

人机交互与人机接口技术实践（语音部分）

基于科大讯飞SDK与ROS多机通信的语音导航系统设计

东大人机协作实验室

主讲人：张兴

2020年05月

目录 >

CONTENTS

01

硬件平台

02

实验原理

03

实验内容

04

拓展应用

目录 >

CONTENTS

01

硬件平台

02

实验原理

03

实验内容

04

拓展应用

硬件平台

➤ 科大讯飞麦克风阵列模块（XFM10621）

- 实物介绍及包装清单

XFM10621评估板搭载XFM10621核心模块，外置UART和I²C通信接口以及Line out和I²S音频输出接口，可供开发者快速体验XFM10621模块远场拾音、声源定位、回声消除、拾音模式切换、唤醒效果监测等各项功能。

包装清单

编号	类型	数量
1	评估板卡	1块
2	LED麦克板卡	1块
3	USB电源线	1根
4	USB转串口数据线	1根
5	录音线	1根
6	评估板说明书	1个
7	5v电压稳定电源	1个

硬件平台

► 科大讯飞麦克风阵列模块 (XFM10621)

• 硬件连接

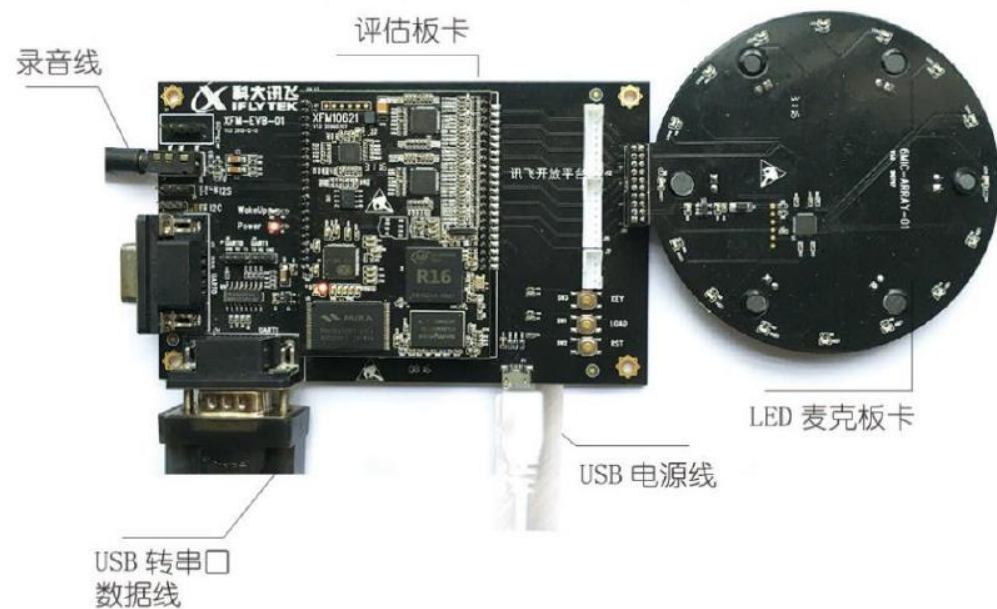
第一步：把环形LED麦克板卡连接到评估板卡上；

第二步：USB转串口数据线的串口端连到评估板卡的UART1上，另一端USB端连接到PC的USB接口上；

第三步：使用USB电源线连接到评估板卡microUSB口上进行供电；

注意：不能使用电脑USB接口直接供电，请使用5V电压稳定电源供电，如：手机充电器等。

第四步：把耳机（自备）插入音频插座上。



硬件平台

➤ 科大讯飞麦克风阵列模块 (XFM10621)

- 快速体验

第一步：打开PC上的SecureCRT工具，并配置串口参数。串口通信参数设置如下：波特率115200；数据位8；奇偶校验位无；停止位1；流控无。

第二步：体验语音唤醒、声源定位、噪声抑制等。说出唤醒词“灵犀灵犀”，这时评估板卡上的WakeUp指示灯闪烁一次，LED麦克板卡上唤醒方向的灯被点亮。

```
-----
WAKE UP!angle:303 score:1371 key_word: lingxilingxi
```

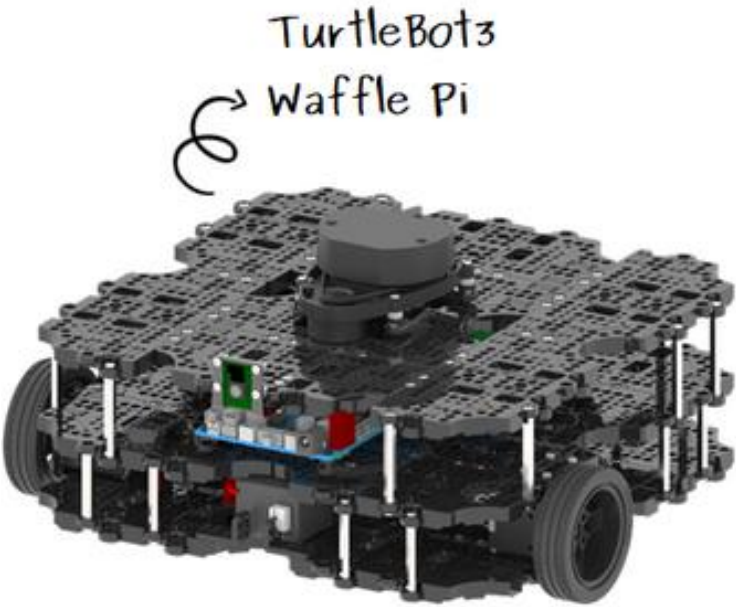
```
##### IFLYTEK XFM10621 #####
##### APP_VER:1.0.7.1012 #####
##### LIB_VER:2.0.5.1020 #####
##### XFM10621 CMD List: #####
CMD:VER
CMD:RESET
CMD:BEAM      usage: BEAM num[0~5]
CMD:CALL_MODE usage: CALL_MODE num[0-1]
CMD:WAKEUP    usage: WAKEUP num[0-1]
```

```
Enter your CMD
```

SecureCRT工具界面显示

硬件平台

➤ 移动机器人Turtlebot3 (Waffle Pi)



Items	Burger	Waffle (Discontinued)	Waffle Pi
Maximum translational velocity最大直行速度	0.22 m/s	0.26 m/s	0.26 m/s
Maximum rotational velocity最大旋转速度	2.84 rad/s (162.72 deg/s)	1.82 rad/s (104.27 deg/s)	1.82 rad/s (104.27 deg/s)
Maximum payload最大负载	15kg	30kg	30kg
Size (L x W x H) 尺寸 长宽高	138mm x 178mm x 192mm	281mm x 306mm x 141mm	281mm x 306mm x 141mm
Weight (+ SBC + Battery + Sensors)总重量	1kg	1.8kg	1.8kg
Threshold of climbing爬坡高度	10 mm or lower	10 mm or lower	10 mm or lower
Expected operating time续航时间	2h 30m	2h	2h
Expected charging time充电时间	2h 30m	2h 30m	2h 30m
SBC (Single Board Computers)板载计算机	Raspberry Pi 3 Model B	Intel® Joule™ 570x	Raspberry Pi 3 Model B

