

Agx Xavier开发套件

Can功能使能指导手册

瑞泰新时代（北京）科技有限公司

电话: +86 010-84284669 / 84280996 / 84278927

邮箱: info@realtimes.cn

网址: <http://www.realtimes.cn>

地址: 北京市朝阳区和平西街和平西苑 20 号楼 B 座 901





手册更新历史

文档版本	更新日期	更新内容	创建人
V1.0	2020-07	创建文档	项目部

RTIMES

本指南提供了 AGX Xavier 上 can 功能使能操作指导说明

1、 注意事项

Jetson AGX Xavier 模块和开发套件上是没有 can 收发器的。

2、 CAN 接口功能使能

为了符合 Raspi 40 针标准, Jetson AGX Xavier 上的 CAN I/O 针配置为 GPIO 功能。要将引脚配置为 CAN 功能, 需要更改 pinmux 配置文件。

方法一: 配置 pinmux 文件, 需要重刷系统。

Jetson AGX Xavier 默认 pinmux_configure 文件为

Linux_for_Tegra/bootloader/t186ref/BCT/tegra19x-mb1-pinmux-p2888-0000-a04-p2822-0000-b01.cfg

更改一下内容:

```
+pinmux.0x0c303000 = 0x0000c400; # can1_dout_paa0: rsvd1, pull-down,
tristate-enable, input-enable
+pinmux.0x0c303008 = 0x0000c458; # can1_din_paa1: rsvd1, pull-down,
tristate-enable, input-enable
+pinmux.0x0c303010 = 0x0000c400; # can0_dout_paa2: rsvd1, pull-up,
tristate-enable, input-enable
+pinmux.0x0c303018 = 0x0000c458; # can0_din_paa3: rsvd1, pull-up,
tristate-enable, input-enable
```

其中 pinmux.0x0c303000 和 pinmux.0x0c303008 用于 CAN1, 而 pinmux.0x0c303010 和 pinmux.0x0c303018 用于 CAN0。

更改后, 重新使用 flash.sh 烧录设备系统。

方法二: 安装工具软件, 更改 xavier 上对应寄存器的值。

工具安装:

```
sudo ./apt-get install busybox
```

然后我们需要重新配置与 CAN 控制器引脚对应的四个寄存器的值。以下命令将重新配置这些引脚:

```
sudo busybox devmem 0x0c303000 32 0x0000C400
```

```
sudo busybox devmem 0x0c303008 32 0x0000C458
```

```
sudo busybox devmem 0x0c303010 32 0x0000C400
```

```
sudo busybox devmem 0x0c303018 32 0x0000C458
```

只在本次开机生效, 如需每次开机都生效, 需将其添加到开机自启动中。

3、 启用 CAN

1、 安装 can 驱动

```
$ sudo modprobe can
$ sudo modprobe can-raw
$ sudo modprobe can-dev
$ sudo modprobe mttcan
```

