(ISO9001:2000)

图形点阵液晶显示模组

型　　号：HGO1601601V1-F-LWH-LSV

版　　本：B

网　　址：http://www.tsingtek.com

邮　　箱：sales@tsingtek.com

清达光电技术有限公司
杭州佳显科技有限公司

版本.	日期	内容	备注
A	2006.09.20	第一版	
B	2007.10.08	符合 RoHS 规范	

产品编码规则

HG 320240 C – B - LW H- NV- L4- TPSD- U- T

(1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11)

(1):产品序列号
- HC→字符
- HG→SMT/COB 图形单色
- HGT→TAB 图形单色
- HGO→COG 图形单色
- HGR→ COLOR STN
- HGF→TFT
- HGS→OLED

(2):规格
- 字符→字符数/每行*行数
- 图形→点/每行*点/每列

(3):产品序列号

(4):显示模式

LCD 模块：
- 省略→STN 黄绿模式
- G→STN 灰模式
- B→STN 蓝模式
- F→FSTN 半透半反
- T→FSTN 透射

OLED 模块：
- Y→黄字
- G→绿字
- B→蓝字
- W→白字

(5):背光类型
- 省略→无背光
- LY→LED 黄绿底光
- SY→LED 黄绿侧光
- LW→LED 白光
- SW→LED 超亮白光
- LB→LED 蓝光
- LR→LED 红光
- LA→LED 琥珀光
- LG→LED 绿光
- EB→EL 蓝光
- EG→EL 绿光
- EW→EL 白光
- CW→CCFL 白光

(6):温度范围
- 省略→常温
- H→宽温
- EH→特宽温

(7):电源
- 省略→5V 单电源
- NV→5V 双电源
- SV→5V 带温度补偿
- LV→3.0/3.3V 单电源
- LNV→3.0/3.3V 双电源
- LSV→3.0/3.3V 带温度补偿
- OV→单电源，V0 不接

(8):背光输入电压说明，请参照液晶详细资料

(9):特殊编号
- 省略→无触摸屏/并行/全屏
- TP→带触摸屏
- S→串行通信
- D→分屏

(10):视角
- 省略→6:00
- U→12:00
- L→9:00
- R→3:00

(11):内部编号
- -01→控制和驱动芯片为 AVANT 产品
- -02→控制和驱动芯片为中微产品

目录

1. 简介

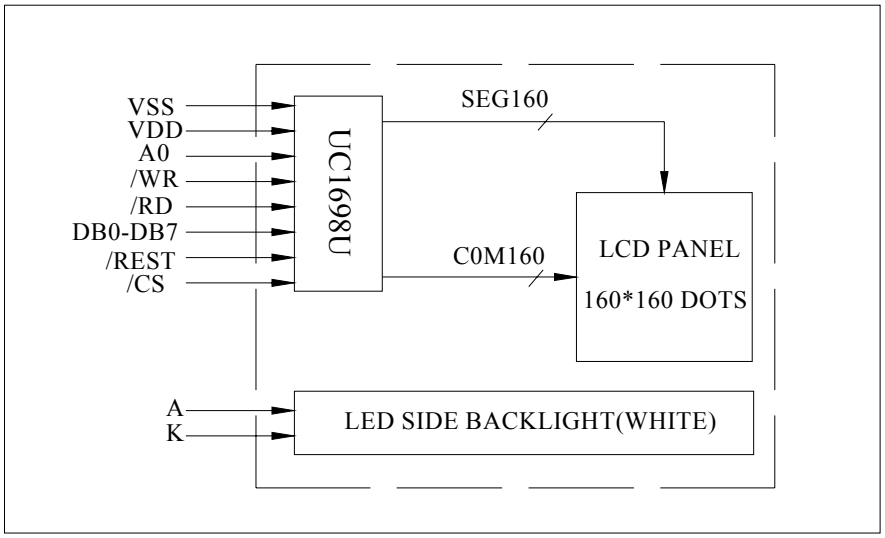
1.1 模块规格

项目	规格
显示类型	FSTN/正性/半透
颜色	显示像素: 黑色
	显示背景: 白色
数据输入格式	8 位并行数据
占空比	1/160 占空比
视角	6 点
驱动 IC	UC1698U
外壳	0.5T
背光	LED 白光
工作温度	-20 °C ~70 °C
存储温度	-30 °C ~ 80 °C
其他	

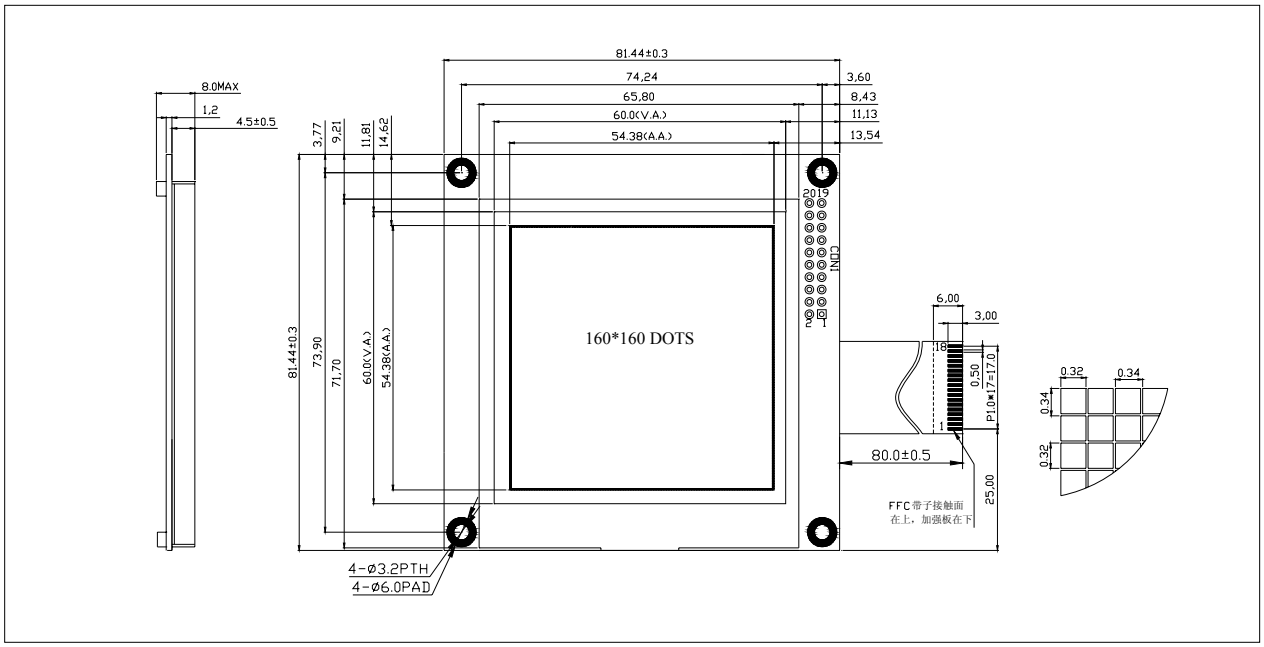
1.2 尺寸规格

项目	规格	单位	备注
外形尺寸	81.44(W)×81.44(H)×8.0MAX.(T)	mm	
可视区	60.0(W)×60.0(H)	mm	
有效区	54.38(W)×54.38(H)	mm	
点阵	160Dots×160Dots	---	
点距离	0.34(W)×0.34(H)	mm	
点大小	0.32(W)×0.32(H)	mm	

1.3 原理结构图



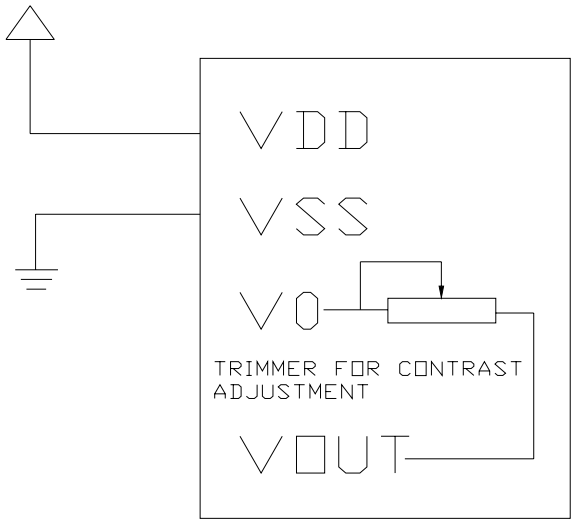
1.4 模块外形图



1.5 接口定义

编号	符号	电平	功能
1	Vss	0V	接地
2	A0	H/L	H: 数据 L: 指令
3	/WR	L	写信号
4	/RD	L	读信号
5	/CS	L	使能信号
6	/REST	L	复位信号，低有效
7	VDD	+3.3V	逻辑电压
8~15	DB0-DB7	H/L	数据线
16	LEDK	0V	LED 背光源-
17	NC	-	空脚
18	LEDA	+3.3V	LED 背光源+

1.6 对比度调整电路



2. 极限参数

（Ta=25 °C, VSS=0V）

参量	符号	范围	单位
电源电压	V_{DD}-V_{SS}	-0.3 ~ 5.5	V
LCD 驱动电压	V_{DD}～V_0	-0.3 ~ 19.8	V
输入电压	V_{IN}	V_{SS} ~ V_{DD}	V
工作温度	Topr	-20 ~70	°C
存储温度	Tstg	-30 ~ 80	°C

3. 电气特性

3.1 电特性

（Ta=25 °C ,VSS=0V）

项目	符号	条件	最小值	典型值	最大值	单位	备注
逻辑工作电压	V_{DD} –V_{SS}	--	2.8	3.3	3.8	V	
输入高电平	V_{IH}	--	0.8 V_{DD}	--	V_{DD}	V	
输入低电平	V_{IL}	--	GND	--	0.2 V_{DD}	V	
逻辑工作电流	I_{DD}	V_{DD} –V_{SS} =3.3V	--	1.2	2.4	mA	
LCD 驱动电压	V_{LCD} Φ=0 θ=0	Ta=25 °C	--	16.5	--	V	

3.2 LED 背光规格

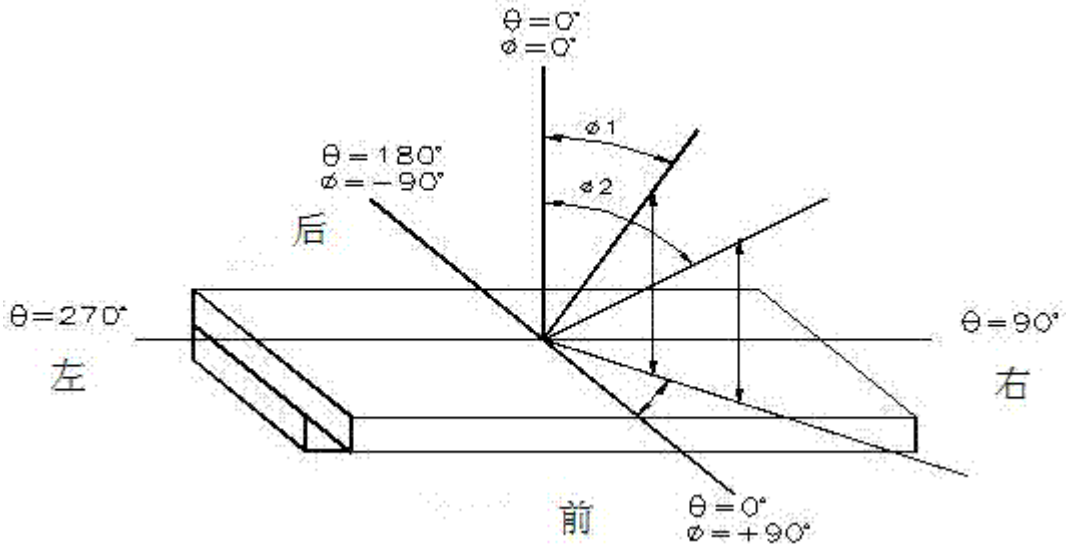
项目	符号	最小值	典型值	最大值	单位	条件
工作电压	Vf	2.9	3.1	3.3	V	If=60 mA
颜色	白色					

3.3 光电特性 （Ta=25 °C ，V_{DD}=3.3V）

项目	符号	条件	最小值	典型值	最大值	单位	备注
视角	ϕ2-ϕ1	K≥3	--	50	--	DEG	1、2
对比度	K	ϕ =0°,θ=0°	3	5	--	--	3
帧频率				70		Hz	
响应时间	Tr(rise)	ϕ =0°,θ=0°	--	250	350	ms	
	Tf(fall)	ϕ =0°,θ=0°	--	250	350	ms	4

