

第 3 章 射频 13.56M 一卡通 门禁系统实训

洞洞板是一种通用设计的电路板,通常其上布满标准 IC 间距(2.54mm)的圆型独立的焊盘,看起来整个板子上都是小孔,又称为“万能电路实验板”。相比专业的印刷电路板,洞洞板具有成本低、使用方便、扩展灵活等特点。

3.1 洞板飞线焊接的基础知识

学习目标:熟悉理解洞洞板的基础知识,学习洞洞板的选择原则及使用焊接技巧。

项目重点是洞洞板的选择及使用。

本项目难点是焊接前的准备、洞洞板的焊接方法、洞洞板的焊接技巧。

3.1.1 洞板飞线基础知识

洞洞板主要有两种,一种焊盘各自独立(图 3-1,以下简称为单孔板),另一种是多个焊盘连在一起(图 3-2,以下简称为连孔板)。

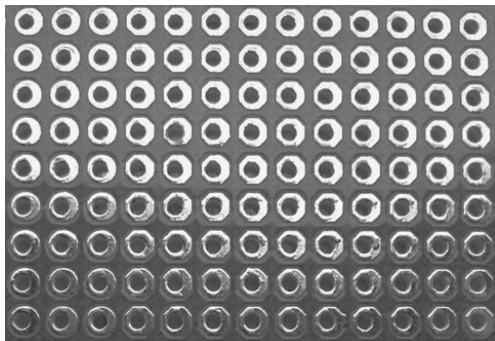


图 3-1 单孔板

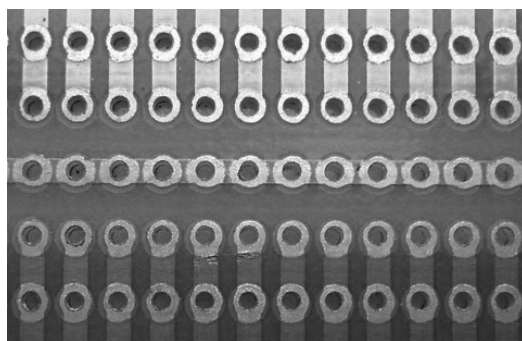


图 3-2 连孔板

单孔板又分为单面板和双面板两种。单孔板较适合数字电路和单片机电路,连孔板则更适合模拟电路和分立电路。因为数字电路和单片机电路以芯片为主,电路较规则;而模拟电路和分立电路往往较不规则,分立元件的引脚常常需要连接多根线,这时如果有多个焊盘连在一起就要方便一些。

洞洞板有两种不同材质:铜板和锡板。铜板的焊盘是裸露的铜,呈现金黄色,平时应该用报纸包好保存以防止焊盘氧化,万一焊盘氧化了(焊盘失去光泽、不好上锡),可以用棉棒蘸酒精清洗或用橡皮擦拭。焊盘表面镀了一层锡的是锡板,焊盘呈现银白色,锡板的基板材

质要比铜板坚硬,不易变形。

1. 焊接前的准备

在焊接洞洞板之前需要准备足够的细导线用于走线。细导线分为单股和多股。单股硬导线可将其弯折成固定形状,剥皮之后还可以当作跳线使用;多股细导线质地柔软,焊接后显得较为杂乱。

洞洞板具有焊盘紧密等特点,这就要求烙铁头有较高的精度,建议使用功率 30W 左右的尖头电烙铁。同样,焊锡丝也不能太粗,建议选择线径为 0.5~0.6mm 的。

2. 洞洞板的焊接方法

对于元器件在洞洞板上的布局,大多数人习惯“顺藤摸瓜”,就是以芯片等关键器件为中心,其他元器件见缝插针的方法。这种方法是边焊接边规划,无序中体现着有序,效率较高。但由于初学者缺乏经验,所以不太适合用这种方法,初学者可以先在纸上做好初步的布局,然后用铅笔画到洞洞板正面(元件面),继而可以将走线也规划出来,方便自己焊接。

对于洞洞板的焊接方法,一般是利用前面提到的细导线进行飞线连接,飞线连接没有太大的技巧,但尽量做到水平和竖直走线,整洁清晰。

3.1.2 洞洞板的焊接技巧

(1) 初步确定电源、地线的布局:电源贯穿电路始终,合理的电源布局对简化电路起到十分关键的作用。某些洞洞板布置有贯穿整块板子的铜箔,应将其用作电源线和地线;如果无此类铜箔,也需要对电源线、地线的布局有个初步的规划。

(2) 善于利用元器件的引脚:洞洞板的焊接需要大量的跨接、跳线等,不要急于剪断元器件多余的引脚,有时候直接跨接到周围待连接的元器件引脚上会事半功倍。另外,本着节约材料的目的,可以把剪断的元器件引脚收集起来作为跳线用材料。

