目录

[一、推挽输出端口初始化 1](#_Toc31976)

[二、BCD转HEX 1](#_Toc22815)

[三、十六进制转换为 BCD 1](#_Toc16998)

[四、 ADC转化（自带ADC模块的stc15） 1](#_Toc32257)

[五、 热敏电阻温度转化 2](#_Toc20814)

[六、 光敏电阻调光 2](#_Toc16314)

[六、多位数码管 3](#_Toc2925)

**一、推挽输出端口初始化**

P1M0 = 0xff;

P1M1 = 0x00;

P2M0 = 0xff;

P2M1 = 0x00;

P3M0 = 0xf8;

P3M1 = 0x00;

P5M0 = 0x10;

P5M1 = 0x00;

**二、BCD转HEX**

unsigned char bcd2hex(unsigned char bcd)

{

unsigned char temp;

temp = bcd & 0x0f;

bcd = bcd >> 4;

bcd %= 0xf;

return (bcd \* 10 + temp);

}

**三、十六进制转换为 BCD**

if ((alarm\_min & 0x0f) >= 0x0a)

alarm\_min = (alarm\_min & 0xf0) + 0x10;

1. **ADC转化（自带ADC模块的stc15）**

void enableAdc()

{

P1ASF = 0x03;

}

/\*\*

\* @brief 禁用 ADC 转换

\*/

void disableAdc()

{

P1ASF = 0x00;

}

/\*\*

\* @brief 读取 ADC 通道的值

\*

\* @param ch ADC 通道序号

\* @return 10 位 ADC 数值

\*/

unsigned int getAdcResult(unsigned char ch)

{

ADC\_CONTR = ADC\_POWER | ADC\_SPEEDLL | ch | ADC\_START;

\_nop\_(); // Must wait before inquiry

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_(); // Must wait before inquiry

\_nop\_();

while (!(ADC\_CONTR & ADC\_FLAG))

; // Wait complete flag

ADC\_CONTR &= ~ADC\_FLAG; // Close ADC

return (ADC\_RES \* 4 + ADC\_RESL); // Return ADC result.

}

//ADC的结果：a=getAdcResult();

1. 热敏电阻温度转化

unsigned int getTemperature(void)

{

adc\_result = getAdcResult(0);//通道序号

return (unsigned int)((3950.0 / (11.33657 + log(6.04 \* (float)adc\_result / (1024.0 - (float)adc\_result))) - 278.15) \* 100);

}

1. 光敏电阻调光

unsigned int getLight(void)

{

adc\_result = getAdcResult(1);

if (adc\_result < 100)

{

return 8;

}

if (adc\_result > 100 && adc\_result < 300)

{

return 10;

}

if (adc\_result > 300 && adc\_result < 500)

{

return 20;

}

if (adc\_result > 500 && adc\_result < 600)

{

return 40;

}

if (adc\_result > 600 && adc\_result < 700)

{

return 60;

}

if (adc\_result > 700 && adc\_result < 800)

{

return 90;

}

if (adc\_result > 800 && adc\_result < 900)

{

return 120;

}

if (adc\_result > 900)

{

return 150;

}

return 8;

}

if (light\_control == 1)

{

light=getLight();

}

else

{

light = 8;

}

LED=0；

delay(light);

七、多位数码管

#define SEG\_PORT P2//段选

sbit SEG\_COM1 = P3 ^ 3;

sbit SEG\_COM2 = P3 ^ 4;

sbit SEG\_COM3 = P3 ^ 5;

sbit SEG\_COM4 = P3 ^ 6;//位选

ENABLE=1;

DISABLE=0;

ALLOFF=0xff;

SEG\_COM1 = ENABLE;

SEG\_PORT = num\_seg\_sel[display\_array[0]];

delay(2);

SEG\_PORT = ALLOFF;

SEG\_COM1 = DISABLE;

// 第二位

// 判断冒号显示时最高位复位

SEG\_COM2 = ENABLE;

if (blink == 0)

SEG\_PORT = num\_seg\_sel[display\_array[1]] - 0x80;

if (blink == 1)

SEG\_PORT = num\_seg\_sel[display\_array[1]];

delay(2);

SEG\_PORT = ALLOFF;

SEG\_COM2 = DISABLE;

// 第三位

SEG\_COM3 = ENABLE;

SEG\_PORT = num\_seg\_sel[display\_array[2]];

delay(2);

SEG\_PORT = ALLOFF;

SEG\_COM3 = DISABLE;

// 第四位

SEG\_COM4 = ENABLE;

