



本项目基于 Arduino 开发平台设计一款闹铃,实现百分百强制性起床功能。

5.1 功能及总体设计

本项目利用 FPM10A 模块的指纹录入和校对功能,通过提前输入所有被监测起床对象的指纹并生成指纹库,在闹铃响后要求指纹库中的所有指纹通过模块输入,才能停止蜂鸣器运作。这种方式要求被监测对象必须全体起床,实现了将拖延起床概率下降为零的功能。

本项目可以实现实时显示日期与具体时间、自由定时闹铃并强制起床、实时监测显示当日温度并给出相应穿衣指南及出门提醒等功能。

要实现上述功能需将作品分成四部分进行设计,即输入部分、处理部分、传输部分和输出部分。输入部分包括指纹和时间调整两部分,分别通过 FPM10A 指纹模块和三个按键实现;处理部分和传输部分通过 C++ 程序控制各个模块和 Arduino 开发板;输出部分使用 I²C1602 液晶显示屏实现。

1. 整体框架图

整体框架如图 5-1 所示。

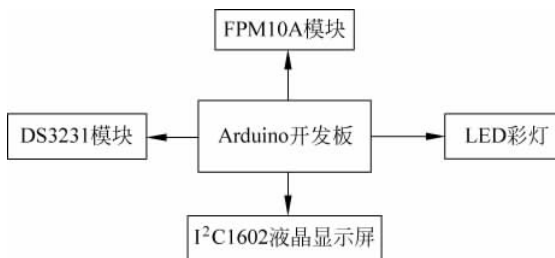


图 5-1 整体框架图

^① 本章根据王誉睿、崔鑫慰的项目设计整理而成。

2. 系统流程图

系统流程如图 5-2 所示。

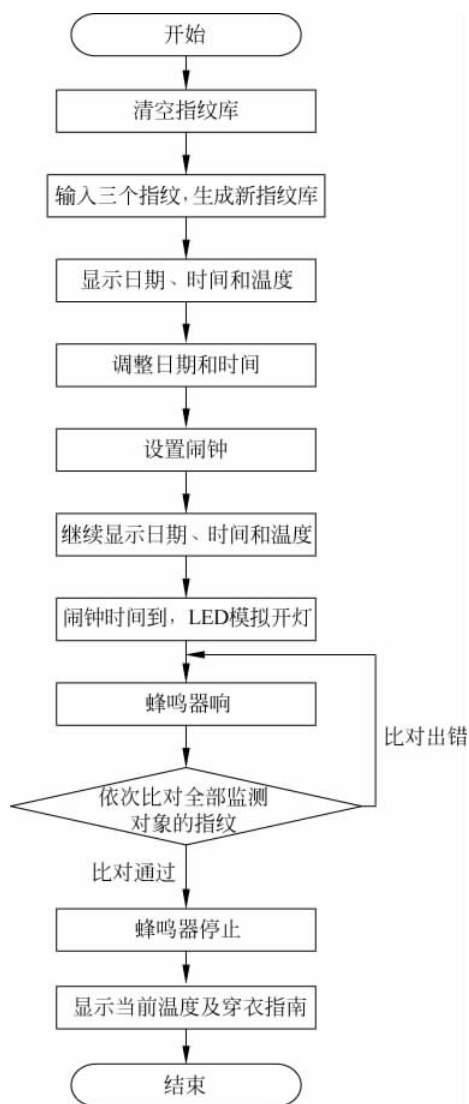


图 5-2 系统流程图

系统流程如图 5-2 所示。程序启动清理指纹库,输入指纹并生成库;更新日期、时间和温度、设置闹钟;当闹钟启动,LED 模拟开灯,蜂鸣器响铃,此时需要输入指纹以关闭闹铃,然后显示当前温度及穿衣指南。

