JHD12864A

CHARACTERISTICS

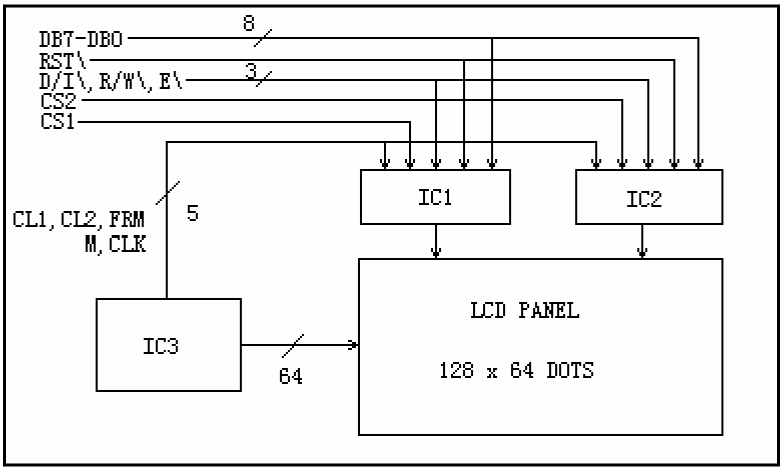
Display Content:128*64 dots

Driving Mode: 1/64D

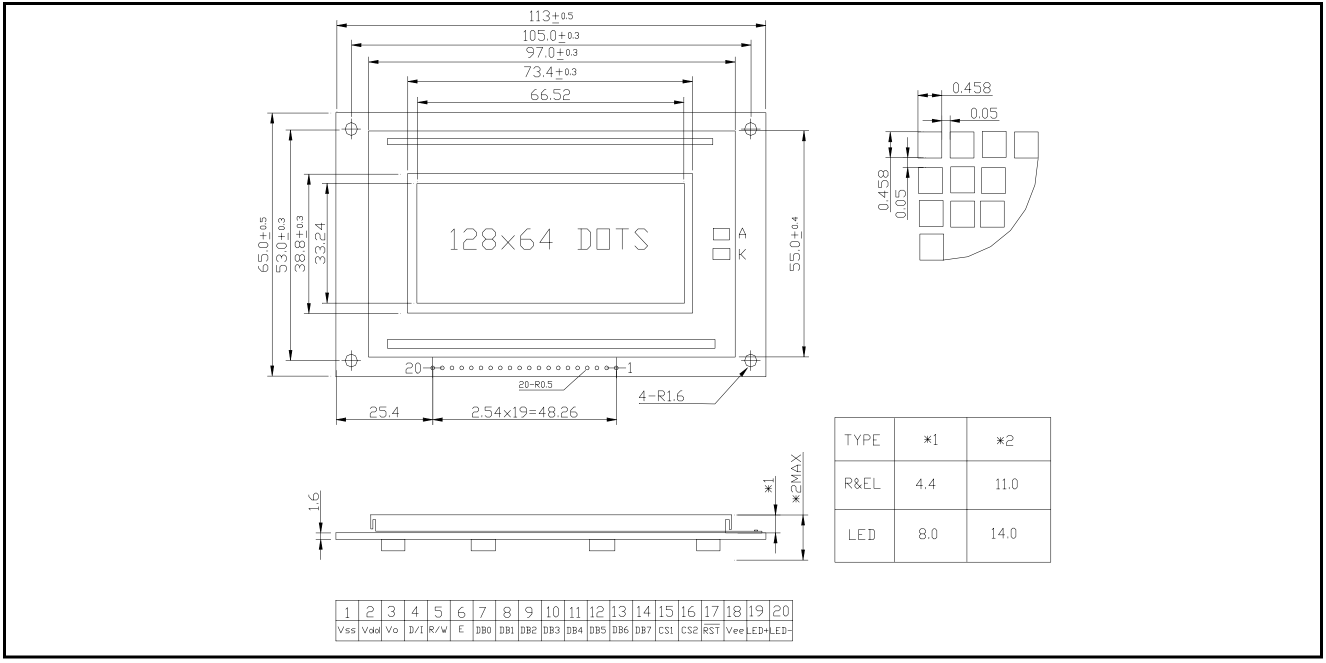
Available Types:

