目录

[1 项目需求分析](#_Toc5301_WPSOffice_Level1) [6](#_Toc5301_WPSOffice_Level1)

[1.1 智能家居现状](#_Toc14157_WPSOffice_Level2) [6](#_Toc14157_WPSOffice_Level2)

[1.2树莓派在智能家居中的分析](#_Toc8139_WPSOffice_Level2) [1](#_Toc8139_WPSOffice_Level2)

[2 项目整体结构设计](#_Toc14157_WPSOffice_Level1) [2](#_Toc14157_WPSOffice_Level1)

[2.1项目功能简介](#_Toc32580_WPSOffice_Level2) [2](#_Toc32580_WPSOffice_Level2)

[2.2 项目设备选型](#_Toc32111_WPSOffice_Level2) [2](#_Toc32111_WPSOffice_Level2)

[2.3 系统组成](#_Toc14040_WPSOffice_Level2) [5](#_Toc14040_WPSOffice_Level2)

[3 项目的设计与实现](#_Toc8139_WPSOffice_Level1) [5](#_Toc8139_WPSOffice_Level1)

[3.1 控制核心](#_Toc4554_WPSOffice_Level2) [5](#_Toc4554_WPSOffice_Level2)

[3.1.1 控制核心部分介绍](#_Toc14157_WPSOffice_Level3) [5](#_Toc14157_WPSOffice_Level3)

[3.1.2控制核心部分接线说明](#_Toc8139_WPSOffice_Level3) [6](#_Toc8139_WPSOffice_Level3)

[3.2 室内智能调温](#_Toc17057_WPSOffice_Level2) [6](#_Toc17057_WPSOffice_Level2)

[3.3 家庭智能预警](#_Toc9607_WPSOffice_Level2) [10](#_Toc9607_WPSOffice_Level2)

[3.3.1 七牛云对象存储介绍](#_Toc32580_WPSOffice_Level3) [10](#_Toc32580_WPSOffice_Level3)

[3.3.2 Python结合Twilio发送短信](#_Toc32111_WPSOffice_Level3) [12](#_Toc32111_WPSOffice_Level3)

[3.3.2 智能预警的实现](#_Toc14040_WPSOffice_Level3) [14](#_Toc14040_WPSOffice_Level3)

[3.4 实时监控](#_Toc11586_WPSOffice_Level2) [16](#_Toc11586_WPSOffice_Level2)

[3.4.1 实时视频](#_Toc4554_WPSOffice_Level3) [16](#_Toc4554_WPSOffice_Level3)

[3.4.2 Tomcat服务器](#_Toc17057_WPSOffice_Level3) [18](#_Toc17057_WPSOffice_Level3)

[3.4.3 实时监控网页的实现](#_Toc9607_WPSOffice_Level3) [18](#_Toc9607_WPSOffice_Level3)

[3.4.4 内网穿透](#_Toc11586_WPSOffice_Level3) [21](#_Toc11586_WPSOffice_Level3)

[3.5 人脸识别门锁](#_Toc4801_WPSOffice_Level2) [22](#_Toc4801_WPSOffice_Level2)

[3.5.1 人脸检测](#_Toc4801_WPSOffice_Level3) [24](#_Toc4801_WPSOffice_Level3)

[3.5.2 人脸裁剪](#_Toc612_WPSOffice_Level3) [25](#_Toc612_WPSOffice_Level3)

[3.5.3 人脸识别](#_Toc20615_WPSOffice_Level3) [27](#_Toc20615_WPSOffice_Level3)

[4 项目总结](#_Toc24905_WPSOffice_Level1) [30](#_Toc24905_WPSOffice_Level1)

[5 参考文献](#_Toc26003_WPSOffice_Level1) [30](#_Toc26003_WPSOffice_Level1)

# 1 项目需求分析

## 1.1 智能家居现状

智能家居作为一个新生产业，处于一个导入期与成长期的临界点，市场消费观念还未形成，但随着智能家居市场推广普及的进一步落实，培育起消费者的使用习惯，智能家居市场的消费潜力必然是巨大的，产业前景光明。 正因为如此，国内优秀的智能家居生产企业愈来愈重视对行业市场的研究，特别是对企业发展环境和客户需求趋势变化的深入研究，一大批国内优秀的智能家居品牌迅速崛起，逐渐成为智能家居产业中的翘楚!

智能家居领域在国内仍存在诸多瓶颈，如：缺乏杀手级应用；标准不统一，包括网络传输标准的不统一以及家电产品之间无法融合；仍然存在技术性问题(包括人机交互技术、云计算、大数据等)、缺乏各家电之间的互联互通；产品价格太高；企业各自为政；产品功能相对单一，产品同质化很严重，功能雷同，创新不足；产品实用性差、用户体验差；安全隐私等问题。

在中国当前的智能家居市场上，已经有一大批国内互联网公司，先后推出了机顶盒、路由器、WIFI插座、智能开关等智能家居产品。但纵观市场，无论是智能化单品，还是构筑智能家居平台和系统，大多数的所谓智能化产品，还停留在“为智能化而智能化”、“为了连接而连接”的层面，一些功能很明显是为了傍上“智能产品”之名而设计，没有找正用户真正的“痛点”，营销噱头远远大于实用性，加上智能家居产品应用场景较为固定，这也导致真正被消费者接受和使用的少之又少。要解决智能家居普及的两个问题，一个是如何让每个家电智能化，二是如何让每个家电互联互通，形成更智能的场景。

当然，不可否认的是，智能家居将成为家电企业下一个竞争的新战场。在智能家居这个新硬件领域，真正可以实现产品市场规模化发展和引领市场潮流的或许还是传统家电企业。目前传统家电行业也正在逐步深入拥抱互联网，产业的繁荣与发展最终也将为用户带来更好的体验。

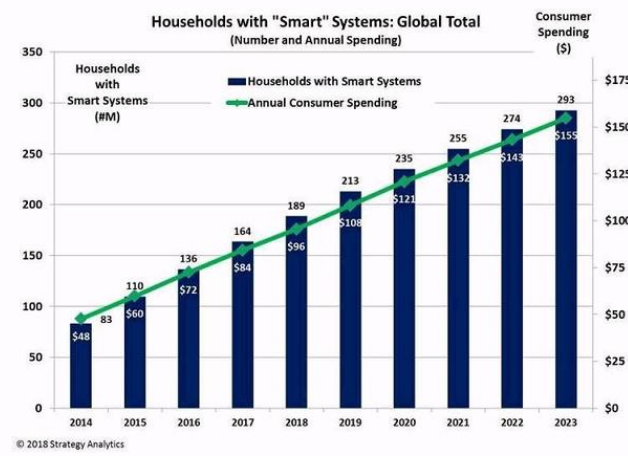


图1-1 智能家居市场规模预测

## 1.2树莓派在智能家居中的分析

家居智能化是现代化生活中的重要一环，在新时代的发挥中，智能家居决定着日常生活的便捷程度。智能家居系统通过先进的计算机技术、 网络通讯技术、智能云端控制、综合布线技术、医疗电子技术融 合了个人的需求达到为人们服务的目的。较好的智能家居系统有利于人们便捷舒适现代化生活的实现。能否开发出廉价实用且可普及的智能家居系统，对人们拥有更加美好的生活是 至关重要的，因此对家居智能化的研究就显得尤为重要。

树莓派其本质是一种体型极小的电脑，有着很丰富实用的功能，这种卡片式电脑也正在逐渐运用于智能家居的各个领域。本研究课题是利用了树莓派的特点，将其作为一个中继点，运用远程监控技术，实现智能家居的远程监控系统。 可以说本系统进行了以下的创新，它是一种基于树莓派的智能家居远程监控系统。把 Linux 操作系统在树莓派中建立起来，并使用智能家居控制程序，构建主控系统，实现温度 远程监控，实时收集家居温度信息，并向家居主人做出反馈。 使用树莓派完成这一设计的优势就在于它小但是精炼，有着上述的诸多功能可以开发，在实际中有着应用可能性。

总的来说，就是制作一个树莓派担当中央控制器，以各种传感器器件采集外部信息数据，然后通过外部手段进行远程的数据传输，打造一个智能家居系统远程监控系统，使日常的生活更加快乐、方便、舒适。

2 项目整体结构设计

## 2.1项目功能简介

基于树莓派的智能家居监控系统是以树莓派作为控制终端，通过多种传感器采集信息，采用相应的硬件连接和软件支持，使用Python编程，可实现室内智能调温，家庭的智能预警，实时监控以及人脸识别门锁等功能于一体的智能家居系统。

1.室内智能调温：根据DHT11温湿度传感器测的室内温湿度信息，L298N与电机风扇相连并自动根据温湿度信息开关，与此同时将温湿度信息记录在MySQL数据库中进行保存。

2.家庭智能预警：当家中无人时，人体红外传感器模块监测是否有人活动，当监测到有人体活动时，蜂鸣器报警示意，与此树莓派摄像头抓拍照片并将照片传到七牛云服务器中，并短信通知用户。

