

# ROS理论与实践

## —— 第8讲：机器人语音交互



主讲人 胡春旭






机器人博客“古月居”博主

《ROS机器人开发实践》作者

武汉精锋微控科技有限公司 联合创始人

华中科技大学 人工智能与自动化学院 硕士



-  1. 讯飞开放平台使用简介
-  2. ROS语音识别与语音输出
-  3. ROS机器人语音交互



# 1. 讯飞开放平台使用简介



# 1. 讯飞开放平台使用简介



产品 ▾ 解决方案 ▾ 资料库 ▾ iFLYOS ▾ 服务市场 ▾ AI大学 AI大赛 <sup>HOT</sup> 1024 生态平台 ▾

EN 登录 | 注册

让世界享受 **A.I.** 的乐趣

• LET THE WORLD ENJOY THE FUN OF AI •

<https://www.xfyun.cn/>



# 1. 讯飞开放平台使用简介





讯飞开放平台  
OPEN PLATFORM

 平台首页



 我的应用 > my\_ros\_voice

\* 应用名称

my\_ros\_voice

\* 应用分类

智能硬件-机器人-服务机器人

\* 应用功能描述

ROS语音交互

提交

返回我的应用

创建应用



## 语音听写（流式版） SDK

SDK名称	版本	操作	
Android MSC	1138	下载	文档
iOS MSC	1174	下载	文档
Linux MSC	1126	下载	文档
Windows MSC	1126	下载	文档
Java MSC	4.0	下载	文档



SDK下载

SDK在集成开发测试阶段服务量限制为最高500次/每日，完成提测申请或商务合作后可以解除。

选择应用

my\_ros\_voice

+

选择平台

Linux

选择您需要的AI能力

 离线命令识别

 离线语音合成

 语音唤醒

 语音听写 (流式版)

 在线语音合成 (流式版)

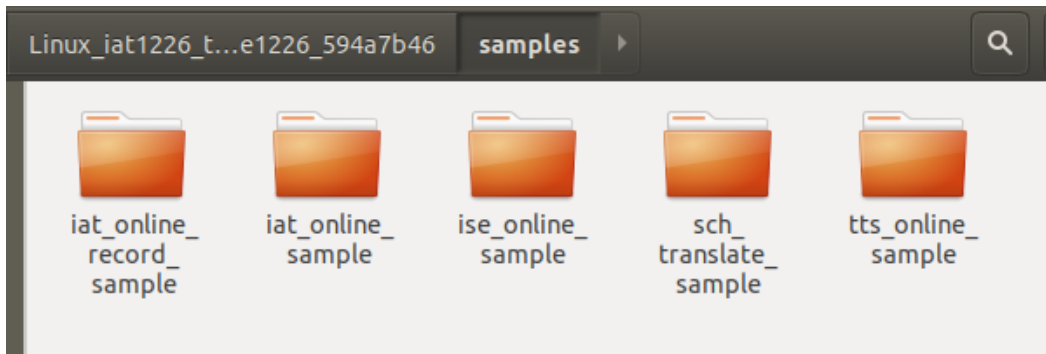
 语音评测

 AIUI平台

SDK下载

阅读文档

下载SDK



讯飞SDK的samples

## 拷贝并安装库

```
$ sudo cp libmsec.so /usr/lib/
```

```
$ sudo apt install sox
```

```
$ sudo apt install libsox-fmt-all
```

```
hcx@hcx-pc:~/Linux_iat1226_tts_online1226_594a7b46/bin$ ./iat_online_record_sample
Want to upload the user words ?
0: No.
1: Yes
0
Where the audio comes from?
0: From a audio file.
1: From microphone.
1
Demo recognizing the speech from microphone
Speak in 15 seconds
Start Listening...
Result: [ 你好, 欢迎学习机器人课程。 ]

Speaking done
Not started or already stopped.
15 sec passed
```

[illegible]

