

电子科技大学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

创课总结报告



课程设计题目 智能台灯的设计

组长 张轶超 2014080107031

组员 黄泽熙 2014030302014

组员 冯齐 2014000202008

摘要

随着物联网的发展，智能家居越来越普及。台灯作为生活中必不可少的一部分，它的智能化发展也显得越来越重要。我们的产品，志在做最智能化的、最科技化的、最美观的台灯。我们的台灯包括全 3D 打印外壳、触控开关、光感变光、红外线感光、遥控控制、并且可以通过我们的 APP 来控制台灯，调节台灯的亮度，定时开关，以及报警模式。

第一章 硬件

1.1. 硬件设计

本次创课使用的硬件主要是 arduino 单片机。Arduino 是一个开源软硬件工具，用于电子产品设计。它包含可编程控制电路板（通常称为 微控制器）和一个在电脑上运行的集成开发环境(IDE),以编写和传送程序给电路板。

Arduino 单片机和 stc51 还有其他单片机不同的地方在于它的函数都在他的库函数之内，调用的时候不用重新定义，直接修改参数即可。所以在编程上带来许多方便。

1.2. 需求分析

本次设计用硬件主要实四个模块：触控开关，红外人体感知和光感和射频开关。

1.3. 具体过程

触控开关的主要设计思路就是当按键检测到有触控的时候，就开始调整亮度，亮度是由占空比来调节，当占空比到 30%达到最亮占空比降为 0%时达到最暗。

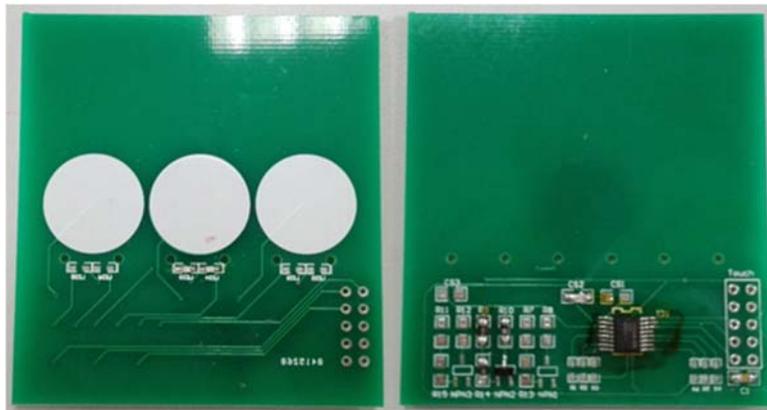


图 1-1 触控开关

红外人体感知模块：

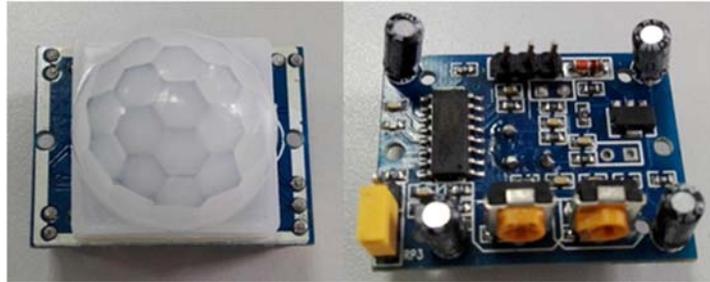


图 1-2 红外传感器

当红外传感器检测到有人靠近时，就输出高电平使灯点亮。否则，就为低电平。

光感模块：

实现灯光亮度根据外界变化而调节。当外界光强度变化时候，灯的亮度也会随着外界光亮度的变化而变化。

射频模块：

当给远程遥控器接上电的时候，遥控器就会一直发送高电平，接收器会收到高电平，就会给灯输出高电平，拔下电源，等就会灭。

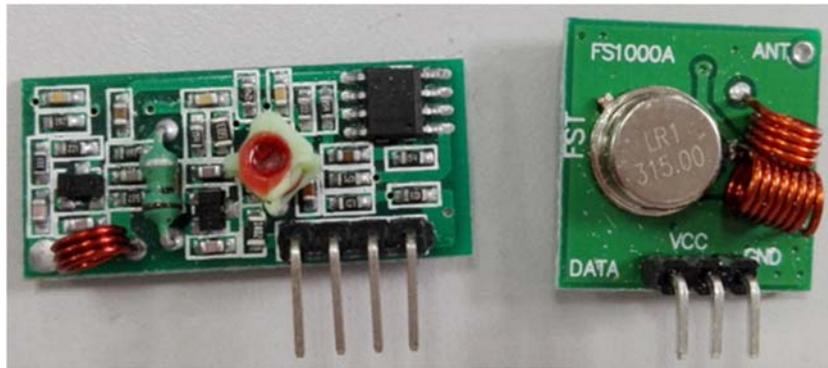


图 1-3 集成模块

集成模块：

集成模块是最难的，因为必须保证每一个模块独立运行互不影响。

有一种方法就是优先级，优先级最大的模块最优先运行，而优先级较低的模块在不满足优先级高的条件下才能运行。

第二种方法就是进行模块选择，用一个按键，一边接高电平，一遍接到数据都入口，当没有按下按键时，读入高电平。设置一个计数器，没读到一个电平的上升沿，计数器加1，计数器再模3用来选择模式。有触控模式，红外模式，光感模式和射频模式。

本次实验采用的是第二种方法。

发挥部分：

在试验中发现当手远离触控板一定距离的时候，K02 引脚也会读取到高电平。这时候我们

是并不希望它读取到高电平并进行亮度调节的。所以这时候我们写了一个按下检测的程序。思路就是 K02 引脚读取到高电平时，并不直接进行亮度的调节，而是经过一段时间（100ms）再次读取一个电平的值。如果此时 K02 还为高电平，就进行亮度调节。