(3) 善于设置跳线:特别要强调这一点,多设置跳线不仅可以简化连线,而且要美观得多。

(4) 善于利用元器件自身的结构:善于利用元器件自身的结构,可以利用这一特点来简化连线。

(5) 善于利用排针:排针有许多灵活的用法,比如两块板子相连,就可以用排针和排座,排针既起到了两块板子间的机械连接作用又起到电气连接的作用。这一点借鉴了计算机的板卡连接方法。

(6) 在需要的时候隔断铜箔:在使用连孔板的时候,为了充分利用空间,必要时可用小刀割断某处铜箔,这样就可以在有限的空间放置更多的元器件。

(7) 充分利用双面板:双面板比较昂贵,既然选择它就应该充分利用它。双面板的每一个焊盘都可以当作过孔,灵活实现正反面电气连接。

(8) 充分利用板上的空间:芯片座里面隐藏元件,既美观又能保护元件。

3.2 射频 13.56M 一卡通门禁系统硬件设计

学习目标：通过一卡通门禁系统硬件设计，实践洞洞板飞线焊接技巧，理解掌握物联网硬件的基本工作原理。

本实训项目重点：一卡通门禁读写器的原理图设计；洞洞板飞线焊接实现基础硬件底板。

项目难点：理解原理图的含义；根据原理图，焊接实现硬件底板；焊接、调试与故障排除技巧。

3.2.1 射频 13.56M 一卡通门禁系统原理图识读

射频 13.56M 一卡通门禁系统硬件原理，分别如图 3-3～图 3-6 所示。

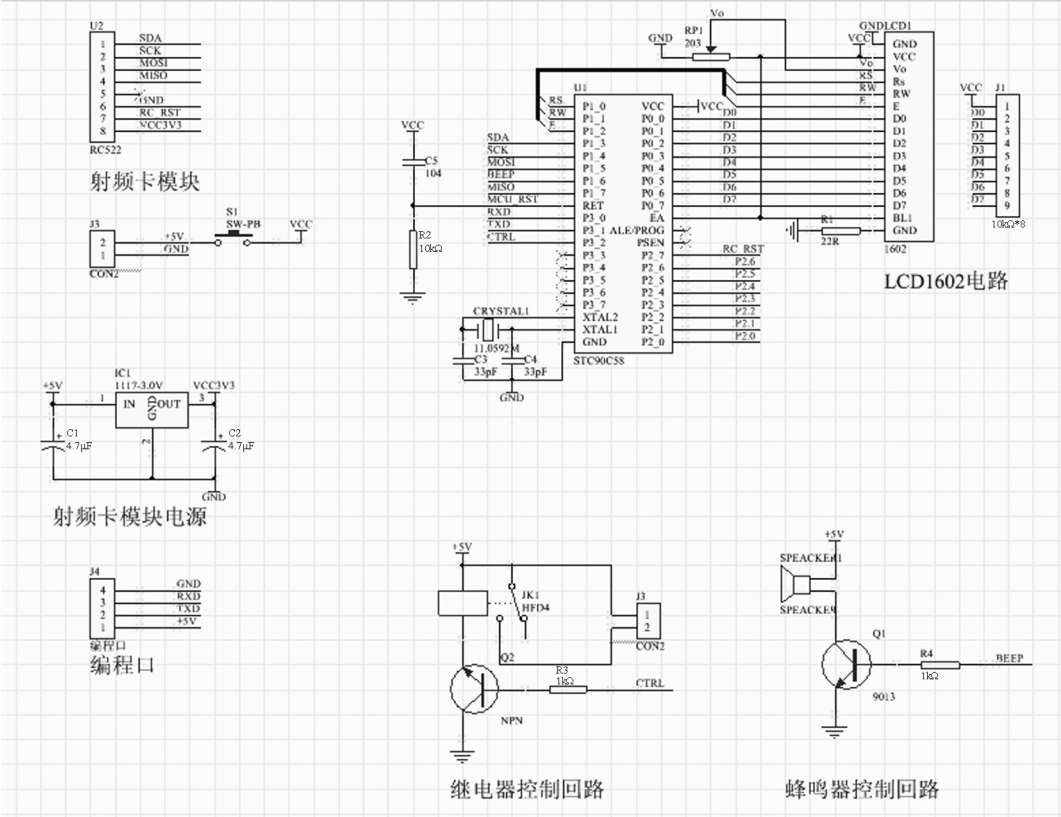


图 3-3 射频 13.56M 一卡通门禁系统硬件总体原理图

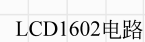


Figure 1-1 shows the pin connection diagram of the RC522 module. The module is labeled U2 and has 8 pins. The connections are as follows:

- Pin 1: SDA
- Pin 2: SCK
- Pin 3: MOSI
- Pin 4: MISO
- Pin 5: GND
- Pin 6: RC RST
- Pin 7: VCC3V3
- Pin 8: GND

The power supply circuit is shown below the module. It includes a 5V input (J3) connected to the +5V pin of the RC522 module. A switch (S1) is connected to the +5V pin. The output of the switch is connected to the VCC3V3 pin of the RC522 module. The power supply circuit also includes a 1117-3.0V voltage regulator (IC1) with a 4.7μF capacitor (C1) on the input and a 4.7μF capacitor (C2) on the output.