3. 总电路图

总电路如图 5-3 所示,引脚连接如表 5-1 所示。本项目含三个按键,从左到右依次为整

体复位功能、设置转换功能(转换进入下一个数字的设置进程)、数字调节功能(以加一的方式调节当前数字)。

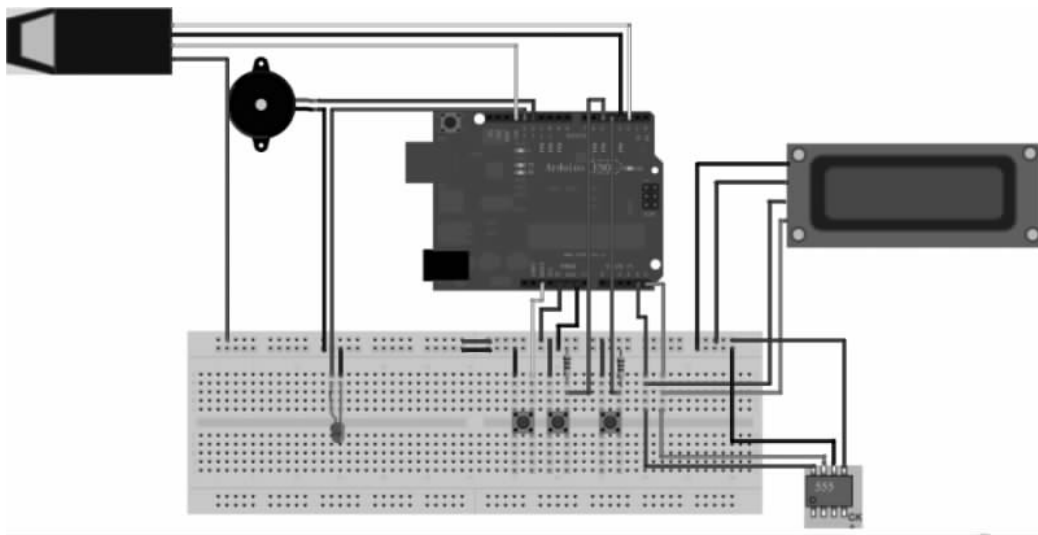


图 5-3 总电路图

表 5-1 引脚连接表

元件及引脚名		Arduino 开发板引脚
DS3231 时钟模块	SCL	A5
	SDA	A4
	VCC	5V
	GND	GND
I ² C1602 液晶显示屏	SCL	A5
	SDA	A4
	VCC	5V
	GND	GND
FPM10A 指纹模块	TXD	3
	RXD	2
	VCC	5V
	GND	GND
按键开关 1	右上引脚	RES
	左上引脚	GND
按键开关 2	右上引脚	5,通过 150Ω 电阻接 GND
	左上引脚	5V
按键开关 3	右上引脚	4,通过 150Ω 电阻接 GND
	左上引脚	5V

续表

元件及引脚名		Arduino 开发板引脚
无源蜂鸣器	正极	12
	负极	GND
彩色 LED	正极	13
	负极	GND

5.2 模块介绍

本项目主要包括 FPM10A 指纹模块、可调时钟及显示模块。下面分别给出各模块的功能介绍及相关代码。

5.2.1 FPM10A 指纹模块

本部分内容包括 FPM10A 指纹模块的功能介绍及相关代码。

1. 功能介绍

FPM10A 模块具有输入指纹、读取指纹、比对指纹、删除指纹、获取指纹容量、清理指纹库等多项功能。此模块通过数据包的形式与 Arduino 开发板进行通信,并通过两者的 TXD 引脚和 RXD 引脚传输信息。元件包括 FPM10A 模块、Arduino 开发板和导线若干,电路连接如图 5-4 所示。

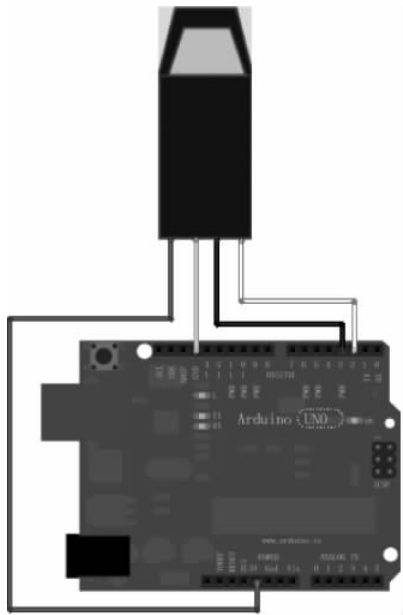


图 5-4 FPM10A 与 Arduino 开发板连线图

2. 相关代码

```

#include <SoftwareSerial.h>
#include "FPMXX.h"
uint8_t FPM_Address[] = { 0xFF,0xFF,0xFF,0xFF };
uint8_t FPM_Password[] = { 0x00,0x00,0x00,0x00 };
SoftwareSerial mySerial(3, 2);
FPMXX fs = FPMXX(&mySerial, FPM_Address, 9600);
uint16_t size = 0;
void setup()
{
    Serial.begin(9600);
    Serial.println(" -- -Test FPM10A Sensor -- -");
    /* ~~~~~ 清空指纹库 ~~~~~ */
    Serial.print("Clear fingers ");
    fs.emptyFinger();
    Serial.println("ok");
    /* ~~~~~ 输入三个指纹,生成新指纹库 ~~~~~ */
    for (int i = 1; i <= 3; i++)
    {
        Serial.print("Store finger ");
        Serial.println(i);
        delay(2000);
        while (fs.getFingerImage() != FPMXX_CODE_OK) {} //第一次扫描指纹
        fs.image2tz(1); //合成第一个特征模板并存放于 CharBuffer1 中
        delay(1000);
        while (fs.getFingerImage() != FPMXX_CODE_OK) {} //同一指纹再扫描一遍
        fs.image2tz(2); //合成第二个特征模板并存放于 CharBuffer2 中
        fs.buildTemplate(); //两遍特征合成模板
        Serial.print("Save on id:");
        Serial.print(i);
        fs.storeFinger(1, i); //保存指纹模板于指纹库中
        Serial.println(" ok");
    }
    /* ~~~~~ 比对并删除指纹 ~~~~~ */
    Serial.println("Now match the fingerprints");
    int i = 1;
    do
    {
        Serial.print("Match finger ");
        Serial.println(i);
        delay(2000);
        uint8_t r;
    }

