

R818 产品使用说明书

| | | |
|------|------------|------|
| 版本 | 日期 | 修订说明 |
| V1.0 | 2022.04.06 | 创建文档 |

目录

| | |
|--|----|
| 1. 概述..... | 3 |
| 1.1. 功能简介..... | 3 |
| 2. 使用方法简介..... | 3 |
| 2.1. 麦克风初始检测..... | 3 |
| 2.1.1. 使用 Linux 软件 audacity 进行录音初体验..... | 3 |
| 2.1.2. 接收唤醒后返回的唤醒数据。..... | 4 |
| 2.1.3. 命令行查找设备和录音。..... | 4 |
| 3. 开放平台应用及功能库的获取..... | 5 |
| 3.1. 讯飞开放平台应用创建与集成..... | 5 |
| 3.2. 离线命令词识别..... | 7 |
| 3.2.1. 添加离线命令词应用..... | 7 |
| 3.2.2. 集成方法..... | 8 |
| 3.3. AIUI 集成..... | 8 |
| 3.3.1. 配置 AIUI 的功能..... | 8 |
| 3.3.2. 集成方法..... | 9 |
| 4. AIUI 平台库的配置和移植..... | 10 |
| 4.1. 动态库的配置..... | 10 |
| 4.2. AIUI 平台交互..... | 11 |
| 5. 代码运行方法简介..... | 11 |
| 6. 功能包介绍..... | 12 |
| 7. 离线命令词语法设置..... | 15 |
| 8. 离线问答句及串口协议设置..... | 16 |
| 9. 错误码..... | 17 |

1. 概述

1.1. 功能简介

本文档是讯飞环形麦克风阵列语音识别案例。

- 1) 本案例支持在线识别和离线识别,且离线识别优先。
- 2) 离线识别需要用户根据指引设置 **bnf** 语法以及问答库。
- 3) 在线识别会返回识别结果和对应的回答,且回答会直接语音合成进行播报。ROS 程序支持以话题形式发布出去,见话题/question,/answer。
- 4) 离线识别会返回识别结果以及默认回答"好的"。离线的回答设置合成播报,并可以下发协议到用户设置的串口, 需要保证有权限。
- 5) 本案例支持说话人检测, 支持全双工。
- 6) 当麦克风被唤醒时, 会返回唤醒的角度, ROS 程序支持以话题的形式发布, 见话题/angle.

2. 使用方法简介

使用 R1818 唤醒麦克风阵列套件之前, 请首先阅读 R818 麦克风系列开发套件的《产品白皮书》、《产品规格书》、《使用手册》和《协议手册》四个文档来了解其使用方法, 并按照接口要求将麦克风、串口和扬声器的线连接好, 连接好以后阅读一下代码进行使用。

2.1. 麦克风初始检测

R818 麦克风内部集成了科大讯飞的语音算法, 在使用时, 将其串口和电源线插在电脑上便可使用, 麦克风使用串口返回唤醒基本信息(唤醒人, 唤醒角度等), 使用连接到电脑上的 UAC 接口返回音频信息。

2.1.1. 使用 Linux 软件 audacity 进行录音初体验

连接好以后在 linux 平台系统设置 sound 的 input 下多了一个音频输入硬件(XFM-DP-V0.0.18), 默认已经选中此麦克风可以进行操作。在看到已经连接

成功以后，可以安装 **audacity** 软件进行录音和播放进行测试安装命令如下：

```
$ sudo apt-get install audacity
```

安装好软件以后，可以在该软件中完成录音和播放对该设备进行测试。

2.1.2. 接收唤醒后返回的唤醒数据。

在检测完成以后可以连接串口，下载 **linux** 下串口通信工具进行快速体验。

第一步： 将所有连线连接好， 在 **linux** 平台打开 **cutecom** 串口工具。安装命令为：

```
$ sudo apt-get install cutecom.
```

