

# ***Rockchip***

## ***ROS使用指南***

**发布版本:1.01**

**日期:2019.12**

## 免责声明

本文档按“现状”提供，福州瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

## 商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

## 版权所有 © 2018 福州瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

福州瑞芯微电子股份有限公司

Fuzhou Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园 A 区

18 号网址：[www.rock-chips.com](http://www.rock-chips.com)

客户服务电话：+86-591-83991906

客户服务传真：+86-591-83951833

客户服务邮箱：[www.rock-chips.com](mailto:www.rock-chips.com)

# 前言

## 概述

本文档主要介绍 Rockchip Linux SDK 上使用 ROS 的方法。

## 产品版本

芯片名称	内核版本
RK3XXX	4.40

## 读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

## 修订记录

日期	版本	作者	修改说明
2018-12	V1.00	王征增	Ros已经移入buildroot中，移除原先的编译方式，改用新的编译
2019-12	V1.01	王征增	修复beginner_tutorials编译错误

# 1 简述

Rockchip Linux SDK 集成了 ROS。ROS 提供一系列程序库和工具以帮助软件开发者创建机器人应用软件。

Rockchip 所集成的 ROS 版本为 Indigo 和 kinetic 两个版本。

# 2 编译

在 buildroot/configs/rockchip 下面有 ros\_indigo.config 和 ros\_kinetic.config 两个默认配置。在编译 rootfs 前，先将 ros\_xxx.config 加入到 rootfs 对应的 config 中。

以 RK3308 Linux SDK 为例。其他方法类似。修改  
buildroot/configs/rockchip\_rk3308\_release\_defconfig

```
diff --git a/configs/rockchip_rk3308_release_defconfig
b/configs/rockchip_rk3308_release_defconfig
index f905f16..a2afac1 100644
--- a/configs/rockchip_rk3308_release_defconfig
+++ b/configs/rockchip_rk3308_release_defconfig
@@ -135,3 +135,4 @@ BR2_TARGET_ROOTFS_SQUASHFS=y
 # BR2_TARGET_ROOTFS_TAR is not set
 BR2_PACKAGE_HOST_MKE2IMG=y
 BR2_PACKAGE_HOST_VBOOT_UTILS=y
+#include "ros_indigo.config"
```

然后运行 ./build.sh,

或者先 source envsetup.sh 选择对应 rockchip\_rk3308\_release, 然后再运行 make

第一次编译需要几个小时的时间。编译完成后会生成

buildroot/output/rockchip\_rk3308\_release/images/rootfs.squashfs。这就是 rootfs 的固件

# 3 添加新的ros代码

提供一个 ros\_sample: [https://github.com/DZain/ROS\\_Sample.git](https://github.com/DZain/ROS_Sample.git)

将该工程同步至 external 下，并改名为 beginner\_tutorials。

然后在 buildroot 中添加以下文件和修改：

1) vi buildroot/package/rockchip/ros/beginner\_tutorials/Config.in  
添加以下内容：

```
1 config BR2_PACKAGE_BEGINNER_TUTORIALS
2     bool "beginner tutorials"
3     select BR2_PACKAGE_ROSCPP
4     select BR2_PACKAGE_ROSPY
5     select BR2_PACKAGE_STD_MSGS
6     select BR2_PACKAGE_GENMSG
7     help
8         beginner tutorials
```

Config 中的select项，取决于工程中的依赖项。

2) vi buildroot/package/rockchip/ros/beginner\_tutorials/beginner\_tutorials.mk

```
1 BEGINNER_TUTORIALS_VERSION = 1.0.0
2 BEGINNER_TUTORIALS_SITE_METHOD = local
3 BEGINNER_TUTORIALS_SITE = $(TOPDIR)/../external/beginner_tutorials
4
5 BEGINNER_TUTORIALS_DEPENDENCIES = roscpp rospy std-msgs genmsg
6
7 ${eval ${catkin-package}}
```

3) 将beginner\_tutorials 添加入buildroot

```
@@ -46,6 +46,7 @@ source "package/rockchip/ros/cmake_modules/Config.in"
source "package/rockchip/ros/rospack/Config.in"
source "package/rockchip/ros/orocos_kinematics_dynamics/Config.in"
source "package/rockchip/ros/image-common/Config.in"
+source "package/rockchip/ros/beginner_tutorials/Config.in"
```

4) 编译

SDK根目录下source envsetup.sh, 选择rockchip\_rk3308\_release（已经跑过的，就不用再跑）。

配置make menuconfig，使用/进入搜索，搜BEGINNER\_TUTORIALS（步骤1中，定义在Config.in），并选中。

保存配置。

使用make编译。

（或则直接使用make beginner\_tutorials。重新编译使用make beginner\_tutorials-dirclean && make beginner\_tutorials）

## 4 烧写

请参考 Rockchip Linux SDK 的发布文档查看如何烧写固件。这里不再赘述。只需将编译 ROS 生成的 rootfs.img 烧到对应的 rootfs 分区即可

## 5 运行

运行 ROS 步骤如下

1、配置环境变量

```
source /opt/ros/indigo/setup.sh
```

2、运行 roscore

```
roscore &
```

3、运行代码

以上面beginner\_tutorials为例：

```
roslaunch beginner_tutorials talker
```

运行结果：

```
[ INFO] [1501923947.458788791]: hello world 0
[ INFO] [1501923947.558904332]: hello world 1
[ INFO] [1501923947.658774958]: hello world 2
```

```
[ INFO] [1501923947.758644458]: hello world 3  
[ INFO] [1501923947.858779666]: hello world 4  
[ INFO] [1501923947.958779291]: hello world 5
```

（beginner\_tutorials代码为一对程序，talker发送，listener监听，单独开talker，计算会一直累加。单独开listener无现象，同时开talker后，两个程序计算同时累加打印）