3.实时监控：将家庭中的温湿度以及家庭闯入信息显示到网页，并通过mjpg-streamer软件将局域网内摄像头的实时监控情况呈现在网页中。

4.人脸识别门锁：当触摸传感器感应到有人触碰时，树莓派摄像头抓拍照片，并启动人脸识别功能，通过Face++云平台实现人脸图像的识别处理，根据人脸识别判断的结果控制电磁锁的开关。

## 2.2 项目设备选型

|  |  |  |  |
| --- | --- | --- | --- |
| 元件名称 | 参考规格 | 数量 | 实物参考图 |
| Raspberry Pi | 3代B型 | 一块 |  |
| 网线 | 多种规格 | 1条 |  |
| USB连接线 | 多种规格 | 1条 |  |
| Micro-SD | 16GB 80MB/s TF(Micro SD)Class10 UHS-I | 一个 |  |
| 温湿度传感器 | DHT11 | 1个 |  |
| 人体红外传感器模块 | HC-SR501 | 1个 |  |
| 蜂鸣器模块 | 无源蜂鸣器 | 1个 |  |
| 摄像头 | 树莓派摄像头 | 1个 | IMG_20180627_082938 |
| 步进电机 | Uln2003驱动板+5V步进电机 | 一个 |  |
| 杜邦线 | 公对公  母对母 | 数根 | timg |
| 触摸传感器 | 多种规格 | 1个 |  |
| 磁开关传感器 |  | 1个 |  |

## 2.3 系统组成

系统主要由控制终端、感应单元、受控单元及云服务四个部分组成：控制终端为树莓派；感应单元由各种传感器组成；受控单元包括各种家具器件及继电器 ；云服务主要指 Face++ 云平台、七牛云平台。

感应单元将传感器采集的信息数据传递给控制终端树莓派，其中，部分信息直接在本地处理，部分信息上传至云端，云端对图像信息进行处理后，将结果返回给树莓派。根据处 理后的结果，树莓派发出相应控制指令到受控单元，控制相 连的各器件。系统整体的信息数据流向方式如图2-1所示。

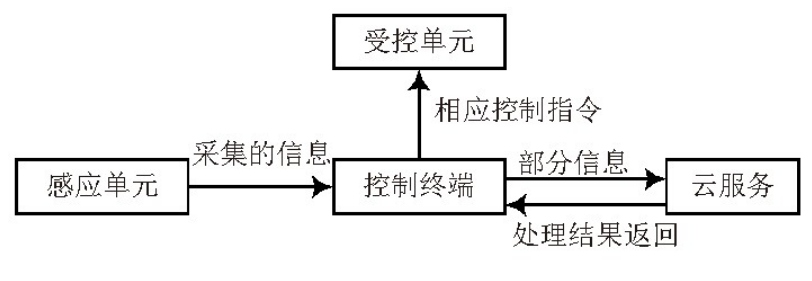


图2-1 系统的信息数据流向图

# 3 项目的设计与实现

## 3.1 控制核心

### 3.1.1 控制核心部分介绍

系统对数据传输能力要求较高，在视频数据与图像数据上需要保证流畅 及稳定性，普通的单片微控制器（如 Arduino）难以实现。考虑到成本与功能 需求，该系统采用树莓派 Raspberry Pi 3b 作为控制终端。

树 莓 派 Raspberry Pi 3bBroadcom BCM2837 为片上系统，其运行频率为 1.2 GHz，拥有四核 ARM 处理器， 1 GB RAM 存储器，与 40 个 GPIO 引脚，CSI 摄像头接口，HDMI 接口，4 个 USB 2.0 端口，10/100 以太网端口，RAC av 端口， USB 电源接口，DSI 显示连接器，搭载蓝牙和 WiFi 模块。

树莓派的 GPIO 引脚定义如图3-1、3-2所示。GPIO 引脚的编号方式有三种，分别为基于引脚的物理位置进行编号、C语言GPIO库wiringPi约定的编号方式、Python语言GPIO库RPo.GPIO约定的编号方式。

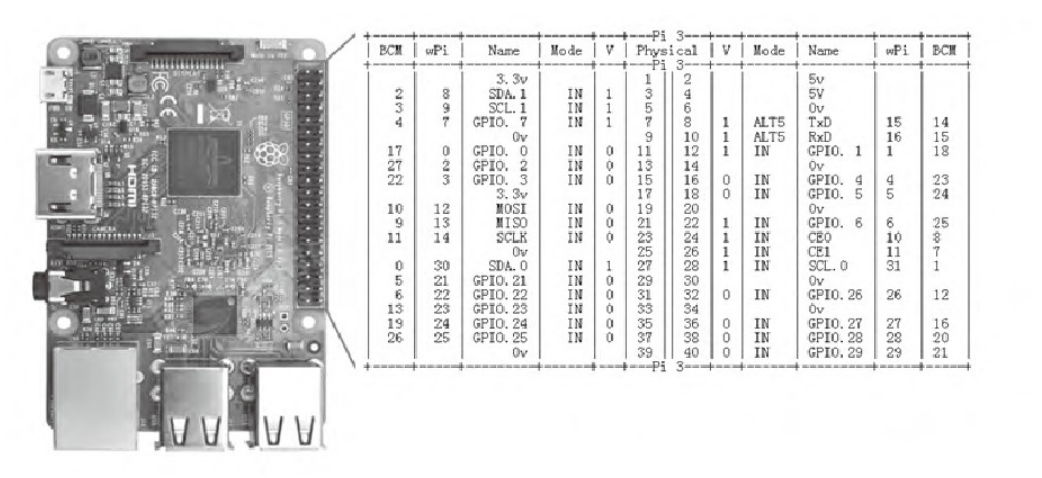


图3-1 Raspberry Pi 3b GPIO 引脚定义

### **3.1.2控制核心部分接线说明**

（1）DHT11温湿度传感器：+5V接树莓派+5V，GND接地，IN1口接7口。

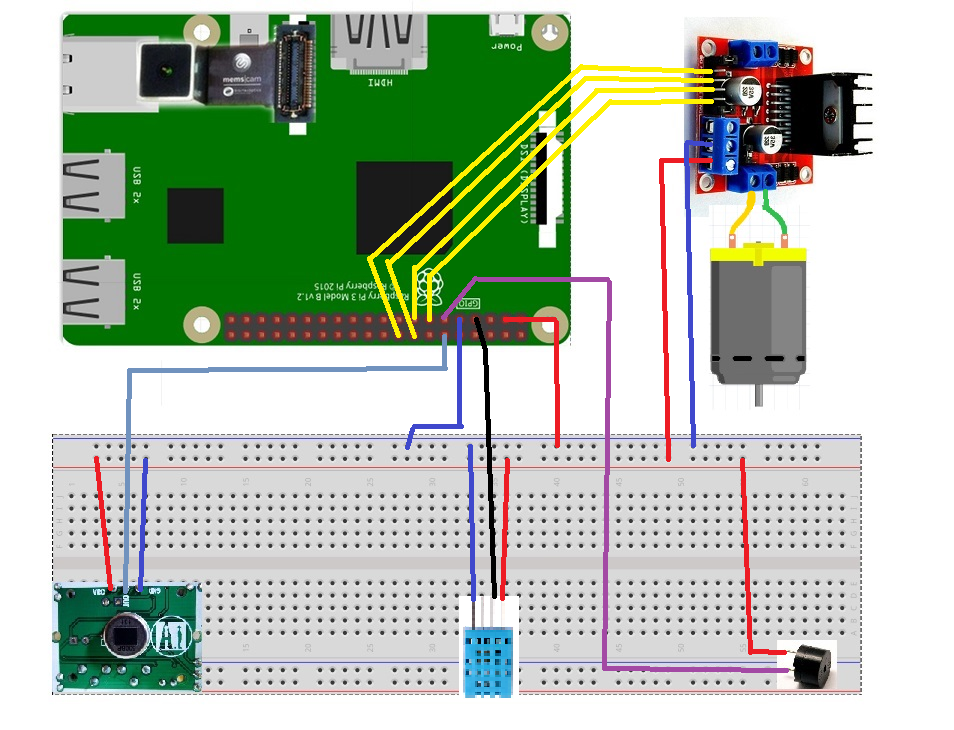
（2）人体传感器HC-SR501：数据引脚接12口，+5V接树莓派+5V，GND接地。

（3）蜂鸣器：+5V接树莓派+5V，另一根线接11口。

（4）电机驱动L298N：INT1、INT2、INT3、INT4分别接树莓派的13口、15口、16口、18口。+5V接树莓派5V，GND接地。OUT3、OUT4分别接电机的两极。

## **3.1.3 控制核心部分实物接线图**

图3-2 Raspberry Pi实物连接图



## 3.2 室内智能调温

根据DHT11温湿度传感器所测得的室内温度与湿度数据，在树莓派中对数据信息进行分析与处理，并存入MySQL数据库中。根据当前温湿度情况，通过二路继电器控制风扇的打开与关闭。相关硬件连接如图3-3所示。