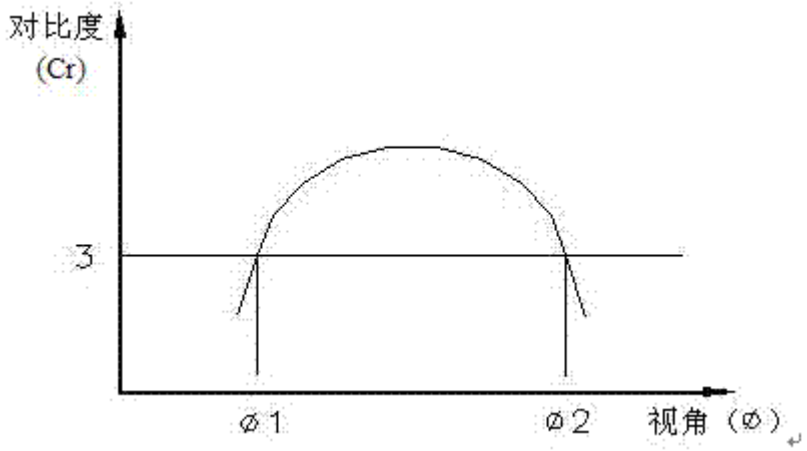
备注：

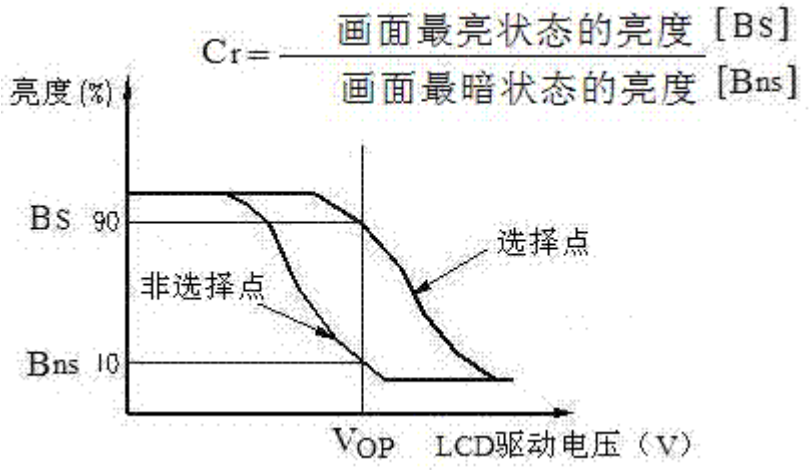
1: 视角 θ, ϕ 的定义：



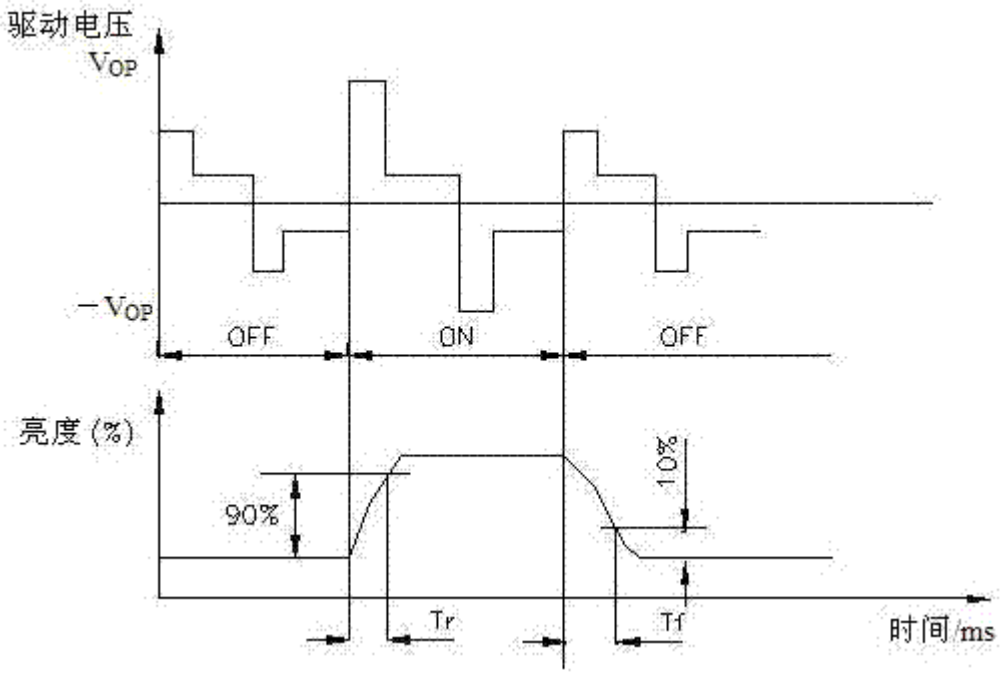
2: 视角范围的定义 Δϕ=|ϕ 2-ϕ 1|

3: 对比度的定义



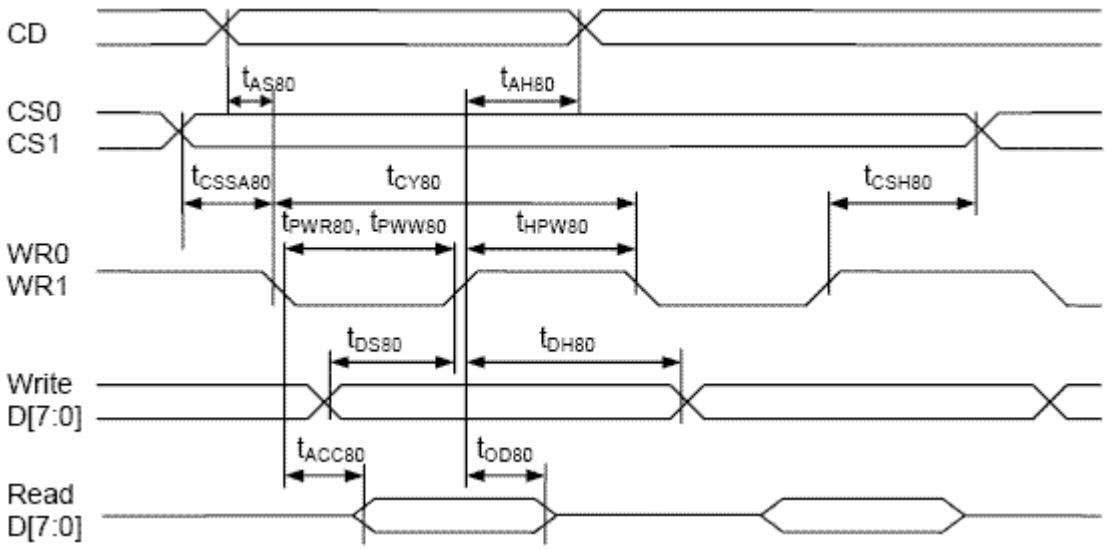

Cr= 画面最亮状态的亮度 [Bs] / 画面最暗状态的亮度 [Bns]
亮度(%)
Bs 90
Bns 10
非选择点
选择点
VOP LCD驱动电压（V）

4: 响应时间的定义


驱动电压
VOP
-VOP
OFF
ON
OFF
亮度(%)
90%
10%
Tr
Tf
时间/ms

4.时序说明

并行 8080 时序:



Symbol	Signal	Description	Condition	Min.	Max.	Units
t_{AS80} t_{AH80}	CD	Address setup time Address hold time		0 0	–	nS
t_{CY80}		System cycle time 16-bit bus (read) (write) 8-bit bus (read) (write)	 LC[7:6]=10b LC[7:6]=01b	 170 130 100 80 90	–	nS
t_{PWR80}	WR1	Pulse width 16-bit (read) 8-bit		85 50	–	nS
t_{PWW80}	WR0	Pulse width 16-bit (write) 8-bit	 LC[7:6]=10b LC[7:6]=01b	65 40 45	–	nS
t_{HPW80}	WR0, WR1	High pulse width 16-bit bus (read) (write) 8-bit bus (read) (write)	 LC[7:6]=10b LC[7:6]=01b	 85 65 50 40 45	–	nS
t_{DS80} t_{DH80}	D0~D15	Data setup time Data hold time		30 0	–	nS
t_{ACC80} t_{OD80}		Read access time Output disable time	C_L = 100pF	– 15	60 30	nS
T_{CSSA80} t_{CSH80}	CS1/CS0	Chip select setup time		5 5		nS

5. 控制器指令及功能介绍

COMMAND TABLE

The following is a list of host commands supported by UC1698u

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

	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 & PM	0	1	GE	MX	MY	WA	DE	WS	MD	MS	Get {Status, Ver, PMO, Product Code, PID, MID}	N/A
				Ver	PMO[6:0]								
				Product Code (8h)				PID[1:0]		MID[1:0]			
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	0	#	#	#	Set CA[6:4]	0
5	Set Temp. Compensation	0	0	0	0	1	0	0	1	#	#	Set TC[1:0]	0
6	Set Power Control	0	0	0	0	1	0	1	0	#	#	Set PC[1:0]	10b
7	Set Adv. Program Control	0	0	0	0	1	1	0	0	0	R	Set APC[R][7:0], R = 0 or 1	N/A
	(double-byte command)	0	0	#	#	#	#	#	#	#	#		
8	Set Scroll Line LSB	0	0	0	1	0	0	#	#	#	#	Set SL[3:0]	0
	Set Scroll Line MSB	0	0	0	1	0	1	#	#	#	#	Set SL[7:4]	0
9	Set Row Address LSB	0	0	0	1	1	0	#	#	#	#	Set RA[3:0]	0
	Set Row Address MSB	0	0	0	1	1	1	#	#	#	#	Set RA[7:4]	0
10	Set V_{BIAS} Potentiometer	0	0	1	0	0	0	0	0	0	1	Set PM[7:0]	40H
	(double-byte command)	0	0	#	#	#	#	#	#	#	#		
11	Set Partial Display Control	0	0	1	0	0	0	0	1	0	#	Set LC[8]	0
12	Set RAM Address Control	0	0	1	0	0	0	1	#	#	#	Set AC[2:0]	001b
13	Set Fixed Lines	0	0	1	0	0	1	0	0	0	0	Set {FLT, FLB}	0
		0	0	#	#	#	#	#	#	#	#		
14	Set Line Rate	0	0	1	0	1	0	0	0	#	#	Set LC[4:3]	10b
15	Set All-Pixel-ON	0	0	1	0	1	0	0	1	0	#	Set DC[1]	0
16	Set Inverse Display	0	0	1	0	1	0	0	1	1	#	Set DC[0]	0
17	Set Display Enable	0	0	1	0	1	0	1	#	#	#	Set DC[4:2]	110b
18	Set LCD Mapping Control	0	0	1	1	0	0	0	#	#	#	Set LC[2:0]	0
19	Set N-Line Inversion	0	0	1	1	0	0	1	0	0	0	Set NIV[4:0]	1DH
				-	-	-	#	#	#	#	#		

20	Set Color Pattern	0	0	1	1	0	1	0	0	0	#	Set LC[5]	0 (BGR)
21	Set Color Mode	0	0	1	1	0	1	0	1	#	#	Set LC[7:6]	10b
22	Set COM Scan Function	0	0	1	1	0	1	1	#	#	#	Set CSF[2:0]	000b
23	System Reset	0	0	1	1	1	0	0	0	1	0	System Reset	N/A
24	NOP	0	0	1	1	1	0	0	0	1	1	No operation	N/A
25	Set Test Control	0	0	1	1	1	0	0	1	TT		For testing only. Do not use.	N/A
	(double-byte command)	0	0	#	#	#	#	#	#	#	#		
26	Set LCD Bias Ratio	0	0	1	1	1	0	1	0	#	#	Set BR[1:0]	11b: 12
27	Set COM End	0	0	1	1	1	1	0	0	0	1	Set CEN[6:0]	159
		0	0	-	#	#	#	#	#	#	#		
28	Set Partial Display Start	0	0	1	1	1	1	0	0	1	0	Set DST[6:0]	0
		0	0	-	#	#	#	#	#	#	#		
29	Set Partial Display End	0	0	1	1	1	1	0	0	1	1	Set DEN[6:0]	159
		0	0	-	#	#	#	#	#	#	#		
30	Set Window Program Starting Column Address	0	0	1	1	1	1	0	1	0	0	Shared with MTP commands: Set WPC0	0
		0	0	-	#	#	#	#	#	#	#		
31	Set Window Program Starting Row Address	0	0	1	1	1	1	0	1	0	1	Shared with MTP commands: Set WPP0	0
		0	0	#	#	#	#	#	#	#	#		
32	Set Window Program Ending Column Address	0	0	1	1	1	1	0	1	1	0	Shared with MTP commands: Set WPC1	127
		0	0	-	#	#	#	#	#	#	#		
33	Set Window Program Ending Row Address	0	0	1	1	1	1	0	1	1	1	Shared with MTP commands: Set WPP1	159
		0	0	#	#	#	#	#	#	#	#		
34	Window Program Mode	0	0	1	1	1	1	1	0	0	#	Set AC[3]	0: Inside
35	Set MTP Operation control	0	0	1	0	1	1	1	0	0	0	Set MTPC[4:0]	10H
		0	0	-	-	-	#	#	#	#	#		

	Command	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0	Action		Default
36	Set MTP Write Mask	0 0 0	0 0 0	1 - -	0 # -	1 # -	1 # -	1 # -	0 # -	0 # #	1 # #	Set MTPM[6:0] MTPM1[1:0]		0
37	Set V_{MTP1} Potentiometer	0 0	0 0	1 #	1 #	1 #	1 #	0 #	1 #	0 #	0 #	Shared with Window Program commands	Set MTP1	N/A
38	Set V_{MTP2} Potentiometer	0 0	0 0	1 #	1 #	1 #	1 #	0 #	1 #	0 #	1 #		Set MTP2	N/A
39	Set MTP Write Timer	0 0	0 0	1 #	1 #	1 #	1 #	0 #	1 #	1 #	0 #		Set MTP3	N/A
40	Set MTP Read Timer	0 0	0 0	1 #	1 #	1 #	1 #	0 #	1 #	1 #	1 #		Set MTP4	N/A

NOTE:

- All other bit patterns other than commands listed above may result in undefined behavior.
- The interpretation of commands (36)~(40) depends on the setting of register MTPC[3].
  - Commands (37)~(40) are shared with commands (30)~(33). These two sets of commands share exactly the same code and control registers. When MTPC[3]=0, they are interpreted as Window Program commands and registers. When MTPC[3]=1, they function as MTP Control commands and registers.
- After MTP ERASE or PROGRAM operation, before resuming normal operation, please always
  a) Remove TST4 power source,
  b) Do a full V_{DD} ON-OFF-ON cycle.
- Under 16-bit bus mode and CD=0, D[15:8] is ignored and only D[7:0] is used. As a result, the bus cycles for commands under 16-bit bus and 8-bit bus are the same, and double-byte commands still need two bus cycles under 16-bit bus mode.

