

PTcheck检测报告简明打印版

对比结果 (相似度) :

总相似度: 16% (相似字数占总字数的百分比)

红色相似度: 3% (句子相似度70%-100%的字数占总字数的百分比)

橙色相似度: 13% (句子相似度40%-70%的字数占总字数的百分比)

验真伪二维码



编号: a0efefae-d2f4-428a-8895-fcaae84dfb14

标题: 164480550726939699

作者: 1644805507269396991

长度: 15524字符 (不计空格)

段落: 250个

句子: 434句

词语: 8563个

时间: 2018/5/21 6:31:41

比对库: 学术期刊、学位论文、互联网资源

相似资源列表:

(1) 16%即2483字来源于其他网络

全文简明报告:

1引言

1.1 课题背景

{40%:随着科技的迅猛发展,智能化发展逐渐成为一种新常态,而人们生活水平也日益提高,智能家居逐渐进入人们的生活,而万能遥控器的出现则更加方便了人们的日常生活。}当你在大冬天顶着极大的压力辛辛苦苦完成高强度的工作后,下班路上冒着风雪一路吹着冷风,冻得瑟瑟发抖的时候,是不是想早早回去就能进入暖和而又舒适的房间呢,此时的你只需要你拿着手机登录进入万能遥控器的远程遥控系统,然后打开空调、调好温度,即可一回到家就能感受到家的温暖;当你晚上看着看着电视犯困就直接去卧室上床睡觉,才想起来电视忘记关机但你又懒得下床的时候,万能遥控器又能帮助你解决这普通遥控器根本不可能解决的一大难题;当你家各种家电设备的遥控器太多,有些遥控器在外观上还很相似导致你总是拿错的时候,还有一些家电的遥控器根本找不到的时候,此时此刻你只需要一个万能遥控器即可,根本不需要如此多的遥控器,不但占地方,还容易弄混,一个万能遥控器便能轻松帮你解决这些烦恼。

{84%:由于我国的红外遥控器所使用的红外遥控收发器件具有传输效率高,成本低,电路实现简单,抗干扰强等特点,目前在家用电器上被广泛使用。}而它目前以及未来的研发方向就是远程遥控,这就解决了当用户离家电设备很远的地方,也可以在想要启动和关闭的时候在较远的距离外对其进行控制的问题。

万能遥控器虽然和普通遥控器在根本构造组成以及外形上比较相近,但是万能遥控器它是可以双向通信的;层次结构上与普通的嵌入式结构相同,主要是由嵌入式芯片、嵌入式操作系统、嵌入式程序组成,不过在嵌入式操作系统里面加入了WEB系统,即一个页面支持,主要为了便于进行远程控制。本文设计的万能遥控器主要能够实现两个功能,一是具有自学习能力,适配海量设备,二是能够进行远程控制。

本课题主要运用了Python、SQLite3、Redis、HTML、Django框架以及Raspberry Pi的GPIO编程、Ubuntu系统以及Lirc等技术，设计了可以通过自学习功能来适配不同型号不同品牌的家电设备并且可以实现远程遥控的万能遥控器，解决了家庭中用一个遥控器来控制很多家电设备的问题，也满足了高速发展的科技下的新时代人们对于更加智能、方便、快捷、舒适的生活需求。

1.2 课题任务

本课题所研究的遥控器主要由红外遥控码接收模块、遥控按键控制模块、用户登录模块以及红外遥控码发射模块这几部分构成。其中红外遥控码接收模块以及红外遥控码发射模块用到了一块红外拓展板，此拓展板上集成好了红外接收器和红外发射器，并集成了两个可用的按键，分别对应GPIO27和GPIO18。{46%:红外接收器采用一个特殊的三极管，而红外发射器实际上就是一个内部材料不同于普通二极管的特殊的发光二极管，红外发射头采用陶瓷共鸣器;} {52%:遥控按键控制模块主要采用了两块4*4矩阵键盘;} 用户登录模块是用来让用户能够对家用电器进行远程控制，主要利用的器件是Raspberry Pi。Raspberry Pi是一种基于ARM的微型计算机主板，它的特点是配备有一颗ARM11系列的、频率为700MHz的CPU芯片。{49%:Raspberry Pi以SD卡为内存硬盘，卡片主板包括了四个USB接口，能直接使用鼠标键盘相连，同时还要音频和HDMI输出口，}能够处理音频、视频等文件。{42%:Raspberry Pi以其低能耗、移动便携性、GPIO等特性，并且其系统是基于Linux的，而要想实现远程控制则需要搭建WEB，}故而用Raspberry Pi完成本课题比较方便一点。

{53%:本设计主要使用了LIRC软件，即Linux Infrared Remote Control，意思是基于Linux的红外线遥控，}当用户需要遥控家电设备时，只需要按下遥控键盘的按键，后台就会判断出按的是哪个按键，根据不同的按键，LIRC会发送相应的红外指令，然后对用户需要的家用电器进行控制。

1.3 论文结构

2基于嵌入式的万能遥控器相关技术

本章节主要介绍本系统所涉及的主要技术，如Python、SQLite3、Redis、HTML、Django框架以及Raspberry Pi的GPIO编程、Ubuntu系统以及Lirc。

