HX1230 liquid crystal display module

User's Guide

(V0.2 and)

Guangxi studious Technology Co., Ltd.

2017.01

table of Contents

1 Outline 1

2 Pin 1

3 operating 1

3.1 Operation Timing 1

3.2 Instruction Set 2

3.3 DDRAM MAP (Memory map) 3

3.4 Initialization process 3

3.4.1. Reset 3

3.4.2 An internal power supply disposed 3

3.4.3 Set the highlighted 3

3.4.4 All set point display 4

3.4.5 Set display switch 4

3.4.6 Setting scan start line 4

3.4.7 Set up DDRAM of Y address 4

3.4.8 Set up DDRAM of X High three low nibble address 4

3.4.9 Remove DDRAM Memory data (clear screen) 5

3.5 Routine 5

1 Outline

HX1230 The liquid crystal display module Is a simple structure, compact monochrome dot-matrix display;

- Resolution 96, line 68
- Only need four IO can drive
- Single LED backlight, lower power consumption
- Maximum serial rate 4.0Mbits / s
- External RST (reset) pin input
- Voltage range: 2.7V ~ 5.0V
- Low power consumption for battery powered system
- Temperature range: -25 ~ 70 ℃

2 Pin

Pin Function Module Description Table 1 Fig.

table 1 Pin Function Module Description

Pin Symbol		Function Description	Remark
1	RST	Module reset	Active Low
2	CE	Module Enable	Active Low
3	N / C	Suspended	
4	DIN	Serial Data	Data out pin
5	CLK	Serial Clock	Rising, DIN effective data
6	VCC	LCD power supply	3.3-5.0V
7	BL	Backlight Power	3.3-5.0V
8	GND	Ground	

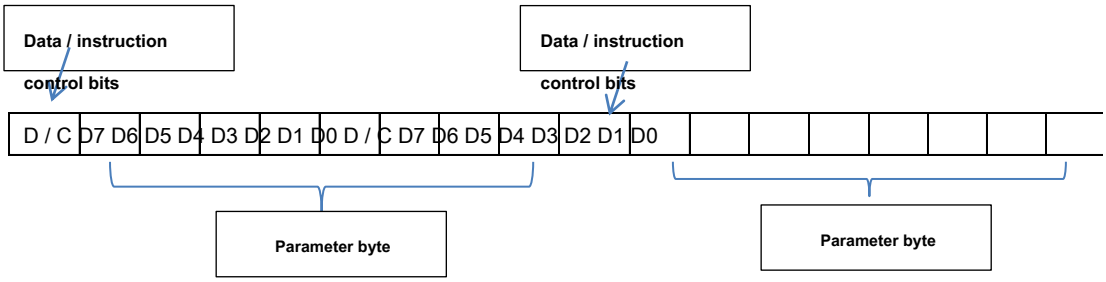
3 operating

3.1 Operation Timing

RST pin is used to reset the display module, the reset time is recommended 10ms ~ 100ms.

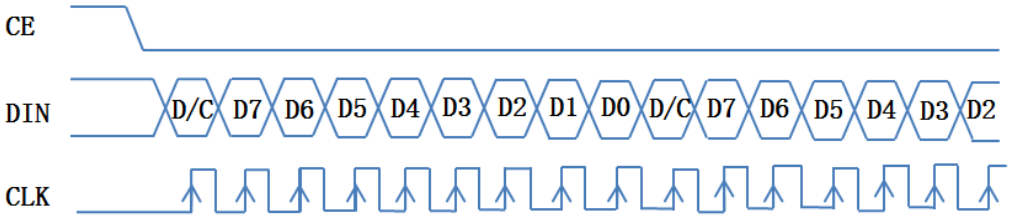
CE pin only valid serial data is low. Serial data pin DIN, the data format as shown in FIG.

Consists of a control bit D / C, add a parameter byte (high to low post) composed 9Bit format, when the D / C data / instruction control bit is 1, the data byte following the parameters; if D / C control bit is 0, the parameter is an instruction byte behind;



Map 1 Instruction format

Serial Clock CLK, when the rising edge of CLK, the data Din to be sampled.



