Translated from Chinese (Simplified) to English - www.onlinedoctranslator.com

Instruction Manual

Product Type: Dot MatrixLCDLCD module

Product number:12864B-V2.0With Chinese character library

client: _____

Customer Number: _____

date: _____

Confirmation (stamp)

manufacturer	client

head record

1 .revision history

Version	issue date	New/revised content
V2.0	2016-7-25	new system

Note: The upgraded version is backward compatible without further notice.

Compatibility issues affect performance, please contact our company to solve

2.overview

12864BIs a Chinese character libraryDot-matrix liquid crystal display module, which can be used to display letters, numbers, Chinese characters and custom graphics. Divided into 4-bit parallel, 8-bit parallel and serial data transmission. Provide displayRAM, Chinese font library and character generator.

There are 2M Chinese font library ROM (CGROM) inside the module, including 8192 16x16 Chinese glyphs, 32K half-width alphabetic character ROM (HCGROM), 126 8x16 alphabetic glyphs, and a 128x64 drawing RAM (DGRAM) It can realize mixed display of text and graphics, 4x16x16 custom character RAM (CGRAM) can provide character creation function, and 4x16x16 display RAM (DDRAM).

Display resolution: 128 X 64dots

Display color and backlight color: STN blue, yellow green, gray; backlight black, white, yellow green

Polarizing film: fully transparent/semi-transparent

Observation angle:6:00

Display duty cycle:1/32Drive Bias:1/6

Control chip:ST7920or compatibleIC

Chinese charactersROM (CGROM): 2M bits (8192Chinese glyphs) character generator ROM (CGROM): 16K bits (126half-width characters) display dataRAM (DDRAM): 64X16 bits (most4OKx16words,LCDDisplay up to4OKx8characters or16half-width characters) **size (Unit: mm)**

Dimensions: 93X70X12.5

Viewable area:72X40

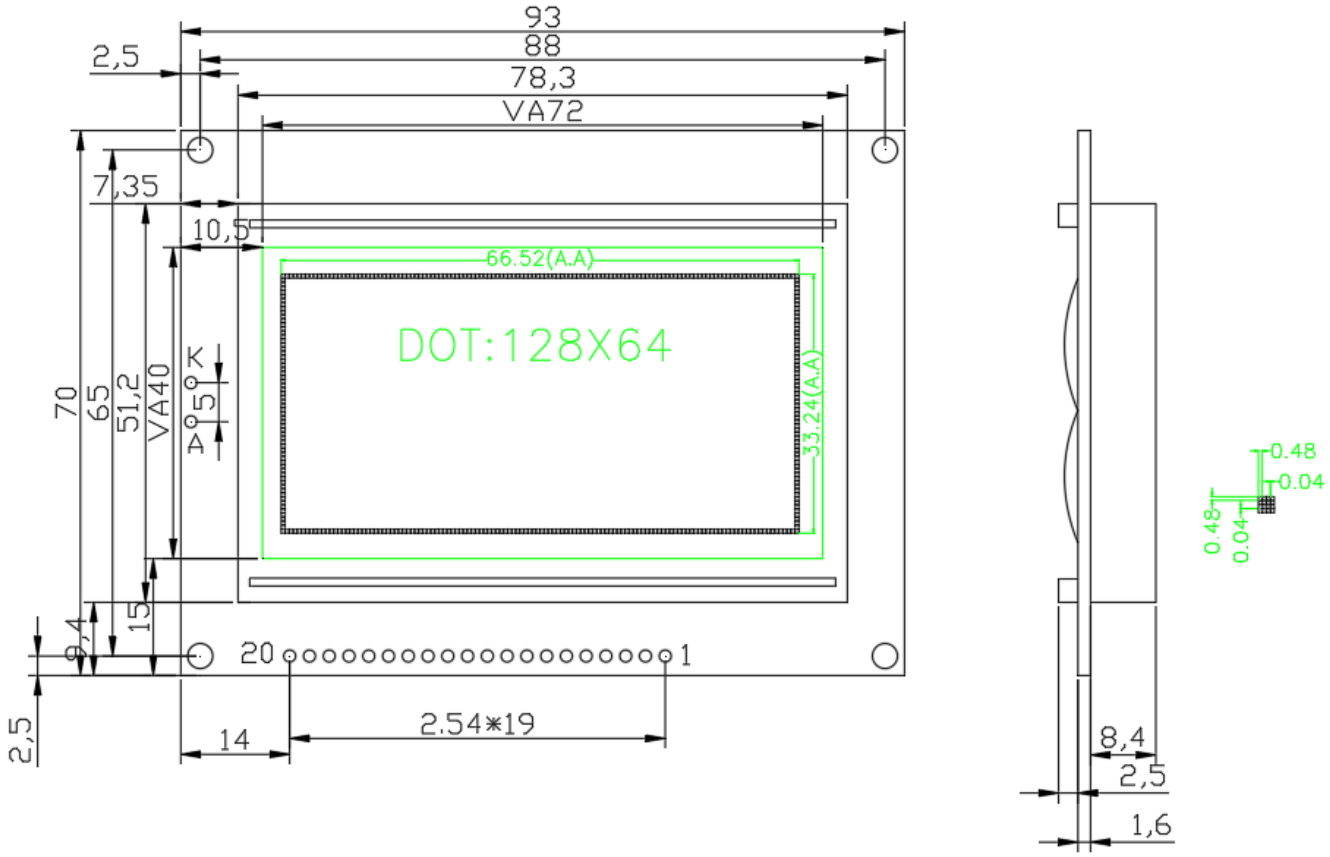
Character size:66.52X33.24 Point Size:0.48X0.48

Pixel Pitch:0.52X0.52

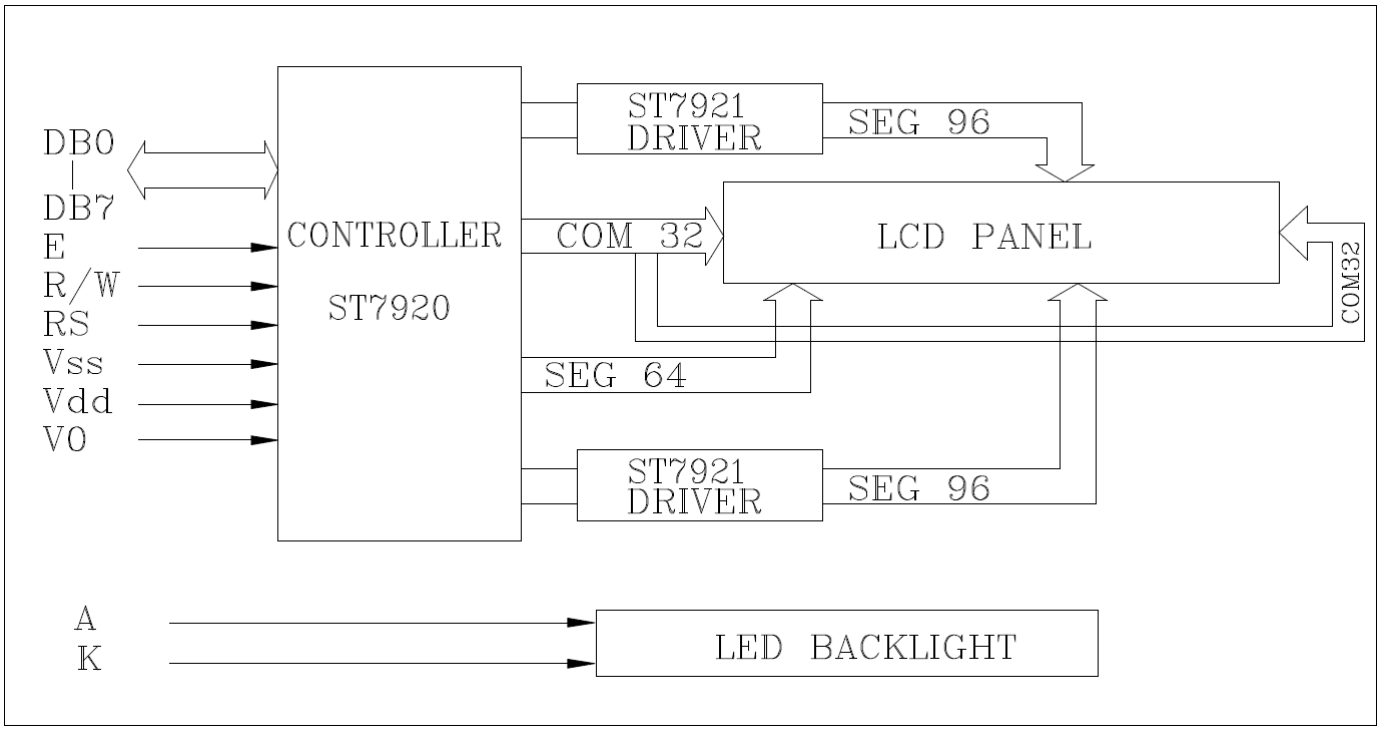
weight: **g** Contrast:V0External adjustment or internal fixed contrast

Operating Voltage: +3.3V or +5Vdefault5V

3. Dimensions:



4.Hardware block diagram:



5.electrical characteristics

5.1Limit parameter

parameter name	symbol	condition	typical value		unit
			minimum value	maximum value	
voltage	Vdd		- 0.3	+ 6.0	V
LCDdriving voltage	Vlcd		- 0.3	+ 7.0	V
Input voltage	Vi		- 0.3	Vdd+0.3	V
Operating temperature(T)	Top	-	- 20	70	℃
Storage temperature(T)	Tstg	-	- 30	80	℃

5.2.1DC parameters1(Ta=25oC, Vdd=4.5V~5.5V)

parameter name	symbol	condition	nominal value			unit
			the smallest	typical	maximum	
voltage	Vdd-GND	-	4.5	5.0	5.5	V
Operating current (excluding backlight)	Idd	Vdd=5V	-	0.45	0.75	mA
LCDdriving voltage	Vdd-V5		3.0	-	Vdd	V
ledBacklight working current	If	Vf=2.8~3.0V	51	54	60	mA
ledBacklight power consumption	PD		230	270	330	mW
input high level	Vih		0.7Vdd	-	Vdd	V
input low level	Vil		- 0.3	-	0.6	V
output high level	Voh	Ioh=-0.1mA	0.8Vdd	-	Vdd	V
output low level	Vol.	Io1=0.1mA	0	-	0.4	V

5.2.2DC parameters2(Ta=25oC, Vdd=2.7V~4.5V)

parameter name	symbol	condition	nominal value			unit
			the smallest	typical	maximum	
voltage	Vdd-GND	-	2.7	3.3	4.5	V
Operating current (excluding backlight)	Idd	Vdd=3.3V	-	0.25	0.45	mA
LCDdriving voltage	Vdd-V5		3.0	-	7	V
ledBacklight working current	If	Vf=2.8~3.0V	51	54	60	mA
ledBacklight power consumption	PD		138	179	270	mW
input high level	Vih		0.7Vdd	-	Vdd	V
input low level	Vil		- 0.3	-	0.6	V
output high level	Voh	Ioh=-0.1mA	0.8Vdd	-	Vdd	V
output low level	Vol.	Iol=0.1mA	-	-	0.1	V

