

一、CastleX功能包更新教程

1.上下位通信功能包

(1)下载[ros-melodic-castlex-bringup_amd64.deb](#)包

(2)安装功能包

```
sudo dpkg -i ros-melodic-castlex-bringup_amd64.deb
roscd castlex_bringup/scripts/
sudo chmod +x *.py
```

(3)启动指令

```
roslaunch castlex_bringup castlex_stm32_bringup.launch
```

(4)话题说明

2.激光雷达功能包

(1)下载[ros-melodic-arvlidar-ros_amd64.deb](#)包

(2)安装功能包

```
sudo dpkg -i ros-melodic-arvlidar-ros_amd64.deb
```

(3)启动指令

```
roslaunch arvlidar_ros castlexlidar_A2.launch
```

(4)话题说明

3.语音系统功能包

(1)下载[ros-melodic-castlex-voice-system_amd64.deb](#)包

(2)安装功能包

```
sudo dpkg -i ros-melodic-castlex-voice-system_amd64.deb
roscd castlex_voice_system/scripts/
sudo chmod +x *.py
```

(3)启动指令

```
# 启动语音交互功能（需要联网）
roslaunch castlex_voice_system castlex_voice_system.launch

# 启动语音导航功能
roslaunch castlex_voice_system castlex_voice_nav.launch

# 物联网相应功能
roslaunch castlex_voice_system castlex_iot.launch

# 喷雾消杀功能
roslaunch castlex_voice_system castlex_spray_kill.launch

# 紫外线消杀功能
roslaunch castlex_voice_system castlex_ultraviolet_disinfection.launch

# 讲解功能
roslaunch castlex_voice_system castlex_com.launch

# 语音抓取物体功能(机械臂)
roslaunch castlex_voice_system castlex_voice_object.launch
```

其他命令词自行修改launch文件中的bnf文件

4.物联网功能包

(1)下载ros-melodic-iot-wifi-system_amd64.deb包

(2)安装功能包

```
sudo dpkg -i ros-melodic-iot-wifi-system_amd64.deb
```

(3)启动指令

```
roslaunch iot_wifi_system castlex_iot_device.launch
```

二、功能包参数说明

1. castlex_voice_system功能包

节点名称(Name)	描述	备注
castlex_awake_node	语音唤醒	唤醒词是小谷小谷
castlex_tts_node	语音合成	提供发布/voiceWords话题，类型为std_msgs::String，合成音频文件保留在主目录下，需要联网
castlex_nlu_node	语义理解	调用图灵机器人问答库，需要联网
castlex_asr_node	语音识别	语音转文字
offline_command_word	离线命令词识别	FILE_path为命令词bnf文件路径，需要结合命令词解析程序使用

2.iot_wifi_system功能包

物联网灯：

设备名称(Name)	话题(Topic)	话题类型(Type)	数据(data)	备注
灯1（开）	/Lighting_CMD_Topic	std_msgs::Int32/std_msgs.msg.Int32	1	001
灯1（关）	/Lighting_CMD_Topic	std_msgs::Int32/std_msgs.msg.Int32	6	110
灯2（开）	/Lighting_CMD_Topic	std_msgs::Int32/std_msgs.msg.Int32	2	010
灯2（关）	/Lighting_CMD_Topic	std_msgs::Int32/std_msgs.msg.Int32	5	101
灯3（开）	/Lighting_CMD_Topic	std_msgs::Int32/std_msgs.msg.Int32	4	100
灯3（关）	/Lighting_CMD_Topic	std_msgs::Int32/std_msgs.msg.Int32	3	011
所有灯（关）	/Lighting_CMD_Topic	std_msgs::Int32/std_msgs.msg.Int32	0	000
所有灯（开）	/Lighting_CMD_Topic	std_msgs::Int32/std_msgs.msg.Int32	7	111

物联网门铃：

设备名称(Name)	话题(Topic)	话题类型(Type)	数据(data)	备注
门铃1（开）	/Door_CMD_Topic	std_msgs::Int32/std_msgs.msg.Int32	1	01
门铃2（开）	/Door_CMD_Topic	std_msgs::Int32/std_msgs.msg.Int32	2	10
所有门铃（开）	/Door_CMD_Topic	std_msgs::Int32/std_msgs.msg.Int32	3	11

物联网闸机：

设备名称(Name)	话题(Topic)	话题类型(Type)	数据(data)	备注
闸机（关）	/Gateway_CMD_Topic	std_msgs::Int32/std_msgs.msg.Int32	0	0
闸机（开）	/Gateway_CMD_Topic	std_msgs::Int32/std_msgs.msg.Int32	1	1