目录 >

CONTENTS

01

硬件平台

02

实验原理

03

实验内容

04

拓展应用

实验原理

➤ 语音识别算法概述

- 语音识别定义

以语音为研究对象，通过语音信号处理和模式识别让机器自动识别和理解人类口述的语言。

- 语音识别涉及学科领域

信号处理、物理学（声学）、模式匹配、通信及信息理论、语言语音学、生理学、计算机科学（研究软硬件算法以便更有效地实现用于识别系统中的各种方法）等。

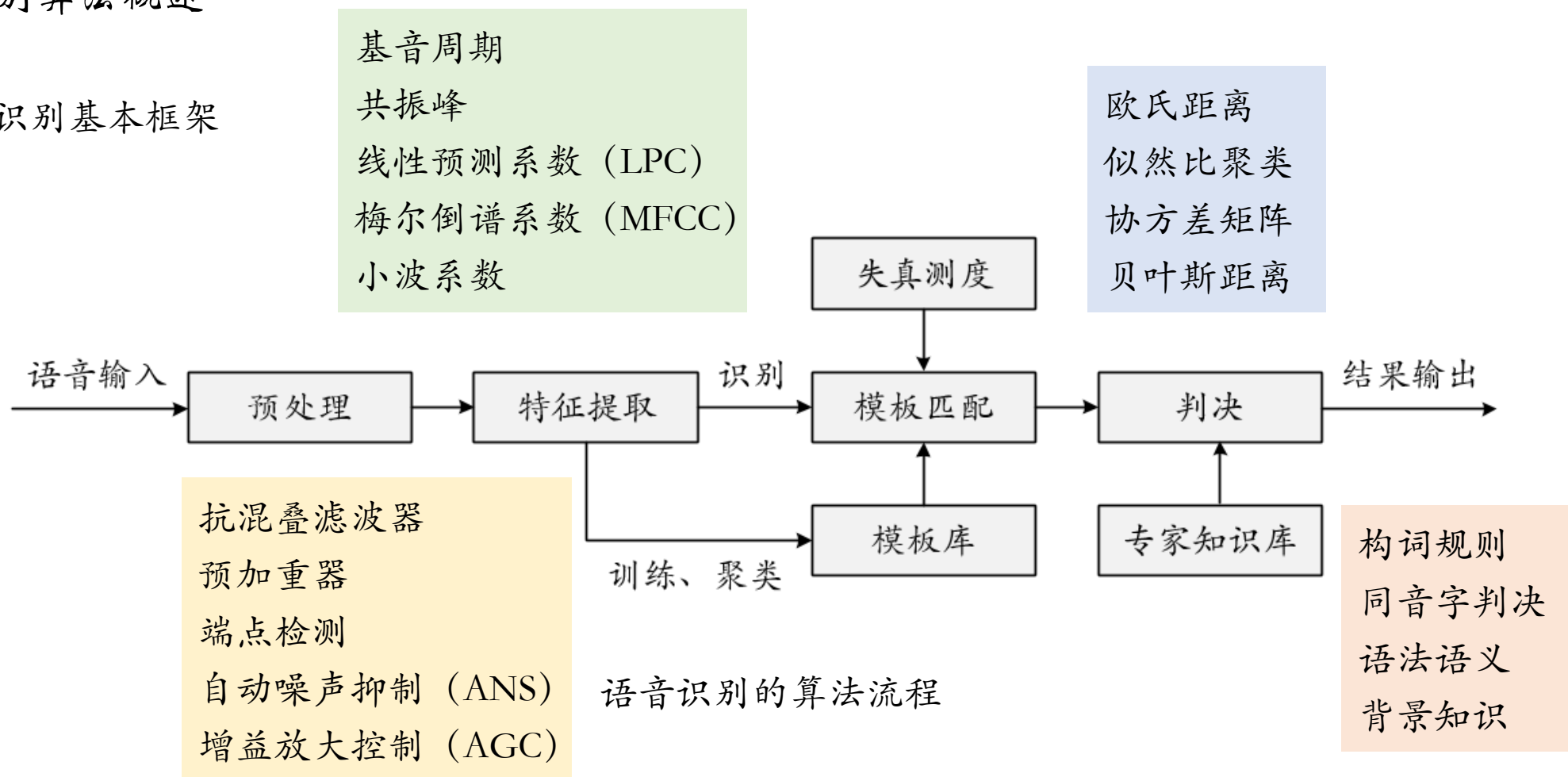
- 身边的语音识别

Siri、Google now、智能家居、车载导航、 . . .

实验原理

➤ 语音识别算法概述

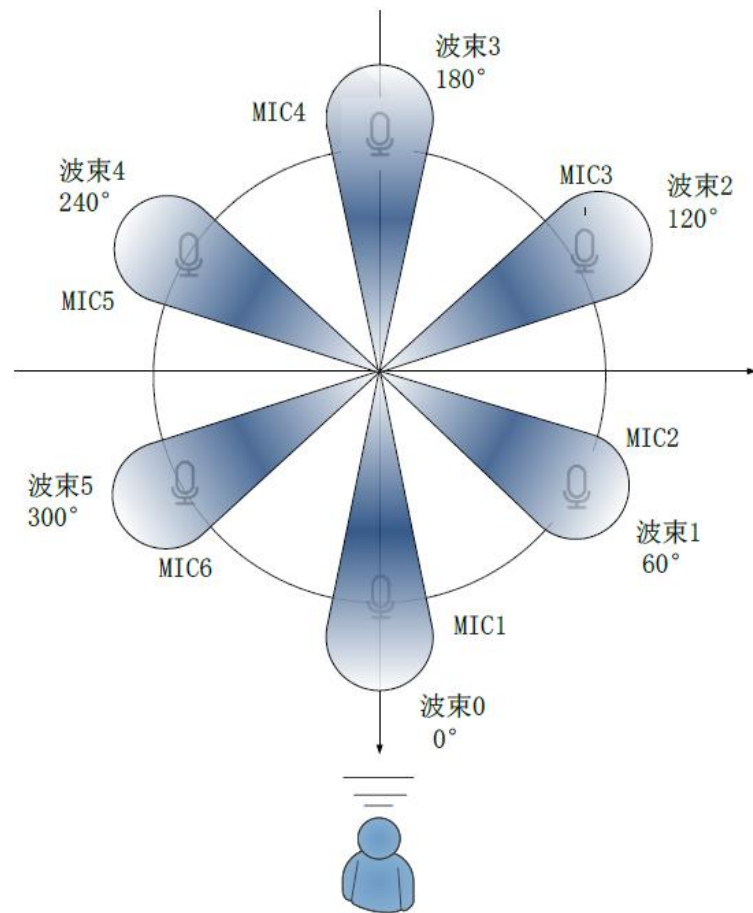
- 语音识别基本框架



实验原理

► 科大讯飞六线麦拾音波束

6麦环形阵列形成6个拾音波束（波束0~5），各自对应 60° 范围。当通过唤醒确定声源角度（算法内部自动确定声源方向的一个拾音波束进行拾音）或指定一个拾音波束进行拾音时，阵列算法会增强波束范围内的声音，削弱波束外的声音，以增强录音信噪比。如指定波束1进行拾音，由于每个波束的范围为 60° ，所以 $30^\circ \sim 90^\circ$ 范围内录音得到增强，波束1范围外的声音会被减弱。