1.4. 调试



图 1-4 当灯管强度稳定在 50，当光照改变的时候，亮度值随之而变。

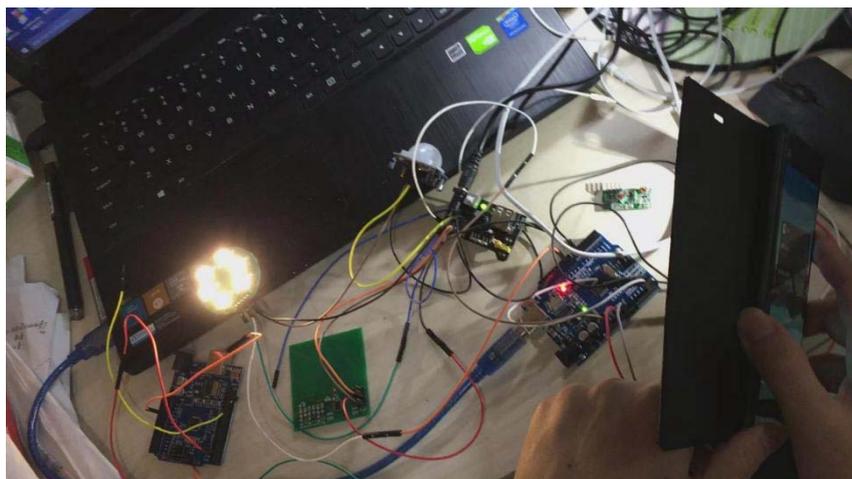


图 1-5 用手机调节外界灯光

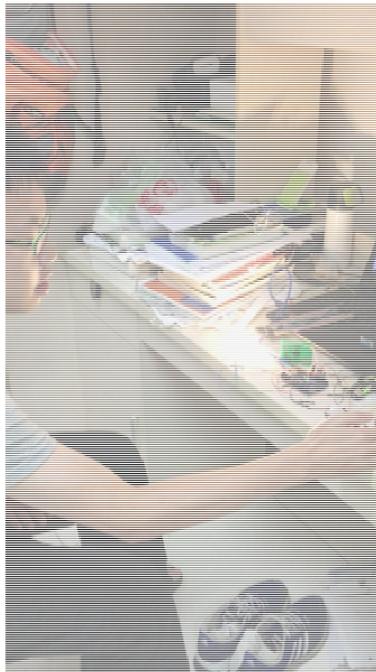


图 1-6 手在运动的同时，红外感知器感知到运动，灯骤然变亮

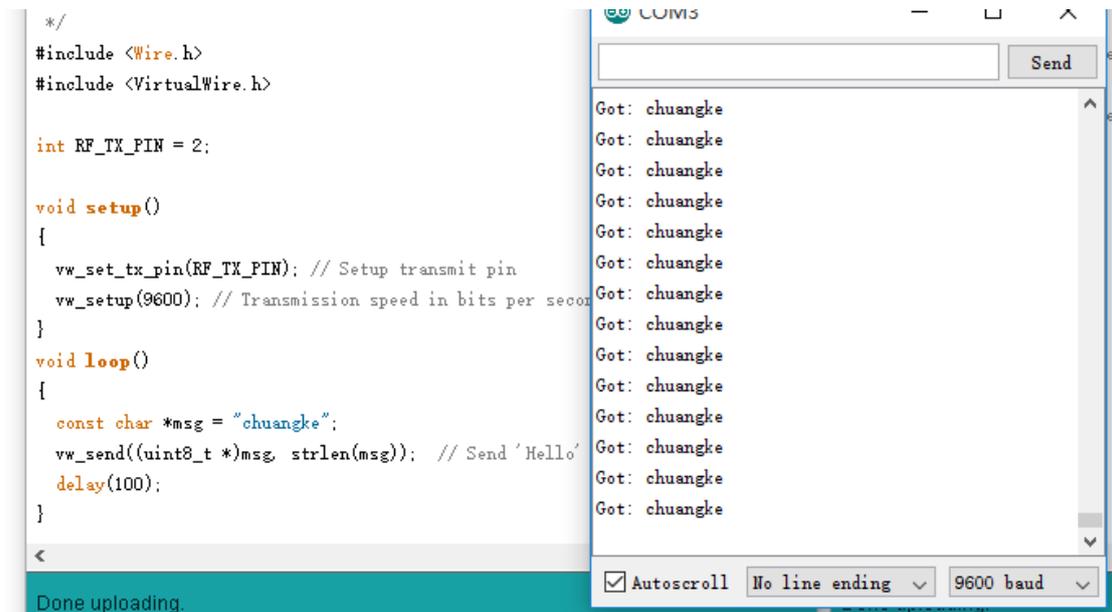


图 1-7 射频模块接收到的信息

1.5. WIFI 模块

本次设计中，我们成功地利用烧录装置为创课所提供的 WIFI 模块 ESP8266 进行了烧录。烧录完成后，手机和其他设备均可以正常连接 WIFI 模块的 WIFI 广播，其广播名为 XPG-GAgent-2FAE。

第二章 3D 打印

2.1. 设计软件

本此设计三维图建模使用 Solidworks, SolidWorks 软件是世界上第一个基于 Windows 开发的三维 CAD 系统. SolidWorks 遵循易用、稳定和创新三大原则, 使用它, 能够大大缩短设计时间。

