**西 安 邮 电 大 学**

**毕 业 设 计（论 文）**

题 目： 基于树莓派的人脸识别系统的开发与实现

院 （系）： 自动化学院

专 业： 自动化

班 级： 自动1403班

学生姓名： 彭大富

导师姓名： 马翔 职称： 讲师

起止时间：2018年 3 月5日至2018年6月15日

**毕业设计（论文）诚信声明书**

本人声明：本人所提交的毕业论文《基于树莓派的人脸识别系统的开发与实现》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。

本人完全意识到本声明的法律后果由本人承担。

论文作者： 时间： 年 月 日

指导教师已阅： 时间： 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 | 彭大富 | 学号 | 06141083 | 专业班级 | 自动1403班 |
| 指导教师 | 马翔 | 题目 | 基于树莓派的人脸识别系统的开发与实现 | | |
| 选题目的（为什么选该课题）  人工智能（AI）是未来发展的趋势，而AI又分好多种类，比如机器学习，计算机视觉，等等。自动化专业的我对这方面也稍有涉及，也想要更加深入的取了解一下这个方面的东西。 | | | | | |
| 前期基础（已学课程、掌握的工具，资料积累、软硬件条件等）  1、网络基础：了解网络通信的有关协议  2、Python：学习AI的主流编程语言  3、OpenCv：进行人脸检测 4、腾讯优图开发SDK（python-sdk）：提供人脸识别  5、树莓派：开发系统（软件环境、硬件设备）  6、回归、聚类、分类、降维等算法知识 | | | | | |
| 要研究和解决的问题（做什么）  通过使用树莓派采集人脸照片，进行刷脸签到，记录在考勤系统中，web端使用PC端浏览器调用摄像头刷脸登录查看自身考勤记录 | | | | | |
| 工作思路和方案（怎么做）  1、树莓派安装usb摄像头  2、控制树莓派使用usb摄像头拍照，并使用文件IO将照片保存  3、使用Opencv检测出人脸，将人脸照片重新设置尺寸，仅保存人脸照片，命名face.jpg   1. 将face.jpg通过优图Python-SDK传至人脸库进行人脸预测 2. 预测结果进行阈值筛选，签到成功   6、签到记录使用web端展示 | | | | | |
| 指导教师意见  签字 年 月 日 | | | | | |

**摘 要**

众所周知，无论人们在什么企业上班，不管是弹性上班制还是按时按点上班制度，都是需要进行签到的；无论学生们在什么学校上课，老师也是会在课堂上进行点名。因此，本课题设计基于树莓派的人脸识别考勤系统，拟提供多元考勤方式中一种较为新颖的方式。

系统是基于树莓派开发板的，分为WEB部分和硬件部分。web部分采用了当前较为流行的python语言，并使用其WEB框架Django，完成了一个考勤记录的查看、补签、审批等。硬件部分使用opencv进行配合树莓派的摄像头进行人脸检测，并把一帧一帧的图片中人脸部分画出一个框，将人脸包裹，当按键按下时，opencv将当前帧的图片保存，后台用腾讯优图的python-sdk库将人脸识别，返回人脸信息，然后进行签到逻辑。

基于树莓派的人脸识别签到系统，因为树莓派的轻便，而使得整个系统的使用方便快捷，具有很大的实用性。

**关键字：** 人脸识别、树莓派、考勤系统

**Abstract**

As we all know, no matter what companies are working in, whether they are working on flexible schedules or on time, they all need to sign in. No matter what school students are in, the teachers will also be named in class. Therefore, this project is designed based on the Raspberry Pi face recognition attendance system, which is intended to provide a more novel way of multi-attendance.

The system is based on the Raspberry Pi development board and is divided into the WEB section and the hardware section. The web part adopts the current popular python language and uses its web framework Django to complete the review, supplement, and approval of an attendance record. The hardware uses opencv to cooperate with the Raspberry Pi camera for face detection, and draws a frame of the face part of the frame-by-frame picture to wrap the face. When the button is pressed, opencv will take a picture of the current frame. Save, background recognition using Tencent python-sdk library to face recognition, return face information, and then check-in logic.

Based on the Raspberry Pi-based face recognition sign-in system, because of the lightness of the Raspberry Pi, the entire system is easy to use and has great practicality.

**Key words：**Face recognition, Raspberry Pi, Attendance system

**第一章 绪论**

**1.1 课题背景**

人们总能遇到这样的窘境：当你去上班的时候你才发现你忘记带了工卡，或者你的工卡突然刷不了，那么你将无法签到，同时还要面对上级的询问；当你上课你没有带一卡通，或者你的一卡通掉了还没有补办，那你不得不去找老师，说明你的情况。那么，有什么办法可以解决这种窘境呢？如果人们能什么都不带，仅适用自己的特征来代替不就好了？基于树莓派的人脸识别考勤系统就是基于这个出发点而实现的一个系统，它的出发点就是为了解决当人们不依靠工卡等其它卡片设备进行考勤时，可以使用人脸识别来签到，这样的话，就算你忘记带卡了，你也可以很方便的签到了。

基于树莓派的人脸识别考勤系统，主要是用python配合opencv和腾讯公司优图子公司的人脸识别开发库python-sdk实现的，做到就算没有带一卡通之类的信息证明卡片也能进行正常签到了。那么，要实现这么一个功能，课题便要求要解决人脸识别、签到逻辑、签到查询和查看、补签、审批等问题。本课题采用的主控板是树莓派开发板，树莓派不仅是一个linux服务器，同时它也提供了强大的GPIO编程功能，而且还具有高性能、低功耗的特点，这样的话web服务可以直接部署在树莓派中，硬件方面的东西也可以直接由树莓派控制，极大地方便了开发人员的操作。

