**树莓派智能音乐小管家**

作为一个每天晚睡的程序员，最不想听到的就是天天重复的一样的闹钟。精彩的一天，应该有一个不一样的开始。对我这种喜爱音乐的人来说，此时期待的就是一个有着不同音乐的早晨。

于是，我做了这个音乐小管家。原本只是想做个闹钟而已，但随着我一边开发一边扩展，现在它已经可以语音整点报时，天气播报，定时音乐，回家欢迎。音乐是在百度音乐电台随机抓取的，而回家欢迎是利用一点小策略实现了简单的智能身份识别，能识别出主人，家人，朋友，以及陌生人，并在进门的时候给予不同的欢迎语音，同时献上一首音乐。

下面请听我详细地介绍吧。

首先准备硬件。

树莓派一只，版本无所谓，B或者B+都行。

3.5mm接口的音响一个。

Wifi网卡一枚

杜邦线若干

人体红外传感器一个（这只是个触发开关，对应你喜欢的触发方式，可以用任何你习惯或喜欢的传感器或开关设备）

**搭建硬件**

树莓派接音箱和wifi网卡都不用说了，非常简单。这里主要说下传感器的接线。引脚编号配合后面会讲到的代码

红外传感器一共有3个引脚，+，-，out。

当传感器感应到人体红外时，out引脚会发出高电平。程序中树莓派捕获该引脚上的高电平，做出相应操作。

+是正极，接入树莓派的OUT引脚，这里用的是16号引脚。-是负极，接入树莓派的GND引脚，这里用的是14号引脚。Out是传感器的数据引脚，接入树莓派的in引脚，这里是12号引脚。

**语音和音乐播放**

这里我采用的是omxplayer这个播放器。在树莓派中配置这个播放器，你需要做这几件事情。

1. 把声音模块加入内核

sudo modprobe snd-bcm2835

2. 安装omxplayer

sudo apt-get install omxplayer

现在你可以使用omxplayer命令来测试这个播放器了，参数是声音文件路径。

要实现语音报时，需要实现文字转语音。这里我采用了google翻译的api（[http://translate.google.cn/translate\_tts?tl=%s&q](http://translate.google.cn/translate_tts?tl=%25s&q)=需要转换的文字），利用url请求获取google translate服务器转换的语音，然后用omxplayer播放出来。嗯，还是很悦耳的女声呢。让她尽情地在你的程序中撒娇卖萌吧！

我写的是python程序，利用urllib和requests库，请求这个接口。在树莓派环境中要注意的是，需要指定浏览器类型，否则会因为参数的编码造成转换错误，你听到的会是由参数乱码转换的语音。

Google文字转语音的实现如下：

words = “要转换的文字”

param = {'q':words}

data = urllib.urlencode(param)

googleAPIurl = [http://translate.google.cn/translate\_tts?tl=%s&](http://translate.google.cn/translate_tts?tl=%25s&)

s = requests.Session()

s.headers.update({'User-Agent': 'Mozilla/5.0 (Windows NT 5.1; rv:17.0) Gecko/20100101 Firefox/17.0','Connection': 'Keep-Alive'}) #设置浏览器类型

ret = s.post((googleAPIurl%lan)+data)

ext = ret.headers["content-type"].split("/")[1]

filename =("sound.%s" % ext)

with open(filename, "wb") as f:

f.write(ret.content)

通过这段程序，语音将被保存为一个名为sound.mpeg的文件。用omxplayer播放这个文件，就能听到音箱里传来你文字的语音。

接下来是随机抓取音乐。

音乐来自<http://fm.baidu.com> 这是百度的随心听音乐电台。这里面的音乐其实已经是被百度筛选归类过的，有多个不同风格的音乐频道，音乐的品味和质量也非常不错。所以在这里面随机我还是比较放心的，一般不会遇到恶俗的音乐，即使有，我也可以屏蔽对吧。

音乐抓取的代码如下。

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

import urllib

import re,json,random,time

import ConfigHelper

fm = 'http://fm.baidu.com'

listurl = "http://fm.baidu.com/dev/api/?tn=playlist&format=json&id="

songurl = "http://music.baidu.com/data/music/fmlink?type=mp3&rate=320&songIds="

#屏蔽频道

disChannels = ConfigHelper.GetDisabled("disChannels")

#屏蔽歌手

disSinger = ConfigHelper.GetDisabled("disSinger")

def GetChannelList():