第二步： 配置串口工具设置， 并打开串口。选择识别出来的 **USB 转 TTL** 串口号， 设置波特率： **115200**； 数据位 **8**； 奇偶校验： **无**； 停止位： **1** 流； 控制： **无**。

第三步： **R818** 降噪板 接通电源， 查看串口启动信息。接通电源后， 大概 **1** 分钟左右（**1** 分钟为最长时间，实际小于该时常）， 串口会收到启动信息。

第四步： 体验语言唤醒、 声源定位、 噪声抑制、 回声消除。对着麦克风说“小飞小飞” 这时电路板被唤醒， 串口将输出唤醒信息， 信息中包含使用的唤醒词信息、 唤醒角度信息等。打开 **PC** 机录音软件开始录音， 此时可以从刚才唤醒的方向对着麦克风说话， 也可以从其他的角度对着麦克风说话。也可以在 **PC** 机上播放音乐， 在进行从不同角度对着麦克风说话（注意喇叭不要离麦克风太近， 以免喇叭声音过高造成失真严重）。结束录音， 收听录音内容。输出的为单通道 **16k 16bit pcm** 音频。正常情况， 可以听到唤醒方向的说话声音洪亮清晰， 其他角度声音小且模糊。**PC** 机播放的音乐， 基本听不到。

2.1.3. 命令行查找设备和录音。

1) 查询设备编号：

```
$ arecord -l
```

可以看到终端罗列出了该电脑中的设备， 只需要看到 **card 2: XFM-DP-V0.0.18**, 为该 **R818** 麦克风套装的编号

2) 命令行录音：

```
$ record -D hw:2,0 -r 16000 -f s16_LE -t wav 1.wav
```

生成的录音为 1.wav，默认在 /home 目录下，其中 -D 为设备型号的选择 hw:2,0(card 2,device 0)表示设备型号为 card 2: [XFM-DP-V0.0.18] ,device 0 :USB Audio[USB Audio]，查找到设备型号后要检查代码 Globble.h 文件中的 *pcm_name 型号是否和查询到的型号一致。

3. 开放平台应用及功能库的获取

3.1. 讯飞开放平台应用创建与集成

首先到讯飞开放平台注册账户，然后点击右上角“控制台”，进入自己的控制台平台。如果是首次使用讯飞 开发平台，则控制台中应用是空的，点击创建新应用。

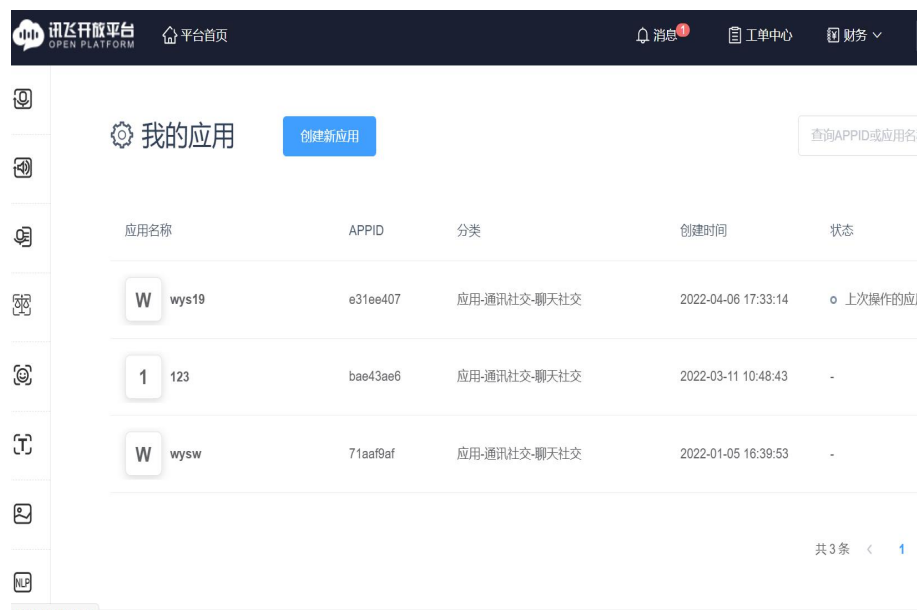


图 1 进入控制台



图 2 创建新应用

点击提交后，可看到用户的应用列表，选择刚刚创建的应用，则进行该应用配置界面，在左侧会有”语音识别”，“语音合成”，“语音扩展”，“人脸识别”，“图像识别”等菜单。