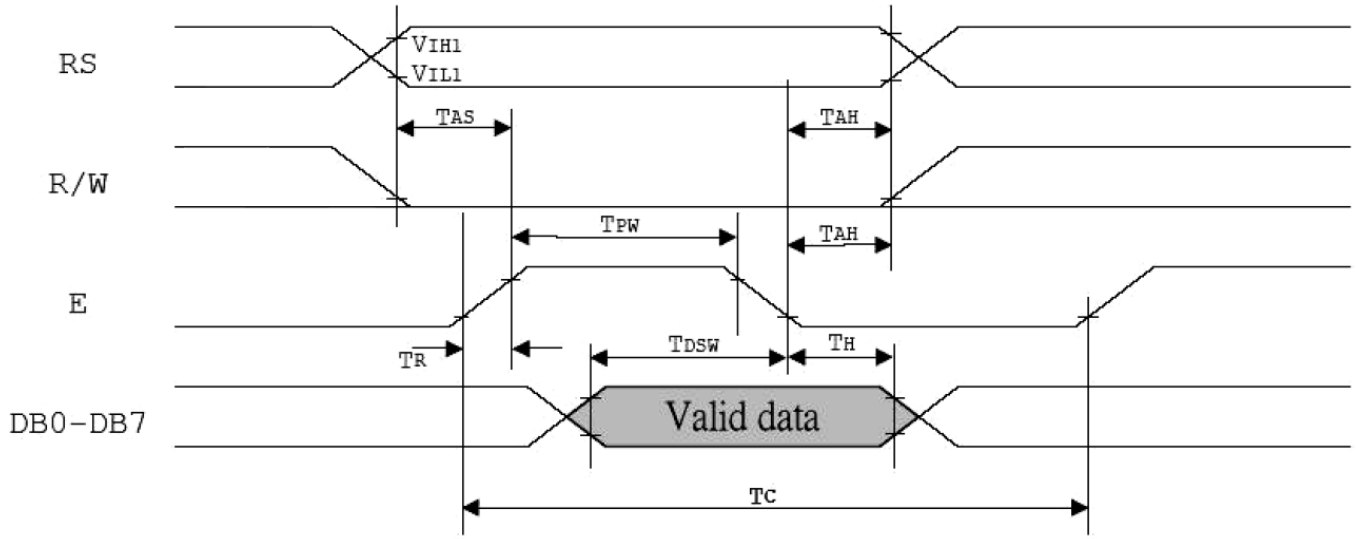
5.3.1AC parameter1(Ta=25oC, Vdd=4.5V~5.5V)parallel interface mode

参数	符号	测试条件	最小	典型	最大	单位
写模式（从 MPU 写数据到 LCM ）						
E 周期	T_C	E	1200	-	-	ns
E 脉冲宽度	T_{PW}	E	140	-	-	ns
E 上升/下降时间	T_R，T_F	E	-	-	25	ns
地址建立时间	T_{AS}	RS、RW、E	10	-	-	ns
地址保持时间	T_{AH}	RS、RW、E	20	-	-	ns
数据建立时间	T_{DSW}	DB0～DB7	40	-	-	ns
数据保持时间	T_H	DB0～DB7	20	-	-	ns
读模式（从 LCM 读数据到 MPU）						
E 周期	T_C	E	1200	-	-	ns
E 脉冲宽度	T_{PW}	E	140	-	-	ns
E 上升/下降时间	T_R，T_F	E	-	-	25	ns
地址建立时间	T_{AS}	RS、RW、E	10	-	-	ns
地址保持时间	T_{AH}	RS、RW、E	20	-	-	ns
数据延迟时间	T_{DDR}	DB0～DB7	-	-	100	ns
数据保持时间	T_H	DB0～DB7	20	-	-	ns

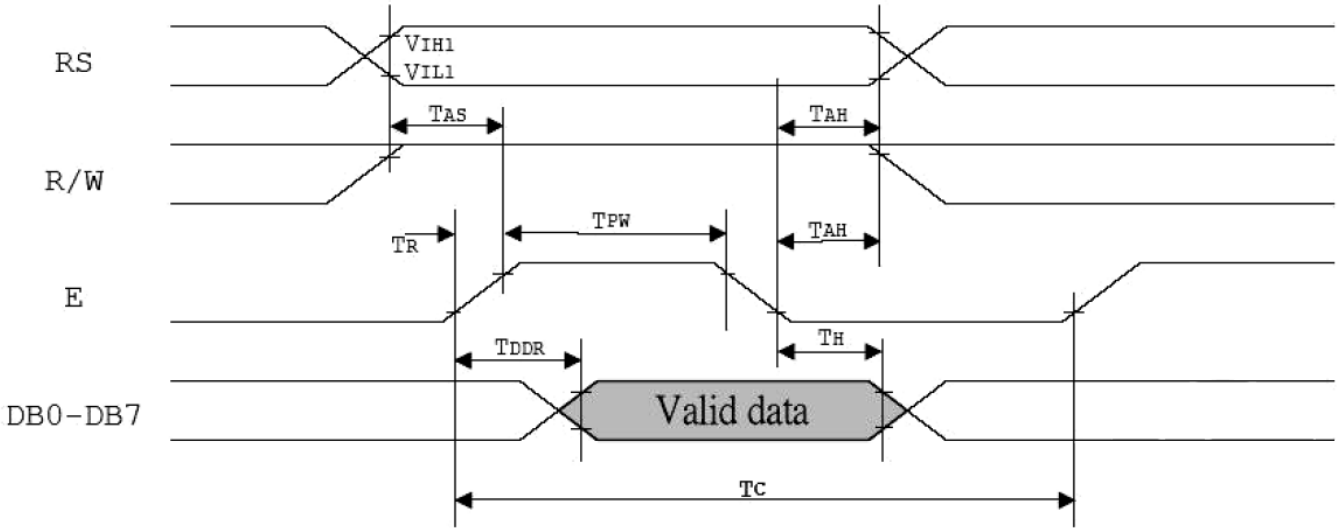
5.3.2AC parameter2(Ta=25oC, Vdd=2.7V~4.5V)parallel interface mode

参数	符号	测试条件	最小	典型	最大	单位
写模式（从 MPU 写数据到 LCM ）						
E 周期	T_C	E	1800	-	-	ns
E 脉冲宽度	T_{PW}	E	160	-	-	ns
E 上升/下降时间	T_R，T_F	E	-	-	25	ns
地址建立时间	T_{AS}	RS、RW、E	10	-	-	ns
地址保持时间	T_{AH}	RS、RW、E	20	-	-	ns
数据建立时间	T_{DSW}	DB0～DB7	40	-	-	ns
数据保持时间	T_H	DB0～DB7	20	-	-	ns
读模式（从 LCM 读数据到 MPU）						
E 周期	T_C	E	1800	-	-	ns
E 脉冲宽度	T_{PW}	E	320	-	-	ns
E 上升/下降时间	T_R，T_F	E	-	-	25	ns
地址建立时间	T_{AS}	RS、RW、E	10	-	-	ns
地址保持时间	T_{AH}	RS、RW、E	20	-	-	ns
数据延迟时间	T_{DDR}	DB0～DB7	-	-	260	ns
数据保持时间	T_H	DB0～DB7	20	-	-	ns

Parallel port mode test waveform



write mode



read mode

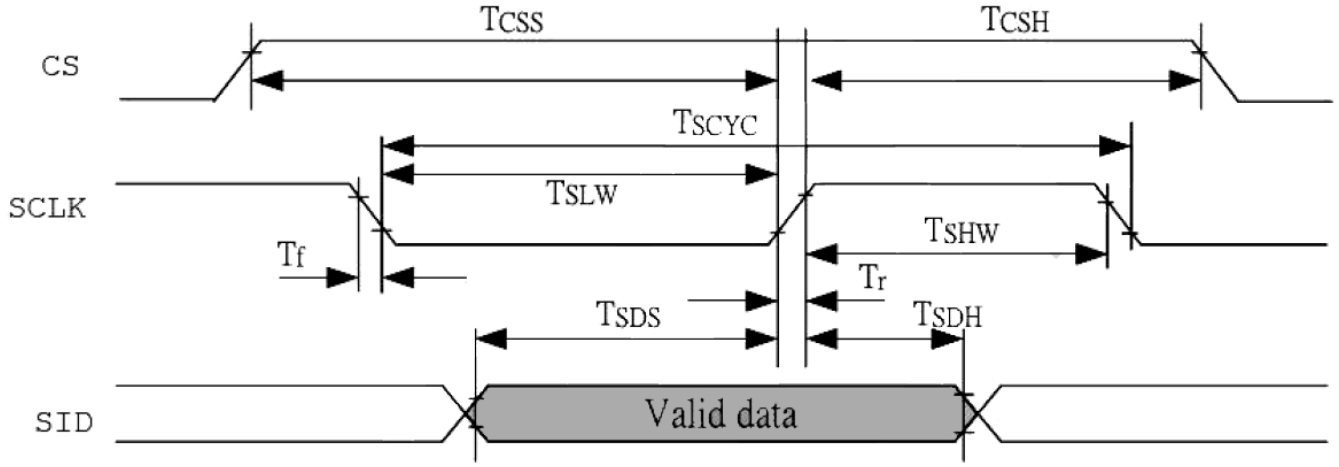
5.3.3AC parameter1(Ta=25oC, Vdd=4.5V~5.5V)serial interface mode

参数	符号	测试条件	最小	典型	最大	单位
SCLK 高脉宽	T_{SHW}	E	200	-	-	ns
SCLK 低脉宽	T_{SLW}	E	200	-	-	ns
SID 数据建立时间	T_{SDS}	RW	40	-	-	ns
SID 数据保持时间	T_{SDH}	RW	40	-	-	ns
CS 建立时间	T_{CSS}	RS	60	-	-	ns
CS 保持时间	T_{CSH}	RS	60	-	-	ns

5.3.4AC parameter2(Ta=25oC, Vdd=2.7V~4.5V)serial interface mode

参数	符号	测试条件	最小	典型	最大	单位
SCLK 高脉宽	T_{SHW}	E	300	-	-	ns
SCLK 低脉宽	T_{SLW}	E	300	-	-	ns
SID 数据建立时间	T_{SDS}	RW	40	-	-	ns
SID 数据保持时间	T_{SDH}	RW	40	-	-	ns
CS 建立时间	T_{CSS}	RS	60	-	-	ns
CS 保持时间	T_{CSH}	RS	60	-	-	ns

Serial port mode test waveform



serial interface mode

6.Interface Description

foot number	symbol	Function	Remark
1	VSS	Negative pole of power supply	
2	VDD	Positive pole of power supply	
3	V0	LCDBias supply	adjustable contrast
4	RS (CS)	Data/Command Select (H:dataL:command) (serial chip enable)	
5	R/W(SID)	Read/Write Select (H:readL:write) (serial data transfer)	
6	E(SCLK)	Enable signal (serial clock transmission)	
7	D0	data bit0	
8	D1	data bit1	
9	D2	data bit2	
10	D3	data bit3	
11	D4	data bit4	
12	D5	data bit5	
13	D6	data bit6	
14	D7	data bit7	
15	PSB	Interface Mode Selection (H:parallel port,L:serial port)	Short to the back of the module Point direction must be consistent
16	NC	empty feet	
17	RST	System reset pin	
18	Vout	LCDDouble voltage output	
19	LEDA	ledbacklight positive	
20	LEDK	ledBacklight negative	

passVoutConnect a resistor toV0adjustable contrast

on the back of the moduleP.S.The shorting point is selected for the serial and parallel port mode. Note that the two shorting points cannot be shorted at the same time, otherwise the module will be short-circuited

7.instruction description

module has4bit/8bit-parallel and serial communication modes via the back of the moduleP.S.The short contact point is selected,Pfor4bit/8bit-parallel communication mode,SFor serial communication mode.

During read and write operations, use the2indivual8bit register, one is data registerDR, and the other is the instruction registerIR . data registerDRas write and readDDRAM/CGRAM/GDRAMas well asIRAMdata, in the access targetRAMWhen the address is selected by the instruction, each time the data register (DR) access actions will automatically use the last selected targetRAMAddress when subject to write or read.

by settingRS/RWVarious operations on bits:

RS	RW	operate
L	L	Write instruction operation (MPUWrite command code toIR)
L	h	Read busy flag (DB7) and address counter (DB0~DB6)
h	L	write data operation (MCUwrite data toDR)
h	h	read data operation (MCUfromDRread data)

7.1busy flag (BF)

BFIf it is high, it means that the internal operation is in progress, so within this time, the next instruction will not be executed. whenRS="0 "andR/W="1"(during read instruction operation),BFThe value can be changed fromDB7port read, when executing the next instruction, must confirmBFnot for"1".

7.2Address Counter (AC)

Address Counter (AC) to storeDDRAM/CGRAM/GDRAMthe address of one, which can be set by setting the instruction register (IR) to change, after which just write or readDDRAM/CGRAM/GDRAMvalue, the address counter (AC) will be automatically increased by one, whenRS="0"andR/W="1", the value in the address counter can be changed fromDB0~DB6read out.