  Example:

  8-bit bus mode:

  Set PL[1:0] = 2'b11 : D[7:0] = 0010 1011

  Set PM[7:0] = 8'h8b : 1st D[7:0] = 1000 0001

  2nd D[7:0] = 1000 1011

  16-bit bus mode:

  Set PL[1:0] = 2'b11: D[15:0] = 0000 0000 0010 1011

  Set PM[7:0] = 8'h8b: 1st D[15:0] = 0000 0000 1000 0001

  2nd D[15:0] = 0000 0000 1000 1011

COMMAND DESCRIPTION

(1) WRITE DATA TO DISPLAY MEMORY

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Write data	1	0	8-bit data written to SRAM							

UC1698u will convert input RAM data to 16-bit of RGB data. Please refer to command `Set Color Mode` for detail of data-write sequence.

(2) READ DATA FROM DISPLAY MEMORY

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Read data	1	1	8-bit data read from SRAM							

Each RGB triplet is stored as 16-bit in the display RAM. Each 16-bit of RGB data takes 1 (/ 2) RAM read cycles for 16 (/ 8) –bit bus mode, respectively. The read out RGB data is *after-extension* for 64K color mode.

R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
1st 8-bit Read								2nd 8-bit Read							

Write/Read Data Byte (commands (1) and (2)) operation uses internal Row Address register (RA) and Column Address register (CA). RA and CA can be programmed by issuing commands `Set Row Address` and `Set Column Address`. If wrap-around (WA, AC[0]) is OFF (0), CA will stop incrementing after reaching the CA boundary, and system programmers need to set the values of RA and CA explicitly. If WA is ON (1), when CA reaches end of column address, CA will be reset to 0 and RA will be increased or decreased, depending on the setting of Row Increment Direction (RID, AC[2]). When RA reaches the boundary of RAM (i.e. RA = 0 or 127), RA will be wrapped around to the other end of RAM and continue.

(3) GET STATUS & PM

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Get Status	0	1	GE	MX	MY	WA	DE	WS	MD	MS
	0	1	Ver	PMO[6:0]						
	0	1	Product Code				PID[1:0]		MID[1:0]	

Status1 definitions:

GE : Green Enhancing enable flag. Green Enhance Mode is disabled when GE = 1.
MX : Status of register LC[1], mirror X.
MY : Status of register LC[2], mirror Y.
WA : Status of register AC[0]. Automatic column/row wrap around.
DE : Display enable flag. DE=1 when display is enabled
WS : MTP Operation succeeded
MD : MTP Option (1 for MTP version, 0 for non-MTP version)
MS : MTP action status

Status2 definitions:

Ver : IC Version Code. 0 or 1.
PMO[6:0] : PM offset value.

Status3 definitions:

Product Code : 1000b (8h)
PID[1:0] : Provide access to ID pins connection status.
MID[1:0] : LCM manufacturer's configuration.

If multiple `Get Status` commands are issued consecutively within one single CD 1⇨0⇨1 transaction, the `Get Status` command will return {Status1, Status2, Status3, Status1, Status2, Status3, Status1..} alternately.

(4) SET COLUMN ADDRESS

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Column Address LSB CA[3:0]	0	0	**0**	**0**	**0**	**0**	CA3	CA2	CA1	CA0
Set Column Address MSB CA[7:4]	0	0	**0**	**0**	**0**	**1**	0	CA6	CA5	CA4

Set SRAM column address for read/write access. CA is counted in RGB triplets, not individual SEG electrode.

CA value range: **0~127**

(5) SET TEMPERATURE COMPENSATION

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Temperature Comp. TC[1:0]	0	0	**0**	**0**	**1**	**0**	**0**	**1**	TC1	TC0

Set V_{BIAS} temperature compensation coefficient (%-per-degree-C)

Temperature compensation curve definition:

00b = -0.00%/°C 01b = -0.05%/°C 10b = -0.15%/°C 11b = -0.25%/°C

(6) SET POWER CONTROL

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Panel Loading PC[1:0]	0	0	**0**	**0**	**1**	**0**	**1**	**0**	PC1	PC0

Set PC[0] according to the capacitance loading of LCD panel.

Panel loading definition: **0b : LCD ≤ 13nF** 1b : 13nF < LCD ≤ 22nF

Set PC[1] to program the build-in charge pump stages. Before changing PC[1] value, always ensure the IC is in a RESET state. Avoid changing PC[1] when the display is enabled.

Pump control definition: 0b = External V_{LCD} **1b = Internal V_{LCD} (x10)**

(7) SET ADVANCED PROGRAM CONTROL

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set APC[R][7:0]	0	0	**0**	**0**	**1**	**1**	**0**	**0**	**0**	R
(Double-byte command)	0	0	APC register parameter							

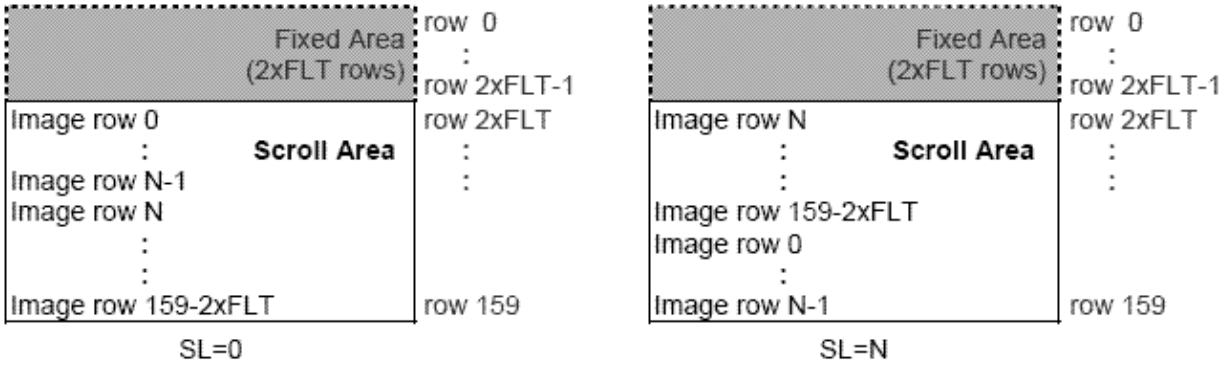
For UltraChip only. Please do NOT use.

(8) SET SCROLL LINE

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Scroll Line LSB SL[3:0]	0	0	**0**	**1**	**0**	**0**	SL3	SL2	SL1	SL0
Set Scroll Line MSB SL[7:4]	0	0	**0**	**1**	**0**	**1**	SL7	SL6	SL5	SL4

Set the scroll line number.

Scroll line setting will scroll the displayed image up by SL rows. The valid value for SL is between 0 (no scrolling) and 159-2x(FLT+FLB) (full scrolling). FLT and FLB are the register values programmed by the `Set Fixed Lines` command.



(9) SET ROW ADDRESS

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Row Address LSB RA [3:0]	0	0	**0**	**1**	**1**	**0**	RA3	RA2	RA1	RA0
Set Row Address MSB RA [7:4]	0	0	**0**	**1**	**1**	**1**	RA7	RA6	RA5	RA4

Set SRAM row address for read/write access.

Possible value = **0~159**

(10) SET V_{BIAS} POTENTIOMETER

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set V_{BIAS} Potentiometer. PM [7:0]	0	0	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**
(Double-byte command)	0	0	PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0

Program V_{BIAS} Potentiometer (PM[7:0]). See section *LCD Voltage Setting* for more detail.

Effective range: **0 ~ 255**

(11) SET PARTIAL DISPLAY CONTROL

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Partial Display Enable LC[8]	0	0	1	0	0	0	0	1	0	LC8

This command is used to enable partial display function.

LC[8] : **0**b: Disable Partial Display, Mux-Rate = CEN+1 (DST, DEN not used.)
1b: Enable Partial Display, Mux-Rate = DEN-DST+1+ LC[0]x(FLT+FLB)x2

(12) SET RAM ADDRESS CONTROL

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set AC [2:0]	0	0	1	0	0	0	1	AC2	AC1	AC0

Program registers AC[2:0] for RAM address control.

AC[0]: WA, Automatic column/row wrap around.
0: CA or RA (depends on AC[1]= 0 or 1) will stop incrementing after reaching boundary
1: CA or RA (depends on AC[1]= 0 or 1) will restart, and RA or CA will increment by one step.

AC[1]: Auto-Increment order
0 : column (CA) increment (+1) first until CA reaches CA boundary, then RA will increment by (+/-1).
1 : row (RA) increment (+/-1) first until RA reach RA boundary, then CA will increment by (+1).

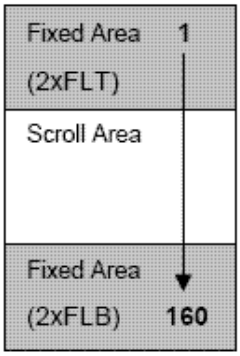
AC[2]: RID, row address (RA) auto increment direction (**0**/1 = +/- 1)
When WA=1 and CA reaches CA boundary, RID controls whether row address will be adjusted by +1 or -1.

AC[2:0] controls the auto-increment behavior of CA and RA. For Window Program mode (AC[3]=ON), see section *Command Description* (32) ~ (35) for more details. If WPC[1:0] and WPP[1:0] values are the default values, the behavior of CA, RA auto-increment will be the same, no matter what the setting of AC[3] is.

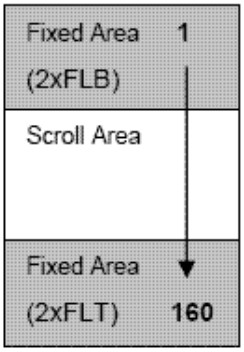
(13) SET FIXED LINES

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Fixed Lines {FLT,FLB}	0	0	1	0	0	1	0	0	0	0
(Double-byte command)	0	0	FLT[3:0]				FLB[3:0]			

The fixed line function is used to implement the partial scroll function by dividing the screen into scroll and fixed area. The `Set Fixed Lines` command will define the fixed area, which will not be affected by the SL scroll function. The fixed area covers the top 2xFLT and bottom 2xFLB rows for mirror Y (MY) is 0, or covers the top 2xFLB and bottom 2xFLT rows for MY=1. One example of the visual effect on LCD is illustrated in the figure below.



MY = 0



MY = 1

When partial display mode is activated, the display of these 2x(FLT+FLB) lines is also controlled by LC[0]. Before turning on LC[0], ensure:

MY=0 DST ≥ FLTx2
DEN ≤ (CEN-FLBx2).

MY=1 DST ≥ FLBx2
DEN ≤ (CEN-FLTx2)

(14) SET LINE RATE

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Line Rate LC [4:3]	0	0	1	0	1	0	0	0	LC4	LC3

Program LC [4:3] for line rate setting (Frame-Rate = Line-Rate / Mux-Rate). The line rate is automatically scaled down by 2/3, 1/2, 1/3 and 1/4 at Mux-Rate = 108, 80, 56, and 40.

The following are line rates at Mux Rate = 109 ~ 160.

00b: 25.2 Klps 01b: 30.5 Klps **10b: 37.0 Klps** 11b: 44.8 Klps

In On/Off Mode

00b: 8.5 Klps 01b: 10.4 Klps **10b: 12.6 Klps** 11b: 15.2 Klps

(Klps: Kilo-Line-per-second)

(15) SET ALL PIXEL ON

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set All Pixel ON DC [1]	0	0	1	0	1	0	0	1	0	DC1

Set DC[1] to force all SEG drivers to output ON signals. This function has no effect on the existing data stored in display RAM.

(16) SET INVERSE DISPLAY

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Inverse Display DC [0]	0	0	1	0	1	0	0	1	1	DC0

Set DC[0] to force all SEG drivers to output the inverse of the data (bit-wise) stored in display RAM. This function has no effect on the existing data stored in display RAM.

