越狱程序—PART1

#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， R1电磁锁

sbit OUT2 = P1^0; //输出2， R2电磁锁

sbit OUT3 = P1^1; //输出3， R3电磁锁

sbit OUT4 = P1^2; //输出4， R4电磁锁

sbit OUT5 = P1^3; //输出5， R5电磁锁

sbit OUT6 = P1^4; //输出6， 激光发射1

sbit OUT7 = P1^5; //输出7， 激光发射2

sbit OUT8 = P1^6; //输出8， 报警灯1

sbit OUT9 = P1^7; //输出9， 报警灯2

sbit OUT10 = P4^3; //输出10， 报警灯3

sbit OUT11 = P3^2; //输出11，

sbit OUT12 = P3^3; //输出12，

sbit M10 = P2^6; //电缸1+ 单片机输出信号告诉电话响铃！蓝色线

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， 告诉单片机打开R5的门！ 红色线！

sbit INPUT15 = P4^5; //输入15， R1电磁锁反馈输入

sbit INPUT14 = P4^1; //输入14 R2电磁锁反馈输入

sbit INPUT13 = P4^6; //输入13 R3电磁锁反馈输入

sbit INPUT12 = P0^7; //输入12 激光接收1

sbit INPUT11 = P0^6; //输入11，按键类型 激光接收2

sbit INPUT10 = P0^5; //输入10，门禁类型 R1门禁

sbit INPUT9 = P0^4; //输入9，门禁 R2门禁

sbit INPUT8 = P0^3; //输入8，门禁 R3门禁

sbit INPUT7 = P0^2; //输入7，门禁 R4门禁

sbit INPUT6 = P0^1; //输入6，门禁 激光接收1

sbit INPUT5 = P0^0; //输入5，门禁类型 激光接收2

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,step\_flag1=0;

unsigned char flag\_d1=0,flag\_d2=0,flag\_d3=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; //设置P4口部分特殊引脚为普通IO口

Game\_Start=0;

delay\_ms(8000); //延时一段时间，开启背景音乐

delay\_ms(8000);

delay\_ms(8000);

delay\_ms(8000);

mp3(0X33);

mp3(0x01); //音效代入，背景音乐起

delay\_ms(5000);

flag\_d1=0,flag\_d2=0,flag\_d3=0;

while(1)

{

if((INPUT15==0)&&(flag\_d1==0)) //铁门1电磁锁有反馈，发出监狱关门的音效，音效2

{

delay\_ms(50);

if((INPUT15==0)&&(flag\_d1==0))

{

mp3(0X33);

mp3(0x02);

flag\_d1=1;

delay\_ms(2000);

}

}

if((INPUT14==0)&&(flag\_d2==0)) //铁门2电磁锁有反馈，发出监狱关门的音效，音效2

{

delay\_ms(50);

if((INPUT14==0)&&(flag\_d2==0))

{

mp3(0X33);

mp3(0x02);

flag\_d2=1;

delay\_ms(2000);

}

}

if((INPUT13==0)&&(flag\_d3==0)) //铁门3电磁锁有反馈，发出监狱关门的音效，音效2

{

delay\_ms(50);

if((INPUT13==0)&&(flag\_d3==0))

{

mp3(0X33);

mp3(0x02);

flag\_d3=1;

delay\_ms(2000);

}

}

if((flag\_d3==1)&&(flag\_d2==1)&&(flag\_d1==1)&&(Game\_Start==0)) //三个电磁锁都处于关闭状态时

{

delay\_ms(50);

if((INPUT15==0)&&(INPUT14==0)&&(INPUT13==0))

{

Game\_Start=1; //游戏开始的flag

delay\_ms(4000);

OUT6=1; //打开激光

mp3(0x03); //音效3，激光启动声

delay\_ms(2000);

mp3(0x04); //音效4，监狱嘈杂的声音，单次播放

delay\_ms(8000);

delay\_ms(8000);

delay\_ms(8000);

delay\_ms(8000);

delay\_ms(8000);

mp3(0X33);

mp3(0x07); //背景声音，紧张氛围

}

}

if(INPUT10==0) //R1门禁密码输入正确，开R1的电磁锁

{

delay\_ms(50);

if(INPUT10==0)

{

OUT1=0; //开R1的门，打开房间1报警灯

OUT8=1;

door1=1;

}

}

if(INPUT9==0) //R2门禁密码输入正确，开R2的锁

{

delay\_ms(50);

if(INPUT9==0)