7.3Display DataRAM(DDRAM)

modularDDRAMsupply64x2indivual8BITspace, can control up to4OK16Character(64characters) Chinese character display, when writing display dataRAM, can be displayed separatelyCGROM,HCGROMandCGRAMglyphs, the module can display three glyphs, which are half-widthHCGROMglyph,CGRAMFont and ChineseCGROMFont, the choice of three fonts is determined by theDDRAMThe encoding choice written in the0000H~0006Hchoose from the codeCGRAMcustom glyphs for 02H~7FHThe encoding selects the half-width alphanumeric font, as forA1The above encoding will be automatically combined with the next8BITThe bytes consist of two BYTEEncoding to form the encoding of Chinese charactersBIG5(A140H~D75FH)orGB(A1A0H~F7FFH), the detailed glyph encodings are as follows:

1.Show half-width glyphs: set8BITdata writingDDRAMin the range of02H~7FHencoding.

2.showCGRAMfont: will16BITdata written toDDRAMIn total, there are0000H,0002H,0004H, 0006HFour codes.

3.Display Chinese characters: will16bit data writeDDRAMmiddle

scopeBIG5(A140H~D75FH)orGB(A1A0H~F7FFH),Will16bit data by consecutively writing two8write in bitsDDRAM, write high first8bit, then write low8bit.

7.4Chinese character generationROM(CGROM) and half-width glyphsROM(HCGROM)

glyph generationROMhave8192indivual16x16Chinese glyph images of dots and126indivual16x8An alphanumeric symbol image for points, which uses twoBYTEto provide glyph encoding options, withDDRAMWrite the glyph code to be displayed toDDRAM, the hardware will automatically read fromCGROMThe glyphs to be displayed are displayed on the screen.

7.5glyph generationRAM(CGRAM)

glyph generationRAMIt has the function of custom glyph (word creation), and can provide four groups16x16custom glyph space for dots,

Users can customize glyph images not provided by internal glyphs toCGRAMin, thenCGRAMCustom glyphs in theDDRAMdisplayed on the screen.

CGRAMThe coding of fonts and Chinese fonts can only appear at the starting position of each address counter, as shown in the following table:

80		81		82		83		84		85		86		87		88		89		8A		8B		8C		8D		8E		8F	
h	L	h	L	h	L	h	L	h	L	h	L	h	L	h	L	h	L	h	L	h	L	h	L	h	L	h	L	h	L	h	L
L	C	m	6	4	3	2																									
middle		arts		Character		library		L	C	D.		mold		Group																	

DDRAM/CGRAMAddress map:

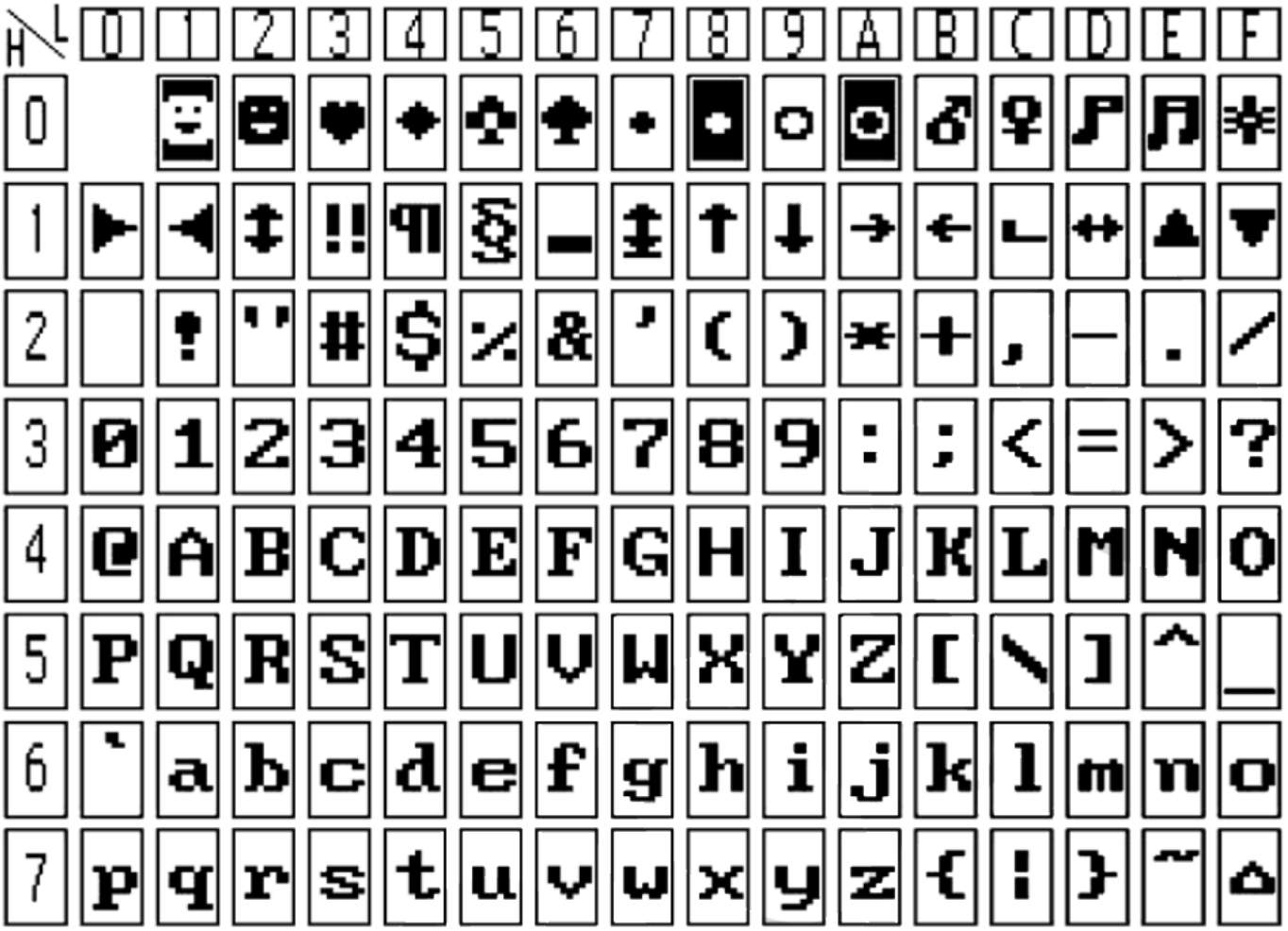
DDRAM 数据					CGRAM 地址		CGRAM 数据高 8 位										CGRAM 数据低 8 位							
B15～B4	B3	B2	B1	B0	B5	B4	B3	B2	B1	B0	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	X	00	X	00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
					0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
					0	0	1	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
					0	0	1	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0
					0	1	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0
					0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
					0	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0
					0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
					1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
					1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
					1	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0
					1	0	1	1	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0
					1	1	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0
					1	1	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
					1	1	1	0	0	0	1	0	0	0	0	1	1	0	1	1	0	0	0	0
					1	1	1	1	0	1	0	0	1	1	1	0	0	0	0	0	1	1	0	0
0	X	01	X	01	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
					0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
					0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0
					0	0	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
					0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
					0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
					0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
					0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
					1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0
					1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0
					1	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0
					1	0	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0
					1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0
					1	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0
					1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0
					1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note:1. DDRAMdata (glyph code)1~2bit andCGRAMaddress of4~5bit sync coincidence (2bit:4group image).

2. CGRAMaddress0~3bits specify the column address of the glyph image, the total instruction16OK(4bit), the16line is the display of the cursor display area, cursor display and16Row data adopts logicalORway to generate display results.

3.The row and column pixels of the displayed image correspond toCGRAMdata0~15bits (15position on the far left).

4.choose toCGRAMimage data,DDRAMdata4~15bit must be set to0, as for0bit and3Bit value can be any value



16x8half-width glyph symbol table

7.6drawingRAM(GDRAM)

drawing displayRAMsupply128x64indivual8bit storage space (set by extended instructions for drawingRAMaddress), can control up to256x64The 2D drawing space of the point, when changing the drawingRAM, set by the extension commandGDRAMaddress, first set the vertical address and then set the horizontal address (write two consecutive8bit data to complete the vertical and horizontal coordinate address setting), and then write two8bit data to plotRAM, while the address counterACOne will be added automatically, and the entire drawing will be writtenRAMThe steps are as follows:

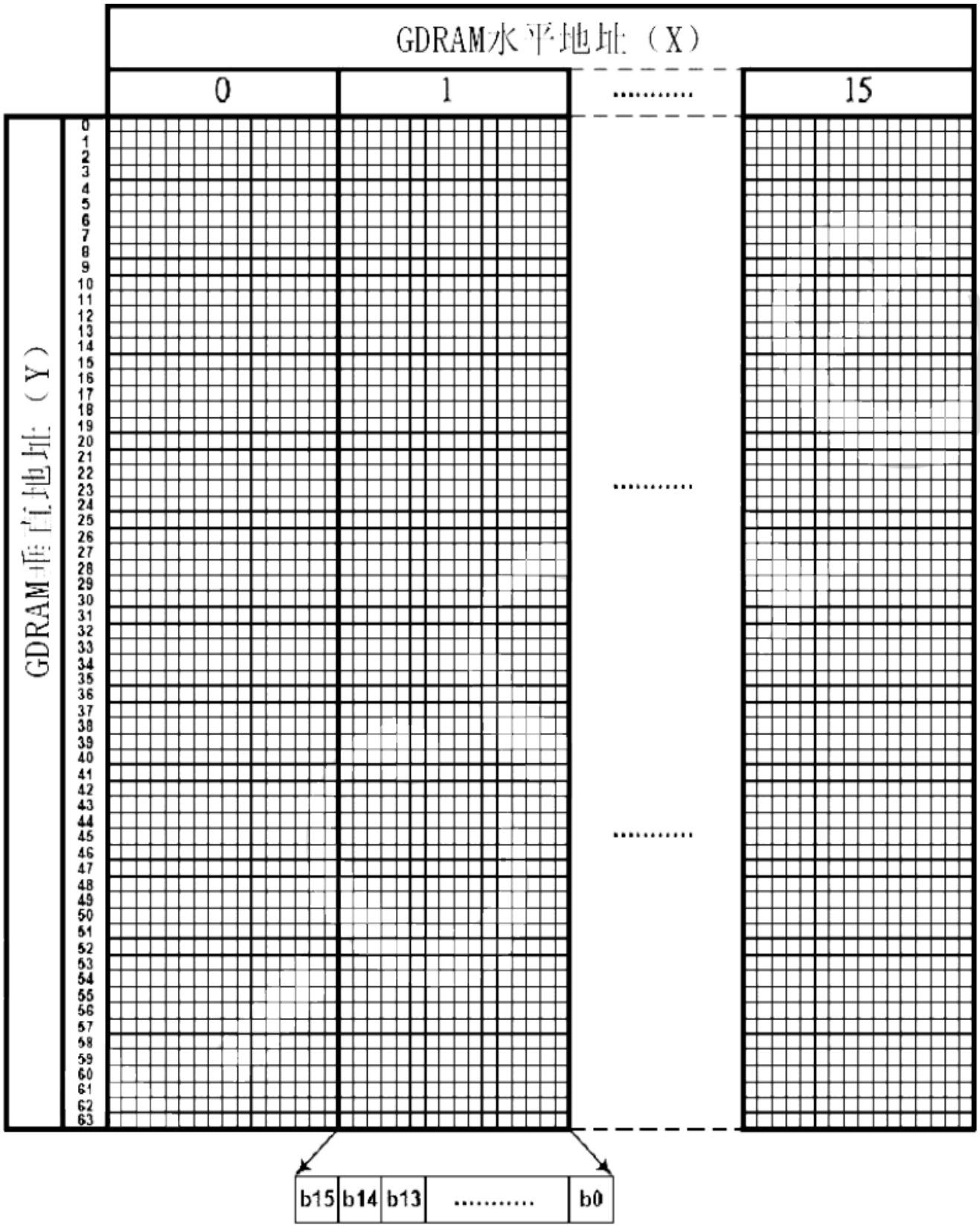
1.First put the vertical8sitting sign (Y) to write the drawingRAMaddress