STN (Yellow Green, Grey, B/W)
Reflective with El or Led
El/100VAC,400HZ
LED/4.2VDC

APPLICATION CIRCUIT



Display Content



TYPE	*1	*2
R&EL	4.4	11.0
LED	8.0	14.0

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Vss	Vdd	Vo	D/I	R/W	E	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7	CS1	CS2	$\overline{\text{RST}}$	Vee	LED+	LED-

LIMIT PARAMETER

PARAMETER	Symbol	Testing Criteria	Standard Values		UINT
			MIN	MAX	
Supply Voltage	V_{DD}-V_{SS}	Ta=25°C	0	6.5	V
LCD Voltage	V_{DD}-V_0		0	18.0	V
Input Voltage	V_1		0	V_{DD}	V

ELECTRIC PARAMETER

PARAMETER		Symbol	Testing Criteia	Standard Values			UNIT
				MIN	Typical	MAX	
Voltage	LOGIC	V_{DD} -V_{SS}	-	4.75	5.0	5.25	V
	LCD	V_{DD} -V_0	-	4.5	5.5	6.5	V
Current	LOGIC	I_{DD}	-	-	2.5	-	mA
	LCD	I_{EE}	-	-	2.0	-	mA
LCD Drive Voltage (recommend)			0 ° C	-	6.2	-	V
			25 ° C	-	5.5	-	V
			40 ° C	-	4.8	-	V
Input Voltage	‘H’ Level	V_{IH}	High		-	V_{DD}	V
	‘L’ Level	V_{IL}	Low		-	0.3V_{DD}	V

PIN CONFIGURATION

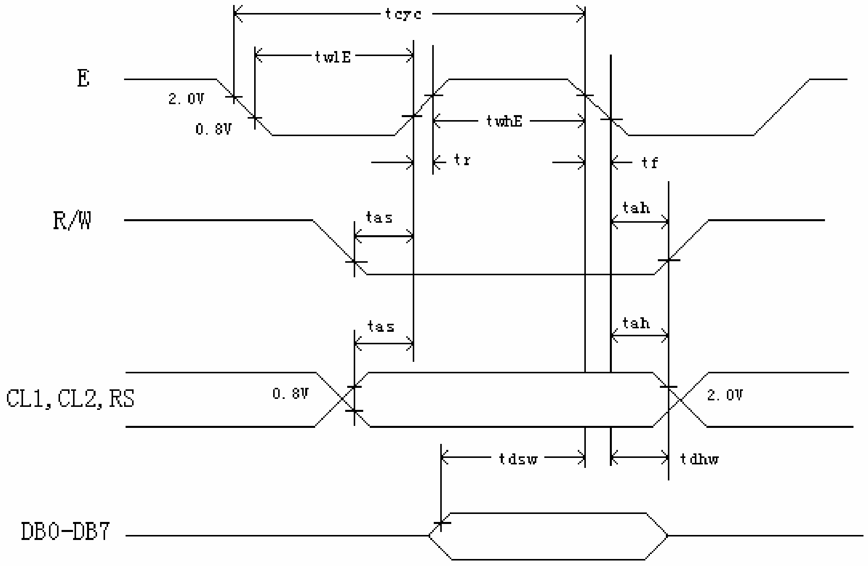
PIN	SYMBOL	LEVEL	INSTRUCTION
1	VSS	0V	Ground contact (GND)
2	VDD	5.0V	Power Supply Voltage
3	V0	LCD Drive Voltage	Adjust Contrast
4	D/I	H/L	H:DATA；L:COMMAND
5	R/W	H/L	H:READ; L:WRITE
6	E	H,H→L	IC select signal
7	DB0	H/L	DATA 0
8	DB1	H	DATA 1
9	DB2	L	DATA 2

10	DB3	H/L	DATA 3
11	DB4	H/L	DATA 4
12	DB5	H/L	DATA 5
13	DB6	H/L	DATA 6
14	DB7	H/L	DATA 7
15	CS1	H/L	Select signal 1,High effective
16	CS2	H/L	Select signal 2,High effective
17	RES	H/L	Reset signal, low effective
18	V_{EE}	-10.0V	LCD Drive negative voltage
19	LED+		Back LED Anode
20	LED-		Back LED Negative