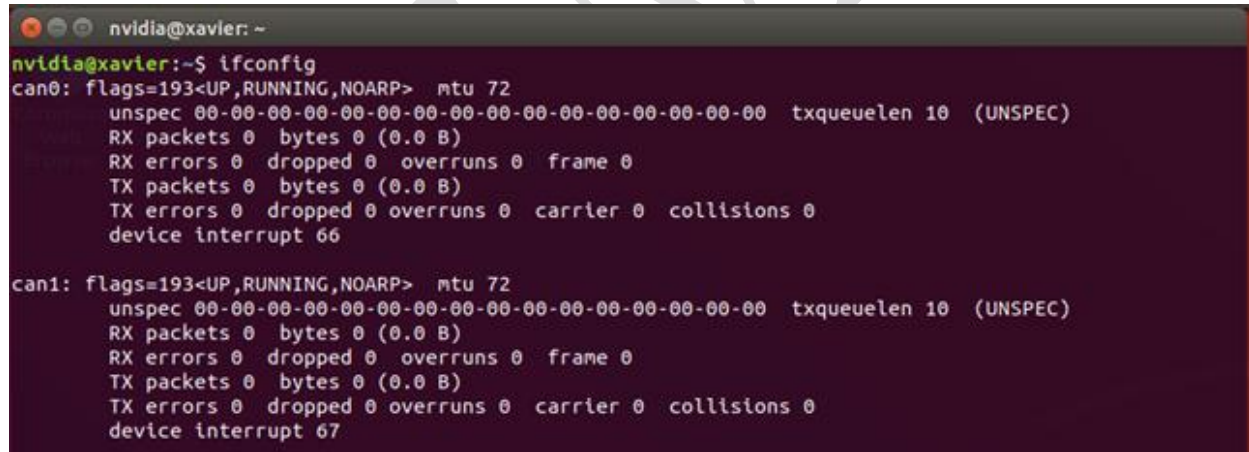
2、 下一步是根据 CAN 总线和收发器的规格配置 CAN 控制器。在这里，我将 CAN 比特率设置为 500 Kbps，启用灵活数据速率（FD），将有效负载比特率设置为 2 Mbps 并启用总线错误报告。如果您不确定所有这些设置，则可以在此之后设置 bitrate 值并忽略其他选项。

```
$ sudo ip link set can0 type can bitrate 500000 dbitrates 2000000 berr-reporting on fd on
$ sudo ip link set can1 type can bitrate 500000 dbitrates 2000000 berr-reporting on fd on
```

3、 启动 CAN 接口：

```
$ sudo ip link set up can0
$ sudo ip link set up can1
```

要检查 CAN 控制器是否已成功启用，请使用该 ifconfig 命令。在输出，你应该能够看到 can0 和 can1 网络设备之间，如图下图：



```
nvidia@xavier: ~
nvidia@xavier:~$ ifconfig
can0: flags=193<UP, RUNNING, NOARP> mtu 72
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 66
can1: flags=193<UP, RUNNING, NOARP> mtu 72
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 67
```

4、 CAN 验证

安装 can-utils

```
$ sudo apt-get install can-utils
```

发送 CAN 消息

```
$ cansend can0 123#abcdabcd
```

下面的命令可以生成随机的 CAN 消息

```
$ cangen -v can0
```

接收 CAN 消息

```
$ candump can0
```

4、 CAN 收发器的链接

3.3V	1	2	5.0V
I2C_GP5_DAT	3	4	5.0V
I2C_GP5_CLK	5	6	GND
MCLK05	7	8	UART1_TX
GND	9	10	UART1_RX
UART1_RTS	11	12	I2S2_CLK
PWM01	13	14	GND
GPIO27_PWM2	15	16	GPIO8_AO_DMIC_IN_DAT
3.3V	17	18	GPIO35_PWM3
SPI1_MOSI	19	20	GND
SPI1_MISO	21	22	GPIO17_40HEADER
SPI1_SCK	23	24	SPI1_CS0
GND	25	26	SPI1_CS1
I2C_GP2_DAT	27	28	I2C_GP2_CLK
CAN0_DIN	29	30	GND
CAN0_DOUT	31	32	GPIO9_CAN1_GPIO0_DMIC_CLK
CAN1_DOUT	33	34	GND
I2S_FS	35	36	UART1_CTS
CAN1_DIN	37	38	I2S_SDIN
GND	39	40	I2S_SDOUT



一般将扩展接头连接器的引脚#29 (CAN0_DIN) 和引脚#31 (CAN0_DOUT) 连接到其中一个收发器上的 CRX 和 CTX 引脚。同样，引脚#37 (CAN1_DIN) 和引脚#33 (CAN1_DOUT) 需要连接到另一个收发器上的 CRX 和 CTX 引脚。我们还需要将扩展插头连接器上的两组 3.3V 和 GND 引脚连接到收发器上的 3V3 和 GND 引脚。收发器上的 CANH 和 CANL 引脚需要连接到要与之通信的两条 CAN 总线的 CAN 高电平和 CAN 低电平线。