```

```

r = fs.getFingerImage(); //扫描指纹
if (r == FPMXX_GET_IMAGE_NO_FINGER)
{
    Serial.println("No finger"); //没有检测到指纹
}
else if (r == FPMXX_GET_IMAGE_FAILED) //扫描失败
{
    Serial.println("Scan failed");
}
else //扫描成功后与指纹库中指纹依次进行比对
{
    fs.image2tz(1); //生成指纹特征
    uint16_t id = 0;
    if (fs.searchFinger(1, 0, 999, &id) == FPMXX_CODE_OK) //比对成功
    {
        Serial.print("Find finger: ");
        Serial.println(id);
        fs.deleteFinger(id,1); //比对成功后删除指纹库中该指纹
        i += 1;
    }
    else //比对失败
    {
        Serial.println("Finger Wrong");
    }
}
} while(i <= 3);
Serial.println("All fingers has been matched"); //所有指纹均比对成功
}
void loop()
{
}

```

5.2.2 可调时钟及显示部分

本部分内容包括可调时钟及显示部分的功能介绍及相关代码。

1. 功能介绍

本部分实现的功能是断电后保持计时,显示日期、时间和温度;调整日期、时间,设置闹钟,闹钟开启后蜂鸣器响起且LED点亮,根据当前温度显示穿衣指南。其中,DS3231和I²C1602用I²C总线与Arduino开发板的SDA、SCL引脚连接,三个按键从左到右依次为整体复位功能、设置转换功能(转换进入下一个数字的设置进程)、数字调节功能(以加一的方式调节当前数字)。元件包括DS3231模块、I²C1602液晶显示屏、三个按键、无源蜂鸣器、LED彩灯、Arduino开发板和导线若干,电路连接如图5-5所示。

