

浙江大学

本科实验报告

课程名称： 嵌入式系统原理与设计

姓 名： 袁东琦

学 院： 信息与工程学院

系： 信电系

专 业： 信息工程

学 号： 3200103602

指导教师： 杜歆

2023 年 4 月 9 日

浙江大学实验报告

专业：__信息工程__

姓名：__袁东琦__

学号：__3200103602__

课程名称：__嵌入式系统原理与设计__ 指导老师：__杜歆__ 成绩：__

实验名称：__基于 RK3399ProX 的智能音箱__

一、设计目标

通过使用百度 API 的语音识别和语音合成功能，编写代码设计一个可以通过热词唤醒且具有一定功能的智能音箱。具体为通过录音及语音识别手段，获取关键词，并对关键词做出相应的反馈，如“播放音乐”则智能音箱可以做到自动播放音乐，“明天天气”则可以自动获取天气预报信息并通过语音合成功能将其合成语音播放。

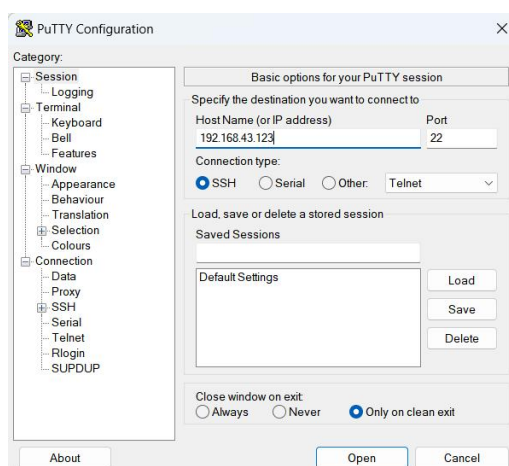
二、设计过程

1) 申请百度 API 账号并创建语音识别和语音合成的应用。并下载相应的 SDK 文件。

应用名称	AppID	API Key	Secret Key	HTTP SDK	创建时间	创建人	操作
Speech_Recognition	31199437	sfsam... 展开 复制	CEory... 展开 复制	语音技术无 ②	2023-03-12 14:18:54	*****1997	报表 更新Secret Key 删除

从中获取到 API Key 和 Secret Key，用于之后的代码调用语音识别和语音合成。

2) 按照文档进行刷机和配置无线网络，配置成功后使用局域网连接 RK3399ProX，IP 地址如下图所示，此方法避免了 putty 窗口中不定时出现大量调试信息干扰我们调试的情况。

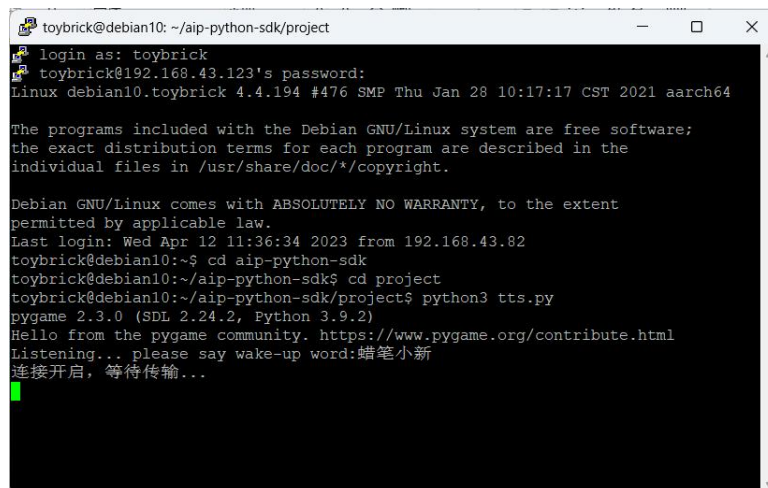


3) 修改百度提供的语音识别和语音合成 demo 代码，使其能够自动读取文件进行语音识别或语音合成，并同时 will tts.py 作为主函数模块，在其中编写主要的逻辑函数 wakeup() 以及 looprecord()。

Wakeup() 函数用于热词唤醒，其中调用 looprecord() 函数。具体逻辑见下图：

```
def wakeup():
    while (1):
        print('Listening... please say wake-up word:蜡笔小新')
        socket_service()
        TEXT = get_text(demoPath)
        if TEXT is not None:
            if '蜡笔小新' in TEXT:
                tts('在的')
                loopRecord()
```

