

20 路 CANFD 通信模块 FL1042 用户手册

Rev 1.0



版权声明:

Copyright ©2012-2018 芯驿电子科技（上海）有限公司

公司网址:

[Http://www.alinx.com.cn](http://www.alinx.com.cn)

技术论坛:

<http://www.heijin.org>

官方旗舰店:

<http://alinx.jd.com>

邮箱:

avic@alinx.com.cn

电话:

021-67676997

传真:

021-37737073

ALINX 微信公众号:



文档修订记录:

版本	时间	描述
1.0	2024/9/24	First Release

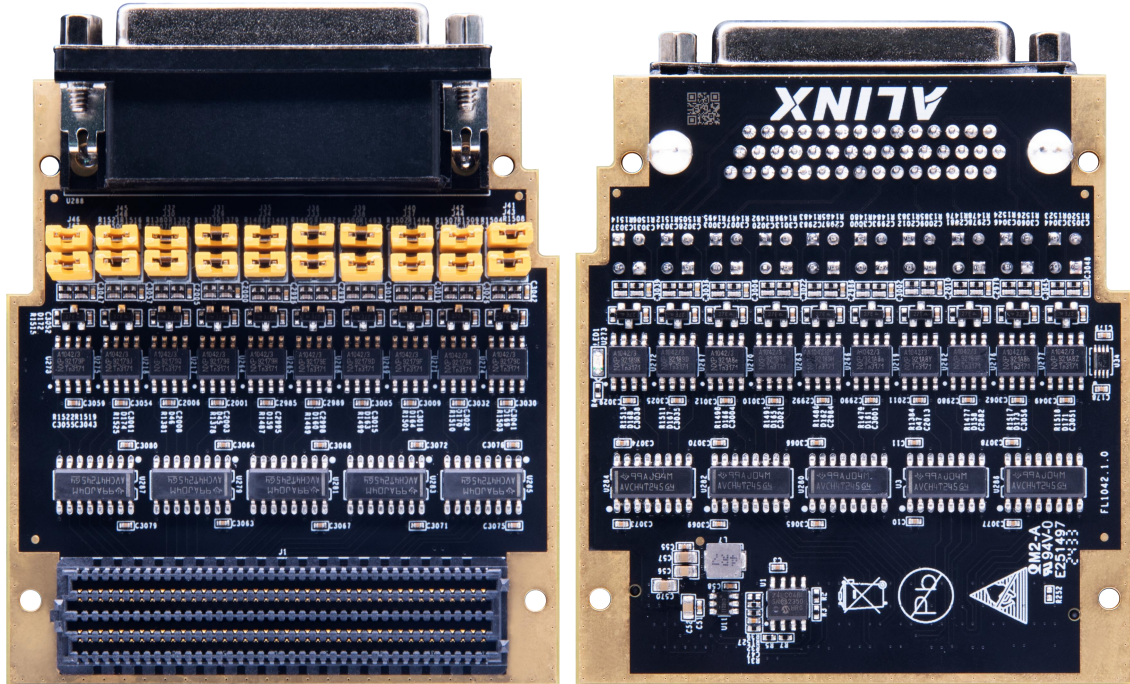
目录 Table of Contents

第一部分 FL042 模块说明介绍	5
1.1 FL1042 模块的参数说明	5
1.2 FL1042 模块的结构图	6
第二部分 模块功能说明	6
2.1 FL1042 模块原理框图	6
2.2 模块 FMC LPC 的引脚分配:	7
2.3 DB44 母头接口分配	8
第三部分 FMC 模块安装和测试	10
3.1 软件测试	11
3.2 测试过程	11
3.3 测试结果	12

第一部分 FL042 模块说明介绍

FL1042 为 20 路 CANFD 通信模块，CANFD 信号通过 DB44 母头连接器和外部信号通信，同时预留了跳帽使能 CANFD 总线终端电阻。CANFD 收发芯片选用了恩智浦公司的 TJA1042T/3/1J 芯片为用户提供 CAN 通信服务。