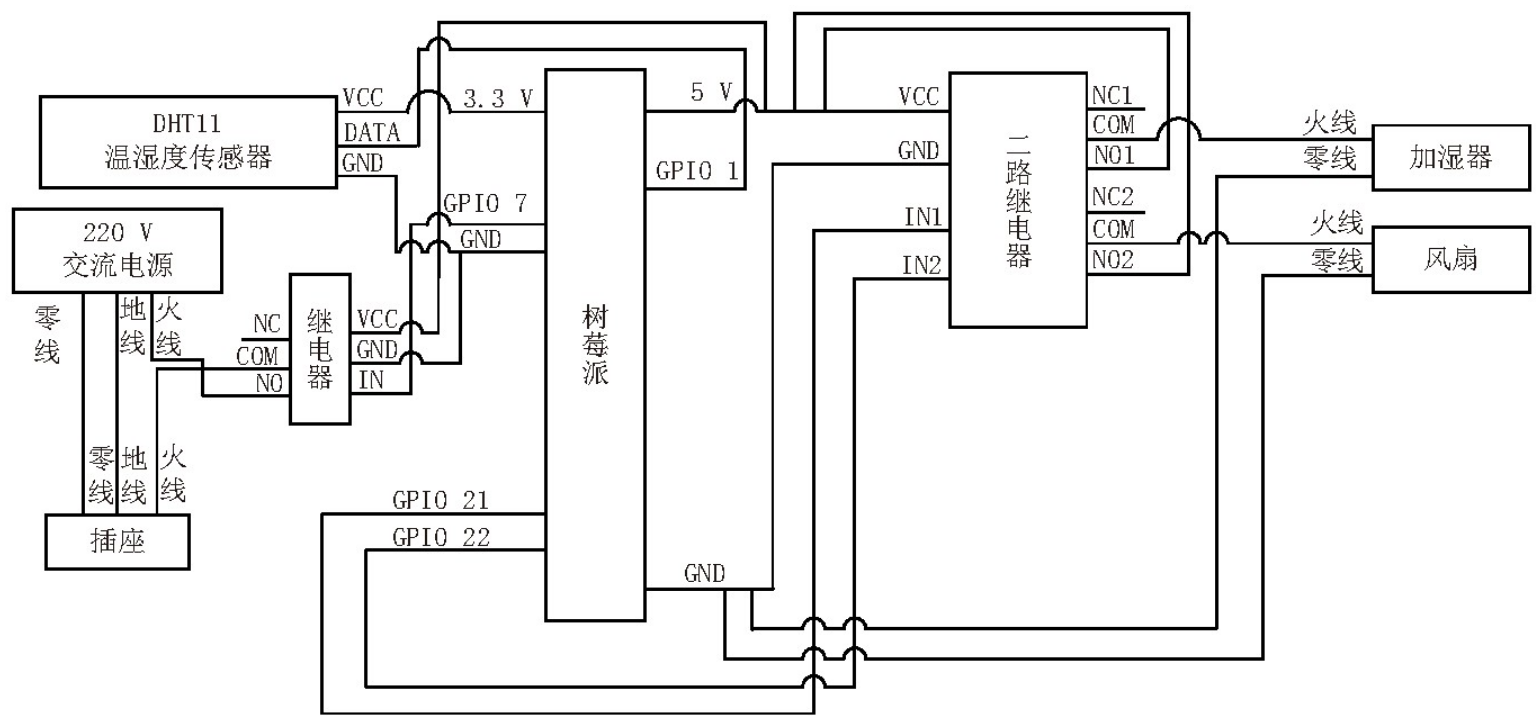


图3-3 温湿度调节相关硬件连接图

加湿器采用额定电压为DC 5 V、额定功率约2 W的小型加湿器；风扇采用供电电压为 DC 5 V、输出电流为1 A、功率约3 W的小型风扇。小型加湿器与风扇可使用树莓派供电。 若考虑功率更大的加湿器或风扇，可将加湿器或风扇插在与树莓派相连的插座上，通过 220 V 交流电源供电。

由于树莓派的 GPIO 引脚在 Output 状态默认为 0，因此 采用高电平触发的 5 V 二路继电器。树莓派提供5 V电源引脚与继电器的常开（No1、No2）触点相连，在常态（不通电）下处于断开状态，风扇与加湿器处于关闭状态。

DHT11 温湿度传感器含有电阻式感湿元件与 NTC 测温 元件，可输出经过校验后的温度与湿度数字信号，适用于室内温湿度的测量。其湿度测量范围为 20% ～ 95% ；温度测量 范围为 0 ～50℃。

利用 DHT11 温湿度传感器的检测数据控制加湿器与风扇的开关状态，每隔 5 min 判断一次。由于人体适宜的湿度为45% ～ 65%，因此在湿度小于 45% 的情况下打开加湿器， 湿度大于 65% 的情况下关闭加湿器 ；温度大于 27℃时风扇打开，温度小于23℃时，风扇关闭。该功能实现的程序流程如图3-4所示。

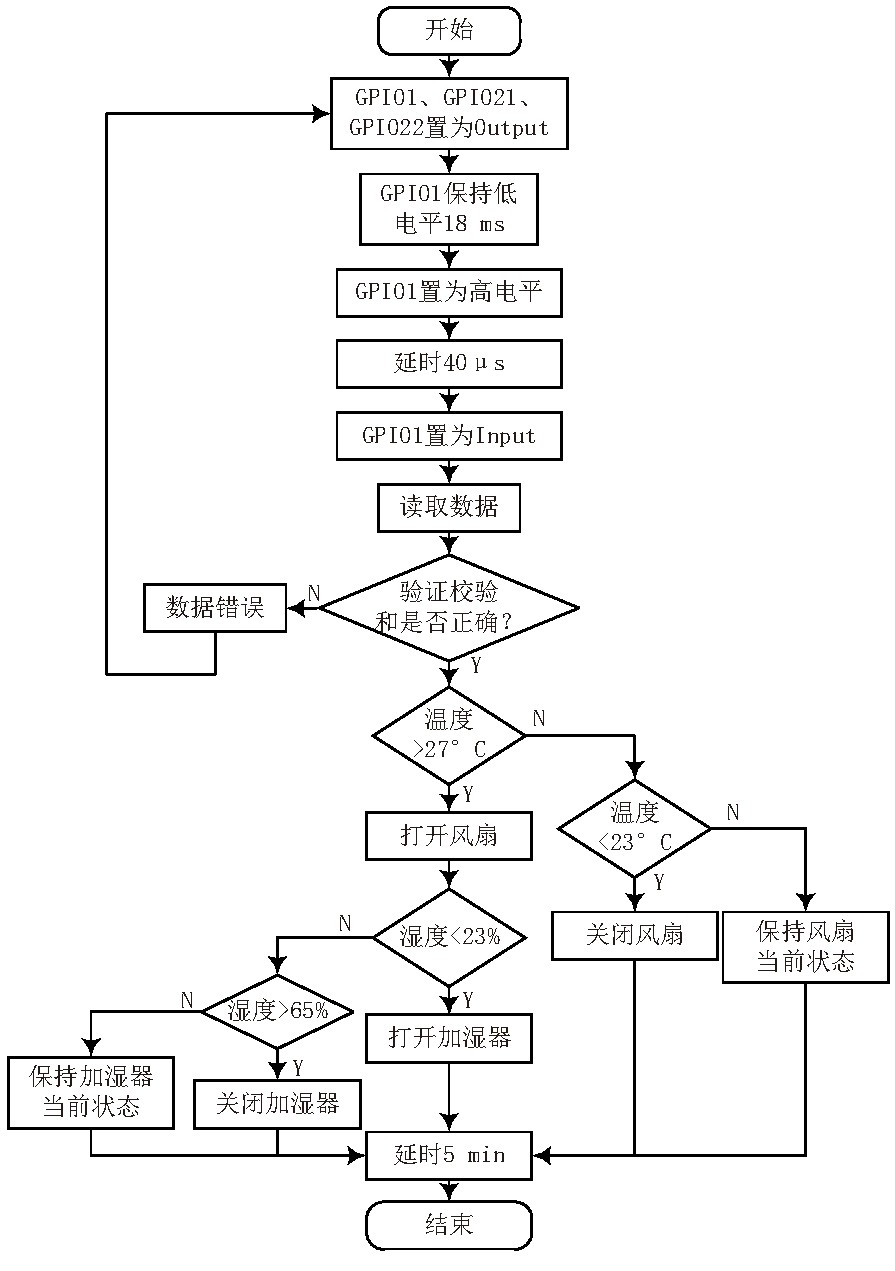


图3-4 室内温湿度调节程序流程图

室内智能调温的代码：

#coding: utf8

import Adafruit\_DHT

import MySQLdb

import datetime

#引入gpio的模块

import RPi.GPIO as GPIO

import time

# Set sensor type : Options are DHT11,DHT22 or AM2302

sensor=Adafruit\_DHT.DHT11

# Set GPIO sensor is connected to

gpio=4

dt=datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")