2.2. 设计理念

设计理念来源于小时候看的一部电视剧-宝莲灯。模仿宝莲灯的粗细相间的灯杆设计, 与之不同的是在灯罩的设计上, 我们采用的是蜂窝状的柱形灯罩, 而非莲花状的灯罩。

2.3. 设计步骤

2.3.1. 灯罩设计

灯罩由三部分组成:

图 2-1 和图 2-2 是灯的外罩, 完美的尺寸和契合度, 可以将灯完整的镶嵌其中。具有弧度的设计, 使外形看起来简单、美观。

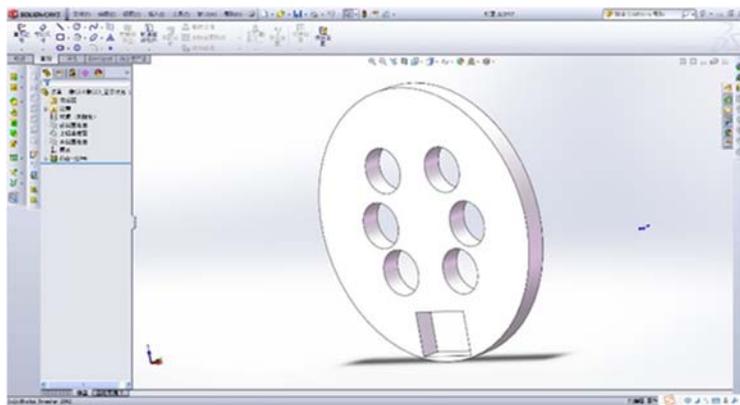


图 2-1 灯的外罩图 1

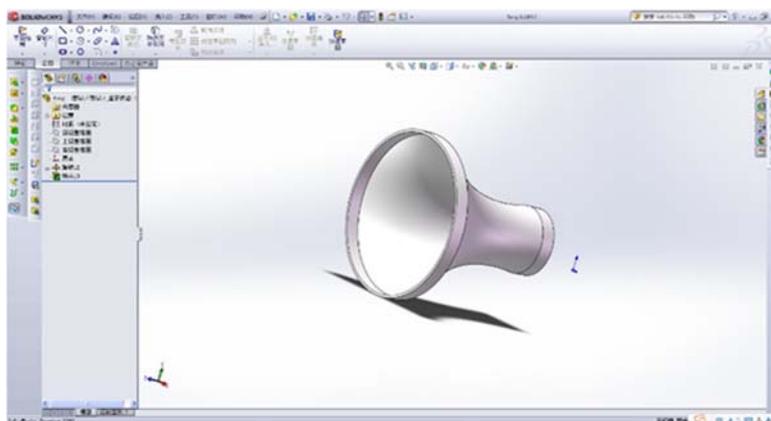


图 2-2 灯的外罩图 2

图 2-3 是架构在灯罩上的罩子。蜂窝状的全镂空设计，不仅能达到使光射向各个方向作用，而且美观大方。男女都可以使用的设计，将会增加商品的销售机会。

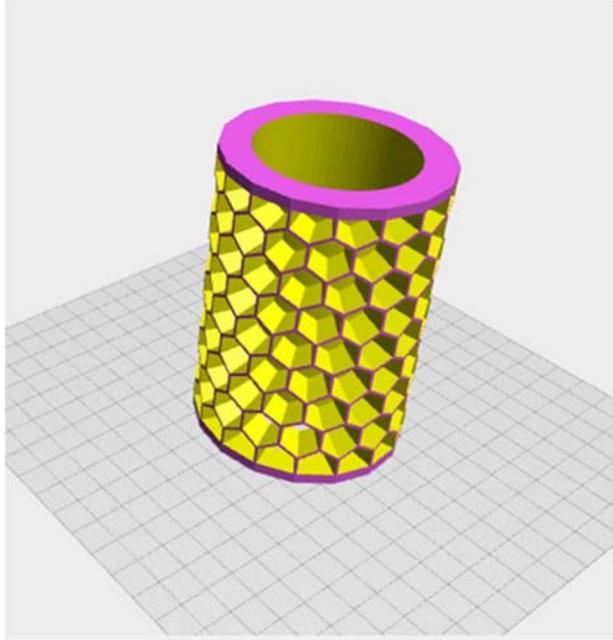


图 2-3 灯罩

2.3.2. 灯杆设计

灯杆的设计模仿宝莲灯而来，粗细相间的设计使产品看起来凹凸有致，空壳的设计方便了台灯的走线，宽大的地盘使台灯稳重大方，底盘上的开口为红外感应球的安装提供了方便。

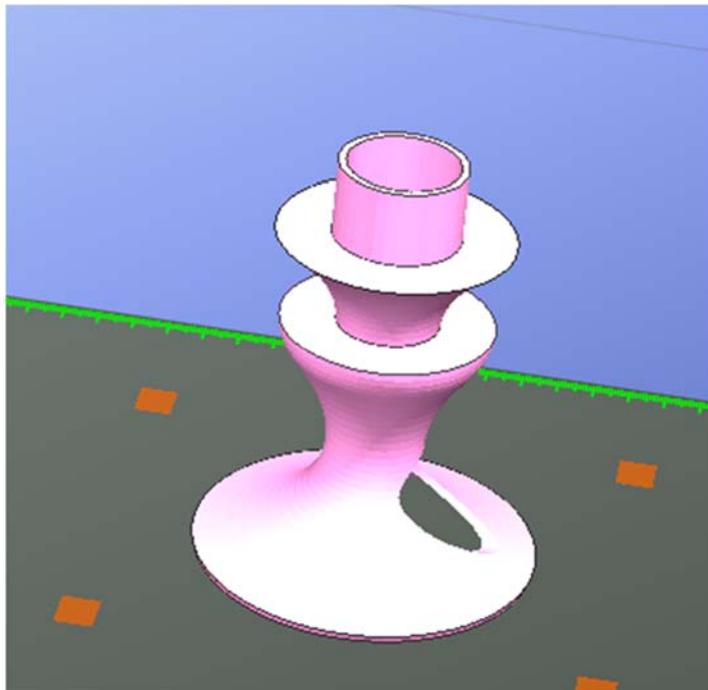


图 2-4 灯罩

2.3.3. 灯台设计

灯台的设计是空壳状的长方体，主要是为了增大体积，来装盛所有的电路板。
图 5 是盖子，圆孔状的洞是和灯杆相契合的，方便了走线。两个方形的孔洞是触摸屏的放置地点。图 6 是主题，下方的两个方形孔洞是为了方便 USB 线的接通，上方的洞是光感的放置地点。

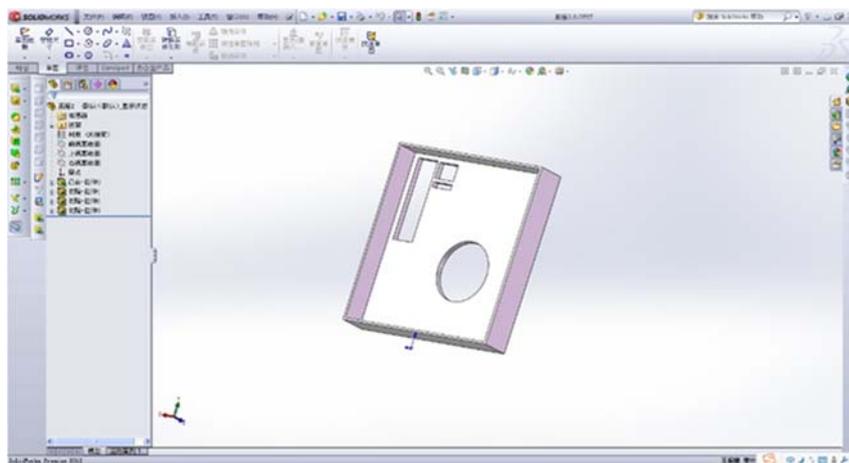


图 2-5 灯台 1

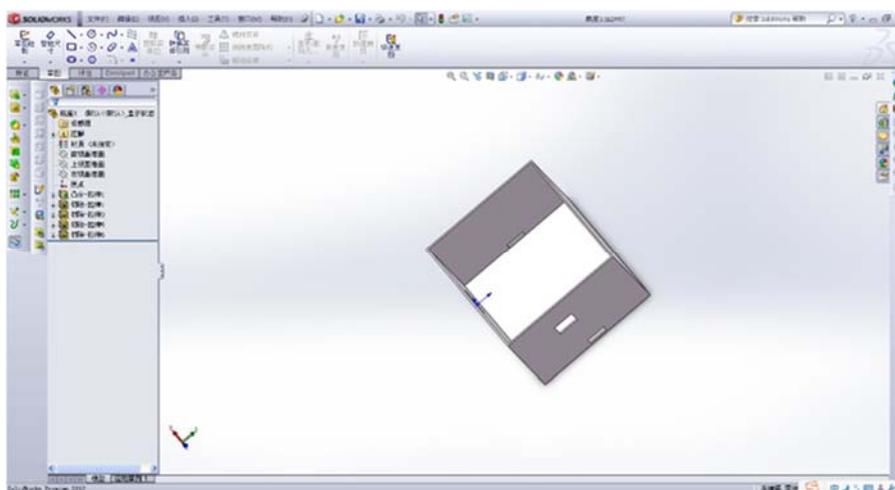


图 2-6 灯台 2

2.4. 打印

本次 3D 模型全部由 up box 打印机制作，up box 是一款方便使用、操作简捷的打印机，可以满足最大 25cm 的设计图纸。

第三章 安卓开发

3.1. 简介

