1 SDIO Loopback Test Guide

1.1 Requirement

1.1.1 Hardware requirement

Table 1-1 Hardware requirement

Host	Raspberry Pi 3B+ with Linux raspberrypi 6.6.28+rpt-rpi-v7
Device	AmebaDPlus EVB

NOTE

a) *Connect AmebaDPlus EVB to Raspberry Pi 3B+ as Figure 1-2 shows.*

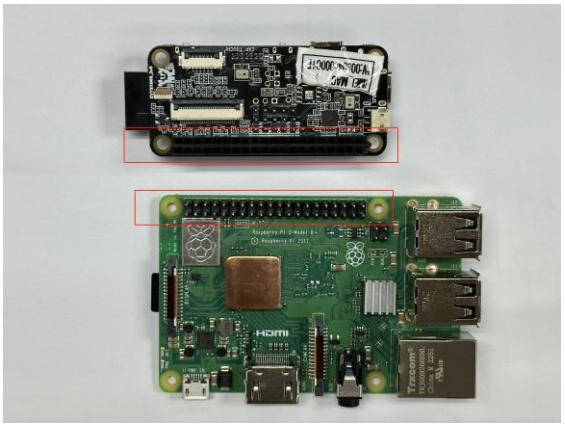

Figure 1-1 AmebaDPlus and Raspberry Pi 3B+

Figure 1-2 Connect AmebaDPlus to Raspberry Pi 3B+

b) *If PC supporting SDIO acts as SDIO host, it is recommended to connect AmebaDPlus EVB to PC by adapter PCB as Figure 1-3 shows instead of by Dupond line. Contact HW for how to modify circuit.*

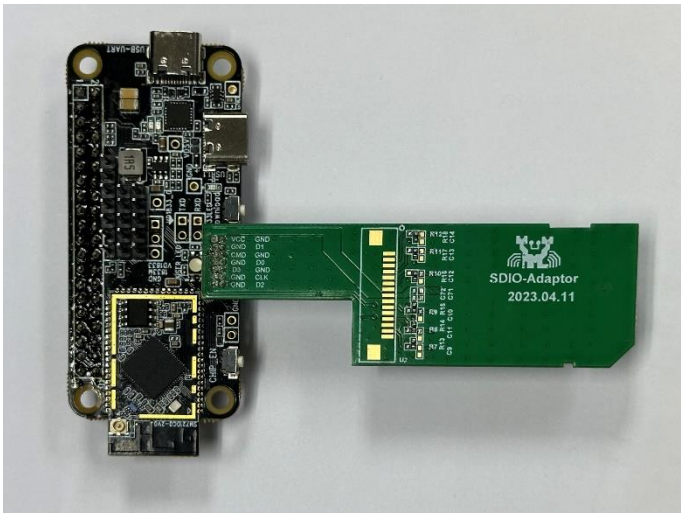

Figure 1-3 Connect AmebaDPlus to PC by adapter PCB

1.1.2 Software requirement

Table 1-2 Software requirement

Host	Host driver SDK
Device	AmebaDPlus SDK

NOTE

Host driver SDK is based on the kernel version mentioned in Table 1-1. If there are compile errors caused by different kernel version, user should solve these errors.

1.2 Operation Flow

CAUTION

If Raspberry Pi acts as SDIO host, it is required to connect AmebaDPlus EVB to Raspberry Pi before powering on Raspberry Pi and disconnect them after powering off Raspberry Pi.

1.2.1 Compile

1.2.1.1 Device

Copy component/example/peripheral/mbed/SDIO/sdio_device_loopback/src/main.c to path amebadplus_gcc_project/project_km4/src/, and replace old main.c.
Return to path *amebadplus_gcc_project/*. Run "make all" to generate image km4_boot_all.bin and km0_km4_app.bin.
Refer to component/example/peripheral/mbed/SDIO/sdio_device_loopback/README.md for more information.