#设置GPIO模式

GPIO.setmode(GPIO.BOARD)

#设置in1到in4接口

IN1 = 13

IN2 = 15

IN3 = 16

IN4 = 18

#初始化接口

def init():

GPIO.setup(IN1,GPIO.OUT)

GPIO.setup(IN2,GPIO.OUT)

GPIO.setup(IN3,GPIO.OUT)

GPIO.setup(IN4,GPIO.OUT)

#后退

def cabk(sleep\_time):

GPIO.output(IN1,GPIO.LOW)

GPIO.output(IN2,GPIO.HIGH)

GPIO.output(IN3,GPIO.LOW)

GPIO.output(IN4,GPIO.HIGH)

time.sleep(sleep\_time)

GPIO.cleanup()

# Use read\_retry method. This will retry up to 15 times to

# get a sensor reading (waiting 2 seconds between each retry).

humidity, temperature = Adafruit\_DHT.read\_retry(sensor, gpio)

# Reading the DHT11 is very sensitive to timings and occasionally

# the Pi might fail to get a valid reading. So check if readings are valid.

if humidity is not None and temperature is not None:

print('Temp={0:0.1f}\*C Humidity={1:0.1f}%'.format(temperature, humidity))

print "temperature:",temperature

print "humidity:",humidity

conn=MySQLdb.connect(

host='localhost',

port=3306,

user='root',

passwd='root',

db='jiankong',

)

sql="insert into dht2(time,temperature,humidity) values('%s','%d','%d')"%(dt,temperature,humidity)

cur=conn.cursor()

cur.execute(sql)

cur.close()

conn.commit()

conn.close()

if temperature > 25:

init()

cabk(30)

else:

print('Failed to get reading. Try again!')

## 3.3 家庭智能预警

根据HC-SR501人体红外传感器来检测人体的活动情况，当检测到有人体活动的时候，蜂鸣器启动报警，与此同时树莓派摄像头启动并抓拍照片并将其自动上传至七牛云服务器上，与此同时短信示警用户。

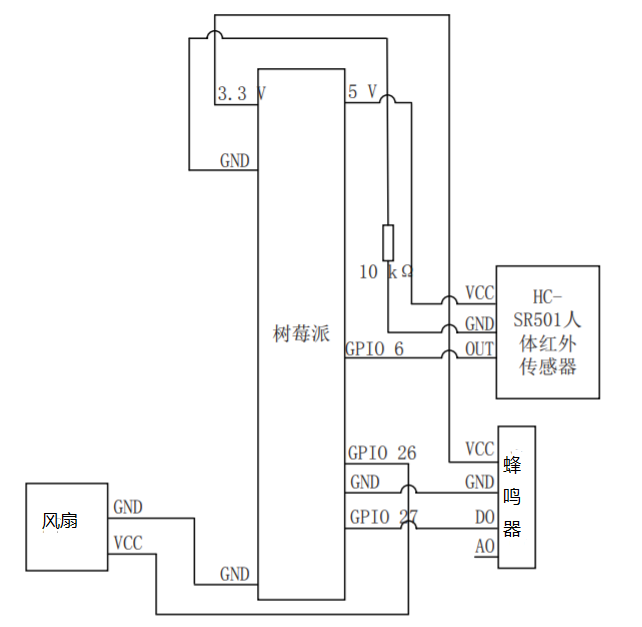


图3-5 家庭智能预警硬件连接图

### 3.3.1 七牛云对象存储介绍

在七牛云https://portal.qiniu.com/signup?code=3lmzbyfpvvfbm

官网上注册账号并进行实名认证，并在对象存储中创建jiankong这个存储空间。

图3-6 七牛云对象存储空间建立

图3-6 七牛云对象存储空间的建立

将树莓派抓拍的照片上传七牛云服务器的代码

*# -\*- coding: utf-8 -\*-*

**import** time

**from** qiniu **import** Auth, put\_file, etag, urlsafe\_base64\_encode

**import** qiniu.config

**import** os

*#需要填写你的 Access Key 和 Secret Key*

access\_key = ' ' *#这里的密钥填上刚才我让你记住的密钥对*

secret\_key = ' ' *#这里的密钥填上刚才我让你记住的密钥对*

*#构建鉴权对象*

q = Auth(access\_key, secret\_key)

*#要上传的空间*

bucket\_name = ' '*#这里填写你在七牛云创建的空间存储名称*

*#上传到七牛后保存的文件名*

key = '%s\_%s\_%s\_%s\_%s\_%s.jpg'%(time.localtime()[0],time.localtime()[1],time.localtime()[2],time.localtime()[3],time.localtime()[4],time.localtime()[5])

*#生成上传 Token，可以指定过期时间等*

token = q.upload\_token(bucket\_name, key, 3600)

*#要上传文件的本地路径*

localfile = 'current\_photo.jpg'

ret, info = put\_file(token, key, localfile)

filename = 'current\_photo.jpg'

**if** os.path.exists(filename):

    os.remove(filename)

将图片上传成功之后可在七牛云服务器上进行查看

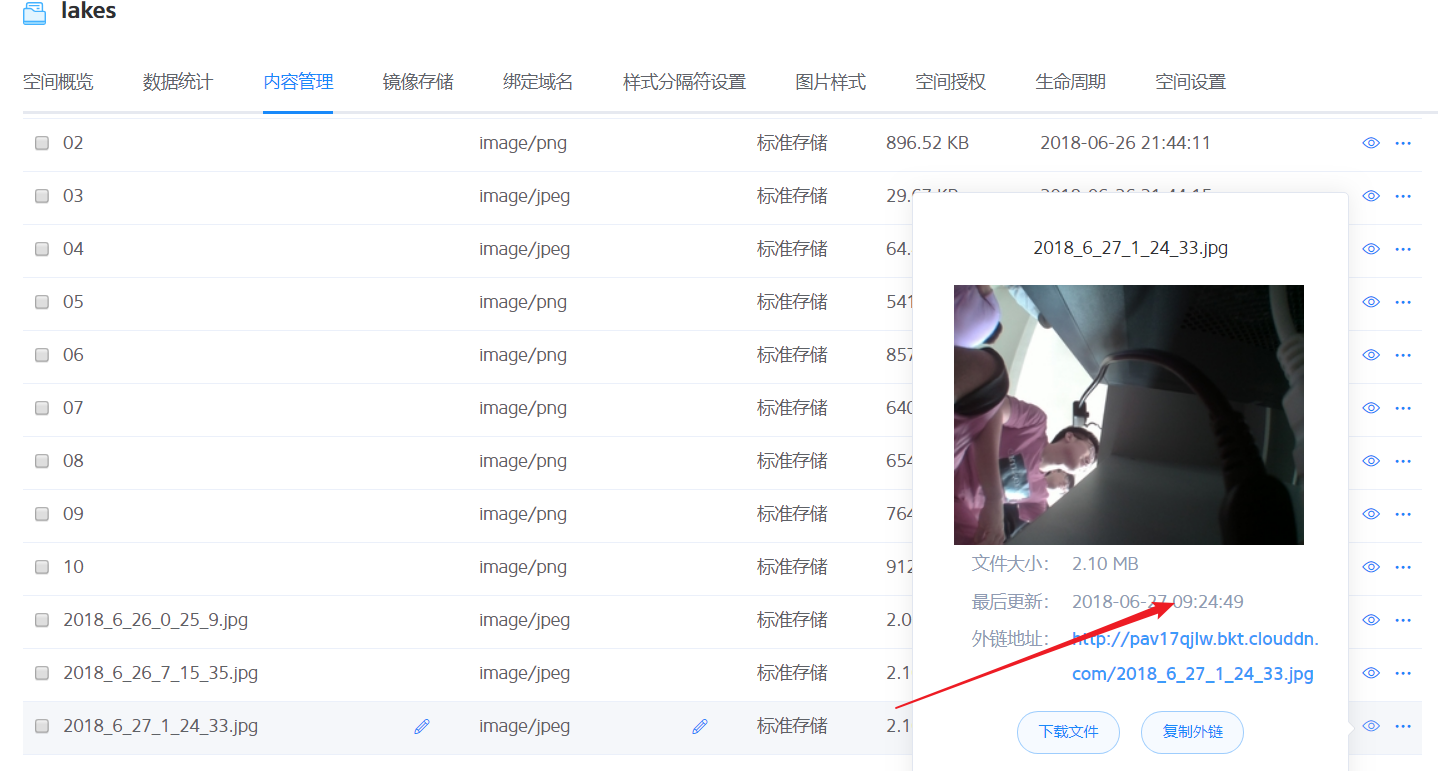


图3-7 图片成功上传至七牛云服务器

### 3.3.2 Python结合Twilio发送短信

先在Twilio官网上注册账号[https://www.twilio.com](https://www.twilio.com/" \t "https://blog.csdn.net/fengfengdiandia/article/details/_blank) 并建立项目应用，如下图所示，成功建立应用，并提供好所需钥匙。