FL1042 模块实物照片如下：



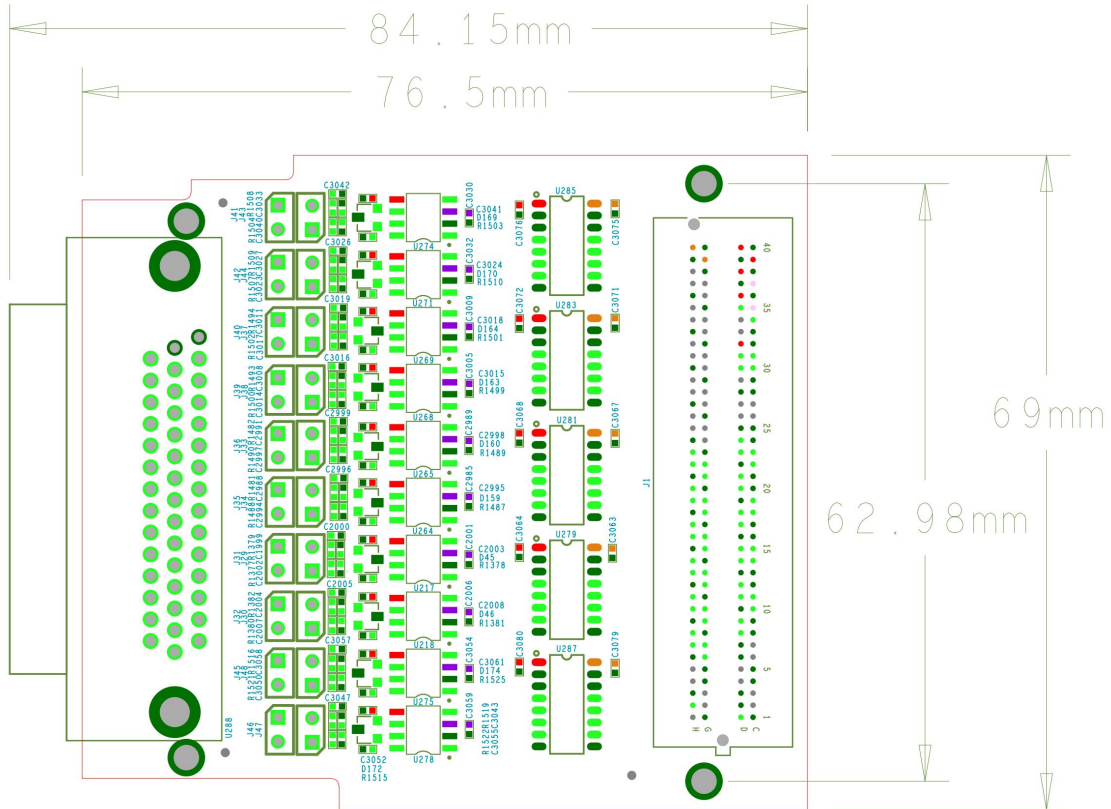
FL1042 模块实物图

1.1 FL1042 模块的参数说明

以下为 FL1042 20 路 CANFD 通信模块的详细参数：

- 通信接口：20 路 CANFD 通信接口（DB44 母头连接器）
- 使用车规 TJA1042 芯片，速度高达 1MBit/s。
- FMC 类型：LPC

1.2 FL1042 模块的结构图

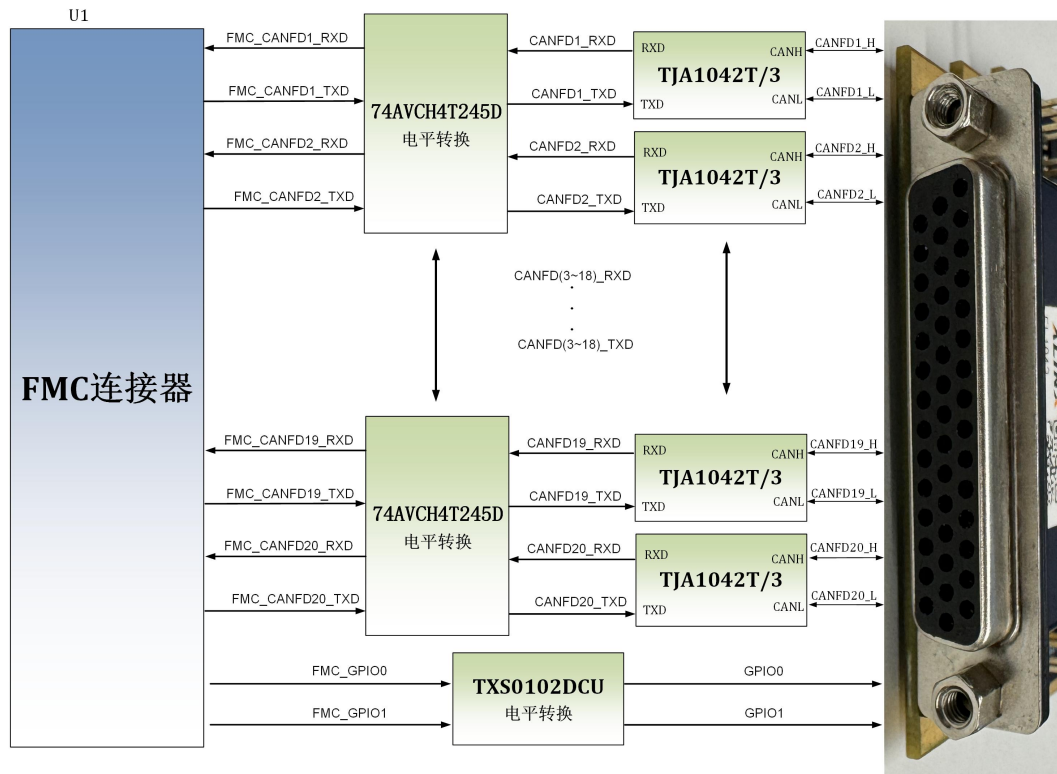


FL1042 模块尺寸结构图

第二部分 模块功能说明

2.1 FL1042 模块原理框图

FL1042 模块的原理设计框图如下:



FL1042 模块设计框图

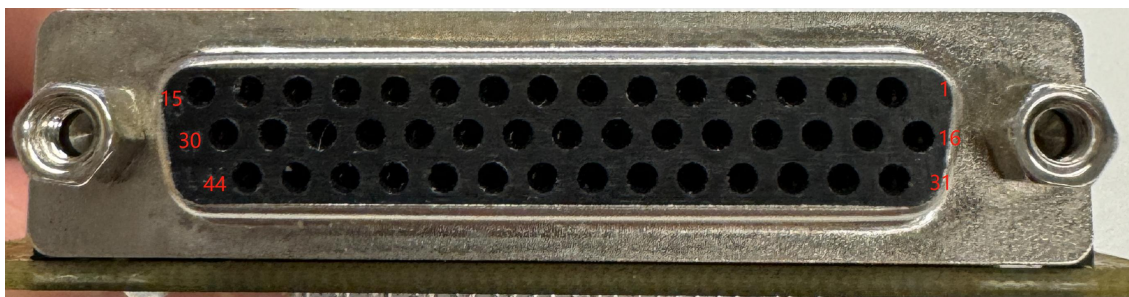
2.2 模块 FMC LPC 的引脚分配:

下面只列了光模块的信号，电源和 GND 的信号并没有列出，详细管脚分配参考原理图。下图为 FL1042 模块的管脚分配。

FMC 管脚序号	网络名	FMC 管脚信号	说明
D11	FMC_CANFD_TXD1	LA05_P	CANFD1 信号发送端