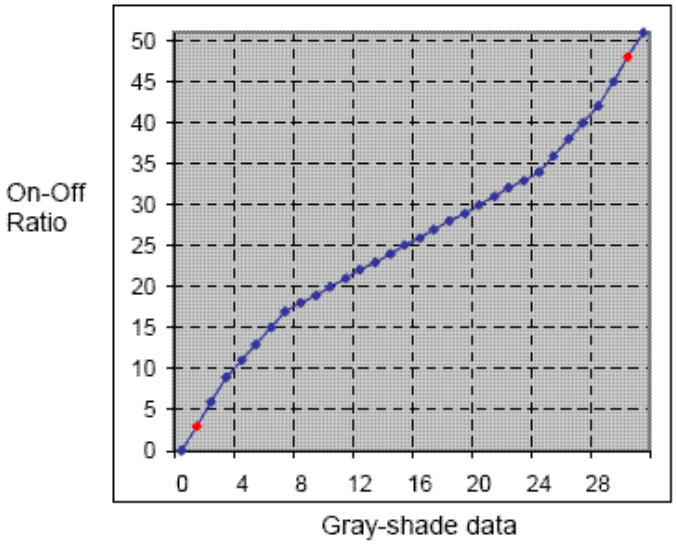
(17) SET DISPLAY ENABLE

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Display Enable DC [4:2]	0	0	1	0	1	0	1	DC4	DC3	DC2

This command is for programming register DC[4:2].

When DC[2] is set to **0**, the IC will put itself into Sleep mode. All drivers, voltage generation circuit and timing circuit will be halted to conserve power. When DC[2] is set to 1, UC1698u will first exit from Sleep mode, restore the power and then turn on COM drivers and SEG drivers. There is no other explicit user action or timing sequence required to enter or exit the Sleep mode.

DC[3] controls the gray shade modulation modes. UC1698u has two gray shade modulation modes: an On/Off mode and a 32-shade mode. The modulation curves are shown below. Horizontal axes are the gray shade data. The vertical axes are the ON-OFF ratio.



DC[4] Green Enhance Mode. Refer to command `Set Color Mode` for more information.

0b: Green Enhancing Mode enabled **1b: Green Enhancing Mode disabled**

NOTE:

1. For red and blue colors, when PWM is off, the shades mapped to data 1 and 30 (shown as red points above) are achieved by special dithering. When PWM is on, these shades are produced by PWM.
2. Green shades are created by combining FRC (default: Off) and special dithering. When PWM is off, six of the shades (1, 2, 3, 59, 60, and 61) are created by special dithering while they are created by PWM when PWM is on. Data 62 and 63 are mapped to the same shade.
3. When the internal DC-DC converter starts to operate and pump out current to V_{LCD}, there will be an in-rush pulse current between V_{DD2} and V_{SS2} initially. To avoid this current pulse from causing potential harmful noise, do *NOT* issue any command or write any data to UC1698u for 5~10mS after setting DC[2] to 1.

(18) SET LCD MAPPING CONTROL

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set LCD Mapping Control LC [2:0]	0	0	1	1	0	0	0	MY	MX	LC0

This command is used for programming LC[2:0] to control COM (row) mirror (MY), SEG (column) mirror (MX).

LC[2] controls Mirror Y (MY): MY is implemented by reversing the mapping order between RAM and COM electrodes. The data stored in RAM is not affected by the MY action. MY will have immediate effect on the display image.

LC[1] controls Mirror X (MX): MX is implemented by selecting the CA or 127-CA as write/read (from host interface) display RAM column address so this function will only take effect after rewriting the RAM data.

LC[0] controls whether soft icon sections (2xFLT, 2xFLB) are displayed during partial display mode.

(19) SET N-LINE INVERSION

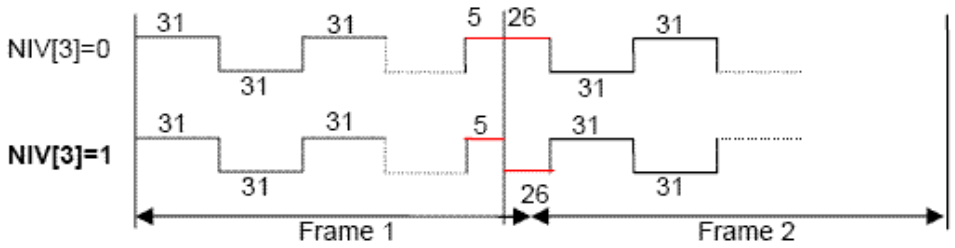
Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set N-line inversion NIV[3:0]	0	0	1	1	0	0	1	0	0	0
(Double-byte command)	0	0	-	-	-	NIV4	NIV3	NIV2	NIV1	NIV0

N-Line Inversion:

NIV[2:0]: 000b: 11 lines; 001b: 19 lines; 010b: 21 lines; 011b: 25 lines; 100b: 29 lines; **101b: 31 lines**; 110b: 37 lines; 111b: 43 lines

NIV[3]: 0b: non-XOR; **1b: XOR**

NIV[4]: 0b: Disable NIV; **1b: Enable NIV**



(20) SET COLOR PATTERN

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Color Pattern LC [5]	0	0	1	1	0	1	0	0	0	LC5

UC1698u supports on-chip swapping of R⇔B data mapping to the SEG drivers.

LC[5]	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	…	SEG382	SEG383	SEG384
0	B	G	R	B	G	R	…	B	G	R
1	R	G	B	R	G	B	…	R	G	B

The definition of R/G/B input data is determined by LC[7:6], as described in *Set Color Mode* below.

(21) SET COLOR MODE

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Color Mode LC [7:6]	0	0	1	1	0	1	0	1	LC7	LC6

Program color mode and RGB input pattern. Color mode (LC[7:6]) definition:

Note: *For serial bus modes, please refer to 8-bit tables below.*

Green Enhance Mode disabled (DC[4]=1):

LC[7:6] = 01b (RRRR-GGGG-BBBB, 4K-color)

12 bits of input RGB data are stored to 16 RAM bits. No dither is performed. Every 3 bytes of input data will be merged into 2 sets of RGB data.

Data Write Sequence (8-bit)	***D[7:0]***							
1st Write Data Cycle	R3	R2	R1	R0	G3	G2	G1	G0
2nd Write Data Cycle	B3	B2	B1	B0	R3	R2	R1	R0
3rd Write Data Cycle	G3	G2	G1	G0	B3	B2	B1	B0

Data Write Sequence (16-bit)	***D[15:0]***															
1st Write Data Cycle	0	0	0	0	R3	R2	R1	R0	G3	G2	G1	G0	B3	B2	B1	B0
2nd Write Data Cycle	0	0	0	0	R3	R2	R1	R0	G3	G2	G1	G0	B3	B2	B1	B0

LC[7:6] = **10b** (RRRRR-GGGGGG-BBBBB, 64K-color)

16 bits of input data are stored to 16 RAM bits directly.

Data Write Sequence (8-bit)	***D[7:0]***							
1st Write Data Cycle	R4	R3	R2	R1	R0	G5	G4	G3
2nd Write Data Cycle	G2	G1	G0	B4	B3	B2	B1	B0

Data Write Sequence (16-bit)	***D[15:0]***															
1st Write Data Cycle	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

Green Enhance Mode enabled (DC[4]=0):

LC[7:6] = 01b (RRRR-GGGGG-BBB, 4K-color)

12 bits of input data are extended and stored to 16 RAM bits. Every 3 bytes of input data will be merged into 2 sets of RGB data.

Data Write Sequence (8-bit)	***D[7:0]***							
1st Write Data Cycle	R3	R2	R1	R0	G4	G3	G2	G1
2nd Write Data Cycle	G0	B2	B1	B0	R3	R2	R1	R0
3rd Write Data Cycle	G4	G3	G2	G1	G0	B2	B1	B0

Data Write Sequence (16-bit)	***D[15:0]***															
1st Write Data Cycle	0	0	0	0	R3	R2	R1	R0	G4	G3	G2	G1	G0	B2	B1	B0
2nd Write Data Cycle	0	0	0	0	R3	R2	R1	R0	G4	G3	G2	G1	G0	B2	B1	B0

LC[7:6] = 10b (RRRRR-GGGGGG-BBBBB, 64K-color)

The behaviors of 8-bit input mode and 16-bit input mode do not change with DC[4] setting. Refer to previous section for more information on these two input modes.

(22) SET COM SCAN FUNCTION

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set COM Scan Function CSF[2:0]	0	0	1	1	0	1	1	CSF2	CSF1	CSF0

COM scan function

CSF[0]: Interlace Scan Function

0b: LRM sequence: AEBCD-AEBCD

1b: LRM sequence: AEBCD-EBCDA

CSF[1]: FRC Function

0b: FRC Disable

1b: FRC Enable

CSF[2]: Shade-1, Shade-30 option

0 : Dither directly on input data(SRAM Change)

1 : PWM on SEG output stage

(23) SYSTEM RESET

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
System Reset	0	0	1	1	1	0	0	0	1	0

This command will activate the system reset. Control register values will be reset to their default values. Data stored in RAM will not be affected.

(24) NOP

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
No Operation	0	0	1	1	1	0	0	0	1	1

This command is used for "no operation".

(25) SET TEST CONTROL

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set TT	0	0	1	1	1	0	0	1	TT	
(Double-byte command)	0	0	Testing parameter							

This command is used for UltraChip production testing. Do *NOT* use.

(26) SET LCD BIAS RATIO

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Bias Ratio BR [1:0]	0	0	1	1	1	0	1	0	BR1	BR0

Bias ratio definition:

00b =5 01b = 10 10b = 11 **11b = 12**

(27) SET COM END

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set CEN	0	0	1	1	1	1	0	0	0	1
(Double-byte command)	0	0	-	*CEN* register parameter						

This command programs the ending COM electrode. CEN defines the number of used COM electrodes, and it should correspond to the number of pixel-rows in the LCD. When the LCD has less than 160 pixel rows, the LCM designer should set CEN to N-1 (where N is the number of pixel rows) and use COM1 through COM-N as COM driver electrodes.

(28) SET PARTIAL DISPLAY START

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set DST	0	0	1	1	1	1	0	0	1	0
(Double-byte command)	0	0	-	*DST* register parameter						

This command programs the starting COM electrode, which has been assigned a full scanning period and will output an active COM scanning pulse.

(29) SET PARTIAL DISPLAY END

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set DEN	0	0	1	1	1	1	0	0	1	1
(Double-byte command)	0	0	-	*DEN* register parameter						

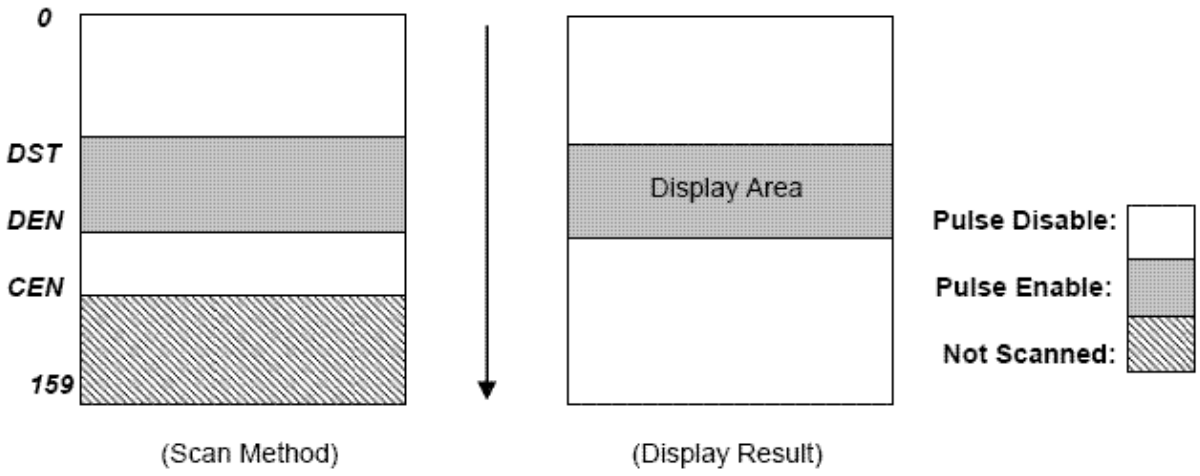
This command programs the ending COM electrode, which has been assigned a full scanning period and will output an active COM scanning pulse.

CEN, DST, and DEN are 0-based index of COM electrodes. They control only the COM electrode activity, and do not affect the mapping of display RAM to each COM electrodes. The image displayed by each pixel row is therefore not affected by the setting of these three registers.

When LC[8]=1b, the Mux-Rate is narrowed down to DST-DEN+1+(FLT+FLB)xLC[0]x2. When MUX rate is reduced, reduce the line rate accordingly to reduce power. Changing MUX rate also require BR and V_{LCD} to be reduced.

For minimum power consumption, set LC[8]=1b, set (DST, DEN, FLT, FLB, CEN) to minimize MUX rate, use slowest line rate which satisfies the flicker requirement, use On/Off mode, set PC[0]=0b, disable N-Line Inversion, and use lowest BR, lowest V_{LCD} which satisfies the contrast requirement. When Mux-Rate is under 40, it is recommended to set BR=5 for optimum power saving.

In either case, DST/DEN defines a small subsection of the display which will remain active while shutting down all the rest of the display to conserve energy.



(30) SET WINDOW PROGRAM STARTING COLUMN ADDRESS

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set WPC0 (Double-byte command)	0	0	1	1	1	1	0	1	0	0
	0	0	-	*WPC0[6:0]* register parameter						

This command is to program the starting column address of RAM program window.