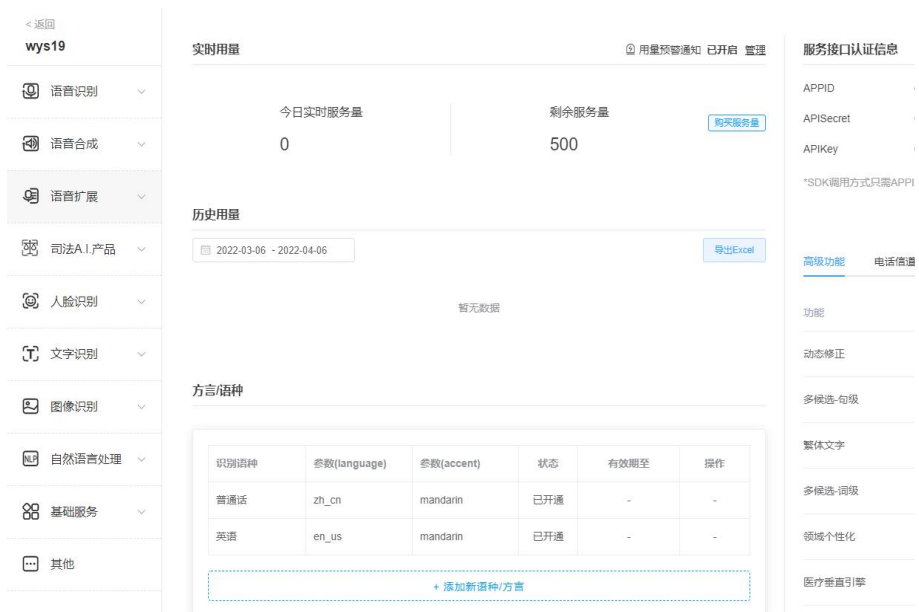


图 3 下载离线命令词识别 SDK

创建完成之后就可以下载 sdk 使用了， 共有两种方式可以选择，如下述所示：

- 1) 在开放平台主页，在菜单中选择 SDK 下载，然后选择应用名称，平台以及要使用的能力，此时下载的 SDK 会将所选的能力打包下载，其中的 lib 下的

libmsc.so 动态库也可看做是打包在一起的，其适用于你勾选的所有能力。

2) 选择单个能力，然后下载单个能力的 SDK 包，此时 SDK 里的动态库仅仅适用于该能力，若下载多个能力，则会有多个 SDK 包，对应多个 libmsc.so 动态库。也可以在每个能力的右下角选择聚合 sdk 的生成，生成的 sdk 同方法一相同。

3.2. 离线命令词识别

3.2.1. 添加离线命令词应用

如图 3 所示，打开语音识别选择离线命令词识别，便可以选择所需平台的离线命令词识别的应用进行下载，linux 平台给出了 x86,x64 版本，如有其他版本的需求，可以联系商务部门进行定制。

The screenshot shows a web interface for the 'Offline Command Word Recognition SDK'. On the left is a sidebar with a menu including 'Voice Recognition', 'Voice Synthesis', and 'Voice Extension'. The 'Voice Recognition' section is expanded, showing 'Offline Command Word Recognition' as a selected option. The main content area is titled 'Real-time Usage' and displays two metrics: 'Today's Real-time Package Volume' (0) and 'Remaining Package Volume' (10). A 'Buy Now' button is visible next to the remaining volume. On the right, there is a table titled 'Offline Command Word Recognition SDK' listing SDKs for different platforms: Android MSC (version 1143), IOS MSC (version 1174), Linux MSC (version 1227), and Windows MSC (version 1126). Below the table, a note states: '讯飞语音SDK也支持多种能力的打包组合，若有需要请前往[聚合SDK](#)下载下载组'.