Map 2 Timing

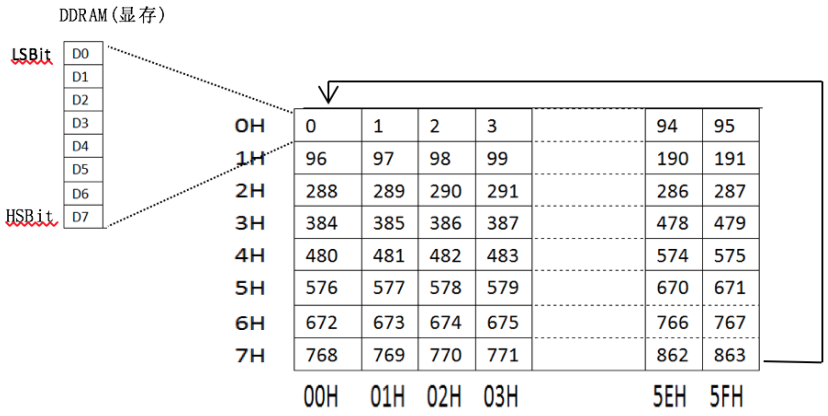
3.2 Instruction Set

HX1230 Instruction Set Table 2 Fig.

table 2 HX1230 Instruction Set

instruction	D / C	Command byte								description
		D7	D6	D5	D4	D3	D2	D1	D0	
An internal power supply is provided	0	0	0	1	0	W3	W2 W1 W0 when W are "1111", open; When W is "1000", closed;			
Contrast settings	0	1	0	0	B4	B3	B2	B1	B0	HX1230 model (not available);
Set reverse video	0	1	0	1	0	0	1	1	When N / R	When N / R is "0", the normal display; When N / R is "1", displayed in reverse;
All set points 0		10		10		0	1	0	When the	N / O if N / O is "0", to close the whole display; When the N / O is "1", opens the whole display;
Set display switch	0	10		10		1	1	1	When the	N / S when N / S is "0", the display is turned off; When N / S is "1", the display is opened;
Set the DDRAM Y address	0	1	0	11		0	Y2	Y1		RAM addresses provided Y0 Y 0≤Y≤7
Set the DDRAM low four X address 0	0	0	0	00		X3	X2	X1		X0 X address of RAM is provided
X address is set upper three bits 0 DDRAM	0	0	0	01		0	X6	X5	X4	0≤Y≤95
Setting scan start line 0		0	1	S5	S4	S3	S2	S1		Setting scan start line S0 S: 0≤Y≤63
Write data	1	D7	D6	D5	D4	D3	D2	D1		Writing data to the display D0 in RAM

3.3 DDRAM MAP (Memory map)



Map 3 HX1230 Memory Map

HX1230 display module DDRAM memory is an array of rows 9, 96, as shown in FIG. 3; 1 in each DDRAM write, the column address is automatically incremented point to the next byte DDRAM; 96 byte line when finished, will automatic row address plus one; when wrote the last line of the last one, it will automatically jump back row address 0, 0's.

currently using HX1230 display module , The modulus should be provided: a female code, reverse, row line;

3.4 Initialization process

3.4.1. Reset

Connected to the power supply, and the internal registers DDRAM Contents undefined, must give RST A reset pulse signal. Reset time is recommended 10ms - 100ms .

3.4.2 An internal power supply is provided

instruction	D / C	Command byte								description
		D7	D6	D5	D4		D3	D2	D1	D0	
Set an internal power source 0		0	0	1	0	W3	W2	W1	W0	when W are "1111", open; When W is "1000", closed;	

3.4.3 Set reverse video

instruction	D / C	Command byte								description
		D7	D6	D5	D4		D3	D2	D1	D0	
Set reverse video	0	1	0	1	0	0	1	1	When N / R	When N / R is "0", the normal display; When N / R is "1", displayed in reverse;	

3.4.4 All set point display

instruction	D / C	Command byte								description
		D7	D6	D5	D4	D3	D2	D1	D0	
Set all significant Illustrates the point	0	1	0	1	0	0	1	0	When the N / O if N / O is "0", to close the whole display; When the N / O is "1", opens the whole display;	

3.4.5 Set display switch

instruction	D / C	Command byte								description
		D7	D6	D5	D4	D3	D2	D1	D0	
Set display switch 0		10		10		1	1	1	When the N / S when N / S is "0", the display is turned off; When N / S is "1", the display is opened;	

3.4.6 Setting scan start line

instruction	D / C	Command byte								description
		D7	D6	D5	D4	D3	D2	D1	D0	
Setting scan play Starting line	0	0	1	S5	S4	S3	S2	S1	Setting scan start line S0 S: 0≤Y≤63	