87

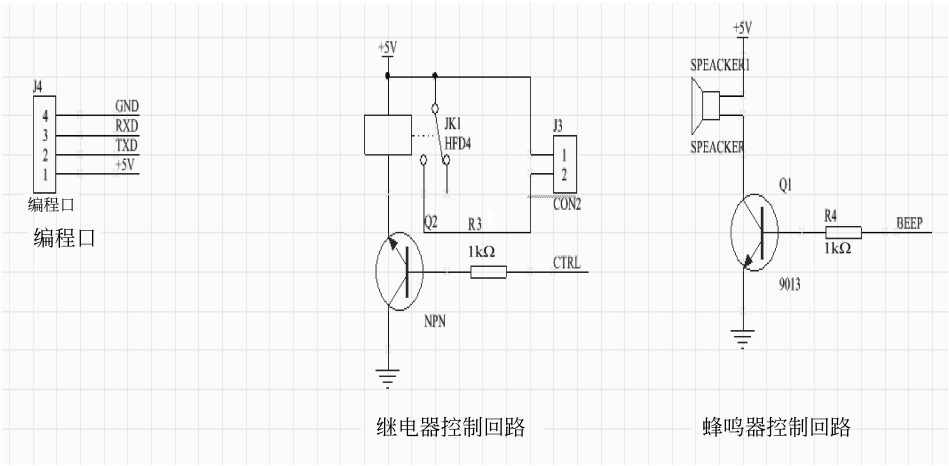


图 3-6 射频 13.56M 一卡通门禁系统继电器和蜂鸣器控制回路原理图

射频 13.56M 一卡通门禁系统所需要的元器件及材料准备,如表 3-1 所示。

表 3-1 一卡通门禁实训元件明细表

规格名称	位号	数量
洞洞板(9cm×15cm)		1
STC89C52 单片机	U1	1
DIP-40 普通芯片座	U1	1
按键	SW1-SW12	12
自锁开关	S1	1
瓷片电容 104 33pF	C3、C4	2
11.0592M 晶振		1
LCD1602		1
LCD1602 20 孔排针底座	LCD	1
A103G 排阻 1kΩ	J1	1
滑动变阻器 W203		1
电解电容 2A104J	C5	1
S9012 三极管	1	1
S9013 三极管	1	1
电阻 10kΩ	R2	1
电阻 1kΩ	R3、R4	2
电阻 22R	R1	1
RC522 芯片	U2	1
RC522 芯片 8 孔排针底座	U2	
有极性电解电容 4.7μF 50V	C1、C2	
有源蜂鸣器		1
HFD4/5 继电器	J3	1

续表

规格名称	位 号	数 量
直流 5V 电源插孔(内正外负)		1
4 针排针底座	J4	1
2 孔接线座		1
1117-3.0V 稳压器		1
磁力锁		1
12V 变压适配器		1
面包板		1
杜邦线		若干