#抓取频道列表

mf = urllib.urlopen(fm)

html = mf.read()

html = html.decode('utf8')

start = html.find("{", html.find("rawChannelList"))

end = html.find(";", start)

listjson = html[start:end].strip()

data = json.loads(listjson)

channel\_id\_list = []

for item in data['channel\_list']:

channel\_id\_list.append(item['channel\_id'])

#随机一个频道

flag = True

channelId = ""

while flag:

channelId = channel\_id\_list[random.randint(0, len(channel\_id\_list)-1)]

if channelId not in disChannels:

flag = False

return GetSongList(channelId)

def GetSongList(channelid):

mf = urllib.urlopen(listurl+channelid)

html = mf.read()

html = html.decode('utf8')

data = json.loads(html)

song\_list = []

for item in data['list']:

song\_list.append(str(item['id']))

#随机一首歌曲

flag = True

singer = ""

songInfo = ""

while flag:

singer,songInfo = GetSongLink(song\_list[random.randint(0, len(song\_list)-1)])

if singer not in disSinger:

flag = False

return songInfo

def GetSongLink(songid):

mf = urllib.urlopen(songurl+songid)

html = mf.read()

html = html.decode('utf8')

jdata = json.loads(html)

data = jdata['data']

songList = data['songList']

songLink = songList[0]['songLink']

return (songList[0]['artistName'],songList[0]['songName']+"|"+songList[0]['artistName']+"|"+songLink)

def GetMusic():

return GetChannelList()

if \_\_name\_\_ == "\_\_main\_\_":

print GetMusic();

你可以去定义你想随机的频道，也可以屏蔽你不喜欢的频道。我屏蔽了儿歌、DJ、网络歌曲，屏蔽信息写在配置文件中。

我设置了一个ini配置文件，程序的一些参数我都是配置在这个文件中，写了一个脚本来读取和设置。

配置文件操作的脚本程序如下

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

import ConfigParser

iniPath = "/home/pi/MyPrograms/MusicSteward/Configs/base.ini"

#获取基础配置

def GetIni(key,value="",title="base"):

try:

config = ConfigParser.ConfigParser()

config.readfp(open(iniPath),"rb")

return config.get(title,key).decode("gbk")

except Exception,e:

return value

pass

#获取屏蔽信息

def GetDisabled(key,title="disabled"):

try:

result = GetIni(key,"",title)

return result.split(",")

except Exception,e:

return []

pass

#获取语音词组

def GetWord(key,title="word"):

try:

return GetIni(key,"突然不知道该说啥好呢",title)

except Exception,e:

print e

return "突然不知道该说啥好呢"

pass

def SetValue(title,key,value):

config = ConfigParser.ConfigParser()

config.readfp(open(iniPath),"rb")

config.set(title, key, value)

config.write(open(iniPath, "w"))

实现了这两块，就可以很轻易地编写定时程序了，在你想要的任何时候播放音乐和语音。

**天气播报**

天气播报相对容易，各方面资料也很多，这里采用中国天气网的公共接口来获取当前天气数据，城市代码可以到<http://www.weather.com.cn>网站上找到。脚本如下：

# -\*- coding: UTF-8 -\*-

import sys,urllib2,json

cityList = [

{'code':"101280601", 'name':"深圳"}

]

def getCityWeather\_RealTime(cityName):

cityID = getCityCodeFromName(cityName)

url = "http://www.weather.com.cn/data/sk/" + str(cityID) + ".html"

try:

req=urllib2.Request(url)

stdout = urllib2.urlopen(url)

weatherInfomation = stdout.read().decode('utf-8')

data=json.loads(weatherInfomation)

msg = data["weatherinfo"]

result = u"气温:%s摄氏度，风向:%s，湿度:%s"%(msg["temp"],msg["WD"],msg["SD"])

except (SyntaxError) as err:

return u"天气获取异常"

except:

return u"天气获取失败"

else:

return result

finally:

None

def getCityCodeFromName(cityName):

for item in cityList:

#n = unicode(item['name'],'utf-8')

#print n,cityName

if item['name']==cityName:

#print 'equal'

return item['code']

return ''

通过这个脚本获取天气数据，然后调用之前的语音接口，就可以将它播报出来。

**回家欢迎**

这个功能稍微特别，它可以在任何时候感应到我回家的事件，从而触发一系列操作。这里我的小小策略就是利用wifi做了身份识别。比如我下班回家，我还不需要进家门，刚出电梯，我的手机就已经自动连上家里的wifi了。通过对手机mac地址的匹配，就能知道来者身份是我，还是我家人，或是我朋友，或是陌生人。