1.2.1.2 Host

Copy component/example/peripheral/mbed/SDIO/sdio_device_loopback_Raspberry folder to Raspberry Pi.
(1) Switch to path *sdio_device_loopback_Raspberry/*. Run "make all" to generate driver rtk_sdio.ko.

```
wlan5@raspberrypi:~/sdio_device_loopback_Raspberry $ make all
#make -C /lib/modules/6.6.28+rpt-rpi-v7/build SUBDIRS=/home/wlan5/sdio_device_loopback_Raspberry clean
make -C /lib/modules/6.6.28+rpt-rpi-v7/build M=/home/wlan5/sdio_device_loopback_Raspberry clean
make[1]: Entering directory '/usr/src/linux-headers-6.6.28+rpt-rpi-v7'
  CLEAN   /home/wlan5/sdio_device_loopback_Raspberry/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-6.6.28+rpt-rpi-v7'
#make -C /lib/modules/6.6.28+rpt-rpi-v7/build SUBDIRS=/home/wlan5/sdio_device_loopback_Raspberry modules
make -C /lib/modules/6.6.28+rpt-rpi-v7/build M=/home/wlan5/sdio_device_loopback_Raspberry modules
make[1]: Entering directory '/usr/src/linux-headers-6.6.28+rpt-rpi-v7'
make[3]: Warning: File '/home/wlan5/sdio_device_loopback_Raspberry/src/sdio_ops.c' has modification time 386174 s in the future
  CC [M]  /home/wlan5/sdio_device_loopback_Raspberry/src/sdio_ops.o
  CC [M]  /home/wlan5/sdio_device_loopback_Raspberry/src/sdio_intf.o
  CC [M]  /home/wlan5/sdio_device_loopback_Raspberry/src/sdio_chardev.o
  CC [M]  /home/wlan5/sdio_device_loopback_Raspberry/src/sdio_loopback.o
  LD [M]  /home/wlan5/sdio_device_loopback_Raspberry/rtk_sdio.o
make[3]: warning:  Clock skew detected.  Your build may be incomplete.
  MODPOST /home/wlan5/sdio_device_loopback_Raspberry/Module.symvers
  CC [M]  /home/wlan5/sdio_device_loopback_Raspberry/rtk_sdio.mod.o
  LD [M]  /home/wlan5/sdio_device_loopback_Raspberry/rtk_sdio.ko
make[1]: Leaving directory '/usr/src/linux-headers-6.6.28+rpt-rpi-v7'
wlan5@raspberrypi:~/sdio_device_loopback_Raspberry $
```

Figure 1-4 Generate driver file

(2) Switch to path *test/*. Run "make" to generate sdio_cmd_console.

```
wlan5@raspberrypi:~/sdio_device_loopback_Raspberry/test $ make
gcc -c main.c
gcc -o sdio_cmd_console main.o
wlan5@raspberrypi:~/sdio_device_loopback_Raspberry/test $
```

Figure 1-5 Generate console file

1.2.2 Configure

1.2.2.1 Device

Download images in 1.2.1.1 to AmebaDPlus EVB by Ameba Image Tool.
Reset AmebaDPlus and the following log will be displayed. SDIO device will wait until host starts loopback test.

```
[SPDIO-I] SDIO_Device_Init==>
[SPDIO-I] TXBDWPtr=0x0 TXBDRPtr=0x0
[SPDIO-I] TX_BD0 @ 2001A2A0 20019F60
[SPDIO-I] TX_BD1 @ 2001A2AC 20019F64
[SPDIO-I] TX_BD2 @ 2001A2B8 20019F68
[SPDIO-I] TX_BD3 @ 2001A2C4 20019F6C
[SPDIO-I] TX_BD4 @ 2001A2D0 20019F70
[SPDIO-I] TX_BD5 @ 2001A2DC 20019F74
[SPDIO-I] TX_BD6 @ 2001A2E8 20019F78
[SPDIO-I] TX_BD7 @ 2001A2F4 20019F7C
[SPDIO-I] TX_BD8 @ 2001A300 20019F80
[SPDIO-I] TX_BD9 @ 2001A30C 20019F84
[SPDIO-I] TX_BD10 @ 2001A318 20019F88
[SPDIO-I] TX_BD11 @ 2001A324 20019F8C
[SPDIO-I] TX_BD12 @ 2001A330 20019F90
[SPDIO-I] TX_BD13 @ 2001A33C 20019F94
[SPDIO-I] TX_BD14 @ 2001A348 20019F98
[SPDIO-I] TX_BD15 @ 2001A354 20019F9C
[SPDIO-I] TX_BD16 @ 2001A360 20019FA0
[SPDIO-I] TX_BD17 @ 2001A36C 20019FA4
[SPDIO-I] TX_BD18 @ 2001A378 20019FA8
[SPDIO-I] TX_BD19 @ 2001A384 20019FAC
[SPDIO-I] <==SDIO_Device_Init
SDIO device starts loopback!
```

Figure 1-6 SDIO device reset log

1.2.2.2 Host

Configure SDIO host as the following:

(1) Open a new console window. Run "dmesg -w" to monitor and display kernel messages in real-time.

(2) Switch to path *sdio_device_loopback_Raspberry/*. Run "insmod rtk_sdio.ko" to insert kernel module file. When SDIO device is detected, the following log will be displayed on the dmesg window.

```
[  288.977124] rtk_sdio: loading out-of-tree module taints kernel.
[  288.978028] rtk_sdio module init start
[  288.978252] rtk_sdio_probe():++
[  288.978260] vendor=0x024c device=0x8722 class=0x07
[  288.978674] Create chardev (major=239 minor=0) OK.
[  288.979583] probe sdio device success
[  288.979730] rtk_sdio module init ret=0
```

Figure 1-7 SDIO device probe success

(3) Return to path *test/*. Run "./sdio_cmd_console" to enter console.