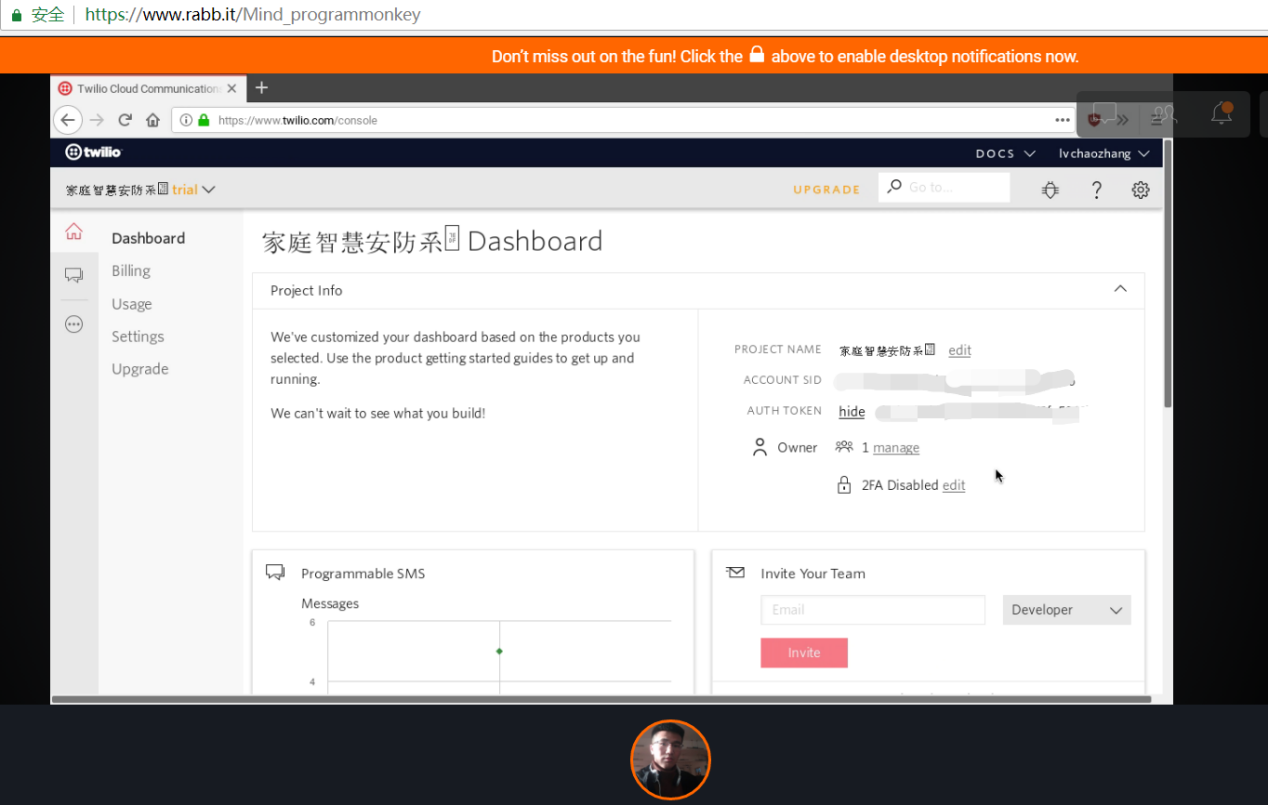


图3-8 家庭智慧安防项目成功创建

Python结合Twilio免费给手机发短信，其代码如下

from twilio.rest import Client

# Your Account SID from twilio.com/console

account\_sid = "ACXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

# Your Auth Token from twilio.com/console

auth\_token = "your\_auth\_token"

client = Client(account\_sid, auth\_token)

message = client.messages.create(

to="+15558675309",

from\_="+15017250604",

body="Hello from Python!")

print(message.sid)

注意这里to中的电话号码是你注册twilio的号码，from是你注册账号时免费得到的twilio电话号码。

通过上述python代码便可以向用户发送短信，下面是接收的短信情况。

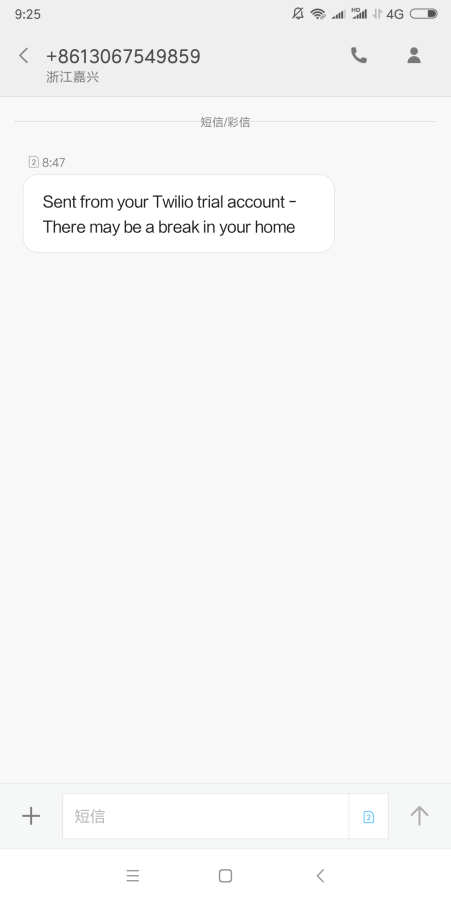


图3-9 预警短信

### 3.3.2 智能预警的实现

当人体感应模块感应到人体活动时，蜂鸣器警报报警，与此同时树莓派摄像头模块抓拍照片传到七牛云服务器，并且还会通过上述的twilio向用户发送一条示警短信。

智能预警的代码如下：

# -\*- coding: utf-8 -\*-

import RPi.GPIO as GPIO

import MySQLdb

import time

import picamera

from qiniu import Auth, put\_file, etag, urlsafe\_base64\_encode

import qiniu.config

import os

#需要填写你的 Access Key 和 Secret Key

access\_key = '' #这里的密钥填上刚才我让你记住的密钥对

secret\_key = '' #这里的密钥填上刚才我让你记住的密钥对

#构建鉴权对象

q = Auth(access\_key, secret\_key)

#要上传的空间

bucket\_name = 'jiankong'

#上传到七牛后保存的文件名

key = '%s\_%s\_%s\_%s\_%s\_%s.jpg'%(time.localtime()[0],time.localtime()[1],time.localtime()[2],time.localtime()[3],time.localtime()[4],time.localtime()[5])

#生成上传 Token，可以指定过期时间等

token = q.upload\_token(bucket\_name, key, 3600)

#

camera=picamera.PiCamera()

def uploadfile():

camera.capture("current\_photo.jpg")

#要上传文件的本地路径

localfile = 'current\_photo.jpg'

ret, info = put\_file(token, key, localfile)

filename = 'current\_photo.jpg'

if os.path.exists(filename):

os.remove(filename)

def init():

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BOARD)

GPIO.setup(12, GPIO.IN)

GPIO.setup(11,GPIO.OUT)

pass

def beep():

for i in range(1,6):

GPIO.output(11, GPIO.LOW)

time.sleep(0.5)

GPIO.output(11, GPIO.HIGH)

time.sleep(0.5)

print "the Buzzer will make sound"

def detct():

for i in range(1, 31):

if GPIO.input(12) == True:

print time.strftime('%Y-%m-%d %H:%M:%S',time.localtime(time.time()))+ " Someone is closing!"

conn=MySQLdb.connect(

host='localhost',

port=3306,

user='root',

passwd='root',

db='jiankong',

)

sql="insert into buzzer(time,descr) values('%s','%s')"%(time.strftime('%Y-%m-%d %H:%M:%S',time.localtime(time.time())),'someone is closing')

cur=conn.cursor()

cur.execute(sql)

cur.close()

conn.commit()

conn.close()

beep()

uploadfile()

else:

GPIO.output(11, GPIO.HIGH)

print time.strftime('%Y-%m-%d %H:%M:%S',time.localtime(time.time()))+" Noanybody!"

time.sleep(10)

time.sleep(2)

init()

detct()