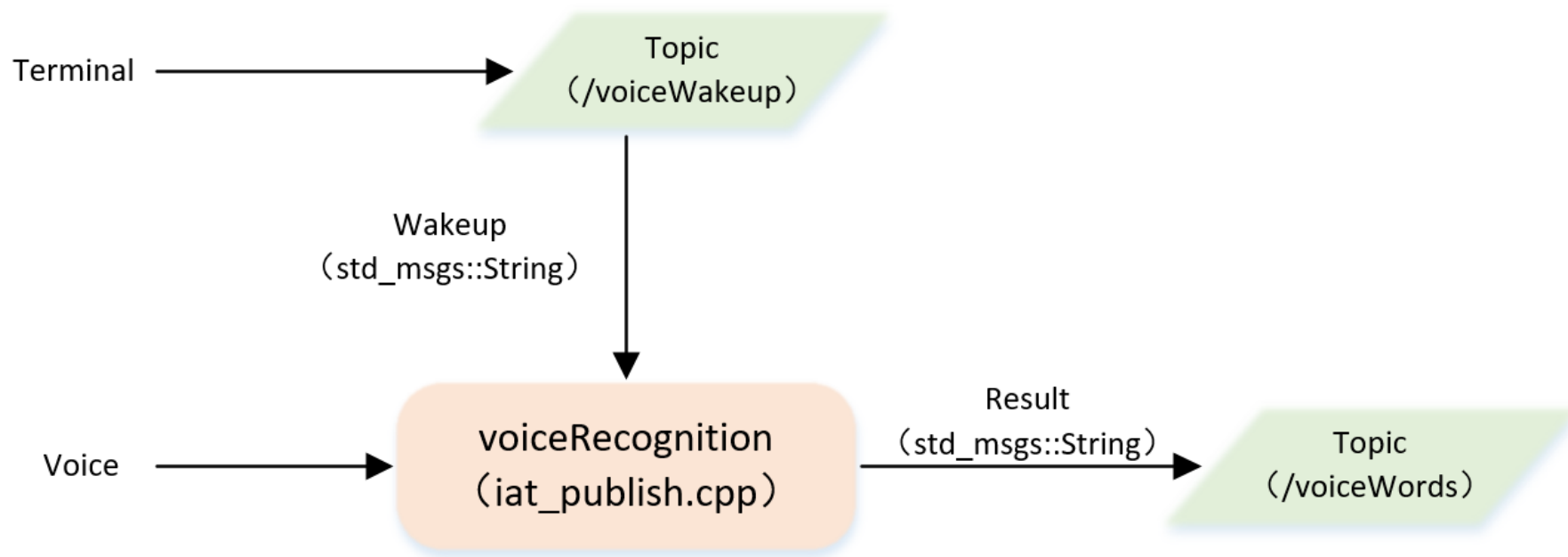
\* 科大讯飞的SDK带有ID号，每个人每次下载后的ID都不相同，更换SDK之后需要修改代码中的APPID。你也可以直接使用本课程的libmsc.so文件，否则需要将源码中的APPID修改为自己下载SDK中的ID。



## 2. ROS语音识别与语音输出



## 2. ROS语音识别与语音输出



语音识别功能框图





## 2. ROS语音识别与语音输出

- subscriber: 接收唤醒信号, 将wakeupFlag变量置位;
- publisher: 主循环中调用SDK的语音识别功能, 识别成功后置位resultFlag变量, 将识别出来的字符串发布。

robot\_voice/src/iat\_publish.cpp

```
int main(int argc, char* argv[])
{
    // 初始化ROS
    ros::init(argc, argv, "voiceRecognition");
    ros::NodeHandle n;
    ros::Rate loop_rate(10);

    // 声明Publisher和Subscriber
    // 订阅唤醒语音识别的信号
    ros::Subscriber wakeUpSub = n.subscribe("voiceWakeup", 1000, WakeUp);
    // 订阅唤醒语音识别的信号
    ros::Publisher voiceWordsPub = n.advertise<std_msgs::String>("voiceWords", 1000);

    ROS_INFO("Sleeping...");
    int count=0;

    while(ros::ok())
    {
        // 语音识别唤醒
        if(wakeupFlag)
        {
            ROS_INFO("Wakeup...");

            printf("Demo recognizing the speech from microphone\n");
            printf("Speak in 8 seconds\n");

            demo_mic(session_begin_params);

            printf("8 sec passed\n");

            wakeupFlag=0;
        }

        // 语音识别完成
        if(resultFlag){
            resultFlag=0;
            std_msgs::String msg;
            msg.data = g_result;
            voiceWordsPub.publish(msg);
        }

        ros::spinOnce();
        loop_rate.sleep();
        count++;
    }
    exit:
    MSPLogout(); // Logout...

    return 0;
}
```