2.1 Python介绍

2.1.1 Python简介

{48%:Python是一种简洁并且能简单上手但其实不简单的解释性语言，大部分人也叫它为脚本语言，它没有编译这个环节，类似于PHP和Perl语言。}

{65%:使用Python语言对一个函数的设计，或者是作为面向对象的设计，它都具有非常强的可读性，相比于类似Java、C++等语言中一些英文关键字的设计，}Python语言以其独特的标点符号、命名方式、对齐方式都为人所赞美。尽管Python语言运算速度比不上C，但是其依靠各种强大的内置库和丰富的第三方库都让Python变得越来越火热，尤其是现在机器学习库的发布，{50%:更是让Python隐隐约有超过Java使用量的趋势。}

2.1.2 Python特点

Python的主要特点是：上手简单、方便维护和阅读、丰富的三方库以及跨平台兼容、各种数据库支持。

2.2 SQLite3介绍

SQLite3是一种非常轻的但是支持标准SQL语句的数据库。Sqlite3与Mysql等主流数据库一样，也支持事务操作，但是却没有像其他数

据库一样连接繁琐，它是不需要依赖服务器存在的、不需要进行繁琐配置的一种实现了自给自足的轻量级数据库。

SQLite3不需要配置意味着它根本不需要安装，无论是Linux系统还是Windows系统，都自带了SQLite3的环境，{66%:并且如果要用Python使用SQLite3，只需要安装SQLite3的Python库，即可简单使用API进行事务的ACID操作。}

2.3 Redis介绍

{55%:Redis是一种非关系性的数据库，但是Redis的主要目的并不是作为数据库使用。}Redis是基于内存的，读写速度非常快，因此Redis一般用于做缓存处理，并且Redis是key-value的，所以不管是在线程中或者是不同的会话中，都能获取到设置的值，这样就可以用来做在线和离线的一种交互。

Redis支持设置一个key的过期时间，当时间到了之后，Redis会自动删除该key-value对，这样可以用来做一种验证码的功能。

{41%:同时，Redis的CRDU是原子性的，不会发生线程安全问题。}

{49%:Redis拥有丰富的数据类型操作，如String、List、Set、Hash、Sorted Set这五种，}其中Sorted Set是一个没有重复并且每个元素带有一个Score的数据类型，可以很方便的做一个热榜排行功能。

2.4 HTML介绍

{55%:Hyper Text Markup Language，简称HTML，开发人员可以在内部使用各种标记标签来描述你的网页设计。}

{60%:Cascading Style Sheets，简称CSS，它值的是层叠样式表，当用户编写了HTML代码后，会使用CSS来美化用户的页面。}

JavaScript，简称js，它是由客户端执行的，用来控制前端中的各种逻辑，相比python、java语言的，js是非常轻编的语言。

2.5 Django框架介绍

{100%:Django是一个高级Python Web框架，鼓励快速开发和干净实用的设计。}{83%:由经验丰富的开发人员构建，它可以处理Web开发的大部分麻烦，因此您可以专注于编写应用程序，而无需重新发明轮子。}它是免费且开源的。、

{52%:快速：Django旨在快速的为开发人员搭建一个WEB应用程序。}

安全：防止SQL注入，帮助开发人员避免很多常见的安全错误。

实时性：可以在服务运行时修改代码，并自动热部署。

2.6 Raspberry Pi的GPIO编程

{55%:Raspberry Pi是一种卡片式电脑，虽然小，但是功能非常多。}Raspberry Pi的引脚包括两个5V引脚、两个3.3V引脚、8个GND引脚以及28个可以控制的GPIO引脚。Raspberry Pi实物图如图2.1所示，引脚对照表如图2.2所示。

图2.1 Raspberry Pi实物图

{67%:图2.2 Raspberry Pi引脚对照}

{57%:Raspberry Pi的引脚图及GPIO整体电路图分别如图2. 3和图2. 4所示。}

{83%:图2. 3 Raspberry Pi整体引脚图}

图2. 4 Raspberry PiGPIO引脚电路图

{60%:Raspberry Pi的复位硬启动电路如图2. 5所示。}

{60%:图2. 5 Raspberry Pi的复位电路图}

{48%:Raspberry Pi的引脚工作方式有两种，一种是Board，这种工作方式将使用板子编号将引脚接入电路，}不用考虑不同Raspberry Pi不同版本中每个引脚代表不同功能，只与板子中的引脚编号有关；一种是采用BCM工作方式，当采用BCM方式时，接入的引脚必须是BCM编号对应的Board编号。

{57%:Python程序可以设置Raspberry Pi的GPIO口输入或者输出状态。}当Python程序设置Raspberry中某个GPIO为输入状态时，可以自行焊接上拉电阻来解决输入为float类型，也可以通过软件设置上拉状态。当一个GPIO口被定义成输出状态时，可以由python程序控制输出高低状态，同时也可以通过软件来读取此时输出状态是什么。