D12	FMC_CANFD_RXD1	LA05_N	CANFD1 信号接收端
C10	FMC_CANFD_TXD2	LA06_P	CANFD2 信号发送端
C11	FMC_CANFD_RXD2	LA06_N	CANFD2 信号接收端
C14	FMC_CANFD_TXD3	LA10_P	CANFD3 信号发送端
C15	FMC_CANFD_RXD3	LA10_N	CANFD3 信号接收端
H10	FMC_CANFD_TXD4	LA04_P	CANFD4 信号发送端
H11	FMC_CANFD_RXD4	LA04_N	CANFD4 信号接收端
G15	FMC_CANFD_TXD5	LA12_P	CANFD5 信号发送端
G16	FMC_CANFD_RXD5	LA12_N	CANFD5 信号接收端
G12	FMC_CANFD_TXD6	LA08_P	CANFD6 信号发送端
G13	FMC_CANFD_RXD6	LA08_N	CANFD6 信号接收端
H13	FMC_CANFD_TXD7	LA07_P	CANFD7 信号发送端
H14	FMC_CANFD_RXD7	LA07_N	CANFD7 信号接收端
D14	FMC_CANFD_TXD8	LA09_P	CANFD8 信号发送端
D15	FMC_CANFD_RXD8	LA09_N	CANFD8 信号接收端

