/\*本16\*16点阵带字的为上\*/

#include<reg52.h>

#define uint unsigned int

#define u16 unsigned int

#define uchar unsigned char

#define u8 unsigned char

sbit H\_stcp = P2^6; //输出时钟线

sbit H\_shcp = P2^7; //HC595芯片输入时钟线

sbit H\_ds = P2^5; //数据线

sbit HC138\_A0 = P2^0; //

sbit HC138\_A1 = P2^1; //

sbit HC138\_A2 = P2^2; //

sbit HC138\_A3 = P2^3; //

#define WORDNUM 40

u16 volatile W\_BCNT; //显示字标志

u8 volatile W\_LCNT; //显示位移标志

u8 speed=5; //移动速度

// 春(0) 风(1) 得(2) 意(3) ，(4) 大(5) 展(6) 宏(7) 图(8) ，(9) 人(10) 杰(11) 地(12) 灵(13) ，(14) 前(15)

// 程(16) 万(17) 里(18) ，(19) 出(20) 类(21) 拔(22) 萃(23) ，(24) 鹏(25) 程(26) 万(27) 里(28) ，(29) 寿(30) 与(31)

// 天(32) 齐(33) ，(34) 缘(35) 定(36) 三(37) 生(38) 。(39)

unsigned char code Word[WORDNUM][32] = {

{

0x02,0x22,0x2A,0x2A,0x2B,0x2B,0x2A,0xFA,0xFE,0x2E,0x2B,0x2B,0x2A,0x2A,0x22,0x02,

0x20,0x20,0x60,0xC0,0xFF,0x7F,0x49,0x49,0x49,0x49,0x7F,0xFF,0xC0,0x60,0x20,0x20,/\*"春",0\*/

},

{

0x00,0x00,0x00,0x7F,0x7F,0x4C,0x4F,0x43,0x43,0x47,0x4C,0x48,0x7F,0x7F,0x00,0x00,

0x1F,0x1F,0x0E,0xFC,0xF0,0x0C,0x3C,0xF0,0xF0,0x38,0x0C,0x04,0xF8,0xFE,0x07,0x01,/\*"风",1\*/

},

{

0x00,0x01,0x7D,0x7D,0x55,0x55,0x55,0x55,0x7D,0x7D,0xCC,0xEF,0x33,0x19,0x08,0x00,

0x40,0x40,0x40,0xFE,0xFF,0x43,0x42,0x48,0x58,0x50,0x40,0xFF,0xFF,0x80,0xC0,0x40,/\*"得",2\*/

},

{

0x08,0x08,0x48,0x4B,0x6B,0x7A,0x5A,0xCA,0xCA,0x5A,0x7A,0x6B,0x4B,0x48,0x08,0x08,

0x06,0x0E,0x08,0xE7,0xE7,0xA1,0xA1,0xAD,0xBD,0xB1,0xA1,0xEF,0xEE,0x0C,0x0E,0x02,/\*"意",3\*/

},

{

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1C,0x1E,0x1A,0x00,0x00,/\*"，",4\*/

},

{

0x04,0x04,0x04,0x04,0x04,0x04,0x04,0xFF,0xFF,0x04,0x04,0x04,0x04,0x04,0x04,0x04,

0x01,0x01,0x03,0x06,0x0C,0x38,0xF0,0xC0,0xC0,0xF0,0x38,0x0C,0x06,0x03,0x01,0x01,/\*"大",5\*/

},

{

0x00,0x78,0x79,0x49,0x4F,0x4F,0x49,0x49,0x4F,0x4F,0x49,0x49,0x7F,0x7F,0x00,0x00,

0x22,0x22,0x32,0x3E,0xEC,0xEC,0x38,0x34,0xE6,0xFF,0x3F,0x20,0xF8,0xFE,0x07,0x01,/\*"展",6\*/

},

{

0x30,0x3A,0x2A,0x22,0x22,0x22,0x22,0x62,0xEE,0xAF,0x23,0x22,0x22,0x32,0x3A,0x08,

0x00,0x00,0x03,0x07,0x0E,0x0A,0xC2,0xF2,0x3A,0x8F,0xE7,0x72,0x18,0x0C,0x06,0x02,/\*"宏",7\*/

