

# VLCM32040 使用说明

## 一. 外形尺寸及接口定义：

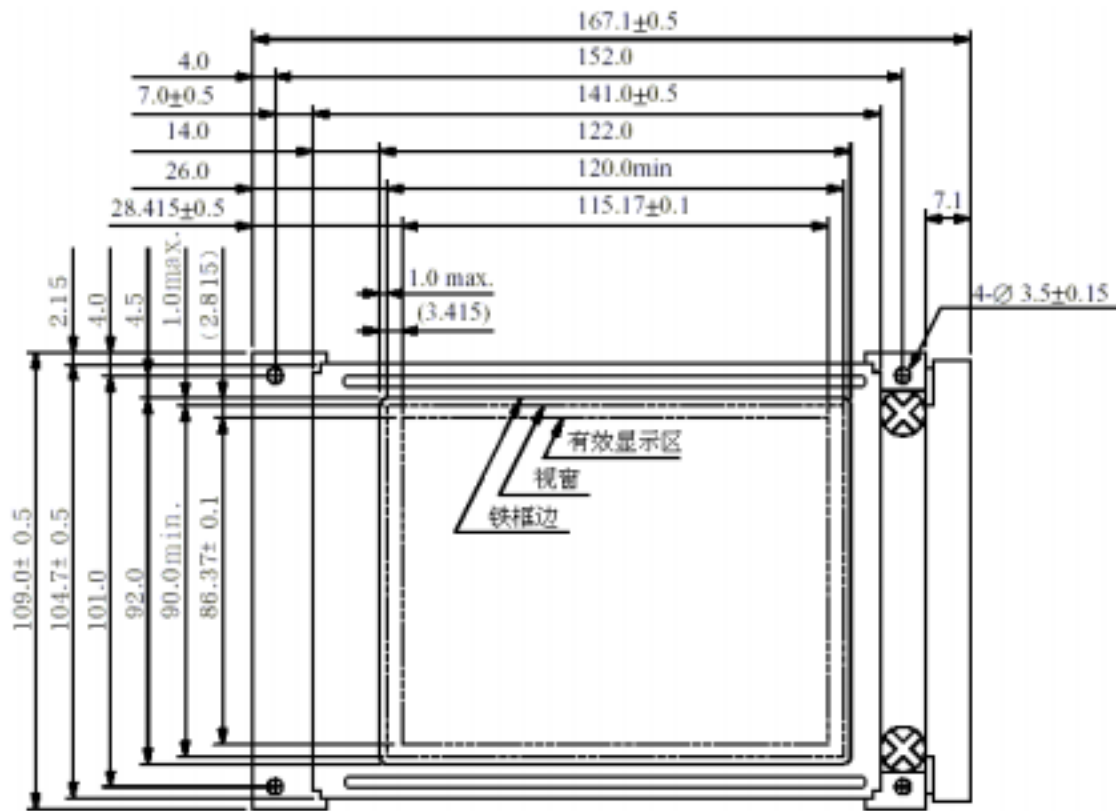
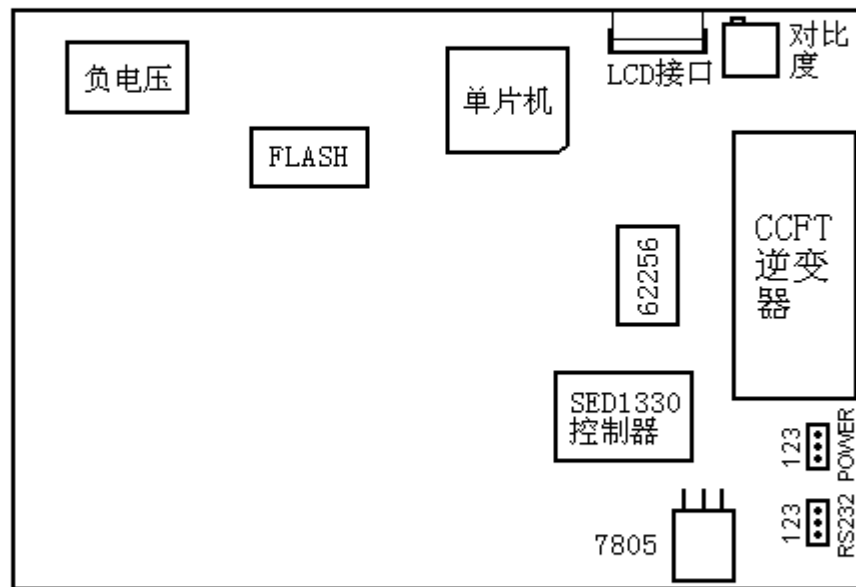
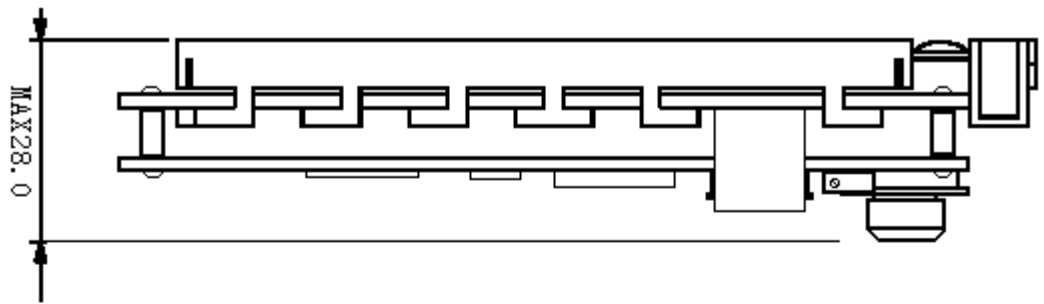


