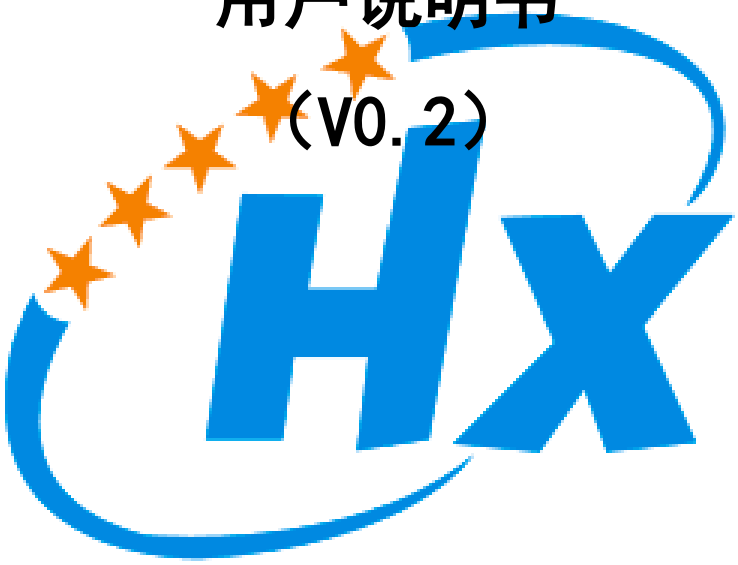
HX1230 液晶显示模块

用户说明书

（V0.2）



广西好学科技有限公司

2017.01

目 录

1 概述

HX1230 液晶显示模块是一款结构简单、小巧的单色点阵显示器；

- 分辨率为 96 列，68 行
- 只需要 4 个 IO 就可以驱动
- 单个 LED 背光，功耗更低
- 串行速率最高 4.0Mbits/s
- 外部 RST（复位）输入引脚
- 电压范围：2.7V~5.0V
- 低功耗，适用于电池供电系统
- 使用温度范围：-25~70 ℃

2 引脚

模块的引脚功能说明如表 1 所示。

表 1 模块的引脚功能说明

序号	引脚名称	功能说明	备注
1	RST	模块复位	低电平有效
2	CE	模块使能	低电平有效
3	N/C	悬空	
4	DIN	串行数据	数据出入引脚
5	CLK	串行时钟	上升沿，DIN 数据有效
6	VCC	液晶电源	3.3-5.0V
7	BL	背光灯电源	3.3-5.0V
8	GND	接地	

3 操作

3.1 操作时序

RST 引脚是用于显示模块复位，复位时间建议 10ms~100ms。

CE 引脚只有低电平时串行数据才有效。

DIN 串行数据引脚，数据格式如图 1 所示。

由一个控制位 D/C，加一个参数字节(高位先低位后)组成的 9Bit 的格式，当 D/C 数据/指令控制位为 1 时，后面的参数字节是数据；当 D/C 控制位为 0 时，后面的参数字节是指令；

