B 系列液晶显示控制器使用手册（V1203）

目录

一、B 系列产品介绍

B 系列总线型液晶显示控制模块是世龙电子推出的适用于工控和仪器仪表行业的显示控制模块，其核心电路采用 XILLIX 公司的大规模可编程集成电路（FPGA）编程实现，性能稳定可靠。B 系列所有产品均为 LCD 真彩液晶屏套件，且显示屏中每个点影射显示缓存中的一个字，只需输入 XY 坐标，便可直接读写相应点数据，不用计算点在显示缓存中的位置。提高了读写速度、简化了程序编写。

1、产品特性

- 可通过单片机编程控制显示。适配 MCU：51，96，X86，8088，Z80，DSP，ARM。
- 支持单点、多点和 8 点写屏方式，便于字符操作，提高显示速度，使显示多样化。多点和 8 点写屏方式，写一个字节，对应屏中 8 个点，比单点写入方式速度快很多。
- 自动填充、拷贝图形图像，自动显示光标鼠标，方便用户操作和显示。
- 真彩液晶显示，显示色彩鲜艳，图像清晰
- 电阻式触摸输入，精度高，价格便宜，不易损坏
- 采用 I/O 或总线方式连接

2、命名规则

B 系列产品的命名包括了该产品的部分功能信息，随产品性能的增加，命名可能会改变。

B	43	42	W	T	_	256
①	②	③	④	⑤		⑥

① 表示产品为 B 系列
② 表示产品显示尺寸，一般由两位或三位数字组成，例如：43 即 4.3 寸,102 即 10.2 寸；
③ 表示产品的分辨率，42 即分辨率为 480×272；64 即分辨率为 640×480；84 即分辨率为 800×480；86 即分辨率为 800×600；
④ W 表示显示屏为宽屏即 16:9，无 W 则表示显示屏为正屏即 4:3。
⑤ T 说明该产品带有触摸屏 ，N 说明该产品不带触摸屏
⑥ 表示产品的彩色方式，256 表示彩色方式为 256 色，16 表示彩色方式为 65535 色。

示例：

B8086T_256：B 表示 B 系列产品，80 表示显示屏尺寸为 8.0 寸，86 表示显示屏分辨率为 800×600，不含 W 说明该型号产品为正屏，含有 W 说明该产品有触摸屏，256 表示彩色方式为 256 色。

3、产品详细列表

B 系列液晶显示控制模块按其尺寸可就将其分为七大类，从 4.3 寸到 10,2 寸不等。每一类中包含 4 种型号的产品，即 256 色不带触摸，256 色带触摸，65536 色不带触摸，65536 色带触摸，具体如下表：

型号	B4342WN_16 B4342WT_16 B4342WN_256 B4342WT_256	B5064N_16 B5064T_16 B5064N_256 B5064T_256	B6584WN_16 B6584WT_16 B6584WN_256 B6584WT_256	B7084WN_16 B7084WT_16 B7084WN_256 B7084WT_256	B8084WN_16 B8084WT_16 B8084WN_256 B8084WT_256	B8086N_16 B8086T_16 B8086N_256 B8086T_256	B10284WN_16 B10284WT_16 B10284WN_256 B10284WT_256
显示尺寸	4.3″	5.0″	6.5″	7.0″	8.0″	8.0″	10.2″
显示比例	16:9	4:3	16:9	16:9	16:9	4:3	16:9
分辨率	480×272	640×480	800×480	800×480	800×480	800×600	800×480
显示色彩	后缀为 16 的是 TFT16 位色（即 65535 彩色），后缀为 256 的是 TFT8 位色（即 256 彩色）						
显示页数	256 色为两页显存，65536 色为单页显存						
亮度	500cd/m^2	200cd/m^2	400cd/m^2	250cd/m^2	450cd/m^2	450cd/m^2	250cd/m^2
对比度	500	500	500	500	500	500	500
背光灯	LED	LED	LED	LED	LED	LED	LED
触控类型	型号名中含 T 的表示有触摸输入，含 N 的表示无触摸输入						
对外接口	20P FPC 接线器						
额定电压	直流 5V						
工作温度	-20℃ ～ 70℃						
储存温度	-30℃ ～ 80℃						
额定功率	1.25W	2.85W	2W	2.65W	2W	1.75W	4W
显示尺寸	97.1×55.9	101.6×76.2	145.4×78.7	155.7×89.1	179×102.36	164.8×124.3	222×132.4
开孔尺寸	115×66	128×95.3	163.4×94.7	172×107.2	201×122.3	193.8×133	244.8×135.8
重量	120g	200g	220g	275g	385g	445g	580g
读写周期	40ns						
备注	重量一栏为不带触摸屏的重量，有待进一步添加和更改。						

二、控制板接口定义

B 系列产品的对外接口为 20P 的 FPC 接线器或 20 针脚的双压排线（连接控制板与适配 MCU），具体定义如下：

引脚	符号	输入/输出	功能	备注
1	VCC	P	液晶屏逻辑电源	5V/400mA
2	VCC	P	液晶屏逻辑电源	5V
3	SINT	O	触摸屏触发中断，低有效	3.3V 上拉
4	DATA0	I/O	数据总线	3.3/5VCMOS 电平
5	DATA1	I/O	数据总线	3.3/5VCMOS 电平
6	DATA2	I/O	数据总线	3.3/5VCMOS 电平
7	DATA3	I/O	数据总线	3.3/5VCMOS 电平
8	DATA4	I/O	数据总线	3.3/5VCMOS 电平
9	DATA5	I/O	数据总线	3.3/5VCMOS 电平
10	DATA6	I/O	数据总线	3.3/5VCMOS 电平
11	DATA7	I/O	数据总线	3.3/5VCMOS 电平
12	CS	I	片选信号，低电平对屏操作有效	3.3/5VCMOS 电平
13	WR	I	写操作信号，低电平有效。	3.3/5VCMOS 电平
14	RD	I	读操作信号，低电平有效。	3.3/5VCMOS 电平
15	A3	I	寄存器地址	3.3/5VCMOS 电平
16	A0	I	寄存器地址	3.3/5VCMOS 电平
17	A1	I	寄存器地址	3.3/5VCMOS 电平
18	A2	I	寄存器地址	3.3/5VCMOS 电平
19	GND	P	电源地	
20	GND	P	电源地	

由于屏背光电源功耗较大，控制板与用户主板连线会有一定的压降，为保障控制板能正常工作，要注意 20P 连接线质量。

三、功能寄存器及其说明

B 系列产品的功能寄存器地址由 CS，WR、RD 与 A0、A1、A2、A3、HB/LB 组合而成，具体如下：

CS	A3A2A1A0	WR	RD	HB/LB（1）	寄存器	功能
写操作						
0	0000	1-0-1	1	0	XL	X 坐标低 8 位寄存器(3)
0	0001	1-0-1	1	0	XH	X 坐标高 8 位寄存器(3)
0	0010	1-0-1	1	0	YL	Y 坐标低 8 位寄存器(3)
0	0011	1-0-1	1	0	YH	Y 坐标高 8 位寄存器(3)
0	0100	1-0-1	1	0	FRONTL	前景色寄存器低 8 位（6）
0	0100	1-0-1	1	1	FRONTH	前景色寄存器高 8 位(2)（6）
0	0101	1-0-1	1	0	BACKL	背景色寄存器低 8 位
0	0101	1-0-1	1	1	BACKH	背景色寄存器高 8 位(2)
0	0110	1-0-1	1	0	DATAL	数据寄存器低 8 位
0	0110	1-0-1	1	1	DATAH	数据寄存器高 8 位(2)
0	0111	1-0-1	1	0	CONTROL0	功能控制寄存器 0
0	1000	1-0-1	1	0	XSL	复制源 X 坐标低 8 位寄存器（7）
0	1000	1-0-1	1	1	XSH	复制源 X 坐标高 8 位寄存器（7）
0	1001	1-0-1	1	0	YSL	复制源 Y 坐标低 8 位寄存器（7）
0	1001	1-0-1	1	1	YSH	复制源 Y 坐标高 8 位寄存器（7）
0	1010	1-0-1	1	0	RECLL	复制或填充矩形宽度低 8 位寄存器
0	1010	1-0-1	1	1	RECLH	复制或填充矩形宽度高 8 位寄存器
0	1011	1-0-1	1	0	RECHL	复制或填充矩形高度低 8 位寄存器
0	1011	1-0-1	1	1	RECHH	复制或填充矩形高度高 8 位寄存器
0	1100	1-0-1	1	0	XML	光标或鼠标 X 坐标低 8 位寄存器
0	1100	1-0-1	1	1	XMH	光标或鼠标 X 坐标高 8 位寄存器
0	1101	1-0-1	1	0	YML	光标或鼠标 Y 坐标低 8 位寄存器
0	1101	1-0-1	1	1	YMH	光标或鼠标 Y 坐标高 8 位寄存器
0	1110	1-0-1	1	0	BRI	背光亮度调节
0	1110	1-0-1	1	1	STIME	触摸响应时间间隔/横竖调节(4)
0	1111	1-0-1	1	0	CONTROL1	功能控制寄存器 1
0	1111	1-0-1	1	1	CURSOR	光标高度调节
读操作						
0	0110	1	0	0	BUSY	0x00:闲。0XFF 忙(5)
0	1000	1	0	0	SXL	触摸位置 X 低 8 位(4)
0	1001	1	0	0	SXH	触摸位置 X 高 8 位(4)
0	1010	1	0	0	SYL	触摸位置 Y 低 8 位(4)
0	1011	1	0	0	SYH	触摸位置 Y 高 8 位(4)
1	×	×	×	×		不选通

说明：

（1） CONTROL0 寄存器第 6 位。

（2） 16 位真彩控制时颜色值和数据的高位。

（3） 矩形填充左上角 XY 坐标，或块复制目标位置矩形左上角 XY 坐标。

（4） 有触摸屏功能的控制板可用。此寄存器的最高位为横屏竖屏显示控制位。

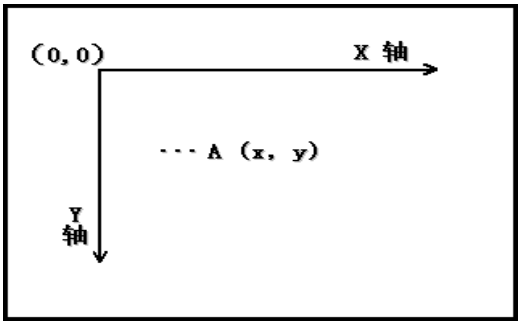
（5） 控制板正在进行块填充或块复制时，处于‘忙’状态，BUSY 值为 0XFF，结束后为 0X00。用户可通过读取该值判断忙闲。

