**越狱程序PART2**

#include "STC12C5A60S2.h"

#include "intrins.h"

#include "stdio.h"

#include"uart.h"

#define uchar unsigned char

#define uint unsigned int

sbit OUT1 = P4^2; //输出IO口1， R6审讯室电磁锁

sbit OUT2 = P1^0; //输出2， R7医务室电磁锁

sbit OUT3 = P1^1; //输出3， 通风管的电磁锁

sbit OUT4 = P1^2; //输出4， R6审讯室照明灯

sbit OUT5 = P1^3; //输出5， 魔鬼图控制继电器

sbit OUT6 = P1^4; //输出6， 炸弹MP3音效控制器

sbit OUT7 = P1^5; //输出7，

sbit OUT8 = P1^6; //输出8，

sbit OUT9 = P1^7; //输出9，

sbit OUT10 = P4^3; //输出10，

sbit OUT11 = P3^2; //输出11，

sbit OUT12 = P3^3; //输出12，

sbit M10 = P2^6; //电缸1+

sbit M11 = P2^7; //电缸1-

sbit M20 = P3^4; //电缸2+

sbit M21 = P3^5; //电缸2-

sbit M30 = P3^6; //电缸3+

sbit M31 = P3^7; //电缸3-

sbit OUT21 = P4^0; //输出IO控制口21

sbit OUT22 = P2^0; //输出22

sbit OUT23 = P2^1; //输出23

sbit OUT24 = P2^2; //输出24

sbit OUT25 = P2^3; //输出25

sbit OUT26 = P2^4; //输出26

sbit OUT27 = P2^5; //输出27

sbit INPUT16 = P4^4; //输入IO控制口16， 触摸开关1

sbit INPUT15 = P4^5; //输入15， 触摸开关2

sbit INPUT14 = P4^1; //输入14 触摸开关3

sbit INPUT13 = P4^6; //输入13 触摸开关4

sbit INPUT12 = P0^7; //输入12 触摸开关5

sbit INPUT11 = P0^6; //输入11，按键类型 触摸开关6

sbit INPUT10 = P0^5; //输入10，门禁类型 炸弹感应霍尔开关

sbit INPUT9 = P0^4; //输入9，门禁 通风管门禁

sbit INPUT8 = P0^3; //输入8，门禁 人体感应开关

sbit INPUT7 = P0^2; //输入7，门禁 炸弹遥控器！

sbit INPUT6 = P0^1; //输入6，门禁 医务室的门禁

sbit INPUT5 = P0^0; //输入5，门禁类型

sbit RXD = P3^0;

sbit TXD = P3^1;

#define uchar unsigned char

#define uint unsigned int

sfr P4sw=0xbb;

void sys\_init(void);

void delay\_ms(int count);

void mp3\_init();

void mp3(unsigned char comd);

unsigned char Game\_Start=0,door1=0,door2=0,door3=0;

unsigned char flag\_boom=0,flag\_R6=0,flag\_devil=0,i=0,flag\_X=0;

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

函数名称： void main(void)

函数功能：

调用函数： void delay(unsigned int loop\_time)

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

void main()

{

sys\_init();

P4sw = 0x70;

Game\_Start=0;

flag\_boom=0;

flag\_R6=0;

flag\_devil=0;

flag\_X=0;

while(1)

{

if((INPUT10==0)&&(flag\_X==0))

{

mp3(0X33);

mp3(0X04);

delay\_ms(1000);

flag\_X=1;

}

else if((INPUT10==1)&&(flag\_X==1))

{

flag\_X=0;

delay\_ms(500);

}

if((INPUT7==0)&&(INPUT10==0)&&(flag\_R6==0)) //按下遥控器7，并且人体感应开关无感应8,并且炸弹放在指定位置

{

delay\_ms(50);

if((INPUT7==0)&&(INPUT10==0)&&(flag\_R6==0))

{

OUT6=1; //开启MP3，炸弹音效

for(i=0;i<20;i++) //延时一段时间后关房间的灯

{

delay\_ms(500);

}

OUT4=1; //R6的照明灯灭

OUT1=0; //R6的门打开

delay\_ms(500);

OUT6=0;

delay\_ms(500);

mp3(0X33);

mp3(0X01);

delay\_ms(3000);

flag\_R6=1; //R6通过！

}

}

if(INPUT6==0) //R7的门禁按下时

{

OUT5=1; //魔鬼图启动

flag\_devil=1;