本次创课需要设计一个智能台灯的远程控制安卓应用程序，使其可通过 WIFI 模块与台灯通信，实现台灯的开关等基本功能与亮度调节等进阶功能。本设计以开源应用程序 dantang-Gizwits-SmartSocket_Android-master（下称 SmartSocket）为框架，实现了包括即时开关、定时开关、亮度调节、报警模式等功能。本技术报告将从开源应用程序的框架分析、用户界面优化与额外功能添加三个方面介绍本设计。

3.2. 开源应用程序的框架分析

SmartSocket 的整体项目工程结构如表 3-1。

com.gizwits.powersocket	-智能云插座独有代码，包含控制部分和侧边栏部分
com.gizwits.powersocket.activity.control	-智能云插座控制界面 activity
com.gizwits.powersocket.activity.slidebar	-智能云插座侧边栏 activity
com.gizwits.framework	-机智云设备开源 APP 框架，包含除控制 Activity 外的代码，暂时机智云实验室中的其他开源 APP 所用框架一致
com.gizwits.framework.activity	-机智云设备开源 APP 框架相关 activity
com.gizwits.framework.adapter	-机智云设备开源 APP 框架相关数据适配器
com.gizwits.framework.config	-机智云设备开源 APP 框架配置类
com.gizwits.framework.entity	-机智云设备开源 APP 框架实体类
com.gizwits.framework.sdk	-机智云设备开源 APP 框架操作 SDK 相关类
com.gizwits.framework.utils	-机智云设备开源 APP 框架工具类
com.gizwits.framework.widget	-机智云设备开源 APP 框架自定义控件
com.gizwits.framework.XpgApplication Application	-机智云设备开源 APP 框架自定义
com.xpg.XXX	-机智云通用开发 API
zxing	-第三方二维码扫描控件