(31) SET WINDOW PROGRAM STARTING ROW ADDRESS

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set WPP0 (Double-byte command)	0	0	1	1	1	1	0	1	0	1
	0	0	*WPP0[7:0]* register parameter							

This command is to program the starting row address of RAM program window.

(32) SET WINDOW PROGRAM ENDING COLUMN ADDRESS

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set WPC1 (Double-byte command)	0	0	1	1	1	1	0	1	1	0
	0	0	-	*WPC1[6:0]* register parameter						

This command is to program the ending column address of RAM program window.

(33) SET WINDOW PROGRAM ENDING ROW ADDRESS

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set WPP1 (Double-byte command)	0	0	1	1	1	1	0	1	1	1
	0	0	*WPP1[7:0]* register parameter							

This command is to program the ending row address of RAM program window.

(34) SET WINDOW PROGRAM MODE

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set Window Program Enable AC[3]	0	0	1	1	1	1	1	0	0	AC3

This command controls the Window Program function.

AC[3]=0: Inside Mode

When Window Programming is under "Inside" mode , the CA and RA increment and wrap-around will be performed automatically around the boundaries as defined by registers WPC0, WPC1, WPP0, and WPP1, so that the CA/RA address will stay *within* the defined window of SRAM address, and therefore allow effective data update within the window.

AC[3]=1: Outside Mode

When Window Programming is under "Outside" mode, the CA and RA increment and wrap-around boundary will cover the entire UC1698u SRAM map (CA: 0~127, RA:0~159). However, when CA/RA points to a memory location within the window defined by registers WPC0, WPC1, WPP0, and WPP1, the SRAM data update operation will be suspended, the existing data will be retained and the input data will be ignored.

The direction of Window Program will depend on the WA (AC[0]), RID (AC[2]), auto-increment order (AC[1]) and MX (LC[1]) register setting.

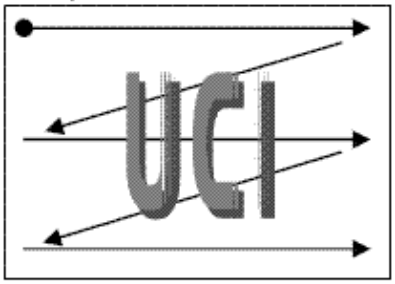
- WA (AC[0]) decides whether the program RAM address advances to next row / column after reaching the specified window column / row boundary.
- RID (AC[2]) controls the RAM address increasing from WPP0 toward WPP1 (RID=0) or the reverse direction (RID=1).
- Auto-increment Order (AC[1]) directs the RAM address increasing vertically (AC[1]=1) or horizontally (AC[1]=0).
- MX (LC[1]) results the RAM column address increasing from 127-WPC0 to 127-WPC1 (MX=1) or from WPC0 to WPC1 (MX=0).

By different combination of RID, AC[1], MX, and by setting CA, RA at proper corners of the "window", effects such as mirrors and rotations can be easily achieved.

Setting or resetting AC[3] does not affect the values of CA and RA. So, always remember to reposition CA and RA properly after changing the setting of AC[3].

Auto-increment order = 0 MX=0 RID = 0

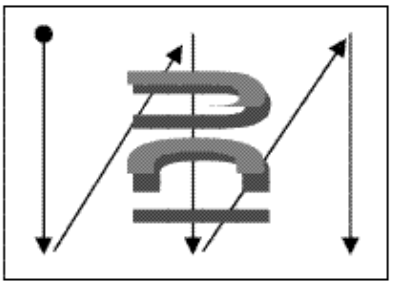
(WPP0,WPC0)



(WPP1,WPC1)

Auto-increment order = 1 MX=0 RID = 0

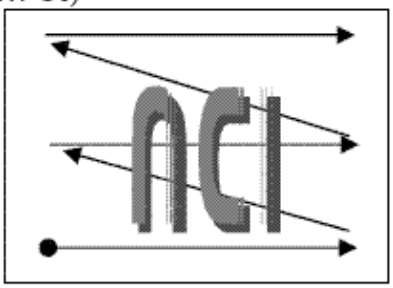
(WPP0,WPC0)



(WPP1,WPC1)

Auto-increment order = 0 MX=0 RID = 1

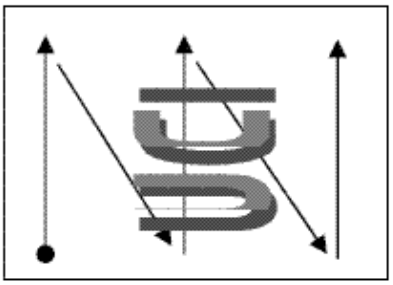
(WPP0,WPC0)



(WPP1,WPC1)

Auto-increment order = 1 MX=0 RID = 1

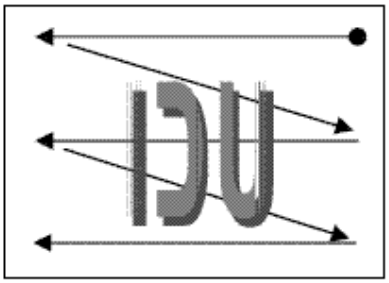
(WPP0,WPC0)



(WPP1,WPC1)

Auto-increment order = 0 MX=1 RID = 0

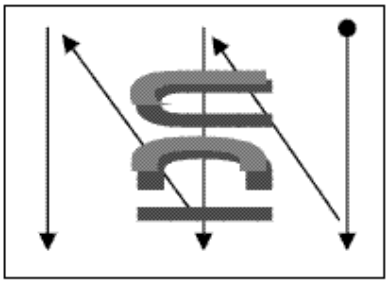
(WPP0,127-WPC0)



(WPP1,127-WPC1)

Auto-increment order = 1 MX=1 RID = 0

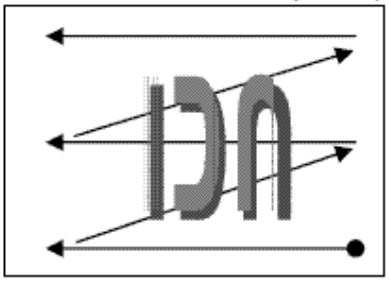
(WPP0,127-WPC0)



(WPP1,127-WPC1)

Auto-increment order = 0 MX=1 RID = 1

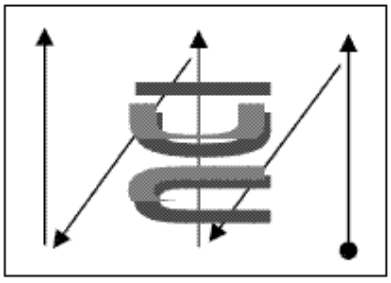
(WPP0,127-WPC0)



(WPP1,127-WPC1)

Auto-increment order = 1 MX=1 RID = 1

(WPP0,127-WPC0)



(WPP1,127-WPC1)

(35) SET MTP OPERATION CONTROL

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set MTPC (Double-byte command)	0	0	1	0	1	1	1	0	0	0
	0	0	-	-	-	*MTPC* register parameter				

This command is for MTP operation control:

MTPC[2:0] : MTP command

000 : Sleep
001 : MTP Read
010 : MTP Erase
011 : MTP Program
1xx : For UltraChip use only.

MTPC[3] : MTP Enable (automatically cleared each time after MTP command is done)
MTPC[4] : MTP value valid (ignore MTP value when L)

DC[2] and MTPC[3] are mutually exclusive. Only one of these two control flags can be set to ON at any time. In other words, when DC[2] is ON, all MTP operations will be blocked, and, when MTP operation is active, set DC[2] to 1 will be blocked.

The following commands, (37) ~ (41), are used as MTP commands only when MTPC[3]=1.

(36) SET MTP WRITE MASK

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set MTPM (Triple-byte command)	0	0	1	0	1	1	1	0	0	1
	0	0	-	MTPM[6:0] register parameter						
	0	0	-	-	-	-	-	-	MTPM1 [1:0]	

This command enables Write to each of the 7 individual MTP bits.

When MTPM[x]=1, the x-th bit of the MTP memory will be programmed to "1". MTPM[x]=0 means no Write action for x-th bit. And the content of this bit will not change.

The amount of "programming current" increases with the number of 1's in MTPM. If the "programming current" appears to be too high for the LCM design (e.g. TST4 ITO trace is not wide enough to supply the current), use multiple write cycles and distribute the 1's evenly into these cycles.

MTPM[6:0] : Set PMO value
MTPM1[1:0]: Set MID value

This command is only valid when MTPC[3]=1.

(37) SET V_{MTP1} POTENTIOMETER

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set MTP1 (Double-byte command)	0	0	1	1	1	1	0	1	0	0
	0	0	Shared register parameter							

This command is for fine tuning V_{MPT1} setting (use with BR=00) and is only valid when MTPC[3]=1.

(38) SET V_{MTP2} POTENTIOMETER

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set MTP2 (Double-byte command)	0	0	1	1	1	1	0	1	0	1
	0	0	Shared register parameter							

This command is for fine tuning V_{MTP2} PM setting (use with BR=01) and is only valid when MTPC[3]=1.

(39) SET MTP WRITE TIMER

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set MTP3 (Double-byte command)	0	0	1	1	1	1	0	1	1	0
	0	0	Shared register parameter							

This command is only valid when MTPC[3]=1.

(40) SET MTP READ TIMER

Action	C/D	W/R	D7	D6	D5	D4	D3	D2	D1	D0
Set MTP4 (Double-byte command)	0	0	1	1	1	1	0	1	1	1
	0	0	Shared register parameter							

This command is only valid when MTPC[3]=1.

Display Data RAM Address Map

HGO160160 DISPLAY DATA RAM

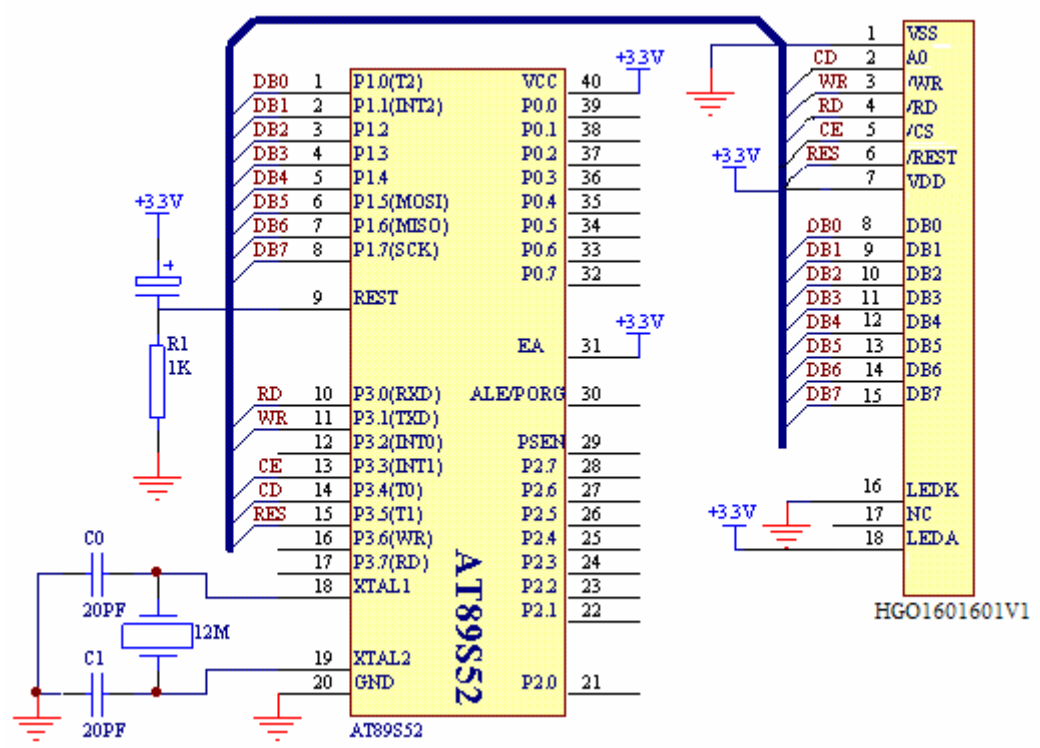
Row Adderss \ Col Adderss	1	2	3	4	5	6	7	8	9						157	158	159	160			MY=0 SL=0	MY=0 SL=16	MY=1 SL=0	MY=1 SL=16
	25H			26H			27H								59H			5AH						
00H																					COM1	COM17	COM160	COM16
01H																					COM2	COM18	COM159	COM15
02H																					COM3	COM19	COM158	COM14
03H																					COM4	COM20	COM157	COM13
04H																					COM5	COM21	COM156	COM12
05H																					COM6	COM22	COM155	COM11
06H																					COM7	COM23	COM154	COM12
07H																					COM8	COM24	COM153	COM13
08H																					COM9	COM25	COM152	COM14
	160*160 DOTS																							
8EH																					COM143	COM159	COM18	COM34
8FH																					COM144	COM160	COM17	COM33
90H																					COM145	COM1	COM16	COM32
91H																					COM146	COM2	COM15	COM31
92H																					COM147	COM3	COM14	COM30
93H																					COM148	COM4	COM13	COM29
94H																					COM149	COM5	COM12	COM28
95H																					COM150	COM6	COM11	COM27
96H																					COM151	COM7	COM10	COM26
97H																					COM152	COM8	COM9	COM25
98H																					COM153	COM9	COM8	COM24
99H																					COM154	COM10	COM7	COM23
9AH																					COM155	COM11	COM6	COM22
9BH																					COM156	COM12	COM5	COM21
9CH																					COM157	COM13	COM4	COM20
9DH																					COM158	COM14	COM3	COM19
9EH																					COM159	COM15	COM2	COM18
9FH																					COM160	COM16	COM1	COM17