Raspberry Pi的引脚设置为输入时，可以在代码中为该引脚注册一个中断，当电平变化时触发中断，执行相应函数。

2. 7 强大的Lirc

LIRC是一个包，允许用户解码和发送许多（但不是所有）常用遥控器的红外信号。

最近的Linux内核可以将一些红外遥控器用作常规输入设备。有时候这会让LIRC变得多余。但是，LIRC提供了更多的灵活性和功能，并且在许多场景中仍然是正确的工具。

{55%:LIRC最重要的部分是lired守护进程，用于解码设备驱动程序接收到的IR信号，并在套接字上提供信息。}如果硬件支持，它还接受要发送的IR信号的命令。

Lirc允许您使用遥控器控制电脑。只需按一下按钮，即可将X11事件发送到应用程序，启动程序等等。可能的应用是显而易见的：红外鼠标，电视调谐卡或CD-ROM遥控器，远程关机，用计算机编程VCR和/或卫星调谐器等。在Raspberry Pi上使用lirc很受欢迎。

2. 7. 1 安装Lirc

{61%:在Raspberry Pi下安装Lirc是非常简单的，只需要在终端执行一下图2. 6的命令就好了。}

图2. 6 linux安装lirc

安装完之后执行图2. 7的命令：

图2. 7 leafpad打开lirc配置文件

修改成图2. 8的内容：

图2. 8 更改lirc的相关信息

2.7.2 配置Lirc引脚

Lirc需要配合红外接收器和红外发射模块使用的，因此用户需要在Raspberry Pi中配置红外拓展板中红外接收和发送引脚。

{44%:首先运行如图2.9所示命令，打开Raspberry Pi的模块定义文件: }

{65%:图2.9 打开Raspberry Pi模块文件}

添加如图2.10里面的内容:

图2.10 为lirc添加相关信息

保存配置后，重启Raspberry Pi使得配置生效。要使用lirc录制红外，则只需在终端输入如图2.11的命令。其中的name参数是为你接下来录制的设备取一个名字，名字可以是任意的，本课题采用的是通过UUID来为每一个设备命名，以防止学习的重复。

图2.11 开始使用lirc录制红外

录制开始时，lirc要求使用者必须拿要学习的遥控器按下该遥控器的每一个按键，每一个至少按一次，一共需要按满80次，lirc通过接收到的信号来计算该遥控器的gap值，计算完毕后，才能正式开始学习功能。学习期间，lirc要求使用者必须键入lirc库里面存在的按键值，输入完之后，才接收对应遥控器的按键的红外信号。当使用者不再进行学习时，按回车即可退出学习。

学习完lirc会将学习到的按键信息以文本文件保存在当前目录，文件名为使用者录制时设定的name参数。{40%:但是现在还不能使用刚学习的遥控器，使用者还需要将这个文件复制到/etc/lirc/lircd.conf中才能使用。使用lirc发射红外指令的命令如图2.12所示。

图2.12 使用lirc发送红外指令

在该命令中，name代表的是lirc录制时的name参数，key_name代表的时使用者为每个学习到的按键取得名字。

3 基于嵌入式的万能遥控器的设计与功能

本章节主要是从大范围的讲述整个系统的基本情况如系统的模块框架、系统的开发流程图以及系统的设计方案，具体的硬件内容实现和软件内容实现将在后面章节中一一讲述。

3.1 整体架构

基于嵌入式的万能遥控器分为硬件部分和软件部分，其中软件部分有：信息验证模块（登录模块、登出模块、修改密码模块）、信息展示模块（远程遥控器模块、学习完后的遥控器在WEB端命名模块）以及远程遥控模块（通过RESTful API进行网络请求来远程遥控模块），{48%:硬件部分有：遥控器学习模块（键盘扫描模块、红外指令存储模块）以及遥控器的使用模块（查询当前遥控器属于哪个设备模块、发送红外指令模块）。}其整体架构如图3.1所示。

图3.1 基于嵌入式的万能遥控器整体架构图

远程遥控系统通过登录模块进行用户名密码验证，验证通过后将可进入系统，进入系统后可以控制远程设备，通过网页样式设计的一个虚拟遥控器，每一个按键会触发一个网络请求，后台会将对应的红外编码发出去。在开发板中，编写了键盘扫描程序，不同的按键

代表了不同的信息，用户按下学习键即可开始学习一台新的设备。

3.2 整体功能

{40%:基于嵌入式的万能遥控器的难点有以下几个：1、如何接收红外信号和发送红外信号？}这个问题将使用lirc和红外接收发送扩展板解决；2、接收到的信号如何保存保存呢？后面将使用SQLite3作为整个系统的数据库，以用来存储；3、如何远程遥控？这个问题将使用Django来搭建一个web服务。系统整体功能流程图如图3.2所示。

图3.2 万能遥控器系统功能流程图

要远程遥控设备要进行登录，避免被人随意使用。为画流程图方便，所以画了两个数据库框图，但其实是同一的数据库。上面表示主要用于web端删改查操作，下面的主要表示用于学习设备时，为学好的设备产生的文本文件的路径以及学了哪些按键进行持久化操作。