| SDK名称 | 版本 |
|-------------|------|
| Android MSC | 1143 |
| IOS MSC | 1174 |
| Linux MSC | 1227 |
| Windows MSC | 1126 |

图 4 离线命令词识别 SDK

如果需要离线命令词识别和其他功能共同使用可以点击右下角聚合 SDK 下载，该功能和 3.1 的方式一方法一致。

3.2.2. 集成方法

若在使用离线命令词识别时出现 10102 等错误时，需要替换为自己的 APPID 和离线识别引擎，APPID 我们在创建应用的时候就已得到，离线识别引擎 common.jet 在 bin/msc 中，找到 common.jet 后就可以进行替换了。

1) 替换 sdk：将用户自己的 common.jet 文件替换/speech_comand /config/AIUI/asr/路径下的 common.jet。将用户自己的 appid 替换 speech_command /include 中 Globble.h 文件中的“string offline_appid”。然后再次编译。

3.3. AIUI 集成

3.3.1. 配置 AIUI 的功能

打开图 3 界面最后一个“其他”选项，选择 AIUI 的服务管理进入后配置方法如下：

其中情景模式默认为“main”，可以结合用户场景需求根据使用说明来设置其他情景模式。在这里用户需 要将“语义理解”，“兜底设置”，“语音合成”三个高级设置勾选，在语义理解中用户关注“语义技能”，首先选择“商店技能”，然后“添加商店技能”，你可以在弹出的技能商店中选择你需要的技能，如“天气”，“星座”等，如图 5 所示。点击确定即可。添加完后要选择页面右上角的“保存修改”才可生效。这个时候，就可以使用 你刚添加的技能进行对话了。



图 5 选择所需要的 AIUI 功能

在“兜底测试”中将“图灵机器人”，“讯飞闲聊”等开启后，就可以避免在一些语料未开通时，机器人依然可以答复诸如“这些我还没学会”等等的兜底回复。除此，AIUI 还提供了一些自定义的技能便于用户使用，选择“我的技能”菜单，系统会跳转到技能控制台，然后根据创建方法在该平台创建完技能并发布后，再回到图中 AIUI 应用配置界面，在语音理解-语义技能中的自定义技能中添加刚创建的技能然后“保存修改”就可以了。之后就可以根据自己创建的技能以及技能标识进行交互判断了。当然，关于 AIUI 和其相关的技能应用还有很多，用户可根据官网的文档进行系统的学习，以充分使用科大讯飞开放平台的能力。完成“应用配置”后，用户就可以下载 AIUI 的 SDK 了，选择“开发工具”，可以看到如图 6 所示的界面



图 6 选择 AIUI 所需要的平台

图 6 选择所需要的 SDK 进行下载，linux 平台选择了提供了 x86、x64 版本，如果为其他平台，可联系商务部门进行定制和适配。

3.3.2. 集成方法

若在使用过程中在线人机交互案例时出现”11201”的错误，是因为 appid 选择的 AIUI 应用是免费的，每天交互次数受限，你可根据如下方法进行修改，来使用你创建的“aiui”应用，方法是在“AIUI 开放平台”上找到“APPID”和“APPKEY”，在进行替换的时候只用这两个参数就可以了，替换方法将这两

