#include "app.h"

extern const uint8\_t code HZ\_tab[][16];

uint8\_t flag = 0;

uint8\_t able[2] = {0};

extern uint8\_t key\_val;

void Delay1000ms() //@11.0592MHz

{

unsigned char i, j, k;

\_nop\_();

i = 8;

j = 1;

k = 243;

do

{

do

{

while (--k);

} while (--j);

} while (--i);

}

int main(void)

{

int i = 0;

Init\_DS18B20();

InitLcd();

LCD\_ShowChinaLanguage(0 ,0 , &HZ\_tab[0][0], 5);

LCD\_ShowChinaLanguage(2 ,0 , &HZ\_tab[10][0], 5);

LCD\_ShowChinaLanguage(2 ,7 , &HZ\_tab[68][0], 1);

LCD\_ShowChinaLanguage(4 ,0 , &HZ\_tab[20][0], 5);

LCD\_ShowChinaLanguage(4 ,7 , &HZ\_tab[68][0], 1);

LCD\_ShowChinaLanguage(6 ,0 , &HZ\_tab[30][0], 5);

Timer0Init();

while(1)

{

;

}

}

#include "app.h"

extern const uint8\_t code HZ\_tab[][16];

uint8\_t flag = 0;

uint8\_t able[2] = {0};

extern uint8\_t key\_val;

void Delay1000ms() //@11.0592MHz

{

unsigned char i, j, k;

\_nop\_();

i = 8;

j = 1;

k = 243;

do

{

do

{

while (--k);

} while (--j);

} while (--i);

}

int main(void)

{

int i = 0;

Init\_DS18B20();

InitLcd();

LCD\_ShowChinaLanguage(0 ,0 , &HZ\_tab[0][0], 5);

LCD\_ShowChinaLanguage(2 ,0 , &HZ\_tab[10][0], 5);

LCD\_ShowChinaLanguage(2 ,7 , &HZ\_tab[68][0], 1);

LCD\_ShowChinaLanguage(4 ,0 , &HZ\_tab[20][0], 5);

LCD\_ShowChinaLanguage(4 ,7 , &HZ\_tab[68][0], 1);

LCD\_ShowChinaLanguage(6 ,0 , &HZ\_tab[30][0], 5);

Timer0Init();

while(1)

{

;

}

}

#include "temp.h"

//\*\*\*\*\*\*\*\*带返回值的延时子程序\*\*\*\*\*\*\*\*\*\*\*

void delay\_TEMP(unsigned char time)

{

unsigned char n;

n=0;

while(n<time)n++;

return;

}

//\*\*\*\*\*\*\*\*DS18B20复位\*\*\*\*\*\*\*\*\*

void Init\_DS18B20(void)

{

DQ=1;

delay\_TEMP(8);

DQ=0;

delay\_TEMP(85);

DQ=1;

delay\_TEMP(34);

}

//\*\*\*\*\*\*\*单片机向DS18B20写一个字节\*\*\*\*\*\*

unsigned char ReadOneChar(void)

{

unsigned char i=0;

unsigned char dat=0;

for(i=8;i>0;i--)

{

DQ=1;

delay\_TEMP(1);

DQ=0;

dat>>=1;

DQ=1;

delay\_TEMP(1);

if(DQ) dat|=0x80;

delay\_TEMP(10);

}

return (dat);

}

//\*\*\*\*\*\*单片机读DS18B20一个字节\*\*\*\*\*\*\*\*\*\*

void WriteOneChar(unsigned char dat)

{

unsigned char i=0;

for(i=8;i>0;i--)

{

DQ=0;

DQ=dat&0x01;

delay\_TEMP(5);

DQ=1;

dat>>=1;

}

delay\_TEMP(10);

}

//\*\*\*\*\*\*\*单片机读取DS18B20的温度\*\*\*\*\*\*\*\*\*

float ReadTemperature(void)

{

unsigned char tempL=0;

unsigned char tempH=0;

float temperature;

Init\_DS18B20();

WriteOneChar(0xcc);

WriteOneChar(0x44);

delay\_TEMP(30);

Init\_DS18B20();

WriteOneChar(0xcc);

WriteOneChar(0xbe);

tempL=ReadOneChar();

tempH=ReadOneChar();

temperature=((tempH\*256)+tempL)\*0.0625;

delay\_TEMP(30);

return(temperature); //带返回值的程序，返回的值为温度值

}

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

\* 函 数 名 : datapros()

\* 函数功能 : 温度读取处理转换函数

\* 输 入 : temp

\* 输 出 : 无

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

uint8\_t Ds18b20DisplayData[4] = {0};

uint8\_t \* Ds18b20\_datapros(float temp)

{

int n = 10 \* temp; //浮点型的m变成整型的n

Ds18b20DisplayData[0] = n/100 + '0';

Ds18b20DisplayData[1] = n%100/10 + '0';

Ds18b20DisplayData[2] = '.';

Ds18b20DisplayData[3] = n%10 + '0'; //整型的n拆位

return Ds18b20DisplayData;

}

#ifndef \_\_TEMP\_H\_

#define \_\_TEMP\_H\_

#include <app.h>

//---重定义关键词---//

#ifndef uchar

#define uchar unsigned char

#endif

#ifndef uint

#define uint unsigned int

#endif

//--定义使用的IO口--//

sbit DQ=P3^4;

//--声明全局函数--//

void delay\_TEMP(unsigned char time) ;

//\*\*\*\*\*\*\*DS18B20复位\*\*\*\*\*\*\*\*\*

void Init\_DS18B20(void);

//\*\*\*\*\*\*\*单片机向DS18B20写一个字节\*\*\*\*\*\*

unsigned char ReadOneChar(void);

//\*\*\*\*\*\*单片机读DS18B20一个字节\*\*\*\*\*\*\*\*\*\*

void WriteOneChar(unsigned char dat);