## 3.4 实时监控

将家庭中的温湿度以及家庭闯入信息显示到网页，并通过mjpg-streamer软件将局域网内摄像头的实时监控情况呈现在网页中。

### 3.4.1 实时视频

a.更新列表

sudo apt-get update    #更新软件列表

b.安装工具

sudo apt-get install subversion  #Subversion是一个自由开源的版本控制系统

sudo apt-get install libjpeg8-dev  #JPEG支持库

sudo apt-get install imagemagick

sudo apt-get install libv4l-dev   #4l是小写"L"

sudo apt-get install cmake  #下载编译工具

c.从github上下载文件

sudo apt-get install git

git clone <https://github.com/jacksonliam/mjpg-streamer.git>

cd mjpg-streamer/mjpg-streamer-experimental #进入下载目录后进入左侧路径

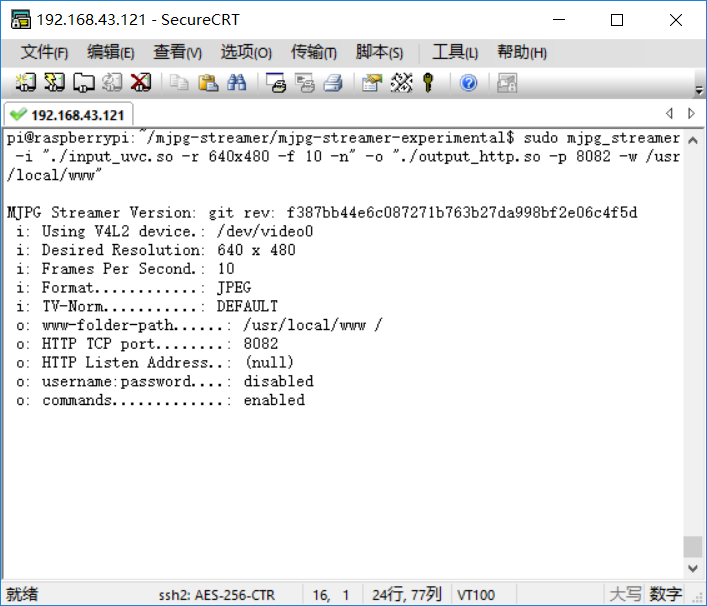
d.对文件进行编译

make all  #编译

sudo make install #安装

e.运行命令

sudo mjpg\_streamer -i "./input\_uvc.so -r 640x480 -f 10 -n" -o "./output\_http.so -p 8082 -w /usr/local/www"  #此命令尤为重要，如下图所示，输出信息，说明成功！



f.查看效果

游览器输入http://树莓派ID地址:8082/?action=stream可以查看实时视频。http://树莓派ID地址:8082/?action=snapshot可以进行拍照。

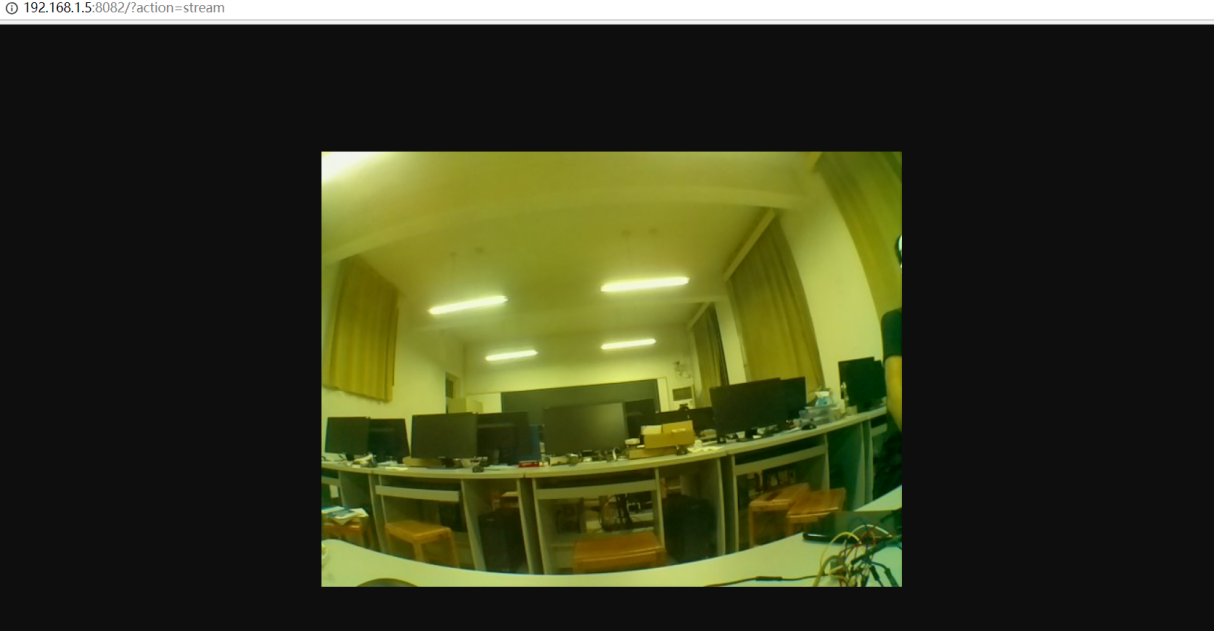


图3-10 实时监控的画面

### 3.4.2 Tomcat服务器

（1）tomcat2的安装

sudo apt-get install apache2

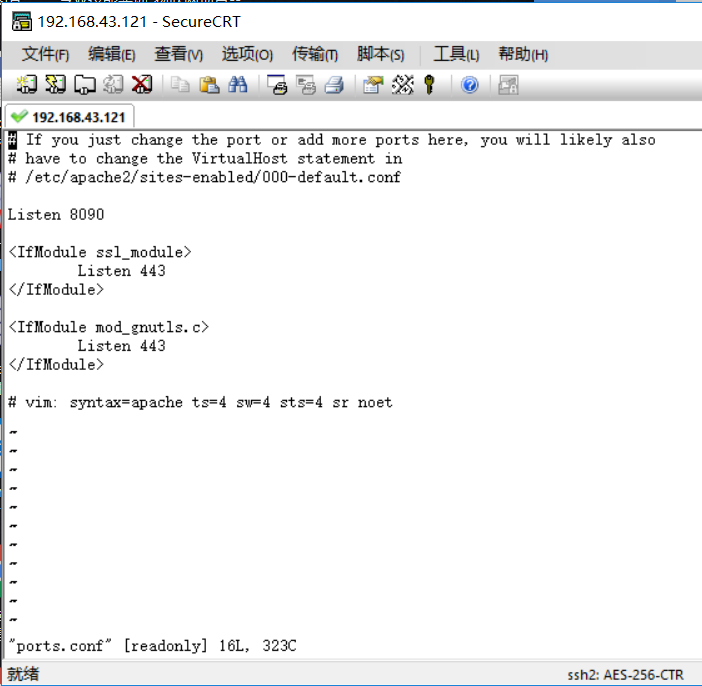
（2）tomcat2的端口的修改

cd /etc/apache2

打开端口配置文件进行修改

vim ports.conf

将Listen的端口修改，这里我修改为8090.



（3）进入tomcat2的工作目录

cd /var/www/html/

将写好的网页放在这里即可通过tomcat来访问。

### 3.4.3 实时监控网页的实现

安装相关软件依赖

sudo apt-get install php5 libapache2-mod-php5

这里通过php来跟数据库进行连接，获取数据库中的数据，并显示在网页上。网页这里通过了bootstrap来修饰一下，引用bootstrap的代码段。

!-- Bootstrap -->

<link href="https://cdn.bootcss.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">

<!-- jQuery (Bootstrap 的所有 JavaScript 插件都依赖 jQuery，所以必须放在前边) -->

<script src="https://cdn.bootcss.com/jquery/1.12.4/jquery.min.js"></script>

<!-- 加载 Bootstrap 的所有 JavaScript 插件。你也可以根据需要只加载单个插件。 -->

<script src="https://cdn.bootcss.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

jiankong.php的网页

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>家庭实时监控系统</title>

<!-- Bootstrap -->

<link href="https://cdn.bootcss.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">

<!-- jQuery (Bootstrap 的所有 JavaScript 插件都依赖 jQuery，所以必须放在前边) -->

<script src="https://cdn.bootcss.com/jquery/1.12.4/jquery.min.js"></script>

<!-- 加载 Bootstrap 的所有 JavaScript 插件。你也可以根据需要只加载单个插件。 -->

<script src="https://cdn.bootcss.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

</head>

<body>

<center>

<table class="table table-striped">

<h1>家庭实时监控系统</h1>

<br>

<h3 align="left" class="bg-primary">温湿度信息</h3>

<tr><th>时间(yyyy-mm-dd)</th><th>温度(°C)</th><th>湿度(%RH)</th></tr>

<?php

$servername = "localhost";

$username = "root";

$password = "root";

$dbname = "jiankong";

// 创建连接

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("连接失败: " . $conn->connect\_error);

}

$sql = "select \* from dht2 order by id desc limit 3";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

// 输出数据

while($row = $result->fetch\_assoc()) {

echo"<tr><td>".$row["time"]."</td><td>".$row["temperature"]."</td><td>".$row["humidity"]."</td><tr>";

}

} else {

echo "0 结果";

}

$conn->close();

?>

</table>

<table class="table table-striped">

<h3 align="left" class="bg-primary">安防情况</h3>

<tr><th>序号</th><th>时间</th><th>描述</th></tr>

<?php

$servername = "localhost";

$username = "root";

$password = "root";

$dbname = "jiankong";

// 创建连接

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("连接失败: " . $conn->connect\_error);

}

$sql = "select \* from buzzer order by id desc limit 3";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

// 输出数据

while($row = $result->fetch\_assoc()) {

echo"<tr><td>".$row["id"]."</td><td>".$row["time"]."</td><td>".$row["descr"]."</td><tr>";

}

} else {

echo "0 结果";