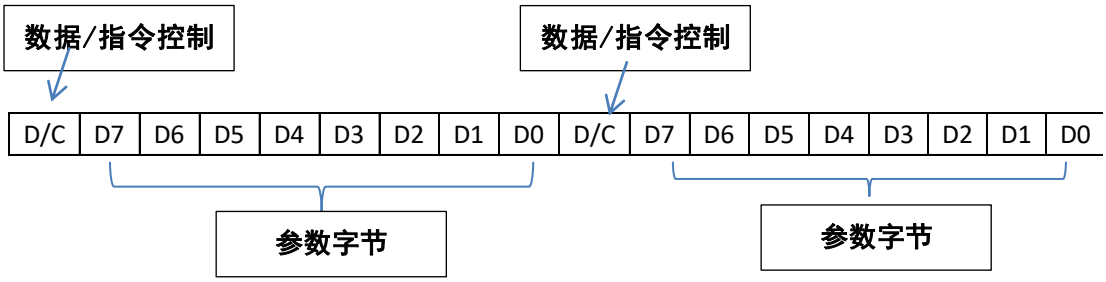


图 1 指令格式

CLK 串行时钟，当 CLK 上升沿时，对 Din 数据进行采样。

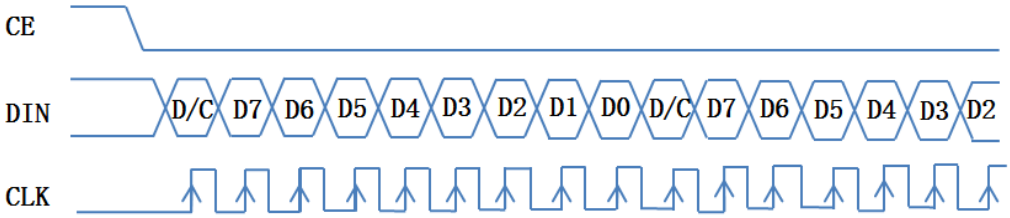


图 2 时序

3.2 指令集

HX1230 指令集如表 2 所示。

表 2 HX1230 指令集

指令	D/C	命令字节								描述
		D7	D6	D5	D4	D3	D2	D1	D0	
设置内部电源	0	0	0	1	0	W3	W2	W1	W0	当 W 都为“1111”时，开启； 当 W 为“1000”时，关闭；
设置对比度	0	1	0	0	B4	B3	B2	B1	B0	HX1230 型号（不可用）；
设置反显	0	1	0	1	0	0	1	1	N/R	当 N/R 为“0”时，正常显示； 当 N/R 为“1”时，反转显示；
设置全部显示点	0	1	0	1	0	0	1	0	N/O	当 N/O 为“0”时，关闭全显示； 当 N/O 为“1”时，打开全显示；
设置显示开关	0	1	0	1	0	1	1	1	N/S	当 N/S 为“0”时，关闭显示； 当 N/S 为“1”时，打开显示；
设置 DDRAM 的 Y 地址	0	1	0	1	1	0	Y2	Y1	Y0	设置 RAM 的 Y 地址 0≤Y≤7
设置 DDRAM 的 X 地址的低四位	0	0	0	0	0	X3	X2	X1	X0	设置 RAM 的 X 地址 0≤Y≤95
设置 DDRAM 的 X 地址的高三位	0	0	0	0	1	0	X6	X5	X4	
设置扫描起始线	0	0	1	S5	S4	S3	S2	S1	S0	设置扫描起始线 S:0≤Y≤63
写数据	1	D7	D6	D5	D4	D3	D2	D1	D0	写数据到显示 RAM 中

3.3 DDRAM MAP（显存地图）

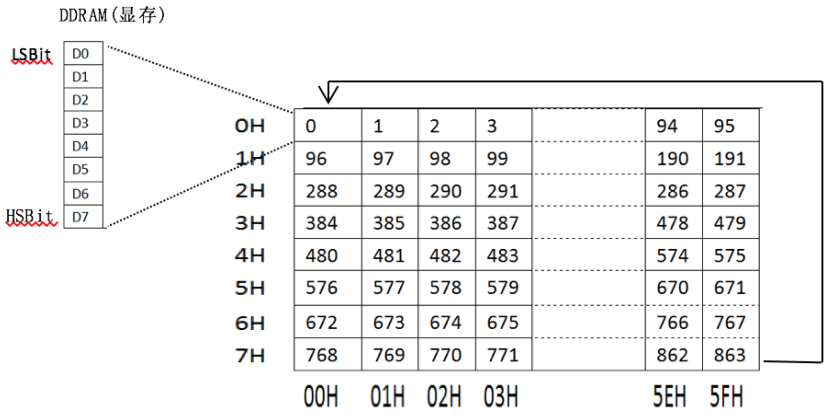


图 3 HX1230 显存地图

HX1230 显示模块 DDRAM 显存是 9 行，96 列的阵列，如图 3 所示；在每次 DDRAM 写入 1 次，列地址会自动加一，指向下一个 DDRAM 字节；当一行 96 字节写完后，会自动对行地址加一；当写到最后一行最后一列时，会自动跳回 0 行，0 列的地址。

