**一步一步学MSP430FR5739之LCD5110**

OS: Windows8 Pro 32 bit

IDE: Code Composer Studio v5.1.1

Board: TI MSP430FR5739

今天要做的就是给MSP430FR5739配一个LCD。

第一步WDTCTL = WDTPW + WDTHOLD; // Stop WDT

1、首先定义好相应管教

P2DIR=BIT5+BIT6; //设置I/O口的方向

P1DIR=BIT0+BIT1+BIT2;

**#define** DIN\_H P1OUT|=BIT1

**#define** DIN\_L P1OUT&=~BIT1

**#define** CLK\_H P1OUT|=BIT0

**#define** CLK\_L P1OUT&=~BIT0

**#define** DC\_H P1OUT|=BIT2

**#define** DC\_L P1OUT&=~BIT2

**#define** CE\_H P2OUT|=BIT5

**#define** CE\_L P2OUT&=~BIT5

**#define** RST\_H P2OUT|=BIT6

**#define** RST\_L P2OUT&=~BIT6

2、调用LCD初始化程序

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*名称:LCD\_Init(void)

\*\*功能：初始化液晶

\*\*说明：

\*\*备注:

\*\*日期:2011年2月27日

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** **LCD\_Init**(**void**)

{

P2DIR=BIT5+BIT6;//+BIT4+BIT0+BIT1;//设置P4口的方向

P1DIR=BIT0+BIT1+BIT2;

RST\_L;//复位

RST\_H;

write\_command(0x21);//工作模式, 水平寻址, 扩展指令

write\_command(0x13);//设置偏置系统(BSx) 1:48

write\_command(0xc8);//设置电压VLCD = 3.06 + 0.06\*Vop, 对比度调整

write\_command(0x06);//VLCD温度系数2

write\_command(0x20);//工作模式, 水平寻址, 常规指令

write\_command(0x0c);//普通模式

write\_command(0x40);//起始页地址0

write\_command(0x80);//起始列地址0

LCD\_clear(); //清全屏

}

3、调用其他需要的函数

display\_chinese\_string\_huanhang(1,0,4,Characters[0]);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*名称:display\_chinese\_string\_huanhang(uchar x,uchar y,uchar n,const uchar \*str)

\*\*功能：输出一串16\*16的汉字

\*\*说明：x:0-4

y:0-2

n:要显示汉字的个数

str：汉字的首地址

\*\*备注:每行最多只能显示5个汉字(n\*16<=84)并且可以换行显示

\*\*日期:2011年2月27日

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** **display\_chinese\_string\_huanhang**(uchar x,uchar y,uchar n,**const** uchar \*str)

{

uchar i;

**for**(i=0;i<n;i++)

{

display\_chinese(x++,y,str+i\*32);