//\*\*\*\*\*\*\*单片机读取DS18B20的温度\*\*\*\*\*\*\*\*\*

float ReadTemperature(void);

uint8\_t \* Ds18b20\_datapros(float temp) ;

#endif

#include "timer.h"

extern const uint8\_t code HZ\_tab[][16];

extern uint8\_t save\_data;

extern uint8\_t key\_able[];

uint8\_t motor\_count = 0;

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

\* 函 数 名 : Timer0Init

\* 函数功能 : 定时器0初始化

\* 输 入 : 无

\* 输 出 : 无

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

void Timer0Init(void)

{

TMOD|=0X01; // 选择为定时器0模式，工作方式1，仅用TR0打开启动。

TL0 = 0x00; //设置定时初值

TH0 = 0xDC; //设置定时初值

ET0=1; //打开定时器0中断允许

EA=1; //打开总中断

TR0=1; //打开定时器

}

uint8\_t \* \_pDs18b20 = (uint8\_t \*)0;

uint8\_t key\_val = 0;

uint8\_t pwm\_val = 0;

float temp\_val = 0;

void Timer0() interrupt 1

{

static uint8\_t delay\_10ms = 0;

int i = 0;

TL0 = 0x00; //设置定时初值

TH0 = 0xDC; //设置定时初值

delay\_10ms ++;

kscanf();

if(delay\_10ms == 50)

{

delay\_10ms = 0;

temp\_val = ReadTemperature();

\_pDs18b20 = Ds18b20\_datapros(temp\_val); //数据处理函数

LCD\_ShowString\_length(4, 9, \_pDs18b20 ,4);

LCD\_ShowNumber(2, 9, key\_able[1], 2);

switch(key\_able[0])

{

case 0: LCD\_ShowChinaLanguage(0 ,5 , &HZ\_tab[48][0], 2);

P1 &= ~0x10;P1 |= 0x20;P1 |= 0x40;

break;

case 1: LCD\_ShowChinaLanguage(0 ,5 , &HZ\_tab[40][0], 2);

if(key\_able[1] > temp\_val)

key\_able[2] = 3;

else

key\_able[2] = save\_data;

P1 &= ~0x20;P1 |= 0x10;P1 |= 0x40;

break;

case 2: LCD\_ShowChinaLanguage(0 ,5 , &HZ\_tab[44][0], 2);

if(key\_able[1] < temp\_val)

key\_able[2] = 3;

else

key\_able[2] = save\_data;

P1 &= ~0x40;P1 |= 0x10;P1 |= 0x20;

break;

default:key\_able[0] = 0; break;

}

switch(key\_able[2])

{

case 0: LCD\_ShowChinaLanguage(6 ,5 , &HZ\_tab[60][0], 2);

motor\_count = 2;

break;

case 1: LCD\_ShowChinaLanguage(6 ,5 , &HZ\_tab[56][0], 2);

motor\_count = 5;

break;

case 2: LCD\_ShowChinaLanguage(6 ,5 , &HZ\_tab[52][0], 2);

motor\_count = 45;

break;

case 3: LCD\_ShowChinaLanguage(6 ,5 , &HZ\_tab[64][0], 2);

motor\_count = 0;

break;

default:key\_able[3] = 0; break;

}

}

if(pwm\_val++ < motor\_count)

{

MOTOR\_Bit = 0;

}

else if(pwm\_val > 50)

{

pwm\_val = 0;

}

else

{

MOTOR\_Bit = 1;

}

}

#ifndef \_TIMER\_H

#define \_TIMER\_H

#include "app.h"

void Timer0Init(void);

#endif

#include "key.h"

uint8\_t key\_able[3]= {0, 27, 0};

uint8\_t save\_data = 0;

void Delay5ms() //@11.0592MHz

{

unsigned char i, j;

i = 9;

j = 244;

do

{

while (--j);

} while (--i);

}

uint8\_t kscanf(void)

{

static uint8\_t key\_mode = 1;

static last\_keyval = 1, keyflag = 1;

if((P1&0x0f)^0x0f != 0 && last\_keyval != (P1&0x0F) && keyflag)

{

keyflag = 0;

switch(P1&0x0f)

{

case 0x0E:

key\_able[0]++;

if(key\_able[0] > 2)

key\_able[0] = 0;

if(key\_able[2] == 3)

key\_able[2] = 0;

break;

case 0x0D:

key\_able[1]++;

if(key\_able[1] > 30)

key\_able[1] = 16;

break;

case 0x0B:

key\_able[1]--;

if(key\_able[1] < 16)

key\_able[1] = 30;

break;

case 0x07:

key\_able[2]++;

if(key\_able[2] > 2)

key\_able[2] = 0;

save\_data = key\_able[2];

break;

}

}

if(P1 & 0x0F == 0X0F)

keyflag = 1;

last\_keyval = P1&0x0F;

return key\_mode;

}

#ifndef \_KEY\_H

#define \_KEY\_H

#include "app.h"

sbit key1 = P2^7;

uint8\_t kscanf(void);

void Delay5ms(); //@11.0592MHz

#endif

#include "AMPIRE12864.h"

#include "ascllfont.h"

void CheckState()

{

E = 1;

RS = 0;

RW = 1;

LcdDataPort = 0xff;

while(!busy);

}

void LcdDelay(unsigned int time)

{

while(time --);

}

void WriteData(uint8\_t dat)

{

CheckState();

E = 1;

RS = 1;

RW = 0;

LcdDataPort = dat;

E = 0;

}

void SendCommand(uint8\_t command)

{

CheckState();

E = 1;

RW = 0;

RS = 0;

LcdDataPort = command;

E = 0;

}

void SelectScreen(uint8\_t screen) //0-全屏，1—左屏，2-右屏

{

switch(screen)

{

case 0 :

CS1 = 0;

LcdDelay(2);

CS2 = 1;

LcdDelay(2);

break;

case 1 :

CS1 = 1;

LcdDelay(2);