（6） 块填充时以此值为填充颜色值。

（7） 块复制源位置矩形左上角 XY 坐标。

1. 坐标寄存器

世龙液晶显示控制器使用的坐标系与普通笛卡尔坐标系不同，而与通常的图像坐标系相同，即以可视区域的左上角作为坐标系原点，向右为 X 轴正向，向下为 Y 轴正向。如下图所示：



对于 B 系列控制板，根据其控制的 LCD 的分辨率为 M×N，所以我们将液晶屏的横向（X 轴）分为 M 个坐标点，纵向（Y 轴）分为 N 个坐标点。具体地址如下：

块填充左上角、块复制目标左上角 X 地址寄存器 XL，XH：地址：A=0000，0001
块复制源左上角 XS 地址寄存器 XSL，XSH：地址：A=1000，CONTROL0（6）
光标（顶）或鼠标（尖）XM 地址寄存器 XML，XMH：地址：A=1100，CONTROL0（6）

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0

块填充左上角、块复制目标左上角 Y 地址寄存器 YL，YH：地址：A=0010，A=0011
块复制源左上角 YS 地址寄存器 YSL，YSH：地址：A=1001，CONTROL0（6）
光标（顶）或鼠标（尖）YM 地址寄存器 YML，YMH：地址：A=1101，CONTROL0（6）

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

2. 颜色寄存器

B 系列控制板有两种颜色模式，分别是 RGB332 和 RGB565 模式。RGB332 即是用一个字节来表示一个像素点的值，其中红色分量(R)占 3 位、绿色分量(G)占 3 位、蓝色分量(B)占 2 位；RGB565 即是用两个字节来表示一个像素点的值，其中红色分量(R)占 5 位、绿色分量(G)占 6 位、蓝色分量(B)占 5 位。颜色寄存器分前景色寄存器和背景色寄存器两种，RGB332 模式时颜色寄存器的高字节无效，具体地址如下：

前景色寄存器 FRONTL，FRONTH:A=0100、control(6)
背景色寄存器 BACKL，BACKH:A=0101、control(6)
256 色（纯红=0xE0　纯绿＝0x1C　纯蓝＝0x03 写入颜色寄存器高字节）：

D7	D6	D5	D4	D3	D2	D1	D0
R2	R1	R0	G2	G1	G0	B1	B0
红色表示位			绿色表示位			蓝色表示位	

16 位真彩色（纯红=0xF800　纯绿＝0x07E0　纯蓝＝0x001F）

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
红色表示位					绿色表示位						蓝色表示位				

3. 功能控制寄存器

功能控制寄存器 CONTROL0：地址 A=1010

D7	D6	D5	D4	D3	D2	D1	D0
on/off	HB/LB	IP	WP	×	XINC	Write1	Write0

on/off：=1 背光开，=0 背光关(和总线控制板有区别)

HB/LB：用 HB/LB 控制位和 4 根地址线组合可写 24 个寄存器。

IP：显示页号，=0 为第 0 页，=1 为第 1 页。

WP：操作页号，=0 对第 0 页操作，=1 对第 1 页操作。

XINC：为 X 坐标自动增加控制位，=1 时允许 X 自动增加，写满一行后自动换行，=0 时则禁止增加。单点写屏时，X 自动加 1，多点（或 8 点）写屏时自动加 8；

Write1，Write0 为写入方式：

Write1，Write0＝00 ：单点写入方式，直接将颜色值写入数据寄存器，与前景色、背景色寄存器内容无关，每次写入一个点；单点写屏时，必须先写高字节(A=1001)，再写低字节(A=1000)。

Write1，Write0＝01 ：多点写入方式，将点位信息写入数据寄存器，例如：写入数据寄存器的数据为‘01010101B’则表示一次显示的 8 个点的颜色依此是‘原色、前景色、原色、前景色、原色、前景色、原色、前景色’(即 0 表示原色，1 表示前景色)。

Write1，Write0＝10 ：8 点写入方式，将点位信息写入数据寄存器，例如：写入数据寄存器的数据为‘01010101B’则表示一次显示的 8 个点的颜色依此是‘背景色、前景色、背景色、前景色、背景色、前景色、背景色、前景色’(即 0 表示背景色，1 表示前景色)。

多点和 8 点写入方式，一次可写入 8 个点，适用于写字符和块清屏，将点位信息写入数据寄存器。原色：显示屏原有颜色，前景色、背景色是事先存入前景、背景寄存器中的颜色值。

单点写屏时，将颜色值写入该寄存器，多点写屏时，将点位信息写入该寄存器，如字符点阵信息，字符颜色为前景寄存器值。8 点写屏时，字符颜色为前景寄存器值，字符背景颜色为背景寄存器值。

如下图，显示屏原有颜色是一幅照片，“多点”两个字是用多点写入方式写入的，只写前景色蓝色，不写背景色；而“8 点” 两字是用 8 点写入方式写入的，前景色为绿色，背景色为白色，前景和背景同时写入。

功能控制寄存器 CONTROL1：地址 A=1111，CONTROL0（6）=0

D7	D6	D5	D4	D3	D2	D1	D0
0	Cross/cursor	SP	DP	m/c	mcon/off	copyset	fillset

控制板共有两页显存，可进行页内或页之间复制，

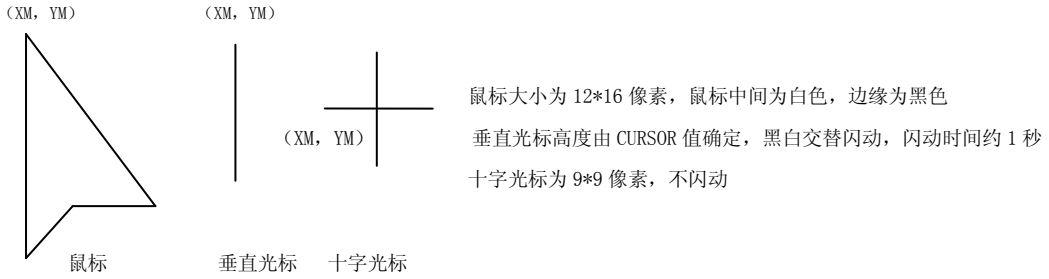
SP（source page）:复制源页号；

DP（dest page）:复制目标页号；

例如：SP，DP=00: 0 页内容复制到 0 页；=10：1 页内容复制到 0 页，矩形位置由 XS，YS，X，Y 确定，矩形大小由 RECL，RECH 确定。

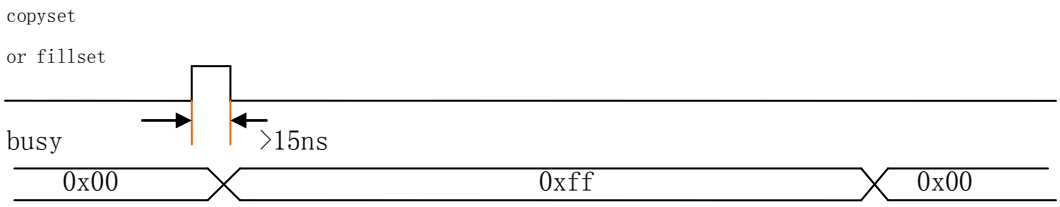
m/c: 光标和鼠标不能同时显示，=1 显示鼠标，=0 显示垂直光标或十字光标。鼠标实心为白色，边缘为黑色如下图所示，XM，YM 为鼠标尖端处。

Cross/cursor: 光标有两种，垂直线和十字，=1 为十字光标，=0 为垂直光标，垂直线光标长度由寄存器 cursor 值确定，线宽为 1 个像素点，XM，YM 为垂直线顶端，黑白交替闪动，闪动时间约 1 秒左右。十字光标实心为白色水平直线和垂直线（线长 = 9 个像素点），白线边缘为黑边，不闪动，XM，YM 为十字交叉点。



Mcon/off: 光标和鼠标显示开关控制，=1 显示光标或鼠标，=0 不显示光标和鼠标

Copyset、fillset:块复制和块填充启动位，如下图：



Copyset,fillset 启动宽度>15ns。

块填充：写入 XL，XH，YL，YH，WP，RECL，RECH，FRONT，Write1 Write0=11，将 fillset 置 1，再复位，控制板自动填充，填充色为 FRONT 值，busy=0xff 为忙，再次为 0x00 时填充完毕。填充速度为 160ns（180ns，16 位真彩）一个像素点。当 RECL=1 时，为一条垂直线，当 RECH=1 时为一条水平线。

块复制：写入 XS，YS，X，Y，RECL，RECH，SP，DP，Write1 Write0=11，将 copyset 置 1 再复位，控制板自动复制，busy 为 1 时忙，再次为 0x00 时复制完毕。复制速度为 320ns（360ns，16 位真彩）一个像素点。

4. 数据寄存器

数据寄存器 DATAL，DATAH：地址 A=0110、control(6)，先写高，后写低。单点写屏时数据寄存器写入的数据是显示的颜色值，多点和 8 点写屏时写入数据寄存器的数据是对应点位显示颜色的控制字。

5. **矩形宽、高寄存器**

块填充、块复制矩形宽度 RECL 地址寄存器 RECLL，RECLH：地址：A=1010，CONTROL0（6）

块填充、块复制矩形高度 RECH 地址寄存器 RECLL，RECLH：地址：A=1011，CONTROL0（6）

矩形宽度寄存器 RECL

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	L9	L8	L7	L6	L5	L4	L3	L2	L1	L0

矩形高度寄存器 RECH

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	H9	H8	H7	H6	H5	H4	H3	H2	H1	H0

6. **亮度调节寄存器 BRI**

亮度调节 BRI 寄存器：A=1110，CONTROL0（6）= 0，亮度调节范围为 0X00～0X3F，为了向上兼容性，0x00 最亮，以下顺序是 0x3f、0x3e、0x3d、...、0x01。