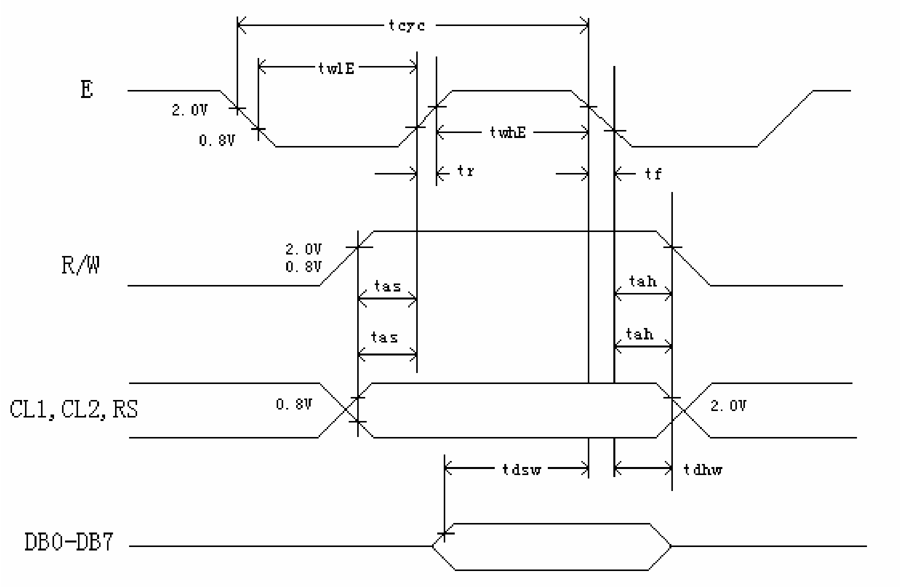
AC Characteristics (V_{dd}=5V±10%,Vss=0V Ta=25°C)

MPU Interface

Characteristic	Symbol	Min	TYPE	Max	Unit
E cycle	Tcyc	1000	---	---	ns
E high level width	TwhE	450	---	---	ns
E low level width	TwlE	450	---	---	ns
E rise time	Tr	---	---	25	ns
E fail time	Tf	---	---	25	ns
Address set-up time	Tas	140	---	---	ns
Address hold time	Tah	10	---	---	ns
Data set-up time	Tdsw	200	---	---	ns
Data delay time	Tddr	---	---	320	ns
Data hold time(write)	Tdhw	10	---	---	ns
Data hold time(read)	Tdhr	20	---		ns



MPU Write Timing



MPU Read Timing

OPERATING PRINCIPLES & METHODS

I/O Buffer

Input buffer controls the status between the enable and disable of chip. Unless the CS1 or CS2 is in active mode, input or out of data and instruction do not execute. Therefore internal state is not changed But RSTB can operate regardless of CS1 and CS2.

Input Register

Input register is provided to interface with MPU which is different operating frequency. Input register stores the data temporarily before writing it into display data RAM.

When CS1 or CS2 is in the active mode, R/W and RS select the input register. The data from MPU is written into input register and then write it into display data RAM. Data is latched when falling of the E signal and written automatically into the display data RAM by internal operation.

Output Register

Output register stores the data temporarily from display data RAM when CS1 or CS2 is in active mode and R/W and RS=H. Stored data in display data RAM is latched in output register. When CS1 or CS2 is in active mode and R/W=H, RS=L, status data(busy check) can be read out.

To read the contents of display data RAM, twice access of read instruction is needed. In first access, data in display data RAM is latched into output register. In second access, MPU can read data which is latched. That is to read the data in display data RAM, it needs dummy read. But status read does not need dummy read.

RS	R/W	Function
0	0	Instruction
	1	Status read (busy check)

1	0	Data write (from input register to display data RAM)
	1	Data read (from to display data RAM to output register)

Reset

System reset can be initialized by setting RSTB terminal at low level when turning power on, receiving instruction from MPU. When RSTB becomes low, following procedure is occurred.

-Display off

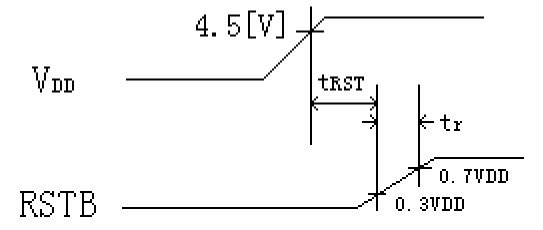
-Display start line register become set by 0. (Z-address 0)

While RSTB is low level, no instruction except status read can be accepted. Reset status appears ad DB4. After DB4 is low, any instruction can be accepted.