SEG\_PORT = num\_seg\_sel[display\_array[3]];

delay(2);

SEG\_PORT = ALLOFF;

SEG\_COM4 = DISABLE;

// 消隐 关闭所有显示

SEG\_COM1 = SEG\_COM2 = SEG\_COM3 = SEG\_COM4 = ENABLE;

SEG\_PORT = ALLOFF;

SEG\_COM1 = SEG\_COM2 = SEG\_COM3 = SEG\_COM4 = DISABLE;

八、PWM波最简形式

while(1)

{

PWM=1;

i=57;//0.5ms

while(i--);

PWM=0;

x=2280;//20ms

while(x--);

}

1、占空比=i/x+i

2、周期=i+x

3、频率=1/周期

应用：

1、舵机

2、电机调速

3、蜂鸣器发生

4、led呼吸灯

1. 串口发射程序

void fstwo()

{

TMOD=0x20;

SCON=0x40;

PCON=0x00;

TH1=0xfd;

TL1=0xfd;

TR1=1;

}

void send(uchar date)

{

SBUF=date;

while(TI==0);

TI=0;

}

1. 外部中断

主函数要调用中断系统启动

void uart() interrupt 4 //4就是外部中断

{

a=SBUF; //SBUF是外面进来的数据

RI=0;

}

比如说

void uart() interrupt 4 //4就是外部中断

{

a=SBUF; //SBUF是外面进来的数据

if(a==0x01)

{

LED=0;

}

RI=0;

}

1. 延时函数

void delay (uint z); //声明函数，写在main函数的前面

void delay (uint z)

{

uint x,y;

for(x=z;x>0;x--)

for(y=114;y>0;y--);//114是根据晶振周期算的

}//写在main函数的前面，写在后面的话要声明函数

就可以在main函数里用这个

delay(延时几毫秒);

1. 统一结构框架

#include <STC15X.h>

void main(void)

{

}

十三、51单片机音乐模板

#include <reg52.h>

sbit beep = P1^5;

unsigned char timer0H, timer0L, time;

code unsigned char FREQH[] = {

0xF2, 0xF3, 0xF5, 0xF5, 0xF6, 0xF7, 0xF8,

0xF9, 0xF9, 0xFA, 0xFA, 0xFB, 0xFB, 0xFC, 0xFC,

0xFC, 0xFD, 0xFD, 0xFD, 0xFD, 0xFE,

0xFE, 0xFE, 0xFE, 0xFE, 0xFE, 0xFE, 0xFF};

code unsigned char FREQL[] = {

0x42, 0xC1, 0x17, 0xB6, 0xD0, 0xD1, 0xB6,

0x21, 0xE1, 0x8C, 0xD8, 0x68, 0xE9, 0x5B, 0x8F,

0xEE, 0x44, 0x6B, 0xB4, 0xF4, 0x2D,

0x47, 0x77, 0xA2, 0xB6, 0xDA, 0xFA, 0x16};

code unsigned char song[] = {

6,1,2,6,1,1,3,2,1,2,2,2,1,2,2,7,1,2,7,1,1,7,1,1,2,2,2,1,2,1,7,1,1,

6,1,2,6,1,1,1,3,1,7,2,1,1,3,1,6,1,2,6,1,1,1,3,1,7,2,1,1,3,1,7,2,1,1,3,1

};

//一个音符有三个数字。前为第几个音、中为第几个八度、后为时长（以半拍为单位）。

//6, 2, 3 分别代表：６, 中音, ３个半拍;

//5, 2, 1 分别代表：５, 中音, １个半拍;

void delay(unsigned char t)

{

unsigned char t1;

unsigned long t2;

for(t1 = 0; t1 < t; t1++)

for(t2 = 0; t2 < 4000; t2++);

TR0 = 0;

}

void sing()

{

TH0= timer0H;

TL0 = timer0L;

TR0 = 1;

delay(time);

}

void main(void)

{

unsigned char k, i;

TMOD = 1;

ET0 = 1;

EA = 1;

while(1)

{

i = 0;

time = 1;

while(time)

{

k = song[i] + 7 \* song[i + 1] - 1;

timer0H = FREQH[k];

timer0L = FREQL[k];

time = song[i + 2];

i += 3;

if(i>69)

{

i=0;

}

sing();

}

}

}

void t0int() interrupt 1

{

TR0 = 0;

beep = !beep;

TH0 = timer0H;

TL0 = timer0L;

TR0 = 1;

}

//双线程

//必须有1个中断

//定时器中断控制音调（发声）

//延时函数控制一个音调的发声时长