本课题在分析了各种考勤情况，主要解决了忘记带卡、代打卡、考勤参数设置不灵活的问题。

**1.2 课题任务**

本课题主要是研究基于树莓派的人脸识别考勤系统，主要实现了人脸识别和考勤相关的内容。设计使用了opencv计算机视觉函数库，配合树莓派摄像头进行视屏流的采集分析，检测是否有人脸存在。当用户按下按键的使用，将用户的人脸传给python-sdk进行识别，得到识别结果进行签到逻辑判断。

本课题的设计加了当签到成功时亮一个绿灯，当签到失败的时候亮一个红灯。

为解决想要查看考勤记录的情况，需要有一个网站能查看使用者的考勤记录，网站上理应还有补签的功能，这样就不用手写说明书，也要设计审批的功能，当用户补签时，可选择通过或者拒绝。所有的web端都是由python及其web开发框架Django实现的。

**1.3 前人成果**

[人脸识别](http://product.pconline.com.cn/itbk/digital/dc/1112/2620203.html)特指利用分析比较人脸视觉特征信息进行身份鉴别的计算机技术。人脸识别是一项热门的计算机技术研究领域，它属于生物特征识别技术，是对生物体(一般特指人)本身的生物特征来区分生物体个体。目前人脸识别技术已经相当成熟，并且有许多相关的sdk来帮助不涉及人脸识别算法学习的开发者来快速的使用与人脸识别相关的服务。

**1.4 论文结构**

本论文各章节安排如下：

第一章：绪论：介绍了各课题的背景；

第二章：技术简介：介绍整个系统中所涉及的技术；

第三章：功能分析：简述所有的功能；

第四章：硬件设计：描述硬件模块及程序设计；

第五章：WEB设计：描述网页端考勤记录查看及程序设计；

第六章：总结：描述完成系统过程中遇到的问题及收获。

第二章 技术简介

**2.1 Python & Django**

**2.1.1 Python简介**

如果说当前那种语言最流行，无疑还是Sun Microsystems公司于1995年5月推出的高级程序设计语言Java，但是要说在人工智能领域中哪个语言最流行，那一定是Python。

Python是一种上手非常简单的语言，它非常适合初学者学习。Python除了相对C等高级语言运算速度慢一点以外，几乎再没有任何缺点。开发人员几乎可以用Python做任何事情，因为Python具有非常丰富的第三方库，比如进行矩阵计算可以用numpy，进行深度学习可以用sktlearn，等等。

Python同时也支持面向对象编程，这意味着开发人员的代码重复利用率更高，程序更容易维护。

**2.1.2 Django简介**

Django是一个完全由python编写的web应用框架，它开发的宗旨就是帮助开发者快速开始一个web应用。Django支持MVC架构，这使得模块之间耦合度更低，代码冗余率更低。Django还具有防止跨站访问的功能，这使得用Django搭建的Web应用更加安全。

**2.2 OpenCv for Python**

**2.2.1 OpenCv简介**

OpenCv是一个基于BSD许可发行的跨平台计算机视觉库，它可以运行在Linux、Windows、Android和Mac OS等操作系统中。

OpenCv有很多版本，包括了C++、Java和Python等。本课题使用的就是其Python版本——OpenCv-Python。

OpenCV-Python是OpenCV的python的API接口，它拥有OpenCV C++ API的功能，同时也拥有Python语言的特性。虽然Python语言运行起来比C++语言慢，但是Python语言具有简单、短小、学习快的特点，同时Python也很方便调用C++开发的组件，这样高性能要求的功能可以使用C++来实现。这样一来，开发者就可以既运行像C++一样快的代码，又可以使用简单易用Python语言来开发软件的功能，这就是为什么要实现OpenCV-Python的接口的原因。同时OpenCV-Python也实现与Numpy库的接口规范，这样很方便在Python使用Numpy，比如可以把Numpy的数据结构转给OpenCV,也可以把OpenCV的数据结构转给Numpy，另外也能与SciPY、Matplotlib协同使用，这样可以使用的范围就更加广泛了，所以OpenCV-Python是一个很适合来开发视觉原型、进行视觉的实验等等工作。

**2.2.2 Raspberry Pi搭建python-OpenCV**

本课题没有采用python2.7的版本，而是使用了python3.5的版本。尽管现在仍然有许多第三方库对python3的支持不是很好，但是python3.5的优点已经足以弥补这些缺憾了。