2.Then show the horizontal position (x) to write the drawingRAMaddress

3.WillD15~D8write toRAMmiddle (write the first data)

4.WillD7~D0write toRAMin (write the second data)

GDRAMCoordinate address and data sequence comparison table



7.6command description

The module provides two sets of control instructions, the basic instructions and extended instructions are as follows:

command table1(RE=0: basic instruction set)

instruction	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	execution time	describe
to clear show	0	0	0	0	0	0	0	0	0	1	1.6ms	Will20Hto writeDDRAM, set the address counter address to00H
address return to zero	0	0	0	0	0	0	0	0	1	-	72us	Set the address counter address to00H,and Return the cursor to its original position, DDRAMcontent remains the same
input mode format settings	0	0	0	0	0	0	0	1	I/D	S	72us	Set the cursor when reading and writing data Direction of movement and specified display movement
show switch	0	0	0	0	0	0	1	D.	C	B	72us	Set display switch, cursor switch, cursor position and whitening control bit
shift control	0	0	0	0	0	1	S/C	R/L	-	-	72us	Set cursor movement, display movement direction control bits,DDRAMdata remains unchanged
Function set up	0	0	0	0	1	DL	-	0 RE	-	-	72us	DL=1:8bit transferDL=0:4bit transfer RE=1: Expansion instructionRE=0: basic command
set up CGRAM address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	72us	set in the address counterCGRAMaddress Need to confirm the extension commandSR=0(scroll address orRAMaddress selection)
set up DDRAM address	0	0	1	0 AC6	AC5	AC4	AC3	AC2	AC1	AC0	72us	set in the address counterDDRAMland site,AC6fixed as0
busy reading sign & address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	0us	read busy flagBF, the address counter in content is read out at the same time
write data	1	0	D7	D6	D5	D4	D3	D2	D1	D0	72us	write data to internalRAM (DDRAM/CGRAM/GDRAM)
read data	1	1	D7	D6	D5	D4	D3	D2	D1	D0	72us	from the insideRAM (DDRAM/CGRAM/ GDRAM)read data from

Note: "-" is not considered

command table2(RE=1: extended instruction set)

instruction	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	execution time	describe
standby model	0	0	0	0	0	0	0	0	0	1	72ms	Enter standby mode, perform any other command order to terminate standby mode
scrollingly address or RAMland address selection	0	0	0	0	0	0	0	0	1	SR	72us	SR=1: Allows input of vertical scrolling addresses SR=0: Allow settingCGRAMaddress (basic instruction)
Anti-White choose	0	0	0	0	0	0	0	1	R1	R0	72us	choose4Any line in the line is displayed in reverse video, R1, R0initial value00, perform the first reverse display, and then perform a normal display
expand Function set up	0	0	0	0	1	DL	-	1 RE	G	0	72us	DL=1:8bit transferDL=0:4bit transfer RE=1: Expansion instructionRE=0: basic command G=1:drawing display onG=0:drawing display off
set up IRAM address or to scroll address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	72us	SR=1:AC5~AC0address for vertical scrolling
set up GDRAM address	0	0	1	0 0	AC5 0	AC4 0	AC3 AC3	AC2 AC2	AC1 AC1	AC0 AC0	72us	set in the address counterGDRAMland address, first set the vertical address and then set the horizontal address vertical address rangeAC5~AC0 horizontal address rangeAC3~AC0

Note: 1.Before the module accepts the command, it must be confirmed that the module is not busy, that isBFrequired for0, to accept new instructions.

2.RESelected for basic instructions and extended instructions, when changingREAfter that, the following instructions are subject to the last state, unless changes.

7.6.1clear display

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

by writing20H(space code) to allDDRAMaddress and set the address counter to00H,data can be cleared , place the cursor at the initial position, and set the input mode to increment (I/Dis high).

7.6.2address zero

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	-

"-"for not considering

The instruction is to return the cursor to the initial position, and toDDRAMThe address is set to00HWrite to the address counter and change the display to the initial state,DDRAMThe data in remains unchanged.

7.6.3input mode

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

Sets the direction of movement of the cursor and display

I/D:DDRAMIncrement or decrement of address (cursor or blinking)

whenI/Dfor1, the cursor flickers and moves to the right,DDRAMaddress is incremented; whenI/Dfor0, the cursor flickers and moves to the left,DDRAM The address is decremented.

S: display shift

whenSfor0when, yesDDRAMorCGRAMDuring read and write operations, the entire display will not shift. whenSfor1when, yesDDRAMwrite operation, the entire display will be shifted according to theI/DSet direction displacement.

7.6.4Display switch control

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D.	C	B

Display/cursor and blinking control

D.: display switch control bit

whenD.for1When , the display is on; whenD.for0, the display is off, butDDRAMThe displayed data remains unchanged. C: Cursor switch control bit

whenCfor1When , the cursor is on; whenCfor0, the cursor disappears, butI/DA register holds its data. B: Cursor position highlight control bit

whenBfor1When , the cursor position will be highlighted, and the data where the cursor is located will be displayed in reverse; whenBfor0, the cursor position is highlighted.

7.6.5shift control

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	-	-

Set the cursor movement and display shift control bits, this command does not changeDDRAMcontent

S/C	R/L	operate
0	0	Cursor moves left, address counter decrements1
0	1	Cursor moves to the right, address counter increments1
1	0	So the display moves to the left, the cursor moves along with the display, and the address counter remains unchanged
1	1	All displays move to the right, the cursor moves along with the display, and the address counter remains unchanged

7.6.6function settings

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	-	RE	-	-

DL: Interface data width control bit

whenDLfor1when, means8bit bus connected toMPU

whenDLfor0when, means4bit bus connected toMPU, when as4bit bus mode,8bit data needs to be transmitted via the4bit data2

completed.

RE: Instruction set selection control bit

whenREfor1When , it is an extended instruction set; whenREfor0, it is the basic instruction set.

7.6.7set upCGRAMaddress

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

WillCGRAMThe address is placed into the address counter.

ACRange is00H~3FH

Need to confirm the extension commandSR=0(scroll address andRAMaddress selection)

7.6.8set upDDRAMaddress

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0

WillDDRAMThe address is placed into the address counter.

first rowACRange is80H~8FH second

lineACRange is90H~9FH The third row

ACRange isA0H~AFH fourth rowAC

Range isB0H~BFH Note: Only two of

the four lines are displayed.

7.6.9Read busy flag and address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0

This command shows whether the module is working internally, ifBFfor1, internal work is in progress, need to wait untilBFbe placed0, the next instruction can be executed. In this instruction, the value in the address counter can also be read.

7.6.10write data toRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

to write8bit data toRAM, when written will make (AC) changes, eachRAMaddress(CGRAM,DDRAM…) are continuously written to two8Bit data, the address counter is automatically incremented by one after the second data operation is written.

7.6.11fromRAMRead data

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

fromRAMRead data in, when read will make (AC)Change.

After the current set address command (CGRAM,DDRAM…), if you want to read data, you need to firstDUMMY READThe correct data will be read only once, and it is not required for the second readDUMMY READ, unless the command to set the address is issued again DUMMY READ.

Explanation of extended instruction set

7.6.12standby mode

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

Enter standby mode, execution of any other command can terminate the standby mode; this command does not changeRAMcontent.

7.6.13scroll address orRAMaddress selection

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	SR

whenSR=1, allows entering vertical scrolling address

whenSR=0, allows settingCGRAMaddress (basic command)

7.6.14Negative selection

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	R1	R0

first choice4Any line in the row can be highlighted, and it can be decided whether to highlight or not.

R1,R0The initial value is00, when it is set for the first time, it will be displayed in reverse video, and when it is set again, it will be displayed normally.

R1	R0	illustrate
L	L	The first line is highlighted or displayed normally
L	h	The second line is highlighted or displayed normally
h	L	The third line is highlighted or displayed normally
h	h	The fourth line is highlighted or displayed normally

Note: Only two of the four lines are displayed.

7.6.15Extended function settings

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	-	RE	G	-

DL: Interface width control bit

whenDL=1,for8bitMPUcontrol interface when

DL=0,for4bitMPUcontrol interface RE:

Instruction set selection control bit

whenRE=1, for the extended instruction set

whenRE=0, for the basic instruction set

G: Drawing display control bit

whenG=1, the plot displays on

whenG=0, plot display off

The actions of the same instruction cannot be changed at the same timeREandDL,G, need to change firstDLorGchange laterREto ensure correct setting.

7.6.16set scroll address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

SR=1:AC5~AC1address for vertical scrolling

7.6.17set drawingGDRAMaddress

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	AC5	AC4	AC3	AC2	AC1	AC0

set upGDRAMaddress to address counter (AC) First set the vertical address and then the horizontal address (write two consecutive8bit

data to complete the vertical and horizontal coordinate address) vertical address rangeAC5~AC0

horizontal address rangeAC3~AC0

drawingRAMThe address counter will only automatically add one to the horizontal address, when the horizontal address is equal to0FH,will be set to00H, the vertical address will not automatically increase by one, so when writing data continuously, the program needs to judge whether the vertical address needs to be reset.

8.Operation Timing

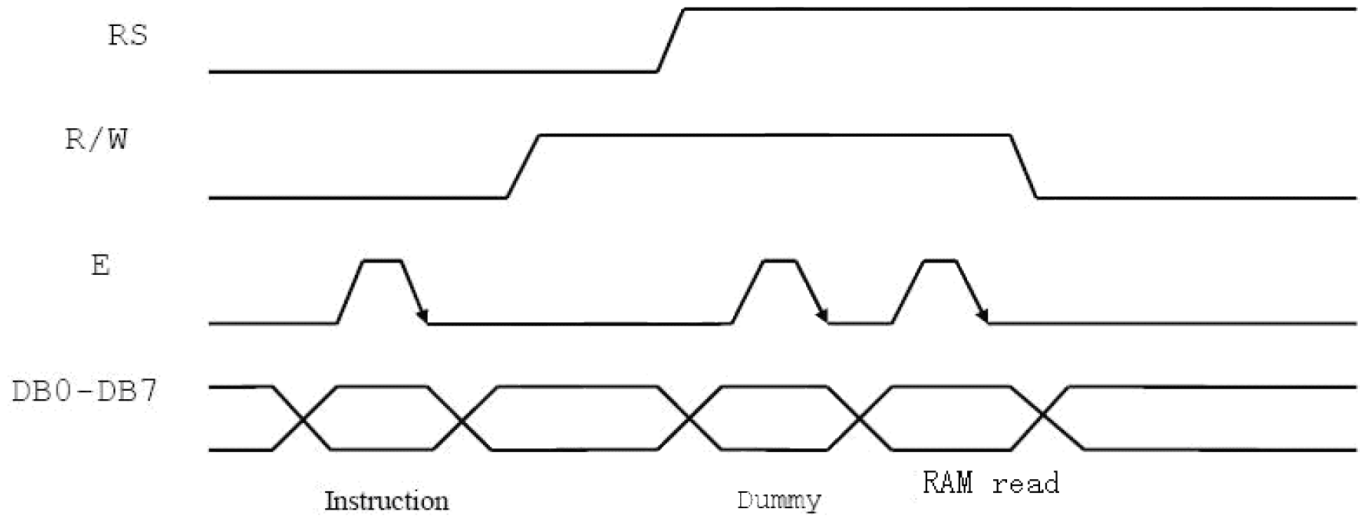
8.1Parallel Data Transfer Interface

When the interface selects parallel transmission mode, it can be changed by commandDLto choose8bit or4bit parallel interface, the main control system will be controlled by (RS,RW,E.,DB0~DB7) to complete the data transfer.