实验原理

➤ 基于Raspberry Pi的控制系统实现

Turtlebot3主控板（OpenCR）的通信接口包括与PC通信的USB和用于其他嵌入式设备的UART，SPI，I2C，CAN。所以在控制系统硬件的选择上具有较大灵活性。主要有以下两种方案：

- 使用单处理器

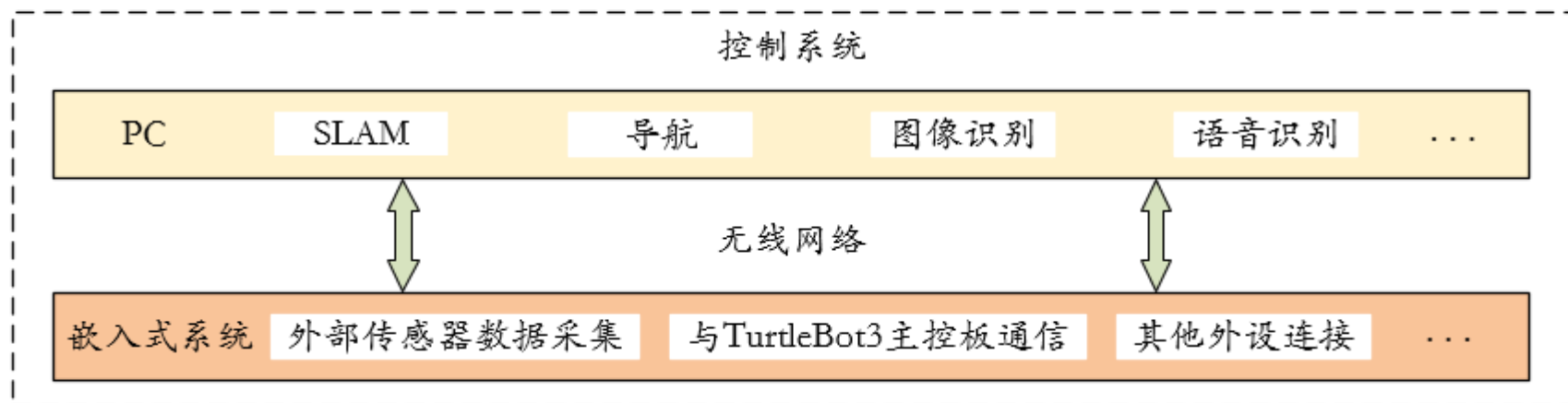
直接使用PC作为控制系统平台，控制系统的功能在PC上使用ROS系统实现，通过USB串口与Turtlebot3主控通信，采集机器人信息并且控制机器人移动。这种方案简单易用，处理器性能强大，可以很快实现ROS中的功能；但是PC体积较大，灵活性欠佳，接口种类、数量比较少，而且无法进行远程监控，不作为推荐方案。

实验原理

➤ 基于Raspberry Pi的控制系统实现

- 使用多处理器

针对第一种方案的缺陷，可以利用ROS的分布式特性，采用“PC+嵌入式系统”方案：嵌入式系统具有灵活性强、接口丰富、功耗低等特点，可以在机器人上搭载嵌入式系统作为本体的控制系统，此外再配合PC，实现远程监控、图形化显示，以及处理复杂功能的运算。