}

$conn->close();

?>

</table>

</center>

<h3 align="left" class="bg-primary">实时监控情况</h3>

<div>

<img style="-webkit-user-select: none;cursor: zoom-in;" src="http://192.168.43.121:8082/?action=stream" width="400" height="300"/>

</div>

</body>

</html>

结果展示图



图3-11 家庭实时监控系统网页端的实现

### 3.4.4 内网穿透

内网穿透的前期背景知识，可参考我的这篇博客

<https://blog.csdn.net/mind_programmonkey/article/details/80285596>

具体步骤：

（1）在这个网站下载python版本的

<https://www.ngrok.cc/download.html>

（2）在这个网址中开通免费隧道，然后将网址的IP地址和端口映射上去（具体的步骤可参考上述博客地址）。这里注意因为一共是两个地址，一个是在线监控的地址，一个是网页的地址，端口不一致，所以我们需要开通两个隧道。这里我采取的办法是用两个账号，分别注册，来映射端口



（3）运行

将（1）下载到的python版本运行，粘贴上隧道id，即可成功开启。

成功运行之后即可通过域名来访问我们家庭的监控情况。



图3-12 内网穿透之后的实时监控的画面

## 3.5 人脸识别门锁

当触摸传感器感应到有人触碰时，树莓派摄像头抓拍照片，并启动人脸识别功能，通过Face++云平台实现人脸图像的识别处理，根据人脸识别判断的结果控制电磁锁的开关。

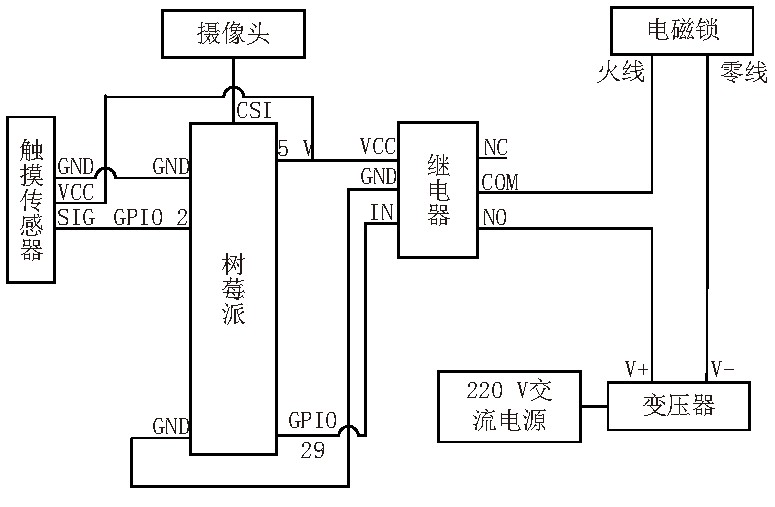


图3-13 人脸识别门锁相关硬件连接图

人脸识别门锁的流程图

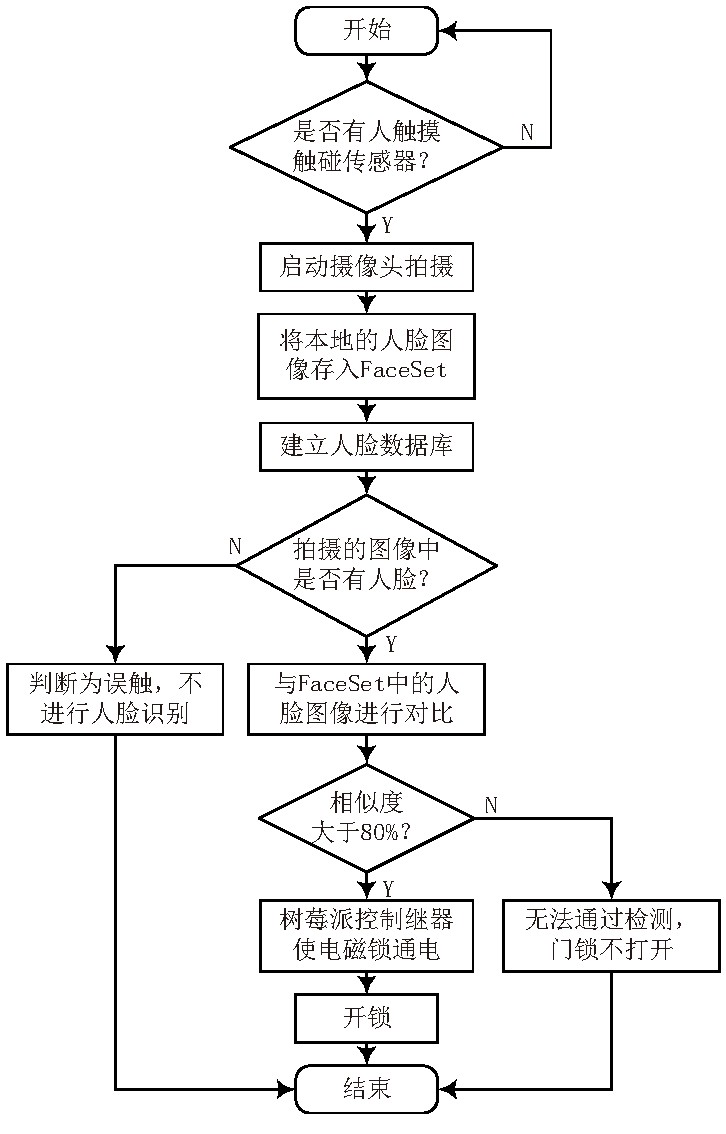


图3-14人脸识别门锁的流程图

### 3.5.1 人脸检测

（1）首先我们新建一个face目录用来存放整个项目的数据和源码



（2）获取原始图像

raspistill -o image.jpg



图3-15 原始图像

（3）对image.jpg原始图像进行人脸识别

jiance.py代码

import cv2

def detect(path):

img = cv2.imread(path)

cascade = cv2.CascadeClassifier("/home/pi/haarcascade\_frontalface\_alt.xml")

rects = cascade.detectMultiScale(img, 1.3, 4, cv2.cv.CV\_HAAR\_SCALE\_IMAGE, (20,20))

if len(rects) == 0:

return [], img

rects[:, 2:] += rects[:, :2]

return rects, img

def box(rects, img):

for x1, y1, x2, y2 in rects:

cv2.rectangle(img, (x1, y1), (x2, y2), (127, 255, 0), 2)

cv2.imwrite('/home/pi/face/handle\_data/image.jpg', img);

rects, img = detect("/home/pi/face/data/image.jpg")

box(rects, img)

（4）识别出的图片存储在handle\_data中



图3-16 检测出的人脸

### 3.5.2 人脸裁剪

（1）face.py代码

#-\*-coding:utf8-\*-

import os

import cv2

import time

import shutil

def getAllPath(dirpath, \*suffix):

PathArray = []

for r, ds, fs in os.walk(dirpath):

for fn in fs:

if os.path.splitext(fn)[1] in suffix:

fname = os.path.join(r, fn)

PathArray.append(fname)

return PathArray

#从源路径中读取所有图片放入一个list，然后逐一进行检查，把其中的脸扣下来，存储到目标路径中

def readPicSaveFace(sourcePath,targetPath,invalidPath,\*suffix):

try:

ImagePaths=getAllPath(sourcePath, \*suffix)

#对list中图片逐一进行检查,找出其中的人脸然后写到目标文件夹下

count = 1

# haarcascade\_frontalface\_alt.xml为库训练好的分类器文件，下载opencv，安装目录中可找到

face\_cascade = cv2.CascadeClassifier('/home/pi/haarcascade\_frontalface\_alt.xml')

for imagePath in ImagePaths:

img = cv2.imread(imagePath)

if type(img) != str:

faces = face\_cascade.detectMultiScale(img, 1.1, 5)

if len(faces):

for (x, y, w, h) in faces:

# 设置人脸宽度大于128像素，去除较小的人脸

if w>=128 and h>=128:

# 以时间戳和读取的排序作为文件名称

listStr = [str(int(time.time())), str(count)]

fileName = ''.join(listStr)

# 扩大图片，可根据坐标调整

X = int(x\*1)

W = min(int((x + w)\*1),img.shape[1])

Y = int(y\*1)

H = min(int((y + h)\*1),img.shape[0])

f = cv2.resize(img[Y:H, X:W], (W-X,H-Y))

cv2.imwrite(targetPath+os.sep+'%s.jpg' % fileName, f)

count += 1

print imagePath + "have face"

else:

shutil.move(imagePath, invalidPath)

except IOError:

print "Error"

else:

print 'Find '+str(count-1)+' faces to Destination '+targetPath

if \_\_name\_\_ == '\_\_main\_\_':

invalidPath = r'/home/pi/face/haveNoPeople'

sourcePath = r'/home/pi/face/data'

targetPath = r'/home/pi/face/faceOfPeople'

readPicSaveFace(sourcePath,targetPath,invalidPath,'.jpg','.JPG','png','PNG')