个参数分别替换 AIUITester.cpp 文件中的 自定义文件 Globble.h 里面的字符串 APPID 以及 APIkey。

合并下载 SDK 进入讯飞开发平台主页，上部菜单栏里就可以看见此项功能，然后选择“SDK”下载 如图 7 上方所示，点击后进入如图 7 界面。

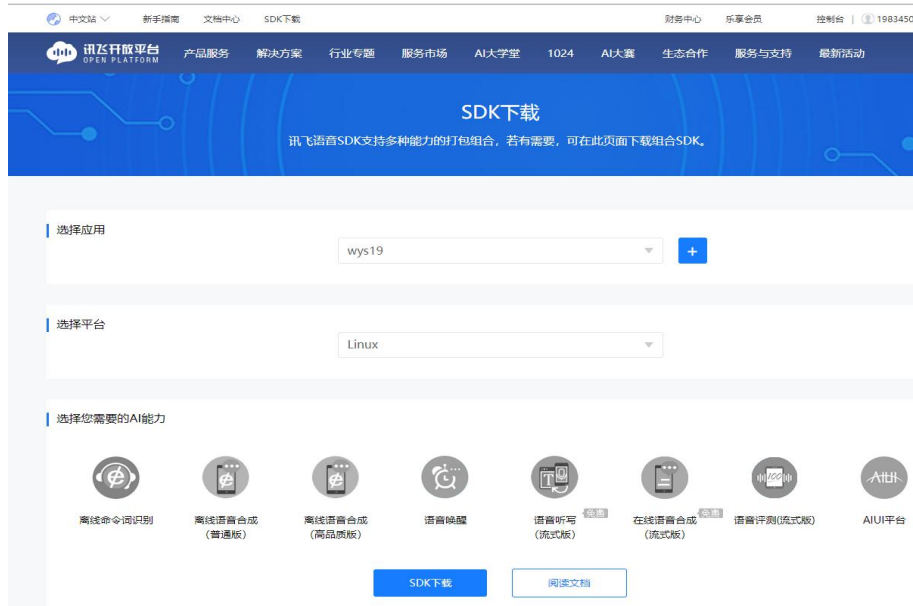


图 7 多种功能 SDK 下载的图示

选择此方式时，应该选择配置好相应功能的应用进行下载，下载好“SDK”以后按照第三章所示的方式进行移植使用。

4. AIUI 平台库的配置和移植

4.1. 动态库的配置

登陆讯飞开放平台（<https://passport.xfyun.cn>），在该网页注册选择进行登陆在主界面选择 SDK 下载，进入后，选择所需要的平台，在此，只讨论在 linux 平台下的使用，选择所需要的 AI 能力，选择完成以后下载所需要的 SDK 进行移植，同时在下载的过程中，也可以点击阅读文档进行

获取所使用的 SDK，并按照要求进行下载和配置，下载后选择相应架构的动态库进行使用。将所需要的库进行下载以后，放置在/speech_command/lib 分配的库中。

放置好后可以在/speech_command/下的 CMakeList.txt 中的相应位置使用

Tatget_link_librariles()链接所需要使用的库，并在第 133 行，选择自己所需要的平台中的库。

4.2. AIUI 平台交互

在完成初始功能的体验后，在使用 AIUI 交互平台时需要使用，麦克风返回的音频流，所以在这里使用 linux 下的 alsa 进行底层录音，在使用 Linux 底层获取音频流时，在使用的过程中，将获取的音频流传入到 AIUI 中，在使用 AIUI 的过程中，首先配置/config/AIUI/aiui.cfg 中将文件路径改为自己所使用的文件路径。

路径修改完毕以后，可以修改/speech_command/include/globle.h 中的在线 AIUI 的 APPID 进行修改使用。

5. 代码运行方法简介

支持两种方式运行：

****方式一****：在终端使用 roslaunch 指令执行：

...

```
$ roslaunch speech_command speech-command.launch
```

...

****方式二****：在终端使用 rosrund 指令执行：

...

```
$ roscore
```

```
$ rosrund speech_command speech_command_node
```

...

