

# 西安邮电大学

## 毕业设计（论文）

题目： 基于嵌入式的智能灯光控制系统

学院： 自动化学院

专业： 电气工程及其自动化

班级： 电气 1304 班

学生姓名： 胡可欣

学号： 06134136

导师姓名： 何鹏举/邓颖娜 职称： 教授/讲师

起止时间： 2016 年 12 月 05 日 至 2017 年 06 月 10 日

## 毕业设计（论文）声明书

本人所提交的毕业论文《基于嵌入式的智能灯光控制系统》是本人在指导教师指导下独立研究、写作的成果，论文中所引用他人的文献、数据、图件、资料均已明确标注；对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明并表示感谢。

本人完全理解《西安邮电大学本科毕业设计（论文）管理办法》的各项规定并自愿遵守。

本人深知本声明书的法律责任，违规后果由本人承担。

论文作者签名：

日期：      年    月    日

## 西安邮电大学本科毕业设计(论文)选题审批表

|            |   |     |            |     |       |   |      |
|------------|---|-----|------------|-----|-------|---|------|
| 申 报 人      | 何鹏举/邓颖娜   | 职 称 | 教授/讲师      | 学 院 | 自动化学院 |   |      |
| 题目名称       | 基于嵌入式的智能灯光控制系统  |     |            |     |       |   |      |
| 题目来源       | 科研  |     |            |     | 教学    | √ | 其它   |
| 题目类型       | 硬件设计  | √   | 软件设计       |     | 论文    |   | 艺术作品 |
| 题目性质       | 应用研究  |     | √          |     | 理论研究  |   |      |
| 题目简述       | <p>智能灯光系统是对灯光进行智能控制与管理的系统,跟传统照明相比,可实现灯光软启、调光、一键场景、一对一遥控及分区灯光全开全关等管理,并可用遥控、定时、集中、远程等多种控制方式,从而达到智能照明的节能、环保、舒适、方便的功能。</p>  |     |            |     |       |   |      |
| 对学生知识与能力要求 | <p>掌握微机原理与接口技术及单片机原理及应用,理解嵌入式系统的基本组织结构与工作原理,具有一定的模拟电路技术基础及网络通信原理,熟悉常用传感器原理及使用方法,具有嵌入式软件设计能力及解决实际问题的动手能力。</p>  |     |            |     |       |   |      |
| 具体任务以及预期目标 | <p>设计实现智能灯光系统,能够通过 web 界面查询系统工作状态,对系统设备进行设置管理,设置管理区域,情景模式:整个照明系统的灯可以实现一键全开和一键全关的功能;通过日程管理模块,可以对灯光的定时开闭进行定义。例如,在每天早晨 7:00,将卧室的灯光缓缓开启到一个合适亮度;在深夜,自动关闭全部的灯光照明;对于固定模式的场景、您无需逐一地开关灯和调光,只进行一次编程,就可以按一个键控制一组灯,这就是场景设置功能。只需一次轻触操作即可实现多路灯光场景的转换;还可以得到想要的灯光和电器的组合场景,如回家模式、离家模式、会客模式、就餐模式、影院模式、夜起模式;通过不同的声响控制灯的开关。</p>   |     |            |     |       |   |      |
| 时间进度       | <p>2016 年 12 月 05 日—2016 年 12 月 10 日 选取毕设题目<br/>         2016 年 12 月 11 日—2017 年 01 月 06 日 查阅资料,撰写提交开题报告<br/>         2017 年 01 月 07 日—2017 年 03 月 04 日 确定系统设计方案<br/>         2017 年 03 月 05 日—2017 年 03 月 31 日 系统软件程序设计<br/>         2017 年 04 月 01 日—2017 年 04 月 15 日 软硬件的联合调试<br/>         2017 年 04 月 16 日—2017 年 04 月 31 日 系统功能完善<br/>         2017 年 05 月 01 日—2017 年 05 月 25 日 撰写毕业设计论文<br/>         2017 年 05 月 26 日—2017 年 06 月 01 日 修改、装订论文<br/>         2017 年 06 月 02 日—2017 年 06 月 10 日 准备毕业答辩</p> |     |            |     |       |   |      |
| 系(教研室)主任签字 | 年 月 日   |     | 主管院长<br>签字 |     | 年 月 日 |   |      |

# 西安邮电大学本科毕业设计（论文）开题报告

|      |         |    |                |      |           |
|------|---------|----|----------------|------|-----------|
| 学生姓名 | 胡可欣     | 学号 | 06134136       | 专业班级 | 电气 1304 班 |
| 指导教师 | 何鹏举/邓颖娜 | 题目 | 基于嵌入式的智能灯光控制系统 |      |           |

## 选题目的

随着信息技术的应用领域向着人类生活逐渐渗透，智能家居有着非常好的发展前景。智能灯光控制系统作为智能家居系统的一部分，也是依托家庭网络的一个智能系统。对于灯光控制系统的研究，将有助于推动整个智能家居系统的发展。

而人们生活质量的提高,使得灯具已不单纯是室内基本照明的工具,而且是建筑装饰的一种实用艺术品。当家里有各式各样的灯具之后,精心的搭配在一起,使其达到最适合气氛的效果是高品质生活的需要。同时目前灯光的控制主要还是手动形式,逐个地去控制所有的灯具,这样不仅麻烦而且效率低下,给人们的生活带来了一些不便,尤其是对于一切特殊人群而言,而且这也不符合现代人追求舒适生活的标准。因此,设计一个可以对灯光进行方便的控制,并且可以同时提供场景组合、远程查询等功能的智能化灯光系统不仅具有实用价值而且市场前景也相当广阔。可以用于多个方面,对人们的生活进行相对应的改变,也有利于现代科技的发展和生活的进步。

智能家居的开发与建设是 21 世纪科技发展的必然趋势。在照明领域,人们已经不能满足于单纯地提供亮度这一功能,而是面向系统控制方式的灵活和视觉上的艺术美感发展,智能照明系统就是在这样的背景下产生的。传统的控制方式能量流和信息流合一,控制简单、有效、直观,但其一旦布线完成后系统就不能再改动。相比之下,智能家居系统拥有安全、方便、高效、快捷、智能化和个性化的独特魅力,因此具有非常广阔的市场前景,相信具有在不久的将来就会在普通家庭中普及。照明灯光是生活中不可缺少的部分。那么智能照明控制系统也就是智能家居系统中不可缺少的一部分,对其的研究有很大的社会价值。

## 前期基础

已学课程：通过老师的讲解，基于嵌入式的智能灯光控制系统的设计需要掌握及复习参考已学课程微机原理与接口技术，单片机原理及应用，并且需要掌握理解嵌入式系统的基本组织结构与工作原理，复习已学课程模拟电路技术基础以及网络通信原理，学习了解传感器原理及使用方法，并且培养具有嵌入式软件的设计的能力以及需要解决的在实际的操作当中需要解决问题的动手能力。

掌握的工具：ARM 微控制器，对 RTC 与光敏传感器有一定了解。

资料积累：[1]周立功.ARM 微控制器基础与实战[M].北京航空航天大学出版社,2003

[2]李文杰.无线 LED 智能照明控制系统的低功耗研究与设计[D].华南理工大学

[3]李明亮，刘小龙，牟宏磊，徐宪清.基于 ARM11 的智能家居设计与实现[M].北京：北京航空 航天大学出版社，2013.5

软硬件条件：该课题主要是基于嵌入式系统，所以嵌入式处理器是核心，它负责控制和辅助系统的运行。所以应对嵌入式有详细的了解，而目前常见的嵌入式处理器有 x86、PowerPC、MIPS、ARM、SC-400、386EX 等，ARM 系列芯片有其强大的性能与低功耗的特点，众多的嵌入式操作系统都可以移植进 ARM 开发板中，主流的嵌入式操作系统有 VxWorks、Linux、WinCE、Android 等。设计者需选择 ARM 系类的芯片，并选择一种适合智能灯光控制的嵌入式操作系统。对于开发编译软件也可考虑为文本编译器，具体可视情况而定。

要研究和解决的问题

(1) 设计实现智能灯光系统, 能够通过 web 界面查询系统工作状态, 对系统设备进行设置管理, 设置管理区域, 情景模式: 整个照明系统的灯可以实现一键全开和一键全关的功能。

(2) 通过日程管理模块, 可以对特定的灯具的开闭的时间上进行自定义定义。例如, 在每天早晨 7:00, 将卧室的灯光缓缓开启到一个合适亮度; 在深夜, 自动关闭全部的灯光照明。

(3) 对于固定模式的场景、您无需逐一地开关灯和调光, 只进行一次编程, 就可以按一个键控制一组灯, 这就是场景设置功能。

(4) 只需一次轻触操作即可实现多路灯光场景的转换; 还可以得到想要的灯光和电器的组合场景, 如回家模式、离家模式、会客模式、就餐模式、party 模式、夜晚模式 1 和夜晚模式 2。

工作思路和方案

工作思路:

首先必须清楚, 智能家居系统是一个很庞大的课题, 而且涉及众多领域的知识, 而本课题主要设计的智能灯光控制系统将会以在智能家居系统中有代表性功能为主要的研究方向, 实现日常常用功能并便于扩展。

其次需要根据本课题的要求进行系统设计、流程设计, 并且对该课题涉及到的软件、硬件根据功能要求进行选型, 选择一个较为适合的元器件。并根据要求查阅相关资料进行。

然后在软硬件设计时进行分模块设计。

此外, 还要注意电路的设计, 在电路的设计当中应当注意原理图与实际图的差别, 提前进行排版与设计, 以防止后期出现排版不下, 或电路出现故障等问题, 注意电路设计的美观。

方案:

2016 年 12 月 05 日—2016 年 12 月 10 日 选取毕设题目

2016 年 12 月 11 日—2017 年 01 月 06 日 查阅资料, 撰写提交开题报告

2017 年 01 月 07 日—2017 年 03 月 04 日 确定系统设计方案

2017 年 03 月 05 日—2017 年 03 月 31 日 系统软件程序设计

2017 年 04 月 01 日—2017 年 04 月 15 日 软硬件的联合调试

2017 年 04 月 16 日—2017 年 04 月 31 日 系统功能完善

2017 年 05 月 01 日—2017 年 05 月 25 日 撰写毕业设计论文

2017 年 05 月 26 日—2017 年 06 月 01 日 修改、装订论文

2017 年 06 月 02 日—2017 年 06 月 10 日 准备毕业答辩

指导教师意见

签字:

年 月 日

# 西安邮电大学毕业设计（论文）成绩评定表

|                |  |    |   |    |          |          |           |
|----------------|--|----|---|----|----------|----------|-----------|
| 学生姓名           | 胡可欣  | 性别 | 女 | 学号 | 06134136 | 专业<br>班级 | 电气 1304 班 |
| 课题名称           | 基于嵌入式的智能灯光控制系统   |    |   |    |          |          |           |
| 指导<br>教师<br>意见 | <div>评分（百分制）：      指导教师(签字)： _____      年   月   日</div>                    |    |   |    |          |          |           |
| 评阅<br>教师<br>意见 | <div>评分（百分制）：      评阅教师(签字)： _____      年   月   日</div>                    |    |   |    |          |          |           |
| 验收<br>小组<br>意见 | <div>评分（百分制）：      验收教师(签字)： _____      年   月   日</div>                    |    |   |    |          |          |           |
| 答辩<br>小组<br>意见 | <div>评分（百分制）：      答辩小组组长(签字)： _____      年   月   日</div>                  |    |   |    |          |          |           |
| 评分比例           | 指导教师评分    (%) 评阅教师评分    (%) 验收小组评分    (%) 答辩小组评分    (%)                    |    |   |    |          |          |           |
| 学生总评<br>成绩     | 百分制成绩  |    |   |    | 等级制成绩    |          |           |
| 答辩委员<br>会意见    | <div>毕业论文(设计)最终成绩(等级)：</div> <div>学院答辩委员会主任(签字、学院盖章)：      年   月   日</div> |    |   |    |          |          |           |

## 摘 要

随着电子信息技术不断发展，其技术也不断的应用于人类的生产生活。现如今，人们对于生产生活的品质有了更深层次的追求，灯光的智能化是目前智能家居生活当中不可或缺的一部分。对于灯光的控制系统的深入的研究，对于推动整个智能家居系统的发展起到重要的作用。

本文首先对国内外的智能灯光控制系统现状、发展趋势及技术手段进行了分析研究，从而对设计智能灯光控制系统进行设计。介绍了在智能灯光控制系统所采用的一些较为重要的技术手段，与当前情况下的通信技术进行比较，将 ARM 处理器的优点结合起来，并且具体阐述了以 STM32F103Z 为核心的智能灯光控制器的设计。该灯光控制器具有灯光控制，场景设置等功能。提出较为具体的设计方案，并搭建硬件平台及相应外围电路模块，最终完成系统的主控制器的硬件模块。

其次是软件部分的设计与开发，智能家居系统软件为嵌入式软件，以 uCOS 操作系统为主要软件系统。在最终完成软件与硬件的设计后，对上述方案所设计的智能灯光控制系统进行性能及功能测试。

**关键词：**智能灯光控制；嵌入式处理器；ARM 微控制器；uCOS 操作系统

## ABSTRACT

With the continuous development of electronic information technology, and gradually applied to human daily life. Nowadays, people have deeply understanding for the pursuit of life quality. Intelligent lighting is an essential part of smart home. In-depth study of lighting control system may have positive impact on the development of the intelligent home systems.