Raspberry Pi安装opencv采用的是编译源码进行安装，这是一个非常艰巨并且也非常耗时的操作。

首先更新树莓派，确保树莓派的更新库是最新的，运行如图2.1的命令：



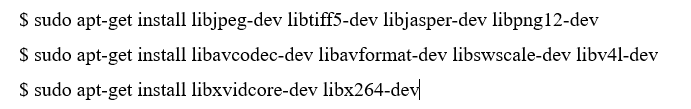
**图2.1 更新树莓派软件**

此时，树莓派的软件更新库已经是最新的了。由于本系统采用源码安装，所以需要为树莓派安装编译环境。在终端中执行图2.2的命令。



**图2.2 安装编译环境**

OpenCV是做图像处理的，所以需要为它安装一些图片I/O包和视频支持包，这些库允许开发者能灵活的使用OpenCV，运行图2.3命令。



**图2.3 安装图像和视频I/O包**

OpenCV内置了highgui 库，这个库是为了在显示屏中显示一个图像并组建一个GUI，编译highgui库，需要安装GTK开发库，执行图2.4命令。



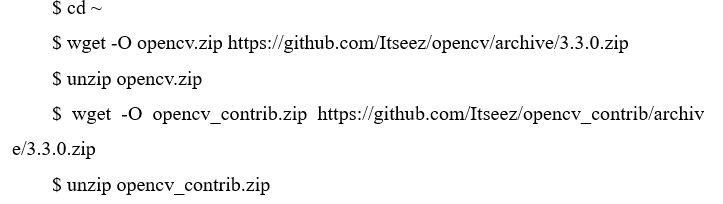
**图2.4 安装图像显示支持**

OpenCV中有许多矩阵相关的库，可以安装一些额外的依赖来进行优化。执行图2.5命令。



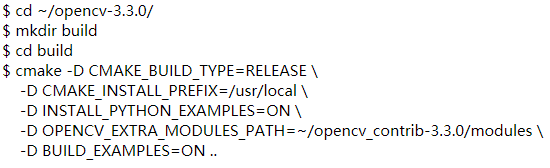
**图2.5 优化OpenCV的库**

准备工作做完了，现在可以下载OpenCV源码了，本课题采用的是3.3.0的版本，可以从github中将源码下载下来，同时为了完整安装opencv，系统也会下载opencv\_contrib存储库。下载解压代码如图2.6所示。



**图2.6 下载OpenCV源码**

源码下载好了，现在就可以进行编译安装了。在opencv目录创建一个新的文件夹，在进行编译之前需要cmake进行环境依赖的检测，如图2.7所示。



**图2.7 编译OpenCV源码**

没有问题之后，使用图2.8所示命令完成编译和安装。



**图2.8编译OpenCV源码**

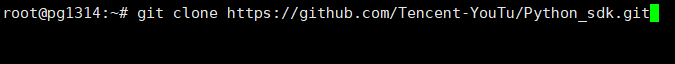
接下来就是长达五六个小时的等待了。

**2.3 腾讯优图人脸识别开发库——Python-SDK**

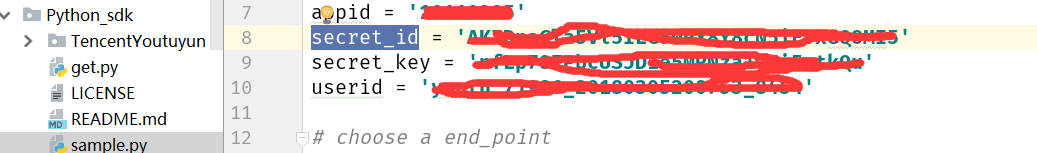
腾讯优图人脸识别开发库可以帮助一个非深度学习研究者快速开始一些人脸识别项目。

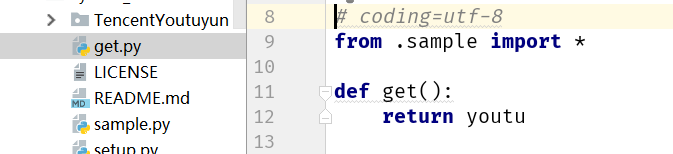
**2.3.1 准备步骤**

1、下载Python-SDK，输入图2.9所示命令。

**图2.9 下载Python-SDK**

2、在优图主页注册并获取到appid、secret\_id、secret\_key、userid，然后在Python\_sdk下面的sample.py填写获得的验证信息，如图2.10所示。

**图2.10 填写验证**

3、在Python\_sdk下面新建一个get.py文件，并编写代码如图2.11所示。****

**图2.11 编写get.py文件**

4、在开发人员想要使用Python\_sdk的文件中，只要导入get.py就能使用了。

**2.3.2 常用API**

1、人脸识别接口命令如图2.12所示。



**图2.12 人脸识别api**

参数中的group\_id是指人脸组别id，本系统可以为不同属性的组别同时命名不同的id，比如自动1403班可以为一个组，每个班都可以作为单独的一个组别，在人脸识别中，指定特定的组别，系统只会在这个组别中进行人脸匹配，而不是整个人脸库。

2、增加人脸接口命令如图2.13所示。



**图2.13 增加人脸的api**

参数列表中的person\_id是指每一个人脸的id信息，本系统使用的是学生的学号作为person\_id；images是人脸照片的url或者是人脸照片路径，URL其实就是一个照片文件打开后进过base64编码产生的编码，所以在Python-SDK内部，如果开发者传进去的是路径，内部程序也是使用python的open方法将照片打开，并使用base64编码将其编码后，再进行后续的操作。参数中data\_type为0或1选择，代表是路径还是URL。

3、增加个体接口命令如图2.14所示。



**图2.14 增加个体的api**

在进行添加一个人脸之前，需要新建一个个体，这个有属于的组别，有本身的名字，个体建好之后才能在这个个体中添加自己的人脸信息，每个人可以添加10张人脸，但是不能添加相似度过高的人脸，为了提高人脸识别准确率，最好添加10张不同场景的人脸。