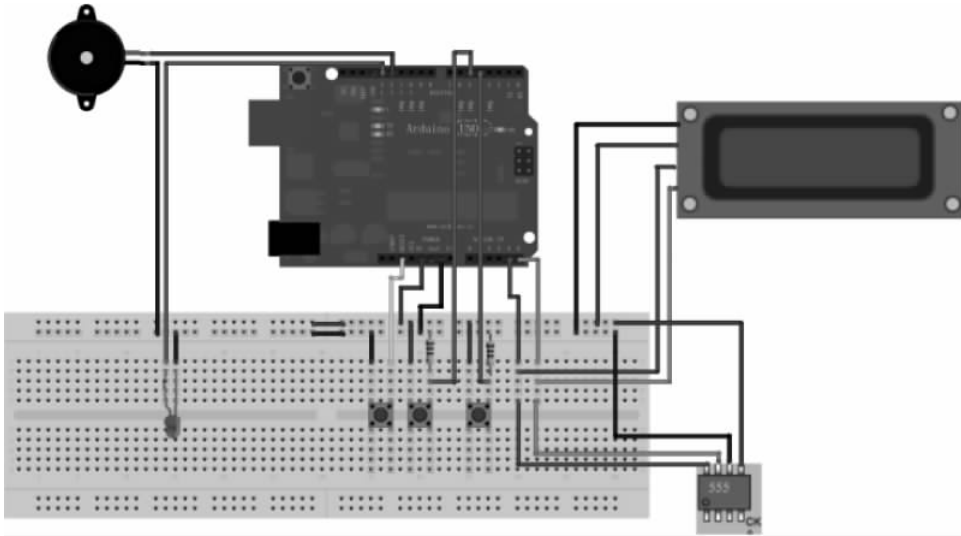


图 5-5 可调时钟及显示部分与 Arduino 开发板连接图

2. 相关代码

```

#include <Wire.h>
#include <DS3231.h>
#include <LiquidCrystal_I2C.h>
DS3231 Clock;
bool h12, PM, Century = false;
byte year, month, date, DoW, hour, minute, second;
bool ADy, A12h, Apm;
byte A1Day, A1Hour, A1Minute, A1Second, A1Bits;
uint8_t bell[8] = {0x4,0xe,0xe,0xe,0x1f,0x0,0x4}; //绘制闹钟标志
LiquidCrystal_I2C lcd(0x27,16,2);
const byte Set = 5; //按键
const byte Adj = 4;
byte button[2] = {Set, Adj};
/* ~~~~~ 显示时间 ~~~~~ */
void Pritime()
{
    int second,minute, hour,date,month,year,dow, temperature;
    second = Clock.getSecond();
    minute = Clock.getMinute();
    hour = Clock.getHour(h12, PM);
    date = Clock.getDate();
    month = Clock.getMonth(Century);
    year = Clock.getYear();
    dow = Clock.getDoW();
    temperature = Clock.getTemperature();
}

```

```
lcd.setCursor(0, 0); //光标停在第一行
lcd.print("20");
if (year >= 10) //显示年份
{
    lcd.print(year, DEC);
}
else
{
    lcd.print("0");
    lcd.print(year, DEC);
}
lcd.print(' - ');
lcd.setCursor(5, 0);
if (month >= 10) //显示月份
{
    lcd.print(month, DEC);
}
else
{
    lcd.print("0");
    lcd.print(month, DEC);
}
lcd.print(' - ');
lcd.setCursor(8, 0);
if (date >= 10) //显示日期
{
    lcd.print(date, DEC);
}
else
{
    lcd.print("0");
    lcd.print(date, DEC);
}
lcd.setCursor(11, 0);
switch (dow) //显示星期
{
    case 1:
        lcd.print("Mon");
        break;
    case 2:
        lcd.print("Tue");
        break;
    case 3:
        lcd.print("Wed");
        break;
    case 4:
        lcd.print("Thu");
```



```
        break;
    case 5:
        lcd.print("Fri");
        break;
    case 6:
        lcd.print("Sat");
        break;
    case 7:
        lcd.print("Sun");
        break;
    }
    lcd.setCursor(0, 1); //光标移至第 2 行
    if (hour >= 10) //显示小时
    {
        lcd.print(hour, DEC);
    }
    else
    {
        lcd.print("0");
        lcd.print(hour, DEC);
    }
    lcd.print(':');
    lcd.setCursor(3, 1);
    if (minute >= 10) //显示分钟
    {
        lcd.print(minute, DEC);
    }
    else
    {
        lcd.print("0");
        lcd.print(minute, DEC);
    }
    lcd.print(':');
    lcd.setCursor(6, 1);
    if (second >= 10) //显示秒钟
    {
        lcd.print(second, DEC);
    }
    else
    {
        lcd.print("0");
        lcd.print(second, DEC);
    }
    lcd.setCursor(9, 1);
    if (Clock.checkAlarmEnabled(1)) //显示闹钟标识
    {
        lcd.write(0x00);
    }
}
```

```

    }
    else
    {
        lcd.print(' ');
    }
    lcd.setCursor(12, 1);
    lcd.print(temperature);           //显示温度
    lcd.write(0xdf);
    lcd.print("C");
}
/* ~~~~~ 时间冒号闪烁 ~~~~~ */
void Tictime()
{
    lcd.setCursor(2, 1);
    lcd.print(' ');
    lcd.setCursor(5, 1);
    lcd.print(' ');
}
/* ~~~~~ 显示设置时间 ~~~~~ */
void PritimeSet (byte syear, byte smonth, byte sdate, byte sdow, byte shour, byte sminute, byte
ssecond)
{
    lcd.setCursor(0, 0);             //光标停在第一行
    lcd.print("20");
    if (syear >= 10)                 //显示年份
    {
        lcd.print(syear, DEC);
    }
    else
    {
        lcd.print("0");
        lcd.print(syear, DEC);
    }
    lcd.print(' - ');
    lcd.setCursor(5, 0);
    if (smonth >= 10)                //显示月份
    {
        lcd.print(smonth, DEC);
    }
    else
    {
        lcd.print("0");
        lcd.print(smonth, DEC);
    }
    lcd.print(' - ');
    lcd.setCursor(8, 0);
    if (sdate >= 10)                 //显示日期