MX		1	2	3	4	5	6	7	8	9						157	158	159	160		
	0	SEG112	SEG113	SEG114	SEG115	SEG116	SEG117									SEG268	SEG269	SEG270	SEG271	SEG272	SEG273
	1	SEG271	SEG272	SEG273	SEG268	SEG269	SEG270									SEG115	SEG116	SEG117	SEG112	SEG113	SEG114

Example for memory mapping: let MX=0, MY=1, SL=0, LC[7:6]=10b
(RRRRR-GGGGGG-BBBBB, 64K-color)

6. 应用参考示例

（1）应用电路：



（2）示例程序 ：

```
/*******************************************
24*24 汉字写方式采用 4K 的 RGB，
当模块采用单色屏时 4 个二进制表示一个像素点
二进制的最高为 1 时，像素点被点亮。最高位是 0
时，像素点灭；如 1000 像素点亮， 0111 像素点灭，
程序采用 8，4，2，1 法分解字节
type=1,ascii;type=2,Chinese character
本程序是睬用 6*8 块的方式纵向写入汉字，避免横
向一个地址写 3 个像素的麻烦
*******************************************/

/*LCD controller uc1698
SEGMENT SEG112~SEG271
since the segment is dissymmetrical,do not use
X-mirror extern MPU Crystal:20M */
#include<reg51.h>
#include<stdio.h>
#include<intrins.h>
#define uchar unsigned char
#define uint unsigned int
sbit wr=P3^0;
sbit rd=P3^1;
sbit cs=P3^2;
sbit cd=P3^4;
sbit rst=P3^5;
#define dport P1
uchar code hanzi[]=
{
/* 点阵：24x24
   提取点阵方向：纵向
   字节掉转：是
   字节方式：C 语言 */
0x00,0x80,0x84,0x0E,0x1C,0x3C,0x18,0x0C,  // 清
0x4C,0xEC,0xEC,0xEC,0xEC,0xFF,0xFF,0xFF,
```

```
0xEC,0xEC,0xEC,0xEC,0xEC,0x4C,0x00,0x00,
0x00,0x01,0x03,0xC7,0xEE,0xC2,0x06,0x06,
0x06,0xF6,0xF6,0xF6,0xB6,0xB7,0xB7,0xB7,
0xB6,0xB6,0xF6,0xF6,0xF6,0x06,0x06,0x00,
0x00,0x20,0x7C,0x7F,0x0F,0x00,0x00,0x00,
0x00,0xFF,0xFF,0xFF,0x0D,0x0D,0x0D,0x0D,
0x6D,0xED,0xFF,0xFF,0x7F,0x00,0x00,0x00,

0x00,0x00,0x04,0x0E,0x3E,0x78,0x30,0x00,  // 达
0xE0,0xE0,0xE0,0xE0,0xE0,0xE0,0xFF,0xFF,
0xE1,0xE0,0xE0,0xE0,0xE0,0xE0,0x00,0x00,
0x00,0x0E,0x0E,0x0E,0xFE,0xFE,0x00,0x00,
0x00,0x00,0x80,0xC0,0xF0,0x7E,0x3F,0x1F,
0x38,0x78,0xE0,0xC0,0x80,0x00,0x00,0x00,
0x00,0x60,0x70,0x78,0x1F,0x1F,0x18,0x3A,
0x77,0x77,0x63,0x63,0x61,0x60,0x60,0x60,
0x60,0x60,0x61,0x67,0x63,0x63,0x60,0x00,

0x00,0x00,0x00,0x00,0x18,0x38,0xF8,0xF0,  // 光
0xC0,0x00,0x00,0xFF,0xFF,0x03,0x00,0x80,
0xE0,0xF8,0x7C,0x1C,0x08,0x00,0x00,0x00,
0x00,0x0E,0x0E,0x0E,0x0E,0x0E,0x0E,0x0F,
0xFE,0xFE,0x3E,0x0F,0x0F,0xFE,0xFE,0xFE,
0x0F,0x0E,0x0E,0x0E,0x0E,0x0E,0x0E,0x00,
0x00,0x20,0x70,0x70,0x70,0x38,0x3C,0x1F,
0x0F,0x03,0x00,0x00,0x00,0x3F,0x7F,0x7F,
0x70,0x70,0x70,0x70,0x7C,0x7C,0x38,0x00,

0x00,0x00,0x00,0xF0,0xF0,0xF0,0x70,0x70,  // 电
0x70,0x70,0xFF,0xFF,0xFF,0x70,0x70,0x70,
0x70,0xF0,0xF0,0xF0,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0xFF,0xFF,0xFF,0x8C,0x8C,
0x8C,0x8C,0xFF,0xFF,0xFF,0x8C,0x8C,0x8C,
0x8C,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x0F,0x0F,0x0F,0x03,0x03,
0x03,0x03,0x7F,0xFF,0xFF,0xE3,0xE3,0xE3,
0xE3,0xE3,0xE3,0xE3,0xFC,0x78,0x78,0x00,

0x00,0x40,0x40,0x40,0x40,0xFF,0xFE,0x42,  // 技
0x60,0x40,0x20,0x20,0x20,0x20,0xFF,0xFE,
0x22,0x20,0x20,0x30,0x38,0x30,0x20,0x00,
0x00,0x20,0x60,0x30,0x10,0xFF,0xFF,0x04,
0x02,0x00,0x04,0x3C,0xC4,0x04,0x07,0x07,
0x04,0xE4,0xFC,0x1E,0x04,0x00,0x00,0x00,
0x00,0x20,0x20,0x20,0x60,0xFF,0x7F,0x00,

0x00,0x80,0x40,0x20,0x30,0x1B,0x0E,0x06,
0x0F,0x19,0x30,0x70,0x70,0x20,0x20,0x00,

0x00,0x80,0x80,0x80,0x80,0x80,0x80,0x80,  // 术
0x80,0x80,0x80,0xFF,0xFE,0x82,0x80,0x80,
0x84,0x9C,0xB8,0x90,0xC0,0xC0,0x80,0x00,
0x00,0x00,0x00,0x00,0x00,0x80,0xC0,0x70,
0x3C,0x0F,0x03,0xFF,0xFF,0x00,0x0F,0x38,
0xE0,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x08,0x04,0x06,0x03,0x01,0x00,0x00,
0x00,0x00,0x00,0xFF,0x7F,0x00,0x00,0x00,
0x00,0x01,0x03,0x06,0x0E,0x04,0x04,0x00,

0x00,0x10,0x10,0x10,0x10,0x10,0x10,0x10,  // 有
0x90,0xF0,0x78,0x1F,0x16,0x12,0x10,0x10,
0x10,0x90,0x90,0x18,0x1C,0x18,0x10,0x00,
0x00,0x40,0x40,0x20,0x10,0x18,0x0C,0xFF,
0xFF,0x11,0x11,0x11,0x11,0x11,0x11,0x11,
0x11,0xFF,0xFF,0x01,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,
0x7F,0x01,0x01,0x01,0x01,0x21,0x21,0x61,
0xE1,0x7F,0x3F,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0xFE,0xFC,0x04,0x04,0xE4,  // 限
0x3E,0x0E,0x04,0xFE,0xFC,0x44,0x44,0x44,
0x44,0x44,0x44,0xFE,0xFE,0x04,0x00,0x00,
0x00,0x00,0x00,0xFF,0xFF,0x80,0x83,0x84,
0xF8,0xF0,0x00,0xFF,0xFF,0x08,0x08,0x78,
0x88,0x08,0x88,0x5F,0x6F,0x30,0x20,0x00,
0x00,0x00,0x00,0xFF,0x7F,0x00,0x00,0x03,
0x01,0x00,0x40,0xFF,0x7F,0x30,0x10,0x08,
0x01,0x07,0x0C,0x18,0x30,0x70,0x20,0x20,

0x00,0x00,0x00,0x00,0x00,0x80,0xE0,0x78,  // 公
0x1E,0x0C,0x04,0x00,0x00,0x07,0x1A,0x60,
0xC0,0x80,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x20,0x10,0x0C,0x06,0x03,0x00,0x00,
0xC0,0xF0,0x3E,0x0C,0x04,0x00,0x80,0x00,
0x00,0x01,0x07,0x0E,0x1C,0x0C,0x08,0x00,
0x00,0x00,0x00,0x10,0x30,0x38,0x36,0x13,
0x11,0x10,0x10,0x10,0x10,0x10,0x10,0x11,
0x16,0x1C,0x78,0x30,0x00,0x00,0x00,0x00,

0x00,0x00,0x40,0x44,0x44,0x44,0x44,0x44,  // 司
0x44,0x44,0x44,0x44,0x44,0x44,0x44,0x64,
```

```
0x64,0x44,0x04,0xFE,0xFE,0x04,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0xFC,0xF8,0x08,
0x08,0x08,0x08,0x08,0x08,0xFC,0xFC,0x08,
0x00,0x00,0x00,0xFF,0xFF,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x0F,0x07,0x02,
0x02,0x02,0x02,0x02,0x02,0x0F,0x07,0x20,
0x20,0x60,0xE0,0x7F,0x3F,0x00,0x00,0x00,
};
uchar code ascii[]=
{
/*--  文字:  0  --*/
/*--  宋体 18;  此字体下对应的点阵为：宽 x 高
      =12x24   --*/
0x00,0x00,0x80,0xC0,0x60,0x20,0x20,0x60,0xC0,0x
80,0x00,0x00,0x00,0xFE,0xFF,0x01,
0x00,0x00,0x00,0x00,0x01,0xFF,0xFE,0x00,0x00,0x
01,0x07,0x0E,0x18,0x10,0x10,0x18,
0x0E,0x07,0x01,0x00,
/*--  文字:  1  --*/
/*--  宋体 18;  此字体下对应的点阵为：宽 x 高
      =12x24   --*/
0x00,0x00,0x80,0x80,0x80,0xC0,0xE0,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x10,0x10,0x10,0x1F,0x1F,0x10,
0x10,0x10,0x00,0x00,
/*--  文字:  2  --*/
/*--  宋体 18;  此字体下对应的点阵为：宽 x 高
      =12x24   --*/
0x00,0x80,0x40,0x20,0x20,0x20,0x20,0x60,0xC0,0x8
0,0x00,0x00,0x00,0x03,0x03,0x00,
0x80,0x40,0x20,0x38,0x1F,0x07,0x00,0x00,0x00,0x1
C,0x1A,0x19,0x18,0x18,0x18,0x18,
0x18,0x1F,0x00,0x00,
/*--  文字:  3  --*/
/*--  宋体 18;  此字体下对应的点阵为：宽 x 高
      =12x24   --*/
0x00,0x80,0xC0,0x20,0x20,0x20,0x60,0xC0,0x80,0x
00,0x00,0x00,0x00,0x03,0x03,0x00,
0x10,0x10,0x18,0x2F,0xE7,0x80,0x00,0x00,0x00,0x0
7,0x0F,0x10,0x10,0x10,0x10,0x18,
0x0F,0x07,0x00,0x00,
/*--  文字:  4  --*/
/*--  宋体 18;  此字体下对应的点阵为：宽 x 高
      =12x24   --*/
0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0xE0,0xF0,0x
00,0x00,0x00,0x00,0xC0,0xB0,0x88,
0x86,0x81,0x80,0xFF,0xFF,0x80,0x80,0x00,0x00,0x0
0,0x00,0x00,0x00,0x10,0x10,0x1F,
0x1F,0x10,0x10,0x00,
/*--  文字:  5  --*/
/*--  宋体 18;  此字体下对应的点阵为：宽 x 高
      =12x24   --*/
0x00,0x00,0xE0,0x60,0x60,0x60,0x60,0x60,0x60,0x6
0,0x00,0x00,0x00,0x00,0x3F,0x10,
0x08,0x08,0x08,0x18,0xF0,0xE0,0x00,0x00,0x00,0x0
7,0x0B,0x10,0x10,0x10,0x10,0x1C,
0x0F,0x03,0x00,0x00,
/*--  文字:  6  --*/
/*--  宋体 18;  此字体下对应的点阵为：宽 x 高
      =12x24   --*/
0x00,0x00,0x80,0xC0,0x40,0x20,0x20,0x20,0xE0,0x
C0,0x00,0x00,0x00,0xFC,0xFF,0x21,
0x10,0x08,0x08,0x08,0x18,0xF0,0xE0,0x00,0x00,0x0
1,0x07,0x0C,0x18,0x10,0x10,0x10,
0x08,0x0F,0x03,0x00,
/*--  文字:  7  --*/
/*--  宋体 18;  此字体下对应的点阵为：宽 x 高
      =12x24   --*/
0x00,0x00,0xC0,0xE0,0x60,0x60,0x60,0x60,0x60,0x
E0,0x60,0x00,0x00,0x00,0x03,0x00,
0x00,0x00,0xE0,0x18,0x07,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x1F,0x1F,0x00,
0x00,0x00,0x00,0x00,
/*--  文字:  8  --*/
/*--  宋体 18;  此字体下对应的点阵为：宽 x 高
      =12x24   --*/
0x00,0x80,0xC0,0x60,0x20,0x20,0x20,0x20,0x60,0x
C0,0x80,0x00,0x00,0x87,0xEF,0x2C,
0x18,0x18,0x30,0x30,0x68,0xCF,0x83,0x00,0x00,0x0
7,0x0F,0x08,0x10,0x10,0x10,0x10,
0x18,0x0F,0x07,0x00,
/*--  文字:  9  --*/
/*--  宋体 18;  此字体下对应的点阵为：宽 x 高
      =12x24   --*/
0x00,0x00,0xC0,0xC0,0x20,0x20,0x20,0x20,0xC0,0x
80,0x00,0x00,0x00,0x1F,0x3F,0x60,
0x40,0x40,0x40,0x20,0x10,0xFF,0xFE,0x00,0x00,0x
00,0x0C,0x1C,0x10,0x10,0x10,0x08,
0x0F,0x03,0x00,0x00,
```

```
/*-- 文字: A --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x80,0xE0,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x7C,
0x43,0x40,0x47,0x7F,0xF8,0x80,0x00,0x00,0x10,0x18,0x1F,0x10,0x00,0x00,0x00,0x00,
0x13,0x1F,0x1C,0x10,
/*-- 文字: B --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x20,0xE0,0xE0,0x20,0x20,0x20,0x20,0x60,0xC0,0x80,0x00,0x00,0x00,0xFF,0xFF,0x10,
0x10,0x10,0x10,0x18,0x2F,0xE7,0x80,0x00,0x10,0x1F,0x1F,0x10,0x10,0x10,0x10,0x10,
0x18,0x0F,0x07,0x00,
/*-- 文字: C --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x80,0xC0,0x40,0x20,0x20,0x20,0x20,0x60,0xE0,0x00,0x00,0xFC,0xFF,0x01,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x00,0x00,0x01,0x07,0x0E,0x18,0x10,0x10,0x10,
0x08,0x04,0x03,0x00,
/*-- 文字: D --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x20,0xE0,0xE0,0x20,0x20,0x20,0x20,0x40,0xC0,0x80,0x00,0x00,0x00,0xFF,0xFF,0x00,
0x00,0x00,0x00,0x00,0x01,0xFF,0xFE,0x00,0x10,0x1F,0x1F,0x10,0x10,0x10,0x18,0x08,
0x0E,0x07,0x01,0x00,
/*-- 文字: E --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x20,0xE0,0xE0,0x20,0x20,0x20,0x20,0x20,0x20,0x60,0x80,0x00,0x00,0xFF,0xFF,0x10,
0x10,0x10,0x10,0x7C,0x00,0x00,0x00,0x00,0x10,0x1F,0x1F,0x10,0x10,0x10,0x10,0x10,
0x10,0x18,0x06,0x00,
/*-- 文字: F --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x20,0xE0,0xE0,0x20,0x20,0x20,0x20,0x20,0x60,0x60,0x80,0x00,0x00,0xFF,0xFF,0x10,
0x10,0x10,0x10,0x7C,0x00,0x00,0x01,0x00,0x10,0x1F,0x1F,0x10,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,
/*-- 文字: . --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
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,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1C,0x1C,0x1C,0x00,
0x00,0x00,0x00,0x00,
/*-- 文字: : --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x60,
0xE0,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x18,0x1C,0x1C,0x00,0x00,
0x00,0x00,0x00,0x00,
};
uchar code zimu[]=
{
/*-- 文字: c --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xE0,0xF8,0x18,
0x04,0x04,0x04,0x3C,0x38,0x00,0x00,0x00,0x00,0x03,0x0F,0x0C,0x10,0x10,0x10,0x10,
0x08,0x06,0x00,0x00,
/*-- 文字: o --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xE0,0xF0,0x18,
0x0C,0x04,0x04,0x0C,0x18,0xF0,0xE0,0x00,0x00,0x03,0x0F,0x0C,0x10,0x10,0x10,0x10,
0x0C,0x0F,0x03,0x00,
/*-- 文字: n --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x04,0xFC,0xFC,
0x08,0x08,0x04,0x04,0xFC,0xF8,0x00,0x00,0x00,0x10,0x1F,0x1F,0x10,0x00,0x00,0x10,
```

```
0x1F,0x1F,0x10,0x00,
/*-- 文字: t --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x04,0x04,0x04,
0xFF,0xFF,0x04,0x04,0x04,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x0F,0x1F,0x10,0x10,
0x10,0x0C,0x00,0x00,
/*-- 文字: r --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x04,0x04,0x04,0xFC,
0xFC,0x10,0x08,0x04,0x04,0x0C,0x0C,0x00,0x10,0x
10,0x10,0x1F,0x1F,0x10,0x10,0x10,
0x00,0x00,0x00,0x00,
/*-- 文字: o --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0xE0,0xF0,0x18,
0x0C,0x04,0x04,0x0C,0x18,0xF0,0xE0,0x00,0x00,0x
03,0x0F,0x0C,0x10,0x10,0x10,0x10,
0x0C,0x0F,0x03,0x00,
/*-- 文字: l --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x20,0x20,0x20,0xE0,0xF0,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x10,0x10,0x10,0x1F,0x1F,0x10,
0x10,0x10,0x00,0x00,