系统检测到串口和麦克风阵列后，会自动开机并播报"AIUI 启动成功"，然后等待用户唤醒，用户说出"小飞小飞"即可使用。唤醒后即进入交互状态，本案例默认提供的交互方式是 continues 方式，即一次唤醒多次交互的方式，用户在该方式下可以体验到中途打断的功能，在它回答问题中，可以接着说出下一个问题，这时会自动回答下一个问题。值得注意的是，服务端保存用户交互历史的时间最长为 120s，即如果用户说"明天天气怎么样"，接下来说"后天呢？"，会默认

回答后天天气状况，多轮对话最长时间为 120s，值得注意的是，AIUI 引擎在长时间无有效交互时会进入休眠模式，此时需要重新唤醒进行交互。此时间默认为 180s，可在 aiui.cfg 中进行修改。在正常交互过程中事件的状态包含三种：

- 1) "tts"：即文字到语音的转化
- 2) "asr"：离线语音识别
- 3) "nlp"：在线语义理解

当用户使用技能命令词“你去休息吧”，事件的状态标识为“nlp”，此时麦克风会进入休眠模式，不会进行交互。若要重新进入交互模式需要再次使用“小飞小飞”进行唤醒。

特别地，可以按“ctrl +C”键来退出本程序。

6. 功能包介绍

目录如下：

...

其目录结构如下：

```

├── audio
│   └── sound
├── bin
│   ├── AIUI
│   ├── config
│   ├── msc
│   └── tts.pcm
├── CMakeLists.txt
├── config
│   ├── AIUI
│   ├── call.bnf
│   ├── gm_continuous_digit.abnf
│   ├── offline_QA.txt
│   └── output.txt

```

- | └─ userwords.txt
- |─ include
 - | └─ aiui
 - | └─ AIUITester.h
 - | └─ AudioPlayer.h
 - | └─ AudioRecorder.h
 - | └─ cJSON.h
 - | └─ cJSON_Utils.h
 - | └─ FileUtil.h
 - | └─ formats.h
 - | └─ Global.h
 - | └─ hidapi.h
 - | └─ json
 - | └─ jsoncpp-master
 - | └─ libusb.h
 - | └─ linuxrec.h
 - | └─ msp_cmn.h
 - | └─ msp_errors.h
 - | └─ msp_types.h
 - | └─ protocol_proc_unit.h
 - | └─ qise.h
 - | └─ qisr.h
 - | └─ qtts.h
 - | └─ queue_internal.h
 - | └─ queue_simple.h
 - | └─ RingBuffer.h
 - | └─ speech_recognizer.h
 - | └─ TestListener.h
 - | └─ WriteAudioThread.h
- |─ launch

```
|   └─ speed-command.launch
|── lib
|   └─ arm32
|   └─ arm64
|   └─ x64
|   └─ x86
|── package.xml
|── readme.md
|── src
|   └─ aiuiMain.cpp
|   └─ AIUITester.cpp
|   └─ AudioPlayer.cpp
|   └─ AudioRecorder.cpp
|   └─ FileUtil.cpp
|   └─ TestListener.cpp
|   └─ WriteAudioThread.cpp
└─ tmp
    └─ all.pcm
    └─ config.txt
    └─ in.pcm
    └─ system.tar
```

...

其中：

- 1) **audio**: 用于存放录制的音频文件，音频文件的命名是自定义的，文件中设置了自己的离线音频，该音频可由网站生成（<https://www.iflyos.cn/tts-file>），放置到此文件夹下进行使用。
- 2) **bin**: 用于存放可执行文件，**call.bnf** 为自定义的离线命令词识别语法，可根据自己实际场景进行更改，同时也放置了，**offline_QA.txt** 文档用于自定义离线语法和需要播报语音所存放的路径，离线语法要和 **call.bnf** 同步并且需要一致。