```

```

    {
        lcd.print(sdate, DEC);
    }
else
    {
        lcd.print("0");
        lcd.print(sdate, DEC);
    }
lcd.setCursor(11, 0);
switch (sdow) //显示星期
    {
    case 1:
        lcd.print("Mon");
        break;
    case 2:
        lcd.print("Tue");
        break;
    case 3:
        lcd.print("Wed");
        break;
    case 4:
        lcd.print("Thu");
        break;
    case 5:
        lcd.print("Fri");
        break;
    case 6:
        lcd.print("Sat");
        break;
    case 7:
        lcd.print("Sun");
        break;
    }
lcd.setCursor(0, 1); //光标移至第 2 行
if (shour >= 10) //显示小时
    {
        lcd.print(shour, DEC);
    }
else
    {
        lcd.print("0");
        lcd.print(shour, DEC);
    }
lcd.print(':');
lcd.setCursor(3, 1);
if (sminute >= 10) //显示分钟
    {

```

```

        lcd.print(sminute, DEC);
    }
    else
    {
        lcd.print("0");
        lcd.print(sminute, DEC);
    }
    lcd.print(':');
    lcd.setCursor(6, 1);
    if (ssecond >= 10) //显示秒钟
    {
        lcd.print(ssecond, DEC);
    }
    else
    {
        lcd.print("0");
        lcd.print(ssecond, DEC);
    }
}
/* ~~~~~ 时间设置位闪烁 ~~~~~ */
void TicSet(byte c, byte l)
{
    lcd.setCursor(c, l);
    lcd.print(" ");
}
/* ~~~~~ 显示闹钟 ~~~~~ */
void PriAl()
{
    lcd.setCursor(0, 0); //光标移至第 1 行
    lcd.print("Alarm ");
    if (Clock.checkAlarmEnabled(1)) //显示闹钟状态
    {
        lcd.print("On");
    }
    else
    {
        lcd.print("Off");
    }
    Clock.getAlTime(AlDay, AlHour, AlMinute, AlSecond, AlBits, ADy, Al2h, Apm);
    lcd.setCursor(0, 1); //光标移至第 2 行
    if (AlHour >= 10) //显示闹钟小时
    {
        lcd.print(AlHour, DEC);
    }
    else
    {
        lcd.print("0");
    }
}

```

```

    lcd.print(A1Hour,DEC);
  }
  lcd.print(':');
  lcd.setCursor(3, 1);
  if (A1Minute >= 10) //显示闹钟分钟
  {
    lcd.print(A1Minute,DEC);
  }
  else
  {
    lcd.print("0");
    lcd.print(A1Minute,DEC);
  }
  lcd.print(':');
  lcd.setCursor(6, 1);
  if (A1Second >= 10) //显示闹钟秒钟
  {
    lcd.print(A1Second,DEC);
  }
  else
  {
    lcd.print("0");
    lcd.print(A1Second,DEC);
  }
}
/* ~~~~~ 按键检测 ~~~~~ */
unsigned long LastP;
boolean buttonPress(byte button)
{
  if(digitalRead(button) == HIGH) //检测按键是否按下
  {
    unsigned long NowP = millis();
    unsigned long buttonChange = NowP - LastP; //与上次按键时间进行对比
    if (buttonChange > 500) //按键间隔是否大于 500ms
    {
      LastP = millis(); //记录按键按下的时间
      return true;
    }
  }
  else
  {
    return false;
  }
}
else
{
  return false;
}
}

```

```

}
/* ~~~~~ 调整日期和时间 ~~~~~ */
void SetTime()
{
    bool h12, PM, Century = false;
    byte syear, smonth, sdate, sdow, shour, sminute, ssecond;
    if(buttonPress(Set))
    {
        lcd.clear();
        delay(200);
        Clock.getTime(syear, smonth, sdate, sdow, shour, sminute, ssecond);
        bool SetY = true;
/* ~~~~~ 进入年份设置状态 ~~~~~ */
        while(SetY)
        {
            unsigned long Rt = millis() % 1000;
            if(Rt < 500) //每隔 500ms 闪烁一次
            {
                PritimeSet(syear, smonth, sdate, sdow, shour, sminute, ssecond);
            }
            else
            {
                lcd.setCursor(0,0);
                lcd.print(" ");
            }
/* ~~~~~ 设置年份 ~~~~~ */
            if(buttonPress(Adj))
            {
                syear = syear + 1;
            }
/* ~~~~~ 进入月份设置状态 ~~~~~ */
            if(buttonPress(Set))
            {
                lcd.clear();
                bool SetMon = true;
                while(SetMon)
                {
                    unsigned long Rt = millis() % 1000;
                    if(Rt < 500)
                    {
                        PritimeSet(syear, smonth, sdate, sdow, shour, sminute, ssecond);
                    }
                    else
                    {
                        TicSet(5,0);
                    }
/* ~~~~~ 设置月份 ~~~~~ */