/*-- 文字: l --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x20,0x20,0x20,0xE0,0xF0,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x10,0x10,0x10,0x1F,0x1F,0x10,
0x10,0x10,0x00,0x00,
/*-- 文字: e --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0xE0,0xF8,
0x48,0x44,0x44,0x44,0x4C,0x78,0x70,0x00,0x00,0x0
0,0x03,0x0F,0x0C,0x18,0x10,0x10,
0x10,0x08,0x04,0x00,
/*-- 文字: r --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x04,0x04,0x04,0xFC,
0xFC,0x10,0x08,0x04,0x04,0x0C,0x0C,0x00,0x10,0x
10,0x10,0x1F,0x1F,0x10,0x10,0x10,
0x00,0x00,0x00,0x00,
/*-- 文字: : --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x0E,0x0E,0x0E,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x1C,0x1C,0x1C,
0x00,0x00,0x00,0x00,
/*-- 文字: u --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x04,0xFC,0xFE,
0x00,0x00,0x00,0x04,0xFC,0xFE,0x00,0x00,0x00,0x
00,0x0F,0x1F,0x18,0x10,0x10,0x08,
0x1F,0x0F,0x08,0x00,
/*-- 文字: c --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0xE0,0xF8,0x18,
0x04,0x04,0x04,0x3C,0x38,0x00,0x00,0x00,0x00,0x0
3,0x0F,0x0C,0x10,0x10,0x10,0x10,
0x08,0x06,0x00,0x00,
/*-- 文字: 1 --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x80,0x80,0x80,0xC0,0xE0,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x10,0x10,0x10,0x1F,0x1F,0x10,
0x10,0x10,0x00,0x00,
/*-- 文字: 6 --*/
```

```
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x80,0xC0,0x40,0x20,0x20,0x20,0xE0,0xC0,0x00,0x00,0x00,0xFC,0xFF,0x21,
0x10,0x08,0x08,0x08,0x18,0xF0,0xE0,0x00,0x00,0x01,0x07,0x0C,0x18,0x10,0x10,0x10,
0x08,0x0F,0x03,0x00,
/*-- 文字: 9 --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0xC0,0xC0,0x20,0x20,0x20,0x20,0xC0,0x80,0x00,0x00,0x00,0x1F,0x3F,0x60,
0x40,0x40,0x40,0x20,0x10,0xFF,0xFE,0x00,0x00,0x00,0x0C,0x1C,0x10,0x10,0x10,0x08,
0x0F,0x03,0x00,0x00,
/*-- 文字: 8 --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x80,0xC0,0x60,0x20,0x20,0x20,0x20,0x60,0xC0,0x80,0x00,0x00,0x87,0xEF,0x2C,
0x18,0x18,0x30,0x30,0x68,0xCF,0x83,0x00,0x00,0x07,0x0F,0x08,0x10,0x10,0x10,0x10,
0x18,0x0F,0x07,0x00,
/*-- 文字: u --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x04,0xFC,0xFE,
0x00,0x00,0x00,0x04,0xFC,0xFE,0x00,0x00,0x00,0x00,0x0F,0x1F,0x18,0x10,0x10,0x08,
0x1F,0x0F,0x08,0x00,
/*-- 文字: s --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x30,0x78,
0xCC,0xC4,0x84,0x84,0x84,0x0C,0x1C,0x00,0x00,0x00,0x1E,0x18,0x10,0x10,0x10,0x11,
0x19,0x0F,0x06,0x00,
/*-- 文字: t --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x04,0x04,0x04,
0xFF,0xFF,0x04,0x04,0x04,0x00,0x00,0x00,0x00,0x00,0x0F,0x1F,0x10,0x10,0x10,0x08,
0x1F,0x1F,0x10,0x18,
/*-- 文字: t --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x04,0x04,0x04,
0xFF,0xFF,0x04,0x04,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x1F,0x10,0x10,
0x10,0x0C,0x00,0x00,
/*-- 文字: u --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x04,0xFC,0xFE,
0x00,0x00,0x00,0x00,0x04,0xFC,0xFE,0x00,0x00,0x00,0x00,0x0F,0x1F,0x18,0x10,0x10,0x08,
0x1F,0x0F,0x08,0x00,
/*-- 文字: s --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x30,0x78,
0xCC,0xC4,0x84,0x84,0x84,0x0C,0x1C,0x00,0x00,0x00,0x1E,0x18,0x10,0x10,0x10,0x11,
0x19,0x0F,0x06,0x00,
/*-- 文字: : --*/
/*-- 宋体 18; 此字体下对应的点阵为：宽 x 高 =12x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x0E,0x0E,0x0E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1C,0x1C,0x1C,
0x00,0x00,0x00,0x00,
};
void delay_us(uint us)
{
```

```
    while(us--);
}
void delay_ms(uint ms)
{
    uchar i;
    while(ms--)
        for(i=0;i<120;i++);
}
void write(bit flag,uchar dat)
{
    cs=0;
    cd=flag;            //flag=0,write
command;flag=1,write data
    wr=0;
    rd=1;
    dport=dat;
    wr=1;
}
void init(void)//*
{
    cs=0;
    /*reset*/
    rst=0;
    delay_ms(2);        //power on reset
    rst=1;
    delay_ms(200);
    write(0,0xe2);      //reset by command
    delay_ms(2);
    /*power control*/
    write(0,0xe9);      //Bias Ratio:1/10 bias
    write(0,0x2b);      //power control set
                          as internal power
    write(0,0x25);     //set temperate
                          compensation as 0%
    write(0,0x81);      //electronic potentionmeter
    write(0,0xbf);
    /*display control*/
    write(0,0xa4);      //all pixel off
    write(0,0xa6);      //inverse display off
    /*lcd control*/
    write(0,0xc0);     //19:partial display and
                          MX disable,MY enable
    write(0,0xa3);     //line rate 15.2klps
    write(0,0xd1);      //rgb-rgb
    write(0,0xd5);      //4k color mode
    write(0,0x84);      //12:partial display
                          control disable
    /*n-line inversion*/
    write(0,0xc8);
    write(0,0x10);      //enable NIV
    /*com scan fuction*/
    write(0,0xda);          //enable
FRC,PWM,LRM sequence
    /*window*/
    write(0,0xf4);      //wpc0:column
    write(0,0x25);      //start from 130
    write(0,0xf6);      //wpc1
    write(0,0x5A);      //end:272
    write(0,0xf5);      //wpp0:row
    write(0,0x00);      //start from 0
    write(0,0xf7);      //wpP0
    write(0,0x9F);      //end 160
    write(0,0xf8);      //inside mode
    write(0,0x89);      //RAM control,显示方向
    write(0,0xad);      //display on,select
          on/off mode.Green Enhance mode disable
    /*scroll line*/
    write(0,0x40);      //low bit of scroll line
    write(0,0x50);      //high bit of scroll line
    write(0,0xc4);      //19,enable FLT and FLB
    write(0,0x90);      //14:FLT,FLB set
    write(0,0x00);
    /*partial display*/
    write(0,0x84);       //12,set partial display
                          control:off
    write(0,0xf1);      //com end
    write(0,0x9f);      //160
    write(0,0xf2);      //display start
    write(0,0);         //0
    write(0,0xf3);      //display end
    write(0,159);       //160
}
void words(uchar x,uchar y,uchar type,uchar *p)
          //type=1,ascii;type=2,Chinese character
{
uchar i,k,j,m,n,l,x0,dat0,dat1,dat2,dat3,dat4,dat5,dat6;
 x=37+x;//地址从 25H 开始的，每个地址对应 3 个
 像素点【3 个像素点对应 RGB】X=0~54
    x0=0x00|(x&0x0f);
    x=0x10|((x&0xf0)>>4);          //右移 4 位
```

```
    for(i=0;i<3;i++)//纵向 3 个字节  3*8=24 行
    {
        n=i*12*type;
        for(j=0;j<8;j++)//纵向每个字节 8 个点
        {
            m=i*8+j;
                //每 8 行的变换，每行 3 个字节
            write(0,0x89);
              //SET RAM ADDRESS CONTROL
            write(0,x0);
              //Set Column Address LSB CA[3:0]
            write(0,x);
             //Set Column Address MSB CA[7:4]
            write(0,0x60|((y+m)&0x0f));
              //Set Row Address LSB RA [3:0]
            write(0,0x70|(((y+m)&0xf0)>>4));
              //Set Row Address MSB RA [7:4]
            for(k=0;k<2*type;k++)
//数据转换 每次转换完成 6 个点的转换，转换后 6 个点变成 2 个 12 点，3 个字节，字符是 2 次循环【12 点】
            {
                l=k*6+n;
                dat6=0x01<<j;
      //取出对应位，j 表示 8 位数据变量
                dat0=(*(p+l))&dat6;
     //数组对应的数据
                dat0=dat0>>j;
                dat0<<=7;
     //转换后的高四位
                dat1=(*(p+l+1))&dat6;
     //数组对应的数据
                dat1=dat1>>j;
                dat1<<=3;
    //转换后的低四位
                dat2=*(p+2+l)&dat6;
    //数组对应的数据
                dat2=dat2>>j;
                dat2<<=7;
    //转换后的高四位
                dat3=(*(p+3+l))&dat6;
    //数组对应的数据
                dat3=dat3>>j;
                dat3<<=3;
    //转换后的低四位
                dat4=(*(p+4+l))&dat6;
     //数组对应的数据
                dat4=dat4>>j;
                dat4<<=7;
     //转换后的高四位
                dat5=(*(p+5+l))&dat6;
     //数组对应的数据
                dat5=dat5>>j;
                dat5<<=3;
      //转换后的低四位

     //转换后的合成 8 位数
                write(1,dat0|dat1);
                write(1,dat2|dat3);
                write(1,dat4|dat5);
            }
            write(0,0x88);
//SET RAM ADDRESS CONTROL，停止自加
        }
    }
    write(0,0x89);          //SET RAM
                             ADDRESS CONTROL
}
void picture(uchar *p)   //120*120 图形显示程序
{
    uchar i,k;
    for(i=0;i<120;i++)
    {
            write(0,0x60|(i&0x0f));
          //Set Row Address LSB RA [3:0]
            write(0,0x70|((i&0xf0)>>4));
      //Set Row Address MSB RA [7:4]
        write(0,0x05);
    //culomn address LSB//25H 开始
        write(0,0x12);
    //culomn address MSB
            for(k=0;k<60;k++)
     //120 个点，每个点占 4 位
            {
                write(1,*p++);//写数据
            }
    }
}
void lcdscan(uchar dat1,uchar dat2)
//奇数行时写入 dat2  偶数行写入 dat1
```

```
{
    uchar i,j;
    write(0,0x60);      //row address LSB
    write(0,0x70);      //row address MSB
    write(0,0x05);      //culomn address LSB
    write(0,0x12);      //culomn address MSB
    for(j=0;j<160;j++)
    {
        for(i=0;i<27;i++)
        {
            if(j%2==0)
            {
                write(1,dat1);
                write(1,dat1);
                write(1,dat1);
            }
            else
            {
                write(1,dat2);
                write(1,dat2);
                write(1,dat2);
            }
        }
    }
}
 void lcdscan11(uchar dat1)
{
    uchar i,j;
    write(0,0x60);      //row address LSB
    write(0,0x70);      //row address MSB
    write(0,0x05);      //culomn address LSB
    write(0,0x12);      //culomn address MSB
    for(j=0;j<1;j++)
    {
        for(i=0;i<1;i++)
    {
    write(1,dat1);
    write(1,dat1);
    write(1,dat1);
        delay_ms(50);
    }}
}
void character(void)
{
    uchar *q;
    uchar i,j;
    //temp[3],table[6];
    lcdscan(0x00,0x00);
for(i=0;i<4;i++)//清达光电
    {
        q=hanzi+i*72;
      j=9*i+4;   //   j=8*i+4;// 每个汉字是24*24 点的，计算其地址，24 个点占用地址为 8 个
        words(j,10,2,q);
    }
    for(i=0;i<11;i++)//controller:
    {
        q=zimu+i*36;
        j=4*i+4;
        words(j,40,1,q);
    }
    for(i=11;i<18;i++)//uc1698u
    {
        q=zimu+i*36;
        j=4*(i-11)+4;
        words(j,70,1,q);
    }
    for(i=18;i<25;i++)//status:
    {
        q=zimu+i*36;
        j=4*(i-18)+1;
        words(j,100,1,q);
    }
}
void main(void)
{
//    uchar *p;
    //uchar i,j,k;
//loop:
init(); /*
while(1)
{
            write(0,0x84);
    //12,set partial display off
                lcdscan(0xf0,0x0f);     //雪花
                delay_ms(500);
        write(0xff,0xff);               //all pixel on
        delay_ms(500);
                lcdscan(0xff,0x00);     //横线
                delay_ms(500);
```

```
                character();                                    // character();
        delay_ms(500);                                         lcdscan11(0xff);
                */                                                 while(1);
                lcdscan(0x00,0x00);                        lcdscan(0x00,0x00);
                  write(0,0x8f);                                 //      goto loop;
                  write(0,0xc0);                               // }
                //write(0,0xdf);                             }
```