3.2.2 射频 13.56M 一卡通门禁系统制作步骤

(1) 将元器件从试验工具箱中取出,如图 3-7 所示,识别各个元器件,必要时请测量相关参数(如电阻值)。

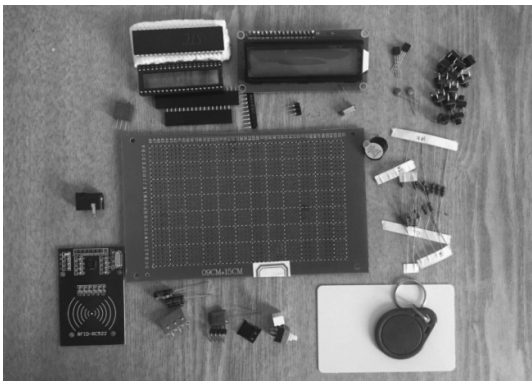


图 3-7 射频 13.56M 一卡通门禁系统元器件

(2) 将元器件按照图 3-8 所示进行布局,记好各个元器件的坐标位置,特别是区分正负极性的电容、三极管。

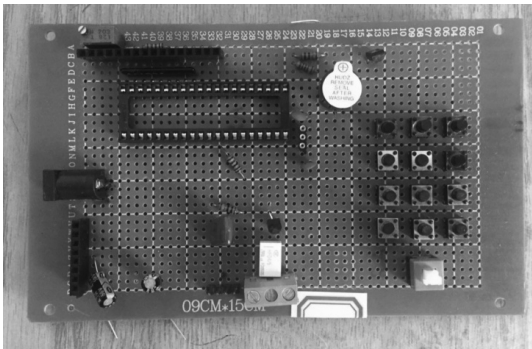


图 3-8 射频 13.56M 一卡通门禁系统元器件布局

(3) 按照射频 13.56M 一卡通门禁系统原理图,仔细焊接好每条线,建议用坐标连线法,要有耐心,不能着急,如图 3-9 所示。

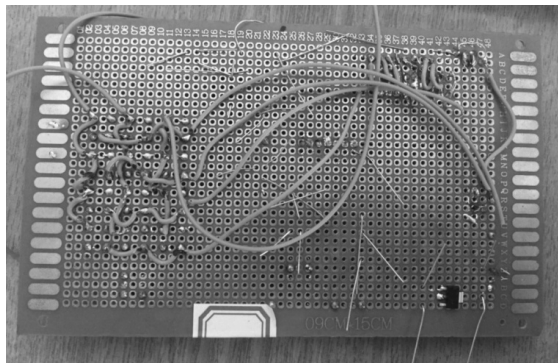


图 3-9 射频 13.56M 一卡通门禁系统焊接过程示意图(背面)

(4) 焊接完成,将 CPU 小心插入底座,注意方向,月牙形状底座与 CPU 月牙相一致,如图 3-10 所示。

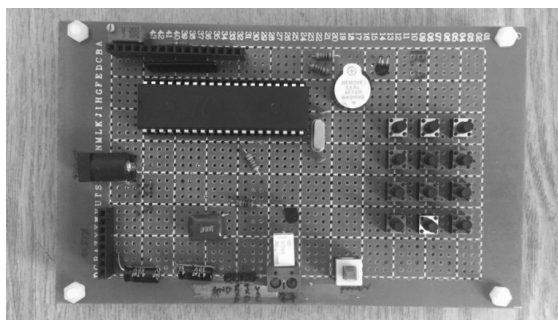


图 3-10 射频 13.56M 一卡通门禁系统焊接过程示意图(正面)