The conditions of power supply at initial power up are shown in table 1.

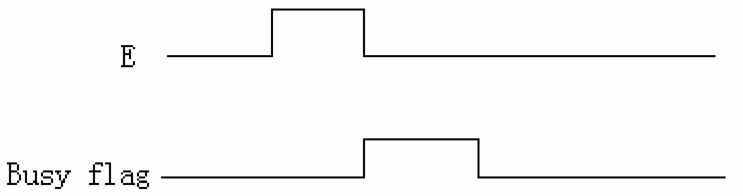
Table 1. Power Supply Initial Conditions

Item	Symbol	Min	Type	Max	Unit
Reset time	tRST	1.0	---	---	us
Rise time	tr	---	---	200	ns



Busy Flag

Busy flag indicates that KS0108B is operating or not operating. When busy flag is high, KS0108B is in internal operating. When busy flag is low, KS0108B can accept the data or instruction.DB7 indicates busy flag of the KS0108B.



Display ON/OFF Flip-Flop

The display on/off flip-flop makes on/off the liquid crystal display. When flop-flop is reset(logical low). selective voltage or non selective voltage appears on segment output terminals. When flip-flop is set (logical high). non selective voltage appears on segment output terminals regardless of display RAM data.

The display on/off flip-flop can change status by instruction. The display data at all segment disappear while RSTB is low. The status of the flop-flop is output to DB5 by read instruction.

X page Register

X page register designates page of the internal display data RAM. It has not count function. An address is set by instruction.

Y Address Counter

Y address counter designates address of the internal display data RAM. An address is set by instruction and is increased by 1 automatically by read or write operations of display data.

Display Data Ram

Display data RAM stores a display data for liquid crystal display. To express on state of dot matrix of liquid crystal display. write data 1. The other way. off state writes 0.

Display Start Line Register

The display start line register indicates address of display data RAM to display top line of liquid crystal display. Bit data (DB0:5) of the display start line set instructions is latched in display start line register. It is used for scrolling of the liquid crystal display screen.