This paper first present research status, trends and technology of intelligent lighting control system, Introduce some vital technology which intelligent lighting control system have adopt, compared with current communication technology, combined with the virtue of ARM processor. This article also elaborate the design of intelligent lighting controller with STM32F103Z as the core processor in detail. This lighting control module provide lighting control and scene control functionality. Finally, provide specific design solution with every circuit modules on the hardware platform.

The second is the design and development of software, intelligent Home Furnishing system software for the embedded software, the main software system in uCOS operating system. In the design of the final completion of software and hardware, intelligent lighting control system and the design scheme of the performance and function text.

**Keywords:** intelligent lighting control; embedded processor; ARM; uCOS



# 目 录

|                           |    |
|---------------------------|----|
| 第一章 引言.....               | 1  |
| 1.1 研究背景.....             | 1  |
| 1.2 发展趋势.....             | 1  |
| 第二章 智能灯光控制系统整体设计.....     | 2  |
| 2.1 系统总体任务分析.....         | 2  |
| 2.2 系统总体设计方案.....         | 2  |
| 第三章 智能灯光控制系统所用关键性技术.....  | 4  |
| 3.1 嵌入式技术.....            | 4  |
| 3.2 红外遥控技术.....           | 4  |
| 3.3 RTC 实时定时技术.....       | 5  |
| 第四章 嵌入式系统.....            | 6  |
| 4.1 嵌入式系统软件介绍.....        | 6  |
| 4.2 uCOS II 操作系统.....     | 7  |
| 第五章 智能灯光控制系统硬件设计.....     | 9  |
| 5.1 STM32F103ZET6 设计..... | 9  |
| 5.2 红外遥控设备.....           | 11 |
| 5.3 LCD 液晶显示屏.....        | 13 |
| 第六章 智能灯光控制系统软件设计.....     | 14 |
| 6.1 红外系统设计.....           | 14 |
| 6.2 实时时钟设计.....           | 20 |
| 6.3 LCD 模块设计.....         | 22 |
| 6.4 定时模块场景的设计.....        | 23 |
| 6.5 远程控制模块的设计.....        | 24 |
| 第七章 智能灯光控制系统测试.....       | 26 |

|                       |    |
|-----------------------|----|
| 7.1 红外模块测试.....       | 26 |
| 7.2 RTC 实时时钟模块测试..... | 28 |
| 7.3 上位机模块设测试.....     | 30 |
| 结束语.....              | 33 |
| 致 谢.....              | 34 |
| 参考文献.....             | 35 |

## 第一章 引言

### 1.1 研究背景

在本世纪 80 年代初期的时候，人们开始注意到这样一个问题，驾驶员的驾驶视野的好与坏或将要影响到其行车安全性能的高低。也正是由于这个问题的出现，大众为了改善行车安全性，降低交通事故的频繁发生。人们开始将视线转移到汽车灯光系统上面，希望通过对灯光的改进，减小危险事故的发生。基于这种状况，日本小系车灯公司领先进入到对智能灯光系统(ILS—Intelligent Lighting System)这个复杂的研究领域<sup>[1]</sup>。

### 1.2 发展趋势

现如今对于智能灯光控制系统的研究已经成为智能生活空间研究的必然趋势。一些经济发达的国家如新加坡、日本、欧美等率先开启智能生活方面的研究。不同文化与信息交流沟通使得人们对于传统生活的态度有很大转变，灯具已经不再像从前一样，仅仅是室内照明工具，还用在建筑风格领域当中，作为是一种艺术装潢品。现如今灯光控制还是以手动形式为主。单一去控制每个灯具，不仅不利于人们生活，特别是一些特殊人士，而且效率低下。所以设计一个较为灵活的灯具控制方式，或是场景控制方式尤为重要。这样智能灯光控制系统不仅具有较高实用价值，还可用于人们生产生活的多个方面，占有着较好的市场前景。

智能灯光控制系统主要利用了嵌入式技术、自动控制技术、传感器技术、网络通信技术。将灯光控制系统设计方案与人们的生活紧密联系起来，通过智能化产品使人们的生产生活更为便捷、高效、舒适、安全。

本次毕业设计的课题主要是基于嵌入式系统，其核心内容则是嵌入式处理器，负责的内容是控制系统运行或者是辅助系统运行。所以应对嵌入式有详细了解，本课题中主要采取 ARM 系类芯片，并选择 uCOS 操作系统。

## 第二章 智能灯光控制系统整体设计

### 2.1 系统总体任务分析

此次毕业设计主要基于 stm32f103zet6 开发板，本次控制系统开发板为最小系统开发板。通过移入 uCOS 操作系统，并在此系统上建立主要任务。设计实现智能化光控制系统，首先要对系统设备进行设置管理规划，设置管理区域，并且使整个室内空间照明灯光达到一键按下全开或关的效果。还需根据用户日程对于灯光的控制进行设计，通过定时来对于灯光进行准点开关。譬如，在清晨 7:00 点时，做到自动将所需房间灯光亮起；而在深夜 10:00 点时，自动关闭房间内部所有灯具。然是对场景模式进行设计，根据用户的生活工作需要，对室内生活空间的场景模式进行规划。

设计者可定义一些用户经常涉及的场景，这样用户就不用逐一开启或关闭所需房间灯光。设计者可根据需要编写相应编程，达到按下某个按键就可控制某几盏灯的功能。而且只要改不同按键操作就可实现不同灯光场景模式的变换。这样控制系统就可由不同房间的灯光组合成新的场景，如会客模式、就餐模式、party 模式、回家模式、离家模式。在此过程中还可应用红外遥控器，通过远程遥控达到预期效果。

### 2.2 系统总体设计方案

#### 2.2.1 硬件设计

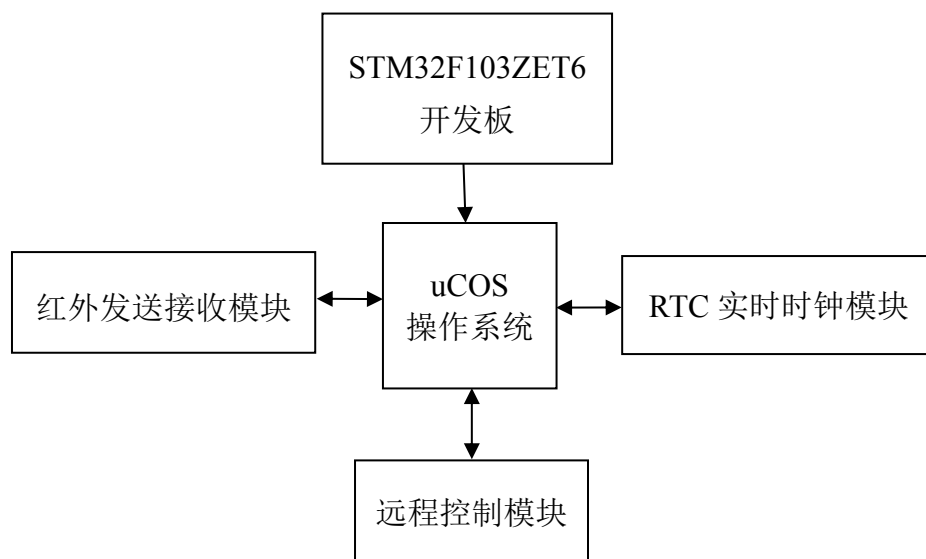


图 2.1 系统硬件设计结构图

### 2.2.2 软件设计

软件设计中的主要流程如下图 2.2 所示。

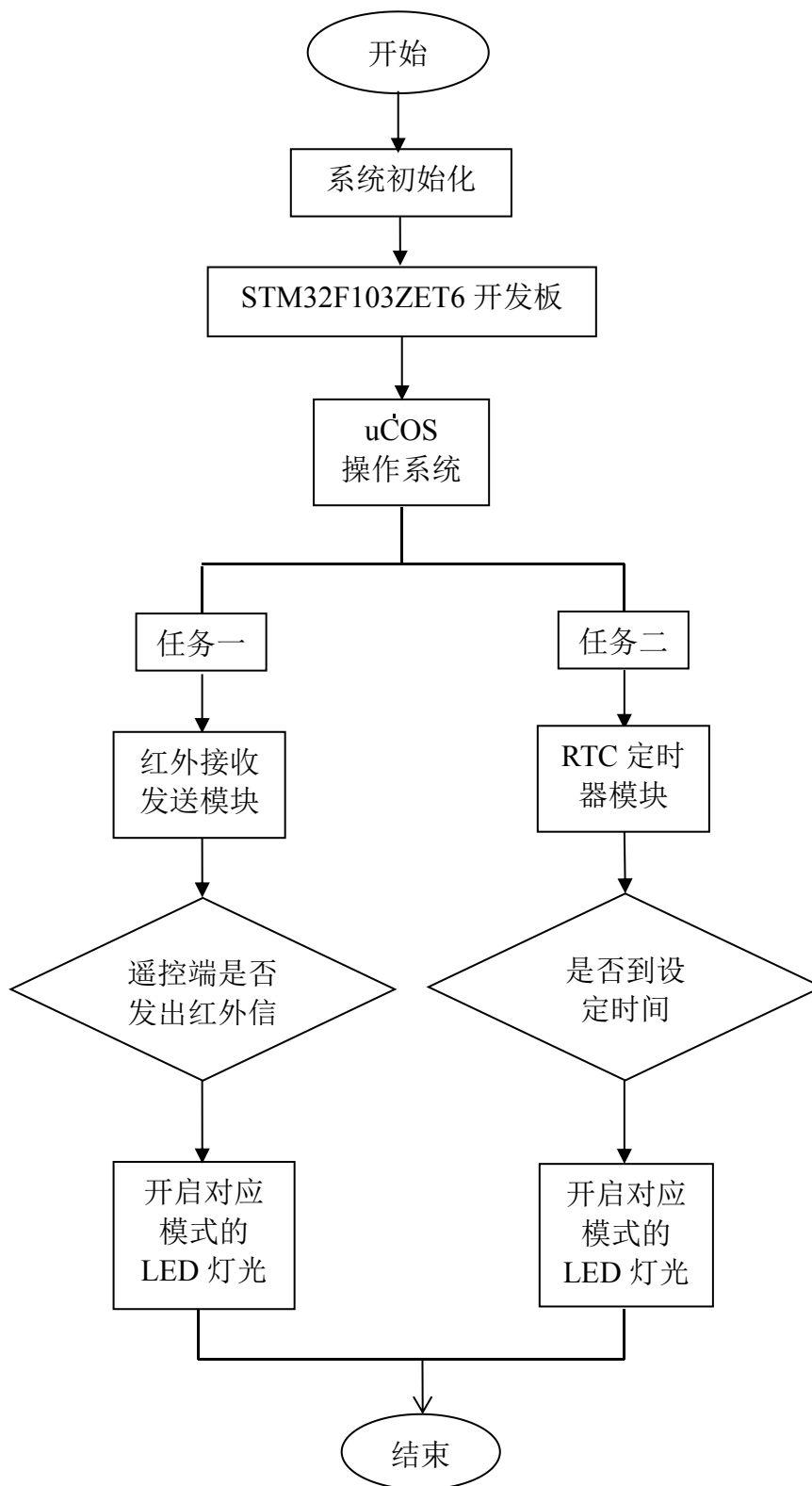


图 2.2 系统软件设计结构图

### 第三章 智能灯光控制系统所用关键性技术

在当今这个科技迅速发展的时代，设计者在对智能灯光控制系统设计之时，不仅会考虑是否符合本国国情，同时也会想到这种设计到底符不符合当前情况下的市场需求。在系统控制设计方面，有些设计侧重于放松、休闲娱乐，而有些则是侧重于环保节能，构建健康舒适的生活空间。