图 1.液晶外形尺寸



- POWER电源接口: 1. +9VDC (经7805稳压后供单片机)  
 2. GND (电源地) 0V  
 3. +9VDC/500mA (供逆变器使用)
- RS232通信接口: 1. BUSY忙标志信号输出, TTL电平, 低为忙。  
 2. RX控制命令输入, TTL电平, 19200BP, 8位数据, 1个起始位, 1个停止位。  
 3. GND信号地

图 2. VLCM320240 侧背面图示

二 . VLCM320240 与电脑的连接 :

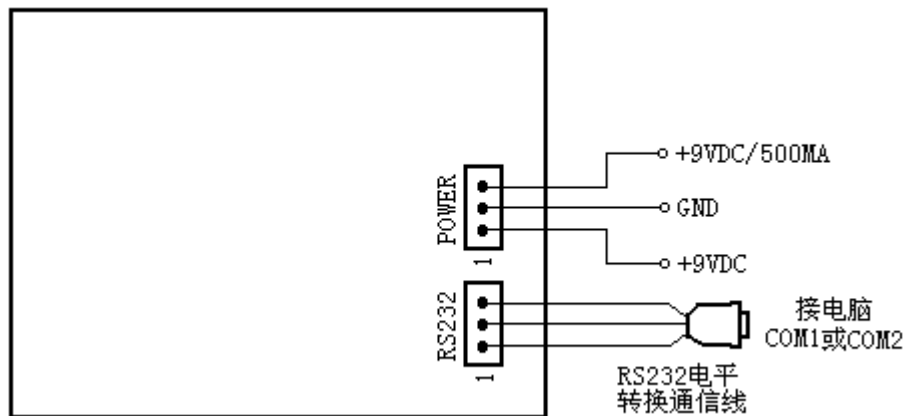


图 3. 控制板与电脑的连接

按上图将 LCD 模块，控制板及电脑连好，背光电源应单独供给，打开电源此时在 LCD 显示屏上看到已存的图案，并可调节对比度电位器将对对比度调到满意，此时就进行显示界面的编辑和模拟。



图 4 Visual LCM1.3 仿真软件主画面

### 三． 显示界面的模拟生成：

首先将各显示界面中用到的图案进行归纳，并在电脑上充分利用已有资源完成各种图样的编辑，资料下载及仿真调试，所配仿真调试软件 Visual LCM1.3 可方便地进行上述工作，将所配仿真调试软件 Visual LCM1.3 安装在电脑上并运行之（注意在运行前应先将其已占用电脑串口资源的应用程序关闭）。