7. **光标高度调节寄存器**

光标高度调节 CURSOR 寄存器，A=1111，CONTROL0（6）= 1，光标为一条黑白交替闪动的垂直线，闪动时间约 1 秒左右，光标高度调节范围为 0X01～0X3F，光标显示位置由 XM 和 YM 决定。光标和鼠标只能同时显示一个。鼠标大小不可调。

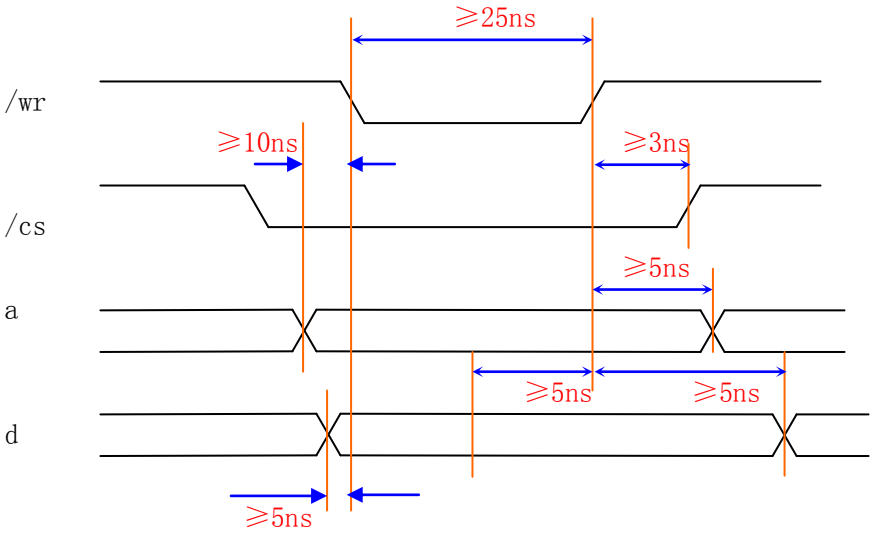
8. **触摸响应时间间隔/横竖调节寄存器 STIME：**

触摸响应时间间隔/横竖调节寄存器 STIME：地址 A=1110，CONTROL0（6）=1，寄存器的最高位 VH 为显示屏的横竖显示控制位，VH=0 时横屏显示，VH=1 时竖屏显示，其他位为触摸响应时间间隔设置位。

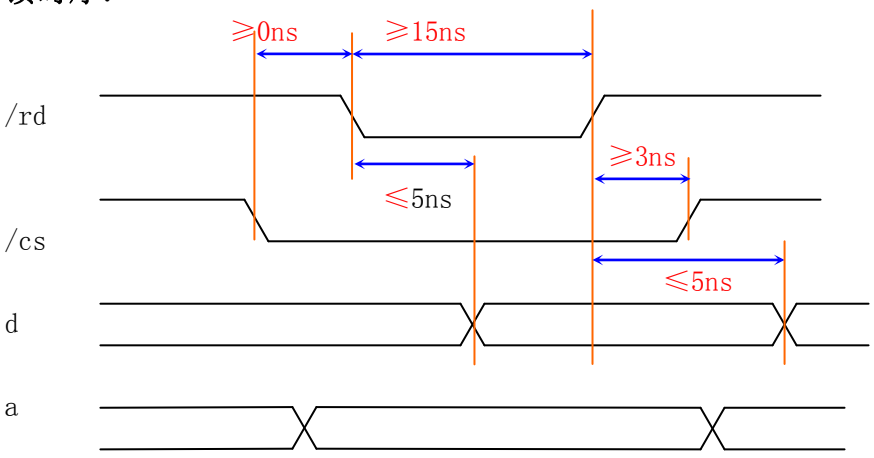
D7	D6	D5	D4	D3	D2	D1	D0
VH	时间间隔数据位						

四、CPU 接口时序

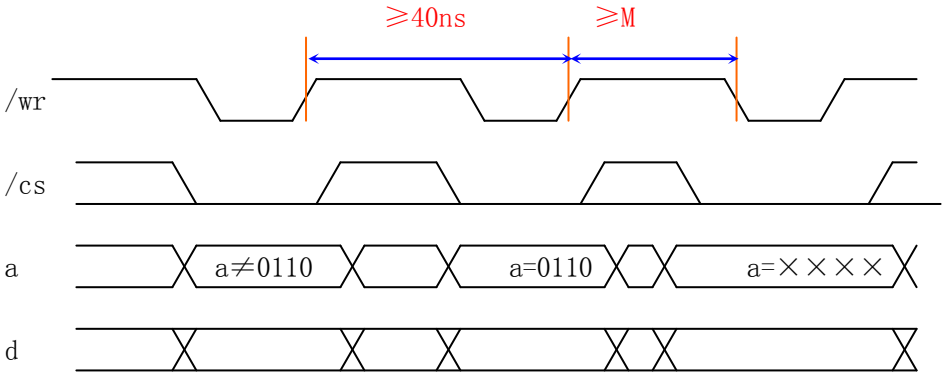
写时序：



读时序：



单点写(M=180ns)，多点或 8 点写（M=1440ns)：



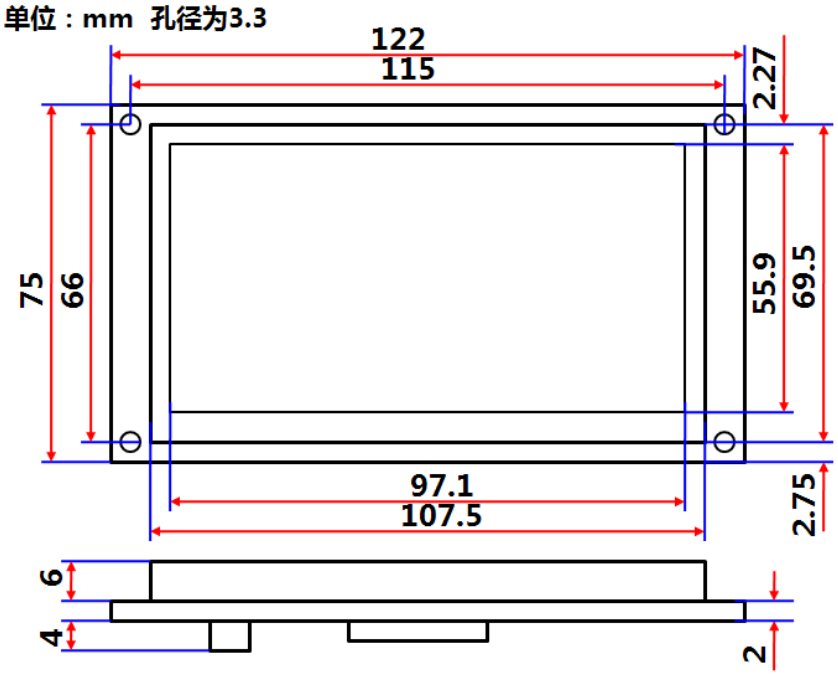
符号	参数说明	最小	最大	单位
T1	地址建立时间	0	-	ns
T3	地址保持时间	5	-	ns
T2	读写同期	20	-	ns
T4	读写脉冲宽度	20	-	ns
T5	写数据建立时间	25	-	ns
T7	写数据保持时间	10	-	ns
T6	读数据建立时间	5	-	ns
T8	读数据保持时间	10	-	ns

****除 DATAL 寄存器，其他寄存器写入间隔只要大于 40ns，由于新板采用 SDRAM，写数据寄存器（DATAL）对时间间隔要求更大一些，单点写入方式时，写完 DATAL 之后的 180ns 不要对控制板进行任何写操作，多点或 8 点写入 DATAL 之后的 1440ns 不要对控制板进行任何写操作。