```

```

    if(buttonPress(Adj))
    {
        smonth = smonth + 1;
        if(smonth > 12)
        {
            smonth = 1;
        }
    }
}
/* ~~~~~ 进入日期设置状态 ~~~~~ */
if(buttonPress(Set))
{
    lcd.clear();
    bool SetD = true;
    while(SetD)
    {
        unsigned long Rt = millis() % 1000;
        if(Rt < 500)
        {
            PritimeSet(syear, smonth, sdate, sdow, shour, sminute, ssecond);
        }
        else
        {
            TicSet(8, 0);
        }
    }
}
/* ~~~~~ 设置日期 ~~~~~ */
if(buttonPress(Adj))
{
    sdate = sdate + 1;
    if(sdate > 31)
    {
        sdate = 1;
    }
}
}
/* ~~~~~ 进入星期设置状态 ~~~~~ */
if(buttonPress(Set))
{
    lcd.clear();
    bool SetW = true;
    while(SetW)
    {
        unsigned long Rt = millis() % 1000;
        if(Rt < 500)
        {
            PritimeSet(syear, smonth, sdate, sdow, shour, sminute, ssecond);
        }
        else
        {

```

```
        lcd.setCursor(11,0);
        lcd.print(" ");
    }
/* ~~~~~ 设置星期 ~~~~~ */
    if(buttonPress(Adj))
    {
        sdow = sdow + 1;
        if(sdow > 7)
        {
            sdow = 1;
        }
    }
/* ~~~~~ 进入小时设置状态 ~~~~~ */
    if(buttonPress(Set))
    {
        lcd.clear();
        long AdjP = 0;
        bool SetH = true;
        while(SetH)
        {
            unsigned long Rt = millis() % 1000;
            if(Rt < 500)
            {
                PvertimeSet(syear, smonth, sdate, sdow, shour, sminute, ssecond);
            }
            else
            {
                TicSet(0,1);
            }
        }
/* ~~~~~ 设置小时 ~~~~~ */
        if(buttonPress(Adj))
        {
            AdjP = AdjP + 1;
            shour = shour + 1;
            if(shour > 23)
            {
                shour = 0;
            }
        }
/* ~~~~~ 进入分钟设置状态 ~~~~~ */
        if(buttonPress(Set))
        {
            lcd.clear();
            bool SetMin = true;
            while(SetMin)
            {
                unsigned long Rt = millis() % 1000;
```



```

    if(Rt < 500)
    {
        PritimeSet(syear, smonth, sdate, sdow, shour, sminute, ssecond);
    }
    else
    {
        TicSet(3,1);
    }
/* ~~~~~ 设置分钟 ~~~~~ */
    if(buttonPress(Adj))
    {
        AdjP = AdjP + 1;
        sminute = sminute + 1;
        if(sminute > 59)
        {
            sminute = 0;
        }
    }
/* ~~~~~ 进入秒钟设置状态 ~~~~~ */
    if(buttonPress(Set))
    {
        lcd.clear();
        bool SetS = true;
        while(SetS)
        {
            unsigned long Rt = millis() % 1000;
            if(Rt < 500)
            {
                PritimeSet(syear, smonth, sdate, sdow, shour, sminute, ssecond);
            }
            else
            {
                TicSet(6,1);
            }
        }
/* ~~~~~ 设置秒钟 ~~~~~ */
        if(buttonPress(Adj))
        {
            AdjP = AdjP + 1;
            ssecond = ssecond + 1;
            if(ssecond > 59)
            {
                ssecond = 0;
            }
        }
/* ~~~~~ 保存并退出时间设置 ~~~~~ */
        if(buttonPress(Set))
        {

```

```

    Clock.setYear(syear);
    Clock.setMonth(smonth);
    Clock.setDate(sdate);
    Clock.setDoW(sdow);
    if (AdjP > 0)
    {
        Clock.setHour(shour);
        Clock.setMinute(sminute);
        Clock.setSecond(ssecond);
    }
/* ~~~~~ 进入闹钟小时设置状态 ~~~~~ */
    lcd.clear();
    delay(200);
    bool SetA1H = true;
    while(SetA1H)
    {
        unsigned long RA1 = millis() % 1000;
        if(RA1 < 500) //每 500ms 闪烁一次
        {
            PriA1();
        }
        else
        {
            TicSet(0,1);
        }
    }
/* ~~~~~ 设置闹钟小时 ~~~~~ */
    if(buttonPress(Adj))
    {
        {
            Clock.getA1Time(A1Day, A1Hour, A1Minute, A1Second, A1Bits, ADy, A12h, Apm);
            A1Hour = A1Hour + 1;
            if(A1Hour < 24)
            {
                Clock.setA1Time(A1Day, A1Hour, A1Minute, A1Second, B1000, ADy, false, Apm);
                //设置闹钟每天响一次
            }
            else
            {
                {
                    A1Hour = 0;
                    Clock.setA1Time(A1Day, A1Hour, A1Minute, A1Second, B1000, ADy, false, Apm);
                }
            }
        }
    }
/* ~~~~~ 进入闹钟分钟设置状态 ~~~~~ */
    if(buttonPress(Set))
    {
        {
            lcd.clear();
            bool SetA1M = true;
            while(SetA1M)