在使用 HX1230 显示模块时，取模应该设置：阴码，逆向，列行式；

3.4 初始化过程

3.4.1 复位

接上电源后，内部寄存器和 DDRAM 的内容不确定，必须给 RST 一个复位脉冲信号。复位时间建议 10ms－100ms。

3.4.2 设置内部电源

指令	D/C	命令字节								描述
		D7	D6	D5	D4	D3	D2	D1	D0	
设置内部电源	0	0	0	1	0	W3	W2	W1	W0	当 W 都为“1111”时，开启； 当 W 为“1000”时，关闭；

3.4.3 设置反显

指令	D/C	命令字节								描述
		D7	D6	D5	D4	D3	D2	D1	D0	
设置反显	0	1	0	1	0	0	1	1	N/R	当 N/R 为“0”时，正常显示； 当 N/R 为“1”时，反转显示；

3.4.4 设置全部显示点

指令	D/C	命令字节								描述
		D7	D6	D5	D4	D3	D2	D1	D0	
设置全部显示点	0	1	0	1	0	0	1	0	N/O	当 N/O 为“0”时，关闭全显示； 当 N/O 为“1”时，打开全显示；

3.4.5 设置显示开关

指令	D/C	命令字节								描述
		D7	D6	D5	D4	D3	D2	D1	D0	
设置显示开关	0	1	0	1	0	1	1	1	N/S	当 N/S 为“0”时，关闭显示； 当 N/S 为“1”时，打开显示；

3.4.6 设置扫描起始线

指令	D/C	命令字节								描述
		D7	D6	D5	D4	D3	D2	D1	D0	
设置扫描起始线	0	0	1	S5	S4	S3	S2	S1	S0	设置扫描起始线 S：0≤Y≤63

3.4.7 设置 DDRAM 的 Y 地址

指令	D/C	命令字节								描述
		D7	D6	D5	D4	D3	D2	D1	D0	
DDRAM 的 Y 地址设置	0	1	0	1	1	0	Y2	Y1	Y0	设置 RAM 的 Y 地址：0≤Y≤7

3.4.8 设置 DDRAM 的 X 地址的高三位低四位

指令	D/C	命令字节								描述
		D7	D6	D5	D4	D3	D2	D1	D0	
设置 DDRAM 的 X 地址的低四位	0	0	0	0	0	X3	X2	X1	X0	设置 RAM 的 X 地址： 0≤Y≤95
设置 DDRAM 的 X 地址的高三位	0	0	0	0	1	0	X6	X5	X4	

3.4.9 清除 DDRAM 显存的数据（清屏）

```
set_XY(0,0);    //设置坐标。
for(i=0;i<9;i++)
{
    for(j=0;j<96;j++)
    {
        write_LCD(0x00,1);
    }
}
```