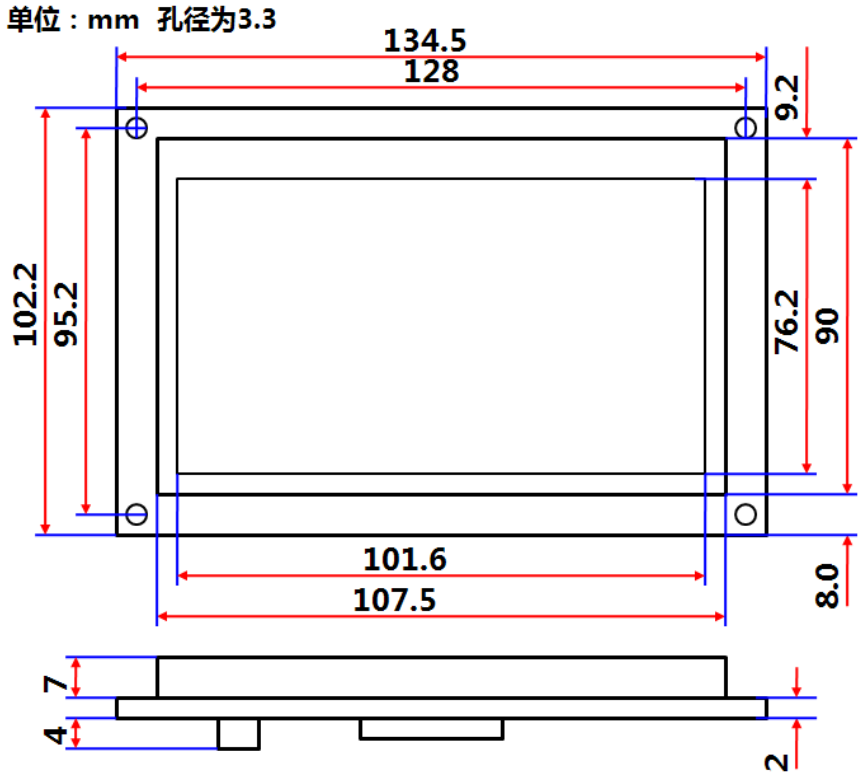
五、控制模块尺寸

B 系列液晶显示控制模块，其结构采用套件方式组装，用铁框以卡扣的方式将控制板、液晶屏和触摸屏固定在一起，外贴防尘防划伤膜。具体尺寸如下：

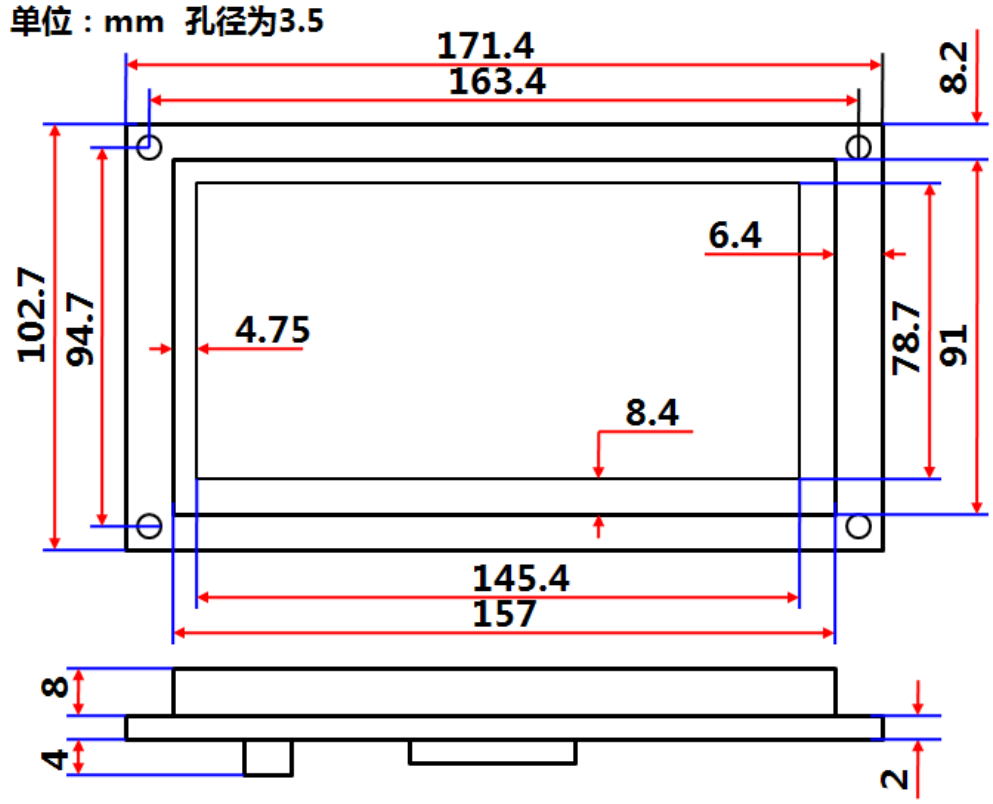
B4342WN_16　B4342WT_16　B4342WN_256　B4342WT_256 尺寸图



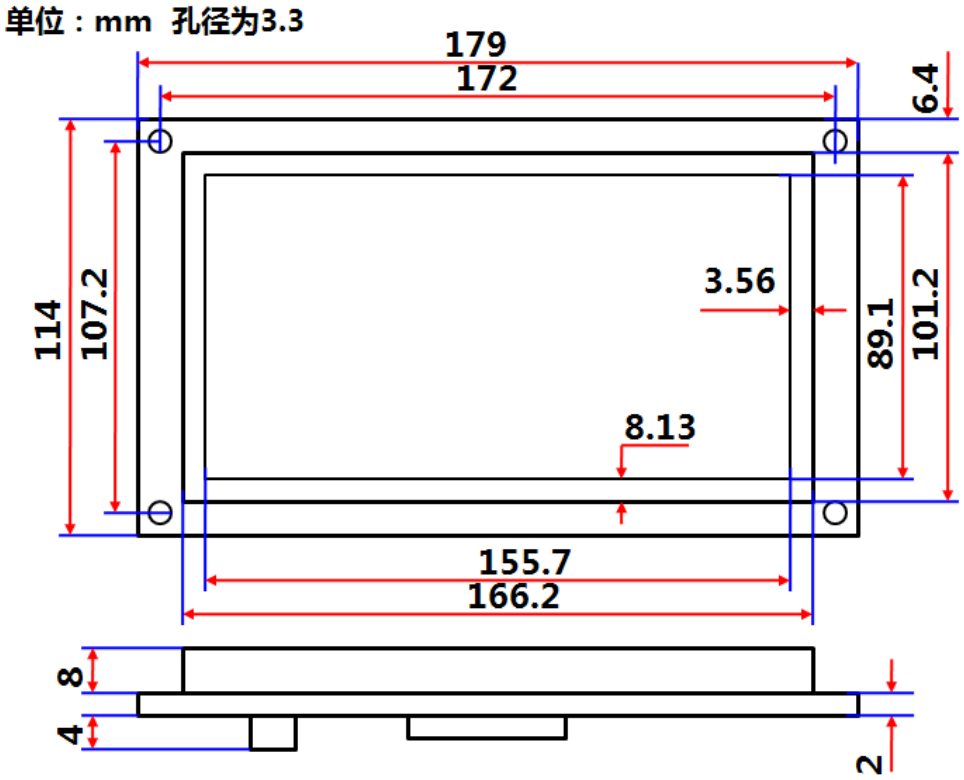
B5064N_16　B5064T_16　B5064N_256　B5064T_256 尺寸图



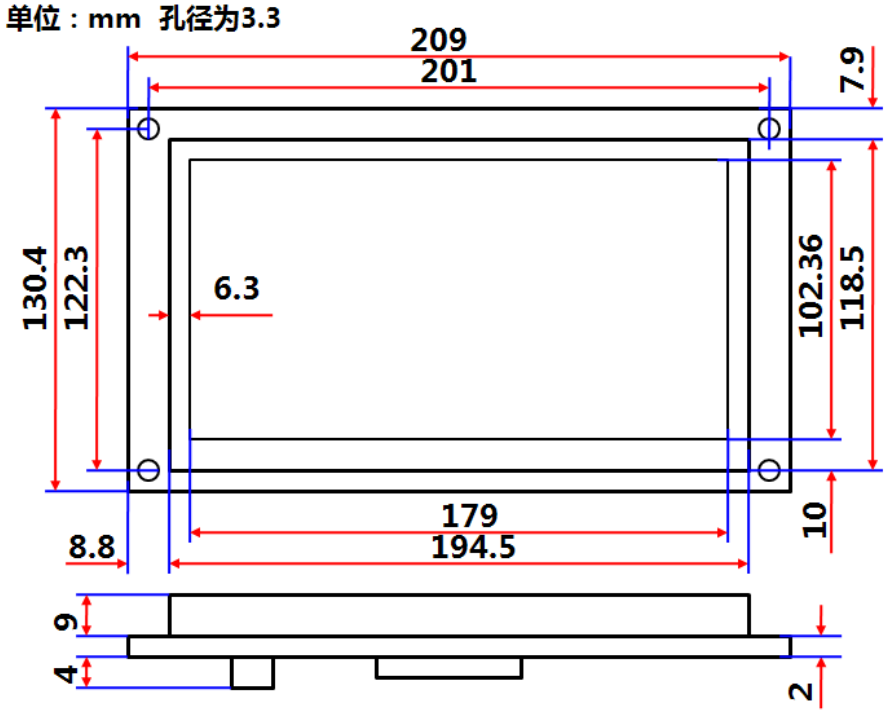
B6584WN_16　B6584WT_16　B6584WN_256　B6584WT_256 尺寸图



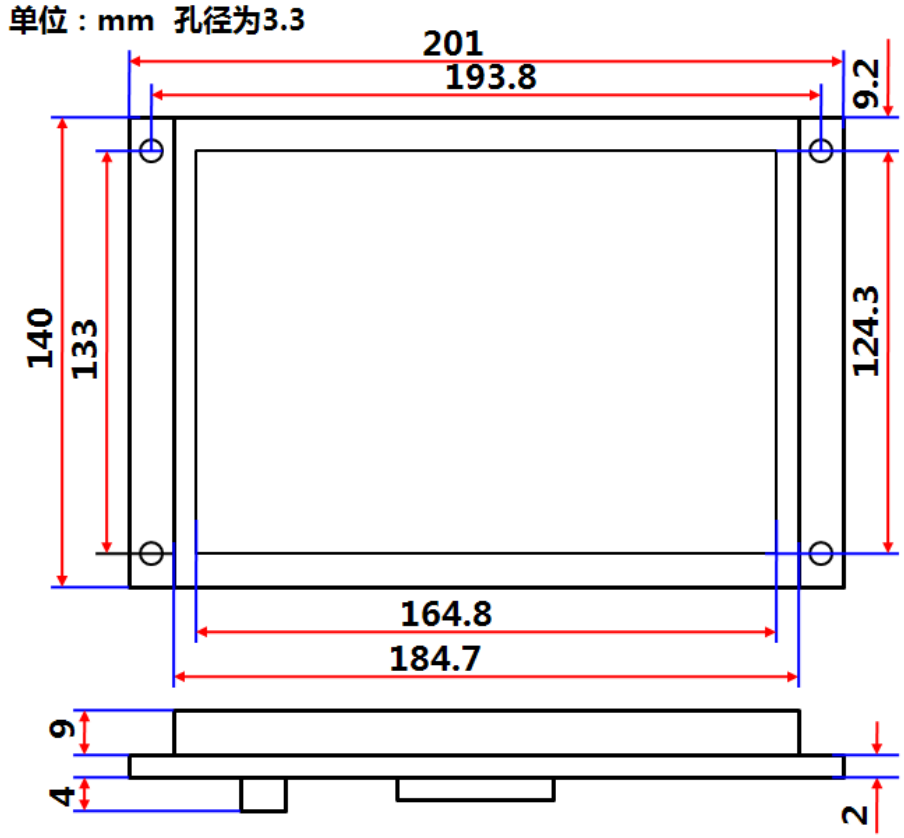
B7084WN_16　B7084WT_16　B7084WN_256　B7084WT_256 尺寸图