},

{

0x00,0x7F,0x7F,0x40,0x48,0x4C,0x4E,0x4B,0x7B,0x7E,0x4C,0x46,0x42,0x7F,0x7F,0x00,

0x00,0xFF,0xFF,0x42,0x42,0xC2,0x96,0xB6,0xAE,0xAA,0xCA,0x42,0x42,0xFF,0xFF,0x00,/\*"图",8\*/

},

{

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1C,0x1E,0x1A,0x00,0x00,/\*"，",9\*/

},

{

0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xFF,0xFF,0x03,0x00,0x00,0x00,0x00,0x00,0x00,

0x01,0x03,0x06,0x0C,0x38,0xF0,0xC0,0x00,0x00,0xC0,0xF0,0x38,0x0C,0x06,0x03,0x01,/\*"人",10\*/

},

{

0x00,0x10,0x10,0x11,0x13,0x16,0x1C,0xFF,0xFF,0x1C,0x16,0x13,0x11,0x10,0x10,0x00,

0x40,0x47,0xCF,0x88,0x00,0x06,0x0E,0xE8,0xE6,0x0E,0x08,0x00,0x8C,0xCE,0x43,0x41,/\*"杰",11\*/

},

{

0x00,0x0F,0x0F,0x0C,0xFF,0xFF,0x03,0x1F,0x1F,0x05,0x04,0xFF,0xFF,0x04,0x04,0x04,

0x1E,0xDE,0xE2,0x62,0xFA,0xFA,0x02,0xFE,0xFC,0x10,0x10,0xF0,0xF8,0x0C,0x0C,0x08,/\*"地",12\*/

},

{

0x00,0x00,0x7F,0x7F,0x49,0x49,0x49,0x49,0x49,0x49,0x49,0x49,0x49,0x41,0x00,0x00,

0x01,0x01,0x03,0x62,0x76,0x1C,0x18,0xF0,0xF0,0x18,0x0C,0x66,0x72,0x13,0x01,0x01,/\*"灵",13\*/

},

{

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1C,0x1E,0x1A,0x00,0x00,/\*"，",14\*/

},

{

0x10,0x10,0x17,0xD7,0xF0,0x33,0x13,0x10,0x17,0x17,0x74,0xF4,0x97,0x17,0x10,0x10,

0x00,0x00,0xFE,0xFF,0x03,0xF2,0xF0,0x00,0xFE,0xFF,0x93,0x92,0xFF,0xFF,0x00,0x00,/\*"前",15\*/

},

{

0x00,0x7C,0x7C,0x44,0x44,0x44,0x44,0x7C,0x7C,0x44,0xC4,0xFF,0x7F,0x25,0x24,0x24,

0x82,0x92,0x92,0x92,0xFE,0xFE,0x92,0x92,0x92,0x62,0xE0,0xFF,0xFF,0xE0,0x70,0x10,/\*"程",16\*/

},

{

0x20,0x20,0x20,0x23,0x23,0x22,0x22,0x22,0x22,0x3F,0x3F,0x20,0x20,0x20,0x20,0x20,

0x00,0x00,0x00,0xFC,0xFE,0x03,0x03,0x02,0x00,0x80,0xE0,0x78,0x1C,0x06,0x03,0x01,/\*"万",17\*/

},

{

0x00,0x00,0x7F,0x7F,0x49,0x49,0x49,0x7F,0x7F,0x49,0x49,0x49,0x7F,0x7F,0x00,0x00,

0x02,0x02,0x22,0x22,0x22,0x22,0x22,0xFE,0xFE,0x22,0x22,0x22,0x22,0x22,0x02,0x02,/\*"里",18\*/

},

{

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1C,0x1E,0x1A,0x00,0x00,/\*"，",19\*/

},

{

0x00,0x00,0x3F,0x3F,0x02,0x02,0x02,0xFF,0xFF,0x02,0x02,0x02,0x3E,0x3E,0x00,0x00,

0x00,0x3F,0x3F,0x02,0x02,0x02,0x02,0xFE,0xFE,0x02,0x02,0x02,0x02,0x3E,0x3E,0x00,/\*"出",20\*/

},

{

0x00,0x08,0x08,0x49,0x6B,0x2E,0x0C,0xFF,0xFF,0x0C,0x2E,0x6B,0x49,0x08,0x08,0x00,

0x21,0x21,0xA3,0xA2,0x26,0x2C,0x38,0x70,0x70,0x38,0x2C,0x26,0xA2,0xA3,0xA1,0x21,/\*"类",21\*/