先用 Visual LCM1.3 中的“图文编辑”将 LCD 预显示位图画好存档，所做图样必须是黑白两色（图象->属性）并垂直翻转（图象->翻转）的 BMP 文档资料，且大小不超过 320X240 像素，在 BMP 文档列表中选定某位图后通过“添加”一一列入“下载位图排序”列表中，将模块与电脑由 RS232 电平转换线相接并通电，在“串口选择”中选好模块所接串口号，用“连线测试”检查连线和串口号及电源是否正常，若联机成功，点击“下载资料”后电脑将排序列表中的位图按序号由小到大一一下载到模块内 FLASH 中，下载时间与列表中位图总和大小及电脑速度有关，结束时通过点击“浏览资料”可在 LCD 屏上以点（00，00）为顶点一一观察刚刚下载的所有位图，且顺序与排序列表中的相同，也可直接点击排序列表中的位图进行浏览，所有位图无误后，可进行 LCD 显示界面的仿真调试。

LCD 显示界面的生成用八种命令来实现，在主画面的“发送命令”下边的命令列表框中罗列出这八种命令的含义及语法规则，若输入错误的命令将会提示错误信息，命令及各参数说明如下：

表 1 命令功能说明 (下划线部分为命令参数)

命令格式	功能说明
BMP <u>nxy</u>	在以点 (x, y) 为顶点处显示位图 n
CLR	清除屏上所有显示 (无参数命令)
CLS <u>xywh</u>	清除以点 (x, y) 为顶点, 宽 w 高 h 的矩形
INV <u>xywh</u>	反色以点 (x, y) 为顶点, 宽 w 高 h 的矩形
SDT <u>xy</u>	在点 (x, y) 处画点
HZA <u>xy</u>	设置点 (x, y) 为汉字或 ASCLL 字符显示起点
HZD <u>汉字显示 123</u>	显示汉字"汉字显示 123", 一次输入最多 127 个
ASD <u>\$123ABCabc</u>	显示 ASCLL 字符串, 一次输入最多 255 个

表 2 命令参数说明

参数	取值范围	含义	进制	举例
N	0=<n<=序号最大值	位图序号	十六进制	0B
X	0=<x<=013FH	显示操作区域顶点横坐标 (两字节)	十六进制	0012
Y	0=<y<=EFH	显示操作区域顶点纵坐标 (一字节)	十六进制	09
W	0<w<=FFH	显示操作区域横向宽度点数 (一字节)	十六进制	0F
H	0<h<=FFH	显示操作区域纵向高度点数 (一字节)	十六进制	0E
"汉"	GB_2312 标准字库	欲显示的汉字串, 一次最多 127 个	汉字内码	BABA
"\$12"	标准 ASCII 字符	欲显示的 ASCII 字符串, 一次最多 255 个	ASCII 码	34H

注: 1.若位图不为 320X240 的整屏, 宽度应小于 256, 否则不能显示。

2.HZD 和 ASD 命令的显示起始地址都用 HZAxy 设置, 并所设地址在下一 HZAxy 之前不改变。

各命令位置参数在分辨率 320X240LCD 显示屏上图示如下:

汉字或 ASCII 字符显示命令中的参数即为所显每个汉字的两个字节 GB 码或 ASCII 字符的 ASCII 码, 如"汉"的 GB 码为 BABA, "A"的 ASCII 码为 41H, 汉字的 GB 码或 ASCII 字符的 ASCII 码可经编译软件 MBUG 自动生成, 无需关心其值究竟为多少, 编辑修改非常方便。