**2.4 Sqlite**

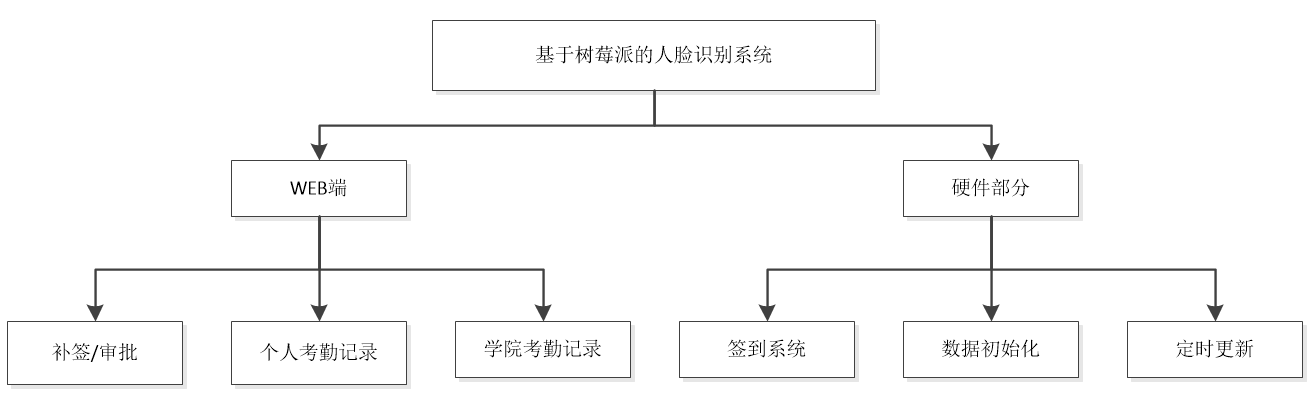
SQLite是一个进程库，实现了一个[自包含的](https://www.sqlite.org/selfcontained.html)、[无服务器](https://www.sqlite.org/serverless.html)、[零配置](https://www.sqlite.org/zeroconf.html)、[事务性](https://www.sqlite.org/transactional.html)的SQL数据库引擎。SQLite的代码是开源的，因此可以用于任何目的、商业或私人用途。SQLite是世界上[部署最广泛的](https://www.sqlite.org/mostdeployed.html)数据库，其应用程序数量超过了官方可以计算的数量，其中包括几个[备受瞩目的项目。](https://www.sqlite.org/famous.html)SQLite是一个嵌入式SQL数据库引擎。与大多数其他SQL数据库不同，SQLite没有单独的服务器进程。SQLite直接读取和写入普通磁盘文件。包含多个表，索引，触发器和视图的完整SQL数据库包含在单个磁盘文件中。数据库[文件格式](https://www.sqlite.org/fileformat2.html)是跨平台的 - 您可以在32位和64位系统之间或者在[大端](http://en.wikipedia.org/wiki/Endianness)和[小端](http://en.wikipedia.org/wiki/Endianness)体系结构之间自由复制数据库。这些特性使SQLite成为[应用程序文件格式](https://www.sqlite.org/appfileformat.html)的热门选择。

同时，Sqlite支持标准SQL语句。

**第三章 设计方案**

**3.1 基于树莓派的人脸识别签到系统架构**

人脸识别签到系统包括了签到结果展示部分和硬件签到部分。签到结果展示又分了个人签到记录的查看、全部人员签到记录查看（仅管理员有权限）、异常打卡补签以及管理员审批等部分。硬件部分则涉及签到、定时更新数据（未签到的人加进数据库）、以及一开始的数据初始化。系统架构如图3.1所示。



**图3.1 人脸识别系统架构图**

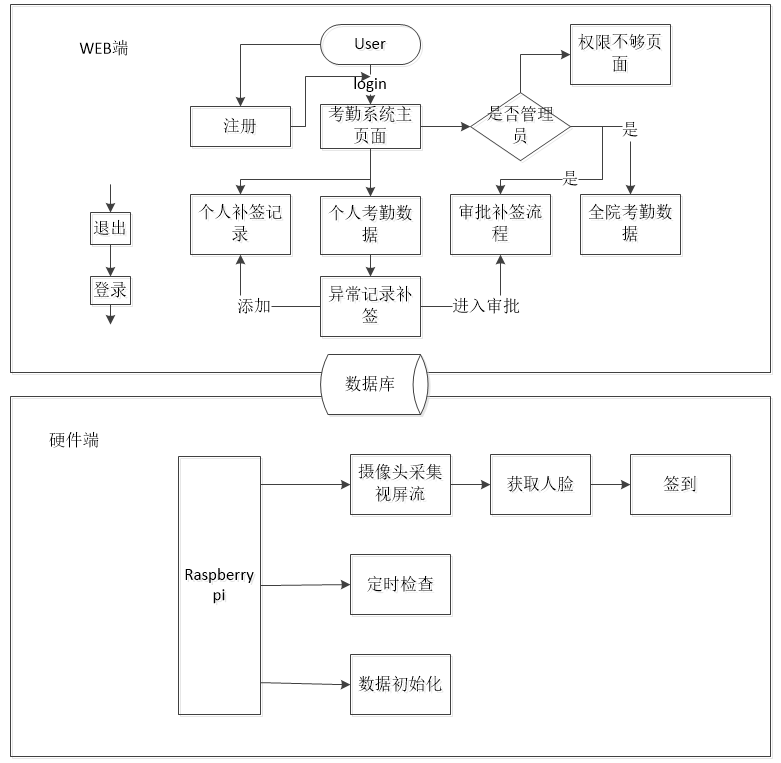
从整个系统来看，本课题的功能并不算复杂，主要分为两大部分，然后再细分了许多功能，每个功能有独立的逻辑处理，功能与功能之间的通信只能通过预留的API接口。这样做的目的是为了让代码利用率更高。

**3.2 人脸识别考勤系统的功能流程**

在系统运行之前，首先要做的是将一些数据进行初始化，比如学生信息的导入、课表的导入，然后才能启动WEB服务、考勤程序和定时数据更新程序。本课题采用三个进程来启动这些程序，三个进程互不干扰的执行，每个进程只负责自己的任务，考勤程序只负责对刷脸的学生进行考勤统计，定时数据更新则是在几个特点的时间扫描数据库，看看有没有旷课的学生，而WEB服务则只是负责考勤记录的查看与异常考勤的补签流的操作。