在进行智能灯光控制方案设计方面，设计者需综合考虑到现有科技研发水平、用户需求、国民生活水平现状，选择出一个最符合要求的智能灯光控制方案。

在此次设计中，所用系统主要采用了嵌入式技术、红外接收发送技术、外部定时器技术等几种技术，从而完成设计的功能要求。

#### 3.1 嵌入式技术

早期嵌入式处理器主要是 4 位机，现如今深受开发者与产品研发者青睐的是 32 位与 64 位机。嵌入式处理器使用范围很广主要是因为以下四个特点：1) 可完成多个实时任务，中断响应时间与其它处理器短；2) 由于其对存储区有相对强大的保护功能，这样会更便于软件判断；3) 嵌入式处理器可在较短时间内开发出能满足对应需求的应用；4) 相对功耗比较低等特点<sup>[2]</sup>。

目前通用的嵌入式处理器主要有 PowerPC、x86、ARM、MIPS、SC-400 等等。而 ARM 系列芯片以其较为强大的性能，使得很多嵌入式操作系统都可移植到 ARM 开发板中<sup>[3]</sup>。同时 ARM 嵌入式微处理器高性价比、低功耗、使用方法较为便捷的特点，使其拥有较多开发商，在市场上颇受欢迎。这样也有利于以后系统软件升级。

市面上常见使用人群较多的嵌入式操作系统主要有 uCOS、Android、WinCE、Linux 等。本次设计中选择的是最为符合的 ARM 系类芯片。还需选择适合智能灯光控制、便于操作、简单快捷的嵌入式操作系统。

#### 3.2 红外遥控技术

社会在发展时代在进步，更多方便快捷的方式，正在广泛应用于人们的生产生活中。而在这些无线遥控中，红外遥控以其独特优势深受大众喜爱，并且应用于各个领域。例如目前家庭中一些电器产品，或者是交通行业中的输设备、又或者是金融行业还有商业设施中。除此之外，由于红外遥控设备有着可靠高性，不去干扰其他电器设备正常工作，隔离电气干扰，以及不会影响到周边环境特性，广泛应用于一些高压、有辐射存在、或者是一些有毒气体存在的危险性较高的工业生产工作环境。

如今人眼可察觉到的可见光，按照波长大小依次排列为红、橙、黄、绿、青、蓝、紫。红光是在  $0.62\ \mu\text{m} \sim 0.76\ \mu\text{m}$  这个波长的范围内，红外线就是比红光波长还要长的可见光。红外遥控器是通过将波长是在  $0.76\ \mu\text{m} \sim 1.5\ \mu\text{m}$  之间的红外

线作为传递信号，传送出目的信号。目前所使用的红外设备中，普遍有着体积小、功耗低、功能性强、成本较低的特点。此外，红外遥控设备电路调试工作较为简易。如果将其发射信号进行编码，就可以实现多种红外遥控功能。由于红外遥控设备是基于无线遥控，使用时会比较方便、快捷，红外遥控设备也将会适用于越来越多领域。

### 3.3 RTC 实时定时技术

RTC 英文全称是 Real-Time Clock，中文名称是实时时钟，也可称为是时钟芯片，多以集成电路为核心。实时时钟芯片的适用范围、应用领域都是比较广的。实时时钟不仅可作为电子系统的时间基准，还可以为使用者提供较为精准的实时时间。

RTC 主要功能就是提供稳定、可靠、精确的系统时间，显示时、分、秒，年、月、日等信息。此外在系统处于关机情况下，实时时钟可由其后备电池供电从而正常工作。可是对于时间来说，无论其显示结果如何，这是有误差存在。在这个情况下 RTC 外围器件中如何去匹配较为合适的电容，就显得尤为重要。目前较为常用的 RTC 外围仅仅由一个匹配的电容、一个高精度晶振还有电阻组成。

虽然没有实时 RTC 时钟控制系统也是可以计算实际时间，但是如果要是使用了实时时钟，就有了以下优点：

- 1)会使消耗功率较低(尤其在使用辅助电源时会显得很重要)；
- 2)会优先让主系统处理更为需要时效性的工作；
- 3)某些情况下会与其他方式输出时间相比会更准确一些。

通常情况下所使用实时时钟有备用电池作为电源，在主电源无法使用或者是断电时，实时时钟通过使用这个备用的电池供电，从而继续进行计算时间。在本次毕业设计也是主要采用 RTC 来获得较为精准的时间，通过对其定时，与 LED 灯联系起来，达到每天早晨 7 点，自动将卧室灯光亮起。到了晚上 10 点，自动关闭所有室内灯光的功能。

## 第四章 嵌入式系统

嵌入式系统有很多定义，如今国内外普遍认同度比较高的说法是：嵌入式系统是以应用为中心，以计算机技术作为其根基，其软硬件可进行剪裁，并且在实际应用中对于性能、功耗、可靠性、成本预算等方面严格要求的计算机系统<sup>[4]</sup>。嵌入式系统和通用计算机相似，都是由软件和硬件相组成。本节则是重点介绍了嵌入式系统软件部分。

### 4.1 嵌入式系统软件介绍

嵌入式软件可包括应用软件和操作系统，其核心部分还是操作系统。嵌入式操作系统性能模块有：设置驱动接口、系统内核、文件系统、用程序接口等<sup>[5]</sup>。

起初由于所需功能较为简单，并受到了科学技术发展水平限制，没有在嵌入式系统中安装操作系统。但是随着科技不断发展，使得嵌入式技术的多重优点不断显露出来，并得到了飞速发展。

嵌入式系统本身是实时操作系统，其操作系统是在当前命令或任务接收之时，可立即做出响应，并及时完成的操作系统。嵌入式实时操作系统还可划分为硬实时操作系统和软实时操作系统<sup>[6]</sup>。软实时操作系统要求是该事件反应时间极快。而硬实时操作系统则不仅要求任务反应时间要快，还要求了在所规定时间内完成了对事件的处理。

嵌入式实时操作系统都是提供“微内核”的，其余一些特殊模块或辅助模块需要用户根据自己的需求自行添加。而这些嵌入式实时操作系统大多都是提供以下三个方面的功能支持。

（1）内存管理。主要是在程序使用内存时，通过利用实时操作系统内存来获得充足内存空间，主要体现在动态内存管理上，可重复使用。

（2）多任务管理。着重体现了嵌入式实时操作系统在执行多个任务时，可提供多个管理函数，开发者尤其是程序设计员就可设计多个任务，来满足更多需求。而通常情况下，嵌入式实时操作系统控制任务的启动、运行、暂停、结束等状态。同时通过这种控制使得每个任务的每个动作都在规定时间完成。

（3）外围资源管理。除了嵌入式系统微处理器还有内存外，还有很多外围设备才称得上是完整的嵌入式系统。这些外围设备可包括显示器、通信端口、键盘、外部控制器等。由于内部资源有限，还需对多种外围设备进行合理配置管理，才能达到资源优化配置。

常用的嵌入式操作系统虽然多种多样，大体可分为两类，一类是商用型操作系统，另一类则是免费型操作系统。商用型操作系统在稳定性能和可靠性能上表现更为突出，售后服务与技术支持也比较完善。而免费型操作系统无需付费，学生们使用较为广泛，可查阅使用的公开资料较多，如 Linux 和 uCOS 等。



## 4.2 uCOS II 操作系统

本次设计是基于嵌入式的智能灯光控制系统，根据执行的任务，任务数量确定了 ARM 系列嵌入式微处理器，还需选择相对适合的操作系统。在上述操作系统中，由于 uCOS II 操作系统不论是在价格方面，相对公开程度、操作性能还有其抢占优先级的特点，都是较为适合此次设计使用。

### 4.2.1 uCOS II 操作系统介绍

uCOS II 实际上是基于优先级可率先进行抢占的硬实时操作内核，同时也是较为完整的、可进行抑制、固话、裁剪的实时处理多任务内核。uCOS II 操作系统自发布以来，就受到了广泛应用，已经被移植到超过 40 种不同结构的 CPU 上。从 8 位系统开始运行，到 64 位系统上运行，表明其应用领域很广泛。uCOS II 操作系统在 2000 年时通过了关于航空项目的资格认证，这个认证是关于美国联邦航空管理局对于商用航空飞机与 RTCA-178B 准则相符的认证。再次证明该操作系统可用于更为严苛、安全性、可靠性、稳定性要求更高的系统。uCOS II 操作系统的每项功能、每个函数都经过了无数次测试，使其具有高度稳定性和可靠性。同时其开放式的源代码，以及灵活的可移植性都受到了广大开发者的追求。

### 4.2.2 uCOS II 操作系统优点

#### 1) 可公开查阅源代码

uCOS II 操作系统源代码晰易懂，并且其结构协调，uCOS II 操作系统源代码漂亮、和谐，解释较为仔细，组织合理。

#### 2) 具有可移植性

uCOS II 操作系统中与微处理器相关程序都是由汇编语言编写而完成，其汇编部分较低，使得 uCOS II 操作系统更加方便移入到不同微型处理器当中。

#### 3) 可固化性强

uCOS II 操作系统可应用于嵌入式应用中，只需有合适的软件工具，就可实现 uCOS II 操作系统嵌入到产品中，作为新的产品出现。

#### 4) 可裁剪性高

只需采用一些 uCOS II 应用程序中使用的系统服务，就可减少产品中 uCOS II 操作系统存储空间，其可裁剪性只需在应用程序中定义相应功能即可。

#### 5) 多任务

uCOS II 操作系统管理任务有 64 个，由于任务不同使得优先级也不相同，一般情况下 uCOS II 操作系统并不能使用时间片轮转调度法，因为这种调度方法主要是应用在调度一些优先级相同的任务。

#### 6) 可确定性好

uCOS II 操作系统在函数调用或者服务执行时，其时间可确定，也就表明了，

除某些特殊服务或函数外，uCOS II 操作系统执行服务时间并不取决于其任务的多少。

#### 7) 任务栈

uCOS II 操作系统是允许其每个任务都是有独立的栈空间，这样就可适当降低所使用程序对于 RAM 的需求。

#### 8) 稳定性和可靠性

uCOS II 操作系统在 2000 年时通过了关于航空项目资格认证，这个认证是关于美国联邦航空管理局对于商用航空飞机与 RTCA DO-178B 准则相符的认证。这个标准主要是应用在航空设备操作的软件领域。由于对于软件的稳定性和安全性都有一定，所以对操作系统选择方面也有要求。这一结果表明，这个操作系统在质量方面的认可程度，也表明了该操作系统在其他场合中也展示出良好的性能。为了使其该操作系统可用于更为严苛、安全性、可靠性、稳定性要求更高的系统，uCOS II 操作系统经过了无数次测试，再次证明了其操作系统有足够的安全性和稳定性。