3.5 例程

```
/*****************************************************************************
    注意：例程适用于广西好学科技有限公司出品的型号 HX1230 显示模块
    接线：
        RST --> P1.0
        CE  --> P1.1
        DIN --> P1.2
        CLK --> P1.3
        BL-->P1.4
*****************************************************************************/
#include "stc12c5a60s2.h"
#include  <stdio.h>
#include "char_tab.h"

sbit HX_RST = P1^0;    //定义复位引脚
sbit HX_CE   = P1^1;    //定义使能引脚
sbit HX_DIN = P1^2;    //定义数据引脚
sbit HX_CLK = P1^3;    //定义时钟引脚
sbit HX_BL   = P1^4;    //定义背光引脚

void delay(unsigned int t);                    //延时函数
void write_HX(char volue,bit DC);              // 向 HX1230 显示模块写入一个指令/数据
DC=0:指令  DC=1:数据
void initinal_HX(void);                        //初始化 HX1230 屏幕
void set_XY(unsigned char x,unsigned char y); //定位坐标
void clr_HX(void);                             //清屏函数
void Display_Picture(char *ch);                //显示图片函数
void english_display8x8(char x , char y , char input);      //显示一个 8*8 英文字符
void sping_english8x8(char x , char y , char *ch );         //显示一串 8x8 的字符串
void display_betty_logo(int power);                         //显示电量
```

```
void english_display8x16(char x , char y , char input);     //显示一个16*8 英文字符
void sping_english8x16(char x , char y , char *ch );        //显示一串16x8 的字符

/*****************************************************************************
    函数名称：delay（）
    功能描述：用于程序延时
*****************************************************************************/
void delay(unsigned int t)
{
    unsigned int i,j;
    for(i=0;i<t;i++)
    {
        for(j=0;j<1000;j++);
    }
}

/*****************************************************************************
    函数名称：write_LCD(char volue,bit DC)
    功能描述：向 HX1230 显示模块写入一个字节的指令或数据
*****************************************************************************/
void write_HX(char volue,bit DC)    //写入一个 lcd 指令/数据    DC=0:指令   DC=1:数据
{
    int i;
    HX_CE   = 0;            //使能 HX1230 显示模块的操作
    HX_DIN = DC;            //指令或数据
    HX_CLK = 1;             //产生一个上升沿写入控制位
    HX_CLK = 0;

    for(i=0;i<8;i++)        //写一个指令或数据字节
    {
        if(volue&0x80)
        {
            HX_DIN = 1;
        }
        else
        {
            HX_DIN = 0;
        }
        HX_CLK = 1;
        volue=volue<<1;
        HX_CLK = 0;
    }
    HX_CE = 1;              //禁止 HX1230 显示模块的操作
```

```
}

/*****************************************************************************
      函数名称：initinal_LCD(void)
      功能描述：初始化 HX1230 模块的控制寄存器
*****************************************************************************/
void initinal_HX(void)       //初始化 lcd
{

      HX_CLK = 0;
      HX_RST = 0;
      delay(50);
      HX_RST   = 1;
      HX_CE = 0;
      delay(1);
      HX_CE = 1;
      delay(1);

      write_HX(0x2f,0);         //设置内部电源(ON: 0x2f / OFF:0x28 )，开启内部电源开关
      write_HX(0x90,0);         //设置对比度（不可以用）
      write_HX(0xa6,0);         //设置反显（正常：0Xa6/反显：0xa7），设置正常显示
      write_HX(0xa4,0);         //设置全部显示点(关闭：0XA4/开启：A5)，关闭全显示
      write_HX(0xaf,0);         //设置显示开关(打开：0xAF/ 关闭：0xAE)，打开显示
      write_HX(0x40,0);         //设置扫描起始线，设置扫描起始线为0
      write_HX(0xb0,0);         //设置 DDRAM 的Y 地址，设置 RAM 的Y 地址为0
      write_HX(0x10,0);         //设置 DDRAM 的X 地址的高三位，RAM 的X 地址的高三位为0
      write_HX(0x00,0);         //设置 DDRAM 的X 地址的低四位，RAM 的X 地址的低四位 0
      clr_HX();
}

/*****************************************************************************
      函数名称：set_XY(unsigned char x,unsigned char y)
      功能描述：设定显存 DDRAM 的坐标
*****************************************************************************/
void set_XY(unsigned char x,unsigned char y)          //定位坐标
{
      write_HX(0xb0 + y , 0);                         //设置 DDRAM 的Y 地址
      write_HX(0x10 | (( x & 0x7f ) >> 4), 0);        //设置 DDRAM 的X 地址的高三位
      write_HX(0x0f & x, 0);                          //设置 DDRAM 的X 地址的低四位
}

/*****************************************************************************
      函数名称：void clr_HX(void)
```

```
    功能描述：清屏
*******************************************************************************/
void clr_HX(void)                //清屏函数
{
    unsigned char i,j;
    set_XY(0,0); //设置坐标。
    for(i=0;i<9;i++)
    {
        for(j=0;j<96;j++)
        {
            write_HX(0x00,1);
        }
    }
}

/*******************************************************************************
    函数名称：void Display_Picture(char *ch)