**if**(x==5)

{

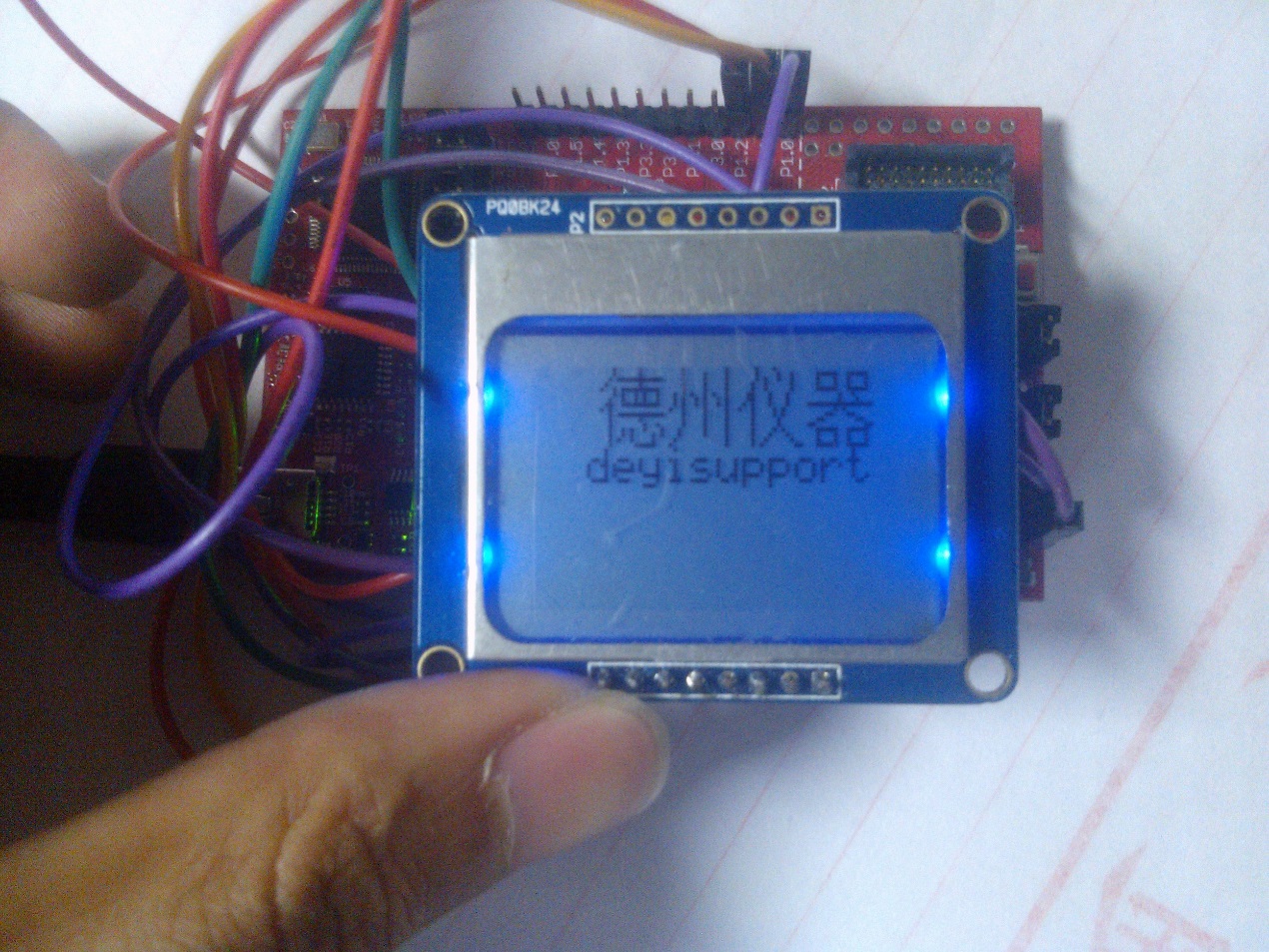
x=0;

y++;

}

}

}

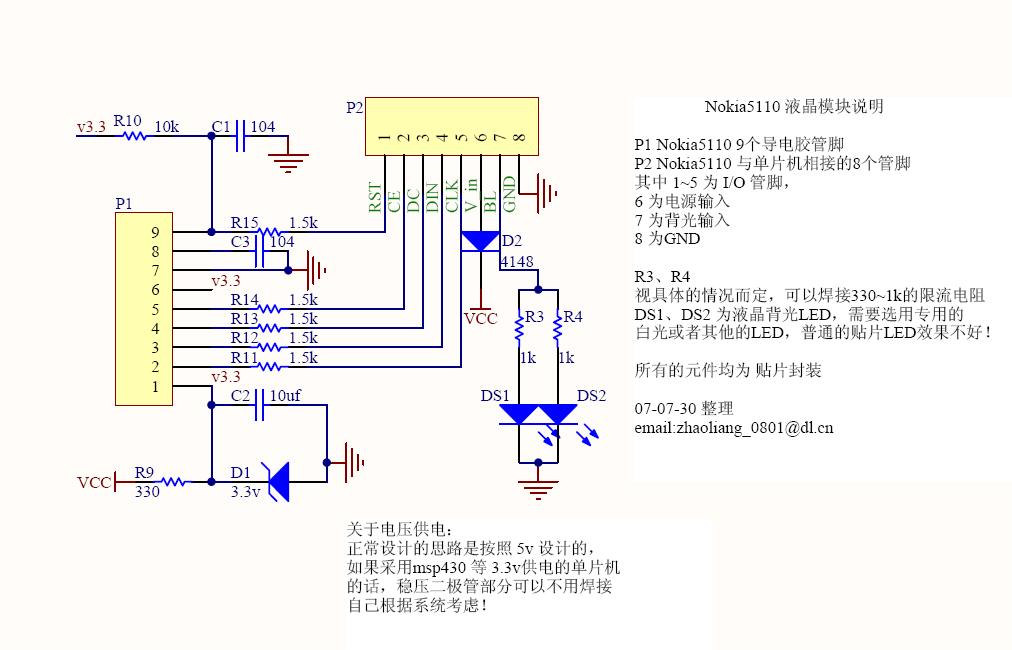


代码：



LCD5110说明：]