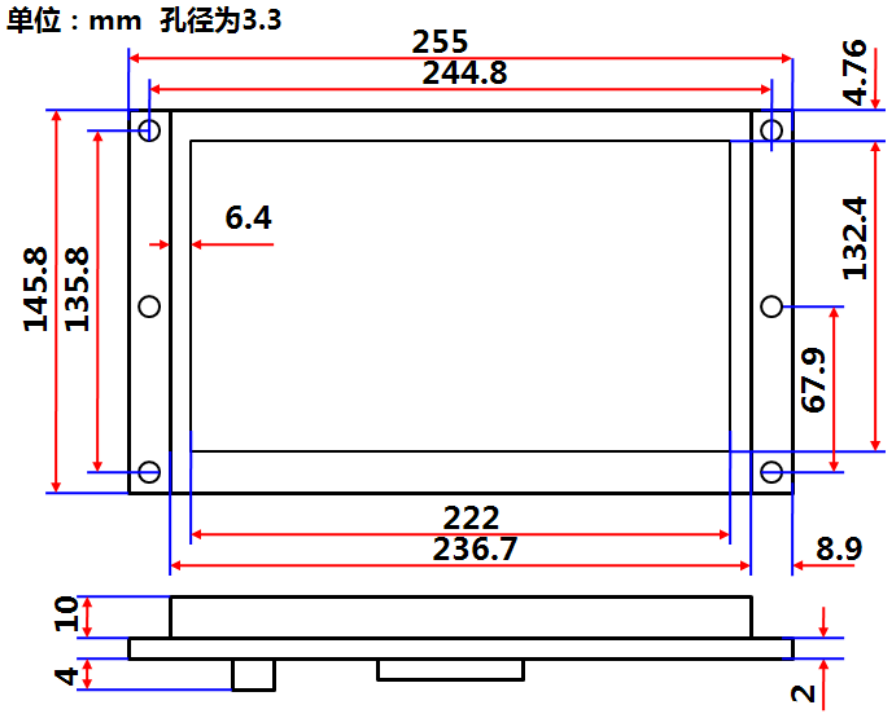
B8084WN_16　B8084WT_16　B8084WN_256　B8084WT_256 尺寸图



B8086N_16　B8086T_16　B8086N_256　B8086T_256 尺寸图



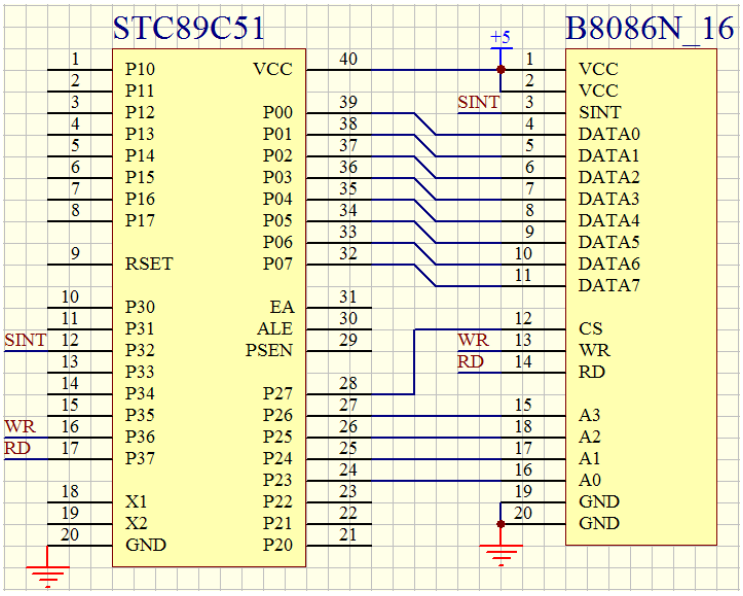
B10284WN_16　B10284WT_16　B10284WN_256　B10284WT_256 尺寸图



六、应用程序

以下将提供 8051 和 ARM 两种适配 CPU 和 B8086N_16 之间简单的控制显示的演示程序。

1、 C51 演示程序：C51 和 B8086N_16 之间数据的传输可以用总线（并口）的方式，也可以用串口的方式，总线方式比 I/O 方式速度要快很多，建议用户用总线方式，本示例程序是用总线的方式传输数据的，C51 和 B8086N_16 的接线图如下，其中 C51 的外围电路请用户自行补足，另外用户也可根据各自的工作方便，自行配置接线：



C51 演示程序（演示程序的显示结果见 20 页）

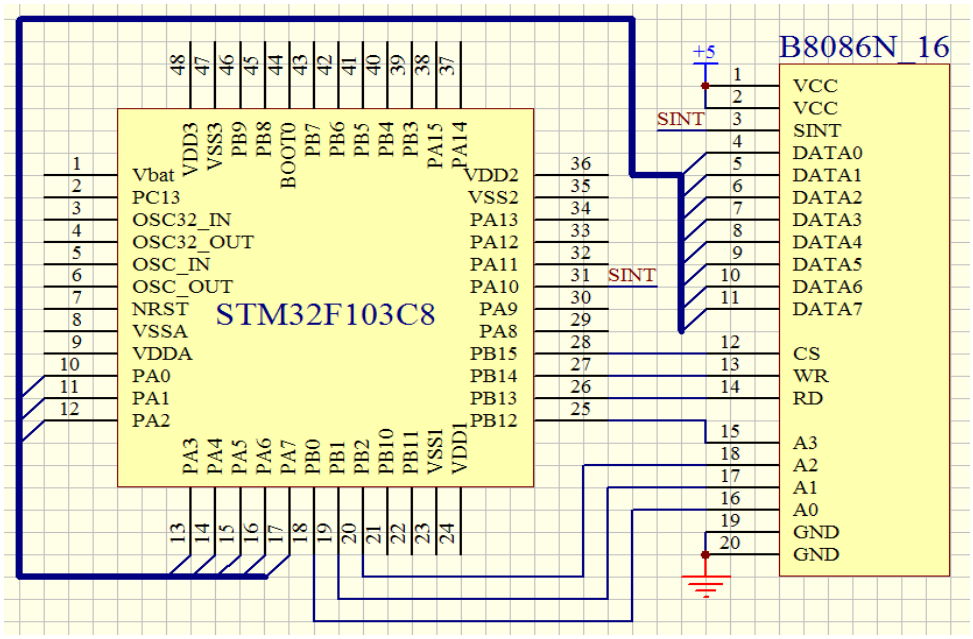
ziku.h 文件中包括：48×48 点阵的汉子“世龙电子”，24×24 点阵的汉子“上下一页校准”和 64×48 点阵的字符“ ”

```
#include <reg52.h>
#include <ziku.h>
#define uchar unsigned char
#define uint unsigned int
#define R 0xf800    //红色
#define G 0x07e0    //绿色
#define B 0x001f    //蓝色
#define WHITE 0xffff   //白色
#define BLACK 0x0000   //黑色
#define V   800/40       //控制板的横向分辨率为 800
#define H   600/40       //控制板的纵向分辨率为 600

uchar xdata *Xl=(uchar *)0x0000;                  //a=0000
uchar xdata *Xh=(uchar *)0x0800;                  //A=0001
uchar xdata *Yl=(uchar *)0x1000;                  //A=0010
uchar xdata *Yh=(uchar *)0x1800;                  //a=0011
uchar xdata*front=(uchar *)0x2000;                      //A=0100
uchar xdata*back=(uchar *)0x2800;                 //A=0101
uchar xdata *Data=(uchar *)0x3000;                      //A=0110
uchar xdata *Control0=(uchar *)0x3800;            //A=0111
uchar xdata *Xs=(uchar *)0x4000;                  //a=1000
uchar xdata *Ys=(uchar *)0x4800;                  //A=1001
uchar xdata *L0=(uchar *)0x5000;                        //A=1010
uchar xdata *H0=(uchar *)0x5800;                  //a=1011
uchar xdata*Xm=(uchar *)0x6000;                   //A=1100
uchar xdata*Ym=(uchar *)0x6800;                   //A=1101
uchar xdata *Bri=(uchar *)0x7000;                 //A=1110
uchar xdata *Control1=(uchar *)0x7800;            //A=1111

//延迟 time=1 为 0.7ms 子程序
void delay (uint time)
{
  uint a,b;
  for(a=0;a<time;a++){ for(b=0;b<88;b++);}
}
//画实心矩形
void drawrectangle(uint color,uint x0,uint y0,uint l,uint h,uchar page)
{
        *front=color;
        *Xl=x0;                                                 //写矩形左上角 XY 坐标
        *Xh=x0>>8;
        *Yl=y0;
        *Yh=y0>>8;
        *L0=l;
        *H0=h;
        *Control0=0x40|page;                    //control0(6)=1
        *front=color>>8;
        *L0=l>>8;
        *H0=h>>8;
        *Control0=0x03|page;                    //control0(6)=0,control(1,0)='11'
        *Control1=0x01;                                 //fillset=1
        *Control1=0;                                    //fillset=0
        while(*Data!=0){while(*Data!=0);}  //不是连续两次为 0，则重新判别
        *Control0=page;
}
//用画实心矩形方式画横线，高度为 0x01
void drawline(uint color,uint x0,uint y0,uint x1)
{
        uint i;
        i=x1-x0;
        drawrectangle(color,x0,y0,i,0x01,0x83);
}
//用画实心矩形方式画垂直线，矩形宽度为 0x01
void drawline_v(uint color,uint x0,uint y0,uint y1)
{
        uint i;
        *Control0=0x83;
        i=y1-y0;
        drawrectangle(color,x0,y0,0x01,i,0x80);
}
//写字符,多点或 8 点写屏方式  w=字符宽度/8
void writeword(uint fcolor,uint bcolor,uint x,uint y,uchar w,uchar h,uchar *pData,uchar page)
{
        uchar i,j;uint n=0;
        *Control0=page|0x40;*front=fcolor>>8;*back=bcolor>>8;
        *Control0=page;*front=fcolor;*back=bcolor;
        for(j=0;j<h;j++)
        {
                *Yl=y;*Yh=y>>8;
```

```
            y++;*Xl=x;*Xh=x>>8;
            for(i=0;i<w;i++)
            {
                *Data=pData[n++];                    //一次写入 8 个点
            }
        }
}
// 画按钮 长=l  宽=h  边宽=w     mode：按钮的凹凸 =0 凸，=1 凹
void drawbutton(uchar color,uint x0,uint y0,uint l,uint h,uint w,uchar mode)
{
    uint i,lcolor,dcolor;
      if(mode==0) {lcolor=0x841f;dcolor=0x0010;}  //16 位纯蓝的亮暗边色
      else {lcolor=0x0010;dcolor=0x841f;}
      drawrectangle(lcolor,x0,y0,l,w,0x80);
      drawrectangle(lcolor,x0,y0,w,h,0x80);
      for(i=1;i<=w;i++)
      {
          drawrectangle(dcolor,x0+i,y0+h-i,l-i,1,0x80);
          drawrectangle(dcolor,x0+l-i,y0+i,1,h-i,0x80);
      }
}
// 界面 0
void interface0()
{
    drawrectangle(B,0,0,V*40,H*40,0x83); //全屏清蓝色
      drawrectangle(WHITE,V*2,H*4,V*25,H*25,0x83);  //白色矩形
      drawbutton(B,V*1,H*2,V*27,H*29,H/2,0) ;
      drawbutton(B,V*29,H*2,V*10,H*29,H/2,1) ;

      drawbutton(B,V*6,H*33,V*6,H*4,H/2,0); //按钮 8 点写 "上一页"
      writeword(BLACK,WHITE,V*6+(V*6-75)/2,H*33+(H*4-24)/2,3,24,shang,0x85);
      writeword(BLACK,WHITE,V*6+(V*6-75)/2+25,H*33+(H*4-24)/2,3,24,yi,0x85);
      writeword(BLACK,WHITE,V*6+(V*6-75)/2+50,H*33+(H*4-24)/2,3,24,ye,0x85);

      drawbutton(B,V*17,H*33,V*6,H*4,H/2,0);  //按钮 8 点写 "下一页"
      writeword(BLACK,WHITE,V*17+(V*6-75)/2,H*33+(H*4-24)/2,3,24,xia,0x85);
      writeword(BLACK,WHITE,V*17+(V*6-75)/2+25,H*33+(H*4-24)/2,3,24,yi,0x85);
      writeword(BLACK,WHITE,V*17+(V*6-75)/2+50,H*33+(H*4-24)/2,3,24,ye,0x85);

      drawbutton(B,V*31,H*5,V*6,H*4,H/2,0);  //按钮 多点写 'WL'
      writeword(BLACK,WHITE,V*31+(V*6-64)/2,H*5+(H*4-48)/2,8,48,wl,0x85);
      writeword(R,BLACK,V*31+(V*6-64)/2,H*5+(H*4-48)/2,8,48,huan,0x85);

      drawbutton(B,V*31,H*10,V*6,H*4,H/2,0); //按钮 多点写 '世'
      writeword(BLACK,WHITE,V*31+(V*6-48)/2,H*10+(H*4-48)/2,6,48,shi,0x85);
      drawbutton(B,V*31,H*15,V*6,H*4,H/2,0); //按钮 多点写 '龙'
      writeword(BLACK,WHITE,V*31+(V*6-48)/2,H*15+(H*4-48)/2,6,48,lon,0x85);
      drawbutton(B,V*31,H*20,V*6,H*4,H/2,0); //按钮 多点写 '电'
      writeword(BLACK,WHITE,V*31+(V*6-48)/2,H*20+(H*4-48)/2,6,48,dian,0x85);
      drawbutton(B,V*31,H*25,V*6,H*4,H/2,0); //按钮 多点写 '子'
      writeword(BLACK,WHITE,V*31+(V*6-48)/2,H*25+(H*4-48)/2,6,48,zi,0x85);

      drawbutton(B,V*31,H*33,V*6,H*4,H/2,0); //按钮 8 点写 "校准"
      writeword(BLACK,WHITE,V*31+(V*6-50)/2,H*33+(H*4-24)/2,3,24,jiao,0x85);
      writeword(BLACK,WHITE,V*31+(V*6-50)/2+25,H*33+(H*4-24)/2,3,24,zhun,0x85);
}
//画点
void writepoint(uint color,uint x,uint y)
{
    *Xl=x;*Xh=x>>8;
      *Yl=y;*Yh=y>>8;
      *Control0=0xc0;
      *Data=color>>8;
      *Control0=0x80;
      *Data=color;  //写色值
}
// 显示鼠标 光标 CONTROL1=0x0c(鼠标) =0x44(十字光标) =0x04(|字光标)
void displaycursou(uint x,uint y,uint psw)
{
      *Control1=psw;
      *Xm=x;*Ym=y;
      *Control0=0xc0;
      *Xm=x>>8;*Ym=y>>8;
      *Control0=0x80;
}
//十字光标演示--光标移动一次画一个，演示效果为在屏用十字光标画一个 'WL'
void draw_WL(uint color)
{
      uint i,x,y,time;
      uchar j,k;
      time=5;
      x=2*V+(25*V-20*H-2*V)/2;y=4*H+5*H;
      for(k=0;k<2;k++)                //write "W"
      {
```

```
            for(i=0;i<3*H;i++)
            {
                for(j=0;j<5;j++)
                {
                    displaycursou(x,y,0x44);
                    writepoint(color,x,y);      //写点
                    writepoint(color,x+1,y);
                    y++;delay(time);            //为增加演示效果，延时
                }x++;
            }
            for(i=0;i<3*H;i++)
            {
                for(j=0;j<5;j++)
                {
                    displaycursou(x,y,0x44);
                    writepoint(color,x,y);      //写点
                    writepoint(color,x+1,y);
                    y--;delay(time);            //为增加演示效果，延时
                }x++;
            }
        }
        x=x+2*V;
        for(i=0;i<15*H;i++)                     //write "L"
        {
            displaycursou(x,y,0x44);
            writepoint(color,x,y);      //写点
            writepoint(color,x+1,y);
            y++;delay(time);            //为增加演示效果，延时
        }
        for(i=0;i<8*H;i++)
        {
            displaycursou(x,y,0x44);
            writepoint(color,x,y);      //写点
            writepoint(color,x,y+1);
            x++;delay(time);        //为增加演示效果，延时
        }
}
void main()
{
        delay(3);
        *Bri=0x3f;    //亮度最亮
        interface0();//界面 0
        draw_WL(B);   //十字光画一个 'WL'
        while(1);
 }
```