## 2. ROS语音识别与语音输出

在CMakeLists.txt中加入编译规则

```
add_executable(iat_publish
  src/iat_publish.cpp
  src/speech_recognizer.c
  src/linuxrec.c)
target_link_libraries(
  iat_publish
  ${catkin_LIBRARIES}
  libmsc.so -ldl -lpthread -lm -lrt -lasound
)
```

\* 编译过程需要链接SDK中的libmsc.so库，须提前将此库拷贝到系统路径/usr/lib/



## 2. ROS语音识别与语音输出

### 语音识别示例

```
$ roscore
```

```
$ rosrun robot_voice iat_publish
```

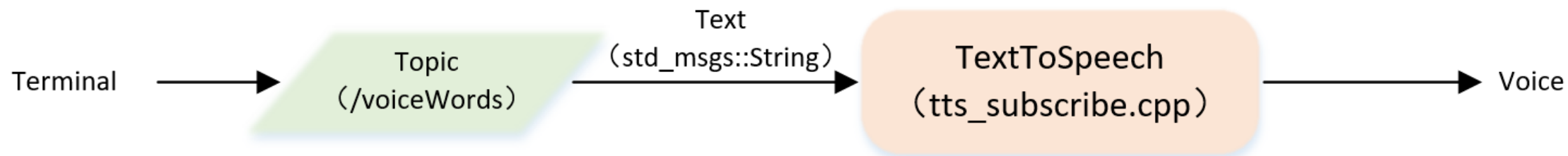
```
$ rostopic pub /voiceWakeup std_msgs/String "data: 'any string'"
```

```
hcx@hcx-pc:~$ rosrun robot_voice iat_publish
[ INFO] [1566368910.649395424]: Sleeping...
waking up
[ INFO] [1566368917.248451264]: Wakeup...
Demo recognizing the speech from microphone
Speak in 8 seconds
Start Listening...
Result: [ 欢迎来到机器人的世界。 ]

Speaking done
Not started or already stopped.
8 sec passed
```



## 2. ROS语音识别与语音输出



语音输出功能框图



## 2. ROS语音识别与语音输出

- subscriber：订阅voiceWords话题，接收输入的字符串。
- voiceWordsCallback：使用SDK接口将字符串转换成中文语音。

```
void voiceWordsCallback(const std_msgs::String::ConstPtr& msg)
{
    char cmd[2000];
    const char* text;
    int ret = MSP_SUCCESS;
    const char* session_begin_params = "voice_name = xiaoyan, text_encoding = utf8, sample_rate = 16000, speed = 50, volume = 50, pitch = 50, rdn = 2";
    const char* filename = "tts_sample.wav"; //合成的语音文件名称

    std::cout<<"I heard : "<<msg->data.c_str()<<std::endl;
    text = msg->data.c_str();

    /* 文本合成 */
    printf("开始合成 ...\n");
    ret = text_to_speech(text, filename, session_begin_params);
    if (MSP_SUCCESS != ret)
    {
        printf("text_to_speech failed, error code: %d.\n", ret);
    }
    printf("合成完毕\n");

    popen("play tts_sample.wav", "r");
    sleep(1);
}
```

robot\_voice/src/tts\_subscribe.cpp



## 2. ROS语音识别与语音输出

在CMakeLists.txt中加入编译规则

```
add_executable(tts_subscribe src/tts_subscribe.cpp)
target_link_libraries(
  tts_subscribe
  ${catkin_LIBRARIES}
  libmsc.so -ldl -pthread
)
```