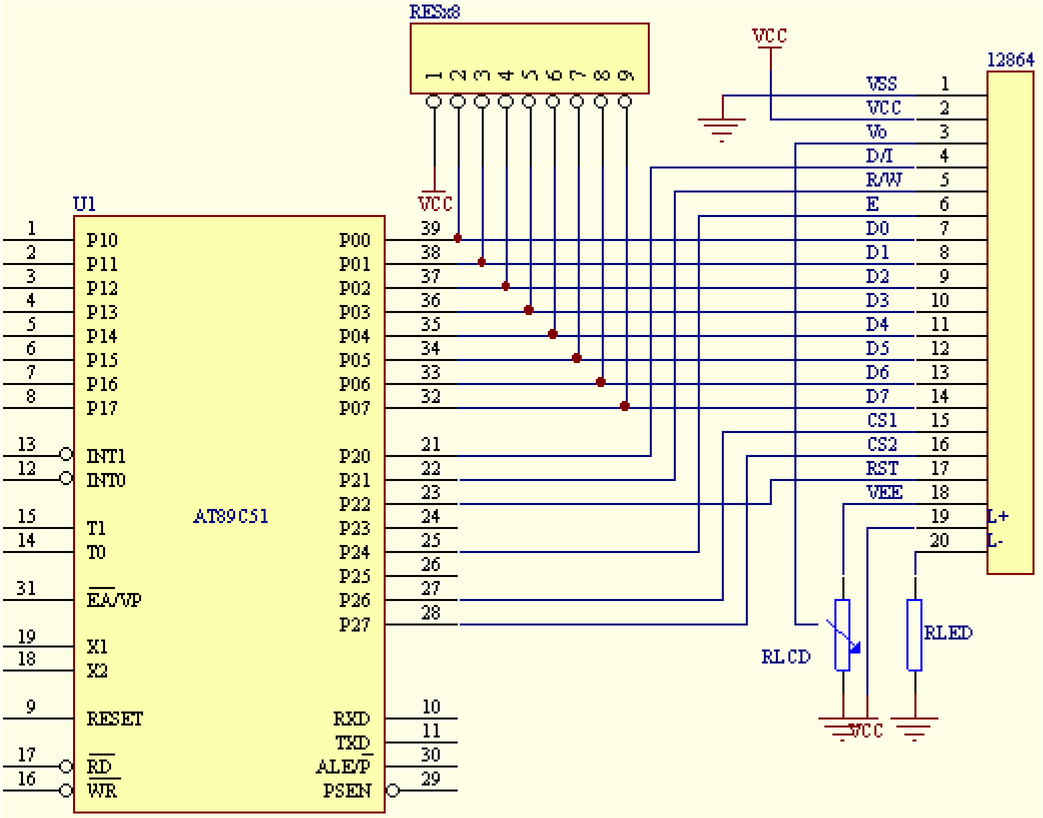
Display Control Instruction

The display control instructions control the internal state of the KS0108B. Instruction is received from MPU to KS0108B for the display control. The following table shows various instructions.

Instruction	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Function
Display ON/OFF	0	0	0	0	1	1	1	1	1	0/1	Controls the display on or off. Internal status and display RAM data are not affected. 0:OFF, 1:ON
Set Address	0	0	0	1	Y address(0-63)						Sets the Y address in the Y address counter.
Set Page	0	0	1	0	1	1	1	Page			Sets the X

(X address)								(0~7)			address at the X address register.
Display Start Line	0	0	1	1	Display start line (0~63)						Indicates the display data RAM displayed at the top of the screen
Status Read	0	1	BUSY	0	ON/OFF	RESET	0	0	0	0	Read status. Busy 0 :Ready 1 :In operation ON/OF 0 :Display ON 1 :Display OFF RESET 0 :Normal 1 :Reset
Write Display Data	1	0	Write Data								Writes data(DB0:7) into display data RAM. After writing instruction, Y address is increased by 1 automatically.
Read Display Data	1	1	Read Data								Reads data(DB0:7) from display data RAM to the data bus.

Give a Example (Only Reference)



*Different model have different with LCD negative and back light. Different model may change connect order.