C18	FMC_CANFD_TXD9	LA14_P	CANFD9 信号发送端
C19	FMC_CANFD_RXD9	LA14_N	CANFD9 信号接收端
H16	FMC_CANFD_TXD10	LA11_P	CANFD10 信号发送端
H17	FMC_CANFD_RXD10	LA11_N	CANFD10 信号接收端
D17	FMC_CANFD_TXD11	LA13_P	CANFD11 信号发送端
D18	FMC_CANFD_RXD11	LA13_N	CANFD11 信号接收端
G18	FMC_CANFD_TXD12	LA16_P	CANFD12 信号发送端
G19	FMC_CANFD_RXD12	LA16_N	CANFD12 信号接收端
C22	FMC_CANFD_TXD13	LA18_P_CC	CANFD13 信号发送端
C23	FMC_CANFD_RXD13	LA18_N_CC	CANFD13 信号接收端
G21	FMC_CANFD_TXD14	LA20_P	CANFD14 信号发送端
G22	FMC_CANFD_RXD14	LA20_N	CANFD14 信号接收端
D20	FMC_CANFD_TXD15	LA17_P_CC	CANFD15 信号发送端
D21	FMC_CANFD_RXD15	LA17_N_CC	CANFD15 信号接收端
H19	FMC_CANFD_TXD16	LA15_P	CANFD16 信号发送端
H20	FMC_CANFD_RXD16	LA15_N	CANFD16 信号接收端
G9	FMC_CANFD_TXD17	LA03_P	CANFD17 信号发送端
G10	FMC_CANFD_RXD17	LA03_N	CANFD17 信号接收端
H7	FMC_CANFD_TXD18	LA02_P	CANFD18 信号发送端
H8	FMC_CANFD_RXD18	LA02_N	CANFD18 信号接收端
G6	FMC_CANFD_TXD19	LA00_P_CC	CANFD19 信号发送端
G7	FMC_CANFD_RXD19	LA00_N_CC	CANFD19 信号接收端
D8	FMC_CANFD_TXD20	LA01_P_CC	CANFD20 信号发送端
D9	FMC_CANFD_RXD20	LA01_N_CC	CANFD20 信号接收端
H22	FMC_GPIO0	LA19_P	GPIO 使能管脚 0
H23	FMC_GPIO1	LA19_N	GPIO 使能管脚 1
C30	SCL	SCL	EEPROM 的 I2C 时钟
C31	SDA	SDA	EEPROM 的 I2C 数据