表 3-1 SmartSocket 项目工程结构表

其中，二次开发中需要留意的主要包括 com.gizwits.powersocket 与 com.gizwits.framework.sdk，前者为主界面的，用户界面的美化与新功能的添加都基于该包，后者为通信所用的 sdk 包，在增加功能的通信部分需要修改其中对应的内容。下面基于这两个包中的 MainControlActivity.java 与 CmdCenter.java 介绍 Socket 的框架。

3.2.1. 响应函数与发送信号

MainControlActivity.java 依次定义了各个控件的响应函数，一个响应函数的范例如下

```
public void onClick(View v) {  
    if (mView.isOpen()) {
```

```

        return;
    }
    switch (v.getId()) {
    case R.id.btnPower:
        mCenter.cPowerOn(mXpgWifiDevice, !btnPower.isSelected());
        updatePower(!btnPower.isSelected());
        break;
    case R.id.ivMenu:
        mView.toggle();
        break;
    case R.id.btnAppointment:
        startActivity(new Intent(MainControlActivity.this,
AppointmentActivity.class));
        break;
    }
}

```

该响应函数定义了开关按键、菜单、与预约按键的响应。在开关按键的响应中，响应的内容调用了 CmdCenter.java 中的 cPowerOn 函数实现，而该函数就实现向设备发送信号更新设备状态。

3.2.2. 接收信号

在向设备发送信号后，如果设备能够接收信号并正常更新状态，则会反馈设备状态。在通过其他方法调整设备状态时（如红外线感应），也会向应用程序反馈设备状态。所有的状态反馈都通过 didReceiveData 函数进行唤醒，将值更新到 statuMap 中，接着调用调用 handler 的 RECEIVED 进行后续处理。

3.2.3. 界面更新

界面更新由 Handler 处理 UPDATA_UI 实现，其中会调用 update 系列函数，一个典型的函数如下

```

private void updatePower(boolean isPower) {
    if (isPower) {
        btnPower.setSelected(true);
    } else {
        btnPower.setSelected(false);
    }
}

```

值得注意的是，在应用程序界面开关并不会直接改变开关的状态，而是一直要等到信号传递到设备并得到设备的反馈后，才会更新界面的显示。应用程序和设备的信息交流与应用程序界面和设备状态更新的关系如图 3-1。

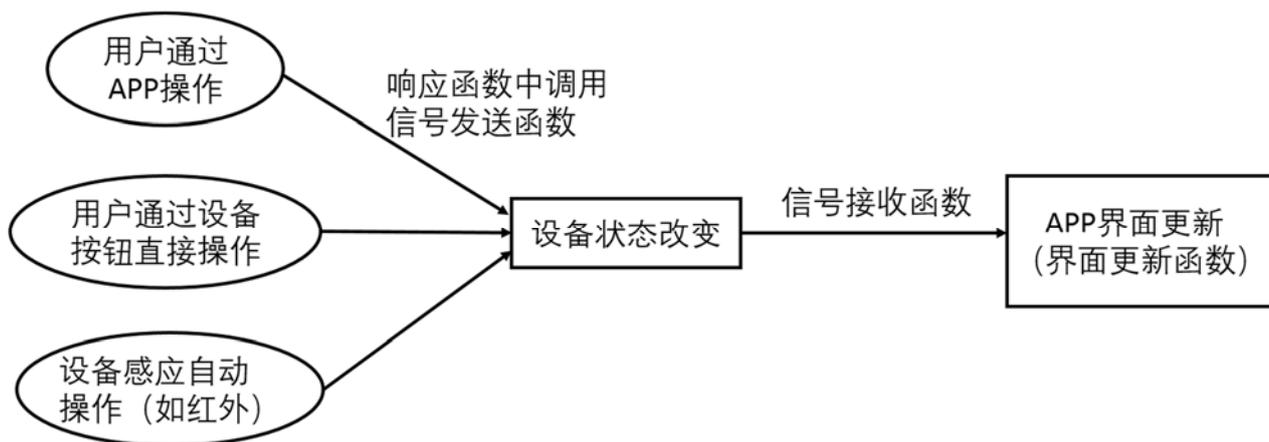


图 3-1 应用程序和设备的信息交流与应用程序界面和设备状态更新的关系图

3.3. 用户界面优化

3.3.1. 启动界面优化与应用图标更换

为了更好的体现智能灯的环保节能特性，将原本机智云系列的图标与启动界面全部更换成新的图标与启动界面，如图 3-2、3-3。



图 3-2 应用程序启动界面



图 3-3 应用程序图标

3.3.2. 主界面优化

对主控界面进行整体的界面重排，优化，更换图标及效果图，最终得到的开灯与关灯状态的整体样式如图 3-4、3-5.注意到在关灯模式下亮度调节/呼吸灯模块为灰色，处于不可用状态。



图 3-4 关灯下样式图



图 3-5 开灯下样式图

3.4. 额外功能的添加

由于基于的框架 powersocket 本身为控制智能插座的应用程序，因此仅具有定时与开关机功能，缺乏其他适用于台灯的功能，因此在对框架理解的基础上添加台灯专属的功能是很有必要的.以下将从亮度调节、报警模式与对话框管理三个功能介绍本应用程序相较于 powersocket 的优越性。

3.4.1. 亮度调节/呼吸灯/流水灯模块

本台灯的硬件设备可以支持无极调光，因此在应用程序中也设定了对应了亮度调节栏.根据安卓 MVC 的框架机制与 powersocket 通信架构图 3-1，该功能的添加可分为以下步骤。

- i. 修改主界面的布局文件 activity_main_control.xml，添加对应的控件。
- ii. 在主界面的控制文件中 MainControlActivity.java，依次添加控件的响应函数，信号接收函数与

界面更新函数.