尽管每个进程只负责自己的任务，但是同处于一个系统下，它们之间还是有通信的。

考勤程序会将考勤结果持久化到数据库中，它只有插入操作；定时更新程序会从数据库中查询出没有该进行签到但是没有签到的学生，将他们的考勤设为旷课，定时更新程序有查询插入操作；而WEB端，也是从数据库中查询考勤记录，并展示在页面中，补签等操作也是与数据库打交道，即WEB端则涉及数据库的查改增操作。所以几个进程的能够进行通信，完全依赖于使用了同一个数据库。当然，本系统并不需要担心由多进程引发的安全问题，这得益于sqlite的数据库操作是原子操作。人脸考勤系统的整体功能流程如图3.2所示。



**图3.2 人脸识别考勤系统的功能流程**

三个进程执行三个程序是可行的，首先定时更新函数，它是利用定时器做的，只有到了设定的时间才会执行更新函数，其他时候仅仅只是做一个无限循环，既没有连接到数据库也没有进行网络资源的占用，所以对CPU的消耗是可观的；再说签到功能程序，仅仅是占用了Raspberry Pi的视频设备，而且为了节省网络资源，系统还特意设计了按键，只有按键按下的时候，才截取一帧照片，进行签到逻辑的运算；而WEB端的设计将所有的静态资源：js文件、css文件和图片等文件都放在了静态文件夹下，这使得在客户端访问的时候，除了第一次访问需要下载静态资源，后面的访问将使用缓存，极大地减少了服务器的压力，提高了访问速度。

**3.3 设计原则**

**本课题设计的第一原则为可行性原则：**所有的设计都是具有理论基础的，都是经过大量调试验证的，只有确定可行了，才会继续下一步的操作。这是整个系统的开发中最重要、最核心的原则。

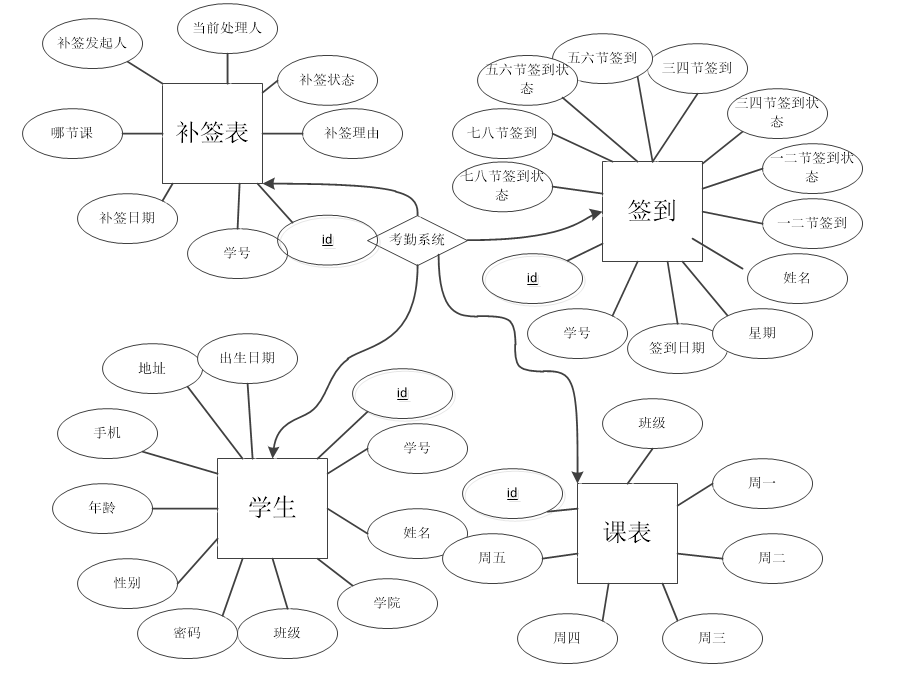
**分块设计原则：**本课题采用切割划分出许多不同的功能块，在确保第一原则的前提下，划分出尽可能详细的，功能单一的块，这样能使得代码的可重用率极大地提升了。

**低能耗原则：**对于连接资源，在逻辑结束后，一定要释放，避免浪费资源。系统采取的主控设备也是能耗非常低的设备。

**第四章 数据库设计**

**4.1 数据库设计ER图**

本课题一共设计了四张表存在sqlite数据库中，包括了T\_XUPT\_PERSON（学生信息表）、T\_XUPT\_TIMETABLE（课表）、T\_XUPT\_ATTEND（签到表）、T\_XUPT\_REATTEND（补签表）。表的设计遵循数据表设计的三大范式，数据表的ER图见图4.1所示。