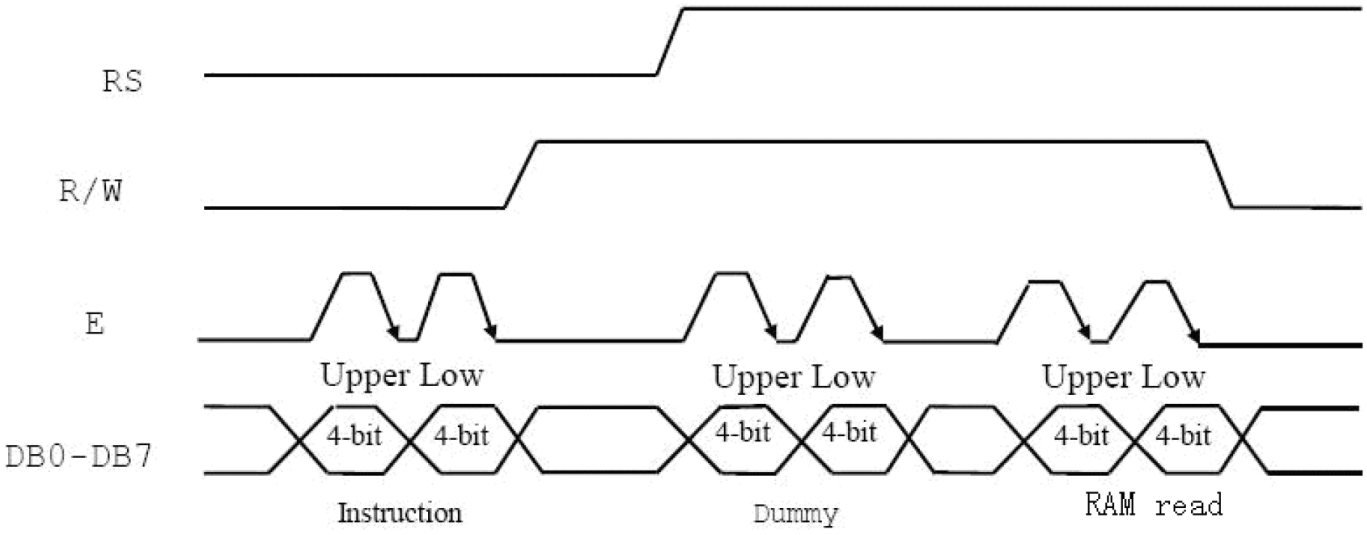
Looking at a complete process, after setting the address command (CGRAM,DGRAM,IRAM…), if you want to read data, you need to firstDUMMY READThe correct data will be read only once, and it is not necessary to read the second timeDUMMY READ, unless the reset address command is required againDUMMY READ.

exist4bit-parallel transfer mode, each8bit instruction or data will be split into two8bit data, high4bit is placed first8 bit data high4bit, low4bit is placed in the second8bit data high4bit, two8bit data low4bits are not considered.

The timing is shown in the figure below:



8Bit Parallel Bus Mode Timing Diagram



4Bit Parallel Bus Mode Timing Diagram

8.2Serial data transfer interface

When the interface is selected as the serial transmission mode, two data transmission lines will be used as the transmission of serial data, and the main control system will cooperate with the transmission of the synchronous clock terminal (SCLK) with the serial data receiver (SID) to complete the serial transmission action.

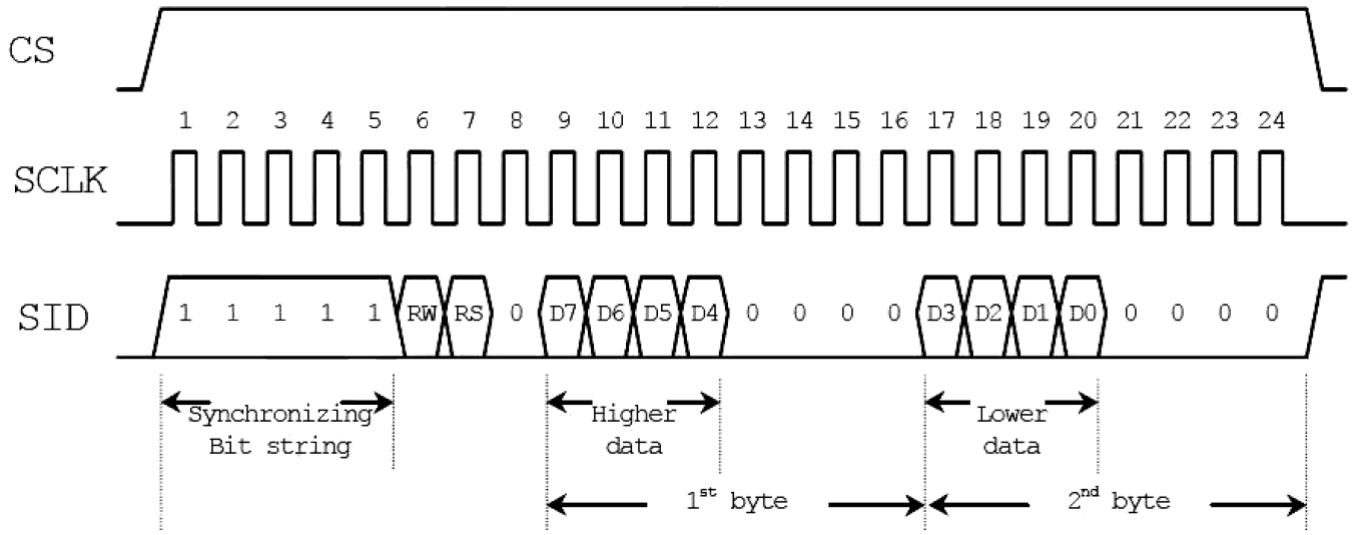
When the main control system connects multiple modules, the chip selection pin (CS) will be used in conjunction with the chip select pin (CS) is set to a high level, the serial transmission data will be received, when the chip selection pin (CS) is low level, the serial transmission count and serial data inside the module will be reset, that is to say, in this state, the data in transmission will be terminated and cleared, and the serial data count to be transmitted will be reset Back to the first point, in the case of a system that only uses one module, only the synchronous clock and serial data transmission pins need to be used, and the chip selection pin is fixed to be connected to a high level.

The command execution time needs to be considered when transmitting data serially, and the next command must not be transmitted until the previous command is completely executed, because there is no serial send/receive buffer inside the module.

From the point of view of a complete serial transmission process, the first transmission8Bit start bit, the module needs to receive5indivual1(sync code), At this time, the transmission count will be reset and the serial transmission will be synchronized, and the following two bits will be set as the transmission direction bit (RW) and register select bits (RS), the last8Bit is0.

After receiving the synchronization code andRWandRSdata8After the start code, each8bit instructions will be split into two 8bit data transfer, high4bit is placed first8bitLSBpart low4bit is placed in the second8bitLSBpart, other4bit is0.

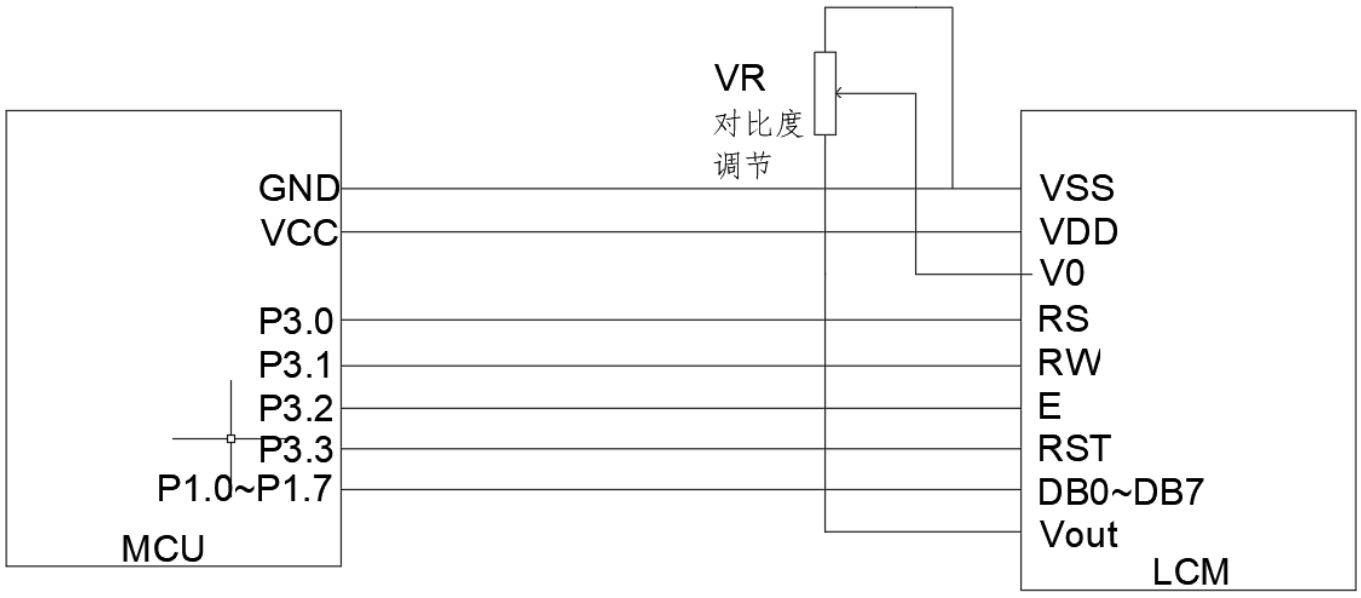
The timing is shown in the figure below:



Serial mode data transmission timing diagram

9.application routine

9.1Parallel Transmission Wiring Diagram



并行传输接线方法

9.2parallelC51routine

```
# include <STC15.H>
```

```
# include <string.h>
# include <INTRINS.H>
#define uchar unsigned char
#define uint unsigned int
#define db P1
sbit rs=P3^0;
sbit rw=P3^1;
sbit e=P3^2;
sbit rst=P3^3;
 uchar code IC_DAT1[]={
"0123456789:;<=>?"
"PQRSTUVWXYZ[/]^_"                    //0x80
"@ABCDEFGHIJKLMNO"
"'abcdefghijklmno" };
uchar code IC_DAT2[]={
"Hunan Risun Display Technology"      //0x88
"Shenzhen Yaoxingyang Industry"       //0x98
"Hunan Zhijiang Co., Ltd."
"Limited Professional Mods"};
////////////////////////////////////////////////////////// ////////////////////// //
BitmapLattice Data Sheet                                  //
//picture:D:\..12864.bmp,Horizontal modulo left high bit, data arrangement: from left to right from top to  //
bottom //size of the picture:128*64                       //
////////////////////////////////////////////////////////// //////////////////////
unsigned char code nBitmapDot[] =                 //data sheet
{
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0xFB,0xC0,0xCF,0xF3,0xE0,0xDF,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7F,
    0xFB,0xC0,0xCF,0xF3,0xE0,0xDF,0xFE,0x00,0x01,0x20,0x84,0x6C,0x08,0x40,0x00,0x60,0x1B,0xE1,0xF7,0x8F,0x07,0xD8,0x06,0x5F,0xF1,
    0x24,0x84,0xB4,0x44,0x47,0xBF,0x6F,0xDB,0xFF,0xF3,0x8F,0x0F,0xDB,0xF6,0x32,0x91,0x24,0x95,0x6C,0x24,0x84,0xA1,0x6F,0xD9,0xFE,
    0x3D,0x8C,0x04,0x1B,0xF6,0x04,0x43,0xA4,0x8E,0xB4,0x20,0x84,0xA1,0x6F,0xDB,0xFE,0xBD,0xC8,0xBE,0x1B,0xF6,0x09,0x21,0x24,0x84,
    0x48,0x01,0x05,0x21,0x6F,0xD8,0xFF,0xFF,0xE1,0xFF,0x1B,0xF6,0x41,0x01,0x24,0x9F,0x7C,0x00,0x05,0x3F,0x6F,0xD8,0x71,0xE1,0x80,
    0x8E,0x1B,0xF6,0x2F,0xF1,0x24,0x8A,0xD0,0xFF,0xE4,0xA1,0x6F,0xD8,0x31,0xC1,0x80,0x04,0x1B,0xF6,0x01,0x01,0x24,0x8A,0x7C,0x00,
    0x04,0xA1,0x60,0x1B,0x1E,0x01,0xF0,0x7B,0xD8,0x06,0x03,0x81,0xA4,0x8A,0x50,0x11,0x04,0xA1,0x7F,0xFB,0x0E,0x49,0xFA,0x77,0xDF,0xFE,0x25,0x43,0x2
    0x8B,0x50,0x40,0x44,0x3F,0x00,0x03,0x9E,0xF2,0x6E,0x60,0xC0,0x00,0x01,0x00,0x80,0x92,0x7C,0x80,0x24,0x21,0x00,0x03,0x0E,0xF6,
    0x6F,0x60,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0xDB,0xD9,0xFD,0x80,0x7C,0x3E,0x3E,0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x6F,0xD9,0xC9,0xFF,0x81,0x7F,0x7E,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x60,0x60,0xC0,0x7E,0x33,0x99,
    0xDE,0xF6, 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x19,0x99,0xE9,0x93,0x1F,0x1E,0x76,0xC0,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0xDB,0xD9,0xFD,0x80,0x7C,0x3E,0x3E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x6F,0xD9,0xC9,0xFF,0x81,0x7F,0x7E,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x60,0x60,0xC0,0x7E,0x33,0x99,0xDE,0xF6,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x19,0x99,0xE9,0x93,0x1F,0x1E,0x76,0xC0,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0xDB,0xD9,0xFD,0x80,0x7C,0x3E,0x3E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x6F,0xD9,0xC9,0xFF,0x81,0x7F,0x7E,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x60,0x60,0xC0,0x7E,0x33,0x99,0xDE,0xF6,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x19,0x99,0xE9,0x93,0x1F,0x1E,0x76,
```