(5) 插上 RS522 芯片和 LCD1602,注意方向,如图 3-11 所示。

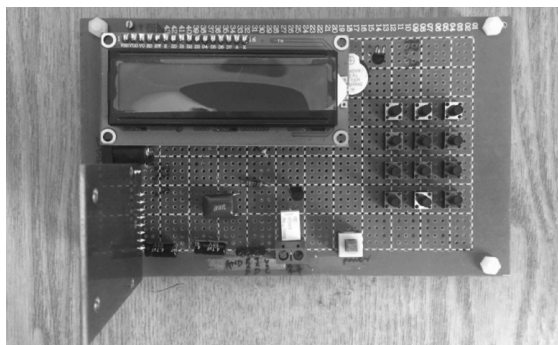


图 3-11 RS522 芯片和 LCD1602 的接入操作图

(6) 拿出 USB5V 电源线,上电,听见蜂鸣器的声音,按下自锁开关,LCD 出现欢迎界面,如图 3-12 所示。

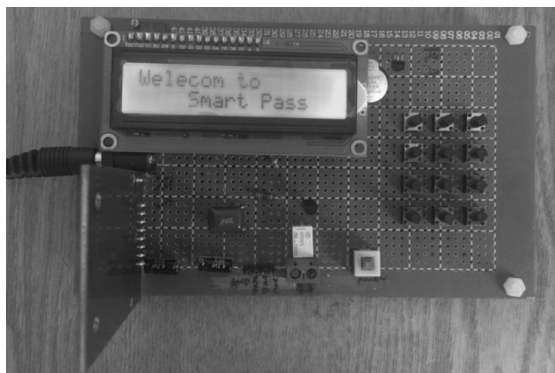


图 3-12 射频 13.56M 一卡通门禁系统 LCD1602 显示欢迎界面

(7) 将钥匙卡放进 RS522 天线附近,听见读卡声音,LCD 显示钥匙卡 ID 号,如图 3-13 所示。

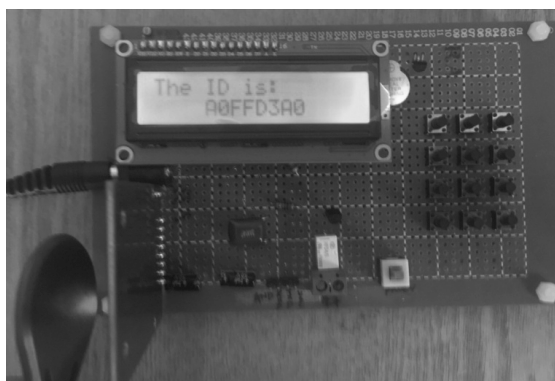


图 3-13 射频 13.56M 一卡通门禁系统钥匙卡读卡界面

(8) 将钥匙白卡放进 RS522 天线附近,听见读卡声音,LCD 显示钥匙白卡号,如图 3-14 所示。



图 3-14 射频 13.56M 一卡通门禁系统白卡读卡界面

(9) 安装小锁芯电路,如图 3-15 和图 3-16 所示。

(10) 观察刷钥匙卡及白卡,锁开现象,如图 3-17 和图 3-18 所示。

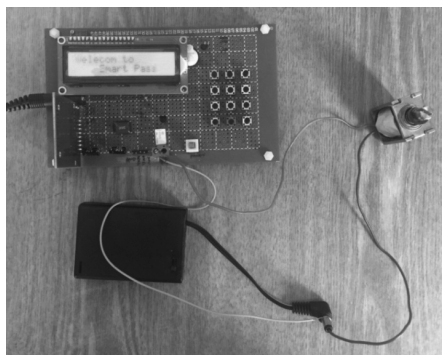


图 3-15 安装小锁芯电路示意图一

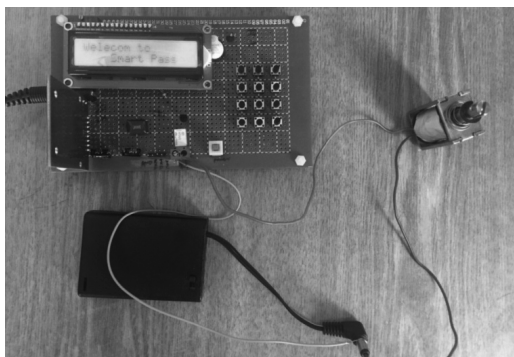


图 3-16 安装小锁芯电路示意图二

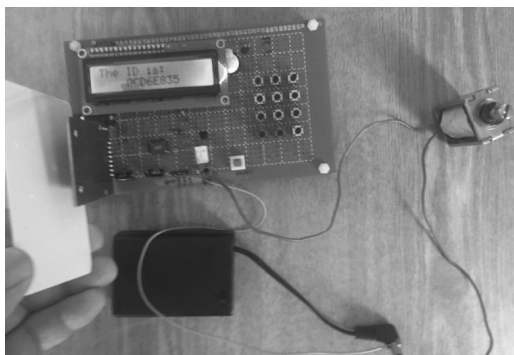


图 3-17 观察刷白卡,锁开操作示意图

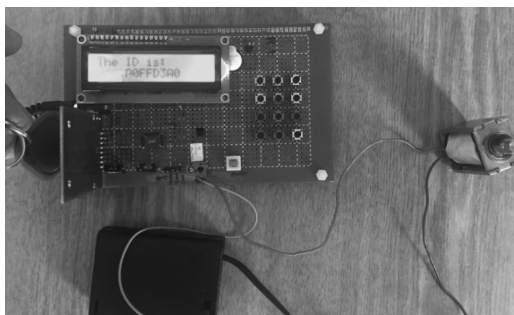


图 3-18 观察刷钥匙卡,锁开操作示意图

3.3 射频 13.56M 一卡通门禁系统软件设计

学习目标：通过射频 13.56M 一卡通门禁系统软件设计，实践 Keil μ Vison 集成开发环境使用技巧，理解掌握物联网软件工程开发的一般步骤。

项目重点：(1)一卡通门禁系统软件工程设计；(2)Keil μ Vison 集成开发环境使用技巧。

项目难点：(1)理解工程目录的组织结构；(2)物联网软件工程的开发的一般步骤，实现所需功能；(3)Keil μ Vison 集成开发环境中，注意积累程序调试技巧。