CS2 = 0;

LcdDelay(2);

break;

case 2 :

CS1 = 0;

LcdDelay(2);

CS2 = 0;

LcdDelay(2);

break;

}

}

void ClearScreen(uint8\_t screen) // screen 0-全屏，1—左屏，2-右屏

{

uint8\_t i,j;

SelectScreen(screen);

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

{

SetLine(i);

SetColum(0);

for(j = 0;j < 64; j ++)

WriteData(0);

}

}

void SetLine(uint8\_t line) //line -> 0 : 7

{

line = line & 0x07;

line = line | 0xb8; //1011 1xxx

SendCommand(line);

}

void SetColum(uint8\_t column) //column -> 0 :63

{

column = column & 0x3f;

column = column | 0x40; //01xx xxxx

SendCommand(column);

}

void SetStartLine(uint8\_t startline) //startline -> 0 : 63

{

startline = startline & 0x3f;

startline = startline | 0xc0; //11xxxxxx

SendCommand(startline);

}

void SetOnOff(uint8\_t onoff) //1-开显示 0-关

{

if(onoff == 1)

SendCommand(0x3f); //0011 111x

else

SendCommand(0x3e);

}

void SetClear(void)

{

LCD\_ShowString(0, 0, " ");

LCD\_ShowString(2, 0, " ");

LCD\_ShowString(4, 0, " ");

LCD\_ShowString(6, 0, " ");

}

void ResetLcd()

{

Reset = 0;

LcdDelay(2);

Reset = 1;

LcdDelay(2);

RS0 = 0;

LcdDelay(2);

RS1 = 0;

LcdDelay(2);

SetOnOff(1);

}

void InitLcd()

{

ResetLcd();

SendCommand(0x30);

SetOnOff(0);

ClearScreen(2);

SetLine(0);

SetColum(0);

SetStartLine(0);

SetOnOff(1);

}

void Show1616(uint8\_t lin,uint8\_t column,uint8\_t \*address)

{

uint8\_t i;

if(column < 64)

{

SelectScreen(0);

}

else

{

SelectScreen(1);

}

SetLine(lin);

SetColum(column);

for(i = 0;i < 16;i ++)

WriteData(\*(address ++));

SetLine(lin + 1);

SetColum(column);

for(i = 0;i < 16;i ++)

WriteData(\*(address ++));

}

void Show1608(uint8\_t lin,uint8\_t column,uint8\_t \*address)

{

uint8\_t i;

SetLine(lin);

SetColum(column);

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

WriteData(\*(address ++));

SetLine(lin + 1);

SetColum(column);

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

WriteData(\*(address ++));

}

void LCD\_ShowChinaLanguage(uint8\_t lin,uint8\_t column,uint8\_t \*p, uint8\_t Language\_num)

{

uint8\_t i = 0;

column \*= 16;

while(Language\_num > i ++)

{

Show1616(lin, column, p);

p += 32;

column += 16;

}

}

//在指定位置显示一个字符,包括部分字符

//x:0~127

//y:0~63

void LCD\_ShowChar(uint8\_t lin,uint8\_t column,uint8\_t chr)

{

if(column \* 8 < 64)

{

SelectScreen(0);

}

else

{

SelectScreen(1);

}

chr = chr - ' ';//得到偏移后的值

Show1608(lin, column \* 8, &led\_asc2\_1608[chr][0]);

}

//m^n函数

unsigned long LCD\_pow(uint8\_t m,uint8\_t n)

{

unsigned long result=1;

while(n--)result\*=m;

return result;

}

//显示2个数字

//x,y :起点坐标

//len :数字的位数

//num:数值(0~4294967295);

void LCD\_ShowNumber(uint8\_t lin,uint8\_t column,unsigned long num,uint8\_t len)

{

uint8\_t t,temp;

uint8\_t enshow=0;

for(t = 0; t < len; t++)

{

temp=(num/LCD\_pow(10,len-t-1))%10;

if(enshow==0&&t<(len-1))

{

if(temp==0)

{

LCD\_ShowChar(lin, column, ' ');

column += 1;

continue;

}else enshow=1;

}

LCD\_ShowChar(lin, column, temp + '0');

column += 1;

}

}

//显示字符串

//x,y:起点坐标

//\*p:字符串起始地址

//用16字体

void LCD\_ShowString(uint8\_t lin,uint8\_t column,const uint8\_t \*p)

{

while(\*p!='\0')

{

LCD\_ShowChar(lin, column, \*p);

column++;

p++;

}

}

void LCD\_ShowString\_length(uint8\_t lin,uint8\_t column, uint8\_t \*p, uint8\_t length)

{

while(length-- > 0)

{

LCD\_ShowChar(lin, column, \*p);

column++;

p++;

}

}

#ifndef \_\_ASCLLFONT\_H

#define \_\_ASCLLFONT\_H

//常用ASCII表

//偏移量32

//ASCII字符集

//偏移量32

//大小:16\*8

const uint8\_t code HZ\_tab[][16] = {

{0x40,0x40,0x42,0xCC,0x00,0x20,0x22,0x22,0xA2,0x62,0x22,0x22,0x22,0x20,0x00,0x00},

{0x00,0x40,0x20,0x1F,0x20,0x44,0x4E,0x45,0x44,0x44,0x44,0x45,0x46,0x4C,0x40,0x00},/\*"运",0\*/

{0x00,0x10,0x88,0xC4,0x33,0x00,0x40,0x42,0x42,0x42,0xC2,0x42,0x42,0x42,0x40,0x00},

{0x02,0x01,0x00,0xFF,0x00,0x00,0x00,0x00,0x40,0x80,0x7F,0x00,0x00,0x00,0x00,0x00},/\*"行",1\*/

{0x10,0x10,0xD0,0xFF,0x90,0x14,0xE4,0xAF,0xA4,0xA4,0xA4,0xAF,0xE4,0x04,0x00,0x00},