2、 ARM 演示程序：选用的 ARM 型号为 STM32F103C8，采用 GPIO 端口的方式通信数据，其接线方式和工程文件菜单如下图所示，其中 STM32F103C8 的外围电路请用户自行补足，另外用户也可根据各自的工作方便，自行配置引脚接线：



Stm32 演示程序（演示程序的显示效果见 20 页）

```
#include "stm32f10x_lib.h"
#include "B8086.h"
#include "ziku.h"
#define      uchar unsigned char
#define      uint unsigned int

#define   R   0xf800          //红色
#define   G   0x07e0   //绿色
#define   B   0x001f          //蓝色
#define   WHITE   0xffff    //白色
#define   BLACK   0x0000   //黑色
#define V   800/40          //控制板的横向分辨率为 800
#define H   600/40          //控制板的纵向分辨率为 600

/* A3A2A1A0，CONTROL0(6)=0 表示 FRONT、BACK 、DATA、XS、YS、RECL、RECH、XM、YM 的低八位 */
/* A3A2A1A0，CONTROL0(6)=1 表示 FRONT、BACK 、DATA、XS、YS、RECL、RECH、XM、YM 的高八位 */
/* A=1110，CONTROL0(6)=1 表示触摸响应时间寄存器 */
/* A=1111，CONTROL0(6)=1 表示光标高度调节寄存器 */
#define XL              0x0000     // A = 0000 X 坐标低 8 位寄存器
#define XH              0x0001     // A = 0001 X 坐标高 8 位寄存器
#define YL              0x0002     // A = 0010 Y 坐标低 8 位寄存器
#define YH              0x0003      // A = 0011 Y 坐标高 8 位寄存器
#define FRONT           0x0004     // A = 0100 前景色寄存器
#define BACK            0x0005     // A = 0101 背景色寄存器
#define DATA            0x0006     // A = 0110 数据寄存器
#define CONTROL0    0x0007     // A = 0111 功能控制寄存器 0
#define XS              0x1000     // A = 1000 复制源 X 坐标寄存器
#define YS              0x1001     // A = 1001 复制源 Y 坐标寄存器
#define RECL            0x1002     // A = 1010 复制或填充矩形宽度寄存器
#define RECH            0x1003     // A = 1011 复制或填充矩形高度寄存器
#define XM              0x1004     // A = 1100 光标或鼠标 X 坐标寄存器
#define YM              0x1005      // A = 1101 光标或鼠标 Y 坐标寄存器
#define BRI             0x1006      // A = 1110 背光亮度调节寄存器
#define CONTROL1    0x1007     // A = 1111 功能控制寄存器 1
//延时
void delay(uint time)
{
        uint i,j;
        for(j=0;j<time;j++)
         for(i=0;i<8000;i++);
}
 //对控制板写数据
void writedata(uint reg,uint dat)
{
        uint pb;
        pb=0xEFF8|reg;     // PB12|PB2|PB1|PB0=0,其余置 1
        GPIOB->ODR=pb;
        GPIOA->ODR=dat;
        GPIOB->BRR=GPIO_Pin_15;              //CS=0
        GPIOB->BRR=GPIO_Pin_14;              //WR=0
        GPIOB->BSRR=GPIO_Pin_14;   //WR=1
        GPIOB->BSRR=GPIO_Pin_15;   //CS=1
}
//填充法 画实心矩形        左上角坐标（x0,y0） 矩形长=l 矩形宽=h
void drawrectangle(uint color,uint x0,uint y0,uint l,uint h,uint psw)
{
        writedata(CONTROL0,psw);
        writedata(FRONT,color);
        writedata(CONTROL0,0X40|psw); //control0(6)=1
        writedata(FRONT,color>>8);
        writedata(CONTROL0,0X00|psw); //control0(6)=0
        writedata(YL,y0);
        writedata(YH,y0>>8);
        writedata(XL,x0);
        writedata(XH,x0>>8);
        writedata(RECL,l);
        writedata(RECH,h);
        writedata(CONTROL0,0X40|psw); //control0(6)=1
        writedata(RECL,l>>8);
        writedata(RECH,h>>8);
        writedata(CONTROL0,0X00|psw); //control0(6)=0
        writedata(CONTROL1,0X01);        //fillset=1
        writedata(CONTROL1,0X00);        //fillset=0
        GPIOA_ConfigIN();                //配置输入
        GPIOB->BRR=GPIO_Pin_15;       //CS=0
        GPIOB->BSRR=GPIO_Pin_14;     //WR=1
        GPIOB->BRR=GPIO_Pin_13;       //RD=0
        while((GPIOA->IDR&0x00ff)!=0x00){while((GPIOA->IDR&0x00ff)!=0x00);}   //BUSY 不是连续两次为 0，则重新判别
        GPIOB->BSRR=GPIO_Pin_15;     //CS=1
        GPIOB->BSRR=GPIO_Pin_13;     //RD=1
        GPIOA_Configuration();       //配置输出
        writedata(CONTROL0,psw);
```