- iii. 在配置文件 `Jsonkeys` 文件中添加亮度控件的对应项.
 - iv. 在信号发送控制文件 `CmdCenter.java` 中添加信号发送函数.
 - v. 在云端新建亮度的数据点 `Brightness`.
 - vi. 为了保证在关机状态下亮度不可调, 修改开关机的状态更新函数.
- 呼吸灯/流水灯模块的设置与亮度调节模块类似.

3.4.2. 报警模式

报警模式是利用本设备硬件红外感应功能的一个拓展.设备的红外功能使得当人进入台灯红外感应范围内时,自动亮灯,当人离开时自己熄灯.在报警模式开启的情况下,当人进入台灯的红外感应范围内时,灯会进入闪烁模式,同时蜂鸣器鸣响,起到报警的作用.报警模式的添加方式与亮度调节类似,这里不再赘述.

3.4.3. 对话框管理

由于台灯的部分功能再开启和关闭时会产生较大的影响,因此再开启或关闭部分功能时让用户再一次确定自己的选择是很有必要的.本设计中,当用户试图关闭台灯和开启报警模式时,都会弹出对话框让用户确认自己的选择,以防因为误触造成的不期待的后果.对话框的效果如图 3-6、3-7.



图 3-6 关闭设备时的对话框



图 3-7 开启报警模式的对话框

对话框管理的实现依赖于控制文件 `DialogManager.java`.在该控制文件中新增对应关闭设备与开启报警模式的 `message` 处理方式,并在 `MainControlActivity.java` 修改对应的响应函数,调用 `DialogManager` 进行处理.

3.5. 小结

本设计实现了智能台灯应用程序的用户界面重做美化,并在定时开关的基础上添加了亮度调节、报警模式与对话框提示等额外的功能,很好地完成了本次创课的安卓开发要求.

3.6. 附录

以下提供本次开发的代码核心部分.完整代码由于过于繁琐不便贴出,可联系作者 Eitima@163.com.

i. `MainControlActivity.java`

```
/**
 * Project Name:SmartBulb Final
 * File Name:MainControlActivity.java
 * Date:2016-07-13
 * Copyright (c) 2016 Eitima. All rights reserved.
 * Note: This is only the core part of the file. For the full part of this project,
contact the author at Eitima@163.com.
*/
    /** The m PowerOff dialog. */
    private Dialog mPowerOffDialog;
//Brightness
    /** The Brightness Seekbar */
    private SeekBar sbBrightness;
    /*The brightness layout*/
    private RelativeLayout rlBrightness;
    /*ImageView when PowerOff*/
    private ImageView ivPowerOff;
    /*The footer*/
    private LinearLayout llFooter;
    /*PowerOff Indicator*/
    private boolean isPowerOff=false;
    /*The Alarm Mode Switch*/
    private Switch sAlarmMode;
    /*The Flow Mode Switch*/
    private Switch sFlow;
    /*The Breath Mode Switch*/
    private Switch sBreath;
    /*The m Alarm Mode Dialog*/
    private Dialog mAlarmModeDialog;
    /*The Alarm Mode Indicator*/
```

```

private boolean isAlarmMode=false;
@SuppressLint("HandlerLeak")
Handler handler = new Handler() {
    public void handleMessage(Message msg) {
        super.handleMessage(msg);
        handler_key key = handler_key.values()[msg.what];
        switch (key) {
            case UPDATE_UI:
                if (mView.isOpen())
                    break;
                if (statuMap != null && statuMap.size() > 0) {
                    handler.removeMessages(handler_key.GET_STATUE_TIMEOUT.ordinal());
                    // 更新当前亮度
                    String brightness = (String) statuMap.get(JsonKeys.BRIGHTNESS);
                    if (!StringUtil.isEmpty(brightness)) {
                        updateBrightness(Integer.parseInt(brightness));
                    }
                    //更新报警模式开关
                    updateAlarmMode((Boolean) statuMap.get(JsonKeys.ALARM_MODE));
                    //更新呼吸灯模式开关
                    updateBreath((Boolean) statuMap.get(JsonKeys.BREATH));
                    //更新流水灯模式开关
                    updateFlow((Boolean) statuMap.get(JsonKeys.FLOW));
                }
                break;
        }
    }
};

private void initView() {
    sbBrightness = (SeekBar) findViewById(R.id.sbBrightness); //亮度的seekbar
    rlBrightness=(RelativeLayout)findViewById(R.id.rlBrightness);
    llFooter=(LinearLayout)findViewById(R.id.llFooter);
    ivPowerOff=(ImageView)findViewById(R.id.ivPowerOff);
    sAlarmMode=(Switch)findViewById(R.id.sAlarmMode); //报警模式的switch

    /*关机提示的dialog*/
    mPowerOffDialog = DialogManager.getPowerOffDialog(this, new OnClickListener()
    {
        @Override
        public void onClick(View arg0) {
            mCenter.cPowerOn(mXpgWifiDevice, false);
            DialogManager.dismissDialog(MainControlActivity.this, mPowerOffDialog);
        }
    });
}
// 报警模式的dialog