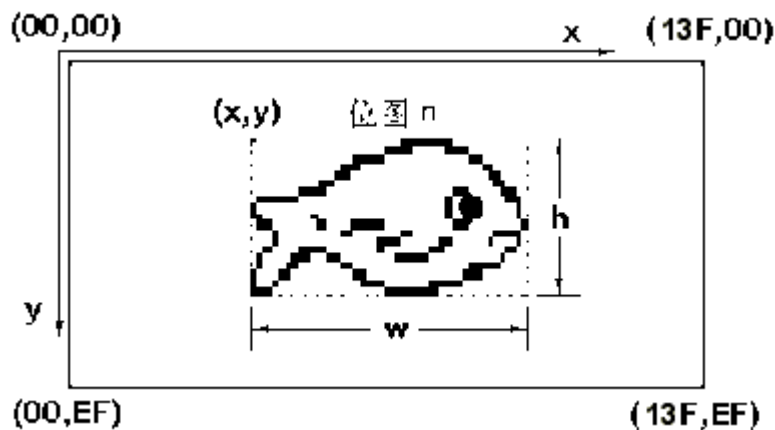


图 5 位置参数在 320X240 模块显示屏上示意

在主画面的“发送命令”左边的命令输入框中输入正确的命令后点击“发送命令”, 此时 LCD 屏上会有相应的显示动作, 若在调试过程中发现要修改某位图时可以

用“图文编辑”先将该位图垂直翻转后重新修改后按前述步骤再一次下载修改后的资料，直到对显示界面满意为止，到此 LCD 显示界面就以基本完成，剩下的工作就是将所用到的命令通过用户控制器在相应时刻或状态串行发送给 LCD 模块即可。

#### 四．脱离电脑与单片机实时运行：

最终要将模块脱离电脑通过模块上的 SIO 接入用户控制电路中运行，

现举例说明 VLCM320240 模块如何与单片机 AT89C2051 接口，为保证波特率的准确，建议 AT89C2051 用 11.0592MHz 的晶体，连接图示如下：

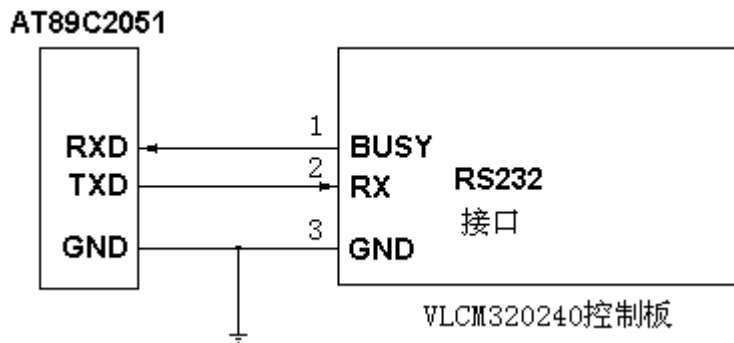


图 5 单片机 AT89C2051 与 VLCM320240 控制板的连接

在编写单片机 AT89C2051 的显示控制软件时需注意以下事项：

- 1．串行口模式设为模式 1（1 个起始位，8 个数据位，1 个停止位）；
- 2．波特率设为 19200（TH1=TL1=0FDH，SMOD=1）；
- 3．在发送命令之前需检测模块为闲态（PIN1=1），否则所发的命令可能失效；
- 4．串口发送的必须是完整命令的 ASCII 码（汉字为其机内码，ASCII 字符为其 ASCII 码），如命令“BMP0B000C0D”，实际由串口发送内容是“424D503042303030433044”，顺序为“B->M->P->0->B->0->0->0->C->0->D”；
- 5．在每条命令的首末还要加上命令结束码“回车键”的 ASCII 码“0D”，以表明该命令的开始结束。
- 6．若单片机 AT89C2051 的串口已被占为他用，那么用一根口线来模拟生成一个通讯协议为(19200,n,8,1)的软串口发送命令即可。