## 第五章 智能灯光控制系统硬件设计

### 5.1 STM32F103ZET6 设计

选择 STM32 系列芯片主要是其高性能、低功耗、低成本的特点。通常来讲 STM32 芯片可分为基本型 STM32F103 系列芯片及增强型 STM32F101 系列芯片。而 STM32F103 系列芯片还可根据容量进行分类：HD 系列是大于 256K，MD 系列是小于 256K，而 LD 系列则是小于 64K。根据本次设计中要求，选择了增强型的 LD 系列的 STM32F103 芯片。

本次设计中选择的是最小系统核心板开发板。该开发板是基于 ARM Cortex-M3 为核心的 32 位微处理器，具有 512K 片内 FLASH 相当于硬盘，除此之外还有 64K 片内 RAM 相当于内存。在下图所示的图中，可看到 BOOT 区，该区为了方便于串口下载程序。设计者先将事先编辑好的程序调试好，通过 BOOT 区串口下载到芯片中，使其到达期望的功能。同时也要注意 CPU 操作电压要控制范围在 2.0V-3.6V 之间。此外，所选的开发板还有 80 多个 IO 口，大多都是兼容 5V 电源。使用前需仔细检测每一个 IO 口引脚，确保其不被占用，或者这个 IO 口是否满足所设功能的要求。下图 5.1 为 STM32F103ZET6 主版图。

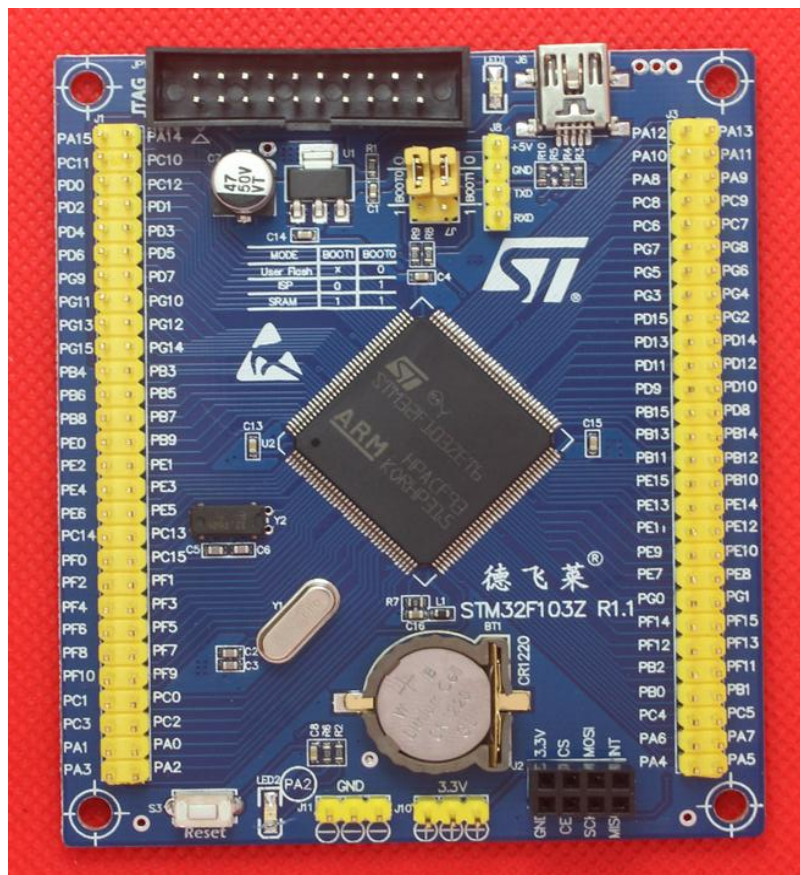


图 5.1 STM32F103ZET6 的主版图



### 5.1.3 STM32F103ZET6 的 I/O 口与其外围电路接口的分配情况

表 5.1 I/O 口分配情况

| STM32 的 I/O 引脚 | 外围电路引脚    | 说明           |
|----------------|-----------|--------------|
| PE9            | LED1      | 进门灯          |
| PE10           | LED2      | 客厅灯          |
| PE11           | LED3      | 餐厅灯          |
| PE12           | LED4      | 厨房灯          |
| PE13           | LED5      | 主卧灯          |
| PE14           | LED6      | 次卧灯          |
| PE15           | LED7      | 洗手间灯         |
| PD0~PD7        | D0~D7     | LCD 数据口      |
| PB0            | E         | LCD 使能端      |
| PB1            | RW        | LCD 读写选择端    |
| PB2            | RS        | LCD 数据/命令选择端 |
| PE0            | DO        | 红外接收端口       |
| TXD            | 串口服务器 RXD | 串口转以太网模块     |
| RXD            | 串口服务器 TXD | 串口转以太网模块     |

## 5.2 红外遥控设备

在本次的设计中所采用的是一款新型超薄 38K 通用型红外遥控模块，采用 NEC 编码格式，可用于灯光设备的配置、单片机开发板与学习板等应用场合。

### 5.2.1 红外遥控设备的使用注意事项

在使用前需给红外遥控器安装电池，要将遥控器与电池之间的隔离薄片取出。同时在此次设计中还要将红外遥控器与红外接收模块结合起来使用。使用前要解码红外遥控器的发射信号，然后才能实现相应操作。

在此之前还应注意查看红外遥控的技术参数，根据查阅资料了解到，红外遥

控距离应在 8 米之内，遥控时要注意到红外遥控探头要对准红外接收装置探头。连接电路时需注意红外接收器 D0 端应该接 STM32F103ZET6PB9 接口，VCC 引脚应该接到开发板上+5V 电源，而 GND 应该接到开发板上接地引脚。常用的红外遥控器如图 5.3 所示, 红外接收器如图 5.4 所示。



图 5.3 红外接收器

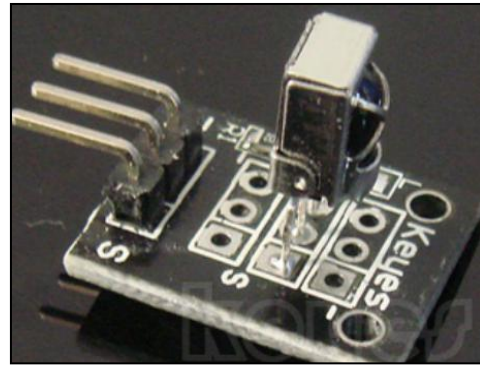


图 5.4 红外接收器

### 5.2.2 红外遥控按键的解码

在使用红外遥控设备之前，必须要对红外遥控设备每一个按键进行解码。通过将红外遥控器按键发射来的信号解码成 10 进制编码，才可在进行红外功能程序编码时实现自定义功能按键，完成所需任务。在解码过程中需要注意红外遥控器和红外接收头位置，确认红外接收头这一端是否接收到红外遥控器端发出的信号。并且通过红外解码不难发现，每一个按键都是有其唯一的十进制编码。以下是测试所得的红外遥控器按键解码。

表 5.2 红外遥控键码

| 按键  | 代码  | 按键   | 代码  | 按键   | 代码  |
|-----|-----|------|-----|------|-----|
| CH- | 162 | CH   | 98  | CH+  | 226 |
| ◀◀  | 34  | ▶▶   | 2   | ▶    | 194 |
| —   | 224 | +    | 168 | EQ   | 144 |
| 0   | 104 | 100+ | 152 | 200+ | 176 |
| 1   | 48  | 2    | 24  | 3    | 122 |
| 4   | 16  | 5    | 56  | 6    | 90  |
| 7   | 66  | 8    | 74  | 9    | 83  |



### 5.3 LCD 液晶显示屏

在本次设计中，主要通过使用 LCD1602 液晶显示屏来表示出实时时间，如年月日、时分秒，还可以表示周期数，以方便用户根据实时时间的对灯光进行有效的控制。此次毕设中所采用的液晶显示屏是一个 32 字符型液晶显示屏。LCD 液晶显示屏实物图与引脚图如下所示。

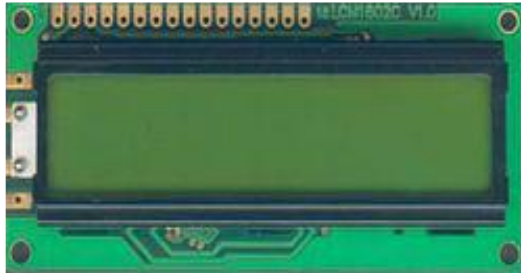


图 5.5 LCD 液晶显示屏

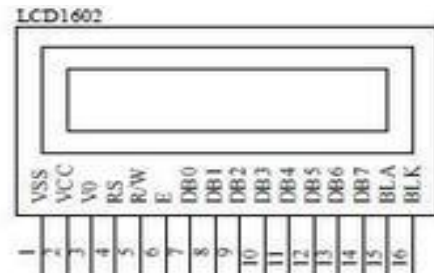


图 5.6 LCD 液晶显示屏引脚图

连接时应注意 LCD 数据口 D0~D7 对应的是 STM32F103ZET6 的 PD 端口，依次从 PD0 接入到 PD7。如图 LCD 引脚 VSS 和引脚 VCC 是需要接+5V 电源正极的。。在实际当中 STM32 核心板电源引脚较少，需将其+5V 电源极引出单独给 VCC、VSS 供电。引脚 V0 可调节显示屏对比度大小，连接 STM32F103ZET6 核心开发版 PB2 端口。引脚 RS 是一个选择寄存器端口，相对应的是 STM32F103ZET6 开发版 PB1 端口。选择数据寄存器时则需要置高电平，选择指令寄存器时则需要置低电平。引脚 RW 是读写信号，连接在 STM32F103ZET6 核心开发版 PB0 端口。选择写信号操作时需要置高电平，进行读操作时需要置低电平。引脚 E 是使能端。引脚 15、16 是 LCD 背灯点亮电源。15 引脚 BLA 是背灯电源的正极，接入电源正极。16 引脚 BLK 是背灯电源的负极，接入外接电源接地引脚。

## 第六章 智能灯光控制系统软件设计

本次智能灯光控制系统设计中软件设计主要有红外接收和发送模块设计、实时时钟模块设计、LCD 显示模块设计、远程控制模块设计此外还需对多个场景的模式进行设计。在 LCD 液晶显示屏上显示出实时时间，再设定指定时间。到指定时间后对应灯光亮起或熄灭，从而更方便的服务于使用者。