3.3.1 Keil μ Vison 工程源文件(*.c)的程序设计

操作思路如下。

Keil μ Vison 工程的一般目录结构，如图 3-19 所示。

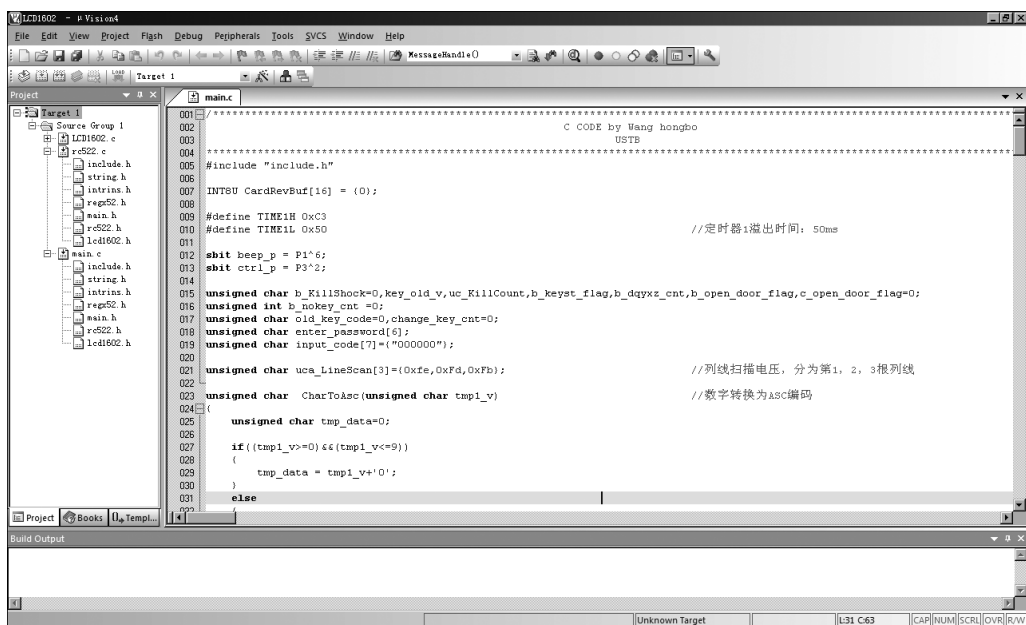


图 3-19 Keil μ Vison 工程的一般目录结构

(1) main.c 文件如下。

```

/*****
C CODE by Wang hongbo
*****/

#include "include.h"

INT8U CardRevBuf[16] = {0};

#define TIME1H 0xC3
#define TIME1L 0x50                                     //定时器 1 溢出时间:50ms

```

```

sbit beep_p = P1^6;
sbit ctrl_p = P3^2;
unsignedchar b_KillShock=0, key_old_v, uc_KillCount, b_keyst_flag;
unsignedchar b_dqyxz_cnt, b_open_door_flag, c_open_door_flag=0;
unsigned int b_nokey_cnt =0;
unsigned char old_key_code=0, change_key_cnt=0;
unsigned char enter_password[6];
unsigned char input_code[7]={"000000"};
unsigned char uca_LineScan[3]={0xfe, 0xFd, 0xFb}; //列线扫描电压, 分为第 1、2、3 根列线
unsigned char CharToAsc(unsigned char tmp1_v) //数字转换为 ASC 编码
{
    unsigned char tmp_data=0;

    if((tmp1_v>=0) && (tmp1_v<=9))
    {
        tmp_data = tmp1_v+ '0';
    }
    else
    {
        tmp_data = (tmp1_v-10) + 'A';
    }
    return tmp_data;
}

void display_id(unsigned char * str) //显示卡 ID 号
{
    unsigned char i =0;
    unsigned char tmp_v[16]={0};
    unsigned char tmp1_v=0;
    for(i=0; i<4; i++) //转成 ASCII 码
    {
        tmp_v[i] = ' ';
    }
    for(i=0; i<4; i++) //转成 ASCII 码
    {
        tmp1_v = ( * (str+i) ) &0x0f;
        tmp_v[i * 2+1+4] =CharToAsc(tmp1_v);
        tmp1_v = (( * (str+i) ) >>4);
        tmp_v[i * 2+4] =CharToAsc(tmp1_v);
    }
    for(i=0; i<4; i++) //转成 ASCII 码
    {
        tmp_v[i+12] = ' ';
    }
    tmp_v[15]='\0';
    GotoXY(0, 0);
}

```



```

        case 21: return '4'; break;
        case 22: return '5'; break;
        case 23: return '6'; break;
        case 31: return '7'; break;
        case 32: return '8'; break;
        case 33: return '9'; break;
        case 42: return '0'; break;
        default: break;
    }
    return 0xFF;
}

void vKeyProcess(unsigned char ucKeyCode)
{
    unsigned char tmp_v, i=0;
    if(ucKeyCode == old_key_code)
    {
        change_key_cnt++;
        if(change_key_cnt < 200)
        {
            return;
        }
    }
    else
    {
        change_key_cnt = 0;
    }
    if(ucKeyCode == 43)                                /* 确定键 */
    {
        /* 012345 作为主人预设密码使用 */
        if((input_code[0] == '0') && (input_code[1] == '1') && (input_code[2] ==
            '2') && (input_code[3] == '3') && (input_code[4] == '4') && (input_code
            [5] == '5'))
        {
            GotoXY(0, 0);
            Print("The Password is:");
            GotoXY(0, 1);
            Print("      ok!      ");
            c_open_door_flag = 0x55;
        }
        else
        {
            GotoXY(0, 0);
            Print("The Password is:");
            GotoXY(0, 1);
            Print("      Error!!      ");
        }
    }
}