},

{

0x08,0x09,0x69,0xE9,0x89,0xF9,0xFF,0x0F,0x08,0x0A,0x0B,0xFF,0xFF,0x08,0x08,0x08,

0x01,0xC3,0xF6,0x3C,0x3C,0xF6,0xC3,0xF1,0xFD,0x0E,0x02,0xFE,0xFF,0xC3,0x42,0x40,/\*"拔",22\*/

},

{

0x20,0x24,0x24,0x24,0xF7,0xF7,0x24,0x2C,0x3C,0x34,0xF7,0xF7,0x24,0x24,0x24,0x20,

0x10,0x10,0x50,0xD0,0x90,0x90,0xD0,0x7F,0x7F,0xD0,0x90,0x90,0xD0,0x50,0x10,0x10,/\*"萃",23\*/

},

{

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1C,0x1E,0x1A,0x00,0x00,/\*"，",24\*/

},

{

0x00,0x00,0x3F,0x21,0xA4,0x68,0x3F,0x00,0x7F,0x44,0x7F,0x00,0x7F,0x44,0x7F,0x00,

0x00,0x7E,0x41,0x4A,0x48,0x48,0xC8,0x00,0xFF,0x41,0xFC,0x02,0xFD,0x44,0xF8,0x06,/\*"鹏",25\*/

},

{

0x00,0x7C,0x7C,0x44,0x44,0x44,0x44,0x7C,0x7C,0x44,0xC4,0xFF,0x7F,0x25,0x24,0x24,

0x82,0x92,0x92,0x92,0xFE,0xFE,0x92,0x92,0x92,0x62,0xE0,0xFF,0xFF,0xE0,0x70,0x10,/\*"程",26\*/

},

{

0x20,0x20,0x20,0x23,0x23,0x22,0x22,0x22,0x22,0x3F,0x3F,0x20,0x20,0x20,0x20,0x20,

0x00,0x00,0x00,0xFC,0xFE,0x03,0x03,0x02,0x00,0x80,0xE0,0x78,0x1C,0x06,0x03,0x01,/\*"万",27\*/

},

{

0x00,0x00,0x7F,0x7F,0x49,0x49,0x49,0x7F,0x7F,0x49,0x49,0x49,0x7F,0x7F,0x00,0x00,

0x02,0x02,0x22,0x22,0x22,0x22,0x22,0xFE,0xFE,0x22,0x22,0x22,0x22,0x22,0x02,0x02,/\*"里",28\*/

},

{

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1C,0x1E,0x1A,0x00,0x00,/\*"，",29\*/

},

{

0x02,0x22,0x2A,0x2A,0x2B,0x2B,0x2A,0x2A,0xFA,0xFE,0x2F,0x2B,0x2A,0x2A,0x22,0x02,

0x00,0x40,0x40,0x40,0xFE,0xFF,0x43,0x42,0x4C,0x5C,0xD0,0xE0,0x70,0x18,0x0C,0x04,/\*"寿",30\*/

},

{

0x00,0x10,0x11,0x11,0x11,0x11,0x11,0x11,0x11,0x11,0x11,0xF9,0xFF,0x07,0x00,0x00,

0x00,0x00,0xFC,0xFE,0x03,0x13,0x12,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10,/\*"与",31\*/

},

{

0x02,0x02,0x42,0x42,0x42,0x42,0x42,0x7F,0x7F,0x42,0x42,0x42,0x42,0x42,0x02,0x02,

0x01,0x01,0x03,0x06,0x0C,0x38,0xF0,0xC0,0xC0,0xF0,0x38,0x0C,0x06,0x03,0x01,0x01,/\*"天",32\*/

},

{

0x00,0x20,0x21,0x21,0x33,0x3A,0x2E,0x64,0xE4,0xAE,0x3A,0x33,0x21,0x21,0x20,0x00,

0x80,0x80,0x80,0x00,0xFF,0xFF,0x00,0x00,0x00,0x00,0xF8,0xFE,0x07,0x81,0x80,0x80,/\*"齐",33\*/

},

{

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1C,0x1E,0x1A,0x00,0x00,/\*"，",34\*/

},

{

0x05,0x65,0x7C,0x5C,0x55,0x57,0xD7,0xF5,0x34,0x04,0x0C,0xCE,0xF7,0x3D,0x0C,0x04,

0x08,0xB8,0xF0,0xFE,0xFF,0xDB,0x6E,0xB6,0x92,0x92,0x48,0x48,0x4C,0xE6,0xE6,0x44,/\*"缘",35\*/