B8086.h 文件中是对 STM32F103C8 的库函数 RCC,GPIO 口的设定，详见后文 20 页

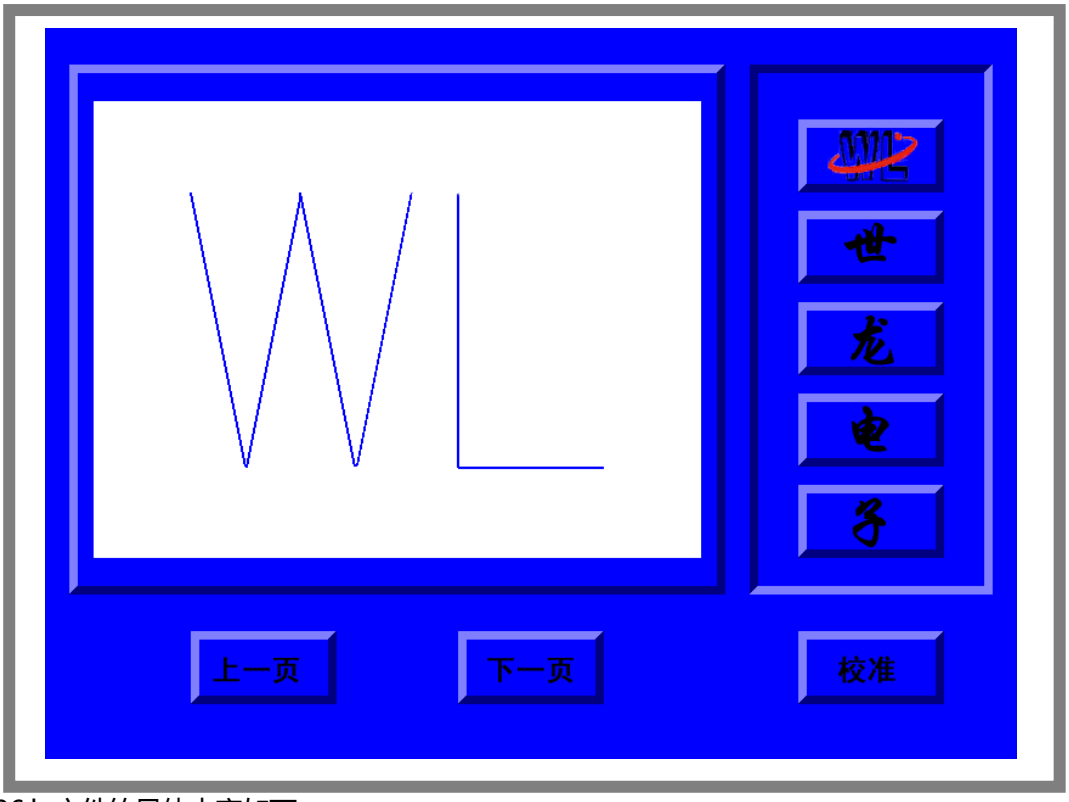
ziku.h 文件中包括：48×48 点阵的汉子“世龙电子”，24×24 点阵的汉子“上下一页校准”和 64×48 点阵的字符“WL”

```
}
// 画实心矩形      左上角坐标 ( x0,y0 )  矩形长=l  矩形宽=h
void writerectangle(uint color,uint x0,uint y0,uint l,uint h,uint psw)
{
        uint i,j,x;
        writedata(CONTROL0,psw);
        for(i=0;i<h;i++)
        {
            writedata(YL,y0);
            writedata(YH,y0>>8);
            y0++;
              x=x0;
              for(j=0;j<l;j++)
              {
                    writedata(XL,x);
                    writedata(XH,x>>8); x++;
                     writedata(CONTROL0,0X40|psw);
                    writedata(DATA,color>>8);
                    writedata(CONTROL0,0X00|psw);
                     writedata(DATA,color);
              }
        }
}
//画点
void writepoint(uint color,uint x,uint y)
{
    writedata(XL,x);
        writedata(XH,x>>8);
        writedata(YL,y);
        writedata(YH,y>>8);
        writedata(CONTROL0,0Xc0);
        writedata(DATA,color>>8);
        writedata(CONTROL0,0X80);
        writedata(DATA,color); //写色值
}
//用画实心矩形的方法画水平直线
void drawline_v(uint color,uint x0,uint y0,uint l)
{
        drawrectangle(color,x0,y0,l,1,0x83);
}
//用画实心矩形的方法画垂直直线
void drawline_h(uint color,uint x0,uint y0,uint l)
{
        drawrectangle(color,x0,y0,1,l,0x83);
}
//写字符  w=字符宽度/8
void writeword(uint fcolor,uint bcolor,uint x,uint y,uint l,uint h,uchar *pdata,uchar psw)
{
    uint i,j,n=0;
        writedata(CONTROL0,psw);
        writedata(FRONT,fcolor);
        writedata(BACK,bcolor);
        writedata(CONTROL0,0X40|psw);
        writedata(FRONT,fcolor>>8);
        writedata(BACK,bcolor>>8);
        writedata(CONTROL0,0X00|psw);
        for(i=0;i<h;i++)
        {
            writedata(YL,y);
              writedata(YH,y>>8);
              writedata(XL,x);
              writedata(XH,x>>8);
              for(j=0;j<l;j++)
              writedata(DATA,pdata[n++]);
              y++;
        }
}
// 画按钮 长=l   宽=h   边宽=w mode：按钮的凹凸 =0 凸，=1 凹
void drawbutton(uchar color,uint x0,uint y0,uint l,uint h,uint w,uchar mode)
{
    uint i,lcolor,dcolor;
        if(mode==0) {lcolor=0x841f;dcolor=0x0010;}       //16 位纯蓝的亮暗边色
        else {lcolor=0x0010;dcolor=0x841f;}
        writerectangle(lcolor,x0,y0,l,w,0x80);
        writerectangle(lcolor,x0,y0,w,h,0x80);
        for(i=1;i<=w;i++)
        {
            writerectangle(dcolor,x0+i,y0+h-i,l-i,1,0x80);
            writerectangle(dcolor,x0+l-i,y0+i,1,h-i,0x80);
        }
}
// 显示鼠标 光标 CONTROL1=0x0c(鼠标) =0x44(十字光标) =0x04(|字光标)
void displaycursou(uint x,uint y,uint psw)
{
```

```
    writedata(CONTROL1,psw);
      writedata(CONTROL0,0Xc0);
      writedata(CONTROL1,48);
      writedata(CONTROL0,0X80);
      writedata(YM,y);writedata(XM,x);
      writedata(CONTROL0,0XC0);
      writedata(YM,y>>8);writedata(XM,x>>8);
      writedata(CONTROL0,0X80);
}
//十字光标演示--光标移动一次画一个，演示效果为在屏用十字光标画一个 'WL'
void draw_WL(uint color)
{
      uint i,x,y,time;
      uchar j,k;
      time=5;
      x=2*V+(25*V-20*H-2*V)/2;y=4*H+5*H;
      for(k=0;k<2;k++)                    //write "W"
      {
            for(i=0;i<3*H;i++)
            {
                  for(j=0;j<5;j++)
                  {
                        displaycursou(x,y,0x44);
                        writepoint(color,x,y);        //写点
                        writepoint(color,x+1,y);
                        y++;delay(time);              //为增加演示效果，延时
                  }x++;
            }
            for(i=0;i<3*H;i++)
            {
                  for(j=0;j<5;j++)
                  {
                        displaycursou(x,y,0x44);
                        writepoint(color,x,y);        //写点
                        writepoint(color,x+1,y);
                        y--;delay(time);              //为增加演示效果，延时
                  }x++;
            }
      }
      x=x+2*V;
      for(i=0;i<15*H;i++)                       //write "L"
      {
            displaycursou(x,y,0x44);
            writepoint(color,x,y);        //写点
            writepoint(color,x+1,y);
            y++;delay(time);              //为增加演示效果，延时
      }
      for(i=0;i<8*H;i++)
      {
            displaycursou(x,y,0x44);
            writepoint(color,x,y);        //写点
            writepoint(color,x,y+1);
            x++;delay(time);          //为增加演示效果，延时
      }
}
// 界面 0
void interface0()
{
    drawrectangle(B,0,0,V*40,H*40,0x83); //全屏清蓝色
      drawrectangle(WHITE,V*2,H*4,V*25,H*25,0x83);   //白色矩形
      drawbutton(B,V*1,H*2,V*27,H*29,H/2,0) ;
      drawbutton(B,V*29,H*2,V*10,H*29,H/2,1) ;

      drawbutton(B,V*6,H*33,V*6,H*4,H/2,0); //按钮 8 点写 "上一页"
      writeword(BLACK,WHITE,V*6+(V*6-75)/2,H*33+(H*4-24)/2,3,24,shang,0x85);
      writeword(BLACK,WHITE,V*6+(V*6-75)/2+25,H*33+(H*4-24)/2,3,24,yi,0x85);
      writeword(BLACK,WHITE,V*6+(V*6-75)/2+50,H*33+(H*4-24)/2,3,24,ye,0x85);

      drawbutton(B,V*17,H*33,V*6,H*4,H/2,0);   //按钮 8 点写 "下一页"
      writeword(BLACK,WHITE,V*17+(V*6-75)/2,H*33+(H*4-24)/2,3,24,xia,0x85);
      writeword(BLACK,WHITE,V*17+(V*6-75)/2+25,H*33+(H*4-24)/2,3,24,yi,0x85);
      writeword(BLACK,WHITE,V*17+(V*6-75)/2+50,H*33+(H*4-24)/2,3,24,ye,0x85);

      drawbutton(B,V*31,H*5,V*6,H*4,H/2,0);   //按钮 多点写 'WL'
      writeword(BLACK,WHITE,V*31+(V*6-64)/2,H*5+(H*4-48)/2,8,48,wl,0x85);
      writeword(R,BLACK,V*31+(V*6-64)/2,H*5+(H*4-48)/2,8,48,huan,0x85);

      drawbutton(B,V*31,H*10,V*6,H*4,H/2,0); //按钮 多点写 '世'
      writeword(BLACK,WHITE,V*31+(V*6-48)/2,H*10+(H*4-48)/2,2,6,48,shi,0x85);

      drawbutton(B,V*31,H*15,V*6,H*4,H/2,0); //按钮 多点写 '龙'
      writeword(BLACK,WHITE,V*31+(V*6-48)/2,H*15+(H*4-48)/2,2,6,48,lon,0x85);

      drawbutton(B,V*31,H*20,V*6,H*4,H/2,0); //按钮 多点写 '电'
```

```
        writeword(BLACK,WHITE,V*31+(V*6-48)/2,H*20+(H*4-48)/2,6,48,dian,0x85);

        drawbutton(B,V*31,H*25,V*6,H*4,H/2,0); //按钮 多点写 '子'
        writeword(BLACK,WHITE,V*31+(V*6-48)/2,H*25+(H*4-48)/2,6,48,zi,0x85);

        drawbutton(B,V*31,H*33,V*6,H*4,H/2,0); //按钮 8 点写 "校准"
        writeword(BLACK,WHITE,V*31+(V*6-50)/2,H*33+(H*4-24)/2,3,24,jiao,0x85);
        writeword(BLACK,WHITE,V*31+(V*6-50)/2+25,H*33+(H*4-24)/2,3,24,zhun,0x85);
}
//主程序
int main()
{
        delay(100);
        initialization(); //初始化
        GPIOB->BSRR=GPIO_Pin_15;    //CS=1
        GPIOB->BSRR=GPIO_Pin_14;    //WR=1
        GPIOB->BSRR=GPIO_Pin_13;    //RD=1

        writedata(BRI,0);   //显示屏调到最亮
        delay(100);
        interface0();
        draw_WL(B);
        while(1);
}
```

