```
//*********************************************************
//                          指纹识别锁程序
//        作者：666 小奇
//        硬件：Arduino    AS608 指纹识别模块
//        提示：AS608 与 FPM10A 程序兼容
//        说明：本程序未经作者允许禁止转载
//         QQ :1792498113
//        E-mil:liujiaqi7998@qq.com
//        接线：A4-SDA    A5-SCL    指纹识别模块接 PIN2,PIN3 详见下面介绍
//                蜂鸣器-PIN 9    舵机-PIN 8    自动休眠(可选)PIN -7
//        项目使用了  11492  字节，  占用了  (35%)  程序存储空间。
//        全局变量使用了 860 字节，(41%)的动态内存
//        余留 1188 字节局部变量。最大为 2048 字节。
//**********************************************************


//引用库声明************************************************
#include <Adafruit_Fingerprint.h>    //指纹识别模块库
#include <SoftwareSerial.h>    //软串口通讯库
#include <Wire.h>    //I2C 总线库
#include <LiquidCrystal_I2C.h>    //I2C1062 屏幕库
#include <pt.h>    //多线程库
#include <DS3231.h>
//**********************************************************

//1602 液晶自定义符号*************************************************
byte lcda[8] = {      0x08, 0x0f, 0x12, 0x0f, 0x0a, 0x1f, 0x02, 0x02,}; //年
byte lcdb[8] = {      0x0f, 0x09, 0x0f, 0x09, 0x0f, 0x09, 0x13, 0x01,}; //月
byte lcdc[8] = {      0x0f, 0x09, 0x09, 0x0f, 0x09, 0x09, 0x0f, 0x00,}; //日
byte lcdd[8] = {      0x18, 0x18, 0x07, 0x08, 0x08, 0x08, 0x07, 0x00,}; //°C
byte lcde[8] = {      0x04, 0x04, 0x0a, 0x15, 0x04, 0x04, 0x14, 0x0c,}; //小
byte lcdf[8] = {      0x04, 0x1f, 0x04, 0x0a, 0x1f, 0x1a, 0x1a, 0x0e,}; //奇
//*******************************************************************


//初始化时间模块*****************************************************
DS3231 Clock;
bool Century = false;
bool h12;
bool PM;
byte ADay, AHour, AMinute, ASecond, ABits;
bool ADy, A12h, Apm;
int year, month, date, DoW, week , hour, minute, second, temperature;
char dis1[16] = {0}, dis2[16] = {0};
//*****************************************************************


//指纹识别初始化********************************************
```

```
int getFingerprintIDez();
// pin #2 is IN from sensor (GREEN wire)    指纹传感器输入
// pin #3 is OUT from arduino    (WHITE wire)    指纹传感器输出
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
// On Leonardo/Micro or others with hardware serial, use those! #0 is green wire, #1 is white
//Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial1);

//**********************************************************


//1062 初始化*************************************************
LiquidCrystal_I2C lcd(0x3F, 16, 2);
//  设置为 16 个字符，2 行 LCD 显示，地址 0x27
//如果 1062 无显示那么注意地址设置
//**********************************************************


//舵机设置**************************************************
int servopin = 8    ; //设置舵机接口 10
void servopulse(int angle)//定义一个脉冲函数
{
  int pulsewidth = (angle * 11) + 500; //将角度转化为 500-2480 的脉宽值
  digitalWrite(servopin, HIGH);      //将舵机接口电平至高
  delayMicroseconds(pulsewidth);    //延时脉宽值的微秒数
  digitalWrite(servopin, LOW);        //将舵机接口电平至低
  delayMicroseconds(20000 - pulsewidth);
}
//**********************************************************


//简单配置*************************************************
int Rtime = 0 ;//错误次数延时
static int counter1 , counter2 , counter3 ; //counter 为定时计数器
//*********************************************************