{0x04,0x03,0x00,0xFF,0x00,0x89,0x4B,0x2A,0x1A,0x0E,0x1A,0x2A,0x4B,0x88,0x80,0x00},/\*"模",2\*/

{0x10,0x10,0x90,0x90,0x90,0x90,0x90,0x10,0x10,0xFF,0x10,0x10,0x11,0x16,0x10,0x00},

{0x00,0x20,0x60,0x20,0x3F,0x10,0x10,0x10,0x00,0x03,0x0C,0x10,0x20,0x40,0xF8,0x00},/\*"式",3\*/

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},

{0x00,0x00,0x36,0x36,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/\*"：",4\*/

{0x40,0x40,0x42,0xCC,0x00,0x40,0xA0,0x9E,0x82,0x82,0x82,0x9E,0xA0,0x20,0x20,0x00},

{0x00,0x00,0x00,0x3F,0x90,0x88,0x40,0x43,0x2C,0x10,0x28,0x46,0x41,0x80,0x80,0x00},/\*"设",5\*/

{0x00,0x17,0x15,0xD5,0x55,0x57,0x55,0x7D,0x55,0x57,0x55,0xD5,0x15,0x17,0x00,0x00},

{0x40,0x40,0x40,0x7F,0x55,0x55,0x55,0x55,0x55,0x55,0x55,0x7F,0x40,0x40,0x40,0x00},/\*"置",6\*/

{0x10,0x60,0x02,0x8C,0x00,0x00,0xFE,0x92,0x92,0x92,0x92,0x92,0xFE,0x00,0x00,0x00},

{0x04,0x04,0x7E,0x01,0x40,0x7E,0x42,0x42,0x7E,0x42,0x7E,0x42,0x42,0x7E,0x40,0x00},/\*"温",7\*/

{0x00,0x00,0xFC,0x24,0x24,0x24,0xFC,0x25,0x26,0x24,0xFC,0x24,0x24,0x24,0x04,0x00},

{0x40,0x30,0x8F,0x80,0x84,0x4C,0x55,0x25,0x25,0x25,0x55,0x4C,0x80,0x80,0x80,0x00},/\*"度",8\*/

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},

{0x00,0x00,0x36,0x36,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/\*"：",9\*/

{0x10,0x0C,0x04,0x84,0x14,0x64,0x05,0x06,0xF4,0x04,0x04,0x04,0x04,0x14,0x0C,0x00},

{0x04,0x84,0x84,0x44,0x47,0x24,0x14,0x0C,0x07,0x0C,0x14,0x24,0x44,0x84,0x04,0x00},/\*"实",10\*/

{0x00,0xFE,0x22,0x5A,0x86,0x00,0x20,0x22,0x22,0x22,0xE2,0x22,0x22,0x22,0x20,0x00},

{0x00,0xFF,0x04,0x08,0x07,0x10,0x0C,0x03,0x40,0x80,0x7F,0x00,0x01,0x06,0x18,0x00},/\*"际",11\*/

{0x10,0x60,0x02,0x8C,0x00,0x00,0xFE,0x92,0x92,0x92,0x92,0x92,0xFE,0x00,0x00,0x00},

{0x04,0x04,0x7E,0x01,0x40,0x7E,0x42,0x42,0x7E,0x42,0x7E,0x42,0x42,0x7E,0x40,0x00},/\*"温",12\*/

{0x00,0x00,0xFC,0x24,0x24,0x24,0xFC,0x25,0x26,0x24,0xFC,0x24,0x24,0x24,0x04,0x00},

{0x40,0x30,0x8F,0x80,0x84,0x4C,0x55,0x25,0x25,0x25,0x55,0x4C,0x80,0x80,0x80,0x00},/\*"度",13\*/

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},

{0x00,0x00,0x36,0x36,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/\*"：",14\*/

{0x00,0x40,0x42,0x44,0x58,0x40,0x40,0x7F,0x40,0x40,0x50,0x48,0xC6,0x00,0x00,0x00},

{0x00,0x40,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0xFF,0x00,0x00,0x00},/\*"当",15\*/

{0x08,0x08,0xE8,0x29,0x2E,0x28,0xE8,0x08,0x08,0xC8,0x0C,0x0B,0xE8,0x08,0x08,0x00},

{0x00,0x00,0xFF,0x09,0x49,0x89,0x7F,0x00,0x00,0x0F,0x40,0x80,0x7F,0x00,0x00,0x00},/\*"前",16\*/

{0x00,0x00,0xFE,0x02,0x12,0x22,0xC2,0x02,0xC2,0x32,0x02,0xFE,0x00,0x00,0x00,0x00},

{0x80,0x60,0x1F,0x00,0x20,0x10,0x0C,0x03,0x0C,0x30,0x00,0x0F,0x30,0x40,0xF8,0x00},/\*"风",17\*/

{0x40,0x40,0x42,0xCC,0x00,0x04,0xF4,0x94,0x94,0xFF,0x94,0x94,0xF4,0x04,0x00,0x00},

{0x00,0x40,0x20,0x1F,0x20,0x48,0x44,0x42,0x41,0x5F,0x41,0x42,0x44,0x48,0x40,0x00},/\*"速",18\*/

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},

{0x00,0x00,0x36,0x36,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/\*"：",19\*/

{0x40,0x50,0x4E,0x48,0x48,0xFF,0x48,0x48,0x48,0x40,0xF8,0x00,0x00,0xFF,0x00,0x00},

{0x00,0x00,0x3E,0x02,0x02,0xFF,0x12,0x22,0x1E,0x00,0x0F,0x40,0x80,0x7F,0x00,0x00},/\*"制",0\*/

{0x00,0x02,0x0C,0xC0,0x00,0x40,0x20,0x10,0x0C,0x23,0xCC,0x10,0x20,0x40,0x40,0x00},