演示程序的显示结果如下图：



B8086.h 文件的具体内容如下：

```
#include "stm32f10x_lib.h"
void RCC_Configuration(void)        //时钟配置
{
        RCC_DeInit();               //复位 RCC
        RCC_HSEConfig(RCC_HSE_ON);   //Enable HSE，HSE = 8 MHz
         while (RCC_GetFlagStatus(RCC_FLAG_HSERDY) == RESET);   // Wait till HSE is ready
//      RCC_SYSCLKConfig(RCC_SYSCLKSource_HSE);   //将 HSE 设置为系统时钟源
        RCC_PLLConfig(RCC_PLLSource_HSE_Div1, RCC_PLLMul_4);    //PLLCLK = 8MHz * 4 = 32 MHz
         RCC_PLLCmd(ENABLE);        //启动 PLL
         while (RCC_GetFlagStatus(RCC_FLAG_PLLRDY) == RESET){} //等待 PLL 启动
        RCC_SYSCLKConfig(RCC_SYSCLKSource_PLLCLK);   //将 PLL 设置为系统时钟源
        while (RCC_GetSYSCLKSource() != 0x08){}       //等待系统时钟源的启动
        RCC_HCLKConfig(RCC_SYSCLK_Div1); //配置 HCLK 时钟，HCLK 时钟等于 SYSCLK
        RCC_PCLK2Config(RCC_HCLK_Div1);          //配置高速 APB 时钟，APB2 时钟等于 HCLK
        RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA|RCC_APB2Periph_GPIOB,ENABLE); //使能 GPIOA GPIOB 时钟
}
void GPIOA_Configuration(void)     //GPIOA 配置输出
```

```
{
        GPIO_InitTypeDef GPIO_InitStructure;
        GPIO_InitStructure.GPIO_Pin=GPIO_Pin_7|GPIO_Pin_6|GPIO_Pin_5|GPIO_Pin_4|GPIO_Pin_3|GPIO_Pin_2
                                    |GPIO_Pin_1|GPIO_Pin_0;   //使能选择引脚
        GPIO_InitStructure.GPIO_Speed=GPIO_Speed_50MHz;          //引脚最大输出频率为 50HZ
        GPIO_InitStructure.GPIO_Mode=GPIO_Mode_Out_PP;          //推拉输出
        GPIO_Init(GPIOA,&GPIO_InitStructure);     //GPIOA 初始化
}
void GPIOA_ConfigIN(void)   //GPIOA 输入配置
{
        GPIO_InitTypeDef GPIO_InitStructure;
        GPIO_InitStructure.GPIO_Pin=GPIO_Pin_7|GPIO_Pin_6|GPIO_Pin_5|GPIO_Pin_4|GPIO_Pin_3|GPIO_Pin_2
                                    |GPIO_Pin_1|GPIO_Pin_0;   //使能选择引脚
        GPIO_InitStructure.GPIO_Speed=GPIO_Speed_50MHz;          //引脚最大输出频率为 50HZ
        GPIO_InitStructure.GPIO_Mode=GPIO_Mode_IN_FLOATING;         // 浮空输入
        GPIO_Init(GPIOA,&GPIO_InitStructure);     //GPIOA 初始化
}
void GPIOB_Configuration(void)     //GPIOB 配置输出
{
       GPIO_InitTypeDef GPIO_InitStructure;
       GPIO_InitStructure.GPIO_Pin
       GPIO_Pin_15|GPIO_Pin_14|GPIO_Pin_13|GPIO_Pin_12|GPIO_Pin_2|GPIO_Pin_1|GPIO_Pin_0;
       GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
       GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
       GPIO_Init(GPIOB,&GPIO_InitStructure);
}
void initialization(void)
{
        RCC_Configuration();
        GPIOA_Configuration();
        GPIOB_Configuration();
}
```

附录一、安徽世龙电子技术有限公司服务规范

1、 您购买安徽世龙电子技术有限公司各类控制板和液晶显示模块时，我公司将事先进行检测，确保您所购买的控制板和模块为完好的产品，液晶模块符合生产厂家提供的检测标准。

2、 液晶模块属于元器件类产品，不属于设备，不能享受保修服务。

3、 如果在使用过程中，您不小心损坏了液晶模块，我们将为您提供维修服务；

（1） 由于产品质量问题造成液晶模块显示不正常的，我公司将提供免费维修，必要时可以更换模块；

（2） 由于客户原因使模块受损的，我公司将尽力维修，如果我公司不能维修的，将返回生产厂家进行维修，这类情况将收取相应的维修成本费用。

4、 如果由于液晶屏的物理损伤造成液晶模块不正常工作的，一般该模块只能报废。

5、 在我公司购买的液晶产品出现需要翻修的情况时，请认真填写《返修单》，如果没有《返修单》的请尽量就使用情况和故障现象详细描述，并和故障产品一并返回到我公司。

附录二、液晶模块装配与使用注意事项、运输、产品责任等

一、 **处理保护膜**

在装好的模块屏表面有一保护膜，以防在装配时玷污显示表面，所以在整机装配结束前不得揭去，以免弄赃或损坏显示面。

二、 **加装垫**

在模块与前面板之间最好加一块 0.1 毫米左右的垫，面板还应保持平整，以免在装配后产生扭曲。

三、 **严防静电**

模块中的控制、驱动电路是低压、微功耗的 CMOS 电路，极易被静电击穿，是一种不可

修复的损坏，务必注意，不可大意。所以，在操作、装配以及使用中都应极其小心，要严防静电。为此：

1、 不要用手随意去摸外引线、电路和 IC 等；

2、 焊接使用的烙铁必须良好接地，没有漏电。

四、 装配操作时的注意事项

1、 液晶模块是精心设计组装而成的，请勿随意自行加工、维修；

2、 金属框架不得随意扭动、拆卸；不要随意修改、加工 PCB 板外形、装配孔、线路及部件；不要随意更改导电胶条，尤其注意有些模块侧面的柔性电缆需要保护，不能损伤；不要修改任何内部支架；不要碰、摔、折曲、扭动模块和背景部分等；

五、 焊接、在焊接模块外引线、接口电路时，应按如下规定进行：

1、 如果液晶模块与其他外围电路连接需要焊接或改变原有的连接头的话，确认通过质检；

2、 烙铁温度：280±10℃；焊接时间：＜3～4S；焊接材料：共晶型、低熔点：重复焊不得超过 3 次。

六、模块的作用

1、液晶模块的外引线决不允许接错，不允许与 PCB 上不相关的焊盘、过孔等短路，否则可能造成过流、过压等液晶模块元器件有损的现象；

2、模块使用接入电源及断开电源时，必须在正电源（5±0.25V）稳定以后接入，才能输入电平信号。如在电源稳定前或断开后输入信号电平，有可能损坏模块中的 IC 电路等。

3、点阵液晶模块显示时的对比度、视角与温度、驱动电压关系很大，所以，如果 V_{EE} 调整过高，不仅影响显示，还会缩短模块的使用寿命。

4、模块在规定工作温度范围以下使用时，显示响应很慢，而在规定工作温度范围以上使用时，整个显示面又会呈现全显示状态，这种现象不是模块被损坏，只需恢复到规定温度范围内，一切又将恢复正常。（不应在超过存储极限温度的范围外使用或存储，如果温度低于结晶温度，液晶就会结晶，破坏定向层，使器件报废；如果温度过高，液晶将会变成各向同性的液晶，失去液晶态，也就失去了液晶器件的功能。）

5、用力按压显示部位，会产生异常显示。可切断电源，稍待片刻，重新上电，即恢复正常。

七、模块的存储

若长期（如几年）存储，我们推荐以下方式：

装入聚乙烯口袋（最好有防静电涂层）并将口密封；放置暗处，避强光；决不能在表面压放任何物品；严格避免在极限温度、湿度条件以外存放（液晶用的偏振片怕高温、怕潮湿）。

[运输损坏]

如果用户收到的货物在运输过程中已经损坏，若是包装受损的话，用户首先应该在得到送货人允许的前提下打开包装，如果货物受损，用户应该向运输公司索赔；否则一定要原封不动地保留货箱、包装材料及货物，并与安徽世龙电子技术有限公司联系。

[产品责任]

公司保证所有售出的产品符合生产厂家的质量要求，并对其承担质量保证的责任，若用户在购买产品的 30 天内发现产品的质量确有问题，经安徽世龙电子技术有限公司或液晶生产厂家检测，系产品本身的质量问题，安徽世龙电子技术有限公司将负责维修或换货或退货，安徽世龙电子技术有限公司承担的产品责任不超过用户购买货品价值，并不对用户使用产品所造成的间接损失负责。由于用户对产品使用不当而导致产品的损坏（例如静电，焊接、连线不当，过流、过压使用等）、安徽世龙电子技术有限公司将不承担任何责任，但可尽力为用户提供维修服务，并将根据具体情况收取适当费用。