从上面的整体流程图可知，整体系统都是分为几个大的模块，模块与模块之间留有通信的接口，使得整个系统中的各个模块又紧密相连。每个模块之间又有许多的小功能点，每个功能点只负责自身独有的逻辑任务，这使得耦合性降低极低，提高了代码的可重用性。

本课题采取SQLite3作为数据库，并且取数据库的名字为db.sqlite3，它是WEB部分与硬件交互的桥梁。硬件将学习完成的情况持久化到数据库中，WEB部分从数据库中读取，以实现远程功能。

3.3 设计原则

低功耗设计：整个系统基于Raspberry Pi开发板，仅需USB供电，并且其余器件也是选用的低且可控功耗的器件。

模块化设计：整个系统采用模块化设计，每个模块单独实现功能再整合成完整系统。每个模块下面又分了很多小功能，每个小功能也尽量是实现高内聚低耦合方式，所以不会产生重复开发的情况，让得系统更简洁、更高效。

能优化则优化：整个系统开发过程中，不管是软件还是硬件，对于能优化的部分都进行了优化，比如说开发中硬件的绝大部分都是采用杜邦线连接，{45%:而不是在PCB板子中焊接，减少了电路与电路之间的干扰，提高了抗干扰能力。}

4 硬件设计

此章节将详细的描述整个硬件的部分，包括整个硬件电路的设计、红外模块电路的设计、按键模块的设计以及指示灯模块电路设计。

4.1 硬件整体方案

尽管硬件是模块化设计的，但是每个模块之间都互相有联系。系统以Raspberry Pi为主控，用2个4*4的矩阵键盘作为遥控键值的输入，以红外拓展板作为红外信号的接收和发射，Raspberry Pi的GPIO编程作为整个业务的逻辑判断，设计图如图4.1所示。

图4.1 嵌入式万能遥控器硬件设计

4.2 红外拓展板与Raspberry Pi

4.2.1 红外拓展板

{69%:红外拓展实物图如图4.2所示。}

图4.2 红外拓展板

这个红外拓展板是针对Raspberry Pi的专用红外控制拓展板，用户可以使用该拓展板配合Raspberry Pi完成各种遥控和被遥控的功能。
{44%:红外拓展板使用5v及3.3v配合供电，红外接收引脚连Raspberry Pi 的GPIO18 (BCM)，}{57%:红外发射引脚接Raspberry Pi的GPIO17 (BCM)。}这个红外拓展板的工作特点是：

{94%: (1) 红外线接收功能：工作频率：38KHZ，接收距离：18-20m，接收度角度：+/-45度；}

{95%: (2) 红外线发射功能：波长：940nm，发射距离：7-8m；}

(3) 可以配合Raspberry Pi进行lirc的使用

红外拓展板中红外接收器电路如图4.3所示。

4.3 红外接收电路

红外拓展板中红外发射器电路如图4.4所示。

4.4 红外发射电路

4.2.2 红外协议之NEC协议

现在市面上的红外遥控器基本都是基于NEC协议的。NEC协议的载波为38Khz，协议规定：Logic"1" 时间为 2.25 ms， Logic"0" 时间为1.12ms，其中不管是Logic"1" 还是Logic"0" 脉冲时间都为560us。如图4.5所示。

4.5 NEC协议逻辑01示意图

NEC协议规定了其自身的格式：先发送13. {59%:ms的开始码，开始码由9ms的高电平和4.5ms的低电平组成，接下来发送的是地址码，共8bit，然后发送的是8bit的地址反码，第五部分是命令码，}这个是用来判断键值的，最后一部分是命令反码。整个协议中命令反码的作用是用来做数据校验的。NEC脉冲链如图4.6所示。

4.6 NEC脉冲链示意图

当一直按着一个键时，NEC并不是一直按照其脉冲链发送上述的信号，而是第一次的时候严格按照其协议发送，而后面发送的都是重复码。{85%:重复码是循环发送的，间隔为110ms，重复码由9ms高电平、2.25ms低电平以及560us高电平组成。}NEC重复发送载波协议如图4.7所示，NEC重复码如图4.8所示。

4.7 NEC重复发送载波协议示意图

4.8 NEC重复码

4.2.3 红外编码

{50%:所谓的编码其实就是对于红外发射来讲的，按照NEC规定的协议，在高电平时间内，Raspberry Pi的GPIO17 (BCM) 输出固定频率载波；}低电平则直接输出低。{89%:红外接收头接到载波时输出高电平，没有载波时输出低电平，这样就完成了数据解码。}

4.3 Raspberry Pi和2个4*4键盘

基于嵌入式的万能遥控器一共设计了25个按键，如果直接焊接25个按钮并使用25个引脚作为输入的话，那么不仅代码工作量要增加很多，并且大多数还是一些重复性的代码，还会导致Raspberry Pi的GPIO也将占用多达25个，这样既不好管理，也让代码冗余，所以选择了2个4*4的矩阵键盘。

4*4矩阵键盘是使用16个按键组合设计的一种特殊的键盘，它只需要接Raspberry Pi 8个GPIO口，通过程序设计进行键盘扫描，程序可以很轻易的获取到用户按下的到底是什么键。