}

if(flag\_devil==1)

{

if((INPUT13==1)&&(INPUT14==1))

{

OUT2=0;

mp3(0X33);

mp3(0X02);

delay\_ms(6000);

mp3(0X33);

mp3(0X03);

flag\_devil=0;

}

else if((INPUT12==1)&&(INPUT11==1))

{

OUT2=0;

mp3(0X33);

mp3(0X02);

delay\_ms(6000);

mp3(0X33);

mp3(0X03);

flag\_devil=0;

}

else if((INPUT16==1)&&(INPUT14==1))

{

OUT2=0;

mp3(0X33);

mp3(0X02);

delay\_ms(6000);

mp3(0X33);

mp3(0X03);

flag\_devil=0;

}

else if((INPUT16==1)&&(INPUT13==1))

{

OUT2=0;

mp3(0X33);

mp3(0X02);

delay\_ms(6000);

mp3(0X33);

mp3(0X03);

flag\_devil=0;

}

else if((INPUT16==1)&&(INPUT12==1))

{

OUT2=0;

mp3(0X33);

mp3(0X02);

delay\_ms(6000);

mp3(0X33);

mp3(0X03);

flag\_devil=0;

}

else if((INPUT16==1)&&(INPUT11==1))

{

OUT2=0;

mp3(0X33);

mp3(0X02);

delay\_ms(6000);

mp3(0X33);

mp3(0X03);

flag\_devil=0;

}

}

if(INPUT9==0) //通风管门禁

{

OUT3=0;

}

}

}

void sys\_init()

{

P0M1=0X00;

P0M0=0Xff; //设置P0为推挽输出

P1M1=0X00;

P1M0=0XFF; //设置P1为推挽输出

P2M1=0X00;

P2M0=0XFF; //设置P2口为推挽输出

P3M1=0X00;

P3M0=0XFC; //设置P3为推挽输出

P4M1=0X00;

P4M0=0XFF; //设置P4为推挽输出

//R6-R7-通风管 所有电磁锁吸合

OUT1=1; //

OUT2=1;

OUT3=1;

OUT4=0; // R6审讯室照明

OUT5=0; // 魔鬼图关闭

OUT6=0; // 炸弹MP3音效关闭

OUT7=1; // R4门禁打开

OUT8=0; // 报警灯1关闭

OUT9=0; // 报警灯2关闭

OUT10=0; // 报警灯3关闭

INPUT16=1; //

INPUT15=1; //

INPUT14=1;

INPUT13=1;

INPUT11=1; //6个触摸开关，触摸时返回高电平

INPUT12=1;

INPUT10=1; //霍尔开关

INPUT9=1; //通风管门禁

INPUT8=1; //人体感应开关

INPUT7=1; //炸弹遥控器

INPUT6=1; //医务室的门禁

INPUT5=1;

uart\_init(); //串口初始化

mp3\_init(); //MP3初始化 \*/

}

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

/\* \*/

/\* 1MS延时函数 \*/

/\* \*/

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

void delay\_ms(int count) // /\* X1ms \*/

{

int i,j;

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

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

}

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

/\* \*/

/\* MP3操作指令 \*/

/\* \*/

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

void mp3(unsigned char comd)

{

switch(comd)

{

case 0X01: send(0X7E);//指令开始位

send(0X04);//说明指令长度

send(0X42);//指定文件夹播放

send(0X01);//定位到名称为“01”的文件夹

send(0x01);//播放第1个音频文件

send(0XEF);//指令结束位

break;

case 0X02: send(0X7E);//指令开始位

send(0X04);//说明指令长度

send(0X42);//指定文件夹播放

send(0X01);//定位到名称为“01”的文件夹

send(0x02);//播放第2个音频文件

send(0XEF);//指令结束位

break;

case 0X03: send(0X7E);//指令开始位

send(0X04);//说明指令长度

send(0X42);//指定文件夹播放

send(0X01);//定位到名称为“01”的文件夹

send(0x03);//播放第3个音频文件

send(0XEF);//指令结束位

break;

case 0X04: send(0X7E);//指令开始位

send(0X04);//说明指令长度

send(0X42);//指定文件夹播放

send(0X01);//定位到名称为“01”的文件夹

send(0x04);//播放第4个音频文件

send(0XEF);//指令结束位

break;