```

```

    {
    unsigned long RA1 = millis() % 1000;
    if(RA1 < 500)
    {
        PriA1();
    }
    else
    {
        TicSet(3,1);
    }
}
/* ~~~~~ 设置闹钟分钟 ~~~~~ */
if(buttonPress(Adj))
{
Clock.getA1Time(A1Day, A1Hour, A1Minute, A1Second, A1Bits, ADy, A12h, Apm);
A1Minute = A1Minute + 1;
if(A1Minute < 60)
{
Clock.setA1Time(A1Day, A1Hour, A1Minute, A1Second, B1000, ADy, false, Apm);
}
else
{
A1Minute = 0;
Clock.setA1Time(A1Day, A1Hour, A1Minute, A1Second, B1000, ADy, false, Apm);
}
}
/* ~~~~~ 进入闹钟秒钟设置状态 ~~~~~ */
if(buttonPress(Set))
{
lcd.clear();
bool SetA1S = true;
while(SetA1S)
{
unsigned long RA1 = millis() % 1000;
if(RA1 < 500)
{
PriA1();
}
else
{
TicSet(6,1);
}
}
/* ~~~~~ 设置闹钟秒钟 ~~~~~ */
if(buttonPress(Adj))
{
Clock.getA1Time(A1Day, A1Hour, A1Minute, A1Second, A1Bits, ADy, A12h, Apm);
A1Second = A1Second + 1;
if(A1Second < 60)

```

```

        {
            Clock.setAlTime(A1Day, A1Hour, A1Minute, A1Second, B1000, ADy, false, Apm);
        }
        else
        {
            A1Second = 0;
            Clock.setAlTime(A1Day, A1Hour, A1Minute, A1Second, B1000, ADy, false, Apm);
        }
    }
}
/* ~~~~~ 进入闹钟开关设置状态 ~~~~~ */
if(buttonPress(Set))
{
    lcd.clear();
    bool SetA1R = true;
    while(SetA1R)
    {
        unsigned long RA1 = millis() % 1000;
        if(RA1 < 500)
        {
            lcd.setCursor(0, 0);
            PriA1();
        }
        else
        {
            lcd.setCursor(7, 0);
            lcd.print(" ");
        }
    }
}
/* ~~~~~ 设置闹钟开关 ~~~~~ */
if(buttonPress(Adj))
{
    digitalWrite(Adj, LOW);

    if(Clock.checkAlarmEnabled(1))
    {
        Clock.turnOffAlarm(1);
    }
    else
    {
        Clock.turnOnAlarm(1);
    }
}
}
/* ~~~~~ 保存并退出闹钟设置 ~~~~~ */
if(buttonPress(Set))
{
    lcd.clear();
    delay(200);
    goto exit;
}

```



```

        lcd.print("Jacket,Trousers");
    else if(temperature > 5 && temperature <= 13)
        lcd.print("Sweater,Overcoat");
    else if(temperature > - 10 && temperature <= 5)
        lcd.print("Down jacket");
    else if(temperature <= - 10)
        lcd.print("COLD ALERT!");
    else
        lcd.print("ABNORMAL!!!");
}
void setup()
{
    Wire.begin();
    lcd.init();
    lcd.backlight();
    lcd.createChar(0, bell);          //自定义闹钟标志
    pinMode(Set, INPUT);             //按键引脚
    pinMode(Adj, INPUT);
    pinMode(12, OUTPUT);             //蜂鸣器引脚
    pinMode(13, OUTPUT);             //LED 引脚
    bool a = false, b = false, c = false; //设置局部分量作为辅助实现功能
do
{
    unsigned long Rt = millis() % 1000;
    if(Rt < 500)                    //显示时间并设置冒号,每隔 500ms 闪烁一次
    {
        Pvertime();                 //显示当前日期、时间和温度
    }
    else
    {
        Tictime();                  //时间冒号闪烁
    }
    if(buttonPress(Set)) a = true;
}while(a == false);                //Set 按键按下后进入调整日期和时间进程
Clock.turnOffAlarm(1);             //关掉当前的闹钟
do
{
    unsigned long Rt = millis() % 1000;
    if(Rt < 500)                    //显示时间并设置冒号,每隔 500ms 闪烁一次
    {
        Pvertime();                 //显示当前日期、时间和温度
    }
    else
    {
        Tictime();                  //时间冒号闪烁
    }
}