0x02,0x00,0x44,0x01,0x00,0xF7,0xC0,0x88,0x0F,0x1B,0x09,0xCD,0xBB,0x1F,0x1C,0xFE,

0x7F,0xF0,0x44,0x01,0x00,0x94,0x40,0x88,0x7F,0x60,0xF6,0x00,0x4F,0x07,0xDF,0xCE,0x40,0x10,0x44,0x1F,0xF
0x61,0xE6,0x10,0xEF,0x1F,0xDF,0xFE,0x09,0x00,0x44,0x02,0x00,0xA7,0xC1,0x04,0x73,0x9B,0x01,0xB1,0xF0,0x
0x44,0x87,0xF8,0xA4,0x41,0x24,0x03,0xF1,0x87,0x3D,0xF1,0x9C,0x74,0x30,0x11,0x01,0x45,0x0C,0x08,0x94,0x4
0x08,0x81,0x9C,0x26,0xF0,0x09,0x00,0xC6,0x17,0xF8,0x97,0xC4,0x41,0x6F,0xDF,0xC9,0x80,0x00,0x9F,0x1B,0x
0x08,0x95,0x20,0x40,0x6F,0x9F,0xD9,0x25,0x10,0xDF,0x19,0xDE,0x02,0x00,0x44,0x07,0xF8,0x95,0x40,0x88,0x
0xE4,0x78,0x08,0x04,0xC0,0x44,0x04,0x08,0xE4,0x81,0x04,0x50,0x30,0xC6,0x00,0x00,0x7F,0x39,0x3E,0x18,0x2
0x43,0xFE,0x00,0x78,0xC6,0x35,0x60,0x7F,0x7B,0x3E,0x60,0x13,0xFF,0x84,0x18,0x84,0x20,0x02,0x1F,0xC4,0x0
0x00,0x00,0x00,0x00,0x00,0x6C,0xDE,0x20,0x00,0x80,0xDF,0x18,0x1E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0xC0,0x7D,0x01,0x0E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x63,0x23,0xCC,0x09,0x40,0x60,0xC3,0x86,0x
0x08,0x00,0x1E,0x78,0x78,0xDF,0x00,0x80,0xE6,0x7F,0xF8,0x3F,0xE0,0x10,0x01,0x40,0x04,0x03,0xE0,0x3C,0x7
0x7F,0xFC,
0x00,0x20,0x10,0x01,0x40,0xFF,0xE0,0x40,0x04,0xE0,0xC3,0x01,0xF2,0x78,0x26,0xFE,0x7F,0xA1,0xFF,0x0F,0xF
0x84,0x0C,0xF0,0x08,0x03,0xF6,0x7F,0x74,0x7E,0x00,0x21,0x11,0x09,0x48,0x20,0x81,0xF8,0x01,0xF8,0x18,0x0
0x3F,0x21,0x11,0x09,0x48,0x11,0x00,0x30,0x7C,0xE3,0x3E,0x00,0x21,0xE0,0x1E,0x1C,0x21,0x21,0x11,0x09,0x4
0xFF,0x6E,0x78,0x36,0x21,0x21,0x11,0x0C,0x08,0x04,0x00,0x21,0x7F,0x3B,0xF9,0x3F,0xFE,0x65,0x58,0x26,0x3
0x0A,0x01,0x24,0x67,0x61,0xF9,0xB3,0xCF,0x63,0xDE,0x06,0x20,0x20,0x10,0x0F,0xF8,0x31,0x82,0x22,0x60,0x9
0x18,0x0E,
0x01,0xC0,0x10,0x08,0x08,0xC0,0x64,0x61,0x61,0xD9,0x76,0x06,0x60,0x07,0x18,0x1E,0x00,0x00,0x00,0x00,0x0
0x00,0x63,0x03,0xC0,0xC8,0x73,0x8F,0xE1,0x88,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x62,0x0B,0x80,0xF
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x60,0x1F,0x00,0xF7,0xCD,0x98,0xFF,0x18,0x00,0x00,0x00,0x00,0x0
0x07,0xF8,0x30,0x70,0xF8,0xC3,0x8C,0x00,0x10,0x10,0x00,0x78,0x04,0x00,0x20,0x00,0x01,0xF8,0x30,0x79,0xF
0xE0, 0x88, 0x7F, 0xF8, 0x37, 0xB7, 0xEE, 0xEF, 0xDB, 0xCE, 0x10, 0x90, 0x82, 0x02, 0x10, 0x20,
0x81, 0x04, 0x60, 0x18, 0x61, 0xF3, 0xF0, 0xE4, 0xC3, 0x00, 0x1E, 0x93, 0x01, 0x87, 0xE0, 0x11,
0x06, 0x03, 0x6F, 0xDB, 0x19, 0xFE, 0x6F, 0xFD, 0xFF, 0x38, 0x22, 0x90, 0xFE, 0x00, 0xC0, 0x0E,
0x01, 0xFC,
0x6F,0xDB,0x19,0xFE,0x6F,0xFB,0x7F,0xBC,0x52,0x90,0x10,0x03,0x08,0x31,0x80,0x20,0x6F,0xD9,0x8E,0xC1,0x
0x90,0x10,0x0F,0xFC,0xD1,0x60,0x20,0x6F,0xDB,0x1E,0x49,0x99,0x9B,0x1E,0x42,0x04,0x91,0xFF,0x00,0x84,0x
0x3E,0xCC,0x00,0x7C,0x66,0x36,0x08,0x10,0x10,0x04,0x90,0x11,0x00,0x20,0x6F,0xD9,0x35,0xBB,0x9D,0xEF,0x
0x08,0x88,0x21,0x00,0x20,0x60,0x18,0x01,0xB7,0xBF,0xE7,0xF8,0x06,0x20,0x73,0xFF,0x91,0x84,0x41,0x07,0xF
0xFE,

```
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
};
void delay(unsigned int m)              //Delay program
            {
              unsigned int i,j;
              for(i=0;i<m;i++)
                for(j=0;j<20;j++);
            }

void delayms(unsigned int n)            //time delay10xnoms program
            {
              unsigned int i,j;
              for(i=0;i<n;i++)
               {
                for(j=0;j<1000;j++);
               }
            }

/************************************************** ******************
* name:TransferData()
*  Function: Write command or data
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* / void TransferData(uchar data1,bit DI)
{
          rw=0;
          rs=DI;
          //delay(1);
          db=data1;
          e=1;
          delay(1);
          e=0;
              delay(1);
}
void initinal(void)          //LCDFont initializer
{
            rst=0;
            delay(10);
            rst=1;
         delayms(5);             //more than the40MSdelay program
       TransferData(0x30,0); //Extended Function Set :8BITset up,RE=0: basic //instruction set, G=0
                                  :graphic display OFF
         delay(10);            //more than the100uSdelay program
         TransferData(0x30,0); //Function Set
         delay(5);             ////more than the37uSdelay program
```

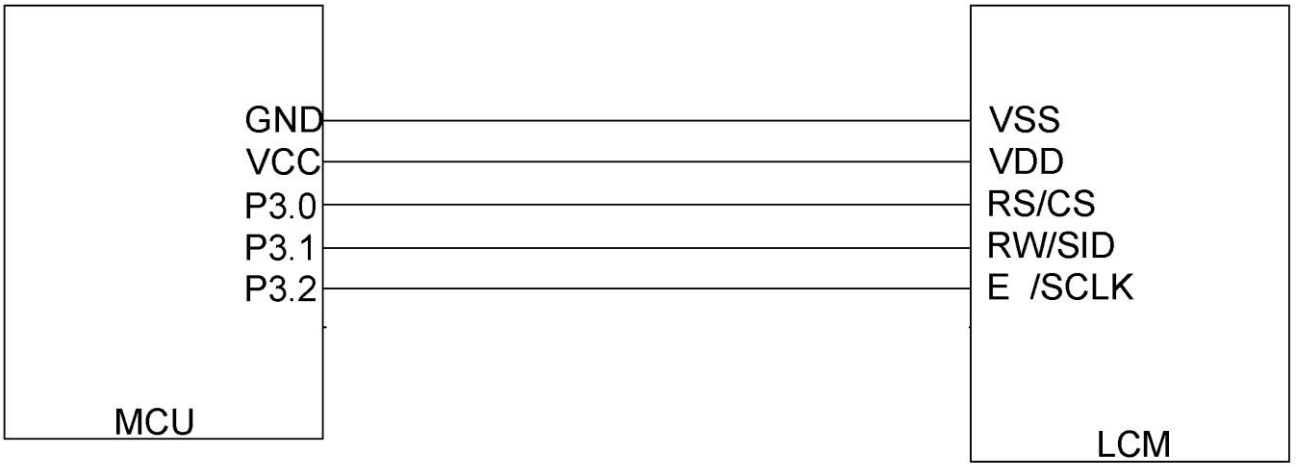
```
        TransferData(0x03,0); //Display on Control
        delay(10);            //more than the100uSdelay program
        TransferData(0x0C,0); //Display Control,D=1,show on
        delay(10);            //more than the100uSdelay program
        TransferData(0x01,0); //Display Clear
        delay(2);             //more than the10mSdelay program
        TransferData(0x06,0); //Enry Mode Set,Cursor plus from right to left1bit shift
        delay(10);            //more than the100uSdelay program
       }
/************************************************** ******************
* name:disp_dat(data1,data2)
* Function: display graph data
* Input data1,data2
* output: none
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* / void disp_dat(unsigned char dat1, unsigned char dat2)
{
  int i,j;
 TransferData(0x36,0);
  for(i=0;i<16;i++)              //
   {
    TransferData((0x80 + (i*2)),0); //SETvertical addressVERTICAL ADD
    TransferData(0x80,0); //SEThorizontal addressHORIZONTAL ADD
    for(j=0;j<16;j++)
     {
      TransferData(dat1,1);
     }
    TransferData((0x80 + (i*2)+1),0); //SETvertical addressVERTICAL ADD
    TransferData(0x80,0); //SEThorizontal addressHORIZONTAL ADD
                for(j=0;j<16;j++)
     {
                        TransferData(dat2,1);
     }
   }
for(i=0;i<16;i++)              //
   {
    TransferData((0x80 + (i*2)),0); //SETvertical addressVERTICAL ADD
    TransferData(0x88,0); //SEThorizontal addressHORIZONTAL ADD
    for(j=0;j<16;j++)
     {
      TransferData(dat1,1);
     }
    TransferData((0x80 + (i*2)+1),0); //SETvertical addressVERTICAL ADD
```

```
    TransferData(0x88,0);          //SEThorizontal addressHORIZONTAL ADD
                for(j=0;j<16;j++)
    {
                    TransferData(dat2,1);
    }
     }
}

void disp_bmp(uchar code *bmpadd)
{ uchar i,j;
    uint k=0;
  TransferData(0x36,0);
   for(i=0;i<32;i++)               //
   {
    TransferData((0x80 + i),0); //SETvertical addressVERTICAL ADD
    TransferData(0x80,0); //SEThorizontal addressHORIZONTAL ADD
    for(j=0;j<16;j++)
    {
      TransferData(bmpadd[k],1);
                    k++;
    }
                }
   for(i=0;i<32;i++)               //
   {
    TransferData((0x80 + i),0); //SETvertical addressVERTICAL ADD
    TransferData(0x88,0); //SEThorizontal addressHORIZONTAL ADD
    for(j=0;j<16;j++)
    {
      TransferData(bmpadd[k],1);
                    k++;
    }
                }

    }
/************************************************** ******************
* name:LCD_MESG()
* Function: display text function
* input: none
* output: none
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* / void lcd_mesg(unsigned char code *adder1)
{
    unsigned char i;
    TransferData(0x30,0);
```

```
    TransferData(0x30,0);
     TransferData(0x80,0); //Set Graphic Display RAM Address
    for(i=0;i<32;i++)
      {
            TransferData(*adder1,1);
            adder1++;
      }
    TransferData(0x30,0);
    TransferData(0x30,0);
     TransferData(0x90,0); //Set Graphic Display RAM Address
    for(i=0;i<32;i++)
      {
            TransferData(*adder1,1);
            adder1++;
      }
}

////////////////////////////////////////////////////// //////
void bmpclear()
{
uchar i,j;
    TransferData(0x34,0);

    for(i=0;i<32;i++)
    {
    TransferData(0x80+i,0);
    TransferData(0x80,0);
      for(j=0;j<32;j++)
      TransferData(0,1);
    }
}

/************************************************** ******************
* name:Main()
* Function: main function
* input: none
* output: none
************************************************************************/
void    main(void)
   {
    initial();          //transferLCDShow image (extended) initializer
       while(1)
       {
        TransferData(0x30,0);
            TransferData(0x01,0);//txtclear();
```

```
                    delayms(2);
      disp_dat(0xff,0xff);
      delayms(1000);
      disp_dat(0xaa,0xaa); //Seperate Rows I
      delayms(1000);
      disp_dat(0xff,0x00); //Seperate Columns II
      delayms(1000);
      disp_dat(0xaa,0x55); //Seperate Dots I
      delayms(1000);
      disp_dat(0x55,0xaa); //Seperate Dots I
      delayms(1000);
      //txtclear();
      bmpclear();
  lcd_mesg(IC_DAT1);        //show western
  delayms(1200);
      TransferData(0x30,0);
            TransferData(0x01,0);//txtclear();
                    delayms(2);
  disp_bmp(bmp1);
  delayms(1200);
      bmpclear();
      TransferData(0x34,0);
  TransferData(0x04,0);
  lcd_mesg(IC_DAT2); //Display Chinese characters
  delayms(1200);

      TransferData(0x34,0);
  TransferData(0x04,0);
     }
   }
```