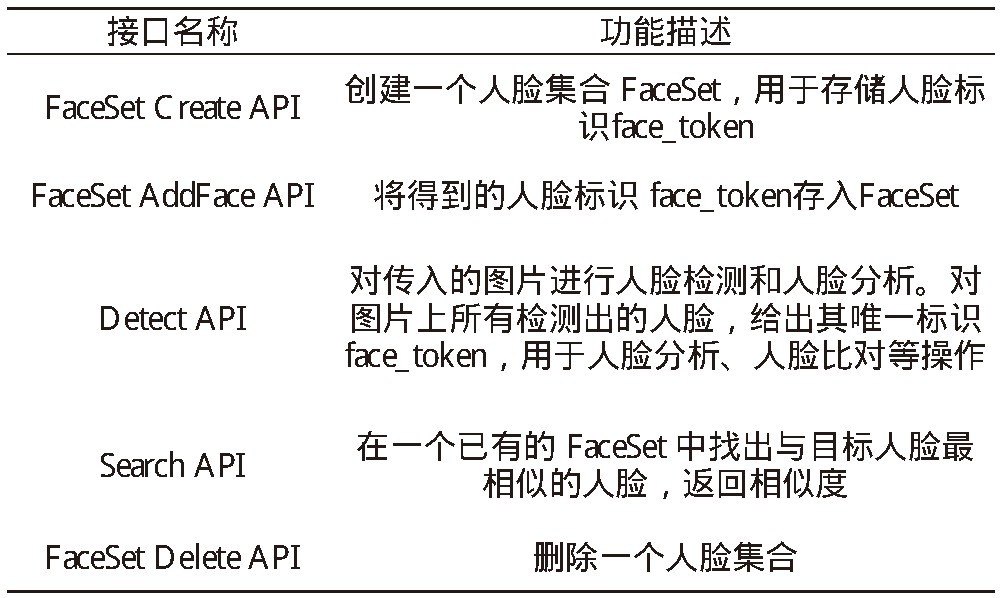
（2）裁剪出的人脸图像



图3-17 检测出的人脸

### 3.5.3 人脸识别

人脸识别功能基于Face++云平台，其API以Http请求的方式进行调用，所使用的接口及其功能描述见表1所列。



首先我们先需要将本地的人脸图像及相关信息上传至Face++云平台，存入Face Set中，建立人脸数据库。然后我们就可以判断拍摄的待识别图片中是否有人脸，若有人脸，将其余FaceSet中的人脸图像进行对比，当相似度大于80%，则打开门锁。

（1）获取本地人脸图像的face\_token的值

#coding=utf8

import cv2

import requests

import json

url = 'https://api-cn.faceplusplus.com/facepp/v3/detect'

files = {'image\_file':open('owner.jpg', 'rb')}

payload = {'api\_key': '\*\*填入你的api值\*\*\*',

'api\_secret': '\*\*填写你的api值\*\*',

'return\_landmark': 0,

'return\_attributes':'gender,age,glass'}

r = requests.post(url,files=files,data=payload)

data=json.loads(r.text)

print r.text

width = data['faces'][0]['face\_rectangle']['width']

top = data['faces'][0]['face\_rectangle']['top']

height = data['faces'][0]['face\_rectangle']['height']

left = data['faces'][0]['face\_rectangle']['left']

img = cv2.imread("owner.jpg")

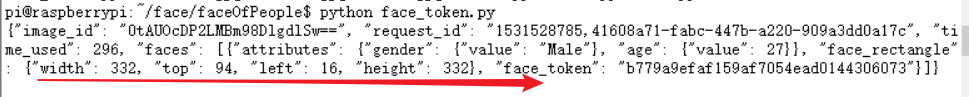
vis = img.copy()

cv2.rectangle(vis, (left, top), (left+width, top+height),(0, 255, 0), 2)

cv2.imshow("Image", vis)

cv2.waitKey (0)

获取的face\_token值，需保存，后续要用。



（2）获取人脸集合的face\_tokens

face\_tokens.py所需的代码

#coding=utf8

import requests

url = 'https://api-cn.faceplusplus.com/facepp/v3/faceset/create'

payload = {'api\_key': '\*\*\*申请的值\*\*\*\*',

'api\_secret': '\*\*\*申请的值\*\*\*',

'display\_name':'jiankong\_lcz',

'outer\_id':'jiankong\_lcz',

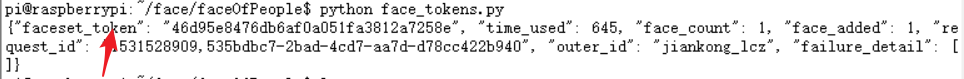
'face\_tokens':'上一步图片的face\_token的值'

}

r = requests.post(url,data=payload)

print r.text

返回的faceset\_token值，也需要保存



（3）通过上述两步，我们已经将本地人脸图像存入人脸集合中，下面我们打开摄像头进行拍摄识别人脸。

#coding=utf8

import cv2

import cv2.cv as cv

import requests

import json

import picamera

import os

camera=picamera.PiCamera()

camera.capture("current\_photo.jpg")

filename='current\_photo.jpg'

url = 'https://api-cn.faceplusplus.com/facepp/v3/search'

payload = {'api\_key': '\*\*\*填入你申请的key值\*\*\*\*',

'api\_secret': '\*\*\*填入你申请的key值\*\*\*\*',

'faceset\_token':'\*\*上一步你获取对的faceset值\*\*',

}

files = {'image\_file':open('current\_photo.jpg', 'rb')}

r = requests.post(url,files=files,data=payload)

data=json.loads(r.text)

print r.text

if os.path.exists(filename):

os.remove(filename)

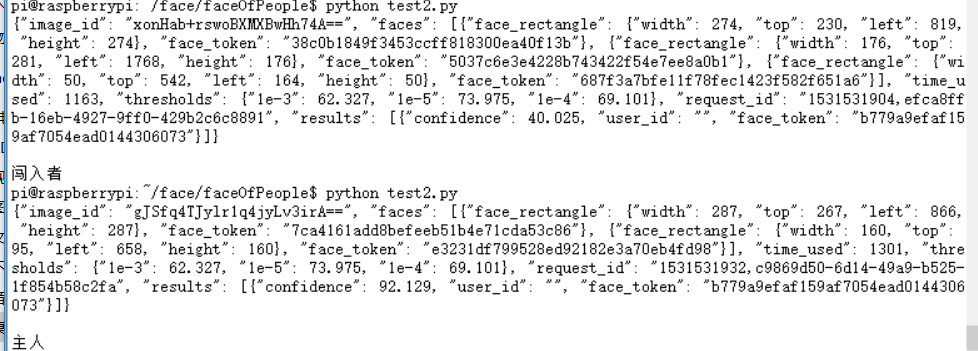
if data["results"][0]["face\_token"] == "\*\*\*owner图片的face\_token值\*\*\*" and data["results"][0]["confidence"]>=data["thresholds"]["1e-5"]:

print'\n主人'

else:

print '\n闯入者'

首先这个会调用树莓派的摄像头获取当前图像，并截图，截图成功之后会将这张人脸图片上传至云端进行对比并返回结果。



# 4 项目总结

本文介绍了通过实现基于树莓派的智能家居监控系统的可行性，设计了各功能的硬件连接图与软件实现方案。以树莓派作为控制终端，可实现室内智能调温，家庭的智能预警，实时监控以及人脸识别门锁等功能于一体的智能家居系统。对各项功能进行实验与操作， 已全部达到预期效果。系统功能集成度较高，设计易于搭建， 相关硬件成本低廉，具有广阔的应用市场与发展前景。

**5 参考文献**

[1]邱丽华 .人脸信息认证系统基于设计模式的设计与实现 [D]. 武汉：华中科技大学， 2008.

[2] Rushi Gajjar. 树莓派 + 传感器 ：创建智能交互项目的实用方法、工具及最佳实践 [M]. 胡训强，张欣景，译 . 北京：机械工业出版社，2016.

[3]张毅刚，刘旺，邓立宝 . 单片机原理及接口技术 [M]. 北京 ：人民邮电出版社，2016.

[4] Alex Bradbury， Ben Everard.树莓派Python编程指南[M].王文峰，译 . 北京 ：机械工业出版社，2015.

[5] Tero Karvinen，Kimmo Karvinen，Ville Valtokari. 传感器实战全攻略 [M].于欣龙，李泽，译 . 北京 ：人民邮电出版社，2016.

[6] 眭晋华，史红梅，眭广瑞 . 空气湿度及其影响 [J]. 山西科技，2011，26（3）：143-144.

[7] 来清民 . 传感器与单片机接口及实例 [M]. 北京 ：北京航空航天大学出版社，2008.

[8] face++ API 文 档 [EB/OL].https ：//console.faceplusplus.com.cn/documents/4887579.

[9] 王琦，阮淑云 . 光敏电阻传感器的特性研究 [J]. 都市家教（上半月），2014（6）：188.

[10] 李萌 . 智能楼道照明节电控制系统的设计 [J]. 电子制作，2014（10X）：49-50.