用 MCS-51 汇编语言编写简单的显示控制源程序如下（实际运行已通过，也可直接移植到其他 51 系列单片机上，实际应用请注意现场保护）：

```

BUSY BIT P3.0 ;定义忙信号口线，
                ;若 BUSY BIT P3.1 当然也可以，因为发送命令与忙信号
                ;不可能同时出现，还可减少一根口线

X_V EQU 30H ;定义各参数单元
XH_V EQU 31H
Y_V EQU 32H
NUBER EQU 33H
WIDTH EQU 34H
HIGHT EQU 35H

REG_ASC_XH EQU 36H ;定义参数 x 高四位 ASCII 码单元
REG_ASC_XL EQU 37H ;定义参数 x 低四位 ASCII 码单元
REG_ASC_XHH EQU 38H ;定义参数 xH 高四位 ASCII 码单元
    
```

```

REG_ASC_XHL EQU 39H ;定义参数 xH 低四位 ASCII 码单元
REG_ASC_YH EQU 3AH ;定义参数 y 高四位 ASCII 码单元
REG_ASC_YL EQU 3BH ;定义参数 y 低四位 ASCII 码单元
REG_ASC_NH EQU 3CH ;定义参数 n 高四位 ASCII 码单元
REG_ASC_NL EQU 3DH ;定义参数 n 低四位 ASCII 码单元
REG_ASC_WH EQU 3EH ;定义参数 w 高四位 ASCII 码单元
REG_ASC_WL EQU 3FH ;定义参数 w 低四位 ASCII 码单元
REG_ASC_HH EQU 40H ;定义参数 h 高四位 ASCII 码单元
REG_ASC_HL EQU 41H ;定义参数 h 低四位 ASCII 码单元
ORG 00H ;程序开始
AJMP MAIN
ORG 30H
MAIN: MOV SP,#60H ;设堆栈指针
LCALL INIT_SIO ;初始化串口
LCALL DELAY
MOV NUBER,#00H
BMP_GO: MOV X_V,#00H ;在 x=0,y=0 为顶点处以最快速度显示所有已下载图片
MOV XH_V,#00H ;若某图号未下载则不显示
MOV Y_V,#00H
LCALL ORDER_BMP ;发送位图显示命令子程 ,其它命令编写类似
INC NUBER
MOV A,NUBER
CJNE A,#6,BMP_GO ;常数 6 为预显示位图个数
MOV NUBER,#00H
LCALL DELAY ;延时观察 LCD 显示屏
LCALL ORDER_CLR ;清 LCD 屏
LCALL DISPLAY_ASCLL;浏览 512 个 ASCLL 字符
LCALL DELAY
LCALL DELAY
LCALL ORDER_CLR
MOV X_V,#52 ;设置汉字显示首地址 XH=0,X=7,Y=13
MOV XH_V,#0
MOV Y_V,#34
LCALL ORDER_HZA
MOV R7,#8 ;欲显汉字个数
MOV DPTR,#TAB_HZ ;汉字内码存放表头
LCALL ORDER_HZ ;显示汉字串子程
LCALL DELAY ;延时观察 LCD 显示屏
AJMP MAIN ;观察 LCD 显示屏
INIT_SIO: MOV SCON,#50H ;串行口:方式 1
MOV TMOD,#21H ;定时器 T1 为波特率发生器 , 方式 2

```

```

MOV TH1,#0FDH ;波特率=19200, Fosc=11.0592MHz
MOV TL1,#0FDH
ORL PCON,#80H ;SMOD=1
SETB TR1 ;启动 T1 波特率发生器
RET

ORDER_HZA: ;设置汉字或 ASCLL 字符显示首地址命令
MOV A,X_V
MOV R0,#REG_ASC_XH
LCALL H_ASC ;将参数 x 转换为两个字节的 ASCII 码
MOV A,XH_V
MOV R0,#REG_ASC_XHH
LCALL H_ASC ;将参数 xH 转换为两个字节的 ASCII 码
MOV A,Y_V
MOV R0,#REG_ASC_YH
LCALL H_ASC ;将参数 y 转换为两个字节的 ASCII 码
LCALL BUSY_CK
MOV A,#"H" ;发送命令码 HZA 的 ASCII 码
LCALL SEND_BYTE
MOV A,#"Z"
LCALL SEND_BYTE
MOV A,#"A"
LCALL SEND_BYTE
MOV R0,#REG_ASC_XHH
LCALL SEND_CS ;发参数 xH 的 ASCII 码
MOV R0,#REG_ASC_XH
LCALL SEND_CS ;发参数 x 的 ASCII 码
MOV R0,#REG_ASC_YH
LCALL SEND_CS ;发参数 y 的 ASCII 码
LJMP ORDER_END

ORDER_HZ:
LCALL BUSY_CK ;检测模块闲状态
MOV A,#"H" ;先发送命令码 HZD 的 ASCII 码
LCALL SEND_BYTE
MOV A,#"Z"
LCALL SEND_BYTE
MOV A,#"D"
LCALL SEND_BYTE

NEXT_HZ:
CLR A ;取汉字的高八位内码
MOVC A,@A+DPTR
INC DPTR ;调整内码指针
LCALL SEND_BYTE
CLR A ;取汉字的低八位内码
MOVC A,@A+DPTR
INC DPTR ;调整内码指针