```

```

mAlarmModeDialog=DialogManager.getAlarmModeDialog(this, new OnClickListener(){

    @Override
    public void onClick(View arg0){
        mCenter.cAlarmMode(mXpgWifiDevice, true);
        DialogManager.dismissDialog(MainControlActivity.this,
mAlarmModeDialog);
    }
});
/*亮度的seekbarの設定*/
/*亮度信息输出*/
    sbBrightness.setOnTouchListener(new OnTouchListener() {
        @Override
        public boolean onTouch(View v, MotionEvent event) {
            // TODO Auto-generated method stub
            switch (event.getAction()) {
                case MotionEvent.ACTION_DOWN:

sbBrightness.getParent().requestDisallowInterceptTouchEvent(true);
                    break;
                case MotionEvent.ACTION_CANCEL:

sbBrightness.getParent().requestDisallowInterceptTouchEvent(false);
                    break;
            }
            return false;
        }
    });
/*停止拖动后, 由mcenter利用wifi发射当前的progress*/
    sbBrightness.setOnSeekBarChangeListener(new OnSeekBarChangeListener()
{

    @Override
    public void onStopTrackingTouch(SearchBar seekBar) {
        mCenter.cBrightness(mXpgWifiDevice, seekBar.getProgress());
    }

    @Override
    public void onStartTrackingTouch(SearchBar seekBar) {

    }

    @Override
    public void onProgressChanged(SearchBar seekBar, int progress,
boolean fromUser) {

```

```

        }
    });
    /*报警模式的响应函数的设定*/

    sAlarmMode.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
            // TODO Auto-generated method stub
            if(isChecked)
                mAlarmModeDialog.show();
            else
                mCenter.cAlarmMode(mXpgWifiDevice, false);
        }
    });
    /*呼吸灯模式的响应函数的设定*/

    sBreath.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {

        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
            // TODO Auto-generated method stub
            if(isChecked)
                mCenter.cBreath(mXpgWifiDevice, true);
            else
                mCenter.cBreath(mXpgWifiDevice, false);
        }
    });
}
/*流水灯模式的响应函数的设定*/

    sFlow.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {

        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
            // TODO Auto-generated method stub
            if(isChecked)
                mCenter.cFlow(mXpgWifiDevice, true);
            else

```

```

        mCenter.cFlow(mXpgWifiDevice, false);
    }
});

private void initEvents() {
private void updatePower(boolean isPower) {
    if (isPower) {
        btnPower.setSelected(true);
        /*调整brightness是否出现*/
        ivPowerOff.setVisibility(View.INVISIBLE);
        llFooter.setVisibility(View.VISIBLE);
    } else {
        btnPower.setSelected(false);
        powerOff();
    }
    isPowerOff=isPower;//用于dialog
}
private void powerOff(){
    Bitmap mBitmap = Bitmap.createBitmap(llFooter.getWidth(),
llFooter.getHeight(), Config.ARGB_8888);
    Canvas canvas = new Canvas(mBitmap);
    llFooter.draw(canvas);
    mBitmap = getTransparentBitmap(mBitmap, 50);
    ivPowerOff.setVisibility(View.VISIBLE);
    ivPowerOff.setImageBitmap(mBitmap);
    llFooter.setVisibility(View.INVISIBLE);
}
/*Brightness的update*/
private void updateBrightness(int num) {
    sbBrightness.setProgress(num);
}

/*Alarm Mode 的update*/
private void updateAlarmMode(boolean checked){
    sAlarmMode.setChecked(checked);
    isAlarmMode=checked;
}

/*Breath Mode的update*/
private void updateBreath(boolean checked){
    sBreath.setChecked(checked);
}

/*Flow Mode的update*/
private void updateFlow(boolean checked){
    sFlow.setChecked(checked);
}
}

```

```

@Override
public void onClick(View v) {
    if (mView.isOpen()) {
        return;
    }
    switch (v.getId()) {
    case R.id.btnPower:
        if (isPowerOff) {
            mPowerOffDialog.show();
        } else {
            mCenter.cPowerOn(mXpgWifiDevice, true); //通信,
        }
        break;
    }
}

```

ii. CmdCenter.java

```

/**
 * Project Name:SmartBulb Final
 * File Name:CmdCenter.java
 * Date:2016-07-13
 * Copyright (c) 2016 Eitima. All rights reserved.
 * Note: This is only the core part of the file. For the full part of this project,
 contact the author at Eitima@163.com.
 */

```

```

public void cBrightness(XPGWifiDevice xpgWifiDevice, int progress) {
    cWrite(xpgWifiDevice, JsonKeys.BRIGHTNESS, progress);
    new Timer().schedule(new timer(xpgWifiDevice), 1000);
}

public void cBreath(XPGWifiDevice xpgWifiDevice, boolean checked){
    cWrite(xpgWifiDevice,JsonKeys.BREATH,checked);
    new Timer().schedule(new timer(xpgWifiDevice), 1000);
}

public void cAlarmMode(XPGWifiDevice xpgWifiDevice,boolean checked){
    cWrite(xpgWifiDevice,JsonKeys.ALARM_MODE,checked);
    new Timer().schedule(new timer(xpgWifiDevice), 1000);
}

public void cFlow(XPGWifiDevice xpgWifiDevice,boolean checked){
    cWrite(xpgWifiDevice,JsonKeys.FLOW,checked);
    new Timer().schedule(new timer(xpgWifiDevice), 1000);
}