7. 质量标准

7.1 合格质量标准

检验项目	检测标准	AQL
电气特性	GB2828-81 检测水平 II 常规检测 单个样品检测	0.65
非电气特性	GB2828-81 检测水平 II 常规检测 单个样品检测	1.5
尺寸测量	GB2828-81 检测水平 II 常规检测 单个样品检测	1.5

7.2 检验环境条件

- 室温: 25±3 °C
- 湿度: 65±20%RH

7.3 检验标准

7.3.1 加电检测

检测项目	检测标准
无显示	任何像素有不显示的情况，视为不合格品
显示错误	不允许不当操作 在所选择模式，出现异常显示或显示位置不正确
显示不正常	任何一列显示不正常，视为不合格品
过流	总电流要求与模块所需电流相匹配，不允许超过模块正常工作的最大电流值.
视角	视角不要接近规格书所标最小值，如果有接近最小值的产品做不合格品处理.
对比度	对比度不要接近规格书所标最小值，如果有接近最小值的产品做不合格品处理.
LCD驱动电压	见产品规格书

7.3.2 不加电检测

检测项目	检测标准
模块尺寸	见模块外形图，尺寸不允许超出公差范围
液晶屏面板划伤	如果有效区的划伤长和宽尺寸大于下面所示组合，我们做不合格处理。 数目：一个或更多 宽度：0.1 长度 ：3.0 三个或更多 宽度：0.05 长度：2.0 三个或更多 宽度：0.03 长度：3.0 当损坏超出这些尺寸，按不合格品处理.

8.可靠性

测试项目	测试条件	备注
高温存储	70 °C,12hr.	2
低温存储	-20°C,4hr	2
湿度存储	40 °C,90%RH,96hr	1、2
高温运行	40°C, 典型运行条件,48hr	
低温运行	0°C, 典型运行条件,48hr	
震动	加速度: 100m/s^2 ,冲击时间: 11ms, XYZ每个方向6次	
机械振动	10~55Hz sweep（扫描速度）, 3G, 振幅=0.75mm(max)， XYZ各方向20分钟	

备注1：在模块上不允许有任何水珠.

备注2：模块应该在正常条件下储存4小时后进行检测。

9. 出厂测试报告

（VDD=+3.3V ,Ta=25℃）

项目	条件	检测结果	备注
高温存储	80℃,120 hrs	无异常	---
低温存储	- 30℃,120 hrs	无异常	---
高温运行	70℃,240 hrs	无异常	---
低温运行	- 20℃,240 hrs	无异常	---
高温湿存储	50℃,90% RH,120 hrs	无异常	---
高温湿运行	40℃,90% RH,120 hrs	无异常	---
热震动	-20℃, 30min→+25℃, 5min→+60℃, 30min	无异常	10 cycles

10. 注意事项

10.1 使用过程中注意事项

在液晶显示模块出厂前，我们已经做了精确的装配和调试, 因此客户在使用操作时请注意以下几点:

（1） 模块上装有 LCD 屏，必须避免剧烈震动、冲击、挤压和从高处掉落。

（2） 液晶显示模块避免扭动,拆卸金属钮角.

（3） 液晶显示模块避免在印有线路的工作平台上操作.

（4） 避免接触,调整,修改导电橡胶.

（5） 防止施加直流电.

（6） 液晶显示屏中的液晶材料是有害物质,当不慎溅落到手,身体,衣服等处时,应尽快用肥皂冲洗干净。

10.2 安装注意

液晶模块由两片带有偏光片的薄玻璃组装而成，它被固定在带有安装孔的 PCB 板上之后，很容易损坏。必须谨慎处理 LCD，模块才可以被安装。

10.3 LCD 处理及清洗注意事项

在清理显示屏表面时，使用带溶剂（建议如下）的软布，轻轻擦拭。

（1）异丙醇

（2）乙醇

（3）不要用干燥或者比较硬的材料擦拭显示屏表面，否则很容易损坏表面偏光片。

以下溶剂请不要使用：

（1）水

（2）酮

（3）芳烃

10.4 严防静电

LCD 驱动电路是低压、微功耗的 COMS 电路，因此我们建议将任何没有使用的输入终端连接到 VDD 或 VSS 上，在打开电源之前，请不要输入任何信号，并且保证人体、工作台及组装设备良好接地，严防静电.。我们推荐以下措施：

（1）在装配使用液晶显示模块前,请不要将其从包装袋中取出. 液晶显示模块所使用的包装袋是经过防静电处理的特殊包装袋. 在储存液晶显示模块时也要带有包装袋储存,或者放在能充分接地的容器中储存.

（2）在操作液晶显示模块时,要始终保持操作人充分接地. 使人体和液晶模块保持同一电位。

（3）在操作过程中所需的设备要充分接地.尤其是驱动器,必须良好接地,没有漏电,以避免干扰.

（4）液晶模块表面都有一层保护膜, 目的在于避免造成 LCD 的偏光片划伤,沾染污渍等. 请慢慢揭去液晶显示模块保护膜.如果快速揭去保护膜都将产生静电.

（5）注意厂房的湿度：厂房湿度范围: 50~60%RH

10.5 电流保护装置

液晶显示模块上没有装电流保护装置,因此,在使用时应预备好电流保护装置.驱动电压直流成分越小越好，最好不超过 50mV，长时间施加过大的直流成分会使电极产生电化学反应而老化。

10.6 操作注意事项

（1） 液晶模块如果输入电压过高会缩短它的寿命，所以对液晶模块的输入电压进行限制是很重要的。

（2） 液晶模块在低温运行时响应时间相对于正常工作温度将明显变慢，另一方面，在高温环境运行 LCD 屏将变黑。但是，这些现象并不意味着模块故障或 LCD 失控，当温度调整到正常工作范围时，模块工作恢复正常。

（3） 如果在运行过程中有些显示区域无法驱动，导致某些字符显示异常，但是重启一次将会恢复正常。

（4） 终端如果有轻微裸露都将引起电化学反应导致终端开路。

（5） 如果工作温度在最高工作温度，那么要求湿度小于等于 50%RH.

10.7 焊接注意事项:

在焊接液晶显示模块时须注意以下几点:

※ 液晶显示模块上只有输入/输出连线处可以焊接.

※ 焊接所需的烙铁必须绝缘.

（1）焊接时所需条件:

电铁的温度: 280℃±10℃

焊接时间: < 3-4S

焊接材料: 低熔点,可充分熔化的焊锡

避免使用融化后易流动的焊锡,因为在焊接时易渗透到液晶显示模块里面,在清理时易对液晶模块造成污染.此外,为了避免焊接时焊锡对液晶显示模块的污染,应在焊接完成后再揭去液晶显示模块的保护膜.

（2）重复焊接时注意事项:

由于连接线是穿过模块的焊盘与模块焊接的,所以在拆除时需等到焊锡完全熔化时再移动连接线.若焊锡未能完全熔化就用力移动连接线,就极易造成焊盘损坏或脱落.在拆除连接线时最好使用”吸枪”.此外还应注意,重复焊接不得超过 3 次.

10.8 包装与存储

当液晶显示模块需要长时间储存时,应遵循以下原则:

如果储存方法不当,将影响偏光片的质量,使显示效果不佳；容易造成焊盘的氧化，容易焊接。

（1）储存时尽可能使用出厂时的原包装.

（2）储存散装的液晶显示模块时,应先装入防静电袋,封口严密.

（3）为防止模块性能退化，不要暴露在高湿温环境或有阳光直射的位置对它直接操作或存储。

（4）储存应保持低湿度,最佳储存温度范围为: 0℃~35℃

（5）存储时不允许任何东西接触到偏光片表面。

11.使用须知

（1）在合作双方认为有必要提供定制样品的情况下应该提供样品。合同在样品设计好并且双方确认后生效。

（2）在遇到以下情况，必须经双方代表讨论并且同意后处理问题：

-当产品规格书出现问题时。

-当一个新的问题出现，而在此产品规格书中没有说明时。

-如果客户的检测标准或运行条件改变要告知清达，这些改变将使产品规格书出现问题。

-当一个新的问题在客户操作过程中出现，经分析样品也存在该问题时。