case 0X05: send(0X7E);//指令开始位

send(0X04);//说明指令长度

send(0X42);//指定文件夹播放

send(0X01);//定位到名称为“01”的文件夹

send(0x05);//播放第5个音频文件

send(0XEF);//指令结束位

break;

case 0X06: send(0X7E);//指令开始位

send(0X04);//说明指令长度

send(0X42);//指定文件夹播放

send(0X01);//定位到名称为“01”的文件夹

send(0x06);//播放第6个音频文件

send(0XEF);//指令结束位

break;

case 0X07: send(0X7E);//指令开始位

send(0X04);//说明指令长度

send(0X42);//指定文件夹播放

send(0X01);//定位到名称为“01”的文件夹

send(0x07);//播放第7个音频文件

send(0XEF);//指令结束位

break;

case 0X08: send(0X7E);//指令开始位

send(0X04);//说明指令长度

send(0X42);//指定文件夹播放

send(0X01);//定位到名称为“01”的文件夹

send(0x08);//播放第8个音频文件

send(0XEF);//指令结束位

break;

case 0X09: send(0X7E);//指令开始位

send(0X04);//说明指令长度

send(0X42);//指定文件夹播放

send(0X01);//定位到名称为“01”的文件夹

send(0x09);//播放第9个音频文件

send(0XEF);//指令结束位

break;

case 0X0A: send(0X7E);//指令开始位

send(0X04);//说明指令长度

send(0X42);//指定文件夹播放

send(0X01);//定位到名称为“01”的文件夹

send(0x0A);//播放第10个音频文件

send(0XEF);//指令结束位

break;

case 0X0B: send(0X7E);//指令开始位

send(0X04);//说明指令长度

send(0X42);//指定文件夹播放

send(0X01);//定位到名称为“01”的文件夹

send(0x0B);//播放第11个音频文件

send(0XEF);//指令结束位

break;

case 0X0C: send(0X7E);//指令开始位

send(0X04);//说明指令长度

send(0X42);//指定文件夹播放

send(0X01);//定位到名称为“01”的文件夹

send(0x0C);//播放第12个音频文件

send(0XEF);//指令结束位

break;

case 0X0D: send(0X7E);//指令开始位

send(0X04);//说明指令长度

send(0X42);//指定文件夹播放

send(0X01);//定位到名称为“01”的文件夹

send(0x0D);//播放第13个音频文件

send(0XEF);//指令结束位

break;

case 0X0E: send(0X7E);//指令开始位

send(0X04);//说明指令长度

send(0X42);//指定文件夹播放

send(0X01);//定位到名称为“01”的文件夹

send(0x0E);//播放第14个音频文件

send(0XEF);//指令结束位

break;

case 0X0F: send(0X7E);//指令开始位

send(0X04);//说明指令长度

send(0X42);//指定文件夹播放

send(0X01);//定位到名称为“01”的文件夹

send(0x0F);//播放第15个音频文件

send(0XEF);//指令结束位

break;

case 0X31: send(0X7E);//指令开始位

send(0X03);//说明指令长度

send(0X31);//设置音量

send(0X20);//20

send(0XEF);//指令结束位

break;

case 0X32: send(0X7E);//指令开始位

send(0X03);//说明指令长度

send(0X31);//设置音量

send(0X14);//20

send(0XEF);//指令结束位

break;

case 0X33: send(0X7E);//指令开始位

send(0X03);//说明指令长度

send(0X33);//设置循环模式

send(0X04);//单曲播放

send(0XEF);//指令结束位

break;

case 0X34: send(0X7E);//指令开始位

send(0X03);//说明指令长度

send(0X33);//设置循环模式

send(0X02);//单曲循环！

send(0XEF);//指令结束位

break;

case 0X35: send(0X7E);//指令开始位

send(0X03);//说明指令长度

send(0X31);//设置音量

send(0X07);//20

send(0XEF);//指令结束位

break;

}

}

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

/\* \*/

/\* MP3初始化 \*/

/\* \*/

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

void mp3\_init()

{

mp3(0x33); //设置循环模式为文件夹循环

mp3(0x31); //设置音量为20

//mp3(0x00); //开机播放

/\*不关闭串行中断的话一上电就会发送STOP，这样会影响曲名和播放歌曲的对应关系\*/

ES=0; //串行中断禁止

delay\_ms(100); //等待MP3准备好

ES=1;

}