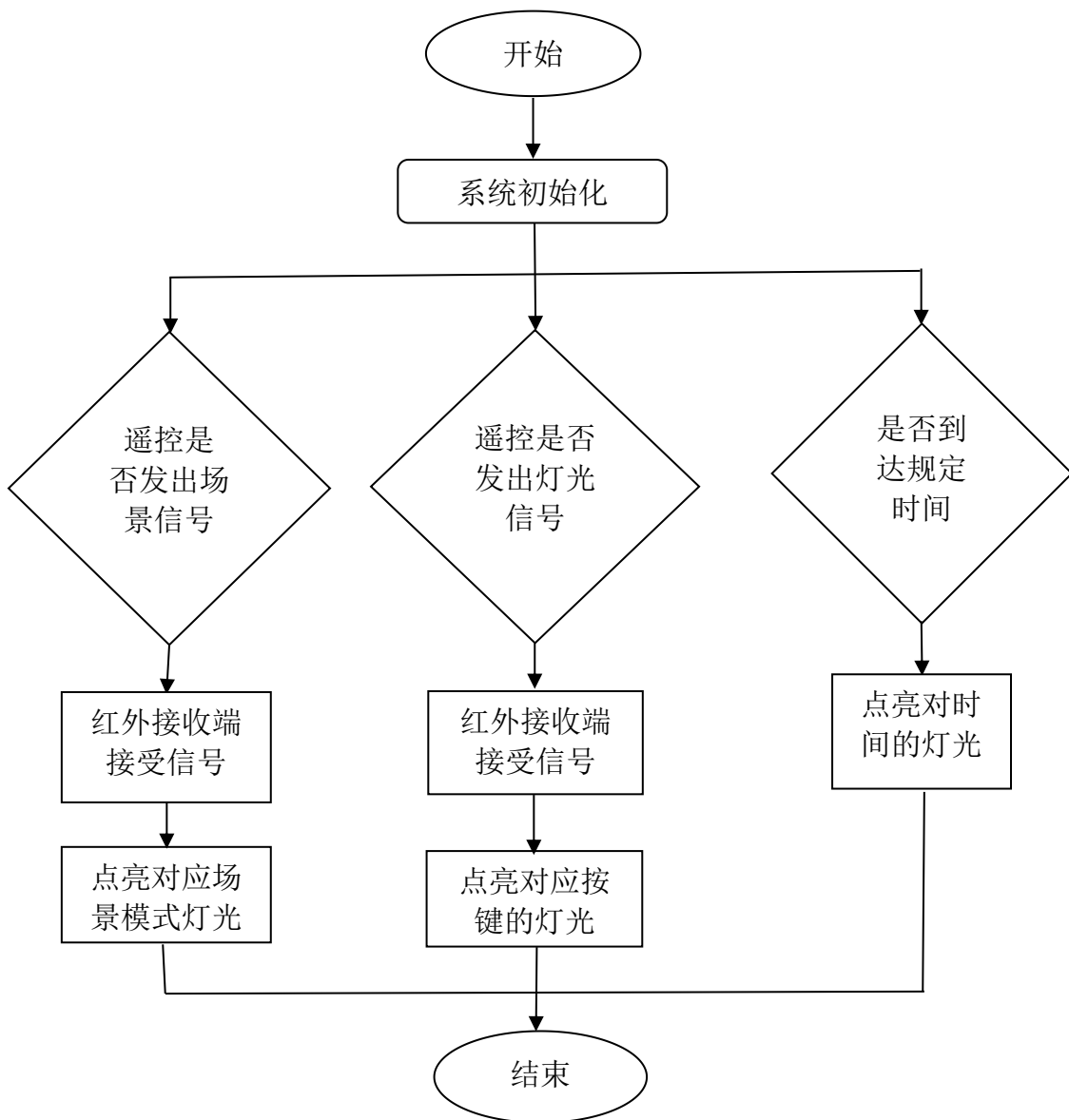


图 6.1 系统主设计流程图

### 6.1 红外系统设计

在红外方面，可事先根据房屋大小、布局，设定好多路灯光和多种场景模式。通过红外遥控端发送信号，控制房屋内部灯光开闭。



### 6.1.1 场景模块设计

用户可以根据自己的生活需求对场景模块进行自定义场景设计,对有限的室内空间进行合理的规划。这样用户可对一些固定模式场景进行编程,使用者可在一定空间范围之内按下一个按键,进而控制对应的一组灯具。并且可以根据用户实际需要不同的按键进行切换,达到不同灯光场景的改变<sup>[7]</sup>。而这些场景也仅仅是将不同房间的灯光进行多样化组合,例如回家模式、离家模式、会客模式、就餐模式、party 模式、夜晚模式 1 以及夜晚模式 2。下面就是对应场景模式灯光组合:

1) 回家模式:回家模式是在用户刚一进家门时,进门灯、餐厅灯、厨房灯和客厅灯,均可达到一键同时打开的效果。让使用者有种温馨明亮安全的家庭氛围,这也避免了在刚一进家门时,由于屋内光线较黯、视线较差而引发的摔倒、碰撞等安全事故。用户可直接对红外遥控器进行操作,按下回家模式对应按键,即可完成有关场景打开。

2) 离家模式:离家模式是用户在离开家门之前,按下某个按键就可关闭室内所有灯具。免去逐个关闭家用灯具的麻烦,既节约了用电,也避免了在离开家门后家用灯具由于长时间打开温度过高,损耗家用灯具寿命或造成一些不必要的安全隐患。用户可以直接使用红外遥控端,按下离家模式红外遥控按键,室内所有灯光就可同时关闭。

3) 会客模式:会客模式是使用户在家中会客时,有良好的会客氛围,可实现一键打开客厅和餐厅灯具。为了使会客氛围更加轻松,可以同时打开餐厅灯光,便于主人招待访客。用户可以按下红外遥控端会客模式按键,会客模式场景就可打开。

4) 就餐模式:就餐模式就是为了更方便于用户就餐之前,在没进入到厨房时,就可以达到一键打开厨房和餐厅的灯。此时,用户就可以更为方便的出入厨房和餐厅,也避免了在一些情况下,进出厨房会发生的突发情况。用户可以按下红外遥控端就餐模式按键,就餐模式场景就可打开。

5) Party 模式:party 模式就是为了在用户想要在家中举行一些小型朋友聚餐、生日宴会或者是一些娱乐活动时而应景产生的,可以使用户在刚一进回家后,可以一键打开进门灯、客厅灯、餐厅灯、厨房灯、主卧灯、次卧灯和洗手间灯。再将所有的灯打开后,立马就有了一种热闹的场景,使人们很快就融入到当前的氛围当中。用户操作起来也更为便捷,不用逐一去每个房间,打开每一盏灯。而将每一盏灯都打开,也会使客人更为方便。用户可以一键按下红外遥控端 party 模式,完成对所有灯光的一键打开。

6) 夜晚模式 1:夜起模式 1 主要针对于主卧房间。可以在夜晚时,对于有需要夜起的人,做到一键打开主卧洗手间的灯。可以使在夜晚临时需要起床去洗

手间的用户，不用在黑暗的情况下起床打开主卧房间的灯和卫生间的灯。这样可以减小由于室内光线较差，一些不安全因素的发生。尤其是对于一些老人和在漆黑中视力不太好的夜盲症患者，这样的模式会更为安全。用户选择按下红外遥控端的夜晚模式 1，完成主卧夜晚灯光打开。

7) 夜晚模式 2：夜晚模式 2 主要是针对于次卧的房间。模式的主要功能和作用跟夜晚模式 1 类似，主要也是用在屋内灯光较黯的情况下。而唯一的不同点就是，夜晚模式 2 可以一键打开次卧的灯和洗手间的灯。在本次设计中，设计者将次卧的房间设计成了儿童房，这样就可以使小朋友在夜晚或者光线不好时，减少发生不必要的危险。使家长们也更为放心。用户可以按下红外遥控端的夜晚 2 模式按键，完成对于夜晚次卧灯具的控制。

特别注意：在用户在执行现有场景模式的选择和切换的同时，也可以根据用户自身目前的实际需要来选择打开或者是关闭某一盏的灯，或者是某几盏的灯，这样也可以现有的几种固定的常用模式下随机的组成几种新的场景模式<sup>[8]</sup>。譬如：用户已经打开了回家模式的按键，现在可以根据用户当前回家后想要进洗手间的需要，按下洗手间灯的按键，这两者之间却是互不影响的。或者用户在打某个场景之后，过了一段时间，发现某几盏灯已经不需要使用时，也是可以对其进行关闭的。

下面是现有模式对应的红外遥控按键和灯光的配置：打开灯具√ 关闭灯具×

表 6.1 场景模式按键设置

| 场景模式        | 红外<br>按键 | 进门<br>灯 | 客厅<br>灯 | 餐厅<br>灯 | 厨房<br>灯 | 主卧<br>灯 | 次卧<br>灯 | 洗手间<br>灯光 |
|-------------|----------|---------|---------|---------|---------|---------|---------|-----------|
| 回家模式        | CH+      | √       | √       | √       | √       |         |         |           |
| 离家模式        | ▶II      | ×       | ×       | ×       | ×       | ×       | ×       | ×         |
| 会客模式        | EQ       |         | √       | √       |         |         |         |           |
| 就餐模式        | 200+     |         |         | √       | √       |         |         |           |
| Party<br>模式 | 3        | √       | √       | √       | √       | √       | √       | √         |
| 夜晚模<br>式 1  | 6        |         |         |         |         | √       |         | √         |
| 夜晚模<br>式 2  | 9        |         |         |         |         |         | √       | √         |

下面则是对场景模式主要程序的设计。

```

void GoHomeMode(void) // 回家模式
{
    ENTER_LED = 0;
    GUEST_LED = 0;
    RESTAURANT_LED = 0;
    KITCHEN_LED = 0;
    MASTER_ROOM_LED = 1;
    SECOND_ROOM_LED = 1;
    TOILET_LED = 1;
}

void LeaveHomeMode(void) // 离家模式
{
    AllLigthsOff();
}

void GuestMode(void) // 会客模式
{
    GUEST_LED = 0;
    RESTAURANT_LED = 0;
    KITCHEN_LED = 1;
    MASTER_ROOM_LED = 1;
    SECOND_ROOM_LED = 1;
    TOILET_LED = 1;
}

void RepastMode(void) // 就餐模式
{
    ENTER_LED = 1;
    GUEST_LED = 1;
    RESTAURANT_LED = 0;
    KITCHEN_LED = 0;
    MASTER_ROOM_LED = 1;
    SECOND_ROOM_LED = 1;
    TOILET_LED = 1;
}

// 模式选择, 1~7对应7种模式
// mode: 为模式选择
void SelectMode(u8 mode)
{
    switch(mode)
    {
        case 1 : GoHomeMode(); break;
        case 2 : LeaveHomeMode(); break;
        case 3 : GuestMode(); break;
        case 4 : RepastMode(); break;
        case 5 : PartyMode(); break;
        case 6 : Night_1Mode(); break;
        case 7 : Night_2Mode(); break;
        default : break;
    }
}

```

图 6.2 场景模块程序设计

### 6.1.2 灯光模块设计

灯光不仅可以通过场景模块的设计, 实现多个灯具的打开和关闭。还可以根据用户的实际需要, 对每个灯具都可以通过红外遥控按键进行打开和关闭的操作。这样就可以不局限于现有的模式<sup>[9]</sup>。例如: 用户正在客厅看书, 就仅仅需要打开客厅的灯具, 此时只需客厅的灯具关闭其他房间的灯具。

下面是 LED 灯光的程序设计。

```
#include "led.h"

//LED IO初始化
void LED_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOE, ENABLE); //使能PE端口时钟

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9 | GPIO_Pin_10 | GPIO_Pin_11 | GPIO_Pin_12 | GPIO_Pin_13;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; //推挽输出
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz; //IO口速度为50MHz
    GPIO_Init(GPIOE, &GPIO_InitStructure); //根据设定参数初始化GPIOB0
    //status: 灯状态, 为0表示灭, 为1表示亮
    void SelectLedStatus(u8 led_num, u8 status)
    {
        switch(led_num)
        {
            case 1 :
            {
                if(status == 1)
                {
                    ENTER_LED = 0;
                }
                if(status == 0)
                {
                    ENTER_LED = 1;
                }
            }

            case 2 :
            {
                if(status == 1)
                {
                    GUEST_LED = 0;
                }
                if(status == 0)
                {
                    GUEST_LED = 1;
                }
            }
            break;

            case 3 :
            {
                if(status == 1)
                {
                    RESTAURANT_LED = 0;
                }
            }
            case 8 :
            {
                if(status == 1)
                {
                    AllLigthsOn();
                }
                if(status == 0)
                {
                    AllLigthsOff();
                }
            }
            break;

            default : break;
        }
    }
}
```