{

OUT2=0; //开R2的门，开房间2报警灯

OUT9=1;

door2=1;

}

}

if(INPUT8==0) //R3门禁密码输入正确，开R3的锁

{

delay\_ms(50);

if(INPUT8==0)

{

OUT3=0; //开R3的门，开房间3报警灯

OUT10=1;

door3=1;

}

}

if((door1==1)&&(door2==1)&&(door3==1)&&(step\_flag1==0))

{

delay\_ms(5000);

delay\_ms(5000);

mp3(0x08); //三个门全开时，神秘人语音出现，指示玩家下一步的动作

delay\_ms(5000);

OUT1=1;

OUT2=1;

OUT3=1;

door1=0;

door2=0;

door3=0;

delay\_ms(8000);

delay\_ms(8000);

delay\_ms(5000);

mp3(0x09); //秒表声 思考时间

delay\_ms(5000);

delay\_ms(2000);

mp3(0x0A); //神秘人语音第2段

delay\_ms(8000);

delay\_ms(8000);

delay\_ms(8000);

mp3(0X33);

mp3(0x0B); //R4门开启前的紧张音效

delay\_ms(3000);

}

if((INPUT11==0)&&(INPUT12==0)) //2个激光接收都不接收

{

OUT7=0; //R4的门禁断电！

}

else //2个激光有一个可以接收到

{

OUT7=1; //R4的门禁通电！

}

if(INPUT7==0) //R4门禁密码输入正确，开R4的锁

{

delay\_ms(50);

if(INPUT7==0)

{

step\_flag1=1;

OUT4=0; //开R4的门

OUT6=0; //关闭激光1

OUT8=0; //关闭3个报警灯

OUT9=0;

OUT10=0;

mp3(0X33);

mp3(0x0C); // 音效12，R4门开启后短暂音效

delay\_ms(5000);

delay\_ms(5000);

delay\_ms(5000);

delay\_ms(5000);

M10=0; // 延时一段时间后告诉电话开始响铃

delay\_ms(500);

M10=1;

mp3(0X33);

mp3(0x0D); // 音效13，神秘紧张氛围音效，循环播放，直至打开R5的门

}

}

if(INPUT16==0) //电话端传来开门信号，INPUT16接地

{

delay\_ms(100);

if(INPUT16==0)

{

delay\_ms(2000);

OUT5=0; //打开R5的门

delay\_ms(8000);

delay\_ms(8000);

delay\_ms(8000);

mp3(0x33); //放置一段代入语音

mp3(0X0E);

delay\_ms(8000);

delay\_ms(8000);

delay\_ms(2000);

while(1)

{

mp3(0X32); //降低此时的音量

mp3(0X33);

mp3(0x02); // 音效12，R5门开启后短暂音效，关闭音响

delay\_ms(6000);

ES=0;

}

}

}

}

}

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

/\* \*/

/\* 系统初始化函数 \*/

/\* \*/

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

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 P3.0 3.1 RXD TXD设置为普通IO口 防止音响的异响

P4M1=0X00;

P4M0=0XFF; //设置P4为推挽输出

//R1-R5所有电磁锁吸合，初始状态

OUT1=1;

OUT2=1;

OUT3=1;

OUT4=1;

OUT5=1;

OUT6=0; // 激光关闭

OUT7=1; // R4门禁供电状态（初始）

OUT8=0; // 报警灯1关闭

OUT9=0; // 报警灯2关闭

OUT10=0; // 报警灯3关闭

M10=1; //告诉电话响铃，初始为高电平，触发后为低电平

INPUT16=1; //输入,接地时电话告诉单片机打开R5的门

INPUT15=1; //电磁锁1反馈，输入检测，先写1

INPUT14=1;

INPUT13=1;

INPUT11=1; //激光接收管

INPUT12=1;

INPUT10=1;

INPUT9=1;

INPUT8=1;

INPUT7=1;

INPUT6=1;

INPUT5=1;

uart\_init(); //串口初始化

mp3\_init(); //MP3初始化 \*/

}

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

/\* \*/

/\* 1ms延时程序 \*/

/\* \*/

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

void delay\_ms(int count) // /\* X1ms \*/ 延时1ms程序

{

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(0X10);//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;

case 0X36: send(0X7E);//指令开始位

send(0X03);//说明指令长度

send(0X31);//设置音量

send(0X10);//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;

}