（申明以下内容部分来自Baidu文库）



Nokia5110是一款经典机型，可能由于经典的缘故，旧机器很多，所以很多电子工程师就把旧机器的屏幕拆下来，自己驱动Nokia5110，用于开发的设备显示，取代LCD1602。

使用Nokia5110液晶的四大理由

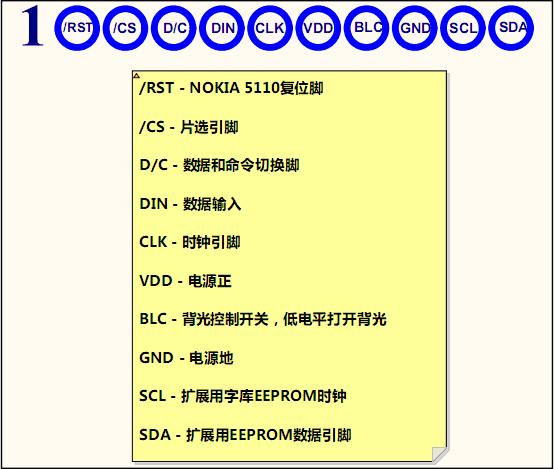
1)     性价比高，LCD1602可以显示32个字符，而Nokia5110可以显示15个汉字，30个字符。Nokia5110裸屏仅8.8元，LCD1602一般15元左右，LCD12864一般50~70元。

2)     接口简单，仅四根I/O线即可驱动，LCD1602需11根I/O线，LCD12864需12根。

3)     速度快，是LCD12864的20倍，是LCD1602的40倍。

4)     Nokia5110工作电压3.3V，正常显示时工作电流200uA以下，具有掉电模式，适合电池供电的便携式移动设备。

Nokia5110原理图：

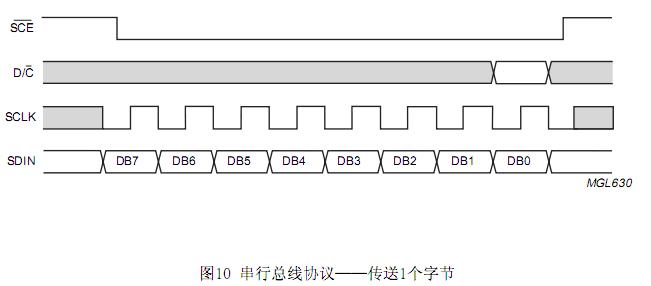


商家把该引出的管脚都引出了，下面就可以连接C8051F的相应管脚，写程序驱动5110了。

Nokia5110和Nokia3110是兼容的，用的都是同一款驱动芯片PHILIPS PCD8544。

**(1)   SPI接口时序写数据/命令**

Nokia5110(PCD8544)的通信协议是一个没有MISO只有MOSI的SPI协议，如果单片机有富裕的SPI接口，也可以利用硬件SPI，但通常没有必要，只需要软件程序模拟即可。



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*名称:write\_command(uchar cmd)

\*\*功能：向液晶中写命令

\*\*说明：cmd：要写的命令

\*\*备注:

\*\*日期:2011年2月27日

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** **write\_command**(uchar cmd)

{

uchar i;

CE\_L;//允许写入数据

DC\_L;//DC=0写入命令

**for**(i=0;i<8;i++)

{

**if**(cmd&0x80)

DIN\_H;

**else**

DIN\_L;

CLK\_L; //上升沿采样

cmd<<=1;

CLK\_H;

}

CE\_H;//禁止写入

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*名称:write\_data(uchar dat)

\*\*功能：向液晶中写入数据

\*\*说明：dat:待写入的数据

\*\*备注:参考数据手册

\*\*日期:2011年2月27日

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** **write\_data**(uchar dat)

{

uchar i;

CE\_L;//

DC\_H;

**for**(i=0;i<8;i++)

{

**if**(dat&0x80)

DIN\_H;

**else**

DIN\_L;

CLK\_L;

dat<<=1;

CLK\_H;

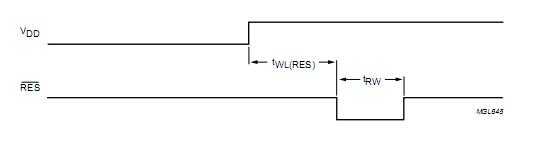
}

CE\_H;