Program Example (Only Reference)

```
#include <reg51.h>
#include <intrins.h>
typedef unsigned char uchar;
typedef unsigned int  uint;
sbit L_DI   =  P2^0;
sbit L_RW   =  P2^1;
sbit L_E    =  P2^4;
sbit L_CS1  =  P2^6;
sbit L_CS2  =  P2^7;
sbit L_SET  =  P2^3;
sbit TEST   =  P3^5;
float abc;
uchar code X1[] = {
    0x30,0x50,0xd0,0x90,0xf0,0x30,0x80,0x60,0x3c,0xe8,0x20,0xa0,0xe0,
0x70,0x00,0x00,
    0x04,0x06,0x03,0x11,0x13,0x12,0x18,0x0c,0x07,0x01,0x03,0x06,0x0c,
0x1c,0x18,0x18
        };  /*欢*/
uchar code X2[] = {
    0x40,0x58,0x58,0xd0,0x10,0xf8,0x98,0x08,0x88,0xf8,0xf0,0x10,0xf8,
0x70,0x00,0x00,
    0x0c,0x04,0x04,0x07,0x04,0x05,0x0d,0x0d,0x08,0x1f,0x1b,0x19,0x38,
0x38,0x18,0x18
        }; /*迎*/
uchar code X3[] = {
    0x30,0x30,0xf8,0xf8,0x08,0x10,0xd0,0x50,0x50,0xfc,0xfc,0x50,0xd0,
0xf0,0x10,0x00,
    0x00,0x00,0x1f,0x1f,0x10,0x10,0x13,0x1b,0x0f,0x07,0x05,0x0d,0x09,
0x19,0x38,0x18
        }; /*使*/
uchar code X4[] = {
    0x00,0x00,0x00,0x00,0xfc,0xf8,0x48,0x48,0x48,0xf8,0xf8,0x48,0x48,
0x48,0xfc,0xf8,
    0x10,0x10,0x18,0x1e,0x0f,0x01,0x01,0x01,0x01,0x3f,0x1f,0x01,0x01,
0x01,0x3f,0x3f
        }; /*用*/
uchar code X5[] = {
    0x18,0xfc,0x08,0x48,0x48,0xb8,0xd8,0x58,0xe8,0xb8,0xb8,0x08,0x08,
0xf8,0xf8,0x00,
    0x00,0x1f,0x11,0x11,0x11,0x14,0x15,0x16,0x16,0x10,0x11,0x11,0x11,
0x3f,0x1f,0x00
        }; /*图*/
uchar code X6[] = {
    0xc0,0xc8,0xc8,0xf8,0xf8,0xc8,0xf8,0xf8,0xc8,0xc8,0x00,0x10,0x90,
0xd8,0x48,0x48,
    0x18,0x18,0x0c,0x0f,0x03,0x00,0x1f,0x1f,0x10,0x10,0x11,0x19,0x18,
0x0c,0x04,0x06
        }; /*形*/
uchar code X7[] = {
    0x00,0x00,0x00,0xe0,0xe0,0x40,0x40,0x7c,0x7c,0x58,0x58,0x58,0xd0,
0xd0,0x00,0x00,
    0x10,0x18,0x08,0x0d,0x03,0x19,0x0d,0x01,0x01,0x0d,0x19,0x01,0x07,
0x0c,0x18,0x38
        }; /*点*/
uchar code X8[] = {
    0x08,0xf8,0xf8,0x48,0xe8,0x98,0x90,0xd8,0xf0,0xbc,0xfc,0xd0,0x90,
0x90,0x90,0x10,
    0x00,0x3f,0x1f,0x02,0x03,0x05,0x06,0x07,0x07,0x06,0x3f,0x3f,0x06,
0x06,0x04,0x04
        }; /*阵*/
uchar code X9[] = {
    0x10,0x08,0x1c,0xfe,0x00,0x00,0x00,0x00,0x00,0x30,0x08,0x04,0x04,
0x98,0xf0,0x00,
    0x80,0x80,0xff,0x80,0x00,0x00,0x00,0x00,0x00,0xa0,0xe0,0x90,0x8a,
0x83,0xc1,0x00
       }; /*12*/
uchar code X10[] = {
    0x00,0x78,0xc8,0x84,0x04,0x84,0xf8,0x00,0x00,0x80,0x80,0x00,0x00,
0x00,0x80,0x80,
    0x00,0x78,0x84,0x83,0x83,0x8e,0xfc,0x40,0x00,0x80,0x63,0x17,0x1c,
0xf2,0xc1,0x80
       }; /*8x*/
uchar code X11[] = {
    0x00,0x80,0xe0,0x10,0x08,0x04,0x04,0x00,0x00,0x00,0x80,0x60,0x10,
0xf8,0x00,0x00,
    0x00,0x7f,0xc2,0x81,0x81,0x81,0x7f,0x18,0x00,0x0e,0x09,0x08,0x08,
0xff,0x08,0x08
       }; /*64*/
uchar code X12[] = {
    0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,
0xaa,0x55,0xaa,
    0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,
0xaa,0x55,0xaa
       }; /*....*/
uchar code X13[] = {
    0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,
0xff,0xff,0xff,
    0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,
0xff,0xff,0xff
       }; /*全黑*/
uchar code X14[] = {
    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
       }; /*全白*/
uchar code X15[]={
    0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0x01,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00
       }; /*上边*/
uchar code X16[]={
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,
    0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x80
       }; /*下边*/
```

```
uchar code X17[]={
    0xff,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,
    0xff,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00
        }; /*左边*/
uchar code X18[]={
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0xff,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0xff
        }; /*右边*/
uchar code X19[]={
    0xff,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0x01,