键盘扫描原理：如图4.9所示，由于该键盘是4*4的，有四行四列，所以会有八个引脚，假设这八个引脚分别是Raspberry Pi的GPIO10、GPIO4、GPIO3、GPIO2、GPIO24、GPIO23、GPIO15和GPIO14，前四个引脚的初始状态为[1, 1, 1, 1]，后四个引脚的初始状态为[0, 0, 0, 0]，当有按键按下时一定会有后四个引脚中的某个低电平把前四个引脚中的某个高电平拉低，也就是说前四个引脚和后四个引脚中分别会有一个低电平，其余引脚皆为高电平，当两次状态不一致时，程序就判断为有按键按下。具体是哪个按键又该怎么进行判断呢？

图4.9 设计的矩阵键盘1

首先程序将GPIO14引脚置低电平，其余均为高电平，如果此时引脚状态不是[1, 1, 1, 1, 1, 1, 1, 0]，那么按下的按键一定是为第一行中的某个，这是因为按键按下将导致电路导通，高电平被拉低，有以下几种情况：

- (1) 若为S1按下，GPIO2引脚的电平被拉低，则输出值为1110 1110；
- (2) 若为S2按下，GPIO3引脚的电平被拉低，则输出值为1101 1110；
- (3) 若为S3按下，GPIO4引脚的电平被拉低，则输出值为1011 1110；
- (4) 若为S4按下，GPIO10引脚的电平被拉低，则输出值为0111 1110；

那如果引脚状态就是设置的[1, 1, 1, 1, 1, 1, 1, 0]呢，则代表按键按下的不在第一行，程序将以此对GPIO15、GPIO23、GPIO24置低电平，并进行上述同样原理的判断。

所以，在键盘扫描过程中，会从第一行开始逐行扫描，并且无论哪一行的哪个键被按下引脚状态都会唯一对应一个特定的值，
{45%:通过一个字典做一个key-value的映射，由此就可以判断出具体是哪个按键按下了。}

图4.9是本系统设计的第一块4*4矩阵键盘，其按键与对应的键值都有在图中标出。第二块4*4矩阵键盘如图4.10所示。

图4.10 设计的矩阵键盘2

本系统的键盘扫描程序是通过Python的第三方库pad4pi实现的，pad4pi是专门为Raspberry Pi实现键盘扫描的一个第三方库，使用者仅需要在创建一个keypad程序时传入选择的4个行GPIO引脚、4个列GPIO引脚和一个4*4的列表KEYPAD，KEYPAD里面存储的是每个按键对应的键值。pad4pi内部也是根据上述原理做的一个程序，它先将使用者给出的行列引脚进行内部绑定，然后创建一个字典，由于在4*4键盘中，每个按键按下程序设计使得8个引脚的输出都是固定的值，这个值它在内部先定义了16个字符串，在KEYPAD列表传进去时，pad4pi内部首先创建一个字典，然后会对列表进行循环，然后在字典中追加一一对应的key-value对，当程序读取到8个引脚状态改变时并扫描到哪一行时，直接根据此时的引脚状态获取到按键值，并将此按键值传递到中断函数中，作为中断函数的参数。图4.11是安装pad4pi的指令。

图4.11 安装pad4pi

4.4 Raspberry Pi和提示灯

系统设计了4个提示灯，编号为ABCD，当在学习状态是，A灯会亮；当B灯亮时则说明使用者应该要按下矩阵键盘中的某个按键，好让程序能获得一个键值，使lirc能继续学习；当C灯亮时，说明使用者虽然按下了按键，但是10s内没有收到红外信号；而D灯是当使用者把Raspberry Pi当做普通遥控器时，如果按下一个有用的按键，则D灯闪烁。相关电路图如图4.12所示。

图4.12 提示灯电路图

这四个引脚定义在OUT_PINS列表中，其中GPIO1、GPIO13、GPIO16、GPIO19分别为OUT_PINS[0]、OUT_PINS[1]、OUT_PINS[2]、OUT_PINS[3]。后面的论述中将使用OUT_PINS代替。

4.5 学习功能与遥控功能的实现

学习与遥控整体程序流程图如图4.13所示。

图4.13 学习与遥控流程图

4.5.1 学习功能

系统学习与本地遥控的代码是单独放在一个文件的，不依赖于web系统，有Python单独启动一个进程运行。代码里面通过一个死循环来确保程序一直运行而不结束，通过对键盘的扫描并实现中断，在中断函数里面更改当前按键信息，循环判断按键是否是学习键来开始学习、退出学习或者是普通遥控功能。当学习结束之后进入数据库更新操作。

系统学习功能与遥控功能的实现依赖于强大的lirc软件，但是lirc是基于Linux的，所以用python操作的话必须要使用Python与linux交互的工具，这里采用的是Python的第三方库pexpect和subprocess。首先安装这两个库，执行如图4.14所示的命令。