```

```
LCALL SEND_BYTE
DJNZ R7,NEXT_HZ ;共显示 7 个汉字
MOV A,#0DH ;命令结束
LCALL SEND_BYTE
RET
```

DISPLAY\_ASCLL:

```
LCALL BUSY_CK ;检测模块闲状态
MOV A, #"A" ;先发送命令码 ASD 的 ASCII 码
LCALL SEND_BYTE
MOV A, #"S"
LCALL SEND_BYTE
MOV A, #"D"
LCALL SEND_BYTE
MOV A,#0
MOV R4,#12
```

N\_ASC:

```
MOV SBUF,A
JNB TI,$
CLR TI
INC A
DJNZ R4,N_ASC
MOV SBUF,#20H
JNB TI,$
CLR TI
MOV A,#0EH
MOV R4,#242
```

N\_ASC1:

```
MOV SBUF,A
JNB TI,$
CLR TI
INC A
DJNZ R4,N_ASC1
MOV SBUF,#0DH
JNB TI,$
CLR TI
LCALL BUSY_CK ;检测模块闲状态
MOV A, #"A" ;先发送命令码 ASD 的 ASCII 码
LCALL SEND_BYTE
MOV A, #"S"
LCALL SEND_BYTE
MOV A, #"D"
LCALL SEND_BYTE
MOV A,#0
MOV R4,#12
```



```

N_ASC2:
    MOV SBUF,A
    JNB TI,$
    CLR TI
    INC A
    DJNZ R4,N_ASC2
    MOV SBUF,#20H
    JNB TI,$
    CLR TI
    MOV A,#0EH
    MOV R4,#242

N_ASC3:
    MOV SBUF,A
    JNB TI,$
    CLR TI
    INC A
    DJNZ R4,N_ASC3
    MOV SBUF,#0DH
    JNB TI,$
    CLR TI
    LCALL BUSY_CK ;检测模块闲状态
    MOV A,"A" ;先发送命令码 ASD 的 ASCII 码
    LCALL SEND_BYTE
    MOV A,"S"
    LCALL SEND_BYTE
    MOV A,"D"
    LCALL SEND_BYTE
    MOV A,#0
    MOV R4,#12

N_ASC4:
    MOV SBUF,A
    JNB TI,$
    CLR TI
    INC A
    DJNZ R4,N_ASC4
    MOV SBUF,#20H
    JNB TI,$
    CLR TI
    MOV A,#0EH
    MOV R4,#77

N_ASC5:
    MOV SBUF,A
    JNB TI,$
    CLR TI

```

```
INC A
DJNZ R4,N_ASC5
LJMP ORDER_END
```

ORDER\_CLR:

```
LCALL BUSY_CHK ;检测模块闲状态
MOV A, #"C" ;先发送命令码 HZD 的 ASCII 码
LCALL SEND_BYTE
MOV A, #"L"
LCALL SEND_BYTE
MOV A, #"R"
LCALL SEND_BYTE
LJMP ORDER_END
```

ORDER\_BMP:

```
MOV A,X_V
MOV R0, #REG_ASC_XH
LCALL H_ASC ;将参数 x 转换为两个字节的 ASCII 码
MOV A,XH_V
MOV R0, #REG_ASC_XHH
LCALL H_ASC ;将参数 xH 转换为两个字节的 ASCII 码
MOV A,Y_V
MOV R0, #REG_ASC_YH
LCALL H_ASC ;将参数 y 转换为两个字节的 ASCII 码
MOV A,NUBER
MOV R0, #REG_ASC_NH
LCALL H_ASC ;将参数 n 转换为两个字节的 ASCII 码
LCALL BUSY_CHK ;检测模块闲状态
MOV A, #"B" ;先发送命令码 BMP 的 ASCII 码
LCALL SEND_BYTE
MOV A, #"M"
LCALL SEND_BYTE
MOV A, #"P"
LCALL SEND_BYTE
MOV R0, #REG_ASC_NH
LCALL SEND_CS ;发参数 n 的 ASCII 码
MOV R0, #REG_ASC_XHH
LCALL SEND_CS ;发参数 xH 的 ASCII 码
MOV R0, #REG_ASC_XH
LCALL SEND_CS ;发参数 x 的 ASCII 码
MOV R0, #REG_ASC_YH
LCALL SEND_CS ;发参数 y 的 ASCII 码
```