2.3 DB44 母头接口分配

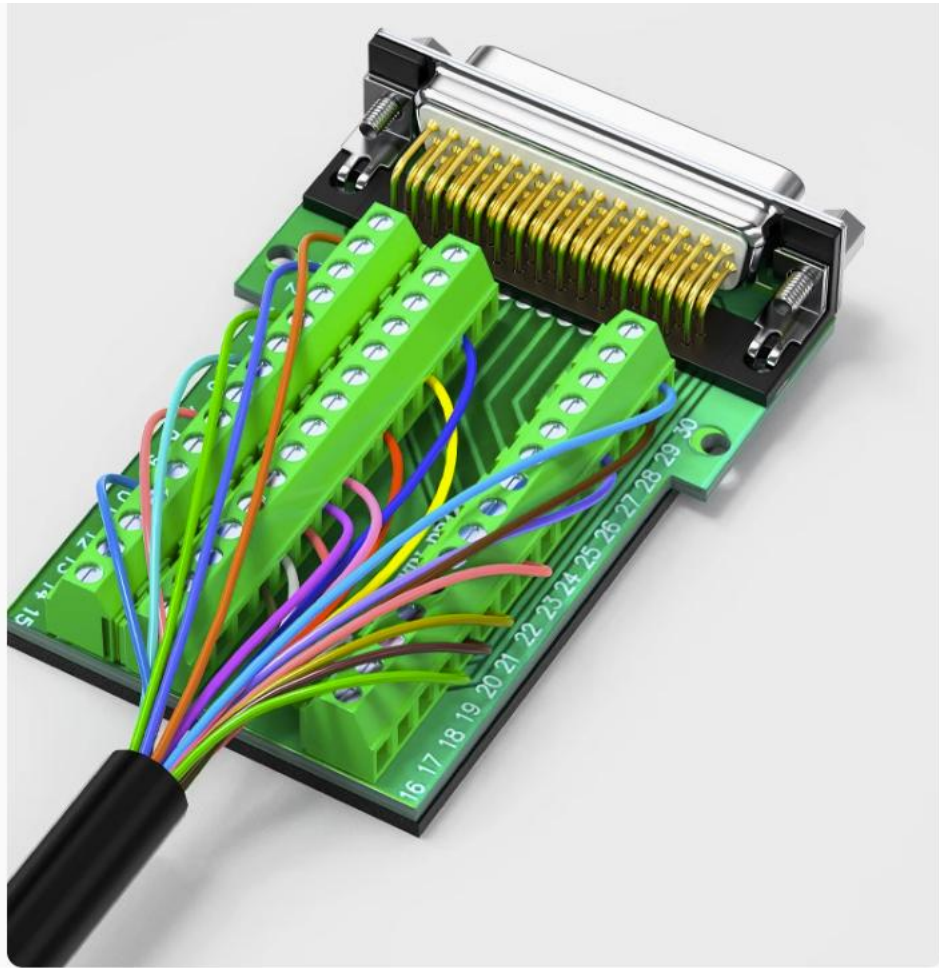


DB44 母头管脚序号	网络名	说明
1	CANFD20_H	CANFD20 高电平信号
2	CANFD20_L	CANFD20 低电平信号

3	CANFD2_H	CANFD2 高电平信号
4	CANFD2_L	CANFD2 低电平信号
5	CANFD1_H	CANFD1 高电平信号
6	CANFD1_L	CANFD1 低电平信号
7	CANFD6_H	CANFD6 高电平信号
8	CANFD6_L	CANFD6 低电平信号
9	CANFD5_H	CANFD5 高电平信号
10	CANFD5_L	CANFD5 低电平信号
11	CANFD10_H	CANFD10 高电平信号
12	CANFD10_L	CANFD10 低电平信号
13	CANFD9_H	CANFD9 高电平信号
14	CANFD9_L	CANFD9 低电平信号
15	GND	地信号
16	CANFD19_H	CANFD19 高电平信号
17	CANFD19_L	CANFD19 低电平信号
18	CANFD4_L	CANFD4 低电平信号
19	CANFD4_H	CANFD4 高电平信号
20	CANFD3_L	CANFD3 低电平信号
21	CANFD3_H	CANFD3 高电平信号
22	CANFD11_L	CANFD11 低电平信号
23	CANFD11_H	CANFD11 高电平信号
24	CANFD12_L	CANFD12 低电平信号
25	CANFD12_H	CANFD12 高电平信号
26	CANFD14_L	CANFD14 低电平信号
27	CANFD14_H	CANFD14 高电平信号
28	CANFD16_H	CANFD16 高电平信号
29	CANFD16_L	CANFD16 低电平信号
30	GND	地信号
31	GPIO0	GPIO 使能管脚 0
32	GPIO1	GPIO 使能管脚 1
33	CANFD18_H	CANFD18 高电平信号
34	CANFD18_L	CANFD18 低电平信号
35	CANFD17_L	CANFD17 低电平信号
36	CANFD17_H	CANFD17 高电平信号
37	CANFD7_L	CANFD7 低电平信号
38	CANFD7_H	CANFD7 高电平信号
39	CANFD8_L	CANFD8 低电平信号
40	CANFD8_H	CANFD8 高电平信号
41	CANFD13_H	CANFD13 高电平信号
42	CANFD13_L	CANFD13 低电平信号
43	CANFD15_H	CANFD15 高电平信号
44	CANFD15_L	CANFD15 低电平信号