图4.14 安装pexpect和subprocess

Pexpect能很方便的与Raspberry Pi的linux系统通信，当使用pexpect执行一条linux命令时，它会创建一个阻塞的线程，直到这条linux命令运行结束或者异常退出，在线程阻塞时，pexpect可以对linux命令的返回结果进行匹配，匹配成功返回匹配的索引，开发者可以根据不同的索引做出不同的操作。如果什么也匹配不到，则会一直阻塞，直到用户主动发送退出指令或者超时。

{49%:Subprocess则是可以运行linux命令并将结果以字符串形式返回。}这两个库是完成学习功能的基石。

数据初始化：本课题采用的Raspberry Pi的GPIO工作方式BCM，所以在开始的时候设置GPIO的工作方式。Raspberry Pi GPIO工作中，如果特定功能的引脚设置为其它用途，将会发出警告，所以代码中将警告设置为不警告。此外还初始化了几个全局变量：study (bool类型，用于判断是否按下学习键，默认为False)、study_key (list类型，用于保存按下过的键值)、press_key (str类型，用于保存当前按下的按键的键值)和remote_name (str类型，用于保存当前遥控器的名字)。

键盘扫描和中断的注册：首先定义矩阵键盘对应键值、行引脚和列引脚，由于本课题使用了两个键盘，所以不管是键值、行引脚列引脚都要定义两种，如图4.15所示。接着将定义的值分别传入pad4pi，这样键盘扫描程序就定义好了，再使用pad4pi为每个键盘注册中断，中断函数为同一个 (printKey)。

图4.15 键盘扫描及中断函数注册

中断函数里面的内容很简单，它带了一个参数key，进入中断时，这个key值就是当前使用者按下的那个键的键值。中断函数流程如图4.16。

图4.16 触发中断时的逻辑处理

学习逻辑功能实现：当使用者第一次按下学习键时，全局变量study将会在中断函数中变成True，程序运行学习功能。学习功能逻辑使用的是利用pexpect执行lirc的录制红外的命令，命令执行后，可能产生的结果一共有7中，如图4.17所示。

图4.17 录制过程中可能返回的结果

现在说明这7种情况：

- ① Lirc录制指令刚运行时，需要按下回车才能继续执行；
- ② Lirc需要再接收一个回车，才能开始录制，计算gap值；
- ③ 在开始录制时，如果10s内没有接收到红外信号，linux命令退出程序逻辑在匹配退出是处理，此时不处理；
- ④ 输入一个想要的键值，按回车继续
- ⑤ 输入键值后，10s没有接收到红外信号；
- ⑥ Linux命令执行完毕
- ⑦ 超出一定时间没有匹配到linux返回的上述任何一个值，超时继续等待，此时不处理。

学习的整个功能流程图如图4.18所示。

图4.18 学习功能流程示意

当index为3时，对应了情况④，此时需要等待使用按下按键触发中断，将press_key赋值，程序会将press_key发送到linux进程，重复如此，直到用户再次按下学习键，学习结束。这个逻辑流程图如图4.19所示。

图4.19 按键与学习

当学习完成后，程序会将当前学习好的文本内容复制到lirc的当前设备信息中，并且调用update函数，将学习完成的按键持久化到数据库中，流程图如图4.20所示。

图4.20 学习完成功能示意流程图

Web和硬件能够交互就是依靠这最后一步了，将学习好的遥控信息持久化到数据库中，update函数流程图如图4.21所示。

图4.21 数据持久化流程示意图

4.5.2 遥控功能

如果使用者没有按下学习键，那么此时Raspberry Pi相当于遥控器，使用者可以按下任何一个键（如果该键学习过），
{59%:Raspberry Pi就会通过lirc将对应的红外信号发出。} 万能遥控器的遥控功能逻辑示意图如图4.22所示。

图4.22 万能遥控的遥控功能示意

因为lirc发送红外是需要两个条件的，第一个是设备的名字，第二个是设备的按键名字，所以只有两个都不为空的时候才能进行发射。但是，有时候会有没有把矩阵键盘上的所有按键都学完，那么就会发送失败，即什么也不操作，发送成功时让灯闪一下。获取remote_name时是依赖lirc的打印当前设备名字的命令：“sudo irsend list “ ””，这条命令会返回一条字符串，格式下：“irsend:”，“*”代表的就是当前设备的名字，程序使用了字符串分割来获取remote_name。

4.6 硬件设计总结

硬件设计就是整个学习和遥控功能，依赖于Raspberry Pi、红外拓展板、指示灯和两个矩阵键盘。在整个设计中，使用proteus对硬件电路进行了多次的设计与修改，使用visio绘画了多种可能的流程图。设计过程中进行了大量的实验，让学习与遥控部分变得更加的可靠。

5 WEB端设计

本章节将详细描述WEB端部分，包括了数据的设计、信息验证模块、远程遥控模块、设备命名模块。

5.1 WEB端整体架构

WEB端使用Django快速开始的一个项目，项目命名为xupt-ghj，一共分为6大模块，每个模块之间都是采用的RESTful设计。其总体架构图如图5.1所示。

图5.1 WEB端整体架构

5.2 基于嵌入式的万能遥控器的数据库设计

本课题只使用了两个表，一个是T_XUPT_USER表，它里面存储着一些登录验证信息，如账号密码姓名等；一个是T_XUPT_DEVICE，它里面存储的是设备的真实名字、学习过的按键、绝对路径、使用者起的名字以及一个用来表示是否命名完成的状态。