图 6.3 灯光模块程序设计

在本次设计中,设计者选择了一个小户型的两室一厅房间,则室内灯光在遥控设备上的分配如下表所示。

表 6.2 灯光按键设置

| 灯具名称 | 进门灯 | 客厅灯 | 餐厅灯 | 厨房灯  | 主卧灯 | 次卧灯 | 洗手间灯 |
|------|-----|-----|-----|------|-----|-----|------|
| 开灯按键 | CH- | ◀◀  | —   | 0    | 1   | 4   | 7    |
| 关灯按键 | CH  | ▶▶  | +   | 100+ | 2   | 5   | 8    |

### 6.1.3 红外遥控设计

首先需要给红外遥控器供电，然后将红外遥控器与红外接收器模块结合起来。通过将红外遥控器方面发射过来的按键得信号解码成是 10 进制码。如果要实现按下一个按键，可以实现相对应的功能，那么就得事先编辑好按键所对应的功能程序。

下面则是红外遥控设计中的程序设计。

```
void Remote_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;
    NVIC_InitTypeDef NVIC_InitStructure;
    TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;
    TIM_ICInitTypeDef TIM_ICInitStructure;

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB,ENABLE); //使能PORTB时钟
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4,ENABLE); //TIM4 时钟使能

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9;           //PB9 输入
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPD;       //上拉输入
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOB, &GPIO_InitStructure);
}

void TIM4_IRQHandler(void)
{
    #if SYSTEM_SUPPORT_OS
        OSIntEnter();
    #endif
    if(TIM_GetITStatus(TIM4,TIM_IT_Update) !=RESET)
    {
        if(RmtSta&0x80) //上次有数据被接收到了
        {
            RmtSta&=~0x10; //取消上升沿已经被捕获标记
            if((RmtSta&0X0F)==0X00) RmtSta|=1<<6; //标记已经完成一次按键的键值信息采集
            if((RmtSta&0X0F)<14) RmtSta++;
        }
        else
        {
            RmtSta&=~(1<<7); //清空引导标识
            RmtSta&=0XF0; //清空计数器
        }
    }
}
```

```

if(TIM_GetITStatus(TIM4,TIM_IT_CC4)!=RESET)
{
    if(RDATA)//上升沿捕获
    {
        TIM_OC4PolarityConfig(TIM4,TIM_ICPolarity_Falling);           //CC4P=1 设置为下降沿捕
        TIM_SetCounter(TIM4,0);           //清空定时器值
        RmtSta|=0X10;           //标记上升沿已经被捕获
    }else //下降沿捕获
    {
        Dval=TIM_GetCapture4(TIM4);           //读取CCR4也可以清CC4IF标志位
        TIM_OC4PolarityConfig(TIM4,TIM_ICPolarity_Rising);           //CC4P=0 设置为上升沿捕获
        if(RmtSta&0X10)           //完成一次高电平捕获
        {
            if(RmtSta&0X80)//接收到了引导码
            {
                if(Dval>300&&Dval<800)           //560为标准值,560us
                {
                    RmtRec<<=1;           //左移一位.
                    RmtRec|=1;           //接收到1
                }else if(Dval>2200&&Dval<2600) //得到按键键值增加的信息 2500为标准值2.5ms
                {
                    RmtCnt++;           //按键次数增加1次
                    RmtSta&=0XF0;           //清空计数器
                }
            }else if(Dval>4200&&Dval<4700) //4500为标准值4.5ms
            {
                RmtSta|=1<<7;           //标记成功接收到了引导码
                RmtCnt=0;           //清除按键次数计数器
            }
        }
    }
}
TIM_ClearITPendingBit(TIM4,TIM_IT_Update|TIM_IT_CC4);
#ifdef SYSTEM_SUPPORT_OS
    OSIntExit();
#endif

```

图 6.4 红外收发程序编写

## 6.2 实时时钟设计

实时时钟的设计是为了使用户达到更为合理的规划室内灯光的效果,通过对时间的合理安排,从而对室内的灯光进行更为舒心的控制。在此基础上首先要知道实时时间。而实时时钟的目的有些类似于家中的钟表,提醒用户执行某些动作的时间到了。设计者在设计任务或者是场景时,首先是要设置当下的时间通过完成对时间的设定才能够完成有关的定时任务。例如:早晨的叫醒服务,还有晚上的方便式自动关灯功能。

而在时钟设计的程序设计当中,首先要找到之前建立的工程,并新建一个 RTC 的文件夹,将其放入到 HARDWARE 文件夹里面。然后去打

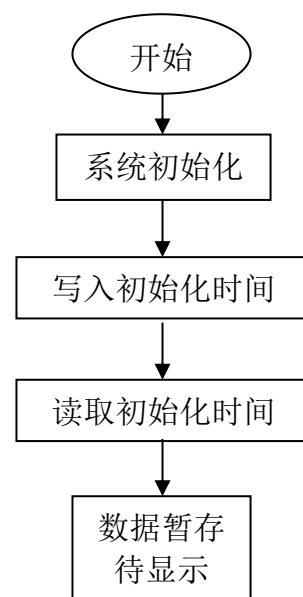


图 6.5 钟模块设计



开 USER 文件夹内的工程，在其内部新建一个名为 rtc.c 的文件和名为 rtc.h 的头文件，将其保存在 RTC 的文件夹下，并将 RTC 的文件夹加入到头文件包含得路径中。首先是对其进行初始化定义，并且设置优先级的先后顺序。然后设置实时时间，并进行参数的配置。特别注意的是在配置 RTC 实时时钟时对寄存器的使用。以下便是实时时钟的程序编辑。

```
static void RTC_NVIC_Config(void)
{
    NVIC_InitTypeDef NVIC_InitStructure;
    NVIC_InitStructure.NVIC_IRQChannel = RTC_IRQn;    //RTC全局中断
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0; //先占优先级1位,从优先级3位
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0; //先占优先级0位,从优先级4位
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;    //使能该通道中断
    NVIC_Init(&NVIC_InitStructure);    //根据NVIC_InitStructure中指定的参数初始化外设NVIC寄存器
}

u8 RTC_Init(void)
{
    //检查是不是第一次配置时钟
    u8 temp=0;
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_PWR | RCC_APB1Periph_BKP, ENABLE); //使能PWR和BKP外设
    PWR_BackupAccessCmd(ENABLE); //使能后备寄存器访问
    if (BKP_ReadBackupRegister(BKP_DR1) != 0x5050)    //从指定的后备寄存器中读出数据:读出了与写入
    {
        BKP_DeInit(); //复位备份区域
        RCC_LSEConfig(RCC_LSE_ON); //设置外部低速晶振(LSE),使用外设低速晶振
        while (RCC_GetFlagStatus(RCC_FLAG_LSERDY) == RESET&&temp<250) //检查指定的RCC标志位设置与否
        {
            temp++;
            RTC_WaitForSynchro(); //等待RTC寄存器同步
            RTC_ITConfig(RTC_IT_SEC, ENABLE); //使能RTC秒中断
            RTC_WaitForLastTask(); //等待最近一次对RTC寄存器的写操作完成
            RTC_EnterConfigMode(); //允许配置
            RTC_SetPrescaler(32767); //设置RTC预分频的值
            RTC_WaitForLastTask(); //等待最近一次对RTC寄存器的写操作完成
            RTC_Set(2017,4,22,18,43,00); //设置时间
            RTC_ExitConfigMode(); //退出配置模式
            BKP_WriteBackupRegister(BKP_DR1, 0x5050); //向指定的后备寄存器中写入用户程序数据
        }
        else//系统继续计时
        {
            RTC_WaitForSynchro(); //等待最近一次对RTC寄存器的写操作完成
            RTC_ITConfig(RTC_IT_SEC, ENABLE); //使能RTC秒中断
            RTC_WaitForLastTask(); //等待最近一次对RTC寄存器的写操作完成
        }
    }
    //RTC时钟中断
    //每秒触发一次
    //extern ul6 tcnt;
}

void RTC_IRQHandler(void)
{
    #if SYSTEM_SUPPORT_OS
    OS_CPU_SR cpu_sr;
    OS_ENTER_CRITICAL();
    #endif
    if (RTC_GetITStatus(RTC_IT_SEC) != RESET)//秒钟中断
    {
        RTC_Get(); //更新时间
    }
}
```

图 6.6 RTC 实时时钟程序

### 6.3 LCD 模块设计

本模块的设计主要是为了显示当前设置的时间，可以通过程序来修改系统当前的时间，并且显示出当前的年月日、时分秒，还可以显示出当前这一天在本月的周期数。从而使用户可以更合理的进行控制。在系统设定时间时，如果出现设定错误，超出正常时间值，如 25:49 等，LCD 显示屏还会出现 ERR。如若时间设定正确，则显示 OK。

首先需要对其进行初始化定义，以及端口的配置，并进行推挽输出。将其每一个引脚与 STM32F103 核心开发板的 I/O 端口进行配置。编写到 LCD1602 液晶显示屏程序当中去。同时需要根据任务要求进行相应程序编写。下面将是 LCD 模块的主要程序。

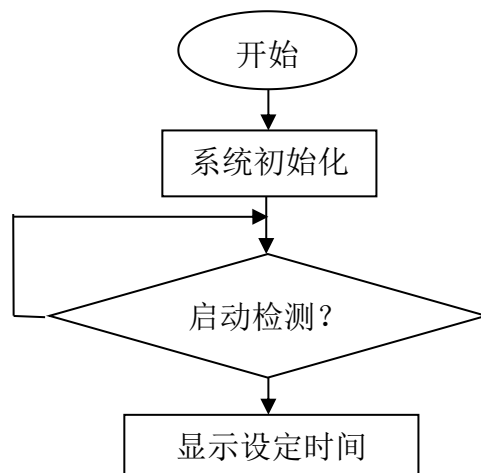


图 6.7 LCD 模块设计

```

#include "lcd1602.h"
#include "delay.h"
#include <string.h>

void LCD1602_GPIO_Configuration(void)
{
    GPIO_InitTypeDef  GPIO_InitStructure;

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB | RCC_APB2Periph_GPIOD, ENABLE); //使能PD端口时钟

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_2 | GPIO_Pin_3 | GPIO_Pin_4;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; //推挽输出
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz; //IO口速度为50MHz
    GPIO_Init(GPIOD, &GPIO_InitStructure); //初始化GPIOD0~7数据口

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2 | GPIO_Pin_1 | GPIO_Pin_0;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; //推挽输出
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz; //IO口速度为50MHz
    #if SYSTEM_SUPPORT_OS
    while(1)
    {
    #endif
        if(t != calendar.sec)
        {
            t = calendar.sec;
            LCD1602_Show_Num(0, 0, calendar.w_year);
            LCD1602_Show_Num(5, 0, calendar.w_month);
            LCD1602_Show_Num(8, 0, calendar.w_date);
            switch(calendar.week)
            {
                case 0:
                    LCD1602_Show_Str(15, 0, "7");
                    break;
                case 1:
                    LCD1602_Show_Str(15, 0, "1");
                    break;
            }
        }
    #endif
}
  
```



```

#ifdef SYSTEM_SUPPORT_OS
    while(1)
    {
#endif
        if(t != calendar.sec)
        {
            t = calendar.sec;
            LCD1602_Show_Num(0, 0, calendar.w_year);
            LCD1602_Show_Num(5, 0, calendar.w_month);
            LCD1602_Show_Num(8, 0, calendar.w_date);
            switch(calendar.week)
            {
                case 0:
                    LCD1602_Show_Str(15, 0, "7");
                    break;
                case 3:
                    LCD1602_Show_Str(15, 0, "3");
                    break;
                case 4:
                    LCD1602_Show_Str(15, 0, "4");
                    break;
                case 5:
                    LCD1602_Show_Str(15, 0, "5");
                    break;
            }
            if((calendar.hour == morning_hour) && (calendar.min == morning_min) && (calendar.sec == 0))
            {
                ENTER_LED = 1;
                GUEST_LED = 1;
                RESTAURANT_LED = 1;
                KITCHEN_LED = 0;
                MASTER_ROOM_LED = 0;
                SECOND_ROOM_LED = 0;
                TOILET_LED = 0;
            }
            if((calendar.hour == night_hour) && (calendar.min == night_min) && (calendar.sec == 0))
            {
                AllLightsOff();
            }
        }
    }
}