与 FL1042 模块 DB44 母头对接的公头模块可以根据需要购买，购买链接

https://detail.tmall.com/item.htm?id=651040594908&app=chrome&bxsign=scdg4XYBIzBLBaY4Q-crBoDJ_RDtYfDMQxqAr3qj4oHCoNrq6mFg_eLae8xrddG9j2m6buzx7Pa2tY5NO5S6eFAVKOgcCHRkBINVIsRQ0QeqIV1BZ4QWAD7Nh67TOZLWxGY&cpp=1&share crt_v=1&shareurl=true&short_name=h.g2pNWivZblk107b&sp tk=M3NVMFd1bkdvSnQ=&spm=a2159r.13376460.0.0&tbSocialPopKey=shareItem&tk=3sU0WunGoJt&un=3a99f6caf591640d4b68d5f77a161062&un_site=0&ut_sk=1.ZSPP2lo1gCcDAAi61yL9RdTv_21380790_1715738970335.TaoPassword-Weixin.1



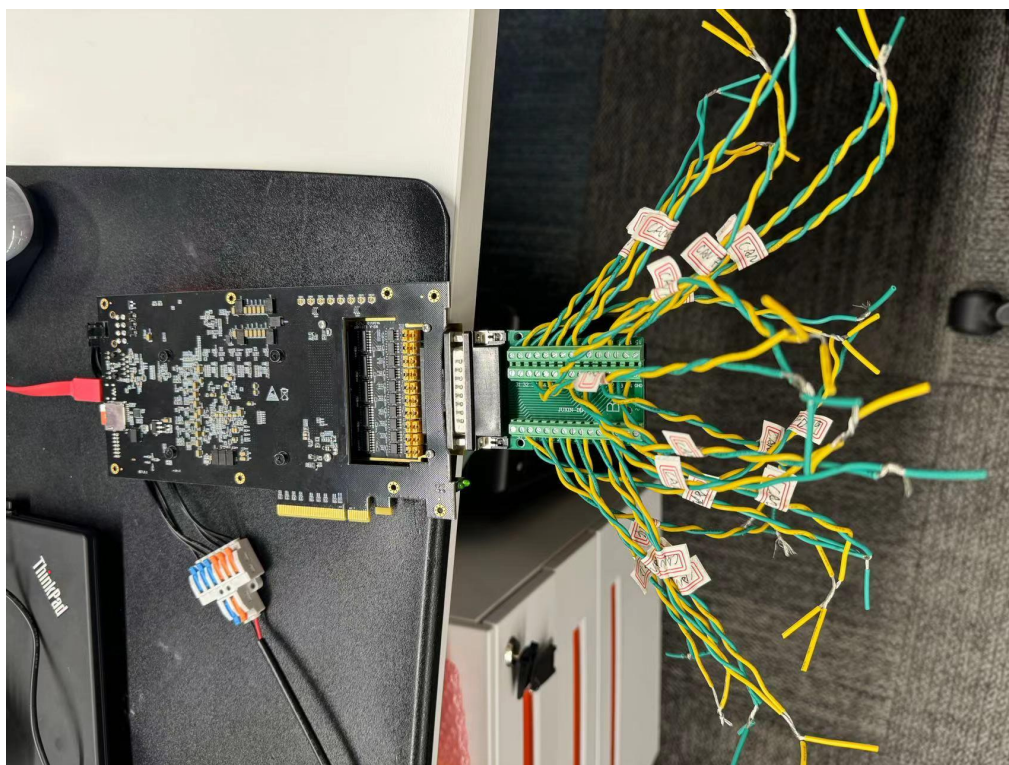
第三部分 FMC 模块安装和测试

FL1042 模块和 FPGA 开发板的硬件连接很简单，只要把 FMC 接口跟开发板的 FMC 接口对插就可以，然后用螺丝固定。以下为 BZ7 开发板的和 FL042 模块的硬件连接图：



3.1 软件测试

将 FL1042 模块插在 BZ7 开发板上, 使用跳帽将 20 个电阻使能, 将 can0 和 can1 的高对高, 低对低连接, can2 和 can3、can4 和 can5.....同上, 将 20 路 can 都两两连接