\* 编译过程需要链接SDK中的libmsc.so库，须提前将此库拷贝到系统路径/usr/lib/



## 2. ROS语音识别与语音输出

### 语音输出示例

```
$ roscore
```

```
$ rosrun robot_voice tts_subscribe
```

```
$ rostopic pub /voiceWords std_msgs/String "data: '你好，我是机器人'"
```

```
hcx@hcx-pc:~$ rosrun robot_voice tts_subscribe

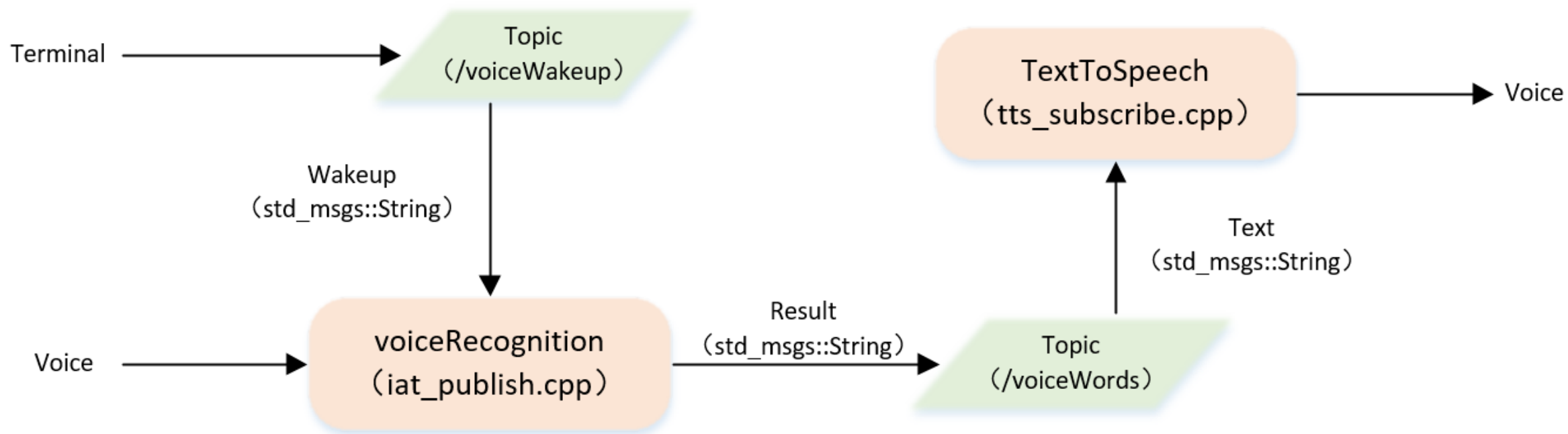
#####
## 语音合成 (Text To Speech, TTS) 技术能够自动将任意文字实时转换为连续的 ##
## 自然语音，是一种能够在任何时间、任何地点，向任何人提供语音信息服务的 ##
## 高效便捷手段，非常符合信息时代海量数据、动态更新和个性化查询的需求。 ##
#####

I heard :你好，我是机器人
开始合成 ...
正在合成 ...
>>>>
合成完毕

tts_sample.wav:
```



## 2. ROS语音识别与语音输出



语音识别与语音输出功能框图





## 2. ROS语音识别与语音输出

### 将语音输入与语音输出结合

```
$ roslaunch robot_voice repeat_voice.launch
```

```
$ rostopic pub /voiceWakeup std_msgs/String "data: 'any string'"
```