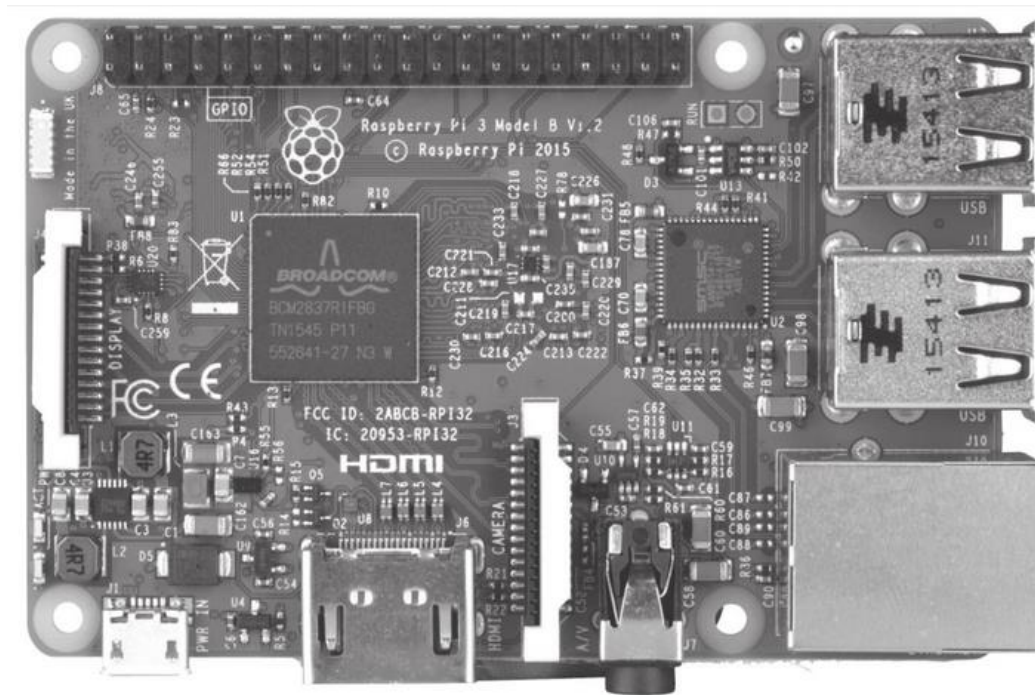


使用多处理器的控制系统结构示意图

实验原理

➤ 基于Raspberry Pi的控制系统实现

- 使用多处理器

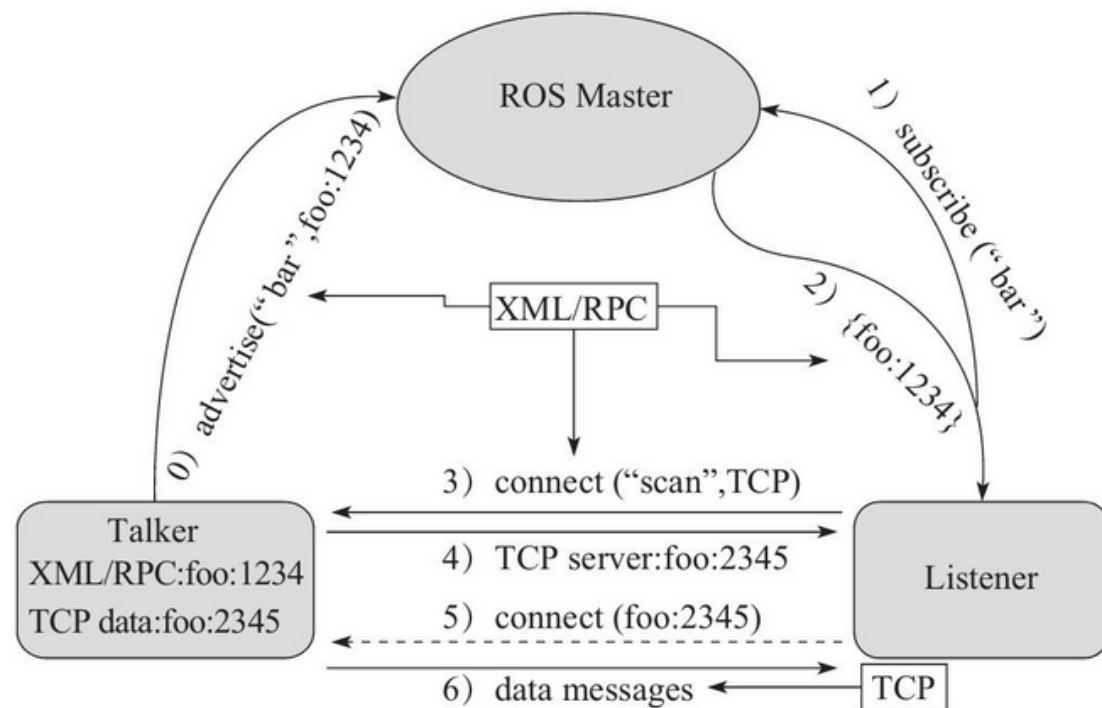


Raspberry Pi嵌入式系统

实验原理

➤ ROS分布式多机通信

ROS是一种分布式框架，为用户提供多节点（进程）之间的通信服务，所有软件功能和工具都建立在这种分布式通信机制上，所以**ROS的通信机制是最底层也是最核心的技术**。节点之间通过松耦合的方式进行组合，在很多应用下，节点可以运行在不同的计算平台上，通过Topic、Service进行通信。ROS中只允许存在一个Master，在多机系统中Master只能运行在一台机器上，其他机器需要通过ssh的方式和Master取得联系。因此，在多机ROS系统中需要进行一些配置。



基于发布/订阅模型的话题通信机制

实验原理

➤ ROS分布式多机通信

0.Talker注册

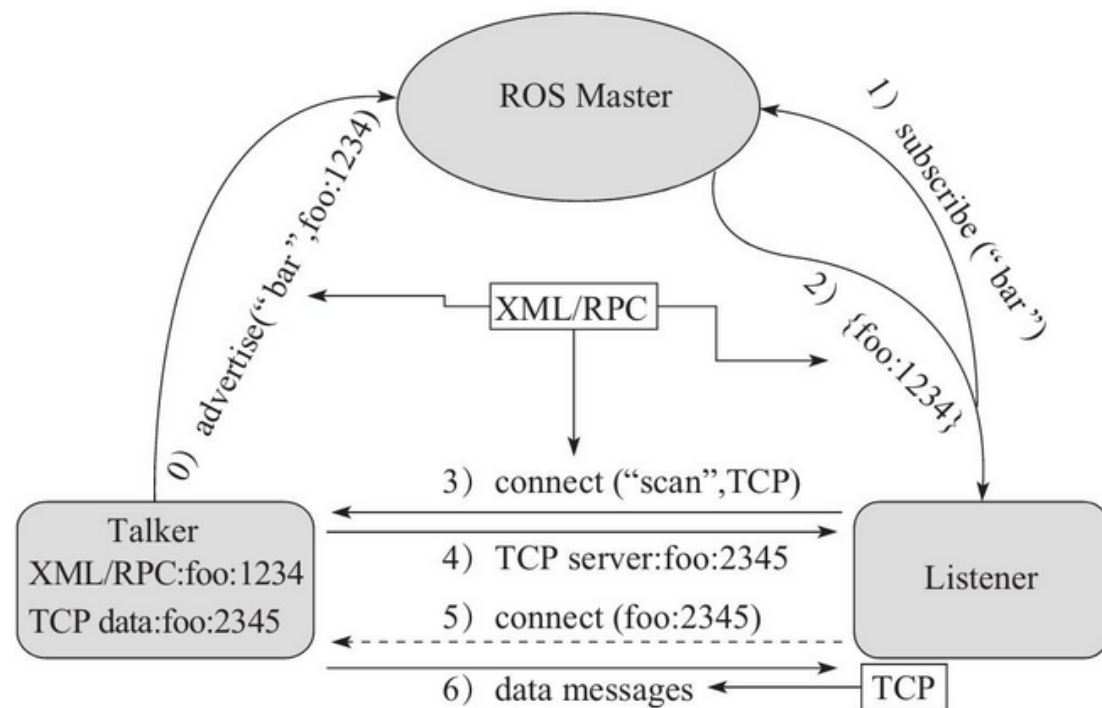
Talker启动，通过1234端口使用RPC向ROS Master注册发布者的信息，包含所发布消息的话题名；ROS Master会将节点的注册信息加入注册列表中。

1.Listener注册

Listener启动，同样通过RPC向ROS Master注册订阅者的信息，包含所需要订阅的话题名。

2.ROS Master进行信息匹配

Master根据Listener的订阅信息从注册列表中进行查找，如果没有找到匹配的发布者，则等待发布者的加入；如果找到匹配的发布者信息，则通过RPC向Listener发送Talker的RPC地址信息。



基于发布/订阅模型的话题通信机制

实验原理

➤ ROS分布式多机通信

3.Listener发送连接请求

Listener接收到Master发回的Talker地址信息，尝试通过RPC向Talker发送连接请求，传输订阅的话题名、消息类型以及通信协议（TCP/UDP）。