    功能描述：显示一张大小为 96*68 的图片
*******************************************************************************/
void Display_Picture(char *ch)              //显示图片函数
{
    unsigned char i,j;

    set_XY(0,0); //设置坐标。
    for(i=0;i<9;i++)
    {
        for(j=0;j<96;j++)
        {
            write_HX(*(ch+(i * 96) + j),1);
        }
    }
}

/*******************************************************************************
    函数名称：english_display8x8(char x , char y , char input)
    功能描述：显示一个 8*8 的英文字符
*******************************************************************************/
void english_display8x8(char x , char y , char input)      //显示一个8*8 英文字符
{
    char i, *ch;
    ch = ENGLISH_tab8x8 + 8 * ( input - 32 );
    set_XY(x,y);
    for(i=0;i<8;i++)
```

```
    {
        write_HX(*(ch+i),1);
    }
}

/*******************************************************************************
    函数名称：sping_english8x8(char x , char y , char *ch )
    功能描述：显示一个 8*8 的英文字符串
*******************************************************************************/
void sping_english8x8(char x , char y , char *ch )        //显示一串 8x8 的字符串
{
    char i=0;
    while(*(ch+i)!='\0')
    {
        english_display8x8(x+8*i ,y, *(ch+i));
        i++;
    }
}

/*******************************************************************************
    函数名称：display_betty_logo(int power)
    功能描述：显示电池电量图标
*******************************************************************************/
void display_betty_logo(int power)                        //显示电量
{
    char i,volue=0,Power_mark=0x00;
    int k;
    set_XY(80,0);
    k=(0xff-power)/36;
    for(i=0;i<k;i++)
    {
        Power_mark|=0x01<<i;
    }
    for(i=0;i<10;i++)
    {
        volue = *(bettey_logo+i);
        if(i>=2&&i<9)
        {
            if(Power_mark&0x01<<(i-2))
            {
                volue&=~0x3c;
            }
        }
```

```
            write_HX(volue,1);
        }
}

/*******************************************************************************
    函数名称：english_display16x8(char x , char y , char input)
    功能描述：显示一个 8*16 英文字符
*******************************************************************************/
void english_display8x16(char x , char y , char input)        //显示一个 8*16 英文字符
{
    char i, *ch;
    ch = ENGLISH_tab8x16 + 16 * ( input - 32 );
    set_XY(x,y);
    for(i=0;i<8;i++)
    {
        write_HX(*(ch+i),1);
    }
    set_XY(x,y+1);
    for(i=0;i<8;i++)
    {
        write_HX(*(ch+i+8),1);
    }
}

/*******************************************************************************
    函数名称：sping_english8x16(char x , char y , char *ch )
    功能描述：显示一个 16*8 英文字符串
*******************************************************************************/
void sping_english8x16(char x , char y , char *ch )                    //显示一串 8*16 的字符
{
    char i=0;
    while(*(ch+i)!='\0')
    {
        english_display8x16(x+8*i ,y, *(ch+i));
        i++;
    }
}

/*******************************************************************************
    函数名称：void chinese_display(char x , char y , char *ch)      //显示一个汉字
    功能描述：显示一个 16*16 的汉字
*******************************************************************************/
```

```
void chinese_display(char x , char y , char *ch)      //显示一个汉字
{
      char i;
      set_XY(x,y);
      for(i=0;i<16;i++)
      {
            write_HX(*(ch+i),1);
      }
      set_XY(x,y+1);
      for(i=0;i<16;i++)
      {
            write_HX(*(ch+i+16),1);
      }
}

/*****************************************************************************
      函数名称：void main(void)
      功能描述：主函数
*****************************************************************************/
void main(void)
{
      char PChar[30] = 0;
      int i = 0;

      initinal_HX();                                    //初始化

      while(1)
      {
            Display_Picture(Tab_Logo);                  //显示图片函数
            delay(1000);
            clr_HX();                                   //清屏
            sping_english8x8(0, 1 , "abcdefghijkl" );   //显示一串 8x8 的字符
            sping_english8x16(0,2, "abdcefghijkl");
            chinese_display(0,4,tab_chinese);
            display_betty_logo(80);                     //显示电量
            sprintf(PChar,"Count:%d", i);
            sping_english8x8(0, 6 , PChar );            //显示一串 8x8 的字符
            i++;

            delay(1000);
      }
}
```