{0x02,0x02,0x7F,0x00,0x00,0x00,0x01,0x09,0x11,0x21,0xD1,0x0D,0x03,0x00,0x00,0x00},/\*"冷",1\*/

{0x40,0x50,0x4E,0x48,0x48,0xFF,0x48,0x48,0x48,0x40,0xF8,0x00,0x00,0xFF,0x00,0x00},

{0x00,0x00,0x3E,0x02,0x02,0xFF,0x12,0x22,0x1E,0x00,0x0F,0x40,0x80,0x7F,0x00,0x00},/\*"制",2\*/

{0x08,0x08,0x88,0xFF,0x48,0x48,0x00,0x08,0x48,0xFF,0x08,0x08,0xF8,0x00,0x00,0x00},

{0x81,0x65,0x08,0x07,0x20,0xC0,0x08,0x04,0x23,0xC0,0x03,0x00,0x23,0xC4,0x0F,0x00},/\*"热",3\*/

{0x40,0x40,0x42,0xCC,0x00,0x88,0x89,0x8E,0x88,0xF8,0x88,0x8C,0x8B,0x88,0x80,0x00},

{0x00,0x40,0x20,0x1F,0x20,0x40,0x50,0x48,0x46,0x41,0x42,0x44,0x58,0x40,0x40,0x00},/\*"送",4\*/

{0x00,0x00,0xFE,0x02,0x12,0x22,0xC2,0x02,0xC2,0x32,0x02,0xFE,0x00,0x00,0x00,0x00},

{0x80,0x60,0x1F,0x00,0x20,0x10,0x0C,0x03,0x0C,0x30,0x00,0x0F,0x30,0x40,0xF8,0x00},/\*"风",5\*/

{0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x00},

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/\*"一",6\*/

{0x10,0x10,0xD0,0xFF,0x90,0x10,0x40,0x42,0x5C,0x40,0x7F,0x40,0x50,0xCE,0x00,0x00},

{0x04,0x03,0x00,0xFF,0x00,0x01,0x40,0x44,0x44,0x44,0x44,0x44,0x44,0xFF,0x00,0x00},/\*"档",7\*/

{0x00,0x00,0x08,0x08,0x08,0x08,0x08,0x08,0x08,0x08,0x08,0x08,0x08,0x00,0x00,0x00},

{0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x00},/\*"二",8\*/

{0x10,0x10,0xD0,0xFF,0x90,0x10,0x40,0x42,0x5C,0x40,0x7F,0x40,0x50,0xCE,0x00,0x00},

{0x04,0x03,0x00,0xFF,0x00,0x01,0x40,0x44,0x44,0x44,0x44,0x44,0x44,0xFF,0x00,0x00},/\*"档",9\*/

{0x00,0x04,0x84,0x84,0x84,0x84,0x84,0x84,0x84,0x84,0x84,0x84,0x84,0x04,0x00,0x00},

{0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x00},/\*"三",10\*/

{0x10,0x10,0xD0,0xFF,0x90,0x10,0x40,0x42,0x5C,0x40,0x7F,0x40,0x50,0xCE,0x00,0x00},

{0x04,0x03,0x00,0xFF,0x00,0x01,0x40,0x44,0x44,0x44,0x44,0x44,0x44,0xFF,0x00,0x00},/\*"档",11\*/

{0x00,0x10,0x88,0xC4,0x33,0x40,0x48,0x48,0x48,0x7F,0x48,0xC8,0x48,0x48,0x40,0x00},

{0x02,0x01,0x00,0xFF,0x00,0x02,0x0A,0x32,0x02,0x42,0x82,0x7F,0x02,0x02,0x02,0x00},/\*"待",12\*/

{0x10,0x10,0xD0,0xFF,0x90,0x10,0x00,0xFE,0x02,0x02,0x02,0xFE,0x00,0x00,0x00,0x00},

{0x04,0x03,0x00,0xFF,0x00,0x83,0x60,0x1F,0x00,0x00,0x00,0x3F,0x40,0x40,0x78,0x00},/\*"机",13\*/

{0x06,0x09,0x09,0xE6,0xF8,0x0C,0x04,0x02,0x02,0x02,0x02,0x02,0x04,0x1E,0x00,0x00},

{0x00,0x00,0x00,0x07,0x1F,0x30,0x20,0x40,0x40,0x40,0x40,0x40,0x20,0x10,0x00,0x00},/\*"℃",0\*/

};

const uint8\_t code led\_asc2\_1608[95][16]={

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/\*" ",0\*/

{0x00,0x00,0x00,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x33,0x00,0x00,0x00,0x00},/\*"!",1\*/

{0x00,0x10,0x0C,0x02,0x10,0x0C,0x02,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/\*""",2\*/

{0x00,0x40,0xC0,0x78,0x40,0xC0,0x78,0x00,0x00,0x04,0x3F,0x04,0x04,0x3F,0x04,0x00},/\*"#",3\*/

{0x00,0x70,0x88,0x88,0xFC,0x08,0x30,0x00,0x00,0x18,0x20,0x20,0xFF,0x21,0x1E,0x00},/\*"$",4\*/

{0xF0,0x08,0xF0,0x80,0x60,0x18,0x00,0x00,0x00,0x31,0x0C,0x03,0x1E,0x21,0x1E,0x00},/\*"%",5\*/

{0x00,0xF0,0x08,0x88,0x70,0x00,0x00,0x00,0x1E,0x21,0x23,0x2C,0x19,0x27,0x21,0x10},/\*"&",6\*/

{0x00,0x12,0x0E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/\*"'",7\*/

{0x00,0x00,0x00,0xE0,0x18,0x04,0x02,0x00,0x00,0x00,0x00,0x07,0x18,0x20,0x40,0x00},/\*"(",8\*/

{0x00,0x02,0x04,0x18,0xE0,0x00,0x00,0x00,0x00,0x40,0x20,0x18,0x07,0x00,0x00,0x00},/\*")",9\*/