这里是让程序定时扫描我的wifi网关，在路由器网关的页面用正则表达式提取当前连接的主机，通过差异化比较识别出新登陆的主机。

这里由于大家所用路由器厂家各异，所以登陆网关和提取页面信息的方式未必适用，这可能就需要大家自己去研究一下自己路由器的配置了。我的脚本实现如下。

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

import urllib,json,time,threading

import sys,subprocess

import SpeakHelper

import InfraredHelper

import ConfigHelper

oldList = []

class CheckGuest(threading.Thread):

def \_\_init\_\_(self):

threading.Thread.\_\_init\_\_(self)

self.thread\_stop = False

self.oldList = []

def run(self):

while not self.thread\_stop:

try:

rpm = "http://admin:123456 @192.168.1.1/userRpm/WlanStationRpm.htm?Page=1 "

mf = urllib.urlopen(rpm)

html = mf.read()

#html = html.decode('utf8')

start = html.find("(", html.find("var hostList = new Array("))

end = html.find(";", start)

listjson = html[start:end].strip()

listjson = listjson.replace("(\n","").replace(",\n0,0 )","").replace(",","")

data = listjson.split("\n")

count = 0;

listmac = []

guests = ""

isKnower = False

for i in range(0,len(data)/4):

listmac.append(data[i\*4].replace("\"",""))

if len(self.oldList)>0 and len(listmac)>len(self.oldList):

newmac = set( self.oldList ) ^ set( listmac )

for item in newmac:

if ConfigHelper.GetFamily(item) != "":

isKnower = True

guests += ConfigHelper.GetFamily(item)+"，"

elif ConfigHelper.GetFriend(item) != "":

guests += ConfigHelper.GetFriend(item)+"，"

elif ConfigHelper.GetNone(item) != "":

None

else:

guests += ConfigHelper.GetWord("myfriend")

if len(guests) > 0:

if isKnower:

SpeakHelper.Speak("zh-CN", guests + ConfigHelper.GetWord("backhome"))

guests += ConfigHelper.GetWord("iwelcome")

InfraredHelper.CheckInfrared(guests,3)

else:

if guests.find(ConfigHelper.GetWord("myfriend")) == -1:

time.sleep(20)

guests += ConfigHelper.GetWord("owelcome")

SpeakHelper.soundStart(guests, 1)

listmac = []

self.oldList = listmac

except Exception,e:

SpeakHelper.Speak("zh-CN", ConfigHelper.GetWord("scanerror"))

pass

finally:

time.sleep(5)

def stop(self):

self.thread\_stop = True

完成这一步，还有点缺陷。因为wifi连接上之后，你可能还在门外，还没开门。如果这时候开始欢迎，未免有些美中不足。这里我采用人体红外识别的方式，来感应到我开门进家的事件。由于本人是租的房，不太好动电路，而且树莓派放在距离门口较远的位置，所以利用红外感应开关对着门口扫描比较合适。开门的事件其实有很多种更好的办法实现，但由于条件限制，我就先这么简单地做了。

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

import RPi.GPIO as GPIO

import SpeakHelper

import time

import ConfigHelper

GPIO.setmode(GPIO.BOARD)

GPIO.setwarnings(False)

GPIO.setup(12, GPIO.IN)

GPIO.setup(16, GPIO.OUT)

def CheckInfrared(talk,cnt=3):

flag = True

count = 0

GPIO.output(16, GPIO.HIGH)

while flag and count < int(ConfigHelper.GetIni("infrared","60")):

if(GPIO.input(12)==True):

GPIO.output(16, GPIO.LOW)

SpeakHelper.soundStart(talk, cnt)

#time.sleep(3)

flag = False

count = int(ConfigHelper.GetIni("infrared","60"))

count += 1

time.sleep(1)

if flag:

SpeakHelper.Speak("zh-CN", ConfigHelper.GetWord("unfind"))#未扫描到对象

完成这一步之后，每当我一回到家，一开门，就有一个悦耳的女声欢迎我说“主人，你终于回来啦，人家等了你好久呢，我为你准备了几首歌，放松一下吧！“然后就听见音乐响起。哈哈，真是惬意啊。

现在，我享受着每个有音乐的早晨，每天都听到不一样的音乐；每天下班回家，就能听到它的迎接；朋友来我家玩，一蹭网它就欢迎朋友并自我介绍；每个整点都提醒报时，告诉我温度湿度。每天夜晚都提醒我休息，同时温柔地送上一首歌曲。真像一个小小管家!

完整的程序代码我已经提交到这里，欢迎下载。