}

**(2)   Nokia5110的初始化**

接通电源后，内部寄存器和RAM的内容是不确定的，这需要一个RES低电平脉冲复位一下。当VDD变为高电平，达到VDDmin(或更高)之后，最多100ms，RST输入低电平(电平幅度<0.3VDD)。



5110复位时序

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*名称:LCD\_Init(void)

\*\*功能：初始化液晶

\*\*说明：

\*\*备注:

\*\*日期:2011年2月27日

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** **LCD\_Init**(**void**)

{

P2DIR=BIT5+BIT6;//+BIT4+BIT0+BIT1;//设置P4口的方向

P1DIR=BIT0+BIT1+BIT2;

RST\_L;//复位

RST\_H;

write\_command(0x21);//工作模式, 水平寻址, 扩展指令

write\_command(0x13);//设置偏置系统(BSx) 1:48

write\_command(0xc8);//设置电压VLCD = 3.06 + 0.06\*Vop, 对比度调整

write\_command(0x06);//VLCD温度系数2

write\_command(0x20);//工作模式, 水平寻址, 常规指令

write\_command(0x0c);//普通模式

write\_command(0x40);//起始页地址0

write\_command(0x80);//起始列地址0

LCD\_clear(); //清全屏

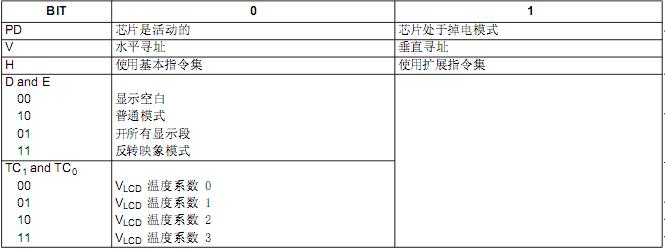
}

**(3)设置Nokia5110液晶的坐标**

Nokia5110(PCD8544)的命令集：



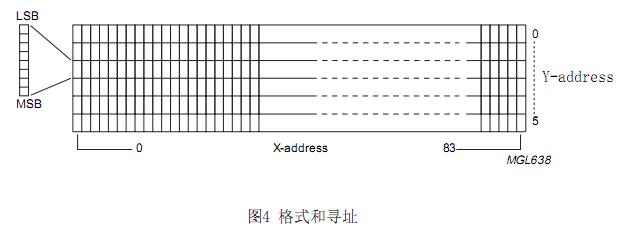
命令字符解释：



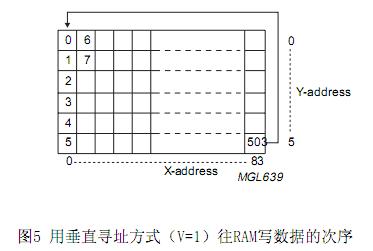
要注意的是“功能设置”命令的“H”和“V”。

H=0使用基本指令集命令，H=1使用扩展指令集命令。

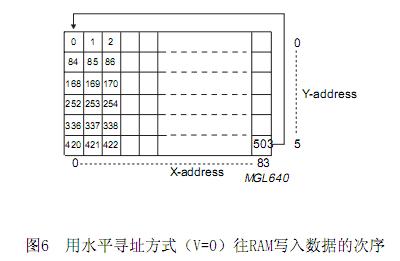
V=0水平寻址，V=1垂直寻址。



48\*84点阵的格式如上图，数据以字节8bit为单位下载到PCD8544的48\*84bit显示数据RAM矩阵。列通过地址指针寻址，地址范围为X 0~83(1010011)，Y 0~5(101)。地址不允许超过这个范围。



在垂直寻址(V=1)，Y地址在每个字节之后递增，经最后的Y地址(Y=5)之后，Y绕回0，X递增到下一列的地址。



在水平寻址模式(V=0)，X地址在每个字节之后递增，经最后的X地址(X=83)之后，X绕回0，Y递增到下一行的地址。

经每一个最后地址之后(X=83,Y=5)，地址指针绕回地址(X=0,Y=0)。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*名称:set\_pos(uchar x,uchar y)

\*\*功能：设置RAM地址

\*\*说明：x:0-83

y:0-5

\*\*备注:参考数据手册

\*\*日期:2011年2月27日

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** **set\_pos**(uchar x,uchar y)