4.Talker确认连接请求

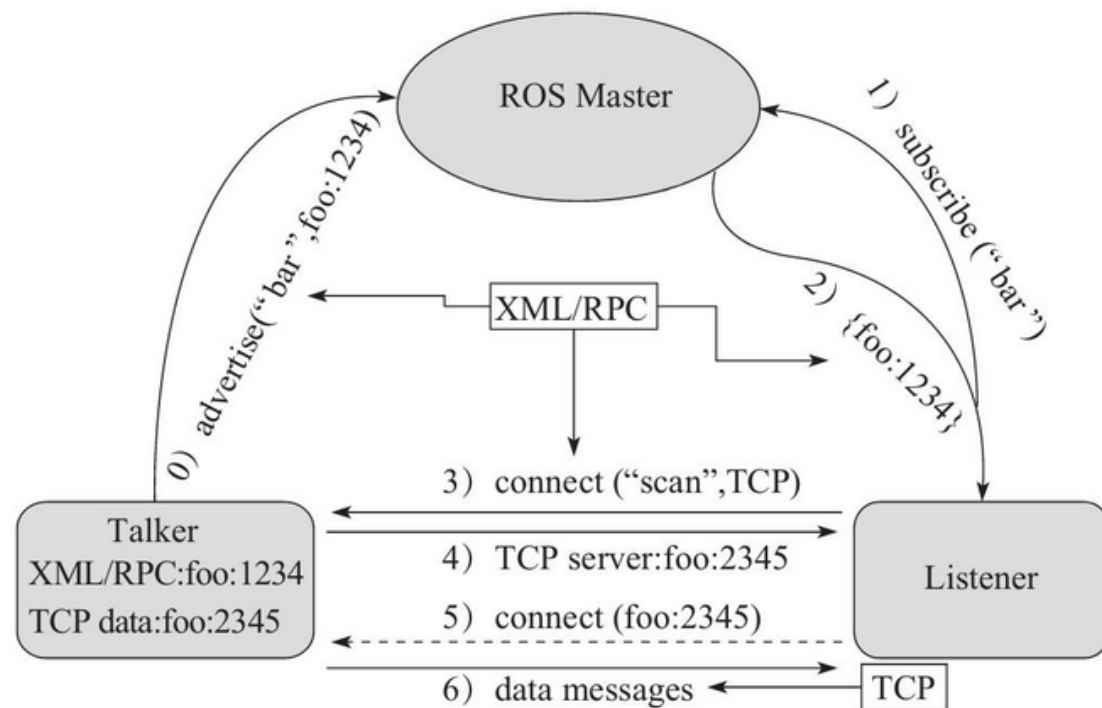
Talker接收到Listener发送的连接请求后，继续通过RPC向Listener确认连接信息，其中包含自身的TCP地址信息。

5.Listener尝试与Talker建立网络连接

Listener接收到确认信息后，使用TCP尝试与Talker建立网络连接。

6.Talker向Listener发布数据

成功建立连接后，Talker开始向Listener发送话题消息数据。



基于发布/订阅模型的话题通信机制

实验原理

➤ ROS时间戳同步

```
Header header
string child_frame_id
geometry_msgs/PoseWithCovariance pose
geometry_msgs/TwistWithCovariance twist
```

[std_msgs/Header Message](#)

File: `std_msgs/Header.msg`

Raw Message Definition

```
# Standard metadata for higher-level stamped data types.
# This is generally used to communicate timestamped data
# in a particular coordinate frame.
#
# sequence ID: consecutively increasing ID
uint32 seq
#Two-integer timestamp that is expressed as:
# * stamp.sec: seconds (stamp_secs) since epoch (in Python the variable is called 'secs')
# * stamp.nsec: nanoseconds since stamp_secs (in Python the variable is called 'nsecs')
# time-handling sugar is provided by the client library
time stamp
#Frame this data is associated with
string frame_id
```

ROS中有一种特殊的类型：Header，包含ROS中常用的时间戳和坐标系信息，会经常看到在一个msg文件中的第一行有Header header。

很多ROS消息定义中会包含一个标准格式的头信息std_msgs/Header：seq是消息的顺序标识，不需要手动设置，Publisher在发布信息时会自动累加；stamp是消息中与数据相关联的时间戳，可以用于时间同步；frame_id是消息中与数据相关联的参考坐标系id。

对于ROS应用来说，在多机操作中时间同步，在tf或者其他话题时间戳同步上十分重要。在双轮差速平台中，由于树莓派和用来进行建图导航的主机时间不一致，导致了tf之间的时间不一致，在rviz中会报tf warned错误，无法完成正常的建图等工作。

目录 >

CONTENTS

01

硬件平台

02

实验原理

03

实验内容

04

拓展应用

实验内容

➤ 科大讯飞语音听写SDK

➤ 基于语音听写SDK添加ROS接口

➤ 语音控制仿真环境下的机器人移动

➤ 语音控制实际机器人turtlebot3移动

实验内容

➤ 科大讯飞语音听写SDK

1.科大讯飞开放平台<https://www.xfyun.cn/>注册账号，并下载SDK



实验内容

➤ 科大讯飞语音听写SDK

2.科大讯飞开放平台<https://www.xfyun.cn/>创建应用

讯飞开放平台

产品 ▾ 解决方案 ▾ 资料库 ▾ iFLYOS ▾ 服务市场 ▾ AI大学 HOT

控制台

应用管理

我的应用

创建应用

AIUI智能硬件

SDK下载

财务中心

账户

充值

优惠券

购物车

订单管理

用户中心

创建应用

* 应用名称

ros_voice123

* 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

* 应用平台

Linux ▾

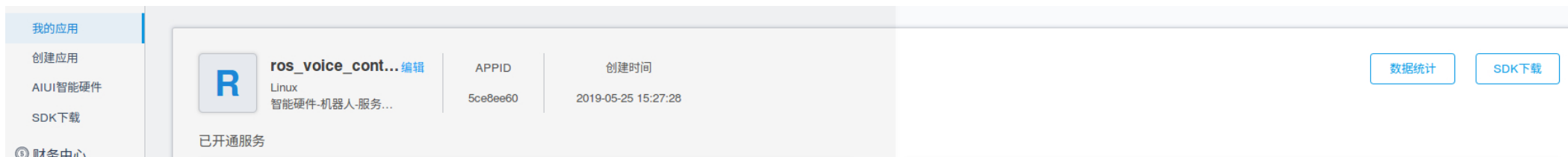
☒ 我已阅读并接受 [《讯飞用户服务协议》](#)

提交

实验内容

➤ 科大讯飞语音听写SDK

3. 下载SDK



4. 选择需要的AI能力

选择语音听写，下载压缩包，解压

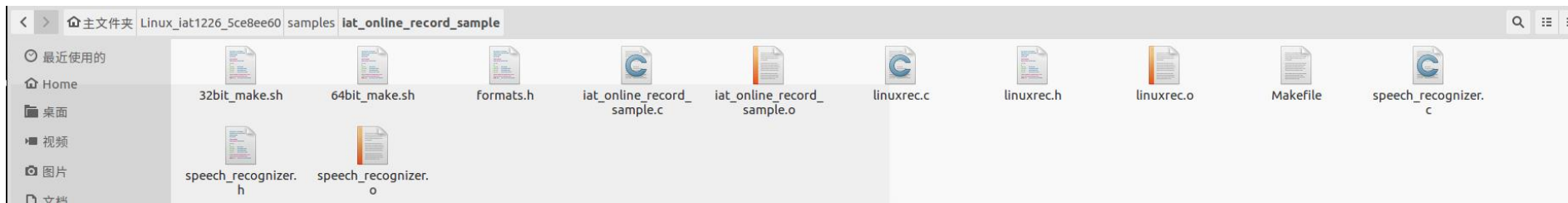
5. 环境配置

- (1) `sudo apt-get install gcc-multilib g++-multilib module-assistant`
- (2) `cd /home/zx/Linux_iat1226_5ce8ee60`(换成自己的路径)
- (3) `sudo cp libs/x64/libmsc.so /usr/local/lib/`
- (4) `sudo ldconfig`