{0x40,0x40,0x80,0xF0,0x80,0x40,0x40,0x00,0x02,0x02,0x01,0x0F,0x01,0x02,0x02,0x00},/\*"\*",10\*/

{0x00,0x00,0x00,0x00,0xE0,0x00,0x00,0x00,0x00,0x01,0x01,0x01,0x0F,0x01,0x01,0x01},/\*"+",11\*/

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x90,0x70,0x00,0x00,0x00,0x00,0x00},/\*",",12\*/

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x01,0x01,0x01,0x01,0x01,0x00},/\*"-",13\*/

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x30,0x30,0x00,0x00,0x00,0x00,0x00},/\*".",14\*/

{0x00,0x00,0x00,0x00,0xC0,0x38,0x04,0x00,0x00,0x60,0x18,0x07,0x00,0x00,0x00,0x00},/\*"/",15\*/

{0x00,0xE0,0x10,0x08,0x08,0x10,0xE0,0x00,0x00,0x0F,0x10,0x20,0x20,0x10,0x0F,0x00},/\*"0",16\*/

{0x00,0x00,0x10,0x10,0xF8,0x00,0x00,0x00,0x00,0x00,0x20,0x20,0x3F,0x20,0x20,0x00},/\*"1",17\*/

{0x00,0x70,0x08,0x08,0x08,0x08,0xF0,0x00,0x00,0x30,0x28,0x24,0x22,0x21,0x30,0x00},/\*"2",18\*/

{0x00,0x30,0x08,0x08,0x08,0x88,0x70,0x00,0x00,0x18,0x20,0x21,0x21,0x22,0x1C,0x00},/\*"3",19\*/

{0x00,0x00,0x80,0x40,0x30,0xF8,0x00,0x00,0x00,0x06,0x05,0x24,0x24,0x3F,0x24,0x24},/\*"4",20\*/

{0x00,0xF8,0x88,0x88,0x88,0x08,0x08,0x00,0x00,0x19,0x20,0x20,0x20,0x11,0x0E,0x00},/\*"5",21\*/

{0x00,0xE0,0x10,0x88,0x88,0x90,0x00,0x00,0x00,0x0F,0x11,0x20,0x20,0x20,0x1F,0x00},/\*"6",22\*/

{0x00,0x18,0x08,0x08,0x88,0x68,0x18,0x00,0x00,0x00,0x00,0x3E,0x01,0x00,0x00,0x00},/\*"7",23\*/

{0x00,0x70,0x88,0x08,0x08,0x88,0x70,0x00,0x00,0x1C,0x22,0x21,0x21,0x22,0x1C,0x00},/\*"8",24\*/

{0x00,0xF0,0x08,0x08,0x08,0x10,0xE0,0x00,0x00,0x01,0x12,0x22,0x22,0x11,0x0F,0x00},/\*"9",25\*/

{0x00,0x00,0x00,0xC0,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x30,0x30,0x00,0x00,0x00},/\*":",26\*/

{0x00,0x00,0x00,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xE0,0x00,0x00,0x00,0x00},/\*";",27\*/

{0x00,0x00,0x80,0x40,0x20,0x10,0x08,0x00,0x00,0x01,0x02,0x04,0x08,0x10,0x20,0x00},/\*"<",28\*/

{0x00,0x40,0x40,0x40,0x40,0x40,0x40,0x00,0x00,0x02,0x02,0x02,0x02,0x02,0x02,0x00},/\*"=",29\*/

{0x00,0x08,0x10,0x20,0x40,0x80,0x00,0x00,0x00,0x20,0x10,0x08,0x04,0x02,0x01,0x00},/\*">",30\*/

{0x00,0x70,0x48,0x08,0x08,0x88,0x70,0x00,0x00,0x00,0x00,0x30,0x37,0x00,0x00,0x00},/\*"?",31\*/

{0xC0,0x30,0xC8,0x28,0xE8,0x10,0xE0,0x00,0x07,0x18,0x27,0x28,0x2F,0x28,0x17,0x00},/\*"@",32\*/

{0x00,0x00,0xC0,0x38,0xE0,0x00,0x00,0x00,0x20,0x3C,0x23,0x02,0x02,0x27,0x38,0x20},/\*"A",33\*/

{0x08,0xF8,0x88,0x88,0x88,0x70,0x00,0x00,0x20,0x3F,0x20,0x20,0x20,0x11,0x0E,0x00},/\*"B",34\*/

{0xC0,0x30,0x08,0x08,0x08,0x08,0x38,0x00,0x07,0x18,0x20,0x20,0x20,0x10,0x08,0x00},/\*"C",35\*/

{0x08,0xF8,0x08,0x08,0x08,0x10,0xE0,0x00,0x20,0x3F,0x20,0x20,0x20,0x10,0x0F,0x00},/\*"D",36\*/

{0x08,0xF8,0x88,0x88,0xE8,0x08,0x10,0x00,0x20,0x3F,0x20,0x20,0x23,0x20,0x18,0x00},/\*"E",37\*/

{0x08,0xF8,0x88,0x88,0xE8,0x08,0x10,0x00,0x20,0x3F,0x20,0x00,0x03,0x00,0x00,0x00},/\*"F",38\*/

{0xC0,0x30,0x08,0x08,0x08,0x38,0x00,0x00,0x07,0x18,0x20,0x20,0x22,0x1E,0x02,0x00},/\*"G",39\*/

{0x08,0xF8,0x08,0x00,0x00,0x08,0xF8,0x08,0x20,0x3F,0x21,0x01,0x01,0x21,0x3F,0x20},/\*"H",40\*/

{0x00,0x08,0x08,0xF8,0x08,0x08,0x00,0x00,0x00,0x20,0x20,0x3F,0x20,0x20,0x00,0x00},/\*"I",41\*/

{0x00,0x00,0x08,0x08,0xF8,0x08,0x08,0x00,0xC0,0x80,0x80,0x80,0x7F,0x00,0x00,0x00},/\*"J",42\*/