3) **include**: 包含麦克风阵列启动、给定案例中需要的头文件。其中 **Global.h** 为用户接口, 可以根据需要进行修改。

4) **lib**: 包含麦克风阵列启动、给定案例中需要的动态库文件, 为了兼容不同的平台, 给定了 ARM64、x64 系统、x86 系统的动态库。

5) **launch**: 启动文件。

6) **src**: **aiuiMain.cpp** 为主节点, 也是识别的外围, 负责与外通信。

AIUITester.cpp 为语音识别的实际执行节点, 里面获取麦克风阵列降噪音频送到引擎中, 并返回识别结果。

6) **tmp**: 包含文件麦克风阵列板所需的资源文件, 常态下可设为隐藏文件, 用户可忽略。

7. 离线命令词语法设置

本例程中希望可以离线识别的命令词为"向前走"、"向后走"、"向左走"、"向右走"。基于此设置的离线语法文件见/config/call.bnf 所示:

```
...  
#BNF+IAT 1.0 UTF-8;  
!grammar call;  
!slot <want>;  
!slot <dialpre>;  
!slot <dialsuf>;  
!slot <contact>;  
!start <callstart>;  
<callstart>:<want>;  
<want>:向前走|向后走|向左走|向右走;  
...
```

此 bnf 比较简单, 没有设置隐藏字段, 在交互时用户说的话必须包括命令词才可以识别。若用户交互的话中仅仅包括"前走", 则离线引擎是拒绝识别的。但当话语中包括"向前走", 才可以被离线引擎识别。

假如想"前走"也可以识别, 或想"前走"、"向前走"、"往前走"、"你往前走"、

"你往前移动"、"向前动动"等都被识别且表达一个意思，可丰富 bnf 语法如下所示：

```

```
#BNF+IAT 1.0 UTF-8;
```

```
!grammar call;
```

```
!slot <want>;
```

```
!slot <direction>;
```

```
!slot <do>;
```

```
!slot <what>;
```

```
!start <callstart>;
```

```
<callstart>:[<want>]<dowhat>;
```

```
<want>:向|往|你往|你向;
```

```
<dowhat>:<direction><do>;
```

```
<direction>:左!id(10001)|右!id(10001)|前!id(10001)|后!id(10001);
```

```
<do>:走!id(10001)|移动!id(10001)|转!id(10001)|动!id(10001)|动动!id(10001);
```

```

该 bnf 语法包含命令词"【你】【往、向】【前、后、左、右】【走、转】"，用户说出这四组关键字中任一组合均可识别，如“往前走”、“向后走”、“向左转”、“左转”等。可以发现通过修改该文件，说话时就不会太受限制。

7) 离线命令词识别语料需要根据用户的实际应用场景来设置,以上只是给出了简单的例子，故在使用的过程中，可以在讯飞开放平台 sdk 下载选择离线命令词识别中参考语法开发指南中的《BNF 语法开发指南》来进行深层次的学习完成所需语法的创建和使用。