3.2 测试过程

- 使能 FL1042 模块

通过 GPIO 使能 FL1042 模块

```

root@petalinux:~# cat start.sh
#!/bin/sh

echo $1 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio$1/direction
echo 0 > /sys/class/gpio/gpio$1/value
root@petalinux:~#
root@petalinux:~# ./start.sh 364
root@petalinux:~#

```

- 测试 CANFD 信号的收发

先使能互相连接的两路 can，然后一个发送数据，另一个接收数据并保存下来，查看保存的文件数据是否正确来验证 can 收发是否正常

```

root@petalinux:~# cat cantest.sh
ip link set can$1 down
ip link set can$1 up type can bitrate 1000000 sample-point 0.875 dbitrate 1000000 dsample-point 0.875 fd on
ip link set can$2 down
ip link set can$2 up type can bitrate 1000000 sample-point 0.875 dbitrate 1000000 dsample-point 0.875 fd on
candump can$2 >> ./can$2_rec.log &
cansend can$1 12345678##1000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
cansend can$1 5A1#11.2233.44556677.88

root@petalinux:~# ./cantest.sh 0 1
[ 2033.680959] IPv6: ADDRCONF(NETDEV_CHANGE): can0: link becomes ready
root@petalinux:~# [ 2034.686306] IPv6: ADDRCONF(NETDEV_CHANGE): can1: link becomes ready

root@petalinux:~# cat can1_rec.log
can1 12345678 [64] 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 00 01 02 03 04 05 06 07 08 09 0A 0B
0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
can1 5A1 [8] 11 22 33 44 55 66 77 88
can1 12345678 [64] 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 00 01 02 03 04 05 06 07 08 09 0A 0B
0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
can1 5A1 [8] 11 22 33 44 55 66 77 88
root@petalinux:~#

root@petalinux:~# cat can3_rec.log
can3 12345678 [64] 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 00 01 02 03 04 05 06 07 08 09 0A 0B
0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
can3 5A1 [8] 11 22 33 44 55 66 77 88
root@petalinux:~#

root@petalinux:~# cat can5_rec.log
can5 12345678 [64] 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 00 01 02 03 04 05 06 07 08 09 0A 0B
0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
can5 5A1 [8] 11 22 33 44 55 66 77 88
root@petalinux:~#

root@petalinux:~# cat can7_rec.log
can7 12345678 [64] 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 00 01 02 03 04 05 06 07 08 09 0A 0B
0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
can7 5A1 [8] 11 22 33 44 55 66 77 88
root@petalinux:~#

root@petalinux:~# cat can9_rec.log
can9 12345678 [64] 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 00 01 02 03 04 05 06 07 08 09 0A 0B
0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
can9 5A1 [8] 11 22 33 44 55 66 77 88
root@petalinux:~#

root@petalinux:~# cat can11_rec.log
can11 12345678 [64] 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 00 01 02 03 04 05 06 07 08 09 0A 0B
0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
can11 5A1 [8] 11 22 33 44 55 66 77 88
root@petalinux:~#

root@petalinux:~# cat can13_rec.log
can13 12345678 [64] 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 00 01 02 03 04 05 06 07 08 09 0A 0B
0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
can13 5A1 [8] 11 22 33 44 55 66 77 88
root@petalinux:~#

root@petalinux:~# cat can15_rec.log
can15 12345678 [64] 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 00 01 02 03 04 05 06 07 08 09 0A 0B
0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
can15 5A1 [8] 11 22 33 44 55 66 77 88
root@petalinux:~#

root@petalinux:~# cat can17_rec.log
can17 12345678 [64] 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 00 01 02 03 04 05 06 07 08 09 0A 0B
0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
can17 5A1 [8] 11 22 33 44 55 66 77 88
root@petalinux:~#

root@petalinux:~# cat can19_rec.log
can19 12345678 [64] 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 00 01 02 03 04 05 06 07 08 09 0A 0B
0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
can19 5A1 [8] 11 22 33 44 55 66 77 88
root@petalinux:~#

```

3.3 测试结果

由上图可知，保存下的文件中数据皆正常，因此 20 路 can 收发功能正常。