```

```

    }
    SetTime(); //设置日期、时间和闹钟
    if (Clock.checkAlarmEnabled(1)) b = true;
}while(b == false); //闹钟打开后进入重新显示日期时间进程
do
{
    unsigned long Rt = millis() % 1000;
    if(Rt < 500) //显示时间并设置时间冒号每隔 500ms 闪烁一次
    {
        Pvertime(); //重新显示当前日期、时间和温度
    }
    else
    {
        Tictime(); //时间冒号闪烁
    }
    if(Clock.checkIfAlarm(1)) c = true;
}while(c == false); //闹钟时间到后蜂鸣器响、LED 点亮和显示穿衣指南
}
void loop()
{
    //蜂鸣器响
    delay(100); //延迟 100ms
    tone(12,2000,50); //在引脚 12 输出频率 2000Hz,维持 50ms
    delay(100);
    tone(12,2000,50);
    delay(100);
    tone(12,2000,50);
    delay(100);
    tone(12,2000,50);
    //LED 点亮模拟开灯
    digitalWrite(13,HIGH);
    //显示穿衣指南
    ClothingGuide();
    delay(500);
}

```

5.3 产品展示

产品整体外观如图 5-6 所示。上部是 Arduino 开发板,右上角是 FPM10A 指纹传感器,右下角是 I²C1602 液晶显示屏,面包板从左至右依次是无源蜂鸣器、RESET 按键开关、设置转换按键开关、数字调节按键开关、LED 彩灯,面包板右下角是 DS3231 时钟模块。

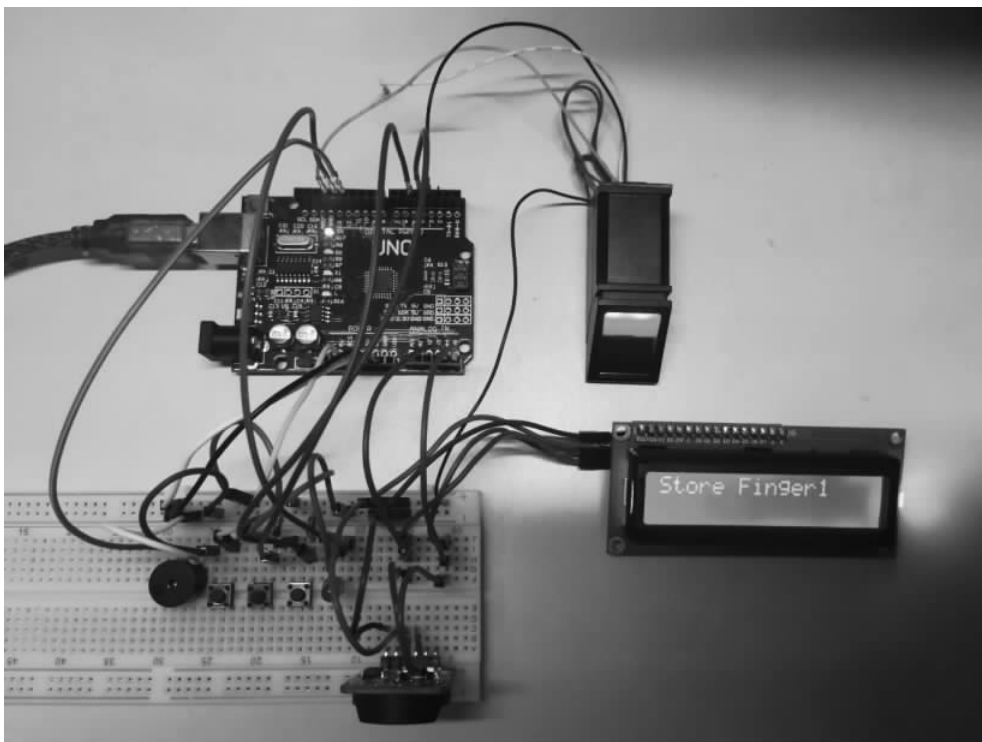


图 5-6 产品整体外观图

5.4 元件清单

完成本项目所用到的元件及其数量如表 5-2 所示。

表 5-2 元件清单

元件/测试仪表	数量
按键开关	3 个
FPM10A 指纹模块	1 个
DS3231 时钟模块	1 个
I ² C1602 液晶显示屏	1 个
无源蜂鸣器	1 个
LED 彩灯	1 个
150Ω 电阻	2 个
Arduino UNO 开发板	1 个
导线	若干
面包板	1 个