实验内容

► 科大讯飞语音听写SDK

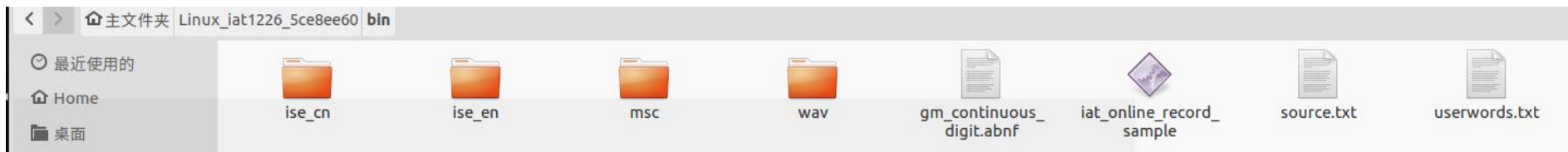
6.进入下图所示的文件目录



右键，在终端打开执行：(1)source 64bit_make.sh(2)make

7.测试SDK

编译成功后，会在bin文件夹下找到可执行文件。右键选择在终端打开，执行：
./iat_online_record_sample 弹出提示，分别选择0和1。



实验内容

➤ 基于语音听写SDK添加ROS接口

本节基于科大讯飞提供的免费语音识别SDK，使用ROS系统构建一个中文语音交互的机器人应用。

1.新建ros工作空间voice_ws及功能包

```
(1)mkdir -p ~/voicetb3_ws/src
(2)cd ~/voicetb3_ws/
(3)catkin_make
(4)cd
(5)echo "source ~/voicetb3_ws/devel/setup.bash" >> ~/.bashrc
(6)source .bashrc
(7)cd voicetb3_ws/src
(8)创建功能包robot_voice:
catkin_create_pkg robot_voice std_msgs rospy roscpp
```

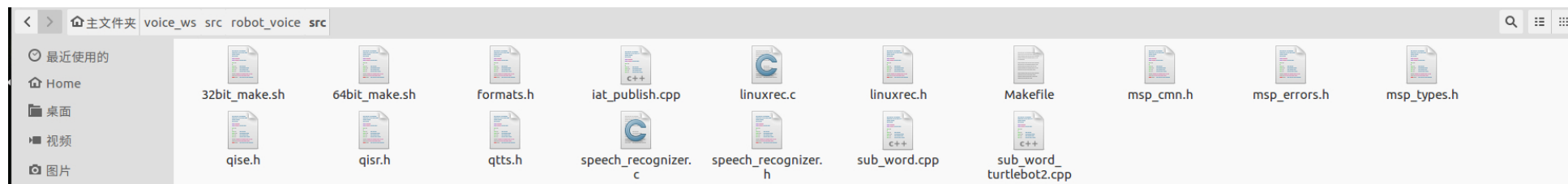
创建完成后，代码空间src会生成一个robot_voice功能包，其中已经包含package.xml和CMakeLists.txt文件。

实验内容

➤ 基于语音听写SDK添加ROS接口

本节基于科大讯飞提供的免费语音识别SDK，使用ROS系统构建一个中文语音交互的机器人应用。

2. 本部分实验基于科大讯飞SDK中的iat_online_record_sample例程（samples文件夹下），将该例程中的代码拷贝到功能包robot_voice的src文件目录下，如下图所示。将下载的SDK文件的include文件夹中的头文件分别copy到，robot_voice/include/robot_voice文件目录下以及robot_voice/src文件目录下。



针对主要代码文件iat_online_record_sample.c进行修改，添加需要的ROS接口，修改完成后重命名文件为robot_voice/src/iat_publish.cpp（注意文件名后缀为cpp），该cpp文件已给，需要将appid更换为自己的appid（讯飞开放平台查<https://www.xfyun.cn/>我的应用中查看）。

```
const char* login_params = "appid = 5ce8ee60, work_dir = .";
```

实验内容

➤ 基于语音听写SDK添加ROS接口

本节基于科大讯飞提供的免费语音识别SDK，使用ROS系统构建一个中文语音交互的机器人应用。

3.编写CmakeLists.txt

用所给的CMakeLists.txt替换掉robot_voice目录下的CMakeLists.txt。

4.编译工作空间

(1)cd ~/voice_ws/

(2)catkin_make

5.编译完成后使用如下命令进行测试

(1)roscore

(2)roslaunch robot_voice iat_publish

实验内容

➤ 基于语音听写SDK添加ROS接口

本节基于科大讯飞提供的免费语音识别SDK，使用ROS系统构建一个中文语音交互的机器人应用。

出现如下界面，即成功：

```
roscore http://zx-FX80GE:11311/
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://zx-FX80GE:41401/
ros_comm version 1.12.14

SUMMARY
=====

PARAMETERS
* /rostdistro: kinetic
* /rosversion: 1.12.14

NODES

auto-starting new master
process[master]: started with pid [23963]
ROS_MASTER_URI=http://zx-FX80GE:11311/

setting /run_id to 05c65ef4-8084-11e9-be7e-4cedfb2bca62
process[rosout-1]: started with pid [23976]
started core service [/rosout]
```

```
zx@zx-FX80GE: ~
zx@zx-FX80GE:~$ rosrn robot_voice iat_publish
[ INFO] [1558964174.659022305]: Sleeping...
[ INFO] [1558964174.659056354]: Wakeup...
Demo recognizing the speech from microphone
Speak in 10 seconds
Start Listening...
Result: [ 你好。 ]
10 sec passed
[ INFO] [1558964177.997442775]: Wakeup...
Demo recognizing the speech from microphone
Speak in 10 seconds
Start Listening...
Result: [ 机器人。 ]
10 sec passed
[ INFO] [1558964181.218724202]: Wakeup...
Demo recognizing the speech from microphone
Speak in 10 seconds
Start Listening...
10 sec passed
[ INFO] [1558964184.441941151]: Wakeup...
Demo recognizing the speech from microphone
Speak in 10 seconds
Start Listening...
10 sec passed
```