},

{

0x30,0x38,0x2A,0x22,0x22,0x22,0x22,0x63,0xE3,0xA2,0x22,0x22,0x22,0x32,0x38,0x08,

0x02,0x02,0x22,0x22,0x22,0x22,0x22,0xFE,0xFE,0x02,0x06,0x7C,0x7C,0x06,0x03,0x01,/\*"定",36\*/

},

{

0x00,0x20,0x21,0x21,0x21,0x21,0x21,0x21,0x21,0x21,0x21,0x21,0x21,0x21,0x20,0x00,

0x04,0x04,0x04,0x04,0x04,0x04,0x04,0x04,0x04,0x04,0x04,0x04,0x04,0x04,0x04,0x04,/\*"三",37\*/

},

{

0x00,0x08,0x08,0x08,0x08,0x08,0x08,0xFF,0xFF,0x08,0x08,0x78,0x7C,0x0E,0x03,0x01,

0x02,0x02,0x42,0x42,0x42,0x42,0x42,0xFE,0xFE,0x42,0x42,0x42,0x42,0x42,0x02,0x02,/\*"生",38\*/

},

{

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x18,0x3C,0x24,0x3C,0x18,0x00,0x00,/\*"。",39\*/

},

};

unsigned char WordTemp[32] = {

0x01,0x01,0x61,0x71,0x19,0x0D,0x05,0xFF,0xFF,0x01,0x7D,0x7C,0x44,0x47,0x47,0x40,

0x02,0x06,0x0C,0x18,0x70,0xE4,0x86,0xFF,0xFF,0x00,0x7C,0x7E,0x43,0xC3,0xC2,0x00,/\*"张",0\*/

};

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

//延时函数

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

void delay\_ms(uint xms)

{

uint i, j;

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

for(j = 118; j > 0; j--);

}

#define HC595\_QA 0x80

#define HC595\_QB 0x20

#define HC595\_QC 0x04

#define HC595\_QD 0x01

#define HC595\_QE 0x10

#define HC595\_QF 0x40

#define HC595\_QG 0x02

#define HC595\_QH 0x08

void sendbyte\_All(u8 group1,u8 group2) //利用HC595输出

{

u8 z;

for(z = 0; z < 8; z++) //循环8次移入数据

{

H\_shcp = 0;

switch(z)

{

case 0:H\_ds = group2 & HC595\_QH;break;

case 1:H\_ds = group2 & HC595\_QG;break;

case 2:H\_ds = group2 & HC595\_QF;break;

case 3:H\_ds = group2 & HC595\_QE;break;

case 4:H\_ds = group2 & HC595\_QD;break;

case 5:H\_ds = group2 & HC595\_QC;break;

case 6:H\_ds = group2 & HC595\_QB;break;

case 7:H\_ds = group2 & HC595\_QA;break;

}

H\_shcp = 1; //上升沿输入数据

}

for(z = 0; z < 8; z++) //循环8次移入数据

{

H\_shcp = 0;

switch(z)

{

case 0:H\_ds = group1 & HC595\_QH;break;

case 1:H\_ds = group1 & HC595\_QG;break;

case 2:H\_ds = group1 & HC595\_QF;break;

case 3:H\_ds = group1 & HC595\_QE;break;

case 4:H\_ds = group1 & HC595\_QD;break;

case 5:H\_ds = group1 & HC595\_QC;break;

case 6:H\_ds = group1 & HC595\_QB;break;

case 7:H\_ds = group1 & HC595\_QA;break;

}

H\_shcp = 1; //上升沿输入数据

}

H\_stcp = 0;

H\_stcp = 1; //上升沿使数据并行输出

}

void HC138Out(u8 i,u8 l)

{

u8 j;

switch(i)

{

case 7:j=5;break;

case 6:j=2;break;

case 5:j=1;break;

case 4:j=7;break;

case 3:j=0;break;

case 2:j=6;break;

case 1:j=4;break;

case 0:j=3;break;

}

HC138\_A0 = j&0x01;

HC138\_A1 = j&0x02;

HC138\_A2 = j&0x04;

if(l)

HC138\_A3=0;

else

HC138\_A3=1;

}

void WordDisplay(u8 \*word)

{

uchar i;