```

```

    }
    else
    {
        if(ucKeyCode==41)                                /* 取消键,取消输入 */
        {
            b_dqyxz_cnt =0;
            for(i=0;i<6;i++)
            {
                input_code[i]='0';
            }
            input_code[6]='\0';

            c_open_door_flag =0;
        }
        else
        {
            tmp_v =ucGetNum(ucKeyCode);                    /* 数字按键处理 */
            if(tmp_v !=0xff)
            {
                input_code[b_dqyxz_cnt]=tmp_v;
                if(b_dqyxz_cnt <5)
                {
                    b_dqyxz_cnt++;
                }
            }
        }
        GotoXY(0,0);
        Print("Input The Code:");
        GotoXY(0,1);
        Print("                ");
        GotoXY(4,1);
        Print(input_code);
    }
    old_key_code =ucKeyCode;
}

void init_timer()
{
    TH1=TIME1H;
    TL1=TIME1L;
    TR1=1;                                                //开启定时器 1
    ET1=1;                                                //开定时器 1 中断
}

//*****
//*****定时器 1 中断,用于计时功能和防抖动标志清除以及显示报告

```

```

//*****
void vTimer1(void) interrupt 3
{
    unsigned char tmp_v;
    if(b_KillShock==0)
    {
        P2 &=0xf8;
        nop();
        nop();
        tmp_v =P2&0x78;
        if(tmp_v !=0x78)                //有按键按下
        {
            if(key_old_v !=tmp_v)        //第一次判断有按键按下
            {
                key_old_v =tmp_v;
                uc_KillCount=0;
            }
            else
            {
                uc_KillCount++;
            }
            if(uc_KillCount >=5)          //去抖操作
            {
                b_KillShock =0x01;
            }
        }
    }
    else
    {
        P2 &=0xf8;
        nop();
        nop();
        tmp_v =P2&0x78;
        if(tmp_v !=0x78)                //有按键按下
        {
            if(key_old_v !=tmp_v)        //第一次判断有按键按下
            {
                key_old_v =tmp_v;
                uc_KillCount=0;
            }
            else
            {
                uc_KillCount++;
            }
            if(uc_KillCount >=20)         //去抖操作
            {

```

```

        b_keyst_flag=0x01;                //去抖后按键有效标识
        b_nokey_cnt =0;
    }
}
else
{
    b_nokey_cnt++;
    if(b_nokey_cnt >=5000)
    {
        b_keyst_flag=0;
        b_KillShock=0;
        b_nokey_cnt =0;
    }
}
}
//恢复定时器 1 溢出时间
TH1=TIME1H;
TL1=TIME1L;
}

void init_all(void)
{
    EA =0;
    init_rc522();                        //初始化 rc522 模块
    LCD_Initial();                      //初始化 LCD1602
    init_timer();                       //定时器初始化
    EA =1;
}

char card_charge()
{
    if( PcdRequest( PICC_REQIDL, &CardRevBuf[0] ) !=MI_OK )
        //寻天线区内未进入休眠状态的卡,返回卡片类型 2 字节
    {
        if( PcdRequest( PICC_REQIDL, &CardRevBuf[0] ) !=MI_OK )
            //寻天线区内未进入休眠状态的卡,返回卡片类型 2 字节
        {
            return -1;
        }
    }
    if( PcdAnticoll( &CardRevBuf[2] ) !=MI_OK )    //防冲撞,返回卡的序列号 4 字节
    {
        return -1;
    }
    if( PcdSelect( &CardRevBuf[2] ) !=MI_OK )      //选卡
    {

```

```

        return -1;
    }
    display_id(&CardRevBuf[2]);
    return 0;
}
main()
{
    unsigned int Count=0;
    unsigned char i=0;
    unsigned char opdoor_flag=0;
    unsigned char keycode=0;
    unsigned char tmp_v[16]="Welecom to";
    unsigned char tmp1_v[16]="Smart Pass";
    init_all();
    beep_p=0;
    ctrl_p=1;
    Delay1ms(10);
    GotoXY(0,0);
    Print(tmp_v);
    GotoXY(4,1);
    Print(tmp1_v);
    while(1)
    {
        if(card_charge()==0)                                /* 刷卡处理 */
        {
            b_open_door_flag=0x55;                            /* 继电器动作标识 */
        }
        else
        {
            b_open_door_flag=0;
        }
        if(b_open_door_flag!=0)
        {
            beep_p=1;                                          /* 蜂鸣器动作 */
            ctrl_p=0;                                          /* 继电器动作 */
        }
        else                                                  /* 无刷卡 */
        {
            if(b_KillShock!=0)                                /* 键显模式 */
            {
                if(b_keyst_flag!=0)
                {
                    vKeyProcess(ucKeyScan());                /* 按键处理 */
                    b_keyst_flag=0;
                }
                if(c_open_door_flag!=0)                        /* 按键输入密码正确 */