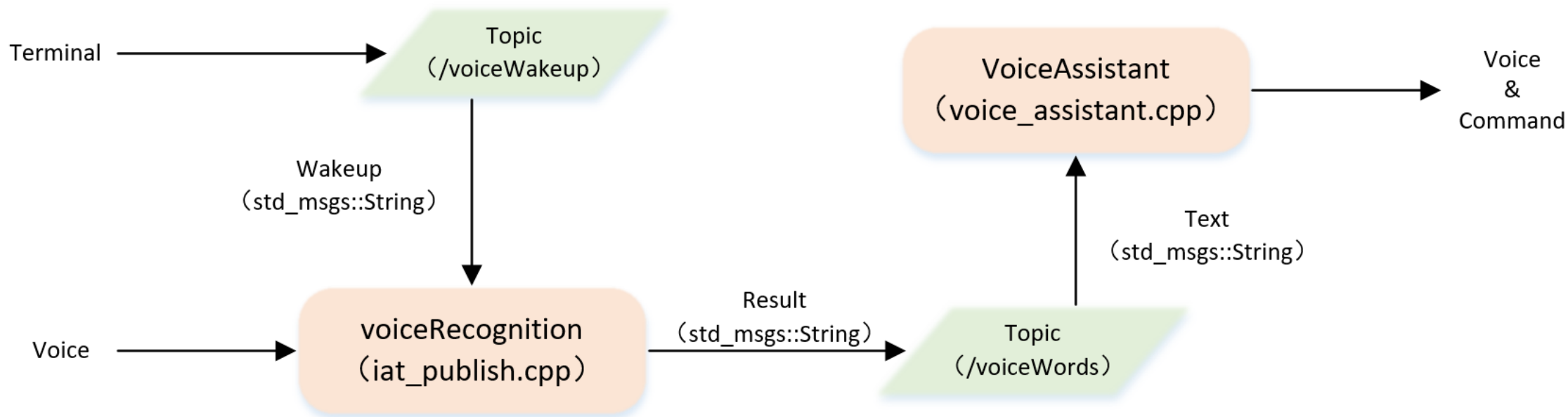
```
#####  
## 语音合成 (Text To Speech, TTS) 技术能够自动将任意文字实时转换为连续的 ##  
## 自然语音, 是一种能够在任何时间、任何地点, 向任何人提供语音信息服务的 ##  
## 高效便捷手段, 非常符合信息时代海量数据、动态更新和个性化查询的需求。 ##  
#####  
  
waking up  
[ INFO] [1566369187.838221586]: Wakeup...  
Demo recognizing the speech from microphone  
Speak in 8 seconds  
Start Listening...  
Result: [ 你好, 我是机器人。 ]  
  
Speaking done  
Not started or already stopped.  
8 sec passed  
I heard :你好, 我是机器人。  
开始合成 ...  
正在合成 ...  
>>>>  
合成完毕
```



## 3. ROS机器人语音交互



### 3. ROS机器人语音交互



语音交互功能框图



## 3. ROS机器人语音交互

```
std::string dataString = msg->data;
if(dataString.find("你是谁") != std::string::npos
|| dataString.find("名字") != std::string::npos)
{
    char nameString[100] = "我是你的语音小助手，你可以叫我小R";
    text = nameString;
    std::cout<<text<<std::endl;
}
else if(dataString.find("你几岁了") != std::string::npos
|| dataString.find("年龄") != std::string::npos)
{
    char eageString[100] = "我已经四岁了，不再是两岁的小孩子了";
    text = eageString;
    std::cout<<text<<std::endl;
}
else if(dataString.find("你可以做什么") != std::string::npos
|| dataString.find("干什么") != std::string::npos)
{
    char helpString[100] = "你可以问我现在时间";
    text = helpString;
    std::cout<<text<<std::endl;
}
else if(dataString.find("时间") != std::string::npos)
{
    //获取当前时间
    struct tm *ptm;
    long ts;

    ts = time(NULL);
    ptm = localtime(&ts);
    std::string string = "现在时间" + to_string(ptm->tm_hour) + "点" + to_string(ptm->tm_min) + "分";

    char timeString[40] = {0};
    string.copy(timeString, sizeof(string), 0);
    text = timeString;
    std::cout<<text<<std::endl;
}
else
{
    text = msg->data.c_str();
}
```

- subscriber: 订阅voiceWords话题，接收输入的字符串。
- voiceWordsCallback: 通过if判断关键词，完成语音回复。



### 3. ROS机器人语音交互

在CMakeLists.txt中加入编译规则

```
add_executable(voice_assistant src/voice_assistant.cpp)
target_link_libraries(
    voice_assistant
    ${catkin_LIBRARIES}
    libmsc.so -ldl -pthread
)
```