如果一直无法识别唤醒词，程序将始终在这个循环中运行，一旦正确识别唤醒词，则程序自动进入 looprecord() 函数，并在该函数中循环，直到达成退出该函数的条件，退出后程序重新回到 wakeup() 函数的循环中，等待下一次唤醒。实验效果图如下：

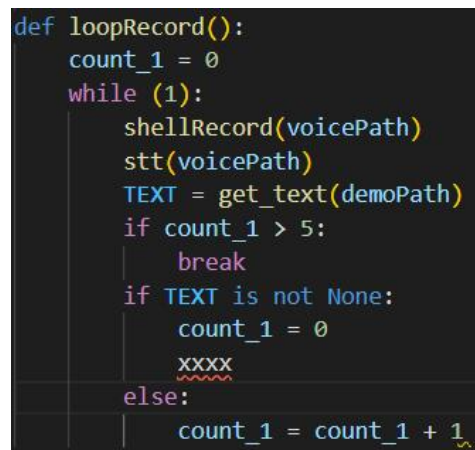


```
toybrick@debian10: ~/aip-python-sdk/project
login as: toybrick
toybrick@192.168.43.123's password:
Linux debian10.toybrick 4.4.194 #476 SMP Thu Jan 28 10:17:17 CST 2021 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

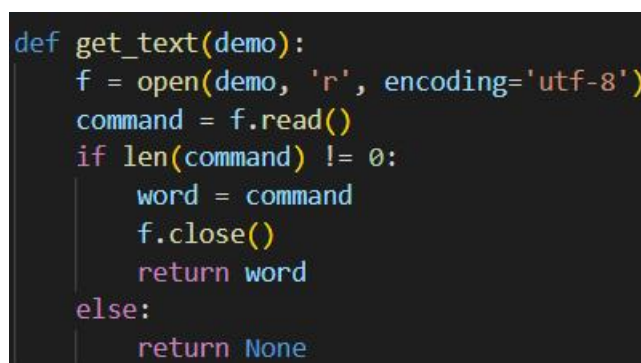
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 12 11:36:34 2023 from 192.168.43.82
toybrick@debian10:~$ cd aip-python-sdk
toybrick@debian10:~/aip-python-sdk$ cd project
toybrick@debian10:~/aip-python-sdk/project$ python3 tts.py
pygame 2.3.0 (SDL 2.24.2, Python 3.9.2)
Hello from the pygame community. https://www.pygame.org/contribute.html
Listening... please say wake-up word:蜡笔小新
连接开启, 等待传输...
```

Looprecord() 函数包含所有对于智能音箱的操作，通过判断语音识别获得的文档中是否包含关键词来做出相应的反应，在一定条件下会自动跳出循环，实现智能音箱自动关机的功能。基本逻辑见下图：



```
def loopRecord():
    count_1 = 0
    while (1):
        shellRecord(voicePath)
        stt(voicePath)
        TEXT = get_text(demoPath)
        if count_1 > 5:
            break
        if TEXT is not None:
            count_1 = 0
            XXXX
        else:
            count_1 = count_1 + 1
```

在实验中我还发现，语音识别的代码有时会在音频文件没有声音的情况下报错，经过检查发现，语音识别的代码默认识别得到的 word 不为 none，因此会出现得到的 demo.txt 文件中内容为 none 的情况，从而导致 tts 时读取文件出现错误。因此我对 get_text 函数进行修改，当读取得到的内容不为零时，正常输出，为零则返回 none。



```
def get_text(demo):
    f = open(demo, 'r', encoding='utf-8')
    command = f.read()
    if len(command) != 0:
        word = command
        f.close()
        return word
    else:
        return None
```

4) 编写 function.py 模块，其中包括天气预报、播放音乐、关闭音乐、调节音量等函数。
天气预报调用天气 API:

5) 编写 record_audio.py 模块, 其中包含两个录音函数, shellRecord_old()函数适用于在 RK3399ProX 上进行录音, shellRecord()函数使用 pyaudio 进行录音, 在 RK3399ProX 调用会报出很多错误, 例如 invalld value for card、no such file or dictionary 等, 会导致录音无法成功进行。但在 shellRecord_old()函数以及 RK3399ProX 自带的 3.5mm 接口进行录音时, 我发现录音获得的结果也不尽人意, 无法正确识别的结果占大多数。在通过播放录音文件后我发现, 录音文件基本没有声音, 在查找 toybrick 开源社区里的相关帖子并使用其中的方法进行尝试后, 还是无法获得理想的录音效果, 具体配置方法见下图。

