsigned char i, j;
    for(i=0; i<9; i++)
    {
        cs1=0;
        transfer_command(0xb0+i);
        transfer_command(0x10);
        transfer_command(0x00);
        for(j=0; j<132; j++)
        {
            transfer_data(0x00);
        }
    }
}

```

//=====display a picture of 128\*64 dots=====

```

void disp_grap(char *dp)
{
    int i, j;
    for(i=0; i<8; i++)
    {
        cs1=0;
        transfer_command(0xb0+i); //set page address,
        transfer_command(0x10);
        transfer_command(0x00+4);
    }
}

```

```
        for(j=0; j<128; j++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
}

//=====transfer command to LCM=====
void transfer_command(int data1)
{
    cs1=0;
    rs=0;
    rd=0;
    wr=0;
    P1=data1;
    rd=1;
    cs1=1;
    rd=0;
}

//-----transfer data to LCM-----
void transfer_data(int data1)
{
    cs1=0;
    rs=1;
    rd=0;
    wr=0;
    P1=data1;
    rd=1;
    cs1=1;
    rd=0;
}

//=====delay time=====
void delay(int i)
{
    int j,k;
    for(j=0; j<i; j++)
    for(k=0; k<990; k++);
}

//=====delay time=====
void Delay1(int i)
{
```

```
int j,k;
for(j=0;j<i;j++)
for(k=0;k<10;k++);
}

//-----wait a switch, jump out if P2.0 get a signal"0"-----
void waitkey()
{
repeat:
    if (P2&0x01) goto repeat;
    else delay(6);
    if (P2&0x01) goto repeat;
    else
    delay(40);;
}
char code graphic1[]={
/*-- 调入图像：。。。 --*/
/*-- 宽度 x 高度=128x64 --*/

    略

};
```