```

iii. DialogManger.java

```

/**
 * Project Name:SmartBulb Final

```

```
* File Name:DialogManager.java
* Date:2016-07-13
* Copyright (c) 2016 Eitima. All rights reserved.
* Note: This is only the core part of the file. For the full part of this project,
contact the author at Eitima@163.com.
```

```
*/
```

```
public static Dialog getPowerOffDialog(final Activity ctx, OnClickListener r) {
    final Dialog dialog = new Dialog(ctx, R.style.noBackgroundDialog) {
        };
    LayoutInflater inflater = LayoutInflater.from(ctx);
    View v = inflater.inflate(R.layout.dialog_power_off, null);
    Button leftBtn = (Button) v.findViewById(R.id.left_btn);
    Button rightBtn = (Button) v.findViewById(R.id.right_btn);
    leftBtn.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            dismissDialog(ctx, dialog);
        }
    });
    rightBtn.setOnClickListener(r);

    dialog.requestWindowFeature(Window.FEATURE_NO_TITLE);
    dialog.setCanceledOnTouchOutside(false);
    dialog.setCancelable(false);
    dialog setContentView(v);
    return dialog;
}
```

```
public static Dialog getAlarmModeDialog(final Activity ctx, OnClickListener r) {
    final Dialog dialog = new Dialog(ctx, R.style.noBackgroundDialog) {
        };
    LayoutInflater inflater = LayoutInflater.from(ctx);
    View v = inflater.inflate(R.layout.dialog_alarm_mode, null);
    Button leftBtn = (Button) v.findViewById(R.id.left_btn);
    Button rightBtn = (Button) v.findViewById(R.id.right_btn);
    leftBtn.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            dismissDialog(ctx, dialog);
        }
    });
    rightBtn.setOnClickListener(r);
}
```

```

dialog.requestWindowFeature(Window.FEATURE_NO_TITLE);
dialog.setCanceledOnTouchOutside(false);
dialog.setCancelable(false);
dialog setContentView(v);
return dialog;
}

```

iv. activity_main_control.xml

```

<!-- * Project Name:SmartBulb Final
* File Name:activity_main_control.xml
* Date:2016-07-13
* Copyright (c) 2016 Eitima. All rights reserved.
* Note: This is only the core part of the file. For the full part of this project,
contact the author at Eitima@163.com.
_ -->

```

```

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/rlHeader"
    android:background="@color/background_light_gray"
    android:scrollbars="none" >

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/background_light_gray"
        android:gravity="center_horizontal" >

        <Button
            android:id="@+id/btnPower"
            android:layout_width="200dp"
            android:layout_height="200dp"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="20dp"
            android:background="@drawable/selector_power"

            />

        <LinearLayout
            android:id="@+id/llConsumption"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"

```

```

        android:layout_marginTop="20dp"
        android:layout_below="@id/btnPower"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/consumption"
        android:textColor="@color/text_blue"
        android:textSize="18sp" />

    <TextView
        android:id="@+id/tvConsumption"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingLeft="5dp"
        android:textColor="@android:color/holo_blue_bright"
        android:textSize="18sp" />
</LinearLayout>
        <!-- 新添的亮度模块到此截止 -->
    <!-- 以下为新增的呼吸灯模块 -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <RelativeLayout
            android:id="@+id/rlBreath"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="0dp"
            android:orientation="horizontal"
            android:gravity="center">

            <TextView
                android:id="@+id/tvBreath"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:textColor="@color/text_blue"
                android:text="@string/breath"
                android:textSize="18sp"
            />
            <TextView
                android:id="@+id/tvSpace2"
                android:layout_width="50dp"

```

```

        android:layout_height="match_parent"
        android:visibility="invisible"
        android:layout_toRightOf="@id/tvBreath"
    />
    <Switch
        android:id="@+id/sBreath"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_toRightOf="@id/tvSpace2"
    />
</RelativeLayout>
<RelativeLayout
    android:id="@+id/rLFlow"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:orientation="horizontal"
    android:gravity="center">

    <TextView
        android:id="@+id/tvFlow"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:textColor="@color/text_blue"
        android:text="@string/flow"
        android:textSize="18sp" />
<!--
        android:layout_below="@id/tvBreath" -->

    <TextView
        android:id="@+id/tvSpace3"
        android:layout_width="50dp"
        android:layout_height="match_parent"
        android:visibility="invisible"
        android:layout_toRightOf="@id/tvFlow"
    />
    <Switch
        android:id="@+id/sFlow"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_toRightOf="@id/tvSpace3"
    />
</RelativeLayout>

</LinearLayout>
</LinearLayout>

```

```

<!--          新添的呼吸灯模块到此截止 -->
<!--          新添的亮度模块到此截止 -->
<!--          以下为新添的关机模块 -->
<ImageView
    android:id="@+id/ivPowerOff"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@id/LLFooter"
    android:layout_alignLeft="@id/LLFooter"
    android:layout_alignRight="@id/LLFooter"
    android:layout_alignTop="@id/LLFooter"
    android:visibility="invisible"
    android:background="@color/white"/>
<!--          新增的关机模块到此截止 -->
<!--          以下为新增的报警模块 -->
<RelativeLayout
    android:id="@+id/LLAlarmMode"
    android:layout_below="@id/LLFooter"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:orientation="horizontal"
    android:gravity="center">

    <TextView
        android:id="@+id/tvAlarmMode"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:textColor="@color/text_blue"
        android:text="@string/alarm_mode"
        android:textSize="18sp"
    />
    <TextView
        android:id="@+id/tvSpace"
        android:layout_width="50dp"
        android:layout_height="match_parent"
        android:visibility="invisible"
        android:layout_toRightOf="@id/tvAlarmMode"
    />
    <Switch
        android:id="@+id/sAlarmMode"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_toRightOf="@id/tvSpace"
    />

```

```
                </RelativeLayout>
<!--                新增的报警模式到此为止 -->

                </ScrollView>
            </RelativeLayout>
        </LinearLayout>
```