ORDER\_END:

```
MOV A,#0DH ;发送命令结束标志"0DH"
MOV SBUF,A
```

```

        JNB TI,$
        CLR TI
        RET
BUSY_CHK:
        MOV  A,#0DH           ;结束上一次命令
        MOV  SBUF,A
        JNB TI,$
        CLR TI
        JNB  BUSY,$          ;等待模块为闲状态
        RET
SEND_CS:  MOV  A,@R0           ; 发送参数高四位的 ASCII 码
        LCALL SEND_BYTE
        INC  R0
        MOV  A,@R0           ; 发送参数低四位的 ASCII 码
        LCALL SEND_BYTE
        RET
SEND_BYTE: MOV  SBUF,A        ;发送一个字节
        JNB  TI,$
        CLR  TI
        RET
H_ASC:
        MOV  B, A            ;将一字节的十六进制数转换为两字节的
ASCII 码
        ANL  A, #0F0H
        SWAP A
        CJNE A, #9, PD_G
XY9:    ADD  A, #30H
TURN_LOW:  MOV  @R0, A
        INC  R0
        AJMP LOW_BIT
PD_G:    JC   XY9
        ADD  A, #37H
        AJMP TURN_LOW
LOW_BIT:  MOV  A, B
        ANL  A, #0FH
        CJNE A, #9, PD_G1
XY_9:    ADD  A, #30H
TURN_LOW1:  MOV  @R0, A
        RET
PD_G1:   JC   XY_9
        ADD  A, #37H
        AJMP TURN_LOW1
DELAY:   MOV  R7,#12         ;延时

```

```

DL3:      MOV    R6,#200
DL2:      MOV    R5,200
DL1:      NOP
          NOP
          NOP
          DJNZ   R5,DL1
          DJNZ   R6,DL2
          DJNZ   R7,DL3
          RET
          ORG 400h
TAB_HZ:   DB "深圳华强电子世界"      ;汉字内码表

          END

```

附录：

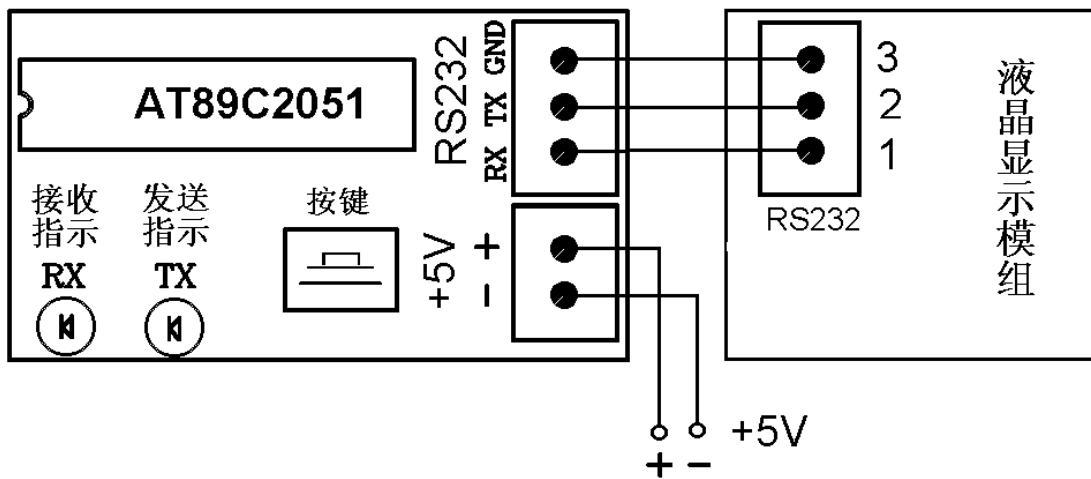
ASCLLZ 字符表 (8X16 点阵)

	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	{		}	~	△

汉字库请参见 GB2312，可在 WINDOWS 附件中的造字程序中浏览。



实物照片



AT89C2051 控制演示板与 VLCM 的连接