

Rockchip CAN FD 开发文档

文件标识: RK-KF-YF-162

发布版本: V1.1.0

日期: 2024-03-22

文件密级: 绝密 秘密 内部资料 公开

免责声明

本文档按“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 © 2024 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。

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

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

前言

概述

产品版本

芯片名称	内核版本
RK3576	6.1

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
2024-03-22	V1.1.0	Elaine	增加RK3576

目录

Rockchip CAN FD 开发文档

1. CAN FD 驱动
 - 1.1 驱动文件
 - 1.2 DTS 节点配置
 - 1.3 内核配置
 - 1.4 CAN FD 通信测试工具
 - 1.5 CAN FD 常用命令接口
2. CAN FD 常见问题排查
 - 2.1 无法收发
 - 2.2 概率性不能收发
3. CAN FD 比特率和采样点计算
4. CAN FD 变速
 - 4.1 环路延时测试

1. CAN FD 驱动

1.1 驱动文件

驱动文件所在位置：

```
drivers/net/can/rockchip/rk3576_canfd.c
```

1.2 DTS 节点配置

主要参数：

- `interrupts = <GIC_SPI 121 IRQ_TYPE_LEVEL_HIGH>;`
转换完成，产生中断信号。
- `clock`

```
clocks = <&cru CLK_CAN0>, <&cru HCLK_CAN0>;
clock-names = "baudclk", "apb_pclk";
resets = <&cru SRST_CAN0>, <&cru SRST_H_CAN0>;
reset-names = "can", "can-apb";
```

时钟属性，用于驱动开关clk；reset属性，用于每次复位总线。

- `pinctrl`

```
&can0 {
    assigned-clocks = <&cru CLK_CAN0>;
    assigned-clock-rates = <2000000000>;
    pinctrl-names = "default";
    pinctrl-0 = <&can0m0_pins>;
    status = "okay";
};
```

配置can_h和can_l的iomux作为can功能使用。

- `rx-max-data`

```
&can0 {
    rockchip,rx-max-data = <4>;
};
```

可配置rx接收byte，如果只有CAN帧，可以配置成4（rx byte支持8），这样rx fifo深度可以有64个。如果是CANFD帧，可以配置成18（rx byte支持64），这样rx fifo深度是14帧。

- `dma`

```
&can0 {  
    dmas = <&dmac0 20>;  
    dma-names = "rx";  
};
```

rx有dma功能，但是如果不想使用dma可以删除上面属性即可。

1.3 内核配置

```
+CONFIG_CAN=y  
+CONFIG_CANFD_RK3576=y
```

1.4 CAN FD 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 FD 常用命令接口

1. 查询当前网络设备:

```
ifconfig -a
```

2. CAN FD启动:

关闭CAN:

```
ip link set can0 down
```

设置仲裁段1M波特率, 数据段3M波特率:

```
ip link set can0 type can bitrate 1000000 dbitrate 3000000 fd on
```

打印can0信息:

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

启动CAN:

```
ip link set can0 up
```

3. CAN FD发送:

发送 (标准帧,数据帧, ID:123, date:DEADBEEF) :

```
cansend can0 123##1DEADBEEF
```

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

```
cansend can0 00000123##1DEADBEEF
```

4. CAN FD接收:

开启打印, 等待接收:

```
candump can0 &
```

5. CAN FD 常用命令:

```
ip link set can0 down
ip link set can0 type can bitrate 500000 sample-point 0.8 dbitrate 2000000
sample-point 0.8 fd on
ip -details -statistics link show can0
ip link set can0 up
echo 4096 > /sys/class/net/can0/tx_queue_len
candump can0 &
```

CAN发送:

dlc 帧格式测试

扩展帧

```
cangen can0 -g 1 -e -I i -L i -D r
```

标准帧

```
cangen can0 -g 1 -I i -L i -D r
```

远程帧 标准帧

```
cangen can0 -g 1 -R -I i -L i -D r
```

远程帧 扩展帧

```
cangen can0 -g 1 -e -R -I i -L i -D r
```

```
CANFD 不变速
扩展帧
cangen can0 -g 1 -e -f -I i -L i -D i
cangen can0 -g 1 -e -f -I r -L i -D r
标准帧
cangen can0 -g 1 -f -I i -L i -D i
cangen can0 -g 1 -f -I r -L i -D r

CANFD BRS
扩展帧
cangen can0 -g 1 -e -f -b -I i -L i -D i
cangen can0 -g 1 -e -f -b -I r -L i -D r
标准帧
cangen can0 -g 1 -f -b -I i -L i -D i
cangen can0 -g 1 -f -b -I r -L i -D r
```

2. CAN FD 常见问题排查

2.1 无法收发

回环模式测试：

```
ip link set can0 down
ip link set can0 type can bitrate 500000 sample-point 0.8 dbitrate 2000000
sample-point 0.8 fd on loopback on
ip -details -statistics link show can0
ip link set can0 up
echo 4096 > /sys/class/net/can0/tx_queue_len
candump can0 &
```

回环模式下，cansend后candump可以接收，说明控制器工作正常。这种状态下，只要检查：IOMUX是否正确；硬件连接是否正确；终端120欧姆电阻有没有接入；转换芯片是否正常。

2.2 概率性不能收发

先确认比特率是否是精准的，下面命令可以看到can当前的实际比特率以及配置信息。如果比特率偏差会造成收发异常，需要根据比特率调整输入时钟，以分到精准的比特率。

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

采样点调整，上面can命令会打印当前配置的采样点，尽量保证同网络中采样点一致。可以保障收发的稳定性。

3. CAN FD 比特率和采样点计算

目前CAN架构根据输入频率和比特率自动计算。采样点的规则按照CIA标准协议：

```

/* Use CiA recommended sample points */
if (bt->sample_point) {
    sample_point_nominal = bt->sample_point;
} else {
    if (bt->bitrate > 800000)
        sample_point_nominal = 750;
    else if (bt->bitrate > 500000)
        sample_point_nominal = 800;
    else
        sample_point_nominal = 875;
}

```

比特率计算公式（详细原理可以百度，这里只介绍芯片配置相关）：

```
BitRate = clk_can / (2 * (brq + 1) / ((tseg2 + 1) + (tseg1 + 1) + 1))
```

```
Sample = (1 + (tseg1 + 1)) / (1 + (tseg1 + 1) + (tseg2 + 1))
```

brq、tseg1、tseg2 见CAN的TRM中BITTIMING寄存器。

4. CAN FD 变速

对于超过5M的变速，可能会出现不稳定，此时需要先测试环路延时，然后根据环路延时配置TDC。

4.1 环路延时测试

如下操作，读回来的0x2ac00110寄存器的值就是loop delay

```

io -4 0x2ac00108 0x80
cansend 001#aaaaaaaa
io -4 0x2ac00110

```

loop delay测试好后，配置tdc

```
io -4 0x2ac00108 0x19
```

tdc_offset 配置成loop delay的一半，使能tdc_enable，详细见trm canfd章节RKCAN_FD_TDC寄存器。最终TDC是需要代码化，写到rk3576_canfd_set_bittiming()函数的tdc配置中去。

并且要求CANFD变速网络上的采样点一致，否则brs位接收可能会有异常。采样点设置参考：

```

ip link set can0 type can bitrate 500000 sample-point 0.8 dbitrate 2000000
sample-point 0.8 fd on

```