9.3Serial port wiring diagram



串口接线方法

9.4serial portC51routine

```
# include <STC15.H>
# include <string.h>
# include <INTRINS.H>
#define uchar unsigned char
#define uint unsigned int
#define db P1
sbit rs=P3^0;
sbit rw=P3^1;
sbit e=P3^2;
sbit rst=P3^3;
uchar code IC_DAT1[]={
"0123456789:;<=>?"
"PQRSTUVWXYZ[/]^_"                    //0x80
"@ABCDEFGHIJKLMNO"
"'abcdefghijklmno" };
uchar code IC_DAT2[]={
"Hunan Risun Display Technology"      //0x88
"Shenzhen Yaoxingyang Industry"       //0x98
"Hunan Zhijiang Co., Ltd."
"Limited Professional Mods"};
////////////////////////////////////////////////////////////////////////////
```

```
// BitmapLattice Data Sheet                                    //
//picture:D:\..12864.bmp,Horizontal modulo left high bit, data arrangement: from left to right from top to    //
bottom //size of the picture:128*64                            //
////////////////////////////////////////////////////// ///////////////////////
unsigned char code nBitmapDot[] =                    //data sheet
{

    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0xFB,0xC0,0xCF,0xF3,0xE0,0xDF,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0xFB,0xC0,0xCF,0xF3,0xE0,0xDF,0xFE,0x00,0x01,0x20,0x84,0x6C,0x08,0x40,0x00,0x60,0x1B,0xE1,0xF7,0x8F,0x07,0xD8,0x06,0x5F
    0x24,0x84,0xB4,0x44,0x47,0xBF,0x6F,0xDB,0xFF,0xF3,0x8F,0x0F,0xDB,0xF6,0x32,0x91,0x24,0x95,0x6C,0x24,0x84,0xA1,0x6F,0xD9
    0x3D,0x8C,0x04,0x1B,0xF6,0x04,0x43,0xA4,0x8E,0xB4,0x20,0x84,0xA1,0x6F,0xDB,0xFE,0xBD,0xC8,0xBE,0x1B,0xF6,0x09,0x21,0x2
    0x48,0x01,0x05,0x21,0x6F,0xD8,0xFF,0xFF,0xE1,0xFF,0x1B,0xF6,0x41,0x01,0x24,0x9F,0x7C,0x00,0x05,0x3F,0x6F,0xD8,0x71,0xE1,
    0x8E,0x1B,0xF6,0x2F,0xF1,0x24,0x8A,0xD0,0xFF,0xE4,0xA1,0x6F,0xD8,0x31,0xC1,0x80,0x04,0x1B,0xF6,0x01,0x01,0x24,0x8A,0x7C
    0x04,0xA1,0x60,0x1B,0x1E,0x01,0xF0,0x7B,0xD8,0x06,0x03,0x81,0xA4,0x8A,0x50,0x11,0x04,0xA1,0x7F,0xFB,0x0E,0x49,0xFA,0x77
    0x8B,0x50,0x40,0x44,0x3F,0x00,0x03,0x9E,0xF2,0x6E,0x60,0xC0,0x00,0x01,0x00,0x80,0x92,0x7C,0x80,0x24,0x21,0x00,0x03,0x0E,
    0x6F,0x60,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0xDB,0xD9,0xFD,0x80,0x7C,0x3E,0x3E,0x00,0x00,0x00,0x00
    0x00,0x00,0x00,0x6F,0xD9,0xC9,0xFF,0x81,0x7F,0x7E,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x60,0x60,0xC0,0x7E,0x33,
    0xDE,0xF6,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x19,0x99,0xE9,0x93,0x1F,0x1E,0x76,0x02,0x00,0x44,0x01,0x00,0xF7,0xC0,
    0x88,0x0F,0x1B,0x09,0xCD,0xBB,0x1F,0x1C,0xFE,0x7F,0xF0,0x44,0x01,0x00,0x94,0x40,0x88,0x7F,0x60,0xF6,0x00,0x4F,0x07,0xDF
    0x40,0x10,0x44,0x1F,0xFC,0x94,0x40,0x88,0x7F,0x61,0xE6,0x10,0xEF,0x1F,0xDF,0xFE,0x09,0x00,0x44,0x02,0x00,0xA7,0xC1,0x04,
    0xF1,0x9C,0x74,0x30,0x11,0x01,0x45,0x0C,0x08,0x94,0x42,0x22,0x83,0xE3,0xC6,0x08,0x81,0x9C,0x26,0xF0,0x09,0x00,0xC6,0x17,
    0x97,0xC4,0x41,0x6F,0xDF,0xC9,0x80,0x00,0x9F,0x1B,0xCE,0x7F,0xF0,0x44,0x04,0x08,0x95,0x20,0x40,0x6F,0x9F,0xD9,0x25,0x10
    0x19,0xDE,
    0x02,0x00,0x44,0x07,0xF8,0x95,0x40,0x88,0x70,0x00,0xC0,0x12,0x96,0xE4,0x78,0x08,0x04,0xC0,0x44,0x04,0x08,0xE4,0x81,
    0x04,0x50,0x30,0xC6,0x00,0x00,0x7F,0x39,0x3E,0x18,0x20,0x44,0x04,0x08,0x86,0x43,0xFE,0x00,0x78,0xC6,0x35,0x60,0x7F,0x7B,
    0x60,0x13,0xFF,0x84,0x18,0x84,0x20,0x02,0x1F,0xC4,0x06,0x08,0x10,0xC0,0xE6,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x8E,0x06,0x00,0x0C,0x80,0xF6,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x6C,0xDE,0x20,0x00,0x80,0xDF,0x18,0x1E,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x63,0x63,0xF0,0x00,0xC0,0x7D,0x01,0x0E,0x00,0x00,0x00,0x00,0x00,0x00
    0x00,0x00,0x63,0x23,0xCC,0x09,0x40,0x60,0xC3,0x86,0x00,0x00,0x10,0x1F,0xFC,0x08,0x00,0x1E,0x78,0x78,0xDF,0x00,0x80,0xE6
    0xF8,
    0x3F,0xE0,0x10,0x01,0x40,0x04,0x03,0xE0,0x3C,0x78,0xDF,0x05,0xE2,0xF4,0x7F,0xFC,0x00,0x20,0x10,0x01,0x40,0xFF,0xE0,0x40,
    0x04,0xE0,0xC3,0x01,0xF2,0x78,0x26,0xFE,0x7F,0xA1,0xFF,0x0F,0xF8,0x20,0x80,0x84,0x0C,0xF0,0x08,0x03,0xF6,0x7F,0x74,0x7E,
    0x21,0x11,0x09,0x48,0x20,0x81,0xF8,0x01,0xF8,0x18,0x03,0xEE,0xFF,0x38,0x3E,0x01,0x40,0x04,0x03,0xE0,0x3C,0x78,0xDF,0x05,
    0xF2,0x78,0x26,0xFE,0x7F,0xA1,0xFF,0x0F,0xF8,0x20,0x80,0x84,0x0C,0xF0,0x08,0x03,0xF6,0x7F,0x74,0x7E,0x00,0x21,0x11,0x09,
    0x20,0x81,0xF8,0x01,0xF8,0x18,0x03,0xEE,0xFF,0x38,0x3E,0x01,0x40,0x04,0x03,0xE0,0x3C,0x78,0xDF,0x05,0xE2,0xF4,0x7F,0xFC,
    0xF2,0x78,0x26,0xFE,0x7F,0xA1,0xFF,0x0F,0xF8,0x20,0x80,0x84,0x0C,0xF0,0x08,0x03,0xF6,0x7F,0x74,0x7E,0x00,0x21,0x11,0x09,
    0x20,0x81,0xF8,0x01,0xF8,0x18,0x03,0xEE,0xFF,0x38,0x3E,
```

```
    0x3F,0x21,0x11,0x09,0x48,0x11,0x00,0x30,0x7C,0xE3,0x3E,0x00,0x21,0xE0,0x1E,0x1C,

    0x21,0x21,0x11,0x09,0x48,0x11,0x00,0xC2,0x7C,0x83,0x3E,0x46,0xF7,0xE0,0x1E,0x0C,0x21,0x21,0xFF,0x0A,0x
    0x1B,0x38,0x0F,0xFF,0x6E,0x78,0x36,0x21,0x21,0x11,0x0C,0x08,0x04,0x00,0x21,0x7F,0x3B,0xF9,0x3F,0xFE,0x6
    0x10,0x08,0x08,0x0A,0x01,0x24,0x67,0x61,0xF9,0xB3,0xCF,0x63,0xDE,0x06,0x20,0x20,0x10,0x0F,0xF8,0x31,0x8
    0x06,0x60,0x0F,0x18,0x0E,0x01,0xC0,0x10,0x08,0x08,0xC0,0x64,0x61,0x61,0xD9,0x76,0x06,0x60,0x07,0x18,0x1
    0x00,0x00,0x00,0x00,0x63,0x03,0xC0,0xC8,0x73,0x8F,0xE1,0x88,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x6
    0x87,0xFF,0x8C,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x60,0x1F,0x00,0xF7,0xCD,0x98,0xFF,0x18,0x00,0x00,0x00,0x00,0x0
    0x00,0x00,0x00,0x07,0xF8,0x30,0x70,0xF8,0xC3,0x8C,0x00,0x10,0x10,0x00,0x78,0x04,0x00,0x20,0x00,0x01,0xF
    0x01,0x00,0xFF,0xE0,0x88,0x7F,0xF8,0x37,0xB7,0xEE,0xEF,0xDB,0xCE,0x10,0x90,0x82,0x02,0x10,0x20,0x81,0x
    0xF0,0xE4,0xC3,0x00,0x1E,0x93,0x01,0x87,0xE0,0x11,0x06,0x03,0x6F,0xDB,0x19,0xFE,0x6F,0xFD,0xFF,0x38,0x
    0x0E,0x01,0xFC,0x6F,0xDB,0x19,0xFE,0x6F,0xFB,0x7F,0xBC,0x52,0x90,0x10,0x03,0x08,0x31,0x80,0x20,0x6F,0x
    0x1E, 0xC0, 0x0A, 0x90, 0x10, 0x0F, 0xFC, 0xD1, 0x60, 0x20, 0x6F, 0xDB, 0x1E, 0x49, 0x99, 0x9B,
    0x1E, 0x42, 0x04, 0x91, 0xFF, 0x00, 0x84, 0x11, 0x03, 0xFE, 0x6F, 0xDB, 0x3E, 0xCC, 0x00, 0x7C,
    0x66, 0x36, 0x08, 0x10, 0x10, 0x04, 0x90, 0x11, 0x00, 0x20, 0x6F, 0xD9, 0x35, 0xBB, 0x9D, 0xEF,
    0xE6, 0x26,
    0x10,0x10,0x10,0x08,0x88,0x21,0x00,0x20,0x60,0x18,0x01,0xB7,0xBF,0xE7,0xF8,0x06,0x20,0x73,0xFF,0x91,0x8
```

```
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0xF8,0x20,0x46,0x0F,0x9F,0x0E,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
};
void   delay(unsigned int m)            //Delay program
            {
              unsigned int i,j;
              for(i=0;i<m;i++)
                for(j=0;j<20;j++);
            }
void   delayms(unsigned int n)          //time delay10xnoms program
            {
              unsigned int i,j;
              for(i=0;i<n;i++)
                                                                    {
                for(j=0;j<1000;j++);
                                                                    }
            }

/************************************************** ******************
* name:sendbyte()
* Function: Send data according to the serial port communication protocol of LCD
* Enter :zdata
* output: none
************************************************************************/
void sendbyte(unsigned char zdata)
{
```

```
        unsigned int i;
        e = 0;
        for(i=0; i<8; i++)
        {rw=(bit)(zdata & 0x80);
                zdata=zdata<<1;
                e = 0;
                e = 1;
        }
        }

/*********************************************** ******************
* name:TransferData()
* Function: Write serial port instructions or data
* Enter :data1,DI
* output: none
*************************************************************************/
void TransferData(uchar data1,bit DI)
{
        if(DI==0)sendbyte(0xf8);
        else sendbyte(0xfa);
        sendbyte(data1 & 0xf0);
        sendbyte((data1 << 4) & 0xf0);
}

void initinal(void)             //LCDFont initializer
{
                rst=0;
                delay(10);
                rst=1;
         delayms(5);              //more than the40MSdelay program
         TransferData(0x30,0); //Extended Function Set :8BITset up,
                                  //RE=0: basic instruction set, G=0 : graphic display OFF //more than
         delay(10);              the100uSdelay program
         TransferData(0x30,0); //Function Set
         delay(5);              ////more than the37uSdelay program
         TransferData(0x03,0); //Display on Control
         delay(10);              //more than the100uSdelay program
         TransferData(0x0C,0); //Display Control,D=1,show on
         delay(10);              //more than the100uSdelay program
         TransferData(0x01,0); //Display Clear
         delay(2);              //more than the10mSdelay program
         TransferData(0x06,0); //Enry Mode Set,Cursor plus from right to left1bit shift
         delay(10);              //more than the100uSdelay program
        }
/*********************************************** ******************
* name:disp_dat(data1,data2)
* Function: display graph data
* Input data1,data2
* output: none
*************************************************************************
* / void disp_dat(unsigned char dat1, unsigned char dat2)
```

```
{
   int i,j;
  TransferData(0x36,0);
   for(i=0;i<16;i++)               //
    {
      TransferData((0x80 + (i*2)),0); //SETvertical addressVERTICAL ADD
      TransferData(0x80,0); //SEThorizontal addressHORIZONTAL ADD
      for(j=0;j<16;j++)
       {
         TransferData(dat1,1);
       }
         TransferData((0x80 + (i*2)+1),0); //SETvertical addressVERTICAL ADD
      TransferData(0x80,0); //SEThorizontal addressHORIZONTAL ADD
                              for(j=0;j<16;j++)
       {
                                      TransferData(dat2,1);
       }
    }
 for(i=0;i<16;i++)               //
    {
      TransferData((0x80 + (i*2)),0); //SETvertical addressVERTICAL ADD
      TransferData(0x88,0); //SEThorizontal addressHORIZONTAL ADD
      for(j=0;j<16;j++)
       {
         TransferData(dat1,1);
       }
         TransferData((0x80 + (i*2)+1),0); //SETvertical addressVERTICAL ADD
      TransferData(0x88,0); //SEThorizontal addressHORIZONTAL ADD
                              for(j=0;j<16;j++)
       {
                                      TransferData(dat2,1);
       }
        }
 }
 ////////////////////////////////////////////////////// ////////
 void disp_bmp(uchar code *bmpadd)
 { uchar i,j;
         uint k=0;

   TransferData(0x36,0);

    for(i=0;i<32;i++)               //
     {
      TransferData((0x80 + i),0); //SETvertical addressVERTICAL ADD
      TransferData(0x80,0); //SEThorizontal addressHORIZONTAL ADD
      for(j=0;j<16;j++)
       {
         TransferData(bmpadd[k],1);
                                 k++;
       }
                              }
 for(i=0;i<32;i++)               //
    {
```

```
      TransferData((0x80 + i),0); //SETvertical addressVERTICAL ADD
      TransferData(0x88,0); //SEThorizontal addressHORIZONTAL ADD
      for(j=0;j<16;j++)
       {
         TransferData(bmpadd[k],1);
                              k++;
       }
                        }

         }
/************************************************** ******************
* name:LCD_MESG()
* Function: display text function
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * / void lcd_mesg(unsigned char code *adder1)
{
        unsigned char i;
        TransferData(0x30,0);
        TransferData(0x30,0);
       TransferData(0x80,0); //Set Graphic Display RAM Address

        for(i=0;i<32;i++)
                {
                        TransferData(*adder1,1);
                        adder1++;
                }

        TransferData(0x30,0);
        TransferData(0x30,0);
       TransferData(0x90,0); //Set Graphic Display RAM Address

        for(i=0;i<32;i++)
                {
                        TransferData(*adder1,1);
                        adder1++;
                }
}
////////////////////////////////////////////////////// //////
void bmpclear()
{
uchar i,j;
        TransferData(0x34,0);

        for(i=0;i<32;i++)
        {
        TransferData(0x80+i,0);
        TransferData(0x80,0);
                for(j=0;j<32;j++)
                TransferData(0,1);
        }
}

/************************************************** ******************
```

```
* name:Main()
*  Function: main function
**************************************************************************/
void     main(void)
    {
     initial();            //transferLCDShow image (extended) initializer
       while(1)
          {
       TransferData(0x30,0);
       TransferData(0x01,0);//txtclear();
       delayms(2);
        disp_dat(0xff,0xff);
        delayms(1000);
       disp_dat(0xaa,0xaa); //Seperate Rows I
       delayms(1000);
       disp_dat(0xff,0x00); //Seperate Columns II
       delayms(1000);
       disp_dat(0xaa,0x55); //Seperate Dots I
       delayms(1000);
       disp_dat(0x55,0xaa); //Seperate Dots I
       delayms(1000);
       //txtclear();
       bmpclear();
   lcd_mesg(IC_DAT1);        //show western
   delayms(1200);
       TransferData(0x30,0);
       TransferData(0x01,0);//txtclear();
       delayms(2);
   disp_bmp(bmp1);
   delayms(1200);
       bmpclear();
       TransferData(0x34,0);
   TransferData(0x04,0);
   lcd_mesg(IC_DAT2); //Display Chinese characters
   delayms(1200);
   TransferData(0x34,0);
   TransferData(0x04,0);
      }
   }
```