    0xff,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00
        }; /*左上*/
uchar code X20[]={
    0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0xff,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0xff
        }; /*右上*/
uchar code X21[]={
    0xff,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,
    0xff,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x80
        }; /*左下*/
uchar code X22[]={
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0xff,
    0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0xff
        }; /*右下*/
uchar code X23[]={
    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
        }; /*空白*/
#pragma REGISTERBANK(0)
void initcomm(void)
```

```
{
    PCON = 0x80;TMOD = 0X21; SCON = 0X50;
    TH1 = 0XFa;TL1 = 0XFa; TR1 = 1;ES = 1;
    TH0 = 0;TL0 = 0;TR0 = 1; ET0 = 1;
    EA =0;
}
void delay(void)
{
    uchar i,j;
    abc = 0.1*5;
    for (i = 0;i != 0xff;i++)
        for (j = 0;j != 0xff;j++) ;
}
void LCD_C_D(bit flag,uchar ldata)          /*flag=1 con flag=0 data*/
{
    uchar data i;
    L_DI = !flag;    L_E = 0;    L_RW = 0;    P0 = ldata;
    _nop_();
    L_E = 1;
    _nop_();    _nop_();
    L_E = 0;    L_RW = 1;    L_DI = 1;
    for ( i = 0; i < 10 ;i++) ;
}

void dis_one_zi(uchar x_add,uchar y_add,uchar code *po)
{
    uchar i,j,w;
    j = x_add*2+0xb8;
    w = (y_add < 4 ? y_add : y_add-4)*16+0x40;
    L_CS1 =  (y_add < 4);
    L_CS2 = !(y_add < 4);
    LCD_C_D(1,j);
    LCD_C_D(1,w);
    for (i = 0; i < 32;i++)
    {
       if (i == 16)
        {
            LCD_C_D(1,j+1);
            LCD_C_D(1,w);
        }
        LCD_C_D(0,*po++);
    }
    L_CS1 = L_CS2 = 0;
}
```

```
void CLRLCD(uchar number)
{
    uchar data i,j;
    L_CS1 = L_CS2 = 1;
    for (i = 0xb8; i < 0xc0;i++)
    {
        LCD_C_D(1,i);
        LCD_C_D(1,0X40);
        for (j = 0; j < 0x40; j++)
            LCD_C_D(0,number);
    }
    L_CS1 = L_CS2 = 0;
}

void main(void) using 0
{
    uchar data i,j,number;
    uchar code *p;
    initcomm();
    L_SET = 0;
    for ( j = 0; j != 0xff; j++);
    L_SET = 1;
    for ( j = 0; j != 0xff; j++);
    LCD_C_D(1,0X3E); /**/
    LCD_C_D(1,0XC0); /**/
    LCD_C_D(1,0X3F); /**/
    CLRLCD(0);
    while(1)
    {
        CLRLCD(0);
    LCD_C_D(1,0X3E);
    p=X15;/*上下左右*/
       for (number=1;number<7;number++)
       {   dis_one_zi(0,number,p);        }
    p=X16;
       for (number=1;number<7;number++)
       {   dis_one_zi(3,number,p);        }
    p=X17;
       for (number=1;number<3;number++)
       {   dis_one_zi(number,0,p);        }
    p=X18;
       for (number=1;number<3;number++)
       {   dis_one_zi(number,7,p);        }
    /*四个角*/
    p=X19; dis_one_zi(0,0,p);
    p=X20; dis_one_zi(0,7,p);
    p=X21; dis_one_zi(3,0,p);
    p=X22; dis_one_zi(3,7,p);
    /*欢迎使用*/
        p=X1; dis_one_zi(1,1,p);
    p=X2;  dis_one_zi(1,2,p);
    p=X3;  dis_one_zi(1,3,p);
    p=X4;  dis_one_zi(1,4,p);
    /*图形点阵*/
    p=X5;  dis_one_zi(2,5,p);
    p=X6;  dis_one_zi(2,6,p);
    p=X7;  dis_one_zi(1,5,p);
    p=X8;  dis_one_zi(1,6,p);
    /*128X64*/
    p=X9;  dis_one_zi(2,1,p);
    p=X10; dis_one_zi(2,2,p);
    p=X11; dis_one_zi(2,3,p);
    p=X23; dis_one_zi(2,4,p);

    LCD_C_D(1,0X3F);
        delay();        delay();
    CLRLCD(0);
    LCD_C_D(1,0X3E);
    p=X12;
       for (number=0;number<4;number++)
       {
          for (i = 0; i < 8;i++)
          {   dis_one_zi(number,i,p);  }
        }
    LCD_C_D(1,0X3F);
    delay();        delay();
    CLRLCD(0);
    LCD_C_D(1,0X3E);
    p=X13;
       for (number=0;number<4;number++)
       {
          for (i = 0; i < 8;i++)
          {   dis_one_zi(number,i,p);  }
        }
    LCD_C_D(1,0X3F);
    delay();        delay();
    CLRLCD(0);
    }
```

```
}
```