3.4.7 Set up DDRAM of Y address

instruction	D / C	Command byte								description
		D7	D6	D5	D4	D3	D2	D1	D0	
Y address provided DDRAM 0		1	0	1	1	0	Y2	Y1	The RAM address Y0 of Y: 0≤Y≤7	

3.4.8 Set up DDRAM of X High three low nibble address

instruction	D / C	Command byte								description
		D7	D6	D5	D4	D3	D2	D1	D0	
Set the DDRAM low four X address 0		0	0	0	0	X3 X2 X1 X0 setting the X RAM Address:				0≤Y≤95
X address is set upper three bits 0 DDRAM		0	0	0	1	0	X6 X5 X4			

3.4.9 Remove DDRAM Memory data (clear screen)

```
set_XY (0,0);          // Set coordinates.
for (i = 0; i <9; i ++) {

        for (j = 0; j <96; j ++) {

                write_LCD (0x00,1);}}
```

3.5 Routine

```
/ ********************************************** ***************************
        Note: suitable for routine studious Technology Co., Ltd. Guangxi produced models HX1230 display module
        wiring:
                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;          // Define the reset pin
sbit HX_CE = P1 ^ 1;           // Definitions enable pin
sbit HX_DIN = P1 ^ 2;          // Defined data pins
sbit HX_CLK = P1 ^ 3;          // Defined clock pin
sbit HX_BL = P1 ^ 4;           // Defined backlight pin

void delay (unsigned int t);                                    // Delay function
void write_HX (char volue, bit DC);                             // to HX1230 Display Module Writes a command / data
DC = 0: instruction DC = 1: data
void initinal_HX (void);                                        // initialization HX1230 screen
void set_XY (unsigned char x, unsigned char y); // Positioning coordinates
void clr_HX (void);                                             // Clear screen function
void Display_Picture (char * ch);                               // Display image function
void english_display8x8 (char x, char y, char input);           // Display a 8 * 8 English characters
void sping_english8x8 (char x, char y, char * ch);              // To display a string 8x8 String
void display_betty_logo (int power);                            // Display power
```

```
void english_display8x16 (char x, char y, char input);          // Display a 16 * 8 English characters
void sping_english8x16 (char x, char y, char * ch);              // To display a string 16x8 character of

/ ************************************************** ****************************
        Function Name: Delay () Function
        Description: program delay
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * /
void delay (unsigned int t) {

        unsigned int i, j; for (i =
        0; i <t; i ++) {

                for (j = 0; j <1000; j ++);}}



/ ************************************************** ****************************
        Function Name: write_LCD (char volue, bit DC) Description: to HX1230 display module Write
        a byte instructions or data
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * /
void write_HX (char volue, bit DC)          // Write a lcd Instruction / data DC = 0: instruction DC = 1: data
{
        int i;
        HX_CE = 0;                          // Enable HX1230 Display Module Operations
        HX_DIN = DC;                        // Instructions or data
        HX_CLK = 1;                         // Generating a rising edge of the write control bit
        HX_CLK = 0;

        for (i = 0; i <8; i ++)             // Write a command or data byte
        {
                if (volue & 0x80) {

                        HX_DIN = 1;}
                else {


                        HX_DIN = 0;}

                HX_CLK = 1; volue =
             volue << 1; HX_CLK = 0;}


        HX_CE = 1;                          // Ban HX1230 The operation display module
```