10.Precautions

1.LCD Monitor(LCD)

Liquid crystal displays are made of glass, organic sealants, organic fluids, and polymer-based polarizers. Pay attention to the following items when handling:, (1). Keep the temperature within the range of use and storage. Excessive temperature and humidity can cause polarization degradation, peeling off of the polarizer, or air bubbles. (2). Do not use more thanHBThe harder stuff of the pencil lead touches the exposed polarizer. To remove the dust on the surface of the display, gently wipe it with cotton, chamois or other soft materials soaked in cleaning oil.

(3) Immediately wipe off saliva or water droplets.ITOProlonged contact with water can cause deformation or discoloration of the LCD surface. (4) Glass is easily shattered by rough handling. Especially around the corners and edges. (5). Do not drive the LCD with DC voltage.

2.LCD module

2.1Mechanical Precautions

LCMThe assembly and adjustment have high precision. Avoid excessive vibration and do not make any alterations or modifications. The following points should be noted. (1). Do not alter the lugs on the metal frame in any way.

(2) .Do not modify by drilling additional holes, changing its profile, moving its components, or modifying its patternPCB. (3).

Do not touch elastomeric connectors, especially when plugged into backlit panels (eg,EL).

(4) .InstallLCMwhen, make surePCBThe board is not subject to any stress such as bending or twisting. The elastomeric contacts are so fine that a slight misalignment of any element can result in missing pixels.

(5). Avoid pressing against the metal bezel, or the elastomeric connector may deform and lose contact, resulting in lost pixels.

2.2.static electricity

LCMIncludeCMOS LSI, the same precautions should be taken with such devices, namely

(1). When the operator comes into contact with the module, it should be grounded. Do not touch any conductive parts with any part of your body, such asLSIPad,PCB Copper conductors and interface terminals on the

(2). Modules should be stored in antistatic bags or other antistatic containers. (3).

Use only a properly grounded soldering iron.

(4). If using an electric screwdriver, it should be well grounded and prevent commutator sparks.

(5). Normal anti-static precautions should be observed for work clothes and workbenches; for the latter, conductive (rubber) mats are recommended. (6).

Since dry air can induce static electricity, the recommended relative humidity is50-60%.

2.3.welding

(1). Solder only toI/Oterminals.

(2) . Use only properly grounded and non-leakage soldering

irons. (3). Soldering temperature:280°C±10°C (4). Soldering

time:3arrive4Second. (5). Use low temperature solder filled

with resin flux.

(6). If flux is used, it should cover theLCDsurface to avoid flux spatter. Flux residues should be removed after protection.

2.4.operate

(1) viewing angle can be changed byLCDdriving voltageV0to adjust.

(2) The driving voltage should be kept within the specified range, too high voltage will shorten the life of the display. (3) The response time increases with decreasing temperature.

(4). At temperatures above its operating range, the display may turn black or dark blue; this (but do not press on the display area) may cause parts of the display to "break".

(5). Mechanical damage during operation (such as pressing on the display area) may cause a "break" in the line segment.

2.5.store

If fluid leaks from a damaged glass cell, wash any exposed body parts with soap and water. Never swallow liquid. Very low toxicity, but should always be taken with care.

2.6.limited warranty

Unless otherwise agreed with the customer, within one year from the date of shipment, will repair or repair anyLCDandIC, the confirmation of this date shall be based on the shipping documents, and the warranty liability is limited to the repair and/or replacement carried out in accordance with the above clauses. Will not be responsible for any subsequent or consequential events.