//线程 1，回到主页面状态舵机复位******************************
static int protothread1(struct pt *pt)

{
  PT_BEGIN(pt);    //线程开始
  while (1) //每个线程都不会死
  {
    PT_WAIT_UNTIL(pt, counter1 == 35 );
    //如果时间满了 3.5 秒左右，则继续执行，否则记录运行点，退出线程 1
    counter1 = 0; //计数器置零
    for (int i = 0; i < 80; i++) //发送 50 个脉冲
    {
      servopulse(90);     //引用脉冲函数
    }
    lcdplay();
  }
  PT_END(pt); //线程结束
}
//线程 2，错误重试等待时间清 0*********************************
static int protothread2(struct pt *pt) //线程 2，控制灯 2
{
  PT_BEGIN(pt); //线程开始
  while (1) {      //每个线程都不会死
    PT_WAIT_UNTIL(pt, counter2 == 150);
    Rtime = 0 ;
    counter2 = 0; //计数清零
  }
  PT_END(pt);    //线程结束
}
//线程 2，长时间无操作提示（pin7 低电平）可用作自动关机*********
static int protothread3(struct pt *pt) //线程 2，控制灯 2
{
  PT_BEGIN(pt); //线程开始
  while (1) {      //每个线程都不会死
    PT_WAIT_UNTIL(pt, counter3 == 200);
    digitalWrite(7, LOW);
  }
  PT_END(pt);    //线程结束
}
//************************************************************


//lcd1602 创建特殊字符（中文）*********************************
void diy()
{
  lcd.createChar(0, lcda);
  lcd.createChar(1, lcdb);
  lcd.createChar(2, lcdc);
  lcd.createChar(3, lcdd);
  lcd.createChar(4, lcde);
  lcd.createChar(5, lcdf);
}
//*************************************************************


//读取 DS3231 参数*********************************************
void ReadDS3231()
{
  Wire.begin();
  second = Clock.getSecond();
  minute = Clock.getMinute();
```

```
  hour = Clock.getHour(h12, PM);
  week = Clock.getDoW();
  date = Clock.getDate();
  month = Clock.getMonth(Century);
  year = Clock.getYear();
  temperature = Clock.getTemperature();
}
void get_dis()                    //1602 液晶上每一位上显示的数据
{
  ReadDS3231();
  dis1[0] = '2';
  dis1[1] = '0';
  dis1[2] = 0x30 + year / 10;
  dis1[3] = 0x30 + year % 10;
  dis1[4] = 0;
  dis1[5] = 0x30 + month / 10;
  dis1[6] = 0x30 + month % 10;
  dis1[7] = 1;
  dis1[8] = 0x30 + date / 10;
  dis1[9] = 0x30 + date % 10;
  dis1[10] = 2;
  dis1[11] = ' ';
  dis1[12] = ' ';
  switch (week)
  {
    case 1: {
        dis1[13] = 'M';
        dis1[14] = 'o';
        dis1[15] = 'n';
      }
      break;
    case 2: {
        dis1[13] = 'T';
        dis1[14] = 'u';
        dis1[15] = 'e';
      }
      break;
    case 3: {
        dis1[13] = 'W';
        dis1[14] = 'e';
        dis1[15] = 'd';
      }
      break;
    case 4: {
        dis1[13] = 'T';
        dis1[14] = 'h';
        dis1[15] = 'u';
      }
      break;
    case 5: {
```

```
                dis1[13] = 'F';
                dis1[14] = 'r';
                dis1[15] = 'i';
            }
            break;
        case 6: {
                dis1[13] = 'S';
                dis1[14] = 'a';
                dis1[15] = 't';
            }
            break;
        case 7: {
                dis1[13] = 'S';
                dis1[14] = 'u';
                dis1[15] = 'n';
            }
            break;
    }
    dis2[0] = ' ';
    dis2[1] = 0x30 + hour / 10;
    dis2[2] = 0x30 + hour % 10;
    dis2[3] = ':';
    dis2[4] = 0x30 + minute / 10;
    dis2[5] = 0x30 + minute % 10;
    dis2[6] = ':';
    dis2[7] = 0x30 + second / 10;
    dis2[8] = 0x30 + second % 10;
    dis2[9] = ' ';
    dis2[10] = ' ';