\* 编译过程需要链接SDK中的libmsc.so库，须提前将此库拷贝到系统路径/usr/lib/



### 3. ROS机器人语音交互

#### 智能语音助手示例

```
$ roslaunch robot_voice voice_assistant.launch
```

```
$ rostopic pub /voiceWakeup std_msgs/String "data: 'any string'"
```

```
[ INFO] [1566369259.535482648]: Wakeup...  
Demo recognizing the speech from microphone  
Speak in 8 seconds  
Start Listening...  
Result: [ 你是谁? ]
```

```
Speaking done  
Not started or already stopped.  
8 sec passed  
I heard :你是谁?  
我是你的语音小助手, 你可以叫我小R  
开始合成 ...  
正在合成 ...  
>>>>>  
合成完毕
```

```
[ INFO] [1566369292.699814275]: Wakeup...  
Demo recognizing the speech from microphone  
Speak in 8 seconds  
Start Listening...  
Result: [ 你几岁了? ]
```

```
Speaking done  
Not started or already stopped.  
8 sec passed  
I heard :你几岁了?  
我已经四岁了, 不再是两岁的小孩子了  
开始合成 ...  
正在合成 ...  
>>>>>  
合成完毕
```

```
[ INFO] [1566369324.064703842]: Wakeup...  
Demo recognizing the speech from microphone  
Speak in 8 seconds  
Start Listening...  
Result: [ 现在时间。 ]
```

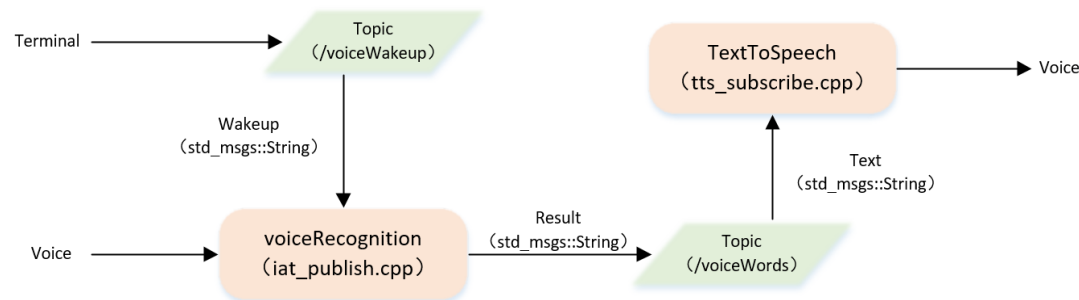
```
Speaking done  
Not started or already stopped.  
8 sec passed  
I heard :现在时间。  
现在时间14点35分  
开始合成 ...  
正在合成 ...  
>>>  
合成完毕
```



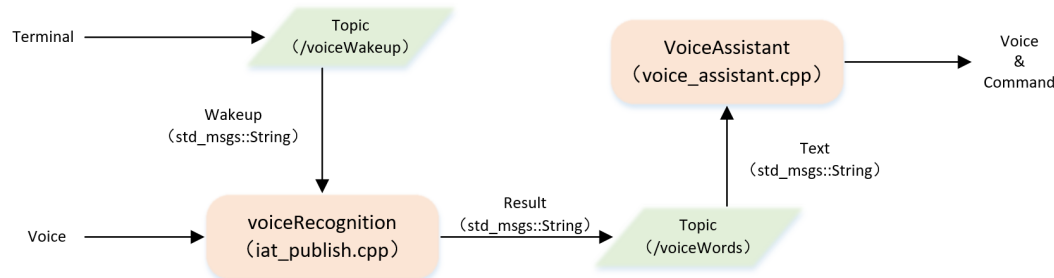
## 讯飞开放平台使用简介

- 注册应用、下载SDK
- SDK的基本使用方法

## ROS语音识别与语音输出



## ROS机器人语音交互





## 1. 根据本讲内容及源码，实现语音控制机器人运动场景：

- 通过“向前”、“向后”、“向左”、“向右”、“停止”等语音命令，控制Gazebo中仿真机器人运动；
- 收到命令后，语音输出一段话，例如：“太阳当空照，花儿对我笑”；
- 可在以上基础功能上添加更多创意功能。





- 讯飞开放平台  
<http://www.xfyun.cn/>
- 百度AI开放平台  
<http://ai.baidu.com/>
- ROS探索总结（二十八）——机器听觉  
<http://www.guyuehome.com/514>
- 《ROS机器人开发实践》  
第八章

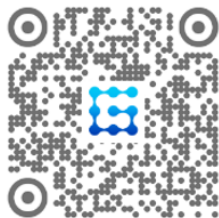




# Thank You

怕什么真理无穷，进一寸有一寸的欢喜

更多精彩，欢迎关注



 古月居



 古月春旭