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

{

sendbyte\_All(0,0); //消影

HC138Out(i,0);

sendbyte\_All(\*(word+i),\*(word+i+8)); //输出信息

}

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

{

sendbyte\_All(0,0); //消影

HC138Out(i,1);

sendbyte\_All(\*(word+i+16),\*(word+i+24)); //输出信息

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*定时器0、定时器1初始化\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void time0\_init()

{

TMOD = 0x01; //设置定时器0为工作模式1 (16-bit)

TL0 = 15536; //给定时器低位赋初始值 65536-15536=50000us=50ms,即50ms一个中断

TH0 = 15536 >> 8; //给定时器高位赋初始值

TR0 = 1; //开启定时器0

ET0 = 1; //允许定时器0中断

EA = 1; //开启全局中断

}

//按键

sbit Key1=P1^3; //设置键

sbit Key2=P1^4; //加按键

sbit Key3=P3^2; //减按键

sbit Key4=P3^3; //交通管制键

//========================================================================

// 函数: u8 Key\_Scan()

// 应用: temp=u8 Key\_Scan();

// 描述: 按键扫描并返回按下的键值

// 参数: NONE

// 返回: 按下的键值

// 版本: VER1.0

// 日期: 2015-05-29

// 备注: 该函数带松手检测,按下键返回一次键值后返回0,直至第二次按键按下

//========================================================================

u8 Key\_Scan()

{

static u8 key\_up=1;//按键按松开标志

if(key\_up&&(Key1==0||Key2==0||Key3==0||Key4==0))

{

delay\_ms(10);//去抖动

key\_up=0;

if(Key1==0) return 1;

else if(Key2==0)return 2;

else if(Key3==0)return 3;

else if(Key4==0)return 4;

}

else if(Key1==1&&Key2==1&&Key3==1&&Key4==1)

key\_up=1;

return 0;// 无按键按下

}

void main()

{

uchar key\_can; //按键值

uchar mode=0; //模式

uchar modej=0; //模式

uchar i;

time0\_init();

while(1)

{

key\_can=Key\_Scan();

if(1==key\_can)

{

TR0 = 1; //开启定时器0

if(modej) //判断暂停标志,此处处理是为了防止暂停再启动方向会变

{

modej=0;

mode++;

if(mode>=2)

{

mode=0;

}

}

mode++;

if(mode>=2)

{

mode=0;

}

}

if(2==key\_can)

{

TR0 = 0; //关闭定时器0,暂停

modej=1; //暂停标志

}

if(4==key\_can)

{

speed+=1; //速度加1

}

if(3==key\_can)

{

speed-=1; //速度减1

if(speed==0)

speed=1;

}

if(1==mode)//往右移

{

for(i = 0; i < 16-W\_LCNT; i++)

{

WordTemp[i]=Word[W\_BCNT][i+W\_LCNT];

};

for(i = 0; i <W\_LCNT ; i++)

{

WordTemp[i+16-W\_LCNT]=Word[W\_BCNT+1][i];

};

for(i = 16; i < 32-W\_LCNT; i++)

{

WordTemp[i]=Word[W\_BCNT][i+W\_LCNT];

};

for(i = 16; i <16+W\_LCNT ; i++)

{

WordTemp[i+16-W\_LCNT]=Word[W\_BCNT+1][i];

};

}

if(0==mode)//往左移

{

for(i = 0; i < 16-W\_LCNT; i++)

{

WordTemp[i+W\_LCNT]=Word[W\_BCNT][i];

};

for(i = 0; i < W\_LCNT ; i++)

{

WordTemp[i]=Word[W\_BCNT+1][16-W\_LCNT+i];

};

for(i = 16; i < 32-W\_LCNT; i++)

{

WordTemp[i+W\_LCNT]=Word[W\_BCNT][i];

};

for(i = 16; i <16+W\_LCNT ; i++)

{

WordTemp[i]=Word[W\_BCNT+1][16-W\_LCNT+i];

};

}

WordDisplay(WordTemp);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*定时器0中断服务程序\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void time0\_int() interrupt 1

{

static uchar volatile value; //定时10ms中断一次

TL0 = 15536; //给定时器低位重装初始值 65536-15536=50000us=50ms,即50ms一个中断

TH0 = 15536 >> 8; //给定时器高位重装初始值

value ++;

if(value >= speed)

{

value = 0;

W\_LCNT++;

if(W\_LCNT >= 16)

{

W\_LCNT=0;

W\_BCNT++;

if(W\_BCNT >= (WORDNUM-1))

{

W\_BCNT=0;

}

}

}

}