1.2.3 Loopback Test

1.2.3.1 Host

(1) Type in "loopback start" to start data loopback with SDIO device. Log will be displayed on dmesg window.

```
AT: loopback start
```

Figure 1-8 Type in "loopback start" in console window

```
[  379.178593] rtw_sdio_loopback_start
[  379.179268] start rtw_sdio_loopback_thread
[  379.179531] Send successed [1]!, sz=1500
[  379.180245] recv successed [1], size=1500!
[  379.392708] Send successed [2]!, sz=1500
[  379.393219] recv successed [2], size=1500!
[  379.612708] Send successed [3]!, sz=1500
[  379.613194] recv successed [3], size=1500!
[  379.832707] Send successed [4]!, sz=1500
[  379.833204] recv successed [4], size=1500!
[  380.052794] Send successed [5]!, sz=1500
[  380.053386] recv successed [5], size=1500!
[  380.272783] Send successed [6]!, sz=1500
```

Figure 1-9 Loopback test log in dmesg window

(2) Type in "loopback stop" to stop data loopback with SDIO device.

```
AT: loopback stop
```

Figure 1-10 Type in "loopback stop" in console window

```
[  385.772613] exit rtw_sdio_loopback_thread
```

Figure 1-11 Loopback end log in dmesg window

(3) Type in "sdio [func] [read/write] [addr] [len] ([val])" to access SDIO device registers.
Refer to the annotation of function sdio_ctl in */test/main.c* to get more information.

```
AT: sdio 1 0 0x20 2
value: 0x00000014
```

Figure 1-12 Type in read cmd in console window

```
[  393.395741] DBG: 1, 0, 20, 2
[  393.395779] rtk_sdio_dbg: fn: 1 read 2 bytes from register @ 0x0020
[  393.395986] rtk_sdio_dbg: fn: 1 register(offset: 0x0020): 0x00000014
```

Figure 1-13 SDIO register information in dmesg window

1.2.3.2 Device

After SDIO host starts loopback test, log of SDIO device will show in Ameba Trace Tool.

```
SDIO device starts loopback!
loopback package, size = 1484 (cnt = 1) heap=319424
[SPDIO-I] SDIO_Return_Rx_Data(2)<==
loopback package, size = 1484 (cnt = 2) heap=319424
[SPDIO-I] SDIO_Return_Rx_Data(2)<==
loopback package, size = 1484 (cnt = 3) heap=319424
[SPDIO-I] SDIO_Return_Rx_Data(2)<==
loopback package, size = 1484 (cnt = 4) heap=319424
[SPDIO-I] SDIO_Return_Rx_Data(2)<==
loopback package, size = 1484 (cnt = 5) heap=319424
[SPDIO-I] SDIO_Return_Rx_Data(2)<==
loopback package, size = 1484 (cnt = 6) heap=319424
[SPDIO-I] SDIO_Return_Rx_Data(2)<==
loopback package, size = 1484 (cnt = 7) heap=319424
[SPDIO-I] SDIO_Return_Rx_Data(2)<==
loopback package, size = 1484 (cnt = 8) heap=319424
[SPDIO-I] SDIO_Return_Rx_Data(2)<==
loopback package, size = 1484 (cnt = 9) heap=319424
[SPDIO-I] SDIO_Return_Rx_Data(2)<==
loopback package, size = 1484 (cnt = 10) heap=319424
[SPDIO-I] SDIO_Return_Rx_Data(2)<==
```

Figure 1-14 SDIO device loopback test log

1.3 Trouble Shooting

1.3.1 Host cannot probe inserted device

1.3.1.1 Description

When AmebaDPlus EVB is connected to Raspberry Pi, SDIO host cannot probe connected SDIO host. After resetting AmebaDPlus EVB, host still cannot probe device.

1.3.1.2 Reason

Raspberry Pi is configured to detect SDIO device only once during boot.

1.3.1.3 Solution

Add "dtoverlay=sdio, poll_once=off" to */boot/firmware/config.txt* as Figure 1-15 shows and reset Raspberry Pi, so that polling mode of SDIO host will be enabled to check whether SDIO device is inserted every 1s.

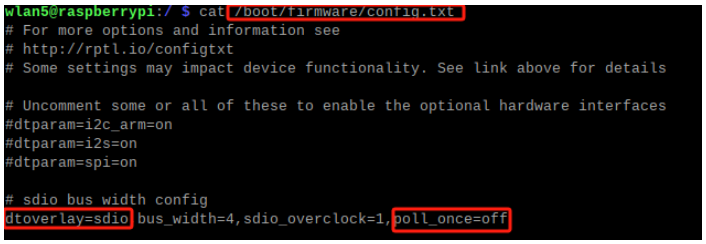

Figure 1-15 Modify configuration file

1.4 Revision History

Date	Version	Author	Revision Log
2024/11/7	R03	Katherine_wang	Add details of hardware connection due to modification of EVB.
2024/10/18	R02	Katherine_wang	Update file directory and add details of trouble shooting
2024/06/12	R01	katherine_wang	Draft