8. 离线问答句及串口协议设置

在本例中,路径/config/offline_QA.txt 下存放的是离线引擎识别到的语句与串口协议的对应关系。如下：

```

```
向左走|往左走|左走:AA0200FF:/audio/offline_left.mp3
```



向前走|往前走|前走:AA0100FF:/audio/offline\_left.mp3

向右走|往右走|右走:AA0300FF:/audio/offline\_left.mp3

向后走|往后走|后走:AA0400FF:/audio/offline\_left.mp3

...

说明:

- 1) ":"用于分割识别到的语句、串口协议和需要播报语音的路径。
- 2) "|"用于分割多个识别到的语句,即"向前走"和"往前走"均对应同一个协议。
- 3) 本例中协议字头是"AA",尾部是"FF",中间字段来区分协议。协议格式可以根据需要任意修改。
- 4) 协议会被当做 string 下发到用户在 Global.h 中设置的串口中。
- 5) 地址自定义可以任意放在 speech\_command 路径下面的任意文件夹下,但是建议只修改后面的文件名字,默认文件夹为 audio 目录下,在这里采用文件中设置了自己的离线音频,该音频可由网站生成 (<https://www.iflyos.cn/tts-file>), 放置到此文件夹下进行使用。

## 9. 错误码

### 11201

...

若在使用过程中,可能会出现不能交互,且报错代码为 11201 的情况,原因是本案例绑定的 appid 是个人的,单日的交互次数受限,用户可以替换为自己的 appid,用户可通过修改 Global.h 文件中的自定义的字符串 appid 以及 key,分别对应 id 和密码。然后开通常用的语料和技能进行使用,在后续使用中若再次遇到报错代码为 11201 的情况,用户可前往控制台检查对应 appid 次数是否超过限制次数,超过可在官网产品页面领取免费包或者购买套餐包,未超过可提交技术工单。离线服务接口报错其他常见错误可见 <https://www.xfyun.cn/document/error-code/>。

...

### 找不到麦克风设备, 请重新插拔

...

该现象表示系统未检测到麦克风。

'''

### 无法打开音频设备，请重新插拔

'''

该现象表示系统已经占用麦克风设备。

'''

### “在启动后出现 libcjson.so.1 不存在”

'''

需要安装 cJSON，且需要将/usr/local/lib 中的 libcjson.so.1 移动到/usr/lib 下。

'''

### “23300”或 “bnf\_recognise\_error”

'''

1.离线命令词识别时语法 bnf 有误，请检查标点符号，槽定义等是否有误，可参考 bnf 编写说明书。

2.配置文件 aiui.cfg 中配置的情景模式参数 scene 使用测试环境 xxx\_box :如果在配置文件中使用了测试环境 xxx\_box ,那么将受到如下报错,原因为离线语法构建不支持下划线.

'''

### “10407”或“unfit\_appid\_and\_lib\_error”

'''

appid 与动态库 libmsc.so 不匹配，可参考 4.2.2 进行修改，在修改后一定要编译后再运行.

'''

### “10102”或“build\_grammer\_error”

'''

语法资源文件 common.jet 路径错误。如果工程中配置 common.jet 路径错误或资源文件不存在，为避免出错，建议修改为绝对路径；

'''

### “11212”或“low\_confidence error or 11212\_license\_expired\_error”

'''

---

离线命令词识别过程中，离线授权体验期过期，需重新购买，也可重新注册讯飞开放平台获取新 **appid** 及资源文件。

...

### “11210”

...

**appid** 与离线资源即 **common.jet** 不匹配。

...

### “11201”

...

在线业务超次数或离线资源装机量超限。如离线命令词识别中，一个试用版的 **appid** 和资源在超过 10 台机器上使用。

...

### “文件无法打开...”

...

在执行案例中，若在录音时出现文件无法打开的问题，需确认音频文件是否有读写的权限，可删除该录音文件后再运行，或者使用超级用户权限。

...

### “在启动例程时一直显示麦克风未启动”

...

请检查是否有开启录音后，未停止录音的操作。若有，则需要手动停止录音或者断开与麦克风的连接来结束通信。

...

### build grammar error, errcode = -1 或 error reason is grammar content is null

...

离线语法 **xxx.bnf** 文件路径错误，如果工程中配置 **xxx.bnf** 路径错误或语法文件不存在

...

### build grammar error, errcode = 10111 或 error reason is build grammar fail

...

代码中构建语法前未初始化 MSC(调用 MSPLogin 接口)。如果没有在代码中初始化 AIUI 前进行 msc sdk 初始化,那么在调用构面语法的时候将会受到下面的错误信息

```

build grammar error, errcode = -1 或 error reason is ASR not start

```

配置文件 aiui.cfg 中未修改交互模式 intent\_engine\_type 为 local 或 mixed:如果在交互模式 intent\_engine\_type=cloud 下进行语法构建。

```

其他能力应用错误

```

可参照 <https://shimo.im/sheet/w3yUy39uNKs0J7DT> 进行修改。

```