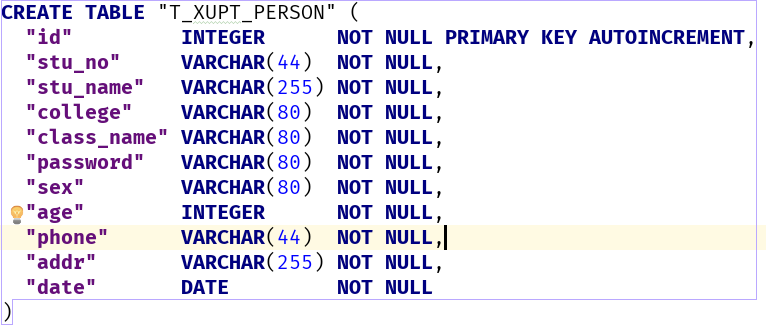


**图4.1 人脸识别考勤系统ER图设计**

这就是整个人脸识别考勤系统的ER图设计，接下来几个小节将详细描述每个表的具体设计与实现。

**4.2 学生表的设计**

学生表在数据库中的表名是T\_XUPT\_PERSON，一共包含11个字段，其建表语句见图4.2。

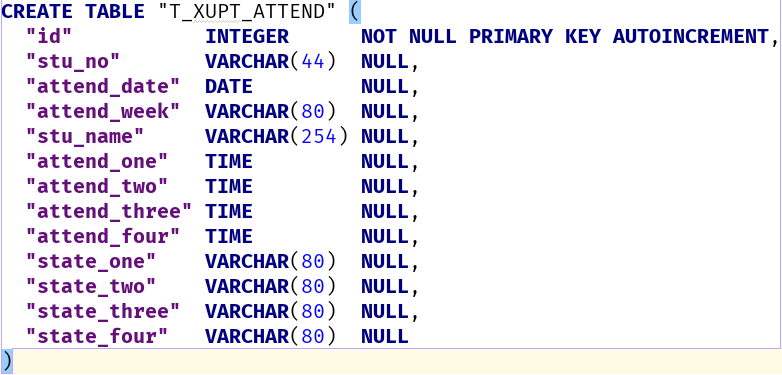


**图4.2 创建T\_XUPT\_PERSON**

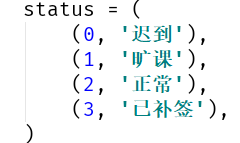
学生表选取一个自增的INTEGER类型作为表的主键字段，其中密码字段存的其实是学生的身份证号。

**4.3 签到表的设计**

签到表的设计参照了西安邮电大学一天八节课的情况，八节课又分为早上一二节、早上三四节、下午五六节、下午七八节，所以设计了8个字段分别表示这四个不同时间段的考勤时间和考勤状态，考勤状态分为4种，见图4.4。签到表T\_XUPT\_ATTEND一共有13个字段，除了上述8个，还包括了签到的星期、日期的字段。建表语句见图4.3所示。



**图4.3 创建T\_XUPT\_ATTEND**

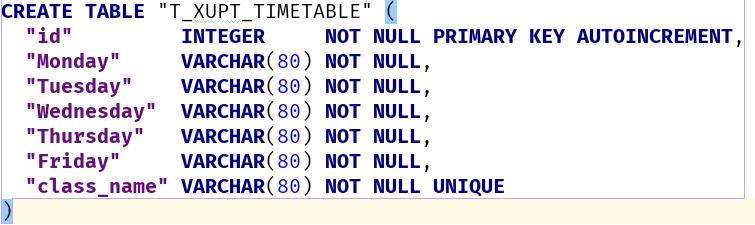


**图4.4 四种签到状态**

采用的无意义的主键列，仅仅是为了方便索引，其它字段允许为空，意味着增大了容错率。

**4.4 上课表的设计**

上课表（T\_XUPT\_TIMETABLE）的设计很简单，使用了无意义的自增属性作为id，然后有不可重复的班级字段、周一到周五这几个字段。建表语句如图4.5。

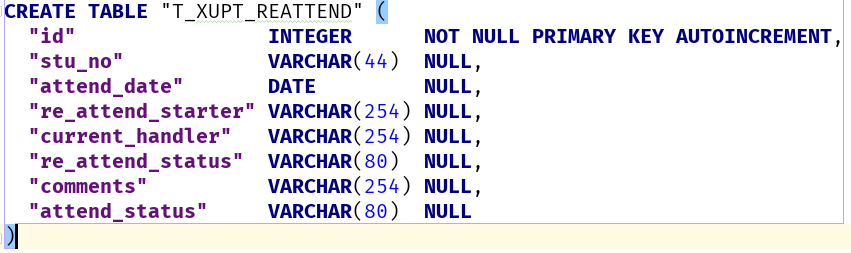


**图4.5 创建T\_XUPT\_TIMETABLE**

本系统采用长度为8的数字字符串来表示一个班级的一天的课表，比如周三早上一二节有课和下午七八节有课，则Wednesday字段存的是11000011字符串。

**4.5 补签表的设计**

补签表（T\_XUPT\_REATTEND）的设计是由于有些学生可能迟到、或者忘记打卡而导致的打卡异常，用来记录补签的表。建表语句见图4.6。



**图4.6 创建T\_XUPT\_REATTEND**

补签状态分三种：处理中、审批通过、审批不通过，分别用数字1、2、3表示。字段attend\_status存的是0、1、2、3这几个数字中的一个，分别代表的是早上一二节、早上三四节、下午五六节、下午七八节课。

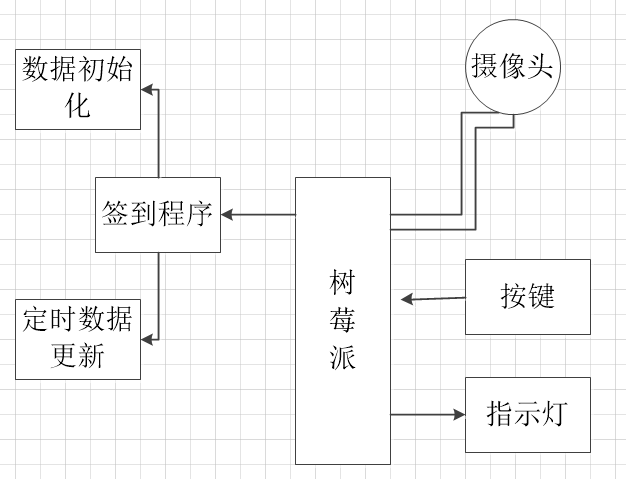
**4.6 数据库设计总结**

表的设计是按照范式法则并结合实际情况进行设计的，每个字段都有必要存在而不是赘余，数据库E-R图的设计经历了多次的修改重建，阅读了大量的数据库文献，保证了数据表的优化。