实验内容

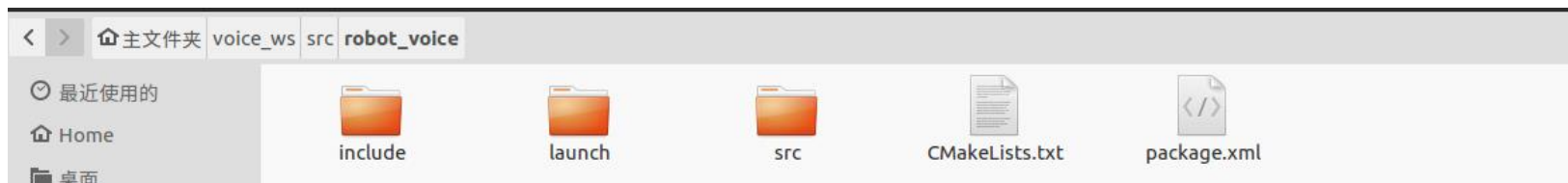
➤ 语音控制仿真环境下的机器人移动

本节利用ROS仿真环境下的“小乌龟”机器人，进行语音导航算法的调试与验证。

1.将sub_word.cpp（已给）拷贝到robot_voice/src目录下，在robot_voice目录下的CMakeLists.txt中添加如下内容：

```
add_executable(sub_word
src/sub_word.cpp)
target_link_libraries(
sub_word
${catkin_LIBRARIES})
```

2.在robot_voice目录下新建launch文件夹，如图所示。将voice_control.launch(已给)拷贝到launch文件夹下。



实验内容

➤ 语音控制仿真环境下的机器人移动

本节利用ROS仿真环境下的“小乌龟”机器人，进行语音导航算法的调试与验证。

3.编译工作空间

(1)`cd ~/voicetb3_ws/`

(2)`catkin_make`

4.终端命令行，开始测试：

```
roslaunch robot_voice voice_control.launch
```

实验内容

➤ 语音控制实际机器人turtlebot3移动

一、PC端安装turtlebot3源码及依赖包

1.安装turtlebot3依赖包

```
sudo apt-get install ros-kinetic-joy ros-kinetic-teleop-twist-joy ros-kinetic-teleop-twist-keyboard ros-kinetic-laser-proc ros-kinetic-rgbd-launch ros-kinetic-depthimage-to-laserscan ros-kinetic-rosserial-arduino ros-kinetic-rosserial-python ros-kinetic-rosserial-server ros-kinetic-rosserial-client ros-kinetic-rosserial-msgs ros-kinetic-amcl ros-kinetic-map-server ros-kinetic-move-base ros-kinetic-urdf ros-kinetic-xacro ros-kinetic-compressed-image-transport ros-kinetic-rqt-image-view ros-kinetic-gmapping ros-kinetic-navigation ros-kinetic-interactive-markers
```

实验内容

➤ 语音控制实际机器人turtlebot3移动

一、PC端安装turtlebot3源码及依赖包

2.源码安装turtlebot3功能包

```
$ cd ~/voicetb3_ws/src/  
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git  
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3.git  
$ cd ~/voicetb3_ws && catkin_make
```

如果catkin_make完成没有任何错误，使用TurtleBot3的准备将完成。

实验内容

➤ 语音控制实际机器人turtlebot3移动

二、网络配置

1. 时间同步

在turtlebot3终端输入指令：date

检查时间是否同步，若不同步需进一步配置：

(1) 主从分别安装chrony

```
sudo apt-get install chrony
```

(2) 主从分别安装 ntpdate

```
sudo apt-get install ntpdate
```

(3) 主从分别同步时间

```
sudo ntpdate ntp.ubuntu.com
```

(4) 检测时间是否同步

```
date
```

2. ROS需要IP地址在turtlebot和远程PC之间进行通信

分别在turtlebot3和PC，执行如下命令获得对应的IP地址：

```
$ ifconfig
```

修改.bashrc

```
$ gedit ~/.bashrc
```

TURTLEBOT3配置如下：

```
$ ROS_MASTER_URI = http://IP_OF_PC:11311
```

```
$ ROS_HOSTNAME = IP_OF_TURTLEBOT
```

保存，让环境生效：

```
source ~/.bashrc
```

实验内容

➤ 语音控制实际机器人turtlebot3移动

二、网络配置

2. ROS需要IP地址在turtlebot和远程PC之间进行通信

PC配置如下:

```
$ ROS_MASTER_URI = http://IP\_OF\_PC:11311
```

```
$ ROS_HOSTNAME = IP_OF_PC
```

```
$ export TURTLEBOT3_MODEL=waffle_pi
```

ROS_MASTER 运行在远程PC上。

保存, 让环境生效:

```
$ source ~/.bashrc
```

3.启动测试

[Remote PC] 启动 roscore

```
roscore
```

[TurtleBot] 启动 waffle_pi

```
roslaunch turtlebot3_bringup turtlebot3_robot.launch
```

[Remote PC] 启动 rviz

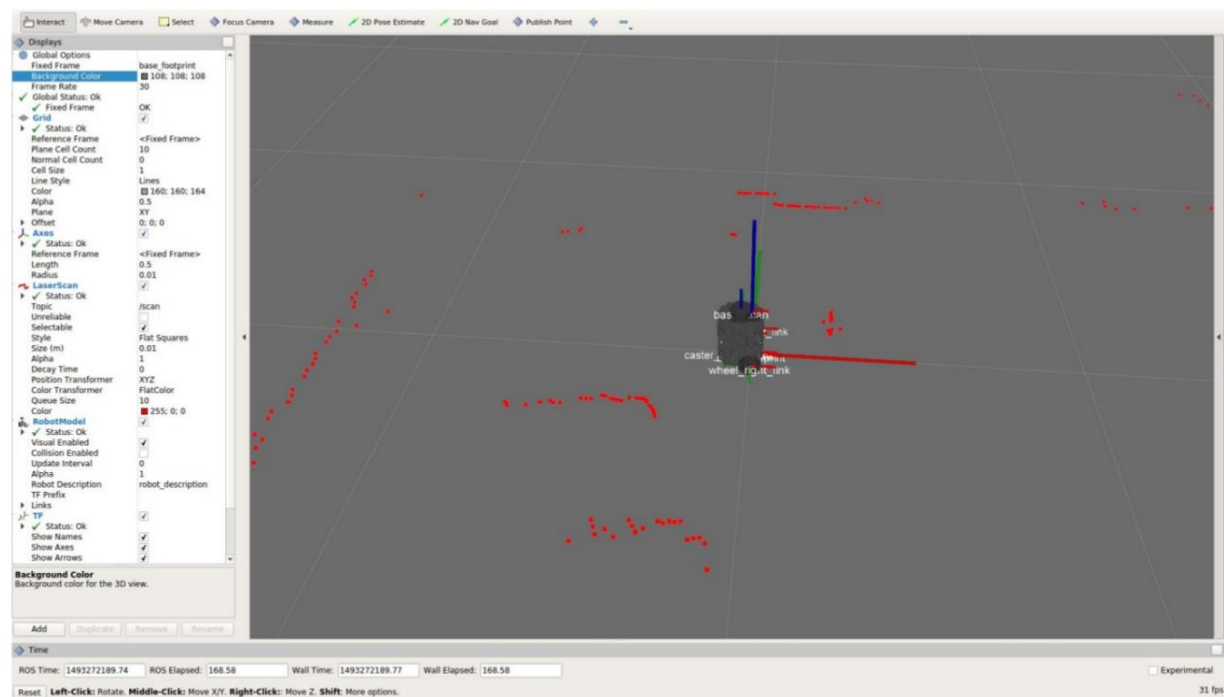
```
roslaunch turtlebot3_bringup turtlebot3_model.launch
```