物联网窗帘：

设备名称(Name)	话题(Topic)	话题类型(Type)	数据(data)	备注
窗帘（关）	/Trashcan_CMD_Topic	std_msgs::Int32/std_msgs.msg.Int32	0	0
窗帘（开）	/Trashcan_CMD_Topic	std_msgs::Int32/std_msgs.msg.Int32	1	1

3.arvlidar_ros功能包

castlexlidar_A2.launch：

名称(Name)	描述	备注
----------	----	----

名称(Name)	描述	备注
arvlidarNode	节点，用于启动激光雷达	位于arvlidar_ros功能包下
serial_port	USB串口	类型为string
serial_baudrate	波特率	无
frame_id	坐标系名称	
inverted	激光雷达是否倒装	默认为False
angle_compensate	是否对激光雷达数据进行裁剪	True
min_angle	裁剪最小角度	左边：-180~0；右边：0~180
max_angle	裁剪最大角度	左边：-180~0；右边：0~180

4.castlex_bringup功能包

设备名称 (Name)	话题(Topic)	话题类型(Type)	数据 (data)	备注
货仓（关）	/Warehouse_control	std_msgs::Int32/std_msgs.msg.Int32	0	's(0)'
货仓（开）	/Warehouse_control	std_msgs::Int32/std_msgs.msg.Int32	1	's(1)'
货仓灯光控制	/Warehouse_light_control	std_msgs::Int32/std_msgs.msg.Int32	0	0-8399
紫外线消杀 （关）	/ultraviolet_disinfection	std_msgs::Int32/std_msgs.msg.Int32	0	"ZOFF"
紫外线消杀 （开）	/ultraviolet_disinfection	std_msgs::Int32/std_msgs.msg.Int32	1	"ZON"
喷雾消杀 （关）	/spray_kill	std_msgs::Int32/std_msgs.msg.Int32	0	'x(0)'
喷雾消杀 （打开一级）	/spray_kill	std_msgs::Int32/std_msgs.msg.Int32	1	'x(8399)'
喷雾消杀 （打开二级）	/spray_kill	std_msgs::Int32/std_msgs.msg.Int32	2	'x(8000)'
喷雾消杀 （打开三级）	/spray_kill	std_msgs::Int32/std_msgs.msg.Int32	3	'x(7400)'

防跌落、防碰撞、超声波传感器控制：

传感器名称(Name)	话题(Topic)	话题类型(Type)	数据 (data)	备注
-------------	-----------	------------	--------------	----

传感器名称(Name)	话题(Topic)	话题类型(Type)	数据(data)	备注
防跌落、防碰撞、 超声波传感器（全开启）	/sensor_switch	std_msgs::Int32/std_msgs.msg.Int32	7	'G(1) (1) (1) (1)'
防跌落、防碰撞、 超声波传感器（全关闭）	/sensor_switch	std_msgs::Int32/std_msgs.msg.Int32	0	'G(0) (0) (0) (0)'
只开启超声波传感器	/sensor_switch	std_msgs::Int32/std_msgs.msg.Int32	1	'G(0) (0) (1) (1)'
只开启防跌落传感器	/sensor_switch	std_msgs::Int32/std_msgs.msg.Int32	2	'G(0) (1) (0) (1)'
开启防跌落和超声波传感器	/sensor_switch	std_msgs::Int32/std_msgs.msg.Int32	3	'G(0) (1) (1) (1)'
只开启防碰撞传感器	/sensor_switch	std_msgs::Int32/std_msgs.msg.Int32	4	'G(1) (0) (0) (1)'
开启超声波和防碰撞传感器	/sensor_switch	std_msgs::Int32/std_msgs.msg.Int32	5	'G(1) (0) (1) (1)'
开启防碰撞与防跌落	/sensor_switch	std_msgs::Int32/std_msgs.msg.Int32	6	'G(1) (1) (0) (1)'

传感器数据读取：

传感器名称(Name)	话题(Topic)	话题类型(Type)	数据(data)	备注
防碰撞传感器	/AntiCollision_data	Int32		0/1
超声波传感器	/ultrasonic_data	Float32MultiArray		左、右、预留

传感器名称(Name)	话题(Topic)	话题类型(Type)	数据(data)	备注
防跌落传感器	/FallPrevention_data	Float32MultiArray		左、右、后
陀螺仪	/gyro_data	Float32		单位：度
电量	/battery_capacity	Float32		