http://shop111128253.taobao.com/
TEL: 15676107689/15277013241

```
}

/ ********************************************** ***************************
        Function Name: initinal_LCD (void)
        Function Description: Initializes control registers module HX1230
* ********************************************************************************************/
void initinal_HX (void)              // initialization 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);          // Set the internal power supply ( ON: 0x2f / OFF: 0x28) , Internal power switch is turned on
        write_HX (0x90,0);           // Set contrast (not used)
        write_HX (0xa6,0);           // Set reverse display (normal: 0Xa6 / Anti obvious: 0xa7 ), Normal display settings
        write_HX (0xa4,0);           // Setting all of the display point (closed: 0XA4 / ON: A5) , Close full display
        write_HX (0xaf, 0);          // Display setting switch (open: 0xAF / shut down: 0xAE) Open display
        write_HX (0x40,0);           // Setting scan start line, the scanning start line is provided 0
        write_HX (0xb0,0);           // Set up DDRAM of Y Address, set RAM of Y Address 0
        write_HX (0x10,0);           // Set up DDRAM of X High three addresses, RAM of X High three address is 0
        write_HX (0x00,0);           // Set up DDRAM of X The low four addresses, RAM of X The lower four bits of address 0
        clr_HX ();}



/ ********************************************** ***************************
        Function Name: set_XY (unsigned char x, unsigned char y) Function
        Description: Sets the coordinate memory DDRAM
* ********************************************************************************************/
void set_XY (unsigned char x, unsigned char y)                  // Positioning coordinates
{
        write_HX (0xb0 + y, 0);                                 // Set up DDRAM of Y address
        write_HX (0x10 | ((x & 0x7f) >> 4), 0);                 // Set up DDRAM of X High three addresses
        write_HX (0x0f & x, 0);                                 // Set up DDRAM of X The lower four bits of address
}

/ ********************************************** ***************************
        Function name: void clr_HX (void)
```

```
    Description: clear screen
*********************************************************************************************/
void clr_HX (void)                    // Clear screen function
{
    unsigned char i, j; set_XY (0,0); // Set
    coordinates.
    for (i = 0; i <9; i ++) {

        for (j = 0; j <96; j ++) {

            write_HX (0x00,1);}}}




/ ********************************************** ***************************
    Function name: void Display_Picture (char * ch)
    Description: a display size of 96 * 68 pictures
*********************************************************************************************/
void Display_Picture (char * ch)                    // Display image function
{
    unsigned char i, j;

    set_XY (0,0); // Set coordinates.
    for (i = 0; i <9; i ++) {

        for (j = 0; j <96; j ++) {

            write_HX (* (ch + (i * 96) + j), 1);}}}




/ ********************************************** ***************************
    Function Name: english_display8x8 (char x, char y, char input) Description:
    English characters a display 8 * 8
*********************************************************************************************/
void english_display8x8 (char x, char y, char input)                    // Display a 8 * 8 English characters
{
    char i, * ch;
    ch = ENGLISH_tab8x8 + 8 * (input - 32); set_XY (x, y);
    for (i = 0; i <8; i ++)
```

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



/ *************************************************** *****************************
        Function Name: sping_english8x8 (char x, char y, char * ch) Function: a
        character string display English 8 * 8
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
void sping_english8x8 (char x, char y, char * ch)                  // To display a string 8x8 String
{
        char i = 0;
        while (* (ch + i)! = '\ 0') {

                english_display8x8 (x + 8 * i, y, * (ch + i)); i ++;}}




/ *************************************************** *****************************
        Function Name: display_betty_logo (int power)
        Description: displays battery icon
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
void display_betty_logo (int power)                          // Display power
{
        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);}}



/ *************************************************** ****************************
    Function Name: english_display16x8 (char x, char y, char input) Function Description:
    displaying a 8 * 16 English characters
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
void english_display8x16 (char x, char y, char input)          // Display a 8 * 16 English characters
{
    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);}}



/ *************************************************** ****************************
    Function Name: sping_english8x16 (char x, char y, char * ch) Description:
    displaying a string of English 16 * 8
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
void sping_english8x16 (char x, char y, char * ch)             // To display a string 8 * 16 character of
{
    char i = 0;
    while (* (ch + i)! = '\ 0') {

        english_display8x16 (x + 8 * i, y, * (ch + i)); i ++;}}




/ *************************************************** ****************************
    Function name: void chinese_display (char x, char y, char * ch)          // display a Chinese character
    Description: a display Chinese characters 16 *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
```

```
void chinese_display (char x, char y, char * ch)                // Display a Chinese character
{
        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);}}



/ *************************************************** ****************************
        Function Name: void main (void)
        Description: The main function
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * /
void main (void) {

        char PChar [30] = 0; int i
        = 0;

        initinal_HX ();                                         // initialization

        while (1) {

                Display_Picture (Tab_Logo);                     // Display image function
                delay (1000);
                clr_HX ();                                      // Clear screen
                sping_english8x8 (0, 1, "abcdefghijkl");        // To display a string 8x8 character of
                sping_english8x16 (0,2, "abdcefghijkl");
                chinese_display (0,4, tab_chinese);
                display_betty_logo (80);                        // Display power
                sprintf (PChar, "Count:% d", i);
                sping_english8x8 (0, 6, PChar);                 // To display a string 8x8 character of
                i ++;

                delay (1000);}}
```