RK3399Pro入门教程 (15) Debian下音频配置教程

发表于 2020-12-10 17:32:41 查看: 9069 | 回复: 0 | [复制链接] | 只看该作者

本帖最后由 jefferyzhang 于 2020-12-11 10:49 编辑

```
1、RK3399Prod上有三张声卡, 耳机喇叭播放所用的声卡是 card 0
cat /proc/asound/cards
0 [rockchiprk809co]: rockchip_rk809- - rockchip_rk809-codec
rockchip_rk809-codec
1 [rockchiphdmi ]: rockchip_hdmi - rockchip_hdmi
rockchip_hdmi
2 [RKmsm261s4030h0]: RK_msm261s4030h - RK_msm261s4030h0
RK_msm261s4030h0

2、如果要通过耳机播放有两种方式
①、aplay -DHeadphoneNormal sn.wav(HeadphoneNormal 在/usr/share/alsa/alsa.conf有配置)
②、amixer cset numid=1,iface=MIXER,name='Playback Path' 3 && aplay -Dhw:0,0 sn.wav
注意: 如果是喇叭HeadphoneNormal 改成SpeakerNormal, amixer cset numid=1,iface=MIXER,name='Playback Path' 2, 其他类推。
3、如果要通过板载mic录音
①、arecord -DMainMicCapture -r 44100 -f S16_LE -c 2 -d 10 /etc/main.wav
②、amixer cset numid=2,iface=MIXER,name='Capture MIC Path' 1 && arecord -Dhw:0,0 -r44100 -f S16_LE -c 2 -d 10 /etc/main.wav
注意: 如果是耳机mic 录音 MainMicCapture改成FreeMicCapture, amixer cset numid=2,iface=MIXER,name='Capture MIC Path' 2。
4、通过cat /proc/asound/cards 知道默认声卡是card 0, 如果不是, 通过arecord、aplay、amixer配置时, 需要修改对应的声卡号。
5、如果想通过hdmi out播放aplay -Dhw:1,0 sn.wav
```

因此我选择使用在电脑上录音, 然后使用 socket 连接电脑和 RK3399ProX 的方法, 将录音传到板子上, 从而获得了较为理想的语音识别结果。实现效果如下所示:

```
录音开始
录音完毕
{"corpus_no": "7219968501185938070", "err_msg": "success.", "err_no": 0, "result": ["暂停."], "sn": "516972108601681029913"}
暂停。
client filepath: demo.txt
demo.txt 发送成功...
```

6) 通过 winSCP 将代码文件以及音频文件传输到 RK3399ProX, 在 putty 通过命令行安装对应的 python 库文件, 运行 tts.py, 同时在电脑运行 record_audio.py, 录音调试, 程序能够正常运行。

三、总结

这次嵌入式系统的设计从零开始学习了很多东西, 包括 linux 系统基本的命令, python 编写的一些技巧规范, 在刚开始做的时候会有无从下手的感觉, 就很茫然地先申请了百度语音 API 的账号, 在网上找了一些类似的做过语音识别的资料, 看了些资料之后也算有了一点自己的思路。按照文档刷机加配置无线网络还是比较顺利的完成了。之后编写代码文件的过程还算比较顺利, 毕竟在电脑上调试的时候还是不太会报错的, 但是将代码放到板子上后还是会出现一系列问题, 感觉主要原因还是对 linux 系统不是很熟悉, 导致很多时候都做了无用功, 比如一开始总会出现 channel count non available 的问题, 在查找了 toybrick 开源社区里的相关资料后通过输入一些我之前不了解的命令使得问题得到了解决, 因此感觉对于这个系统的开发最重要的还是要熟悉了解各种命令, 不过最后问题解决程序成功运行后还是很有成就感的。做的这个智能音箱总体感觉还是难度不大的, 毕竟基本不依靠额外的硬件设备, 关键基本上还是在写代码上, 理解了智能音箱的基本逻辑之后编写代码还是比较容易的。感觉收获最多的还是真正体验了一遍嵌入式系统开发的流程吧, 从对 linux 系统一无所知, 到学会了一点基本的操作方法, 学会通过修改配置文件来得到自己想要的效果, 最后能够让程序正常在板子上运行起来。