{46%:数据库的建表操作是由使用者在Django的model中定义字段类型，再经由Django自动的创建表。} 表T_XUPT_USER的定义如图5.2所示。

图5.2 表T_XUPT_USER的设计

表T_XUPT_DEVICE的字段有点多，因为它包含了每个按键，而系统设计的时候采用了两个矩阵键盘一共使用了23个学习键值。表T_XUPT_DEVICE的定义如图5.2所示。

图5.3 表T_XUPT_DEVICE的设计

表设计完成后，在命令行先执行如图5.4的命令，这个命令意思是说创建了一些数据库日志，在之后使用者可以通过这些日志进行数据的更新和回滚；然后再执行如图5.5的命令，这是说读取数据库日志，再进行数据库的更新。

{47%:图5.3 创建数据库日志}

图5.3 更新数据库

{55%:接下来django就在sqlite3数据库中创建了这两个表。}

5.3 信息验证模块

信息验证模块包括两个模块，一个是登录模块，这是进入远程遥控系统的唯一路径；一个是密码修改模块，这是考虑到使用者忘记密码无法登录时，可以通过修改密码来重置密码，再进行登录。框架图如图5.4所示。

图5.4 信息验证框架

5.3.1 登录模块

登录模块是整个远程遥控系统的起点，页面使用jQuery进行渲染修饰，当用户输入完账号密码点击登录时，首先js会对使用者的输入进行非空校验，如果账号密码有一个为空都会进行提示不能为空，校验通过后，浏览器会向后台发送一个携带了用户名密码的请求到后台，整个逻辑流程图如图5.5所示。

图5.5 登录框架

后台接收到用户名密码后，进行登录校验，校验成功，则在当前会话中保存用户的登录信息，本系统保存的是用户的账号，然后返回ok。{69%:如果登录校验失败了，则返回none。}前台收到返回的消息时，则进行判断，是ok则跳home页面，否则停留在login页面。Login页面如图5.6所示。

图5.5 登录页面

{89%:当用户名或者密码为空时提示如图5.7所示。}

图5.5 账号密码为空时提示

5.3.2 修改密码

{59%:使用者可以通过点击登录页面中的“Forgot password?”来进入修改密码页面，修改密码页面如图5.6所示。

图5.6 修改密码页面

使用者通过输入自己的邮箱来获取验证码，验证码为4位随机数字，保存在redis数据库中，并设置了一分钟有效期，使用者必须要验证码失效之前进行验证码的确认。流程图如图5.7所示。

图5.6 获取验证码流程

其中给邮箱发送验证码功能是通过以在163申请的一个邮箱账号配合python内置的smtplib库和MIMEText，验证码通过4次循环生成4个随机数生成。

使用者得到验证码后，可进行验证码校验，校验通过后redis将删除验证码，并将邮箱作为key重新设置value值为True，超时时间为

5分钟，使用者需要在5分钟内进行密码修改，否则需要重新获取验证码。{93%:密码修改界面如图5.7所示。}

图5.7 更改使用者的密码界面

使用者需要输入两次密码，在后台和前端都有进行两次输入的密码是否一致的验证，两次一致时，密码才会修改成功，然后返回到登录页面。

使用者登录成功后将看到home页面，home页面如图5.8所示。

图5.8 主页面显示

Home页面中可以从左侧功能导航栏进入查看未命名设备和我的家电页面。

5.4 未命名设备

在硬件设计部分，红外学习功能使用的是lirc，它在录制时需要指定一个name参数，学习代码中使用的是uuid来代替name参数，但是uuid仅仅只是一串在同一时空下不重复的字符串，本身并没有任何意义。所以在学习完成之后，使用者需要登录到系统中，此时系统会提示刚刚学习的设备没有命名，没有命名的设备是不允许被使用的。

使用者可以通过任何页面的右上角的信息进入未命名页面，学习过的所有未命名的设备都会在这个页面显示，如图5.9所示。这些数据由跳转到这个页面时候携带而来的，在跳转到这个页面之前，后台先进行查询操作，由于没有命名的设备，在数据库中的记录中的state字段会是0，所以未命名设备查询所有设备并过滤state字段为0，相关逻辑如图5.10所示。

图5.9 未命名设备的展示

图5.10 获取未命名设备

使用者点击删除时，ajax请求会携带该设备的id，后台收到id之后，执行sql删除语句。删除不能恢复。

{71%:点击命名时，弹出输入框，如图5.11所示。}

图5.11 输入新的名字

确定后，后台处理逻辑流程图如图5.12所示。

图5.12 重命名的逻辑处理

重命名逻辑判定同名设备是不行的（基于一种智能家居想法，一个家庭里面可以做到为每一个自己学习的设备起一个不同的名字），命名成功后，将刷新本页面，同时该设备不再在该页面中，会出现在我的家电页面中。