{

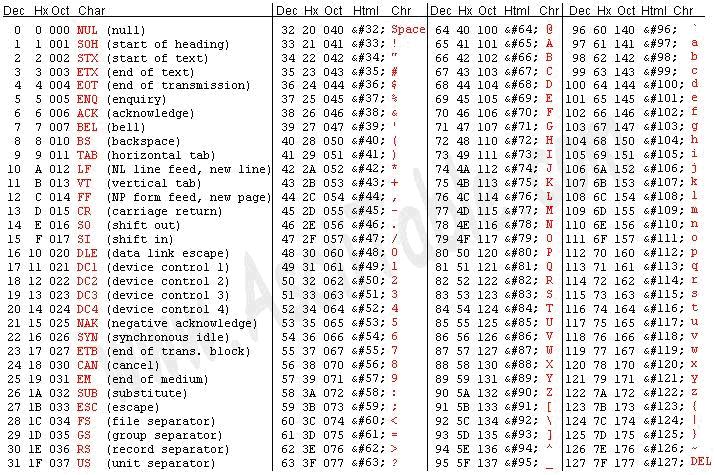
write\_command(0x40 | y);

write\_command(0x80 | x); //0<x<84;0<y<6;

}

**(4)显示英文字符**

英文字符占用6\*8个点阵，通过建立一个ASCII的数组font6x8[][6]来寻址。而这个数字的行号对应（字符ASCII码-32）。ASCII码表如下：



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*名称:display(uchar x,uchar y,uchar c)

\*\*功能：液晶输出字符（6\*8字符）

\*\*说明：x:0-83

y:0-5

c：待显示的字符

\*\*备注:每行最多只能显示14个字符（6\*n<=84)

\*\*日期:2011年2月27日

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** **display\_char**(uchar x,uchar y,uchar c)

{

uchar i;

c-=32; //获取地址指针

x\*=6;//字宽

set\_pos(x,y);//设置显示地址

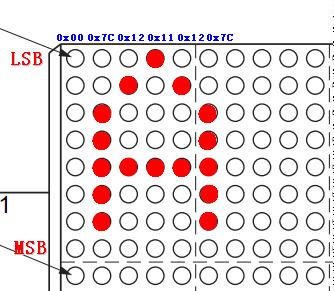
**for**(i=0;i<6;i++)

write\_data(font6x8[c][i]);//写入点阵数据

}

如要在NOKIA5110输出大写“A”，对应数组的行号为33，对应的元素为

{ 0x00, 0x7C, 0x12, 0x11, 0x12, 0x7C }，水平寻址(V=0)



NOKIA5110 一行可以显示14个字符，可以显示六行。

**(5)显示汉字**

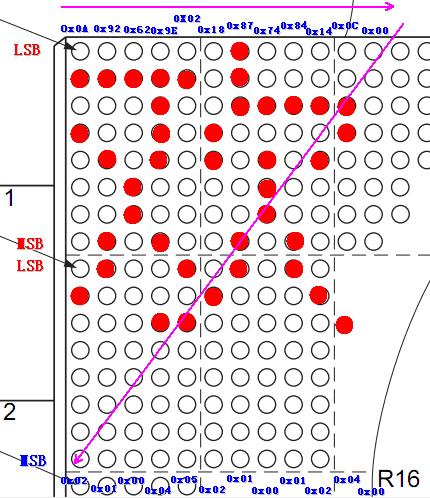
显示汉字可以采用两种点阵方式，一种是12\*12点阵，一种是16\*16点阵，网上也有一个自动提取字模的小软件“5110液晶显示屏汉字提取程序”，这个小程序只适用于水平寻址(V=0)。

5110液晶显示屏汉字提取程序 下载：

采用12\*12点阵汉字时，由于不是8的整数倍数，因此行与行之间只能隔开，这样才能完整显示一个汉字。

//欢

{0x0A,0x92,0x62,0x9E,0x02,0x18,0x87,0x74,0x84,0x14,0x0C,0x00,0x02,0x01,0x00,0x04,0x05,0x02,0x01,0x00,0x01,0x02,0x04,0x00},



12\*12点阵

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*名称:display\_chinese(uchar x,uchar y,const uchar \*str)

\*\*功能：显示单个16\*16的汉字

\*\*说明：x:0-4

y:0-2

str：汉字的首地址

\*\*备注:每行最多只能显示5个汉字(n\*16<=84)

\*\*日期:2011年2月27日

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** **display\_chinese**(uchar x,uchar y, **const** uchar \*str)

{

uchar i,j;

x\*=16;//字宽

y\*=2;//两行显示

**for**(i=0;i<2;i++)

{

set\_pos(x,y+i);//设置显示的地址

**for**(j=0;j<16;j++)

{

write\_data(str[j+i\*16]);

}

}

}