```

图 6.8 LCD1602 液晶显示屏的程序

## 6.4 定时模块场景的设计

本设计的预设任务中不仅有对灯光配置的场景设计，还有对于在特定的时间打开特定的灯的时间场景的设计。例如：在清晨的 7 点钟时，可以通过设置同时打开厨房的灯、主卧的灯、次卧的灯还有洗手间的灯。

这种场景在设计上，不仅起到了清晨的叫醒功能，到点后灯光可以自动打开，叫醒正在熟睡中的人们，还可以同时打开厨房的灯还有洗手间的灯。使用户不用在室内光线较差的情况进行活动，也可以更为方便的进行起床后的各项工作。在夜晚的 10 点钟时，可以自动的关闭室内的所有灯具，方便用户更好的入睡，可以不用逐一的

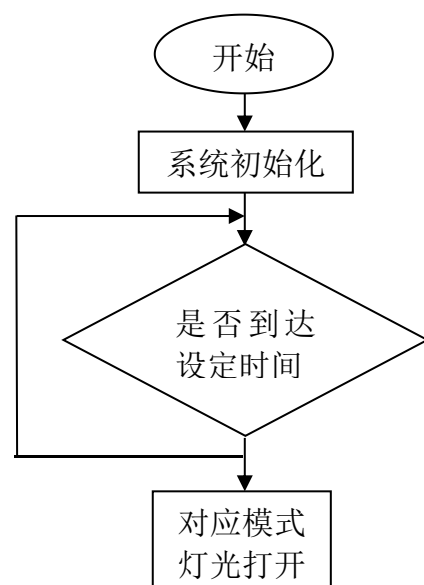


图 6.9 定时模块设计

去关闭已经打开的灯光。图 6.10 就是定时模块的主要程序编写。

```
//定时函数
extern u8 morning_hour, morning_min, night_hour, night_min;
void timing(u8 mornin_H, u8 morning_M, u8 night_H, u8 night_M)
{
    printf("正在设置定时时间, 请稍等.....");
    LCD1602_Show_Str(11, 1, " ");
    delay_ms(1000);
    if((mornin_H > 23) | (morning_M > 59) | (night_H > 23) | (night_M > 59))
    {
        printf("设置定时时间参数错误, 请检查!");
        LCD1602_Show_Str(11, 1, "T>Err");
    }
    else
    {
        morning_hour = mornin_H;
        morning_min = morning_M;
        night hour = night H;
        printf("设置定时时间完成!");
        LCD1602_Show_Str(11, 1, "T>Ok ");
    }
}
```

图 6.10 定时模块主程序

## 6.5 远程控制模块的设计

本次毕业设计不仅能够通过红外模块进行对灯光的操作, 还可通过网络端口, 对灯光、时钟的定时, 实时时钟的设定进行控制。实现远程的控制, 需要对网络模块进行设定, 需要设置串口转网络的调试助手 TCP-232。并需要提前对上位机的相关部分进行程序的编写。通过上位机操作按键控制室内灯光, 然后达到预设功能<sup>[10]</sup>。

### 6.5.1 网络模块的设计

本次设计中所使用的是一款集多个功能于一体的以太网串口数据转换的模块, 其内部由 TCP/IP 协议所集成的。用户可以通过对其的使用, 来对此次的嵌入式的设备进行网络模块的设计。而这个串口数据转换模块则需要通过 TCP-232 软件进行相应的配置。

下图 6.11 便是 TCP-232 的参数设计软件。TCP-232 是一个可以调试网络的设备, 使用前需要对它的参数进行设定。使用者可以将其使用的设备通过串口服务器 TCP-232 与以太网连接起来。从而实现了所使用的设备的网络管理与控制的功能。打开 TCP-232 串口服务器后, 首先需要将网口通过 TTL 连接到电脑端, 在进行搜索模块的操作, 选择搜索到的模块。然后按下读取参数的按键, 将所选的模块内所有的参数均读取出来, 并显示在右边框内, 如下图 6.12 所示。其次是需要根据用户的实际情况对模块 IP 和网关 IP 进行相应的修改。最后是要将修改好的参数进行一键写入的操作, 这样便完成了对网络模块的设定。

### 6.5.2 远程模块设计

除了以上控制方法，设计者还另外申请了腾讯的云服务器。这样就可以不仅仅局限于室内的局域网，甚至可以在很远的地方，只需要登陆上设计者的云主机平台，就可打开云主机中的上位机从而进行操作。需要注意的是需要提前在云主机中添加上位机模块。在这种情况下，用户即使是在几千米外甚至更远的地方，只需要用户的主机可以上网，室内的设备端也是与网线相连的，就可以对室内的灯光进行自由控制。

远程服务器 IP: 139.199.78.41

## 第七章 智能灯光控制系统测试

### 7.1 红外模块测试

在红外方面根据房屋大小、布局，设定好多路灯光和多种场景模式。通过红外遥控端发送信号，控制房屋内部灯光开闭。下面是对红外模块上的按键做的一些测试。

#### 7.1.1 测试目的

用户通过 USB 端口给 STM32F103ZET6 核心板提供 5V 电源，这样红外接收端灯光就会亮起。用户可将红外遥控器对准红外接收端，按下红外发送端遥控器上的灯光按键，相对应的灯光就会亮起，达到一键按下时一盏或几盏灯亮起。