**第五章 硬件设计与实现**

**5.1 硬件设计架构**

在硬件部分，本课题设计了三个功能点：数据初始化、签到、定时的数据库更新，程序的主要入口在签到功能里面，签到功能程序开始，创建了两个进程来启动数据初始化和定时更新，其中数据初始化执行完进程就结束，定时更新则在无限循环，设计架构图见图5.1所示。

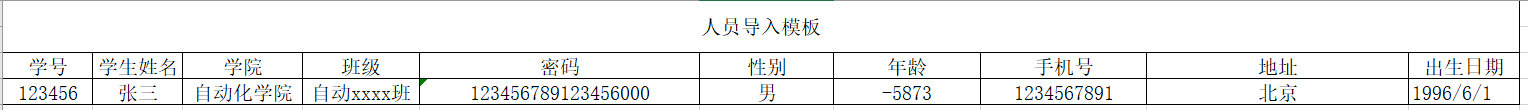


**图5.1 硬件部分架构**

指示灯模块是为了显示一个签到是否成功的模块。

**5.2 数据初始化**

本课题的设计要求管理员做一个excel表格，表格里面要有学生的信息，这些信息要求和数据库中的T\_XUPT\_PERSON所规定的一致，放在sheet0中，模板如图5.2。在sheet1中要求管理员填写每个班级的课表，有课填写1，否则填写0，模板见图 5.3所示。

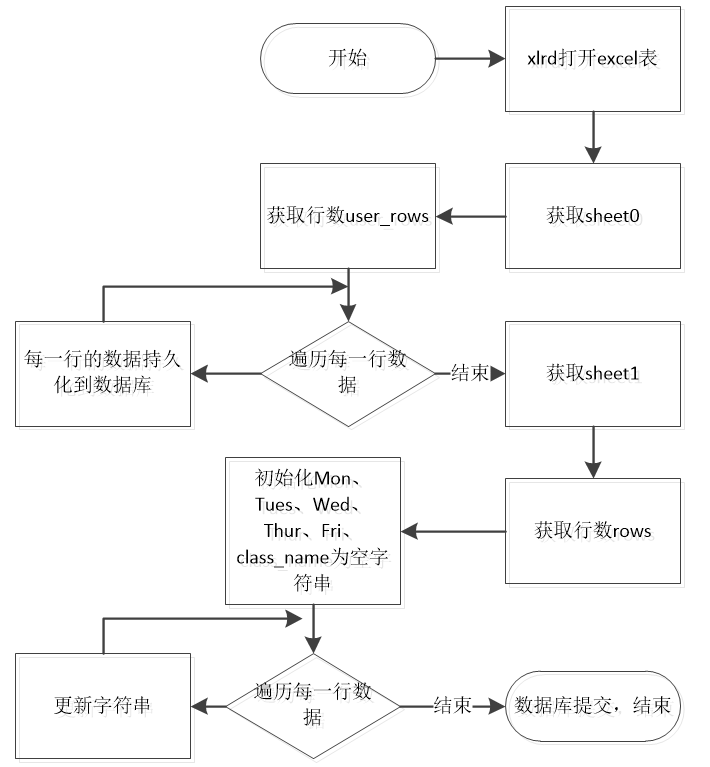


**图5.2 人员导入模板**



**图5.3 课表模板**

当签到程序运行时，会启动一个进程调用初始化数据程序。数据初始化通过python的xlrd库来完成，xlrd是python读取excel表格的一个库。数据初始化流程图见图5.4所示。



**图5.4 数据初始化流程**

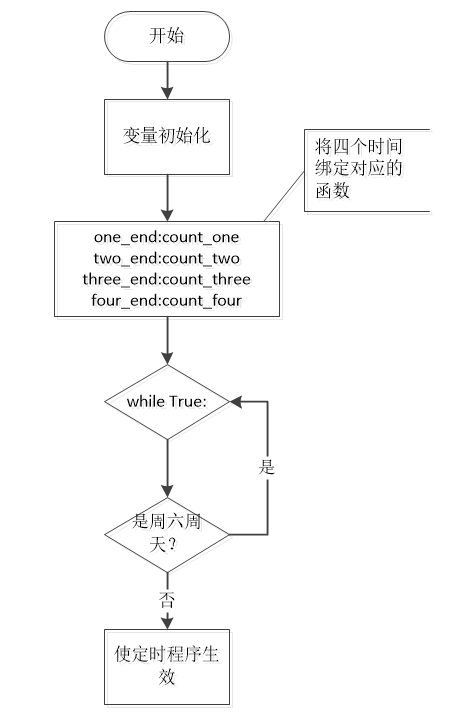
在流程图中，前面的循环是将学生信息插入读取到后插入到数据库中，后面的循环是遍历每列数据，因为模板里面存的是01，所以数据库存的是一天8节课的信息，需要全部遍历完成才能获取到8位长度的01课表串。

**5.3 定时更新数据库**

定时更新数据库程序是非常有必要的，因为在上课前20分钟到下课这个时间段是可以正常刷脸签到的，但是存在这个时间段没有签到的可能性，那么这些人在签到表里是没有记录的，同时他们也属于旷课存在的。

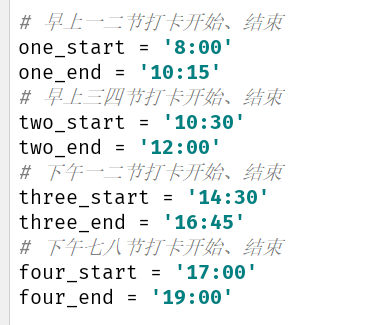
设计这个程序的目的就是排除掉签到的同学，把需要签到但是没有签到的同学将他们的考勤状态设置为旷课。

