

# Rockchip Can 开发文档

---

发布版本：V1.0.0

日期：2021-01-26

文件密级：绝密 秘密 内部资料 公开

## 免责声明

本档按“现状”提供，瑞芯微电子股份有限公司（“本公司”，下同）不对本档的任何陈述、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 © 2021 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、复制本档内容的部分或全部，并不得以任何形式传播。

瑞芯微电子股份有限公司

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：+86-4007-700-590

客户服务传真：+86-591-83951833

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

## 前言

## 概述

本文提供一个标准模板供套用。后续模板以此份文档为基础改动。

## 产品版本

芯片名称	内核版本
RV1126	4.4 & 4.19

## 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
2021-01-26	V1.0.0	Elaine	第一次版本发布

# 目录

## Rockchip Can 开发文档

1. CAN 驱动
  - 1.1 驱动文件
  - 1.2 DTS 节点配置
  - 1.3 内核配置
  - 1.4 CAN 通信测试工具
  - 1.5 CAN 常用命令接口

# 1. CAN 驱动

---

## 1.1 驱动文件

驱动文件所在位置:

```
drivers/net/can/rockchip/rockchip_can.c
```

## 1.2 DTS 节点配置

主要参数:

- `interrupts = <GIC_SPI 100 IRQ_TYPE_LEVEL_HIGH>;`  
转换完成, 产生中断信号。
- `clock`

```
assigned-clocks = <&cru CLK_CAN>;
assigned-clock-rates = <200000000>;
clocks = <&cru CLK_CAN>, <&cru PCLK_CAN>;
clock-names = "baudclk", "apb_pclk";
```

时钟频率可以修改, 如果CAN的比特率1M建议修改CAN时钟到300M, 信号更稳定。低于1M比特率的, 时钟设置200M就可以。

- `pinctrl`

```
&can {
    pinctrl-names = "default";
    pinctrl-0 = <&canm0_pins>;
    status = "okay";
};
```

配置`can_h`和`can_l`的iomux作为can功能使用。

## 1.3 内核配置

```

Symbol: CAN_ROCKCHIP [=y]
      |
      | Type   : tristate
      |
      | Prompt: Rockchip CAN controller
      |
      | Location:
      |
      |   -> Networking support (NET [=y])
      |
      |     -> CAN bus subsystem support (CAN [=y])
      |
      |       -> CAN Device Drivers
      |
      |         -> Platform CAN drivers with Netlink support (CAN_DEV [=y])
      |
      | Defined at drivers/net/can/rockchip/Kconfig:1
      |
      | Depends on: NET [=y] && CAN [=y] && CAN_DEV [=y] && ARCH_ROCKCHIP [=y]

```

## 1.4 CAN 通信测试工具

canutils是常用的CAN通信测试工具包，内含 5 个独立的程序：canconfig、candump、canecho、cansend、cansequence。这几个程序的功能简述如下：

`canconfig`

用于配置 CAN 总线接口的参数，主要是波特率和模式。

`candump`

从 CAN 总线接口接收数据并以十六进制形式打印到标准输出，也可以输出到指定文件。

`canecho`

把从 CAN 总线接口接收到的所有数据重新发送到 CAN 总线接口。

`cansend`

往指定的 CAN 总线接口发送指定的数据。

`cansequence`

往指定的 CAN 总线接口自动重复递增数字，也可以指定接收模式并校验检查接收的递增数字。

`ip`

CAN波特率、功能等配置。

注意：busybox里也有集成了ip工具，但busybox里的是阉割版本。不支持CAN的操作。故使用前请先确定ip命令的版本（iproute2）。

上面工具包，网络上都有详细的编译说明。如果是自己编译buildroot，直接开启宏就可以支持上述工具包：

```

BR2_PACKAGE_CAN_UTILS=y
BR2_PACKAGE_IPROUTE2=y

```

## 1.5 CAN 常用命令接口

1. 查询当前网络设备:

```
ifconfig -a
```

2. CAN启动:

关闭CAN:

```
ip link set can0 down
```

设置比特率500KHz:

```
ip link set can0 type can bitrate 500000
```

打印can0信息:

```
ip -details link show can0
```

启动CAN:

```
ip link set can0 up
```

3. CAN发送:

发送（标准帧,数据帧,ID:123,date:DEADBEEF）:

```
cansend can0 123#DEADBEEF
```

发送（标准帧,远程帧,ID:123）:

```
cansend can0 123#R
```

发送（扩展帧,数据帧,ID:00000123,date:DEADBEEF）:

```
cansend can0 00000123#12345678
```

发送（扩展帧,远程帧,ID:00000123）:

```
cansend can0 00000123#R
```

3. CAN接收:

开启打印, 等待接收:

```
candump can0
```