{0x08,0xF8,0x88,0xC0,0x28,0x18,0x08,0x00,0x20,0x3F,0x20,0x01,0x26,0x38,0x20,0x00},/\*"K",43\*/

{0x08,0xF8,0x08,0x00,0x00,0x00,0x00,0x00,0x20,0x3F,0x20,0x20,0x20,0x20,0x30,0x00},/\*"L",44\*/

{0x08,0xF8,0xF8,0x00,0xF8,0xF8,0x08,0x00,0x20,0x3F,0x01,0x3E,0x01,0x3F,0x20,0x00},/\*"M",45\*/

{0x08,0xF8,0x30,0xC0,0x00,0x08,0xF8,0x08,0x20,0x3F,0x20,0x00,0x07,0x18,0x3F,0x00},/\*"N",46\*/

{0xE0,0x10,0x08,0x08,0x08,0x10,0xE0,0x00,0x0F,0x10,0x20,0x20,0x20,0x10,0x0F,0x00},/\*"O",47\*/

{0x08,0xF8,0x08,0x08,0x08,0x08,0xF0,0x00,0x20,0x3F,0x21,0x01,0x01,0x01,0x00,0x00},/\*"P",48\*/

{0xE0,0x10,0x08,0x08,0x08,0x10,0xE0,0x00,0x0F,0x10,0x28,0x28,0x30,0x50,0x4F,0x00},/\*"Q",49\*/

{0x08,0xF8,0x88,0x88,0x88,0x88,0x70,0x00,0x20,0x3F,0x20,0x00,0x03,0x0C,0x30,0x20},/\*"R",50\*/

{0x00,0x70,0x88,0x08,0x08,0x08,0x38,0x00,0x00,0x38,0x20,0x21,0x21,0x22,0x1C,0x00},/\*"S",51\*/

{0x18,0x08,0x08,0xF8,0x08,0x08,0x18,0x00,0x00,0x00,0x20,0x3F,0x20,0x00,0x00,0x00},/\*"T",52\*/

{0x08,0xF8,0x08,0x00,0x00,0x08,0xF8,0x08,0x00,0x1F,0x20,0x20,0x20,0x20,0x1F,0x00},/\*"U",53\*/

{0x08,0x78,0x88,0x00,0x00,0xC8,0x38,0x08,0x00,0x00,0x07,0x38,0x0E,0x01,0x00,0x00},/\*"V",54\*/

{0x08,0xF8,0x00,0xF8,0x00,0xF8,0x08,0x00,0x00,0x03,0x3E,0x01,0x3E,0x03,0x00,0x00},/\*"W",55\*/

{0x08,0x18,0x68,0x80,0x80,0x68,0x18,0x08,0x20,0x30,0x2C,0x03,0x03,0x2C,0x30,0x20},/\*"X",56\*/

{0x08,0x38,0xC8,0x00,0xC8,0x38,0x08,0x00,0x00,0x00,0x20,0x3F,0x20,0x00,0x00,0x00},/\*"Y",57\*/

{0x10,0x08,0x08,0x08,0xC8,0x38,0x08,0x00,0x20,0x38,0x26,0x21,0x20,0x20,0x18,0x00},/\*"Z",58\*/

{0x00,0x00,0x00,0xFE,0x02,0x02,0x02,0x00,0x00,0x00,0x00,0x7F,0x40,0x40,0x40,0x00},/\*"[",59\*/

{0x00,0x04,0x38,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x06,0x38,0xC0,0x00},/\*"\",60\*/

{0x00,0x02,0x02,0x02,0xFE,0x00,0x00,0x00,0x00,0x40,0x40,0x40,0x7F,0x00,0x00,0x00},/\*"]",61\*/

{0x00,0x00,0x04,0x02,0x02,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/\*"^",62\*/

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80},/\*"\_",63\*/

{0x00,0x02,0x02,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/\*"`",64\*/

{0x00,0x00,0x80,0x80,0x80,0x00,0x00,0x00,0x00,0x19,0x24,0x24,0x12,0x3F,0x20,0x00},/\*"a",65\*/

{0x10,0xF0,0x00,0x80,0x80,0x00,0x00,0x00,0x00,0x3F,0x11,0x20,0x20,0x11,0x0E,0x00},/\*"b",66\*/

{0x00,0x00,0x00,0x80,0x80,0x80,0x00,0x00,0x00,0x0E,0x11,0x20,0x20,0x20,0x11,0x00},/\*"c",67\*/

{0x00,0x00,0x80,0x80,0x80,0x90,0xF0,0x00,0x00,0x1F,0x20,0x20,0x20,0x10,0x3F,0x20},/\*"d",68\*/

{0x00,0x00,0x80,0x80,0x80,0x80,0x00,0x00,0x00,0x1F,0x24,0x24,0x24,0x24,0x17,0x00},/\*"e",69\*/

{0x00,0x80,0x80,0xE0,0x90,0x90,0x20,0x00,0x00,0x20,0x20,0x3F,0x20,0x20,0x00,0x00},/\*"f",70\*/

{0x00,0x00,0x80,0x80,0x80,0x80,0x80,0x00,0x00,0x6B,0x94,0x94,0x94,0x93,0x60,0x00},/\*"g",71\*/

{0x10,0xF0,0x00,0x80,0x80,0x80,0x00,0x00,0x20,0x3F,0x21,0x00,0x00,0x20,0x3F,0x20},/\*"h",72\*/

{0x00,0x80,0x98,0x98,0x00,0x00,0x00,0x00,0x00,0x20,0x20,0x3F,0x20,0x20,0x00,0x00},/\*"i",73\*/

{0x00,0x00,0x00,0x80,0x98,0x98,0x00,0x00,0x00,0xC0,0x80,0x80,0x80,0x7F,0x00,0x00},/\*"j",74\*/