实验内容

➤ 语音控制实际机器人turtlebot3移动

二、网络配置

出现如下示意图，即成功。



实验内容

➤ 语音控制实际机器人turtlebot3移动

三、PC远程键盘控制turtlebot3测试

[Remote PC] 启动 roscore

```
roscore
```

[TurtleBot] 启动 waffle_pi

```
roslaunch turtlebot3_bringup turtlebot3_robot.launch
```

[Remote PC] 启动进行简单远程操作测试

```
roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

[Remote PC] 如果文件成功启动,终端将显示以下内容。

```
/home/zx/voicetb3_ws/src/turtlebot3/turtlebot3_teleop/launch/turtlebot3_teleop_key.launch
ROS_MASTER_URI=http://192.168.1.109:11311

setting /run_id to 95bbc468-852a-11e9-b800-80c5f2b94e7b
process[rosout-1]: started with pid [3860]
started core service [/rosout]
process[turtlebot3_teleop_keyboard-2]: started with pid [3863]

Control Your TurtleBot3!
-----
Moving around:
   w   a   s   d   x
   |   |   |   |
w/x : increase/decrease linear velocity (Burger : ~ 0.22, Waffle and Waffle Pi : ~ 0.26)
a/d : increase/decrease angular velocity (Burger : ~ 2.84, Waffle and Waffle Pi : ~ 1.82)

space key, s : force stop
CTRL-C to quit
```

将鼠标光标定位在该终端下, 注意w,a,x,d为方向键, w按一下表示以0.01m/s的速度前进,一直按表示加速运动, 一定要注意控制好速度, a,x,d同理, 空格键和s表示急停。

实验内容

➤ 语音控制实际机器人turtlebot3移动

四、远程PC+六线麦控制turtlebot3

1.将sub_word_tb3.cpp拷贝到robot_voice/src文件目录下;

2.编写CmakeLists.txt

在“语音控制仿真环境下的机器人移动”实验的CmakeLists.txt文件中添加如下内容:

```
add_executable(sub_word_tb3
```

```
src/sub_word_tb3.cpp)
```

```
target_link_libraries(
```

```
sub_word_tb3
```

```
${catkin_LIBRARIES})
```

实验内容

➤ 语音控制实际机器人turtlebot3移动

四、远程PC+六线麦控制turtlebot3

3.编译工作空间

```
cd ~/voice_ws/
```

```
catkin_make
```

4.终端命令:

```
[Remote PC] 启动 roscore
```

```
roscore
```

```
[TurtleBot] 启动 waffle_pi
```

```
roslaunch turtlebot3_bringup
```

```
turtlebot3_robot.launch
```

```
[Remote PC]
```

```
roslaunch robot_voice voice_control_tb3.launch
```

voice_control_tb3.launch已给。

这里需要注意的是，Waffle Pi的速度指标：
Maximum translational velocity最大直行速度0.26m/s

Maximum rotational velocity最大旋转速度1.82rad/s(104.27deg/s)

查看话题的消息数据：

```
rostopic echo /voiceWords
```

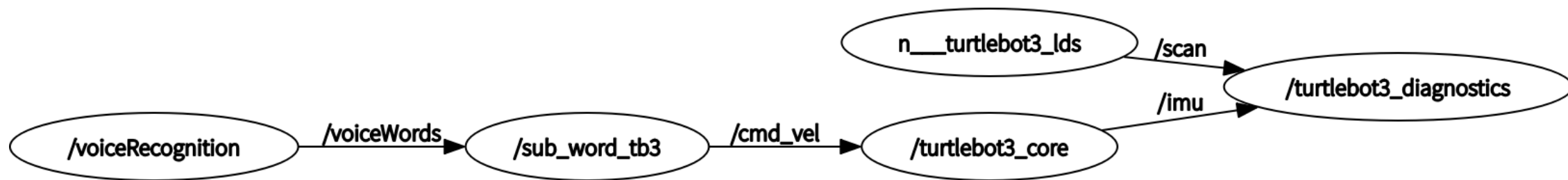
```
rostopic echo /cmd_vel
```

```
rosmmsg show geometry_msgs/Twist
```

实验内容

➤ 语音控制实际机器人turtlebot3移动

四、远程PC+六线麦控制turtlebot3



实验内容

语音识别常见报错代码：讯飞开放平台错误码查询<https://www.xfyun.cn/document/error-code>

10114	会话超时	1. 网络问题，导致请求超时； 2. webapi接口的调用方时间和标准时间有误差，导致引擎认定超时。	[SDK接口] 可能与当前网络质量有关，可以换个网络再测试一下，查看网络环境是否正常，查看是否使用的网络对请求域名或者端口有限制。 [webapi接口] 检查 curTime 是否符合要求，单位是否为秒（10位数字）。另外，checksum 有效期为 5 分钟(用 curTime 计算)，同时 curTime 要与标准时间同步，若时间相差太大，请调整为标准时间。
10407	appid校验失败	appid和MSC库不匹配	确保appid和MSC库相匹配。 检查代码中所有填入appid的地方，并检查其他目录libs下so文件是否有误引用。如果DEMO能用的话，请确认您应用中的 APPID与DEMO中的是否一致；libmsc.so 文件，是否与DEMO中的一致（可通过 Beyond Compare等工具对比 so 文件）。
11201	授权不足	在线服务接口报错： 该APPID的每日交互次数超过限制（未经审核的应用每天允许调用在线语音服务的次数为500次，部分服务支持提交应用审核后获得每天2万次调用权限） 离线服务接口报错： 购买或供体验的装机量已用完	在线服务接口报错： 根据自身情况提交应用审核进行服务量提额，或者联系商务购买企业级正式接口，获得海量服务量权限以便商用。 离线服务接口报错： 体验装机量用完了，可以在平台官网购买正式版，获得更多装机量。

目录 >

CONTENTS

01

硬件平台

02

实验原理

03

实验内容

04

拓展应用

拓展应用

- 参照实验教程，在ROS仿真环境下，完成基于科大讯飞SDK的语音导航系统设计，通过语音对仿真小乌龟进行运动控制。

iflytek_annular6mic_manual	2020/5/12 23:17	文件夹
robot_voice	2020/5/12 23:18	文件夹
Source	2020/5/12 23:18	文件夹
turtlebot3	2020/5/12 23:18	文件夹
turtlebot3_msgs	2020/5/12 23:18	文件夹

直接可用的ROS功能包



- 通过计算图可视化工具（rqt_graph）图形化显示当前ROS系统中的计算图，进一步理解ROS节点之间的基于发布/订阅模型的通信机制。

Thanks for listening!