#### 7.1.2 测试结果

下图是通过红外遥控对灯光测试的结果。

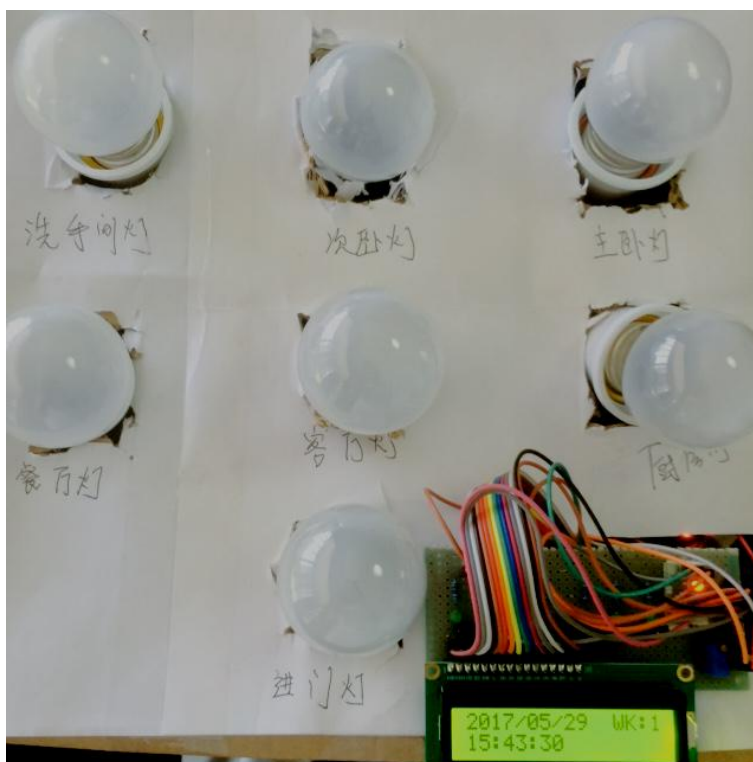


图 7.1 测试前灯光显示结果

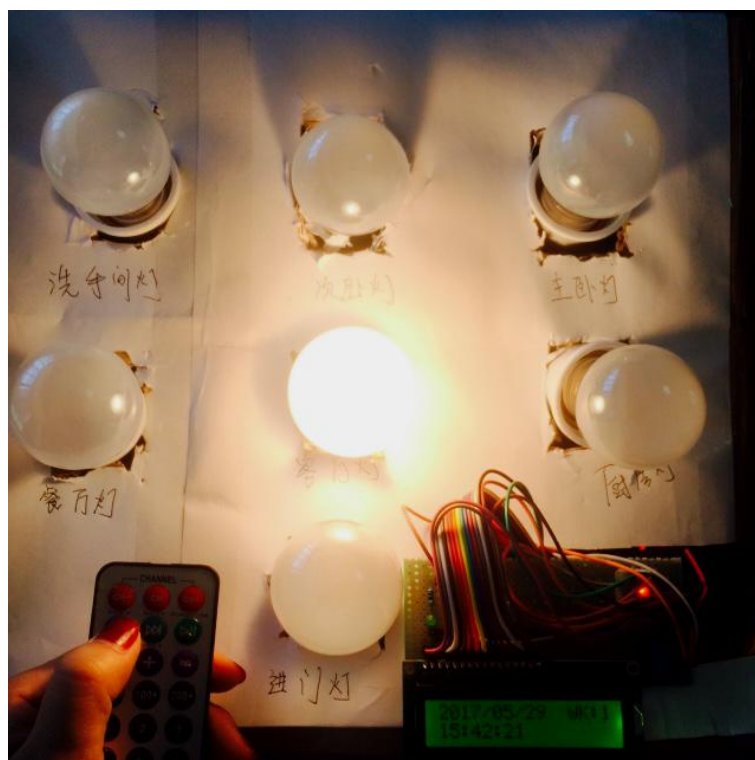


图 7.2 客厅灯光测试结果



图 7.3 回家模式灯光测试结果





图 7.4 party 模式灯光测试结果

## 7.2 RTC 实时时钟模块测试

### 7.2.1 测试目的

可以将 RTC 实时时钟的实时时间显示在 LCD1602 液晶显示屏上, 方便用户查阅。还可在特定的时间打开特定的灯光以满足用户的需求。

### 7.1.2 测试结果

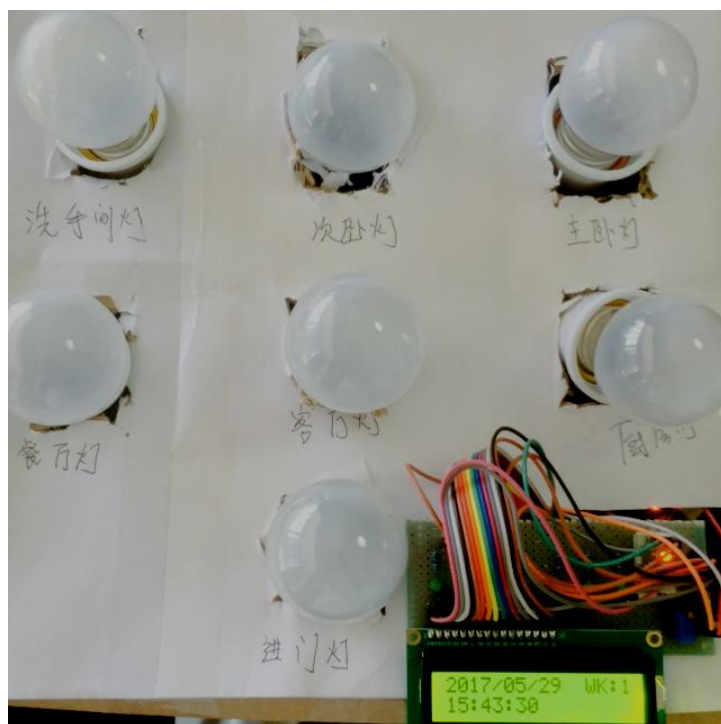


图 7.5 RTC 时钟灯光测试前显示结果



图 7.6 RTC 时钟到时间后开灯测试结果

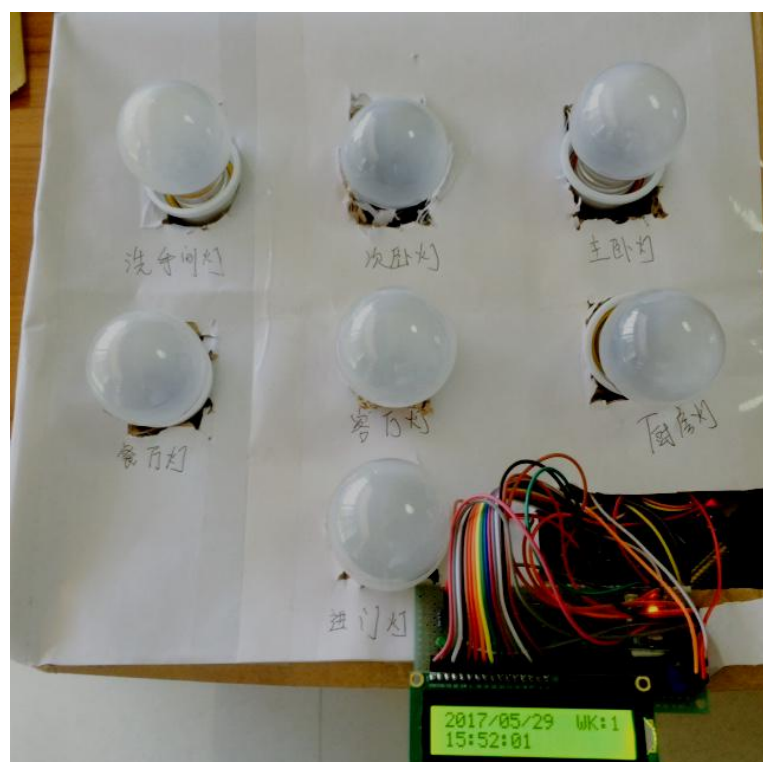


图 7.7 RTC 时钟到时间后开灯测试结果

### 7.3 上位机模块设测试

在远程控制模块方面，主要是为了方便于使用者，设计者在局域网方面进行了设定，使用户可以通过事先配置好的上位机，从电脑端对灯关开闭、场的切换、时钟设定进行手动操作。使用上位机进行控制，避免了程序控制的繁琐，用户可在电脑端进行更方便的控制。通过实现对上位机按键进行程序编写，在按键按下完成预设功能。

下图便是本次毕业设计智能灯光控制系统的上位机。



图 7.8 上位机模块设计

在打开上位机后，首先需要对端口/IP 进行设置，在协议类型中需要选择 TCP 服务器的选项。然后按下连接按键，等待观察连接端口与目标 IP 地址是否有变化，如果出现变化且连接端口是事先设定好的端口，则证明网络已经连通。然后需要进行启动服务器操作。在上位机中用户不仅可以通过编辑命令并发送命令达到用户想要完成的控制功能，还可以通过按下预先编辑好的按键完成相对应的功能操作。如需要结束控制，则直接按下停止服务器按钮即可。

下图是上位机连接好的界面。





图 7.9 连接好的上位机模块

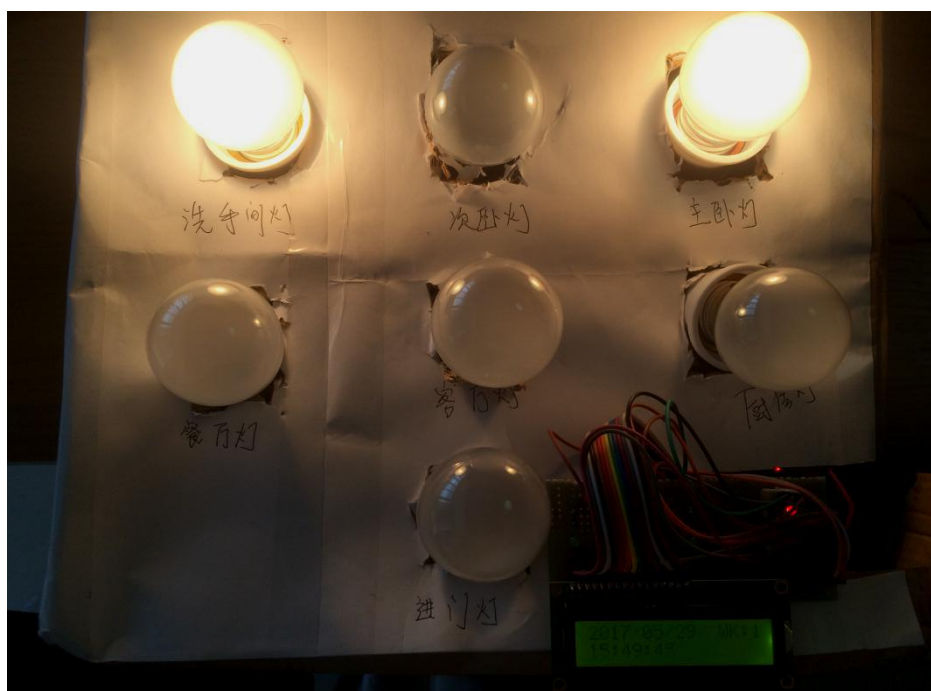


图 7.10 上位机夜晚模式 1 按键测试结果

智能灯光远程控制系统左侧，分别对应的是红外遥控按键上的键码，分别与灯光进行一一对应。第一排是对灯具的打开进行控制，第二排是对灯具的关闭进行控制，最后一排设置的是场景模式。下方分别是全部灯光打开、全部灯光关闭按钮。此外还添加了时间设置按钮，可以重新设定当前时间。旁边则是设置定时打开、定时关闭时间的按钮。下面是上位机编写的主要程序。

```

sendoutdata
End Sub
'连接目标主机
Private Sub Command2_Click()

    If bProtocol = 1 Then
        If constate = 0 Then
            Winsock1.SendData ""
            Command2.Caption = "断开"
            text1.Enabled = False
            remote.Enabled = False
            reip.Enabled = False
            Select Case bProtocol
                Case 1
                    Option1.Enabled = True
                    Option2.Enabled = False
                    Option3.Enabled = False
                Case 2
                    Option1.Enabled = False
                    Option2.Enabled = False
                    Option3.Enabled = False
            End Select
            constate = 1 '连接上
            Command1.Enabled = True
        Else
            Winsock1.Close
            Command2.Caption = "连接"
            text1.Enabled = True
            remote.Enabled = True
            reip.Enabled = True
            Option3.Enabled = True
            constate = 0 '断开连接
            Command1.Enabled = False
        End If
    End If
    If bProtocol = 2 Then
        If constate = 0 Then
            Do While (Winsock1.State <> sckClosed)
                Winsock1.Close
            Loop
            If Winsock1.State <> sckConnected And Winsock1.State <> sckConnecting Then
                'Winsock1.Close
                If text1.Text < 65535 Then
                    text1.Text = text1.Text + 1
                    Winsock1.LocalPort = text1.Text
                End If
                If text1.Text >= 65535 Then
                    text1.Text = 1024
                    Winsock1.LocalPort = text1.Text
                End If
                Winsock1.Connect
                text1.Enabled = False
                remote.Enabled = False
                reip.Enabled = False
            End If
        Else
            Winsock1.Close
            Command2.Caption = "连接"
            text1.Enabled = True
            remote.Enabled = False
            reip.Enabled = False
            Option1.Enabled = True
        End If
    End If
End Sub

```

图 7.11 上位机程序编写

## 结束语

本次毕业设计基于 STM32F103ZET6 核心开发板，实现智能灯光控制等功能。此次设计首先通过红外遥控设备来对灯光进行控制，通过 RTC 实时时钟设定当前的实时时间，再通过 LCD1602 液晶显示屏显示出来。在完成主要任务之后，又设计了远程的控制，使用户不仅仅局限于局域网内的远程控制，用户还可以通过云服务器在远方局域网外进行控制。在设计最后阶段，设计者将 LED 小灯换成了通过继电器进行连接的 220V 的白炽灯泡。这样就可以使此次设计更加贴近于实际生活当中其实用性也更强。使此次设计的成果不仅仅是用于课题的研究，也可以运用于将来的实际生活当中，使设计成果更加的生活化。

本次的毕业设计在设计初期之时考虑到可以使用于用户的家居生活，所以按照两居室的户型来设计的。但在将来的实践中，如果要使用于不同的户型，可能还需要添加相应房间的灯光。如果将来要添加灯具时，通过修改相应的程序，在相应的地方添加上新的灯具并对其进行设定即可。

总之，智能化能够改善家居生活的品质，具有很强的应用前景，具有一定的研究价值。

## 致 谢

首先非常感谢何老师、邓老师和马老师在此次毕业设计中给予我的帮助，此次的毕业设计也是在他们的悉心指导之下完成的。他们为我讲解了此次毕设题目大体的思路，为我打开了方向。从刚开始的毕业设计的选题，到后来设计所需的相关资料的查找与搜索，再到最后的设计的进一步修改与完善，老师都给予我非常多的帮助。每次我的设计出现一些解决不了的问题的时候，在老师耐心的讲解中，我理清思路找到问题，我的毕业设计才可以顺顺利利的进行下去。其次要非常感谢实验室的同学们，有了他们的帮助，使得我在硬件电路焊接的过程中，有了一个非常方便的工作环境，并且在设计中给予了我指导。正是因为有了他们，我们在遇到问题的时候才可以相互鼓励，可以沟通，也让我有了不同的思路来解决问题。

在论文即将结尾的时候，我的毕业设计也即将完成。从刚开始的选题到如今毕业设计将要结束，有太多可敬的师长、同学、朋友给予我帮助，再次诚挚的感谢他们。同时也要感谢我的母校西安邮电大学对我四年的精心培养。

## 参考文献

- [1] 孙景芝,张铁东.楼宇智能化技术[M].武汉:武汉理工大学出版社,2009
- [2] 周立功.ARM嵌入式系统基础教程[M].北京:人民邮电出版社,2008
- [3] 刘凯.ARM嵌入式应用技术基础[M].北京:清华大学出版社,2009
- [4] 贾智平,张瑞华. 嵌入式系统原理与接口技术[M].北京:清华大学出版社,200,1~25
- [5] 沈文斌.嵌入式硬件系统设计与开发实例详解[M].北京:电子工业出版社,2005,15~18
- [6] 刘晓彤.嵌入式智能家居系统远程监控系统的设计与实现[D].北京:北京交通大学,2009
- [7] 李鸿.几种智能家居网络控制系统方案的分析与比较[J].现代电子技术,2010,(03),143~145
- [8] Moller, Sebastian; Krebber, Jan; Smeele, Paula Evaluating the speech output component of a smart-home system, Speech Communication, v 48, n 1, January, 2006, 1~25
- [9] Harvey, Fiona Home smart home, Engineer, v 291, n 7605, Jun 14, 2002, 15~16
- [10] Moller, Sebastian; Krebber, Jan; Smeele, Paula Source:Evaluat\ing the speech output component of a smart-home system, Speech Communication, v 48, n 1, January, 2006, 1-23
- [11] 蔡蓝图.智能家居环境控制系统的设计与实现[D].江苏科技大学学位论文,2010
- [12] 宋凯,严丽平,甘岚.嵌入式 web 服务器的设计与实现[J].计算机工程与设计,2009
- [13] 俞侃.基于嵌入式 ARM 和 GPRS 的智能家居控制系统硬件设计与思考[J].太原城市职业技术学院学报,2013,(4):171-172
- [14] 宋威,孙运强,黄进等.基于 ARM 和 GSM 的家电远程控制系统设计[J].数字技术与应用,2013,(1):8-9
- [15] 黄涛,白创.嵌入式无线互连系统开发.第 1 版.北京:中国电力出版社,2007:168-230