{0x10,0xF0,0x00,0x00,0x80,0x80,0x80,0x00,0x20,0x3F,0x24,0x06,0x29,0x30,0x20,0x00},/\*"k",75\*/

{0x00,0x10,0x10,0xF8,0x00,0x00,0x00,0x00,0x00,0x20,0x20,0x3F,0x20,0x20,0x00,0x00},/\*"l",76\*/

{0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x00,0x20,0x3F,0x20,0x00,0x3F,0x20,0x00,0x3F},/\*"m",77\*/

{0x80,0x80,0x00,0x80,0x80,0x80,0x00,0x00,0x20,0x3F,0x21,0x00,0x00,0x20,0x3F,0x20},/\*"n",78\*/

{0x00,0x00,0x80,0x80,0x80,0x80,0x00,0x00,0x00,0x1F,0x20,0x20,0x20,0x20,0x1F,0x00},/\*"o",79\*/

{0x80,0x80,0x00,0x80,0x80,0x00,0x00,0x00,0x80,0xFF,0x91,0x20,0x20,0x11,0x0E,0x00},/\*"p",80\*/

{0x00,0x00,0x00,0x80,0x80,0x00,0x80,0x00,0x00,0x0E,0x11,0x20,0x20,0x91,0xFF,0x80},/\*"q",81\*/

{0x80,0x80,0x80,0x00,0x80,0x80,0x80,0x00,0x20,0x20,0x3F,0x21,0x20,0x00,0x01,0x00},/\*"r",82\*/

{0x00,0x00,0x80,0x80,0x80,0x80,0x80,0x00,0x00,0x33,0x24,0x24,0x24,0x24,0x19,0x00},/\*"s",83\*/

{0x00,0x80,0x80,0xE0,0x80,0x80,0x00,0x00,0x00,0x00,0x00,0x1F,0x20,0x20,0x10,0x00},/\*"t",84\*/

{0x80,0x80,0x00,0x00,0x00,0x80,0x80,0x00,0x00,0x1F,0x20,0x20,0x20,0x10,0x3F,0x20},/\*"u",85\*/

{0x80,0x80,0x80,0x00,0x80,0x80,0x80,0x00,0x00,0x03,0x0C,0x30,0x0C,0x03,0x00,0x00},/\*"v",86\*/

{0x80,0x80,0x00,0x80,0x80,0x00,0x80,0x80,0x01,0x0E,0x30,0x0C,0x07,0x38,0x06,0x01},/\*"w",87\*/

{0x00,0x80,0x80,0x80,0x00,0x80,0x80,0x00,0x00,0x20,0x31,0x0E,0x2E,0x31,0x20,0x00},/\*"x",88\*/

{0x80,0x80,0x80,0x00,0x00,0x80,0x80,0x80,0x00,0x81,0x86,0x78,0x18,0x06,0x01,0x00},/\*"y",89\*/

{0x00,0x80,0x80,0x80,0x80,0x80,0x80,0x00,0x00,0x21,0x30,0x2C,0x22,0x21,0x30,0x00},/\*"z",90\*/

{0x00,0x00,0x00,0x00,0x00,0xFC,0x02,0x02,0x00,0x00,0x00,0x00,0x01,0x3E,0x40,0x40},/\*"{",91\*/

{0x00,0x00,0x00,0x00,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0x00,0x00,0x00},/\*"|",92\*/

{0x02,0x02,0xFC,0x00,0x00,0x00,0x00,0x00,0x40,0x40,0x3E,0x01,0x00,0x00,0x00,0x00},/\*"}",93\*/

{0x00,0x02,0x01,0x02,0x02,0x04,0x02,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/\*"~",94\*/

};

#endif

#ifndef \_AMPIRE12864\_H

#define \_AMPIRE12864\_H

#include "app.h"

#define LcdDataPort P0 //数据端口

sbit Reset = P2^3; //复位

sbit RS = P2^4; //指令数据选择

sbit E = P2^6; //指令数据控制

sbit CS1 = P3^2; //左屏幕选择，低电平有效

sbit CS2 = P3^3; //右屏幕选择

sbit RW = P2^5; //读写控制

sbit busy = P0^7; //忙标志

void SetOnOff(uint8\_t onoff); //开关显示

void SelectScreen(uint8\_t screen);//选择屏幕

void ClearScreen(uint8\_t screen); //清屏

void Show1616(uint8\_t lin,uint8\_t colum,uint8\_t \*address);//显示一个汉字

void CheckState(); //判断状态

void LcdDelay(uint16\_t time); //延时

void WriteData(uint8\_t dat); //写数据

void SendCommand(uint8\_t command); //写指令

void SetLine(uint8\_t line); //置行地址

void SetColum(uint8\_t colum);//置列地址

void SetStartLine(uint8\_t startline);//置显示起始行

void InitLcd(); //初始化

void ResetLcd(); //复位

void Show1608(uint8\_t lin,uint8\_t column,uint8\_t \*address);

void LCD\_ShowChinaLanguage(uint8\_t lin,uint8\_t column,uint8\_t \*p, uint8\_t Language\_num);

void SetClear(void);

//在指定位置显示一个字符,包括部分字符

//x:0~127

//y:0~63

void LCD\_ShowChar(uint8\_t lin,uint8\_t column,uint8\_t chr);

//m^n函数

unsigned long LCD\_pow(uint8\_t m,uint8\_t n);

//显示2个数字

//x,y :起点坐标

//len :数字的位数

//num:数值(0~4294967295);

void LCD\_ShowNumber(uint8\_t lin,uint8\_t column,unsigned long num,uint8\_t len);

//显示字符串

//x,y:起点坐标

//\*p:字符串起始地址

//用16字体

void LCD\_ShowString(uint8\_t lin,uint8\_t column,const uint8\_t \*p);

void LCD\_ShowString\_length(uint8\_t lin,uint8\_t column, uint8\_t \*p, uint8\_t length);

#endif