定时更新程序使用了python的schedule模块，schedule模块可以通过设置一个时间和指定一个函数，当进程还在运行并且时间到达时，系统就会运行那个函数。本课题一共设置了四个函数，分别会在早上一二节放学、早上三四节放学、下午五六节放学和下午七八节放学执行。选择这几个时间是有道理的，因为在放学后要是学生还没有进行签到，签到系统就会拒绝学生的签到了，那么学生一定是旷课状态。定时更新程序流程见图5.5所示。



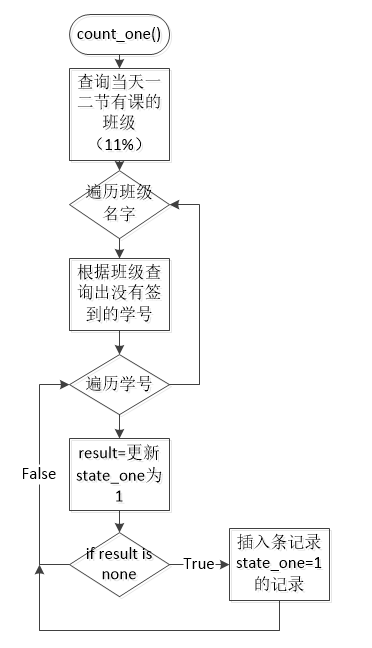
**图5.5 定时更新数据库流程**

由于周六周天不需要上课，所有在程序设计时候加了判断，如果不是周六周天才使得定时器程序生效。具体逻辑业务四个count\_\*决定，它们通过schedule与\*\_end时间进行绑定。本课题对于上下课时间做出图5.6的规定：



**图5.6 上课时间安排**

每次下课放学都会执行相应的函数。一开始根据课表获取当前时间刚下课的班级，然后对这个班级进行遍历，获取当前班级中除了已经签到过的学生的学号，再对这个学号进行遍历，然后对这些人更新数据库，如果他们一天刷脸签到，则在数据库中新增一条当前时间的考勤状态为旷课的记录，如果在前面几节课中已经签到了，则是对那条记录进行更新操作，设置当前时间的考勤状态为旷课。早上一二节统计逻辑流程图如图5.7所示。



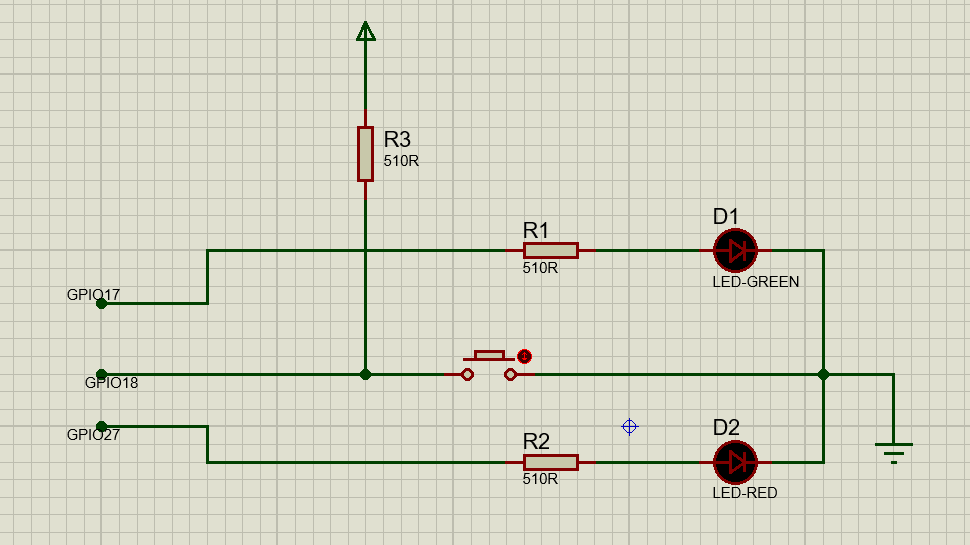
**图5.7 一二节下课数据更新**

剩三个函数的逻辑是一样的，只不过查询班级的时候模糊查询的关键字和更新或者插入的字段不一样，比如三四节课就不是’11%’，而是’??11%’，问号代表占位符，一个问号代表一个任意字符，%代表任意个的任意字符。

**5.4 签到**

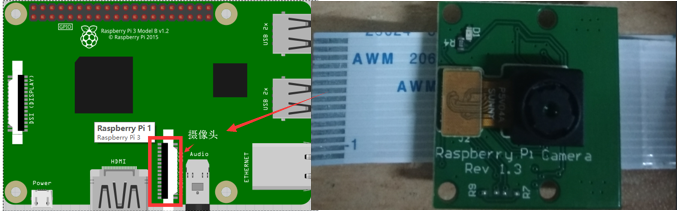
**5.4.1 电路设计**

签到模块是整个课题最重要的部分，也是逻辑最复杂的部分。本课题采用树莓派加一个摄像头、一个按键、和两个led完成硬件部分的操作。电路设计如图5.8所示。



**图5.8 电路设计图**

树莓派和摄像头安装见图5.9：



**图5.9 树莓派和摄像头安装**

摄像头用来拍摄视屏流，为了避免一直进行人脸识别的请求，本课题特意增加了一个按键，当只有按键被按下时才进行人脸识别，如果签到成功，则绿灯会亮，否则红灯亮。

**5.4.2 程序实现**

程序使用OpenCV捕获本机摄像头设备，使用默认的人脸检测器” haarcascade\_frontalface\_default.xml”文件作为检测器，腾讯优图的python-sdk为人脸识别工具，整个签到功能架构见图5.10。