    dis2[11] = 0x30 + temperature / 10;
    dis2[12] = 0x30 + temperature % 10;
    dis2[13] = '.';
    dis2[14] = 0x30 + 0;
    dis2[15] = 3;
}
//************************************************************


//1602 显示完整时间********************************************
void lcdplay()
{
    get_dis();
    int k;
    lcd.setCursor( 0, 0);
    for (k = 0; k < 16; k++)
        lcd.write(dis1[k]);
    lcd.setCursor( 0, 1);
    for (k = 0; k < 16; k++)
        lcd.write(dis2[k]);
}
```

```
//*******************************************************


//********************************************************
//程序初始化**********************************************
static struct pt pt1, pt2, pt3;
void setup()      {
  pinMode(7, OUTPUT); //设定舵机接口为输出接口
  digitalWrite(7, HIGH);
  tone(9, 700, 100);//蜂鸣器提示
  //  设置串行端口的数据速率
  Serial.begin(9600);
  //  设置指纹传感器串行端口的数据速率
  finger.begin(9600);
  // LCD 初始化
  lcd.init(); // initialize the lcd
  lcd.backlight(); //Open the backlight
  //舵机初始化
  pinMode(servopin, OUTPUT); //设定舵机接口为输出接口
  //指纹传感器检查
  if (finger.verifyPassword()) {

    lcd.setCursor(0, 0);
    lcd.print("Found fingerprint sensor!"); //发现指纹传感器
  } else {

    lcd.setCursor(0, 0);
    lcd.print("Did not find fingerprint sensor :(");    //没有发现指纹传感器
    lcd.setCursor(0, 1);
    lcd.print("Check and Rest !");
    tone(9, 700, 100);//蜂鸣器提示
    delay(1000);
    tone(9, 700, 100);//蜂鸣器提示
    delay(3000);
    digitalWrite(7, LOW);
    while (1);
  }
  diy();
  //下面开始 1602 欢迎页面
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Welcome to use");
  lcd.setCursor(0, 1);
  lcd.print("BY 666 ");
  lcd.write(4);
  lcd.print(" ");
  lcd.write(5);
  //下面开始线程初始化
  PT_INIT(&pt1);    //线程 1 初始化
  PT_INIT(&pt2);    //线程2初始化
  PT_INIT(&pt3);    //线程2初始化
  //下面舵机复位
  for (int i = 0; i < 50; i++) //发送50个脉冲
  {
    servopulse(90);      //引用脉冲函数
  }

  tone(9, 700, 100); //蜂鸣器提示完成
}


void loop() {
  protothread1(&pt1);    //执行线程1
  protothread1(&pt2);    //执行线程2
  protothread1(&pt3);    //执行线程2
  getFingerprintIDez() ; //读取指纹
  delay(10); //时间片，每片1秒，可根据具体应用设置大小
  //下面为计时器
  counter1++;
  counter2++;
  counter3++;
}


int getFingerprintIDez() {

  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK) {
    return -1;
  } else { //读取成功
    tone(9, 900, 300); //蜂鸣器提示
    counter3 = 0 ; //计时器清0
    counter1 = 0 ;
  }

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)    return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)
  { wrong();    //指纹错误
    return -1; //返回
  }
  int m ;
  m = finger.fingerID ; //得到指纹序号
  if (m != -1) {
    for (int i = 0; i < 50; i++) //发送50个脉冲，舵机解锁
    {
        servopulse(0);      //引用脉冲函数
    }
    tone(9, 1500, 300);//蜂鸣器提示
    //下面为 1602 提示
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("PASS");
    lcd.setCursor(0, 1);
    lcd.print("Found ID #"); lcd.print(m);
    Rtime = 0 ; //清除错误等待时间
  }
  return m; //返回
}


void wrong() {    //指纹错误
  tone(9, 500, 300); //蜂鸣器提示
  Rtime = Rtime + 1000 ; //计算错误等待时间
  //下面为 1602 显示
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("WRONG");
  lcd.setCursor(0, 1);
  lcd.print("Wait: "); lcd.print(Rtime / 1000); lcd.print(" s");
  counter2 = 0 ;//计时器复位
  delay(Rtime);    //等待时间
}
```