```

```

        {
            beep_p = 1;
            ctrl_p = 0;
            Delay1ms(10);
        }
        else
        {
            beep_p = 0;
            ctrl_p = 1;
        }
    }
else /* 待机界面 */
{
    b_dqyxz_cnt = 0;
    for(i=0;i<6;i++) /* 清输入缓冲 */
    {
        input_code[i]='0';
    }
    input_code[6]='\0';
    GotoXY(0,0);
    Print(tmp_v);
    GotoXY(4,1);
    Print(tmp1_v);
    beep_p = 0; /* 清继电器蜂鸣器 */
    ctrl_p = 1;
}
}
}
}

```

(2) Lcd1602.c 文件如下。

```

#include "include.h"

//Port Definitions*****
sbit LcdRs    =P1^0;
sbit LcdRw    =P1^1;
sbit LcdEn    =P1^2;
sfr  DBPort  =0x80; /*P0=0x80,P1=0x90,P2=0xA0,P3=0xB0.数据端口

//向 LCD 写入命令或数据*****
#define LCD_COMMAND    0 /*Command
#define LCD_DATA       1 /*Data
#define LCD_CLEAR_SCREEN 0x01 /*清屏
#define LCD_HOMING     0x02 /*光标返回原点

```

```
//内部等待函数*****
unsigned char LCD_Wait(void)
{
    LcdRs=0;
    LcdRw=1;  _nop_();
    LcdEn=1;  _nop_();
    DBPort&0x80==0x80
    LcdEn=0;
    return DBPort;
}

void LCD_Write(bit style, unsigned char input)
{
    LcdEn=0;
    LcdRs=style;
    LcdRw=0;  _nop_();
    DBPort=input;  _nop_();           //注意顺序
    LcdEn=1;  _nop_();           //注意顺序
    LcdEn=0;  _nop_();
    LCD_Wait();
}

//设置显示模式*****
#define LCD_SHOW      0x04        //显示开
#define LCD_HIDE      0x00        //显示关
#define LCD_CURSOR    0x02        //显示光标
#define LCD_NO_CURSOR 0x00        //无光标
#define LCD_FLASH     0x01        //光标闪动
#define LCD_NO_FLASH   0x00        //光标不闪动
void LCD_SetDisplay(unsigned char DisplayMode)
{
    LCD_Write(LCD_COMMAND, 0x08|DisplayMode);
}

//设置输入模式*****
#define LCD_AC_UP      0x02
#define LCD_AC_DOWN    0x00        //默认值
#define LCD_MOVE       0x01        //画面可平移
#define LCD_NO_MOVE    0x00        //默认值

void LCD_SetInput(unsigned char InputMode)
{
    LCD_Write(LCD_COMMAND, 0x04|InputMode);
}

//移动光标或屏幕*****
#define LCD_CURSOR     0x02
```

```

#define LCD_SCREEN    0x08
#define LCD_LEFT      0x00
#define LCD_RIGHT     0x04
void LCD_Move(unsigned char object, unsigned char direction)
{
    if(object==LCD_CURSOR)
        LCD_Write(LCD_COMMAND, 0x10|direction);
    if(object==LCD_SCREEN)
        LCD_Write(LCD_COMMAND, 0x18|direction);
}
//初始化 LCD*****
void LCD_Initial()
{
    LcdEn=0;
    LCD_Write(LCD_COMMAND, 0x38);      //8 位数据端口, 2 行显示, 5×7 点阵
    LCD_Write(LCD_COMMAND, 0x38);
    LCD_SetDisplay(LCD_SHOW|LCD_NO_CURSOR);    //开启显示, 无光标
    LCD_Write(LCD_COMMAND, LCD_CLEAR_SCREEN);  //清屏
    LCD_SetInput(LCD_AC_UP|LCD_NO_MOVE);      //AC 递增, 画面不动
}
//*****
void GotoXY(unsigned char x, unsigned char y)
{
    if(y==0)
        LCD_Write(LCD_COMMAND, 0x80|x);
    if(y==1)
        LCD_Write(LCD_COMMAND, 0x80|(x-0x40));
}
void Print(unsigned char * str)
{
    while(* str!='\0')
    {
        LCD_Write(LCD_DATA, * str);
        str++;
    }
}

```

(3) Rc522.c 文件如下。

```

/*****      C CODE by Wang hongbo
温馨提示:RC522 支持多种串行模式,此处使用 spi 总线模式
*****/
#include "include.h"
void delay_ns(unsigned int ns)
{

```