5.5 我的家电

我的家电页面展示了所有学习好了并且命名好了的家电信息，在前端的设计只显示了序号、自己命名的名字、创建时间、类型以及操作列。家电信息的获取和上文中未命名设备信息的获取原理是一样的。

操作列同样有两个功能，一个是使用该设备，一个是删除该设备，删除逻辑和未命名设备的删除是同一个，所以系统设计的时候也只

是设计了一个API，两个删除都是使用ajax请求同一个动作。

整个远程遥控系统的中心在于点击了使用设备。本课题设计了一个网页版的虚拟遥控器，如图5.13所示，当使用者点击使用按钮时，则会进入虚拟按键界面。这个按键的所有按键都是与硬件中的两个4*4矩阵键盘的键值是一一对应的。

图5.13 网页虚拟遥控器示意

由于存在学习过程中只学了一部分按键，所以可能不会所有按键都有效，本课题的设计是如果该设备没有学习那个按键，则使这个按键禁用，让使用者不能点击，如图5.14所示。

图5.14 尚未学习的按键示意

没有学习到的按键会变灰且不能点击。这个功能的实现是基于jQuery这个js文件封装库的。逻辑如图5.15所示。

图5.15 按键能否点击逻辑图

部分代码如图5.16所示。

图5.16 按键能否点击代码

系统设计时为每个按钮都增加了一个value属性，值为" name-key"，name是使用者当前使用的设备名字，key为每个按钮对应的键值。同时设计时也为每一个按钮添加了一个点击事情，所有的点击事情指定了同一个函数，这个函数将接受点击的那个按键的value值。远程遥控前端js逻辑如图5.17所示。

图5.17 远程遥控发出网络请求

经过ajax请求，后台就能拿到对应的设备名字以及键值了，后台相应的发送红外的代码如图5.18所示。

图5.18 远程遥控发射红外代码

首先通过自定义的名字来获取想要使用的设备的绝对路径和当前设备的真实名字(uuid)，然后将其复制到lirc的设备默认路径。接着使用subprocess运行lirc红外发送命令，如果返回结果包含拒绝信息，那么是lirc没有启动，所以启动lirc后再发送一次红外指令，{41%:如果没有返回字符串则代表发送成功，则直接返回True，前端将提示发送成功;}有返回字符串，则表示红外发射失败，前端提示发送失败信息。

5.6 WEB端总结

自此，整个基于嵌入式的万能遥控器的设计就结束了。Web端的设计查阅了非常之多的资料，尤其是前端和后台的交互方面的资料，课题设计里每一种情况都有做考虑，所以本课题的容错率是较高的。

6 总结

整个系统设计是一个从无到有的过程，从开始拿到毕设课题时的迷茫到现在完成了毕设，整个过程有太多的艰辛，伴随着太多的谷歌查资料、图书馆看书。所有相关技术的一点点积累，才逐渐有了硬件、WEB的联系交互，才逐渐有了硬件分哪些模块、WEB分哪些模块的想法。

系统设计的过程让笔者自我沉淀许多，在这个自我积累的过程中，笔者也收获了很多。尽管想清楚怎么分模块，但是对于各个模块的功能实现还是一筹莫展，经过多次上网搜索和去图书馆参阅资料，最终发现了lirc这个开源软件。确定了基石，那么建高楼的材料就更好确定了，因为lirc是运行在linux系统上，所以便采用了Raspberry Pi作为整个系统的容器。

选好材料后，对模块的划分又更加明确了，然后是查看文档，将相关的技术熟悉起来。

设计的过程无疑是非常消耗时间且无聊的，但是在这个过程中能学到很多东西却又让人非常开心，尤其是当一步一步设计都有成果时让人非常雀跃。

基于嵌入式的万能遥控器还不是一个非常完善的作品，还有许多性能问题依旧可以设计的更加高效，当然这是由于作者目前的水平有限。它目前有的功能：

- (1) 学习功能
- (2) 遥控功能
- (3) 远程遥控功能

后面的话会完善更多的功能，作者亦将不断创新、学习。

7 展望

近年来，红外遥控器的发展也越来越快，许多遥控器上都出现了学习功能，然而绝大多数遥控器其实并没有学习功能，更不要说是远程遥控。人们的需求在增加，一个一个的遥控器已经很难满足人们的需求了，因为一旦当遥控器坏了还得要跑去买一个专门的遥控器。基于嵌入式的万能遥控器，及支持学习功能，有支持远程遥控功能，试想，人们远在千里之外都能遥控家里的家电的设备，
{40%:这又怎能不让人喜欢、怎能不满足人们的需求呢，所以基于嵌入式的万能遥控器具有非常深远的意义和非常广阔的市场前景。}

本系统的设计仅仅采用了网页版的远程遥控，笔者认为这是完全不够的，基于嵌入式的万能遥控器还有开发的必要性，可以介入微信的控制、可以适配安卓APP或者是ios苹果应用的控制，也可以接入一个网络摄像头，这样人们还可以试试查看设备的运行情况，或者增加自定义定时的功能。打造一个越来越智能的遥控器，您觉得您会喜欢吗？

检测报告由PTcheck文献相似度检测系统生成

Copyright 2007-2018 PTcheck